Uber and Lyft Price Prediction

- Team 05 -

Preston Chen | Megan Crawford | Mitch Feren | Alex Sampson

Fall 2021

CONTENTS

I. Introduction

As the world continues to modernize, Uber and Lyft have dominated the market as players in the ride-sharing industry. Uber currently has around 93 million monthly active users around the world and have sourced around 6 billion rides per year[1]. Rising inflation and a labor shortage have caused the prices of Ubers and Lyfts to skyrocket in the past year, leaving consumers with fewer, more expensive options. According to the research Firm Rakuten Intelligence, the price of an Ubers and Lyfts in April 2021 was up 40% from the previous year, while the number of drivers was down 22%[2]. Lyft has emerged as the top competitor to Uber and continues to battle to secure drivers and riders in an attempt to wrestle the industry out of Uber's hands. Pricing of these services vary heavily and ultimately, many factors play into the final pricing of a cab ride. Uber and Lyft do not publish their own data on ride prices, so it can be difficult for consumers to accurately predict what their ride price will be. How should consumers approach this battle? What can consumers expect to pay with differing situations?

II. What we are exploring

First, we will identify the most important factors in determining the price of an Uber or a Lyft. The results will allow consumers to approximate the price of a ride given different circumstances in planning their commutes. Additionally, these results will help inform our prediction model for predicting the price of each app's rides.

Second, we'd like to identify which app is best for ordering a shared ride. To accomplish this goal, we plan on using a linear regression model to predict the price of each provider's ride given a set of conditions. We will assume our riders are deciding which app based on the price offered, and by comparing these prices, we will assume our riders will choose the cheaper option. This ride analysis can be useful when deciding between which app to use, as knowledge about the price and the factors influencing this price, from the previous analysis, can lead to a more informed decision.

Finally, on the driver side, we would like to synthesize the two previous analyses to form a location and time recommendation that will lead to the greatest captured income for drivers. By learning which factors have the most influence on the price, and what this predicted price is, we can recommend specific pickup and drop-off locations to drivers, as well as the best times to drive for Uber and Lyft.

III. Data

In order to collect proprietary data with the intention of predicting Uber and Lyft prices using weather and ride data, the project owners built a custom application in Scala to query data at regular intervals. They queried prices every 5 minutes and weather data every hour to simulate the prices charged by each of the companies. The dataset contains 693,072 observations of prices and 6,277 observations of weather conditions in the Greater Boston area from the end of November 2018 to the beginning of December 2019, spanning a little more than a week.

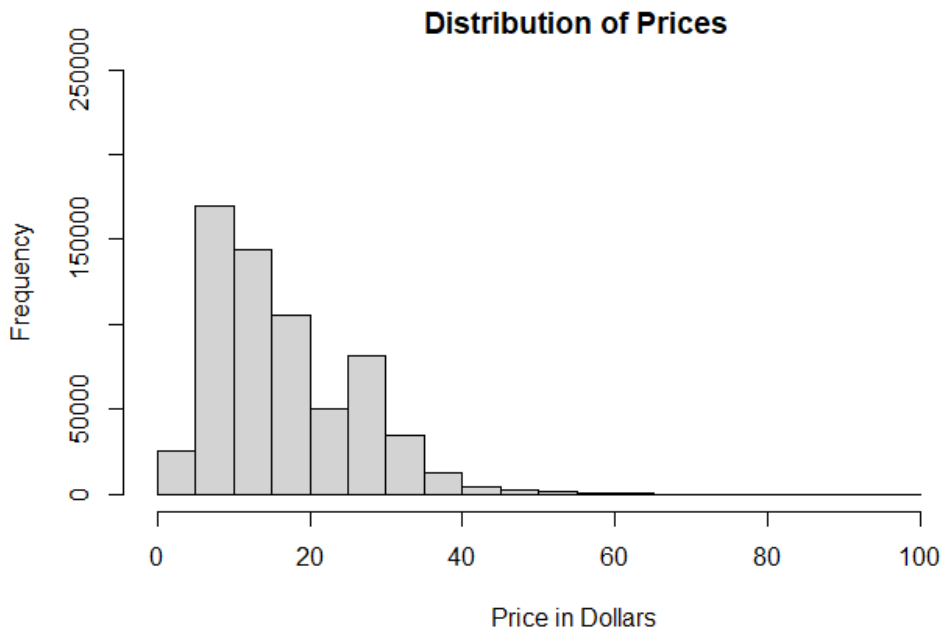
Each row of the set represents a single ride with details on the distance, source, and type of ride that was called. Additionally, a separate data set with the weather for the Boston area was included on Kaggle. The weather data set includes data on the temperature, location, clouds, pressure, rain, humidity, wind and time. We merged the ride data with the weather data to produce a master data set which we used for analysis and modeling. Overall, the outcome variable for the data set is the price of the ride.

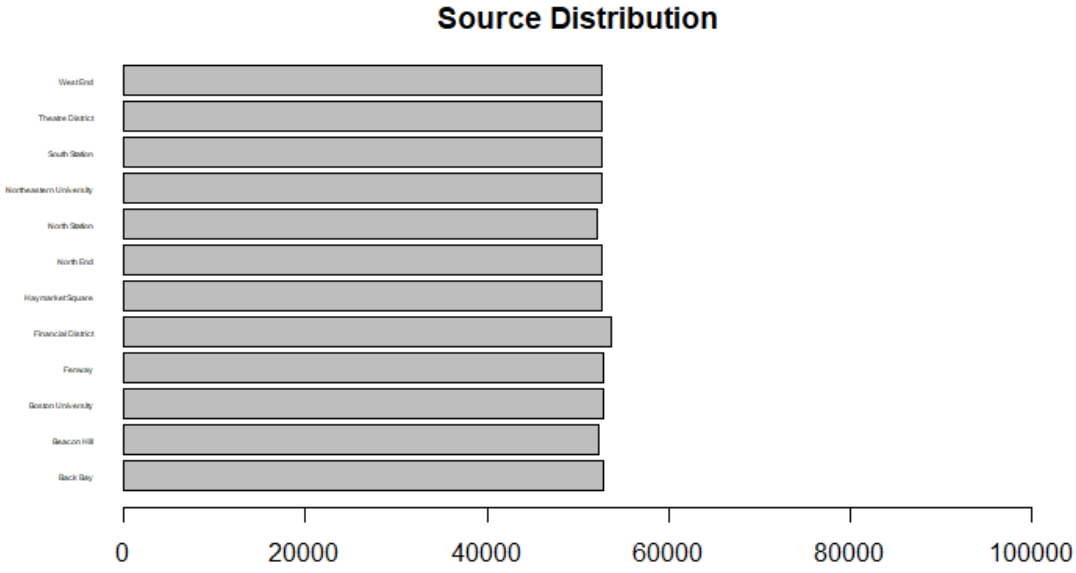
The specific variables in our dataset are: distance, cab type (either Uber or Lyft), time, destination, source, price, surge multiplier, temperature, clouds, pressure, rain, and humidity.

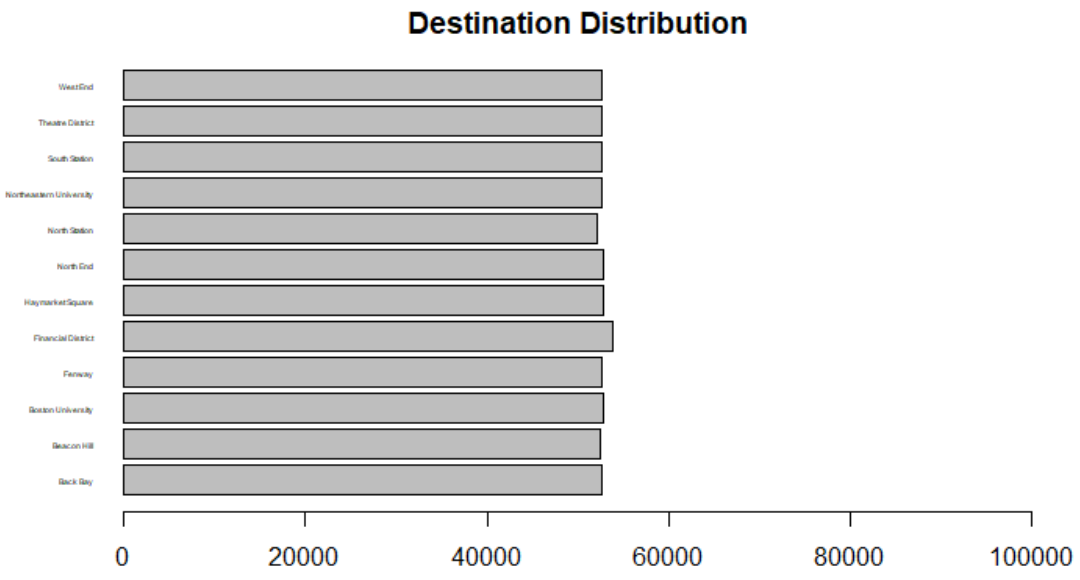
IV. Exploratory Data Analysis

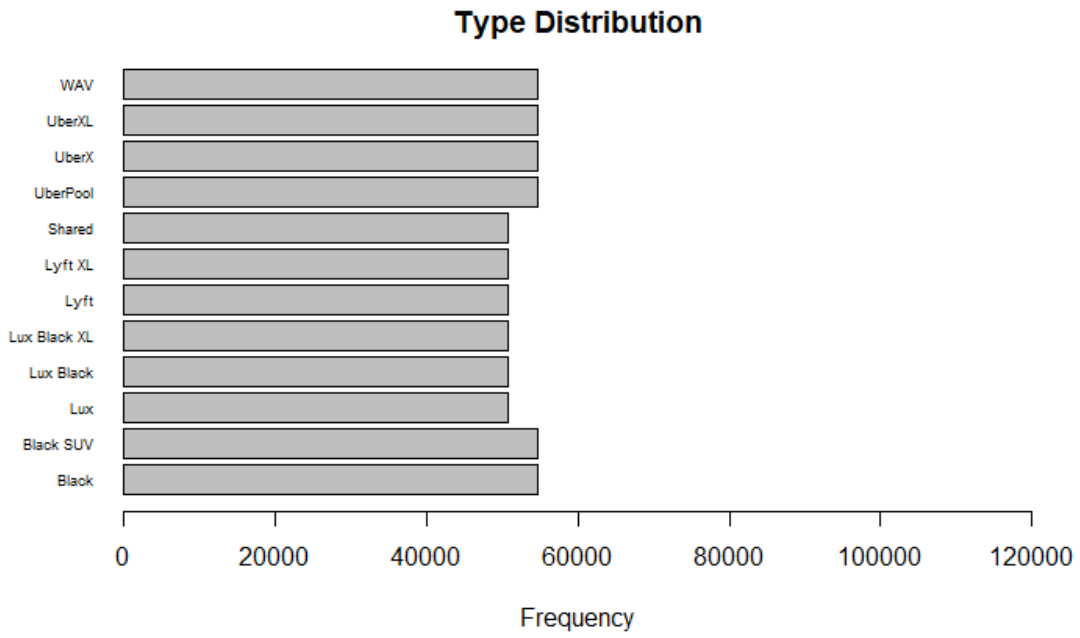
A. Data Visualizations

Because of the size and variability of our dataset, visualizations will be a necessary tool to help identify outliers, intercorrelations, and summary information about the distribution of our data. Our plan is to utilize histograms, correlation matrices, and barplots to help explore our dataset and communicate our findings through our analysis. Specifically, we analyzed the distribution of prices to learn more about how frequent certain prices occurred in our Boston dataset. We also visualized the distribution of source and destination locations by their frequency in our dataset. Additionally, we engaged in PCA to gain a more accurate and holistic understanding of their interactions, after looking at some of the summary statistics in our data. Finally, we used another barplot to compare the frequency of Uber and Lyft ride types within our dataset before splitting each into its own model progression.

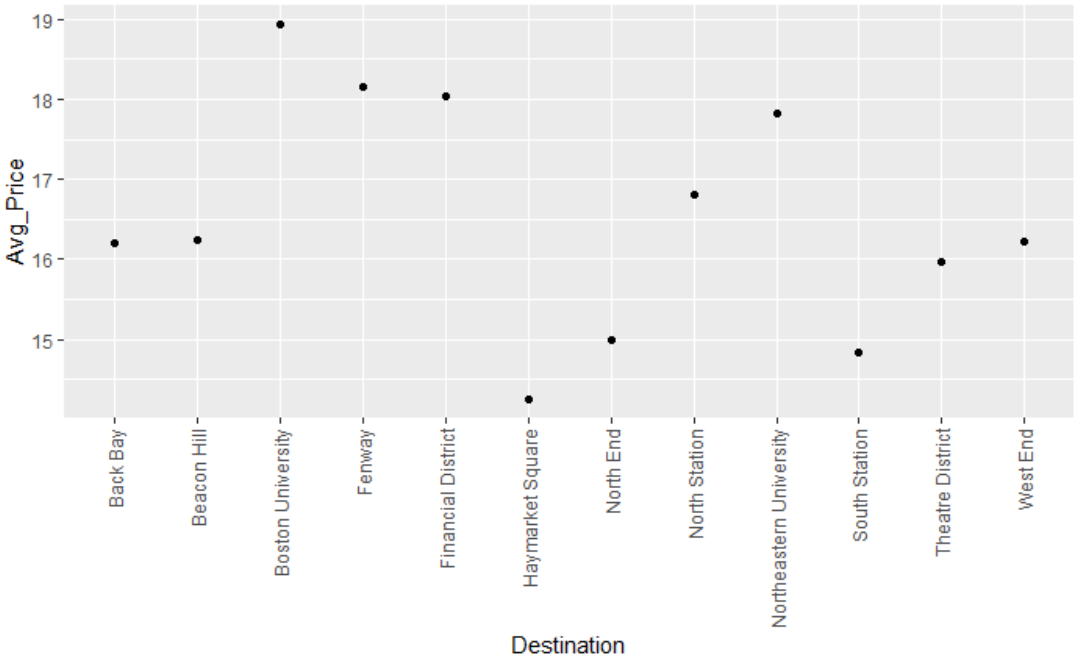

Distribution of Prices
Frequency
0
50000
150000
250000
0
20
40
60
80
100
Price in Dollars


Source Distribution
West End
Theatre District
South Station
Northeastern University
North Station
North End
Haymarket Square
Financial District
Fenway
Boston University
Beacon Hill
Back Bay
0
20000
40000
60000
80000
100000

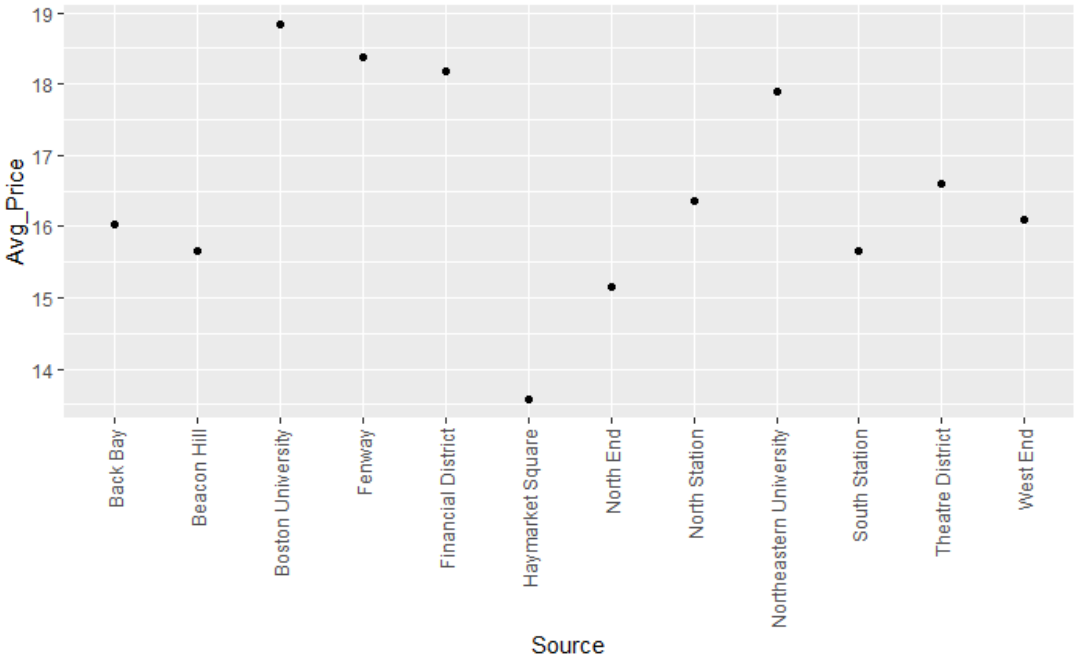

Destination Distribution
West End
Theatre District
South Station
Northeastern University
North Station
North End
Haymarket Square
Financial District
Fenway
Boston University
Beacon Hill
Back Bay
0
20000
40000
60000
80000
100000

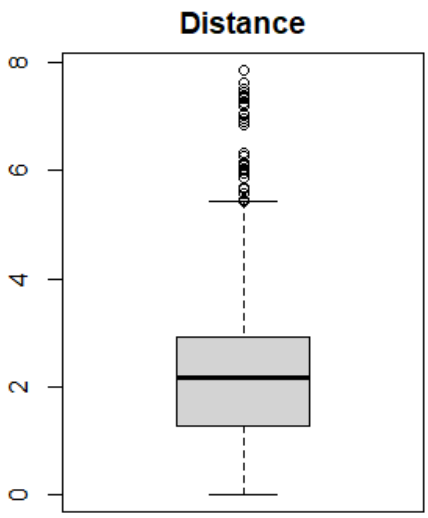

Type Distribution
WAV
UberXL
UberX
UberPool
Shared
Lyft XL
Lyft
Lux Black XL
Lux Black
Lux
Black SUV
Black
0
20000
40000
60000
80000
100000
120000
Frequency

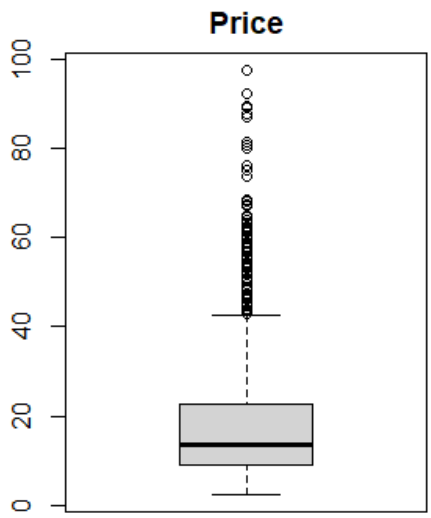
Average Price By Destination

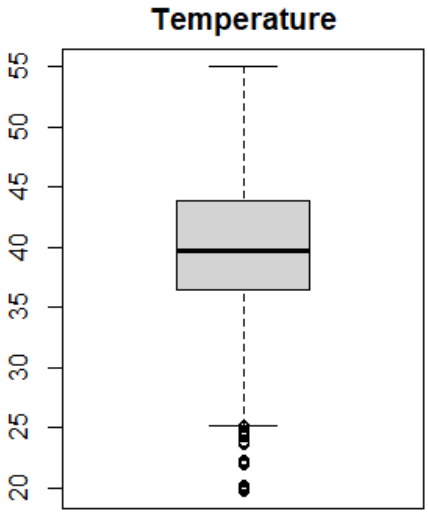


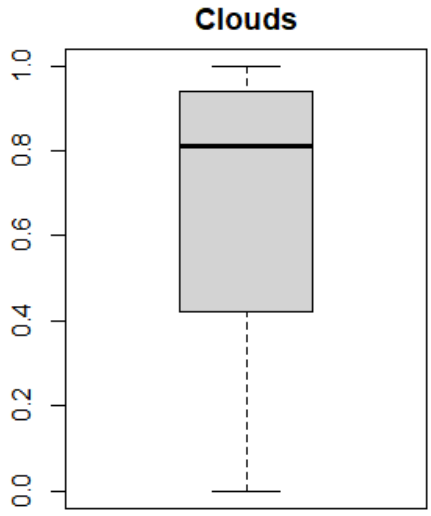
Average Price By Source

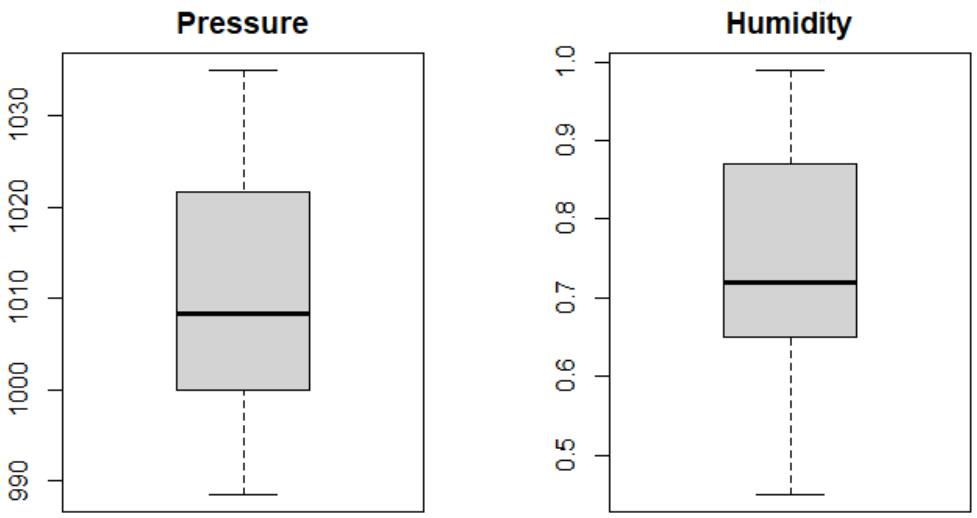



Distance
0
2
4
6
8

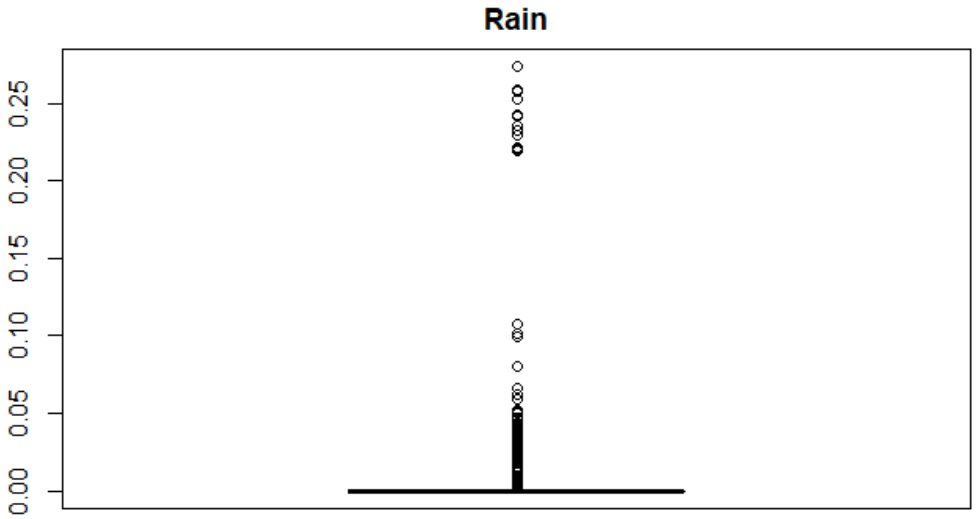

Price
0
20
40
60
80
100


Temperature
20
25
30
35
40
45
50
55


Clouds
0.0
0.2
0.4
0.6
0.8
1.0


Pressure
990
1000
1010
1020
1030

Humidity
0.5
0.6
0.7
0.8
0.9
1.0


Rain
0.00
0.05
0.10
0.15
0.20
0.25

B. Data Summaries

We utilized the summary function in R to learn more about the scope and values within each numerical column of our dataset. An initial overview of these numerical columns revealed that there may be outliers in the price column because of the large maximum value of $97.5 compared to the median ($13.50) and the mean ($16.54).

```
    distance         price            temp          clouds          pressure
 Min.   :0.020   Min.   : 2.50   Min.   :19.78   Min.   :0.0000   Min.   : 988.6
 1st Qu.:1.270   1st Qu.: 9.00   1st Qu.:36.42   1st Qu.:0.4200   1st Qu.: 999.9
 Median :2.160   Median :13.50   Median :39.71   Median :0.8100   Median :1008.4
 Mean   :2.189   Mean   :16.54   Mean   :39.33   Mean   :0.6669   Mean   :1010.3
 3rd Qu.:2.930   3rd Qu.:22.50   3rd Qu.:43.88   3rd Qu.:0.9400   3rd Qu.:1021.7
 Max.   :7.860   Max.   :97.50   Max.   :55.06   Max.   :1.0000   Max.   :1035.1
      rain             humidity           wind             day              hour
 Min.   :0.000000   Min.   :0.4500   Min.   : 0.300   Min.   :0.000   Min.   : 0.00
 1st Qu.:0.000000   1st Qu.:0.6500   1st Qu.: 3.620   1st Qu.:1.000   1st Qu.: 5.00
 Median :0.000000   Median :0.7200   Median : 6.670   Median :3.000   Median :12.00
 Mean   :0.005805   Mean   :0.7465   Mean   : 6.753   Mean   :2.786   Mean   :11.46
 3rd Qu.:0.000000   3rd Qu.:0.8700   3rd Qu.: 9.760   3rd Qu.:5.000   3rd Qu.:17.00
 Max.   :0.274000   Max.   :0.9900   Max.   :18.180   Max.   :6.000   Max.   :23.00
```

C. Data Preparation

To prep the dataset, we needed to merge our weather data with our rideshare simulated data. Using Python, we loaded both csv files into our Jupyter Notebook environment, into two separate objects. After dealing with our missing values by filling some of the weather data with zeros, and dropping rides with null values, we were able to begin converting our timestamp into a workable format. Using the to_datetime function within the pandas package, we converted the time stamp to a traditional YYYY-MM-DD HH:MM:SS format for the rides and weather dataframes. Then, we concatenated this time stamp with our location, to get a more unique record of a specific time and place. Using this column as our matching value, we then merged the dataframes, matching specific weather data from times and places to the corresponding source location from our rides dataframe. We used an inner join to do so, created a new column named “date_time”, and further validated the data by dropping additional null values in the temperature and location columns. This left us with a merged dataset of the weather and rideshare data.

After the merge and resizing of the data was complete, duplicated rows were deleted, leaving only one of each ride ID (id column) and the weather from that location and time. NA values were also omitted from the data set. Next, rows with an outlier in the price column were removed and the data was split into two sets: uber rides and lyft rides. Our dataset was too large for some of our data manipulation, so we took a random sample of 50,000 observations for Uber and Lyft, leaving us with 100,000 total observations. This should still provide a random sample, as the queried data had no time series component. Finally, training sets were created using a random sample of 70% of each set. The validation sets (30% of each set) were created using the rows that were not included in the training sets.

V. Data Reduction

A. Variable Selection in our Model

For both Uber and Lyft, we ran a backward elimination method on training data from the respective training data sets to select the variables that were most important in predicting the price of a ride.

For the Uber training data set, the backward elimination yielded a prediction of price based on the source, distance, type of Uber (name column), humidity, and wind. For the Lyft training data set, the same method yielded a prediction of price based on the source, distance, type of Lyft(name column), surge multiplier, and hour.

A forward selection algorithm yielded all of the variables included in the original model for both the Uber and Lyft models. While running a stepwise selection yielded the same selected variables as our backward elimination and an exhaustive search was not feasible due to the large number of variables in the data set. When comparing the accuracy of each of these methods, the differences in accuracy were negligible. There was under .01 difference in RMSE between each method for both Uber and Lyft models.

B. Principal Component Analysis & Regression

We decided to implement PCA in order to reduce the number of numerical variables, as we suspected multicollinearity between a lot of our weather variables[3]. In order to perform PCA, we excluded our categorical variables, which included our time stamps, id numbers, source, destination, and product id. Initially, we ran the model without scaling, and found that pressure accounted for most of the variation. Upon performing PCA, we found there to be significant correlations between the variables, so we decided to implement a regression model.

In order to build a linear regression model with the principal components, we loaded the package “pls”[4]. Once again, we standardized the data and built different models for Uber and Lyft. Each model contained 4 principal components which accounted for 90% of the variance.

Unsurprisingly, PCR did not improve upon the accuracy of our other models. This is because source and destination, two extremely important factors for determining the price, are categorical variables and thus not considered by PCR. Although our numerical variables may have had multicollinearity, we were still able to create a better model without removing the correlations.

VI. Summary

A. Predictive Accuracy

The resulting accuracy of our finalized models are as follows:

Uber Backwards Elimination Model:

RMSE	MAE	MAPE
2.3	1.6	12

Uber PCR:

RMSE	MAE	MAPE
8.8	NA	55

Lyft Backwards Elimination Model:

RMSE	MAE	MAPE
2.2	1.6	14

Lyft PCR:

RMSE	MAE	MAPE
9.8	NA	73

B. Conclusion

From our analysis, we were able to predict the price of an Uber and a Lyft with a low RMSE, low MAE, and reasonable MAPE. To contextualize this model, we will apply our weights to the following situation.

Will Hunting just had a great time at the Red Sox game, but now needs to get home. He is looking to travel 3.7 miles from Fenway Park back to his apartment. He really enjoys the uberX and Lyft experiences. Unfortunately, the humidity is 90%, and the wind is 10.12 mph. The game was great, but now it is 6 pm, and he is stuck with the tail end of commuter traffic home. This causes there to be a surge multiplier of 1.0, which app should Will Hunting take to get home for the lowest price?.

Using the weights from our backward elimination models for each Uber and Lyft, we can predict the price of an Uber and a Lyft given Will Hunting's situation above. Our backward elimination Uber model takes in the source, distance, type of ride, humidity, and wind speed in determining price. Multiplying our coefficients by our appropriate values from our test case, we get a predicted price of $13.34 for Will Hunting's Uber ride home.

Similarly, our Lyft backward elimination model takes in source, distance, type of ride, surge multiplier, and hour of the day as variables to predict the price. Again passing in our test case values and multiplying them by our coefficients, we predict this Lyft ride to cost $13.77.

From this test case, it would seem that Will Hunting could save around $.45 on his ride home by using Uber's app. The main factor leading to this difference was the surge multiplier, though our distance was a key metric in both models. Our analysis of these pricing models is outlined in the following sections, as well as how drivers can leverage this data to offer the most rides.

Based on our analysis, we found that the pricing models of both Uber and Lyft are largely based on distance and the type of ride (number of passengers). This is demonstrated by the large coefficients for distance and type of ride within both our Uber and Lyft Backwards Elimination models and corresponding statistical significance. However, based on the data, Lyft differs from Uber with their surge multiplier for price which is implemented for people calling rides from high demand areas. Within our data set, the surge multiplier was solely applied to Lyft rides but further research unveiled that both companies have been using surge pricing since around 2014-2015.

Both companies also utilize specialized ride offerings for those seeking a premium experience and those seeking discounts. For example, Uber has the UberPool feature which cheapens the ride fare and matches riders with others in the area who are heading to a similar destination. Meanwhile, Lyft Shared offers a similar 'shared cab' service for their app. Uber offers premium rides in an 'Uber Black' and Lyft offers a similar experience in a 'Lyft Black'.

The most important factors in predicting the price of an Uber or a Lyft are distance and the type of ride (premium, size). These factors were statistically significant to a significance level of .001 and were included in each model produced by the assorted variable selection methods that we utilized. For Uber rides, each additional mile for a ride adds $2.41 to the ride total holding all other factors constant. And for Lyft rides, each additional mile adds $3.15 to the price holding all other factors constant. Based on this data, users planning longer distance rides should consider taking an Uber rather than a Lyft, assuming no other variables are changing.

Most weather variables were excluded when variable selection methods were run. Temperature, rain, clouds, and pressure were removed from the models during the stepwise and backwards elimination methods for both models. Humidity and wind were included within the Uber models but were removed along with other weather variables in the Lyft models. Although humidity and wind were included within the Uber backwards elimination model, these factors were not as statistically significant as distance and type of ride. For these reasons, weather should not be heavily considered when determining whether to take an Uber or Lyft.

Finally, our exploratory analysis visualizations highlighted that the most expensive rides were sourced from Boston University. The most expensive destination was Boston University as well, likely due to the frequency of Ubers from visiting college students, who may not have

access to their own car while on campus. This would directly influence the surge multiplier, as described above. However, the most common source and destination was the Financial District. This is likely rideshare users commuting to work in the early mornings, or commuting home after the typical workday hours. Overall, it seems that the best places for drivers to station themselves is at Boston University, or in the Financial District to offer the most rides throughout the day.

References:

[1] Salas, Erick Burgueño. “Uber's Users of Ride-Sharing Services Worldwide 2017-2020.” *Statista*, 20 Oct. 2021, https://www.statista.com/statistics/833743/us-users-ride-sharing-services/#:~:text=In%20the%20fourth%20quarter%20of,billion%20U.S.%20dollars%20in%202020.

[2] Conger, Kate. “Prepare to Pay More for Uber and Lyft Rides.” *The New York Times*, The New York Times, 11 June 2021, https://www.nytimes.com/article/uber-lyft-surge.html.

[3] Upadhyay, Roopam. “Step by Step Regression Modeling Using Principal Component Analysis - Case Study Example (Part 5).” *YOU CANalytics* |, 29 Apr. 2017, http://ucanalytics.com/blogs/step-step-regression-models-pricing-case-study-example-part-5/.

[4] Alice, Michy. “Performing Principal Components Regression (PCR) in R: R-Bloggers.” *R - Bloggers*, 21 July 2016, https://www.r-bloggers.com/2016/07/performing-principal-components-regression-pcr-in-r/.

Appendix: R Codes

````
---
title: "R Notebook"
output:
  html_document:
    df_print: paged
  html_notebook: default
  pdf_document: default
---

Uber and Lyft Price Prediction
Preston Chen, Megan Crawford, Mitch Feren, Alex Sampson
Prof. Huynh
Predictive Analytics Project Draft

Installing Packages
```{r}
#install.packages("tidyverse")
````
````

```
#install.packages("pls")
library (pls)
library(forecast)
library(zoo)
library(ggplot2)
library(dplyr)
library(tidyverse)
library(lubridate)



options(scipen = 999)
```

Loading Data, Data Exploration, Data Preparation
```{r}
# Loading in the data from a csv
cab_rides <- read.csv('Team05_Report_Data.csv')

```

```{r}
# Looking at first couple observations
head(cab_rides)
```

```{r}
# removes the rows with a duplicate ID
cab_rides <- distinct(cab_rides, cab_rides$id, .keep_all = TRUE )
cab_rides <- na.omit(cab_rides)
```

```{r}

# Visualizations


summary(cab_rides) #summary stats for the data


hist(cab_rides$price,xlim = c(0,110), ylim = c(0,250000), main = "Distribution of Prices",
xlab = "Price in Dollars ") #distribution of prices


#Looking at the distribution of source and destination locations
```

```
counts <- table(cab_rides$source)
barplot(counts, main="Source Distribution",las = 1, horiz = TRUE, cex.names=0.3, xlim
= c(0,100000))


counts2 <- table(cab_rides$destination)
barplot(counts2, main="Destination Distribution",las = 1, horiz = TRUE, cex.names=0.3,
xlim = c(0,100000))



#Distribution of the type of rides
counts3 <- table(cab_rides$name)
barplot(counts3, main="Type Distribution",las = 1, horiz = TRUE, cex.names=0.55, xlim
= c(0,120000)
       ,xlab = "Frequency")


#Price by each destination

agg_dest <- aggregate(cab_rides$price,
         by = list(cab_rides$destination),
         FUN = mean,
         na.rm = TRUE)


agg_dest$Avg_Price <- agg_dest$x
agg_dest$Destination <- agg_dest$Group.1


ggplot(aes(x=Destination,y=Avg_Price),data=agg_dest, main = "Price by Destination")+
  geom_point() +
  scale_x_discrete(guide = guide_axis(angle = 90))


#Price by each source

agg_source <- aggregate(cab_rides$price,
         by = list(cab_rides$source),
         FUN = mean,
         na.rm = TRUE)


agg_source$Avg_Price <- agg_source$x
agg_source$Source <- agg_source$Group.1
```

```
ggplot(aes(x=Source,y=Avg_Price),data=agg_source, main = "Price by Source")+
  geom_point() +
  scale_x_discrete(guide = guide_axis(angle = 90))

#Boxplot of Ride and Distance
par(mfrow=c(1,2))
boxplot(cab_rides$distance, main = "Distance")
boxplot(cab_rides$price, main = "Price")


#Boxplot of Temperature and Clouds
par(mfrow=c(1,2))
boxplot(cab_rides$temp, main = "Temperature")
boxplot(cab_rides$clouds, main = "Clouds")



#Boxplot of Pressure and Humidity
par(mfrow=c(1,2))
boxplot(cab_rides$pressure, main = "Pressure")
boxplot(cab_rides$humidity, main = "Humidity")


#Boxplot of Rain
boxplot(cab_rides$rain, main = "Rain")
```
```{r}
# Eliminating outlier rows in price column
price_outliers <- subset(cab_rides, cab_rides['price'] > 42.75)
cab_rides <- setdiff(cab_rides, price_outliers)
```

```{r}
# Setting seed and breaking data into Uber and Lyft sets
RNGkind(sample.kind = "Rounding")
set.seed(123)

uber_rides <- cab_rides %>% filter(cab_type == 1)
lyft_rides <- cab_rides %>% filter(cab_type == 0)
```

```{r}
```

```
# Separates Uber and Lyft data into training (70% of data set) and validation (30%) sets
uber.sample.rows <- sample(rownames(uber_rides), 50000)
uber_sample <- uber_rides[uber.sample.rows,]

lyft.sample.rows <- sample(rownames(lyft_rides),50000)
lyft_sample <- lyft_rides[lyft.sample.rows,]

uber.train.index <- sample(rownames(uber_sample), dim(uber_sample)[1]*0.7)
uber.train <- uber_sample[uber.train.index,]
uber.valid.index <- setdiff(rownames(uber_sample), uber.train.index)
uber.valid <- uber_sample[uber.valid.index, ]

lyft.train.index <- sample(rownames(lyft_sample), dim(lyft_sample)[1]*0.7)
lyft.train <- lyft_sample[lyft.train.index,]
lyft.valid.index <- setdiff(rownames(lyft_sample), lyft.train.index)
lyft.valid <- lyft_sample[lyft.valid.index, ]
```

Principal Component Analysis

```{r}
head(uber.train)
options(scipen = 999, digits = 2)

# Uber PCA
pca.uber <- prcomp(na.omit(uber.train[, c(3, 8, 14, 16:18, 20:21, 24:25)]), scale. = T)
summary(pca.uber)
pca.uber$rotation[,1:8]

#Lyft PCA
pca.lyft <- prcomp(na.omit(lyft.train[, c(3, 8, 14, 16:18, 20:21, 24:25)]), scale. = T)
summary(pca.lyft)
pca.lyft$rotation[,1:8]
```

Principal Component Regression

```{r}
library (forecast)
#Eliminate the non-numerical variables
uber.train.pcr <- uber.train[, c(3, 8, 14, 16:18, 20:21, 24:25)]
uber.valid.pcr <- uber.valid[, c(3, 8, 14, 16:18, 20:21, 24:25)]
lyft.train.pcr <- lyft.train[, c(3, 8, 14, 16:18, 20:21, 24:25)]
lyft.valid.pcr <- lyft.valid[, c(3, 8, 14, 16:18, 20:21, 24:25)]

#Uber PCR


library (pls)
```

```
pcr_model.uber<- pcr(price ~., data = uber.train.pcr, scale = TRUE, validation = "CV")
summary(pcr_model.uber)

pcr_predict.uber <- predict(pcr_model.uber, uber.train.pcr, ncomp = 4)

sqrt(mean((pcr_predict.uber - uber.valid.pcr$price)^2))

#Lyft PCR
```

Uber Model and Variable Selection

```{r}
# Builds out Uber model
uber.lm <- lm(price ~ source + distance + surge_multiplier + temp + clouds + name +
          pressure + rain + humidity + wind + day + hour, data = uber.train)
summary(uber.lm)
```

```{r}
# Uber model with backwards elimination
uber.lm.back <- step(uber.lm, direction = 'backward')
```

```{r}
# Model summary
summary(uber.lm.back)
# Uber: Backwards elimination model accuracy
uber.lm.back.pred <- predict(uber.lm.back, uber.valid)
accuracy(uber.lm.back.pred, uber.valid$price)
```

```{r}
# Uber model with forwards selection
uber.lm.forward <- step(uber.lm, direction = 'forward')
summary(uber.lm.forward)
```
```{r}
# Uber: Forward selection model accuracy
uber.lm.forward.pred <- predict(uber.lm.forward, uber.valid)
accuracy(uber.lm.forward.pred, uber.valid$price)
```

```{r}
# Uber model with stepwise selection
```

```
uber.lm.step <- step(uber.lm, direction = 'both')
summary(uber.lm.step)
```
```{r}
# Uber: stepwise selection model accuracy
uber.lm.step.pred <- predict(uber.lm.step, uber.valid)
accuracy(uber.lm.step.pred, uber.valid$price)
```

Lyft Model and Variable Selection
```{r}
# Builds out Lyft model
lyft.lm <- lm(price ~ source + distance + surge_multiplier + temp + clouds + name +
          pressure + rain + humidity + wind + day + hour, data = lyft.train)
summary(uber.lm)
```

```{r}
# Lyft model with backwards elimination
lyft.lm.back <- step(lyft.lm, direction = 'backward')
```

```{r}
# Model summary
summary(lyft.lm.back)
# Lyft: backwards elimination model accuracy
lyft.lm.back.pred <- predict(lyft.lm.back, lyft.valid)
accuracy(lyft.lm.back.pred, lyft.valid$price)
```

```{r}
# Lyft model with forward selection
lyft.lm.forward <- step(lyft.lm, direction = 'forward')
summary(lyft.lm.forward)
```

```{r}
# Lyft: forwards selection model accuracy
lyft.lm.forward.pred <- predict(lyft.lm.forward, lyft.valid)
accuracy(lyft.lm.forward.pred, lyft.valid$price)
```

```{r}
# Lyft model with stepwise selection
lyft.lm.step <- step(lyft.lm, direction = 'both')
summary(lyft.lm.step)
```

```{r}
# Lyft: Stepwise selection model accuracy
lyft.lm.step.pred <- predict(lyft.lm.step, lyft.valid)
accuracy(lyft.lm.step.pred, lyft.valid$price)
```

Principle Component Analysis and Principle Component Regression
```{r}

#pc11 or pc12 or none

```
```{r}
#Principal Component Regression


pcr_model<- pcr(price~., data = uber.train, scale = TRUE, validation = "CV")
summary(pcr_model)

pcr_predict <- predict(pcr_model, uber.train, ncomp = 10)

mean((pcr_predict-uber.valid)^2)


```