UBER-DATA-ANALYSIS

**2022**

***PRICE PREDICTION USING MACHINE LEARNING***

4/15/2022

R. Suresh ram



MACHINE LEARNING PROJECT REPORT

**UBER-DATA-ANALYSIS**



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

**DECLARATION**

I, hereby declare that the work which is being presented in the B.tech. Project **“Uber Data Analysis”**, in partial fulfilment of the requirements for the award ofthe **Bachelor of Technology** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of **Lovely Professional University**, Punjab, is an authentic record of my own work carried under the supervision of **Dr. Sagar pande.**

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech Machine Learning Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

We owe special debt of gratitude to Dr. Sagar pande for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance.

Have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

**R. Suresh ram**

INDEX

1. Abstract

2. Introduction

3. Literature review

4. Packages used

5. Machine learning concepts

6. Proposed work and Implementation

* 6.1 Data preparation
* 6.2 Exploratory data analysis
* 6.3 Data visualisation
* 6.4 Bar plots
* 6.5 Label Encoding
* 6.6 Filling NAN values

7. ML Model

* 7.1 Creating model
* 7.2 Recursive Feature Selection
* 7.3 Drop useless columns
* 7.4 Binning
* 7.5 Modelling
* 7.6 Regression models
* Linear regression
* Decision tree regression
* Random forest regression
* 7.7 Testing
* Mean Squared Error (MSE),
* Mean Absolute Error (MAE),
* Root Mean Squared Error (RMSE)

8. Price Prediction Analysis

9. Conclusion

10. Reference

**ABSTRACT**

Uber was founded just eleven 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.

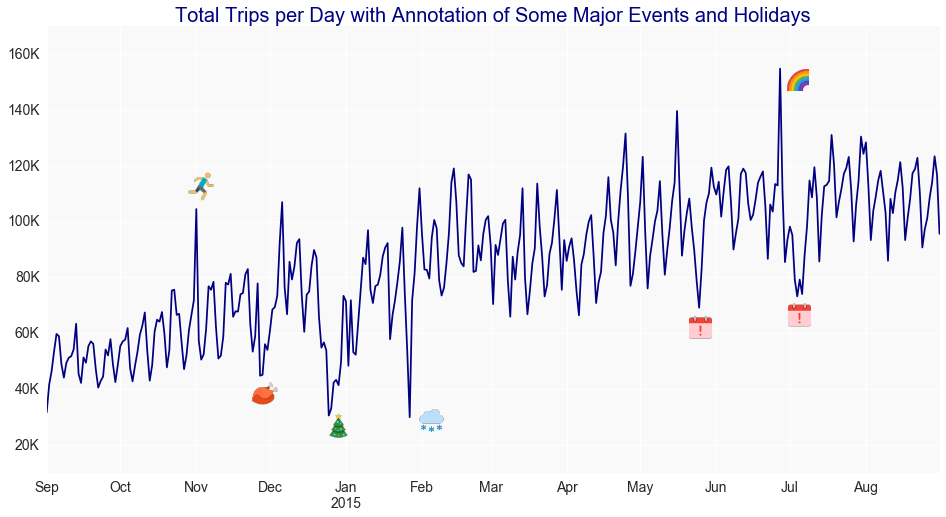
We must choose the algorithm which improves the accuracy and reduces over fitting. We got many experiences while doing the data preparation of Uber Dataset of Boston of the year 2018. It was also very interesting to know how different factors affect the pricing of Uber cabs.

**INTRODUCTION**

Uber Technologies, Inc., commonly known as Uber, was a ride-sharing company and offers [vehicles for hire](https://en.wikipedia.org/wiki/Vehicles_for_hire), [food delivery](https://en.wikipedia.org/wiki/Food_delivery) ([Uber Eats](https://en.wikipedia.org/wiki/Uber_Eats)), [package delivery](https://en.wikipedia.org/wiki/Package_delivery), [couriers](https://en.wikipedia.org/wiki/Courier), [freight transportation](https://en.wikipedia.org/wiki/Freight_transport), and, through a partnership with [Lime](https://en.wikipedia.org/wiki/Lime_(transportation_company)), [electric bicycle](https://en.wikipedia.org/wiki/Electric_bicycle) and [motorized scooter](https://en.wikipedia.org/wiki/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](http://www.tc.umn.edu/~ssen/IDSC6050/Case15/Group15_index.html).

Together, the pair developed the Uber app to help connect riders and local drivers. The service was [initially launched in San Francisco](http://blog.uber.com/tag/history/) 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](https://en.wikipedia.org/wiki/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.

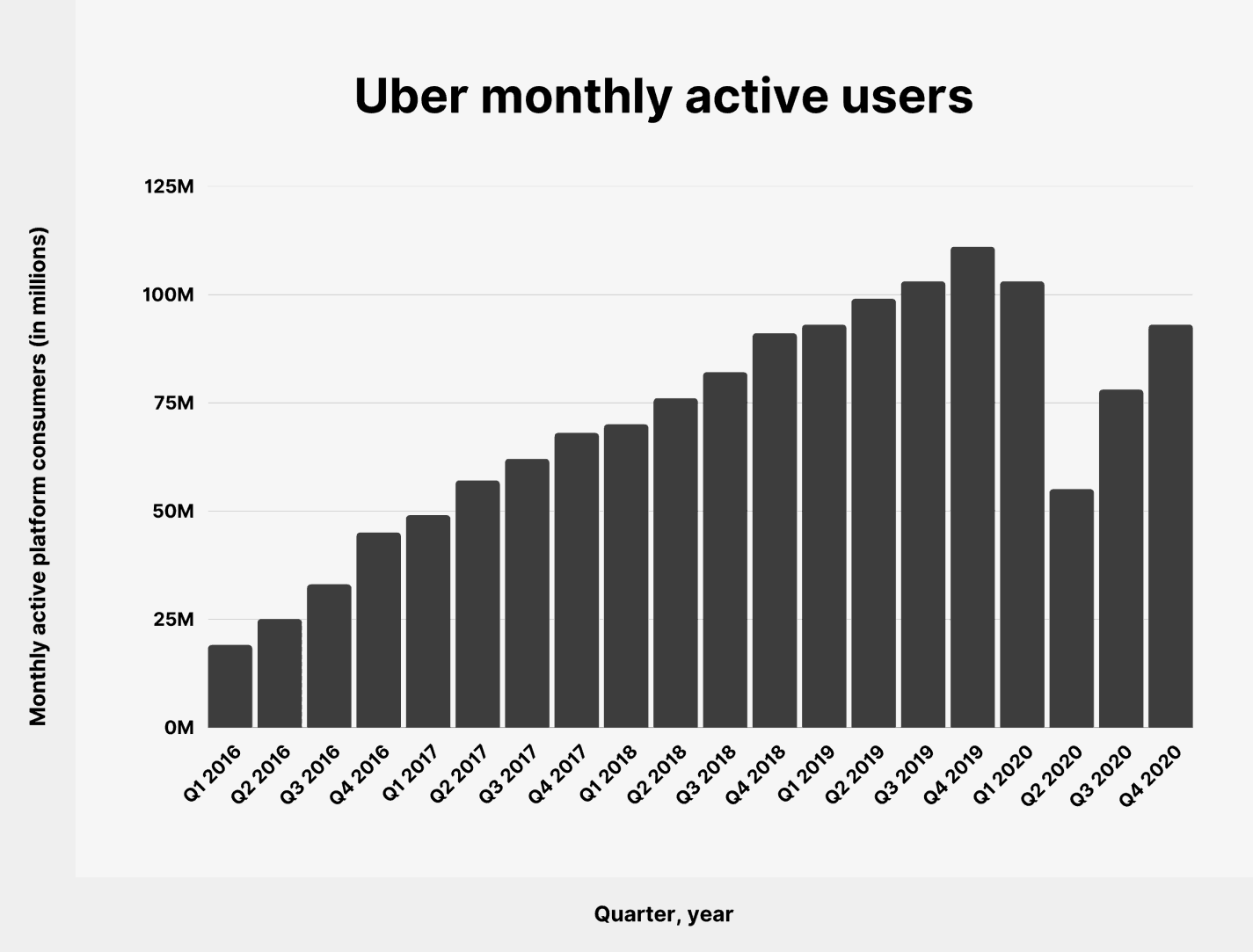


**How does Uber price work?**

If you request a ride on Saturday night, you may find that the price is different from the cost of the same trip a few days earlier. That’s because of our dynamic pricing algorithm, which converts prices according to several variables, such as the time and distance of your route, traffic, and the current need of the driver. In some cases, this may mean a temporary increase in price during very busy times.

**Why are Uber rates changing?**

As demand increases, Uber uses flexible costs to encourage more drivers to get on the road and help address a number of passenger requests. When we inform you of an increase in Uber fees, we also inform drivers. If you decide to proceed and request your ride, you will receive a warning in the app to make sure you know that ratings have changed.



**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’.

**Organization of the Project Report:**

The first section of this paper presents the concept of exploratory data analysis which told general information about the dataset. Then from the next section feature engineering part was started in which we plot many charts and deal with columns to extract the features helpful for our predictions in many ways. In the last part, we did modeling and testing in which we apply different models to check the accuracy and for further price prediction.

**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.

**Packages used:**

**1. Numpy:** It is a python library in whichwe use Numpy here for calculating any numerical and mathematical values. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

**2. Pandas:** It is a python package where we use it for importing the data sets and used for analyzing the data. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

**3. Matplotlib:** It is a comprehensive library for creating static, animated, and interactive visualizations in Python.is a comprehensive library for creating static, animated, and interactive visualizations in Python.

**4. Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

**Models:**

**Regression model:** Regression models are used to predict a continuous value. Predicting prices of a house given the features of house like size, price etc is one of the common examples of Regression.

**MACHINE LEARNING**

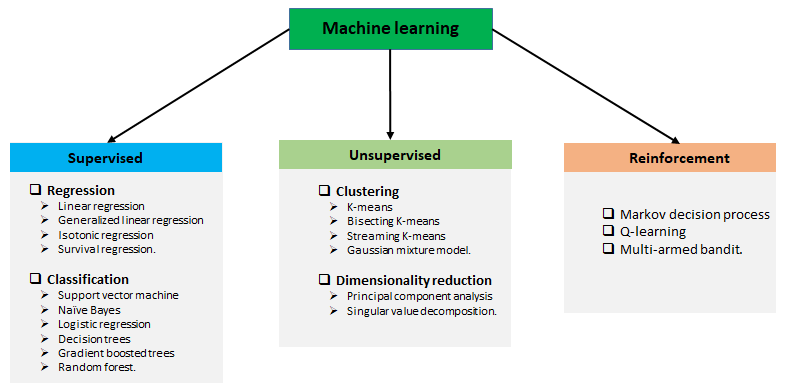
**What is Machine Learning?**

Machinelearning (ML) is the [scientific study](https://en.wikipedia.org/wiki/Branches_of_science) of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and [statistical models](https://en.wikipedia.org/wiki/Statistical_model) that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use to perform a specific task without using explicit instructions, relying on patterns and [inference](https://en.wikipedia.org/wiki/Inference) instead. It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence).

Machine learning algorithms are used in a wide variety of applications, such as [email filtering](https://en.wikipedia.org/wiki/Email_filtering) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task

**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.



**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](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis). 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.

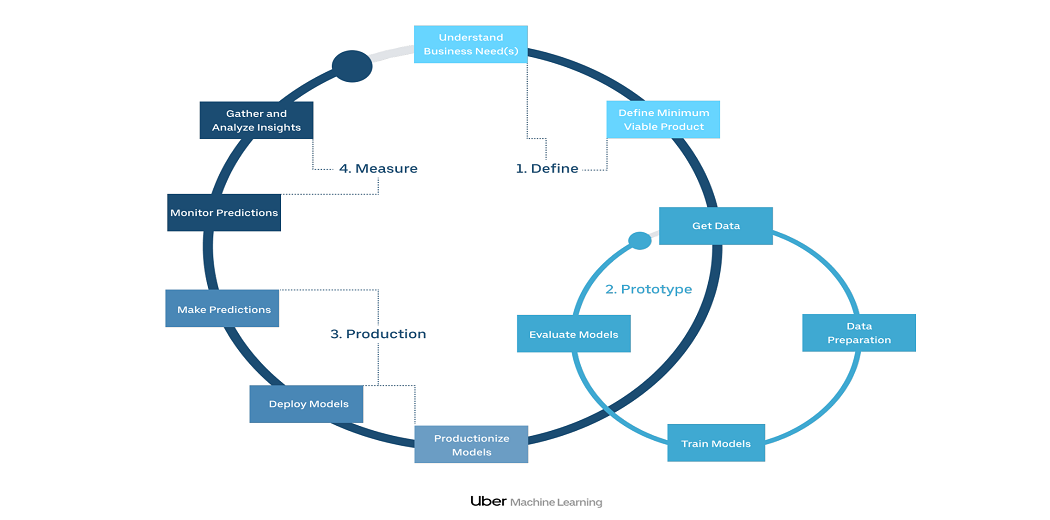
**Unsupervised learning:**

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.

**Reinforcement learning:**

Reinforcement learning is an area of machine learning concerned with how [software agents](https://en.wikipedia.org/wiki/Software_agent) ought to take [actions](https://en.wikipedia.org/wiki/Action_selection) 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.

**1. PROPOSED WORK & IMPLEMENTATION**



**Methodology**

**1.1** **Data Preparation:**

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: - integers, float, and object. The dataset is not complete which means we have also null values in a column named price of around 55095.

**The 57 columns are**:

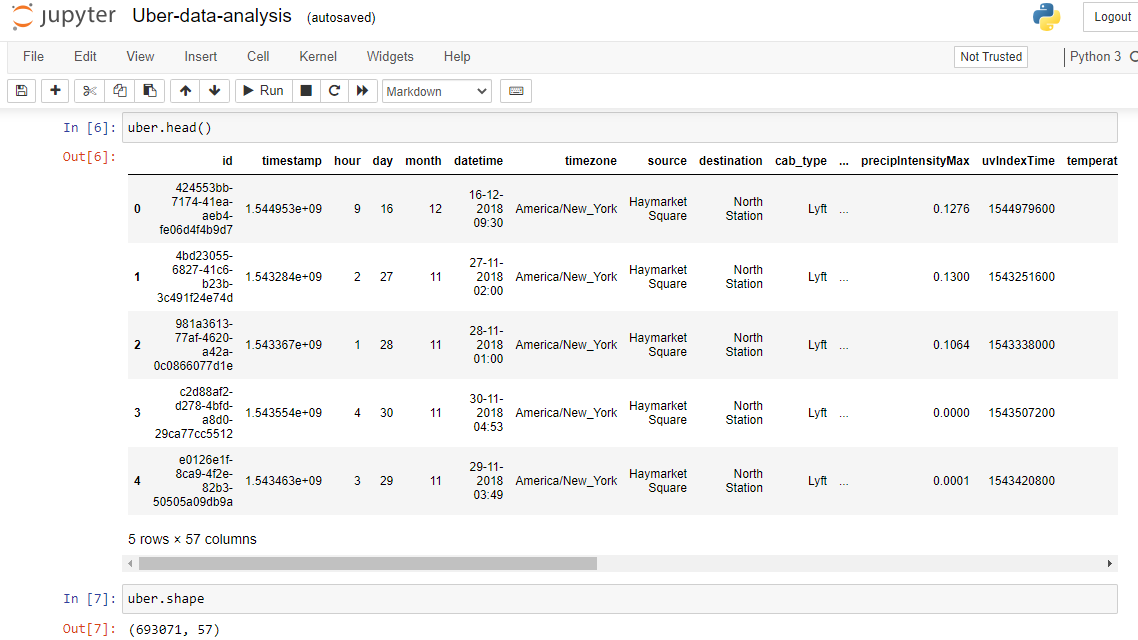
Id, timestamp, hour, day, month, datetime, timezone, source, destination, product\_id, name, distance, surge\_multiplier, latitude, longitude, temperature, apparentTemperature, short\_summary, long summary, precipIntensity, precipProbability, humidity, windSpeed, windGust, windGustTime, visibility, temperatureHigh, temperatureHighTime, temperatureLow, temperatureLowTime, apparentTemperatureHigh, apparentTemperatureHighTime, apparentTemperatureLow, apparentTemperatureLowtime, icon, dewPoint, pressure, windBearing, cloudCover, uvIndex, visibility.1, ozone, sunriseTime, sunsetTime, moonPhase, precipintensityMax, uvindexTime, temperatureMin, temperatureMinTime, temperatureMax, temperatureMaxTime, apparentTemperatureMin, apparentTemperatureMinTime, apparentTemperatureMax, apparentTemperatureMaxTime, price.

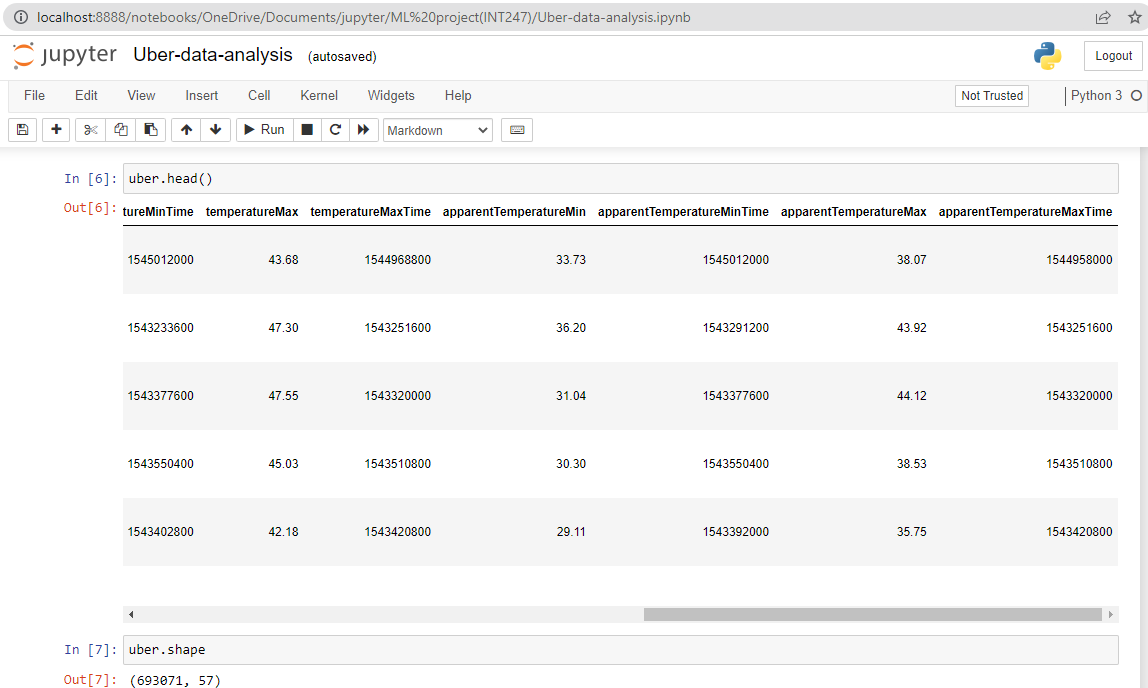
**Importing of the dataset:**

Here we use the pandas for importing the dataset.

****

**1.2 Exploratory data analysis:**

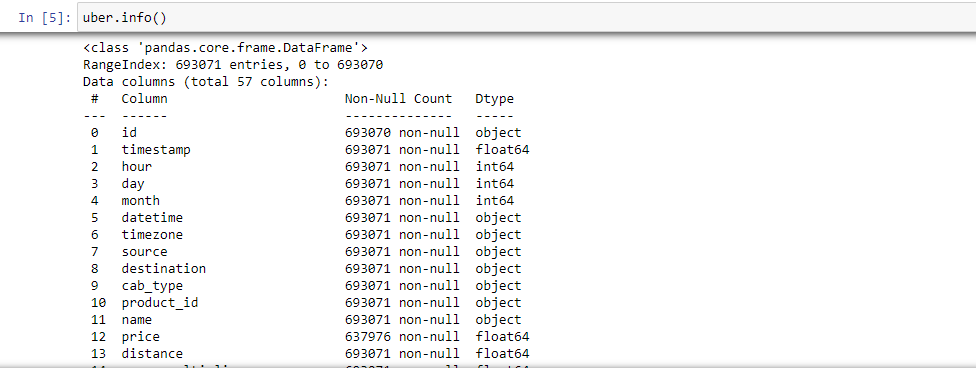


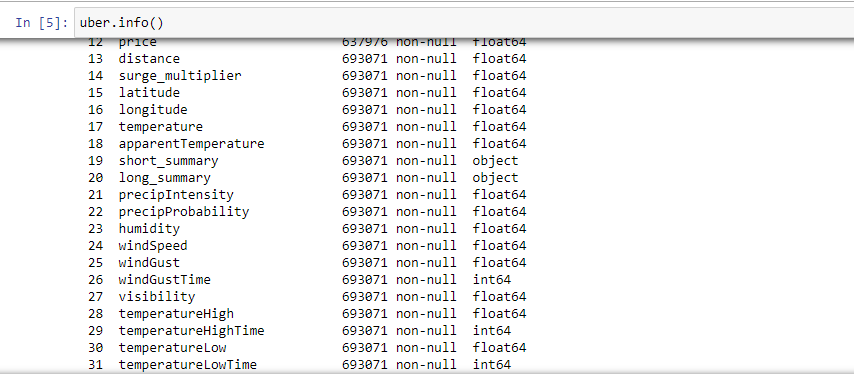
****

**1.2 Data head**

**Information in the data set:**

The info () method prints information about the DataFrame. The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column.

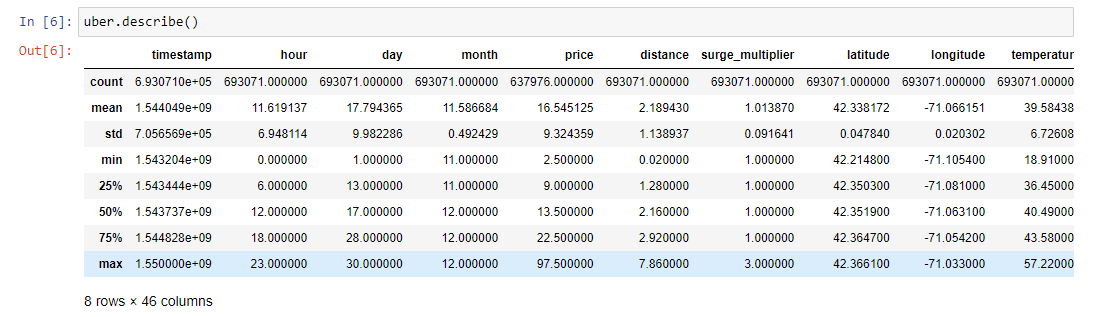
****

****

Here we have 693071 rows and 57 columns for the dataset. We can also there are three different types of data types like float, object and integer.

**Description of the dataset:**

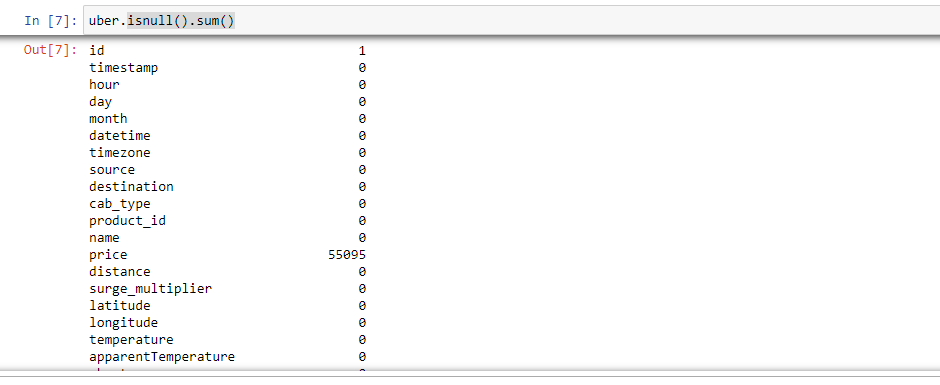
The describe () method returns description of the data in the DataFrame. It analyses both numeric and object series and also the DataFrame column sets of mixed data types.



For the description we got 8 rows and 46 columns like count, mean, std, min, etc…..

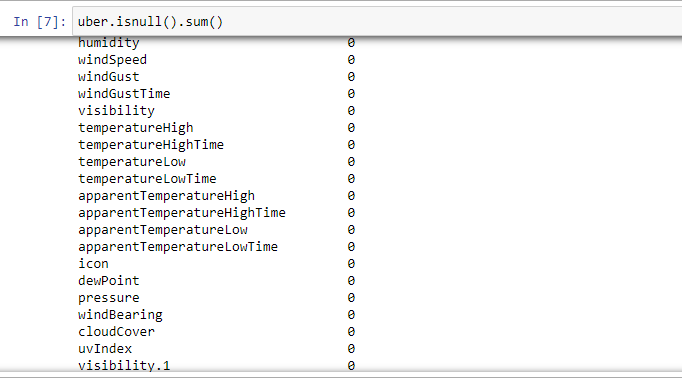
**Checking null values:**

In this we can check the number of missing values in the dataset. A simple way to deal with data containing missing values is to skip rows with missing values in the dataset.



We can see here that the attribute Price has 55095 null values and the rest of the attributes has no null values which is up to apparent temperature.

Now let’s check the other attributes if whether they have any null values in them as price or not.



So here we can see there are no null values in the attributes ecpect the price attribute which have nearly 55000 null values.

**1.3 Data Visualization:**

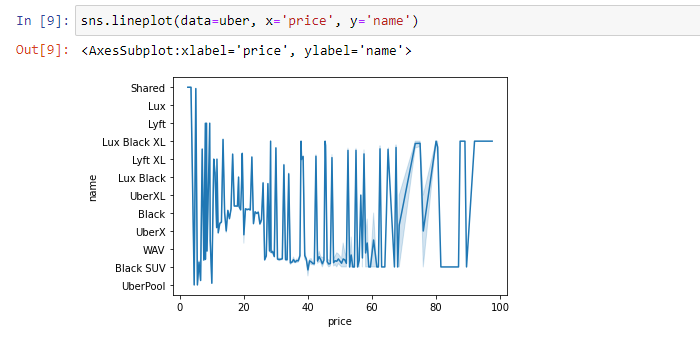
In the data visualization we here represent the data in terms of bar charts, graphs, plots etc... that is created as a visual representation of information. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

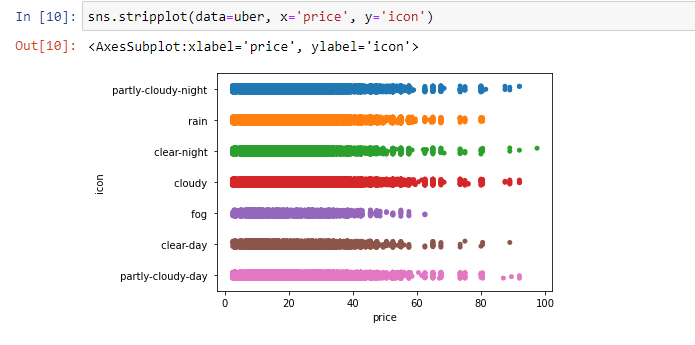


For the data visualization we import both matplotlib and seaborn for the charts.

1. For the first graph we differentiate between name and price and make a line chart and taken the price in the x-axis and name on the y-axis.

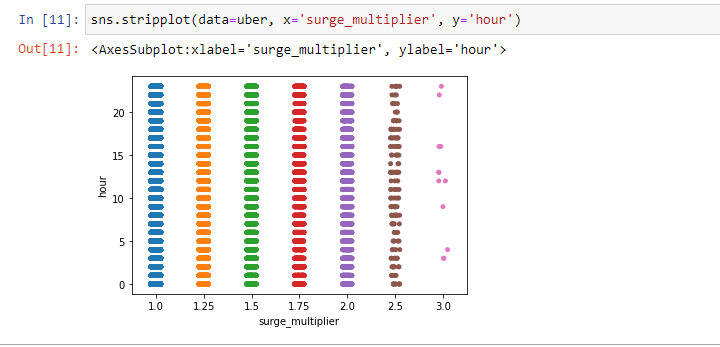
We have nearly 12 different names for the cars like shared, Lux, Lyft, Lux black XL, Lyft XL etc…



2. 

Here we used the stripplot for differentiating the both price and icon of the uber. Here we can observe that icon represents the weather of the day. Here we observe that When the day’s weather is becoming so bad the price is also decreasing and the availability of the uber is also low on that day.

3. Now we used the same stripplot for surge multiplier and hour for differentiating between both of them.

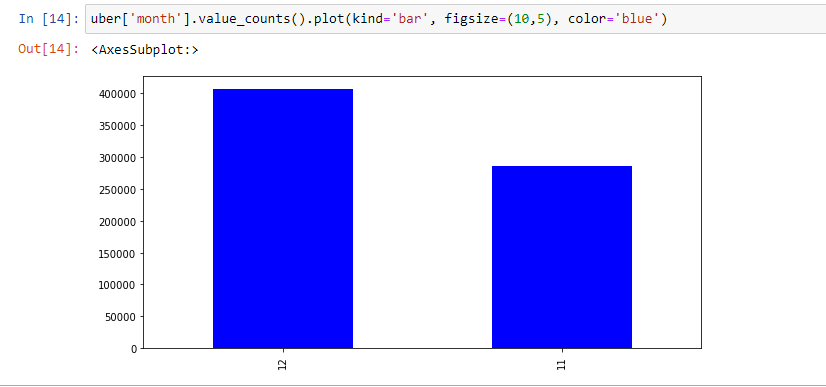


Uber uses a multiplier pricing algorithm to come up with surge pricing amounts for drivers and riders. This multiplier might cause prices to go up by 25%, 50%, or even double. During the busiest of times when demand is extremely high, it's not uncommon to see the multiplier 3x or 4x the regular price of an Uber ride.

Here we observe that when the surge multiplier increases the prices are decreasing slowely but they also not vanished.

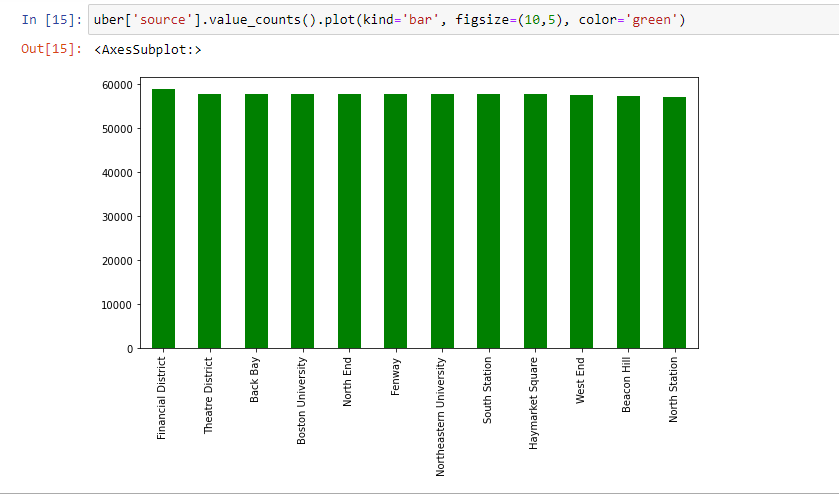
**Bar plots:**

**1.**



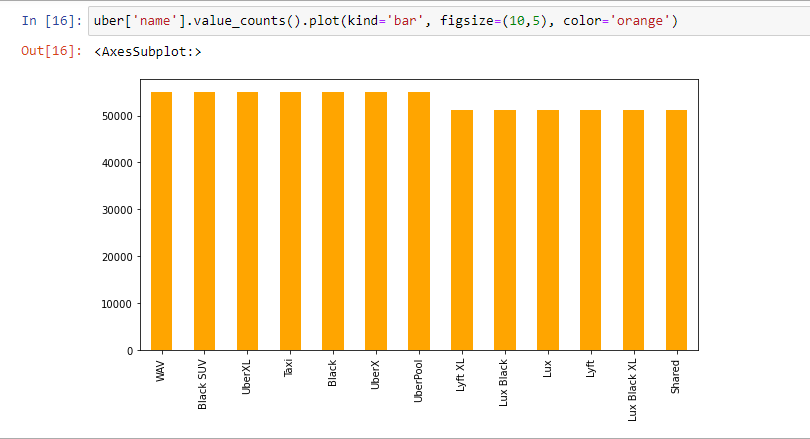
Here we made the bar plot on the “month” and we observe that we have the data on only 2 month that is november and december. Also december has more customers compared to november with more than nearly 150000 customers.

2.



Here we made the bar plot on the “source” and we observe that we have 12 states in this dataset. And we came to know that these 12 sources customers use the uber equally.

3.

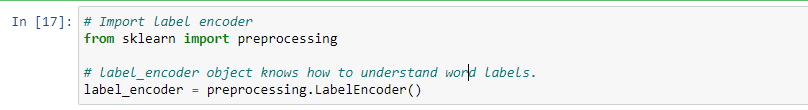


Now we made the bar plot on the attribute “name” and found that every uber named car has the almost same count.

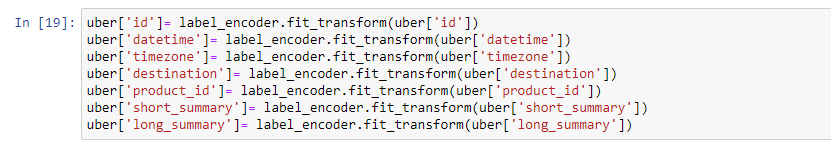
**1.4 Label encoding:**

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.

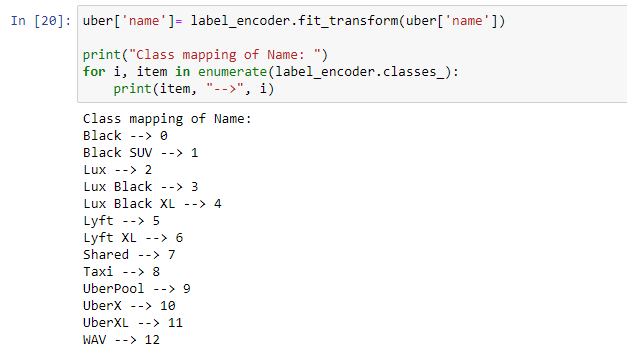
So for doing the label encoding first we have to import the preprocessing model.



So first we have to fit some values into the attributes.

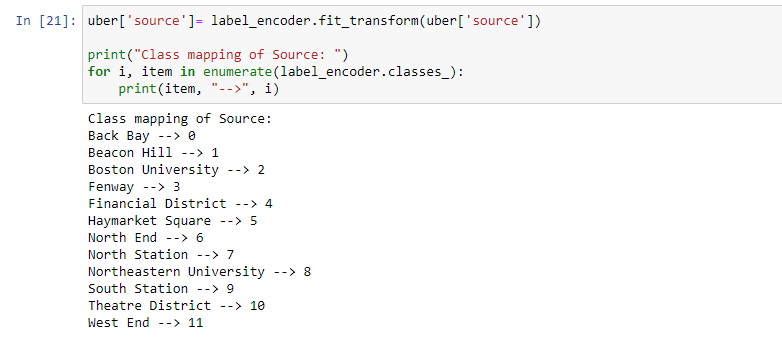


1.



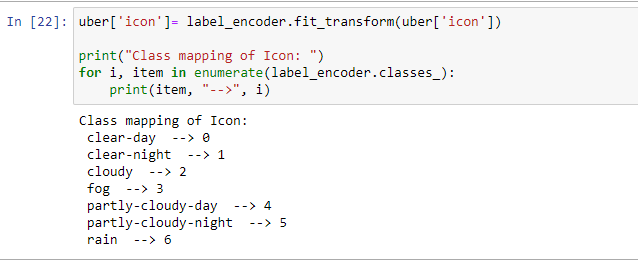
Here we have done the labe encoding on “name” attributes. By label encoding we assign the values to the names.

2.



Here we have done the labe encoding on “source” attributes. By label encoding we assign the values to the source.

**3.**

****

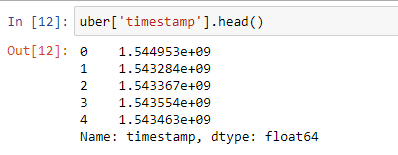
Here we have done the labe encoding on “icon” attributes. By label encoding we assign the values to the icon.

The icon says the weather condition on the day.

So in the label encoding we assign the numerical values to the categorical values so our data can be so simplified for the machine learning model process.

**Converting Timestamp to Date time value:**

We know the timestamp values are float type. So for not getting errors while doing the ml model we are converting them to integer type values.



Now converting them to integer data type.



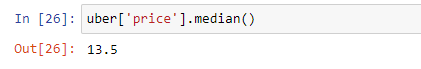
So we give some values to timestamps and converted them into integer types.

The timestamps converted to the integer type and we observe that the time and date is located in the month of November and December.

**1.5 Filling NAN Values:**

We know that the attribute “price” has 55077 null values. So in this we fill the null values with some value.

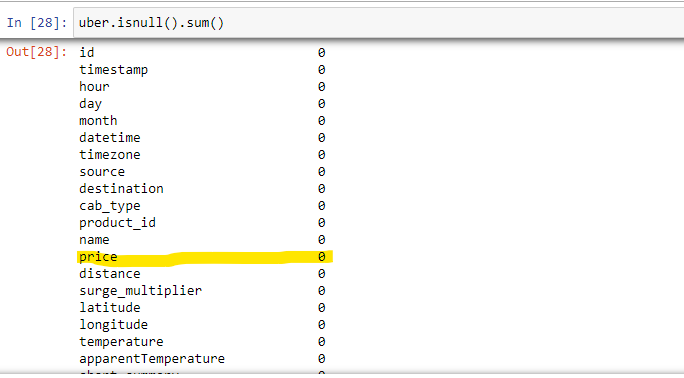
So before we fill first we have to find the median of the price attribute. Then we can keep the value in null values.



We found that median for price is 13.5.



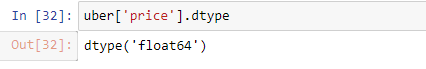
In this we are filling the null values with 13.5.



So here we finally fill the null values of price.

We can see that one heading with yellow colour can say that the attribute price has no null values.

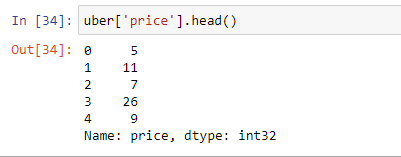
Now we can see that the attribute price is also in float datatype.



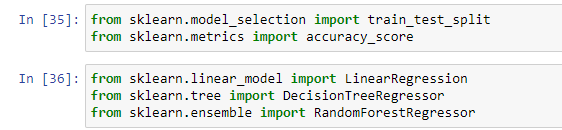
So converting the float to integer type.



Now we can check the price head function.



**2. ML Model**



For doing ml model we have to import the models like train test split and,

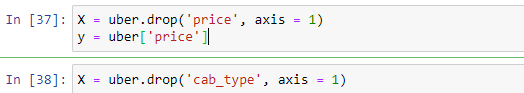
Also for predicting the accuracy of the data we have to import the accuracy.

Also from linear models we import the linear regression model.

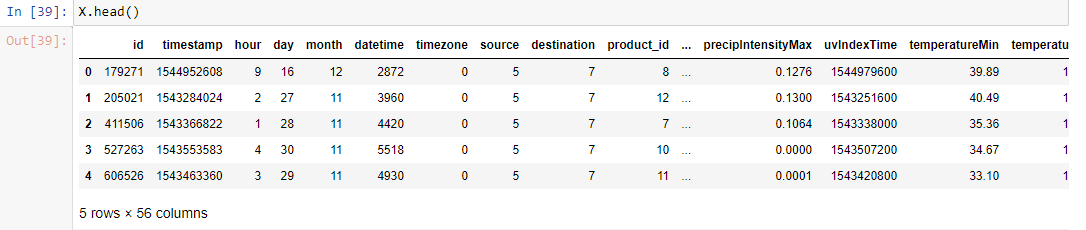
From tree we import the decision tree

From ensemble we import the Random forest.

We are making the ML model on the “price” which is based on cab\_type so we are dropping the price column and cab\_type column from the data set.

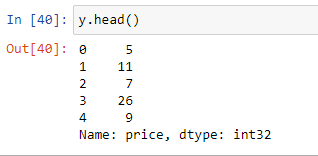


Now we can check the head function of the dataset.

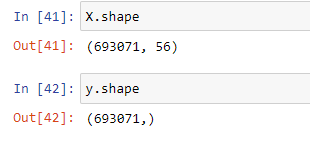


So here the both price and cab\_type columns are dropped.

Now we have the price.



Let’s check the x and y shapes.



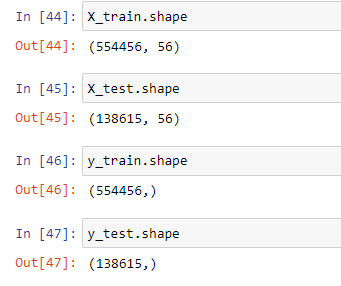
Here we can see that in x- shape one column is missing that is price and that comes into the test shape.

Now we have to split the data for training and testing process by giving the test size.



Here I have given the test size as 0.2 and random state as 42.

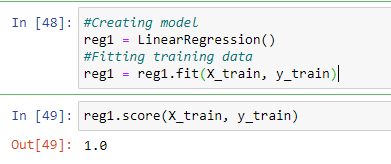
So the 80% data will be in training process and 20% will be in testing process.



So here we can see the data is splitted into training and testing processes.

**2.1 Creating model:**

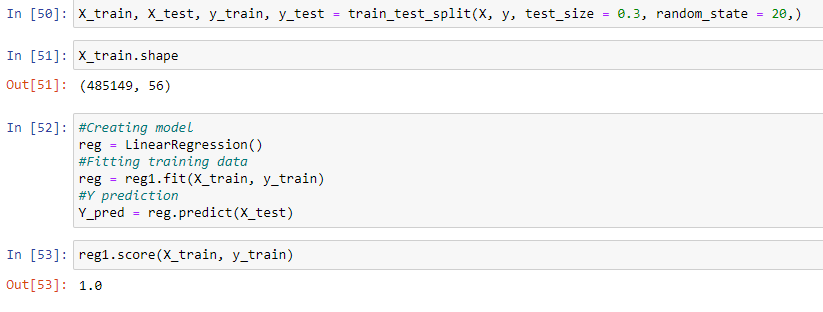
Now we will make a prediction on the linear regression model.



Here we get the prediction as 1.0

Next we make another train test split model with different test size.

Now I will take the test size as 0.3 and check the linear regression model again whether it will get same value as compared to before or not.



So here we can observe that when we change the testing size the linear regression model is same for all of the sizes.

**2.2 Recursive feature selection**:

Recursive feature elimination (RFE) is a feature selection method that fits a model and removes the weakest feature (or features) until the specified number of features is reached.

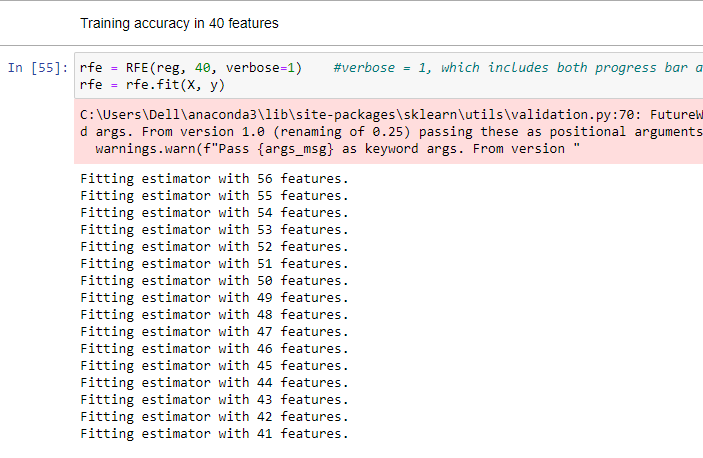
Feature selection is an important task for any machine learning application. This is especially crucial when the data has many features. The optimal number of features also leads to improved model accuracy. So we use RFE for feature selection in our data.

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.



So now we check the training accuracy in 40 features.

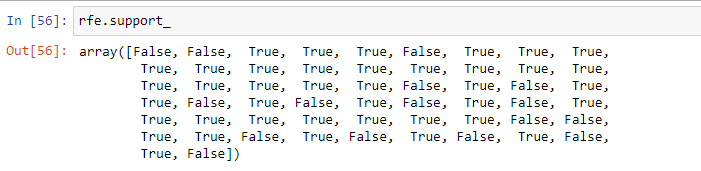
In this we use verbose so that it will include both progress bar and one line per epoch when verbose is 1.



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​ |

Now we check the support of our prediction.

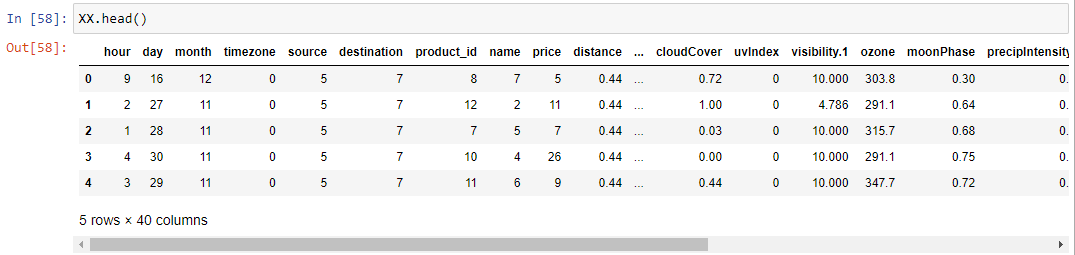


Now we will give an extra variable xx for this rfe support.

From now we will call it with xx variable.

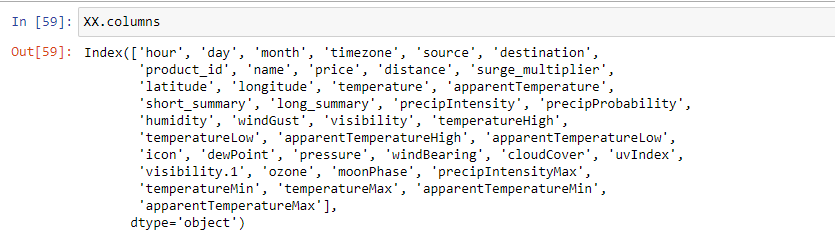


Now we check the head of this xx dataset



Now we got the 25 columns from the rfe and removed the other columns which are affecting our price prediction.

Now we will check the columns of the xx data frame.

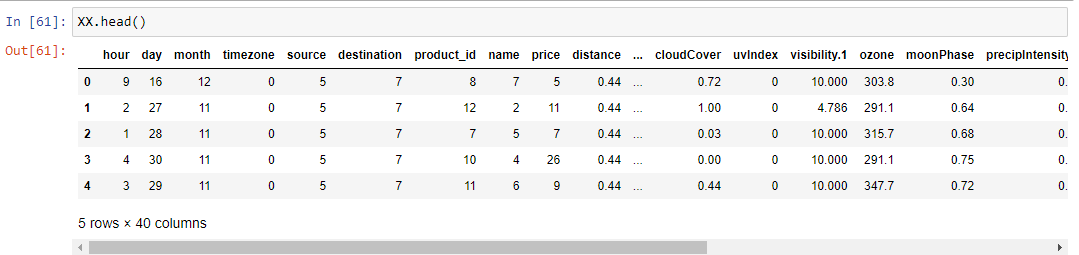


Checking the shape



So here we can see that the shape of the xx data frame is also changed according to the rfe model with same rows but with different columns like unusual columns are removed from the data frame.

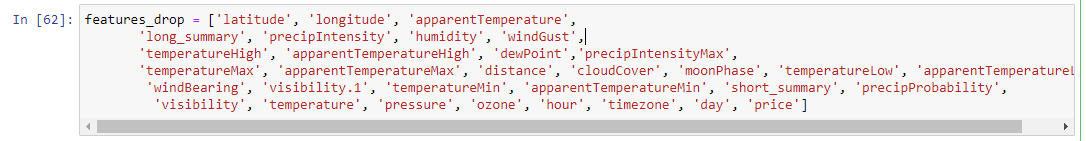
Now checking the head of xx data frame



**2.3 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.

First we have to check the columns which are affecting and not wanted for the price prediction analysis.



**2.4 Binning:**

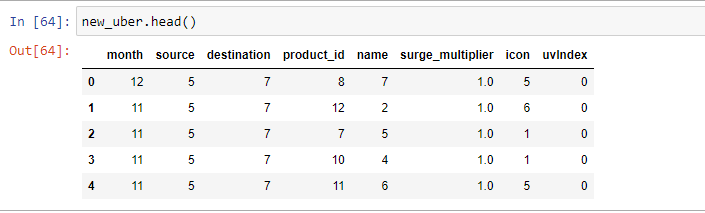
Many times we use a method called data smoothing to make the data proper. During this process, we define a range also called bin and any data value within the range is made to fit into the bin. This is called the binning. Binning is used to smoothing the data or to handle noisy data.

Now we will drop the columns which are not necessary for the price prediction of our data

By using drop () function.

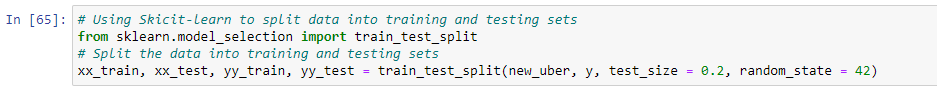


Now let’s check the final dataset by using head function.



So we can see now that we have only 8 columns which is the final dataset that we got after doing binning.

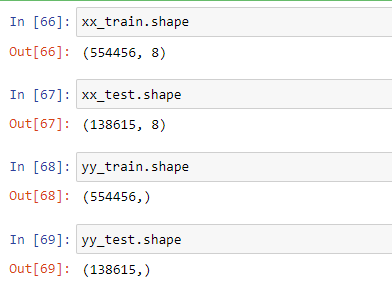
Now we will do the train test split to the new updated dataset.



Here we have given the test size as 0.2 and random state as 42.

We also assign xx to the training data and yy to the testing data.

Now we will check the sizes of the xx\_train, xx\_test and yy\_train, yy\_test which is having only 8 columns and nearly 609500 rows.



**2.5 Modelling:**

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.

**2.6 Regression Models:**

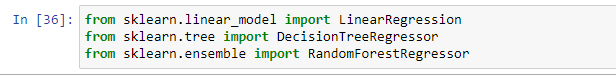
In these models we have used 3 models for the dataset. They are:

Linear regression,

Decision tree Regressor,

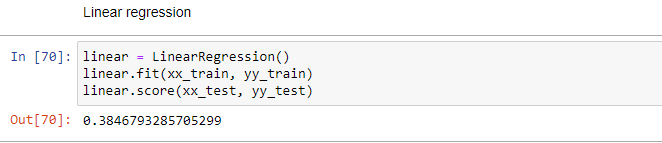
Random forest Regressor.

First we have to import the models.



**1. Linear regression:**

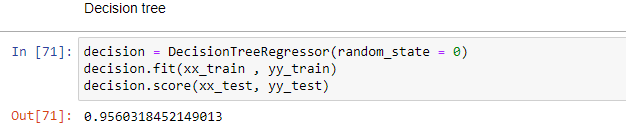
Linear Regression is a supervised machine learning algorithm where the predicted output is continuous in the range such as salary, age, price, etc. It is a statistical approach that models the relationship between input features and output. The input features are called the independent variables, and 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.



Here we fit the model with linear regression model and so finally we got our prediction value as 0.3846793285705299.

**2. Decision tree regression:**

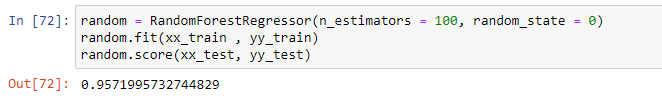
Decision tree is a supervised learning algorithm which can be used for both classification and regression problem. This model is very good at handling tabular data with numerical or categorical features. It uses a tree-like structure flow chart to solve the problem. A decision tree is arriving at an 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.



Here we fit the model with Decision tree regression model and so finally we got our prediction value as 0.9560318452149013.

**3. Random forest regression:**

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.



Here we fit the model with Decision tree regression model and so finally we got our prediction value as 0.9571995732744829.

So finally we can have the table of all regression models:

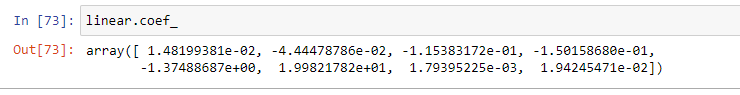
|  |  |  |
| --- | --- | --- |
| **Serial No.​** | **Models** | **prediction** |
| 1​ | Linear Regression​ | 0.384679328570529 |
| 2​ | Decision Tree​ | 0.956031845214901 |
| 3​ | Random Forest​ | 0.957199573274482 |

**2.7 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.

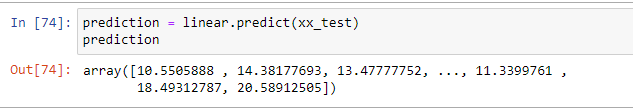
Machine learning applications are not 100% accurate and approx. never will be. There are some of the reasons why testers cannot ignore learning about machine learning. The fundamental reason is that these applications learning limited by data they have used to build algorithms. For example, if 99% of 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.

Finding the linear coefficients in the testing:



Here we got some linear coefficients.

Now predicting the linear of xx\_test:



Now predicting the astype:



**Mean Squared Error (MSE),**

**Mean Absolute Error (MAE),**

**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.

**1. Mean Squared Error (MSE):**

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**

****

**2. Mean Absolute Error (MAE):**

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.

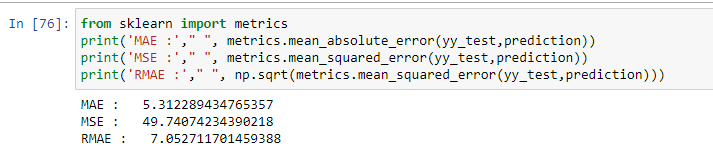


**3. Root Mean Squared Error (RMSE):**

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.



Final program of our problem models;



So finally making the box for the regression problem prediction.

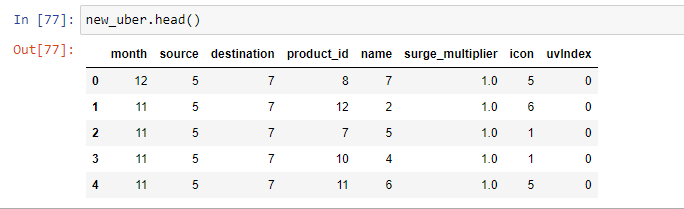
|  |  |  |
| --- | --- | --- |
| **Serial No.​** | **Models** | **Prediction** |
| 1​ | Mean Absolute Error | 5.312289434 |
| 2​ | Mean Squared Error | 49.740742343 |
| 3​ | Root Mean Absolute Error | 7.052711701 |

3. **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.

Now let’s check our data head before predicting the price

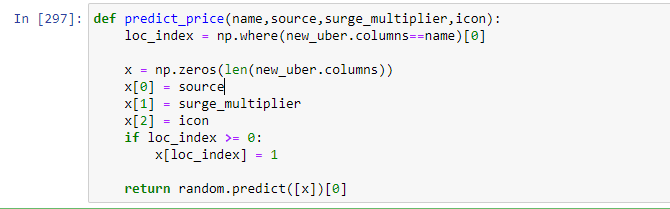


So here we can see that we have the updated data set with minimum number of columns which are sufficient for the price prediction analysis.

Now let’s do the program for the price prediction. For this we have to assign some attributes to the function so that the uber can calculate the price based on that attributes.

Here we use only that attributes in which we have given the label encoding values like we have changed from categorical to numerical values.

So from the label encoding we have converted values of name, source, and icon.



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

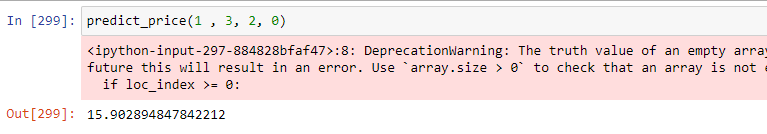


Follow these instructions before predicting the price:

* 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



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.

**ADVANTAGES and DISADVANTAGES of UBER:**

**Advantages:**

1. Uber does the door-to-door convenience for the customers.
2. Uber has the more safety and comfortable things in their vehicles.
3. Uber makes its customers to rate the driver’s behaviour and their skills.
4. Uber uses the drivers who are so good in driving skills.
5. Different service tiers are available.
6. It is very easy to contract a vehicle to use through Uber.
7. Drivers are not forced to pay service costs for their vehicle.

**Disadvantages:**

1. Uber always increases its pricing on the occasions.

2. Uber uses automated system to increase prices based on supply and demand.

3. Uber charges surge pricing and primetime pricing.

4. Drivers sometimes try to game the system for bigger profits.

5. Customers may find themselves spending more than they normally would.

6. Uber is not available in all locations.

7. It functions off of an automated system.

**CONCLUSION**

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.

**REFERENCE**

* Abel Brodeurand & Kerry Nield (2018) an empirical analysis of taxi, Lyft and Uber rides: Evidence from weather shocks in NYC
* Junfeng Jiao (2018) Investigating Uber price surges during a special event in Austin, TX
* Anna Baj-Rogowska (2017) Sentiment analysis of Facebook posts: The Uber Case
* Anastasios Noulas, Cecilia Mascolo, Renaud Lambiotte, and Vsevolod Salnikov (2014) OpenStreetCab: Exploiting Taxi Mobility Patterns in New York City to Reduce Commuter Costs.

**Summary:**

1. In this document I have explained about each and every code cell point.
2. I’ve explained every code along with the pictures.
3. In this I’ve also explained about the concepts clearly with their definitions.
4. I have added some more topics in this document.
5. Also added the advantages and disadvantages.
6. Compared to last document this document has everything about my project