Taxi Fare Prediction

Introduction

Nowadays, we live in a data-drive world, everything can be quantified and analyzed. Hence, Analytics can be applied to any type of problem or business to make it more efficient and lucrative.

Taxi drivers can benefit from researching and getting insights from the data that they themselves produce and guide them to maximize their profits.

Objetive

Therefore, we are going to analyze a random sample of 49999 New York taxi trips made in 2015, however we are going to limit the are to Manhattan. Furthermore, we will build a random forest model and regression tree model that can predict the locations and times when the biggest fares can be earned.

```
library(tidyverse)
library(ggmap)
library(viridis)
library(tree)
library(lubridate)
library(randomForest)

df <- read.csv('Taxi_fares.csv')
glimpse(df)
```

```
## Rows: 49,999
## Columns: 7
## $ medallion         <chr> "4D24F4D8EF35878595044A52B098DFD2", "A49C37EB966E...
## $ pickup_datetime   <chr> "2013-01-13T10:23:00Z", "2013-01-13T04:52:00Z", "...
## $ pickup_longitude  <dbl> -73.94646, -73.99827, -73.95346, -73.98137, -73.9...
## $ pickup_latitude   <dbl> 40.77273, 40.74041, 40.77586, 40.72473, 40.76000,...
## $ trip_time_in_secs <int> 600, 840, 60, 720, 240, 540, 0, 120, 720, 180, 36...
## $ fare_amount       <dbl> 8.0, 18.0, 3.5, 11.5, 6.5, 8.5, 2.5, 4.0, 14.0, 4...
## $ tip_amount        <dbl> 2.50, 0.00, 0.70, 2.30, 0.00, 1.70, 0.00, 0.00, 2...
```

Data Cleaning and Base Map

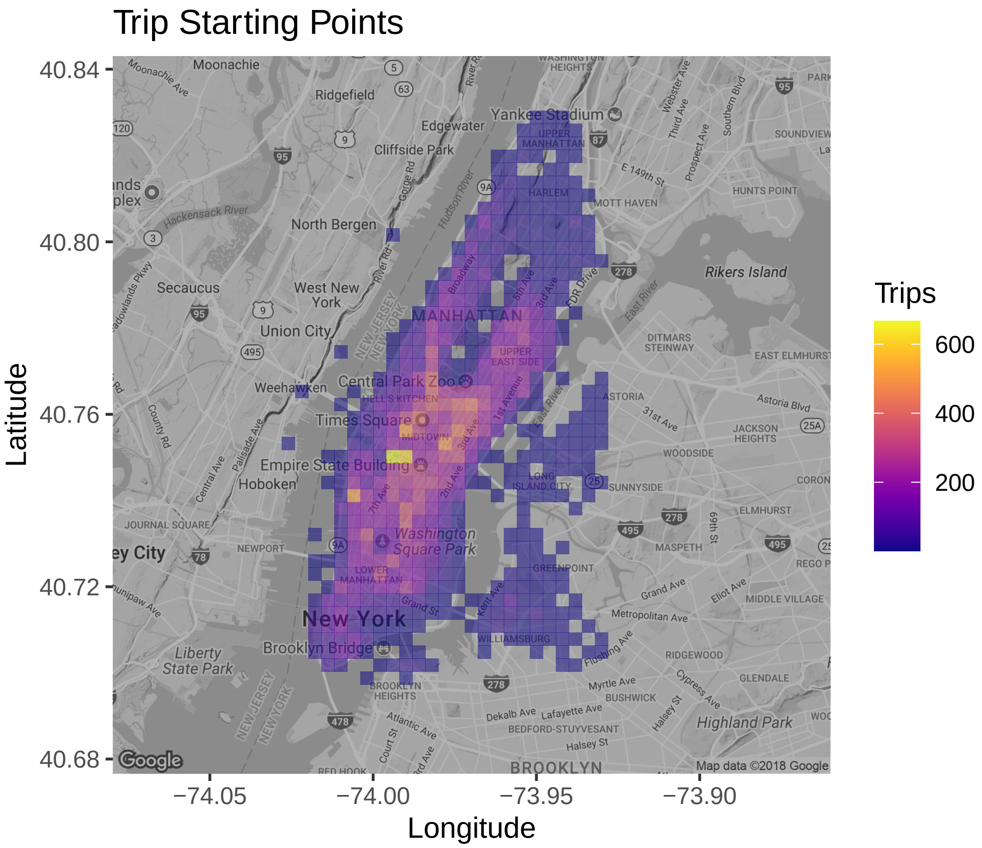
Preparing and cleaning our data its a vital step. This can make the difference between success or failure when we want to build machine learning models.

```
df <- df %>%
      rename(long = pickup_longitude, lat=pickup_latitude) %>%
      filter(fare_amount > 0 | tip_amount > 0) %>%
      mutate(total = log(fare_amount + tip_amount)) %>%
      mutate(total2 = fare_amount + tip_amount)

df <- df %>%
      filter(between(lat, 40.70, 40.83) &
           between(long, -74.025, -73.93))

manhattan <- readRDS("manhattan.rds")
```

Exploratory Data Analysis

Lets start by visualizing where to people tend to start a taxi trip.

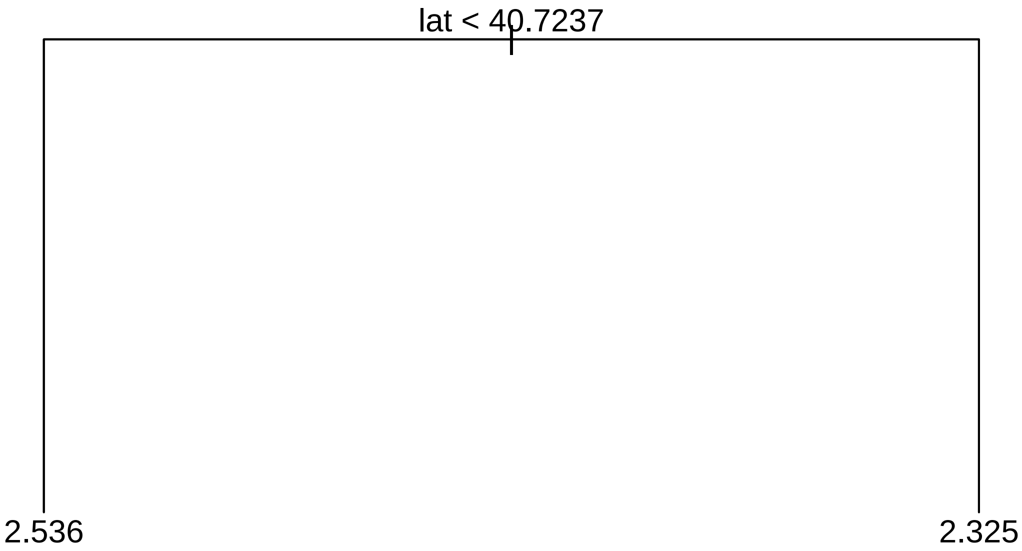
```
ggmap(manhattan, darken=0.3)+
    scale_fill_viridis(option='plasma') +
    geom_bin2d(data = df, aes(x = long, y = lat), bins = 55, alpha = 0.5) +
    labs(title='Trip Starting Points',x='Longitude', y='Latitude', fill = 'Trips')
```



The map shows that most trips start at around a particular area. According to Foursquare.com this specific area corresponds to a highly concentrated business and tourist one.

Lets predict the total fare with latitude and longitude by employing a regression tree. This algorithm will try to find cutpoints in those predictors that results in a decision tree with the best predictive capability.

```
tree <- tree(total ~ lat + long, data = df)
plot(tree); text(tree)
```

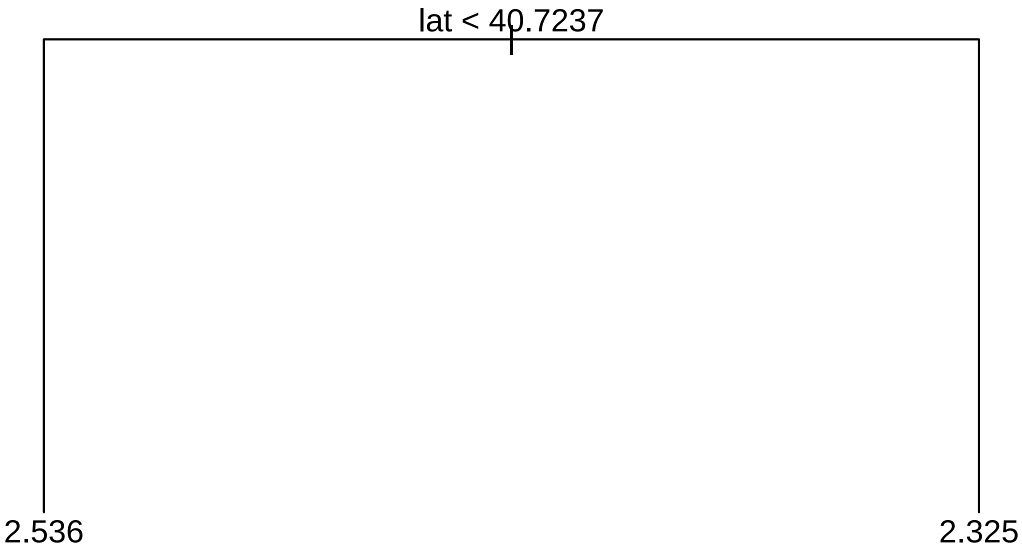


We have a start but it is actually a very simple model that posses only one split:

The model predicts that trips where the latitude is upper than 40.7237 are more expensive, which makes sense as it is downtown.

Somehow disappointing information that actually does not need any computational knowledge to get to that conclusion, just common sense. Let's add more predictors so see how far we can go.

```
df <- df %>%
      mutate(hour = hour(pickup_datetime),
             wday = wday(pickup_datetime, label=T),
             month = month(pickup_datetime, label=T))

tree <- tree(total ~ lat + long + hour + wday + month, data = df)
plot(tree); text(tree); summary(tree)
```



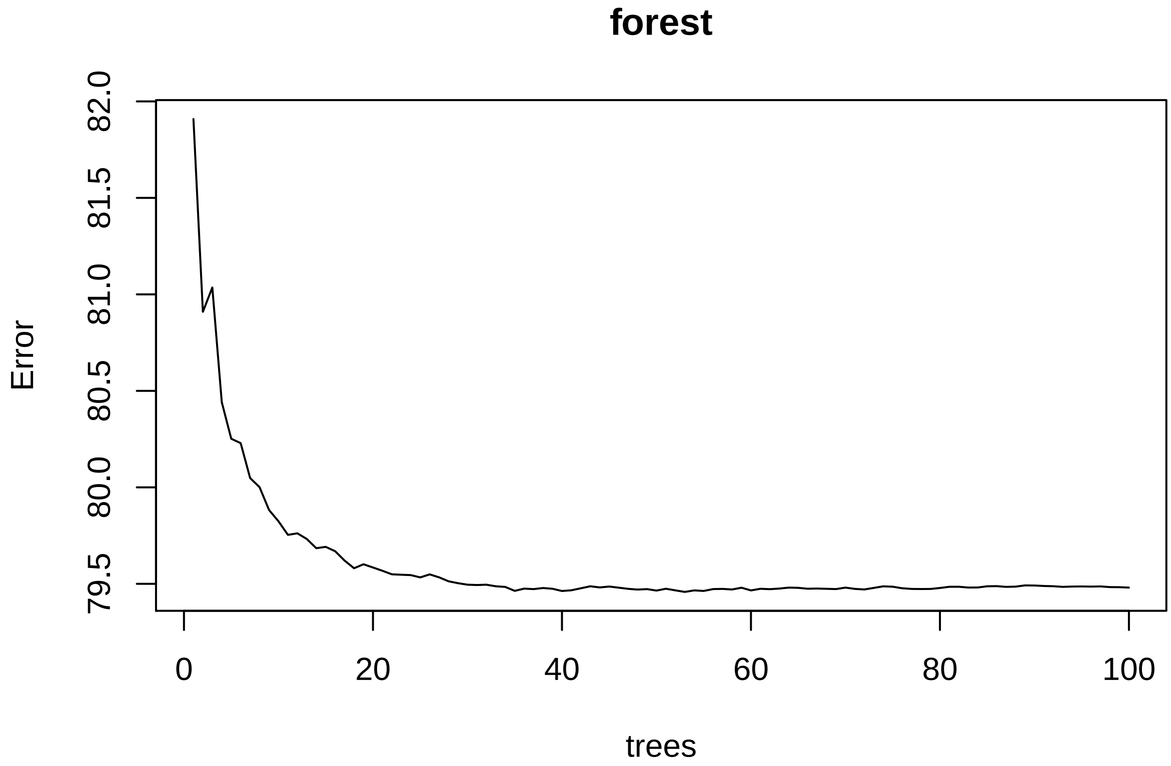
```
##
## Regression tree:
## tree(formula = total ~ lat + long + hour + wday + month, data = df)
## Variables actually used in tree construction:
## [1] "lat"
## Number of terminal nodes:  2
## Residual mean deviance:  0.3041 = 13910 / 45760
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.61900 -0.37880 -0.04244  0.00000  0.32660  2.69900
```

The model hasn't changed at all, even after including time variables. Likely due to latitude being the most promising variable to split and the other variables not being enough to be included.

Hence, let's change the strategy and use a random forest model. This algorithm creates different trees to fit to subsets of the data, an hopefully will include the other variables in some of the trees that make it up.

```
forest <- randomForest(total2 ~ lat + long + hour + wday + month,
                       data=df, ntree=100, sampsize=20000)
forest; plot(forest)
```

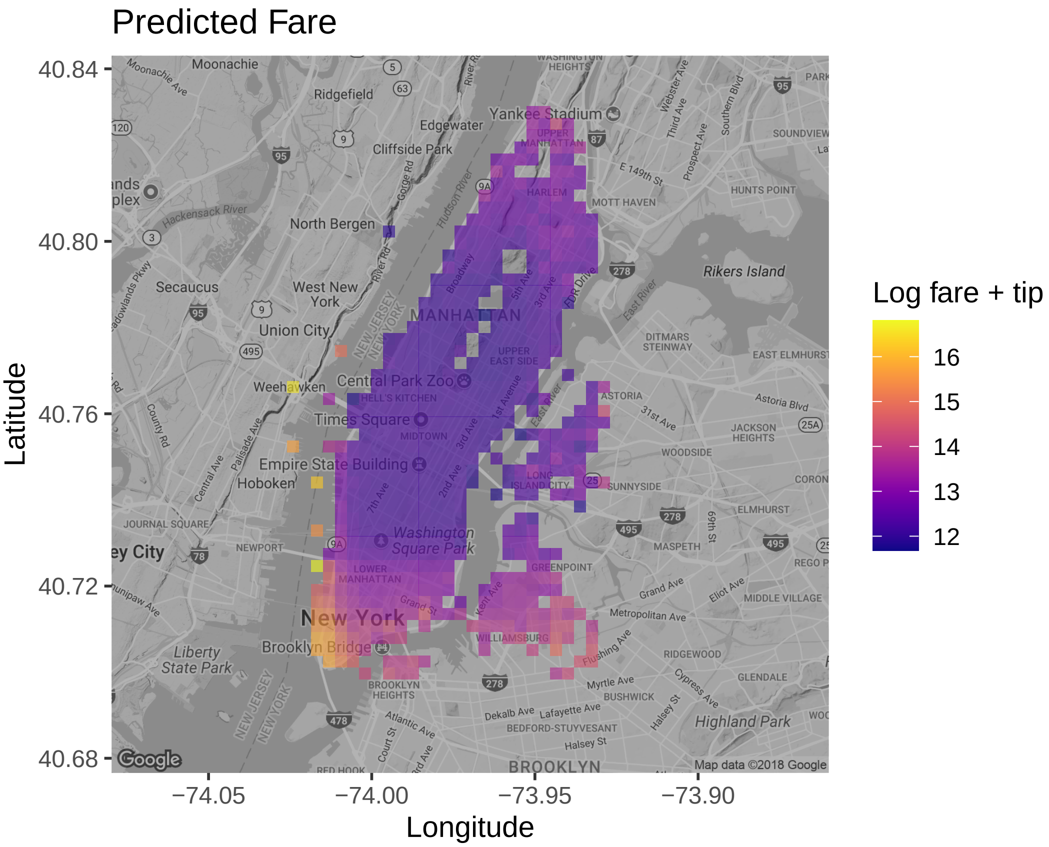
```
##
## Call:
##  randomForest(formula = total2 ~ lat + long + hour + wday + month,      data = df, ntree = 100, samps
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 79.48056
##                     % Var explained: 1.25
```



The plot above shows the mean of squared residuals, in other words, the average of the squared errors the model makes. Compared to the single tree model, this new one has a slightly lower error.

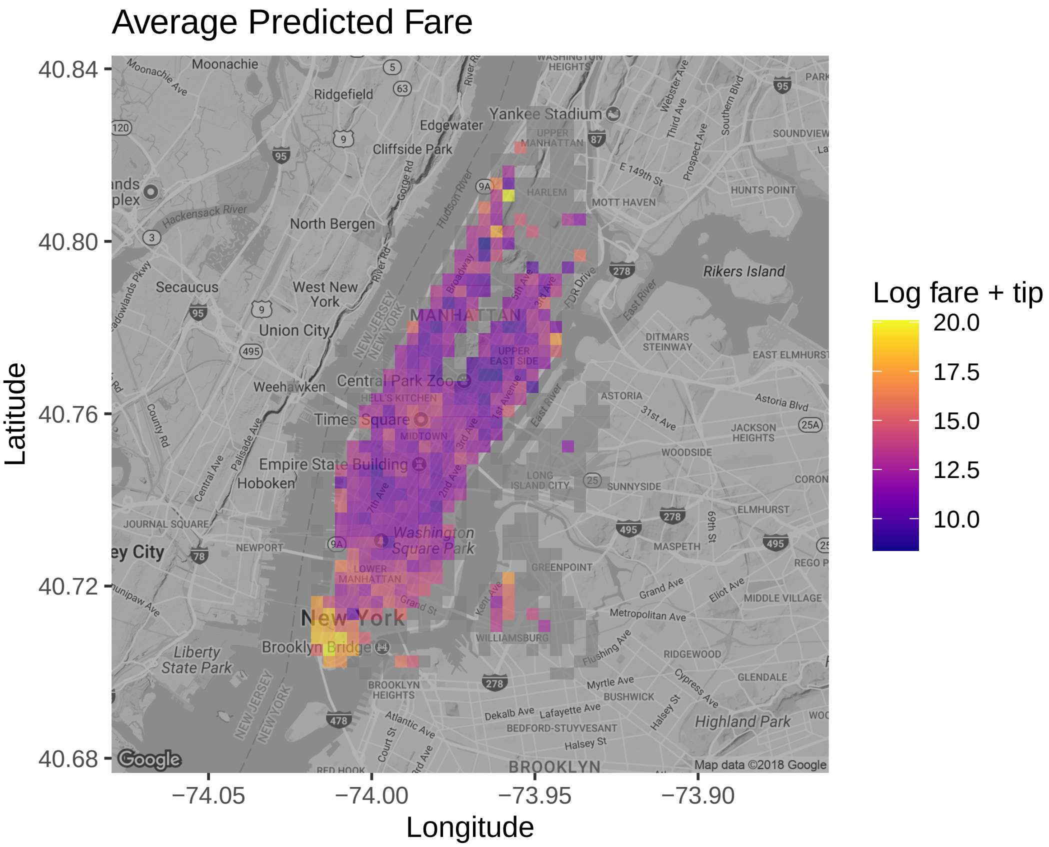
Thus, let's go ahead and use this model to look at the predictions projected into the map.

```
df$pred_total <- forest$predicted
ggmap(manhattan, darken=0.3) +
    scale_fill_viridis(option = 'plasma') +
    stat_summary_2d(data=df, aes(x = long, y = lat, z = pred_total),
                    fun = mean, alpha = 0.6, bins = 60) +
    labs(title='Predicted Fare', x='Longitude', y='Latitude', fill='Log fare + tip')
```



It looks like the predicted fares we see at the map are predicted to be high at downtown while lower at midtown. This can be somehow useful, but trips wont always have a high fare, therefore lets create a new map showing the predicted mean fares.

```
mean_fare <- function(x) {
    ifelse( length(x) >= 15, mean(x), NA)
}

ggmap(manhattan, darken=0.3) +
    stat_summary_2d(data=df, aes(x = long, y = lat, z = total2),
                    fun = mean_fare,
                    alpha = 0.6, bins = 60) +
  scale_fill_viridis(option = 'plasma') +
  labs(title='Average Predicted Fare', x='Longitude', y='Latitude', fill='Log fare + tip')
```



Conclusions and Recomendations

The random forest model is a good tool to capture patterns in our data. So far, for taxi drivers it is more profitable to work around downtown since thats where people hang out more in comparison to other areas at Manhattan.

Further work may include the plotting of predictors over time, or a combination of time and space. If you are curious enough, you should try it.