UBER DATA ANALYSIS

Price prediction using Machine Learning

Uber uses a mixture of internal and external data to estimate fares. Uber calculates fares automatically using street traffic data, GPS data and its own algorithms that make alterations based on the time of the journey. It also analyses external data like public transport routes to plan various services.so we are here to predict the price of uber based on different aspects.

Research paper on UBER-DATA-ANALYSIS

Using MACHINE LEARNING

As a project work for Course:

MACHINE LEARNING FOUNDATION (INT 247)

Submitted by:

Name	: R. Suresh Ram (B54)
Registration Number	: 11908772
Program	: CSE B.TECH
Semester	: Sixth
School	: School of Computer Science and Engineering
Name of the University	: Lovely Professional University
Date of submission	: 15th April, 2022

Abstract:

Uber was founded just 13 years ago, and it was already one of the fastest-growing companies in the world. In Boston, UberX claims to charge 30% less than taxis – a great way to get customers' attention.

Nowadays, we see applications of Machine Learning and Artificial Intelligence in almost all the domains so we try to use the same for Uber cabs price prediction. In this project, we did experiment with a real-world dataset and explore how machine learning algorithms could be used to find the patterns in data.

We mainly discuss about the price prediction of different Uber cabs that is generated by the machine learning algorithm. Our problem belongs to the regression supervised learning category. We use different machine learning algorithms, for example, Linear Regression, Decision Tree, Random Forest Regressor, and Gradient Boosting Regressor but finally, choose the one that proves best for the price prediction.

We must choose the algorithm which improves the accuracy and reduces over fitting. We got many experiences while doing the data preparation of Uber Dataset of Boston of the year 2018. It was also very interesting to know how different factors affect the pricing of Uber cabs.

1. Introduction

1.1 Overview:

Uber Technologies, Inc., commonly known as Uber, was a ride-sharing company and offers vehicles for hire, food delivery (Uber Eats), package delivery, couriers, freight transportation, and, through a partnership with Lime, electric bicycle and motorized scooter rental. It was founded in 2009 by Travis Kalanick and Garrett Camp, a successful technology entrepreneur. After selling his first startup to eBay, Camp decided to create a new startup to address San Francisco's serious taxi problem.

Together, the pair developed the Uber app to help connect riders and local drivers. The service was initially launched in San Francisco and eventually expanded to Chicago in April 2012, proving to be a highly convenient great alternative to taxis and poorly-funded public transportation systems. Over time, Uber has since expanded into smaller communities and has become popular throughout the world. In December 2013, USA Today named Uber its tech company of the year.

In Supervised learning, we have a training set and a test set. The training and test set consists of a set of examples consisting of input and output vectors, and the goal of the supervised learning algorithm is to infer a function that maps the input vector to the output vector with minimal error. We applied machine learning algorithms to make a prediction of Price in the Uber Dataset of Boston. Several features will be selected from 55 columns. Predictive analysis is a procedure that incorporates the use of computational methods to determine important and useful patterns in large data.

1.2 Objective:

The objective is to first explore hidden or previously unknown information by applying exploratory data analytics on the dataset and to know the effect of each field on price with every other field of the dataset. Then we apply different machine learning models to complete the analysis. After this, the results of applied machine learning models were compared and analyzed on the basis of accuracy, and then the best performing model was suggested for further predictions of the label 'Price'.

2. Literature Review

As we are researching on Uber and found what different researchers had done. So, they do research on the Uber dataset but on different factors. The rise of Uber as the global alternative has attracted a lot of interest recently. Our work on Uber's predicting pricing strategy is still relatively new. In this research, "Uber Data Analysis" we aim to shed light on Uber's Price. We are predicting the price of different types of Uber based on different factors. Some of the other factors that we found in other researches are:

Abel Brodeurand & Kerry Nield (2018) analyses the effect of rain on Uber rides in New York City after entering Uber rides in the market in May 2011, passengers and fare will decrease in all other rides such as taxi-ride. Also, dynamic pricing makes Uber drivers compete for rides when demand suddenly increases, i.e., during rainy hours. On increasing rain, the Uber rides are also increasing by 22% while the number of taxi rides per hour increases by only 5%. Taxis do not respond differently to increased demand in rainy hours than non-rainy hours since the entrance of Uber.

Some papers take a comparison between the iconic yellow taxi and its modern competitor, Uber. (Vsevolod Salnikov, Renaud Lambiotte, Anastasios Noulas, and Cecilia Mascolo, 2014) identify situations when UberX, the cheapest version of the Uber taxi service, tends to be more expensive than yellow taxis for the same journey.

Our observations show that it might be financially advantageous on average for travellers to choose either Yellow Cabs or Uber depending on the duration of their journey. However, the specific journey they are willing to take matters.

3. MACHINE LEARNING

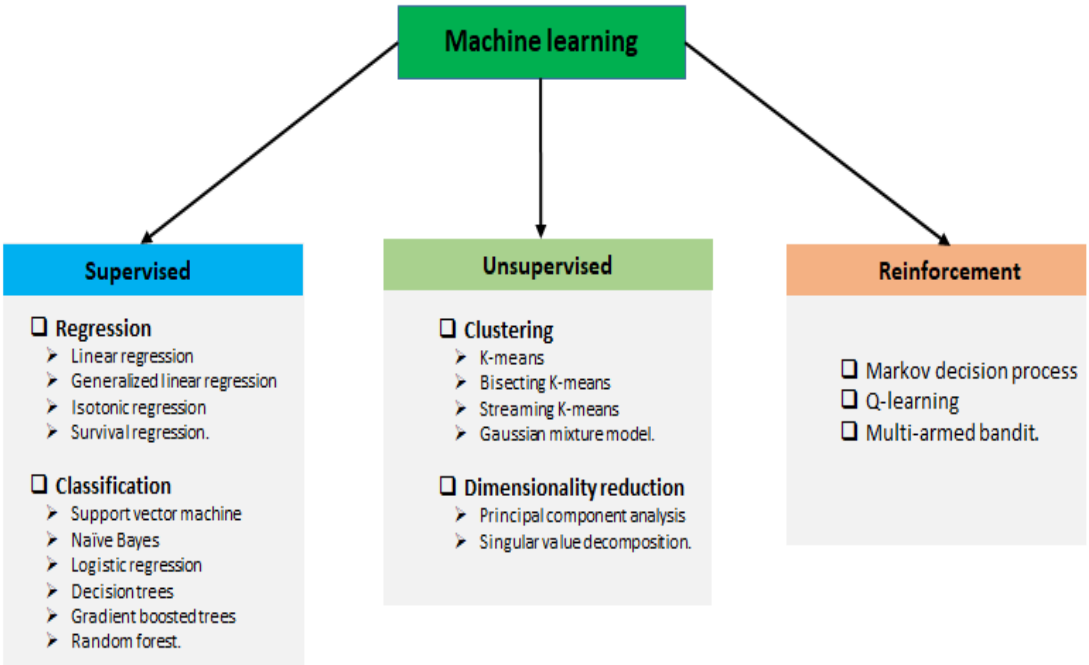
3.1 What is Machine Learning?

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.

Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

3.2 Types of Learning Algorithms

The types of machine learning algorithms differ in their approach, the type of data they input, and the type of task or problem that they are intended to solve.



3.2.1 Supervised learning:

Supervised learning is when the model is getting trained on a labelled dataset. The labelled dataset is one that has both input and output parameters. Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range.

3.2.2 Unsupervised learning:

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labelled, classified, or categorized.

3.2.3 Reinforcement learning:

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment to maximize some notion of cumulative reward. In this learning, system is provided feedback in terms of rewards and punishments as it navigates its problem space.

4. Methodology

4.1 Description of the dataset

The data we used for our project was provided on the www.kaggle.com website. The original dataset contains 693071 rows and 57 columns which contain the data of both Uber and Lyft. The dataset has many fields that describe us about the time, geographic location, and climatic conditions when the different Uber cabs opted.

Data has 3 types of data-types which were as follows: - integer, floats, and objects. The dataset is not complete which means we have also null values in a column named price of around 55095.

The attributes are

Id ,timestamp ,hour ,day ,month ,date time ,time zone ,source ,destination ,cab_type ,precipintensityMax ,uvindexTime , temperature ,temperatureMin Time ,temperatureMax Time ,apparentTemperatureMin ,apparentTemperatureMax , etc.

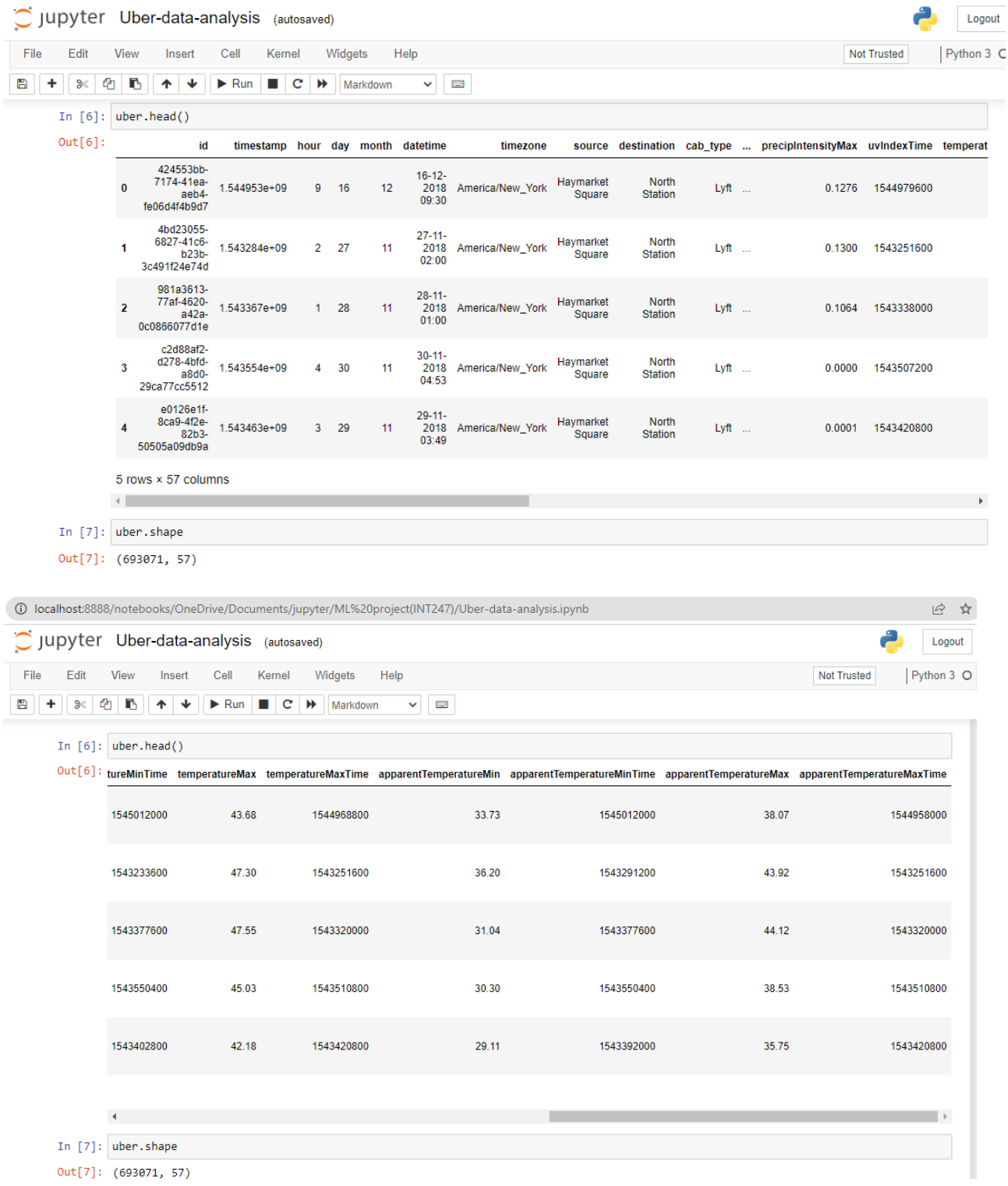


Fig: 4.1 Data head

4.2 Data Visualization:

Data visualization is a graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

For the same purpose, we have to import matplotlib and seaborn library and plot different types of charts like strip plot, scatter plot, and bar chart.

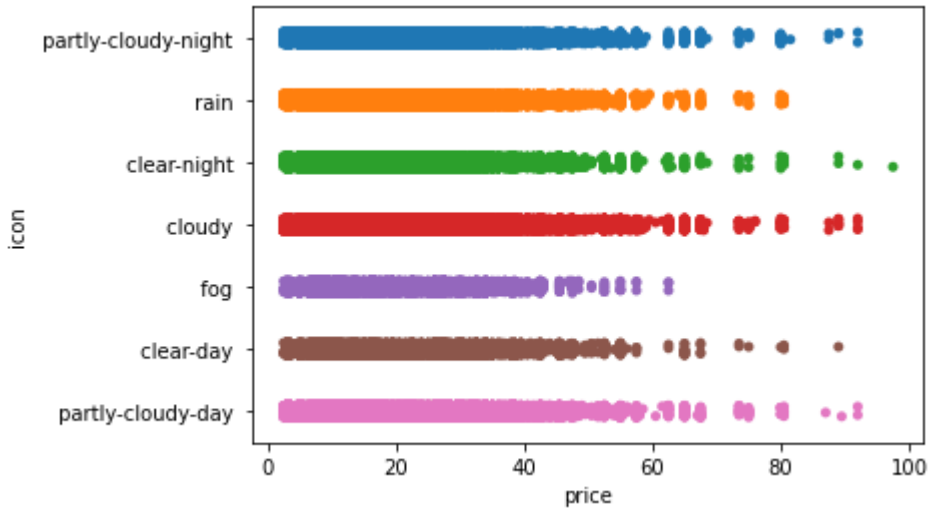


Fig: 4.2.1 Strip-plot between Name and Price

From the above chart, it was clear that Shared trip was cheapest among all and BlackSuv was most expensive. UberX and UberPool have almost same prices and Lux has moderate price. There is no graph for taxi which reveals that in the dataset there were no values of taxi was given.

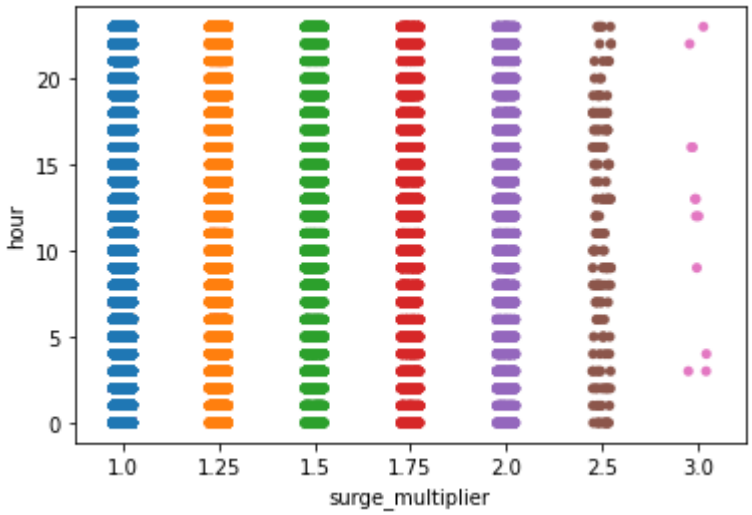


Fig. 4.2.2 Strip-plot between hour and surge_multiplier

From the above graph we can say that after 2.0 the surge_multiplier and hour are decreasing rapidly.at 2.5 it is quiet low but at 3.0 it is so low.

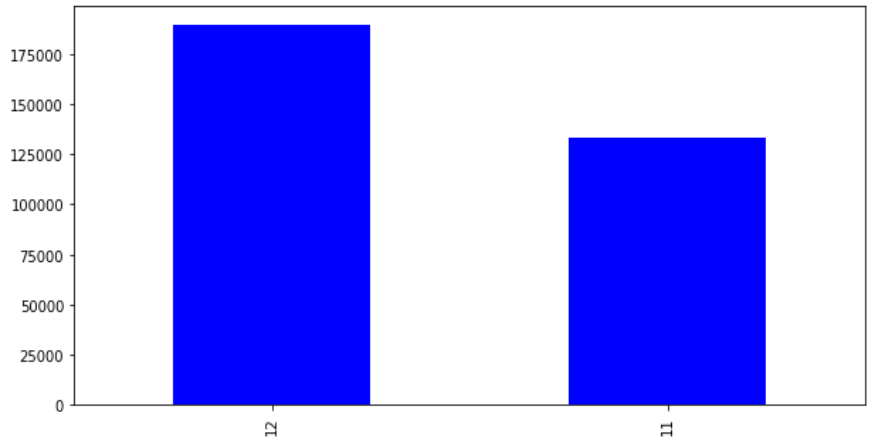


Fig: 4.2.3 Bar chart of month

From the above bar chart, it was clear that the data consists of all the information of only two months that is November and December.

4.3 Label encoding:

Our data is a combination of both categorical variables and continuous variables, most of the machine learning algorithms will not understand, or not be able to deal with categorical variables. Meaning, machine learning algorithms will perform better when the data is represented as a number instead of categorical. Hence label encoding comes into existence. Label Encoding refers to converting the categorical values into the numeric form to make it machine-readable. So we did label encoding as well as class mapping to get to know which categorical value is encoded into which numeric value.

```
In [20]: uber['name']= label_encoder.fit_transform(uber['name'])

         print("Class mapping of Name: ")
         for i, item in enumerate(label_encoder.classes_):
             print(item, "-->", i)
```

```
Class mapping of Name:
Black --> 0
Black SUV --> 1
Lux --> 2
Lux Black --> 3
Lux Black XL --> 4
Lyft --> 5
Lyft XL --> 6
Shared --> 7
Taxi --> 8
UberPool --> 9
UberX --> 10
UberXL --> 11
WAV --> 12
```

4.4 Filling NAN values:

To check missing values in Pandas DataFrame, we use a function isnull(). So we find that the price column in our dataset consists of 55095 Nan values. Now to fill these null values we use the fillna() function. We fill missing values with the median of the remaining dataset values and convert them to integer because price cannot be given in float. Now for the visualization purpose, we make a bar chart of the value count of price.

4.5 RFE (Recursive Feature Elimination):

Feature selection is an important task for any machine learning application. This is especially crucial when the data has many features. The optimal number of features also leads to improved model accuracy. So we use RFE for feature selection in our data.

RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is wrapped by RFE, and used to help select features. This is in contrast to filter-based feature selections that score each feature and select those features with the largest score.

On applying RFE in our dataset with Linear Regression model first we divide our dataset into dependent (features) and independent (target) variables then split it into train and test after that we found different accuracies in different number of features (k value) as follows:

Serial No.	No. of Feature (K)	Prediction
1	40	0.8050662132

Table 4.5: RFE Accuracy Table

4.6 Drop useless columns:

After applying RFE we get our 25 best features but still, there are many features which do not affect the price directly so we drop those features according to it. And eight features remained in our dataset. We use a method called drop() that removes rows or columns according to specific column names and corresponding axis.

4.7 Binning:

Many times we use a method called data smoothing to make the data proper. During this process, we define a range also called bin and any data value within the range is made to fit into the bin. This is called the binning. Binning is used to smoothing the data or to handle noisy data.

So after dropping useless features, some features are not in range so to make all the features in the same range we apply binning and get our final dataset which is further used for modeling.

So the eight attributes which we left are month, source, destination, product id, name, surge_multiplier, icon, uvIndex.

```
new_uber.head()
```

	month	source	destination	product_id	name	surge_multiplier	icon	uvIndex
0	12	5	7	8	7	1.0	5	0
1	11	5	7	12	2	1.0	6	0
2	11	5	7	7	5	1.0	1	0
3	11	5	7	10	4	1.0	1	0
4	11	5	7	11	6	1.0	5	0

Fig: 4.7 new data head

4.8 Modeling:

The process of modeling means training a machine-learning algorithm to predict the labels from the features, tuning it for the business needs, and validating it on holdout data. When you train an algorithm with data it will become a model. One important aspect of all machine learning models is to determine their accuracy. Now to determine their accuracy, one can train the model using the given dataset and then predict the response values for the same dataset using that model and hence, find the accuracy of the model.

In this project, we use Scikit-Learn to rapidly implement a few models such as Linear Regression, Decision Tree and Random Forest.

4.8.1 Linear regression:

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous in the range such as salary, age, price, etc. It is a statistical approach that models the relationship between input features and output. The input features are called the independent variables, and the output is called a dependent variable. Our goal here is to predict the value of the output based on the input features by multiplying it with its optimal coefficients.

4.8.2 Decision tree:

Decision tree is a supervised learning algorithm which can be used for both classification and regression problem. This model is very good at handling tabular data with numerical or categorical features. It uses a tree-like structure flow chart to solve the problem. A decision tree is arriving at an estimate by asking a series of questions to the data, each question narrowing our possible values until the model gets confident enough to make a single prediction.

4.8.3 Random forest:

Random forest is a supervised learning algorithm which can be used for both classification and regression problem. It is a collection of Decision Trees. In general, Random Forest can be fast to train, but quite slow to create predictions once they are trained. This is due because it has to run predictions on each tree and then average their predictions to create the final prediction.

Serial No.	Models	prediction
1	Linear Regression	0.747545073
2	Decision Tree	0.961791729
3	Random Forest	0.962269474

Table: 4.8 Model Accuracy Table

4.9 Testing:

In Machine Learning the main task is to model the data and predict the output using various algorithms. But since there are so many algorithms, it was really difficult to

choose the one for predicting the final data. So we need to compare our models and choose the one with the highest accuracy.

Machine learning applications are not 100% accurate, and approx never will be. There are some of the reasons why testers cannot ignore learning about machine learning. The fundamental reason is that these applications learning limited by data they have used to build algorithms. For example, if 99% of emails aren't spammed, then classifying all emails as not spam gets 99% accuracy through chance. Therefore, you need to check your model for algorithmic correctness. Hence testing is required. Testing is a subset or part of the training dataset that is built to test all the possible combinations and also estimates how well the model trains. Based on the test data set results, the model was fine-tuned.

Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) are used to evaluate the regression problem's accuracy. These can be implemented using sklearn's mean_absolute_error method and sklearn's mean squared_error method.

4.9.1 Mean Absolute Error (MAE):

It is the mean of all absolute error. MAE (a range from 0 to infinity, lower is better) is much like RMSE, but instead of squaring the difference of the residuals and taking the square root of the result, it just averages the absolute difference of the residuals. This produces positive numbers only and is less reactive to large errors. MAE takes the **average** of the error from every sample in a dataset and gives the output.

Hence, **MAE = True values – Predicted values**

4.9.2 Mean Squared Error (MSE):

It is the mean of square of all errors. It is the sum, overall the data points, of the square of the difference between the predicted and actual target variables, divided by the number of data points. MSE is calculated by taking the average of the square of the difference between the original and predicted values of the data.

4.9.3 Root Mean Squared Error (RMSE):

RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model. RMSE (ranges from 0 to

infinity, lower is better), also called Root Mean Square Deviation (RMSD), is a quadratic-based rule to measure the absolute average magnitude of the error.

```
In [76]: from sklearn import metrics
         print('MAE :'," ", metrics.mean_absolute_error(yy_test,prediction))
         print('MSE :'," ", metrics.mean_squared_error(yy_test,prediction))
         print('RMAE :'," ", np.sqrt(metrics.mean_squared_error(yy_test,prediction)))

         MAE :   5.312289434765357
         MSE :   49.74074234390218
         RMAE :   7.052711701459388
```

Serial No.	Models	prediction
1	Mean Absolute Error	5.312289434
2	Mean Squared Error	49.740742343
3	Root Mean Absolute Error	7.052711701

Table: 4.9 Error table for Linear Regression

5. PRICE PREDICTION FUNCTION:

After finding the errors for both linear regression and random forest algorithm, we build a function name “predict_price” whose purpose is to predict the price by taking 4 parameters as input. These four parameters are cab name, source, surge multiplier, and icon (weather). As the dataset train on the continuous values and not on categorical values, these values are also passed in the same manner i.e. in integer type. We create a manual for users which gives instructions about the input like what do you need to type for a specific thing and in which sequence.

We use random forest model in our function to predict the price. First, we search for all the desired rows which have the input cab name and extract their row number. After then we create an array x which is of the length of the new dataset and it’s initially all values are zero. After creating the blank array we assign the input values of source, surge multiplier, and icon to the respected indices. Following it we check the count of all desired rows if it was greater than zero or not. If the condition gets

true, we assign the value 1 to the index of x array and return the price using the predict function with trained random forest algorithm.

It somehow works like a hypothesis space because it gives an output for any input from input the space.

```
In [297]: def predict_price(name,source,surge_multiplier,icon):
              loc_index = np.where(new_uber.columns==name)[0]

              x = np.zeros(len(new_uber.columns))
              x[0] = source
              x[1] = surge_multiplier
              x[2] = icon
              if loc_index >= 0:
                  x[loc_index] = 1

              return random.predict([x])[0]
```

Here we used the Numpy function for giving the inputs of different aspects for the uber to make the price prediction.

1st input → name

2nd input →source

3rd input →surge_multiplier

4th input →icon

Now we will write the predict function for the price

```
In [298]: pre= random.predict(xx_test)
```

Follow these instructions before predicting the price:

- cab_name: Black SUV --> 0 , Lux --> 1 , Shared --> 2 , Taxi --> 3 , UberPool --> 4 , UberX --> 5

- Source: Back Bay --> 0 , Beacon Hill --> 1 , Boston University --> 2 , Fenway --> 3 , Financial District --> 4 , Haymarket Square --> 5 , North End --> 6 , North Station --> 7 , North eastern University --> 8 , South Station --> 9 , Theatre District --> 10 , West End --> 11

- Surge_multiplier: Enter Surge Multiplier value from 0 to 4

- Icon: clear-day --> 0 , clear-night --> 1 , cloudy --> 2 , fog --> 3 , partly-cloudy-day --> 4 , partly-cloudy-night --> 5 , rain --> 6

**predict_price (cab_name, source, surge_multiplier, icon) **

Now let's check the price by giving some inputs

```
In [299]: predict_price(1 , 3, 2, 0)

<ipython-input-297-884828bfaf47>:8: DeprecationWarning: The truth value of an empty array
future this will result in an error. Use `array.size > 0` to check that an array is not e
  if loc_index >= 0:

Out[299]: 15.902894847842212
```

So in this we have given the inputs as:

Name → 1: Lux

Source → 3: Fenway

Surge_multiplier → 2

Icon → 0: clear day

So for overall for these inputs we get the price as 15.9.

CONSLUSION

Before working on features first we need to know about the data insights which we get to know by EDA. Apart from that, we visualize the data by drawing various plots, due to which we understand that we don’t have any data for taxi’s price, also the price variations of other cabs and different types of weather. Other value count plots show the type and amount of data the dataset has. After this, we convert all categorical values into continuous data type and fill price Nan by the median of other values. Then the most important part of feature selection came which was done with the help of recursive feature elimination. With the help of RFE, the top 40 features were selected. Among those 40 features still, there are some features which we think are not that important to predict the price so we drop them and left with 16 important columns.

We apply three different models on our remaining dataset among which Decision Tree, Random Forest, prove best with 96%+ accuracy on training for our model. This means the predictive power of all these three algorithms in this dataset with the chosen features is very high but in the end, we go with random forest because it does not prone to overfitting and design a function with the help of the same model to predict the price.

REFERENCES

1. Abel Brodeurand & Kerry Nield (2018) An empirical analysis of taxi, Lyft and Uber rides: Evidence from weather shocks in NYC

2. Junfeng Jiao (2018) Investigating Uber price surges during a special event in Austin, TX

3. Anna Baj-Rogowska (2017) Sentiment analysis of Facebook posts: The Uber Case

4. Anastasios Noulas, Cecilia Mascolo, Renaud Lambiotte, and Vsevolod Salnikov (2014) OpenStreetCab: Exploiting Taxi Mobility Patterns in New York City to Reduce Commuter Costs.