Machine Learning Engineer Nanodegree

Capstone Project

Gabriel Augusto

March 10th, 2019

I. Definition

Project Overview

The group of Journalism Globo made a study of flights delay from more than 2.8 million flights in Brazil revealing that approximately 20% of the entire scheduled flights had a delay [1]. Airline delays cost usually billions of dollars per year having many causes for example extreme weather, late-arriving, security.

Weather is one of the main cause of airline delays having for about 40% of total delay minutes [2]. Many company, like Google [3], have interests on studying flights delays caused by problematic weather condition, helping airlines companies to schedule solutions for delay or informing flight status for passengers. .

In this project i analyzed 6 airport flights in Brazil from 2016 to 2017 :

SBSP,Congonhas

SBGR,Guarulhos - Governador Andre Franco Montoro

SBKP,Viracopos

SBCT,Afonso Pena

SBPA,Salgado Filho

SBGL,Aeroporto Internacional Do Rio De Janeiro/Galeao

The idea was to create a model that predict if a flight will be canceled or will have a delay using METeorological Aerodrome Report (METAR) information to train the system and test with other METAR of forecast predictions, these airports were chosen because they have big infrastructure and are relatively close to each other.

Problem Statement

The main goal in this project is to create a model that is able to predict if a flight is going to have a delay or if it would be canceled following the steps below:

1. Get METAR data combined with flight data from the region that is being study.
2. Keep only data that contains flights concluded with success or that had been canceled or delayed because bad meteorological conditions.
3. AtrasoVoo data for a best model creation (if after want to creat a regression model) and every flight with weather problem and delay higher than 10 minutes are not considered.
4. Execute multiple machine learning algorithms treatening imbalanced data (the number of flights with no problem is much higher than cancelled or delayed flights)
5. Check results for the executed machine learning algorithms

The expected results are a high hit rate for testing data treatening false positive and true negative as well.

Metrics

Fbeta Score (explicar a escolha do Fbeta score)

The metric chosen was FBeta Score if beta equals to 1, to define it is necessary to know the concepts of Precision and Recall that follows the equations below:

$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$ and $Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$

Using Precision and recall we obtain the Fbeta Score, multiplyed for $(1 + \beta)$, which $\beta\ ,\ 0 \leq \beta \leq 1:$

$$F\beta = (1 + \beta) \star \frac{precision * recall}{precision + recall}$$

For β in the range of 0.0 to 0.5 gives weight to Recall

For β in the range of 0.5 to 1.0 gives weight to Precision.

And create a confusion matrix to help understanding the results.

II. Analysis

Data Exploration

The dataset of this project were created merging two different datasets.

The first one was obtained from Iowa Environmental Mesonet containing METAR information of brazilian airports in 2016 [4]. The second dataset was obtained from a Kaggle competition (BrFlights2) containing flights tracked by the National Civil Aviation Agency (ANAC) [5], in Brazil, from January 2015 to August 2017.

The datasets were combined by the rounded hour and the respective airport having at the end all information from Metar and information from ANAC about the status of the flight (if a flight was delayed or cancel and it’s departure and arrival time).

The first one contained these columns:

Airport and Hour that were used to merge with the second dataset.

Cod or Codigo.Justificativa -> If the flight was delayed or canceled 1, else 0

tmpf: Air Temperature in Fahrenheit, typically @ 2 meters

dwpf: Dew Point Temperature in Fahrenheit, typically @ 2 meters

relh: Relative Humidity in %

drct: Wind Direction in degrees from north

sknt: Wind Speed in knots

alti: Pressure altimeter in inches

mslp: Sea Level Pressure in millibar

vsby: Visibility in miles

gust: Wind Gust in knots

skyc1, skyc2, skyc3, skyc4 -> sky coverage, related with visibility

skyl1, skyl2, skyl3, skyl4 -> sky level, related with visibility

The second dataset contained four columns:

Airport, Hour used to combine with the first dataset.

Codigo.Justificativa contained the name of flight status

Situacao.Voo saying if the flight was Ok or canceled

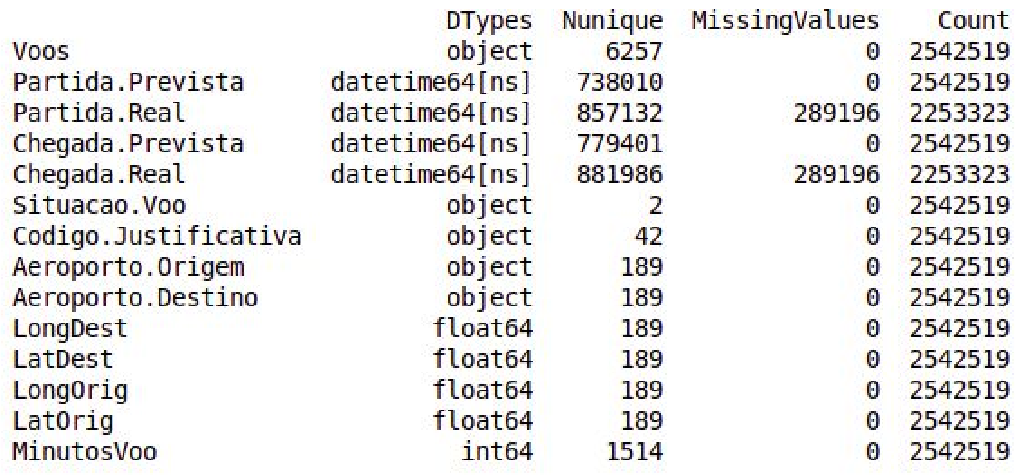
Estimative of time of departure (Partida.Prevista) and arrival (Chegada.Prevista) and the real departure (Partida.Real) time and arrival (Chegada.Real), and with them was created the column AtrasoVoo with the delayed minutes of a flight if the flight got early than expected were considered 0 and if the flight was canceled it was considered 100000 (to treat the missing values for arrival time when the flight is canceled)

In many columns as tmpf… there is the letter M that means not relevant time so it's set to 0.

In skyc1, skyc2, skyc3, skyc4 the categories are changed for numbers (0-8) as follows

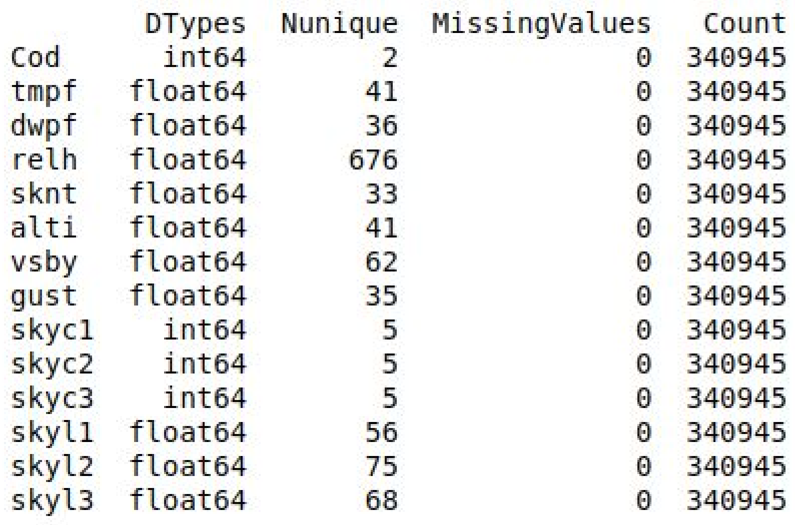
M = 0, FEW = 2, SCT = 4 , BKN = 6, OVC: 8, VV= : 8, /// = 8, NSC=0, NCD= 2 (according to https://www.skybrary.aero/index.php/Meteorological_Terminal_Air_Report_(METAR))

and skyl1, skyl2, skyl3, skyl4 has letter M as well, but in this case means 10000.



```
                              DTypes  Nunique  MissingValues    Count
Voos                          object     6257              0  2542519
Partida.Prevista      datetime64[ns]   738010              0  2542519
Partida.Real          datetime64[ns]   857132         289196  2253323
Chegada.Prevista      datetime64[ns]   779401              0  2542519
Chegada.Real          datetime64[ns]   881986         289196  2253323
Situacao.Voo                  object        2              0  2542519
Codigo.Justificativa          object       42              0  2542519
Aeroporto.Origem              object      189              0  2542519
Aeroporto.Destino             object      189              0  2542519
LongDest                     float64      189              0  2542519
LatDest                      float64      189              0  2542519
LongOrig                     float64      189              0  2542519
LatOrig                      float64      189              0  2542519
MinutosVoo                     int64     1514              0  2542519
```

The dataset in the end of merging contains the columns below:



```
        DTypes  Nunique  MissingValues   Count
Cod      int64        2              0  340945
tmpf   float64       41              0  340945
dwpf   float64       36              0  340945
relh   float64      676              0  340945
sknt   float64       33              0  340945
alti   float64       41              0  340945
vsby   float64       62              0  340945
gust   float64       35              0  340945
skyc1    int64        5              0  340945
skyc2    int64        5              0  340945
skyc3    int64        5              0  340945
skyl1  float64       56              0  340945
skyl2  float64       75              0  340945
skyl3  float64       68              0  340945
```

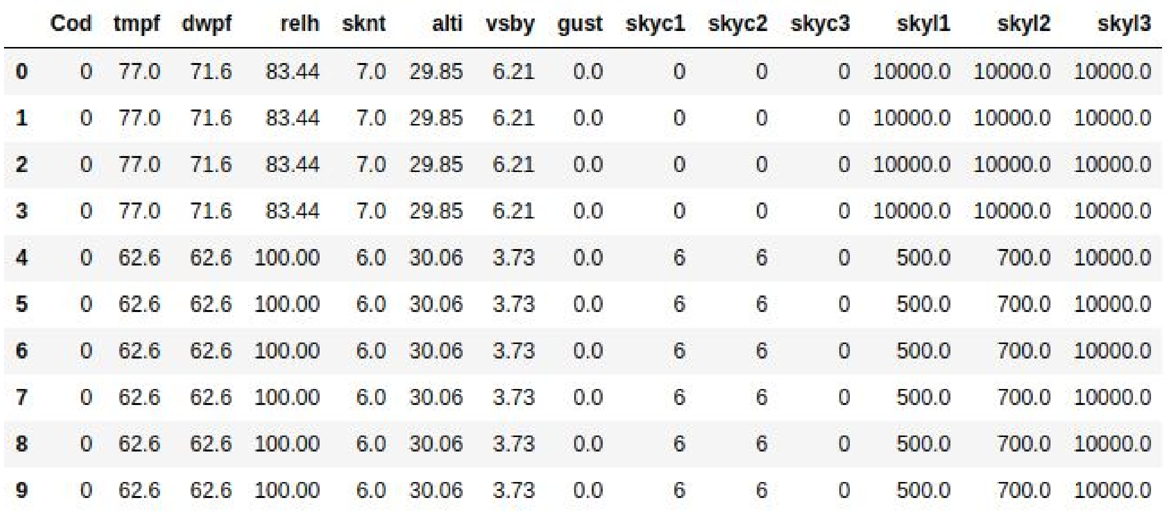
DTypes: shows the type of variable.

Nunique: How many different values the column of the dataset has.

MissingValues: As the name says if there are any null values.

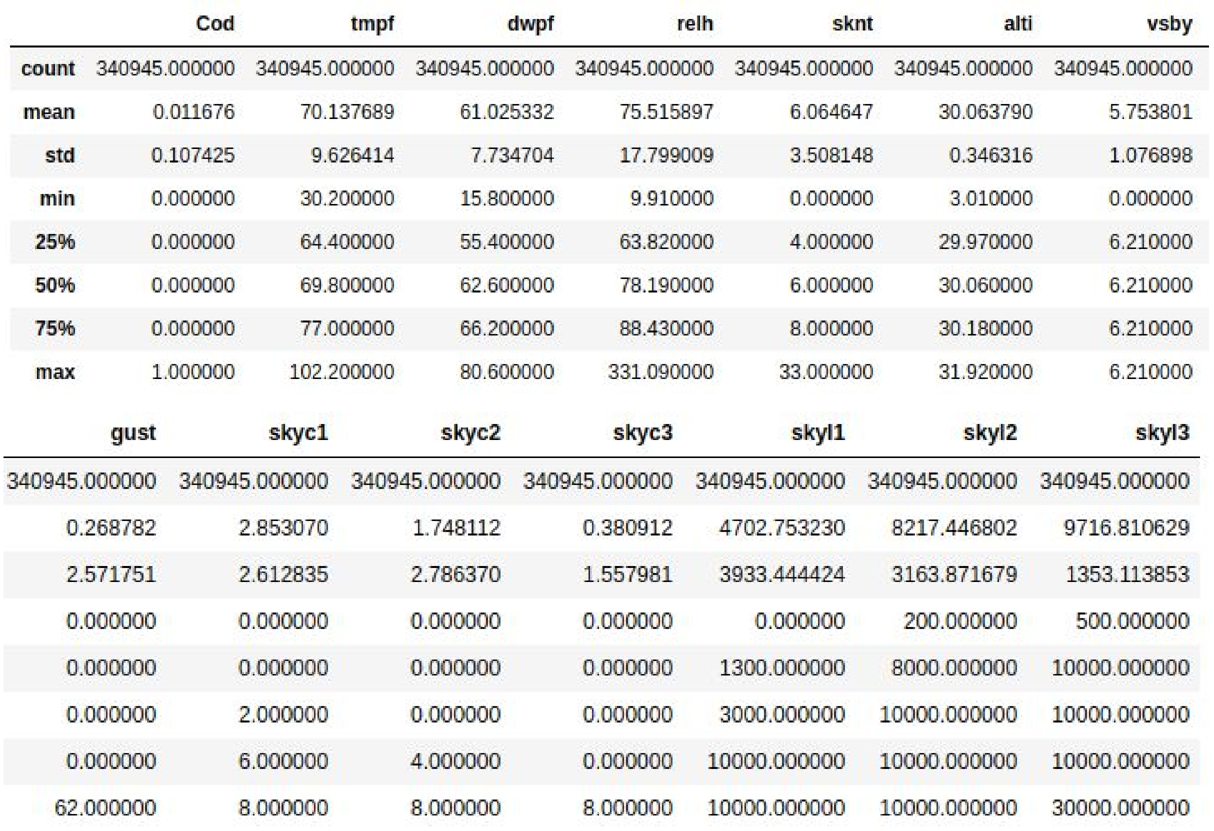
Count: Number of registers of each column

Example of merged dataset:



	Cod	tmpf	dwpf	relh	sknt	alti	vsby	gust	skyc1	skyc2	skyc3	skyl1	skyl2	skyl3
0	0	77.0	71.6	83.44	7.0	29.85	6.21	0.0	0	0	0	10000.0	10000.0	10000.0
1	0	77.0	71.6	83.44	7.0	29.85	6.21	0.0	0	0	0	10000.0	10000.0	10000.0
2	0	77.0	71.6	83.44	7.0	29.85	6.21	0.0	0	0	0	10000.0	10000.0	10000.0
3	0	77.0	71.6	83.44	7.0	29.85	6.21	0.0	0	0	0	10000.0	10000.0	10000.0
4	0	62.6	62.6	100.00	6.0	30.06	3.73	0.0	6	6	0	500.0	700.0	10000.0
5	0	62.6	62.6	100.00	6.0	30.06	3.73	0.0	6	6	0	500.0	700.0	10000.0
6	0	62.6	62.6	100.00	6.0	30.06	3.73	0.0	6	6	0	500.0	700.0	10000.0
7	0	62.6	62.6	100.00	6.0	30.06	3.73	0.0	6	6	0	500.0	700.0	10000.0
8	0	62.6	62.6	100.00	6.0	30.06	3.73	0.0	6	6	0	500.0	700.0	10000.0
9	0	62.6	62.6	100.00	6.0	30.06	3.73	0.0	6	6	0	500.0	700.0	10000.0

A describing stats of each column is showed below, showing the count, mean, standard deviation, the minimum value, maximum value, where 25%, 50% and 75% of data is:



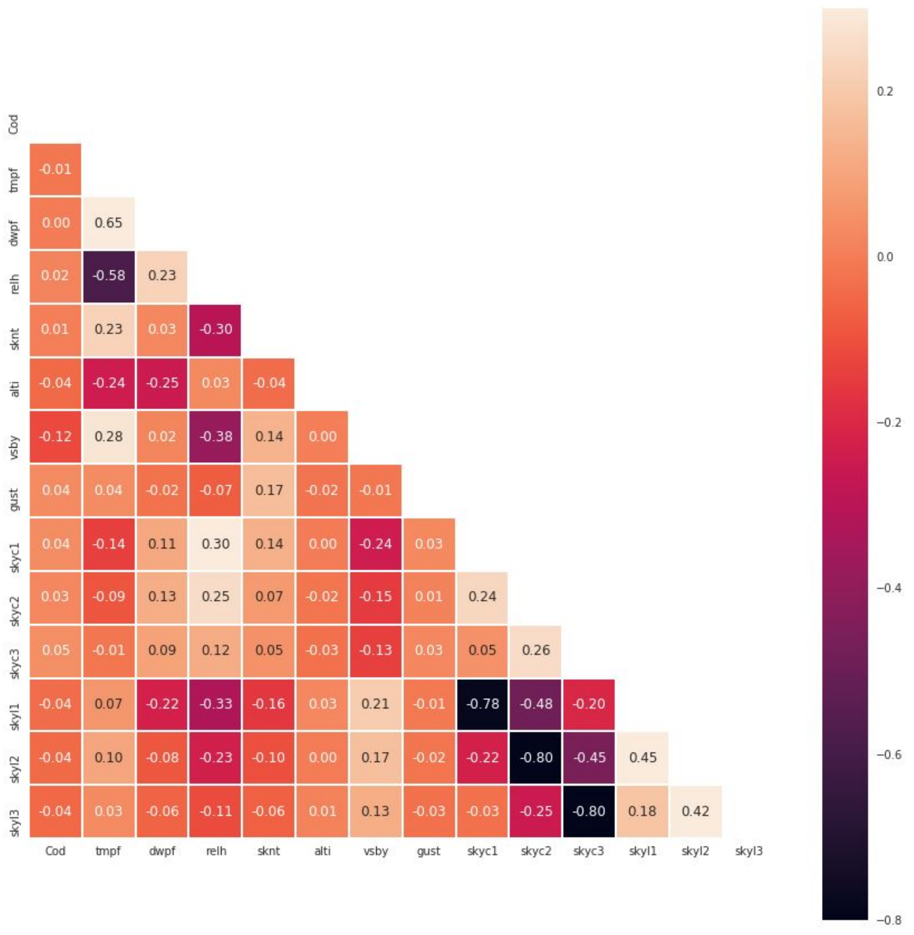
	Cod	tmpf	dwpf	relh	sknt	alti	vsby
count	340945.000000	340945.000000	340945.000000	340945.000000	340945.000000	340945.000000	340945.000000
mean	0.011676	70.137689	61.025332	75.515897	6.064647	30.063790	5.753801
std	0.107425	9.626414	7.734704	17.799009	3.508148	0.346316	1.076898
min	0.000000	30.200000	15.800000	9.910000	0.000000	3.010000	0.000000
25%	0.000000	64.400000	55.400000	63.820000	4.000000	29.970000	6.210000
50%	0.000000	69.800000	62.600000	78.190000	6.000000	30.060000	6.210000
75%	0.000000	77.000000	66.200000	88.430000	8.000000	30.180000	6.210000
max	1.000000	102.200000	80.600000	331.090000	33.000000	31.920000	6.210000

gust	skyc1	skyc2	skyc3	skyl1	skyl2	skyl3
340945.000000	340945.000000	340945.000000	340945.000000	340945.000000	340945.000000	340945.000000
0.268782	2.853070	1.748112	0.380912	4702.753230	8217.446802	9716.810629
2.571751	2.612835	2.786370	1.557981	3933.444424	3163.871679	1353.113853
0.000000	0.000000	0.000000	0.000000	0.000000	200.000000	500.000000
0.000000	0.000000	0.000000	0.000000	1300.000000	8000.000000	10000.000000
0.000000	2.000000	0.000000	0.000000	3000.000000	10000.000000	10000.000000
0.000000	6.000000	4.000000	0.000000	10000.000000	10000.000000	10000.000000
62.000000	8.000000	8.000000	8.000000	10000.000000	10000.000000	30000.000000

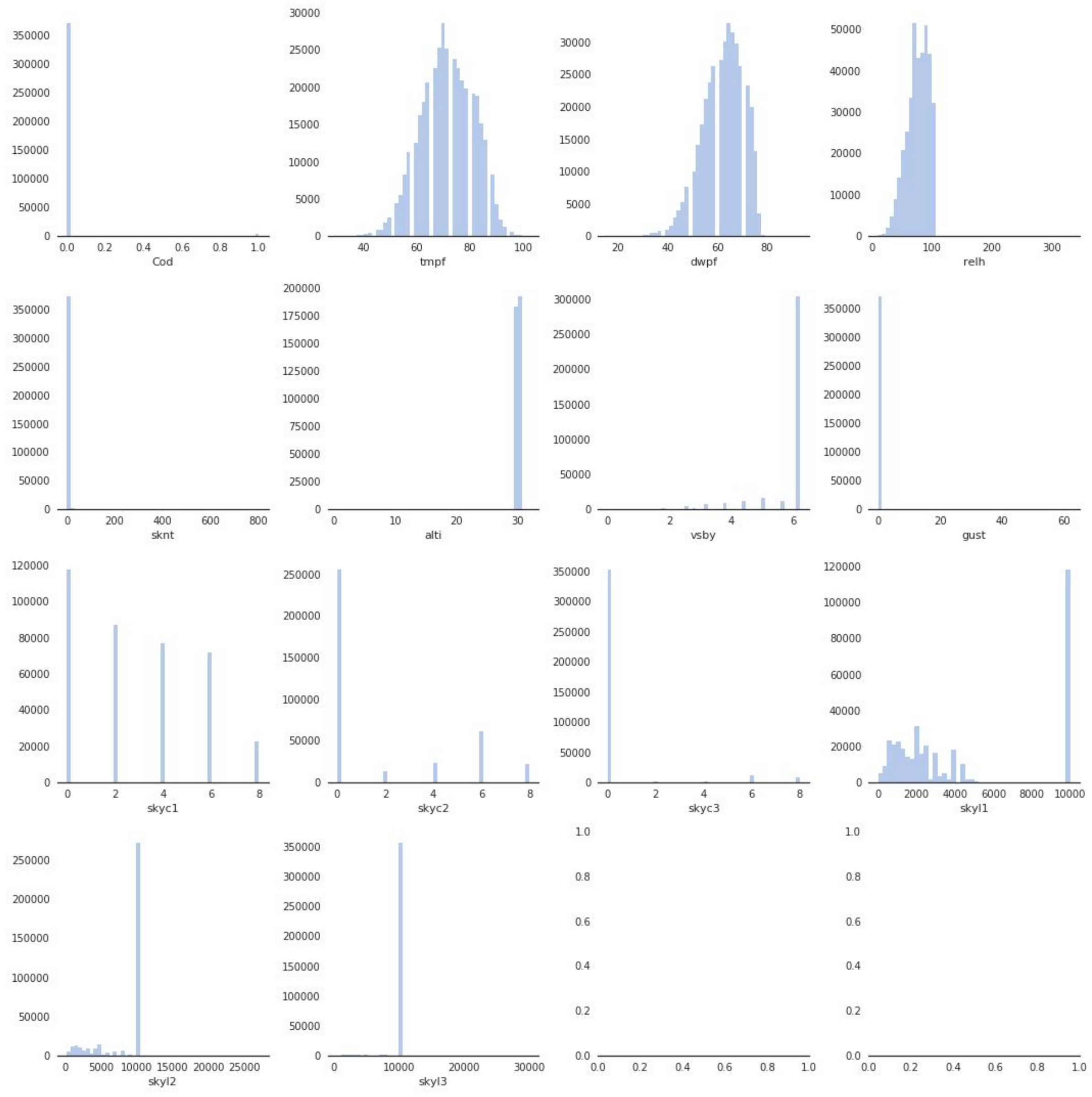
Exploratory Visualization

The image below shows the correlation between the columns. Sky coverages (skyc1,skyc2,skyc3) are inversely correlated with the sky levels (skyl1, skyl2, skyl3); the temperature of the airport are a bit correlated with the dew point. The most important observation is that nothing very correlated with the “Codigo.Justificativa” that we are trying to predict, so probably if there is a solution, starts by the combination of the columns of the dataset.

---------- Dataset Heatmap----------



The distribution of values from each column are represented below:



Codigo.Justificativa that we are trying to predict have only three values, but we can see that the values are imbalanced. The tmpf, dwpf, relh, drct skyl 1 and 2 shows to have a bigger standard deviation than the other columns.

Algorithms and Techniques

There are many Data Mining approaches for Data Balancing. I chose to use a popular approach named RandomUnderSampler.

Clustering is an Unsupervised Learning Approach. But RandomUnderSampler, only uses the concept of finding cluster centroid (clusters are created encircling data-points belonging to the majority class), as already instances are labelled. The cluster centroid is found by obtaining the average feature vectors for all the features, over the data points belonging to the majority class in feature space.

[4]

For classify the test flights i chose 3 algorithms there are popular in classification problems and easy to use and understand:

1. Random Forest :

   “A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting”. [5] Random forest obtains a class vote from each tree, and then classifies a sample using majority vote.

   $\hat{C}_{rf}(x) = \text{majority vote}\{\hat{C}_b(x)\}^B$ [2]

2. KNN :

   “Neighbors-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point”. [6] To assign a label to the point x, the k-Nearest-Neighbors classifier draws a sphere centered on x enclosing exactly k training points. Afterwards, it examines the label of k training points closest to . Then the label having the largest vote is assigned to the test point x ([7] apud [2])

.

3. Adaboost : An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases [8]. AdaBoost is a method of converting weak classifier into highly accurate prediction rule. It learns week learner on weighted example set sequentially and combines weak hypotheses linearly. Produced sequence of classifiers is dependent on the previous one and focuses on the previous one's errors. Samples that are incorrectly predicted in the previous classifiers are chosen more often or weighted more heavily when estimating a new classifier ([7] apud [2])
4. k-Fold Cross-Validation : Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The general procedure is as follows:

   Shuffle the dataset randomly.

   Split the dataset into k groups

   For each unique group:

   Take the group as a hold out or test data set

   Take the remaining groups as a training data set

   Fit a model on the training set and evaluate it on the test set

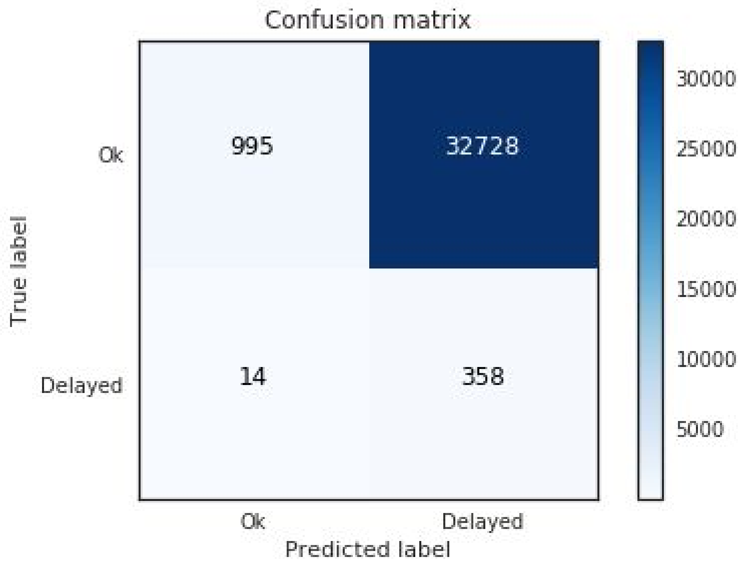
   Retain the evaluation score and discard the model

   Summarize the skill of the model using the sample of model evaluation scores

   [12]

Benchmark

The benchmark model chosen is a simple decision Tree. To compare its results with the random forest, KNN and Adaboost. The decision tree gives a terrible answer classifying a lot of On-time flight as delayed.



	Benchmark
F1 Score (%)	2.14
Accuracy (%)	0.39
AUC	49.59

III. Methodology

Data Preprocessing

Flights data:

1. Hour to datetime
2. Change name of chosen airport to their respective code example: Guarulhos to SBGR
3. Keep only empty Codigo.Justificativa:

   and the rows that contains:

   AEROPORTO ORIGEM ABAIXO DOS LIMITES, AEROPORTO DESTINO ABAIXO DOS LIMITES, ATRASO DEVIDO RETORNO - CONDICOES METEOROLOGICAS, CANCELAMENTO - CONEXAO AERONAVE/VOLTA - VOO DE IDA CANCELADO - CONDICOES METEOROLOGICAS

   if Codigo.Justificativa is null = 0

   else if Codigo.Justificativa is AEROPORTO ORIGEM ABAIXO DOS LIMITES, AEROPORTO DESTINO ABAIXO DOS LIMITES, ATRASO DEVIDO RETORNO - CONDICOES METEOROLOGICAS is set to 1,

   else = 2

4. Keep only data from flights of chosen airports, Create columns Aeroporto.Importante, Hora.Prevista, Hora.Real if at departure:

   Aeroporto.Importante = Aeroporto.Origem, Hora Prevista = Partida.Prevista e Hora.Real = Partida.Real

   else

   Aeroporto.Importante = Aeroporto.Destino, Hora Prevista = Chegada.Prevista e Hora.Real = Chegada.Real

5. Create column AtrasoVoo that contains the flight delay in minutes if the flight was canceled the it values is set to a huge value (100000), to if want later to make a regression of the delay time.

Airport data:

1. Merge files of chosen airports (one file per airport)
2. Replace values for skycs, skyls and other categorical columns

After these steps:

Merge tables by Station and valid from the second dataset with Aeroporto.Importante and Hora.Prevista from the first dataset. Remove unnecessary columns.

To finish the preprocess was treated the imbalanced data using RandomUnderSampler.

The data was normalized and splitted for training and test data. After that was used gridSearch with cross-validation (cv=10) having the scoring method as the accuracy to help training the data, changing a few hyperparameters trying to get a better result, the n_estimators in Random Forest and in Adaboost and the value of K for KNN.

Implementation

The metrics chosen were accuracy score and Fbeta score that uses precision and recall score as said before. For all algorithms was used RandomUnderSampler technique. Because when used oversampling technique grew almost exponentially.

Was observed in papers that the problem of classification in Ok, Delayed and Canceled, was reduced to a binary problem Ok and Delayed following the steps of [2]. So canceled flights by weather conditions had the Cod (Codigo.Justificativa) changed for 1 as delayed flight and On-time flight were kept 0.

Flights with a few minutes (1-10 minutes) delay were deleted. After that, flights with missing values were treated.

If the column tmpf or dwpf or vsby or alti or relh or sknt having missing data (represented for the letter “M”) the row was dropped)

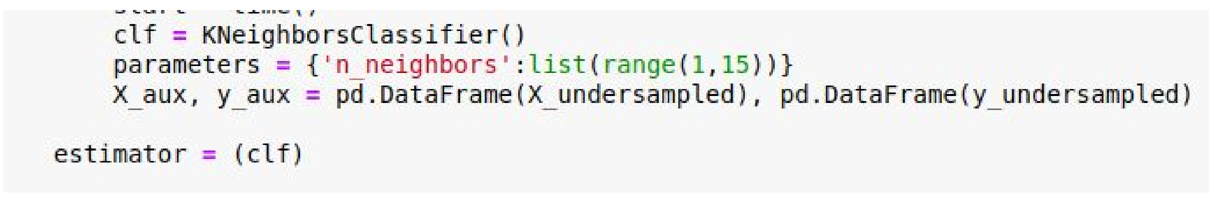
The missing data of columns gust, skyc1, skyc2, skyc3 the cell value was turned to 0, meaning that the data was not significant.

For skyl1, skyl2, skyl3 had missing values the cell value was turned to 10000, meaning that the data was not significant.

At last were treated categorical data of skyc1, skyc2, skyc3 as said before.

The data was splitted in training data (90%) and test data(10%).

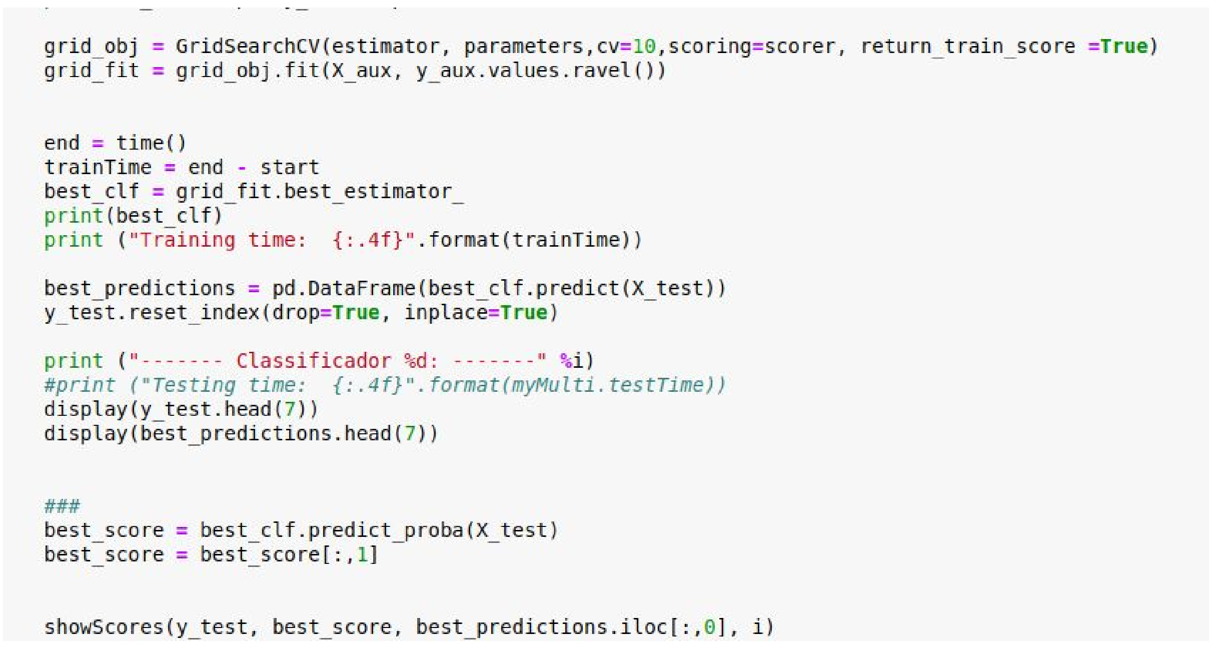
Example of classifier and their parameters (in this case KNN):



```
     clf = KNeighborsClassifier()
     parameters = {'n_neighbors':list(range(1,15))}
     X_aux, y_aux = pd.DataFrame(X_undersampled), pd.DataFrame(y_undersampled)

 estimator = (clf)
```

Usage of grid search to implement 10 fold cross validation using the classifier (estimator) and their parameters and scoring as F1-score and fitting with the training data. After that was kept the best predictor in the variable best_clf and used it to show it scores (function showScores) comparing the best_predictions with the test data (y_test) for each classifier.



```
grid_obj = GridSearchCV(estimator, parameters,cv=10,scoring=scorer, return_train_score =True)
grid_fit = grid_obj.fit(X_aux, y_aux.values.ravel())


end = time()
trainTime = end - start
best_clf = grid_fit.best_estimator_
print(best_clf)
print ("Training time:  {:.4f}".format(trainTime))

best_predictions = pd.DataFrame(best_clf.predict(X_test))
y_test.reset_index(drop=True, inplace=True)

print ("------- Classificador %d: -------" %i)
#print ("Testing time:  {:.4f}".format(myMulti.testTime))
display(y_test.head(7))
display(best_predictions.head(7))


###
best_score = best_clf.predict_proba(X_test)
best_score = best_score[:,1]


showScores(y_test, best_score, best_predictions.iloc[:,0], i)
```

Resuming the biggest complication was:

1. Big dataset (but even being a big data set probably it needed to be bigger to train the model)
2. Imbalanced data

3. Multiclass problem that was turned to a binary problem to use the roc curve metric easily
4. Treat categorical data of the problem

Refinement

For trying to refine the results. i used Undersampling technique for RandomForest, Adaboost and KNN, using the combination oversampling and undersampling was used SMOTENN that balance the data and remove noisy samples. The SmoteNN was used only for RandomForest and Adaboost, because for KNN the training time it would be very high.

(Was used for all cases the random_state =200)

Tuning hyperparameters:

For Random Forest with undersampling:

"n_estimators": [100,500,900]

For Adaboost with undersampling:

"base_estimator": [DecisionTreeClassifier(max_depth=8)], "n_estimators": [100,500,900], "learning_rate":[1,2]

For KNN (only undersampling):

n_neighbours varying from 1 to 15.

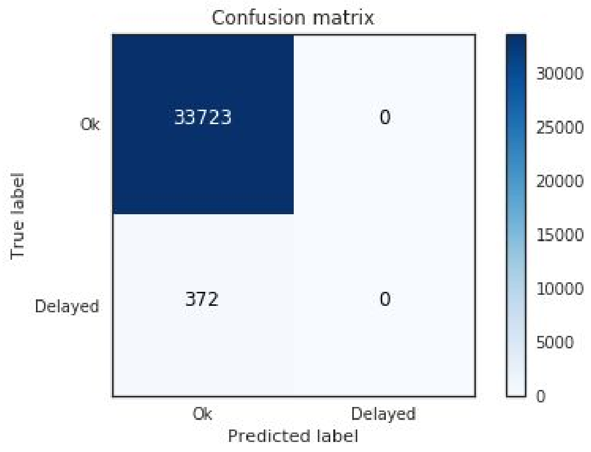
IV. Results

Model Evaluation and Validation

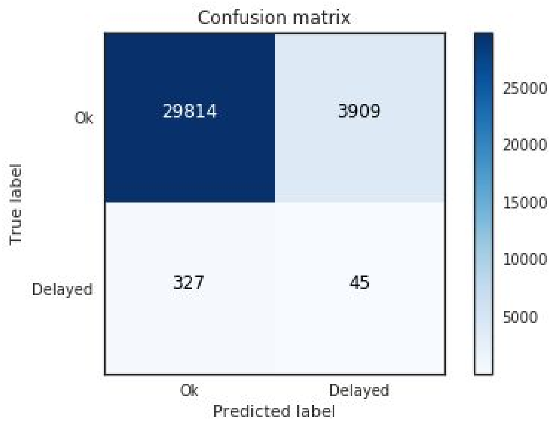
The results obtained are represented below:

Confusion Matrix:

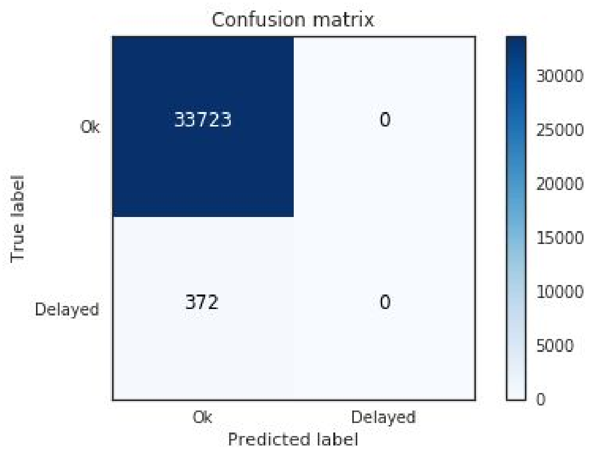
Random Forest :



KNN:



Adaboost:



Accuracy and Fbeta Score:

	Random Forest	KNN	Adaboost	Benchmark
F1 Score (%)	0.0	2.08	0.0	2.14
Accuracy (%)	98.90	87.57	98.90	0.39
AUC	58.90	50.25	55.62	49.59

The benchmark had terrible results, the classifiers used (Random Forest, KNN and Adaboost seems to have a better results, but they don't. They practically overfit, probably because there is not delayed samples enough to train the model and showing that it can be different for each airport and maybe year season. So probably it will not work well for unseen data as we can see in the confusion matrix and it can not be trusted for predicting delayed flights because it's always saying that the flight will be Ok(for random forest and adaboost), or for KNN that has a big false positive rate and false negative rate as well. But this problem may have solution commented in Improvement section.

Refinement

The best model was RandomFOrest with the followed parameters :

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=-1, oob_score=False, random_state=200, verbose=0, warm_start=False)

Justification

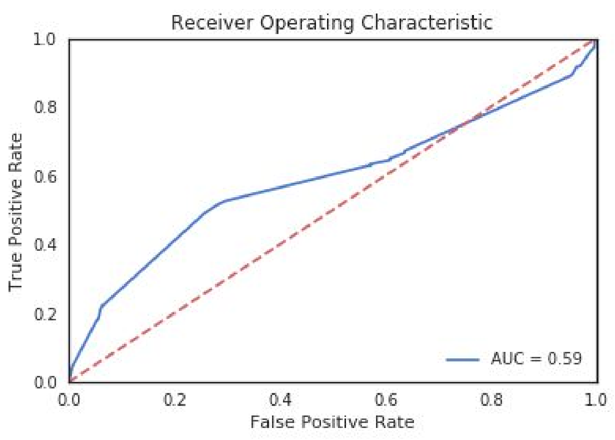
The benchmark established show that the problem is not so simple and it randomize the prediction, so the created models started to optimize the prediction, but it overfitted generalizing all results probably because it were used data from different airports with different weather and infrastructure conditions and also were used that from only one year.

V. Conclusion

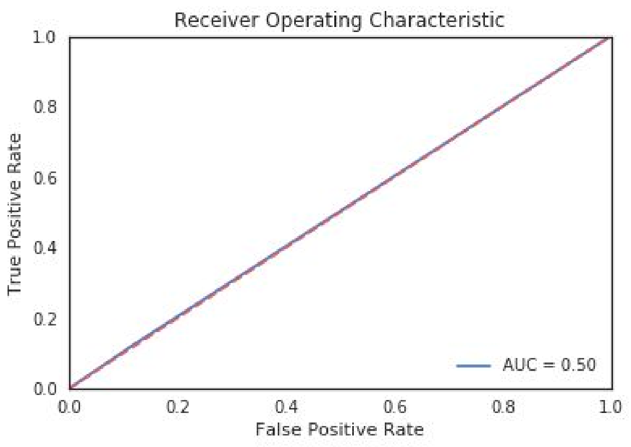
Free-Form Visualization

ROC Curve

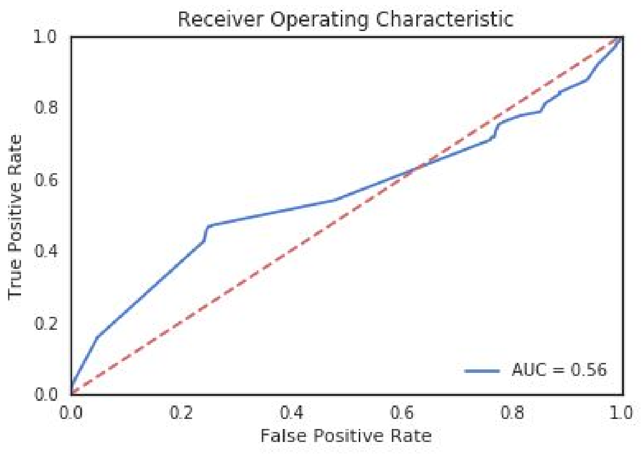
Random Forest:



KNN:



Adaboost:



These Roc curves shows a problem because the ROC Curve is too low (close to 0.5), so probably it’s necessary more data to train the model and obtain better results and made the prediction from each airport at time because the different conditions to departure and arrival from each airport.

The next study of these type of data need a bigger dataset (containing more than one year of data) having only one airport to reduce the noise added to the system .

Reflection

The project needed to merge two datasets and to understand the meaning of each row. The next step was to clean unnecessary columns data at the end the problem was to choose how to identify that a flight was delayed or canceled, needing to choose between a regression problem (estimate the time delay) and after a classification (if time is low the flight was Ok, if the time was medium was a delay and if the time was high probably was canceled), so i initially chose only to apply classification algorithms because it would be easy to check input data inconsistency.

Improvement

Improvement that could be made:

1. Test predictions only for one airport each time, because each airport has different conditions compared to others starting from the region and climate, but it would be necessary a dataset containing data of many years (not just one year).
2. After predicting if a flight is going to be canceled or delayed, make a regression to estimate the delayed minutes of flights.
3. Try to predict the delay using the METAR of an airport hours before the flight arrives.
4. Combine METAR and forecast to only predict flight delay.

V. References

1. https://infograficos.oglobo.globo.com/economia/raio-x-dos-atrasos-dos-voos.html
2. https://ieeexplore.ieee.org/document/7777956
3. https://techcrunch.com/2018/01/31/google-flights-will-now-predict-airline-delays-before-the-airlines-do/
4. https://mesonet.agron.iastate.edu/request/download.phtml?network=BR__ASOS
5. https://www.kaggle.com/ramirobentes/flights-in-brazil/home

6. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
7. https://scikit-learn.org/stable/modules/neighbors.html
8. C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.
9. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html
10. https://stackoverflow.com/questions/48817300/sklearn-plot-confusion-matrix-combined-across-trainingtest-sets (Confusion matrix plot)
11. https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152 (Multiclass ROC plot)
12. https://machinelearningmastery.com/k-fold-cross-validation/