A computational intelligence based prediction model for flight departure delays

by

JOHANNA MMAKWENA HOPANE

A dissertation submitted in fulfilment for the Degree

of

Magister Commercii

in

Information Technology Management

College of Business and Economics

UNIVERSITY OF JOHANNESBURG

Supervisor: DR B.N Gatsheni

2019

DECLARATION

I certify that the dissertation submitted by me for the degree Masters of Commerce (Information Technology Management) at the University of Johannesburg is my independent work and has not been submitted by me for a degree at another university.

JOHANNA MMAKWENA HOPANE

ACKNOWLEDGEMENTS

I hereby like to express my gratitude to the following individuals who enabled this document to be successfully and timeously completed:

- First, GOD
- Supervisor, Dr B.N. Gatsheni
- My family and friends
- Faculty of Management for funding me with the NRF Supervisor-linked bursary

ABSTRACT

Flight departure delays are a major problem at OR Tambo International airport (ORTIA). There is a high delay for flights to depart, especially at the beginning of the month and at the end of the month. The increasing demand for flights departing at ORTIA often leads to a negative effect on business deals, individuals' health, job opportunities and tourists. When flights are delayed departing, travellers are notified at the airport every 30 minutes about the status of the flight and the reason the flight is delayed if it is known. This study aims to construct a flight delays prediction model using machine learning algorithms. The flight departures data were obtained from ORTIAs website timetable for departing flight schedules. The flight departure data for ORTIA to any destination (i.e. Johannesburg (JNB) Airport to Cape Town (CPT)) for South African Airways (SAA) airline was used for this study. Machine learning algorithms namely Decision Trees (J48), Support Vector Machine (SVM), K-Means Clustering (K-Means) and Multi-Layered Perceptron (MLP) were used to construct the flight departure delays prediction models. A cross-validation (CV) method was used for evaluating the models. The best prediction model was selected by using a confusion matrix. The results showed that the models constructed using Decision Trees (J48) achieved the best prediction for flight departure delays at 67.144%, while Multi-layered Perceptron (MLP) obtained 67.010%, Support Vector Machine (SVM) obtained 66.249% and K-Means Clustering (K-Means) obtained 61.549%. Travellers wishing to travel from ORTIA can predict flight departure delays using this tool. This tool will allow travellers to enter variables such as month, week of month, day of week and time of day. The entered variables will predict the flight departure status by examining target concepts such as On Time, Delayed and Cancelled. The travellers will only be able to predict flight departures status, although they will not have full knowledge of the flight departures volume. In that case, they will depend on the flight information display system (FIDS) board. This study can predict and empower travellers by providing them with a tool that can determine the punctuality of the flights departing from ORTIA.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

OR Tambo International Airport (ORTIA) in Ekurhuleni, Johannesburg is the biggest airport in Africa. It accommodates 28 million passengers annually and is the busiest airport situated in South Africa with about 220 934 aircraft movement with in Africa. It is known to be Africa's portal to the world. No other airport in Africa can handle as many domestic and international flights (Gauteng Tourism Authority, 2017). There are parallel north-south runways, namely the western runway, which is 03L/21R of 14 495 ft and 4 418 m and 03R/21L 11 155 ft and 3 400 m, making it one of the world's longest international runways. The airport has a $3.2 billion economic impact and 128.2 thousand social impact for travelling passenger's statistics. All the ORTIA's major airlines (Airlink, Comair, FlySafair, South African Airways, South African Express, Mango and Kulula) land at ORTIA, "and you can catch a flight to any regional and most international destinations from the airport" (Gauteng Tourism Authority, 2017). The seven major airlines operate from ORTIA (Excelsior Digital, 2016). The airport is the hub of South Africa's largest carrier, South African Airways (SAA), and several of the smaller local airlines (Wikimedia Foundation, 2017).

ORTIA is expected to suffer from considerable congestion because of the expected 1% gross domestic product (GDP) growth for 2017 (from the forecast of 1.2% at the March meeting); 1.5% in 2018 (from 1.7% expected in March); and 1.7% in 2019 (from 2% forecast in March). "Global economy is to grow 2.7% this year" (Tiso Blackstar Group, 2017). A rebound in net exports is expected. Growth in regions with large numbers of commodity exporters will strengthen in 2017. Growth in Sub-Saharan Africa is forecast to pick up to 2.6% in 2017, and average 3.4% in 2018-2019" (Tiso Blackstar Group, 2017), which leads to more investments and business in South Africa. In this case, air transportation is likely to be used more frequently. The airport's flight departure delays have been problematic in ORTIA, where it is the core airport for the South African Airways (SAA).

This dissertation focuses on departure delays from ORTIA to domestic and international destination. The lack of departure punctuality during the week affects the economy negatively, as the estimation is that more than 17% of ORTIA's flights depart 15 minutes late, which leads to departure delays. Businesses are also being negatively affected, as business meetings are cancelled, and interviews are missed resulting in missed opportunities. The health of individuals is negatively affected as medical emergency services are delayed from reaching individuals whose lives are in peril. The main attributes that contribute to flight departure delays at ORTIA are aircraft arriving late, air carrier delay, national aviation system delay, extreme weather, security delays, equipment, closed runways and the volume of airlines (Tiso Blackstar Group, 2017).

Computational Intelligence, however, has been used in tackling the challenges in the industry for different purposes with much success. Johnson and Savage (2006) provided a solution to flight departure delays. The use of regression techniques using an average variable cost (AVC) function has been used to reduce flight departure delays; a steady-state bottleneck congestion model that is used for highways (Walters, 1961).

Dynamics simulation was used to construct a model for flight delays reduction patterns (Bakhshandeh, Shahgholian & Shahraki, 2013). This model had a set of factors (push back facilities, lack of materials, crisis management, the number and size of parking space plane) and parameters (crew delay, weather conditions, human error, airport size) in the air transportation industry that can be effective in reducing flight delays that were enumerated. With this model, it can be concluded that all the factors and parameters are involved, directly and indirectly, in reducing and/or increasing flight delays.

The econometric model by Rosen (2002) was developed "to estimate one component of the marginal benefit of airport infrastructure by quantifying the impact of changes in customer demand on travel time with a fixed level of airport infrastructure". The model looked at the impact of the size of an air carrier's network on the length and variation of travel time. Given a constant level of fixed airline capital, increases in passenger demand for air travel should translate into longer average flight times across all carriers. Additional customers mean that airport terminals and runways are more congested, planes take longer to load with passengers and baggage, and resources such as planes and crews must be used more frequently to cope with the demand. “These factors make it more difficult for the airlines to run an on-time flight” (Rosen, 2002).

1.2 Background to OR Tambo international airport and its air transportation

OR Tambo International Airport (ORTIA) “is the busiest airport (handles 28 million passengers and employs more than 1 800 people) in South Africa and is a major regional hub in the aviation industry” (Peck, 2015). It is also located “where significant weather, such as thunderstorms are frequent occurrences and, thus, weather, as a geographic phenomenon, is a critical aspect of its’ aviation operations” (Peck, 205). It, therefore, is “an ideal airport to base a study regarding adverse aviation weather and its impacts on aviation operations, with specific reference to delays” (Peck, 2015). This airport conducts direct flights to all continents with the exception of Antarctica. It serves “as the primary airport for domestic and international travel to/from South Africa” (Wikimedia Foundation, 2017).

“ORTIA has two terminals handling domestic and international flights. Terminal A handles International traffic and Terminal B Domestic flights. Both consists of two levels with departures on the upper level and arrival on the lower level. There is no shortage of shopping opportunities for visitors and it easily compares with some of Johannesburg’s bigger malls in terms of shopping and dining options” (Gauteng Tourism Authority, 2017).

A shortage of officials to man the immigration desks is continuing to cause delays more than two months after the capturing of biometric data was rolled out (Tourism Update, 2016). Technical issues with the air traffic navigation system causes communication challenges, resulting in flight delays (Independent Media, 2017). South Africa “is a developing country and a major economic role-player on the African continent” (Leonard & Bekker, 2013). After the political change of 1994, “international sanctions were lifted, and world isolation ceased” (Leonard & Bekker, 2013). The country has become a popular tourist destination. The Gauteng province “is the hub of the South African economy with its mining, manufacturing, services, and supporting industries; hence, air transport systems are required to handle the air travel demand” (Leonard & Bekker, 2013).

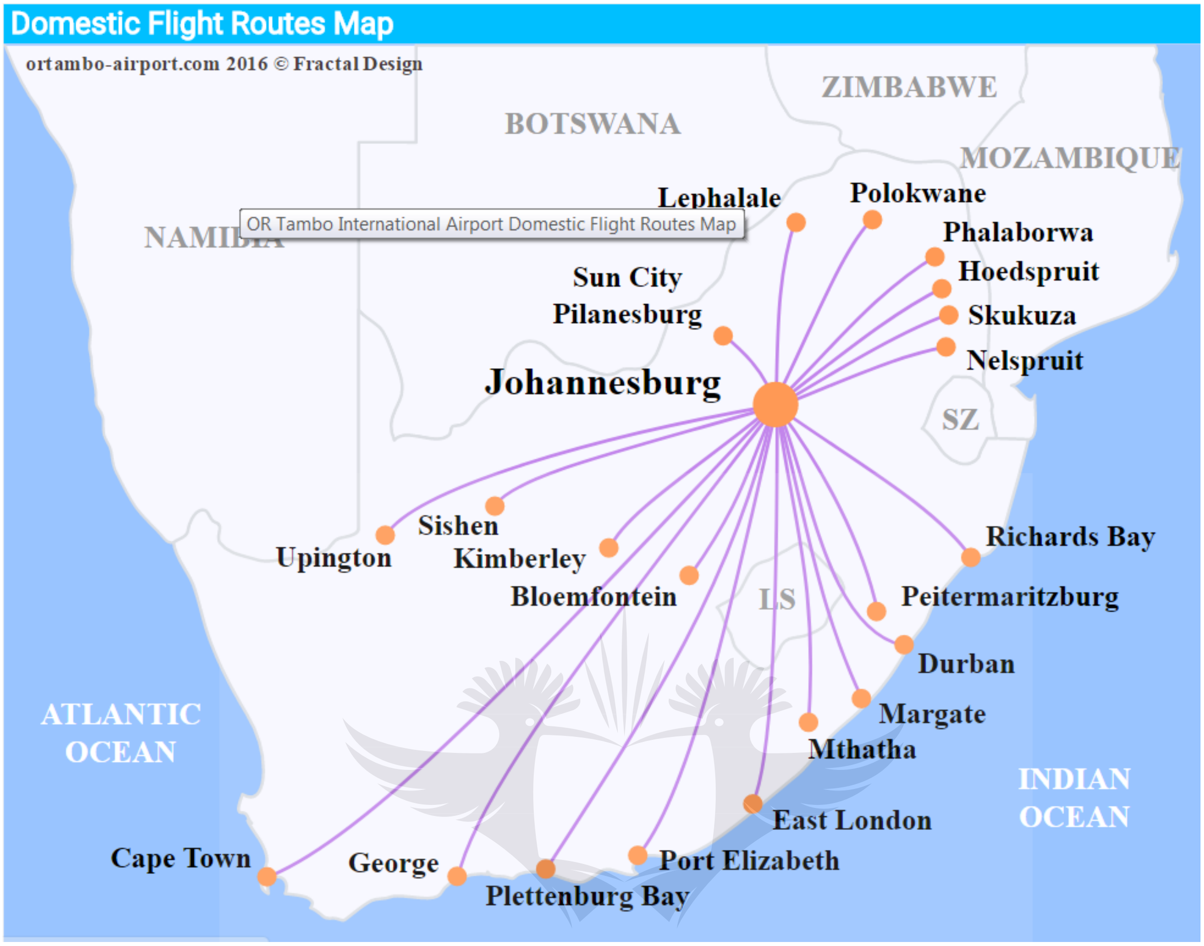


Figure 1.1: Map of the Domestic Flight Routes at ORTIA (Fractal Design, 2016)

1.3 Research problem

Airports Company South Africa (ACSA) confirmed that several South African Airways flights operating from its network of airports across the country have either been cancelled or delayed indefinitely. South African Airways “has advised its customers and stakeholders that it is experiencing operational delays and flight cancellations due to industrial action by some members of its cabin crew” (Tiso Blackstar Group, 2017). Several cabin crew “members are currently not available to enable (SAA) to operate all its flights” (Tiso Blackstar Group, 2017). Delays were being experienced “from Johannesburg to Port Elizabeth route and from Johannesburg to East London route” (Tiso Blackstar Group, 2017). Cancellations were being experienced from

Johannesburg to Durban routes and Johannesburg to Cape Town route. The route from Johannesburg to Durban “departed after 51 minutes of delay and route from Johannesburg to Cape Town departed after 20 minutes delay” (Tiso Blackstar Group, 2017). SAA has tried to monitor the delays and cancellations and is working with the stakeholders to minimise disruptions to the airport’s operations where possible but with not much success (Tiso Blackstar Group, 2017). This research will potentially help to provide a computational Intelligence tools/system that will assist in reducing flights departure delays, flights departure cancellations and improve the airport’s service through the reduction of operational delays at the ORTIA.

1.4 Aim of the study

This study aims to construct a flight departure delays prediction model using machine learning algorithms. The model will be constructed for the ORTIA to improve flight departing flow.

1.5 Objectives

This research predicts to empower travellers by proving them with a tool that can determine the punctuality of the flights departing from ORTIA and to reduce the impact of the attributes that contribute the most in causing flight departure delays using computational intelligence. We would like to have the ability foretell SAA if the flight will be delayed to depart or not. The objective of this study is to develop a model for predicting flight departure delays for OR Tambo flights using computational intelligence methods.

1.5.1 Sub -Objectives

The sub-objectives of this study as identified by the researcher, are:

- to retrieve the flight departure schedule timetables for ORTIA collected from the Flights Time Table Website http://ortambo-airport.com/flights/departures.php
- to train the dataset to construct the flights departure delays.
- to identify the variables (Delayed, On Time and Cancelled) that influence delays for flights to depart.

- to use the identified variables to construct the flight delays prediction model.
- to train and test the collected data to construct the flight delay prediction model.
- To validate the model for robustness and as proof of concept.

1.6 Research methodology

This study used a quantitative research methodology. The data were collected from the flights timetable on the website. The data were imported into Microsoft SQL Server Management Studio to meet the standard that is required to load onto WEKA (Waikato Environment for Knowledge Analysis) software.

1.7 Dissertation contribution

This study uses variables, machine learning algorithm methods to construct the best flight delays prediction model for the ORTIA. The attribute selection, machine learning algorithms that were selected are Decision Trees (J48), Multi-Layer Perceptron (MLP), Support Vector Machine (SVM) and K-Means Clustering(K-Means) to construct the live flight departure delays prediction model. This study would enable the ORTIA users to make appropriate flying decisions to avoid flight departure delays.

1.8 Structure of the dissertation

This dissertation is divided into five chapters:

Chapter one introduces the study, the research problem, the problem statement, the aim of the study, objectives of the study, research methodology, dissertation contribution and the structure of the dissertation.

Chapter two presents the literature review and the theoretical framework of the study.

Chapter three outlines and discusses the research methodology of the study, data collection processes and algorithms used.

Chapter four presents the experiments and results of the study.

Finally, chapter five presents the discussion of the results, a conclusion and includes recommendations.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Several researchers have proposed various methods and algorithms to solve the problem of flight delays. The literature review focuses on related work on predicting flight departure delays.

2.2 Related research

Choi, Kim, Briceno and Mavris (2016) used supervised machine learning algorithms to predict weather-induced airline delays. “Algorithms such as random forest, AdaBoost, k-Nearest-Neighbors and Decision Trees were used” (Choi et al., 2016). The model’s true positive rate (TPR) predicted an accuracy of 81.37% for Random Forest, 78.05% for AdaBoost, 61.69% for kNN and 77.02% for Decision Trees using sampling techniques. The “Synthetic Minority Over-Sampling Technique (SMOTE) and random under-sampling were used for balancing their data set” (Choi et al., 2016). The weakness of this model is that the costs of false positives and false negatives were not considered when building this model, which could have increased the model’s prediction accuracy. Additionally, the uncertainty in weather forecasts was not considered, which could have enhanced the model’s predictive performance.

The artificial neural network (ANN) was used by Alonso and Loureiro (2015) to implement a unimodal model for predicting flight departure delays at Porto Airport. They established “that the arrival delay and the ground operation time are significant variables for departure delay prediction” (Alonso & Loureiro, 2015). Arrival delay variables had an importance of 50.35% and ground operation time of 49.65%. Their implementation “is weak as it cannot distinguish flights whose departure delay falls in $[-\infty, 0]$ minutes from flights whose departure delay falls in $[0, 15]$ minutes” (Alonso & Loureiro, 2015).

A Time-Based Flow Management (TBFM) “departure scheduling tool checkbox default ON vs. default OFFs” by Parke, Mohlenbrink, Brasil, Speridakos, Yoo, Omar, Buckley, Gabriel and Belfield (2016) was tested with Human-in-the-Loop (HITL) to simulate conditions that included Baseline, Expected Departure Clearance Times (EDCT) and Estimated Departure Clearance Time

(EDCT) plus RTA condition to reduce departure delays at airports that are near other airports or higher volume airports. The tool tested "was a TBFM departure scheduling tool—checkbox default ON versus default OFF" (Parke et al., 2016). When ON, "this tool could compress airborne aircraft that made space for flight departures" (Parke et al., 2016). Departure delays "were reduced with the checkbox in the ON position" (Parke et al, 2016). The TBFM departure scheduling tool "predicted that the departure delay was reduced with the checkbox in the ON position and arrival delay was increased" (Parke et al., 2016). This study did not mention their tool's prediction accuracy. Parke et al. (2016) also did not evaluate the prediction tool.

A multi-agent system was developed by Breil, Delahaye, Lapasset and Féron (2016). This multi-agent system used a speed regulation algorithm, departure delay selection algorithm, conflict detection algorithm and hypothesis algorithm. The multi-agent systems "was for air traffic conflicts resolution by local speed regulation and departure delay" (Breil, 2016). "The multi-agent systems also reduced the number of conflicts". It also "decreases overall traffic complexity, which becomes easier to manage by air traffic controllers" (Breil et al., 2016). "These algorithms could solve up to 100% of the conflicts in the tested scenario. They are also resilient to perturbations like non-cooperative agents". They could also "eventually be implemented on board collaborative decision processes, removing the need to rely on ground equipment. The speed regulation algorithm alone, however, cannot solve all conflicts (Breil et al., 2016)". To successfully solve all conflicts, aircraft had to execute other types of manoeuvres in addition to speed regulation, like delay of the departure time, flight level changes or heading changes. The study by Breil et al. (2016) did not measure the difference in computation time and results optimality returned by the multi-agent system and by a global optimisation method.

Liu and Ma (2008) constructed a Bayesian network (BN) model. "Through the learning of BN, it was found that 93.9% of arrival-delays and departure-delays occur between 8:00 and 21:00 in a day" (Liu & Ma, 2008). The departure of these flights "gets delayed because of the delay-propagation from arrival delay" (Liu & Ma, 2008). "Then departure delay also was HEAVY DELAY with a probability of 62.9%. Thus, the propagation in states of MEDIUM and HEAVY DELAY could not be avoided" (Liu & Ma, 2008). The data they obtained was from March and

April. The study could have improved if data for a whole year were used instead of using data for only March and April.

Novianingsih and Hadianti (2014) constructed a probability distribution for flight departure delay durations and departure delay-time using Modelling Flight Departure Delay Distributions. “The distribution of departure delay duration and departure delay-time were modelled by solving optimization models for estimating parameters of appropriated density” (Novianingsih & Hadianti, 2014). They used “the two-stage Genetic Algorithm (GA) to find the optimal (near-optimal) parameters which gave better optimum solutions than one-stage GA” (Novianingsih & Hadianti, 2014). The density functions used were Log-normal, Gamma, and Rayleigh. They used the LSE to evaluate each stage of the two-stage GA. Novianingsih and Hadianti “realized an improvement of the LSE for each density in the second stage”. Using the least square error (LSE) in the second stage, “the best model for the distribution of departure delay duration was Log-normal with 0.0064 least square error duration” (Novianingsih & Hadianti, 2014). The mixture of four normal distributions was found to be good in modelling departure delay-time distribution with 0.0075. The disadvantage is that the departure delay distribution should be modelled “as a function of both delay duration and delay-time” (Novianingsih & Hadianti, 2014).

The stochastic gate reassignment model (SGRM) “is for gate reassignments where there were temporary gate shortages and stochastic flight delays” (Tang, 2011). “The findings on numerical tests based on Taiwan Taoyuan Airport (TTY) operations and characteristics of the results could serve as useful reference materials for planners and operators” (Tang, 2011). The SGRM was found to have more advantages than the traditional gate reassignment model (TGRM) “when considering stochastic flight delays and violations. The test results showed that the model could be applicable to TTY Airport operations, even in the case of larger system scales” (Tang, 2011).

Results showed that the reassignment of the traditional gate reassignment model (TGRM) is too optimistic and would produce large disturbances. “The variation percentage of the SGRM (0%) was significantly better than that of the TGRM (36.84%)” (Tang, 2011). However, “the only simultaneous incidents of temporary gate shortages and stochastic flight delays were addressed, and this is a problem” (Tang, 2011). Airport closures and unexpected airport congestion “were not considered in the (SGRM)" (Tang, 2011).

Wei and Hongshan (2010) used a nonparametric directional output distance function "to analyse the efficiency of 10 airports around Yangtze River Delta in 2007". The results indicated "that when the delays were considered, airport efficiency increased especially in small and less congested airports". All 10 airports increased all outputs by 85.2% "to achieve efficient utilization of facilities" (Wei & Hongshan, 2010). Their results are weak because they were not evaluated.

Qianya, Lei, Rong, Bin and Xinhong (2015) used "An Analysis Method for Flight Delays based on Bayesian Network, which used standard variance with posterior (Sposterior) and standard variance with prior (Sprior) values for flight delay". They analysed and predicted the delay during the flight. Qianya et al. (2015) "selected 5 000 set flight data as the historical data for Bayesian network parameter learning and selected 300 data set to be predicted". They could "predict the delay conditions of the downstream flights timely when the delay happened upstream" (Qianya et al., 2015). By the results comparison, the Sposterior was 81.95% and the Sprior was 55.30%. The Sposterior was 26.65% more than the Sprior; therefore, it was seen that the Bayesian Network was an effective analysis method for flight delays. The study by Qianya et al. (2015) did not evaluate root mean squared error (RMSE).

Yao, Jiandong and Tao (2010) analysed "flight delay propagation caused by cross-flight plan awaiting resources and proposed a model and the corresponding prediction algorithm of flight delay propagation". The simulation results "showed that the model can be used to effectively compute flight delay propagation caused by cross-flight plan awaiting resources" (Yao et al., 2010). By calculating (EPD) time of estimated propagation delay subtracting "(TSD) time of scheduled departure" from (TED) time of estimated departure in minutes, it "provided [the] airport a clear flight delay warning rank of 10% green, 20% blue, 40% yellow, 60% orange and 100% red through the relationship between (DAFD) degree of airport flight delays, RADP prediction rank of airport delays". Yao et al. (2010) did not evaluate their results, which is problematic.

Kim, Choi, Briceno and Mavris (2016) showed that the deep architecture that includes Shallow, Stacked Recurrent Neural (RNN), "deep input-to-hidden function and deep hidden-to-output function” improved the accuracy of the airport delay prediction models by 3-5% compared to the shallow one. The different “deep architectures of RNN achieved about 90% accuracy for the day-to-day delay status prediction by applying deep Long Short-Term Memory (LSTM)” (Kim et al., 2016). A reliable delay “status of a single day could be acquired by feeding the delay status of a day to the individual flight delay model” (Kim et al., 2016). It gave state-of-the-art results “in predicting individual flight delays” (Kim et al., 2016). The deep model “achieved an accuracy of between 86% to 87%” (Kim et al., 2016). The study could have applied “other deep architectures to the prediction and analysis task of flight delays”. It could have yielded “important patterns in flight delay data” and improved the model’s prediction accuracy.

Xing and Tang (2016) presented “a model based on the genetic algorithm (GA) for optimizing airport flight delays in order to reduce serious air traffic flight delays”. They optimised the distribution of the flight delay by putting forward concentrated distribution and dispersed distribution that were used to achieve the flight punctuality rate of 70%. They, however, did not compute the flight delays prediction.

Liu and Ma (2009) used “a new Bayesian Network structure learning algorithm, named Target-fixed Stochastic-ordered K2 (TSK2). The TSK2 was based on setting Bayesian Network parameters”. After Liu and Ma (2009) set Bayesian Network parameters and training the Bayesian Network structure, based “on TSK2 by 66% of the data set, the flight delay had the correctly classified instances of more than 98%”. From Liu and Ma study’s results from the experiments, “the TSK2 has been proved to be an efficient and fast algorithm with 98.2883%”. The weakness of the study is that only 66% of the data set was used for the experiments.

Ding and Tong (2008) combined “the characteristics of airport flight operation and statistical data of flight delays to form early warning grading standards of flight delays”. The negative selection algorithm “was applied to diagnose and to forecast the state of the airport scheduled flight delay” (Ding & Tong, 2008). The researchers “used the corresponding flights-early-warning data according to the different days and different time period in a week and risen the weight factor of the weekly aviation running-data based on the basic aviation data”. The early warning accuracy rate “was more than 80% in 24 days, while in only 6 days it was lower than 80%, which indicates high accuracy" (Ding & Tong, 2008). This means "that this method can forecast the accumulative delay value of a specified 15 time-periods" in a month accurately with above 80%, 92% accuracy. The study could have improved if the data of more than one month and more time period was used.

An optimisation model by Lv and Wang (2009) based on Markov's theory and to "calculate the state transition probability matrix and make forecast was presented". If the flight normal rate was above 80%, it meant a good effect and the forecast error being controlled within 5%. When the normal rate was common and extremely low, below 50%, it "could be used in the near time stage and the error is within 10%". Because "of the increase of data fluctuation, if we still use the initial two hours' data to forecast the whole day's delay situation, the biggest value of m was 39 for probability matrix (Pm) in the study, the effect was not very good" (Lv & Wang, 2009). The characteristic of Markov's process "is to get the state transition probability matrix using a developing trend of system states in the past time stage, multiply it with the current state, and work out the system's intending developing trend". This characteristic "has decided that it suits the system which has continuous condition change". The disadvantage "is if the difference of the external conditions between the preceding time section and the latter time section is very big, then the system's condition is unstable, so it is not suitable to use this process to solve this kind of situation" (Lv & Wang, 2009).

Schaefer and Millner (2001) presented a "Policy Assessment Tool (DPAT) that models the propagation of delay throughout a system of airports and sectors". Locally, "delay increases with increasing duration of Instrument Meteorological Conditions (IMC)". IMC "obtained results for local flight departure and arrival delays due to IMC, propagation for IMC, comparisons to VMC results, and a comparison of propagated delays to entire system" (Schaefer & Millner, 2001). "95% confidence intervals were for average scheduled arrival delay for the first through fifth legs of an aircraft itinerary for all flights leaving an IMC airport during the 8-hour period after IMC begins" (Schaefer & Millner, 2001). The disadvantage is that they focused on delays incurred due to inclement weather at one airport and the results for the departure delays were not mentioned.

At "Newark Liberty International Airport (EWR)" Manley and Sherry (2008) examined the "trade-off between flight delays and passenger delays as well as airline equity and passenger equity in

Ground Delay Program (GDP) slot allocation". A "GDP Rationing Rule Simulator (GDP-RRS) was developed to calculate efficiency and equity metrics for all stakeholders". A "comparison of alternate GDP rationing rules established that passenger delays can be significantly decreased with a slight increase in total flight delays". Compared "to the traditional Ration-by-Schedule, Ration-by-Aircraft size (RBAc) decreased the total passenger delay by 10% with 0.4% increase in total flight delay, and Ration-by-Passengers (RBPax) decreased total passenger delay by 22% with only 1.1% increase in total flight delay". Ration-by-Schedule (RBS) "was preferred only if airlines were the main focus of the system". RBS resulted "in the minimum total inequity for both airlines and passengers and this was achieved at the expense of a large efficiency loss due to high passenger delays" (Manley & Sherry, 2008). The weakness of the results is that "the impact of GDP rationing rules over the long-run was not investigated, more alternative rationing schemes like Ration-by-Environmental impact were not included". The impact "of GDP scope and the impact of airline cancellation policy were also not included". Comparing "airport metrics to airline and passenger metrics" can further improve their results.

Evans and Clarke (2002) examined the operations at Newark International Airport (EWR) and "responses to delays, to determine the responses that were most effective in reducing delays". Nearly "85% of delays at EWR are caused by adverse weather". The responses "namely the application of restrictions, re-routing with the help of the National Playbook, and the use of decision-aiding tools such as the Dynamic Spacing Program (DSP) and the Integrated Terminal Weather System (ITWS) to delays have reduced the growth in delays at EWR, and may thus be beneficial to reduce the growth of delays at other airports" (Evans & Clarke, 2002). It, however, is "the only response expected to reduce delays significantly airspace redesign". Improved interfacility communications have been identified as one tactical response to delays. The disadvantage of the study is that "the responses at Newark will not address local problems at other airports".

Choi et al. (2016) "proposed the cost-sensitive classifier to identify individual flight delays". The "misclassification costs of on-time class and delays class were analysed for the model". A subsampling method called costing "was used to reflect the cost analysis results". The Random

Forest, AdaBoost, K-Neighbors and Decision Tree were transformed into the cost-sensitive classifiers. These machine learning algorithms "were applied to the subsampled set extracted by the costing method". The "weighted error rate of the model was measured for the various cost ratios between false positive error and false negative error". First, "the 'costing' method was utilized to predict flight delays". Thus, Choi et al. (2016) "were able to achieve good predictive performance from these cost-sensitive classifiers". In addition, the asymmetric misclassification "costs were reflected in the performance evaluation". Recurrent neural networks or convolution neural networks combined with the sampling methods "could have been applied to improve the performance of the model" (Choi et al., 2016).

2.3 Theoretical framework of the study

"Collaborative Decision Making (CDM) allows airport operators, aircraft operators/ground handlers, ATC, Network Operations dealing with Air Traffic Flow Management and other aviation stakeholders to participate in the Air Traffic Management (ATM) decisions that affect them" (ATNS SOC, 2018). Airport CDM "aims to improve the overall efficiency of operations at an airport, with a focus on the aircraft turn-around and pre-departure sequencing process" (ATNS SOC, 2018). One of the main outputs of the CDM process "is more accurate Calculated Take Off Times which can be used to improve en route and sector planning of the ATM Network". "The Airport CDM concept aims at improving operational efficiency at airports by reducing delays, improving the predictability of events during the progress of a flight and optimising the utilisation of resources. With Airport CDM the network is also served with more accurate take-off information to derive Air Traffic Flow and Capacity Management (ATFM) slots" (ATNS SOC, 2018).

"The improved decision making by CDM Partners is made possible by sharing of accurate and timely information and adapted operational procedures, automatic processes and user-friendly tools" (ATNS SOC, 2018). Thus, Airports Company South Africa has Airport Management centres (AMCs) at ORTIA, Cape Town and King Shaka international airports. "The AMCs allow for all airport stakeholders that include airline operators or ground handlers, Air Traffic Control (ATC) representatives, customs or immigration, and airport operators to come together and make decisions that allow for efficient and flexible operations at the airport" (ATNS SOC, 2018).

2.4 Chapter conclusion

A summary of findings from the literature review on flight delays prediction models showed that most of the related work was conducted using machine learning algorithms and there are a few studies that used Bayesian Network (BN) and Multi-Layered Perceptron (MLP) methods. The BN and MLP can be implemented to train classifiers to improve their performance. The previous studies showed that different machine learning algorithms, BN and MLP methods can be used in the prediction of flights delays.

CHAPTER 3: METHODS

3.1 Introduction

This chapter introduces the research methodology, data collection and machine learning tools that will be used for constructing the flight delays prediction model.

3.2 Research design

Figure 3.1 shows an overview of the entire process used for the construction of the flight departure delays prediction model.

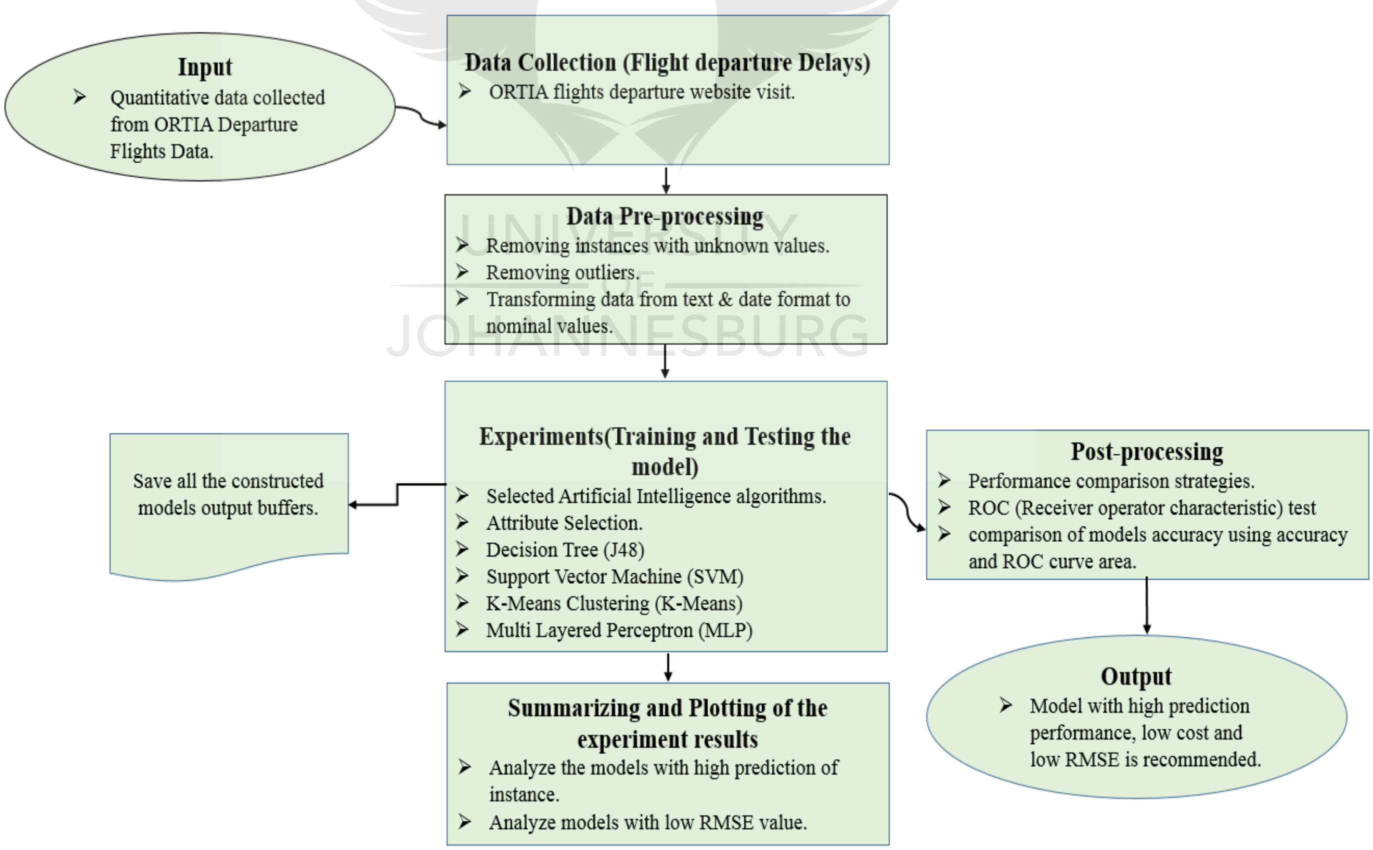


Figure 3.1: The process of constructing the flight departure delays prediction model

Figure 3.1 shows the steps in which data were collected, pre-processed, and the experiments carried out. The post-processing of results is discussed in Chapter 4. The outliers were removed otherwise a non-representative model would be produced. Using Microsoft SQL Server Management Studio, the data were converted to nominal data, “Which has values that are distinct symbols” (Witten, Frank & Hall, 2011).

3.3 Research methodology

3.3.1 Research approach

This is a quantitative research study where a variable “is defined as a formal, objective, systematic process to describe and test relationships and examines cause and effect interactions among” (Burns & Grove, 1993). Data for flights departure for 2017 were collected from the website http://ortambo-airport.com/flights/departures.php. The data were pre-processed using a Microsoft SQL Server Management Studio from text and date format to nominal values. Machine learning algorithms that include Decision Trees (J48), Support Vector Machine (SVM), Multi-layered Perceptron (MLP) and K-Means Clustering (K-Means)) were used to carry out the experiments. The results were evaluated using the confusion matrices and ROC Curves covered in section 4.3 and 4.4.

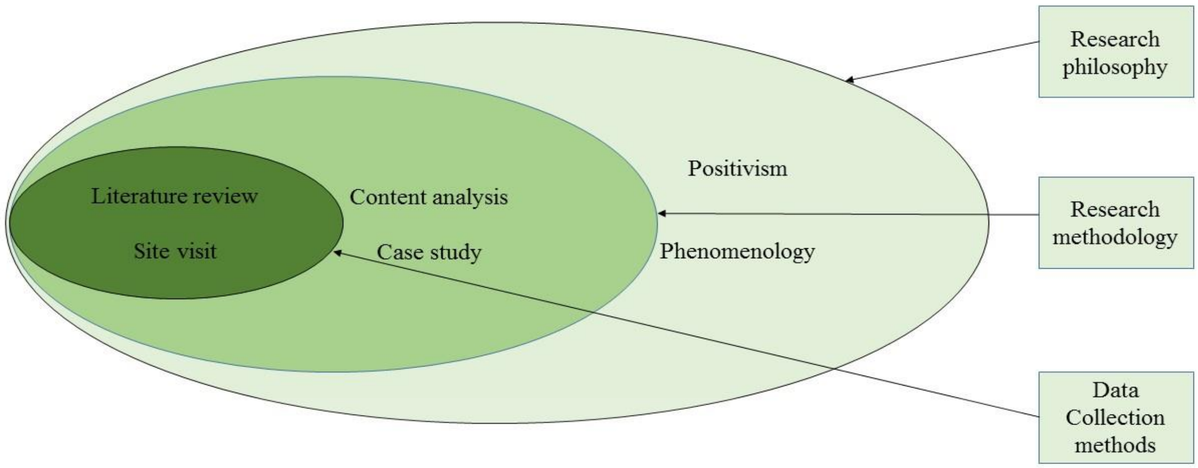


Figure 3.2: The research onion (Source: Adapted from Saunders et al., 2012)

The structure of the approach in this dissertation is based on the research onion process as shown in Figure 3.2 (Saunders, Lewis & Thornhill, 2012).

3.3.2 Research philosophy

The epistemologies comprising objectivism, subjectivism and constructivism (Kayrooz & Trevitt, 2005) guided how to think about this study and how to structure this work. “An epistemology is a fundamental understanding of the nature of knowledge. Objectivism asserts that research can lead to knowing and to verifying an objective truth” (Kayrooz & Trevitt, 2005). Subjectivism asserts that there are infinite interpretations of events, with none of them being superior to another. “Constructivism says that the objective world is meditated by an individual’s conceptual lens or framework. Quantitative approaches align with objectivism whereas qualitative approaches align with subjectivism. Both qualitative and quantitative approaches can be found in constructivism” (Kayrooz & Trevitt, 2005). Hallebone and Priest (2009) state that “all research paradigms take a view of the assumed nature of reality being studied as realist or idealist or relativist”. “It is the combination of a direct or indirect way of knowing absolutist or constructionist with the type of ontology (a view of reality) that defines and carves the boundaries of the research paradigm. The research is based on the quantitative in approach is aligned to a positivist approach" (Hallebone & Priest, 2009).

Figure 3.3 depicts the research underpinnings. It "provides a visual cross-section from epistemology to the methodology, boxing the concepts" (Yeganeh, Su & Chrysostome, 2004) or items objectivism, positivism and the quantitative approach that apply to this research.

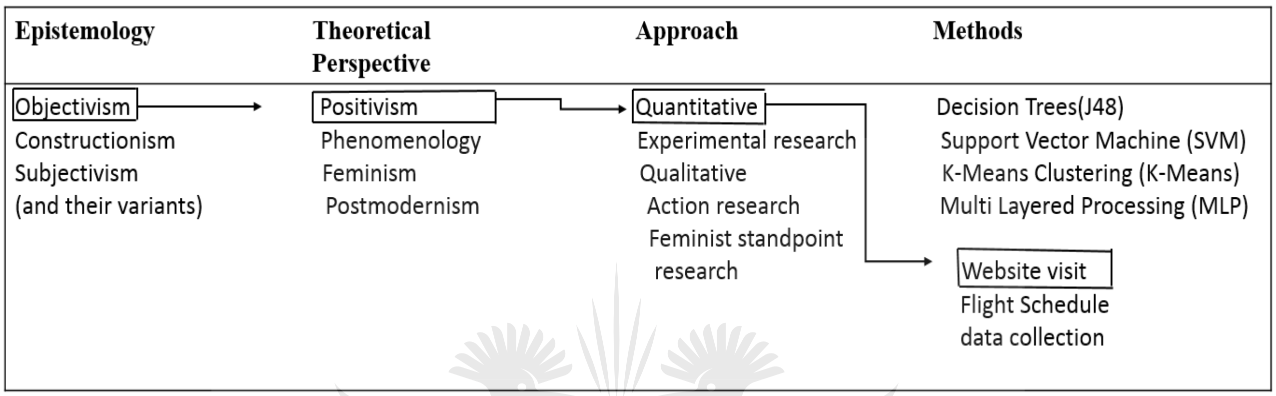


Figure 3.3: Research underpinnings this research (Source: Kayrooz & Trevitt, 2005)

3.3.3 Research settings

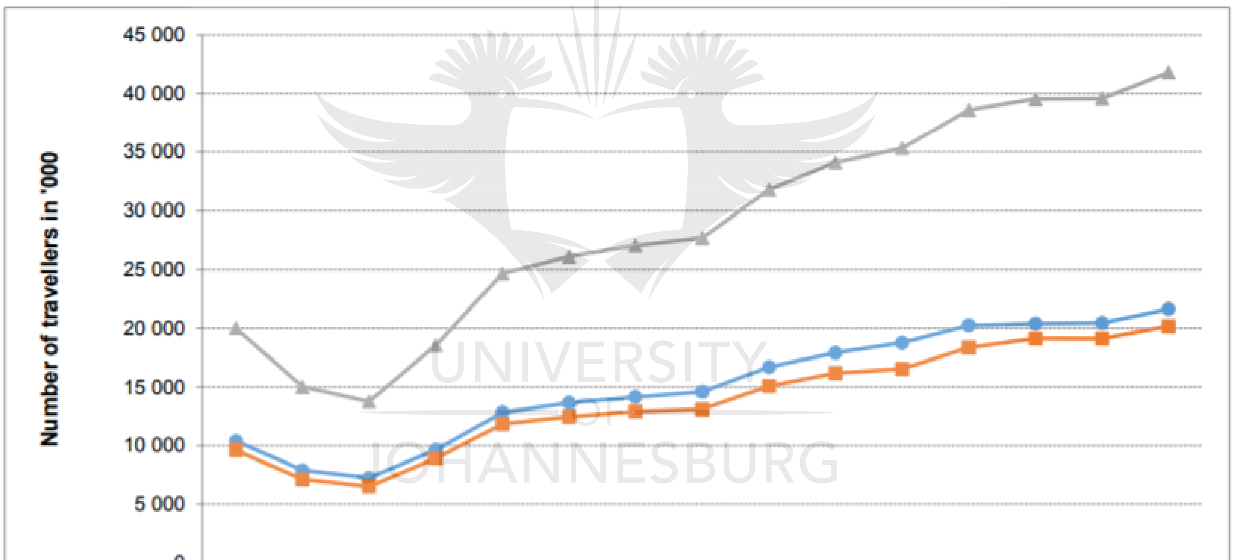
This research study was conducted with flight departure data collected from (ORTIA) for (SAA) airline. SAA has approximately 82 aircrafts per day departing (ORTIA) Johannesburg (JNB) (Peck, 2009).

3.3.4 Research methods

Figure 3.3 shows "the research underpinnings applicable to this research project" (Yeganeh et al., 2004). It provides "a visual cross-section from epistemology to the methodology, boxing the concepts that are applicable to this research" (Yeganeh et al., 2004). The boxed items objectivism, positivism and the quantitative approach were all discussed under the headings of research philosophy and research methodology.

3.3.4.1 All travellers

Figure 3.4 “presents detailed information derived from the data published in 2016 Tourism and Migration monthly statistical releases (P0351). In addition, data from 2015 and earlier are presented to provide a comparison in observed trends and patterns. Travellers comprise passengers in transit, arrivals into and departures from South Africa regardless of residency status” (Lehohla, 2016). Data presented in Figure 3.4 “shows that approximately 41 818 travellers (arrivals plus departures) were recorded in 2016 compared to 20 016 recorded in 2002, indicating that the overall number of travellers doubled over this 15-year period”. “There were 21 636 thousand arrivals and 20 182 departures in 2016 compared to 10 400 arrivals and 9 616 departures in 2002. Arrivals increased by 5.8% while departures increased by 5.6% during the same period” (Lehohla, 2016).



	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014*	2015*	2016*
Arrivals	10 400	7 894	7 250	9 631	12 826	13 657	14 147	14 586	16 700	17 925	18 767	20 245	20 404	20 454	21 636
Departures	9 616	7 119	6 512	8 920	11 841	12 456	12 904	13 104	15 073	16 180	16 525	18 366	19 126	19 119	20 182
Total	20 016	15 013	13 762	18 551	24 667	26 113	27 051	27 690	31 773	34 105	35 292	38 611	39 530	39 573	41 818

*2014-2016 data excludes travellers in transit

Figure 3.4: Number of arrivals and departures of travellers by year of travel, 2002–2016 (Source: Lehohla, 2016)

Table 3.1: The distribution of instances for constructing the flight delay prediction model

Target Concept	**Training dataset**
Delayed	822
Cancelled	37
On Time	1375
Total Training	2234
Total Attributes	(5) 4 attributes plus the Target Concept

3.3.4.2 Sampling

A sample is composed of elements that are representative a population (Strydom & Venter, 2002). The training dataset sample size of 2 234 data points was used.

3.3.4.3 Dataset collection

Quantitative departure flight primary data shown in Annexure 3A was collected from daily visits to a website for (ORTIA), JHB (IATA code JNB) flight departures. The data were in a table format, and it was for flights that originate from ORTIA for domestic and international destinations. This data was for all flights departing from ORTIA up until midnight (CAT). The data was provided in 3-hour time slots, so the researcher could use the tables filter to select the desired time slot (Fractical Design, 2016).

3.3.4.4 Month of travel

The distribution of passengers by Lehohla (2016) "arriving and departing at ORTIA classified by month of travel is provided" is shown in Figure 3.5. The results "show that the highest proportion of arrivals was recorded in January (10,7%) followed by December (10,3%), while the lowest proportion was recorded in February (6,0%). Of the departures, the highest percentage was recorded in December (13,8%) and the lowest in February (6,1%)" (Lehohla, 2016).

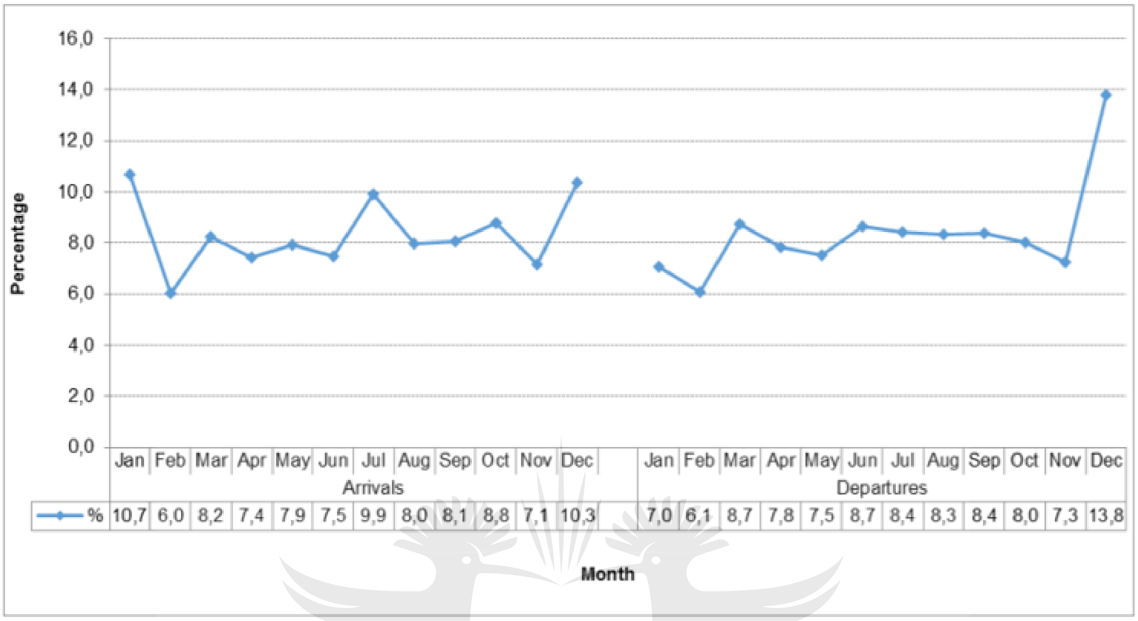
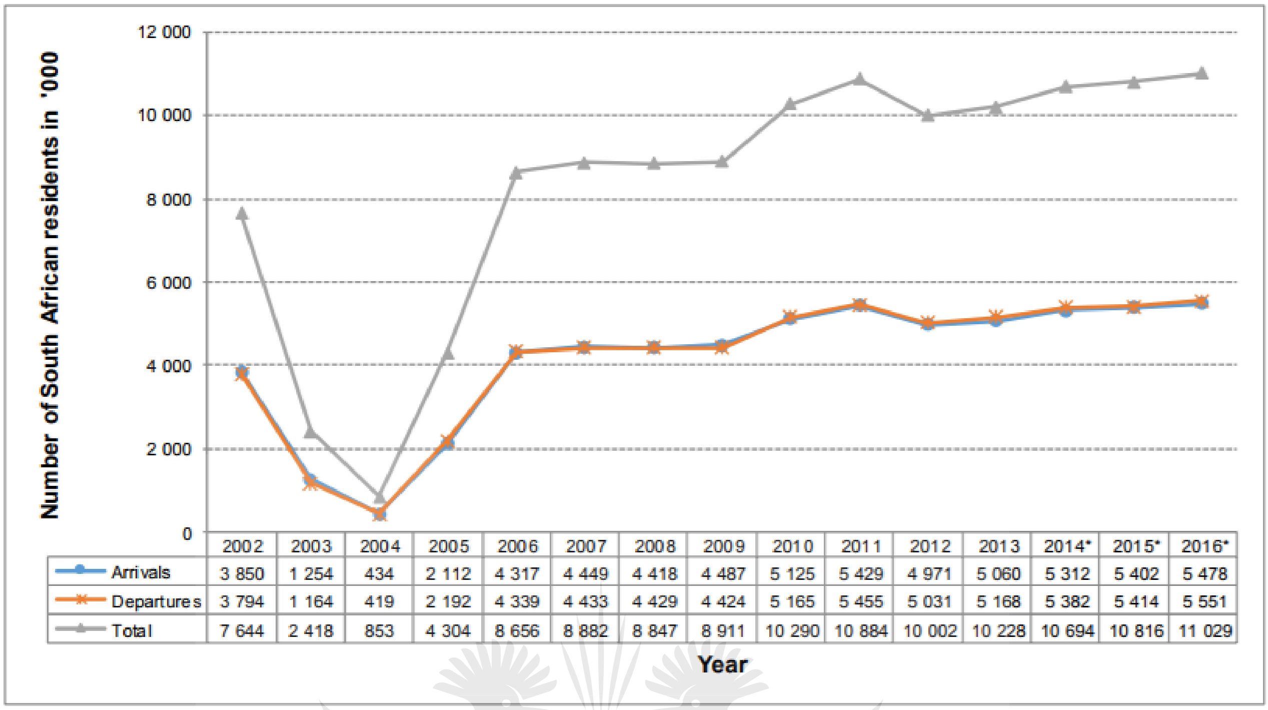


Figure 3.5: Percentage distribution of arrivals and departures of passengers by month of travel, 2016 (Source: Lehohla, 2016)



	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014*	2015*	2016*
Arrivals	3 850	1 254	434	2 112	4 317	4 449	4 418	4 487	5 125	5 429	4 971	5 060	5 312	5 402	5 478
Departures	3 794	1 164	419	2 192	4 339	4 433	4 429	4 424	5 165	5 455	5 031	5 168	5 382	5 414	5 551
Total	7 644	2 418	853	4 304	8 656	8 882	8 847	8 911	10 290	10 884	10 002	10 228	10 694	10 816	11 029

Figure 3.6: Number of arrivals and departures of passengers at ORTIA by year of travel, 2002–2016 (Source: Lehohla, 2016)

The data in Figure 3.6 "indicate that arrivals and departures of South African residents declined drastically in 2003 and 2004 then increased from 2005 to 2016" (Lehohla, 2016). These declines in the volume "of these travellers were due to the changes in the Immigration Act No.13 of 2002 regarding the recording of information on international movements of South African resident resulting in information on arrival and departure of residents not being consistently collected" (Lehohla, 2016). The misinterpretation "of the Act led to some immigration officers collecting while others were not collecting this data). Collection resumed after the implementation of the Immigration Amendment Act, 2004 (Act No.19 of 2004)". Figure 3.6 "shows that there were nearly as many departures as arrivals over the years" (Lehohla, 2016).

3.3.4.5 The data collection procedure

Data were collected daily from the ORTIA JNB Flight Departures website in 2017. The data included parameters such as the flight number, carrier, destination airport, departure time and flight status. The data were recorded in text, date and time format.

3.3.4.6 Validations

The prediction model was constructed using machine learning algorithms that include the Decision Trees (J48), Support Vector Machines (SVM), Multi-layered Perceptron (MLP) and the K-Means Clustering (K-Means) algorithm and data. The results were evaluated using a combination of confusion matrices of all the constructed models and the ROC curves covered in section 4.3 and 4.4.

3.3.4.7 Limitations of the study

The data dating back 5 years were not available or accessible, and data on weather were not available. The airlines were not comfortable with giving away most of their data such as the airlines financial costs data.

3.4 Attribute selection

Each feature used in the construction of a model "can increase the cost and running time of a recognition", classification or prediction system, designing and implementing systems with a highly discriminative small feature set ensures the achievement of high prediction rates (Tiwari & Singh, 2010). There "is an opportunity for improving the usefulness of machine learning techniques for automatically generating useful prediction procedures" (Saeys, Larra-naga & Inza, 2007).

Feature selection, "also known as feature reduction or attribute selection, has to do with selecting a highly discriminative feature for constructing a robust learning model" (Hall, 1999). Removing "the most irrelevant and redundant features from the data does improve the performance of a learning model" (Lei & Liu, 2003) by:

1. reducing or mitigating the effect of the curse of dimensionality thus speeding up the learning process due to a number of features;
2. enhancing generalisation capability of the model to be constructed from these features; and
3. improvising model interpretability.

The feature selection algorithms used in Chapter 4 comprise “Feature ranking and subset selection” (Kantardzic, 2011). The former ranks the features by a metric and eliminates all features that do not achieve a certain score. “The latter searches the set of possible features to get an optimal subset”. Feature selection provides a better understanding of the data by identifying a relation between features. Stepwise regression is a greedy feature selection “algorithm that adds the best feature (or deletes the worst feature)” in each cycle. There is a need to determine the stopping criteria when training the model, and this can be done by cross-validation (Michie, Spiegelhalter & Taylor, 1994).

The architecture of a feature selection system is shown in Figure 3.7. A full set of features “is provided as an input representing positive and negative examples of the various classes for which the prediction is to be performed” (Vafale & De Jongh, 1992). A search procedure or algorithm “is used to explore the space of all features of the given feature set”. The performance “of each of the selected feature subset is measured by use of an evaluation function” shown in Figure 3.8 that measures the specified classification result. The best feature subset found “becomes the set of features to be used in the actual design of a model” (Vafale & De Jongh, 1992).

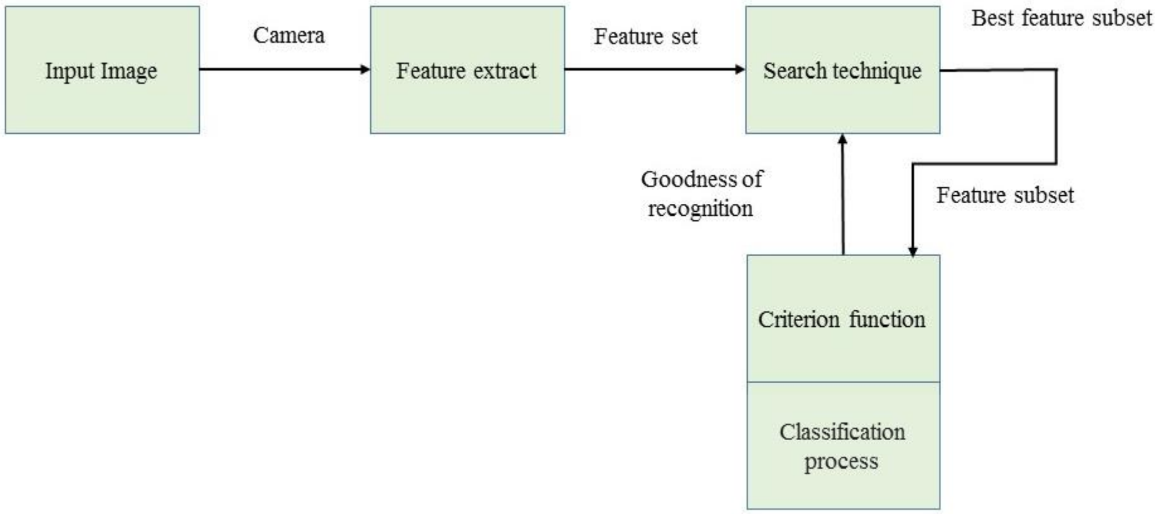


Figure 3.7: Feature Selection process diagram (Miao & Hou, 2004)

The valuation procedure in Figure 3.7 is done by excluding the features that are not effective (Miao & Hou, 2004). The next phase is to create a prediction model using the new training data set comprising highly discriminative features to make rules that capture the underlying function that describes the solution to the problem domain (Dash, Choi, Scheuermann & Liu, 2002). A class description "is formed by a set of decision rules describing all the training examples given for that particular class. Once a model has been constructed, it must be evaluated (post-processing) by checking the prediction performance of the induced rules on the unseen data" (Dash et al., 2002) which is discussed in Chapter 4.

The advantage of using attributes selection is that it improves the performance of rule induction techniques. This is a step towards automating the construction of prediction systems for difficult problems (Hall, 2000) that incorporate hundreds of attributes. The search techniques described in section 3.6 sit in the adaptive features selection process as shown in Figure 3.7.

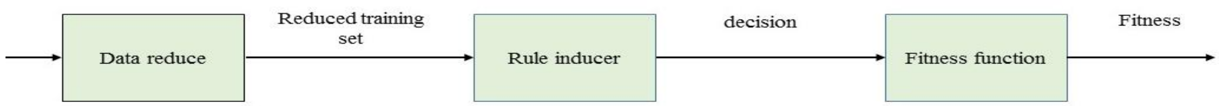


Figure 3.8: Feature subset evaluation procedure (Vafale & De Jongh, 1992)

3.5 Machine learning algorithms used for the study

3.5.1 Decision Trees (J48)

The Decision Trees learning "is a method for approximating discrete valued target functions, in which the learned function is represented by a Decision Tree" (Michell, 1997–1999). Learned trees "can also be represented as sets of if-then rules to improve human readability. Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance" (Wang & Hu, 2002). Decision tree learning is a practical method for inductive inference (Wang & Hu, 2002). Each node in the tree "specifies a test of some attribute of the instance and each branch descending from that node corresponds to one of the possible values for this attribute". An instance is classified "by starting at the root node of the trees, testing the attributes specified by this node and then moving down the tree branch corresponding to the value of the attribute in the tree" (Wang & Hu, 2002) as shown in Figure 3.9. This process is repeated for the subtree rooted at the new node.

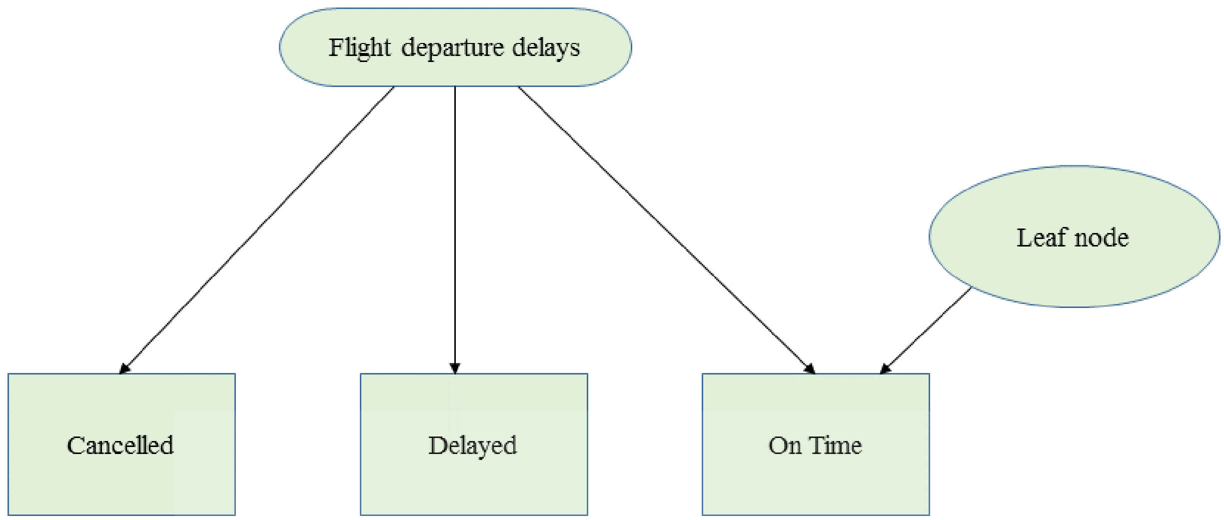


Figure 3.9: Illustration of a Decision Tree (J48) (Wang & Hu, 2002)

The Decision Tree learning algorithm suited for:

- instances that are represented by attribute-value pairs. For example, attribute ‘Week of Month’ and its value ‘Month Beginning’, ‘Month Middle’, ‘Month End’.
- “The target concepts have discrete output values. It can easily deal with instances that are assigned to a Boolean decision, such as ‘true’ and ‘false’, ‘p (positive)’ and ‘n (negative)’,” but in this study, the target concepts are ‘Cancelled’, ‘Delayed’ and ‘On Time’.

The Decision Tree algorithm has the following advantages (Callan, 2003):

- It is easy to “understand Decision Tree models after a brief explanation”.
- “They have value even with little hard data. Important insights can be generated based on experts describing a situation (its alternatives, probabilities and costs) and their preferences for outcomes” (Callan, 2003).
- They allow the addition of new possible scenarios.
- They “help determine worst, best and expected values for different scenarios”.
- “They can be combined with other decision techniques” (Callan, 2003).

Disadvantages of Decision Trees are as follows:

- "For categorical variables with a different number of levels, information gained in decision trees are biased in favour of those attributes with more levels".
- Calculations can get complex, particularly if many values are uncertain and or if many outcomes are linked (Khoonsari & Motie, 2012).

3.5.1.1 Interactive dichotomize 3 (ID3) algorithm

The ID3 algorithm is used for learning decision trees, and it is easy to use and effective (Quinlan, 1986). "It is based on the Concept Learning System (CLS) algorithm". It builds a decision tree "from some fixed or historic symbolic data to learn to classify and predict the classification of new data. The data must have several attributes with different values. Meanwhile, this data also has to belong to diverse predefined, discrete classes (i.e. Cancelled/Delayed)" (Quinlan, 1986). The Decision tree chooses the most effective attributes for decision-making by using the information gain (IG) measure (Quinlan, 1986).

3.5.1.2 Entropy and information gain

Figure 3.9 shows a typical decision tree. In this example, passengers decided to take flights in a state where the flight departure time is very early in the morning from 06:00 am, according to the state of business travelling purposes from the Airports Company South Africa. The basic ID3 method by (Quinlan, 1986) selects each instance attribute classification by using the information gain measure defined by equation 3.1, and it begins at the top of the tree. The ID3 by Quinlan (1986) selects the attribute of each node of the tree using the information gain measure, which measures the worth of the attribute. The quantity Entropy, defined by equation 3.2, is a measure of how attributes are mixed, to choose the best attribute from the candidate attributes.

$$Gain\ (S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{Sy}{S}\, Entropy\ (Sy) \qquad \text{Equation 3.1}$$

Entropy (S) = "where S is a collection of examples, A is an attribute and Sv is the subset of S for which attribute A has value v" (Quinlan, 1986).

$$Entropy = \sum - p_i \log_2 p_i \qquad \text{Equation 3.2}$$

p_i probability of class i

The Decision Tree (J48) has been considered for this study because it handles continuous attributes, "training data with missing attribute values and post pruning trees after their creation" (Michell, 1997–1999).

3.5.2 Artificial Neural Networks (ANN)

The artificial neural network (ANN) consists of data processing units called neurons (nodes) that are arranged in layers as shown in Figure 3.10. The ANN model "is constructed through the learning process which involves modifying the weights of those nodes that are responsible for the error" (Callan, 2003).

A neural network can be made to cluster data by the process of competition. A network's nodes can also be made to undergo a self-organisation process so that the location of the network's nodes, when drawn in the input space, mimic the topology of the training data. A typical similarity measure like the Euclidean distance can be used. For a self-organising feature map, all cluster nodes connect to the input node. The input node serves to distribute the input features of a sample to all cluster nodes (Callan, 2003). During training, nodes compete for a sample. The winning node, which is the node closest to the input sample, adapts its weights so that it becomes closer (i.e. more similar) to the current input sample (Callan, 2003).

There are two phases of training. During the first phase, "nodes that are neighbours of the winning node are allowed to update their weights" (Callan, 2003). The neighbourhood of nodes that are allowed to update their weights decreases during the first phase. During the second phase, "all weights are adjusted by small amounts until the network converges" (Callan, 2003). Once trained, the network can be used to classify an unknown sample based on similarity to the sample with which the network was trained.

Another artificial neural network is the radical basis function. It solves a non-linear problem by casting input samples into a higher dimensional space in a non-linear way. The first layer of weights is learned using an unsupervised technique. The weights are the centres of a set of base-functions. A commonly used basis function is the Gaussian. The second layer of weights can be found using singular value decomposition (SVD), but the output nodes must be linear (Callan, 2003).

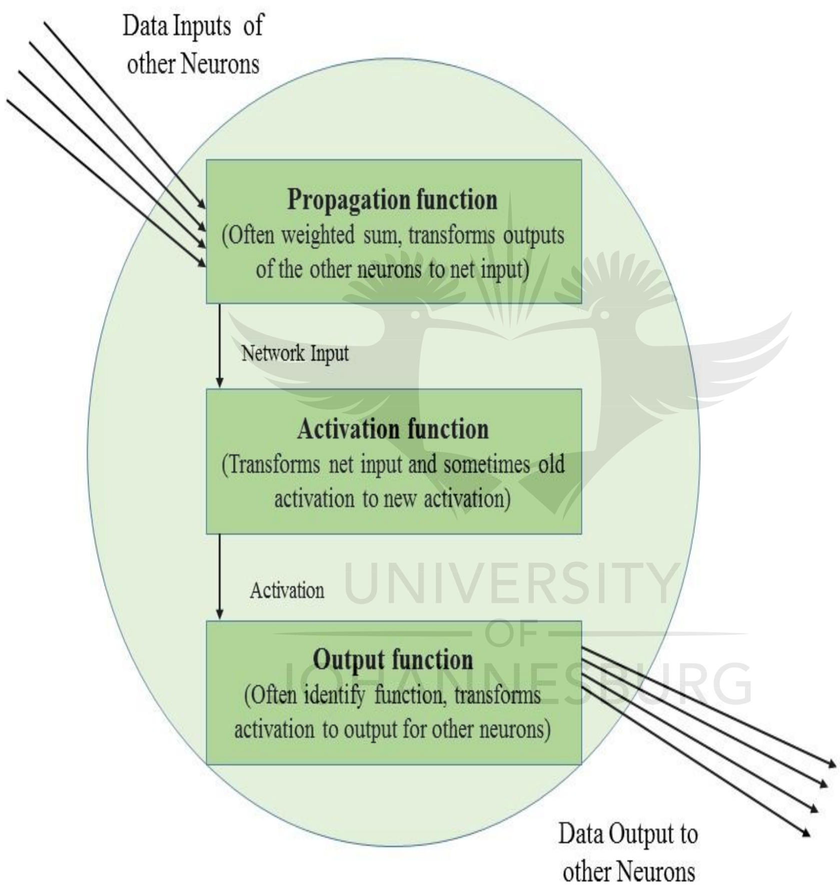


Figure 3.10: Data processing of a neuron. The activation function of a neuron implies the threshold value (Callan, 2003)

3.5.2.1 A simple artificial neuron

A basic computational element (model neuron) "is called a node or unit. It receives input from some other units, or from an external source. Each input has an associated weight *w* as shown in Figure 3.11, which can be modified to model synaptic learning" (Callan, 2003). The unit computes some "function *f* of the weighted sum of its inputs. Its output, in turn, can serve as input to other units" (Callan, 2003).

- The weighted sum "is called the net input to unit *i*, often written *neti*. The w*ij* refers to the weight from unit *j* to unit *i*" (Callan, 2003).
- The function *f* is the unit's activation function.
- "*f* is the identity function, and the unit's output is just its net input. This is called a linear unit" (Callan, 2003).

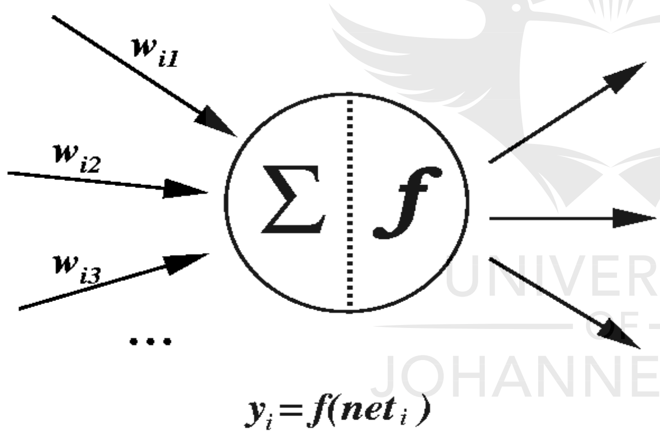


Figure 3.11: A simple artificial neuron (Callan, 2003)

"The learning methods in neural networks are classified into three basic type" (Callan, 2003):

- supervised learning – there is a teacher provided;
- unsupervised learning – there is not teacher; and
- reinforced learning.

These methods “are further categorized based on the rules used as Hebbian, gradient descent, competitive learning and stochastic learning” as shown in Figure 3.12.

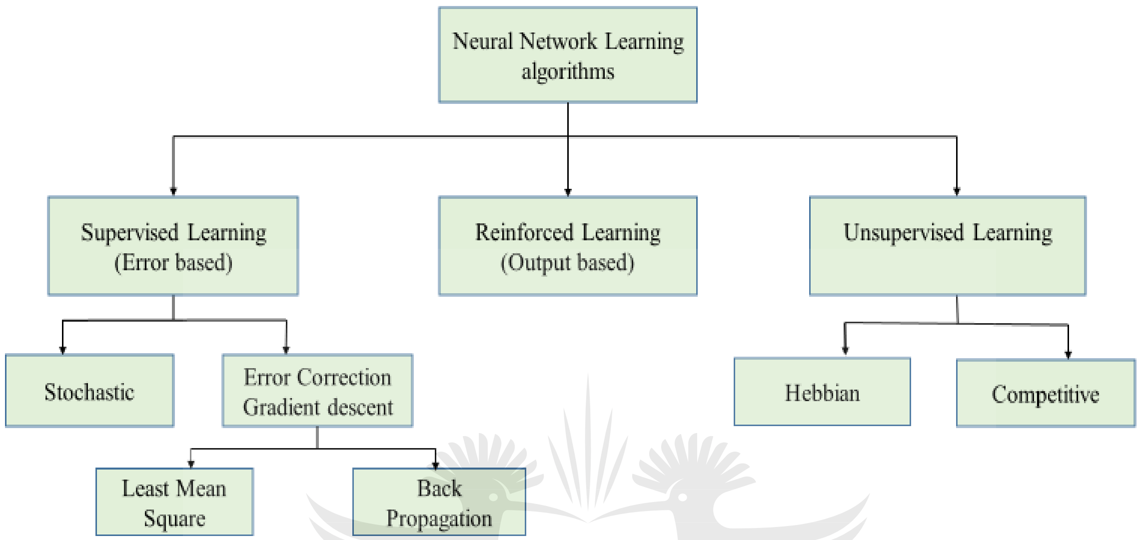


Figure 3.12: Learning methods in neural networks (Callan, 2003)

How to calculate the prediction error

$Error_i = Output_i(1 - Output_i)(Actual_i - Output_i)$ Equation 3.3

where: Errori is “the error from the i-th node, Outputi is the value predicted by a network, Actuali is the real value (which the network should learn)” (Callan, 2003).

$Error_i = Output_i (1 - Output_i)\sum_{j=1} Error_j w_{ij}$ Equation 3.4

The advantages of neural networks are as follows (Callan, 2003):

- “They can work fine in case of incomplete data (where there is missing data)”
- “When an element of the neural network fails, it can continue without any problem by their parallel nature”.

- A neural network “learns and does not need to be reprogrammed”.
- Complex linear and nonlinear relationships can be derived using neural networks.
- “Neural networks are flexible as they can operate using one or more descriptors and/or response variables. They can also be used with categorical and continuous data”.
- Neural networks are less sensitive to noise than statistical regression models are.

The disadvantage of a neural network is that the neural network needs training to operate and requires high processing time for large neural networks (Callan, 2003). ANN is a black box and thus is not possible to explain how the results are computed. In the case of optimising parameters, there are many parameters to be set in a neural network and optimising the network one must ensure to avoid overtraining (Callan, 2003).

3.5.3 K-Means Clustering (K-Means)

K-Means Clustering “is an algorithm to classify or to group your objects based on attributes or features into K number of groups (clusters)” (Teknomo, 2007). K “is positive integer number”. The grouping “is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid” as shown in Figure 3.13 (Teknomo, 2007).

This algorithm “will do the three steps below until convergence

Iterate until stable (= no object move group)” (Teknomo, 2007):

1. Determine the centroid coordinate

2. “Determine the distance of each object to the centroids”

3. “Group the object based on minimum distance (find the closest centroid)”

If the number of data points is less than K, then we assign each data point as the centroid of the cluster (Teknomo, 2007). Each centroid “will have a cluster number. If the number of data is bigger than K, for each data, we calculate the distance to all centroid and get the minimum distance” (Teknomo, 2007). The data “is said belong to the cluster that has a minimum distance from this data. Since we are not sure about the location of the centroid, we need to adjust the centroid location based on the current updated data" (Teknomo, 2007). Then "assign all the data to this new centroid". "This process is repeated until no data is moving to another cluster anymore" (Teknomo, 2007).

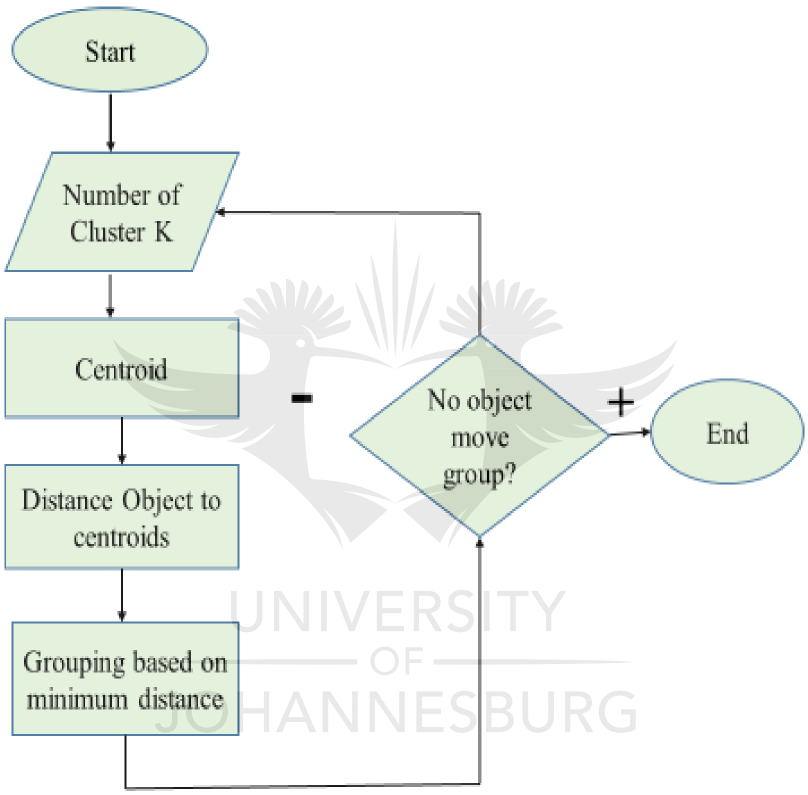


Figure 3.13: How the K-Means Clustering Algorithm Works (Teknomo, 2007)

K-Means Clustering method "partitions *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean" (Teknomo, 2007). It is an unsupervised learning algorithm. The K-Means algorithm clusters "are based upon the data point's proximity to the mean of the cluster". The cluster's "mean is then recomputed, and the process begins again. Here's how the algorithm works" (Teknomo, 2007):

1. The algorithm arbitrarily "selects *k* points as the initial cluster centers ('means')".
2. Each point in the dataset "is assigned to the closest cluster, based upon the Euclidean distance between each point and each cluster center".
3. "Each cluster center is recomputed as the average of the points in that cluster".
4. Steps 2 and 3 "are repeated until the clusters converge.

A cluster "is an ordered list of objects, which have some common characteristics. The objects belong to an interval [a, b], in our case [0, 1]" (Teknomo, 2007). The distance between two clusters "involves some or all elements of the two clusters, the clustering method determines how the distance should be computed" (Teknomo, 2007).

3.5.3.1 Advantages and disadvantages of K-Means Clustering:

It is "its simplicity and speed which allows it to run on large datasets. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments1" (Teknomo, 2007). It minimises "intra-cluster variance but does not ensure a global minimum of variance" (Teknomo, 2007). It is the requirement "for the concept of a mean to be definable, which is not always the case". For such datasets, the k-medoids variants are appropriate. An alternative "could be going for k-medians clustering" (Teknomo, 2007).

This algorithm "can have empty clusters if no points are allocated to a cluster during the assignment step and this is a problem" (Teknomo, 2007). The solution would be "to choose a replacement centroid, otherwise, the squared error will be larger than necessary" (Teknomo, 2007). When outliers are present, "the resulting cluster centroids (prototypes) may not be as representative as they otherwise would be and thus, the SSE will be higher as well" (Teknomo, 2007). In post-processing, "to get better clustering, reduce the SSE that is a difficult task" (Teknomo, 2007).

3.5.4 Multi-Layer Perceptron Neural Network (MLP)

Roy et al. (2005) says "the MLP data processing paradigm is inspired by the biological nervous system processes information". The MLP is composed of highly interconnected processing elements organised in layers. The MLP "is applicatory to non-linearly separable data" (Le &

Zuidema, 2014). To handle non-linearly separable data, the perceptron is extended to a complex structure called a hidden layer(s). In MLP, neuron layers are stacked in such a way that the output of a neuron in a layer is only allowed to be an input to neurons in the upper (Le & Zuidema, 2014) as shown in Figure 3.14.

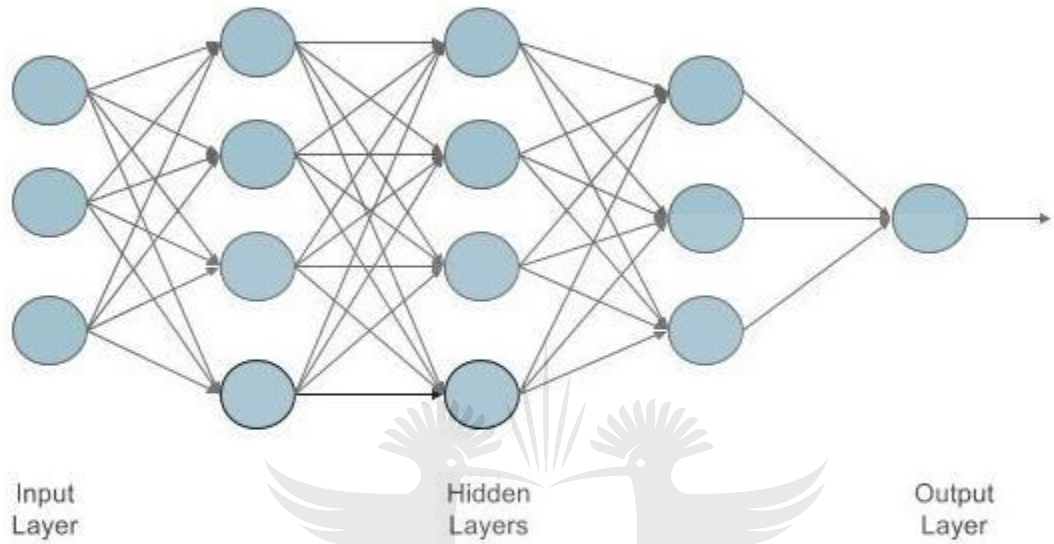


Figure 3.14: A three-layer MLP network consisting of input, hidden and output layers (Roy et al., 2005)

The advantages of a Multi-Layer Perceptron are as follow:

- "It has an ability to learn how to do tasks based on the data given for training called initial experience" (Ruck, Rogers, Kabrisky, Oxley & Suter, 1990).
- MLP "yield the required decision function directly via training".
- "A two-layer back propagation network with sufficient hidden nodes has been proven to be sufficient" (Cybenko, 1989).

The disadvantages of MLP are as follow:

- When there is a new concept to be introduced, it retains the whole model (Su, Jean & Chang, 1996).
- The MLP takes a long time to train and produce a good model.
- The MLP has been applied in text to phoneme mapping (Turban & Frenzel, 1992), among other applications.
- MLP "requires tuning a number of hyperparameters such as the number of hidden neurons, layers, and iterations" (Pedregosa et al., 2011).
- MLP "is sensitive to feature scaling" (Pedregosa et al., 2011).

3.5.5 Support Vector Machine (SVM)

SVMs is a "supervised learning method used for classification and regression" (Vapnik, 1995). It is a generalised linear classifier. An SVM "can simultaneously minimize the empirical classification error and maximize the geometric margin. An SVM is thus called a Maximum Margin Classifier" (Vapnik, 1995). The SVM "is based on the Structural Risk Minimization (SRM)" (Vapnik, 1995). The SVM maps the input vector to a higher dimensional space where a maximal separating hyperplane is constructed. Two parallel hyperplanes "are constructed on each side of the optimal hyperplane that separate the data. The optimal hyperplane maximizes the distance between the two parallel hyperplanes" (Vapnik, 1995). An assumption "is made that the larger the margin between these parallel hyperplanes the better the generalization ability" (Vapnik, 1995).

SVM "can start with a search for an optimal hyperplane that satisfies the request of classification. Secondly, an algorithm can be used to make the margin of the separation beside the optimal hyperplane maximum while ensuring the accuracy of correct classification" (Vapnik, 1998). The separable data can be classified into classes effectively. Structural risk "minimization-based learning algorithms, SVM have better generalization abilities compared to other traditional empirical risk minimization (ERM) based learning algorithms" (Vapnik, 1998).

For a binary problem, there are training data points $\{Xi, Yi\}$, $I = 1, \dots \dots \dots \dots I, yiE\{-1, 1\}$. Boser, Guyon and Vapnik (1992) suggested "there are some hyperplanes which separate the positive from the negative examples". Point x which lies on the hyperplane satisfy $w.x + b = 0$, where w is normal to the hyperplane, $b/||w||$ in Figure 3.7 is the perpendicular distance from the hyperplane to the origin. Suppose that all the training data satisfies constraints in equation 3.5 (Madzarov, Gjorgjevikj & Chorbev, 2009):

"$xi.w + b \geq +1$ *for* $yi = +1$,"

"$xi.w + b \leq -1$ *for* $yi = -1$," (Madzarov et al., 2009) Equation 3.5

These constraints can "be combined into one set of inequalities" (Madzarov et al., 2009). The points in Figure 3.15 that lie on the hyperplane *H1* and *H2* hyperplane are called support vectors.

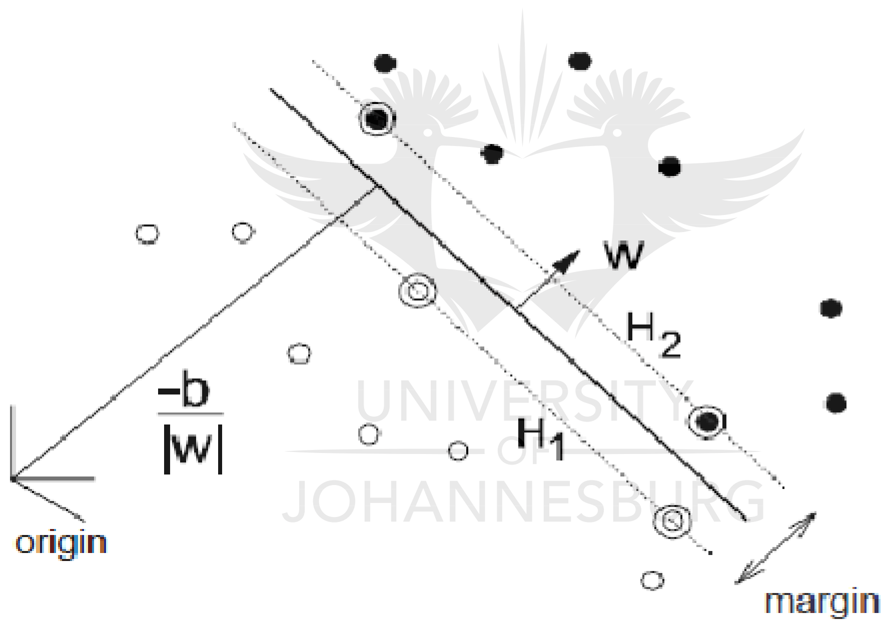


Figure 3.15: Linear separating hyperplanes for the separable case, support vectors are circled (Madzarov et al., 2009)

3.5.5.1 Multiclass for Support Vector Machine (MSVM)

SVM is "originally designed for binary classification for a linear separating hyperplane to separate the data into two class, SVM's have been expanded to multi-class classifiers which can separate the data into multi-classes” (Chang & Lin, 2011). When dealing with a multi-class a method called SMO is needed (Peck, 2009), libsvm (Chang & Lin, 2011) and liblinear (Fan, Chang, Hsieh, Wang & Lin, 2008). The difference between the binary classification and multiple class problems is that the latter has yi € $\{1, 2, \ldots.k\}$ classes. The methods used to extend the binary classification are one-against-all, which constructs k SVM models where k presents the number of target concepts (Bottou, et al., 1994). One-against-one is another major method that constructs $k(k-l)/2$ classifiers where each one is trained on data from two classes.

The last method “is the direct acyclic graph SVM (DAGSVM) whose training phase is the same as the one-against-one method by solving $k(k-l)/2$ binary SVM classifiers” (Platt, Cristianini & Shwawe-Taylor, 2000). The one-against-all method has been used for this study and libsvm (library for support vector machine) tools from WEKA were used to construct the SVM models. The libsvm library automatically detects more than two-classes and, therefore, it will start to train a multi-class SVM using the one-against-all strategy by default (Chang & Lin, 2011).

There is also the kernel trick in SVM. Kernels used in this study are the radial basis function (RBF). This kernel nonlinearly “maps samples into the higher dimensional space” (Hsu, Chang & Lin, 2003). RBF kernels “can handle the case when the relation between class labels and attributes are nonlinear” (Hsu et al., 2003).

The advantages of SVM are as follow:

- SVMs “deliver a unique solution since the optimality problem is convex”.
- “The kernel in SVMs gaining flexibility has been the choice of the form of the threshold separating solvent from insolvent companies” (Auria & Moro, 2008). “This is always the case of non-regularity in the data”.
- The SVM “prediction of accuracy is generally high”.
- “SVM provides fast evaluation of the learned target function” (Madzarov et al., 2009).
- “SVMs provide a good out-of-sample generalization, if the parameters C and r (in the case of a Gaussian kernel) are chosen well” (Auria & Moro, 2008).

The disadvantages of SVM is that it takes longer when generating training results and it is difficult to understand the learned functions (Vapnik, 1995). The SVM lacks the transparency of results (Auria & Moro, 2008). The SVMs requires parameter turning to get an accurate model. The SVM has been applied successfully in trading (Blokker, 2008), object recognition (Wang & Hu, 2002) and face detection (Kumar & Poggio, 2000).

3.6 Search Methods

A search algorithm can only have the knowledge provided in the search space (a pool of attributes or features). These search methods are assigned to the feature selection process of Figure 3.7.

3.6.1 Best-First Search (BFS)

The best-first search (BFS) technique (shown in Figure 3.16) allows us switching between paths, thus gaining the benefits of depth-first and breadth-first search. At each step (level), the most promising node is chosen. Nodes are presented by either the parent node or the child node as shown in Figure 3.16. If one of the nodes chosen "generates nodes that are less promising, another node at the same level can be chosen and in effect, the search changes from depth to breadth-first search" (Sharma, 2008). If resulting nodes are no better than the previously unexpanded node and branch, "the search method reverts (backtracks) to the descendants of the first choice and proceeds". The best-first search "sorts nodes in the frontier list by increasing values of an evaluation function, *f (n)*, that incorporates domain-specific information" (Sharma, 2008). A key component in BFS algorithm "is a heuristic function, *h (n)*, which is estimates the cost of the cheapest path from the n to a goal node" (Sharma, 2008).

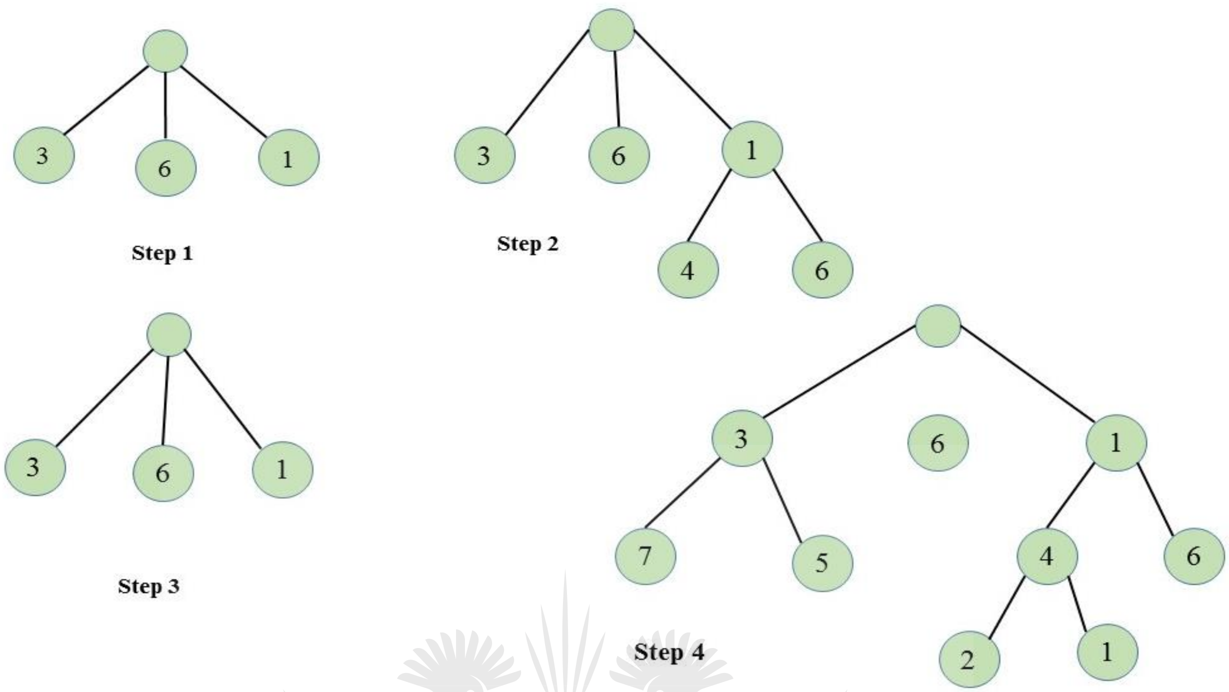


Figure 3.16: Shows BFS method; Numbers are nodes; There are links connecting nodes (Sharma, 2008)

The main advantage of BFS is that BFS "are saved to enable revisits if an impasse occurs on the apparent best path". BFS was used in section 4.3 when doing attribute selection to select the best three attributes from a pool of attributes for the study.

3.6.2 Ranker Search

The ranker method "ranks attributes by their individual evaluation use in conjunction with attributes evaluators with the parameter generate ranking (true or false)" (Dinakaran & Thangaiah, 2013). The number to select attribute values is set according to which attributes can be discarded from the set of attributes when using the ranker search method. This method ranks attributes according to the selected attribute from the supplied dataset. Ranker provides a rating to the attributes by their score provided by the evaluator (Kohavi & John, 1997). This method is used when selecting the best three attributes among the rest of the attributes in the study.

3.6.3 Greedy Best-Best Search (GBS)

The Greedy-Best (GBS) search "uses, as an evaluation function, $f(n) = h(n)$ sorting nodes in the front list by increasing values off them" (Russell & Norvig, 2003). The GBS "tries to expand the node that is closest to the goal with the assumption that it is likely to lead to a solution quickly" (Russell & Norvig, 2003). The GBS often performs well. GBS "tend to find good solutions quickly, although they are not always optimal ones".

3.7 Cross-Validation (CV)

Cross-validation is a model validation technique. In other words, CV is used for assessing how a model will generalise to a data set. One can decide on a fixed (n) of folds for the dataset. Fold is when the dataset is broken down to 10 sets of size $n/10$. The data is split into n folds. In each turn, one split fold is used as a test and the remaining folds for training. In this study, a 10-folds cross-validation was used, in which 9-folds of the data were used for training and 1-fold cross-validation were used for testing (re-evaluating models).

The procedure for the 10-folds cross-validation (Tan, Steinbach & Kumar, 2006) is as follows:

Step 1: "Break data into 10 sets of size $n/10$"

Step 2: "Train on 9 datasets and test on 1".

Step 3: On the next iterations, ensure each of the 10-folds have been used as a test set.

Step 4: "Repeat 10 times and take a mean accuracy".

This approach "has the advantage that all the data can be used for training and none has to be held back in a separate test set" (Allen, 1974).

3.8 Supervised Learning

Cunningham, Cord & Delany, 2008) "defines Supervised Learning as mapping between a set of input and output data and using the mapping to predict outputs for unseen data". It generalises well. Supervised learning occurs when the agent is given feedback about the value of the action (Talwar & Kumar, 2013).

The main advantage of Supervised Learning "is that all classes manipulated by the algorithm are meaningful to humans" (Sathya & Abraham, 2013). It can be easily used "for discriminative pattern classification, and for data regression" (Sathya & Abraham, 2013). In this study, supervised learning was used in section 4.3 on all experiments.

3.9 Discussion of the selected algorithms of the study

The Decision Trees (J48), the Multi-Layer Perceptron (MLP), the Support Vector Machine (SVM) algorithms, K-Means Clustering(K-Means) are used in Section 4.3 for experiments. The Decision Trees, MLP and SVM are all supervised learning algorithms (Maimon & Rokach, 2005).

Decision Trees easily compute the expected value and more than one decision maker can be easily involved with the decision processes (Kohavi & John, 2002). The J48 algorithms perform well in the presence of noise (Tan et al., 2006). J48 smaller-sized trees are easy to interpret as it generates understandable rules.

The MLP is a preferred technique for gesture recognition and they yield the required decision function directly via training (Su et al., 1996). The drawback of the MLP method is that it is difficult to interpret its model. The MLP and J48 can be compared since both can handle interaction between variables. The MLP does not make any assumptions regarding the underlying probability density functions (Cybenko, 1989).

The SVM "performs well on data sets that have many attributes, even if there are very few instances on which to train the model. Compared to the MLP, SVM produces results quicker" (Kantardzic, 2011). The SVM "can model complex, real-world problems such as text and image classification". The SVM allows explicit control over the complexity of the derived models by tuning some parameters (Kordon, 2009). When constructing the SVM in Section 4.3, the parameters that were set are gamma and the cost *C*. The cost parameter was kept constant while the gamma value was changed five times to obtain more than one model.

The K-Means algorithm “is an efficient algorithm because it is simpler in its operations” (Teknomo, 2004 - 2016). This algorithm has been used in financial analysis and image segmentation (Teknomo, 2004 - 2016). One of the problems with the K-Means algorithm “is that empty clusters can be obtained if no points are allocated to a cluster during the assignment step” (Teknomo, 2004 - 2016). To address this problem, “choose a replacement centroid, since otherwise, the squared error will be larger than necessary. When outliers are present, the resulting cluster centroids (prototypes) may not be representative and the SSE will be higher which needs to be reduced SSE which is a difficult task” (Teknomo, 2004 - 2016). The advantage of this algorithm “is its simplicity and the speed which allows it to run on large datasets” (Teknomo, 2004 - 2016).

3.10 Chapter conclusion

Quantitative research methods were discussed in this chapter. This chapter described the research methodology, including travellers, sampling, data collection methods and the algorithms used to construct the prediction models. The selected algorithms for the study are Decision Trees (J48), the MLP, the SVM and the K-Means algorithm. The reason for selecting these algorithms was considering their strengths that they could help to construct the best model for predicting the flight punctuality. All the selected algorithms and search methods are used in section 4.3.

CHAPTER 4: EXPERIMENTS AND RESULTS

4.1 Introduction

This chapter contains all the experiments that were conducted using machine learning algorithms. Waikato Environment for Knowledge Analysis (WEKA) is an open source software that contains a number of machine learning algorithms. In the experiments, four machine learning algorithms and the attributes selection tool were used for obtaining the results in this study. All the results obtained from the flight departure delay predictions were compared, and the best prediction model was chosen.

4.1.1 Data collection

The data used for this study was collected from OR Tambo International Airport (ORTIA) in Johannesburg (JNB). This is the "Departure Flights" as shown in Annexure 4A. This data were pre-processed using Microsoft SQL Server Management Studio. The data were in date, time and text format as shown in Table 4.1. Table 4.1 only shows a sample of 8 of the 2 234 instances of flight departure delays for illustration purposes.

Table 4.1 A sample of 8 Instances (Flight Departures from JNB (Adapted from: Website http://ortambo-airport.com/flights/departures.php))

Date: Fri 07-Jul-2017

Time Period: 9:00 PM - 12:00 AM ◉ Departures ○ Arrivals

Airport: (JNB) O.R. Tambo International Airport Johannesburg, ZA

Instance	Flight	Carrier	Destination	Departure	Status
1	ET 849	Ethiopian Airlines	(CPT) Cape Town	4:45 AM	En Route Delayed
2	SA 7203 ^	South African Airways	(CPT) Cape Town	4:45 AM	Landed Delayed
3	SA 8773	South African Airways	(SIS) Sishen	5:50 AM	Landed
4	KEM 402	CemAir	(SIS) Sishen	5:55 AM	Scheduled
5	JE 241	Mango	(DUR) Durban	5:55 AM	Scheduled
6	JE 123	Mango	(CPT) Cape Town	5:55 AM	Cancelled
7	SA 2003 ^	South African Airways	(CPT) Cape Town	5:55 AM	Landed
8	FA* 200	Safair	(CPT) Cape Town	5:55 AM	En Route

4.2 Pre-processing data

In the data, some flight status values were labelled as “unknown”. The “unknown” is also called missing data. The median of the surrounding values to replace the “unknown” values could have been used in the place of “unknown” data. The study could have used the mean of the whole flight status column values in place of the “unknown values”. Since this study is focused on departing flights, all the flight data with the status values labelled as “landed” (which means the flights have arrived) were removed. The flight status values labelled “scheduled” were removed from the data, as these values tell us that the flight has not yet departed. Only South African Airways carriers were considered for this study and all its destinations. The “departure’” column, which represents the departure time of flights that have departed, was in a string (text) format when it was collected.

Pre-processing involved converting the string format into integer format using SQL Server Management Studio scripts shown in Figure 4.1. This conversion is shown in more detail in Annexure 4B.

```
ALTER TABLE [AdventureWorks 2016].[dbo]. [South African Airways Flight Departure Data]

ADD TimeOfTheDay AS

CASE

WHEN [Departure] = ‘2:55 AM’

THEN ‘0255’

WHEN [Departure] = ‘4:45 AM’

THEN ‘0445’

.

.

ELSE ‘Unknown Time’

END
```

Figure 4.1: Conversion of ‘Departure’ time from character format into string format and add the conversion results to a column named “TimeOfTheDay”

The data was obtained from a website in a table format. This data contained text and date format that represented attributes and instances that influence flight departure delays. The “Date” is represented as “Month” (MON), “DayOfMonth” (DOM) and “DayOfWeek” (DOW) as shown in Table 4.2, “Flight” as the flight number for each airline, “Carrier” as the name of the airline, “Destination” as to where the flight is flying to, “Departure” as the ‘DepartureTime’ (DT) and ‘TimeOfDay’ (TOD), “Status” as the class to show the three attributes of either the departing flight was “Delayed”, “Cancelled” or “On Time” as shown in Table 4.1 (Gopalakrishnan & Balakrishnan, 2017).

This study used data for South African Airways (SAA) only. The study also used data for seven days a week. The sample of the text, data and numeric data before pre-processing is shown in Annexure 4A. The number of instances used to build the models was 2 234.

The data in Table 4.1 are Departure Flights for SAA carrier that were collected from (ORTIA) website. All data from the status column labelled as “Landed” and “Scheduled” were discarded since this study considered departing flights only. “En Route” status indicates the flights that have already departed and were considered to be “On Time” in this study. The conversion procedure for converting flight departures data from text and date format to nominal values is in section 4.2.1.

4.2.1 Conversion of text and numeric flight departure delays to nominal values

The nominal values included grouping the daytime into morning, midday and evening. Morning departures are for travellers that are either doing same-day workshops or going for work purposes. The same applies for evening departures, it is normally for travellers returning from workshops or going for events the following day. The nominal values were also categorised by the beginning of the month, middle of the month and month end since a later period is always record-peak volumes for travellers. The nominal values were also categorised by day of week, as on weekends the traffic is high and flight prices are higher as opposed to the beginning of the week or midweek.

The conversion of ORTIA Departure Flights for the SAA carrier to nominal data was performed using a SQL Server Management Studio. What follows are the methods that were used for converting text and numerical data to nominal values. Month (03 July 2017 to 31 August 2017), WeekOfMonth (MonthBeginning, MonthMiddle, MonthEnd). MonthBeginning (> = 1 and < = 10), MonthMiddle (> 10 and < = 20), MonthEnd(> 20 and < = 31) DayOfWeek(Monday–Sunday) and TimeOfDay (Morning, Afternoon, Evening, AfterMidnight), Morning (> = 0500 and < = 1200), Afternoon (> 1200 and < = 1800), Evening (> 1800 and < = 2400), AfterMidnight (> 2400).

These attributes mean the following: “week of month” is the specific week of the month in which the flight departed, and this could be the first week of the month or week that falls in the middle of the month or the last week of the month. The attribute “Day of week” is the day on which the flight departed, counting from Monday to Sunday. “The time of day” is the time at which the flight departed, which can either be in the morning or afternoon, or evening or after midnight.

4.2.2 The recipe for converting data into nominal data

The step-by-step process to convert the flight data from text and date, and numeric to nominal was done using SQL Server Management Studio Scripts. Figure 4.2 shows the query referencing for (SAA) Flight departure data in a SQL server database with values converted to nominal.

- There are 2 234 instances that were converted from text and numeric data to nominal values.
- Converting ‘WeekOfMonth’ variables to nominal using SQL Server Management Studio was done follows:

1) ALTER TABLE [AdventureWorks2016].[dbo].[South African Airways Flight Departure Data]
2) ADD WeekOfMonth AS
3) CASE
4) WHEN (CAST([DaysOfMonth]) as INT)) >= 1 AND (CAST ([DaysOfMonth as INT)) <= 10
5) THEN ‘Month Beginning’

```
6)  WHEN (CAST([DaysOfMonth]) as INT)) >10 AND (CAST([DaysOfMonth]) as INT)) <=
    20
7)  THEN ‘Month Middle’
8)  WHEN (CAST([DaysOfMonth]) as INT)) >20 AND (CAST([DaysOfMonth]) as INT)) <=
    31
9)  THEN ‘Month End’
10) ELSE ‘Unknown Week’
11) End
```

Figure 4.2: Conversion of ‘DaysOfMonth’ from integer (number) format into nominal data for ‘WeekOfMonth’

What follows is the explanations of the lines of the code in Figure 4.2.

1. Type Figure 4.2 into a new query, the query referencing the table South African Airways Flight Departure Data in any database with values that should be converted to nominal.

2. The first line in Figure 4.2 means modify a table in the database named as [AdventureWorks2016], table named as.[dbo].[South African Airways Flight Departure Data].

3. The second line means we are adding a column name called ‘WeekOfMonth’.

4. The third line means we are using a CASE to provide when-then logic type of logic to the SQL in creating the ‘WeekOfMonth’ column.

5. The fourth line means that, when the value in the column named ‘DaysOfMonth’ is read by the SQL query, the value must be CAST, which is to convert the values data type from string (letters) data type into integer (number) data type first, and after the value is casted/converted to an integer, we check if the value is greater or equal to 1 and less than or equal to 10.

6. The fifth line means we are converting the 'DaysOfMonth' to 'Month Beginning'.

7. The sixth line means that, when the value in the column named 'DaysOfMonth' is read by the SQL query, the value must be CAST, which is to convert the values data type from string (letters) data type into integer (number) data type first, and after the value is casted/converted to an integer, we check if the value is greater than 10 and less than or equal to 20.

8. The seventh line means we are converting the 'DaysOfMonth' to 'Month Middle'.

9. The eighth line means that, when the value in the column named 'DaysOfMonth' is read by the SQL query, the value must be CAST, which is to convert the values data type from string (letters) data type into integer (number) data type first, and after the value is casted/converted to an integer, we check if the value is greater than 20 and less than or equal to 31.

10. The ninth line means we are converting the 'DaysOfMonth' to 'Month End'.

11. The tenth line means that if the 'DaysOfMonth' values do not match the logic, then we convert to 'Unknown Week'.

12. The eleventh line means end of the when-then logic.

Converting to 'TimeOfDay' variables to nominal using SQL Server Management Studio was done follows:

1) ALTER TABLE [AdventureWorks2016].[dbo].[South African Airways Flight Departure Data]
2) ADD TimeOfDay AS
3) CASE
4) WHEN [TimeOfTheDay] >= 0500 AND [TimeOfTheDay] <=1200
5) THEN 'Morning'

6) WHEN [TimeOfTheDay] > 1200 AND [TimeOfTheDay] <=1800

7) THEN ‘Afternoon’

8) WHEN [TimeOfTheDay] > 1800 AND [TimeOfTheDay] <=2400

9) THEN ‘Evening’

10) WHEN [TimeOfTheDay] > 2400

11) THEN ‘After Midnight’

12) ELSE ‘Unknown Time’

13) END

Figure 4.3: Conversion of ‘TimeOfDay’ from integer(number) format into nominal data to ‘TimeOfDay’

What follows is the explanation of each line of code:

1. Type Figure 4.3 into a new query, the query referencing the table South African Airways Flight Departure Data in any database with values that should be converted to nominal.

2. The first line Figure of 4.3 means to modify a table in the database named as [AdventureWorks2016] to a table named as.[dbo].[South African Airways Flight Departure Data].

3. The second line means we are adding a column name called ‘TimeOfDay’.

4. The third line means we are using a CASE to provide when-then logic type to the SQL in creating the ‘TimeOfDay’ column.

5. The fourth line means that, when the value in the column named ‘TimeOfTheDay’ is read by the SQL query, we check if the value is greater or equal to 0500 and less than or equal to 1200.

6. The fifth line means we are converting the ‘TimeOfTheDay’ to ‘Morning’.

7. The sixth line means that, when the value in the column name

d ‘TimeOfTheDay’ is read by the SQL query, we check if the value is greater than 1200 and less than or equal to 1800.

8. The seventh line means we converting the ‘TimeOfTheDay’ to ‘Afternoon’.

10. The eighth line means that, When the value in the column named ‘TimeOfTheDay’ is read by the SQL query, we check if the value is greater than 1800 and less than or equal to 2400.

11. The ninth line means we converting the ‘TimeOfTheDay’ to ‘Evening’.

12. The tenth line means that, when the value in the column named ‘TimeOfTheDay’ is read by the SQL query, we check if the value is greater than 2400.

13. The eleventh line means we converting the ‘TimeOfTheDay’ to ‘After Midnight’.

14. The twelfth line means, when the ‘TimeOfTheDay’ values do not match the logic then we convert to ‘Unknown Time’.

15. The thirteenth line means the end of the when-then logic.

What follows is the code Converting ‘TargetConcept’ variables to nominal using SQL Server Management Studio:

1) ALTER TABLE [AdventureWorks2016].[dbo].[South African Airways Flight Departure Data]
2) ADD TargetConcept AS
3) CASE
4) WHEN [Status] = ‘Delayed’
5) THEN ‘Delayed’
6) WHEN [Status] = “Cancelled”
7) THEN “Cancelled”

```
8)  WHEN [Status] = 'En Route'
9)  THEN 'On Time'
10) ELSE 'Unknown TargetConcept'
11) END
```

Figure 4.4: Conversion of values from 'Status' column into nominal data to 'TargetConcept'

What follows is the explanation for each line of code:

1. Type Figure 4.4 into a new query, the query referencing the table South African Airways Flight Departure Data in any database with values that should be converted to nominal.

2. The first line Figure 4.4 means modify a table in the database named [AdventureWorks2016], table named as [dbo].[South African Airways Flight Departure Data].

3. The second line means we are adding a column named as 'TargetConcept'.

4. The third line means we are using a CASE to provide when-then logic type of logic to the SQL in creating the 'TargetConcept' column.

5. The fourth line means that, when the text in the column named 'Status' is read by the SQL query, we check that when the text is same as 'Delayed'.

6. The fifth line means that we are converting the 'Status' to 'Delayed'.

7. The sixth line means that, when the text in the column named 'Status' is read by the SQL query, we check if the text is the same as 'Cancelled'.

8. The seventh line means we converting the 'Status' to 'Cancelled'.

9. The eighth line means that, when the text in the column named Status is read by the SQL query, we check if the text is same as 'En Route'.

10. The ninth line means we converting the 'Status' to 'On Time'.

11. The tenth line means, if the ‘Status’ text does not match to the logic then we convert to ‘Unknown TargetConcept’.

12. The eleventh line means end of when-then logic.

WEKA takes the comma delimited (*.csv) as input. Microsoft Office Word 2013 does not support this format and only saves the file in semicolon (;) format. After saving the file in a *.csv format, right click on the file, then on the dropdown menu select ‘Notepad’, then right click the notepad file. A menu box will be populated under ‘find’ type semicolon (;) and in the textbox, replace the type comma (,) then click on button replace all. This process means that the file, when uploaded on WEKA, appears in the expected format, the comma delimited (*.csv) format.

The data in Table 4.2 is the same the data from Table 4.1 that has been converted to nominal values. The flight departure delays data was categorised into three target concepts namely “On Time” (a flight departed at the scheduled time), “Delayed” (a flight did not depart at the scheduled departing time) and “Cancelled” (a flight will no longer depart). The sample of the nominal data after pre-processing is shown in detail in Annexure 4C.

Table 4.2 A randomly selected sample of only 8 instances out of 2 234 after data pre-processing

Instance No:	**Attributes**				**Prediction**
	Month (MON)	**Week Of Month (WOM)**	**Day Of Week (DOW)**	**Time Of Day (TOD)**	**Target Concept (TC)**
10	Jul	Month Beginning	Monday	Morning	Delayed
99	Jul	Month Middle	Tuesday	Afternoon	On Time
450	Jul	Month Beginning	Friday	Evening	Delayed
501	Jul	Month End	Wednesday	Afternoon	Cancelled
900	Aug	Month Beginning	Saturday	Morning	Delayed
920	Aug	Month Middle	Thursday	Evening	On Time
1001	Aug	Month Beginning	Sunday	Afternoon	Delayed
1095	Aug	Month End	Monday	Evening	Cancelled

4.3 Experiments and results

In this section, flight departure delays prediction models for OR Tambo were constructed using an attribute selection tool and machine learning algorithms from the WEKA software. These prediction models were created using the training data and testing data.

4.3.1 Experiment for attribute selection

In this study, the “Ranker” and “GreedyStepwise” search methods attribute evaluators were used for attribute selection. The “CfsSubsetEval” and “InfoGainAttributeEval” attribute evaluators were used for attribute evaluation. The “Ranker” search method was used to rank the attributes construct flight departure delays prediction models. The “GreedyStepwise” search method was used to select the best two attributes, and the “Ranker” was used to select the best three attributes. The search method that was determined using the WEKA tool when an attribute evaluator was chosen during attribute selection.

The “InfoGainAttributeEval” evaluator for the decision tree algorithm “evaluates attributes by measuring their information gain with respect to the class” (Witten et al., 2011). It discretises “numeric attributes first using the MDL-based discretisation 35 method (it can be set to binarize them instead)” (Witten et al., 2011). This method, along with the next three, “can treat missing data as a separate value or distribute the counts among other values in proportion to their frequency” (Witten et al., 2011).

The Ranker search method ranks and sorts individual attributes according to their evaluation. It also ranks “individual attributes (not subsets) according to their evaluation” (Witten, et al., 2011). The ranker “not only ranks attributes but also performs attribute selection by removing the lower-ranking ones” (Witten et al., 2011). You can set “a cut-off threshold below which attributes are discarded or specify how many attributes to retain” (Witten et al., 2011). You can specify “certain attributes that must be retained regardless of their rank” (Witten et al., 2011). Using ranking with InfoGainAttributeEval and the Ranker search from WEKA, “can eliminate less useful attributes” (Witten et al., 2011). What follows in Figure 4.5 is ranking using the information gain evaluation measure.

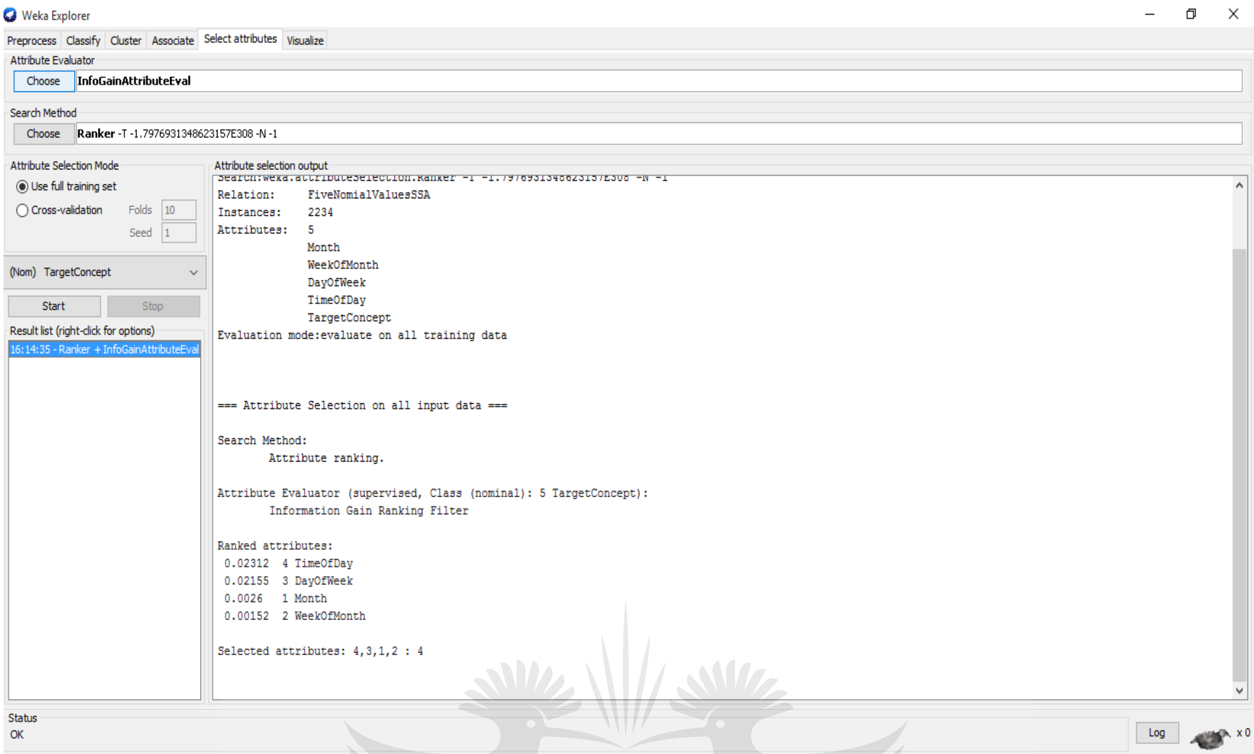


Figure 4.5: Ranking of 4 attributes using "InfoGainAttributeEval" Attribute Evaluator and "Ranker" Search Method

Results in Figure 4.5 shows that the fourth attribute in the training data table "Time of Day" was ranked first as the most effective attribute, the third attribute in the training data table, "Day of Week", ranked second. The first attribute in the training data "Month" ranked as third and the second attribute in the training data "Week of Month" ranked as fourth.

The selected attributes were 4, 3, 1, 2 : 4 (the colon and 4 means that there are 4 attributes ranked). The order of ranking start from highest effective to lowest effective attribute. Thus, 4, 3, 1, 2 is the order of the ranked attributes. The attributes are 4 (Time of Day), 3 (Day of Week), 2 (Week of Month), 1 (Month). The ranked attributes will be used for experiments in different combinations by removing the attributes with a low ranked order (fewer effective attributes). "TimeOfDay" was ranked the highest effective attribute and "WeekOfMonth" was ranked the lowest effective attribute.

The GreedyStepwise search method in each step iteratively evaluates attributes. It enables forward selection and backward elimination searches. The single best informative attributes are then selected and used for constructing the model (Witten et al., 2011). It "searches greedily through the space of attribute subsets" (Witten et al., 2011). With this method, a number of attributes to retain can be specified, as shown in Figure 4.5.

The "CfsSubsetEval" attribute evaluator "assesses the discriminative ability of each attribute individually and the degree of redundancy among them" (Witten et al., 2011). It prefers "sets of attributes that are highly correlated with the class but with low intercorrelation" (Witten et al., 2011). An option "iteratively adds attributes that have the highest correlation with the class, provided that the set does not already contain an attribute whose correlation with the attribute in question is even higher" (Witten et al., 2011).

4.3.1.1 Experiment 1: A model constructed from 2 attributes selected by "CfsSubsetEval" attribute evaluator and "GreedyStepwise" search method

The procedure for attribute selection allows one to discard irrelevant attributes and thus, reduce the dimensionality of the dataset. It also allows a combination of attributes per model to be selected as follows:

Step 1: Load data on WEKA

The training data set from section 4.2 was loaded on WEKA as follows:

1. On WEKA Click on button "Explorer", to show all attributes ("Month", "WeekOfMonth", "DayOfWeek", "TimeOfDay" plus the Class attribute). One can select all or some of the attributes.

2. On Attribute Selection Mode click "Use training set".
3. On the pre-process tab, click on "Open File", then select your *.CSV file. The button "Visualize all" will let you bring up a screen showing all the distributions at once as shown in Figure 4.6 "The Attributes frame allows the user to modify the set of attributes using select and remove options".

4. Click Save button to save the dataset file as “*.arff”.

NB: All the experiments that follow start with the same steps (from 1 to 5). Thus, in subsequent experiments steps 1 to 5 are assumed.

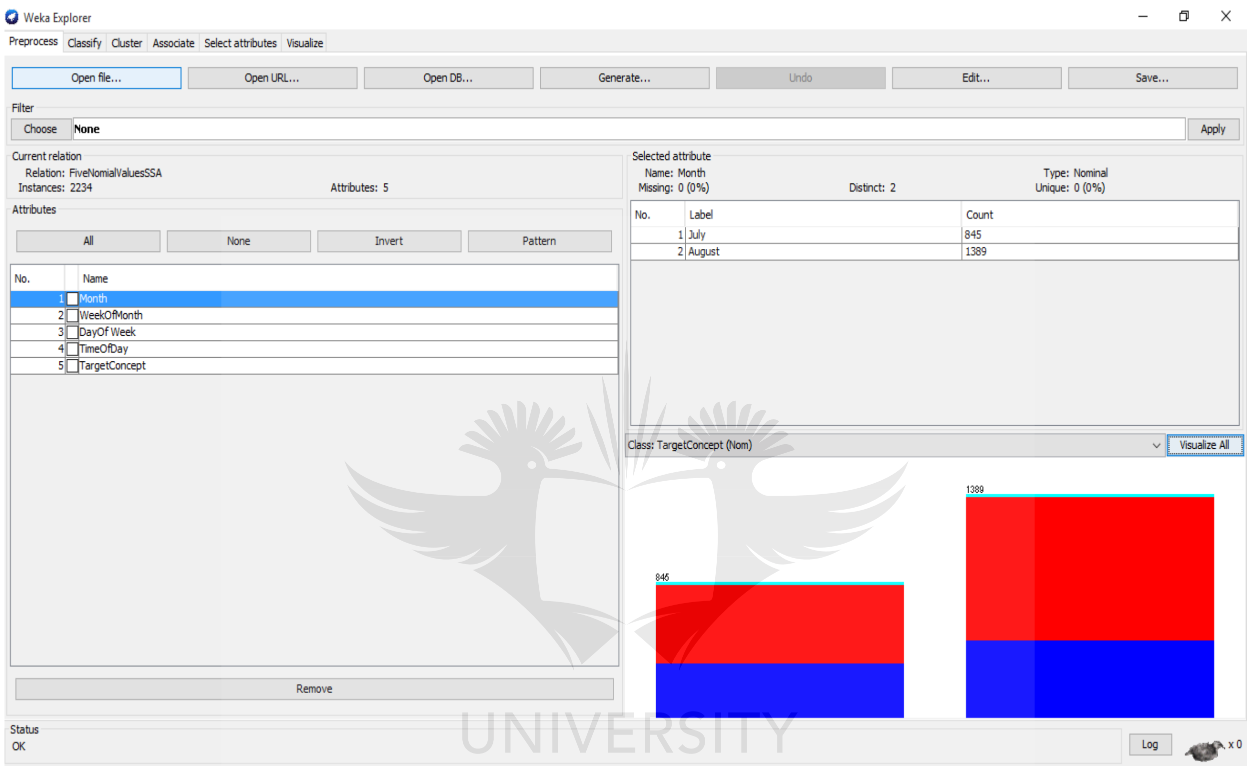


Figure 4.6: Flight departures (nominal data) for attributes

Step 2: Attribute Selection using “CfsSubsetEval” evaluation with the “GreedyStepwise” search method.

1. Click on the “Select attributes” tab as shown in Figure 4.6.
2. Under the “Attribute Evaluator” frame, choose relevant evaluation method “CfsSubsetEval”, Under the “Search Method” frame, choose “GreedyStepwise”.
3. In the Attribute Selection mode, click “Use training set” and then click on the start button. Results are shown in Figure 4.6.

4. In Figure 4.6, tick the attributes from results (based on ranking order) to be removed and click on “Remove” button, then you will be left with two attributes.

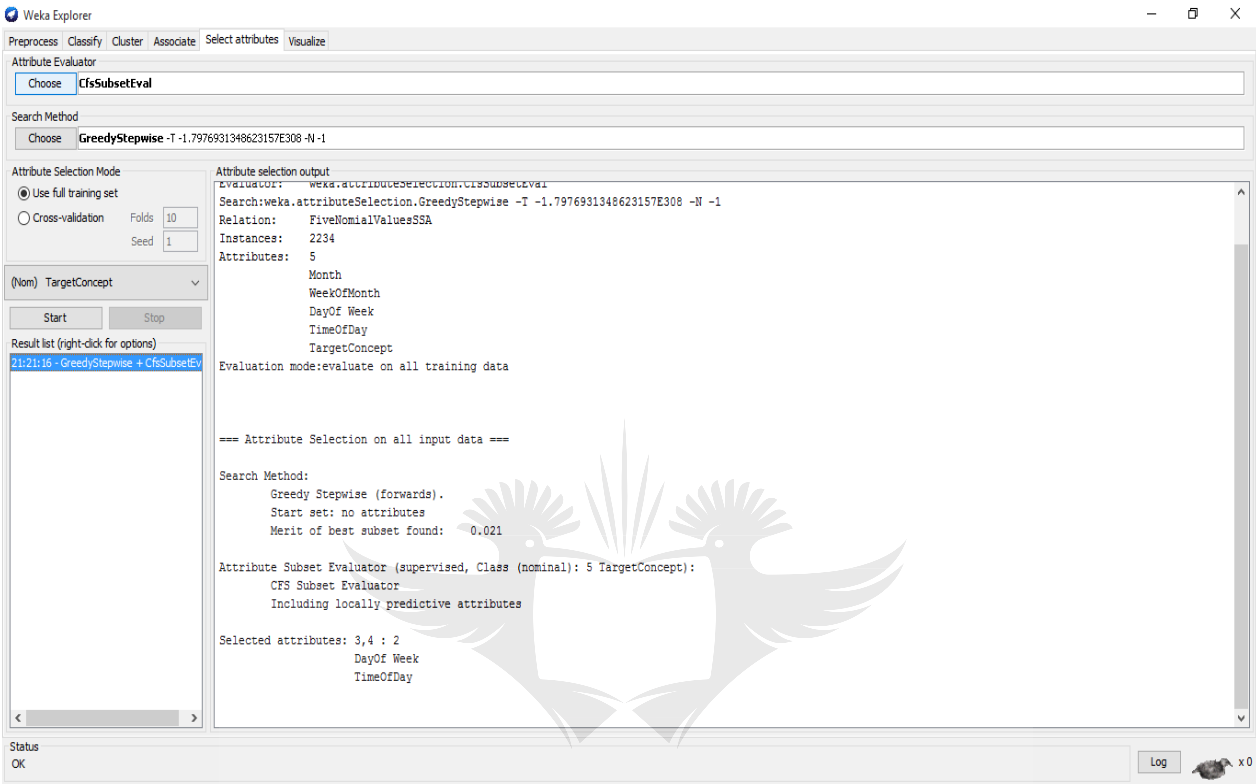


Figure 4.7: Two selected attributes using “CfsSubsetEval” attribute evaluator and “GreedyStepwise” search method

Figure 4.7 shows the selected attributes 3, 4 : 2. The 2 after the colon means there were 2 attributes selected that are effective. The 3 comma 4 means the third attribute from the attributes list which is “DayOfWeek” and the fourth attribute from the attributes list which is “TimeOfDay” were the most effective attributes. Figure 4.7 means that the 2 sets of the best attributes are DOW and TOD. In Figure 4.7 the “Selected attributes are 3, 4 : 2” (the 2 after the colon means there were 2 attributes selected that are highly discriminative). The 3, 4 before the colon means the third attribute from the attributes list which is “DayOfWeek” (DOW) and the fourth attribute from the attributes list which is “TimeOfDay” (TOD) were the most effective attributes.

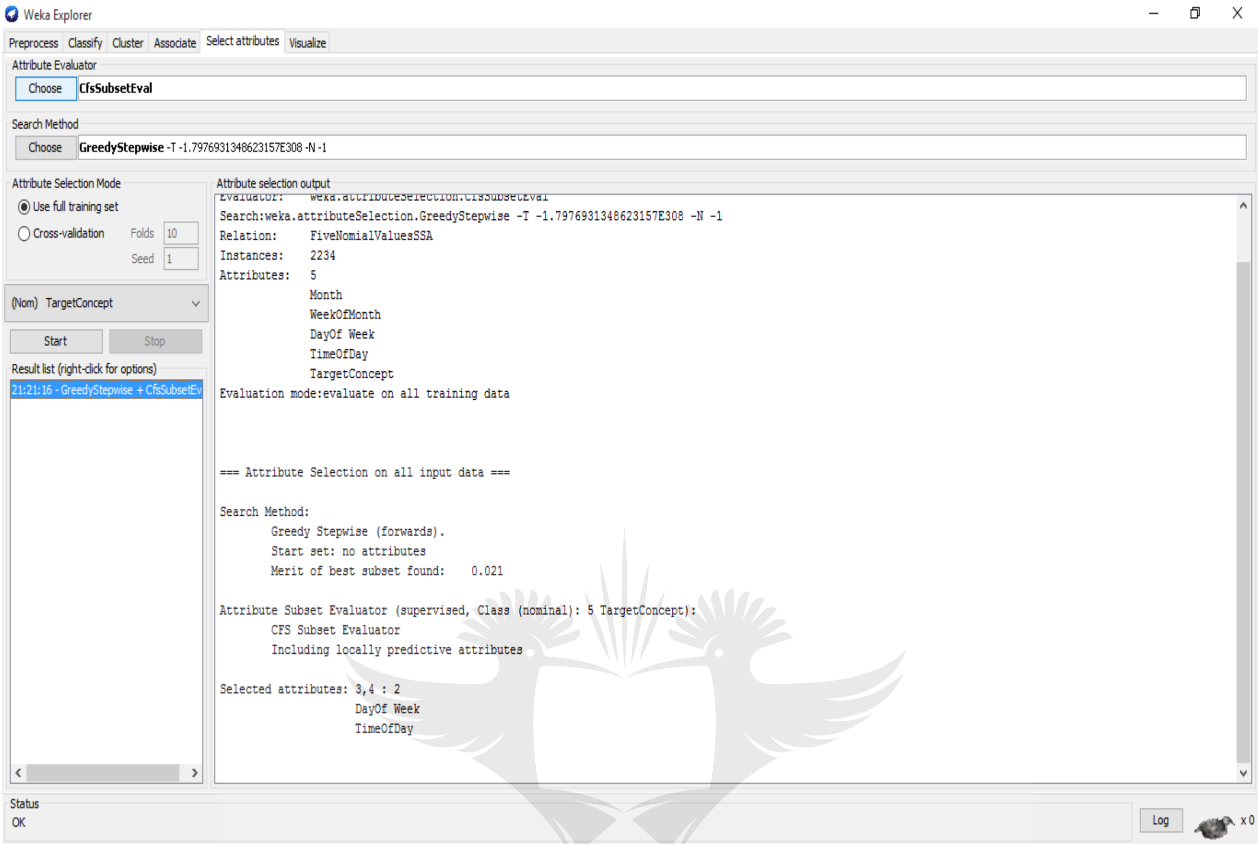


Figure 4.8: Results of search method when a combination of 2 best attributes are selected

Step 3: First use the best two sets of attributes combination in building the model. One can also use other different sets combination of the attributes.

=== Attribute Selection on all input data ===

Search Method:

Greedy Stepwise (forwards).

Start set: no attributes

Merit of best subset found: 0.021

"Attribute Subset Evaluator (supervised, Class (nominal)"(Witten, et al., 2011) : 5 TargetConcept):

"CFS Subset Evaluator"

"Including locally predictive attributes"

"Selected attributes": DayOfWeek

TimeOfDay

Figure 4.9: The 2 attributes that have been selected to be the best out of all the attributes

The procedure for attribute selection shown in Step 4 shows the attribute selection steps in detail.

Step 4: constructing the model

1. On Pre-process tab under "attribute" highlight "2 attributes" to be removed then click "remove" button.

2. You will be left with 2 attributes, DOW &TOD and the "Target Concept". On Classify tab, click the choose button under "Classifier", then select classification - J48 decision trees.

3. On the Test options panel, use "Cross-validation" option and 10 "Folds".

4. Right-click on the algorithm, "weak.gui.GenericObjectEditor" comes up to the screen. Set the value of "confidenceFactor" to 0.66 and leave the other field as is. Press the "Ok" button. Then click on the "start" button.

5. Right click on "result list" and "save model".

6. Repeat the same steps for all the remaining attributes until 6 models are constructed using the available attributes.

7. Evaluate all the results and then compare their confusion matrices and RMSE.

8. Repeat the same process for the other combinations of input attributes

The first execution of input attributes was DOW and TOD, then the subsequent input attributes were MON and DOW, WOM and DOW, MON and WOM, MON and TOD and TOD and WOM for model 1 to model 6, respectively.

Table 4.3: The J48 prediction and RMSE pruning using 0.25, 0.66 and 0.95 Confidence Factor Intervals and a combination of 2 attributes with different combinations

#	Algorithm	Input Attributes	Confidence Factor	Correct prediction instances (%)	RMSE
1	J48	DOW and TOD	0.25	64.593	0.392
			0.66	65.443	0.390
			0.95	65.443	0.390
2	J48	MON and DOW	0.25	62.399	0.398
			0.66	63.429	0.397
			0.95	63.429	0.397
3	J48	WOM and DOW	0.25	63.877	0.396
			0.66	63.474	0.397
			0.95	63.474	0.397

#	Algo rith m	Input Attributes	Confidence Factor	Correct prediction instances (%)	RMSE
4	J48	MON and WOM	0.25	61.549	0.402
			0.66	61.549	0.402
			0.95	61.549	0.402
5	J48	MON and TOD	0.25	61.549	0.402
			0.66	61.683	0.398
			0.95	61.683	0.398
6	J48	TOD and WOM	0.25	61.549	0.402
			0.66	61.773	0.398
			0.95	61.773	0.398

In Table 4.3, the decision tree (J48) algorithm produced the departure flights prediction performance results by pruning the 'confidenceFactor' to 0.25, 0.66 and 0.95. In Table 4.3, the results mean that, the 0.66 and 0.95 confidence factor produce the same prediction results and the same RMSE. The results, therefore, mean that the best results are found using either the two confidence factors. Table 4.3 shows the flight departures prediction performance of using different combinations of two attributes per model. The combination of DOW and TOD obtained the highest prediction results of 65.443% and had the RMSE of 0.390.

Figure 4.10 shows the pruning of the confidence factor for J48 tree to 0.66 on the experiments results in Table 4.2.

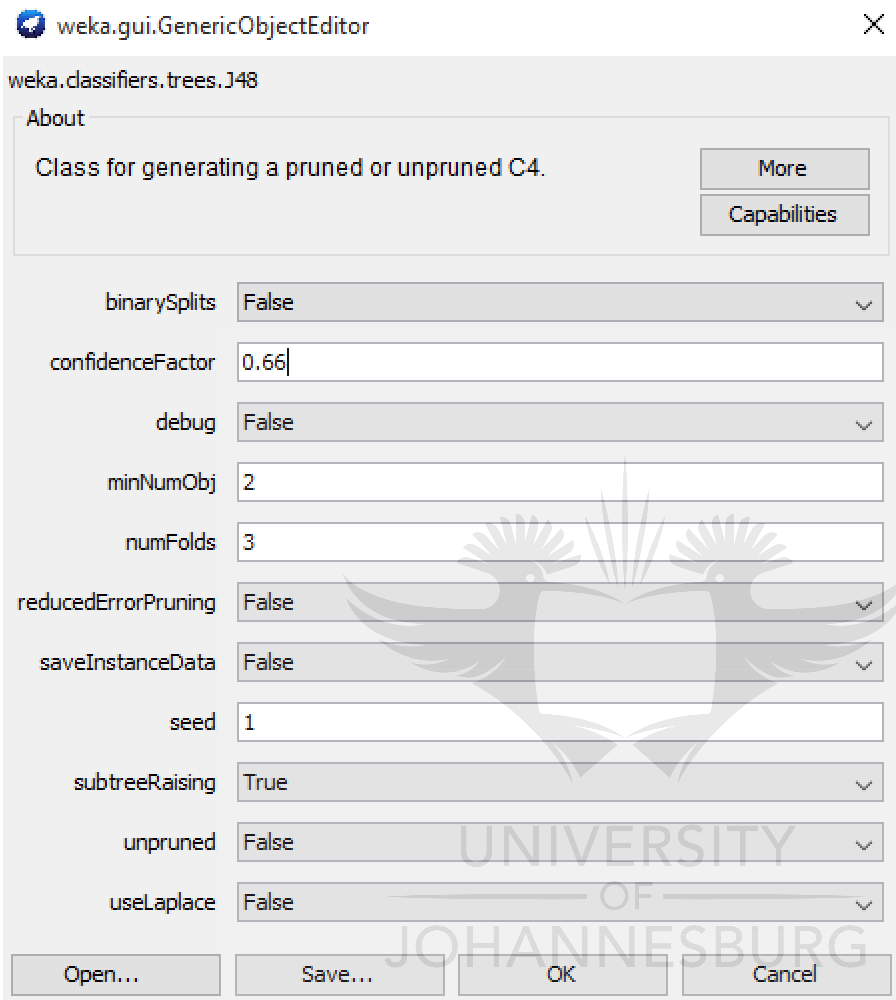


Figure 4.10: Pruning the tree using the confidence factor of 0.66

Table 4.4: The confusion matrix for a model constructed from DOW and TOD attribute

Actual	Prediction			
		a	**b**	**c**
	A	279	543	0
	B	192	1 183	0
	C	3	34	0

The rows in Table 4.4 represent the instances in an actual state of flight departures, and the columns represent the instances in a predicted target concept. The confusion matrix "Delayed", "On Time" and "Cancelled" are represented by a, b and c, respectively, and the same format will be used to present all the following confusion matrices. Table 4.4 shows that 279 instances were correctly predicted as "Delayed", 543 were incorrectly predicted as "On Time", 192 were incorrectly predicted as "Delayed", 1 183 were correctly predicted as "On Time", 3 instances were incorrectly predicted as "Delayed", and 34 instances were incorrectly predicted for "On Time".

4.3.1.2 Experiment 2: A model constructed using 3 attributes

The procedure for attribute selection where a combination of three attributes and an algorithm were used to construct the model is as follows:

Step 1: Load data

The procedure is the same as Experiment 1 (Step 1) covered in section 4.3.1.1.

Step 2: Attribute selection

1. Click on “Select attributes” tab as shown in Figure 4.11.

2. Under “Attribute Evaluator” frame click “choose” to select the relevant evaluation method e.g. “InfoGainAttributeEval”.

3. Under “Search Method” frame, click on the “choose” button to choose the relevant search method; click “Ranker”.

4. On Attribute Selection Mode click “Use training set”, Click on start button.

5. Output results set will be displayed, with attributes ranker to give you the ranked attributes in order of effectiveness.

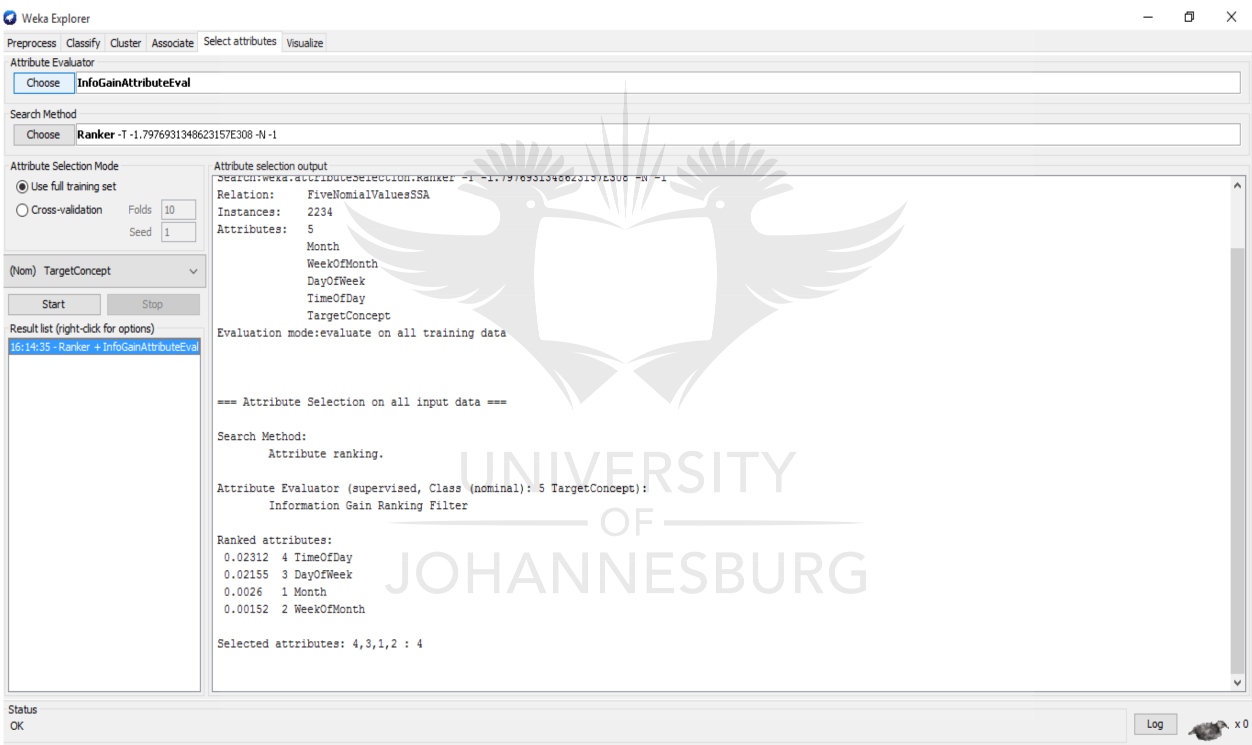


Figure 4.11: Four attributes when using “InfoGainAttributeEval” attribute evaluator and “Ranker” search method

The results in Figure 4.11 show the ranked attribute that will give good results when selected with a combination of other attributes. In Figure 4.11 the Ranked attributes 4, 3, 1, 2 : 4 results are displayed. The 4 after the colon means there were 4 attributes ranked that were sorted according to their evaluation. The 4 comma 3 comma 1 comma 2 means the fourth attribute ("TimeOfDay") was the highest ranked, the third attribute ("DayOfWeek") was ranked second, the first attribute ("Month") was ranked third and lastly, the second attribute ("WeekOfMonth") was the lowest ranked.

The results in Figure 4.11 mean that the four highly discriminative attributes are (TOD, DOW, MON and WOM, respectively.

Step 3: Attribute the selection using three attributes in model construction

In Figure 4.11, click on the pre-process tab, remove the lowest ranked attribute shown in Figure 4.11 by clicking on the check box of the attribute, then click on the "remove" button. This leaves you with three attributes and the target concept. Each time, the lowest ranked attribute is removed then, click on classify tab to select the classifier J48, click on choose on the populated drop-down menu, select the preferred algorithm J48, then click on "Start button".

1. On the "result list" right click and "save model"
2. click on Pre-process tab, click button "Undo" to have all the original attributes on the dataset.
3. Click on the second lowest ranked attribute, click start button and "Save model".
4. Click on the "pre-process tab" to repeat the same procedure as done in number (1, 2 and 3 above) to construct the model using three different attributes and the target concept.

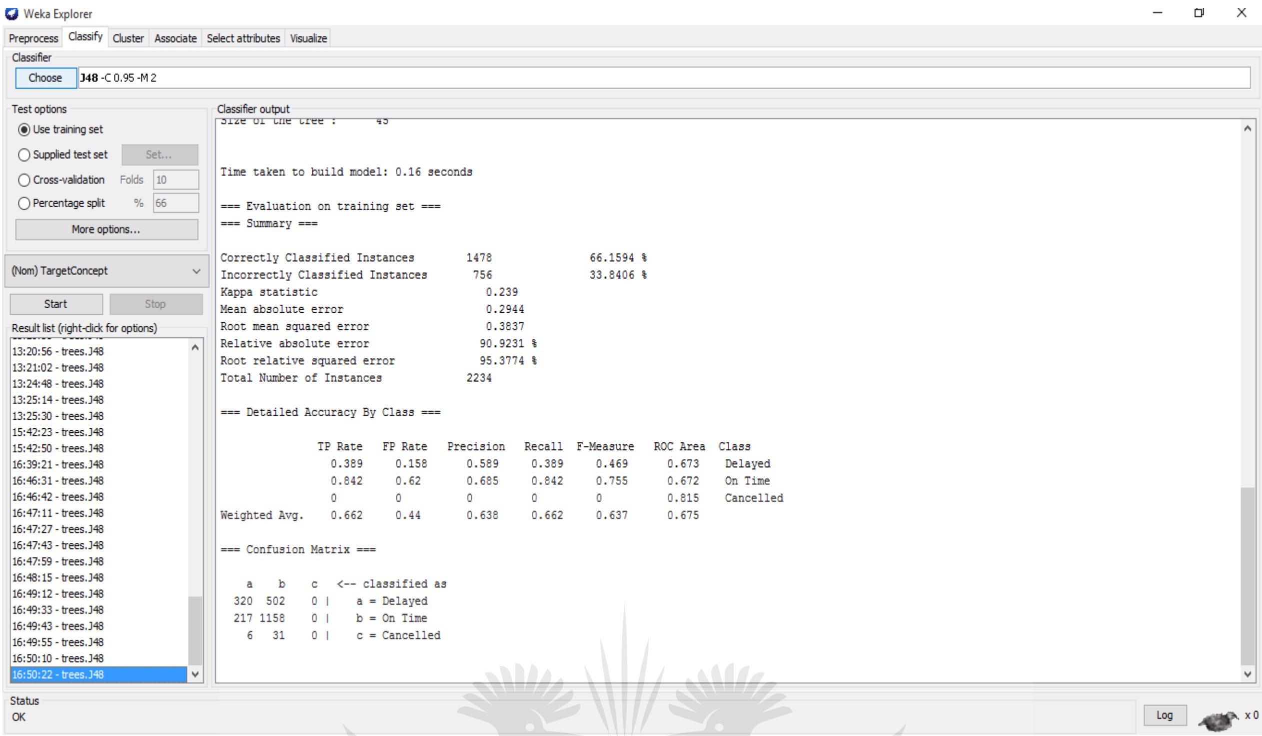


Figure 4.12: A model constructed from TOD, DOW and WOM attributes

Table 4.5: The results after creating a model using only 3 attributes per model

#	Algorithm	Attributes	Confidence Factor	Correct prediction instances (%)	RMSE Training
1	J48	TOD, DOW and WOM	0.95	66.159	0.384
2	J48	TOD, DOW and MON	0.95	66.070	0.385
3	J48	TOD, WOM and MON	0.95	61.952	0.396
4	J48	DOW, WOM and MON	0.95	64.951	0.392

The results in Table 4.5 mean that a combination of a set of attributes TOD, DOW and WOM has the lowest RMSE of 0.384 and the highest prediction performance of 66.159 %.

Table 4.6: Confusion matrix of J48 model constructed from TOD, DOW and WOM attributes

	Prediction			
		a	**b**	**c**
Actual	**A**	320	502	0
	B	217	1 158	0
	C	6	31	0

The values in Table 4.6 mean that 320 instances were correctly predicted as “Delayed” and 502 were incorrectly predicted as “On Time”, 1 158 instances were all correctly predicted as “On Time”, 217 instances were incorrectly predicted as “Delayed”, 6 instances were incorrectly predicted as “Delayed” and 51 instances were incorrectly predicted as “On Time”.

4.3.1.3 Experiment 3: A model constructed using 4 attributes per model

The procedure for attribute selection where a combination of three attributes per model was used, is as follows:

Step 1: Load data

The procedure is the same as Experiment 1 (Step 1) covered in section 4.3.1.1.

Step 2: Attribute selection

The procedure is the same as Experiment 2 (Step 1 and Step 2) in section 4.3.1.2.

Step 3: Use four attributes combinations in constructing the models

Using step 2 of Figure 4.11, you will use four attributes, click on classify tab to select the classifier J48, click on choose on the populated drop-down menu, select the preferred algorithm (J48) to use to construct the model, then click on the “Start button”.

1. On the “result list” right click and “save model”.
2. This result list also included the “Confusion Matrix”, “Detailed Accuracy by Class”, “Root mean square error” and the “Correctly classified instance”.

The results in Table 4.7 of the prediction performance and the RMSE of four attributes TOD, DOW, MON and WOM plus Class attribute of 67.144 % and 0.380, respectively.

Table 4.7: The results after creating a prediction model using all 4 attributes plus Class attribute

#	**Algorithm**	**Attributes**	**Correct prediction instances (%)**	**RMSE Training**
1	J48	DOW, TOD, WOM and MON	67.144	0.380
2	SVM	DOW, TOD, WOM and MON	66.249	0.474
3	K-Means	DOW, TOD, WOM and MON	61.549	N/A
4	MLP	DOW, TOD, WOM and MON	67.010	0.376

The results in Table 4.7 show the prediction values for the J48, SVM, MLP and K-Means models. The result shows that J48 is the best model, as it had a 67.144% prediction performance and an RMSE of 0.380. Four attributes (MON, WOM, DOW, TOD and Class attribute) were used during the construction of the models in Table 4.7.

Table 4.8: The Confusion matrix of J48 model constructed from the combination of 4 (TOD), (DOW), (MON) and (WOM) attributes

	Prediction			
Actual		**a**	**b**	**c**
	A	285	537	0
	B	160	1 215	0
	C	6	31	0

The values in Table 4.8 show that 285 instances were correctly predicted as “Delayed”, 537 were incorrectly predicted as “On Time”, 1 215 instances were all correctly predicted as “On Time”, 6 instances were incorrectly predicted as “Delayed”, and 31 instances were incorrectly predicted as “On Time”.

4.3.2 Experiments for training machine learning algorithms

In this section, experiments for training decision trees (J48), multi-layer perceptron (MLP), support vector machine (SVM) and K-Means clustering (K-Means) models are carried out.

Step 1: Load data

The procedure is the same as in Experiment 1 (Step 1) covered in section 4.3.1.1.

Step 2: Constructing the model

This set is for constructing prediction models using data and the MLP, SVM and K-means algorithms.

1. Click on Classify tab after loading dataset into WEKA.
2. On the Test option frame, click on the "choose" button and then select the classifier (algorithm) to be applied to the dataset (e.g. J48)
3. Specify location of the training set, in this case where *.csv file has been saved) – see Figure 4.13.
4. Click start button to create the model as shown on Figure 4.13.
5. On the result list frame, right click on the populated menu list and then click "Save model".

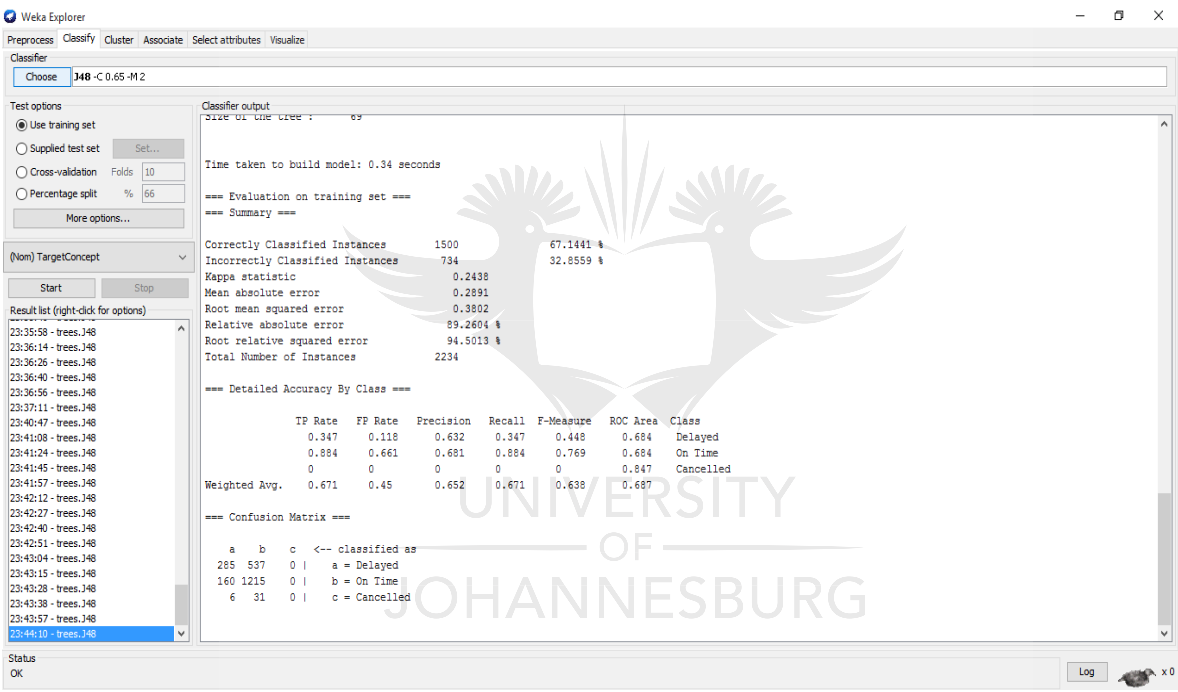


Figure 4.13: Training model using J48, showing the performance, confusion matrix and accuracy by class

4.3.2.1 Root Mean Square Error (RMSE)

RMSE is a "measure of the difference between locations that are known and locations that have been interpolated. RMSE is derived by squaring the difference between known and unknown points, adding those together, dividing that by the number of test points, and then taking the square root" (Chai & Draxler, 2014) of the results as shown in Equation 4.1. The RMSE of a prediction model is defined by equation 4.1 (Chai, et al., 2014).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(Predicted_i - Actual_i \right)^2}{N}}$$

Equation 4.1

The RMSE values "can be used to distinguish model performance in a calibration period with that of a validation period. It can also be used to compare the individual model performance to that of other predictive models".

The procedure for producing results in Table 4.8

The results in Table 4.8 were obtained by changing the value of Gamma (γ) to (0.01, 0.02, 0.03, 0.04 and 0.05). The gamma parameter "automatically defines the distance which a single training example can reach, with 'low' values meaning 'far' and 'high' values meaning 'close'" (Renukadevi & Thangaraj, 2013). Changing the Gamma (γ) parameter enables the study to find the predicted instance percentage.

Table 4.9: The SVM model constructed using 4 attributes that include the target concept for the July and August flight departure delays prediction performance

#	Algorithm	Gamma (γ)	Correct Predicted instances (%)	RMSE
1	SVM (Libsvm)	0.01	64.235	0.483
2	SVM (Libsvm)	0.02	65.130	0.482
3	SVM (Libsvm)	0.03	65.936	0.477
4	SVM (Libsvm)	0.04	66.067	0.476
5	SVM (Libsvm)	0.05	66.249	0.474

In Table 4.9, the SVM model obtained the best prediction results of 66.249 % and the lowest RMSE of 0.474 when Gamma was tuned to 0.05. The results mean 66.249% of the flights were correctly predicted as "On Time".

Procedure for producing results in Table 4.10

The results in Table 4.10 were obtained by changing the value of Gamma (γ) to (0.1, 0.2, 0.3, 0.4 and 0.5). Four different combinations of three attributes were used for constructing the prediction models.

Table 4.10: Shows the SVM model for flight departure delays using 3 attributes

#	Algorithm	Attributes	Gamma (γ)	Correct Predicted instances (%)	RMSE
1	SVM (Libsvm)	TOD, DOW and WOM	0.1	64.15	0.489
			0.2	64.100	0.489
			0.3	65.533	0.479
			0.4	65.533	0.479
			0.5	65.846	0.477
2	SVM (Libsvm)	TOD, DOW and MON	0.1	63.697	0.492
			0.2	64.593	0.486
			0.3	65.443	0.480
			0.4	65.443	0.480
			0.5	65.801	0.478
3	SVM (Libsvm)	TOD, WOM and MON	0.1	61.549	0.506
			0.2	61.549	0.506
			0.3	61.683	0.505
			0.4	61.773	0.505
			0.5	61.773	61.773
4	SVM (Libsvm)	DOW, WOM and MON	0.1	64.011	0.490
			0.2	64.011	0.490
			0.3	64.369	0.487
			0.4	64.951	0.48
			0.5	64.951	0.483

The results of the prediction performance and the RMSE in Table 4.10 show that this set of attributes TOD, DOW and WOM has the highest correctly predicted instances of 65.846% when the value of Gamma (γ) is set to 0.5 and an RMSE of 0.477.

Procedure for producing the below results in Table 4.11

The SVM algorithm was used to produce results in Table 4.11 by changing the value of Gamma (γ) to (0.1, 0.2, 0.3, 0.4 and 0.5). Six different combinations of 2 attributes were used for prediction.

The value of c-Cost = 0.1 was kept constant during the experiment while changing the Gamma parameter.

Table 4.11: Shows the SVM model for flight departure delays using 2 attributes

#	**Algorithm**	**Attributes**	**Gamma (γ)**	**Correct Predicted instances (%)**	**RMSE**
1	SVM (Libsvm)	TOD and DOW	0.1	63.787	0.491
			0.2	62.480	0.500
			0.3	65.197	0.482
			0.4	65.175	0.482
			0.5	65.175	0.482
2	SVM (Libsvm)	MON and DOW	0.1	62.936	0.497
			0.2	62.936	0.497
			0.3	62.936	0.497
			0.4	62.936	0.500
			0.5	63.429	0.494
3	SVM (Libsvm)	WOM and DOW	0.1	63.474	0.494
			0.2	63.474	0.494

#	Algorithm	Attributes	Gamma (γ)	Correct Predicted instances (%)	RMSE
			0.3	63.474	0.494
			0.4	63.474	0.494
			0.5	64.056	0.490
4	SVM (Libsvm)	WOM and MON	0.1	61.549	0.506
			0.2	61.549	0.506
			0.3	61.549	0.506
			0.4	61.549	0.506
			0.5	61.549	0.506
5	SVM (Libsvm)	MON and TOD	0.1	61.549	0.506
			0.2	61.549	0.506
			0.3	61.773	0.505
			0.4	61.773	0.505
			0.5	61.773	0.505
6	SVM (Libsvm)	TOD and WOM	0.1	61.549	0.506
			0.2	61.549	0.506
			0.3	61.728	0.505
			0.4	61.773	0.505
			0.5	61.773	0.505

In Table 4.11 are the prediction performance and the RMSE for attributes TOD and DOW, which gave the highest prediction performance 65.175% when the value of Gamma (γ) is set to 0.5 and an RMSE of 0.482.

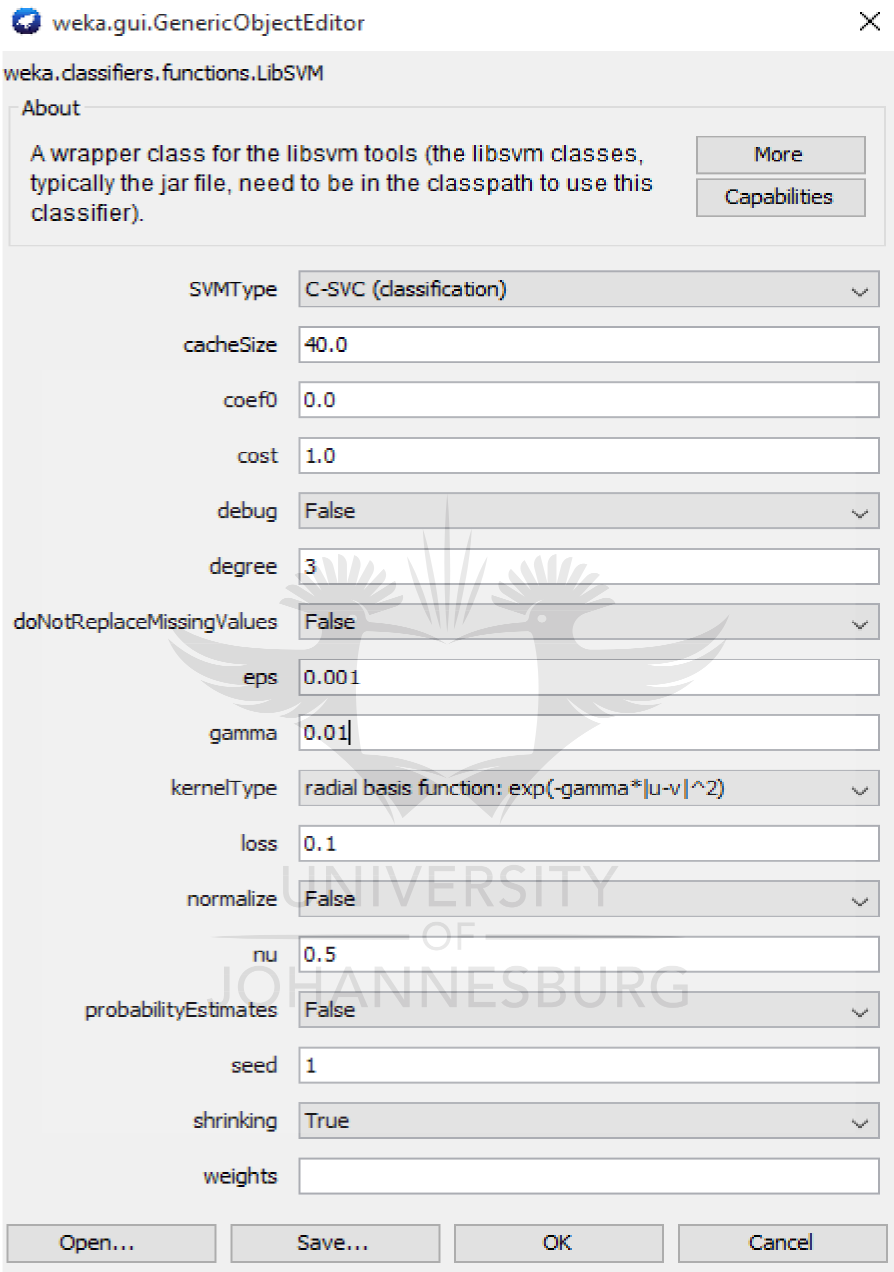


Figure 4.14: Tuning the gamma for SVM model is done

Table 4.12: The Confusion matrix for training data using the SVM algorithm for 2 attributes TOD and DOW

	Prediction			
		a	**b**	**c**
Actual	**A**	307	515	0
	B	226	1 149	0
	C	6	31	0

The results in Table 4.12 mean that the 307 instances were correctly predicted as "Delayed" and 515 instances were incorrectly predicted as "On Time", 1 149 instances were correctly predicted as "On Time", 226 instances were incorrectly predicted as "Delayed", 6 instances were incorrectly predicted as "Delayed", and 31 instances were incorrectly predicted as "On Time".

The previous sections have shown that although the prediction performance is good, it is not optimal. This has motivated the use of K-Means Clustering (K-Means) and Multi-Layered Perceptron (MLP) methods covered in section 4.3.3.

4.3.3 The clustering K-means method experiments

The K-means algorithm and the flight departures data were used to construct predictive models. All the attributes were used to determine the best clustering model. In Section 4.3.3, models were constructed using a combination of five, four, three and two attributes. These attributes would have been evaluated to determine which combination of attributes performs best. The K-means method enables simplicity and has speed, which allows it to run on large datasets.

4.3.3.1 Experiment 4: A model constructed using K-means method using 5 clusters (5 Attributes)

The procedure for designing the K-means method model I follows:

Step 1: loading data

The procedure is the same as in Experiment 1 (Step 1) covered in section 4.3.1.1

Step2: Using the K-Means method

1. Click 'Cluster' tab on the Explorer window after loading the dataset into WEKA, then click 'Cluster' tab.

2. In the 'Clusterer' box (Aksenova, 2004) "click on 'Choose' button. In pull-down menu select WEKA → Clusterers, and select the cluster scheme 'SimpleKMeans'".

Step 3: Setting the number of clusters

Once the clustering algorithm "is chosen, right-click on the algorithm, the GUI comes up on the screen" (Aksenova, 2004). Set the value in "numClusters" box to 5 because there are five clusters in the dataset. Leave the value of 'seed' as is.

Step 4: Steps followed using K-Means clustering

1. Choose 'Cluster mode'. "Click on 'Classes to cluster evaluation' radio-button in 'Cluster mode' box and select 'TargetConcept' in the pull-down box" (Aksenova, 2004).

2. Once the options have been specified, click on the 'Start' button to execute the algorithm.

3. When training is complete, "the 'Cluster' output area on the right panel of 'Cluster' window is filled with text describing the results of training and testing. A new entry appears in the 'Result list' box on the left of the result" (Aksenova, 2004).

4. Right-click the 'Result list' on the last line and select save model. Then give it a name "K-MeansModel".

Table 4.13: The Cluster Centroids for 4 Attributes using the K-Means algorithm

		Cluster Number				
Attribute	Full Data (2 234)	0 (921)	1 (622)	2 (248)	3 (200)	4 (243)
Month	August	August	July	August	July	August
WeekOfMonth	Month End	Month End	Month Beginning	Month Middle	Month End	Month Middle
DayOfWeek	Wednesday	Sunday	Monday	Friday	Wednesday	Tuesday
TimeOfDay	Morning	Morning	Morning	Morning	Morning	Evening

The cluster centroids have instances as shown in Table 4.13, which mean that cluster 0 has 41% of clustered instances with 921 instances in the month of August, during end of the month, on Sunday in the morning. Cluster 1 has 28% of clustered instances with 622 instances in the month of July, during beginning of the month on Monday in the morning. Cluster 2 has 11% of clustered instances with 248 instances in the month of August, during the middle of the month on Monday in the morning. Cluster 3 has the least number of instances (200) 9% of clustered instances with 200 instances in the month of July, during end of the month on Wednesday in the morning. Cluster 4 has 11% clustered instances with 243 instances in the month of August, during the middle of the month on Tuesday in the Evening. The incorrectly clustered instances are more that 62%, at 62.3545%.

4.3.3.2 Experiment 5: A model constructed using K-Means using 4 clusters (4 Attributes)

The procedure for designing the K-Means method model is as follows:

Step 1: loading data

The procedure is the same as in Experiment 1 (Step 1) covered in section 4.3.3.1

Step2: Using K-Means method

The procedure is the same as in Experiment 4 (Step 2) covered in section 4.3.3.1

Step 3: Setting the number of clusters

Once the clustering algorithm is chosen, "right-click on the algorithm, GUI comes up on the screen" (Aksenova, 2004). Set the value in "numClusters" box to 4 because we constructing a K-Means model using 4 attributes. Leave the value of 'seed' as is as shown in Figure 4.14.

Step 4: Proceedings in using K-means method

The procedure is the same as in Experiment 4 (Step 4) covered in section 4.3.3.1.

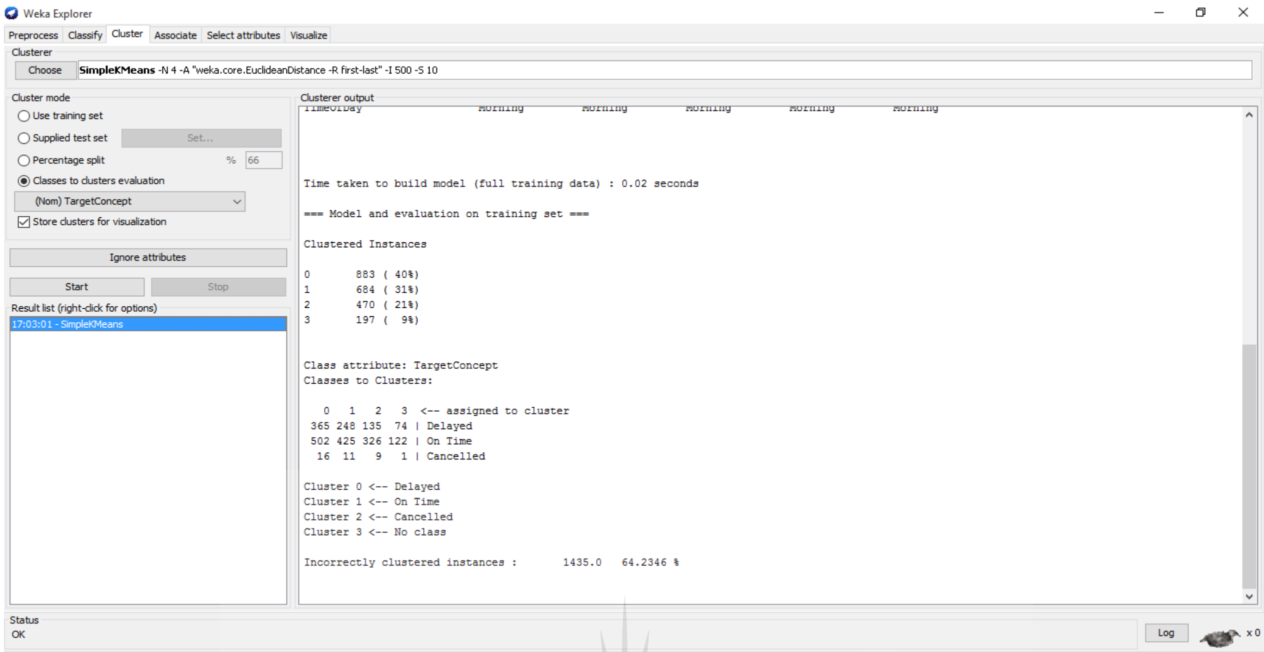


Figure 4.15: K-Means model constructed from dataset from 4 attributes

The procedure for producing the prediction model using three attributes and results are in Table 4.14. The results from a model constructed from a combination set of attributes TOD, DOW and WOM has the lowest RMSE of 0.384 and a predictive performance of 66.159 %. The K-Means model in Table 4.14 were obtained by changing the value of “Number Of Clusters”. Eight different combination of three attributes were used and their predictive performances were compared. K-Means algorithm does not output RMSE, hence there are no results for RMSE. The value of Number of Seeds was kept constant at = 5.

Table 4.14: The K-means model constructed from a combination of 3 attributes per model

#	Algorithm	Attributes	Number of Clusters	Number of Seed	Correctly Clustered instances (%)
1	K-Means	TOD, DOW and WOM	1	5	61.549
2	K-Means	TOD, DOW and WOM	2	5	60.340
3	K-Means	TOD, DOW and MON	1	5	61.549
4	K-Means	TOD, DOW and MON	2	5	60.340
5	K-Means	TOD, WOM and MON	1	5	61.549
6	K-Means	TOD, WOM and MON	2	5	54.655
7	K-Means	DOW, WOM and MON	1	5	61.549
8	K-Means	DOW, WOM and MON	2	5	60.340

Table 4.14 shows the clusters of a K-Means model constructed from the combination of (TOD), (DOW), (MON) and (WOM) attributes. The best prediction model for two attributes is with 1 cluster 0 and 5 seed with 61.549 %. The incorrectly predicted instances from K-Means clustering

was 38.4512%; hence, the correctly predicted instances are (100% – Incorrectly predicted Instances (38.4512 %)) = 61.549 % as shown in Figure 4.16.

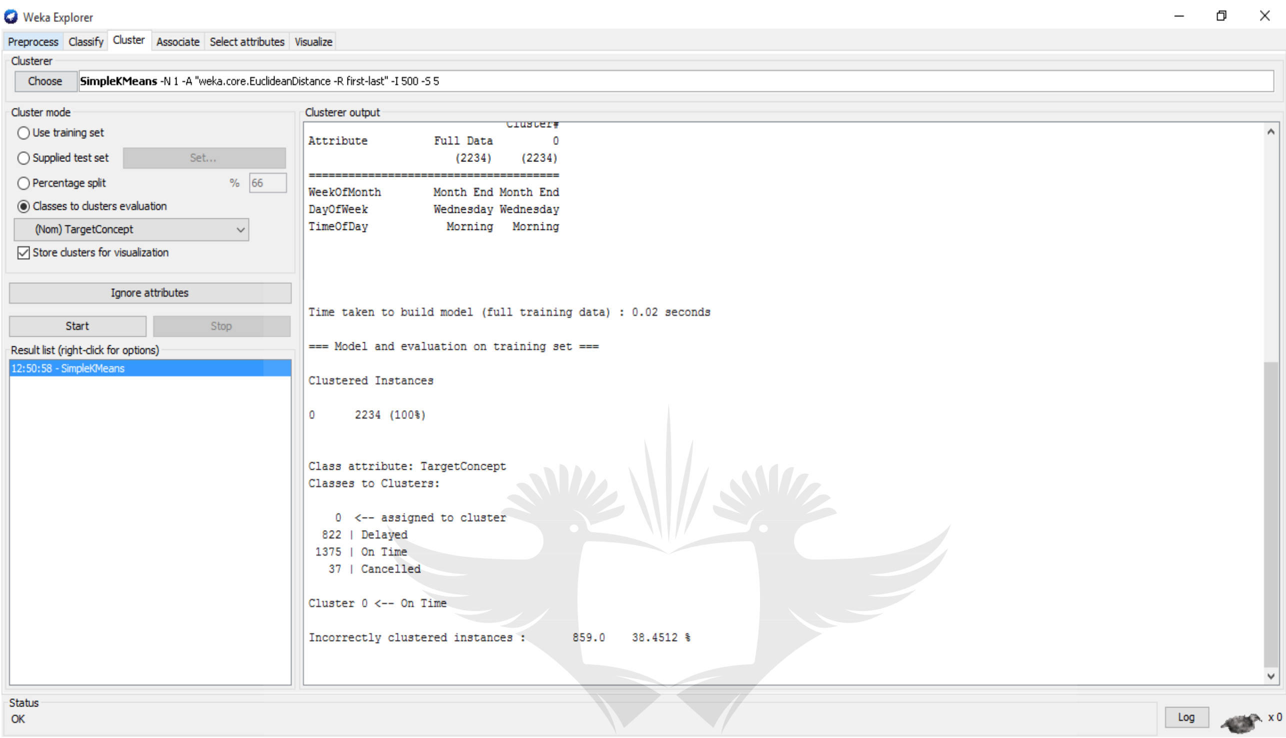


Figure 4.16: Shows the best K-Means model using 3 attributes per model

Procedure for creating a K-Means model using 2 attributes.

The K-Means model in Table 4.15 was obtained by changing the value of Number of Clusters. Eight different combinations of three attributes were used to compare the highly correct predicted instances. K-Means algorithm does not output RMSE; hence, there are no results for RMSE. The value of the Seed parameter was kept constant at 5.

Table 4.15: The results after creating a model using only 2 attributes per model

Number	**Algorithm**	**Attributes**	**Number of Clusters**	**Number of Seed**	**Correctly Clustered instances (%)**
1	K-Means	DOW and TOD	1	5	61.549
2	K-Means	DOW and TOD	2	5	56.535
3	K-Means	MON and DOW	1	5	61.549
4	K-Means	MON and DOW	2	5	55.864
5	K-Means	WOM and DOW	1	5	61.549
6	K-Means	WOM and DOW	2	5	54.253
7	K-Means	MON and WOM	1	5	61.549
8	K-Means	MON and WOM	2	5	53.581
9	K-Means	MON and TOD	1	5	61.549
10	K-Means	MON and TOD	2	5	54.834
11	K-Means	TOD and WOM	1	5	61.549
12	K-Means	TOD and WOM	2	5	61.728

Figure 4.17 shows cluster 0 with correctly clustered instances (100% – 38.4512%) at 61.549%. In Table 4.15 the attribute combination of TOD and WOM has the highest percentage of two number of clusters (cluster 2) with correctly clustered instances to (100% – 38.272%) at 61.728%.

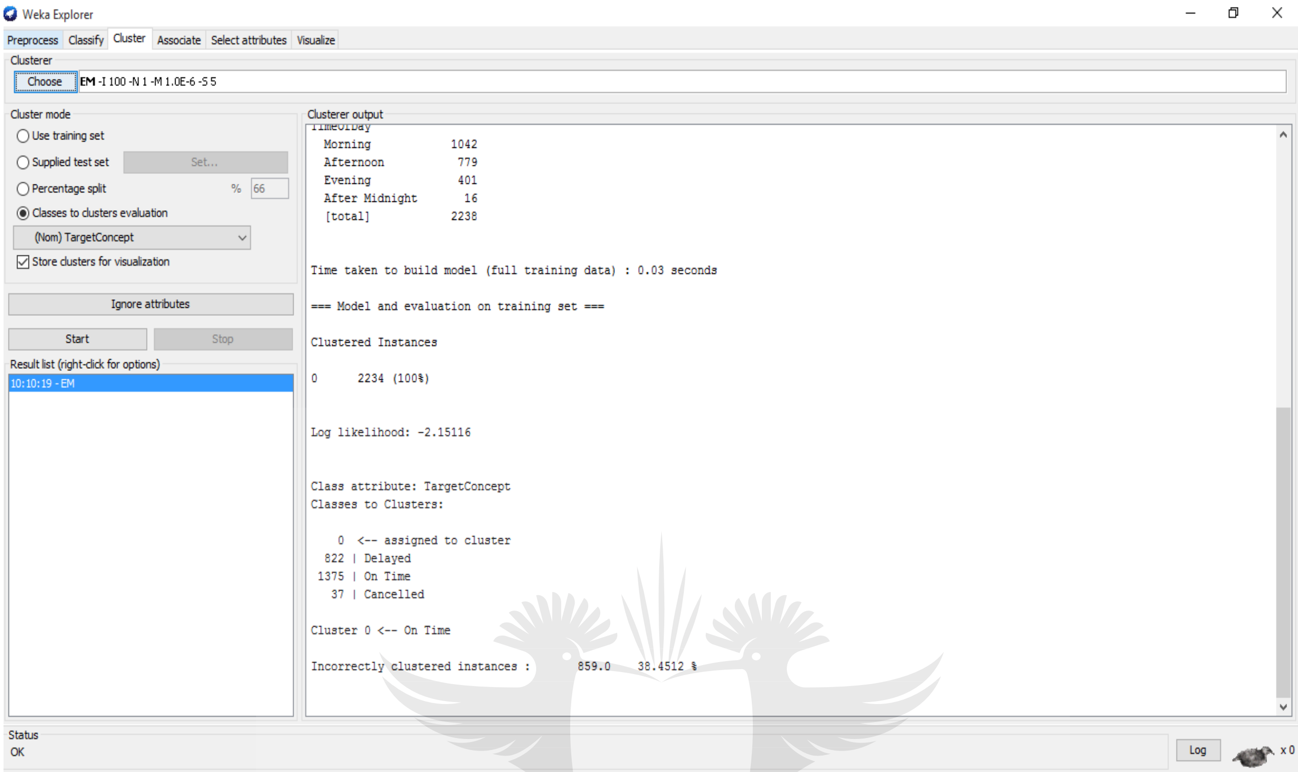


Figure 4.17: Shows K-Means single cluster results with TOD and WOM attributes used to construct the model

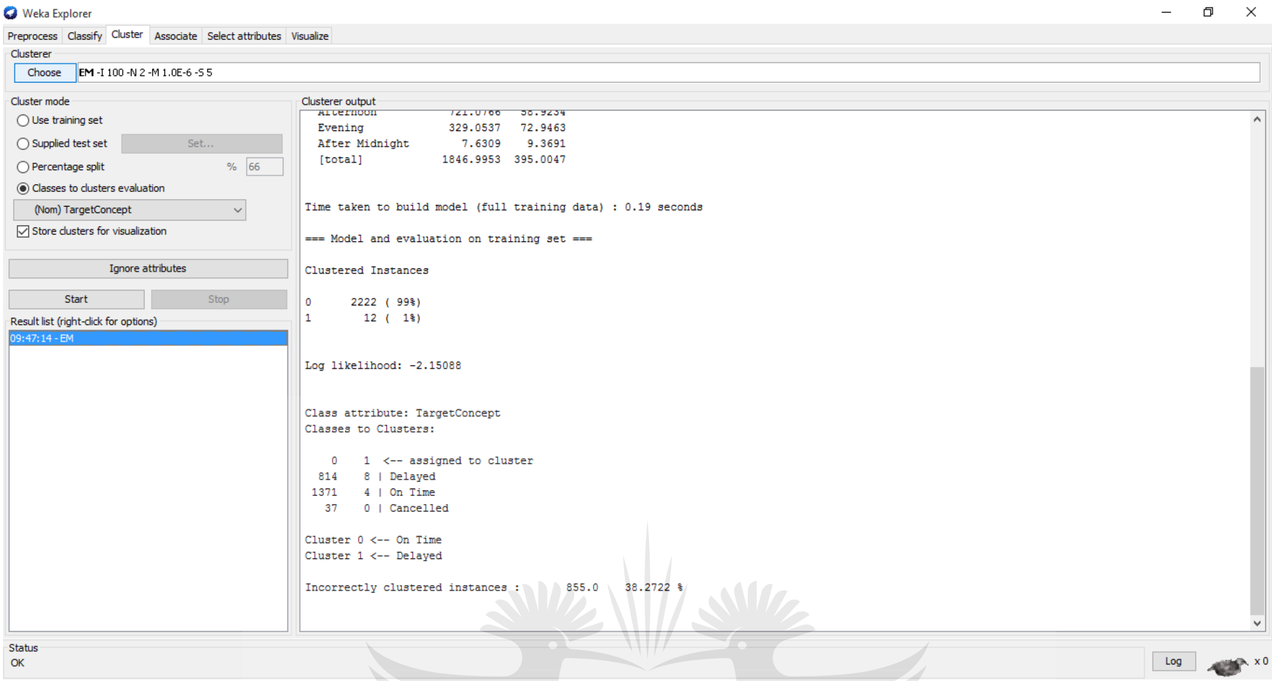


Figure 4.18: Shows K-Means cluster 2 results of TOD and WOM attributes model

4.3.3.3 Experiment 6: A model constructed using Multi-Layer Perceptron (MLP) for 4 attributes (MON, WOM, DOW, TOD and Class attribute)

The procedure for designing the MLP method model is as follows:

Step 1: Loading data

The procedure is the same as in Experiment 1 (Step 1) covered in section 4.3.1.1.

Step2: Data Classification using MLP method

1. Click on Classify tab after loading dataset into WEKA. On the Test option frame, click on "choose" button → expand functions and select the 'MultilayerPerceptron'.
2. Right-click on the algorithm, GUI comes on the screen. Set the value of "learningRate" to 0.1 and set the value of momentum to 0.0. Press the "Ok" button (Shown in Figure 4.19).
3. On the test options panel select "Use training set" radio-button.

4. Click on the ‘Start’ button to execute the algorithm.
5. “When the training set is complete, the ‘Classifier’ output area on the right panel of ‘Classifier’ window is filled with text describing the results of training and testing. A new entry appears in the ‘Result list’ box on the left of the result”.
6. Right-click the ‘Result list’ on the last line and select save model. Then give it a name “MLP Model” as shown in Figure 4.20.

Train the MLP involved varying one parameter at a time. These parameters included the learning rate, the momentum, the number of hidden layers, the number of inputs as shown in Figure 4.19 and cross-validation is varied as well. Only a combination of parameters that gave the best model are shown.

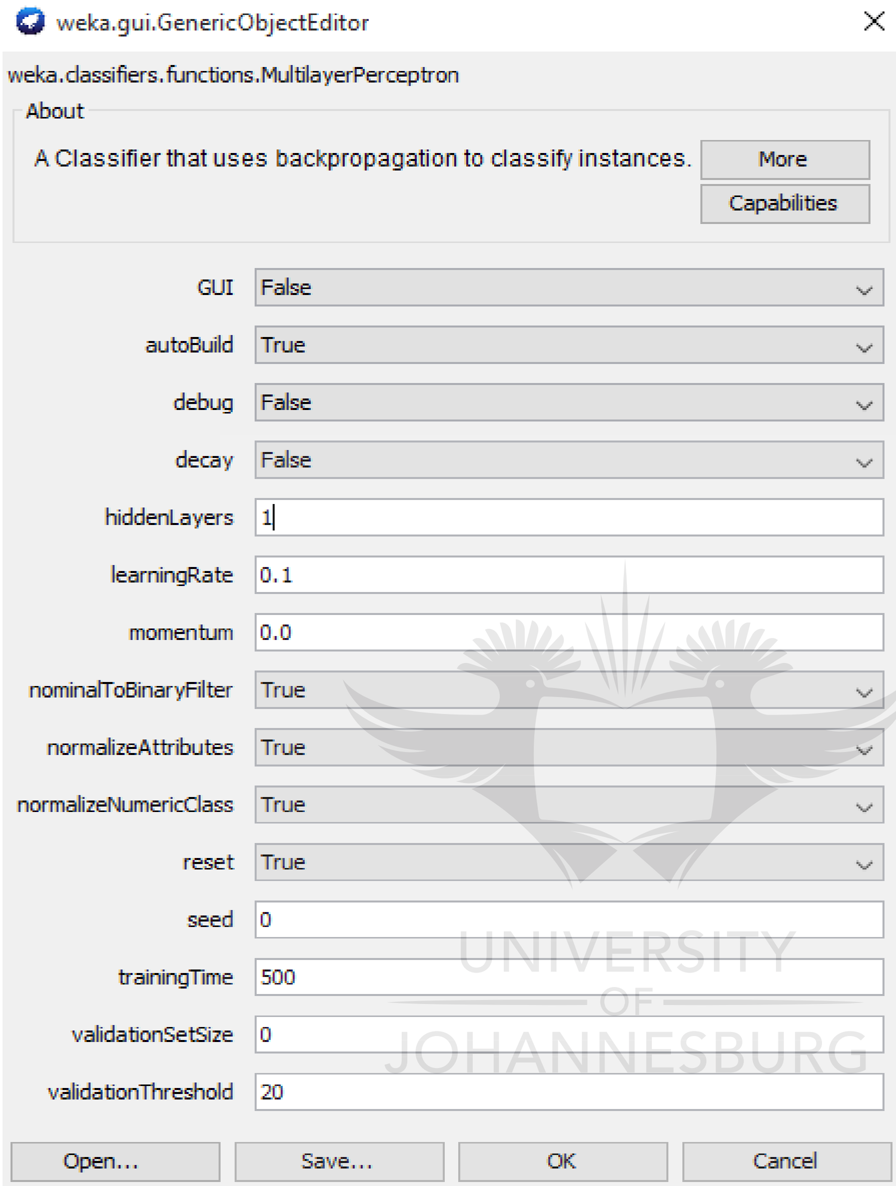


Figure 4.19: Weak.gui.GenericObjectEditor in tuning for MLP model

N.B η = Learning rate, α = Momentum

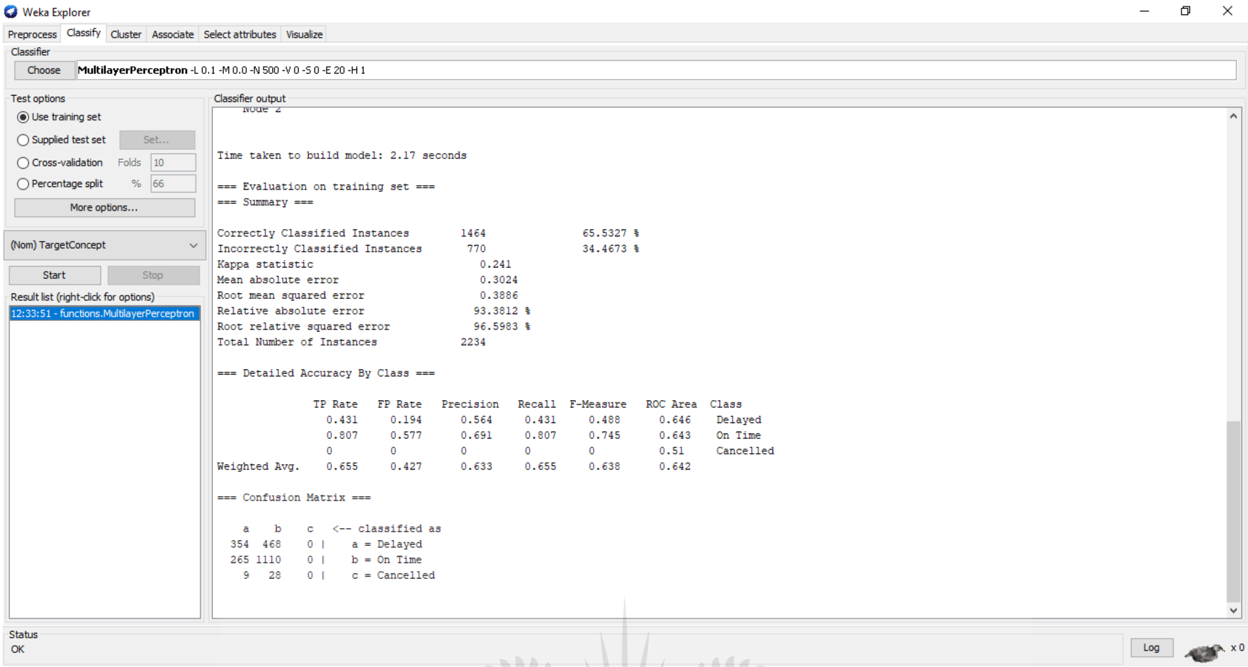


Figure 4.20: The MLP model shows the classification performance, confusion matrix and evaluation measures

In Figure 4.20 the prediction performance for MLP is 65.533 %.

Table 4.16: The Confusion matrix of the model constructed from 4 attributes TOD, DOW, MON and WOM

	Prediction			
		a	**b**	**c**
Actual	**A**	354	468	0
	B	265	1 110	0
	C	9	28	0

Table 4.16 shows the confusion matrix of MLP model constructed from the combination of 4 attributes (MON, WOM, DOW, TOD and Class attribute). The values of the confusion matrix show that 354 instances were correctly predicted as "Delayed", 468 were incorrectly predicted as "On Time", 265 instances were incorrectly predicted as "Delayed", 1 110 instances were correctly predicted as "On Time", 9 instances were incorrectly predicted as "Delayed", and 28 instances were incorrectly predicted as "On Time". There were no predicted instances for "Cancelled" class due to insufficient data.

Procedure creating an MLP model using 2 attributes.

1. All the experiments used "Use training set" Test options.
2. Using the MLP algorithm, 2 attributes were used. The Learning Rate (η) and Momentum (α) parameters were tuned. The values for the (η) used are to (0.1, 0.2, 0.3,0.4 or 0.5) and values for the (α) used are to (0.0, 0.1, 0.2, 0.3, 0.4 or 0.5).
3. Only 2 attributes and the target concept were used for experiments in Table 4.17.
4. The value of Hidden Layer put at 1 and was kept constant throughout this experiment.

5. The architecture of MLP was 2 inputs, 1 hidden layer and 3 outputs. The three classes namely" "Delayed", On-Time and "Cancelled.

Table 4.17: Shows each MLP model from 2 attributes and changing the (η) and (α)

Number	Algorithm	Learning Rate (η)	Momen tum (α)	Input (Attributes)	Hidden Layer	Outputs	Correct Prediction Instances (%)	RMSE
1	MLP	0.1	0.0	DOW and TOD	1	3	65.040	0.390
		0.2	0.0	DOW and TOD	1	3	65.085	0.391
		0.3	0.0	DOW and TOD	1	3	65.085	0.391
		0.4	0.0	DOW and TOD	1	3	62.534	0.391
		0.5	0.0	DOW and TOD	1	3	62.534	0.392
2	MLP	0.1	0.1	DOW and TOD	1	3	65.040	0.390
		0.2	0.1	DOW and TOD	1	3	65.085	0.391
		0.3	0.1	DOW and TOD	1	3	62.534	0.391
		0.4	0.1	DOW and TOD	1	3	62.534	0.391
		0.5	0.1	DOW and TOD	1	3	62.534	0.391
3	MLP	0.1	0.2	DOW and TOD	1	3	65.040	0.390
		0.2	0.2	DOW and TOD	1	3	65.085	0.391
		0.3	0.2	DOW and TOD	1	3	62.534	0.391
		0.4	0.2	DOW and TOD	1	3	62.534	0.392
		0.5	0.2	DOW and TOD	1	3	63.877	0.391

4	MLP	0.1	0.3	DOW and TOD	1	3	65.040	0.390
		0.2	0.3	DOW and TOD	1	3	65.085	0.391
		0.3	0.3	DOW and TOD	1	3	62.534	0.392
		0.4	0.3	DOW and TOD	1	3	62.534	0.393
		0.5	0.3	DOW and TOD	1	3	63.877	0.392
5	MLP	0.1	0.4	DOW and TOD	1	3	65.040	0.391
		0.2	0.4	DOW and TOD	1	3	62.534	0.391
		0.3	0.4	DOW and TOD	1	3	62.534	0.392
		0.4	0.4	DOW and TOD	1	3	63.877	0.391
		0.5	0.4	DOW and TOD	1	3	60.340	0.397
6	MLP	0.1	0.5	DOW and TOD	1	3	65.085	0.391
		0.2	0.5	DOW and TOD	1	3	62.534	0.392
		0.3	0.5	DOW and TOD	1	3	62.534	0.393
		0.4	0.5	DOW and TOD	1	3	60.340	0.397
		0.5	0.5	DOW and TOD	1	3	60.340	0.399

Results in Table 4.17 shows the MLP model prediction model constructed from two attributes while changing the (η) term and the (α) term. The prediction performance of this MLP model and the RMSE show that the combination of attributes DOW and TOD when η= 0.1; α= 0.0 or η= 0.1; α= 0.1 or η= 0.1; α= 0.2 or η= 0.1; α= 0.3 has obtained prediction performance of 65.040% and the lowest RMSE of 0.390.

Table 4.18: The Confusion matrix of the model constructed from 2 attributes η= 0.1; α = 0.0

	Prediction			
		a	**b**	**c**
Actual	**A**	308	514	0
	B	230	1 145	0
	C	6	31	0

Results in Table 4.18 mean that 308 instances were correctly predicted as “Delayed” and 514 were incorrectly predicted as “On Time”, 1 145 instances were correctly predicted as “On Time” and 230 instances were incorrectly predicted as “Delayed”, 6 instances were incorrectly predicted as “Delayed” and 31 instances were incorrectly predicted as “On Time”. There were not instances predicted for “Cancelled” class due to insufficient data.

Procedure for creating an MLP model using 2 attributes

1. All the experiments below used “Use training set” Test options.
2. Using the MLP algorithm, the two attributes were used.
3. The (η) and the (α) parameters were tuned. The values for (η) were tuned to (0.1, 0.2, 0.3,0.4 or 0.5) and values for the (α) were pruned to (0.0, 0.1, 0.2, 0.3, 0.4 or 0.5).
4. Only two attributes were used as in the experiments in Table 4.19. The value of Hidden Layer was put at 1 and was kept constant throughout this experiment.
5. The architecture of the MLP is 2 : 1 : 3. The three outputs are the three classes namely “Delayed”, On-Time and “Cancelled”.

Table 4.19: Shows MLP models constructed using 2 attributes and changing the (η) and (α)

#	Algorithm	Input Attributes	Learning Rate (η)	Momentum (α)	Hidden Layer	Outputs	Correct Prediction Instances (%)	RMSE
1	MLP	DOW and TOD	0.1	0.0	1	3	65.040	0.390
			0.2	0.0	1	3	65.085	0.391
			0.3	0.0	1	3	65.085	0.391
			0.4	0.0	1	3	62.534	0.391
			0.5	0.0	1	3	62.534	0.392
2	MLP	DOW and TOD	0.1	0.1	1	3	65.040	0.390
			0.2	0.1	1	3	65.085	0.391
			0.3	0.1	1	3	62.534	0.391
			0.4	0.1	1	3	62.534	0.391
			0.5	0.1	1	3	62.534	0.391
3	MLP	DOW and TOD	0.1	0.2	1	3	65.040	0.390
			0.2	0.2	1	3	65.085	0.391
			0.3	0.2	1	3	62.534	0.391
			0.4	0.2	1	3	62.534	0.392
			0.5	0.2	1	3	63.877	0.391
4	MLP	DOW and TOD	0.1	0.3	1	3	65.040	0.390
			0.2	0.3	1	3	65.085	0.391
			0.3	0.3	1	3	62.534	0.392

			0.4	0.3	1	3	62.534	0.393
			0.5	0.3	1	3	63.877	0.392
5	MLP	DOW and TOD	0.1	0.4	1	3	65.040	0.391
			0.2	0.4	1	3	62.534	0.391
			0.3	0.4	1	3	62.534	0.392
			0.4	0.4	1	3	63.877	0.391
			0.5	0.4	1	3	60.340	0.397

Table 4.19 shows the MLP model prediction model constructed from two attributes while changing the (η) term and the (α) term. The prediction performance of this MLP model and the RMSE show that the combination of attributes DOW and TOD when η= 0.2; α= 0.0 or η= 0.2; α= 0.1 or η= 0.2; α= 0.2 0r η= 0.2; α= 0.3 obtained prediction performance of 65.085% and RMSE of 0.391.

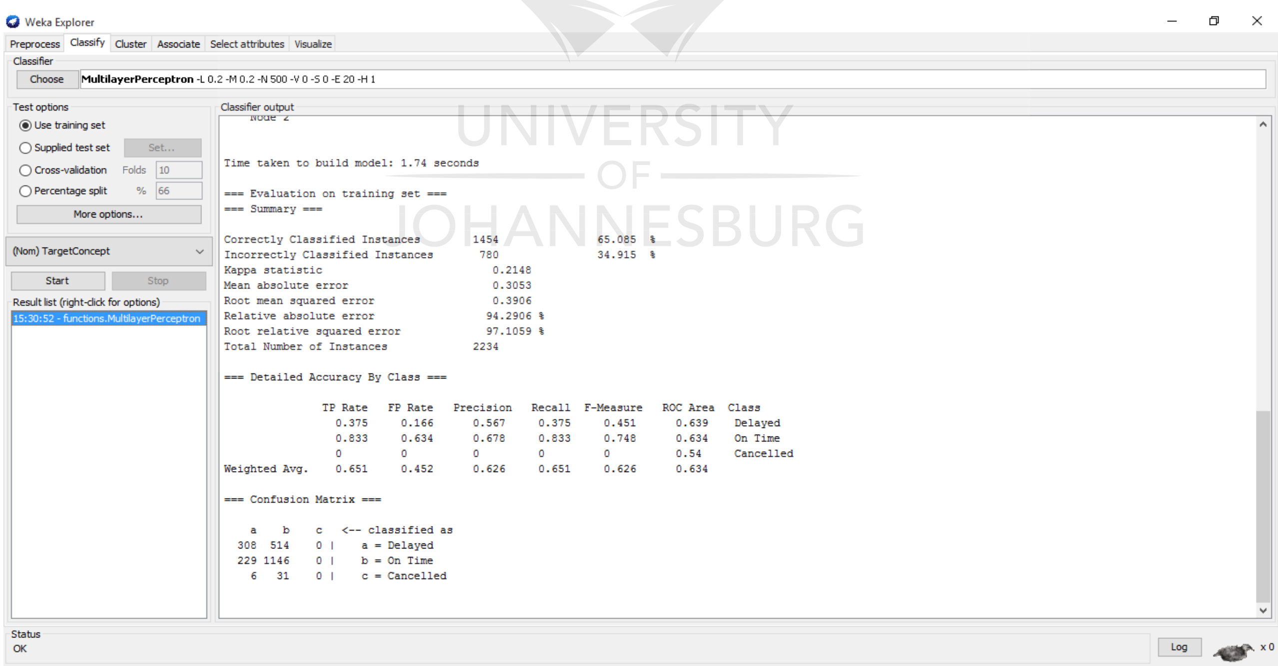


Figure 4.21: Shows the best MLP Model using 2 attributes per η= 0.2; α = 0.2

Table 4.20: The Confusion matrix of the best MLP Model using 2 attributes η= 0.2; α = 0.2

	Prediction			
		a	**b**	**c**
Actual	**A**	308	514	0
	B	229	1 146	0
	C	6	31	0

The results in Table 4.20 mean that 308 instances were correctly predicted as "Delayed" and 514 were incorrectly predicted as "On Time", 1 146 instances were correctly predicted as "On Time" and 229 instances were incorrectly predicted as "Delayed", 6 instances were incorrectly predicted as "Delayed", when they were "Cancelled", and 31 instances were incorrectly predicted as "On Time".

Procedure for creating an MLP model using 3 attributes.

1. All the experiments that follow used "Use training set" Test options.
2. Using the MLP algorithm, all the attributes were used. The Learning Rate (η) and Momentum (α) parameters were tuned. The values for n were tuned to (0.1, 0.2, 0.3,0.4 or 0.5) and values for m were tuned to (0.0, 0.1, 0.2, 0.3, 0.4 or 0.5).
3. Only 3 attributes and the target concept were used for the experiments in Table 4.21.
4. The value of Hidden Layer was put at 1 and was kept constant throughout this experiment.
5. The Outputs on results in Table 4.21 are the three classes namely "Delayed", On-Time and "Cancelled".

Table 4.21: Shows MLP models constructed from 3 attributes and changing (η) and (α)

#	Algorithm	Input (Attributes)	Learning Rate (η)	Momentum (α)	Hidden Layer	Outputs	Correct Predicted instances (%)	RMSE
1	MLP	TOD, DOW and WOM	0.1	0.0	1	3	64.593	0.389
			0.2	0.0	1	3	64.235	0.389
			0.3	0.0	1	3	63.921	0.389
			0.4	0.0	1	3	63.429	0.390
			0.5	0.0	1	3	63.250	0.391
2	MLP	TOD, DOW and WOM	0.1	0.1	1	3	64.593	0.389
			0.2	0.1	1	3	64.235	0.389
			0.3	0.1	1	3	63.429	0.390
			0.4	0.1	1	3	63.250	0.391
			0.5	0.1	1	3	63.250	0.391
3	MLP	TOD, DOW and WOM	0.1	0.2	1	3	64.235	0.389
			0.2	0.2	1	3	64.235	0.389

			0.3	0.2	1	3	63.250	0.390
			0.4	0.2	1	3	63.250	0.391
			0.5	0.2	1	3	63.429	0.392
4	MLP	TOD, DOW and WOM	0.1	0.3	1	3	64.235	0.389
			0.2	0.3	1	3	63.921	0.389
			0.3	0.3	1	3	63.250	0.391
			0.4	0.3	1	3	63.250	0.392
			0.5	0.3	1	3	63.429	0.393
5	MLP	TOD, DOW and WOM	0.1	0.4	1	3	64.235	0.389
			0.2	0.4	1	3	63.921	0.389
			0.3	0.4	1	3	63.249	0.391
			0.4	0.4	1	3	63.474	0.392
			0.5	0.4	1	3	63.071	0.393
6		TOD, DOW and WOM	0.1	0.5	1	3	64.235	0.389
			0.2	0.5	1	3	63.249	0.391
			0.3	0.5	1	3	63.429	0.392
			0.4	0.5	1	3	63.071	0.393

			0.5	0.5	1	3	63.518	0.395

Table 4.21 shows the results when three attributes per model were used for the prediction model while changing the learning rate and the momentum. The highest prediction performance and the RMSE for these attributes TOD, DOW and WOM when η= 0.1; α=0.0 or η= 0.1; α=0.1 has the highest correctly predicted instance of 64.593% and the best RMSE of 0.389.

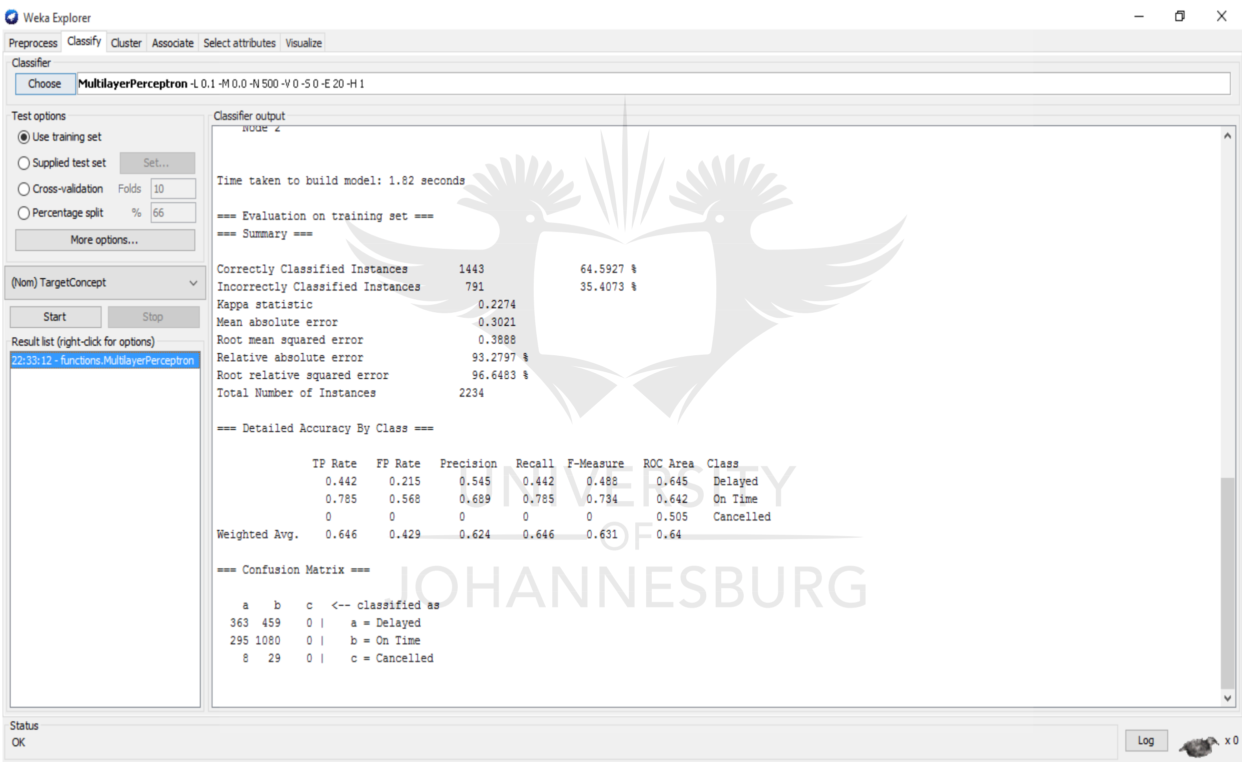


Figure 4.22: Shows the best MLP Models created using 3 attributes

Procedure for creating an MLP model using 4 attributes.

1. All the experiments used “Use training set” Test options.
2. Using the MLP algorithm, all the attributes were used. The (η) and (α) parameters were tuned to (0.1, 0.2, 0.3,0.4 or 0.5) and the values to (0.0, 0.1, 0.2, 0.3, 0.4 or 0.5) respectively.

3. Four attributes (MON, WOM, DOW, TOD and Class attribute**)** were used for the experiments in Table 4.22. The value of Hidden Layer was put at 1 and was kept constant throughout this experiment.
4. The Outputs in Table 4.22 represent the three classes namely “Delayed”, On-Time and “Cancelled”.

Table 4.22: Shows the MLP model created using 4 attributes and changing (η) and (α)

#	Algorithm	Attributes (Inputs)	Learning Rate (η)	Momentum (α)	Hidden Layer	Outputs	Correct Predicted instance (%)	RMSE
1	MLP	TOD, DOW, WOM and MON	0.1	0.0	1	3	65.533	0.389
			0.2	0.0	1	3	64.727	0.389
			0.3	0.0	1	3	64.369	0.390
			0.4	0.0	1	3	64.369	0.390
			0.5	0.0	1	3	65.309	0.391
2	MLP	TOD, DOW, WOM and MON	0.1	0.1	1	3	64.951	0.389
			0.2	0.1	1	3	64.503	0.390
			0.3	0.1	1	3	64.369	0.390
			0.4	0.1	1	3	65.309	0.390
			0.5	0.1	1	3	65.309	0.391
3	MLP		0.1	0.2	1	3	64.951	0.389

		TOD, DOW, WOM and MON	0.2	0.2	1	3	64.369	0.390
			0.3	0.2	1	3	64.369	0.390
			0.4	0.2	1	3	65.309	0.391
			0.5	0.2	1	3	64.861	0.392
4	MLP	TOD, DOW, WOM and MON	0.1	0.3	1	3	64.817	0.389
			0.2	0.3	1	3	64.369	0.390
			0.3	0.3	1	3	65.309	0.391
			0.4	0.3	1	3	65.309	0.392
			0.5	0.3	1	3	64.861	0.393
5	MLP	TOD, DOW, WOM and MON	0.1	0.4	1	3	64.817	0.389
			0.2	0.4	1	3	64.369	0.390
			0.3	0.4	1	3	65.309	0.391
			0.4	0.4	1	3	64.861	0.393
			0.5	0.4	1	3	64.861	0.395
6	MLP	TOD, DOW, WOM and MON	0.1	0.5	1	3	64.727	0.389
			0.2	0.5	1	3	65.309	0.391
			0.3	0.5	1	3	64.861	0.393

			0.4	0.5	1	3	64.861	0.396
			0.5	0.5	1	3	57.431	0.397

Table 4.22 shows the results when four attributes were used for constructing the prediction model while changing the learning rate and the momentum parameters. The best performance using attributes TOD, DOW, WOM and MON are $\eta = 0.1$; $\alpha = 0.0$, prediction performance of 65.533% and the best lowest RMSE of 0.389.

Table 4.23: The Confusion matrix of the best MLP Model constructed from the combination of (TOD), (DOW) and (WOM) attributes

		Prediction		
		a	**b**	**c**
Actual	**A**	363	459	0
	B	295	1 080	0
	C	8	29	0

The results in Table 4.23 mean that 363 instances were correctly predicted as "Delayed" and 459 were incorrectly predicted as "On Time", 1 080 instances were all correctly predicted as "On Time" and 295 were incorrectly predicted as "Delayed", 8 instances were incorrectly predicted as "Delayed" and 29 instances were incorrectly predicted as "On Time".

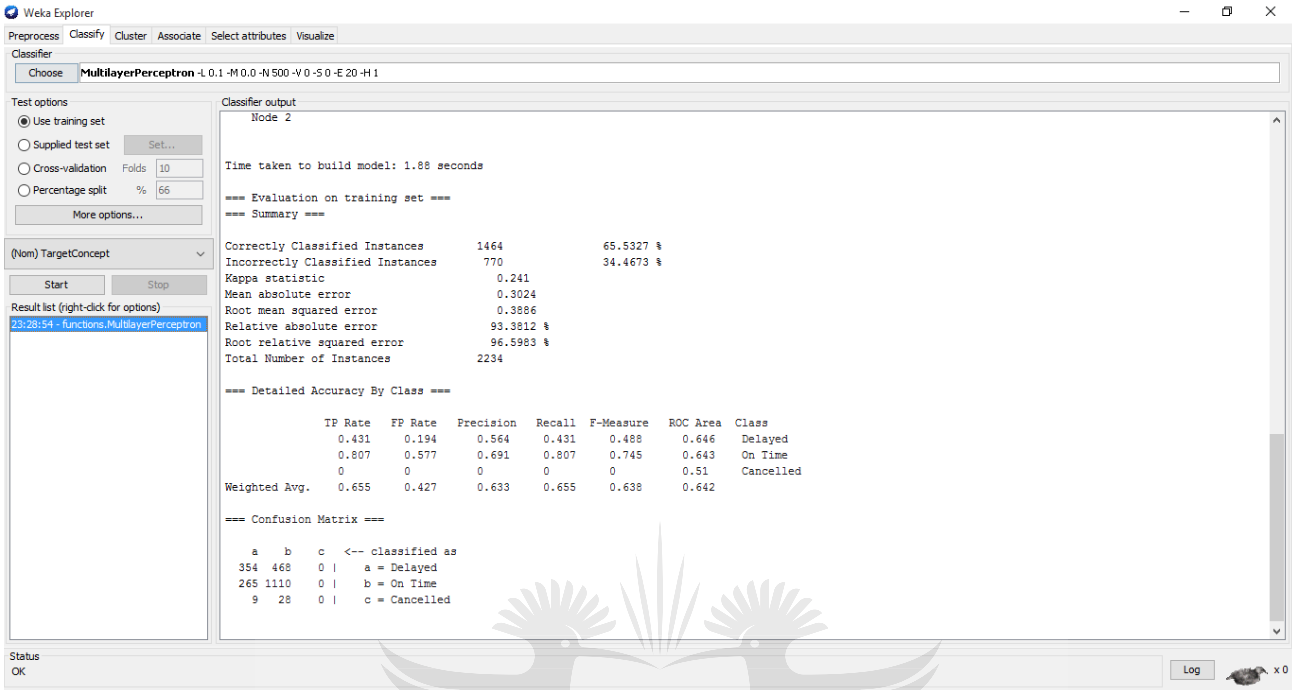


Figure 4.23: Shows the best MLP model created using 4 attributes

Table 4.24: The Confusion matrix of the best MLP Model constructed from the combination of (TOD), (DOW), (MON) and (WOM) attributes

	Prediction			
Actual		**a**	**b**	**c**
	A	354	468	0
	B	265	1 110	0
	C	9	28	0

The results in Table 4.24 mean that 354 instances were correctly predicted as “Delayed” and 468 were incorrectly predicted as “On Time”, 1 110 instances were correctly predicted as “On Time”, 265 were incorrectly predicted as “Delayed”, 9 instances were incorrectly predicted as “Delayed” and 28 instances were incorrectly predicted a “On Time”.

4.3.4 Summary of the prediction performance and the RMSE results

This section of the study plots the prediction performance and RMSE results from section 4.3 in Table 4.3 using the ‘confidenceFactor’ of 0.25, 0.66 and 0.95.

Figures 4.24 and 4.25 show the results where attributes selection (combination of two attributes per model) were used to determine the best performing model. The flight departures data were used to construct the Decision Tree (J48) model. The attributes ((DOW & TOD), (MON & DOW), (WOM & DOW), (MON & WOM), (MON & TOD) & (TOD & WOM)) were selected using the attributes selection option on WEKA software.

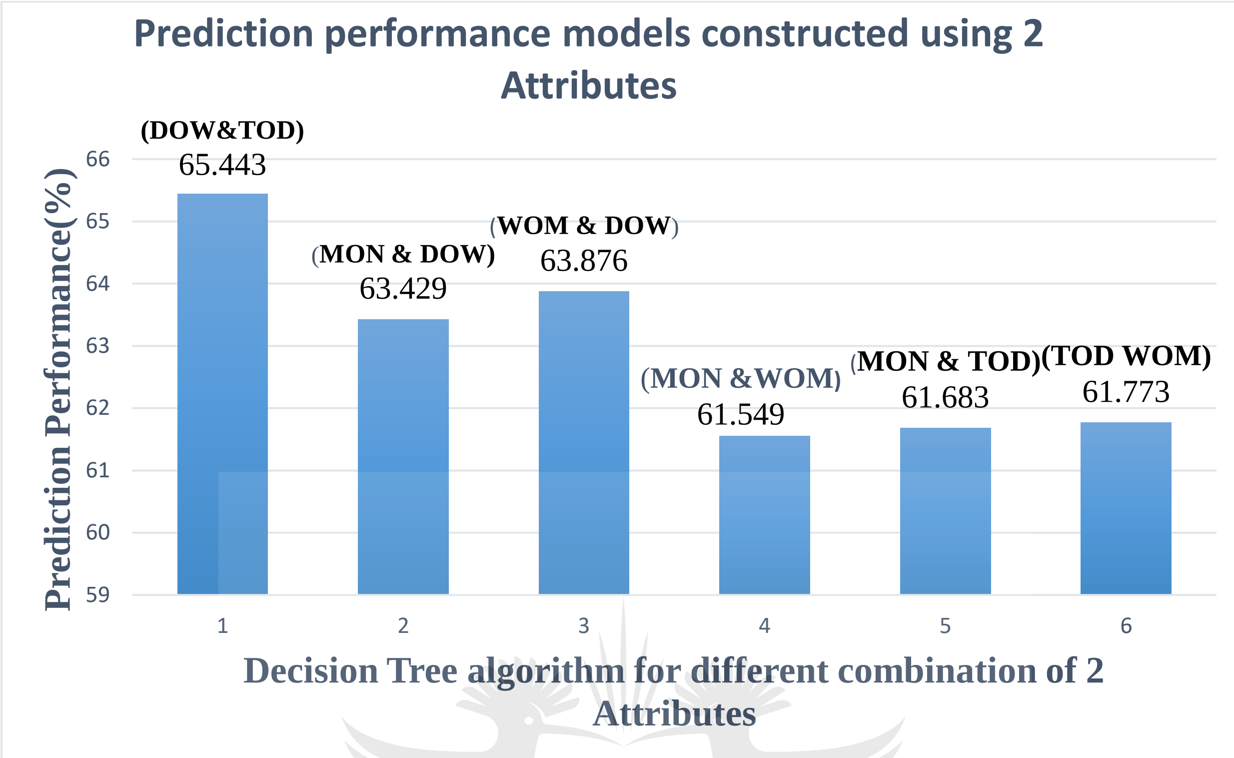


Figure 4.24: The prediction performance of the decision tree models constructed from J48 and a combination of 2 attributes (DOW and TOD), (MON and DOW), (WOM and DOW), (MON and WOM), (MON and TOD) and (TOD and WOM)

The results in Figure 4.24 mean that the average prediction of (DOW and TOD) model for the departure flight data was the best at 65.443% prediction performance.

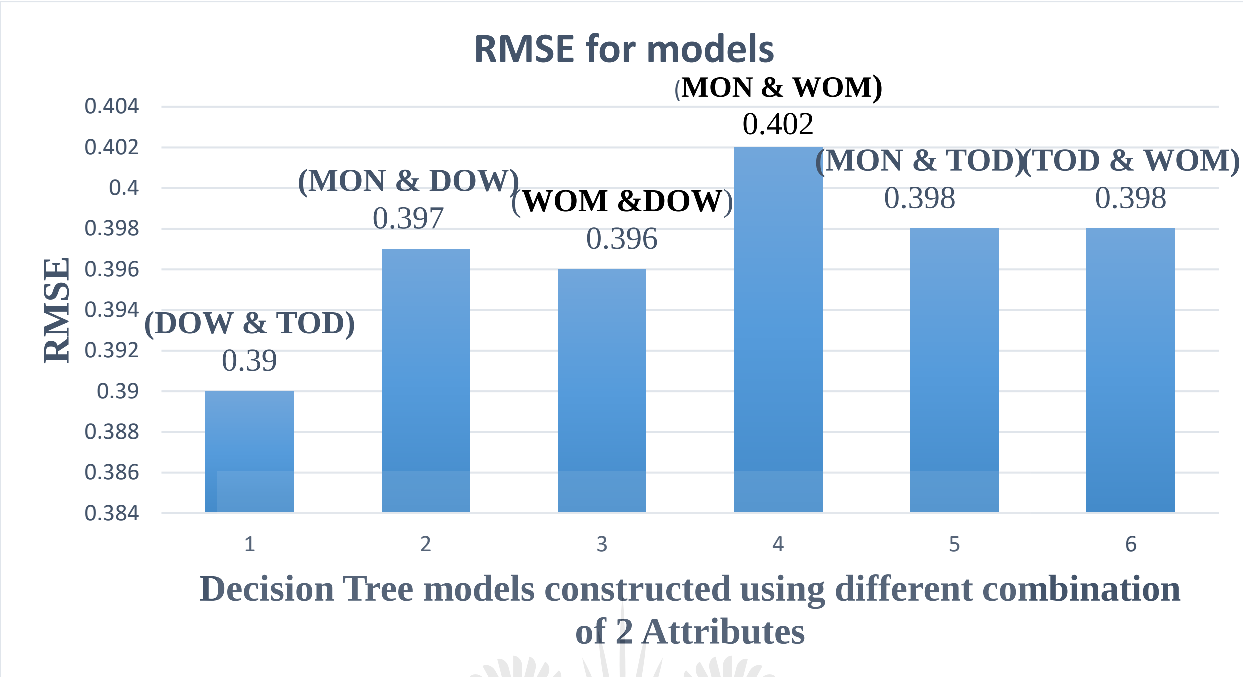


Figure 4.25: The RMSE for selected decision tree models of (DOW and TOD), (MON and DOW), (WOM and DOW), (MON and WOM), (MON and TOD) and (TOD and WOM)

Figure 4.25 shows that the lowest RMSE for the 6 decision tree models is 0.39.

Figures 4.26 and 4.27 show the results where attribute selection (combination of 3 attributes per model) was used to identify the best performing attributes among the rest when used with the J48 algorithm.

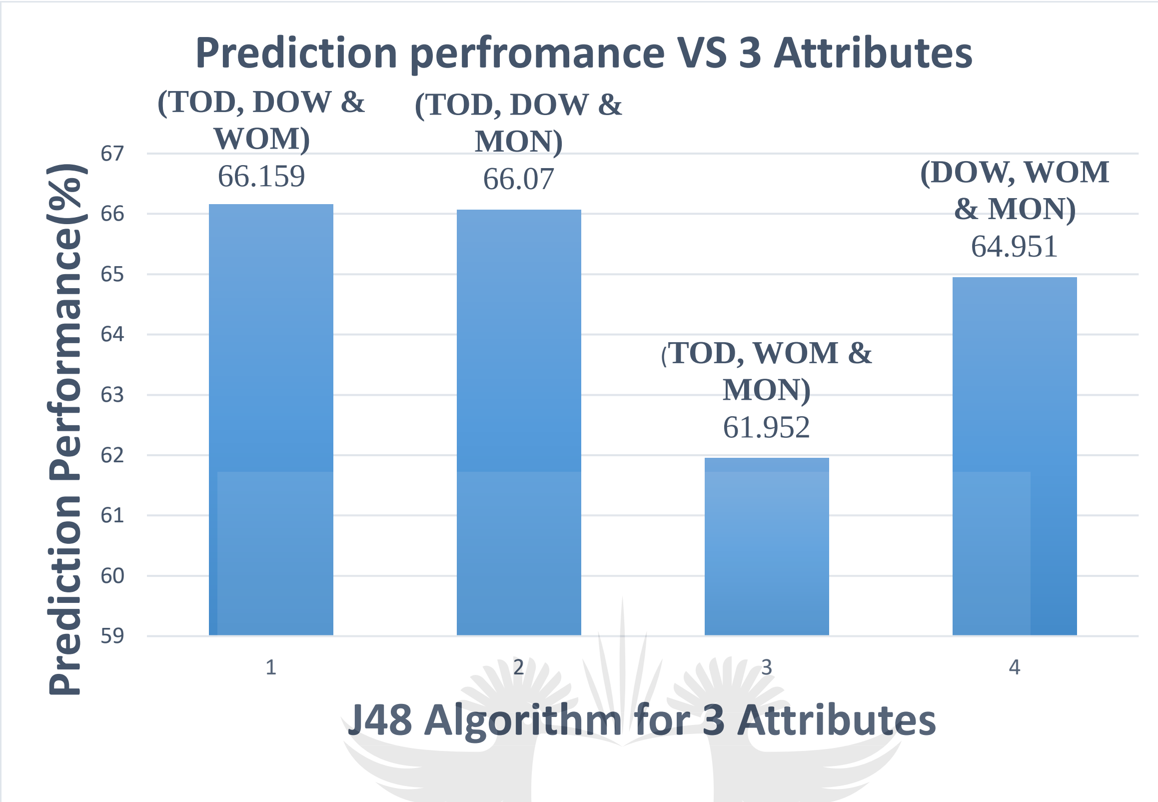


Figure 4.26: The prediction performance of models obtained when a combination of 3 attributes were used with J48 algorithm. On the x-axis (TOD, DOW and WOM), (TOD, DOW and MON), (TOD, WOM and MON) and (DOW, WOM and MON) are the 4 models

In Figure 4.26 the results revealed that the best prediction performance of decision tree models from J48 and a combination (TOD, DOW and WOM) attributes was 66.159%.

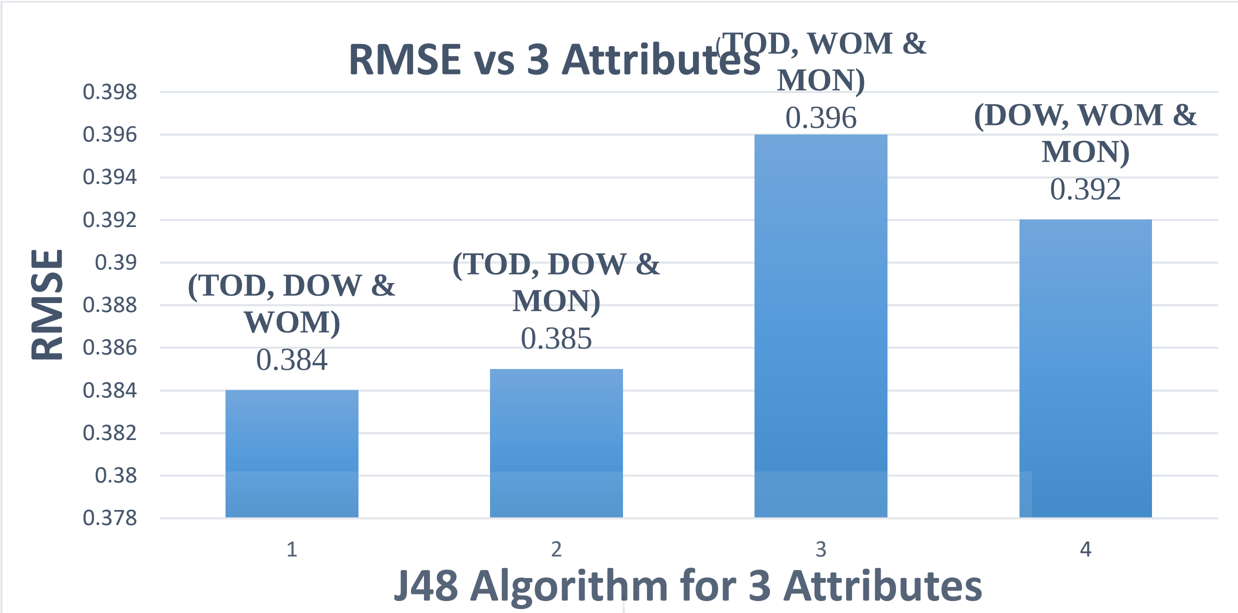


Figure 4.27: Shows RMSE for decision tree models from selected attributes (TOD, DOW and WOM), (TOD, DOW and MON), (TOD, WOM and MON) and (DOW, WOM and MON)

The results in Figure 4.27 shows that the combination of the attributes that include (TOD, DOW and WOM) had the lowest RMSE of 0.384.

Figure 4.28 and 4.29 show the results when flight departures data were used for constructing models using Decision (J48), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and their RMSE values recorded.

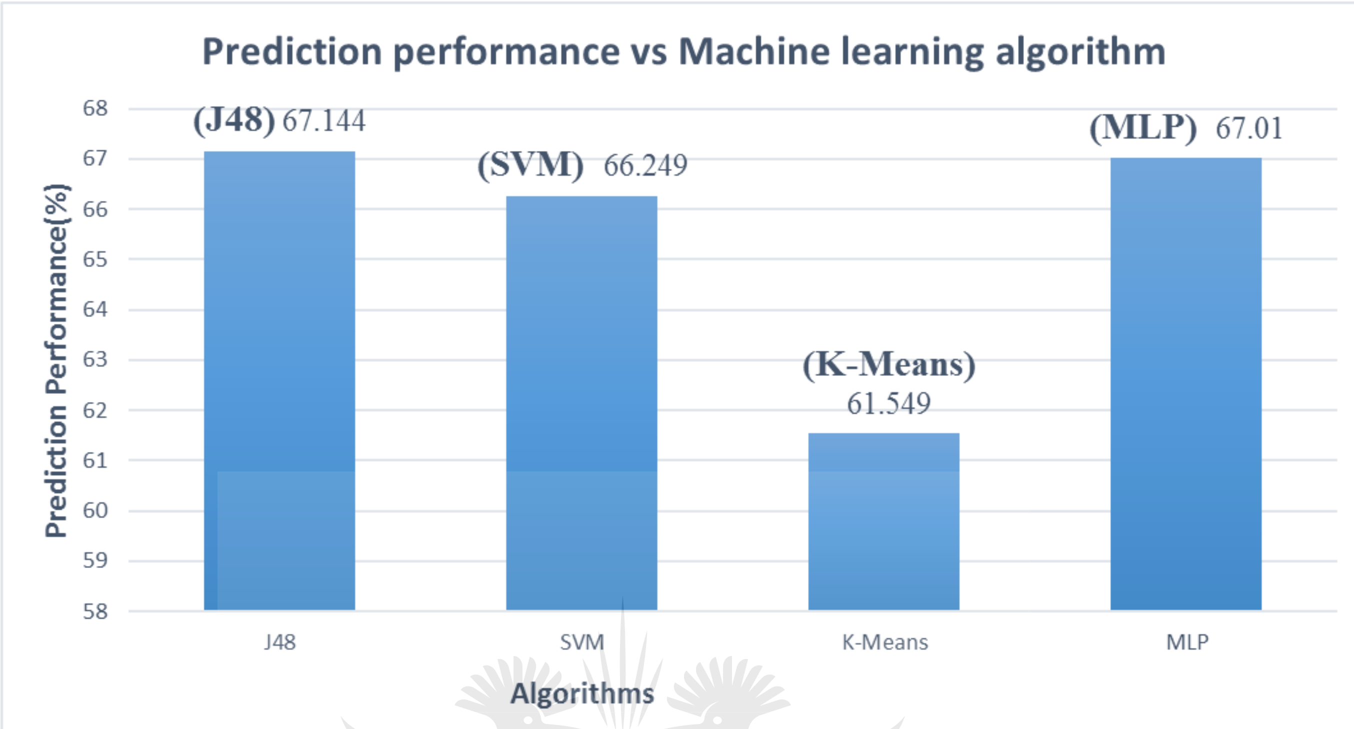


Figure 4.28: The prediction performance of J48, SVM, K-Means and MLP models for the flight departures data. All the attributes (DOW, TOD, MON & WOM) plus the Class attribute were used during the design of the models

The results of Figure 4.28 mean that J48 was the best model among SVM, K-Means, and MLP.

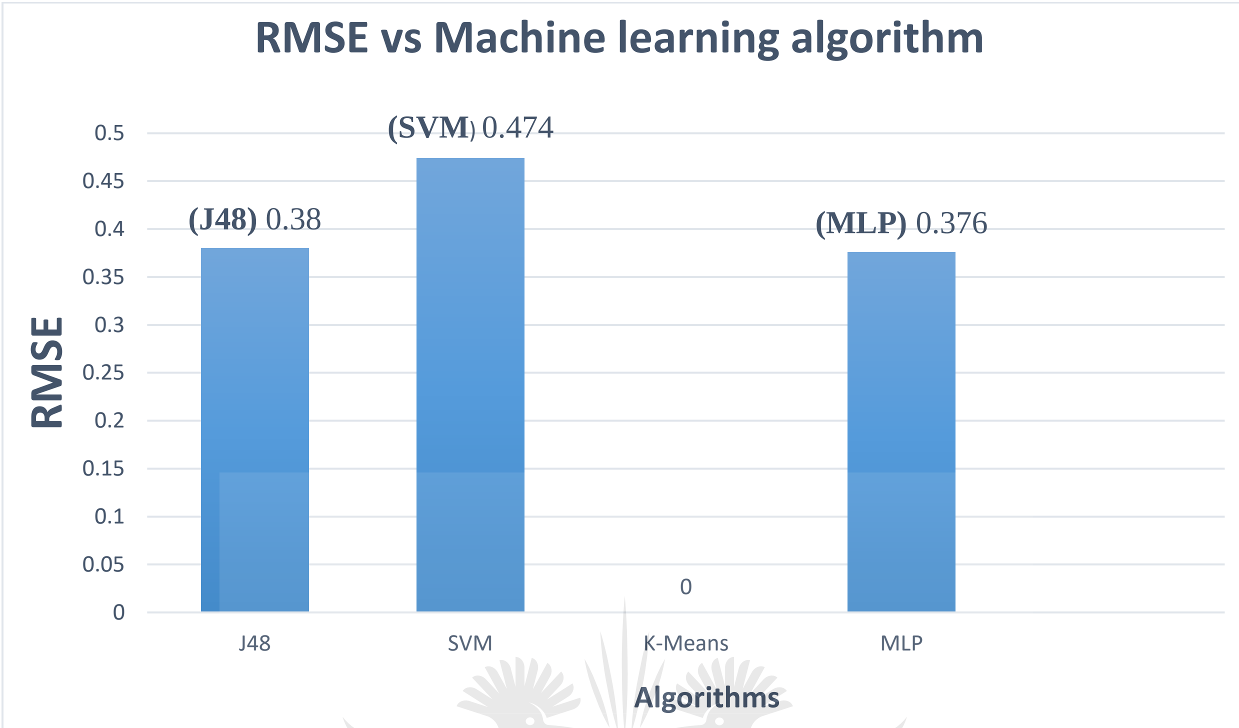


Figure 4.29: The RMSE for the J48, SVM, K-Means and MLP algorithms. All the attributes (DOW, TOD, MON & WOM) plus the Class attribute where used to construct these models

Figure 4.29 shows the average RMSE of the Decision tree, the Support Vector Machine, the K-Means Clustering and MLP for the JNB flight departures data for the year 2017. K-Means clustering has 0 RMSE because the RMSE was not applicable for the K-Means algorithm. The results indicate that MLP has the lowest RMSE of 0.376.

4.4 Data post processing

This section explains how the results from the experiments were evaluated using measures that include the ROC area and the confusion matrices.

4.4.1 Performance comparison strategies

It is a requirement "that the performance of various models are compared with each other to justify the selection of a model for a given problem" (Witten et al., 2011). The "Confusion matrix" was used for this purpose as was covered in section 4.3.

4.4.2 ROC (Receiver operator characteristic) test

The ROC "is a plot of the true positive rate against the false positive rate" (Witten et al., 2011). The ROC curve is used to measure accuracy of a predictive or classifier model (Witten et al., 2011). The ROC Curves were used in measuring the accuracy of flight departure delay models. The ROC curve is a measure of 'Sensitivity' or the true positivity rate against the 'Specificity' or the false positive rate. The Area Under the ROC Curve (AUC) is a measure of how a parameter can be distinguished into specified class output values or attributes.

A ROC curve demonstrates several things:

The closer the curve "follows the upper left-hand border and then the top border of the ROC space, the more accurate the results (higher AUC)" (Remco et al., 2008; Witten et al., 2011). The ROC curve from the conducted experiments in Table 4.26, Table 4.28 and Table 4.30 generated the ROC curves for J48, SVM and MLP algorithms for "Delayed," "On Time" and "Cancelled" Class's. The MLP algorithm generated an ROC curve of an area 0.71 for the "Delayed" Class and 0.709 for the "On Time" Class, which had a better accuracy than J48 and SVM algorithms. The J48 algorithm, however, generated an ROC curve of an area of 0.847 for the "Cancelled" Class, which had a better accuracy than SVM and MLP algorithms. The higher the area being close to 1 the better the performance accuracy.

4.4.3 Performance analysis of the models

Testing was conducted separately for each model namely the Decision Trees (J48), SVM and MLP, which were supervised learning algorithms and K-means which was an unsupervised learning algorithm. The same training and dataset were used for all three models.

“Accuracy (AC) = The proportion of the total number of predictions that were correctly computed (Shodhganga@INFLIBNET, 2002-2013)”, using the equation 4.2.

$$AC = \frac{\sum True\ positive + \sum True\ negative}{\sum Total\ (positives + negatives)} \qquad \text{Equation 4.2}$$

True Positives (TP): These “are the correctly predicted positive values, that is, the value of actual class is yes and the value of predicted class is also yes” (Shodhganga@INFLIBNET, 2002-2013). E.g. The TP for Class “Delayed”, “On Time” and “Cancelled” using J48 algorithm were 0.347, 0.884 and 0, respectively. Similarly, the TP were 0.348, 0.868 and 0, respectively for Class “Delayed”, “On Time” and “Cancelled” using SVM algorithm. Lastly, the TP for Class “Delayed”, “On Time” and “Cancelled” using MLP algorithm were 0.401, 0.849 and 0, respectively.

True Negatives (TN): These “are the correctly predicted negative values, which means that the value of actual class is no and value of predicted class is also no” (Shodhganga@INFLIBNET, 2002-2013).

False Positives (FP): When “actual class is no and predicted class is yes” (Shodhganga@INFLIBNET, 2002-2013). E.g. The FP for Class “Delayed”, “On Time” and “Cancelled” using J48 algorithm were 0.118, 0.661 and 0, respectively. Similarly, the FP were 0.135, 0.657 and 0, respectively for Class “Delayed”, “On Time” and “Cancelled” using SVM algorithm. Lastly, the FP for Class “Delayed”, “On Time” and “Cancelled” using MLP algorithm were 0.152, 0.609 and 0, respectively.

False Negatives (FN): When “actual class is yes but predicted class in no” (Shodhganga@INFLIBNET, 2002-2013).

Table 4.25: The Confusion matrix of the J48 Model using 4 attributes (MON, TOD, DOW & WOM) plus the Class attribute

		Predicted		
Actual		**a**	**b**	**c**
	A	285	537	0
	B	160	1 215	0
	C	6	31	0

In Table 4.25, the J48 model was constructed from the combination 4 attributes (MON, TOD, DOW & WOM) plus the Class attribute with 67.144% prediction and RMSE of 0.380. The values of the confusion matrix mean that 285 instances were correctly predicted as "Delayed" and 537 were incorrectly predicted as "On Time". In Table 4.25 it can be seen that 1 215 instances were all correctly predicted as "On Time" and 160 were incorrectly predicted as "Delayed". There were no instances predicted for "Cancelled" class due to insufficient data.

Table 4.26: Shows Class and ROC Area values observed using J48 Model using 4 attributes (MON, TOD, DOW & WOM) plus the Class attribute

Class	**ROC Area**
A – Delayed	0.684
B – On Time	0.684
C – Cancelled	0.847

Table 4.26 shows the class name and ROC Area values for each of the classes observed for Decision Trees (J48) model. The “Delayed” and “On Time” class have obtained 0.68 performance accuracy and “Cancelled” class obtained 0.847 performance accuracy. “The average of these respective values is used for final performance comparison.”

The ROC curves that follow represent excellent, good and poor, plotted on separate Figures. The accuracy of the test depends “on how well the test separates the group being tested into positive values and negative values. Accuracy is measured by the AUC” (Shodhganga@INFLIBNET, 2002-2013). An area of 1.0 means perfect results. An area of 0.5 means Good results. An area of 0 means poor results.

Steps to visualise ROC Curves

All the experiments that follow start with steps 1 to 3. The difference is the options that are based on the type of algorithm and change the type of class to select. Thus, in subsequent ROC Curves steps 1 to 3 are assumed.

4.4.3.1 Experiment 1: Accuracy evaluation for J48, SVM & MLP models using ROC Curve for “Delayed”, “On Time” and Cancelled”

Step 1:

The procedure is the same as Experiment 1 (Step 1) covered in section 4.3.1.1.

Step 2:

Click on the classify tab to select the classifier (e.g. J48, MLP and SVM), click on choose on the populated drop-down menu, select the preferred algorithm (e.g. J48, MLP or SVM), to create the model click on “Start button”.

Step 3:

1. Right click on the ‘Result list’.

2. On the popped-up list, scroll down and hover over ‘Visualize threshold curve’.

3. On the popped-up sub list of three classes will be displayed, select - (e.g. “Delayed”, “On Time” or “Cancelled”).

4. Figure 4.30 will be displayed for this first experiment displaying the “Delayed” Class and the following Figures, E.g. Figure 4.31 (“On Time”) and Figure 4.32 (“Cancelled”) class’s will be displayed based on algorithm and class options.

5. On ‘Select Instance’ drop down, click the drop down and choose ‘Rectangle’.

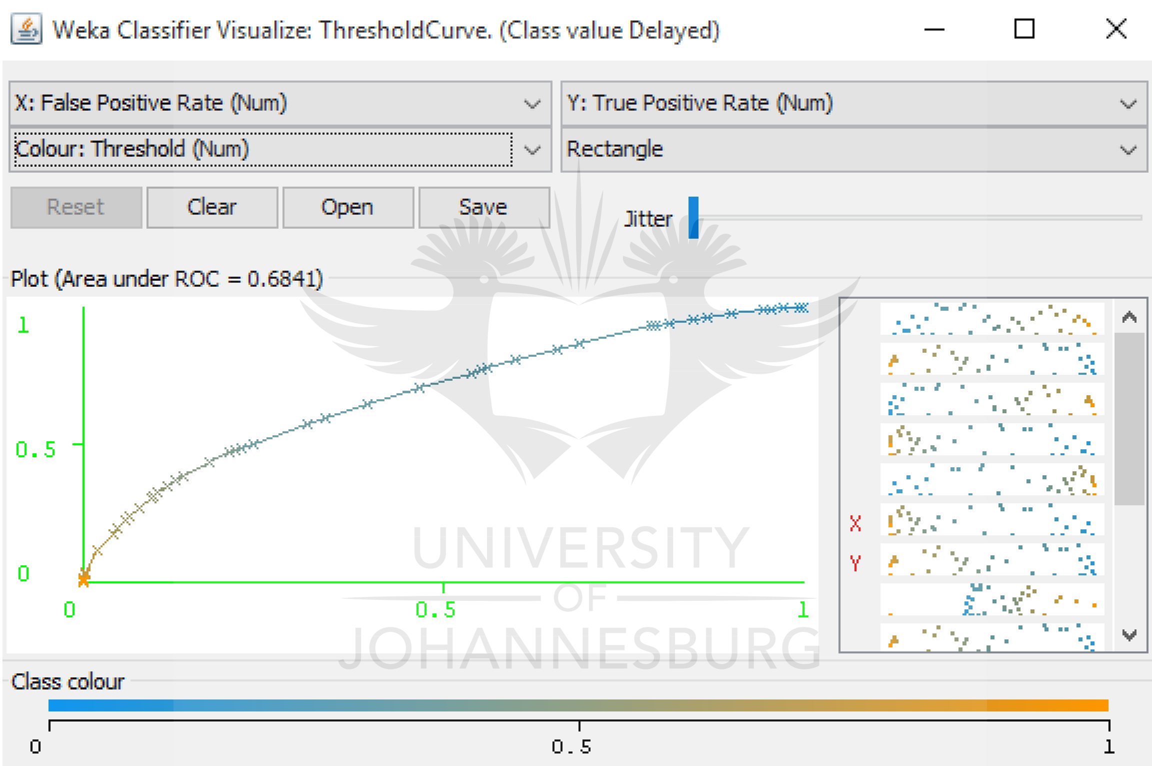


Figure 4.30: Evaluation using ROC Curve for “Delayed” Flights from J48 algorithm

Figure 4.30 shows the area under ROC curve for “Delayed” flights is 0.68, which indicates the performance accuracy for “Delayed” class using J48 algorithm are good.

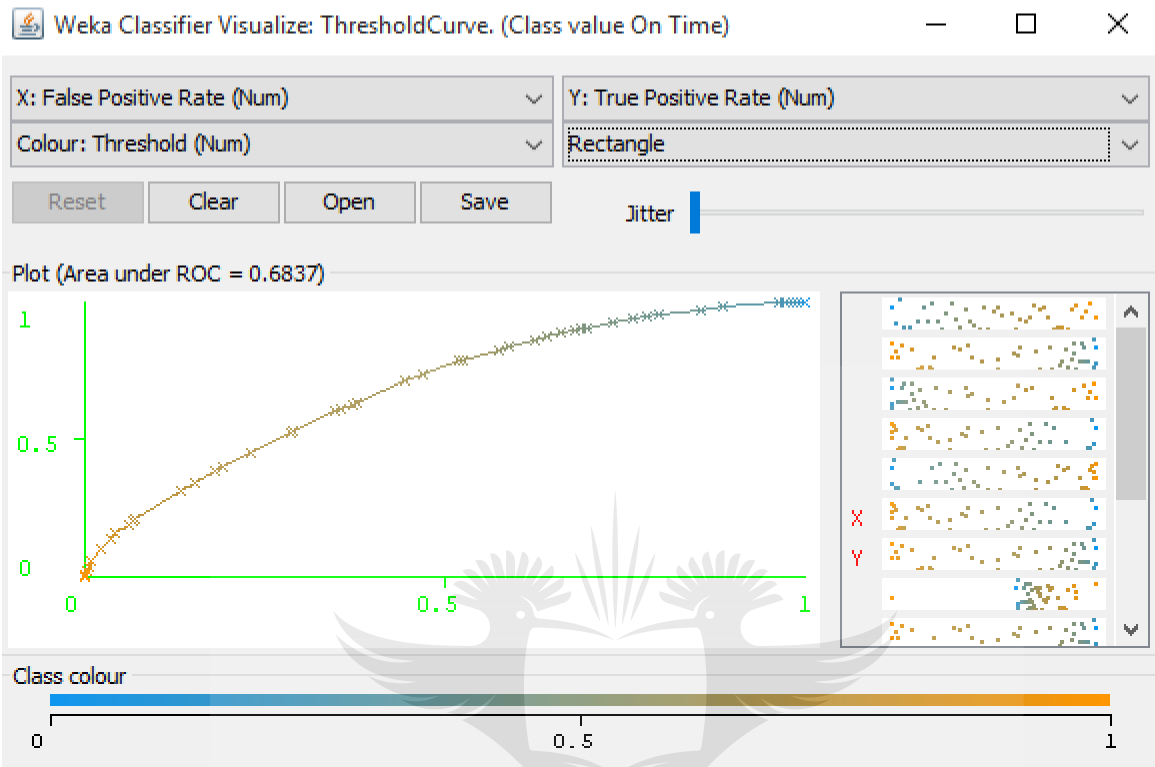


Figure 4.31: Evaluation using ROC Curve for "On Time" Flights using J48 algorithm

Figure 4.31 shows the area under ROC curve for "On Time" flights is 0.68, which indicates the performance accuracy for the On-Time class using J48 algorithm are good.

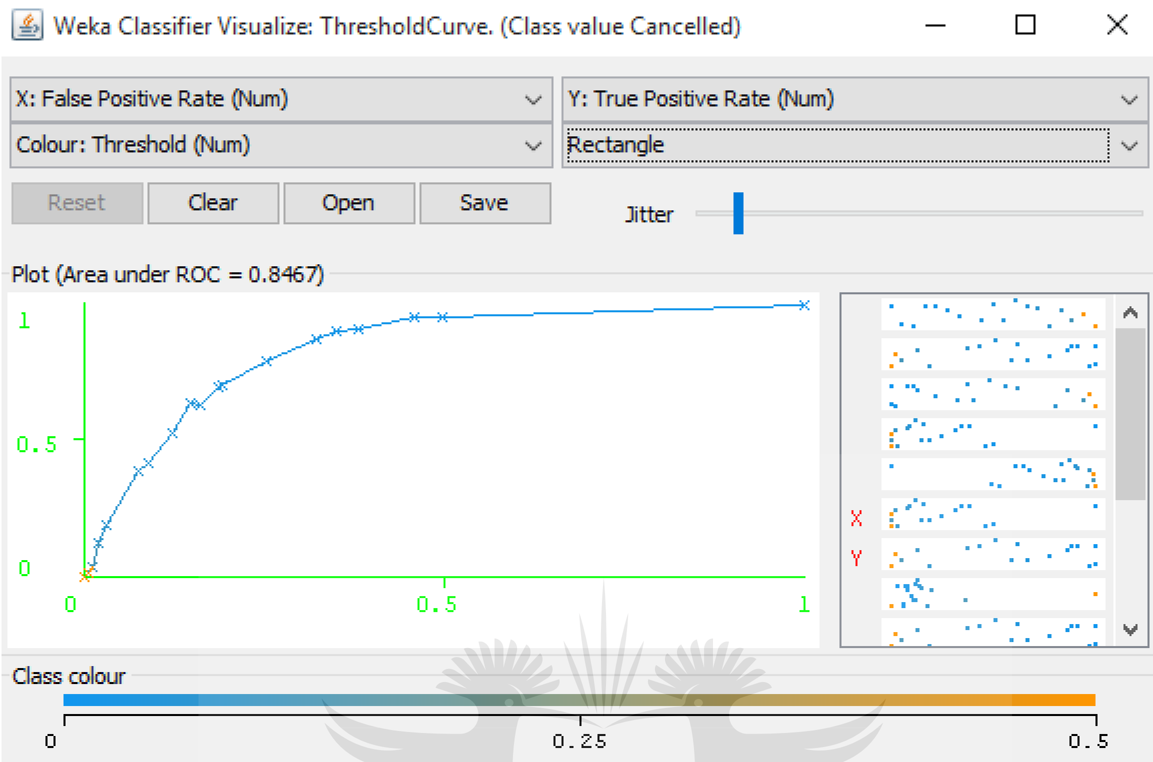


Figure 4.32: Evaluation using ROC Curve for "Cancelled" Flights

Figure 4.32 shows the area under ROC curve for "Cancelled" flights is 0.85, which indicates the performance accuracy of the "Cancelled" class using J48 algorithm are very good.

The accuracy (AC) was calculated using the confusion matrix in Table 4.28.

$$\text{Accuracy} = \frac{\sum True\ positive + \sum True\ negative}{\sum Total\ (positives + negatives)}$$

(285 + 1215 / (285 + 537 +160 +1215 +6 +31)

= 1500/2234

J48 AC = 0.671

Decision Tree (J48) algorithm obtained 67.1 % performance accuracy.

Table 4.27: The Confusion matrix of the SVM Model constructed from the combination of (TOD), (DOW), (MON) and (WOM) attributes plus the Class attribute

	Predicted			
		a	**b**	**c**
Actual	**A**	286	536	0
	B	181	1 194	0
	C	9	28	0

The SVM model obtained 66.249% prediction and RMSE of 0.474. The results in Table 4.27 show that 286 instances were correctly predicted as "Delayed", 536 were incorrectly predicted as "On Time", 1 194 instances were all correctly predicted as "On Time", 181 were incorrectly predicted as "Delayed", 9 instances were incorrectly predicted as "Delayed", and 28 instances were incorrectly predicted as "On Time".

Table 4.28: Shows Class and ROC Area values for SVM model using 4 attributes (MON, TOD, DOW & WOM) plus the Class attribute

Class	**ROC Area**
A – Delayed	0.607
B – On Time	0.606
C – Cancelled	0.500

In Table 4.28, the performance of the classes is shown in Figure 4.32. The average of these respective values "is used for final performance comparison" (Shodhganga@INFLIBNET, 2002-2013).

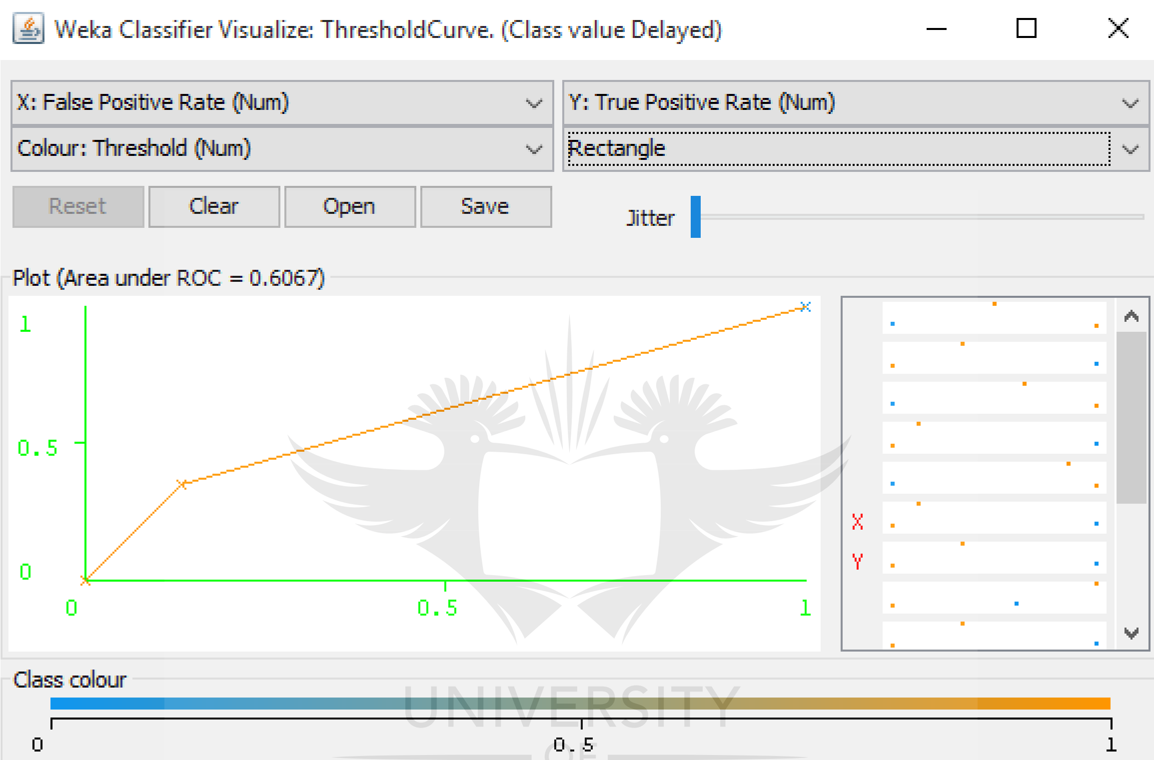


Figure 4.33: Evaluation using ROC Curve for "Delayed" Flights

The results in Figure 4.33 mean SVM model for "Delayed" flights have a good performance accuracy.

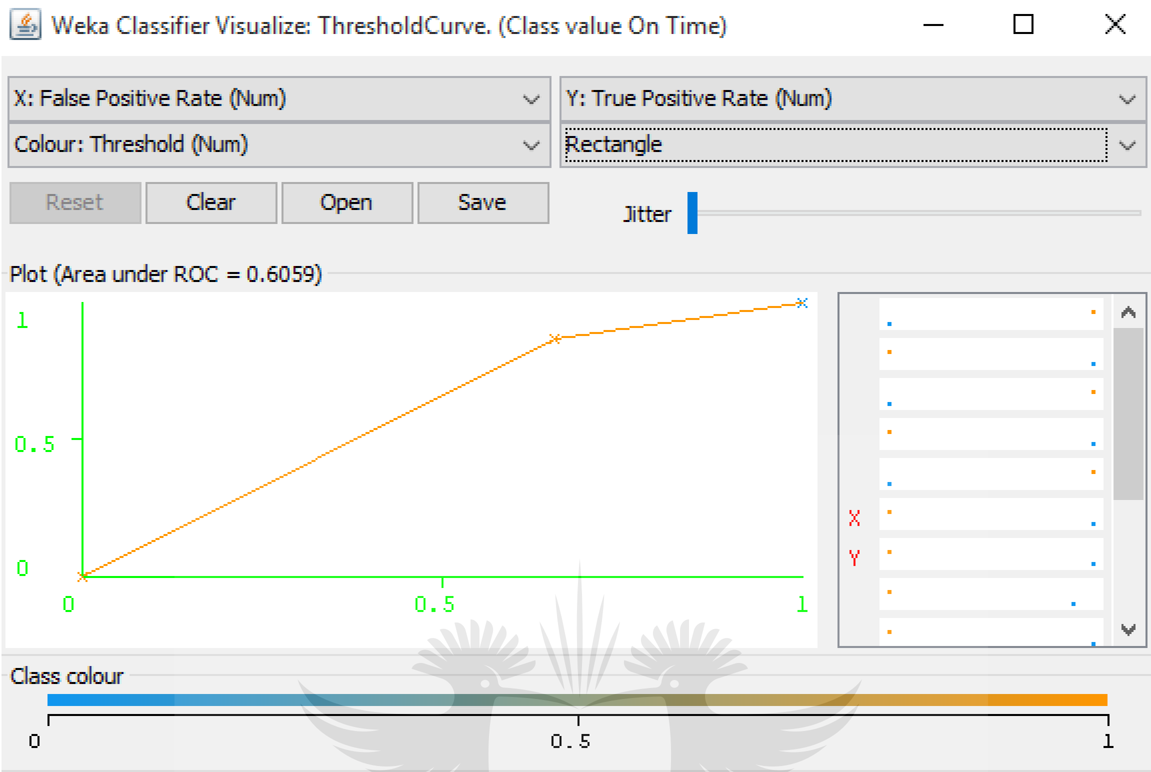


Figure 4.34: Evaluation using ROC Curve for "On Time" Flights

The results in Figure 4.34 mean the SVM model for "On Time" flights have a good performance accuracy.

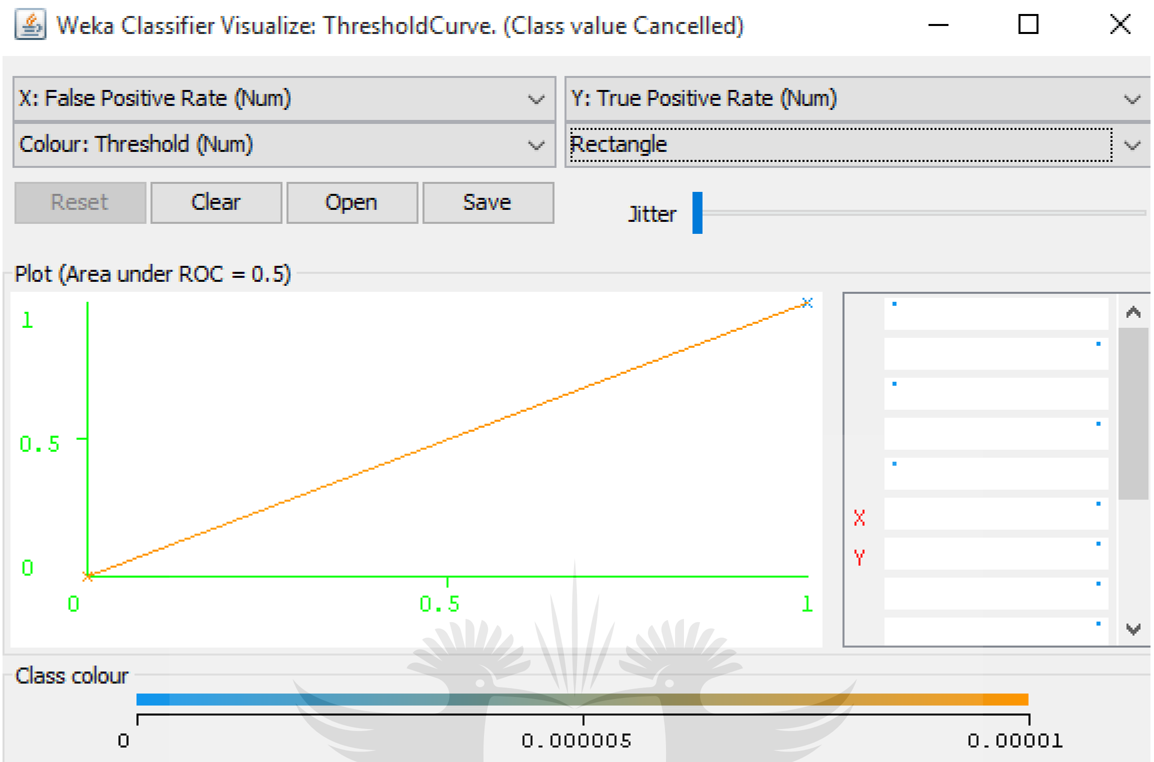


Figure 4.35: Evaluation using ROC Curve for "Cancelled" Flights using SVM algorithm

The results in Figure 4.35 mean that the SVM model for "Cancelled" flights at (0.5) performance accuracy is good.

The accuracy (AC) was calculated using the confusion matrix in Table 4.26.

$$\text{Accuracy} = \frac{\sum True\ positive + \sum True\ negative}{\sum Total\ (positives + negatives)}$$

(286 + 1194/ (286 +536 + 181 +1194+ 9 +28))

= 1480/2234

SVM AC = 0.662

The SVM model obtained 66.2 % performance accuracy. The rate indicates a moderate accuracy as it is approaching 1, which is the highest accuracy.

Table 4.29: The Confusion matrix of the MLP Model using 4 attributes (MON, TOD, DOW & WOM) plus the Class attribute

Actual	**Predicted**			
		a	**b**	**c**
	A	330	492	0
	B	208	1 167	0
	C	6	31	0

The results in Table 4.29 mean that the MLP model constructed from the combination of (TOD), (DOW), (MON) and (WOM) attributes performance of 67.010% prediction is moderate. The results mean that 330 instances were correctly predicted as “Delayed”, 492 were incorrectly predicted as “On Time”, 1 167 instances were correctly predicted as “On Time”, 208 instances were incorrectly predicted as “Delayed”, 6 instances were incorrectly predicted as “Delayed” and 31 instances were incorrectly predicted as “On Time”.

Table 4.2: Class and ROC Area values observed for MLP model using 4 attributes (MON, TOD, DOW & WOM)

Class	ROC Area
A – Delayed	0.710
B – On Time	0.709
C – Cancelled	0.672

Table 4.30 shows the class name and ROC Area values for each of the classes observed for the MLP model. The average of these respective values “is used for final performance comparison” (Shodhganga@INFLIBNET, 2002-2013).

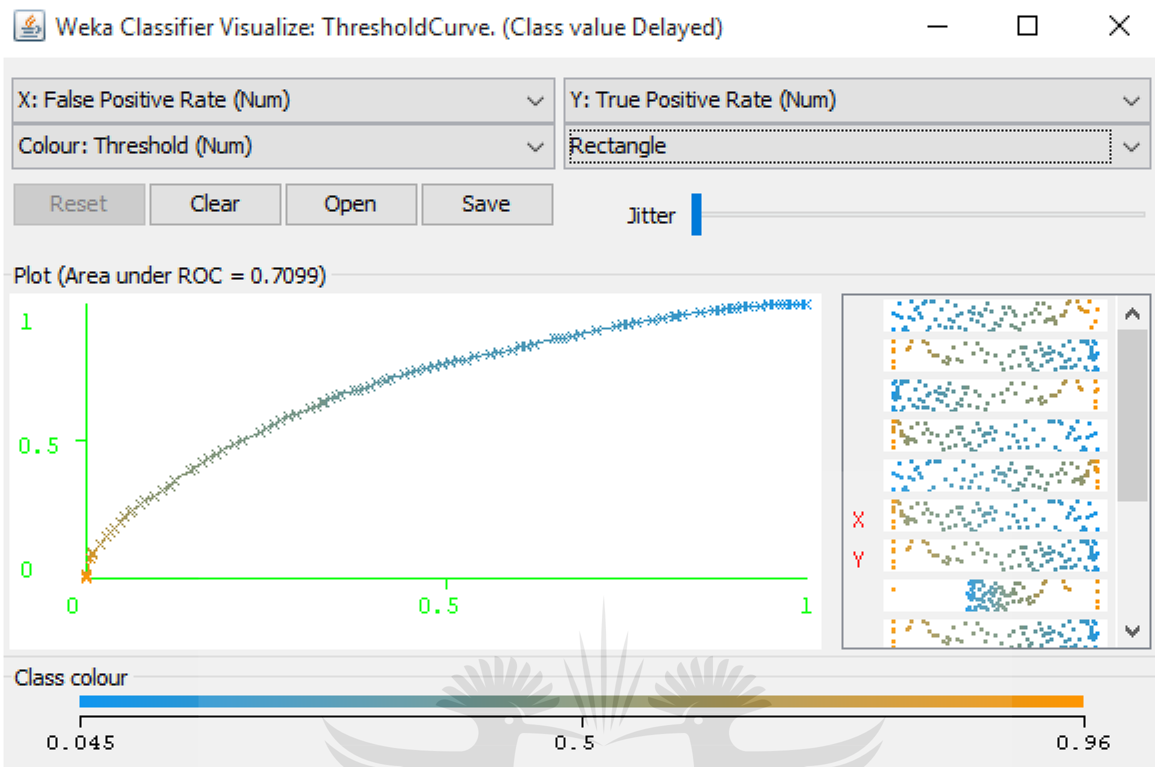


Figure 4.36: Evaluation of using ROC Curve for "Delayed" Flights

The results ROC of 0.7099 in Figure 4.36 mean the MLP curve for "Delayed" flights has a good performance accuracy.

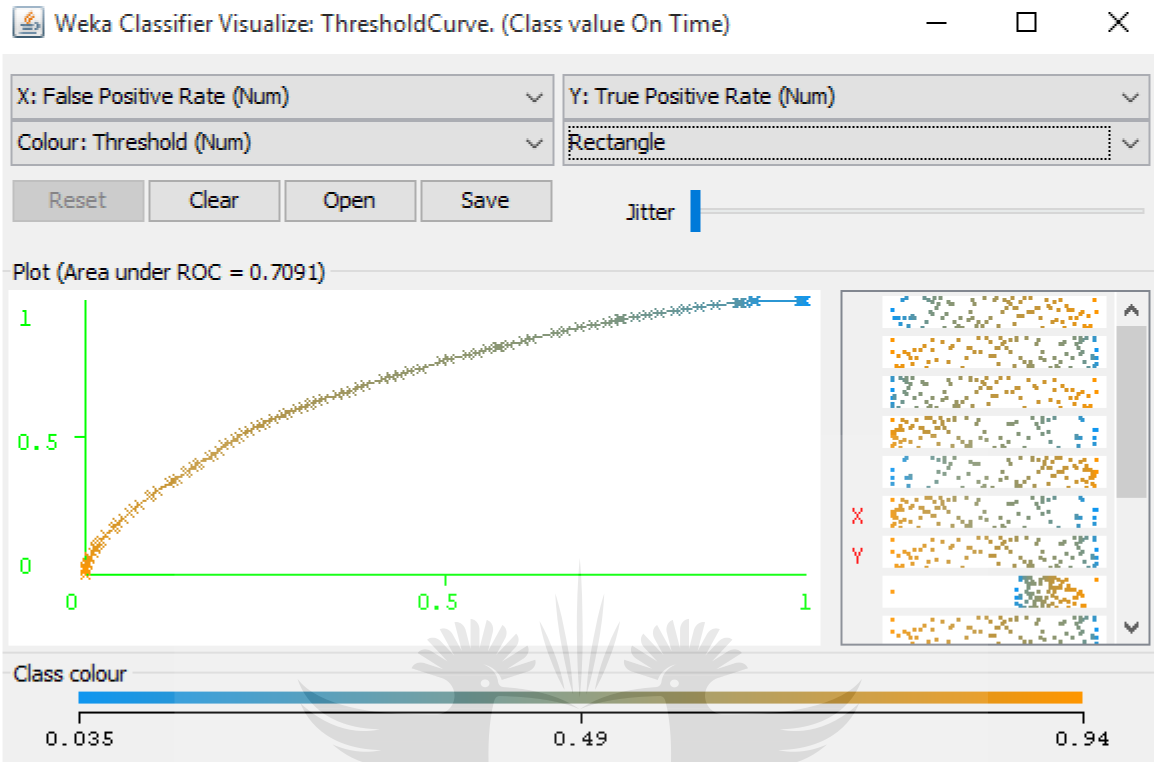


Figure 4.37: Evaluation of using ROC Curve for "On Time" Flights

The results ROC of 0.709 in Figure 4.37 mean the MLP model "On Time" flights have a good performance accuracy.

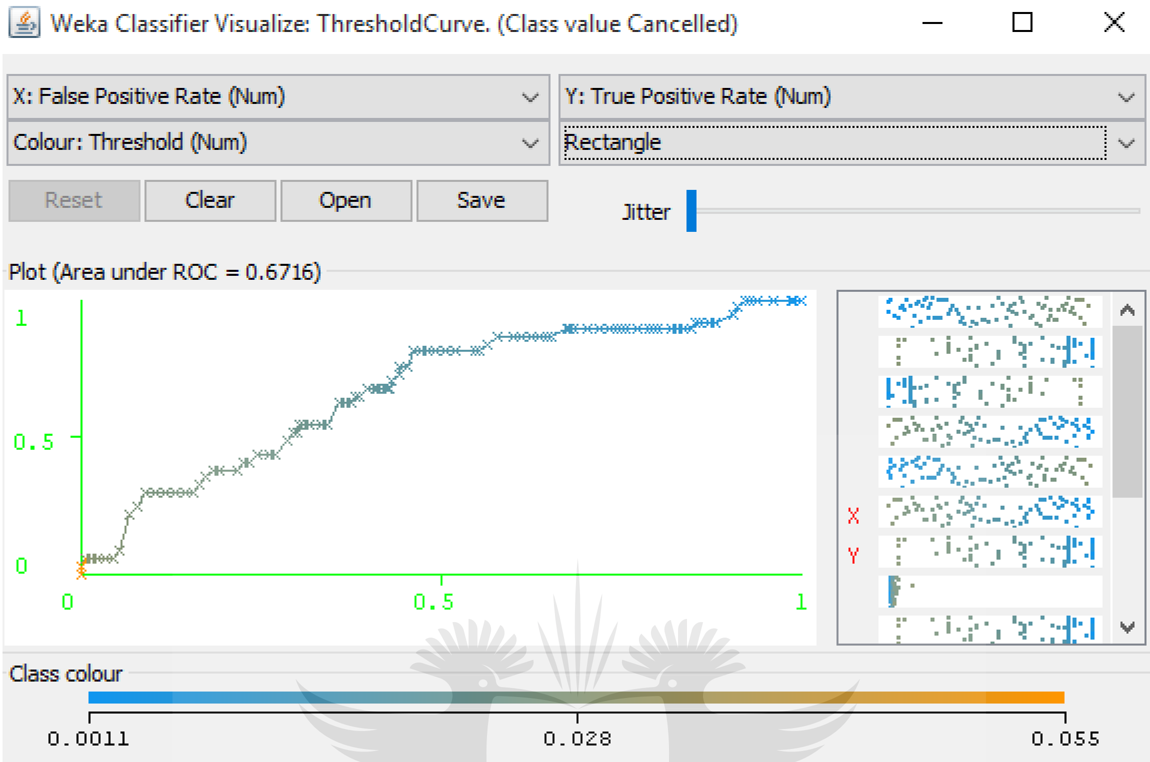


Figure 4.38: Evaluation using ROC Curve for "Cancelled" Flights

Figure 4.38 shows the AUC for "Cancelled" flights is 0.672, which indicates the performance accuracy for the "Cancelled" class using MLP algorithm are good.

The accuracy (AC) was calculated using the confusion matrix in Table 4.28

$$\text{Accuracy} = \frac{\sum True\ positive + \sum True\ negative}{\sum Total\ (positives + negatives)}$$

(330 +1167/ (330 + 492+ 208 +1167 +6 +31))

= 1497 /2234

MLP AC = 0.67

The MLP algorithm obtained 67 % performance accuracy.

4.4.4 Comparison of models

Table 4.31: Shows the summary of model comparisons

Model	Accuracy	ROC Area
J48	67.1%	0.739
SVM	66.2%	0.571
MLP	67%	0.697

In Table 4.31, the accuracy of the Decision Trees (J48) algorithm is the highest at 67.1% performance accuracy. Table 4.31 also shows the J48 algorithm obtained the highest ROC area of 0.739, which indicates that the decision tree is the best suited tool for this task.

4.5 Chapter conclusion

Pre-processing for the flight departures data was conducted using the SQL Server Management Studio. Machine learning algorithms, namely the Decision Trees (J48), the MLP, the SVM and the K-Means Clustering were used. The prediction results from the 4 models were compared to determine the best performing model. The evaluation results in Table 4.31 mean that the Decision Tree (J48) was the best suited for the task.

ROC was used to measure the accuracy of the classifiers for the 3 models namely the Decision Trees (J48), the MLP and the SVM. Decision Trees (J48) obtained the highest accuracy of 0.739, the MLP obtained second best accuracy of 0.697, and the SVM obtained third best accuracy of 0.376.

CHAPTER 5: DISCUSSION, CONCLUSION AND RECOMMENDATIONS

5.1 Discussion

The results indicate that the Decision Tree (J48) algorithm is useful for predicting flight departure delays. It scored 67.144% prediction accuracy and 0.380 RMSE value compared to the SVM, K-Means and MLP algorithms, as shown in Table 4.7 in section 4.3. Table 4.31 in section 4.3 shows J48 is the highest at 67.1% for performance accuracy.

The results were influenced by the "Cancellation" attribute that had a few instances (37) compared to the "On-Time" attribute that had 1 375 instances and the "Delayed" attribute that had 822 instances, as shown in Table 3.1 section 3.3. In addition, only departures were considered, and the names of destinations were ignored, as the objective was to get a critical mass of departures because there are only a few flights per day. The highest number of flights between JNB and CPT are 12 per day.

This study used 4 SVM algorithms and obtained better results than Bandyopadhyay and Guerrero's (2012) study and the data of this study was cleaned. Bandyopadhyay and Guerrero (2012) used 2 algorithms and did not perform data cleaning. This study used cross-validation, whereas Bandyopadhyay and Guerrero (2012) did not used cross-validation.

The results of this study, however, are poorer than those of Choi et al. (2016) because of a missing weather attribute that included 15 years of weather data. The results were also poorer than those of Evans et al. 2002) because of the weather attribute used. The results of Kim et al. (2016) were better than this study's results because Kim et al. (2016) combined multiple models based on the deep learning paradigm, Recurrent Neural Networks (RNN). This study's results were poorer than those obtained Xing & Tang (2016) because this study only focused on the On-time variable and not the delays attribute.

The implications of these results are that tourists will be less frustrated as the airline will have forewarned them or given them a window of time for the flight departure. SA grew in double digits by 12.8%, to reach over 10.0 million international tourists in 2016 and the demand of flights at

ORTIA has grown tremendously (Traveller24, 2017). The travellers will have the ability to foretell if the departing flights will depart on time, or be delayed, or be cancelled using this study's model. The economy will benefit using this study's model as the loss of revenue will be reduced regarding flights being delayed at ORTIA.

5.2 Conclusion and recommendations

This study addressed the live flight departure delays that affected the passengers, businesses, economy and the tourists. This study focused on the SAA carrier domestic flights departing from Johannesburg ORTIA. The data were collected from ORTIA live departures website for a period of two months. A quantitative method was used for this study and a site visit was conducted to collect the data.

Prediction models for live flight departure delays at Johannesburg ORTIA were constructed based on machine learning algorithms. The results showed that the Decision Trees (J48) algorithm outperformed all the constructed algorithms such as SVM, K-Means and MLP.

The results of this study elaborated that the attribute selection method and machine learning algorithms are good in predicting live flight departures.

Recommendations

SAA should adopt results from this dissertation (the model) as such a decision is likely to create an excellent passenger experience. The live flight departure delay problems are likely to be completely eradicated.

REFERENCE LIST

Aksenova., S. S. (ed). 2004. Weka explorer tutorial. School of Engineering and Computer Science, Department of Computer Science, California State University, Sacramento. California. [Online] Available at https://de.scribd.com/document/290525169/WEKA-Explorer-Tutorial-pdf [Accessed 03 January 2018].

Allen D.M. 1974. The relationship between variables selection and data augmentation and a method for prediction. Technometrics, 16(1):125-127.

Alonso, H. & Loureiro, A. 2015. Predicting flight departure delay at Porto Airport: A preliminary study. 2015 7th International Joint Conference on Computational Intelligence (IJCCI), 12-14 Nov. 2015, Lisbon, Portugal.

ATNS SOC Limited. 2018. Annual Reports for The Africa Indian Ocean (AFI) & Very Small Aperture Terminal (VSAT), Gauteng. [Online] Available at https://www.atns.com/reports.php [Accessed 16 November 2018].

Auria, L. & Moro, R.A. 2008. Support Vector Machines (SVM) as a Technique for solvency analysis. German Institute for Economic Research, DIW Berlin, pp. 1-88.

Bakhshandeh, R., Shahgholian, K. & Shahraki, A. 2013. Model for reducing flights delays using system dynamics (Case Study in Iran Airports Company). Interdisciplinary Journal of Contemporary Business in Research, 4(9):746 – 757.

Bandyopadhyay, R. & Guerrero, R. 2012. Predicting airline delays. [Online] Available at http://cs229. stanford. edu/proj2012/BandyopadhyayGuerrero-PredictingFlightDelays

Blokker, J. 2008. The application of SVM to algorithmic trading. Stanford University, CS229 Term Projects, pp.1-4.

Boser, E., Guyon, I. & Vapnik, V. 1992. A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM Press, 27-29 July 1992, Pittsburgh, Pennsylvania, USA.

Bottou L., Cortes C., Denker J., Drucker H., Guyon I., Jackel L., LeCun Y., Muller U., Sackinger E., Simard P. & Vapnik V. 1994. Comparison of classifier methods: A case study in handwriting digit recognition. Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5), 9-13 Oct. 1994, Jerusalem, Israel.

Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, W. & Scuse, D. 2008. Weka manual for version 3-6-0.

Breil, R., Delahaye, D., Lapasset, L. & Féron, É. 2016. Multi-agent systems for air traffic conflicts resolution by local speed regulation and departure delay. 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 25-29 Sept. 2016, Sacramento, CA, USA.

Burns, N. & Grove, S.K. 1993. The practice of nursing research: Conduct, critique and utilization. 2nd ed. Philadelphia: W.B. Saunders.

Callan, R. 2003. Artificial intelligence. New York: Macmillan.

Chai T. & Draxler R.R. 2014. Root Mean square error (RMSE) or Mean absolute error (MAE): Arguments against avoiding RMSE in literature. Geoscientific Model Development, 7(3):1247-1250.

Chang, C.C. & Lin, C.J. 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):27.

Choi, S., Kim, Y.J., Briceno, S. & Mavris, D. 2016. Prediction of weather-induced airline delays based on machine learning algorithms. 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 25-29 Sept. 2016, Sacramento, CA, USA.

Choi, S., Kim, Y.J., Briceno, S. & Mavris, D. 2017. Cost-sensitive prediction of airline delays using machine learning. 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), 17-21 Sept. 2017, St. Petersburg, FL, USA.

Cunningham, P., Cord, M. & Delany, S.J. 2008. Supervised learning. In: P. Cunningham, & M. Cord, Machine learning techniques for multimedia. Berlin, Heidelberg: Springer.

Cybenko, G. 1989. Approximation by super position of a sigmoidal function. Mathematics of control, Signals and systems, 2(4):303-314.

Dash, M., Choi, K., Scheuermann, P. & Liu. H. 2002. Feature selection for clustering – A filter solution. 2002 IEEE International Conference on Data Mining, 2002. Proceedings, 9-12 Dec. 2002, Maebashi City, Japan.

Dinakaran, S. & Thangaiah, P.R J. 2013. Role of attribute selection in classification algorithms. The International Journal of Science & Engineering Research, 4(6):67-71.

Ding, J. & Tong, G. 2008. Real-time sub-time early warning of airport scheduled flight delay based on immune algorithm. Second International Symposium on Intelligent Information Technology Application, 20-22 Dec. 2008, Shanghai, China.

Evans, A.D. & Clarke, J-P. 2002. Air traffic control response to delays -A system study of Newark International Airport. MT International Center for Air Transportation.

Excelsior Digital (Pty) Ltd. 2016. Home to search and book cheap flights at OR Tambo, Johannesburg Airport. [Online] Available at http://johannesburg-airport.co.za/ [Accessed 16 August 2017].

Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R. & Lin, C.J. 2008. LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research, 9(9):1871-1874.

Fractal Design. 2016. Domestic flights route maps, schedules and airline contact, Johannesburg. [Online] Available at http://ortambo-airport.com/flights/dom.php [Accessed 16 August 2017].

Fractical Design. 2016. OR Tambo Airport JNB Live Flight Departures, Johannesburg, Gauteng. [Online] Available at http://ortambo-airport.com/flights/departures.php [Accessed 31 August 2017].

Gauteng Tourism Authority & Flow Communications. 2017. Visitors at OR Tambo International Airport, Johannesburg. [Online] Available at http://www.gauteng.net/attractions/or_tambo_international_airport/ [Accessed 16 August 2017.

Gopalakrishnan, K. & Balakrishnan, H. 2017. A comparative analysis of models for predicting delays in air traffic networks. Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Cambridge, MA, USA.

Hall, M.A. 1999. Correlation-based feature selection for machine learning. Waikato, Department of Computer Science. The University of Waikato.

Hall, M.A. 2000. Correlation-based feature selection for discrete and numeric class machine learning. Correlation-based feature selection for discrete and numeric class machine learning, 29 June – 02 July 2000, San Francisco, CA, USA, Morgan Kaufman.

Hallebone, E. & Priest, J. 2009. Business and management research: Paradigms and practices. New York: Palgrave McMillan.

Hsu, C.W., Chang C.C. & Lin, C.J. 2003. A practical guide to support vector classification. Department of Computer Science, National Taiwan University.

Independent Media (Pty) Ltd. 2017. News for navigation system issues delay flights at OR Tambo, Gauteng. [Online] Available at http://www.iol.co.za/news/south-africa/gauteng/navigation-system-issues-delay-flights-at-or-tambo-7328164 [Accessed 23 May 2017].

Johnson, T. & Savage, I. 2006. Departure delays, the pricing of congestion and expansion proposals at Chicago O' Hare Airport. Journal of Air Transport Management, 12(4):182-190.

Kantardzic, M. 2011. Data mining: Concepts, models, methods and algorithms. Hoboken, New Jersey: John Wiley & Sons.

Kayrooz, C. & Trevitt, C. 2005. Research in organisations and communities: Tales from the real world. Crows Nest: Allen and Unwin.

Khoonsari, P.E. & Motie, A. 2012. A comparison of efficiency and robustness of ID3 and C4.5: Algorithms using dynamic test and training data sets. International Journal of Machine Learning and Computing, 2(5):540-543.

Kim, Y.J., Choi, S., Briceno, S. & Mavris, D. 2016. A deep learning approach to flight delay prediction. 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 25-29 Sept. 2016, Sacramento, CA, USA.

Kohavi, R. & John, G.H. 1997. Wrappers for feature Subset Selection. Artificial Intelligence, 97(1-2):273-324.

Kohavi, R. & Quinlan, J.R. 2002, Data mining tasks and methods: Classification: decision-tree discovery. In: Handbook of data mining and knowledge discovery, pp 267-276. Oxford University Press.

Kordon, A.K. 2009. Applying computational intelligence: How to create value. IEEE Computational Intelligence Magazine, 5(2):108-109.

Kumar, V. & Poggio, T. 2000. Learning-based approach to real time tracking and analysis of faces. Proceedings Fourth IEEE Int. Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), 28-30 March 2000, Grenoble, France.

Le, P. & Zuidema, W. 2014. Perceptron and multilayered perceptron. International Conference on Measuring Technology and Mechatronics Automation, pp. 1-8.

Lehohla, P. 2016, Tourism report by Statistics South Africa, South Africa. [Online] Available at http://www.statssa.gov.za/publications/Report-03-51-02/Report-03-51-022016.pdf [Accessed 10 July 2017].

Lei, Y. & Liu, H. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. Conference: Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), 21-24 August 2003, Washington, DC, USA.

Leonard, T. & Bekker, J. Apron layout design layout and flight-to-gate assignment at Lanseria International Airport. South African Journal of Industrial Engineering, 24(1):192-206.

Liu, Y.J. & Ma, S. 2008. Flight delay and delay propagation analysis based on Bayesian Network. 2008 International Symposium on Knowledge Acquisition and Modeling, 21-22 Dec. 2008, Wuhan, China.

Liu, Y-J. & Yang, F. 2009. Initial flight delay modeling and estimating based on an improved Bayesian network structure learning algorithm. 2009 Fifth International Conference on Natural Computation, 14-16 Aug. 2009, Tianjin, China.

Lv, X, & Wang, H.2009. Flight delay alarming analysis for an airport based on Markov. 2009 First International Workshop on Education Technology and Computer Science, 7-8 March 2009, Wuhan, Hubei, China.

Madzarov, G., Gjorgjevikj, D. & Chorbev, I. 2009. A multi-class SVM classifier utilizing binary decision tree. Informatica, 33(2):233-241.

Maimon, O. & Rokach, L. (Eds.). 2005. Data mining and mining and knowledge discovery handbook (Vol. 2). New York: Springer.

Manley, B. & Sherry, L. 2008. Impact of ground delay program rationing rules on passenger and airline equity. 2008 Integrated Communications, Navigation and Surveillance Conference, 2-7 May 2008, Bethesda, MD, USA.

Miao, D. & Hou, L. 2004. A comparison of rough set methods and representative inductive learning algorithms. Fundamental Informaticae, 59(2-3):203-219.

Michell M. 1999 – 1997. An introduction to genetic algorithm. Singapore: McGraw-Hill.

Michie, D., Spiegelhalter, D.J. & Taylor, C.C. 1994. Machine learning, neural and statistical classification. [Online] Available at URL:http://minds.jacobsuniversity.de/sites/default/files/uploads/teaching/share/MantasRecommendedBook.pdf

Novianingsih, K. & Hadianti, R. 2014. Modeling flight departure delay distributions. 2014 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), 21-23 Oct. 2014, Bandung, Indonesia.

Parke, B., Mohlenbrink, C., Brasil, C., Speridakos, C., Yoo, H.S., Omar, F., Buckley, N., Gabriel, C. & Belfield, A. 2016. Reducing departure delays for adjacent center airports using time-based flow management scheduler: Checkbox ON or OFF? 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 25-29 Sept. 2016, Sacramento, CA, USA.

Peck, L. 2015. The impact of weather on aviation delays at O.R.Tambo International Airport, South Africa. University of South Africa.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., … Duchesnay, E. 2011. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12:2825-2830.

Platt J.C., Cristianini N. & Shwawe-Taylor J. 2000. Large margin DAG's for multiclass classification, 2000. Advances in Neural Information Processing Systems12(13):547-553

Platt, J.C. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines.

Qianya, L., Lei, W., Rong, F., Bin, W. & Xinhong, H. 2015. An analysis method for flight delays based on Bayesian network. The 27th Chinese Control and Decision Conference (2015 CCDC), 23-25 May 2015, Qingdao, China.

Quinlan, J.R. 1986. Induction of decision trees. Machine Learning, 1(1):81-106.

Renukadevi, N.T., Thangaraj, P. Dr. 2013. Performance evaluation of Svm-Rbf kernel for medical image classification. Global J. Comput. Sci. Technol. Graph. Vis., 13(4):15-20.

Roos, L.L., Stranc, L., James, R.C. & Li, J. 1997. Complications, comorbidities, and mortality: Improving classification and prediction. Health Serv Res, 32(2):229-238.

Rosen, A. 2002. Flight delays on US Airlines: The impact of congestion externalities in Hub and Spoke networks. Department of Economics Stanford University.

Roy, K., Chaudhuri C., Kundu, M., Nasipuri, M. & Basu, D.K. 2005. Comparison of the multilayer perceptron and the nearest neighbour classifier for handwritten digit recognition. Journal of Information Science and Engineering, 21(6):1245-1257.

Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E. & Suter, B.W. 1990. The multilayer perceptron as an approximation to a Bayes optimal discriminant function. Neural Networks. IEEE Transactions 1(4):226-298.

Russell, S. & Norvig, P. 2003. Artificial Intelligence: A modem approach. 2nd ed. EUA: Prentice Hall.

Saeys Y., Inza I. & Larra-naga P. 2007. A review of feature selection technique in bio-informatics. Bio-informatics, 23(19):2507-2517.

Sathya, R. & Abraham, A. 2013. Comparison of supervised and unsupervised learning algorithms for pattern classification. Int J Adv Res Artificial Intell, 2(2):34-38.

Saunders, M.N.K., Lewis, P. & Thornhill, A. 2012. Research methods for business students. 6th ed. Harlow, England: Pearson Education.

Schaefer, L. & Millner, D. 2001. Flight delay propagation analysis with the Detailed Policy Assessment Tool. IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236), 7-10 Oct. 2001, Tucson, AZ, USA.

Sharma, P. 2008. Artificial Intelligence., 1st ed.09 February. pp.64.

Shodhganga@INFLIBNET. 2002-2013. A reservoir of Indian theses @ INFLIBNET, Central University of Kashmir, Department of Management Studies. [Online] Available at http://shodhganga.inflibnet.ac.in/handle/10603/206091?mode=full [Accessed 19 November 2018].

Strydom, H. & Venter, L. 2002. Sampling and sampling methods. In: A.S de Vos, Research at grass roots for the social sciences and human service professions, (ed). Pretoria: Van Schaik.

Su M.C., Jean W.F. & Chang H.T. 1996. A static hand gesture recognition system using a composite neural network. Proceedings of IEEE 5th International Fuzzy Systems, 11 Sept. 1996, New Orleans, LA, USA.

Talwar, A. & Kumar, Y. 2013. Machine learning: An artificial intelligence methodology. International Journal of Engineering and Computer Science, 2(12):3400-3404.

Tan, P.N., Steinbach, M. & Kumar, V. 2006. Classification: Basic concepts, decision trees and model evaluation. Lecture notes: Introduction to data mining. pp.145-205.

Tang, C-H. 2011. A gate reassignment model for the Taiwan Taoyuan Airport under temporary gate shortages and stochastic flight delays. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 41(4):637-650.

Teknomo, K. 2004–2016. K-Means Clustering Tutorials. [Online] Available at http://people.revoledu.com/kardi/tutorial/kMean/WhatIs.htm and http://people.revoledu.com/kardi/tutorial/kMean/NumericalExample.htm [Accessed 18 August 2017).

Tiso Blackstar Group (Pty) Ltd. 2017. Economy about Reserve Bank leaves rates unchanged as economic outlook deteriorates, Johannesburg. [Online] Available at https://www.businesslive.co.za/bd/economy/2017-05-25-reserve-bank-leaves-rates-unchanged-as-economic-outlook-deteriorates/ [Accessed 21 August 2017].

Tiwari, R. & Singh, M.P. 2010. Correlation-based attribute selection using genetic algorithm. International Journal of Computer Application, 4(8):28-34.

Tourism Update. 2016. News for travellers still facing delays at OR Tambo, 2016, Southern & East African. [Online] Available at http://www.tourismupdate.co.za/article/113388/Travellers-still-facing-delays-at-OR-Tambo [Accessed 06 July 2017].

Traveller24. 2018. News on US alert for SA doe to OR Tambo crime wave, South Africa. [Online] Available at http://www.traveller24.com/News/Alerts/us-issues-safety-alert-for-sa-due-to-or-tambo-crime-wave-20170728 [Accessed 08 March 2018].

Trivedi, S. & Dey, S. Effect of attribute selection methods on machine learning classifiers for detecting email spams. RACS '13 Proceedings of the 2013 Research in Adaptive and Convergent Systems, 01 – 04 October 2013, Montreal, Quebec, Canada.

Turban, E. & Frenzel, L.E. 1992. Expert system and applied artificial intelligence. Prentice Hall Professional Technical Reference. pp.53-54.

Vafale, H. & De Jong, K. 1992. Genetic algorithms as a tool for feature selection in machine learning. Proceedings Fourth International Conference on Tools with Artificial Intelligence TAI '92, 10-13 Nov. 1992, Arlington, VA, USA.

Vapnik, V. 1995. The nature of statistical learning theory. Verlag: Springer.

Vapnik, V. 1998. Statistical learning theory. New York: John Wiley and Sons.

Wang, Y. & Hu, J. 2002. A machine learning based approach for table detection on the web. Proceedings of the 11th international conference on World Wide Web, 7-11 May 2002, Honolulu, Hawaii, USA.

Wei, W. & Hongshan, X. 2010. Impact of flights delays on productivity of Yangtze River Delta Airports. 2010 International Conference on Optoelectronics and Image Processing, 11-12 Nov. 2010, Haikou, China.

Wikimedia Foundation. 2017. O.R. Tambo International Airport. [Online] Available at https://en.wikipedia.org/wiki/O._R._Tambo_International_Airport [Accessed 16 August 2017].

Witten, I.H., Eibe, F. & Hall, M.A. 2011. Data mining: Practical machine learning tools and techniques. Diane Cerra, Elserivier.

Witten, I.H., Frank, E. & Hall, M.A. 2011. Data mining: Practical machine learning tools and techniques. 3rd ed. San Francisco, CA, USA Morgan Kaufmann.

Xing, Z. & Tang, Y. 2016. The model for optimizing airport flight delays allocation. 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 27-28 Aug. 2016, Hangzhou, China.

Yao, R., Jiandong, W. & Tao, X. 2010. Notice of retraction: A flight delay prediction model with consideration of cross-flight plan awaiting resources. 2010 2nd International Conference on Advanced Computer Control, 27-29 October 2010, Shenyang, China.

Yeganeh, H., Su, Z. & Chrysostome, E.V.M. 2004. A critical review of epistemological and methodological issues in cross-cultural research. Journal of Comparative International Management, 7(2):66–86.

ANNEXURE 3A

Flight	Carrier	Destination	Departure	Status
SA 40	South African Airways	(VFA) Victoria Falls	10:50 AM	Landed
SA 8809	South African Airways	(PTG) Polokwane	10:50 AM	Landed
SA 547	South African Airways	(DUR) Durban	10:55 AM	Cancelled
UA 7237 ^	United Airlines	(DUR) Durban	10:55 AM	Cancelled
SV 446	Saudia	(JED) Jeddah	11:00 AM	Landed
MN 925	Comair	(GRJ) George	11:05 AM	Landed
BA 6241	British Airways	(PLZ) Port Elizabeth	11:10 AM	Landed
CX 7321 ^	Cathay Pacific	(PLZ) Port Elizabeth	11:10 AM	Landed
SA 8841	South African Airways	(MQP) Nelspruit	11:10 AM	Landed
SA 8230	South African Airways	(APL) Nampula	11:10 AM	Landed
BP 212	Air Botswana	(MUB) Maun	11:10 AM	Scheduled
QR 4672 ^	Qatar Airways	(MUB) Maun	11:10 AM	Scheduled
FA* 230	Safair	(PLZ) Port Elizabeth	11:15 AM	Scheduled
BA 6285	British Airways	(VFA) Victoria Falls	11:25 AM	Landed
QR 6302 ^	Qatar Airways	(VFA) Victoria Falls	11:25 AM	Landed
SA 8214	South African Airways	(BEW) Beira	11:30 AM	En Route
SA 8260	South African Airways	(VNX) Vilanculos	11:30 AM	En Route
SA 8204	South African Airways	(POL) Pemba	11:30 AM	En Route
FA* 272	Safair	(DUR) Durban	11:30 AM	Landed
TK 38	Turkish Airlines	(MPM) Maputo	11:35 AM	Landed Delayed
SA 8859	South African Airways	(HDS) Hoedspruit	11:35 AM	Landed
JE 529	Mango	(PLZ) Port Elizabeth	11:40 AM	Scheduled
SA 2063 ^	South African Airways	(PLZ) Port Elizabeth	11:40 AM	Scheduled
SA 8306	South African Airways	(BBK) Kasane	11:45 AM	Landed
SA 8300	South African Airways	(MUB) Maun	11:45 AM	Landed
SA 8853	South African Airways	(PHW) Phalaborwa	11:45 AM	En Route
MN 103	Comair	(CPT) Cape Town	11:50 AM	Landed Delayed
AF 6646 ^	Air France	(CPT) Cape Town	11:50 AM	Landed Delayed
KQ 4634 ^	Kenya Airways	(CPT) Cape Town	11:50 AM	Landed Delayed

ANNEXURE 4A

Random sample data as received from ORTIA live departure flights website.

Live Flight Departures - JNB
Date: Fri 07-Jul-2017
Time Period: 9:00 PM - 12:00 AM ◉ Departures ○ Arrivals
Airport: (JNB) O.R. Tambo International Airport Johannesburg, ZA

Flight	Carrier	Destination	Departure	Status
SA 475	South African Airways	(ELS) East London	3:10 PM	En Route Delayed
JJ 6331 ^	LATAM Airlines Brasil	(ELS) East London	3:10 PM	En Route Delayed
SA 347	South African Airways	(CPT) Cape Town	3:10 PM	Scheduled Delayed
CA 7606 ^	Air China	(CPT) Cape Town	3:10 PM	Scheduled Delayed
ET 1100 ^	Ethiopian Airlines	(CPT) Cape Town	3:10 PM	Scheduled Delayed
JJ 6321 ^	LATAM Airlines Brasil	(CPT) Cape Town	3:10 PM	Scheduled Delayed
SA 8769	South African Airways	(UTN) Upington	3:20 PM	Landed
SA 1125	South African Airways	(MBD) Mmabatho	3:20 PM	Scheduled
MN 105	Comair	(CPT) Cape Town	3:25 PM	En Route
KQ 4638 ^	Kenya Airways	(CPT) Cape Town	3:25 PM	En Route
SA 1013	South African Airways	(BFN) Bloemfontein	3:30 PM	Landed Delayed
SA 8857	South African Airways	(PHW) Phalaborwa	3:30 PM	Landed
SA 8845	South African Airways	(MQP) Nelspruit	3:30 PM	Cancelled
SA 8777	South African Airways	(SIS) Sishen	3:30 PM	Cancelled
SA 8743	South African Airways	(PZB) Pietermaritzburg	3:30 PM	Landed
SA 1783	South African Airways	(GBE) Gaborone	3:45 PM	En Route Delayed
IB 7557 ^	Iberia	(PLZ) Port Elizabeth	3:45 PM	Scheduled
QR 6324 ^	Qatar Airways	(PLZ) Port Elizabeth	3:45 PM	Scheduled
QR 1367	Qatar Airways	(DUR) Durban	3:45 PM	Landed
AY 6009 ^	Finnair	(DUR) Durban	3:45 PM	Landed
SA 8164	South African Airways	(LUN) Lusaka	3:45 PM	En Route
SA 1509	South African Airways	(GRJ) George	3:50 PM	Scheduled Delayed
MN 805	Comair	(ELS) East London	3:55 PM	Landed
AF 6650 ^	Air France	(ELS) East London	3:55 PM	Landed
EY 1271 ^	Etihad Airways	(ELS) East London	3:55 PM	Landed
KQ 4602 ^	Kenya Airways	(ELS) East London	3:55 PM	Landed

ANNEXURE 4B

SQL script that converted the string format data into integer format data for ‘Time of the day’

```
ALTER TABLE [AdventureWorks2016].[dbo].[South African Airways Flight Departure Data]
 ADD TimeOfTheDay AS
 CASE
 WHEN [Departure] = '2:55 AM'
 THEN '0255'
 WHEN [Departure] = '4:45 AM'
 THEN '0445'
 WHEN [Departure] = '5:50 AM'
 THEN '0550'
 WHEN [Departure] = '5:50 AM'
 THEN '0550'
 WHEN [Departure] = '5:55 AM'
 THEN '0555'
 WHEN [Departure] = '6:10 AM'
 THEN '0610'
 WHEN [Departure] = '6:15 AM'
 THEN '0615'
  WHEN [Departure] = '6:20 AM'
 THEN '0620'
  WHEN [Departure] = '6:25 AM'
```

```
THEN '0625'
WHEN [Departure] = '6:30 AM'
THEN '0630'
WHEN [Departure] = '6:35 AM'
THEN '0635'
WHEN [Departure] = '6:40 AM'
THEN '0640'
WHEN [Departure] = '6:50 AM'
THEN '0650'
WHEN [Departure] = '6:55 AM'
THEN '0655'
WHEN [Departure] = '7:00 AM'
THEN '0700'
WHEN [Departure] = '7:10 AM'
THEN '0710'
 WHEN [Departure] = '7:25 AM'
THEN '0725'
WHEN [Departure] = '7:30 AM'
THEN '0730'
 WHEN [Departure] = '7:55 AM'
THEN '0755'
 WHEN [Departure] = '8:00 AM'
THEN '0800'
```

WHEN [Departure] = '8:10 AM'

THEN '0810'

WHEN [Departure] = '8:20 AM'

THEN '0820'

WHEN [Departure] = '8:25 AM'

THEN '0825'

WHEN [Departure] = '8:30 AM'

THEN '0830'

WHEN [Departure] = '8:40 AM'

THEN '0840'

WHEN [Departure] = '8:45 AM'

THEN '0845'

WHEN [Departure] = '9:10 AM'

THEN '0910'

WHEN [Departure] = '9:20 AM'

THEN '0920'

WHEN [Departure] = '9:25 AM'

THEN '0925'

WHEN [Departure] = '9:30 AM'

THEN '0930'

WHEN [Departure] = '9:40 AM'

THEN '0940'

WHEN [Departure] = '9:45 AM'

```
THEN '0945'
WHEN [Departure] = '9:55 AM'
THEN '0955'
WHEN [Departure] = '10:00 AM'
THEN '1000'
WHEN [Departure] = '10:05 AM'
THEN '1005'
 WHEN [Departure] = '10:10 AM'
THEN '1010'
WHEN [Departure] = '10:15 AM'
THEN '1015'
 WHEN [Departure] = '10:20 AM'
THEN '1020'
 WHEN [Departure] = '10:25 AM'
THEN '1025'
WHEN [Departure] = '10:30 AM'
THEN '1030'
 WHEN [Departure] = '10:35 AM'
THEN '1035'
 WHEN [Departure] = '10:40 AM'
THEN '1040'
 WHEN [Departure] = '10:45 AM'
THEN '1045'
```

```
WHEN [Departure] = '10:50 AM'
THEN '1050'
WHEN [Departure] = '10:55 AM'
THEN '1055'
 WHEN [Departure] = '11:00 AM'
THEN '1100'
WHEN [Departure] = '11:10 AM'
THEN '1110'
WHEN [Departure] = '11:15 AM'
THEN '1115'
WHEN [Departure] = '11:25 AM'
THEN '1125'
WHEN [Departure] = '11:20 AM'
THEN '1120'
WHEN [Departure] = '11:30 AM'
THEN '1130'
WHEN [Departure] = '11:45 AM'
THEN '1145'
WHEN [Departure] = '11:55 AM'
THEN '1155'
 WHEN [Departure] = '11:50 AM'
THEN '1150'
 WHEN [Departure] = '12:15 PM'
```

```
THEN '1215'
 WHEN [Departure] = '12:10 PM'
THEN '1210'
WHEN [Departure] = '12:25 PM'
THEN '1225'
WHEN [Departure] = '12:30 PM'
THEN '1230'
WHEN [Departure] = '12:40 PM'
THEN '1240'
WHEN [Departure] = '12:50 PM'
THEN '1250'
WHEN [Departure] = '12:55 PM'
THEN '1255'
WHEN [Departure] = '1:00 PM'
THEN '1300'
WHEN [Departure] = '1:10 PM'
THEN '1310'
WHEN [Departure] = '1:15 PM'
THEN '1315'
WHEN [Departure] = '1:20 PM'
THEN '1320'
WHEN [Departure] = '1:35 PM'
THEN '1335'
```

```
WHEN [Departure] = '1:40 PM'
THEN '1340'
WHEN [Departure] = '1:45 PM'
THEN '1345'
WHEN [Departure] = '1:50 PM'
THEN '1350'
WHEN [Departure] = '1:55 PM'
THEN '1355'
WHEN [Departure] = '2:05 PM'
THEN '1405'
WHEN [Departure] = '2:10 PM'
THEN '1410'
WHEN [Departure] = '2:15 PM'
THEN '1415'
 WHEN [Departure] = '2:30 PM'
THEN '1430'
WHEN [Departure] = '2:35 PM'
THEN '1435'
WHEN [Departure] = '2:50 PM'
THEN '1450'
WHEN [Departure] = '2:55 PM'
THEN '1455'
WHEN [Departure] = '3:10 PM'
```

```
THEN '1510'
WHEN [Departure] = '3:20 PM'
THEN '1520'
WHEN [Departure] = '3:30 PM'
THEN '1530'
 WHEN [Departure] = '3:40 PM'
THEN '1540'
WHEN [Departure] = '3:45 PM'
THEN '1545'
WHEN [Departure] = '3:50 PM'
THEN '1550'
WHEN [Departure] = '3:55 PM'
THEN '1555'
WHEN [Departure] = '4:00 PM'
THEN '1600'
WHEN [Departure] = '4:05 PM'
THEN '1605'
 WHEN [Departure] = '4:15 PM'
THEN '1615'
 WHEN [Departure] = '4:20 PM'
THEN '1620'
WHEN [Departure] = '4:25 PM'
THEN '1625'
```

```
WHEN [Departure] = '4:30 PM'

THEN '1630'

WHEN [Departure] = '4:45 PM'

THEN '1645'

WHEN [Departure] = '4:55 PM'

THEN '1655'

WHEN [Departure] = '5:00 PM'

THEN '1700'

WHEN [Departure] = '5:15 PM'

THEN '1715'

WHEN [Departure] = '5:05 PM'

THEN '1705'

WHEN [Departure] = '5:20 PM'

THEN '1720'

WHEN [Departure] = '5:25 PM'

THEN '1725'

WHEN [Departure] = '5:30 PM'

THEN '1730'

WHEN [Departure] = '5:40 PM'

THEN '1740'

WHEN [Departure] = '5:45 PM'

THEN '1745'

WHEN [Departure] = '5:50 PM'
```

```
THEN '1750'
WHEN [Departure] = '5:55 PM'
THEN '1755'
WHEN [Departure] = '6:05 PM'
THEN '1805'
WHEN [Departure] = '6:15 PM'
THEN '1815'
WHEN [Departure] = '6:30 PM'
THEN '1830'
WHEN [Departure] = '6:20 PM'
THEN '1820'
WHEN [Departure] = '6:25 PM'
THEN '1825'
WHEN [Departure] = '6:35 PM'
THEN '1835'
WHEN [Departure] = '6:45 PM'
THEN '1845'
WHEN [Departure] = '7:00 PM'
THEN '1900'
WHEN [Departure] = '7:05 PM'
THEN '1905'
WHEN [Departure] = '7:10 PM'
THEN '1910'
```

WHEN [Departure] = '7:20 PM'

THEN '1920'

WHEN [Departure] = '7:25 PM'

THEN '1925'

WHEN [Departure] = '7:40 PM'

THEN '1940'

WHEN [Departure] = '7:30 PM'

THEN '1930'

WHEN [Departure] = '7:45 PM'

THEN '1945'

WHEN [Departure] = '8:00 PM'

THEN '2000'

WHEN [Departure] = '8:15 PM'

THEN '2015'

WHEN [Departure] = '8:30 PM'

THEN '2030'

WHEN [Departure] = '8:35 PM'

THEN '2035'

WHEN [Departure] = '8:45 PM'

THEN '2045'

WHEN [Departure] = '8:50 PM'

THEN '2050'

WHEN [Departure] = '9:10 PM'

```
THEN '2110'
WHEN [Departure] = '9:15 PM'
THEN '2115'
WHEN [Departure] = '9:40 PM'
THEN '2140'
WHEN [Departure] = '9:45 PM'
THEN '2145'
WHEN [Departure] = '10:20 PM'
THEN '2220'
ELSE 'Unknown Time'
```

ANNEXURE 4C

Sample data after the data has been converted to nominal values so that it can be uploaded on WEKA for experiments.

Instances	**Month**	**Week Of Month**	**Day Of Week**	**Time Of Day**	**Target Concept**
1	Jul	Month Beginning	Monday	Morning	Delayed
2	Jul	Month Beginning	Monday	Morning	Delayed
3	Jul	Month Beginning	Monday	Morning	Delayed
4	Jul	Month Beginning	Monday	Morning	On Time
5	Jul	Month Beginning	Monday	Morning	On Time
6	Jul	Month Beginning	Monday	Morning	Cancelled
7	Jul	Month Beginning	Monday	Morning	On Time
8	Jul	Month Beginning	Monday	Morning	Delayed
9	Jul	Month Beginning	Monday	Morning	Cancelled
10	Jul	Month Beginning	Monday	Morning	Delayed

11	Jul	Month Beginning	Monday	Morning	On Time
12	Jul	Month Beginning	Monday	Morning	Delayed
13	Jul	Month Beginning	Monday	Morning	On Time
14	Jul	Month Beginning	Monday	Morning	On Time
15	Jul	Month Beginning	Monday	Morning	On Time
16	Jul	Month Beginning	Monday	Morning	On Time
17	Jul	Month Beginning	Monday	Morning	Delayed
18	Jul	Month Beginning	Monday	Morning	On Time