Predicting flows of US air travel passengers

Aicha BOKBOT | Arthur KRIEFF
Submission: arthurkrieff "lastone_AichaArthur"

INTRODUCTION

We are considering an airline company that has 100-200 airplanes with crew deserving numerous connecting cities in the United States.

The purpose of the project is to predict the number of air passengers that will travel at a given date on a given route. Another goal is to find a way to effectively allocate aircrafts to routes.

In order to address this regression problem (we are trying to predict a continuous variable), two challenges must be faced:

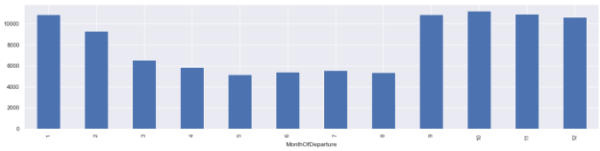
1) finding a good set of features to be used to predict the air passenger flow
2) using an efficient and robust learning algorithm to train the regressor

In this paper, we first give a description of the features we used, followed by the various approaches and algorithms we implemented. At each step, we mention our successful as well as unsuccessful approaches.

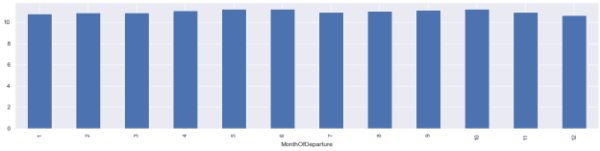
I. FEATURE ENGINEERING

The available historical data is the following: for each **date of departure**, we know the **number of passengers** (*called **logPax***), the **departure** and **arrival airports** and the **weeks to departure** when the tickets were booked.

The flights considered are between January 2011 and March 2013. The airports served are the 20 busiest American airports at the time.



total volume of passenger by month of departure



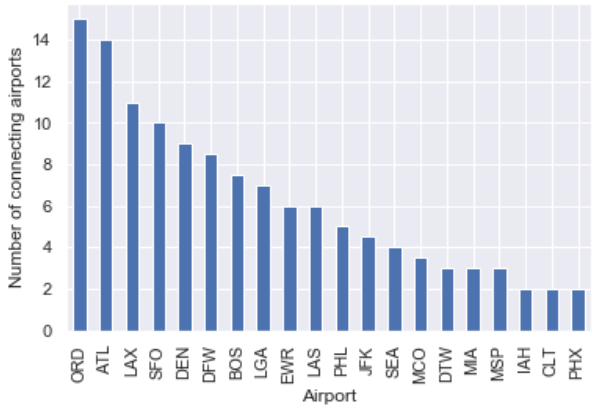
average number of passengers per flight by month of departure

We notice that the average number of passengers per flight is approximately constant from one month to another, whereas the total volume of passengers by month shows a peak season from September to January and a low season from February to August. This implies that the airline company allocates more flights during the peak season in comparison to the low season.

These are the features that we judged relevant to the analysis:

Features derived from initial data:

- **Day of month**, **weekday**, **week**, **month** and **year** of departure
- **Connectivity index**: sum of the number of connecting airports to departure and arrival airports, based on the training set
- **Routes**: we tried two main ways to represent the routes:
  - One column per route (e.g. "JFK-SFO") – which gives 126 variables
  - One column per airport (no distinction between departure and arrival). We chose this representation as it minimizes the number of features



Number of connecting airports for each airport

Features collected from external sources:

- **Holidays:** categorical variable indicating whether the date of departure corresponds to the same day or day before a US Federal Holiday [2]
- **Temperature**: monthly average temperature for departure and arrival airport [3]
- **Humidity:** monthly average humidity for departure and arrival airport [3]

- **Population size of airport:** annual passengers in 2012 for departure and arrival airport [1]
- **Inversed distance** in km between departure and arrival airports [4]
- **Purchasing Power Index** of departure and arrival cities [5]

The idea of the three last features was brought to us by empirical studies showing that air passenger volume can be modelled using a gravity model composed of GDP, population and distance [6].

As for the pre-processing of data, we encoded the temporal variables derived from the date of departure.

II. MODELS

After selecting our variables, we started working on the regression models.

II.1 – Linear Regression

We first considered Linear Regression.

(i) Motivations

By looking at the correlations between the target and explanatory variables, we decided to assume that the target variable was linear with respect to the numerical variables (WeeksToDeparture, std_wtd for instance) and run a Linear Regression – even though the scatter plots extracted from the data didn't show any obvious linearity.

(ii) Results

The model did not perform well on the training set (poor R_2 score) and it performed even worse on the testing set. Some coefficients were aberrant (order of 10^{10}). How can we explain this performance ?

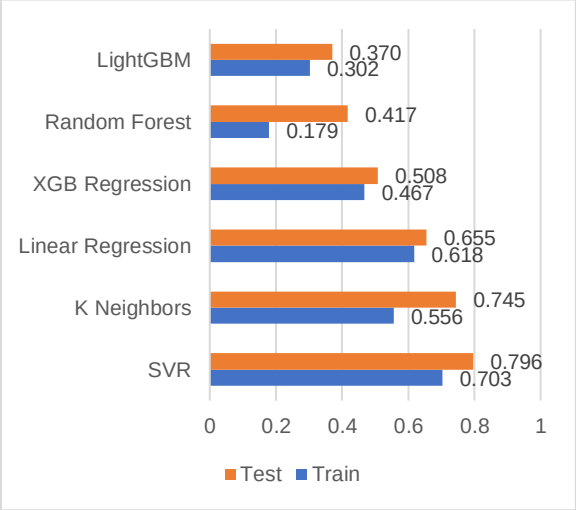
Firstly, by the presence of many categorical variables (days, months, journeys…), which the model considers as numerical variables. Secondly, by the actual non-linearity between the numerical variables and the target variable.

By examining more the data, we realized that several variables were not linear with respect to Y: the mean_temperature or the inversed distance between airports for instance.
As a result, we decided to put aside the Linear Regression model.

II.2 – Non-Linear Regression

(i) Selection of the best models

After some research, we came across several Regression models that do not assume linearity. To have a quick idea of the best models, we built a loop to test the following models : Random Forest Regressor, Support Vector Regression, Decision Tree Regressor, XGB Regressor, LightGBM Regressor, K Neighbour Regressor and the MLP Regressor. The models were tested with their default hyperparameters.



Performance of algorithms

The best scores were given by the Random Forest Regressor, the XGB Regressor and the LightGBM Regressor. As a result, we decided to focus on these three models.

(ii) Hyperparameter tuning

To improve the accuracy of the models, we performed a GridSearchCV on the three models.

For Random Forest, we focused on optimizing: max_depth, max_features, n_estimators, min_samples_leaf and min_samples_split.

For XGBoost, we focused on: Max_depth, min_child_weight, n_estimators, gamma, subsample, colsample_bytree, reg_alpha and learning_rate.

As for LGBM Regressor, we optimized: colsample_bytree, min_child_weight, max_bin min_data_in_leaf, num_leaves, n_estimators, reg_alpha, reg_lambda, subsample and learning_rate.

The scores on the testing set were the following:

- Random Forest: 0.400
- XGBoost: 0.351
- LightGBM: 0. 346

Therefore, we chose to keep the LightGBM model.

III. MODEL IMPROVEMENTS

III.1 – Clusterization

We thought about applying a clusterization method to further improve the model.

(i) Motivations

The idea of clustering came with the following thought: some flights may be intended more for business purposes while others may be intended more for leisure purposes. Consequently, the number of passengers would depend on the purpose of the flight (=on the cluster). The goal was then to, firstly, identify the cluster for each flight, and secondly, to apply a regression model based on the cluster. We would have then two different regression models, one for each cluster.

(ii) Results

- **KMeans**

To cluster our data, we fitted the KMeans models to our training set, we then predicted the cluster for our training set. After that, we fitted two LightGBM Regressors on the training set: one for the cluster 0 and one for the cluster 1. We were then able to predict the cluster for our testing set and to predict the logPax value, depending on the predicted cluster.

Since we had new models, we performed two GridSearchCV to get the best hyperparameters for our two regression models. Unfortunately, the scores were worse than before.

- **KModes**

Since KMeans doesn't deal well with encoded categorical variables, we tried to use KModes instead. However, it did not lead to good results as it performs well only with categorical variables.

- **KPrototypes**

We lastly decided to try the KPrototypes algorithm, which is built to deal with datasets combining both categorical and numerical variables, but the results were still not good.

As a result, we decided not to cluster our data.

III.2 – Feature Selection

As a reminder, we decided to use the LightGBM regressor and obtained scores of 0.019 on the training set and 0.345 on the testing test.

Our model does much better on the training set than on the test set, which allows us to think that the model is overfitting.
One way to reduce overfitting is through feature selection by removing irrelevant input features.

We first tried removing some features manually. We noticed that the model gives slightly better scores by ignoring the following variables:

- day of month
- temperature
- humidity
- Purchasing Power Index

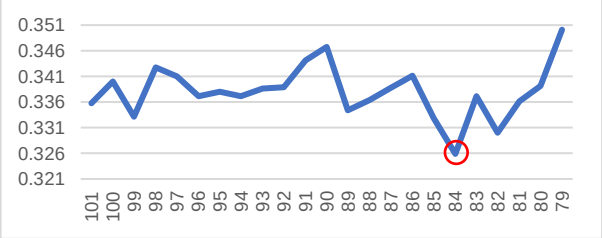
This could be explained by the fact that the day of month variable is redundant with the other temporal variables and add a granularity which complexifies the model.
As for the temperature and humidity, it might not be a decision criteria for the air travellers inside the United States.
Lastly, the Purchasing Power Parity might not add much value to our model because the 20 American cities observed are approximately equivalent in terms of Purchasing Power – this variable could however be relevant when comparing flights between countries with different economic levels.

By removing those features, we now have a dataset of 101 features and a score of 0.029 on the training set and 0.336 on the testing set.

We implement feature selection using the SelectFromModel class that allows to transform a dataset into subsets with selected features and apply a model on those subsets.

```
thresholds = sort(model.feature_importances_)
for thresh in thresholds:
    # select features using threshold
    selection = SelectFromModel(model, threshold=thresh, prefit=True)
    select_X_train = selection.transform(X_train)
    # train model
    selection_model = LGBMRegressor(colsample_bytree= 0.8,
                                    min_child_weight= 0.01,
                                    min_data_in_leaf=5,
                                    num_leaves= 70,
                                    n_estimators=800,
                                    reg_alpha= 0,
                                    reg_lambda= 0.1,
                                    subsample= 0.5,
                                   learning_rate=0.15,
                                   max_bin=100)
    selection_model.fit(select_X_train, y_train)
    # eval model
    select_X_test = selection.transform(X_test)
    y_pred = selection_model.predict(select_X_test)
    score = score_type(y_test, y_pred)
```

These are the results obtained by using this method:



We see that the score on the testing set reaches a minimum when using 84 features.
So we decided to keep only the 84 features with the highest feature importance.

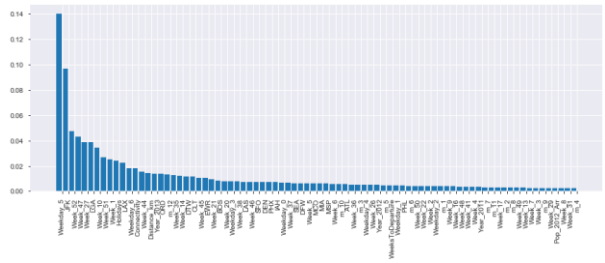
These are the features removed:

- the standard deviation of Weeks to departure
- the population size of departure airports
- the weeks 11, 12, 15, 18, 19, 23, 24, 25, 28, 30, 32, 33, 34, 39, 30, 42, 43

III.3 – Dimension Reduction

(i) PCA

We thought about performing a PCA to reduce the dimension of our dataset.



Feature importance

A significant portion of the features has very low feature importance (as shown in the graph above). This lead us to hypothesize that applying our model on principal components could boil down to a better performance with reduced overfitting.

We implemented PCA and found that two principal components explain 99.99% of the data's variance. However, applying our model on the data set fitted and transformed to PCA lead to poor performance (RMSE of 0.8).

(ii) FAMD

Because PCA only works well with numerical variables, we tried to find a method that works for both categorical and numerical variables. The **F**actor **A**nalysis of **M**ixed **D**ata (FAMD - library Prince) allowed us to reduce the dimension of our dataset. However, once applied on our model, the results were still not good.

Therefore, we decided not to reduce the dimension of our dataset.

IV. CONCLUSION

In a nutshell, we opted for the LightGBM model and improved it with Feature Selection. Our final features and model gave us a score of 0.268 on the Ramp platform.

As for the second question about finding a way to effectively allocate aircraft to routes, we suggest the airline company considers these four elements:

- air travelers demand forecasting
- connectivity of airports served
- aircraft availability
- matching the competition with other airline companies

REFERENCES

[1] "List of the busiest airports in the United States", Wikipedia (link)
[2] US Federal Holiday Calendar built-in Python function
[3] External data provided by the RAMP data
[4] Open flights website (link)
[5] Numbeo (link)
[6] Modeling monthly flows of global air travel passengers: An open-access data resource (link)