# Uber and Lyft cab prices prediction

***1.1 Abstract***

In this project, we are analyzing data of two well-known cab service providers Uber and Lyft in the US. Both these companies offer ride-hailing services that allow users to hire vehicles with drivers through websites or mobile applications. Whether it is a short distance ride, or you are traveling from one city to the other, these cab services have helped in numerous ways to people and literally made their life a lot easier.

From booking the rented car to paying the bill, both cab services have almost similar core features. But there are some unique cases where these two cab services are in competition with each other. There are many factors affecting the prices for Lyft and Uber. One such is pricing, especially dynamic pricing called “surge” in Uber and “Prime Time” in Lyft.

***1.2 Hypothesis and key questions***

The key question we formulated were:

1. Which is the most booked cab type?
2. Price comparison between the providers based on average distances.
3. What is the average price for different cab type group (i.e., SUV, UberX)?

The hypothesis we want to drive from the cabs\_rides dataset is **“Uber is more expensive than Lyft”.**But after all the analysis our hypothesis has been proved wrong.

***1.3 Tools Used***

Tools that were used in the project were:

1. Pandas’ library in Python using Jupyter Notebook for data cleaning, modelling, and exploratory analysis.
2. Tableau for visualizing the data

***Data Acquisition***

The data has been collected from different sources, including real-time data collection using Uber and Lyft API (Application Programming Interface) queries. The dataset covers Boston’s selected locations, to make accurate visualization and model we need clean data, which should not have nulls or missing values, for which we are going to perform preprocessing techniques.

***Data Description***

The sources we are going to use in this project are collected from Uber and Lyft API queries. This data set is of cab rides collected for a week in Nov - Dec '18 collected at a regular interval of 5 mins and has 10 attributes and data types as mentioned below.

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Data** T**ype** | **Description** |
| Distance | Numeric | The attribute distance is the distance between source and destination. |
| Cab\_type | Categorical | This attribute is a choice between the provider uber or lyft. |
| Time\_stamp | Numeric | The time when the data was queried, means was taken to perform the analysis. |
| Destination | Categorical | The destination for the ride. |
| Source | Categorical | The starting point for the ride. |
| Price | Numeric | The cost for the booking in USD. |
| Surge multiplier | Numeric | The multiplier by which the price was increased. |
| Id | Categorical | The booking id. |
| Product id | Categorical | Unique identification for the Cab\_type |
| Name | Categorical | The visible type of the cab, example: Uberpool, Uberxl. |

***Data Preparation***

Decided to change the column name to be short-term and more meaningful and eliminate some columns which were not useful.

Data Processing and cleaning is a crucial step in any data analysis problem. It is often a very good idea to prepare your data in such a way to best expose the structure of problem to machine learning algorithms that you intend to use. The null and duplicate values in the dataset are removed.

Used .str() method to clean the columns. This attribute is a way to access speedy string operations in Pandas that largely mimic operations on native Python strings or compiled regular expressions, such as. split(), .replace(), and .capitalize(). With .str.string(),removing any leading and trailing spaces in the columns data.

**Graphical user interface, application

Description automatically generated**

***Exploratory Data Analysis***

Exploratory Data Analysis plays a key role in understanding the data before performing any further analysis. To better understand the dataset, doing exploratory analysis on the data to analyze the features and try to understand which feature has large predictive value and which does not bring considerable predictive value. In the process, built some plots and did correlation and heat map for better understanding.

Calculated the mean value for Distance and the median value for Price data to understand how the data is scattered in the complete dataset in terms of distance and price.

Graphical user interface, chart, application, treemap chart

Description automatically generated

Plotted a correlation matrix for the price, distance, and surge\_multipler for the Uber and Lyft data. As shown in the below diagram, Price and Distance are correlated in a moderate manner. Similarly, Distance and Surge\_multipler are correlated in a moderate manner.

***Incorporating Models and Algorithms***

Used ***Linear regression*** model which is probably one of the most important and widely used regression techniques. It’s among the simplest regression methods. One of its main advantages is the ease of interpreting results.

With Python by using training data which was preprocessed, when implementing linear regression of **Distance** dependent variable 𝑦 on the set of **Price** variables 𝐱 = (𝑥1, ..., 𝑥r), where 𝑟 is the number of predictors, assume a linear relationship between 𝑦 and 𝐱: 𝑦 = 𝛽0 + 𝛽1𝑥1 + ⋯ + 𝛽r𝑥r + 𝜀.

R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

In our model, Test

(R2) score is 0.92 which says the Distance, Price, time\_stamp, SurgeMultiplier attributes performance moves relatively in line with the index. So, the model got correctly fitted into the data.

Graphical user interface, text

Description automatically generated

***Visualizations***

Used tableau visualization software to visually represent the answers to the key questions framed.

Reported which is the most booked cab type in Lyft and Uber. Grouped the data of CabType column and calculated the occurrence with the help of python code. Used bar chart to plot the visualization for this data. From the results, UberXL, WAV, Black SUV are the most booked cab type with total booking of 55096 compared to other cab types.

Bar graph would help **to compare Total number of rides between different groups of car types which tracks the difference of their ride count**. When trying to analyze change over time, however, bar graphs function better when the changes are considerable.

Therefore, Bar graph helps to visualize relationships between numerical variables by using various math functions like (Min, Max, count, etc.,) Average function would help us by relating Price and Distance to get an estimated value. Where, the plot needs to be placed at least one measure on the Columns shelf and at least one measure on the Rows shelf which greatly works for two variables (Price and distance).

Chart, bar chart

Description automatically generated

**Figure 1: Bar graph showing Most booked cab types in Lyft and Uber – Using PYTHON**

The below graph is plotted between counts of rides and service provider. The graphs show a greater number of Uber services booked than Lyft. Rides booked for Uber were 78,255 more in count than Lyft for November 2018 in Boston.

Chart, bar chart

Description automatically generated

**Figure 2: Bar graph showing total ride count for Uber and Lyft – Using PYTHON**

Used price and distance attributes to get the relation between them. It is used to find the estimated price for distance. For this, used Average function in tableau for both Price and Distance (Average Price on y-axis and Average Distance on x- axis) to show a scatter plot for both providers.

From the visualization, observed that the LYFT provided service for price $17.35 for 2.19 miles and UBER provides service for price $15.80 for 2.19 miles. Where Uber price is cheaper. But there are other factors to consider concluding which provider gives the best service.



**Figure 3: Scatter plot showing Average Distance Vs Average Price for Uber Lyft – Using Tableau**

Tried to visualize the Average price for each cab type. Here, considered Cab Type and Price attributes to get the process work. For Price, used Average function.

Below are the calculations of the obtained Average Price for each cab type. Observed “***LUX Black XL”*** has the highest price and “***UberPool and Shared*** has the lowest prices.

Tree maps are visualizations for hierarchical data using various colors for better understanding and:

They capture two types of information in the data:

* The value of individual data points
* the structure of the hierarchy

By seeing the hierarchical structure, identifying the highest and lowest points (In our case, the highest and lowest priced cab type).

Chart, treemap chart

Description automatically generated

**Figure 4: Tree Map showing Price Variation of each Cab Type in hierarchical way – Using Tableau**

***Results***

From the fig 3, scatter plot, LYFT is expensive than Uber.

***Lesson learnt***

Should have chosen a dataset with broader scope so that we could have explored the dataset and its parameters with more visualizations.

***Citation***

“Uber Related Data Analysis Using Machine Learning.” *IEEE Xplore*, ieeexplore.ieee.org/document/9432347/keywords#keywords.

RaviMunde. “Uber & Lyft Cab Prices.” *Kaggle*, 23 June 2019, www.kaggle.com/ravi72munde/uber-lyft-cab-prices.

bking2415. “bking2415/Uber-Data-Analysis: Data Visualization Concepts in Tableau and R.” *GitHub*, github.com/bking2415/Uber-data-analysis.

“Uber and Lyft Cab Prices: Data Analysis and Visualization.” *Analytics Vidhya*, 17 June 2021, www.analyticsvidhya.com/blog/2021/06/uber-and-lyft-cab-prices-data-analysis-and-visualization/.

(<https://www.analyticsvidhya.com/blog/2021/06/uber-and-lyft-cab-prices-data-analysis-and-visualization/>). We will use data already compiled from this API provided on Kaggle (<https://www.kaggle.com/ravi72munde/uber-lyft-cab-prices>).

<https://help.tableau.com/current/guides/get-started-tutorial/en-us/get-started-tutorial-home.htm>

<https://docs.python.org/3/tutorial/>