Capstone Project

Machine Learning Engineer Nanodegree

Diogo Dutra

November 20, 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. The prediction will certainly benefit those passengers running tight on schedule by allowing them to reorganize their tasks on advance and the air traffic control to fill the landing slots with less no-shows.

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.

There are other similar initiatives. Google Flights [2] shows on its app an estimative but "*you shouldn't take its predictions at face value*". SITA Lab [3] alleges that "*able to provide an accurate prediction of within 15 minutes of the flight arrival for around 80% of flights 6 hours before touch down*". None of these cases mentions much more than using Machine Learning techniques over historical data.

Other academic studies reveals some solid results. A publication entitled "Airline Delay Predictions using Supervised Machine Learning" [4] applied polynomial fit for long flight duration. On "Iterative machine and deep learning approach for aviation delay prediction" [5] neural networks and deep networks were applied to classify the flights into "DELAY" or "NO DELAY" resulting in accuracy up to 92%.

The methodology here also uses the supervision learning technique to collect the benefit of having the schedule and the real arrival data. The time difference in minutes is calculated and combined into a table with more departure and weather data. Initially, some specific supervision algorithms with light computing cost were considered as candidates and then the best candidate is refined for the final model.

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 [6]. The weather data is the METAR [7] type and specific to the departure airport available on the INPE's BDM website [8].

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 "Naïve 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 [9] (MSE):

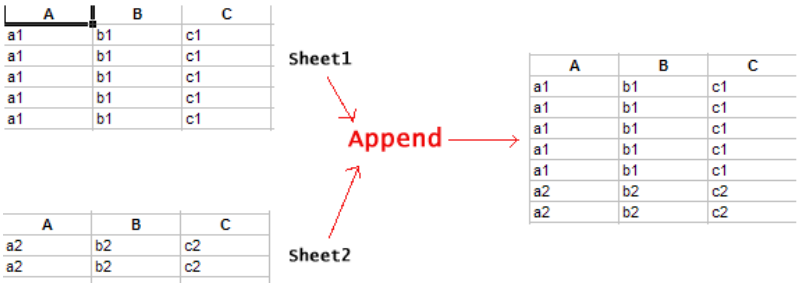
$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

The MSE is appropriate for our regression problems since it is differentiable, contributing to the stability of the algorithms. It also heavily punishes the bigger errors over smaller errors, which is the desired behavior for predicting flight delay. In other words, mistaking some predictions for a few minutes is not an issue compared to one mistake of more than 10 minutes.

Analysis

Data Preprocessing

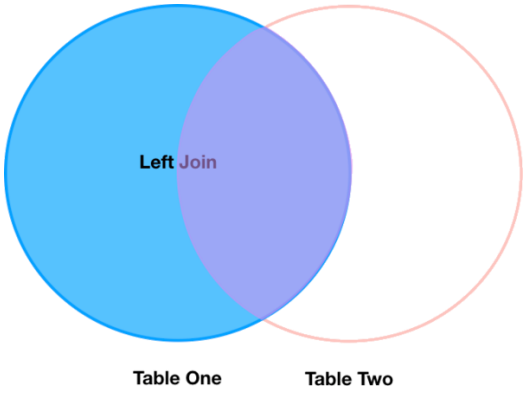
Both departure and weather data files separated by months. Therefore, they were downloaded separately, converted into tables and then appended as illustrated below.



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

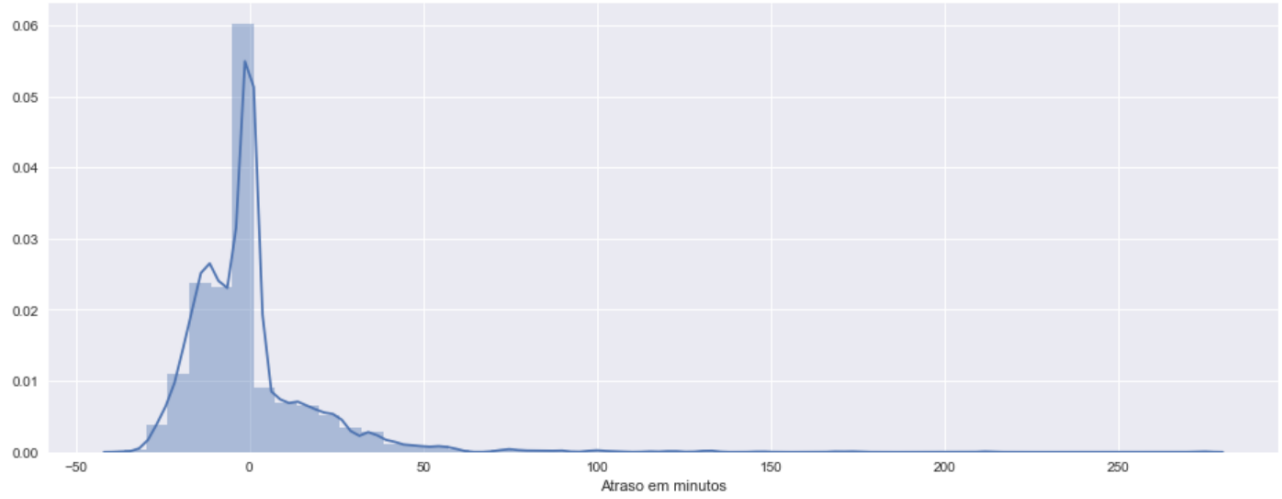
The **"Flight Delay"** column is calculated from the difference between the scheduled and the real arrivals in minutes. This column is the label to be predicted by the regression model.

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.

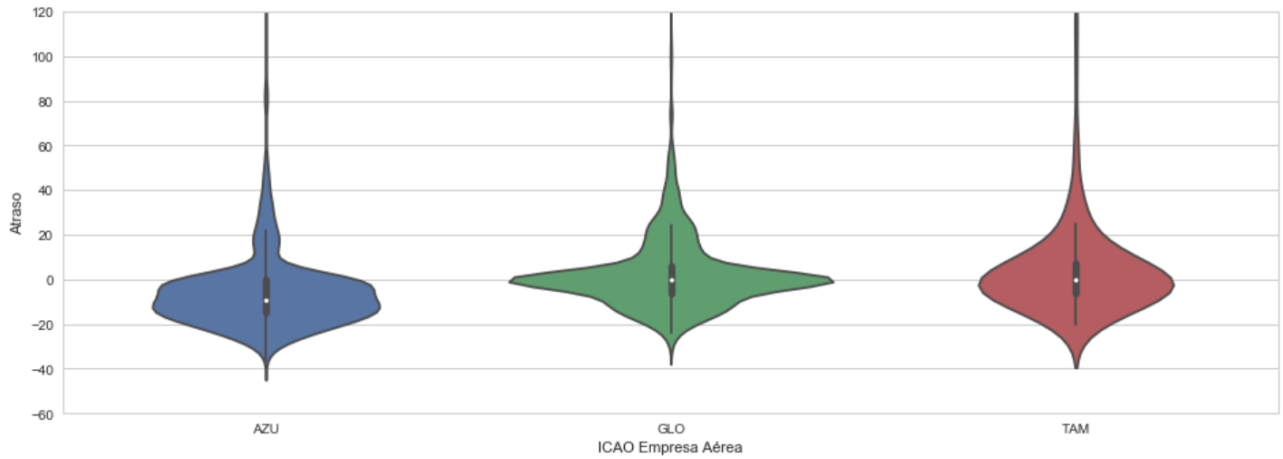
The resulted input table has 2599 samples (rows) with 17 features and 1 label (18 columns): Airliner, Scheduled Time of Departure, Flight Number, Day of the Month, Month, Day of the Week, Type of First-Layer Cloud, Wind Direction, Height of the First-Layer Cloud, Wind Speed, Coverage of the First-Layer Cloud, Air Pressure at Sea-Level, Air Temperature, Dew Temperature, Horizontal Visibility, Type of Sky and Flight Delay in minutes.

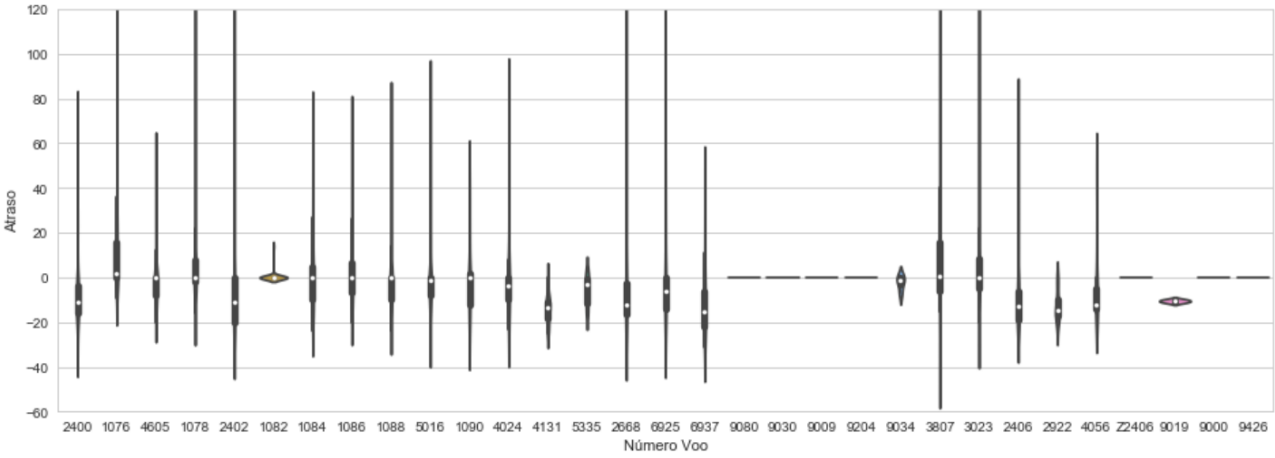
Exploratory Visualization

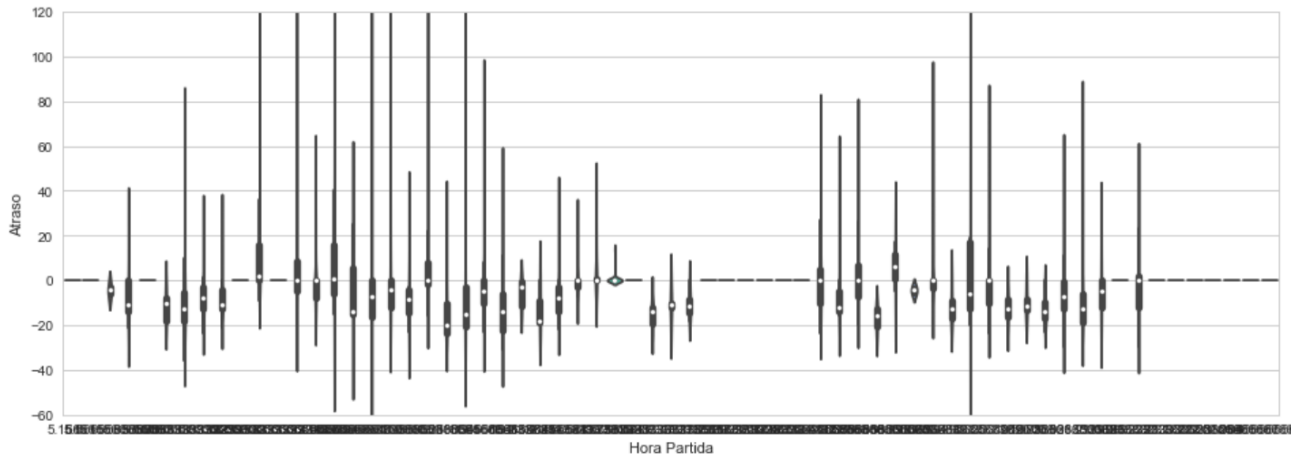
The **“Flight Delay”** label is Poisson-like distributed as illustrated below, with mean -0.2 minutes, standard deviation 19.38, minimum of -36 minutes, 25% quartile -11 minutes, 50% quartile -1 minutes, 75% quartile 1 minutes and maximum of 274 minutes.

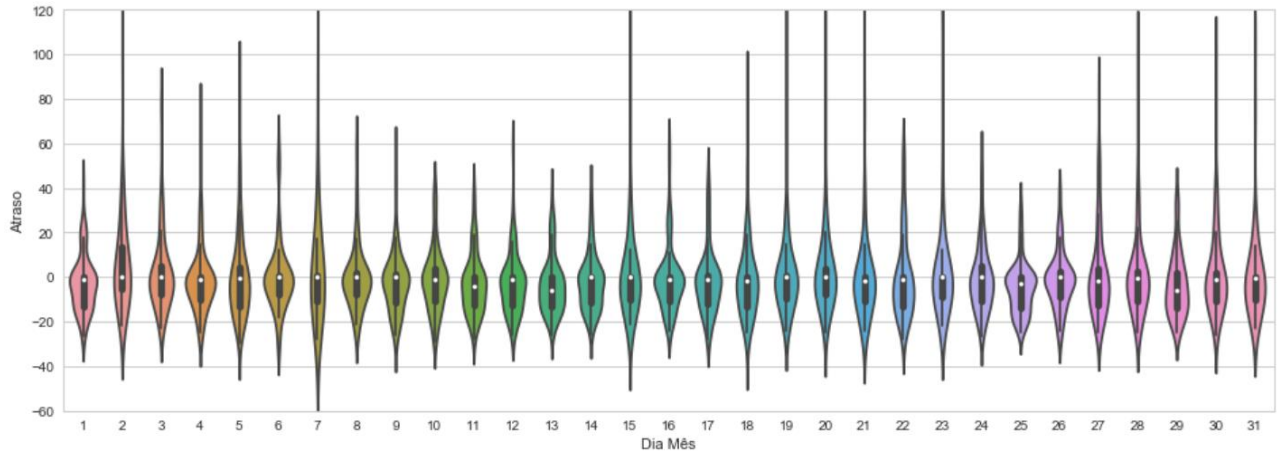


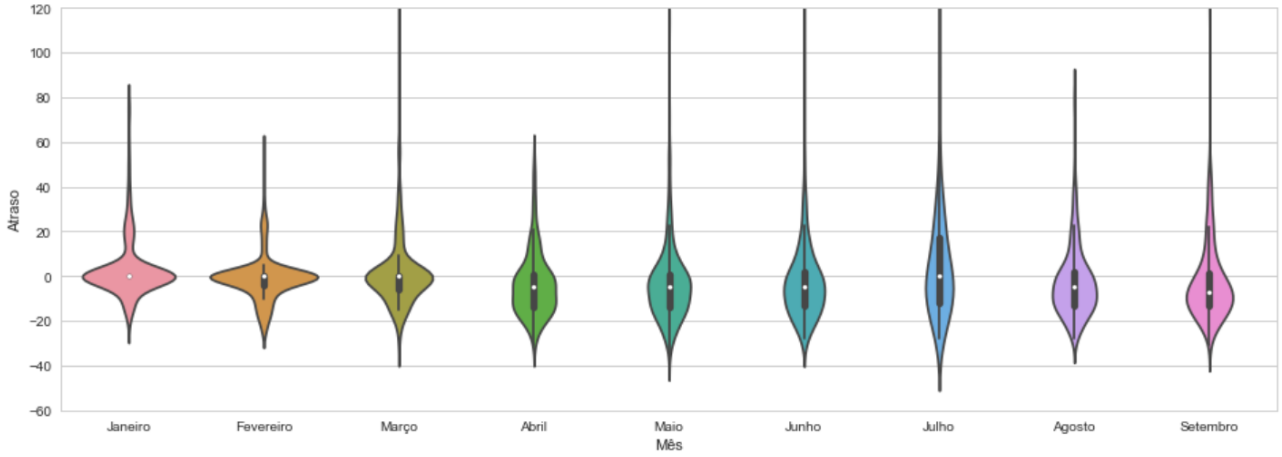
The prepared input table is used for a series of violin plots to better understand how is the data variation over its dimensions, focusing on how the delays are distributed by each feature.

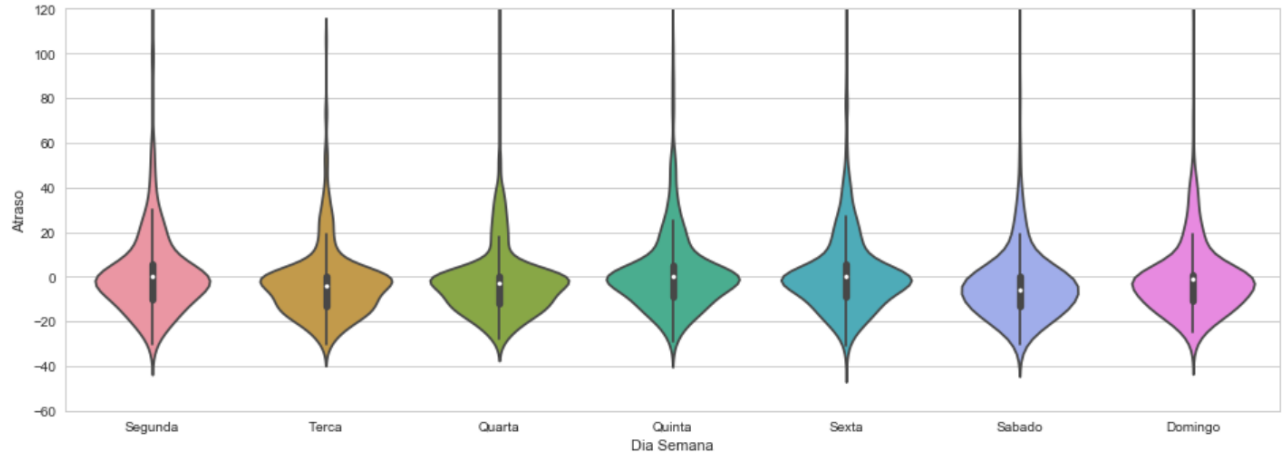


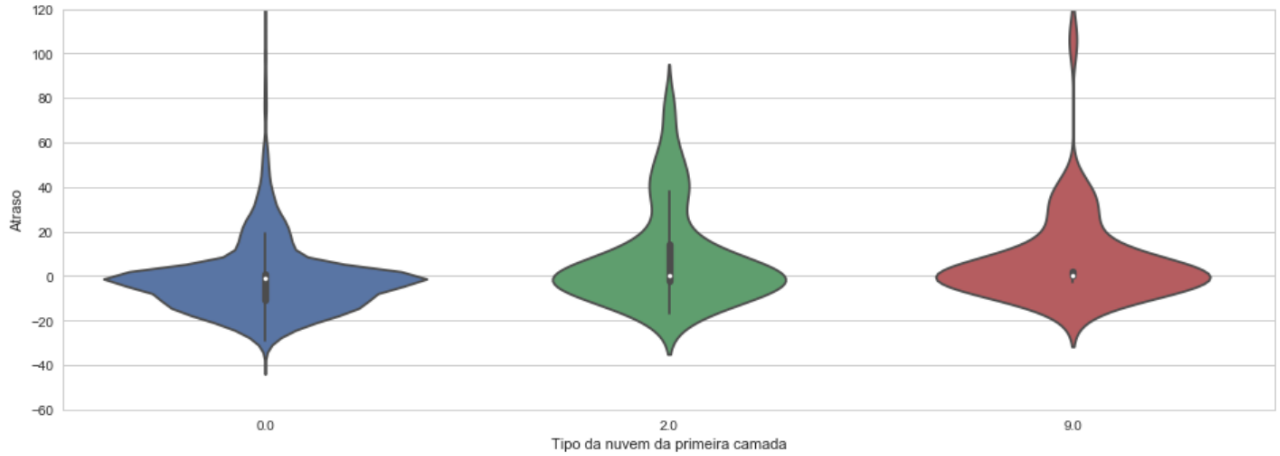


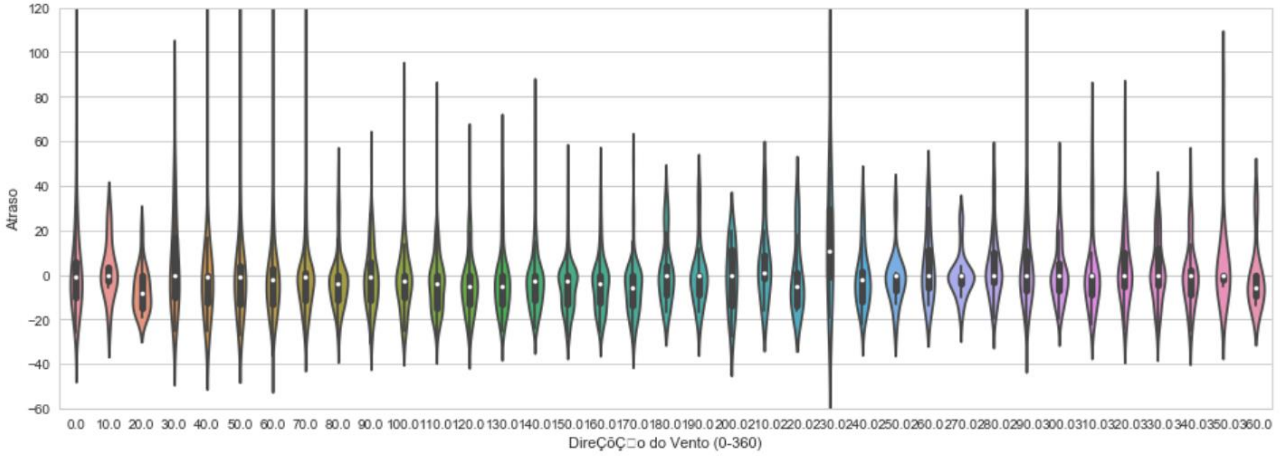

Atraso
Hora Partida

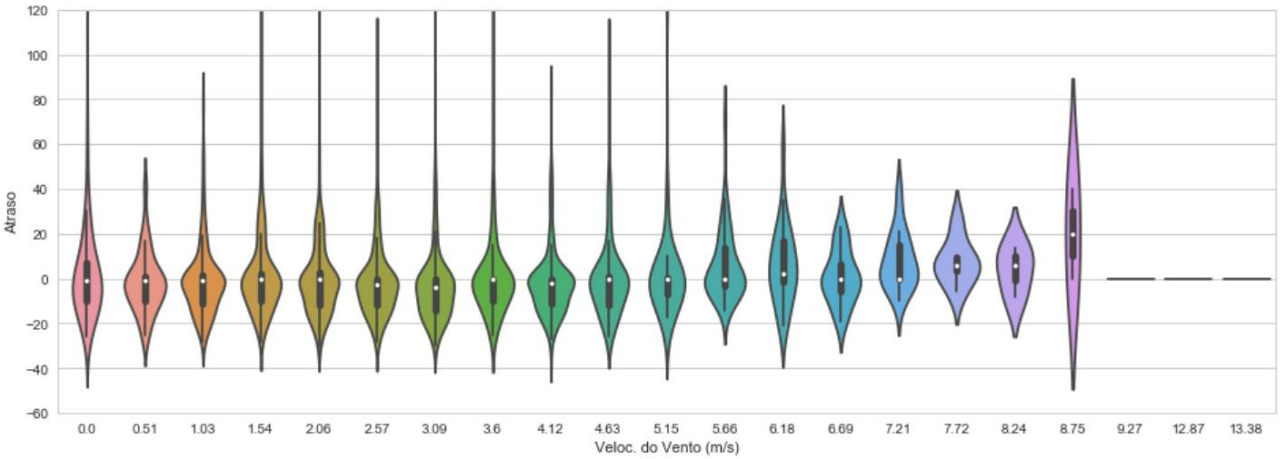

Atraso
Dia Mês

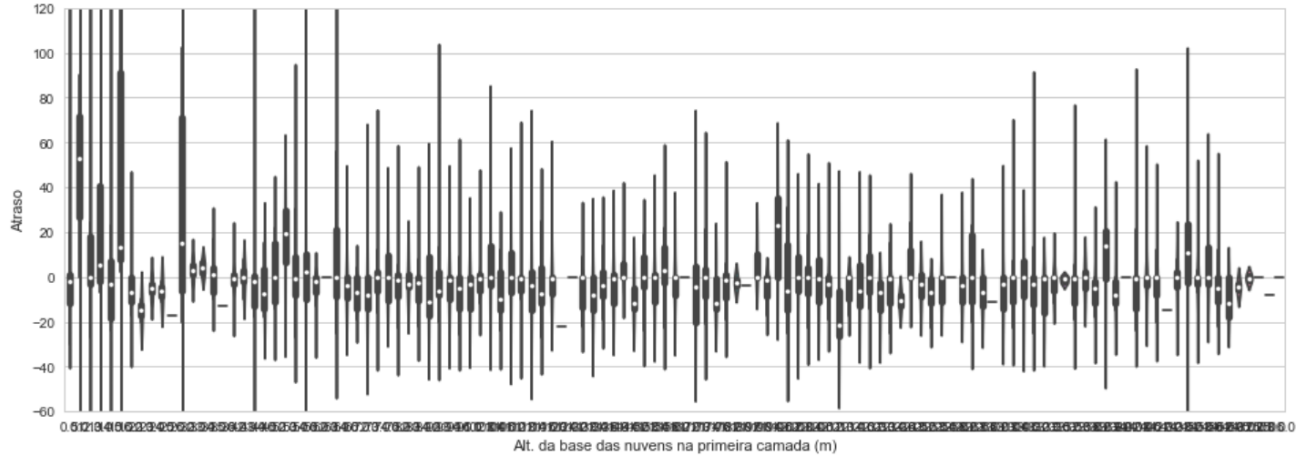

Atraso
Janeiro
Fevereiro
Março
Abril
Maio
Junho
Julho
Agosto
Setembro
Mês

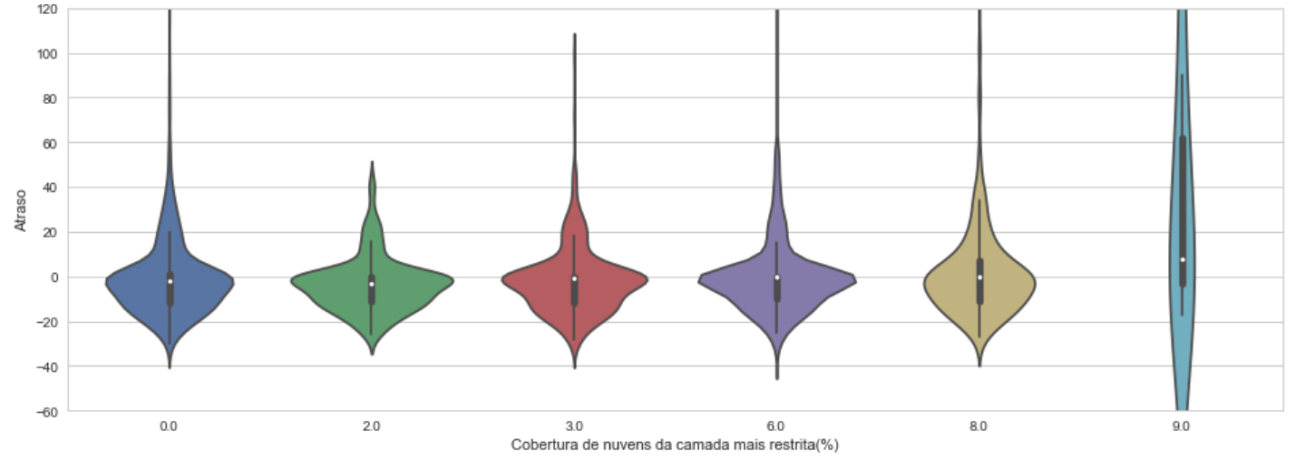

Atraso
Segunda
Terca
Quarta
Quinta
Sexta
Sabado
Domingo
Dia Semana

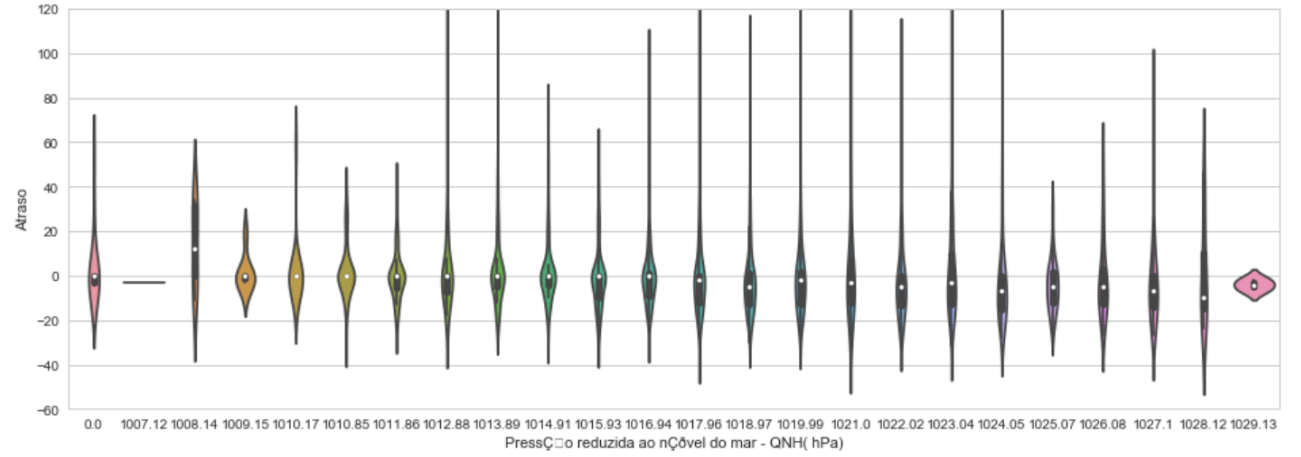

Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0
2.0
9.0
Tipo da nuvem da primeira camada

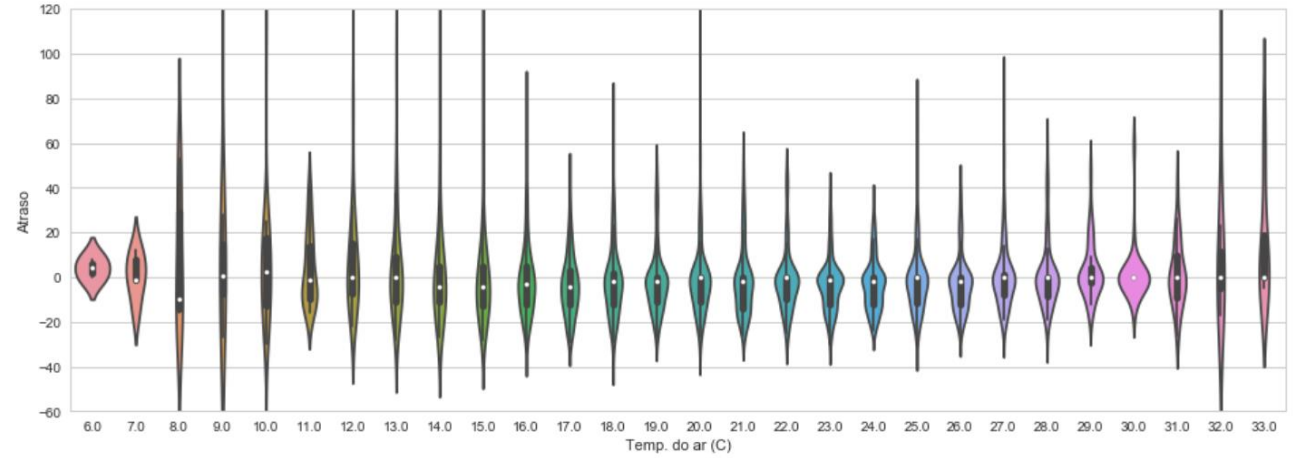

Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0 10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 100.0 110.0 120.0 130.0 140.0 150.0 160.0 170.0 180.0 190.0 200.0 210.0 220.0 230.0 240.0 250.0 260.0 270.0 280.0 290.0 300.0 310.0 320.0 330.0 340.0 350.0 360.0
DireÇõÇ□o do Vento (0-360)

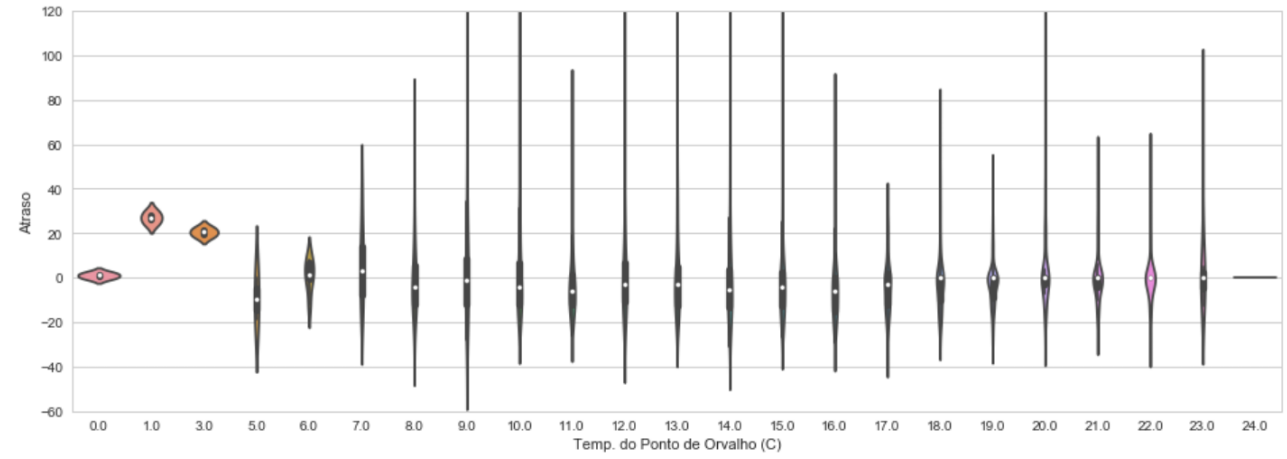

Atraso
120
100
80
60
40
20
0
−20
−40
−60
0.0 0.51 1.03 1.54 2.06 2.57 3.09 3.6 4.12 4.63 5.15 5.66 6.18 6.69 7.21 7.72 8.24 8.75 9.27 12.87 13.38
Veloc. do Vento (m/s)

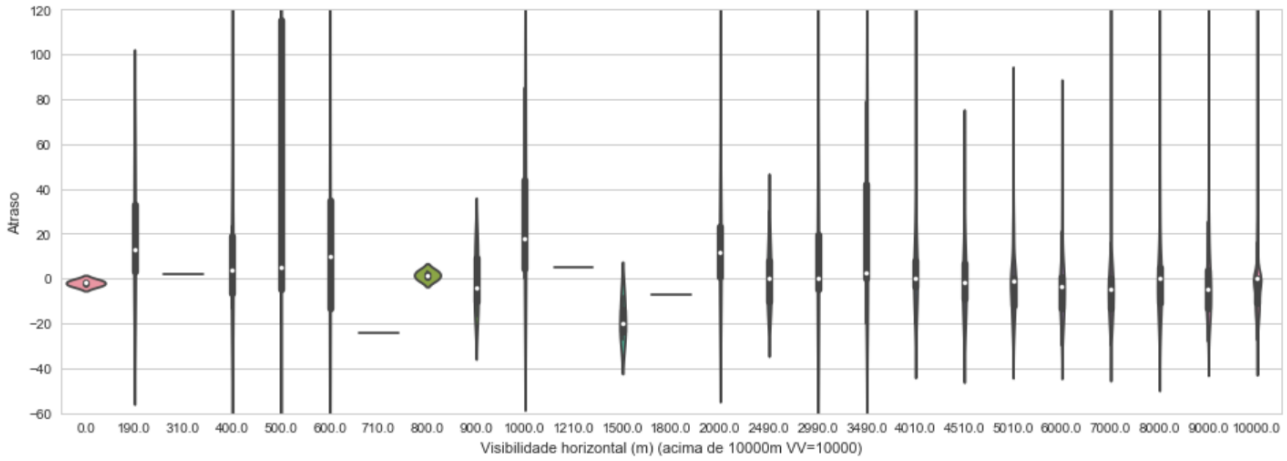

Atraso
Alt. da base das nuvens na primeira camada (m)

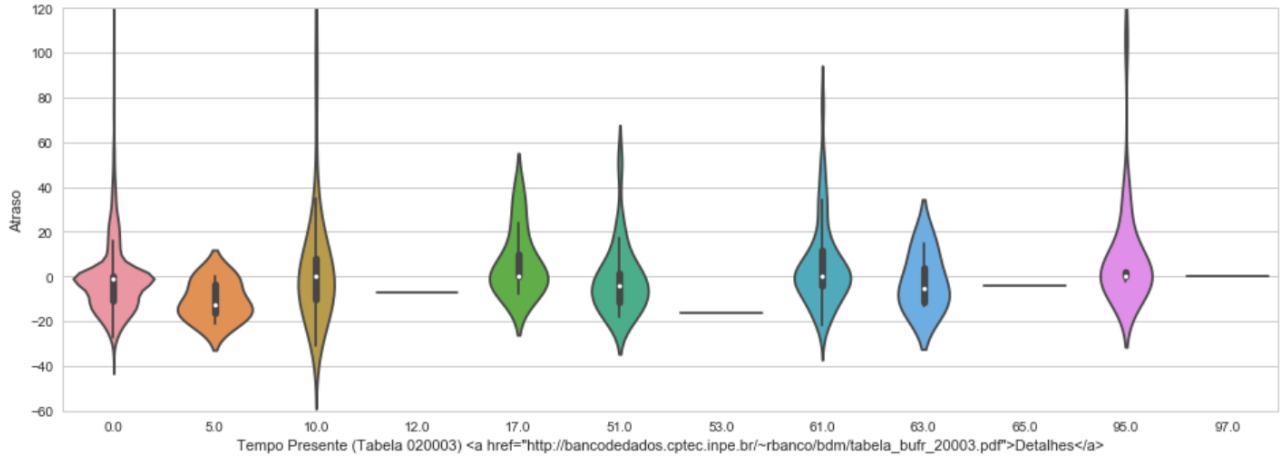

Atraso
0.0
2.0
3.0
6.0
8.0
9.0
Cobertura de nuvens da camada mais restrita(%)


Atraso
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
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)







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 20º leads to more arrivals in advance while 230º 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 little to none 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 and dew point temperature).

Algorithms and Techniques

The appropriate algorithms are the Supervised Learning Regression type because the input data is a table with the output **"Flight Delay"** label calculated in minutes.

The data visualization shows that the correlation between the features and the label is not elementary like a linear regression. Therefore, the more sophisticated techniques are more likely to perform better. However, extreme computationally heavy algorithms such as high order polynomial regressions and neural network are discarded since there is limited resources available for the present project. Additionally, the techniques shall be capable of handling both numerical and categorical features.

Considering these statements, four regression models are considered as candidates:

- **Decision Tree Regression** [10], that builds a series of questions over the attributes, similarly to a tree structure, in order to subset down to the smaller remaining samples with similar values. It is quite light to handle large amount of data and it may be a good fit for our non-linear categorical data. However, special attention is likely required to avoid overfitting and suboptimal solutions [11];
- **Random Forest Regression** [12], an additive model composed of a series of independent Decision Tree models that are trained in parallel over a smaller resampling by random the data (bagging method). It is light enough to handle thousands of samples, but our categorical features might be a challenge due to sparse data and also some tendency to overfitting the train data [13];
- **Support Vector Regression** [14], that remaps the original dimensions to another hyperplane in order to better separate the data maximizing the margin between them. To lower the burden of computational load, only the amount of samples included in the computation is only large enough until the boundary drawn inside the margin tends to stay steady despite the additional considered samples [15]. It may be an accurate model on smaller datasets, but it may be less effective for our noisy and overlapping dataset;
- **Gradient Boosting Regression** [16] is also an additive model composed of a series of Decision Tree models but, differently from Random Forest, the models are all weak classifiers (ie: low-depth trees) and dependent to each other since the next classifier is trained in series and added in order to improve the already trained ensemble (boosting method) [17]. Usually it performs better than Random Forest, however it is more sensitive to computational load and overfitting.

Methodology

Implementation

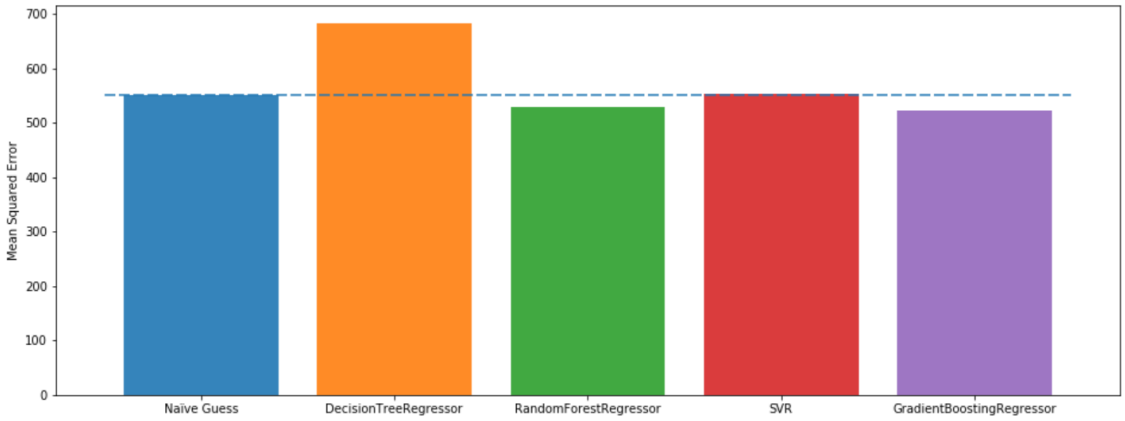
The complete source code is available for public access on the link below. For this reason, instead of explaining the methodology on every detail, it is described only the overall steps, values and results.

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

After applying hot-encoding of the categorical features, the input table was enlarged from 18 to 147 columns, leading to a sparse dataset.

Then, the dataset is split into 1300 samples for train and 1299 samples for test. The split criteria is the chronological order as the oldest for train and the most recent for the test.

Each of the candidate models previously listed were instantiated with the standard initial parameters from the scikit-learn Python`s module. The candidate models were trained with a random 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 as illustrated below.



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:

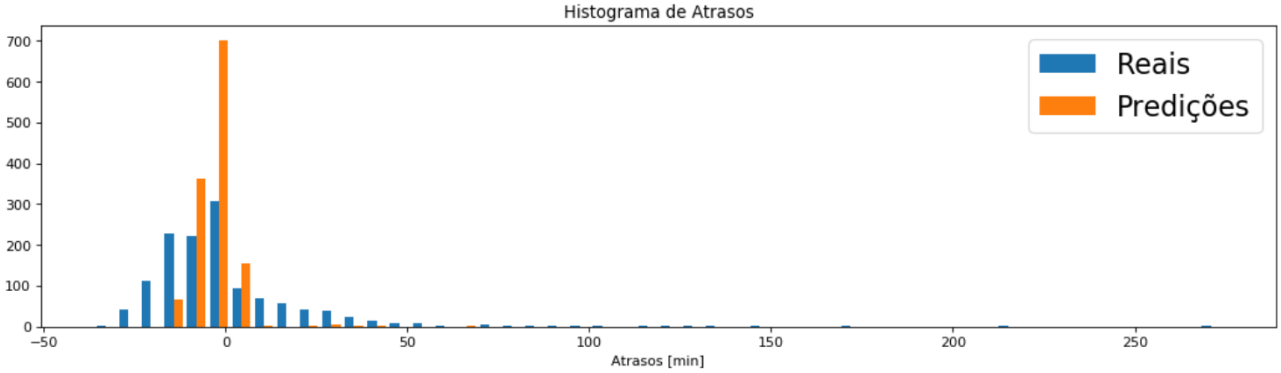
- Number of estimators 95;
- Learning rate from 0.095;
- Maximum depth 2;
- Minimum samples split 2.

Validation and Justification

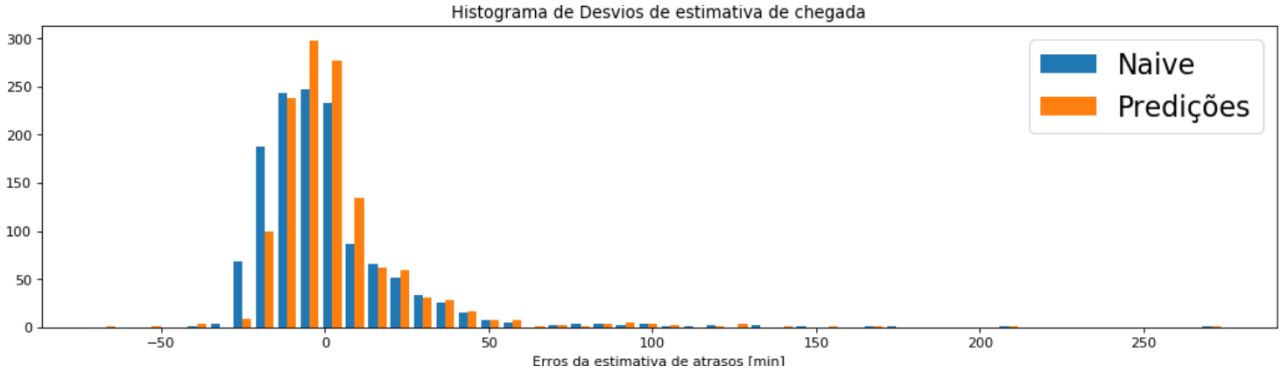
The final model performed only a little better over the evaluation metrics on the test sample than the standard guess, but much better than the Naïve Guess, as summarized in the table below. Despite the fact that the Mean Absolute Error is not used as evaluation metrics during the present methodology, 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

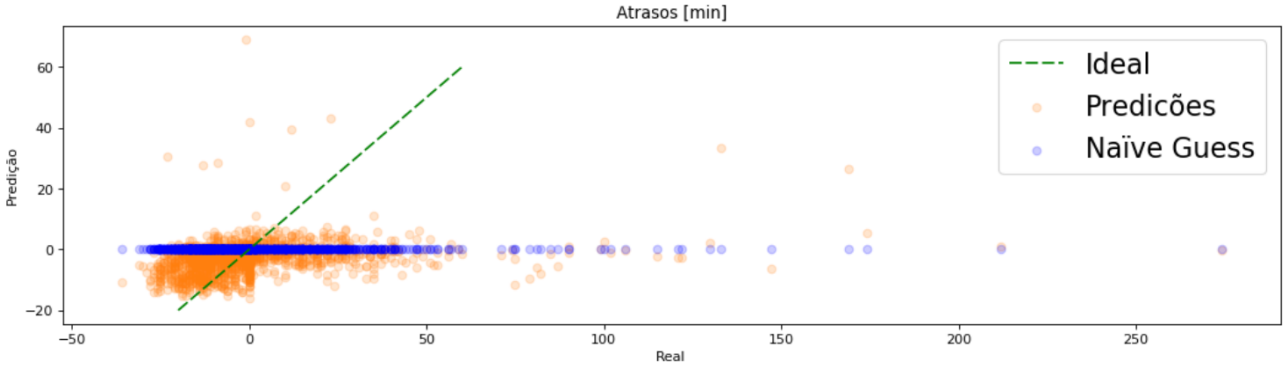
Considering that the Naïve Guess estimates all flight delays as null values, the final model histogram shows that the predictive model sometimes estimates other non-zero values as illustrated below.



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.

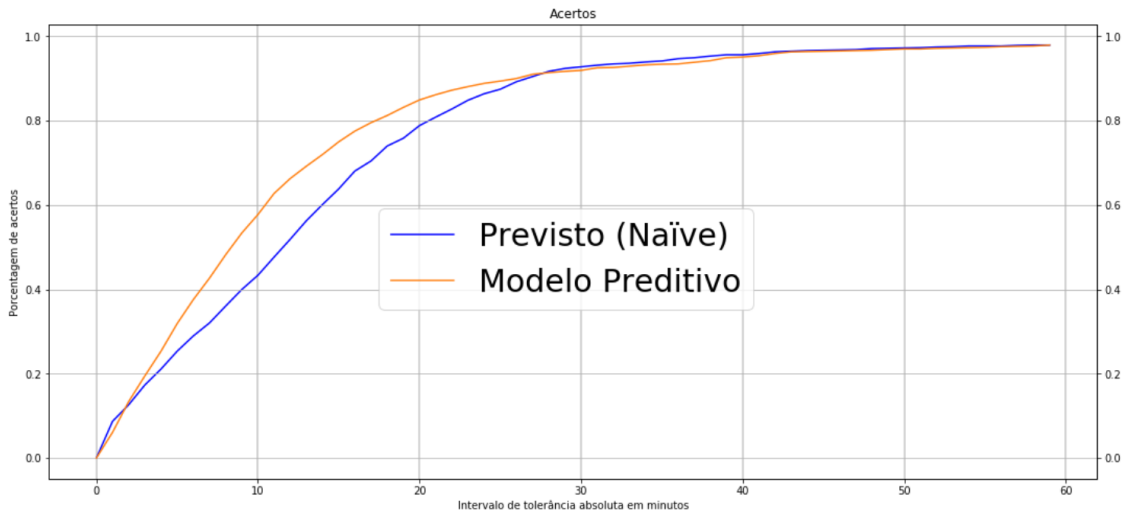


The plot comparing the real flight delays with the ideal, Naïve and model predictions shows that the model does perform a little better job to predict the arrivals in advance than the Naïve Guess. However, again little improvement can be observed for the late arrivals and there are still considerable quantities of noise in general. These rare cases are considered failures for the predictive model since it is very important to predict the long delay cases.



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%.



This result is directly comparable to the SITA Lab study [3] that is "able to provide an accurate prediction of within 15 minutes of the flight arrival for around 80% of flights 6 hours before touch down". The present results is capable of predicting within 15 minutes around 75% of flights before the take-off; or 80% of flights within 17 minutes.

Conclusion

Results

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.

Reflections

The scheduled departure and weather 2018 historical data specific for the one-way São Paulo to Rio de Janeiro trip has been downloaded and combined to create the input table with numerical and categorical features. The Flight Delay label in minutes has been calculated from the scheduled and recorded arrivals. The data has been explored visually, leading to the realization that the data is non-linearly distributed. Four different Supervised Learning Regression algorithms were considered as candidates and compared by Mean Squared Error. The best candidate, Gradient Boosting, has been refined by a series of hyper-parameters experimentations resulting in the final model. It was concluded that the final model performs better than the Naïve Guess by correctly estimating the flight delay of 75% of flight within 15 minutes of error tolerance.

The prediction of flight delay is considered accomplished since the final model performs better than the Naïve Guess over the evaluation metrics and similar to other public literature [3].

However, the correlation between the features and the label showed to be weak during the data exploration due to the high-unexplained variance of the data observed by the Poisson-like long tail. This high noise is the main reason for a poor performance regarding the long flight delays and it is likely caused by the inaccessible data mentioned before, such as disasters, maintenance and security issues [1].

Improvements

The fact that the final model did not performed much better than the best candidate means either that 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.

Longer flight delays may have extra attention since they are extremely important for a predictive model. There are few cases of this kind but none of them were correctly predicted. For this reason, a further analysis could emphasize these data to look for exceptional relationships with the current features to search for extra insights.

In addition, more data may be gathered 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;
- Real-time reports such as technical or security issues;
- Integration with online news in order to consider exceptional events such as natural disasters;
- In-flight reports from other resuming flights to the air traffic control, but with the limitation of having these additional data much closer to the arrival event.

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] "Google is now using machine learning to predict flight delays," The Verge, [Online]. Available: https://www.theverge.com/2018/1/31/16955580/google-flights-app-delays-machine-learning-economy. [Accessed 19 November 2018].

[3] "Using Artificial Intelligence to predict flight delays," IT Review, [Online]. Available: https://www.sita.aero/air-transport-it-review/articles/using-artificial-intelligence-to-predict-flight-delays. [Accessed 19 November 2018].

[4] P. Chandraa, P. N and K. R, "Airline Delay Predictions using Supervised Machine Learning," *International Journal of Pure and Applied Mathematics*, 2018.

[5] V. Venkatesh, A. Arya, P. Agarwal, S. Balana and S. Lakshmi, "Iterative machine and deep learning approach for aviation delay prediction," *2017 4th IEEE Uttar Pradesh Section International*, October 2017.

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

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

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

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

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

[11] "Decision tree learning," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree_learning. [Accessed 19 November 2018].

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

[13] "Random forest," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Random_forest. [Accessed 19 November 2018].

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

[15] "Support vector machine," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine#Regression. [Accessed 19 November 2018].

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

[17] "Gradient boosting," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Gradient_boosting. [Accessed 19 November 2018].

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