RESTAURANT VISITOR FORECASTING

Using analysis techniques and XGBoost Regressor

By : MANIK MARWAHA (a1797063)

Report submitted for **COMP SCI 7209 Big Data Analysis and Project** at the School of Computer Science, University of Adelaide towards the Master of Data Science

Abstract

In this project, I have used extreme gradient boost regressor multiple to predict the number of visitors in a restaurant for a given time period. We were provided a separate training and testing datasets. Exploratory data analysis was followed by various preprocessing techniques for the purpose of feature engineering. The parameters were then tuned from the training sets. These hyper tuned parameters were used to train the model. The model was trained by splitting dataset into 8 folds. For this purpose TimeSeriesSplit function was applied with parameter n_folds=8. The root mean square logarithmic error (RMSLE) was then calculated for training and validation sets for each fold. The mean RMSLE for the validation set is found to be 0.5093 and the Normalized RMSLE is 0.8151. The high value of normalized error shows high accuracy of the model for making predictions.

INTRODUCTION

The restaurant business owners always want to know how many customers they can expect each day as this may help them to calculate their daily profits/loss in advance and hence make suitable changes to help them run their business more efficiently. However such predictions depend upon many factors like weather, time of day, day of the week, season, local competition. For this purpose, Recruit Holdings which owns a restaurant review service and a sales service organized a Kaggle competition and provided historical data about reservations and the number of visitors. The purpose of this competition was to help restaurant owners predict the visitors for given time period(last week of April and May 2017).

Prediction methods mainly includes regression(a continuous value also called predicted variable), classification or density estimation, in which some attribute of data is predicted using some other attributes(such variables are called predictor variables).

Visitor prediction can be formulated as a *regression task*. Regression analysis is a statistical technique used to estimate the relationship between a dependent/target variable (visitors) and single or multiple independent (interdependent) variables (predictors) that impact the target variable. Regression analysis also lets researchers determine how much these predictors influence a target variable. In a regression task, a target variable is always numeric.

Predictive analytics is carried out by analyzing the current and historical data to forecast the probability of future events or values in the context of the number of visitors.

DATASETS DESCRIPTION

The overall dataset for forecasting restaurant visitors has been adopted and made available via Kaggle (Kaggle 2017). There are a total of 8 datasets available. There are 2 systems - HPG and AIR which store details of different restaurants, hence each dataset is related to one of these 2 systems. These 8 data sets along with their columns have been discussed below:

1) Hpg_reserve.csv : All the reservations made in HPG system are present in this file
    - Hpg_store_id - id of the restaurant
    - Visit_datetime – reservation time
    - Reserve_datetime- time reservation was made
    - Reserve_visitor s - visitor count given in the reservation
2) Hpg_store_info.csv : has information of all restaurants in HPG system
    - Hpg_store_id - id of the restaurant
    - Hpg_genre_name : genre of restaurant
    - Latitude
    - Longitude
    - Hpg_area_name : area of restaurant
3) Air_reserve.csv : All the reservations made in AIR system are present in this file
    - air_store_id - id of the restaurant
    - Visit_datetime – reservation time
    - Reserve_datetime- time reservation was made
    - Reserve_visitor s - visitor count given in the reservation

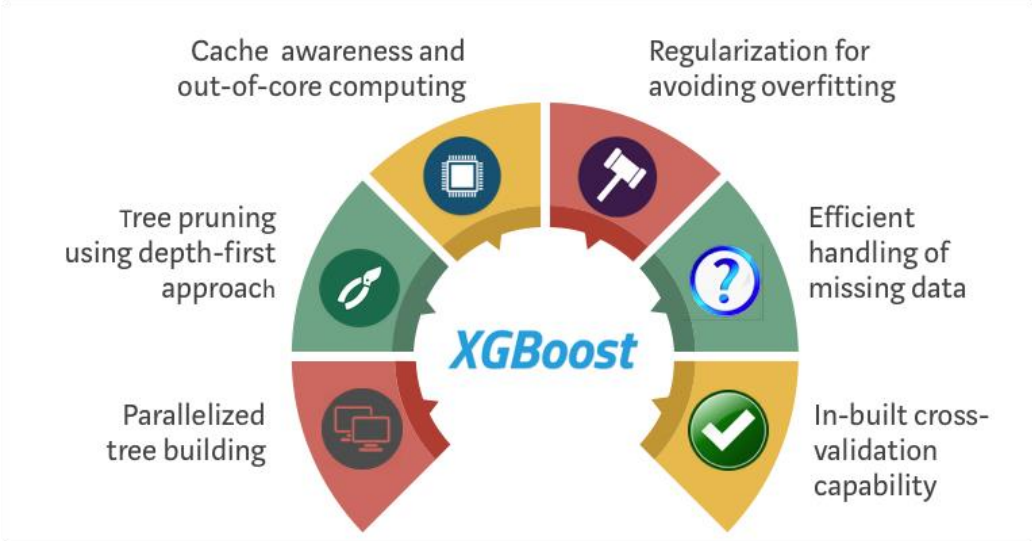
4) Air_store_info.csv : has information of all restaurants in AIR system
    - air_store_id - id of the restaurant
    - air_genre_name : genre of restaurant
    - Latitude
    - Longitude
    - air_area_name : area of restaurant
5) air_visit_data : contains all the historical data of visitors in AIR system
    - air_store_id : restaurant id
    - visit_date – date of visit
    - visitors – total number of visitors on the given day
6) store_id_relation : can be used to join certain restaurants that use both HPG and AIR systems
    - air_store_id
    - hpg_store_id
7) sample_submission_csv : contains the format of submission
    - id
    - visitors – the Prediction to be made
8) date_info.csv : gives information about calender dates
    - calender_date
    - holiday_flg – value is 1 if holiday, 0 if no holiday
    - day_of_week

EXTREME GRADIENT BOOST REGRESSOR MODEL (XGBoost)

Gradient boosting is a very popular machine learning algorithm which uses an ensemble of weak learners model. In this case we have taken decision tree as the week learner. Gradient boosting can be used for regression and classification problems. It works very similar to other boosting methods and allows a differentiable loss function to be optimised. Just like other boosting methods gradient boosting also adds new models to compensate for the errors made by the previous models. In the end we obtain an ensemble of these models which helps to perform prediction or classification tasks.

XGBoost is a modified version of gradient boosting model which gives a higher speed and more efficient performance. It is a combination of hardware and software optimization techniques that consumes less amount of time and lesser computation and in return gives superior results. XGboost is a very important algorithm that can be used for small/medium structured/tabular data.

ADVANTAGES OF XGBOOST



Source – www.towardsdatascience.com

IMPLEMENTATION

All testing was performed in Jupyter Notebook using a Python kernel. Scikit-learn(sklearn) was used as the primary machine learning library, responsible for the imported Linear Regression, Label Encoder, imputer and also to calculate the mean squared error. Matplotlib library was used to plot different plots for evaluator data analysis. As mentioned previously the datasets were collected from Kaggle and read by python kernel using pandas.read_csv.

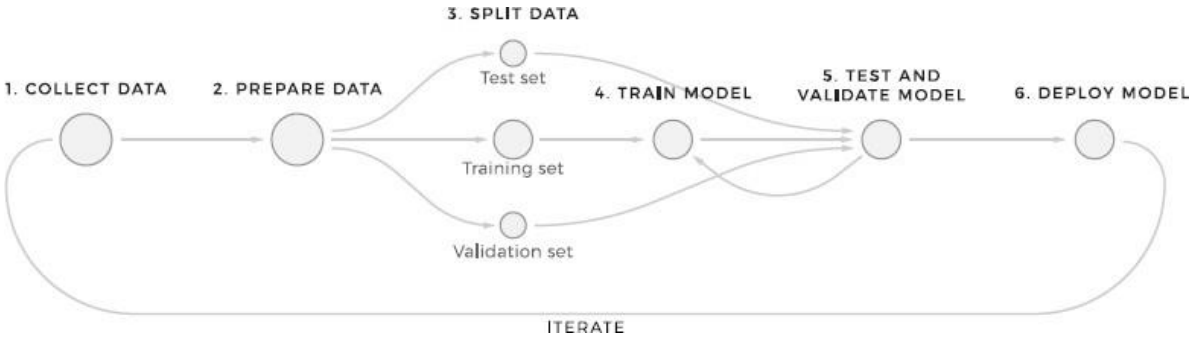


Figure 2 – Implementation of this project

COLLECTING DATA

As mentioned previously the datasets were collected from Kaggle and read by python kernel using pandas.read_csv. All the datasets were stored in a dictionary named files with keys of the dictionary being names of the datasets.

LIBRARIES USED

Scikit Learn is the main library used for programming purposes: it provided us various metrics like mean squared error and mean squared log error. Label encoder to handle categorical features and

TimeSplitSeries function for time series analysis were also imported from sklearn. It is also used for parameter tuning by importing GidSearchCV.

Datetime library was used to extract different date and time features from the dataset.

XGBoost is the main machine learning algorithm used for prediction. XGBoost library provides us with gradient boosting regression and classification techniques. So that has also been used to train our model and make predictions.

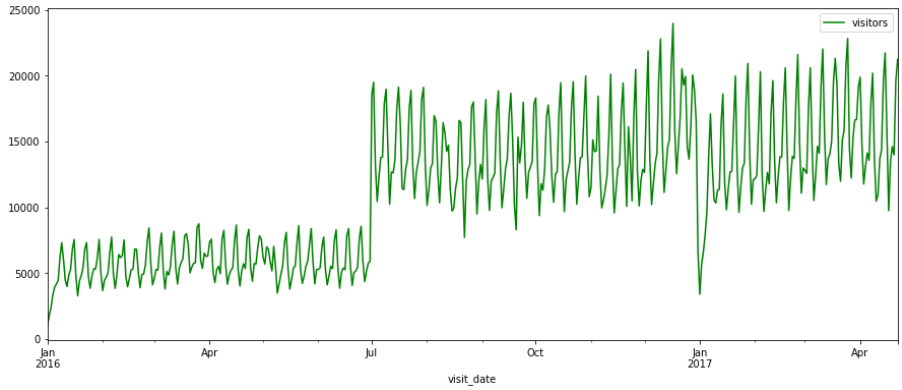
Numpy and pandas are also used as a part of different pre-processing techniques.

Matplotlib is used for plotting different graphs and plots for exploratory data analysis.

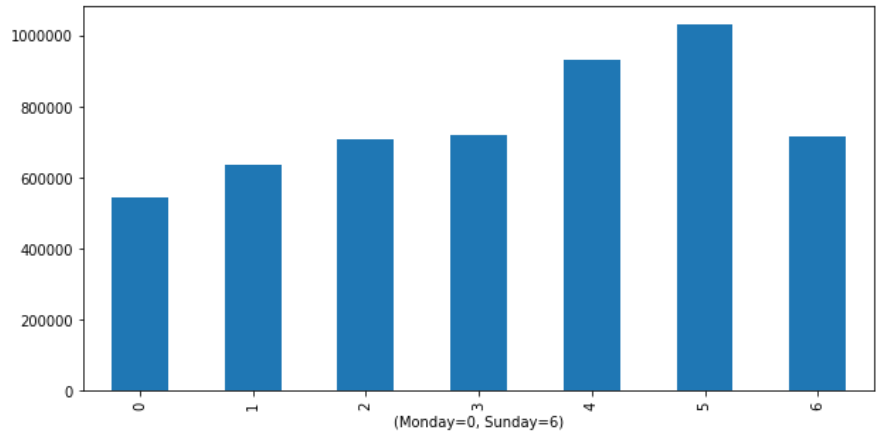
EXPLORATORY DATA ANALYSIS

For any predictions and classifications, before we pre-process the data it is very important to analyse the datasets to find any specific patterns or certain useful information which may help us in the process of pre-processing. Certain observations I made during the analysis are listed below along with the plots.

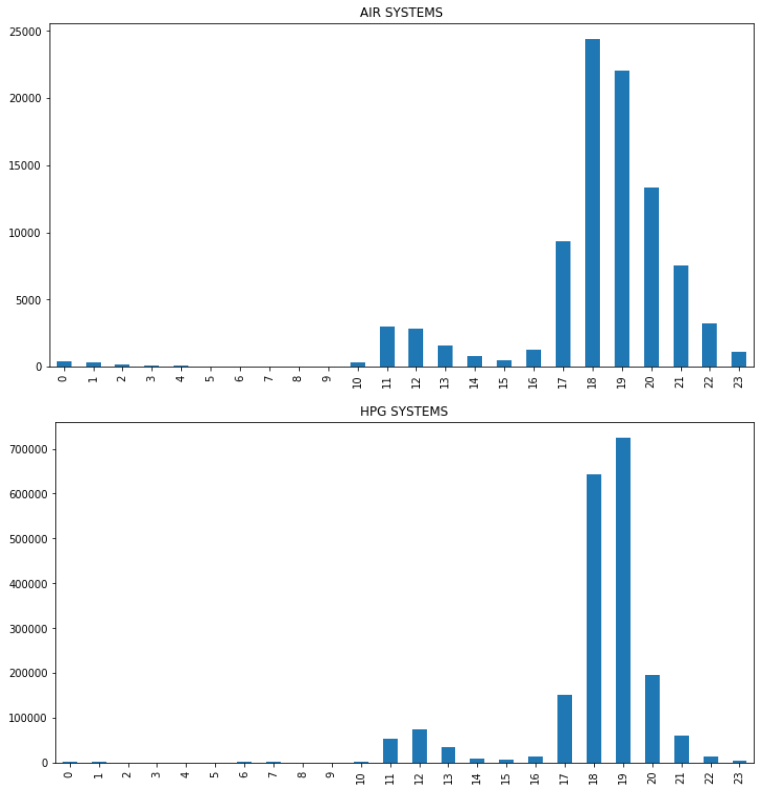
1) This graph shows that from July 2016 there was a sudden spike in the total number of visitors visiting the restaurant. One of the main reasons for this can be new restaurants opening up in the area.



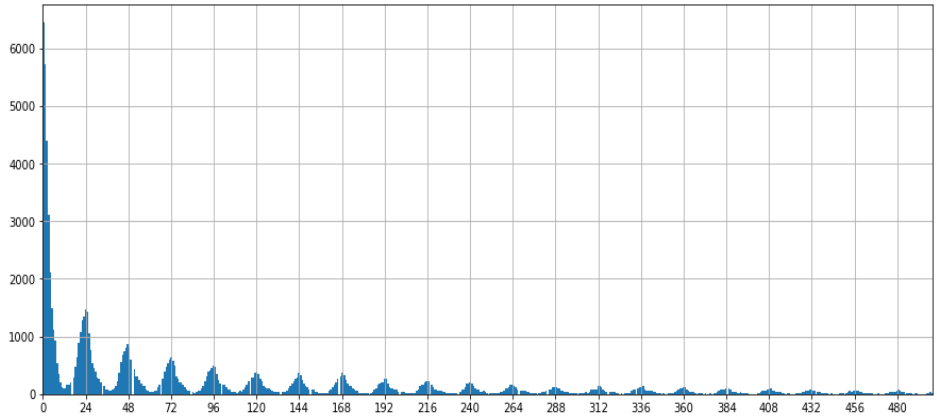
2) We also observe that maximum number of visitors visit the restaurant on the weekends especially on Fridays and Saturdays. Saturdays are mainly holidays that why have the maximum visitors. This is shown in the bar plot below.



3) From the reservation datasets of both the air and hpg systems, we observe that maximum reservations were made for dinner. This shows that maximum people visited the restaurant for dinner and hence evening times experienced more rush of customers.



4) Lastly we observe that most people make reservations a few hours before their visit to the restaurant. While a minority of people do so as multiples of 24 hours ahead of time.



DATA PRE-PROCESSING

The next step involves preparing the data for analysis. This usually includes the pre-processing of data. In this project 4 main steps for pre-processing of data are used:

- Taking care of missing values
- Feature engineering
- Taking care of categorical features (transforming features)
- Splitting of training dataset into a small training and validation sets

HANDLING MISSING VALUES – we check the number of missing or null values in each dataset by

using dataframe.isnull().sum(). We observe that none of the datasets contain a null value therefore no imputation is required of the missing values.

FEATURE ENGINEERING – Since this prediction problem had many datasets and a lot of data to be computed the feature engineering was a lengthy process.

Step 1 : first we merged the hpg reservation dataset with the store_id_relation dataset using pandas.merge(). Then for both reservations datasets of the AIR system and HPG system we find the time difference between the time reservation was made and the time for which reservation was made and added a new column to both these datasets called 'reserve_datetime_difference'. The 2 columns used to calculate this difference were removed from the datasets (reservation date and visit date columns).

Step 2 : Next I extracted the date time features from the dataset 'air_visit_data' which contains all the historical data of the visitors visiting restaurants of AIR system. Day, year, month, date were separated from the 'visit_date' column for all the data points (samples).
I also used the 'sample submission' dataset to extract different features. The 'id' column in the sample submission held information like the visit date and id of the restaurant for which predictions were to be made. So from the 'id' column we extracted the store_id, day, month, year, date for which predictions were to be made and stored this extracted information into new columns in the same dataset. Hence, this way I modified the sample submission dataset.
Also, the total number of unique restaurants for which prediction was to made were found.

Step 3 : the next step is to create statistical features from restaurant visits every day of the week. This include :-
Minimum number of visitors per store
Maximum number of visitors per store
Mean number of visitors per store
Median number of visitors per store
Visitors count per store
These statistical features are then added to the visitors dataset of the AIR system.. these features play a very important role in predicting the visitors count.

CATEGORICAL FEATURES – categorical features like 'genre_name' are transformed using LabekEncoder of scikit learn. Leabelenoder.fit_transform() is used to convert this features and create dummy columns for each different value of genre_name. LabelEncoder is also used to transform 'day_of_week' features of 'holiday' dataset. Once these features have been transformed they are added to the air_visit_data and sample submission datasets.

SPLIT DATA TO TRAINING AND TESTING SETS
Once feature selection is done the data can be finally be split to training and testing sets. This is the final test before training the model and applying it to make the predictions.
Training Dataset Overview:-

1. Total number of unique restaurants in AIR system:- 829
2. unique genre restaurants in AIR system:- 14
3. Different locations of AIR restaurants :- 103
4. Average daily visitors:- 20.973761245180636
5. Training data duration:-2016-01-01 to 2017-04-22

Test Dataset Overview:-

1. Unique number of restaurants:- 821

2. Test data duration:- 2017-04-23 to 2017-05-31

EVALUATION METRIC

Generally for regression tasks root mean square error (RMSE) is used for evaluating the results, but for this problem we will use the root mean square logarithmic error (RMSLE).

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(p_i + 1) - log(a_i + 1))^2}$$

n : number of data points for which prediction has to be made

p_i : predicted value

a_i : actual value

REASONS FOR USING RMSLE

1) Penalizes underprediction more as compared to over prediction.
   In case of overprediction the restaurant may not suffer much losses. But in case of underprediction, restaurants especially the smaller ones may suffer huge losses as alloy food can get wasted.
2) Penalize relative to magnitude of number. Example:

Actual = 1000	predicted = 1400	Actual =10000	Predicted = 10400
RMSE = 400	RMSLE = 0.33	RMSE = 400	RMSLE = 0.039

PARAMETER HYPERTUNING

GridSearhCV was imported from scikit learn library using sklearn.model_selection. XGBoost has many parameters and different values were tested for each parameters. The parameters combination with the best RSMLE score was chosen. The best parameter values are listed in the table below.

PARAMETER	DEFINATION	BEST VALUE
Learning rate	Shrinkage of feature weights to prevent overfitting	0.1
N_estimators	Number of trees to be used in model	150
Min_child_weight	Minimum number of instances in each node	0.8
subsample	Percentage of samples used per tree	0.6
Colsample_bytree	Percentage of features used per tree	0.5
Max_depth	Depth of each tree	8
Reg_alpha	Lasso regularisation o weights	0.2

Hypertuning of these parameters was a very lengthy process and took more than 36 hours. However this problem can be tackled by using super computer with higher speeds.

TRAINING TESTING AND DEPLOYING THE MODEL

The final part of the prediction project is to train test and deploy the model on the testing set. For this I have made 3 functions build_model(), train_model(), predict_on_test().

Build_model() function just has an instance of the XGBoost regressor with parameters values that have been hyper tuned. This regressor model is imported from XGBoost library which has to be installed in the windows/anaconda.

Train_model() is used to train the XGBoost model. This is done by using TimeSeriesSplit with number of folds of the dataset =8. TimeSeriesSplit is imported from scikit library using sklearn.model_prediction. Since this project is a time series data sample, it allows train/test indices to split data samples that are observed at fixed time intervals, in train/test sets. In each split, test indices must be higher than before, and thus shuffling in cross validator is inappropriate. Hence, successive training sets are supersets of those that come before them. Working of TimeSeriesSplit is shown in the figure below.

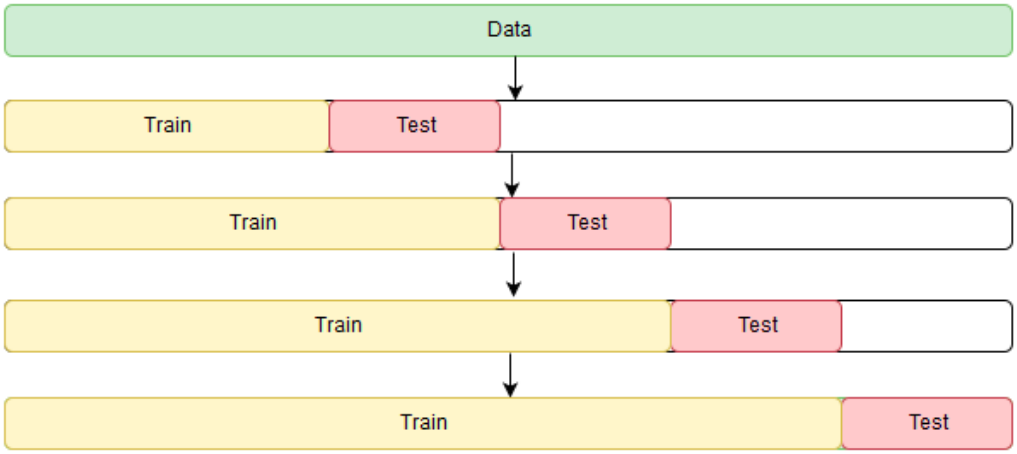


Fig : TimeSeriesSplit

Once we split the datasets into training and validation subsets we find the RMSLE value for both the training and validations for each fold. The results are shown in the image below.

```
====================Training Model============================
Fold-1: Train_RMSLE: 0.45490392972183574, Validation_RMSLE: 0.5280511297518786
Fold-2: Train_RMSLE: 0.4607994636835134, Validation_RMSLE: 0.5252536508045705
Fold-3: Train_RMSLE: 0.4718266942079219, Validation_RMSLE: 0.5219361593380812
Fold-4: Train_RMSLE: 0.4769121674135842, Validation_RMSLE: 0.511563305213142
Fold-5: Train_RMSLE: 0.478274876597487, Validation_RMSLE: 0.5005854416572273
Fold-6: Train_RMSLE: 0.47842986055850273, Validation_RMSLE: 0.4871619431107943
Fold-7: Train_RMSLE: 0.4772062865396963, Validation_RMSLE: 0.47023888955376286
Fold-8: Train_RMSLE: 0.47515987390040315, Validation_RMSLE: 0.5293887229164065
Mean_RMSLE of validation set: 0.5093
Normalized_RMSLE using Standard Deviation:0.8150981223591759

==============Finished Training====================
```

The output above shows that
Root mean squared log error = 0.5093
Normalized root mean squared log error (NRMSLE) = 0.8151
(here the results have been normalised by using standard deviation i.e the RMSLE is divided by the standard deviation of data)
Standard deviation is defined as the spread of values.

MODEL DEPLOYMENT
predict_on_test() - the trained model was then used to make predictions on the testing set. The predictions have been saved to a file named 'submission.csv' using dataframe.to_csv() in the format that was specified in the sample submission

REFERENCES

Kaggle. Forecasting restaurant visitors. URL : https://www.kaggle.com/c/recruit-restaurant-visitor-forecasting

Website : GeekForGeeks .URL https://www.geeksforgeeks.org/python-programming-language/

Website: Stackoverflow. URL https://stackoverflow.com/

Website: TowardsDataScience. URL https://towardsdatascience.com/