EMINES School of Industrial Management

EMINES Datacamp - Outcoder challenge

NUMBER OF FLIGHT PASSENGERS PREDICTION

Yassine Doufani
yassine.doufani@emines.um6p.ma

Youssef BEN ALLAL
youssef.benallal@emines.um6p.ma

Supervisor(s): Mr. Mathurin Massias
INRIA - EMINES UM6P

17/01/2022

Contents

1. Introduction

1.1 Task

The goal of the challenge is to predict the number of passengers per plane on some flights in the US. The data is provided to us by a single company. From the company point of view, the interest of this challenge is to be able to evaluate the percentage of no-show reservations, in order to properly calibrate overbooking. Some passengers make reservation but do not show up on the flight, leading to empty seats in the plane. Estimating the number of passengers effectively boarding the plane is thus important for the company. The left-out data has dates that come after the training data, so a time series approach is possible.

1.2 General description

During this competition, we have tried to perform multiple approaches to predict the target which represents a transformation of the number of passengers. We started the journey with a simple data exploration to understand the problem and the given data, we chose Random Forest as our baseline model. The next step was mostly about features engineering and searching for external data which can explain the target, give more information about the flights and enrich the dataset with new features. Finally, we tried to perform different regression models on the processed data and evaluate each model based on the metric chosen for the competition.

2. Data

2.1 Data sources

- Available Data : The training data is made available as a dataframe, whose columns are:

- flight_date: the flight's takeoff day
- from: the IATA code of the departure airport
- to: the IATA code of the arrival airport
- avg_weeks: average number of weeks between booking and flight date, across passengers
- target : transformation of the number of passengers boarding the plane

- External data (added) :

The external data approach began with using **stock data (closing price)** which was taken from American Airlines Group Inc. (AAL), Finance.Yahoo during the same period. So, what we did exactly is merging the closing price of AAL with flights data using the date column.

In terms of **weather data**, we collected data first using Meteostat python library and then verified the initial data with an other source that we found on a flight delays Github repository and that features the NOAA data (The National Oceanic and Atmospheric Administration) using latitude and longitude during the same date period. We assumed that the following variables may have an important impact on the percentage of no-show reservations in the day of the flight.
Some of the weather data collected for both arrival and destination airport are temperature, precipitation, visibility, wind speed, weather event... .

We also computed the **distances** between airports using airport geographical codes.

2.2 Data exploration

2.2.1 Data distribution

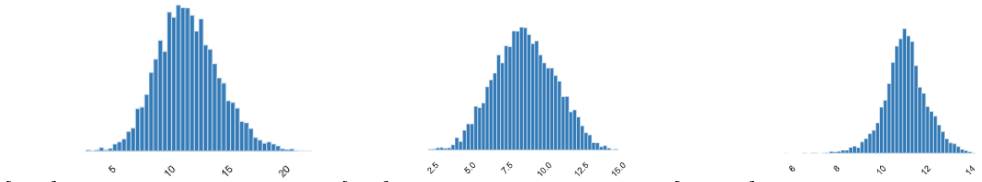

Figure 1: Features distribution

Both std_weeks and avg_weeks feature are close to normal distribution. The avg_weeks has a skewness coefficient of 0.26 and the std_weeks has a skewness of 0.04. On average, most of the passengers books their flight between 10 and 15 weeks before the flight date. the target feature is a little bit skewed to the right with a skewness coefficient of -0.15 which is not yet a large value to consider applying a feature normalization.

2.2.2 Temporal and spatial exploration

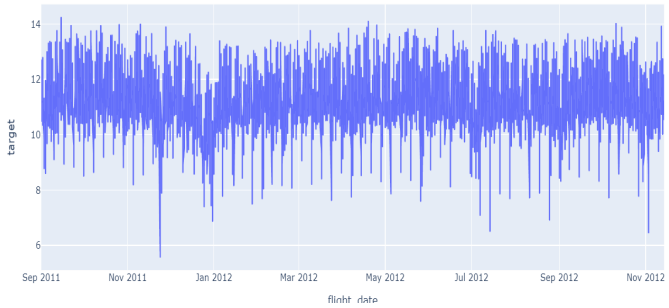

Figure 2: Target evolution over time

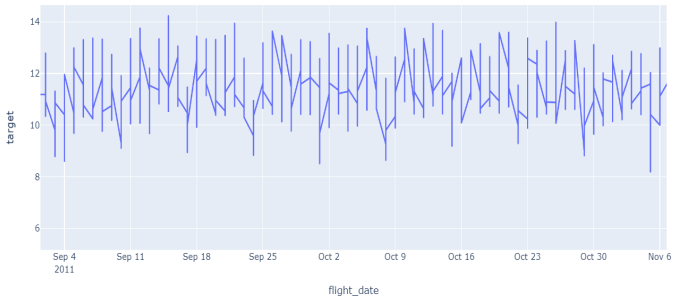

Figure 3: Target evolution on a weekly scale

Concerning the temporal evolution of the target we notice a certain change before and in the new years eve, we also notice a certain monthly periodicity in the first visualisation and a weekly periodicity in the second visualisation where we notice a change in target at the end of each week. As we don't have a hourly information we cannot include the intraday feature. But it could be interesting to add other dates information such as holiday data especially the key US holidays like new years eve and independance day, also adding a month and weekend variable could have an impact on the model performance.

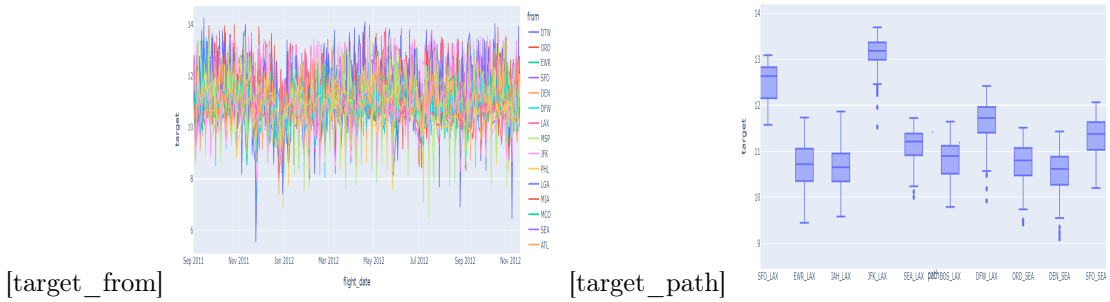

Figure 4: spatial analysis of target

Here we first visualise the temporal change of the target considering the destination airport, and the variation of the target considering the path of some flights. We notice that in general when including the path the variation of the target gets smaller which means that the path variable is more meaningful than the destination or arrival variable alone.

2.2.3 Weather impact

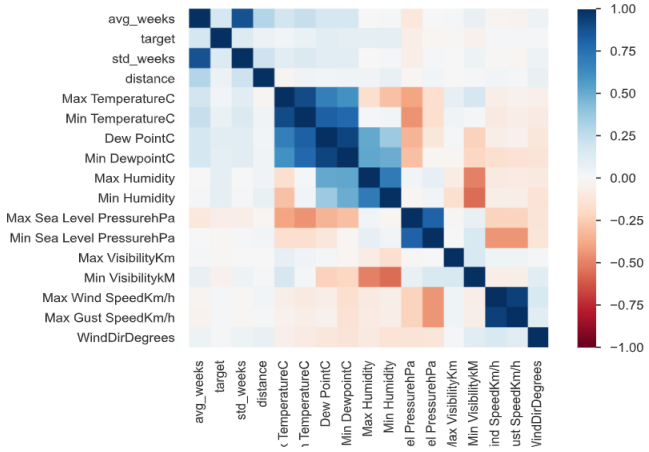

Figure 5: Correlation Matrix including weather data

After we added the external data including the weather we decided to compute the correlation matrix in order to see if our intuition about weather impact on target is verified. However, we notice that most the weather variables correlates poorly to the target variable, the mean of pearson's correlation for weather data is about 0.08. In the external data part of the modelling we will include only weather data with an absolute correlation larger than the mean.

3. Modeling

3.1 Baseline Model

Our baseline model is a model where we encode both the iata code of destination and arrival airport.
The features used for this regression model are **std_weeks**, **avg_weeks**, **iata_from(label_encoded)**, **iata_to(label_encoded)**.
In this step we don't do any related transformation to the feature engineering pipeline that we will present in the next part.
We summarize in the following table the model performances after performing a five fold cross validation for the model evaluation:

RMSE Results		
Model	RMSE mean	R^2 mean
Random Forest	0.78	0.31
Linear regression	0.81	0.25
Decision Tree Regressor	1.01	0.10

3.2 Feature engineering

In this step, the objective is simply extracting derivative features from the features that the dataset originally contained. In a supervised approach we wanted to make use of the flight date column, so we extracted the following features:
Year, month, weekday, week (the number of the week the flight is happening), weekend (a flag indicating if the flight is happening during a weekend), days_to_nye (the number of days between New Year's Eve and the flight), days_to_ind (the number of days between Independence Day and the flight).

Besides the previous features extracted for the flight date column, we added new a column that indicates the path which a flight is taking by a simple fusion of the departure and arrival airports columns. Ex: departure: ATL, arrival: JFK $\longrightarrow$ path: ATL-JFK. After creating the path column, we performed a OneHotEncoding processing to allows the representation of the path to be more expressive, this made the number of features increase by 130 columns.

We simulated the models performances after this feature engineering pipeline using the recent machine learning named Pycaret. The table below summarizes the scores obtained from pycaret simulation :

RMSE Results		
Model	RMSE mean	R^2 mean
Catboost Regressor	0.37	0.81
Light GBM	0.38	0.78
XGboost	0.40	0.77
Random Forest	0.51	0.67
Extra trees regressor	0.55	0.59
Linear Regression	0.63	0.59

3.3 External data

In addition to our previous feature engineering pipeline we added external data described on the data section of the report. The external data enhances a little bit the average model performance but increase the model variance, for example we got an averge RMSE close to 0.35 for catboost but with a standard deviation of score of 0.15. In general all our submissions including the external data especially when including the distance between airports overfitted and gave us scores close to 1.5 in leaderboard. This is despite of the feature importance of this external data.

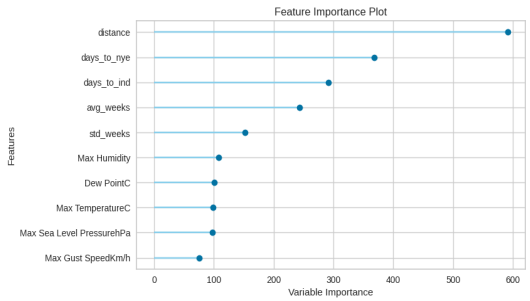

Figure 6: Feature importance using external data

4. Conclusion

In this challenge, we tried to think of different approaches. The time-series approach was not an intuitive or direct one since there are many flights per day and the hourly information was not present. We tried then to explore the reframing of series data into supervised data, by taking into account lagged value of different features including the target but this method didn't give a better result than the supervised approach presented in the report.

In a second search, we tried to look for approaches that take into account the correlation between series, we found the vector auto regression time series model but we didn't go further since most of the paths don't have daily flights.

Finally, The supervised approach blending most performant estimators in addition to the feature engineering pipeline gave better results. We tried to tune the models especially Catboost to reach an RMSE close to 0.30 in our validation data but the best validation RMSE was 0.35, which gave us a result of 0.40 in the submission.

This challenge was an opportunity for us to look for different approaches, although we mainly focused on a supervised approach with boosting algorithms, we are still motivated to go further in the graph approach that we didn't have time to implement and adapt for our problem.