Capstone Project

Machine Learning Engineer Nanodegree

Diogo Dutra

November 19, 2018

Predict Flight Delay

Definition

Project Overview – Air Traffic

Aerial commute is increasingly important as the globalization advances and the world population grows. However, the air traffic is also becoming a challenge, especially for the most used regional hubs. While transportation infrastructure is mainly a role for the governments, predicting the flight delays may be accessible for private initiative and it will benefit those passengers running tight on schedule by allowing them to reorganize their tasks on advance.

The most common causes of flight delays are varied [1]. On one hand some causes are not related to accessible data, but on the other hand others are within reach in advance of the flight. The inaccessible data will remain as noise caused from security, maintenance and disaster issues. The accessible data are weather and congestion that may be useful to predict some of the flight delays.

The inspiration for such topic is clear for the author because of a combination of being a frequent flyer and an experienced aeronautical engineer.

Problem Statement – How much will be the flight delay?

The predictive model shall be able to answer the following question:

Given the departure information, how many minutes will be the flight delay?

The input data are all the available data before the scheduled moment for the take-off such as time, date, airliner, flight number, air temperature, dew temperature, air pressure, visibility and type of sky on the departure airport.

Datasets and Inputs - ANAC and BDM

The considered data of flight departures and arrivals are the world busiest regional aerial commute in order to maximize the amount of available data. The Brazilian one-way commute from São Paulo

(Guarulhos airport) to Rio de Janeiro (Santos Dumont airport) is the most numerous by quantity of departures per day.

The 2018 historical dataset from January to September is available on the ANAC website [2]. The weather data is the METAR [3] type and specific to the departure airport available on the INPE's BDM website [4].

Both departure and weather data will be merged into one single table considering the nearest date and time. Such table will be the sample input to the pipeline.

Benchmark – Better than Naïve

The obtained predictive model should ideally be more accurate than the "naive guess", where "naive guess" means estimating null delay for every flight. For comparison, the same evaluation metrics are considered for both the predictive model and the "naïve guess".

Evaluation Metric - MSE

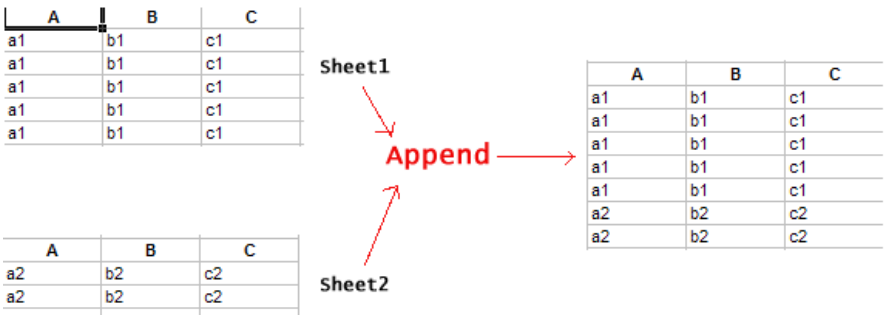
The metric to evaluate the performance of the models is the mean squared error [5] (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Analysis

Data Preprocessing

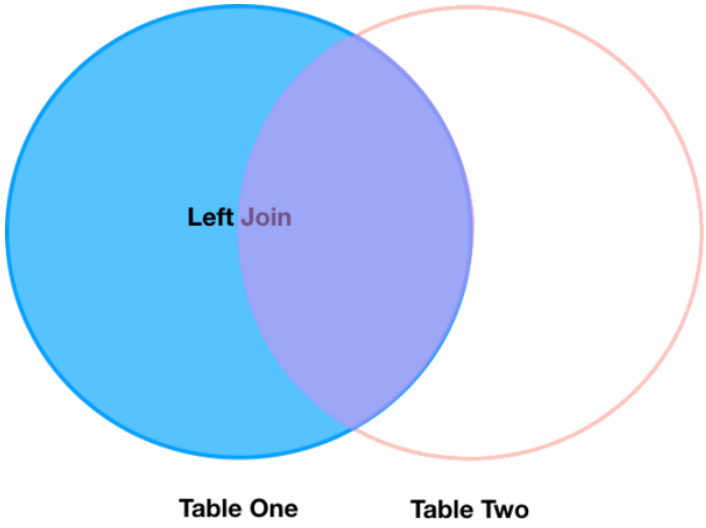
Both departure and weather data files separated by months. Therefore, they were downloaded separately, converted into tables and then appended as illustrated below.



SOURCE: HTTP://WWW.DIGDB.COM/EXCEL_ADD_INS/COMBINE_APPEND_TABLES_SHEETS_FILES/1.GIF

The departure table was filtered to discard all routes except the ones regarding the airliners one-way flights from Guarulhos (SBGR) to Santos Dumont (SBRJ) airports.

The departure and weather tables were merged by the nearest common date and time. The merge type was from left to right, meaning that useless weather data were discarded as illustrated below.



Source: https://datacarpentry.org/python-ecology-lesson/fig/left-join.png

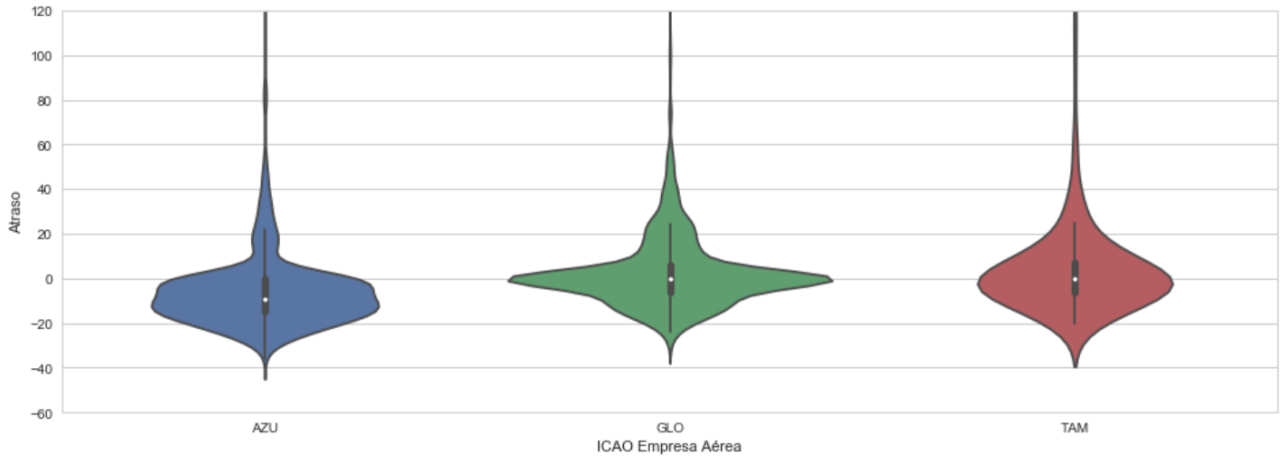
The **"Flight Delay"** column is calculated from the difference between the scheduled and the real arrivals in minutes. This column is the label to be predicted by the regression model.

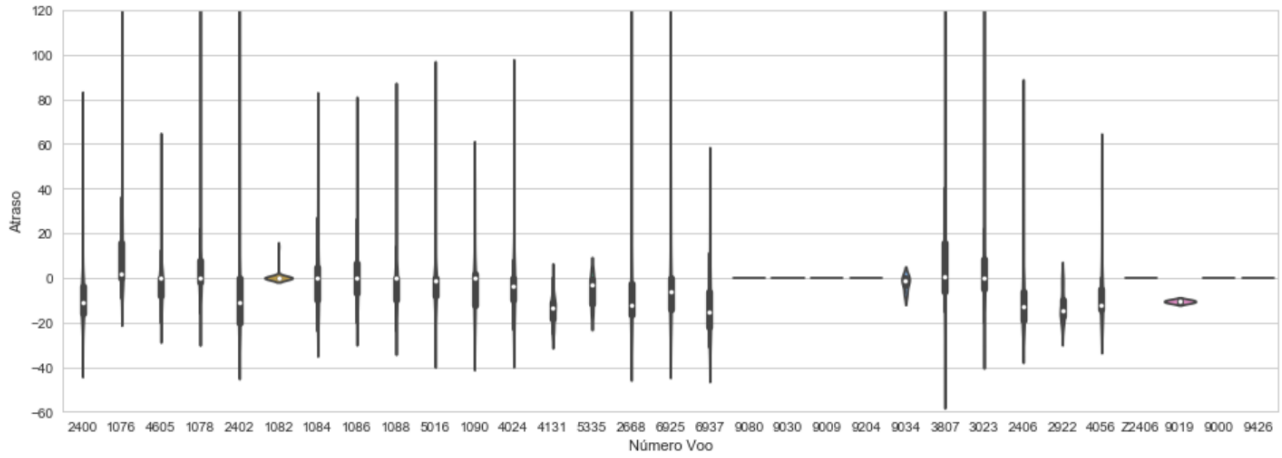
The features considered as categorical are: airliner, flight number, day of the month, month, day of the week, type of the first layer cloud, wind direction, coverage of the first layer cloud and type of sky.

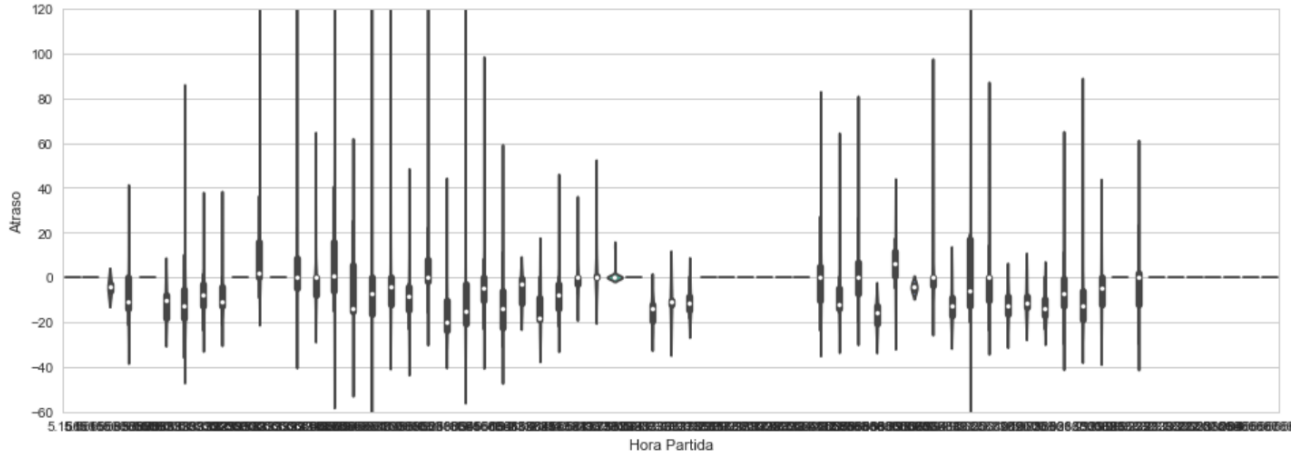
The resulted input table has 2599 samples (rows) with 17 features and 1 label (18 columns).

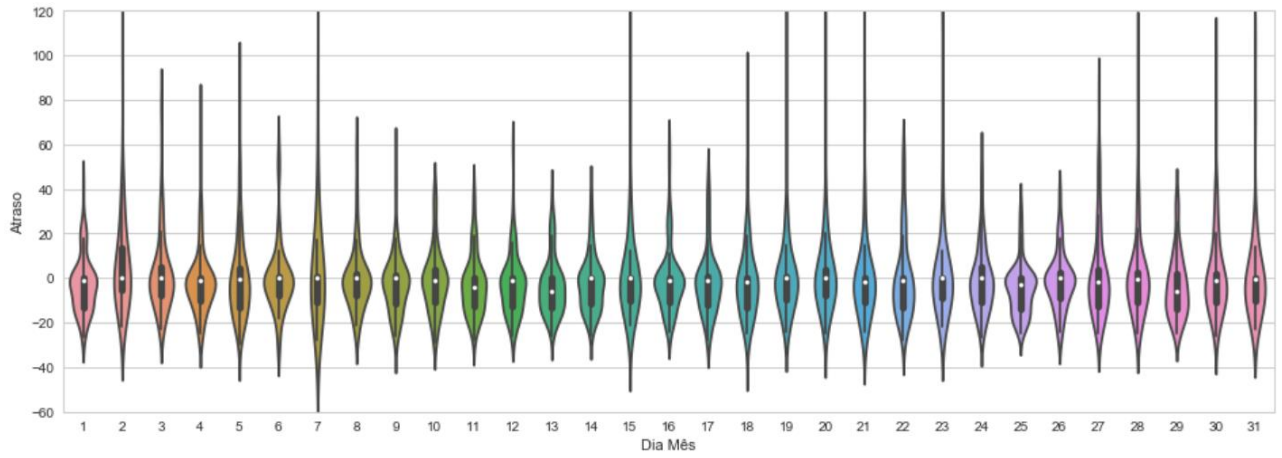
Exploratory Visualization

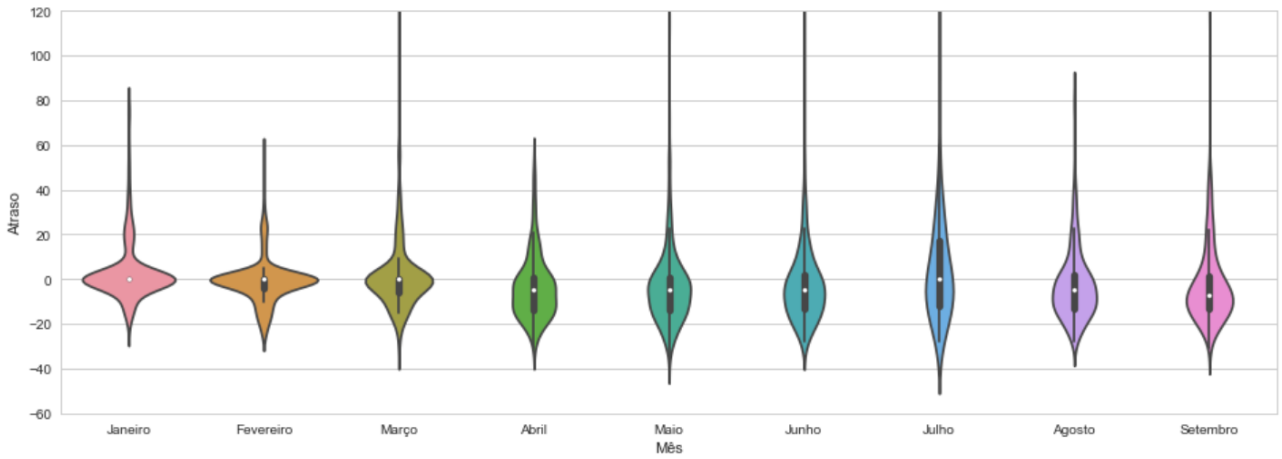
The prepared input table is used for a series of violin plots to better understand how is the data variation over its dimensions, focusing on how the delays are distributed by each feature.

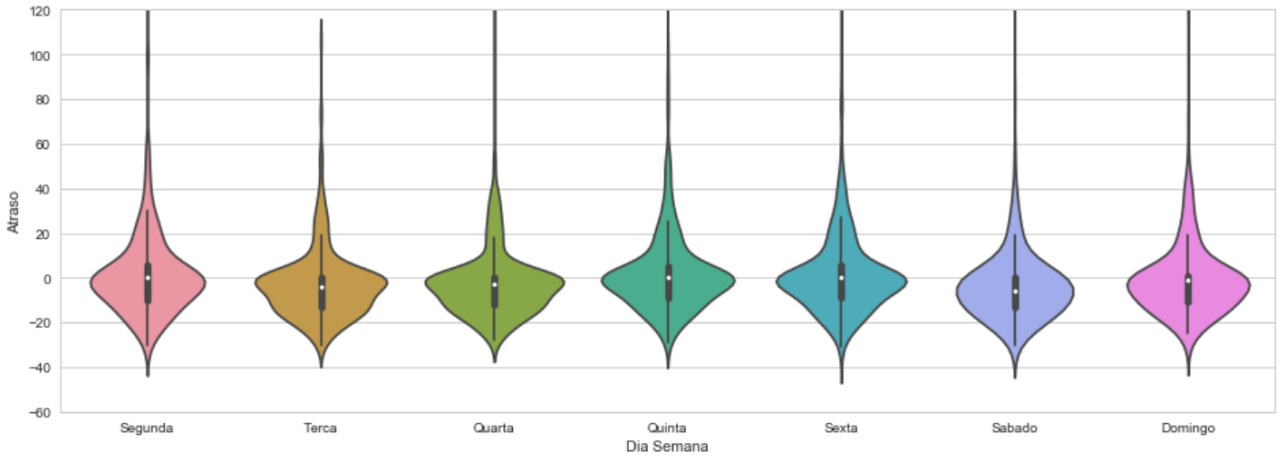


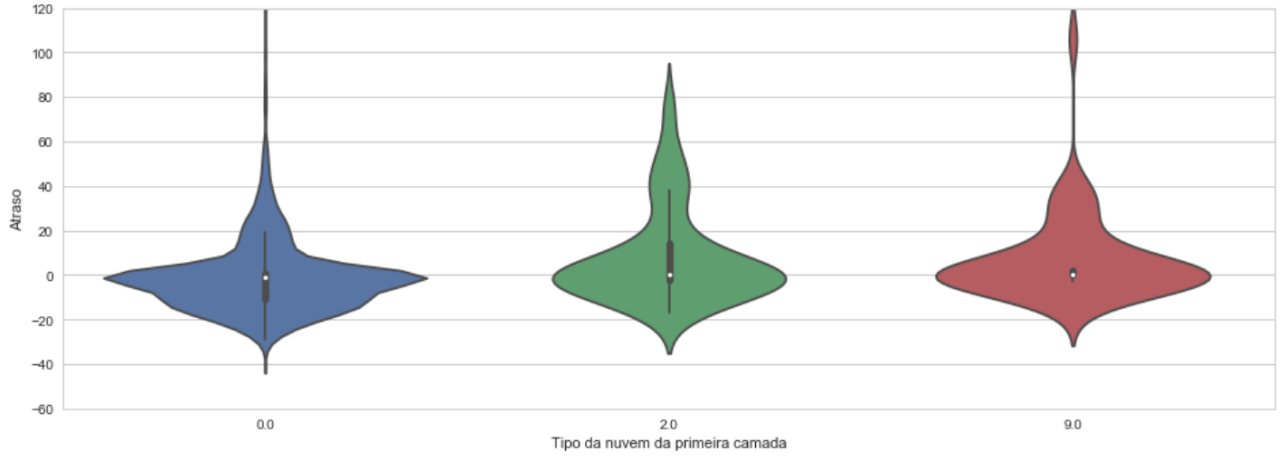

Atraso
Número Voo
2400 1076 4605 1078 2402 1082 1084 1086 1088 5016 1090 4024 4131 5335 2668 6925 6937 9080 9030 9009 9204 9034 3807 3023 2406 2922 4056 Z2406 9019 9000 9426

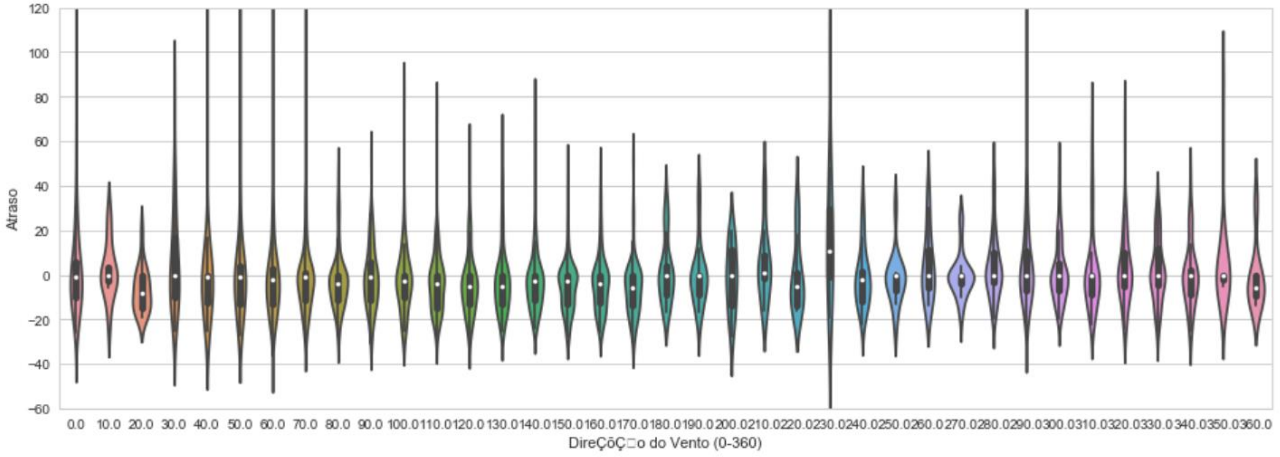

Atraso
Hora Partida

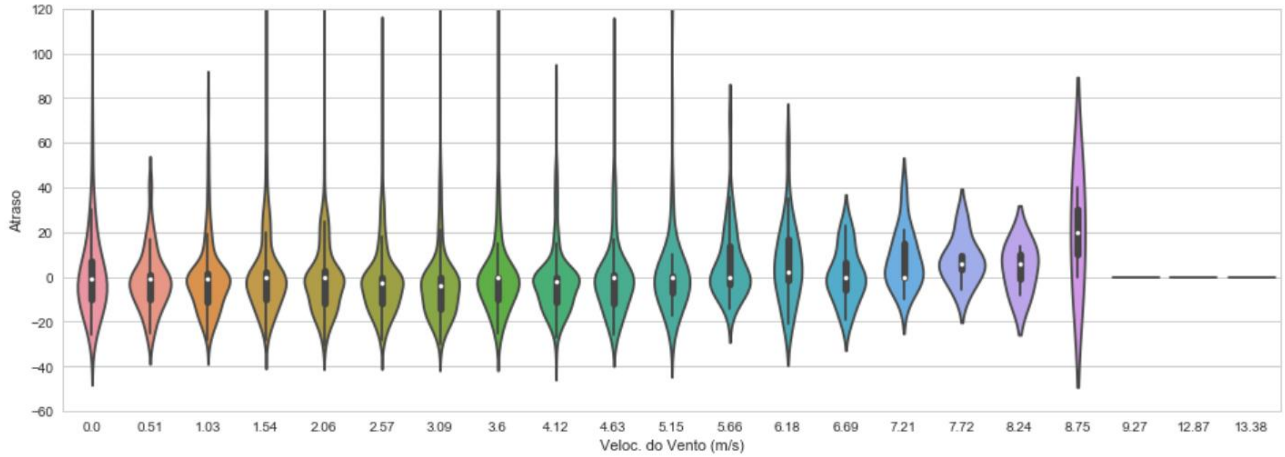

Atraso
Dia Mês
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

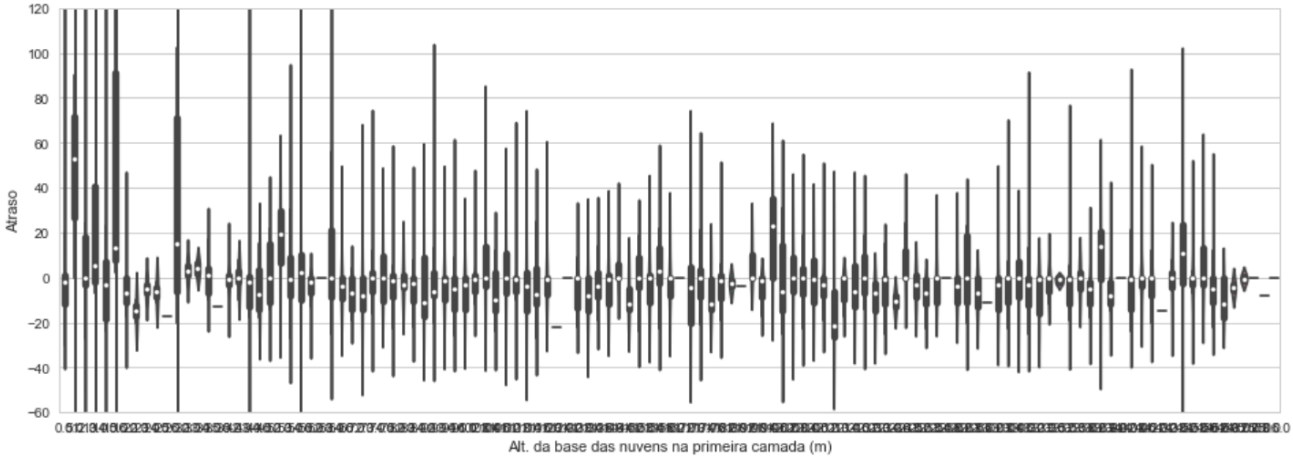

Atraso
Janeiro
Fevereiro
Março
Abril
Maio
Junho
Julho
Agosto
Setembro
Mês

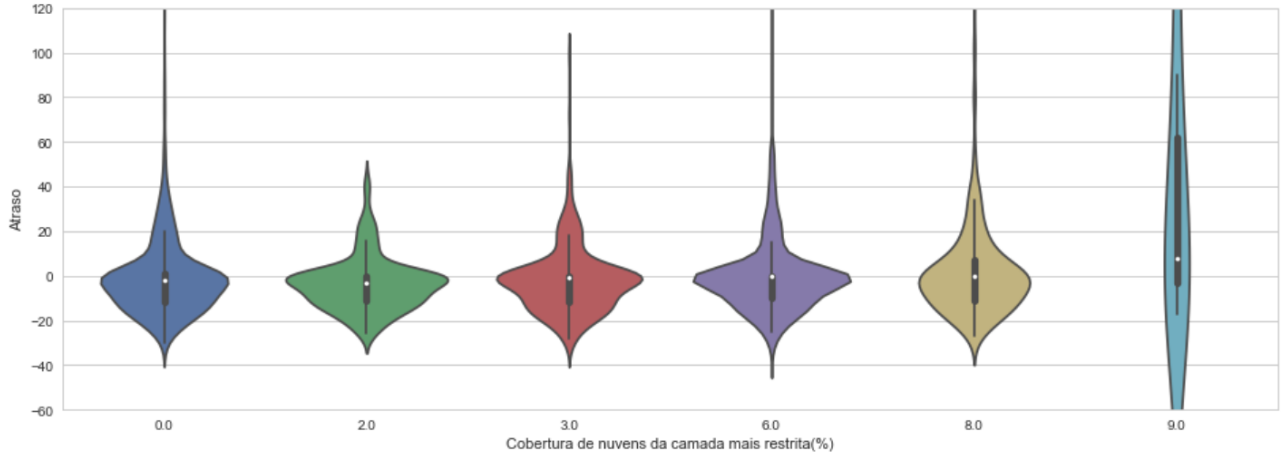

Atraso
120
100
80
60
40
20
0
−20
−40
−60
Segunda
Terca
Quarta
Quinta
Sexta
Sabado
Domingo
Dia Semana

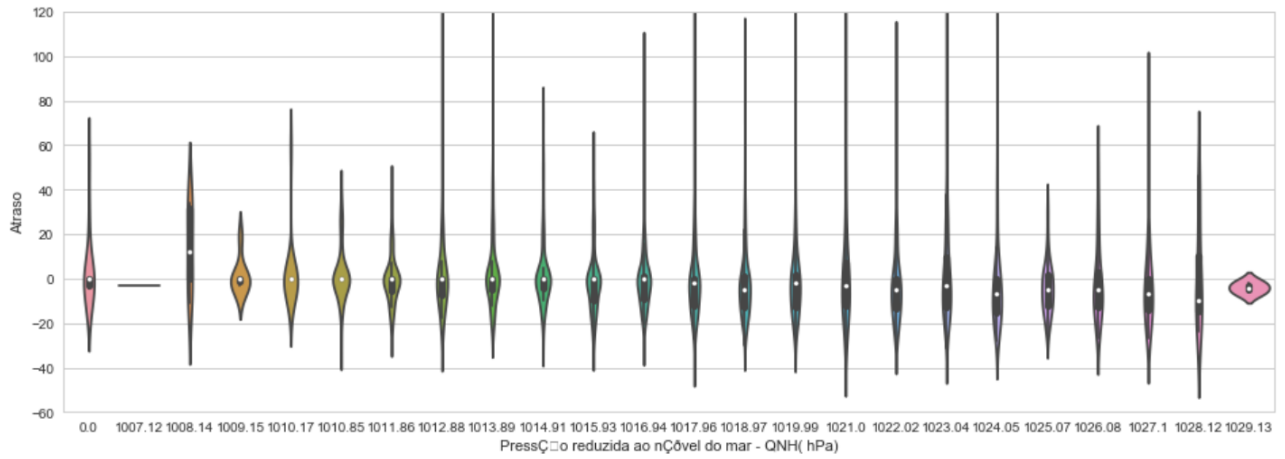

Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0
2.0
9.0
Tipo da nuvem da primeira camada

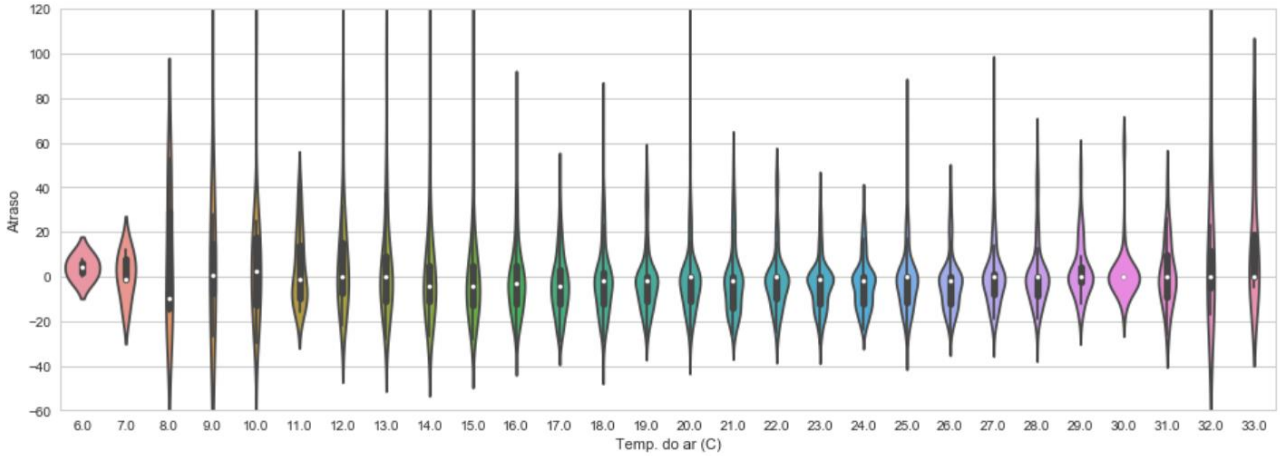

Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 100.0 110.0 120.0 130.0 140.0 150.0 160.0 170.0 180.0 190.0 200.0 210.0 220.0 230.0 240.0 250.0 260.0 270.0 280.0 290.0 300.0 310.0 320.0 330.0 340.0 350.0 360.0
DireÇõÇ□o do Vento (0-360)

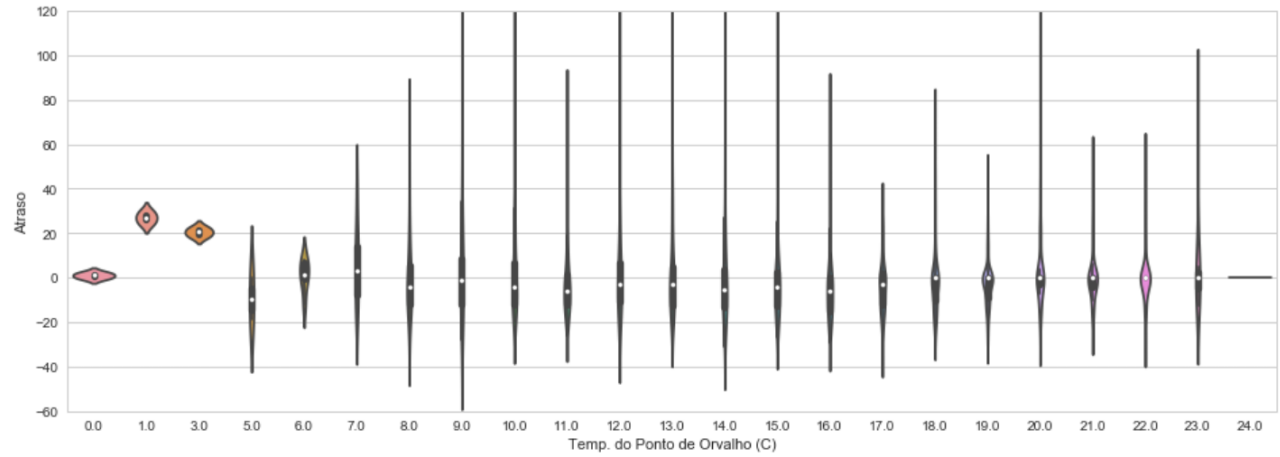

Atraso
120
100
80
60
40
20
0
-20
-40
-60
0.0 0.51 1.03 1.54 2.06 2.57 3.09 3.6 4.12 4.63 5.15 5.66 6.18 6.69 7.21 7.72 8.24 8.75 9.27 12.87 13.38
Veloc. do Vento (m/s)

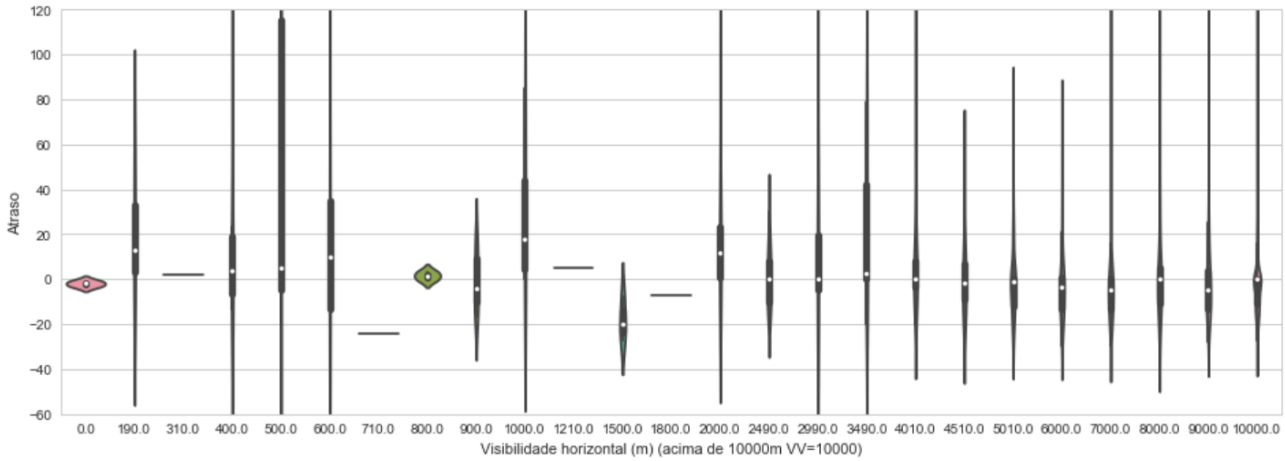

Atraso
120
100
80
60
40
20
0
-20
-40
-60
Alt. da base das nuvens na primeira camada (m)

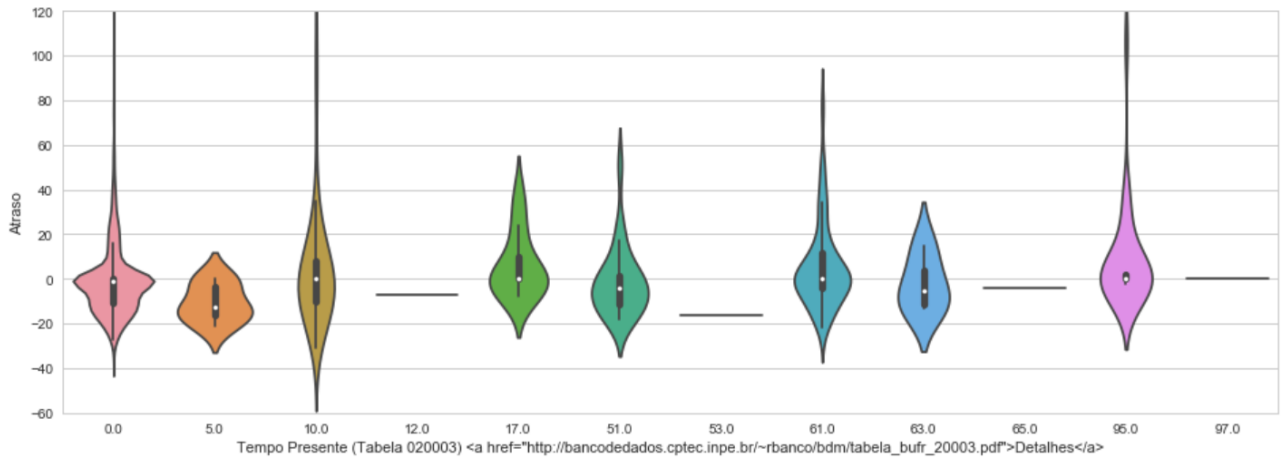

Atraso
120
100
80
60
40
20
0
-20
-40
-60
0.0 2.0 3.0 6.0 8.0 9.0
Cobertura de nuvens da camada mais restrita(%)


Atraso
120
100
80
60
40
20
0
-20
-40
-60
0.0 1007.12 1008.14 1009.15 1010.17 1010.85 1011.86 1012.88 1013.89 1014.91 1015.93 1016.94 1017.96 1018.97 1019.99 1021.0 1022.02 1023.04 1024.05 1025.07 1026.08 1027.1 1028.12 1029.13
PressÇ□o reduzida ao nÇôvel do mar - QNH(hPa)


Atraso
120
100
80
60
40
20
0
−20
−40
−60
6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0 30.0 31.0 32.0 33.0
Temp. do ar (C)


Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0 1.0 3.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0
Temp. do Ponto de Orvalho (C)


Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0 190.0 310.0 400.0 500.0 600.0 710.0 800.0 900.0 1000.0 1210.0 1500.0 1800.0 2000.0 2490.0 2990.0 3490.0 4010.0 4510.0 5010.0 6000.0 7000.0 8000.0 9000.0 10000.0
Visibilidade horizontal (m) (acima de 10000m VV=10000)


Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0 5.0 10.0 12.0 17.0 51.0 53.0 61.0 63.0 65.0 95.0 97.0
Tempo Presente (Tabela 020003) Detalhes

The plots remarks some useful insights:

- Most of the Azul's flights (AZU) has arrivals in advance while the others Gol (GLO) and Tam (TAM) are usually on time.
- Some flight numbers have arrivals in advance.
- Some times of the day have arrivals in advance while a few others have late arrivals.
- The months after March tend to have more arrivals in advance.
- Saturdays usually have more arrivals in advance.
- Wind direction 20^o leads to more arrivals in advance while 230^o tend to late arrivals.
- Wind speeds over 7 m/s leads to more late arrivals.
- Horizontal visibility over 10km altitude with 1.500m tend to have more arrivals in advance.
- Some types of sky tend to lead to more arrivals in advance.
- Some features have unnoticed contribution to the variation of flight delays (day of the month, type of cloud first layer, height of the cloud first layer, most restricted could coverage, seal-level air pressure, air temperature, dew point temperature

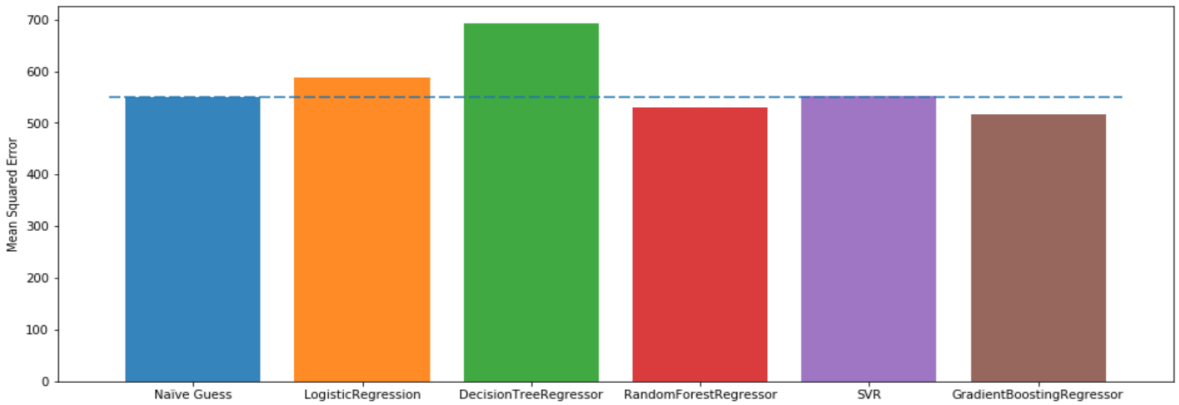
Methodology

Implementation

The output is the flight delay estimative as an integer number. Consequently, the type of model must be regression. For this reason, five regression models were considered as the candidate models:

- Logistic Regression [6];
- Decision Tree Regression [7];
- Random Forest Regression [8];
- Support Vector Regression [9];
- Gradient Boosting Regression [10].

Each of the candidate models were instantiated with the standard initial parameters from the scikit-learn Python`s module. The candidate models were trained with half of the samples and their respective MSE were computed regarding the other half of the samples for comparison with the Naïve Guess. Only the Random Forest Regressor and the Gradient Boosting Regressor performed better than the Naïve Guess, and the later performed the best.



Refinement

The best candidate model Gradient Boosting Regressor was selected for the next step. The refinement process is divided in two steps.

The first step, the Random Search, is to choose nine random combinations of hyper-parameters to evaluate if there is another combination different from the scikit-learn standards that leads to the better MSE result. The hyper-parameters range were:

Number of estimators from 1 to 1000;
- Learning rate from 0.001 to 0.9;
- Maximum depth from 2 to 10;
- Minimum samples split from 2 to 50.

The best MSE from the previous step (the nine random plus the standard from the scikit-learn) are selected to the second step, the Grid Search. In the second step the same hyper-parameters are varied each 5% over each side to evaluate again if there is another combination that leads to the better MSE result.

Results

Model Evaluation

The final model after Random Search and Grid Search steps is the Gradient Boosting Regressor with standard hyper-parameters from the scikit-learn except for the following:

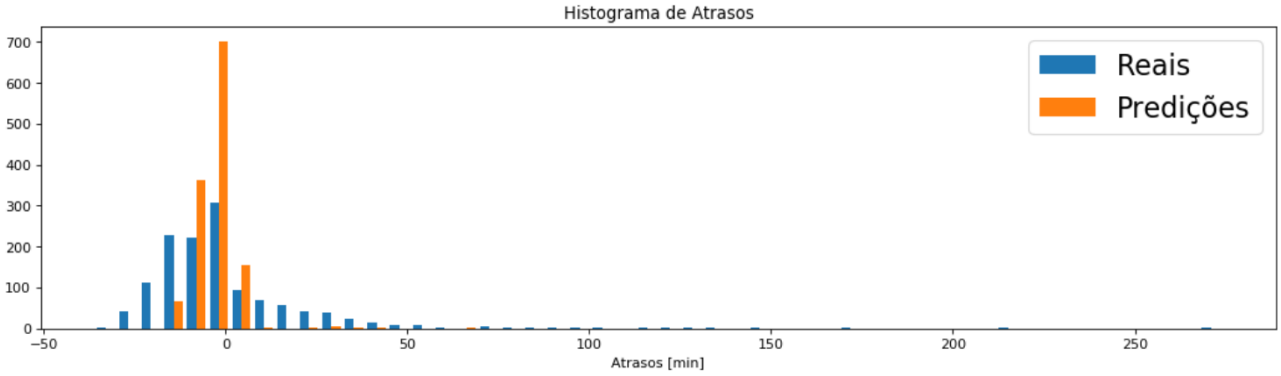
- Number of estimators 95;
- Learning rate from 0.095;
- Maximum depth 2;
- Minimum samples split 2.

Validation and Justification

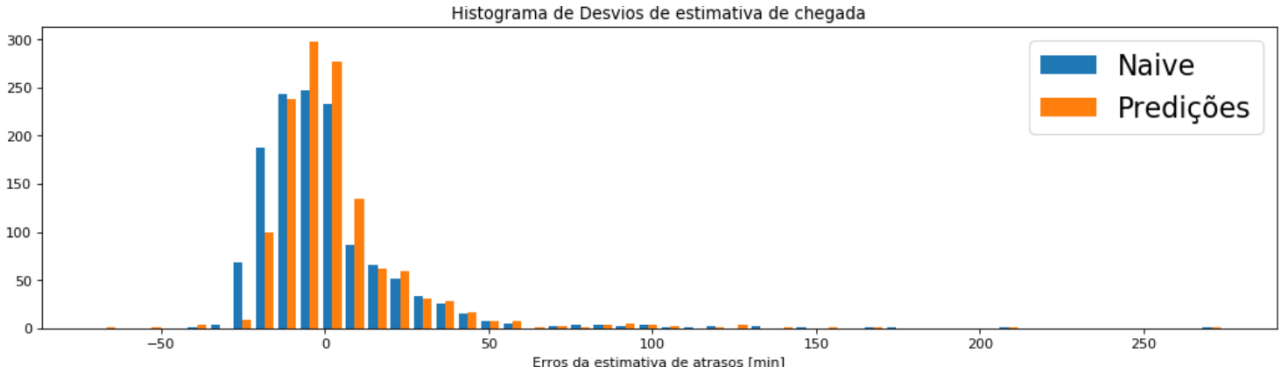
The final model performed only a little better over the evaluation metrics on the test sample than the standard guess, but much better than the Naïve Guess, as summarized in the table below. Despite the fact that the Mean Absolute Error is not used as evaluation metrics during the present methodology, it is presented as well for additional comparison to show that the final model tends to predict the flight delay with 1.5 minutes less error (- 10.5%) than the Naïve Guess.

	Naïve Guess (predicting null delay for every flight)	**Best Candidate** (standard Gradient Boosting)	**Final Model** (Gradient Boosting after changing Hyper-Parameters)
Mean Squared Error	550.8	521.2	520.1
Mean Absolute Error	14.3	12.8	12.8

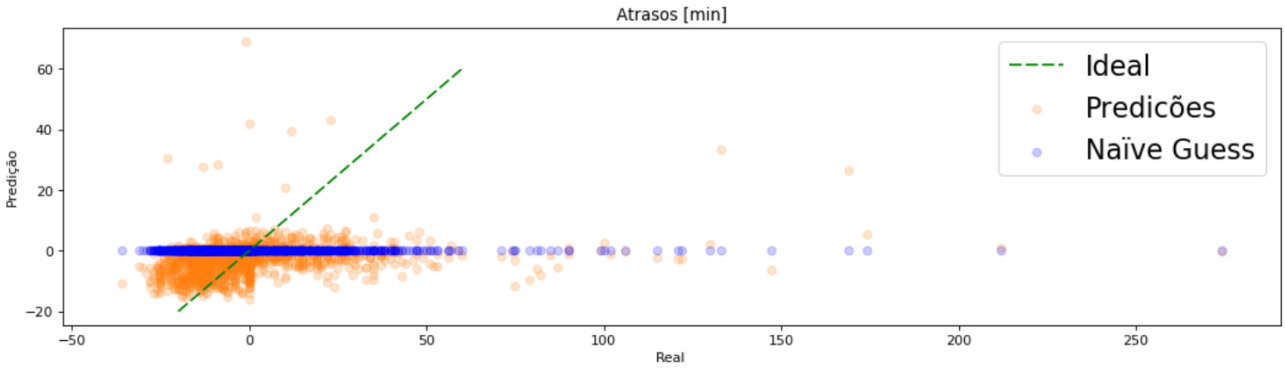
Considering that the Naïve Guess estimates all flight delays as null values, the final model histogram shows that the predictive model sometimes estimates other non-zero values as illustrated below.



Comparing the estimative errors between the Naïve Guess and the final model, the overall performance of the predictive model is good since its histogram has more zeros as the mode and its mean is closer to zero as illustrated below. On the one hand, the better contributions of the predictive model is having less mistakes about the negative delays, meaning that it can predict some of the arrivals in advance. On the other hand, some fewer mistakes are more common for the late arrivals less than 30 minutes.

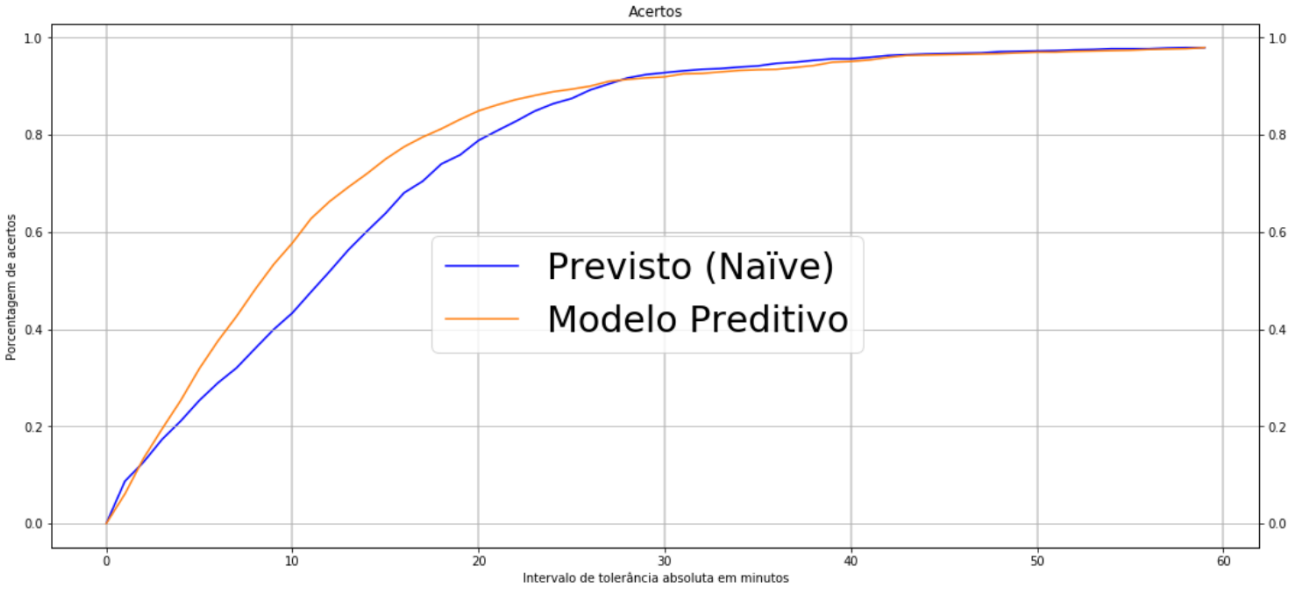


The plot comparing the real flight delays with the ideal, Naïve and model predictions shows that the model does perform a little better job to predict the arrivals in advance than the Naïve Guess. However, again little improvement can be observed for the late arrivals and there are still considerable quantities of noise in general.



Free-Form Visualization

Finally, a plot of percent of successful predictions (y axis) over the threshold of tolerance in minutes (x axis) comparing both Naïve Guess and the final model is illustrated below. It shows that the predictive model is remarkable over the Naïve when the former is intended to predict with tolerance between 5 and 25 minutes of tolerance. The larger difference is around 10 minutes when the predictive model estimates correctly around 58% of the flight delays while the Naïve Guess estimates correctly only 45%.



Conclusion

Reflection

The results presented clearly shows that the obtained predictive model has better performance than just guessing every flight delay as null value. In summary, the performance of the predictive model has absolute error mean of 12.8 minutes, which is 1.5 minutes lower (- 10.5%) than the Naïve Guess.

Improvements

The prediction of flight delay is considered accomplished since the final model performs better than the Naïve Guess over the evaluation metrics.

However, the fact that the final model did not performed much better than the best candidate means that either the final model has not been appropriately refined or that the standard parameters from the scikit-learn were already close enough to the best solution. This analysis may be further detailed considering more options of hyper-parameters for the Random Search and for the Grid Search.

In addition, more data may be employed for better predictions of the flight delays such as:

- The weather related to the arrival airport.
- The delay on other regional hubs with direct connections in the same day.

Source Code

The source code and additional documents of this project can be accessed on the link below. It includes the downloaded data, data preparation, exploration with plots, the creation of the predictive model following the present methodology and all the results described in this document.

https://github.com/diogodutra/flight-delay-prediction

References

[1] "Flight cancellation and delay," [Online]. Available: https://en.wikipedia.org/wiki/Flight_cancellation_and_delay. [Accessed 17 November 2018].

[2] ANAC. [Online]. Available: http://www.anac.gov.br/. [Accessed 17 November 2018].

[3] "METAR," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/METAR. [Accessed 17 November 2018].

[4] "Centro de Previsão de Tempo e Estudos Climáticos," INPE, [Online]. Available: http://bancodedados.cptec.inpe.br/. [Accessed 17 November 2018].

[5] "Mean squared error," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Mean_squared_error. [Accessed 18 November 2018].

[6] "Logistic Regression," Scikit-Learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. [Accessed 18 November 2018].

[7] "Decision Tree Regressor," Scikit-Learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html. [Accessed 18 November 2018].

[8] "Random Forest Regressor," Scikit-Learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. [Accessed 18 November 2018].

[9] "Epsilon-Support Vector Regression," Scikit-Learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html. [Accessed 18 November` 2018].

[10] "Gradient Booster Regressor," Scikit-Learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html. [Accessed 18 November 2018].