# Ultimate Technologies Data Science Challenge

## Part 1

### Summary

Demand for Ultimate Technologies has a weekly cycle, a daily cycle, and a 12-hour cycle.

- Weekly – Demand starts low at the beginning, then builds throughout the weekdays and is largest on weekends; from under approximately 100 logins on Monday to over 200 on Saturday!
- Daily – There are three peaks throughout a day; one in the morning, one in the afternoon, and one at night
- 12-Hour – This is the approximate time between peaks; days begin mid-peak, then there is another peak during the day, and days end on a peak as well
- Mornings and Evenings are busier on Weekdays, while Afternoons and Nights are busier on the weekend

When fit with an appropriate model that takes these cycles into account, there is no clear trend in the demand; the demand is neither increasing nor decreasing over time.

### Methodology

Data was loaded from the .json file, and then logins were aggregated into 15-minute intervals to measure demand in those intervals. After initial plots looking at 15-minute demand over different time horizons, the data was then aggregated and explored at biweekly, weekly, daily, and 12-hour intervals.

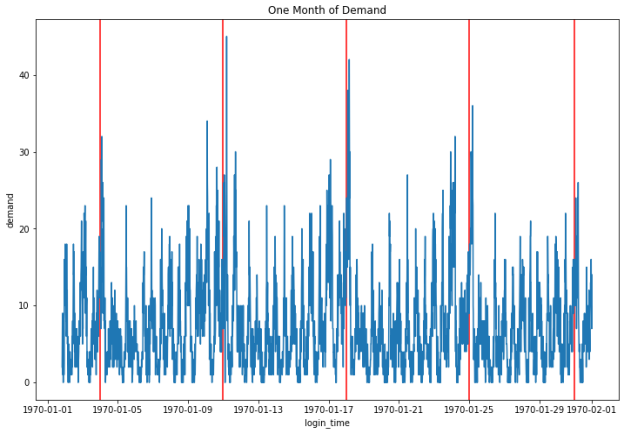
Each aggregation was plotted along with a simple linear regression line. All of the linear models show a slight positive trend. The residuals for the linear models are problematic; the lines are not rigorous. After examining ACF and PACF plots, which show the seasonality quite well, SARIMA models were fit to the daily and 12-hour aggregations as well. These models capture the behavior of the demand. They show no evidence of linear trend.

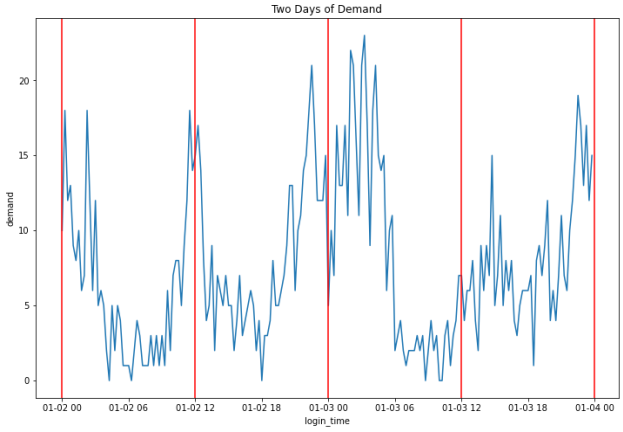
The daily data was then explored for more trends. Separated into weeks and plotted together, the lines show a regular growth throughout the week to a high on Saturday night.

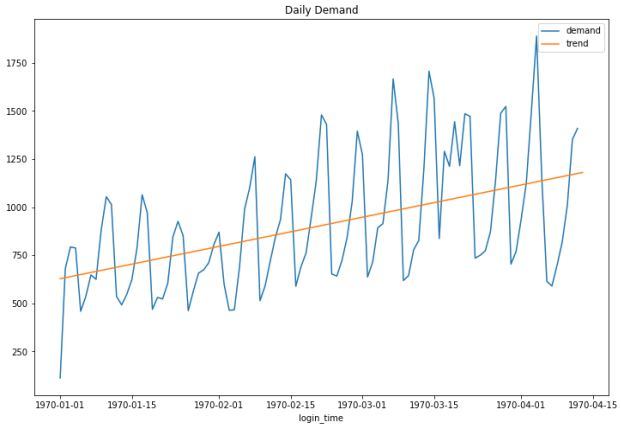
Data was then split into morning, afternoon, evening, and night by 6-hour intervals, and a stacked bar plot showed that while mornings and evening are busier on weekdays, afternoons and nights are busier on weekends.

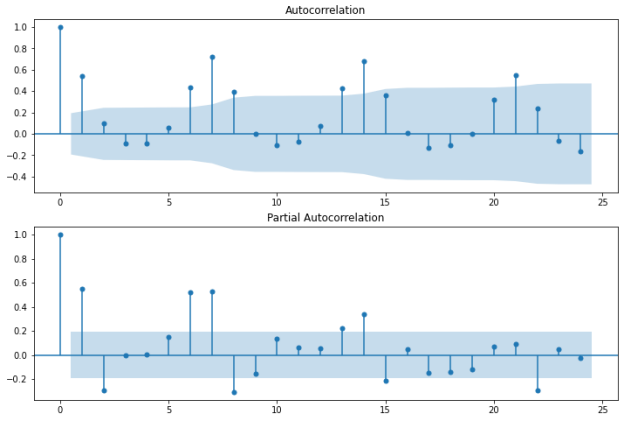
For more detailed methodology, see the related Jupyter Notebook. Plots are on the following page.

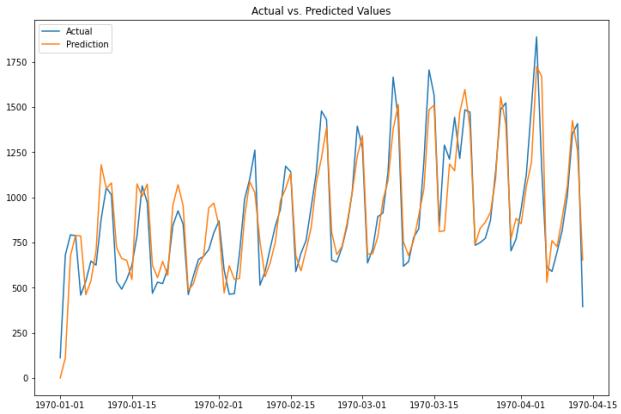
# Part 1: Figures

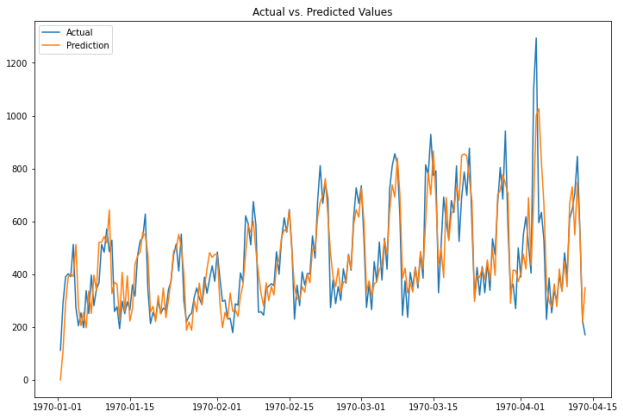


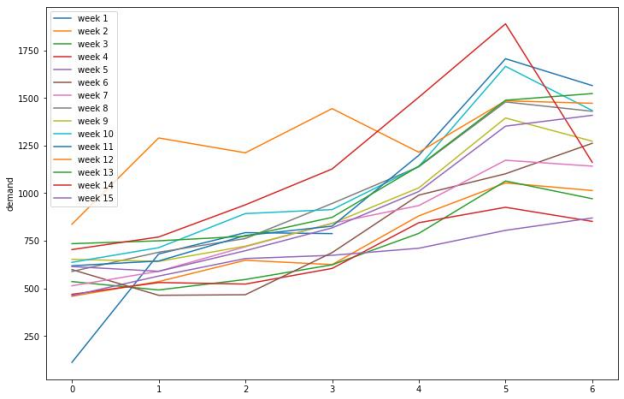



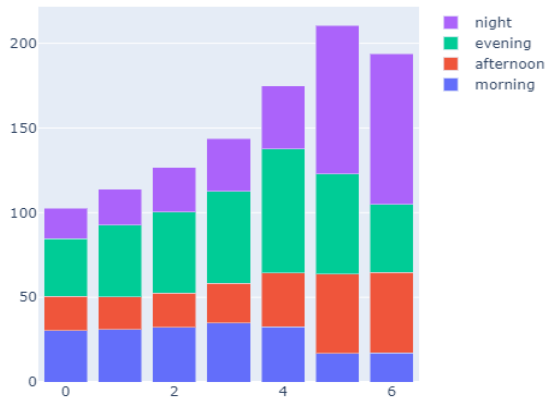



## Part 2

The issue in this part is that there are two neighboring cities separated by a toll-bridge, and we want to convince driver partners in those cities to work BOTH cities, rather than sticking to their side.

We've got too cities, Metropolis and Gotham, that are separated by a toll bridge. On weekends, both cities are busy pretty much all day. On weekdays, Metropolis is busy during the day, but Gotham is busy at night. Driver partners tend to either work in Gotham or Metropolis, but not both. We want to encourage them to cross over more often by reimbursing the tolls to cross the bridge.

### How can we measure the effectiveness of the program?

Before we ask this question, we really should be more specific about what success looks like. Do we want more drivers to work both cities in the same day? Do we just care that they are working both cities intermittently? Do we want more drivers to cross the bridge when the other city is busy? Do we want more drivers willing to take passengers from one city to the other?

While all of these share the consideration that we want to be able to measure change in our metric, no two of them can be measured by the same metric, so we will consider them each separately.

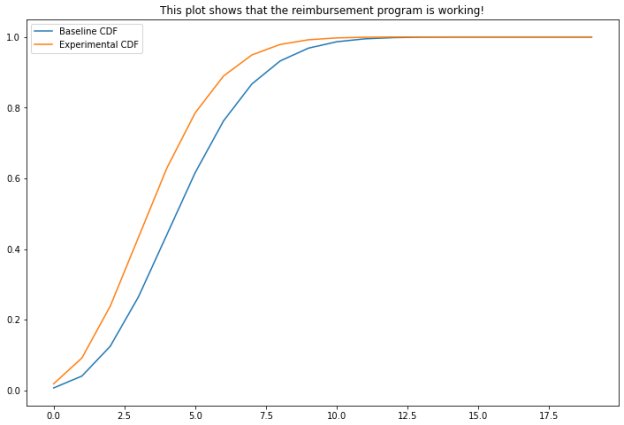
My best guess is that we just want more people to cross over sometimes, even if they don't work both cities in the same day, so I will save that version of success for last. Now let's talk about the others.

**We want more drivers to take passengers from one city to the other** - In this case, we can guess that drivers rarely accept assignments that they know cross the bridge. Thus, I propose that we measure the number of drivers that inter-city assignments are sent to before being accepted. With a large number of these measurements, we can construct the distribution (it should be approximately a Poisson distribution, but perhaps we can just use the empirical distribution) or the number of drivers such a trip is sent to before being accepted. This will give us a Cumulative Distribution Function (CDF), which is our real metric. For reference, this is how a CDF works:

The probability of an inter-city trip being accepted by the first $N$ drivers to whom it is assigned is $P(n \leq N)$, the value of the CDF at $N$.
Now that we have a baseline CDF, we can make sure to let drivers know that tolls will be reimbursed, then allow them to operate as usual while we collect the same data again with the reimbursement program in place. I will explain why we can simply do this in the next paragraph. Once we have sufficient data (sample size will need to be determined according to the necessary precision), we will calculate the new empirical CDF, and test for stochastic dominance.

A quick explanation of the outcome of the stochastic dominance test is: If the baseline CDF is lower than the experimental CDF at every number $N$, then the baseline CDF *stochastically dominates* the experimental CDF. Check the plot below for a visual that would indicate that the program is effective. Because stochastic dominance simply requires the baseline CDF to be greater than or equal to the experimental CDF at every point, all we need to do to test for stochastic dominance is plot the experimental CDFs like in the visual below. Thus, this is a

nonparametric comparison with no underlying assumptions (except that the empirical CDF is representative of the true CDF, and so we can just collect data freely and not worry about whether it is i.i.d. or normally distributed or anything like that.
There are three orders of stochastic dominance, but if the baseline CDF stochastically dominates the experimental CDF at any order, then we can conclude that the reimbursement program has raised the probability of a driver accepting an assignment that crosses the bridge.




**We want drivers to work Metropolis in the daytime and Gotham at night, in the same day** - In this case, we need to track which drivers complete one assignment in Metropolis, and then later complete at least one assignment in Gotham. It would be good to track both the total number of driver partners that work in both cities in the same day and the proportion of driver partners that do so; we we need to know how many drivers, in raw numbers, are crossing over the bridge so we can plan for the expense of reimbursement; we also want to track the proportion, because the proportion of drivers who cross will be more indicative of the success of the program. For example, say that 10% of driver partners work both cities before the program; over the first month with reimbursements in place, the total number of drivers grows by 20%, but the number of drivers that work both cities grows by 10%...in this scenario, the proportion of drivers that work both cities actually declines to about 9%.

We should have access to the entire population of assignments, so we could just compute the population proportion before and after the program starts and compare them. However, this might be very computationally expensive, so instead we should just use simple random sampling to grab a representative sample of drivers (necessary sample size determined by necessary power and precision), then use a two-sample hypothesis test for difference in proportions. The major benefit of this method is that it is easy to understand; if the proportions are significantly different, then the reimbursement program is having and effect. But the method is not foolproof; we will be unable to separate the effect of the program from the effects of time or from the timing of the reimbursement program rollout. We can try to avoid these issues by comparing time at least a few months before the program with time at least a few months after the program starts, but that should only reduce the interfering effects of time and the program beginning itself.

For a more informative (but more technical) test, we could use intervention analysis. Here's how it works: We track the population proportion for several months (a large sample of weekdays) both before and after the program begins, then fit an appropriate ARIMA model

to the data from before the rollout. We use that ARIMA model to predict the proportion for the time series after the program is in effect, and find the difference between the observed values and the predictions. Then, we can use those residuals to measure the effect of reimbursement on inter-city partners. There are several documented patterns that might be fit to the residuals, but it seems most likely that we should be estimating a gradually increasing effect that move the proportion to a new long-term mean. The new pattern model will look like this:

$$z_t = \frac{\delta_0(1 - w_1^{t-T+1})}{1 - w_1} I_t$$

Wherein $I_t$ is 0 before the program begins and 1 afterward. We use the residuals to estimate $\delta_0$ and $w_1$.

We could also just fit an ARIMA model with $I_t$ as an exogenous variable to estimate a constant effect of the reimbursement program. The best choice of test to estimate the effect of them program is all about the tradeoff between technical sophistication and ease of explanation. The $z_t$ ARIMA method is the hardest to explain to a non-technical audience, but with appropriate visualizations and a good choice of words, any stakeholder ought to be able to understand it.

**We want drivers to work in both cities, but don't care if it is on the same day** - This one is pretty similar to the previous version of success, but requires some small changes. Instead of using the proportion of drivers that work in both cities on any given day, we want to look at the proportion of drivers that complete enough assignments to justify reimbursing a bridge toll in each city. For the sake of this hypothetical, let's assume it is 3 assignments. For example, if a driver partner completed three assignments in Metropolis on Monday and then worked in Gotham over the rest of the week, that driver would count. Then, we would track the proportion of drivers that work in both cities over the course of several months before the reimbursement program and then a few months after the program. Then, we have the option of using the ARIMA method from the previous section or using a simpler method to estimate the effect of the reimbursement program.

To summarize, the best way to measure the effectiveness of the program depends on the actual goal of the project, which is not well defined in the question. However, three possible success scenarios and an appropriate metric for each has been outlined in the answer above. Since the company has access to the entire population of drivers and assignments, we could avoid the need for approximations by simply using all of the data, but that might be expensive and so we could also easily obtain an appropriately sized random sample of drivers/assignments as needed. We also have several options for tests that tell us the size of the effect the program is having; the best test will have the appropriate combination of sophistication and explainability.

## Part 3

### Summary

The features that predict retention, in order of importance:

1. avg_dist – When people use the service for even short tides, they are more likely to be retained
2. weekday_pct – The random forest says this is an important feature, but the logistic regression coefficient was near zero; it is hard to quantify the effect of this feature on retention
3. avg_rating_by_driver – Interestingly, clients with higher ratings from the drivers were less likely to be retained
4. surge_pct – The random forest says this is important, but logistic regression disagrees. However, intuitively, a person who is often charged with a surge multiplier is less likely to keep using the service
5. trips_in_first_30_days – People who take a lot of trips in the first 30 days are more likely to be retained
6. avg_rating_of_driver – People who rate their driver lower are more likely to be retained, which is also interesting. More research is needed to decipher this one and avg_rating_by_driver
7. phone – iPhone users are more likely to be retained. It is possible that differences in user experience in the app account for this one
8. ultimate_black_user – Ultimate Black Users are more likely to be retained. After all, they pay extra for a premium experience
9. city – Ultimate Technologies is doing better in King's Landing than in other cities, especially Astapor. We should look into marketing efforts and performance in the three cities

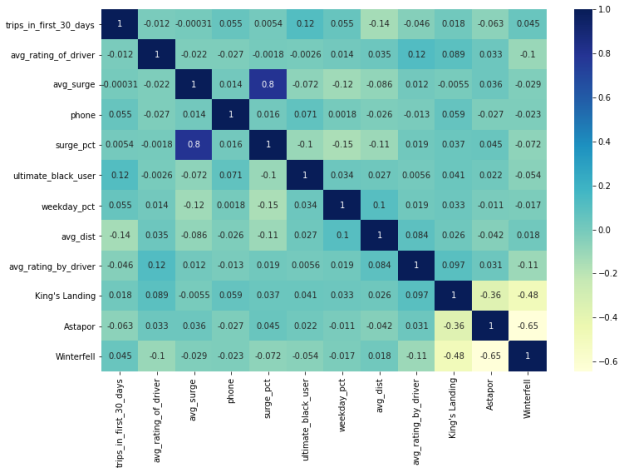
### Suggestions

1. Encourage clients to take more trips, especially short ones, and especially in the first 30 days
    a. Perhaps there could be an introductory period during which new clients are given incentives to use the service more often. Something like, "Every 5$^{th}$ trip is free in your first month!" Financial and Business analysts will have to figure this out.
    b. Pricing structure could be adjusted to make short trips more affordable, thereby encouraging users to take shorter trips and become more dependent on the service.
2. Check differences between iPhone and Android apps
    a. iPhone users are more likely to be retained. Are iPhone users having a better experience in the app? UX Researchers should investigate.
3. Investigate differences between cities
    a. Why are people in King's Landing more likely to be retained? Is that city better suited to services like this? Or are there more young people, or some other demographic that is more likely to be retained? Is marketing doing a better job in King's Landing? We need more information to investigate this.

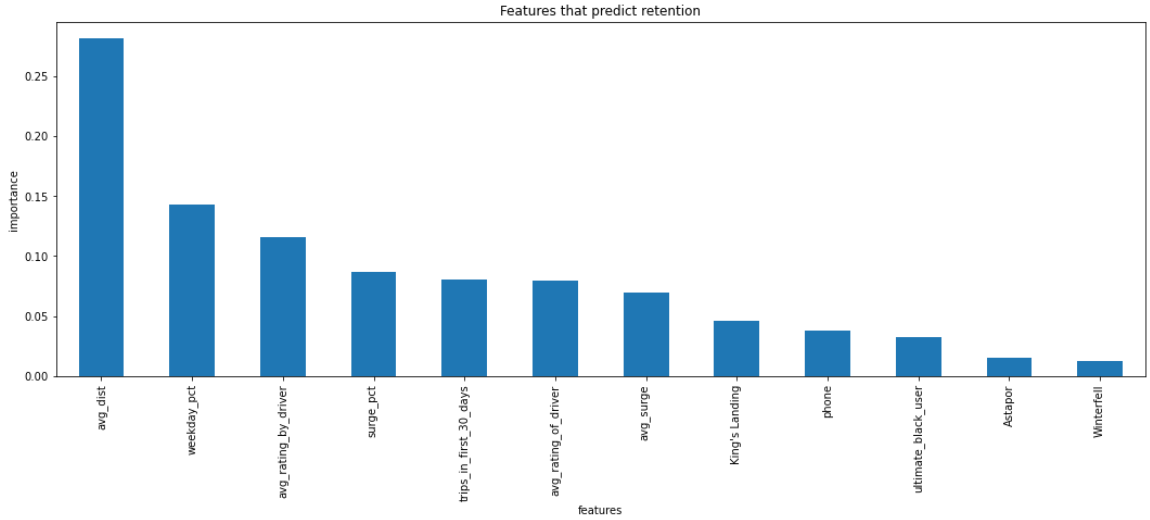
### Methodology

After bringing the data from the .json file, categorical features such as ultimate_black_user, city, and phone were one-hot encoded. Then, observations with missing values were dropped, which left us with 41,744 datapoints for modelling. The positive and negative target classes were well balanced, about 41% to 59%.

Exploratory data analysis proceeded with investigation of feature distributions and correlation between features. Most features were skewed, and the only strong correlation was between avg_surge and surge_pct.




A grid-search was performed to find the Random Forest model with the hyperparameters that maximized the Recall. Recall was selected because the goal is to predict retention, and so we want a model that captures True Positives well. The final recall was 0.69, and feature importances as shown in the graph below.




The avg_surge feature was then dropped so a logistic regression model could be fit to give insight on the magnitude and direction of the effect of each feature. Coefficients are shown in the plot below.