



# A Systematic Approach to Deal with Highly Imbalanced Data when Predicting Flight Cancellations and Delays

Rik Hendrickx



# A Systematic Approach to Deal with Highly Imbalanced Data when Predicting Flight Cancellations and Delays

by

Rik Hendrickx

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Tuesday September 15, 2020 at 14:00h.

Student number: 4380908  
Project duration: October 15, 2019 – September 15, 2020  
Thesis committee: Dr. Ir. B.F. Lopes dos Santos, TU Delft, Chairman  
Dr. M.A. Mitici, TU Delft, Supervisor  
Dr. Ir. D.M. Pool, TU Delft, External Member

*Cover photo: Jeffrey Schäfer.*



"When once you have tasted flight, you will forever walk the earth with your eyes turned skyward, for there you have been, and there you will always long to return."

- *Leonardo da Vinci*



# Preface

Dear reader

The document you are about to read represents my final thesis work, which is the last step towards obtaining the degree of Master of Science at the Delft University of Technology, faculty of Aerospace Engineering. This work originates from my deeply rooted passion and fascination towards civil aviation. The complexity of air transport operations introduces great challenges to be solved and with this research I do my best to contribute knowledge to the field of on-time performance prediction. The main question I try to answer, is how to successfully perform flight cancellation and delay predictions, while effectively dealing with their inherent data imbalance.

I would like to acknowledge and thank some people who made this thesis not only possible, but also interesting, fun and challenging. First of all, I would like to express my gratitude towards the department of Air Transport Operations of the Delft University of Technology and more in particular to Mrs. Mihaela Mitici, for the daily supervision and guidance and to Mr. Mike Zoutendijk, for joining the supervision team and providing useful feedback. Furthermore, from Royal Schiphol Group I would like to thank Mr. Jeffrey Schäfer and Mrs. Hélène van Riemsdijk-Schouten for providing the highly essential flight schedule data and airport operations expertise, and Mr. Kevin van Haagen, for introducing a challenging test environment.

Lastly, I will be ever grateful to my friend Hadewich De Meester, my partner Mike Slotweg and my parents for their never ending love and support and the much needed distraction in difficult and stressful times.

After six challenging, yet inspiring years, my life as a student will come to an end. As much as I have anticipated this moment, it will still be very difficult to say goodbye to this great old life. However, this end also marks the beginning of a new chapter. A chapter full of new challenges, surprises and joyful moments I am very much looking forward to. Unfortunately, due to the Corona crisis, I am not able to celebrate this moment with all my family and friends. Still, when this crisis is over, we *will* celebrate. Hence, I am given a new special moment in Delft to look forward to. Now, without further ado, I introduce you into my final work at the TU Delft, which I am very proud of. May this thesis prove valuable to its readers and inspire future work in similar or even different research fields.

*Rik Hendrickx*  
*Delft, September 2020*





# Contents

	iii
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Abbreviations</b>	<b>xiii</b>
<b>I Scientific Paper</b>	<b>1</b>
<b>II Literature Study (Previously graded under AE4020)</b>	<b>21</b>
<b>Abstract</b>	<b>23</b>
<b>1 Introduction</b>	<b>25</b>
<b>2 Methodology</b>	<b>27</b>
<b>3 Data Management</b>	<b>29</b>
3.1 Sources & Features . . . . .	29
3.1.1 Flight Data . . . . .	29
3.1.2 Weather Data . . . . .	30
3.2 Pre-processing . . . . .	32
3.3 Encoding . . . . .	33
3.4 Dealing with Imbalanced datasets . . . . .	34
3.5 Feature Selection . . . . .	35
<b>4 Flight Cancellation Determinants</b>	<b>37</b>
<b>5 Machine Learning</b>	<b>41</b>
5.1 Machine Learning Fundamentals . . . . .	41
5.1.1 Definition . . . . .	41
5.1.2 Learning Types & Tasks . . . . .	41
5.2 Train-Test Split . . . . .	43
5.3 Prediction Performance . . . . .	43
5.3.1 Determinants of Performance . . . . .	43
5.3.2 Confusion Matrix . . . . .	44
5.3.3 Area Under the ROC Curve . . . . .	45
5.3.4 Regression Error Metrics . . . . .	47
5.4 Algorithms . . . . .	48
5.4.1 Decision Tree . . . . .	49
5.4.2 Random Forest . . . . .	49
5.4.3 Neural Networks . . . . .	49
5.4.4 k-Nearest-Neighbours . . . . .	50
5.4.5 Logistic Regression . . . . .	50
5.4.6 Boosting Algorithms . . . . .	50
<b>6 Research Approach</b>	<b>51</b>
6.1 Knowledge Gap . . . . .	51
6.2 Scope . . . . .	51
6.3 Research Question and Objective . . . . .	51
6.3.1 Research Question . . . . .	52
6.3.2 Research Objective . . . . .	52

---

6.4	Research Framework . . . . .	52
6.5	Research Planning . . . . .	54
<b>7</b>	<b>Conclusion</b>	<b>55</b>
<b>III</b>	<b>Elaborations on Thesis Work</b>	<b>57</b>
<b>8</b>	<b>Imbalance</b>	<b>59</b>
<b>9</b>	<b>Features</b>	<b>61</b>
9.1	Additional Features . . . . .	61
9.2	Feature Selection . . . . .	62
	<b>Bibliography</b>	<b>65</b>
<b>A</b>	<b>Gantt Chart</b>	<b>67</b>

# List of Figures

2.1	General methodological flow and layout in topical literature about flight delay and cancellation prediction, linked to the structure of this literature review. . . . .	28
3.1	Example of One-Hot Encoding. . . . .	33
3.2	Example of Target Encoding. . . . .	34
3.3	Example of Periodic Encoding. . . . .	34
3.4	Example of Ordinal Encoding. . . . .	34
4.1	Sample period of flight cancellations (left) and route-month percentage of flight cancellations and airline alliances (right). . . . .	38
5.1	Visualisation of the machine learning pipeline. . . . .	42
5.2	An example of a machine learning model. . . . .	42
5.3	An example of a Receiver Operating Characteristic Curve. . . . .	46
5.4	ROC curves from literature, comparing performance with and without weather data. . . . .	48
5.5	An example of a simple Neural Network from literature. . . . .	50
6.1	Thesis machine learning model flow chart. . . . .	53
6.2	Research planning flow diagram. . . . .	54
8.1	Pseudocode of the SMOTE algorithm, explained. . . . .	59
9.1	Visualisation of the Pearson Correlation Coefficient. . . . .	63



# List of Tables

3.1	Summary of data sources and features for flight schedule data used in topical literature, with their target variable and prediction horizon. . . . .	30
3.2	Summary of data sources and features for weather data used in topical literature, with their target variable and prediction horizon. . . . .	31
3.3	Summary of data cleaning techniques used in topical literature, with corresponding target and prediction horizon. . . . .	32
3.4	Summary of encoding techniques used in topical literature, with corresponding target and prediction horizon. . . . .	33
3.5	A summary of data sampling methods in topical literature, to account for imbalanced data. . . .	35
3.6	Summary of feature selection techniques used in topical literature, with their target and prediction horizon. . . . .	36
5.1	An example of a confusion matrix. . . . .	44
5.2	Accuracy, precision and recall results from topical papers, with their target and prediction horizon. . . . .	45
5.3	Summary of classification thresholds used in topical literature, with corresponding target and prediction horizon. . . . .	46
5.4	AUC results from topical papers, with their target and prediction horizon. The best result is highlighted in bold. . . . .	47
5.5	Algorithms used in topical literature, with the target variable and prediction horizon. The algorithm with the highest performance in its respective paper is highlighted in bold. . . . .	49
9.1	Selected features for cancellations with explanation and correlation with the target 'cancelled'. .	64
9.2	Selected features for departure delay with explanation and correlation with the target 'delayed'. .	64
9.3	Selected features for arrival delay with explanation and correlation with the target 'delayed'. . .	64



# List of Abbreviations

<b>Abbreviation</b>	<b>Meaning</b>
AAS	Amsterdam Airport Schiphol
AC	Aircraft
AD	Arrival Delay
ASPM	Aviation System Performance Metrics
AUC	Area Under the ROC Curve
BTS	Bureau of Transportation Statistics
C	Cancellation
CCFP	Collaborative Convective Forecast Product
CIWS	Corridor Integrated Weather System
CV	Cross Validation
DD	Departure Delay
DL	Deep Neural Networks
DT	Decision Tree
FAA	Federal Aviation Administration
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
IQR	Interquartile Range
KNMI	Koninklijk Nederlands Meteorologisch Instituut
kNN	k-Nearest-Neighbours
LAMP	Localized Aviation MOS Product
LR	Logistic Regression
MAE	Mean Absolute Error
METAR	Meteorological Aerodrome Reports
MOR	Minority Over-sampling with Replacement
NCWD	National Convective Weather Diagnostic
NGS	Nagasaki Airport
NN	Neural Network
NOAA	National Oceanic and Atmospheric Administration
NRT	Narita Airport
RF	Random Forest
RFE	Recursive Feature Elimination
RMSE	Root Mean Squared Error
RMU	Random Majority Under-sampling
ROC	Receiver Operating Characteristic
RRMSE	Relative Root Mean Squared Error
R-SMOTE	Randomised Synthetic Minority Over-sampling TEchnique
RUS	Random Undersampling
SMOTE	Synthetic Minority Over-sampling TEchnique
TN	True Negatives
TP	True Positives
TPR	True Positive Rate
USA	United States of America
WITI	Weather Impacted Traffic Index
XGB	XGBoost





**I**

Scientific Paper



# A Systematic Approach to Deal with Highly Imbalanced Data when Predicting Flight Cancellations and Delays

Rik Hendrickx

Faculty of Aerospace Engineering, Delft University of Technology, HS 2926, Delft, The Netherlands

## Abstract

As on-time performance is one of the main contributors to success in the world of commercial aviation, predictions on flight delays and cancellations can significantly improve operational efficiency and thus quality of service. Since flight delays and cancellations are occasional and infrequent events, operational on-time performance data is inherently imbalanced. This is especially the case for cancellations, as on average 1.6% of flights are cancelled, while about 33% of the flights is delayed. For this research, flight operational data is combined with weather data to predict flight delays and cancellations on prediction horizons of hours to months before the flight, by means of Neural Network and Random Forest machine learning algorithms. Since these algorithms naturally tend towards the usage of balanced data, the need exists to find a systematic approach to deal with the imbalance issues, in order to make accurate predictions. Hence, an imbalanced data approach is proposed, which analyses model performance with indicators such as precision and F1-score on varying data imbalance ratios. The imbalance ratios are obtained through the use of sampling techniques such as Synthetic Minority Oversampling and Random Undersampling. It is concluded that the highest precision is found without any sampling while for the highest F1-score sampling is essential. Additionally, the research confirms that severely imbalanced data, like the cancellation data, yields the worst performance when compared to medium imbalanced data, like the delay data.

**Keywords:** Imbalanced data, Sampling techniques, Flight cancellations, Flight delays, Binary classification

## 1. Introduction

Flight on-time performance is an important measure for service quality. In 2018, more than 11 million flights were operated in Europe alone and, compared to 2017, the annual traffic has increased by 3.8% (Eurocontrol, 2018b). Due to the continuous increment in air-traffic and demand, airspace and airports are getting more and more crowded. They are operating at maximum capacity without being able to correspondingly increase it, leading to challenging flow situations. A capacity gap of 1.5 million flights or 8% of the demand has even been forecast by 2040 (Eurocontrol, 2018a). When also considering factors such as bad weather, eventually, flight delays and cancellations become inevitable. In 2018, the average delay per flight was estimated to be 14.7 minutes. This implies an increase of 17% compared to 2017, which saw an average delay of 12.3 minutes per flight (Eurocontrol, 2018c). The duration of these interruptions seems to keep growing, which leads to detrimental effects on the airline's quality record and hence they become a very costly obstacle (Alderighi and Gaggero, 2018). Therefore, it could be of great value for airlines, airports and travellers if there existed a way to accurately predict flight delays and cancellations.

Fortunately, airport coordinators have access to multiple strategic flight data sources, which could provide insights with respect to flight delays and cancellations, if assessed in a mathematical, predictive fashion. Unfortunately, there is no single, optimal way for carrying out prediction problems. However, in the last decade, there is one technique that has gained a lot of momentum within data science and especially within prediction research (Sternberg et al., 2017). It goes by

the name of machine learning and it provides a powerful way to make predictions based on what it learned from (past) data. If these self learning algorithms succeed in predicting on-time performance, crucial bottlenecks could be revealed, even on different moments in time. Days, weeks and even months ahead of the operation, predictions could be made, allowing airport coordinators, passengers and airlines to reap the benefits of these insights.

As flight delays and especially cancellations are infrequent events, the data available to carry out these predictions will have an imbalanced class distribution. This implies that more flights are likely to be on time, compared to the amount of flights that are cancelled or delayed. This data imbalance can tremendously interfere with the correct classification of these flights, since practically all classification algorithms assume balanced class distributions and are intended to optimise for classification accuracy (Zhao et al., 2018). Multiple techniques exist to cope with the imbalance problem and one of them is the utilisation of sampling techniques. By respectively under- or oversampling the majority or minority classes, the classification performance could be optimised. Hence, the imbalance problem introduces the need for the first goal of this research, namely an appropriate systematic approach to deal with this kind of situation, in order to reach the second goal, the successful classification of flight delays and cancellations.

In response to the need for a solution to the imbalance problem, this paper presents a systematic approach to analyse and deal with the effects of highly imbalanced datasets. This is done by means of sampling techniques and machine learning algorithms,

while assessing the binary classification performance of flight delays and cancellations at an airport for individual flights. The analysis is performed on different prediction horizons, namely 1 hour (only for delays), 1 day, 1 week and 6 months prior to the flight, allowing airport coordinators to assess eventual bottlenecks at various operational levels and timings. In addition to flight operational data, weather data is included to broaden the spectrum of influential factors with respect to on-time performance. Amsterdam Airport Schiphol (AAS) provides a set of flight operational data, whereas weather data is obtained through the Koninklijk Nederlands Meteorologisch Instituut (KNMI, Royal Dutch Meteorological Institute) and METAR weather reports (KNMI, 2020; IowaStateUniversity, 2020). By combining these specific datasets and sources, a realistic scenario of a large, busy European hub-airport is reflected. To the best of our knowledge, this paper is the first to propose a systematic approach to deal with the inherent imbalance of airport operational on-time performance datasets.

This paper contributes new insights to the current body of knowledge concerning highly imbalanced datasets and air transport on-time performance. More in particular, up until now, flight cancellations have been given a lot less attention than flight delays in research. Hence, the inclusion of both on-time performance issues, together with the systematic approach to deal with their inherent data imbalance, can definitely help uncover new insights that could eventually benefit the industry. Additionally, AAS could benefit from this research as it might allow them to assess flight schedules at different times before the operation and act early in order to avoid or reduce the amount of flight delays and cancellations.

The remainder of this paper is structured as follows. Section 2. discusses the binary classification model, covering the features, algorithms and performance indicators. Subsequently, section 3. presents the systematic approach to deal with the inherent data imbalance and addresses the classification results. Section 4. elaborates on the results in a discussion, whereas section 5. concludes the research and summarises the most important observations.

### 1.1. Related work

Multiple researches have been carried out on different topics regarding imbalanced data. Firstly, Zhao et al. (2018) establish an approach to handle imbalanced healthcare data by incorporating multiple different rebalancing or sampling techniques. The proposed framework successfully improves the detection of rare healthcare events due to look-alike sound-alike mix-ups. A 45% increase in relevant performance is observed when combining a logistic regression machine learning algorithm with a sampling technique called Synthetic Minority Oversampling Technique (SMOTE), (Chawla et al., 2002). Another article that studies the effects of data imbalance is Hassanzadeh

et al. (2014). In the article, four different rebalancing strategies are presented, combined with a binary classification framework for scientific artefacts in the evidence based medicine domain. A 15% increase in relevant performance is observed, with the appropriate rebalancing of the data.

Flight delays have been the centre of attention for many researches carried out in the past, whereas flight cancellations have not, unfortunately. Cao and Kanafani (1997) and Jarrah et al. (1993) are two examples utilising on-time performance data in order to propose an accurate decision-support tool, integrating flight delays and cancellations. They apply network models with minimum cost and maximum profit objectives, respectively. The tool should return an optimal set of flights either to delay or cancel. Furthermore, Seelhorst (2014) investigates flight cancellation behaviour by using an econometric discrete choice model. The purpose of the research is to find factors that influence flight cancellations and to predict cancellation probabilities. All of this is done on a timeline of 160 days ahead of the operation. Finally, the results of the research are incorporated in a queueing model, which visualises the effects flight cancellations have on flight delays. Alderighi and Gaggero (2018) research the effect of an airline being part of a global alliance on cancellations. The conclusion of the research is that airlines belonging to an alliance are more likely to have flight cancellations compared to non-alliance airlines.

The field of on-time performance has also been extensively combined with machine learning techniques. When assessing the classification task, multiple literature sources can be found. Firstly, Choi et al. (2016) assess the prediction of airline delay, on prediction horizons of 5 days, 1 day and 0 days. The authors utilise multiple classification algorithms, namely Decision Trees, Random Forest, AdaBoost and the k-Nearest-Neighbors classifier. They also investigate the contribution of adding weather data to the flight operational data, which is found to increase the prediction performance for most of the algorithms. The Random Forest classifier is found to have the best classification performance. Secondly, Horiguchi et al. (2017) predict flight delays with prediction horizons of 5 months, 1 week and 1 day before the operation. Multiple algorithms are tested for optimal performance, which are Random Forest, XGBoost and Deep Neural Network. The authors train and test their algorithms on airline data, originating from a low cost carrier and decide not to include weather data. They conclude that the models can effectively predict flight delay on the prediction horizon of one day before the operation, in specific airports and weeks. Thirdly, Lambelho et al. (2020) predict not only flight delays, but also flight cancellations and is therefore a rather unique paper. Also, the authors use the prediction outcomes to rank strategic flight schedules for London Heathrow Airport. Furthermore, Kim et al. (2016) focus on

deep learning algorithms to predict flight delays, several hours before the operation. Weather data is also included in this research and the authors limit their data to USA airports. It is found that the Recurrent Neural Networks architecture results in a reliable delay prediction of a single day. Chen and Li (2019) propose an air traffic delay prediction model combining multi-label Random Forests and an approximated delay propagation model. The authors conclude that their approach appears to be practical and accurate for flight delay prediction. Additionally, they find that departure delay and late arriving aircraft delay are the most important data features for the prediction. Finally, Alonso and Loureiro (2015) perform multiclass predictions for flight departure delay at Porto Airport, several hours before the flight. Interestingly, the prediction result is one of multiple classes, bound by delay times. Examples of these classes are [0,15], ]15,30] and ]30,60] (all minutes delay). Neural Networks and Decision Trees are used in the prediction, with the Neural Networks resulting in simpler implementation and better test results.

Not only classification tasks, but also regression tasks (estimating delay time instead of predicting whether a flight is delayed or not) are widely covered in literature. Manna et al. (2017) investigate the prediction of flight delays with a prediction horizon of several months before the operation for USA airports. Using Gradient Boosted Decision Trees, the authors find that the model is a good predictor of flight delay patterns with good accuracy and limited errors. Next, Kalliguddi and Leboulluec (2017) estimate flight delay, with Random Forests outperforming any other of the algorithms under consideration. The prediction horizon here is several hours ahead of the operation. The authors also conclude that late aircraft delay, carrier delay, weather delay and national airspace delay have the most effect on on-time performance.

## 1.2. Data description

In order to successfully predict flight delays and cancellations, a multitude of datasets is used as a solid training base for the machine learning algorithms. First and foremost, flight operational data forms the foundation for the final dataset. The data is provided by AAS. For the flight cancellations, the data ranges from 2015 up to and including 2018. This dataset contains 1,956,418 flights and is centered around origin-

destination (O-D) pairs, in which either the origin or the destination is always AAS. Furthermore, it encompasses informative data elements (from now on referred to as features) such as date, time, origin or destination airport, airline, flight number, etc. For a complete list of features of the flight schedule datasets and the weather datasets, please consult Table 1. Of all flights in the set, 30,695 or 1.6% of them is cancelled. This is expected as flight cancellations are infrequent events. However, it implies that the dataset is extremely imbalanced. A cancelled flight is defined as a flight that was scheduled to fly, however eventually did not. The flights are operated by 256 different airline companies, flying to and from 649 unique origin/destination airports. About 54% of all flights are within the Schengen zone.

The flight operational dataset for flight delays is slightly different as it spans all flights in 2019, counting as many as 479,400 flights. In contrast to the data for the cancellation prediction, this dataset is centered around turnarounds and follows a single aircraft arriving and departing again. This allows for a more elaborate analysis, as arrivals and departures, which might influence each other in terms of delay, are coupled. Some examples of data features from this dataset are airline, in-block time, off-block time, scheduled turnaround time, origin or destination airport, departure delay, etc. This dataset is also imbalanced, however, less severely imbalanced than the cancellation set. 82,350 flights depart with a delay, which is 34% of all departures, and 57,253 of the arrival flights arrive with a delay, which represents 24% of all arrivals. The definition of a delayed flight, is when the actual departure or arrival time is later than or equal to 16 minutes, as this is the threshold utilised by AAS. The histograms in Figure 1 show the delay distribution of the flights, with the number of flights on the y-axis and the departure/arrival delay in minutes on the x-axis. Furthermore, there are 49 different aircraft types and 99 different airlines present in the dataset, flying from 336 unique origin airports and to 323 unique destination airports.

The second main data type is weather data. Two different weather datasets are incorporated in the research, namely the weather at AAS, provided by KNMI (KNMI, 2020) and the weather at the origin or destination airport (other than AAS), provided in METAR reports (IowaStateUniversity, 2020). The

Table 1: Data features in the original delay, cancellation and weather datasets combined. Detailed feature explanation is performed in section 2.1.2. (n=numerical feature, c=categorical feature, p=periodical feature).

dataset	Features
Weather	Wind speed & direction <sup>n</sup> , Gust speed <sup>n</sup> , Temperature <sup>n</sup> , Dew point Temperature <sup>n</sup> , Sunshine time <sup>n</sup> , Global radiation <sup>n</sup> , Precipitation time <sup>n</sup> , Precipitation amount <sup>n</sup> , Pressure <sup>n</sup> , Horizontal visibility <sup>n</sup> , Cloud coverage <sup>n</sup> , Relative humidity <sup>n</sup> , Mist <sup>n</sup> , Rain <sup>n</sup> , Snow <sup>n</sup> , Storm <sup>n</sup> , Ice <sup>n</sup>
Flight schedule	Flight number <sup>c</sup> , Airline <sup>c</sup> , Handler <sup>c</sup> , Aircraft type <sup>c</sup> , Aircraft registration <sup>c</sup> , Flight nature <sup>c</sup> , Service type <sup>c</sup> , Codeshare <sup>c</sup> , Data and Time <sup>p</sup> , Turnaround time <sup>c</sup> , Aircraft category <sup>c</sup> , Widebody or Narrowbody <sup>c</sup> , Remote or Connected gate <sup>c</sup> , Airport <sup>c</sup> , Schengen <sup>c</sup> , Continent <sup>c</sup> , Country <sup>c</sup> , Number of daily visits <sup>n</sup>

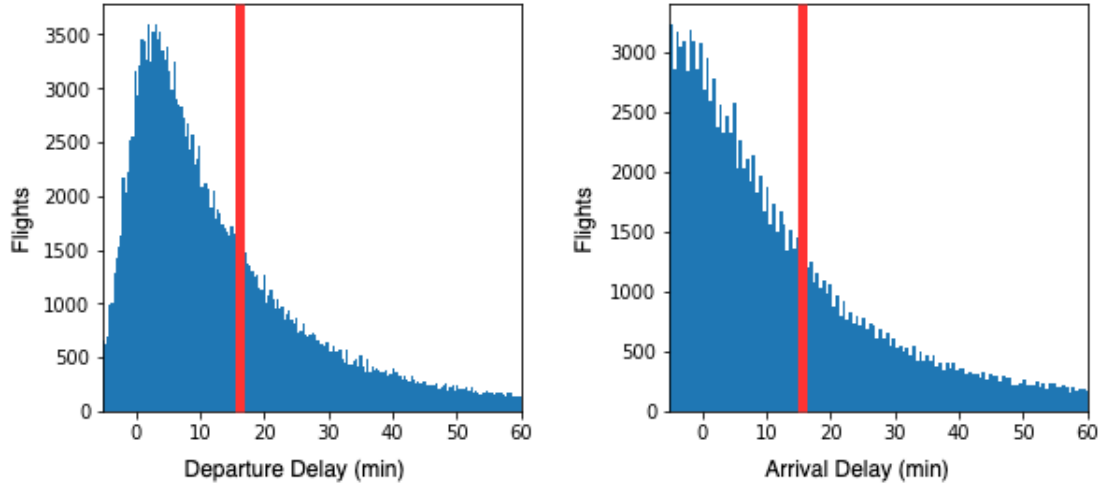


Figure 1: Histogram showing the departure and arrival delay distribution in the delay dataset. Vertical red marker shows the delay threshold of 16 minutes.

KNMI set contains hourly weather data from 2015 up to and including 2019, whereas the METAR weather includes half-hourly weather data, likewise from 2015 up to and including 2019. Both datasets include features such as temperature, wind speed, visibility, pressure, etc. The KNMI dataset is the smallest, with approximately 42 thousand samples, while the METAR dataset includes more than 45 million data samples. This is due to the fact that the set encompasses 5 years of weather for all origin and destination airports served from AAS.

## 2. Binary classification model for flight delay and cancellation prediction

In this section, the binary classification model is covered entirely, encompassing the feature encoding, feature selection, binary classification algorithms and model performance indicators. A flow diagram for the model can be seen in Figure 2.

### 2.1. Features

#### 2.1.1. Feature encoding

The first step in preparing the data to be fed into the machine learning algorithms, is making sure it is well-structured and numerical. It may not contain

any missing values. Therefore, after having interpolated and removed some missing features, it is time to look at the categorical data features. Machine learning algorithms can only process numerical information. Therefore, it is essential to perform encoding on the categorical data, i.e. turning words and letters, such as the airline name, into numbers. Three different encoding techniques are used in the research. A simple form of binary encoding, target encoding and periodic encoding. Examples of the encoding techniques can be found in Table 2.

Firstly, the categorical features containing a lot of categories (e.g. the feature Airport contains approximately 650 different categories), are target encoded. The correlation with the target variable (cancelled or delayed) is determined per flight. This is then translated into the probability that the variable is cancelled or delayed, which represents that specific variable in the dataset (Lambelho et al., 2020). Secondly, periodic data features such as hour, day of week and month are encoded using trigonometric functions, to account for their periodicity (Horiguchi et al., 2017). In the example in Table 2, the feature *Month* is encoded, which ensures that month 12 (December) and month 1 (January) are sequential months. This is done by introduc-

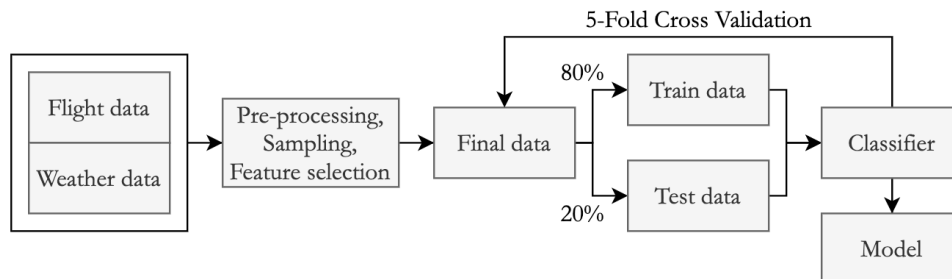


Figure 2: A flow diagram of the model.

Table 2: Example of the encoding techniques.

Original features			Encoded features		
City (A)	Month (B)	Cancelled (C)	Target encoding (A)	Periodic encoding (B)	Binary encoding (C)
London	1	Yes	1	$\sin(\frac{2\pi 1}{12}), \cos(\frac{2\pi 1}{12})$	1
Brussels	3	Yes	0.5	$\sin(\frac{2\pi 3}{12}), \cos(\frac{2\pi 3}{12})$	1
Paris	5	No	0.33	$\sin(\frac{2\pi 5}{12}), \cos(\frac{2\pi 5}{12})$	0
London	7	Yes	1	$\sin(\frac{2\pi 7}{12}), \cos(\frac{2\pi 7}{12})$	1
Paris	8	Yes	0.33	$\sin(\frac{2\pi 8}{12}), \cos(\frac{2\pi 8}{12})$	1
Paris	10	No	0.33	$\sin(\frac{2\pi 10}{12}), \cos(\frac{2\pi 10}{12})$	0
Brussels	12	No	0.5	$\sin(\frac{2\pi 12}{12}), \cos(\frac{2\pi 12}{12})$	0

ing a sine and cosine component for month  $m$ , which are defined as  $\sin(\frac{2\pi m}{12})$  and  $\cos(\frac{2\pi m}{12})$ . Lastly, the binary encoding technique is applied to features containing binary information, such as *arrival* or *departure*, *snow* or *no snow* and *Schengen* or *Non-Schengen*. This is simply changed into 1 and 0.

After the encoding, in order to eliminate any unwanted feature domination or ranking, feature scaling to a 0-1 scale is performed (Horiguchi et al., 2017; Choi et al., 2016).

### 2.1.2. Feature selection

When the flight operational and weather datasets are merged, one final, large dataset is obtained. This results in a high number of features in both the cancellation and delay case. It has been proven that incorporating a large number (or all) of the features is not always beneficial for prediction performance (Flach, 2012). Having too many features could result in higher computational loads and longer run times, but also in decreased performance, as not all features have a high target correlation. This is also known as ‘the curse of dimensionality’ (Pechenizkiy, 2005). Therefore, it is important to select a set of the most relevant features.

In this paper, the main feature selection technique is based on Pearson’s correlation coefficient. This coefficient measures the linear association strength between two features. A correlation coefficient of +/-1 resembles a perfect positive/negative correlation. The higher the correlation of a feature with the target, the better performance this feature has in classification. However, when comparing features with other features for inter-correlation (so not with the target variable), a high correlation coefficient could lead to multicollinearity for some machine learning classifiers. Hence, one of the two features must be abandoned, preferably the one with the lowest correlation with the target. This method is an example of a *filter* method, with a subset of relevant features going into the model after selection/filtering. The minimum threshold for selecting a feature is set at +/-0.1 and the maximum threshold for multicollinearity is set at +/-0.8 for cancellations and +/-0.7 for delays.

The selected sets of features can be seen in Tables 3, 4 and 5, for cancellations, departure delay and arrival

delay respectively. From left to right, the first column shows the feature name, the second column gives a short explanation of the feature and the remaining columns show whether the feature is incorporated in the dataset, for each prediction horizon. An S indicates the inclusion of a scheduled feature (known months in advance), an A indicates the inclusion of an actual feature (only possible 1 hour in advance as the actual information of that day is used) and a W indicates a weather feature. The different types of weather forecast features are explained in the next paragraph. Please note that there is a multitude of features that is created and added to the initial datasets, but they are filtered out by the feature selection. These are features such as distance between airports, monthly route frequency, monthly route market share, aircraft seats, last flight of the day indicator, airline alliances, seasons and days of the week.

Since weather forecasts are less accurate or unavailable for earlier prediction horizons of 1 week and 6 months, the weather features consist of averaged weather measurements for these horizons. For the prediction horizon of 1 day before the flight, the hourly weather measurements are used from the KNMI set, as they are. This implies that it is assumed that the weather is predicted without error one day ahead. For the prediction horizon of 1 week, a daily average is taken of all weather features. Analogously, for the 6 months prediction horizon, a monthly average is taken. For the METAR weather, a different averaging technique is applied, since this concerns the weather at the origin and destination airport. In the data there is no information on the departure time at the origin (in case of arrival at Schiphol) or on the arrival time at the destination (in case of departure at Schiphol). As it is difficult to estimate flight times due to different flying speeds combined with shifting time zones, it is chosen to take 6-hourly average weather information at the origin or destination airport (not Schiphol) for the horizons of 1 hour and 1 day. This assumption encompasses the flight times and time zones and still gives a 6-hourly averaged indication of the weather at the airport. For the horizons of 1 week and 6 months, again a daily and monthly average is taken.

Table 3: Selected features for cancellations. (S=Scheduled, A=Actual, W=Weather).

Feature	Explanation	1 day / 1 week / 6 months
Airport	Origin or destination airport of the flight	S
Flight Number	Unique flight number of the flight	S
Country	Origin or destination country of the flight	S
Airline	Airline company operating the flight	S
Servicetype	Category of the commercial flight; passenger or freight	S
AC Registration	Registration number of the aircraft operating the flight	S
Handler	Apron handler, handling baggage, fuel,...	S
Wind Speed	Windspeed at origin/destination airport	W
Pressure	Air pressure at origin/destination airport	W
Visibility	Horizontal visibility at the origin/destination airport	W
Snow	Indicator is snow presence at the origin/destination airport	W

Table 4: Selected features for departure delay. (S=Scheduled, A=Actual, W=Weather).

Feature	Explanation	1 hour	1 day / 1 week / 6 months
Flight number	Unique flight number of the flight	S	S
AC registration	Registration number of the aircraft	S	S
AC type	Type of the aircraft operating the flight	S	S
Handler	Apron handler, handling baggage, fuel,...	S	S
Airline	Airline company operating the flight	S	S
Destination airport	Destination airport of the flight	S	S
Daily visits	Number of times this route is operated per day.	S	S
Month	Month in which the flight is operated	S	S
Time	Time at which the flight is operated	S	S
Total arr (past hr)	Total number of arrivals in the past hour	A	S
Total dep (past hr)	Total number of departures in the past hour	A	S
Wind gust speed (origin)	Maximum wind speed at the origin.	W	W
Temperature (origin)	Temperature at the origin.	W	W
Temperature (destination)	Temperature at the destination	W	W
Total arr delay (past hr)	Total minutes of arrival delay in the past hour	A	
Total dep delay (past hr)	Total minutes of departure delay in past hour	A	
Arrival Delay	If the flight had a delay when it arrived	A	

Table 5: Selected features for arrival delay. (S=Scheduled, A=Actual, W=Weather).

Features	Explanation	1 hour	1 day / 1 week / 6 months
Flight number	Unique flight number of the flight	S	S
AC registration	Registration number of the aircraft	S	S
AC type	Type of the aircraft operating the flight	S	S
Handler	Apron handler, handling baggage, fuel,...	S	S
Origin airport	Origin airport of the flight	S	S
Month	Month in which the flight is operated	S	S
Time	Time at which the flight is operated	S	S
Wind gust speed (destination)	Maximum wind speed at the destination	W	W
Total arr delay (past hr)	Total minutes of arrival delay in past hour	A	
Total dep delay (past hr)	Total minutes of departure delay in past hour	A	



## 2.2. Binary classification algorithms

Supervised learning algorithms are needed to perform the binary classification task for flight delays and cancellations. Numerous algorithms exist that are able to perform this task, however, given that this research is bounded in resources, it is important to only select a few and compare their performance. Therefore, two popular types of algorithms are chosen in order to compare their performance. These are Random Forests (RF) and Neural Network (NN). However, before the algorithms are covered, it needs to be highlighted how they work based on the data that is fed into them.

Machine learning algorithms need training data, in order to *learn*, and test data, in order to *evaluate* how good they were trained. Typically, this train-test split is taken at 70%-30% or 80%-20% respectively. Additionally, a method called K-Fold Cross Validation (K-F CV) makes this aspect more robust. This technique is a process of training the same data set K times, each time with different  $(100 - \frac{100}{K})\%$  batch of the data used for training and  $\frac{100}{K}\%$  batch of the data used for validation. A 5-F CV is used in the models, defining the train-test split as 80%-20%, before the data is fed into the algorithms.

In terms of the algorithms themselves, firstly there is the RF. It is composed of multiple Decision Trees, hence it is called an ensemble method. A large group of independent trees is assembled, after which their verdicts are averaged, reducing the variance. All trees in the group are noisy but unbiased. Each tree carries out a class vote, after which the RF classifies using the majority vote (Choi et al., 2016).

Secondly, a NN consists of multiple layers of neurons, stacked together in order to produce a final output. The first and last layer are called the input and output layer and all layers in between are called hidden layers. The neurons are functions of the outputs of all neurons in the previous layer and have activation functions that are fired (activated) when a certain threshold is reached. Popular activation functions are *ReLU*, *Tanh* and *Sigmoid*. The aim of the NN is to learn and set the network parameters, which are the biases and weights of every neuron in each layer, in order for the outcome to be equal to the groundtruth (Kuhn and Jamadagni, 2017). The term Deep Learning comes from Deep NN's, which essentially is a NN with multiple hidden layers, creating more 'depth'.

## 2.3. Performance indicators

To successfully perform this research, it is essential to evaluate how good or bad the models are performing. This can be done using several *performance indicators*. These indicators provide evidence of the capabilities of the model in terms of correctly predicting flight (arrival/departure) delays and cancellations. They are an essential part of the analysis, as multiple models and scenarios are being evaluated and the optimal

combination should be selected after detailed comparisons.

At the basis of most performance indicators lies the confusion matrix (CM). Essentially, the CM gives the number of class-dependent errors. An example of a CM for cancellation classification can be seen in Table 6.

Table 6: Example of a confusion matrix.

	Actually cancelled	Actually flying
Predicted cancelled	TP	FP
Predicted flying	FN	TN

The confusion matrix essentially projects the predictions (left-most column) on the actuals (top row). The following metrics are defined: true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) (Lambelho et al., 2020). The TP are the flights that are predicted as cancelled and actually are cancelled. Right next to the TP, there are the FP, depicting the flights that are predicted as cancelled by the model, but are actually flying. Below there are the FN, the flights that are predicted as flying but actually end up being cancelled. Finally, in the bottom-right there are the TN, or flights that are predicted to fly and actually fly. Several important performance metrics can be derived from the confusion matrix, namely accuracy, precision, recall, F1-score and the Area Under the Curve.

- *Accuracy* is 'how many of the flights were correctly predicted'. Translating this to the CM:  $\frac{TP+TN}{TP+FP+FN+TN}$ .
- *Precision* means 'how many of the flights that were predicted as cancelled (delayed), are actually cancelled (delayed)'. Mathematically, this is  $\frac{TP}{TP+FP}$ .
- *Recall* tells you 'how many of the actually cancelled (delayed) flights, were predicted correctly'. In CM terms, this is  $\frac{TP}{TP+FN}$ .
- *F1-score* gives the harmonic mean between the precision and recall, which means that it punishes the lowest values between the two.
- The Receiver Operating Characteristic curve (ROC) and more specifically the *Area Under the Curve (AUC)* plots the relation between the True Positive Rate (TPR), and the False Positive Rate (FPR), as a function of classification threshold. In terms of the CM, this translates to  $TPR = \frac{TP}{TP+FN}$  and  $FPR = \frac{FP}{FP+TN}$ .

### 3. Dealing with imbalance: a systematic approach

As has been mentioned during the data description, the cancellation dataset is highly imbalanced and the delay dataset is medium imbalanced. Since it has been proven more efficient for prediction performance of machine learning algorithms to have a balanced training and testing dataset, rather than an imbalanced one (Chawla et al., 2002; Gao et al., 2015), the need arises for a systematic approach to deal with the imbalance problem. This section presents this approach, called the imbalance analysis, in several steps. The entire systematic approach can be summarised in the flow diagram, visible in Figure 3. The step numbers correspond to the steps explained in the paragraphs below.

#### Step 1: Identify relevant performance metrics

The first step of the imbalance analysis, is identifying which performance metrics are worth observing. The three most important indicators, especially for imbalanced data, are precision, recall and F1-score, since they show the actual relations between the predicted situation and the actual situation. This is where accuracy fails to deliver the necessary insights. However, since there is always a tradeoff between recall and precision (Zhao et al., 2018), it is of importance to investigate which metric would be of interest for this specific case study. In the end, three different scenarios can be drawn up. One where it is chosen to go for highest precision at the cost of a low recall, one where the highest recall is most beneficial at the cost of a low precision, or the one where the highest F1-score (harmonic mean between precision and recall) is favoured.

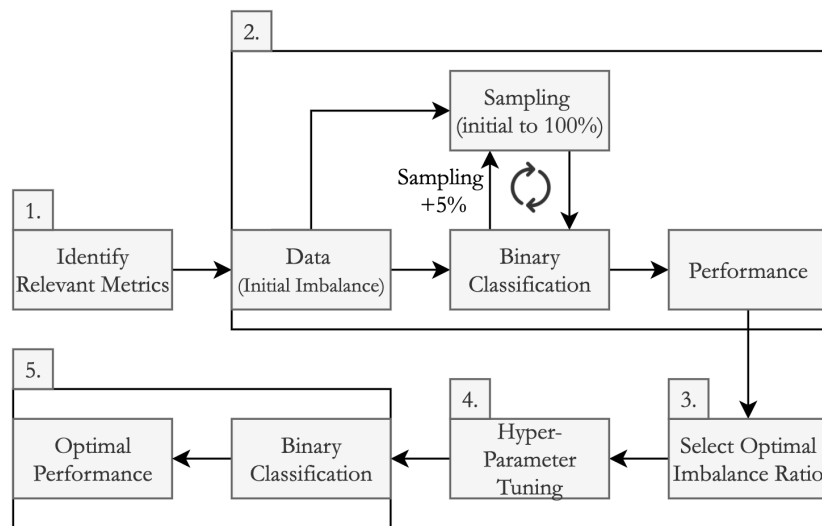


Figure 3: A flow diagram of the systematic approach to deal with imbalanced data.

ARRIVALS / DEPARTURES				
Terminal	Flight	Destination	Time	Status
A	K123	Brussels	12:00	
B	A421	London	12:15	
<u>A</u>	<u>C523</u>	<u>Istanbul</u>	<u>13:00</u>	<u>Cancelled</u>
A	E856	Paris	14:00	
A	R274	Budapest	15:00	
B	K876	Antwerp	17:00	Cancelled
<u>B</u>	<u>E746</u>	<u>Dublin</u>	<u>19:00</u>	<u>Cancelled</u>
A	A998	New York	19:45	
A	D876	Dubai	20:15	
B	D763	Athens	20:45	
A	P387	Vienna	21:00	
B	M818	Sydney	21:45	Cancelled
B	G765	Montreal	22:00	
A	B923	Lisbon	22:10	Cancelled
B	C872	Brussels	23:00	

ARRIVALS / DEPARTURES				
Terminal	Flight	Destination	Time	Status
<u>A</u>	<u>K123</u>	<u>Brussels</u>	<u>12:00</u>	
B	A421	London	12:15	
<u>A</u>	<u>C523</u>	<u>Istanbul</u>	<u>13:00</u>	<u>Cancelled</u>
<u>A</u>	<u>E856</u>	<u>Paris</u>	<u>14:00</u>	
A	R274	Budapest	15:00	
B	K876	Antwerp	17:00	Cancelled
<u>B</u>	<u>E746</u>	<u>Dublin</u>	<u>19:00</u>	<u>Cancelled</u>
A	A998	New York	19:45	
<u>A</u>	<u>D876</u>	<u>Dubai</u>	<u>20:15</u>	
<u>B</u>	<u>D763</u>	<u>Athens</u>	<u>20:45</u>	
A	P387	Vienna	21:00	
<u>B</u>	<u>M818</u>	<u>Sydney</u>	<u>21:45</u>	<u>Cancelled</u>
B	G765	Montreal	22:00	
<u>A</u>	<u>B923</u>	<u>Lisbon</u>	<u>22:10</u>	<u>Cancelled</u>
B	C872	Brussels	23:00	

Figure 4: A visual explanation of high precision and low recall (left) and high recall and low precision (right). This is a fictional flight schedule board, with the actual flight status mentioned on the board, whereas the predicted cancelled flights are in red, underlined italics.

The first two scenarios can be visualised in Figure 4. In the figure two flight schedule boards can be seen, with information for passengers, such as terminal, flight number, destination, time and flight status. The boards can be assumed to be real time information, and five of the flights appear to have been cancelled. The flights in underlined, red italics are the ones the model predicted to be cancelled. So, the difference between the actually cancelled flights (with status ‘cancelled’) and the predicted ones is clearly visible. The left figure visualises a high precision and low recall. In other words, the cancelled flights predicted by the model have a high likelihood of actually being cancelled, at cost of only capturing just a few (2 out of 5). The right figure visualises a high recall and low precision. This means that the model manages to capture most of the cancelled flights, at the cost of wrongly capturing a lot of non-cancelled flights. The third scenario, going for the highest F1-score, would be the optimal combination of the two aforementioned scenarios.

It is decided that the most useful and optimal way of implementation of the model for AAS is to go for the highest precision. With an eye on operations and professional airline relationships, it is better to have a short prediction list with high certainty cancellations, than to have an elaborate, long list with low certainty and therefore lots of wrong predictions. Additionally, it is decided to include the highest F1-score in further analysis as well. Broadening the spectrum of machine learning prediction, other performance indicators, like recall, might be of a higher importance for applications in industries or sectors other than aviation. As the F1-score represents the harmonic mean of both precision and recall, it gives a better overall overview of the model performance. One could say that from a purely mathematical point of view, the F1-score is the most important metric for averaged model performance (especially here, for imbalanced data). Therefore, in order to make this research applicable to other fields of study, it is chosen to also present the optimal F1-score performance. Additionally, it presents the opportunity to develop an approach to deal with the imbalanced data not only utilising precision, but also other performance indicators.

## Step 2: Sample and plot performance evolution

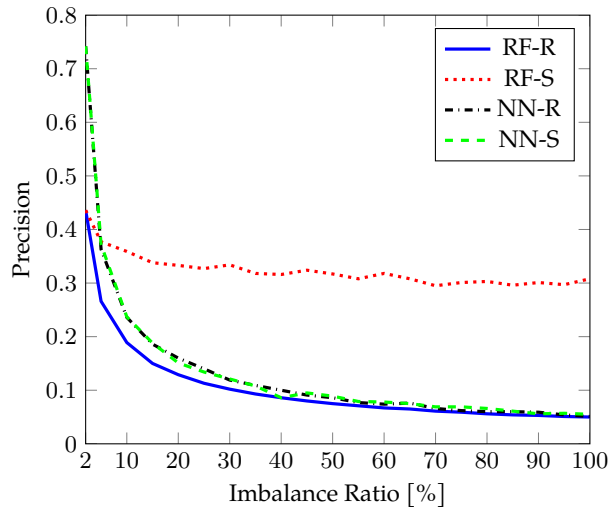
The second step in the approach is to investigate the evolution of the relevant performance indicators, by varying the so called ‘imbalance ratio’. It is defined as the ratio of delayed/cancelled flight to non-delayed/non-cancelled flights. The data is sampled at different degrees, starting at ‘no sampling’ (base imbalance ratio) and going to 100% sampling, with steps of 5% per iteration. So, for example, the data is sampled, in case of cancellations, at 2%, 5%, 10%, 15%, ..., 95% and 100%. After each iteration, the data is fed into the binary classification algorithm and the performance is obtained. The start here is at the base imbalance ratio of 2% since that is the lowest ratio

possible (without sampling). Here, ‘imbalance ratio of 15%’ means that the amount of cancelled flights is 15% of the amount of non-cancelled flights. Consequently, 100% imbalance ratio refers to perfect re-sampling, where the number of delayed/cancelled and non-delayed/non-cancelled flights are equal. The ideal way to visualise the metric evolution is by plotting the performance metric in a graph as a function of the imbalance ratio.

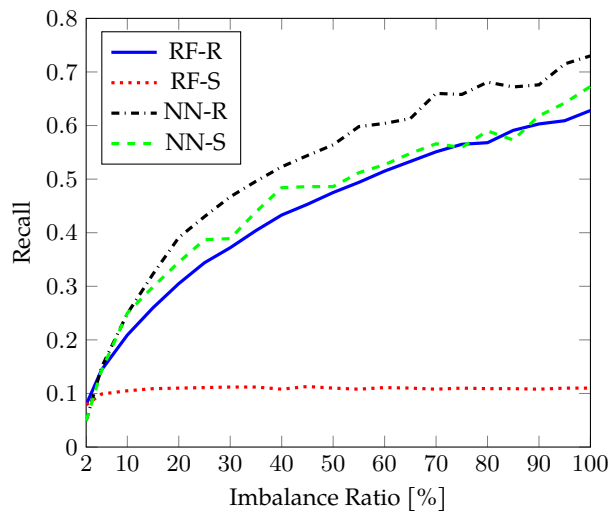
In order to perform the sampling of the data, Chawla et al. (2002) have proposed an interesting technique. It goes by the name of Synthetic Minority Oversampling Technique (SMOTE) and is centered on oversampling the minority class, by creating synthetic samples. SMOTE generalises the decision region of this minority class, by multiplying the difference between a data sample and one of its nearest neighbours with a random number between 0 and 1 and then by adding the result to the sample under consideration. In essence, synthetic samples are created on the lines between the minority samples and their nearest neighbours. The authors also combine their sampling technique with Random Undersampling (RUS). RUS works by randomly removing majority samples from the dataset, in order to bring the minority and majority shares closer together. This technique is also included in this research by means of combination and comparison. Apart from only applying RUS and SMOTE separately on the data, different combinations of SMOTE and RUS are tested, at multiple imbalance ratios.

All three scenarios (cancellations, departure delay and arrival delay) are run with the RF and NN classifiers, for a series of imbalance ratios. The 1 day prediction horizon is used for the flight cancellations, whereas the 1 hour prediction horizon is used for the flight delays. The models are run with the default hyper-parameter settings and hyper-parameter tuning is performed at a later stage. The performance metric graphs for the cancellations can be seen in Figure 5. It respectively displays the precision, recall and F1-score in function of the imbalance ratio for the two algorithms (RF and NN) and two sampling techniques (S for SMOTE and R for RUS). Hence, NN-S means a Neural Network sampled with SMOTE. The minimum on the x-axis is 1.6% (rounded to 2% in the graphs), as this is the base ratio of cancelled flights to non-cancelled flights for unsampled data. Scenarios with sampling techniques which combine SMOTE and RUS at different imbalance ratios are also investigated. However, the results are entirely in line with the results shown in the graphs and changes in the graphs seem to correspond only to the changes in imbalance ratio rather than to the different combination styles of SMOTE and RUS. Therefore, they are not included in these performance graphs.

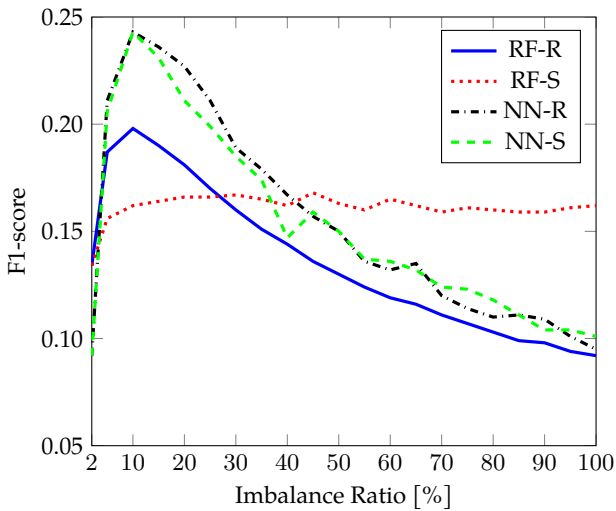
As precision and F1-score are the identified relevant metrics, these graphs are central in the analysis. In Figure 5 it can be observed that the precision score has a maximum for all algorithms at the base imbalance



(a)



(b)



(c)

Figure 5: Precision (a), recall (b) and F1-score (c) in function of imbalance ratio, for cancellation prediction. (RF = Random Forest, NN = Neural Network, R = RUS, S = SMOTE)

ratio without sampling (1.6%), after which it rapidly decreases for the next few imbalance ratios, just for it to start decreasing more gradually with increasing imbalance ratio. The inverse can be seen for the recall, which starts at a minimum and gradually increases with increasing imbalance ratio. Again, the imbalance ratio is defined as the ratio of cancelled flights to non-cancelled flights, which can be increased by means of sampling techniques, which could be SMOTE (creating minority samples) or RUS (removing majority samples). Finally, as the F1-score is dependent both on precision and recall, its course is in line with how high the harmonic mean of the two aforementioned indicators is. Therefore, the peak is observed at the point where the difference between precision and recall is the smallest. One combination of algorithm and sampling technique, however, seems to react rather insensitive to the imbalance ratios for both recall and precision, namely the RF with SMOTE.

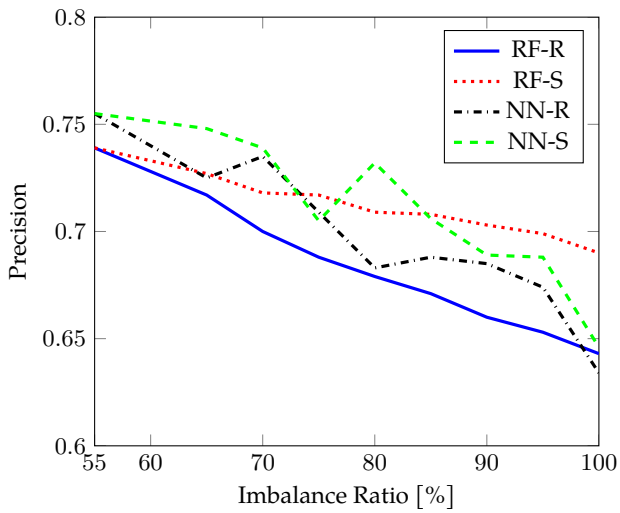
For the departure delays, the points on the performance metric plots range between the point without sampling, which is 55% (i.e. the ratio of delayed flights over the non-delayed flights) and 100% RUS and SMOTE. Also here, precision, recall and F1-score are the centre of the imbalance analysis. Their graphs can be seen in Figure 6. The general graph trends seem to be the same as for the cancellations. Precision decreases with increasing imbalance ratio and recall increases with increasing imbalance ratio, for both algorithms and sampling techniques. The F1-score seems to also gradually increase with the imbalance ratio.

Finally, for the arrival delays, the precision, recall and F1-score graphs can be seen in Figure 7. The minimum on the x-axis, also known as the no sampling point or base imbalance ratio, lies at 33%. Again, there is a clear decreasing trend for precision and an increasing trend for recall, with the F1-score graph corresponding to their harmonic mean.

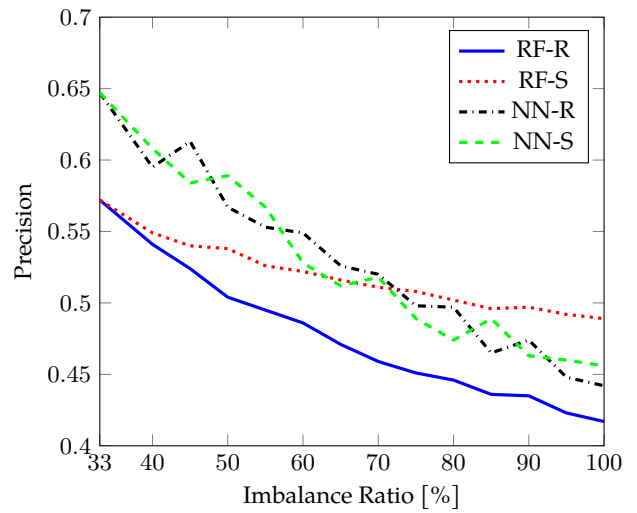
### Step 3: Select optimal imbalance ratio

When the performance metric evolution on the various imbalance ratios is obtained, it is time to select the optimal imbalance ratio from the performance plots. This is done by looking for the scenario in which the relevant performance metric has reached a maximum.

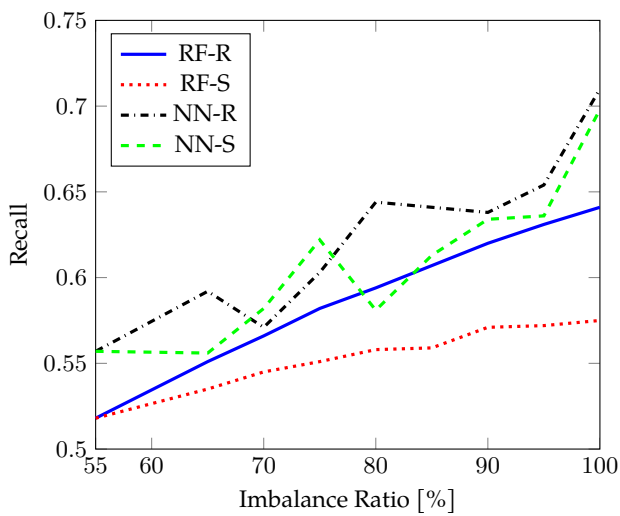
For the cancellations, the maximum precision point for the NN classifier can be found in Figure 5a at the 1.6% ratio, i.e. the point with no sampling. For RF, the exact same observation can be made, i.e. highest precision at the base imbalance ratio, without sampling. The optimal F1-score for NN, lies on the 10% point in Figure 5c, this time sampled with SMOTE. The optimal F1-score for RF, lies on the 10% RUS point. Do bear in mind that these results are for the 1 day prediction horizon and the other horizons are added in a later stage. It can already be observed that for both the precision and F1-score, the NN classifier performs better than the RF.



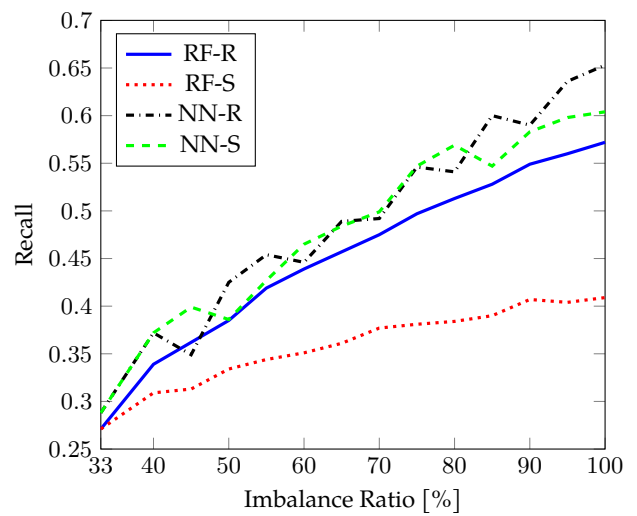
(a)



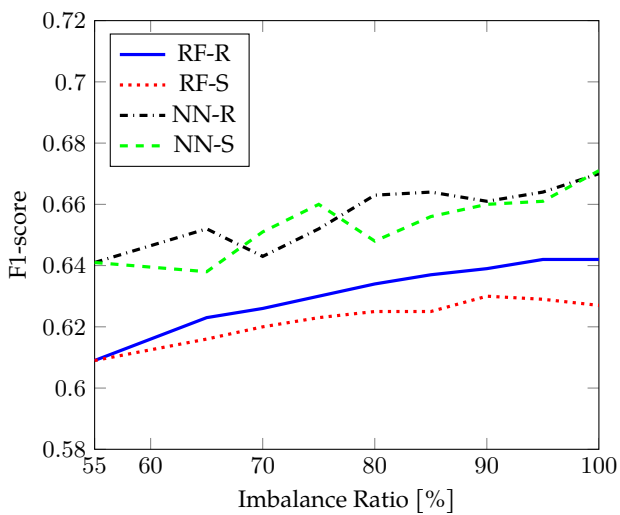
(a)



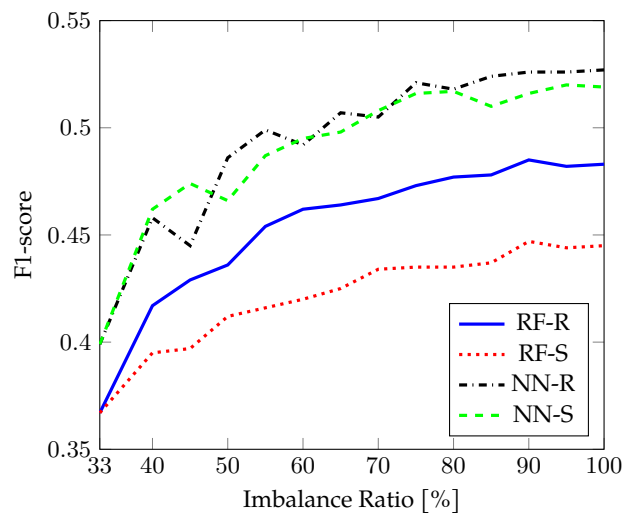
(b)



(b)



(c)



(c)

Figure 6: Precision (a), recall (b) and F1-score (c) in function of imbalance ratio, for departure delay prediction. (RF = Random Forest, NN = Neural Network, R = RUS, S = SMOTE)

Figure 7: Precision (a), recall (b) and F1-score (c) in function of imbalance ratio, for arrival delay prediction. (RF = Random Forest, NN = Neural Network, R = RUS, S = SMOTE)

Just like it was the case for cancellations, it is better to have a shorter list with higher certainty departure delays, compared to the opposite, from an airport operational point of view. Therefore, the points with highest precision and highest F1-score are chosen once more, this time for departure delays. For both classifiers, the maximum precision is observed in Figure 6a at the 55% point, or the no sampling point. For the F1-score, the NN scores best at 100% SMOTE and the RF at 100% RUS, as can be seen in Figure 6c.

For the arrival delays, the highest precision is again found for no sampling (33% imbalance) for both classifiers, as can be seen in Figure 7a. The highest F1-score, is found at 100% RUS for NN and 90% RUS for RF in Figure 7c. A summary of all optimal selected imbalance ratios for each scenario and each algorithm can be seen in Table 7.

#### Step 4: Perform hyper-parameter tuning

Following the selection of the optimal imbalance ratios, hyper-parameter tuning can now be performed for all classifiers, each time on their respective selected imbalance ratios. The binary classification machine learning algorithms contain hyper-parameters, which are comparable with internal tuning buttons. When tuning these parameters, the performance of the algorithm can be optimised for a specific scenario. Therefore, this technique is applied in this research as well. The hyper-parameters under consideration for the RF classifier are the number of trees, selection criterion, maximum tree depth and maximum features per tree. For the NN classifier, they are the hidden layer size, the batch size, activation function, solver and the learning rate.

The hyper-parameter tuning is performed utilising the *RandomizedSearchCV* function from Scikit Learn. This function generates random parameter combinations and uses cross-validation to find the optimal set. The final sets of parameters can be seen in Table 8 for the Neural Network classifier and in Table 9 for the Random Forest classifier. It is possible that, after the tuning, the result is not as good as the default model parameter settings. When this is the case, the default hyper-parameters are chosen as the definitive ones.

#### Step 5: Obtain optimal performance

The final step in this systematic approach is running the RF and NN classifiers again, on the selected imbalance ratios, with their tuned hyper-parameters. Then, the final and optimal results can be generated. For flight cancellations, the results are generated for each of the three prediction horizons, namely 1 day, 1 week and 6 months before the flight and for flight delays, the 1 hour prediction horizon is added. All the results are the mean of a 5-Fold CV.

The final results for the cancellation prediction are visible in Table 10. It can be observed that the precision performance of RF is better than NN for the

no sampling (high precision) scenario. The opposite is observed for the F1-score with the 10% sampling (high F1-score) scenario. Additionally, the results show that the performance is quite steady when it comes to increasing the prediction horizon, except for the F1-score at the 10% sampling, which seems to decrease with increasing prediction horizon. In general, some small fluctuations are observed, however, no large differences are present. These final results also show and confirm very distinctly the effect of choosing two imbalance ratios or sampling points (no sampling vs SMOTE/RUS) and therefore also confirm the need for a systematic imbalance approach to the classification problem. Large differences can be observed when comparing the precision, recall and F1-scores of the two sampling points.

RF yields the highest precision at the 6 months prediction horizon with no sampling. The precision score is 0.892, i.e., of all predicted cancelled flights, 89% is actually cancelled. This, however, corresponds to a very low recall of 0.034, which indicates that about 3% of all actually cancelled flights are effectively predicted as cancelled. This low recall is in fact the case for all of the 'high precision' points. The accuracy score lies very high, at 0.986, however, this is to be expected for non-sampled data with an extreme imbalance. Lastly, AUC seems to show a decent score of 0.811. For the highest F1-score, NN seems to score the best on the 1 day prediction horizon with approximately 0.249. This is caused by a precision of 0.263 and a recall of 0.237. In the same analogy as before, it is observed that about 26% of all predicted cancellations are actually cancelled, combined with the fact that approximately 24% of all actual cancellations are predicted as cancelled. Again, the accuracy score here is very high, at 0.978, since 10% sampling still yields a highly imbalanced dataset. Also here, AUC is decent with 0.854.

The final results for the departure delay prediction are shown in Table 11. For the no sampling scenario, the RF classifier slightly outperforms the NN classifier, especially for precision. For F1-score in the sampled case NN is better on the 1 hour, 1 week and 6 months timings, however the differences are very small and for 1 day RF is best. Also, note that for this scenario there is no single F1-score below 0.5. When looking at the overall performance over all the prediction horizons, it is clear that the 1 hour timing is the best one. This is as expected, since this prediction horizon contains information from the same day, such as arrival delay.

The overall highest precision can be found at the 1 hour prediction horizon with no sampling, with RF just slightly beating NN with a precision of 0.783, which goes together with a recall of 0.511. This implies that about 51% of all actually delayed flights are predicted as delayed and 78% of all predicted delays are actually delayed. Accuracy and AUC values are 0.779 and 0.827 respectively, which are both decent scores, however, less informative of the actual performance due to the imbalance in the data. The highest F1-score

Table 7: The optimal imbalance ratios selected from the best performing points on the performance plots, with the Neural Network (NN) and Random Forest (RF) classifiers.

	Cancellations		Departure Delay		Arrival Delay	
	NN	RF	NN	RF	NN	RF
Highest precision	no sampling	no sampling	no sampling	no sampling	no sampling	no sampling
Highest F1-score	10% SMOTE	10% RUS	100% SMOTE	100% RUS	100% RUS	90% RUS

Table 8: Final Neural Network (NN) hyper-parameters.

	Sampling	Hidden layer size	Batch size	Activation	Solver	Learning rate
Cancellations	no sampling	100 (1 layer)	1000	ReLU	sgd	constant
	10% SMOTE	100 (1 layer)	1000	ReLU	adam	constant
Departure Delay	no sampling	100 (1 layer)	auto	ReLU	adam	constant
	100% SMOTE	100 (1 layer)	auto	ReLU	adam	constant
Arrival Delay	no sampling	100 (1 layer)	1000	logistic	sgd	adaptive
	100% RUS	100 (1 layer)	auto	ReLU	adam	constant

Table 9: Final Random Forest (RF) hyper-parameters.

	Sampling	Number of trees	Criterion	Max depth	Max features
Cancellations	no sampling	100	Entropy	10	0.2
	10% RUS	300	Entropy	6	1.0
Departure Delay	no sampling	500	Gini	8	0.1
	100% RUS	500	Entropy	6	1.0
Arrival Delay	no sampling	100	Gini	6	0.1
	90% RUS	300	Entropy	6	0.7

is observed at the 1 hour prediction horizon for NN with 100% SMOTE. Its value is 0.671, combined with a precision and recall of 0.646 and 0.698 respectively. In other words, about 65% of all predicted delays are actually delayed and approximately 70% of all actually delayed flights are correctly predicted as delayed. Also in this case, accuracy and AUC show decent values of 0.757 and 0.822 respectively.

The final results for the arrival delay prediction are shown in Table 12. For the no sampling scenario, RF seems to be better for precision than NN on the timings of 1 hour, 1 day and 1 week. NN seems to have a

better precision score at the 6 months prediction horizon, compared to RF. For F1-scores in the sampled case (100% & 90% RUS), the NN outperforms the RF classifier on each prediction horizon.

Of all prediction horizons, again, the 1 hour timing is the best one, as expected. The highest precision score is observed at RF with no sampling, namely 0.783, together with a recall of 0.052. In other words, of all predicted arrival delays, 78% are actually delayed, while only 5% of all the actual delays are captured by the model. The highest F1-score is for NN with 100% RUS, namely 0.531. Taking the precision and recall into ac-

Table 10: The final performance indicator results for cancellation prediction, for each prediction horizon.

	Indicator	1 day		1 week		6 months	
		NN	RF	NN	RF	NN	RF
no sampling	Accuracy	0.986	0.986	0.986	0.986	0.986	0.986
	<b>Precision</b>	<b>0.809</b>	<b>0.853</b>	<b>0.765</b>	<b>0.861</b>	<b>0.876</b>	<b>0.892</b>
	Recall	0.041	0.035	0.043	0.035	0.036	0.034
	F1-score	0.079	0.068	0.082	0.068	0.070	0.066
10% SMOTE (NN) / 10% RUS (RF)	AUC	0.772	0.850	0.788	0.853	0.746	0.811
	Accuracy	0.978	0.981	0.976	0.976	0.979	0.984
	Precision	0.263	0.284	0.232	0.210	0.251	0.349
	Recall	0.237	0.198	0.264	0.204	0.199	0.060
	<b>F1-score</b>	<b>0.249</b>	<b>0.233</b>	<b>0.247</b>	<b>0.207</b>	<b>0.222</b>	<b>0.103</b>
	AUC	0.854	0.839	0.861	0.836	0.811	0.797

Table 11: The final performance indicator results for departure delay prediction, for each prediction horizon.

	Indicator	1 hour		1 day		1 week		6 months	
		NN	RF	NN	RF	NN	RF	NN	RF
no sampling	Accuracy	0.780	0.779	0.682	0.681	0.667	0.672	0.657	0.671
	<b>Precision</b>	<b>0.766</b>	<b>0.783</b>	<b>0.614</b>	<b>0.660</b>	<b>0.585</b>	<b>0.667</b>	<b>0.612</b>	<b>0.649</b>
	Recall	0.539	0.511	0.303	0.203	0.520	0.260	0.460	0.274
	F1-score	0.633	0.619	0.406	0.311	0.551	0.374	0.526	0.385
	AUC	0.826	0.827	0.691	0.691	0.705	0.706	0.692	0.690
100% SMOTE (NN) /	Accuracy	0.757	0.741	0.666	0.645	0.639	0.649	0.639	0.641
100% RUS (RF)	Precision	0.646	0.623	0.524	0.493	0.552	0.529	0.553	0.515
	Recall	0.698	0.648	0.491	0.601	0.588	0.614	0.604	0.606
	<b>F1-score</b>	<b>0.671</b>	<b>0.648</b>	<b>0.507</b>	<b>0.542</b>	<b>0.570</b>	<b>0.568</b>	<b>0.577</b>	<b>0.557</b>
	AUC	0.822	0.798	0.679	0.685	0.698	0.700	0.696	0.685

Table 12: The final performance indicator results for arrival delay prediction, for each prediction horizon.

	Indicator	1 hour		1 day		1 week		6 months	
		NN	RF	NN	RF	NN	RF	NN	RF
no sampling	Accuracy	0.774	0.770	0.768	0.765	0.751	0.747	0.749	0.745
	<b>Precision</b>	<b>0.720</b>	<b>0.783</b>	<b>0.692</b>	<b>0.713</b>	<b>0.723</b>	<b>0.724</b>	<b>0.726</b>	<b>0.685</b>
	Recall	0.093	0.052	0.054	0.028	0.076	0.050	0.057	0.037
	F1-score	0.165	0.097	0.101	0.054	0.138	0.094	0.106	0.070
	AUC	0.720	0.743	0.680	0.693	0.702	0.718	0.686	0.686
100% RUS (NN) /	Accuracy	0.737	0.730	0.710	0.640	0.650	0.695	0.629	0.678
90% RUS (RF)	Precision	0.457	0.444	0.406	0.362	0.409	0.443	0.393	0.412
	Recall	0.632	0.620	0.528	0.624	0.685	0.582	0.641	0.522
	<b>F1-score</b>	<b>0.531</b>	<b>0.518</b>	<b>0.459</b>	<b>0.458</b>	<b>0.512</b>	<b>0.503</b>	<b>0.487</b>	<b>0.461</b>
	AUC	0.767	0.757	0.712	0.700	0.727	0.732	0.710	0.698

count, it can be said that 46% of all predicted arrival delays are actually delayed and 63% of all actual delays have been captured.

## 4. Discussion

### 4.1. Key findings

First of all, the general performance of the departure and arrival delay models is better than the performance of the cancellation prediction model. Precision and recall do not lie as far apart as they do there and the overall F1-scores are a lot better. For departure delay, there is not even a single time the F1-score lies beneath 0.5 in the sampled case with SMOTE and RUS. It is likely that the reasons for these observations lie in the original imbalance ratio of the datasets. Recall that the base imbalance ratio of cancellations is about 2% and of the departure and arrival delays respectively 55% and 33%. The effect of the severely imbalanced cancellation data reflects in the results, just like the medium imbalance influences the results of the delays. Also note that the imbalance ratio of the departure delays is higher compared to the arrival delays, which explains why the departure delay performance is slightly better than the arrival delay. This logic also explains why the accuracy and AUC are a little lower

for the delays, compared to the cancellations. Imagine that the classifier classifies all flights as not delayed or not cancelled. For the cancellations, the accuracy would automatically be 98%, as only 2% is cancelled. As the base imbalance ratio of the delays lies higher, this accuracy will naturally be a bit lower. This is also the reason why accuracy is not the best performance metric when analysing highly imbalanced data.

A second point of discussion, is the improvement of certain performance metrics when performing the imbalance analysis of the machine learning algorithms at different imbalance ratios. It is observed that precision and recall respectively decrease and increase with increasing imbalance ratio. Precision is the main metric of interest for this case study, but since there is a decrease in score with increasing imbalance ratio, it can be said that balancing with sampling techniques does not positively influence the precision. However, when looking at the other relevant metric, F1-score, a clear improvement can be observed when increasing the imbalance ratio. For cancellation predictions, the F1-score increases with 158%, from 0.092 at the no sampling point to 0.237 for 10% SMOTE, with the NN classifier. For departure delays, the F1-score of the NN



classifier increases with 5%, from 0.641 to 0.671. Finally, for the arrival delay the F1-score increases with 32%, from 0.399 to 0.572, also with the NN classifier. Observe here that each time the most increase towards the highest F1-score is seen with the NN classifier. In combination with the fact that in the final results NN almost always outperforms RF in terms of F1-score, it can be concluded that for F1-score NN is the optimal machine learning algorithm to use.

Subsequently, the inclusion of the 1 hour prediction horizon for the delay predictions and its implications on the model performance require some discussion. This topic is especially interesting when comparing the arrival delay with the departure delay performance. This short prediction horizon enables information of that same day (actual features) to be included. The departure delay in particular, has an advantage over arrival delay, which was already visible in Tables 4 and 5. There, it can be seen that in the departure delay case some features, like the actual arrival delay are present, which is not applicable for the prediction of the arrival delay. When observing the recall of the no sampling scenario for the arrival delay model, it can be noted that it is quite low, just like it is for the cancellation predictions. Apart from the fact that more imbalance is present, compared to the departure delay, this can be explained by the fact that there is no such feature as the 'arrival delay' feature that comes with the departure delay for the 1 hour prediction horizon. This feature holds a high predictive power and, when left out, it results in lower performance. Now, one can think of what prediction horizon could be the most useful for airport operations. Since most scheduling tasks are done a day to a week in advance, these two timings have most importance. When predicting at these time horizons, there is still time to take significant actions to mitigate the effect of the delay or cancellation. Still, a 1 hour and 6 months prediction horizon prove valuable, since they could provide insights at, firstly, very short moments in the future if quick actions are an option and, secondly, at a slot allocation level.

Two main reasons are given for the difference in performance between the departures and arrivals, namely the imbalance and the inclusion of some features on the 1 hour prediction horizon. Both can simultaneously contribute to the performance difference, however they can also be seen separately, when just comparing the 1 day, 1 week and 6 months time horizon, since the specific set of actual features is not present there. In that situation, the imbalance difference is mostly responsible for the performance difference.

Finally, a systematic approach to deal with these highly imbalanced datasets is established. In this particular case study, in order to obtain highest precision, it is best not to sample the data. For highest recall, sampling at high imbalance ratios is optimal. For F1-score, it is necessary to analyse the metric evolution

with increasing imbalance ratios to find the optimal point, since it depends on both precision and recall.

## 4.2. Future directions and recommendations

As this research is limited within the time-frame of a thesis project, there is still room for improvement and a lot can be done in the field of study. One particular element is the weather data incorporated in the research. A first suggestion would be to use an actual weather forecast, in order to predict the weather on different time horizons. In this paper, the actual weather is assumed to be an accurate forecast for 1 hour and 1 day prediction horizons and it is averaged to introduce forecast uncertainty for 1 week and 6 months. An actual weather forecast could be beneficial for the model as the assumptions made might influence how well this research reflects real-life scenarios.

Additionally, a pre-processing step is made in order to prepare the data to be fed into the machine learning algorithms, or classifiers. As stated earlier, feature scaling is performed on all features. This step is essential for some classifiers, but not for all. However, in order to keep the research within certain workable limits, it was decided to perform the scaling step on all data, for all classifiers. The fact that some classifiers do need this measure, does not necessarily mean that it negatively affects the performance when utilising it with classifiers that do not need it. Therefore, it is suggested that taking a more detailed look into which algorithms really need this pre-processing step, can be a move in the right direction for future analysis.

Finally, another suggestion, as only two sampling techniques were included in the research, would be to perform the imbalance analysis with other sampling techniques, such as Random Oversampling, which could lead to new insights. The same applies to the machine learning algorithms, as different algorithms and sampling techniques could react very differently. Examples of different algorithms are k-Nearest Neighbours or boosting algorithms.

## 5. Conclusion

In this research, a systematic approach to deal with highly imbalanced data for machine learning algorithms is developed, in order to successfully perform the binary classification of flight delays and cancellations. The machine learning algorithms are trained and tested with flight operational data from Amsterdam Airport Schiphol and weather data from KNMI and METAR. The predictions are evaluated on prediction horizons of 1 hour, 1 day, 1 week and 6 months before the flight. The imbalance of the data is mitigated by investigating the effects of the sampling techniques Random Undersampling (RUS) and Synthetic Minority Oversampling Technique (SMOTE) on the classifier performance.

The imbalance analysis and its results show that optimal performance scores can be obtained by investigat-

ing indicator evolution with varying data imbalance ratios. When the optimal imbalance ratios are found, the model can be tuned in order to even further enhance the indicator score. The most essential performance indicators for this case study are precision and F1-score. Optimal precision is shown to be found at base imbalance ratios (data without sampling), for all algorithm and sampling technique combinations. In order to find the highest F1-score, sampling is shown to be essential.

For flight cancellations, the highest precision is observed at the 6 months prediction horizon, with the Random Forest classifier without sampling (base imbalance ratio). The highest F1-score is found with the Neural Network classifier at 10% SMOTE, on the 1 day horizon. In the case of the flight departure delays, the highest precision is detected at the 1 hour prediction horizon, with the Random Forest classifier without sampling (base imbalance ratio). The highest F1-score is observed with the Neural Network classifier at 100% SMOTE, on the 1 hour horizon. Finally, for the flight arrival delays, the highest precision is encountered with the Random Forest classifier without sampling (base imbalance ratio), on the 1 hour prediction horizon. The optimal F1-score is observed with Neural Network at 1 hour before the flight, sampled at 100% RUS.

The proposed imbalance approach and prediction models could be a base framework for multiple on-time performance predictions for major European hub-airports. As the F1-score was also included in the analysis, the spectrum of applications can be broadened to other fields of study, where perhaps performance metrics other than precision could be of interest. The research could be elaborated by adding more machine learning algorithms, such as k-Nearest Neighbours or boosting algorithms, and taking into account other types of sampling techniques, such as Random Oversampling.

## 6. References

- Alderighi, M. and Gaggero, A. A. (2018). Flight cancellations and airline alliances: Empirical evidence from europe. *Transportation Research Part E: Logistics and Transportation Review*, 116:90–101.
- Alonso, H. and Loureiro, A. (2015). Predicting flight departure delay at porto airport: A preliminary study. In *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, volume 3, pages 93–98. IEEE.
- Cao, J.-M. and Kanafani, A. (1997). Real-time decision support for integration of airline flight cancellations and delays part i: mathematical formulation. *Transportation Planning and Technology*, 20:3:183–199.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.
- Chen, J. and Li, M. (2019). Chained predictions of flight delay using machine learning. In *AIAA Scitech 2019 Forum*, page 1661.
- Choi, S., Kim, Y. J., Briceno, S., and Mavris, D. (2016). Prediction of weather-induced airline delays based on machine learning algorithms. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–6. IEEE.
- Eurocontrol. (2018a). European aviation in 2040, challenges of growth.
- Eurocontrol. (2018b). Network manager annual report.
- Eurocontrol. (2018c). Network operations report 2018.
- Flach, P. (2012). *Machine Learning*. Cambridge University Press New York.
- Gao, K., Khoshgoftaar, T. M., and Napolitano, A. (2015). Combining feature subset selection and data sampling for coping with highly imbalanced software data. In *SEKE*, pages 439–444.
- Hassanzadeh, H., Groza, T., Nguyen, A., and Hunter, J. (2014). Load balancing for imbalanced data sets: classifying scientific artefacts for evidence based medicine. *Pricai 2014: Trends in Artificial Intelligence*, 8862:972–984.
- Horiguchi, Y., Baba, Y., Kashima, H., Suzuki, M., Kayahara, H., and Maeno, J. (2017). Predicting fuel consumption and flight delays for low-cost airlines. In *Twenty-Ninth IAAI Conference*, pages 4686–4693.
- IowaStateUniversity. (2020). Asos-awos-metar data download.
- Jarrah, A. I. Z., Yu, G., Krishnamurthy, N., and Rakshit, A. (1993). A decision support framework for airline flight cancellations and delays. *Transportation Science*, 27(3):266–280.
- Kalliguddi, A. M. and Leboulluec, A. K. (2017). Predictive modeling of aircraft flight delay. *Universal Journal of Management*, 5(10):485–491.
- Kim, Y. J., Choi, S., Briceno, S., and Mavris, D. (2016). A deep learning approach to flight delay prediction. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–6. IEEE.
- KNMI. (2020). Uurgegevens van het weer in nederland.
- Kuhn, N. and Jamadagni, N. (2017). Application of machine learning algorithms to predict flight arrival delays. *CS229 Autumn 2017*.
- Lambelho, M., Mitici, M., Pickup, S., and Marsden, A. (2020). Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions. *Journal of Air Transport Management*, 82:101737.
- Manna, S., Biswas, S., Kundu, R., Rakshit, S., Gupta, P., and Barman, S. (2017). A statistical approach to predict flight delay using gradient boosted decision tree. In *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–5. IEEE.
- Pechenizkiy, M. (2005). The impact of feature extrac-

- tion on the performance of a classifier: knn, naïve bayes and c4. 5. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 268–279. Springer.
- Seelhorst, M. (2014). *Flight Cancellation Behavior and Aviation System Performance*. Ph.D. thesis, UC Berkeley.
- Sternberg, A., Soares, J., Carvalho, D., and Ogasawara, E. (2017). A review on flight delay prediction. *arXiv preprint arXiv:1703.06118*.
- Zhao, Y., Wong, Z. S., and Tsui, K. L. (2018). A framework of rebalancing imbalanced healthcare data for rare events’ classification: a case of look-alike sound-alike mix-up incident detection. *Journal of Healthcare Engineering*, 2018:11.



# II

Literature Study  
(Previously graded under AE4020)



# Abstract

This report represents the literature review for the Air Transport Operations Master Thesis, AE4020. The review is centered around flight cancellation prediction using machine learning algorithms. Cancellations may have negative impacts on airport operations, so predicting them would be beneficial for airports, hence the relevance of this research. The purpose of this report is to ensemble the knowledge already gained in topical literature and to compare and discuss the different methods and techniques used. Finally, a scope, research question and objective can be formulated and the thesis work can actually begin.

The first interesting piece of information gathered is the methodological flow in the literature. They all seem to have roughly the same structure and approach. This approach serves as a backbone structure and layout for the literature study as well. The papers are mostly structured as follows; analysis of data from trustworthy sources, pre-processing, sampling and feature selection. This first part can be taken together as 'data management'. Then there is the assemblance of final training data, with incorporation of eventual specific new features. This is translated to 'cancellation behaviour'. Finally, there is the big chunk of 'machine learning', that deals with training, performance evaluation, testing and the different algorithms.

The first part, data management, is very important in the research, since machine learning algorithms need clearly structured data. Two main types are addressed, namely flight schedule and weather data. For the flight schedule data, sources mostly are airports, airlines or governmental instances. For the weather, this can be national weather services or online service providers. The next steps in the data management are cleaning and normalisation. Missing data can either be interpolated or removed and data can be normalised to e.g. a 0-1 scale, in order to eliminate misleading feature importance due to larger numerical values. Naturally, the datasets do often contain categorical data, which needs to be encoded to form numerical data. Different techniques are used in literature, namely One-Hot encoding, Binary Encoding, Target encoding and Ordinal encoding. Also, trigonometric functions are used to encode periodic data. Furthermore, since flight delay or cancellation data is often imbalanced, sampling needs to be performed to balance out the data for the machine learning algorithms. The following techniques are often used in literature; synthetic minority oversampling, random majority undersampling and minority oversampling with replacement. Finally, the two most popular feature selection techniques are recursive feature elimination and the Pearson correlation coefficient.

Secondly, several features will need to be created to form specific flight cancellation behavioural features. Different aspects or determinants are seemingly important in strategic cancellations within airlines. If an airline is in an alliance, flights are more likely to be cancelled. Also, flights are less likely to be cancelled on competitive routes. When operating from a hub, cancellation rates are also often lower. Subsequently, passenger inconvenience is an important factor, as airlines often try to minimise it by not cancelling flights on infrequently flown routes, heavily loaded flights or final flights of the day. Lastly, flights are less likely to be cancelled on routes with a high average revenue.

The third big 'chunk' is machine learning. In this case, supervised learning algorithms are of importance for this research. A distinction can be made between classification and regression algorithms, the former predicting a certain class of a limited set, the latter predicting a continuous value. The algorithms first need historical data for training, after which they can be tested on unseen data. These train-test splits often vary in literature between 70/30 and 90/10. Also, K-fold Cross Validation is a popular way to go, in which iteratively another set of the K% set of the data is used for testing. It is also noted that sometimes authors train and test with different prediction timings. Then, after training and testing, performance must be evaluated. Popular metrics for evaluation in classification are the confusion matrix, with corresponding accuracy, precision and recall, and the Area Under the ROC curve. For regression, error metrics like mean absolute error or root mean squared error are appropriate. Finally, some frequently used algorithms in literature are Decision Trees, Random Forests, Neural Networks, k-Nearest-Neighbours, Logistic Regression and Boosting algorithms.

Finally, after having gained all the knowledge from the state-of-the-art, the knowledge gap becomes clear, namely that, to the best of knowledge, flight cancellations predictions with machine learning, historical flight and weather data and comparing different prediction timings is an uncharted field. Flight delay prediction was the main topic of research on all-but-one of the sources. It becomes evident that the industry needs more thorough research on flight cancellation predictions. Therefore, the scope of the research is the prediction

of individual flight cancellations in a European airport with machine learning algorithms, using flight and weather data and a prediction horizon of hours to days before the flight. The research question that is meant to be answered by this research is:

*Which machine learning algorithm, trained with historical flight schedule and weather data, produces accurate flight cancellation predictions, on prediction horizons of several hours to days before the flight?*

The research objective is closely following the research question and is:

*To develop a machine learning algorithm that can predict flight cancellations using several prediction horizons from hours to days before the flight.*

The proposed methodology for the thesis is as follows. Flight schedule data originates from Amsterdam Airport Schiphol and weather data from KNMI. A small part of the historical data will serve as unseen or forecast data, averaging the weather to account for forecast uncertainty. Target encoding, 0-1 scale normalisation, SMOTE and RMU and Pearson's correlation coefficient will be used for data pre-processing. Also, the cancellation behaviour features will be incorporated in the data. The pre-processing order will be first sampling, then feature selection, after which the un-sampled data will be used for training. 10-Fold Cross Validation will serve as training method and evaluation. Logistic Regression, Random Forests and k-Nearest-Neighbours will serve as the training algorithms. Different prediction timings are included in the model, ranging between 1 hour, 1 day and 10 days before the flight. The confusion matrix and Area Under the ROC Curve will be used as performance evaluators. The thesis itself is predicted to last approximately 6 months and will probably finish somewhere in the end of August 2020 with the thesis defence.



# Introduction

The aviation industry is immensely competitive and, nowadays, on-time performance is a very, if not the most important measure for an airline's service quality. Regularly, new airlines are created and existing airlines keep developing and expanding. In 2018, more than 11 million flights were operated in Europe alone and, compared to 2017, the annual traffic has increased by 3.8% (Eurocontrol, 2018a). Due to the continuous increment in air-traffic and demand, airspace and airports are getting more and more crowded, operating at maximum capacity without being able to correspondingly increase it, leading to challenging flow situations and even a forecast capacity gap of 1.5 million flights or 8% of the demand by 2040 (Eurocontrol, 2018b). Consider factors such as bad weather and, eventually, flight delays and cancellations become inevitable. These interruptions can have detrimental effects on the airline's quality record and can become a very costly obstacle (Alderighi and Gaggero, 2018). Therefore, it could be of great value for airlines, airports and travellers if there existed a way to accurately predict flight delays and cancellations.

How does one exactly predict these flight statuses? Unfortunately, there is no single optimal way for carrying out all prediction problems. However, in the last decade, there is one technique that has gained a lot of momentum within data science and especially within prediction research (Sternberg et al., 2017). It is called machine learning and it provides a powerful way to make predictions based on what it learned from (past) data. Numerous algorithms exist, however, the *no free lunch theorem* states that there is no single one that outperforms others when testing over all possible problems (Flach, 2012). Hence, it is purposeful to evaluate the performance of different algorithms and see which yields the highest prediction performance.

This paper will represent the literature review for a research addressing the prediction of flight cancellations using machine learning algorithms. Up until now, flight cancellations have been given a lot less attention than flight delays in research. Hence, the investigation of this topic can help uncover new insights that could eventually benefit the industry. The purpose of this review is to ensemble all the knowledge gained during a period of thorough reading of scientific papers, books and articles on the state-of-the-art related to the subject of the research as stated above. The different methods and techniques used in literature will be compared, contrasted and discussed. The aim is to provide the necessary expertise and methodological insights in order to successfully start the thesis, after having created a structured research framework with a clear objective, research questions and well-defined scope.

The review will kick-off with a short chapter, chapter 2, which is dedicated towards the general methodological flow of work, found in most of the topical literature. Afterwards, there is a chapter involving data management, chapter 3. Data pre-processing, analysis and manipulation will be a large part of the research, as machine learning algorithms in particular need clearly structured data to be able to perform efficiently. Next in line is chapter 4, covering all details related to flight cancellation behaviour. Literature that defines the determinants of and reasons for cancellations will be looked into and it will be made clear that new data features, focused around cancellations, will have to be created. Subsequently, chapter 5 will deal with the machine learning algorithms themselves and with the performance evaluators used to analyse and compare their behaviour. Finally, a clear research approach will be formulated in chapter 6. Here, the knowledge gap will be established, together with the scope, research questions, objectives and the research planning.



# 2

## Methodology

This relatively short chapter serves as a kind of summary of the methodologies commonly used in literature treating flight delay or cancellation predicting using machine learning. A large body of knowledge already exists and has been carefully moulded into numerous articles and papers. They are all unique, however, they share a largely overlapping flow of work, or general steps. After reading about 20 papers that researched the topic stated above, the flow diagram in Figure 2.1 was established.

The diagram represents the different steps in the methodology of tackling the general prediction problem and it also provides a general basis for the layout of this literature review. Chapter 3 on Data Management will cover the blocks of Data (source), pre-processing, sampling and feature elimination. Then, somewhat connecting to these data steps and the assembly of the final data block, there is chapter 4 on cancellation behavior, where a closer look will be taken at features that are highly important for cancellations, but not yet present in the data. Hence, feature creation is central there. Afterwards, there is chapter 5 about Machine Learning, covering literature that is related to the blocks of the train-test split, the algorithms and performance evaluators. This layout will also likely be the basis for the flow and subsections of the final thesis paper that will be written after this literature review, especially since this diagram was derived from summarising and generalising the layout and sections of multiple topical papers.

As an example, (Choi et al., 2016) researches the prediction of weather-induced airline delays. It starts with data collection, followed by data pre-processing. Then the classification algorithms are explained, after which the results of the experiments and performance of the models are evaluated. This makes up the body of the paper, which was preceded by an introduction with relevant literature and a methodology and followed by a conclusion.

Furthermore, (Kuhn and Jamadagni, 2017) starts with its dataset and features, after which machine learning models are applied. Subsequently, there are the results and discussion. Also here there is an introduction and related work before the main body and a conclusion after.

The aforementioned papers are only two examples of the large body of topical literature that has been covered for this literature review. The general methodological flow established in this chapter, pointed out in the diagram, has been derived from all of these. However, referencing to each and one of them in this chapter alone would make for an extensive list of repetitive explanations references, whereas there is a clear bibliography section at the end of this review. So, for more examples of this methodologies, please consult some of the other references.

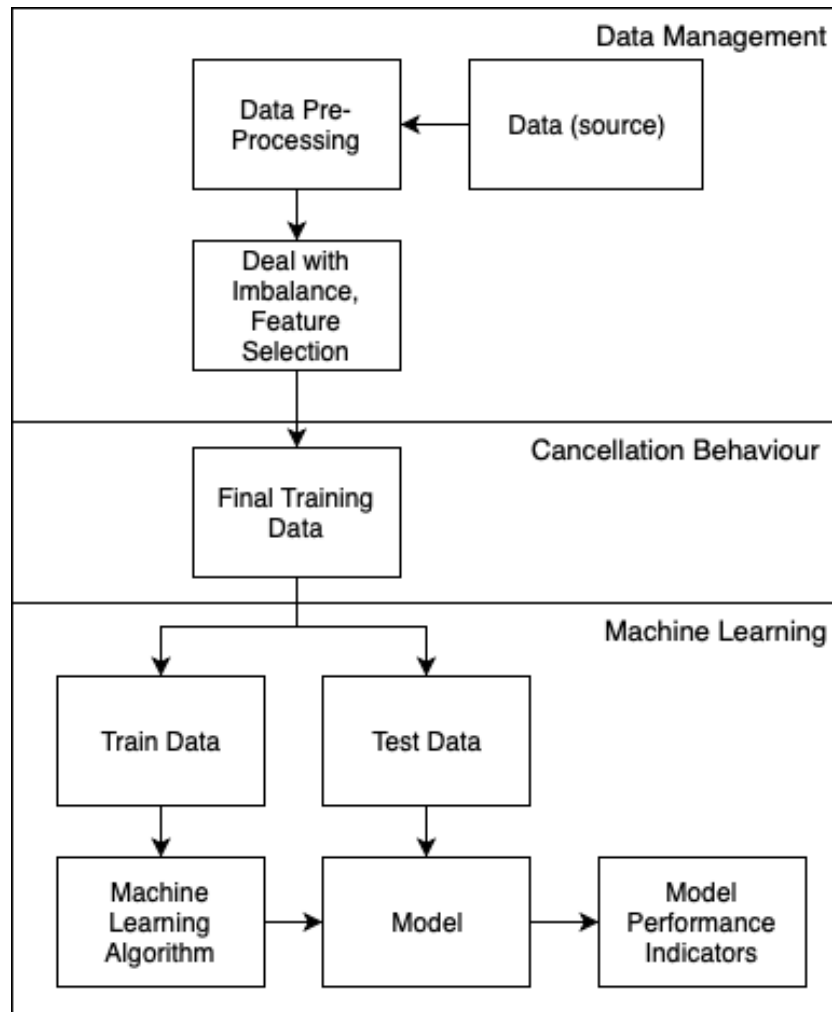


Figure 2.1: General methodological flow and layout in topical literature about flight delay and cancellation prediction, linked to the structure of this literature review.

# 3

## Data Management

A large part of the research consists of data analysis and pre-processing. Therefore, this chapter is dedicated to review the data, methods and techniques that are currently used in research closely related to the research topic. The chapter will start off with some notions about the data sources and data features, after which data pre-processing will be discussed. This incorporates data cleaning and encoding. Next up, methods to deal with imbalanced data are addressed and finally a review on feature selection methods will wrap up this chapter.

### 3.1. Sources & Features

One of the first questions that one should ask in research that involves large databases, is where and how to get the required data. The two types of data needed for the thesis are, as stated below, flight data and weather data. This section will present the main findings concerning the data sources and it will also mention and discuss the types of features used.

#### 3.1.1. Flight Data

The flight schedule data is the most evident data type that is used in this kind of research. Predictions are most likely to be performed on flight schedules, as these types of data are mostly fixed several months in advance, due to slot allocations (Lambelho et al., 2020). Also, this type of data has often been collected for a long time, providing a solid historical base-set, which is essential to obtain a model with a good prediction performance. This type of data mostly comes from and is centered around airports or airlines themselves, or originates from government instances. A summary of references that used flight schedule data can be found in Table 3.1.

In the United States of America (USA), there is the Bureau of Transportation Statistics (BTS), which is a highly popular source for air traffic performance data. It has been used in a lot of flight delay prediction papers, such as (Chen and Li, 2019), (Kim et al., 2016), (Choi et al., 2016) and (Kuhn and Jamadagni, 2017). Furthermore, (Sternberg et al., 2017) states that databases from the Federal Aviation Administration (FAA) are also generally used for the USA, like (Chen and Li, 2019) who combined the BTS source with FAA's Aircraft System Performance Metrics (ASPM) source, and that the Eurocontrol database is a common source for Europe. In (Horiguchi et al., 2017), delay and fuel consumption prediction is performed for low-cost airlines and a combination of flight and passenger data is obtained from a low-cost airline named Peach Aviation. Data provided by Egyptair is used in for flight delay research in (Al-Tabbakh et al., 2018). These two papers provide interesting cases in which machine learning algorithms are fed with data provided by airlines. In terms of data being provided by airports, there is (Lambelho et al., 2020) that uses data from London Heathrow Airport, and (Alonso and Loureiro, 2015), using flight schedule data from Porto Airport.

It should be noted that airport data can provide a more general view over the flight schedule, whereas airline data offers a database with a more company-based point of view, which is less generalisable towards other airlines. However, they do offer passenger and reservation information, which could reveal otherwise unknown patterns leading to cancellations.

Regarding the data features used, there is a lot of recurrence in the feature choice. Information about the date and time (Year, Month, Day, Season, Hour, Quarter), flight, aircraft and airline information and airport

information are almost always present. Other time related features such as taxi-in time or wheels-on time are sometimes present. Also, for relatively short prediction horizons, departure or arrival delay is also popular. Flight distance (the length of the flight route) and Schengen/International can also be useful information. Seats or load factor (how many seats are taken on the aircraft) seems useful when specific airline information is available.

Table 3.1: Summary of data sources and features for flight schedule data used in topical literature, with their target variable and prediction horizon.

Reference	Source	Example Features	Target	Prediction Horizon
(Chen and Li, 2019)	BTS & ASPM	Day of Month and Week, Departure/Arrival Time, Departure Delay Group, Arrival Delay Group, Scheduled Departures/Arrivals	Flight Delay (Classification)	Several Hours
(Kim et al., 2016)	BTS	Day of Week, Season, Month, Data, Origin/Destination Airport, Departure/Arrival Time, Origin/Destination Airport Delay	Flight Delay (Classification)	Several Hours
(Choi et al., 2016)	BTS	Quarter of Year, Month, Day of Week/Month, Departure/Arrival Time, Delay Indicator,	Airline Delay (Classification)	5 Days, 1 Day, 0 Days
(Kuhn and Jamadagni, 2017)	BTS	Date, Day of Week, Flight No., Tail no, Origin/Destination Airport, Departure/Arrival Time, Delay, Flight Time, Delay, Cancellations, Taxi-in, Wheels-on	Flight Arrival Delay (Classification)	Several Hours
(Horiguchi et al., 2017)	Peach Aviation	Year, Month, Day of Week/Month/Year, Departure/Arrival Time, Origin/Destination Airport, Airframe ID, Domestic/International Flight, No of Pax, Pax Gender, Pax Age	Flight Delay (Classification)	5 Months, 1 Week, 1 Day
(Al-Tabbakh et al., 2018)	Egypt Air	Date, Flight No., Origin/Destination City, Departure/Arrival Time, Aircraft Type, Aircraft Registration	Flight Delay (Classification)	Several Hours
(Lambelho et al., 2020)	London Heathrow	Airline, Terminal, Aircraft, Distance, Airport, Country, Seats, Year, Month, Hour, Day of Week/Month/Year, Arrival Delay	Flight Delay and Cancellations (Classification)	6 Months
(Alonso and Loureiro, 2015)	Porto Airport	Arrival Delay, Origin/Destination, Day of Week, Hour, Date, Month, Meteorological Conditions, Airline, Aircraft Type, Parking Stand, Ground Operation Time	Flight Delay (Classification)	Several Hours

### 3.1.2. Weather Data

As is apparent from research done in (Rupp and Holmes, 2006), (Seelhorst, 2014) and (Alderighi and Gaggero, 2018), weather is one of the most decisive factors influencing flight cancellations. Combining planned flight schedules with weather forecasts on different time horizons, could yield more accurate predictions. Just like it was done for the flight data, a summary with sources and features is present in Table 3.2.

A model was built in (Klein et al., 2010), for predicting airport delay using multiple weather forecast prod-

ucts. For this they used the toolset and metric of the Weather Impacted Traffic Index (WITI), which quantifies the impact of traffic demand and weather on the national airspace system. The WITI model uses multiple weather sources, namely National Convective Weather Diagnostic (NCWD) for determining en-route weather and Meteorological Aerodrome Reports (METAR), for weather at the airports. The twin model of WITI is the forecast WITI, or WITI-FA, which uses Localized Aviation MOS Product (LAMP), Collaborative Convective Forecast Product (CCFP) and Corridor Integrated Weather System (CIWS). Furthermore, the National Oceanic and Atmospheric Administration (NOAA)'s weather database is used in the research of (Choi et al., 2016), aiming to predict flight delays combined with weather influence as well. NOAA was used for training machine learning algorithms, whereas the weather forecast data for testing was obtained from World Weather Online API. Moreover, the research topic in (Nigam and Govinda, 2017) was flight delay prediction with weather data, however no single data source or reference for the weather data was mentioned in the paper.

As is evident from the sources stated above, most of them are USA-based, evidently for USA-based research. Exceptions are METAR, which contains data from stations all over the world and the World Weather Online API source. Most sources are generally trustworthy, originating from governmental databases or specific service providers. Additionally, as was the case for (Klein et al., 2010) and (Choi et al., 2016), multiple sources of datasets are often combined to develop the models (Sternberg et al., 2017).

In general, it is also visible from the papers that the prediction performance with weather is significantly better than performance without weather. Also, it is evident from (Choi et al., 2016) that the prediction performance of delay prediction using weather forecast (which includes uncertainty) is a lot lower on days before the operation, compared to 0h before the operation (actual weather). Additionally, in Chapter 4, there is Figure 4.1, which indicates that there are a lot more cancellations during bad weather, so weather should definitely be an asset for the dataset.

Regarding the most present features, the following points can be identified. Wind is always present, be it as wind speed ( $m/s$ ), direction (deg) or gust speed ( $m/s$ ). As wind is a major influence on airport operations, this makes sense. Visibility ( $m$ ) is also always present, since this determines things like decision height and runway visual range. Precipitation and snow ( $mm$ ) seem to be of interest as well. Then there are the weather codes, which are often, depending on the source, rankings or numbers assigned to specific weather conditions such as thunderstorms, blizzards, mist, snowstorms,...

Table 3.2: Summary of data sources and features for weather data used in topical literature, with their target variable and prediction horizon.

Reference	Source	Example Features	Target	Prediction Horizon
(Klein et al., 2010)	METAR, NCWD, LAMP, CCFP, CIWS	En-route Convective Weather, Local Convective Weather, Wind, Snow, Ceiling, Queuing Delay and Ripple Effects	Airport Delay (Regression)	4 Hours
(Choi et al., 2016)	NOAA, World Weather Online	Wind Direction, Wind Speed, Visibility, Precipitation, Snow Depth, Snow Accumulation, Weather Codes (Intensity, Precipitation, Obscuration)	Airline Delay (Classification)	5 Days, 1 Day, 0 Days
(Nigam and Govinda, 2017)	No Sources	Visibility, Temperature, Weather Type, Humidity, Wind Speed, Wind Direction, Pressure, Altimeter, Pressure Change, Pressure Tendency	Flight Delay (Classification)	Several Hours
(Kim et al., 2016)	NOAA	Wind Direction, Wind Speed, Cloud Height, Visibility, Precipitation, Snow Accumulation, Intensity, Descriptor, Observation Code	Flight Delay (Classification)	Several Hours
(Chen and Li, 2019)	NOAA	Visibility, Temperature, Humidity, Wind (Gust) Speed, Pressure, Weather Type (drizzle, mist, thunderstorm, snowstorm)	Flight Delay (Classification)	Several Hours

## 3.2. Pre-processing

Machine learning algorithms require specifically pre-processed and structured data, in order to function efficiently. This section summarises findings of the three main steps to successfully pre-process data for machine learning, namely interpolation, generalisation, discretisation, removal and normalisation. A summary of data cleaning techniques used in topical literature can be found in Table 3.3.

Table 3.3: Summary of data cleaning techniques used in topical literature, with corresponding target and prediction horizon.

Reference	Cleaning Method	Target	Prediction Horizon
(Manna et al., 2017)	Removal (Outliers) Normalisation (0-1 Scale)	Flight Delay (Regression)	Several Months
(Choi et al., 2016)	Interpolation (Weather Data) Generalisation (Cancelled, Diverted to Delay) Normalisation (no scale)	Airline Delay (Classification)	5 Days, 1 Day, 0 Days
(Horiguchi et al., 2017)	Discretisation (Pax into age intervals) Normalisation (0-1 Scale)	Flight Delay (Classification)	5 Months, 1 Week, 1 Day
(Belcastro et al., 2016)	Generalisation (to Delay) Removal (Cancelled and Diverted)	Flight Delay (Classification)	Several Hours

### a) Interpolation

With data cleaning, multiple data operations are actually encompassed. In (Choi et al., 2016), weather data is used in combination with flight data for flight delay prediction. However, the weather data contains missing values. This is where the data cleaning will come in handy, since the authors solve this problem by linear interpolation using two neighbouring values. This means that a new value will be calculated and assumed, based on neighbouring values. When data is missing, interpolation can be a useful operation to avoid removing data that might contain important information. In a study on the effect of fitting distribution for interpolation methods performed by (Noor et al., 2014), three interpolation techniques are discussed, namely linear, quadratic and cubic. It is demonstrated that every interpolation technique provides a highly suitable fit for the data. This again proves that interpolation is a useful data cleaning technique.

### b) Generalisation

Furthermore, in the flight dataset of (Choi et al., 2016), cancelled and diverted flights are assumed delayed. Hence, the data is generalised towards delays. Data on diverted and cancelled flights were also filtered out from a flight dataset in (Belcastro et al., 2016). From the weather dataset, all non-airport related weather observation locations were removed. Again, this method of filtering irrelevant data contributes to the generalisation of the dataset towards the target, which is in this case again delays.

### c) Discretisation

Data cleaning methods are employed in (Horiguchi et al., 2017) as well. They deal with flight data and passenger reservation data. Data discretisation is performed, since passengers are grouped within age intervals. The transformed data feature will then be the number of people within a certain interval. This discretisation allows usage of limited computational packages or allow an improved prediction performance of machine learning models (Sternberg et al., 2017).

### d) Removal

The flight dataset in (Manna et al., 2017) contained a large number of outliers. They worked with flight delay data and delay times ranged approximately between -90 minutes and 1850 minutes. Therefore, the interquartile range (IQR) was used to sanitise or clean the data. IQR represents the difference between the 75th (Q3) and the 25th (Q1) percentile. The delay times were minimised to range between  $Q1 - 1.5 \cdot IQR$  and  $Q3 + 1.5 \cdot IQR$ . Outlier removal seems interesting, since these outliers represent irrelevant data and might introduce over-fitting or decrease model performance (Sternberg et al., 2017).



### e) Normalisation

It can also be noted that data normalisation is a key operation in data pre-processing. In (Choi et al., 2016), normalisation is performed to scale the feature range, however, the authors do not mention their methods neither the normalisation range. In (Horiguchi et al., 2017) the authors apply min-max normalisation to scale the features to be in a range from 0 to 1. The same 0 to 1 scale normalisation is also applied in (Manna et al., 2017). In the data operation of normalisation, a common scale is applied to the numerical feature data, while respecting the differences and variations within the ranges of the values. It eliminates misleading feature importance due to larger numerical values, whereas larger values do not necessarily match with higher predictor value.

## 3.3. Encoding

datasets often contain numerical data, but also categorical data. Since machine learning algorithms only accept numerical values, it is of important to transform all categorical data to numerical using encoding techniques (Potdar et al., 2017). A summary of some encoding techniques used in topical literature is present in Table 3.4.


Table 3.4: Summary of encoding techniques used in topical literature, with corresponding target and prediction horizon.

Reference	Encoding Method	Target	Prediction Horizon
(Chen and Li, 2019)	One-Hot Encoding	Flight Delay (Classification)	Several Hours
(Lambelho et al., 2020)	Target Encoding Periodic Encoding	Flight Delay and Cancellations (Classification)	6 Months
(Horiguchi et al., 2017)	One-Hot Encoding Periodic Encoding	Flight Delay (Classification)	5 Months, 1 Week, 1 Day
(Chakrabarty, 2019)	Ordinal Encoding One-Hot Encoding	Flight Arrival Delay (Classification)	Several Months

### a) One-Hot Encoding

In (Potdar et al., 2017), it is stated that the most popular encoding technique used is One-Hot encoding, also known as One-of-K encoding. (Chen and Li, 2019) and (Horiguchi et al., 2017) use this technique to transform their categorical features to numerical ones. In this technique, a variable with  $n$  observations and  $m$  values is transformed to  $m$  binary variables with  $n$  observations each (Potdar et al., 2017). However, as it is stated in (Lambelho et al., 2020), high cardinality (numbers of elements in a set) of the data may make this technique less suitable for encoding, just like binary encoding, which uses binary bit strings. An example of One-Hot Encoding can be found in Figure 3.1.

Flight No.	Airport			
1	Amsterdam			
2	Paris			
3	London			




Flight No.	Amsterdam	Paris	London
1	1	0	0
2	0	1	0
3	0	0	1

Figure 3.1: Example of One-Hot Encoding.

### b) Target Encoding

Target encoding is a more suitable technique for data with high cardinality, as suggested by (Lambelho et al., 2020). In his research, flight schedule data was used with the objective to predict flight delays and cancellations. Using a delay classifier as an example, target encoding encodes a categorical feature, such as an airline, based on the probability that a flight of that airline will be delayed. Furthermore, the feature *airport* was encoded using its geographical coordinates and target encoding. An example of Target Encoding can be seen in Figure 3.2

Flight No.	Airport	Cancelled
1	Amsterdam	Yes
2	Paris	No
3	London	Yes
4	Amsterdam	No




Flight No.	Airport	Cancelled
1	0.5	Yes
2	0	No
3	1	Yes
4	0.5	No

Figure 3.2: Example of Target Encoding.

### c) Periodic Encoding

A different encoding technique is used for periodic features. Trigonometric functions are used in (Horiguchi et al., 2017) and (Lambelho et al., 2020) to transform periodic data, such as *departure day of year and scheduled departure and arrival time* into a numerical feature vector. For example, this allows the model to treat New Year's Eve and New Year's Day as consecutive dates. For departure day of year  $d$  this is done by integrating  $\sin(2\pi d/365)$  and  $\cos(2\pi d/365)$  in a feature vector. An example of this Periodic Encoding can be seen in Figure 3.3.

Flight No.	Time (h)
1	01:00
2	06:00
3	12:00
4	15:00
5	17:00
6	21:00
7	23:00




Flight No.	Sin Time	Cos Time
1	0.2588	0.9659
2	1	0
3	0.5	-0.8660
4	-0.7071	-0.7071
5	-0.9659	-0.2588
6	-0.7071	0.7071
7	-0.2588	0.9659

Figure 3.3: Example of Periodic Encoding.

### d) Ordinal Encoding

Another encoding technique is ordinal encoding, also known as label encoding, where an integer is assigned to each category. (Chakrabarty, 2019) utilises label encoding to encode all flight data features. The downside here is that an order or ranking becomes inevitable, which might not be actually existing (Potdar et al., 2017). Please find an example of Ordinal Encoding in Figure 3.4.

Flight No.	Airport
1	Amsterdam
2	Paris
3	London
4	Amsterdam



Flight No.	Airport
1	1
2	2
3	3
4	1

Figure 3.4: Example of Ordinal Encoding.

## 3.4. Dealing with Imbalanced datasets

As only about 1-3% of all flights are cancelled on average (Rupp and Holmes, 2006), the data will be highly imbalanced. Using this imbalanced data for machine learning applications may result in extended training time and a degraded prediction performance, as is stated in (Gao et al., 2015), who performed a study on the combination of feature selection and data sampling for imbalanced data, and in (Mollineda et al., 2007), who reviewed the most important researches on the topic of dealing with class imbalances. A summary of some papers that incorporated data sampling to deal with imbalances is presented in Table 3.5.

Table 3.5: A summary of data sampling methods in topical literature, to account for imbalanced data.

Reference	Sampling	Target	Prediction Horizon
(Chen and Li, 2019)	SMOTE	Flight Delay (Classification)	Several Hours
(Choi et al., 2016)	SMOTE & RMU	Airline Delay (Classification)	5 Days, 1 Day, 0 Days
(Chakrabarty, 2019)	R-SMOTE	Flight Arrival Delay (Classification)	Several Months
(Belcastro et al., 2016)	RMU	Flight Delay (Classification)	Several Hours

#### a) SMOTE

To avoid imbalanced data leading to a biased flight delay prediction model, (Chen and Li, 2019) applies the synthetic minority oversampling technique (SMOTE). This technique is centered around oversampling the minority class, by creating synthetic samples. SMOTE generalises the decision region of this minority class, by multiplying the difference between a data sample and its nearest neighbour with a random number between 0 and 1 and then by adding the result to the sample under consideration. This technique was researched and developed in a research performed by (Chawla et al., 2002). In essence, synthetic samples are created on the lines between the minority samples and their nearest neighbours. Additionally, the airline delay prediction model from (Choi et al., 2016) performs better with SMOTE in terms of minority class recognition, compared to without SMOTE. (Chakrabarty, 2019) has used a somewhat modified version of the SMOTE algorithm, namely Randomised-SMOTE (R-SMOTE). Instead of looking for the nearest neighbours, it randomly selects minority class samples and then performs the SMOTE algorithm between them.

#### b) RMU

The SMOTE technique is also used in (Choi et al., 2016) for predicting airline delay, in combination with random majority undersampling (RMU). In RMU, the majority is under-sampled, removing data samples on a random basis. In the SMOTE research paper by (Chawla et al., 2002), it is suggested that SMOTE in combination with RMU “performs better than plain undersampling”. On the contrary, (Belcastro et al., 2016) uses just RMU in his flight delay prediction algorithm to account for data imbalance and manages to achieve quite good results.

#### c) MOR

Minority oversampling with replacement (MOR) is a technique that essentially creates duplicates of the minority class samples by copying them and supplying them to the dataset. It was concluded that the technique did not significantly improve the recognition of the minority class. In terms of decision regions in feature space, it is stated by the SMOTE developers in (Chawla et al., 2002), that MOR does not cause the decision boundary to expand towards the majority class region. In essence, this could lead to overfitting, since the same samples are only duplicated and the objective of generalisation (being able to successfully predict on unseen data) is not satisfied.

#### d) When to Sample

Finally, in the research by (Gao et al., 2015) on the combination of feature selection (which will be discussed in section 3.5) and data sampling for imbalanced data, it is suggested that the following order of data manipulations is a highly suitable approach; Feature selection should only be performed once the data sampling is complete. Then, after the most important features are extracted, they are translated to and extracted from the *unsampled* dataset. In other words, the data sampling is only performed in order to provide an equally balanced dataset for feature selection.

### 3.5. Feature Selection

Often, datasets are quite large and a high number of attributes or data features is present. Unfortunately, not all features are relevant for the intended classification task and the high number might drastically raise computational complexity of machine learning algorithms. This is also known as *the curse of dimensionality* (Pechenizkiy, 2005). There might be irrelevant features, not affecting the target in any way, and there might be redundant features, not adding any additional information to the target. Therefore, machine learning might

not efficiently work, before the most relevant features are selected and the irrelevant and redundant features are removed (Dash and Liu, 1997). This section addresses feature selection for intended use in machine learning, by describing several selection techniques. Additionally, a summary of feature selection techniques and papers is given in Table 3.6.

Table 3.6: Summary of feature selection techniques used in topical literature, with their target and prediction horizon.

Reference	Selection	Target	Prediction Horizon
(Kuhn and Jamadagni, 2017)	RFE	Flight Arrival Delay (Classification)	Several Hours
(Manna et al., 2017)	Correlation	Flight Delay (Regression)	Several Months
(Chen and Li, 2019)	RFE	Flight Delay (Classification)	Several Hours
(Lambelho et al., 2020)	RFE	Flight Delay and Cancellations (Classification)	6 Months
(Alonso and Loureiro, 2015)	Literature & Expert Knowledge	Flight Delay (Regression)	Several Hours
(Kalliguddi and Leboulluc, 2017)	Correlation	Flight Delay (Regression)	Several Hours

#### a) Pearson's Correlation Coefficient

Now, a couple of feature selection techniques will be covered, starting with (Manna et al., 2017). This paper analyses the correlation between the features themselves and between the features and the target, using Pearson's correlation coefficient. This coefficient measures the linear association strength between between two features. A correlation coefficient of  $\pm 1$  resembles a perfect positive/negative correlation. The higher the correlation of a feature with the target (applied to the thesis topic, the target is the *cancelled or not* feature), the better performance this feature will have in classification. However, when comparing features with other features (so not with the target feature), a coefficient higher than 0.5 points at multicollinearity. Hence, one of the two features must be abandoned, preferably the one with the lowest correlation with the target. This method is an example of a *filter* method, with a subset of relevant features going into the model after selection/filtering. The Pearson correlation was also used in (Gao et al., 2015), a paper in which the order operation of data sampling and feature elimination was researched. (Kalliguddi and Leboulluc, 2017) uses the Pearson Correlation coefficient to perform preliminary data analysis and analyse the correlation between the variables.

#### b) Recursive Feature Elimination

A different feature selection technique is also applied regularly, namely recursive feature elimination (RFE). In this technique a learning algorithm is used, generating a feature importance and ranks the features according to this importance. Afterwards, the least important feature will be eliminated. Elimination performance will then be evaluated with cross-validation, until the feature set yielding the highest performance is found. In (Granitto et al., 2006) Random Forests (RF) and Support Vector Machines are used to rank features. This paper is not present in Table 3.6 since it researched agroindustrial products for biochemical characteristics. In (Kuhn and Jamadagni, 2017) a Decision Tree was used and (Chen and Li, 2019) utilises RF again. These are all examples of the *wrapper* method, which, in contrast to the *filter* method, uses all features in a machine learning algorithm and eliminates them afterwards, based on the performance of the model. This makes the *wrapper* method computationally a lot more expensive than the *filter* method, especially for data with high dimensionality.

#### c) Literature and Expert Knowledge

Others often base their feature selections on literature, engineering sense or expert knowledge, such as (Alonso and Loureiro, 2015).

# 4

## Flight Cancellation Determinants

It is important to look at cancellations from a more strategic, aerospace engineering point of view. When doing so, some concepts influencing cancellation can pop up, that might be of interest for the prediction. Even more interesting is that, even though they possess high-prediction value, they might not directly be present in the raw datasets that are used for model training. (Rupp and Holmes, 2006), (Seelhorst, 2014) and (Alderighi and Gaggero, 2018) are examples of literature that specifically research determinants of cancellation and cancellation behaviour. Please note that these determinants were (to the best of knowledge) **not used before** as features in machine learning models for cancellation prediction, especially since research papers on cancellation prediction are very scarce. They were identified during investigations into cancellation behaviour. Therefore, obtained data might need some alteration or extra manipulation in order to add and represent some of these interesting cancellation determinants.

### **a) Airline in Alliance**

The first new determinant that is treated, is whether an airline is part of an alliance or not. This largely addressed in the research by (Alderighi and Gaggero, 2018). They found out that airlines belonging to global alliances, are more likely to have flight cancellations compared to non-alliance airlines. Additionally, it was found that the average delay duration for these airlines is higher. Figure 4.1 shows graphs from the research, indicating that for both bad and good weather, the cancellation rate for alliance airlines is higher. These graphs also highlight the fact that bad weather is certainly influential on flight cancellations, i.e. more cancellations can be observed during bad weather. Examples from global alliances from the research are Oneworld, Skyteam and Star Alliance.

### **b) Competitiveness**

From (Rupp and Holmes, 2006), it is shown that routes having a high competitiveness show lower cancellation rates. On the other hand, routes with low competitiveness, also called monopoly routes, show higher cancellation rates. The authors explain the increase on cancellations on monopoly routes with the fact that these routes are mostly on smaller airports lacking mechanics. Hence, this is “an airport effect rather than monopoly effect” (Rupp and Holmes, 2006). This can also be translated towards market-share, i.e. airline carriers that have a higher route-level market share, cancel their flights more often.

### **c) Hub Operations**

Carriers are said to cancel flights from and to their hub less frequently, which is particularly distinct for large hub operations (Rupp and Holmes, 2006). This is because they can therefore better maintain their flight network. This statement is also confirmed in the research by (Seelhorst, 2014), who states that “these flights are important to airlines due to the large number of connecting passengers at hub airports, so this result is not surprising”. The author also points out that a flight originating at the hub is even less likely to be cancelled than a flight going to a hub, since that leaves passengers stranded at the hub rather than at their origin or destination.

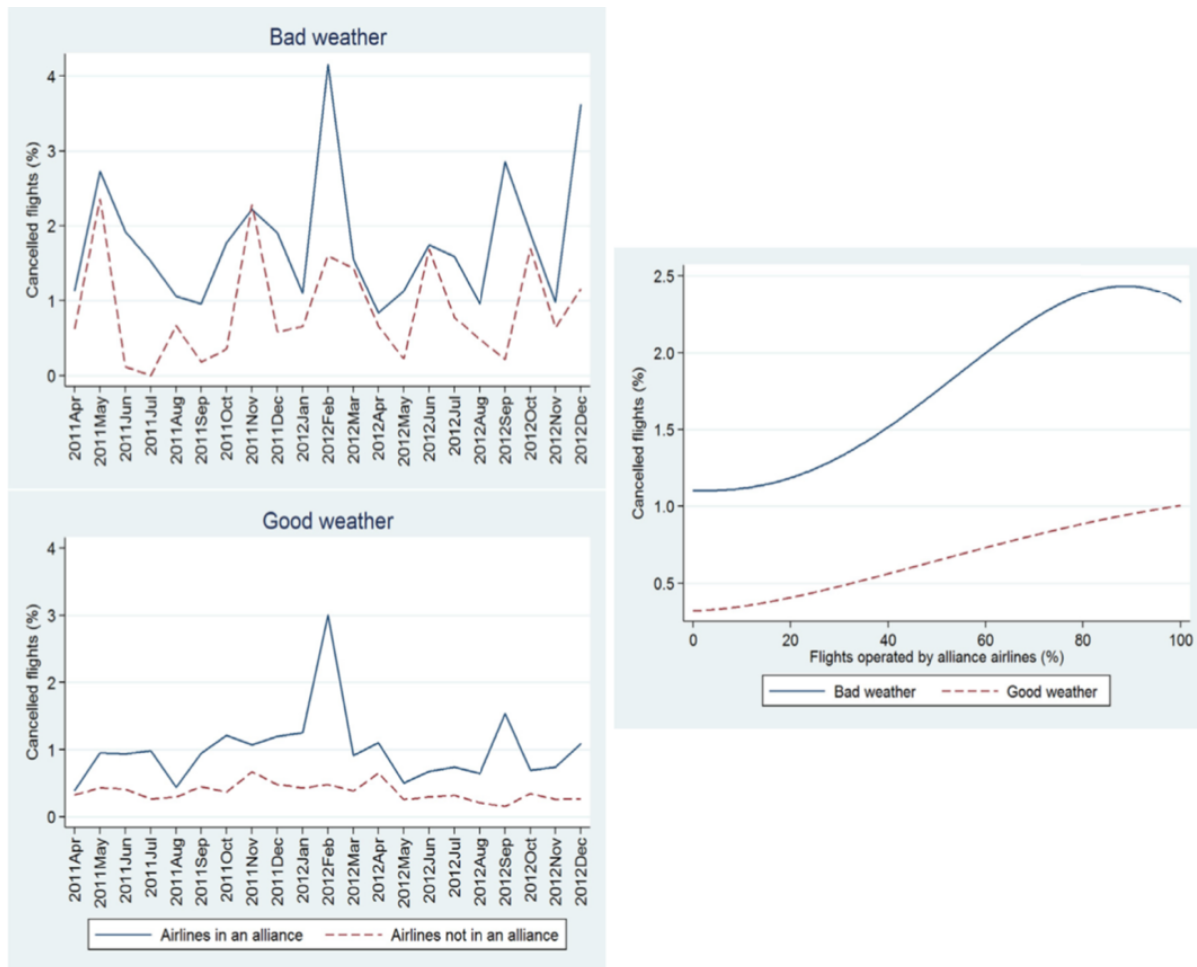


Figure 4.1: Sample period of flight cancellations (left) and route-month percentage of flight cancellations and airline alliances (right). (Alderighi and Gaggero, 2018)

#### d) Passenger Inconvenience

Airlines tend to strategically avoid passenger inconvenience when deciding on flight cancellations. Three types of determinants will be discussed below, namely the route frequency, load factor and whether a flight is one of the final flights of the day.

- Routes flown infrequently are cancelled less, since airlines try to minimise passenger inconvenience (Rupp and Holmes, 2006). This implies that on high frequency routes, flights are cancelled more often. This is a logical consequence since airlines have less options to reschedule the passengers if e.g. a route is only flown once or twice a week. This is also confirmed by (Pai, 2010), who found higher cancellation rates on on high frequency routes.
- In his research, (Rupp and Holmes, 2006) also found out that flight being served by fuller planes are cancelled less often. However, their models show that fuller planes do require longer boarding times and therefore experience flight delays more frequently. So in conclusion, fuller planes are not likely to experience cancellations, but more likely to be delayed.
- (Rupp and Holmes, 2006) finds a significant reduction in cancellations for the final flight(s) of the day. Carriers seem to have less options to reschedule when the at a later time of the day, which is mostly around the final flights. If a flight gets cancelled at night, there is the risk that the passengers will have to be reimbursed for a hotel stay overnight or that they will go to a competing airline. Also, from a flight network perspective, not cancelling the last flight of the day sets the carrier up for regular operations on the following morning, which would certainly be beneficial.

**e) Revenue Maximisation**

The revenue maximisation objective plays an important role for flight cancellations, since (Rupp and Holmes, 2006) found support to back up the fact that there is a significant reduction in flight cancellations on certain routes with a high average revenue. Hence, carriers often strategically take cancellations into account when trying to maximise their revenue. This effect is most common on small and mid-sized airports. Furthermore, (Seelhorst, 2014) adds that airlines tend to avoid cancelling high fares, since they are associated with high-value customers, representing a large source of income for the carrier. This can be linked to the passenger inconvenience as well, since the airline favours the high-value customers over low-value customers.





# 5

## Machine Learning

Proper knowledge on the fundamentals of machine learning is an unquestionable necessity when considering a thesis topic involving these types of models. This chapter will therefore report on the basics of machine learning, different algorithms used in relevant literature and how performance of such algorithms is evaluated.

### 5.1. Machine Learning Fundamentals

This section will cover the fundamentals of machine learning, by firstly discussing the definition, aim and the meaning of ‘the machine learning pipeline’. Afterwards, the two main learning types will be addressed, followed by the description of some popular machine learning tasks.

#### 5.1.1. Definition

The ‘machine learning pipeline’ is accurately described as follows: “A computer program is said to learn from **experience**  $E$  with respect to some class of **tasks**  $T$  and **performance** measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .” (Tom M. Mitchell, 1997). Essentially, this definition states that machine learning models need (the relevant) experience (or data) in order to learn and apply it (or generalise it). An important notion is that *learning* is not the *task*. However, the ability to be able to perform the task is obtained by learning. Four different examples of tasks are classification, regression, anomaly detection and machine translation (Migut, 2019). This concept is visualised in the diagram in Figure 5.1. The experience is the training data, the task is pointed out on the diagram and the performance is measured with the output. The performance measure  $P$  of the algorithms, enables the performance to be evaluated quantitatively. The different techniques for assessing this, are addressed in section 5.3. Lastly, the domain objects on the diagram, with corresponding features, represent the unseen data on which the model is tested. (Choi et al., 2016) has visualised his machine learning model in his paper as can be seen in Figure 5.2. It has a lot of resemblance with the pipeline presented in Figure 5.1.

#### 5.1.2. Learning Types & Tasks

In machine learning, there are two main types of learning, namely *supervised* learning and *unsupervised* learning. The former requires training data that is labelled, whereas the latter uses unlabelled data (Flach, 2012). For example, if flight cancellations need to be predicted, supervised learning requires a dataset containing flight data, labelled as ‘cancelled’ or ‘not cancelled’. Unsupervised learning, uses a dataset with flight data that has no indication on the cancellation status. Given the nature of the thesis topic, the rest of this literature review will be focused on supervised learning.

##### a) Classification

In machine learning, classification is most common in terms of tasks. The algorithm will try to assign a data sample to a certain class, part of a small set of class labels. In other words, it aims to specify to “which of  $k$  categories some input belongs to” (Migut, 2019). Hence, these types of algorithms are often called *classifiers*. Generally, the set of class labels encompasses a finite set of classes, the simplest case containing only two of them. This scenario is known as *binary classification*. There are only two classes, mostly depicted as a

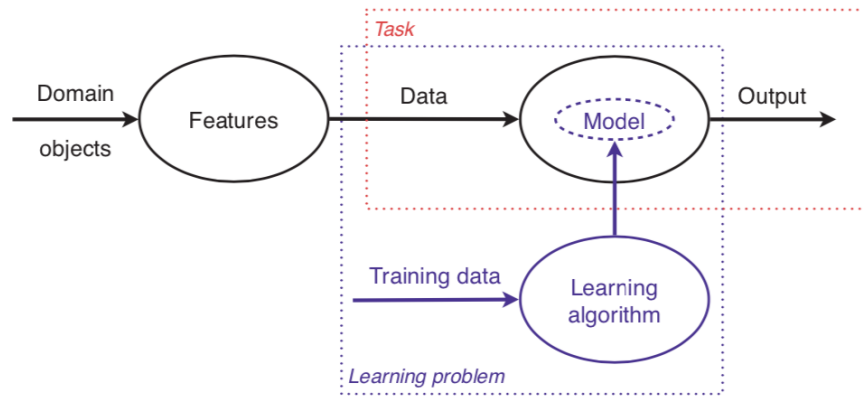


Figure 5.1: Visualisation of the machine learning pipeline. (Flach, 2012)

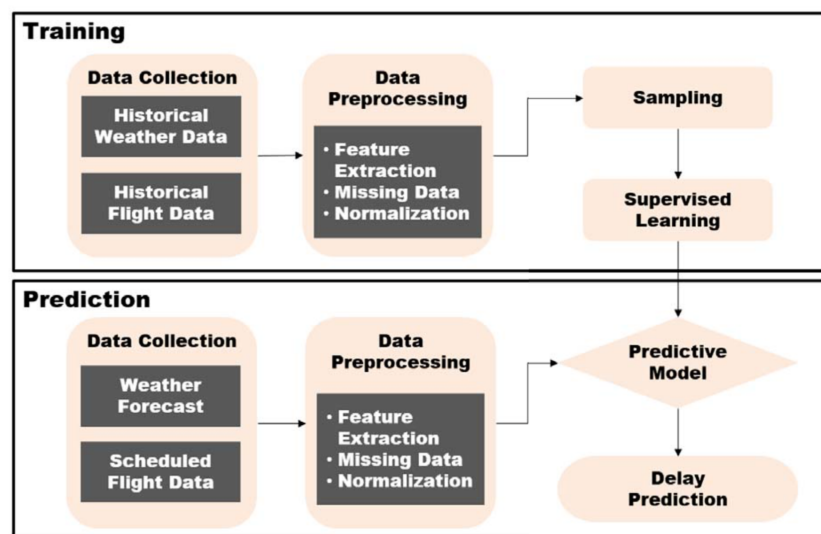


Figure 5.2: An example of a machine learning model from (Choi et al., 2016), with resemblance with the machine learning pipeline.

positive (+) and a negative (-) class. This task often returns in literature, especially in on-time performance predictions. It is predicted whether a flight is delayed (+) or not (-) and cancelled (+) or not (-) (Lambelho et al., 2020), (Choi et al., 2016). It is important to note that it might be counter intuitive to call a cancellation 'positive' and flying 'negative', however, this is decided based on the prediction goals of the model. In these examples, the main goal of the model is to correctly identify cancellation or a delay, therefore these are the positives.

## b) Regression

The second common task is regression. (Kalliguddi and Leboulluc, 2017), (Horiguchi et al., 2017), (Manna et al., 2017) and (Gopalakrishnan and Balakrishnan, 2017) are all examples of papers using machine learning regression tasks to predict the exact amount of flight delay. Here, the target variable has a real value. This implies that a switch has been made from a relatively low resolution in classification, towards an infinite resolution. In other terms, instead of e.g. looking whether a flight is cancelled or not, one looks at how many flights are cancelled (for example, on a day). This implies a real value in return. To match this infinite resolution, the regression tasks asks a high precision of its function estimator, making it prone to overfitting. Moreover, it is possible that fluctuations in the data are present, which are impossible for the model to capture. Therefore it is assumed that a regressor only captures the approximate function trend. (Flach, 2012)

## 5.2. Train-Test Split

As stated before, machine learning models rely on past data to ‘learn’ and generalise towards unseen data. In practice, this process is often approximated by dividing a large historical dataset into a training and test set. This section aims to summarise the different train-test splits made and methods used in literature. Important to note is that machine learning takes on the *independent and identically distributed* data assumption. This implies that it is assumed that the data samples in the test and training sets are *independent* AND that both datasets are *identically* distributed (Migut, 2019).

### a) Single Split

The term ‘single split’ refers to datasets that were split up in training and test sets only once, according to a fixed ratio. Example ratios are 70/30, 80/20 and 90/10 in percentages. In (Horiguchi et al., 2017), 47,000 flights out of 54,000 were used for training a fuel consumption and flight delay prediction model and naturally 7,000 flights are reserved for testing. This equals a 87/13 ratio for train-test split. A significantly higher number of training and test instances is present in (Manna et al., 2017), where the flight delay prediction model is trained on 2,175,534 flights and tested on 543,883 flights. This corresponds to a 80/20 split. Furthermore, (Kuhn and Jamadagni, 2017) utilises a 70/30 training-test split, again for flight data, with the aim to construct a flight delay classification prediction model. (Nigam and Govinda, 2017) performs binary flight delay classification predictions and utilises exactly the same ratio of 70/30.

### b) K-Fold Cross Validation

Some papers choose to go for the K-Fold Cross Validation (CV) method. This technique is a process of training the same data set K times, each time with different  $100 - 100/k\%$  batch of the data used for training and  $100/k\%$  batch of the data used for validation. For example, 10-Fold CV uses a different 90% batch for training and a 10% batch for validation. This produces a more reliable outcome compared to a single split (Flach, 2012).

In (Gao et al., 2015) 5-Fold Cross Validation was utilised. This means that 5 times a different 20% batch of data is used for testing. Interestingly, (Kuhn and Jamadagni, 2017) first utilises a 70/30 training-test split, as stated above. However, afterwards he applies 10-Fold CV on the training set. This way, the 30% testing data is kept for final validation, whereas the CV is used only on the training set. This is in contrast to (Gao et al., 2015), who did not first split the data, but directly performed the CV on the entire data set. Another interesting variation on the CV and train-test split technique, is found in (Choi et al., 2016). The entire historical dataset is subject to a 10-Fold CV technique, just like (Gao et al., 2015). However, then new data is added to the problem, consisting of planned flight schedules and weather forecasts. This forecast test set only contained 56 flights, whereas the historical training set contained data for flight over approximately 10 years. Remarkable here is that actual real-life data could be used for testing the model, outside of the historical dataset, validated with 10-Fold CV.

## 5.3. Prediction Performance

As there exist numerous machine learning algorithms and multiple experimental set-ups or scenarios are often present, it is essential to have a way to evaluate the prediction performance of the different models and scenarios. In this section the performance of the models is central and multiple ways to assess the performance will be addressed. These are the confusion matrix and receiver operating characteristic (ROC) curve for classification and multiple error metrics for regression.

### 5.3.1. Determinants of Performance

The performance of a machine learning algorithm is determined by the algorithm’s ability to make little training errors and to keep a small gap between the test and training errors. This can be translated into two main challenges, namely *underfitting* and *overfitting*. Underfitting is the result of a training error being too big, whereas overfitting indicates a too large gap between training and test error (Migut, 2019). Both symptoms of decreased performance.

(Kim et al., 2016) uses the ‘dropout technique’ to prevent the flight delay prediction model from overfitting (also called regularisation). Here, random units from Neural Networks are dropped out during training, which is said to improve accuracy of the model. In (Choi et al., 2016), weather-induced flight delays are predicted and the author concluded that after testing, the model performance had degraded due to overfitting to the training data. Furthermore, (Kuhn and Jamadagni, 2017) applies L2-regularisation to prevent their flight

arrival delay prediction model from overfitting. A certain optimal point in capacity should be found, corresponding to an optimal generalisation gap. This can be linked to the *least general generalisation* theorem (Flach, 2012). This means that the aim is to generalise, with the contradictory use of the least general method. In other words, a function with the appropriate generalisation capacity must be selected (not too high, to avoid overfitting), however the least general (least simple) one must be selected (to avoid underfitting).

### 5.3.2. Confusion Matrix

Essentially, the confusion matrix gives the number of class-dependent errors. It provides a detailed view on these errors and it can be utilised to estimate the overall cost that a specific classifier might incur Migut (2019). An example of a confusion matrix can be seen in Table 5.1.

Table 5.1: An example of a confusion matrix.

n=100	Actually Cancelled	Actually Flying	
Predicted Cancelled	4 (TP)	2 (FP)	6
Predicted Flying	1 (FN)	93 (TN)	94
	5	95	

The confusion matrix essentially projects the predictions (left-most column) on the actuals (top row). In this case,  $n = 100$  samples have been predicted. The row labelled as 'predicted cancelled' shows all flights that were predicted as cancelled, intuitively. The same intuition counts for the row 'predicted flying'. Both are divided into 'actually cancelled' flights and 'actually flying' flights. The top-left value (4), equals to the flights that were predicted as cancelled and actually were cancelled, also known as the true positives (TP). Right next to the TP, there are the false positives (FP). This value (2) depicts the flights that were predicted as cancelled by the model, but were actually flying. Below there is the bottom left value (1), corresponding to the false negatives (FN). In this case, meaning flights that were predicted as flying but actually ended up being cancelled. The FP and FN are exactly what is to be avoided when trying to correctly classify flight cancellations. Finally, the bottom-right value (93) are the true negatives (TN), or flights that were predicted to fly and actually also flew. Several performance metrics can be derived from the confusion matrix, namely accuracy, precision and recall.

#### a) Accuracy

Accuracy is the first metric that will be dealt with. It eventually boils down to 'how many of the flights were correctly predicted'. Translating this to a confusion matrix, it can be derived from the simple sum of the True Positives (TP) and the True Negatives (TN), divided by the number of samples  $n$ . Mathematically, this is expressed in Equation 5.1. One could think that this high accuracy of 97% would indicate that the model has a good performance. Naturally, it is essential to look at the precision and recall as well, since a high accuracy alone does not give the full picture.

$$\frac{TP + TN}{n} = \frac{4 + 93}{100} = 0.97 \quad (5.1)$$

#### b) Precision

Precision means 'how many of the flights that were predicted as cancelled, are actually cancelled'. This can easily be found by dividing the TP by the total of predicted cancellations. Mathematically, this is expressed in Equation 5.2. When considering the actual purpose of the model, precision is a lot more valuable than accuracy. It shows how good the model actually is in what it is intended to do, e.g. predict cancellations accurately. The 67% precision already indicates that the model is not as good as the accuracy would make one think.

$$\frac{TP}{TP + FP} = \frac{4}{4 + 2} = 0.67 \quad (5.2)$$

#### c) Recall

Furthermore, a bit in line with precision, there is recall. It tells you 'how many of the actually cancelled flights, were predicted correctly'. So, instead of looking at the total of the predicted cancellations, you look at the total of actual cancellations. In confusion matrix terms, this means dividing the TP with the sum of the TP

and False Negatives (FN). Mathematically, this is expressed in Equation 5.3. Again, this metric carries more value than the accuracy and is comparable to the precision in terms of reasoning why. The 80% recall is still not as good as the accuracy would suggest, but is already better than the precision.

$$\frac{TP}{TP + FN} = \frac{4}{5} = 0.80 \quad (5.3)$$

Now, consider the following literature that utilised these metrics for model evaluation, which is summarised in Table 5.2.

Table 5.2: Accuracy, precision and recall results from topical papers, with their target and prediction horizon.

Reference	Accuracy	Precision	Recall	Target	Prediction Horizon
(Choi et al., 2016)	5 Days: 0.2679 1 Day: 0.3036 0 Days: 0.8036	N/A	N/A	Airline Delay	5 Days, 1 Day, 0 Days
(Kuhn and Jamadagni, 2017)	N/A	DT: 0.93 LR: 0.92 NN: 0.91	DT: 0.88 LR: 0.89 NN: 0.90	Flight Arrival Delay	Several Hours
(Lambelho et al., 2020)	DD: 0.794 AD: 0.791 C: 0.987	DD: 0.516 AD: 0.567 C: 0.608	DD: 0.516 AD: 0.553 C: 0.592	Flight Delay and Cancellations	6 Months
(Chakrabarty, 2019)	0.86	0.88	0.86	Flight Arrival Delay	Several Months

First of all, there are the results of (Choi et al., 2016). The author uses Random Forests to predict flight delays, using flight and weather data. Interestingly, he has applied different time horizons to the predictions and compares the model performance on the different prediction timings by comparing their confusion matrices, and more in particular, the accuracies. Solely based on the accuracy, the 0 day forecast horizon has the best performance, outperforming the other two timings by far. The author states that this is probably due to uncertainties in the weather forecast, in combination with imperfections in his model. However, when calculating the precision (assuming delay as positive and on-time as negative), it is clear that the model performs way worse than the accuracy is indicating. E.g., for the 0 days result, which is said to be the best prediction timing. The precision here is 33% and the recall is a staggering 10%. The author simply does not consider evaluating these performance metrics, leading to a somewhat distorted conclusion. Hence, this evaluation points out the importance of precision and recall for these types of evaluation. Please mind that the precision and recall can be calculated for both the on-time as positive and delay as negative and the other way around. The way this is done, is determined by the aim of the model and by looking at the most important prediction goal. Here, this is the correct prediction of flight delays.

A second example is taken from (Kuhn and Jamadagni, 2017), who compares the performance of three different learning algorithms. These are Decision Tree (DT), Logistic Regression (LR) and Neural Networks (NN). As can be seen, the author does take into account the precision and recall within his analysis and he does not even look at accuracy. The results indicate a fairly good prediction performance for all algorithms.

Furthermore, (Lambelho et al., 2020) utilises all performance indicators. In his research, he compared three different algorithms, however, for simplicity here only the results of LightGBM are shown. The author predicts three different elements, namely Departure Delay (DD), Arrival Delay (AD) and Cancellations (C). Again, here it is clear how accuracy could be deceiving and precision and recall point out that the prediction performance is very mediocre.

Lastly, (Chakrabarty, 2019) is used as an example. The machine learning algorithm used was a gradient boosting classifier and the results are shown in the table. All performance indicators show that the algorithm has a good prediction power, especially since the precision and recall are quite high.

### 5.3.3. Area Under the ROC Curve

A second recurrent measure of performance is the Area Under the ROC Curve (AUC). The ROC curve plots the relation between the True Positive Rate (TPR), which is actually the same as the recall, and the False Positive Rate (FPR), which is the probability of a false alarm, as a function of classification threshold. Mathematically, this translates to  $TPR = \frac{TP}{TP + FN}$  and  $FPR = \frac{FP}{FP + TN}$ . The classification threshold is of importance for e.g. flight delay prediction, since it gives the algorithm the indication after which time a flight is classified as delayed and not delayed. A summary of classification thresholds used in topical literature can be seen in Table 5.3.

Table 5.3: Summary of classification thresholds used in topical literature, with corresponding target and prediction horizon.

Reference	Classification Threshold	Target	Prediction Horizon
(Belcastro et al., 2016)	15min, 60min	Flight Delay	Several Hours
(Choi et al., 2016)	15min	Airline Delay	5 Days, 1 Day, 0 Days
(Horiguchi et al., 2017)	15min	Flight Delay	5 Months, 1 Week, 1 Day
(Alonso and Loureiro, 2015)	]0,15min], ]15min,30min], ]30min,60min]	Flight Departure Delay (Multiclass)	Several Hours
(Rebollo and Balakrishnan, 2014)	45min, 60min, 90min	Air Traffic Delay	2 Hours

Multiple ROC curves can be put into the same graph, enabling the comparison of multiple classifiers (Migut, 2019). An example of an ROC curve is present in Figure 5.3. The most optimal point in this curve would be the top-left point, at (0,1). This corresponds to a TPR of 100 and FPR of 0, implying that all of the actual positive samples have been classified (predicted) correctly.

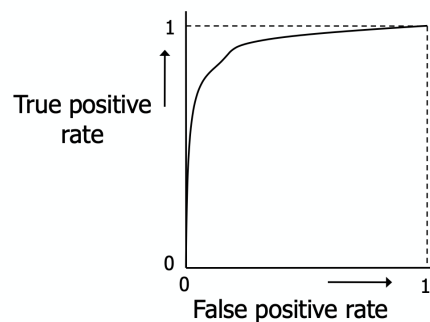


Figure 5.3: An example of a Receiver Operating Characteristic Curve (Migut, 2019).

The AUC is always a number between 0 and 1 and is another way of measuring the performance of the models. The higher the AUC, the better the predictive capabilities (Flach, 2012). A perfect classifier should therefore give an AUC of 1.0, whereas a random classifier gives an AUC of 0.5. This random classifier is also often shown in the ROC diagram as a diagonal line from (0,0) to (1,1). Table 5.4 gives a summary of some AUC results of topical papers. The best result is highlighted in bold per paper, except for (Kuhn and Jamadagni, 2017) since he obtained the same AUC for all algorithms.

The results of (Choi et al., 2016) indicate that Random Forests has the best predictive power of the four classifiers used in the paper. Please note that these AUC results were obtained after 10-Fold Cross Validation and used current weather (0 Days prediction horizon).

The flight delay prediction model of (Horiguchi et al., 2017) was evaluated using AUC. He compares three different machine learning algorithms in his results and depicts different prediction timings as well, namely one day before, one week before and five months before the flight. From the results, the author could conclude that the only relatively good predictions are achieved with information available one day before the flight. Also remarkable is that for Neural Networks, five months before the flight the AUC is even equal to that of a random classifier.

(Kuhn and Jamadagni, 2017) had a particular result, since all algorithms scored the same in the AUC evaluation. Therefore, the author had to rely on other performance indicators such as precision and recall for more insights into the performance.

In (Lambelho et al., 2020) three different algorithms are used for the prediction of three different targets, namely departure and arrival delay, and cancellations. For each of the cases, the LightGBM algorithm seemed to have the highest AUC performance.

Table 5.4: AUC results from topical papers, with their target and prediction horizon. The best result is highlighted in bold.

Reference	AUC	Target	Prediction Horizon
(Choi et al., 2016)	<b>Random Forests: 0.68</b> AdaBoost: 0.66 k-Neighbours: 0.66 Decision Tree: 0.64	Airline Delay	0 Days
(Horiguchi et al., 2017)	<b>Neural Networks (1 Day): 0.647</b> XGBoost (1 Day) : 0.634 Random Forests (1 Day): 0.604 Neural Networks (1 Week): 0.584 XGBoost (1 Week): 0.573 Random Forests (1 Week): 0.560 Neural Networks (5 Months): 0.5 XGBoost (5 Months): 0.542 Random Forests (5 Months): 0.534	Flight Delay	5 Months, 1 Week, 1 Day
(Kuhn and Jamadagni, 2017)	Decision Tree: 0.96 Logistic Regression: 0.96 Neural Network: 0.96	Flight Arrival Delay	Several Hours
(Lambelho et al., 2020)	<b>Departure Delay LightGBM: 0.786</b> Departure Delay Neural Network: 0.754 Departure Delay Random Forest: 0.744 <b>Arrival Delay LightGBM: 0.803</b> Arrival Delay Neural Network: 0.774 Arrival Delay Random Forest: 0.758 <b>Cancellation LightGBM: 0.929</b> Cancellation Neural Network: 0.840 Cancellation Random Forest: 0.862	Flight Delay and Cancellation	6 Months

In (Choi et al., 2016), ROC curves were constructed to evaluate the performance of flight delay predictions with and without the inclusion of weather data. The curves can be seen in Figure 5.4. From the results it could be derived that weather clearly had a beneficial influence on the prediction performance, as in three out of four cases the blue line lies closer to the optimal point compared to the red dotted line. The author states that the fact that there is hardly any gap between the two lines in the kNN (k-Nearest-Neighbours) graph, is due to the curse of dimensionality. The high number of features is not helpful for kNN, as it is said to become meaningless to measure distance between this high number of sample points. This proves that ROC and AUC are valuable metrics when it comes to performance evaluation of machine learning problems and the comparison of different models and scenarios.

#### 5.3.4. Regression Error Metrics

For regression, different evaluation measures should be applied, since in this case the outcome is not a class but a real value. In (Manna et al., 2017), the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the Coefficient of Determination ( $R^2$ ) were used to evaluate the prediction of the average departure and arrival delays. The author states that “the MAE helps to determine how close the predicted outcomes are to the consequent outcomes”. Equation 5.4 shows the formula for MAE and Equation 5.5 shows the equation for the RMSE. Here,  $n$  is the number of samples,  $y_i$  is the actual outcome and  $\hat{y}_i$  is the predicted outcome. The RMSE “helps to expand and liquidate the large errors”. The formula for  $R^2$  is shown in Equation 5.6 and is denoted by  $R^2$ . Here,  $\bar{y}_i$  represents the mean of  $y_i$ .  $R^2$  is a classic regression analysis tool and shows how close the data is to the fitted regression line.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5.4)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5.5)$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (5.6)$$

In (Horiguchi et al., 2017), fuel consumption prediction is performed using machine learning algorithms. The error metric used for evaluation is the Relative RMSE (RRMSE). It essentially is the RMSE, but normalised by the mean of the values observed. The mathematical notation for the RRMSE (%) can be found in Equation 5.7.

$$RRMSE = \frac{RMSE}{\bar{y}} \times 100 \quad (5.7)$$

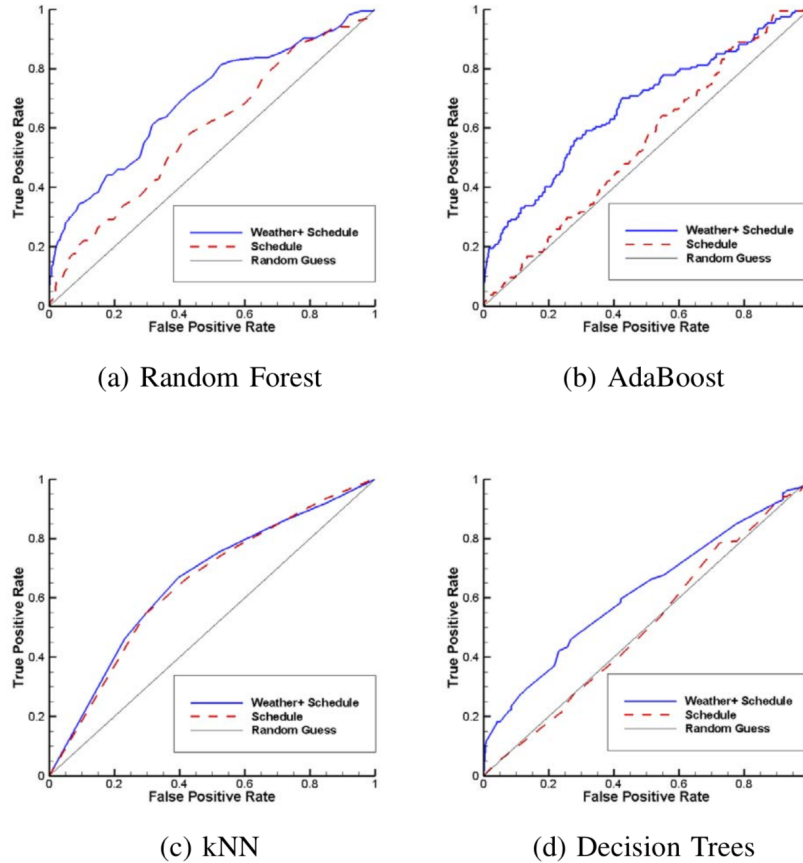


Figure 5.4: ROC curves from (Choi et al., 2016), comparing performance with and without weather data.

## 5.4. Algorithms

The main purpose of this section is to identify the different types of algorithms used in literature, relating to prediction of flight delays or cancellations. The most common and popular algorithms are summarised in Table 5.5, with the corresponding sources, target and prediction horizons. This section also contains a very concise description of the algorithms used, in the subsections underneath.

The algorithm must not be too complex but also not too simple, hence, model comparison is essential (Migut, 2019). This is confirmed in (Kim et al., 2016), where specifically the generalisation performance of the model is evaluated. Revisiting the theorem stated in the introduction to this review, the *no free lunch* theorem, which stated that essentially there is no single machine learning algorithm that is universally better than any other one, it can be concluded that it is key to test multiple models and compare their performance. This is proven to be often the case, as in Table 5.5 often multiple algorithms are chosen and compared. For every paper, the model that got the highest prediction performance is highlighted in bold. For (Kuhn and Jamadagni, 2017), all algorithms had about the same prediction performance, hence no one is highlighted.



Also, since in (Chakrabarty, 2019), (Chen and Li, 2019), (Kim et al., 2016) and (Rebollo and Balakrishnan, 2014) only one model was considered, no highlight is present there either.

Table 5.5: Algorithms used in topical literature, with the target variable and prediction horizon. The algorithm with the highest performance in its respective paper is highlighted in bold.

Reference	Algorithm(s)	Target	Prediction Horizon
(Choi et al., 2016)	Decision Tree, <b>Random Forests</b> , k-Nearest-Neighbours, AdaBoost	Airline Delay (Classification)	5 Days, 1 Day, 0 Days
(Kuhn and Jamadagni, 2017)	Decision Tree, Neural Networks, Logistic Regression	Flight Arrival Delay (Classification)	Several Hours
(Kalliguddi and Leboulluec, 2017)	Decision Tree, <b>Random Forests</b> , Multiple Linear Regression	Flight Delay (Regression)	Several Hours
(Chakrabarty, 2019)	Gradient Boosted Decision Tree	Flight Arrival Delay (Classification)	Several Months
(Horiguchi et al., 2017)	Random Forests, <b>Neural Networks</b> , XGBoost	Flight Delay (Classification)	5 Months, 1 Week, 1 Day
(Gopalakrishnan and Balakrishnan, 2017)	Decision Tree, <b>Neural Networks</b>	Air Traffic Delay (Regression)	Several Hours
(Lambelho et al., 2020)	Random Forests, Neural Networks, <b>LightGBM</b>	Flight Delay and Cancellation (Classification)	6 Months
(Rebollo and Balakrishnan, 2014)	Random Forests	Air Traffic Delay (Classification)	2 Hours
(Chen and Li, 2019)	Random Forests	Flight (Classification)	Several Hours
(Kim et al., 2016)	Neural Networks	Flight Delay (Classification)	Several Hours
(Alonso and Loureiro, 2015)	<b>Neural Networks</b> , Decision Tree	Flight Delay (Classification)	Several Hours

### 5.4.1. Decision Tree

A Decision Tree (DT) is a non-linear classifier that has no predefined structure. The model is built from root to leaf nodes. It sequentially splits the input data into unique regions, with true or false questions at each node. The structure of the tree grows according to the structure and complexity of the data. Essentially, at each node of the DT, a decision is made, splitting up the training data in multiple, smaller subsets (Migut, 2019). At each node, during tree construction, the goal of the true false question is to produce the purest possible labels, or in other words, remove prediction uncertainty. The real challenge is to determine which attribute to ask what question about in a certain node and when. Metrics like Gini-impurity and entropy provide a way to quantify the impurity or uncertainty at a given node (Kuhn and Jamadagni, 2017).

### 5.4.2. Random Forest

Random Forests (RF) are composed of multiple Decision Trees, hence it is called an ensemble method. A large group of uncorrelated trees is assembled, after which they are averaged, reducing the variance. All trees in the group are noisy but unbiased. Each tree carries out a class vote, after which the RF will classify using the majority vote (Choi et al., 2016).

### 5.4.3. Neural Networks

A Neural Network (NN) consists of multiple layers of neurons, stacked together in order to produce a final output. The first and last layer are called the input and output layer and all layers in between are called hidden layers. All neurons in the layers have activation functions, that are fired (activated) when a certain threshold is reached. Popular activation functions are *ReLU*, *Tanh* and *Sigmoid*. The aim of the NN is to learn

and set the network parameters, which are composed of the bias and weights of every layer, in order for the outcome to be equal to the groundtruth (Kuhn and Jamadagni, 2017). The term Deep Learning comes from Deep NN's, which essentially is a NN with multiple hidden layers, creating more 'depth'. Figure 5.5 shows an example of a simple NN. It has one hidden layer with four neurons and uses three features as input layer. The output is a simple binary classification of whether the flight is delayed or not.

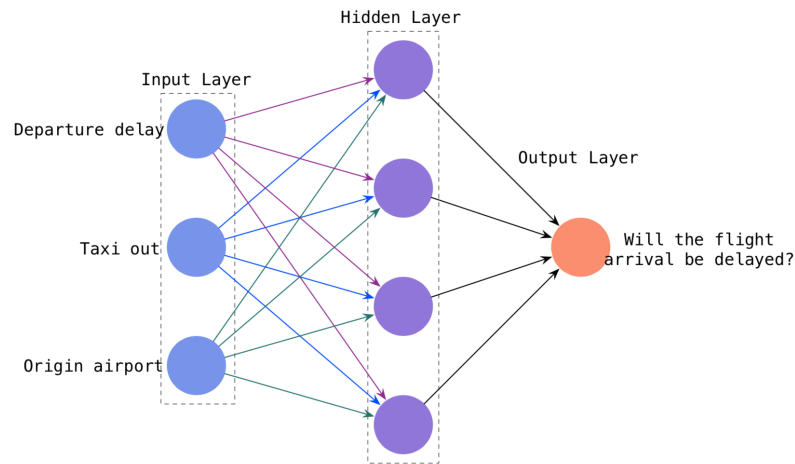


Figure 5.5: An example of a simple Neural Network from (Kuhn and Jamadagni, 2017).

#### 5.4.4. k-Nearest-Neighbours

In order to classify a certain test sample, the non-linear kNN algorithm will define a space centered around that specific sample point, containing precisely the k nearest training points. Subsequently, kNN will check the class labels of these training points. Finally, the class label obtaining the largest vote will be assigned to the test sample. kNN is proven to be most efficient in highly irregular decision boundaries (Choi et al., 2016).

#### 5.4.5. Logistic Regression

Logistic Regression (LR) is a classification algorithm that is similar to linear regression. However, instead of fitting a line to data, LR fits an S-shaped curve, or 'logistic function' (also known as Sigmoid function) to the data. This particular function is  $g(z) = \frac{1}{1+e^{-z}}$ . According to the work done in (Kuhn and Jamadagni, 2017), translating the equation to the one used for LR, one gets the following hypothesis equation  $h(x) = \frac{1}{1+e^{-w^T x}}$ . Here,  $w^T x = w_0 + \sum_{j=1}^n w_j x_j$ . The  $w$  parameter that describes the training data can be found using maximum likelihood estimation and gradient ascent (Ng, 2011).

#### 5.4.6. Boosting Algorithms

Gradient Boosted Decision Tree, AdaBoost, XGBoost and LightGBM are essentially all boosting algorithms. Boosting is, just like Random Forests, an ensemble method. Multiple prediction values are produced on an iterative basis, after which the weighted average of all the values determines the final prediction value. At each step, a new classifier is called upon, boosting the ensemble performance. Important to note, is that boosting takes weak learners, learns from the errors and tries to correct them in successive stages (Manna et al., 2017).

# 6

## Research Approach

This chapter will address the subject of the research approach. After having read a large body of topical literature, a considerable amount of knowledge on the topic of the thesis has been gained. The next important step is to identify what parts of the knowledge are relevant for the thesis. Also, which gaps in the current body of knowledge are present and what combination of techniques or methods should be combined to fill up a certain part of the knowledge gap. Hence, firstly this knowledge gap will shortly be addressed. Afterwards, the scope of the thesis will be discussed, setting the boundaries and choosing appropriate topics to be researched. Subsequently, the research question and objective will be defined and explained. Then, the research framework will be established, in which the data description and methodology will be treated. In this research framework, the parts of the research that will be varied in order to introduce some different research scenarios to compare will also be decided upon. Finally, the research planning will close this chapter, addressing the different phases and milestones of research. A detailed Gantt Chart will also be a part of this final section.

### 6.1. Knowledge Gap

From the literature reviewed in this report, some gaps in the body of knowledge can readily be distinguished. Firstly, it is clear that most of the papers deal with flight *delay* prediction. There is one exception that incorporated cancellations as a small part (e.g. (Lambelho et al., 2020)), however there is no single research paper that solely considers the prediction of flight cancellations using machine learning algorithms. This makes this research quite interesting given its uniqueness in the topic of cancellations. The elements of combining flight schedule data with weather data and different prediction adds to the knowledge gap of cancellation predictions.

### 6.2. Scope

A lot of knowledge has been gained during this literature review and now some choices need to be made in order to give the thesis form. This section will define the boundaries of the research and the appropriate topics that will be included in the work. The scope will be largely based on the knowledge gap, defined in the section above.

The actual scope, i.e. the set-up for the thesis that will be considered, is the following. The research will address the prediction of flight cancellations of individual flights. This will be done using machine learning algorithms and centered around a European Airport. The data that will be used to train and test the model, will consist of flight schedule data, combined with weather data, since it is evident from the literature that this data type is of great importance for cancellation (and delay) prediction. Also, prediction horizons of hours to days before the flight will be utilised. Now, the final research questions and objectives can be formulated.

### 6.3. Research Question and Objective

This section will cast the research scope into some research questions, which are eventually needed to be answered by the thesis work. Also, a reserach objective will be defined.

### 6.3.1. Research Question

The main research question identified for this research is stated below:

***Which machine learning algorithm, trained with historical flight schedule and weather data, produces accurate flight cancellation predictions, on prediction horizons of several hours to days before the flight?***

This question clearly frames the central challenges associated with the research. It defines the knowledge required in order to be successful in the research, as well as the data that needs to be gathered. Specifically, this is knowledge related to binary classification machine learning algorithms, data analysis and accuracy evaluation procedures. The data required is operational flight schedule data, which will be obtained from Amsterdam Airport Schiphol and weather data, obtained from KNMI. Additionally, several sub-questions can be formulated to elaborate the on the fundamentals of the main research question.

- ***Which data features are selected in order to achieve the best performance?***

This sub-question addresses an important challenge early-on in the data analysis. As the data has a high number of features and is highly dimensional, the computational load and internal complexity of the algorithms in the model can become very high and non-reliable results may be produced. Therefore, it is of great value to search for, select and extract the most relevant data features, based on their correlation with the target (cancellation) and their internal correlation.

- ***What are the values of the performance indicators after training?***

When the model has been trained on the historical data, it is important to assess the predictive quality or performance of the trained model. This question ensures that the performance evaluators are checked.

- ***What are the values of the model's performance indicators after testing on prediction horizons of hours to days before the flight?***

As the previous sub-question addressed the model performance after training, this question fires up the need for investigating the performance of actual prediction. A real-life, unseen dataset containing planned flight schedules and weather forecasts in the future will be used to execute this part of the research.

- ***Which classifier performs best in training and prediction?***

The *no free lunch theorem*, states that in machine learning there is no single algorithm that outperforms others when testing over all possible problems Flach (2012). This implies that the way each algorithm performance is highly dependable on the type, size and structure of the data it is subject to. Hence, it is purposeful to evaluate the performance of this model using multiple algorithms and compare them both after training and after prediction on the different time horizons.

### 6.3.2. Research Objective

Now that the research questions have been formulated, it is time to state the main objective of the research. This will put the work in perspective within the larger picture of industry relevance and contribution to the current body of knowledge. It will encompass the essentials that are aimed to be reached by the end of the research. This objective is:

***To develop a machine learning algorithm that can predict flight cancellations using several prediction horizons from hours to days before the flight.***

Breaking down this objective results in multiple sub-objectives to be reached. The first one is to pre-process and prepare data for it to be used in the machine learning algorithms, by cleaning, sampling, normalising, encoding and by incorporating cancellation related features. The second sub-objective is to extract the most important features from the data by performing an analysis on their Pearson's Correlation Coefficient. The third and final sub-objective is to determine the best algorithm for each scenario by evaluating each model's confusion matrix and ROC curve on unseen data.

## 6.4. Research Framework

Now that the scope and research questions are identified, the proposed data and methodology for the thesis can be formulated.

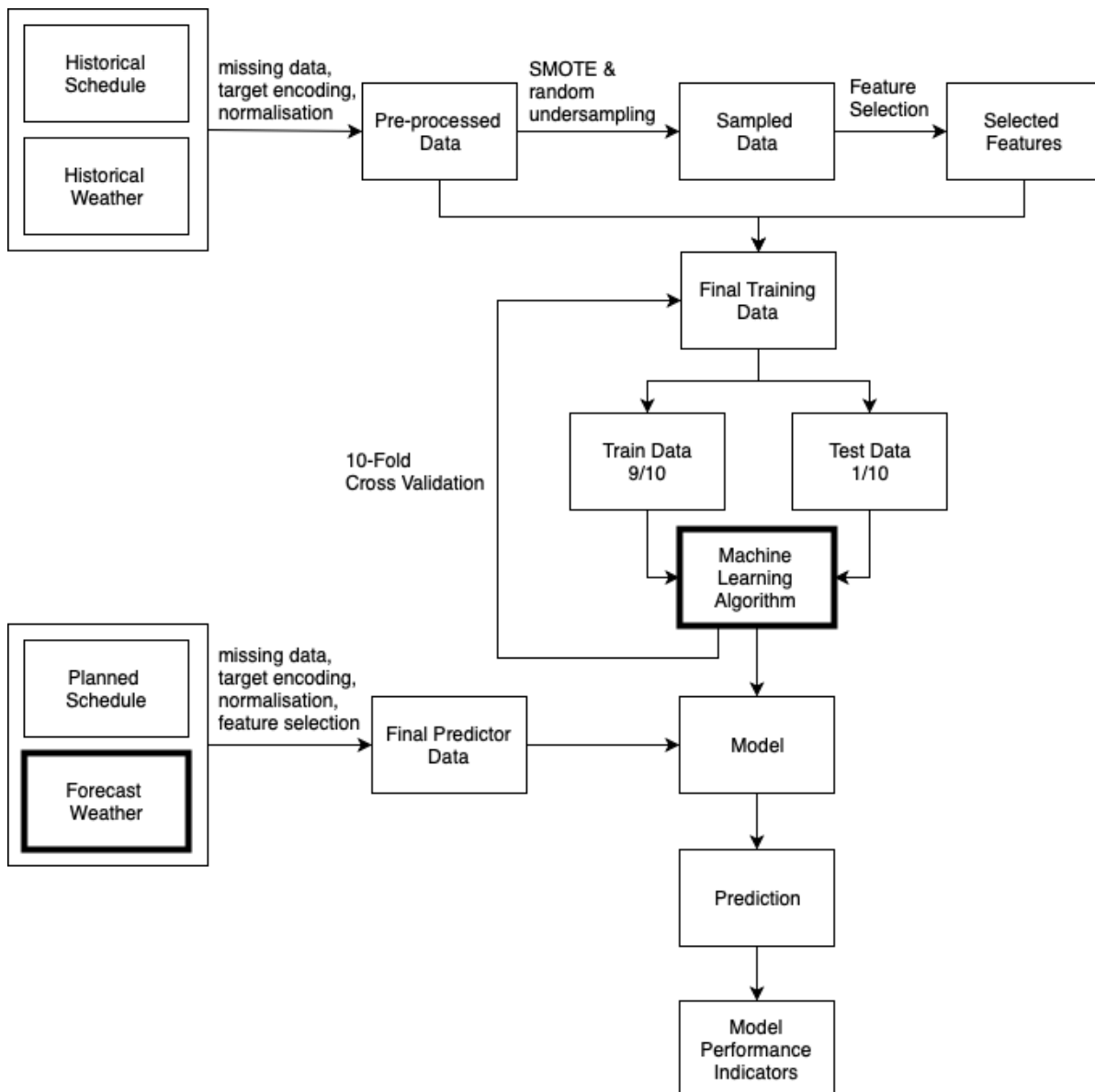


Figure 6.1: Thesis machine learning model flow chart.

### a) Data Description

First of all, the data sources. The flight schedule data will be obtained from Amsterdam Airport Schiphol operations and the weather data will be obtained from Koninklijk Nederlands Meteorologisch Instituut (KNMI). The data will date back 5 years. A part of the historical data will be set aside to function as 'forecast data'. This will likely be a small set of one month.

### b) Methodology

The entire model flow chart and methodology is visualised in a model flow chart, see Figure 6.1. For data pre-processing, the following techniques will be utilised. The target encoding technique seems the most suitable, given the high cardinality of the data. Furthermore, normalisation to a 0-1 scale will be performed. Sampling will be done using the popular SMOTE and random undersampling technique. Finally, features will be analysed and selected using Pearson's Correlation Coefficient. As suggested in literature, the order will be first sampling and then feature selection, after which the training will be done with the *unsampled* dataset (in which only the selected features are present).

10-Fold Cross validation will be utilised during training and three different supervised learning, binary

classification algorithms will be used and compared. Firstly, a linear algorithm will be used, namely Logistic Regression. Subsequently, a somewhat non-linear algorithm is chosen, namely Random Forests. The third algorithm that will be used is the non-linear k-Nearest-Neighbours. The algorithm block in the diagram is highlighted in bold since this is a varying block, in which the scope is not necessarily limited to one algorithm, but multiple are chosen. As, during the research, it appears that another algorithms is highly interesting, it might be opted to add or replace one. The forecast weather block is also highlighted in bold, since this is, just like the algorithm block, a variation block. The prediction horizon for the weather will vary between hours to days before the operation. The research will start with 1 hour before, one day before and 10 days before, however, if it seems other timings are of more interest, they might still be added or altered. Since there is only historical weather data available, the forecast data will be approximated by taking several averages of the data. This way, some kind of uncertainty is introduced, just like forecast data would have uncertainty. Finally, the model will be evaluated using the popular evaluation techniques based on the confusion matrix (accuracy, recall and precision) and the ROC and AUC.

In conclusion, nine different scenarios are there to be tested at the start of the research, with optional expansion or alteration. These are combinations of the three classification algorithms with the three prediction timings.

## 6.5. Research Planning

The research planning for this thesis is based on the Air Transport Operations Master Thesis Procedures. The planning with all relevant phases is depicted in Figure 6.2. The first phase is the literature study, in which relevant literature is identified, research questions are defined, research methodologies are established and project planning is performed. The total duration of this first phase is approximately 8 weeks. It is divided into three smaller phases, the first one of which takes about 4 weeks and is centered around reading papers and books. The second one takes approximately 2 weeks and focuses on comparing and contrasting the different papers. The third one is also 2 weeks and consists of the writing of the literature review. The literature review is concluded the handover of the report and a thesis kick-off meeting.

The second large phase is called the initial phase and is set to take approximately 3,5 months (or about 14 weeks). In this phase the actual work will be performed. It will consist of the initiation of the research, data analysis, model development, etc. This phase is concluded with a midterm presentation and review, in which a basic, working model should be presented.

The third phase is called the final phase and will take approximately 2,5 months (or 10 weeks). It is divided into two smaller phases, the first one of about 6 weeks, in which the model is further developed, tested and validates, more scenarios are experimented upon, etc. The second one is about 4 weeks and is there solely to write the thesis report and paper. This phase is concluded with a green light meeting. After this meeting there is at least one month before the final defence of the thesis, after which the degree will be handed out. The more detailed variant of the research planning can be found in the Gantt Chart in Appendix A.

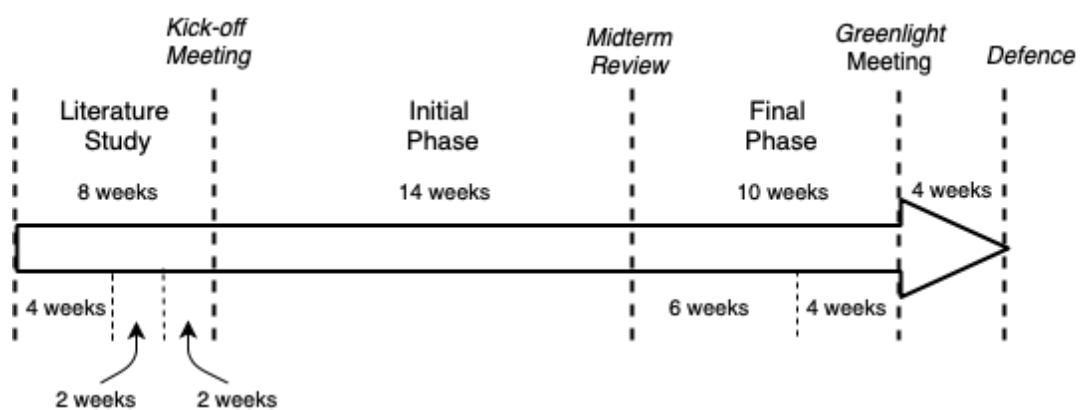


Figure 6.2: Research planning flow diagram.

# 7

## Conclusion

The aim of this literature review was to get familiar with the state-of-the-art related to the prediction of flight cancellations using machine learning algorithms. All the knowledge gained during thorough reading of scientific papers, books and articles was summarised, contrasted and compared. It can be concluded that there is a large body of knowledge about the prediction of flight statuses with machine learning, more specifically flight delays. However, it is also clear that there is very little work done on the level of flight cancellations themselves. Additionally, a general methodological flow or work trend was identified to be common in most topical literature. Three main blocks, namely data management, cancellation or delay behaviour and machine learning, are common thread to build a successful models. This flow was also used as a backbone to construct this literature review.

In data management, it seems important to balance out imbalanced data. Since flight cancellation data will highly likely be out of balance, it will be necessary to sample the data to balance it. Also, feature selection seems to have a beneficial effect on the prediction performance of the models, since not all features will add prediction value to the model. Furthermore, since machine learning algorithms cannot handle categorical data, encoding is essential. Machine learning essentially splits the final data into a training and test set, learns the training set with selected classifiers and then tests the performance in the test set. After assessing the literature and defining the knowledge gap, the following research question has been defined:

*Which machine learning algorithm, trained with historical flight schedule and weather data, produces accurate flight cancellation predictions, on prediction horizons of several hours to days before the flight?*

This questions helps identify the scope of the research, which is the prediction of individual flight cancellations in a European airport with machine learning algorithms, using flight and weather data and a prediction horizon of hours to days before the flight.

The general methodology of the work that will try to answer the research question, was decided to take the following form, based on methodologies taken in literature. Historical flight schedule and weather data will be pre-processed and sampled with SMOTE, after which features will be extracted. Also, some specific cancellation determinants have been identified and they will be integrated into the feature set. The final data will be fed into Logistic Regression, Random Forests and k-Nearest-Neighbours classifiers. 10-Fold Cross Validation will be performed during training and the model will then be tested on an unseen set of planned flight schedule and weather forecast data. Confusion Matrices and AUC will be used in order to evaluate the performance of the model.

This research is definitely of interest for the aviation industry and airport industry, since such models enable the airport to identify unforeseen flight cancellations in advance and thereby allow them to alleviate their negative impact on airport operations. This thesis could initiate more research that could eventually lead to the operational benefits as stated above.

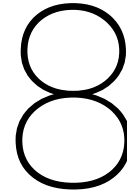




# III

## Elaborations on Thesis Work





# Imbalance

In this chapter some elaborations related to the imbalanced nature of the data for the model will be commented on. In particular, the algorithm for the SMOTE technique will be explained more in detail.

As the datasets are highly imbalanced, it is of high interest to investigate how to mitigate this imbalance and if this mitigation has any positive effect on the model performance. In the literature study it was found that SMOTE is highly suitable for dealing with imbalanced datasets. Since this technique is used in the thesis research, its working principles and theoretical background will be explained here, by means of the pseudocode, retrieved from the original SMOTE paper Chawla et al. (2002). The pseudocode with the explanation can be seen in Figure 8.1.

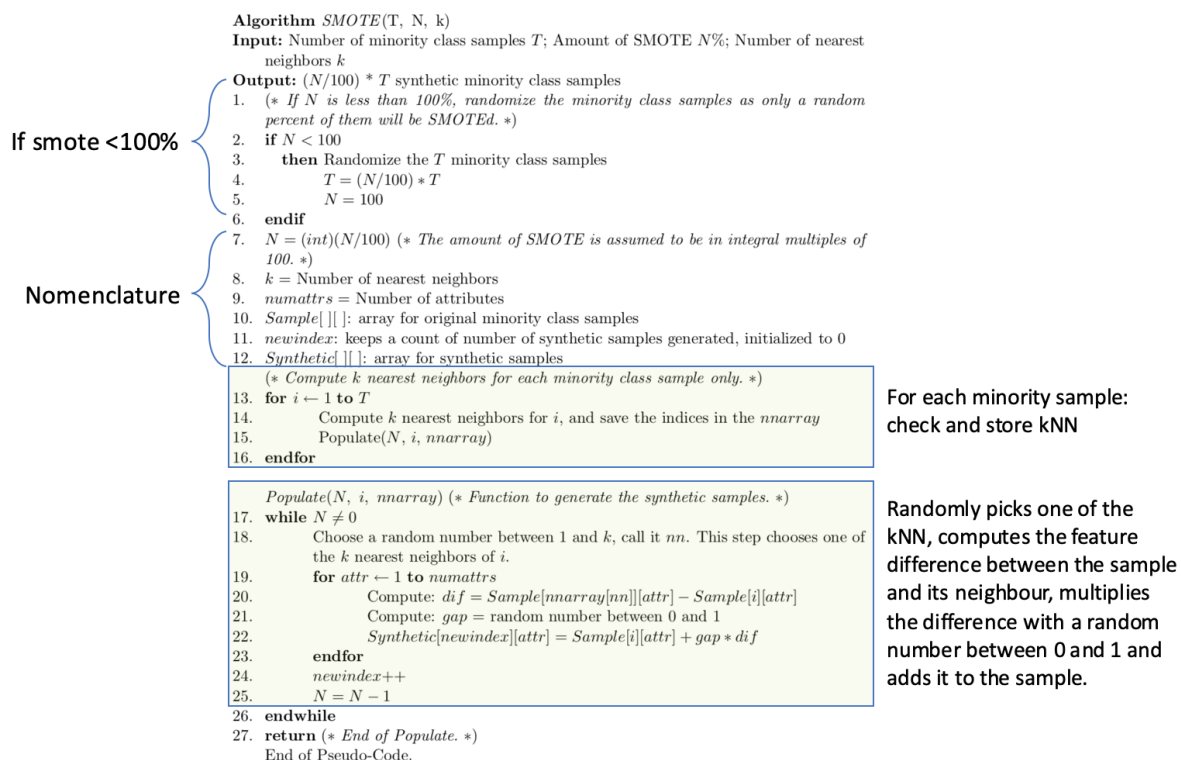


Figure 8.1: Pseudocode of the SMOTE algorithm, explained (Chawla et al., 2002).

The top part of the code is only applicable when SMOTE is applied to <100%. In that case, only a random percentage of the minority class samples will be SMOTEd. The second part of the pseudocode covers nomenclature for the rest of the algorithm. The real interesting part starts after the nomenclature. In lines 13-16 the algorithm will check and store the *k* nearest neighbours, specified earlier in the nomenclature. These neigh-

bours will play an important role in the next part of the algorithm, which starts at line 17. There, the algorithm will randomly choose one of these  $k$  neighbours and compute the difference between each of the features of the original sample and its randomly picked neighbour. This difference is multiplied with a random number between 0 and 1 and subsequently added to the original sample. This now forms a new, synthetically created sample and this process is iterated until the amount that has been specified, is reached.

The idea behind the second sampling technique used in the research, Random Undersampling (RUS), is that it randomly will remove samples from the majority class. Since this is much more straightforward compared to SMOTE, it does not require any more attention.

# 9

## Features

This chapter will elaborate on the extra features added to the base dataset, data that has been researched but eventually not added and the details of the feature selection method.

### 9.1. Additional Features

#### Alliances

Two features were added to the cancellations dataset, namely an alliance indicator (indicating whether an airline was part of an alliance, yes or no) and the alliance name itself. In order to know whether airlines were part of an alliance, a data was used from an online source Pointshogger (2019). The alliance information on that website was updated on May 2019 at the moment of consultation. Subsequently, a list of airlines with their respective alliances was exported from that source, after which they could be easily added to the base dataset. The reasons for adding these extra features were findings in a research by Alderighi and Gaggero (2018), who researched to effects of alliance memberships on flight cancellations.

#### Frequency and Market Share

In a research by Rupp and Holmes (2006), it was uncovered that competitiveness on routes might have an influence on flight cancellations. Therefore, this competitiveness was translated into market share and frequency features and added to the base dataset. Both features monthly averages and are defined in equations 9.1 and 9.2. Here,  $F$  is frequency,  $MS$  is market share,  $OD$  is Origin-Destination pair,  $m$  is month,  $y$  is year and  $A$  is airline.

$$F(OD_{m,y}) = \sum OD_{m,y} \quad (9.1)$$

$$MS(A_{OD,m,y}) = \frac{\sum A_{OD,m,y}}{OD_{m,y}} \quad (9.2)$$

#### Distance

In the research by Lambelho et al. (2020), a feature called 'Distance' was included. As this seemed a rather useful data feature, it was decided to include it in this research as well. It is defined as the great circle distance between the origin and destination airport. It was approximated using the haversine formula, assuming the Earth is a perfect sphere. The derivation goes as follows. The starting point for the derivation is the central angle of two points on a sphere,  $\theta$ , which is defined as:

$$\theta = \frac{d}{r} \quad (9.3)$$

Here,  $d$  is the great circle distance between two airports, which is what needs to be found in the end, and  $r$  is the Earth's radius. This can be rewritten in to get  $d$  as

$$d = \theta \cdot r \quad (9.4)$$

Now, the haversine of the central angle  $\theta$  is defined as:

$$hav(\theta) = hav(\phi_2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)hav(\lambda_2 - \lambda_1) \quad (9.5)$$

Here,  $\phi_1$  and  $\phi_2$  are the lateral coordinates of airport 1 and airport 2 and  $\lambda_1$  and  $\lambda_2$  are the longitudinal coordinates, all in radians. The haversine of an angle  $\alpha$  can also be defined as:

$$hav(\alpha) = \sin^2\left(\frac{\alpha}{2}\right) = \frac{1 - \cos(\alpha)}{2} \quad (9.6)$$

Now, combining equations 9.4, 9.5 and 9.6 yields:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (9.7)$$

The lateral and longitudinal coordinates of the airports were obtained through OurAirports (2020).

### Final Flight of the Day

In addition to competitiveness, Rupp and Holmes (2006) also investigated the influence of a final flight of the day on the flight cancellations. Therefore, it was decided to incorporate this as a binary feature in the dataset. A final flight of the day was therefore defined as a flight that arrived or departed between 22:00 and 24:00 and labelled with 1 if it arrived or departed within that timeframe and 0 if it did not.

### Seats

As the feature 'Seats' was included in the final set of Lambelho et al. (2020), to whose research this thesis quite significantly leans towards, it was included into the base data as well. Approximate seat number per aircraft type were obtained from Seatguru (2020) and linked to their respective aircraft type in the base dataset.

### Eurocontrol Features

A database was provided by AAS containing information on restrictions and regulations within the European airspace due to certain events. These events could occur in a (zone of) aerodrome(s) or a (zone of) airspace(s). The data included different features such as event type (weather, air traffic control capacity, airport capacity,...), time info (start, end, duration, notice before the start), air traffic flow management delay info and location info. This info was particularly interesting for the flight cancellation predictions, as cancellations are often influenced by unforeseen events. When analysing the database, it is essential to look at the elements that could be useful for this research. These are in particular events at AAS, events with at least 24hours notice (for a 1 day prediction horizon) and the event type. After analysis, it was clear that there was no single event in AAS that had more than 3 hours of notice (on average between 0 and 200min notice), which means that there was no sufficient information available to make the 1 day prediction horizon. Therefore, it was decided not to include this database in any further steps of this research.

## 9.2. Feature Selection

As the combined datasets with flight operational data and weather data contains a high number of total features, it would be of interest to check the most important features for the machine learning algorithms, as some features might be irrelevant or redundant for the classification task. Additionally, the high number of features might raise computational complexity. Here, the feature selection method will be explained in a little more detail.

First, the data will be assembled and pre-processed, which is essential before the feature selection is performed. Afterwards, the dataset is split up and the target variable (cancelled/delayed) is isolated. Then, a correlation matrix is set up, using Pearson's Correlation Coefficient ( $\rho$ ). Figure 9.1 visualises the definition of the correlation coefficient. This correlation matrix shows the correlation (between -1 and 1) between each feature and the target, but also the inter-correlation between all features themselves. The correlation is found using the Python Pandas function `DataFrame.corr()`. Mind that  $\rho$  is different from  $R^2$ , which is defined as the quality of a fit (how good  $y$  is explained by  $x$ ). By definition,  $R^2 = \rho \cdot \rho$ . Hence,  $\rho$  can be negative,  $R^2$  cannot.

From now on, absolute correlation will be utilised, which is the absolute version of the regular correlation (which can be both negative and positive). When the correlation matrices are obtained for all 3 cases (cancellations, arrival delay and departure delay), it is observed that the average absolute correlations are quite

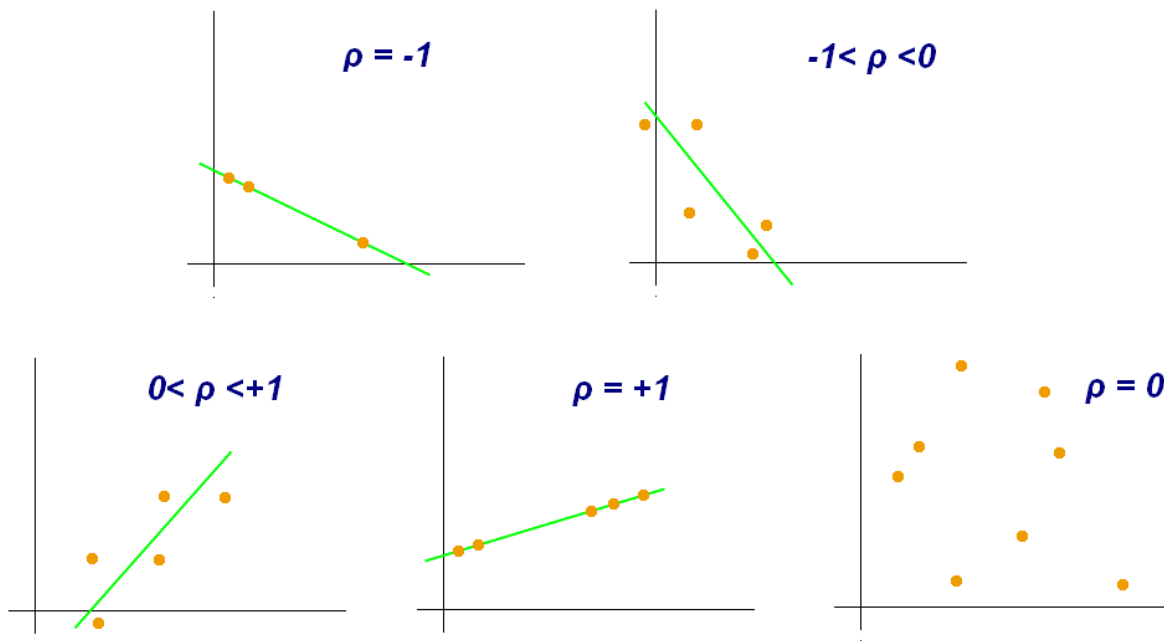


Figure 9.1: Visualisation of the Pearson Correlation Coefficient (Kiatdd, 2012).

low (mostly below 0.3). Subsequently, it is necessary to select a threshold for selecting the feature for the final dataset. After careful examination of the correlation matrices, the threshold of 0.1 is chosen as selection threshold. In other words, all features with an absolute target correlation above 0.1 are selected, the others are not. This specific threshold was chosen by trial and error, while evaluating how many features would be selected at a certain threshold and how they were spread around this threshold. The same analysis goes for the inter-correlation. When two features are selected that have an inter-correlation (a high correlation between these two features), only one can be selected for the final dataset, in order to avoid multicollinearity. For the cancellations, this threshold is set at 0.8, whereas for the delays it is set a bit lower, at 0.7.

The selected features with their corresponding correlation with the target feature can be seen in Tables 9.1, 9.2 and 9.3 for cancellations, departure delay and arrival delay respectively. Please note that there is one feature, namely 'Month' for arrival delay, that has an absolute correlation below 0.1. Since both 'Time' and 'Month' were selected for the departures and 'Time' was already selected for the arrivals, it seemed necessary to also include 'Month', for the sake of consistency and completeness in that dataset.

Table 9.1: Selected features for cancellations with explanation and correlation with the target 'cancelled'.

Feature	Explanation	Correlation with Target
Airport	Origin or destination airport of the flight	0.2
Flight Number	Unique flight number of the flight	0.35
Country	Origin or destination country of the flight	0.19
Airline	Airline company operating the flight	0.27
Servicetype	Category of the commercial flight	0.26
AC Registration	Registration number of the aircraft operating the flight	0.34
Handler	Apron handler, handling baggage, fuel,...	0.2
Wind Speed	Windspeed at origin/destination	0.11
Pressure	Air pressure at origin/destination, reduced to mean sea level	-0.11
Visibility	Horizontal visibility at origin/destination	-0.1
Snow	Indicator is snow presence at origin/destination	0.13

Table 9.2: Selected features for departure delay with explanation and correlation with the target 'delayed'.

Feature	Explanation	Correlation with Target
Flight number	Unique flight number of the flight	0.33
AC registration	Registration number of the aircraft	0.27
AC type	Type of the aircraft operating the flight	0.16
Handler	Apron handler, handling baggage, fuel,...	0.12
Airline	Airline company operating the flight	0.17
Destination airport	Destination airport of the flight	0.2
Daily visits	Number of times this route is operated per day	-0.11
Month	Month in which the flight is operated	-0.1
Time	Time at which the flight is operated	-0.13
Total arr (past hr)	Total number of flights that have arrived in the past hour	0.12
Total dep (past hr)	Total number of flights that have departed in the past hour	0.14
Wind gust speed (origin)	Maximum wind speed at the origin	0.13
Temperature (origin)	Temperature at the origin	0.1
Temperature (destination)	Temperature at the destination	0.12
Total arr delay (past hr)	Total minutes of arrival delay in the past hour	0.21
Total dep delay (past hr)	Total minutes of departure delay in the past hour	0.31
Arrival Delay	If the flight had a delay when it arrived	0.37

Table 9.3: Selected features for arrival delay with explanation and correlation with the target 'delayed'.

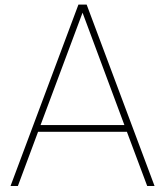
Features	Explanation	Correlation with Target
Flight number	Unique flight number of the flight	0.34
AC registration	Registration number of the aircraft	0.28
AC type	Type of the aircraft operating the flight	0.17
Handler	Apron handler, handling baggage, fuel,...	0.13
Origin airport	Origin airport of the flight	0.22
Month	Month in which the flight is operated	-0.053
Time	Time at which the flight is operated	-0.15
Wind gust speed (destination)	Maximum wind speed at the destination	0.15
Total arr delay (past hr)	Total minutes of arrival delay in the past hour	0.29
Total dep delay (past hr)	Total minutes of departure delay in the past hour	0.24



# Bibliography

- S. M. Al-Tabbakh, H. El-Zahed, et al. Machine learning techniques for analysis of egyptian flight delay. *Journal of Scientific Research in Science*, 35(part 1):390–399, 2018.
- M. Alderighi and A. A. Gaggero. Flight cancellations and airline alliances: Empirical evidence from europe. *Transportation Research Part E: Logistics and Transportation Review*, 116:90–101, 2018.
- H. Alonso and A. Loureiro. Predicting flight departure delay at porto airport: A preliminary study. In *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, volume 3, pages 93–98. IEEE, 2015.
- L. Belcastro, F. Marozzo, D. Talia, and P. Trunfio. Using scalable data mining for predicting flight delays. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):5, 2016.
- N. Chakrabarty. A data mining approach to flight arrival delay prediction for american airlines. *arXiv preprint arXiv:1903.06740*, 2019.
- N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- J. Chen and M. Li. Chained predictions of flight delay using machine learning. In *AIAA Scitech 2019 Forum*, page 1661, 2019.
- S. Choi, Y. J. Kim, S. Briceno, and D. Mavris. Prediction of weather-induced airline delays based on machine learning algorithms. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–6. IEEE, 2016.
- M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156, 1997.
- Eurocontrol. Network manager annual report. 2018a. URL <https://www.eurocontrol.int/publication/network-manager-annual-report-2018>.
- Eurocontrol. European aviation in 2040, challenges of growth. *Annex 1*, 2018b. URL <https://www.eurocontrol.int/publication/challenges-growth-2018>.
- P. Flach. *Machine Learning*. Cambridge University Press New York, 2012.
- K. Gao, T. M. Khoshgoftaar, and A. Napolitano. Combining feature subset selection and data sampling for coping with highly imbalanced software data. In *SEKE*, pages 439–444, 2015.
- K. Gopalakrishnan and H. Balakrishnan. A comparative analysis of models for predicting delays in air traffic networks. ATM Seminar, 2017.
- P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi. Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, 83(2):83–90, 2006.
- Y. Horiguchi, Y. Baba, H. Kashima, M. Suzuki, H. Kayahara, and J. Maeno. Predicting fuel consumption and flight delays for low-cost airlines. In *Twenty-Ninth IAAI Conference*, 2017.
- A. M. Kalliguddi and A. K. Leboulluec. Predictive modeling of aircraft flight delay. *Universal Journal of Management*, 5(10):485–491, 2017.
- Kiatdd. Wikimedia file:correlation coefficient.png. 2012. URL [https://commons.wikimedia.org/wiki/File:Correlation\\_coefficient.png](https://commons.wikimedia.org/wiki/File:Correlation_coefficient.png).
- Y. J. Kim, S. Choi, S. Briceno, and D. Mavris. A deep learning approach to flight delay prediction. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–6. IEEE, 2016.

- A. Klein, C. Craun, and R. S. Lee. Airport delay prediction using weather-impacted traffic index (witi) model. In *29th Digital Avionics Systems Conference*, pages 2–B. IEEE, 2010.
- N. Kuhn and N. Jamadagni. Application of machine learning algorithms to predict flight arrival delays. CS229, 2017.
- M. Lambelho, M. Mitici, S. Pickup, and A. Marsden. Assessing strategic flight schedules at an airport using machine learning-based flight delay and cancellation predictions. *Journal of Air Transport Management*, 82:101737, 2020.
- S. Manna, S. Biswas, R. Kundu, S. Rakshit, P. Gupta, and S. Barman. A statistical approach to predict flight delay using gradient boosted decision tree. In *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–5. IEEE, 2017.
- G. Migut. Machine learning. In *CSE2510, A course by the Faculty of Electrical Engineering, Mathematics Computer Science*, Delft, The Netherlands, 2019. Delft University of Technology.
- R. Mollineda, R. Alejo, and J. Sotoca. The class imbalance problem in pattern classification and learning. In *II Congreso Espanol de Informática (CEDI 2007)*. ISBN, pages 978–84, 2007.
- A. Ng. Cs229 lecture notes. *Standfor University Lecture*, 2011.
- R. Nigam and K. Govinda. Cloud based flight delay prediction using logistic regression. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 662–667. IEEE, 2017.
- M. Noor, A. Yahaya, N. A. Ramli, and A. M. M. Al Bakri. *Filling missing data using interpolation methods: study on the effect of fitting distribution*, volume 594. Trans Tech Publ, 2014.
- OurAirports. Open data downloads. 2020. URL <https://ourairports.com/data/>.
- V. Pai. On the factors that affect airline flight frequency and aircraft size. *Journal of Air Transport Management*, 16(4):169–177, 2010.
- M. Pechenizkiy. The impact of feature extraction on the performance of a classifier: knn, naïve bayes and c4.5. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 268–279. Springer, 2005.
- Pointshogger. List of airline alliances. 2019. URL <https://pointshogger.boardingarea.com/list-of-airline-alliances-updated-may-13-2019/>.
- K. Potdar, T. S. Pardawala, and C. D. Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175(4):7–9, 2017.
- J. J. Rebollo and H. Balakrishnan. Characterization and prediction of air traffic delays. *Transportation research part C: Emerging technologies*, 44:231–241, 2014.
- N. G. Rupp and G. M. Holmes. An investigation into the determinants of flight cancellations. *Economica*, 73(292):749–783, 2006.
- Seatguru. Seatguru: Airline seat maps. 2020. URL <https://www.seatguru.com/>.
- M. Seelhorst. *Flight Cancellation Behavior and Aviation System Performance*. PhD thesis, UC Berkeley, 2014.
- A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara. A review on flight delay prediction. *arXiv preprint arXiv:1703.06118*, 2017.



## Gantt Chart



