Data analytics in predicting flight delays

Student Supervisor

London

September 2022

# Abstract

This paper examines the role of flight delays in airlines and passengers’ life. Currently, life after the pandemic has increasingly been hard to acknowledge and the ongoing war has consequences that can be felt in the airline industry. These two factors have gone to an increased number of job openings, a lack of staff in airports, and Traffic Control and so to flight delays. The recent heat wave has globally made the job of airline employees even harder.

The main concern regarding delays is the money loss for both sides and this is what we are trying to prevent and minimize, given the economic crisis that is also predicted to follow.

The research is evidence that longer flights are more likely to be delayed than others.

An ethics statement was used to mention that this work reflects the author’s research and analysis truthfully and completely. For further information please go to Appendix E.

# Acknowledgements

I would like to express my deepest appreciation to my supervisor, Roushanak Rahmat for her knowledge and expertise shared with me. She always provided me with constructive criticism and feedback and was available to meet every time I requested. She guided me through every step of this work and was very patient and responsive.

Many thanks to my family for their emotional support. I believe they will support me in my dreams of starting a career in data analytics and be patient with me.

Table of Contents

[Abstract 2](#_Toc114171264)

[Acknowledgements 2](#_Toc114171265)

[List of Abbreviations 5](#_Toc114171266)

[List of Figures and Tables 7](#_Toc114171267)

[1. Introduction 8](#_Toc114171268)

[1.1. Problem statement 9](#_Toc114171269)

[1.2. Objectives 10](#_Toc114171270)

[2. Literature review 11](#_Toc114171271)

[2.1. Implications of the research 11](#_Toc114171272)

[2.2. The steps of Data Analytics 12](#_Toc114171273)

[2.3.1. K Neighbors Classifier (k-NN Classifier) 16](#_Toc114171274)

[2.3.2. Extreme Gradient Boosting (XGBClassifier) 18](#_Toc114171275)

[2.3.3. Support Vector Machines Classifier (SVMs) 18](#_Toc114171276)

[2.3.4. Logistic Regression 19](#_Toc114171277)

[2.3.5. Random Forest Classifier 20](#_Toc114171278)

[2.3.6. Neural Network Classifier 22](#_Toc114171279)

[2.3. Evaluation 23](#_Toc114171280)

[3. Methodology 26](#_Toc114171281)

[3.1. Data Gathering 26](#_Toc114171282)

[3.2. Data cleaning/ Pre-processing/ Data Preparation 26](#_Toc114171283)

[3.3. Machine learning modelling 29](#_Toc114171284)

[3.3.1. k-NN Classifier 33](#_Toc114171285)

[3.3.2. XGBClassifier 35](#_Toc114171286)

[3.3.3. SVM Classifier 36](#_Toc114171287)

[3.3.4. Logistic Regression 37](#_Toc114171288)

[3.3.5. Random Forest Classifier 37](#_Toc114171289)

[3.3.6. Neural Network Classifier 38](#_Toc114171290)

[3.4. Evaluation of result/ Quantitative analysis 38](#_Toc114171291)

[4. Discussion 41](#_Toc114171292)

[5. Conclusion 42](#_Toc114171293)

[Bibliography List 44](#_Toc114171294)

[Appendix A 46](#_Toc114171295)

[Appendix B 47](#_Toc114171296)

[Appendix C 48](#_Toc114171297)

[Appendix D 49](#_Toc114171298)

[Appendix E 51](#_Toc114171299)

# List of Abbreviation

International Civil Aviation Organization / ICAO

Alaska Airlines AS / ASA

American Airlines AA/AAL

Air Canada AC/ACA

Aeromexico AM / AMX

Continental Airlines CO / COA

Delta Airlines DL / DAL

FedEx FX / FDX

Hawaiian Airlines HA / HAL

Northwest Airlines NW / NWA

Polar Air Cargo PO / PAC

Southwest Airlines SW / SWA

United Airlines UA / UAL

United Parcel (UPS) 5X / UPS

Virgin Atlantic VS / VIR

VivaAerobús VB / VIV

WestJet WS / WJ

ATL - Hartsfield-Jackson Atlanta International Airport - Georgia

AUS - Austin-Bergstrom International Airport - Texas

BNA - Nashville International Airport - Tennessee

BOS - Boston Logan International Airport - Massachusetts

BWI - Baltimore-Washington International Thurgood Marshall Airport - Washington

CLT - Charlotte Douglas International Airport - North Carolina

DAL - Dallas Love Field - Texas

DCA - Ronald Reagan Washington National Airport - Arlington, Virginia

DEN - Denver International Airport - Colorado

DFW - Dallas/Fort Worth International Airport - Texas

DTW - Detroit Metropolitan Airport - Michigan

EWR - Newark Liberty International Airport - New Jersey

FLL - Fort Lauderdale–Hollywood International Airport - Florida

HNL - Daniel K. Inouye International Airport - Honolulu, Hawaii

HOU - William P. Hobby Airport - Houston, Texas

IAD - Dulles International Airport - Virginia

IAH - George Bush Intercontinental Airport - Houston, Texas

JFK - John F. Kennedy International Airport - Queens, New York

LAS - McCarran International Airport - Las Vegas, Nevada

LAX - Los Angeles International Airport - California

LGA - LaGuardia Airport - Queens, New York

MCO - Orlando International Airport - Florida

MDW - Chicago Midway International Airport - Illinois

MIA - Miami International Airport - Florida

MSP - Minneapolis–Saint Paul International Airport - Minnesota

MSY - Louis Armstrong New Orleans International Airport - Louisiana

OAK - Oakland International Airport - California

ORD - O'Hare International Airport - Chicago, Illinois

PDX - Portland International Airport - Oregon

PHL - Philadelphia International Airport - Pennsylvania

PHX - Phoenix Sky Harbor International Airport - Arizona

RDU - Raleigh-Durham International Airport - North Carolina

SAN - San Diego International Airport - California

SEA - Seattle–Tacoma International Airport - Washington

SFO - San Francisco International Airport - California

SJC - Norman Y. Mineta San Jose International Airport - California

SLC - Salt Lake City International Airport - Utah

SMF - Sacramento International Airport - California

STL - St. Louis Lambert International Airport - Missouri

TPA - Tampa International Airport – Florida

# List of Figures and Tables

[Table 1. Quantitative analysis for Classifier Models 38](#_Toc113547343)

[Figure 1. The steps of Data Analytics (G2, 2021) 13](#_Toc113880335)

[Figure 2. Cross-validation workflow (Ojala and Garriga, 2010) 14](#_Toc113880336)

[Figure 3. Steps of the k-NN Model (Navlani, 2018) 16](file:///C:\Users\maria\Desktop\Maria%20Moraru%2010103766.edited.docx#_Toc113880337)

[Figure 4. Steps of XGBoost Model (Shah,2020) 17](#_Toc113880338)

[Figure 5. Steps of SVM Model (Navlani, 2019) 18](#_Toc113880339)

[Figure 6. Logistic Regression Model (Remanan, 2018) 19](#_Toc113880340)

[Figure 7. Steps of Random Forest Model (Chauhan 2021) 20](#_Toc113880341)

[Figure 8. The steps for Neural Network (sci-kit-learn, 2022) 21](#_Toc113880342)

[Figure 9. Data Preview in Python 27](#_Toc113880343)

[Figure 10. Heatmap for numerical features 27](#_Toc113880344)

[Figure 11. Create a data frame and separate target values 28](#_Toc113880345)

[Figure 12. Airline Delays 29](#_Toc113880346)

[Figure 13. Day of Week Delays 29](#_Toc113880347)

[Figure 14. Departure Airport Delays 30](#_Toc113880348)

[Figure 15. Destination Airport Delays 30](#_Toc113880349)

[Figure 16. Flight length Delays 31](#_Toc113880350)

[Figure 17. Python code for classifier implementing the k-NN 32](#_Toc113880351)

[Figure 18. k-NN accuracy score Python code 33](#_Toc113880352)

[Figure 19. Elbow method for optimal K 33](#_Toc113880353)

[Figure 20. k-NN accuracy after the Elbow method 34](#_Toc113880354)

[Figure 21. Accuracy for XGB Classifier 35](#_Toc113880355)

[Figure 22. Python code and accuracy for Random Forest 36](#_Toc113880356)

[Figure 23. Python code for Neural Network 37](#_Toc113880357)

[Figure 24. Python Code for Quantitative Analysis 37](#_Toc113880358)

[Figure 25. Data Preparation 44](#_Toc113880359)

[Figure 26. Python Code for SVM 44](#_Toc113880360)

[Figure 27. Code and accuracy for Logistic Regression 45](#_Toc113880361)

[Figure 28. Python Code for Random Forest 46](#_Toc113880362)

[Figure 29. Python code for Neural Networks using Keras library 47](#_Toc113880363)

# Introduction

In the current era, where all processes are being digitalized, companies are using data analytics to test and predict the next steps to help them improve their activity and get the maximum outcome of it. Getting the best of their products and services can help them better to handle crises such as a pandemic, economical changes, or competitors. These days, data analysts use big data to handle clear and uncleaned data such as tables, text, and images. The process that was most used in ’00 is statistics. Contemporary, social media is one of the largest data generators alongside banking, transactions and online sources, healthcare. Upon collection of data, it needs to be managed to process, and prepare for modelling. The following step is analysing the data and examining and getting the results under patterns and relationships between features.

Airlines are using online content, databases, and recent big data to compare, criticize and improve their services. Data analytics can help the industry by reducing the time to prepare the flight while at ground level, making airspace management more efficient, and approaching each passenger individually by personalizing services offered and create them the best comfort. The process can help in airline marketing, customer care, operations, maintenance, competition, and pricing.

Contemporary, big data is not fully exploited in the airline industry because of a lack of experience and knowledge, time and human resources, programs, and tools.

The few studies made on this topic assign the project by creating a base for the original needs. The current research is designed to adopt a quantitative approach to predict flight delays. Its objective is to reduce the costs for low-cost airlines and the waiting time for passengers. The main questions to be answered are:

1. What are the biggest flight predictors that influence departure and take-off?
2. Where are the origins of the features to influence the airlines in working in proper conditions?
3. How can data analytics help in such a way as to reduce loss and lower the initial cost of working?
4. Can a global model be developed to be used by low-cost airlines and get classification outcomes for flight delays?
5. When more exactly in the process of a flight, the predicted model can be used?

Terrestrial traffic, maritime and air grew exponential in the past decades and brought the need for a supervised system for air traffic. The old surveillance technology such as primary surveillance radar (PSR) and secondary surveillance radar (SSR) could not handle the increase. As recently seen, new technologies such as automatic dependent surveillance-broadcast (ADS-B) came to light, and help flights transmit their current position and details, such as international civil aviation organization (ICAO) longitude, latitude, flight number and speed.

## Problem statement

The asked questions are about the data quality, features in the dataset, quality of the tools to use, and the if the research questions were applied.

Each row corresponds to a flight and in the airline section we can find the company name abbreviation, then comes the flight number and the time column is the flight duration. The airline's classification will be based on their punctuality and show in a chart if they had or not a delay in the past. This would provide the reader with a general image of the different airlines, the impact of the departure airport location, and if some carriers have made a habit out of it. The provided dataset is collected within a year and will have a look if any major change varies as time goes on.

Upon considering the main problem to be addressed, if a flight is going to be delayed or not, with the use of classification models, will have in mind airlines and departure airports, 6 days of the week to predict if a delay is going to be on the 7th day. Once the outliers will be removed, a comparison graph will be created, giving a general conception.

Before building a model, data is divided into **training and test** sets. A general model is then built on the training set and tested on the test set. Data is normalized and standardized and so feature scaling, normalization and standardization will not be processed.

Once a model is processed, a comparison will take place between the prediction and the actual value. The scoring model and testing it will also have a huge impact on the decision of keeping it for further enhancement and improvement of new data that comes in.

## 1.2. Objectives

We can build **a classification** model with high accuracy and a low error, as a division of the supervised classification models. The planned libraries to be exploited in this research will be matplotlib, seaborn, base map, pandas, NumPy, scikit-learn, SciPy, regression, and figures.

Commonly used by inexperienced engineers' outputs are graphics and this will take advantage of them in this research as well. Customer loyalty turns into the regular purchase of products and services and so to the company shareholders. As customers change their minds easily, we can’t give an exact prediction of the flights.

The exclusive codes used to identify an airline are the two-digit IATA codes. These codes are used commercially for things like airline tickets, luggage tags, and boarding permits and are specified by the International Air Transport Association (IATA).

ICAO oversees The Three-Digit Code for An Airline (International Civil Aviation Organization). These codes are employed for official functions, such as flight plans with ATC. They are primarily given out to airports without any commercial aviation service.

# Literature review

Big data analytics has constant updates on working processes, staff knowledge, and IT systems (Hausladen, 2020). Passengers have multiple ways of reaching their target destination. Carriers keep on updating their services on board by providing Wi-Fi and charging points, extended foot space, travel pillows, the menu on board, warm socks and scarves, not to mention their biggest advantage of a short time to destination. As more companies come to life, each of them takes the advantage of using big data to offer more and more services on board and beforehand in the airport through self-check-ins. Customers don't have to queue there and the airport space management is improved. Flights are nowadays being purchased through mobile versions of the websites where they can choose all the affiliate services with it. This is how airlines collect data about passengers, store them in the cloud or physical servers until big data comes in and gives them personalized offers and collects personal opinions about the flights (Park, 2019). Big data is a great help in passengers' behaviour and satisfaction as well. This is mainly to improve customer retention and to attract new customers from the competition (Park, 2019).

Given the recent situation where flights are constantly delayed and airlines are alimented, the crew staff has a big role in customer retention. Future sees that the number of budget companies will increase because of the increasing population in Asian countries (Liau, 2014).

In data analysis, the methodology of problems is taking place in three steps. The first step of the process is the assessment of the dataset, then comes in place the main one where the data analysts find the answers to the problem, and the final one is where the report is finalized (Ader, 2008). Herman Ader is going through the research cycle with several statistical techniques being used one at a time. The main concern in the process is to apply the best learning models to get the best results. To do that you first need to understand the whole process, take it step by step and inspire by other great works. The supervisor emphasises understanding the problem and why data analytics should be used.

## 2.1. Implications of the research

Existing research on big data in airlines focuses more on opportunities and challenges while the practical part of working with big data is giving a better future to the industry. Mainly they focus on the importance of visualization and delay prediction, rather than creating a model to be generally used in prediction at take-off and before that (RAM, 2016).

The main purpose of an airline is to take the passengers safely to the destination and provide great service at check-in and on board. To accomplish this, a flight delay must be announced well before the customer reaches the airport to prevent negative opinions that can be said to others too (Heidari, 2020).

Flight delays are the main cause of technical and financial issues for all airlines. In this article, the factors are analyzed to avoid delays and to offer some options on how to get rid of these hold-ups. (Zamkova,2017)

Ever since 1914 when the first passenger flight took place in a hydroplane, the chance of them being on spot started to decrease with the traffic being higher.

The multitude of studies on data analytics in flight delays cover creating a flight schedule to prevent, investigating flight delays with few data science techniques, and analyzing compensation expenses for the basic crew and passenger costs.

The decision made to analyze factors that influence flight delays and to create a model that best predicts is a flight is going to be delayed or not was made given the recent costs increase and longer flight delays happening over the summer.

## 2.2. The steps of Data Analytics

The processes needed to make up the data analysis process are part of big data. Identifying the crucial steps in a data analysis process is a no-brainer. To make sure that the data is properly evaluated and produces useful and actionable information, each step is equally crucial. Let's examine the five crucial actions that comprise a data analysis process flow.

1. We first need to define why we need data analysis to know what models to apply. In this research, we will use machine learning algorithms to automate the data analytics workflow.
2. Collecting data is the second step of data analysis. Carriers have passenger details when they book the flight. To purchase a ticket a customer has to create an account, where a user ID is given to log in with an email and password. Then the website or the app stores the device type and location plus the flight details. Once the payment is processed and the passenger is on board, all his collected data goes to a database that is managed by a business analyst. Our main source of collecting data is Kaggle and a US airline company.
3. Cleaning data is the most time spending step of all. Unnecessary data must be removed from the dataset; in this case, the ID column is not relevant. Once removed, each record needs to be checked for missing values, outliers or extreme values and wrong information entered. This step is part of the exploratory analysis that has its hands-on correlation map or heatmap.
4. Manipulating data is a business analyst’s job when he is encoding data. If some of the features are categorical, they must be encoded to numerical as models can be built with only numerical values.
5. Analyzing data is the main part. It is used in customer segmentation as part of marketing for loyal and potential customers. This is how a traveller's profile can be viewed by the marketing department. The carrier can then predict whether the customer buys tickets on a pattern, travelling for work, how many of the clients place orders, their average age, and percentages to be used in expanding the business.
6. Visualizing data is the output to be used on new data. It can include images, charts, or presentations. At this stage, the results will be interpreted and applied.

In Figure 1 we can see the main steps for data analysis.

Diagram

Description automatically generated

Figure 1. The steps of Data Analytics (G2, 2021)

Considering the tool we used to execute this machine learning project, google colab, Figure 1 presents are stages to be followed to get the predicted outcome.

The Python version used by google colab is currently 3.7.13. No hardware accelerator was used, as a central processing unit (CPU) only on a google colab managed cloud virtual machine. The CPU model name is Intel(R) Xeon(R) Processor with one core CPU @ 2.20GHz.

Firstly, we import the libraries to be used NumPy, matplotlib, seaborn, base map, pandas, sklearn, and SciPy. Then we import the dataset which was saved on the local machine. Within this dataset, we take the data for training our model and create a vector with the independent variables and a different vector with the dependent variable. The next step is to visualize the data in a heatmap to see the correlation between features. After that, we split the data into training and testing sets and feature scale the dependent variables to improve the performance of our model. The final step is to train (using the .fit function on the train set), test our model (using the .predict function on the test set), and visualize the training set results. Lastly, we evaluate the model through the confusion matrix

Whenever we perform any machine learning activity, we have our labelled data. We never pass our entire data to the model because when we are measuring the accuracy, we will not be able to say if the model is performing well or not.

A methodological error is using the prediction function on the same features and evaluating it on the same set of data. A model that simply repeats the labels of the samples it has just seen would have 100% accuracy and be unable to make any predictions about data that has not yet been seen. Overfitting is the definition of this process. It is customary to reserve a portion of the available data as a test set (X test, y test) when conducting a (supervised) machine learning experiment to avoid this problem. It should be noted that the term "experiment" does not just refer to academic purposes because machine learning experiments sometimes begin in commercial contexts as well. The standard cross-validation workflow in model training is shown in the Figure 2 flowchart.

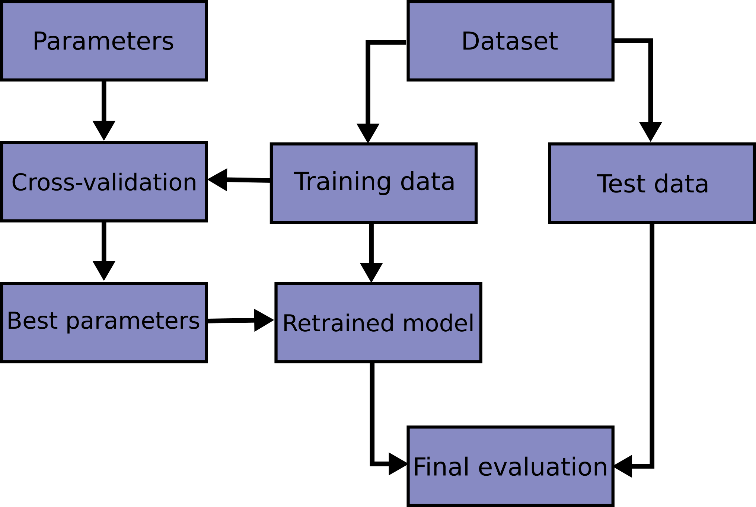


Figure 2. Cross-validation workflow – best parameters to retrain the model (Ojala and Garriga, 2010)

In Figure 2 we can see how overfitting comes in if the same date would be used for testing. The model would be built to repeat the labels of the samples that it has just seen, the accuracy would be 100%, and it would not be able to predict anything useful on data that has not been seen. The figure is showing how data is divided and considers the parameters (features) used. Performing cross-validation will assign all elements exactly once. Then the best parameters are chosen, and the model is retrained for final evaluation.

The library sci-kit-learn has the function train\_test\_split that randomly splits into training and testing sets. The most common ratio is 80% for training and 20% for testing and this is what was used in the current work.

The dataset was divided into training data and testing data. The data that the model will learn from is the training data. We will utilize the testing data to determine how well the model performs on unobserved data.

Our dataset can be easily divided into training and testing data using the Scikit-learn function "train test split." We divided it into 20% data for testing and 80% for training which is the most common one.

It requires 5 parameters for the train test split. The input and target data that we previously divided up are the first two parameters. The 'test size' value will then be set to 0.2. Accordingly, 20% of the total amount of data will be used for testing, leaving 80% of the data as training data for the model to use. We can duplicate our results by setting 'random state' to 1, which guarantees that we always receive the same split.

Our training split will represent the percentage of each value in the y variable if "stratify" is set to y. Setting "stratify" to y will guarantee that the random split has 25% of delayed flights and 75% of flights not delayed.

A loss function is very straightforward: it's a way to gauge how effectively your program models your dataset. Your loss function will produce a greater value if your forecasts were completely incorrect. If they're decent, it will produce a lower number. Your loss function will indicate whether you are making progress when you alter various aspects of your algorithm to try to enhance your model. The loss function outputs which features are dependent on predicting the model.

The accuracy of the final model will be the one helping customers' decision-making for flight weather on Google flights, Tripadvisor, or other flight search engines.

In the data analytics subsection, we used analyzed different methods. There are lots of different classifiers but the ones that were chosen, have their literature review below.

### 2.3.1. K Neighbors Classifier (k-NN Classifier)

Neighbours-based class is under the class of learning based on criteria, defined as learning in main instances because it is not building a general model, is just keeping the same entries from the training set. A key point is defined as the data class that has the most representatives among its nearest neighbours after classification is defined by voting against each close neighbour.

There are two different nearest neighbour classifiers that scikit-learn implements: KNeighborsClassifier learns based on the closest neighbours of each query point, where is an integer value that the user specifies. The user-specified floating-point value is used by RadiusNeighborsClassifier to implement training on the number of neighbours that are close to a certain set distance from each entry.



Figure 3. Steps of the k-NN Model – from initializing data to finding neighbours (Navlani, 2018)

The most popular method in KNeighborsClassifier is the -neighbours classification. The best amount to use depends heavily on the available data; generally, a greater value reduces noise effects but muddies up classification boundaries.

Radius-based neighbours classification in RadiusNeighborsClassifier may be a preferable option when the data is not consistently sampled. To classify locations in neighbourhoods with fewer residents, the user chooses a fixed radius, which reduces the number of nearest neighbours used. Due to the so-called "curse of dimensionality," this method loses effectiveness for high-dimensional parameter spaces.

The fundamental nearest neighbours categorization employs uniform weights, meaning that a query point's value is determined by the simple majority vote of its closest neighbours. It is preferable to consider that the nearest neighbour contributes more to the fit function.

### 2.3.2. Extreme Gradient Boosting (XGBClassifier)

A machine learning modelling complex and scalable is tree boosting. In this article, we introduce XGBoost, a measurable gradient boosting system used to solve problems on an edge. For sparse data, we suggest a novel sparsity-aware approach, and for approximation tree learning, a weighted quantile sketch. We also offer insights on cache access patterns, data compression, and sharding to create a scalable tree-boosting system. XGBoost scales beyond billions of examples while utilizing significantly fewer resources than current systems by leveraging these discoveries.

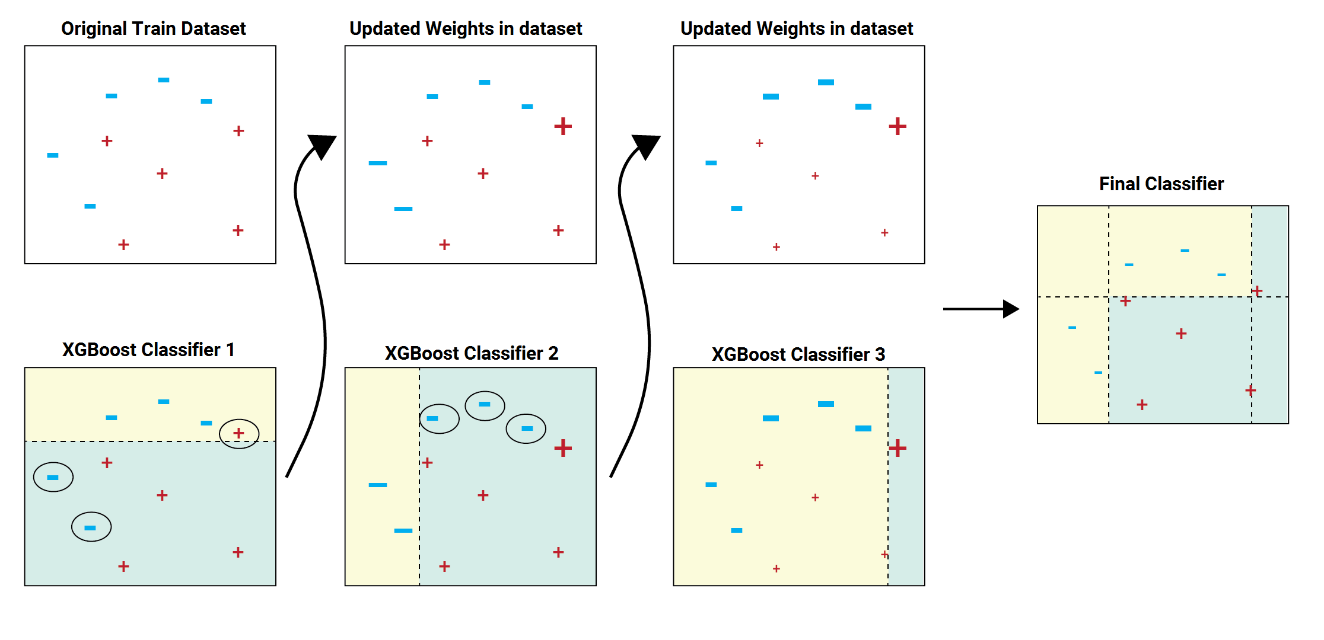


Figure 4. Steps of XGBoost Model – updated weights to the final classifier (Shah, 2020)

At each split point during training, the tree grower learns, based on the potential gain, no matter if the sample containing no value is taking left or right turn. The process of assigning them happens during prediction.

Splits on whether the feature value is absent or not can be performed when the missingness pattern is predictive.

Samples with missing values are mapped to the kid with the most samples if no missing values for a particular feature were discovered during training.

### 2.3.3. Support Vector Machines Classifier (SVMs)

Support vector machines (SVMs) can be used for classifying data, performing regression analysis, and identifying outliers under supervised learning.

The positive opinions of the researcher who has practised this technique, include efficiency in high-dimensional datasets, and even if the number of features exceeds the number of samples the output is considered useful.

It is not consuming CPU because it only uses a portion of the training points (known as support vectors) in the decision function.

Support vector machines have some negative feedback from their users, including:

Avoid over-fitting when the Kernel function is in place and regularisation terms if the number of dimensions is significantly higher than the number of samples.

Probability estimates are not directly provided by SVMs; instead, they are computed via an expensive five-fold cross-validation method (see Scores and probabilities, below). Versatile, the decision can specify a variety of Kernel functions, and it can take values of customer-specified values too.



Figure 5. Steps of SVM Model – classify using the hyperplane (Navlani, 2019)

Linear Support Vector Classification (SVC) is a technique able to perform multi-class and binary classification on a dataset.

### 2.3.4. Logistic Regression

Linear regression is used to solve regression problems when we are trying to predict a value whereas logistic regression is used to solve a classification problem and predicts discrete values. Based on the historical data, we let the system learn to predict whether this particular flight is going to be delayed or not.

If you try to draw a straight line to determine the probability, of whether a flight is going to be delayed or not, the predicted value can exceed 0 or 1. The probability is calculated between 0 and 1 using the Sigmoid curve as shown in equation (1) and figure (6). We use a different mechanism to find the value of probability.

(1)

The most common threshold value is 0.5, so if a probability is equal to or higher than 0.5 the used value is 1, and if the value is less than 0.5 we consider the value as 0.

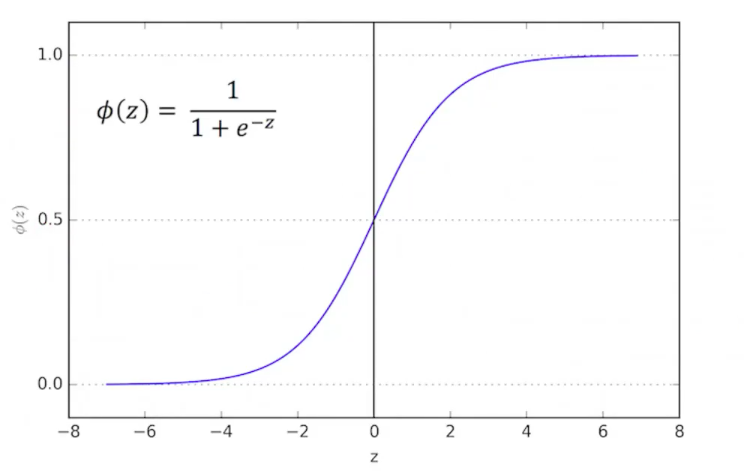


Figure 6. Logistic Regression Model - sigmoid "function" (Remanan, 2018)

In this model, a logistic function is used to simulate the probabilities describing the potential outcomes of a single experiment. With optional Elastic-Net regularisation, this solution can fit the binary, One-vs-Rest, or multinomial logistic regression.

### 2.3.5. Random Forest Classifier

A Random Forest is a meta estimator that considers the average of factors to improve predicted accuracy and reduce overfitting after giving numerous decision tree classifiers to different dataset subsamples. If bootstrap is set to true (the default value), the size of the sub-sample is given by the maximum samples argument; otherwise, trees are built using the entire dataset. We chose this method because is the most common one in predicting the discrete output.

The default loss function for the random forest is log loss. Please note that usually it is referred to as cross-entropy or information gain rather than log loss.

The Random Forest technique and the Extra-Trees method are two averaging algorithms based on randomized decision trees that are included in the sklearn.ensemble module. Both algorithms use perturb-and-combine methods that were created especially for trees. This means adding randomization to the classifier design results in the creation of a diverse group of classifiers. The average forecast of the individual classifiers is used to represent the ensemble prediction. For the same reason as other classifiers, forest classifiers also require the fitting of two arrays: a dense array X of shape (n samples, n features) with the training samples, and an array Y of shape (n\_samples,) with the target values (class labels) for the training set.



Figure 7. Steps of Random Forest Model – Features divided into trees and classes (Chauhan 2021)

Each tree in a random forest ensemble is built using a sample from the training set that was constructed with a replacement.

Additionally, the perfect split can be determined by either all input features or a randomly selected subset of size max features when dividing each node throughout the creation of a tree.

These two randomness sources serve to lessen the variance of the forest estimator. Individual decision trees do, in fact, frequently show substantial variance and a propensity for overfitting. The inserted randomness in forests creates decision trees with partially dissociated prediction errors. Some mistakes can be eliminated by averaging their predictions.

By merging several trees, random forests reduce variation, sometimes at the expense of a modest increase in bias. In actual use, the variance reduction is frequently large, producing an improved model all around.

Unlike the original publication, which allowed each classifier to vote for a single class, the scikit-learn method combines classifiers by averaging their probabilistic predictions.

### 2.3.6. Neural Network Classifier

A supervised learning system called a multi-layer perceptron (MLP) trains on a dataset to learn a function, where the number of input dimensions is the number of output dimensions. It can learn a non-linear function approximator for either classification or regression given a set of features and a target. It can contain one or more non-linear layers, shown as hidden layers, between the input layer and the output layer, which makes a clear difference from logistic regression. Figure 8 displays an MLP with one hidden layer and scalar output.

Shape

Description automatically generated

Figure 8. The steps for Neural Network – one hidden layer (scikit-learn, 2022)

The input layer is made up of a group of neurons that represent the input features. The values from the preceding layer are transformed by each neuron in the hidden layer using a weighted linear summation, followed by a non-linear activation function, such as the hyperbolic tan function. The values from the final hidden layer are sent to the output layer, where they are converted into output values.

The positive ideas about Neural Network (NN) are that it has the power to learn non-linear models in real-time, as fast as online goes through partial. fit. The negative side of it is the hidden layers with a non-convex loss function where there are more minimums. Therefore, different initialization chosen randomly can lead to different accuracy.

NN considers not just the default hyperparameters such as the number of hidden neurons, layers, and iterations. It also has a high variation after feature scaling.

## 2.3. Evaluation

The data analyst's main concern is to provide the best advice and improve the predicted model. Working with databases is a risky commitment. It is hard to explain it in writing, this is a complex process and not a step-by-step activity.

The predicted outcome sometimes can be similar. Getting documented before the twenty-first century was not an easy thing to do, but nowadays information is complex and is all about taking the time to read and understand.

We can evaluate the models through Confusion Matrix, accuracy score, and F1 Score.

The Confusion Matrix is the number of entries for which the predicted label matches the actual label represented by the diagonal elements, whilst the incorrect labels assigned by the classifier are represented by the off-diagonal elements. The confusion matrix's diagonal values should be high, indicating numerous accurate predictions.

The graphics display the confusion matrix both, as shown in equation (2), with and without class support size normalization (number of elements in each class). When there is a class imbalance, this form of normalization might be useful for providing a clearer understanding of which class is being misclassified.

Actual Values

Predicted Values (2)

The second metric is only used in classification problems to evidentiate the percentage of correct predictions, as shown in equation (3).

(3)

A classifier's **precision** can be thought of as a gauge of its accuracy. It is described for each class as the proportion of true positives to the total of true and false positives. In other words, "what percentage of all instances labelled as positive was accurate?”, as shown in equation (3).

(3)

The capacity of a classifier to accurately detect all positive cases is measured by the **recall**, which is also known as the completeness of the classifier, as shown in equation (4).

(4)

It is described as the ratio of true positives to the total of true positives and false negatives for each class. In other words, "what percentage of all instances that were genuinely positive were accurately classified?"

The **F1 score** is a weighted harmonic mean of recall and precision, with 1.0 representing the best result and 0.0 the lowest, as shown in equation (5).

(5)

F1 score typically performs worse than accuracy measures because they incorporate precision and recall into their computation. It is often recommended to compare classifier models using the weighted average of F1, rather than overall accuracy.

The number of real instances of the class in the given dataset is known as **support**. The requirement for stratified sampling or rebalancing may be indicated by unbalanced support in the training data, which may point to structural flaws in the classifier's reported scores. Support remains constant across models but diagnoses the evaluation procedure instead.

# Methodology

Under the supervised learning models, we can find classification and regression models. In the last one, the output values of the target class are continuous, and the target value must be integer or float.

The current problem is a classification one because the nature of the prediction is if a flight is going to be delayed or not. The best methods that can be used are k-NN, XGBGradient, Logistic Regression, SVM, and Random Forest.

## 3.1. Data Gathering

The given dataset has 539383 records and 9 features (8 input features and 1 target output). ID feature will be removed as it does not influence the prediction. The data contains the records of all the flights from a year and was collected from Kaggle. The purpose is to predict if a flight will be delayed or not by predicting a model and emphasizing the steps to get there. Second, if the flight is not going to be delayed at take-off but some of the dependent variables change at the last moment, the aim is to predict how long a delay is going to be. The code for the data shape is in Appendix A.

Regression can be used to predict continuous values like the length of the delay and the Classification model can be used to predict whether a flight is going to be delayed or not. Regression is the process of finding the correlation between the dependant variables (x1, x2, x3, x4, x5, x6, x7, x8) and the independent variable (y) which is the length of the delay. It is a mapping function that maps the 8 input variables to the continuous output variable. Classification will help to identify a mapping function that best divides the dataset into two different classes (delayed coded as 1 or not delayed coded as 0) based on the 8 given features.

## 3.2. Data cleaning/ Pre-processing/ Data Preparation

To transform raw feature vectors into a representation that is better suited for the downstream estimators, the sklearn. pre-processing package offers some common utility functions and transformer classes.

In general, standardizing the data set is advantageous for learning algorithms. Robust scalers or transformers are preferable if there are any outliers in the collection. In comparing the effect of different scalers on data with outliers, the behaviours of various scalers, transformers, and normalizers on a dataset with marginal outliers are emphasized.

Scaling and mean removal are examples of standardization. Many machine learning estimators used in scikit-learn frequently require dataset standardization; if the individual features do not more or less resemble standard normally distributed data, they may behave poorly: Gaussian with a mean of 0 and a variation of 1.

We frequently ignore the distribution's shape and simply adapt the data to scale by dividing non-constant features by their standard deviation and centre it by subtracting each feature's mean value.

For instance, many components utilized in the objective function of a learning algorithm may presumptively assume that all characteristics are centred around zero or that variance occurs in the same order for all features. A feature may dominate the distribution if its variance is orders of magnitude greater than that of other features.

There are 18 different airlines in the dataset**.** Label encoding was used to convert this categorical feature into a numerical one. Then the 18 airlines were converted as AA is now 1, AS is 2, and so on (see the List of Abbreviations).

There are 293 different airlines in the dataset. Label Encoder was used to convert the Airport From a categorical feature into a numerical one. Then the 293 airports were converted as ABE is now 0, ABI is 1, and so on (see the List of Abbreviations).

Label encoding was used to convert the airport from a categorical feature into a numerical one. Then the 293 airports were converted as ABE is now 0, ABI is 1, and so on (see The List of Abbreviations). As it can be observed, the same code was obtained for Airport From converted feature too.

Classification can perform only with numerical values so, for data preparation we will clean the data by one hot encoding, used when there is not a binary value, the three string features of Airlines, Airport from, and Airport to, check for null values, and replace the values with the mean, median or delete them (see Appendix F).

As part of the exploratory analysis will check for outliers and define the dataset.

There are 6585 unique flights in the dataset, each showing if a delay happened or not with that aircraft given the other features.

Once the data is cleaned and standardized, we can proceed to the next step where we share the data into subsets for training and testing.

The dataset has nine features carrier abbreviation, some flight, name of the leaving airport, name of the landing airport, the day of the week when the flight will take off, its time, the flight length and if it has been delayed (1) or not (0).

In Figure 9 we can see the first ten entries of the dataset.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 9. Data Preview in Python

One or more features will be irrelevant to our problem statement, we will remove them as well if they have a low impact on the prediction. The final decision will be made against the initial problem and consider dropping it or validating it. Other than that, can try to use small sample techniques like bootstrapping or exact tests. The first attempts were tried on a set of twenty samples.

Each value in a matrix is represented by a different hue in a heatmap, which is a graphical representation of data.

The primary message of the chart might be to draw attention to variable 1. However, if you're also interested in other variable variations, you should perform some normalization.

Graphical user interface, Teams

Description automatically generated

Figure 10. Heatmap for numerical features

As seen in Figure 10, the highest correlation is between the length of the flight and the day of the week when the flight is scheduled to take place.

We then store in the X variable all the numerical values (predictors), and in y the target value ‘Delay’.

A picture containing text

Description automatically generated

Figure 11. Create a data frame and separate target values

The categorical features were converted to numerical using a label encoder (see Appendix D) and their usability was proven to decrease the accuracy of the models. The decision to use only the numerical predictors was then made.

## 3.3. Machine learning modelling

In the machine learning subsection, we used different methods. The ones that were chosen are detailed below.

The following five methods were used because they are the easiest to use and understand and the supervisor suggested them.

In Figure 12 we can see the delays for each airline company in our dataset, using ggplot2 which offers a very consistent and understandable approach to plot the data. In addition to guaranteeing that each plot contains a few fundamental components, ggplot2's method of plotting also greatly simplifies the code, making it easier to read. However, for a frequent Python user, applying the grammar of graphics might be very difficult because of common charting libraries like matplotlib and seaborn lack consistent syntax.

Chart

Description automatically generated Figure 12. Airline Delays

Given the airline name, the dataset contains the most delayed flights for NorthWest airline (WN) 65k, and this airline is not recommended to fly with.

In Figure 13 we talk about the delays on each of the weekdays.

Chart, bar chart

Description automatically generated

Figure 13. Day of Week Delays

Given the day of the week, the dataset contains the most delayed flights on Wednesday (3rd day) 42k, and it is recommended to avoid flying on this day. Overall the day of the week does not have a significant impact on the delay.

In Figure 14 we talk about which taking-off airport has the most delays.

Chart

Description automatically generated Figure 14. Departure Airport Delays

Given the departure airport, the dataset contains the most delayed flights leaving Jackson Atlanta International Airport in Georgia (ATL) over 14k, and it is recommended to search for a nearby airport instead of leaving here.

In Figure 15 we talk about the delays taking place at landing.

**Chart

Description automatically generated**

Figure 15. Destination Airport Delays

Given the destination airport, the dataset contains the most delayed flights landing at Jackson Atlanta International Airport in Georgia (ATL) over 13k, and it is recommended to avoid this airport for both taking off and landing.

In Figure 16 we talk about the delays considering the length of the flight.

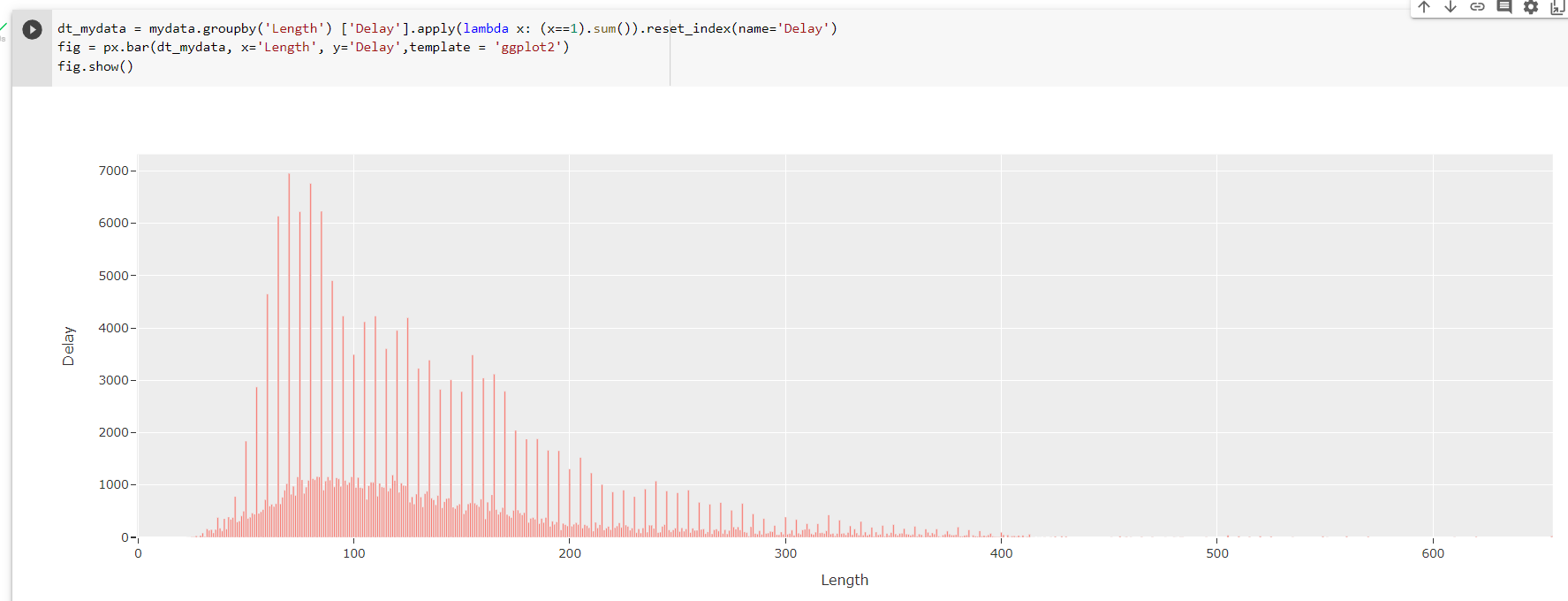
****

Figure 16. Flight length Delays

Given the length of the flight, the dataset has over 6.9k delayed flights for the longer flights of 70, and it is recommended to fly on short lengths. They record the highest delay.

Neural Networks are used in time constraining problems to increase company purposes and predict the fitted model. They process images in multiple areas. The information collected online can be best analyzed and design social trends. Online agents can create bots and fake updates that will have a huge impact on customers' decisions on purchasing a ticket if a company will promote multiple delays in the online environment.

The interface used is H2O ML because it is useful for accessing data, preparing, and model development. Compared to SparkML it has a better-defined confusion matrix and compared to BigML has multiple visualization tools.

We must save the training and testing data in a .csv file without creating the index column. It then creates the two clusters and stores them in h2o. The X variable (predictor) will now retain only the training columns, transform Delay to categorical values, store it in Y (response variable) and remove the categorical value Delay from the X variable. It took one hour to run all models.

Then the leaderboard was run on the cross-validation data to rank the models. The leaderboard is stored at aml and predictions are generated on the test set, but this can be made on the leader model object directly.

The test dataset was then combined with the prediction and a flight delay rate was predicted for each flight.

The following Classification model will be used for the current prediction.

### 3.3.1. k-NN Classifier

The K Neighbors Classifier will be used as an algorithm to predict if a new flight is going to be delayed or not. A supervised machine learning model is called k-Nearest-Neighbors (k-NN). When a model learns from labelled data, this is referred to as supervised learning. An input set of objects and an output set are both inputs to a supervised learning model. The model is then trained using that data to figure out how to translate inputs into desired outputs, allowing it to develop prediction skills for data it hasn't yet seen. This is the easiest method to use, and understand, and was chosen because is a robust method.

First I dropped the string value columns and stored the 4 numerical features in the X variable. The Y variable contains just the Delay feature. There are no null values in the dataset. Python has a machine learning library called Scikit-learn. In this research, we'll demonstrate how to create a k-NN model with Scikit-learn to determine if a flight is delayed or not.

We divided into 20% data for testing and 80% for training as is the most common one and printed the variables to use for k-NN to see if they are the same size.

We will create a new k-NN classifier and set 'n\_neighbors' to 3. This is equivalent to that if at least 2 out of the 3 nearest points to a new data point are no delay flights, then the new data coming in will be labelled as ‘no delay’.

The model must be trained. We will use the 'fit' function to fit our new model to the training data by passing in the training data as parameters.

Text

Description automatically generated

Figure 17. Python code for classifier implementing the k-NN

The 'predict' function on our model can be used to make predictions on our test data after the model has been trained. As was evident while reviewing 'y' earlier, 0 denotes the absence of delay, and 1 denotes the presence of a delay.

Let's check the model's accuracy on the entire test set now. To determine how well our model can make predictions, and correspond with the actual outcomes, we will utilize the 'score' function and pass our test input and target data.

Graphical user interface, text

Description automatically generated

Figure 18. k-NN accuracy score Python code

Our model is approximately 50% accurate. Although it's a good beginning, we will examine how to improve model performance below.

To get the best K the ***Elbow method*** was used. This showed that the optimal K is 5. After performing the model with K=5, the accuracy grew to 58%.

Chart

Description automatically generated

Figure 19. Elbow method for optimal K

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 20. k-NN accuracy after the Elbow method

The loss function for k-NN cannot be minimized during training as there isn't one. In actuality, this algorithm has never been trained. For k-NN, the only "training" that takes place is memorizing the data (making a local copy) so that you may perform a search and a majority vote during prediction. Technically, there is no optimization because no function is fitted to the data (it cannot be trained using gradient descent).

The code for k-NN is in Appendix A.

### 3.3.2. XGBClassifier

If the SciPy environment is up and running, pip can be used to quickly install XGBoost. We will load the data from the file in this step and get it ready for an XGBoost model's training and evaluation. Importing the classes and functions we wish to use will be the first step.

The X and Y data must be divided into a training dataset and a test dataset. The test set will be used to generate fresh predictions from which we can assess the performance of the XGBoost model after it has been trained on the training data. The split method from the scikit-learn library will be used for this. To guarantee that we always get the same data split each time this example is executed, we also give a seed for the random number generator.

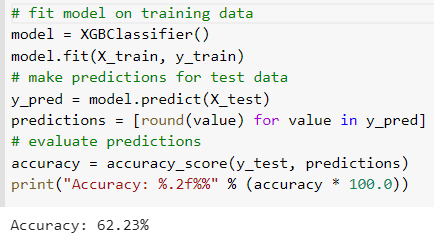


Figure 21. Python Code and Accuracy for XGB Classifier

Given the capabilities of the model and the moderate complexity of the problem, 62% is a good accuracy score on this issue.

The loss function used for predicting probabilities for binary classification problems is "binary: logistic". If the hyperparameter named objective is not given a value, the *XGBClassifier* automatically chooses one of the loss functions considering what it has seen so far.

### 3.3.3. SVM Classifier

We chose this method because it works better with large datasets.

Support vector machines (SVM) Classification methods are memory-efficient due to the usage of a portion of the training points (known as support vectors) in the decision function. Different Kernel functions can be given for the decision function, making it versatile. There are common kernels available, but one can be defined as well.

The model uses Support Vector Classification, a linear kernel, and a random state. It is then fitted to the training subset and predictions are made on the testing subset.

The default loss function for SVM is hinge loss. Its value shows a 72% error, so the model is not as good as the previous ones. The code for SVM is in Appendix A.

### 3.3.4. Logistic Regression

Ignoring its name, logistic regression is a linear model for classification as opposed to regression. In the literature, logit regression, maximum-entropy classification, and the log-linear classifier are also used to refer to logistic regression. The logistic function is used to simulate the probabilities describing the potential outcomes.

The loss function for logistic regression is log loss while the loss function for linear regression is squared loss.

The Logistic regression model is 55% accurate, which is less good than the previous ones. The code for Logistic Regression can be found in Appendix B.

### 3.3.5. Random Forest Classifier

This is the most commonly used machine learning algorithm due to its amazing performance in classification predictive modelling.

******From the sklearn library, we imported four classes to use on this model. It contains just a few parameters to provide an ensemble of decision tree algorithms. Each tree creates a sampling with replacement from a sample of the training subset.

**Text

Description automatically generated** Figure 22. Python code and accuracy for Random Forest

As seen in Figure 22, the model is using three repeats and 9 folds in the stratified k-fold cross-validation. The accuracy considers the mean and standard deviation for all folds and repeats. The model used only default values for this algorithm.

The number of trees in the forest was not set as the default value of one hundred is widely used.

### 3.3.6. Neural Network Classifier

Neural networks are trained on the subset of R4 as there are 4 features to consider and one target. It learns a non-linear function approximator for the classification of the output of R2 (the output targets are 0 and 1). We then have 5 hidden layers in Figure 23.

Feature scaling was not used here as the model is sensitive to it.

Text

Description automatically generated

Figure 23. Python code for Neural Network

As seen in Figure 23, the first parameter given to train is the solver and was set for weight optimization of the method, alpha was set to 0.00001 to strengthen the regularization term, five hidden layers in a tuple, and the random state was set to 1 to initialize the model.

## 3.4. Evaluation of result/ Quantitative analysis

We will evaluate the models' trough accuracy score, Confusion Matrix, and Classification report (precision, recall, and F1 Score).

All the different scores for all the Classifiers that were used are presented in the below table as quantitative analysis.

The code for the three evaluation methods can be seen in Figure 24.







Figure 24. Python Code for Quantitative Analysis

The model evaluation methods that were used are available in the sklearn library. All methods have two parameters, the 20% subset to be used for testing and the prediction of the test results.

In Table 1 we can easily spot the highest scores for each model.

Table 1. Quantitative analysis for Classifier Models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Confusion Matrix | Precision | Recall | F1- Score | Accuracy |
| k-NN |  | 67% for 0  61% for 1 | 71% for 0  56% for 1 | 69% for 0  58% for 1 | 58% |
| XGBoost |  | 64% for 0  59% for 1 | 73% for 0  49% for 1 | 68% for 0  54% for 1 | 62% |
| Logistic Regression |  | 58% for 0  52% for 1 | 76% for 0  32% for 1 | 66% for 0  40% for 1 | 56% |
| SVM |  | 74% for 0  46% for 1 | 9% for 0  96% for 1 | 16% for 0  62% for 1 | 55% |
| Random Forest |  | 61% for 0  58% for 1 | 77% for 0  40% for 1 | 68% for 0  47% for 1 | 95% |
| Neural Networks |  | 55% | 71% | 71% | 55% |

Based on the outputs in Table 1 we can say that Random Forest had the highest accuracy and this is why we choose to use this model for further predictions.

The confusion matrix consists of four different categories based on predicted values and actual values. Predicted values can be positive or negative and actual values can be true or false. As so, all the analyzed models have their highest values for the true positive component, the second highest value for the false negative component, the second lowest for the true negative component, and the lowest value for the false negative component.

The true positive cases predicted for k-NN (39410) are higher than the true negative ones and the false negative predicted cases (23544) are higher than the false positive ones and we conclude that this is a good model.

The true positive cases predicted for SVM (39243) are higher than the true negative ones and the false negative predicted cases (20598) are higher than the false positive ones and we conclude that this is a good model.

The true positive cases predicted for Random Forest (42312) are higher than the true negative ones and the false negative predicted cases (25523) are higher than the false positive ones and we conclude that this is a good model.

The balanced F-score is sometimes referred to as the F-measure or F1 score. It can be described as a harmonic mean of precision and recall, with the best value being 1 and the poorest being 0. Precision and recall both contribute in the same percentage to the F1 score. Precision is the percentage of how exact the classifier is, and although they all have similar values, we can conclude that the prediction for a flight not being delayed is more exact than the prediction for a flight being delayed. Recall notifies how complete the classifier is. We can see that our classifiers correctly found all positive components for class 0 better recall than for class 1.

The average F1 score for every class in the multi-class and multi-label situation, with weighting based on the average parameter. We use this to compare classifier models, not accuracy. We can conclude that all three components of the classification report show higher scores for the 0 class.

The support element of the classification report gives the diagnosis of it. There is no high discrepancy between classes 0 and 1 and this means that the above scores are strong and do not require sampling or rebalancing.

The Random Forest’s model accuracy was 95%. Given the fact that is the highest score, we will choose this one as the best one to present further. This is the best model because it has the highest accuracy.

Accuracy defines the percentage of accurate predictions. As seen in Table 1, the highest value for the number of correct outcomes divided by the number of total predictions is given by Random Forest.

# 4. Discussion

We used five different algorithms, and we choose them because they are the most robust ones. After all, they work better on large datasets with lots of features too.

The highest accuracy was proven by the SVM Classification model as this is effective in high dimensional space and uses a subset of points from the training set in the decision function.

Far from that, we could find Random Forest, which is easier to understand than the previous one, uses the fit function on one hundred trees and sub-samples, no size of the tree was set initially, and this was consuming a lot of memory. At every branch, the split is different every time we ran the model as the model is permuting them randomly if the improvement of the criterion is the same for some splits that were included when looking for the best split. It is necessary to fix random states to have predictable behaviour during fitting. The accuracy of Random Forest was 95%.

To improve memory consumption, the size of the tree was set to a lower value and the output was not modified considerably.

The difficult part was the exploratory analysis of the data and standardising it to create a model. It will be made easier with the tools offered by python. The models that were planned to be built helped to predict flight delays. Any change to the default values could lead to improving the models’ performance.

Upon trying to plot the first charts, the observation made was that the count was for all the flights and not just for the ones that were delayed, which makes the case of this research.

When improving the models, the start was given by the default values and then they were changed to get better performance.

# 5. Conclusion

The initial problem of not knowing if a flight is going to be delayed or not at take-off was considered to be of high importance for airlines and passengers.

The statement was debated by multiple data analysts and commented on by students, graduates, and trainees in the field. They concluded with models being able to predict flight delays based on their features, both at take-off and landing.

The present dissertation analyzed the eight features of flights and concluded that out of the six tested models, the best model to predict delays for new flights is by using the Random Forest model for this application. We gathered data, cleaned and prepared our data, processed, trained, tested, and analyzed.

Machine learning modelling was a very useful tool in the process of getting the best results. The outcome of the prediction showed how by using the four most important features in the dataset, so less information, we can give a new flight a tag, in the same way how we can recommend an airport or not based on the delays that took place in a period.

For this application, Random Forest has the best performance. If new data comes in, the model will be applied, but it depends on the nature of the data. This can have a different impact and different machine learning models are suitable for different applications. If new data comes in, it can work well on Random Forest, this can be tuned, or other models can have a better performance. If the new data is similar to the one in the current dataset, the predicted model is the suitable one. This is why the testing was performed on data that the model has not seen during training.

In the given timeframe these were the most on-hand algorithms to follow because the nature of the problem was a classification one, the six models were chosen. Also, they are the most common, best and well-known methods.

By now, the six models were studied, the nature of the data was understood, and the Random Forest is responding and dealing better with this data. Because we have different categories, Random Forest is easier to classify them in different branches.

The plans for this research include improving the model, improving Neural Network as well, because of the potential of this and maybe merging some of these algorithms with each other.

Another way to improve the model’s performance is by getting more data and trying to make the data more complicated because in reality there are more features to think about.

The central argument was to provide airlines with a better understanding of how to predict if a flight is going to be delayed with so less information. The final reflection is that working with data analytics is challenging and requires time and patience. Machine learning on flight delays predicted with the help of a deep learning algorithm like a random forest that the flight are going to be delayed or not on all the given records.

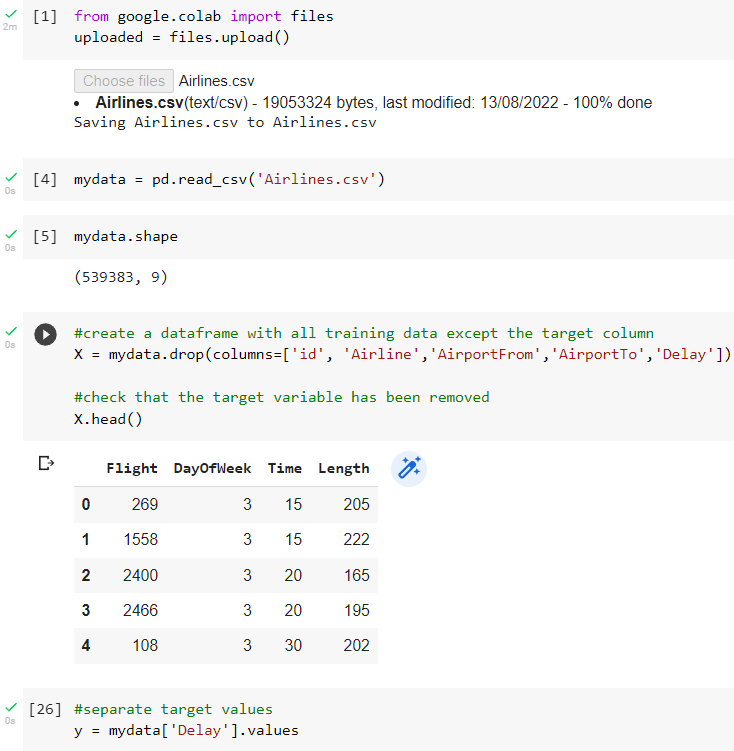
The analytics for flight delays analyzed in this work can also be found on GitHub at ……………..

# Bibliography List

1. HAUSLADEN, Iris; SCHOSSER, Maximilian. Towards a maturity model for big data analytics in airline network planning. *Journal of Air Transport Management*, 2020, 82: 101721.
2. PARK, Eunil. The role of satisfaction on customer reuse to airline services: An application of Big Data approaches. *Journal of Retailing and Consumer Services*, 2019, 47: 370-374.
3. PARK, Eunil, et al. Determinants of customer satisfaction with airline services: An analysis of customer feedback big data. *Journal of Retailing and Consumer Services*, 2019, 51: 186-190.
4. LIAU, Bee Yee; TAN, Pei Pei. Gaining customer knowledge in low-cost airlines through text mining. Industrial management & data systems, 2014.
5. Kaggle (2017), Predicting flight delays, Available at: <https://www.kaggle.com/code/fabiendaniel/predicting-flight-delays-tutorial/notebook> (Accessed on 10/07/2022)
6. RAM, Jiwat; ZHANG, Changyu; KORONIOS, Andy. The implications of big data analytics on business intelligence: A qualitative study in China. Procedia Computer Science, 2016, 87: 221-226.
7. ADER, Herman J. Phases and initial steps in data analysis. Chapter, 2008, 14: 333-356.
8. M. Heidari and S. Rafatirad, "Using Transfer Learning Approach to Implement Convolutional Neural Network model to Recommend Airline Tickets by Using Online Reviews," 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA, 2020, pp. 1-6, DOI: 10.1109/SMAP49528.2020.9248443.)
9. G2 (2021) What Is the Data Analysis Process? 5 Key Steps to Follow. Available at: https://www.g2.com/articles/data-analysis-process (Accessed: 30 August 2022).
10. Machine Learning in Python (2022). Available at: <https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation> (Accessed: 30 August 2022).
11. [Scikit-learn: Machine Learning in Python](http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), Pedregosa, *et al.*, JMLR 12, pp. 2825-2830, 2011.
12. Machine Learning Mastery (2022) Your First Deep Learning Project in Python with Keras Step-by-Step. Available at: <https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/> (Accessed: 30 August 2022).
13. Towards Data Science (2018) Building a k-Nearest-Neighbors (k-NN) Model with Scikit-learn. Available at: https://towardsdatascience.com/building-a-k-nearest-neighbors-k-nn-model-with-scikit-learn-51209555453a (Accessed: 1 August 2022).
14. Medium (2020) Evaluating a Random Forest model. Available at: https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56 (Accessed: 1 August 2022).
15. [Scikit-learn: Machine Learning in Python](http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), Available at: <https://scikit-learn.org/stable/about.html#citing-scikit-learn> (Accessed: 1 August 2022).
16. Navlani A (2018), KNN Classification Tutorial using Scikit-learn, digital image, Available at: https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn(Accessed: 20 August 2022).
17. Navlani A (2019), Support Vector Machines with Scikit-learn Tutorial, digital image, Available at: https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python (Accessed: 20 August 2022).
18. Chauhan A (2021), Random Forest Classifier and its Hyperparameters, digital image, Available at: https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6 (Accessed: 20 August 2022).
19. Shah I (2020), Introduction to XGBoost in Python, digital image, Available at: <https://blog.quantinsti.com/xgboost-python/> (Accessed: 08 September 2022)
20. Remanan S (2018), Logistic Regression: A Simplified Approach Using Python, digital image, Available at: https://towardsdatascience.com/logistic-regression-a-simplified-approach-using-python-c4bc81a87c31 (Accessed: 08 September 2022)
21. Bittner, Michael, Paul Meltzer, and Jeffrey Trent. "Data analysis and integration: of steps and arrows." Nature genetics 22.3 (1999): 213-215.

# Appendix A

Uploading the data from the local machine and reading it as .csv, data shape, data preparation and division can be found in the first part. Secondly, we can find the libraries, and fit and predict function codes for SVM.



Text

Description automatically generated

Figure 25. Data Preparation

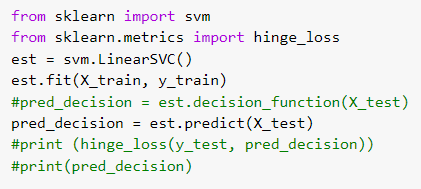


Figure 26. Python Code for SVM

# Appendix B

We can find the libraries, fit and predict function codes, and two evaluation forms for Logistic Regression.

***Graphical user interface, text, application

Description automatically generated***

Figure 27. Code and accuracy for Logistic Regression

# Appendix C

We can find the fit and predict function codes, and the accuracy score for Random Forest.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Figure 28. Python Code for Random Forest

# Appendix D

We can find the fit and predict function codes, and the accuracy score for Neural Network. Two different ways were tried, first with Keras library and second with sklearn.

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Figure 29. Python code for Neural Networks using Keras library

2.

Graphical user interface, text, application

Description automatically generated

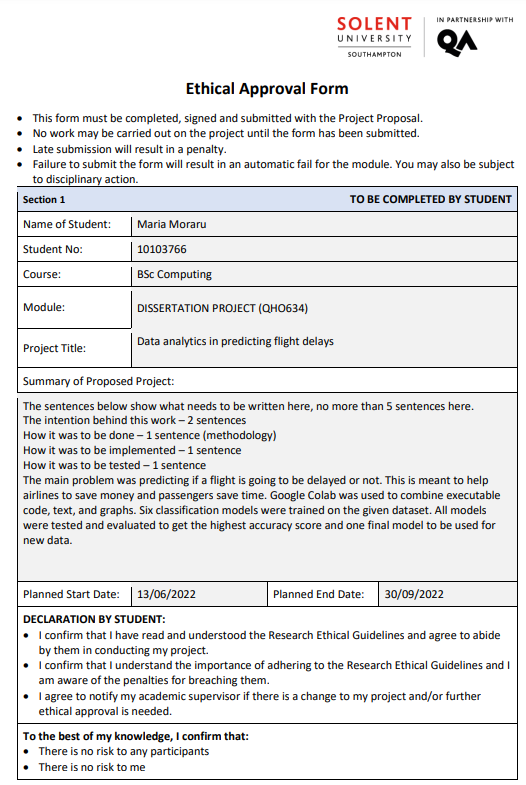
Graphical user interface, text, application

Description automatically generated

Figure 30. Python code for Neural Networks using sklearn library

# Appendix E

The form is a confirmation of the ethical guidelines and mentions the supervisor's approval.



Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated

Figure 31. Ethical Form

# Appendix F

The label encoding for categorical data.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Figure 32. Label encoding