# Ultimate Data Science Challenge

Meskerem Goshime, Springboard
December 12, 2022

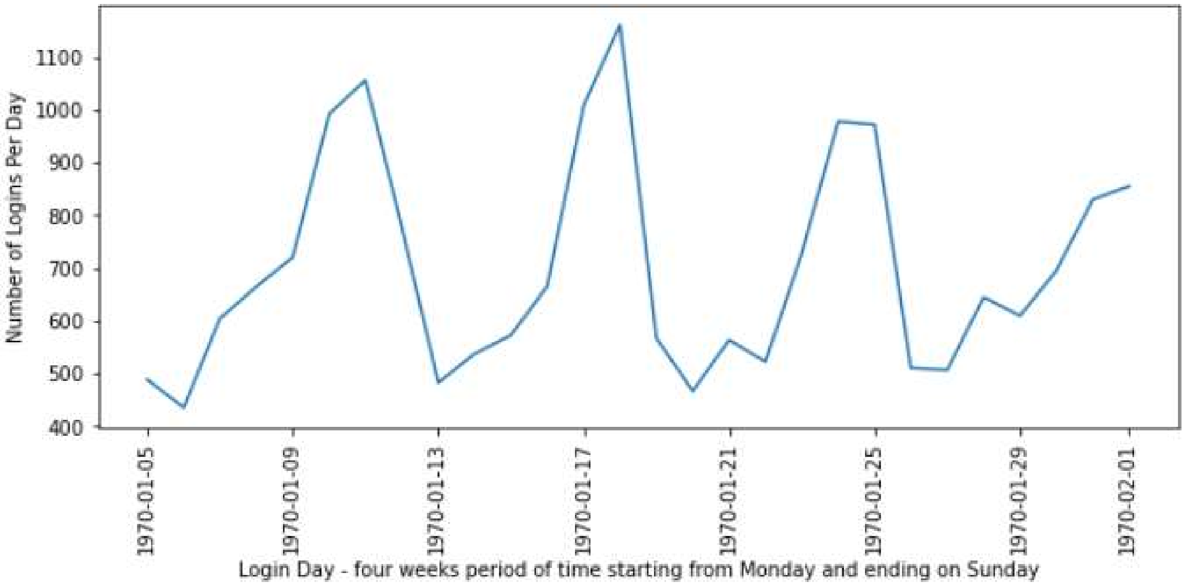
## Part I

**Complete notebook for part 1 on Github: Ultimate Data Science Challenge - Part 1 - Time Series.ipynb**

**Data analyzed: Drivers' login timestamps - logins.json**

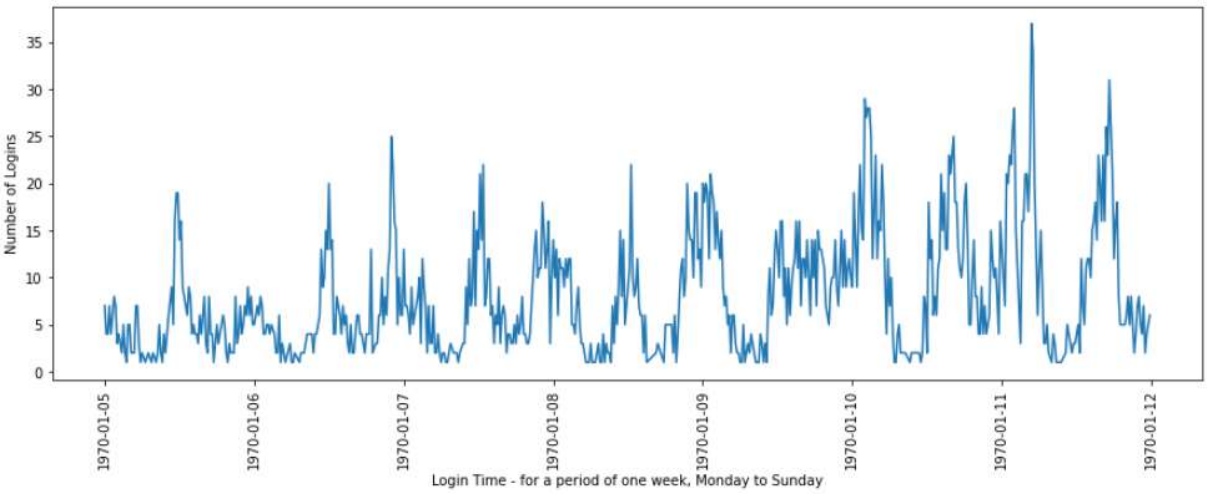
### 1. Weekly cycles

Plotting the data clearly showed weekly cycles. The lowest points in the number of logins per day happen on Tuesdays and the picks happen on Sundays.



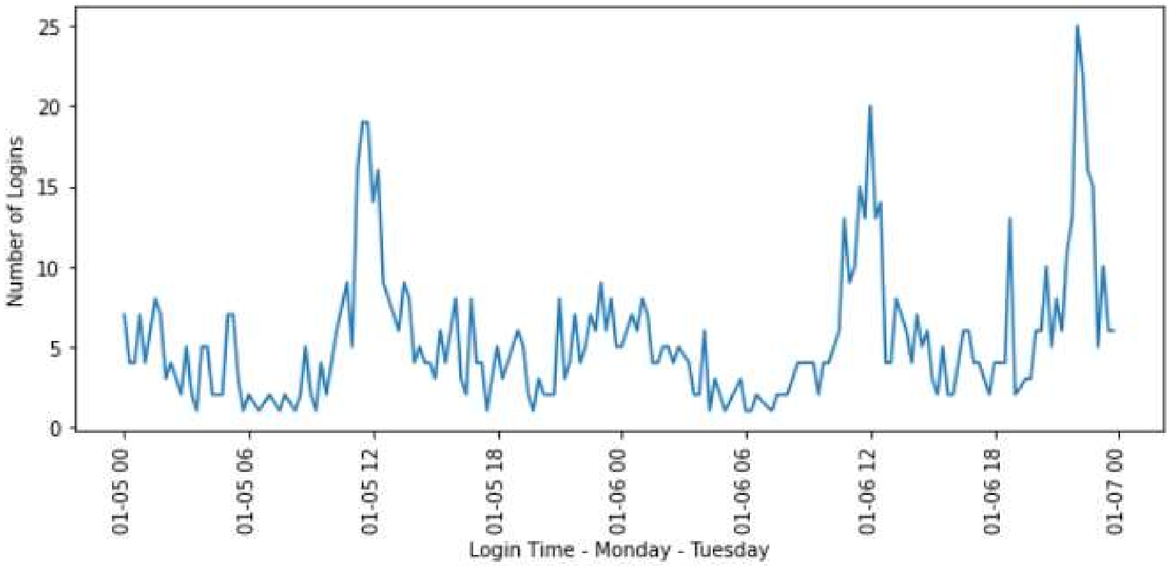
### 2. Daily Cycles

Plotting the hourly logins showed that there are daily cycles. However the pattern seems to change a little bit through the week. Therefore, I zoomed in to different parts of the week to analyze the pattern better..



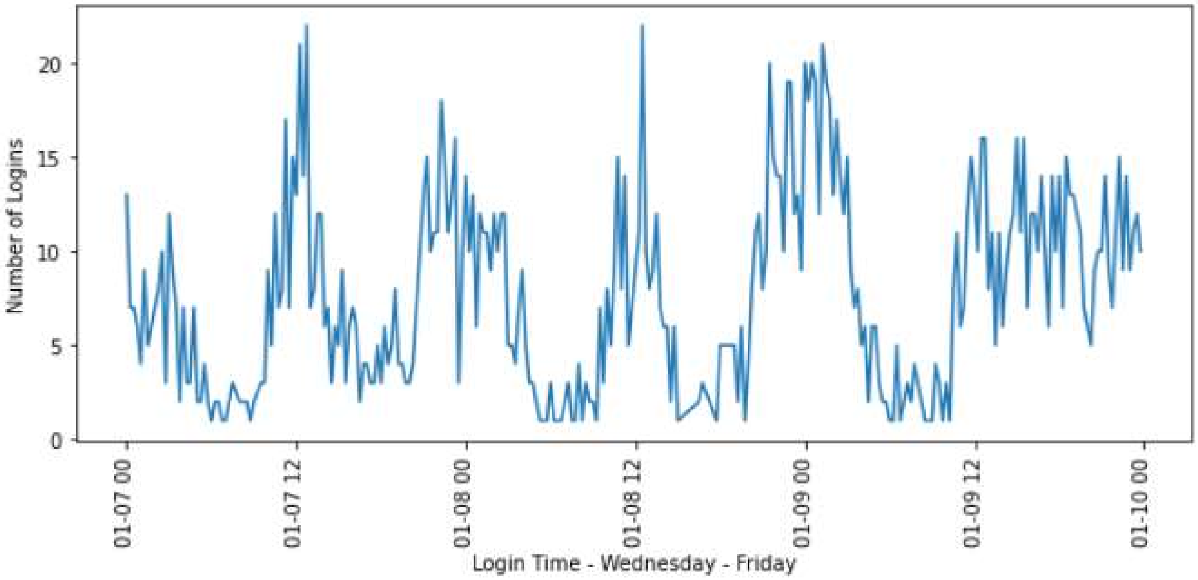
## Monday-Tuesday

The picks on Mondays and Tuesdays seem to be around the 12:00 noon hour.



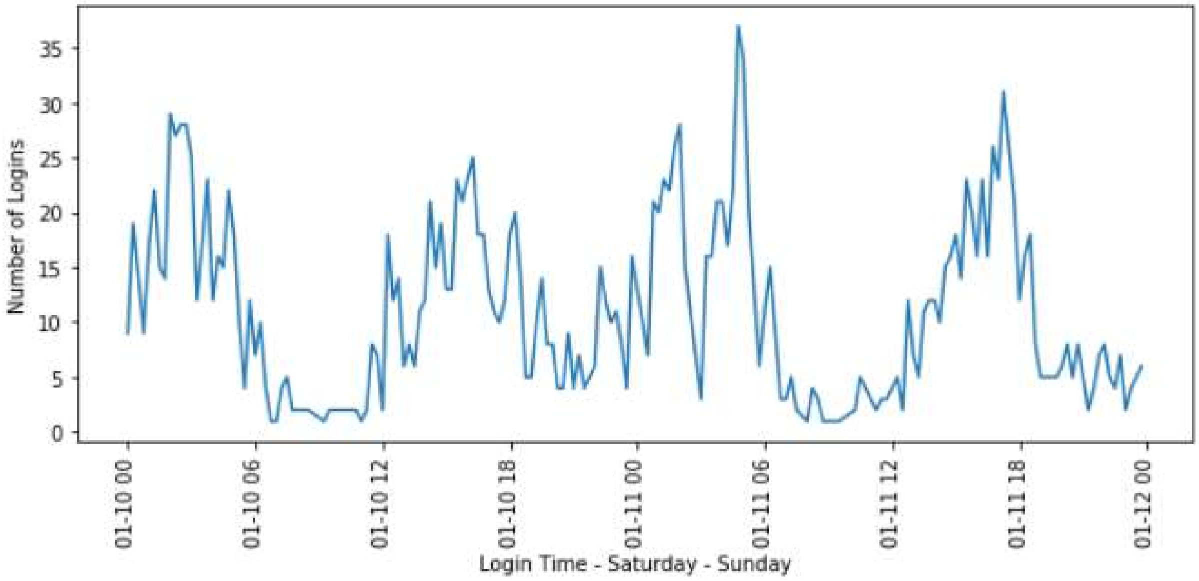
## Wednesday-Friday

There seems to be two picks between Wednesday and Friday. The picks seem to be near midnight and noon hours.



## Saturday-Sunday

The pick hours shift a bit on the weekends. The picks seem to happen between midnight and 6:00 a.m. and near 4:00 p.m.



# Part II

Proposed an experiment to encourage driver partners to be available in both cities, by reimbursing all toll costs.

1. First, I would collect and analyze login time series data for the two cities to confirm their pick times and slow times during different hours and different days of the week. If the data confirms the two cities have different pick times, this information will be a good motivation to the drivers since their goal is to earn more money in a shorter amount of time.
2. I would then collect drivers' login frequency for the two cities before and after the incentive to track changes.
3. I will set my null and alternate hypothesis
    - $H_0$ (Null Hypothesis) - observed change in login frequency is the result of chance
    - $H_1$ (Alternate Hypothesis)  - observed change in login frequency is not the result of chance.
4. I would use ARIMA (Auto Regressive Moving Average) time series model and use the p-value from the model to determine the significance of any change that may be observed. If the p-value is less than a predetermined value (probably 0.05), I will reject the null hypothesis, meaning the incentive has likely resulted in the observed change.

# Part III

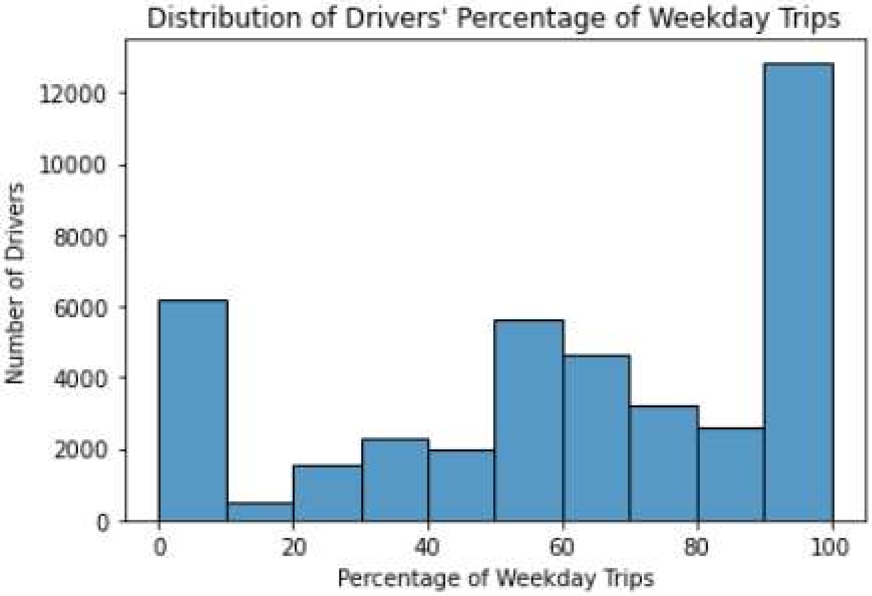
**Complete notebook for part 3 on Github: Ultimate Data Challenge - Part 3 - Modeling and Prediction**

**Data Analyzed: Drivers Data - ultimate_data_challenge.json**

## Exploratory Data Analysis

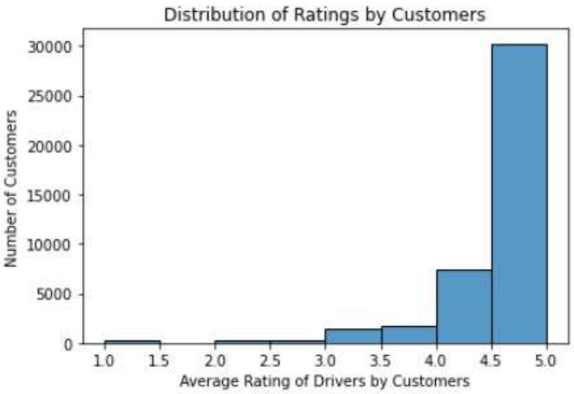
### Drivers Driving on Weekdays Versus Weekends

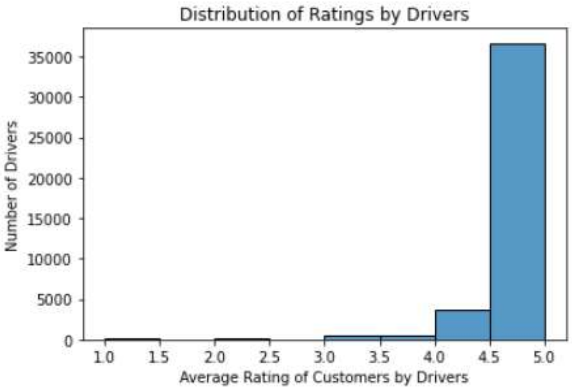
Some 12,000 drivers (about 29%) drive mostly during the weekdays. Some 6,000 drivers (about 15%) drive mostly on the weekends. The rest of the drivers divide their driving time between weekdays and weekends.



## Distribution of Ratings by Customers and by Drivers

Most drivers and customers seem to give a 5 rating. However, drivers seem to rate customers a little higher than customers rating drivers.





# Feature Engineering and Preparing the Data for Modeling

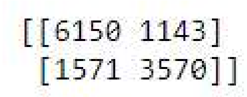
Here are the steps I took:

1. I added a retained column which takes the value of 0 if last trip date is more than 30 days ago and the value of 1 if the last trip date is less than or equal to 30 days.

2. One-hot-encoded the categorical columns, which are the city and phone columns.
3. Ultimate_black_user column had values of True or False. I replaced True values with 1 and False values with 0.
4. I standardized the numerical columns using Standard Scaler..
5. I assigned the column retained as my dependent variable (y) and the rest of the variables as my independent variables (X)
6. I divided the data into training and test sets.

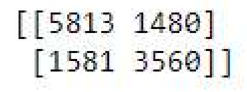
## Modeling

I built a Gradient Boosting Classifier and computed the ROC/AUC score which came out to be 0.86. The confusion matrix was the below.



```
[[6150 1143]
 [1571 3570]]
```

I then built a Random Forest Classifier for which the ROC/AUC score came out to be 0.82. The confusion matrix looked like this.



```
[[5813 1480]
 [1581 3560]]
```

The performance of the Gradient Boosting model was clearly better than the Random Forest. Therefore, I decided to use the Gradient Boosting Model to predict driver retention. Once retention is predicted, the company can focus its efforts to retain those drivers who are at risk of leaving.