Machine Learning Engineer Nanodegree

Capstone Proposal

Michael Moreira Cabral
June 12th, 2018

Proposal

Supervised learning algorithm to classify aircraft turnarounds into different categories of towing behavior.

Domain Background

In the domain of airport operations, it is common to apply stochastic simulation and multi-criteria optimization to perform tasks such as evaluating passenger and baggage flows and aircraft stand allocation.

The proposed study aims to develop a method to enhance the accuracy of an aircraft stand allocation multi-criteria optimization algorithm, which in turn aims to mimic the allocation performed by human allocators from the airport operations team.

With a well calibrated stand allocation model, it is possible to evaluate the impact of new operations procedures and/or infrastructure expansion. For example, if one has a calibrated model, that correctly mimics the ability to allocate passengers in contact gates, it can be measured the impact, in terms of % of passengers processed in boarding bridges, of the construction of new bridges.

ANAC (Agência Nacional de Aviação Civil), Brazilian civil aviation authority, requires that a minimum of 95% of international passengers should be processed in boarding bridges. This is an important performance indicator, which the airport operator should carefully manage.

There are plenty of academic research regarding stand allocation algorithms, such as:

- The over-constrained airport gate assignment problem. H. Ding, A. Lim, B. Rodrigues, Y. Zhu. 2005.
- The use of meta-heuristics for airport gate assignment. Chun-Hung Cheng, Sin. C. Ho, Cheuk-Lam Kwan. 2012.
- Incorporation of recoverable robustness and a revenue framework into tactical stand allocation. Bert Dijk. 2016.

Using these algorithms, it is possible to create and implement rules that mimic the ones that are used by the human allocators from the airport operations team. Doing this, is expected that the final allocation performed by the algorithm should be similar to the one performed during the live operation. There are many groups of rules to be considered, such as:

- Geometric rules (max aircraft size per stand; adjacencies restrictions);
- Type of flight (the available contact gates depends on whether the flight is domestic or international);
- Business rules (to consider the preferable allocations of the airlines)
- Efficiency rules (to emulate the expertise of the human allocators)
- Boundary rules (as the simulator considers just one day, boundary rules are necessary)

The input to a stand allocation algorithm is a flight schedule, with the flight number, airline, arrival/departure times, aircraft type, aircraft registration, flight type and number of passengers. With the times of arrival and departure from each aircraft in the airport, it is possible to imagine “bars” (aircraft turnarounds) into the available lines (aircraft stands). The output of the algorithm can then be visualized with a Gantt chart.

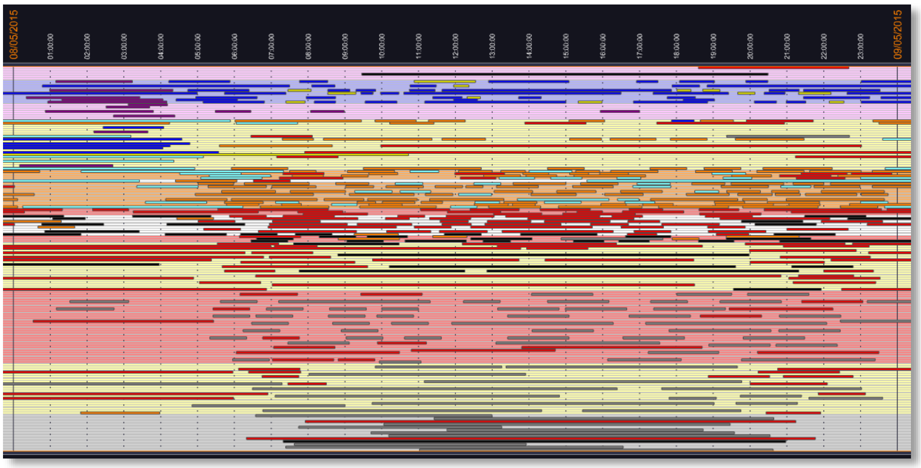

Figure 1 - Stand allocation Gantt chart

However, if you have an aircraft that arrives at 9AM and departs at 9PM it doesn’t necessarily means one will need to allocate a 12h long bar. Probably this turnaround will be broken in to pieces, namely, it will be broken into an arrival movement (first bar), a long stay movement, probably in a remote stand (second bar) and a departure movement (third bar). In order to allocate passengers in boarding bridges, one just need to take into account the first and third bars. These will be the ones to be allocated in the contact gates (more expensive infrastructure). The long stay part, that does not involve passengers processing, can be allocated in a remote stand (cheaper infrastructure).

Therefore, before applying the stand allocation algorithm, it is necessary to perform a preprocessing step, which is the break of the aircraft turnarounds with towings.

The problem with the stand allocation academic research is that they generally take this important preprocessing step for granted. Experience shows, though, that using rules of thumb like “break any movement with more than X hours of total turnaround time” will not be accurate and will lead to final results that are not close to what was done in live operation.

I personally faced this challenge when I was responsible to develop a simulation model, based on a stand allocation model, to estimate the impact of different operational procedures and infrastructure expansion in terms of % of passenngers processed by boarding bridges.

Problem Statement

In order to successfully develop a stand allocation model, that can support many management decisions, it is crucial to also develop a preprocessing step, which is the break of the aircraft turnaround with towings.

Rule of thumb rules usually provided by the allocation personnel are not accurate. That is why a supervised learning classification model might be useful.

There are three categories of turnaround towing behavior:

Category	Label	Description
1	No towings	An aircraft arrives, is allocated in stand A and departs from the same stand.
2	Single tow	An aircraft arrives, is allocated in stand A, is towed to stand B and departs from this one.
3	Multiple tows	An aircraft arrives, is allocated in stand A, is towed to stand B, is towed again to stand C and departs from this one.

The classification model should be able to analyze turnaround input features and correctly predict which type of towing behavior it will present in live operation.

Datasets and Inputs

In order to perform this classification task, a dataset with 10.677 samples will be used. This dataset contains seven features and one label (*tow_type*).

Each sample of this dataset is a complete turnaround, namely, it contains data from the whole operation of an aircraft arriving and departing from the airport. The seven features are:

Feature	Description
acft_type	The ICAO aircraft type code (e.g. A320, E190, B737).
acft_cat	The ICAO aircraft category code (A, B, C, D, E, F).
in_block_hour	The hour when the aircraft in-block occurred (0 to 23).
off_block_hour	The hour when the aircraft of-block occurred (0 to 23).
total_time	Time span, in minutes, between off-block and in-block times).
turnaround_type	The first letter represents the arrival flight type and the second letter represents the departure one (D = domestic and I = international).
turnaround_qualifier	The first letter represents the arrival flight qualifier and the second letter represents the departure one (e.g. J = normal passenger flight; G = extra passenger flight; F = cargo flight).

The label is the column *tow_type*, which can assume the values 1, 2 and 3, as described in the previous section.

Based on my domain knowledge, these features are the best available candidates to determine which kind of towing behavior a turnaround will present. GRU Airport allowed the use of this dataset for the present study.

Solution Statement

A supervised learning algorithm will be developed, in order to predict which kind of towing behavior a turnaround will present, based on its features.

A training data set, labeled data from live operations and containing a set of turnaround features will be used to train three different learning algorithms: DecisionTreeClassifier, SVC and AdaBoostClassifier. The model with the best performance, in terms of F1 score, will be chosen and further tuned.

The final model will be able to predict the towing behavior based on turnaround features. The predicted classes will then be imported to the simulation software, allowing the development of more accurate stand allocation models.

Benchmark Model

The benchmark model is the one I developed at GRU Airport in 2015. Airport simulation softwares and techniques evolved rapidly in the aviation sector. Experts from Paragon and ARC – Airport Research Center, reviewed our stand analytics model (supervised learning for towing classification + multi-criteria optimization for stand allocation).

The supervised learning task was performed by a simple decision tree algorithm, which provides rules that could be inserted in the simulation software.

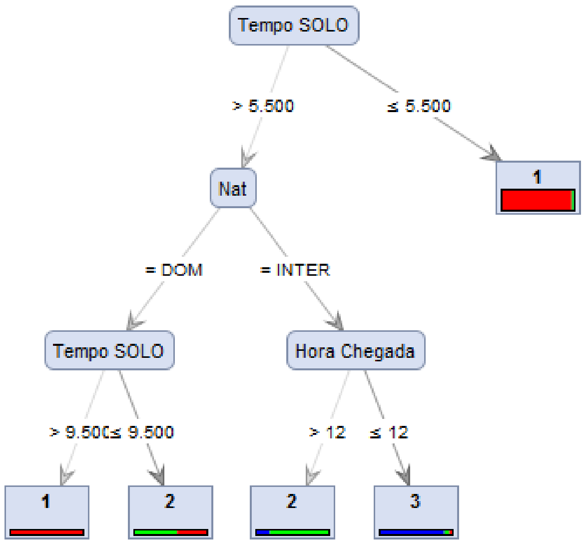

Figure 2 - GRU Airport towing classification model (legacy)

At that moment, I did not know much about machine learning best practices. Today I have better ideas that for sure will enable more creative and efficient solutions. The new results can be compared to the legacy ones with the F1 score.

Evaluation Metrics

The optimal model will be the one with the highest F1 score. As this is a multiple-label classification problem, the final score will be the weighted average of the FI score of each class.

The F1 score for each class can be calculated as:

$$F1 = \frac{(precision \times recall)}{(precision + recall)}$$

Project Design

In order to develop a solution to the presented problem, the following theoretical workflow is going to be executed:

Set up

Import the main libraries necessary for this project (numpy, pandas, seaborn). The sklearn modules will be imported later, when necessary.

Import the *turnaround_data.csv* file into a Pandas DataFrame and then edit the data types.

Data Exploration

Use *DataFrame.describe()* to both numerical and non-numerical features in order to better understand their distribution. Also use *pandas.plotting.scatter_matrix()* to visually analyze the distributions.

Use *seaborn.heatmap* and *DataFrame.corr*() to better understand the correlation between the features.

Feature Selection

Based on the data exploration and on the domain knowledge, try to further narrow the feature space in order to reduce the complexity of the model.

Data Preprocessing

Use *pandas.plotting.boxplot()* to visually look for the presence of outliers. Remove them if it seems a good choice.

Use *pandas.get_dummies* to create dummy variables from the remaining categorical features.

Training and Testing Data Split

Use *sklearn.cross_validation.train_test_split* to shuffle and split the data into training and testing subsets (stratified in the label values). The testing set will have 30% of the total samples.

Training and Evaluating Models

Three sklearn classification models are going to be used: DecisionTreeClassifier, SVC and AdaBoostClasisfier.

The training dataset will be sampled to obtain smaller training set sizes. The three models will be trained with the different training set sizes and the learning curves with the F1 scores in the training and testing sets will be plotted and analyzed.

Choosing the Best Model & Model Tuning

Based on the learning curves and on the maximum scores, the best model will be chosen.

Use the *sklearn.grid_search.GridSearchCV* to tune the chosen model based on its main parameters.