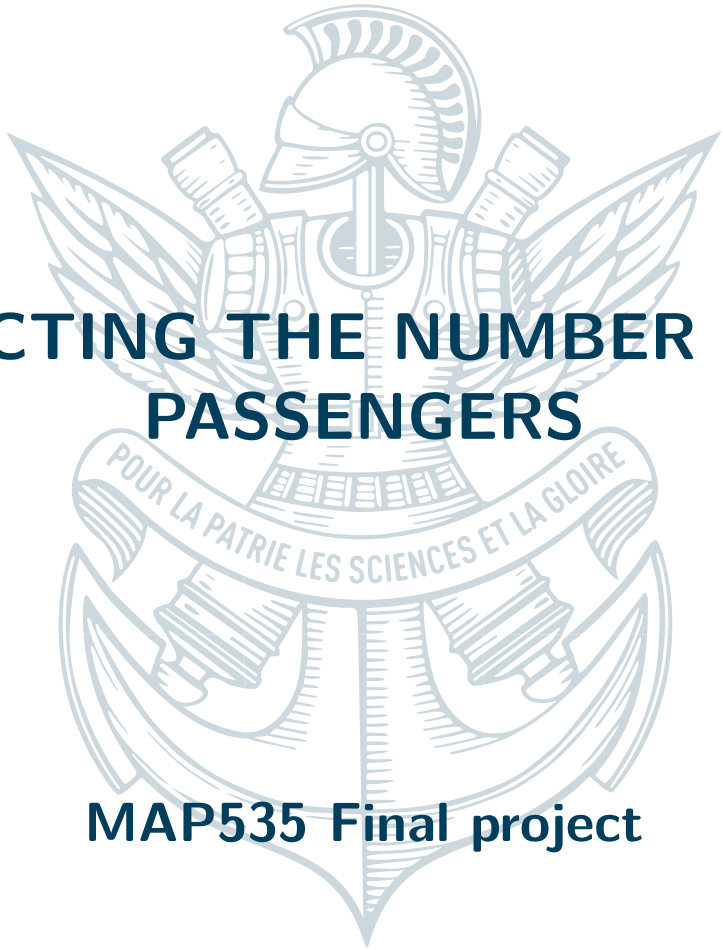

PREDICTING THE NUMBER OF AIR PASSENGERS

MAP535 Final project

December 4, 2020

Nicolas Barbier de la Serre, Martha Braun, Henrique Brito Leao

1 INTRODUCTION

We are working for an airline company who wants to optimize their plane allocation for some of their domestic routes. Part of the problem is to correctly predict the number of passengers who will fly from a given departure to a given destination on a particular day. Within the context of the data we are given, this problem corresponds to predicting log_PAX, a number that is related to the number of passengers on a given flight. Our performance measure is the RMSE of our prediction.

In this report we will describe the process of building a predictor, going through the steps of data collection, feature engineering, feature selection, preprocessing, model selection, and hyperparameter tuning.

2 DATA

2.1 Initial Data

The data we were initially provided with contained information on departure and destination airports, date, weeks to departure, standard deviation of weeks to departure, and the target variable log_PAX.

2.2 Adding Data

After doing research on the factors that influence air passenger traffic, we decided to add new data from four categories: (i) time related data to reflect seasonal periodicity and predictable outliers (e.g.: holiday seasons), (ii) geographic data to capture the distribution of population, (iii) economic data to account for the income of customers as well as relevant price indicators, and (iv) demographic data to define different customer segments. The four categories all contain factors that we believe have an impact on the demand for air transport. They should, thus, help in the prediction of the number of passengers on a specific flight.

Data	**Description**
fuel prices	price of fuel in dollars per day
flight data	the number of passengers per airport per month for all US airways
gdp data	All industries and Finance, insurance, real estate, rental, and leasing GDP
us holidays	dates of US public holidays for each year
working age population	working age group population (15-64 yo) for each metropolitan area
population by sex	population divided by gender for each state

2.3 Feature Engineering

After analysing our data we were able to combine attributes to create more meaningful variables:

- *import_day*: transforms the column *n_days* by applying a periodic function that gives higher weights to important days based on historical data;
- *import_month*: the number of passengers who flew on that month on all routes divided by the total number of passengers in the year. This captures seasonality effects;
- *passengers/departure*: passengers per month divided by total departures in the same month. This gives an estimate of the average passengers per flight;
- *load*: divided number of passengers by the number of seats for a given route and a given month;
- *cancel_rate*: departures performed divided by departures scheduled per month for a given route.

2.4 Feature Selection

We employed a feature importance approach to perform feature selection. For each of our models, we trained an initial version with all features and recovered the feature importances (variance explained, in the case of ensemble trees, and impact of permutation, in case of neural networks). We proceeded to delete the features which had less significance and iterated the process. We also checked at each step the impact of removing features on the CV score and sometimes re-inserted features.

One problem with this approach is that feature importance can be impacted by high correlation. To account for that, before training the first model, we plotted the correlation plot of the original features and dropped columns which were highly correlated to others (see Figure 1 on next page).

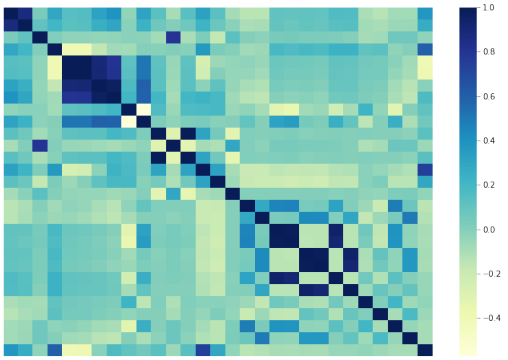

Fig. 1: Feature correlation before feature selection

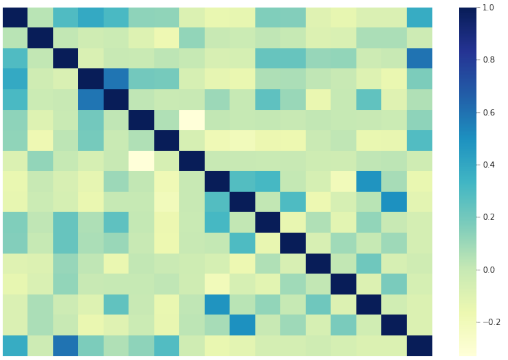

Fig. 2: Feature correlation after feature selection

2.5 PREPROCESSING

- **Numerical Data**: we scaled the quantitative features with `StandardScaler()`;
- **Categorical Data**: we encoded the data using `OneHotEncoder()`. We considered using an `OrdinalEncoder()` first to decrease the computational cost. However, its performance was not satisfactory, as machine learning algorithms assume that two nearby values are more similar than two distant ones (which was not necessarily the case).

3 MODEL SELECTION

Looking at the data, it was apparent that the target value is not linearly correlated to our variables. Based on that we considered two types of models: ensemble trees and neural networks, which are well know for dealing well with non-linearity. Given the high-dimensionality we started with GradientBoosting. As it gave promising results the logical step was to test the model's optimized versions: HistGradientBoosting, XGBoost, and CatBoost. Finally, to leverage on the different approaches of the model types we opted for using a VotingRegressor on CatBoost and a neural network. The results are summarized in the following table:

We maintained Train RMSE score above 0.1 to prevent overfitting. This ensured that our algorithms keep a good predicting power and are able to generalize well to different data.

4 HYPERPARAMETER TUNING

We started by manually exploring different hyperparameter values to get an understanding of the behaviour of the model. Conceptually, we looked for trees with large numbers of estimators, which improves the accuracy of the prediction, but penalized on training time. We also

Model	Train RMSE	CV RMSE
GradientBoosting	0.03	0.365
HistGradientBoosting	0.18	0.349
XGBoost	0.12	0.355
CatBoost	0.11	0.329
Neural Network	0.12	0.316
VotingRegressor	0.10	0.292

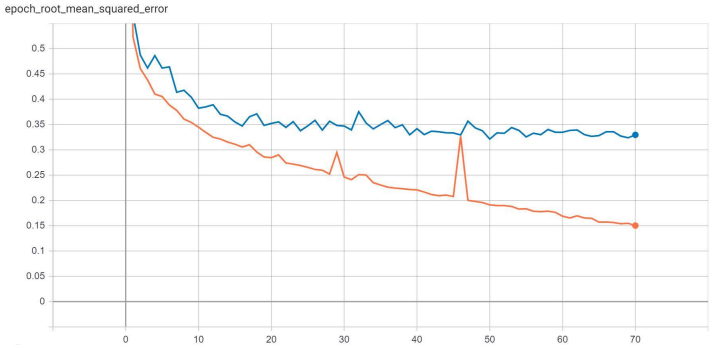

Fig. 3: Performance of Neural Network model on train (orange) and test set (blue) (RMSE by epoch)

employed large degrees of regularization (l2 norm), to avoid overfitting as the data included many features. Lastly, we avoided deep trees, again to avoid overfitting.

Once we defined the approximate ranges, we used a mix of `RandomSearchCV()` and `GridSearchCV()` to get more optimized models for the gradient boosting methods. To tune our neural network's hyperparameters with used the package `Ax` which is part of the *Facebook* Open Source projects. It allowed us to tune hyperparamters such as the number of layers, neurons, the batch size, the learning rate or the optimizer.

5 CONCLUSION

In the end, we achieve a satisfactory predictive power with a scalable model. However, there is potential for improvement. Firstly, the training set we used had only 8902 observations. If we had access to more historical data, the model would have a lower estimation variance and higher predictive power. Secondly, it would be interesting to have access to data with higher periodicity (daily). In this way, we would have access to higher degree of precision for the model to identify trends. Finally, we did not explore airline specific data: company insights like brand preference, pricing, and customer profiles could have a strong influence on the target value.

Final submission: **Lewis** by **nico_bdls**