RECRUIT RESTAURANT VISITOR FORECASTING

Team Name: Invincible Predictors

Apoorv Panse
MT2020013
apoorv.panse@iiitb.org

Deepti Chawda
MT2020050
deepti.chawda@iiitb.org

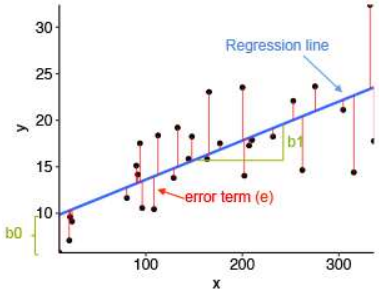
Shahbaz Khan
MT2020160
Shahbaz.khan@iiitb.org

***Abstract*—This is the score improvement report on our work on Recruit restaurant Project. This report contains comparisons on 4 different models that we used to predict the number of visitors for a set of restaurants. Model 1,2 and 3 are same as previous and we added Model 4 which is Gradient Boosting Algorithm XGboost.**

MODEL SELECTIONS AND TRAINING

Model 1: Linear Regression.

- We used scikit-learn library for implementing Linear regression model.
- It uses Ordinary least squares Linear Regression.
- Linear Regression fits a linear model with coefficients to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.



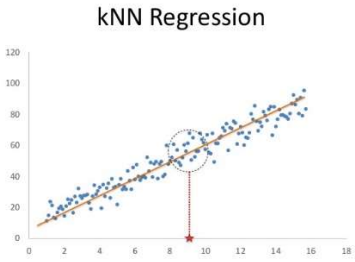

```
#Trying simple Linear Regression model

from sklearn.linear_model import LinearRegression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
y_preds=lr_model.predict(X_test)
rmsle(y_test, y_preds)
```

```
0.5369255002063672
```

Model 2: KNeighbors Regression.

- 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.



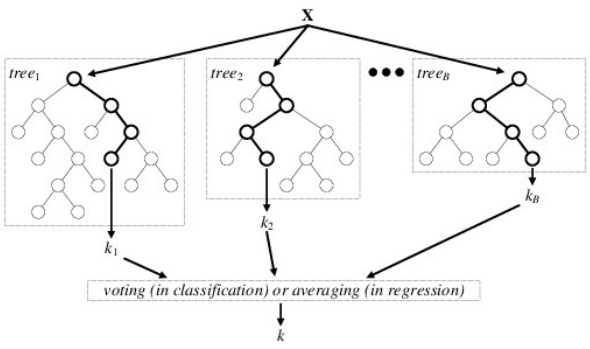

```
#Trying KNeighbors Regression model

from sklearn.neighbors import KNeighborsRegressor
knr_model = KNeighborsRegressor(n_jobs=-1, n_neighbors=10)
knr_model.fit(X_train, y_train)
y_preds=knr_model.predict(X_test)
rmsle(y_test, y_preds)
```

```
0.5398689672131062
```

Model 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.



```
#Trying Random Forest Regressor Regression model

from sklearn.ensemble import RandomForestRegressor

rfrmodel = RandomForestRegressor(n_estimators=200, min_samples_leaf=5,
                                 min_samples_split=15,
                                 max_features=1, n_jobs=-1,
                                 )

rfrmodel.fit(X_train, y_train)
y_preds=rfrmodel.predict(X_test)
rmsle(y_test, y_preds)
```

```
0.5289086869919264
```

Model 4: XGboost Regression

- XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible
- Implements machine learning algorithms under the Gradient Boosting framework.
- XGBoost provides a parallel tree boosting.

```
#Trying XGBoost Regression model

from xgboost import XGBRegressor
xgbmodel = XGBRegressor(
                        max_depth =16,
                        learning_rate=0.1,
                        n_estimators=20,
                        subsample=0.4,
                        colsample_bytree=0.8,
                        seed=5
                       )
xgbmodel.fit(X_train,y_train)

y_preds_xgb=xgbmodel.predict(X_test)
rmsle(y_test,y_preds)
```

```
0.5188255921668121
```

RMSLE ON VARIOUS ALGORITHMS

1	Linear Regression	0.536
2	KNeighbors Regression	0.539
3	Random Forest Regression	0.528
4	XGBoost Regresssion	**0.518**

As XGBoost gave better results, we selected that model for final training. For final training, the entire train data set was passed for model training and Submission dataset was converted in the same format by feature engineering, and Submission data became the test set.

The final submission score in Kaggle was: **0.526**

CONCLUSION AND ACKNOWLEDGEMENT

Gradient Boosting model worked well in improving our score from **0.538** (previous best score) **to 0.526** (current best score). We would like to thank Arjun Verma for suggestion us to use XGboost for score improvement which turned out to be useful.

REFERENCES FOR MODELS USED

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[4] https://xgboost.readthedocs.io/en/latest/