Final Project
Introduction to Data Science
Data Analysis and Classification
US airports delays and cancellation

Ivan Soto

2016-11-26

Contents

1 Overview

1.1 Overview

This research is aimed to identify and select a question that arises in the US Delays and Cancellations domain. The purpose of this document is to present the problem statement, the question derived from it and the approaches that are going to be used to elicit an answer to this question. A descriptive analysis and exploratory analysis will be provided as well.

1.2 Problem Statement

An aviation startup company (Avinalytics) is about to launch a new project, in which they want to understand and characterize the delays and cancellations to create an app capable to determine the likelihood of a flight in the US being cancelled.

First, they want to analyze historical flight data and identify patterns and characteristics for the different flight delays and cancellations, based on the date, the airline operators, the airports, route structures, region or state of the country, etc. The Avinalytics company hires you to perform this task.

1.3 Motivation

It comes as no surprise that there are over 3000 commercial flights every day. As expensive as traveling through air can get, it is expected that this service will be flawless. A great number of people traveling through air anticipate the date they are planning to flight, most likely expecting their travelling experience will be smooth. Unfortunately, it can be seen that is not always the case: flights can be delayed or cancelled for several reasons, that end up disrupting the plans for a traveler.

In order to aid people who often travel through air, the following question has to be answered: Given a flight, **is it likely to be cancelled?**

1.4 Approach

In order to answer this question, historical flight data has been downloaded from the Office of the Assistant Secretary for Research and Technology: Bureau of Transportation Statistics. Data delivered in this way is said to be secondary data, that for our research suffices to get close to answering our question.

For every observation in the data set, the outcome of the flight is stated. This allows us to consider this problem as a supervised learning task, studied under the field of Machine Learning. To be more precise, there are two possible outcomes for each flight: cancelled or not cancelled. From now on, the terms "outcome" and "class" will be used interchangeably, the same goes for the terms "observation" and "training example". Since there are classes to which a training example can belong, this now becomes a classification problem. An

algorithm that implements classification is known as a classifier. A classifier will allow us to tell, for any given example, the class to which it most likely belongs.

With the purpose of selecting the classifier that best formulates a prediction function $f(x)$, the data has to be analyzed through descriptive and exploratory analysis to find patterns: the accuracy of a classifier is subject to the model the data most closely resembles. Once we suspect of a possible model, we train several classifiers by feeding them a training set in order to set them ready to make predictions for any example. Tuning a classifier is important to achieve better accuracy as well as trying several training sets to ensure that the model is not overfitted to the data.

2 Descriptive analysis

2.1 Overview

Before anything else, basic features of the data in this study will be described, in order to provide simple summaries about the sample taken and get a better understanding of it. The data collected from the Bureau of Transportation Statistics contained eleven datasets, each of which contained at least 400,000 observations. Around 40% of the dataset was sampled randomly for further analysis so as to quickly process it to present descriptive and exploratory statistics. For a classifier to work with higher accuracy, the subset of features that best represent the data aligned to the question to be answered has to be chosen. There are algorithms in the field of Machine Learning that reduce the dimensionality of the number of features, and that can result in overall improved accuracy.

2.2 Feature selection

Out of the 110 features available for each observation in the data set, a subset of those were selected as representative features through manual selection. The following gives a brief description of each of the features selected:

- Time Period
  - Quarter
    Period of the year which can be related to seasonal changes.
  - Month
    Period in time that can be related to vacations, trends or important events.
- Airline
  - UniqueCarrier
    Identifier of the brand or airline that owns the flight can help identifying trends about the management of each airline.
  - FlightNum
    Unique identifier from each flight which is ultimately the unique identifier for each observation.
- Origin
  - OriginAirportID
    Identifier from which the flight departed, can help to identify problems related to departure of specific locations or cities.
- Destination

  - DestAirportID
    Id of the airport the flight arrived to, it can be related to air congestion in given location or arrivals management.

- Departure Performance
  - CRSDepTime
    Computer Reservations System Departure Time refers to the time the flight is scheduled it may help to find problems related to specific periods of time in the day.
  - DepDelay
    The difference between the scheduled departure time and the actual departure time of the flight it may help see a relation between time loss and flight cancelations.

- Arrival Performance
  - CRSArrTime
    The computer calculated arrival time.
  - ArrDelay
    Difference between the computer estimate and the actual time of arrival.

- Cancellations and Diversions
  - Cancelled
    The actual class the observation belongs to it can acquire the values of cancelled or not cancelled. This will be the value produced by the function $f(x)$ given an input flight and its conditions.
  - CancellationCode
    Although this might not help to actually classify the data it can be used during the exploratory analysis to figure out the patterns which produce this type of cancelation.

- Flight Summaries
  - Distance
    Physical separation between the origin airport and the destination. It can help to guess problems related to air time, gas or mechanical issues.

2.3 Analyzing the features

For some of the features previously presented, statistics will be presented that may show interesting facts about the data, and which can further aid in the process to get a bit closer to answering our question.

Starting from the feature named "Quarter", the most we can get out of it alone is the quarter in which most of the planes took plane. Figure 1 shows the

number of flights registered in our sample for each of the quarter, and it can be seen that quarter 3 had more flights slightly over quarter 2. It should be noted that flights from March were missing in the dataset, so it makes sense for quarter 1 to have a smaller frequency.

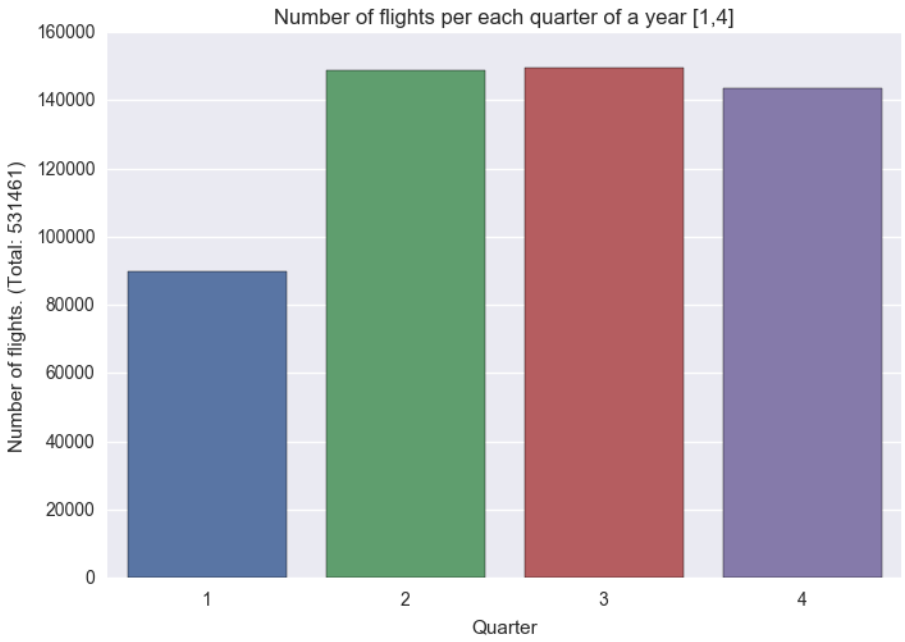

Figure 1: Flights per year quarter

It may prove interesting to show which flight numbers have been on the air the greater number of times. Figure 2 plots the occurrences of each flight number. It can be important to show at some point if this holds true: the more frequent a flight the more prone it is to delays, and the same analysis for cancellations, which is what the question seeks to answer. The mode ended up being flight number 469 with 376 appearances in the dataset.

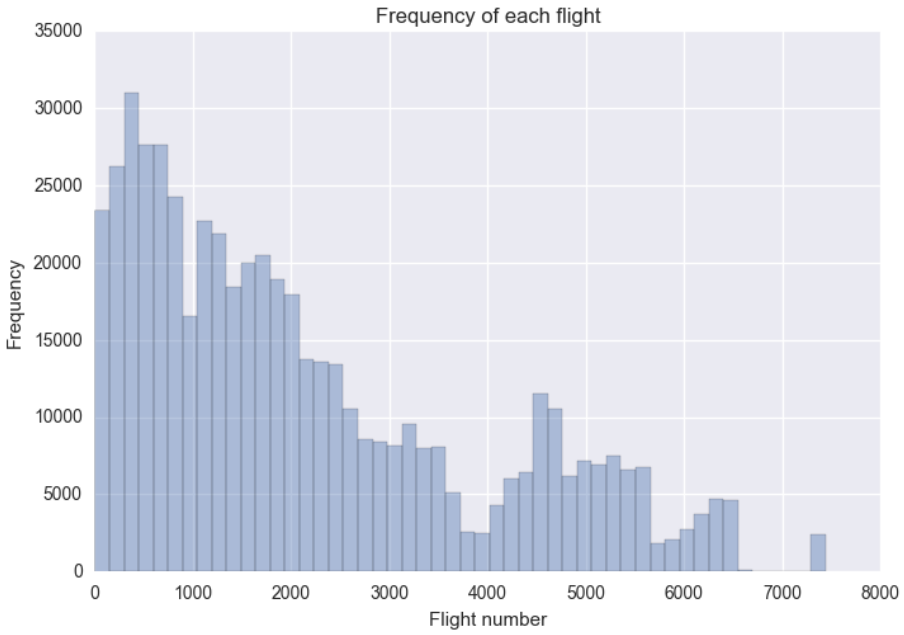

Figure 2: Frequency of flights

The same issue happens for the rest of the categorical variables: we can at most get the mode out of them at this point.

How about non-categorial variables, such as departure delay? Since this is a continuous random variable, statistics such as the mean and the standard deviation can be obtained. A value of 9.3005 was found to be the mean for this feature, which means that on average, flights are delayed by slightly more than 9 minutes. It might be interesting to show, further in the study, how delay time affects the outcome of a flight. As for the standard deviation, 36.807 was calculated, reflecting the extent to which data is spread out with respect to delay time. If this were to be plotted, observations could be made about the continuous probability distribution that this data shows, as illustrated in Figure 3. As seen, the curve is clearly skewed to the right, and that could mean a plane rarely gets delayed for a long period of time.

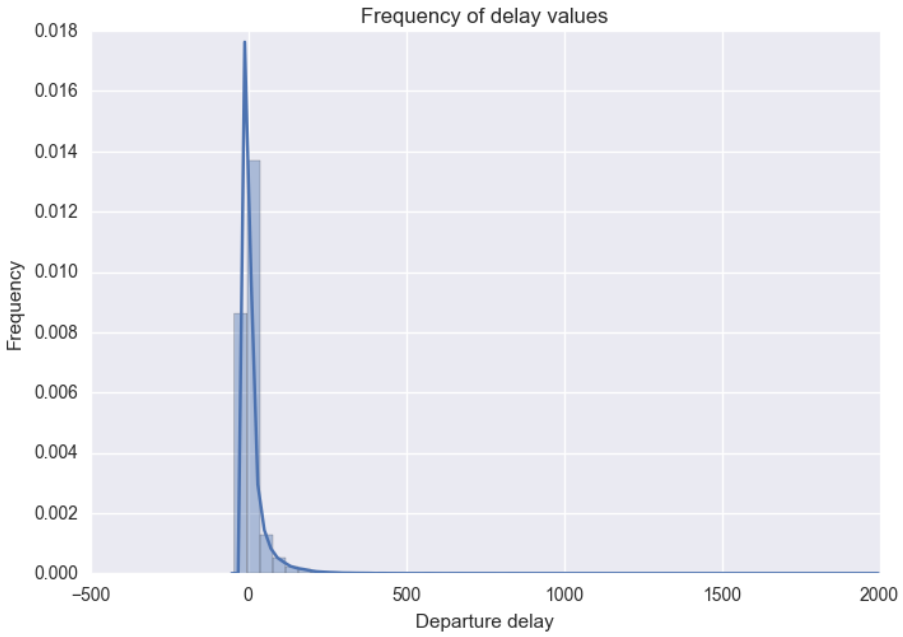

Figure 3: Delay on flights

Distance is also a continuous random variable, and a plot can show the probability distribution it fits. Figure 4 shows this plot and no known distribution can be identified, except it looks skewed to the right. It can only be said that most flights do not travel really long distances.

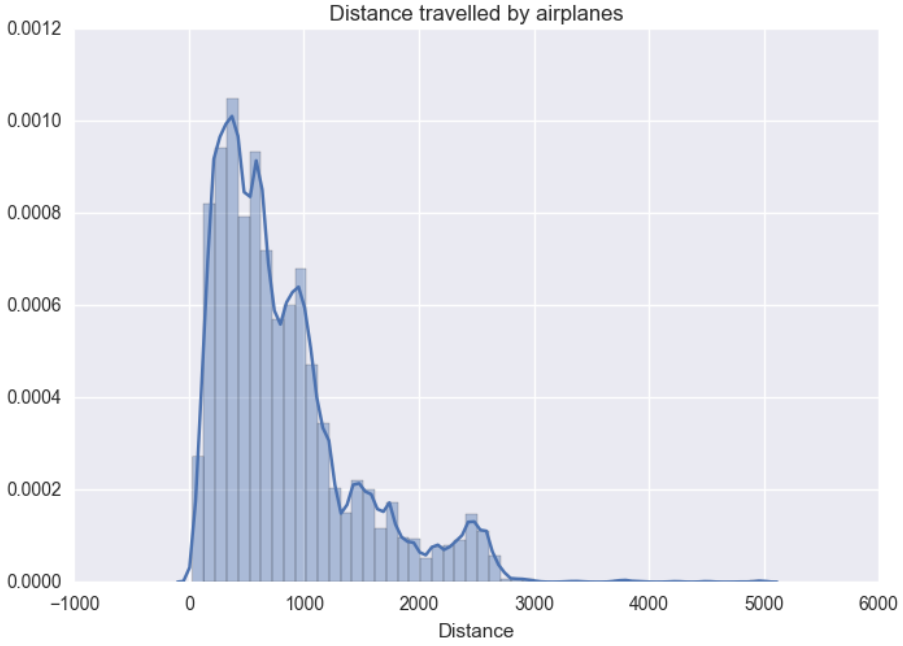

Figure 4: Distance of flights

Let's also see, for some pairs of Origin and Destination Airport, the mean distance that was traveled between them (every flight's distance between a pair of airports is recorded, and then the average is calculated). Figure 5 represents the mean distances using a heatmap.

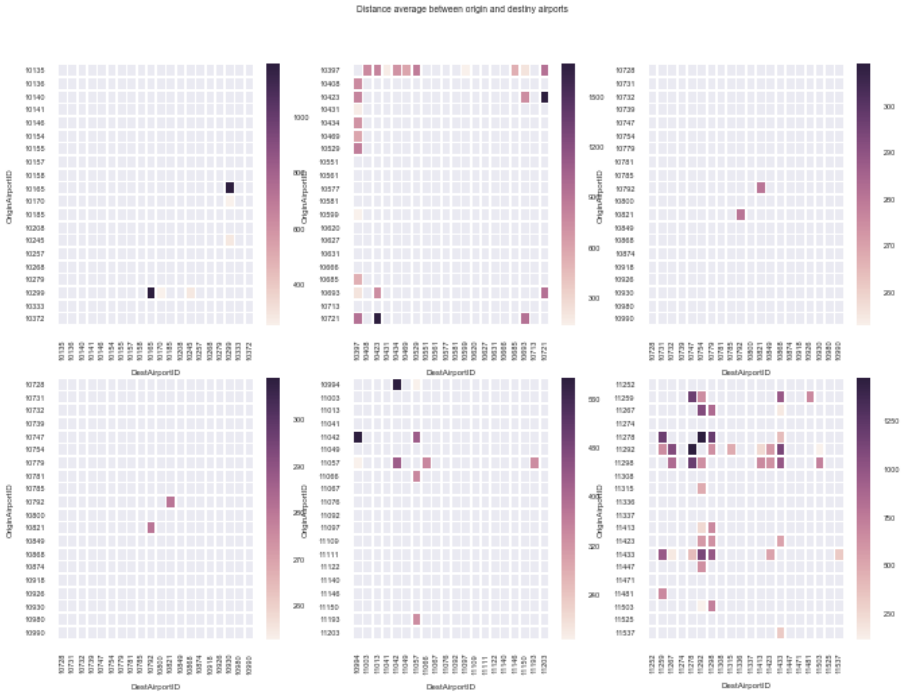

Figure 5: Mean distance between airports

Last but not least, statistics on the "Cancelled" feature. Figure 6 compares the number of cancelled flights to the number of not-cancelled flights. We can clearly see that the number of cancelled flights is really low! Based on this, we can tell with no precise numerical value that the probability of a flight being cancelled is low.

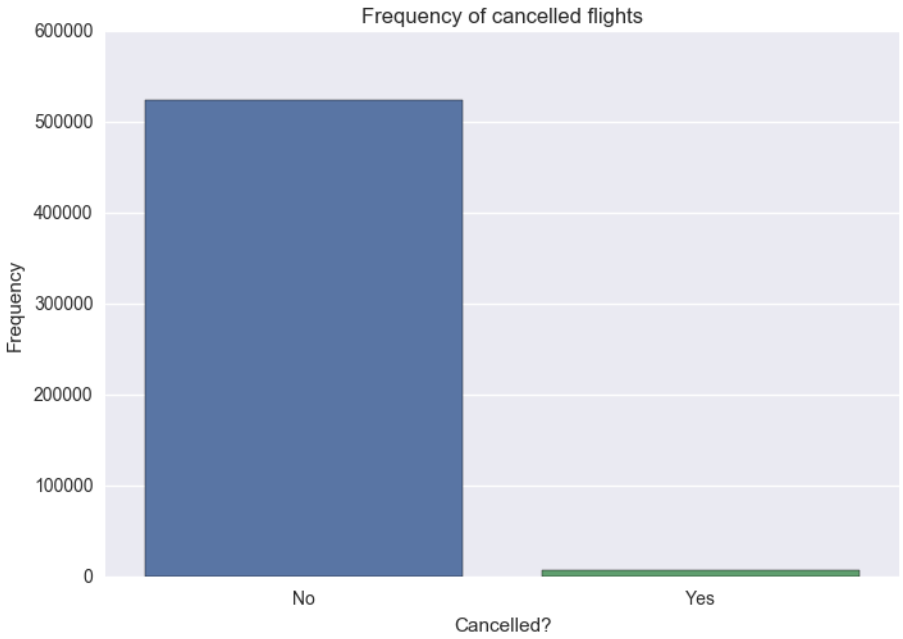

Figure 6: Cancelled flights

3 Exploratory Analysis

3.1 Overview

After a descriptive analysis of the data, we can start finding possible correlations between pairs of features in the dataset. It may first prove interesting to get an image that can tell us quantitatively the extent to which a dependency relationship exists, and a different one that can visualize each relationship.

3.2 Plots

The graphs that accomplish illustrating the degree of a correlation between every pair of features and its visual interpretation that complements that degree (represented as a real number) are the coefficient matrix and the scatterplot matrix respectively. Figure 7 shows a coefficient matrix represented through a heatmap and Figure 8 builds a scatterplot for every pair of features but the *ArrDelay* and *DepDelay* features. The reason is that, for cancelled flights, these numbers ought to be non-existant.

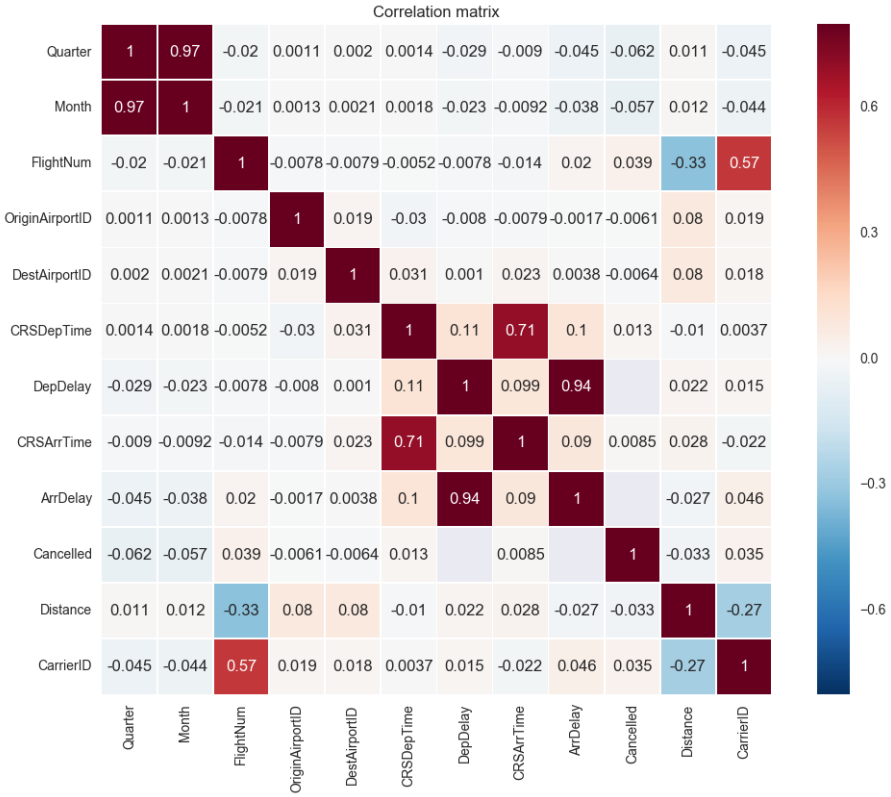

Figure 7: Heatmap on correlation matrix

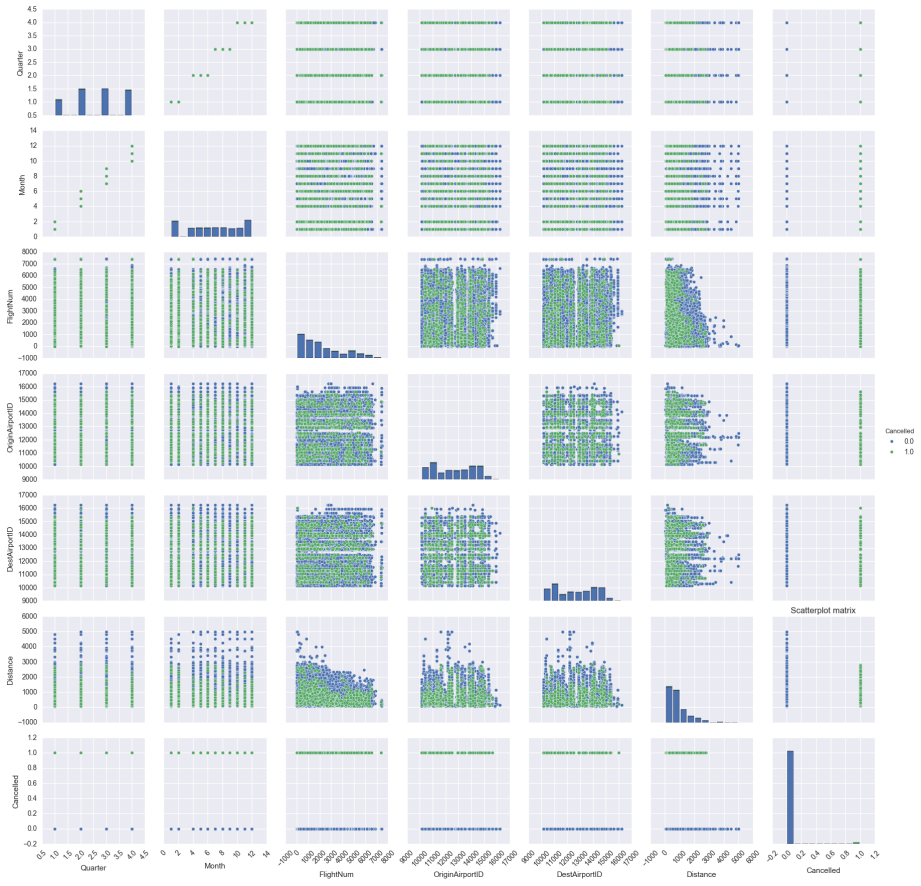

Figure 8: Scatterplot showing each variable against the others

Based on the values obtained from the correlation matrix, we could dig into plotting pair of features from which interesting things may be obtained that the scatterplot has not shown. It might also be the case that plots can be obtained that were not available on the scatterplot. For instance, Figure 9 shows the ratio of cancellation for each Air Carrier, and Figure 10 shows the number of cancelled flights per carrier: if these plots are analyzed carefully, we can see that the ratio of cancellation does not necessary mean that many flights were cancelled by an Air Carrier, relatively speaking. Some carriers with low ratio of cancellation had tons of cancelled flights, but then that means that the total number of flights was really high.

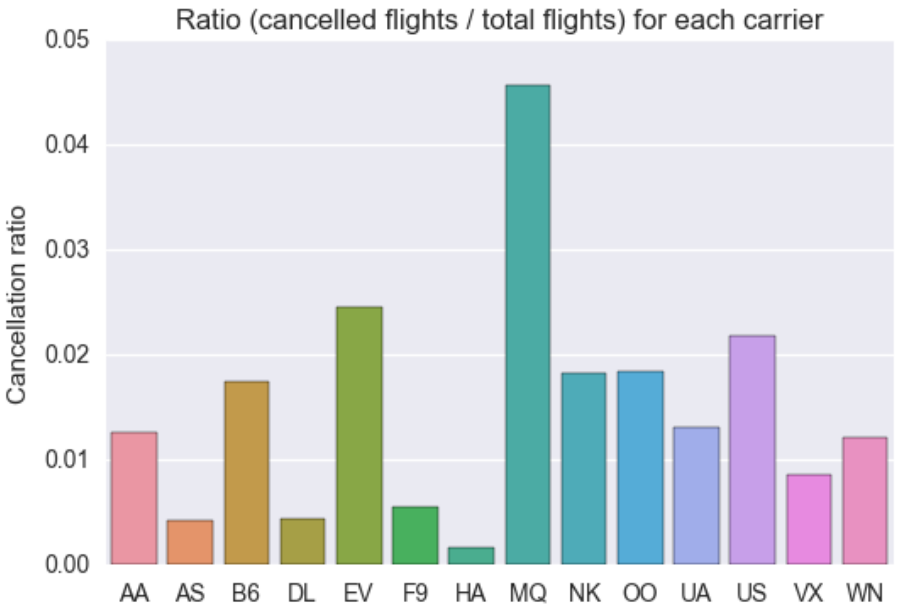

Figure 9: Ratio of cancelled flights on total flights based for each carrier based on historical data

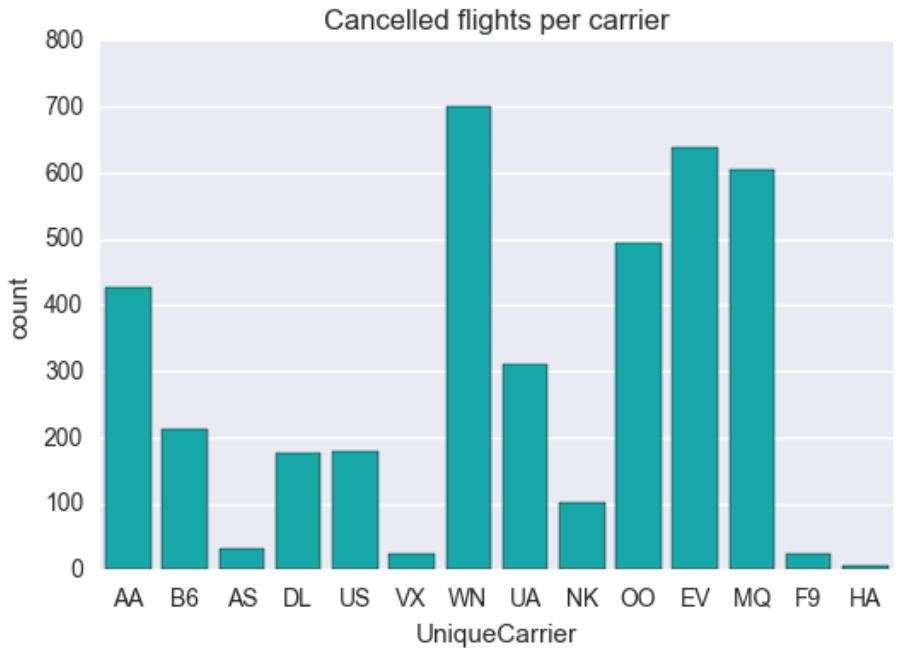

Figure 10: Number of cancelled flights per carrier

Let's recall that the scatterplot did not include *ArrDelay* and *DepDelay*, so these two cannot be left uncompared. It is expected that, if a flight gets delayed by a certain amount of time t, it has to arrive to its destination with an approximate t value. The regression in Figure 11 shows that this holds true.

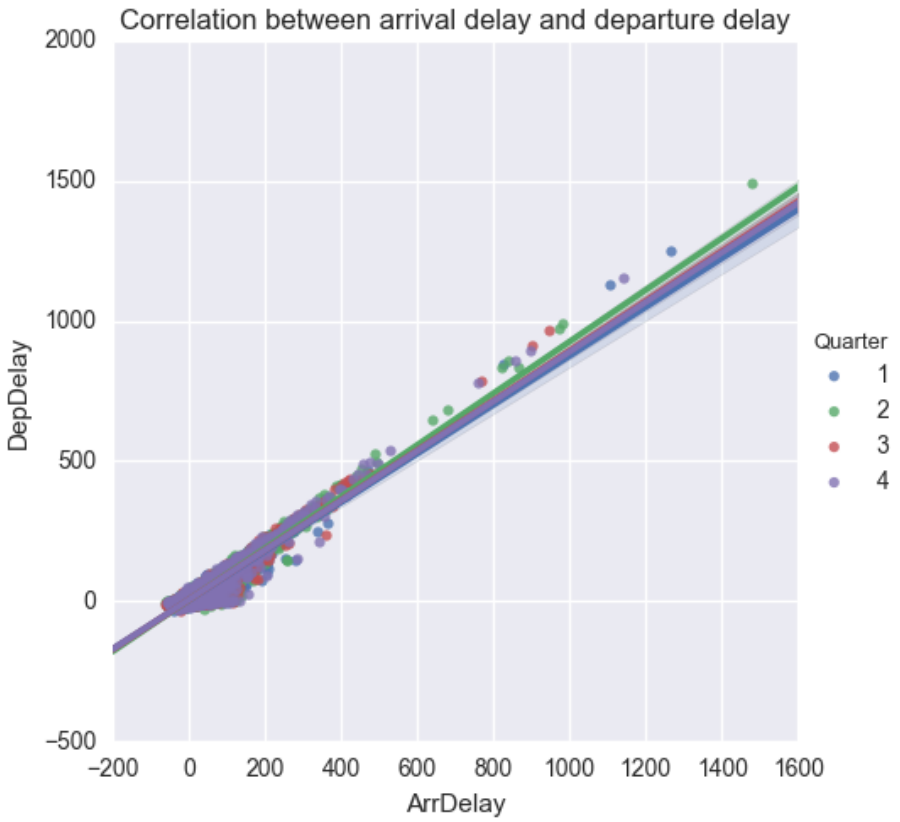

Figure 11: Regression between arrival delay and departure delay

Last, it might interesting to see the number of flights that have been cancelled in each month and the distance they have traveled, to see if these are deterministic factors on a flight's outcome. Figure 12 illustrates, as we know, the number of cancelled flights is low compared to the non-cancelled flights, and January and February seem to be more frequent on cancelled flights compared to the rest of the months. Cancelled flights also do not surpass certain threshold value that may denote really long flights. Based on this, we could even say that long flights are not prone to cancellation.

Based on what we have seen, delays cannot be used to determine the outcome of a flight (we do not know them for cancelled flights in the first place). Carrier, flight number, airports, time and distance give us relevant information to suspect on what the outcome of a flight is going to be. These features can be then given to a classifier that may give us, with certain assurance, what the result of a flight most likely is, as discussed in the following section.

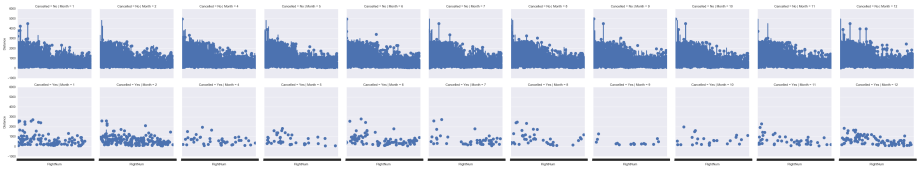

Figure 12: Distances for cancelled and not cancelled flights

4 Classification

4.1 Overview

As mentioned at the beginning of this paper, the question to be answered is reduced to a classification problem. We can conclude several things after reviewing the descriptive and exploratory analysis: one of the key results that were obtained is that the proportion between number of cancelled flights and total numbers is quite low, so we are expecting almost every flight not to be cancelled.

Several classifiers were trained using 85% of the 40% sample obtained out of the whole dataset. The reason is that, a classifier should never be trained using the complete dataset: it's going to fail at predicting unseen data (i.e. it's going to be overfitted). The sample was therefore divided into a training set and a test set. The training set is used to find the optimal parameters for the classifier that will allow it to predict classes on new data. The purpose of the test set is to validate that the model is not overfitted but at the time it validates whether the model can predict accurately.

4.2 Searching for the best hyperparameters

Hyperparameters are not directly learnt from the classifier. Instead, they are set and tuning them according to the problem at hand can improve the results drastically. Depending on the classifier, the hyperparameters that are available for tuning may vary, and there are several approaches to finding the best hyperparameters. The most used algorithms are known as Grid Search and Randomized Search.

Given a set of options for each of the hyperparameters a classifier is allowed to receive, Grid Search performs an exhaustive search in which it tries every possible combination of hyperparameters, calculates the accuracy of the fitted model, and takes the set of hyperparameters for which its accuracy was maximum. Randomized Search, on the other hand, does not try very possibility. Instead, given a number of iterations, it randomly picks a value for each hyperparameter that can be set for the classifier and then calculates the accuracy of the model using that set of hyperparameters. This is done for n iterations,

and at the end it takes the set of hyperparameters for which the accuracy of the classifier is maximum. Randomized Search was picked given its better time performance over Grid Search.

4.3 Cross-validation

A cross-validated search for the best parameters is performed on both Grid Search and Randomized Search, thus avoiding the problem of overfitting. Cross-validation (also known as k-fold cross-validation) is a technique that divides the dataset into k pieces. $k-1$ pieces are used to train the classifier, and the remaining kth piece is used to test the model. Its accuracy is obtained, and this procedure is performed a total of k times. The list of accuracy values are then averaged and that represents the overall accuracy of the model.

4.4 Trained classifiers

4.4.1 K-nearest neighbors

An observation for which we want to predict a class can be vectorized in $\mathbb{R}^n$ space, where n denotes the number of relevant features the observation needs to be fed to the model. KNN (K-nearest neighbors) obtains the k closest labeled observations to be unseen observation to which we wish to predict in $\mathbb{R}^n$ space using any distance metric (Manhattan distance, Euclidean distance, Minkowski distance). Every observation class is queried, and the most seen class is then the predicted class for our unseen observation. After fitting the hyperparameter-optimized KNN and testing it with the remaining 15% test data, the accuracy score obtained was 0.984973, while its f1 score was 0.0.

4.4.2 Logistic regression

A vector $\vec{v} = \begin{bmatrix}\theta_0, & \theta_1, & ..., & \theta_n\end{bmatrix}^\intercal$ is obtained with values for the coefficients that denote a line in $\mathbb{R}^n$ space that minimizes the least squares error with respect to the training set. Using this model, for each trial the probability that it belongs to a certain class is calculated using a logistic function (represented as $f(x) = \frac{1}{1+e^{-x}}$). Randomized Search tried L1 and L2 regularization to find the best Logistic Regression model. Regularization consists of adding a term to a cost function (such as the least squares error function) so that high coefficients are penalized, thus resulting in a model that does not perfectly fit the training set, therefore, solving the overfitting problem. In the following equation, V can be any cost function, and $R(f)$ penalizes based on the complexity of f. $R(f)$ for L1 regularization is simply the sum of the coefficients, while L2 regularization is the sum of the square of the coefficients. Minimization of the whole function is desired to build the best regression model:

$$min_f \sum_{i=1}^{n} V(f(\hat{x}_i), \hat{y}_i) + \lambda R(f)$$

The accuracy score obtained was 0.984 while the f1 score is inclined to 0.0.

4.4.3 Stochastic Gradient Descent

Gradient Descent is an optimization algorithm to find the local minima in a function $f(x)$. In the context of supervised learning, Gradient Descent is used to find the optimal coefficients of a vector $\vec{v}$ that minimizes a cost/loss function. Given $\vec{w}$ initialized to random values, SGD (stochastic gradient descent) updates $\vec{w}$ after obtaining the gradient (a vector that points in the direction to which a function increases) and substracting it to the current $\vec{w}$ vector (that is, in the opposite direction to find the local minima). After n iterations, $\vec{w}$ has a good approximation for the coefficient values. The loss function used to train the SGD classifier was hinge loss, which makes Support Vector Machines allowable. A SVM (Support Vector Machine) builds a hyperplane $\vec{w}$ between the points so that it becomes a decision boundary that divides points into two classes. A trial is classified based on the side of the decision boundary it is in. The accuracy the SGD SVM classifier obtained was 0.9848 along with a 0.000825 f1 score.

4.4.4 Naive Bayes

The idea behind the series of algorithms under "Naive Bayes methods" is to apply Bayes' theorem with a naive assumption that every pair of features are independent. Bayes' theorem is represented by the equation

$$P(y \mid X) = \frac{P(X \mid y)\, P(y)}{P(X)}$$

That is, y is the label to which a trial may belong, and X is the trial's vector representation. This can be interpreted as "the probability to belong to class y given the set of features X". MAP (Maximum A Posteriori) estimator can be used to obtain the class to which the trial most likely belongs to. Expressed mathematically, for every y class, we want to obtain y so that: $y_{MAP} = max_y P(y \mid X)$. There are different Naive Bayes classifiers, all of which differ on the assumptions made on the distribution P. Training was done on the Multinomial and Bernoulli NB (Naive Bayes). Multinomial NB builds a distribution for the random variable represented as a vector $\vec{\theta} = \begin{bmatrix} X_0 & X_1 & ... & X_n \end{bmatrix}$. On the other hand, Bernoulli NB binarizes the feature vector, since a Bernoulli random variable X can only take values 1 and 0 for success and failure respectively. The accuracy score for Multinomial NB was 0.6405 along with a f1 score of 0.044. Bernoulli NB got 0.984 and 0.0 for accuracy and f1 score respectively.

4.4.5 Random Forest

This is known as an ensemble. An ensemble is basically a combination of estimators, for which a reduction function R takes place to output a final value. Random Forest consists of several decision trees, each independent to every other, and every tree is learns from a random sample drawn with replacement, and at the end, the mode of the results (for the classifier version) is taken as the predicted class. For the hyperparameters Randomized Search, criterion used to

split each decision tree were gini and entropy, several values for the max depth of the tree were set and the number of trees the Random Forest contained was fixed to 30. This resulted in 0.984 accuracy and 0.0 f1 score.

4.4.6 Voting classifier

Last but not least, another ensemble was built. Not out of decision trees this time, but out of the Logistic Regression, Random Forest and Bernoulli Naive Bayes classifiers. This is also known as a voting classifier: each classifier predicts a class for the trial, and soft voting took place to make a decision on the final class. Soft voting consists of summing the predicted probabilities from each classifier for each class, and then outputting the class that maximizes this sum. The voting classifier got similar results as well: accuracy was 0.984 while f1 score was 0.0.

4.5 Plots

The following (figures 13 and 14) summarizes the results that were obtained for each of the classifiers that were trained. Accuracy is m/n where m is the number of correct predictions and n is the total number of predictions. The F1 score is calculated using precision and recall. Precision is $T/(T+t)$ and recall is $T/(T+f)$, where T is number of true positives, t is number of false positives and f is number of false negatives.

Figure 13: Accuracy scores for the trained classifiers

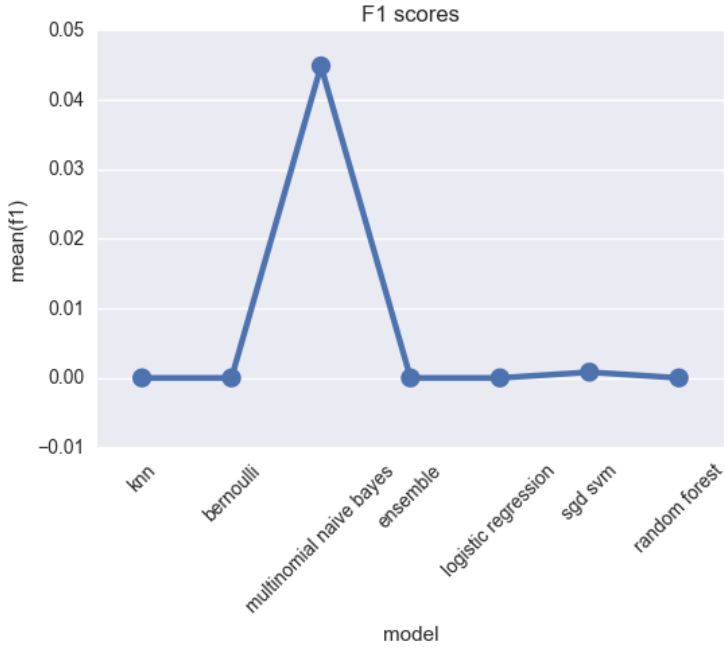

Figure 14: F1 scores for the trained classifiers

5 Conclusion

It is suspected that, since this is a binary classification problem, and the fact that the results from the exploratory analysis hardly showed any important correlations between features, besides the fact that a small number of flights are cancelled, can lead us to believe that the most important feature to take into consideration is the historical data on each unique carrier to predict the outcome of a flight. Every classifier, suspiciously, had almost the same results, and being a binary classification problem may be the reason for classifiers to handle this task easily. It might be interesting to be how they might behave if we transform this into a multiclass problem: not just predicting if it is cancelled or not, but also predict the cancellation code. This might produce most likely different accuracies for each of the classifiers. Also, trying a different dataset for validation might be useful, since the number of cancelled observations in the 2015 dataset is really low, so there is not enough information for the classifier to make a better decision.

6 Appendix

The code for the descriptive, exploratory analysis and classification task can be found in this repository, along with the serialized models: https://github.com/IvanAli/DataScienceITESM

7 Referencias

On-time flights (2016). Document retrieved from: http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time

Flight delay cause (2016). Document retrieved from: http://www.transtats.bts.gov/ot_delay/ot_delaycause1.asp?type=21&pn=1

Flight Stats Global Cancellation and delays (2016). Document retrieved from: http://www.flightstats.com/go/Media/stats.do

Supervised learning – scikit-learn 0.18.1 documentation. Document retrieved from: http://scikit-learn.org/stable/supervised_learning.html

Ensemble methods – scikit-learn 0.18.1 documentation. Document retrieved from: http://scikit-learn.org/stable/modules/ensemble.html