UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

Máster Universitario en Ingeniería Aeronáutica

TRABAJO FIN DE MÁSTER

Prediction of Flight Delays using Machine Learning

Autor: Marcos ALBENDEA MEMBRILLA
Tutor: Alan DOMÍNGUEZ MONTERO

Curso 2022-2023

TITLE: Prediction of Flight Delays using Machine Learning

AUTHOR: Marcos Albendea Membrilla

PROJECT TUTOR: Alan Domínguez Montero

DEGREE: Master of Science in Aerospace Engineering

UNIVERSITY: Universidad Europea de Madrid

COURSE: 2022 - 2023

"*La imaginación es más importante que el conocimiento. El conocimiento es limitado. La imaginación da la vuelta al mundo.*" - Albert Einstein

Resumen

El objetivo principal de este proyecto es pronosticar retrasos en vuelos comerciales, los cuales pueden producirse debido a múltiples factores. Los efectos negativos de los retrasos en los vuelos son principalmente económicos para los viajeros (alteración de los horarios de los pasajeros), para la industria de las aerolíneas (disminución de la demanda de pasajeros debido a la reducción de la reputación de la aerolínea) y, también, para las autoridades aeroportuarias. A su vez, producen un impacto económico indirecto, debido a que un sistema de transporte aéreo ineficiente requiere un aumento de la cantidad de personal necesario. Además, en el ámbito de la sostenibilidad, la existencia de retrasos en vuelos tiene incluso el potencial de perjudicar al medio ambiente debido a un aumento del consumo de combustible y de las emisiones de gases, afectando negativamente al efecto invernadero. Por lo tanto, estas consideraciones muestran la importancia de la estimación de retrasos en vuelos.

El análisis de la predicción de los retrasos en vuelos es útil para establecer un estudio preciso del rendimiento de determinadas aerolíneas y aeropuertos, permitiendo procesos cruciales de toma de decisiones de cada participante clave en el sistema de transporte aéreo. En concreto, la base de datos analizada procede de la Oficina de Estadísticas de Transporte (BTS) del Departamento de Transporte de Estados Unidos (DOT) y representa los retrasos y cancelaciones de vuelos nacionales durante el año 2015.

El objetivo de este análisis es desarrollar un motor de aprendizaje automático en dos fases que pueda predecir con exactitud el retraso en la llegada de un vuelo en minutos utilizando vuelos en tiempo real. En primer lugar, un modelo clasificador estima si el vuelo se retrasará o no, y, si el retraso tiene lugar, entonces, un modelo de regresión predice el retraso de la llegada en minutos. Estas predicciones se evalúan mediante la utilización de Python, un lenguaje de programación de alto nivel, ideal para el propósito de este proyecto.

Abstract

This project's main objective is to forecast airline commercial flight delays that can be produced by many factors. Negative effects of flight delays are mostly financial for travellers (disrupting air passengers schedules), the airline industry (decrease of passengers demand due to the reduction of the airline's reputation), and, also, airport authorities. It also produces and indirect economical impact, as an inefficient air transportation system requires a rise in the amount of required staff. Additionally, in the area of sustainability, it even has the potential to harm the environment due to an increase in fuel consumption and gas emissions, negatively affecting the greenhouse effect. Therefore, these considerations show how important and crucial it has become to estimate flight delays.

The flight delay prediction analysis will be useful for providing a precise study of the performance of specific airlines and airports, enabling crucial decision-making processes from every key participant in the air transportation system. In particular, the analysed dataset proceeds from the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) and represents national flight delays and cancellations during the year 2015.

The goal of this analysis is to develop a two-stage Machine Learning engine that can accurately forecast upcoming flight's arrival delay in minutes using real-time flights. First, a classifier model predicts whether the flight will be delayed or not, and if the delay occurs, then, a regression model predicts the arrival delay in minutes. These predictions are evaluated using Python, a high-level, general-purpose programming language, ideal for the purpose of this project.

Acknowledgements

I would like to thank the following people for helping me to complete this project, and without whom I would not have made it through my Masters Degree:

My family, for all their support. For encouraging me to complete my training for the next stage of my life and always pursue my dreams.

My life partner, Sara Boullón Cano, for her understanding at difficult times, for being the support point I needed and for all her love and affection.

Samuel Motos Olmedo, Rafa Pozuelo de Frutos and Laura Molina Torres. My incredible teammates and friends.

Finally, Alan Domínguez Montero, for fostering my learning and development as academic tutor.

Contents

List of Figures

List of Tables

1

Introduction

1.1. Problem's Introduction

The majority of airlines experience flight delays on a daily basis. There are many different reasons why delays occur and they can range from a few minutes to many hours.

Air carriers, Air Traffic Control (ATC), weather issues, late-arriving aircraft, or security are just a few of the factors or parties pertaining to the Air Transportation System (ATS) that might cause a flight delay. Moreover, seasonal variations across the different months of the year might affect flight delays. This last particularity is a consideration that might be taken into account when the feature set's analysis is performed.

Particularly economically, flight delays affect travellers (varying passengers schedules), air carriers (passengers demand declines as a result of airline's reputational damage) and airport authorities.

It also has an indirect economic effect because a less effective Air Transportation System (ATS) necessitates an employees hiring increment. Furthermore, in terms of sustainability, it also has the potential to hurt the environment because of a rise in fuel consumption and the corresponding gas emissions, negatively affecting the zero emissions target established for the near future. These factors thus demonstrate how critical and important it has become to estimate flight delays.

The main goal of this project is to predict airline commercial flight delays. To accomplish this, predictive analysis is performed using a variety of statistical algorithms from Machine Learning (ML), which enable the analysis of both recent and historical data from a flight delay database to make predictions or simply assess upcoming delays. Python, a high-level, all-purpose programming language that is perfect for this project's goals, is used to assess these predictions.

Depending on the type of existing data and the data that will be produced, a wide range of models are applicable in Machine Learning (ML) approaches.

The performance of particular airlines and airports can be precisely studied using this forecast. The analysed dataset, which includes aircraft delays and cancellations from the year 2015 and contains up to 5.819.079 different domestic flights from the United States of America, proceeds from the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS). Also, the analysis of predicted flight delays will assist in critical decision-making from all significant stakeholders in the aviation industry, including the passengers.

1.2. Objectives and Scope

The main objective and motivation of this project is the prediction of flight delays which can be caused by several reasons. Performing an Exploratory Data Analysis (EDA) on the dataset is a prior step to building a model for prediction. The application of a Machine Learning model allows to verify the viability and functionality of Machine Learning to make this kind of predictions. It is possible to observe and make predictions with Machine Learning methods based on modern statistics, which enable algorithms to learn from previous data. For this purpose, the applied dataset is utilised as preliminary input for making the predictions with the application of the Machine Learning techniques. The dataset is analysed and the key attributes are extracted (Chapter 2). Besides, another important purpose of this project is the development of an applicable procedure in the flight delay prediction field for upcoming projects.

The project's structure relates to the steps that need to be taken for the attainment of a methodology for the prediction of flight delays using Machine Learning. Chapter 2 is the Exploratory Data Analysis (EDA), which analyses the applied dataset from which the data is selected and extracted. It is important to note that this data must be treated (Data Filtering and Data Preprocessing) previously to the implementation of algorithms. Next Chapter, exposes the different Machine Learning methods and its application on the thesis. A detailed analysis of the different Machine Learning techniques is shown in Chapter 3. Finally, Chapters 4 and 5 collect the analysis of results and outcomes of the two-stage Machine Learning Engine and the future work and conclusions extracted from this project.

In relation to the project's scope, due to the wide variety of Machine Learning algorithms, its use will me limited to those algorithms that have proven to be useful for the aimed objective. Previously to choosing between the different Machine Learning algorithms, the data must be filtered and treated to generate functions and utilities to apply the Machine Learning models in the simplest and more efficient way possible.

This project's forecast of flight delays makes it possible to evaluate the effectiveness of various airlines and airports (origin and destination airports). Particular airlines can cause delays, which can be precisely measured using features like the scheduled and the on-time performance, identifying flight delays trends. The obtained prediction, besides other benefits, has positive economical repercussions for every actor belonging to the Air Transportation System (ATS), such as passengers, airlines and airports. These factors thus demonstrate how critical and important it has become to estimate flight delays.

As the applicability of this project for further data analysis of flight delays is one of the primary goals, the source code divided by scripts is permanently contained on the following GitHub repository: https://github.com/MarcosAlbendeaMembrilla/TFM.git

GitHub is a code hosting platform (open source) for the control of Git versions and collaboration projects. It allows different users to work on projects simultaneously. It is mainly used for the source code creation for computer programs.

1.3. State of Art

Traveling on an airplane's commercial flight, can bring along one of the inevitable issues of flying, the delays. There could be a number of causes for the delays, including the late arrival of the plane, queuing for take-off, harsh weather that results in hour-long delays or even cancellations. Whatever the cause, delays are quite annoying and either directly or indirectly harm commercial air traffic users.

Domestic flight delays cost the airline industry, travellers, and other sectors of the economy 32.9 billion dollars annually, according to a US Federal Aviation Administration (FAA) estimation from the year 2010. More than half of the costs are paid by the passengers, who not only loose time waiting for departure, but they also miss connecting flights and spend money on food and accommodation while waiting for their flight.

Using statistics from 2007, the air transportation delays reduced the nation's gross domestic product by 4 billion dollars in that period.[4]

Making a comparison between actual and past data extracted from the Bureau of Transportation Statistics (BTS) website [5], it can be observed how the numbers have improved:

	2008	**2016**
On Time	76.04%	81.42%
Delayed	21.75%	17.17%
Cancelled	1.96%	1.17%
Diverted	0.25%	0.24%

Table 1.1: BTS Flight Data Comparison between 2008 and 2016

Taking these ratios into consideration, the number of on-time flights increased by 5.38%, or an average of 0.6725% per year, between 2008 and 2016. There is, however, still a great deal of space for development.

Insightful conclusions can be formed from the large amount of data on flight travel (16+ million flights annually in the U.S. alone), helping us to better understand flight delays. Due to the sector's expansion, the system must be stretched to its full capacity in order to meet all the demand. Moreover, the quantity of data makes it possible to create and train models that may be able to anticipate these delays.

With new techniques based on modern statistics, it is now possible to observe and forecast events. These techniques are based on Machine Learning (ML), a branch of statistics that allows algorithms to gain knowledge from data. A model can be created using Machine Learning (ML) that analyses all feature sets and discovers the patterns that can be drawn from them. These models are created by Machine Learning algorithms and can be compared to determine the best outcomes (predictions). In the last years, Machine Learning (ML) has altered how we interpret data. It has given us the ability to recognise, identify, and foresee possible outcomes that were previously undetected. Nevertheless, the full possibilities that Machine Learning (ML) has to offer in flight delay prediction are not fully explored yet, and it can be very useful for both business and individual travellers. Consequently, this is a topic that can be very significant, allowing us to draw potentially significant conclusions from the available data.

In the actual big data era, a lot of data is gathered, enabling the inference of meaningful results and making reasoned business decisions. On the other hand, unprocessed data needs to be examined. A thorough Exploratory Data Analysis (EDA) process must be carried out in order to get the most out of the raw data. For a complex, well-structured Machine Learning (ML) model to perform well, this is the first step. As the amount of data increases, it gets more difficult to analyse and explore the data, employing data analysis and data visualization tools.

A look into the future reveals that there are many interconnected factors that affect delay reduction. A remedy for one kind of delay will have a significant impact on the others, creating a compound effect that will enable more effective operations and benefit all involved parties.

One of the solutions stated by Michael Baiada (retired pilot and President of ATH Group), which is likely to be implemented in the near future, asserts that the carriers are responsible for fixing the problem of flight delays. The number of delayed flights would significantly decrease if carriers permitted aircrafts to fly faster and consume more fuel on late-departing flights. [1]

1.4. Introduction to Machine Learning (ML)

Machine Learning is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed." – Arthur Samuel (1959).

In the past few years there has been a development of the Machine Learning techniques. Its application has revolutionised many fields such as publicity and marketing market, mobile phones applications or innovations such as autonomous driving.

For example, if you have a dataset of images of animals, and you want to create a software or an application that can recognise and differentiate them. The first thing that you must implement is interpret the images as a set of feature sets.

Prior to the existence of Machine Learning, each image would be transformed to a vector of features. Then, traditionally, some rules or methods must be written down in order to get computers to be intelligent and detect the different animals.

But, as it can be guessed, it needed a lot of rules, highly dependent on the current dataset, and not generalised enough to detect out-of-sample cases.

This is when Machine Learning entered the scene. Using Machine Learning allows to build a model that looks at all the feature sets, and learns the pattern that can be extracted from them. It is a model built by Machine Learning algorithms. It detects without explicitly being programmed to do so. So, Machine Learning algorithms, inspired by the human learning process, iteratively learn from data and allow computers to find hidden insights. These models help us in a variety of tasks, such as object recognition, summarization, recommendation, and so on.

Currently, these technologies have an enormous potential to improve all the applications they are applied to, achieving incredible results and great reliability, being most of them available for free and in Open Source.

In brief, Artificial Intelligence (AI) tries to make computers intelligent in order to mimic the cognitive functions of humans. So, Artificial Intelligence is a general field with a broad scope including: computer vision, language processing, creativity, and summarization.

Machine Learning is the branch of Artificial Intelligence (AI) that covers the statistical part of Artificial Intelligence. It teaches the computer to solve problems by looking at hundreds or thousands of examples, learning from them, and then using that experience to solve the same problem in new situations.

Deep Learning is a very special field of Machine Learning where computer can actually learn and make intelligent decisions on their own. Deep Learning involves a deeper level of automation in comparison with most of the Machine Learning algorithms.

One of the key sectors where this technologies can be applied, is the flight delays, as it affects the commercial air traffic users either directly or indirectly. The sector's growth forces to squeeze the system up to the maximum grade in order to face all the demand.

1.5. Reasons for Flight Delays

Flight delays are quite common, happening to the majority of airlines every day. Delays can last from very few minutes to various hours and can be produced by different causes. According to the Bureau of Transportation Statistics (BTS), about 20% of all flights are delayed by 15 minutes or more. [6]

The delays cause negative effects, specially economical, on air traffic clients, airlines and also airport authorities. Besides, it can produce environmental issues, with the increase in fuel consumption and carbon emissions produced by this delays. This consequence must be closely taken into account nowadays, considering the Net Zero Carbon Emission Challenge in which the aeronautical industry has embarked, obliged by the regulations established by the authorities.

Airlines report the causes of delay and cancellations to the Bureau of Transportation Statistics (BTS) and they can be divided into the following categories [6]:

- **Air Carrier**: The reason of the delay was due to circumstances within the airline. For example, caused by preparation of the aircraft (refueling, aircraft cleaning, safety and security checks and clearance, baggage loading), late crews or maintenance.

- **Extreme Weather**: Extreme Meteorological conditions (actual or forecasted) such as typhoon, hurricane or snowstorm can either cancel or delay a flight. Delays caused by adverse weather can last for hours or even days, depending of the severity of the circumstances. Also, convective weather inside and outside the terminal area and high winds can be a possible cause of delays.

- **Air Traffic Control (ATC)**: Air passenger traffic has grown enormously in the last years and most of the traffic is concentrated around hubs (great traffic volume), remaining safety as the top priority for air traffic controllers. Sometimes, delays are caused because the airspace is too busy and landing sequence is prioritised, maintaining the aircraft separation. Another reason for flight delays can be infrastructure, as radar coverage can be insufficient in some areas, causing larger periods of time between operations (landings and take-offs) to assure safety.

- **Late-Arriving Aircraft**: As the airline's aircrafts have tight schedules, they can arrive late from a previous route and affect the following flight, causing it to depart late. This is due to airline's priority to maintain as many aircaft in the sky as possible for economical reasons.

- **Security**: Delays can be caused by evacuation of a terminal, re-boarding of aircraft due to a security breach or long queues at screening areas.

The flight delay prediction from this project permits to analyse the performance of the different airlines and airports (origin and destination airports). Individual airlines can cause delays and they can be measured with accuracy, making use of measures like the schedule and the on-time performance, discovering patterns of flight delays.

The average daily delay is also influenced by seasonal effects, and it may vary between different times of the year.

1.6. Python for Machine Learning

Python is a popular and powerful general-purpose programming language that has recently emerged as the preferred language among data scientists. It was created in 1991 by Guido Van Rossum and it's popularity among the years is based on its use simplicity and great applicability to different tasks as Web Programming, Scientific Computation and Machine Learning. It is a "scripts" language which is interpreted, not compiled, being an object's oriented language which needs a virtual environment for its execution. [9]

In this project, Python is used with an Integrated Development Environment (IDE) called Jupyter Lab [3], which executes Jupyter Notebook allowing the Python's code implementation by sections, using "Markdown" code in order to write paragraphs or adding photographs or equations. Its use is based on the fact that it is a standard in Machine Learning or Data Science with Python. Jupyter Notebook can directly execute Python code in cells, observing the outcome, showing code's execution, as it is shown hereunder:

```
[]: print("Prediction of Flight Delays using Machine Learning")
```

```
Prediction of Flight Delays using Machine Learning
```

Jupyter Lab [3] is used as Integrated Development Environment (IDE) with various libraries used to treat data and apply Machine Learning algorithms. A library is a set of functions or objects that can be used combined and that enhance the functionality of Python and simplify programming. There are a series of necessary libraries that are already implemented in Python and must be installed in order to develop the Machine Learning tool, standing out the following ones:

- **Numpy** [11]: It is one biggest numeric computation libraries in Open Source and is part of Scipy. The functions that Numpy offers are related to mathematics in general and allow to create vectors or n-dimensional arrays in Python, enabling operations with them. Furthermore, it includes the majority of common use mathematical functions ranging from statistics to quantum computation. It enables you to do computation efficiently and effectively.

- **Scipy** [13]: SciPy is a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics and much more. SciPy is a good library for scientific and high-performance computation.

- **Pandas** [10]: It is a very high-level Python library that allows to treat data as cells. Its advantage lies in the ease of use, its great variety of commands and its capacity to perform on databases with a great volume of information. During the creation of the tool, it will be used to group and filter the obtained data and to export it. It has many functions for data importing, manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

- **Matplotlib** [2]: Matplotlib is a very popular plotting package that provides 2D plotting as well as 3D plotting. It contains plenty of functions that serve for the representation of data obtained with Pandas or Numpy.

- **Scikit-learn** [8]: Scikit-learn is a collection of algorithms and tools for Machine Learning. Scikit-learn is a free Machine Learning library for the Python programming language. It has most of the classification, regression and clustering algorithms, and it's designed to work with the Python numerical and scientific libraries, Numpy and Scipy.

  On top of that, implementing Machine Learning models with Scikit-learn is really easy with a few lines of Python code. Most of the tasks that need to be done in a Machine Learning pipeline are implemented already in Scikit-learn, including, pre-processing of data, feature selection, feature extraction, train/test splitting, defining the algorithms, fitting models, tuning parameters, prediction, evaluation and exporting the model.

- **Seaborn** [15]: Seaborn is a data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

It is necessary to remark that Machine Learning algorithms benefit from standardisation of the dataset. If there are some outliers, or different scales fields in the data set, it is recommended to fix them. To do this, the pre-processing package of Scikit-learn provides several common utility functions and transformer classes to change raw feature vectors into a suitable form of vector for modelling.

In order to train the model and then, separately, test the model's accuracy, the dataset must be splitted into train an test sets. Scikit-learn can split arrays or matrices into random train and test subsets, in one line of code. Next, the algorithm can be setup. There are different metrics to evaluate the model's accuracy, for example, using a confusion matrix to show the results.

It is important to point out that the entire process of a Machine Learning task can be done simply in a few lines of code using Scikit-learn. Though it is possible, it is much more complicated to do all of this using Numpy or Scipy packages. And of course, it needs much more coding if you use pure Python programming to implement all of these tasks. It stands out that all the algorithms applied in this project are contained in the Scikit-learn library.

1.6.1. Types of Machine Learning Algorithms

Machine Learning is a modality of computing sciences that aims to generate algorithms without having to write them explicitly from a set of data that shares similar characteristics. The Machine Learning techniques can be classified depending on the different parameters:

- If they are trained or not under human supervision.
- If they can learn or not from the data on the way.
- If they work comparing similar data to new data or try to find data patterns and create predictive models.

There are three types of Machine Learning algorithms taking into account if they are trained or not under human supervision:

- **Supervised Learning**: In this type of algorithms, models are trained with human supervision. This is done starting from previous results called *labels* that are supplied to the algorithm together with the data characteristics called *features*. From this, the algorithm will create a model that can predict the *labels* from some *features*.
- **Unsupervised Learning**: In this type of algorithms, models are only trained with *features*, without giving the model any expected results.
- **Semi-Supervised Learning**: It is a mixture form the two previous types of Machine Learning algorithms, there is data that contains *labels* and there is data that doesn't.

Supervise means to observe and direct the execution of a task, project or activity. In order to supervise a Machine Learning model, the model must be "teached". That is, we load the model with knowledge so that we can have it predict future instances. We teach the model by training it with some data from a labelled dataset. It's important to note that the data is labelled.

The names of the different columns are called *Attributes*. The columns, which include the data, are called *Features*. If you plot this data and look at a single data point on a plot, it will have all of these attributes. That would make a row on this chart, also referred to as an observation.

Focusing on the values of the data, there are two kinds of values. The first one is numerical, which is the most commonly used data when dealing with Machine Learning. The second kind is categorical, non-numeric, because it contains characters rather than numbers.

There are two types of Supervised Learning techniques, Classification and Regression. Classification is the process of predicting a discrete class label or category. While Regression is the process of predicting a continuous value as opposed to predicting a categorical value in Classification.

In Unsupervised Learning, the model is not supervised, letting the model work on its own to discover information that may not be visible to the human eye. This means that the unsupervised algorithm trains on the dataset, and draws conclusions on unlabelled data. Generally speaking, Unsupervised Learning has more difficult algorithms than Supervised Learning, since the information about the data is practically or completely unknown, as well as the expected outcomes.

Dimension reduction, Density estimation, Market Basket analysis and Clustering are the most widely used Unsupervised Machine Learning techniques. Dimension reduction and/or feature selection play a large role in this by reducing redundant features to make the classification easier.

Market Basket analysis is a modelling technique based upon the theory that if you buy a certain group of items, you're more likely to buy another group of items.

Density estimation is a very simple concept that is mostly used to explore the data to find some structure within it.

Clustering is one of the most popular Unsupervised Machine Learning techniques used for grouping data points or objects that are somehow similar. In fact, Clustering is mostly used for summarising, discovering the structure of the dataset and the detection of anomalies.

To sum up, the biggest difference between Supervised and Unsupervised Learning is that Supervised Learning deals with labelled data while Unsupervised Learning deals with unlabelled data. In comparison to Supervised Learning, Unsupervised Learning has fewer models and fewer evaluation methods that can be used to ensure the accuracy of the outcome of the model. To conclude, Unsupervised Learning creates an environment which is less controlled, as the machine is creating outcomes for the user.

However, the application of this algorithms isn't direct because there are some steps that need to be taken in advance to obtaining functional models. A crucial step in the application of these algorithms is the previous treatment of the data, filtering and cleaning them from unnecessary characteristics. Furthermore, re-scaling the data is key for an optimal execution, as well as controlling the hyper parameters, which enable the control during the learning process.

1.6.2. Data Filtering and Preprocessing

In order for the results of the model application to be precise and reliable, the data provided to the algorithm must be processed correctly. Data treatment consists in the following steps:

- **Data Filtering**: This step is taken when the data is downloaded from an external source and the integrity of the data is not assured. In addition, it is used when a series of magnitudes relative to these data remain to be calculated. The integrity of the data makes reference to the fact that all the data is complete or it belongs to the parameters previously defined. Within Data Filtering, the generation of relative magnitudes from *features* included in the data or the elimination of the unnecessary characteristics that can deteriorate the performance of the algorithm, are also included.

- **Data Preprocessing**: Once the data is filtered, it is going to be preprocessed. To do this, the data is splitted to generate two different sets of data. One of the sets of data will contain the 80% of the original data and it will be intended for the model's training. The other set of data will have the remaining 20% and it will be used to prove the algorithm's prediction reliability. The reason of doing this is due to the possibility of Machine Learning algorithms having problems, like overfitting of parameters. This causes that the results obtained from the algorithm can be skewed.

  The following step to take is to scale the data according to a standard distribution, changing the values depending on the mean and the standard deviation. Another form of scaling the data is to assure that the *labels* are delimited between two different values.

2

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method of data analysis that uses visual tools to examine data sets and summarise their key features. By identifying patterns, recognising anomalies, testing hypotheses, and verifying assumptions, it improves our understanding of the variables in a data feature set and the connections between them. With the aid of summary statistics and graphical representations, it can also assist in determining whether the statistical procedures considered for data analysis are appropriate.

Exploratory Data Analysis (EDA) techniques were created by American mathematician John Tukey in the 1970s and are a frequently utilised approach in the data discovery process.

The analysed dataset proceeds from a faithful government agency website available online on the internet. The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) is an agency which is responsible for providing the statistics from air traffic delay in the United States of America. It records the on-time performance of domestic flights (departure and landing airports are located on U.S soil) from large carriers. [7]

The main objective of Bureau of Transportation Statistics (BTS) is that this data would be available for the use of flight delay data analysts and costumers. BTS data are used in the Air Travel Consumer Report and the Domestic Airline Fares Consumer Report. To be precise, this dataset represents flight delays and cancellations from the year 2015. [7]

The objective is to study the provided dataset, containing up to 5.819.079 different national flights in the US in the year 2015 and their causes for delay, diversion and cancellation. The data comes from the U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS).

Previous to Machine Learning (ML) model's creation, Exploratory Data Analysis (EDA) procedure can be divided into the following steps:

- The relationship between the different dataset variables is obtained (correlation), along with features statistics and distribution.
- Flights can be divided into categories depending on the delay's length in minutes and these categories are inspected.
- Analysis on cancelled flights and reasons for cancellation.
- Dataset's flight delays distribution is analysed taking into account delays by month, scheduled departure hour or reasons for cancellation.
- Airline companies are examined considering flight's volume, Departure, Arrival and Airline Delays and delay categories.
- Top 20 US airports in terms of flight's volume are selected and studied taking into consideration the number of flights, take-offs and landings, Departure and Arrival Taxi, origin and destination airports and Arrival and Air System Delays.
- Scanning on the influence that the different days of the week (also hours of the day) and months of the year exercise upon Arrival Delays.

The following list shows the attributes or features contained in the dataset, along with their description [5][12]:

1. **YEAR**: 2015. Data Type is *int64*.
2. **MONTH**: 1 (January) - 12 (December). Data Type is *int64*.
3. **DAY**: Day of the month (1 - 31). Data Type is *int64*.
4. **DAY_OF_WEEK**: Day of the week (1 (Monday) - 7 (Sunday)). Data Type is *int64*.
5. **AIRLINE**: Unique airline code (IATA carrier code). Data Type is *object*.
6. **FLIGHT_NUMBER**: Flight number. Data Type is *int64*.
7. **TAIL_NUMBER**: Plane tail number: aircraft registration, unique aircraft identifier. Data Type is *object*.
8. **ORIGIN_AIRPORT**: Origin IATA airport identification code. Data Type is *object*.
9. **DESTINATION_AIRPORT**: Destination IATA airport identification code. Data Type is *object*.
10. **SCHEDULED_DEPARTURE**: Scheduled departure (take-off) time (local, hhmm). Data Type is *int64*.
11. **DEPARTURE_TIME**: Actual departure time (local, hhmm). Real take-off time. Data Type is *float64*.
12. **DEPARTURE_DELAY**: Departure delay, in minutes. Difference between planned and real take-off times. Data Type is *float64*.
13. **TAXI_OUT**: Departure Taxi time, in minutes. Taxiing time in take-off manoeuvre (until aircraft's wheels leave the ground). Data Type is *float64*.
14. **WHEELS_OFF**: Start of flight time, in minutes The time point when the aircraft's wheels leave the ground in take-off. Data Type is *float64*.
15. **SCHEDULED_TIME**: Scheduled flight time, in minutes. Data Type is *float64*.
16. **ELAPSED_TIME**: Real flight time, in minutes (ELAPSED_TIME = TAXI_OUT + AIR_TIME + TAXI_IN). Data Type is *float64*.
17. **AIR_TIME**: Aircraft's time in the air, in minutes. Data Type is *float64*.
18. **DISTANCE**: Flight distance, in miles. Data Type is *int64*.

19. **WHEELS_ON**: Ending of flight time, in minutes. The time point when the aircraft's wheels touch the ground in landing. Data Type is *float64*.
20. **TAXI_IN**: Taxiing time landing manoeuvre, in minutes. Elapsed time between wheels-on in landing and gate arrival at the destination airport. Data Type is *float64*.
21. **SCHEDULED_ARRIVAL**: Scheduled arrival (landing) time (local, hhmm). Data Type is *int64*.
22. **ARRIVAL_TIME**: Actual arrival time (local, hhmm). Real landing time. Data Type is *float64*.
23. **ARRIVAL_DELAY**: Arrival delay, in minutes. Difference between planned and real landing times. A flight is considered as "on time" if it operated less than 15 minutes later than the scheduled arrival time shown in the carrier Computerized Reservations Systems (CRS). Data Type is *float64*.
24. **DIVERTED**: Was the flight diverted? 0 = NO, 1 = YES. Data Type is *int64*.
25. **CANCELLED**: Was the flight cancelled? 0 = NO, 1 = YES. Data Type is *int64*.
26. **CANCELLATION_REASON**: Reason for cancellation (A = Airline, B = Extreme Weather, C = Air System, D = Security). Data Type is *object*.
27. **AIR_SYSTEM_DELAY**: National Airspace System (NAS) delay in minutes. Delays attributable to the National Airspace System (NAS) may include the following conditions: non-extreme weather conditions, airport operations, heavy traffic volume and air traffic control. Data Type is *float64*.
28. **SECURITY_DELAY**: Security delay in minutes. Security delay is caused by evacuation of a terminal, re-boarding of aircraft because of a security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas. Data Type is *float64*.
29. **AIRLINE_DELAY**: Carrier delay in minutes. The air carrier is responsible for this delay. Different examples that cause these delays could be: aircraft cleaning, aircraft damage, awaiting the arrival of connecting passengers or crew, baggage, bird strike, cargo loading, catering, outage-carrier equipment, crew legality (pilot or attendant rest), damage by hazardous goods, engineering inspection, fueling, handling disabled passengers, late crew, lavatory servicing, maintenance, oversales, potable water servicing, removal of unruly passenger, slow boarding or seating, stowing carry-on baggage, weight and balance delays. Data Type is *float64*.

30. **LATE_AIRCRAFT_DELAY**: Late aircraft delay in minutes. This delays take place on the destination airport due to the late arrival of the same aircraft at a previous airport. The ripple effect of an earlier delay at downstream airports is referred to as delay propagation. Data Type is *float64*.

31. **WEATHER_DELAY**: Weather delay in minutes. Weather delay is caused by extreme or hazardous weather conditions that are forecasted or manifest themselves on point of departure, on route or on point of arrival, and prevent from flying. Data Type is *float64*.

The different airlines that perform domestic flights and their associated IATA carrier code, included in the dataset, are the following:

IATA CODE	**AIRLINE**
UA	United Airlines Inc.
AA	American Airlines Inc.
US	US Airways Inc.
F9	Frontier Airlines Inc.
B6	JetBlue Airways
OO	Skywest Airlines Inc.
AS	Alaska Airlines Inc.
NK	Spirit Airlines
WN	Southwest Airlines Co.
DL	Delta Air Lines Inc.
EV	Atlantic Southeast Airlines
HA	Hawaiian Airlines Inc.
MQ	American Eagle Airlines Inc.
VX	Virgin America

Table 2.1: Airlines in the Dataset with associated IATA Carrier Code

First, the correlation matrix shows the correlation coefficients between the different variables. The matrix cells collect the correlation between two variables. It is ideal for data summarising before performing advanced data analysis.

Through the below correlation matrix, it can be stated that some of the dataset features present multicollinearity. This means that they can be linearly predicted from the other variables. In order to perform a better analysis and to use a more manageable dataset, some of those unwanted variables will be deleted and others will be created. The majority of variables have a low correlation with the Arrival Delay. However, Departure Delay has a high correlation with Arrival Delay. Even though, highly correlated features should be removed, it is important to predict Arrival Delays, knowing that the flight was delayed in departure and the magnitude of the delay.

The following correlation matrix can serve as an input and as an indicative:

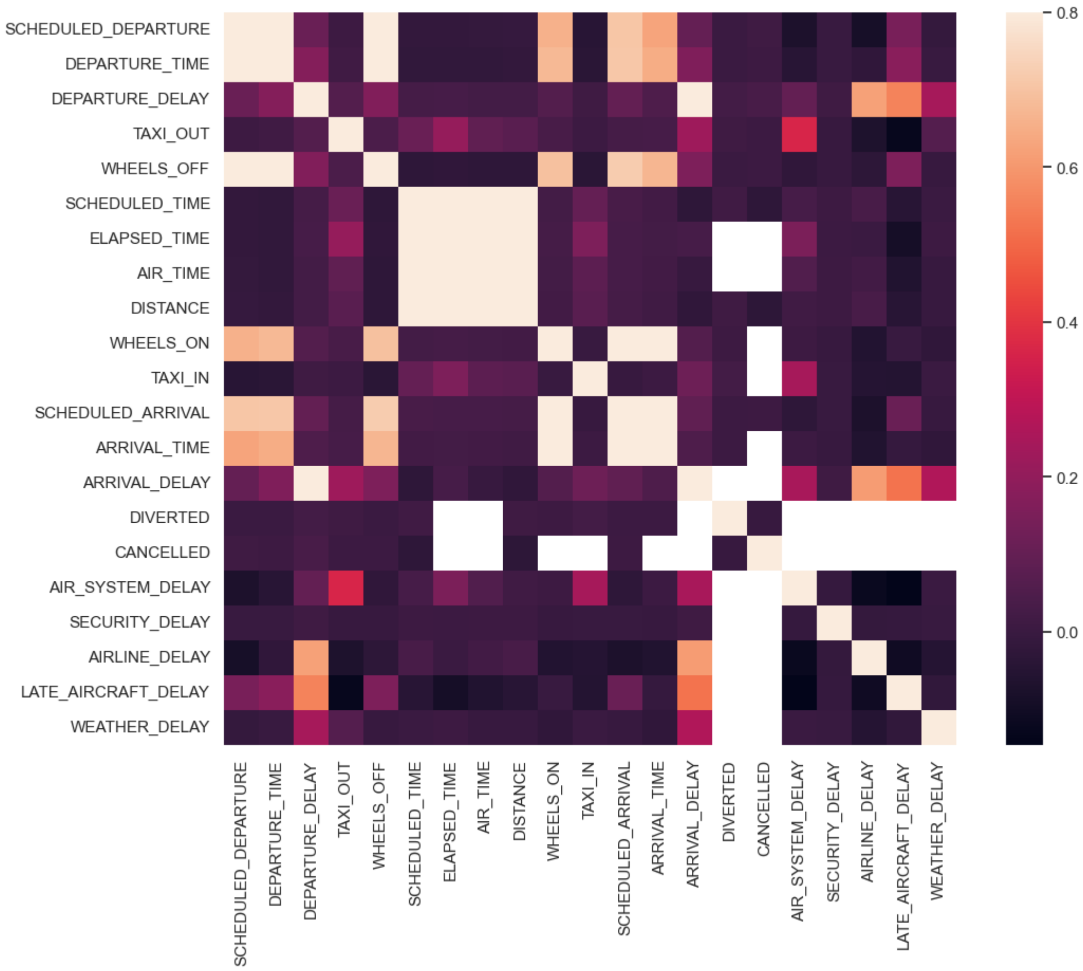


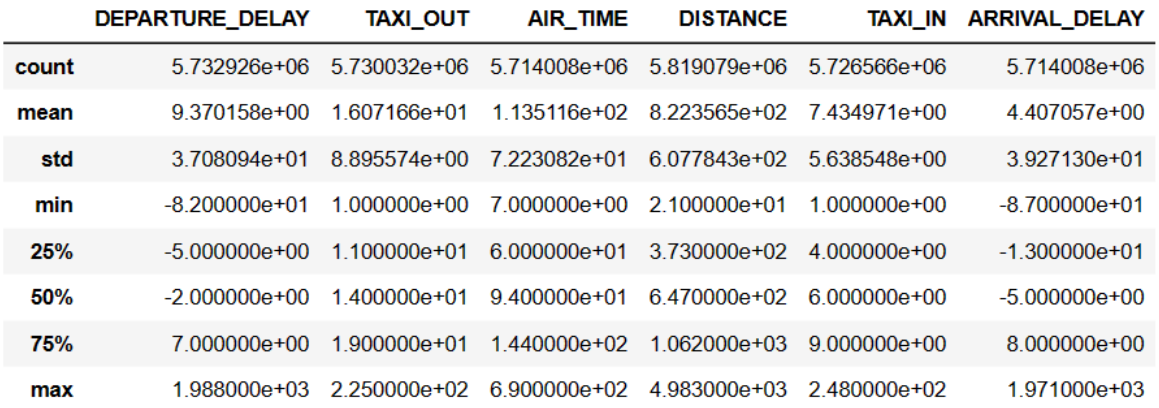
Figure 2.1: Correlation Matrix

To start the analysis, some observations can be stated:

1. The reason of what caused a delay is only available when the Arrival Delay is longer than 15 minutes. Arrival Delay is the sum of Airline Delay, Weather Delay, Air System Delay and Late Aircraft Delay.

2. It is commonly observed that the Elapsed Time is higher than the sum of the time spent in the following stages: Arrival Taxi, Departure Taxi and Air Time (Actual Elapsed Time). This is the reason behind the fact that when flight's departure is on time, flights landing ususally occurs previously to the scheduled landing time. This circumstance also permits to absorb delays on chained flights.

Some statistics from the Bureau of Transportation Statistics dataset are collected in the following table. Studied variables are: Departure Delay (DEPARTURE_DELAY), Departure Taxi (TAXI_OUT), Air Time (AIR_TIME), Distance (DISTANCE), Arrival Taxi (TAXI_IN) and Arrival Delay (ARRIVAL_DELAY).



	DEPARTURE_DELAY	TAXI_OUT	AIR_TIME	DISTANCE	TAXI_IN	ARRIVAL_DELAY
count	5.732926e+06	5.730032e+06	5.714008e+06	5.819079e+06	5.726566e+06	5.714008e+06
mean	9.370158e+00	1.607166e+01	1.135116e+02	8.223565e+02	7.434971e+00	4.407057e+00
std	3.708094e+01	8.895574e+00	7.223082e+01	6.077843e+02	5.638548e+00	3.927130e+01
min	-8.200000e+01	1.000000e+00	7.000000e+00	2.100000e+01	1.000000e+00	-8.700000e+01
25%	-5.000000e+00	1.100000e+01	6.000000e+01	3.730000e+02	4.000000e+00	-1.300000e+01
50%	-2.000000e+00	1.400000e+01	9.400000e+01	6.470000e+02	6.000000e+00	-5.000000e+00
75%	7.000000e+00	1.900000e+01	1.440000e+02	1.062000e+03	9.000000e+00	8.000000e+00
max	1.988000e+03	2.250000e+02	6.900000e+02	4.983000e+03	2.480000e+02	1.971000e+03

Figure 2.2: Bureau of Transportation Statistics Dataset

Violin plot is a graph which portrays statistics summary, with data peaks. Some of the statistics that this kind of plot agglutinates, could be the median of the data and the interquartile range. It enables the visualisation of the distribution, the shape of numerical data and, also, the variable's density. Different features distribution are displayed:

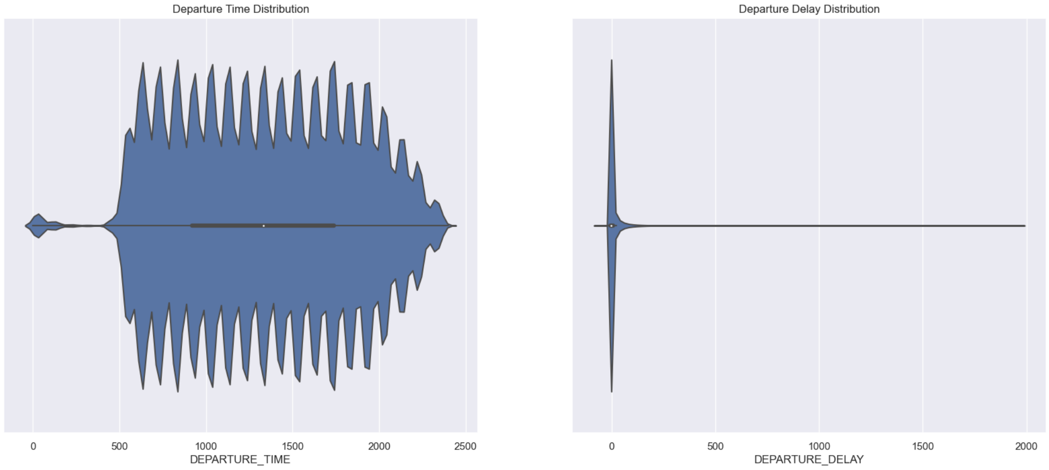


Figure 2.3: Departure Time and Departure Delay Distributions (Violin Plots)

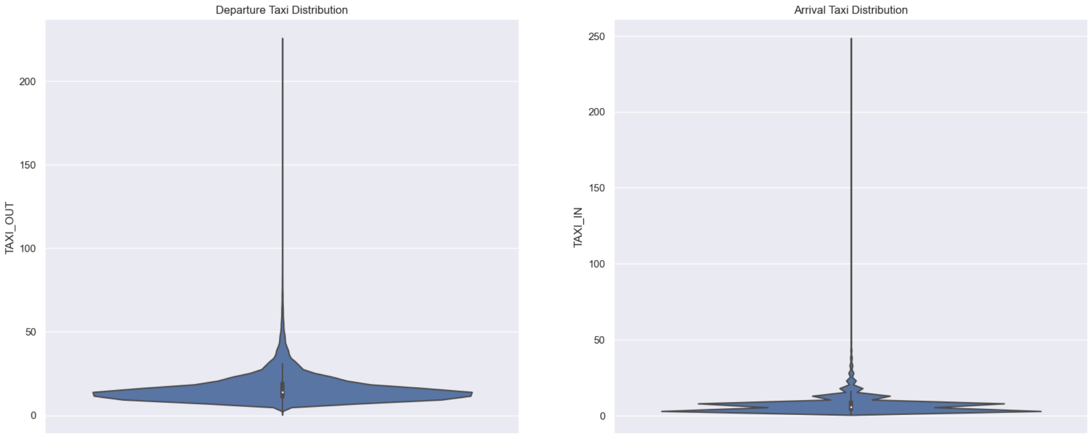


Figure 2.4: Arrival and Departure Taxi Distributions (Violin Plots)

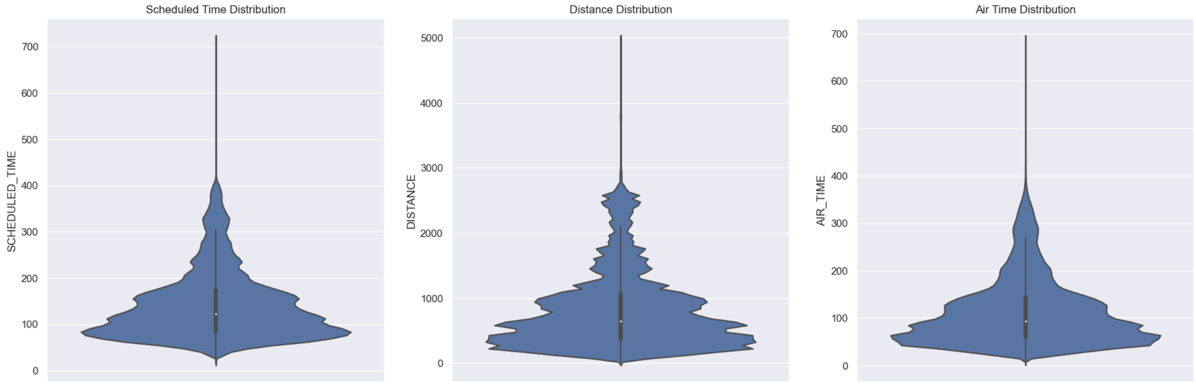


Figure 2.5: Scheduled Time, Distance and Air Time Distributions (Violin Plots)

The Violin plots show that Distance and Scheduled Time have a very similar distribution. Also, Air Time has a related distribution from the two previous features. A heat map (data visualization technique) is represented to study the correlation between these variables:

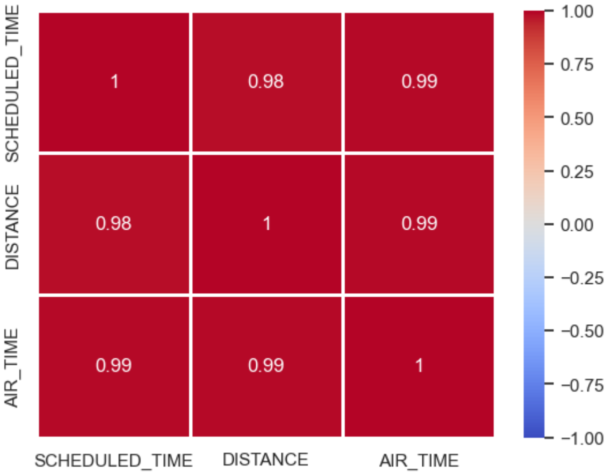


Figure 2.6: Heat Map (Scheduled Time, Distance and Air Time)

Scheduled Time Distribution is highly skewed with a great right tail. This variable can be transformed before building a model. Departure Delay, Departure Taxi, Arrival Taxi, Distance and Air Time are also highly right-tailed and can also be transformed. From the Departure Time Distribution, it can be stated the fact that departing flights are highly concentrated between 5 o'clock in the morning and 9 o'clock in the night.

Even though Departure and Arrival Taxi have a comparable distribution, it can be stated that Departure Taxi time is usually higher than Arrival Taxi time in the destination airport. This could be due to queuing in the Origin Airport and Air System delays, previously to departure.

The heat map graph is the confirmation that Scheduled Time, Distance and Air Time are highly correlated, with a correlation of 0.98 (between Scheduled Time and Distance) and 0.99 (between Air Time and Distance). This correlation relates to the fact that great distances imply more Scheduled Time and more Air Time and shorter distances to take less Scheduled time and Air Time. Only one of these variables will be used for the model creation and the other two will be dropped.

Next, a bivariate analysis is performed to discover the relationships between some of the variables contained in the dataset:

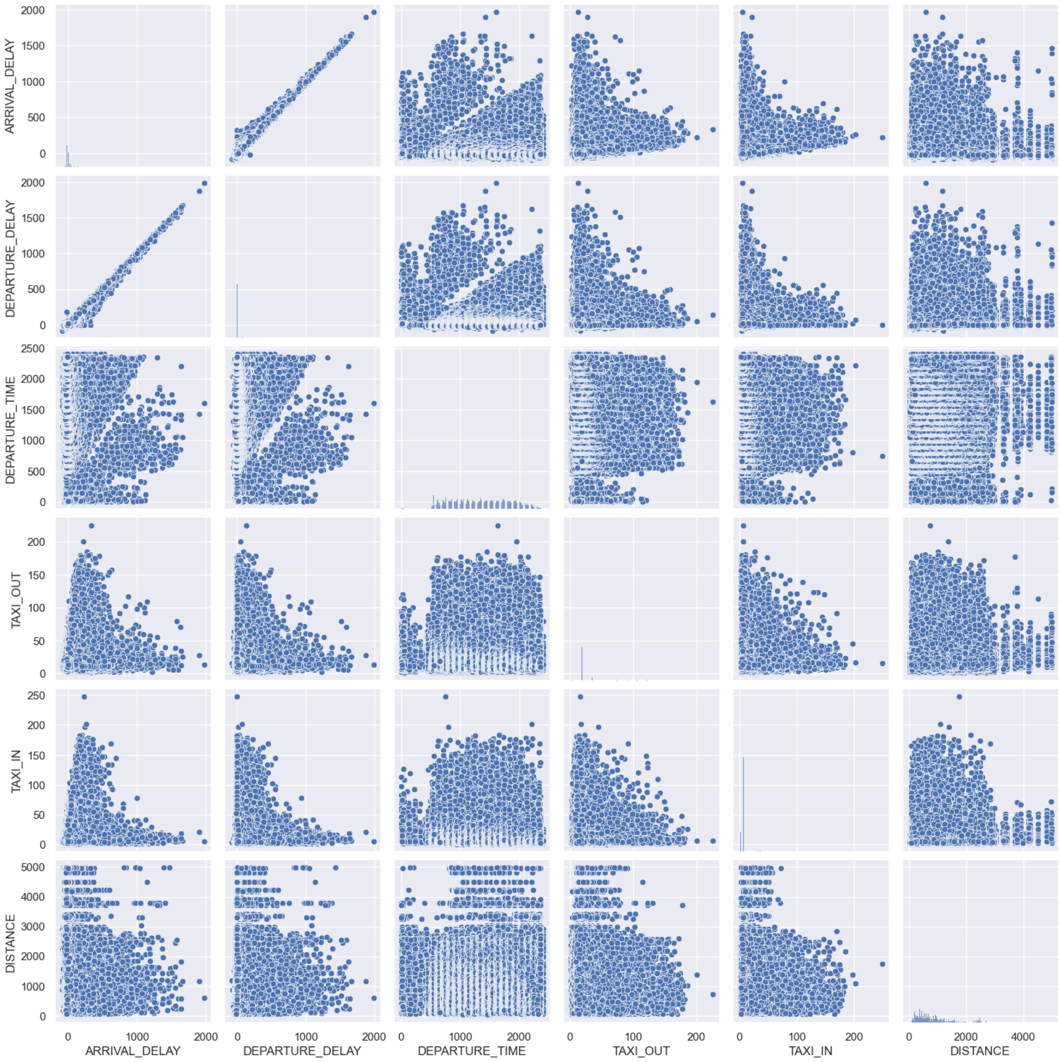


Figure 2.7: Bivariate Analysis

2.1. Exploratory Data Analysis: Flight Delay Categories

Flight Delays can be divided into three categories:

- On Time or Small Delay (up to 15 minutes delay). Generally, this delays are considered not accountable.
- Medium Delay (between 15 and 45 minutes delay)
- Large Delay (more than 45 minutes delay).

DELAY_CATEGORY represents whether the flight was On Time or Early (OnTime_Early), delayed up to 15 minutes (Small_Delay), delayed up to 45 minutes (Medium_Delay), delayed more than 45 minutes (Large_Delay), diverted (Diverted) or cancelled (Cancelled).

The following graphs collect the percentage of each Delay Category with respect to the total number of flights from the dataset and the number of flights that are categorised in each Delay Category:

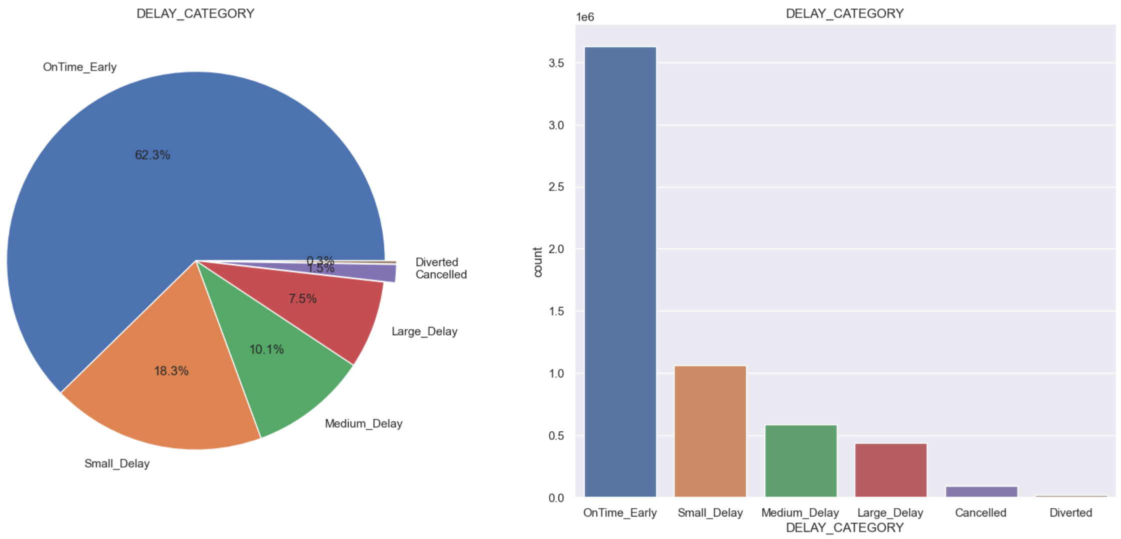
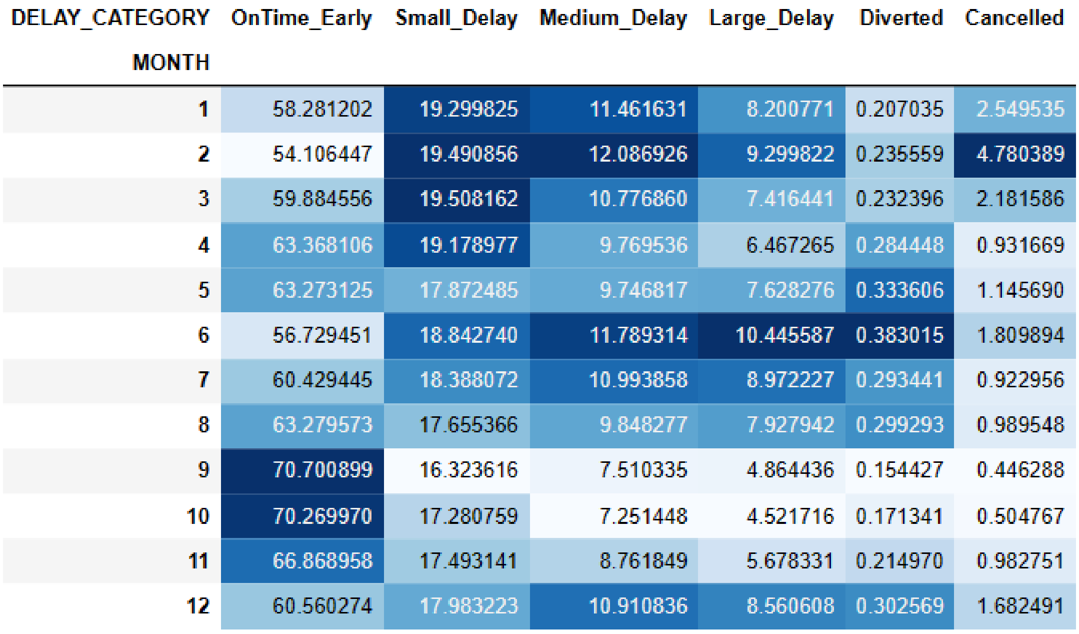


Figure 2.8: Delay Category

In 2015, 62.3% of domestic flights in the US arrived on-time or early considering the scheduled arrival time. 18.3% of them were delayed, but less than 15 minutes. 10.1% of the US commercial flights arrived between 15 and 45 minutes late. Finally, 7.5% of the flights had delays of more than 45 minutes. From another point of view and establishing another interpretation, 90.7% of dataset flights have a delay of less than 45 minutes. It is important to point out the fact that the majority of flights arrived before the Scheduled Arrival time.

On the other hand, just a 1.5% of the flights were cancelled and 0.3% were diverted. The stats from the Delay Categories from 2015's monthly national flights are:



DELAY_CATEGORY	OnTime_Early	Small_Delay	Medium_Delay	Large_Delay	Diverted	Cancelled
MONTH						
1	58.281202	19.299825	11.461631	8.200771	0.207035	2.549535
2	54.106447	19.490856	12.086926	9.299822	0.235559	4.780389
3	59.884556	19.508162	10.776860	7.416441	0.232396	2.181586
4	63.368106	19.178977	9.769536	6.467265	0.284448	0.931669
5	63.273125	17.872485	9.746817	7.628276	0.333606	1.145690
6	56.729451	18.842740	11.789314	10.445587	0.383015	1.809894
7	60.429445	18.388072	10.993858	8.972227	0.293441	0.922956
8	63.279573	17.655366	9.848277	7.927942	0.299293	0.989548
9	70.700899	16.323616	7.510335	4.864436	0.154427	0.446288
10	70.269970	17.280759	7.251448	4.521716	0.171341	0.504767
11	66.868958	17.493141	8.761849	5.678331	0.214970	0.982751
12	60.560274	17.983223	10.910836	8.560608	0.302569	1.682491

Figure 2.9: Percentage of Delay Category per Month

The lower percentage of flights that arrived early or on-time took place in June (6th month). On this month, the percentage of flights that were cancelled or diverted is greater than previous and subsequent months. Also, June is the month with a higher percentage of flights with a large Arrival Delay (more than 45 minutes).

In February, it is collected the biggest percentage of cancelled flights (4.78% of all February flights). This month also registers a high percentage of flights with a small delay (up to 15 minutes of delay).

The best numbers in terms of flights that arrived early or on-time are presented after the Summer months (September, October and November), with a percentage of the number of flights of 70.7%, 70.27% and 66.87%, respectively.

2.2. Exploratory Data Analysis: Cancellation Reason

There are a series of reasons, responsible for a flight cancellation, that are refered in the dataset by the following codes:

- **Airline or Carrier**: A.
- **Extreme Weather**: B.
- **National Airspace System (NAS)**: C.
- **Security**: D.

The following graphs collect the percentage of each Cancellation Reason with respect to the total number of cancelled flights from the dataset and the number of cancelled flights that are categorised in each Cancellation Reason:

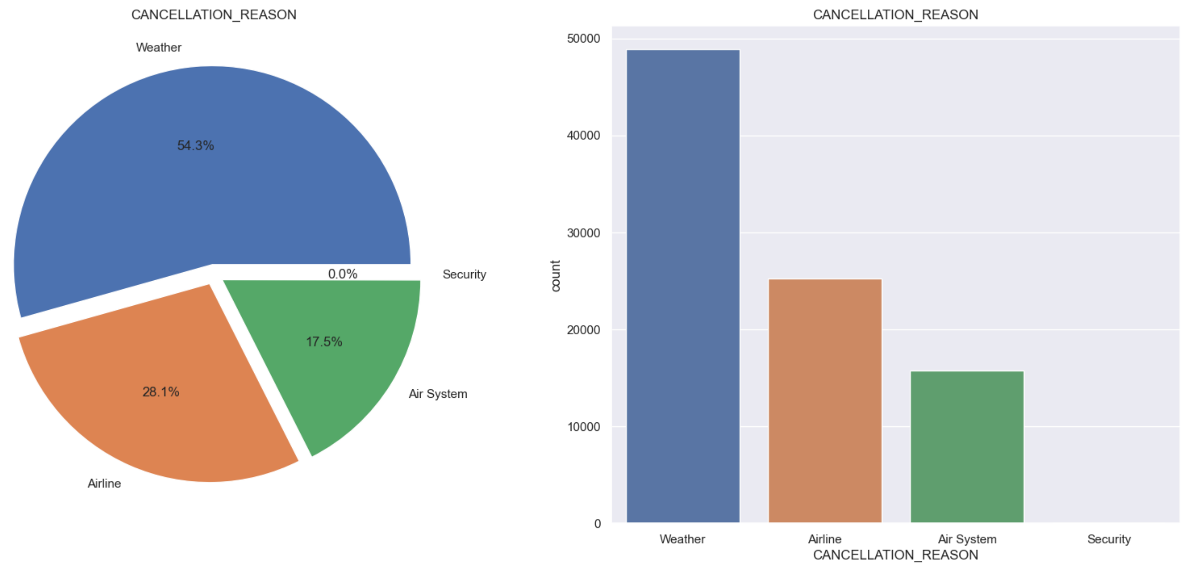
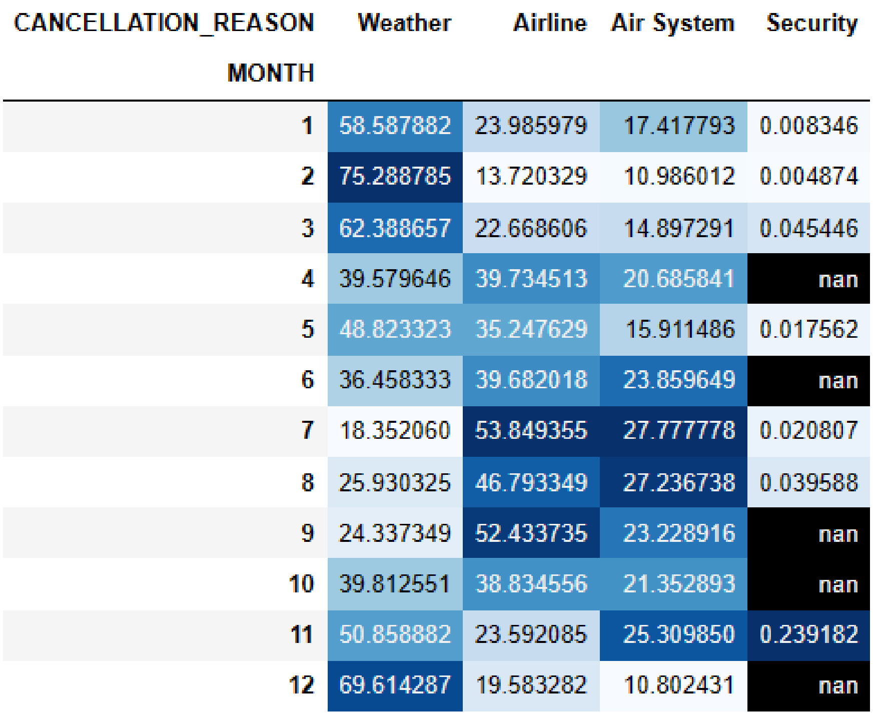


Figure 2.10: Cancellation Reason

From the previous graphs, representing the different reasons for cancellations, it can be extracted that Extreme Weather is the main reason behind more than half of the flights cancelled (54.3% of cancelled flights were due to the weather). Airlines and the National Airspace System (NAS) are responsible for 28.1% and 17.5% respectively of the US domestic cancelled flights. Security isn't the cause for any of the flights that were cancelled.

The stats from the Cancellation Reasons from 2015's monthly cancelled domestic flights are:



CANCELLATION_REASON	Weather	Airline	Air System	Security
MONTH				
1	58.587882	23.985979	17.417793	0.008346
2	75.288785	13.720329	10.986012	0.004874
3	62.388657	22.668606	14.897291	0.045446
4	39.579646	39.734513	20.685841	nan
5	48.823323	35.247629	15.911486	0.017562
6	36.458333	39.682018	23.859649	nan
7	18.352060	53.849355	27.777778	0.020807
8	25.930325	46.793349	27.236738	0.039588
9	24.337349	52.433735	23.228916	nan
10	39.812551	38.834556	21.352893	nan
11	50.858882	23.592085	25.309850	0.239182
12	69.614287	19.583282	10.802431	nan

Figure 2.11: Percentage of Cancellation Reason per Month

The number of cancelled flights per Departure Date (2015) collect certain peaks for determined days throughout the year ant it is showed on the following chart:

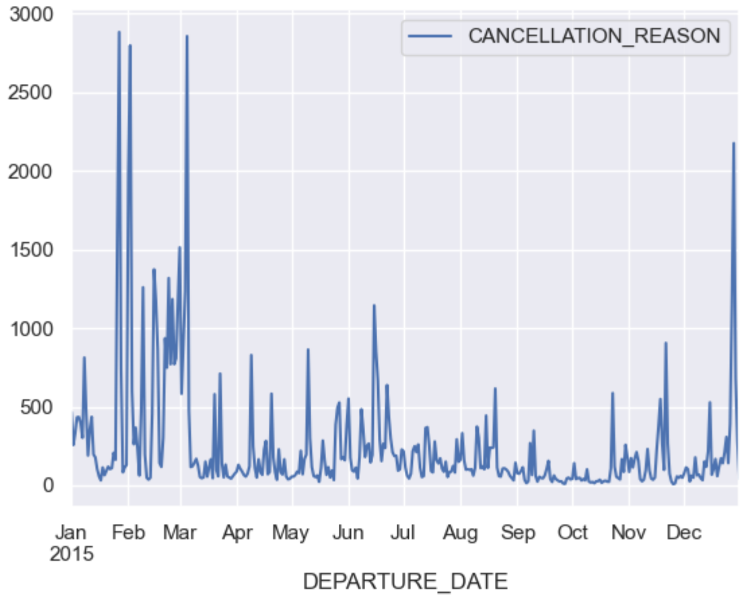


Figure 2.12: Cancelled Flights per Departure Date

During 2015, cancellations were concentrated during the first months of the year, as we can observe some peaks in the final days of January, February and the first days of March. The reason for this cancellations was an extreme weather event, "The February 2015 North American Cold Wave", that mainly affected the eastern half of the United States. The cause of this event was due to the polar vortex advancing in the south direction into the east coast and even affecting the southeast. The cold wave lasted until the start of March. Extreme weather was also the reason behind all the December cancellations that affected the return home from Christmas holidays. This can be corroborated by the following representation, with the US national flights cancellations by calendar:

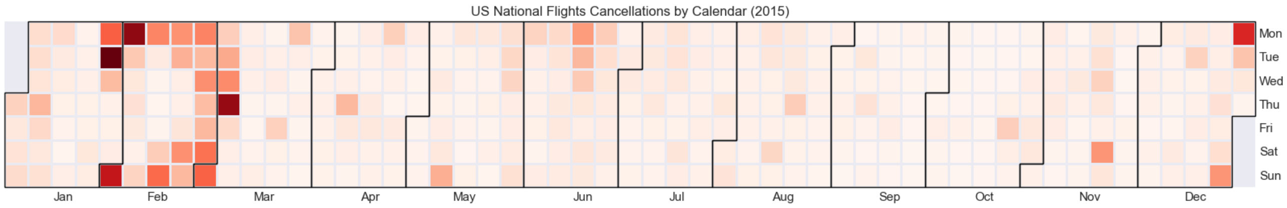


Figure 2.13: US National Flights Cancellations by Calendar

2.3. Exploratory Data Analysis: Flight Delays

For analysing flight delays, the feature set is filtered and only flights with an Arrival Delay value of more than 0 minutes are taken into account. First, a distribution plot for the Arrival Delay variable is plotted. Distribution plots are used for inspecting univariate and bivariate distributions. By contrasting the real data distribution with the theoretical values anticipated from a certain distribution, distribution plots provide a visual assessment of the feature distribution from the dataset. Distribution plots are used to discover whether the sample data originates from a certain distribution.

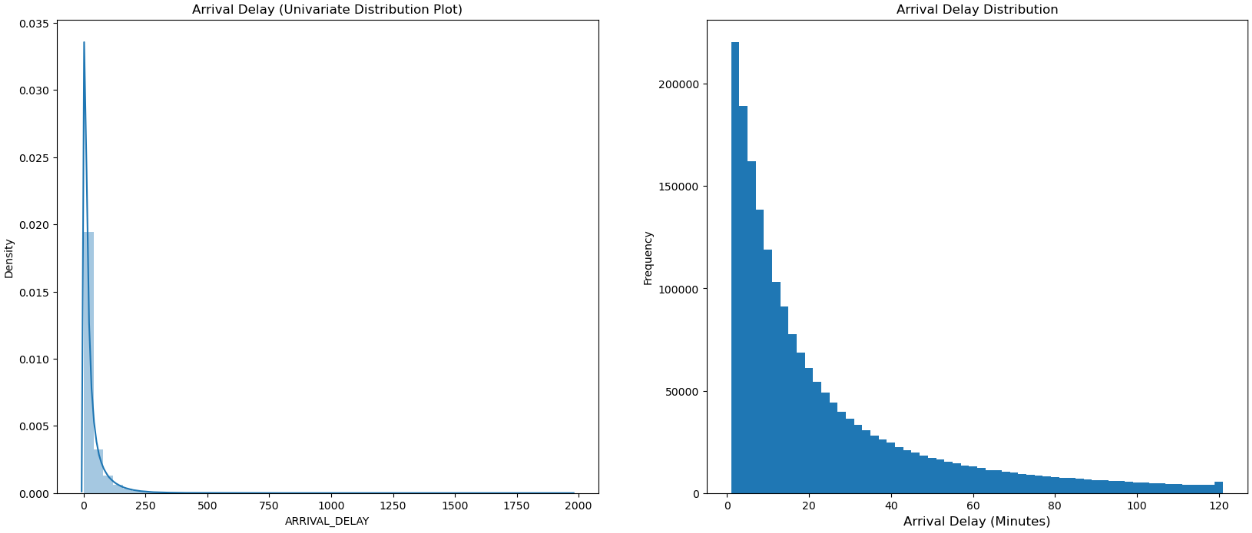


Figure 2.14: Arrival Delay Distribution Plots

As it can be observed from the histogram (univariate distribution plot) and taking into account the skewness and kurtosis metrics, delays are mostly located on the left side of the graph, being the majority of delays short in time. The **Skewness** value of **5.58** indicates that there is a greater number of smaller values for Arrival Delay. On the other hand, the **Kurtosis** great positive value (**66.5**) stipulates a very peaked distribution, as there is a density peak for low values of Arrival Delay.

The graph in the right position shows the Arrival Delay Distribution for a delay up to 2 hours (120 minutes). It displays the frequency of the flight's Arrival Delay in minutes (from 1 minute to 120 minutes of delay). It can be concluded that most of flight delays are short in time. The decrease in the number of flights when the time of Arrival Delay increases can be mathematically resembled to an exponential decay.

The following diagrams represent the Average Arrival Delay and the sum of minutes delayed per month. The joint plot contains the relationship between Arrival Delay and the Scheduled Departure time.

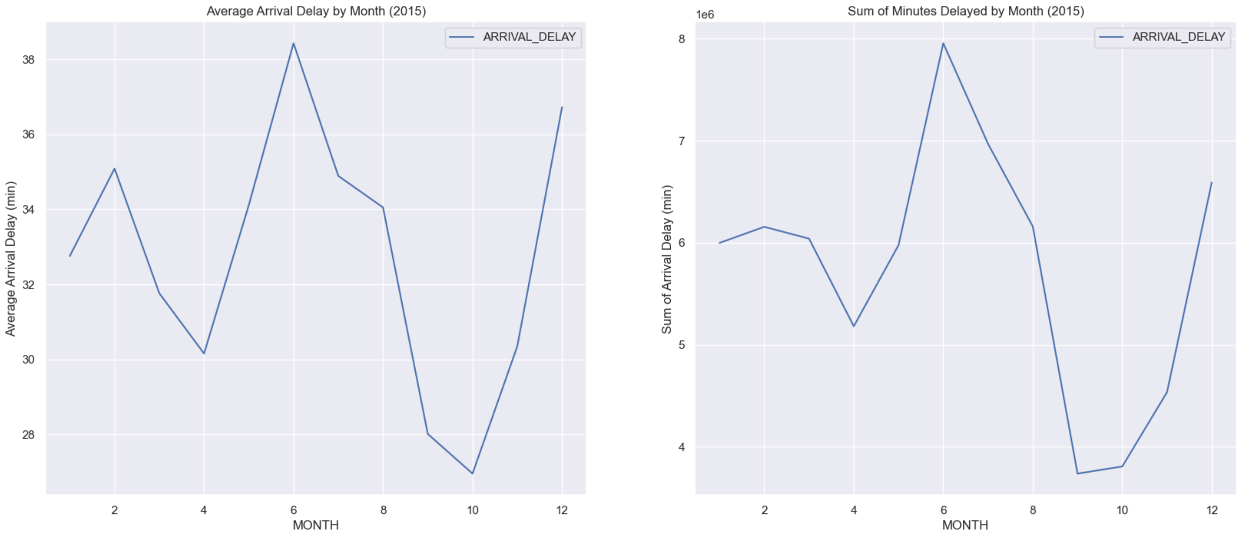


Figure 2.15: Average Arrival Delay by Month & Sum of Minutes Delayed by Month

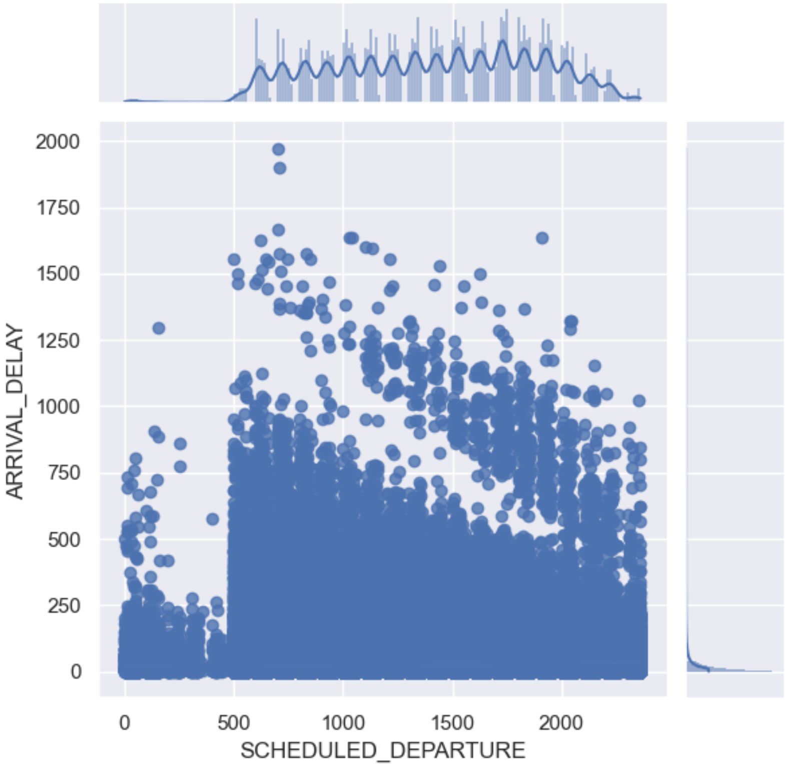


Figure 2.16: Scheduled Departure vs Arrival Delay (Joint Plot)

Based on the average Arrival Delay results, flight delays with high value focused specially on February, June and December. There are also peaks on average delays on July and August.

Taking into account the different day hours when the flight departure was scheduled it can be seen in the top section of the scatterplot how the delays have peaks and are concentrated on the o'clock hours. It can be observed from the scatterplot the fact that starting from 5 in the morning and spreading throughout the different upcoming hours of the day, the delays are visually splitted into two groups (longer and shorter delays). A conclusion that can be extracted from this fact is that the delays that were created in previous flights (chained flights), increased or decreased in following day flights.

The next figure represents the sum of the delays produced by the different reasons per month: Air System (AIR_SYSTEM_DELAY), Airline (AIRLINE_DELAY), Weather (WEATHER_DELAY), Security (SECURITY_DELAY) and Late Aircraft (LATE_AIRCRAFT_DELAY).

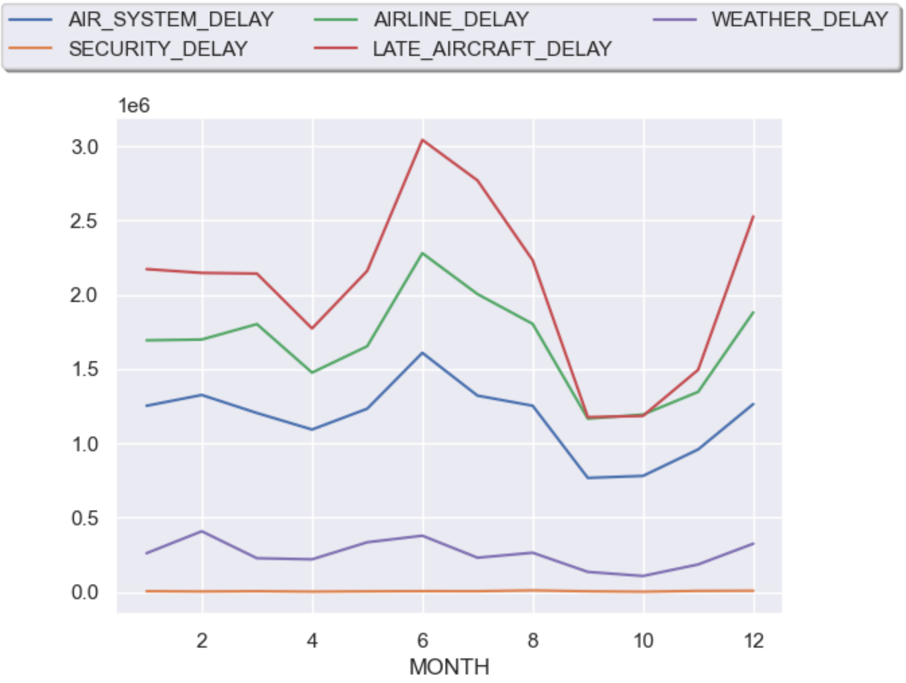


Figure 2.17: Delay Reason per Month

Based on the previous graph, it can be concluded that there are three main reasons that cause most of the flight delays during 2015: Late Aircraft Delay, Airline or Carrier Delay and Air System Delay, in that order. Apparently, Late Aircraft Delay is the most important variable, being the volume of delayed time greater than for the other causes. However, this cause of delay can be affected by the other reasons because of the delay propagation phenomenon.

It is clear that there are no delays caused by Security causes. Also, few delays are caused by Weather reasons, as the sum of delayed minutes for this cause (WEATHER_DELAY) is very low, compared to the other motives.

2.4. Exploratory Data Analysis: Airline or Carrier Delays

First, the displayed table shows the different airlines or carriers ordered by the number of operated national flights from the year 2015. The order is descendent, from the airline with the largest number of flights to the carrier with the least number of flights from the dataset. It also shows a descriptive statistics which summarises the tendency, dispersion and shape of the attribute's (Departure Delay and Arrival Delay) distribution in the dataset [14]:

AIRLINE (IATA CODE)	Flight's Volume	DEPARTURE_DELAY		ARRIVAL_DELAY	
		Mean	Standard Deviation	Mean	Standard Deviation
Southwest Airlines Co. (WN)	1242403	11	31	4	33
Delta Air Lines Inc. (DL)	870275	7	36	0	38
American Airlines Inc.(AA)	712935	9	42	3	44
Skywest Airlines Inc. (OO)	576814	8	38	6	39
Atlantic Southeast Airlines (EV)	554752	9	39	7	41
United Air Lines Inc. (UA)	507762	14	42	5	44
American Eagle Airlines Inc. (MQ)	507762	10	41	6	44
JetBlue Airways (B6)	262042	12	39	7	41
US Airways Inc. (US)	61248	6	29	5	36
Alaska Airlines Inc. (AS)	171439	2	26	-1	29
Spirit Air Lines (NK)	115193	16	44	14	46
Frontier Airlines Inc. (F9)	90090	13	50	13	52
Hawaiian Airlines Inc. (HA)	76041	0	25	2	26
Virgin America (VX)	61248	9	32	5	36

Table 2.2: Departure and Arrival Delay Statistics for each Airline

To complete the above mentioned, the following pie chart extracted from the dataset shows the percentage of domestic flights performed by each airline in the year 2015:

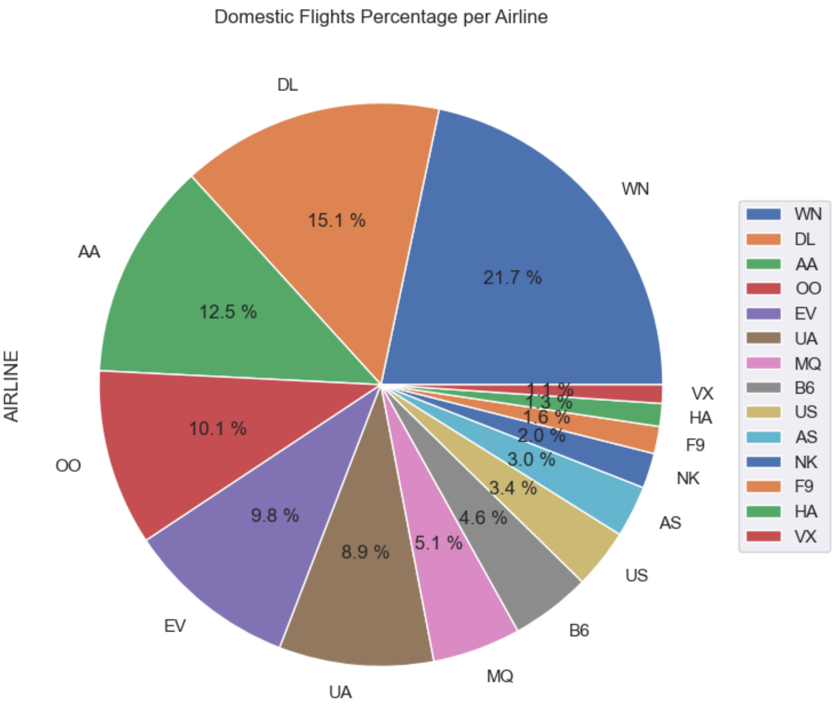


Figure 2.18: Domestic Flights per Airline

Considering the pie chart that shows the flights percentage per airline, it can be seen a great diffference between the carriers. Southwest Airlines is the company that agglutinates more domestic flights, with 21.7% of the flights. This percentage of flights is nearly the same as the sum of the number (percentage) of flights embraced by the 8 smaller airlines (Virgin America (VX), Hawaiian Airlines (HA), Frontier Airlines (F9), Spirit Airlines (NK), Alaska Airlines (AS), US Airways (US), JetBlue Airways (B6) and American Eagle Airlines (MQ)).

The descriptive statistics (Table 2.2) include parameters such as the mean and the standard deviation of Departure and Arrival Delay for each airline. It definitely stands out the mean values of Arrival Delay from Frontier Airlines Inc. and Spirit Airlines, two of the companies with the least flight's volume. This mean values of Arrival Delay (13 and 14 minutes, respectively) are much higher than those presented by other airline companies. The highest Arrival Delay standard deviation values are also achieved by the previous airlines (52 and 46 minutes, respectively). Generalising, mean Departure Delay values for the different airlines are greater than average Arrival Delay for these companies.

Next, the following set of graphs present the average Arrival Delay and the Arrival Delay distribution for every one of the dataset's airlines or carriers. Also, the average Airline Delay and the Airline Delay distribution for each of the carriers. In the case of the average delay by airline graph, it can be calculated the mean Arrival Delay between all airlines taking into account their individual average delay. This value is equal to 5.39 minutes, which is quite low, meaning that the standard for every carrier is to respect the established schedule (the majority of flights arrive on time).

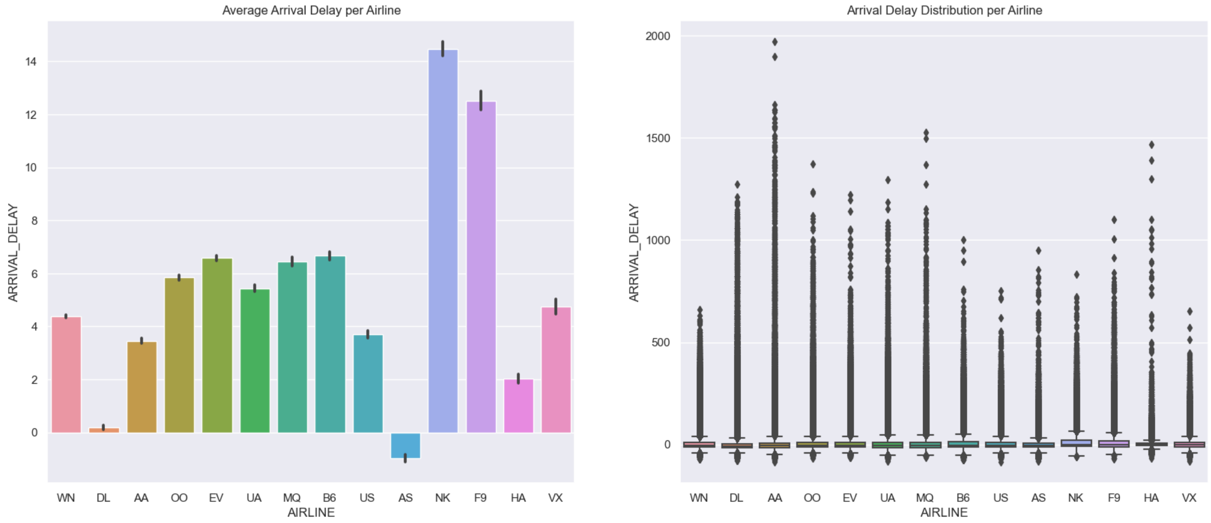


Figure 2.19: Average Arrival Delay per Airline & Arrival Delay Distribution per Airline

Alaska Airlines (AS) shows a negative value of Mean Arrival Delay, arriving earlier than the stipulated Arrival Time. At the same time, Delta Airlines Inc. (DL) has an average Arrival Delay of approximately 0 minutes (On time Arrival). The average Arrival Delay between all the airlines is 5.39 minutes, which corresponds to a Small Delay taking into account the Delay Categories established earlier. Based on the graph's results, no relationship can be drawn between the number of flights from a carrier and the average Arrival Delay.

The Airline Delay will be the next variable to be examined. The feature's data is only included in the dataset when the flight's Arrival Delay value exceeds 15 minutes. So, the dataset must be filtered by the Arrival Delay variable (more than 15 minutes) in order to retrieve the Airline Delay.

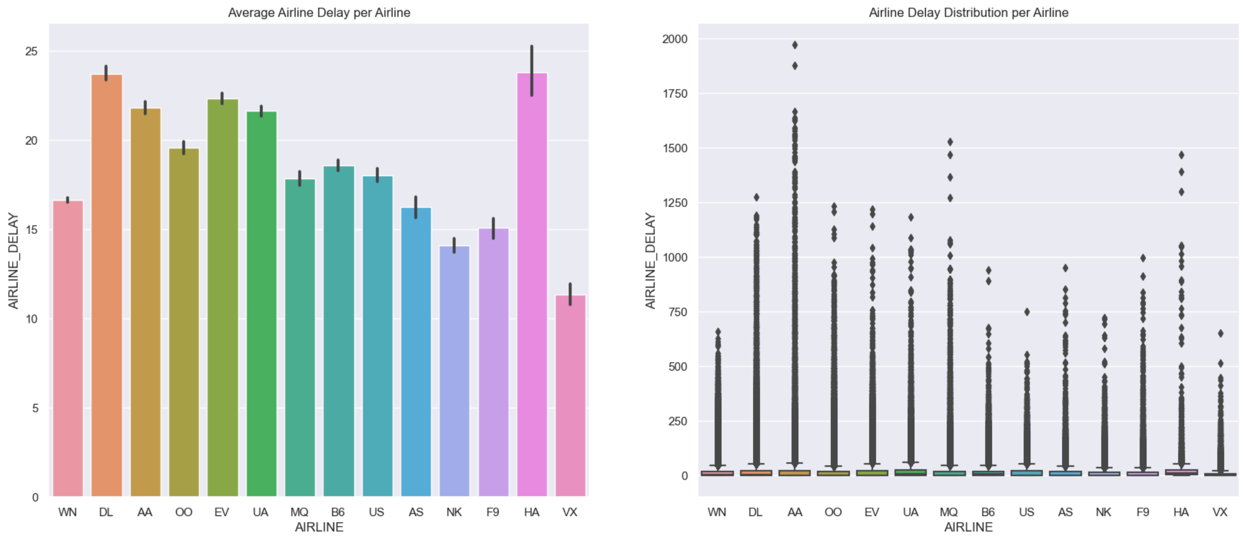


Figure 2.20: Average Airline Delay per Airline & Arrival Delay Distribution per Airline

From the top 5 airlines considering 2015's number of national US flights, it is Southwest Airlines Co. (WN) the company that achieves the best results taking into account the mean Airline Delay (approximately 17 minutes). Delta Airlines Inc. (DL) and Hawaiian Airlines Inc. (HA) present the highest average Airline Delay, which is around 24 minutes. The average Arrival Delay from the different carriers is 17±6 minutes, which is a low value. Nevertheless, also Large Delays occur, what is counteracted in the previous result by the great amount of flights that arrive early or on time in the destination airport.

It cannot be established any relationship between the volume of flights from an Airline and the average Carrier Delay. Some of the airlines with lower volume of flights in 2015, like Alaska Airlines Inc. (AS), Spirit Airlines (NK), Frontier Airlines Inc. (F9) and Virgin America (VX), present the lower average Airline Delays that can be extracted from the dataset. However, the last but one airline in terms of flights volume, Hawaiian Airlines Inc. (HA) has the highest mean Carrier Delay.

Observing the delay distribution by airline, it can be affirmed that American Airlines Inc. (AA) registered the maximum Airline Delay for 2015. This result goes in line with the hypothesis that high delays can relate with high volume of flights.

In conclusion, the carrier is an important factor when determining delays, just like the number of flights.

To continue, the following series of graphs speak for the number of flights from every Delay Category for each of the airlines or carriers:

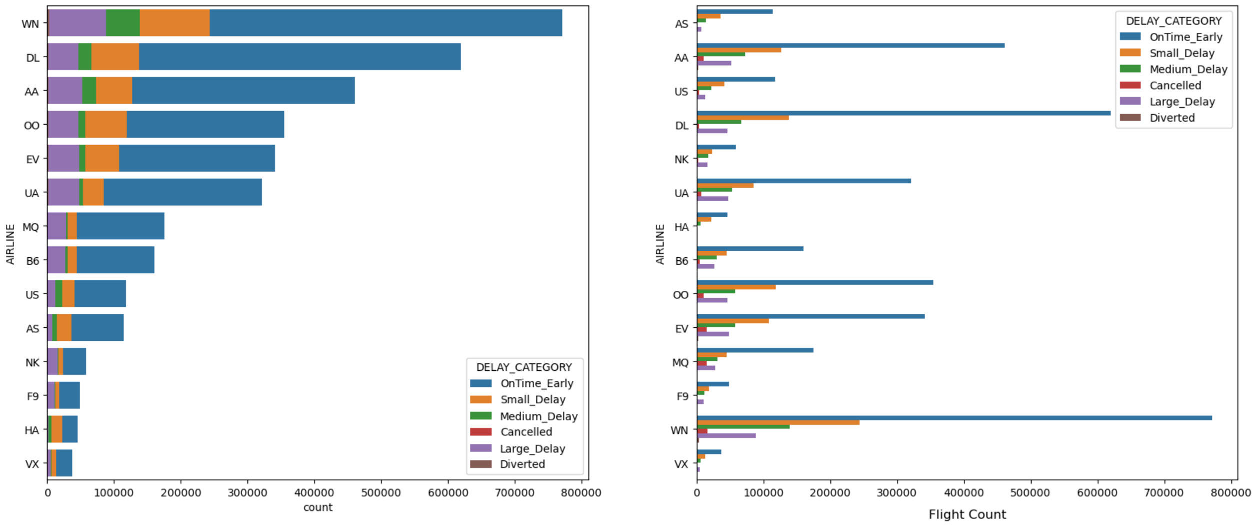


Figure 2.21: Types of Delay (Delay Categories) per Airline

The previous graphs provide a count of the different flight delay categories for each airline. This categories discern whether the flight was On Time or Early (OnTime_Early), delayed up to 15 minutes (Small_Delay), delayed up to 45 minutes (Medium_Delay), delayed more than 45 minutes (Large_Delay), diverted (Diverted) or cancelled (Cancelled).

One conclusion that can be extracted is that independently of the airline, the proportion of large delays (more than 45 minutes) with respect to the total number of flights is very low. Nevertheless, the proportion of the three types of delays (Small_Delay, Medium_Delay and Large_Delay) vary from one airline to another.

In order to end the Exploratory Data Analysis (EDA) for the airline or carrier delays, the mean departure and arrival delay for the distinct airlines is studied:

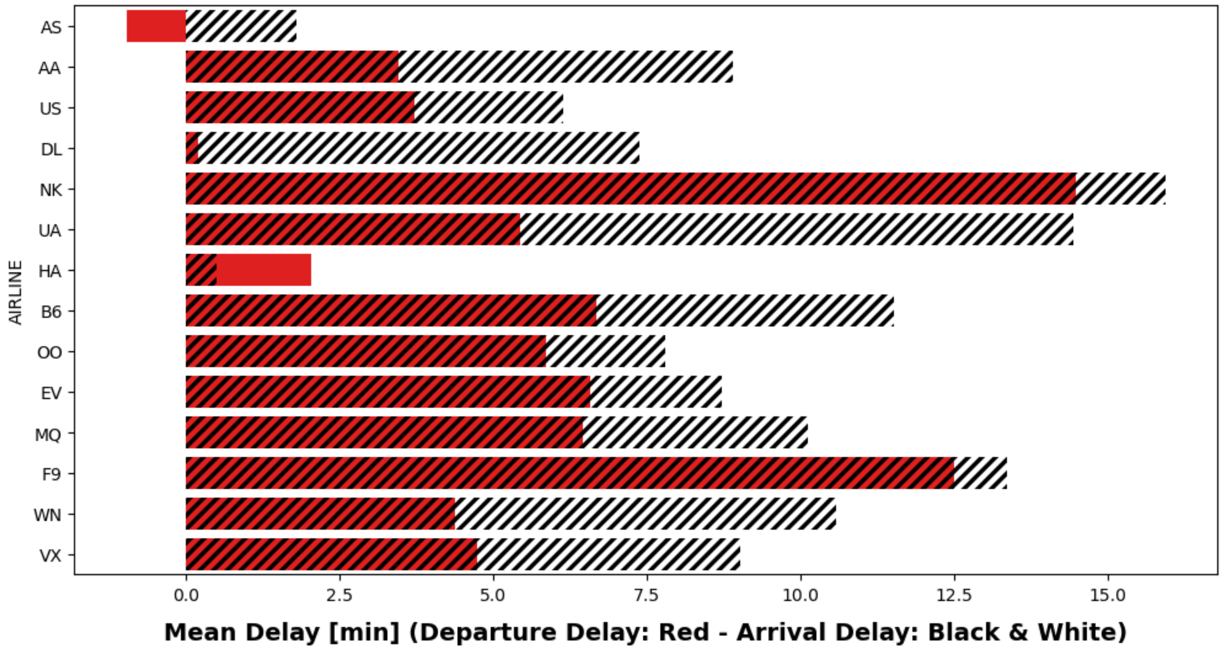


Figure 2.22: Mean Departure and Arrival Delay per Airline

On this bar chart, it can be observed that Arrival Delays are lower than Departure Delays. The explanation behind this statement is the fact that airlines increase the flight speed with the objective of reducing the landing delays with respect to the take-off delays. The only exception to this conclusion is Hawaiian Airlines Inc. (HA), as this carrier has more delay time at arrival than at departure.

2.5. Exploratory Data Analysis: Origin and Destination Airports

There are 322 US Airports that are both Origin and Destination Airports for the national flights contained in this dataset for the year 2015.

As there are too many airports on the feature set, the focus will be placed on the 20 airports with more flight's volume, which are the following:

Airport (IATA CODE)	Number of Flights
Atlanta (ATL)	346,836
Chicago (ORD)	285,884
Dallas Fortworth (DFW)	239,551
Denver (DEN)	196,055
Los Angeles (LAX)	194,673
San Francisco (SFO)	148,008
Phoenix (PHX)	146,815
Houston (IAH)	146,622
Las Vegas (LAS)	133,181
Minneapolis (MSP)	112,117
Orlando (MCO)	110,982
Seattle–Tacoma (SEA)	110,899
Detroit (DTW)	110,899
Boston (BOS)	107,847
Newark (EWR)	101,772
Charlotte (CLT)	100,324
La Guardia - New York (LGA)	99,605
Salt Lake City (SLC)	97,210
New York (JFK)	93,811
Baltimore-Washington (BWI)	86,079

Table 2.3: Airports in the Dataset with associated IATA Carrier Code

The flight volume differs greatly between the studied airports (20 airports with more number of flights). For example, Chicago O'Hare International Airport (ORD) welcomes 285,884 flights in 2015, while Baltimore/Washington International Thurgood Marshall Airport (BWI) counts with 86,079 flights. Hartsfield-Jackson Atlanta International Airport (ATL) is the biggest airport in terms of incoming and outgoing flights in 2015, with nearly 350,000 flights (exactly 346,836 flights).

Another aspect to take into account is that the selected airports are operation bases for some of the airlines previously analysed. Some examples could be: Chicago

O'Hare International Airport (ORD) with United Airlines Inc. (UA), Dallas Fort Worth International Airport (DFW) with American Airlines Inc. (AA) and Seattle-Tacoma International Airport (SEA) with Alaska Airlines Inc. (AS).

Next couple of bar plots count the number of take-offs and landings for each of the considered origin and destination airports:

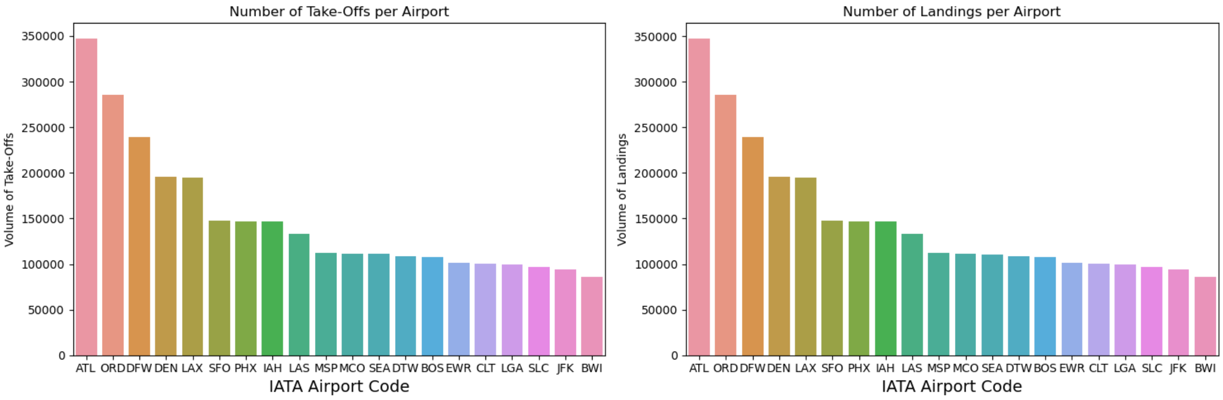


Figure 2.23: Number of Take-Offs and Landings per Airport

As it can be deduced from the previous graphs, the number of Take-Offs and the number of Landings are balanced for each of the airports taken into account (Origin and Departure Airports, respectively). Due to the high number of airports contained in the data set, the origin and destination airports are filtered and only the 20 airports with a greater number of flights are considered.

Following graphs account for the average Departure and Arrival Taxi for every origin and destination airport, respectively:

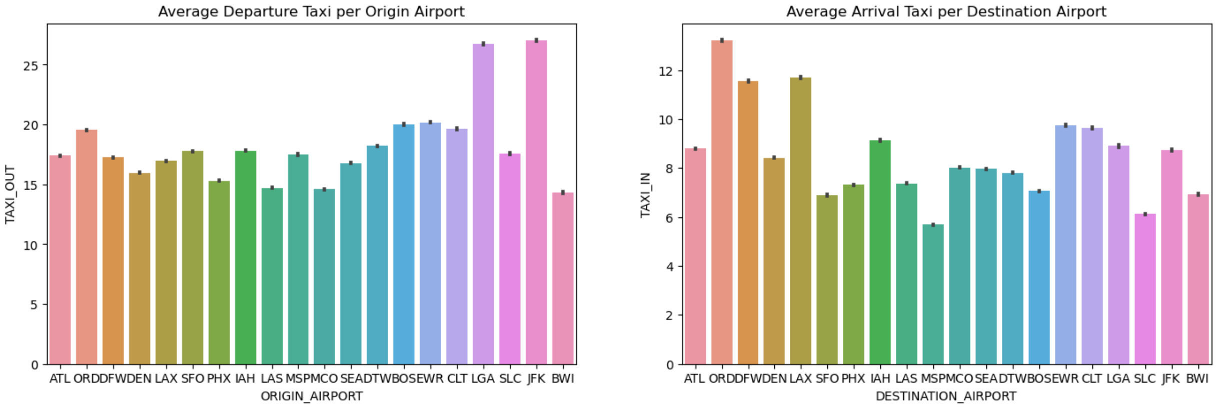


Figure 2.24: Average Departure Taxi per Origin Airport & Average Arrival Taxi per Destination Airport

The graphs show that the average Departure Taxi time for all the airports is much higher than the mean Arrival Taxi for the same airports. For example, Atlanta Airport (ATL) registers a mean Departure Taxi of 17.5 minutes and an average Arrival Taxi of 9 minutes. Another example could be New York (JFK), with a mean Departure Taxi of 28 minutes (the largest of all the considered airports) and an average Arrival Taxi of 9 minutes. The smallest recorded Average Arrival Taxi corresponds to Minneapolis Airport (MSP), being less than 6 minutes.

To continue the analysis, Arrival Delay is the feature studied next. The following set of graphs show the mean Arrival Delay and the Arrival Delay distribution per origin and destination airports:

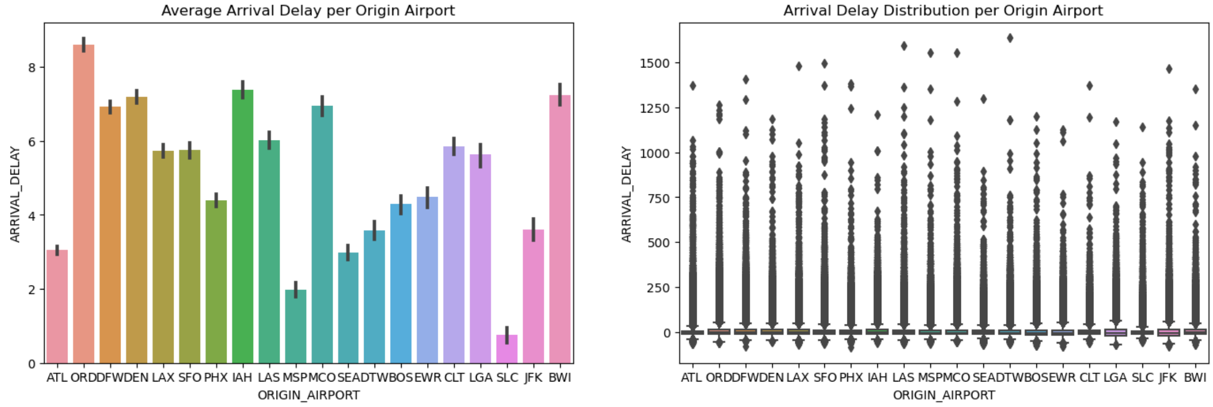


Figure 2.25: Average Arrival Delay per Origin Airport & Arrival Delay Distribution per Origin Airport

Based on the graph's results, no relationship can be drawn between the number of flights from an origin airport and the average Arrival Delay. Taking into account the flight's volume, flights from some departure airports with the biggest number of flights have low mean Arrival Delays (Atlanta (ATL)), whilst flights from other airports with less flight's volume have higher average Arrival Delays (Baltimore-Washington (BWI)). On the contrary, Chicago Airport (ORD), which is the second Departure Airport in terms number of flights, presents flights with the higher mean Arrival Delay. Also, Salt Lake City Airport (SLC), the 18th airport considering flight's volume, has the flights with lowest average Arrival Delay. Therefore, no conclusion can be extracted of the association between volume of flights from the Departure Airport and the delays in the Destination Airport (Arrival Delays)..

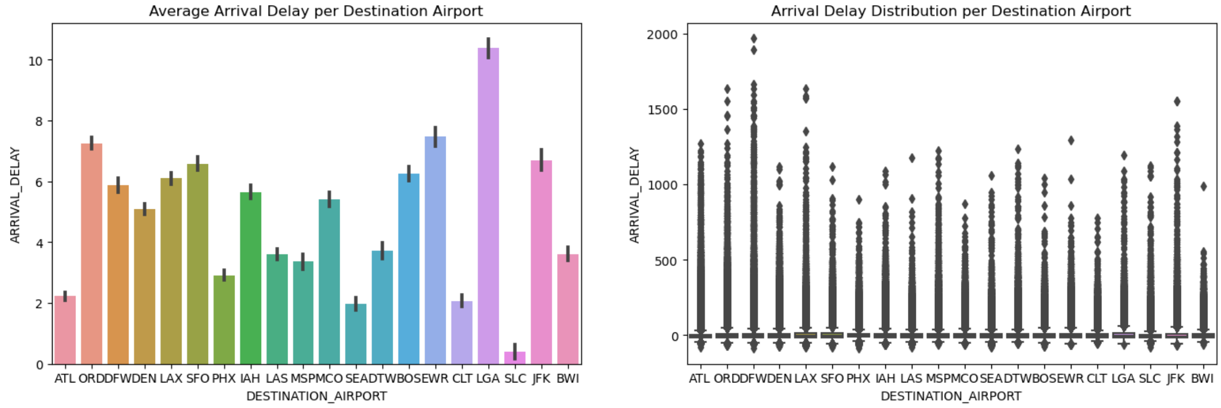


Figure 2.26: Average Arrival Delay per Destination Airport & Arrival Delay Distribution per Destination Airport

The same conclusions can be extracted for the destination airports: there is no connection between the quantity of planes arriving at a certain airport and the mean Arrival Delay or the Arrival Delay distribution. For example, La Guardia - New York Airport (the 16th destination airport regarding the volume of flights) presents the highest average Arrival Delay. On the contrary, Hartsfield-Jackson Atlanta International Airport (ATL), the biggest airport in terms of number of flights, obtains a very low mean Arrival Delay (approximately 2 minutes). Again, Salt Lake City Airport (SLC) (as destination airport) has the lowest average Arrival Delay.

The following analysed variable would be the Air System Delay (National Airspace System (NAS) Delay). It is important to point out that the feature's data is only present in the dataset when the flight's Arrival Delay value is superior than 15 minutes. Thus, to obtain the Air System Delay, dataset must be filtered by the Arrival Delay variable (more than 15 minutes). The subsequent graphs contain the average Air System Delay and the Air System Delay distribution per origin and destination airports:

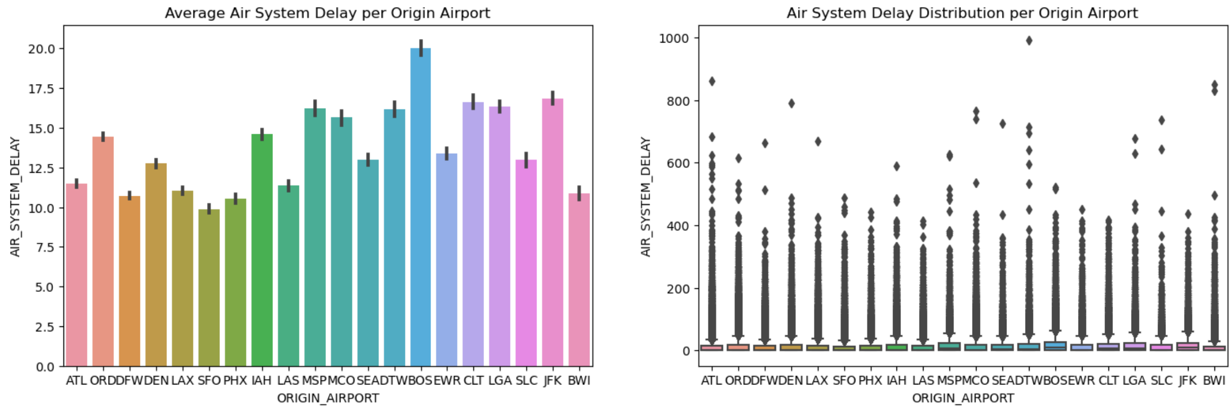


Figure 2.27: Average Air System Delay per Origin Airport & Air System Delay Distribution per Origin Airport

A hypothesis can be established, as, in general, the National Airspace System (NAS) delays are slightly lower for the airports with a highest volume of flights. There are exceptions like Chicago (ORD) and Houston (IAH), where the Air System average delay is a bit higher than the nearest airport in terms of number of national flights. Origin airports with a lower volume of domestic flights often have higher NAS delays. There are inconsistencies for this theory, as Baltimore-Washington (BWI) has one of the lower average Air System delays.

However, this hypothesis doesn't take into account international flights because the dataset is limited to US domestic flights. Some of the US's top international airports, that can absolutely disrupt the conclusions extracted from this dataset are: New York (JFK), Los Angeles (LAX), Newark (EWR) Houston (IAH), Chicago (ORD), Dallas Fortworth (DFW) or Atlanta (ATL).

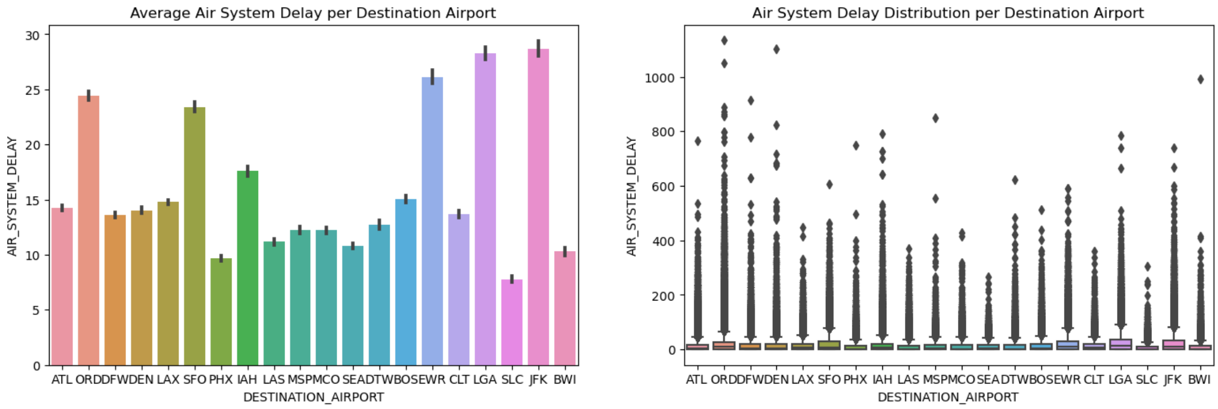


Figure 2.28: Average Air System Delay per Destination Airport & Air System Delay Distribution per Destination Airport

Taking into account the Destination airports, the three higher values for mean Air System Delay correspond to airports with lower volume of flights than others in the dataset (Newark (EWR), La Guardia - New York (LGA) and New York (JFK) airports). However, airports like Salt Lake City (SLC) or Baltimore-Washington (BWI) present the lowest values of average Air System Delay. Even though, in general, the biggest airports in terms of number of landing flights have moderately low values of mean Air System Delay, there are deviations like Chicago O'Hare International Airport (ORD) and San Francisco International Airport (SFO).

The result's variability infers that when considering the Air System Delay feature, each airport (origin or destination) must be taken into account, as the delays heavily depend on individual airports. Also, no relationship can be established between the airport's flight volume and the Air System Delays.

It can be concluded that there is a high variability in mean Arrival Delays, both between the different airports and between the different airlines as well. Some Airlines have higher delay ratios even though the frequency of flights is inferior than for other companies. This means that, in order to achieve a better delay model accuracy, the model must be specific to the carrier and the Departure or Destination Airport, as the location may be an important factor on delay prediction.

2.6. Exploratory Data Analysis: Days of the Week and Months of the Year

The Exploratory Data Analysis (EDA) continues by the influence that the different days of the week and months of the year can exert upon Arrival Delays. First of all, the days of the week are analysed, picking up the average Arrival Delay for each day and, also, the number of flights. The following table shows a tendency that enables us to conclude that number of flights of each day influence the Arrival Delays. As the volume of flights increase, the mean Arrival Delay is higher. This can be verified considering that Saturdays (the day of the week with less volume of flights) have the minimum average Arrival Delay (2 minutes). Also, Wednesdays and Mondays (high volume of flights) have the maximum mean Arrival Delays throughout the week days (6 minutes).

Day of the Week	Mean Arrival Delay (min)	Number of Flights
Monday (1)	6	841.794
Tuesday (2)	4	827.399
Wednesday (3)	4	843.242
Thursday (4)	6	857.886
Friday (5)	5	851.387
Saturday (6)	2	689.745
Sunday (7)	4	802.555

Table 2.4: Mean Arrival Delay and Number of Flights per Day of the Week

The different days of the week are compared taking into account the average Arrival Delay. Most flights are concentrated in daily days, between Mondays and Fridays. Weekend days (Saturday and Sunday) have a lower number of total flights per day.

Average Arrival Delay is slightly higher for the week days that have a greater number of flights. This can be specially seen on Saturdays, the week day with the lower number of flights, which also has by far the lower Average Arrival Delay of all the days (2 minutes). This circumstance proofs the hypothesis that as the frequency of flights increase, Arrival Delays also increase.

Next, the average Arrival Delays based on the months of the year (2015) are studied, along with the number of flights for each month. From the data contained in the table, it can be extracted that the highest mean Arrival Delay occurs in June, starting the summer holidays season. This season (June, July and August) stands out because of the high volume of flights, which generally entails high average Arrival Delays (10, 6 and 5 minutes, respectively). However, the second highest value of mean Arrival Delay takes place in February, which is the month with the minimum number of flights. It is also important to point out that September and October record a negative value of average Arrival Delay, which means that the flights tend to land before the scheduled landing time.

Month of the Year	Mean Arrival Delay (min)	Number of Flights
January (1)	6	457.013
February (2)	8	407.663
March (3)	5	492.138
April (4)	3	479.251
May (5)	4	489.641
June (6)	10	492.847
July (7)	6	514.384
August (8)	5	503.956
September (9)	-1	462.153
October (10)	-1	482.878
November (11)	1	462.367
December (12)	6	469.717

Table 2.5: Mean Arrival Delay and Number of Flights per Month of the Year

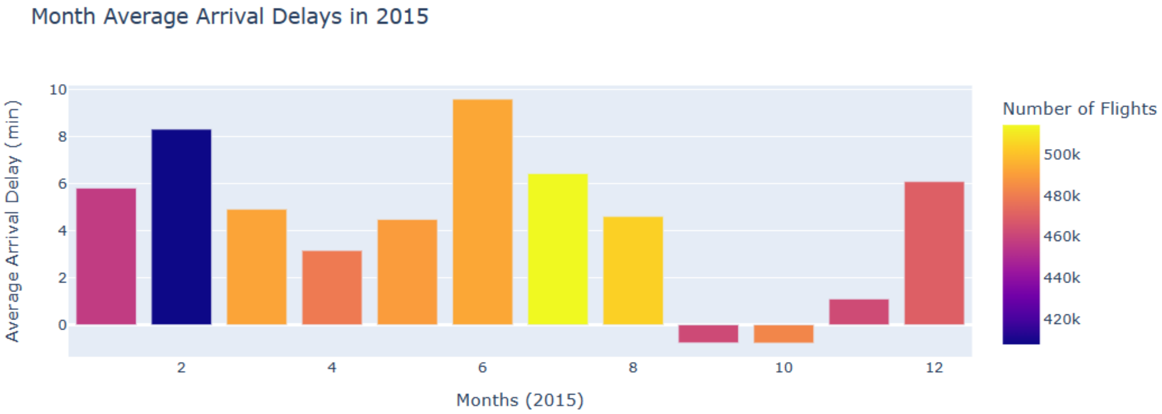


Figure 2.29: Month Average Arrival Delays in 2015

The height of the bars from the previous graph represent the Average Arrival Delay (delay ratio) for the different months of the year. The different colours of the bars indicate the number of flights for each month considering the scale for the number of flight operations. There is a clear tendency that shows the increment in flights during the Spring and Summer seasons, while there is a volume reduction during Fall and Winter. No sequence can be established between the number of flights per month and the registered mean Arrival Delays for those months.

The following calendar heat maps contain the volume of flights and the number of delayed flights for each day in the year 2015. Summer months (June, July and August) group a high number of flights, containing days with a great volume of delays. Also, some of the days from December and January that collect a high number of delayed flights have to be with Christmas holidays outward and return journey periods. "The February 2015 North American ColdWave", and its related bad weather, originated some high delay's volume days in February and March.

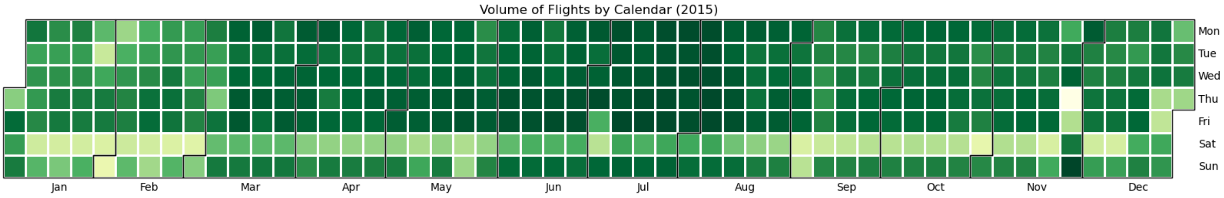


Figure 2.30: Volume of Flights by Calendar (2015)

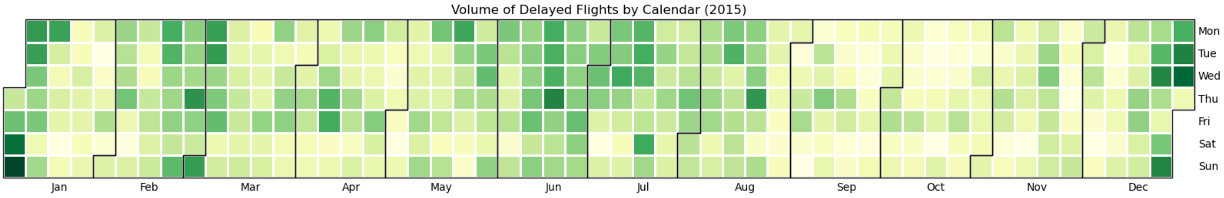


Figure 2.31: Volume of Delayed Flights by Calendar (2015)

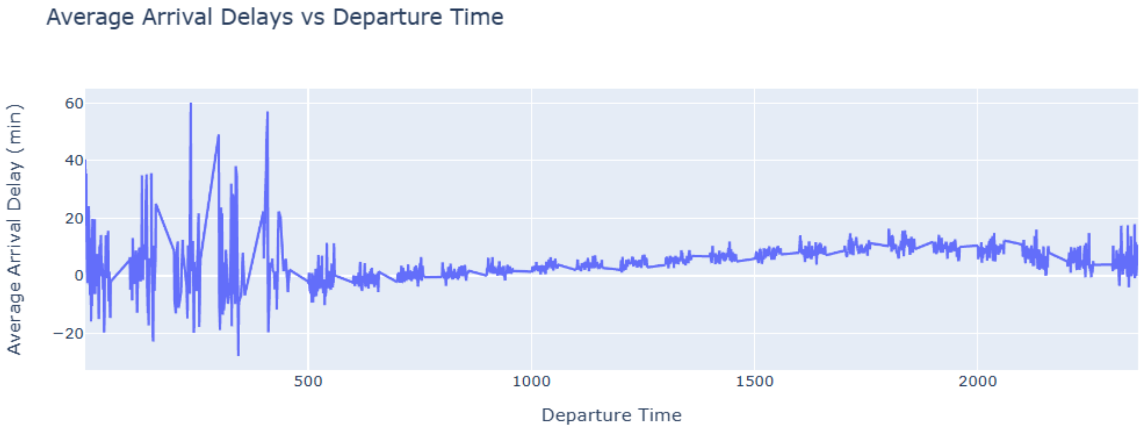


Figure 2.32: Average Arrival Delays vs Departure Time

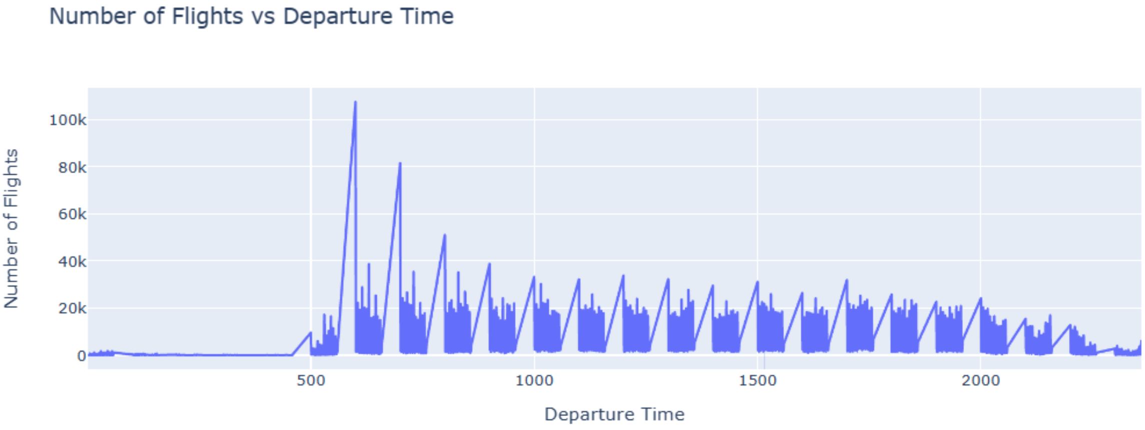


Figure 2.33: Number of Flights vs Departure Time

The day-night difference and the fact of the airport activity reduction during the night, suggests that Scheduled Departure (Departure Time) is an important variable in the creation of models for flight delays. From 21:00 to 6:00, the number of flights decreases dramatically comparing with day hours, even going so far as to not having any flights at certain morning hours.

Considering the mean Arrival Delay, it increases with the Departure hour as the day goes on. Starting the day (5:00 - 6:00 in the morning), flights depart on time and the delay increases constantly until the end of the day. The graph shows that the average Arrival Delay establishes a maximum value at around 21:00 and then, it decreases its value along with the reduction in the number of passengers. This tendency enables to state that, as the frequency of flights increase, the delays grow.

2.7. Exploratory Data Analysis: Conclusions

Flight Delays take place due to several possible factors, such as origin or destination airports, ground problems or specific airlines. Delays can be caused by circumstances such as weather conditions, departure and arrival queues, late aircraft, National Airspace System (NAS) or security reasons. By Exploratory Data Analysis (EDA), some patterns that cause flight delays can be discovered and the reasons behind this phenomenon. Conclusions that can be extracted from the Exploratory Data Analysis (EDA) performed to the studied dataset are:

- Departure Delay variable has a high correlation with Arrival Delay. Even though, highly correlated features should be removed to construct the model, it is interesting to predict Arrival Delays, knowing that the flight was delayed in the departure and the magnitude of the delay (Figure 2.1).
- Distance, Air Time and Scheduled Time have a very similar distribution (highly correlated). Only one of these variables will be used for the model creation and the other two will be dropped (Figures 2.5 and 2.6).
- Scheduled Time, Departure Delay, Departure Taxi, Arrival Taxi, Distance and Air Time are highly skewed (great right tail) and they can be transformed before building a model (Figures 2.3, 2.4 and 2.5).
- Extreme Weather is the main reason behind most of flights that are cancelled. Usually, Security isn't the cause for flight's cancellation (Figures 2.10 and 2.11).
- The vast majority of delays (when they occur) are short in time (Figures 2.8, 2.9 and 2.14).
- Delays have peaks and are concentrated on the o'clock hours (Scheduled Departure) (Figures 2.15 and 2.16).
- There are three main causes that produce most of the flight delays: Late Aircraft, Airline or Carrier and Air System, in descending order. Late Aircraft Delay is the most important reason, being the volume of delays greater than for the other causes (Figure 2.17).
- No connection can be established between the volume of flights from a carrier and the average Arrival Delay and Airline Delay. Every carrier must be considered independently, as it is an important feature when predicting delays (Figures 2.19 and 2.20).
- Independently of the airline, the proportion of large delays (more than 45 minutes) with respect to the total number of flights is very low (Figure 2.21).
- Arrival Delays have lower values than Departure Delays. Increasing flight speed can reduce the delays. Hawaiian Airlines Inc. (HA), is the only carrier that achieves higher arrival than departure delays (Figure 2.22).
- The mean Departure Taxi time for all the studied airports is much higher than the average Arrival Taxi for the same airports (Figure 2.24).
- No relationship can be drawn between the number of flights from an origin or destination airport and the Arrival Delay (Figures 2.25 and 2.26).

- In general, the National Airspace System (NAS) delays are slightly lower for the airports with a greater volume of flights. Also, origin airports with fewer domestic flight departures frequently experience longer Air System delays. However, there are some exceptions and the variability concludes that, taking into account Air System delays, each airport (origin or destination) must be considered, as the delays vary depending on individual airports (Figures 2.27 and 2.28).
- There is a high variability in average Arrival Delays, both between the different airports and the different airlines. To construct an accurate model, it must be specific for every airline and every origin or destination airport, as the location could be a crucial characteristic in the prediction of flight delays.
- Average Arrival Delay is slightly higher for the week days that have a greater number of flights. As the frequency of flights increase, Arrival Delays also increase (Table 2.4).
- No association can be set between the volume of flights per month and the month's mean Arrival Delay (Table 2.5 and Figure 2.29).
- Departure time is an important factor in the creation of models for flight delay prediction. From sunrise to the end of the day, there is an evolution, as the flights depart on time in the morning and the delay increases constantly until the end of the day (Figures 2.32 and 2.33).
- Average Arrival Delays vary widely between the different airports and carriers. This is important since it suggests that a model that is specific to the airline and the origin or destination airport will predict more accurately the delays.

3

Machine Learning Techniques

Considering Machine Learning techniques, there is a great variety of applicable models depending on the type of available data and the data that is going to be obtained. Machine Learning models can be classified into: Supervised and Unsupervised Learning.

3.1. Supervised Learning

This type of Machine Learning is characterised by the fact that the training data that is delivered to the algorithm has the expected solutions called labels. Inside this algorithms two groups can be distinguished: the regression algorithms and the classification algorithms. The difference between them is that the regression algorithms return a continuous value, while the classification algorithms return a categorical value.

3.1.1. Regression

The objective of this class of algorithms is to make a classification from training data, where the expected value is a lineal combination of the characteristics. Historical data from a dataset is used, with one or more of its features, and, from that data, using regression, a regression or estimation model is built for predictions to be made. The key algorithm has the following form:

$$y = a_0 + a + b_1 \cdot x_1 + b_2 \cdot x_2 + ... + b_n \cdot x_n \tag{3.1}$$

In regression there exists two types of variables: a dependent variable and one or more independent variables.

The dependent variable can be seen as the "state", "target" or "final goal" that is studied and predicted, and the independent variables, also known as explanatory variables, can be seen as the "causes" of those "states".

The independent variables are shown conventionally by the letter x and the dependent variable is notated by y. A regression model relates y, or the dependent variable, to a function of x, the independent variables.

The key point in the regression is that the dependent value should be continuous and cannot be a discrete value. However, the independent variable or variables can be measured on either a categorical or a continuous measurement scale.

Basically, there are 2 types of regression models: Simple Regression and Multiple Regression. Simple Regression is when one independent variable is used to estimate a dependent variable. Linearity of regression is based on the nature of relationship between independent and dependent variables. When more than one independent variable is present, the process is called multiple linear regression. Again, depending on the relation between dependent and independent variables, it can be either linear or non-linear regression. Essentially, regression is used when the aim is to estimate a continuous value.

There are many regression algorithms and each of them has a specific condition to which their application is best suited.

3.1.2. Simple Linear Regression

Linear Regression can be used to predict a continuous value, by using other variables. It is the approximation of a linear model used to describe the relationship between two or more variables, being easy to use and understand and highly interpretable.

In Simple Linear Regression, there exists two variables: a dependent variable $\hat{y}$ and an independent variable x. The independent variable is used to estimate the dependent variable or target value. In a Simple Regression problem, the form of the model would be represented by the following equation:

$$\hat{y} = \theta_0 + \theta_1 \cdot x \tag{3.2}$$

In this equation, θ_0 and θ_1 are the parameters that must must be adjusted. θ_1 is known as the slope or gradient of the fitting line (coefficient) and θ_0 is known as the intercept. The key point in the Simple Linear Regression is that the dependent variable must be continuous and cannot be a discrete value. However, the independent variable can be measured on either a categorical or continuous measurement scale. An example of plotted Simple Linear Regression is shown next:

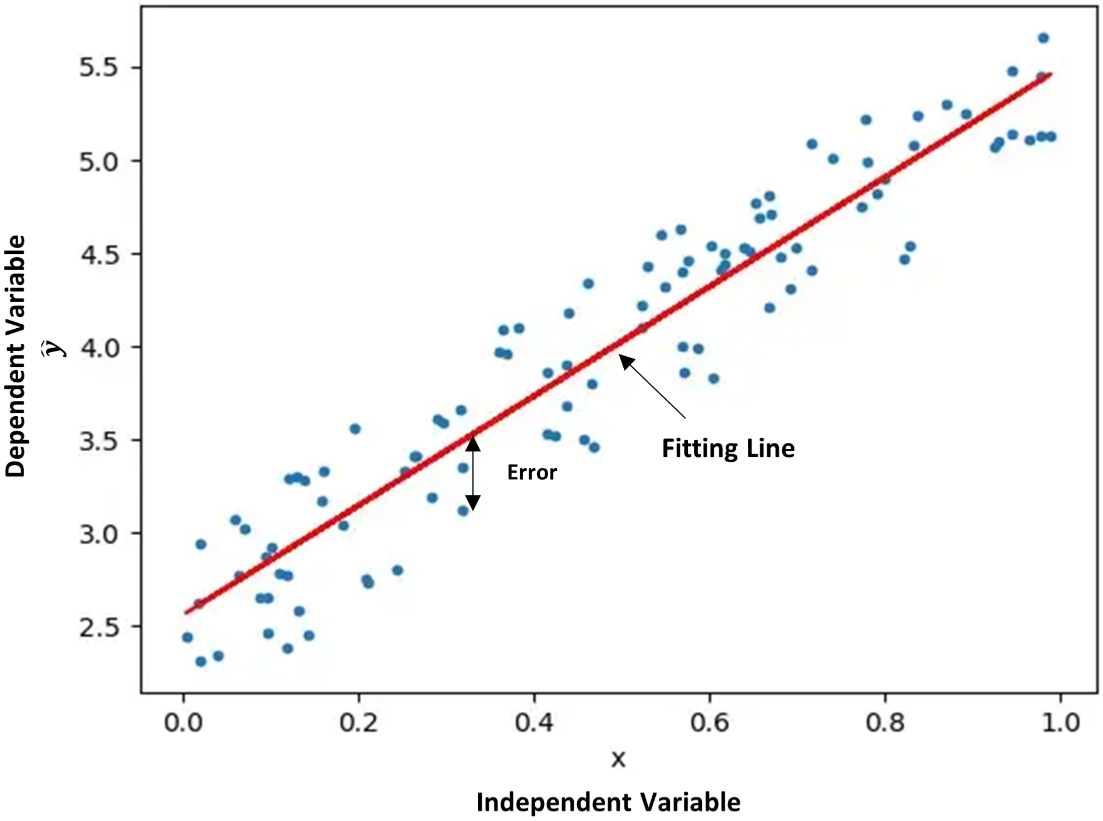


Figure 3.1: Simple Linear Regression

Linear Regression estimates the coefficients of the line, θ_0 and θ_1, which must be calculated to find the best line to "fit" the data. The coefficients can be calculated using a mathematical or an optimisation approach.

The error is the distance from the data point to the fitting line. The mean of all residual errors shows how the line fits with the whole dataset. Mathematically, it can be shown by the Mean Squared Error (MSE). The goal is to find a line where the mean error of the prediction using the fit line should be minimised.

To mathematically calculate the coefficients of the fit line, the mean of the independent ($\bar{x}$) and dependent or target columns ($\bar{y}$) from the dataset must be obtained. The slope of the line, θ_1, is estimated using the following equation:

$$\theta_1 = \frac{\sum_{i=1}^{s} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{s} (x_i - \overline{x})^2} \tag{3.3}$$

The x_i and y_i in the equation refer to the fact that the calculations need to be repeated across all values in the dataset. The intercept of the line, θ_0, also called bias coefficient, derives from the next expression:

$$\theta_0 = \overline{y} - \theta_1 \cdot \overline{x} \tag{3.4}$$

3.1.3. Multiple Linear Regression

Multiple Linear Regression is when multiple independent variables (categorical or continuous measurement scale) are used to estimate a dependent variable (continuous measurement scale). Categorical independent variables must be converted into numerical variables for its use in the algorithm. In summary, it is the extension of the Simple Linear Regression model and it needs a linear relationship between the dependent variable and each of the independent variables.

There are two principal applications for Multiple Linear Regression. First, it can be used to identify the strength of the effect that the independent variables have on the dependent variable. Second, it can be used to predict the impact of changes, enabling the knowledge of how the dependent variable changes when there is a change on the independent variables.

As is the case with Simple Linear Regression, Multiple Linear Regression is a method for predicting a continuous variable, using multiple variables, called independent variables, or predictors. Independent variables are used to predict the value of the target variable, which is also called the dependent variable. In Multiple Linear Regression, the target value, $\hat{y}$, is a linear combination of independent variables, x. The model has the form of the following equation:

$$\hat{y} = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + ... + \theta_n \cdot x_n \tag{3.5}$$

This method is very useful because it can examine which variables are significant predictors of the outcome variable, finding out how each feature impacts the outcome variable.

The form of the model, can be mathematically expressed as a product of two vectors: the parameters vector and the feature set vector. The equation for a multi-dimensional space is:

$$\hat{y} = \theta^T X \tag{3.6}$$

$$\theta^T = [\theta_0, \theta_1, \theta_2, ..., \theta_n] \tag{3.7}$$

$$X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ ... \end{bmatrix} \tag{3.8}$$

X is the vector of the feature set and θ is an n-by-one vector of unknown parameters (vector of coefficients) in a multi-dimensional space, also called weight vector of the regression equation. This name is due to the fact that a higher coefficient related to an independent variable means a higher importance of this variable in comparison with other independent variables, estimating the relative importance of the different independent variables.

The first element of the feature set has to be set to a value of 1, because it turns the θ_0 parameter into the intercept or bias parameter when the feature set vector is multiplied by the parameter vector. In summary, as it is a multi-dimensional space, the principal goal is to find the hyperplane that best fits the data by estimating the values for θ vector (optimised parameters) and minimising the prediction's error. This error is the distance from the data point to the fitted regression model. The Mean Squared Error (MSE) or mean of all residual errors must be minimised in order for the model to faithfully represent the dataset.

There are many ways to estimate the value of the coefficients θ. The most common methods for estimating the optimised parameters are:

- **Ordinary Least Squares**: This method estimates the values of the unkonwn coefficients by minimising the Mean Square Error (MSE). This method picks the parameters of a linear function of a set of explanatory variables by minimising the sum of the squares of the differences between the target dependent variable and the predictions by the linear function. In other words, it minimises the Sum of Squared Errors (SSE) or Mean Squared Error (MSE) between the target variable (y) and the predicted variable ($\hat{y}$) over the data in the dataset. It uses the data as a matrix and linear algebra operations are performed to estimate the optimal values for θ. It is a time consuming method and can only be considered if the number of rows in the dataset is less than 10,000.

  Ordinary Least Squares (OLS) aims the optimal parameters making use of the following methods:

  - Analytically solving the model parameters using closed-form equations.
  - Optimization algorithm (Gradient Descent, Stochastic Gradient Descent, Newton's Method, etc.).

- **Optimization Approach**: It is a process of optimizing the values of the coefficients by iteration, minimising the error of the model on the training data. Gradient Descent, which is a good approach with large datasets, starts optimization with random values for each coefficient, calculates the errors and minimises them by changing the coefficients in multiple iterations.

It is important to point out that too many independent variables may result in an over-fit model, reducing the accuracy of the model as it is not general enough to be used for prediction.

3.1.4. Non Linear Regression

If the data shows a curvy trend (non-linear relationship), then linear regression will not produce very accurate results when compared to a non-linear regression, simply because, as the name implies, linear regression presumes that the data is linear. So, it requires a special estimation method of the Non-Linear Regression procedure.

In fact, many different regressions exist that can be used to fit whatever the dataset looks like, such as quadratic, cubic, and it can go on and on to infinite degrees. Non-Linear Regression is a method to model the non-linear relationship between the independent variables x and the dependent variable y. Basically, any non-linear relationship can be expressed as a non-linear function. There are different types of non-linear functions such as: exponential, logarithmic, quadratic, cubic, sigmoidal or logistic, etc.

If it is assumed that the model for the data points is a polynomial function, it is called "Polynomial Regression", where the relationship between the independent variable x and the dependent variable y is modelled as an N-degree polynomial in x. It can be represented by the following equation:

$$\hat{y} = \theta_0 + \theta_1 \cdot x + \theta_2 \cdot x^2 + ... + \theta_n \cdot x^n \tag{3.9}$$

The parameters of the model (θ_s) must be estimated and the fitted model used to predict unknown or future cases. The estimated parameters make the model fit perfectly to the applied data. It is important to stand out that a Polynomial Regression model can be transformed into a Multiple Linear Regression model, as it is shown below:

$$x_1 = x \qquad x_1 = x^2 \qquad x_n = x^n$$

$$\hat{y} = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + ... + \theta_n \cdot x_n \tag{3.10}$$

This Multiple Linear Regression model can be solved using Least Squares, which is a method for estimating the unknown parameters in a Linear Regression model, by minimising the sum of the squares of the differences between the observed dependent variable in the given dataset (y) and those predicted by the linear function ($\hat{y}$).

There are many types of regression to choose from and it's important to choose a regression that fits the data the best.

Non-Linear Regression is a method to model a non-linear relationship between the dependent variable and a set of independent variables. For a model to be considered non-linear, $\hat{y}$ must be a non-linear function of the parameters θ, not necessarily the features x. The model is non-linear by parameters, being characterised by the difficulty of parameter estimation. A non-linear equation can adopt the shape of exponential, logarithmic, logistic, or many other types.

To discern if a problem is linear or non-linear, in addition to plotting it and visually figuring it out, the correlation coefficient between independent and dependent variables can be calculated, concluding that if for all variables its value is 0.7 or higher there is a linear tendency, and it's not appropriate to fit a Non-Linear Regression.

3.1.5. Model Evaluation and Performance Evaluation Metrics

The goal of regression is to build a model to accurately predict an unknown case. Evaluation must be performed after building the model. Two types of evaluation approaches can be used to achieve this goal: train and test on the same dataset, and train and test split.

The first evaluation approach is the simplest one: train and test on the same dataset. The model is trained on the entire dataset and then, it is tested using a portion of the same dataset. In general, when the test is with a dataset in which the target value is known for each data point, a high percentage of accurate predictions for the model is obtained. This evaluation approach would most likely have a high "training accuracy" and a low "out-of-sample accuracy", since the model knows all of the testing data points from the training phase.

Training accuracy is the percentage of correct predictions that the model makes when using the test dataset. There is a possibility that having a high training accuracy may result in an "over-fit" of the data. This means that the model is overly trained to the dataset, which may capture noise and produce a non-generalised model.

Out-of-sample accuracy is the percentage of correct predictions that the model makes on data that the model has not been trained on. The train and test procedure on the same dataset will most likely have low out-of-sample accuracy due to the likelihood of being over-fit. It's important for the model to have high, out-of-sample accuracy, because the purpose of it is, of course, to make correct predictions on unknown data.

To improve out-of-sample accuracy, another evaluation approach called "Train/Test Split" can be used. Train and Test Split involves splitting the dataset into training and testing sets, which are mutually exclusive. The model is built on the training set. Then, the test feature set is passed to the model for prediction. Finally, the predicted values for the test set are compared with the actual values of the testing set. This will provide a more accurate evaluation on out-of-sample accuracy because the testing dataset is not part of the dataset that has been used to train the data.

Another evaluation model is called "K-Fold Cross-Validation". It performs multiple train and test splits using the same dataset, where each split (fold) is different. Then, the result is averaged to produce a more consistent out-of-sample accuracy. That is, the accuracy of each fold is averaged, considering that each fold is distinct, where no training data in one fold is used in another.

Evaluation metrics are used to explain the performance of a model and provide a key role in the development of a model, as it provides insight to areas that require improvement (accuracy metrics for model evaluation).

In the context of regression, the error of the model is the difference between the data points and the trend line generated by the algorithm. Since there are multiple data points, an error can be determined in multiple ways.

One of the simplest metrics to calculate the accuracy of a model is to compare the actual values, y, with the predicted values, which is noted as $\hat{y}$, for the testing set. Below, different model evaluation metrics (performance metrics) are exposed:

Mean Absolute Error (MAE) is the mean of the absolute value of the errors. The error of the model (average error) is calculated as the average difference between the predicted and actual values for all the rows.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \widehat{y}_j| \tag{3.11}$$

Mean Squared Error (MSE) is the mean of the squared error. It's more frequently used than the Mean Absolute Error (MAE) because the focus is centered more towards large errors. This is due to the squared term exponentially increasing larger errors in comparison to smaller ones. Mean Squared Error (MSE) is a risk function, as it is the expected value of the squared error loss.

$$MSE = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \widehat{y}_j\right)^2 \tag{3.12}$$

Root Mean Squared Error (RMSE) is the square root of the Mean Squared Error (MSE). It is the average absolute difference between the predicted and the actual values across the dataset. This is one of the most popular of the evaluation metrics because Root Mean Squared Error (RSME) can be interpreted in the same units as the response vector (y (actual values) units) making it easy to compare its information.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \widehat{y}_j\right)^2} \tag{3.13}$$

Relative Absolute Error (RAE), also known as Residual Sum of Squares, where $\bar{y}$ is a mean value of y, takes the total absolute error and normalises it by dividing by the total absolute error of the simple predictor.

$$RAE = \frac{\sum_{j=1}^{n}|y_j - \widehat{y}_j|}{\sum_{j=1}^{n}|y_j - \bar{y}_j|} \tag{3.14}$$

Relative Squared Error (RSE) is very similar to Relative Absolute Error (RAE), but is widely adopted by the data science community, as it is used for calculating **R-Squared**. R^2 is a popular metric for the accuracy (performance) of the model (it is not an error), model's ability to predict variances in the dataset. The higher the R-Squared, the better the model fits the data. Furthermore, R^2 also measures the dependence of the target feature on the independent variables of the flight dataset. The best R^2 value is 1.0 and it can be negative (the model can deteriorate).

$$RSE = \frac{\sum_{j=1}^{n}\left(y_j - \widehat{y}_j\right)^2}{\sum_{j=1}^{n}\left(y_j - \overline{y}_j\right)^2} \tag{3.15}$$

$$R^2 = 1 - RSE \tag{3.16}$$

Where, n is the total number of values, y are the actual values, $\hat{y}$ corresponds to the predicted values and $\overline{y}$ is the average or mean value of all the real values of the testing set. R^2 and $RMSE$ are the most important performance metrics, as the regressor's main task is to predict, with the minimum error, the flight delay period for every flight that is delayed. Each of these metrics can be used for measuring and quantifying the prediction. The choice between the different model evaluation metrics depends on the type of model, the data type and the domain of knowledge.

3.2. Unsupervised Learning

This type of Machine Learning algorithms don't learn from labels, they learn directly from features. They are generally used to make clusters or generate groups with similar characteristics. They also have other applications such as dimension reduction of certain problems to simplify its treatment or for data visualisation. Three types of algorithms can be differentiated in this category:

- **Clustering**: This algorithms are responsible for searching and generating groups from the data, being capable of finding patterns and grouping the data using those patterns. They are very useful for a lot of tasks that relate to information classification. This algorithms can be divided into three classes depending on the parameters that are used to form the groups:
  - **Partitional**: Data is divided in non-overlapped groups. Generally, this techniques need that the number of created groups are specified. Besides, they are normally non-deterministic algorithms.
  - **Hierarchical**: Data is divided generating data hierarchies, producing hierarchical trees called dendograms. Besides, they are normally deterministic algorithms.
  - **Based on densities**: Data is divided taking into account the point densities in different regions. The groups are assigned in zones where there are higher point densities.
- **Visualisation and Dimension Reduction**: This algorithms allow to do dimension reduction of certain problems by finding patterns in the data in order to simplify the problems.
- **Anomaly Detection**: This algorithms search for points in the data that stand out from normal points.

4

Two-Stage Machine Learning Engine: Results and Analysis

4.1. Introduction

An essential instrument that can assist airlines aviation authorities in efficiently reducing aircraft delays is a structured prediction system. The goal of this analysis is to develop a two-stage Machine Learning engine that can accurately forecast a flight's Arrival Delay in minutes using real-time flights. First, a classifier predicts whether the flight will be delayed or not, and if the delay occurs, then, a regression model predicts the Arrival Delay in minutes.

As the EDA was performed, it is discovered that the flight dataset is skewed. This imbalance is present due to the Bureau of Transportation Statistics (BTS) and Federal Aviation Administration (FAA) definition of a flight delay. It states that a flight is considered "delayed" if it landed fifteen minutes after the planned time indicated in the Carriers Computerized Reservations Systems (CRS). The flight is therefore considered delayed if the delay time exceeds fifteen minutes and it is established by a combination of the five delay factors (Airline, Weather, Late Aircraft, Air System and Security).

Due to class imbalance, there is a bias towards the majority class, Not Delayed Flights (Class 0). The dataset bias (towards non delayed flights (Class 0)) reduces the performance and results of Class 1 compared to Class 0. In order to remove the bias, the data can be sampled using different sampling methods before the Classification process, including Random Undersampler, Random Oversampler and SMOTE (Synthetic Minority Oversampling Technique).

After preprocessing the dataset, different classifier models are constructed and compared by their performance metrics. Afterwards, some regression models are created and analysed (only picking flights categorised as delayed by the dataset). Classifiers and regressors evaluation metrics allow to compare between the models, taking into account unsampled data and the previously enumerated sampling techniques and selecting the ideal classifier and regressor based on the metrics. To conclude, this project builds a two-stage predictive Machine Learning engine with a classifier to identify which flights will be delayed using flight data and a regressor to forecast the length of the delay for those flights classified as delayed by the classifier. Different pipelined models are evaluated to choose the classifier and regressor models that identify and predict, more accurately, delayed flights and the Arrival Delay period.

4.2. Preprocessing

The raw dataset is subject to data cleaning by removing missing data and redundant attributes. Attributes that directly affect flight delays are identified and unrelated features (like attributes associated to diverted or cancelled flights) are withdrawn. Also, categorical variables are encoded and the dataset is splitted between training and testing sets, by selecting 80% of the data points for training and the remaining 20% of the data points for testing. Dataset splitting is performed by a scikit learn function named *train_test_split*. Encoding categorical values is carried out by the *LabelEncoder* function, which is used for converting categorical features, such as AIRLINE, ORIGIN_AIRPORT and DESTINATION_AIRPORT, into numerical form. This is an important procedure for preprocessing, as this numeric form is needed for the correct functioning of Machine Learning algorithms.

FLIGHT_STATUS is a generated feature to determine whether the flight arrives on time (less than 15 minutes ARRIVAL_DELAY) or it is delayed (more than 15 minutes ARRIVAL_DELAY). FLIGHT_STATUS value is 0 if flights arrive early or on time and 1 for flights that are delayed. Another created column is DAY_OF_WEEK, which sets a value of 0 for weekdays and 1 for weekends. This feature is set due to the conclusion extracted from the Exploratory Data Analysis (EDA), that the number of flights decreases for the weekends with respect to the weekdays and, by consequence, also the duration and number of delays decrease.

The degree to which two variables change, one variable respect to the other variable, is measured by the Correlation Coefficient. Both the strength and the direction of the link are described by the coefficient. Also, the strength and the direction of the connection between one continuous variable and one binary categorical variable is assessed using a Point-Biserial Correlation. SCHEDULED_TIME, AIR_TIME and DISTANCE are features that are highly correlated to each other, as it can be proved with the results obtained in the correlation matrix from the previous Exploratory Data Analysis (EDA). The following representation shows the different features Point Biserial Correlation with the target variable (FLIGHT_STATUS):

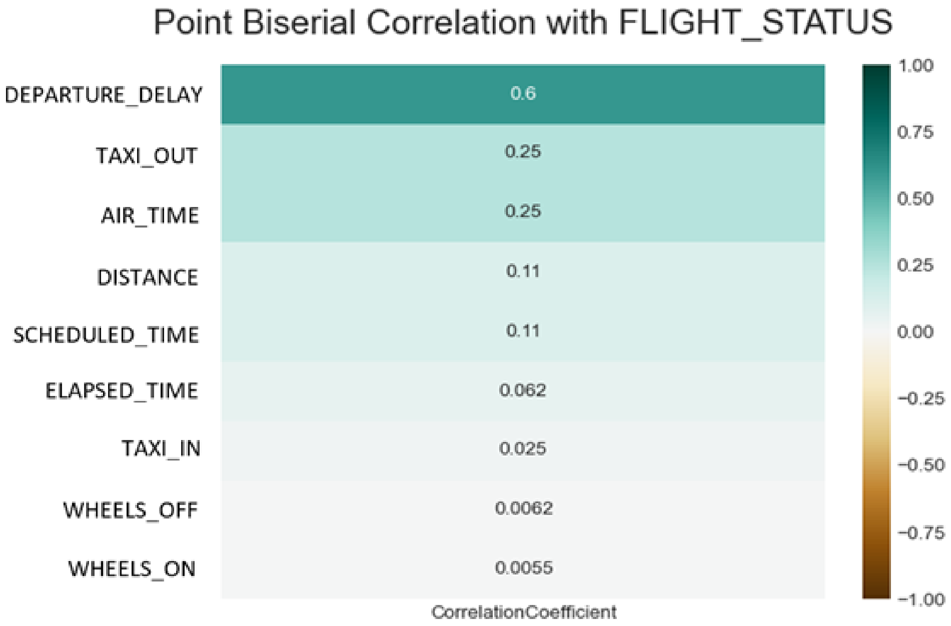


Figure 4.1: Point Biserial Correlation with Flight_Status

In this case, AIR_TIME has the greatest value of Point Biserial Correlation with FLIGHT_STATUS, when compared to the other attributes (SCHEDULED_TIME and DISTANCE, which are removed from the model), so it is the variable considered for modelling. Also, attributes with a low value of Point Biserial Correlation can be removed, as their influence with respect to flight delay prediction is negligible.

For the predictor variable (dependent variable), the necessary features for classification and prediction are chosen and are the following: FLIGHT_STATUS and ARRIVAL_DELAY. For the independent variable, the features related to time and arrival are dropped, because they leak information about the target variables (FLIGHT_STATUS and ARRIVAL_DELAY). These features are the following: SCHEDULED_ARRIVAL, ARRIVAL_TIME and ARRIVAL_DELAY. Furthermore, prediction features, FLIGHT_STATUS and ARRIVAL_DELAY (selected features for the dependent variable) are also dropped for the independent variables. Features that are finally selected for the Independent Variable are the following: MONTH, DAY, DAY_OF_WEEK, AIRLINE, ORIGIN_AIRPORT, DESTINATION_AIRPORT, SCHEDULED_DEPARTURE, DEPARTURE_TIME, DEPARTURE_DELAY, TAXI_OUT, WHEELS_OFF, ELAPSED_TIME, AIR_TIME, WHEELS_ON, TAXI_IN, AIR_TIME and WEEKDAY_TYPE.

4.3. Dataset Class Imbalance

An imbalanced dataset is the case where each class has a different representation in the dataset. It is clear from the classifier evaluation metrics (presented in Section 4.5) that the obtained results for Class 1 (Delayed) are worse than the Class 0 results (Not Delayed). This is due to the dataset inherent bias towards Not Delayed Flights. This bias affects the classifier models performance of Class 1 (outcome of flight delay prediction) relative to Class 0, due to the skewed class distribution.

When the distribution of classes is highly skewed (data imbalance) in a dataset, the model will predict the majority class for every prediction and obtain a great value of classification accuracy. This situation is called the **Accuracy Paradox** and, consequently, accuracy isn't the best performance metric for imbalanced datasets. It is not a reliable metric because it only reflects the correctly classified flights. In classification problems, the focus is placed on the prediction mistakes and the target class. However, other standard metrics, such as Precision and Recall, are used to analyse the performance of the models.

The percentage of delayed (represented by 1.0) and not delayed (represented by 0.0) flights for the imbalanced dataset (unsampled data) is illustrated by the following bar chart. Approximately 82% of the flights from the dataset were not delayed (18% of dataset flights were delayed by more than 15 minutes).

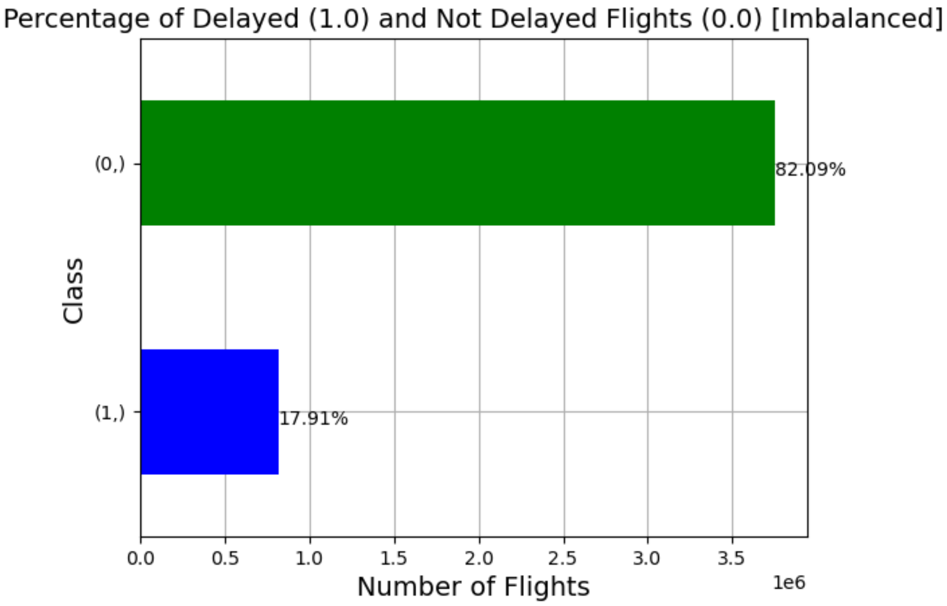


Figure 4.2: Percentage of Delayed and Not Delayed Flights (Unsampled Data)

The different models achieved a higher Precision, Recall, and F1-Score values for non-delayed flights (Class 0) compared to delayed flights (Class 1) (Section 4.5). Due to imbalance, the scores of minority classes are lower than those from majority classes. This is due to the dataset's skewness towards non-delayed flights (Class 0).

In order to overcome this bias, the sample must be transformed (resampling) to gain equality between the two classes (balance the data) and improve the minority class scores. As it can checked by the bar graphs of all the sampling techniques, the percentage of delayed and not delayed flights of the training data is exactly 50%, for both of the classes. Different sampling techniques can be applicated to balance the dataset:

- **Under Sampling**: Class 0 (Not Delayed flights), which is the majority class is under sampled with the intention to balance the distribution. Random Undersampler technique is applied, where the majority class is down-sampled by randomly removing different majority predicted class flights, allowing to reduce its influence in the algorithm and obtaining the desired class distribution.

  Considering, Under Sampling techniques, the principal benefit relates with large training data sets. These techniques lower the number of training data samples, improving run time and storage issues. This advantage also produces a handicap, as it may omit vital data that could be relevant for creating rule classifiers. The chosen data sample by Under Sampling techniques can also be biased, not being a true reflection of the population. As a result, the actual test data set can produce erroneous results.

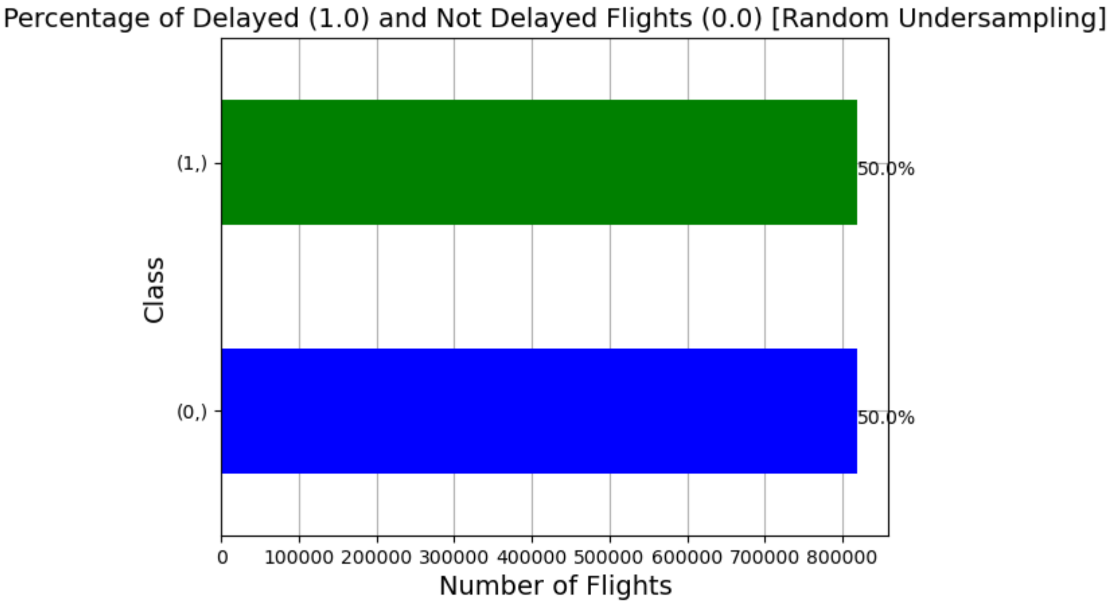


Figure 4.3: Percentage of Delayed and Not Delayed Flights (Random Undersampling)

- **Over Sampling**: Class 1 (Delayed flights), which is the minority class is over sampled with the intention to balance the distribution. Random Oversampler technique is applied, where the minority class is up-sampled by randomly duplicating the minority predicted class flights, allowing to increase its influence in the algorithm and obtaining the desired class distribution.

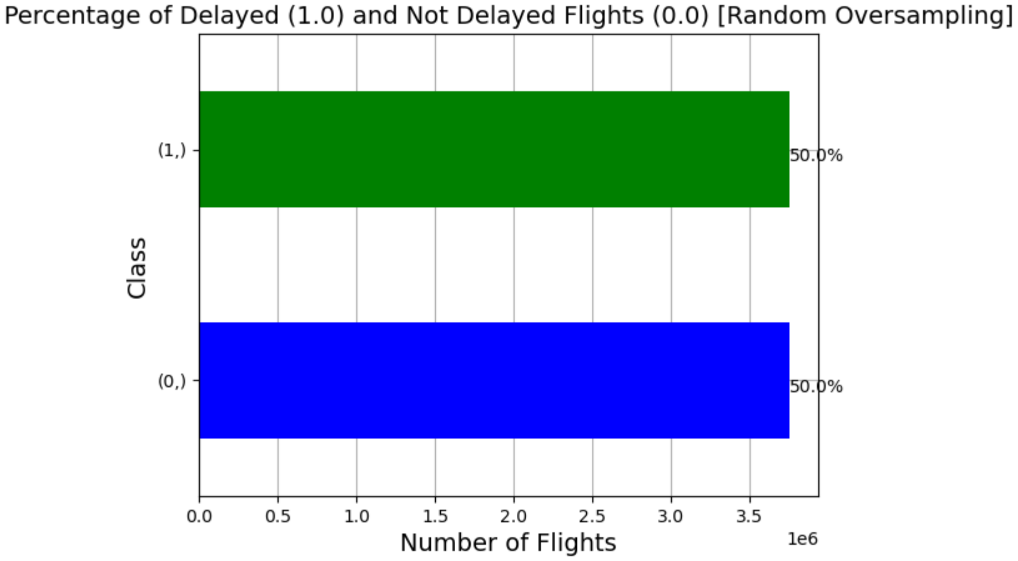


Figure 4.4: Percentage of Delayed and Not Delayed Flights (Random Oversampling)

- **SMOTE (Synthetic Minority Oversampling Technique)**: SMOTE is another Over Sampling technique which creates synthetic data for the minority class, by taking random cases from the minority class and finding its k-nearest neighbours. Afterwards, a point between the point and its nearest k neighbours is selected to generate synthetic data with high correlation to the actual dataset and balance the flight dataset. It synthetically balances the classes of the training set in order to subsequently train the classifier model.

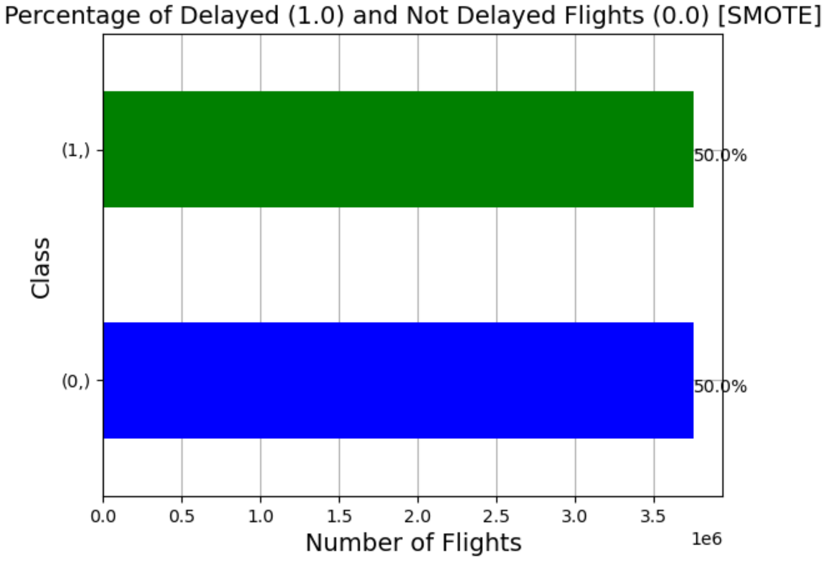


Figure 4.5: Percentage of Delayed and Not Delayed Flights (SMOTE)

When all the performance metrics are compared (Section 4.5), resampling slightly raises Class 1 scores, which was its major objective. As a result, the classification stage can be executed with sampled data from the feature set.

4.4. Classification

Classification process is responsible for classifying the flights as delayed or on-time. The first step to take is the environment setup, which consists of importing the libraries and loading the dataset. Through data analysis, the meaning and the strength of the dataset features are internalized. The features are selected keeping the most relevant features for prediction. During preprocessing, the categorical variables are encoded and the dataset is splitted between training and testing sets in a 80:20 ratio. Train/Test Split is a technique that splits the dataset into training and testing sets, which are mutually exclusive. The testing dataset is not part of the dataset that is used to train the model, which will provide a more accurate evaluation on out-of-sample accuracy. It provides a better understanding of how well the model adapts to new data. This tool enables out-of-sample testing, because each data point in the testing dataset has not been used to train the model, but the outcome is known, which is positive for testing.

Different classifier models are trained (classifier fitted with the training data) and tested, and their perfomance is evaluated through various performance metrics. Classification consists of predicting a sample's class label based on its input features. In this case, classification is the first stage of the two-stage model and its function is to classify scheduled flights as delayed or not. Flights are categorised as delayed when the arrival delay is greater than 15 minutes. Delayed flights have the target variable, FLIGHT_STATUS, equal to 1 and not delayed flights (on time flights) have a null value of the target variable (FLIGHT_STATUS).

The classification models applied are the following:

- **Logistic Regression**: The basic concept of logistic regression is the employment of a logistic function to describe a binary dependent variable. It is used for binary classification problems (two class values), and it is based on the logistic or sigmoid function. In regression analysis, logistic regression (also known as logit regression) estimates logistic model's parameters (a form of binary regression).

- **Decision Tree Classifier**: A non-parametric supervised learning technique for classification and regression is called a Decision Tree (DT). The objective is to learn straightforward decision rules derived from the flight data features in order to build a model that predicts the value of a target variable. By constructing a decision tree, the Decision Tree classifier develops a model for classification. A test on a feature is represented by each node in the tree, and each branch coming from that node represents one of the possible values for that feature.

- **Gradient Boosting Classifier**: Gradient Boosting is a Machine Learning technique that creates a prediction model in the form of different Decision Trees. Also called Gradient Boosted Decision Trees (GBDT), it is a procedure that can be used both for regression and classification problems. Gradient Boosting classifier accepts both binary and multi-class classification.

- **XGBoost Classifier**: XGBoost (Extreme Gradient Boosting) is an optimized Machine Learning algorithm (Gradient Boosted Decision Trees (GBDT) Machine Learning library) based on multiple decision trees that employs a gradient boosting structure (parallel tree boosting). It is designed to upgrade single weak models by merging them with multiple models, to build a jointly strong model. The definitive prediction speaks for the weighted sum of the different tree predictions. It is designed to be scalable, accurate, efficient and malleable, being applied for classification, regression and ranking problems.

- **Random Forest Classifier**: Random Forests or Random Decision Forests is a learning technique for classification, regression, and other problems, which builds a large number of full decision trees in parallel during the training phase (random data set samples) and outputs the class that represents the mean prediction of all the separate trees in order to improve the accuracy and avoid overfitting. Random Forest applies the previously explained technique called bagging to ensure the differences between decision trees. Random Forests, by the combination of different trees, obtains a reduced variance with the disadvantage of a minor increase in bias.

- **Extra Trees Classifier**: Extra Trees Classifier, also known as Extremely Randomized Trees Classifier is a learning method that combines the output of numerous non-related decision trees gathered in a "forest" to produce its classification output. Similar to Random Forests, Extra Trees classifier uses the entire dataset and selects, in a random way, the feature values to split and generate nodes. It achieves a greater reduction of variance than Random Forests, with also a slightly bigger increase in bias. Taking into account computational cost, Extra Trees is a much more efficient classifier than Random Forests.

- **SVMs (Support Vector Machines) Classifier**: Supervised learning procedure used for classification, regression and outlier spotting. It is a Machine Learning algorithm that detects an N-dimensional space hyperplane that accurately classifies the data points. This selected hyperplane maximizes the distance between the data points pertaining to the two different classes. *LinearSVC* utilises a linear kernel and stands out for a quick implementation.

The performance of the different classification models is analysed and compared making use of different performance metrics (Section 4.5). The classifier results are picked up in the correspondent confusion matrix with the following nomenclature:

1. **TP (True Positives)**: Delayed flights correctly classified as delayed.

2. **FP (False Positives)**: On Time flights incorrectly classified as delayed.

3. **TN (True Negatives)**: Not Delayed flights correctly classified as not delayed (on-time).

4. **FN (False Negatives)**: Delayed flights incorrectly classified as not delayed (on-time).

A great performance of the model ensures that the number of False Positives (FP) and False Negatives (FN) is very low. Precision and Recall metrics obtain the costs of False Positives (FP) and False Negatives (FN), respectively.

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. In a confusion matrix, each column of the matrix represents the predicted class data, while each row states for the data in an actual class (dataset value).

	Prediction Result	
Dataset Value	True Positive (TP)	False Negative (FN)
	False Positive (FN)	True Negative (TN)

Table 4.1: Confusion Matrix

4.5. Performance Metrics for Classifier Models

In order to perform an evaluation and comparison of the aforementioned models in the process of the classifier model selection, the following performance evaluation metrics are used:

- **Accuracy**: It is very intuitive as it states for the total number of accurate predictions divided by the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

- **Precision**: It states for the trustability of the result, when the model predicts that a flight belongs to one class or another class. It is a measure of the correct positive predictions compared to the total number of positive predictions, quantifying the relevant classified items.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

- **Recall**: Defines the model's accuracy to predict a certain class. It is a measure of the correct positive predictions compared to the total number of positive results in the dataset.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

- **F1 Score**: Harmonic mean of Precision and Recall (combined metric). It is a measure that combines both Precision and Recall.

$$F1Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4.4}$$

- **Balanced Accuracy**: Modified accuracy utilised for imbalanced data. Balanced Accuracy sets a higher weight to the minority classes and a lower weight to the majority classes. Therefore, the accuracy represents each class in the dataset. It can also be defined as the arithmetic mean of Sensitivity and Specificity. Sensitivity states for the Recall for Class 1 (True Positive Rate) and Specificity for the Recall for Class 0 (True Negative Rate).

Evaluation metrics indicate the model areas that require improvements. The classifier's goal is to accurately predict delayed flights, maximising Recall results. However, it is evident that Balanced Accuracy is the arithmetic mean of Recall for classes 0 and 1. Consequently, it can be regarded as the classification's main metric.

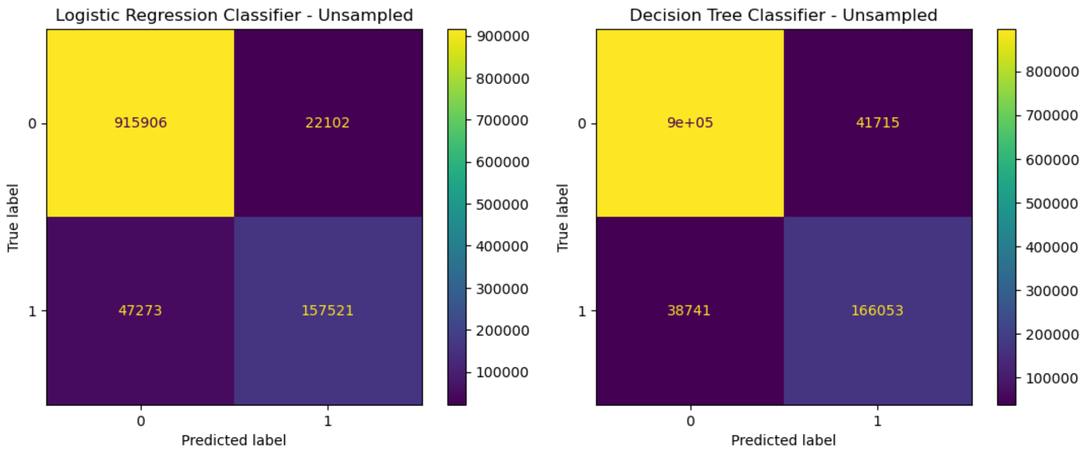
Overall, it can be deduced that Class 1 predictions scored significantly worse results than Class 0 predictions. The scores for Accuracy and Balanced Accuracy for each model for the unsampled dataset differ significantly. This suggests that resampling is necessary to correct the dataset's imbalance.

	High Recall	**Low Recall**
High Precision	Perfectly detected Class by the model	Model doesn't easily detect a Class, but it is effective when it predicts a Class
Low Precision	Class is well detected, but other Classes points are present in the model	Imprecisely detected Class by the model

Table 4.2: Performance Metrics for Classifiers

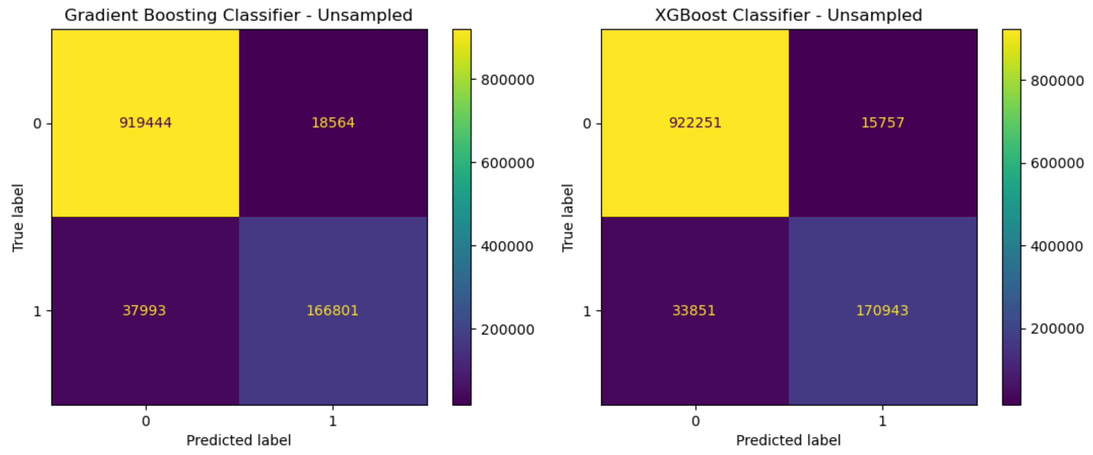
4.5.1. Classifier Models - Unsampled Data

The following graphs (Figures 4.6 to 4.9) collect the Classifier Model's Confusion Matrices for unsampled data (imbalanced dataset):



(a) Logistic Regression Confusion Matrix (b) Decision Tree Confusion Matrix

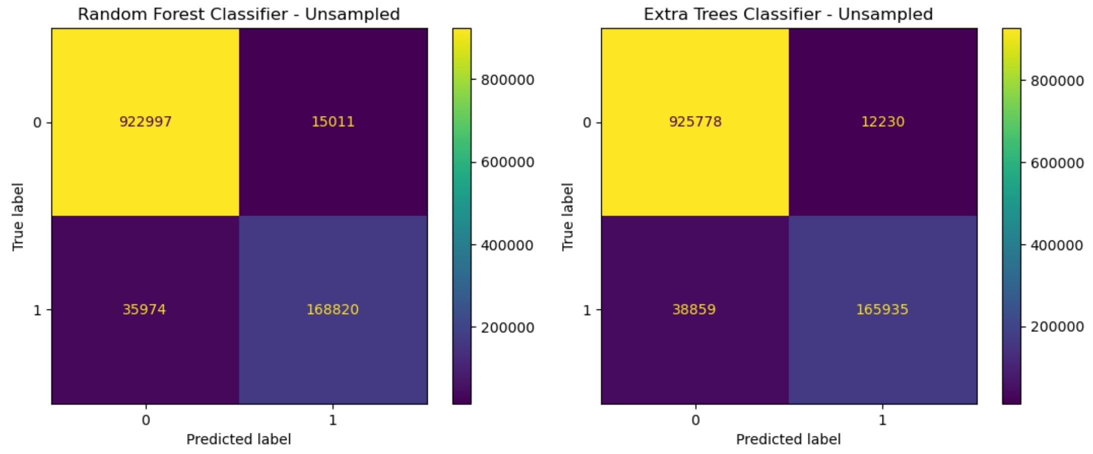
Figure 4.6: Logistic Regression and Decision Tree Confusion Matrix - Unsampled Data



(a) Gradient Boosting Confusion Matrix

(b) XGBoost Confusion Matrix

Figure 4.7: Gradient Boosting and XGBoost Confusion Matrix - Unsampled Data



(a) Random Forest Confusion Matrix

(b) Extra Trees Confusion Matrix

Figure 4.8: Random Forest and Extra Trees Confusion Matrix - Unsampled Data

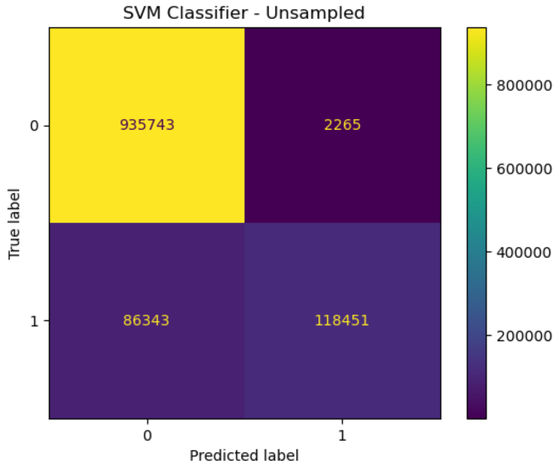


Figure 4.9: SVM (Support Vector Machine) Confusion Matrix - Unsampled Data

The performance metrics for all the Classification models (considering unsampled data) are contained in the following tables. Table 4.3 comprises the different evaluation metrics (precision, recall, F1-score and accuracy) for both of the dataset classes (delayed (1) and not delayed (0) flights). Furthermore, the performance metrics for the test set (balanced accuracy and F1-score) of all the applied classifiers are included in Table 4.4.

Classification Model	**Performance Metrics – Unsampled Data**						
	Precision		**Recall**		**F1-Score**		**Accuracy**
	0	1	0	1	0	1	
Logistic Regression	0.95	0.88	0.98	0.77	0.96	0.82	0.94
Decision Tree Classifier	0.96	0.80	0.96	0.81	0.96	0.80	0.93
Gradient Boosting Classifier	0.96	0.90	0.98	0.81	0.97	0.86	0.95
XGBoost Classifier	0.96	0.92	0.98	0.83	0.97	0.87	0.96
Random Forest Classifier	0.96	0.92	0.98	0.82	0.97	0.87	0.96
Extra Trees Classifier	0.96	0.93	0.99	0.81	0.97	0.87	0.96
SVC (Support Vector Classifier)	0.92	0.98	1.00	0.58	0.95	0.73	0.92

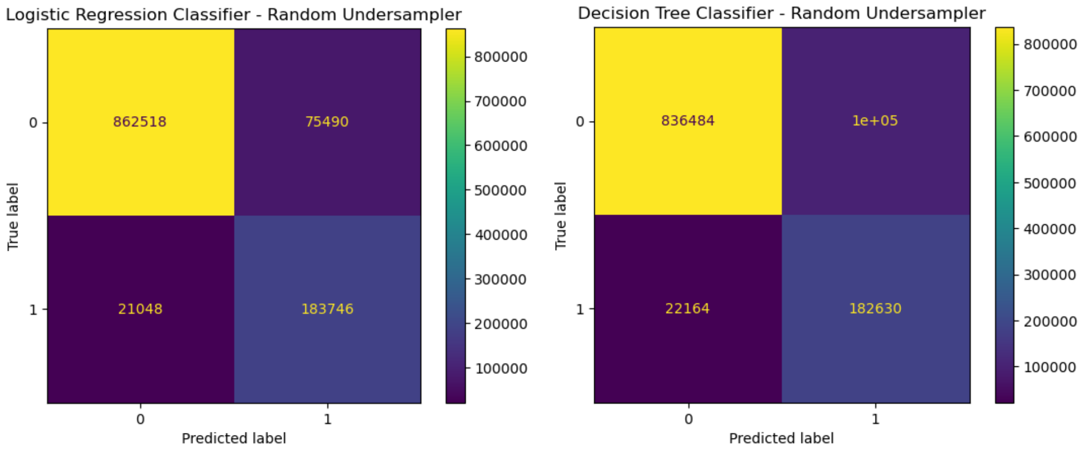
Table 4.3: Classification Models Performance Metrics – Unsampled Data

Classification Model (Unsampled Data)	**Performance Metrics (Test Set)**	
	Balanced Accuracy	**F1-Score**
Logistic Regression	0.873	0.820
Decision Tree Classifier	0.883	0.805
Gradient Boosting Classifier	0.897	0.855
XGBoost Classifier	0.909	0.873
Random Forest Classifier	0.904	0.869
Extra Trees Classifier	0.899	0.867
SVC (Support Vector Classifier)	0.788	0.728

Table 4.4: Test Set Performance Metrics – Unsampled Data

4.5.2. Classifier Models - Random UnderSampler

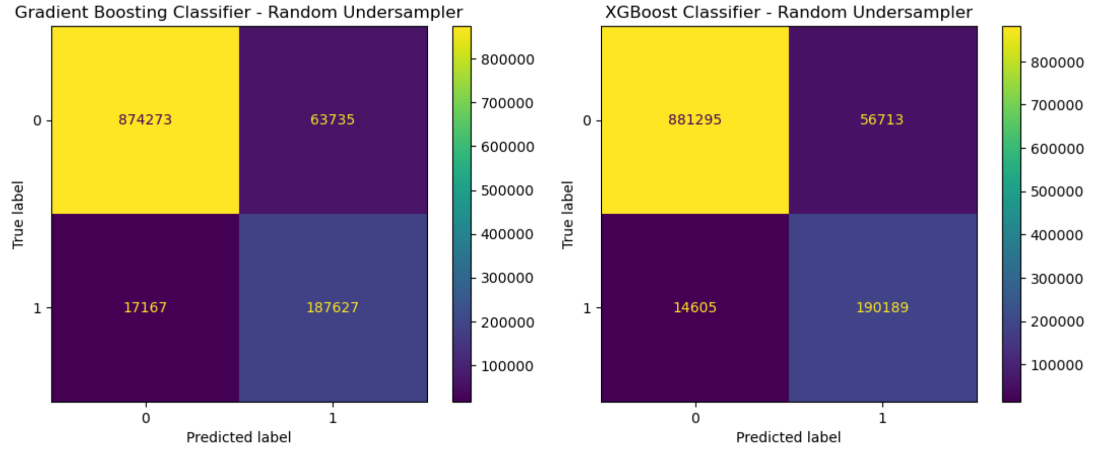
The following graphs (Figures 4.10 to 4.13) collect the Classifier Model's Confusion Matrices for the sampling method, Random Undersampler:



(a) Logistic Regression Confusion Matrix

(b) Decision Tree Confusion Matrix

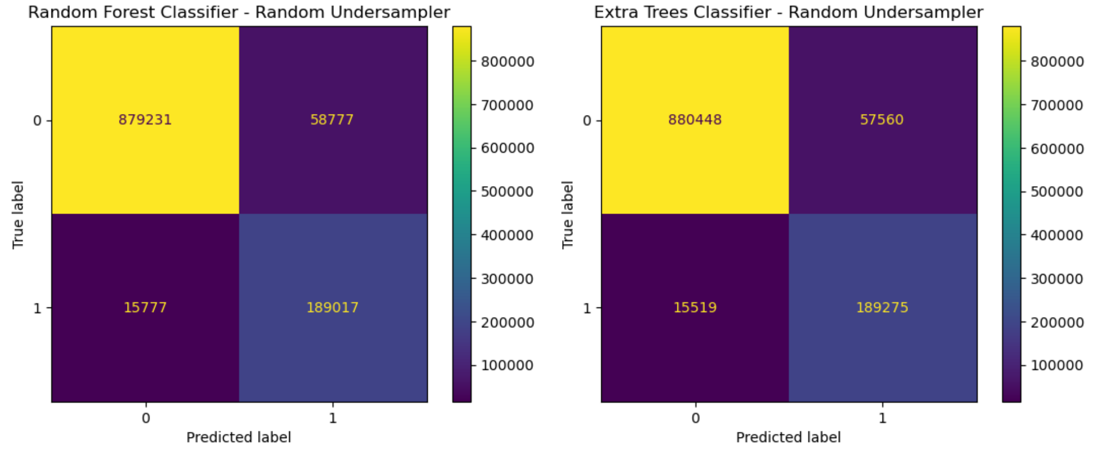
Figure 4.10: Logistic Regression and Decision Tree Confusion Matrix - Random Undersampler



(a) Gradient Boosting Confusion Matrix

(b) XGBoost Confusion Matrix

Figure 4.11: Gradient Boosting and XGBoost Confusion Matrix - Random Undersampler



(a) Random Forest Confusion Matrix

(b) Extra Trees Confusion Matrix

Figure 4.12: Random Forest and Extra Trees Confusion Matrix - Random Undersampler

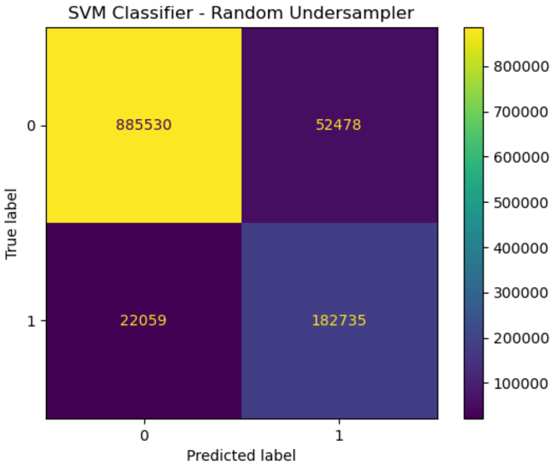


Figure 4.13: SVM (Support Vector Machine) Confusion Matrix - Random Undersampler

The performance metrics for all the Classification models (considering sampling method, Random Undersampler) are contained in the following tables. Table 4.5 comprises the different evaluation metrics (precision, recall, F1-score and accuracy) for both of the dataset classes (delayed (1) and not delayed (0) flights). Furthermore, the performance metrics for the test set (balanced accuracy and F1-score) of all the applied classifiers are included in Table 4.6.

Classification Model	**Performance Metrics – Random Undersampler**						
	Precision		**Recall**		**F1-Score**		**Accuracy**
	0	1	0	1	0	1	
Logistic Regression	0.98	0.71	0.92	0.90	0.95	0.79	0.92
Decision Tree Classifier	0.97	0.64	0.89	0.89	0.93	0.75	0.89
Gradient Boosting Classifier	0.98	0.75	0.93	0.92	0.96	0.82	0.93
XGBoost Classifier	0.98	0.77	0.94	0.93	0.96	0.84	0.94
Random Forest Classifier	0.98	0.76	0.94	0.92	0.96	0.84	0.93
Extra Trees Classifier	0.98	0.77	0.94	0.92	0.96	0.84	0.94
SVC (Support Vector Classifier)	0.98	0.78	0.94	0.89	0.96	0.83	0.93

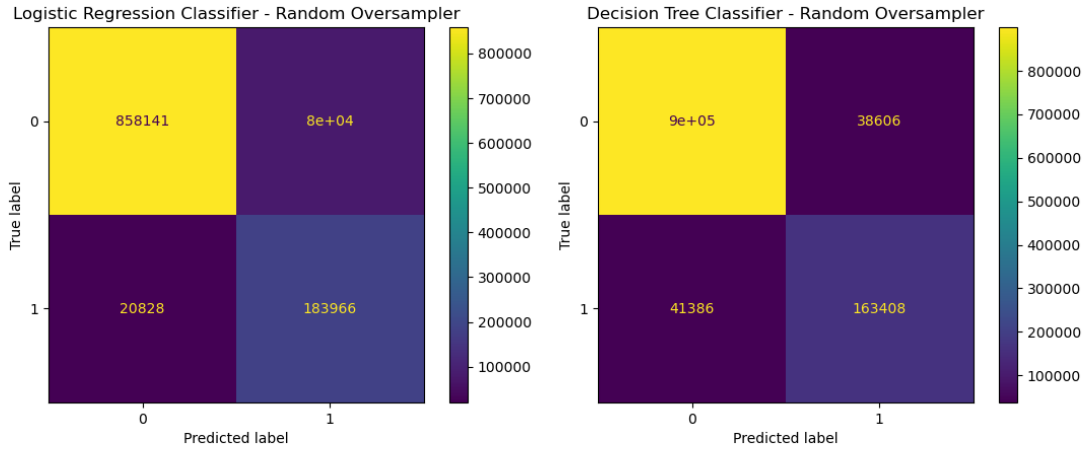
Table 4.5: Classification Models Performance Metrics – Random Undersampler

Classification Model (Random Undersampler)	**Performance Metrics (Test Set)**	
	Balanced Accuracy	**F1-Score**
Logistic Regression	0.908	0.792
Decision Tree Classifier	0.892	0.747
Gradient Boosting Classifier	0.924	0.823
XGBoost Classifier	0.934	0.942
Random Forest Classifier	0.930	0.835
Extra Trees Classifier	0.931	0.838
SVC (Support Vector Classifier)	0.918	0.831

Table 4.6: Test Set Performance Metrics – Random Undersampler

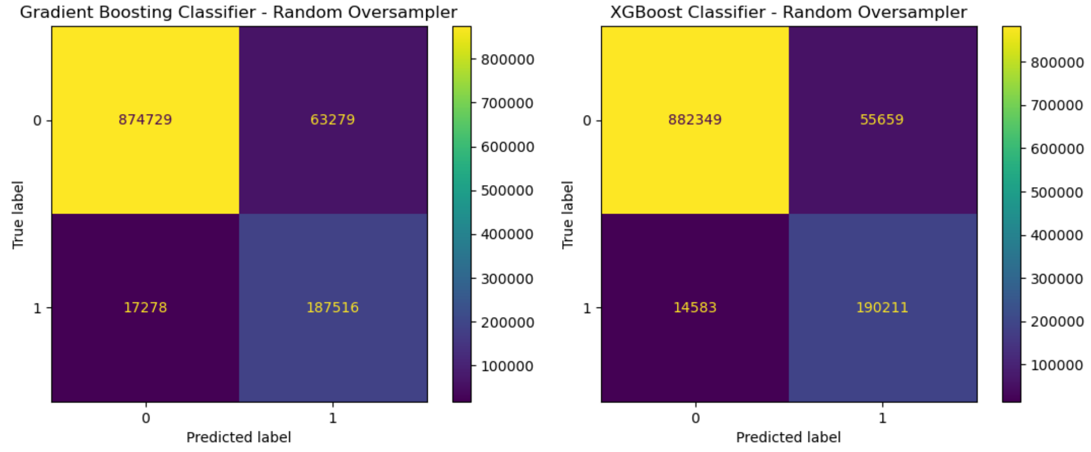
4.5.3. Classifier Models - Random OverSampler

The following graphs (Figures 4.14 to 4.17) collect the Classifier Model's Confusion Matrices for the sampling method, Random Oversampler:



(a) Logistic Regression Confusion Matrix (b) Decision Tree Confusion Matrix

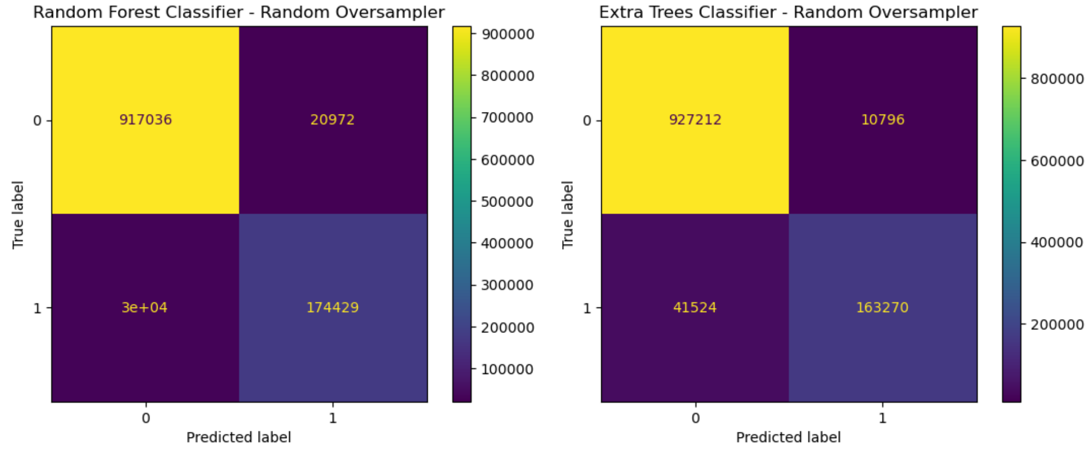
Figure 4.14: Logistic Regression and Decision Tree Confusion Matrix - Random Oversampler



(a) Gradient Boosting Confusion Matrix

(b) XGBoost Confusion Matrix

Figure 4.15: Gradient Boosting and XGBoost Confusion Matrix - Random Oversampler



(a) Random Forest Confusion Matrix

(b) Extra Trees Confusion Matrix

Figure 4.16: Random Forest and Extra Trees Confusion Matrix - Random Oversampler

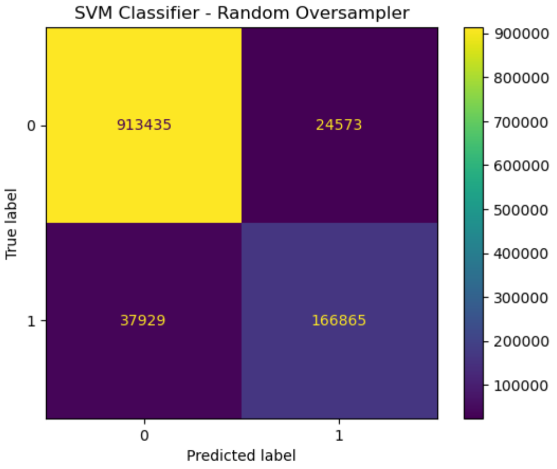


Figure 4.17: SVM (Support Vector Machine) Confusion Matrix - Random Oversampler

The performance metrics for all the Classification models (considering sampling method, Random Oversampler) are contained in the following tables. Table 4.7 comprises the different evaluation metrics (precision, recall, F1-score and accuracy) for both of the dataset classes (delayed (1) and not delayed (0) flights). Furthermore, the performance metrics for the test set (balanced accuracy and F1-score) of all the applied classifiers are included in Table 4.8.

Classification Model	**Performance Metrics – Random Oversampler**						
	Precision		**Recall**		**F1-Score**		**Accuracy**
	0	1	0	1	0	1	
Logistic Regression	0.98	0.70	0.91	0.90	0.94	0.79	0.91
Decision Tree Classifier	0.96	0.81	0.96	0.80	0.96	0.80	0.93
Gradient Boosting Classifier	0.98	0.75	0.93	0.92	0.96	0.82	0.93
XGBoost Classifier	0.98	0.77	0.94	0.93	0.96	0.84	0.94
Random Forest Classifier	0.97	0.89	0.98	0.85	0.97	0.87	0.96
Extra Trees Classifier	0.96	0.94	0.99	0.80	0.97	0.86	0.95
SVC (Support Vector Classifier)	0.96	0.87	0.97	0.81	0.97	0.84	0.95

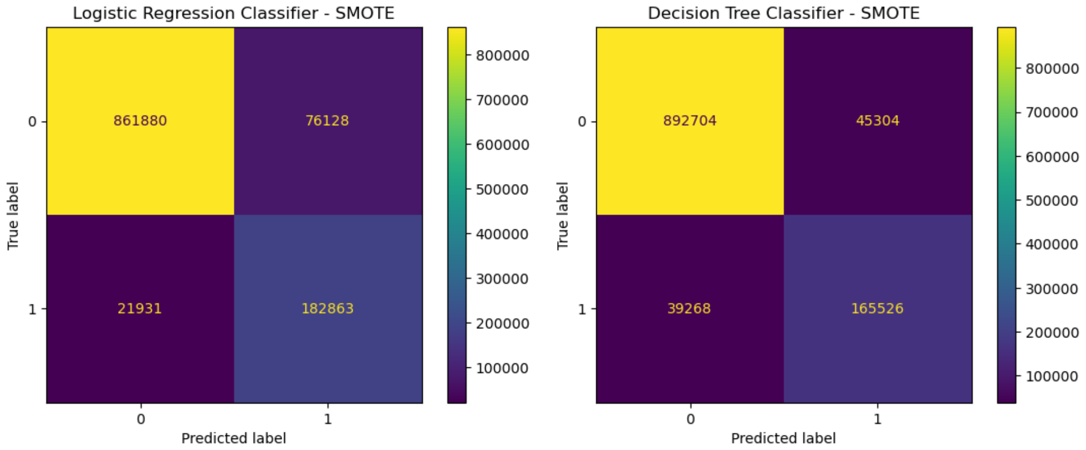
Table 4.7: Classification Models Performance Metrics – Random Oversampler

Classification Model (Random Oversampler)	**Performance Metrics (Test Set)**	
	Balanced Accuracy	**F1-Score**
Logistic Regression	0.907	0.785
Decision Tree Classifier	0.878	0.803
Gradient Boosting Classifier	0.924	0.823
XGBoost Classifier	0.935	0.844
Random Forest Classifier	0.915	0.872
Extra Trees Classifier	0.893	0.861
SVC (Support Vector Classifier)	0.894	0.842

Table 4.8: Test Set Performance Metrics – Random Oversampler

4.5.4. Classifier Models - SMOTE

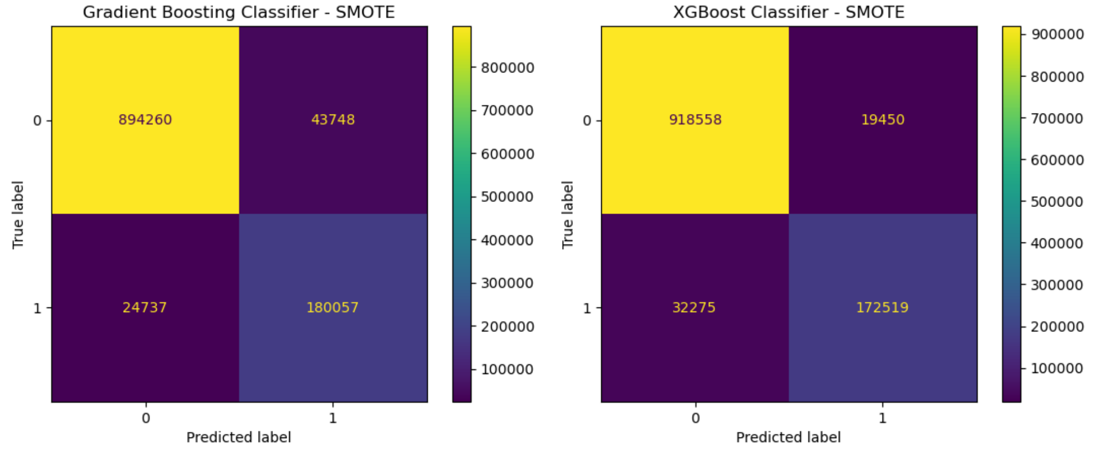
The following graphs (Figures 4.18 to 4.21) collect the Classifier Model's Confusion Matrices for the sampling method, SMOTE:



(a) Logistic Regression Confusion Matrix

(b) Decision Tree Confusion Matrix

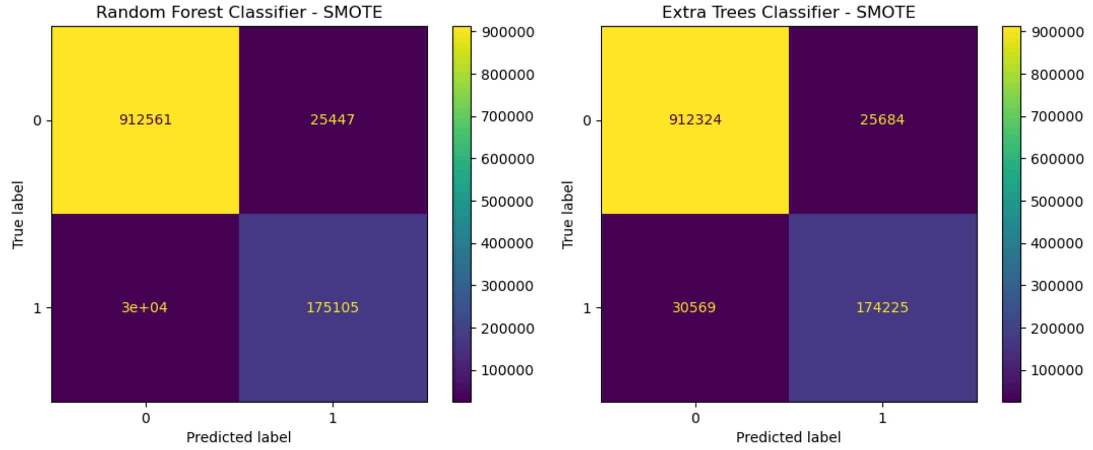
Figure 4.18: Logistic Regression and Decision Tree Confusion Matrix - SMOTE



(a) Gradient Boosting Confusion Matrix

(b) XGBoost Confusion Matrix

Figure 4.19: Gradient Boosting and XGBoost Confusion Matrix - SMOTE



(a) Random Forest Confusion Matrix

(b) Extra Trees Confusion Matrix

Figure 4.20: Random Forest and Extra Trees Confusion Matrix - SMOTE

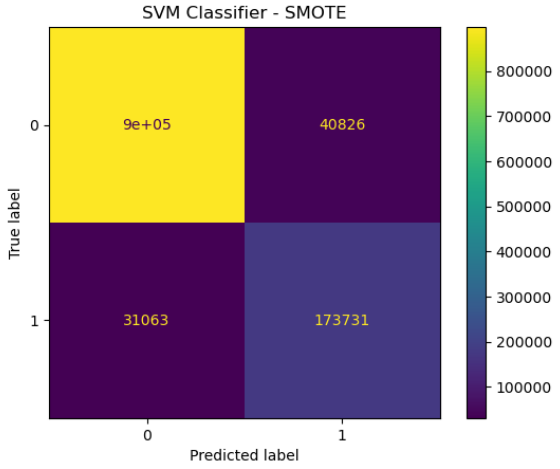


Figure 4.21: SVM (Support Vector Machine) Confusion Matrix - SMOTE

The performance metrics for all the Classification models (considering sampling method, SMOTE) are contained in the following tables. Table 4.9 comprises the different evaluation metrics (precision, recall, F1-score and accuracy) for both of the dataset classes (delayed (1) and not delayed (0) flights). Furthermore, the performance metrics for the test set (balanced accuracy and F1-score) of all the applied classifiers are included in Table 4.10.

Classification Model	**Performance Metrics – SMOTE**						
	Precision		**Recall**		**F1-Score**		**Accuracy**
	0	1	0	1	0	1	
Logistic Regression	0.98	0.71	0.92	0.89	0.95	0.79	0.91
Decision Tree Classifier	0.96	0.79	0.95	0.81	0.95	0.80	0.93
Gradient Boosting Classifier	0.97	0.80	0.95	0.88	0.96	0.84	0.94
XGBoost Classifier	0.97	0.90	0.98	0.84	0.97	0.87	0.95
Random Forest Classifier	0.97	0.87	0.97	0.86	0.97	0.86	0.95
Extra Trees Classifier	0.97	0.87	0.97	0.85	0.97	0.86	0.95
SVC (Support Vector Classifier)	0.97	0.81	0.96	0.85	0.96	0.83	0.94

Table 4.9: Classification Models Performance Metrics – SMOTE

Classification Model (SMOTE)	**Performance Metrics (Test Set)**	
	Balanced Accuracy	**F1-Score**
Logistic Regression	0.906	0.789
Decision Tree Classifier	0.880	0.797
Gradient Boosting Classifier	0.916	0.840
XGBoost Classifier	0.911	0.870
Random Forest Classifier	0.914	0.864
Extra Trees Classifier	0.912	0.861
SVC (Support Vector Classifier)	0.902	0.829

Table 4.10: Test Set Performance Metrics – SMOTE

4.5.5. Classifier Models Conclusions

Techniques like Random Undersampler, Random Oversampler or SMOTE (Synthetic Minority Oversampling Technique) generally get better results in terms of Precision, Recall and F1-Score for Delayed Flights Class (Class 1) than unsampled data, and, therefore, they moderately improve the performance of the different classification models.

First of all, sampling methods pretend to increase different performance evaluation metrics values (Precision, Recall and F1-Score) for the minority class. However, the results show that these metrics values for the Delayed Flights Class are very similar for unsampled data and when different sampling methods are applied, but slightly higher for Random Under and Over Sampling.

It can be explained that classifiers like Random Forest Classifier, Gradient Boosting Classifier and XGBoost Classifier do not show important improvements with resampling, as they are programmed to labour on imbalanced classes. F1-Score considers both Precision and Recall and hence, the classifier selection would be based on the highest F1-Score.

Under this hypothesis, XGBoost, Random Forest and Extra Trees Classifier present the greater F1-Score for both classes, when Random Undersampler technique is applicated. Also, XGBoost and Random Forest Classifiers obtain the highest F1-Score, considering both classes, for Random Oversampler method. Furthermore, the Balanced Accuracy metric and F1-Score (for the test set) for XGBoost Classifier show the highest results compared to other classifiers for Random Undersampler technique (0.934 and 0.942, respectively). The greatest outcomes, when considering Random Oversampler method, for the test set are achieved by XGBoost Classifier (Balanced Accuracy equal to 0.935) and Random Forest Classifier (F1-Score equal to 0.872).

4.6. ROC (Receiver Operator Characteristic) of the Classifiers

A measurement tool for binary classification issues is the Receiver Operator Characteristic (ROC) curve. It is a probability curve that represents the TPR (True Positive Rate) against the FPR (False Positive Rate) at different threshold values.

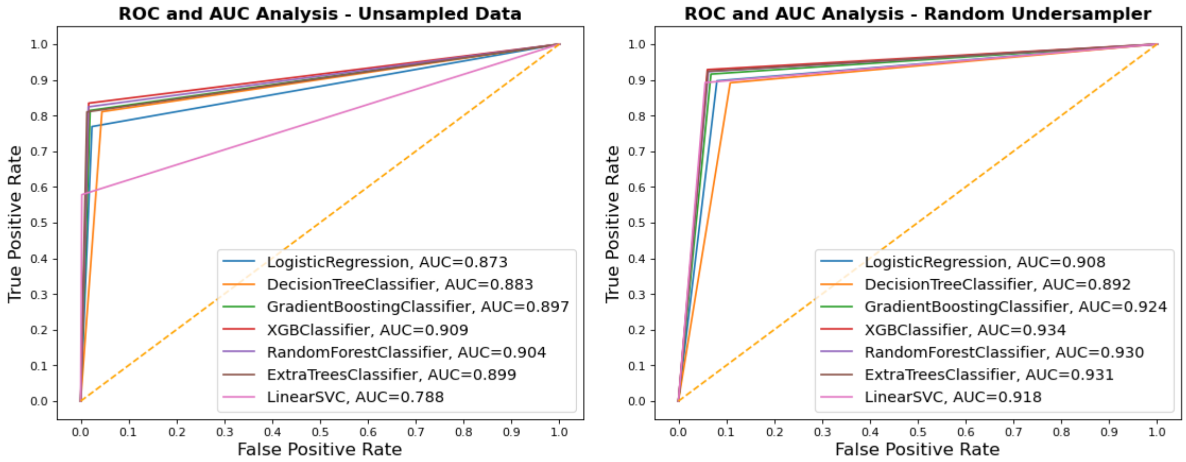
AUC (Area Under the Curve), in this case under the ROC curve, is a parameter that indicates the performance of the classifier model that has been trained. AUC (Area Under the Curve) acts as a conclusion of the ROC curve. The higher the AUC value, the better the performance of the classifier model, being able to distinguish between positive and negative classes:

- **AUC equal to 0.5**: The model cannot discern correctly between classes (represented by the discontinuous yellow line in the following graphs (Figures 4.22 and 4.23)). The classifier can predict random class or constant class for all the flights data.

- **AUC equal to 0.0**: The model's predictions are exactly the opposites of the real classes (all positive class points predicted as negatives, and all negative class points predicted as positives).

- **AUC between 0.0 and 1.0**: Classifier has the possibility to distinguish the classes (the number of True Positives (TP) and True Negatives (TN) are superior than the values of False Positives (FP) and False Negatives (FN)).

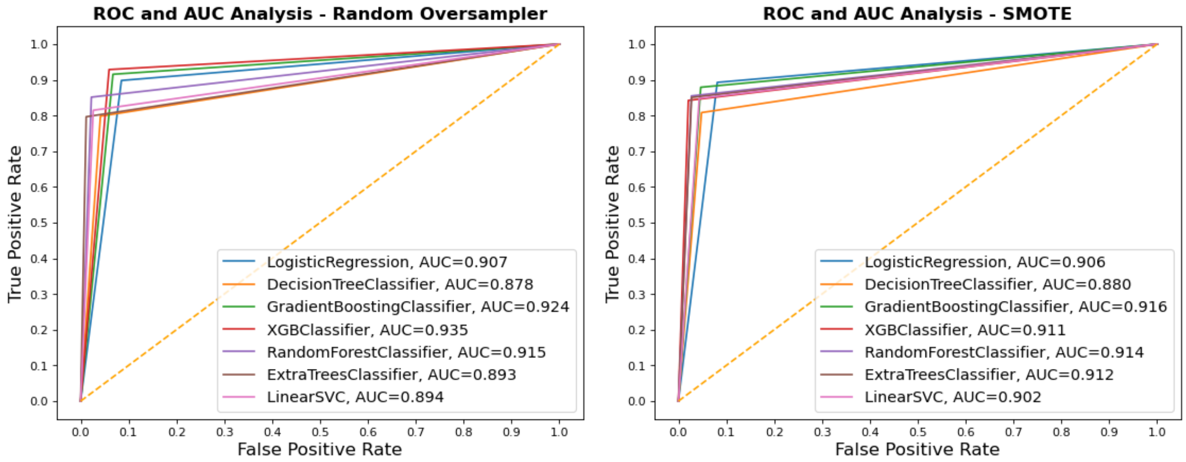
- **AUC equal to 1.0**: Accurate predictions of classes performed by the model. Positive and negative class points are precisely distinguished.



(a) Unsampled Data

(b) Random Undersampler

Figure 4.22: ROC and AUC Analysis - Unsampled Data and Random Undersampler



(a) Random Oversampler

(b) SMOTE

Figure 4.23: ROC and AUC Analysis - Random Oversampler and SMOTE

When applying Random Oversampler, the highest area under the ROC from all of the classifiers is achieved. XGBoost Classifier obtained the highest result for AUC (Area Under the Curve) for the ROC (Receiver Operator Characteristic) (0.909 for Unsampled Data, 0.934 for Random Undersampler and 0.935 for Random Oversampler). Random Forest Classifier also stands out with an AUC for the ROC of 0.915 for Random Oversampler and 0.930 for Random Undersampler.

Except for Gradient Boosting Classifier and XGBoost Classifier (greater AUC for Random Oversampler), all the remaining classifier models obtain their highest score for AUC (Area Under the Curve) from the ROC (Receiver Operator Characteristic) curve for Random Undersampler method, when comparing the results with those obtained for unsampled data and the other sampling methods. However, as it can be observed on the previous sections (Sections 4.5.1 to 4.5.4), the performance evaluation metrics for all the classifiers have a slightly lower value for Random Undersampler technique than for Random Oversampler method.

4.7. Regression

Regression is the method of predicting a continuous value taking into account input features. Regression analysis is a statistical process to acknowledge the relationship between a dependent variable and one or more independent variables. In this case, the regression models are used to predict the arrival delay in minutes, which is the target variable. Regression is the second stage of the two-stage model. By this process, the Arrival Delay in minutes is predicted if the flight was classified as Delayed by the dataset (FLIGHT_STATUS equal to 1). Flights that are delayed by 15 minutes or more are applied for training the regression models. The analysed regression models are: Linear Regressor, Decision Tree Regressor, Gradient Boosting Regressor, XGBoost Regressor, Random Forest Regressor and Extra Trees Regressor.

R-Squared is an excellent performance metric to measure the capacity of a regression model to predict the variations in the dataset. The higher the R^2 Score, the better the capability of a regressor. The regressor's main goal is to predict the period of flight delay with the minimum error. By consequence, the regressor model that achieves the minimum value of MSE, RMSE and MAE is seaked. In conclusion, RMSE and R-Squared are the most important metrics to make the comparison between the different regressors.

4.7.1. Regression Models - Unsampled Data

The performance evaluation metrics (Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Average Error (MAE) and R^2 Score) for all the Regression models (considering unsampled data) are contained in the following table:

Regression Model - Unsampled Data	MSE	RMSE	MAE	R2 Score
Linear Regression	136.48	11.68	8.53	0.9675
Decision Tree Regressor	189.34	13.76	9.71	0.9549
Gradient Boosting Regressor	98.54	9.93	7.14	0.9765
XGBoost Regressor	74.69	8.64	6.14	0.9822
Random Forest Regressor	89.39	9.45	6.69	0.9787
Extra Trees Regressor	92.26	9.61	6.82	0.9780

Table 4.11: Regression Models Performance Evaluation Metrics – Unsampled Data

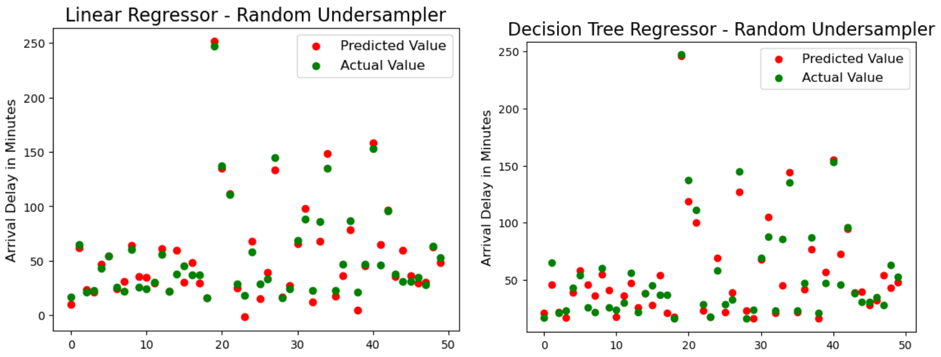
4.7.2. Regression Models - Random UnderSampler

The performance evaluation metrics (Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Average Error (MAE) and R^2 Score) for all the Regression models (considering sampling method, Random Undersampler) are contained in the following table:

Regression Model – Random Undersampler	MSE	RMSE	MAE	R2 Score
Linear Regression	136.48	11.68	8.53	0.9675
Decision Tree Regressor	188.99	13.75	9.70	0.9549
Gradient Boosting Regressor	98.54	9.93	7.14	0.9765
XGBoost Regressor	74.69	8.64	6.14	0.9822
Random Forest Regressor	89.23	9.45	6.68	0.9787
Extra Trees Regressor	92.16	9.60	6.82	0.9780

Table 4.12: Regression Models Performance Evaluation Metrics – Random Undersampler

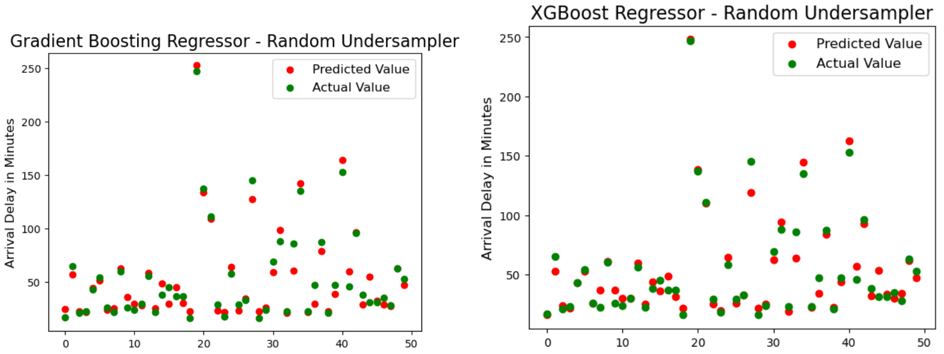
The following graphs (Figures 4.24 to 4.26) contain the predicted value for Arrival Delay in minutes compared with the actual delay value (only for 50 data points (flights)) for the different regression models when sampling method, Random Undersampler is applied.



(a) Linear Regressor

(b) Decision Tree Regressor

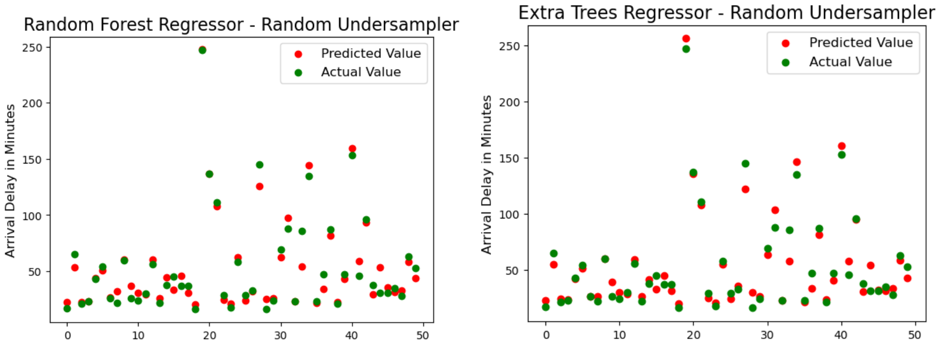
Figure 4.24: Linear Regressor and Decision Tree Regressor - Random Undersampler



(a) Gradient Boosting Regressor

(b) XGBoost Regressor

Figure 4.25: Gradient Boosting Regressor and XGBoost Regressor - Unsampled Data



(a) Random Forest Regressor

(b) Extra Trees Regressor

Figure 4.26: Random Forest Regressor and Extra Trees Regressor - Random Undersampler

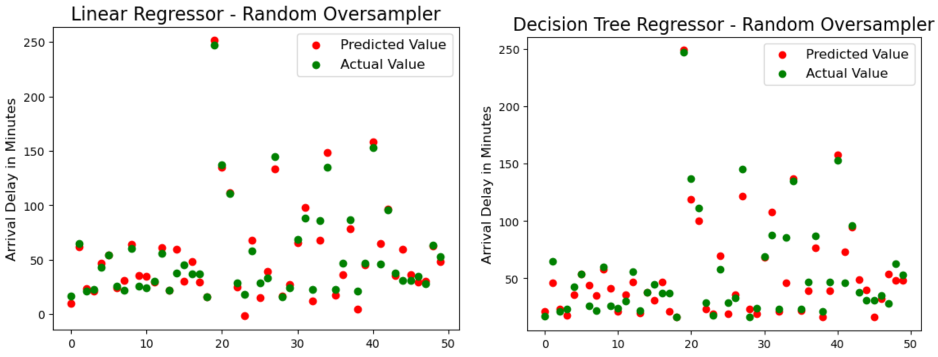
4.7.3. Regression Models - Random OverSampler

The performance evaluation metrics (Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Average Error (MAE) and R^2 Score) for all the Regression models (considering sampling method, Random Oversampler) are contained in the following table:

Regression Model - Random Oversampler	MSE	RMSE	MAE	R2 Score
Linear Regression	136.48	11.68	8.53	0.9675
Decision Tree Regressor	188.40	13.73	9.69	0.9550
Gradient Boosting Regressor	98.54	9.93	7.14	0.9765
XGBoost Regressor	74.69	8.64	6.14	0.9822
Random Forest Regressor	89.36	9.45	6.69	0.9787
Extra Trees Regressor	92.25	9.60	6.83	0.9780

Table 4.13: Regression Models Performance Evaluation Metrics – Random Oversampler

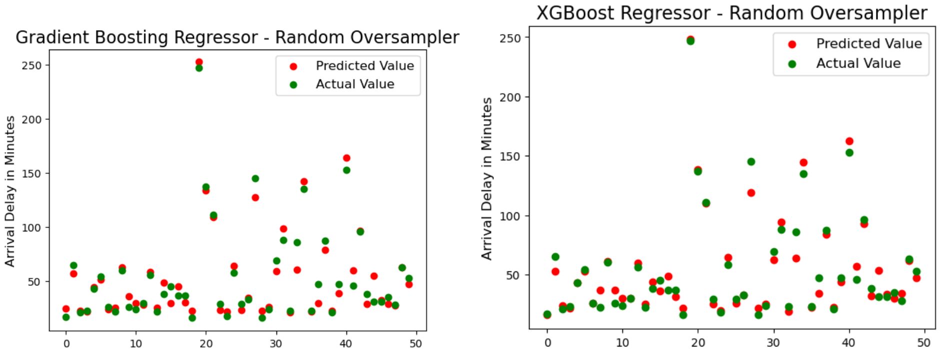
The following graphs (Figures 4.27 to 4.29) contain the predicted value for Arrival Delay in minutes compared with the actual delay value (only for 50 data points (flights)) for the different regression models when sampling method, Random Oversampler is applied.



(a) Linear Regressor

(b) Decision Tree Regressor

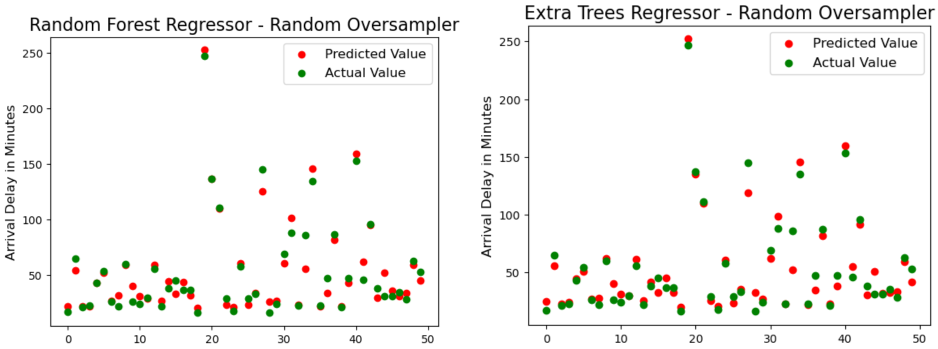
Figure 4.27: Linear Regressor and Decision Tree Regressor - Random Oversampler



(a) Gradient Boosting Regressor

(b) XGBoost Regressor

Figure 4.28: Gradient Boosting Regressor and XGBoost Regressor - Unsampled Data



(a) Random Forest Regressor

(b) Extra Trees Regressor

Figure 4.29: Random Forest Regressor and Extra Trees Regressor - Random Oversampler

4.7.4. Regression Models - SMOTE

The performance evaluation metrics (Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Average Error (MAE) and R^2 Score) for all the Regression models (considering sampling method, SMOTE (Synthetic Minority Oversampling Technique)) are contained in the following table:

Regression Model – SMOTE	MSE	RMSE	MAE	R2 Score
Linear Regression	136.48	11.68	8.53	0.9675
Decision Tree Regressor	188.29	13.72	9.69	0.9551
Gradient Boosting Regressor	98.54	9.93	7.14	0.9765
XGBoost Regressor	74.69	8.64	6.14	0.9822
Random Forest Regressor	89.29	9.45	6.69	0.9787
Extra Trees Regressor	92.01	9.59	6.82	0.9781

Table 4.14: Regression Models Performance Evaluation Metrics – SMOTE

4.7.5. Regressor Models Conclusions

The higher R-Squared scores and the lowest MSE (Mean Squared Error), RMSE (Root Mean Squared Error) and MAE (Mean Average Error) values are achieved by the XGBoost Regressor, when compared with the other regressors results (Linear Regression, Decision Tree, Gradient Boosting, Random Forest and Extra Trees). It can also be concluded that unsampled data obtained higher error results for the different regression models than other sampling methods (Random Undersampler, Random Oversampler and SMOTE (Synthetic Minority Oversampling Technique)). However, the obtained R^2 Scores for all the regressor models are equal when considering sampled or unsampled data.

4.8. Pipelined Model

A pipelined model of two sequential functions is constructed for flight delay prediction. These tasks include the prediction whether a flight will be delayed or not (Classification) and, considering only delayed flights, the prediction of arrival delay in minutes (Regression). To sum up, classifier outputs are fed as an input to the regressor.

Before constructing the pipelined engine, different classification and regression models are analysed and tested to evaluate their performance. To predict the arrival delay of flights accurately, a two-stage Machine Learning engine is programmed. It consists of the pipelined operation of two sequential operations: Classification (predicting whether a flight will be delayed or not) and Regression (considering delayed flights forecasted by the classifier, predicting the arrival delay in minutes). The flights classified as delayed are fed to the trained regressor, which allows to predict the Arrival Delay in minutes. Therefore, classifier results are used for Regression (Pipelining). The two-stage Machine Learning model that prognosticates the commercial flights performance is represented by the following flow chart:

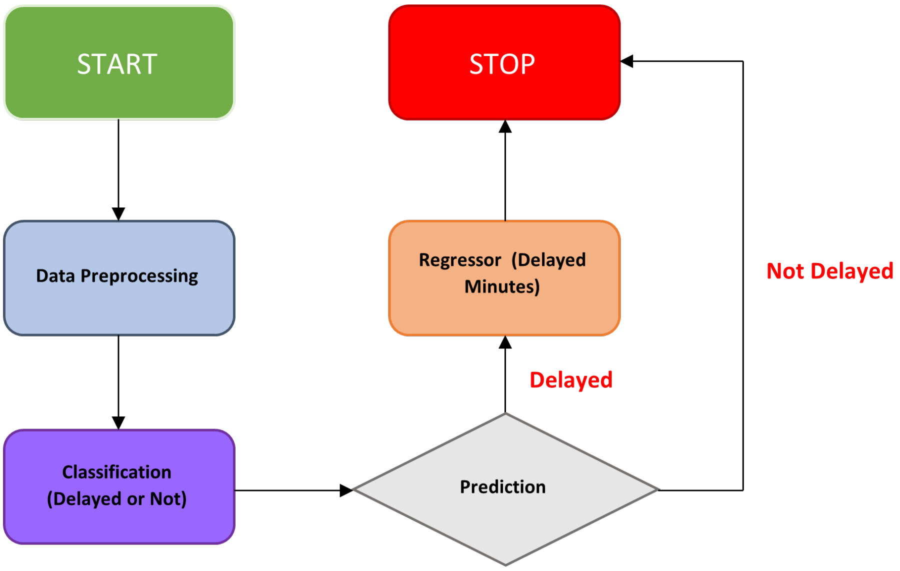


Figure 4.30: Flow Chart of the Pipelined Process

The two-stage flight delay prediction Machine Learning engine is represented by the previous flow chart. Previously to developing the pipelined model, the best classifier and the best regressor can be selected amongst the different analysed models. The data from the flights dataset was pre-processed, and a model was trained using the classifier to accomplish classification. Given that it has the highest F1 Score and a great area under the ROC, the Random Forest Classifier can be selected. Only flights that will be delayed as it was predicted by the classifier model, need to have the flight delay estimated. Delayed flights predicted by the classifier are chosen to do regression and forecast the airplane's Arrival Delay in minutes. Given that it achieved the highest R^2 score and the lowest RMSE and MAE values, the XGBoost Regressor is selected. Resampling moderately improves Class 1 scores, which was its main goal, when all performance metrics are compared. As a result, classification can be performed applying Random Undersampler and Random Oversampler methods.

In terms of Precision, Recall and F1-Score for Delayed Flights Class (Class 1), techniques like Random Undersampler or Random Oversampler slightly outperform Unsampled Data, and as a result, they enhance the performance of the various classification models. Nevertheless, the maximum area under the ROC (0.935) for the XGBoost Classifier is attained, when Random Oversampler is used. Random Undersampler practically obtains the same AUC result as for Random Oversampler (AUC equal to 0.934).

Different pipelined models are tested considering Random Undersampler and Random Oversampler techniques before classification. After classification performed by the classifier, XGBoost Regressor is analysed as it obtained the better RMSE, MAE and R-Squared results from all the considered regression models. Unsampled data isn't considered optimal with respect to the different sampling techniques because the results from the minority class (Class 1) of the evaluation metrics (Precision, Recall and F1-Score) are inferior than the obtained by these techniques, for the different classifiers. With respect to the regression process, XGBoost Regressor achieves the lowest RMSE and MAE values and the highest R-Squared score.

The following tables (Tables 4.15 and 4.16) evaluate the different pipelined models that have been analysed. Table 4.15 picks up the performance evaluation metrics for XGboost, Random Forest and Extra Trees Classifiers, considering Random Undersampler and XGboost and Random Forest Classifiers, considering Random Oversampler.

	XGBoost Regressor							
Evaluation Metrics	**Precision**		**Recall**		**F1-Score**		**Accuracy**	**AUC**
	0	**1**	**0**	**1**	**0**	**1**		
XGBoost Classifier (Random Undersampler)	0.98	0.77	0.94	0.93	0.96	0.84	0.94	0.934
Random Forest Classifier (Random Undersampler)	0.98	0.76	0.94	0.92	0.96	0.84	0.93	0.930
Extra Trees Classifier (Random Undersampler)	0.98	0.77	0.94	0.92	0.96	0.84	0.94	0.932
XGBoost Classifier (Random Oversampler)	0.98	0.77	0.94	0.93	0.96	0.84	0.94	0.935
Random Forest Classifier (Random Oversampler)	0.97	0.89	0.98	0.85	0.97	0.87	0.96	0.915

Table 4.15: Pipelined Models - Evaluation Metrics (Classifiers)

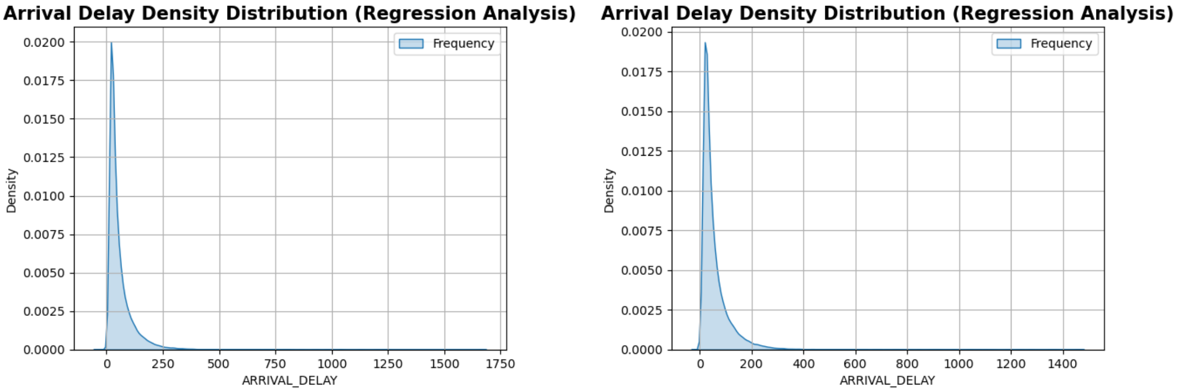
Table 4.16 collects the performance evaluation metrics for the XGBoost Regressor, considering Random Undersampler and Random Oversampler techniques applied on the previous Classifier models.

	XGBoost Regressor			
Evaluation Metrics	**MSE**	**RMSE**	**MAE**	**R2 Score**
Random Undersampler (XGBoost Classifier)	91.55	9.57	6.91	0.9765
Random Undersampler (Random Forest Classifier)	90.70	9.52	6.88	0.9776
Random Undersampler (Extra Trees Classifier)	90.12	9.49	6.85	0.9776
Random Oversampler (XGBoost Classifier)	90.98	9.54	6.93	0.9772
Random Oversampler (Random Forest Classifier)	91.07	9.54	6.83	0.9793

Table 4.16: Pipelined Models - Evaluation Metrics (XGBoost Regressor)

From the results of all the previous analysis, it can be concluded that the best evaluation metrics results (highest Precision, Recall and F1-Score) are achieved for Random Forest Classifier, Random Oversampler and XGBoost Regressor (highest R^2 score and lowest RMSE and MAE values) model.

To commence the regression analysis, the Arrival Delay density distribution for Random Under and Over Sampling shows that the distribution is highly skewed and the vast majority of flights arrival delays have a low value in time, concentrated between 0 and 100 minutes.

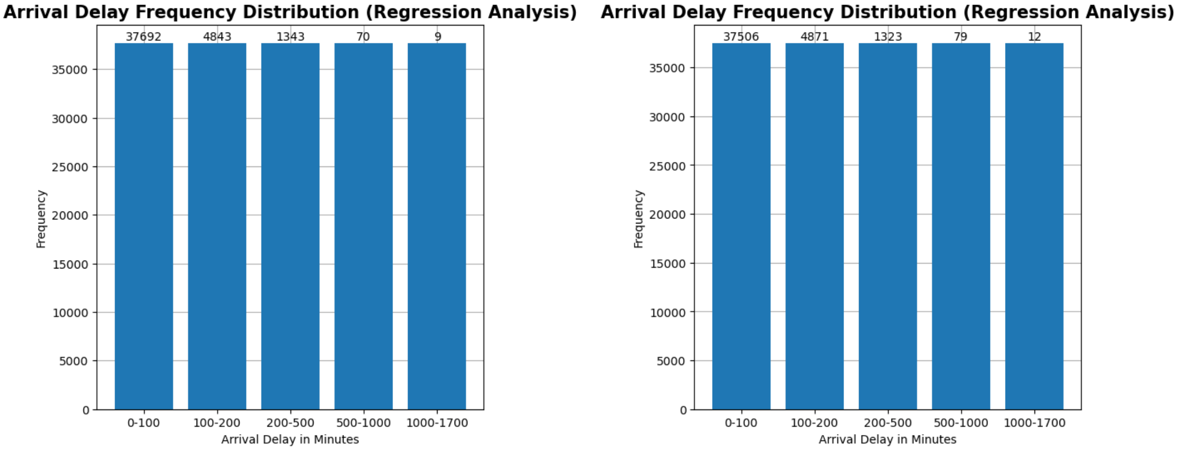


(a) Random Undersampler

(b) Random Oversampler

Figure 4.31: Arrival Delay Density Distribution for Random Undersampler and Random Oversampler (Regression Analysis)

The performance of the XGBoost Regressor is evaluated across different established Arrival Delay period intervals for Random Undersampler and Random Oversampler techniques. These intervals are drawn taking into account that the minimum value of Arrival Delay is 0 minutes (on-time arrival with respect to scheduled arrival time) and that the maximum value of Arrival Delay is 1665 minutes. Regression is performed for various period ranges, predicting the number of minutes by which the flights will be delayed and evaluating and comparing, with performance metrics, the Regression algorithm for the different ranges of the predicted variable.



(a) Random Undersampler

(b) Random Oversampler

Figure 4.32: Arrival Delay Frequency Distribution for Random Undersampler and Random Oversampler (Regression Analysis)

XGBoost Regressor performance for sampled data (Random Oversampler), which is previously classified by Random Forest Classifier, evaluated within different ranges of Arrival Delay in minutes is:

XGBoost Regressor Arrival Delay Period Intervals	**MSE**	**RMSE**	**MAE**	**R2 Score**
0 -100 (Minutes)	75.85	8.71	6.41	0.8576
100 - 200 (Minutes)	143.26	11.97	8.10	0.8007
200 - 500 (Minutes)	166.16	12.89	9.10	0.9639
500 - 1000 (Minutes)	271.69	16.48	12.93	0.9846
1000 - 1700 (Minutes)	410.00	20.25	18.20	0.9742

Table 4.17: Performance Metrics for Arrival Delay Period Intervals (XGBoost Regressor)

This analysis allows to draw conclusions about the ranges in which the model correctly predicts the delays. The MSE, MAE and RMSE scores indicate that Regression is more effective at lower delay intervals. The lowest values of errors occur in the interval from 0 to 100 minutes of delay. The reason behind this fact is that a high percentage of all the dataset delayed flights lie within this range.

Observing the dataset distribution, it indicates that the majority of arrival delays are part of the 0 to 100 minutes interval. Since the majority of the data used to train the regressor belongs to this interval, it has the least MAE and RMSE scores. Only 79 and 91 flights in total (for Random Undersampler and Random Oversampler, respectively) are collected in the ranges from 500 to 1000 and 1000 to 1700 minutes, which is a very small value compared to the dataset's total number of flights. So, by consequence, this ranges delays can be considered negligible and the obtained regressor's results can't be considered representative. XGBoost Regressor performs correctly even in the range from 200 to 500 minutes, as a RMSE value of 12.89 minutes and a MAE score of 9.10 minutes is acceptable when compared to the actual Arrival Delays.

5

Conclusions and Future Work

5.1. Conclusions

Taking into account the tools and software applied in the development of this project, the use of Python has been essential and the multiple libraries available makes it a powerful tool for flight delays data analysis. They have been used for data filtering and application of Machine Learning algorithms.

However, for full exploitation of all its potential, it is necessary to have a previous knowledge base and this could be an important barrier for investigators interested in this matter. Nevertheless, the combination of the Integrated Development Environment (IDE) (Jupyter Lab) and Python makes it easier to learn than other programming languages. The merging of the various libraries, its resources and documentation makes it the best tool in the market nowadays for data processing, with costs only associated to hardware. Overall, this project has proven that the use of Python and its libraries can be an ideal way for the application of Data Science disciplines in flight delays data analysis and it opens the door for future projects.

By Exploratory Data Analysis (EDA), the different conditions that can cause flight delays (weather conditions, departure and arrival queues, late aircraft, National Airspace System (NAS) or security reasons) were analysed and the patterns behind the delays were located. This analysis enabled to summarise the feature set's key features and to understand the relationships between the different variables. The extracted conclusions from the Exploratory Data Analysis (EDA) are applied for the construction of the two-stage Machine Learning for flight delay prediction. There were several goals for this project that were accomplished. Some attributes were discarded in favour of those that were found to be more useful for modelling. In addition, by altering the original characteristics from the dataset used for the prediction, additional features that may be used for modelling were produced. Valuable attributes for modelling were discovered and maintained, while other attributes were dropped. In addition, new features available for modelling were created by transforming original features from the dataset for the prediction.

To train a two-stage Machine Learning model and forecast flights arrival delay, the flight data was pre-processed. There was a built-in bias in favour of the majority class, not delayed flights. In order to remove the bias (solve the disparity between the two classes (delayed and not delayed flights)), the data was sampled using different techniques before classification (Over and Under Sampling methods). The results from all the resampling procedures show that sampling methods moderately improve the scores of the minority class by balancing the dataset, so sampled data (Random Oversampler method) was used for the Classification process. Logistic Regression, Decision Tree Classifier, Gradient Boosting Classifier, XGBoost Classifier, Random Forest Classifier, Extra Trees Classifier and SVM (Support Vector Machine) Classifier are the seven base methods that were modelled for classification purposes. Different algorithms were compared in terms of performance metrics results. After analysing the performance of the models, Random Forest Classifier and XGBoost Regressor were determined as the most accurate classifier and regressor, outperforming all the other algorithms.

The Random Forest Classifier provided the best F1 score (0.97 for Class 0 and 0.87 for Class 1), Precision (0.97 for Class 0 and 0.89 for Class 1) and Recall (0.98 for Class 0 and 0.85 for Class 1) scores across the different classification techniques.

Also, Balanced Accuracy and the F1-Score for the test set have high values comparing with the other analysed algorithms (0.915 and 0.872, respectively).

Taking into account the Regression models, XGBoost Regressor achieved the lowest MSE, RMSE and MAE results (74.69, 8.64 and 6.14, respectively) and the highest R^2 Score (0.9822) for sampled data (Random Oversampler technique), when comparing with all the analysed Regression models.

Different pipelined models were created, tested and compared, feeding the regressor models with the classsifier outputs (XGBoost and Random Forest Classifiers considering Random Undersampler and XGBoost, Random Forest and Extra Trees Classifiers considering Random Oversampler). XGBoost Regressor (best Regression model) was tested with the classifier results and it can be concluded that the best evaluation metrics results (highest Precision, Recall, F1-Score and R^2 Score and lowest MSE, RMSE and MAE (91.07, 9.54 and 6.83, respectively)) are achieved for Random Forest Classifier applied to sampled data (Random Oversampler) and XGBoost Regressor model. In conclusion, the Machine Learning model performed well and the prediction of the flight delay was accurate.

5.2. Future Work

There is extra work that can be performed with this flight delays dataset, like using a Neural Network algorithm for training the model, in order to strengthen the project. Moreover, other over sampling methods can be applied, such as Adaptive Synthetic (ADASYN), which prevents overlapping of the synthetic observations. Also, additional under sampling methods, like Tomek-Link (T.L.) or Condensed Nearest Neighbor (CNN), which apply data cleaning procedures, are also used to handle imbalanced data. In addition to different resampling strategies, Cost-Sensitive Learning can be applicated. It is a field of Machine Learning that takes into account the costs of prediction mistakes (imposing penalties on certain outcomes), when a Machine Learning model is in the training phase. Additionally, a hybrid strategy (under and over sampling) can be utilised to handle imbalanced data, such as SMOTEBoost.

Additional areas of improvement could be to include external datasets such as weather data to complement and complete the flight delay analysis. Also, the geographical area covered by the flights in the dataset can be studied, by picking the origin and departure airports location and the volume of flights in each of the airports for the year 2015. Including additional data from external sources can allow the models to obtain better prediction results.

Another field that can be further developed is feature selection. Performing Exploratory Data Analysis (EDA), some dataset features were removed from the model. Nonetheless, if all the remaining features would not have been included, more effective models might have been developed. Outputs from algorithms such as LASSO, Grid Search or Stepwise Selection may produce a better feature set optimization. Additionally, model's default parameters are applicated in this project, so there is room for development in the model's tuning discipline. This progress can be done by performing an hyper parameters analysis and finding the best hyper parameters values for the creation of the most optimised model.

Bibliography

[1] R. M. Baiada. Air traffic control is not the real cause of airline delays. URL https://www.forbes.com/sites/currentaccounts/2017/03/23/air-traffic-control-is-not-the-real-cause-of-airline-delays/#33bb6ca2c375.

[2] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 2007.

[3] Jupyter. Jupyterlab. URL https://jupyter.org/.

[4] T. N. C. of Excellence For Aviation Operations Research. Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the united states. URL https://isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_10_18_10_V3.pdf.

[5] U. S. D. of Transportation. Bureau of transportation statistics. URL https://www.bts.gov/.

[6] B. of Transportation Statistics. Understanding the reporting of causes of flight delays and cancellations. URL https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations/.

[7] B. of Transportation Statistics (BTS). Flight delays and cancellations of domestic flights operated by large carriers. URL https://www.bts.gov/.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 2011.

[9] G. V. Rossum. Python. URL https://www.python.org/.

[10] P. D. Team. Pandas-dev/pandas: Pandas. *Zenodo*, 2020.

[11] S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 2011.

[12] A. Vera. Flight delay eda (exploratory data analysis). URL https://www.kaggle.com/code/adveros/flight-delay-eda-exploratory-data-analysis.

[13] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature methods*, 2020.

[14] w3resource. Pandas dataframe: describe() function. URL https://www.w3resource.com/pandas/dataframe/dataframe-describe.php.

[15] M. L. Waskom. Seaborn: Statistical data visualization. *Journal of Open Source Software*, 2021.

All the project's source code is available at the following GitHub repository:
`https://github.com/MarcosAlbendeaMembrilla/TFM.git`