ISCTE Business School
Instituto Universitário de Lisboa

A STUDY ON THE PREDICTION OF FLIGHT DELAYS OF A PRIVATE AVIATION AIRLINE

Nuno Óscar Gomes Fernandes

Dissertation submitted as partial requirement for the conferral of Master in Business Management

Supervisor:

Prof. Doutor Carlos J. Costa, Prof. Associado, ISEG (School of Economics and Management), Universidade de Lisboa

September 2017

Summary

The delay is a crucial performance indicator of any transportation system, and flight delays cause financial and economic consequences to passengers and airlines. Hence, recognizing them through prediction may improve marketing decisions. The goal is to use machine learning techniques to predict an aviation challenge: flight delay above 15 minutes on departure of a private airline. Business and data understanding of this particular segment of aviation are revised against literature revision, and data preparation, modelling and evaluation are addressed to lead towards a model that may contribute as support for decision-making in a private aviation environment. The results show us which algorithms performed better and what variables contribute the most for the model, thereafter delay on departure.

Air transportation delays | Machine learning | Private aviation | Prediction

Resumo

O atraso de voo é um indicador fulcral em toda a indútria de transporte aéreo e esses atrasos têm consequências económicas e financeiras para passageiros e companhias aéras. Reconhecê-los através de predição poderá melhorar decisões estratégicas e operacionais. O objectivo é utilizar técnicas de aprendizagem de máquina (machine learning) para prever um eterno desafio da aviação: atraso de voo à partida, utilizando dados de uma companhia aérea privada. O conhecimento do contexto do negócio e dos dados adquiridos, num segmento singular da aviação, são revistos à luz das literatura vigente e a preparação dos dados, a modelização e respectiva avaliação são conduzidos de modo a contribuir para uma ferramenta de apoio à decisão no contexto da aviação privada. Os resultados obtidos revelam quais dos algoritmos utilizados demonstra uma melhor performance e quais as variáveis dos dados obtidos que mais contribuem para o modelo e consequentemente para o atraso à partida.

Atrasos de voo | Aprendizagem de máquina | Aviação privada | Predição

JEL classification system:

C53 Forecasting and Prediction Methods, Simulation Methods

L93 Air Transportation

Executive summary

This study aims to evaluate the accuracy of machine learning (ML) techniques in forecasting air transportation delay, using data of flights between 2014 and 2017 from a private airline company based in Europe.

The first part aims to act as a connecting thread of different contributions from the literature in order to lead the reader to a better understanding of machine learning possibilities. Hence providing guidelines of possible procedures towards reasonable results, thus helping business analysts and managers, ML enthusiasts, to get a first grip hands on ML options using Python through Jupyter mask.

The second part will compare the accuracy of different algorithms for predicting if a flight is delayed or not, and a sensitivity analysis to explicit the relevance of the features used and their contribution. Moreover, then understand how the referred models, their accuracy and limitations can provide a better analysis, predictions and support for decision-making tasks.

The results show us that:

- With knowledge of the Private Airline's business model and access to valid variables and proper data, business analysts have the opportunity to gain a competitive advantage by analysing and predicting flight delays and improve the efficiency of operations. A broad insight of all the relevant variables despite dealing with operational issues flight by flight, in a reactionary way, machine learning process can be used to prevent forecasted operation disruptions;
- Artificial neural networks and logistic regression models prove to work better, and better predictive accuracy for the available extracted and computed variables;
- Through sensitivity analysis, features such as previous flight delay or time of the day of departure are found relevant to the referred algorithms and are contributors to the private airline's delays on departure.

Agradecimentos

A presente dissertação seria impensável sem a contribuição daqueles que aqui anuncio.

Um apreço especial ao meu orientador pelo conhecimento, paciência e o entusiasmo contagiante que deram a cada passo na elaboração deste trabalho um sentido muito próprio e de grande valor. Com o professor presenciei que a virtude do "saber saber" é só para alguns.

Na primeira etapa de mestrado foi também com agrado que me vi envolvido por uma estrutura com visão e valores ao entrar no programa de pós-graduação do INDEG-ISCTE. Aqui, a todos os assistentes, colegas e professores agradeço toda a partilha do que são as valias de uma academia orientada para as competências e para o mercado laboral, e que pude aplicar no desenvolvimento desta dissertação.

À companhia aérea que me disponibilizou os dados para este trabalho, agradeço a oportunidade de poder transformar informação em contribuição para esta área de estudo.

Aos meus pais, e palavras nunca serão suficientes, agradeço todo o apoio, a inesgotável paciência e o carinho, desde sempre. Porque é com eles e por causa deles que aprendo todos os dias.

E aos meus amigos e namorada agradeço todo o apoio e ternura. Cada um tem o seu caminho mas no final fomos nós que partimos o copo.

Index

Figures Index

Table Index

Acronyms List

ANN	Artificial Neural Network
ATA	Actual Time of Arrival
ATD	Actual Time of Departure
AUC	Area Under the Curve
CART	Classification and Regression Tree
CRIPS-DM	Cross Industry Standard Process for Data Mining
EASA	European Aviation Safety Agency
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
IATA	International Air Transport Association
kNN	k-Nearest Neighbour
LDA	Linear Discriminant Analysis
LR	Logistic Regression
ML	Machine Learning
NB	Naïve Bayes
NN	Neural Network
ROC	Receiver Operating Characteristic
SACA	Safety Assessment of Community Aircraft
SAFA	Safety Assessment of Foreign Aircraft
SGD	Stochastic Gradient Descent
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UTC	Universal Time Coordinated

1. Introduction

1.1 Context and motivation

Change is swift at any given moment in the business world. Nowadays, all companies from all industries can create new products and services rooted on data analytics (Davenport, 2013). Thus, analysing a historical business data may prove to be an enhancement opportunity to improve any company (Hazen et al., 2014) and gain a competitive advantage over other competitors - a blue ocean oriented strategy (Kim and Mauborgne, 2004).

As the world changes to Analytics 3.0 (Davenport, 2013), the key consumer of analytics is the business user, a person whose job is not directly related to analytics, but who typically must use analytical tools to improve business processes. Data mining, analytics and business intelligence systems are now improving, bringing close operations and analyses that allows data to be analysed faster and with results quickly reflected in the business course of actions. Mined information is being deployed to broader business areas, which are taking advantage of business analytics in everyday activities in several industries (Kohavi et al., 2002).

Machine learning algorithms tend to be nowadays technically easy to access. One can choose from different platforms and languages to different packages/libraries. Nevertheless, this also raises the risk that they are applied naively, or their output is misinterpreted. The present study aims at converging some of the most common models and present them in a practical way, highlighting the benefits and respective limitations.

On the 261/2004 European regulation (European Parliament, n.d.), it defines rules for compensating passengers in the event of denied boarding, delayed or cancelled flights. In the legislation, it is defined the exact compensation amount independently of the actual ticket price. Whenever one of the referred events are an airline's fault, e.g. technical problems or organizational errors, the airline has to compensate the passenger. In case of an external factor, e.g. bad weather or bird strikes, the airline does not have to pay to the passenger. The referred regulation aims to raise the standards of protection both by strengthening the rights of passengers and to ensure that air carriers operate under equally fair conditions in a liberalised market (Kreibich, 2017).

A delay at departure is defined as when actual time of departure (ATD) is beyond the estimated time of departure (STD). For commercial purposes, there is often a margin of 15 minutes. This is a standard applied throw-out all air transportation airlines and airports. Hence, carriers tend to have an impact on their financial costs when their flight starts to be delayed beyond those 15 minutes, and other operational procedures follow which can overload the management of the flight and with a potential to have an adverse impact on company's brand and client/passenger satisfaction. Kreibich (2017) analyses that the European market size for dealing with denied boarding, delay and cancellation is estimated to be an 11.7 billion Euro market, which considers an annual 26 million eligible passengers, multiplied by an average compensation of 450 Euro. Based on the prevailing provision of 25% per case, the European market is worth approximately 2.9 billion Euro. Further internal surveys in that study showed that the market is by far not exploited as only around 1% of all passengers know about their rights and according (IATA, 2016) with air passenger traffic volume will increase in the near future.

The delay is, therefore, a crucial performance indicator of any transportation system and flight delays cause economic consequences to passengers and airlines; recognizing them through prediction may improve marketing decisions (Boswell and Evans, 1997; Sternberg et al., 2017) as they may influence costs to customers and operational costs to airlines. Hence, delay prediction is of the utmost importance during the decision-making process for every player in the air transportation business.

The Private Airline Company that provides the dataset is a European one, which operates mostly with its wide-body aircraft, for other airlines or travel agencies. One day they can be departing in one of the busiest and important airports in the world and the following day in a third country airport with limited conditions and resources. Their product tends to focus on medium to long-term wet lease or charter contracts worldwide, for long-haul flights. The aim is to supply aircraft with crew, maintenance and insurance (ACMI). In practical terms, the lease can fit the schedule and planning of the client or can be non-schedule flights. In every option, several players influence the preparation and operation of a flight: financial decisions, operating crew, staff availability, airports management, its characteristics and services (ground operations, ATC, etc.), Regulators such as EASA and Civil Aviation Authorities directives, SAFA/SACA Inspections' outcomes, manufacturers, client's requests, hotels booked for crews, transport services from and to the airport, etc. One of the current challenges that every airline faces, and in a private airline the exposure is augmented, is planning and operating under

uncertainty, whether in the context of schedule disruptions and variety of routes flown (Ahmed and Poojari, 2008; Salunke and Deshpande, 2015), and keep a tight control over its operations to ensure compliance with European and any local applicable regulation. The advantages, from the operator's perspective, is that allows a client to avoid an upfront large capital expenditure, thus the residual value risk lies with the operator, a lease can provide flexibility for adjusting capacity and demanding, and the possibility of acquiring an aircraft on short notice via operating lease.

1.2 Objectives

The Private Aviation business model is a unique one, where successful operations come from learning how to operate daily with change. An error in one-step of the chain of events in the preparation of a flight may lead to delays, and subsequent financial expenses and a brand negative impact. Despite noticing the existing absence of works dealing in depth with the overall flight delays analysis of Private Aviation, but aviation, in general, considering features rooted on schedule flight performance, an analysis is carried merging machine learning methods to an aviation challenge:

- Predict flight delay on departure of a private airline (delayed or not delayed more than 15 minutes) where a comparison of nine models is achieved by testing their accuracy;
- As a second objective, a sensitivity analysis is conducted to scrutinize the relevance of the chosen features has on the classification prediction for a flight to be delayed or not more than 15 minutes.

1.3 Methodology

The current approach is intended to serve as a source of information to challenges of business management and application of machine learning techniques on a daily basis can be achieved. Focus meetings were conducted with several stakeholders in the flight preparation process of the Private Airline, to extract, compute and select the best features available so the best fit of the techniques could be possible. Through the modelling stage according to with CRISP for data mining purposes, a cross-validation, run 20 times, is implemented with a stratified k-fold, where both possible outcomes are made equally representative, for the learning training phase, to avoid bias results. When evaluating individually the two best algorithms for the classification, the same cross-validation method was applied, and the several indicators are

retrieved such as accuracy, true positive rate, false negative rate, specificity, sensitivity, ROC/AUC and precision. An exploratory analysis is conducted so the reader may understand better the nature and context of delays and their characteristics, in a singular business segment such as private aviation. For the sensitivity analysis, the best model, logistic regression, and a decision tree based algorithm which provides better insight into its variables are used to retrieve the top rank features that best contribute to the respective model to predict if a flight is delayed or not more than 15 minutes. The results obtained from models' accuracy comparison, individual algorithm evaluation, exploratory and sensitivity analysis are then compared with applicable literature in the discussions chapter. In recent academic dissertations on this subject in the area of business management studies, the code and the techniques are often left aside, focusing more on the steps towards a big data comprehension and organizational need to achieve Analytics 3.0 (Davenport, 2013). For knowledge extraction following CRISP-DM stages, comparison and sensitivity analysis purposes, the python language is used. As a platform the Jupyter Notebook, and python libraries such as spicy, pandas, matplotlib, numpy, scikit-learn and stastmodel. Intentionally, some of the implementation will be along the text or in the annexe as a means to provide a wider guideline of the techniques applied and other try-outs.

1.4 Contribution

Delays are a sensitive subject in the air transportation business, as all the intervening players prior, during and after transporting something or someone by air tend to focus on their respective performance (IATA, n.d.). In practical terms, the main contribution of the present study are as follows:

- Analyse a private airline's operational delays, which are often not the focus of delay analysis throw-out general aviation on-time performance research;
- Expose that machine learning techniques can be accessible to respond to business request and prove valid competitive advantage and a benefit for the sector;
- Using a set of features, methods of pre-processing and nine algorithms, as presented in the subsequent paragraphs, compare the accuracy of predicting the dependent variable of flight delay on departure above 15 minutes of a private airline;
- Using the above set, apply the top individual algorithm with the best accuracy and evaluate them accordingly, employing other relevant performance indicators;

- Run a sensitivity analysis and exhibit the relevant features that may lead to a flight delay in the referred sector.

1.5 Structure

In this dissertation, and to achieve the pre-set objectives, its structure is divided as follows:

- Introduction, with an initial context and motivation of this dissertation, along with its desired goals, the methodology to achieve them and how they can contribute to the related literature and private aviation;
- Review of the applicable literature, with regards to the aviation and business context of flight delays, and to predictive machine learning methods and techniques relevant to this dissertation;
- Methodology in detail with the steps taken. Features used for the model, and how they were selected is explained; how the data is transformed to be readable by the model, and how the comparison is achieved by using data mining standards;
- Results of the comparison and sensitivity analysis of the selected features and models;
- Discussion where the results are put in context of the carrier whose data is being used and compared with previous studies;
- Conclusion and acknowledgement on how the objectives are achieved, what limitations were faced, and aspects of future research.

2. Literature Review

As performance is linked to any business model, flight delay is a major key performance indicator in any airline. On-time performance is for some time a competitive advantage in any company working on air transportation, moreover on airlines (Yimga, 2017).

2.1 Private aviation, delay context and analysis

In a disruptive alike operation such as this one, flight delays are more likely to occur due to a high number of ad-hoc flight request from clients. It is the nature of the business, and the goal is to satisfy clients, its customers and transport safely from a point A to B. As discussed in Laws (1997) mass market destinations to attract sufficient visitors to sustain a developed tourism and visitors need regular access to ACMI flights. Thus most of the high peaks of operational demand are linked to leisure travel market (Buck and Lei, 2004) and during those peak seasons, operations may be working intensively up to 24 hours a day (Williams, 2001).

Ymiga (2017) draws the connection between the recurrent known flight delay of an airline and market choice, where passengers tend to avoid airlines which are associated with delays. On the other end, Deshpande and Arıkan (2012) prove that market share of airlines has a significant impact on the flight schedule and on time probability. This factor comes to a challenge with Private Airline companies when preparing for a flight where often are not a top priority for airport services.

As stated in the 261/2004 regulation, some directives are directly applied to the context of the private carriers whose flight information is being used:

- “Since the distinction between scheduled and non-scheduled air services is weakening, such protection should apply to passengers not only on scheduled but also on non-scheduled flights, including those forming part of package tours”
- “In order to ensure the effective application of this Regulation, the obligations that it creates should rest with the operating air carrier who performs or intends to perform a flight, whether with owned aircraft, under dry or wet lease or on any other basis.”

In a rough summary, the referred legislation states that when an operating air carrier is delayed or reasonably expects a flight to be delayed beyond its scheduled time of departure, passengers

shall be offered by the operating air carrier the appropriate assistance according with the regulation, and receive the applicable compensation.

Forbes (2008) suggests airline prices tend to fall in response to a longer flight delay. Thus, a decrease in quality has a strong negative effect on the market price. Despite this impact being lower in competitive markets where there is more competition rather than in low levels of competition markets, delays always have the potential to injure an airline financially.

A better understanding of delay mechanisms may lead to a better efficiency and robustness of operations and costs. Ionescu et al. (2016) defend that delays are inherently hard to predict in the long-term on a macroscopic level, and delay recording underlies constraints that may lead to underestimation, for example, when predictable delay may have been already prevented by scheduling decisions of an airline. Thus, it is desired to check to which extent the findings of delay analysis may be generalized. Nevertheless, they emphasize that a robust resource scheduling should be achieved through the use of historical information for data-driven detection of delay trends depending on specific relevant spatiotemporal attributes.

As flight delays incur in great costs to airlines (Ferguson et al., 2013; Hansen et al., 2001), trying to validate root causes for them through data analytics and predicting delays is a chance to improve the airline performance and improve data support for future decision-making, and a valid competitive advantage. Sternberg et al. (2017) studied the flight delay problem in different points of view: delay propagation, delay innovation and cancellation analysis. In delay propagation, one studies how delay propagates through the network of the transportation system. On the other hand, considering that new problems may happen eventually, it is also important to predict new delays and understand their causes. Such occurrences fit as delay innovation problems. Finally, under specific situations, delays can lead to cancellations, forcing airlines and passengers to reschedule their itineraries. In this category, researchers focused on cancellation analysis try to figure out which conditions result in cancellations. Hence, the focus of this study is to follow the delay innovation approach, where a classification problem will be answered by supervised learning algorithms, and importance of the features is analysed as its influence on the classification outcome, flight delayed or not delayed more than 15 minutes.

Taylor (1994) states “waiting is a pervasive element of many purchase situations” and assesses the delay experience modelling related delay duration, reason and the degree to which it affects service evaluation. In an empirical examination in an airline service, its results imply that: the

delay affects feelings of anger and uncertainty in passengers and the longer the delay, the more anger and uncertainty. Thus the emotional reactions in turn negatively distress the service evaluation and acceptance for unpunctuality decreases. The relationship between time reliability and the overall evaluation of a service (in this case, a flight) is significant. Another fact withdraw from the referred study is that when passengers were inquired for the reason for the delay, they were most of the time wrong. This may be due to a lack of information presented to them. If delay is not announced by an airline agent, many passengers tend to infer their own. For strategic recommendations, it was conveyed that as delays can affect service evaluations in negatively, organizational management has two kinds of actions: reduce or eliminate delays by operations controlling, or change the consumer's wait experience by perceptions management resulting in less uncertainty and anger. Both options regard a timeless indicator of any industry, i.e. managing expectation (Kotler and Keller, 2011). In this particular context of private aviation, expectations come from both of internal (e.g. managers, crews, etc.) and external sources (clients, passengers, service providers) with an integral influence in their behaviour, satisfaction and loyalty.

On dwelling with a performance indicator of an air transportation company, beyond the process of analysing and forecasting delay, it is essential to occur an applicability of that process along the organizational structure, thus a strategic alignment to identify the value of prediction in order to supply information to answer a decision problem, and deploy specific actions to mitigate it (Poleto et al., 2015). Further than technical tools and methodologies, the overall process should align both the subjective characteristics linked to the decision-makers' perceptions and experiences and the actual context of the problem, removing step by step intuition and increasing data based decisions. Forecast model building is most likely succeed when it is regarded in a broader system context, where constraints, interactions (between company and client) and market plans (between operational manager, client and manufacturer) all have an impact on the final prediction (Fildes et al., 2008). By adopting a wider system attitude, forecasting performance should not only be evaluated by standard error measures but connected to organizational performance measures. Vincent Granville (2015) when addressing the use of historical data, advocates that real-time factor impact any forecasting model, hence those factor should be included in combination with historical data and patterns. Nevertheless, issues arise when there is a need for implementation. Fildes et al. (2008) details further that algorithms which are demanding of data and computation are difficult to put in practice. The processing it needs when applying them to a complex system, sometimes daily within a limited

time window, proves to be intensive. Methods such as support vector machines and artificial neural networks face challenges when trying to be implemented.

Organizational learning and analytics are, therefore, a way to help optimize key processes, function and roles of an airline. And in the private aviation segment, it can be used as a leverage when combining internal and external data, and allow achieve stakeholders' demands, create market advantages and, ultimately, enhance organizational performance by turning information into intelligence("Big opportunities, big challenges," 2014) .

2.2 Machine learning models and algorithms

According with Twagilimana (2006), from the point of view of the majority of data miners, the main data mining tasks are grouped into the following categories:

- Prediction, which consists of building a learning function that predicts based on inputs. If the prediction is a discrete variable with a few values, is called classification; if the prediction is a continuous variable, the task is called regression.
- Clustering where a heterogeneous population is segmented into more homogeneous categories or clusters. Clustering is often done as a prelude to some other form of data mining. For example, in healthcare data, patients with similar diagnoses are grouped together to allow for the detection of deviation in their treatment.
- Summarization is about finding parsimonious summaries of subsets of data.
- Dependency Modelling consists of finding a model that describes significant dependencies between variables.
- Change and Deviation Detection focuses on discovering the most significant variations in the data from previously measured values.

The called "learning" can be categorized as supervised or unsupervised, and sometimes using both for the same objective. Different methods may have different inductive bias, different search strategies, different guiding factors, and different needs regarding the availability of a domain theory. Learning can result in either knowledge augmentation or knowledge (re)compilation (Kelleher et al., 2015).

In a supervised learning method, we have a dataset sample with associated labels (Webb, 2002). In unsupervised learning method, the dataset is not labelled, and we seek to find groups in the

data and the features that distinguish one group from another (Webb, 2002). Unsupervised learning, such as clustering, is where there is only input data (X) and no corresponding output variables. The unsupervised learning goal is to model the data distribution in order to learn more about the data. These are called unsupervised learning because unlike supervised learning there are no correct answers and there is no teacher. Algorithms are left to their own devices to discover and present the interesting structure in the data.

Supervised learning can be further grouped into regression and classification problems (Zhou and Li, 2010):

- Classification - a classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”;
- Regression - a regression problem is when the output variable is a real value, such as “euros” or “weight”;

Hence, due to the nature and characteristics of the data provided the method supervised learning is used, where it is predicted a classification output: delayed or not delayed more than 15 minutes. For future reference, there is distinction between a classification learning algorithm and a classifier. A classification learning algorithm is a general methodology that can be used in a supervised classification problem too, given a specific dataset, learn a specific classifier. Thus, this classifier is the one used to classify new samples with the unknown class assignment (Santafe et al., 2015).

As previously mentioned, supervised learning is the branch of machine learning that is concerned with algorithms that can learn concepts from labelled examples. As an input, the algorithm requires a training set composed of a number of instances that represent the problem being studied, each characterized by a list of relevant features. The task of the algorithm is to build a model that will generate accurate predictions of the labels of future examples (Foulds and Frank, 2010). In practical terms, it uses input variables (x) and an output variable (Y) and an algorithm to learn the mapping function from the input to the output. The goal is when you have new input data (x) that you can predict the output (Y) most close to the actual outcome. It is titled supervised because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. It knows the correct answers as the algorithm iteratively makes predictions on the training data and is continuously amended by the teacher. The process stops when the algorithm achieves an, sometimes already pre-defined,

acceptable level of performance. In common supervised learning, hypotheses are learned from a large number of training instances. Each training set has a label which indicates the desired output of the event described by the instance. In classification, the label indicates the category into which the corresponding example falls into; in regression, the label is a real-valued output such as temperature, height, price, etc. (Zhou and Li, 2010).

As announced before, the overall objective is to gauge prediction models for flight delays of a Private Aviation company through a focus on supervised classification machine learning algorithms. Therefore, to have a better structure for data discovery and avoid blind applications of methods to input data, different methods are defined that help organizations to understand and discover data mining processes (Dunham, 2002; Marbán et al., 2009). This model, help deliver results in time and with cost savings, and in better understanding for the related users. CRISP-DM is considered the popular, meeting both research and industrial needs (Kurgan and Musilek, 2006). Cross-Industry Standard Process for Data Mining is a hierarchical process model where standards for data mining processes are defined. The process is divided into 6 stages: business understanding, data understanding, data preparation, modelling, evaluation and deployment (the latter is a stage not applicable to the current study). The steps were first proposed in early 1996 by a consortium of four companies: SPSS (a provider of commercial DM solutions), NCR (a database vendor), Daimler Chrysler, and OHRA (an insurance company). The last two companies functioned as sources of data and benchmarking studies. The model was officially released, version 1.0, in 2000 (Shearer, 2000; Wirth and Hipp, 2000). As in the research and survey of Kurgan and Musilek (2006), there are plenty of advantages of following the referred standards, resulting in a better reliability of projects and performance of data mining for business decision-making support. In the same survey, from the 6 stages referred, the data preparation is the phase with the most effort and time demand with between 50 and 70% of time spent. Hence, in the previous and subsequent chapters of revision of literature, the information gathering and the steps followed to implement and achieve results encompasses the CRISP-DM stages.

2.2.1 Pre-processing and modelling

Whenever applying machine learning to solving any real-world problem some steps should be taken care of. Data collection and preparation for the learning process is decisive. The quality and the quantity of the data needed is dependent on the selected learning method. Therefore,

data may require, and for performance reasons always is, to be pre-processed before they can be used in the learning process (Zhang and Tsai, 2003). Data pre-processing has become an essential technique in current knowledge discovery scenarios. In a survey by Munson (2012), around 60% rated critically important the preparations of data. Raw data usually comes with many imperfections such as inconsistencies, missing values, noise and/or redundancies. Performance of subsequent learning algorithms will thus be undermined if they are presented with low-quality data (Ramírez-Gallego et al., 2017).

In the present study, the data set will contain different types of variables. It is often a challenge of figuring out how to turn its attributes into distinct values for further processing, using data reduction or projection (Bilalli et al., 2017; Gürbüz et al., 2011), alter the dataset by whether feature is selecting, mapping values to categorical ones, or nominal encoding attributes. Python tools of pandas and scikit-learn (Buitinck et al., 2013) offered several approaches that can be applied to alter the categorical data into appropriate numeric values. Linked to hereby literature revisions, those tools were used to achieve the set goals.

Before implementing classification algorithms, it is recommended that incomplete, noisy, or inconsistent datasets are pre-processed to make the knowledge discovery process easier and more qualified. The most well-known steps for this process are summarization, cleaning, integration and transformations, data and dimensionality reduction, and discretisation (Hsu et al., 2006).

Howley et al. (2006) studied the effects of data pre-processing steps on classifier accuracies and compared the results of classifiers where no pre-processing step was applied and then applied additional techniques, such as normalization or PCA, which lead to better performance, whether on the time of processing and further help the accuracy of the model.

Principal component analysis (PCA), Jolliffe (1986), is applied to reduce the number of variables to a small number of factors that are uncorrelated, removing collinearity characteristic from the dataset (Constantin, 2014). PCA (Jolliffe, 1986; Tipping and Bishop, 1999a, 1999b), although being a well-known dimensionality reduction technique, suffers from the disadvantages of not handling well with high dimensional data and scaling up to large dataset due to its excessive computational complexity.

It is then common sense that complex data often involves a prediction pre-processing step, which at times may be faced as an empirical tuning, avoiding over-fitting model and training dataset(s). The essential problem of over-fitting (Domingos, 2012) is that we would like the prediction task to do well out of sample, but we only fit in-sample. In empirical fine-tuning, we should create an out-of-sample testing inside the original sample. We fit on one part of the data and query which level of regularization leads to the best performance on the other part of the data. For this it's possible to increase the efficiency of this procedure through cross-validation (Kelleher et al., 2015; Refaeilzadeh et al., 2009): where the sample is randomly partitioned into equally sized subsamples (folds). The estimation process then comprises successively holding out one of the folds for evaluation while fitting the prediction function for a range of regularization parameters on all continuing folds. At the end, the parameter with the best estimated average performance is chosen (Mullainathan and Spiess, 2017).

2.2.2 Supervised classification algorithms

Following is presented a summary of the common algorithms that are going to be used for the supervised classification problem and how they work, based on the relevant literature and python library.

2.2.2.1 Logistic Regression (LR)

A logistic regression analysis is a class of conditional probability models used to estimate a relationship between a set of variables (features) describing an entity and the probability that the entity will be in a given final state (Storey et al., 2016). The logistic regression acknowledged as the regression with a twofolded dependent variable is used in a similar way with the Discriminant analysis, however in this case, the independent variable could also be nominal ones (binary or categorical). Logistic regression is commonly used in social sciences as an substitute technique to ordinary least scores (OLS) used in traditional regression models due to often the researchers regarding people behaviours use dichotomous variables instead of continuous variables (Constantin, 2015).

The logistic regression is based on the mathematical notion of "logit", which is the natural logarithm of an odds ratio, where the odds is the ratio of probability of a certain occurrence Y happening (p) to probability of Y not happening (1-p) (Constantin, 2015). The dependent variable can be binary or multinomial. In the latter, the categories of these variables are

transformed into binary ones. Thus, the binary regression models can put into relationship a future specific occurrence with certain current behaviours. A logistic regression also allows its users to determine the relative contribution of each variable on the actual classification.

2.2.2.2 Linear Discriminant Analysis (LDA)

Fisher (Fisher, 1936; Duda et al., 2012; Li et al., 2006) first introduced LDA for two classes and its notion is to convert the multivariate observations **x** to univariate observations **y** such that the **y**'s derived from the two classes are parted as much as possible. If the number of classes is more than two, then a natural extension of Fisher linear discriminant exists using multiple discriminant analysis.

LDA and PCA are a technique for classification of data and dimensionality reduction (Balakrishnama and Ganapathiraju, 1998). LDA differs from PCA from being more towards data classification, and PCA leans on feature classification, hence the latter is used to pre-processing purposes.

Linear discriminant analysis frequently achieves good performances in the tasks of face and object recognition (Li et al., 2006). The basic idea of LDA is to find a linear transformation that best discriminates among classes, and the classification is then performed in the transformed space based on some metric such as Euclidean distance. Mathematically a typical LDA implementation is carried out via scatter matrix analysis (Li et al., 2006). LDA can complete a specification of which is achieved by prescribing the weight vector and a threshold weight. The value x is a measure of the perpendicular distance from the hyperplane (Webb, 2002). In a discriminant analysis of statistics, within-class, between-class, and mixture scatter matrices are used to formulate criteria of class separability (Fukunaga, 2013). Thus, the coefficients of the linear discriminant function are given by the correlation between the desired output and the input X

LDA is basically a tool for classification. It determines the discriminant dimension in response pattern space, on which the ratio of between-class over within-class variance of the data is maximized (Duda et al., 2012). After projection of the data on this linear discriminant dimension, a classification threshold is placed at the midpoint between the two class means. This is equivalent to placing a decision hyperplane orthogonal to the discriminant dimension in response pattern space.

The region and separator line are the defined by linear discriminant function. For a discriminant function of the form of a two-category classifier, it implements the following decision rule: **x** can ordinarily be assigned to either class or can be left undefined, and an equation is defined as decision surface (hyperplane) that separates points (Duda et al., 2012).

2.2.2.3 Classification and Regression Tree (CART)

The CART model represents a typical binary decision tree. Each root node denotes a single input variable (x) and a split point on that variable (numeric ones). The leaf nodes of the tree comprehend an output variable (y) which is used for prediction purposes. The algorithm is based on Classification and Regression Trees by Breiman et al (1984). A CART tree is a binary decision tree that is constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning sample. CART, also known as Automatic Interaction Detection, is a regression tree approach to identify subgroups with different probabilities. Tree models use a sequential process, as Rosenfeld and Lewis (2005) explain, to recognise the predictor variables that best discern groups along the outcome variable of interest. The sample is then divided into two or more branches based on this predictor. Subsequent phases identify the best predictor within each of these branches and this process is repeated until no more variance can be explained with the remaining variables, or some other criterion, has been reached. The culmination point of these branches (nodes) represent subgroups of the original sample that diverge in terms of the probability of the outcome variable. Because the same variables are not necessarily optimal for each branch of the tree, this process concedes interaction effects within the predictor variables that would typically be masked or incomprehensible in a traditional regression approach, deriving a series of decision rules that optimize the discrimination between, for example, flight delay or not delayed.

Trees are a completely different way of partitioning (Breiman, 1984). What is required is that the partition can be attained by consecutive binary partitions based on the different predictors. Once we have a partition in this condition, our prediction is based on the average of the Y's in each partition. We can use this for both classification and regression. Each terminal node gets assigned to one of the classes. A disadvantage in tree creation is how to use the training data to determine the binary splits of X into smaller pieces. The idea is to select each split of a subset so that the data in each of the descendant subsets are more intrinsic to the objective than the data in the parent subset. Decision trees are very "natural" constructs, in particular when the

explanatory variables are categorical (and even better, when they are binary). As disadvantage (Breiman, 1984), when the tree-space is huge we may need a lot of data and we might not be able to find the "best" model at all.

2.2.2.4 K-Nearest Neighbour (kNN) Classifier

K-Nearest Neighbour (K-NN) is a common technique for classifying and clustering data. K-NN is effective, however is often criticised for its run-time growth as k-NN computes the distance to every other record in the data set for each record in turn. Standard k-nearest neighbour (K-NN) is a widely applicable clustering, outlier detection and classification technique that demonstrates high recall accuracy. For classification, K-NN examines those points in a particular data space lying nearest to a query point. K-NN then uses the respective classifications of these nearest neighbours to determine the class of the query point (Hodge and Austin, 2005). This model can provide functionality for unsupervised and supervised neighbours-based learning methods. Supervised neighbours-based learning comes in two options: classification for data with discrete labels, and regression for data with continuous labels. Each unclassified object, a k-nearest neighbour query on the set of classified objects is evaluated (k is a parameter of the algorithm). The object is assigned to the class label of the majority of the resulting objects of the query. For each unclassified object, a K-NN query on the set of classified objects is evaluated, this corresponds again to a k-nearest neighbour join (Böhm and Krebs, 2004).

2.2.2.5 Support Vector Machine (SVM)

The SVM builds a classifier by creating a decision surface, an optimal separating hyper-plane, to screen different categories of data points in the vector space. SVM has been shown to be a very powerful tool for supervised classification (Carrizosa et al., 2010; Lee, 2010). Support Vector Machine requires that each data instance is characterized as a vector of real numbers. Hence, if there are categorical attributes, they have to be converted into numeric data, usually by binarization as dummy value. Scaling before applying SVM is also very important to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges and to avoid numerical difficulties during the calculation (Hsu et al., 2006).

The prediction dataset time of new instances also increases significantly when the quantity of classes increases. Hence, the training/prediction time requirements and scaling are often a difficult to deal with (Li et al., 2006).

2.2.2.6 Gaussian Naïve Bayes (Gaussian NB)

The Gaussian Naive Bayes model performs a Naive Bayes algorithm with likelihood of the features assumed to be Gaussian.

The Gaussian distribution function (Tan et al., 2005) is a bell shaped function having the centre representing the mean value. One of the drawbacks of using such estimation is that the data distribution may not concur with the Gaussian distribution function, as a result, the accuracy of the model may be reduced.

Naive Bayes, applied to classification (Tan and Gilbert, 2003) is a probabilistic classifier based on applying Bayes theorem. Naive Bayes assumes that all the attributes which will be used for classification are independent of each other (Jishan et al., 2015) applying Bayes' theorem with strong (naive) independence assumptions between the features. If some of the features are dependent on each other (in case of a large feature space) the prediction may prove to be poor.

2.2.2.7 Stochastic Gradient Descent (SGD) classifier

Stochastic Gradient Descent (SGD) is an efficient approach to discriminative learning of linear classifiers. SGD is been successfully applied to large-scale and difficult machine learning problems often faced in text classification and natural language processing (Needell et al., 2016). Given that the data is sparse, the classifiers in this module easily scale to problems large number of training examples and features.

In a linear regression, our goal is to find the line (or hyperplane) that minimizes the vertical offsets. Then again, in other words, one defines the best-fitting line as the line that minimizes the mean squared error (MSE) between target variable and predicted output over all samples in the dataset. In the algorithm SGD, it is implemented a linear regression model for performing ordinary least squares regression using a Gradient Descent optimization algorithm.

Essentially, GD optimization can be visualised as a hiker (Raschka, 2017), the weight coefficient, who wants to climb down a mountain, cost function, into a valley, cost minimum,

and each step is determined by the steepness of the slope (gradient) and the leg length of the hiker (learning rate). Using the Gradient Decent (GD) optimization algorithm, the weights are updated incrementally after each epoch (a pass over the training dataset).

In case of very large datasets, using GD can be costly since we are only taking a single step for one pass over the training set thus, the larger the training set, the slower hte algorithm updates the weights and the longer it takes to converges to the least cost.

In Stochastic Gradient Descent (SGD, sometimes also referred to as iterative or on-line GD) update of the weights is done after each training sample, and it is called "stochastic" because the gradient based on a training sample is a stochastic approximation of the actual cost gradient. Due to its stochastic nature, the route towards the global cost minimum can be direct or in zig zag if we are visualizing the cost surface in a 2D space (Raschka, 2017).

In general, the model evaluates and updates the coefficients every iteration called stochastic gradient descent to lessen the error of a model on the training data. The mode this optimization algorithm works is that each training instance is exposed to the model once at a time. The model makes a prediction for a training instance, the error is computed and the model is updated in order to decrease the error for the succeeding prediction. This process is repeated for a pre-defined number of iterations. This technique can be used to discover the set of coefficients in a model that cause the smallest error for the model on the training data.

The advantages of Stochastic Gradient Descent are (Needell et al., 2016): efficiency and ease in implementation (opportunities for code tuning). The disadvantages of Stochastic Gradient Descent include (Needell et al., 2016): SGD requires a number of hyper parameters such as the regularization parameter and the number of iterations and SGD is sensitive to feature scaling. Hence, for better performance, the dataset should be regularised to values between 0 and 1 as each attribute has different units and in turn unalike scales.

2.2.2.8 Artificial Neural Networks (ANN)

In the 1940s mathematician, Pitts and psychologist McCulloch (Ding et al., 2015; Yadav et al., 2006) have put forward neurons mathematical model from the mathematical logic view which opened the prelude of artificial neural network research. Neural network with parallel and distributed information processing network structure has a strong nonlinear mapping ability and

adaptive self-learning, robustness and fault tolerance characteristics. For artificial neural networks (ANN) data preparations are well defined (Coakley and Brown, 2000): data should be scaled to match the input side of the selected transfer function, while the specified target values should be scaled to match the output side. Most software packages will perform scaling of the input data, and will automatically generate the initial training weights. To avoid over-fitting cross-validation showed to be an effective technique (Coakley and Brown, 2000) to achieve proper results.

When applying an ANN algorithm determining the number of hidden layers is still part of the "art" of neural networks. Although the details of the literature proposals vary, the most common models of ANN, as explain by Rumelhart (1994) and Widrow et al. (1994), take the neuron as the basic processing unit. Each processing unit is characterized by an activity level, an output value, a set of input connections, a bias value and a set of output connections (Roberto, 2015). Hidden layer works as a layer of perception units where original input pattern is augmented, then a recodification of the input patterns is processed and then support mapping the input to the output units. The aim is to have right connections from input to hidden layer, so we can find representation that will perform the mapping from input to output through the hidden layers. In practical applications, the number of the hidden layer can be identified through iteration as too few/many hidden nodes employed would lead to under-fitting/over-fitting issues in pattern classification (Ding et al., 2015).

ANNs are ideal for processing nonlinear data, making it the perfect candidate for information forecasting and classification (Moro et al., 2015). In an attempt to mimic the biological brain, the neural network comprises a network of interconnected nodes (also referred to as neurons or processing units) which adjust their memory via weights, which link nodes together. Nodes are the most elementary units of any ANNs.

The great advantage of ANNs is contained within its inherent ability to generalize. Having been trained, the network is able to produce an optimum output on previously unseen data (Rooij et al., 1996). Moreover, when compared to traditional statistical predictive techniques, NNs have shown promising results. The training process of ANN generally involves five steps (Lee, 2010):

- Choose representative training samples and fit them into the input layer as the input value.

- Estimate the predictive value of the network.
- Compare the target value with the predictive one to find the error value.
- Realign the weights in each layer of the network based on the error value.
- Replicate the procedure above up until the error value of each training sample is reduced to a minimum, meaning that the training is completed.

Furthermore, Youn and Gu (2010) notes that provided sufficient nodes exist, one hidden layer can overcome any problems. They continue advocating ANN as an attractive alternative because they are robust and do not require a priori specification of the functional relationship between the variables. In addition, ANNs models are expected to produce higher classification accuracy rates than logistic regression models, because the primary purpose of ANNs is to provide satisfactory results in prediction tests rather than parameter estimation or hypotheses testing (Youn and Gu, 2010). Nevertheless, the findings of the above-referred study indicate that while the NNs model performed reasonably well (in their study), it did not outperform the conventional logistic regression model. The ANNs model, however, has its disadvantage of being unable to clearly ascertain how each input variable has contributed to actual classification of the sample (Palmer et al., 2008; Youn and Gu, 2010).

2.2.2.9 Other algorithms

Some other common algorithms like linear regression were not highlighted because for example on OLS algorithms limitation regarding not be properly handle binary and continuous variables part of the dataset used. The one's mention was chosen based on literature researched as being common in real case scenarios and business practices.

2.2.3 Evaluation and sensitivity analysis of models

As in every prediction method, it is almost unthinkable to carry out an experimental section where the performance is not mentioned and used as a reference. Japkowicz et al. (2006) suggest that the evaluation process for supervised classification algorithms should acknowledge some important steps:

- Choose an evaluation metric according to the properties of the classifier we want to measure;
- Decide the estimation method to be used;

- Revise that the assumptions made by the evaluation metric and the estimation method are fulfilled;
- Run the evaluation method with the chosen metric and estimation method;
- Interpret the results with respect to the domain.

In general, a score is a quality measure to quantify how a classifier behaves when solving a classification problem, and can be obtained through the confusion matrix (Prati et al., 2011; Santafe et al., 2015).

Below, table 1, it is shown the basic terms of the confusion matrix obtained when the algorithm is tested (using the delay scenario) for two possible outcomes delay (1) or not delay (0) more than 15 minutes.

Table 1 - Confusion Matrix (Provost et al., 1998)

	Negative Prediction	Positive Prediction	
Negative Class	True Negative TN	False Positive FP	Total Negative Classes N-
Positive Class	False Negative FN	True Positive TP	Total Positive Classes N+
	Negative Predictions $\tilde{N}^-$	Positive Predictions $\tilde{N}^+$	

The interpretation of the confusion matrix follows:

- **True Positive (TP):** These are cases in which it was predicted delay, and in fact, there was a delay;
- **True Negative (TN):** The algorithm predicted no delay, and in fact, there was no delay;
- **False Positive (FP):** The algorithm predicted delay, but in reality, the flight was on time (also known as a "Type I error");
- **False Negative (FN):** The algorithm predicted no delay, but in fact, a delay occurred (also known as a "Type II error").

Therefore, from the results withdrawn in the confusion matrix, there may be many scores according to how we aim to quantify the behaviour of a model. List of scores in supervised classification may be extensive, including standard scores and those designed ad-hoc for

specific classification problems. In table 2 it is only presented some of the most common scores (Santafe et al., 2015), as seen in table 2.

Table 2 - Confusion Matrix derived indicators

Indicator	Description
Accuracy	$(TP+TN)/(N^{+}+N^{-})$
False positive rate	1-(specificity)
False negative rate	1-(recall)
Specificity	TN/ N^{-}
Precision	$TP/ \tilde{N}^{+}$
Sensitivity, recall, hit rate, or true positive rate	TP/ N^{+}

Accuracy or the percentage of instances that are correctly classified by the model is the most commonly used decision criteria for most model assessments (Han et al., 2011).

On the other hand, the use of scalar measures is also criticized by some authors in favour of graphical methods (Japkowicz et al., 2006; Prati et al., 2011) which are seen as a better choice to capture the complexity of the evaluation process.

Sometimes, classification error (and accuracy) is selected without considering in depth whether it is the most appropriate score to measure the quality of a classifier for the classification problem at hand (Provost and Kolluri, 1999), especially valuable in domains where one class (the positive class) is more relevant than the other or when there are only a few positive samples. Thus, one may be interested in measuring the proportion of positive instances that have been retrieved by the classification model. By contrast, the specificity can be a valuable measure when the negative class is more relevant or when a minority negative class exists (Sayeh and Annie, 2014). The precision is a popular score in information retrieval and medical domains (Rezaeinasab and Rad, 2008; Twagilimana, 2006). In general, precision may be valuable to evaluate classifiers when a false positive classification is especially costly or when the interest lies, for instance, on measuring the reliability of a detection obtained by an automatic detection system.

Recall and specificity trade-off The most popular approach to balance recall and specificity is the ROC analysis (Fawcett, 2006; Provost et al., 1998), which involves a graphical

representation of recall versus false positive rate (ROC curves). When the classification method under study is discrete, the associated ROC curve has a single point. The information about classification performance in the ROC curve can be summarized into a score known as AUC, the area under the ROC curve) (Bradley, 1997; Cortes and Mohri, 2004). Although the score does not capture all the information from the ROC curve, it is more insensitive to skewness in class distribution than non-balanced scores since it is a trade-off between recall and specificity.

With statistical tests (Sts) one aims to obtain enough statistical evidence to know if the algorithms of interest have a different performance with respect to the selected score or not. Therefore, we assume the existence of two complementary hypothesis: H0 (null hypothesis) and H1 (alternative hypothesis). H0 states that both algorithms have the same performance on the basis of the selected score and, a priori, it is assumed to be true. By contrast, H1 states that the two algorithms behave differently.

Table 3 - Statistical test

		Decision	
		Do not reject H_0	Reject H_0
Reality	H_0 is true	Correct	False Positive FP Reject H0, H is true
	H0 is dales, H_1 is true	False Negative FN Do not reject H0,	Correct

Thus, the Sts does not conclude whether H0 is true or false; it is the researcher who, speculating on the fact that a small p-value is caused by a wrong initial assumption, decides to reject H0. Demšar (2006) and Japkowicz et al. (2006) review some of the most relevant objections to the use of Sts when comparing supervised learning algorithms. They stance, however, that statistical tests only measure the improbability of the obtained experimental result if the null hypothesis was correct, and statistical tests only provide certain reassurance about the validity and non-randomness of the published results.

Barboza et al. (2017) that sensitivity has values close to 1 when type I error is low and specificity is close to 1 when type II error is also low. For their study predicting bankruptcy, there is a preference for higher sensitivity because this translates into losses for lenders, whereas specificity is the threshold for gain. Perhaps, the same logic could apply the private aviation in

the first steps of using machine learning models to support decisions and allocation of resources in the planning, control and oversight of flight-related daily operations.

Regarding Sensitivity Analysis, there are numerous statistical and probabilistic tools (regression, smoothing, tests, statistical learning, Monte Carlo, etc.) that aim at determining the model input variables which mostly contribute to an interesting quantity depending on model output (Hamby, 1993; Iooss and Lemaître, 2015). Iooss and Lemaître (2015) stance three methods are to follow: the screening (harsh sorting of the most significant inputs among a large number), the measures of importance (quantitative sensitivity indices) and the deep exploration of the model behaviour (gauging the effects of inputs on their all variation range). Summing up, the referred study defends that the goal is to learn "how the uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input" (Iooss and Lemaître, 2015). Further, in their investigation, they state that based on the characteristics of the different methods, some authors have proposed decision trees to help choose the most appropriate method for its problem and respective model. Shen and Tan's (2005) recursive feature elimination was used to choose 16 variables from a large dataset for a classification prediction problem. Recursive feature elimination (RFE) is a method which performs backward feature elimination: starts with all features, and then removes some irrelevant features according to a ranking criterion until satisfied with a stop criterion. Xie et al. (2006) applied the same method and managed to reduce the number of features, but also keep the classification accuracy using an SVM algorithm. In the present study, the sensitivity analysis method mention is going to be used to evaluate and compare the top rank variables with literature flight delays analysis and prediction studies. This happens due to the fact that the dataset available lacks in relevant features, something already accepted from the beginning and corroborated during interviews with experts and managers of the private airline company. Despite the numerous iterations done, the fact-finding characteristic of this dissertation in the application of machine learning techniques in the private aviation scope justifies the use of this process in the aftermath of comparing accuracies.

2.3 Machine learning predictions and aviation

Machine learning studies are found across a wide range of research fields, and their performance compared in numerous studies.

Dogan and Tanrikulu (2013) apply classification models on several datasets in three phases: first, applying the algorithms on original datasets; second, applying the algorithms on the same datasets where continuous variables are discretised; and third, implementing the algorithms on those same datasets where principal component analysis (PCA) is applied. Overall the best classifiers out of all the trials were k-nearest Neighbours, decision tree (called C4.5), MLP and Logistics also predicted well. In summary, all dataset characteristics and PCA applications were found to affect the success rate significantly, but not the discretisation.

Liu et al. (2017) using twelve data subsets measures the classification accuracies five machine learning algorithms that were ranked in the following decreasing order: support vector machine, artificial neural network, naïve bayes, decision tree and k-nearest neighbour. Before applying the methods, feature selection techniques are used to pre-select variables and improve their accuracy (Domingos, 2012). The automatic feature selection was proved to increase accuracy in ANN and SVM, and the mean average percentage error after the feature selection is lower.

Caruana and Niculescu-Mizil (2006) presents an empirical comparison between ten supervised learning methods: SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps. The learning methods such as boosting, random forests, bagging, and SVMs attain an excellent performance. The models that performed poorest were naive bayes, logistic regression, decision trees, and boosted stumps. Nevertheless, they warn that even the best models sometimes perform poorly, and models with poor average performance occasionally perform exceptionally well.

2.3.1 Prediction studies across industries

Silva (2017) stresses the multidisciplinary character of supervised classification problems. Its benefits come from being used by multiple perspectives and should not only be considered under the tapered perspective of any particular scientific area. Hence, machine learning it stated as a field of computer science that deals with answering problems through learning from examples applicable to a multitude of industries.

Schumacher et al. (2010) test accuracy of Logistic Regression, Neural Networks, and Classification Trees with similar datasets. One has missing data; the other had its instances with missing data removed. The results show that accuracy of all three algorithms in the completed dataset is higher than in the dataset with missing values. The accuracies are 74%, 85% and 84%

respectively for Logistic Regression, Neural Networks, and Classification Trees for predicting the success of students of the given database with no missing data.

Barboza et al. (2017) analysed more than 10.000 firm-year financial observations and compared the best algorithms when predicting bankruptcy. The machine learning techniques that best performs are a random forest (a process of multiple decision trees predicting different samples and random variables of the datasets) that led to 87% accuracy, contrasting with logistic regression and linear discriminant analysis that obtained 69% and 50% accuracy, respectively, in the testing sample. In respect to the algorithms chosen for the present dissertation, while the ANN model had a lower type I error of 6.8%, type II error is higher than that of other machine learning models, having 27.2%. Random forest returns the lowest type II error, with 12.9%, and the best total accuracy rate of 87.1%, as previously mentioned. In the context of their study for bankruptcy predictions, accuracy should not be the only performance metric, but also adjusting classification models by considering different impacts of type I and type II errors. They also encourage decision makers to test and consider the use of machine learning models in their databases. When the goal of the decision maker is to predict and not necessarily explain, then the use of algorithms for prediction purposes should be the focus, and relative contribution of predictors would not be a matter of concern. Thus, results show that machine learning could be a powerful ally to make decisions about corporate operational resolutions, instead of study performance.

Youn and Gu (2010) compare restaurant firm failure prediction models using logistic regression and artificial neural networks. The results show that although many studies have reported ANNs' better prediction accuracy versus traditional techniques, including the logistic regression, the ANNs model does not always show superior performance. The results achieve were 95% to 88% of accuracy for logistic regression and artificial neural network respectively.

Moro et al. (2014) present an approach to predict the success of bank telemarketing. A semi-automatic feature selection was explored, first based on intuitive business familiarity, and then through an automated selection technique. Then four algorithms were compared: logistic regression, decision trees, neural network and support vector machine. Although two metrics were used, the focus is on the AUC results. The neural network presented the best results with an AUC near 0.8. In an increasing order of AUC results are the following: 0.715, 0.757, 0.767, and 0.794 for logistic regression, decision trees, support vector machine and neural network,

respectively. They also advocate that a good model should offer the best compromise between a desirable high true positive rate (TPR) and low false positive rate (FPR). In their best model, the neural network, TPR was 0.71 and FPR 0.24.

Sayeh and Annie (2014) compares ANN with logistic regression (LR) to predict credit rationing decisions. Use leave-one-out cross validation to ensure the robustness of the two classifiers. Based on data from a survey overall classification rate of LR is 74.80% while ANN allowed a proper classification of 71.14%. Findings also suggest that the two classifiers perform poorly when predicting the rare class (least common). The prediction of the two models is more likely to be majority leaning than towards the non-majority. This could be attributed to the fact that least common class may be under-represented and the results could be biased toward the second class (the majority class). They recommend applying a resampling technique to balance data and improve the classification performance.

2.3.2 Prediction studies in the aviation industry and flight delays

When preparing a flight, planning usually focus on solving in a sequential manner where the output of one stage is the input of the next (Papadakos, 2009). Initially, planning solves fleet assignment, where it is decided the appropriate and available aircraft to fly each leg and maximizing revenue. Secondly, maintenance routing where it is assured that required periodic maintenance schedule is complied with. Then crew pairing where the crew is assigned to the legs while following labour rules and legislation, and minimize crew costs. Around this planning features such as client (or broker) demands, ground operations, airport operations, and head office routines pace themselves to achieve better performance (safety wise too) and assure business continuity and support to operating flights.

For a private carrier, the departure can have different contexts. It can be the first flight for a Client, or rotation from one Client to another, or a “simple” operational aircraft rotation, or a first flight out of a schedule maintenance, etc. Wu (2006) sees the rotation process was seen as a whole process, and a sequential optimization algorithm is proposed to improve operational reliability of schedules. In that article, it is defended that the use of a simulation model provides immediate feedback on alternatives and visualization of possible results to the schedule. It was also defended that the use of the optimization model can define buffer times to better cope with operational demands, and moreover, allocate more time to critical flights, and propagate punctuality and less pre-flight times in later legs of the same rotation (Rosenberger et al., 2002).

Cook et al. (2012) estimate that in Europe the average delay experienced by a passenger can be up to 90% higher than the actual delay of the flight. This reinforces that a delay in a stage of flight preparation can lead to passengers (and cargo) poor connectivity along with compensation cost to the airline or the client contracting the Private Airline company.

The trend in the airline delay management literature has been to integrate the management of multiple resources (aircraft, crew and passengers, etc.) in the same system. Santos et al. (2017) analyses how their linear programme approach may deal with operational delays and to help on deciding if subsequent flights should be delayed as well. In their programme function, they take into account airport restraints, such as taxiway capacity and runway separation, and fuel costs and passenger costs. They concluded that their proposal might lead to cost reductions to the airline.

Usually, in a regular flights carrier, disruption to schedule mainly come from crew absences, mechanical failures and bad weather. Despite the data used was based on regular flights, Bratu and Barnhart (2006) present how real-time decision-making algorithms and optimization models may help when recovering from different levels of disruption, help identify departure postponing and/or cancelation (whether current or subsequent flight), help an airline to increase customer retention and a long-term profitability.

Bai (2006) uses neural networks and logistic regression to detect the pattern of airport arrival delay in Orlando Airport. Its outcome shows that arrival delay is highly related to the originate delay. The airport arrival delay is found to show seasonal and weekly patterns, which is related to the schedule performance of the carriers. The precipitation and wind speed were also found to be contributors of airport arrival delay. The capacity of the airport and its constraints were not found to be significant contributor. In addition, the precipitation, flight distance, season, weekday, arrival time and the space between two successive arriving flights were found to contribute to arrival delay of flights. However, flight delay is not necessarily during a peak period but depends on the impacts generated on subsequent flights during that time, and the use of a predictive model can give indications for the appropriate recovery actions to recover/avoid those delays.

Dimopoulos et al. (2017a) analysed delays using data from the United States Department of Transportation to predict flight delays. They could predict with 85.5% accuracy if the flight is going to be delayed and with 56.0% the delay time interval. To gain a more robust measure of

the accuracy over the dataset, a k-fold cross-validation with 3 random shuffled folds was implemented. For the classification problem, the highest accuracy belonged to Logistics regression algorithm and lowest to Gaussian naive Bayes with 64.2%. Going further in the analysis, by observing the average delay, they noticed that: days with most delayed flights are those before or after each holiday and during each holiday; the delay during the afternoon seems always to be affected by earlier delays, which force flights to leave later; and week days and days of month seems to affect the average delay, being Summer months significantly with more delays.

Martinez (2012) uses a dataset composed of records of all USA domestic flights of major airlines, from 1995 to 2010, and accuracies are compared against literature and an online forecasting engine. The goal of the study was to estimate the probability of any flight to be more than x minutes late. Despite being a regression problem and not using an algorithm similar to the ones in the literature revision, it should be highlighted, as accuracy achieved may contribute to accuracy benchmarking and it was a project for a master thesis in computer science in collaboration with Amadeus IT Group SA, a provider for the global travel and tourism industry. The best prediction method achieved was the most specific one, which takes into account all the combination of categorical features and a condition on the arrival hour, with a measured AUC around 0.68.

3. Method

Data mining is a process that allows a deductive learning to find hidden information in a database, fit that same data to a model and carry proper data analysis, and discovery of descriptive and predictive tasks for business purposes (Dunham, 2002). The method for the present study is the CRISP-DM, as approached previously in the literature revision: business understanding, data understanding, data preparation, modelling and evaluation.

3.1 Business and data understanding

The business understanding and data understanding was done via revision of the literature that encompasses as aggregate or in aviation-specific challenges, private airlines market, flight delays, machine learning, its goals and practical case studies. At the same time, meetings with managers and experts from the Private airline were conducted in order to assess the best available features to be used for the following steps.

3.2 Data collection and preparation

Before applying the algorithms to compare their accuracy for prediction purposes, an understanding of the different variables and its characteristics and context is mandatory (Dunham, 2002).

For the objectives set, it is used the above-mentioned Private company flight data from 2014 to 2017 (first quarter). It includes a mix of types of data from categorical as well as continuous features. Before going any further it can acknowledged the challenge with the variables at hand to achieve a perfect performance of the algorithms as the complexity of this sector is vast, and there are still variables that even key players on the market are yet to put them into numbers or even transform them into measurable data. The aim of this analysis with the learning methods is to predict flight delays over15 minutes on departure, based on the available parameters Table 4.

Table 4 - List of features

#	Description	Scale	Variable Code	Origin	Data Type
1	Month	1-12	MONTH	Extracted	Categorical
2	Day	1-31	DAY	Extracted	Categorical
3	Week day	1(Monday)-7(Sunday)	WEEK	Computed	Categorical

#	Description	Scale	Variable Code	Origin	Data Type
4	Season time	Spring, Summer, Autumn, Winter: 1, 2, 3 and 4 respectively	SEASON	Computed	Categorical
5	Private airline flight?	If a own flight (1) or for a client (0)	Private Carrier	Computed	Categorical
6	Engine numbers	Two (2) or four (4) engines aircraft	ENG_NUMBER	Computed	Categorical
7	Aircraft registration	The aircraft registration XX-ABC	Tail_No	Extracted	Categorical
8	Internal code of flight category	(C)Charter, (D) general aviation, (J) normal service, (K) training,(N) business aviation, (P) positioning, (T) technical test and (W) military	STC	Extracted	Categorical
9	Departure airport with IATA code	XXX (three capital letters)	Dep	Extracted	Categorical
10	Departure hour of the day in UTC	Departure hour (0-23)	DEP_HOUR	Computed	Categorical
11	Arrival hour of the day in UTC	Arrival hour (0-23)	ARR_HOUR	Computed	Categorical
12	Flight time from ATD to ATA	Flight time converted in minutes	Flight_time	Computed	Integer
13	Departure delay?	If STD was delayed (1) or not (0)	DEP_DELAY	Extracted	Categorical
14	Arrival delay?	If STA was delayed (1) or not (0)	ARR_DELAY	Extracted	Categorical
15	Client IATA code	XX two capital letters	Client_ID	Extracted	Categorical
16	Flight more than 10 hours? From ATD to ATA.	Yes (1), No (0)	Flight_more10hrs	Computed	Categorical
17	Flight departure occurred during local night office hours)	Yes (1), No (0)	If_Dep_night_OFFICE	Computed	Categorical
18	Year of the flight	Year from 2014 to 2017	YEAR	Extracted	Categorical

#	Description	Scale	Variable Code	Origin	Data Type
19	Flight during weekend?	If weekend (1) or not (0)	Weekend	Computed	Categorical
20	Flight arrival occurred during local night office hours?	Yes (1), No (0)	If_Arrival_night_OFFICE	Computed	Categorical
21	Flight encompassed totally during local night office hours?	Yes (1), No (0)	flight_during_OFFICENIGHT	Computed	Categorical
22	Aircraft age in years	XX for years	AC_age	Computed	Integer
23	Previous flight arrived with delay?	Yes (1), No (0)	If_previous_flight_delayed	Computed	Categorical
24	Previous flight delay at arrival in minutes	XX in minutes	Previous_flight_delay	Computed	Integer
25	Delay IATA code of the previous flight	XX with numerical code	Previous_flight_delay_reason	Computed	Categorical

The dataset is characterized by having 6907 entries and 25 initial features.

```
In: X.shape

Out: (6907, 25)
```

Figure 1 - Shape of the dataset extracted

For the data preparation stage, a thorough collection of the data was made to remove from the dataset any possible data that could have a negative impact on the model's performance. Data from real-world sources are often erroneous, incomplete, and inconsistent, perhaps due to operational error or system implementation flaws. Such low-quality data need to be cleaned prior to data mining (Gürbüz et al., 2011). When working on this stage, several issues were taken into consideration such as error of instances, outliers, missing and irrelevant data and human interaction (as personnel inputs most of the data extracted from the company's database). Thus, data reduction was performed by means of instance and/or feature selection. First, missing data were removed from the dataset by removing instances that did not have all the features with inherit value. Unnecessary space characters or other spelling mistakes were

also cleaned (see annexe A). Feature selection was done manually by identifying the most relevant, explanatory input variables within a dataset (Abdallah and de La Iglesia, 2015; Chandrashekar and Sahin, 2014; Yang and Olafsson, 2006) and the chosen variables are shown in the feature table. The selection was empirical from Private Aviation experience, meetings with experts and managers from the private carrier and literature. In an iterative process, the selection of the features was also achieved in a way to be compatible with the algorithms ahead. In the present study, one has access to a very limited number of variables. In particular, even though we use a considerable number of observations, the database supplied a limited number of variables. Therefore, the impact of feature selection is not prominent detailed.

Before transforming the dataset, data projection (Crone et al., 2006) was applied to the original dataset by transforming raw data into a possible data processing, as being beneficial for the classification algorithm. It comprises techniques of value transformation, e.g. mapping of categorical variables and discretization or scaling of continuous ones. Working with large attribute sets of mixed scale, data mining routinely encounters mixtures of categorical attributes. As some of our categorical variables contained multiple categories, we applied a dummy variable various categorical features (see annexe A).

Further, in the data preparation stage, optimizing pre-processing steps were taken and two transforming options were tested: PCA and MaxAbsScaler (scikit-learn library). The latter scales and translates each feature individually such that the maximal absolute value of each feature in the training set is transformed to be 1,0 (one). The PCA produces orthogonal (i.e. perfectly uncorrelated) axes as output, so without clustering, the PC axes may be used directly in subsequent analyses in place of the original variables, and assures non-collinearity.

```
In:

max_abs_scaler = pre-processing.MaxAbsScaler()

Abs = max_abs_scaler.fit_transform(X)

In:

pca = PCA(n_components=50)

X_PCA = pca.fit(X).transform(X)
```

Figure 2 - Data transformation

Both PCA and MaxAbsScaler were found to have similar results and being better than other pre-processing steps optimizations. PCA stands out for a slightest better performance and accuracy, and better at dealing with correlated features.

For a better application and performance to set objectives, the mathematical aspects of each algorithm are not the focus, but their application in a real-valued challenge using python language, python libraries and Jupyter as the interface. Hence, for this stage of the present methodology a relationship was iteratively explored on how features, algorithms and respective outputs can vary and produce potential valid insights and future business directives.

3.3 Modelling

One of the main objectives of the study is to compare algorithms that may predict flight delays. Hence, several issues were taken into consideration: overfitting, dataset dimension, high dimensionality and integration. Thus, a knowledge extraction procedure was conveyed to assess the accuracy of the algorithms.

As per the revision of literature algorithms, now follows a brief resume of the related supervised learning scikit-learn library algorithms (Pedregosa et al., 2011) being used (see Table 6):

Logistic Regression (LR) - applies ordinary least squares linear regression. It is a multiclass classification problem and logistic regression produces predictions between 0 and 1, a one versus all scheme is used (one model per class) where the algorithm relates every class with all the remaining classes, structuring a model for every class.

Linear Discriminant Analysis (LDA) - a classifier with a linear decision margin, produced by fitting class conditional densities to the data and using Bayes' regulation. The model fits a Gaussian density to each class, assuming that all classes share the same covariance matrix. The fitted model can also be used to reduce the dimensionality of the input by projecting it to the most discriminative directions.

Decision Tree Classifier (CART) - where a non-parametric supervised learning method is used for classification. The goal is to generate a model that predicts the value of a target variable by learning simple decision rules inferred from the features. Decision Tree Classifier is capable of performing multi-class classification on a dataset, and capable of both binary classification and multiclass classification.

K-Neighbours-Classifier (KNN) – the classifier implements the k-nearest neighbours vote. Here, scikit-learn implements nearest neighbours' classifiers: by learning based on the K nearest neighbours of each query point, where k is an integer value pre-specified.

SVC (SVM) – a C-Support Vector Machine Classification and takes as input two arrays: an array X of size *n_samples, n_features* holding the training samples, and an array Y of class labels (strings or integers) with size *n_samples*. After being fitted, the model can then be used to predict new values, and the decision function depends on some subset of the training data, called the support vectors.

GaussianNB (NB) - implements the Gaussian Naive Bayes (Chan et al., 1982) algorithm for classification, where as previously said assumes that features follow a normal distribution.

SGDClassifier (SGD) - this estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated with a decreasing strength schedule (the learning rate).

MLPClassifier – this classifier trains on two arrays: array X of size (*n_samples*, *n_features*), which holds the training samples represented as floating point feature vectors; and array Y of size (*n_samples*), which holds the target values (class labels) for the training samples. After fitting (training), the model can predict labels for new samples. Values larger or equal to 0.5 are rounded to 1, otherwise to 0. Multilayer perceptron (MLP) represent a prominent class of NN (Bigus, 1996; Krycha and Wagner, 1999), implementing a paradigm of supervised learning methods which is routinely used in academic and empirical classification and data mining tasks. Being universal approximators, NN should theoretically be capable of processing any continuous input data or categorical attributes of ordinal, nominal, binary or unary scale.

```
In:

# prepare models

models = []

models.append(('MLPC', MLPClassifier()))

models.append(('LR', LogisticRegression()))

models.append(('LDA', LinearDiscriminantAnalysis()))

models.append(('KNN', KNeighborsClassifier()))

models.append(('CART', DecisionTreeClassifier()))

models.append(('NB', GaussianNB()))

models.append(('SVM', SVC()))

models.append(('SGD', SGDClassifier()))
```

Figure 3 - Supervised Classification Algorithms to evaluate

The evaluation is made through a stratified k-fold cross-validation technique (Kelleher et al., 2015; Refaeilzadeh et al., 2009). Through a leave-one-out cross-validation classification algorithm, the dataset is grouped into 10 equal stratified folds from the dataset to train and test the algorithms (Kohavi, 1995; Silva, 2017). This means that the model will train and test the data 10 times different parts of the dataset. These train and test executions are run 20 times, and the score is measured through the average and standard deviation of accuracy of those 20 executions (see table 6).

```
In:

seed = 20

# evaluate each model in turn

models = []

results = []

names = []

scoring = 'accuracy'

for name, model in models:

kfold = cross_validation. StratifiedKFold(6907, n_folds=10,
shuffle=True, random_state=seed)

cv_results = cross_validation.cross_val_score(model, X_PCA, Y15,
cv=kfold, scoring=scoring)

results.append(cv_results)

names.append(name)

msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())

print(msg)
```

Figure 4 - Modelling the data with PCA transformation and applying the referred algorithms

3.4 Evaluation and sensitivity analysis

The evaluation process for supervised classification algorithms encompasses two steps: comparison of the accuracy of the algorithms chosen, along with the two best accuracies' algorithm in detail (Logistic Regression and MLP Classification) regarding other performance indicators by analysing its confusion matrix. The sensitivity analysis makes use of the recursive feature elimination (RFE) technique. In this scenario, logistic regression and CART are chosen to display the features' rank contribution to the predictive outcome of flight delayed or not delayed (more than 15 minutes) on departure. Despite not having the highest accuracy, the Decision Tree Classifier algorithm was also used to extract the percentage of relevance of each individual feature to the model and therefore to the predictive outcome. Based on this, an

exploratory analysis of the dataset will be drilled-down, along with a sensitivity analysis of the top 5 features in each model used (Logistic Regression and Decision Tree Classifier). On the discussions chapter, the results and other evaluation indicators addressed in the literature revision are with other similar predictive related studies.

4. Results

4.1 For Supervised Classification algorithms accuracy comparison

An empirical comparison of nine supervised learning algorithms using accuracy as performance criteria now follow. The pre-processing methods, besides the cleaning ones, two are chosen to achieve best accuracies, and both outcomes are explored. Through leave-one-out cross-validation classification algorithms were evaluated using 10-fold cross-validation (Kohavi and others, 1995) conducted 20 times. Hence, with a specific python code, it was possible to evaluate the referred nine algorithms (see tables 5 and 6).

Delay at departure leads to delay at arrival?

On dwelling into flight analysis, when delays restrain us from getting on time where we want to go, we first need to look at where exactly we need our focus on. On the variables prior time of departure? Or variables of flight data between departure and arrival? Extracted from the flight's dataset, from 2014 to 2017 77% (see table 7) of the flights with delay at arrival was followed by a delay at the departure airport. This fact is also acknowledged as a trend in other studies (Cao and Fang, 2012), hence the focus on predicting flight delay at the origin airport and their contributors.

Table 5 - Crossing number of arrival delays with departure delays.

	Arrival not delayed	Arrival with delay
Departure with no delay	1899	1084
Departure with delay	335	3593

From the flights that were a delay at arrival did not root from the delay at departure, it may be due variables such as aircraft performance indicators, traffic at both departure and arrival airports, etc. As previously discussed, delays have a major weight on financial expenditures, as most of them root from the delay at the departure airport. The next step is to use the referred dataset, which is composed of the best available variables that may explain a departure delay 15 minutes, and compare their accuracies (see annexe C).

First, PCA pre-processing was applied to the dataset. Using the method and estimators referred to achieve the classification prediction of a delay at departure, the results are:

Table 6 - Results of the modelling with PCA pre-processing (average and standard deviation of the accuracy)

MLPC: 0,7055 (0,0172)	LR: 0,7063 (0,0143)
LDA: 0,7086 (0,0154)	KNN: 0,6611 (0,0138)
CART: 0,6534 (0,0164)	NB: 0,6269 (0,0197)
SVM: 0,7011 (0,0171)	SGD: 0,6096 (0,0405)

LDA, LR and MLPC are the top scorers. This can be explained by the three of them make use of the PCA transformation of the raw data to better achieve higher performance and avoid collinearity. Collinearity on the data set is intrinsic, as many variables were created by condition-based on other variables, and in a Private Aviation environment, a small deviation from the standard process may initiate a chain of events that can easily lead to a delay.

Secondly, MaxAbsScaler pre-processing was used to transform the data. Using the same methods and estimators above referred to predict delay at departure, the results are:

Table 7 - Results of the modelling with MaxAbsScaler pre-processing (average and standard deviation of the accuracy)

MLPC: 0,6996 (0,0173)	CART: 0,6743 (0,0116)
LR: 0,7089 (0,0154)	NB: 0,4608 (0,0190)
LDA: 0,7019 (0,0142)*	SVM: 0,6681 (0,0170)
KNN: 0,6998 (0,0171)	SGD: 0,5950 (0,0651)

In this case, LR and LDA were the top scorers. The MaxAbsScaler was used to standardize the raw data and avoid variables with higher amplitude of values could affect its actual importance on explaining the independent variable. The LDA, although reaching a reasonable scorer, the scikit-learn identifies the already expected collinearity as a problem, and interpretation on the result and upstream stages of processing should be revised.

Following the two best accuracies, an individual evaluation of Logistic Regression and MLP Classifier is presented. To evaluate the both algorithms several evaluation indicators can be used. In the Confusion Matrix table, we achieve the results in table 10.

Table 8 - Confusion Matrix for Logistic Regression and MLP Classifier

	Logistic Regression		MLP Classifier	
	0	**1**	**0**	**1**
0	2556	1419	2928	1047
1	894	2038	1236	1696

In the Figures 1 and 2, it is possible to visualize the ROC and the achieved area under the curve (AUC) for each algorithm. The dashed lines define the threshold below which the prediction is

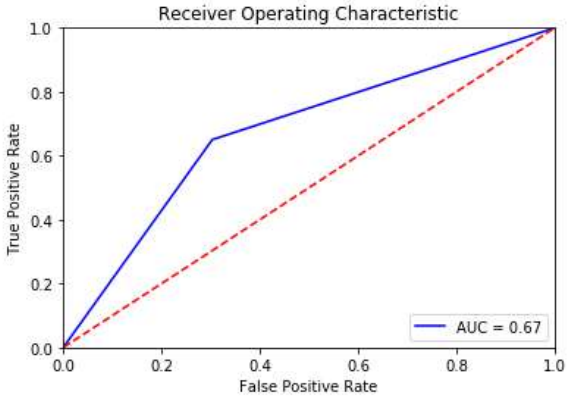


Figure 6 - ROC for Logistic Regression

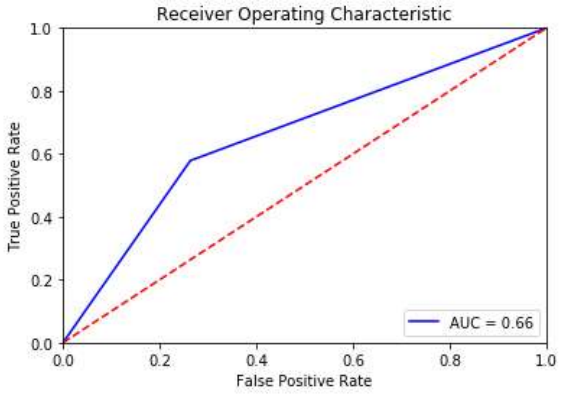


Figure 5 - ROC for MLP Classifier

considered to the prediction close to random.

The neural network (MLP Classifier) predicts with 2283 errors, whereas the Logistic regression gets 2313. The AUC of both algorithms are very similar, hence accuracy of both classifiers achieves analogous results.

In the next table its summarize the performance indicators of both MLP Classifier and Logistic Regression run once each.

Table 9 - Algorithms performance indicators

Performance Indicators	MLP Classifier	Logistic regression
Accuracy	0,66	0,67
False positive rate	0,27	0,36
False negative rate	0,43	0,36
Specificity	0,73	0,64
Precision	0,61	0,58

Performance Indicators	MLP Classifier	Logistic regression
Sensitivity, recall, hit rate, or true positive rate	0,57	0,69
ROC/AUC	0,66	0,67

4.2 For Exploratory and Sensitivity Analysis

Fulfilling the second major objective of the present dissertation, an exploratory and sensitivity analysis is conducted (see annex C). As seen in the histogram Figure 3, the delay difference distribution is vast. Which is something likely to occur. If two flights were chosen to check in detail their delay root cause, over several iterations one would come to the conclusion that a specific delay reason can have different impacts, in minutes, in the actual delay of the flight,

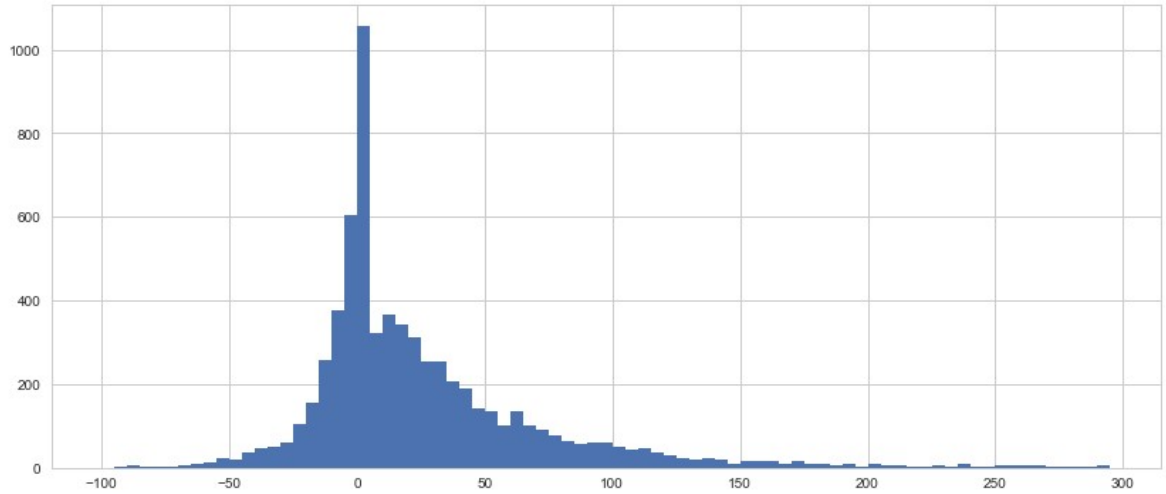


Figure 7 - Delay in minutes histogram

according to with other circumstances not always possible to measure and to transform into data. In Table 12, it is possible to visualize the average of the time difference in minutes of flights delayed (1) and not delayed (0), and in Figure 4 its behaviour over the years. As this variable proved to be too scattered and the tested prediction regression got low accuracies (see annex D), the prediction objective of this study was veered to classification from the beginning.

Table 10 - Average on time performance per delay status

Departure Difference	
DEP_DELAY	Average in minutes
0.0 (no delay)	-93
1.0 (delayed)	58

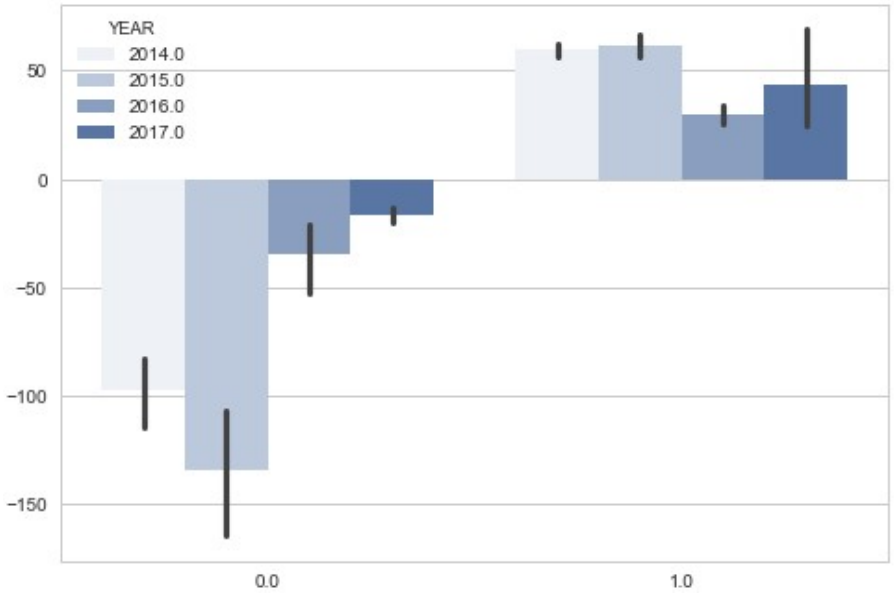


Figure 8 - Delay status' average in minutes along the years

In the Table 13 it was possible to, through a feature selection function called Recursive feature elimination (scikit learn package), identify the rank and relevance of the variables at hand, for logistic regression and CART, respectively. Concerning the top 5 relevance features using a Decision Tree Classifier (CART) algorithm, it is possible to convey that continuous features received higher relevance to the model, in contrast with the Logistic Regression, where binary features were ranked first.

Table 11 - RFE for Logistic Regression

Logistic Regression	Top 5 ranking	Decision Tree Classifier	Top 5 relevance
If_previous_flight_delayed	1	Previous_flight_delay	21,31%
If_Dep_night_OFFICE	2	DAY	14,85%
A Private Airline flight	3	Flight_time	13,34%
Weekend	4	DEP_HOUR	9,26%
Flight_more10hrs	5	ARR_HOUR	8,38%

After using the already specified dataset, one of the features was set aside. The reason of delay from the previous flight was isolated, turned into a dummy variable, and a Logistic regression was applied, hence deeper information can be extracted. In the following Table 14, it is

presented the top 5 reasons for the delay of the previous flight in the IATA code (a standard in the aviation system) that most influence the delay on departure of the consecutive flight.

Table 12 - RFE for Logistic Regression using the feature IATA code delay of the previous flight

Logistic Regression with IATA delay codes	Top 5 ranking
Reason-67: CABIN CREW SHORTAGE, sickness, awaiting standby, flight time limitations, crew meals, valid visa, health documents, etc.	1
Reason-46: AIRCRAFT CHANGE, for technical reasons.	2
Reason-96: OPERATIONS CONTROL, re-routing, diversion, consolidation, aircraft change for reasons other than technical.	3
Reason-75: DE-ICING OF AIRCRAFT, removal of ice and/or snow, frost prevention excluding unserviceability of equipment.	4
Reason-34: SERVICING EQUIPMENT, lack of or breakdown, lack of staff, e.g. steps.	5

5. Discussion

5.1 From Supervised Classification algorithms accuracies

Examination of the comparison outputs may aid in model validation and provide guidance to choose the best available path that applies to real case scenarios and possibly go even further in the analysis of flight delays.

As seen in the previous chapter, after discretising some of the features, selecting the best one available, and transform them into components (PCA technique), it was seen that this procedure proved to help and improve only slightly the accuracy of all the models as previous studies corroborate with this contribution, e.g. Dogan and Tanrikulu (2013) and Howley et al. (2006).

Regarding the models' comparison, artificial neural network and logistic regression were found to be the bet to fit algorithms. Both have an accuracy around 0.70, which may be low in comparison with other studies applying the same type of algorithms, e.g. Youn and Gu (2010), Bai (2006) and Moro et al. (2014). This is due to features availability. Although a manual, interactive and iterative process of creating and selecting features was carried, it's possible to improve the accuracy from an initial 0.5 accuracy on all the models (almost a random probability, where were used the features extracted directly from the database) to around 0.7. The lack of certain relevant private aviation known inputs transformed into representative dataset variables lead to such results, however not far from similar literature.

When running the models individually, both MLP and logistic regression showed slight lower values, as expected, since the repetitions on the comparison for producing the accuracy were higher thus outputting a higher average. As acknowledge Barboza et al. (2017), accuracy should not be the only performance metric, but also adjusting classification models by considering different impacts of type I and type II errors.

Table 13 - Type errors from MLP Classifier and Logistic Regression

Statistical test	MLP Classifier	Logistic Regression
Type error I	1047	1419
Type error II	1236	894

From the Table 10 and are calculated the performance results shown in Table 15 based on the confusion matrix. Following Barboza et al. (2017) interpretation of type errors, there is a

preference for higher sensitivity because may translate into extra focusing on flights that are predictably delayed, whereas specificity is the threshold predictable on time performance.

5.2 Exploratory and Sensitivity Analysis

In a multiparty system, elements can potentially interact in different ways each time because they are interdependent. Take the airline control system—the outcomes it delivers vary tremendously by weather, equipment availability, time of day, etc. So being able to predict how increasingly complex systems interact with each other is alluring (McGrath, 2014).

Over the years, the delay indicator of the Private Airline got better, and percentage of delay on departure decreased (see annex C).

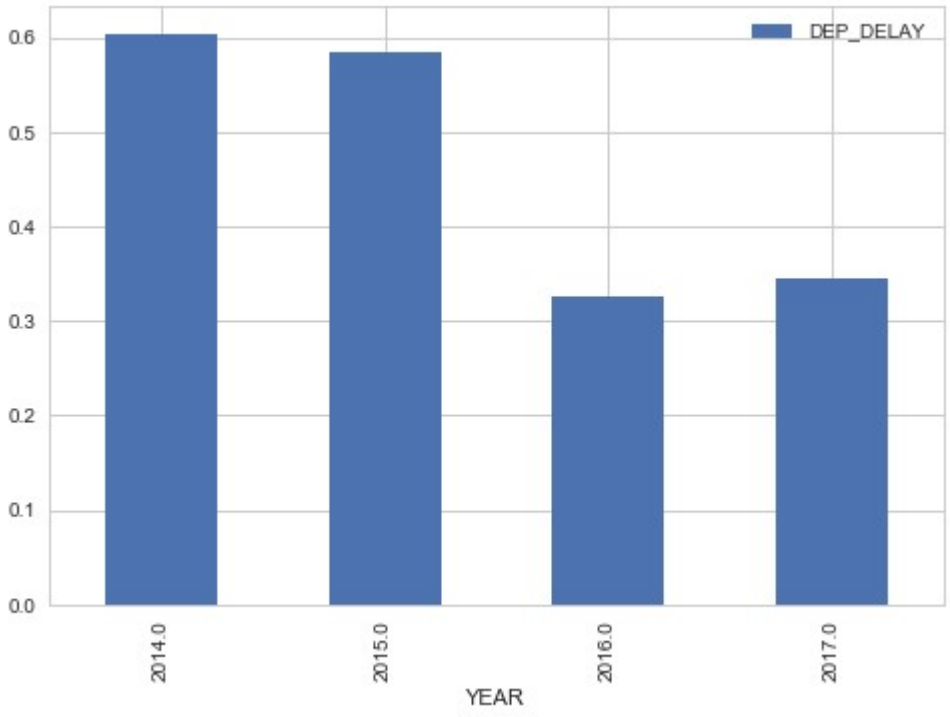


Figure 9 - Average probability of flight delay above 15 minutes along the years

However, it is still vital to understand why delays occur, and in a specific operation like in a private airline, where nature and circumstances of the flights are volatile, it comes as a business priority. Sternberg et al. (2016) observe that Brazilian flight system has difficulties to recover from previous delays especially when operating under adverse meteorological conditions, delays occurrences may increase. Although the weather variable is not a feature, its related range can be interpreted from the present dataset in the IATA code for the delay, specifically

number 75 (Table 14). Nevertheless, previous delays are a major relevant influence on departure delay, and analysis such as Wu (2006) can optimize the operational reliability. Thus, difficulties in recovering from a previous delay scenario, or in other words, delay propagation, is a key performance indicator.

Time-related features increase the chances of delay. If departure at head office night hours or during the weekend in the present study is ranked between the top five, Sternberg et al. (2016) specifies even more in detail, and sees that scheduled departures during late evening and night have around 24% more chances of being delayed. From a temporal perspective, we also find out that Brazilian flight delays are linked to the day of the week and the time of the day, a scope tangible when applying RFE to a CART algorithm.

Rebollo and Balakrishnan (2014) argue that the level of significance of the explanatory variables is expected to vary depending on the desired output of the prediction model (regression vs classification), as well as the forecast horizon. It was, however, assumed that if an explanatory variable is significant for the regression problem, it will also be significant in the classification problem. From the present study point of view, this assumption was not fulfilled on the opposite direction. Using the same features, a test was made by applying the same comparison exercise for a regression problem where the prediction was a flight delay in minutes. Here, the R2 indicator was 0.29, and the mean absolute error in predicting the delay was around 98 minutes (see annexe D), values that if applied could lead to greats costs whether by error whether by potentially allocating resources to avoid a delay what is inexistence. Further inspection of the test error obtained in the referred study of classification problem reveals that the False Negative Rate (FNR) clearly dominates the False Positive Rate (FPR), a similar behaviour occurs in the MLP Classifier's FNR and FPR values. This happens in both situation because both proposed prediction models focus on the delay state, but does not capture localized delays, an integral part of delay analysis for private aviation as seen in the literature revisions chapter.

A logistic regression ranking using Recursive feature elimination (RFE) was applied only to IATA code as delay reason of the previous flight, and it was obtained the results shown in Table 14. The first ranked reason is "cabin crew shortage, sickness, awaiting standby, flight time limitations, crew meals, valid visa, health documents, etc.". As exposed in Papadakos (2009) the planning phase addressed the allocation of the crew (pilots and cabin crew), and as numbers

of cabin crew are dictated by European legislation, and last notice shortage of an element in the cabin crew team, immediately renders a flight not to depart until that vacancy is fulfilled. In Bineid and Fielding (2003) the highest ATA chapter feature that influences actual delay rates for long-haul flights is 71-80 – Power Plant and Engine related – usually an item of no-go or demanding specific maintenance procedures that may lead to further delays and likely to deploy the allocation of a new available aircraft. Thus, corroborating with the second item ranked in Table 14 (reason 46 as aircraft change for technical reasons). The remainder three reasons that contribute to the delay model are related to factors not controlled by the operator, in this the private airline. Weather related difficulties are often hard to control proactively, and airport restrictions come with the business context of both: flying to a different kind of airport (sometimes in third world country), and as seen in Deshpande and Arıkan (2012) due to private aviation's low market share it represents a significant impact on the flight schedule and on time probability. This latter explanation is also reflected in the logistic regression RFE rank, wherein third place appears the parameter Private Airline flight as a contributing factor to the delay mode.

6. Conclusions

The challenges of artificial intelligence have vexed researchers for decades (Mullainathan and Spiess, 2017). Even simple tasks such as digit recognition - challenges that we as humans overcome so effortlessly - proved extremely difficult to program. Introspection into how our mind solves these problems failed to translate into procedures. The real breakthrough comes once we stopped trying to deduce these rules. Instead, the problem is turned into an inductive one: rather than hand-curating the rules, we simply let the data tell us which rules work best. The world of machine learning is vast, as a consequence, to use related model or models' different tweaks and enhancements are possible, and their mastering is a long process of empirical and literature based iterations. Hence, the current dissertation is backed up by current literature revision on defending the initial expectation of overcoming empirical analysis by extracting supports to decision on historical computing data. Knowledge arising from this exploratory application in private aviation is relevant both to the producer of the contents as to the end user (the company). A highlight should be done to the relevancy on how accuracy and sensitivity analysis are related. Despite the low accuracy, the respective variables (and their importance to the model) of the dataset are in line with the aviation delay literature. Thus, it is far clear that the results achieved are the highest possible by considering the singularity of this segment of air transportation. Based on that, well-identified stages from the CRISP-DM method are mandatory whenever one indulges into apply a supervised machine classifier algorithm. Choose want you want to predict; choose the best dependent variables; turn them into a dataset; pre-process it for optimizing purposes (quality and quantity wise); choose the model(s); apply the best iteration process to achieve the best results without over-fitting (getting a bias outcome with irrelevant and potentially dangerous for business purposes).

6.1 Supervised Classification algorithms accuracy comparison

Predictive analytics increasingly allow us to expand the range of interrelationships we can understand. This in turn gives us a better vantage point into the behaviour of the whole system, in turn enabling better strategic decision-making (McGrath, 2014). Comparing the nine algorithms, the Artificial Neural Network (MLP Classifier), Logistic Regression and Logistic Regression got better performances, around 0,70.

6.2 Exploratory and Sensitivity Analysis

Although not with a high accuracy, it was possible to reach a set of variables that influence in the delay on the departure of the private airline. The delay on the previous flight is by far the feature that most influences the following flights on-time performance. This is already a fact that appears in the literature of delay analytics, and the various algorithm can be used to deal with delay propagation and aircraft rotation efficiency. Along with the previous delay, time relates features also influence the delay. Departure during office night hours and weekends tend to delay more often than outside those periods. The remaining variables are linked with the resource allocation the company may provide. Flights where the Private Airline is the carrier and flight time over 10 hours of block times are likely to contribute to delay on departure.

6.3 Machine Learning techniques

Along almost one year for the present dissertation, it was possible to reach an acceptable level of proficiency on how to deal with machine learning techniques, and apply them to the business model. In this particular case, to a private airline study of flight delays and respective data. The implementation codes are, at the far possible extent, shown in the text or in the annex, so future researchers and ML enthusiast take the lessons learnt. Thus, despite the brief, but intended, way to explain the models used, it is possible to current, and future data analytics enthusiasts apply in a quick and proper manner machine learning tasks to their intended business goals. Thus, the main limitations are both the lack of even more relevant feature in the dataset and lack of similar studies focusing on private aviation.

6.4 Contributions to Business Management

An expression such as big data and business intelligence are currently in vogue, and a thirst to analyse company's data is challenging and alluring. Combining the world of private aviation and Machine Learning techniques is not common to find, and there is, with no doubt, a great margin to improve. The overall contribution of this dissertation is to add another step in combining the industry of aviation, in particular, the segment of charters and wet leases, and big data applications. Hence, Private Aviation, although far from other industries in the business analytics world, has currently enough resources and possibilities (Cook et al., 2012) to achieve higher performance learning and inuring from flight-related generated data, and use it as a competitive advantage. The referred processes of Machine Learning and the gain of a

competitive advantage with these tools are very well applicable to every industry that by defining their precise challenges requiring resolutions or better management, may built unique models adequate to the respective singularities, learn from their own specific historical data, and provide better support to their decision makers who may increase thereafter their efficiency and/or effectiveness on business related operations.

6.5 Limitations and future researches

Issues such as data collection, storage, and processing specific to analytics are increasingly considered important issues in overall system design, and data analytic methods are only as good as the data on which they are based (Hazen et al., 2014). In efforts to broaden the effectiveness of analytics in the business process (Kohavi et al., 2002), solutions are emerging that go beyond the customer-facing applications, reaching "behind the scenes" to applications in commercial relationships, marketing actions, supply chain visibility, price optimizations, and workforce analysis.

Along with operational features, maintenance had a major contributing factor for the on-time performance of an aircraft. Therefore, as maintenance data not easily reachable, the ageing fleet is a constant challenge. The problem with component faults is significantly observed in the ageing aircraft. It is, therefore, necessary to anticipate delays so that proper maintenance processes can be initiated before an actual delay occurs. The health of the aircraft is monitored through the fault and alert messages, which are relayed from the different subsystems, during its journey. These faults and alerts are leading indicators of the health of the aircraft (Dattaram and Madhusudanan, 2016). In a Private company, combining maintenance data with operational data and insights to predict possible future constraints/delays may improve the accuracy of models only based on historical data but adhere and compare predictions to future operational limitations. Besides the maintenance issues, from the meetings with managers and experts from the private airline and revision of the literature, it was agreed that features are withdrawn from airport processes and authorization during preparation to depart and take off, including traffic management and procedures, deep impact on the on-time performance of any airline. Hence, a structured dataset with a feature coming from the referred entities connected with the related flight could increase the number of relevant features when constructing a predictive model with ML techniques. Related to an improved dataset and respective variables a more thorough study

is recommended to identify what kind of behaviours one wants when applying a supervised classification model to predict flight delay, and what indicators should get the higher relevance.

7. Bibliographic References

Abdallah, T.A., de La Iglesia, B., 2015. Survey on Feature Selection. ArXiv151002892 Cs.

Ahmed, A.H., Poojari, C.A., 2008. An overview of the issues in the airline industry and the role of optimization models and algorithms. J. Oper. Res. Soc. 59, 267–277. https://doi.org/10.1057/palgrave.jors.2602350

Bai, Y., 2006. Analysis Of Aircraft Arrival Delay And Airport On-time Performance. Electron. Theses Diss.

Balakrishnama, S., Ganapathiraju, A., 1998. Linear discriminant analysis-a brief tutorial. Inst. Signal Inf. Process. 18.

Barboza, F., Kimura, H., Altman, E., 2017. Machine learning models and bankruptcy prediction. Expert Syst. Appl. 83, 405–417. https://doi.org/10.1016/j.eswa.2017.04.006

Big opportunities, big challenges [WWW Document], 2014. URL http://www.ey.com/gl/en/services/advisory/ey-big-data-big-opportunities-big-challenges (accessed 9.26.17).

Bigus, J.P., 1996. Data Mining with Neural Networks: Solving Business Problems from Application Development to Decision Support. McGraw-Hill, Inc., Hightstown, NJ, USA.

Bilalli, B., Abelló, A., Aluja-Banet, T., Wrembel, R., 2017. Intelligent assistance for data pre-processing. CSI Comput. Stand. Interfaces.

Bineid, M., Fielding, J.P., 2003. Development of a civil aircraft dispatch reliability prediction methodology. Aircr. Eng. Aerosp. Technol. 75, 588–594. https://doi.org/10.1108/00022660310503066

Böhm, C., Krebs, F., 2004. The k-Nearest Neighbour Join: Turbo Charging the KDD Process. Know Inf Sys Knowl. Inf. Syst. 6, 728–749.

Boswell, S.B., Evans, J.E., 1997. ANALYSIS OF DOWNSTREAM IMPORTS OF AIR TRAFFIC DELAY.

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. 30, 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2

Bratu, S., Barnhart, C., 2006. Flight operations recovery: New approaches considering passenger recovery. J. Sched. 9, 279–298. https://doi.org/10.1007/s10951-006-6781-0

Breiman, L., 1984. Classification and regression trees. Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, Calif.

Buck, S., Lei, Z., 2004. Charter Airlines: Have They a Future? Tour. Hosp. Res. 5, 72–78. https://doi.org/10.1057/palgrave.thr.6040007

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., Varoquaux, G., 2013. API design for machine learning software: experiences from the scikit-learn project. ArXiv13090238 Cs.

Cao, W., Fang, X., 2012. Airport Flight Departure Delay Model on Improved BN Structure Learning. Phys. Procedia, 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012) 33, 597–603. https://doi.org/10.1016/j.phpro.2012.05.109

Carrizosa, E., Martin-Barragan, B., Morales, D.R., 2010. Binarized Support Vector Machines. Inf. J. Comput. Inf. J. Comput. 22, 154–167.

Caruana, R., Niculescu-Mizil, A., 2006. An Empirical Comparison of Supervised Learning Algorithms, in: Proceedings of the 23rd International Conference on Machine Learning, ICML '06. ACM, New York, NY, USA, pp. 161–168. https://doi.org/10.1145/1143844.1143865

Chan, T.F., Golub, G.H., LeVeque, R.J., 1982. Updating Formulae and a Pairwise Algorithm for Computing Sample Variances, in: COMPSTAT 1982 5th Symposium Held at Toulouse 1982. Physica, Heidelberg, pp. 30–41. https://doi.org/10.1007/978-3-642-51461-6_3

Chandrashekar, G., Sahin, F., 2014. A survey on feature selection methods. Comput. Electr. Eng., 40th-year commemorative issue 40, 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Coakley, J.R., Brown, C.E., 2000. Artificial neural networks in accounting and finance: modeling issues. Int J Intell Syst Acc Fin Mgmt Int. J. Intell. Syst. Account. Finance Manag. 9, 119–144.

Constantin, C., 2015. Using the Logistic Regression model in supporting decisions of establishing marketing strategies. Bull. Transilv. Univ. Brasov Ser. V Econ. Sci. 8(57), 43–50.

Constantin, C., 2014. Principal Component Analysis-A Powerful Tool in Computing Marketing Information. Bull. Transilv. Univ. Brasov Econ. Sci. Ser. V 7, 25.

Cook, A., Tanner, G., Lawes, A., 2012. The Hidden Cost of Airline Unpunctuality. J. Transp. Econ. Policy JTEP 46, 157–173.

Cortes, C., Mohri, M., 2004. AUC optimization vs. error rate minimization, in: Advances in Neural Information Processing Systems. pp. 313–320.

Crone, S.F., Lessmann, S., Stahlbock, R., 2006. The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. Eur. J. Oper. Res. 173, 781–800. https://doi.org/10.1016/j.ejor.2005.07.023

Dattaram, B.A., Madhusudanan, N., 2016. Delay Prediction of Aircrafts Based on Health Monitoring Data. IJBAI Int. J. Bus. Anal. Intell. 4.

Davenport, T.H., 2013. Analytics 3.0 [WWW Document]. Harv. Bus. Rev. URL https://hbr.org/2013/12/analytics-30 (accessed 7.9.17).

Demšar, J., 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7, 1–30.

Deshpande, V., Arıkan, M., 2012. The Impact of Airline Flight Schedules on Flight Delays. Manuf. Serv. Oper. Manag. 14, 423–440. https://doi.org/10.1287/msom.1120.0379

Dimopoulos, C., Lefteris Manousakis, Georgios Pligoropoulos, 2017a. On-time Performance of Commercial Air Travel [WWW Document]. Scribd. URL https://www.scribd.com/document/347753572/Predicting-Delayed-Flights-On-time-Performance-of-Commercial-Air-Travel (accessed 9.13.17).

Dimopoulos, C., Manousakis, L., Pligoropoulos, G., 2017b. On-time performance of commercial air travel.

Ding, S., Zhao, H., Zhang, Y., Xu, X., Nie, R., 2015. Extreme learning machine: algorithm, theory and applications. Artif Intell Rev Artif. Intell. Rev. 44, 103–115.

Dogan, N., Tanrikulu, Z., 2013. A comparative analysis of classification algorithms in data mining for accuracy, speed and robustness. Inf. Technol. Manag. 14, 105–124. https://doi.org/10.1007/s10799-012-0135-8

Domingos, P., 2012. A few useful things to know about machine learning. Commun ACM Commun. ACM 55, 78.

Duda, R.O., Hart, P.E., Stork, D.G., 2012. Pattern Classification. John Wiley & Sons.

Dunham, M.H., 2002. Data Mining: Introductory and Advanced Topics, 1 edition. ed. Pearson, Upper Saddle River, N.J.

European Parliament, C. of the E.U., n.d. Regulation (EC) No 261/2004 of the European Parliament and of the Council of 11 February 2004 establishing common rules on compensation and assistance to passengers in the event of denied boarding and of cancellation or long delay of flights, and repealing Regulation (EEC) No 295/91 (Text with EEA relevance) - Commission Statement [WWW Document]. URL http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32004R0261 (accessed 8.26.17).

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognit. Lett., ROC Analysis in Pattern Recognition 27, 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Ferguson, J., Kara, A.Q., Hoffman, K., Sherry, L., 2013. Estimating domestic US airline cost of delay based on European model. Transp. Res. Part C Emerg. Technol. 33, 311–323. https://doi.org/10.1016/j.trc.2011.10.003

Fildes, R., Nikolopoulos, K., Crone, S.F., Syntetos, A.A., 2008. Forecasting and operational research: a review. J. Oper. Res. Soc. 59, 1150–1172. https://doi.org/10.1057/palgrave.jors.2602597

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugen. 7, 1936–37.

Forbes, S.J., 2008. The effect of air traffic delays on airline prices. Int. J. Ind. Organ. 26, 1218–1232. https://doi.org/10.1016/j.ijindorg.2007.12.004

Foulds, J., Frank, E., 2010. A review of multi-instance learning assumptions. Knowl. Eng. Rev. Knowl. Eng. Rev. 25, 1.

Fukunaga, K., 2013. Introduction to Statistical Pattern Recognition. Academic Press.

Gürbüz, F., Özbakir, L., Yapici, H., 2011. Data mining and preprocessing application on component reports of an airline company in Turkey. ESWA Expert Syst. Appl. 38, 6618–6626.

Hamby, D.M., 1993. A Numerical Comparison of Sensitivity Analysis Techniques (No. WSRC-MS--93-586). Westinghouse Savannah River Co., Aiken, SC (United States). https://doi.org/10.2172/10127567

Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. Elsevier.

Hansen, M.M., Gillen, D., Djafarian-Tehrani, R., 2001. Aviation infrastructure performance and airline cost: a statistical cost estimation approach. Transp. Res. Part E Logist. Transp. Rev. 37, 1–23. https://doi.org/10.1016/S1366-5545(00)00008-9

Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A., 2014. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. Int. J. Prod. Econ. 154, 72–80. https://doi.org/10.1016/j.ijpe.2014.04.018

Hodge, V.J., Austin, J., 2005. A binary neural k-nearest neighbour technique. Knowl. Inf. Syst. 8, 276–291.

Howley, T., Madden, M.G., O'Connell, M.-L., Ryder, A.G., 2006. The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data, in: Applications and Innovations in Intelligent Systems XIII. Springer, London, pp. 209–222. https://doi.org/10.1007/1-84628-224-1_16

Hsu, C.-W., Chang, C.-C., Lin, C.-J., 2006. A practical guide to support vector classification. Department of Computer Science, National Taiwan University, Taipei 106, Taiwan.

IATA, 2016. IATA Forecasts Passenger Demand to Double Over 20 Years [WWW Document]. URL http://www.iata.org/pressroom/pr/Pages/2016-10-18-02.aspx (accessed 9.9.17).

IATA, n.d. Consumer Issues - Flight Delays [WWW Document]. URL http://www-qa.iata.org/policy/consumer-pax-rights/consumer-protection/Pages/flight-delays.aspx (accessed 9.27.17).

Ionescu, L., Gwiggner, C., Kliewer, N., 2016. Data Analysis of Delays in Airline Networks. Bus. Inf. Syst. Eng. 58, 119–133. https://doi.org/10.1007/s12599-015-0391-3

Iooss, B., Lemaître, P., 2015. A Review on Global Sensitivity Analysis Methods, in: Uncertainty Management in Simulation-Optimization of Complex Systems, Operations Research/Computer Science Interfaces Series. Springer, Boston, MA, pp. 101–122. https://doi.org/10.1007/978-1-4899-7547-8_5

Japkowicz, N., Drummond, C., Elazmeh, W., American Association for Artificial Intelligence, AAAI Workshop (Eds.), 2006. Evaluation methods for machine learning: papers from the 2006 AAAI Workshop, July 17, 2006, Boston, Mass. AAAI Press, Menlo Park, Calif.

Jishan, S.T., Rashu, R.I., Haque, N., Rahman, R.M., 2015. Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique. Decis Anal Decis. Anal. 2, 1–25.

Jolliffe, I.T., 1986. Principal Component Analysis, Springer Series in Statistics. Springer New York, New York, NY. https://doi.org/10.1007/978-1-4757-1904-8

Kelleher, J.D., Namee, B.M., D'Arcy, A., 2015. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. MIT Press.

Kim, W.C., Mauborgne, R., 2004. Blue Ocean Strategy [WWW Document]. Harv. Bus. Rev. URL https://hbr.org/2004/10/blue-ocean-strategy (accessed 7.9.17).

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai. Stanford, CA, pp. 1137–1145.

Kohavi, R., Rothleder, N.J., Simoudis, E., 2002. Emerging Trends in Business Analytics. Commun ACM 45, 45–48. https://doi.org/10.1145/545151.545177

Kotler, P.T., Keller, K.L., 2011. Marketing Management, 14 edition. ed. Pearson, Upper Saddle River, N.J.

Kreibich, D.A., 2017. Flightright now : development of a predictive legal service for the compensation of flight delays in realtime.

Krycha, K.A., Wagner, U., 1999. Applications of artificial neural networks in management science: a survey. J. Retail. Consum. Serv. 6, 185–203. https://doi.org/10.1016/S0969-6989(98)00006-X

Kurgan, L.A., Musilek, P., 2006. A survey of knowledge discovery and data mining process models. Knowl. Eng. Rev. 21, 1–24.

Laws, E., 1997. Managing packaged tourism: Relationships, responsibilities and service quality in the inclusive holiday industry. Thomson Learning.

Lee, Y.-J., 2010. Neural network based approach for predicting learning effect in design students. Int. Assoc. Organ. Innov. 2, 250–270.

Li, T., Zhu, S., Ogihara, M., 2006. Using discriminant analysis for multi-class classification: an experimental investigation. Knowl. Inf. Syst. 10, 453–472.

Liu, Y., Bi, J.-W., Fan, Z.-P., 2017. Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. Expert Syst. Appl. 80, 323–339. https://doi.org/10.1016/j.eswa.2017.03.042

Marbán, Ó., Mariscal, G., Segovia, J., 2009. A Data Mining & Knowledge Discovery Process Model. https://doi.org/10.5772/6438

Martinez, V., 2012. Flight Delay Prediction. https://doi.org/10.3929/ethz-a-007139937

McGrath, R.G., 2014. To Make Better Decisions, Combine Datasets [WWW Document]. Harv. Bus. Rev. URL https://hbr.org/2014/09/to-make-better-decisions-combine-datasets (accessed 8.26.17).

Moro, S., Cortez, P., Rita, P., 2015. Using customer lifetime value and neural networks to improve the prediction of bank deposit subscription in telemarketing campaigns. Neural Comput. Appl. 26, 131–139. https://doi.org/10.1007/s00521-014-1703-0

Moro, S., Cortez, P., Rita, P., 2014. A data-driven approach to predict the success of bank telemarketing. Decis. Support Syst. 62, 22–31. https://doi.org/10.1016/j.dss.2014.03.001

Mullainathan, S., Spiess, J., 2017. Machine Learning: An Applied Econometric Approach. J. Econ. Perspect. J. Econ. Perspect. 31, 87–106.

Munson, M.A., 2012. A study on the importance of and time spent on different modeling steps. SIGKDD Explor Newsl ACM SIGKDD Explor. Newsl. 13, 65.

Needell, D., Srebro, N., Ward, R., 2016. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. Math Program Math. Program. Publ. Math. Optim. Soc. 155, 549–573.

Palmer, A., Montaño, J.., Franconetti, F.., 2008. Sensitivity Analysis Applied to Artificial Neural Networks for Forecasting Time Series. Methodol. Methodol. 4, 80–86.

Papadakos, N., 2009. Integrated airline scheduling. Comput. Oper. Res. 36, 176–195. https://doi.org/10.1016/j.cor.2007.08.002

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Silva, A.P.D., 2017. Optimization approaches to Supervised Classification. Eur. J. Oper. Res. 261, 772–788. https://doi.org/10.1016/j.ejor.2017.02.020

Poleto, T., Carvalho, V., Costa, A., 2015. The Roles of Big Data in the Decision-Support Process: An Empirical Investigation. Lect. Notes Bus. Inf. Process. 216, 10–21. https://doi.org/10.1007/978-3-319-18533-0_2

Prati, R.C., Batista, G.E., Monard, M.C., 2011. A Survey on Graphical Methods for Classification Predictive Performance Evaluation. IEEE Trans. Knowl. DATA Eng. 23, 1601–1618.

Provost, F., Kolluri, V., 1999. A Survey of Methods for Scaling Up Inductive Algorithms. Data Min. Knowl. Discov. 3, 131–169. https://doi.org/10.1023/A:1009876119989

Provost, F.J., Fawcett, T., Kohavi, R., others, 1998. The case against accuracy estimation for comparing induction algorithms., in: ICML. pp. 445–453.

Ramírez-Gallego, S., García, S., Krawczyk, B., Woźniak, M., Herrera, F., 2017. A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing Neurocomputing 239, 39–57.

Raschka, S., 2017. python-machine-learning-book: The "Python Machine Learning (1st edition)" book code repository and info resource.

Rebollo, J.J., Balakrishnan, H., 2014. Characterization and prediction of air traffic delays. Transp. Res. Part C Emerg. Technol. 44, 231–241. https://doi.org/10.1016/j.trc.2014.04.007

Refaeilzadeh, P., Tang, L., Liu, H., 2009. Cross-validation. Encycl. Database Syst. 532–538.

Rezaeinasab, M., Rad, M., 2008. Analytical survey of human rabies and animal bite prevalence during one decade in the province of Kerman, Iran. Crit. Care 12, P1. https://doi.org/10.1186/cc6222

Roberto, B.S., 2015. Dry Relaxation Shrinkage Prediction of Bordeaux Fiber Using a Feed Forward Neural. World Acad. Sci. Eng. Technol. Int. J. Chem. Mol. Nucl. Mater. Metall. Eng. 9, 1160–1165.

Rooij, A.J.F., Jain, L.C., Johnson, R.P., 1996. Neural Network Training Using Genetic Algorithms. World Scientific.

Rosenberger, J.M., Schaefer, A.J., Goldsman, D., Johnson, E.L., Kleywegt, A.J., Nemhauser, G.L., 2002. A Stochastic Model of Airline Operations. Transp. Sci. 36, 357–377. https://doi.org/10.1287/trsc.36.4.357.551

Rosenfeld, B., Lewis, C., 2005. Assessing Violence Risk in Stalking Cases: A Regression Tree Approach. Law Hum. Behav. Law Hum. Behav. 29, 343–357.

Rumelhart, D., 1994. The basic ideas in neural networks. Commun. ACM 37.

Salunke, S.S., Deshpande, S., 2015. Improving Classification Accuracy Using Weighted Multiple Regression. Int. J. Eng. Comput. Sci. 4.

Santafe, G., Inza, I., Lozano, J.A., 2015. Dealing with the evaluation of supervised classification algorithms. Artif Intell Rev Artif. Intell. Rev. Int. Sci. Eng. J. 44, 467–508.

Santos, B.F., Wormer, M.M.E.C., Achola, T.A.O., Curran, R., 2017. Airline delay management problem with airport capacity constraints and priority decisions. J. Air Transp. Manag. 63, 34–44. https://doi.org/10.1016/j.jairtraman.2017.05.003

Sayeh, W., Annie, B., 2014. Neural network versus logistic regrssion: the best accuracy in predicting credit rationing decision. Conf. World Finance Bank. Symp.

Schumacher, P., Olinsky, A., Quinn, J., Smith, R., 2010. A Comparison of Logistic Regression, Neural Networks, and Classification Trees Predicting Success of Actuarial Students. J. Educ. Bus. 85, 258–263. https://doi.org/10.1080/08832320903449477

Shearer, C., 2000. The CRISP-DM Model: The new blueprint for data mining. J. Data Warehous. 5, 13–22.

Shen, L., Tan, E.C., 2005. PLS and SVD based penalized logistic regression for cancer classification using microarray data., in: APBC. pp. 219–228.

Sternberg, A., Carvalho, D., Murta, L., Soares, J., Ogasawara, E., 2016. An analysis of Brazilian flight delays based on frequent patterns. Transp. Res. Part E Logist. Transp. Rev. 95, 282–298. https://doi.org/10.1016/j.tre.2016.09.013

Sternberg, A., Soares, J., Carvalho, D., Ogasawara, E., 2017. A Review on Flight Delay Prediction. ArXiv170306118 Cs.

Storey, D.J., Keasey, K., Watson, R., Wynarczyk, P., 2016. The Performance of Small Firms: Profits, Jobs and Failures. Routledge.

Tan, A.C., Gilbert, D., 2003. An Empirical Comparison of Supervised Machine Learning Techniques in Bioinformatics, in: Proceedings of the First Asia-Pacific Bioinformatics Conference on Bioinformatics 2003 - Volume 19, APBC '03. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 219–222.

Tan, P.-N., Steinbach, M., Kumar, V., 2005. Introduction to Data Mining, (First Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Taylor, S., 1994. Waiting for Service: The Relationship between Delays and Evaluations of Service. J. Mark. 58, 56–69. https://doi.org/10.2307/1252269

Tipping, M., Bishop, C., 1999a. Mixtures of Probabilistic Principal Component Analyzers. Neural Comput. 11, 443.

Tipping, M., Bishop, C., 1999b. Probabilistic Principal Component Analysis. J. R. Stat. Soc. Ser. B Stat. Methodol. 61, 611–622.

Twagilimana, J., 2006. Combining Data Mining and Statistical Techniques for Analysis of Outcomes in a Hospital Emergency Department. University of Louisville.

Vincent Granville, 2015. Using Historical Data in the Age of Real Time Decision Making [WWW Document]. URL http://www.datasciencecentral.com/profiles/blogs/using-historical-data-in-the-age-of-real-time-decision-making (accessed 9.26.17).

Webb, A.R., 2002. Statistical Pattern Recognition, 2 edition. ed. Wiley, West Sussex, England ; New Jersey.

Widrow, B., Rumelhart, D.E., Lehr, M.A., 1994. Neural networks: applications in industry, business and science. Commun ACM Commun. ACM 37, 93–105.

Williams, G., 2001. Will Europe's charter carriers be replaced by "no-frills" scheduled airlines? J. Air Transp. Manag., Developments in the Deregulated Air Transport Market 7, 277–286. https://doi.org/10.1016/S0969-6997(01)00022-9

Wirth, R., Hipp, J., 2000. CRISP-DM: Towards a standard process model for data mining, in: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining. pp. 29–39.

Wu, C.-L., 2006. Improving Airline Network Robustness and Operational Reliability by Sequential Optimisation Algorithms. Netw. Spat. Econ. 6, 235–251. https://doi.org/10.1007/s11067-006-9282-y

Xie, Z.-X., Hu, Q.-H., Yu, D.-R., 2006. Improved Feature Selection Algorithm Based on SVM and Correlation, in: Advances in Neural Networks - ISNN 2006, Lecture Notes in Computer Science. Presented at the International Symposium on Neural Networks, Springer, Berlin, Heidelberg, pp. 1373–1380. https://doi.org/10.1007/11759966_204

Yadav, R.N., Kalra, P.K., John, J., 2006. On the use of Multiplicative Neuron in Feedforward Neural Networks. Int. J. Model. Simul. J. Int. Assoc. Sci. Technol. Dev. IASTED 26, 331–336.

Yang, J., Olafsson, S., 2006. Optimization-based feature selection with adaptive instance sampling. Comput. Oper. Res., Part Special Issue: Operations Research and Data Mining 33, 3088–3106. https://doi.org/10.1016/j.cor.2005.01.021

Yimga, J., 2017. Airline on-time performance and its effects on consumer choice behavior. Res. Transp. Econ. https://doi.org/10.1016/j.retrec.2017.06.001

Youn, H., Gu, Z., 2010. Predict US restaurant firm failures: The artificial neural network model versus logistic regression model. Tour. Hosp. Res. 10, 171–187. https://doi.org/10.2307/23745462

Zhang, D., Tsai, J.J.P., 2003. Machine Learning and Software Engineering. Softw. Qual. J. 11, 87–119. https://doi.org/10.1023/A:1023760326768

Zhou, Z.-H., Li, M., 2010. Semi-supervised learning by disagreement. Knowl Inf Syst Knowl. Inf. Syst. Int. J. 24, 415–439.

8. Annexes

A - Data collection and preparation

[]:

```
#Data preparation

import sys

import random

import numpy as np

import pandas as pd

# Data

df = pd.read_csv('FLIGHT2014-2017.csv', header = 0)

#  Delete instances with empty values

df = df[pd.notnull(df['Date'])]

df = df[pd.notnull(df['ATD'])]

# Delete columns not relevant

df.drop(df.columns[[0, 5, 8, 13, 14, 16, 17, 19, 23, 25, 30]], axis=1, inplace=True)

df.drop(df.columns[[5]], axis=1, inplace=True)

# Delete instances not relevant

df[df.Tail_No.str.contains("ACMI1") == False]

df[df.Tail_No.str.contains("ACMI2") == False]

df[df.Tail_No.str.contains("CS--TQP") == False]



df.shape

#Shape after initial transformation (6907, 34)

#Create a new csv file

df.to_csv('flights_DF.csv', index=False, header=True)
```

In []:

```
# from reportlab.pdfgen import canvas
```

```
from reportlab.lib.units import inch, cm

c = canvas.Canvas('ex.pdf')

c.drawImage('ar.jpg', 0, 0, 10*cm, 10*cm)

c.showPage()

c.save()
```

B - Modelling

In [1]:

```
url = "flights_REGRE.csv"

X = pd.read_csv(url)
```

In [2]:

```
X.shape
```

Out[273]:

(6907, 25)

In [3]:

```
X.describe()
```

In [4]:

```
list(X)
```

Out[5]:

```
['MONTH',

'DAY',

'WEEK',

'SEASON',

'ENG_NUMBER',

'Tail_No',

'Private_Carrier',

'STC',

'Dep',

'DEP_HOUR',

'ARR_HOUR',

'Flight_time',
```

```
'DEP_DELAY',

'ARR_DELAY',

'Client_ID',

'Flight_more10hrs',

'If_Dep_night_OFFICE',

'YEAR',

'Weekend',

'If_Arrival_night_OFFICE',

'flight_during_OFFICENIGHT',

'AC_age',

'If_previous_flight_delayed',

'Previous_flight_delay',

'Previous_flight_delay_reason']
```

In [6]:

```
X_1 = X[['Previous_flight_delay', 'Flight_time']]
```

In [7]:

```
MONTH_1 = X[['MONTH']] # Categorical features

MONTH = pd.get_dummies(MONTH_1, columns=["MONTH"], prefix=["MONTH-"])
```

In [8]:

```
WEEK_1 = X[['WEEK']] # Categorical features

WEEK = pd.get_dummies(WEEK_1, columns=["WEEK"], prefix=["WEEK-"])
```

In [9]:

```
SEASON_1 = X[['SEASON']] # Categorical features

SEASON = pd.get_dummies(SEASON_1, columns=["SEASON"], prefix=["SEASON-"])
```

In [10]:

```
Tail_No_1 = X[['Tail_No']] # Categorical features

Tail_No = pd.get_dummies(Tail_No_1, columns=["Tail_No"], prefix=["Tail_No-"])
```

In [11]:

```
Flight_more10hrs_1 = X[['Flight_more10hrs']] # Categorical features

Flight_more10hrs = pd.get_dummies(Flight_more10hrs_1, columns=["Flight_more10hrs"],
prefix=["Flight_more10hrs-"])
```

In [12]:

```
Weekend_1 = X[['Weekend']] # Categorical features

Weekend = pd.get_dummies(Weekend_1, columns=["Weekend"], prefix=["Weekend-"])
```

In [13]:

```
If_Dep_night_OFFICE_1 = X[['If_Dep_night_OFFICE']] # Categorical features

If_Dep_night_OFFICE = pd.get_dummies(If_Dep_night_OFFICE_1, columns=["If_Dep_night_O
FFICE"], prefix=["If_Dep_night_OFFICE-"])
```

In [14]:

```
DEP_HOUR_1 = X[['DEP_HOUR']] # Categorical features

DEP_HOUR = pd.get_dummies(DEP_HOUR_1, columns=["DEP_HOUR"], prefix=["DEP_HOUR-"])
```

In [15]:

```
Private_Carrier_1 = X[[' Private_Carrier ']] # Categorical features

Private_Carrier = pd.get_dummies(Private_Carrier _1, columns=[" Private_Carrier "],
prefix=[" Private_Carrier -"])
```

In [16]:

```
Reason_1 = X[['Previous_flight_delay_reason']] # Categorical features

Reason = pd.get_dummies(Reason_1, columns=['Previous_flight_delay_reason'], prefix=[
"Reason-"])
```

In [17]:

```
STC_1 = X[['STC']] # Categorical features
```

In [18]:

```
STC = pd.get_dummies(STC_1, columns=["STC"], prefix=["STC-"])
```

In [19]:

```
Client_ID_1 = X[['Client_ID']]
```

In [20]:

```
Client_ID = pd.get_dummies(Client_ID_1, columns=["Client_ID"], prefix=["Client_ID-"]
)
```

In [21]:

```
#X = pd.concat([X_1, STC, Client_ID, A_C], axis=1)
```

In [22]:

```
DEP_1 = X[['Dep']] #Categorical features
```

In [23]:

```
DEP = pd.get_dummies(DEP_1, columns=["Dep"], prefix=["Dep-"])
```

In [24]:

```
X_big = pd.concat([X_1, STC, Client_ID, DEP, MONTH, WEEK, SEASON, Flight_more10hrs,
Tail_No, Weekend, If_Dep_night_OFFICE, DEP_HOUR, PRIVATE_CARRIER, Reason], axis=1)
```

In [25]:

```
#print (X.dtypes)
```

In [26]:

```
encoder = OneHotEncoder() # Create encoder object X needs to contain only non-negati
ve integers.
```

```
#X_1_encoded = encoder.fit_transform(X_1).toarray()
```

In [27]:

```
#scaler = StandardScaler().fit(X_1)

#scalerX = scaler.transform(X_1)
```

In [28]:

```
#from sklearn.preprocessing import MinMaxScaler

#scaler_2 = MinMaxScaler(feature_range=(0, 1))

#MinMaxX = scaler_2.fit_transform(X_1)
```

In [29]:

```
#from sklearn import preprocessing

#max_abs_scaler = preprocessing.MaxAbsScaler()

#X_abs = max_abs_scaler.fit_transform(X)
```

In [30]:

```
from sklearn.decomposition import PCA #Best Solution - used to offline colinearity

pca = PCA(n_components=53)

Xbig_PCA = pca.fit(X_big).transform(X_big)
```

In [31]:

```
Y15 = X['DEP_DELAY'].astype('category')
```

In [32]:

```
print(X_PCA)
```

In [33]:

```
#comparing models with variables from flight planning program

# prepare configuration for cross validation test harness
```

```
# prepare models

seed = 20

models = []

models.append(('MLPC', MLPClassifier()))

models.append(('LinReg', LinearRegression())) #Can't handle mix of binary and contin
uous

models.append(('LR', LogisticRegression()))

models.append(('LDA', LinearDiscriminantAnalysis())) #warnings.warn("Variables are c
ollinear

models.append(('KNN', KNeighborsClassifier()))

models.append(('CART', DecisionTreeClassifier()))

models.append(('NB', GaussianNB()))

models.append(('SVM', SVC()))

models.append(('SGD', SGDClassifier()))

# evaluate each model in turn

results = []

names = []

scoring = 'accuracy'

for name, model in models:

#        kfold = cross_validation.KFold(6907, n_folds=2, shuffle=True, random_state=
seed)

        kfold = cross_validation.StratifiedKFold(Y15, n_folds=10, shuffle= True, ra
ndom_state= seed)

        cv_results = cross_validation.cross_val_score(model, Xbig_PCA, Y15, cv=kfol
d, scoring=scoring)

        results.append(cv_results)

        names.append(name)

        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())

        print(msg)

# boxplot algorithm comparison

fig = plt.figure()
```

```
fig.suptitle('Algorithm Comparison')

ax = fig.add_subplot(111)

plt.boxplot(results)

ax.set_xticklabels(names)
```

C - Evaluation

C.1 - Supervised Classification algorithms accuracy comparison

In []:

```
#comparing models with variables from flight planning program
```

In []:

```
# prepare configuration for cross validation test harness

seed = 20

# prepare models

models = []

models.append(('MLPC', MLPClassifier()))

models.append(('LinReg', LinearRegression())) #Can't handle mix of binary and contin
uous

models.append(('LR', LogisticRegression()))

models.append(('LDA', LinearDiscriminantAnalysis())) #warnings.warn("Variables are c
ollinear

models.append(('KNN', KNeighborsClassifier()))

models.append(('CART', DecisionTreeClassifier()))

models.append(('NB', GaussianNB()))

models.append(('SVM', SVC()))

models.append(('SGD', SGDClassifier()))

# evaluate each model in turn

results = []

names = []

scoring = 'accuracy'

for name, model in models:

#        kfold = cross_validation.KFold(6907, n_folds=2, shuffle=True, random_state=
seed)

        kfold = cross_validation.StratifiedKFold(Y15, n_folds=10, shuffle= True, ra
ndom_state= seed)
```

```
        cv_results = cross_validation.cross_val_score(model, Xbig_PCA, Y15, cv=kfol
d, scoring=scoring)

        results.append(cv_results)

        names.append(name)

        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())

        print(msg)

# boxplot algorithm comparison

fig = plt.figure()

fig.suptitle('Algorithm Comparison')

ax = fig.add_subplot(111)

plt.boxplot(results)

ax.set_xticklabels(names)

plt.show()
```

```
MLPC: 0.672941 (0.020490)

LR: 0.676266 (0.013507)

LDA: 0.676121 (0.014639)

KNN: 0.674664 (0.021118)

CART: 0.634709 (0.016296)

NB: 0.656717 (0.013825)

SVM: 0.682343 (0.014549)

SGD: 0.564809 (0.057683)
```

In []:

```
#using neural network
```

In []:

```
predict_ml = MLPClassifier()

predPCA = predict_ml.fit(Xbig_PCA, Y15)
```

In []:

```
kfold = cross_validation.KFold(6907, n_folds=5, shuffle=True, random_state=seed)

scores = cross_val_score(predPCA, Xbig_PCA, Y15, cv=kfold)
```

In []:

```
print("mean: {:.3f} (std: {:.3f})".format(scores.mean(),scores.std()),end="\n\n" )
```

```
mean: 0.671 (std: 0.020)
```

In []:

```
predicted_MLPC = cross_validation.cross_val_predict(MLPClassifier(), Xbig_PCA, Y15,
cv=kfold)
```

In []:

```
from sklearn.metrics import accuracy_score

#print (accuracy_score(Y, predicted_MLPC))
```

In []:

```
from sklearn.metrics import classification_report,confusion_matrix

print (classification_report(Y15, predicted_MLPC))
```

```
             precision    recall  f1-score   support

        0.0       0.70      0.74      0.72      3975

        1.0       0.62      0.58      0.60      2932

avg / total       0.67      0.67      0.67      6907
```

In []:

```
print(confusion_matrix(Y15, predicted_MLPC))
```

```
[[2928 1047]

 [1236 1696]]
```

In []:

```
errors = predicted_MLPC != Y15

print("Nb errors=%i, error rate=%.2f" % (errors.sum(), errors.sum() / len(predicted_
MLPC)))
```

```
Nb errors=2283, error rate=0.33
```

In []:

```
from sklearn.metrics import roc_curve, auc

from sklearn import metrics

fpr, tpr, thresholds = roc_curve(Y15, predicted_MLPC)

roc_auc = metrics.auc(fpr, tpr)

plt.title('Receiver Operating Characteristic')

plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)

plt.legend(loc = 'lower right')

plt.plot([0, 1], [0, 1],'r--')

plt.xlim([0, 1])

plt.ylim([0, 1])

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()
```

In [ ]:

```
#using logistic regression
```

In []:

```
predicted = cross_validation.cross_val_predict(LogisticRegression(fit_intercept = Fa
lse, C = 1e9), Xbig_PCA, Y15, cv=kfold)
```

In []:

```
from sklearn.model_selection import cross_val_score
```

```
print(cross_val_score(LogisticRegression(fit_intercept = False, C = 1e9), Xbig_PCA,
Y, cv=kfold))
```

```
[ 0.66208394  0.67004342  0.6705286   0.69225199  0.67342505]
```

In []:

```
#scores
```

In []:

```
from sklearn.metrics import accuracy_score

print (accuracy_score(Y15, predicted))
```

```
0.665122339655
```

In []:

```
from sklearn.metrics import classification_report,confusion_matrix

print (classification_report(Y15, predicted))
```

```
             precision    recall  f1-score   support

        0.0       0.74      0.64      0.69      3975

        1.0       0.59      0.70      0.64      2932

avg / total       0.68      0.67      0.67      6907
```

In []:

```
print(confusion_matrix(Y15, predicted))
```

```
[[2556 1419]
 [ 894 2038]]
```

In []:

```
errors = predicted != Y15

print("Nb errors=%i, error rate=%.2f" % (errors.sum(), errors.sum() / len(predicted)
))
```

```
Nb errors=2313, error rate=0.33
```

In []:

```
from sklearn.metrics import roc_curve, auc

from sklearn import metrics

fpr, tpr, thresholds = roc_curve(Y, predicted)

roc_auc = metrics.auc(fpr, tpr)

plt.title('Receiver Operating Characteristic')

plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)

plt.legend(loc = 'lower right')

plt.plot([0, 1], [0, 1],'r--')

plt.xlim([0, 1])

plt.ylim([0, 1])

plt.ylabel('True Positive Rate')

plt.xlabel('False Positive Rate')

plt.show()
```

C.2 - Exploratory and Sensitivity Analysis

In [109]:

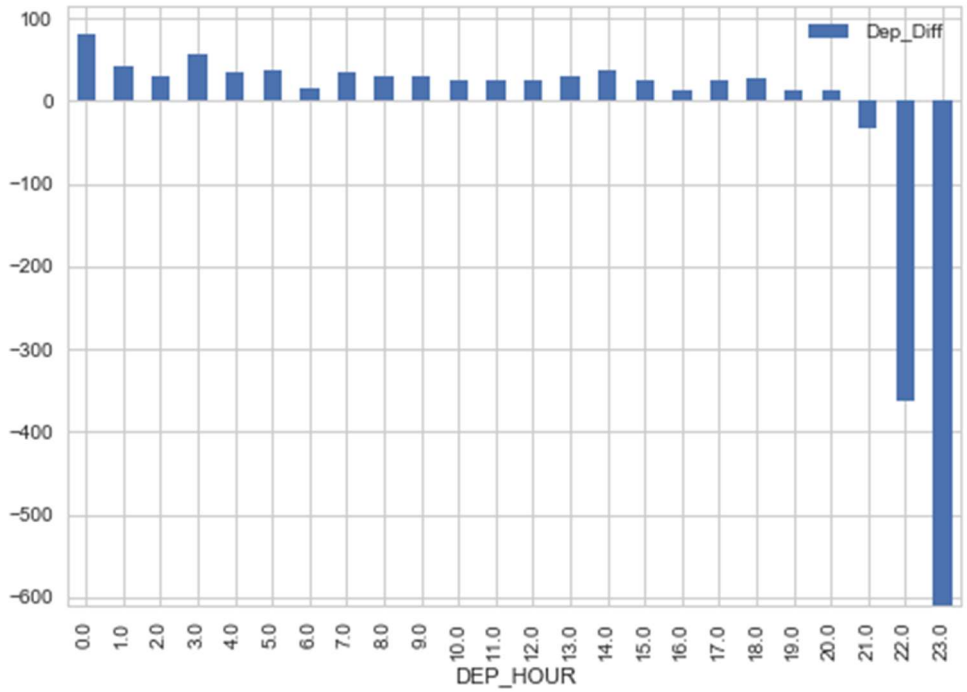
```
import warnings

warnings.filterwarnings('ignore')

import sys

import numpy as np

from scipy import stats, integrate

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib as mpl

#import plotly.plotly as py

%matplotlib inline
```

```
import seaborn as sns
```

In [268]:

```
# Compute average number of delayed flights per month

grouped = X[['Dep_Diff', 'DEP_HOUR']].groupby('DEP_HOUR').mean()

# plot average delays by month

grouped.plot(kind='bar')
```

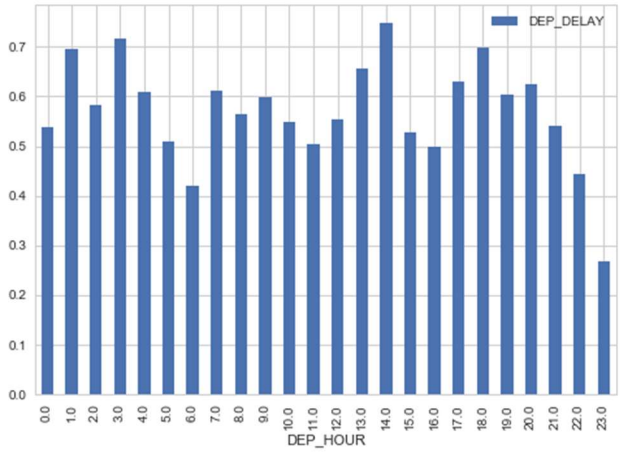
Out[268]:



In [269]:

```
# Compute average number of delayed flights per month

grouped = X[['DEP_DELAY', 'DEP_HOUR']].groupby('DEP_HOUR').mean()

# plot average delays by month

grouped.plot(kind='bar')
```

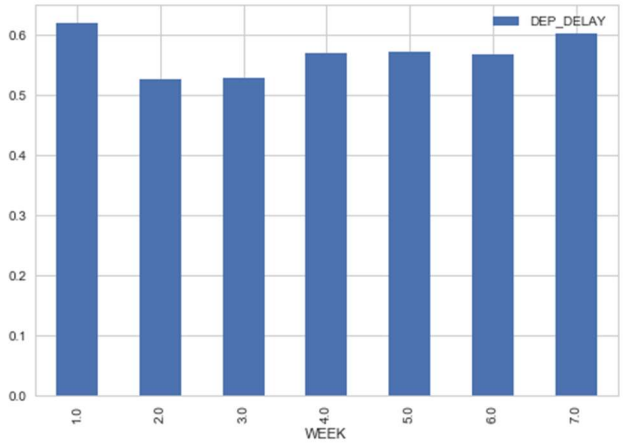
Out[269]:



In [234]:

```
grouped = X[['DEP_DELAY', 'WEEK']].groupby('WEEK').mean()

# plot average delays by month

grouped.plot(kind='bar')
```

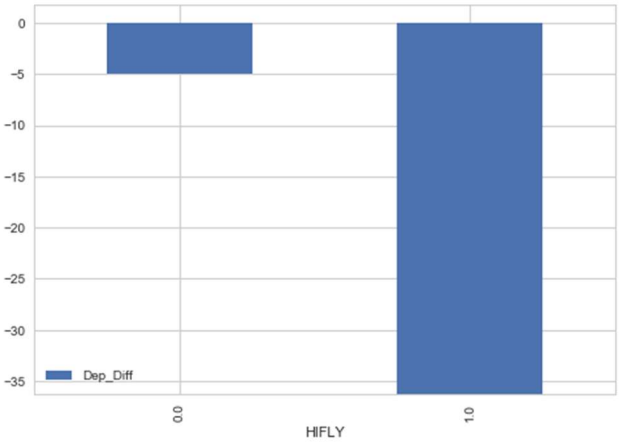
Out[234]:



In [236]:

```
grouped = X[['Dep_Diff', 'PRIVATE_CARRIER']].groupby('PRIVATE_CARRIER').mean()

# plot average delays by month

grouped.plot(kind='bar')
```

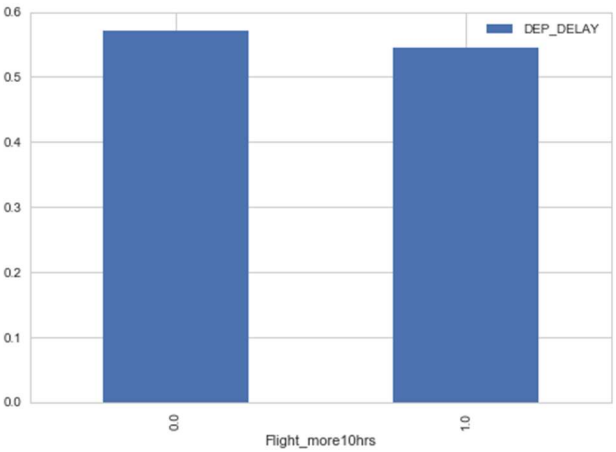
Out[236]:



In [286]:

```
grouped = X[['DEP_DELAY', 'Flight_more10hrs']].groupby('Flight_more10hrs').mean()

# plot average delays by month

grouped.plot(kind='bar')
```

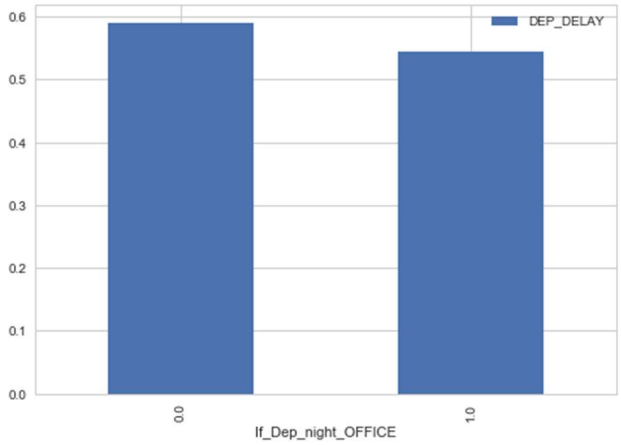
Out[286]:



In [238]:

```
grouped = X[['DEP_DELAY', 'If_Dep_night_OFFICE']].groupby('If_Dep_night_OFFICE').mea
n()

# plot average delays by month

grouped.plot(kind='bar')
```

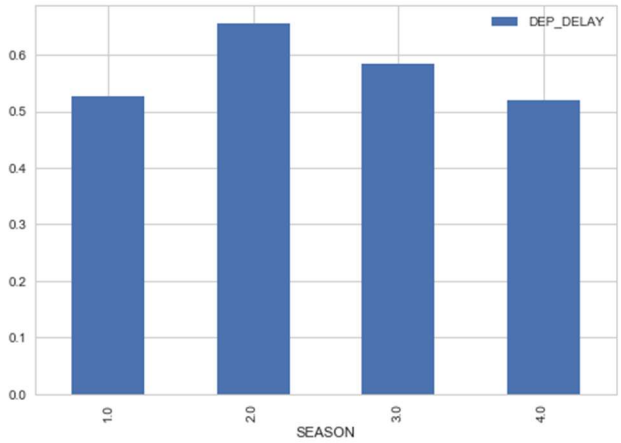
Out[238]:



In [239]:

```
grouped = X[['DEP_DELAY', 'SEASON']].groupby('SEASON').mean()

# plot average delays by month

grouped.plot(kind='bar')
```

Out[239]:

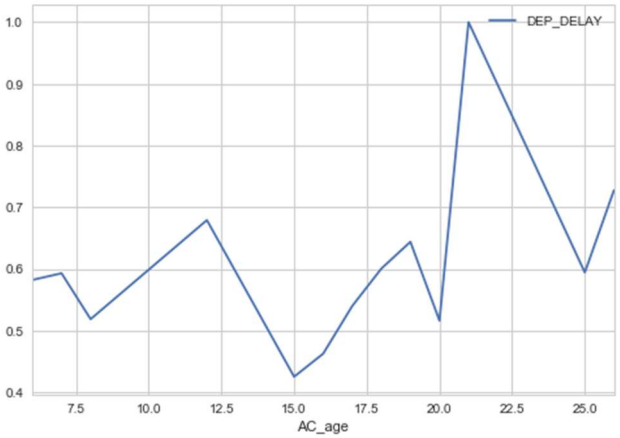


In [287]:

```
grouped = X[['DEP_DELAY', 'AC_age']].groupby('AC_age').mean()

# plot average delays by month

grouped.plot()

X[['DEP_DELAY', 'AC_age']].groupby('AC_age').mean()
```

Out[287]:

	DEP_DELAY
AC_age	
6	0.582164
7	0.592834
8	0.518519
12	0.679012
15	0.425159
16	0.462572
17	0.539866
18	0.600421
19	0.643854
20	0.516129
21	1.000000
25	0.594203
26	0.727273



In [288]:

```
X[['AC_age', 'DEP_DELAY']].groupby('DEP_DELAY').mean()
```

Out[288]:

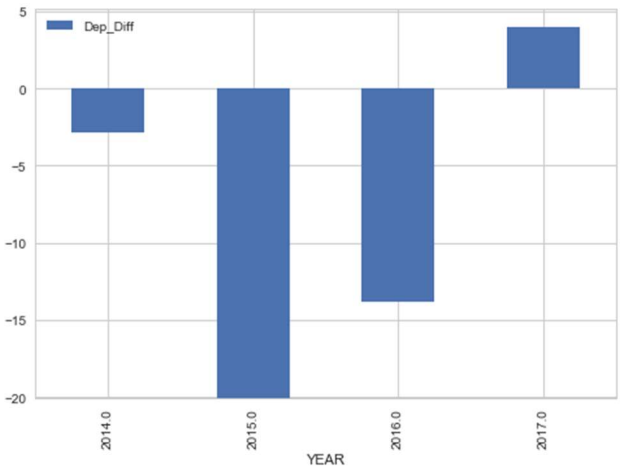
	AC_age
DEP_DELAY	
0.0	15.647434
1.0	15.781202

In [244]:

```
grouped = X[['Dep_Diff', 'YEAR']].groupby('YEAR').mean()

# plot average delays by month

grouped.plot(kind='bar')

X[['Dep_Diff', 'YEAR']].groupby('YEAR').mean()
```

Out[244]:

	Dep_Diff
YEAR	
2014.0	-2.833333
2015.0	-20.045806
2016.0	-13.836334
2017.0	3.961039

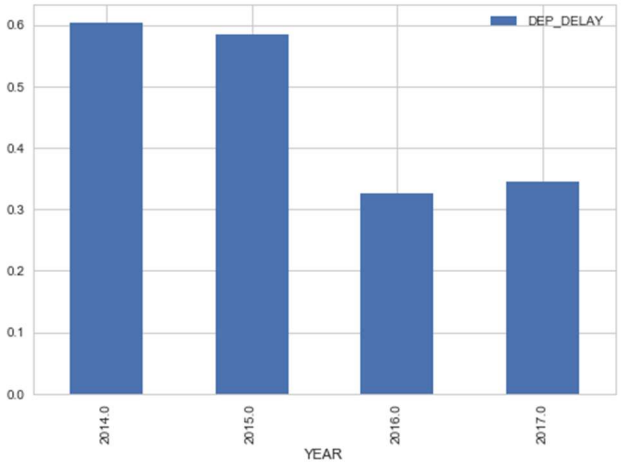


In [212]:

```
grouped = X[['DEP_DELAY', 'YEAR']].groupby('YEAR').mean()

grouped.plot(kind='bar')

X[['DEP_DELAY', 'YEAR']].groupby('YEAR').size()
```

Out[212]:

```
YEAR
2014.0    4596
2015.0    1550
2016.0     611
2017.0     154
```

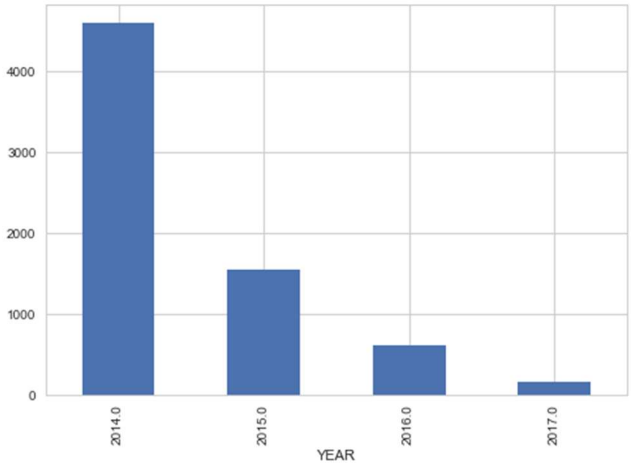


In [289]:

```
grouped = X[['DEP_DELAY', 'YEAR']].groupby('YEAR').size()

grouped.plot(kind='bar')

X[['DEP_DELAY', 'YEAR']].groupby('YEAR').size()
```

Out[289]:

```
YEAR
2014.0    4590
2015.0    1551
2016.0     612
2017.0     154
```



In [292]:

```
X[['YEAR', 'DEP_DELAY']].groupby('DEP_DELAY').size()
```

Out[292]:

```
DEP_DELAY
0.0    2981
1.0    3926
dtype: int64
```

In [297]:

```
X[['Dep_Diff', 'YEAR']].groupby('YEAR').mean()
```

Out[297]:

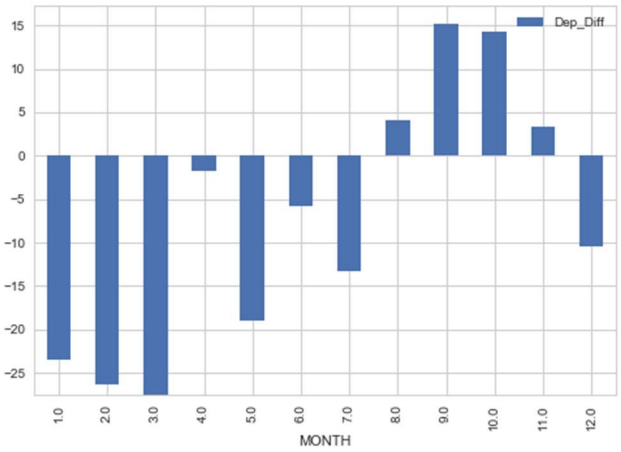
	Dep_Diff
YEAR	
2014.0	-2.864270
2015.0	-19.965184
2016.0	-13.184641
2017.0	3.961039

In [248]:

```
grouped = X[['Dep_Diff', 'MONTH']].groupby('MONTH').mean()

# plot average delays by month

grouped.plot(kind='bar')

X[['Dep_Diff', 'MONTH']].groupby('MONTH').mean()
```

Out[248]:

	Dep_Diff
MONTH	
1.0	-23.423462
2.0	-26.302655
3.0	-27.564263
4.0	-1.814450
5.0	-19.049206
6.0	-5.810169
7.0	-13.363803
8.0	4.153322
9.0	15.117566
10.0	14.312409
11.0	3.305195
12.0	-10.493939

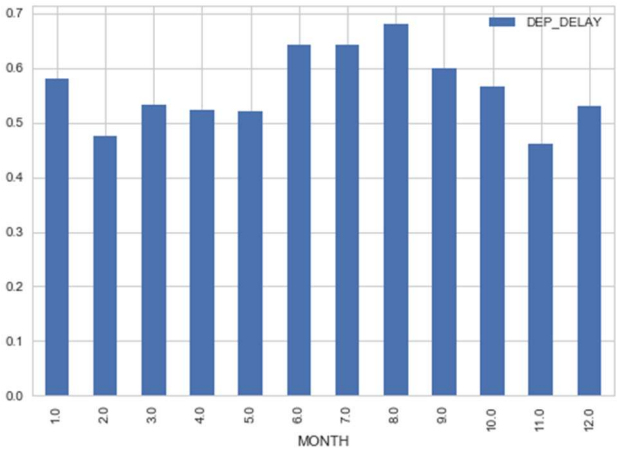


In [232]:

```
grouped = X[['DEP_DELAY', 'MONTH']].groupby('MONTH').mean()

# plot average delays by month

grouped.plot(kind='bar')

X[['DEP_DELAY', 'MONTH']].groupby('MONTH').mean()
```

Out[232]:

	DEP_DELAY
MONTH	
1.0	0.579399
2.0	0.474336
3.0	0.532915
4.0	0.523810
5.0	0.520635
6.0	0.642373
7.0	0.643510
8.0	0.679727
9.0	0.598893
10.0	0.566423
11.0	0.461039
12.0	0.530303

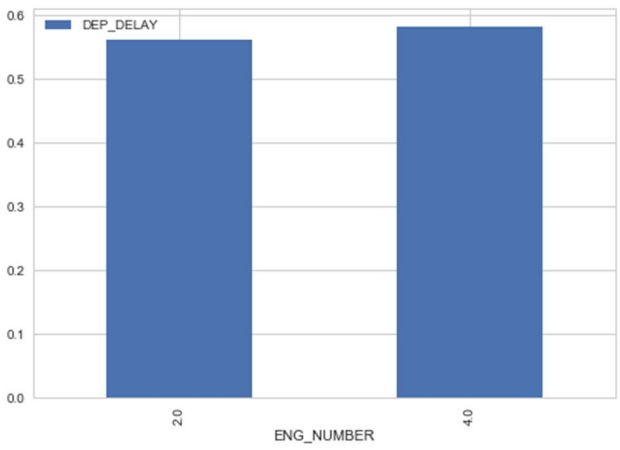


In [249]:

```
grouped = X[['DEP_DELAY', 'ENG_NUMBER']].groupby('ENG_NUMBER').mean()

# plot average delays by month

grouped.plot(kind='bar')

X[['DEP_DELAY', 'ENG_NUMBER']].groupby('ENG_NUMBER').mean()
```

Out[249]:

	DEP_DELAY
ENG_NUMBER	
2.0	0.560682
4.0	0.581840

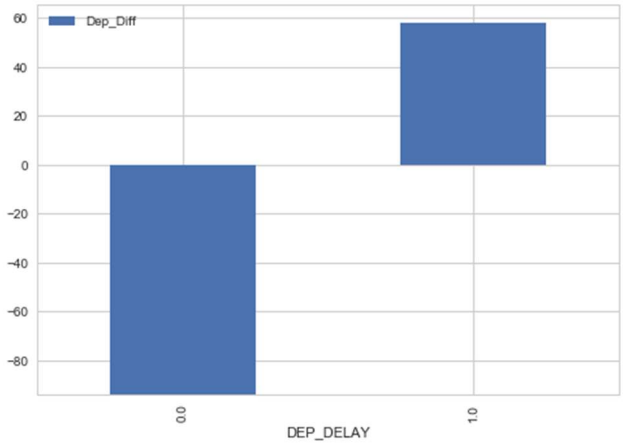


In [216]:

```
grouped = X[['Dep_Diff', 'DEP_DELAY']].groupby('DEP_DELAY').mean()

# plot average delays by month

grouped.plot(kind='bar')

X[['Dep_Diff', 'DEP_DELAY']].groupby('DEP_DELAY').mean()
```

Out[216]:

	Dep_Diff
DEP_DELAY	
0.0	-93.84948
1.0	58.04888



In [217]:

```
X_1.shape
```

Out[217]:

```
(6911, 16)
```

In [252]:

```
X.groupby(['Client_ID','DEP_DELAY']).size().unstack()
```

Out[252]:

DEP_DELAY	0.0	1.0
Client_ID		
1I	1.0	4.0
5H	249.0	314.0
5A	NaN	10.0
6B	2.0	13.0
AB	NaN	4.0
AF	6.0	13.0
AH	69.0	464.0
AS	364.0	162.0

DEP_DELAY	0.0	1.0
Client_ID		
AT	5.0	10.0
B0	1.0	5.0
BA	615.0	283.0
CU	50.0	99.0
DE	8.0	36.0
DK	2.0	14.0
DR	1.0	3.0
DT	7.0	16.0
DY	46.0	223.0
E9	1.0	2.0
EI	1.0	5.0
GL	13.0	25.0
GW	1.0	2.0
HQ	NaN	4.0
IG	1.0	3.0
JN	17.0	16.0
MD	1.0	12.0
ML	1.0	3.0
MT	4.0	56.0
N3	22.0	39.0
N9	1.0	2.0
OR	NaN	4.0
PV	3.0	3.0
PY	1.0	2.0
S4	4.0	6.0
SE	9.0	111.0
SK	2.0	4.0
SN	6.0	17.0
SS	12.0	34.0
ST	NaN	3.0
SV	674.0	658.0
TB	3.0	16.0
TO	1.0	5.0
TP	21.0	77.0
TX	2.0	14.0
V0	7.0	10.0
W3	390.0	288.0
WI	1.0	6.0
XY	355.0	826.0
ZB	2.0	2.0
ZT	1.0	NaN

In [219]:

```
X.groupby(['A/C','DEP_DELAY']).size().unstack()
```

Out[219]:

DEP_DEUAY	0.0	1.0
A/C		
MA-VEA	52.0	110.0
MA-VUN	NaN	4.0
MA-UQM	1.0	10.0

DEP_DEUAY	0.0	1.0
A/C		
HI-UEX	143.0	213.0
HI-UFW	184.0	154.0
HI-UFX	223.0	154.0
HI-UFZ	148.0	469.0
HI-UMU	297.0	431.0
HI-UQM	98.0	521.0
HI-UQP	158.0	266.0
HI-UQW	474.0	445.0
HI-UQY	151.0	153.0
HI-UQZ	341.0	355.0
HI-URI	142.0	381.0
HI-URJ	547.0	219.0

In [220]:

```
X.groupby(['DEP_DELAY','SEASON']).size().unstack()
```

Out[220]:

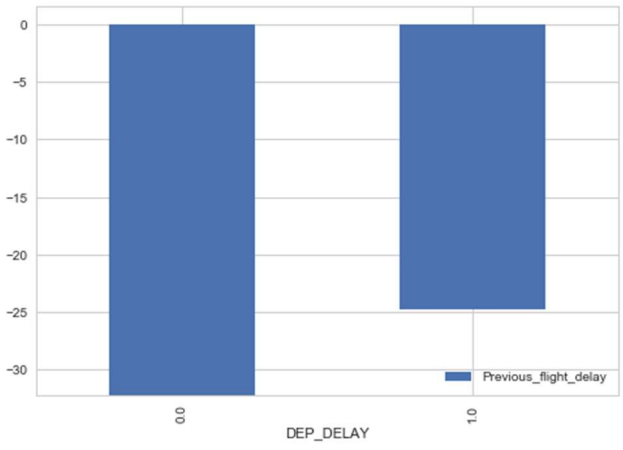
SEASON	1.0	2.0	3.0	4.0
DEP_DELAY				
0.0	890	594	587	912
1.0	987	1130	821	990

In [276]:

```
grouped = X[['Previous_flight_delay','DEP_DELAY']].groupby('DEP_DELAY').mean()

# plot average delays by month

grouped.plot(kind='bar')

X[['Previous_flight_delay','DEP_DELAY']].groupby('DEP_DELAY').mean()
```

Out[276]:

	Previous_flight_delay
DEP_DELAY	
0.0	-32.195236
1.0	-24.768467

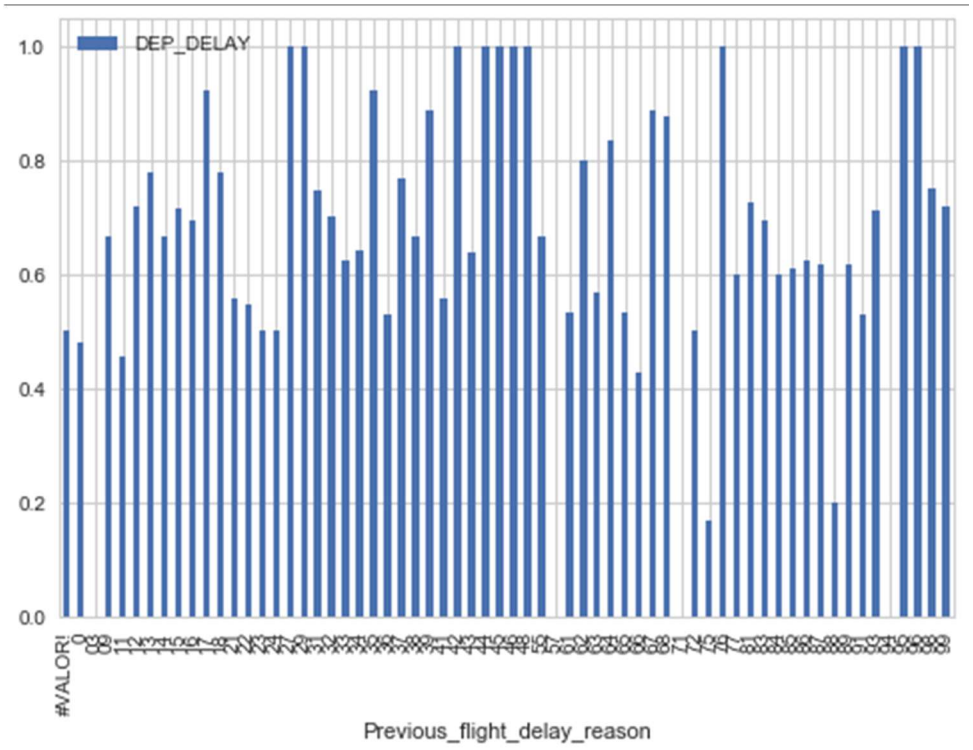


In [284]:

```
#'Previous_flight_delay_reason'

grouped_delayIataReason = X[['DEP_DELAY','Previous_flight_delay_reason']].groupby('P
revious_flight_delay_reason').mean()

# plot average delays by month

grouped_delayIataReason.plot(kind='bar')

delayIataReason = X[['DEP_DELAY','Previous_flight_delay_reason']].groupby('Previous_
flight_delay_reason').mean()

print(delayIataReason)
```

DEP_DELAY	Previous_flight_delay_reason		
0	0.479691	55	0.666667
3	0.000000	57	0.000000
9	0.666667	61	0.533333
11	0.454545	62	0.800000
12	0.718750	63	0.566038
13	0.777778	64	0.833333
14	0.666667	65	0.533333
15	0.715054	66	0.428571
16	0.695652	67	0.888889
17	0.923077	68	0.875000
18	0.777778	71	0.000000
21	0.555556	72	0.500000
22	0.545455	75	0.166667
23	0.500000	76	1.000000
24	0.500000	77	0.600000
27	1.000000	81	0.725806
29	1.000000	83	0.692308
31	0.746032	84	0.600000
32	0.700787	85	0.609756
33	0.625000	86	0.625000
34	0.640000	87	0.615385
35	0.923077	88	0.200000
36	0.527778	89	0.618182
37	0.769231	91	0.529412
38	0.666667	93	0.710394
39	0.888889	94	0.000000
41	0.555556	95	1.000000
42	1.000000	96	1.000000
43	0.636364	98	0.750000
...	...	99	0.717949

[64 rows x 1 columns]



In [281]:

```
delayIataReason.to_HIv('delayIataReason.HIv', index=False, header=True)
```

In [221]:

```
X.groupby(['DEP_DELAY','ARR_DELAY']).size().unstack()
```

Out[221]:

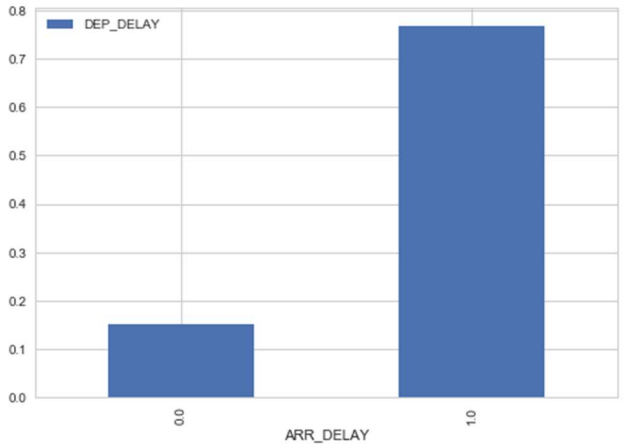
ARR_DELAY	0.0	1.0
DEP_DELAY		
0.0	1899	1084
1.0	335	3593

In [222]:

```
grouped = X[['DEP_DELAY','ARR_DELAY']].groupby('ARR_DELAY').mean()

# plot average delays by month

grouped.plot(kind='bar')

X[['DEP_DELAY','ARR_DELAY']].groupby('ARR_DELAY').mean()
```

Out[222]:

	DEP_DELAY
ARR_DELAY	
0.0	0.149955
1.0	0.768227



In [223]:

```
X["Dep_Diff"].mean()
```

Out[223]:

-7.515120821878165

In [224]:

```
X[['Dep_Diff', 'DEP_DELAY']].groupby('DEP_DELAY').mean()
```

Out[224]:

	Dep_Diff
DEP_DELAY	
0.0	-93.84948
1.0	58.04888

In [225]:

```
X[['Dep_Diff', 'DEP_DELAY']].groupby('DEP_DELAY').std()
```

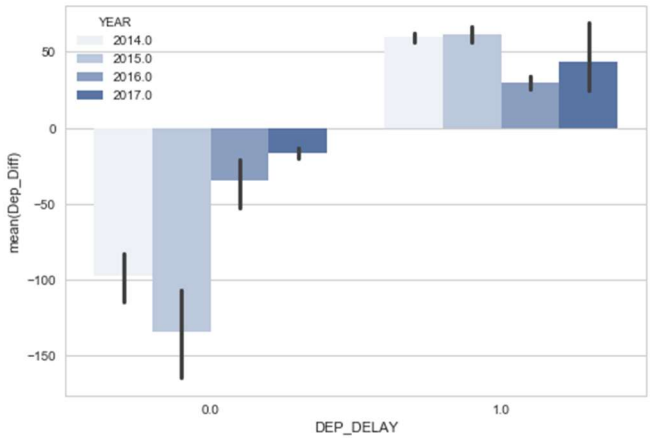
Out[225]:

	Dep_Diff
DEP_DELAY	
0.0	316.177782
1.0	84.365745

In [226]:

```
sns.set(style="whitegrid")

sns.barplot(x="DEP_DELAY", y="Dep_Diff", hue="YEAR", data=X, color="b")
```

Out[226]:



In [227]:

```
bin_values = np.arange(start=-100, stop=300, step=5)

print(bin_values)
```

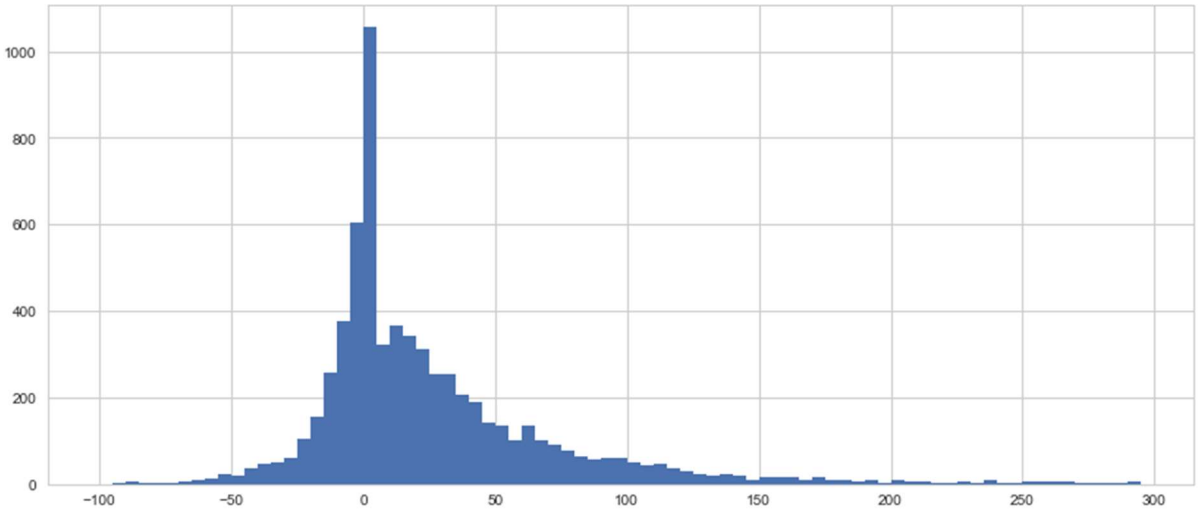
```
[-100  -95  -90  -85  -80  -75  -70  -65  -60  -55  -50  -45  -40  -35  -30
  -25  -20  -15  -10   -5    0    5   10   15   20   25   30   35   40   45
   50   55   60   65   70   75   80   85   90   95  100  105  110  115  120
  125  130  135  140  145  150  155  160  165  170  175  180  185  190  195
  200  205  210  215  220  225  230  235  240  245  250  255  260  265  270
  275  280  285  290  295]
```

In [228]:

```
'''Say you're interested in analyzing length of delays

and you want to put these lengths into bins that represent every 10 minute period.

You can use the numpy method .arange() to create a list of numbers that define those
bins.

The bins of ten minute intervals will range from 50 minutes early (-50) to 200 minut
es late (200).

The first bin will hold a count of flights that arrived between 50 and 40 minutes ea
rly, then 40 and 30 minutes, and so on.'''
```

```
X['Dep_Diff'].hist(bins=bin_values, figsize=[14,6])
```

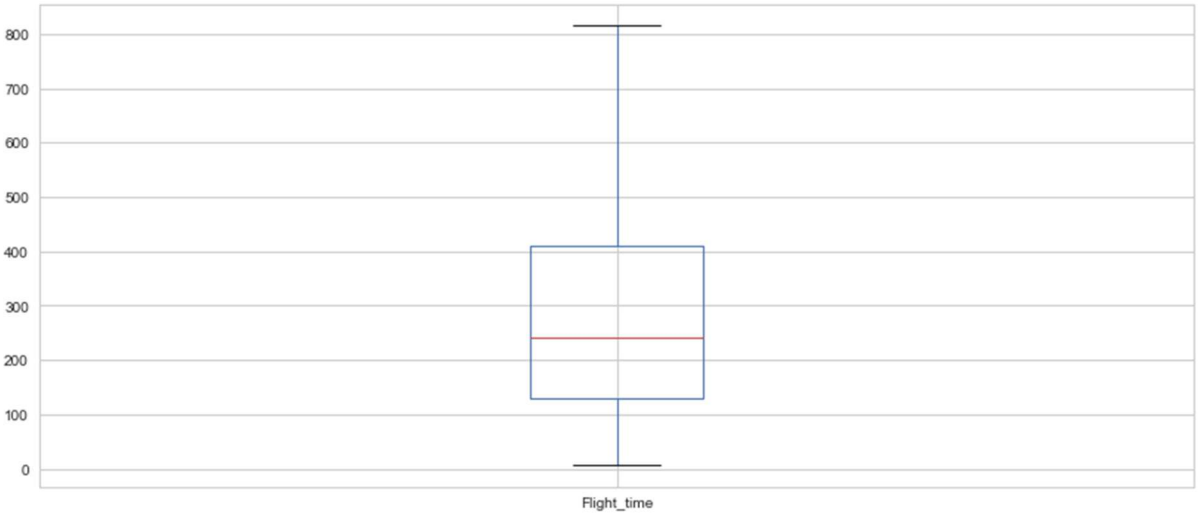
Out[228]:



In [229]:

```
X['Flight_time'].plot(kind='box', figsize=[14,6])
```

Out[229]:

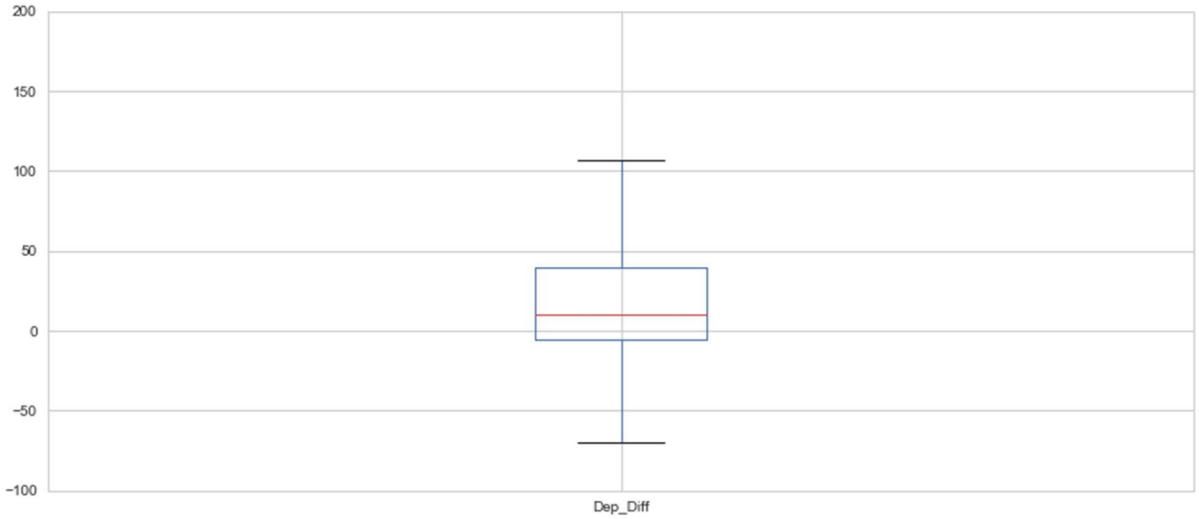


In [230]:

```
X['Dep_Diff'].plot(kind='box', figsize=[14,6])

plt.ylim(-100, 200)
```

Out[230]:

(-100, 200)

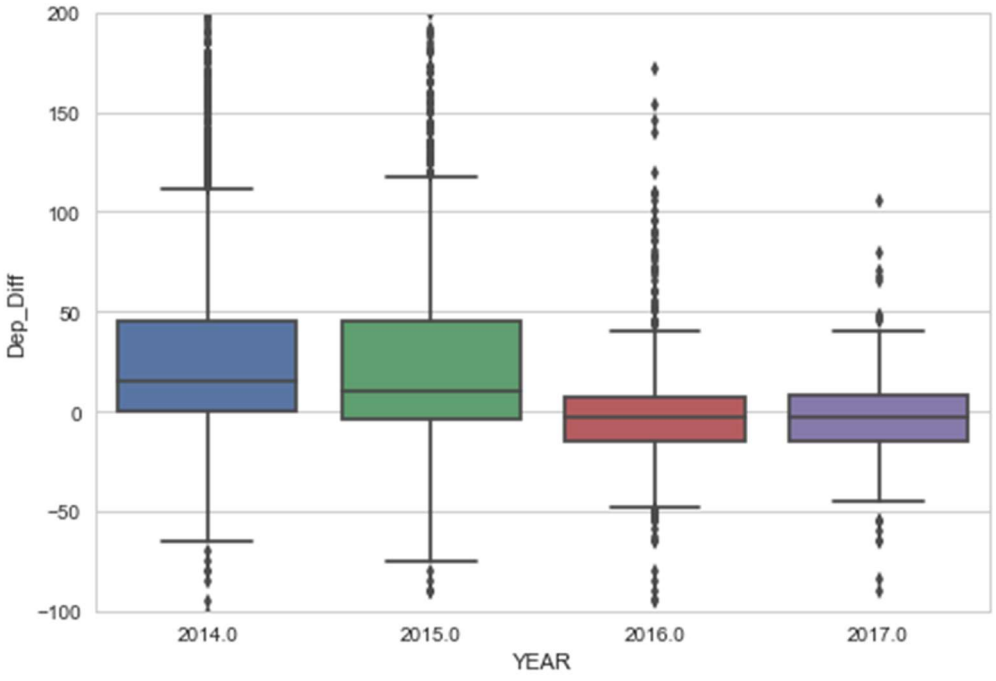


In [231]:

```
sns.boxplot(x="YEAR", y="Dep_Diff", data=X)

plt.ylim(-100, 200)
```

Out[231]:

(-100, 200)



In [1009]:

```
List(X_1)
```

```
['MONTH',
 'DAY',
 'WEEK',
 'SEASON',
 'PRIVATE_CARRIER',
 'ENG_NUMBER',
 'DEP_HOUR',
 'ARR_HOUR',
 'Flight_time',
 'Flight_more10hrs',
 'If_Dep_night_OFFICE',
 'YEAR',
 'Weekend',
 'AC_age',
 'Previous_flight_delay',
 'Previous_flight_delay_reason',
 'If_previous_flight_delayed']
```

In [319]:

```
#from sklearn.feature_selection import RFE

#only numerical variables of X were used = X_1
```

In [628]:

```
model = LogisticRegression()

rfe = RFE(model,20)
```

```
rfe = rfe.fit(X, Y15)
```

In [629]:

```
print(rfe.ranking_)
```

```
 [11 12  9  7  3  6 10  8 15  5  2 16  4 14 17 13  1]
```

In [1006]:

```
model_tree = DecisionTreeClassifier()

model_tree.fit(X, Y15)
```

Out[1006]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=None, splitter='best')
```

In [1009]:

```
print(model_tree.feature_importances_)
```

```
[ 0.07145886  0.1485054   0.05584348  0.03044306  0.01005948  0.01777502
  0.09258624  0.08376543  0.13336745  0.00352178  0.00920474  0.03078244
  0.00953454  0.04811518  0.21309227  0.03706698  0.00487764]
```

D – Supervised regression prediction

In []:

```
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder()

X_categDF_encoded = encoder.fit_transform(X)
```

In []:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=50)

X_X_PCA = pca.fit(X).transform(X)
```

In []:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=50)

X_X_conc = pca.fit(X_conc).transform(X_conc)
```

In []:

```
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder()

#categDF_encoded = encoder.fit_transform(X_1) X needs to contai
n only non-negative integers.
```

In []:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=12)

X_PCA = pca.fit(X_1).transform(X_1)
```

In []:

```
```

In []:

```
# Now we need to combine our features together to make our feat
ure matrix.

#x_final = sparse.hstack((scalingDF_sparse, categDF_encoded))
```

In []:

```
# Let’s also get our target values, which are the delay time.

y_final = dataframe['Dep_Diff']
```

In []:

```
# Finally, we need to split into test/train samples so we can s
ee how well our regressor is doing.

from sklearn.cross_validation import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X_PCA,y_fin
al,test_size = 0.2,random_state = 0) # Do 80/20 split
```

In []:

```
print(x_train)
```

```
[[  2.38503429e+01  -4.39649472e+01  -1.27451001e+01 ...,   1.0
1615937e+00

    5.08433988e-01  -3.49115769e-01]

 [  7.03003163e+01   7.39708369e+01   5.33097082e+01 ...,   1.7
2036013e-01

   -7.26654503e-01   8.05240095e-02]

 [  1.39125353e+01  -4.02055861e+01  -1.41548122e+01 ...,  -3.8
9993937e-01

   -5.04093753e-01  -4.12076906e-01]
```

```
 ...,

 [  1.24690206e+02   2.86876899e+01   4.53151849e+00 ...,  -6.9
5584187e-01

    8.94234339e-01  -2.41991257e-01]

 [  6.45221451e+01   2.02696467e+01  -3.35584873e+01 ...,  -4.2
0767783e-01

    1.42095967e+00  -5.89346104e-01]

 [  2.95467475e+01   2.16224661e+01  -3.38550276e+01 ...,  -5.6
7723282e-01

   -4.72461150e-01  -9.78060277e-02]]
```

In []:

```
# Training The Model
```

In []:

```
from sklearn.linear_model import SGDRegressor

from sklearn.grid_search import GridSearchCV

import numpy as np

SGD_params = {'alpha': 10.0**-np.arange(1,7)} # Suggested range
we try

SGD_model = GridSearchCV(SGDRegressor(random_state = 0), SGD_pa
rams, scoring = 'neg_mean_absolute_error', cv = 5) # Use 5-fold
CV

SGD_model.fit(x_train, y_train) # Fit the model
```

Out[]:

```
GridSearchCV(cv=5, error_score='raise',

       estimator=SGDRegressor(alpha=0.0001, average=False, epsi
lon=0.1, eta0=0.01,

       fit_intercept=True, l1_ratio=0.15, learning_rate='invsca
ling',
```

```
       loss='squared_loss', n_iter=5, penalty='l2', power_t=0.2
5,

       random_state=0, shuffle=True, verbose=0, warm_start=Fals
e),

       fit_params={}, iid=True, n_jobs=1,

       param_grid={'alpha': array([  1.00000e-01,   1.00000e-02
,   1.00000e-03,   1.00000e-04,

         1.00000e-05,   1.00000e-06])},

       pre_dispatch='2*n_jobs', refit=True,

       scoring='neg_mean_absolute_error', verbose=0)
```

In []:

```
from sklearn.metriHI import mean_absolute_error

y_true, y_pred = y_test, SGD_model.predict(x_test) # Predict on
our test set

'Mean absolute error of SGD regression was:'

print(mean_absolute_error(y_true, y_pred))
```

```
4.37627623845e+12
```

In []:

```
from sklearn.metriHI import r2_score

y_true, y_pred = y_test, SGD_model.predict(x_test) # Predict on
our test set

print(r2_score(y_true, y_pred))
```

```
-1.59529458442e+21
```

In []:

```
seed = 20
```

In []:

```
kfold = cross_validation.StratifiedKFold(y_final, n_folds=10, s
huffle= True, random_state= seed)
```

In []:

```
from sklearn.linear_model import SGDRegressor

from sklearn.grid_search import GridSearchCV

import numpy as np



SGD_params_2 = {'alpha': 10.0**-np.arange(1,7)} # Suggested ran
ge we try

SGD_model_2 = GridSearchCV(SGDRegressor(random_state = 0), SGD_
params_2, scoring = 'neg_mean_absolute_error', cv = kfold) # Us
e 5-fold CV

SGD_model_2.fit(X_categDF_encoded, y_final) # Fit the model
```

Out[]:

```
GridSearchCV(cv=sklearn.cross_validation.StratifiedKFold(label
s=[   -5     0 ..., -1240  -125], n_folds=10, shuffle=True, ran
dom_state=20),

       error_score='raise',

       estimator=SGDRegressor(alpha=0.0001, average=False, epsi
lon=0.1, eta0=0.01,

       fit_intercept=True, l1_ratio=0.15, learning_rate='invsca
ling',

       loss='squared_loss', n_iter=5, penalty='l2', power_t=0.2
5,

       random_state=0, shuffle=True, verbose=0, warm_start=Fals
e),

       fit_params={}, iid=True, n_jobs=1,

       param_grid={'alpha': array([  1.00000e-01,   1.00000e-02
,   1.00000e-03,   1.00000e-04,
```

```
          1.00000e-05,   1.00000e-06])},
       pre_dispatch='2*n_jobs', refit=True,
       scoring='neg_mean_absolute_error', verbose=0)
```

In [90]:

```
from sklearn.metriHI import mean_absolute_error


y_true, y_pred = y_final, SGD_model_2.predict(X_categDF_encoded
) # Predict on our test set



'Mean absolute error of SGD regression was:'

print(mean_absolute_error(y_true, y_pred))
```

```
103.72483445
```

In [91]:

```
from sklearn.metriHI import r2_score


y_true, y_pred = y_final, SGD_model_2.predict(X_categDF_encoded
) # Predict on our test set



print(r2_score(y_true, y_pred))
```

```
0.131251355032
```

In []:

```
MLPC_params_2 = {'alpha': 10.0**-np.arange(1,6)} # Suggested ra
nge we try

MLPC_model_2 = GridSearchCV(MLPRegressor(), MLPC_params_2, scor
ing = 'neg_mean_absolute_error', cv = kfold)

MLPC_model_2.fit(X_PCA, y_final) # Fit the model
```

Out[]:

```
GridSearchCV(cv=sklearn.cross_validation.StratifiedKFold(label
s=[   -5     0 ..., -1240  -125], n_folds=10, shuffle=True, ran
dom_state=20),

       error_score='raise',

       estimator=MLPRegressor(activation='relu',  alpha=0.0001,
batch_size='auto', beta_1=0.9,

       beta_2=0.999, early_stopping=False, epsilon=1e-08,

       hidden_layer_sizes=(100,), learning_rate='constant',

       learning_rate_init=0.001, max_iter=200, momentum=0.9,

       nesterovs_momentum=True, power_t=0.5, random_state=None,

       shuffle=True, solver='adam', tol=0.0001, validation_frac
tion=0.1,

       verbose=False, warm_start=False),

       fit_params={}, iid=True, n_jobs=1,

       param_grid={'alpha': array([  1.00000e-01,   1.00000e-02
,   1.00000e-03,   1.00000e-04,

         1.00000e-05])},

       pre_dispatch='2*n_jobs', refit=True,

       scoring='neg_mean_absolute_error', verbose=0)
```

In []:

```
from sklearn.metriHI import mean_absolute_error

y_true, y_pred = y_final, MLPC_model_2.predict(X_PCA) # Predict
on our test set

'Mean absolute error of SGD regression was:'

print(mean_absolute_error(y_true, y_pred))
```

```
98.709094708
```

In []:

```
from sklearn.metriHI import r2_score

y_true, y_pred = y_final, MLPC_model_2.predict(X_PCA) # Predict
on our test set

print(r2_score(y_true, y_pred))
```

0.29436397037

In []:

```
from sklearn.metriHI import mean_squared_error



y_true, y_pred = y_final, MLPC_model_2.predict(X_PCA) # Predict
on our test set



'Mean absolute error of SGD regression was:'

print(mean_squared_error(y_true, y_pred))
```

40614.3373643

In []:

```
 seed3 = 2
```

In []:

```
kfold3 = cross_validation.StratifiedKFold(y_final, n_folds=3, s
huffle= True, random_state= seed3)
```

In []:

```
LogisticRegression_params_3 = {'C': [0.1, 1]} # Suggested range
we try

LogisticRegression_model_3 = GridSearchCV(LogisticRegression(),
LogisticRegression_params_3, scoring = 'neg_mean_absolute_error
', cv = kfold3)
```

```
LogisticRegression_model_3.fit(X_PCA, y_final) # Fit the model
```

Out[]:

```
GridSearchCV(cv=sklearn.cross_validation.StratifiedKFold(label
s=[ -5.   0. ...,  35. -17.], n_folds=3, shuffle=True, random_s
tate=2),

       error_score='raise',

       estimator=LogisticRegression(C=1.0, class_weight=None, d
ual=False, fit_intercept=True,

          intercept_scaling=1, max_iter=100, multi_class='ovr',
n_jobs=1,

          penalty='l2', random_state=None, solver='liblinear',
tol=0.0001,

          verbose=0, warm_start=False),

       fit_params={}, iid=True, n_jobs=1, param_grid={'C': [0.1
, 1]},

       pre_dispatch='2*n_jobs', refit=True,

       scoring='neg_mean_absolute_error', verbose=0)
```

In []:

```
from sklearn.metriHI import mean_absolute_error

y_true, y_pred = y_final, LogisticRegression_model_3.predict(X_
PCA) # Predict on our test set

'Mean absolute error of SGD regression was:'

print(mean_absolute_error(y_true, y_pred))
```

```
73.5943245982
```

In []:

```
from sklearn.metriHI import r2_score

y_true, y_pred = y_final, LogisticRegression_model_3.predict(X_
PCA) # Predict on our test set
```

```
print(r2_score(y_true, y_pred))
```

```
-0.0010525370733
```

In []:

```
from sklearn.metriHI import mean_squared_error


y_true, y_pred = y_final, LogisticRegression_model_3.predict(X_
PCA) # Predict on our test set


'Mean absolute error of SGD regression was:'

print(mean_squared_error(y_true, y_pred))
```

```
52949.1565079
```

In []:

```
from sklearn.pipeline import Pipeline

from sklearn.pipeline import FeatureUnion

from sklearn.decomposition import PCA

from sklearn.feature_selection import SelectKBest
```

In []:

```
# create feature union

features = []

features.append(('pca', PCA(n_components=12)))


feature_union = FeatureUnion(features)
```

In []:

```
estimators = []

estimators.append(('feature_union', feature_union))

estimators.append(('logistic', LogisticRegression()))

model = Pipeline(estimators)
```

In []:

```
results = cross_validation.cross_val_score(model, X_1, y_final,
cv=5, scoring='neg_mean_absolute_error')
```

In []:

```
print(results.mean())
```

```
-40.6375510554
```

```
seed = 2
```

In []:

```
kfold = cross_validation.StratifiedKFold(y_final, n_folds=2, sh
uffle= True, random_state= seed)
```

In []:

```
predicted = cross_validation.cross_val_predict(LogisticRegressi
on(fit_intercept = False, C = 1e9), X_X_PCA, y_final, cv=kfold)
```

In []:

```
from sklearn.model_selection import cross_val_score

print(cross_val_score(LogisticRegression(fit_intercept = False,
C = 1e9), X_X_PCA, y_final, cv=kfold))
```

```
[ 0.0950495   0.09867452]
```

In []:

```
from sklearn.metriHI import accuracy_score

print (accuracy_score(y_final, predicted))
```

0.0968253968254

In []:

```
from sklearn import linear_model

regr = linear_model.LinearRegression()
```

In []:

```
regr.fit(x_train, y_train)
```

Out[]:

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, nor
malize=False)

In []:

```
# The coefficients

print('Coefficients: \n', regr.coef_)
```

Coefficients:

[-0.46440218 -0.5569041   0.41619785 -1.12888167 -1.64731
246

  -0.33191602   2.25288273 -13.49839745 10.64817207 15.933860
39

   7.22574878   7.89312132]

In []:

```
print("Mean squared error: %.2f"
```

```
      % np.mean((regr.predict(x_test) - y_test) ** 2))
```

```
Mean squared error: 6995.98
```

In []:

```
# Explained variance score: 1 is perfect prediction

print('Variance score: %.2f' % regr.score(x_test, y_test))
```

```
Variance score: 0.03
```

In []:

```
from sklearn.metriHI import explained_variance_score

y_true, y_pred = y_test, regr.predict(x_test) # Evaluate test s
et

explained_variance_score(y_true, y_pred)
```

Out[]:

```
0.031186035887383778
```

In []:

```
from sklearn.metriHI import r2_score

y_true, y_pred = y_test, regr.predict(x_test) # Evaluate test s
et

print(r2_score(y_true, y_pred))
```

```
0.0311169663911
```

In []:

```
def delay_prediction(MONTH=1,

 DAY=1,

 WEEK=6,

 SEASON=2,
```

```
 PRIVATE_CARRIER=0,

 ENG_NUMBER=4,

 DEP_HOUR=15,

 ARR_HOUR=18,

 Flight_more10hrs=0,

 If_Dep_night_OFFICE=0,

 YEAR=2015,

 Weekend=0):

    categorical_values = np.zeros(12)

    categorical_values[0] = int(MONTH)

    categorical_values[1] = int(DAY)

    categorical_values[2] = int(WEEK)

    categorical_values[3] = int(SEASON)

    categorical_values[4] = int(PRIVATE_CARRIER)

    categorical_values[5] = int(ENG_NUMBER)

    categorical_values[6] = int(DEP_HOUR)

    categorical_values[7] = int(ARR_HOUR)

    categorical_values[8] = int(Flight_more10hrs)

    categorical_values[9] = int(If_Dep_night_OFFICE)

    categorical_values[10] = int(YEAR)

    categorical_values[11] = int(Weekend)


    categorical_values_encoded = encoder.transform([categorical
_values]).toarray()  #works

    final_test_example = np.c_[categorical_values_encoded]
```

```
    #pca = PCA(n_components=12)

    #X1_PCA = pca.fit(categorical_values).transform(categorical
_values)

    #np.asarray(X1_PCA)

    #X_scenario = pd.categorical_values_encoded

    #X_PCA = pca.fit(X_1).transform(X_1)

    #Combine these into the final test example that goes into t
he model

    # Now predict this with the model

    pred_delay = SGD_model_2.predict(final_test_example)

    print ('Your predicted delay is', int(pred_delay[0]), 'minu
tes.')

    return # End of function
```

In []:

```
delay_prediction(MONTH=1,

 DAY=1,

 WEEK=6,

 SEASON=2,

 PRIVATE_CARRIER=0,

 ENG_NUMBER=4,

 DEP_HOUR=15,

 ARR_HOUR=18,

 Flight_more10hrs=0,

 If_Dep_night_OFFICE=0,

 YEAR=2015,

 Weekend=0)
```

```
Your predicted delay is 40 minutes.
```

In []:

```
X.to_HIv('X.HIv', index=False, header=True)
```

In []:

```
def delay_prediction(pred = pd.read_HIv('X_pred.HIv')):

    pca = PCA(n_components=50)

    X_pred = pca.fit(pred).transform(pred)

    # Combine these into the final test example that goes into
the model

    # Now predict this with the model

    pred_delay = MLPC_model_2.predict(X_pred)

    print ('Your predicted delay is', int(pred_delay[0]), 'minu
tes.')

    return # End of function
```

In []:

```
delay_prediction(pred = pd.read_HIv('X.HIv'))
```

```
Your predicted delay is 7 minutes.
```