Restaurant Visitor Forecasting
Team Name: Error 404

Riya Ghosh (MT2020125)
Parijat Moulik (MT2020059)
Anjishnu Chakrabarti (MT2020116)

Improvement after previous session:

1. New Addition to Feature Engineering:

Derived columns from latitude and longitude:

Since the latitude and longitude had not been considered in the feature selection previously, three derived columns are added from the latitude and longitude data. Two of them are the relative distance from the maximum latitude and longitude and the other column is summation of latitude and longitude. This helps to give proper mapping for each of the restaurant with respect to location.

2. Model Building:

Random Forest:

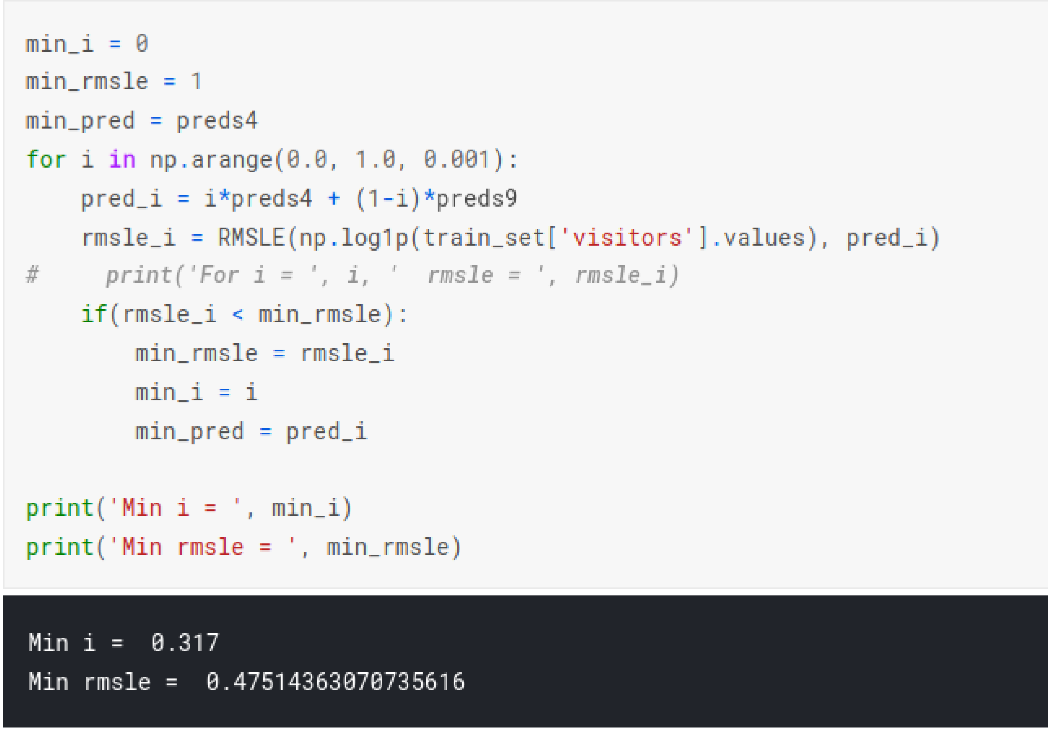
Building a model using a decision tree can tend to overfit. Hence to reduce the variance of the model we are using bagging technique like random forest. Random Forest which is an extension over bagging uses random row sampling and random feature sampling which improves the performance of individual weak learners(decision tree). The accuracy we got using Random forest is 0.72445.

Light Gradient Boosting Model:

Since our training set has more than 2,39,000 entries, XGBoost takes a long time to train the model. Therefore, we selected Light GBM which is much faster than XGBoost regressor. Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks. Since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise. So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. Also, it is surprisingly very fast, hence the word ‘Light’.

Ensemble of XGBoost and LBGM :

In the beginning we made an ensemble model that was an aggregate of XGBoost and LGBM with equal weight. Later on, we used the following function to find out at what proportion should the XGBoost and LGBM model be aggregated to give better performance.



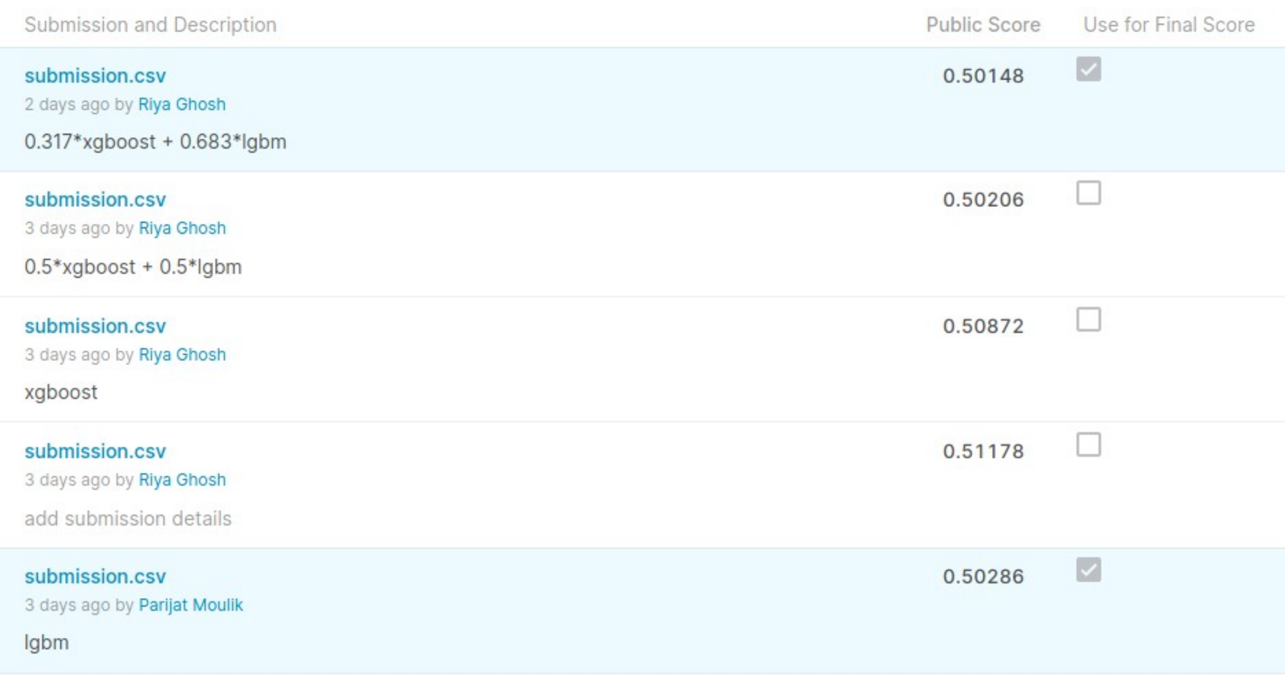
```
min_i = 0
min_rmsle = 1
min_pred = preds4
for i in np.arange(0.0, 1.0, 0.001):
    pred_i = i*preds4 + (1-i)*preds9
    rmsle_i = RMSLE(np.log1p(train_set['visitors'].values), pred_i)
#     print('For i = ', i, '  rmsle = ', rmsle_i)
    if(rmsle_i < min_rmsle):
        min_rmsle = rmsle_i
        min_i = i
        min_pred = pred_i

print('Min i = ', min_i)
print('Min rmsle = ', min_rmsle)
```

```
Min i =  0.317
Min rmsle =  0.47514363070735616
```

From the above discussion, we can conclude that an ensemble containing 31.7% of XGBoost Regressor and 68.3% of LGBM gives the best performance.

3. Model Selection:



Submission and Description	Public Score	Use for Final Score
submission.csv 2 days ago by Riya Ghosh 0.317*xgboost + 0.683*lgbm	0.50148	☑
submission.csv 3 days ago by Riya Ghosh 0.5*xgboost + 0.5*lgbm	0.50206	□
submission.csv 3 days ago by Riya Ghosh xgboost	0.50872	□
submission.csv 3 days ago by Riya Ghosh add submission details	0.51178	□
submission.csv 3 days ago by Parijat Moulik lgbm	0.50286	☑

For the final submission, we have chosen the LGBM model and the ensemble containing 31.7% of XGBoost Regressor and 68.3% of LGBM.