Predictive Modelling with Machine Learning

Flight Price Predictor

Team Name : **SiliconLobby**

Team Member: **Arkadeep Das**

Team Member : **Hiten Sethiya**

Introduction

Optimal timing for airline ticket purchasing from the consumer's perspective is challenging principally because buyers have insufficient information for reasoning about future price movements. In this challenge we simulate various models for computing the best possible expected future prices by constructing new features from the existing features and also introducing new features to make our model more robust.

Data Description

The dataset comprises of the departure and arrival airports, name of the passengers (including Gender and Designation) , birth-date, class of flight, flight time , booking date , Fight date and the fare (the value of which is to be predicted). Based on the features, we needed to predict the fare of the flights from an unseen test dataset.

Data Preparation/ Feature Extraction and Augmentation

By initial inspection of the dataset, we did certain basic cleaning and feature engineering. A lot of data preparation needs to be done according to the model and strategy we use, but here are the basic cleaning we did initially to understand the data better :

1. **Duplicates** : First we observed the presence of duplicated in the dataframe that we loaded from the training dataset. If present we discarded them.
2. **Days to departure** : Difference between booking and departure date :
   We observed that fare was correlated with the number of days to departure with value of correlation = -0.21087 . Hence, it constructs an important feature of our dataset.
3. **Age of the passenger :** The age of the passenger was also found to be correlated with the fare of the flight , with value = 0.105528
4. **Weekends** : The flight prices were highly influenced by the fact that the flight date was on a weekday or a weekend. We grouped all the 7 days of the week based on the observed mean fares on that date . It was observed that on weekends the prices were significantly higher than the weekdays. We also included the major holidays of 2016 in addition to differentiating between weekdays and weekends.
   The correlation was also significantly high with a value of 0.248065.
5. **Designation of the Passenger:** We also considered the designation and the gender of the passenger (Dr, Mr, Mrs, Miss) as a feature extracted from the names. The designation and the gender showed high correlation with the fare of the flights, with a value of 0.226093.
6. **Flight class :** The Flight prices also vary evidently with the class of the Flight.
   The business class flight prices were significantly higher than economy class flights.
7. **Flight Time :** The Flight times also affect the flight prices . So it was also included as a feature. The minutes were discarded and only the hours were kept as a feature.
8. **Month of the Flight:** Flight month is also included as a feature since the fares vary well over months.
9. **Week of the Flight :** Flight week is included as a feature since the flight prices become significantly higher when holidays or vacations are upcoming since a lot of bookings are done in the same week or the week prior to the vacations or holidays.
10. **Day of the Flight :** Flight day is also introduced as a feature and encoded sequentially as for example : Monday : 0 , Tuesday : 1 till Sunday : 7.

New Features introduced :

Certain new features were also introduced which greatly improved the robustness of the model. The features are as below :

1. **Aerial distance of the two airports :** The distance between the two airports was highly correlated with the airfare , value of ~0.33. We scraped the airport distances from web and introduced a new feature distance based on the aerial distance between the arrival and departure airports.
2. **Flight Duration :** The flight duration was also observed to be highly correlated with the flight prices. One interesting observation was also that the average flight duration between two airports change when the arrival and destination airports are swapped.

Feature Encoding:

We encoded the 7 cities sequentially, thereby encoding the 'From' and 'To' features respectively. The holidays were encoded as 3, weekends as 2 and weekdays as 1.
For gender we assigned names with 'Dr' designation as 4, 'Miss' as 0, 'Mrs' as 1 and 'Mr' as 2. All other features were kept with floating point data types. The age was scaled to years.

Model :

For this regression task I have used xgboostregressor. Prior to that I have tried Logistic Regression, DecisionTreeRegressor, RandomForestRegressor and MLPRegressor but xgboost outperformed them all by huge margins.

The XGBoost library implements the gradient boosting decision tree algorithm.
Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. XGBoost is a scalable and accurate implementation of gradient boosting machines and it has proven to push the limits of computing power for boosted trees algorithms as it was built and developed for the sole purpose of model performance and computational speed.

I have used a distributed asynchronous hyperparameter optimization approach (since brute-force grid search takes up a lot of time) for searching the best hyperparameters for fitting each xgboost Regressor model . The parameters I optimised for were 'min_child_weight', 'gamma', 'subsample' , 'colsample_bytree', 'max_depth', 'learning_rate'.

xgb = XGBRegressor(n_estimators=500, objective='binary:logistic',silent=True, nthread=24,max_depth=20,learning_rate=0.02).
Since it is a regression task, i.e predicting flight fare I have set the objective to reg:linear. Optimal number of estimators were found to be 500, because above that it would start over-fitting. Maximum depth of tree was also found to be optimal at 20.

Outliers were also detected in the dataset but did not have any effect in the results.

Predicted score :

R2 score obtained with xgboost on training dataset ~ 0.9999999666

Final R2 score on test dataset : ~ 0.9602