TAXI FARE PREDICTOR REPORT

Ananthoju Pranav Sai
R.No : AI20BTECH11004

For each of the following models, I used 300000 rows of total training data to train them.

The best 2 scores of my models are :

- **MSE : 3.72907**
  Regressor : RandomForestRegressor()
  N_estimators : 200
  Run time : 6-7 min

- **MSE : 3.87664**
  Regressor : XGBRegressor()
  Run time : 3.5sec

- **MSE : 3.94470**
  Regressor : GradientBoostingRegressor()
  N_estimators : 1000
  Run time : 5-6 min

Few other models which I tried but were not included in the final notebook.

- **MSE : 3.73412**
  Trained using 20M rows of the training set.
  Regressor : KNNRegressor()
  N_neighbors : 7
  Run time : 3-4 min

- **MSE : 3.85853**
  Trained using 1M rows of the training set.
  Regressor : GradientBoostingRegressor()
  N_estimators : 5000
  Max_depth : 6
  Run time : ~28min

Using a lesser number of training samples (100,000-400,000) KNNregressor() gave higher (>4 MSE) errors whereas ensemble regressors were working fine but they were relatively slower than KNN.

Using a higher number of training samples (1M-5M) KNNregressor() performed well (<4 MSE) but when the same amount of data is used to train ensemble methods the error was lesser but the run time was very high and few times the kernel died too. So I preferred to use a lesser number of training samples to train the models.

The main reason why random forest performed well was it being an ensemble method. Boosting regressors such as Gradient Boosting and XGBoost also performed better than KNN because of the same reason. KNN seems promising because Taxi Fares for the 2 trips whose starting points and ending points are near would be similar but because of using only a few samples in training, KNN was not able to give better scores whereas the same KNN if used with a higher number of samples it works well. Whereas in ensemble methods changing the **n_estimators** gives better scores even if few samples are used.