Intracity Fare problem:

In train data we had the ID for each trip and their corresponding features like Timestamp: starting time , the starting and destination longitude ,the distance and so on . The label was the fare which was to be calculaed by the regression problem.

It was supervised learning.

Different modes of transport were given and the corresponding luggage weight and wait time seemed to differ a lot . After close inspection we found out that the Taxi Ac and Taxi Non Ac only had values for Total Luggage weight and Wait Time and this thing was later exploited .

Packages used :

1)Numpy

2)Pandas

3)Sklearn

4)Xgboost

5)Python CSV

6)Matplotib

The difference here from 1st problem was that it had missing values:

So we had to tackle it therefore we used **pandas library** and we tried three different approaches:

1) Filling those empty values with -mean/ median / interpolated value

while that of luggage and wait time as 0.0 for missing values

2) Use of **IMPUTER** under sklearn.preprocessing

3) Filling of values with np.nan and use of XGBoost which works well with sparsely given data also and efficiently tackles it well

An Additional column was added called as “DIST” which held the real distances between two places and was calculated as :

SQRT( (destination\_latitude-starting\_latitude)^2 + (destination\_longitude-starting\_longitude )^2 ) \*111.321

The above was the euclidean distance formula and the multiplication factor was =(distance between two latitude minutes)=111.321 km or 69.1 miles

And for round trip it had to be doubled .

The FOLLOWING PROBLEMS WERE TACKLED:

->Now the problem was that the timestamp had dtpye of object and had to be changed into float for fitting into regressor.

So we used the “datetime” package and pandas library to work on it and convert into float which in nanoseconds.

-> The vechicle type was that of string and had to be changed into integer value so we used **Label Encoding**

**->**The DIST value had to formatted and changed into float for use

Now we used Histograms to see by matplotlib.pyplot as how the data was spread and found that : TRAIN and TEST had lots of similariteis between them

The first Algorithm was **RandomForestRegressor** then **XGBRegressor**

**The** following algortihms were also tested: (the r2\_score based on cross validation with split =0.3)

1)XGBoostRegressor [97.36]

2)RandomForestRegressor [97.83]

3)KNearestNeighborsRegressor [41.81]

4)Linear Regression [44.82]

5)Lasso [44.2]

6)GradientBoostingRegressor[ 93.97]

7)DecisionTreeRegressor[ 99.99]

Out of which GradientBoost and DecisionTree were overfitting hence we foucsed more on Xgboost Regressor

We applied GridSearchCV for best parameters but was taking too long so we killed it and then manually tuned the **hyperparameters** using the R2\_score and Mean absolute error

Also we plotted based on feature importance and histograms,scatter plots , correlation also.