PREDICTING THE OCCURRENCE OF GROUND DELAY PROGRAMS AND THEIR IMPACT ON AIRPORT AND FLIGHT OPERATIONS

A Dissertation
Presented to
The Academic Faculty

By

Eugene Mangortey

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Aerospace Engineering

Georgia Institute of Technology

May 2019

PREDICTING THE OCCURRENCE OF GROUND DELAY PROGRAMS AND THEIR IMPACT ON AIRPORT AND FLIGHT OPERATIONS

Approved by:

Dr. Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Olivia Pinon Fischer
School of Aerospace Engineering
Georgia Institute of Technology

Mr. Mike Paglione
Tech Center
Federal Aviation Administration

Mr. Tom Tessitore
Tech Center
Federal Aviation Administration

Dr. Tejas Puranik
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: April 5, 2019

Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do.

Pele

I dedicate this thesis to my parents. Thank you for your love and support!

ACKNOWLEDGEMENTS

When I applied to the Georgia Institute of Technology in the fall of 2012 to pursue my Bachelor's degree, I never expected to present my Master's thesis at the same institution. The past six years have been filled with numerous memories and opportunities. Even though my journey at Georgia Tech is not over, I have made it this far through the encouragement and support of many people.

First and foremost, I would like to express my deepest gratitude to my family. I am the man I am today because of your unconditional love and support.

Second, I would like to thank my advisor, Dr. Dimitri Mavris. You and the Aerospace Systems Design Laboratory have provided me with numerous opportunities to grow and learn, not just as an engineer but also as an individual. Research opportunities in the lab provided me with the opportunity to learn about big data analytics and its numerous applications in the aerospace industry, leading to this thesis.

I would also like to thank Dr. Olivia Pinon Fischer. You are one of the most hard-working people I have met and have been a great source of inspiration, encouragement and direction since my first year of graduate school. Thank you for supporting me throughout this entire process.

Special thanks goes to Mike Paglione and Tom Tessitore, with the Federal Aviation Administration's Technical Center. The opportunity to intern last summer helped shape this thesis. Special thanks also goes to Dr. Tejas Puranik. Thank you for accepting to be on my thesis committee and for providing valuable feedback. I would also like to thank Anya Berges, David Chong, and Robert Liguori for your technical support during my internship and throughout this process. I appreciate your invaluable input and support.

Finally, I would like to thank my colleagues and friends, particularly Joseph Brucculeri. Thanks for being a great support system!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

A flight is delayed when it arrives 15 or more minutes later than scheduled. Delays attributed to the National Airspace System are one of the most common delays and can be caused by the initiation of Traffic Management Initiatives (TMI) such as Ground Delay Programs (GDP). A Ground Delay Program is implemented to control air traffic volume to an airport over a lengthy period when traffic demand is projected to exceed the airport's acceptance rate due to conditions such as inclement weather, volume constraints, closed runways or equipment failures. Ground Delay Programs cause flight delays which affect airlines, passengers, and airport operations. Consequently, various efforts have been made to reduce the impacts of Ground Delay Programs by predicting their occurrence or the optimal time for initiating Ground Delay Programs. However, a few research gaps exist. First, most of the previous efforts have focused on only weather-related Ground Delay Programs, ignoring other causes such as volume constraints and runway-related incidents. Second, there has been limited benchmarking of Machine Learning techniques to predict the occurrence of Ground Delay Programs. Finally, little to no work has been conducted to predict the impact of Ground Delay Programs on flight and airport operations such as their duration, flight delay times, and taxi-in time delays.

This research addresses these gaps by 1) fusing data from a variety of datasets (Traffic Flow Management System (TFMS), Aviation System Performance Metrics (ASPM), and Automated Surface Observing Systems (ASOS)) and 2) leveraging and benchmarking Machine Learning techniques to develop prediction models aimed at reducing the impacts of Ground Delay Programs on flight and airport operations. These models predict 1) flight delay times due to a Ground Delay Program, 2) the duration of a Ground Delay Program, 3) the impact of a Ground Delay Program on taxi-in time delays, and 4) the occurrence of Ground Delay Programs.

Evaluation metrics such as Mean Absolute Error, Root mean Squared Error, Correla-

tion, and R-square revealed that Random Forests was the optimal Machine Learning technique for predicting flight delay times due to Ground Delay Programs, the duration of Ground Delay Programs, and taxi-in time delays during a Ground Delay Program. On the other hand, the Kappa Statistic revealed that Boosting Ensemble was the optimal Machine learning technique for predicting the occurrence of Ground Delay Programs.

The aforementioned prediction models may help airlines, passengers, and air traffic controllers to make more informed decisions which may lead to a reduction in Ground Delay Program related-delays and their impacts on airport and flight operations.

CHAPTER 1

INTRODUCTION

1.1 Flight Delays

Delays are an important indicator of the performance of any transportation system [1]. The number and duration of delays directly and indirectly affect consumers and transportation service providers one way or another. In the aviation sector, a flight is considered to be delayed when it arrives 15 or more minutes later than scheduled [2]. Flight delays can be attributed to [3]:

- Air Carrier: These delays are due to circumstances within the airline's control such as maintenance or crew issues
- Security: These delays are caused by the evacuation of a terminal or concourse, re-boarding of aircraft because of a security breach, an inoperative screening equipment and long lines in excess of 29 minutes at screening areas
- Late Arriving Aircraft: These delays are due to the previous flight with the same aircraft arriving late which causes the present flight to depart late
- Cancelled: A "cancelled" flight is a flight that was not operated, but was in the carrier's computer reservation system within 7 days of the flight's scheduled departure
- Diverted: A "diverted" flight is a flight which is operated from the scheduled origin point to a point other than the scheduled destination point in the carrier's published schedule
- National Airspace System: Delays and cancellations attributable to the National Airspace System refer to a broad set of conditions – non-extreme weather conditions, airport operations, heavy traffic volume, air traffic control, etc.

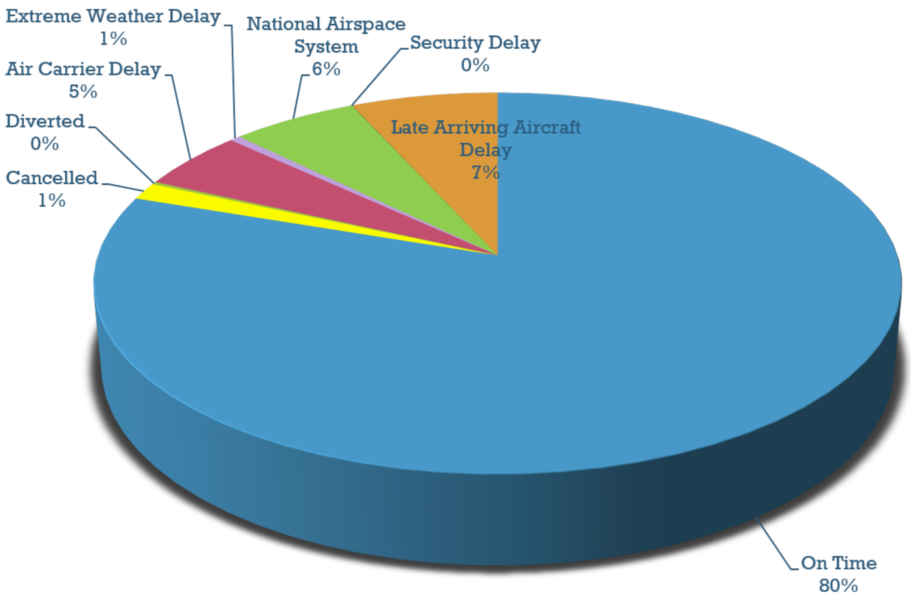

Figure 1.1: Flight Delay Statistics (2017) [4]

Figure 1.1 shows the proportion of on-time flights compared to the different types of delays that occurred in 2017. From Figure 1.1, it can be seen that 80% of flights arrived on time while late arriving aircraft and National Airspace System delays were the highest causes of flight delays. These delays have a significant impact on airlines, passengers, and the United States economy. It is thus important to understand the economic cost of flight delays and to initiate efforts to reduce their impact on airlines, passengers, and the United States economy.

1.2 Economic Cost of Delays

Table 1.1 shows the overall percentage of flight operations by major US carriers that arrived in May 2018 at the 30 largest airports and at all airports in the United States. In particular, it shows that over 20% of flights from major US carriers were delayed for one reason or another which led to increased operational costs for airlines. These costs include but are not limited to fuel, maintenance, crew, and aircraft. Flight delays also lead to loss in

Table 1.1: Overall percentage of reported flight operations arriving on time by carrier (May 2018) [6]

Carrier	Arrivals on time at 30 largest airports (%)	Arrivals on time at all airports (%)
Delta airlines	85.0	85.0
Alaska Airlines	79.7	81.8
Spirit Airlines	79.8	80.3
United Airlines	78.9	78.9
American Airlines	79.0	78.6
Southwest Airlines	76.0	76.4
Frontier Airlines	70.6	71.8
JetBlue Airways	70.5	71.0
TOTAL	77.4	77.9

productivity and business opportunities for business travelers, as well as an opportunity cost of time for leisure passengers. The effects of flight delays on airlines and passengers also have an indirect impact on the economy. Delays may lead to increased fuel costs which lead to increased airfares. Increased airfares as well as delays in general may lead to changes in consumer spending on travel, and tourism good and services [5], which eventually impacts the economy.

Figure 1.2 provides a breakdown of the direct costs of air transportation delays in terms of passengers, airlines, lost demand, and impact on the Gross Demand Product (GDP) of the United States in 2007 dollars. The \$8.3 billion airline component is comprised of increased operating costs for crew, fuel, maintenance, etc. The \$16.7 billion passenger component is comprised of the passenger time lost due to schedule buffers, delayed flights, flight cancellations, and missed connections. The \$3.9 billion cost associated with lost demand represents an estimate of the time or productivity loss incurred by passengers who avoid air travel as a result of delays. As discussed, in addition to the direct costs of flight delays on airlines and passengers, flight delays have indirect effects on the U.S. economy. Indeed, inefficiencies in the aviation industry may lead to increased cost of doing business

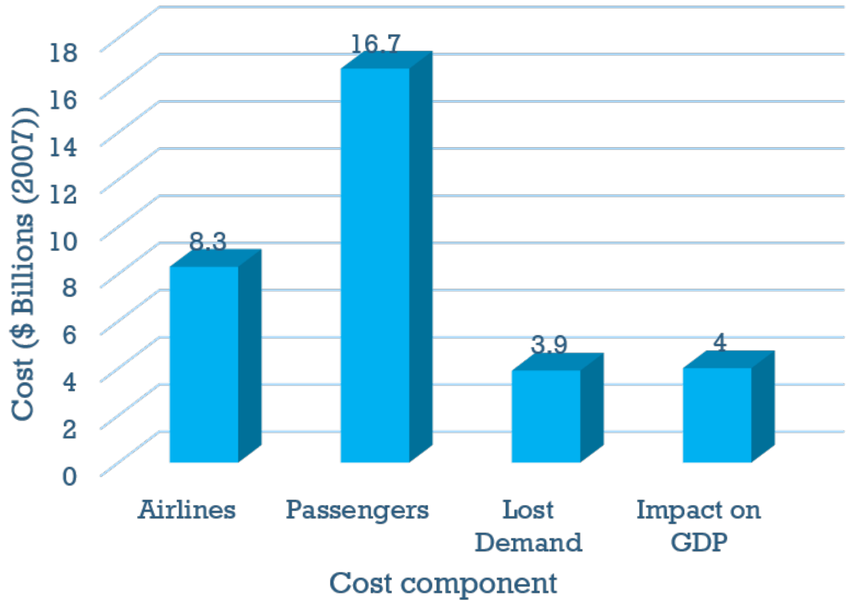

Figure 1.2: Direct cost of air transportation delays (2007 dollars) [7]

for other industries, making the associated businesses less productive [7].

1.3 National Airspace System Delays

As seen in Figure 1.1, late arriving aircraft were the highest causes of flight delays in 2017. However, it is important to note that late arriving aircraft may have been affected by National Airspace System delays or other delays on prior trips leading to that delay. Thus, the proportion of flights affected by National Airspace System delays may be higher. Consequently, the work covered in this research will focus on delays caused by the National Airspace System. The National Airspace System is comprised of air navigation facilities, equipment, airports or landing areas, aeronautical charts, information, services, rules, regulations, procedures, technical information, manpower, and materials [8]. Figure 1.3 shows the causes of National Airspace Delays in 2017. From this figure, it can be seen that inclement weather caused 53% of delays associated with the National Airspace System, followed by volume constraints and closed runways. As discussed previously, flight delays are costly to airlines, passengers, and the United States economy. Thus, efforts have been

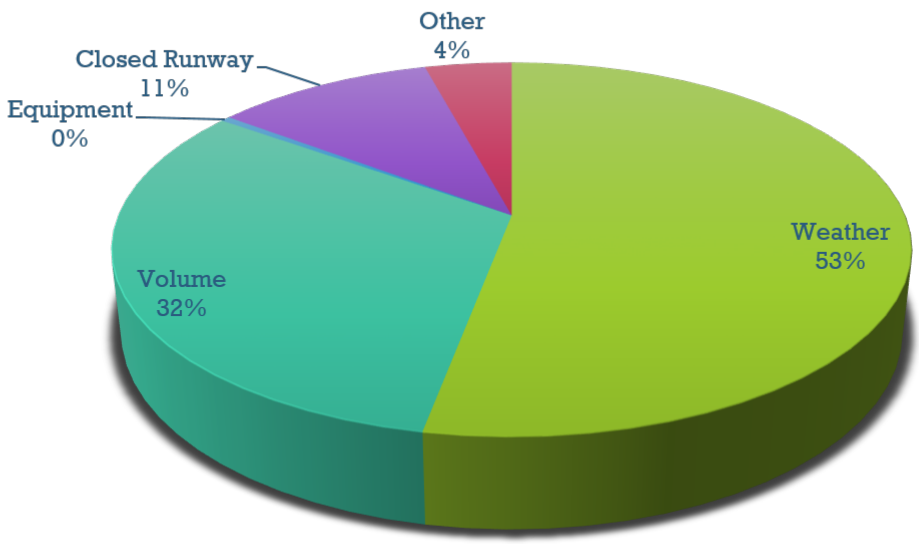

Figure 1.3: Causes of National Airspace System Delays (2017) [9]

made by stakeholders in the aviation industry to reduce the incidence and impacts of flight delays attributed to the National Airspace System.

1.4 Efforts Towards Reducing National Airspace System Delays

Over the years, efforts have been made to minimize delays attributed to the National Airspace System while maintaining or improving aviation safety. In 2008, the Department of Transportation developed a list of initiatives to improve air travel while reducing the impacts of lengthy delays on consumers [10]. These initiatives involved instituting caps on hourly operations at the John F. Kennedy and Newark airports. The Department of Transportation also instituted other measures such as negotiating an agreement with the Department of Defense to open up military airspace for commercial use during the holiday season to reduce the duration and number of flight delays. These actions have proven to be particularly successful [10].

In August 2000, the United States Department of Transportation formed the Air Carrier On-Time Reporting Advisory Committee to consider changes to the on-time reporting

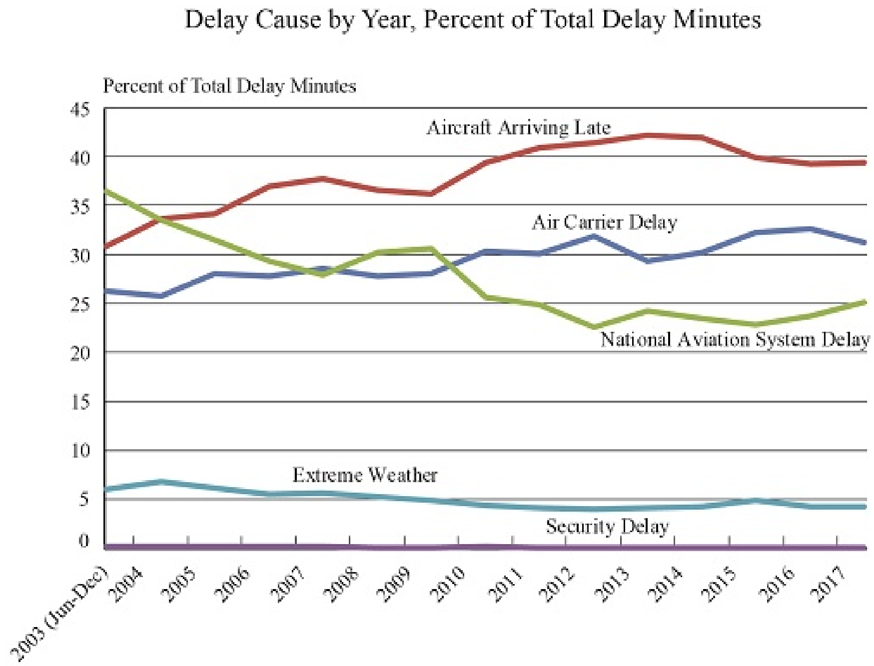

Figure 1.4: Delay Cause by Year (June 2003 - December 2017) [14]

system and provide the public with clear information about the nature and sources of airline delays and cancellations. In 2001, the Bureau of Transportation Statistics conducted a pilot program with four airlines to test the monthly reporting of causes of delay [11]. In November 2002, the Department of Transportation issued a final rule on reporting the causes of flight delays which requires carriers to report on domestic operations to and from U.S. airports, starting June 2003. Currently, air carriers that have 0.5% of total domestic scheduled-service passenger revenue have to report on-time data and the causes of delay. In 2018, this represented 18 air carriers. These reports cover nonstop scheduled-service flights between points within the United States (including territories) [12].

The above mentioned initiatives along with others, such as the Aviation Safety Information Analysis and Sharing (ASIAS) initiative have contributed to decreasing NAS-related delays since 2003 [13], as seen in Figure 1.4. However, much more needs to be done to further reduce delays attributed to Traffic Management Initiatives such as Ground Delay Programs implemented in the National Airspace System.

1.5 Ground Delay Programs (GDP)

Ground Delay Programs (GDP) are Traffic Management Initiatives (TMI) that are initiated when aircraft demand is projected to exceed airport capacity over a long period of time due to conditions such as inclement weather, volume constraints, runway closures, equipment failures etc [15, 16].

Whenever Ground Delay Programs are issued, Traffic Management Personnel use the Enhanced Traffic Management System (ETMS) to predict, on national and local scales, traffic surges, gaps, and volume based on current and anticipated airborne aircraft [17]. This is done by evaluating the projected flow of traffic into airports and sectors, then implementing the least restrictive action necessary to ensure that traffic demand does not exceed system capacity. During Ground Delay Programs, Expected Departure Clearance Times (EDCT) are issued to affected flights. EDCT is the runway release time ("Wheels Off") assigned to aircraft due to Traffic Management Initiatives (TMI) that require holding aircraft on the ground at the departure airport [18]. EDCT are updated whenever conditions improve to reduce delay durations. Figure 1.5 shows a breakdown of the different causes of Ground Delay Programs in 2017.

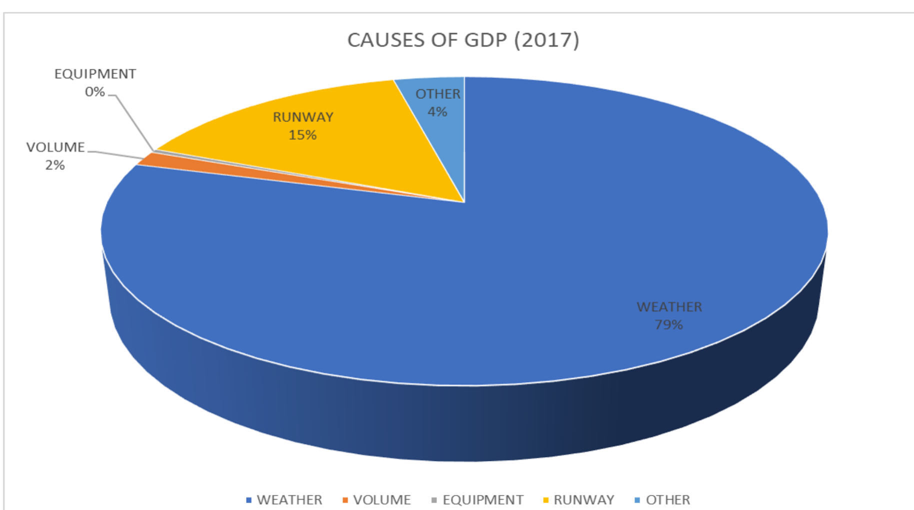

Figure 1.5: Causes of Ground Delay Programs (2017) [19]

1.6 Research objectives

As mentioned previously, the issuance of Ground Delay Programs cause delays which affect airlines, passengers, and airport operations. However, the impacts of such delays can be reduced by:

1.6.1 Predicting flight delay times due to a Ground Delay Program

Currently, the Federal Aviation Administration (FAA) provides the general public with information regarding Traffic Management Initiatives such as Ground Delay Programs whenever they are initiated [20]. This information includes the affected airport, the cause of the Ground Delay Program, its duration, and the maximum and average delay times during the entire Ground Delay Program. Even though maximum and average delay times during the entire duration of a Ground Delay Program provide valuable information to the public, obtaining the maximum and average delay times for each hour during a Ground Delay Program may provide more insight and much more valuable information for passengers and airlines. Consequently, **the first objective of this research involves predicting the maximum delay time per hour of a Ground Delay Program, and computing the average delay time using the number of flights scheduled to arrive during the specified hour**.

1.6.2 Predicting the duration of Ground Delay Programs

The duration and scope of Ground Delay Programs is updated whenever conditions change. Unfortunately, airlines and passengers do not know if the duration or scope of a Ground Delay Program will occur, which consequently hinders their ability to plan appropriately and efficiently. **The second objective of this research thus focuses on predicting the duration of Ground Delay Programs.** This may help airlines and passengers to make more informed decisions.

1.6.3 Predicting the impact of Ground Delay Programs on taxi-in delay times

Ground Delay Programs impact airport operations such as taxi-in times, leading to taxi-in delay times. Taxi-in delay times are defined as the difference between actual taxi-in times and unimpeded taxi-in times. This metric is typically recorded when the duration of the delay is over a minute [21]. **The third objective of this research focuses on predicting taxi-in delay times during Ground Delay Programs.**

1.6.4 Predicting the occurrence of Ground Delay Programs

The final objective of this research focuses on predicting the occurrence of weather and volume-related Ground Delay Programs. This will enable passengers, airlines, and air traffic controllers to make more informed decisions.

The remainder of this document will highlight the literature review conducted, the problem formulation process, the methodology used, the analysis of results, and concluding remarks.

CHAPTER 2
BACKGROUND

Numerous efforts have been made to predict flight delays and their durations, using various approaches [22]. However, researchers and analysts have faced a number of challenges when developing these prediction models. These challenges include, among others, understanding which data may be applicable to their work, where to access the data, and how to efficiently analyze the data.

The aviation industry generates data at record volumes, with most of the data generated and collected focusing on what happens in and around airplanes [23]. This data, characterized by the 'Four V's'[24], is commonly referred to as Big Data:

- Volume: Refers to the scale or amount of Big Data (data at rest). Large corporations typically generate, store and utilize terabytes, exabytes, petabytes and zettabytes of data
- Veracity: Refers to the accuracy of Big Data
- Velocity:Refers to how quickly streaming data (data in motion) is received and processed
- Variety: Refers to the different forms of Big Data. Big Data can be unstructured or structured. Unstructured data can be in the form of audio, video or text files, while structured data usually presents itself in databases with all features having a pretty well defined meaning

Due to the intricate nature of aviation Big Data, data analysts and researchers face a number of challenges associated with analyzing, understanding and identifying trends in aviation Big Data. Particularly, the ingestion, storage, exploration, analysis and visualization of aviation Big Data presents many challenges. Traditional approaches to delay modeling, in particular cannot analytically analyze the enormous volume of data related to traffic, impacts of inclement weather or airport-related activities (runway closures, construction etc.) that traditionally cause Ground Delay Programs. After ingesting and processing aviation Big Data, analysts and researchers need to be able to rapidly and efficiently make sense of the data that is critical to their operations. Machine learning is one approach to help address this problem. New approaches that leverage machine learning techniques have shown some promising results in their ability to predict delays using information about traffic volume, inclement weather etc.

2.1 Machine Learning

Machine Learning is a method of data analysis focused on the development of computer algorithms to transform data into useful actions [25]. Machine Learning has been widely used in various industries. Examples of machine learning applications include forecasts of weather behavior and long-term climate changes, reduction of fraudulent credit card transactions, prediction of popular election outcomes, discovery of genetic sequences linked to diseases, and image recognition.

Machine Learning has also been widely used to augment specialized knowledge of subject-matter experts, and has contributed to improving data generation and aggregation processes. However, it is worth noting that as much as society has greatly benefited from it, machine learning has its limitations. Indeed, it has little flexibility to extrapolate outside of the strict parameters it learned. Thus, it is important for the model to be trained accurately and comprehensively to avoid over-fitting or under-fitting.

Machine Learning is also not an intuitive process, with Machine Learning techniques often acting as black boxes. Machine Learning techniques and their applications also rely on various assumptions. Having a clear understanding of these assumptions is thus critical to the application of any Machine Learning technique/algorithm. Finally, Machine Learning algorithms are divided into three categories: supervised learning, unsupervised learning, and meta-learners/ensembles. Understanding these categories is an integral step towards developing accurate prediction models.

2.1.1 Supervised Learning

The process of training a model that predicts one value using other values in the dataset is known as supervised learning. Models used in supervised learning are known as predictive models. In supervised learning, the algorithm attempts to discover and model the relationship between the value being predicted (target) and other values (predictors). Predictive models can be used to predict not only events in the future, but can also be used to predict previous and real-time events [26].

It is also important to note that supervised learning does not refer to human interference. Rather, it refers to the fact that target features enable learners to assess how well they have learned the desired tasks. Supervised Machine Learning algorithms can be used for two tasks: classification and numeric prediction [25]. In supervised learning, the data is labeled and the algorithms learn to predict the output from the input data.

Classification

Classification is often used to predict which class an instance belongs to. This involves mapping predictors to a target by learning how predictors are related to the target. Examples of classification tasks include predicting whether an email is spam, if an individual has cancer, if a football team will win or lose or if a delay will occur [25].

Numeric Prediction

Numeric predictions are used to predict a numeric target from a set of predictors. The targets are continuous because there are no discontinuities or gaps in the values that they can take. Examples of numeric prediction tasks include predicting income, test scores or counts of items [25].

2.1.2 Unsupervised Learning

The process of training a model that benefits from the insight gained from summarizing data in new and interesting ways is known as unsupervised learning. Models used in unsupervised learning are known as descriptive models. As opposed to predictive models that predict target values, no single feature is more important than the others in a descriptive model. Unsupervised Machine Learning algorithms can be used for tasks such as pattern discovery and clustering [25]. In unsupervised learning, the data is unlabeled and the algorithms learn to understand the structure of the the data.

Pattern Discovery

Pattern discovery is used to identify useful associations within data by extracting knowledge from databases without prior knowledge of existing patterns within the data. An example of a pattern discovery task is identifying goods that are frequently purchased together [25].

Clustering

Clustering or Segmentation Analysis is often used to divide a dataset into groups. It is widely used to explore data in order to identify hidden patterns in data. Clusters are typically modeled using a measure of similarity which is defined by metrics such as probabilistic distance. An example of a clustering task is identifying groups of individuals with similar behavior for an advertising campaign [25].

2.1.3 Meta-learning/Ensembles

Meta-learning algorithms use the results of some learning to inform additional learning. This can be useful for challenging problems or when a predictive algorithm's performance needs to be as accurate as possible [25]. Meta-learning algorithms include bagging, boosting and random forests. One of the benefits of using meta-learners or ensembles is that they may allow a user to spend less time in pursuit of a single best model. Instead, a number of algorithms can be trained and combined together.

Table 2.1 highlights a breakdown of different Machine Learning algorithms and their learning tasks. A description of the algorithms presented in Table 2.1 can be found in Appendix A.

Table 2.1: Machine Learning Algorithms And Their Learning Tasks [25]

Algorithm	Learning Task
Nearest neighbor	Classification
Naive Bayes	Classification
Decision Trees	Classification
Classification Rule Learners	Classification
Linear Regression	Numeric Prediction
Regression Trees	Numeric Prediction
Model Trees	Numeric Prediction
Neural Networks	Dual use
Support Vector Machines	Dual use
Association Rules	Pattern detection
k-means clustering	Clustering
Bagging	Dual use
Boosting	Dual use
Random forests	Dual use

The remainder of this chapter will focus on discussion relevant past studies and identifying limitations in their application of Machine Learning techniques in predicting Ground Delay Programs.

2.2 Review of Past Research

2.2.1 Predicting Ground Delay Program At An Airport Based On Meteorological Conditions

This effort involved developing two models to predict the occurrence of a Ground Delay Program at an airport based on meteorological conditions using Logistic Regression and Decision Trees. The models were developed with two major U.S. airports as test cases: Newark Liberty and San Francisco International airports, using meteorological conditions and traffic demand at that hour. Meteorological conditions were extracted from the Rapid Update Cycle (RUC) database. Traffic demand data was obtained from FAA's Aviation Systems Performance Metrics (ASPM) database. Data on GDP occurrence at an airport was obtained from FAA's National Traffic Management Log (NTML) database. Results from the models indicated that the Logistic Regression model performed better than the Decision Tree in predicting Ground Delay Programs. Even though both models performed well, there is a need to expand the scope of Ground Delay Program prediction models to include other causes such as volume constraints [27].

2.2.2 Decision Support Tool for Predicting Aircraft Arrival Rates, Ground Delay Programs, and Airport Delays from Weather Forecasts

This effort involved developing a decision support tool to predict future airport capacities using Support Vector Machines. Terminal Aerodrome Forecast (TAF) was used as an independent variable within a Support Vector Machine to predict Aircraft Arrival Rates (AAR) as a proxy for airport capacity. Within the decision support tool, the Aircraft Arrival Rates were then used to determine Ground Delay Program (GDP) program rates and duration, as well as passenger delay. Unfortunately, the data was over-fitted and this impacted the model's performance [28].

2.2.3 Ground Delay Program Planning Under Uncertainty in Airport Capacity

This effort involved developing an optimization algorithm to assign flight departure delays under probabilistic airport capacity. The algorithm dynamically adapted to weather forecasts by revising, when necessary, departure delays. San Francisco International Airport served as a use case. The algorithm was applied to assign departure delays to flights scheduled to arrive in the presence of uncertainty during the fog clearance time. Weather forecasts were obtained from an ensemble forecast system for predicting fog burn-off time developed by the National Weather Service (NWS) and MIT Lincoln Labs. Experimental results indicated that overall delays at the San Francisco International Airport could be reduced by up to 25%. However, the work did not include other weather-related conditions in the prediction models nor did it include other causes of Ground Delay Programs [29].

2.2.4 Optimizing Key Parameters of Ground Delay Program with Uncertain Airport Capacity

This effort involved developing a framework to optimize key parameters of Ground Delay Programs such as file time, end time, and distance using a genetic algorithm. The model calculated the optimal Ground Delay Program file time, which was estimated to significantly reduce the delay times. Results showed that, in comparison with actual Ground Delay Programs that occurred, the proposed framework reduced the total delay time, unnecessary ground delay, and unnecessary ground delay flights by 14.7%, 50.8%, and 48.3%, respectively. However, while it is important to predict the optimal GDP file time, predicting the duration of Ground Delay Programs may be much more useful to airlines and passengers [30].

2.2.5 Predicting the initiation of a Ground Delay Program

This effort involved developing a prediction model to support decision making in initiating Ground Delay Programs, and quantifying their impact using Logistic Regression. This research aimed to predict the initiation of a Ground Delay Programs for flight operators.

Results showed that whatever the time horizon, the model's predictions were often incorrect, either predicting a Ground Delay Program when one was not implemented or vice versa. Benchmarking of Machine Learning techniques would have helped in identifying a set of suitable techniques for this model [31].

2.2.6 Ground Delay Program Analytics with Behavioral Cloning and Inverse Reinforcement Learning

This effort involved developing two models to predict Ground Delay Program implementation decisions and provide insights into how and why those decisions were made. These models were developed using behavioral cloning and inverse reinforcement to predict hourly Ground Delay Program implementation at Newark Liberty International and San Francisco International airports. Scheduled flight arrival times and the state of airports were extracted from the FAA's Aviation System Performance Metrics (ASPM) database. Weather data was extracted from Terminal Aerodrome Forecasts (TAF) and Meteorological Terminal Aviation Routine (METAR) weather reports. Results showed that the behavioral cloning model was substantially better than the inverse reinforcement learning models at predicting hourly Ground Delay Program implementation for these airports. However, the models struggled to predict the initialization and cancellation of Ground Delay Programs. Benchmarking of Machine Learning techniques may address these shortcomings [32].

2.2.7 Development of a Data Fusion Framework to support the Analysis of Aviation Big Data

This effort highlighted how the FAA can utilize a data fusion framework for the analysis of aviation big data. The framework was tested for the purpose of predicting the occurrence of weather-related Ground Delay Programs (GDP) at the Newark (EWR), La Guardia (LGA), and Boston Logan (BOS) International Airports. In particular, this research involved fusing data from the System-Wide Information Management (SWIM) Flight Publication Data

Service (SFDPS) [33], Traffic Flow Management System (TFMS) [34], and Meteorological Terminal Aviation Routine (METAR) [35] datasets. The prediction model was developed using Decision Trees and performed well. However, this model focused on Ground Delay Programs caused by inclement weather only. Expanding the scope of this research to include other causes of Ground Delay Programs, and benchmarking different techniques will provide much more information to aviation stakeholders [15].

2.3 Summary of Prior Research and Research Gaps

The review of prior research highlights a couple of limitations and/or gaps. First, prior work in predicting delays associated with Ground Delay Programs has mainly focused on weather-related delays. Other causes such as volume constraints and runways closures have been largely ignored, primarily due to a lack of access to data. This research aims to address these limitations by including the other causes of Ground Delay Programs as predictors in the prediction models.

Second, most efforts have not included details regarding the causes of Ground Delay Programs, primarily due, again, to a lack of access to data. This research aims to address this gap by including details of the causes of Ground Delay Programs as predictors in the prediction models. Examples of details of weather-related Ground Delay Programs include fog, low ceilings, thunderstorms etc.

Third, the type of data used has influenced the Machine Learning techniques implemented. Previous work, for example, used unsupervised data modeling techniques such as Principal Component Analysis and Clustering [36][37][38] which produced poorly performing prediction models. A lack of benchmarking to evaluate and compare the performances of different Machine Learning techniques in predicting the duration of Ground Delay Programs is also a limitation of previous and current efforts. Consequently, this research involved identifying appropriate Machine Learning techniques and benchmarking their performance in the context of the research goals and objectives discussed in section

1.6.

It is also important to note that little to no work has been conducted to predict the impact of Ground Delay Programs on flight and airport operations such as flight delay times and taxi-in delay times. These gaps were filled by this research.

Finally, weather forecasts are not a good indication of the impact that weather has on flight operations. Ignoring this has led to inaccurate or incorrect causalities being made. Previous efforts have erroneously attempted to link all flight delays to weather conditions. This limitation was addressed by using comprehensive datasets which clearly state the cause of the Traffic Management Initiative (TMI) leading to the delay.

CHAPTER 3
PROBLEM FORMULATION

Motivated by the gaps in research highlighted in the previous chapter as well as the overarching need to reduce the impacts of Ground Delay Programs, the research questions formulated below served as a guide to the overall research plan.

3.1 Research Questions and Hypotheses Development

As mentioned in Chapter 2, past studies have only focused on weather-related Ground Delay Programs, primarily due to a lack of access to data. In addition, limited benchmarking of Machine Learning techniques has been performed to identify suitable techniques for developing prediction models for Ground Delay programs. Both of these gaps will be addressed through two main research questions.

3.1.1 Research Question 1

In addition to the gaps previously mentioned, little to no work has been conducted to predict the impact of Ground Delay Programs on flight and airport operations such as their duration, flight delay times, and taxi-in delay times. Thus, the first research question is three-fold:

Research Question 1.1: *Which Machine Learning techniques would lead to accurate predictions of flight delay times due to Ground Delay Programs?*

Research Question 1.2: *Which Machine Learning techniques would lead to accurate predictions of the duration of Ground Delay Programs?*

Research Question 1.3: *Which Machine Learning techniques would lead to accurate predictions of taxi-in delay times during Ground Delay Programs?*

3.1.2 Hypotheses 1

The hypotheses for Research Questions 1.1, 1.2, and 1.3 are:

Hypothesis 1.1: *If dataset(s) containing comprehensive Ground Delay Program data are leveraged, then prediction models can be developed to predict the impact of Ground Delay Programs on flight and airport operations such as their duration, flight delay times, and taxi-in delay times.*

Hypothesis 1.2: *If numerical prediction algorithms are developed and benchmarked, then prediction models can be developed to predict the impact of Ground Delay Programs on flight and airport operations such as their duration, flight delay times, and taxi-in delay times.*

3.1.3 Research Question 2

The second research question seeks to identify Machine Learning techniques to be used in predicting the occurrence of weather and volume-related Ground Delay Programs. It was not be possible to predict the occurrence of other causes of Ground Delay Programs such as equipment failures and runway-related incidents since data on their contributing factors to equipment failures was not readily available.

Research Question 2: *Which Machine Learning technique(s) would lead to accurate predictions of the occurrence of Ground Delay Programs (GDP)?*

3.1.4 Hypotheses 2

The hypotheses for Research Question 2 are:

Hypothesis 2.1: *If dataset(s) containing comprehensive Ground Delay Program data are leveraged, then a model can be developed to predict the occurrence of Ground Delay Programs*

Hypothesis 2.2: *If classification algorithms are developed and benchmarked, then the occurrence of Ground Delay Programs can be accurately predicted*

Figure 3.1 shows the mapping of research questions to hypotheses covered in this section.

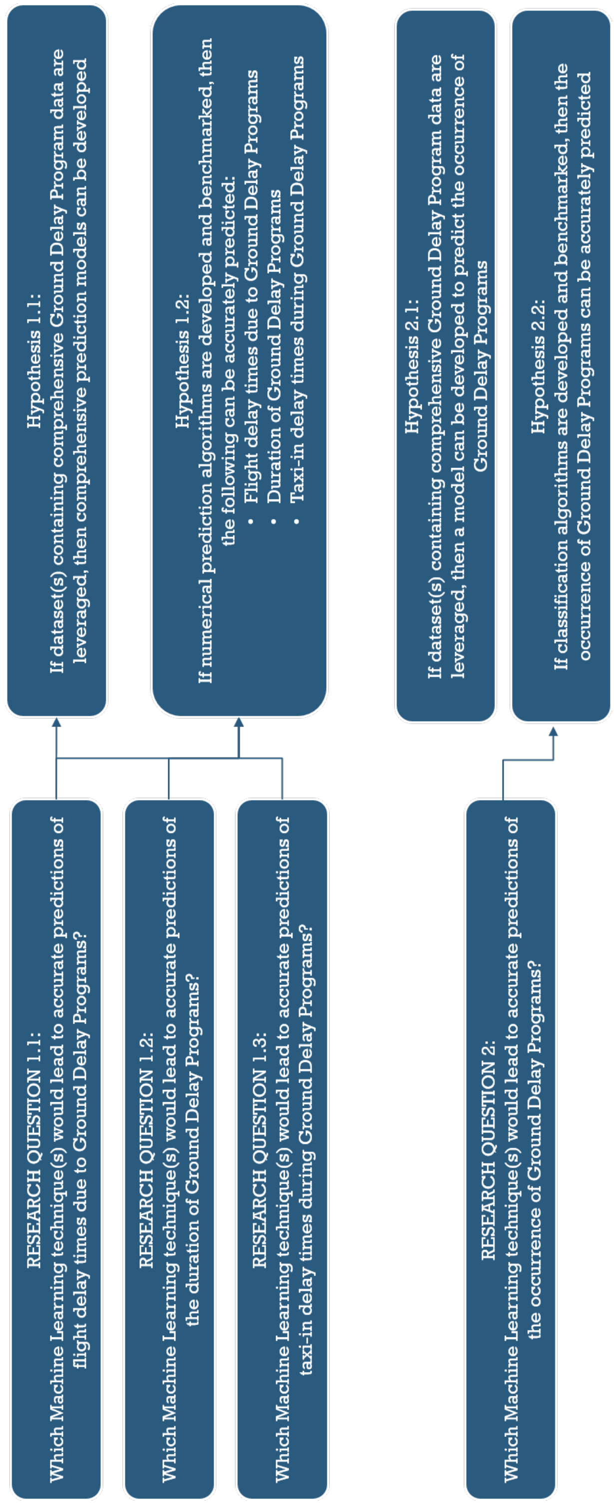

Figure 3.1: Mapping of Research Questions and Hypotheses

CHAPTER 4
METHODOLOGY

In order to successfully answer the research questions outlined in the previous chapter and their relevant hypotheses, it was important to identify and implement an efficient approach. The following steps served as a comprehensive approach:

4.1 Problem Definition

As mentioned in section 1.6, the objectives of this research were to:

1. Predict flight delay times due to Ground Delay Programs
2. Predict the duration of Ground Delay Programs
3. Predict the impact of Ground Delay Programs on taxi-in delay times
4. Predict the occurrence of Ground Delay Programs

In order to achieve the aforementioned objectives, there was a need to analyze Ground Delay Programs and their incidence across the largest airports in the United States. Figure 4.1 shows that the Newark (EWR), San Francisco (SFO), La Guardia (LGA), and Los Angeles (LAX) International Airports had the highest incidence of Ground Delay Programs in 2017. It can also be seen that Los Angeles International Airport had a good distribution of the different types of Ground Delay Programs compared to the other airports. Thus, the aforementioned prediction models were developed for the Los Angeles International Airport.

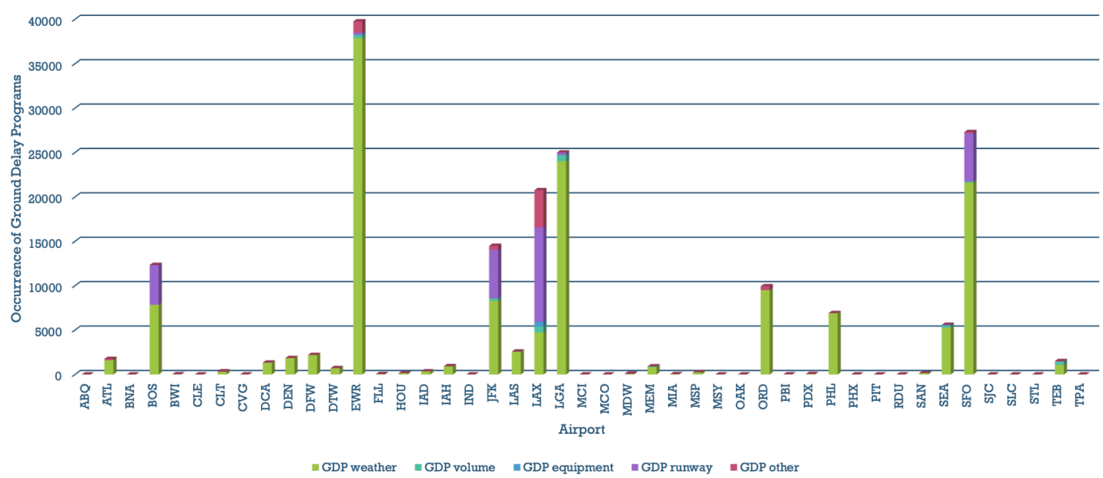

Figure 4.1: Breakdown of Ground Delay Programs by airport (2017) [39]

4.2 Dataset Identification & Acquisition

Three datasets were used to achieve the research objectives highlighted in section 1.6. The data used for this research spanned the duration, January to August 2017. The datasets used were:

1. Traffic Flow Management System (TFMS)

2. Aviation System Performance Metrics (ASPM)

3. Automated Surface Observing Systems (ASOS)

4.2.1 Traffic Flow Management System (TFMS)

The Traffic Flow Management System (TFMS) is used by air traffic management personnel to predict, on national and local scales, traffic surges, gaps, and volume based on current and anticipated airborne aircraft. TFMS provides Aircraft Situation Display (ASDI) data such as aircraft scheduling, routing and positional information. TFMS helps traffic management personnel examine a situation and provide routes and spacing to assist in controlling the flow of traffic. TFMS is comprised of two message streams: TFMS Flight and TFMS Flow [40].

TFMS Flight

The TFMS Flight message stream is comprised of the following messages:

- Flight plan data initial and subsequent amendments
- Departure and arrival time notifications
- Flight cancellations
- Boundary crossings
- Track position reports

TFMS Flow

The TFMS Flow message stream is comprised of the following messages:

- Reroutes: This provides new routes for affected aircraft
- Ground Stop (GS): This Traffic Management Initiative requires aircraft that meet specific criteria to remain on the ground
- Ground Delay Program (GDP): This Traffic Management Initiative causes aircraft to be delayed at their departure airport in order to manage demand and capacity at their arrival airport
- Airspace Flow Program (AFP): This Traffic Management Initiative is issued when volume in an area in the National Airspace System reaches a point where traffic management initiatives are not sufficient

- Collaborative Trajectories Options Program (CTOP): This Traffic Management Initiative automatically assigns delay and/or reroutes around one or more Flow Constrained Area-based airspace constraints in order to balance demand with available capacity.

- Flow Evaluation Area (FEA)/ Flow Constrained Area (FCA): Flow Evaluation Area (FEA) is a line in space that is drawn across a specific area. Traffic managers then monitor the amount of traffic crossing that line to ensure that the amount of traffic does not exceed what that volume of airspace can handle at that time. Once the amount of traffic reaches a point where it is considered to be a potential issue, the FEA becomes a Flow Constrained Area (FCA)

- Air Traffic Control System Command Center (ATCSCC) advisories: Advisories issued by ATCSCC summarize Traffic Management Initiatives such as reroutes, Ground Stops and Ground Delay Programs

- Restrictions: This provides information on areas in the airspace that may be restricted for reasons such as extreme weather or aircraft congestion

- Airport Runway Configurations and Rates: This provides updates on the capacity of airport runways

- Airport Deicing Status: This provides updates on the deicing status of runways of airports

- Route Availability Planning Tool (RAPT) timeline forecast data: This data is used to determine which departure routes need to be closed due to weather conditions and when to reopen those routes as the weather conditions ease

Figure 4.2 shows the distribution of TFMS Flow messages recorded between midnight and 1AM on April 21, 2017. It is important to note that anytime conditions change, an

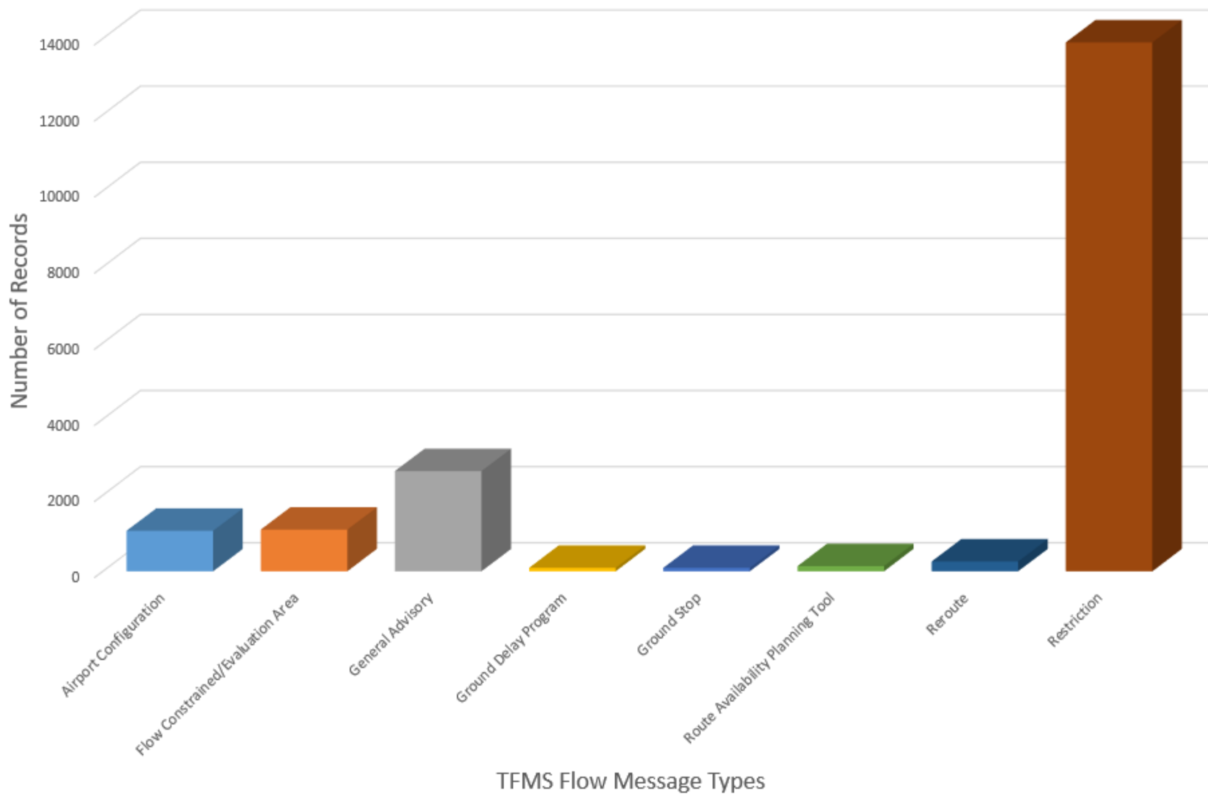

Figure 4.2: Distribution of TFMS Flow messages from midnight to 1AM on April, 21 2017

updated message is generated. Thus, approximately 14,000 restriction messages recorded within an hour could be the same message being updated whenever conditions changed.

The TFMS datasets were obtained from the FAA's Computing Analytics and Shared Services Integrated Environment (CASSIE). CASSIE is a collaborative and flexible environment for conducting research, bringing all FAA divisions, partners and stakeholders together in a shared services environment consisting of Big Data, computing power and analytical tools. CASSIE utilizes Hadoop Hortonworks for data storage and handling. Hadoop is an open-source software framework for distributed storage and processing of big data. In particular, the Hadoop Distributed File System (HDFS) allows for computer clusters to be linked robustly for high performance storage and computation [41]. Another component of Hadoop is NiFi, which automates the movement of data between disparate data sources and systems, making data ingestion fast, easy and secured [42].

4.2.2 Aviation System Performance Metrics (ASPM)

The Aviation System Performance Metrics database provides data from flights operating at 77 airports in the United States referred to as “ASPM airports” [43], flight data from 27 air carriers referred to as “ASPM carriers” [44], airport weather and runway data, and airport arrival and departure rates [45]. ASPM data used for this research was obtained from the online ASPM database in csv format. This database provides a comprehensive overview of air traffic for these airports and air carriers, and is composed of five modules: Metric, Efficiency, Enroute, Dashboards, and Other [45].

Metric

1. Airport Analysis: This metric provides a comparison of actual flight departure and arrival times, and flight plan times at ASPM airports

2. City Pair Analysis: This metric provides a comparison of actual flight departure and arrival times, and flight plan times between city pairs

3. Taxi Times: This metric provides actual and unimpeded taxi times for “ASPM airports”

4. Individual Flights: This metric provides a comparison of actual flight departure and arrival times, and flight plan times for individual flights

5. Cancellations: This metric provides data regarding cancelled flights and completion rates

Efficiency

1. Airport Efficiency: This measure provides Terminal and System Airport Efficiency data for airports

2. Throughput: This measure provides actual airport throughput (number of departures and arrivals) during a specified period of time

Enroute

1. City Pair Enroute: This measure provides average distance and time data for city pairs of 300 miles or more

2. Arrival Airport Enroute: This measure provides average distance and time data from all flights 300 miles or more from their arrival airport

Dashboards

1. AERO: This provides limited next day airport information

Other

1. Weather Factors: This provides information on the severity of weather factors with regards to their impact on flight delays at airport

2. Diversions: This provides information on flight diversions

3. Advisories: This provides a summary of Traffic Management Initiatives and other aviation-related advisories

4. Data Download: This provides detailed data for airports and individual flights

4.2.3 Automated Surface Observing Systems (ASOS)

The Automated Surface Observing Systems (ASOS) provides forecasted weather conditions that are updated every minute for the meteorological, climatological, hydrological, and aviation industries [46, 47]. The ASOS dataset provides a summary of airport weather conditions such as the date and time that the conditions were recorded as well as weather attributes such as ambient temperature, sea level pressure, visibility, wind speed, wind direction, wind gusts, dew point temperature, precipitation accumulation, cloud height and amount, etc. ASOS data used for this research was obtained online and in csv format [48].

4.3 Data Processing

In order to analyze and utilize the data, there was a need to parse the Traffic Flow Management System (TFMS) dataset so as to enable data fusion across the relevant data fields. This section will also highlight steps taken to ensure that all three datasets contained accurate information prior to Data Fusion.

4.3.1 Traffic Flow Management System (TFMS)

The TFMS datasets were in the Flight Information Exchange Model (FIXM) [49] format, which is appropriate for transmitting flight data. Consequently, there was a need to parse the datasets into a much more usable format (csv) for analytical and Data Fusion purposes. This is because csv is a much more structured and comprehensible format compared to FIXM. The TFMS datasets are stored by the FAA as hourly files comprised of all messages generated during that time period. The datasets also have schema or .xsd files which dictate the structure of the FIXM files and indicate if fields are required or not. The schema is critical to ensure that all required fields are extracted in their correct formats. The TFMS parser was developed accordingly using Python and followed the process highlighted in Figure 4.3 below.

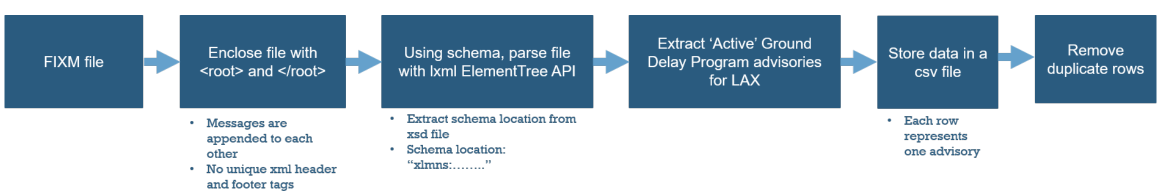

Figure 4.3: TFMS parsing process

1. Since the datasets are comprised of messages generated within the hour, there is no way to distinguish between the beginning of the file and the end of the file. Thus, it was important to enclose each file with a header and footer such as $<$root $>$ and $<\backslash$root $>$ respectively. This ensured that each file had unique starting and end points.

2. Extract the schema location from the .xsd file. The schema location is typically of the format "xlmns:......"

3. Parse the FIXM file using the ElementTree Application Program Interface (API) [50]

4. Extract "Active" Ground Delay Program messages

5. Store each Ground Delay Program message as a row in a csv file

After successfully parsing the TFMS datasets, the data was analyzed to remove duplicate rows, and to ensure that the data was accurate. Parameters extracted for "Active" Ground Delay Programs at the Los Angeles International Airport from January to August 2017 were:

1. Start and end dates and times of Ground Delay Programs

2. Cause of Ground Delay Programs

3. Details of causes of Ground Delay Programs

4. Maximum delay time of Ground Delay Programs

Table 4.1 provides a summary of the different causes of Ground Delay Programs that were extracted from the TFMS datasets, as well as their associated details. Finally, the duration of Ground Delay Programs was computed using their start and end times.

Table 4.1: Summary of causes of Ground Delay Programs at LAX from January - August 2017, and their associated details

Causes of Ground Delay Program	**Details of Ground Delay Programs**
Weather	Fog, Low Ceilings, Thunderstorms, Wind
Volume	Compacted Demand, Multi Taxi, Volume
Runway	Construction
Other	Other

4.3.2 Aviation System Performance Metrics (ASPM)

Data from the Aviation System Performance Metrics database was extracted online in csv format. The following parameters were extracted for the Los Angeles International Airport from January to August 2017:

- Scheduled Arrivals: This parameter states the number of arrivals listed in a published schedule. Cargo flight are typically excluded from this list [21]
- Arrivals For Metric Computation: This includes arrivals to "ASPM airports" as well as flights by "ASPM carriers". General aviation and military flights are not included in this list [21]
- Date and time
- Average taxi-in delay times: This represents the difference between actual taxi-in time and unimpeded taxi-in time [21]

4.3.3 Automated Surface Observing Systems (ASOS)

Automated Surface Observing Systems data was extracted online in csv format. The following parameters were extracted for the Los Angeles International Airport from January to August 2017:

- Date and time

- Air Temperature (Fahrenheit)
- Dew Point Temperature (Fahrenheit)
- Relative Humidity (%)
- Wind Direction (Degrees)
- Wind Speed (Knots)
- Precipitation Accumulation (Inches)
- Pressure Altimeter (Inches)
- Sea level pressure (Millibars)
- Visibility (Miles)
- Wind Gusts (Knots)
- Cloud Coverage Type
- Cloud Altitude (Feet)

Finally, in order to ensure that the ASOS dataset was complete and appropriate for Machine Learning, the dataset was analyzed for missing values. The cloud coverage and altitude parameters particularly had a lot of missing values which meant that no clouds were present. These missing values were replaced with "M" representing missing values.

4.4 Data Fusion

In order to develop a Machine Learning model using the three datasets, it was important to identify the relationships between the datasets so as to fuse them together. Data Fusion is a method of data analysis that involves fusing data from different sources to produce more consistent and useful information than that obtained from a single data source [51]. Consequently, the datasets were fused by date and time in order to have a comprehensive dataset containing Ground Delay Programs of different causes. Some Machine Learning techniques require numerical data rather than categorical data. Thus, after fusing the datasets, there was a need to encode categorical data into numerical data. This can be done in two ways: Integer Encoding and One-Hot Encoding.

Integer Encoding

Integer Encoding involves converting unique categorical data into unique integers [52, 53, 54], as seen in Figure 4.4 where dates were converted into integers.

Date	Delays
1/1/2015	34
1/2/2015	5
1/3/2015	26
1/4/2015	8

Date	Code
1/1/2015	1
1/2/2015	2
1/3/2015	3
1/4/2015	4

Date	Delays
1	34
2	5
3	26
4	8

Figure 4.4: Integer Encoding Process

One-Hot Encoding

One-Hot Encoding involves converting each unique categorical parameter into a binary parameter [55, 56, 57]. From Figure 4.5, it can be seen that four binary variables were created from the four categories (dates).

Date	Delays
1/1/2015	34
1/2/2015	5
1/3/2015	26
1/4/2015	8

1/1/2015	1/2/2015	1/3/2015	1/4/2015	Delays
1	0	0	0	34
0	1	0	0	5
0	0	1	0	26
0	0	0	1	8

Figure 4.5: One-Hot Encoding Process

Machine Learning algorithms often require data to be normalized. Thus, One-Hot Encoding was a more appropriate approach for encoding categorical variables as the encoded variables served as discrete variables for the prediction models. Table 4.2 provides a summary of encoded and non-encoded variables used for this research.

Table 4.2: Summary of encoded and non-encoded variables

Encoded Variables	**Non-Encoded Variables**
Month	Number of actual flight arrivals (Arrivals For Metric Computation)
Hour	Number of scheduled arrivals
Causes of Ground Delay Programs	Duration of Ground Delay Programs (seconds)
Details of causes of Ground Delay Programs	Maximum flight delay times (minutes)
Cloud coverage type	Cloud coverage height
Other weather conditions	

4.4.1 Data Fusion in the context of predicting flight delay times caused by Ground Delay Programs

The Federal Aviation Administration currently provides the general public with information regarding Ground Delay Programs. This information includes the maximum and average delay times that have been assigned to flights during the entire Ground Delay Program. However, knowing the maximum and average delay times per hour during a Ground Delay Program may help airlines and passengers make more informed decisions. In order to achieve this, the following variables were used as predictors for this model with maximum delay time per hour as the target:

- Month
- Hour
- Duration of Ground Delay Programs
- Causes of Ground Delay Programs

- Details of Ground Delay Programs
- Number of actual arrivals
- Number of scheduled arrivals
- Weather conditions

It is important to note that the causes of Ground Delay Programs extracted and used for this prediction model were weather, volume, runway, and other. After predicting the maximum delay time per hour, the number of actual arrivals were used to compute the average delay time per hour during a Ground Delay Program. Both of these delay times were expressed in minutes.

4.4.2 Data Fusion in the context of predicting the duration of Ground Delay Programs

In order to predict the duration of Ground Delay Programs, the following predictors were used:

- Month
- Hour
- Causes of Ground Delay Programs
- Details of Ground Delay Programs
- Number of actual arrivals
- Number of scheduled arrivals
- Weather conditions

The causes of Ground Delay Programs extracted and used for this model were weather, volume, runway, and other.

4.4.3 Data Fusion in the context of predicting the impact of Ground Delay Programs on taxi-in delay times

In order to predict average taxi-in delay times during Ground Delay Programs, the following predictors were used:

- Month
- Hour
- Causes of Ground Delay Programs
- Details of Ground Delay Programs
- Number of actual arrivals
- Number of scheduled arrivals
- Duration of Ground Delay Programs
- Maximum delay time during Ground Delay Programs
- Weather conditions

The causes of Ground Delay Programs extracted and used for this model were weather, volume, runway, and other.

4.4.4 Data Fusion in the context of predicting the occurrence of Ground Delay Programs

In order to predict the occurrence of Ground Delay Programs, the following predictors were used:

- Month
- Hour

- Number of actual arrivals
- Number of scheduled arrivals
- Weather conditions

The targets of this model were weather-related Ground Delay Programs, volume-related Ground Delay Programs, and no Ground Delay Program events. "Normal" was indicated as the cause whenever Ground Delay Programs did not occur.

4.5 Model Generation, Validation, & Testing Process

One of the main outcomes of this research was the identification of Machine Learning techniques that allowed for accurate predictions. This involved the following steps:

4.5.1 Identification of Machine Learning Algorithms

In order to ensure that the prediction models were developed correctly with optimal performance, it was important to identify and use the appropriate Machine Learning algorithms based on the tasks at hand. The targets for the aforementioned models can be broken down into two categories: classification and numerical predictions. Table 4.3 shows which category the different prediction models belong to. Table 4.4 shows the Machine Learning techniques to be used for the classification and numerical prediction tasks.

Table 4.3: Prediction models and their accompanying tasks

Prediction Model	**Task**
Flight delay times caused by GDP	Numerical prediction
Duration of GDP	Numerical prediction
Average taxi-in delay times during GDP	Numerical prediction
Occurrence of GDP	Classification

Table 4.4: Machine Learning algorithms investigated

Numerical Prediction Algorithms	**Classification Algorithms**
Linear Regression	Bagging Ensemble
Regression Trees	Naive Bayes
Model Trees	Decision Trees
Artificial Neural Networks	Boosting Ensemble
Support Vector Machines	Support Vector Machines
Random Forest	Classification Rule Learners
	Random Forest

4.5.2 Model Generation, Validation and Testing with R

After successfully fusing the datasets, the data was partitioned into three sets: training, validation and testing. One process of partitioning data into these three sets is known as the holdout method [58]. As shown in Figure 4.6, half of the data was assigned to the training set which was used to generate the model, one-fourth of the data was assigned to the validation set which is used to iterate and refine the model, and one-fourth of the data was assigned to the test set which was used to generate predictions for evaluations. To ensure that the training, validation and test datasets did not have systematic differences, the fused data was randomly divided between the three sets. The performance of the test dataset alone should never be allowed to influence the performance of the model. Thus, it was important to include the validation set to ensure that a truly accurate estimate of future performances was obtained. During the validation process, the models are also tuned and refined to ensure optimal model performance.

4.6 Evaluation

The final step of this methodology was the evaluation of model performances. Evaluating the performance of learners is vital as it indicates how a learner will perform on future/unseen data. The type of evaluation metric used depends on whether the task involved classifications or numeric predictions, as well as on how "balanced" the dataset the models

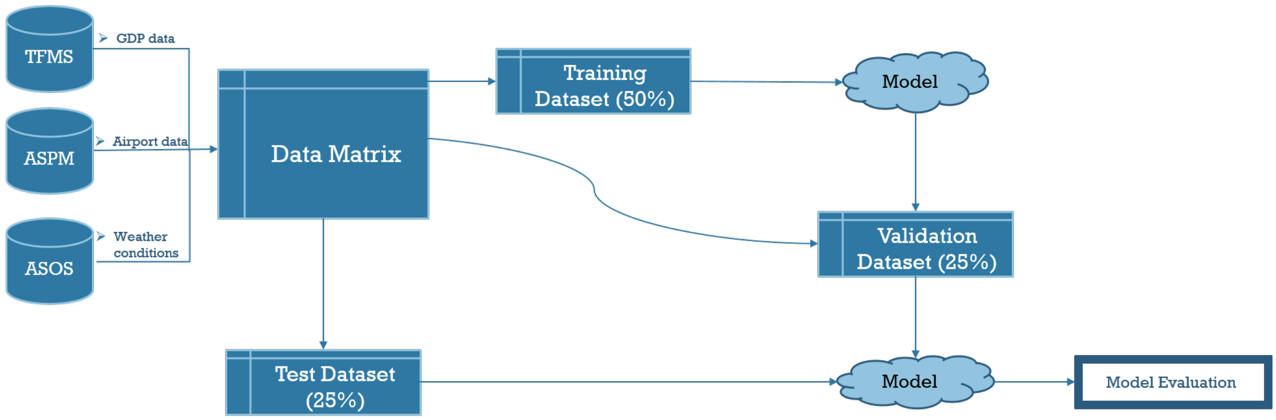

Figure 4.6: Model Generation, Validation and Testing

were being trained on.

4.6.1 Numerical Predictions Evaluation Metrics

Numerical prediction learners are typically evaluated by analyzing how well the model fits the data. Four evaluation metrics were used to evaluate the numerical predictors: R-squared, Pearson's Correlation Coefficient, Mean Absolute Error, and Root Mean Squared Error.

R-squared

R-squared values or the coefficient of determination indicate how close the relationship between predictors and targets follows a fitted regression line [25, 59]. Optimal performance of a prediction model is associated with an R-squared value close to 1.

Pearson's Correlation Coefficient

The correlation between variables is a measure of how close the relationship between predictors and targets follows a straight line. This measure has a maximum value of 1 which corresponds to a perfect linear relationship, while a value of 0 corresponds to a lack of linear relationship between variables [25, 60].

Mean Absolute Error

Mean Absolute Error refers to how far a model's predictions are from the actual values [25]. It is calculated by taking the mean of the absolute values of the difference between the predicted and actual values [61]. The lower the Mean Absolute Error, the better the performance of the model.

Root Mean Squared Error

Root Mean Squared Error refers to the standard deviation of the difference between a model's predictions and the actual values [61]. The lower the Root Mean Squared Error, the better the performance of the model.

4.6.2 Classification Evaluation Metrics

Classification learners are typically evaluated using results obtained from a confusion matrix. A confusion matrix as seen in Table 4.5 is a table that categorizes predictions according to whether they match the actual value. For classification tasks, confusion matrices are used to measure performance using the metrics highlighted in this section.

Table 4.5: Confusion Matrix

	Actual: No	**Actual: Yes**
Predicted: No	True Negative (TN)	False Positive (FP)
Predicted: Yes	False Negative (FN)	True Positive (TP)

True Positive (TP) refers to the correct classification of the class of interest. True Negative (TN) refers to the correct classification of the class that is not of interest. False Positive (FP) refers to the incorrect classification of the class of interest. False Negative (FN) refers to the incorrect classification of the class that is not of interest [25].

Accuracy [25]

This refers to the ratio of the number of true positives and negatives, and the total number of predictions and is specified as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Error Rate [25]

This refers to the proportion of incorrectly classified examples and is specified as:

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN} = 1 - Accuracy$$

Sensitivity [25]

This refers to the proportion of true positives that were correctly classified and is specified as:

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity [25]

This refers to the proportion of negative examples that were correctly classified and is specified as:

$$Specificity = \frac{TN}{FP + TN}$$

Precision [25]

This refers to the proportion of positive examples that were truly positive and is specified as:

$$Precision = \frac{TP}{FP + TP}$$

Recall [25]

This refers to the ratio of true positives to total number of positives and is specified as:

$$Recall = \frac{TP}{TP + FN}$$

Kappa Statistic [25]

A model might have high accuracy because it correctly predicts the most frequent class, particularly when the dataset is unbalanced. Kappa Statistic adjusts accuracy by accounting for the probability of a correct prediction by chance alone, and is appropriate for unbalanced datasets.

CHAPTER 5

ANALYSIS & RESULTS

This chapter highlights the steps taken to develop and evaluate each of the prediction models using R [62]. This includes highlighting any steps taken to tune the models, as well as an evaluation of the models using the evaluation metrics mentioned in section 4.6.

5.1 Predicting flight delay times due to Ground Delay Programs

As mentioned previously, the objective of this model is to predict the maximum delay time per hour, and to compute the average delay time per hour due to Ground Delay Programs. Six Machine Learning techniques were benchmarked to identify a suitable technique for the prediction model: Multiple Linear Regression, Regression Trees, Model Trees, Artificial Neural Networks, Support Vector Machines, and Random Forests. This section highlights steps taken to develop and tune the models using the aforementioned algorithms and provides an analysis of their performance with the validation and testing datasets. In order to ensure that the algorithms were assessed accurately, the data was randomly divided into three categories: training, validation, and testing datasets. The predictors for this model were the causes of Ground Delay Programs, details of Ground Delay Programs, the number of actual arrivals, the number of scheduled arrivals, weather conditions, the duration of Ground Delay Programs, the month, and the hour. The training, validation, and testing datasets had 641, 319, and 322 data points respectively.

5.1.1 Multiple Linear Regression

Steps taken in R to develop a prediction model using the Multiple Linear Regression algorithm are as follows:

1. Load the data using the "read.csv" function [25]

2. Train the model using the "lm" function [63] and the training dataset

3. Test the performance of the model using the "lm" function and the validation dataset

4. Calculate the Mean Absolute and Root Mean Squared Errors using the "residuals" function [64]

5. Obtain the R-squared value using the "r.squared" attribute [65] and compute Correlation by taking the square root of the R-squared value

6. Repeat steps 3 to 5 with the testing dataset

The Multiple Linear Regression algorithm provides insights into the importance of the different predictors. Analysis of the model developed using this technique revealed that the month of March, the duration of Ground Delay Programs, volume-related Ground Delay Programs with multitaxi as their detail, other causes of Ground Delay Programs, weather-related Ground Delay Programs with fog as their detail, and runway related Ground Delay Programs were the most influential predictors for this model, as seen in Figure 5.1.

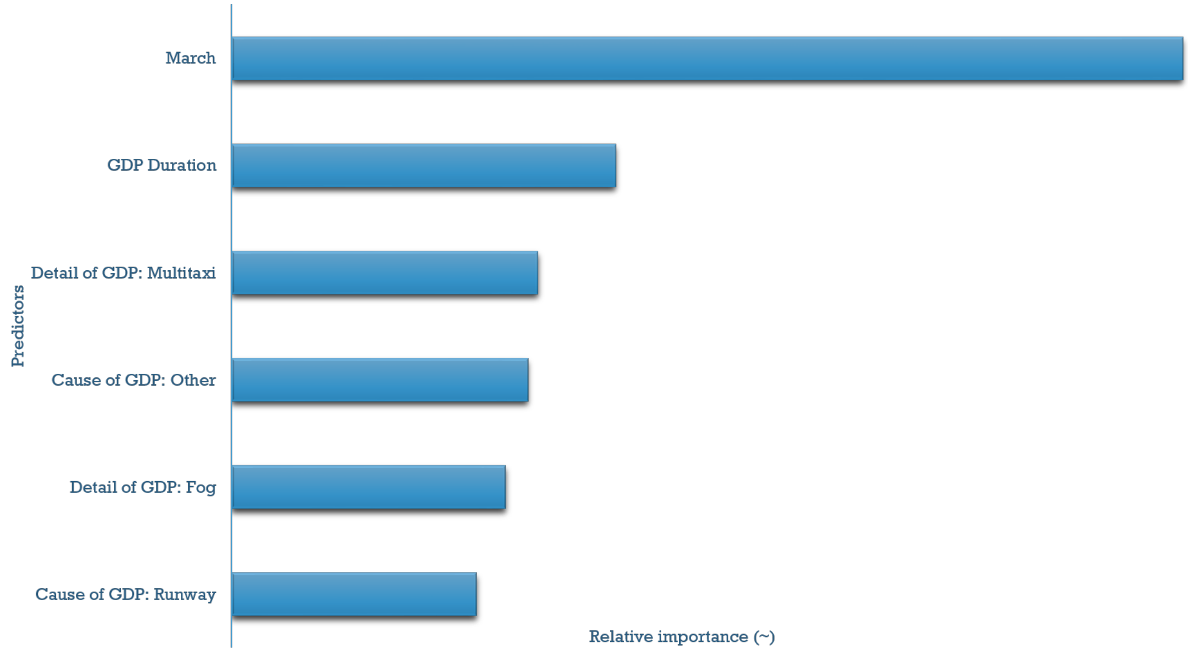

Figure 5.1: Predictor importance for Multiple Linear Regression algorithm for predicting maximum flight delay time during Ground Delay Programs

5.1.2 Regression Trees

Steps taken in R to develop a prediction model using the Regression Trees algorithm are as follows:

1. Load the data using the "read.csv" function [25]
2. Train the model using the "rpart" function [66] and the training dataset
3. Test the performance of the model using the "predict" function [67] and the validation dataset
4. Compute Correlation using the "cor" function [68]
5. Compute the R-squared value by squaring Correlation

6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

7. Repeat steps 3 to 6 with the testing dataset

The Regression Tree algorithm provides insights into the importance of the different predictors. Analysis of the model developed using this technique revealed that the duration of Ground Delay Programs, the month of March, pressure altimeter, the month of January, and sea level pressure were the most influential predictors for this model, as seen in Figure 5.2.

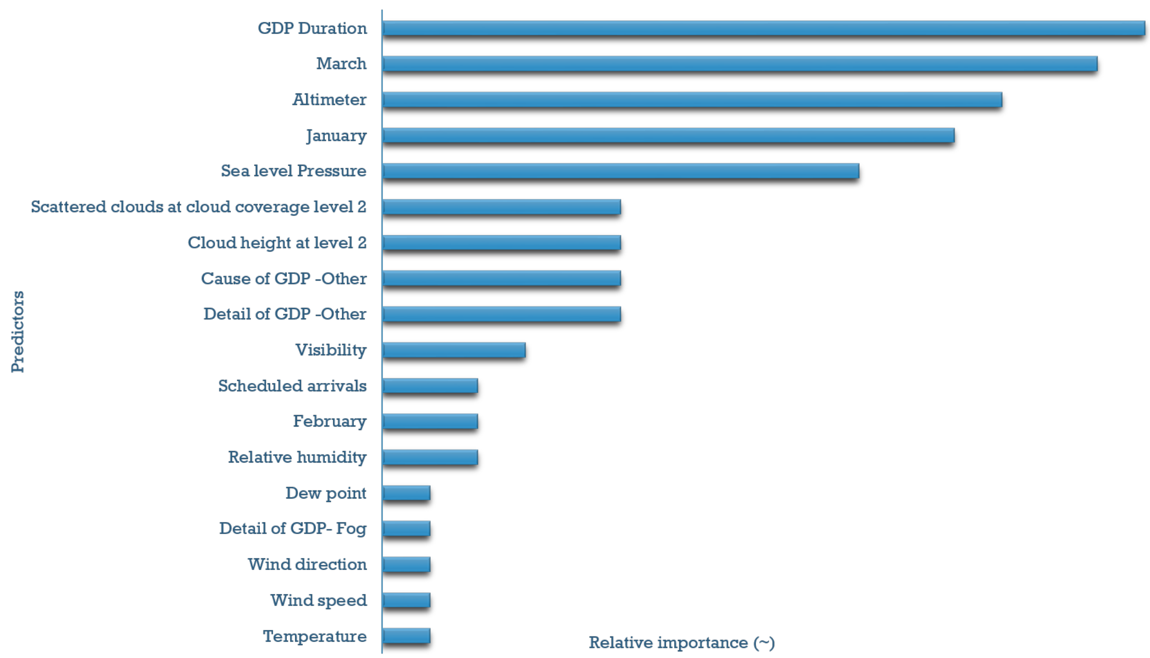

Figure 5.2: Predictor importance for Regression Tree algorithm for predicting maximum flight delay time during Ground Delay Programs

5.1.3 Model Trees

Steps taken in R to develop a prediction model using the Model Trees algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. Train the model using the “M5P” function [69] and the training dataset

3. Test the performance of the model using the “predict” function [67] and the validation dataset

4. Compute Correlation using the “cor” function [68]

5. Compute the R-squared value by squaring Correlation

6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

7. Repeat steps 3 to 6 with the testing dataset

Similarly, the Model Tree algorithm provides insights into the importance of the different predictors. The duration of Ground Delay Programs, pressure altimeter, and sea level pressure were found to be the most influential predictors for this model, as seen in Figure 5.3.

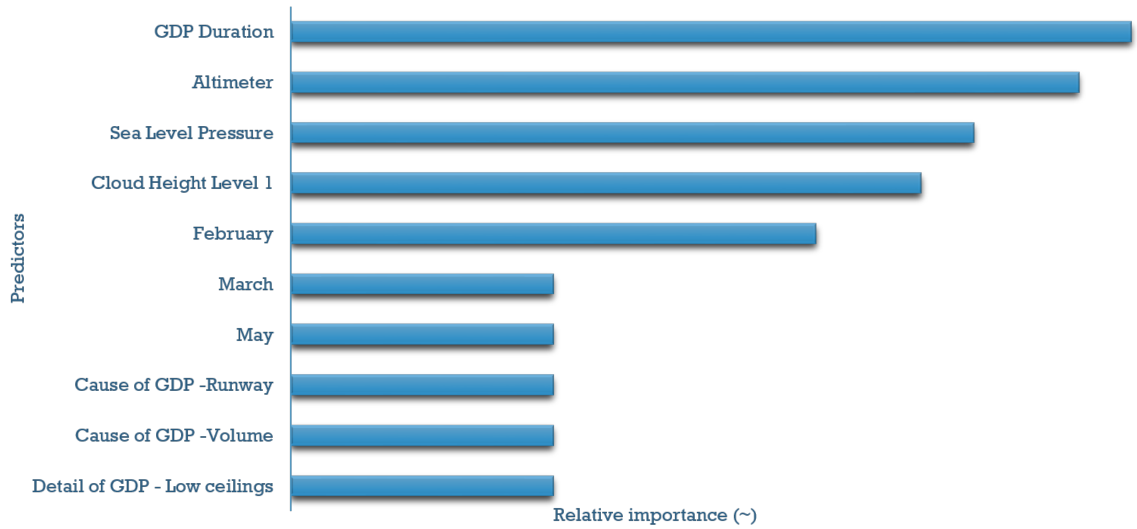

Figure 5.3: Predictor importance for Model Tree algorithm for predicting maximum flight delay time during Ground Delay Programs

5.1.4 Artificial Neural Networks

Steps taken in R to develop a prediction model using the Artificial Neural Networks algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. Normalize continuous variables to a 0 - 1 range since Neural Networks perform optimally when predictors are scaled to a narrow range

3. Train the model with using the “neuralnet” function [70] and the training dataset

4. Vary the number of hidden nodes to identify the optimal hidden node setting for the model. This is achieved by testing the performance of the model using the “compute” function [71] and the validation dataset

5. Compute Correlation using the “cor” function [68]

6. Compute the R-squared value by squaring Correlation

7. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

8. Repeat steps 5 to 7 with the testing dataset using the optimal hidden node setting

Varying the number of hidden nodes and evaluating the model's performance with the validation dataset revealed that the optimal number of hidden nodes was 3 using a logistic activation function which maps inputs into the range of 0 to 1 [72]. Figure 5.4 shows the optimal model's network with input nodes for each of the predictors, followed by three hidden nodes and one output node that predicts maximum delay time. Bias terms, indicated by the blue lines allow the values at the nodes to be updated, like the intercept of a linear equation.

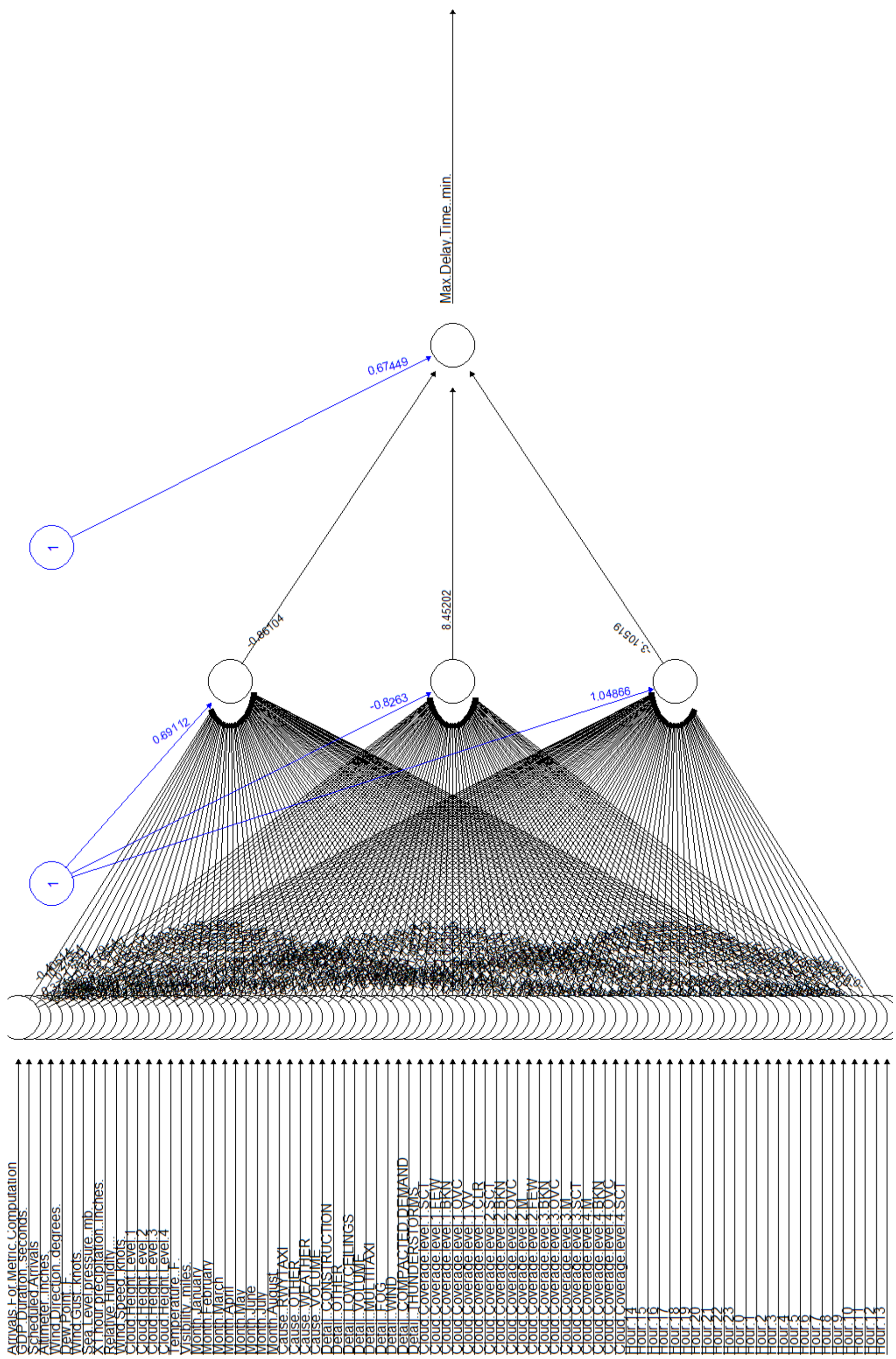

Figure 5.4: Visualization of Neural Network topology for predicting maximum flight delay time during Ground Delay Programs

5.1.5 Support Vector Machines

Steps taken in R to develop a prediction model using the Support Vector Machines algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. The “svm” function [73] by default sets the maximum allowed error between predicted and actual values to 0.1, which often causes over-fitting. In order to reduce the incidence of over-fitting, a cost penalty can be applied to the models. Thus, the maximum allowed error and cost functions were varied from 0 to 1 in steps of 0.1, and 2^n where n varies from 0.5 to 8 in steps of 0.5, respectively to identify the optimal settings for the model

3. Train the model using the “e1071” package [74] and the training dataset

4. Test the performance of the model using the “predict” function [67] and the validation dataset

5. Compute Correlation using the “cor” function [68]

6. Compute the R-squared value by squaring Correlation

7. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

8. Repeat steps 4 to 7 with the testing dataset

Varying the maximum allowed error and cost functions, and evaluating the model’s performance revealed that the optimal model had a maximum allowed error of 0 and cost of 64.

5.1.6 Random Forests

Steps taken in R to develop a prediction model using the Random Forests algorithm are as follows:

1. Load the data using the “read.csv” function [25]
2. Train the model using the “randomForest” function [75] and the training dataset
3. Test the performance of the model using the “predict” function [67] and the validation dataset
4. Compute Correlation using the “cor” function [68]
5. Compute the R-squared value by squaring Correlation
6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values
7. Repeat steps 3 to 6 with the testing dataset

Analysis of the Random Forest algorithm revealed that for the duration of Ground Delay Programs, the months of March and January, other causes of Ground Delay Programs, other causes of Ground Delay Programs with other details, and altimeter pressure were the most influential predictors for this model as seen in Figure 5.5.

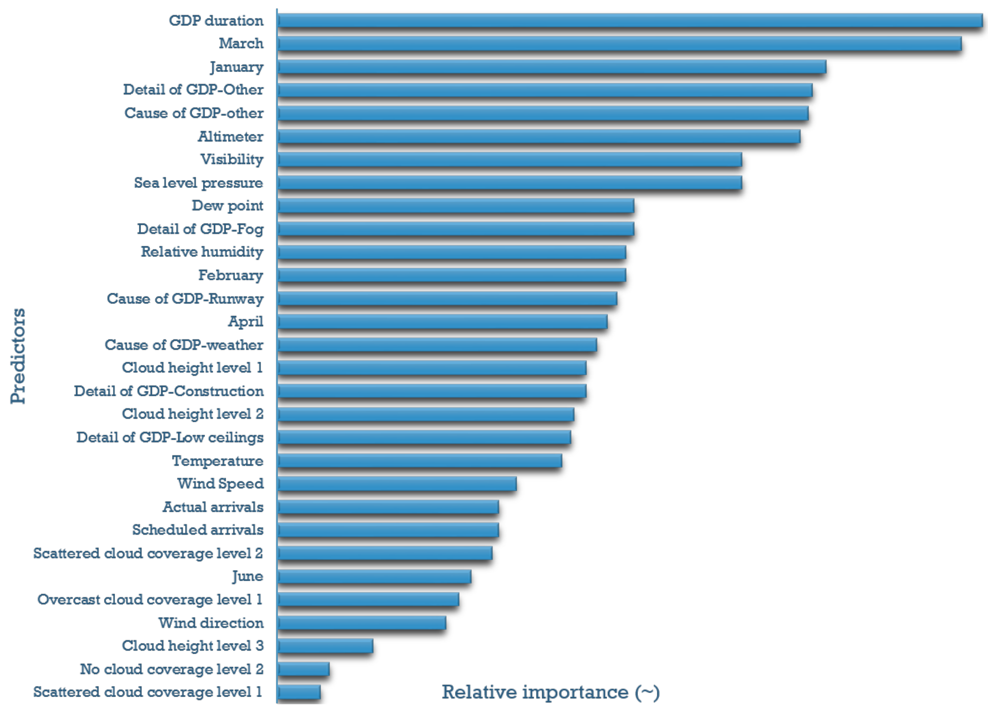

Figure 5.5: Predictor importance for Random Forest algorithm for predicting maximum delay time during Ground Delay Programs

5.1.7 Summary

Table 5.1 provides a summary of the evaluation of the performance of the different Machine Learning techniques with the validation dataset. It can be seen that Random Forests had the best performance, with the highest R-squared and Correlation values, and the lowest Mean Absolute and Root Mean Squared Errors.

Table 5.1: Evaluation of technique performance for predicting maximum flight delay times during Ground Delay Programs with the validation dataset

Algorithm/Metric	**R-squared**	**Mean Absolute Error (minutes)**	**Correlation**	**Root Mean Squared Error (minutes)**
Multiple Linear Regression	0.5	48.7	0.71	76.6
Regression Trees	0.82	17.8	0.91	49.2
Model Trees	0.87	19.4	0.93	40.6
Artificial Neural Networks	0.71	27.3	0.84	58.5
Support Vector Machines	0.68	35.6	0.83	61.5
Random Forests	0.9	14.3	0.95	34.1

Tables 5.2 provides a summary of the evaluation of the performance of the different Machine Learning techniques with the testing dataset. It can be seen that the Random Forest had the lowest Mean Absolute Error value. On the other hand, Model Trees had the highest Correlation and R-squared values, and the lowest Root Mean Squared Error value. However, the R-squared, Correlation, and Root Mean Squared Errors of the Random Forest Ensemble were lower than those of the Model Tree by 0.02, 0.01, and 1.2 respectively. Consequently, **Random Forests** was identified as the best suited algorithm for predicting the maximum delay time per hour for Ground Delay Programs.

Table 5.2: Evaluation of technique performance for predicting maximum flight delay times during Ground Delay Programs with the testing dataset

Algorithm/Metric	**R-squared**	**Mean Absolute Error (minutes)**	**Correlation**	**Root Mean Squared Error (minutes)**
Multiple Linear Regression	0.48	37.9	0.69	65.4
Regression Trees	0.76	16.7	0.87	48.8
Model Trees	0.88	16.4	0.94	32.4
Artificial Neural Networks	0.6	26	0.78	57.7
Support Vector Machines	0.66	34.9	0.81	53.6
Random Forests	0.86	13.8	0.93	33.6

The average delay time per hour for Ground Delay Programs was then computed using:

$$\text{Average delay time per hour} = \frac{\text{Maximum delay time per hour}}{\text{Number of actual arrivals}}$$

5.2 Predicting the duration of Ground Delay Program

The objective of this model is to predict the duration of Ground Delay Programs. Six Machine Learning techniques were benchmarked to identify a suitable technique for the prediction model: Multiple Linear Regression, Regression Trees, Model Trees, Artificial Neural Networks, Support Vector Machines, and Random Forests. This section highlights steps taken to develop and tune the models using the aforementioned algorithms, and provides an analysis of their performance with the validation and testing datasets. In order to ensure that the algorithms were assessed accurately, the data was randomly divided into three categories: training, validation, and testing sets. The predictors for this model were the causes of Ground Delay Programs, details of Ground Delay Programs, number of actual arrivals, number of scheduled arrivals, weather conditions, the month, and the hour. The training, validation, and testing datasets had 641, 319, and 322 data points respectively.

5.2.1 Multiple Linear Regression

Steps taken in R to develop a prediction model using the Multiple Linear Regression algorithm are as follows:

1. Load the data using the “read.csv” function [25]
2. Train the model using the “lm” function [63] and the training dataset
3. Test the performance of the model using the “lm” function and the validation dataset
4. Calculate the Mean Absolute and Root Mean Squared Errors using the ”residuals” function [64]
5. Obtain the R-squared value using the “r.squared” attribute [65] and compute Correlation by taking the square root of the R-squared value
6. Repeat steps 3 to 5 with the testing dataset

The Multiple Linear Regression algorithm provides insights into the importance of the different predictors. Analysis of the model developed using this technique revealed that cloud height level 1, overcast clouds at height level 2, overcast, broken, and few clouds at height level 1 were the most influential predictors for this model, as seen in Figure 5.6.

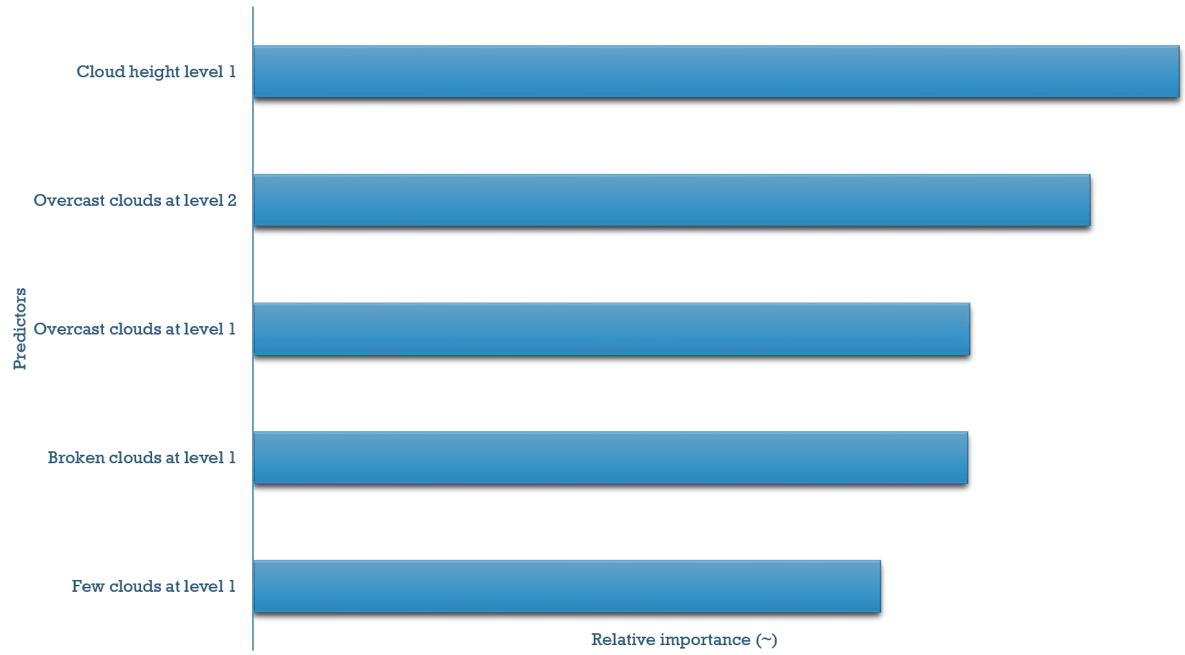

Figure 5.6: Predictor importance for Multiple Linear Regression algorithm for predicting the duration of Ground Delay Programs

5.2.2 Regression Trees

Steps taken in R to develop a prediction model using the Regression Trees algorithm are as follows:

1. Load the data using the “read.csv” function [25]
2. Train the model using the “rpart” function [66] and the training dataset
3. Test the performance of the model using the “predict” function [67] and the validation dataset
4. Compute Correlation using the “cor” function [68]
5. Compute the R-squared value by squaring Correlation

6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values
7. Repeat steps 3 to 6 with the testing dataset

Analysis of the Regression Tree algorithm revealed that dew point, the month of April, cloud height at level 1, pressure altimeter, relative humidity, sea level pressure, and the month of May were the most influential predictors for this model, as seen in Figure 5.7.

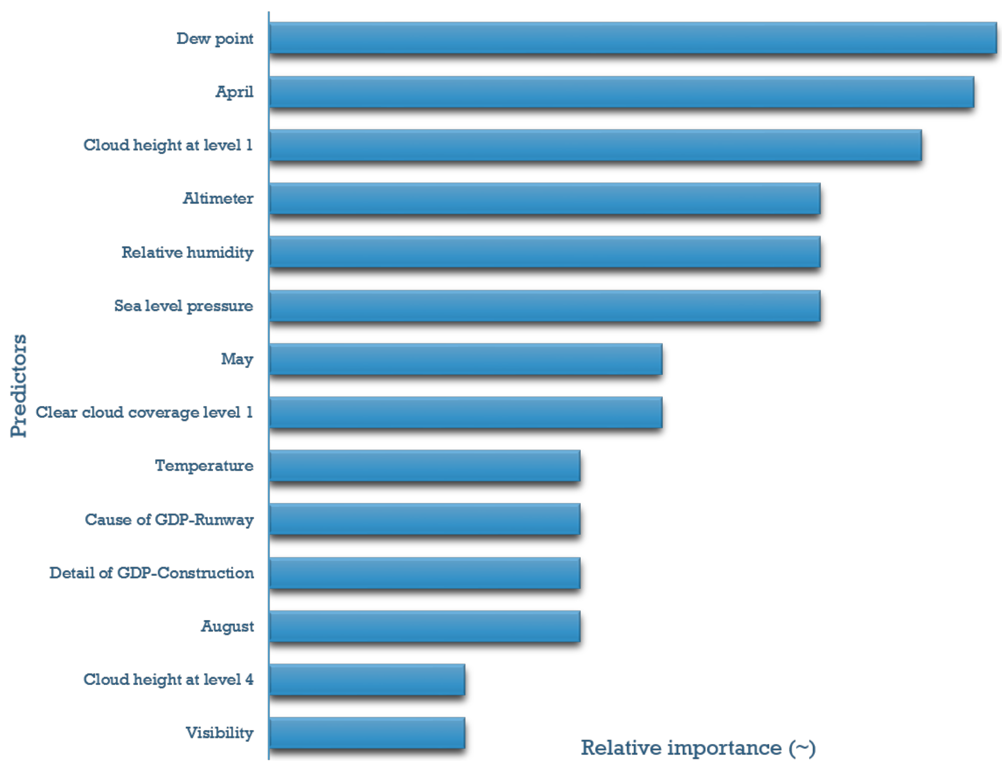

Figure 5.7: Predictor importance for Regression Tree algorithm for predicting the duration of Ground Delay Programs

5.2.3 Model Trees

Steps taken in R to develop a prediction model using the Model Trees algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. Train the model using the “M5P” function [69] and the training dataset

3. Test the performance of the model using the “predict” function [67] and the validation dataset

4. Compute Correlation using the “cor” function [68]

5. Compute the R-squared value by squaring Correlation

6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

7. Repeat steps 3 to 6 with the testing dataset

Analysis of the Model Tree algorithm revealed that runway-related Ground Delay Programs, sea level pressure, and weather-related Ground Delay Programs caused by winds were the most influential predictors for this model, as seen in Figure 5.8.

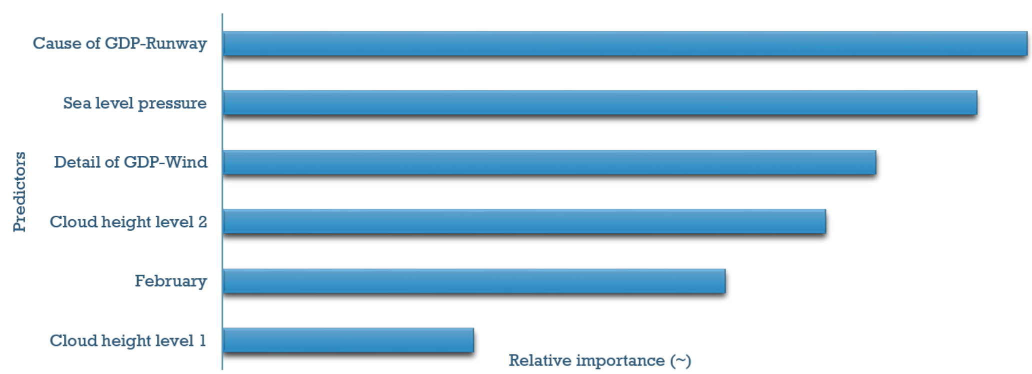

Figure 5.8: Predictor importance for Model Tree algorithm for predicting the duration of Ground Delay Programs

5.2.4 Artificial Neural Networks

Steps taken in R to develop a prediction model using the Artificial Neural Networks algorithm are as follows:

1. Load the data using the “read.csv” function [25]
2. Normalize continuous variables to a 0 - 1 range since Neural Networks perform optimally when predictors are scaled to a narrow range
3. Train the model with using the “neuralnet” function [70] and the training dataset
4. Vary the number of hidden nodes to identify the optimal hidden node setting for the model. This is achieved by testing the performance of the model using the ”compute” function [71] and the validation dataset
5. Compute Correlation using the “cor” function [68]
6. Compute the R-squared value by squaring Correlation
7. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values
8. Repeat steps 4 to 7 with the testing dataset

Varying the number of hidden nodes and evaluating the model’s performance with the validation dataset revealed that the optimal number of hidden nodes was 4 using a logistic activation function which maps inputs into the range of 0 to 1 [72]. Figure 5.9 shows the optimal model’s network with input nodes for each of the predictors, followed by four hidden nodes and one output node that predicts the duration of Ground Delay Programs. Bias terms, indicated by the blue lines allow the values at the nodes to be updated, like the intercept of a linear equation.

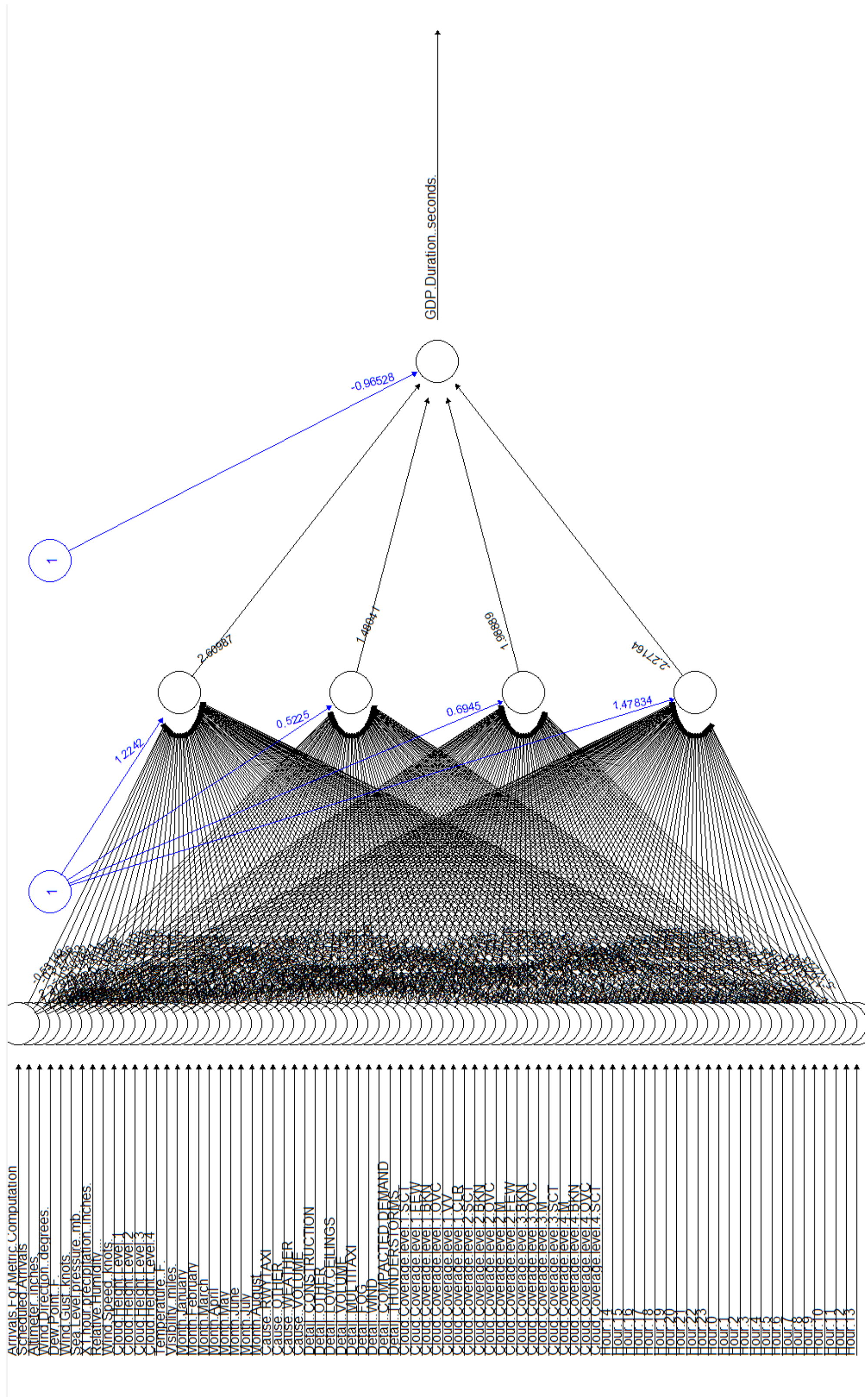

Figure 5.9: Visualization of the Neural Network topology for predicting the duration of Ground Delay Programs

5.2.5 Support Vector Machines

Steps taken in R to develop a prediction model using the Support Vector Machines algorithm are as follows:

1. Load the data using the "read.csv" function [25]

2. The "svm" function [73] by default sets the maximum allowed error to 0.1, which often causes over-fitting. In order to reduce the incidence of over-fitting, a cost penalty can be applied to the models. Thus, the maximum allowed error and cost functions were varied from 0 to 1 in steps of 0.1, and 2^n where n varies from 0.5 to 8 in steps of 0.5, respectively to identify the optimal settings for the model

3. Train the model using the "e1071" package [74] and the training dataset

4. Test the performance of the model using the "predict" function [67] and the validation dataset

5. Compute Correlation using the "cor" function [68]

6. Compute the R-squared value by squaring Correlation

7. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

8. Repeat steps 4 to 7 with the testing dataset

Varying the maximum allowed error and cost functions, and evaluating the model's performance revealed that the optimal model had a maximum allowed error of 0 and cost of 2.

5.2.6 Random Forest

Steps taken in R to develop a prediction model using the Random Forest algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. Train the model using the “randomForest” function [75] and the training dataset

3. Test the performance of the model using the ”predict” function [67] and the validation dataset

4. Compute Correlation using the “cor” function [68]

5. Compute the R-squared value by squaring Correlation

6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

7. Repeat steps 3 to 6 with the testing dataset

Analysis of the Random Forest algorithm revealed that pressure altimeter, dew point, and sea level pressure were the most influential predictors for this model as seen in Figure 5.10.

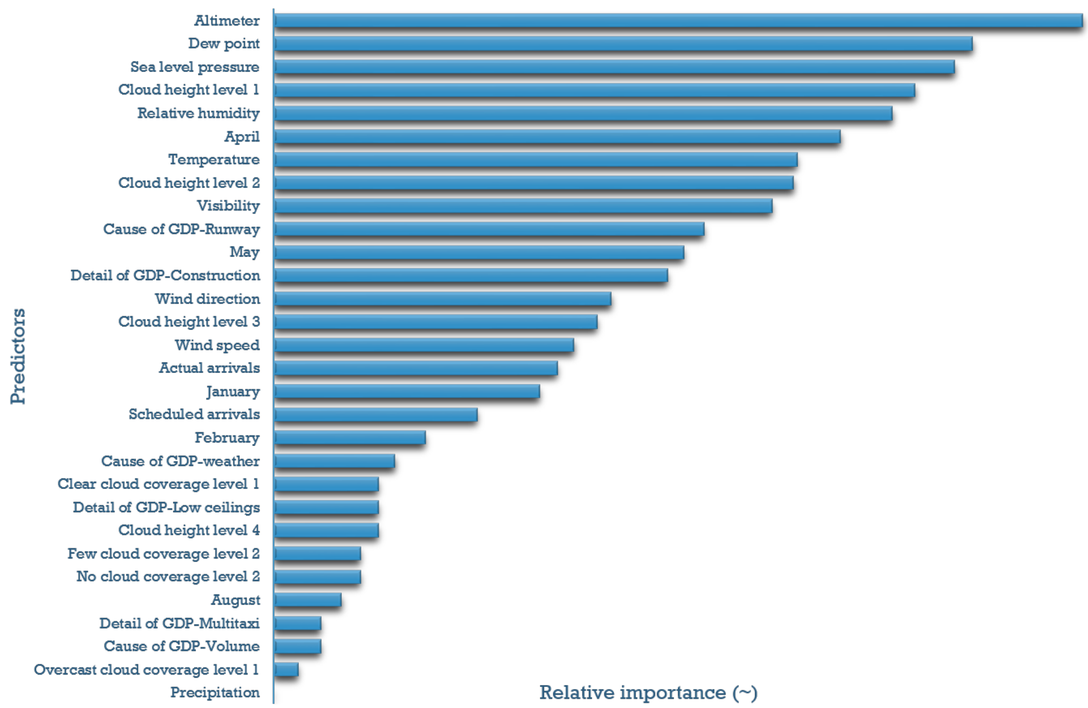

Figure 5.10: Predictor importance for Random Forest Ensemble algorithm for predicting the duration of Ground Delay Programs

5.2.7 Summary

Tables 5.3 provides a summary of the evaluation of the performance of the different Machine Learning techniques with the validation set. It can be seen that Random Forests had the best performance, with the highest R-squared and correlation values, and the lowest Mean Absolute and Root Mean Squared Errors.

Table 5.3: Evaluation of technique performance for predicting the duration of Ground Delay Programs with the validation dataset

Algorithm/Metric	**R-squared**	**Mean Absolute Error (seconds)**	**Correlation**	**Root Mean Squared Error (seconds)**
Multiple Linear Regression	0.47	3130.6	0.68	3849.2
Regression Trees	0.43	3083.3	0.65	4068.1
Model Trees	0.58	2534.3	0.76	3423.7
Artificial Neural Networks	0.26	4187	0.51	5460.8
Support Vector Machines	0.39	3014.9	0.63	3361.2
Random Forest Ensemble	0.6	2410.2	0.77	3361.2

Tables 5.4 provides a summary of the evaluation of the performance of the different Machine Learning techniques with the testing set. It can be seen that the Random Forest Ensemble had the best performance, with the highest R-squared and correlation values, and the lowest Mean Absolute and Root Mean Squared Errors.

Table 5.4: Evaluation of technique performance for the duration of Ground Delay Programs with the testing dataset

Algorithm/Metric	R-squared	Mean Absolute Error (seconds)	Correlation	Root Mean Squared Error (seconds)
Multiple Linear Regression	0.51	3039.4	0.72	3852.5
Regression Trees	0.52	2848.7	0.72	3803.8
Model Trees	0.59	2593.1	0.77	3557.7
Artificial Neural Networks	0.27	4293.9	0.52	5479.7
Support Vector Machines	0.4	3117.8	0.63	4269.5
Random Forest Ensemble	0.68	2282.2	0.82	3200.2

Consequently, the **Random Forest Ensemble** was identified as the best suited algorithm for predicting the duration of Ground Delay Programs.

5.3 Predicting average taxi-in delay times during Ground Delay Programs

The objective of this model is to predict average taxi-in delay times during Ground Delay Programs. Six Machine Learning techniques were benchmarked to identify a suitable technique for the prediction model: Multiple Linear Regression, Regression Trees, Model Trees, Artificial Neural Networks, Support Vector Machines, and Random Forests. This section highlights steps taken to develop and tune the models using the aforementioned algorithms and provides an analysis of their performance with the validation and testing sets. In order to ensure that the algorithms were assessed accurately, the data was randomly divided into three categories: training, validation, and testing sets. The predictors for this model were the causes of Ground Delay Programs, details of Ground Delay Programs, the duration of Ground Delay Programs, number of actual arrivals, number of scheduled arrivals, maximum delay times, weather conditions, the month, and the hour. The training, validation, and testing datasets had 641, 319, and 322 data points respectively.

5.3.1 Multiple Linear Regression

Steps taken in R to develop a prediction model using the Multiple Linear Regression algorithm are as follows:

1. Load the data using the “read.csv” function [25]
2. Train the model using the “lm” function [63] and the training dataset
3. Test the performance of the model using the “lm” function and the validation dataset
4. Calculate the Mean Absolute and Root Mean Squared Errors using the “residuals” function [64]
5. Obtain the R-squared value using the “r.squared” attribute [65] and compute Correlation by taking the square root of the R-squared value
6. Repeat steps 3 to 5 with the testing dataset

The Multiple Linear Regression algorithm provides insights into the importance of the different predictors. Analysis of the model developed using this technique revealed that hours 6, 5, 15, 4, 7, and 3, number of scheduled arrivals, hour 8, and the number of actual arrivals were the most influential predictors for this model, as seen in Figure 5.11.

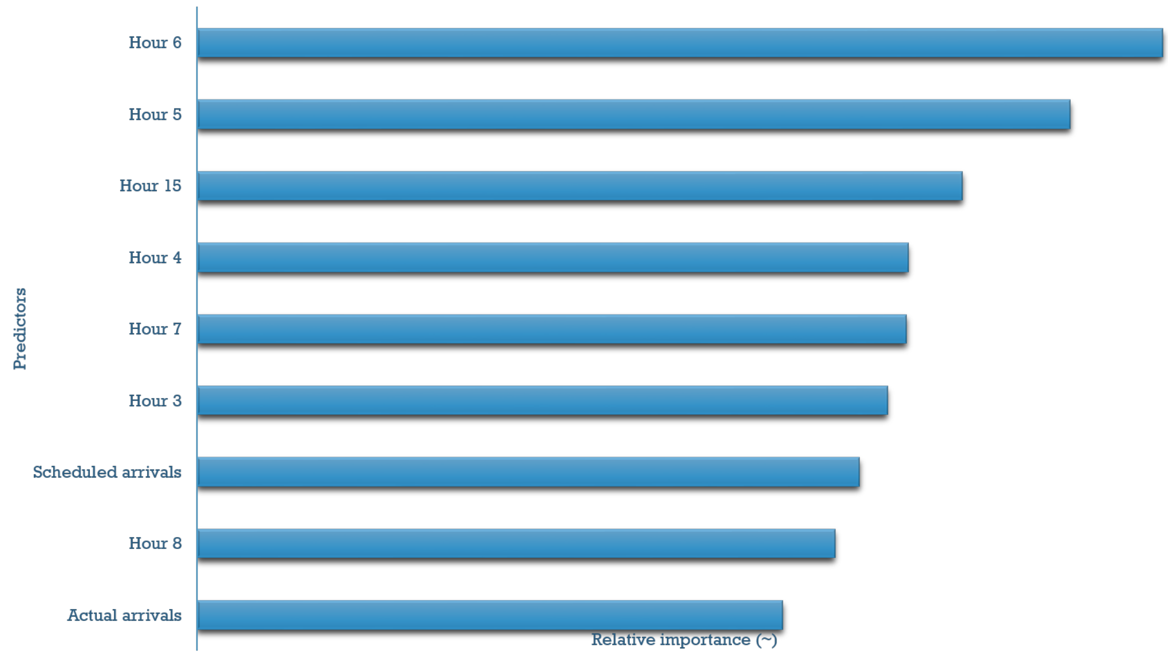

Figure 5.11: Predictor importance for Multiple Linear Regression algorithm for predicting average taxi-in delay times during Ground Delay Programs

5.3.2 Regression Trees

Steps taken in R to develop a prediction model using the Regression Trees algorithm are as follows:

1. Load the data using the "read.csv" function [25]
2. Train the model using the "rpart" function [66] and the training dataset
3. Test the performance of the model using the "predict" function [67] and the validation dataset
4. Compute Correlation using the "cor" function [68]
5. Compute the R-squared value by squaring Correlation

6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

7. Repeat steps 3 to 6 with the testing dataset

Analysis of the Regression Tree algorithm revealed that the actual number of arrivals and number of scheduled arrivals were the most influential predictors for this model, as seen in Figure 5.12.

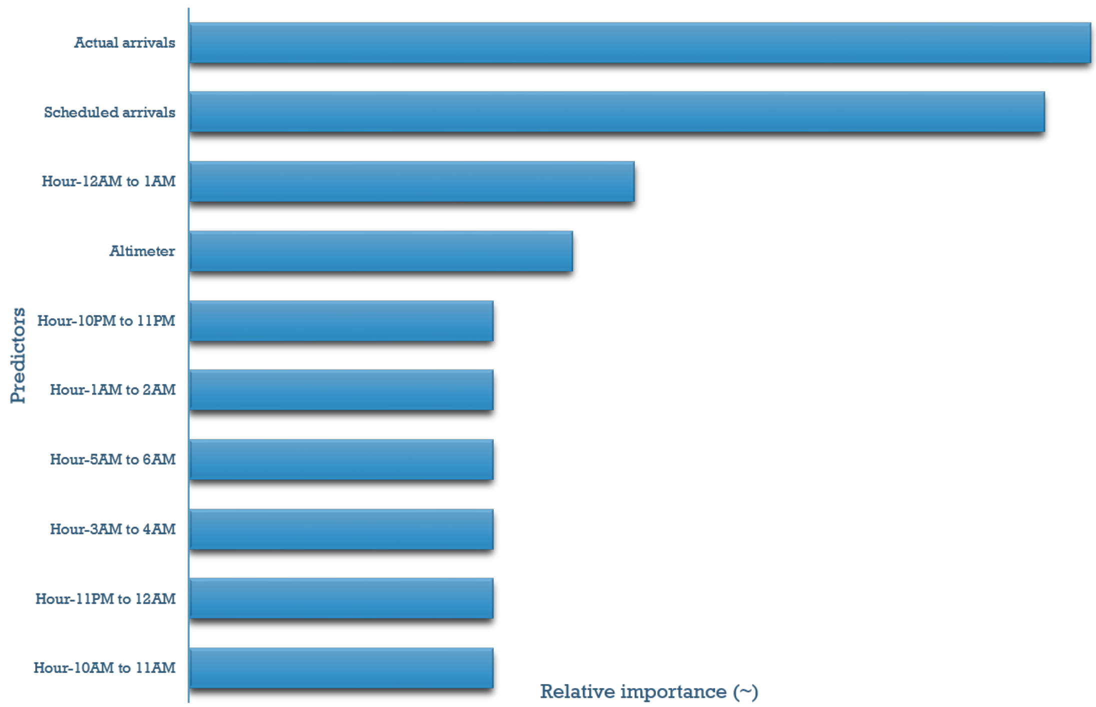

Figure 5.12: Predictor importance for Regression Tree algorithm for predicting average taxi-in delay times during Ground Delay Programs

5.3.3 Model Trees

Steps taken in R to develop a prediction model using the Model Trees algorithm are as follows:

1. Load the data using the “read.csv” function [25]
2. Train the model using the “M5P” function [69] and the training dataset
3. Test the performance of the model using the “predict” function [67] and the validation dataset
4. Compute Correlation using the “cor” function [68]
5. Compute the R-squared value by squaring Correlation
6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values
7. Repeat steps 3 to 6 with the testing dataset

Analysis of the Model Tree algorithm revealed that the actual number of arrivals, relative humidity, sea level pressure, and altimeter pressure were the most influential predictors for this model, as seen in Figure 5.13.

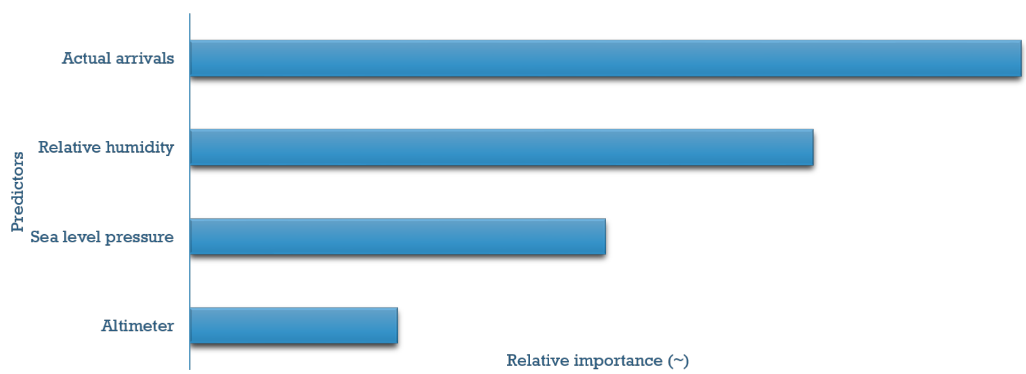

Figure 5.13: Predictor importance for Model Tree algorithm for predicting average taxi-in time delays during Ground Delay Programs

5.3.4 Artificial Neural Networks

Steps taken in R to develop a prediction model using the Artificial Neural Networks algorithm are as follows:

1. Load the data using the "read.csv" function [25]
2. normalize continuous variables to a 0 - 1 range since Neural Networks perform optimally when predictors are scaled to a narrow range
3. Train the model with using the "neuralnet" function [70] and the training dataset
4. Vary the number of hidden nodes to identify the optimal hidden node setting for the model. This is achieved by testing the performance of the model using the "compute" function [71] and the validation dataset
5. Compute Correlation using the "cor" function [68]
6. Compute the R-squared value by squaring Correlation
7. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values
8. Repeat steps 4 to 7 with the testing dataset

Varying the number of hidden nodes and evaluating the model's performance with the validation dataset revealed that the optimal number of hidden nodes was 2 using a logistic activation function which maps inputs into the range of 0 to 1 [72]. Figure 5.14 shows the optimal model's network with input nodes for each of the predictors, followed by two hidden nodes and one output node that predicts average taxi-in delay times during Ground Delay Programs. Bias terms, indicated by the blue lines allow the values at the nodes to be updated, like the intercept of a linear equation.

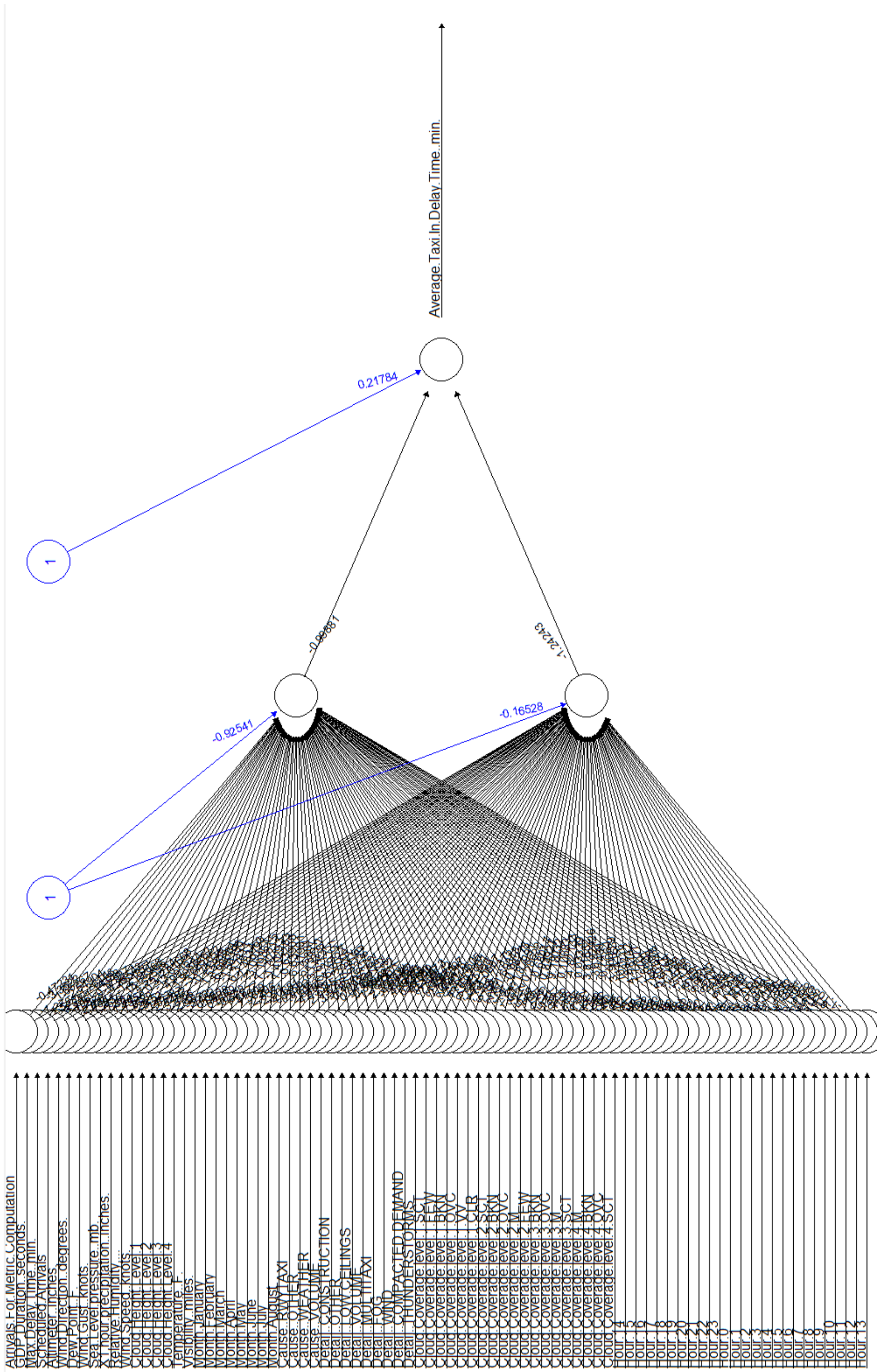

Figure 5.14: Visualization of Neural Network topology for predicting average taxi-in delay times during Ground Delay Programs

5.3.5 Support Vector Machines

Steps taken to develop a prediction model using the Support Vector Machines algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. The “svm” function [73] by default sets the maximum allowed error to 0.1, which often causes over-fitting. In order to reduce the incidence of over-fitting, a cost penalty can be applied to the models. Thus, the maximum allowed error and cost functions were varied from 0 to 1 in steps of 0.1, and 2^n where n varies from 0.5 to 8 in steps of 0.5, respectively to identify the optimal settings for the model

3. Train the model using the “e1071” package [74] and the training dataset

4. Test the performance of the model using the “predict” function [67] and the validation dataset

5. Compute Correlation using the “cor” function [68]

6. Compute the R-squared value by squaring Correlation

7. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

8. Repeat steps 4 to 7 with the testing dataset

Varying the maximum allowed error and cost functions and evaluating the model’s performance revealed that the optimal model had a maximum allowed error of 0 and cost of 2.83.

5.3.6 Random Forests

Steps taken to develop a prediction model using the Random Forests algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. Train the model using the “randomForest” function [75] and the training dataset

3. Test the performance of the model using the ”predict” function [67] and the validation dataset

4. Compute Correlation using the “cor” function [68]

5. Compute the R-squared value by squaring Correlation

6. Compute the Mean Absolute and Root Mean Squared Errors using the difference between the predicted and actual values

7. Repeat steps 3 to 6 with the testing dataset

Analysis of the Random Forests algorithm revealed that the actual number of arrivals, number of scheduled arrivals, altimeter pressure, and sea level pressure were the most influential predictors for this model as seen in Figure 5.15.

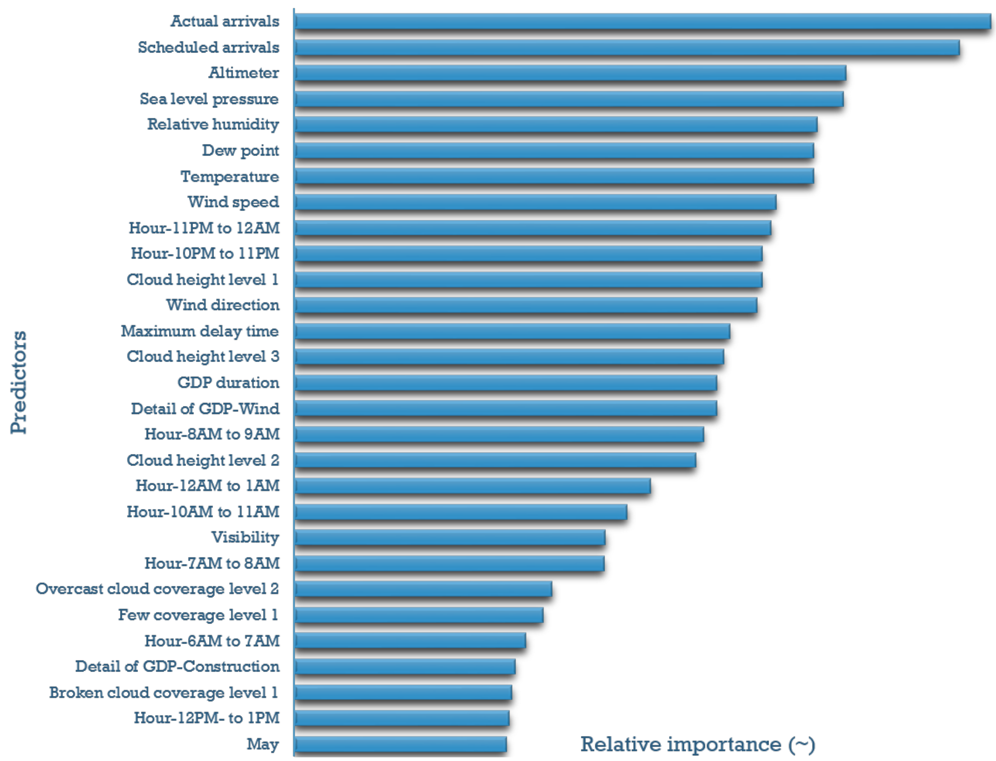

Figure 5.15: Predictor importance for Random Forest algorithm for predicting average taxi-in delay times during Ground Delay Programs

5.3.7 Summary

Tables 5.5 provides a summary of the evaluation of the performance of the different Machine Learning techniques with the validation set. It can be seen that the Multiple Linear Regression had the best performance, with the highest R-squared and correlation values, and the lowest Mean Absolute and Root Mean Squared Errors.

Table 5.5: Evaluation of technique performance for predicting the average taxi-in delay times during Ground Delay Programs with the validation dataset

Algorithm/Metric	R-squared	Mean Absolute Error (minutes)	Correlation	Root Mean Squared Error (minutes)
Multiple Linear Regression	0.66	1.59	0.81	2.1
Regression Trees	0.37	2.25	0.61	2.91
Model Trees	0.33	2.29	0.57	2.99
Artificial Neural Networks	0.44	2.31	0.66	2.93
Support Vector Machines	0.55	1.79	0.75	2.42
Random Forest Ensemble	0.56	1.86	0.75	2.45

Tables 5.6 provides a summary of the evaluation of the performance of the different Machine Learning techniques with the testing set. It can be seen that the Multiple Linear Regression had the best performance, with the highest R-squared and correlation values, and the lowest Root Mean Squared Error.

Table 5.6: Evaluation of technique performance for predicting the average taxi-in delay times during Ground Delay Programs with the testing dataset

Algorithm/Metric	**R-squared**	**Mean Absolute Error (seconds)**	**Correlation**	**Root Mean Squared Error (seconds)**
Multiple Linear Regression	0.6	1.93	0.77	2.56
Regression Trees	0.39	2.33	0.62	3.18
Model Trees	0.37	2.37	0.61	3.23
Artificial Neural Networks	0.38	2.49	0.62	3.33
Support Vector Machines	0.51	1.95	0.71	2.85
Random Forest Ensemble	0.58	1.86	0.76	2.63

Thus, **Multiple Linear Regression** was identified as the best suited algorithm for predicting average taxi-in delay times during Ground Delay Programs.

5.4 Predicting the occurrence of Ground Delay Programs

The objective of this model is to predict the occurrence of weather and volume-related Ground Delay Programs. Seven Machine Learning techniques were benchmarked to identify a suitable technique for the prediction model: Decision Trees, Naive Bayes, Classification Rule Learners, Support Vector Machines, Bagging Ensemble, Boosting Ensemble, and Random Forests. This section highlights steps taken to develop and tune the models using the aforementioned algorithms and provides an analysis of their performance with the validation and testing sets. In order to ensure that the algorithms were assessed accurately, the data was randomly divided into three categories: training, validation, and testing sets. The predictors for this model were the number of actual arrivals, number of scheduled arrivals, weather conditions, the month, and the hour. The training, validation, and testing datasets had 2940, 981, and 980 data points respectively.

5.4.1 Decision Trees

Steps taken in R to develop a prediction model using the Decision Trees algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. Train the model using the “C50” function [76, 25] and the training dataset

3. Test the performance of the model using the “predict” function [67] function and the validation dataset

4. Improve the performance of the model using adaptive boosting, “where multiple decision trees are built and the trees vote for the best class for each example” [25]. This involves adding a “trials” parameter when using the “C50” function. The optimal number of “trials” produces the lowest number of incorrect predictions

5. Create a confusion matrix and obtain evaluation metrics using the “confusionMatrix” function [77]

6. Repeat steps 3 to 5 with the testing dataset and the optimal number of “trials” obtained from step 4

Analysis of the Decision Tree algorithm revealed that the model had an average tree size of 72.9. Figure 5.16 shows that the month, altimeter pressure, dew point, sea level pressure, and visibility were the highest weighted predictors for this model, each contributing 6.254% as seen in Figure 5.16.

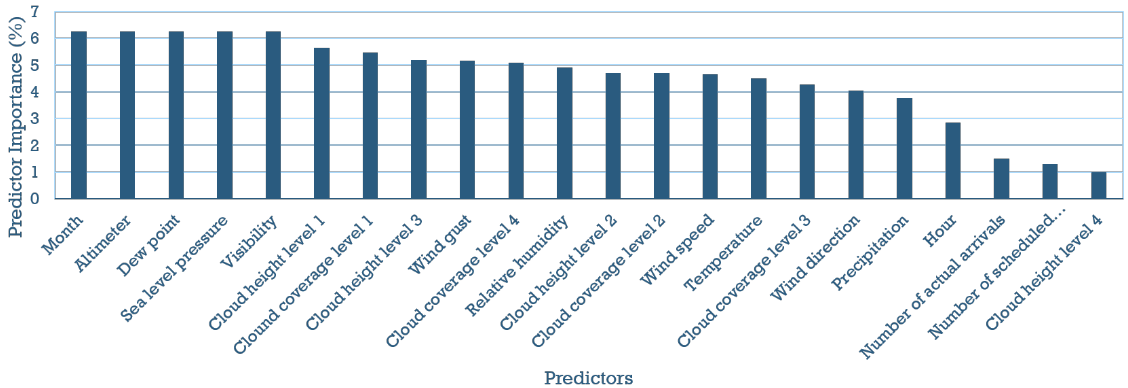

Figure 5.16: Predictor importance for Decision Tree algorithm for predicting the occurrence of Ground Delay Programs

Validation Dataset

Table 5.7 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.942, kappa statistic of 0.58, and a 95% Confidence Interval between 0.925 and 0.956, which is the range that the probability of a correct prediction lies within.

Table 5.7: Confusion matrix from Decision Tree algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	46	11	57
Predicted No GDP	41	883	924
Actual Total	87	894	981

Since the dataset is unbalanced, there was a need to further expand the evaluation of the model by analyzing how the model predicted volume-related Ground Delay Programs, weather-related Ground Delay Programs and no Ground Delay Programs. Table 5.8 shows the detailed confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively.

Table 5.8: Detailed confusion matrix from Decision Tree algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	11	1	1	13
Predicted Weather GDP	4	30	10	44
Predicted No GDP	11	30	883	924
Actual Total	26	61	894	981

From Table 5.8, it can be seen that the model accurately predicted 11 volume-related Ground Delay Programs, and incorrectly predicted 1 weather-related Ground Delay Program and no Ground Delay Program as volume-related Ground Delay Programs. The model also accurately predicted 30 weather-related Ground Delay Programs, and incorrectly predicted 4 volume-related Ground Delay Programs and 10 no Ground Delay programs as weather-related Ground Delayed Programs. Finally, the model accurately predicted 883 no Ground Delay Programs, and incorrectly predicted 11 volume-related Ground Delay Programs and 30 weather-related Ground Delay Programs as no Ground Delay Programs.

Table 5.9 summarizes the detailed evaluation of the Decision Tree algorithm's performance with the validation dataset. Low sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited when predicting volume and weather-related Ground Delay Programs. However, high sensitivity and moderate specificity of no Ground Delay Program predictions show that the model predicted the majority of no Ground Delay Program events accurately.

Table 5.9: Detailed evaluation of the Decision Tree algorithm with validation dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.423	0.492	0.988
Specificity	0.997	0.985	0.529
Precision	0.846	0.681	0.956
Recall	0.985	0.967	0.807

Testing Dataset

Table 5.10 shows the confusion matrix for the testing dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.922, kappa statistic of 0.531, and a 95% Confidence Interval between 0.903 and 0.938, which is the range that the probability of a correct prediction lies within.

Table 5.10: Confusion matrix from Decision Tree algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	51	18	69
Predicted No GDP	54	857	911
Actual Total	105	875	980

Table 5.11 shows the detailed confusion matrix for the testing dataset.

Table 5.11: Detailed confusion matrix from Decision Tree algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	9	2	2	13
Predicted Weather GDP	2	38	16	56
Predicted No GDP	22	32	857	911
Actual Total	33	72	875	980

From Table 5.11, it can be seen that the model accurately predicted 9 volume-related Ground Delay Programs, and incorrectly predicted 2 weather-related Ground Delay Programs and 2 no Ground Delay Programs as volume-related Ground Delay Programs. The model also accurately predicted 38 weather-related Ground Delay Programs, and incorrectly predicted 2 volume-related Ground Delay Programs and 32 no Ground Delay programs as weather-related Ground Delayed Programs. Finally, the model accurately predicted 857 no Ground Delay Programs, and incorrectly predicted 22 volume-related Ground Delay Programs and 32 weather-related Ground Delay Program as no Ground Delay Program.

Table 5.12 summarizes the detailed evaluation of the Decision Tree algorithm's performance with the testing dataset. Low/moderate sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and moderate specificity of no Ground Delay Program predictions show that the model predicted the majority of no Ground Delay Program events accurately.

Table 5.12: Detailed evaluation of the Decision Tree algorithm with testing dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.272	0.53	0.979
Specificity	0.996	0.98	0.486
Precision	0.692	0.678	0.941
Recall	0.975	0.963	0.739

Summary

Overall, with kappa statistic values of 0.580 and 0.531 from the validation and testing datasets respectively, the Decision Tree algorithm had an average performance which can be attributed to the unbalanced nature of the dataset.

5.4.2 Naive Bayes

Steps taken in R to develop a prediction model using the Naive Bayes algorithm are as follows:

1. Load the data using the "read.csv" function [25]
2. Train the model using the "naiveBayes" function [78, 25] and the training dataset
3. Test the performance of the model using the "predict" function [67] function and the validation dataset
4. Create a confusion matrix and obtain evaluation metrics using the "confusionMatrix" function [77]
5. Repeat steps 3 and 4 with the testing dataset

Validation Dataset

Table 5.13 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.448, kappa statistic of 0.0709, and a 95% Confidence Interval between 0.416 and 0.479, which is the range that the probability of a correct prediction lies within.

Table 5.13: Confusion matrix from Naive Bayes algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	72	511	583
Predicted No GDP	15	383	398
Actual Total	87	894	981

Table 5.14 shows the detailed confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively.

Table 5.14: Detailed confusion matrix from Naive Bayes algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	1	0	35	36
Predicted Weather GDP	16	55	476	547
Predicted No GDP	9	6	383	398
Actual Total	26	61	894	981

From Table 5.14, it can be seen that the model accurately predicted 1 volume-related Ground Delay Program, and incorrectly predicted 35 no Ground Delay Programs as volume-related Ground Delay Programs. The model also accurately predicted 55 weather-related Ground Delay Programs, and incorrectly predicted 16 volume-related Ground Delay Programs and 476 no Ground Delay Programs as weather-related Ground Delayed Programs. Finally, the model accurately predicted 383 no Ground Delay Programs, and inaccurately predicted 9 volume-related Ground Delay Programs and 6 weather-related Ground Delay Programs as no Ground Delay Program events.

Table 5.15 summarizes the detailed evaluation of the Naive Bayes algorithm's performance with the validation dataset. Low sensitivity and high specificity for volume-related Ground Delay Programs and no Ground Delay Program events show that the model's performance is limited in predicting volume-related Ground Delay Programs and no Ground Delay Program events. However, high sensitivity and low specificity show that the model predicted majority of weather-related Ground Delay Program events.

Table 5.15: Detailed evaluation of the Naive Bayes algorithm with the validation dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.038	0.902	0.428
Specificity	0.963	0.465	0.828
Precision	0.028	0.101	962
Recall	0.974	0.986	0.124

Testing Dataset

Table 5.16 shows the confusion matrix for the testing dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.449, Kappa Statistic of 0.0708, and a 95% Confidence Interval between 0.418 and 0.481, which is the range that the probability of a correct prediction lies within.

Table 5.16: Confusion matrix from Naive Bayes algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	80	499	579
Predicted No GDP	25	376	401
Actual Total	105	875	980

Table 5.17 shows the detailed confusion matrix for the testing dataset.

Table 5.17: Detailed confusion matrix from Naive Bayes algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	4	0	39	43
Predicted Weather GDP	16	60	460	536
Predicted No GDP	13	12	376	401
Actual Total	33	72	875	980

From Table 5.17, it can be seen that the model accurately predicted 4 volume-related Ground Delay Programs, and incorrectly predicted 39 no Ground Delay Program events as volume-related Ground Delay Programs. The model also accurately predicted 60 weather-related Ground Delay Programs, and incorrectly predicted 16 volume-related Ground Delay Programs and 460 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 376 no Ground Delay Program events, and incorrectly predicted 13 volume-related Ground Delay Programs and 12 weather-related Ground Delay Programs as no Ground Delay Program events.

Table 5.18 summarizes the detailed evaluation of the Naive Bayes algorithm's performance with the testing dataset. Low sensitivity and high specificity for volume-related Ground Delay Programs and no Ground Delay Program events show that the model's performance is limited in predicting volume-related Ground Delay Programs and no Ground Delay Program events. However, high sensitivity and low specificity show that the model predicted majority of weather-related Ground Delay Program events.

Table 5.18: Detailed evaluation of the Naive Bayes algorithm with testing dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**no GDP**
Sensitivity	0.121	0.833	0.429
Specificity	0.959	0.476	0.762
Precision	0.093	0.112	0.938
Recall	0.969	0.973	0.138

Summary

Overall, with kappa statistic values of 0.0709 and 0.0708 from the validation and testing datasets respectively, the Naive Bayes algorithm performed poorly.

5.4.3 Classification Rule Learners

Steps taken in R to develop a prediction model using the Classification Rule Learners algorithm are as follows:

1. Load the data using the "read.csv" function [25]

2. Train the model using the "JRip" function [79, 25] and the training dataset

3. Test the performance of the model using the “predict” function [67] function and the validation dataset

4. Create a confusion matrix and obtain evaluation metrics using the “confusionMatrix” function [77]

5. Repeat steps 3 and 4 with the testing dataset

Validation Dataset

Table 5.19 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.918, kappa statistic of 0.5, and a 95% Confidence Interval between 0.899 and 0.934, which is the range that the probability of a correct prediction lies within.

Table 5.19: Confusion matrix from Classification Rule Learners algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	48	37	85
Predicted No GDP	39	857	896
Actual Total	87	894	981

Table 5.20 shows the detailed confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively.

Table 5.20: Detailed confusion matrix from Classification Rule Learners algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	16	0	9	25
Predicted Weather GDP	4	28	28	60
Predicted No GDP	6	33	857	896
Actual Total	26	61	894	981

From Table 5.20, it can be seen that the model accurately predicted 16 volume-related Ground Delay Programs, and incorrectly predicted 9 no Ground Delay Program events as volume-related Ground Delay Programs. The model also accurately predicted 28 weather-related Ground Delay Programs, and incorrectly predicted 4 volume-related Ground Delay Programs and 28 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 857 no Ground Delay Program events, and incorrectly predicted 6 volume-related Ground Delay Programs and 33 weather-related Ground Delay Programs as no Ground Delay Program events.

Table 5.21 summarizes the detailed evaluation of the Classification Rule Learners algorithm's performance with the validation dataset. Moderate/low sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and moderate specificity of no Ground Delay Program predictions show that the model predicted the majority of no Ground Delay Program events accurately.

Table 5.21: Detailed evaluation of the Classification Rule Learners algorithm with the validation dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.615	0.459	0.959
Specificity	0.991	0.965	0.552
Precision	0.64	0.467	0.956
Recall	0.989	0.964	0.565

Testing Dataset

Table 5.22 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.895, kappa statistic of 0.444, and a 95% Confidence Interval between 0.874 and 0.913, which is the range that the probability of a correct prediction lies within.

Table 5.22: Confusion matrix from Classification Rule Learners algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	60	56	116
Predicted No GDP	45	819	864
Actual Total	105	875	980

Table 5.23 shows the detailed confusion matrix for the testing dataset.

Table 5.23: Detailed confusion matrix from Classification Rule Learners algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	19	0	10	29
Predicted Weather GDP	2	39	46	87
Predicted No GDP	12	33	819	864
Actual Total	33	72	875	980

From Table 5.23, it can be seen that the model accurately predicted 19 volume-related Ground Delay Programs, and incorrectly predicted 10 no Ground Delay Program events as volume-related Ground Delay Programs. The model also accurately predicted 39 weather-related Ground Delay Programs, and incorrectly predicted 2 volume-related Ground Delay Programs and 46 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 819 no Ground Delay Program events, and incorrectly predicted 12 volume-related Ground Delay Programs and 33 weather-related Ground Delay Program as no Ground Delay Program events.

Table 5.24 summarizes the detailed evaluation of the Classification Rule Learners algorithm's performance with the testing dataset. Moderate sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and moderate specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.24: Detailed evaluation of the Classification Rule Learners algorithm with testing dataset for predicting the occurrence of Ground Delay Programs

Metric	Volume-related GDP	Weather-related GDP	No GDP
Sensitivity	0.578	0.542	0.936
Specificity	0.989	0.947	0.571
Precision	0.655	0.448	0.948
Recall	0.985	0.963	0.517

Summary

Overall, with kappa statistic values of 0.5 and 0.444 with the validation and testing datasets respectively, the Classification Rule Learners had an average performance which can also be attributed to the unbalanced nature of the dataset.

5.4.4 Support Vector Machines

Steps taken in R to develop a prediction model using the Support Vector Machines algorithm are as follows:

1. Load the data using the "read.csv" function [25]
2. Train the model using the "ksvm" function [80, 25] and the training dataset
3. Test the performance of the model using the "predict" function [67], the "rbfdot" kernel (radial-based kernel), and the validation dataset
4. Create a confusion matrix and obtain evaluation metrics using the "confusionMatrix" function [77]
5. Repeat steps 3 and 4 with the testing dataset

Validation Dataset

Table 5.25 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.910, kappa statistic of 0.0173, and a 95% Confidence Interval between 0.891 and 0.927, which is the range that the probability of a correct prediction lies within.

Table 5.25: Confusion matrix from Support Vector Machines algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	1	2	3
Predicted No GDP	86	892	978
Actual Total	87	894	981

Table 5.26 shows the detailed confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively.

Table 5.26: Detailed confusion matrix from Support Vector Machines algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	0	0	0	0
Predicted Weather GDP	0	1	2	3
Predicted No GDP	26	60	892	978
Actual Total	26	61	894	981

From Table 5.26, it can be seen that the model accurately predicted 1 weather-related Ground Delay Program event, and incorrectly predicted 2 no Ground Delay Program events as weather-related Ground Delay Programs. The model also accurately predicted 892 no Ground Delay Program events, and incorrectly predicted 26 volume-related Ground Delay Programs and 60 weather-related Ground Delay Programs as no Ground Delay Program events.

Table 5.27 summarizes the detailed evaluation of the Support Vector Machine algorithm's performance with the validation dataset. Extremely low sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and extremely low specificity of no Ground Delay Program predictions shows that the model predicted majority of no Ground Delay Program events accurately.

Table 5.27: Detailed evaluation of the Support Vector Machines algorithm with the validation dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0	0.016	0.998
Specificity	1	0.998	0.011
Precision	N/A	0.33	0.912
Recall	0.974	0.939	0.333

Testing Dataset

Table 5.28 shows the confusion matrix for the testing dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.8969, kappa statistic of 0.081, and a 95% Confidence Interval between 0.876 and 0.915, which is the range that the probability of a correct prediction lies within.

Table 5.28: Confusion matrix from Support Vector Machines algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	5	1	6
Predicted No GDP	100	874	974
Actual Total	105	875	980

Table 5.29 shows the detailed confusion matrix for the testing dataset.

Table 5.29: Detailed confusion matrix from Support Vector Machines algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	0	0	0	0
Predicted Weather GDP	0	5	1	6
Predicted No GDP	33	67	874	974
Actual Total	33	72	875	980

From Table 5.29, it can be seen that the model accurately predicted 5 weather-related Ground Delay Programs, and incorrectly predicted 1 no Ground Delay Program event as a weather-related Ground Delay Program. The model also accurately predicted 874 no Ground Delay Program events, and incorrectly predicted 33 volume-related Ground Delay Programs and 67 weather-related Ground Delay Programs as no Ground Delay Program events.

Table 5.30 summarizes the detailed evaluation of the Support Vector Machines algorithm's performance with the testing dataset. Low sensitivity and moderate/high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, moderate sensitivity and low specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.30: Detailed evaluation of the Support Vector Machines algorithm with the testing dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0	0.069	0.999
Specificity	1	0.999	0.048
Precision	N/A	0.833	0.897
Recall	0.966	0.931	0.833

Summary

Overall, with poor kappa statistic values of 0.0173 and 0.0811 from the validation and testing datasets respectively, the Support Vector Machine algorithm performed poorly.

5.4.5 Bagging Ensemble

Steps taken in R to develop a prediction model using the Bagging Ensemble algorithm are as follows:

1. Load the data using the "read.csv" function [25]
2. Train the model using the "bagging" function [81, 25] and the training dataset
3. Test the performance of the model using the "predict" function [67] and the validation dataset

4. Create a confusion matrix and obtain evaluation metrics using the “confusionMatrix” function [77]

5. Repeat steps 3 and 4 with the testing dataset

Analysis of the Bagging Ensemble algorithm revealed that altimeter pressure, dew point, and sea level pressure were the highest weighted predictors for this model as seen in Figure 5.17.

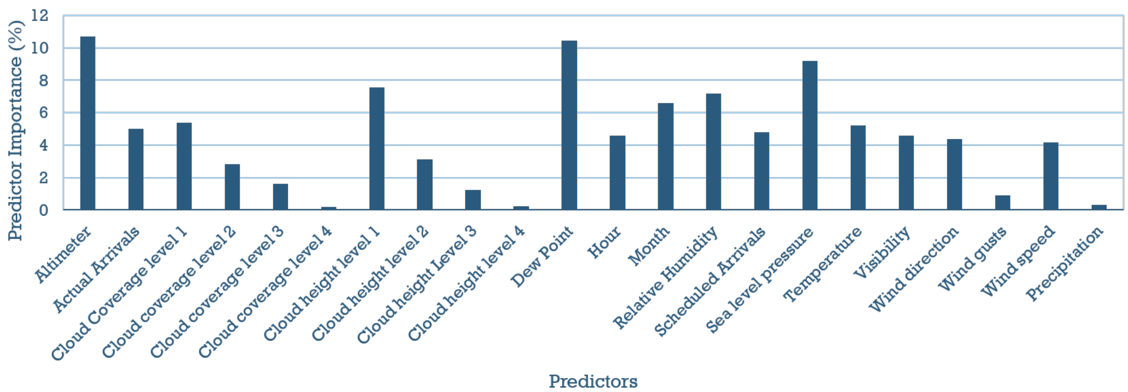

Figure 5.17: Predictor importance for Bagging Ensemble algorithm for predicting the occurrence of Ground Delay Programs

Validation Dataset

Table 5.31 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.937, kappa statistic of 0.474, and a 95% Confidence Interval between 0.919 and 0.951, which is the range that the probability of a correct prediction lies within.

Table 5.31: Confusion matrix from the Bagging Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	32	4	36
Predicted No GDP	55	890	945
Actual Total	87	894	981

Table 5.32 shows the detailed confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively.

Table 5.32: Detailed confusion matrix from the Bagging Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	11	0	0	11
Predicted Weather GDP	3	18	4	25
Predicted No GDP	12	43	890	945
Actual Total	26	61	894	981

From Table 5.32, it can be seen that the model accurately predicted 11 volume-related Ground Delay Programs. The model also accurately predicted 18 weather-related Ground Delay Programs, and inaccurately predicted 3 volume-related Ground Delay Programs and 4 no Ground Delay Programs events as weather-related Ground Delaye Programs. Finally, the model accurately predicted 890 no Ground Delay Program events, and incorrectly predicted 26 volume-related Ground Delay Programs and 61 weather-related Ground Delay Programs as no Ground Delay Program events.

Table 5.33 summarizes the detailed evaluation of the Bagging Ensemble algorithm's performance with the validation dataset. Low sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and low specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.33: Detailed evaluation of the Bagging Ensemble algorithm with the validation dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.423	0.295	0.996
Specificity	1	0.992	0.368
Precision	1	0.72	0.942
Recall	0.985	0.955	0.889

Testing Dataset

Table 5.34 shows the confusion matrix for the testing dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.901, kappa statistic of 0.268, and a 95% Confidence Interval between 0.881 and 0.919, which is the range that the probability of a correct prediction lies within.

Table 5.34: Confusion matrix from the Bagging Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	21	84	105
Predicted No GDP	12	863	875
Actual Total	33	947	980

Table 5.35 shows the detailed confusion matrix for the testing dataset.

Table 5.35: Detailed confusion matrix from the Bagging Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	10	0	2	12
Predicted Weather GDP	1	10	10	21
Predicted No GDP	22	62	863	947
Actual Total	33	72	875	980

From Table 5.35, it can be seen that the model accurately predicted 10 volume-related Ground Delay Programs, and incorrectly predicted 2 no Ground Delay Program events as volume-related Ground Delay Programs. The model also accurately predicted 10 weather-related Ground Delay Programs, and inaccurately predicted 1 volume-related Ground Delay Program and 10 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 863 no Ground Delay Programs, and incorrectly predicted 22 volume-related Ground Delay Programs and 62 weather-related Ground Delay Program as no Ground Delay Program events.

Table 5.36 summarizes the detailed evaluation of the Bagging Ensemble algorithm's performance with the testing dataset. Moderate sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and moderate specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.36: Detailed evaluation of the Bagging Ensemble algorithm with the testing dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.303	0.139	0.986
Specificity	0.998	0.988	0.2
Precision	0.833	0.476	0.911
Recall	0.976	0.935	0.634

Summary

Overall, with kappa statistic values of 0.474 and 0.268 from the validation and testing datasets respectively, the Bagging Ensemble had a moderate performance.

5.4.6 Boosting Ensemble

Steps taken in R to develop a prediction model using the Boosting Ensemble algorithm are as follows:

1. Load the data using the "read.csv" function [25]
2. Train the model using the "boosting" function [82, 25] and the training dataset

3. Test the performance of the model using the "predict" function [67], and the validation dataset

4. Create a confusion matrix and obtain evaluation metrics using the "confusionMatrix" function [77]

5. Repeat steps 3 and 4 with the testing dataset

Analysis of the Boosting Ensemble algorithm revealed that month, dew point, altimeter pressure, and sea level pressure were the highest weighted predictors for this model as seen in Figure 5.18.

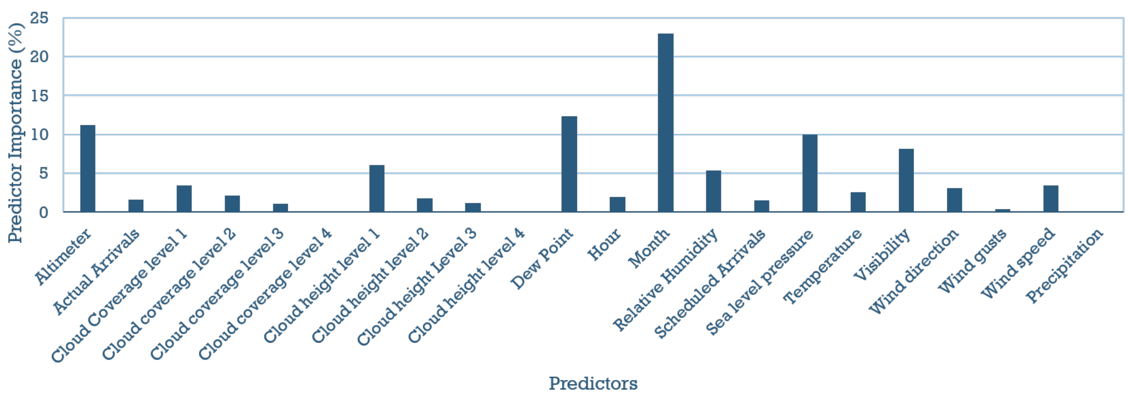

Figure 5.18: Predictor importance for Boosting Ensemble algorithm for predicting the occurrence of Ground Delay Programs

Validation Dataset

Table 5.37 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.948, kappa statistic of 0.629, and a 95% Confidence Interval between 0.932 and 0.961, which is the range that the probability of a correct prediction lies within.

Table 5.37: Confusion matrix from the Boosting Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	50	9	59
Predicted No GDP	37	885	922
Actual Total	87	894	981

Table 5.38 shows the detailed confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively.

Table 5.38: Detailed confusion matrix from the Boosting Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	14	1	0	15
Predicted Weather GDP	4	31	9	44
Predicted No GDP	8	29	885	924
Actual Total	26	61	894	981

From Table 5.38, it can be seen that the model accurately predicted 14 volume-related Ground Delay Programs, and incorrectly predicted 1 weather-related Ground Delay Program as a volume-related Ground Delay Program. The model also accurately predicted 31 weather-related Ground Delay Programs, and inaccurately predicted 4 volume-related Ground Delay Programs and 9 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 885 no Ground Delay Program events, and incorrectly predicted 8 volume-related Ground Delay Program and 29 weather-related Ground Delay Program events as no Ground Delay Program events.

Table 5.39 summarizes the detailed evaluation of the Boosting Ensemble algorithm's performance with the validation dataset. Moderate sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and moderate specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.39: Detailed evaluation of the Boosting Ensemble algorithm with the validation dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.538	0.508	0.989
Specificity	0.999	0.986	0.576
Precision	0.933	0.705	0.959
Recall	0.988	0.968	0.847

Testing Dataset

Table 5.40 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.943, kappa statistic of 0.657, and a 95% Confidence Interval between 0.926 and 0.957, which is the range that the probability of a correct prediction lies within.

Table 5.40: Confusion matrix from the Boosting Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	61	9	70
Predicted No GDP	44	866	910
Actual Total	105	875	980

Table 5.41 shows the detailed confusion matrix for the testing dataset.

Table 5.41: Detailed confusion matrix from the Boosting Ensemble algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	16	2	2	20
Predicted Weather GDP	1	42	7	50
Predicted no GDP	16	28	866	910
Actual Total	33	72	875	980

From Table 5.41, it can be seen that the model accurately predicted 16 volume-related Ground Delay Programs, and incorrectly predicted 2 weather-related Ground Delay Programs and 2 no Ground Delay Program events as volume-related Ground Delay Programs. The model also accurately predicted 42 weather-related Ground Delay Programs, and inaccurately predicted 1 volume-related Ground Delay Program and 7 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 866 no Ground Delay Program events, and incorrectly predicted 16 volume-related Ground Delay Program and 28 weather-related Ground Delay Program events as no Ground Delay Program events.

Table 5.42 summarizes the detailed evaluation of the Boosting Ensemble algorithm's performance with the testing dataset. Low/moderate sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and moderate specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.42: Detailed evaluation of the Boosting Ensemble algorithm with the testing dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.485	0.583	0.989
Specificity	0.996	0.991	0.581
Precision	0.8	0.84	0.952
Recall	0.982	0.968	0.871

Summary

Overall, with kappa statistics values of 0.629 and 0.65 from the validation and testing datasets respectively, the Boosting Ensemble performed well.

5.4.7 Random Forests

Steps taken in R to develop a prediction model using the Random Forest Ensemble algorithm are as follows:

1. Load the data using the “read.csv” function [25]

2. Train the model using the “randomForest” function [75, 25] and the training dataset
3. Test the performance of the model using the “predict” function [67] and the validation dataset
4. Create a confusion matrix and obtain evaluation metrics using the “confusionMatrix” function [77]
5. Repeat steps 3 and 4 with the testing dataset

Analysis of the Random Forests algorithm revealed that altimeter pressure, sea level pressure, the month, and dew point were the highest weighted predictors for this model as seen in Figure 5.19.

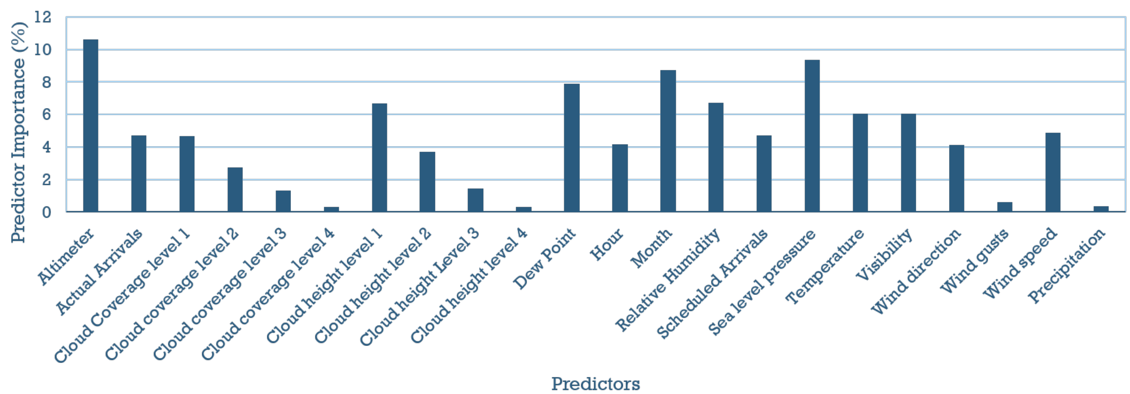

Figure 5.19: Predictor importance for Random Forests algorithm for predicting the occurrence of Ground Delay Programs

Validation Dataset

Table 5.43 shows the confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.944, kappa statistic of 0.559, and a 95% Confidence Interval between 0.928 and 0.957, which is the range that the probability of a correct prediction lies within.

Table 5.43: Confusion matrix from the Random Forest algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	40	4	44
Predicted No GDP	47	890	937
Actual Total	87	894	981

Table 5.44 shows the detailed confusion matrix for the validation dataset, where the last column and row represent the sum of predicted and actual events respectively.

Table 5.44: Detailed confusion matrix from the Random Forest algorithm for predicting the occurrence of Ground Delay Programs using the validation dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	10	1	0	11
Predicted Weather GDP	3	26	4	33
Predicted No GDP	13	34	890	937
Actual Total	26	61	894	981

From Table 5.44, it can be seen that the model accurately predicted 10 volume-related Ground Delay Programs, and incorrectly predicted 1 weather-related Ground Delay Program as a volume-related Ground Delay Program. The model also accurately predicted 26 weather-related Ground Delay Programs, and inaccurately predicted 3 volume-related Ground Delay Programs and 4 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 890 no Ground Delay Program events, and incorrectly predicted 13 volume-related Ground Delay Programs and 34 weather-related Ground Delay Programs as no Ground Delay Program events.

Table 5.45 summarizes the detailed evaluation of the Random Forest Ensemble algorithm's performance with the validation dataset. Low sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and low specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.45: Detailed evaluation of the Random Forest algorithm with the validation dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.385	0.424	0.996
Specificity	0.999	0.992	0.459
Precision	0.909	0.778	0.949
Recall	0.984	0.963	0.909

Testing Dataset

Table 5.46 shows the confusion matrix for the testing dataset, where the last column and row represent the sum of predicted and actual events respectively. The model had an accuracy of 0.927, kappa statistic of 0.508, and a 95% Confidence Interval between 0.908 and 0.942, which is the range that the probability of a correct prediction lies within.

Table 5.46: Confusion matrix from the Random Forest algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual GDP	**Actual No GDP**	**Predicted Total**
Predicted GDP	43	7	50
Predicted No GDP	62	868	930
Actual Total	105	875	980

Table 5.47 shows the detailed confusion matrix for the testing dataset.

Table 5.47: Detailed confusion matrix from the Random Forest algorithm for predicting the occurrence of Ground Delay Programs using the testing dataset

	Actual Volume GDP	**Actual Weather GDP**	**Actual No GDP**	**Predicted Total**
Predicted Volume GDP	7	2	2	11
Predicted Weather GDP	1	33	5	39
Predicted No GDP	25	37	868	930
Actual Total	33	72	875	980

From Table 5.47, it can be seen that the model accurately predicted 7 volume-related Ground Delay Programs, and incorrectly predicted 2 weather-related Ground Delay Programs and 2 no Ground Delay Program events as volume-related Ground Delay Programs. The model also accurately predicted 33 weather-related Ground Delay Programs, and inaccurately predicted 1 volume-related Ground Delay Program and 5 no Ground Delay Program events as weather-related Ground Delayed Programs. Finally, the model accurately predicted 868 no Ground Delay Program events, and incorrectly predicted 25 volume-related Ground Delay Programs and 37 weather-related Ground Delay Programs as no

Ground Delay Program events.

Table 5.48 summarizes the detailed evaluation of the Random Forests algorithm's performance with the testing dataset. Low sensitivity and high specificity for volume and weather-related Ground Delay Program predictions show that the model's performance is limited in predicting volume and weather-related Ground Delay Programs. However, high sensitivity and low specificity of no Ground Delay Program predictions show that the model predicted majority of no Ground Delay Program events accurately.

Table 5.48: Detailed evaluation of the Random Forest algorithm with the testing dataset for predicting the occurrence of Ground Delay Programs

Metric	**Volume-related GDP**	**Weather-related GDP**	**No GDP**
Sensitivity	0.212	0.458	0.992
Specificity	0.996	0.993	0.409
Precision	0.636	0.846	0.933
Recall	0.973	0.958	0.86

Summary

Overall, with kappa statistics values of 0.559 and 0.508 from the validation and testing datasets respectively, the Random Forests algorithm had a moderate performance.

5.4.8 Comparison of techniques

Since the dataset is heavily imbalanced, Accuracy is an inaccurate measure of the performance for these techniques. Kappa Statistic on the other hand, is appropriate for evaluating imbalanced datasets as it adjusts accuracy by accounting for the possibility of a correct prediction by chance alone [25]. The performance of the seven Machine Learning Techniques was thus compared using the Kappa Statistic. Figure 5.20 shows that the **Boosting**

Ensemble had the highest Kappa Statistic for both validation and testing datasets. Thus, it was the best suited algorithm for predicting the occurrence of Ground Delay Programs.

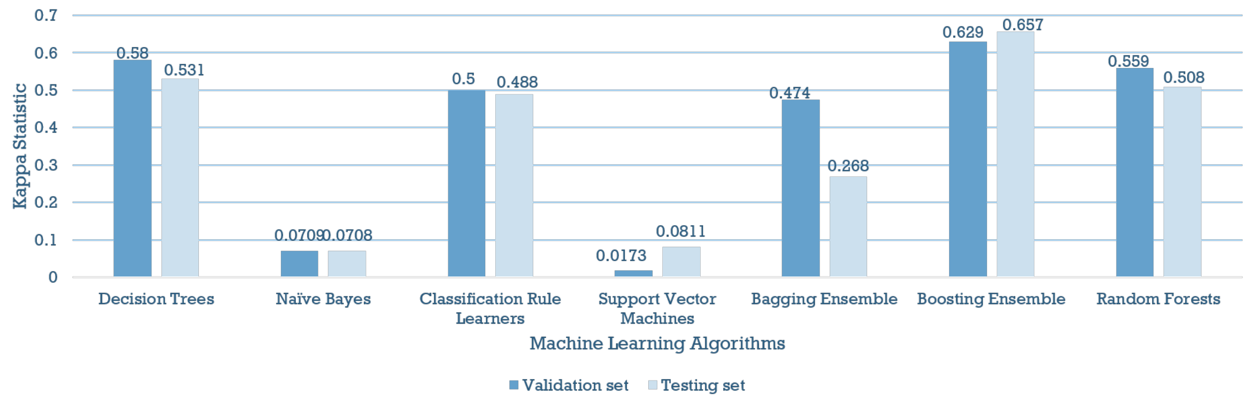

Figure 5.20: Comparison of Machine Learning techniques for predicting the occurrence of Ground Delay Programs using Kappa Statistic

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Review of Research Questions & Hypotheses

The review of prior research and research gaps discussed in section 2.3 led to the formulation of research questions and their associated hypotheses. This section focuses on using the results obtained from the previous chapter to assess the validity of the hypotheses to address the research gaps associated with each of these research questions:

Research Question 1.1: *Which Machine Learning techniques would lead to accurate predictions of flight delay times due to Ground Delay Programs?*

Research Question 1.2: *Which Machine Learning techniques would lead to accurate predictions of the duration of Ground Delay Programs?*

Research Question 1.3: *Which Machine Learning techniques would lead to accurate predictions of taxi-in delay times during Ground Delay Programs?*

Research Question 2: *Which Machine Learning technique(s) would lead to accurate predictions of the occurrence of Ground Delay Programs (GDP)?*

6.1.1 Research Questions 1.1, 1.2 & 1.3

The hypotheses associated with Research Questions 1.1, 1.2, and 1.3 are:

Hypothesis 1.1: *If dataset(s) containing comprehensive Ground Delay Program data are leveraged, then prediction models can be developed to predict the impact of Ground Delay Programs on flight and airport operations such as their duration, flight delay times, and taxi-in delay times*

Hypothesis 1.2: *If numerical prediction algorithms are developed and benchmarked, then prediction models can be developed to predict the impact of Ground Delay Programs* *on flight and airport operations such as their duration, flight delay times, and taxi-in time delays.*

Hypothesis 1.1

A major gap in current research is the inability of researchers to access and utilize detailed Ground Delay Program data for prediction models which has led to the development of mostly weather-related Ground Delay Program prediction models. The Traffic Flow Management System dataset provides data on Ground Delay Programs due to a variety of factors (weather, volume etc.) as well as specific details about these causes. It is important to note that the Traffic Flow Management System dataset is in FIXM format which is not suitable for data analytics purposes. FIXM files have "schema" or ".xsd" files, which dictate the structure of FIXM files. A parser was thus developed in python using the "ElementTree" API and the ".xsd" files to parse the TFMS datasets. This approach ensured that all required fields were extracted in their correct formats. It also facilitated the detection of errors in the datasets.

Leveraging the Traffic Flow Management System dataset thus led to the development of comprehensive models for predicting the impact of Ground Delay Programs on flight and airport operations such as their duration, flight delay times, and taxi-in delay times. Thus, hypothesis 1.1 was validated through the leveraging of the Traffic Flow Management System dataset.

Hypothesis 1.2

Another gap focuses on the lack of benchmarking of Machine Learning techniques in the prediction of Ground Delay Programs. The evaluation of prediction models developed and highlighted in the previous chapter revealed that the Random Forest was the appropriate model for predicting flight delay times due to Ground Delay Programs and the duration of Ground Delay Programs. Multiple Linear Regression was also identified as the appropriate model for predicting average taxi-in delay times during Ground Delay Programs. Hypothesis 1.2 was validated through the development and benchmarking of different numerical prediction algorithms for the prediction of the impact of Ground Delay Programs on flight and airport operations such as their duration, flight delay times, and taxi-in time delays.

6.1.2 Research Question 2

The hypotheses associated with Research Question 2 are:

Hypothesis 2.1: *If dataset(s) containing comprehensive Ground Delay Program data are leveraged, then a model can be developed to predict the occurrence of Ground Delay Programs*

Hypothesis 2.2: *If classification algorithms are developed and benchmarked, then the occurrence of Ground Delay Programs can be accurately predicted*

Hypothesis 2.1

As mentioned previously, a major gap associated with Ground Delay Program-related prediction models is the inability of researchers to develop comprehensive prediction models for Ground Delay Programs. Leveraging the Traffic Flow Management System dataset for this research ensured that volume-related Ground Delay Program data as well as weather-related Ground Delay Program data could be used as predictors, to predict the occurrence of volume and weather-related Ground Delay Programs. Thus, hypothesis 2.1 was validated through the leveraging of the Traffic Flow Management System dataset.

Hypothesis 2.2

Another research gap focuses on the lack of benchmarking of Machine Learning techniques to predict the occurrence of Ground Delay Programs. Prior work have leveraged one or two techniques for the prediction models. However, there is a need to assess the performance of different techniques to identify the appropriate or best suited technique for predicting the occurrence of Ground Delay Programs. Benchmarking the performance of seven classification algorithms revealed that Boosting Ensemble was the best suited technique for predicting the occurrence of Ground Delay Programs. Thus, hypothesis 2.2 was validated through the development and benchmarking of different classification algorithms for accurate predictions of the occurrence of Ground Delay Programs.

6.2 Future Work

6.2.1 Predicting airport capacity during Ground Delay Programs

Whenever Ground Delay Programs are initiated, the capacity of the affected airport is set as the "program rate" for that airport. This research did not incorporate airport capacity as a predictor in the prediction models. Future work will thus involve developing models to predict airport capacity (program rates), and using airport capacity as a predictor for the prediction models.

6.2.2 Predicting the impact of Ground Stops on flight and airport operations

Ground Stops are Traffic Management Initiatives that are implemented at an airport whenever air traffic demand is forecasted to exceed the airport's capacity for a short period of time [16]. Ground Stops, like Ground Delay Programs, impact flight and airport operations. The work carried out for this research can thus be extended to predict the impact of Ground Stops on flight and airport operations such as their duration, flight delay times, and taxi-in time delays.

6.2.3 Predicting the coincidence of Ground Delay Programs and Ground Stops

During Ground Delay Programs, flights are issued Expected Departure Clearance Times (EDCT) [18], which lead to delayed departure times. However, during Ground Stops, flights are grounded until conditions improve and the Ground Stop is terminated. Occasionally, Ground Stops are initiated while Ground Delay Programs are ongoing and vice-versa, whenever conditions change. The coincidence of these Traffic Management Initiatives (TMI) leads to prolonged delays, especially for en-route flights. There is thus a need to predict the coincidence of Ground Delay Programs and Ground Stops, as well as the duration of the coincidence. Doing so would enable airlines and passengers to plan appropriately and efficiently, leading to savings in fuel costs for airlines in particular, and reduction in the opportunity costs for passengers.

6.2.4 Predicting updates to the duration of Ground Delay Programs and Ground Stops

Often, the duration of Ground Delay Programs and Ground Stops may be updated, whenever conditions change. Thus, there is not only a need to improve the prediction of the duration of delays but also to predict the possibility of an update in the duration of a Ground Delay Program or Ground Stop. Doing so would go a long way in helping airlines and passengers in make more informed decisions.

Appendices

APPENDIX A
MACHINE LEARNING ALGORITHMS

A.1 Naive Bayes Classification

The Naive Bayes algorithm is based on Bayesian methods that utilize training data to calculate an observed probability of each outcome based on the evidence provided by feature values. Observed probabilities are then used to predict the most likely class for new features when the classifier is later applied to unlabeled data. This method has been largely used in classifying texts, such as email spam filtering, intrusion detection in computer networks and diagnosing medical conditions given a set of observed symptoms [25].

Bayesian classifiers are optimal for problems in which information from numerous attributes should be considered simultaneously in order to estimate the overall probability of an outcome [25]. The strengths and weaknesses of the Naive Bayes algorithm are highlighted in Table B.1 in Appendix B.

A.2 Decision Trees Classification

Decision Trees are powerful classifiers that utilize a tree structure to model the relationships among the features and the potential outcomes. A major benefit of Decision Tree algorithms is that the flowchart-like tree structure is not necessarily exclusive to the learner's internal use. After the model is created, most Decision Tree algorithms output the resulting structure which provides an insight into how and why the model works or does not work well for a particular task [25].

Decision Trees have been used for credit scoring models in which criteria that causes an applicant to be rejected is clearly documented and free from bias, marketing studies of customer behavior such as satisfaction, and diagnosis of medical conditions based on laboratory tests or symptoms. One of the most used Decision Tree algorithms is the C5.0 algorithm which has become an industry standard because it performs well with most types of problems [25]. The strengths and weaknesses of the C5.0 algorithm are highlighted in Table B.2 in Appendix B.

A.3 Classification Rule Learners

Classification Rule Learners represent knowledge in the form of logical if-else statements that assign a class to unlabeled examples. They form a hypothesis of "if this happens, then that happens." Classification Rule Learners are used in similarly to Decision Trees and have been used to identify conditions that led to hardware failure in mechanical devices, describe the key characteristics of groups of people for customer segmentation, and to find conditions that precede large drops or increases in the prices of shares on the stock market. The 1R algorithm has been widely used as a classification rule learner [25]. The strengths and weaknesses of the 1R algorithm are highlighted in Table B.3 in Appendix B.

A.4 Linear Regression

Linear regression is concerned with specifying the relationship between a single numeric dependent variable (the value to be predicted or target) and one or more numeric independent variables (the predictors). The simplest forms of regression assume that the relationship between independent and dependent variables follows a straight line. Regression analysis is typically used for modeling complex relationships among data elements, estimating the impact of a treatment on an outcome, and extrapolating into the future. Linear regression has been widely used in scenarios focused on examining how populations and individuals vary by their measured characteristics, quantifying the casual relationship between an event and its response(s), and identifying patterns that can be used to forecast future behavior. Regression has also been used for statistical hypothesis testing, which determines whether a premise is likely to be true or false, given observed data [25].

Most real-world regression problems have more than one independent variable. Thus, the Multiple Linear Regression method is typically used for most numeric prediction tasks. The strengths and weaknesses of Multiple Linear Regression are highlighted in Table B.4 in Appendix B.

A.5 Regression and Model Trees

Trees can also be used for numeric prediction and fall into two categories. The first, known as Regression Trees, were introduced in the 1980s as part of the Classification and Regression Tree (CART) algorithm. Regression Trees do not use linear regression methods. Instead, they make predictions based on the average value of instances that reach a tree's leaf. Regression Trees typically perform much better than linear models [83]. The second type of tree for numeric predictions is known as Model Trees. They are lesser known but are more powerful than regression trees. Model Trees are similar to Regression Trees, but at each leaf, a Multiple Linear Regression model is built from examples reaching that node [25]. The strengths and weaknesses of Regression and Model Trees are highlighted in Table B.5 in Appendix B.

A.6 Artificial Neural Networks

An Artificial Neural Network (ANN) models the relationship between a set of input signals and an output signal using a model derived from how a biological brain responds to stimuli from sensory inputs. Just as a brain uses a network of interconnected cells called neurons to create a massive parallel processor, Artificial Neural Networks use a network of artificial neurons or nodes to solve learning problems. Neural networks can be classified based on their activation function, network topology or training algorithm [25].

A neural network's activation function combines a neuron's input signals into a single output signal which is then distributed throughout the network [25]. A neural network's topology refers to the number of neurons and layers in the network, as well as the connections between the neurons and layers. An input node (neuron) processes a feature (value) from the dataset and transforms it into an output signal using the node's activation function. Finally, an output node generates predictions by using its own activation function and the signal received from the input node [25].

Information flow in a neural network is classified in two categories: feedforward and feedback. Feedforward networks involve feeding an input signal continuously in one direction from connection to connection until the signal reaches the output layer. These networks are flexible and permit the simultaneous modeling of multiple outcomes. A neural network with multiple hidden layers is referred to as a Deep Neural Network (DNN) and the practice of training a DNN is called deep learning. On the other hand, a feedback or recurrent network permits signals to travel in both directions, which is useful for learning complex patterns [25].

Artificial Neural Networks have been widely used to develop speech and handwriting recognition programs, automation of smart devices like an office building's environmental controls and sophisticated models of weather and climate patterns [25]. The strengths and weaknesses of Artificial Neural Networks are highlighted in Table B.6 in Appendix B.

A.7 Support Vector Machines

A Support Vector Machine can be imagined as a surface that creates a boundary between points of data plotted in multidimensional that represent examples and their feature values. The goal of a Support Vector Machine is to create a flat boundary called a hyperplane, which divides the space to create fairly homogeneous partitions on either side. An example of a hyperplane can be seen in Figure A.1. The figure shows a hyperplane or boundary that classifies objects as either triangles or rectangles in two dimensions. In this way, Support Vector Machines combine aspects of instance-based Nearest Neighbor learning models and Linear Regression models. Support Vector Machines are used extensively in pattern recognition tasks such as categorizing texts to identify the language used in a document, and the detection of rare yet important events like combustion engine failure [25]. The strengths and weaknesses of Support Vector Machines are highlighted in Table B.7 in Appendix B.

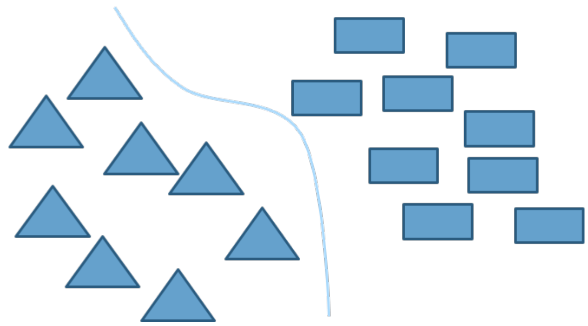

Figure A.1: Two dimensional hyperplane

A.8 Bagging, Boosting and Random Forests

Bagging is an ensemble-based method that generates a number of training datasets by bootstrap sampling the original training data. The datasets are then used to generate a set of models using a single learning algorithm [25].

Boosting is another ensemble-based method that boosts the performance of weak learners to attain the performance of stronger performers. Boosting involves constructing resampled datasets specifically to generate complementary learners. Then, each learner is given a vote based on its past performance. Boosting results in a model with performance that is often quite better than the best of the models in the ensemble [25].

Random Forests are another ensemble-based method that combines the principles of Bagging with random feature selection to add additional diversity to Decision Tree models. Random Forests are widely known for combining versatility and power into a single machine learning approach. As the ensemble only uses a small, random portion of the full dataset, Random Forests can handle extremely large datasets where dimensionality might cause other models to fail [25].

APPENDIX B

ADVANTAGES AND DISADVANTAGES OF THE MACHINE LEARNING ALGORITHMS CONSIDERED

Table B.1: Strengths and weaknesses of Naive Bayes algorithm [25]

Strengths	**Weaknesses**
Simple, fast, and very effective	Relies on an often-faulty assumption of equally important and independent features
Does well with noisy and missing data	Not ideal for datasets with many numeric features
Requires relatively few examples for training, but also works well with very large numbers of examples	Estimated probabilities are less reliable than the predicted classes
Easy to obtain the estimated probability for a prediction	

Table B.2: Strengths and weaknesses of C5.0 algorithm [25]

Strengths	**Weaknesses**
An all-purpose classifier that does well on most problems	Often biased towards splits on features having a large number of levels
Highly automatic learning process, which can handle numeric or nominal features, as well as missing data	Easy to over-fit or under-fit model
Excludes unimportant features	Can have trouble modeling some relationships die to reliance on axis-parallel splits
Can be used on both large and small datasets	Small changes in the training data can result in large changes to decision logic
Results in a model that can be interpreted without a mathematical background	Large trees can be difficult to interpret and the decisions they make may seem counterintuitive
More efficient than other complex models	

Table B.3: Strengths and weaknesses of 1R algorithm [25]

Strengths	Weaknesses
Generates easy-to-understand, human-readable rules	Uses only a single feature
Often performs surprisingly well	Probably overly simplistic
Can serve as a benchmark for more complex algorithms	

Table B.4: Strengths and weaknesses of Multiple Linear Regression [25]

Strengths	Weaknesses
The most common approach for modeling numeric data	Makes strong assumptions about the data
Can be adapted to model almost any modeling task	The model's form must be specified by the user in advance
Provides estimates of both the strength and size of the relationships among features and the outcome	Does not handle missing data
	Only works with numeric features, so categorical data requires extra processing

Table B.5: Strengths and weaknesses of Regression and Model Trees [25]

Strengths	Weaknesses
Combines the strengths of decision trees with the ability to model numeric data	Not as well-known as linear regression
Does not require the user to specify the model in advance	Requires a large amount of training data
Uses automatic feature selection, which allows the approach to be used with a very large number of features	Difficult to determine the overall net effect of individual features on the outcome
May fit some types of data better than linear regression	Large trees can become more difficult to interpret than a regression model
Does not require knowledge of statistics to interpret the model	

Table B.6: Strengths and weaknesses of Artificial Neural Networks [25]

Strengths	Weaknesses
Can be adapted to classification or numeric prediction problems	Extremely computationally intensive and slow to train, particularly if the network's topology is complex
Capable of modeling more complex patterns than nearly any algorithm	Very prone to over-fitting training data
Makes few assumptions about the data's underlying relationships	Results in a complex black box model that is difficult, if not impossible, to interpret

Table B.7: Strengths and weaknesses of Support Vector Machines [25]

Strengths	Weaknesses
Can be used for classification or numeric prediction problems	Finding the best model requires testing of various combinations of kernels and model parameters
Not overly influenced by noisy data and not very prone to over-fitting	Can be slow to train, particularly if the input dataset has a large number of features or examples
May be easier to use than neural networks, particularly due to the existence of several well-supported SVM algorithms	Results in a complex black box model that is difficult, if not impossible, to interpret
Gaining popularity due to its high accuracy and high-profile wins in data mining competitions	

APPENDIX C

CODES

All of the scripts developed in Python and R can be accessed in GitHub:

https://github.gatech.edu/emangortey3/MS-Thesis

REFERENCES

[1] Casey, Jamie and Norwood, Janet L., *Key Transportation Indicators Summary of a Workshop*. National Academy Press, 2002.

[2] Bureau of Transportation statistics, *Airline on-time statistics and delay causes*, `https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp?pn=1`, 2018.

[3] Transportation Research Board, *Defining and Measuring Aircraft Delay and Airport Capacity Thresholds*. National Academy Press, 2014.

[4] Bureau of Transportation Statistics, *Airline On-Time Statistics and Delay Causes*, `https://www.transtats.bts.gov/OT_Delay/ot_delaycause1.asp?type=1&pn=1`, 2018.

[5] Peterson, Everett B., “The economic cost of airline flight delay”, *Journal of Transport Economics and Policy*, vol. 47, pp. 107–121, 1 2013.

[6] Office of Aviation Enforcement and Proceedings, *Air Travel Consumer Report*. U.S. Dept. of Transportation, 2018.

[7] Ball, Michael and Barnhart, Cynthia, *Total Delay Impact Study: A Comprehensive Assessment of the Costs and Impacts of Flight Delay in the United States*. NEXTOR, 2010.

[8] General Accounting Office, *National Airspace System: FAA Has Implemented Some Free Flight Initiatives, but Challenges Remain: Report to Congressional Requesters.* United States, Congress, 1998.

[9] Bureau of Transportation Statistics, *Causes of national aviation system delays national (may, 2018)*, `https://www.transtats.bts.gov/OT_Delay/ot_delaycause1.asp?type=5&pn=1`, 2018.

[10] US Department of Transportation, *Airline delays and consumer issues.* `www.transportation.gov/content/airline-delays-and-consumer-issues-0`, 2017.

[11] Bureau of Transportation Statistics, *Airline on-time performance and causes of flight delays*, `https://www.bts.gov/explore-topics-and-geography/topics/airline-time-performance-and-causes-flight-delays`, 2018.

[12] Office of Secretary, Department of Transportation, *Reporting the Causes of Airline Delays and Cancellations*. 14 CFR Part 234, Rule OST 2000-8164, 2002.

[13] ASIAS, *ASIAS*, `https://portal.asias.aero/web/guest`, 2018.

[14] Bureau of Transportation Statistics, *Delay causes by year as a percent of total delay minutes*, `https://www.bts.gov/content/delay-cause-year-percent-total-delay-minutes`, 2018.

[15] Mangortey, Eugene and Gilleron, Jerome and Dard, Ghislain and Pinon-Fischer, Olivia and Mavris, Dimitri, "Development of a data fusion framework to support the analysis of aviation big data", *AIAA Scitech 2019 Forum, AIAA SciTech Forum, (AIAA 2019-1538)*, 2019.

[16] D. Robyn, *Reforming the air traffic control system to promote efficiency and reduce delays*. The Brattle Group, 2007.

[17] Federal Aviation Administration, *Enhanced traffic management system*, `https://www.fly.faa.gov/Products/Information/ETMS/etms.html`, 2017.

[18] Federal Aviation Administration, *Expected departure clearance times*, `http://aspmhelp.faa.gov/index.php/Expect_Departure_Clearance_Times_(EDCT)`, 2017.

[19] Federal Aviation Administration.

[20] Federal Aviation Administration, *Air traffic control system command center*, `https://www.fly.faa.gov/adv/advAdvisoryForm.jsp`, 2019.

[21] Federal Aviation Administration, *Aspm airport analysis: Definitions of variables*, `https://aspmhelp.faa.gov/index.php/ASPM_Airport_Analysis:_Definitions_of_Variables`, 2019.

[22] Zhou, Tan, Gao, Qiang, Chen, Xin, and Xun, Zongwei, "Flight delay prediction based on characteristics of aviation network", *MATEC Web Conf.*, vol. 259, p. 02 006, 2019.

[23] Burmester, Gerrit and Ma, Hui and Steinmetz, Dietrich and Hartmannn, Sven, "Big data and data analytics in aviation", in *Advances in Aeronautical Informatics*, Springer, 2018, pp. 55–65.

[24] Marr, Bernard, *What is big data? a super simple explanation for everyone.* `http://www.space.com/2995-nasa-jet-bears-nose-grows-sonic-boom-tests.html`, 2012.

[25] Lantz, Brett, *Machine Learning with R: Discover How to Build Machine Learning Algorithms, Prepare Data, and Dig Deep into Data Prediction Techniques with R*. Packt Publishing, 2015.

[26] Maglogiannis, Ilias and and Karpouzis, Kostas and Wallace, Manolis, *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies - Volume 160 Frontiers in Artificial Intelligence and Applications*. I O S Press, Incorporated, 2007.

[27] Avijit, Mukherjee, *Predicting Ground Delay Program At An Airport Based On Meteorological Conditions*. 14th AIAA Aviation Technology, Integration, and Operations Conference, note = `doi:10.2514/6.2014-2713`, 2014.

[28] Smith, David A. and Sherry, Lance, *Decision Support Tool for Predicting Aircraft Arrival Rates from Weather Forecasts*. 2008 Integrated Communications, Navigation and Surveillance Conference, doi:10.1109/icnsurv.2008.4559186, 2008.

[29] Avijit, Mukherjee, *Ground Delay Program Planning under Uncertainty in Airport Capacity*. Transportation Planning and Technology, vol. 35, no. 6, pp. 611–628., note =`doi:10.1080/03081060.2012.710031`, 2012.

[30] Liu, Jixin, *Optimizing Key Parameters of Ground Delay Program with Uncertain Airport Capacity*. Journal of Advanced Transportation, vol. 2017, pp. 1–9., note = `doi:10.1155/2017/7494213`, 2017.

[31] Yi, Liu and Hansen, Mark, *Predicting the Initiation of a Ground Delay Program*. Journal of Aerospace Operations, vol. 5, no. 1-2, pp. 75–84., doi:10.1233/aop-170073, 2018.

[32] Bloem, Michael and Bambos, Nicholas, *Ground Delay Program Analytics with Behavioral Cloning and Inverse Reinforcement Learning*. Journal of Aerospace Information Systems, vol. 12, no. 3, 2015, pp. 299–313., doi:10.2514/1.i010304, 2015.

[33] National Transportation Systems Center, *SWIM Flight Data Publication Service (SFDPS) Reference Manual*. National Transportation Systems Center, 2015.

[34] Federal Aviation Administration, *Traffic flow management system reference manual. 8.9 version*, 2011.

[35] Dr. Eric Kostelich, *Metar format (fm-15) surface meterological airways format*, 2018.

[36] Asencio, M. A., *Clustering Approach for Analysis of Convective Weather Impacting the NAS*. 12th Integrated Communications, Navigation, and Surveillance Conference, Herndon, Virginia, 2012.

[37] Liu P-c, B., *Scenario-Based Air Traffic Flow Management: From Theory to Practice*. From Theory to Practice, Transportation Research - Part B, Vol.42, 2008, pp. 685-702, 2010.

[38] Stolzer, Alan J., *Safety Management Systems in Aviation.*, 2010.

[39] Federal Aviation Administration, *Aviation system performance metrics*, `https://aspm.faa.gov/apm/sys/AnalysisAP.asp`, 2019.

[40] Federal Aviation Administration, *JAVA MESSAGING SERVICE DESCRIPTION DOCUMENT Traffic Flow Management Data Service (TFMData) Vol. 2.0.5*. Federal Aviation Administration, 2016.

[41] Apache Hadoop, *Welcome to Apache™ Hadoop®!*, `hadoop.apache.org/`, 2018.

[42] The Apache Software Foundation, *Apache Nifi*, `https://nifi.apache.org/`, 2018.

[43] Federal Aviation Administration, *Aspm 77*, `https://aspmhelp.faa.gov/index.php/ASPM_77`, 2019.

[44] Federal Aviation Administration, *Aspm carriers*, `https://aspmhelp.faa.gov/index.php/ASPM_Carriers`, 2019.

[45] Federal Aviation Administration, *Aviation system performance metrics (aspm)*, `https://aspmhelp.faa.gov/index.php/Aviation_Performance_Metrics_%28APM%29`, 2019.

[46] National Weather Service, *Automated surface observing systems*, `https://www.weather.gov/asos/asostech`, 2019.

[47] Guttman, Nathaniel and Baker, Bruce, "Exploratory analysis of the difference between temperature observations recorded by asos and conventional methods", *Bulletin of American Meteorological Society*, 1996.

[48] Iowa State University, *Asos-awos-metar data download*, `https://mesonet.agron.iastate.edu/request/download.phtml`, 2019.

[49] Lepori, Hubert., *Introduction to FIXM*. https://www.icao.int/MID/Documents/2017/SWIM Interregional/8.2 Introduction to FIXM.pdf, 2017.

[50] Python Software Foundation, *19.7. xml.etree.elementtree - the elementtree xml api*, `docs.python.org/2/library/xml.etree.elementtree.html`, 2019.

[51] Klein, Lawrence A, *Sensor and data fusion: a tool for information assessment and decision making*. SPIE, 2012.

[52] Cummins, Chris and Petoumenos, Pavlos and Wang, Zheng and Leatheri, Hugh, "End-to-end deep learning of optimization heuristics", *2017 26th International Conference on Parallel Architectures and Compilation Techniques (PACT), Portland, OR, pp. 219-232*, 2017.

[53] Hung, S. and Adeli, H., "A parallel genetic/neural network learning algorithm for mimd shared memory machines", *IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 900-909*, 1994.

[54] G. Hackeling, *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd., 2017.

[55] Duvenaud, David K and Maclaurin, Dougal and Iparraguirre, Jorge and Bombarell, Rafael and Hirzel, Timothy and Aspuru-Guzik, Alan and Adams, Ryan P, "Convolutional networks on graphs for learning molecular fingerprints", in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 2224–2232.

[56] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data", in *Advances in Information Retrieval*, N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, and G. Silvello, Eds., Cham: Springer International Publishing, 2016, pp. 45–57, ISBN: 978-3-319-30671-1.

[57] Feurer, Matthias and Klein, Aaron and Eggensperger, Katharina and Springenberg, Jost and Blum, Manuel and Hutter, Frank, "Efficient and robust automated machine learning", in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 2962–2970.

[58] Ron Kohavi, *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. International Joint Conference on Artificial Intelligence (IJCAI), 1995.

[59] Cameron, Colin and Windmeijer, Frank, "An r-squared measure of goodness of fit for some common nonlinear regression models", *Journal of Econometrics*, vol. 77, no. 2, pp. 329 –342, 1997.

[60] Benesty, J. and Chen, J. and Huang, Y., "On the importance of the pearson correlation coefficient in noise reduction", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 757–765, 2008.

[61] T. Chai and R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature", *Geosci. Model Dev., 7, 1247–1250*, 2014.

[62] The R Foundation, *The r project for statistical computing*, `https://www.r-project.org/`, 2019.

[63] RDocumentation, *Fitting linear models*, `https://www.rdocumentation.org/packages/stats/versions/3.5.2/topics/lm`, 2019.

[64] RDocumentation, *Accessing linear model fits*, `https://www.rdocumentation.org/packages/stats/versions/3.5.2/topics/lm.summaries`, 2019.

[65] Chi Yau, *Coefficient of determination*, `http://www.r-tutor.com/elementary-statistics/simple-linear-regression/coefficient-determination`, 2019.

[66] RDocumentation, *Recursive partitioning and regression trees*, `https://www.rdocumentation.org/packages/rpart/versions/4.1-13/topics/rpart`, 2019.

[67] RDocumentation, *Spatial model predictions*, `https://www.rdocumentation.org/packages/raster/versions/2.8-19/topics/predict`, 2019.

[68] RDocumentation, *Correlation, variance and covariance (matrices)*, `https://www.rdocumentation.org/packages/stats/versions/3.5.3/topics/cor`, 2019.

[69] RDocumentation, *R/weka classifier trees*, `https://www.rdocumentation.org/packages/RWeka/versions/0.1-0/topics/Weka_classifier_trees`, 2019.

[70] RDocumentation, *Training of neural networks*, `https://www.rdocumentation.org/packages/neuralnet/versions/1.44.2/topics/neuralnet`, 2019.

[71] RDocumentation, *Compute*, `https://www.rdocumentation.org/packages/dplyr/versions/0.5.0/topics/compute`, 2019.

[72] A Vehbi Olgac and B. Karlik, “Performance analysis of various activation functions in generalized mlp architectures of neural networks”, *International Journal of Artificial Intelligence And Expert Systems*, vol. 1, pp. 111–122, Feb. 2011.

[73] RDocumentation, *Support vector machines*, `https://www.rdocumentation.org/packages/e1071/versions/1.7-0.1/topics/svm`, 2019.

[74] RDocumentation, *E1071*, `https://www.rdocumentation.org/packages/e1071/versions/1.7-0.1`, 2019.

[75] RDocumentation, *Classification and regression with random forest*, `https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest`, 2019.

[76] RDocumentation, *C5.0 decision trees and rule-based models*, `https://www.rdocumentation.org/packages/C50/versions/0.1.2`, 2019.

[77] RDocumentation, *Confusionmatrix*, `https://www.rdocumentation.org/packages/caret/versions/3.45/topics/confusionMatrix`, 2019.

[78] RDocumentation, *Naive bayes classifier*, `https://www.rdocumentation.org/packages/e1071/versions/1.7-0.1/topics/naiveBayes`, 2019.

[79] RDocumentation, *R/weka rule learners*, `https://www.rdocumentation.org/packages/RWeka/versions/0.4-40/topics/Weka_classifier_rules`, 2019.

[80] RDocumentation, *Support vector machines*, `https://www.rdocumentation.org/packages/kernlab/versions/0.9-27/topics/ksvm`, 2019.

[81] RDocumentation, *Bagging classification and regression trees*, `https://www.rdocumentation.org/packages/ipred/versions/0.4-0/topics/bagging`, 2019.

[82] RDocumentation, *Boostings*, `https://www.rdocumentation.org/packages/adabag/versions/4.2/topics/boosting`, 2019.

[83] Witten, I. and Eibe, Frank and Hall, Mark and Pal, Christopher, *Data mining practical machine learning tools and techniques*. Morgan Kaufmann Publishers, 2005.