8-2024

Assessing the Efficacy: Statistical Models vs. Machine Learning (ML) Approaches for Prediction Modeling In the Construction Industry

Stuti Garg
Clemson University, stutig@clemson.edu

ASSESSING THE EFFICACY: STATISTICAL MODELS VS. MACHINE LEARNING (ML) APPROACHES FOR PREDICTION MODELING IN THE CONSTRUCTION INDUSTRY

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Construction Science and Management

---

by
Stuti Garg
August 2024

Accepted by:
Dr. Vivek Sharma, Committee Chair
Dr. Dhaval Gajjar, Committee Co-Chair
Dr. N. Mike Jackson, Committee Member
Dr. Jason D. Lucas, Committee Member
Dr. Jong Han Yoon, Committee Member

ABSTRACT

The construction industry has witnessed a significant surge in the daily volume of objective data (i.e., precise data sources representing actual project progress) accumulated across a project's lifecycle. This abundance of data presents an opportunity to extract valuable organizational insights and potential remedies for project management issues. The construction technology landscape is gradually evolving towards integrated software platforms to meet customer needs more effectively. However, the construction sector lacks comprehensive predictive analytics solutions for projects or industry-wide applications - a significant portion of descriptive analytics tools rely on trade association surveys or dashboards constructed from collected company data, suffering from infrequent updates or limited detail. Machine learning (ML) has been experiencing increased adoption within the construction sector. This technology is bringing transformations across various aspects of construction project management, such as risk assessment and mitigation, safety management on construction sites, cost estimation and forecasting, schedule management, and the prediction of building energy demand. The study's research objectives are to assess the efficacy of statistical models vis-à-vis ML-based approaches for prediction modeling in the construction industry by the following:

(1) MEASURE the outcomes of the “customer satisfaction” (for the construction coatings sector) prediction model for both ST and ML approaches.

(2) Compare ST vis-à-vis ML-based models predicting customer satisfaction in the construction coatings sector for a non-parametric dataset with limited dimensions.

(3) DEVELOP a norm for handling non-parametric data with limited dimensions for the construction coatings sector.

The norms can assist in decision-making for selecting a method for prediction – statistical or machine learning based on the nature of the dataset, nature of independent variables, nature of factors contributing to the outcome, goal of analysis, and prediction, specifically non-parametric dataset with limited dimensions. These lessons learned can yield substantial advantages for businesses by improving the performance measures of construction projects—a critical measure of project success. This shall benefit industry data analysts conducting project feasibility studies, giving them insights for improved budget allocation and portfolio management plans.

DEDICATION

The thesis is dedicated to my family's unwavering support and unconditional love, whose influence and guidance shaped my academic journey and personal growth. First, to my daughter, Tashya, who has been a faithful companion through this journey and has given me the strength and bolstered my belief in myself daily! To my mother, my role model, a distinguished Professor in Biochemistry at the Punjab University, India, whose passion for science and tireless dedication to education and research have been a constant source of inspiration. Her intellectual mentorship, resilience, and being my best friend through the rollercoaster of life have been my anchors throughout my endeavors. To my father, a retired scientist from the Department of Space, India, whose discipline and commitment to work have instilled in me essential skills and the willingness to excel in every project. His profound insights, visionary planning and excellence have set the benchmark for my aspirations. My brother and sister-in-law have been my constant cheerleaders and my dearest friends. They have taken the utmost care of me, sitting miles away in their unique ways. They have been my guardian angels, watching out for me and giving me more than one reason to be happy and joyous. My niece, Avya, deserves a special mention since her cutest videos have been most therapeutic, joy-giving and calming in stressful times. Above all, I am grateful to my guru, my light, who has guided, blessed, and graced me with hope, courage, and undying faith in myself.

ACKNOWLEDGMENTS

I sincerely thank my advisor, Dr. Vivek Sharma, for his unconditional support, insightful guidance, and encouragement throughout my research journey. His expertise and dedication have been instrumental in shaping this thesis and advancing my understanding of the subject matter. His feedback on my research and coursework has been invaluable, pushing me to strive for excellence and reach my full potential. I am profoundly grateful for his patience, constructive feedback, and the countless hours he invested in helping me navigate the complexities of my research. Working on research with Dr. Sharma has been an enriching experience since it urges the utmost development of skills, sharpening of analytics acumen and high quality.

I would also like to sincerely thank my committee co-chair, Dr. Dhaval Gajjar, for his valuable insights and continuous support. His profound knowledge of the dataset employed in the study and his thoughtful suggestions have contributed significantly to the depth and rigor of this study.

My heartfelt thanks go to Dr. Jackson, Dr. Lucas, and Dr. Yoon for serving on my committee. Their diverse perspectives, expertise, and constructive critiques have significantly enriched my research. I am truly thankful to each of my committee members - their collective wisdom has been pivotal in completing this thesis.

TABLE OF CONTENTS

Table of Contents (Continued) Page

Table of Contents (Continued) Page

LIST OF TABLES

Table Page

List of Tables (Continued)

Table Page

LIST OF FIGURES

Figure Page

List of Figures (Continued)

Figure Page

List of Figures (Continued)

Figure Page

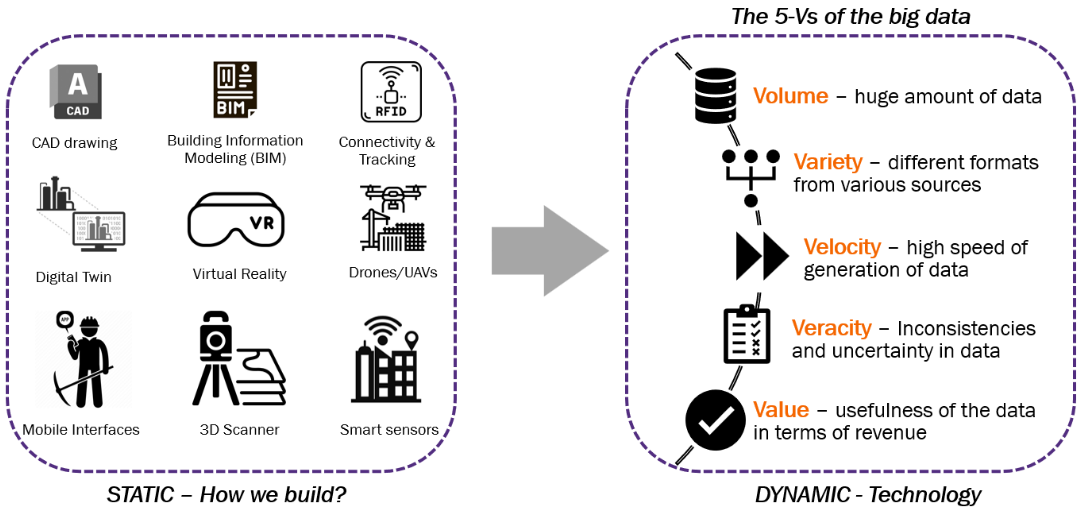
List of Figures (Continued)

Figure Page

CHAPTER ONE - INTRODUCTION

Data in the Construction Industry

The construction industry has always been a data-intensive sector with sources from multiple disciplines and stakeholders integrated throughout the project lifecycle (Bilal et al., 2016). The construction industry has witnessed a significant surge in the daily volume of big data gathered with the easy availability of smartphones, computers, sensors, and day-to-day usage gadgets (Munawar et al., 2022; Yu et al., 2020).


The 5-Vs of the big data
CAD drawing
Building Information Modeling (BIM)
Connectivity & Tracking
Digital Twin
Virtual Reality
Drones/UAVs
Mobile Interfaces
3D Scanner
Smart sensors
STATIC – How we build?
Volume – huge amount of data
Variety – different formats from various sources
Velocity – high speed of generation of data
Veracity – Inconsistencies and uncertainty in data
Value – usefulness of the data in terms of revenue
DYNAMIC - Technology

* (Munawar et al., 2022; Yousif et al., 2021; Cali et al., 2021)

Figure 1: Sources and attributes of data in the construction industry

As illustrated in Fig. 1, data generated in the sector satisfies the five most important attributes of big data, namely, (1) volume, the amount of data, (2) variety, heterogeneous data from different sources, (3) velocity, the speed at which data is generated, (4) veracity, inconsistencies and uncertainty in data, (5) value, usefulness of data in terms of returns and revenue (Cali et al., 2021; Munawar et al., 2022; Yousif et al., 2021). Heightened accessibility to technology with sensing-monitoring systems, wearable gadgets, and smartphones; favorable economic factors and regulations; and finally, the continuous demand for interoperable platforms and cloud computing has resulted in increasing the volume, velocity, variety, and veracity of data in the construction industry (Bilal et al., 2016; Blanco et al., 2023; Davila Delgado et al., 2020; Munawar et al., 2022; Ngo et al., 2020). The construction industry is experiencing digitization primarily owing to the ease of technology accessibility, increased investment in technology with favorable economic regulations, and demand for interoperability across the project stakeholders with rising numbers of one-stop-shop virtual platforms like openBIM (Blanco et al., 2023).

In comparison to other sectors of the data in the construction industry, the volume of data could be lesser by some orders of magnitude. However, the data's dynamic, heterogeneous, complex, and diverse nature is undeniable (Davila Delgado et al., 2020; Yousif et al., 2021). 40% or more of the firms in North America and Europe have analyzed big data and are benefitting from the analyses – 69% increase in strategic decisions, 54% better control of operations, 52% greater customer absorption, 47% lower costs and 8% increase in revenue (Yousif et al., 2021). Analysis and integration of data in the construction sector have numerous benefits, such as assisting in smart, fast and right decisions, design optimization, automation, decreasing risk and boosting productivity – crucial interest and support for the industry (Davila Delgado et al., 2020; Yousif et al., 2021).

The abundance of data available in the construction sector offers the opportunity to derive valuable insights and a data-driven approach to organizing and managing construction projects (Bilal et al., 2016; Munawar et al., 2022; Ngo et al., 2020). In many instances, construction companies do not make the most of the chance provided by the availability of this data. Instead, they tend to rely on traditional risk analysis methods, which heavily rely on subjective data sources and may overlook the interconnections among variables in the data (Gondia et al., 2019). The construction industry has traditionally been termed a laggard in technology adoption (Munawar et al., 2022). Multiple factors contribute to the sector's commonly acknowledged attribute of slow-paced adoption of emerging technologies compared to other industries, such as the uniqueness of each project's budget, timeline, specifications, and stakeholders (Radzi et al., 2019). Simultaneously, the construction technology landscape is gradually evolving towards integrated software platforms to meet customer needs more effectively (Bartlett et al., 2020).

Translating big data into effective insights requires embracing computational interventions, data analytical techniques, statistical analysis, algorithm development, and machine learning (Bilal et al., 2019; Munawar et al., 2022). The advent of big data in the construction sector has piqued interest in comprehending and utilizing the extensive database through sophisticated statistical, computational, and visualization techniques (Ahmed et al., 2022; Bilal et al., 2019). Machine learning has been experiencing increased adoption within the construction sector. This technology is bringing transformations across various aspects of construction project management, such as risk assessment and mitigation, safety management on construction sites, cost estimation and forecasting, schedule management, and the prediction of building energy demand (Nguyen Van & Nguyen Quoc, 2021). The construction sector lacks comprehensive predictive analytics solutions for projects or industry-wide applications - a significant portion of descriptive analytics tools rely on trade association surveys or dashboards constructed from collected company data, suffering from infrequent updates or limited detail (Bartlett et al., 2020). There has been an increase in global investment in architecture-engineering-construction (AEC) between the years 2020 and 2022 by 85% (Blanco et al., 2023), with a forecasted increase in the global market for construction analytics from the years 2020 to 2025 from 6.5 billion dollars to 16.8 billion dollars (Enterprise Solutions, 2023). As per a study conducted by Ngo et al., 2020, 43% of construction firms interviewed claimed that their data volume was 10 – 100 terabytes (TB), where in perspective, 1 TB is equivalent to 1,000 gigabytes (GB).

Research Motivation

The idea of developing a prediction model originated from one of the deliverables of broader research developed by the Construction Industry Institute (CII), Clemson University, and six (6) federal agencies - the Federal Facilities Data Analytics Research Application Program (FF-DARAP). The participating federal agencies include the Department of Energy (DoE), Department of Commerce, Department of State, Naval Facilities Engineering Systems Command (NAVFAC), Ontario Power Generation (OPG), and the Smithsonian Institution. To achieve the goal of finding commonalities among participating federal agencies, comparing cost and schedule data and making informed decisions, FF-DARAP is focused on developing "a better way to track data and extract business intelligence from federal facilities." (Clemson News, 2022)

As illustrated in Fig. 2, the FF-DARAP study was a three-year project divided into three (3) phases with the intent to automate the construction industry data management and analytical systems through the deployment of a computing system and ML algorithms to develop an advanced data warehouse and analytics platform for federal agencies.

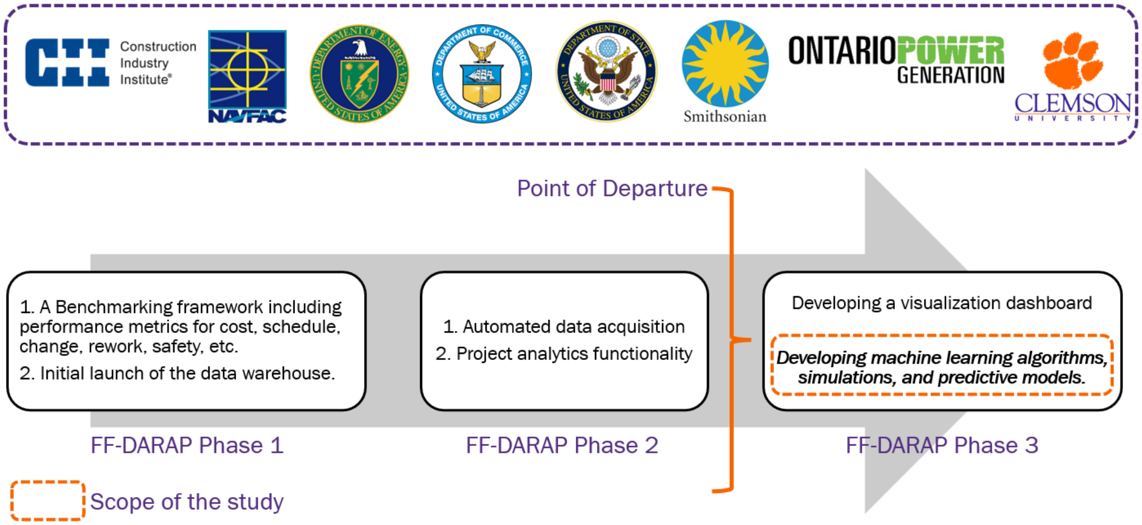

CII Construction Industry Institute
NAVFAC
Smithsonian
ONTARIO POWER GENERATION
CLEMSON UNIVERSITY
Point of Departure
1. A Benchmarking framework including performance metrics for cost, schedule, change, rework, safety, etc.
2. Initial launch of the data warehouse.
1. Automated data acquisition
2. Project analytics functionality
Developing a visualization dashboard
Developing machine learning algorithms, simulations, and predictive models.
FF-DARAP Phase 1
FF-DARAP Phase 2
FF-DARAP Phase 3
Scope of the study

Figure 2: Point of departure and scope of the research

The first phase focused on creating a framework for benchmarking, including performance metrics for cost, schedule, change, rework, safety, best practices implementation, etc. Phase 2 of the study focused on automating data acquisition and integration in the data warehouse with project analytics functionality. The final phase of the FF-DARAP study included the development of ML algorithms, simulations and prediction models, and visualization dashboards for federal agencies to deliver data-driven insights.

Challenge or Opportunity

A prediction model for project success for the FF-DARAP study – posed multiple challenges for the successful deliverable. As illustrated in Fig. 3, it can be observed that with multiple types and sources of input variables in a prediction model, there can be myriad options to choose from in statistical and/or machine learning (ML) algorithms.

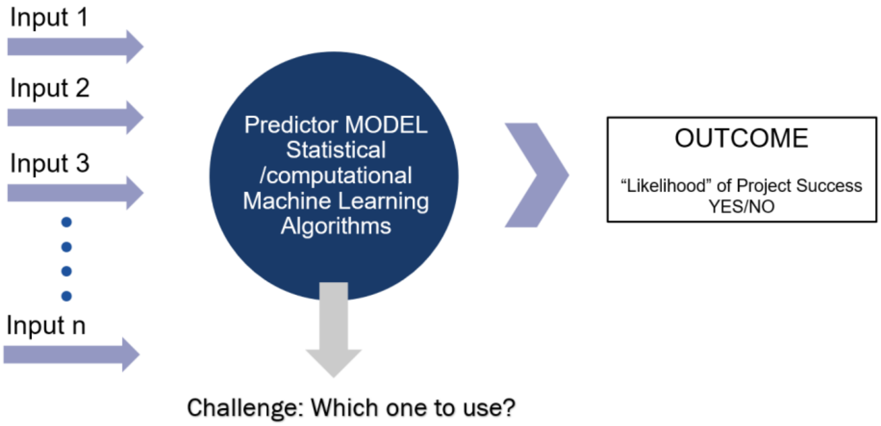

Input 1
Input 2
Input 3
Input n
Predictor MODEL Statistical /computational Machine Learning Algorithms
OUTCOME
"Likelihood" of Project Success YES/NO
Challenge: Which one to use?

Figure 3: Prediction Model – Challenge

This resulted in two essential queries – identifying challenges encountered while engaging in predictive analytics of construction-specific data and delineating factors that need consideration for selecting the most suited prediction model for a given dataset. For example, construction safety data distribution is unique to its metrics, rendering it non-parametric. However, most studies on safety have used traditional statistical methods. ML algorithms recently used for such studies provide conflicting contexts on the use of various computational paradigms.

To define the challenge and discuss the opportunity, Fig. 4 illustrates the multiple aspects of a prediction model architecture, including input data, statistical model and ML-based model.

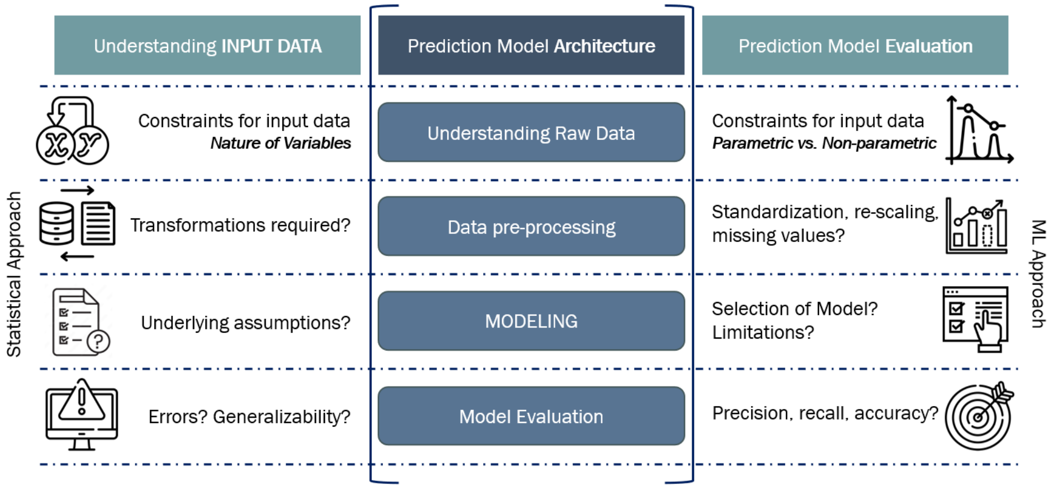

Understanding INPUT DATA
Prediction Model Architecture
Prediction Model Evaluation
Constraints for input data
Nature of Variables
Understanding Raw Data
Constraints for input data
Parametric vs. Non-parametric
Transformations required?
Data pre-processing
Standardization, re-scaling, missing values?
Underlying assumptions?
MODELING
Selection of Model? Limitations?
Errors? Generalizability?
Model Evaluation
Precision, recall, accuracy?
Statistical Approach
ML Approach

Figure 4: The IDEA - An expanded look

As shown in Fig. 4, multiple parameters must be considered while designing the architecture of a prediction model for both statistical and ML-based approaches. For the development of statistical prediction models in the construction industry, studies have evaluated the constraints for input data (nature of variables and the relationship between the variables), transformations performed to comply with essential assumptions, and finally, evaluate the performance of the model, documenting the errors (Kim et al., 2008; Ling et al., 2004; Love & Teo, 2017). Various transformations pose multiple challenges of interpretation and generalizability. While developing ML-based prediction models, existing literature has begun to discuss the nature of input data (dimensionality, non-linear relationship between variables), data pre-processing requirements, selection of suitable model with its advantages and limitations, and finally, the evaluation of the model's prediction accuracy and performance metrics (Chandanshive & Kambekar, 2019; Davila Delgado et al., 2020; Gondia et al., 2019; Poh et al., 2018; Tixier et al., 2016). However, there is a lack of comprehension of the applicability of such approaches to the data from the construction domain. For example, Poh et al., 2018 investigated safety concerning ML algorithms. However, the study did not account for the shape or the skewness of the safety data distribution. Also, Sanni-Anibire et al., 2022 assess the prediction of the risk for delay with multiple ML methods – relying on developing a dataset from subjective data without considering the nature of the data distribution, making it unsuitable for generalization.

Thus, a guide with a decision-making framework will be useful in the landscape of multiple models. The current study quantifies the inherent bias and impact of the nature and distribution of construction-specific data on the generalization of results. Future research, when conducted with a refined understanding of the nature of the dataset and distribution along with the goal of prediction before implementing statistical/ML methods, will genuinely benefit in increasing the accuracy and generalization of predictions in the construction industry.

Research Gap

The methods used to construct the prediction model can be classified into two main parts: (1) statistical-based methods and (2) machine learning (ML)-based methods. Although the statistical-based method can accomplish superior performance for forecasting, strict statistical assumptions, such as multivariate normality and independent predictor variables, must be complied with to impede practical utilization. Considering the dataset's distinct, heterogeneous, and dynamic nature in the construction industry, ML-based methods are increasingly being adopted for prediction modeling. It can be observed that multiple studies have used varying permutations and combinations of algorithms and methods or techniques for pre-processing the data, training, testing and validation. Multiple methods have been adopted to improve the model's performance, treat the imbalance in datasets, and mitigate the overfitting of the model. However, there is no study to assist in selecting the most suited, appropriate combination for a given dataset. The dataset's source defines the features and how the data must be pre-processed, as well as different strategies required to tackle the nature of data and ensure that the algorithm’s output meets the prediction objectives. Researching the relationship between datasets and algorithms could provide valuable guidance based on dataset characteristics, thereby reducing time spent on trial and error (Li et al., 2018). Thus, norms need to be developed that can assist in selecting the method or approach most suited for developing a prediction model for the given dataset.

Aim of the Research

The study proposes developing a framework of norms that assist in identifying the most suitable prediction model for a non-parametric dataset specific to customer satisfaction in the construction coatings industry. These norms could be utilized to develop the decision-making matrix in which input parameters are definitive for prediction model architecture. Its output would be the most suitable model for the given data set. The data used for the current study is sourced from Customer Satisfaction Data for Coating projects.

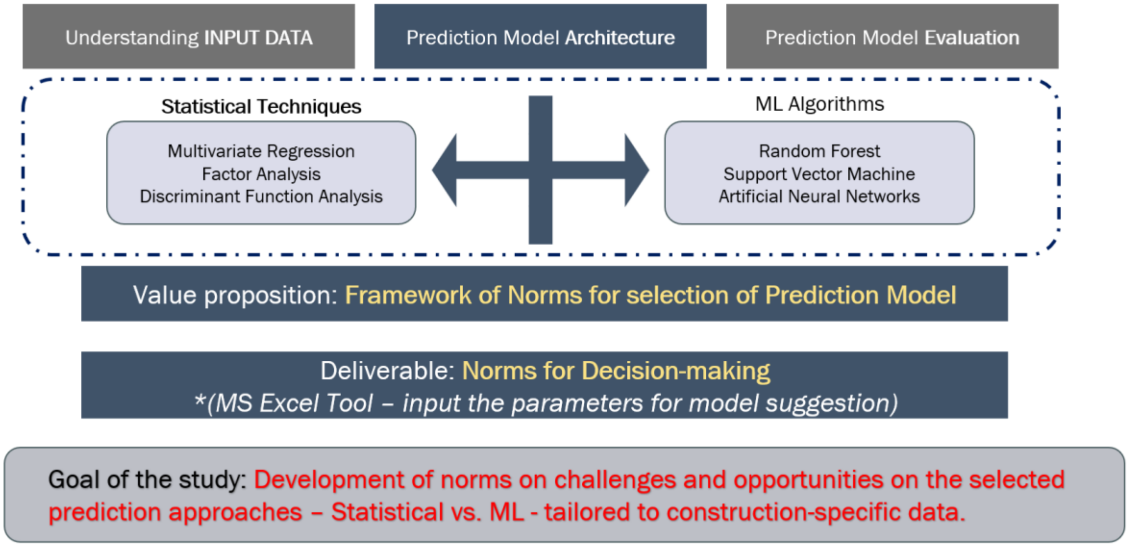

Understanding INPUT DATA
Prediction Model Architecture
Prediction Model Evaluation
Statistical Techniques
Multivariate Regression
Factor Analysis
Discriminant Function Analysis
ML Algorithms
Random Forest
Support Vector Machine
Artificial Neural Networks
Value proposition: Framework of Norms for selection of Prediction Model
Deliverable: Norms for Decision-making
*(MS Excel Tool – input the parameters for model suggestion)
Goal of the study: Development of norms on challenges and opportunities on the selected prediction approaches – Statistical vs. ML - tailored to construction-specific data.

Figure 5: Goal of the study

As illustrated in Fig. 5, the study aims to develop a decision-making framework with norms on various challenges and opportunities for prediction model architecture tailored to construction-specific data between traditional or statistical and new-world or machine learning (ML) based computational algorithms.

CHAPTER TWO - RESEARCH OBJECTIVES AND QUESTIONS

The research objectives of the study are to assess the efficacy of statistical models (ST) vis-à-vis machine learning (ML) based approaches for prediction modeling in the construction industry by the following:

Research Objective 1 - MEASURE the outcomes of the "customer satisfaction" (for the construction coatings sector) prediction model for both ST and ML approaches.

Research Question 1

What are the outcomes of ST and ML-based models for predicting customer satisfaction with a dataset having non-parametric distribution and limited dimensionality specific to the construction coatings sector?

Driven by literature review and input data, the first step is to measure the accuracy of each prediction method individually for prediction outcomes within construction-specific domains, such as customer satisfaction scores of construction coating projects. This objective entails the evaluation of the performance of the prediction model through statistical and ML-based algorithmic approaches. The performance of models is inferred using metrics from the R^2 value for statistical models and Mean Square Error, Mean Absolute Error, and Confusion matrix (CM) for ML-based algorithmic models.

Research Objective 2 - COMPARE ST vis-à-vis ML-based models predicting customer satisfaction in the construction coatings sector for non-parametric dataset with limited dimensions.

Research Question 2

What are the limitations and advantages of ST vis-à-vis ML approaches while predicting customer satisfaction in the construction coatings sector for a non-parametric dataset with limited dimensions?

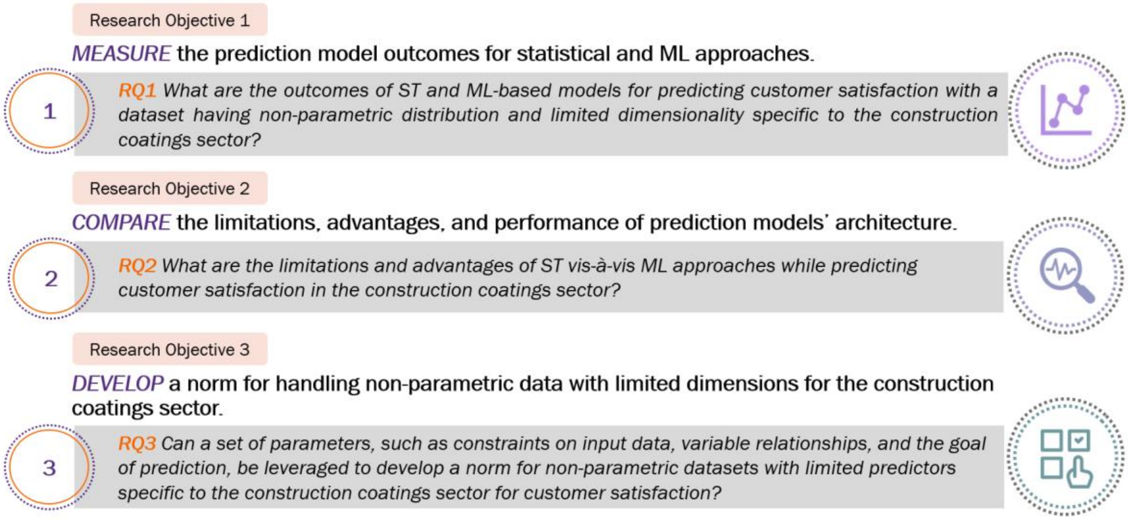
This entails comparing the prediction model outcomes of traditional statistical models and ML-based algorithms by inferring each prediction method's limitations, advantages, techniques, and data pre-processing strategies. The comparison between the two approaches would require documenting the parameters defining each approach under the unique common conditions of a dataset with non-parametric distribution and limited IVs.

Research Objective 3 - DEVELOP a norm for handling non-parametric data with limited dimensions for the construction coatings sector.

Research Question 3

Can a set of parameters, such as constraints on input data, variable relationships, and the goal of prediction, be leveraged to develop a norm for non-parametric datasets with limited predictors specific to customer satisfaction for the coating sector?

The study aims to develop norms on various challenges and opportunities in the selected prediction approaches. The documentation of the norms will depend on construction-specific data used in the study. This will pave the way for a methodology that assists decision-making when selecting a method for prediction models. Initial literature review suggests various factors play a crucial role in the performance of the – ST or ML-based models, such as the nature of the dataset, nature of independent variables, nature of dependent variables, nature of interdependence of variables, nature of factors contributing to the outcome, the goal of analysis and prediction. Fig. 6 illustrates the mapping of the objectives with the study's primary research questions.



Research Objective 1

MEASURE the prediction model outcomes for statistical and ML approaches.

1

RQ1 What are the outcomes of ST and ML-based models for predicting customer satisfaction with a dataset having non-parametric distribution and limited dimensionality specific to the construction coatings sector?

Research Objective 2

COMPARE the limitations, advantages, and performance of prediction models' architecture.

2

RQ2 What are the limitations and advantages of ST vis-à-vis ML approaches while predicting customer satisfaction in the construction coatings sector?

Research Objective 3

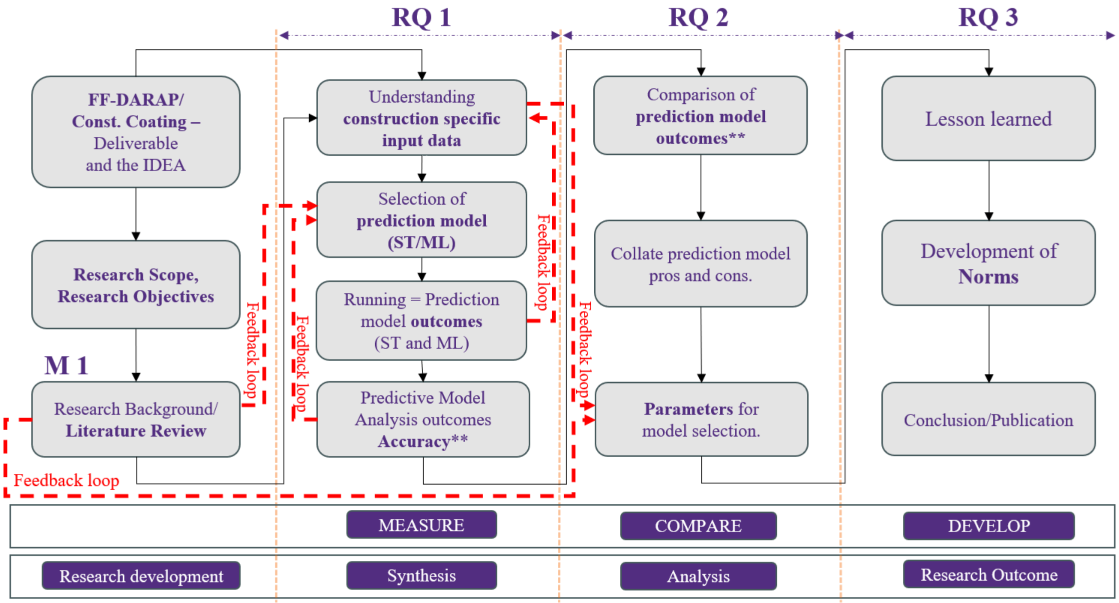
DEVELOP a norm for handling non-parametric data with limited dimensions for the construction coatings sector.

3

RQ3 Can a set of parameters, such as constraints on input data, variable relationships, and the goal of prediction, be leveraged to develop a norm for non-parametric datasets with limited predictors specific to the construction coatings sector for customer satisfaction?

Figure 6: Research Objectives and Questions

The methodology for the study has been developed by mapping the study's objectives and research questions. Fig. 7 below illustrates the overall research framework divided into four phases, namely, (1) Research development, (2) Synthesis, (3) Analysis, and (4) Research Outcome. All the phases, except for the first, are mapped to the three research questions and labeled as “MEASURE” for Research Question 1, “COMPARE” for Research Question 2, and “DEVELOP” for Research Question 3.


RQ 1
RQ 2
RQ 3
FF-DARAP/ Const. Coating – Deliverable and the IDEA
Research Scope, Research Objectives
M 1
Research Background/ Literature Review
Feedback loop
Understanding construction specific input data
Selection of prediction model (ST/ML)
Running = Prediction model outcomes (ST and ML)
Predictive Model Analysis outcomes Accuracy**
Feedback loop
Feedback loop
Feedback loop
Feedback loop
Comparison of prediction model outcomes**
Collate prediction model pros and cons.
Parameters for model selection.
Lesson learned
Development of Norms
Conclusion/Publication
MEASURE
COMPARE
DEVELOP
Research development
Synthesis
Analysis
Research Outcome

ST – Statistical approach
ML – Machine-Learning approach

** model accuracy TBD

Figure 7: Overall Research Framework

M1 – Research Development

After formulating the idea and study objectives, the first essential step of the study was a literature review. A literature review needs to assess the studies already published for architecture and applications of prediction models in the construction industry. The literature review will document features of statistical and ML-based approaches toward developing prediction tools or models. Advantages, limitations, and findings unique to the prediction models and performance metrics adopted for each model will also be documented. Additionally, there is a need to understand and interpret constructions-specific input data – its distribution, size, source, nature of variables, the relationship amongst the independent variables or the predictors, etc.

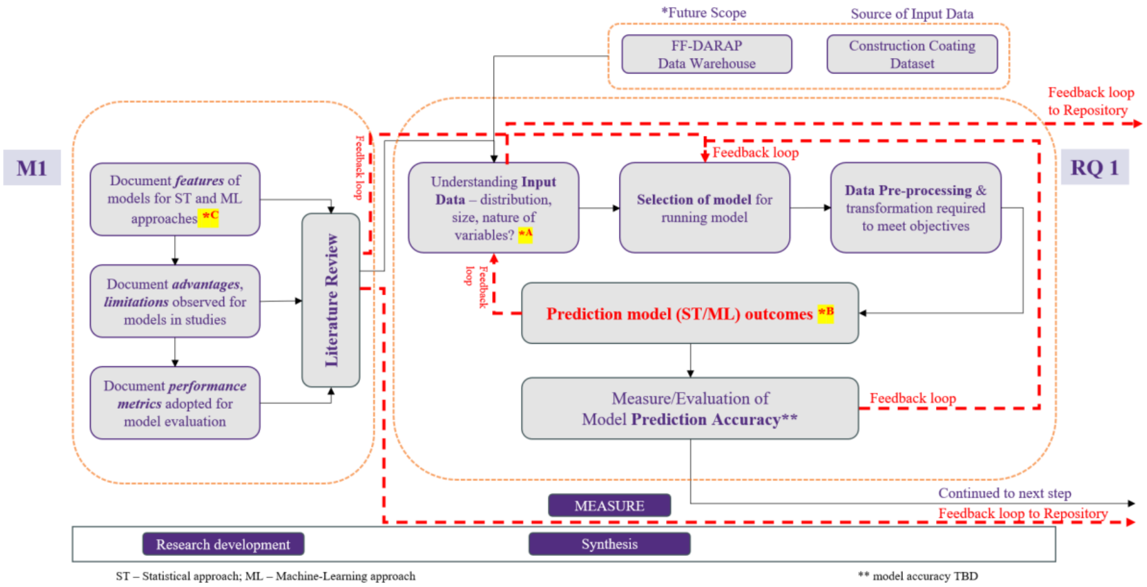

*Future Scope
Source of Input Data
FF-DARAP Data Warehouse
Construction Coating Dataset
Feedback loop to Repository
M1
RQ 1
Document *features* of models for ST and ML approaches *C
Document *advantages*, *limitations* observed for models in studies
Document *performance metrics* adopted for model evaluation
Literature Review
Feedback loop
Understanding **Input Data** – distribution, size, nature of variables? *A
Feedback loop
Selection of model for running model
Data Pre-processing & transformation required to meet objectives
Feedback loop
Prediction model (ST/ML) outcomes *B
Measure/Evaluation of Model **Prediction Accuracy****
Feedback loop
MEASURE
Continued to next step
Feedback loop to Repository
Research development
Synthesis
ST – Statistical approach; ML – Machine-Learning approach
** model accuracy TBD

Figure 8: Methodology - Research Development and Synthesis

RQ1 – MEASURE/EVALUATE – Synthesis

The "Synthesis' phase in the methodology, as illustrated in Fig. 8, answers the first research question, evaluating the outcomes generated using prediction models adopting both the statistical and ML-based approach for a given dataset. The input data source is from the construction coatings sector, documenting customer satisfaction for projects across the country. Based on the literature review, the next step is to select the model/approach/technique/algorithm for running the prediction model. Data pre-processing and transformations, if required to comply with prerequisites for the computational techniques or to meet the analysis objective, are conducted right after. This is followed by the most critical step in this stage, i.e., the implementation of the model – statistical or ML. This step produces the outcomes for a given dataset, which is evaluated for prediction accuracy. The outcomes of the prediction model are a source of feedback for the interpretation of input data since it will allow defining parameters of input data that potentially impact the outcome. Further, the performance evaluation of the outcomes generates a feedback loop for the model selection since it serves as a benchmark for acceptance of the model architecture or tuning and optimizing the model design to improve performance.

RQ2 – COMPARE – Analysis

Fig. 9 illustrates the succeeding stages in the methodology in which the "Analysis" stage works towards comparing the outcomes generated by research question 1 - running different models in the previous step. Further, the advantages and limitations of the selected model/approach are compared, and the performance metrics appropriate for the selected approach are documented. All the observations recorded by comparing the outcomes assist in identifying parameters that define the architecture of the prediction model and are stored in a "repository." This repository also receives the feedback loop from the literature review, in which similar findings were documented from existing scholarly literature.

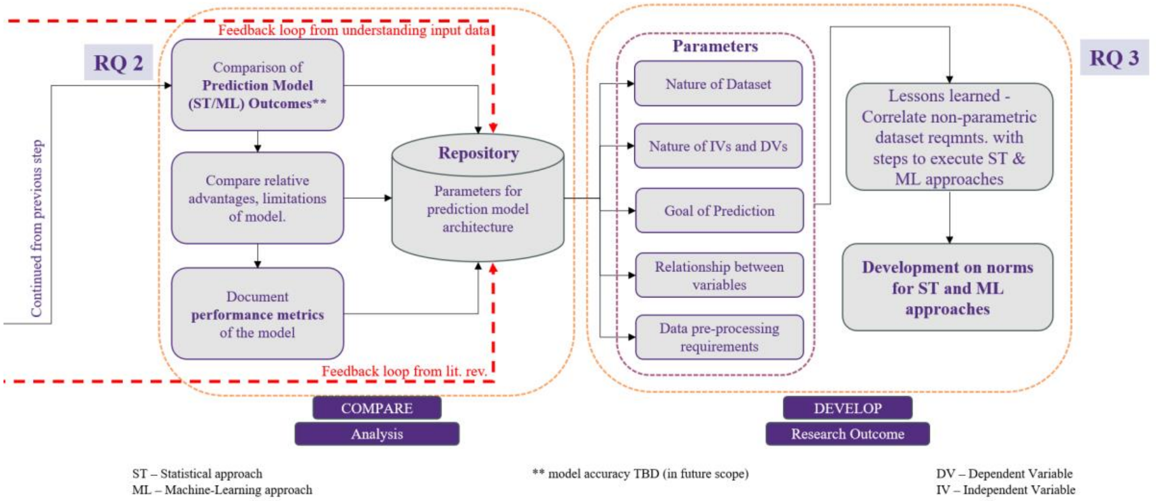

Feedback loop from understanding input data
RQ 2
Continued from previous step
Comparison of Prediction Model (ST/ML) Outcomes**
Compare relative advantages, limitations of model.
Document performance metrics of the model
Repository
Parameters for prediction model architecture
Feedback loop from lit. rev.
COMPARE
Analysis
Parameters
Nature of Dataset
Nature of IVs and DVs
Goal of Prediction
Relationship between variables
Data pre-processing requirements
Lessons learned - Correlate non-parametric dataset reqmnts. with steps to execute ST & ML approaches
Development on norms for ST and ML approaches
RQ 3
DEVELOP
Research Outcome
ST – Statistical approach
ML – Machine-Learning approach
** model accuracy TBD (in future scope)
DV – Dependent Variable
IV – Independent Variable

Figure 9: Methodology - Analysis and Research Outcome

RQ3 – DEVELOP – Research Outcome

The final stage of the research methodology is the "Research Outcome" phase (Fig. 9), which is mapped to the third research question of developing the norms for ST and ML approaches for non-parametric datasets with limited predictors specific to customer satisfaction for the coating sector. Based on the repository developed in the previous stage, the lessons learned about the parameters governing the architecture of the prediction model are tabulated and categorized, correlating the requirements of the non-parametric dataset with steps to execute the ST and ML approaches. This exercise constitutes the framework with norms on various challenges and opportunities for the selected prediction approaches. In the future, the framework could result in formulating a tool that can assist in selecting the appropriate computational technique – statistical and ML- along with other parameters for datasets in the construction coatings sector.

CHAPTER FOUR - LITERATURE REVIEW- M1

As a first step of the research methodology, a literature review was conducted to understand the big picture and study the existing peer-reviewed publications for prediction models in the construction industry.

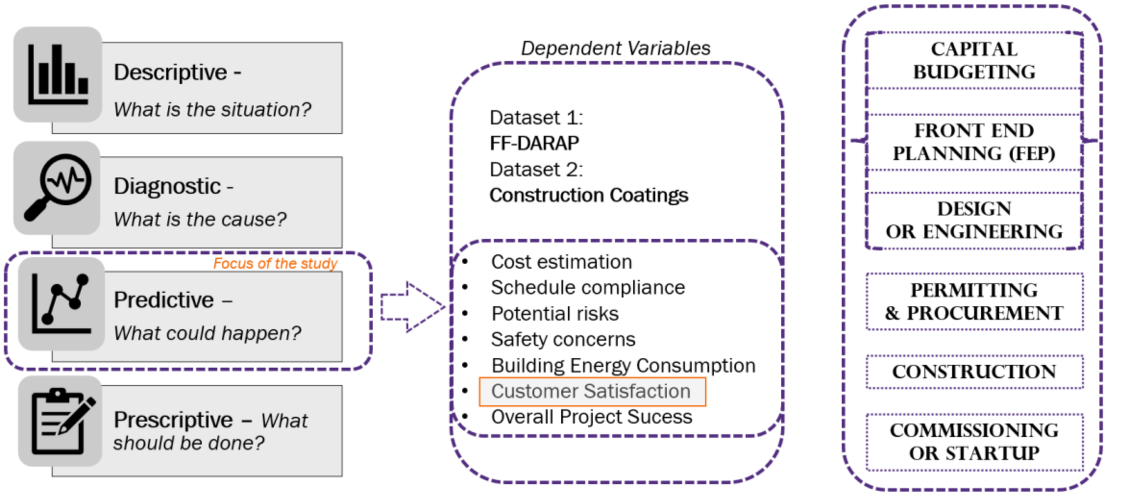
Predictions in the Construction Industry

Prediction models in the construction sector demand greater scrutiny due to their unique attributes and the substantial capital required to commence and sustain projects, making it a high-risk category of projects necessary for reliable predictions (Tayefeh Hashemi et al., 2020). Project characteristics such as low margin, low productivity, and the fragmented and dynamic nature of the construction industry translate into concerns that need addressing, which is challenging (Tariq & Gardezi, 2023). Multiple criteria for success in a construction project have been identified in existing literature, considering the perspectives of all stakeholders, that include but are not limited to compliance with the baseline schedule and stipulated budget, the satisfaction of specified quality parameters, safety, client satisfaction, employees' satisfaction, profitability, cash-flow management, impact on end-user and future-readiness (Caldas & Gupta, 2017; Gunduz & Tehemar, 2020; Shenhar et al., 2001; Silva et al., 2016; Tariq & Gardezi, 2023). Interestingly, researchers have acknowledged the potential influence of early planning processes on project outcomes and laid more emphasis on the criticality of the project pre-planning processes for construction projects (Gibson et al., 2006; Kolltveit & Grønhaug, 2004;

Wang et al., 2012a). Thus, predictions of the project's cost and schedule overruns, safety management, risk identification and assessment, and overall project performance in the early phases of the project's lifecycle are critical for construction projects' successful execution (Assaad et al., 2020; Jaber et al., 2020; Nguyen Van & Nguyen Quoc, 2021a; Tariq & Gardezi, 2023; Tayefeh Hashemi et al., 2020; Tixier et al., 2016; Wang et al., 2012).

Data and Predictive Analytics

Data can be analyzed using four different types of methods, namely, (1) descriptive, (2) diagnostic, (3) predictive, and (4) prescriptive. Descriptive analytics categorize and classify data, identifying and understanding existing patterns and trends, essentially presenting the current scenario. Diagnostic analytics try to find the root cause of the situation: predictive analytics forecast and estimate by interpreting historical data, identifying dominant patterns, and extrapolating the relationships. Prescriptive analytics uses algorithms, optimization models, and insights from descriptive and predictive analytics to evaluate potential decisions with complex and high-volume objectives (Davila Delgado et al., 2020; Ngo et al., 2020). As illustrated in Fig. 10 below, the focus of the study is predictive analytic methods adopted for factors of construction project management across the project lifecycle. The use of predictive analytical techniques has witnessed a significant increase in the construction sector with abundant data, increased digitization of the industry and demand for insightful forecasts by stakeholders (Ngo et al., 2020).



Predictive analytics conducted through various statistical and ML/AI approaches/models are applicable throughout the lifecycle of a project.

Figure 10: Categories of data analytics with factors and project's lifecycle.

Predictive data analytics use statistical, data mining or machine learning approaches (Munawar et al., 2022). The current scope of the study is limited to the statistical and machine-learning approaches discussed in the following sections.

Prediction Modeling in the Construction Industry

Among the many analytical solutions, the prediction models are the holy grail of various sectors for informed decision-making through data. Prediction models deploy statistical and ML techniques to analyze historical data, identify hidden patterns, and extrapolate the relationships to inform decisions that can impact the future of project performance (Bilal et al., 2019; Ngo et al., 2020). The prediction model is illustrated in Fig. 11 below, wherein it can be observed that data collected from real-world construction projects is inferred with statistical methods, which then becomes the foundation for developing the model. Further, the model delivers predictions analyzing phenomena with uncertain outcomes via ML algorithms.

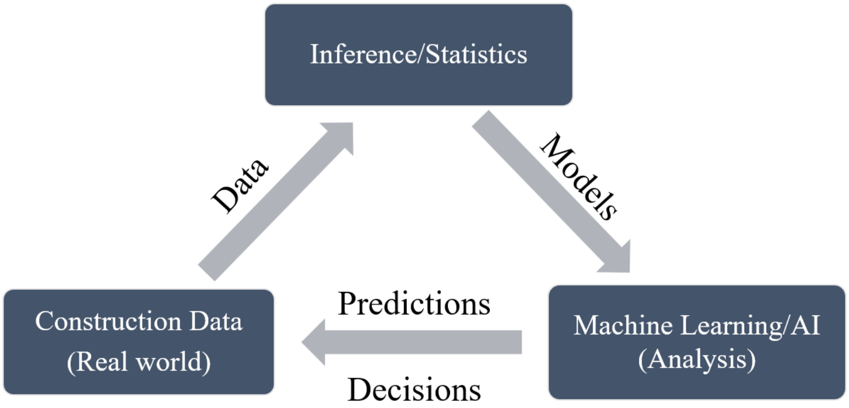
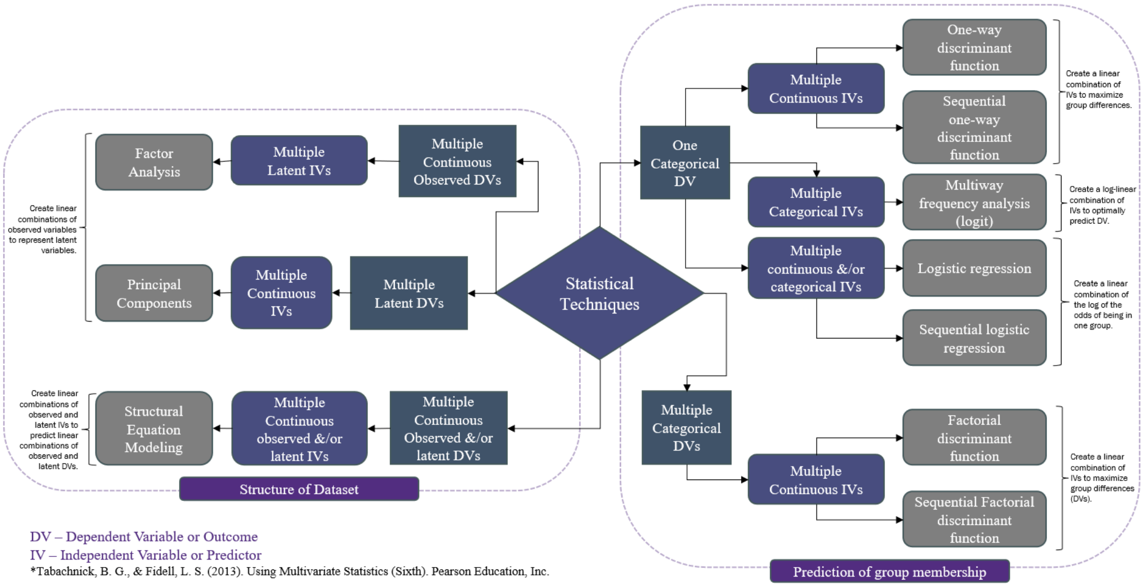


Figure 11: Framework for Prediction Model

Various studies have employed numerous computational methods for prediction modeling from the traditional statistical realm. Further, multiple studies have attempted to predict critical factors as the dependent variable with newer ML models.

Statistical Prediction Techniques in the Construction Industry

Statistics is the study of collecting, analyzing, and interpreting conclusions from data, focusing on selecting the right tools and techniques at every analysis stage (Bilal et al., 2016). Multiple studies have implemented statistical techniques to develop prediction models and have applications in the construction industry (Bilal et al., 2016; Munawar et al., 2022; Poh et al., 2018). Statistical methods formulating prediction models allow the translation of a significant amount of data through analysis to identify patterns and trends. The selection of the proper tool/technique in statistical analysis is the key to critical inferences, conclusions, and forecasts (Bilal et al., 2016; Kim et al., 2008). Fig. 12 below highlights the primary statistical techniques that can be adopted depending on the nature of the independent variables, dependent variables, and the goal of the analysis.


Statistical Techniques
Multiple Continuous Observed DVs
Multiple Latent IVs
Factor Analysis
Multiple Latent DVs
Multiple Continuous IVs
Principal Components
Create linear combinations of observed variables to represent latent variables.
Multiple Continuous Observed &/or latent DVs
Multiple Continuous observed &/or latent IVs
Structural Equation Modeling
Create linear combinations of observed and latent IVs to predict linear combinations of observed and latent DVs.
Structure of Dataset
One Categorical DV
Multiple Continuous IVs
One-way discriminant function
Sequential one-way discriminant function
Create a linear combination of IVs to maximize group differences.
Multiple Categorical IVs
Multiway frequency analysis (logit)
Create a log-linear combination of IVs to optimally predict DV.
Multiple continuous &/or categorical IVs
Logistic regression
Sequential logistic regression
Create a linear combination of the log of the odds of being in one group.
Multiple Categorical DVs
Multiple Continuous IVs
Factorial discriminant function
Sequential Factorial discriminant function
Create a linear combination of IVs to maximize group differences (DVs).
Prediction of group membership
DV – Dependent Variable or Outcome
IV – Independent Variable or Predictor

*Tabachnick, B. G., & Fidell, L. S. (2013). Using Multivariate Statistics (Sixth). Pearson Education, Inc.

Figure 12: Statistical Prediction Techniques

Studies have evaluated parameters such as delays in construction projects (Kim et al., 2008), classification of best practices for project cost and schedule performance (Lee, 2001), identifying actions critical for site safety (J. Gong et al., 2011), cost, time and quality performance of design-bid-build and design-build projects (Ling et al., 2004), construction costs (Hwang, 2009; Lowe et al., 2006) and detection of structural damage (X. Jiang & Mahadevan, 2008) using statistical models. The techniques employed by studies for developing statistical prediction models include but are not limited to factor analysis, Bayesian networks and correlation matrix (Kim et al., 2008), discriminant function analysis (Lee, 2001), bag-of-words and Bayesian network (J. Gong et al., 2011), multivariate regression analysis (Ling et al., 2004; Lowe et al., 2006), dynamic regression (Hwang, 2009) and Bayesian probabilistic assessments (X. Jiang & Mahadevan, 2008). Table 1 below summarizes the studies that have adopted statistical techniques for predicting critical factors for construction projects. The table also delineates the study's goal, focus factors, and references.

Table 1: Summary of Statistical Predictive Analytic Methods

Statistical technique/s employed	Objective of the analysis	Focus factor	Study Reference
Factor Analysis	Identify the main factors of construction delays	Schedule	Kim et al., 2008
Bayesian network			
Correlation matrix			
Discriminant Function Analysis	Cost and schedule performance	Cost	Lee, 2001
		Schedule	
Bag-of-words	Categorize actions of construction workers and equipment	Safety	J. Gong et al., 2011
Bayesian network			
Multivariate regression Analysis	Cost, time, quality, satisfaction – overall project performance	Cost	Ling et al., 2004
		Schedule	
		Quality	
		Owner Satisfaction	
	Cost per square meter	Cost	Lowe et al., 2006
Linear regression	Construction Cost Index	Cost	Hwang, 2009
Categorical regression			
Dynamic regression			
Bayesian network	Structural responses for damage detection	Safety	X. Jiang & Mahadevan, 2008
Link Analysis	Key knowledge areas from post-project reviews *(text-mining)*	Owner Satisfaction	Carrillo et al., 2011
Dimensional matrix analysis			
Poisson Model	Accident occurrences	Safety	Chua & Goh, 2005
Principal Factor Analysis	Total recordable incident rate (TRIR); Severity rate (SR)		Salas & Hallowell, 2016
Multiple Linear Regression			
Negative binomial regression model	Construction injuries	Safety	Love & Teo, 2017

However, the traditional techniques, which are more suited for structured data in smaller sample sizes, are not ideal for processing massive volumes of big data generated on construction projects with increasing integration of technology (Bilal et al., 2019; Yu et al., 2020).

ML Prediction Techniques in the Construction Industry

Data in the construction industry entails a massive, extensive amount of information collected from diverse sources in different formats and at a rapid pace. This cannot be typically analyzed efficiently using traditional data analysis tools or technologies that rely primarily on a relational database and centralized computing methods, generally suited to structured data with limited sample sizes (Yu et al., 2020). Statistical models are deficient in predictive performance when datasets feature many variables, and their effectiveness is constrained by the statistical assumptions on which they are based (Poh et al., 2018). In contrast to statistical modeling, machine learning (ML) holds a significant advantage due to its adaptability to function without the constraints of statistical assumptions and yield higher accuracy in predictions using optimization techniques that reduce the instances of incorrect predictions (Aggarwal, 2015; Gondia et al., 2019; Kim et al., 2008). Moreover, the potential of ML techniques to handle uncertainty and missing data while analyzing voluminous, complicated datasets with interdependent variables of differing structures is undeniable (Gondia et al., 2019; Hashemi et al., 2020). Furthermore, ML can capture linear and non-linear relationships within the phenomenon under investigation (Poh et al., 2018). Thus, with the surge in the amount of big data generated in the construction sector and the growing popularity of ML-based models in the industry – increased adoption of ML-based predictive analytical techniques has been observed in the sector (Gondia et al., 2019; Nguyen Van & Nguyen Quoc, 2021).

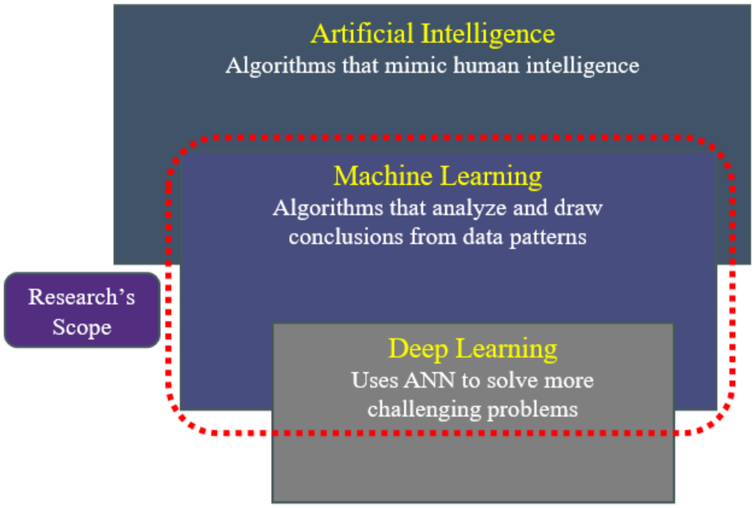

Artificial Intelligence
Algorithms that mimic human intelligence
Machine Learning
Algorithms that analyze and draw conclusions from data patterns
Research's Scope
Deep Learning
Uses ANN to solve more challenging problems

Figure 13: Overview of AI, ML and DL

Artificial Intelligence (AI) refers to the set of algorithms that imitate human intelligence to perform actions such as learning, predicting, classifying, and reasoning – it is a broad term comprising the theory and application of computer systems performing such tasks (Cali et al., 2021). ML is the subset of AI in which algorithms and statistical models are trained to analyze and draw conclusions from data patterns and is the scope of the current study as illustrated in Fig. 13 below (Cali et al., 2021; Duarte & Ståhl, 2018; Shaveta, 2023). Deep learning (DL) is a further subcluster of ML that employs a specific category of ML algorithms called artificial neural networks (ANNs) to solve more challenging problems and learn from data features giving output in the form of time series forecasting, image and audio-video (AV) classification or recognition (Cali et al., 2021).

Before discussing studies that have employed machine learning (ML) algorithms for predictive analytics, it is essential to briefly understand the categorization of ML models and algorithms as illustrated in Fig. 14. ML models can be broadly divided into four (4) types: supervised, unsupervised, reinforcement and ensemble learning (Bilal et al., 2016; Cali et al., 2021).

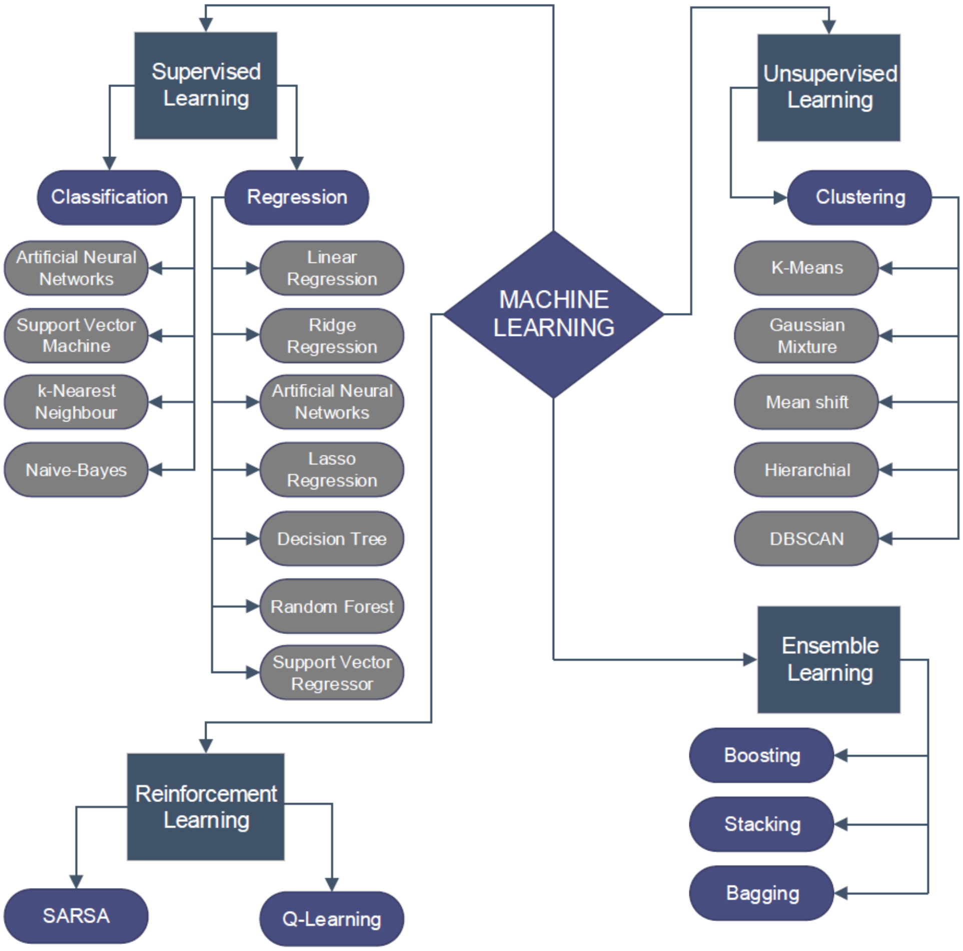


Figure 14: Categories of ML Models and Algorithms

Cost estimation and prediction (Chandanshive & Kambekar, 2019; Q. Jiang, 2020; Mahalakshmi & Rajasekaran, 2019; Tijanić et al., 2020), schedule compliance (Gondia et al., 2019; Wang et al., 2012a), potential risks (Ajayi et al., 2020; Kifokeris & Xenidis, 2019; Sanni-Anibire et al., 2022), safety concerns (P. Gong et al., 2020; S. et al., 2021; Poh et al., 2018; Tixier et al., 2016) and building energy consumption (Mocanu et al., 2016; Rahman et al., 2018; Rahman & Smith, 2018) have been forecasted with the assistance of ML-based algorithmic prediction models. Table 2 below summarizes the literature review conducted to study ML algorithms deployed to conduct the analysis, along with the goal of the prediction model, factors, and the publication reference. The table further delineates the studies' focus areas depending on the ML algorithm category integrated into the prediction model's design and architecture.

Table 2: Summary of Machine Learning Predictive Analytic Methods

ML algorithms employed	Objective of the predictive analysis	Factor	Study Reference
Focus Area 1 - Artificial Neural Networks (ANN)			
Multilayer perceptron, general regression neural network, radial basis function neural network	Costs at different stages of the project lifecycle	Cost	Tijanić et al., 2020
Multi-perceptron network with backpropagation algorithm	The construction cost of the highway at an early stage		Mahalakshmi & Rajasekaran, 2019
Multilayer feed-forward neural network model trained along with a backpropagation algorithm	Building/construction cost		Chandanshive & Kambekar, 2019
Backpropagation neural network (BPNN)			Q. Jiang, 2020
Radial basis function neural network			
ANN			Hashemi et al., 2019
NN	Construction project costs		Gu, 2023
NN	Estimation of S Curve		Chao & Chien, 2009
ANN	Delay and Cost overrun percentages	Schedule	Wang et al., 2012

ML algorithms employed	Objective of the predictive analysis	Factor	Study Reference
ML Regression techniques	Cost Performance Index (CPI) & To Complete CPI	Cost	Jaber et al., 2020
	Schedule Performance Index	Schedule	
ANN (PCA, Modular Neural Network, PNN/TLRNs)	Cost overrun percentage	Cost	El-Kholy, 2021
	Delay overrun percentage	Schedule	
Multilayer perceptron (MLP)	Risk of delay	Risk	Sanni-Anibire et al., 2022
NN	Estimation of S Curve	Schedule	Chao & Chien, 2009
3-layer MLP	Time-series forecasting of thermal load	Energy	Rahman & Smith, 2018
Recurrent Neural Network (RNN)	Power and energy consumption		Mocanu et al., 2016
Conditional Restricted Boltzmann Machines (CRBMs)			
Factored conditional restricted Boltzmann machine (FCRBM)			
ANN trained by: 1. Levenberg-Marquardt algorithm 2. Bayesian regularization	Yearly total energy demand		Geyer & Singaravel, 2018
ANN	Customer Satisfaction	Satisfaction*	Li et al., 2018
Focus Area 2 – Support Vector Machines (SVM)			
Soft-margin SVMs	Class of constructability from identified risk sources	Risk	Kifokeris & Xenidis, 2019
Sequential minimal optimization (SMO) – SVM	Risk of delay		Sanni-Anibire et al., 2022
SVM	Cost Growth	Cost	Wang et al., 2012
	Schedule Growth	Schedule	
SVM	Aggregated Active Power (kW)	Energy	Mocanu et al., 2016
	Energy sub-metering (Wh)		
SVM	Occurrence and severity of accidents	Safety	Poh et al., 2018
SVM	Customer Satisfaction	Satisfaction*	Pandey et al., 2023
SVM			Li et al., 2018
Focus Area 3 – Logistic Regression (LogR)			

ML algorithms employed	Objective of the predictive analysis	Factor	Study Reference
LogR	Cost Growth	Cost	Wang et al., 2012
	Schedule Growth	Schedule	
LogR	Occurrence and severity of accidents	Safety	Poh et al., 2018
LogR	Customer Satisfaction	Satisfaction*	Pandey et al., 2023
Focus Area 4 – Decision Tree (DT)			
DT	Construction project costs	Cost	Gu, 2023
DT	Occurrence and severity of accidents	Safety	Poh et al., 2018
DT	Weight carried by construction workers		Lee & Son, 2021
DT	Project delay risk	Schedule	Gondia et al., 2019
C4.5 DT	Identification of construction delay factors		Kim et al., 2008
Focus Area 5 – Naïve-Bayes Classifier (NBC)			
NBC	Project delay risk <30% Time Overrun (TO)	Schedule	Gondia et al., 2019
	Project delay risk 30-60% TO		
	Project delay risk >60% TO		
Bayesian Networks	Factors of construction delays		Kim et al., 2008
NBC	Customer Satisfaction	Satisfaction*	Pandey et al., 2023
Focus Area 6 – K-Nearest Neighbor (KNN)			
KNN (IBk)	Risk of delay	Risk	Sanni-Anibire et al., 2022
KNN	Occurrence and severity of accidents	Safety	Poh et al., 2018
KNN	Degree of possibility and degree of damage of risk factors for deep foundation construction safety		P. Gong et al., 2020
Focus Area 7 – Random Forest (RF)			
RF	Occurrence and severity of accidents	Safety	Poh et al., 2018
RF	Weight carried by construction workers		Lee & Son, 2021

ML algorithms employed	Objective of the predictive analysis	Factor	Study Reference
RF	Construction injury type, energy type, and body part		Tixier et al., 2016
RF	Green building construction cost	Cost	Alshboul et al., 2022
Random Forest Regressor (RFR)	Construction accident cost		Xia et al., 2024
RF	Construction project costs		Gu, 2023
Random Forest Classifier	Customer Satisfaction	Satisfaction*	Pandey et al., 2023
Focus Area 8 – Deep Neural Network (DNN)			
DNN	Risk-relevant factors	Risk	Ajayi et al., 2020
DNN	Green building construction cost	Cost	Alshboul et al., 2022
DNN	Complexity of accident chains or networks	Safety	Zhou et al., 2014
Deep RNN	Time-series forecasting of thermal load	Energy	Rahman & Smith, 2018
Deep RNN	Medium to long-term Electric Load		Rahman et al., 2018
Focus Area 9 – Gradient Boosting Model/Machine (GBM)			
GBM	Risk-relevant factors	Risk	Ajayi et al., 2020
GBM	Weight carried by construction workers	Safety	Lee & Son, 2021
Light GBM			
Extreme gradient boosting (XGBOOST)	Green building construction cost	Cost	Alshboul et al., 2022
XGBoost	Customer Satisfaction	Satisfaction*	Pandey et al., 2023
XGBoost	Customer Satisfaction	Satisfaction*	Li et al., 2018
Focus Area 10 – Stochastic Gradient Tree Boosting (SGTB)			
SGTB	Construction injury type, energy type, and body part	Safety	Tixier et al., 2016
Focus Area 11 – Ensemble/Stacking			
ANN Ensemble	Cost Growth	Cost	Wang et al., 2012
	Schedule Growth	Schedule	
ANN (MLP) + SVM (SMO) – ANN combining classifier	Risk of delay	Risk	Sanni-Anibire et al., 2022
ANN (MLP) + SVM (SMO) – SVM combining classifier			
Binary Particle Swarm Optimization (BPSO) +AdaBoost+SVM	Degree of possibility and degree of damage of risk factors for deep	Safety	P. Gong et al., 2020
AdaBoost +SVM			

ML algorithms employed	Objective of the predictive analysis	Factor	Study Reference
AdaBoost +KNN	foundation construction safety		
Association Rule Mining + Bayesian Network + Swiss Cheese Model	Risk of construction defects	Risk	Fan, 2020
Multiple attribute decision-making (MADM) algorithm + Core Vector Machine (CVM)	Risk of financial decision	Risk	Hsu, 2019
Evolutionary Support Vector Machine Inference Model (ESIM) = hybrid of SVM and fast messy genetic algorithm (fmGA)	Degrees of overall project success	Success	Cheng et al., 2010
Recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM)	Customer Satisfaction	Satisfaction*	Pandey et al., 2023

** Not specific to the construction industry but selected for reference of the current study's dataset*

Reviewing publications and existing scholarly literature, there were exciting findings and interpretations. For ease of reference and distinction between features of datasets, a classification of studies was developed according to the primary factors being considered by the ML algorithmic prediction models as parameters of construction, which are (1) Cost and Schedule, (2) Risk and Safety, (3) Energy and Success.

Cost & Schedule

In a study conducted by Chandanshive & Kambekar, 2019, with ANN[1], early stopping and Bayesian regularization approaches were implemented to improve the generalization competency of neural networks and mitigate the overfitting of the model. The performance of the Bayesian regularization approach was better at predicting the construction cost of the building at the early stage of construction. Then, in the study by Q. Jiang, 2020, the average error of the Backpropagation neural network (BPNN)[2] model was observed to be 5.54%, which was much lesser than that for Radial basis function neural network (RBFNN)[2] at 11.08%. Further, Hashemi et al., 2019, deployed the Artificial Neural Network (ANN)[3] model trained through the genetic algorithm (GA) application. GA was held accountable for selecting the best ANN architecture based on the evolution of computing capabilities. In a study by Wang et al., 2012, an ANN ensemble with bootstrap aggregating and adaptive boosting was developed for the prediction model to improve prediction accuracy. The overall prediction accuracy for cost success was highest, with SVMs at 92% and Adaptive Boosting ANNs at 84%. The overall prediction accuracy for schedule success was highest, with Adaptive Boosting ANNs at 80% and SVMs at 76%.

Risk & Safety

In the study by Kifokeris & Xenidis, 2019, data was pre-processed with regularized stochastic gradient descent non-negative matrix factorization to manage missing values and factorize the data into vectors. The SVM[5] model was trained and validated using the n-fold cross-validation method. Moreover, in a study by Sanni-Anibire et al., 2022, out of the ensemble methods and ANN and SVM[6] models, the highest prediction accuracy was achieved with the ANN model at 93.75%. Ajayi et al., 2020, employ certain techniques to boost the success rate of DNN[7,] which include Rectified Linear Unit (ReLU), Dropout technique – a DNN regularization scheme for preventing overfitting, and Cross-entropy objective with Softmax activation. A study by Lee & Son, 2021, evaluated the effectiveness of a weight-tracking system developed using smart safety shoes with sensors attached to a mobile device for collecting initial sensing data and a web-based server computer for storing, preprocessing, and analyzing such data. The average accuracy classifying the weight by each classification algorithm showed similar but high accuracy in the following order: random forest[8] (90.9%), light GBM (90.5%), decision tree (90.3%), and GBM (89%).

In their study, Poh et al., 2018 compare the classification performance of risk of the degree of accident across models developed using a decision tree, random forest, logistic regression, KNN and SVM. It was observed that the random forest model had the best accuracy. Moreover, P. Gong et al., 2020, documented that mature classification methods, including decision tree, Bayesian, artificial neural networks, KNN, and SVM, show poor performance on imbalanced data sets and unsatisfactory classification results. Further, the study also deduces that selection is necessary before the formal evaluation of the model to reduce the redundant features and noise in the data set, preventing the risk of the minority samples being regarded as noise and improving the classification effect of minority groups. Infrequent occurrences of high-risk construction incidents and even rarer safety accidents contribute to a scarcity of recorded incidents, predominantly low or general-risk events (impossible, rare, or occasional). Consequently, the *data imbalance ratio* tends to exceed 30 *(maximum imbalance ratio = maximum sample number/least sample number)* within the specific safety risk evaluation information system. Despite this imbalance, the primary objective of construction safety risk assessments is to prioritize the accuracy of high-risk grade evaluations, mainly when dealing with a limited sample size. The conventional K-nearest neighbors (KNN) algorithm, which selects K samples closest to the classification object and relies on majority voting based on class labels, introduces significant bias toward the majority in its classification outcomes. Thus, this study conducted an integrated algorithmic experiment that used SVM as the base classifier, AdaBoost as the integrated framework, and BPSO[9] for feature selection to ensure that the machine learning effect of the imbalanced data set was suitable for practical applications. In a study by Hsu, 2019 it was discussed that SVM10 was based on the statistical learning theory and the principle of structural risk minimization (SRM) among all the ML-based methods. As a method, SVM has been documented to have the following advantages: (1) it only has two free parameters to be decided; (2) its solution is optimal and unique; and (3) it performs well in a small dataset.

Energy and Success

In a study by Cheng et al., 2010, a hybrid project success prediction model is developed wherein the role of SVM[11] is primarily with learning and curve fitting, while fast messy genetic algorithm (fmGA) deals primarily with optimization.

Statistical and ML-based Model Implementation

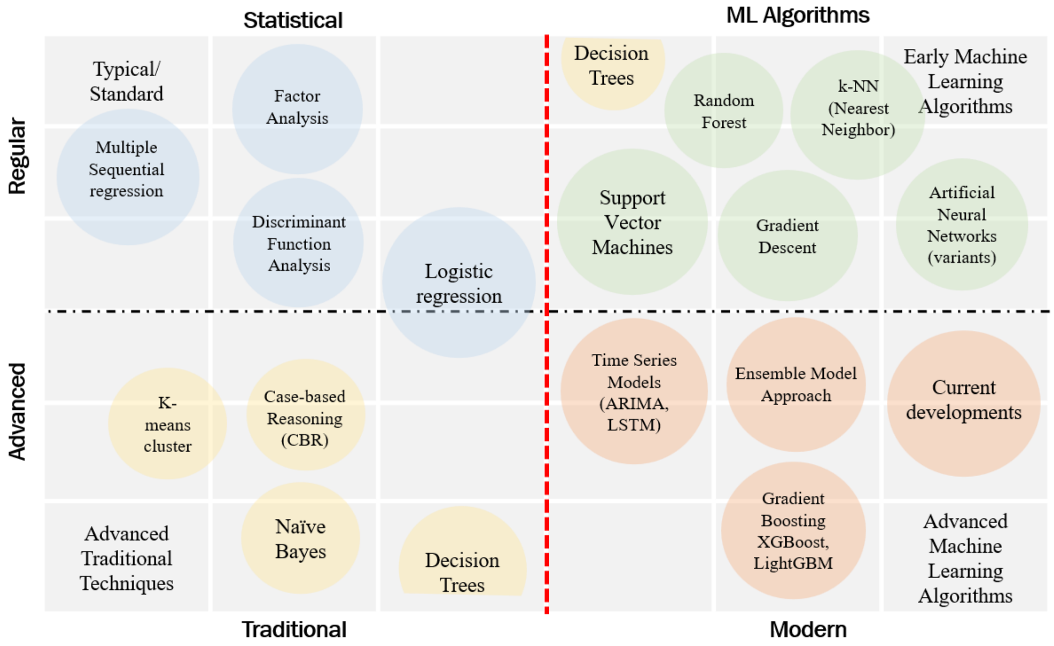

Statistical
ML Algorithms
Regular
Advanced
Typical/ Standard
Multiple Sequential regression
Factor Analysis
Discriminant Function Analysis
Logistic regression
Decision Trees
Random Forest
k-NN (Nearest Neighbor)
Early Machine Learning Algorithms
Support Vector Machines
Gradient Descent
Artificial Neural Networks (variants)
K-means cluster
Case-based Reasoning (CBR)
Time Series Models (ARIMA, LSTM)
Ensemble Model Approach
Current developments
Advanced Traditional Techniques
Naïve Bayes
Decision Trees
Gradient Boosting XGBoost, LightGBM
Advanced Machine Learning Algorithms
Traditional
Modern

Figure 15: Conceptual Quadrant Study Map for ST and ML Techniques

Studying and reviewing peer-reviewed publications and books, it was deduced that the techniques used for prediction can be mapped on a conceptual quadrant illustrated in Fig. 15. This map documents the statistical and ML-based prediction modeling techniques across two axes – “regular” to “advanced” on the vertical axis and “traditional” to “modern” on the horizontal axis. When implemented for designing the prediction model, statistical methods and ML algorithms undergo a step-by-step process that remains similar primarily across different types and objectives of prediction. Fig. 16 below illustrates the methodology adopted for designing and developing a prediction model based on statistical and ML algorithms.

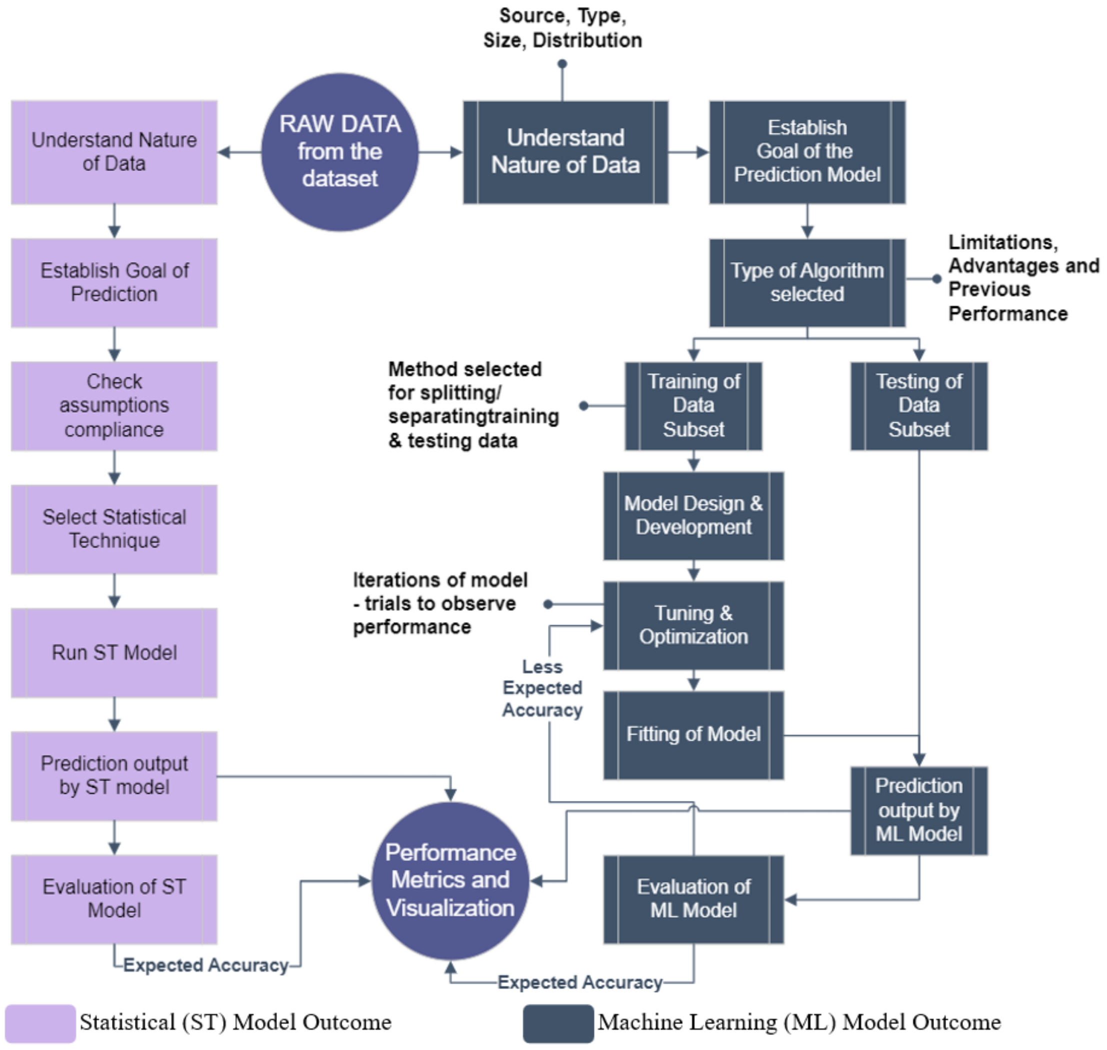

Source, Type, Size, Distribution
Understand Nature of Data
RAW DATA from the dataset
Understand Nature of Data
Establish Goal of the Prediction Model
Establish Goal of Prediction
Type of Algorithm selected
Limitations, Advantages and Previous Performance
Check assumptions compliance
Method selected for splitting/ separatingtraining & testing data
Training of Data Subset
Testing of Data Subset
Select Statistical Technique
Model Design & Development
Iterations of model - trials to observe performance
Tuning & Optimization
Run ST Model
Less Expected Accuracy
Fitting of Model
Prediction output by ST model
Prediction output by ML Model
Performance Metrics and Visualization
Evaluation of ST Model
Evaluation of ML Model
Expected Accuracy
Expected Accuracy
Statistical (ST) Model Outcome
Machine Learning (ML) Model Outcome

Figure 16: Methodology for ST and ML-based techniques for the prediction model

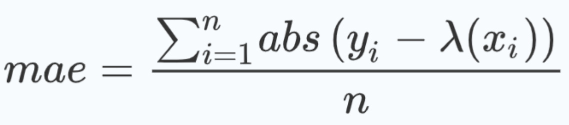
Performance Evaluation of Statistical Prediction Models

The predictive power of statistical models is evaluated through the coefficient of determination (R^2), which measures the goodness of fit for the model. It is also a measure of the strength of the correlation when more than two variables are part of the analysis. However, increasing the number of independent variables or predictors in the model decreases the Sum of Squared Errors (SSE), which increases the value of R^2. Thus, for an unbiased estimate, researchers suggest using adjusted R^2 for model evaluation and selection (Ling et al., 2004; Meyers et al., 2017a).

Performance Evaluation of ML Prediction Models

The performance analysis of ML prediction models determines the model's efficacy in predicting the test/trial data by comparing the observed and the predicted values (Cali et al., 2021). The following terms are used for the evaluation of ML prediction models:

1. **Mean Absolute Error (MAE)** is a model evaluation metric used for regression models. MAE of a model concerning a test set is the meaning of the absolute values of the individual prediction errors on overall instances in the test set. Each prediction error is the difference between the true and predicted values for the instance (Sammut & Webb, 2011b).



$$mae = \frac{\sum_{i=1}^{n} abs\left(y_i - \lambda(x_i)\right)}{n}$$

Figure 17: Equation for Mean Absolute Error

   In Fig. 17 above, where y_i is the true target value for test instance x_i, $\lambda(x_i)$ is the predicted target value for test instance x_i, and n is the number of test instances.

2. **Mean Squared Error (MSE)** is a model evaluation metric used for regression models concerning a test set. It is the mean of the squared prediction errors over all instances in the test set. The prediction error is the difference between an instance's true and predicted values (Sammut & Webb, 2011c).

$$mse = \frac{\sum_{i=1}^{n} \left(y_i - \lambda(x_i)\right)^2}{n}$$

Figure 18: Equation for Mean Squared Error

In Fig. 18 above, y_i is the true target value for test instance x_i, $\lambda(x_i)$ is the predicted target value for test instance x_i, and n is the number of test instances.

3. **Confusion Matrix** summarizes the classification performance of a classifier model with respect to some test data. It is a two-dimensional matrix, indexed in one dimension by the true class of an object and in the other by the class the classifier assigns (Poh et al., 2018; Ting, 2011a). In this context, the four cells of the matrix are designated as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), as indicated in Table 3.

Table 3: Confusion Matrix

		Predicted Condition	
		Positive (YES)	**Negative (NO)**
Actual Condition	**Positive (YES)**	True Positive (TP)	False Negative (FN)
	Negative (NO)	False Positive (FP)	True Negative (TN)

A few classification performance measures are defined in terms of these four classification outcomes:

a. *Specificity* = True negative rate = TN / (TN + FP)

b. *Sensitivity* = True positive rate = Recall = TP/ (TP + FN)

c. *Positive predictive value* = Precision = TP / (TP + FP)

d. *Negative predictive value* = TN / (TN + FN)

4. **Error rate** refers to a measure of the degree of prediction error of a model made with respect to the true model applied in the context of classification models. In this context, ***error rate = P(λ(X)≠Y)***, where XY is a joint distribution, and the classification model λ is a function X → Y (Ting, 2011b). Sometimes, this quantity is expressed as a percentage rather than a value between 0.0 and 1.0. The error rate of a classifier on test data may be calculated as several incorrectly classified objects/total number of objects. The error rate is directly related to accuracy, such that error rate = 1.0 − accuracy (or when expressed as a percentage, error rate = 100 − accuracy).
5. **Accuracy** describes the probability of the correct prediction relative to the total number of predictions, and it can be used as a single measure to evaluate the overall performance of a model (Poh et al., 2018). refers to a measure of the degree to which the predictions of a model match the reality being modeled. The term accuracy is often applied in the context of classification models. In this context, accuracy = P(λ(X) = Y), where XY is a joint distribution, and the classification model λ is a function X → Y. Sometimes, this quantity is expressed as a percentage rather than a value between 0.0 and 1.0 (Sammut & Webb, 2011a). Accuracy is calculated below in Equation (A).

$$Equation\ A: Accuracy\ =\ \frac{1}{2}\left(\frac{TP}{TP\ +\ FN}+\frac{TN}{FP\ +\ TN}\right)$$

CHAPTER FIVE - RESULTS AND DISCUSSION

Data Understanding

The construction coating sector is an integral sector of the construction industry wherein the coatings are applied to the finished surface, such as walls, roofs and floors, to improve surface properties. Considering the specialized skills required for executing the coating applications, Facility Managers (FMs) hire external vendors that are qualified and competent with a proven excellent customer satisfaction record (Gajjar et al., 2024). The dataset used for the study was sourced from one of the largest construction coating manufacturers, which gave a list of projects completed every month for four years. The manufacturer conducted a post-occupancy evaluation (POE) survey to collect information about the manufacturer’s product and the performance of the applicator assigned for the product installation (Gajjar et al., 2024). A total of 2,401 end users responded to the survey documenting the region, season, temperature at which the installation was completed and the overall customer satisfaction score (scale of 1 to 10; 1 being lowest – 10 being highest) for each project.

Independent Variables

Region: After initial data sorting, factorial classification by region (the location of the installation job) was done. U.S. Census Bureau classifies the country into four regions: Midwest, Northeast, South, and West (United States Census Bureau, 2013) based on the Geographic Names Information System (GNIS) identifying its geographical location.

Table 4 below highlights the classification of the coating projects according to the different regions identified, reflecting the South region having the maximum total jobs.

Table 4: Classification of Jobs per Region

Job Region	Total Jobs (#) (N = 2401)	Percentage of Total Jobs	Standard Deviation in Overall Customer Satisfaction
Midwest	348	14.5%	1.76
Northeast	319	13.3%	1.39
South	999	41.6%	1.30
West	735	30.6%	1.40

Season: Based on the month in which the coating system was installed, the data was classified into four distinct seasons – jobs completed in December, January, and February were grouped under the “Winter” season; those in March, April, and May were categorized under “Spring” season; jobs in June, July, and August were categorized in “Summer,” and jobs installed in September, October, and November were categorized as “Fall.” Table 5 below highlights the categorization of seasons in the dataset.

Table 5: Classification of Jobs per Season

Job Season	Total Jobs (#) (N = 2401)	Percentage of Total Jobs	Standard Deviation in Overall Customer Satisfaction
Fall	773	32.2%	1.61
Spring	483	20.1%	1.29
Summer	784	32.7%	1.22
Winter	361	15.0%	1.58

Temperature: For each job, average temperatures were recorded based on the installation date of the job, assisting in the classification of factors according to the installation temperature (The Weather Company, 1995). The temperatures were categorized into increments of 10 °F starting from 21-30 °F until 91-100 °F, as seen in Table 6 below.

Table 6: Classification of Jobs per Installation Temperature Range

Temperature Range (°F)	Total Jobs (#) (N = 2401)	Percentage of Total Jobs	Standard Deviation in Overall Customer Satisfaction
20-30	36	1.5%	1.77
31-40	137	5.7%	1.19
41-50	266	11.1%	1.60
51-60	400	16.7%	1.82
61-70	608	25.3%	1.20
71-80	627	26.1%	1.32
81-90	293	12.2%	1.39
91-100	34	1.4%	0.91

Dependent Variables

Overall Customer Satisfaction: In the post-occupancy evaluation (POE), the data was collected with a customer satisfaction rating on a scale of 1 to 10 – with 1 (one) being the lowest and 10 being the highest. Fig. 19, 20, and 21 below illustrate the distribution of overall customer satisfaction ratings with respect to each of the independent variables: job region, job season and installation temperature. The nature of the variable is ordinal; hence, the distribution is investigated using boxplot distributions for each of the IVs (categorical). The DV was transformed to a categorical nature for the ST techniques and treated as a continuous IV with numeric coding.

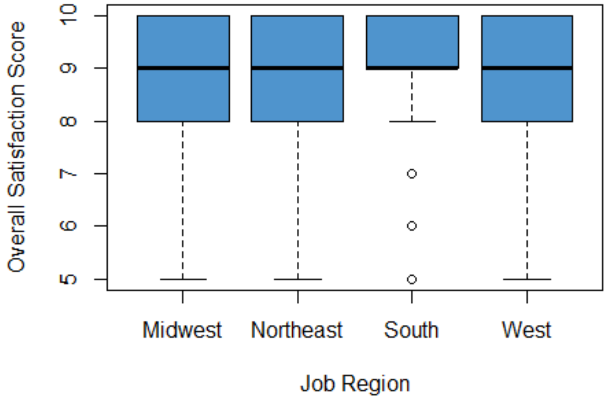


Figure 19: Distribution of Overall Customer Satisfaction-Job Regions

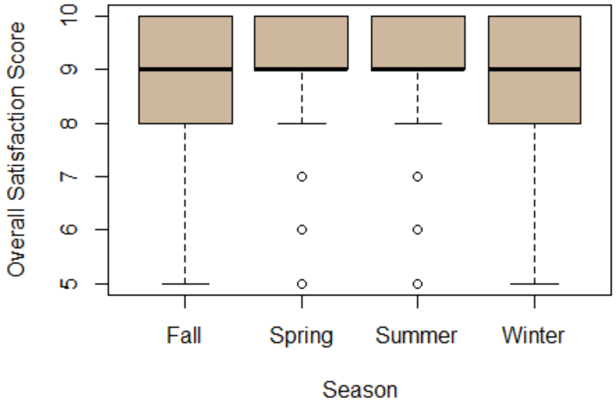


Figure 20: Distribution of Overall Customer Satisfaction-Seasons

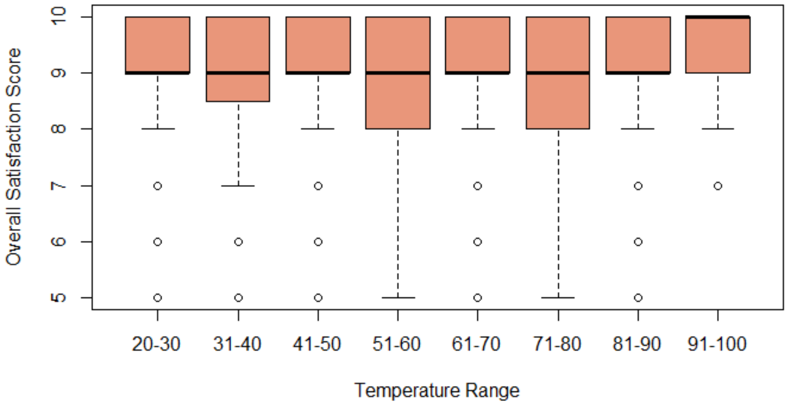


Figure 21: Distribution of Overall Customer Satisfaction-Temperature Ranges

An outlier analysis was conducted on the dataset, with the upper and lower limits calculated using 1.5 times the interquartile range. This resulted in the dataset having 2355 observations, with a minimum overall customer satisfaction value of five (5). Figures 22, 23, and 24 below represent the data distribution after removing the outliers. The IV of temperature range was transformed to a categorical variable of eight (8) levels to that of three (3) levels, namely, "High" for 71-100 °F, "Medium" for 51-70 °F and "Low" for 20-50°F as illustrated in Fig. 25 Further, the DV, "Overall Customer Satisfaction" was transformed from ordinal nature of 1-10 to ordinal nature of three categories, "Good" for customer satisfaction rating of 10, "Neutral" for customer satisfaction rating of 8-9, and "Poor" for customer satisfaction rating of "5, 6, 7".

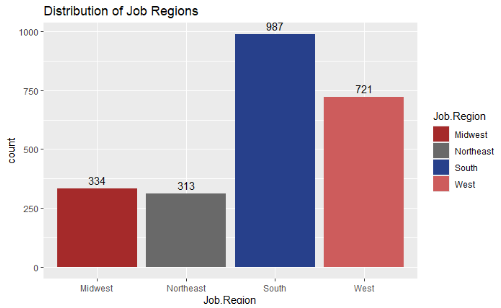


Figure 22: Distribution of Job regions

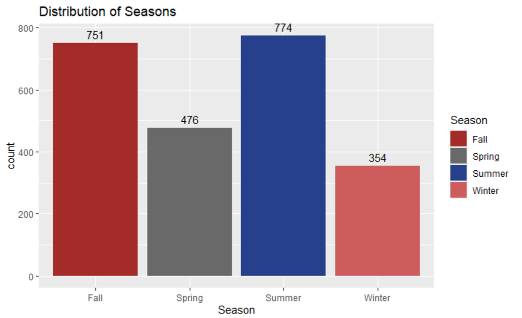


Figure 23: Distribution of Seasons

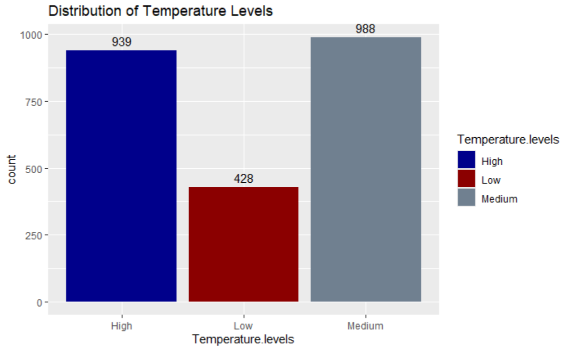


Figure 24: Distribution-Temperature Range Levels

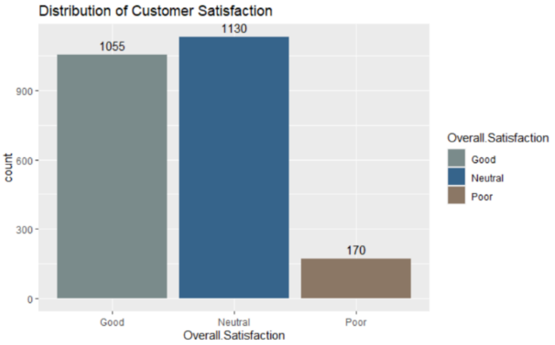
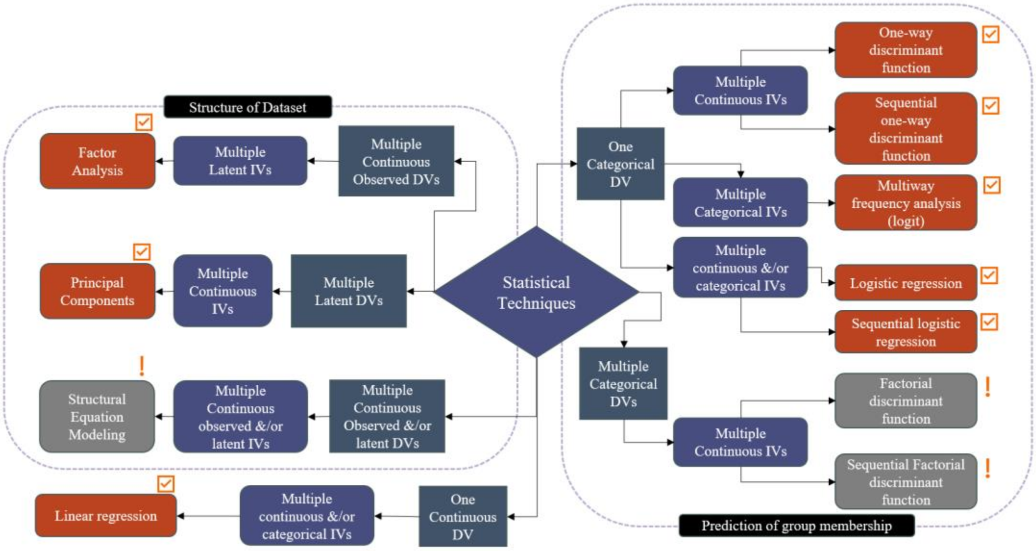


Figure 25: Distribution of Customer Satisfaction

The dataset presents challenges in its characteristics: a limited number of IVs in the equation and a visibly skewed/non-parametric distribution.

Statistical-Regular Prediction Modeling Techniques

The developed "Conceptual Quadrant Study Map for ST and ML Techniques," as illustrated in Fig. 15, functioned as the guiding compass for the techniques available for formulating prediction models. The methods from the statistical-regular (top left quadrant) of the world of prediction modeling were tested for the dataset. Fig. 26 below delineates the ST techniques available for prediction modeling, and the methods highlighted in orange with a tick mark symbol next to them apply to the dataset considered. The methods in grey with an exclamation mark symbol next to them do not apply to the dataset.


Structure of Dataset
Factor Analysis
Multiple Latent IVs
Multiple Continuous Observed DVs
Principal Components
Multiple Continuous IVs
Multiple Latent DVs
Structural Equation Modeling
Multiple Continuous observed &/or latent IVs
Multiple Continuous Observed &/or latent DVs
Linear regression
Multiple continuous &/or categorical IVs
One Continuous DV
Statistical Techniques
One Categorical DV
Multiple Continuous IVs
One-way discriminant function
Sequential one-way discriminant function
Multiple Categorical IVs
Multiway frequency analysis (logit)
Multiple continuous &/or categorical IVs
Logistic regression
Sequential logistic regression
Multiple Categorical DVs
Multiple Continuous IVs
Factorial discriminant function
Sequential Factorial discriminant function
Prediction of group membership

DV – Dependent Variable or Outcome
IV – Independent Variable or Predictor
*Tabachnick, B. G., & Fidell, L. S. (2013). Using Multivariate Statistics (Sixth). Pearson Education, Inc.

Figure 26: Statistical-Regular Prediction Modeling Methods

The methods, namely, Structural equation modeling, Factorial discriminant function and Sequential factorial discriminant function, can be employed in datasets with multiple DVs. In contrast, for the present study, the dataset has one (1) DV, i.e., overall customer satisfaction.

Factor Analysis/Principal Component Analysis

Statistical techniques such as principal components analysis (PCA) and factor analysis (FA) can be used on a single set of variables to identify whether variables form coherent subsets that are comparatively independent of each other. The method of PCA/FA summarizes patterns of correlations among observed variables to reduce many observed variables to fewer factors with a minimum loss of information. It is thus often implemented for dimensionality reduction (Kim et al., 2008). The main objective of this technique is to provide a regression equation for an underlying process by using observed variables or to test a theory about the nature of underlying processes. FA/PCA creates linear combinations of observed variables to represent latent (hidden or unobserved) variables, also called factors (PCA produces components, while FA produces factors). The steps include the selection and measurement of a set of variables, preparing the correlation matrix followed by extracting a set of factors from the correlation matrix, determining the number of factors, (probably) rotating the factors to increase interpretability, and, finally, interpreting the results (Tabachnick & Fidell, 2013).

Results: Firstly, Bartlett's test of sphericity was conducted to evaluate whether the variables are correlated with one another and if this test is not statistically significant, FA cannot be employed. The result was $p < 0.001$; hence, this was followed by the Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy – which observes if the partial correlations within the dataset are approaching zero, suggesting at least one latent factor underlying the variables (Tabachnick & Fidell, 2013). The minimum acceptable value is

0.50, and in our result, it was 0.516. The code below was run on RStudio 2024.04.2 to conduct FA/PCA, detailed in Appendix A.

As evident from the code excerpt in Appendix A, there is only one factor with eigenvalue > 1, which accounts for 29% of the total variance in the variables. Fig. 27 below also illustrates the scree plot and parallel analysis scree plots employed for understanding the number of factors that can be extracted from the dataset. The scree plots the eigenvalues for all the factors in the dataset, which measure the amount of variance accounted for by a factor (Meyers et al., 2017). The parallel analysis scree plots map the eigenvalues from the FA and a comparison of the eigenvalues from the random correlation matrices to that of the observed data.

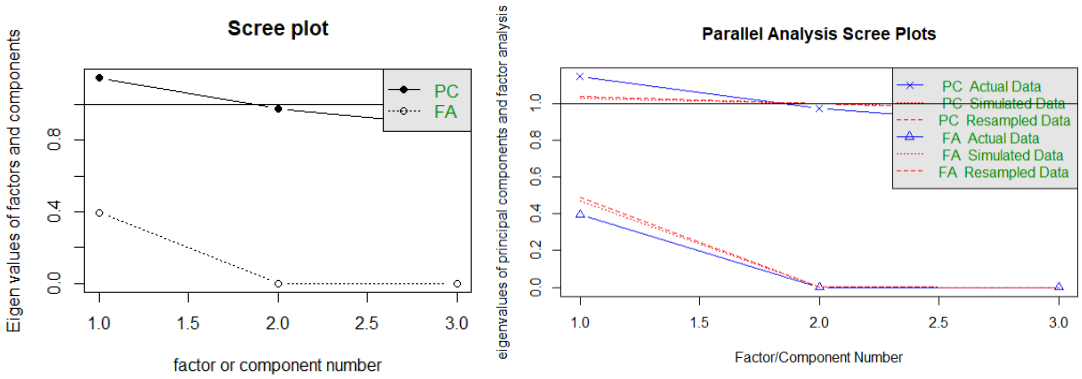


Figure 27: Scree plot and Parallel analysis scree plots for PCA and FA

Observed eigenvalues higher than their corresponding random eigenvalues are more likely to be from "meaningful factors" than observed eigenvalues below their corresponding random eigenvalue (Meyers et al., 2017). The code for running the analysis is shown in Appendix A, and the details of the model developed for performing FA/PCA are tabulated in Table 7 below.

Table 7: Results for model developed on FA/PCA

Variables	**Nature of IV**	**Results**
Region (coded as 1,2,3,4)	Continuous	Bartlett's test of sphericity is significant, p < 0.001 KMO Measure of Sampling Adequacy = 0.516
Season (coded as 1,2,3,4)	Continuous	1 (one) component with an Eigenvalue greater than 1 (one) accounts for 29% of the variance. Pattern matrix could not be generated with only one component extracted.
Average Temperature at Installation (in °F)	Continuous	Factor loadings for each variable: ***Job region: 0.720*** ***Season: 0.541*** ***Average temperature: 0.580***

Advantages: FA/PCA are preferred techniques for dimensionality reduction, primarily when visualizing high-dimensional data with no requirement for multivariate normality. Multivariate normality is assumed in FA when statistical inference is used to determine the number of factors, and the assumption is that the correlation matrix of the variables cannot be an identity matrix (Tabachnick & Fidell, 2013).

Limitations: There are no readily available criteria against which to test the solution. After extraction, infinite rotations are available, accounting for the same amount of variance in the original data but with the factors defined slightly differently (Tabachnick & Fidell, 2013).

Conclusions: The squared loading determined from the standardized loadings is the correlation between each variable and each PC. Thus, ***the correlation between the Job***

region and the first PC (PC1) is 0.72, the correlation between the Job season and PC1 is 0.54, while the correlation between the Average Temperature and the PC1 is 0.58. This method can be applied to datasets with high dimensionality, i.e., a higher number of predictors or IVs, and it plays a crucial role in feature selection and determining critical IVs that contribute to the outcome or DV.

Linear Regression

The linear regression model describes a relationship between the DV or the response/outcome variable and one or more IVs or predictor variables by generalizing a straight line or a linear equation (Hwang, 2009). Assuming there is a vector of random variable **X**, a linear regression model can be represented in matrix terms as Eq. (1), where $\mathbf{Y} = [y_1, \ldots, y_n]^T$, $\mathbf{e} = [e_1, \ldots, e_n]^T$, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_n]^T$

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{e} \tag{1}$$

If the model is valid and appropriate, then the observed value y_i can be determined from the value of x_i with the Eq (1), except for e_i, the unknown random quantity of statistical error for the *i*th case that represents the failure of the model to determine the fitted value. Assumptions in the model include normality and equality of covariance for the errors (Tabachnick & Fidell, 2013). However, when the dataset is larger than the number of discriminating variables, and the sample size of each group is almost the same – the assumptions of normality and covariance can be relaxed (Sharma, 1995). The selected model minimizes the residual sum of squares of errors, and the strength of the established relationship is represented by the coefficient of determination R^2, which represents the proportion of the variability of response explained by regression on predictors (Hwang, 2009).

Results for Linear Regression with Ordinal DV (0, 1, 2): Given the three predictors, the study included developing six (6) types of models with different combinations of IVs and the transformed nature of IVs with results tabulated in Table 8 below. Partial F-tests were also conducted to understand if the addition of the predictors with each new model was statistically significant or not. The adjusted R^2 value was found to be maximum for Model 6, with all the IVs being input in the model as categorical variables and the DV having three (3) possible outcomes and ordinal: 0 (Poor), 1 (Neutral), and 2 (Good). The code for linear regression with ordinal DV (0, 1, 2) is detailed in Appendix B.

Table 8: Summary of Linear regression models with ordinal DV (0, 1, 2)

Model#	IVs in the Model	Nature of IVs	F-statistic	p-value	Adjusted R^2	p-value of partial F-test
Model1	Region (coded as 1,2,3,4)	Continuous	F (1,2353) = 15.98	6.61e-05	0.0063	NA
Model2	Region	Categorical	F (3, 2351) = 7.667	4.249e-05	0.0084	NA
Model3	Region and Season (coded as 1,2,3,4)	Continuous	F (2, 2352) = 8.75	0.00016	0.0065	0.2178
Model4	Region and Season	Categorical	F (6, 2348) = 4.949	4.817e-05	0.001	0.0839
Model5	Region, Season (coded as 1,2,3,4) and Temperature Levels (coded 0,1,2)	Continuous	F (3,2351) = 5.852	0.0005587	0.0061	0.7997
Model6	***Region, Season and Temperature levels***	***Categorical***	***F (8, 2346) = 4.154***	***6.052e-05***	***0.0106***	***0.1724***

Parameter or predictor estimates in multiple linear regression are the unstandardized regression coefficients (β weights), which for the IV represent the change in the DV associated with a one-unit change in that IV if all the other IVs are kept constant (Tabachnick & Fidell, 2013). Table 9 below showcases the predictor estimates and the corresponding p-value from each of the models discussed above, which are statistically significant.

Table 9: Predictor estimates (β) for LR models with ordinal DV (0,1,2)

Model#	**IV**	**IV/Predictor Estimate (β)**	**Standard Error**	***t*-value**	***p*-value**
Model1	Job region code	0.05	0.01	3.997	6.61e-05 ****
Model2	Region-South	0.17	0.04	4.276	1.98e-05 ****
	Region-West	0.14	0.04	3.572	0.000361 ****
Model3	Job region code	0.05	0.01	4.093	4.41e-05 ****
Model4	Region-South	0.17	0.04	4.219	2.55e-05 ****
	Region-West	0.15	0.04	3.615	0.000307 ****
Model5	Job region code	0.05	0.01	4.099	4.29e-05 ****
Model6	Region-South	0.18	0.04	4.467	8.33e-06 ****
	Region-West	0.15	0.04	3.764	0.000171 ****
	Season-Winter	-0.09	0.04	-2.242	0.025055 **
	Temperature Level-Low	0.08	0.04	1.875	0.06 *

*Significant codes: '****' p < 0.001 '***' p < 0.01 '**' p < 0.05 '*' p < 0.1*

Results for Linear Regression with Ordinal DV (5-10): In this linear regression model series, the DV was ordinal from five (5) to ten (10). Three (3) models were tested with the nature of IV retained as categorical for "region" and "season" and ordinal for "temperature." Table 10 below summarizes the results for the models developed and the F-statistic for each model. Table 11 displays the associated p-value of t-tests for each predictor estimate that was statistically significant. It is to be noted that the dataset was not subject to any transformations. The code for linear regression with ordinal DV (5-10) is detailed in Appendix C.

Table 10: Summary of LR models with ordinal DV (5-10)

Model#	IVs in the Model	Nature of IVs	F-statistic	p-value	Adjusted R^2	p-value of partial F-test
Model1	Region	Categorical	F (3, 2351) = 5.977	0.0004689	0.0063	NA
Model2	Region and Season	Categorical	F (6, 2348) = 5.382	1.561e-05	***0.01105***	0.0026 ***
Model3	Region, Season and Temperature levels	Categorical	F (13, 2341) = 3.443	2.597e-05	0.01331	0.089 *

*Significant codes: '****' p < 0.001 '***' p < 0.01 '**' p < 0.05 '*' p < 0.1*

Table 11: Predictor estimates (β) for LR models with ordinal DV (5 – 10)

Model#	IV	IV/Predictor Estimate (β)	Standard Error	*t*-value	*p*-value
Model1	Region-South	0.27	0.07	3.971	7.39e-05 ****
	Region-West	0.22	0.07	3.096	0.00198 ***
Model2	Region-South	0.28	0.07	4.000	6.53e-05 ****
	Region-West	0.23	0.07	3.195	0.00141 ***
	Season-Winter	-0.18	0.07	-2.594	0.00956 ***
Model3	Region-South	0.30	0.07	4.169	3.17e-05 ****
	Region-West	0.25	0.07	3.361	0.00079 ****
	Season-Spring	0.12	0.07	1.771	0.07661 *
	Season-Winter	-0.24	0.08	-3.129	0.00178 ***

*Significant codes: '****' p < 0.001 '***' p < 0.01 '**' p < 0.05 '*' p < 0.1*

Advantages: Linear regression generates coefficients equivalent to contributing estimates of each predictor/IV towards the DV. This is the only method that does not predict the dataset's structure nor the outcome's group membership; instead, it supplies the researcher with an equation of coefficients with each IV.

Limitations: The relationships revealed employing linear/multiple linear regression techniques are not implied to be causal. The relationship is ideal when there is a strong correlation between each IV and the DV but uncorrelated with other IVs (Tabachnick & Fidell, 2013). As can be observed in the results summary above for both cases of DV, the regression solution is extremely sensitive to the combination of variables included in the model. For regression, there are some assumptions that the dataset is required to meet, which are listed below:

1. Residuals (differences between obtained and predicted DV scores) are normally distributed for the predicted DV scores.
2. Residuals have a horizontal-line relationship with predicted DV scores.
3. The variance of the residuals about predicted DV scores is the same for all predicted scores across different groups.

However, it needs to be noted that there are no distributional assumptions about the IVs other than their relationship with the DV. Further, a prediction equation is often enhanced if IVs are normally distributed, primarily because the linearity between the IV and the DV is enhanced (Tabachnick & Fidell, 2013). Residual scatterplots must be examined to test the assumptions of normality, linearity, and homoscedasticity between predicted DV scores and prediction errors (Meyers et al., 2017). For Model 2, a q–q plot was developed – in Fig. 28, quantiles of a theoretical distribution are plotted against the quantiles of the observed data. Extreme deviations from a straight line could indicate that the variable is not normally distributed. Variance inflation factor (VIF) is a statistical measure that quantifies the degree of multicollinearity for each predictor variable in a regression model (Meyers et al., 2017). For Model 2, discussed above, none of the VIFs were more significant than 5 – thus, multicollinearity is not a concern for the fitted model.

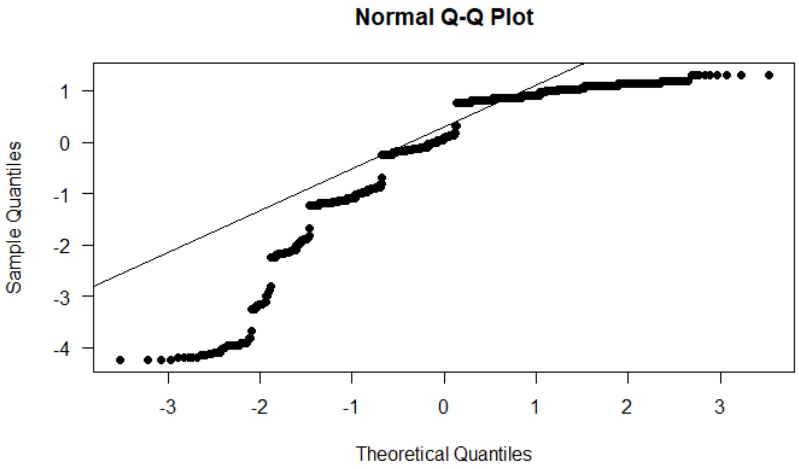


Figure 28: Q-Q Plot for Model 2 - Linear Regression

<u>Conclusions:</u> The model with the highest adjusted R^2 value would be considered the best-fitted model, highlighting the relationship between the IVs and the DV. Model2 with DV in ordinal nature (5-10) was statistically significant, F (6, 2348) = 5.382 ($p < 0.001$) with 1.1% of the variance explained by the model, with only South and West regions as well as Winter season having statistically significant predictor estimates.

Discriminant Function Analysis

The main objective of Discriminant Function Analysis (DFA) is the classification of subjects into one of several categories – i.e., predict group memberships from the set of IVs (Lee, 2001; Tabachnick & Fidell, 2013). In this method, a discriminant score, Y, is calculated for maximizing the separation of groups and minimizing the errors in classification, which represents the DV in multiple regression equations, and X_i's corresponds to the values for the IVs as shown in Eq. (2). The performance metrics for DFA is the proportion of accurate classification (Lee, 2001).

$$\boldsymbol{Y_i = b_0 + b_1 X_{1i} + b_2 X_{2i} + \ldots.} \tag{2}$$

<u>DFA without Data Transformations in SPSS:</u> DFA in IBM SPSS (Statistical Package for Social Sciences) 29 version, using the dataset with categorical IVs, region and season and ordinal IV, temperature range to predict the probability of the dichotomous/binary DV, overall customer satisfaction as 0: No, and 1: Yes.

<u>Results with DFA in SPSS:</u> The detailed result output from SPSS is included in Appendix D. Since the objective of the method is to predict group membership, the results are examined for any significant differences between groups on each of the IVs using the Group Statistics and the Tests of Equality of Group Means tables. Appendix D shows no significant difference between the means of "yes" and "no" in the overall customer satisfaction groups for all IVs.

<u>DFA with Data Transformations in RStudio 2024.04.2:</u> DFA, before being conducted on a dataset, requires two underlying assumptions to be satisfied: multivariate normality and equality of covariance matrices (Lee, 2001). DFA can be affected by the scale/unit in which the predictor variables are measured. Thus, it is generally recommended to standardize/normalize continuous predictors/IVs before the analysis (Meyers et al., 2017b). In Fig. 29, the continuous IV – -average temperature's q-q plot and histogram were plotted to see an approximately normal distribution.

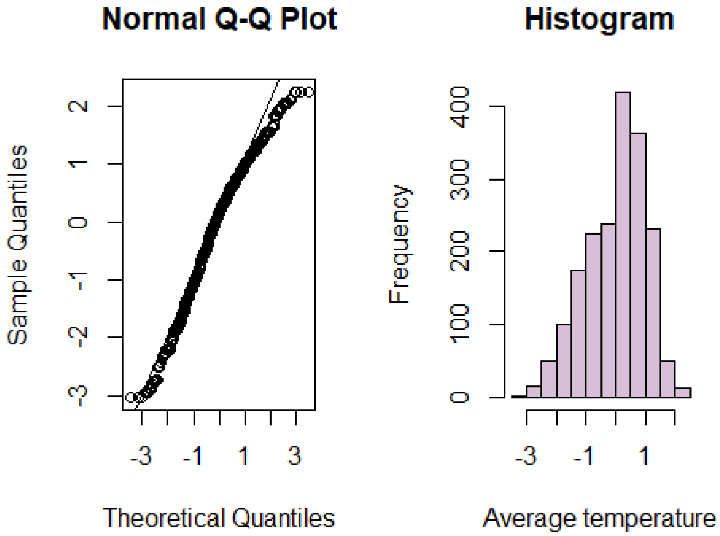


Figure 29: Q-Q plot and histogram for IV - Average Temperature

Data pre-processing: LDA assumes that predictors are normally distributed (Gaussian distribution) and that the different classes have class-specific means and equal variance/covariance (Meyers et al., 2017). Hence, inspecting each variable's univariate distributions and ensuring normal distribution is imperative. Outliers were removed from the data, and the variables were standardized to make their scale comparable. The data was split into training (80%) and testing (20%) sets. This was followed by data normalization, which included estimating the pre-processing parameters and transforming the data accordingly. Normalizing the variable entails that all variables are standardized, each with a mean of 0 and a standard deviation of 1. The code is written in RStudio 2024.04.2 for data pre-processing before DFA, detailed in Appendix E.

Linear Discriminant Analysis (LDA): The LDA algorithm starts by finding directions that maximize the separation between classes and then uses these directions to predict the class of individuals. These directions, called linear discriminants, are a linear combination of predictor variables. The intuition behind the method is to determine a subspace of lower dimension, compared to the original data sample dimension, in which the data points of the original problem are "separable." As illustrated in Fig. 30, data samples in two dimensions are projected in a lower dimension, i.e., a line that must be selected so that projection maximizes the "separability" of the projected samples (Xanthopoulos et al., 2013).

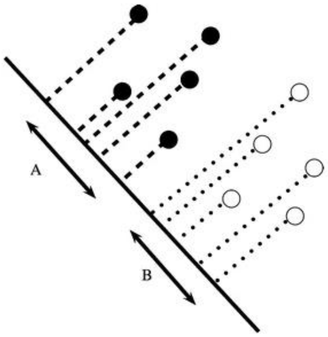


Figure 30: Conceptual schematic for intuition behind LDA

Results for LDA (DFA) in RStudio: LDA determines group means and computes, for each data point, the probability of belonging to the different groups detailed in Table 12 below. The data is affected by the group with the highest probability score. The code detailed in Appendix E outputs the following elements:

- *Prior probabilities of groups*: The proportion of training observations in each group.
- *Group means*: Group center of gravity, which shows the mean of each variable in each group.

- *Coefficients of linear discriminants show the linear combination of predictor variables* to form the LDA decision rule.

Table 12: Summary of results for LDA with Binary DV (Yes/No)

DV	Prior Probabilities of Groups	Group Means – Job Region Code	Group Means - Season Code	Group Means – Avg. Temp.
No	0.072	-0.058	0.138	-0.048
Yes	0.928	0.004	-0.011	0.004

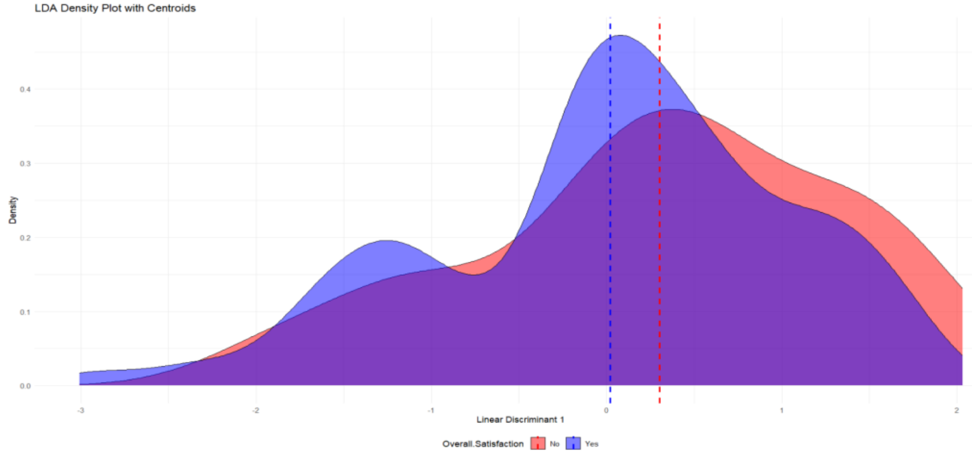


Figure 31: LDA Density Plot with Centroids for Binary DV - Yes/No

In Fig. 31, the density plot for the LDA function is plotted for the scores, highlighting the centroids for each group of DV as dashed vertical lines. It can be observed that even though the accuracy of the model was approximately 93%, the centroids are not well-separated in the input space; thus, the LDA model is not effective in separating the groups.

<u>Advantages:</u> DFA is a one-way analysis that allows sample sizes in groups to be unequal; however, with classification tasks, unequal sample sizes are used to modify the probabilities with which cases are classified. With sample sizes large enough, distortion of

results due to failure of multivariate normality is not expected. Moreover, classification tasks require fewer statistical demands than inference – with approximately 95% accuracy, the shape of distributions should not be a concern.

Limitations: Typically, DFA is used to predict the group membership in groups occurring naturally instead of groups formed by random assignments – which leads to the question of reliability of prediction (Tabachnick & Fidell, 2013). The technique assumes linear relationships among all pairs of IVs within each group – violation of which leads to an increase in Type I error, with a tendency to overestimate the size of association with binary predictors.

Conclusions: If the predictor variables are standardized before computing LDA, the discriminator weights can be used to measure variable importance for feature selection. The weights are ***0.43 for the Job region code, -0.9 for the Season code and 0.27 for Average temperature,*** with the model accuracy turning out to be ***92.78%***. When calculating the CM for the function, the sensitivity (true positive rate) of the model was found to be 0.

Logistic Regression

The method of Logistic Regression (LogR) is an incredibly flexible technique that allows the prediction of a discrete outcome with no assumptions about the distributions of the IVs that need not be normally distributed, linearly related to the DV or of equal variance within each group (Tabachnick & Fidell, 2013). LogR operates based on the natural logarithm, following a logistic S-curve, while the classification of the DV is determined using the probability of the outcome based on the values of its attributes (Poh et al., 2018).

Results of LogR: The LogR was conducted in SPSS, the results of which are detailed in Appendix F. Firstly, the Omnibus Tests of Model Coefficients, which is used to test the goodness of fit of the model, was not statistically significant with $p = 0.054$; which if had been significant would imply that there is a significant improvement in the fit as compared to the null model. However, the Hosmer and Lemeshow Test of good fit was also not statistically significant, with $p = 0.902$, implying the model will adequately fit the data. The model's accuracy was 92.8%, with 0% correct prediction of “No’s.”

Advantages: LogR is a more flexible technique since it has no assumptions about the normal distribution of the IVs, linear relationship to the DV, or equal variance within each group. Moreover, the IVs could be of any nature – continuous, categorical or binary. LogR is the beneficial method when the distribution of the DV is expected to be nonlinear with one or more IVs (Tabachnick & Fidell, 2013). In this method, the model developer can identify which IVs are most predictive of an outcome by examining the magnitude of the β coefficients (parameter estimates) and the corresponding odds ratios (Tu, 1996).

Limitations: Linear regression is more powerful if the DV is continuous and assumptions are satisfied. Overfitting in logistic regression is more challenging to detect with small samples than multiple regression. This is because logistic regression lacks an "adjusted R-squared" metric. A significant difference between adjusted and unadjusted R-squared in multiple regression indicates an insufficient sample size, a warning sign for overfitting.

Conclusions: The Nagelkerke R squared is the adjusted R^2 value, equivalent to 0.014. However, no significant predictor was obtained in the results. The model is also overly optimistic and experiences “overfitting” since it cannot predict any true negatives, only true positives. The model's specificity is 0%, but the sensitivity is 100%, which makes the overall accuracy of the model, i.e., 92.8%, unreliable.

Ordinal Logistic Regression

Ordinal Logistic Regression (OLR) is a statistical analysis technique employed to model the relationship between an ordinal DV and one or more IVs with either a continuous or categorical nature (Agresti, 2002). OLR is an extension of LogR where the log odds of the DV, assuming *k* levels, are linearly related to the IVs. One of the assumptions of the OLR is that of proportional odds, which entails that the effect of an IV is constant for each increase in the level of the DV (Agresti, 2002). The OLR was coded in RStudio 2024.04.2, details of which have been showcased in Appendix G. The IVs were added stepwise to the OLR model and with each formulation, the results outputs are tabulated in Table 13. The effects of each IV on the DV – Overall Satisfaction are illustrated in Fig. 32, while the coefficients for each IV for Model 7 are tabulated in Table 14.

Table 13: Results for OLR models with Ordinal DV (Good, Neutral, Poor)

Model#	IV	DV Rating- Good Prob.	DV Rating- Neutral Prob.	DV Rating- Poor Prob.	Accuracy
Model1	Region	46% in West	54% in Midwest	10% in Midwest	46%
Model2	Season	48% in Spring	52% in Winter	9% in Winter	48%
Model3	Temp. Range	44% in High Temp	50% in Medium Temp	8% in Medium Temp	48%

Model#	IV	DV Rating- Good Prob.	DV Rating- Neutral Prob.	DV Rating- Poor Prob.	Accuracy
Model4	Region and Season	50% in Spring & West	57% in Winter & Midwest	12% in Winter & Midwest	48%
Model5	Region and Temp. Range	48% in West & Low Temp.	55% in Midwest & Medium Temp.	11% in Midwest & Medium Temp.	45%
Model6	Season and Temp. Range	50% in Spring & Low Temp.	53% in Winter & Medium Temp.	9% in Winter & Medium Temp	48%
Model7	Region, Season, *High Temp*.	51% in Spring & West	57% in Winter & Midwest	12% in Winter & Midwest	49%
	Region, Season, *Medium Temp*.	49% in Spring & West	58% in Winter & Midwest	13% in Winter & Midwest	49%
	Region, Season, *Low Temp.*	54% in Spring & West	56% in Winter & Midwest	11% in Winter & Midwest	49%

Table 14: Coefficients summary for OLR Model7

IV	Value	Std. Error	t-value	***p*-value**
Job Region-Northeast	-0.1523	0.1928	-0.7898	*0.4296*
Job Region-South	-0.4105	0.1628	-2.5216	*0.0117***
Job Region-West	-0.4747	0.1656	-2.8665	*0.0042****
Season-Spring	-0.1522	0.1547	-0.9836	*0.3253*
Season-Summer	0.0409	0.1402	0.2919	*0.7704*
Temperature levels-Low	-0.1082	0.1809	-0.5978	*0.5500*
Temperature levels-Medium	0.0906	0.1253	0.7234	*0.4694*
Good | Neutral	-0.5757	0.1798	-3.2021	*0.0014*
Neutral | Poor	2.2187	0.1954	11.3552	*0.0000*

*Significant codes: '****' p < 0.001 '***' p < 0.01 '**' p < 0.05 '*' p < 0.1*

The assumptions required for Ordinal LogR include a check for multicollinearity and proportional odds assumption. The check for multicollinearity is by calculating the VIF, which was less than 10; hence, it is not a concern, and the assumption is not violated. Brant test was conducted on Model 7 to find that the Omnibus test with p = 0.05 was borderline significant, suggesting a potential overall violation; however, none of the IVs show a significant violation, suggesting potential issues with the proportional odds assumption shall not be a problem for the model.

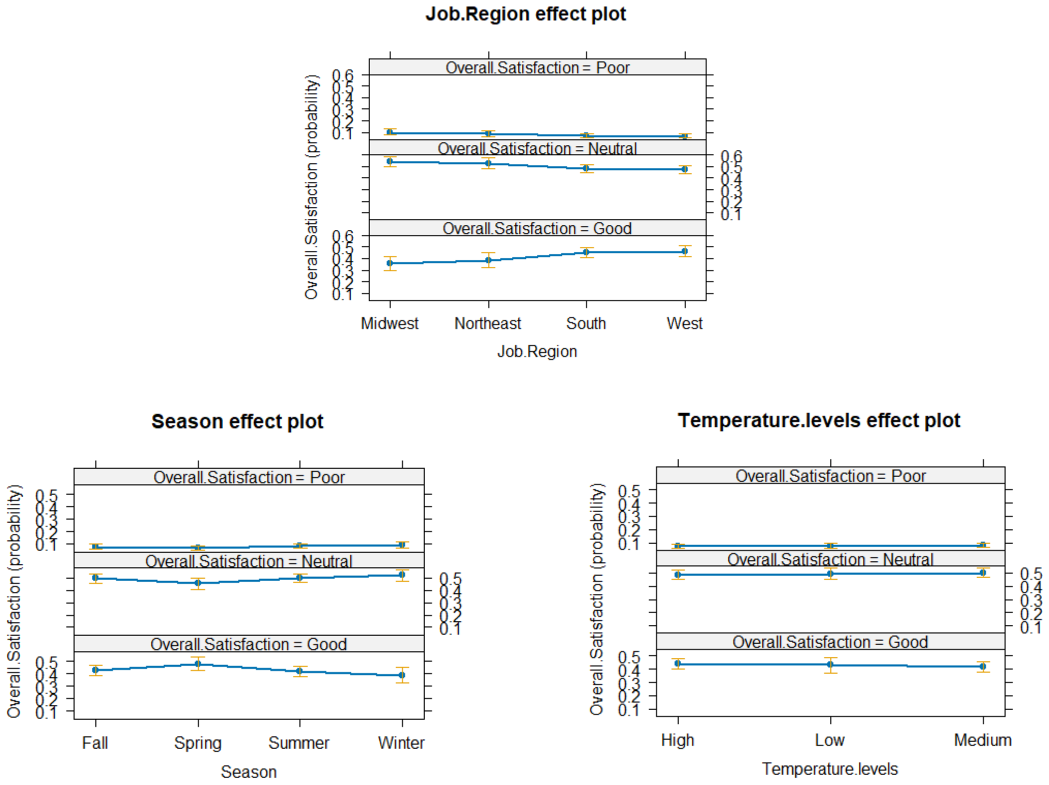


Figure 32: Effect of IVs on Overall Customer Satisfaction

Advantages: The OLR model allows the interpretation of datasets with ordered categorical DV and the IVs being either continuous or categorical. Normality in the distribution of IVs is not a requirement. However, the model assumes the relationship between each pair of DV groups is the same (unlike the Multinomial Logistic Regression, which does not preserve the ordered formation in the DV when returning the information on the contribution of each IV).

Limitations: A fundamental assumption of OLR is the proportional odds assumption, which means that the effect of an IV remains consistent across each level increase in the DV. OLR models can be parameterized in various ways, and different statistical software packages may use different parameterizations. Therefore, it is crucial to be cautious when interpreting the results from ordinal regression models (Agresti, 2002).

Conclusions: With region as IV in the OLR model, the overall accuracy of the model was 46%; with the only season as IV, the accuracy was 48%, and with only temperature range as the IV, the model's accuracy was 47.5%. The maximum accuracy was achieved with all the IVs in the model, which was 49% - which is, however, dismal.

Challenges in Dataset with ST Techniques

There were challenges when employing ST techniques for developing prediction models on the dataset for the study. The non-parametric distribution and non-linear relationship between the variables violated the assumption for conducting the ST techniques primarily. Even attempts at data transformation and pre-processing, like standardization or scaling, could not mitigate the concerns discussed in the limitations for each ST technique above.

Investigating the assumptions data needs to adhere to ST techniques of developing prediction models; the primary one is the normality of data distribution. For IVs of a categorical nature, univariate normality does not apply; hence, for conducting a check for assumptions, the frequency distributions can be examined. Furthermore, for the given study, the DV being ordinal, the categorical IVs were used to construct boxplot frequency distributions (Meyers et al., 2017b). For IVs continuous in nature, e.g., for the given dataset, “Average Temperature (°F)” can be investigated using histogram plots as well as “*q-q plot*” that maps the quantiles of theoretical distribution against the quantiles of the observed data (Fox & Weisberg, 2019). Fig. 33 is the q-q plot for Average Temperature (IV) and the histogram exhibiting almost normal distribution. Furthermore, in the q-q plot, the observed data almost follows the theoretical distribution with few extreme deviations, indicating that the variable is normally distributed.

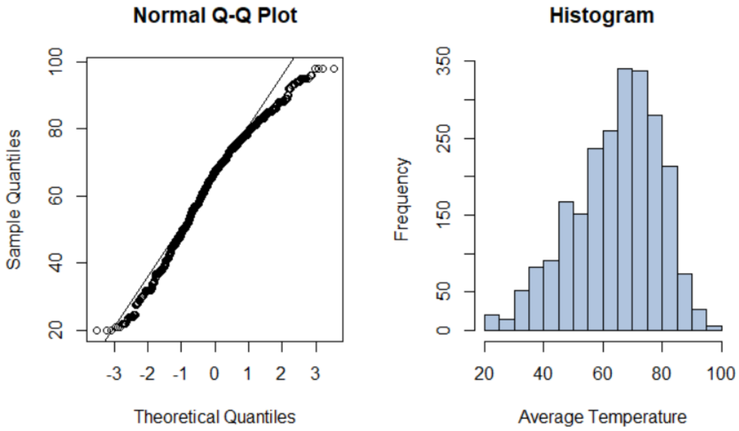


Figure 33: Q-Q plot and Histogram for Average Temperature (°F)

Investigating the DV, "Overall Customer Satisfaction," though ordinal, was treated as a continuous variable, assigning a numeric code for the outcome. Plotting the q-q plots and histogram for the DV, as illustrated in Fig. 34, depicts extreme deviations from the straight line in the q-q plot, exhibiting a distribution that is not normal and negatively skewed. Studies have considered that violation of normality of data distribution is not a significant concern when the sample size has 100 or more observations; however, for insightful conclusions, it should be followed (Lee, 2001; Mishra et al., 2019). This was followed by further confirmation using the Shapiro-Wilk normality test and the Anderson-Darling normality test (Fox & Weisberg, 2019; Mishra et al., 2019; Tabachnick & Fidell, 2013). Both tests were significant, with p < 0.001, confirming that the variable does not follow a normal distribution.

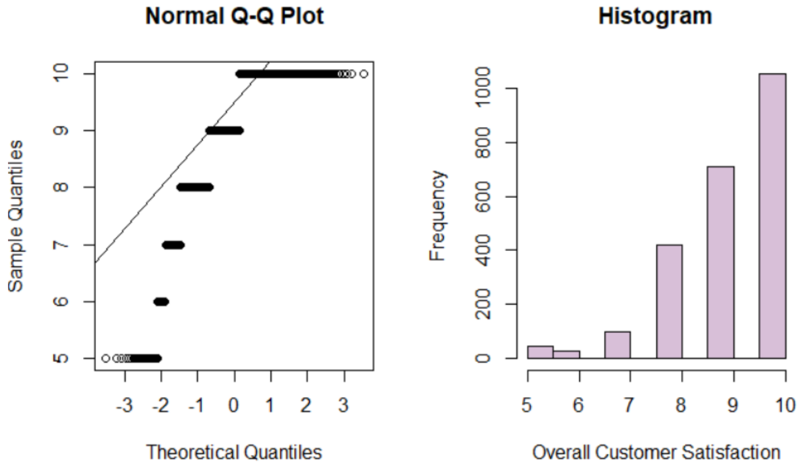


Figure 34: Q-Q Plot and Histogram for Overall Customer Satisfaction

The skewness and kurtosis of the data were investigated using the D'Agostino skewness test and the Anscombe-Glynn kurtosis test (Fox & Weisberg, 2019; Mishra et al., 2019). The results showed that the variable distribution has negative skewness with heavy-tailed distribution, with all results for the normality, skewness and kurtosis test detailed in Table 15.

Table 15: Normality, Skewness and Kurtosis Test Results

Skewness			**Kurtosis**			**Normality**		
Value	**p-value**	**Finding**	**Value**	**p-value**	**Finding**	**SW Test Value**[1]	**AD Test Value**[2]	**Finding**
-1.42	< 2.2e-16	Negative skew	5.30	< 2.2e-16	higher peak & heavy-tailed	0.78 p < 0.001	170.66 p < 0.001	Not normal distribution

[1] *SW Test – Shapiro-Wilk Normality Test*

[2] *AD Test – Anderson-Darling Normality Test*

If a variable is not approximately normal, a transformation could be helpful to meet the assumptions of statistical tests better since failure to meet statistical assumptions can result in decreased power and inflated Type I error (false positive) rates (Tabachnick & Fidell, 2013). Transformations were implemented on the variable (Tabachnick & Fidell, 2013), the results of which have been summarized in Table 16, reflecting the inability to create the desired outcome of normal distribution. Table 16 also includes the results recommended through the systemic approach of the Box-Cox family of transformations wherein, given a strictly positive variable, a power transformation λ can be determined that normalizes the DV (Daimon, 2011).

Table 16: Summary of transformations for Overall Customer Satisfaction

Transformation	**SW Test**[1]	**AD Test**[2]	**DA Test**[3]	**AG Test**[4]	**Conclusion**
subtract data from largest score (+1), then take logarithm	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p < 0.001$	Positively skewed with kurtosis; not normal distribution
subtract data from largest score (+1), then take reciprocal	$p < 0.001$	$p < 0.001$	$p = 0.997$	NA*	Skewness not a concern; not normal distribution
subtract data from largest score (+ 1), then take square root	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p = 0.2084$	Kurtosis not a concern; positively skewed, not normal distribution
Box-Cox Power: 5.25 (data raised to power)	$p < 0.001$	$p < 0.001$	$p < 0.001$	$p < 0.001$	Negatively skewed with kurtosis; not normal distribution

[1] *SW Test – Shapiro-Wilk Normality Test*

[2] *AD Test – Anderson-Darling Normality Test*

[3] *DA Test – D'Agostino Skewness Test*

[4] *Anscombe-Glynn Kurtosis Test*

** Value could not be computed for the test since the data could have become invalid*

Since all attempts to transform the DV to conform to a normal distribution were unsuccessful, the study was directed towards the modern, coming-of-age ML algorithms that did not require the data to follow the parametric distribution and could have complex relationships amongst the variables, not necessarily linear. It is critical to note that though the applied transformations did not yield satisfactory results, the study aimed to document the approach of implementing data transformations for a non-parametric dataset for customer satisfaction in the construction industry.

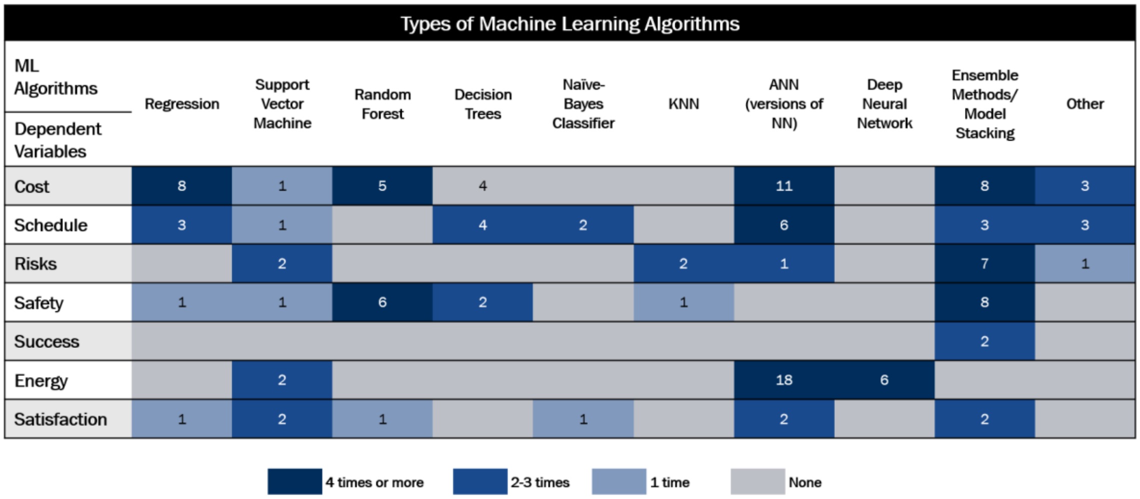
Machine Learning Prediction Modeling Techniques

ML Algorithms Selection

ML techniques have been proven to have an advantage over ST techniques in the ability to work with uncertainty, manage and perform with incomplete data, and unmistakably the ability to predict new cases based on learning from existing complex datasets (Gondia et al., 2019; Hashemi et al., 2020). ML algorithms do not assume that the data has been generated by any parametric model prescribed by the user; however, ST techniques would require formal model structures and data frequency distributions to comply with assumptions (Gondia et al., 2019). However, construction datasets are complex, with a rarity of tractable model structures and distributions adhering to assumptions, and ML models outperform traditional techniques (Poh et al., 2018). ST techniques could result in inaccurate representations of the actual phenomena due to the imposition of assumptions. In contrast, ML techniques are more effective in dealing with variables with linear or non-linear relationships and complex high-order relationships (Gondia et al., 2019).

During the literature review for studies implementing ML algorithms for prediction models, it was observed that neither any specific approach nor logical reasoning was provided for selecting a given ML algorithm. Therefore, in the current study, a framework was developed for selecting the ML algorithm in the given order, frequency of use of the ML technique, the goal of the study, i.e., the outcome being predicted, accuracy or performance of the technique and finally the complexity of nature of the relationship between the variables.

While developing classification prediction models, Decision Tree (DT), Naïve-Bayesian (NB) Classifier, Artificial Neural Network (ANN), Support Vector Machine (SVM) and Ensemble Method are employed (Li et al., 2018). Further, DT and NB have been used by researchers previously for small-sized datasets with a recorded history of satisfactory results (Gondia et al., 2019). In Fig. 35, the frequency of use of each category of ML technique was documented, which was employed to predict one of the listed DVs for a construction-specific scenario. Considering the study's dataset investigating the development of prediction models for customer satisfaction, the most used algorithms are Support Vector Machines and Artificial Neural Networks. Ensemble methods require selecting appropriate ML techniques followed by stacking, which was listed as a future path for the study; hence, the traditional Naïve-Bayes classifier was considered while forecasting customer satisfaction (Pandey et al., 2023).



Types of Machine Learning Algorithms

ML Algorithms / Dependent Variables	Regression	Support Vector Machine	Random Forest	Decision Trees	Naïve-Bayes Classifier	KNN	ANN (versions of NN)	Deep Neural Network	Ensemble Methods/ Model Stacking	Other
Cost	8	1	5	4			11		8	3
Schedule	3	1		4	2		6		3	3
Risks		2				2	1		7	1
Safety	1	1	6	2		1			8	
Success									2	
Energy		2					18	6		
Satisfaction	1	2	1		1		2		2	

■ 4 times or more ■ 2-3 times ■ 1 time ■ None

Figure 35: Frequency Table - ML Algorithms - Construction DVs

Studies have documented the performance of the NB Classifier for assessment and anticipation of time performance of projects (Gondia et al., 2019), for identification of main factors for construction delays (Kim et al., 2008), and for predicting customer satisfaction for an e-commerce dataset (Pandey et al., 2023). In Gondia et al., 2019, the NB classifier outperformed Decision Tree algorithms in both training and testing capabilities, along with more consistent predictions, across the three different categories of the DV.

A study predicting customer satisfaction for a retail dataset noted that SVM exhibited higher testing accuracy than the NB classifier, Random Forest and LogR models (Pandey et al., 2023). In another study by Li et al., 2018, ANN, SVM and XGBoost algorithms were used to predict customer satisfaction in the banking sector. These studies from other sectors were considered since not many studies predict customer satisfaction for the construction coatings sector or the construction industry. While examining SVMs predicting DVs from the construction industry, compared to ANNs, they offer several benefits, including efficient use of high-dimensional feature space, a uniquely solvable optimization problem, and the capability for theoretical analysis through computational learning theory (Wang et al., 2012).

Recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) were used to predict customer satisfaction for an e-commerce retail dataset that is categorized as a type of neural network that outperformed all other models (Pandey et al., 2023). Considering the utilization of ANN algorithms in the construction industry, predictive accuracy was highest for the estimation of project S-curves (Chao & Chien, 2009), highest while estimating construction costs (Gu, 2023), and excellent performance assessing the risk of delays in tall building projects (Sanni-Anibire et al., 2022). Another study tested multiple ANN-based paradigms to predict highway project delays and cost overrun percentages (El-Kholy, 2021).

Naïve-Bayes Classifier

This algorithm is transitioning from the traditional statistical quadrant to the advanced statistical techniques, which qualify as an early ML algorithm technique since the algorithm is based on Bayes' theorem to quantify the conditional probability of random variables. This algorithm aims to identify categories for new data input by calculating joint conditional probabilities of training dataset's IV given their DV classification (Gondia et al., 2019). The NB classifier requires an assumption for the conditional independence of the variable's values with respect to the class, which implies that a particular feature in a class is unrelated to the presence of any other feature (Kononenko & Kukar, 2007).

Furthermore, the NB classifier is more suited for small-size datasets since it converges more quickly and requires considerably less training data with excellent performance for real-world problems (Gondia et al., 2019; Kononenko & Kukar, 2007).

The NB classifier is derived from the Bayes rule (Kononenko & Kukar, 2007) as shown in Eq. (3), where P(C_k) are the prior probabilities of classes $C_{k,}$ k = 1,2,…, m_o, V = {v_1, …,v_a} is the vector of values of attributes describing an example, d(V) is the classifier mapping from example description V to the class, t(V) is the true class for an example described with V, P(V) is the prior probability of an example described with V, and P(V|C_k) is the conditional probability of an example described with V given the class C_k.

$$P(C_k \mid V) \;=\; P(C_k)\,\frac{P(V \mid C_k)}{P(V)} \tag{3}$$

Thus, the NB classifier's objective is to use learning examples to approximate both conditional and unconditional probabilities on the right-hand side of Eq. (4), written below.

$$P(C_k \mid V) \;=\; P(C_k)\prod_{i\,=\,1}^{a}\frac{P(C_k \mid v_i)}{P(C_k)} \tag{4}$$

<u>Data pre-processing:</u> The character IVs were converted to factors, and integer IVs were converted to numeric variables, as explained in the code in Appendix H. This was followed by splitting the dataset into training and testing datasets in the ratio of 70-30.

<u>Results for NB Classifier with Ordinal DV (5-10):</u> The NB Classifier was applied to the dataset of customer satisfaction (code in Appendix H), with outputs including A-priori probabilities referring to the probabilities of each class occurring in the dataset before any evidence (i.e., IVs) is considered. These probabilities are estimated directly from the

training data. In an NB classifier, conditional probabilities refer to the probabilities of observing a particular value of a feature given a specific class. These probabilities are estimated from the training data and are crucial for making predictions. Table 17 highlights the algorithm's output as class conditional probabilities for job region and season. Table 18 details the class conditional probabilities for temperature range levels. In both the tables, the IV with the maximum probability in the given class of DV has been marked in bold. Fig. 36 illustrates the CM heatmap for the NB classifier, enabling us to visually assess the algorithm's performance across multiple classes, which are customer satisfaction scores from 5 to 10 in the study's dataset. The accuracy of the model was 46.95%.

Table 17: Class Conditional Probabilities-Region & Season-Ordinal DV (5-10)

Class of DV	**Job Region**				**Season**			
	Midwest	**Northeast**	**South**	**West**	**Fall**	**Spring**	**Summer**	**Winter**
5	0.17	0.10	**0.50**	0.23	0.23	0.23	0.20	**0.33**
6	0.00	0.13	**0.63**	0.25	**0.44**	0.19	0.31	0.06
7	0.18	0.07	**0.42**	0.33	0.30	0.19	**0.37**	0.14
8	0.21	0.18	0.28	**0.32**	**0.35**	0.14	**0.35**	0.16
9	0.15	0.16	**0.41**	0.29	0.31	0.21	**0.34**	0.14
10	0.10	0.12	**0.45**	0.33	**0.32**	0.22	**0.32**	0.14

Table 18: Class Conditional Probabilities-Temperature Range-Ordinal DV (5-10)

Class of DV	**20-30°F**	**31-40°F**	**41-50°F**	**51-60°F**	**61-70°F**	**71-80°F**	**81-90°F**	**91-100°F**
5	0.03	0.07	0.17	0.13	0.13	**0.33**	0.13	0.00
6	0.00	0.06	0.06	0.13	0.19	**0.44**	0.13	0.00
7	0.00	0.01	0.12	0.16	0.23	**0.32**	0.14	0.01
8	0.02	0.06	0.08	0.17	0.27	**0.28**	0.11	0.02
9	0.01	0.07	0.12	0.14	**0.28**	0.25	0.12	0.01
10	0.01	0.04	0.11	0.17	0.25	**0.27**	0.13	0.02

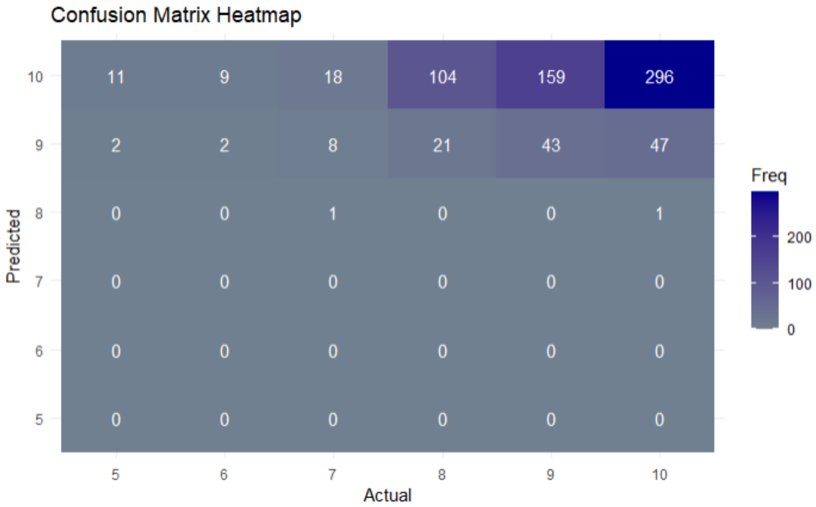


Figure 36: CM Heatmap-NB Classifier-Ordinal DV (5-10)

<u>Results for NB Classifier with Ordinal DV (0,1,2):</u> The NB Classifier was applied to the dataset of customer satisfaction (code in Appendix H), with outputs of conditional probabilities detailed in Tables 19 and 20. Fig. 37 visualizes the CM for the model as a heatmap with an accuracy of 49.45%.

Table 19: Class Conditional Probabilities-Region & Season-Ordinal DV (0,1,2)

Class of DV	**Job Region**				**Season**			
	Midwest	**Northeast**	**South**	**West**	**Fall**	**Spring**	**Summer**	**Winter**
Good (2)	0.10	0.12	**0.45**	0.33	**0.32**	0.22	**0.32**	0.14
Neutral (1)	0.16	0.17	**0.36**	0.30	0.32	0.19	**0.34**	0.15
Poor (0)	0.15	0.08	**0.47**	0.29	0.30	0.20	**0.32**	0.18

Table 20: Class Conditional Probabilities-Temperature Range-Ordinal DV (0,1,2)

Class of DV	**High (71-100°F)**	**Medium (51-70°F)**	**Low (20-50°F)**
Good (2)	0.41	**0.42**	0.17
Neutral (1)	0.39	**0.42**	0.19
Poor (0)	**0.48**	0.35	0.17

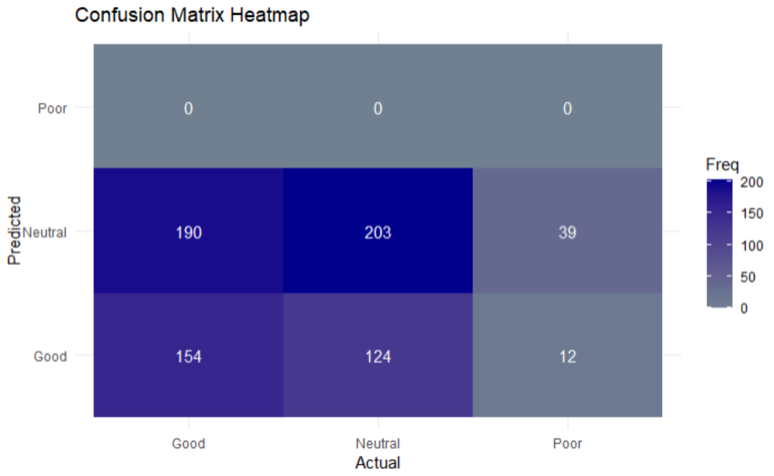


Figure 37: CM Heatmap-NB Classifier-Ordinal DV (0,1,2)

Results for NB Classifier with Binary DV (0,1): The NB Classifier was applied to the dataset of customer satisfaction (code in Appendix H), with outputs of conditional probabilities detailed in Tables 21 and 22. The model exhibited an accuracy of 92.94%, with 0 predictive power for "No," as shown in Fig. 38, the heatmap of the CM for the model.

Table 21: Class Conditional Probabilities-Region & Season-Binary DV (0,1)

Class of DV	Job Region				Season			
	Midwest	Northeast	South	West	Fall	Spring	Summer	Winter
No (0)	0.15	0.08	**0.47**	0.29	0.30	0.20	**0.32**	0.18
Yes (1)	0.14	0.14	**0.40**	0.32	0.32	0.20	**0.33**	0.14

Table 22: Class Conditional Probabilities-Temperature Range-Binary DV (0,1)

Class of DV	20-30°F	31-40°F	41-50°F	51-60°F	61-70°F	71-80°F	81-90°F	91-100°F
No (0)	0.01	0.03	0.13	0.15	0.20	**0.34**	0.13	0.01
Yes (1)	0.01	0.06	0.11	0.16	**0.26**	**0.26**	0.12	0.02

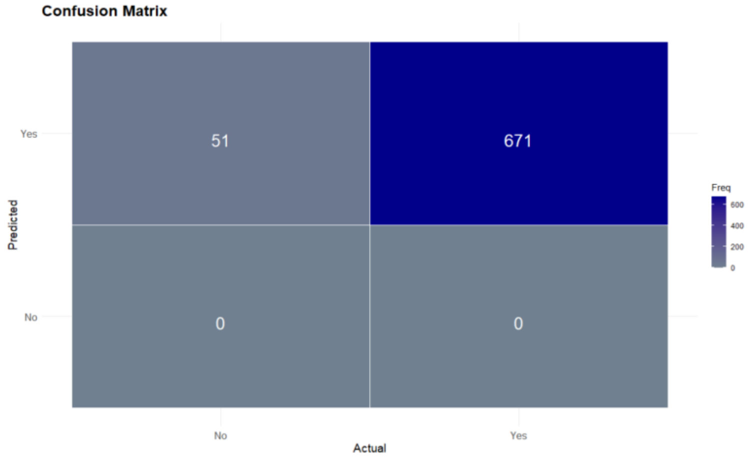


Figure 38: CM heatmap-NB Classifier-Binary DV (0,1)

Advantages: The NB classifier is more suited for small-sized datasets with quicker convergence and the requirement of less training data (Gondia et al., 2019). NB classifier can be applied to binary and ordinal DV scenarios and is widely regarded as a simple and fast-structured algorithm with high computational efficiency. Another advantage of the NB Classifier is that since it assumes the independence of variables, the entire covariance matrix is not required to estimate necessary parameters (Chen et al., 2021).

Limitations: NB classifier follows the laws of independent events' probability; thus, in scenarios of correlated variables, allocation of the increased weight of influence of the IVs on the DV results in a decline in the accuracy of prediction (Chen et al., 2021; Gondia et al., 2019).

Conclusions: NB classifier was employed on the dataset with three variations in the ordinal nature of the DV – range of 5-10, range of 0,1,2 and the binary nature of 0,1. The model's accuracy improved marginally with the change, such as the DV, from 5-10 to a three-tiered range of 0-2, i.e., 47% to 50%. The region with the highest probability of highest customer satisfaction was the South, and the seasons were Fall and Summer for both models. However, changing the DV to a dichotomous/binary nature increased the accuracy significantly, i.e., 93%, with none of the "No's" accurately predicted.

Support Vector Machine

Support Vector Machine (SVM) algorithms are one of the most widely used ML techniques for classification and regression tasks and are gaining popularity in construction research (Poh et al., 2018; Sanni-Anibire et al., 2022). This algorithm operates on vector algebra, placing an optimal class separating hyperplane in the space of original attributes of the dataset that forms the decision boundary to perform classification (Mansoor et al., 2024). In this multi-criterion optimization method, the formed boundaries utilize a selection of a small number of critical boundary instances called support vectors that are then used to build a linear discriminant function to separate them as widely as possible (Kononenko & Kukar, 2007; Sanni-Anibire et al., 2022). However, certain classification tasks may not be linearly separable; the instance-based approach of SVM using "kernel function" transforms the IVs into high dimensional feature spaces, allowing the formation of quadratic, cubic and higher-order decision boundaries (Kononenko & Kukar, 2007; Sanni-Anibire et al., 2022).

The development of the SVM algorithm is training with a labeled training dataset *X* of *n* examples – each consisting of a pair, an input vector x_i and the associated label y_i as shown in Eq. (5).

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \tag{5}$$

i.e., $X = \{x_i , y_i\}_{i=1}^{n}$ where x ∈ R^d and yi ∈ (+1, -1). Considering the case of two-dimensional input for visualization, there can be infinite hyperplanes separating the input dataset. The *"optimal separation hyperplane"* decision level that separates the input space is defined by the location with maximum margin (Cervantes et al., 2020) as illustrated in Fig. 39 and defined by Eq. (6).

$$w^T x_i + b = 0 \tag{6}$$

Assuming the simplest case of SVM, with the geometric margin optimized with the linear classifier y_i = 1, Eq. (7) becomes the combined set of inequalities.

$$y_i \left(\langle w . x_i \rangle + b \right) \geq 1 \, \forall \, i \tag{7}$$

The geometric margin for x^+ y x^- is Eq. (8):

$$\gamma_i = \frac{1}{2} \left(\left\langle \frac{w}{||w||} . x^+ \right\rangle - \left\langle \frac{w}{||w||} . x^- \right\rangle \right) = \frac{1}{2\,||w||} \left[\langle \, w \, . x^+ \rangle - \langle \, w \, . x^- \rangle \right] = \frac{1}{||w||} \tag{8}$$

Here, *w* is the optimal separation hyperplane, and *b* is the bias. Maximizing the generalization ability will result in selecting the optimal separation hyperplane, with the distance between the hyperplane and the training data closest to the hyperplane defined as the *margin*. In Fig. 39, the two parallel hyperplanes, namely, H_1 and H_2, are defined with the solution of quadratic programming, optimizing the geometric margin and minimizing the norm of the vector weights. On maximizing the distance between H_1 and H_2, some data points over H1 and some over H2 are called the *support vectors* (Cervantes et al., 2020; Shalev-Shwartz & Ben-David, 2014). This small set of support vectors thus gives the SVM solution since they participate directly in the definition of the separation hyperplane, and removal or changing of other points without crossing the planes H_1 and H_2 will not modify in any way the generalization skill of the classifier. The objective of training an SVM model is to find the *w* and *b* so that the hyperplane separates the data and maximizes the margin. $\frac{1}{||w||^2}$.

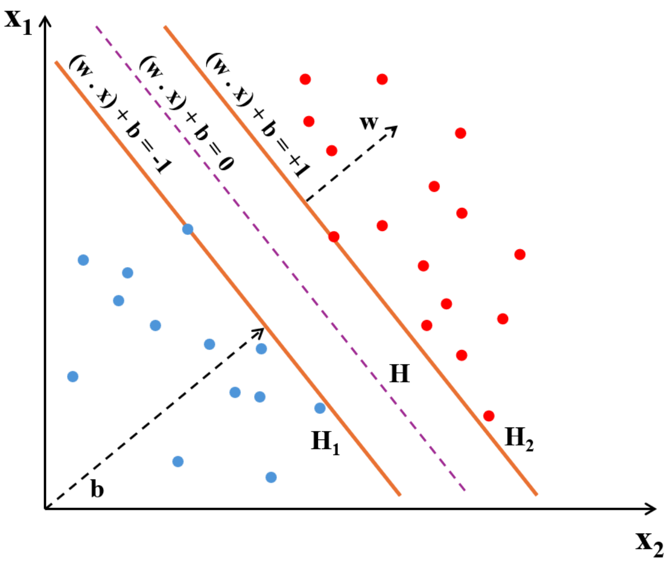


Figure 39: SVM Model with "Red" and "Blue" classification

<u>Data pre-processing:</u> One-hot encoding is one of the most popular coding techniques adopted for categorical variables wherein each level of the categorical IV is compared to a fixed reference level (Potdar et al., 2017). Assuming *x* as a discrete categorical random variable with *n* distinct values x_1, x_2, . . . x_n; then, the one-hot encoding of a particular value x_i is a vector *v* where every component of *v* is zero except for the ith component, which has the value 1 (Hancock & Khoshgoftaar, 2020). For example, variable x that takes values from the set S = {a, b, c}, with x_1 = a, x_2 = b, and x_3 = c, one-hot encoding for x will be (1, 0, 0), (0, 1, 0), and (0, 0, 1). This was followed by splitting the

dataset into training and testing datasets in the ratio of 70-30, as explained in the code detailed in Appendix I.

Results for SVM Classifier with Linear Kernel Function and Binary DV (Yes/No): The accuracy of the SVM classifier with the linear kernel function was 93%; however, the sensitivity of the classifier was 0, which entails that it could only accurately predict the "yes" and could not predict any "no." The CM heatmap is illustrated in Fig. 40.

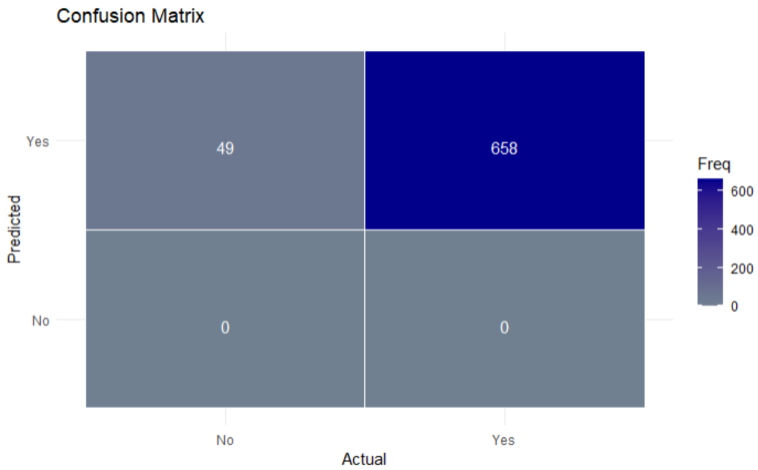


Figure 40: CM Heatmap-SVM Classifier - Linear, Radial and PCA

Results for SVM Classifier with Radial Kernel Function and Binary DV (Yes/No): The accuracy of the SVM classifier with the radial kernel function was also 93%, with the number of support vectors (254) higher just by one count than the count in the linear kernel (253) function result.

Results for SVM Classifier with Principal Components and Binary DV (Yes/No): The next option for training the SVM classifier was after performing PCA on the training data since the one-hot encoding technique had increased the dimensionality. In this method, too, the results for accuracy and sensitivity were similar to the previous two techniques, i.e., 93% with 0% sensitivity. In this method, the decision boundary was plotted for the training and testing datasets, as shown in Fig. 41 and Fig. 42.

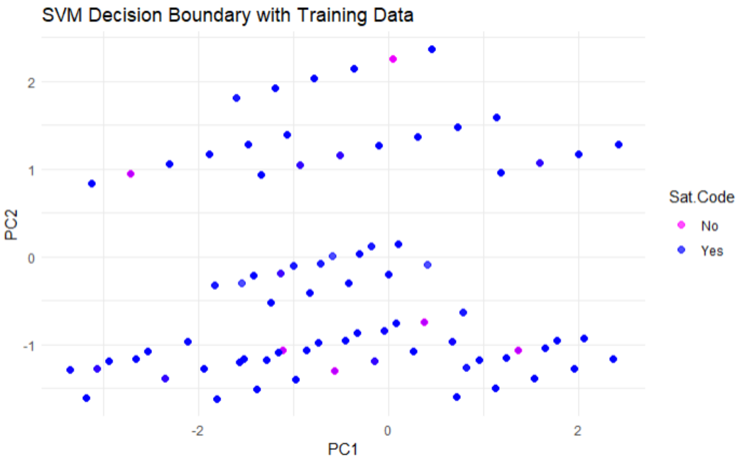


Figure 41: SVM Classifier for Training Dataset on PC1 and PC2

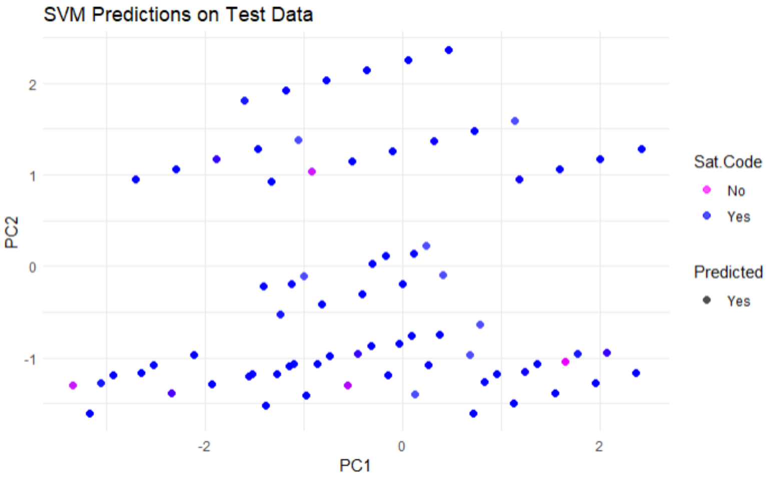


Figure 42: SVM Classifier for Testing Dataset on PC1 and PC2

Results for SVM Classifier with Radial Kernel and Ordinal DV (Good/Neutral/Poor): The SVM classifier was then applied to the dataset with the DV categorized into three (3) classes instead of two (2). The accuracy of this model was only

49% with the CM heatmap illustrated in Fig. 43. The code in RStudio 2024.04.2 has been detailed in Appendix I.

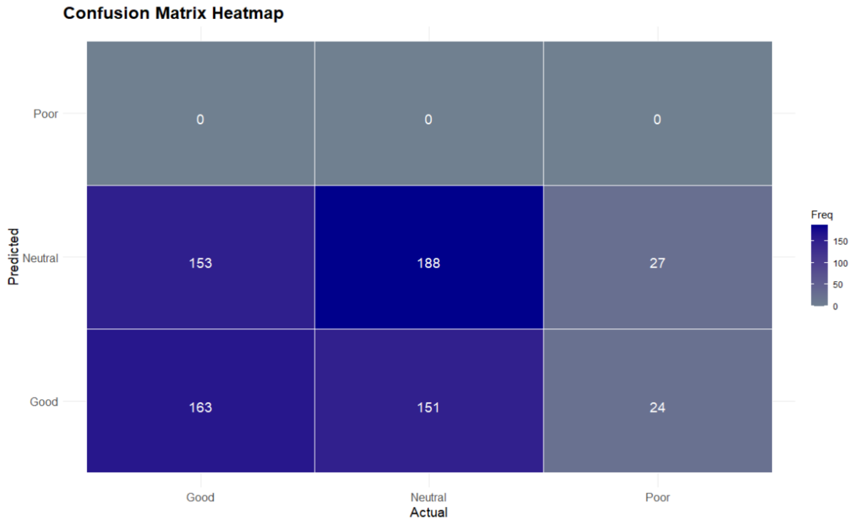


Figure 43: CM heatmap-SVM Classifier-Radial Kernel and Ordinal DV

Results for SVM Classifier with Balanced Dataset - Radial Kernel and Binary DV (Yes/No): It could be observed from data analysis conducted for previous classifiers that there was a strong inherent bias in the dataset with "Yes" datapoints count being 2185 and "No" datapoints being only 170. In this permutation, the imbalance was attempted to be addressed by randomly selecting cases so that a balanced dataset is generated to develop a more accurate classification model. The accuracy of this model was less than before, 47.57%, with Fig. 44 illustrating the CM heatmap for this model.

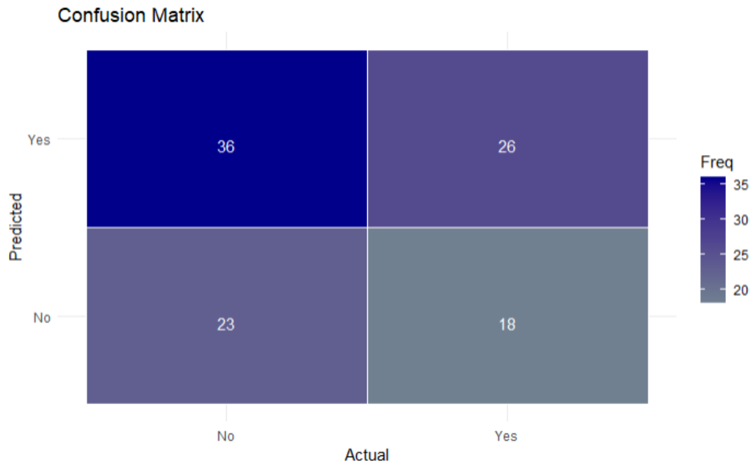


Figure 44: CM heatmap-SVM-Balanced Dataset-Radial Kernel & Binary DV

Results for SVM Classifier with Balanced Dataset - Linear Kernel and Binary DV (Yes/No): Training the SVM model on the balanced dataset with the linear kernel function generated a better accuracy of 50.49% with the CM heatmap for the model in Fig. 45.

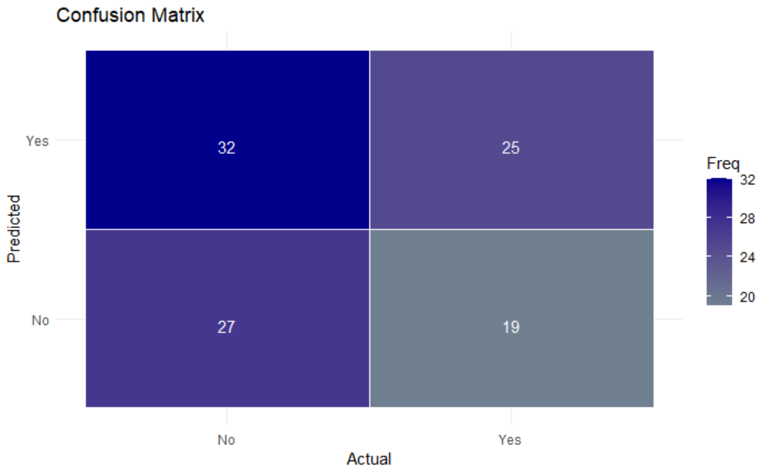


Figure 45: CM heatmap-SVM-Balanced Dataset-Linear Kernel & Binary DV

Advantages: SVM classifiers are computationally less expensive than other ML techniques and require less training data to learn patterns (Mansoor et al., 2024). For classification and regression tasks, separation is executed by selecting a small number of critical boundary instances – and this instance-based approach encourages the use of non-linear terms such as quadratic, cubic and higher-order decision boundaries, also referred to as the *kernel function*. This empowers the SVM to translate IVs into a higher feature space and work with complex relationships between IVs and DVs (Sanni-Anibire et al., 2022).

Limitations: The results for SVM classifiers lack transparency with sometimes high dimensionality in the dataset – it is a better classifier with binary DV than an ordinal nature DV. SVM lacks performance with datasets with overlapping DVs and higher noise, and there is no probabilistic explanation for the outcome/DV classification (Karamizadeh et al., 2014).

Conclusions: The accuracy of SVM classifier models with binary DV with linear and radial kernel and PCA was equal to 93% with no predictive power for “No’s” in all three scenarios. The DV was then transformed to an ordinal nature (0,1,2) to render a model with an accuracy of 49%; however, failing again to predict the lowest rating of DV – Poor. It was noted that the imbalance in the proportion of No’s to Yes’ could lead to overfitting. Thus, to mitigate the same, the dataset was transformed by random selection of the majority class to match the minority class count. Employing the SVM classifier on the balanced dataset resulted in the linear kernel function rendering a higher accuracy of 50% compared to the radial kernel function model’s accuracy of 48%.

Observing a decline in the accuracy of the model after utilizing a balanced dataset, class-imbalance treatment strategies were investigated, such as Sampling Minority Over-Sampling Technique (SMOTE) (Poh et al., 2018), Focal Loss, Weighted Random

Sampling (Shuang et al., 2024), and an increasing number of new minority samples (Zhang et al., 2020). Mitigating class imbalance is crucial to avoid the detrimental effect on the predictive performance of minority class labels in the dataset by the ML algorithm (Poh et al., 2018). Thus, it was noted that exploration of the performance of the model after implementing the strategies would be an insightful path forward for the study.

Artificial Neural Network

Artificial Neural Networks are computational algorithms that are designed to imitate the behavior of biological neural networks, with the basic unit being neurons (value being x) and axons (weight being *w*) (Li et al., 2018; Sanni-Anibire et al., 2022). ANNs are arranged in a multi-layer network with an adaptive system of interconnected neurons (connections between neurons termed *synapses*) capable of simulating complex non-linear relationships by performing multiple parallel computations (Sanni-Anibire et al., 2022). The functionality of ANNs is implemented as a weighted sum of input signals x_i, transformed with a threshold function *f* into an output signal $x_{out,}$ which may be active (1) or inactive (0 or -1) as written in Eq. (8) (Kononenko & Kukar, 2007) Fig. 46 further illustrates the ANN that computed the function defined in Eq. (8).

$$x_{out} = f\left(\sum_i w_i . x_i + w_{bias}\right) \tag{8}$$

The threshold function may be executed as a threshold function, as shown in Eq. (9), or more frequently as a sigmoid function in Eq. (10) because it is continuous and continuously derivable.

$$f(x) = \begin{cases} 1, & x > 0 \\ -1, & x \leq 0 \end{cases} \tag{9}$$

$$f(x) = \frac{1}{1 + e^{-X}} \tag{10}$$

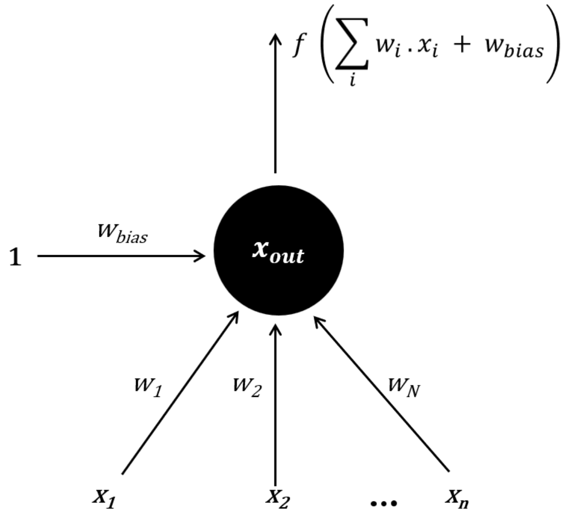


Figure 46: An artificial neuron with N number of incoming synapses

The primary objective of the ANN algorithm is to assign weights in a way that the neural network calculates the desired target function, with several hidden layers and hidden neurons determining the modeling competence of the algorithm (Kononenko & Kukar, 2007; Shalev-Shwartz & Ben-David, 2014).

Data Pre-Processing: IVs for region, season and temperature levels were coded numerically, and the average temperature for job installation was considered as a continuous IV. This was followed by standardizing all the numeric IVs to 0-1. The data was then split into training and testing datasets, with 75% of the data assigned for training the algorithm, with code details in Appendix J.

Results for ANN Models: The results for each of the permutations of six (6) ANN models are tabulated in Table 23 below. The model that showed the best output (Fig. 47) had 5 (five) hidden nodes with 1122 training steps and an R-squared value of 0.0089.

Table 23: Results for ANN Model Trials

Model #	**Nature of Temp. IV**	**# Hidden Nodes**	**# Steps for Training**	**RMSE (Root Mean Square Error)**	**R-squared Value**
Model1	Temp. Code	0	219	0.238	-0.011
Model2		3	11789	0.249	-0.110
Model3		***5***	***1122***	***0.236***	***0.009***
Model4	Avg. Temp.	0	308	0.239	-0.021
Model5		3	238	0.239	-0.019
Model6		5	33919	0.237	-0.006

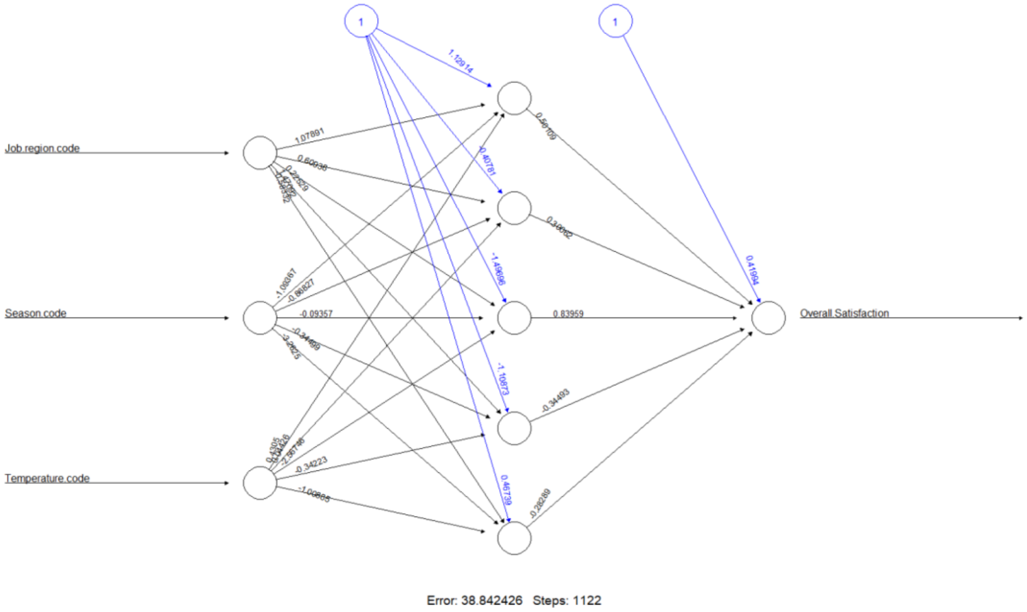


Figure 47: Model3 Output for ANN in RStudio

Advantages: ANN algorithms can produce results with limited formal statistical training and detect complex non-linear relationships and all possible interactions between IVs. The hidden nodes implicitly enable ANN algorithms to learn any nature of the relationship – complex and non-linear (Tu, 1996). It has been noted that a neural network can achieve superior performance with the appropriate arrangement and representation of training data and an optimal configuration (Chao & Chien, 2009).

Limitations: ANN is sensitive to a range of input values, so data normalization as a pre-processing step is necessary (Li et al., 2018). ANN cannot identify causal relationships explicitly and does not quickly determine which IVs are significant contributors to DV (Tu, 1996). Additionally, the training of ANN must account for the noise or randomness typically present in construction data to prevent overtraining, which would result in the neural network only recognizing training data and performing poorly during testing (Chao & Chien, 2009).

Conclusions: ANN was selected for the intended model because their structure of highly connected nodes with nonlinear transfer functions allows them to perform complex multi-attribute mapping, effectively handling inputs' combined and unknown effects on outputs (Chao & Chien, 2009). With six (6) trials, the final configuration of five (5) hidden nodes gave the maximum R-squared value and lowest RMSE.

Summary of Prediction Modeling Techniques Results

For a holistic understanding of the varied techniques to develop the norms on challenges and opportunities in the selected prediction approaches, it is essential to summarize the results for all techniques. Table 24 below delineates the ST and ML world results to draw well-rounded conclusions. The performance of ML models was assessed using performance metrics derived from their respective confusion matrices.

Table 24: Results for ST and ML Prediction Modeling Techniques

Nature of IV	Nature of DV	Assump. Check	Data Pre-Pro*	R^2 value	Accuracy /RMSE	Findings
Factor Analysis/Principal Component Analysis						
Continuous (Region & Season coded as 1,2,3,4, Avg. Temp)	Ordinal (5-10)	KMO-MSA = 0.516 Bartlett's test of sphericity; p < 0.001	NA	NA	NA	Squared Loadings: ***Region: 0.720*** Season: 0.541 Avg. Temp: 0.580
Linear Regression						
Categorical (Region, Season, & Temp. Levels – High, Low, Medium)	Ordinal (0,1,2)	Not normal, negatively skewed	-	0.0106	RMSE = 0.611	F (8, 2346) = 4.154 at p<0.001. South and West are significant at p < 0.001. Winter season is significant at p < 0.05.
Categorical (Region & Season)	Ordinal (5-10)	Not normal, negatively skewed	-	0.0110	RMSE = 1.081	F (6, 2348) = 5.382 at p<0.001. South and West regions are significant at p < 0.001. Winter season is significant at p < 0.01.
Categorical (Region, Season, Temp. Ranges)	Ordinal (5-10)		-	0.0133	RMSE = 1.078	F (13,2341) = 3.443 at p < 0.001. South and West regions are significant at p < 0.001. Winter season is significant at p < 0.01.
Discriminant Function Analysis						
Categorical (Region & Season, Temp. Range)	Binary (Yes/No)	Factors correlated, not normal.	-	-	NA	*Group Statistics and the Tests of Equality of Group Means – no significant difference.*
Continuous (Region & Season coded as			Split (80-20), Standardiza	-	92.78%	LD1 = 0.43 x Job Region Code – 0.9 x Season Code

Nature of IV	Nature of DV	Assump. Check	Data Pre-Pro*	R^2 value	Accuracy /RMSE	Findings
1,2,3,4, Avg. Temp)			tion of IVs (0-1)			+ 0.27 x Avg. Temp. *Overfitting with 0% Sensitivity.*
Logistic Regression						
Categorical (Region, Season, Temp. Ranges)	Binary (Yes/No)	Hosmer and Lemeshow Test of good fit, p = 0.902	-	0.014	92.8%	*Overfitting with 0% Specificity.*
Ordinal Logistic Regression						
Categorical (Region, Season, & Temp. Levels – High, Low, Medium)	Ordinal (0,1,2)	VIF < 10, Brant Test (*p*>0.05), not normal distribution	-	-	49%	Prob. of Good rating: 51% in Spring & West, High Temp. 49% in Spring & West, Medium Temp. 54% in Spring & West, Low Temp.
Naïve-Bayes Classifier						
Categorical (Region, Season, Temp. Ranges)	Ordinal (5-10)	Each class within IV is independent of each other.	Split (70-30)	NA	47%	Rating-10: 44% in South; 32% in Fall & Summer; 27% for 71-80°F
Categorical (Region, Season, Temp. Ranges)	Ordinal (0,1,2)				50%	Good Rating: 45% in South; 32% in Summer & Fall; 42% in Medium Temp. Level
Categorical (Region, Season, Temp. Ranges)	Binary (Yes/No)				93%	*Overfitting with 0% Sensitivity.*
Support Vector Machine – Linear Kernel Function						
Continuous (Region, Season, Temp. Range Codes)	Binary (Yes/No)	NA	One-hot encoding, Split (70-30)	NA	93%	*Overfitting with 0% Sensitivity.*
			Balanced DV, One-hot encoding, Split		50%	*Sensitivity 46% Specificity 57%*

Nature of IV	Nature of DV	Assump. Check	Data Pre-Pro*	R^2 value	Accuracy /RMSE	Findings
			(70-30)			
Support Vector Machine – PCA and Radial Kernel Function						
Continuous (Region, Season, Temp. Range Codes)	Binary (Yes/No)	NA	One-hot encoding, Split (70-30)	NA	93%	*Overfitting with 0% Sensitivity.*
Support Vector Machine – Radial Kernel Function						
Continuous (Region, Season, Temp. Range Codes)	Binary (Yes/No)	NA	One-hot encoding, Split (70-30)	NA	93%	*Overfitting with 0% Sensitivity.*
			Balanced DV, One-hot encoding, Split (70-30)		48%	*Sensitivity 39%* *Specificity 59%*
Continuous (Region, Season Codes, Avg. Temp.)	Ordinal (0,1,2)		Split (70-30)		49%	*Sensitivity for Good: 52%* *Sensitivity for Poor: 0%* *Specificity for Good: 55%* *Specificity for Poor: 100%*
Artificial Neural Network						
Continuous (Region, Season, Temp. Range Codes)	Ordinal (5-10)	NA	Split (75-25), Standardization of IVs (0-1)	0.009	RMSE = 0.236	*The strength of the relationship is not significant.*

** Data Pre-Pro: Data Pre-processing including "Training" and "Testing" Dataset Split*

CHAPTER SIX - CONCLUSIONS

In the study, the outcomes for both ST and ML prediction modeling techniques were measured and compared. The approach, when documented, supplied the researcher with specific parameters and solutions for improving the prediction accuracy and performance of the prediction model for a given dataset. While selecting a technique to develop a prediction model with, the dataset in hand and the goal of prediction are quintessentially the decision makers. The nature of the variables also plays an important role. However, the same could be transformed to meet the requirements of the technique being employed.

Conclusions for Research Objective 1

Research Objective 1: MEASURE the outcomes of the "customer satisfaction" (for the construction coatings sector) prediction model for both ST and ML approaches.

Research Question 1: What are the outcomes of ST and ML-based models for predicting customer satisfaction with a dataset having non-parametric distribution and limited dimensionality specific to the construction coatings sector?

For the ST approach, five techniques: FA/PCA, LR, DFA, LogR and Ording LogR were employed. In each ST technique, the dataset underwent multiple trials with different permutations and combinations of the IVs while predicting the DV. For FA/PCA, DFA, and LogR, only one model was tested for each approach with DV binary in nature while the IVs were continuous. In LR, there were six (6) models tested with ordinal DV (0,1,2) nature and three (3) models tested with ordinal DV (5-10); while for the Ordinal LogR method, seven (7) models were tested with ordinal DV (Good, Neutral, Poor). Furthermore, four (4) types of statistical transformations were carried out to mitigate the non-parametric nature and skewness in the dataset, which led the study to implement methods that could be employed on datasets with class imbalance. As a result of running these models, for the categorical/binary nature of DV, LogR and DFA models exhibited an accuracy of 93%. However, they had a 0% sensitivity towards the prediction of "No" – the minority class label. For the continuous nature of DV, the LR model with all IVs categorical and the ordinal nature of DV (5-10) had the maximum R-squared value of 0.0133. The reasons for the overfitting of the models could be attributed to the heavy negative skewness of the data. The study establishes the steps necessary to address negative skewness; however, it was identified that the limited dimensions (number of IVs) played a significant role in the predictive power.

For the ML approach, with the NB classifier, three models were tested with varying ordinal nature of DV, one with range 5-10 and another with range 0,1,2 and the final model was tested with binary nature of DV. While implementing the SVM classifier, six models were tested with different dataset transformations: linear and radial kernel function for binary DV (Yes/No) with imbalanced and balanced datasets, PCA and radial kernel function for binary DV (Yes/No), and radial kernel function for ordinal DV (0,1,2). Finally, ANN was conducted with six (6) trials, each with a varying number of hidden nodes – three (3) models were executed with Temperature range IV treated as a continuous IV numerically coded. In contrast, three (3) models were executed with Average Temperature used as the IV continuous in nature. Amongst all ML techniques, 93% accuracy was achieved with the NB classifier, IVs categorical in nature and binary DV, and SVM with linear and radial kernel functions as well PCA + radial kernel function, IVs continuous in nature and binary DV. In the ANN method, an R-squared value of 0.009 was achieved with IVs of continuous nature and DV ordinal nature coded from 5-10. The study established the nature of ML algorithms adopted based on the nature of the dataset. It also establishes the importance of the nature of variables, steps, and data transformations needed for various scenarios applicable to the current dataset. Class imbalance in the DV resulted in seemingly high overall accuracy but poor predictive power of the minority label. The study investigated one of the strategies to address the class imbalance; however, the effects were unsatisfactory, and other measures were listed but not investigated, considering the scope of the study.

Conclusions for Research Objective 2

Research Objective 2 - COMPARE ST vis-à-vis ML-based models predicting customer satisfaction in the construction coatings sector for a non-parametric dataset with limited dimensions.

Research Question 2 - What are the limitations and advantages of ST vis-à-vis ML approaches while predicting customer satisfaction in the construction coatings sector for a non-parametric dataset with limited dimensions?

FA/PCA are preferred techniques for dimensionality reduction, with no stringent requirement for multivariate normality; however, an infinite number of rotations are available, resulting in no readily available criteria against which the model can be tested. This method is not a technique that develops a prediction model but rather a technique that predicts the dataset's structure. LR, LogR, and Ordinal LogR methods have a strong advantage in the outcome of the equation of coefficients with each IV, unlike any other ST or ML technique. This enables the quantification of each IV's contribution (statistically significant or otherwise) towards change in the response or the DV in the model. The DFA technique is employed to predict group membership in naturally occurring groups, as opposed to groups created through random assignments, which raises the question of the reliability of such predictions. Additionally, classification tasks with DFA impose fewer statistical requirements - with an accuracy rate of around 95%, the shape of the distributions is not a significant concern.

NB classifier's advantage is in its fast structure and high computational efficiency. Since it assumes the independence of variables, the entire covariance matrix is not required to estimate necessary parameters, which is thus ideal for small-sized training datasets. However, if there is a correlation amongst the variables, allocating an increased weight of influence of the IVs on the DV could decline the prediction accuracy. With the SVM technique, the kernel function trick promotes the inclusion of non-linear terms like quadratic, cubic, and higher-order decision boundaries to handle complex relationships between IVs and DVs. However, SVM suffers in performance with datasets with overlapping DVs and high noise levels, further lacking transparency and not providing a probabilistic interpretation for the classification outcomes. Hidden nodes in ANN algorithms allow implicit learning of any relationship, whether it is complex or non-linear. ANN is sensitive to varying input values, necessitating data normalization as a preprocessing step. It cannot explicitly identify causal relationships and struggles to pinpoint which independent variables (IVs) significantly influence the dependent variable (DV). Training an ANN must consider data's inherent noise or randomness to avoid overfitting. While ANNs aim to minimize empirical errors, SVMs focus on minimizing the generalization error's upper bound, allowing SVMs to generalize effectively even with unseen data.

Conclusions for Research Objective 3

Research Objective 3 - DEVELOP a norm for handling non-parametric data with limited dimensions for the construction coatings sector.

Research Question 3 - Can a set of parameters, such as constraints on input data, variable relationships, and the goal of prediction, be leveraged to develop a norm for non-parametric datasets with limited predictors specific to customer satisfaction for the coating sector?

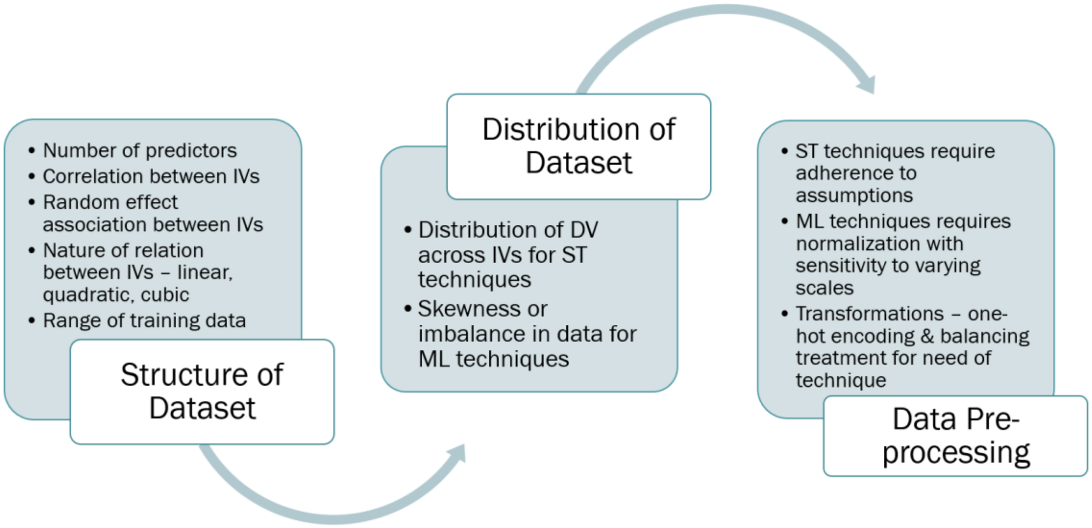
The observations from employing ST and ML techniques on the given dataset to predict the overall customer satisfaction for the construction coatings sector can be best examined through the lens of the following parameters.

Norms on ST and ML Approaches

Attributes of Input Data

Each ST and ML technique investigated in the study has led to a greater understanding of the data pre-processing strategies and treatments that must be adopted for the prediction model to perform efficiently and accurately. ML techniques certainly have the upper hand compared to the ST techniques, necessitating that the data comply with specific assumptions. However, the normality of the data distribution assumption could be relaxed considering the large sample size, the skewness of the data, and the imbalance in the DV, leading to issues like overfitting the models.

With ST techniques predicting the structure of a dataset like FA/PCA, it is ideally suited for a dataset with a high number (> 3) of IVs or predictors to determine IVs contributing significantly to the outcome and thus assist in the development of a prediction model after that only with the identified IVs. Multivariate normality was not a requirement for this method; correlation of factors was necessary. ST techniques predicting group membership, such as DFA and LogR, experienced the challenge of overlapping classes and boundaries that could not accurately predict the minority label of the DV. The fit for the model by the LogR method was not adequate and was corroborated by the visual interpretation of the classification of DV using DFA. LR and Ordinal LogR were the only ST techniques that exhibited statistical significance with the given set of IVs, along with the coefficients of estimates and proportional odds ratio, respectively. Amongst ST techniques, LR and Ordinal LogR successfully mitigate the class imbalance and generate results. The primary limitation of ML is the range of the training data, implying that an ML model is only applicable to scenarios covered by the training data. Splitting the given dataset into training and testing was mandatory for all the ML techniques to validate the model's performance. While implementing ML techniques, each required the dataset to be in a specific format; however, with the NB classifier, the dataset was not subjected to normalization, which was necessary for ANN, which is sensitive to a variation in the scale of all the variables. For SVM, one-hot encoding was performed for categorical variables to ensure more accurate predictive performance. Further, the dataset was balanced by random sampling of minority classes, which declined the SVM classifier's performance. For the ANN model, it was noted that increasing the number of IVs could help identify complex interactions, which could help the model perform better.


Structure of Dataset

- Number of predictors
- Correlation between IVs
- Random effect association between IVs
- Nature of relation between IVs – linear, quadratic, cubic
- Range of training data

Distribution of Dataset

- Distribution of DV across IVs for ST techniques
- Skewness or imbalance in data for ML techniques

Data Pre-processing

- ST techniques require adherence to assumptions
- ML techniques requires normalization with sensitivity to varying scales
- Transformations – one-hot encoding & balancing treatment for need of technique

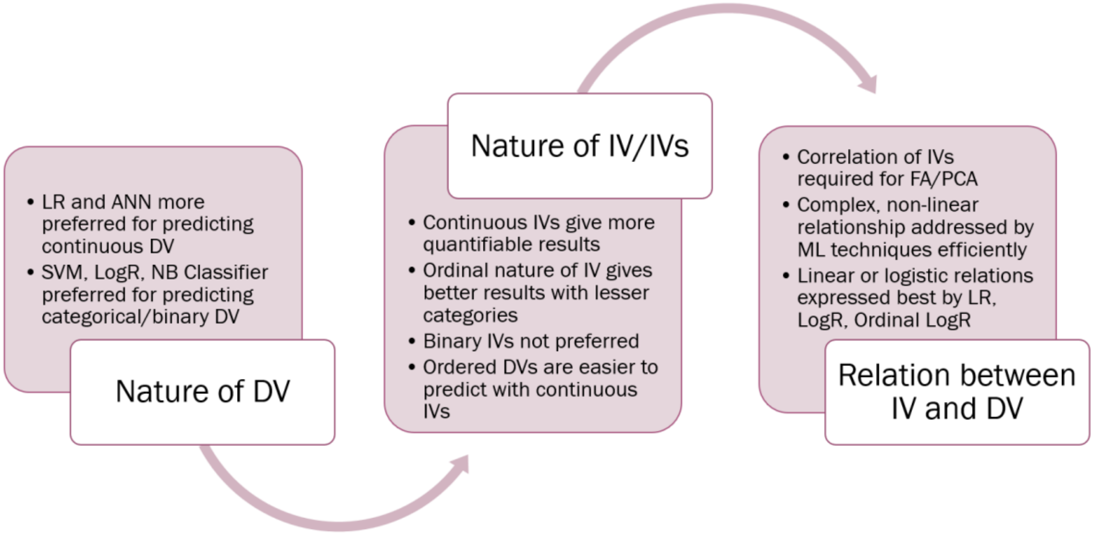
Figure 48: NORM 1-Attributes of Input Data

Fig. 48 illustrates the lessons learned documented for ST and ML approaches regarding the first norm–input data attributes. It can be observed that while selecting the ST or ML approach, it is critical to identify specific attributes of the input data being used for the prediction model that have been categorized as structure of the dataset, distribution of the dataset and pre-processing techniques that would be employed for the dataset for optimal performance of the model.

Nature of Variables

Observing the performance and interpretation of both ST and ML prediction models, it can be concluded that the preferred nature of IVs would be continuous. This nature of IVs allows for more quantifiable explanations and makes it easy to visualize and formulate statistical inferences. Since the given dataset consisted of two categorical IVs, transformation using one-hot encoding and numerical coding was implemented. Temperature was a more flexible IV since the dataset consisted of values of "Average Temperature (°F)" and Temperature range, which was ordinal. For techniques such as LR and ANN, the prediction of a DV with a continuous nature would be more suited to the mechanics of the method. Techniques were also tested for performance with the DV transformed to binary/dichotomous and ordinal (with three levels instead of six) nature.

Amongst ST techniques, the transformation of the IV- "Temperature range" from an ordinal variable with eight tiers to that of three tiers – High, Medium, and low- allowed for better interpretation of the models. Since this ranking could only be preserved with Ordinal LogR, other methods mandated that the DV be transformed to binary. The NB classifier performed better for ML techniques when DV was ordinal with only three tiers. The NB classifier and SVM models exhibited overfitting when the nature of DV was binary in contrast to the scenario when the DV was ordinal, or the dataset was balanced.



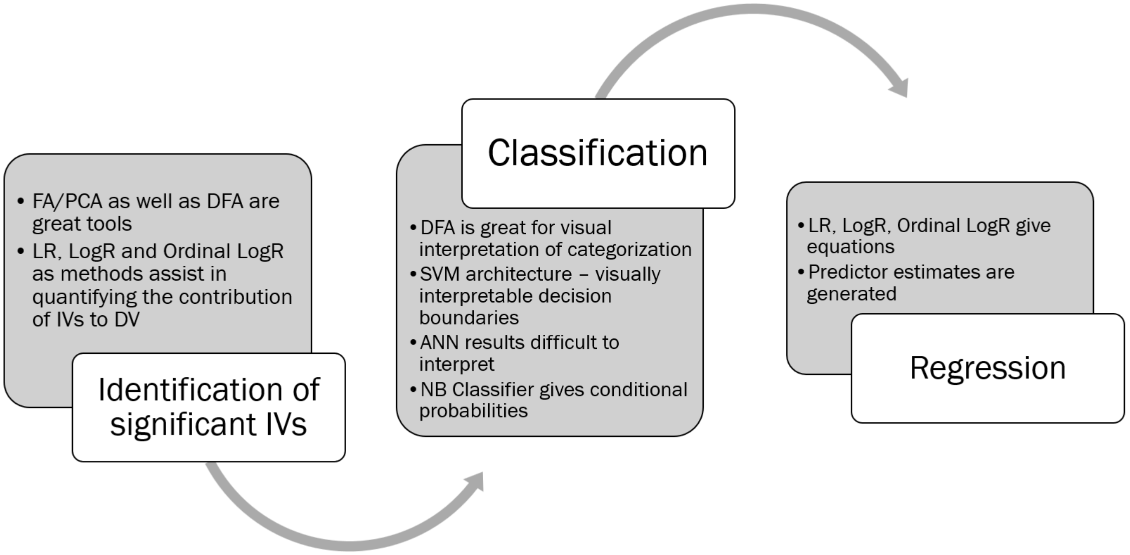
• LR and ANN more preferred for predicting continuous DV
• SVM, LogR, NB Classifier preferred for predicting categorical/binary DV
Nature of DV
Nature of IV/IVs
• Continuous IVs give more quantifiable results
• Ordinal nature of IV gives better results with lesser categories
• Binary IVs not preferred
• Ordered DVs are easier to predict with continuous IVs
• Correlation of IVs required for FA/PCA
• Complex, non-linear relationship addressed by ML techniques efficiently
• Linear or logistic relations expressed best by LR, LogR, Ordinal LogR
Relation between IV and DV

Figure 49: NORM 2-Nature of Variables

In conclusion, in Fig. 49, the norm for the nature of the variable has been classified into three critical criteria: nature of DV, nature of IV/IVs and the relation between IV and DV. For selecting the prediction model, it is essential to consider the nature of the variables – both input and outcome and the relation between them.

Goal of Prediction

The selection of the ST/ML technique for developing the prediction model is contingent upon the desired result and interpretability of the outcome. While FA/PCA is necessary for identifying factors that significantly contribute to the DV, DFA is an excellent tool that maximizes the separation between outcome categories – visual interpretation for both techniques is more accessible. LR, LogR and Ordinal LogR are regression equations that present the most significant advantage, i.e., the generation of predictor estimates (β) that account for the contribution of change in DV by the corresponding IV (for logistic regressions, it is the predicted change in logarithmic odds).



FA/PCA as well as DFA are great tools
LR, LogR and Ordinal LogR as methods assist in quantifying the contribution of IVs to DV
Identification of significant IVs
Classification
DFA is great for visual interpretation of categorization
SVM architecture – visually interpretable decision boundaries
ANN results difficult to interpret
NB Classifier gives conditional probabilities
LR, LogR, Ordinal LogR give equations
Predictor estimates are generated
Regression

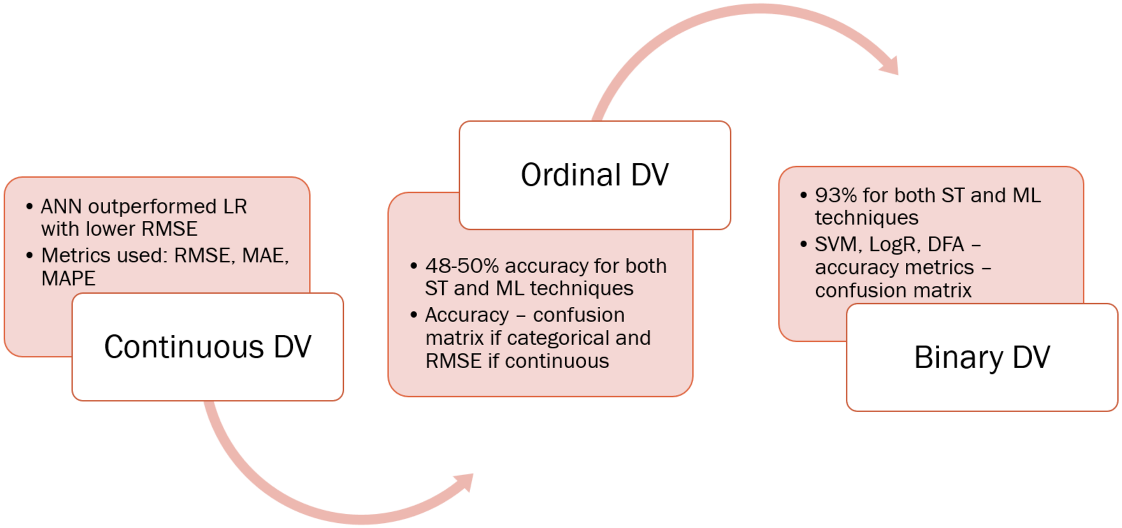
Figure 50: NORM 3-Goal of Prediction

NB Classifier supplies results in conditional probabilities for each group of IV, whereas SVM and ANN results are not explanatory. Both SVM and ANN generate a CM exhibiting the model's predictive power for each class label of DV across all IVs. The architecture of the SVM model is visually interpretable through decision boundaries plotted to classify the DV. In contrast, the ANN model structure consists of hidden nodes and input and output layers. Artificial Neural Networks (ANNs) are powerful tools, but their parameters lack specific physical meanings related to the dataset, making them black boxes that are difficult to interpret. Support Vector Machines (SVMs) are effective for classification, particularly for binary data, as they provide the optimal classifier by maximizing margins between classes. However, training SVMs is time-consuming, especially with large datasets.

In summary, it can be observed in Fig. 50 that the norm for the prediction goal has been divided into three main objectives for developing prediction models: identification of significant IVs, classification and regression. As discussed above, depending on the goal of the prediction model, an appropriate ST or ML technique can be selected.

Accuracy - Performance

Depending on the nature of the DV, the accuracy or performance of the model is evaluated. As illustrated in Fig. 51, the performance of the selected ST and ML techniques have been sorted according to the type of DV – continuous, ordinal and binary.



• ANN outperformed LR with lower RMSE
• Metrics used: RMSE, MAE, MAPE
Continuous DV
Ordinal DV
• 48-50% accuracy for both ST and ML techniques
• Accuracy – confusion matrix if categorical and RMSE if continuous
• 93% for both ST and ML techniques
• SVM, LogR, DFA – accuracy metrics – confusion matrix
Binary DV

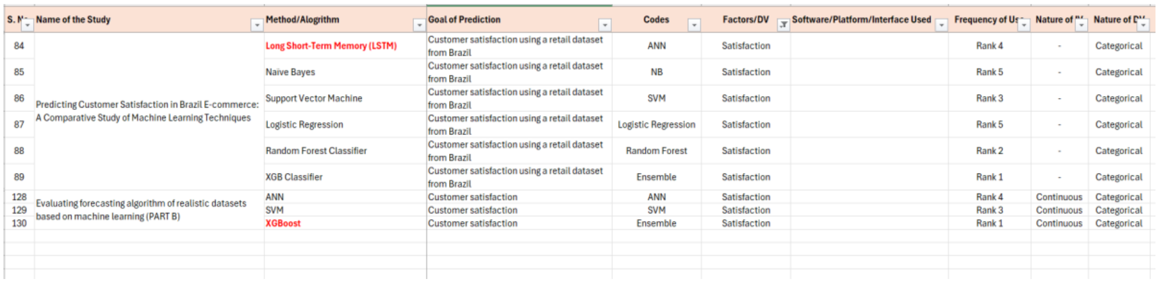
Figure 51: NORM 4-Performance and Accuracy

For the prediction of continuous outcome, ANN outperformed LR with a much lower RMSE of 0.236. Examining the prediction of the ordinal or binary outcome of the DV, it is interesting to note that the accuracy of models both ST and ML was 93% while predicting the binary outcome with overfitting and 0% predictive power for minority class "No." Nonetheless, the prediction of ordinal outcome resulted in accuracy ranging from 48% - 50% for Ordinal LogR and ML techniques.

Investigating the performance of algorithms applied to the dataset in the study, it was observed that the measure of accuracy depends on the type of the DV. It is essential to understand that each ML algorithm selected for the study was compared for its performance in predicting customer satisfaction. The literature review revealed that algorithms were selected in previous studies without any logical reasoning on a random basis. The current study thus developed an approach for selecting the ML technique that a future researcher can adopt to predict customer satisfaction in the construction industry with a dataset characterized by non-parametric distribution and a limited number of predictors. Moreover, with the implementation of the selected ST and ML techniques, it was noted that accuracy in prediction models is higher if the DV is binary or ordinal with fewer categories. Further, the performance of algorithms predicting a continuous DV is more quantifiable and easier to interpret and infer.

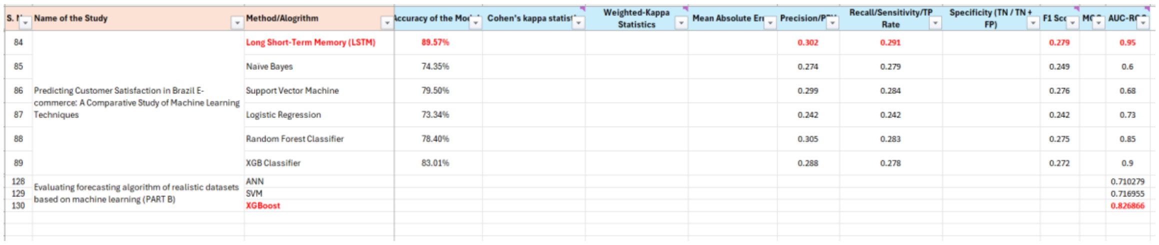
Norms - Repository

An Excel spreadsheet was populated to map different studies using ML algorithms to predict different DVs, documenting norms for each ML algorithm-based prediction model. The sheet tabulated “name of the study,” “method/algorithm” deployed, “goal of prediction” of the study, ranking of the ML algorithm per “frequency of use,” “nature of IV,” “nature of DV,” and finally the “performance metrics” used to evaluate the model. Considering a scenario – understanding optimal ML techniques for predicting customer satisfaction, the factors/DV column is filtered to “Satisfaction,” as shown in Fig. 52. This instantly lists the algorithms used to predict customer satisfaction and the nature of DV. Further, a column ranks each of the ML algorithms based on their frequency of use for the nature of DV.



S. N	Name of the Study	Method/Alogrithm	Goal of Prediction	Codes	Factors/DV	Software/Platform/Interface Used	Frequency of Us	Nature of I	Nature of D
84	Predicting Customer Satisfaction in Brazil E-commerce: A Comparative Study of Machine Learning Techniques	Long Short-Term Memory (LSTM)	Customer satisfaction using a retail dataset from Brazil	ANN	Satisfaction		Rank 4	-	Categorical
85		Naive Bayes	Customer satisfaction using a retail dataset from Brazil	NB	Satisfaction		Rank 5	-	Categorical
86		Support Vector Machine	Customer satisfaction using a retail dataset from Brazil	SVM	Satisfaction		Rank 3	-	Categorical
87		Logistic Regression	Customer satisfaction using a retail dataset from Brazil	Logistic Regression	Satisfaction		Rank 5	-	Categorical
88		Random Forest Classifier	Customer satisfaction using a retail dataset from Brazil	Random Forest	Satisfaction		Rank 2	-	Categorical
89		XGB Classifier	Customer satisfaction using a retail dataset from Brazil	Ensemble	Satisfaction		Rank 1	-	Categorical
128	Evaluating forecasting algorithm of realistic datasets based on machine learning (PART B)	ANN	Customer satisfaction	ANN	Satisfaction		Rank 4	Continuous	Categorical
129		SVM	Customer satisfaction	SVM	Satisfaction		Rank 3	Continuous	Categorical
130		XGBoost	Customer satisfaction	Ensemble	Satisfaction		Rank 1	Continuous	Categorical

Figure 52: Norms Sheet-Selection of DV-"Satisfaction"



S. N	Name of the Study	Method/Alogrithm	ccuracy of the Mo	Cohen's kappa statis	Weighted-Kappa Statistics	Mean Absolute Er	Precision/P	Recall/Sensitivity/TP Rate	Specificity (TN / TN + FP)	F1 Sc	MC	AUC-R
84	Predicting Customer Satisfaction in Brazil E-commerce: A Comparative Study of Machine Learning Techniques	Long Short-Term Memory (LSTM)	89.57%				0.302	0.291		0.279		0.95
85		Naive Bayes	74.35%				0.274	0.279		0.249		0.6
86		Support Vector Machine	79.50%				0.299	0.284		0.276		0.68
87		Logistic Regression	73.34%				0.242	0.242		0.242		0.73
88		Random Forest Classifier	78.40%				0.305	0.283		0.275		0.85
89		XGB Classifier	83.01%				0.288	0.278		0.272		0.9
128	Evaluating forecasting algorithm of realistic datasets based on machine learning (PART B)	ANN										0.710279
129		SVM										0.716955
130		XGBoost										0.826866

Figure 53: Norms Sheet-Performance metrics-ML techniques-“Satisfaction”

As shown in Fig. 52, it can be immediately interpreted that continuous IVs, ANN, SVM and XGBoost have been utilized to predict the categorical nature of DV. To understand the performance accuracy of the ML techniques for the given DV, Fig. 53 shows the screenshot of the metrics documenting the performance of each of the techniques. As observed, Long Short-Term Memory and XGBoost have exhibited the highest accuracy or closer to 1 value of AUC-ROC (area under the ROC curve plotted between TPR and FPR).

XGBoost is based on the gradient boosting algorithm, which sequentially builds an ensemble of decision trees. At the same time, Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to model sequences and time-series data (Li et al., 2018; Pandey et al., 2023). Both the ML techniques were considered outside the scope of the study and listed as future paths.

Limitations of the Study

- The study is limited to evaluating and comparing prediction models for the given dataset and parameters. Future studies will be required to investigate whether the norms developed can be extended to factors other than customer satisfaction in the construction coatings sector.
- The approach has been documented for a dataset exhibiting a non-parametric distribution that is negatively skewed.
- The number of input predictors or IVs in the dataset was limited - increasing the number of IVs could improve the model's predictability and increase strength in the relationship between the IVs and the DV.

- The construction industry suffers from a significant lack of recorded and published data suitable for ML applications. Consequently, the study relied on data primarily subjective and validated by subject matter experts.
- Overfitting makes ML much more difficult because a good performance on the training dataset does not mean a good performance on the test dataset.

Path Forward

Increasing number of predictors/IVs

Increasing the number of predictors or IVs in the dataset can increase the R-square value computed for a continuous DV scenario (Tabachnick & Fidell, 2013). This would also result in the improvement of the dimensionality of the dataset, becoming more coherent for prediction modeling since the performance of ML models is less influenced by the size of the dataset but depends on the interaction terms to perform well, which are present with a higher number of predictors (Bailly et al., 2022).

Mitigating Class Imbalance and Overfitting

Overfitting experienced during the implementation of ML techniques results in increasing difficulty for the development of prediction models since ensuring a good performance on the training dataset does not imply a good performance on the test dataset (Li et al., 2018). Hence, one of the future paths for the study is exploring different class-imbalance treatment strategies. There are two common strategies for balancing majority and minority classes: under-sampling and over-sampling. Unlike under-sampling, over-sampling avoids the risk of data loss (Shuang et al., 2024).

Sampling Minority Over-Sampling Technique (SMOTE): Instead of duplicating cases, this technique works in the n-dimensional feature space, generating synthetic features for the underrepresented class near its decision boundary using the K-nearest neighbor (KNN) algorithm (Poh et al., 2018). This approach has the advantage of assisting ML models to generalize more effectively.

Focal Loss (FL): The FL technique assigns greater attention to harder samples, which are difficult to discern, thus becoming liable to misclassification. A smaller number of hard samples can result in inadequate training of the ML algorithm, while increasing easy samples can also reduce the effectiveness of the training. FL strategizes by down-weighting the loss assigned to well-classified samples to perform better in highly imbalanced datasets (Shuang et al., 2024).

Weighted Random Sampling (WRS): WRS is an oversampling method – and with a put-back strategy, exposure of samples from the minority class of DV is enhanced during the training of the ML algorithm (Shuang et al., 2024). Three main components of this strategy include weights (probability of selection of each sample to be determined by inverse class frequency), replacement (minority class samples to be resampled to construct balance in the dataset) and generator (training samples are picked using the sampling indices with the WRS not being used in the validation and testing phase).

Contractor-Owner Information

On studying the results and understanding the dataset with its limitations, it could be prudent to include the contractor executing the job and the customer/owner for which it was executed in the dataset. Grouping cases from the same organization, i.e., projects executed by the same contractor for the same customer, could have an interaction or a relationship. This would result in predicting customer satisfaction utilizing multi-level regression, wherein random effect association from the same customer/same contractor independent of other jobs could contribute to predictive power. Moreover, the perspectives of both the customer and the contractor with the revised nested nature of data would account for the non-independence of the different jobs and accurate identification of significant contributors.

ML techniques – XGBoost and LSTM

The dataset can be utilized for prediction, implementing ML algorithms of XGBoost (Extreme Gradient Boosting) and LSTM (Long Short-Term Memory) as a future path to the study.

XGBoost employs an ensemble of decision trees to generate predictions. The architecture usually involves multiple decision trees, each trained on a weighted version of the dataset with varying depths. This approach helps to address problems caused by class imbalance (Alshboul et al., 2022; Li et al., 2018; Pandey et al., 2023). XGBoost trains faster than both ANN and SVM. Although XGBoost has numerous parameters, which can be both an advantage and a disadvantage, many of these parameters serve similar functions, such as preventing overfitting. Fortunately, the shorter training time of XGBoost allows for quicker parameter adjustments.

Recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) can identify long-term dependencies in sequential data. The name "long short-term memory" is abbreviated as LSTM. A typical architecture includes an embedding layer for transforming categorical data into continuous representations, LSTM layers with varying numbers of units, and a dense output layer for predictions (Pandey et al., 2023).

APPENDICES

Appendix A - Code in RStudio - Factor Analysis/Principal Component Analysis

```
#Conducting PCA test

PCA_Test <- eigen(cor(pca_manu_new))
PCA_Test
## eigen() decomposition
## $values
## [1] 1.1483684 0.9737754 0.8778562
##
## $vectors
##            [,1]        [,2]       [,3]
## [1,] -0.6718418  0.02101168  0.7403966
## [2,] -0.5052741 -0.74390738 -0.4373784
## [3,] -0.5415965  0.66795234 -0.5104047
variance_pc1 <- (1.1483684/4)*100
variance_pc1
## [1] 28.70921

#Principal Components Analysis

PCA_Test1 <- principal(pca_manu_new, nfactors = 3, rotate = "none",
                       scores = TRUE)
PCA_Test1
## Call: principal(r = pca_manu_new, nfactors = 3, rotate = "none", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  PC1   PC2   PC3 h2       u2 com
## Job.region.code 0.72 -0.02 -0.69  1  2.2e-16 2.0
## Season.code     0.54  0.73  0.41  1 -2.2e-16 2.5
## Avg.temp...F.   0.58 -0.66  0.48  1  2.2e-16 2.8
##
##                        PC1  PC2  PC3
## SS loadings           1.15 0.97 0.88
## Proportion Var        0.38 0.32 0.29
## Cumulative Var        0.38 0.71 1.00
## Proportion Explained  0.38 0.32 0.29
## Cumulative Proportion 0.38 0.71 1.00
##
## Mean item complexity =  2.4
## Test of the hypothesis that three components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi-square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

Appendix B - Code in RStudio – Linear Regression with Ordinal DV (0, 1, 2)

Model1: Region (Continuous – Categorical IV coded as 1,2,3,4)

```
regionfit <- lm(Overall.Satisfaction.Code ~ Job.region.code, data = manudatav1)
summary(regionfit)
summary(regionfit)$r.sq
## Call:
## lm(formula = Overall.Satisfaction.Code ~ Job.region.code, data = manudatav1)

## Residuals:
##    Min      1Q  Median      3Q     Max
##-1.4321 -0.3814 -0.2800  0.6186  0.7200

##Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
##(Intercept)      1.22929    0.03877  31.704  < 2e-16 ***
##Job.region.code  0.05070    0.01268   3.997 6.61e-05 ***
##---
##Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

##Residual standard error: 0.6138 on 2353 degrees of freedom
##Multiple R-squared:  0.006744,          Adjusted R-squared:  0.006322
##F-statistic: 15.98 on 1 and 2353 DF,  p-value: 6.61e-05

##[1] 0.006744148
```

Model2: Region (Categorical IV)

```
regionfit_cat <- lm(Overall.Satisfaction.Code ~ Job.Region, data = manudatav1)
summary(regionfit_cat)
summary(regionfit_cat)$r.sq
regionfit_cat$terms
## Call:
## lm(formula = Overall.Satisfaction.Code ~ Job.Region, data = manudatav1)

## Residuals:
##    Min      1Q  Median      3Q     Max
##-1.4205 -0.4205 -0.2545  0.5795  0.7455

##Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
##(Intercept)          1.25449    0.03355  37.394  < 2e-16 ***
##Job.RegionNortheast  0.05541    0.04823   1.149 0.250736
##Job.RegionSouth      0.16598    0.03881   4.276 1.98e-05 ***
```

```
##Job.RegionWest       0.14495    0.04058   3.572 0.000361 ***
##---
##Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

##Residual standard error: 0.6131 on 2351 degrees of freedom
##Multiple R-squared:  0.009689,           Adjusted R-squared:  0.008425
##F-statistic: 7.667 on 3 and 2351 DF,  p-value: 4.249e-05
```

Model3: Region and Season (Categorical IVs coded as 1,2,3,4)

```
regionseasonfit <- lm(Overall.Satisfaction.Code ~ Job.region.code + Season.code, data =
manudatav1)
summary(regionseasonfit)
summary(regionseasonfit)$r.sq
##Call:
##lm(formula = Overall.Satisfaction.Code ~ Job.region.code + Season.code,
    data = manudatav1)

##Residuals:
##    Min      1Q  Median      3Q     Max
##-1.4528 -0.4007 -0.2964  0.6139  0.7473

##Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
##(Intercept)      1.25888    0.04560  27.607  < 2e-16 ***
##Job.region.code  0.05212    0.01274   4.093 4.41e-05 ***
##Season.code     -0.01458    0.01183  -1.233    0.218
##---
##Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

##Residual standard error: 0.6137 on 2352 degrees of freedom
##Multiple R-squared:  0.007385,          Adjusted R-squared:  0.006541
##F-statistic:  8.75 on 2 and 2352 DF,  p-value: 0.0001637
```

ANOVA test between Model1 and Model3

```
summary(regionseasonfit)$r.sq - summary(regionfit)$r.sq
anova(regionfit, regionseasonfit)
##[1] 0.0006413373
##Analysis of Variance Table

##Model 1: Overall.Satisfaction.Code ~ Job.region.code
##Model 2: Overall.Satisfaction.Code ~ Job.region.code + Season.code
##  Res.Df    RSS Df Sum of Sq      F Pr(>F)
##1   2353 886.40
##2   2352 885.83  1   0.57234 1.5196 0.2178
```

Model4: Region and Season (Categorical IVs)

```
regionseasonfit_cat <- lm(Overall.Satisfaction.Code ~ Job.Region + Season, data =
manudatav1)
summary(regionseasonfit_cat)
summary(regionseasonfit_cat)$r.sq
##Call:
##lm(formula = Overall.Satisfaction.Code ~ Job.Region + Season,
##    data = manudatav1)

##Residuals:
##   Min      1Q  Median      3Q     Max
##-1.4608 -0.4047 -0.2573  0.5868  0.8165

##Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
##(Intercept)          1.254265   0.037154  33.759  < 2e-16 ***
##Job.RegionNortheast  0.059877   0.048266   1.241 0.214889
##Job.RegionSouth      0.167366   0.039673   4.219 2.55e-05 ***
##Job.RegionWest       0.147417   0.040784   3.615 0.000307 ***
##SeasonSpring         0.039195   0.036952   1.061 0.288930
##SeasonSummer         0.003063   0.031522   0.097 0.922600
##SeasonWinter        -0.070739   0.040106  -1.764 0.077899 .
##---
##Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

##Residual standard error: 0.6126 on 2348 degrees of freedom
##Multiple R-squared:  0.01249,Adjusted R-squared:  0.009966
##F-statistic: 4.949 on 6 and 2348 DF,  p-value: 4.817e-05
```

ANOVA test between Model2 and Model4

```
summary(regionseasonfit_cat)$r.sq - summary(regionfit_cat)$r.sq
anova(regionfit_cat, regionseasonfit_cat)
##[1] 0.002800629
##Analysis of Variance Table

##Model 1: Overall.Satisfaction.Code ~ Job.Region
##Model 2: Overall.Satisfaction.Code ~ Job.Region + Season
##  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
##1   2351 883.77
##2   2348 881.27  3    2.4993 2.2197 0.08389 .
##---
##Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Model5: Region, Season and Temperature Levels (Continuous IVs)

```
allvarfit <- lm(Overall.Satisfaction.Code ~ Job.region.code + Season.code + Temperature.Code,
data = manudatav1)
summary(allvarfit)
summary(allvarfit)$r.sq
##Call:
##lm(formula = Overall.Satisfaction.Code ~ Job.region.code + Season.code +
    Temperature.Code, data = manudatav1)

##Residuals:
##    Min      1Q  Median      3Q     Max
##-1.4539 -0.3972 -0.2968  0.6086  0.7423

##Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
##(Intercept)       1.263387   0.048943  25.814  < 2e-16 ***
##Job.region.code   0.052373   0.012777   4.099 4.29e-05 ***
##Season.code      -0.014526   0.011833  -1.228     0.22
##Temperature.Code -0.004411   0.017378  -0.254     0.80
##---
##Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##Residual standard error: 0.6138 on 2351 degrees of freedom
##Multiple R-squared:  0.007413,	Adjusted R-squared:  0.006146
##F-statistic: 5.852 on 3 and 2351 DF,  p-value: 0.0005587
```

ANOVA test between Model3 and Model5

```
summary(allvarfit)$r.sq - summary(regionseasonfit)$r.sq
anova(allvarfit, regionseasonfit)
##[1] 2.71985e-05
##Analysis of Variance Table

##Model 1: Overall.Satisfaction.Code ~ Job.region.code + Season.code + Temperature.Code
##Model 2: Overall.Satisfaction.Code ~ Job.region.code + Season.code
##  Res.Df    RSS Df Sum of Sq      F Pr(>F)
##1   2351 885.81
##2   2352 885.83 -1 -0.024272 0.0644 0.7997
```

Model6: Region, Season and Temperature Levels (Categorical IVs)

```
allvarfit_cat <- lm(Overall.Satisfaction.Code ~ Job.Region + Season + Temperature.levels, data
= manudatav1)
summary(allvarfit_cat)
summary(allvarfit_cat)$r.sq
```

```
##Call:
##lm(formula = Overall.Satisfaction.Code ~ Job.Region + Season +
##    Temperature.levels, data = manudatav1)

##Residuals:
##    Min      1Q  Median      3Q     Max
##-1.5227 -0.4217 -0.2424  0.5836  0.7980

##Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
##(Intercept)               1.21443    0.04441  27.346  < 2e-16 ***
##Job.RegionNortheast       0.05734    0.04835   1.186 0.235780
##Job.RegionSouth           0.17950    0.04019   4.467 8.33e-06 ***
##Job.RegionWest            0.15510    0.04121   3.764 0.000171 ***
##SeasonSpring              0.04684    0.03717   1.260 0.207714
##SeasonSummer              0.02772    0.03466   0.800 0.423906
##SeasonWinter             -0.09438    0.04209  -2.242 0.025055 *
##Temperature.levelsLow     0.08197    0.04372   1.875 0.060926 .
##Temperature.levelsMedium  0.02799    0.03058   0.915 0.360164
##---
##Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##Residual standard error: 0.6124 on 2346 degrees of freedom
##Multiple R-squared:  0.01397,Adjusted R-squared:  0.01061
##F-statistic: 4.154 on 8 and 2346 DF,  p-value: 6.052e-05
```

ANOVA test between Model4 and Model6

```
summary(allvarfit_cat)$r.sq - summary(regionseasonfit_cat)$r.sq
anova(allvarfit_cat, regionseasonfit_cat)
##[1] 0.001478599
##Analysis of Variance Table

##Model 1: Overall.Satisfaction.Code ~ Job.Region + Season + Temperature.levels
##Model 2: Overall.Satisfaction.Code ~ Job.Region + Season
##  Res.Df    RSS Df Sum of Sq      F Pr(>F)
##1   2346 879.96
##2   2348 881.27 -2   -1.3195 1.759 0.1724
```

Appendix C - Code in RStudio – Linear Regression with Ordinal DV (5 – 10)

Model1: Region (Categorical IV)

```
regionfit <- lm(Overall.Satisfaction ~ Job.Region, data = manudata1)
summary(regionfit)
## Call:
## lm(formula = Overall.Satisfaction ~ Job.Region, data = manudata1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1530 -0.1530  0.0032  0.8470  1.1198
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.88024    0.05938 149.555  < 2e-16 ***
## Job.RegionNortheast  0.11657    0.08537   1.365  0.17225
## Job.RegionSouth      0.27275    0.06869   3.971 7.39e-05 ***
## Job.RegionWest       0.22240    0.07183   3.096  0.00198 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.085 on 2351 degrees of freedom
## Multiple R-squared:  0.007569,   Adjusted R-squared:  0.006302
## F-statistic: 5.977 on 3 and 2351 DF,  p-value: 0.0004689
```

Model2: Region and Season (Categorical IV)

```
regionseasonfit <- lm(Overall.Satisfaction ~ Job.Region + Season, data = manudata1)
summary(regionseasonfit)
## Call:
## lm(formula = Overall.Satisfaction ~ Job.Region + Season, data = manudata1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2445 -0.2445  0.0819  0.8576  1.3121
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.87166    0.06565 135.129  < 2e-16 ***
## Job.RegionNortheast  0.12973    0.08529   1.521  0.12838
## Job.RegionSouth      0.28043    0.07010   4.000 6.53e-05 ***
## Job.RegionWest       0.23029    0.07207   3.195  0.00141 **
## SeasonSpring         0.09246    0.06530   1.416  0.15693
## SeasonSummer         0.03085    0.05570   0.554  0.57976
## SeasonWinter        -0.18381    0.07087  -2.594  0.00956 **
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.083 on 2348 degrees of freedom
## Multiple R-squared:  0.01357,    Adjusted R-squared:  0.01105
## F-statistic: 5.382 on 6 and 2348 DF,  p-value: 1.561e-05
```

ANOVA test between Model1 and Model2

```
summary(regionseasonfit)$r.sq - summary(regionfit)$r.sq
## [1] 0.005998202
anova(regionfit, regionseasonfit)
## Analysis of Variance Table
## Model 1: Overall.Satisfaction ~ Job.Region
## Model 2: Overall.Satisfaction ~ Job.Region + Season
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1   2351 2768.5
## 2   2348 2751.8  3    16.733 4.7592 0.002599 **
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model3: Region, Season and Temperature Levels (Categorical IVs)

```
allvarfit <- lm(Overall.Satisfaction ~ Job.Region + Season +
                    Temperature.range, data = manudata1)
summary(allvarfit)
## Call:
## lm(formula = Overall.Satisfaction ~ Job.Region + Season + Temperature.range,
##     data = manudata1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3202 -0.4092  0.1004  0.8598  1.3388
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             9.04154    0.20016  45.171  < 2e-16 ***
## Job.RegionNortheast      0.11899    0.08544   1.393  0.16383
## Job.RegionSouth         0.30404    0.07293   4.169 3.17e-05 ***
## Job.RegionWest          0.24620    0.07326   3.361  0.00079 ***
## SeasonSpring            0.11730    0.06621   1.771  0.07661 .
## SeasonSummer            0.07426    0.06344   1.171  0.24190
## SeasonWinter           -0.23835    0.07618  -3.129  0.00178 **
## Temperature.range31-40  -0.02306    0.20999  -0.110  0.91256
## Temperature.range41-50  -0.08436    0.19872  -0.425  0.67122
## Temperature.range51-60  -0.26095    0.19846  -1.315  0.18868
## Temperature.range61-70  -0.14267    0.19840  -0.719  0.47213
## Temperature.range71-80  -0.27967    0.20061  -1.394  0.16342
## Temperature.range81-90  -0.24420    0.21233  -1.150  0.25022
## Temperature.range91-100 -0.09002    0.27477  -0.328  0.74323
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.081 on 2341 degrees of freedom
## Multiple R-squared:  0.01876,    Adjusted R-squared:  0.01331
## F-statistic: 3.443 on 13 and 2341 DF,  p-value: 2.597e-05
```

ANOVA test between Model2 and Model3

```
summary(allvarfit)$r.sq - summary(regionseasonfit)$r.sq
## [1] 0.005192761
anova(regionseasonfit, allvarfit)
## Analysis of Variance Table
##
## Model 1: Overall.Satisfaction ~ Job.Region + Season
## Model 2: Overall.Satisfaction ~ Job.Region + Season + Temperature.range
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1   2348 2751.8
## 2   2341 2737.3  7    14.486 1.7698 0.08905 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Appendix D - SPSS – Discriminant Function Analysis

Table 25: DFA - Group Statistics Table

Overall Customer Satisfaction		Mean	Std. Deviation	Valid N (listwise)	
				Unweighted	Weighted
No	Region	2.85	1.042	170	170.000
	Season	2.37	1.166	170	170.000
	Temperature Range	5.87	1.474	170	170.000
Yes	Region	2.89	.994	2185	2185.000
	Season	2.31	1.066	2185	2185.000
	Temperature Range	5.94	1.496	2185	2185.000
Total	Region	2.89	.997	2355	2355.000
	Season	2.31	1.074	2355	2355.000
	Temperature Range	5.93	1.494	2355	2355.000

Table 26: DFA - Tests of Equality of Group Means

	Wilks' Lambda	F	df1	df2	Sig.
Region	1.000	.247	1	2353	.619
Season	1.000	.576	1	2353	.448
Temperature Range	1.000	.336	1	2353	.562

Appendix E - Code in RStudio – Discriminant Function Analysis

```
# Split the data into training (80%) and test set (20%)
set.seed(123)
training.samples <- manu_dfa$Overall.Satisfaction %>% createDataPartition(p = 0.8, list =
FALSE)
train.data <- manu_dfa[training.samples, ]
test.data <- manu_dfa[-training.samples, ]

# Estimate pre-processing parameters
preproc.param <- train.data %>%  preProcess(method = c("center", "scale"))

# Transform the data using the estimated parameters
train.transformed <- preproc.param %>% predict(train.data)
test.transformed <- preproc.param %>% predict(test.data)
str(train.transformed)

## 'data.frame':    1884 obs. of  4 variables:
##  $ Job.region.code     : num  0.126 0.126 0.126 -1.874 -1.874 ...
##  $ Season.code         : num  -1.22 -1.22 -1.22 -1.22 -1.22 ...
##  $ Avg.temp            : num  0.552 0.552 0.552 0.281 0.281 ...
##  $ Overall.Satisfaction: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 2 2 ...

str(test.transformed)

## 'data.frame':    471 obs. of  4 variables:
##  $ Job.region.code     : num  0.126 -1.874 -1.874 0.126 -1.874 ...
##  $ Season.code         : num  -1.22 -1.22 -1.22 -1.22 -1.22 ...
##  $ Avg.temp            : num  0.552 0.281 -0.666 0.146 1.229 ...
##  $ Overall.Satisfaction: Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 2 1 2 ...
```

Linear Discriminant Analysis

```
# Fit the model
model <- lda(Overall.Satisfaction ~., data = train.transformed)
model
## Call:
## lda(Overall.Satisfaction ~ ., data = train.transformed)
##
## Prior probabilities of groups:
##         No        Yes
## 0.07218684 0.92781316
##
## Group means:
##     Job.region.code Season.code     Avg.temp
## No      -0.05802448  0.13804724 -0.048367227
## Yes      0.00451449 -0.01074052  0.003763125
##
## Coefficients of linear discriminants:
##                         LD1
## Job.region.code  0.4280086
## Season.code     -0.9026584
## Avg.temp         0.2735021

# Make predictions
predictions <- model %>% predict(test.transformed)

# Model accuracy
mean(predictions$class==test.transformed$Overall.Satisfaction)
## [1] 0.9278132
```

Appendix F - SPSS – Logistic Regression

Table 27: LogR - Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	13.824	7	.054
	Block	13.824	7	.054
	Model	13.824	7	.054

Table 28: LogR - Hosmer and Lemeshow Test

	Chi-square	df	Sig.
1	3.462	8	.902

Table 29: LogR - Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	1207.287[a]	.006	.014

a: Estimation terminated at iteration number 6 because parameter estimates changed by less than .001.

Table 30: LogR - Variables in the Equation

		β	Std. Error	Wald	df	Sig.	Exp (β)	95% C.I for Exp (β)	
								Lower	Upper
Step 1[a]	TempRangeCode	-.078	.068	1.293	1	.255	.925	.810	1.058
	JobRegionCode			4.286	3	.232			
	JobRegionCode(1)	.652	.321	4.122	1	.042	1.920	1.023	3.602
	JobRegionCode(2)	.302	.241	1.570	1	.210	1.353	.843	2.171
	JobRegionCode(3)	.318	.245	1.676	1	.195	1.374	.849	2.222
	SeasonCode			9.731	3	.021			
	SeasonCode(1)	.336	.245	1.885	1	.170	1.400	.866	2.263
	SeasonCode(2)	.390	.226	2.974	1	.085	1.477	.948	2.302
	SeasonCode(3)	-.428	.241	3.144	1	.076	.652	.406	1.046
	Constant	2.608	.411	40.239	1	<.001	13.571		

Appendix G - Code in RStudio – Ordinal Logistic Regression

Dividing data into training and test sets

```
#Random Sampling
samplesize = 0.60*nrow(olsdat)
set.seed(100)
index = sample(seq_len(nrow(olsdat)), size = samplesize)

#Creating training and test set
datatrain = olsdat[index,]
datatest = olsdat[-index,]
```

Model1: Only Region as Predictor

```
library(MASS)
olsmodel_R = polr(Overall.Satisfaction ~ Job.Region, data = datatrain, Hess = TRUE)
summary(olsmodel_R)

## Call:
## polr(formula = Overall.Satisfaction ~ Job.Region, data = datatrain,
##     Hess = TRUE)
##
## Coefficients:
##                        Value Std. Error t value
## Job.RegionNortheast -0.1336     0.1917  -0.697
## Job.RegionSouth     -0.3972     0.1566  -2.536
## Job.RegionWest      -0.4424     0.1629  -2.716
##
## Intercepts:
##              Value   Std. Error t value
## Good|Neutral -0.5973  0.1353    -4.4134
## Neutral|Poor  2.1894  0.1552    14.1097
##
## Residual Deviance: 2555.765
## AIC: 2565.765

#Compute confusion table and misclassification error
predictcustsatR = predict(olsmodel_R,datatest)
table(datatest$Overall.Satisfaction, predictcustsatR)

##          predictcustsatR
##           Good Neutral Poor
##   Good       0     447    0
##   Neutral    0     433    0
##   Poor       0      62    0
```

```
mean(as.character(datatest$Overall.Satisfaction) != as.character(predictcustsatR))
## [1] 0.5403397

#Plotting the effects
library("effects")
effect(focal.predictors = "Job.Region", olsmodel_R)
## Job.Region effect (probability) for Good
## Job.Region
##   Midwest Northeast     South      West
## 0.3549580 0.3861127 0.4501366 0.4613576
##
## Job.Region effect (probability) for Neutral
## Job.Region
##   Midwest Northeast     South      West
## 0.5443353 0.5246539 0.4798572 0.4715249
##
## Job.Region effect (probability) for Poor
## Job.Region
##    Midwest  Northeast      South       West
## 0.10070670 0.08923343 0.07000618 0.06711748

# Compute confusion matrix
conf_matrixR <- confusionMatrix(data = predictcustsatR,
                  reference = datatest$Overall.Satisfaction)
# Extract performance metrics
accuracyR <- conf_matrixR$overall['Accuracy']
# Print performance metrics
print(paste("Accuracy:", accuracyR))
## [1] "Accuracy: 0.459660297239915"
```

Model2: Only Season as Predictor

```

library(MASS)
olsmodel_S = polr(Overall.Satisfaction ~ Season, data = datatrain, Hess = TRUE)
summary(olsmodel_S)
## Call:
## polr(formula = Overall.Satisfaction ~ Season, data = datatrain,
##     Hess = TRUE)
##
## Coefficients:
##                 Value Std. Error t value
## SeasonSpring -0.22519     0.1488 -1.5130
## SeasonSummer  0.02017     0.1261  0.1599
## SeasonWinter  0.15397     0.1668  0.9232
##
## Intercepts:
```

```
##              Value   Std. Error t value
## Good|Neutral -0.2970  0.0926    -3.2062
## Neutral|Poor  2.4816  0.1253    19.8054
##
## Residual Deviance: 2561.106
## AIC: 2571.106

#Compute confusion table and misclassification error
predictcustsatS = predict(olsmodel_S,datatest)
table(datatest$Overall.Satisfaction, predictcustsatS)
##          predictcustsatS
##           Good Neutral Poor
##   Good     103     344    0
##   Neutral   88     345    0
##   Poor      10      52    0
mean(as.character(datatest$Overall.Satisfaction) != as.character(predictcustsatS))
## [1] 0.5244161

#Plotting the effects
library("effects")
Effect(focal.predictors = "Season", olsmodel_S)
##
## Season effect (probability) for Good
## Season
##      Fall    Spring    Summer    Winter
## 0.4262796 0.4820432 0.4213546 0.3891194
##
## Season effect (probability) for Neutral
## Season
##      Fall    Spring    Summer    Winter
## 0.4965658 0.4553857 0.5000424 0.5220244
##
## Season effect (probability) for Poor
## Season
##       Fall     Spring     Summer     Winter
## 0.07715466 0.06257103 0.07860297 0.08885621

# Compute confusion matrix
conf_matrixS <- confusionMatrix(data = predictcustsatS, reference = datatest$Overall.Satisfacti
on)

# Extract performance metrics
accuracyS <- conf_matrixS$overall['Accuracy']

# Print performance metrics
print(paste("Accuracy:", accuracyS))
```

```
## [1] "Accuracy: 0.475583864118896"
```

Model3: Only Temperature Range as Predictor

```
library(MASS)
olsmodel_T = polr(Overall.Satisfaction ~ Temperature.levels, data = datatrain, Hess = TRUE)
summary(olsmodel_T)
## Call:
## polr(formula = Overall.Satisfaction ~ Temperature.levels, data = datatrain,
##     Hess = TRUE)
##
## Coefficients:
##                          Value Std. Error t value
## Temperature.levelsLow    0.04603     0.1464  0.3145
## Temperature.levelsMedium 0.08817     0.1139  0.7743
##
## Intercepts:
##              Value   Std. Error t value
## Good|Neutral -0.2353  0.0830    -2.8357
## Neutral|Poor  2.5380  0.1192    21.2980
##
## Residual Deviance: 2565.376
## AIC: 2573.376

#Compute confusion table and misclassification error

predictcustsatT = predict(olsmodel_T,datatest)
table(datatest$Overall.Satisfaction, predictcustsatT)
##          predictcustsatT
##           Good Neutral Poor
##   Good       0     447    0
##   Neutral    0     433    0
##   Poor       0      62    0
mean(as.character(datatest$Overall.Satisfaction) != as.character(predictcustsatT))
## [1] 0.5403397

#Plotting the effects
library("effects")
Effect(focal.predictors = "Temperature.levels", olsmodel_T)
## Temperature.levels effect (probability) for Good
## Temperature.levels
##      High       Low    Medium
## 0.4414515 0.4301334 0.4198360
## Temperature.levels effect (probability) for Neutral
## Temperature.levels
##      High       Low    Medium
## 0.4853121 0.4934437 0.5007132
##
## Temperature.levels effect (probability) for Poor
## Temperature.levels
##       High        Low     Medium
## 0.07323641 0.07642289 0.07945080

# Compute confusion matrix
conf_matrixT <- confusionMatrix(data = predictcustsatT, reference = datatest$Overall.Satisfacti
on)

# Extract performance metrics
accuracyT <- conf_matrixS$overall['Accuracy']

# Print performance metrics
print(paste("Accuracy:", accuracyT))
## [1] "Accuracy: 0.475583864118896"
```

Model4: Region and Season as Predictors

```

library(MASS)
olsmodelRS = polr(Overall.Satisfaction ~ Job.Region + Season, data = datatrain, Hess = TRUE)
summary(olsmodelRS)
## Call:
## polr(formula = Overall.Satisfaction ~ Job.Region + Season, data = datatrain,
##     Hess = TRUE)
##
## Coefficients:
##                        Value Std. Error t value
## Job.RegionNortheast -0.14127     0.1920 -0.7357
## Job.RegionSouth     -0.38963     0.1605 -2.4278
## Job.RegionWest      -0.44278     0.1638 -2.7025
## SeasonSpring        -0.14040     0.1538 -0.9129
## SeasonSummer         0.04913     0.1269  0.3871
## SeasonWinter         0.22423     0.1698  1.3209
##
## Intercepts:
##              Value   Std. Error t value
## Good|Neutral -0.5756  0.1502    -3.8312
## Neutral|Poor  2.2163  0.1687    13.1390
##
## Residual Deviance: 2551.619
## AIC: 2567.619
#Compute confusion table and misclassification error

```

```
predictcustsatRS = predict(olsmodelRS,datatest)
table(datatest$Overall.Satisfaction, predictcustsatRS)
##          predictcustsatRS
##           Good Neutral Poor
##   Good      92     355    0
##   Neutral   72     361    0
##   Poor       9      53    0

mean(as.character(datatest$Overall.Satisfaction) != as.character(predictcustsatRS))
## [1] 0.5191083

library("effects")
Effect(focal.predictors = c("Job.Region", "Season"), olsmodelRS)
##
## Job.Region*Season effect (probability) for Good
##            Season
## Job.Region       Fall    Spring    Summer    Winter
##   Midwest   0.3599549 0.3928948 0.3487155 0.3100695
##   Northeast 0.3931014 0.4270507 0.3814438 0.3410703
##   South     0.4536508 0.4886193 0.4415039 0.3988726
##   West      0.4668530 0.5019043 0.4546468 0.4116819
##
## Job.Region*Season effect (probability) for Neutral
##            Season
## Job.Region       Fall    Spring    Summer    Winter
##   Midwest   0.5417482 0.5205699 0.5485462 0.5698912
##   Northeast 0.5204317 0.4969478 0.5281288 0.5530299
##   South     0.4775909 0.4510860 0.4865246 0.5165476
##   West      0.4677148 0.4407429 0.4768518 0.5077635
##
## Job.Region*Season effect (probability) for Poor
##            Season
## Job.Region        Fall     Spring     Summer     Winter
##   Midwest   0.09829692 0.08653529 0.10273833 0.12003927
##   Northeast 0.08646687 0.07600144 0.09042736 0.10589984
##   South     0.06875826 0.06029471 0.07197154 0.08457980
##   West      0.06543219 0.05735284 0.06850146 0.08055459

# Compute confusion matrix
conf_matrixRS <- confusionMatrix(data = predictcustsatRS,
                          reference = datatest$Overall.Satisfaction)
# Extract performance metrics
accuracyRS <- conf_matrixRS$overall['Accuracy']
# Print performance metrics
print(paste("Accuracy:", accuracyRS))
## [1] "Accuracy: 0.480891719745223"
```

Model5: Region and Temperature Range as Predictors

```
library(MASS)
olsmodelRT = polr(Overall.Satisfaction ~ Job.Region + Temperature.levels, data = datatrain, He
ss = TRUE)
summary(olsmodelRT)
## Call:
## polr(formula = Overall.Satisfaction ~ Job.Region + Temperature.levels,
##     data = datatrain, Hess = TRUE)
##
## Coefficients:
##                              Value Std. Error t value
## Job.RegionNortheast      -0.14562     0.1925 -0.7564
## Job.RegionSouth          -0.40334     0.1576 -2.5588
## Job.RegionWest           -0.46273     0.1644 -2.8146
## Temperature.levelsLow    -0.01601     0.1492 -0.1073
## Temperature.levelsMedium  0.09483     0.1158  0.8190
##
## Intercepts:
##              Value   Std. Error t value
## Good|Neutral -0.5708  0.1499    -3.8077
## Neutral|Poor  2.2172  0.1683    13.1755
##
## Residual Deviance: 2554.862
## AIC: 2568.862

#Compute confusion table and misclassification error
predictcustsatRT = predict(olsmodelRT,datatest)
table(datatest$Overall.Satisfaction, predictcustsatRT)
##          predictcustsatRT
##           Good Neutral Poor
##   Good      65     382    0
##   Neutral   71     362    0
##   Poor       8      54    0

mean(as.character(datatest$Overall.Satisfaction) != as.character(predictcustsatRT))
## [1] 0.5467091

library("effects")
Effect(focal.predictors = c("Job.Region", "Temperature.levels"), olsmodelRT)

## Job.Region*Temperature.levels effect (probability) for Good
##            Temperature.levels
## Job.Region       High       Low    Medium
##   Midwest   0.3610629 0.3647643 0.3394864
##   Northeast 0.3952891 0.3991221 0.3728608
##   South     0.4582451 0.4622220 0.4348132
##   West      0.4730210 0.4770132 0.4494604
##
## Job.Region*Temperature.levels effect (probability) for Neutral
##            Temperature.levels
## Job.Region       High       Low    Medium
##   Midwest   0.5407219 0.5384294 0.5535736
##   Northeast 0.5186600 0.5160777 0.5333316
##   South     0.4739280 0.4709563 0.4911128
##   West      0.4628120 0.4597745 0.4804373
##
## Job.Region*Temperature.levels effect (probability) for Poor
##            Temperature.levels
## Job.Region        High        Low     Medium
##   Midwest   0.09821516 0.09680636 0.10694006
##   Northeast 0.08605089 0.08480017 0.09380760
##   South     0.06782690 0.06682169 0.07407401
##   West      0.06416699 0.06321234 0.07010236

# Compute confusion matrix
conf_matrixRT <- confusionMatrix(data = predictcustsatRT, reference = datatest$Overall.Satisf
action)

# Extract performance metrics
accuracyRT <- conf_matrixRT$overall['Accuracy']

# Print performance metrics
print(paste("Accuracy:", accuracyRT))
## [1] "Accuracy: 0.453290870488323"
```

Model6: Season and Temperature Range as Predictors

```
library(MASS)
olsmodelST = polr(Overall.Satisfaction ~ Season + Temperature.levels, data = datatrain, Hess =
TRUE)
summary(olsmodelST)
## Call:
## polr(formula = Overall.Satisfaction ~ Season + Temperature.levels,
##     data = datatrain, Hess = TRUE)
##
## Coefficients:
##                          Value Std. Error  t value
## SeasonSpring          -0.22744     0.1510 -1.50652
## SeasonSummer           0.03522     0.1398  0.25203
## SeasonWinter           0.17046     0.1730  0.98509
## Temperature.levelsLow -0.01400     0.1753 -0.07985
## Temperature.levelsMedium  0.09505     0.1224  0.77647
##
## Intercepts:
##              Value   Std. Error t value
## Good|Neutral -0.2525  0.1313    -1.9230
## Neutral|Poor  2.5274  0.1566    16.1365
##
## Residual Deviance: 2560.216
## AIC: 2574.216

#Compute confusion table and misclassification error

predictcustsatST = predict(olsmodelST,datatest)
table(datatest$Overall.Satisfaction, predictcustsatST)
##          predictcustsatST
##           Good Neutral Poor
##   Good     103     344    0
##   Neutral   88     345    0
##   Poor      10      52    0

mean(as.character(datatest$Overall.Satisfaction) != as.character(predictcustsatST))
## [1] 0.5244161

library("effects")
Effect(focal.predictors = c("Season", "Temperature.levels"), olsmodelST)
## Season*Temperature.levels effect (probability) for Good
##         Temperature.levels
## Season        High       Low    Medium
##   Fall   0.4372075 0.4406549 0.4139759
##   Spring 0.4937342 0.4972335 0.4700075
##   Summer 0.4285602 0.4319918 0.4054571
##   Winter 0.3958088 0.3991613 0.3733179
##
## Season*Temperature.levels effect (probability) for Neutral
##         Temperature.levels
## Season        High       Low    Medium
##   Fall   0.4888351 0.4863408 0.5052875
##   Spring 0.4464536 0.4437366 0.4646062
##   Summer 0.4950334 0.4925839 0.5111531
##   Winter 0.5176782 0.5154256 0.5323564
##
## Season*Temperature.levels effect (probability) for Poor
##         Temperature.levels
## Season         High        Low     Medium
##   Fall   0.07395738 0.07300436 0.08073655
##   Spring 0.05981220 0.05902983 0.06538633
##   Summer 0.07640633 0.07542432 0.08338977
##   Winter 0.08651295 0.08541306 0.09432577

# Compute confusion matrix
conf_matrixST <- confusionMatrix(data = predictcustsatST, reference = datatest$Overall.Satisfaction)

# Extract performance metrics
accuracyST <- conf_matrixST$overall['Accuracy']

# Print performance metrics
print(paste("Accuracy:", accuracyST))

## [1] "Accuracy: 0.475583864118896"
```

Model7: Region, Season and Temperature Range as Predictors

#Build ordinal logistic regression model - region, season, temperature predictors

```
library(MASS)
olsmodel = polr(Overall.Satisfaction ~ Job.Region + Season + Temperature.levels , data = datatrain, Hess = TRUE)
summary(olsmodel)
## Call:
## polr(formula = Overall.Satisfaction ~ Job.Region + Season + Temperature.levels,
##     data = datatrain, Hess = TRUE)
##
## Coefficients:
##                              Value Std. Error t value
## Job.RegionNortheast      -0.15225     0.1928 -0.7898
## Job.RegionSouth          -0.41053     0.1628 -2.5216
## Job.RegionWest           -0.47475     0.1656 -2.8665
## SeasonSpring             -0.15220     0.1547 -0.9836
## SeasonSummer              0.04091     0.1402  0.2919
## SeasonWinter              0.27099     0.1781  1.5215
## Temperature.levelsLow    -0.10817     0.1809 -0.5978
## Temperature.levelsMedium  0.09063     0.1253  0.7234
##
## Intercepts:
##              Value   Std. Error t value
## Good|Neutral -0.5757  0.1798    -3.2021
## Neutral|Poor  2.2187  0.1954    11.3552
##
## Residual Deviance: 2549.835
## AIC: 2569.835
```

```
#Compute confusion table and misclassification error

predictcustsat = predict(olsmodel,datatest)
table(datatest$Overall.Satisfaction, predictcustsat)
##          predictcustsat
##           Good Neutral Poor
##   Good     121     326    0
##   Neutral   91     342    0
##   Poor      14      48    0

mean(as.character(datatest$Overall.Satisfaction) != as.character(predictcustsat))
## [1] 0.5084926

library("effects")
Effect(focal.predictors = c("Job.Region", "Season", "Temperature.levels"), olsmodel)
## Job.Region*Season*Temperature.levels effect (probability) for Good
## , , Temperature.levels = High
##
##            Season
## Job.Region       Fall    Spring    Summer    Winter
##   Midwest   0.3599269 0.3956839 0.3505572 0.3001322
##   Northeast 0.3956963 0.4326051 0.3859569 0.3330512
##   South     0.4588062 0.4967626 0.4486668 0.3926620
##   West      0.4747878 0.5128135 0.4645984 0.4080768
##
## , , Temperature.levels = Low
##
##            Season
## Job.Region       Fall    Spring    Summer    Winter
##   Midwest   0.3852057 0.4218191 0.3755641 0.3233329
##   Northeast 0.4218317 0.4593254 0.4118875 0.3574968
##   South     0.4857593 0.5237874 0.4755477 0.4187358
##   West      0.5018092 0.5397749 0.4915827 0.4344421
##
## , , Temperature.levels = Medium
##
##            Season
## Job.Region       Fall    Spring    Summer    Winter
##   Midwest   0.3393242 0.3742311 0.3302142 0.2814472
##   Northeast 0.3742432 0.4105104 0.3647132 0.3132341
##   South     0.4364025 0.4741293 0.4263682 0.3712724
##   West      0.4522558 0.4901612 0.4421429 0.3863818
##
##
## Job.Region*Season*Temperature.levels effect (probability) for Neutral
## , , Temperature.levels = High
```

```
##
##            Season
## Job.Region      Fall    Spring    Summer    Winter
##   Midwest   0.5419885 0.5188965 0.5476792 0.5750637
##   Northeast 0.5188882 0.4931432 0.5253770 0.5578475
##   South     0.4739125 0.4449010 0.4814388 0.5209246
##   West      0.4618505 0.4322792 0.4695682 0.5104465
##
## , , Temperature.levels = Low
##
##            Season
## Job.Region      Fall    Spring    Summer    Winter
##   Midwest   0.5258716 0.5008417 0.5321428 0.5632069
##   Northeast 0.5008328 0.4735244 0.5078070 0.5434801
##   South     0.4534381 0.4235421 0.4612712 0.5030171
##   West      0.4409535 0.4106682 0.4489321 0.4918187
##
## , , Temperature.levels = Medium
##
##            Season
## Job.Region      Fall    Spring    Summer    Winter
##   Midwest   0.5542769 0.5329981 0.5594342 0.5835087
##   Northeast 0.5329904 0.5087630 0.5390196 0.5685389
##   South     0.4904011 0.4623521 0.4976113 0.5348863
##   West      0.4787864 0.4500346 0.4862261 0.5250969
##
##
## Job.Region*Season*Temperature.levels effect (probability) for Poor
## , , Temperature.levels = High
##
##            Season
## Job.Region       Fall     Spring     Summer     Winter
##   Midwest   0.09808466 0.08541962 0.10176364 0.12480405
##   Northeast 0.08541558 0.07425175 0.08866608 0.10910126
##   South     0.06728128 0.05833642 0.06989441 0.08641344
##   West      0.06336175 0.05490731 0.06583340 0.08147672
##
## , , Temperature.levels = Low
##
##            Season
## Job.Region       Fall     Spring     Summer     Winter
##   Midwest   0.08892263 0.07733919 0.09229313 0.11346016
##   Northeast 0.07733550 0.06715019 0.08030551 0.09902316
##   South     0.06080260 0.05267043 0.06318116 0.07824704
##   West      0.05723734 0.04955685 0.05948526 0.07373920
##
```

```
## , , Temperature.levels = Medium
##
##            Season
## Job.Region        Fall     Spring     Summer     Winter
##   Midwest   0.10639892 0.09277075 0.11035160 0.13504409
##   Northeast 0.09276639 0.08072661 0.09626719 0.11822694
##   South     0.07319636 0.06351866 0.07602051 0.09384132
##   West      0.06895773 0.05980428 0.07163097 0.08852137

# Compute confusion matrix
conf_matrix <- confusionMatrix(data = predictcustsat,
                    reference = datatest$Overall.Satisfaction)

# Extract performance metrics
accuracy <- conf_matrix$overall['Accuracy']

# Print performance metrics
print(paste("Accuracy:", accuracy))
## [1] "Accuracy: 0.491507430997877"

# Check for assumptions
# Load necessary libraries
library(MASS)
library(car)
library(brant)

# Checking for multicollinearity
# Calculating VIF
vif_results = vif(olsmodel)
print(vif_results)
```

```
                      GVIF Df GVIF^(1/(2*Df))
Job.Region         1.173687  3        1.027051
Season             1.642341  3        1.086202
Temperature.levels 1.610699  2        1.126558
```

```
# Brant test for proportional odds assumption
brant_results = brant(olsmodel)
print(brant_results)

------------------------------------------------------------
Test for                  X2      df      probability
------------------------------------------------------------
Omnibus                   15.71   8       0.05
Job.RegionNortheast       2.94    1       0.09
Job.RegionSouth           0.51    1       0.48
Job.RegionWest          1.11    1       0.29
SeasonSpring            0.08    1       0.78
SeasonSummer            1.76    1       0.18
SeasonWinter            2.13    1       0.14
Temperature.levelsLow           1.29    1       0.26
Temperature.levelsMedium        2.38    1       0.12
------------------------------------------------------------

H0: Parallel Regression Assumption holds
```

Appendix H - Code in RStudio – Naive-Bayes Classifier

Model1: Naive-Bayes Classifier for Ordinal DV (5-10)

#Division of dataset

```
set.seed(2)
nbdat1 <- nbdat[,-c(1,3,5,6,8)]
id <- sample(2, nrow(nbdat1),prob = c(0.7,0.3),replace = T)
nrow(nbdat1)
## [1] 2355
nbtrain <- nbdat1[id == 1,]
nbtest <- nbdat1[id == 2,]
summary(nbtrain)
##      Job.Region     Season    Temperature.range Overall.Satisfaction
##  Midwest  :226   Fall  :524   71-80  :436       Min.   : 5.000
##  Northeast:229   Spring:332   61-70  :418       1st Qu.: 8.000
##  South    :664   Summer:539   51-60  :261       Median : 9.000
##  West     :514   Winter:238   81-90  :200       Mean   : 9.064
##                               41-50  :182       3rd Qu.:10.000
##                               31-40  : 88       Max.   :10.000
##                               (Other): 48
str(nbtrain)
## 'data.frame':    1633 obs. of  4 variables:
##  $ Job.Region          : Factor w/ 4 levels "Midwest","Northeast",..: 3 3 3 1 3 1 1 1 1 1 ...
##  $ Season              : Factor w/ 4 levels "Fall","Spring",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Temperature.range   : Factor w/ 8 levels "20-30","31-40",..: 6 6 6 5 2 5 5 2 2 2 ...
##  $ Overall.Satisfaction: num  10 10 10 5 9 9 8 9 10 10 ...
```

#Build the Naive Bayes Model

```
library(e1071)
library(caret)
## Loading required package: ggplot2
## Loading required package: lattice
nbmodel1 <- naiveBayes(Overall.Satisfaction ~ .,data = nbtrain)
nbmodel1
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##           5           6           7           8           9          10
## 0.018371096 0.009797918 0.044703001 0.179424372 0.312308634 0.435394979
```

```
##
## Conditional probabilities:
##     Job.Region
## Y       Midwest  Northeast      South       West
##   5  0.16666667 0.10000000 0.50000000 0.23333333
##   6  0.00000000 0.12500000 0.62500000 0.25000000
##   7  0.17808219 0.06849315 0.42465753 0.32876712
##   8  0.20819113 0.18430034 0.28327645 0.32423208
##   9  0.14509804 0.16078431 0.40784314 0.28627451
##   10 0.10267229 0.11673699 0.44585091 0.33473980
##
##     Season
## Y          Fall    Spring    Summer    Winter
##   5  0.2333333 0.2333333 0.2000000 0.3333333
##   6  0.4375000 0.1875000 0.3125000 0.0625000
##   7  0.3013699 0.1917808 0.3698630 0.1369863
##   8  0.3481229 0.1433447 0.3481229 0.1604096
##   9  0.3058824 0.2117647 0.3411765 0.1411765
##   10 0.3234880 0.2222222 0.3164557 0.1378340
##
##     Temperature.range
## Y          20-30      31-40      41-50      51-60      61-70      71-80
##   5  0.03333333 0.06666667 0.16666667 0.13333333 0.13333333 0.33333333
##   6  0.00000000 0.06250000 0.06250000 0.12500000 0.18750000 0.43750000
##   7  0.00000000 0.01369863 0.12328767 0.16438356 0.23287671 0.31506849
##   8  0.01706485 0.06143345 0.08191126 0.16723549 0.26621160 0.27645051
##   9  0.01568627 0.06666667 0.12352941 0.14313725 0.27647059 0.24509804
##   10 0.01125176 0.04500703 0.11251758 0.17018284 0.24613221 0.26722925
##     Temperature.range
## Y          81-90     91-100
##   5  0.13333333 0.00000000
##   6  0.12500000 0.00000000
##   7  0.13698630 0.01369863
##   8  0.11262799 0.01706485
##   9  0.11568627 0.01372549
##   10 0.12939522 0.01828411
```

```
summary(nbmodel1)

##           Length Class  Mode
## apriori   6      table  numeric
## tables    3      -none- list
## levels    6      -none- character
## isnumeric 3      -none- logical
## call      4      -none- call
```

```
prenb1 <- predict(nbmodel1, nbtest)
confusionMatrix(table(prenb1, nbtest$Overall.Satisfaction))

## Confusion Matrix and Statistics
##
## prenb1   5   6   7   8   9  10
##     5    0   0   0   0   0   0
##     6    0   0   0   0   0   0
##     7    0   0   0   0   0   0
##     8    0   0   1   0   0   1
##     9    2   2   8  21  43  47
##     10  11   9  18 104 159 296
##
## Overall Statistics
##
##                Accuracy : 0.4695
##                  95% CI : (0.4326, 0.5067)
##     No Information Rate : 0.4765
##     P-Value [Acc > NIR] : 0.6588
##
##                   Kappa : 0.0492
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 5 Class: 6 Class: 7 Class: 8 Class: 9 Class: 10
## Sensitivity           0.00000  0.00000   0.0000  0.00000  0.21287    0.8605
## Specificity           1.00000  1.00000   1.0000  0.99665  0.84615    0.2037
## Pos Pred Value            NaN      NaN      NaN  0.00000  0.34959    0.4958
## Neg Pred Value        0.98199  0.98476   0.9626  0.82639  0.73456    0.6160
## Prevalence            0.01801  0.01524   0.0374  0.17313  0.27978    0.4765
## Detection Rate        0.00000  0.00000   0.0000  0.00000  0.05956    0.4100
## Detection Prevalence  0.00000  0.00000   0.0000  0.00277  0.17036    0.8269
## Balanced Accuracy     0.50000  0.50000   0.5000  0.49832  0.52951    0.5321
```

Model2: Naive-Bayes Classifier for Binary DV (0,1)

#Division of dataset

```
set.seed(2)
nbdat2 <- nbdatcat[,-c(1,3,5,6,8,9,11)]
id <- sample(2, nrow(nbdat2),prob = c(0.7,0.3),replace = T)
nrow(nbdat2)
## [1] 2355
```

```
nbtraincat <- nbdat2[id == 1,]
nbtestcat <- nbdat2[id == 2,]
summary(nbtraincat)
##     Job.Region     Season    Temperature.range
##  Midwest  :226   Fall  :524   71-80  :436
##  Northeast:229   Spring:332   61-70  :418
##  South    :664   Summer:539   51-60  :261
##  West     :514   Winter:238   81-90  :200
##                               41-50  :182
##                               31-40  : 88
##                               (Other): 48
##  Overall.Satisfaction.Dichotomous
##  No : 119
##  Yes:1514
##
str(nbtraincat)
## 'data.frame':    1633 obs. of  4 variables:
##  $ Job.Region                      : Factor w/ 4 levels "Midwest","Northeast",..: 3 3 3 1 3 1 1 1 1 1 ...
##  $ Season                          : Factor w/ 4 levels "Fall","Spring",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Temperature.range               : Factor w/ 8 levels "20-30","31-40",..: 6 6 6 5 2 5 5 2 2 2 ...
##  $ Overall.Satisfaction.Dichotomous: Factor w/ 2 levels "No","Yes": 2 2 2 1 2 2 2 2 2 2 ...
```

#Build the Naive Bayes Model-Categorical

```
library(e1071)
library(caret)
nbmodel2 <- naiveBayes(Overall.Satisfaction.Dichotomous ~ .,data = nbtraincat)
nbmodel2
##
## Naive Bayes Classifier for Discrete Predictors
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         No        Yes
## 0.07287201 0.92712799
##
## Conditional probabilities:
##      Job.Region
## Y        Midwest  Northeast      South       West
##   No  0.15126050 0.08403361 0.47058824 0.29411765
##   Yes 0.13738441 0.14464993 0.40158520 0.31638045
##
##      Season
## Y          Fall    Spring    Summer    Winter
##   No  0.3025210 0.2016807 0.3193277 0.1764706
##   Yes 0.3223250 0.2034346 0.3309115 0.1433289
##
##      Temperature.range
## Y           20-30       31-40       41-50       51-60       61-70       71-80
##   No  0.008403361 0.033613445 0.126050420 0.151260504 0.201680672 0.336134454
##   Yes 0.013870542 0.055482166 0.110303831 0.160501982 0.260237781 0.261558785
##      Temperature.range
## Y           81-90      91-100
##   No  0.134453782 0.008403361
##   Yes 0.121532365 0.016512550

summary(nbmodel2)
##           Length Class  Mode
## apriori   2      table  numeric
## tables    3      -none- list
## levels    2      -none- character
## isnumeric 3      -none- logical
## call      4      -none- call

prenb2 <- predict(nbmodel2, nbtestcat)
confusionMatrix(table(prenb2, nbtestcat$Overall.Satisfaction))
## Confusion Matrix and Statistics
##
## prenb2  No Yes
##    No    0   0
##    Yes  51 671
##
##               Accuracy : 0.9294
##                 95% CI : (0.9082, 0.947)
##     No Information Rate : 0.9294
##     P-Value [Acc > NIR] : 0.5372
##
##                  Kappa : 0
##
##  Mcnemar's Test P-Value : 2.534e-12
##
##            Sensitivity : 0.00000
##            Specificity : 1.00000
##         Pos Pred Value :     NaN
##         Neg Pred Value : 0.92936
##             Prevalence : 0.07064
##         Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
##       'Positive' Class : No
```

Appendix I - Code in RStudio – Support Vector Machine

#Perform One Hot Encoding

```
encoded_data <- model.matrix(~ . -1, data = svm1[, c(1,2)])
svm1E <- cbind(svm1[, -c(1,2)], encoded_data)
str(svm1E)
## 'data.frame':    2355 obs. of  9 variables:
##  $ Temp.Code   : num  7 7 7 7 6 6 6 6 3 6 ...
##  $ Sat.Code    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 1 1 2 2 ...
##  $ Region.Code1: num  0 0 0 0 1 1 1 1 0 1 ...
##  $ Region.Code2: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region.Code3: num  1 1 1 1 0 0 0 0 1 0 ...
##  $ Region.Code4: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season.Code2: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season.Code3: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season.Code4: num  0 0 0 0 0 0 0 0 0 0 ...

summary(svm1E)
##    Temp.Code     Sat.Code    Region.Code1     Region.Code2     Region.Code3
##  Min.   :2.000   No : 170   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:5.000   Yes:2185   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :6.000              Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :5.935              Mean   :0.1418   Mean   :0.1329   Mean   :0.4191
##  3rd Qu.:7.000              3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :9.000              Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   Region.Code4     Season.Code2     Season.Code3     Season.Code4
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.3062   Mean   :0.2021   Mean   :0.3287   Mean   :0.1503
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000

set.seed(123)
train_index <- sample(1:nrow(svm1E), 0.7 * nrow(svm1E))
train_svm <- svm1E[train_index, ]
test_svm <- svm1E[-train_index, ]
str(train_svm)
## 'data.frame':    1648 obs. of  9 variables:
##  $ Temp.Code   : num  4 6 7 4 7 7 8 6 6 6 ...
##  $ Sat.Code    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Region.Code1: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region.Code2: num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Region.Code3: num  0 1 1 1 1 1 1 1 0 0 ...
##  $ Region.Code4: num  1 0 0 0 0 0 0 0 1 0 ...
##  $ Season.Code2: num  0 1 0 0 0 0 0 1 0 0 ...
##  $ Season.Code3: num  0 0 0 0 1 1 1 0 0 1 ...
##  $ Season.Code4: num  1 0 0 1 0 0 0 0 0 0 ...

str(test_svm)
## 'data.frame':    707 obs. of  9 variables:
##  $ Temp.Code   : num  7 6 3 9 5 6 6 7 6 8 ...
##  $ Sat.Code    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Region.Code1: num  0 1 1 0 0 1 1 0 0 0 ...
##  $ Region.Code2: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region.Code3: num  1 0 0 1 1 0 0 1 1 1 ...
##  $ Region.Code4: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season.Code2: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season.Code3: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season.Code4: num  0 0 0 0 0 0 0 0 0 0 ...
```

SVM classifier - Linear Kernel

```
svmlinear <- svm(Sat.Code ~ . , data = train_svm, kernel = "linear", cost = .1, scale = FALSE)
print(svmlinear)
## Call:
## svm(formula = Sat.Code ~ ., data = train_svm, kernel = "linear",
##     cost = 0.1, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  253
```

#Prediction and Accuracy

```
# Predict on test data
svmpredL <- predict(svmlinear, newdata = test_svm)

# Evaluate performance
accuracyL <- mean(svmpredL == test_svm$Sat.Code)
cat("Accuracy:", accuracyL, "\n")

## Accuracy: 0.9306931

# Confusion matrix
CM_linear <- confusionMatrix(svmpredL, test_svm$Sat.Code)
print(CM_linear)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No    0   0
##        Yes  49 658
##
##                Accuracy : 0.9307
##                  95% CI : (0.9094, 0.9483)
##     No Information Rate : 0.9307
##     P-Value [Acc > NIR] : 0.5379
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 7.025e-12
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.93069
##              Prevalence : 0.06931
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
##        'Positive' Class : No
```

SVM classifier - Radial Kernel

```
svm_model <- svm(Sat.Code ~ ., data = train_svm, kernel = "radial")
summary(svm_model)
## Call:
## svm(formula = Sat.Code ~ ., data = train_svm, kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  254
##
##  ( 133 121 )
##
## Number of Classes:  2
```

```
##
## Levels:
##  No Yes
```

#Prediction and Accuracy

```
# Predict on test data
svmpred <- predict(svm_model, newdata = test_svm)

# Evaluate performance
accuracy <- mean(svmpred == test_svm$Sat.Code)
cat("Accuracy:", accuracy, "\n")
## Accuracy: 0.9306931

# Confusion matrix
CM_SVM <- confusionMatrix(svmpred, test_svm$Sat.Code)
print(CM_SVM)
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No    0   0
##        Yes  49 658
##
##                Accuracy : 0.9307
##                  95% CI : (0.9094, 0.9483)
##     No Information Rate : 0.9307
##     P-Value [Acc > NIR] : 0.5379
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 7.025e-12
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.93069
##              Prevalence : 0.06931
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
##        'Positive' Class : No
##
```

SVM classifier – Principal Component Analysis and Radial Kernel

```
library(e1071)
library(ggplot2)
library(RColorBrewer)

# Perform PCA on the training data (excluding the response variable)
pca <- prcomp(train_svm[, -which(names(train_svm) == "Sat.Code")], scale. = TRUE)

# Predict principal components for the training data
pca_data_train <- data.frame(predict(pca, newdata = train_svm[, -which(names(train_svm) == "
Sat.Code")]))
pca_data_train$Sat.Code <- train_svm$Sat.Code

# Predict principal components for the test data
pca_data_test <- data.frame(predict(pca, newdata = test_svm[, -which(names(test_svm) == "Sat.
Code")]))
pca_data_test$Sat.Code <- test_svm$Sat.Code

# Train SVM model using the first two principal components
svm_model_pca <- svm(Sat.Code ~ PC1 + PC2, data = pca_data_train, kernel = "radial")

# Summary of PCA data
summary(pca_data_train)
##      PC1              PC2              PC3              PC4
##  Min.   :-3.3519   Min.   :-1.6264   Min.   :-1.8725   Min.   :-3.2482
##  1st Qu.:-0.8641   1st Qu.:-1.1799   1st Qu.:-0.9077   1st Qu.:-1.1282
##  Median :-0.0970   Median :-0.6315   Median :-0.3005   Median : 0.2432
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 1.1817   3rd Qu.: 1.1730   3rd Qu.: 1.1180   3rd Qu.: 0.9214
##  Max.   : 2.4213   Max.   : 2.3594   Max.   : 2.3884   Max.   : 1.7346
##      PC5              PC6               PC7              PC8
##  Min.   :-2.8634   Min.   :-1.90241   Min.   :-1.5044   Min.   :-2.259e-15
##  1st Qu.:-0.4867   1st Qu.:-0.34058   1st Qu.:-0.3911   1st Qu.:-8.158e-16
##  Median : 0.1262   Median : 0.01604   Median :-0.1198   Median :-3.076e-16
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 3.995e-17
##  3rd Qu.: 0.7032   3rd Qu.: 0.50158   3rd Qu.: 0.4232   3rd Qu.: 3.167e-16
##  Max.   : 2.3295   Max.   : 1.63340   Max.   : 1.7816   Max.   : 3.647e-15
##  Sat.Code
##  No : 121
##  Yes:1527
##
summary(pca_data_test)
##      PC1                PC2                 PC3               PC4
##  Min.   :-3.351898   Min.   :-1.6099616   Min.   :-1.87248   Min.   :-3.21279
##  1st Qu.:-0.824024   1st Qu.:-1.1798881   1st Qu.:-0.67753   1st Qu.:-1.16356
##  Median :-0.037707   Median :-0.4177651   Median :-0.24041   Median : 0.27856
##  Mean   :-0.002877   Mean   : 0.0005052   Mean   : 0.03377   Mean   :-0.03066
##  3rd Qu.: 1.131876   3rd Qu.: 1.1729823   3rd Qu.: 1.11796   3rd Qu.: 0.88599
##  Max.   : 2.421250   Max.   : 2.3593658   Max.   : 2.38838   Max.   : 1.73459
##       PC5                PC6                PC7                PC8
##  Min.   :-2.88493   Min.   :-1.90241   Min.   :-1.50440   Min.   :-2.259e-15
##  1st Qu.:-0.48674   1st Qu.:-0.31932   1st Qu.:-0.47553   1st Qu.:-8.158e-16
##  Median : 0.14397   Median : 0.02881   Median :-0.11979   Median :-3.076e-16
##  Mean   : 0.01895   Mean   : 0.04381   Mean   :-0.04337   Mean   : 2.050e-17
##  3rd Qu.: 0.72477   3rd Qu.: 0.55006   3rd Qu.: 0.35376   3rd Qu.: 3.931e-16
##  Max.   : 2.32950   Max.   : 1.77820   Max.   : 1.78165   Max.   : 3.647e-15
##  Sat.Code
##  No : 49
##  Yes:658
##
# Predict and evaluate the model on the test set
pred <- predict(svm_model_pca, newdata = pca_data_test)

# Confusion matrix to evaluate the model
table(Predicted = pred, Actual = pca_data_test$Sat.Code)
##          Actual
## Predicted  No Yes
##       No    0   0
##       Yes  49 658

CM_svmpca <- confusionMatrix(pred, pca_data_test$Sat.Code)
print(CM_svmpca)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No    0   0
##        Yes  49 658
##
##                Accuracy : 0.9307
##                  95% CI : (0.9094, 0.9483)
##     No Information Rate : 0.9307
##     P-Value [Acc > NIR] : 0.5379
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 7.025e-12
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##         Pos Pred Value :     NaN
##         Neg Pred Value : 0.93069
##             Prevalence : 0.06931
##         Detection Rate : 0.00000
##   Detection Prevalence : 0.00000
##      Balanced Accuracy : 0.50000
##
##       'Positive' Class : No
```

SVM classifier – Data Pre-processing for a Balanced Dataset

```
#Load Data
library(caret)
## Loading required package: ggplot2
## Loading required package: lattice
library(e1071)
svm1 <- read.csv("ManudataSVM1.csv", header = T)
str(svm1)
## 'data.frame':    2355 obs. of  4 variables:
##  $ Region.Code: int  3 3 3 3 1 1 1 1 3 1 ...
##  $ Season.Code: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Temp.Code  : int  7 7 7 7 6 6 6 6 3 6 ...
##  $ Sat.Code   : chr  "Yes" "Yes" "Yes" "Yes" ...

summary(svm1)
##   Region.Code    Season.Code     Temp.Code       Sat.Code
##  Min.   :1.00   Min.   :1.00   Min.   :2.000   Length:2355
##  1st Qu.:2.00   1st Qu.:1.00   1st Qu.:5.000   Class :character
##  Median :3.00   Median :2.00   Median :6.000   Mode  :character
##  Mean   :2.89   Mean   :2.31   Mean   :5.935
##  3rd Qu.:4.00   3rd Qu.:3.00   3rd Qu.:7.000
##  Max.   :4.00   Max.   :4.00   Max.   :9.000

# Dependent variable column is named 'Sat.Code'
class_counts <- table(svm1$Sat.Code)
print(class_counts)
##
##   No  Yes
##  170 2185

# Identify the minority class
minority_class <- names(which.min(class_counts))
minority_count <- min(class_counts)

# Separate the data into two data frames based on class
```

```
minority_df <- svm1[svm1$Sat.Code == minority_class, ]
majority_df <- svm1[svm1$Sat.Code != minority_class, ]

# Randomly sample from the majority class to match the minority class count
set.seed(1)  # For reproducibility
majority_sampled_df <- majority_df[sample(nrow(majority_df), minority_count), ]

# Combine the sampled majority class with the minority class
balanced_df <- rbind(minority_df, majority_sampled_df)

# Shuffle the data frame to mix the classes
set.seed(1)  # For reproducibility
balanced_df <- balanced_df[sample(nrow(balanced_df)), ]
str(balanced_df)
## 'data.frame':    340 obs. of  4 variables:
##  $ Region.Code: int  3 4 1 4 1 4 3 4 3 3 ...
##  $ Season.Code: int  4 4 3 3 3 2 3 2 1 4 ...
##  $ Temp.Code  : int  4 6 8 5 7 7 9 6 4 7 ...
##  $ Sat.Code   : chr  "Yes" "No" "No" "Yes" ...

summary(balanced_df)
##   Region.Code     Season.Code      Temp.Code       Sat.Code
##  Min.   :1.000   Min.   :1.000   Min.   :2.000   Length:340
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:5.000   Class :character
##  Median :3.000   Median :2.500   Median :6.000   Mode  :character
##  Mean   :2.853   Mean   :2.362   Mean   :5.979
##  3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:7.000
##  Max.   :4.000   Max.   :4.000   Max.   :9.000
```

```
#Conversion to factors
```

```
balanced_df$Region.Code <- as.factor(balanced_df$Region.Code)
balanced_df$Season.Code <- as.factor(balanced_df$Season.Code)
balanced_df$Temp.Code <- as.numeric(balanced_df$Temp.Code)
balanced_df$Sat.Code <- as.factor(balanced_df$Sat.Code)

str(balanced_df)
## 'data.frame':    340 obs. of  4 variables:
##  $ Region.Code: Factor w/ 4 levels "1","2","3","4": 3 4 1 4 1 4 3 4 3 3 ...
##  $ Season.Code: Factor w/ 4 levels "1","2","3","4": 4 4 3 3 3 2 3 2 1 4 ...
##  $ Temp.Code  : num  4 6 8 5 7 7 9 6 4 7 ...
##  $ Sat.Code   : Factor w/ 2 levels "No","Yes": 2 1 1 2 2 2 2 1 2 2 ...

summary(balanced_df)
##  Region.Code Season.Code   Temp.Code     Sat.Code
##  1: 52       1:106       Min.   :2.000   No :170
##  2: 36       2: 64       1st Qu.:5.000   Yes:170
##  3:162      3:111      Median :6.000  
##  4: 90      4: 59      Mean   :5.979  
##                        3rd Qu.:7.000  
##                        Max.   :9.000  
```

#Perform One Hot Encoding

```
encodeddata <- model.matrix(~ . -1, data = balanced_df[, c(1,2)])
svm_new <- cbind(balanced_df[, -c(1,2)], encodeddata)
str(svm_new)
## 'data.frame':    340 obs. of  9 variables:
##  $ Temp.Code   : num  4 6 8 5 7 7 9 6 4 7 ...
##  $ Sat.Code    : Factor w/ 2 levels "No","Yes": 2 1 1 2 2 2 2 1 2 2 ...
##  $ Region.Code1: num  0 0 1 0 1 0 0 0 0 0 ...
##  $ Region.Code2: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Region.Code3: num  1 0 0 0 0 0 1 0 1 1 ...
##  $ Region.Code4: num  0 1 0 1 0 1 0 1 0 0 ...
##  $ Season.Code2: num  0 0 0 0 0 1 0 1 0 0 ...
##  $ Season.Code3: num  0 0 1 1 1 0 1 0 0 0 ...
##  $ Season.Code4: num  1 1 0 0 0 0 0 0 0 1 ...

summary(svm_new)
##    Temp.Code     Sat.Code   Region.Code1     Region.Code2     Region.Code3   
##  Min.   :2.000   No :170   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:5.000   Yes:170   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :6.000             Median :0.0000   Median :0.0000   Median :0.0000  
##  Mean   :5.979             Mean   :0.1529   Mean   :0.1059   Mean   :0.4765  
##  3rd Qu.:7.000             3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000  
##  Max.   :9.000             Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
##   Region.Code4     Season.Code2     Season.Code3     Season.Code4   
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000  
##  Mean   :0.2647   Mean   :0.1882   Mean   :0.3265   Mean   :0.1735  
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000  
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
```

#Splitting Data into Training and Testing Datasets

```
set.seed(123)
trainindex <- sample(1:nrow(svm_new), 0.7 * nrow(svm_new))
trainsvm1 <- svm_new[trainindex, ]
testsvm1 <- svm_new[-trainindex, ]

str(trainsvm1)
## 'data.frame':    237 obs. of  9 variables:
##  $ Temp.Code   : num  5 7 7 6 5 8 7 5 8 6 ...
##  $ Sat.Code    : Factor w/ 2 levels "No","Yes": 2 2 2 1 2 1 1 2 2 2 ...
##  $ Region.Code1: num  1 0 0 0 0 0 0 0 0 0 ...
##  $ Region.Code2: num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Region.Code3: num  0 0 0 0 1 0 1 0 1 1 ...
##  $ Region.Code4: num  0 1 1 1 0 1 0 0 0 0 ...
##  $ Season.Code2: num  0 1 0 0 0 0 0 0 0 1 ...
##  $ Season.Code3: num  0 0 0 1 0 0 0 0 1 0 ...
##  $ Season.Code4: num  0 0 0 0 1 0 0 0 0 0 ...

str(testsvm1)
## 'data.frame':    103 obs. of  9 variables:
##  $ Temp.Code   : num  6 8 4 5 7 5 6 7 7 6 ...
##  $ Sat.Code    : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 1 1 2 ...
##  $ Region.Code1: num  0 1 0 1 0 0 0 1 1 0 ...
##  $ Region.Code2: num  0 0 1 0 0 0 1 0 0 0 ...
##  $ Region.Code3: num  0 0 0 0 0 1 0 0 0 0 ...
##  $ Region.Code4: num  1 0 0 0 1 0 0 0 0 1 ...
##  $ Season.Code2: num  0 0 0 0 0 1 0 0 0 0 ...
##  $ Season.Code3: num  0 1 0 0 0 0 0 1 1 1 ...
##  $ Season.Code4: num  1 0 1 0 1 0 0 0 0 0 ...
```

SVM classifier – Balanced Dataset, Linear Kernel and Binary DV

#Train SVM classifier - Linear Kernel

```
svmfit1 <- svm(Sat.Code ~ . , data = trainsvm1, kernel = "linear", cost = .1, scale = FALSE)
print(svmfit1)
##
## Call:
## svm(formula = Sat.Code ~ ., data = trainsvm1, kernel = "linear",
##     cost = 0.1, scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  214
```

#Prediction and Accuracy

```
# Predict on test data
svmpredL1 <- predict(svmfit1, newdata = testsvm1)

# Evaluate performance
accuracyL1 <- mean(svmpredL1 == testsvm1$Sat.Code)
cat("Accuracy:", accuracyL1, "\n")
## Accuracy: 0.5048544
```

```
library(pROC)

# Generate Confusion Matrix
conf_matrix1 <- table(Predicted = svmpredL1, Actual = testsvm1$Sat.Code)
print(conf_matrix1)
##          Actual
## Predicted No Yes
##       No  27  19
##       Yes 32  25

CM_Bal_L <- confusionMatrix (svmpredL1, testsvm1$Sat.Code)
CM_Bal_L
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  27  19
##        Yes 32  25
##
##                Accuracy : 0.5049
##                  95% CI : (0.4046, 0.6049)
##     No Information Rate : 0.5728
##     P-Value [Acc > NIR] : 0.93183
##
##                   Kappa : 0.0249
##
##  Mcnemar's Test P-Value : 0.09289
##
##             Sensitivity : 0.4576
##             Specificity : 0.5682
##          Pos Pred Value : 0.5870
##          Neg Pred Value : 0.4386
##              Prevalence : 0.5728
##          Detection Rate : 0.2621
##    Detection Prevalence : 0.4466
##       Balanced Accuracy : 0.5129
##
##        'Positive' Class : No
```

SVM classifier – Balanced Dataset, Radial Kernel and Binary DV

```
svm_model1 <- svm(Sat.Code ~ ., data = trainsvm1, kernel = "radial")
summary(svm_model1)
## Call:
## svm(formula = Sat.Code ~ ., data = trainsvm1, kernel = "radial")
##
```

```
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  203
##
##  ( 102 101 )
##
##
## Number of Classes:  2
##
## Levels:
##  No Yes
```

#Prediction and Accuracy

```
# Predict on test data
svmpred1 <- predict(svm_model1, newdata = testsvm1)

# Evaluate performance
accuracy1 <- mean(svmpred1 == testsvm1$Sat.Code)
cat("Accuracy:", accuracy1, "\n")
## Accuracy: 0.4757282

library(pROC)

# Generate Confusion Matrix
conf_matrix <- table(Predicted = svmpred1, Actual = testsvm1$Sat.Code)
print(conf_matrix)
##          Actual
## Predicted No Yes
##       No  23  18
##       Yes 36  26

CM_Bal <- confusionMatrix (svmpred1, testsvm1$Sat.Code)
CM_Bal
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  23  18
##        Yes 36  26
##
##                Accuracy : 0.4757
##                  95% CI : (0.3764, 0.5765)
##     No Information Rate : 0.5728
##     P-Value [Acc > NIR] : 0.9813
##
##                   Kappa : -0.0183
##
##  Mcnemar's Test P-Value : 0.0207
##
##             Sensitivity : 0.3898
##             Specificity : 0.5909
##          Pos Pred Value : 0.5610
##          Neg Pred Value : 0.4194
##              Prevalence : 0.5728
##          Detection Rate : 0.2233
##    Detection Prevalence : 0.3981
##       Balanced Accuracy : 0.4904
##
##        'Positive' Class : No
```

SVM classifier – Radial Kernel and Ordinal DV

```
## classification mode
# default with factor response
model <- svm(Overall.Satisfaction ~ ., data = svmdat)
print(model)

Call:
svm(formula = Overall.Satisfaction ~ ., data = svmdat)

Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  2213

# alternatively the traditional interface
x <- subset(svmdat[, -4])
y <- svmdat$Overall.Satisfaction
model <- svm(x, y)
print(model)
summary(model)

Call:
svm.default(x = x, y = y)
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  2213

( 1005 1038 170 )

Number of Classes:  3
Levels:
 Good Neutral Poor

# test with train data
pred <- predict(model, x)
# (same as:)
pred <- fitted(model)

# Check accuracy:
table(pred, y)
# Predict on training data
pred <- predict(model, svmdat)

# Calculate accuracy
accuracy <- mean(pred == svmdat$Overall.Satisfaction)
cat("Accuracy:", accuracy, "\n")
Accuracy: 0.5227176

# Load necessary libraries
library(e1071)
library(ggplot2)

# Assuming svmdat is already loaded and prepared as described
# Train SVM model
model <- svm(Overall.Satisfaction ~ ., data = svmdat)

# Predict on training data
pred <- predict(model, svmdat)

# Create confusion matrix
conf_mat <- table(pred, svmdat$Overall.Satisfaction)
print(conf_mat)


         y
pred      Good Neutral Poor
  Good     754     653  106
  Neutral  301     477   64
  Poor       0       0    0
```

Appendix J - Code in RStudio – Artificial Neural Network

```
#normalization of data
normalize <- function(x) {return((x - min(x)) / (max(x) - min(x)))}

#apply to our dataset
ANN.norm <- as.data.frame(lapply(dataANN, normalize))
summary(ANN.norm)

#splitting data into training and testing
train_test_split_index <- 0.75 * nrow(ANN.norm)
train_ANN <- ANN.norm[1:train_test_split_index,]
test_ANN <- ANN.norm[(train_test_split_index + 1): nrow(ANN.norm),]
summary(train_ANN)
summary(test_ANN)

#ANN Level 1 Model
manu_model1 <- neuralnet(Overall.Satisfaction ~ Job.region.code + Season.code +
                    Temperature.code, data = train_ANN)

# Mean Absolute Error (MAE)
mae1 <- mean(abs(predicted_sat1 - test_ANN$Overall.Satisfaction))

# Root Mean Squared Error (RMSE)
rmse1 <- sqrt(mean((predicted_sat1 - test_ANN$Overall.Satisfaction)^2))

# Mean Absolute Percentage Error (MAPE)
mape1 <- mean(abs((predicted_sat1 - test_ANN$Overall.Satisfaction) /
test_ANN$Overall.Satisfaction)) * 100

# Compute MAPE, excluding cases where actual value is zero or close to zero
mape11 <- mean(abs((predicted_sat1 - test_ANN$Overall.Satisfaction) /
pmax(test_ANN$Overall.Satisfaction, 1e-10))) * 100

# R-squared (R2)
r_squared1 <- 1 - sum((test_ANN$Overall.Satisfaction - predicted_sat1)^2) /
sum((test_ANN$Overall.Satisfaction - mean(test_ANN$Overall.Satisfaction))^2)

# Print the performance metrics
print(paste("Mean Absolute Error (MAE):", mae1))
print(paste("Root Mean Squared Error (RMSE):", rmse1))
print(paste("Mean Absolute Percentage Error (MAPE):", mape1))
print(paste("Mean Absolute Percentage Error with no zeroes (MAPE):", mape11))
print(paste("R-squared (R2):", r_squared1))
```

```
[1] "Mean Absolute Error (MAE): 0.176842892847017"
[1] "Root Mean Squared Error (RMSE): 0.237904398353323"
[1] "Mean Absolute Percentage Error (MAPE): Inf"
[1] "Mean Absolute Percentage Error with no zeroes (MAPE): 26792805666.0472"
[1] "R-squared (R2): -0.0109352488319885"
```

#ANN Level 1 Model with Temp as continuous

```
manu_model_V1 <- neuralnet(Overall.Satisfaction ~ Job.region.code + Season.code +
                    Avg.temp, data = train_ANN)

# Mean Absolute Error (MAE)
mae_V1 <- mean(abs(predicted_sat_V1 - test_ANN$Overall.Satisfaction))

# Root Mean Squared Error (RMSE)
rmse_V1 <- sqrt(mean((predicted_sat_V1 - test_ANN$Overall.Satisfaction)^2))

# Mean Absolute Percentage Error (MAPE)
mape_V1 <- mean(abs((predicted_sat_V1 - test_ANN$Overall.Satisfaction) /
test_ANN$Overall.Satisfaction)) * 100

# Compute MAPE, excluding cases where actual value is zero or close to zero
mape_V11 <- mean(abs((predicted_sat_V1 - test_ANN$Overall.Satisfaction) /
pmax(test_ANN$Overall.Satisfaction, 1e-10))) * 100

# R-squared (R2)
r_squared_V1 <- 1 - sum((test_ANN$Overall.Satisfaction - predicted_sat_V1)^2) /
sum((test_ANN$Overall.Satisfaction - mean(test_ANN$Overall.Satisfaction))^2)

# Print the performance metrics
print(paste("Mean Absolute Error (MAE):", mae_V1))
print(paste("Root Mean Squared Error (RMSE):", rmse_V1))
print(paste("Mean Absolute Percentage Error (MAPE):", mape_V1))
print(paste("Mean Absolute Percentage Error with no zeroes (MAPE):", mape_V11))
print(paste("R-squared (R2):", r_squared_V1))
```

```
[1] "Mean Absolute Error (MAE): 0.177678058993484"
[1] "Root Mean Squared Error (RMSE): 0.239117009439303"
[1] "Mean Absolute Percentage Error (MAPE): Inf"
[1] "Mean Absolute Percentage Error with no zeroes (MAPE): 26973629870.8172"
[1] "R-squared (R2): -0.0212670919025226"
```

#ANN Level 2 Model

```
manu_model2 <- neuralnet(Overall.Satisfaction ~ Job.region.code + Season.code +
                    Temperature.code, data = train_ANN, hidden = 3)

# Mean Absolute Error (MAE)
```

```
mae2 <- mean(abs(predicted_sat2 - test_ANN$Overall.Satisfaction))

# Root Mean Squared Error (RMSE)
rmse2 <- sqrt(mean((predicted_sat2 - test_ANN$Overall.Satisfaction)^2))

# Mean Absolute Percentage Error (MAPE)
mape2 <- mean(abs((predicted_sat2 - test_ANN$Overall.Satisfaction) /
test_ANN$Overall.Satisfaction)) * 100

# Compute MAPE, excluding cases where actual value is zero or close to zero
mape22 <- mean(abs((predicted_sat2 - test_ANN$Overall.Satisfaction) /
pmax(test_ANN$Overall.Satisfaction, 1e-10))) * 100

# R-squared (R2)
r_squared2 <- 1 - sum((test_ANN$Overall.Satisfaction - predicted_sat2)^2) /
sum((test_ANN$Overall.Satisfaction - mean(test_ANN$Overall.Satisfaction))^2)

# Print the performance metrics
print(paste("Mean Absolute Error (MAE):", mae2))
print(paste("Root Mean Squared Error (RMSE):", rmse2))
print(paste("Mean Absolute Percentage Error (MAPE):", mape2))
print(paste("Mean Absolute Percentage Error with no zeroes (MAPE):", mape22))
print(paste("R-squared (R2):", r_squared2))
```

```
[1] "Mean Absolute Error (MAE): 0.182674518798517"
[1] "Root Mean Squared Error (RMSE): 0.249282999626445"
[1] "Mean Absolute Percentage Error (MAPE): Inf"
[1] "Mean Absolute Percentage Error with no zeroes (MAPE): 27965345149.7684"
[1] "R-squared (R2): -0.10995078012386"
```

#ANN Level 2 Model with temperature as continuous

```
manu_model_V2 <- neuralnet(Overall.Satisfaction ~ Job.region.code + Season.code +
                    Avg.temp, data = train_ANN, hidden = 3)

# Mean Absolute Error (MAE)
mae_V2 <- mean(abs(predicted_sat_V2 - test_ANN$Overall.Satisfaction))

# Root Mean Squared Error (RMSE)
rmse_V2 <- sqrt(mean((predicted_sat_V2 - test_ANN$Overall.Satisfaction)^2))

# Mean Absolute Percentage Error (MAPE)
mape_V2 <- mean(abs((predicted_sat_V2 - test_ANN$Overall.Satisfaction) /
test_ANN$Overall.Satisfaction)) * 100

# Compute MAPE, excluding cases where actual value is zero or close to zero
```

```
mape_V22 <- mean(abs((predicted_sat_V2 - test_ANN$Overall.Satisfaction) /
pmax(test_ANN$Overall.Satisfaction, 1e-10))) * 100

# R-squared (R2)
r_squared_V2 <- 1 - sum((test_ANN$Overall.Satisfaction - predicted_sat_V2)^2) /
sum((test_ANN$Overall.Satisfaction - mean(test_ANN$Overall.Satisfaction))^2)

# Print the performance metrics
print(paste("Mean Absolute Error (MAE):", mae_V2))
print(paste("Root Mean Squared Error (RMSE):", rmse_V2))
print(paste("Mean Absolute Percentage Error (MAPE):", mape_V2))
print(paste("Mean Absolute Percentage Error with no zeroes (MAPE):", mape_V22))
print(paste("R-squared (R2):", r_squared_V2))
```

```
[1] "Mean Absolute Error (MAE): 0.178014897120367"
[1] "Root Mean Squared Error (RMSE): 0.238887380435124"
[1] "Mean Absolute Percentage Error (MAPE): Inf"
[1] "Mean Absolute Percentage Error with no zeroes (MAPE): 26927387354.4251"
[1] "R-squared (R2): -0.0193065459568953"
```

#ANN Level 3 Model

```
manu_model3 <- neuralnet(Overall.Satisfaction ~ Job.region.code + Season.code +
                    Temperature.code, data = train_ANN, hidden = 5)

# Mean Absolute Error (MAE)
mae3 <- mean(abs(predicted_sat3 - test_ANN$Overall.Satisfaction))

# Root Mean Squared Error (RMSE)
rmse3 <- sqrt(mean((predicted_sat3 - test_ANN$Overall.Satisfaction)^2))

# Mean Absolute Percentage Error (MAPE)
mape3 <- mean(abs((predicted_sat3 - test_ANN$Overall.Satisfaction) /
test_ANN$Overall.Satisfaction)) * 100

# Compute MAPE, excluding cases where actual value is zero or close to zero
mape33 <- mean(abs((predicted_sat3 - test_ANN$Overall.Satisfaction) /
pmax(test_ANN$Overall.Satisfaction, 1e-10))) * 100

# R-squared (R2)
r_squared3 <- 1 - sum((test_ANN$Overall.Satisfaction - predicted_sat3)^2) /
sum((test_ANN$Overall.Satisfaction - mean(test_ANN$Overall.Satisfaction))^2)

# Print the performance metrics
print(paste("Mean Absolute Error (MAE):", mae3))
print(paste("Root Mean Squared Error (RMSE):", rmse3))
print(paste("Mean Absolute Percentage Error (MAPE):", mape3))
print(paste("Mean Absolute Percentage Error with no zeroes (MAPE):", mape33))
print(paste("R-squared (R2):", r_squared3))
```

```
[1] "Mean Absolute Error (MAE): 0.175348475161211"
[1] "Root Mean Squared Error (RMSE): 0.23555364550629"
[1] "Mean Absolute Percentage Error (MAPE): Inf"
[1] "Mean Absolute Percentage Error with no zeroes (MAPE): 26115010227.3766"
[1] "R-squared (R2): 0.00894431571688448"
```

#ANN Level 3 Model with temperature as continuous predictor

```
manu_model_V3 <- neuralnet(Overall.Satisfaction ~ Job.region.code + Season.code +
                    Avg.temp, data = train_ANN, hidden = 5)

# Mean Absolute Error (MAE)
mae_V3 <- mean(abs(predicted_sat_V3 - test_ANN$Overall.Satisfaction))

# Root Mean Squared Error (RMSE)
rmse_V3 <- sqrt(mean((predicted_sat_V3 - test_ANN$Overall.Satisfaction)^2))

# Mean Absolute Percentage Error (MAPE)
mape_V3 <- mean(abs((predicted_sat_V3 - test_ANN$Overall.Satisfaction) /
test_ANN$Overall.Satisfaction)) * 100

# Compute MAPE, excluding cases where actual value is zero or close to zero
mape_V33 <- mean(abs((predicted_sat_V3 - test_ANN$Overall.Satisfaction) /
pmax(test_ANN$Overall.Satisfaction, 1e-10))) * 100

# R-squared (R2)
r_squared_V3 <- 1 - sum((test_ANN$Overall.Satisfaction - predicted_sat_V3)^2) /
sum((test_ANN$Overall.Satisfaction - mean(test_ANN$Overall.Satisfaction))^2)

# Print the performance metrics
print(paste("Mean Absolute Error (MAE):", mae_V3))
print(paste("Root Mean Squared Error (RMSE):", rmse_V3))
print(paste("Mean Absolute Percentage Error (MAPE):", mape_V3))
print(paste("Mean Absolute Percentage Error with no zeroes (MAPE):", mape_V33))
print(paste("R-squared (R2):", r_squared_V3))
```

```
[1] "Mean Absolute Error (MAE): 0.176767969040936"
[1] "Root Mean Squared Error (RMSE): 0.237334927892811"
[1] "Mean Absolute Percentage Error (MAPE): Inf"
[1] "Mean Absolute Percentage Error with no zeroes (MAPE): 26516070439.1652"
[1] "R-squared (R2): -0.00610130088159555"
```

REFERENCES

Aggarwal, C. C. (2015). *Data Mining*. Springer International Publishing. https://doi.org/10.1007/978-3-319-14142-8

Agresti, A. (2002). *Categorical Data Analysis*. Wiley. https://doi.org/10.1002/0471249688

Ahmed, R., Shaheen, S., & Philbin, S. P. (2022). The role of big data analytics and decision-making in achieving project success. *Journal of Engineering and Technology Management - JET-M*, *65*. https://doi.org/10.1016/j.jengtecman.2022.101697

Ajayi, A., Oyedele, L., Owolabi, H., Akinade, O., Bilal, M., Davila Delgado, J. M., & Akanbi, L. (2020). Deep Learning Models for Health and Safety Risk Prediction in Power Infrastructure Projects. *Risk Analysis*, *40*(10), 2019–2039. https://doi.org/10.1111/risa.13425

Alshboul, O., Shehadeh, A., Almasabha, G., & Almuflih, A. S. (2022). Extreme Gradient Boosting-Based Machine Learning Approach for Green Building Cost Prediction. *Sustainability (Switzerland)*, *14*(11). https://doi.org/10.3390/su14116651

Bailly, A., Blanc, C., Francis, É., Guillotin, T., Jamal, F., Wakim, B., & Roy, P. (2022). Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, *213*. https://doi.org/10.1016/j.cmpb.2021.106504

Bartlett, K., Blanco, J. L., Fitzgerald, B., Johnson, J., Mullin, A. L., & Ribeirinho, M. J. (2020). *Rise of the platform era: The next chapter in construction technology*.

Bilal, M., Oyedele, L. O., Kusimo, H. O., Owolabi, H. A., Akanbi, L. A., Ajayi, A. O., Akinade, O. O., & Davila Delgado, J. M. (2019). Investigating profitability performance of construction projects using big data: A project analytics approach. *Journal of Building Engineering*, *26*. https://doi.org/10.1016/j.jobe.2019.100850

Bilal, M., Oyedele, L. O., Qadir, J., Munir, K., Ajayi, S. O., Akinade, O. O., Owolabi, H. A., Alaka, H. A., & Pasha, M. (2016). Big Data in the construction industry: A review of present status, opportunities, and future trends. In *Advanced Engineering Informatics* (Vol. 30, Issue 3, pp. 500–521). Elsevier Ltd. https://doi.org/10.1016/j.aei.2016.07.001

Blanco, J. L., Rockhill, D., Sanghvi, A., & Torres, A. (2023). *From start-up to scale-up: Accelerating growth in construction technology*. https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/from-start-up-to-scale-up-accelerating-growth-in-construction-technology

Caldas, C., & Gupta, A. (2017). Critical factors impacting the performance of mega-projects. *Engineering, Construction and Architectural Management*, *24*(6), 920–934. https://doi.org/10.1108/ECAM-05-2016-0117

Cali, U., Kuzlu, M., Pipattanasomporn, M., Kempf, J., & Bai, L. (2021). Foundations of Big Data, Machine Learning, and Artificial Intelligence and Explainable Artificial Intelligence. In *Digitalization of Power Markets and Systems Using Energy Informatics*. Springer, Cham. https://doi.org/https://doi.org/10.1007/978-3-030-83301-5_6

Carrillo, P., Harding, J., & Choudhary, A. (2011). Knowledge discovery from post-project reviews. In *Construction Management and Economics* (Vol. 29, Issue 7, pp. 713–723). https://doi.org/10.1080/01446193.2011.588953

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Chandanshive, V. B., & Kambekar, A. R. (2019). Estimation of Building Construction Cost Using Artificial Neural Networks. *Journal of Soft Computing in Civil Engineering*, *3*(1), 91–107. https://doi.org/10.22115/SCCE.2019.173862.1098

Chao, L.-C., & Chien, C.-F. (2009). Estimating Project S-Curves Using Polynomial Function and Neural Networks. *Journal of Construction Engineering and Management*, *135*(3), 169–177. https://doi.org/10.1061/ASCE0733-93642009135:3169

Chen, H., Hu, S., Hua, R., & Zhao, X. (2021). Improved naive Bayes classification algorithm for traffic risk management. *Eurasip Journal on Advances in Signal Processing*, *2021*(1). https://doi.org/10.1186/s13634-021-00742-6

Cheng, M. Y., Wu, Y. W., & Wu, C. F. (2010). Project success prediction using an evolutionary support vector machine inference model. *Automation in Construction*, *19*(3), 302–307. https://doi.org/10.1016/j.autcon.2009.12.003

Clemson News. (2022, January 13). *Clemson researcher using data to find the secret of more efficient government buildings*. Clemson News. https://news.clemson.edu/clemson-researcher-using-data-to-find-the-secret-of-more-efficient-government-buildings/

Daimon, T. (2011). Box–Cox Transformation. In *International Encyclopedia of Statistical Science* (pp. 176–178). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04898-2_152

Davila Delgado, J. M., Oyedele, L., Bilal, M., Ajayi, A., Akanbi, L., & Akinade, O. (2020). Big Data Analytics System for Costing Power Transmission Projects. *Journal of Construction Engineering and Management*, *146*(1). https://doi.org/10.1061/(asce)co.1943-7862.0001745

Duarte, D., & Ståhl, N. (2018). Machine Learning: A Concise Overview. In *Data Science in Practice* (pp. 27–58). Springer Link. https://doi.org/10.1007/978-3-319-97556-6_3

El-Kholy, A. M. (2021). Exploring the best ANN model based on four paradigms to predict delay and cost overrun percentages of highway projects. *International Journal of Construction Management*, *21*(7), 694–712. https://doi.org/10.1080/15623599.2019.1580001

Enterprise Solutions. (2023, April 25). *How Big Data and Analytics are Transforming the Construction Industry?* Matellio - Navigating Ideas. https://www.matellio.com/blog/how-big-data-and-analytics-are-transforming-the-construction-industry/#:~:text=At%20a%20CAGR%20of%2021.3,(Source%3A%20Technavio)

Fan, C.-L. (2020). Defect Risk Assessment Using a Hybrid Machine Learning Method. *Journal of Construction Engineering and Management*, *146*(9). https://doi.org/10.1061/(ASCE)CO.1943

Fox, J., & Weisberg, S. (2019). *An R Companion to Applied Regression* (3rd ed.). Sage.

Gajjar, D., Sharma, V., Garg, S., & Davis, D. (2024). Evaluating the Impact of Region, Season and Temperature on Customer Satisfaction for Construction Coating Projects. *Journal of Facility Management Education and Research*, *7*(1). www.jfmer.org

Geyer, P., & Singaravel, S. (2018). Component-based machine learning for performance prediction in building design. *Applied Energy*, *228,* 1439–1453. https://doi.org/10.1016/j.apenergy.2018.07.011

Gibson, G. E., Wang, Y.-R., Cho, C.-S., & Pappas, M. P. (2006). What Is Preproject Planning, Anyway? *Journal of Management in Engineering*, *22*(1), 35–42. https://doi.org/10.1061/ASCE0742-597X200622:135

Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2019). *Machine Learning Algorithms for Construction Projects Delay Risk Prediction*. https://doi.org/10.1061/(ASCE)

Gong, J., Caldas, C. H., & Gordon, C. (2011). Learning and classifying actions of construction workers and equipment using Bag-of-Video-Feature-Words and

Bayesian network models. *Advanced Engineering Informatics*, *25*(4), 771–782. https://doi.org/10.1016/j.aei.2011.06.002

Gong, P., Guo, H., Huang, Y., & Guo, S. (2020). Safety risk evaluations of deep foundation construction schemes based on imbalanced data sets. *Journal of Civil Engineering and Management*, *26*(4), 380–395. https://doi.org/10.3846/jcem.2020.12321

Gu, S. (2023). Construction Cost Index Prediction Based on Machine Learning. *International Conference on Applied Intelligence and Sustainable Computing, ICAISC 2023*. https://doi.org/10.1109/ICAISC58445.2023.10199679

Gunduz, M., & Tehemar, S. R. (2020). Assessment of delay factors in construction of sport facilities through multi criteria decision making. *Production Planning and Control*, *31*(15), 1291–1302. https://doi.org/10.1080/09537287.2019.1704903

Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00305-w

Hashemi, S. T., Ebadati, O. M., & Kaur, H. (2019). A hybrid conceptual cost estimating model using ANN and GA for power plant projects. *Neural Computing and Applications*, *31*(7), 2143–2154. https://doi.org/10.1007/s00521-017-3175-5

Hashemi, S. T., Ebadati, O. M., & Kaur, H. (2020). Cost estimation and prediction in construction projects: a systematic review on machine learning techniques. In *SN Applied Sciences* (Vol. 2, Issue 10). Springer Nature. https://doi.org/10.1007/s42452-020-03497-1

Hsu, M. F. (2019). Integrated multiple-attribute decision making and kernel-based mechanism for risk analysis and evaluation. *Journal of Intelligent and Fuzzy Systems*, *36*(3), 2895–2905. https://doi.org/10.3233/JIFS-171366

Hwang, S. (2009). Dynamic Regression Models for Prediction of Construction Costs. *Journal of Construction Engineering and Management*, *135*(5), 360–367. https://doi.org/10.1061/ASCECO.1943-7862.0000006

Jaber, F. K., Jasim, N. A., & Al-Zwainy, F. M. S. (2020). Forecasting techniques in construction industry: Earned value indicators and performance models. *Scientific Review Engineering and Environmental Sciences*, *29*(2), 234–243. https://doi.org/10.22630/PNIKS.2020.29.2.20

Jiang, Q. (2020). Estimation of construction project building cost by back-propagation neural network. *Journal of Engineering, Design and Technology*, *18*(3), 601–609. https://doi.org/10.1108/JEDT-08-2019-0195

Jiang, X., & Mahadevan, S. (2008). Bayesian Probabilistic Inference for Nonparametric Damage Detection of Structures. *Journal of Engineering Mechanics*, *134*(10), 820–831. https://doi.org/10.1061/ASCE0733-93992008134:10820

Karamizadeh, S., Abdullah, S. M., Halimi, M., Shayan, J., & Rajabi, M. javad. (2014). Advantage and Drawback of Support Vector Machine Functionality. *IEEE 2014 - International Conference on Computer, Communications, and Control Technology (I4CT)*, 63–65. https://doi.org/10.1109/I4CT.2014.6914146

Kifokeris, D., & Xenidis, Y. (2019). Risk source-based constructability appraisal using supervised machine learning. *Automation in Construction*, *104*, 341–359. https://doi.org/10.1016/j.autcon.2019.04.012

Kim, H., Soibelman, L., & Grobler, F. (2008). Factor selection for delay analysis using Knowledge Discovery in Databases. *Automation in Construction*, *17*(5), 550–560. https://doi.org/10.1016/j.autcon.2007.10.001

Kolltveit, B. J., & Grønhaug, K. (2004). The importance of the early phase: The case of construction and building projects. *International Journal of Project Management*, *22*(7), 545–551. https://doi.org/10.1016/j.ijproman.2004.03.002

Kononenko, I., & Kukar, M. (2007). *Machine Learning and Data Mining*. Woodhead Publishing Limited. https://doi.org/10.1533/9780857099440

Lee, S. H. (2001). *Discriminant function analysis for categorization of best practices* [The University of Texas at Austin]. http://libproxy.clemson.edu/login?url=https://www.proquest.com/dissertations-theses/discriminant-function-analysis-categorization/docview/304721101/se-2?accountid=6167

Lee, S. H., & Son, J. (2021). Development of a safety management system tracking the weight of heavy objects carried by construction workers using fsr sensors. *Applied Sciences (Switzerland)*, *11*(4), 1–15. https://doi.org/10.3390/app11041378

Li, Z., Zhong, X., & Cui, Z. (2018). Evaluating forecasting algorithm of realistic datasets based on machine learning. *ACM International Conference Proceeding Series, Part F137692*, 72–76. https://doi.org/10.1145/3194206.3194238

Ling, F. Y. Y., Chan, S. L., Chong, E., & Ee, L. P. (2004). Predicting Performance of Design-Build and Design-Bid-Build Projects. *Journal of Construction Engineering and Management*, *130*(1), 75–83. https://doi.org/10.1061/ASCE0733-93642004130:175

Love, P. E. D., & Teo, P. (2017). Statistical Analysis of Injury and Nonconformance Frequencies in Construction: Negative Binomial Regression Model. *Journal of Construction Engineering and Management*, *143*(8). https://doi.org/10.1061/(asce)co.1943-7862.0001326

Lowe, D. J., Emsley, M. W., & Harding, A. (2006). Predicting Construction Cost Using Multiple Regression Techniques. *Journal of Construction Engineering and Management*, *132*(7), 750–758. https://doi.org/10.1061/ASCE0733-93642006132:7750

Mahalakshmi, G., & Rajasekaran, C. (2019). Early Cost Estimation of Highway Projects in India Using Artificial Neural Network. In B. B. Das & N. Neithalath (Eds.), *Sustainable Construction and Building Materials* (Vol. 25, pp. 659–672). Springer Nature Singapore Pte Ltd. https://doi.org/https://doi.org/10.1007/978-981-13-3317-0_59

Mansoor, A., Liu, S., Ahmed Bouferguene, ;, Al-Hussein, M., & Asce, M. (2024). *Crane Signalman Hand-Signal Classification Framework Using Sensor-Based Smart Construction Glove and Machine-Learning Algorithms*. https://doi.org/10.1061/JCEMD4

Meyers, L. S., Gamst, G., & Guarino, A. J. (2017a). *Applied multivariate research: Design and interpretation* (3rd ed.). Sage.

Meyers, L. S., Gamst, G., & Guarino, A. J. (2017b). *Applied Multivariate Research: Design and Interpretation, 3rd Edition* (3rd ed.). Sage.

Mishra, P., Pandey, C. M., Singh, U., Gupta, A., Sahu, C., & Keshri, A. (2019). Descriptive statistics and normality tests for statistical data. *Annals of Cardiac Anaesthesia*, *22*(1), 67–72. https://doi.org/10.4103/aca.ACA_157_18

Mocanu, E., Nguyen, P. H., Gibescu, M., & Kling, W. L. (2016). Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks*, *6*, 91–99. https://doi.org/10.1016/j.segan.2016.02.005

Munawar, H. S., Ullah, F., Qayyum, S., & Shahzad, D. (2022). Big Data in Construction: Current Applications and Future Opportunities. In *Big Data and Cognitive Computing* (Vol. 6, Issue 1). MDPI. https://doi.org/10.3390/bdcc6010018

Ngo, J., Hwang, B. G., & Zhang, C. (2020). Factor-based big data and predictive analytics capability assessment tool for the construction industry. *Automation in Construction*, *110*. https://doi.org/10.1016/j.autcon.2019.103042

Nguyen Van, T., & Nguyen Quoc, T. (2021). Research Trends on Machine Learning in Construction Management: A Scientometric Analysis. *Journal of Applied Science and Technology Trends*, *2*(03), 96–104. https://doi.org/10.38094/jastt203105

Pandey, T. N., Vasudev, A., Sagayanathan, D., Anjan, G., Arshad, D., & Patra, S. S. (2023). Predicting Customer Satisfaction in Brazil E-commerce: A Comparative Study of Machine Learning Techniques. *Proceedings - 4th IEEE 2023 International Conference on Computing, Communication, and Intelligent Systems, ICCCIS 2023*, 505–510. https://doi.org/10.1109/ICCCIS60361.2023.10425505

Poh, C. Q. X., Ubeynarayana, C. U., & Goh, Y. M. (2018). Safety leading indicators for construction sites: A machine learning approach. *Automation in Construction*, *93*, 375–386. https://doi.org/10.1016/j.autcon.2018.03.022

Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. *International Journal of Computer Applications*, *175*(4), 7–9. https://doi.org/10.5120/ijca2017915495

Radzi, A. R., Bokhari, H. R., Rahman, R. A., & Ayer, S. K. (2019). *Key Attributes of Change Agents for Successful Technology Adoptions in Construction Companies: A Thematic Analysis*.

Rahman, A., & Smith, A. D. (2018). Predicting heating demand and sizing a stratified thermal storage tank using deep learning algorithms. *Applied Energy*, *228*, 108–121. https://doi.org/10.1016/j.apenergy.2018.06.064

Rahman, A., Srikumar, V., & Smith, A. D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy*, *212*, 372–385. https://doi.org/10.1016/j.apenergy.2017.12.051

Sammut, C., & Webb, G. I. (2011a). Accuracy. In *Encyclopedia of Machine Learning* (pp. 9–10). Springer US. https://doi.org/10.1007/978-0-387-30164-8_3

Sammut, C., & Webb, G. I. (2011b). Mean Absolute Error. In *Encyclopedia of Machine Learning* (pp. 652–652). Springer US. https://doi.org/10.1007/978-0-387-30164-8_525

Sammut, C., & Webb, G. I. (2011c). Mean Squared Error. In *Encyclopedia of Machine Learning* (pp. 653–653). Springer US. https://doi.org/10.1007/978-0-387-30164-8_528

Sanni-Anibire, M. O., Zin, R. M., & Olatunji, S. O. (2022). Machine learning model for delay risk assessment in tall building projects. *International Journal of Construction Management*, *22*(11), 2134–2143. https://doi.org/10.1080/15623599.2020.1768326

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning

Sharma, S. (1995). *Applied Multivariate Techniques*. John Wiley and Sons Inc.

Shaveta. (2023). A review on machine learning. *International Journal of Science and Research Archive*, *9*(1), 281–285. https://doi.org/10.30574/ijsra.2023.9.1.0410

Shenhar, A. J., Dvir, D., Levy, O., & Maltz, A. C. (2001). Project Success: A Multidimensional Strategic Concept. In *long range planning* (Vol. 34). www.lrpjournal.com

Shuang, Q., Liu, X., Wang, Z., & Xu, X. (2024). Automatically Categorizing Construction Accident Narratives Using the Deep-Learning Model with a Class-Imbalance Treatment Technique. *Journal of Construction Engineering and Management*, *150*(9). https://doi.org/10.1061/JCEMD4.COENG-14515

Silva, G. A. . S. K., Warnakulasuriya, B. N. F., & Arachchige, B. J. H. (2016). Criteria for Construction Project Success: A Literature Review. *13th International Conference on Business Management*, 697–717. https://ssrn.com/abstract=2910305Electroniccopyavailableat:https://ssrn.com/abstract=2910305Electroniccopyavailableat:https://ssrn.com/abstract=2910305

Tabachnick, B. G., & Fidell, L. S. (2013). *Using Multivariate Statistics* (Sixth). Pearson Education, Inc.

Tariq, J., & Gardezi, S. S. S. (2023). Study the delays and conflicts for construction projects and their mutual relationship: A review. In *Ain Shams Engineering Journal* (Vol. 14, Issue 1). Ain Shams University. https://doi.org/10.1016/j.asej.2022.101815

The Weather Company. (1995). *Weather Underground*. https://www.wunderground.com/

Tijanić, K., Car-Pušić, D., & Šperac, M. (2020). Cost estimation in road construction using artificial neural network. *Neural Computing and Applications*, *32*(13), 9343–9355. https://doi.org/10.1007/s00521-019-04443-y

Ting, K. M. (2011a). Confusion Matrix. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 209–209). Springer US. https://doi.org/10.1007/978-0-387-30164-8_157

Ting, K. M. (2011b). Error Rate. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 331–331). Springer US. https://doi.org/10.1007/978-0-387-30164-8_262

Tixier, A. J. P., Hallowell, M. R., Rajagopalan, B., & Bowman, D. (2016). Application of machine learning to construction injury prediction. *Automation in Construction*, *69*, 102–114. https://doi.org/10.1016/j.autcon.2016.05.016

Tu, J. V. (1996). Advantages and Disadvantages of Using Artificial Neural Networks versus Logistic Regression for Predicting Medical Outcomes. *Journal of Clinical Epidemiology*, *49*(11), 1225–1231. https://doi.org/https://doi.org/10.1016/S0895-4356(96)00002-9

United States Census Bureau. (2013). *Census Regions and Divisions of the United States*.

Wang, Y. R., Yu, C. Y., & Chan, H. H. (2012a). Predicting construction cost and schedule success using artificial neural networks ensemble and support vector machines classification models. *International Journal of Project Management*, *30*(4), 470–478. https://doi.org/10.1016/j.ijproman.2011.09.002

Wang, Y. R., Yu, C. Y., & Chan, H. H. (2012b). Predicting construction cost and schedule success using artificial neural networks ensemble and support vector machines classification models. *International Journal of Project Management*, *30*(4), 470–478. https://doi.org/10.1016/j.ijproman.2011.09.002

Xanthopoulos, P., Pardalos, P. M., & Trafalis, T. B. (2013). Linear Discriminant Analysis. In *Robust Data Mining* (pp. 27–33). Springer. https://doi.org/10.1007/978-1-4419-9878-1_4

Xia, X., Xiang, P., Khanmohammadi, S., Gao, T., & Arashpour, M. (2024). Predicting Safety Accident Costs in Construction Projects Using Ensemble Data-Driven Models. *Journal of Construction Engineering and Management*, *150*(7). https://doi.org/10.1061/JCEMD4.COENG-14397

Yousif, O. S., Zakaria, R. B., Aminudin, E., Yahya, K., Mohd Sam, A. R., Singaram, L., Munikanan, V., Yahya, M. A., Wahi, N., & Shamsuddin, S. M. (2021). Review of Big Data Integration in Construction Industry Digitalization. In *Frontiers in Built Environment* (Vol. 7). Frontiers Media S.A. https://doi.org/10.3389/fbuil.2021.770496

Yu, T., Liang, X., & Wang, Y. (2020). Factors Affecting the Utilization of Big Data in Construction Projects. *Journal of Construction Engineering and Management*, *146*(5). https://doi.org/10.1061/(asce)co.1943-7862.0001807

Zhang, J., Zi, L., Hou, Y., Deng, D., Jiang, W., & Wang, M. (2020). A C-BiLSTM approach to classify construction accident reports. *Applied Sciences (Switzerland)*, *10*(17). https://doi.org/10.3390/APP10175754

Zhou, Z., Irizarry, J., & Li, Q. (2014). Using network theory to explore the complexit of subway construction accident network (SCAN) for promoting safety management. *Safety Science*, *64*, 127–136. https://doi.org/10.1016/j.ssci.2013.11.029