SHIVAJI UNIVERSITY, KOLHAPUR

A PROJECT REPORT

ON

Uber’s Pickup Analysis

Submitted to,

Department of Mathematics

Shivaji University, Kolhapur

in the partial fulfillment of

M.Sc-Tech Mathematics

Part-III, Sem-VI

(2022-2023)

Presented by

Mr. Omkar Balwant Jadhav

Under The Guidance of

Mr. Harsh Kumar Sager

Next Innovate Techno Solutions (P) Ltd., Roorkee

2022-2023

Department of Mathematics

SHIVAJI UNIVERSITY, KOLHAPUR

CERTIFICATE

This is to certify that, **Omkar Balwant Jadhav** has satisfactorily completed the project entitled **Uber's Pickup Analysis** in the partial fulfillment of M.Sc. Tech Mathematics Part – III, Sem VI during the academic year 2022-2023.

Place: Kolhapur

Date: .... /..../.....

Internal Examiner External Examiner HOD

Prof. Dr.(Mrs)S.H.Thakar

Dept. of Mathematics

Shivaji University, Kolhapur

ACKNOWLEDGEMENT

Every project is always a scheduled, guided, and coordinated team effort aimed at achieving common minimum goals. These minimum goals cannot be achieved without the guidance of a project guide.

It is with immense pleasure that we present our report to our project guide, **Mr. Harsh Kumar Sager**. We find no words to describe his efforts and total confidence in our potential to see this project to completion. He has always been a source of inspiration. We would like to express our gratitude to the CEO of Next Innovate Techno Solutions Pvt. Ltd. (NITS), **Mr. Anshul Tyagi**, and our Head of the Mathematics Department, **Dr. (Mrs.) S. H. Thakar**, for their continuing support and encouragement. We sincerely express our gratitude to our parents for their blessings in making this project successful.

Finally, we are thankful to NITS and our Mathematics Department, as well as all our friends who have helped us realize our efforts.

Thanking all of them, again.

Date:…./…./….

Place: Kolhapur

Mr. Omkar Balwant Jadhav

M.Sc. Tech-Mathematics Part-III

DECLARATION

I hereby declare that the project report entitled **Uber's Pickup Analysis** have not formed earlier the basis for the award of any degree of this or any other university examining body.

Further, I declare I have not violated any of the previous under copyright act.

Place: Kolhapur

Date: …./…./ ....

Mr. Omkar Balwant Jadhav

M.Sc. Tech-Mathematics

Part-III

Next Innovate Techno Solutions (P) Ltd.

(NITS)

Registered under Ministry of Micro, Small and Medium Enterprises

Registered Address: Plot No: 256, NITS, Roorkee, Uttarakhand - 247668
Block, DLF Phase – 3, Sector – 24, Gurugram, Haryana – 122002
Email: info@nextinnovatetechnosolutions.com Website: www.nextinnovatetechnosolutions.com

NEXT INNOVATE TECHNO
SOLUTION PVT LTD

LETTER OF CONFIRMATION

Ref No:- **NPLG/2021/DS-BJO5** Date:- 25th January 2023

This is to inform you that Mr. Omkar Balwant Jadhav has been successfully enrolled for 4 months training program in Data Science conducting by Next Innovate Techno Solutions (P) Ltd with effect from 25th January 2023 to 15th May 2023.

During this 4 months training program in Data Science students will be covering an Industrial Project. This Project will be guided by Mr. Harsh Kumar Sagar (Project Coordinator) based on topic according to their area of interest in Data Science.

After the completion of 4 months training program students will be getting:

- ✓ Certificate of Completion
- ✓ Letter of Recommendations
- ✓ 100% Placement Assistance
- ✓ Stipend

We hope you achieve every success in your future endeavours.

From Next Innovate Techno Solutions Pvt. Ltd.

Mr. Anshul Tyagi
Chief Executive Officer, CEO
Next Innovate Techno Solutions Pvt. Ltd.
anshul@nits.com www.nextinnovatetechnosolutions.com

Next Innovate Techno Solutions (P) Ltd.

(NITS)

NEXT INNOVATE TECHNO SOLUTION PVT LTD

Registered under Ministry of Micro, Small and Medium Enterprises

Registered Address: Plot No: 256, NITS, Roorkee, Uttarakhand - 247668
Block, DLF Phase – 3, Sector – 24, Gurugram, Haryana – 122002
Email: info@nextinnovatetechnosolutions.com Website: www.nextinnovatetechnosolutions.com

LETTER OF RECOMMENDATIONS

Ref No:- **NPLG/2021/DS-BJO5** Date:- 15th May 2023

This is to inform you that Mr. Omkar Balwant Jadhav has completed 4 months training program in Data Science at Next Innovate Techno Solutions Pvt. Ltd. With effect from 25th January 2023 to 15th May 2023.

He has excellent communication skills. In addition he is extremely organized and reliable. He can work independently and is able to follow through to ensure that the job gets done. He is flexible and willing to work on any project that assigned to him.

He has completed his final major project as **"Uber's Pickup Analysis"**

The project has been verified & submitted successfully and the final score is 73% .

We hope you achieve every success in your future endeavours.

From Next Innovate Techno Solutions Pvt. Ltd.

NEXT INNOVATE TECHNO SOLUTIONS ★ CERTIFIED

Mr. Anshul Tyagi
Chief Executive Officer, CEO
Next Innovate Techno Solutions Pvt. Ltd.
anshul@nits.com www.nextinnovatetechnosolutions.com

The certificate is verified under "An ISO 21001" certifications

For verifications mail the reference id of certificate at info@nextinnovatetechnosolutions.com

COMPANY INFORMATION

Next Innovate Techno Solutions is one of the renowned IT Companies in Roorkee. Our company provides several IT, Web, Writing, and Learning solutions. Our team works their best to provide you with IT services. We have an expert team of creative designers, skilled developers, Experienced writers, and well-trained staff**.**

Along with IT and Software services, NITS also operates in Ed-Tech, where we offer various training services to all candidates as well as to working professionals.

As an IT and Software company, NITS also deals in Software Development and provides solutions to problems and challenges faced by growing businesses and companies transitioning into the new era and world of technological revolution. So that keeping up with the cutting-edge technologies isn't so edgy for you!

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 overfitting. We got many experiences while doing the data preparation of Uber Dataset of Boston of the year 2022. It was also very interesting to know how different factors affect the pricing of Ubers_data.

INDEX

List of Figures

List of Tables

INTRODUCTION

1.1 Motivation and 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.

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 56 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'.

1.3 Issues and Challenges

1. **Overfitting in Regression Problem:-** Overfitting a model is a condition where a statistical model begins to describe the random error in the data rather than the relationships between variables. This problem occurs when the model is too complex. In regression analysis, overfitting can produce misleading R-squared values. When this occurs, the regression coefficients represent the noise rather than genuine relationships. However, there is another problem. Each sample has its unique quirks. Consequently, a regression model that becomes tailor-made to fit the random quirks of one sample is unlikely to fit the random quirks of another sample. Thus, overfitting a regression model reduces its generalizability outside the original dataset.

2. **Strip-plot and Scatter diagram:-** One problem with strip plots is how to display multiple points with the same value. If it uses the jitter option, a small amount of random noise is added to the vertical coordinate and if it goes with the stack option it increments the repeated values to the vertical coordinate which gives the strip plot a histogram-like appearance.

   Scatter plot does not show the relationship for more than two variables. Also, it is unable to give the exact extent of correlation**.**

3. **Label Encoding:-** It assigns a unique number(starting from 0) to each class of data which may lead to the generation of priority issues in the training of data sets. A label with high value may be considered to have high priority than a label having lower value but actually, there is no such priority relation between the attributes of the same classes.

4. **Computational Time:-** Algorithms like support vector machine(SVM) don’t scale well for larger datasets especially when the number of features are more than the number of samples. Also, it sometimes runs endlessly and never completes execution.

1.4 Contribution

Each team member is responsible and has willing participation in the group. The work within the group is equally done by each team member. First, the project work is divided like one has to be done the exploratory data analysis part, two members work on feature engineering, and the rest work of modeling and testing was equally divided among all four members. And the second part i.e. written work is done in pairs like two members work on report and the other two works on presentation.

1.5 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.

1.6 Understanding The Data

Understanding the data is a crucial step in data analysis as it helps in gaining insights, identifying patterns, and making informed decisions. Here are some steps to understand the data effectively:

1. Data Collection: Start by collecting the data relevant to your analysis. This can be obtained from various sources such as databases, files, APIs, or web scraping.

2. Data Description: Examine the overall characteristics of the data. This includes understanding the variables (columns) and their data types, the size of the dataset, and any missing values or outliers present. You can use summary statistics like mean, median, standard deviation, etc., to describe numerical variables, and frequency tables for categorical variables.

3. Data Visualization: Visualize the data to gain insights and identify patterns. Use techniques like histograms, box plots, scatter plots, and bar charts to understand the distribution of variables, relationships between variables, and any outliers or anomalies. Libraries like Matplotlib and Seaborn in Python can be helpful for creating visualizations.

4. Data Cleaning: Deal with missing values, outliers, and inconsistencies in the data. Decide on the appropriate strategies for handling missing data (e.g., imputation) and outliers (e.g., removal or transformation). Ensure data consistency and correctness by resolving any inconsistencies or errors.

5. Data Exploration: Dive deeper into the data to explore specific aspects or relationships. Compute additional statistics, calculate correlations between variables, and perform exploratory data analysis (EDA). EDA techniques like grouping, filtering, and aggregating can provide further insights.

6. Feature Engineering: If required, create new features (variables) from the existing data that might be more informative or suitable for analysis. This can involve transformations, combinations, or extracting relevant information from existing variables.

7. Domain Knowledge: Utilize domain knowledge or subject matter expertise to interpret the data and identify relevant patterns or insights. Understanding the context and domain-specific nuances can help in meaningful data analysis.

8. Data Documentation: Keep track of the data exploration process, including any transformations or decisions made. Documenting the steps taken and the insights gained will help in reproducibility and provide a reference for future analysis.

By following these steps, you can develop a solid understanding of the data and its characteristics, which will serve as a foundation for performing effective data analysis and making data-driven decisions.

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.

Anderson concluded from surveying San Francisco drivers that driver behavior and characteristics are likely determining the overall vehicle miles traveled (VMT). Full-time drivers are likely to increase overall VMT, while occasional drivers are more likely to reduce overall VMT. We also analyze the research on the driving behavior of the driver while driving on the road. The driver has been categorized based on ages and genders that focus on their driving reactions from how they braking, speeding, and steer handling. For gender differences, male driver practice higher-risk of driving while female drivers are lacks of pre-caution over obstacles and dangerous spot. More or less, adult drivers which regularly drive vehicles can manage the vehicle quite well as compared with young drivers with less experience. In conclusion, the driver's driving behavior is related to their age, gender, and driving experiences.

ANACONDA NAVIGATOR AND PYTHON

Anaconda Navigator is a graphical user interface (GUI) that comes bundled with the Anaconda distribution. Anaconda is a popular distribution of the Python programming language and is widely used for data science and scientific computing tasks.

Python, on the other hand, is a versatile and powerful programming language known for its simplicity and readability. It has a large and active community that contributes to its extensive ecosystem of libraries and frameworks, making it suitable for a wide range of applications.

Anaconda Navigator provides an intuitive interface to manage and launch applications, environments, and packages in Anaconda. It simplifies the process of setting up and managing different Python environments, which can be useful when working on multiple projects with different dependencies. Navigator allows users to easily switch between environments, install or update packages, and access popular integrated development environments (IDEs) such as Jupyter Notebook, JupyterLab, and Spyder.

With Anaconda Navigator, users can also explore and install packages from the Anaconda repository, which offers a vast collection of pre-built packages for scientific computing, install packages, and access popular tools and IDEs, making data analysis, machine learning, and more. This eliminates the need for manual installation and configuration of packages, streamlining the development process.

Overall, Anaconda Navigator enhances the user experience by providing a user-friendly interface to manage Python environmentsit an excellent choice for beginners and professionals alike who are working with Python for data science and scientific computing

Python is a popular programming language known for its simplicity and readability. It has a vast ecosystem of libraries that extend its capabilities for various tasks. In the field of scientific computing and data analysis, four essential libraries are frequently used: NumPy, Matplotlib, Seaborn, and scikit-learn. Let's explore each of these libraries:

1. NumPy:

NumPy is short for Numerical Python and provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. It serves as the foundation for many other libraries in the scientific Python ecosystem. NumPy is known for its speed and versatility and is widely used for numerical computations, data manipulation, and linear algebra operations.

2. Matplotlib:

Matplotlib is a comprehensive plotting library that allows you to create a wide range of static, animated, and interactive visualizations in Python. It provides a MATLAB-like interface and supports a variety of plots, including line plots, scatter plots, bar plots, histograms, 3D plots, and more. Matplotlib can be used to create publication-quality figures and is highly customizable.

3. Seaborn:

Seaborn is a data visualization library that builds on top of Matplotlib. It provides a higher-level interface for creating attractive and informative statistical graphics. Seaborn simplifies the process of generating complex visualizations such as heatmaps, pair plots, violin plots, and categorical plots. It offers a wide range of built-in themes and color palettes to enhance the aesthetics of your plots.

4. scikit-learn:

scikit-learn is a powerful machine learning library that provides efficient tools for data mining and data analysis. It offers a wide range of supervised and unsupervised learning algorithms, including classification, regression, clustering, dimensionality reduction, and model selection. scikit-learn is built on top of NumPy and integrates well with other scientific Python libraries. It also provides utilities for data preprocessing, model evaluation, and model persistence.

These libraries are widely used in the field of data science and provide a strong foundation for performing various tasks, including data manipulation, visualization, and machine learning. They are actively maintained and have extensive documentation and community support, making them ideal choices for scientific computing and data analysis in Python.

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.

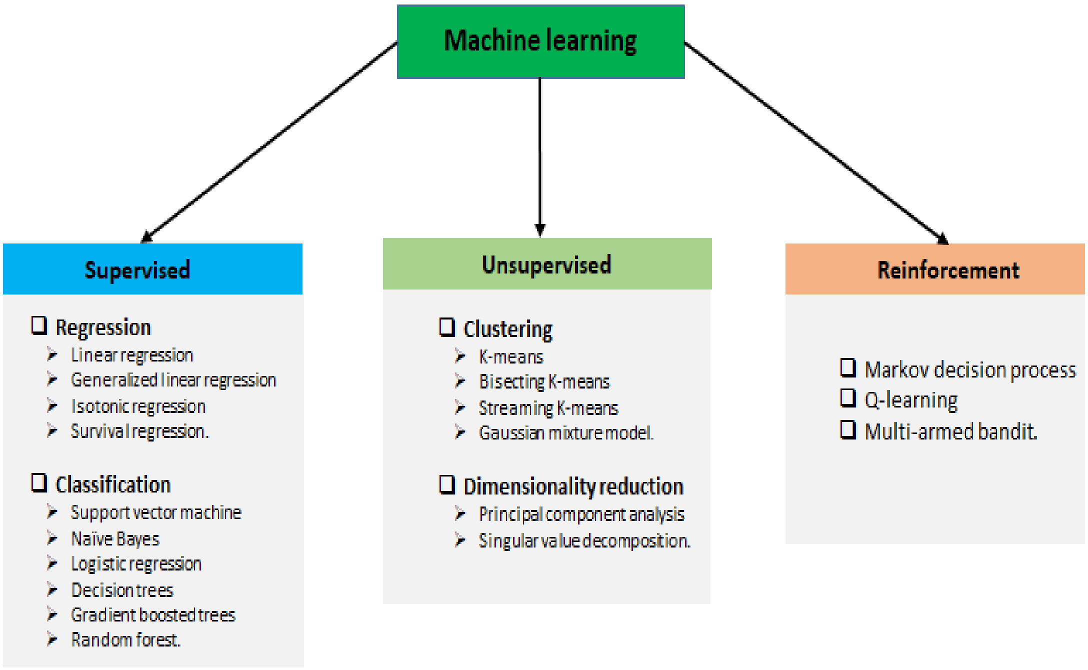


Fig. 3.1 Types of ML Courtesy of Packt-cdn.com

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 labeled, 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.

3.3 Machine Learning Models:

Here's a brief overview of some commonly used machine learning models:

1. Linear Regression:

Linear regression is a simple and widely used model for regression tasks. It assumes a linear relationship between the input variables and the target variable and tries to find the best-fit line that minimizes the difference between the predicted and actual values.

2. Logistic Regression:

Logistic regression is a classification model used when the target variable is categorical. It estimates the probability of a binary outcome based on the input variables using a logistic function. It's often used for binary classification problems.

3. Decision Trees:

Decision trees are versatile models that can be used for both classification and regression tasks. They split the data based on feature values to create a hierarchical structure of decisions. Each leaf node represents a class label or a numerical value.

4. Random Forests:

Random forests are an ensemble learning method that combines multiple decision trees. Each tree is trained on a different subset of the data, and predictions are made by averaging or voting among the individual trees. Random forests are known for their robustness and ability to handle high-dimensional data.

5. Support Vector Machines (SVM):

Support Vector Machines are powerful models used for both classification and regression tasks. SVMs find a hyperplane that maximally separates the data into different classes or predicts a continuous variable. They are particularly effective when dealing with complex decision boundaries.

6. Naive Bayes:

Naive Bayes is a probabilistic model based on Bayes' theorem with the assumption of independence between features. It's often used for text classification tasks and is known for its simplicity and efficiency.

7. Neural Networks:

Neural networks are a class of models inspired by the human brain's structure and function. They consist of interconnected layers of artificial neurons that can learn complex patterns and relationships in the data. Neural networks have gained popularity due to their ability to handle large and high-dimensional datasets, especially in deep learning.

These are just a few examples of machine learning models, and there are many more variations and advanced models available. The choice of model depends on the specific problem and the characteristics of the dataset. It's important to experiment and choose the most appropriate model based on the task at hand.

3.4 Statistical Terms For Machine Learning

1. Prediction:

The function aims to predict the price based on four input parameters: cab name, source, surge multiplier, and icon (weather). By using the random forest model trained on a dataset, it leverages the learned patterns and relationships to provide a predicted price. This prediction allows users to estimate the price they might expect based on the given inputs.

2. Inference:

The function extracts specific rows from the dataset that match the input cab name, indicating a categorical variable. By identifying the row numbers, it ensures that the subsequent assignment of input values to the appropriate indices in the array is correct. This step allows for making inferences about the relationship between the cab name and the price.

3. Hypothesis Space:

The function's ability to generate output for any input within the specified input space relates to the concept of a hypothesis space in statistical learning. The random forest model, trained on continuous values, enables the function to make predictions based on input values, including both categorical and continuous variables. It explores the hypothesis space to identify patterns and relationships between the input parameters and the predicted price.

In summary, the statistical use of the "Predict Price" function lies in its ability to leverage a trained random forest algorithm to predict prices based on input parameters. It allows for making inferences about the relationship between the cab name and the price and explores the hypothesis space to generate predictions for various inputs within the specified space.

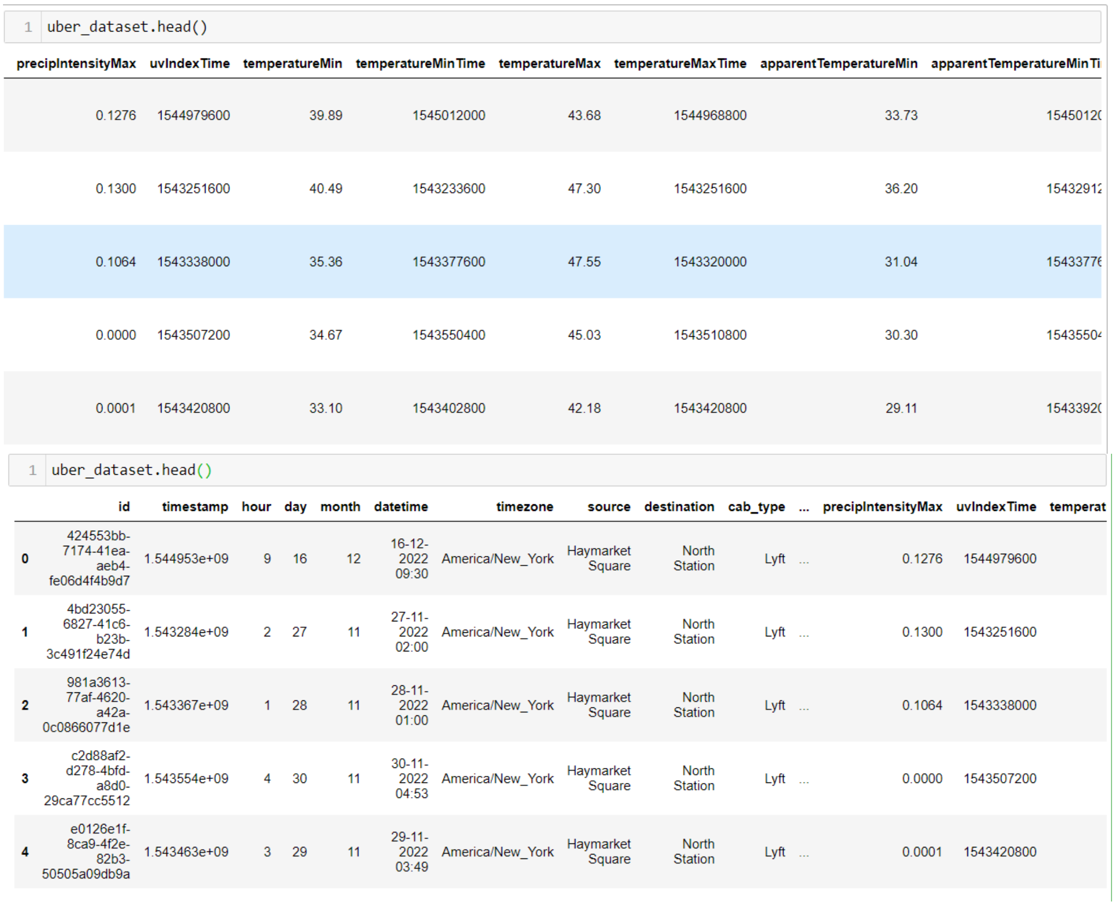
PROPOSED WORK & IMPLEMENTATION

4.1 Data Preparation

4.1.1 Collecting Data

The data we used for our project was provided by Next Innovate Techno Solutions . The original dataset contains 693071 rows and 57 columns which contain the data of both Uber and Lyft. But for our analysis, we just need the Uber data so we filter out the data according to our purpose and got a new dataset that has 322844 rows and 56 columns. 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, float, and object. The dataset is not complete which means we have also null values in a column named price of around 55095.



```
uber_dataset.head()
```

precipIntensityMax	uvIndexTime	temperatureMin	temperatureMinTime	temperatureMax	temperatureMaxTime	apparentTemperatureMin	apparentTemperatureMinTi
0.1276	1544979600	39.89	1545012000	43.68	1544968800	33.73	1545012
0.1300	1543251600	40.49	1543233600	47.30	1543251600	36.20	1543291
0.1064	1543338000	35.36	1543377600	47.55	1543320000	31.04	1543377
0.0000	1543507200	34.67	1543550400	45.03	1543510800	30.30	1543550
0.0001	1543420800	33.10	1543402800	42.18	1543420800	29.11	1543392

```
uber_dataset.head()
```

	id	timestamp	hour	day	month	datetime	timezone	source	destination	cab_type	...	precipIntensityMax	uvIndexTime	temperat
0	424553bb-7174-41ea-aeb4-fe06d4f4b9d7	1.544953e+09	9	16	12	16-12-2022 09:30	America/New_York	Haymarket Square	North Station	Lyft	...	0.1276	1544979600	
1	4bd23055-6827-41c6-b23b-3c491f24e74d	1.543284e+09	2	27	11	27-11-2022 02:00	America/New_York	Haymarket Square	North Station	Lyft	...	0.1300	1543251600	
2	981a3613-77af-4620-a42a-0c0866077d1e	1.543367e+09	1	28	11	28-11-2022 01:00	America/New_York	Haymarket Square	North Station	Lyft	...	0.1064	1543338000	
3	c2d88af2-d278-4bfd-a8d0-29ca77cc5512	1.543554e+09	4	30	11	30-11-2022 04:53	America/New_York	Haymarket Square	North Station	Lyft	...	0.0000	1543507200	
4	e0126e1f-8ca9-4f2e-82b3-50505a09db9a	1.543463e+09	3	29	11	29-11-2022 03:49	America/New_York	Haymarket Square	North Station	Lyft	...	0.0001	1543420800	

Fig. 4.1 Data Head

4.1.2 Filtering Data

Filtering data refers to the process of selectively extracting or displaying a subset of data based on specific criteria or conditions. It is a fundamental operation in data analysis and allows users to focus on relevant information and remove unnecessary or unwanted data. The filtering process involves applying logical conditions to a dataset, where rows or columns that meet the specified criteria are retained, while others are excluded. This helps in narrowing down the dataset to a more manageable size or isolating specific subsets that meet certain requirement. Some most useful filtering technique named as row filtering, Column Filtering , Text Filtering etc.

4.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical step in the data analysis process that focuses on understanding and summarizing the main characteristics of a dataset. It involves examining and visualizing the data to uncover patterns, spot anomalies, and gain insights before applying any formal statistical techniques.

4.3 Handling NaN Values

Handling NaN (Not a Number) values is an essential task in data analysis and processing. NaN values typically arise from missing or incomplete data. To handle NaN values, various approaches can be employed. One common strategy is to remove rows or columns containing NaN values, but this may result in data loss. Another approach is to fill NaN values with a specific value, such as the mean or median of the column. Alternatively, interpolation methods can be used to estimate missing values based on the surrounding data points. Ultimately, the choice of handling NaN values depends on the specific dataset and the analysis goals, and it is important to consider the potential impact on the accuracy and integrity of the data.

4.3.1. 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

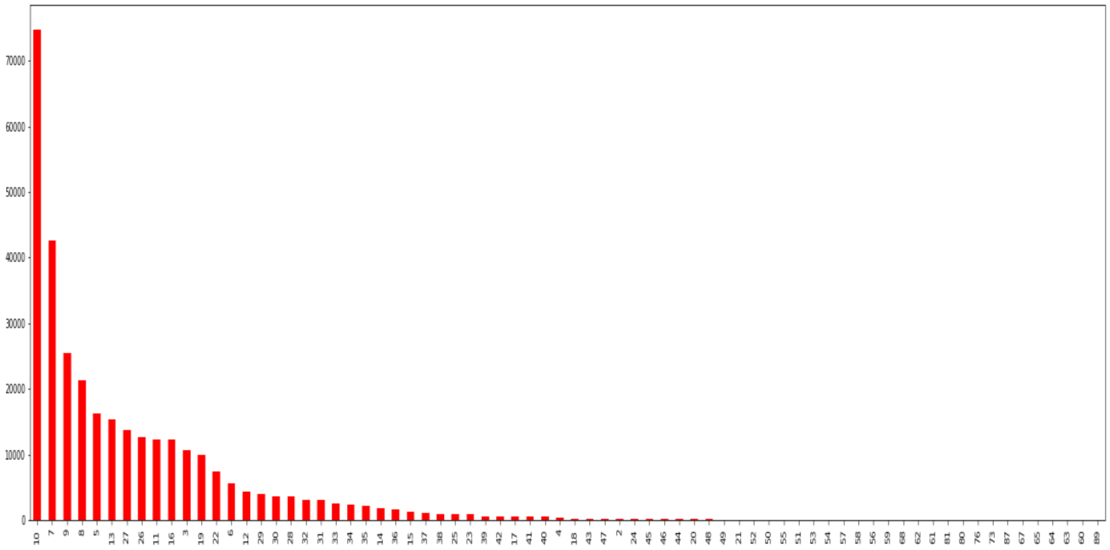


Fig. 4.8 Bar Chart of Price

4.4 Feature Engineering

Feature engineering is the most important part of the data analytics process. It deals with, selecting the features that are used in training and making predictions. All machine learning algorithms use some input data to create outputs. This input data comprise features, which are usually in the form of structured columns. Algorithms require features with some specific characteristics to work properly. A bad feature selection may lead to a less accurate or poor predictive model. To filters out all the unused or redundant features, the need for feature engineering arises. It has mainly two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the performance of machine learning models.

“According to a survey in Forbes, data scientists spend 80% of their time on data preparation.”

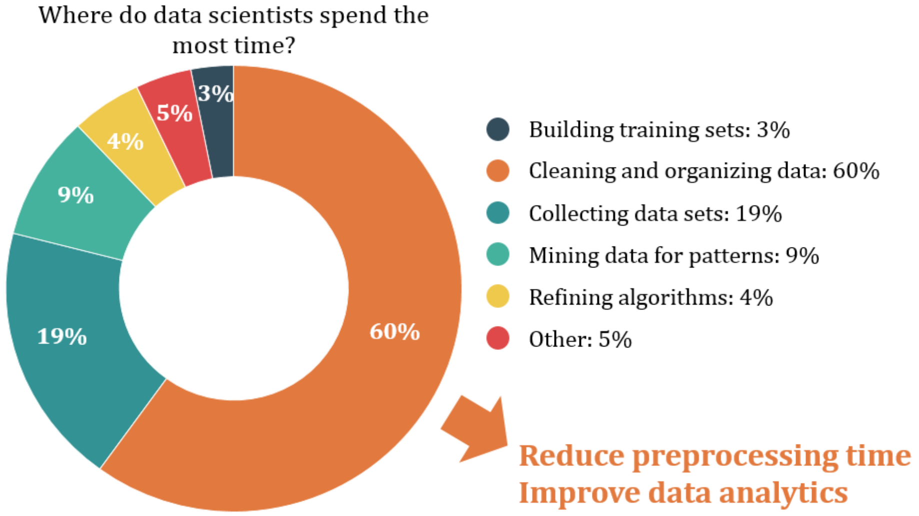


Fig. 4.7 Feature Engineering Courtesy of Digitalag

4.5 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.

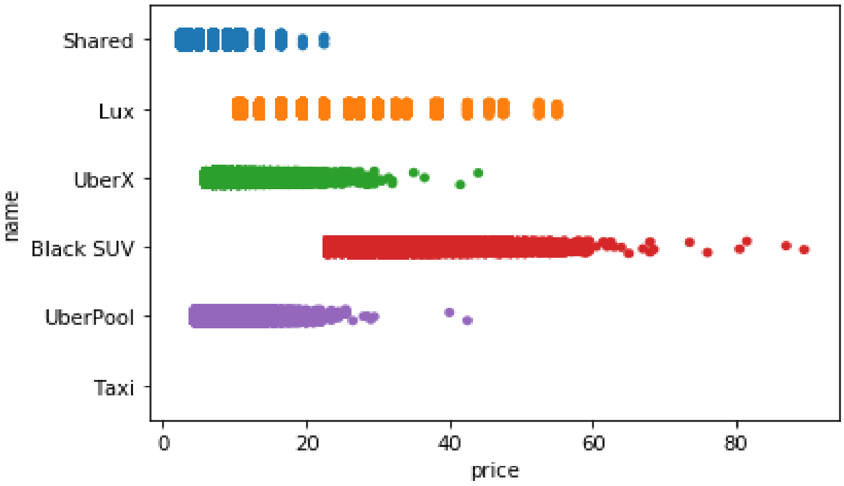


Fig. 4.2 Strip-plot between Name and Price

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.

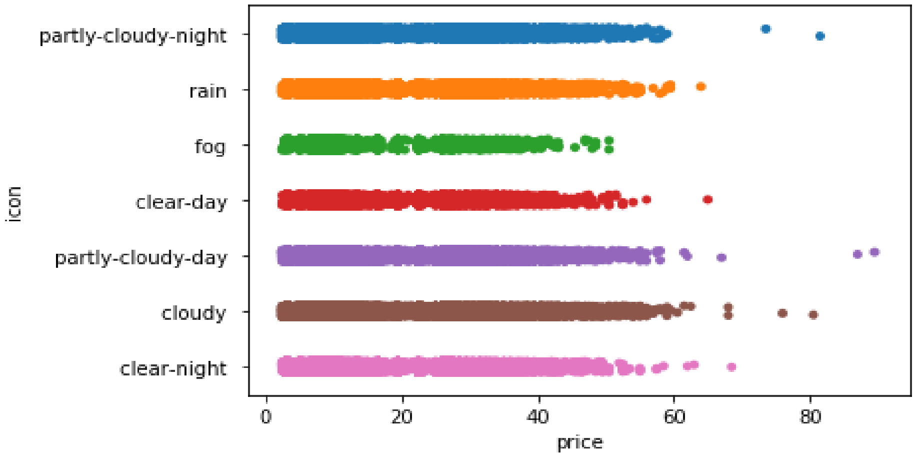


Fig. 4.3 Strip-plot between Icon and Price

From the above chart, it was clear there were some outliers in cloudy type weather, some data has an anonymously high price above 80 while the other was below 60. In this plot, we analyze that in cloudy-day weather price was the highest while in foggy weather price was minimum.

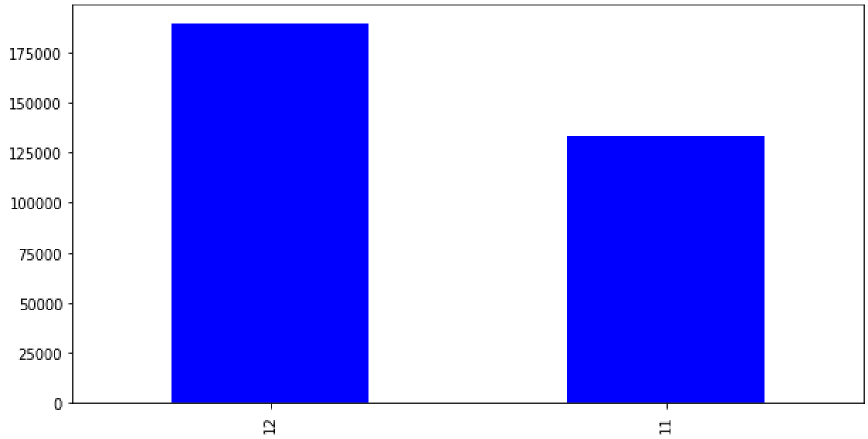


Fig. 4.4 Bar-Chart of Month

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.

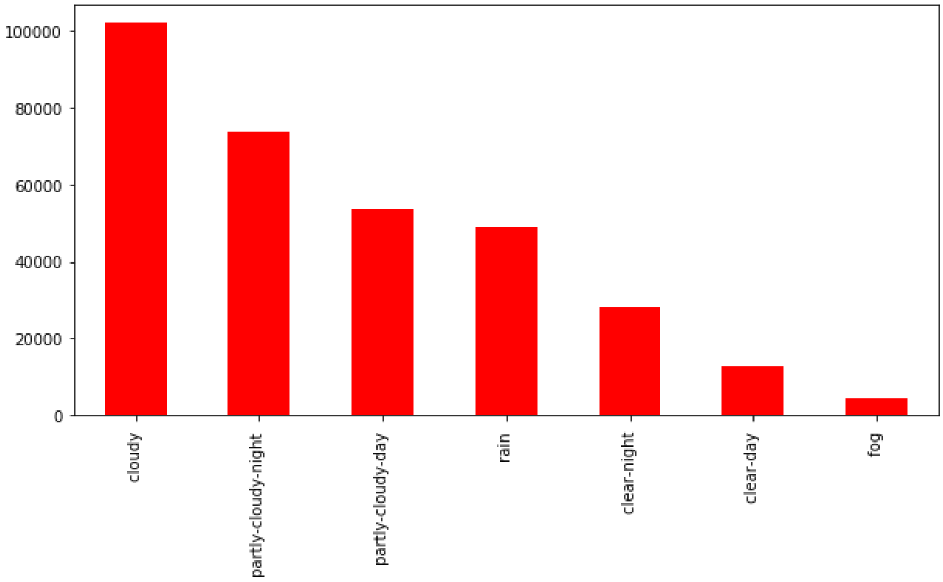


Fig. 4.5 Bar-Chart of Icon

The above bar chart represents the value count of the icon column and from the graph, it was clear that cloudy weather has the most data due to which we can say that may be in cloudy weather cab also opted most.

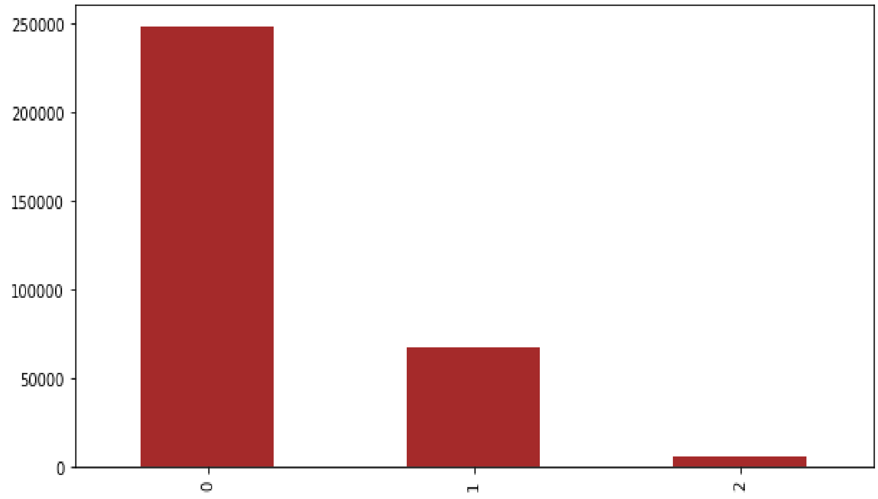


Fig.4.6 Bar-Chart of UV-Index

The above bar chart represents the value count of the UV-index column and from the graph, it was clear that when UV-index is 0, the dataset has the most data due to which we can say that when there is less UV-index cab was opted most.

4.6 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.

4.7 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.

There are two important configuration options when using RFE:

- The choice in the number of features to select (k value)
- The choice of the algorithm used to choose features.

RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remain. Hence RFE technique is effective at selecting those features (columns) in a training dataset that are most relevant in predicting the target variable.

4.7.1 Training Accuracy

We are implementing recursive feature elimination through scikit-learn via sklearn.feature_selection.RFE class.

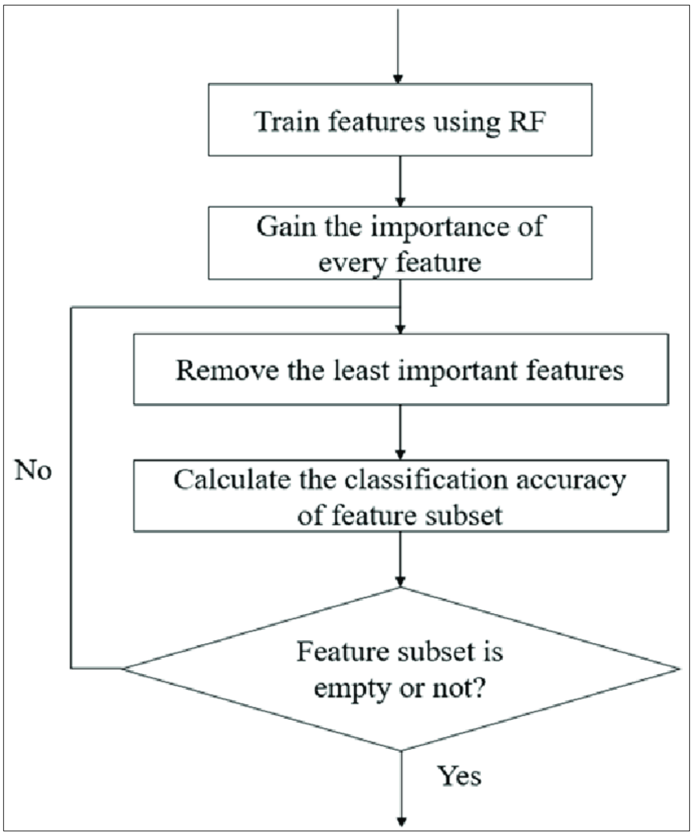

Train features using RF
Gain the importance of every feature
Remove the least important features
Calculate the classification accuracy of feature subset
Feature subset is empty or not?
No
Yes

Fig. 4.9 Recursive Feature Elimination Courtesy of Researchgate

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:

Table 4.1: RFE Accuracy Table

Serial No.	No. of Feature (K)	Accuracy
1	56	0.8631583766941027
2	40	0.8631583766941028
3	15	0.8631583766941028

4	25	0.8631583766941029

From the above table, it was clear that 25 features have the highest accuracy as compared to all other k values which mean these 25 features are the best features given by RFE. So, we only consider these 25 features for further working and rest we eliminate. Now our dataset reduces from 56 features to 25 features.

4.7.2 Drop Useless Columns

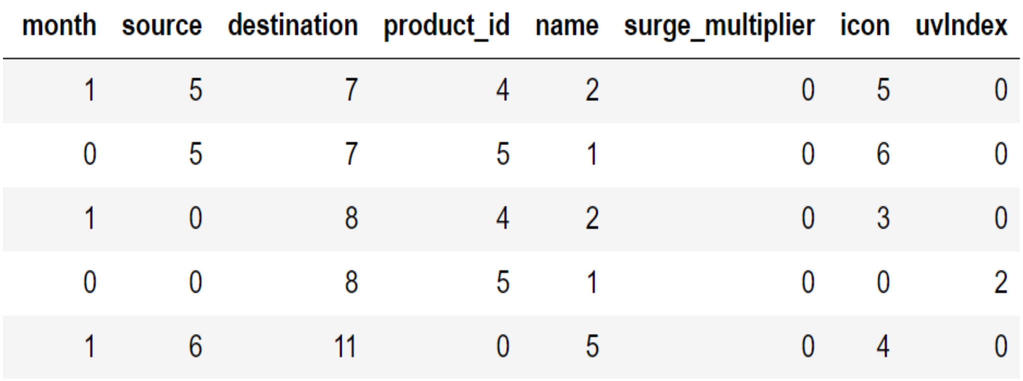
After applying RFE we get our 25 best features but still, there are many features which do not affect the price directly so we drop those features according to it. And eight features remained in our dataset. We use a method called drop() that removes rows or columns according to specific column names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

4.7.3 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.

4.7.4 Final Dataset

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.



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

Fig. 4.10 Final Dataset after Feature Engineering

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, Random Forest, and Gradient Boosting.

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. The name linear regression was come due to its graphical representation.

There are two types of Linear Regression:-

- **Simple Linear Regression-** In a simple linear regression algorithm the model shows the linear relationship between a dependent and a single independent variable. In this, the dependent variable must be a continuous value while the independent variable can be any continuous or categorical value.
- **Multiple Linear Regression-** In a multiple linear regression algorithm the model shows the linear relationship between a single dependent and more than one independent variable.
-

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. The order of the question as well as their content is being determined by the model. In addition, the questions asked are all in a True/False form. Here in our project, we are focusing on decision tree regression only. It is used for the continuous output problem. Continuous output means the output of the result is not discrete. It observes features of an object and trains a model in the structure of a tree to predict data that produce meaningful continuous output.

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. A more accurate prediction requires more trees, which results in a slower model. In most real-world applications the random forest algorithm is fast enough, but there can certainly be situations where run-time performance is important and other approaches would be preferred. A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which aggregates many decision trees, with some helpful modifications. Random forest first splits the dataset into n number of samples and then apply decision tree on each sample individually. After that, the final result is that predicted accuracy whose majority is higher among all.

Random Forest depends on the concept of ensemble learning. An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. A model comprised of many models is called an Ensemble model.

Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between those trees while building random forest model.

4.8.4. Gradient Boosting

Gradient boosting is a technique which can be used for both classification and regression problem. This model combines the predictions from multiple decision trees to generate the final predictions. Also, each node in every other decision tree takes a different subset of features for selecting the best split. But there is a slight difference in gradient boosting in comparison to random forest that is gradient boosting builds one tree at a time and combines the results along the way. Also, it gives better performance than random forest. The idea of gradient boosting originated in the observation by Leo Breiman that boosting can be interpreted as an optimization algorithm on a suitable cost function. Gradient Boosting trains many models in a gradual, additive, and sequential manner.

The modeling is done in the following steps:-

- First, we split the dataset into a training set and a testing set.
- Then we train the model on the training set.
- And at last, we test the model on the testing set and evaluate how well our model performs.

So after applying these models we get the following accuracy:

Table 4.2: Model Accuracy Table

Serial No.	Models	Accuracy
1	Linear Regression	0.7578915690209413
2	Decision Tree	0.9605524230411967
3	Random Forest	0.9610441804498846
4	Gradient Boosting Regressor	0.9621167871791544

4.8.9 K-fold Cross Validation

We also apply cross validation using linear regression algorithm. It is a technique where the datasets are split into multiple subsets and learning models are trained and evaluated on these subset data. It is a resampling procedure used to evaluate machine learning models on a limited data sample. It is one of the most widely used technique. In this, the dataset is divided into k-subsets (folds) and are used for training and validation purpose for *k iteration* times. Each subsample will be used at least once as a validation dataset and the remaining (*k-1*) as the training dataset. Once all the iterations are completed, one can calculate the average prediction rate for each model. The error estimation is averaged over all k trials to get the total effectiveness of our model.

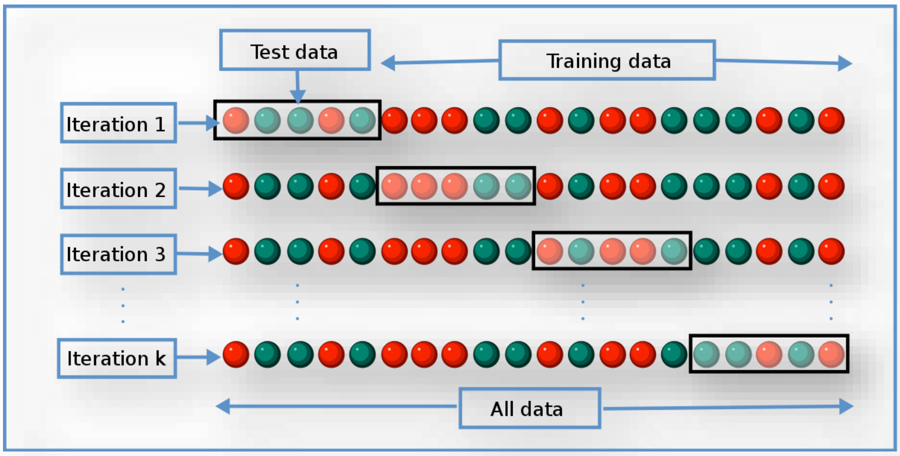


Fig. 4.11 Cross-Validation Courtesy of Wikimedia

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 (ranges 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 our project, we perform testing on two models: Linear Regression and Random Forest.

Linear Regression Model Testing:

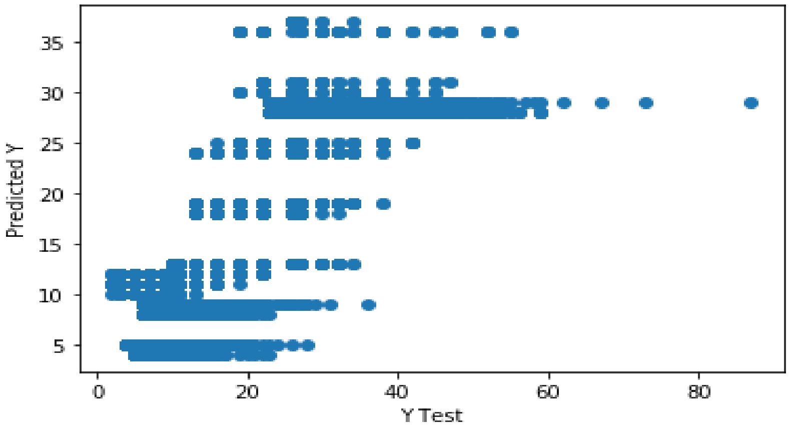


Fig. 4.12 Scatter Plot for Linear Regression

We draw a scatter plot between predicted and tested values and then find errors like MSE, MAE, and RMSE. After that, we also draw a distribution plot of the difference between actual and predicted values using the seaborn library. A distplot or distribution plot represents the overall distribution of continuous data variables.

Table 4.3: Error table for Linear Regression

Serial No.	Models	Accuracy
1	Mean Absolute Error	3.0513559138286173
2	Mean Squared Error	18.61467577320386
3	Root Mean Absolute Error	4.31447282680096

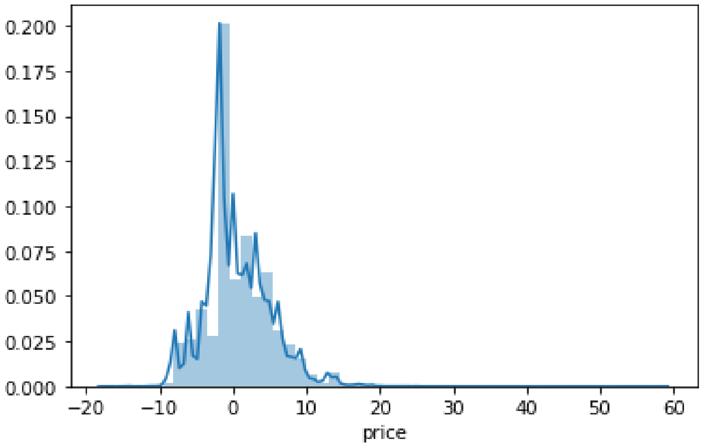


Fig. 4.13 Dist Plot for Linear Regression

Random Forest Model Testing:

Similarly, we draw scatter plot, dist plot, and find all three errors for random forest also.

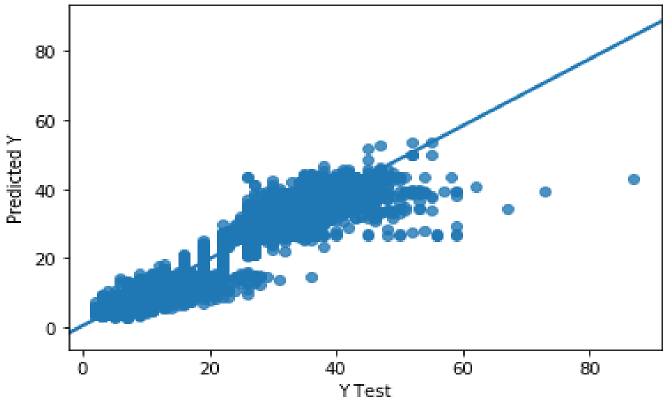


Fig. 4.14 Scatter Plot for Random Forest

Table 4.4 Error table for Random Forest

Serial No.	Models	Accuracy
1	Mean Absolute Error	0.998140620865906
2	Mean Squared Error	2.9447657109843504
3	Root Mean Absolute Error	1.7160319667722832

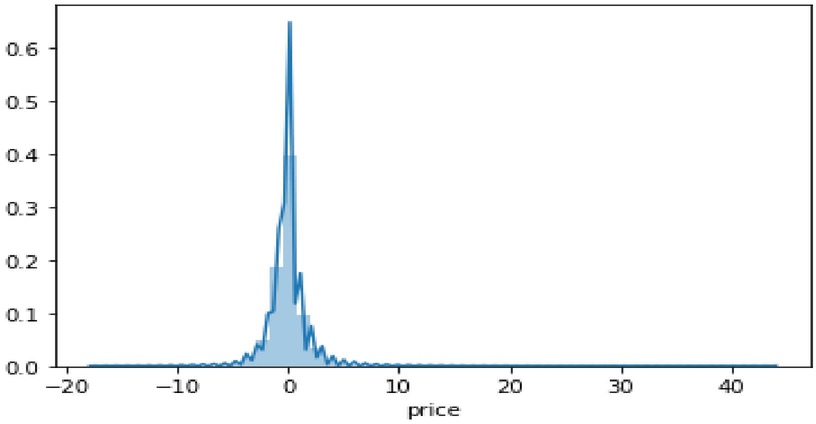


Fig. 4.15 Dist Plot for Random Forest

4.10 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 thelength 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.

4.11 Use of Price Prediction Function

To use the "Predict Price" function for price prediction, follow these steps:

1. Import the necessary libraries and load the trained random forest model.
2. Define the "Predict Price" function that takes the following four parameters as input: cab name, source, surge multiplier, and icon (weather). Ensure that the input parameters are in the expected data type (integer).
3. Inside the function, search for the rows in the dataset that match the input cab name and extract their row numbers.

4. Create an array, "x," with the same length as the dataset and initialize all values to zero.

5. Assign the input values of source, surge multiplier, and icon to their respective indices in the "x" array.

6. Check the count of the desired rows. If it is greater than zero, assign the value 1 to the corresponding index in the "x" array.

7. Use the trained random forest model's predict function to predict the price using the "x" array.

8. Return the predicted price as the output of the function.

To use the function, follow these instructions:

1. Ensure you have the required inputs: cab name, source, surge multiplier, and icon (weather).

2. Convert the cab name, source, surge multiplier, and icon values to the expected data type (integer).

3. Call the "Predict Price" function, passing the converted inputs as arguments.

4. Capture the returned predicted price as the output.

Make sure to replace `'random_forest_model.pkl'` with the correct filename/path of your trained random forest model. Adjust the indices and dataset based on your specific data structure.

By following these steps, you can utilize the "Predict Price" function to obtain price predictions based on the provided input parameters.

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 25 features were selected. Among those 25 features still, there are some features which we think are not that important to predict the price so we drop them and left with 8 important columns.

We apply four different models on our remaining dataset among which Decision Tree, Random Forest, and Gradient Boosting Regressor 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.

FUTURE ENHANCEMENTS

Based on the conclusion drawn from the analysis of Uber's data, there are several potential future enhancements that can be considered for the data analysis project:

1 Incorporate Price Data: Since the current analysis revealed that there is no data available for taxi prices, one potential enhancement would be to gather and include pricing information in the dataset. This could provide valuable insights into the relationship between price and other variables, allowing for more accurate predictions and a deeper understanding of pricing dynamics.

2. Include Other Cab Price Variations: The analysis mentioned that there was no information on the price variations of other cabs. Adding this data to the dataset could further enhance the predictive power of the model, as it would allow for comparisons and insights into how Uber's pricing compares to other transportation options.

3. Incorporate Weather Data: Weather conditions can significantly impact demand and pricing in the transportation industry. Integrating weather data into the analysis could provide insights into how different weather patterns affect Uber's operations and pricing. This information could be leveraged to optimize pricing strategies during specific weather conditions.

4. Explore Additional Feature Engineering: Although the analysis selected the top 25 features using Recursive Feature Elimination (RFE), there may be other relevant features that were not considered. Future enhancements could involve exploring additional feature engineering techniques, such as dimensionality reduction or incorporating external data sources, to identify and include additional informative variables.

5. Evaluate Other Models: While the analysis identified Decision Tree, Random Forest, and Gradient Boosting Regressor as the best-performing models, there may be other algorithms that could yield even better results. It would be beneficial to evaluate and compare the performance of other machine learning models, such as Support Vector Machines or Neural Networks, to ensure the selected model is truly optimal for the dataset.

6. Fine-tune Model Hyperparameters: Hyperparameter tuning can significantly improve the performance of machine learning models. Conducting a systematic search for optimal hyperparameters for the chosen Random Forest model could potentially further improve its accuracy and generalization capabilities.

7. Implement Online Price Prediction Functionality: The analysis mentioned designing a function using the Random Forest model to predict prices. One future enhancement could be to implement this prediction functionality in a real-time or online setting. This would enable the model to provide price predictions in real-time as new data becomes available, allowing for dynamic pricing strategies and more accurate fare estimates for users.

8. Continuously Update and Refine the Model: Data-driven models benefit from continuous updates and refinements as new data becomes available. It is crucial to establish a process to regularly update the model, retrain it with the latest data, and incorporate new insights and features to maintain its accuracy and relevance over time.

By considering these future enhancements, Uber's data analysis project can continue to evolve and provide valuable insights for optimizing pricing strategies, understanding customer behavior, and improving overall operational efficiency.

LIMITATIONS OF PROJECT

1. Limited insights: The analysis was based on limited data insights obtained from exploratory data analysis (EDA). This means that the conclusions drawn may not fully capture the complexity of the problem or capture all relevant factors that influence taxi prices.

2. Lack of data on taxi prices: The dataset used in the analysis did not have information on the actual prices of taxis. This limitation restricts the accuracy of the predictions made by the model and raises questions about the reliability of the findings.

3. Lack of data on price variations of other cabs: The analysis did not consider the price variations of other types of cabs. This limitation can affect the accuracy of the predictions, as different types of cabs may have different pricing patterns.

4. Limited data on weather: The dataset used in the analysis did not include information on weather conditions. This limitation can impact the accuracy of the predictions, as weather conditions can influence taxi demand and pricing.

5. Subjective feature selection: The feature selection process was based on subjective judgments, dropping certain features based on perceived importance. This limitation introduces bias and may result in the exclusion of potentially important features for predicting taxi prices.

6. Limited number of features: The final model was built using only 8 features, which may not capture all relevant factors influencing taxi prices. This limitation could affect the accuracy and reliability of the model's predictions.

7. Lack of external validation: The analysis did not include external validation of the model's performance using a separate dataset. This limitation raises concerns about the generalizability of the findings and the model's ability to accurately predict taxi prices in real-world scenarios.

8. Assumptions about algorithm performance: The conclusion states that Decision Tree, Random Forest, and Gradient Boosting Regressor performed best with high accuracy. However, this conclusion is based on assumptions and may not hold true when applied to different datasets or scenarios.

REFERENCES

- Abel Brodeurand & Kerry Nield An empirical analysis of taxi, Lyft and Uber rides: Evidence from weather shocks in NYC
- https://www.singlegrain.com/blog-posts/business/10-lessons-startups-can-learn-ubers-growth/
- https://www.researchgate.net/publication/305524879_Dynamic_Pricing_in_a_Labor_Market_Surge_Pricing_and_Flexible_Work_on_the_Uber_Platform
- https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789808452/1/ch01lvl1sec19/label-encoding
- https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_algorithms_performance_metrics.htm
- https://blog.paperspace.com/implementing-gradient-boosting-regression-python/
- https://www.studytonight.com/post/what-is-mean-squared-error-mean-absolute-error-root-mean-squared-error-and-r-squared
- https://statisticsbyjim.com/regression/overfitting-regression-models/