Response For Takehome Challenge

Yi Zhang
yi.zhang.pipal@gmail.com 412-951-7529

PAR1 - sql

Q1.

```
SELECT
PERCENTILE_CONT(0.9)
WITHIN GROUP ( ORDER BY ABS(actual_eta – predicted_eta)  )
OVER (PARTITION BY [city_id])
FROM (
SELECT actual_eta, predicted_eta , city_id
FROM trips
WHERE city_id  IN (SELECT city_id FROM cities WHERE citi_name IN [‘Qarth’, 'Meereen'])
AND DATEDIFF(day, request_at, getdate()) between 0 and 30 )
```

Q2.

Assumption: I assume the question is asking the ratio based on sign up city, instead of trip city, and the first trip can take place anywhere which is not necessary in the sign up city

```
-- Success signup :
WITH
A AS (
SELECT rider_id, city_id, _ts, CAST(_ts AS DATE) AS date FROM events
WHERE city_id IN
SELECT city_id FROM cities WHERE citi_name IN [‘Qarth’, 'Meereen']
AND CAST(_ts AS DATE) >=  ‘2016-01-01’
AND CAST(_ts AS DATE) <=  ‘2016-01-07’
AND event_name = ‘sign_up_success’),

-- Complete trips:
B AS (
SELECT MIN(request_at) AS time, client_id
FROM trips
WHERE status = ‘completed’),

-- Time diff
C AS (
SELECT A.city_id AS city_id, A.date  AS date, COUNT(*) AS success FROM B
JOIN A ON A.rider_id = B.client_id
WHERE DATEDIFF(hour, B.time, A._ts ) <= 160
GROUP BY A.date, A.city_id),

--sign on counts
D AS (
SELECT COUNT(*) AS count, date, city_id FROM A
GROUP BY date, city_id)

-- getRatio:
SELECT C.success / D.count::float AS successRatio FROM C
JOIN D ON (C.city_id = D.city_id AND D.date = C.date)
```

Part 2 - Data analysis

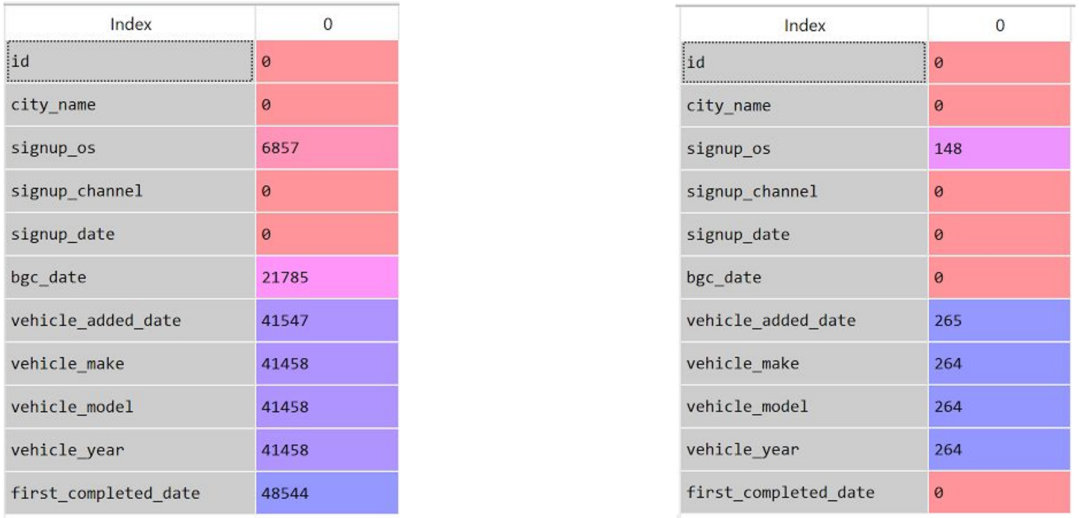
1. Because of personal laptop problem, i have to convert xml file to csv to work.
2. I assume the NA in raw data is the same as blank cell

Q1.

Before any of the action is taken on the data, we first need to understand the raw data, this process includes:

1. What is data type for each features
2. Any missing data?
3. What are the statistics of features (column)
4. Any correlation between features

After analyzed the data, there are 10 columns (besides the id column) and 54681 instances and huge portion of missing data. Among these data, 6137 drivers did their first trip, which is 11.29%.



Index	0
id	0
city_name	0
signup_os	6857
signup_channel	0
signup_date	0
bgc_date	21785
vehicle_added_date	41547
vehicle_make	41458
vehicle_model	41458
vehicle_year	41458
first_completed_date	48544

Index	0
id	0
city_name	0
signup_os	148
signup_channel	0
signup_date	0
bgc_date	0
vehicle_added_date	265
vehicle_make	264
vehicle_model	264
vehicle_year	264
first_completed_date	0

Table 1. Missing data counts in raw data. Left: missing data in full data set; right: missing data in the sub dataset in which driver made trips

Table 1 indicates there are lots of missing data and the missing data are not completely random. For example, vehicle_added_date, vehicle_make, vehicle_model, and vehicle_year are highly correlated in the missing data perspective, and their missing pattern highly correlated to whether the driver made a trip, since the missing ratio in original full data set is much higher than in the subset (41458/54681 vs. 264/6137 which indicates 4% missing rate when trips are made vs. 85% missing rate when trips are never made). Signup_cannel is not that significant as vehicle columns, but also carries indication related to whether a trip is made.

Based on this information, following features could be considered:

a. City_name: categorical
b. Signup_os: categorical

c. Signup_channel: categorical
d. vehicle _info_offered: boolean
e. Vehicle_date: categorical
f. Vehicle_made: categorical
g. Vehicle_model: categorical
h. Vehicle_year: categorical

Next step, i would like to understand whether absolute date carries information or only the date difference carries information.

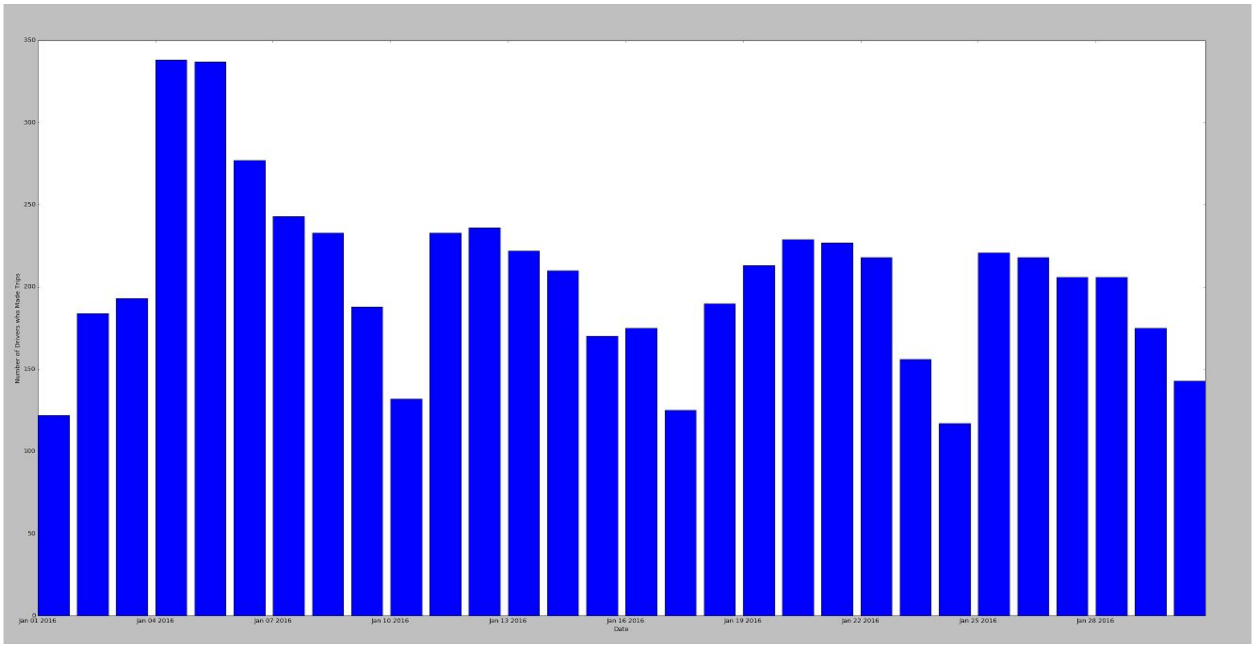


Figure 1. The number of drivers who made their trips on each sign on day

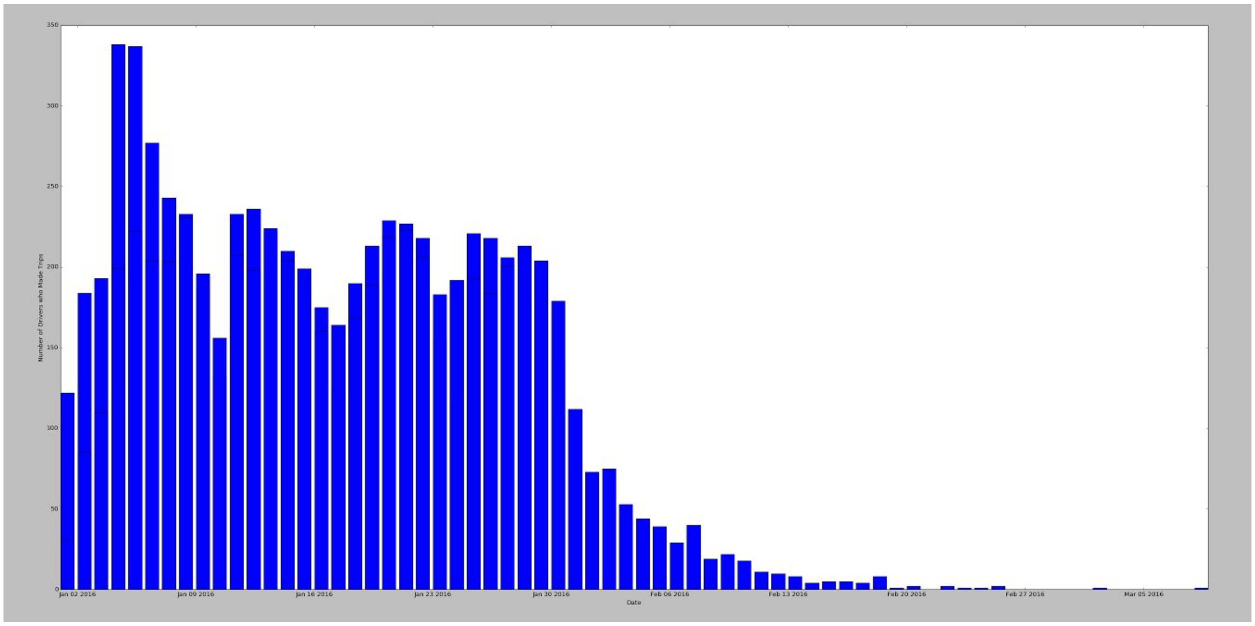


Figure 2. The number of drivers who made their trips on each back ground check day

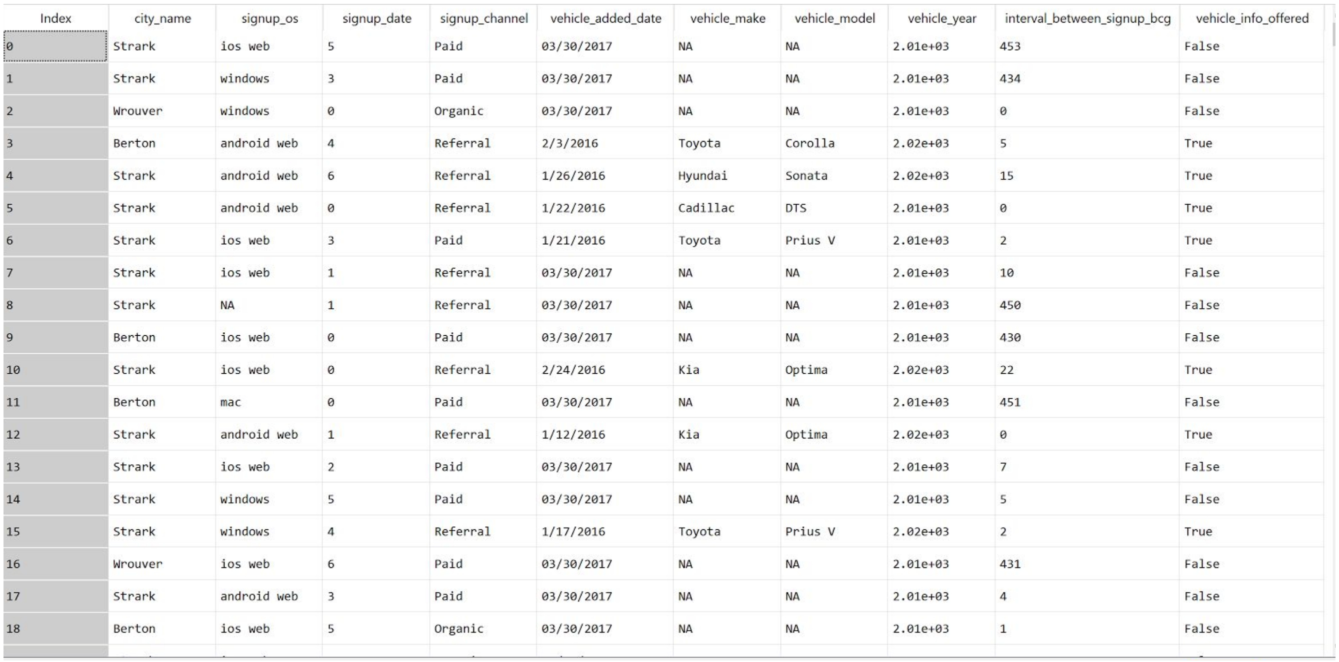
Interestingly, absolute date carries some information. One can see clear pattern on Figure 1 which indicates weekly pattern.
Therefore, following features are also used:

i. weekday_sign_on: categorical
j. Days_between_sign_on_and_bgc: numerical

Obviously, there are correlations among features, and we have mixture of feature types, while most of which are categorical data. Therefore, tree-based classifier/predictor is a natural choice. Another important factor which need to be considered is the imbalanced data. Since only 11.29% of the driver did their first trip, i am facing a problem of unbalanced data. These prior probability needs to be carefully considered:

1. This is the nature of the data, which one cannot and should not manually adjust it to balanced data
2. Unbalanced prior will have effect on the final decision boundary. Therefore, besides accuracy, different types of error have to be analyzed separately. If in practical there is a preference to minimize certain kind of error, weights need to be used on loss function

Q2.
Based on the analysis on Q1, I first generated the feature as following (sample):



Index	city_name	signup_os	signup_date	signup_channel	vehicle_added_date	vehicle_make	vehicle_model	vehicle_year	interval_between_signup_bcg	vehicle_info_offered
0	Strark	ios web	5	Paid	03/30/2017	NA	NA	2.01e+03	453	False
1	Strark	windows	3	Paid	03/30/2017	NA	NA	2.01e+03	434	False
2	Wrouver	windows	0	Organic	03/30/2017	NA	NA	2.01e+03	0	False
3	Berton	android web	4	Referral	2/3/2016	Toyota	Corolla	2.02e+03	5	True
4	Strark	android web	6	Referral	1/26/2016	Hyundai	Sonata	2.02e+03	15	True
5	Strark	android web	0	Referral	1/22/2016	Cadillac	DTS	2.01e+03	0	True
6	Strark	ios web	3	Paid	1/21/2016	Toyota	Prius V	2.01e+03	2	True
7	Strark	ios web	1	Referral	03/30/2017	NA	NA	2.01e+03	10	False
8	Strark	NA	1	Referral	03/30/2017	NA	NA	2.01e+03	450	False
9	Berton	ios web	0	Paid	03/30/2017	NA	NA	2.01e+03	430	False
10	Strark	ios web	0	Referral	2/24/2016	Kia	Optima	2.02e+03	22	True
11	Berton	mac	0	Paid	03/30/2017	NA	NA	2.01e+03	451	False
12	Strark	android web	1	Referral	1/12/2016	Kia	Optima	2.02e+03	0	True
13	Strark	ios web	2	Paid	03/30/2017	NA	NA	2.01e+03	7	False
14	Strark	windows	5	Paid	03/30/2017	NA	NA	2.01e+03	5	False
15	Strark	windows	4	Referral	1/17/2016	Toyota	Prius V	2.02e+03	2	True
16	Wrouver	ios web	6	Paid	03/30/2017	NA	NA	2.01e+03	431	False
17	Strark	android web	3	Paid	03/30/2017	NA	NA	2.01e+03	4	False
18	Berton	ios web	5	Organic	03/30/2017	NA	NA	2.01e+03	1	False

Several place need to be noted:

1. All missing date are replaced by an arbitrary late future date (I used today's date). This works for current data set since it contains long ago historic data. In practical, need to use another way to fill the nan
2. Fill missing date with an arbitrary date could cause trouble if different classifiers are using. Since such missing data imputation method will cause the chance of variance on that feature, which will huge affect the discrimination power on distance based methods. However, since i am going to use tree-based method, such effect is not that huge
3. Because of implementation, i have to do one-hot encoding, although tree-structure classifier by definition can directly working with categorical data. The encoding enlarge the feature space from 10 to 511, and some features have more different values than others, which may bring some problem to the classifier since random forest has a bias toward feature with more unique values.
4. With encoding, the correlation between features get stronger, which will affect the classifier as well
5. Hyper parameter is determined based on a simple grid search. With complex model or large amount data, such simple grid search can be very CPU and time consumed and therefore more optimized search methods should be considered.

Random Forest Results:

Hyper parameter: 50 tree used

Feature importance is shown as below:

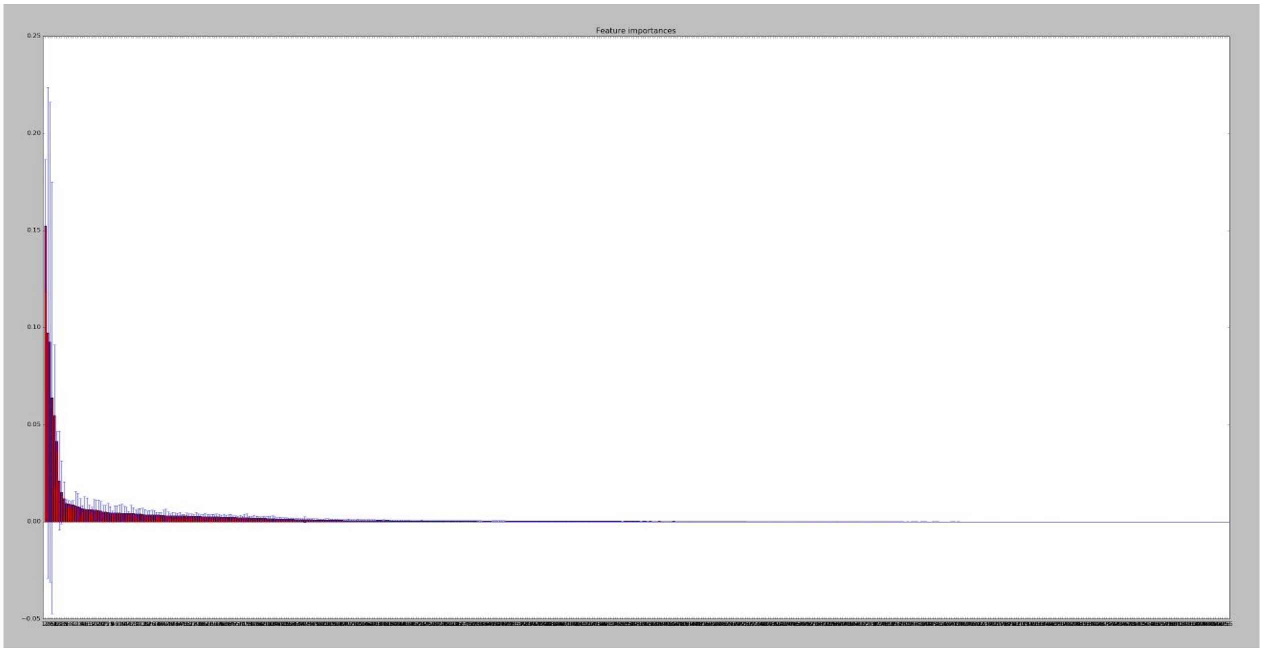
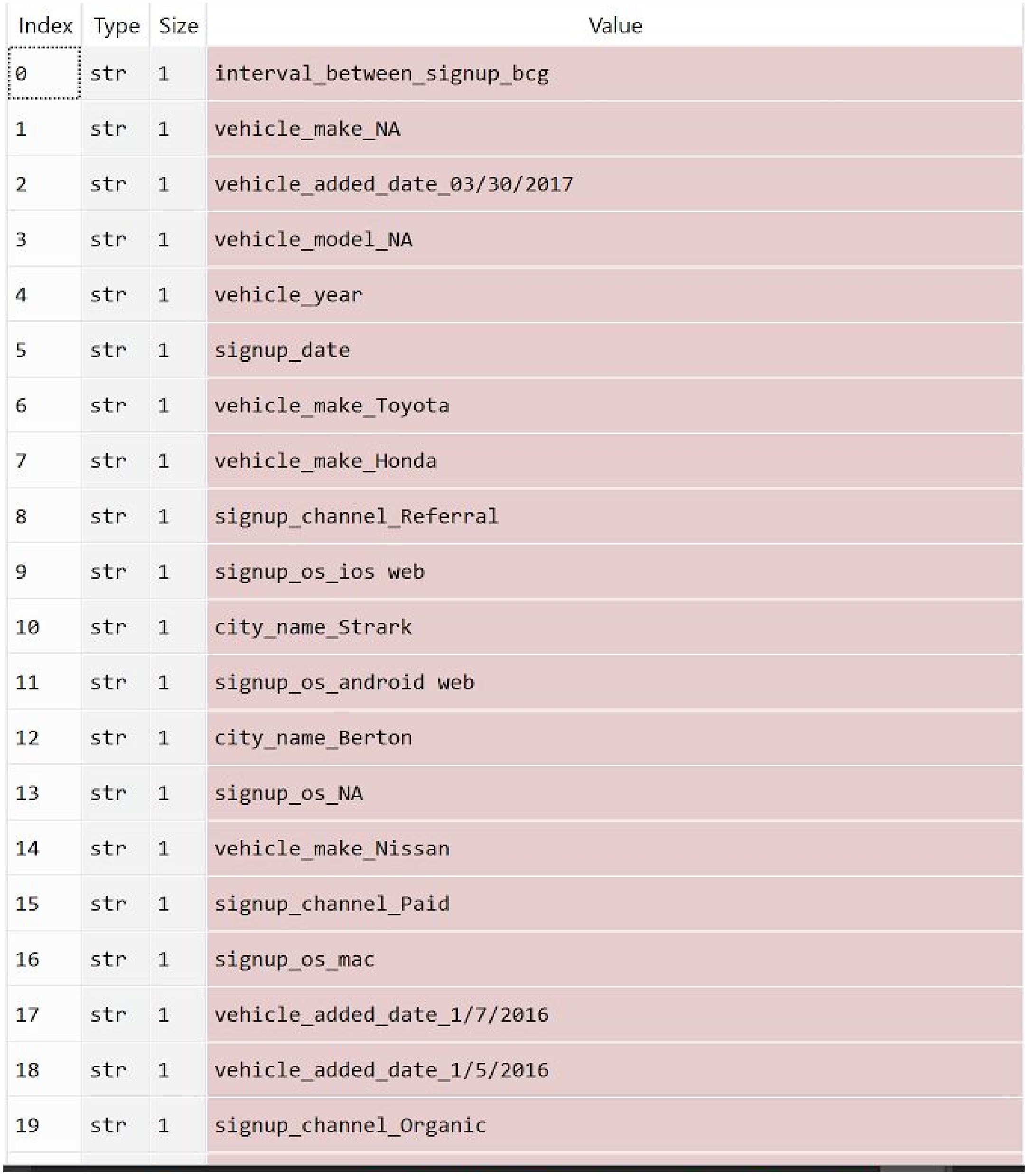


Figure 3. Feature importance based on random forest

Figure 3 shows the feature importance from random forest. Since after encoding there are 511 features, therefore a long tail is shown in the bar plot. If i sample the first 20 important features, they are:



Index	Type	Size	Value
0	str	1	interval_between_signup_bcg
1	str	1	vehicle_make_NA
2	str	1	vehicle_added_date_03/30/2017
3	str	1	vehicle_model_NA
4	str	1	vehicle_year
5	str	1	signup_date
6	str	1	vehicle_make_Toyota
7	str	1	vehicle_make_Honda
8	str	1	signup_channel_Referral
9	str	1	signup_os_ios web
10	str	1	city_name_Strark
11	str	1	signup_os_android web
12	str	1	city_name_Berton
13	str	1	signup_os_NA
14	str	1	vehicle_make_Nissan
15	str	1	signup_channel_Paid
16	str	1	signup_os_mac
17	str	1	vehicle_added_date_1/7/2016
18	str	1	vehicle_added_date_1/5/2016
19	str	1	signup_channel_Organic

Table 2. Top 20 features based on random forest

As expected, the time interval between sign up and background check is a very important indicator, and whether a vehicle information is offer is important as well. It worth to mention that since these features are based on encoding and the original features contains large portion of categorical data, variance based feature selection methods are not good choice (PCA for example). Wrapper based methods or embedding based methods are good choices.
Interestingly, the testing accuracy is
Accuracy: 0.996151
False Alarm Rate: 0.000277
Miss of Detection: 0.032203
Area under the ROC curve : 0.953404

in the test set, which is rarely happen in real practice (even higher than training). I guess is because the data set is small and synthetic. Obviously these metrics reflects the imbalance problem, which I will discuss it with logistic regression

Since the one-hot encoding is already performed on the data set, I am going to try logistic regression as well.
One good thing for logistic regression is it not required normalization on features, therefore our imputation won't cause too much trouble
In the test stage logistic regression perform worse than random forest:
Accuracy: 0.936658
False Alarm Rate: 0.041294
Miss of Detection: 0.236506
Area under the ROC curve : 0.965891

ROC curve is shown as following:

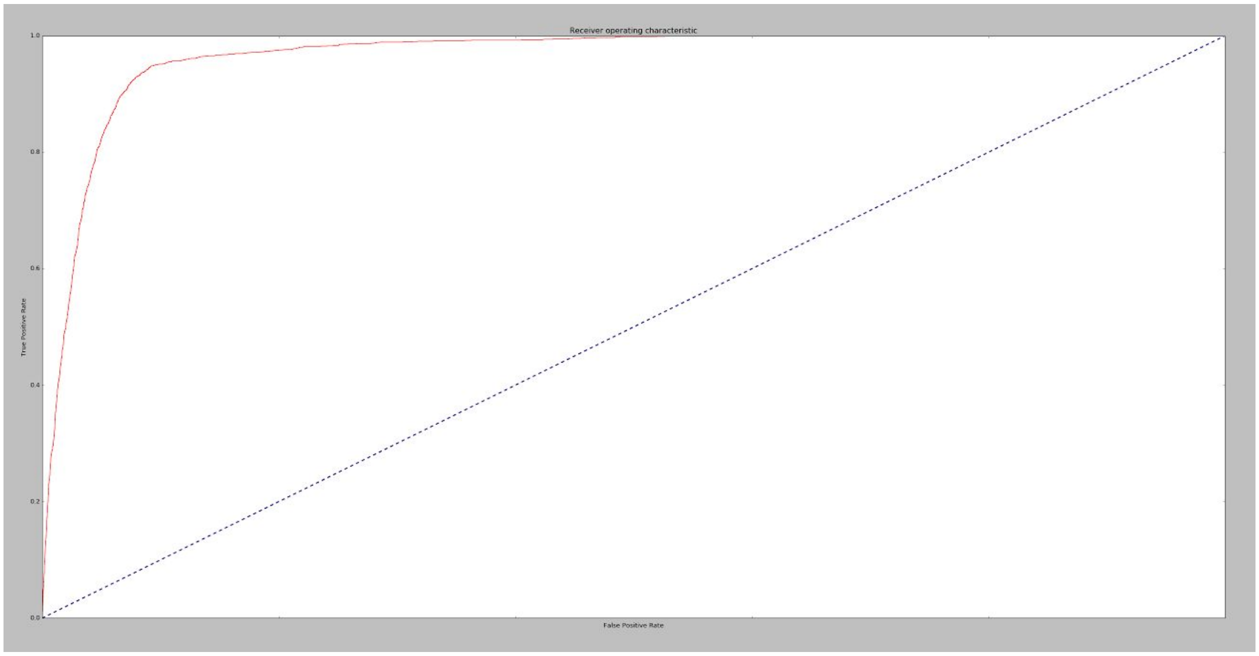


Figure 4. ROC Curve for Logistic Regression

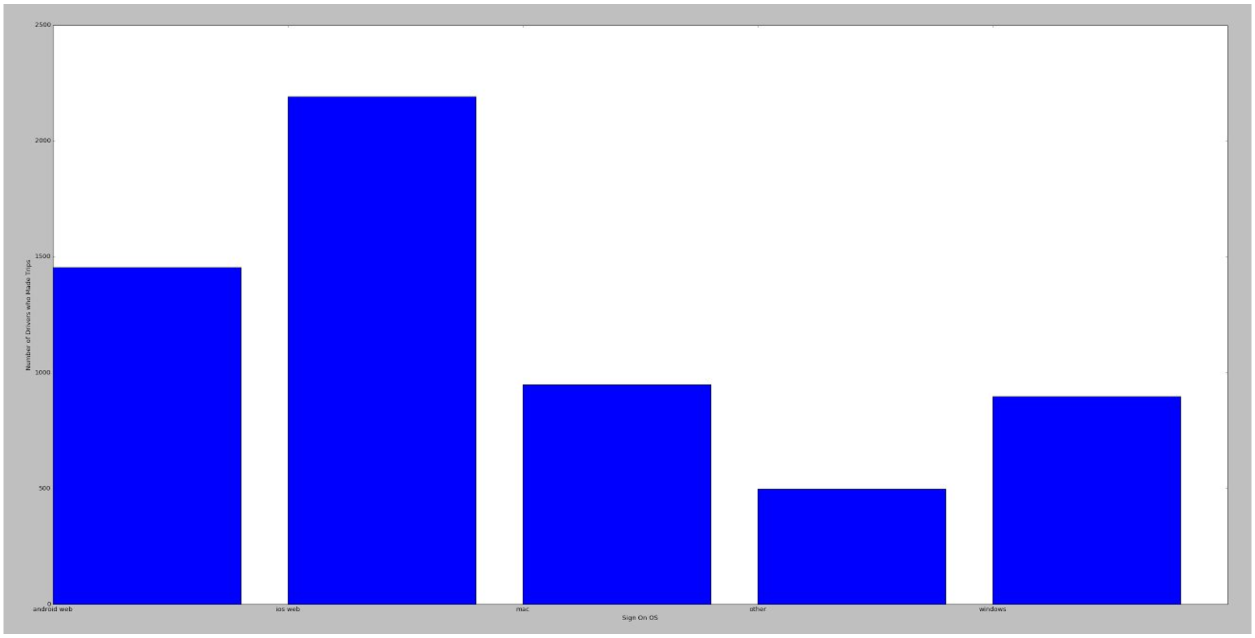
Based on this results, I can clearly see the imbalanced data problem. A low false alarm rate is achieved while the detection rate is sacrificed. If we want to aggressively detecting potential drivers, weights should be added to the loss function
If the weight of logistic regression is adjusted to balanced we got
Accuracy: 0.915600
False Alarm Rate: 0.087774
Miss of Detection: 0.057900
Area under the ROC curve : 0.965690
The overall accuracy decreased, however the detection rate got improved significantly.

To summarize, in current data set, random forest performs better the logistic regression. Besides these two methods, there are many ways to perform such driver prediction. However, since we have correlations between feature and mixture of feature types, we are constrained to many solutions leveraging such assumptions. We also need to consider better way to impute missing data if a distance based method is using, for example K Nearest Neighbors.

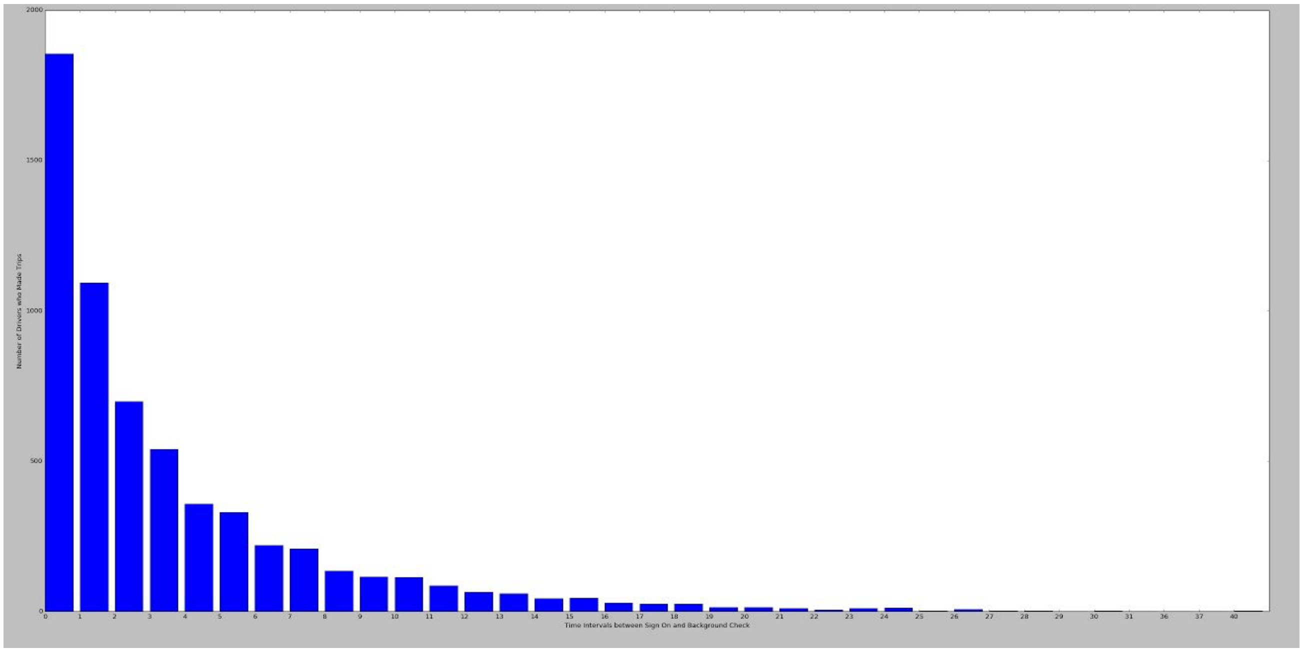
Q3.

We can first exam some correlations:

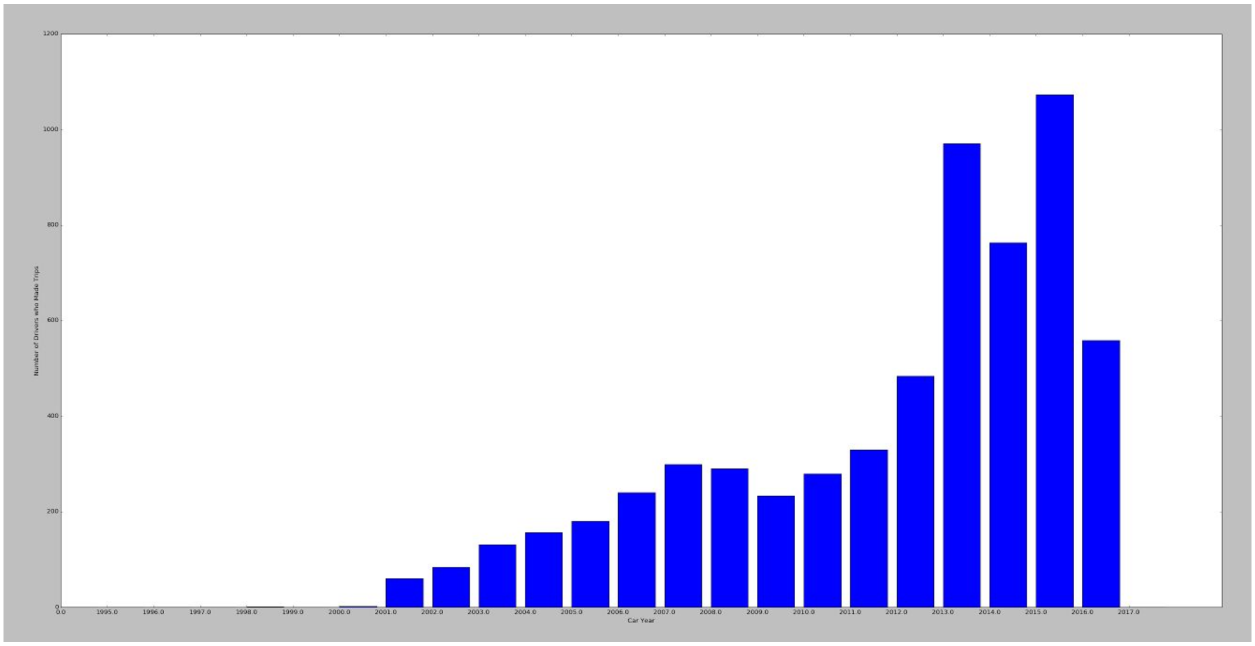
a. Sign on OS



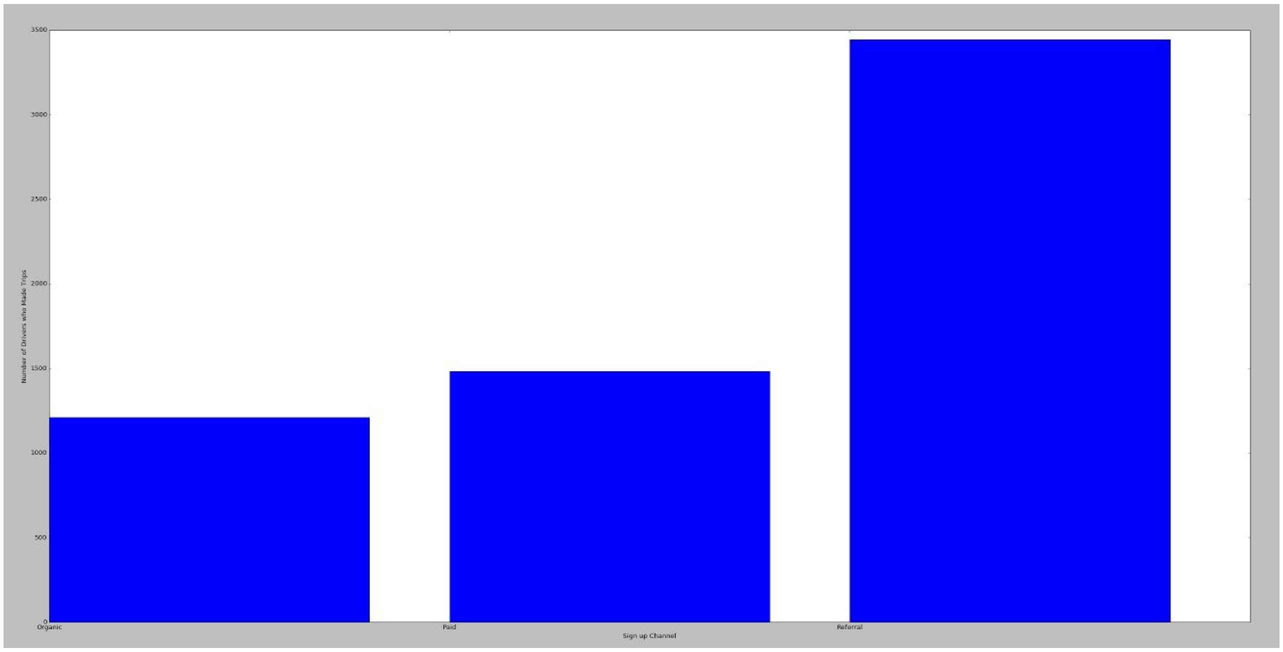
b. Time interval between sign on date and background check date



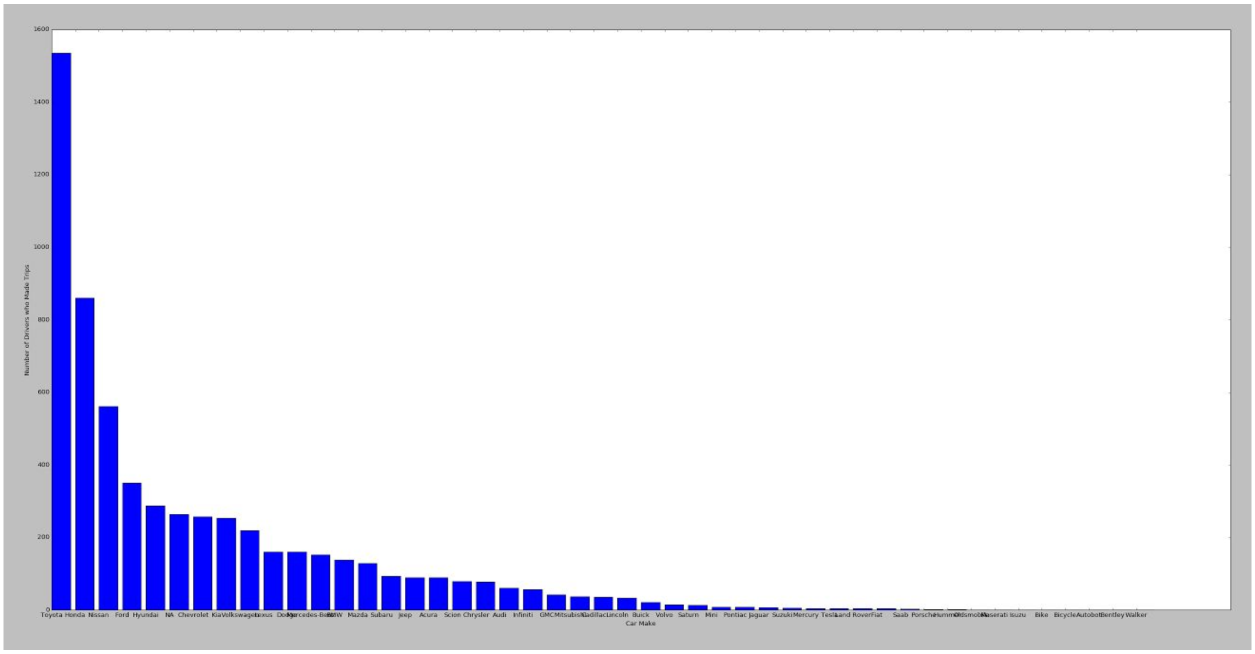
c. Year of Car



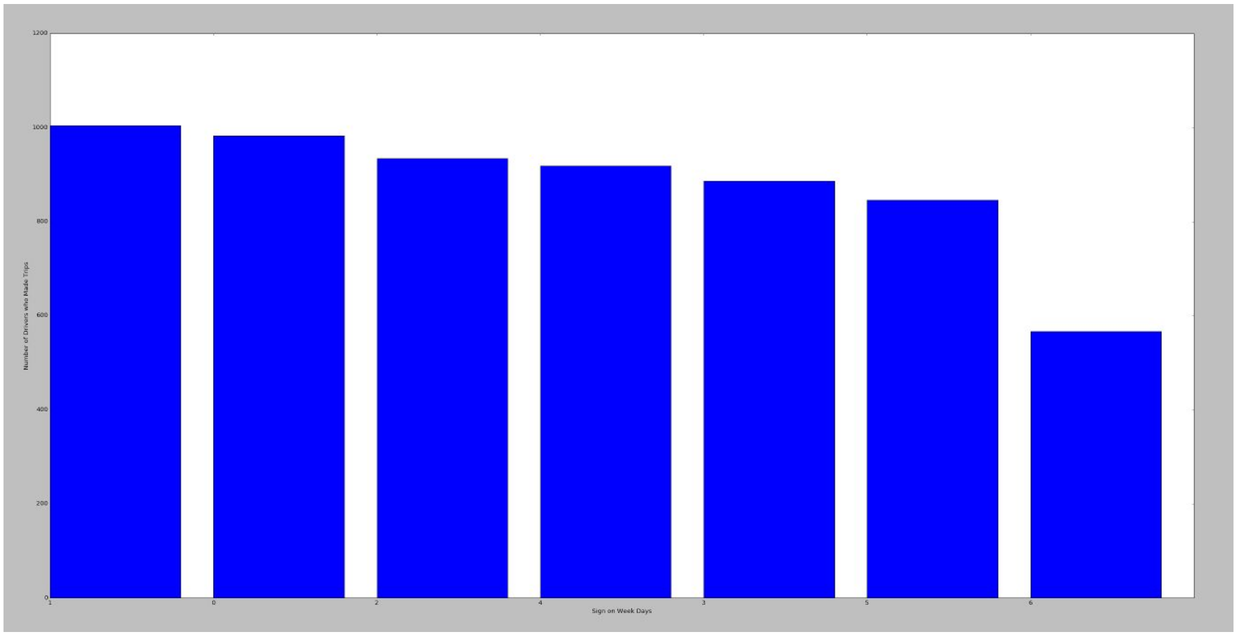
d. Sign up Channel



e. Car make



f. Sign on Week Day



To summarize the plots above:

People who

1. Got background check quickly done after signon
2. Sign on through mobile
3. Sign on through referring
4. Drive newer cars
5. Sign on on weekdays instead of weekends
6. Drive japanese economic cars

Have higher chance to be real drivers.

So based on these information Uber should consider:

1. Encourage referring
2. Focus on people sign on during weekdays or put special promotions on weekdays
3. Encourage people signed on to get background check done as soon as possible
4. Focus on people driving newer cars or considering help people get newer cars, as special finance plan/leasing plan
5. Target certain dealership (lower end economic cars for example) for special discount on Uber drivers