Report on the Company X's data challenge

From: Eugene Yankovsky
To: Analytics Department
Date: 08/01/2016

Summary

Part 1. Exploratory data analysis of user logins aggregated in 15-min intervals

There is strong evidence of the weekday and hourly cycles in a weekday in user logins in 15-min intervals that tend to gradually rise in median and variance over four months in Jan-Apr, 1970:

1. The number of logins and their spread tend to increase from Monday (3-6 logins) to Friday (6-11 logins), reach its' peak on Saturday (8-16 logins), and slightly level off on Sunday (5-14 logins in median).
2. Each weekday median logins tend to increase by 1-3 logins on average each month from January to March and stabilize in April.
3. There 2 distinctive patterns on workdays and on weekends with the slight difference between Saturday (most active day) and Sunday.

On the weekends, there are a highest peak at 2-4 am (25-28 logins), a lower peak at 2-4 pm (11-13 logins) and one trough at 7-9am (1-3 logins). In contrast to Sunday, Saturday has an extra minor peak at 10-11 pm (15 logins).

Workdays' logins tend to follow a pattern with 2 peaks and 2 troughs: a higher peak at 11 am (16-20 logins), lower peak 10 pm (12-22 logins), troughs at 6 pm (1-3 logins) and 5 pm (4-6 logins). Thursday and Friday logins tend be higher in median and spread than Monday-Wednesday logins, with Friday's second peak higher than the first one.

Part 2. Experiment and metrics design for reimbursing toll costs between two cities

Three-factor (treatment, driving regime, and city) model with randomized block experimental design is suggested to capture the effect of the toll costs reimbursement on daily average Net Cash Flow from a driver in a city. After estimating the model effects, the manager can make decision on rolling out the treatment gradually, starting from the factors' combinations generating highest NCF.

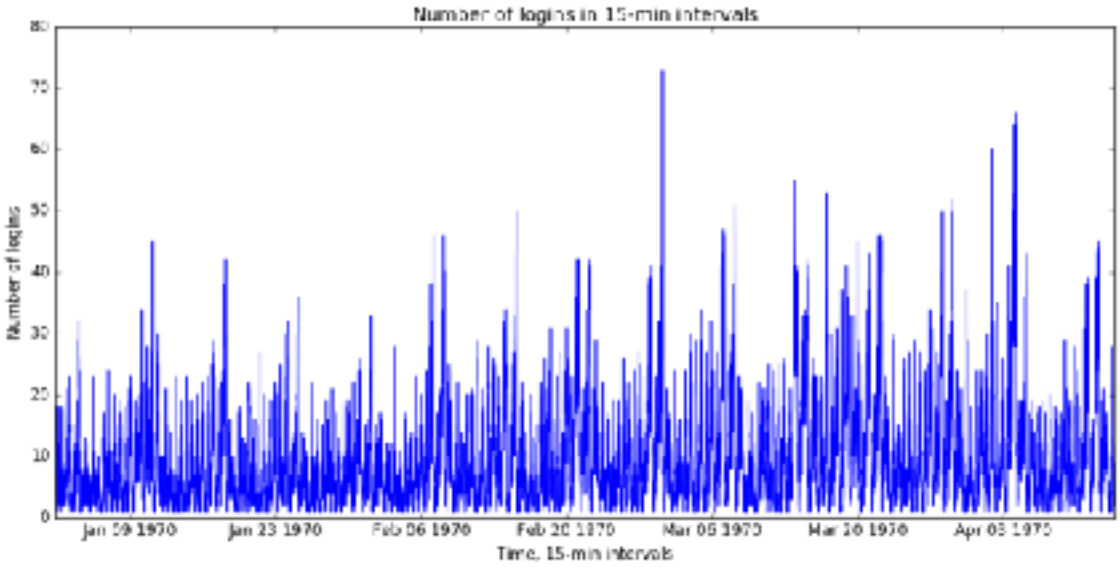
Part 3. Predictive modeling for the rider retention

Decision tree analysis with 95% accuracy discovered significant difference in the retention factors in three cities. Only rider tenure had consistent effect in all of them: lower tenure was related to higher chance of the attrition, while medium and high tenure was related to retention. The other factors tend to interact with the tenure levels. Company X manager smight consider a promotion campaign giving some incentives (e.g., discounts or using Company X Black service at the price of a regular ride) to the lower tenure riders. If there is a need to increase the data goodness of fit, the neural network model using important predictors from the decision tree analysis can be considered as the viable next step.

Part 1. Exploratory data analysis of user logins in 15-min intervals

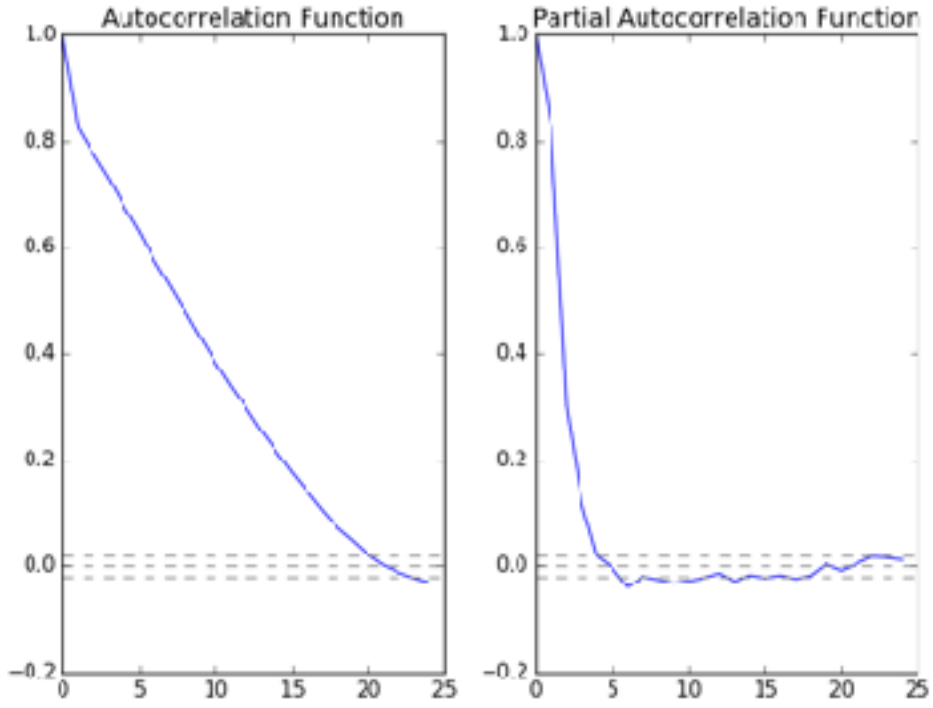
The user logins from Jan.1st to Apr.13th, 1970 were aggregated in 15-min intervals. Chart 1.1 below displays significant cyclical behavior in the data.

Chart 1.1. User logins aggregated in 15-min intervals



The ACF and PACF charts at Exhibit 1.1 below clearly indicate on high correlations between the current and the lagged observation, with significant autocorrelation in up to 20 interval lags.

Exhibit 1.1. Time series analysis of the user logins in 15-min intervals



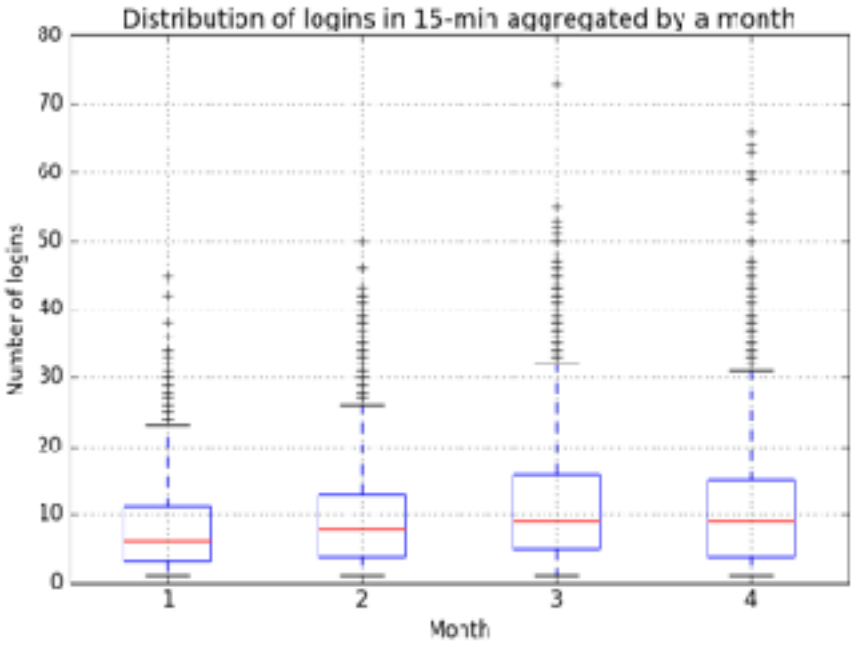
Analysis of monthly trends and seasonality

The logins in 15-min intervals aggregated monthly follow a skewed distribution with a lot of outliers in the upper value tail. The monthly median about 8-10 logins and standard deviation of about 8-12 (estimated

by 25%-75% percentile range) tend to increase from January to March-April. To note, April sample is incomplete and only 13 daily observations.

To note, it is unlikely that any non-parametric tests will detect significant difference in monthly medians because of high data spread and the median values.

Chart 1.2. Side-by-side box-plots of the user logins in 15 min intervals aggregated monthly



Note:

1, 2, 3, and 4 denote correspondingly January, February, March, and April.

Analysis of weekly trends and seasonality

Chart 1.3 below suggests that the logins aggregated weekly also follow a skewed distribution with a lot of outliers in the upper value tail. Therefore, the median is an appropriate measure of the central moment of the logins distributions.

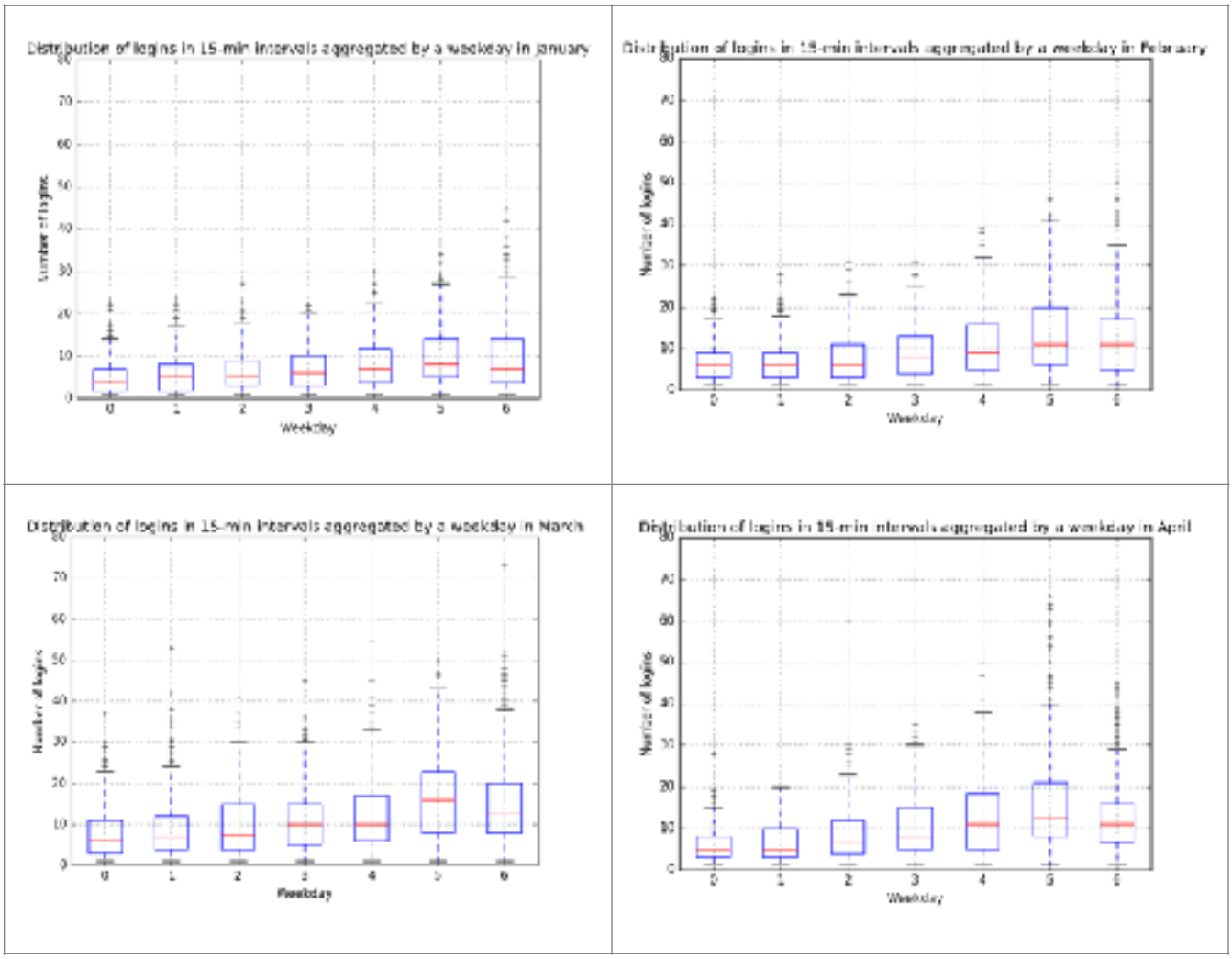
There are two distinctive trends in weekday distribution of logins:

4. The number of logins and their spread tend to increase from Monday (3-6 logins in median) to Friday (6-11 logins in median), reach its' peak on Saturday (8-16 logins in median), and slightly level off on Sunday (5-14 logins in median).
5. Each weekday median logins tend to increase by 1-3 logins on average each month from January to March and stabilize in April.

To note, statistical non-parametric tests might detect significant difference only between Monday's and Sunday's medians and no significant difference between the other days' medians because of the high data spread in comparison to the median values because of high data spread.

There is likely no interaction between weekday and month in the user logins. Therefore, all four monthly data sets will be treated as one in the next analysis.

Chart 1.3. Side-by-side box plots of logins in 15 min intervals aggregated by a weekday in a month



Note:

0, 1, 2, 3, 4, 5, and 6 denote correspondingly Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday.

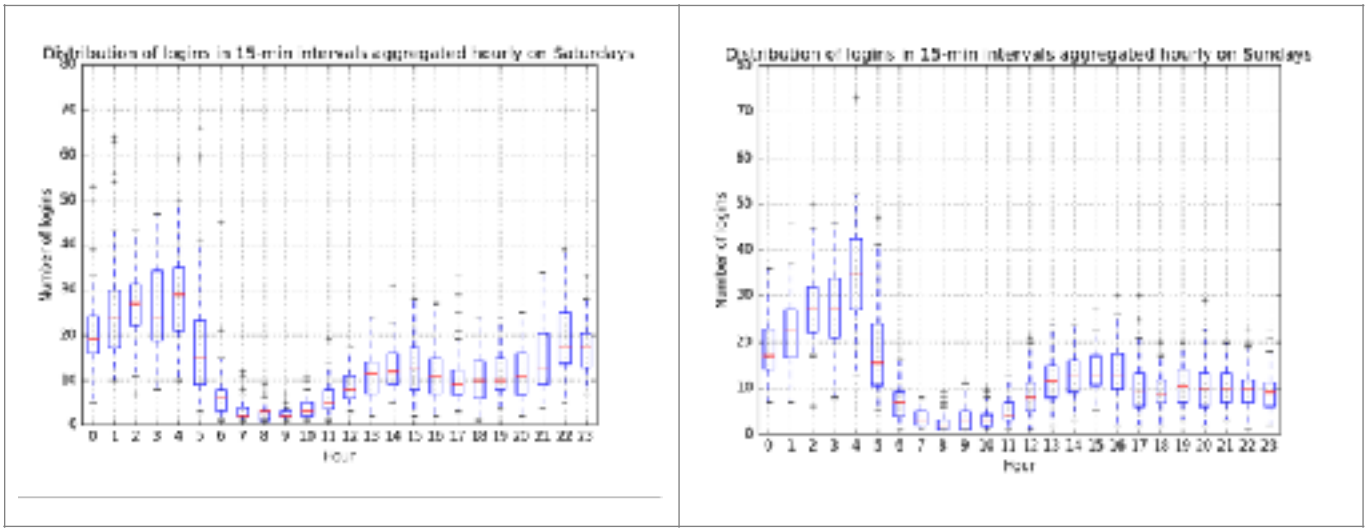
Analysis of hourly patterns in a weekday cycle

Saturdays' and Sundays' logins tend to follow similar hourly pattern with the difference around 8-11 pm. On the weekend, the number of logins in 15-min interval tend to be the lowest median (1-2 logins) and in spread (2-3 logins) at 8 am, then gradually increase to 3 pm (15.00) with the median of 12-14 logins, level off from 4 pm (16.00) to 5-6 pm with 10 logins in median.

On Sundays, the number of logins stays about 10 in median from 5 pm to 11 pm, while logins rise from 10 to 16 logins on Saturdays at the same time. See details on Chart 1.4.

To note, the non-parametric tests of equality of medians will likely detect significant difference between peak and trough logins.

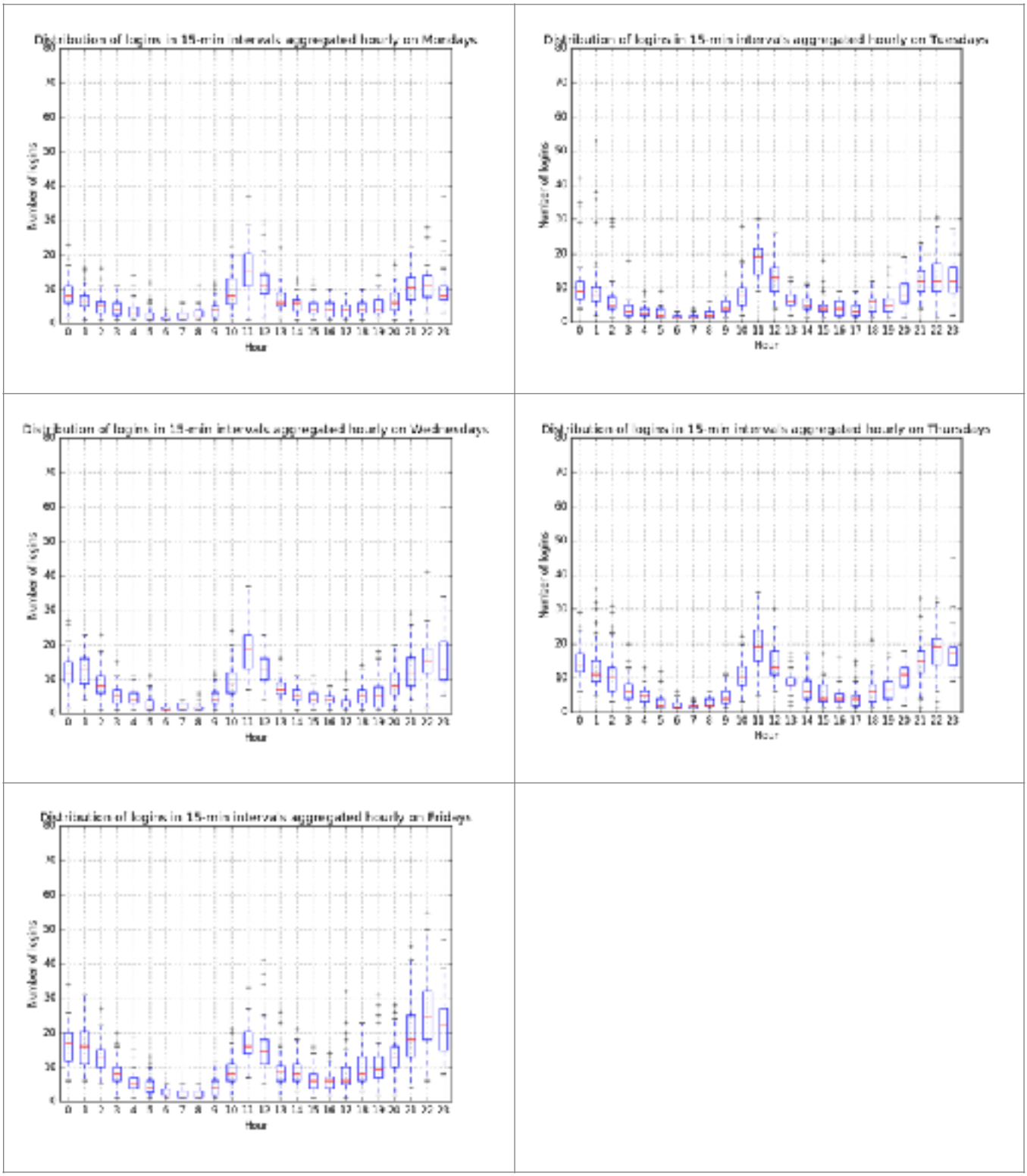
Chart 1.4. Side-by-side box-plots of logins in 15 min intervals aggregated hourly



The workdays' logins tend to follow the same wave-like pattern with two troughs and two peaks, although Thursday's and Friday's patterns are scaled up by 5-8 logins.

In the morning, the logins tend to rapidly grow in median and in variance from the global minimum of 1-3 logins in median at 6 pm to the highest daily peak of 16-20 logins at 11 am. Then the users' activity gradually decline to 4-6 in median around 5 pm (17.00), reach the next peak of 12-18 in median at 10 pm (22.00), and then gradually decline down to 1-3 at 6pm. To note, the first peak is usually higher than the second on Monday-Wednesday, and lower than the second on Friday.

Chart 1.5. Side-by-side box-plots of logins in 15 min intervals aggregated hourly



Part 2. Experiment and metrics design for reimbursing toll costs between two cities

2.1. The key measures of success of the experiment are Net Cash Flow indicators:

Daily average NCF from drivers in a city c =

Daily total NCF from drivers in a city c =

where

$=$, is daily Net Cash Flow generated by driver i in a driving regime r in a city c;

is a sample size in a pilot study or in population of drivers in a city c if the treatment is rolled out to all drivers.

Motivation:

1. NCF is commonly used financial metric of the business efficiency that can be easily understood by managers with financial and accounting view on doing business.
2. Using daily NCF at the driver's level will allow statistical testing of the effect of the bridge toll reimbursement;
3. Since there is business intelligence on significant difference between weekend and workday driving regimes only, we will ignore effects of the specific days (Sunday, Monday, Tuesday, etc.) on the driver's NCF following Ockham's razor problem-solving principle[1].
4. Adding an effect of the city into the model can give valuable information on significance of a) cities' main effect and b) interaction with the treatment and driving regimes. This information can be critical for making decision on rolling up the treatment from a sample in a pilot study to the population of drivers in both cities.

[1] See details on Ockham's razor at https://en.wikipedia.org/wiki/Occam%27s_razor/

2.2. The experimental design and application

2.2.1. Model assumptions

The distributions of are from the same log-normal family with the different means to account for the potentially significant difference between the driving regimes in two cities.

Table 2.2.1. Assumptions on distribution of

Driving regime	**Gotham city**	**Metropolis city**
Weekend		
Workday daytime		
Workday night time		

(2.2.1)

where

= overall mean;

= treatment effect of a toll reimbursement with two levels of treatment: 1 or “yes” for giving treatment, 0 or “no” for not giving (control).

= fixed main effect of a driving regime to account for differences in daily driver’s NCF related to 3 regime effects identified by business with 3 effect levels: 1) weekend, 2) workday daytime, 3) workday night time;

= fixed main effect of a city of driver’s residence to account for likely differences in daily driver’s NCF related to the cities’ specifics (i.e., demand and supply for driving services).

= fixed interaction effect of the treatment and driving regime to account for potential interaction between treatment effects in different driving regimes;

= fixed interaction effect of the treatment and cities to account for potential interaction between treatment effects in different cities;

= fixed interaction effect of the driving regimes and cities to account for potential interaction between driving regimes in different cities;

= fixed interaction effect of the treatment, driving regime, and cities’ effect to account for potential interaction between all three factors;

= driver (experimental unit) i nested in both treatment j and city c of the driver’s residence with variance between experimental units,)

= measurement error with variance between the measurements on an experimental unit,)

2. Study design and application

To reduce effect of heterogeneity between the drivers’ and consequently reduce the variance in observational unit (, I offer to use Randomized Complete Block Design (also called Latin Square approach) with 3 blocking factors: 1) driving regime, 2) city of drivers’ residence, 3) drivers. The treatment of two levels (1 or “yes” to toll reimbursement, 0 or “no” to toll reimbursement) are randomly assigned to

drivers from each cities. In the simplified example below, there is a Latin cube with 3 driving regimes, 2 cities, and 6 randomly chosen drivers (3 drivers nested in each city).

Table 2.2.2. A simplified example of assigning treatments (yes/no) to the random sample of drivers.

Driving regime (fixed factor)	**Gotham city (fixed factor, c = 1)**			**Metropolis city (fixed factor, c = 2)**		
	Driver 1	Driver 2	Driver 3	Driver 4	Driver 5	Driver 6
Weekend (r=1)	Yes (1)	No (0)	No (0)	No (0)	Yes (1)	Yes (1)
Workday daytime (r=2)	No (0)	Yes (1)	No (0)	Yes (1)	No (0)	Yes (1)
Workday night time (r=3)	Yes (1)	Yes (1)	Yes (1)	No (0)	No (1)	Yes (1)

Note:

Drivers 1, 2, and 3 are randomly selected from Gotham Company X drivers;

Drivers 4, 5 and 6 are randomly selected from Metropolis Company X drivers.

Caveat:

Days with driving regimes are preferred to be assigned randomly. However, it is strongly advised to exclude holidays (New Year, Christmas, Thanksgiving, 4th of July, other days that may reflect local cultural and religious traditions) from the list of candidate days because they will result in outlying observations in drivers' NCF.

3. Sample size

It is desired to have a sample size that will allow detecting significant difference for each fixed effect, either main or interaction ones. The sample size can be obtained from the statistical tables or estimated by special software for the power of F-test for the null hypothesis : treatment effect = 0:

Power of *F*-test = (1-β) = Pr{F* >F(1 - α; r, }

where

Power of *F*-test = (1-β) is probability of rejecting null hypothesis when it is false or the power of the test;

(1-β) where β is the probability of Type II error. It is usually set 0.90 or higher;

(1- α) where α is probability of Type I error. Test α-level that is usually 0.01 or 0.05;

is a noncentrality parameter that is a function

is the standard deviation parameter that can be estimated from the Company X historical data daily driver's NCF;

D is the treatment effect or the difference between the treatment mean and the means across all levels of this treatment that is significant for Company X.

The easier solution will have equal size samples for each block in the experimental design.

The formulas for calculating – parameters for 3 main effects, 2 two-way interaction effects and three-way interaction effect are available in the statistical textbooks[2]. Solving a system of sequential equations for each of the effects' sample size will all calculate all of them.

[2] All formulas and details of their applications for calculating sample sizes for main and interaction effects in three-factor studies can be found on p.717, 862, and 1021 of "Applied linear statistical models" by Kutner, Nachtsheim, Neter, and Li, 5th edition, 2005

2.3. Interpretations of the results and recommendations to the city operations team

The key outcome of the model (2.2.1) parameters' estimation will be:

1. **Size of all main and interaction effects**

To note, the main effect of treatment (toll costs reimbursements), its' two- and three-way interaction effects with the driving regime and the city will be of primary ineterest. The driving regime's and city's main and interaction effect are needed to control their effects on the driver's daily NCF.

2. **Results of statistical tests**

Once the two items above are calculated, the business decision matrix for the daily average and total $NCF_{(r,c)}$ from drivers in each city can be aggregated. See a hypothetical example in Table 2.3.1 and consequent Table 2.3.2 assuming all main and interaction effects were found statistically different from 0.

Table 2.3.1. An example of the models' estimated aggregated into daily average NCF per driver

Driving regime	Gotham city	Metropolis city
Weekend	$120	$130
Workday daytime	$56	$200
Workday night time	$265	$50

Table 2.3.1. An example of a business matrix with estimated daily total $NCF_{(r,c)}$ for all drivers in each city

Driving regime	Gotham city	Metropolis city
Weekend	$12,000 (4)	$13,000 (3)
Workday daytime	$5,600 (5)	**$20,000 (2)**
Workday night time	**$26,500 (1)**	$5,000 (6)

Based on the business decision matrix in Table 2.3.1 and potential budget constraints, Company X might consider rolling the treatment gradually, starting from the factor combinations generating the highest NCF (see rank of business attractiveness in the Table 2.3.1 brackets) to lower ones.

To note, Company X managers should monitor the effectiveness of the program because the effect of the treatment might have diminishing return over time. It can be related to the market saturation with Company X's services, the competitors' counter moves, changes in government regulations, and others.

Part 3. Predictive modeling for the rider retention

3.1. Plan for predictive modeling analysis

Data cleaning and transformation described in the next section 3.2 are driven by intended modeling approaches that may consist of 2 stages:

Stage 1. Decision Tree Analysis using CHAID method and 10-fold cross-validation

This step is done to utilize the advantages of the decision tree analysis:

1. Use in modeling **all** observations including the missing ones that will be classified as a valid value;
2. Build classification predictive model since it is robust to the outliers;
3. Do segmentation of data by the city and build a separate model for each city, if the city variable happens to be among the most important in a model using all data. The segmentation by the city has two advantages:
   a) Potential user of the analysis are city operations teams that will appreciate having classification rules and inferences that are tuned to their cities;
   b) Spliting the data in more homogeneous segments should improve overall data fit by predictive models.
4. To discover all interactions between explanatory variables however intricate their interaction are;
5. If the decision tree model has perfect accuracy, use the model for
   a) making inferences about predictors' effects and
   b) predicting rider retention by scoring the new data even if the new data include missing values.

Step 2. Application of the Neural Network modeling

If a) the decision tree accuracy is not satisfactory and b) percentage of missing values are relatively small, build the neural network model:

1. Since the neural network model is notorious for over-fitting the data and issues of generalizing the model to the new data, we will limit a pool of candidate covariates by the list of predictors with high importance in the decision tree analysis on stage 1;
2. The neural network models will be built using only non-missing values. As a result, if the model's significant predictor with missing values in the new data will result in missing value in the dependent variable;
3. Since the neural network is sensitive to the outliers, we will use dummy variables to isolate effect of the outliers on the retention variable and standardization of the continuous variables;
4. A neural network model often has complex predictors combinations in its' layers that makes it hard if possible) to use for making inferences on sign and size of the predictors' effects. Therefore, the neural network will be used only for developing a predictive model.

3.2. Data cleaning and transformations

Since most of continuous variables' histograms and quintile-quintile (Q-Q) plots suggested (see an example on Chart 3.2.1) that they follow a distribution with significant skew and fat tail (outliers), the following transformations were undertaken:

1) natural log transformation to normalize the skew;
2) dummy variables were assigned to the values associated with the 99% percentile in the fitted normal distribution.

Application of both log-transformed predictor and the dummy variable for its' outliers is intended to isolate effect of outliers on the predictors' coefficient and improve goodness-of-fit. See detailed description of transformations in Table 3.2.21.

Decision tree analysis was done both using SPSS (first) and Python (second). Since Python decision tree was more restrictive than SPSS, transformations in their predictors were different.

The decision tree estimated in SPSS included a list of the following predictors:

city, phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_weekday_pct, ln_surge_pct, ln_trips_in_first_30_days

The decision tree estimated in Python included a list of the following predictors:

astapor, kings_landing, winterfell, iphone, android, Company X_black, tenure, avg_dist, avg_rating_by_driver, avg_rating_of_drver, weekday_pct, surge_pct, surge_avg, trips_in_first_30_days

1. Since decision trees in Python do not accept missing values as a valid class, all missing observations in both Average rating by driver and Average rating of driver were substituted with a value 99.
2. Since Python decision trees, do not accept negative values, all predictors were used without natural logarithm transformation.
3. Since decision tree analysis in Python, do not accept non-numeric values in predictors:

a) instead of city variable with "Astapor", "King's Landing", and "Winterfell" values, the analysis used three dummy variables taking 1 if the city is one these cities values and 0 otherwise;

b) instead of phone with "iPhone" and "Android" values, the analysis used two dummy variables taking 1 if the phone is one of the phone types and 0 otherwise.

Chart 3.2.1. Histograms for key predictors

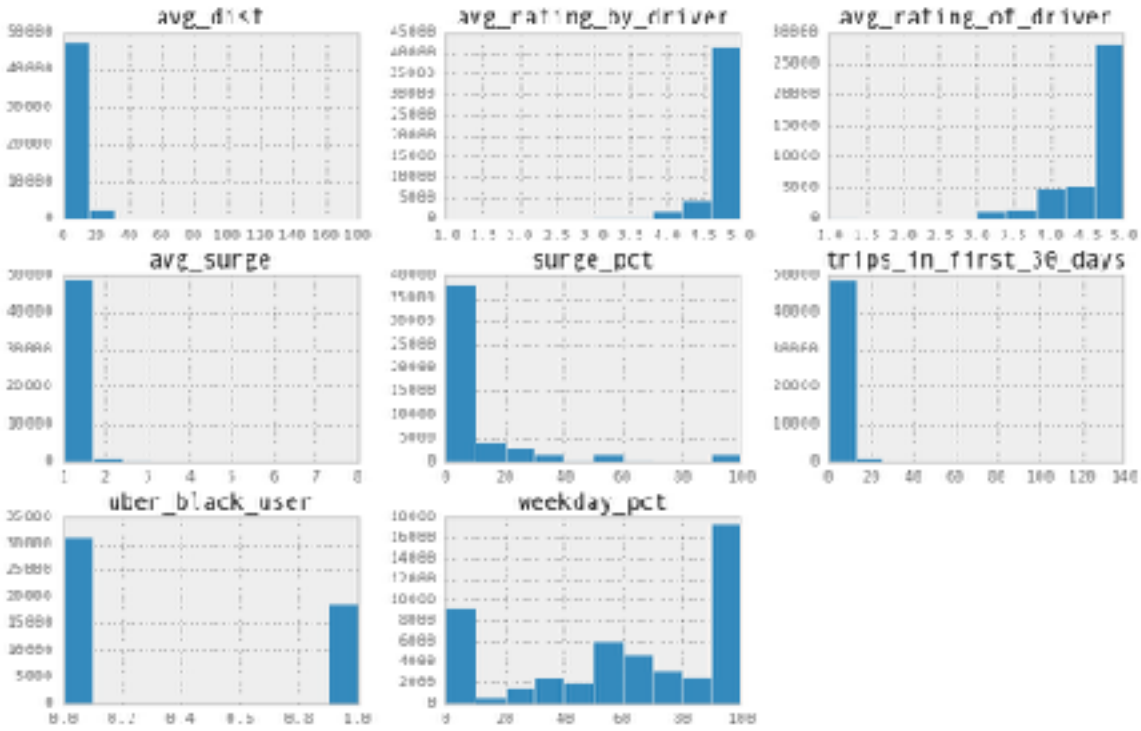
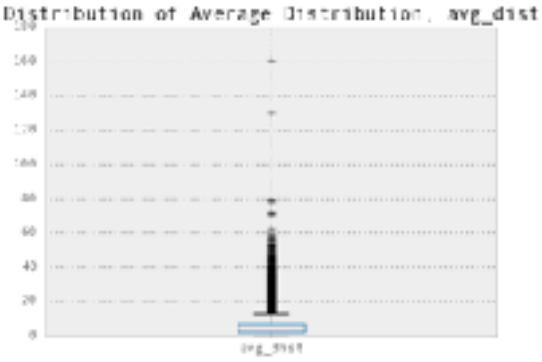
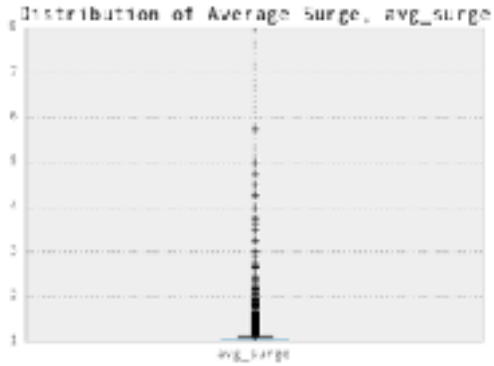
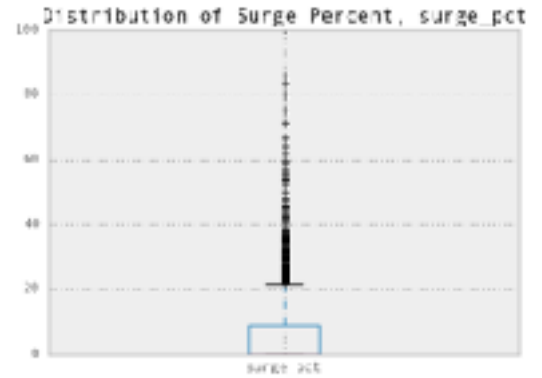
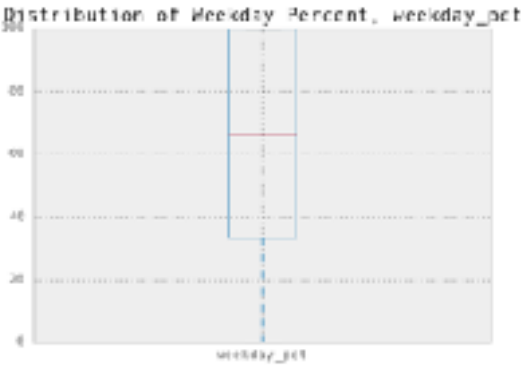

avg_dist
avg_rating_by_driver
avg_rating_of_driver
avg_surge
surge_pct
trips_in_first_30_days
uber_black_user
weekday_pct

Chart 3.2.2 Chart 3.2.1. Box plots for key predictors

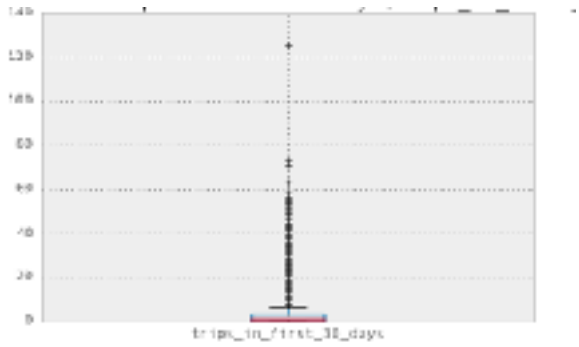




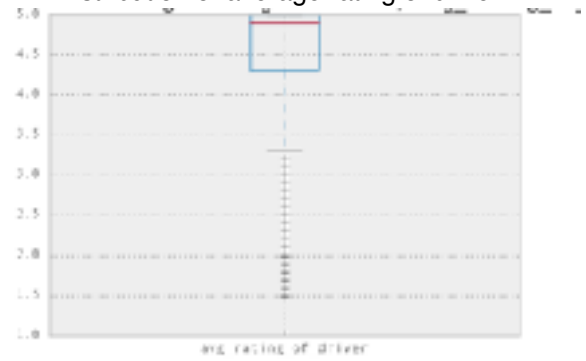




Distribution of trips in first 30 days



Distribution of average rating of driver



Detailed list of all transformations applied is provide in Table 3.2.1

Table 3.2.1. List of the data transformations

No	New variable name	Transformation	Reason for the transformation	Usage in the models
1	Retention variable to identify riders that are active in their 6th month on the system	if last_trip_date>='01jun2014'd then active2=1; else active2=0;	Model's dependent variable	Decision tree, Neural network
2	Tenure with the company	tenure = last_trip_date - signup_date;	Candidate predictor	Decision tree, Neural network

3	Trips in the 1st 30 days:
	Dummy variables for the outliers
		If trips_in_first_30_days>30
		then trips_in_first_30_days_outlier=1;
		else trips_in_first_30_days_outlier=0;
			Candidate predictor	Neural network (only)

4	Trips in the 1st 30 days: ln transformation to normalize a right skewed distribution with fat upper tail	ln_trips_in_first_30_days = log(trips_in_first_30_days+0.001);	Candidate predictor	Decision tree, Neural network
5	Average surge: ln transformation to normalize a right skewed	ln_avg_surge=log(avg_surge+0.001);	Candidate predictor	Decision tree, Neural network
6	Weekday trips percentage: ln transformation to normalize a right skewed	ln_weekday_pct=log(weekday_pct+0.001);	Candidate predictor	Decision tree, Neural network
7	Average rating by driver: substituting 'NaN' with missing value to keep the variable as numeric (not character one)	if avg_rating_by_driver='NaN' then avg_rating_by_drv=.; else avg_rating_by_drv= avg_rating_by_driver;	Candidate predictor	Decision tree, Neural network
8	Average rating of driver: substituting 'NaN' with missing value to keep the variable as numeric (not character one)	if avg_rating_of_driver='NaN' then avg_rating_of_drv=.; else avg_rating_of_drv= avg_rating_of_driver;	Candidate predictor	Decision tree, Neural network

No	New variable name	Transformation	Reason for the transformation	Usage in the models
9	City: Dummy variables for each city	If city=”Astapor” then astapor=1; else astapor=0; If city=”King’s Landing” then kingsland=1; else kingsland=0; If city=”Winterfell” then winterfell=1; else Winterfell=0;	Candidate predictor; segmentation variable	Decision tree, Neural network
10	Company X black user: binary transformation for the neural model	if Company X_black_user='TRUE ' then Company X_black=1; else Company X_black=0;	Candidate predictor	Decision tree, Neural network
11	Iphone dummy: binary transformation for the neural model	if phone='iPhone ' then iphone=1; else iphone=0;	Candidate predictor	Decision tree, Neural network (only)
12	Android dummy: binary transformation for the neural model	if phone='Android' then android=1; else android=0;	Candidate predictor	Decision tree, Neural network
13	Average surge: Dummy variable for the outlying values	if avg_surge>7 then avg_surge_outlier=1; else avg_surge_outlier=0;	Candidate predictor	Neural network (only)
14	Average distribution: Dummy variable for the outlying values	If avg_dist>80 then avg_dist_outlier=1; else avg_dist_outlier=0;	Candidate predictor	Neural network (only)

Missing observations

Several transformations required for the neural modeling resulted in missing values:

1. As a result of creating binary variables for Iphone and Android, 396 missing values in “phone” variable will be ignored.

2. Substituting ‘NaN’ with missing value in “average rating by driver” and “average rating of driver” will result in 201 (0.4%) and 8,122 (16.2%) missing observations.

As a result, 8,256 (16.5%) observations will be missing and might be ignored while building the neural networking model if the covariates with missing values are selected as predictors. In contrast to the neural network modeling, all observations will be used in decision tree modeling.

Note:

Since decision tree analysis in Python do not accept missing values as a valid class, all missing observations in both Average rating by driver and Average rating of driver were substituted with a value 99.

3.3. Decision Tree Analysis using CHAID method and 10-fold cross-validation results

3.3.1. Decision tree analysis results for the whole data set

Model estimated in SPSS	Model estimated in Python
Candidate predictors: city, phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_weekday_pct, ln_surge_pct, ln_trips_in_first_30_days	**Candidate predictors:** astapor, kings_landing, winterfell, iphone, android, Company X_black, tenure, avg_dist, avg_rating_by_driver, avg_rating_of_drver, weekday_pct, surge_pct, surge_avg, trips_in_first_30_days
Selected important predictors (sorted by importance): tenure, city, phone, Company X_black_user, ln_trips_in_first_30_days	**Selected important predictors sorted by importance):** tenure, weekday_pct, kings_landing, avg_rating_of_driver, trips_in_30_days, iphone, android, Company X_black_user, astapor
Risk Method / Estimate / Std. Error Resubstitution / .244 / .001 Cross-Validation / .245 / .001	**Accuracy** Method Estimate Std. Error Cross-Validation 0.957 0.005
Classification Predicted / Percent Correct 0 / 1 29993 / 1203 / 96.1% 998 / 17006 / 94.7% 62.0% / 38.0% / 95.6%	**Classification** Observed / Predicted / Percent Correct 0 / 1 0 / 6964 / 412 / 94.41% 1 / 182 / 4966 / 96.46% Overall Percentage / 57.06% / 42.94% / 95.26%
See more analysis details in Appendix A.3.3.1 and decision tree chart in /EYankovskyCompany XChallengeAppendix/ Part3_dtree_all_SPSS.pdf	See more analysis details in Appendix B.3.3.1 and decision tree chart in /EYankovskyCompany XChallengeAppendix/ Part3_dtree_all.svg

Conclusion:

Difference in model specifications is caused by different transformations applied in SPSS and Python. The last one had more constraints on predictors to be non-missing, non-negative and numeric. Due to time constraints on this report, conclusions below are derived using SPSS results only.

Since the city is one of the most important factors, we will apply segmentation of all data by city first and apply decision tree analysis to each city data. This segmentation will give more information on the cities' specifics that will result in better model fit and give better business insight.

3.3.2. Decision tree analysis results for Astapor

Model estimated in SPSS	Model estimated in Python
Candidate predictors: phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_weekday_pct, ln_surge_pct, ln_trips_in_first_30_days	**Candidate predictors:** iphone, android, Company X_black, tenure, avg_dist, avg_rating_by_driver, avg_rating_of_drver, weekday_pct, surge_pct, surge_avg, trips_in_first_30_days
Selected important predictors (sorted by importance): tenure, ln_weekday_pct, phone, Company X_black_user, ln_trips_in_first_30_days	**Selected important predictors (sorted by importance):** tenure, avg_rating_by_driver, avg_rating_of_driver, avg_surge, surge_pct, avg_dist, trips_in_30_days, Company X_black_user, iphone
Risk Method / Estimate / Std. Error Resubstitution .040 .002 Cross-Validation .051 .002	**Accuracy** Method / Estimate / Std. Error Cross-Validation 0.964 0.016
Classification Predicted: 0 / 1 / Percent Correct 11776 530 95.7% 273 3955 93.5% 72.9% 27.1% 95.1%	**Classification** Observed / Predicted: 0 / 1 / Percent Correct 0 1057 78 93.13% 1 44 1860 97.69% Overall Percentage 36.23% 63.77% 95.99%
See more analysis details in Appendix A.3.3.2 and decision tree chart in /EYankovskyCompany XChallengeAppendix/ Part3_dtree_astapor_SPSS.pdf	See more analysis details in Appendix B.3.3.2 and decision tree chart in /EYankovskyCompany XChallengeAppendix/ Part3_dtree_astapor_Python.svg

3.1.3. Decision tree analysis results for King's Landing

Model estimated in SPSS	Model estimated in Python
Candidate predictors: phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_weekday_pct, ln_surge_pct, ln_trips_in_first_30_days	**Candidate predictors:** iphone, android, Company X_black, tenure, avg_dist, avg_rating_by_driver, avg_rating_of_drver, weekday_pct, surge_pct, surge_avg, trips_in_first_30_days
Selected important predictors sorted by importance): tenure, Company X_black_user, phone, ln_trips_in_first_30_days	**Selected important predictors sorted by importance):** tenure, avg_dist, avg_rating_by_driver, android, Company X_black_user, trips_in_first_30_days, weekday_pct
Risk Method / Estimate / Std. Error Resubstitution .045 .002 Cross-Validation .045 .002	**Accuracy** Method / Estimate / Std. Error Cross-Validation 0.964 0.016
Classification Observed / Predicted (0, 1, Percent Correct) 0 3569 128 96.7% 1 257 6106 96.0% Overall Percentage 37.8% 62.2% 96.5%	**Classification** Observed / Predicted (0, 1, Percent Correct) 0 1023 99 91.18% 1 19 1898 99.01% Overall Percentage 34.29% 65.71% 96.12%
See more analysis details in Appendix A.3.3.3 and decision tree chart in /EYankovskyCompany XChallengeAppendix/ Part3_dtree_kingslanding_SPSS.pdf	See more analysis details in Appendix B.3.3.3 3 and decision tree chart in /EYankovskyCompany XChallengeAppendix/ Part3_dtree_kingslanding_Python.pdf

3.3.4. Decision tree analysis results for Winterfell

<table>
<tr><th>Model estimated in SPSS</th><th>Model estimated in Python</th></tr>
<tr><td>Candidate predictors:
phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_weekday_pct, ln_surge_pct, ln_trips_in_first_30_days</td><td>Candidate predictors:
iphone, android, Company X_black, tenure, avg_dist, avg_rating_by_driver, avg_rating_of_drver, weekday_pct, surge_pct, surge_avg, trips_in_first_30_days</td></tr>
<tr><td>Selected important predictors (sorted by importance):
tenure, ln_avg_dist, Company X_black_user, phone, ln_trips_in_first_30_days, ln_surge_pct</td><td>Selected important predictors (sorted by importance):
tenure, weekday_pct, trips_in_first_30_days, Company X_black_user, iphone, surge_pct, avg_surge, android</td></tr>
<tr><td>Risk
<table><tr><th>Method</th><th>Estimate</th><th>Std. Error</th></tr><tr><td>Resubstitution</td><td>.047</td><td>.001</td></tr><tr><td>Cross-Validation</td><td>.047</td><td>.001</td></tr></table></td><td>Accuracy
Method Estimate Std. Error
Cross-Validation 0.956 0.007</td></tr>
<tr><td>Classification
<table><tr><th rowspan="2">Observed</th><th colspan="2">Predicted</th><th rowspan="2">Percent Correct</th></tr><tr><th>0</th><th>1</th></tr><tr><td>0</td><td>14351</td><td>772</td><td>94.9%</td></tr><tr><td>1</td><td>315</td><td>7898</td><td>95.2%</td></tr><tr><td>Overall Percentage</td><td>62.8%</td><td>37.2%</td><td>95.3%</td></tr></table></td><td>Classification
Observed Predicted
0 1 Percent Correct
0 4326 144 96.78%
1 196 2335 92.26%
Overall Percentage 64.59% 35.41% 95.14%</td></tr>
<tr><td>See more analysis details in Appendix A.3.3.4 and decision tree chart in
/EYankovskyCompany XChallengeAppendix/ Part3_dtree_winterfell_SPSS.pdf</td><td>See more analysis details in Appendix B.3.3.4 and decision tree chart in
/EYankovskyCompany XChallengeAppendix/ Part3_dtree_winterfell_Pyhton.svg</td></tr>
</table>

3.4. Results of the Neural Network modeling

Since the decision trees in each city data set have a good 95% accuracy using all data (including 16% missing data), the decision was made to refrain from doing neural network modeling.
If the 95% accuracy is not sufficient for the business user, the neural network modeling can be estimated for the predictors that are proved to be important in the decision tree analysis above.

3.5. Insights from the predictive modeling

Analysis was done using both SPSS and Python. Since decision tree analysis in Python has more constraints for predictors (non-missing, non-negative, and numeric values), the models' lists of predictors varied in transformations and resulted in different models' specifications in SPSS and Python. Due to time constraints on this project, conclusions below are derived from SPSS model only. To note, tenure variable was important predictor in both SPSS and Python model with the same direction of the effect.

Results of the decision tree analysis suggest that factors affecting riders' decision to keep using Company X service vary significantly from city to city. Only rider tenure had consistent effect in all 3 cities: being a lower tenure (below 120-140 days) rider tend to increase probability of attrition, while medium to high tenure is associated with staying with Company X. Other factors seem to interact with the tenure levels. See details in Table 3.4 below.

For the business, these findings imply

1) The first impression does matter

It could be reasonable to consider a promotion campaign that will give some incentives (e.g., discounts or using Company X Black at the price of a regular ride) to lower tenure riders and let them move to medium tenure group with lower churn risk.

2) Pay attention to the city's specifics

Riders in three cities seem to have different factors behind their decision to stay with Company X.

Table 3.4. Summary of the effect of significant predictors of probability of a rider retention

No	Predictor	Astapor	King's landing	Winterfell
1	Tenure (last_trip_date – signup_date)	**Negative** for low tenure riders **Positive** for high tenure riders	**Negative** for low tenure riders **Positive** for medium and high tenure riders	**Negative** for low tenure riders **Positive** for medium and high tenure riders
2	The percent of weekday trips (ln_weekday_pct)	**Negative** only for low tenure riders, and medium tenure Android users	Not relevant	Not relevant
3	Primary device for a user (phone)	A decision maker for medium tenure: **Negative** for Android user's **Positive** for Iphone users In other cases, effects follow tenure level: Having any device is bad for low tenure and good for high tenure.	Multiple mixed effects although it matters only for low-medium tenure riders	A decision maker for medium tenure: **Negative** for Android user's **Positive** for Iphone users
4	If a rider took an Company X Black service in the first 30 days (Company X_black_user)	**Positive**, but only matters to medium tenure and Iphone users	**Positive** for low tenure, and medium tenure with more trips in the first 30 days	**Positive** for medium tenure users with iPhone and higher tenure riders; correlates with tenure in either value (yes/no): negative for low, positive for high tenure.
5	Number of trips in the first 30 days (ln_trips_in_first_30 _days)	**Positive,** any number of trips is good for medium tenure riders (relevant for next splits by other factors)	**Positive,** any number of trips is good for medium and high tenure riders (relevant for next splits by other factors).	**Negative** for medium tenure users with Androids for any number of trips **Positive** for high tenure riders for any number of trips
6	The average distance per trip in first 30 days (ln_avg_dist)	Not relevant	Not relevant	**Negative** for low tenure for any distance (relevant for next splits by other factors).
7	The percent of trips taken with multiplier >1 (ln_surg_pct)	Not relevant	Not relevant	**Positive** in any percent for medium-high tenure

Note:
The tenure split levels vary in different cities. For the sake of explaining in this table, they are aggregated in groups:
Low tenure is less than ~140 days;
Medium tenure is (140; 150] days;
High tenure is more than 150 days.
Check the decision tree chart and tables for the details.

Appendix A.3.3.1. Part 3 results of decision tree modeling for all data

Model Summary

Specifications	Growing Method	CHAID
	Dependent Variable	active2
	Independent Variables	city, phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_trips_in_first_30_days, ln_weekday_pct, ln_surge_pct
	Validation	Cross Validation
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	tenure, city, phone, Company X_black_user, ln_trips_in_first_30_days
	Number of Nodes	34
	Number of Terminal Nodes	21
	Depth	3

Growing Method: CHAID

Dependent Variable: active2

Risk

Method	Estimate	Std. Error
Resubstitution	.044	.001
Cross-Validation	.045	.001

Classification

	Predicted		
Observed	0	1	Percent Correct

0	29993	1203	96.1%
1	998	17806	94.7%
Overall Percentage	62.0%	38.0%	95.6%

Decision tree chart is attached to this report in a file

/EYankovskyCompany XChallengeAppendix/Part3_dtree_all_SPSS.pdf

Tree Table

Node	0		1		Total		Predicted Category	Parent Node	Primary Independent Variable				
	N	Percent	N	Percent	N	Percent			Variable	Sig.[a]	Chi-Square	df	Split Values
0	31196	62.4%	18804	37.6%	50000	100.0%	0						
1	2497 1	100.0%	0	.0%	24971	49.9%	0	0	tenure	.000	42477.814	4	<= 109.0
2	4484	88.0%	613	12.0%	5097	10.2%	0	0	tenure	.000	42477.814	4	(109.0, 130.0]
3	1615	31.7%	3479	68.3%	5094	10.2%	1	0	tenure	.000	42477.814	4	(130.0, 145.0]
4	126	2.6%	4800	97.4%	4926	9.9%	1	0	tenure	.000	42477.814	4	(145.0, 154.0]
5	0	.0%	9912	100.0%	9912	19.8%	1	0	tenure	.000	42477.814	4	> 154.0
6	1877	91.6%	172	8.4%	2049	4.1%	0	2	city	.000	144.521	2	Astapor
7	507	74.4%	174	25.6%	681	1.4%	0	2	city	.000	144.521	2	King's Landing
8	2100	88.7%	267	11.3%	2367	4.7%	0	2	city	.000	144.521	2	Winterfell
9	641	43.8%	824	56.2%	1465	2.9%	1	3	city	.000	353.291	2	Astapor
10	184	12.8%	1251	87.2%	1435	2.9%	1	3	city	.000	353.291	2	King's Landing
11	790	36.0%	1404	64.0%	2194	4.4%	1	3	city	.000	353.291	2	Winterfell
12	51	4.6%	1059	95.4%	1110	2.2%	1	4	city	.000	42.527	2	Astapor
13	13	.8%	1716	99.2%	1729	3.5%	1	4	city	.000	42.527	2	King's Landing
14	62	3.0%	2025	97.0%	2087	4.2%	1	4	city	.000	42.527	2	Winterfell
15	710	94.9%	38	5.1%	748	1.5%	0	6	phone	.000	16.827	1	Android

16	1167	89.7%	134	10.3%	1301	2.6%	0	6	phone	.000	16.827	1	iPhone; <blank>
17	201	84.8%	36	15.2%	237	.5%	0	7	phone	.000	20.513	1	Android; <blank>
18	306	68.9%	138	31.1%	444	.9%	0	7	phone	.000	20.513	1	iPhone
19	1541	90.6%	160	9.4%	1701	3.4%	0	8	Company X_black_user	.000	21.212	1	FALSE
20	559	83.9%	107	16.1%	666	1.3%	0	8	Company X_black_user	.000	21.212	1	TRUE
21	242	61.3%	153	38.7%	395	.8%	0	9	phone	.000	67.389	1	Android
22	399	37.3%	671	62.7%	1070	2.1%	1	9	phone	.000	67.389	1	iPhone; <blank>
23	122	20.3%	480	79.7%	602	1.2%	1	10	ln_trips_in_first_30_days	.000	55.740	2	<= -6.908
24	29	11.0%	235	89.0%	264	.5%	1	10	ln_trips_in_first_30_days	.000	55.740	2	(-6.908, .001]
25	33	5.8%	536	94.2%	569	1.1%	1	10	ln_trips_in_first_30_days	.000	55.740	2	> .001
26	296	56.1%	232	43.9%	528	1.1%	0	11	phone	.000	121.352	1	Android
27	494	29.7%	1172	70.3%	1666	3.3%	1	11	phone	.000	121.352	1	iPhone; <blank>
28	20	10.8%	165	89.2%	185	.4%	1	12	phone	.000	19.570	1	Android
29	31	3.4%	894	96.6%	925	1.9%	1	12	phone	.000	19.570	1	iPhone; <blank>
30	12	1.3%	879	98.7%	891	1.8%	1	13	ln_trips_in_first_30_days	.016	8.719	1	<= .001

31	1	.1%	837	99.9%	838	1.7%	1	13	ln_trips_in_first_30_days	.016	8.719	1	> .001
32	44	5.0%	841	95.0%	885	1.8%	1	14	ln_trips_in_first_30_days	.000	21.344	1	<= .001
33	18	1.5%	1184	98.5%	1202	2.4%	1	14	ln_trips_in_first_30_days	.000	21.344	1	> .001

Appendix A.3.3.2. Part 3 results of decision tree modeling for Astapor

Model Summary

Specifications	Growing Method	CHAID
	Dependent Variable	active2
	Independent Variables	phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_trips_in_first_30_days, ln_weekday_pct, ln_surge_pct
	Validation	Cross Validation
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	tenure, ln_weekday_pct, phone, Company X_black_user, ln_trips_in_first_30_days
	Number of Nodes	22
	Number of Terminal Nodes	14
	Depth	3

Risk

Method	Estimate	Std. Error
Resubstitution	.049	.002
Cross-Validation	.051	.002

Classification

	Predicted		
Observed	0	1	Percent Correct
0	11776	530	95.7%
1	273	3955	93.5%
Overall Percentage	72.9%	27.1%	95.1%

The decision tree chart is attached to this report in a file
/EYankovskyCompany XChallengeAppendix/ Part3_dtree_astapor_SPSS.pdf

Tree Table

Node	0		1		Total		Predicted Category	Parent Node	Primary Independent Variable				
	N	Percent	N	Percent	N	Percent			Variable	Sig.[a]	Chi-Square	df	Split Values
0	12306	74.4%	4228	25.6%	16534	100.0%	0						
1	9901	100.0%	0	.0%	9901	59.9%	0	0	tenure	.000	13576.521	4	<= 111.0
2	1591	93.5%	111	6.5%	1702	10.3%	0	0	tenure	.000	13576.521	4	(111.0, 128.0]
3	763	46.3%	885	53.7%	1648	10.0%	1	0	tenure	.000	13576.521	4	(128.0, 145.0]
4	51	3.2%	1554	96.8%	1605	9.7%	1	0	tenure	.000	13576.521	4	(145.0, 158.0]
5	0	.0%	1678	100.0%	1678	10.1%	1	0	tenure	.000	13576.521	4	> 158.0
6	258	92.8%	20	7.2%	278	1.7%	0	2	ln_weekday_pct	.000	32.913	2	<= -6.908
7	193	85.0%	34	15.0%	227	1.4%	0	2	ln_weekday_pct	.000	32.913	2	(-6.908, 3.689]

8	1140	95.2%	57	4.8%	1197	7.2%	0	2	ln_weekday_pct	.0000	32.913	2	> 3.689
9	284	63.7%	162	36.3%	446	2.7%	0	3	phone	.0000	74.278	1	Android
10	479	39.9%	723	60.1%	1202	7.3%	1	3	phone	.0000	74.278	1	iPhone;<blank>
11	20	8.0%	231	92.0%	251	1.5%	1	4	phone	.0000	22.194	1	Android
12	31	2.3%	1323	97.7%	1354	8.2%	1	4	phone	.0000	22.194	1	iPhone;<blank>
13	437	97.5%	11	2.5%	448	2.7%	0	8	phone	.0111	8.399	1	Android
14	703	93.9%	46	6.1%	749	4.5%	0	8	phone	.0111	8.399	1	iPhone;<blank>
15	184	57.1%	138	42.9%	322	1.9%	0	9	ln_weekday_pct	.0000	21.380	1	<= 4.589
16	100	80.6%	24	19.4%	124	.7%	0	9	ln_weekday_pct	.0000	21.380	1	> 4.589
17	316	49.7%	320	50.3%	636	3.8%	1	10	Company X_black_user	.0000	54.507	1	FALSE

18	163	28.8%	403	71.2%	566	3.4%	1	10	Company X_black_user	.000	54.507	1	TRUE
19	19	4.9%	365	95.1%	384	2.3%	1	12	ln_trips_in_first_30_days	.000	20.146	2	<= -6.908
20	10	2.1%	461	97.9%	471	2.8%	1	12	ln_trips_in_first_30_days	.000	20.146	2	(-6.908, .694]
21	2	.4%	497	99.6%	499	3.0%	1	12	ln_trips_in_first_30_days	.000	20.146	2	> .694

Appendix A.3.3.3. Part 3 results of decision tree modeling for King's Landing

Model Summary

Specifications	Growing Method	CHAID
	Dependent Variable	active2
	Independent Variables	phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_trips_in_first_30_days, ln_weekday_pct, ln_surge_pct
	Validation	Cross Validation
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	tenure, Company X_black_user, phone, ln_trips_in_first_30_days
	Number of Nodes	20
	Number of Terminal Nodes	12
	Depth	3

Risk

Method	Estimate	Std. Error
Resubstitution	.045	.002
Cross-Validation	.045	.002

Classification

	Predicted		
Observed	0	1	Percent Correct
0	3569	198	94.7%
1	257	6106	96.0%
Overall Percentage	37.8%	62.2%	95.5%

The decision tree chart is attached to this report in a file

/EYankovskyCompany XChallengeAppendix/ Part3_dtree_kingslanding_SPSS.pdf

Tree Table

Node	0		1		Total		Predicted Category	Parent Node	Primary Independent Variable				
	N	Percent	N	Percent	N	Percent			Variable	Sig.[a]	Chi-Square	df	Split Values
0	3767	37.2%	6363	62.8%	10130	100.0%	1						
1	3051	100.0%	0	.0%	3051	30.1%	0	0	tenure	.000	8648.932	4	<= 108.0
2	599	60.1%	398	39.9%	997	9.8%	0	0	tenure	.000	8648.932	4	(108.0, 134.0]
3	100	9.9%	911	90.1%	1011	10.0%	1	0	tenure	.000	8648.932	4	(134.0, 144.0]
4	17	1.6%	1023	98.4%	1040	10.3%	1	0	tenure	.000	8648.932	4	(144.0, 150.0]
5	0	.0%	4031	100.0%	4031	39.8%	1	0	tenure	.000	8648.932	4	> 150.0
6	142	44.2%	179	55.8%	321	3.2%	1	2	Company X_black_user	.000	49.549	1	TRUE
7	457	67.6%	219	32.4%	676	6.7%	0	2	Company X_black_user	.000	49.549	1	FALSE
8	67	16.4%	342	83.6%	409	4.0%	1	3	ln_trips_in_first_30_days	.000	32.462	1	<= -6.908

9	33	5.5%	569	94.5%	602	5.9%	1	3	ln_trips_in_first_30_days	.000	32.462	1	> -6.908
10	16	2.9%	529	97.1%	545	5.4%	1	4	ln_trips_in_first_30_days	.003	12.057	1	<= .001
11	1	.2%	494	99.8%	495	4.9%	1	4	ln_trips_in_first_30_days	.003	12.057	1	> .001
12	81	36.5%	141	63.5%	222	2.2%	1	6	phone	.000	17.528	1	iPhone
13	61	61.6%	38	38.4%	99	1.0%	0	6	phone	.000	17.528	1	Android; <blank>
14	285	61.0%	182	39.0%	467	4.6%	0	7	phone	.000	29.822	1	iPhone
15	172	82.3%	37	17.7%	209	2.1%	0	7	phone	.000	29.822	1	Android; <blank>
16	42	13.6%	267	86.4%	309	3.1%	1	8	phone	.022	7.178	1	iPhone
17	25	25.0%	75	75.0%	100	1.0%	1	8	phone	.022	7.178	1	Android; <blank>
18	8	2.6%	298	97.4%	306	3.0%	1	9	Company X_black_user	.002	9.875	1	TRUE

19	25	8.4 %	271	91.6 %	296	2.9 %	1	9	Company X_black_user	.002	9.875	1	FALSE

Appendix A.3.3.4. Part 3 results of decision tree modeling for Winterfell

Model Summary

Specifications	Growing Method	CHAID
	Dependent Variable	active2
	Independent Variables	city, phone, Company X_black_user, tenure, ln_avg_dist, avg_rat_by_drv, avg_rat_of_drv, ln_trips_in_first_30_days, ln_weekday_pct, ln_surge_pct
	Validation	Cross Validation
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	tenure, ln_avg_dist, Company X_black_user, phone, ln_trips_in_first_30_days, ln_surge_pct
	Number of Nodes	23
	Number of Terminal Nodes	14
	Depth	3

Risk

Method	Estimate	Std. Error
Resubstitution	.047	.001
Cross-Validation	.047	.001

Classification

Observed	Predicted		
	0	1	Percent Correct
0	14351	772	94.9%
1	315	7898	96.2%
Overall Percentage	62.8%	37.2%	95.3%

The decision tree chart is attached to this report in a file
/EYankovskyCompany XChallengeAppendix/Part3_dtree_winterfell_SPSS.pdf

Tree Table

Node	0		1		Total		Predicted Category	Parent Node	Primary Independent Variable				
	N	Percent	N	Percent	N	Percent			Variable	Sig.[a]	Chi-Square	df	Split Values
0	15123	64.8%	8213	35.2%	23336	100.0%	0						
1	11708	100.0%	0	.0%	11708	50.2%	0	0	tenure	.000	1986 8.982	4	<= 104.0
2	2243	96.1%	90	3.9%	2333	10.0%	0	0	tenure	.000	1986 8.982	4	(104.0, 126.0]
3	1008	45.2%	1221	54.8%	2229	9.6%	1	0	tenure	.000	1986 8.982	4	(126.0, 142.0]
4	164	7.3%	2090	92.7%	2254	9.7%	1	0	tenure	.000	1986 8.982	4	(142.0, 153.0]
5	0	.0%	4812	100.0%	4812	20.6%	1	0	tenure	.000	1986 8.982	4	> 153.0
6	1271	96.5%	46	3.5%	1317	5.6%	0	2	ln_avg_dist	.007	16.992	2	<= 1.596
7	239	91.6%	22	8.4%	261	1.1%	0	2	ln_avg_dist	.007	16.992	2	(1.596, 1.842]
8	733	97.1%	22	2.9%	755	3.2%	0	2	ln_avg_dist	.007	16.992	2	> 1.842
9	400	64.0%	225	36.0%	625	2.7%	0	3	phone	.000	123.630	1	Android
10	608	37.9%	996	62.1%	1604	6.9%	1	3	phone	.000	123.630	1	iPhone; <blank>

11	129	10.1%	1152	89.9%	1281	5.5%	1	4	Company X_black_user	.000	34.344	1	FALSE
12	35	3.6%	938	96.4%	973	4.2%	1	4	Company X_black_user	.000	34.344	1	TRUE
13	177	95.2%	9	4.8%	186	.8%	0	7	Company X_black_user	.001	10.810	1	FALSE
14	62	82.7%	13	17.3%	75	.3%	0	7	Company X_black_user	.001	10.810	1	TRUE
15	200	76.0%	63	24.0%	263	1.1%	0	9	ln_trips_in_first_30_days	.000	28.596	1	<= -6.908
16	200	55.2%	162	44.8%	362	1.6%	0	9	ln_trips_in_first_30_days	.000	28.596	1	> -6.908
17	467	47.2%	522	52.8%	989	4.2%	1	10	Company X_black_user	.000	95.074	1	FALSE
18	141	22.9%	474	77.1%	615	2.6%	1	10	Company X_black_user	.000	95.074	1	TRUE
19	100	13.0%	671	87.0%	771	3.3%	1	11	ln_surge_pct	.000	17.983	1	<= -6.908
20	29	5.7%	481	94.3%	510	2.2%	1	11	ln_surge_pct	.000	17.983	1	> -6.908
21	29	5.4%	505	94.6%	534	2.3%	1	12	ln_trips_in_first_30_days	.004	11.475	1	<= .694
22	6	1.4%	433	98.6%	439	1.9%	1	12	ln_trips_in_first_30_days	.004	11.475	1	> .694

Appendix B.3.3.1. Part 3 results of decision tree modeling for all data in Python

1-fold cross-validation results

Confusion metric

```
array([[6964,   412],
       [  182, 4966]])
```

Accuracy

0.953

10-fold cross-validation results

Accuracy: 0.957 (+/- 0.005)

Decision tree chart:

/EYankovskyCompany XChallengeAppendix/Part3_dtree_all_Python.svg

Decision tree model (split rules):

```
if ( tenure <= 130.5 ) {
if ( tenure <= 123.5 ) {
if ( tenure <= 120.5 ) {
return [[ 15070.      0.]]
} else {
if ( avg_dist <= 0.919999957085 ) {
return [[ 1.  2.]]
} else {
if ( weekday_pct <= 8.35000038147 ) {
return [[ 38.  10.]]
} else {
return [[ 331.   22.]]
}
}
}
} else {
if ( tenure <= 127.5 ) {
if ( kings_landing <= 0.5 ) {
if ( avg_rating_of_driver <=
3.45000004768 ) {
return [[ 20.  11.]]
} else {
return [[ 413.   87.]]
}
} else {
if ( weekday_pct <= 45.6500015259 ) {
return [[ 21.   1.]]
} else {
return [[ 53.  39.]]
}
}
} else {
if ( trips_in_first_30_days <= 0.5 )
{
if ( iphone <= 0.5 ) {
return [[ 55.   9.]]
} else {
return [[ 72.  42.]]
}
} else {
if ( Company X_black_user <= 0.5 ) {
return [[ 103.   69.]]
} else {
return [[ 41.  65.]]
}}}}
} else {
if ( tenure <= 144.5 ) {
if ( kings_landing <= 0.5 ) {
if ( android <= 0.5 ) {
if ( Company X_black_user <= 0.5 ) {
return [[ 347.  496.]]
} else {
return [[ 145.  518.]]
}
} else {
if ( tenure <= 139.5 ) {
return [[ 223.   93.]]
} else {
return [[ 72.  99.]]
}
}
} else {
if ( tenure <= 134.5 ) {
if ( trips_in_first_30_days <= 0.5 )
{
return [[ 30.  41.]]
} else {
return [[ 16.  79.]]
}
} else {
if ( trips_in_first_30_days <= 0.5 )
{
return [[  33.  181.]]
} else {
return [[  20.  368.]]
}
}
}
} else {
if ( tenure <= 147.5 ) {
if ( android <= 0.5 ) {
if ( Company X_black_user <= 0.5 ) {
return [[  38.  304.]]
} else {
return [[   8.  350.]]
}
} else {
if ( astapor <= 0.5 ) {
return [[  12.  114.]]
} else {
return [[ 17.  25.]]
}
}
} else {
if ( tenure <= 150.5 ) {
if ( trips_in_first_30_days <= 1.5 )
{
return [[  28.  410.]]
} else {
return [[   3.  619.]]
}} else {
return [[    0.  7956.]]
}}}}
```

Appendix B.3.3.2. Part 3 results of decision tree modeling for Astapor data in Python

1-fold cross-validation results

Classification matrix

```
array([[1057,   78],
       [  44, 1860]])
```

Accuracy

0.959

10-fold cross-validation results

Accuracy: 0.964 (+/- 0.016)

Decision tree chart:

/EYankovskyCompany XChallengeAppendix/Part3_dtree_astapor_Python.svg

Decision tree model (split rules):

```
if ( tenure <= 127.5 ) {
if ( tenure <= 123.5 ) {
if ( tenure <= 121.5 ) {
return [[ 2347.     0.]]
} else {
if ( avg_rating_by_driver <= 4.25 ) {
return [[ 0.  2.]]
} else {
if ( avg_rating_of_driver <= 3.5 ) {
return [[ 0.  1.]]
} else {
return [[ 32.   3.]]
}
}
}
} else {
if ( avg_surge <= 1.19000005722 ) {
if ( tenure <= 124.5 ) {
if ( surge_pct <= 12.1499996185 ) {
return [[ 17.   3.]]
} else {
return [[ 1.  2.]]
}
} else {
if ( avg_dist <= 3.82000017166 ) {
return [[ 13.  20.]]
} else {
return [[ 32.  25.]]
}
}
} else {
return [[ 16.   2.]]
}
}
} else {
if ( tenure <= 139.5 ) {
if ( tenure <= 132.5 ) {
if ( trips_in_first_30_days <= 0.5 )
{
if ( avg_dist <= 14.5749998093 ) {
return [[ 48.  38.]]
} else {
return [[ 0.  7.]]
}
} else {
if ( Company X_black_user <= 0.5 ) {
return [[ 20.  38.]]
} else {
return [[  5.  40.]]
}}
} else {
if ( iphone <= 0.5 ) {
if ( tenure <= 138.5 ) {
return [[ 27.  54.]]
} else {
return [[  1.  15.]]
}
} else {
if ( Company X_black_user <= 0.5 ) {
return [[  31.  137.]]
} else {
return [[   9.  135.]]
}}}}
 else {
if ( tenure <= 145.5 ) {
if ( trips_in_first_30_days <= 0.5 )
{
if ( tenure <= 140.5 ) {
return [[  0.  27.]]
} else {
return [[  22.  145.]]
}
} else {
if ( avg_dist <= 0.745000004768 ) {
return [[ 1.  0.]]
} else {
return [[   5.  290.]]
}
}
} else {
if ( tenure <= 148.5 ) {
if ( avg_dist <= 1.31500005722 ) {
return [[  1.  12.]]
} else {
return [[   3.  376.]]
}
} else {
if ( tenure <= 150.5 ) {
return [[   1.  282.]]
} else {
return [[    0.  2805.]]
}}}}}
```

Appendix B.3.3.2. Part 3 results of decision tree modeling for King's Landing data in Python

1-fold cross-validation results

Classification matrix

```
array([[1023,   99],
       [  19, 1898]])
```

Accuracy

0.961

10-fold cross-validation results

Accuracy: 0.964 (+/- 0.016)

Decision tree chart:

/EYankovskyCompany XChallengeAppendix/Part3_dtree_kings_landing_Python.svg

Decision tree model (split rules):

```
if ( tenure <= 125.5 ) {
if ( tenure <= 123.5 ) {
if ( tenure <= 121.5 ) {
return [[ 2356.     0.]]
} else {
if ( avg_dist <= 0.47499999404 ) {
return [[ 0.  1.]]
} else {
if ( avg_rating_by_driver <=
4.15000009537 ) {
return [[ 0.  1.]]
} else {
return [[ 36.   3.]]
}
}
}
} else {
if ( avg_dist <= 4.125 ) {
if ( avg_rating_by_driver <=
4.94999980927 ) {
return [[ 8.  3.]]
} else {
return [[ 7.  9.]]
}
} else {
if ( avg_dist <= 8.42000007629 ) {
return [[ 11.   0.]]
} else {
return [[ 10.   7.]]
}
}
}
} else {
if ( tenure <= 134.5 ) {
if ( android <= 0.5 ) {
if ( Company X_black_user <= 0.5 ) {
if ( tenure <= 132.5 ) {
return [[ 58.  73.]]
} else {
return [[ 15.  47.]]
}
} else {
if ( tenure <= 130.5 ) {
return [[ 14.  35.]]
} else {
return [[  5.  60.]]
}
}
} else {
if ( trips_in_first_30_days <= 4.5 )
{
if ( avg_dist <= 4.66499996185 ) {
return [[ 17.  24.]]
} else {
return [[ 36.  17.]]
}
} else {
return [[ 0.  6.]]
}
}
} else {
if ( tenure <= 141.5 ) {
if ( trips_in_first_30_days <= 0.5 )
{
if ( tenure <= 137.5 ) {
return [[ 19.  53.]]
} else {
return [[  14.  114.]]
}
} else {
if ( weekday_pct <= 41.0999984741 ) {
return [[  8.  43.]]
} else {
return [[  10.  204.]]
}
}
} else {
if ( tenure <= 146.5 ) {
if ( trips_in_first_30_days <= 0.5 )
{
return [[  15.  153.]]
} else {
return [[   3.  238.]]
}
} else {
if ( avg_dist <= 27.9200000763 ) {
return [[  2.00000000e+00
3.33400000e+03]]
} else {
return [[  1.  21.]]
}
}
}
}
}
```

Appendix B.3.3.3. Part 3 results of decision tree modeling for Winterfell data in Python

1-fold cross-validation results

Classification matrix

```
array([[4326,  144],
       [ 196, 2335]])
```

Accuracy

0.951

10-fold cross-validation results

Accuracy: 0.956 (+/- 0.007)

Decision tree chart:

/EYankovskyCompany XChallengeAppendix/Part3_dtree_winterfell_Python.svg

Decision tree model (split rules):

```
if ( tenure <= 133.5 ) {
if ( tenure <= 126.5 ) {
if ( tenure <= 122.5 ) {
if ( tenure <= 120.5 ) {
return [[ 9407.     0.]]
} else {
if ( weekday_pct <= 6.25 ) {
return [[ 17.   5.]]
} else {
return [[ 127.    6.]]
}
}
} else {
if ( trips_in_first_30_days <= 3.5 ) {
if ( tenure <= 124.5 ) {
return [[ 91.  23.]]
} else {
return [[ 134.   14.]]
}
} else {
if ( weekday_pct <= 91.3000030518 ) {
return [[ 37.  12.]]
} else {
return [[ 1.  3.]]
}
}
}
} else {
if ( Company X_black_user <= 0.5 ) {
if ( iphone <= 0.5 ) {
if ( surge_pct <= 18.3500003815 ) {
return [[ 106.   20.]]
} else {
return [[ 17.  10.]]
}
} else {
if ( avg_surge <= 1.00499999523 ) {
return [[ 106.   74.]]
} else {
return [[ 63.  16.]]
}
}
} else {
if ( android <= 0.5 ) {
if ( trips_in_first_30_days <= 1.5 ) {
return [[ 35.  45.]]
} else {
return [[ 14.  52.]]
}
} else {
if ( trips_in_first_30_days <= 0.5 ) {
return [[ 23.   6.]]
} else {
return [[ 21.  21.]]
}
}
}
}
}
} else {
if ( tenure <= 144.5 ) {
if ( android <= 0.5 ) {
if ( Company X_black_user <= 0.5 ) {
if ( weekday_pct <= 91.0999984741 ) {
return [[ 134.  282.]]
} else {
return [[ 62.  57.]]
}
} else {
if ( weekday_pct <= 98.4499969482 ) {
return [[  32.  240.]]
} else {
return [[ 25.  58.]]
}
}
} else {
if ( tenure <= 138.5 ) {
if ( trips_in_first_30_days <= 0.5 ) {
return [[ 41.  10.]]
} else {
return [[ 41.  27.]]
}
} else {
if ( trips_in_first_30_days <= 2.5 ) {
return [[ 49.  49.]]
} else {
return [[ 12.  38.]]
}
}
}
} else {
if ( tenure <= 150.5 ) {
if ( trips_in_first_30_days <= 1.5 ) {
if ( android <= 0.5 ) {
return [[  27.  280.]]
} else {
return [[ 16.  67.]]
}
} else {
if ( tenure <= 146.5 ) {
return [[  10.  102.]]
} else {
return [[   5.  365.]]
}
}
} else {
return [[    0.  3800.]]
}}}
```