Part 1 - Exploratory Data Analysis

See the accompanying jupyter notebook (https://bit.ly/3bPvNN1)

The graph below indicates that there are daily cyclic patterns for login times.

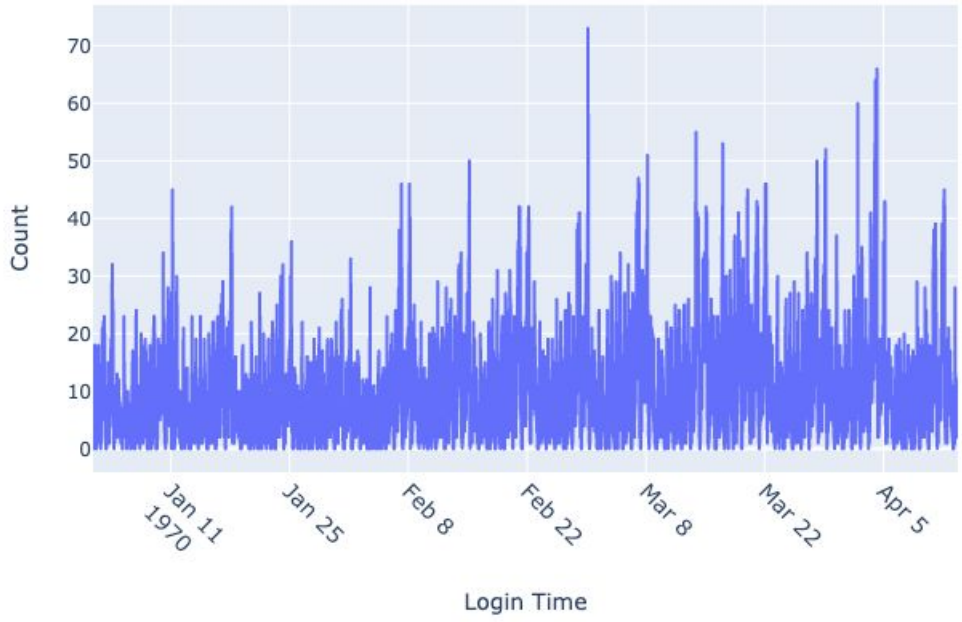


Figure 1: User logins aggregated by 15 minute intervals

It looks like the login counts may be trending upwards, but the data is a bit noisy. Smoothing by calculating the trailing average over the past 28 days to uncover any macro level trends and it does indeed look like user logins are trending upwards.

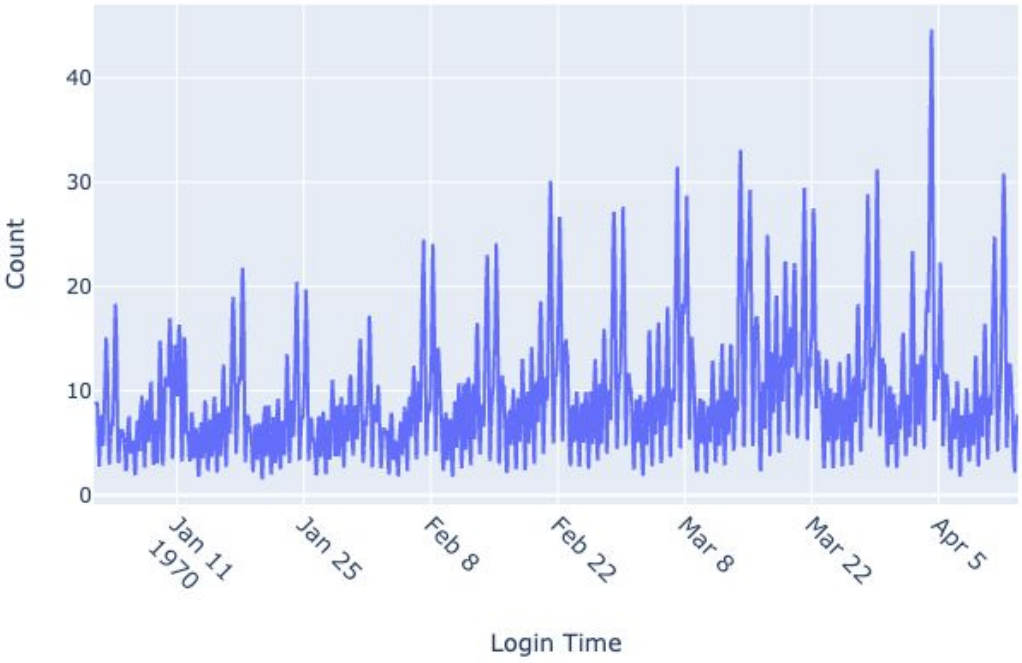


Figure 2: User logins 28 day trailing

Taking a closer look at several days randomly, user login activity spikes midday and then again in late evening/early morning hours.

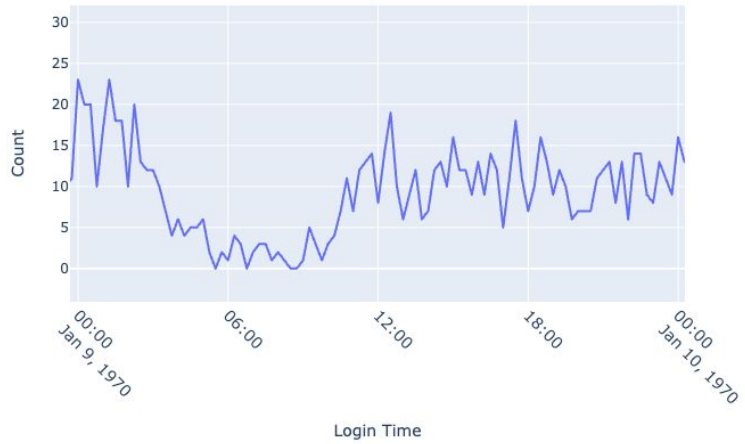


Figure 3: User login activity for Jan 9, 1970

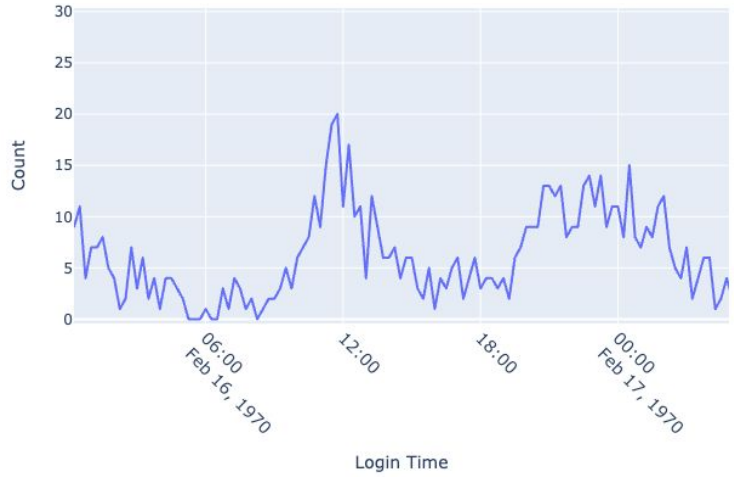


Figure 4: User login activity for Feb 16, 1970

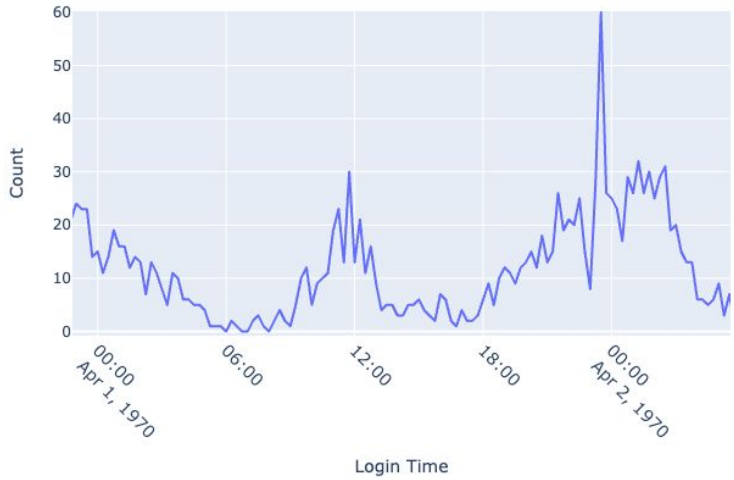


Figure 5: User login activity for April 1, 1970

Grouping the data by hour and aggregating based on average logins supports this observation. Average logins spike around 11:00pm - 1:00am and then again around 11:00 am. Mid-morning hours see the lowest average logins.

Figure 6: Mean login by hour

Taking a look at counts by day of the week, it is clear that more logins occur on Friday, Saturdays and Sundays.

Figure 7: Total Logins by Day of Week

Taking a closer look at logins by day of week, Saturdays and Sundays seem to have the majority of login activity in the late evening / early morning.

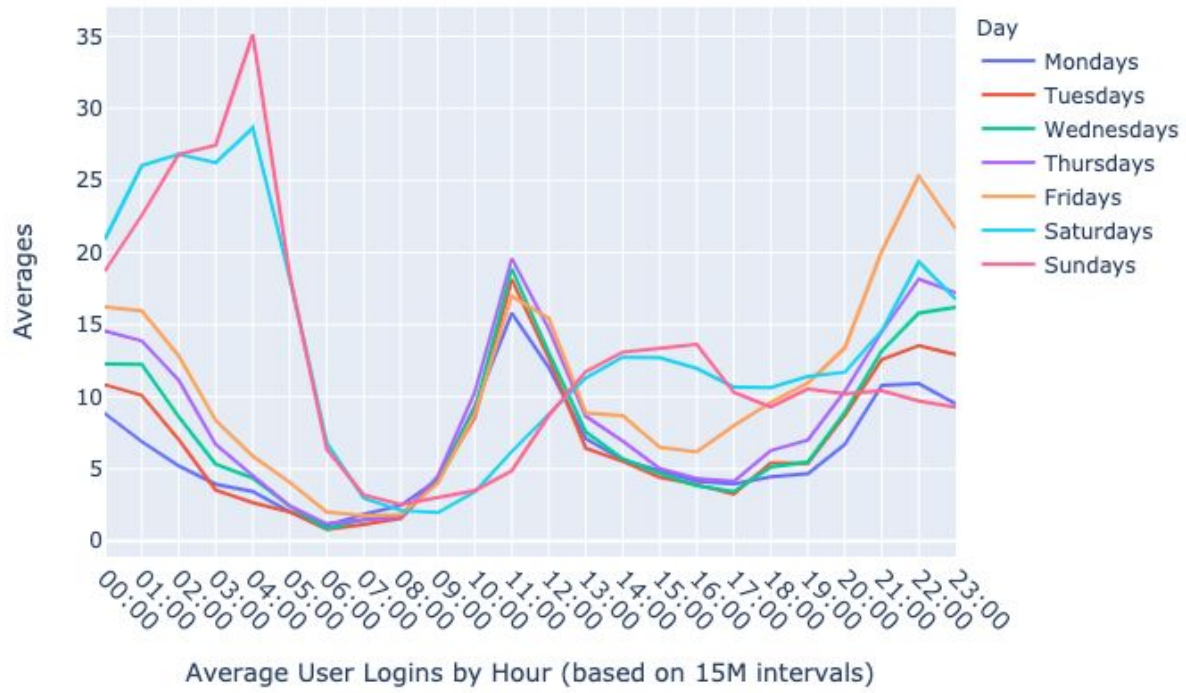


Figure 8: Average logins by hour for each day of the week

Additional considerations:
Using Pandas Profiling, it was noted that there are 877(0.9%) duplicates in the dataset. It is hard to tell if these are true duplicates or two users logging in at the same time. More information is needed to make that determination. Also, there seems to be a large spike in logins on March 1, 1970 at 04:30. It would be beneficial to get an understanding of any events, campaigns, or releases that occurred during this time.

Part 2 - Experiment and metrics design

1. Considering the purpose of the experiment is to increase driver availability in each city, I am assuming there may be some issues with the amount of time that riders are waiting for a ride. Therefore, rider wait time would be a useful metric, assuming the KPI is consistent across demographic groups and locations. Additional KPIs that would be useful to measure are the number of toll reimbursements issued, evening Gotham-bound tolls, daytime Metropolis-bound tolls, daily miles driven, and average trips completed per day.
2. I would use an A/B test to test the hypothesis that providing toll reimbursements would encourage driver partners availability in both cities. Randomly assign driver partners in each city to 2 groups. The first group would be the control group and the second group would be to test the hypothesis.
    a. Group 1 would NOT be offered reimbursements for tolls. Group 2 would be offered reimbursements for tolls.

b. A hypothesis test where H_0 is toll reimbursements are impervious to average rider wait time and H_1 is toll reimbursements do have an effect on average rider wait time. Using permutation tests, we can calculate the p-value for the test statistic. A p-value of < .05 would be considered statistically significant and we would reject the null hypothesis.
c. I would track the metrics over time using graphs and smoothing to reveal macro level trends and evaluate them as we make changes. I would also split the data by city, demographics(i.e. gender, age, etc.), car type, device to identify any micro level trends. It would be beneficial to run the tests for at least 4 weeks to gather enough data. I would make recommendations based on the macro and micro level changes and results of the test. If the results are statistically significant we can reject the null hypothesis and make a recommendation that the company should move forward with rolling out the toll reimbursements. If we do not find any statistical significance we may recommend to run another test for longer to obtain more data or recommend the city managers find another method to increase driver availability.

Part 3 - Predictive Modeling

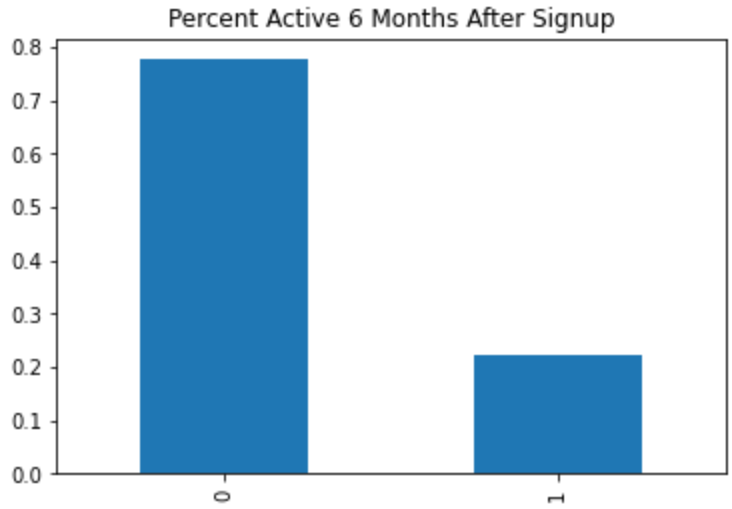
See accompanying jupyter notebook (https://bit.ly/3cXqPzm)

Data Wrangling

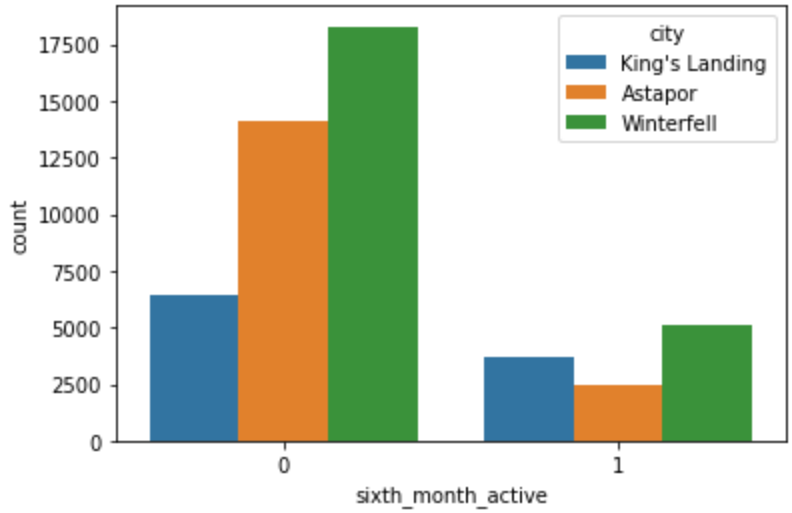
1. Converted signup and last trip columns to datetime datatype
2. Replaced null values. For phone column replaced with 'Other' for avg_rating columns, replaced with mean value for column
3. Created a new column with boolean values for if the user was active after 6 months.

EDA

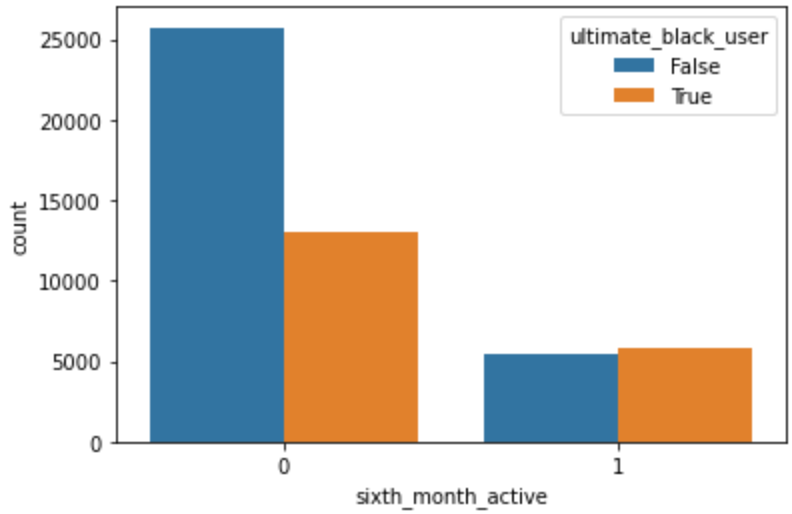
~22% of users are active 6 months after signup



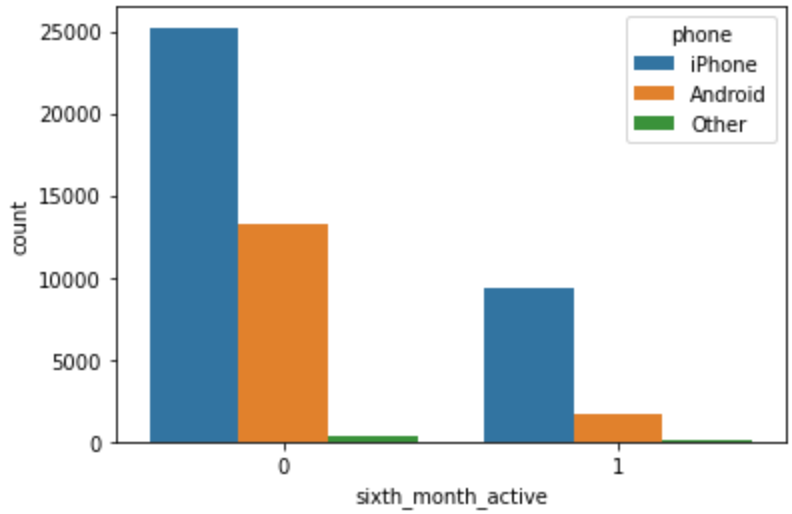
King’s Landing had a higher percentage of users active 6 months post-signup



Ultimate Black Users had a highers percentage of users active 6 months post-signup



iPhone Users had a higher percentage of users active 6 months post-signup



Statistical Analysis

- Chi-square tests were used to evaluate the statistical significance of discrete variables in the data set. All of the categorical variables: phone, ultimate_black_users, and city are statistically significant with a p-value less than 0.05.
- ANOVA tests were used to evaluate the continuous variables and a p-value of less than 0.05 was again used as the cutoff for statistical significance. 4 of the 7 numerical variables were found to be significant: trips_in_firs_30_days, surge_pct, avg_ist, and avg_rating_by_driver.

Machine Learning

1. I created a dataset that included all of the statistically significant variables and included the target variable sixth_month_active
2. I create preprocessors to normalize the continuous variables and encode the categorical variables.
3. I then created a pipeline using K-Nearest Neighbors.
4. Using GridSearchCV I did a bit of hyperparameter tuning and cross validation
5. I split the data into training and test sets
6. I finally fit the model on the training data and tested it on the test set
7. I then calculated accuracy, precision and recall of the model

The model has 79% accuray and performs much better on determining which users will not be active after 6 months of signup, which is ok because those are the users we more than likely would like to target to mitigate the risk of them being inactive. Moving forward it would be if we could tune the model to perform better on category 1. Possibly by obtaining more data, using another model, or the ensemble method.