Restaurant Visitor Forecasting

Team: AKATSUKI

Aditya Satam
MT2020058
International Institute of Information Technology Bangalore
aditya.satam@iiitb.org

Shubham Ahlawat
MT2020163
International Institute of Information Technology Bangalore
shubham.ahlawat@iiitb.org

Mohit Choudhary
MT2020134
International Institute of Information Technology Bangalore
mohit.choudhary@iiitb.org

***Abstract*— this is a detailed report on our work on time series forecasting problem predicting visitors on Restaurant Visitor Forecasting project hosted on Kaggle.**

We are trying to understand the data given in the competition from two sources Hot Pepper Gourment(hpg) and AirREGI/Restaurant Board(air) and presenting a model using traditional machine learning methods to predict the visitors.

***Index Terms*— Introduction, Dataset, Observation from data, Observation Summary from Dataset, Data Preprocessing and Feature Extraction, Model Selection, Comparison of Models, References.**

INTRODUCTION

Running a thriving local restaurant isn't always as charming as first impressions appear. There are often all sorts of unexpected troubles popping up that could hurt business.

One common predicament is that restaurants need to know how many customers to expect each day to effectively purchase ingredients and schedule staff members. This forecast isn't easy to make because many unpredictable factors affect restaurant attendance, like weather and local competition. It's even harder for newer restaurants with little historical data.

Recruit Holdings has unique access to key datasets that could make automated future customer predictions possible. Specifically, recruit Holdings owns Hot Pepper Gourmet (a restaurant review service), AirREGI (a restaurant point of sales service), and Restaurant Board (reservation log management software).

Recruit Holdings In this competition, we're challenged to use reservation and visitation data to predict the total number of visitors to a restaurant for future date. This information will help restaurants be much more efficient and allow them to focus on creation an enjoyable dining experience for their customers.

DATASET

In this competition, we are provided a time-series forecasting problem centered on restaurant visitors. The data comes from two separate sites:

- Hot Pepper Gourmet (hpg) : similar to Yelp, here users can search restaurants and also make a reservation online
- AirREGI / Restaurant Board (air): similar to Square, a reservation control and cash register system

You must use the reservations, visits, and other information from these sites to forecast future restaurant visitor totals on a given date. The training data covers the dates from 2016 until early (first week) April 2017. The test set covers the mid weeks (second and third weeks) of April 2017. The training and testing set both omit days where the restaurants were closed.

File Descriptions:

This is a relational dataset from two systems. Each file is prefaced with the source (either air_ or hpg_) to indicate its origin. Each restaurant has a unique air_store_id and hpg_store_id. Note that not all restaurants are covered by both systems

and that we have been provided data beyond the restaurants for which we must forecast. Latitudes and Longitudes are not exact to discourage the de-identification of restaurants.

Metadata:

1. Air_reserve.csv

This file contains reservations made in the air system. Note that the reserve_datetime indicates the time when the reservation was created, whereas the visit_datetime is the time in the future where the visit will occur.

a. air_store_id - the restaurant's id in the air system
b. visit_datetime - the time of the reservation
c. reserve_datetime - the time the reservation was made
d. reserve_visitors - the number of visitors for that reservation

2. Hpg_reserve.csv

This file contains reservations made in the hpg system.

a. hpg_store_id - the restaurant's id in the hpg system
b. visit_datetime - the time of the reservation
c. reserve_datetime - the time the reservation was made
d. reserve_visitors - the number of visitors for that reservation

3. Air_store_info.csv

This file contains information about select air restaurants. Column names and contents are self-explanatory.

a. air_store_id
b. air_genre_name
c. air_area_name
d. latitude
e. longitude

Note: latitude and longitude are the latitude and longitude of the area to which the store belongs

4. Hpg_store_info.csv

This file contains information about select Hpg restaurants. Column names and contents are self-explanatory.

a. hpg_store_id
b. hpg_genre_name
c. hpg_area_name
d. latitude
e. longitude

Note: latitude and longitude are the latitude and longitude of the area to which the store belongs

5. Store_id_relation.csv

This file allows you to join select restaurants that have both the air and hpg system.

a. hpg_store_id
b. air_store_id

6. Date_info.csv

This file gives basic information about the calendar dates in the dataset.

a. calendar_date
b. day_of_week
c. holiday_flg - is the day a holiday in Japan

7. Train.csv

This file contains historical visit data for the air restaurants.

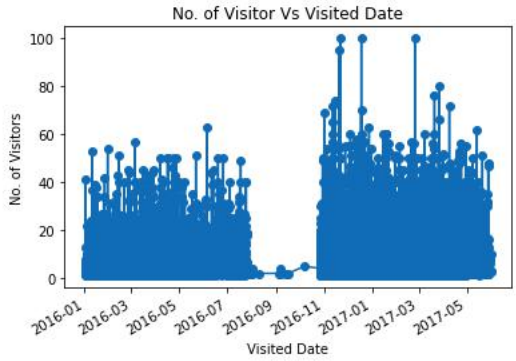
a. air_store_id
b. visit_date - the date
c. visitors - the number of visitors to the restaurant on the date

OBSERVATION

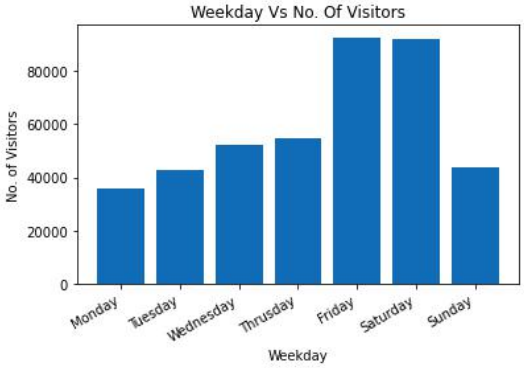
Before getting into preprocessing and feature extraction, it is very important to get to know the distribution of data in order to get better insights while feature selection. We are presenting a few of those here:

Air Reserve Data:

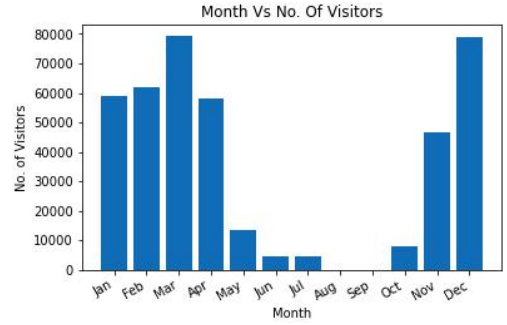
It contains 4 features (air_store_id, visit_datetime, reserve_datetime and reserve_visitors) and 92378 rows.



This shows that there are missing rows in Aug-Sep-Oct 2016. We filled them by mean value or by 75% of data value. Another trend we see here that no. of visitor increases in 2017 as compared to 2016 and there may be few outliers that no. of visitors reaches to 100.



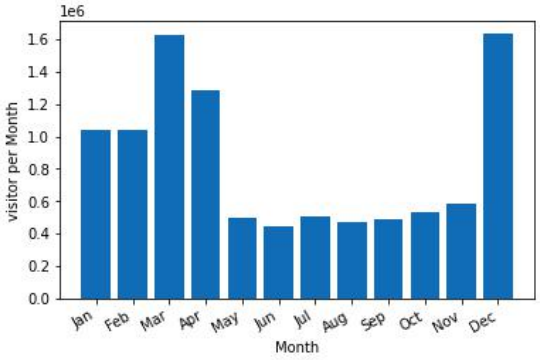
No. of visitors are very low on Monday and increases very much in weekends (Friday, Saturday) but decreases on Sunday. Sunday has fewer visitors because may be Monday is working day and people not for late due to that.



This above graph shows that March and December have the highest no. Of visitors, we also see that there are missing values in Aug-Sept and May-July has very less visitors.

HPG Reserve Data:
There are lots of data compared to Air Store Data, it contains 4 features (hpg_store_id, visit_datetime, reserve_datetime, reserve_visitors) and contains 2000320 rows.

```
plt.xlabel('Month')
plt.ylabel('visitor per Month')
plt.bar(month, visitor_per_month)
plt.gcf().autofmt_xdate()
```



Here also we see that March and December has highest no. Of visitors and here that data is not missing in Aug and Sept month.

```
hpg_resr_data.describe()
```

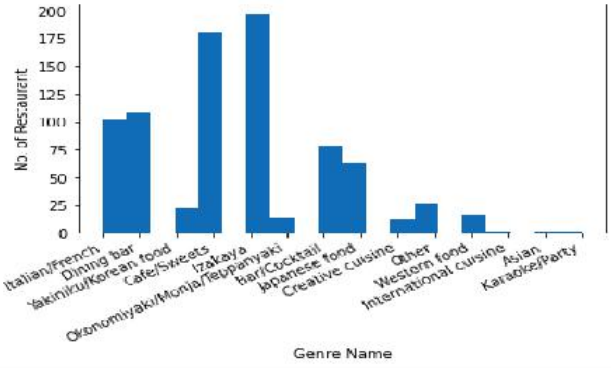
	reserve_visitors
count	2000320.000000
mean	5.073785
std	5.416172
min	1.000000
25%	2.000000
50%	3.000000
75%	6.000000
max	100.000000

The mean and 75% of number of visitors is below 6 but maximum is 100 visitors, it contains outliers and we removed by 75% values.

Air Store Info:
It contains 5 features (air_store_id, air_genre_name, air_area_name, latitude,

longitude), and contains 829 rows. It means there are total of 829 unique restaurants in Air data.

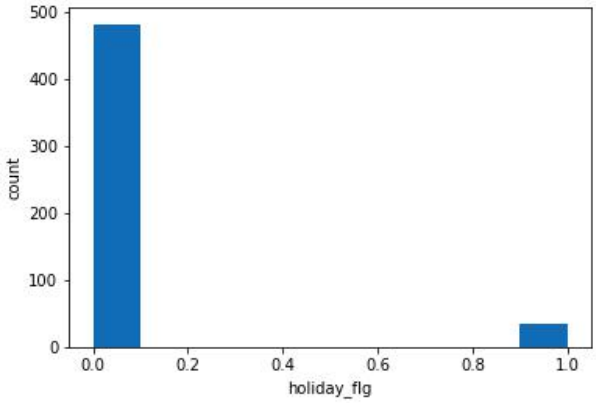
Comparing relation between number of restaurant and genre name:



Here we see that Izakaya has the highest no. of restaurants followed by Cake/Sweets genre, and Asia and Karaoke/Party has lowest no. of restaurants.

HPG Store Info:
It has more no. of unique restaurants than air store info data. It contains 5 features (hpg_store_id, hpg_genre_name, hpg_area_name, latitude, longitude) and 4690 rows. It means it has 4690 unique restaurants data.
In this Japanese genre has more no. of restaurants than others.

Date Info Relation:
It contains 3 features calender_date, day_of_week, holiday_flag and 517 rows of data.
We remove day_of_week feature because we can detect day of week by calender date so it is completely dependent of calender_date feature.



This above graph tells that nearly 6% out of total data has holidays and rest is non-holidays.

Store Id Relation:
It contains 2 features air_store_id and hpg_store_id and contain 150 rows shows the relation between air and hpg dataset. We check the data of one row of store id relation and there are slightly different entries in few rows, so we consider them same and take air_store_id data.

OBSERVATION SUMMARY FROM EDA

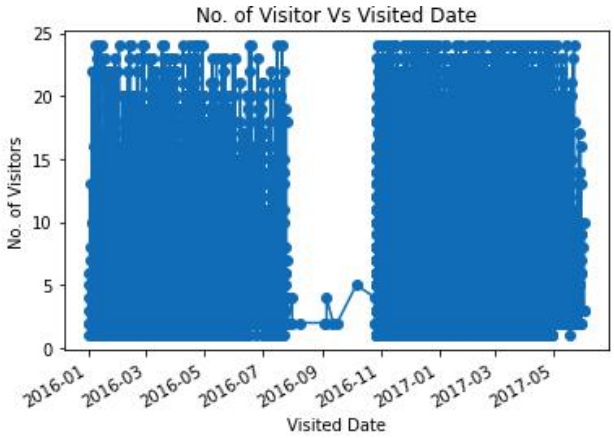
1. Training Dataset Overview: -
   -Total number of unique AIR restaurants: - 829
   -Total restaurants common in AIR and HPG: - 150
   -Total unique genre in AIR restaurants: - 14
   -Total number of AIR restaurant's locations: - 103
   -Average daily visitors: - 20.9
   -Training data duration: - 2016–01–01 to 2017–04–22
2. Test Dataset Overview: -
   -Total unique restaurants: - 821
   -Test data duration: - 2017–04–23 to 2017–05–31
3. No. of visitors increases in 2017 as compared to 2016.
4. Almost 90% of the restaurants have less than 40 visitors per day.
5. The spread of AIR reservations is higher than that of HPG reservations.

6. The number of unreserved visitors is far more than the number of reserved visitors.
7. No. of visitors are very low on Monday and increases very much in weekends (Friday, Saturday) but decreases on Sunday.
8. March and December have highest no. of visitors and May and June has less no. of visitors.
9. The mean and 75% of no. of visitors is below or equal to 6, but max is 100 visitors, so it contains outliers which we removed.
10. There are 14 unique genres, out of which Izakaya is most popular followed by Cake/Sweets genre.
11. International cuisine, Asian, and Karaoke/Party are the least preferred genre having only 0.2% each.
12. We observed hike of no. of visitors at the end of each month.
13. We have more no. of visitors on holiday than on the working day.
14. The highest number of visitors is on evening time.

DATA PREPROCESSING AND FEATURE EXTRACTION

Detecting Outliers:
In Air Reserve Data, there are only 6 rows having visitors greater than 75 and we notice these are outliers so to remove them replace with 75% value.
We set the mean visitors to 25 by calculating through 4 standard deviation mean. All the data which has visitors greater than mean is 1198 only i.e., nearly 1% of original data, so we remove them.

Data after removing outliers in Air Reserve Data:



Now all the data are up to 25 after removing all the outlier, now only we have to fill the missing data of Aug.-Sept 2016. Similarly, in Hpg reserve data we calculate by four standards deviation and detect the outliers which are above of it. Total outlier come is 23736 which is also nearly 1% of the original data, after removing those total entries left is: 1976584.

to get result_finale with latitude,longitude,other air_genre_name unique value as additional columns corresponding to air_store_id in final.

Preparing Data After Outliers Remove:
After removing outliers in both air and hpg reserve data, we join both the data with the help of Store Id relation and remove null and duplicate values, so now it contains nearly 1 Million record. Then we left join the data with air_store_id to get result_finale with latitude, longitude, other air_genre_name unique value as additional columns corresponding to air_store_id in final. Now the data contain 106, 288 records and 19 features.

Work on cuisine dataset: We divided all the cuisine rank-wise with respect to no. of visitors and add this to our final set of features.

Out[26]:

air_genre_name	total_resr_visitors_wrt_cuisine	Cuisine Rank
Izakaya	1364005	1
Cafe/Sweets	1129535	2
Italian/French	642357	3
Dining bar	610573	4
Japanese food	350122	5
Bar/Cocktail	318492	6
Other	155263	7
Yakiniku/Korean food	141541	8
Western food	103790	9
Creative cuisine	86136	10
Okonomiyaki/Monja/Teppanyaki	77591	11
Asian	19534	12
Karaoke/Party	14294	13
International cuisine	8606	14

Work on location dataset: We also divide all the area into ranks named Location Rank and add it to our final set of features.

Out[50]:

latitude	longitude	total_resr_visitors_wrt_location	Location Rank
35.661777	139.704051	29897	1
34.386245	132.455018	29808	2
33.589216	130.392813	27221	3
43.770635	142.364819	22076	4
34.710895	137.725940	20166	5
...	...	...	...
35.708146	139.666288	10	69
35.711877	139.796697	7	70
34.688241	135.187254	6	71
34.799767	135.360073	2	72
35.602125	139.671958	2	72

74 rows × 2 columns

After observing and modifying features of data set, we reached to our final feature set 12 features. On these 12 features we apply various models and extract the results:

```
In [116]: train_x.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 239673 entries, 0 to 239672
Data columns (total 12 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   holiday_flg          239673 non-null  int64
 1   weekdays             239673 non-null  int64
 2   Location Rank        239673 non-null  int64
 3   Cuisine Rank         239673 non-null  int64
 4   reserve_visitors     239673 non-null  int64
 5   latitude             239673 non-null  float32
 6   longitude            239673 non-null  float32
 7   air_restaurant_id    239673 non-null  int64
 8   year                 239673 non-null  int64
 9   month                239673 non-null  int64
 10  day                  239673 non-null  int64
 11  mean_visitors_wrt_id 239673 non-null  int64
```

MODEL SELECTION

1. XGBoost Regression:

It stands for extreme Gradient Boosting. Gradient boosting is an ensemble approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction.
“As the winner of an increasing amount of Kaggle competitions, XGBoost showed us again to be a great all-round algorithm worth having in your toolbox.”

We tried various hyper tuning, we found out that:

Max Depth = 8
Learning rate = .009

It gives score around 73% and Root Mean Square Logarithmic Error = 0.499.

2. Decision Tree Regression:

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

On hyper parameter tuning, we found out

Max Depth = 40
Min_sample_split = 10

Here also we used (log(y)) as it helps a decision tree to pack values in a leaf because the values are “closer” to each other.
It gives score around 61% and Root Mean Square Logarithmic Error = 0.59

3. Random Forest Regression:

A random forest is a Meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
On hyper tuning we found out that

max_depth =45
n_estimators =60

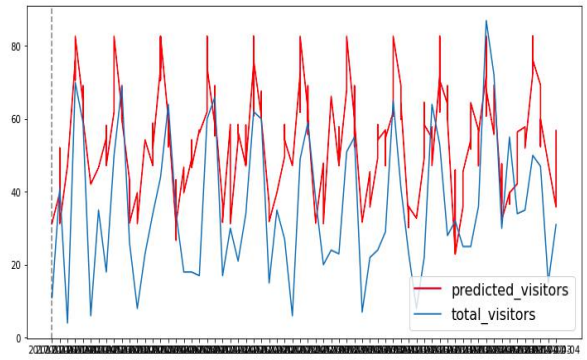
Here also we used log(y).
It gives RMSLE = 52%

4. K-Neighbours Regressor:

Regression based on k-nearest neighbors.

The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.
Init signature: KNeighborsRegressor (n_neighbors=5, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs)
We tried various values of hyper parameter, and conclude that the value of n_neighbours = 20 is giving best results.
First, we separate out the dataset into 2 parts 80% for train data and 20% test data and make a relation between actual visitors and predicted visitors:

<matplotlib.legend.Legend at 0x7f2d29ad1940>



After we train the model and calculate the visitors for the test data, and submission to Kaggle, The Root Mean squared Logarithmic Error is: 0.73887.

It gives score of 55% and gives RMSLE = 0.73.

COMPARISION OF MODELS

1. XGBoost:
Score: 0.73
RMSLE: 0.499

2. Decision Tree Regression:
Score: 0.61
RMSLE: 0.59

3. Random Forest Regression:
Score: 0.64
RMSLE: 0.52

4. K-Neighbour Regressor:
Score: 0.55
RMSLE: 0.73

PROJECT LINK

The link for our project files is available at https://www.kaggle.com/c/restaurant-visitor-forecasting.

CODE LINK

https://github.com/MohitJain11/Restaurant-Visitor-Forecasting.git

REFERENCES

1. https://medium.com/analytics-vidhya/recruit-restaurant-visitor-forecasting-f9ef87ba1073

2. https://pandas.pydata.org/

3. https://www.youtube.com/watch?v=J_LnPL3Qg70

4. https://www.youtube.com/watch?v=xGoRCVryUDk

5. https://www.youtube.com/user/dataschool

6. https://www.youtube.com/channel/UCh9nVJoWXmFb7sLApWGcLPQ