Data_challenge

Bowei.Zhang

September 4, 2016

Introduction

This is part of data interview challenge that I did for Uber:

Uber's Driver team is interested in predicting which driver signups are most likely to start driving. To help explore this question, we have provided a sample dataset of a cohort of driver signups in January 2015. The data was pulled a few months after they signed up to include the result of whether they actually completed their first trip. It also includes several pieces of background information gather about the driver and their car.
We would like you to use this data set to help understand what factors are best at predicting whether a signup will start to drive, and offer suggestions to operationalize those insights to help Uber.

Data description:
id: driver_id
city_id : city_id this user signed up in
signup_os : signup device of the user ("android", "ios", "website", "other")
signup_channel : what channel did the driver sign up from ("offline", "paid", "organic", "referral")
signup_timestamp : timestamp of account creation; local time in the form 'YYYYMMDD'
bgc_date : date of background check consent; in the form 'YYYYMMDD'
vehicle_added_date : date when driver's vehicle information was uploaded; in the form 'YYYYMMDD'
first_trip_date : date of the first trip as a driver; in the form 'YYYYMMDD'
vehicle_make: make of vehicle uploaded (i.e. Honda, Ford, Kia)
vehicle_model: model of vehicle uploaded (i.e. Accord, Prius, 350z)
vehicle_year: year that the car was made; in the form 'YYYY'

Required Packages:

```
require(ggplot2)  #For plots
require(cowplot)  #For better arrange plots
require(caTools)  #For stratify sampling
require(glmulti)  #For All Subset model selection
require(ROCR)     #For generating ROC curve to evaluate the binary classifier
```

Exploratory Analysis

Input sample dataset

```
# read sample data, and change all blanks into 'NA'
signup <- read.csv("C:/Users/bowei.zhang/Desktop/ME/Career/Analytics_Example/Data_Challenge/ds_challenge_v2_1_data.csv", header=T, na.strings=c("","NA"))

# Look at the size of the data and data type of each variables
str(signup)
```

```
## 'data.frame':    54681 obs. of  11 variables:
##  $ id                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ city_name           : Factor w/ 3 levels "Berton","Strark",..: 2 2 3 1 2 2 2 2 2 1 ...
##  $ signup_os           : Factor w/ 5 levels "android web",..: 2 5 5 1 1 1 2 2 NA 2 ...
##  $ signup_channel      : Factor w/ 3 levels "Organic","Paid",..: 2 2 1 3 3 3 2 3 3 2 ...
##  $ signup_date         : Factor w/ 30 levels "1/1/16","1/10/16",..: 12 14 3 22 2 10 6 19 26 1
8 ...
##  $ bgc_date            : Factor w/ 74 levels "1/1/16","1/10/16",..: NA NA 3 54 18 10 8 56 NA
 NA ...
##  $ vehicle_added_date  : Factor w/ 78 levels "1/1/16","1/10/16",..: NA NA NA 54 19 15 14 NA N
A NA ...
##  $ vehicle_make        : Factor w/ 46 levels "Acura","Audi",..: NA NA NA 43 18 9 43 NA NA NA
 ...
##  $ vehicle_model       : Factor w/ 368 levels "200","3-Sep",..: NA NA NA 73 294 91 237 NA NA
 NA ...
##  $ vehicle_year        : int  NA NA NA 2016 2016 2006 2014 NA NA NA ...
##  $ first_completed_date: Factor w/ 57 levels "1/10/16","1/11/16",..: NA NA NA 51 NA NA 14 NA
 NA NA ...
```

```
# Take a look at typical value of each variables
summary(signup)
```

```
##        id          city_name          signup_os      signup_channel
##  Min.   :    1   Berton :20117   android web:14944   Organic :13427
##  1st Qu.:13671   Strark :29557   ios web    :16632   Paid    :23938
##  Median :27341   Wrouver: 5007   mac        : 5824   Referral:17316
##  Mean   :27341                   other      : 3648
##  3rd Qu.:41011                   windows    : 6776
##  Max.   :54681                   NA's       : 6857
##
##   signup_date       bgc_date     vehicle_added_date  vehicle_make
##  1/5/16 : 2489   1/29/16: 1125   1/26/16:  377      Toyota : 3219
##  1/4/16 : 2460   1/28/16: 1103   1/28/16:  370      Honda  : 1845
##  1/1/16 : 2282   1/27/16: 1071   1/22/16:  336      Nissan : 1311
##  1/6/16 : 2207   1/30/16: 1071   1/29/16:  331      Ford   :  778
##  1/7/16 : 2078   1/22/16: 1028   1/24/16:  328      Hyundai:  677
##  1/21/16: 2024   (Other):27498   (Other):11392      (Other): 5393
##  (Other):41141   NA's   :21785   NA's   :41547      NA's   :41458
##  vehicle_model    vehicle_year   first_completed_date
##  Civic  :  689   Min.   :   0    1/23/16:  257
##  Corolla:  688   1st Qu.:2008    1/30/16:  243
##  Camry  :  683   Median :2013    1/29/16:  218
##  Accord :  595   Mean   :2011    1/22/16:  215
##  Prius V:  522   3rd Qu.:2015    1/26/16:  209
##  (Other):10046   Max.   :2017    (Other): 4995
##  NA's   :41458   NA's   :41458   NA's   :48544
```

Dependent Variable

From the summary above we could see the whole dataset contains 54681 records with 11 vaiables, `first_completed_date` is the dependent variable that we want to predict, we will treat NA in `first_completed_date` as the driver who signup but do not have first trip, so we will make a new binary variable call `first_trip` with value `C` as a driver completed a first trip with Uber and value `N` as a driver does not complete a first trip with Uber

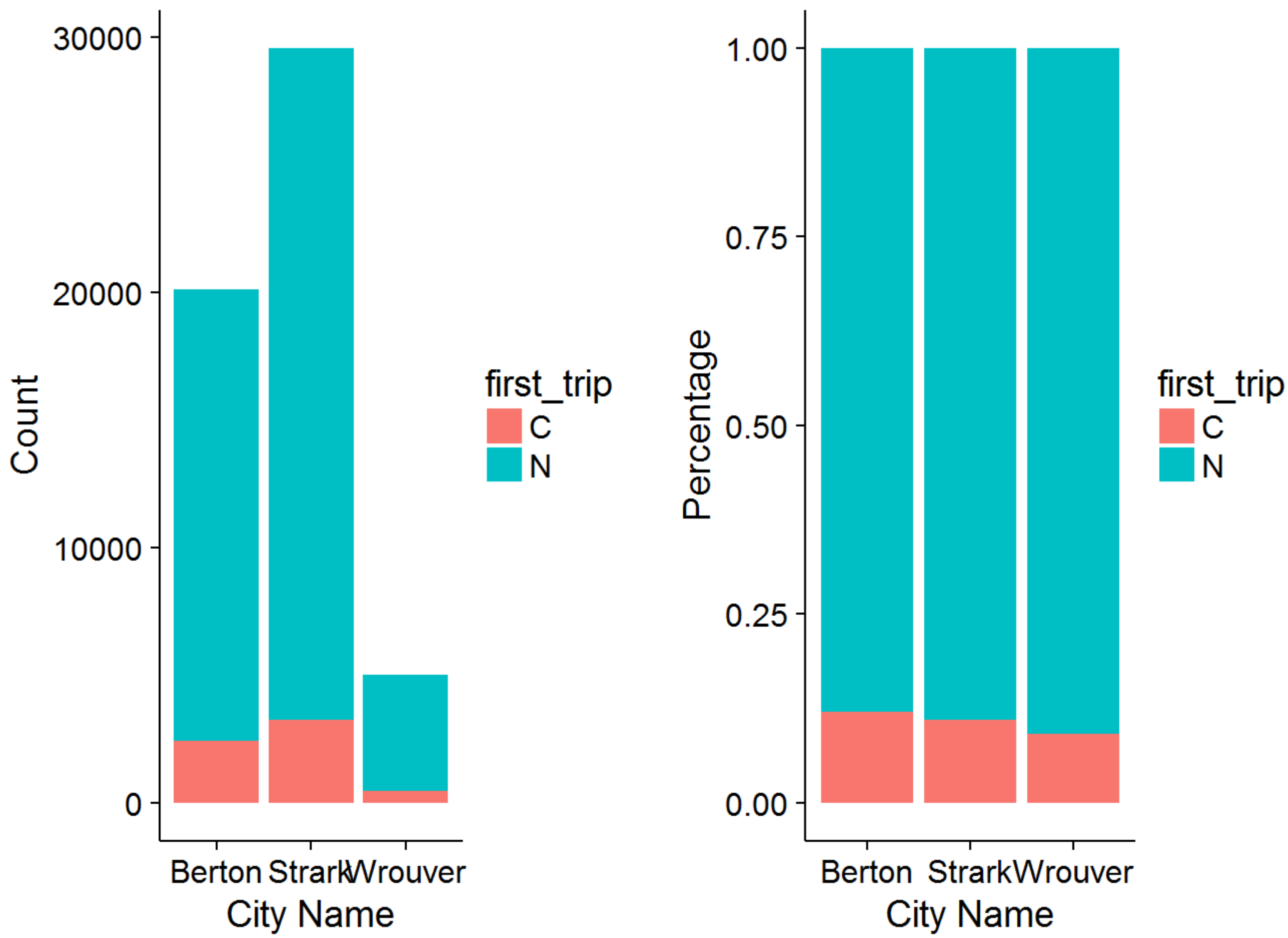
```
signup$first_trip <- as.factor(ifelse(is.na(signup$first_completed_date),"N","C"))
```

Now let's explore different independent variables:
City Name

```
#Plot city_name to see first trip rate
city_name1 <- ggplot()+ geom_bar(data =signup, aes(x = city_name, fill = first_trip)) +labs(x =
"City Name",y = "Count")


city_name2 <- ggplot(data =signup, aes(x = city_name, fill = first_trip)) +
    geom_bar(aes(fill = first_trip), position = 'fill')+labs(x = "City Name",y = "Percentage")

plot_grid(city_name1, city_name2, ncol = 2, nrow = 1)
```



From the plot we cuold see that **Strark** has the highest signups but **Berton** has the highest percetage first trip rate.

Signup Channel

```
#Plot signup_channel to see first trip rate
signup_channel1 <- ggplot(data =signup, aes(x = signup_channel, fill = first_trip))+ geom_bar()
+labs(x = "Signup Channel",y = "Count")


signup_channel2 <- ggplot(data =signup, aes(x = signup_channel, fill = first_trip)) +
    geom_bar(aes(fill = first_trip), position = 'fill')+labs(x = "Signup Channel",y = "Percentag
e")

plot_grid(signup_channel1, signup_channel2, ncol = 2, nrow = 1)
```

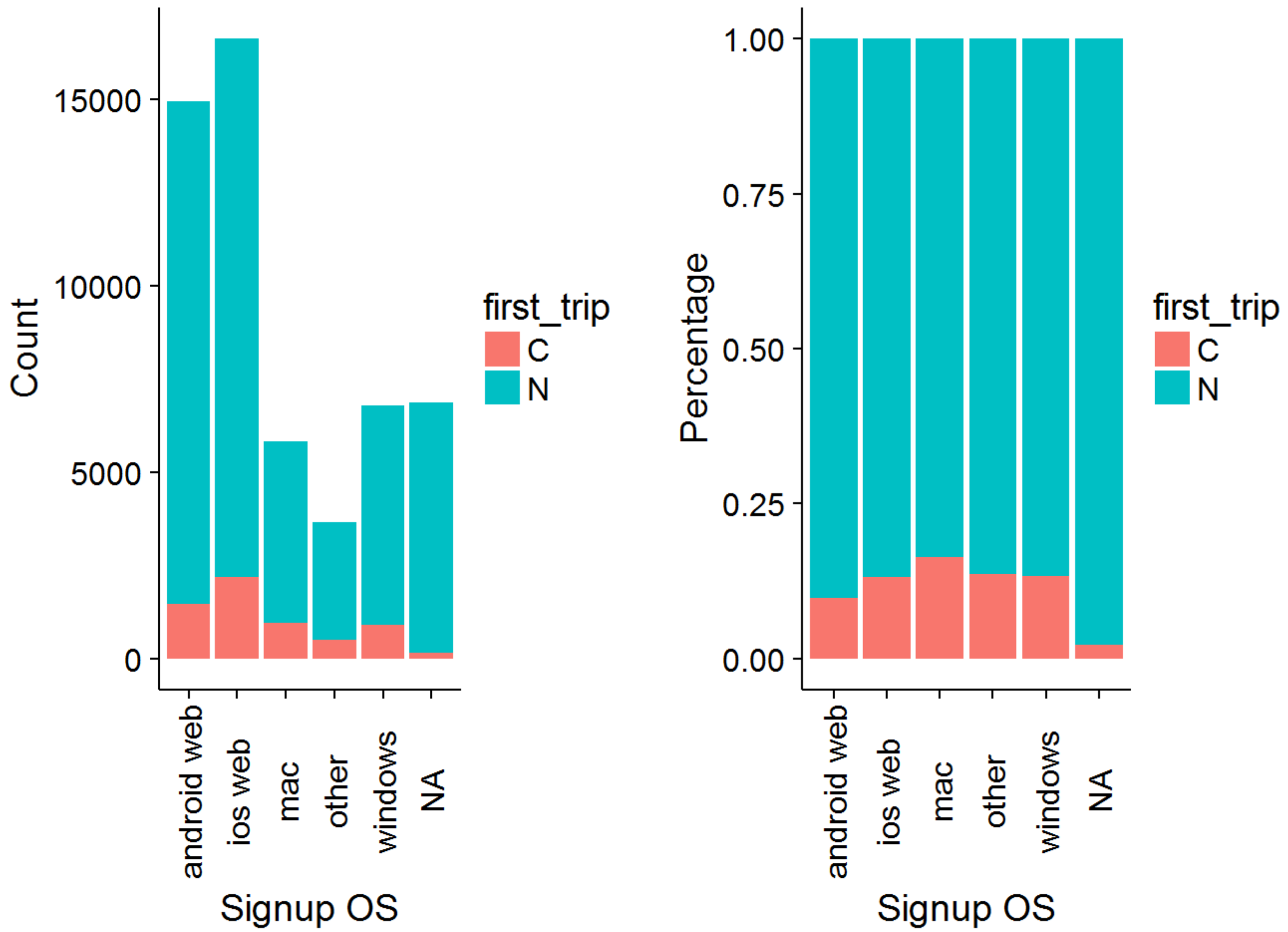
From the plots we could see that **Referral** has both highest first trip counts and rate, while **Paid** has highest signups but lowest first trip rate, which means **Referral** is the most efficient channel in terms of conversion rate(first trip rate), **Paid**, on the other hand, though we get nearly half signups from it,is the most unefficient channel.

Signup OS

```
#Plot signup_channel to see first trip rate
signup_os1 <- ggplot(data =signup, aes(x = signup_os, fill = first_trip))+ geom_bar() +labs(x =
"Signup OS",y = "Count") + theme(axis.text.x = element_text(angle=90, vjust=0.5))


signup_os2 <- ggplot(data =signup, aes(x = signup_os, fill = first_trip)) +
    geom_bar(aes(fill = first_trip), position = 'fill')+labs(x = "Signup OS",y = "Percentage") +
 theme(axis.text.x = element_text(angle=90, vjust=0.5))


plot_grid(signup_os1, signup_os2, ncol = 2, nrow = 1)
```



From the plots we could see that **Mac** has the highest first trip rate while **ios web** has the highest signups; And more than half signups coming from **mobile/tablet (andriod + ios)**

Modeling

Feature Engineering

Since all data are coming from the same cohort, we will use date difference instead of multiple dates to measure the duration between signup and take action (background check or add vehicle), which represent their decision time.
We will have 2 new variables:
`bgc_duration = bgc_date - signup_date`
`vehicle_added_duration = vehicle_added_date - signup_date`

```
signup$bgc_duration <- as.numeric(as.Date(as.character(signup$bgc_date), format="%m/%d/%y")-
                  as.Date(as.character(signup$signup_date), format="%m/%d/%y"))

signup$vehicle_added_duration <- as.numeric(as.Date(as.character(signup$vehicle_added_date), for
mat="%m/%d/%y")-          as.Date(as.character(signup$signup_date), format="%m/%d/%y"))
```

Missing Value

For variables `vehicle_make` , `vehicle_model` and `vehicle_year` , since about *75% (41458/54681)* are missing values, we just need to change them to binary variables (have value/missing value)

```
signup$have_vehicle_make <- as.factor(ifelse(is.na(signup$vehicle_make),"N","Y"))
signup$have_vehicle_model <- as.factor(ifelse(is.na(signup$vehicle_model),"N","Y"))
signup$have_vehicle_year <- as.factor(ifelse(is.na(signup$vehicle_year),"N","Y"))
```

For variables `bgc_duration` , `vehicle_added_duration` , we could assume that all missing values to be a very large number (since they maybe take action in the future), let's say *1000*

```
signup$bgc_duration <- ifelse(is.na(signup$bgc_duration),1000,signup$bgc_duration)
signup$vehicle_added_duration <- ifelse(is.na(signup$vehicle_added_duration),1000,signup$vehicle
_added_duration)
```

For variable `signup_os` , *12.5% (6857/54681)* are missing value, since *98% (6709/6857)* of the missing value are drivers without first trip, so we could ignore it

```
table(signup[is.na(signup$signup_os),]$first_trip)
```

```
##
##    C    N
##  148 6709
```

Data Preparation for Model

Select certain columns for building up a predictive modeling, `new_signup` will be out dataset for building up the model

```
new_columns <- c("city_name","signup_os","signup_channel","first_trip", "bgc_duration","vehicle_
added_duration","have_vehicle_make","have_vehicle_model","have_vehicle_year" )
new_signup <- signup[new_columns]
str(new_signup)
```

```
## 'data.frame':    54681 obs. of  9 variables:
##  $ city_name             : Factor w/ 3 levels "Berton","Strark",..: 2 2 3 1 2 2 2 2 2 1 ...
##  $ signup_os             : Factor w/ 5 levels "android web",..: 2 5 5 1 1 1 2 2 NA 2 ...
##  $ signup_channel        : Factor w/ 3 levels "Organic","Paid",..: 2 2 1 3 3 3 2 3 3 2 ...
##  $ first_trip            : Factor w/ 2 levels "C","N": 2 2 2 1 2 2 1 2 2 2 ...
##  $ bgc_duration          : num  1000 1000 0 5 15 0 2 10 1000 1000 ...
##  $ vehicle_added_duration: num  1000 1000 1000 5 16 4 7 1000 1000 1000 ...
##  $ have_vehicle_make     : Factor w/ 2 levels "N","Y": 1 1 1 2 2 2 2 1 1 1 ...
##  $ have_vehicle_model    : Factor w/ 2 levels "N","Y": 1 1 1 2 2 2 2 1 1 1 ...
##  $ have_vehicle_year     : Factor w/ 2 levels "N","Y": 1 1 1 2 2 2 2 1 1 1 ...
```

```
summary(new_signup)
```

```
##    city_name           signup_os      signup_channel  first_trip
##  Berton :20117   android web:14944   Organic :13427   C: 6137
##  Strark :29557   ios web    :16632   Paid    :23938   N:48544
##  Wrouver: 5007   mac        : 5824   Referral:17316
##                  other      : 3648
##                  windows    : 6776
##                  NA's       : 6857
##   bgc_duration    vehicle_added_duration have_vehicle_make
##  Min.   :   0.0   Min.   :  -5.0         N:41458
##  1st Qu.:   5.0   1st Qu.:1000.0         Y:13223
##  Median :  20.0   Median :1000.0
##  Mean   : 404.4   Mean   : 763.5
##  3rd Qu.:1000.0   3rd Qu.:1000.0
##  Max.   :1000.0   Max.   :1000.0
##  have_vehicle_model have_vehicle_year
##  N:41458            N:41458
##  Y:13223            Y:13223
##
##
##
##
```

Build up Predictive model

Initial Thoughts

The purpose of the model is to predict whether or not a driver signup will start driving, it is a classification problem.The dependent variable is binary, it is better to handle by **logistic regression** since it is more robust to binary output and really easy to interpret (We need to get insights from the model to generate more first trips, intepretable is as important as predictivity) **The alternative method are Decision Tree/Random Forest, Support Vector Machine,K-Nearest Neighbor, Neural Networks and Boosting**

```
set.seed(1)
# Change dependent variable to 0 and 1 to fit logistic regression
new_signup$first_trip <- ifelse(new_signup$first_trip =="C",1,0)

# split train data (75%) and test data (25%)
train_rows = sample.split(new_signup$first_trip, SplitRatio=0.75)
train = new_signup[ train_rows,]
test  = new_signup[!train_rows,]

#Build up logistic regression

model <- glm(first_trip~city_name+signup_os+signup_channel+bgc_duration+vehicle_added_duration+h
ave_vehicle_year,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = first_trip ~ city_name + signup_os + signup_channel +
##     bgc_duration + vehicle_added_duration + have_vehicle_year,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2301  -0.1656  -0.0924   0.0000   3.4113
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.324e+02  4.040e+00  32.781  < 2e-16 ***
## city_nameStrark        -1.104e-01  5.390e-02  -2.048  0.04057 *
## city_nameWrouver       -2.306e-01  9.769e-02  -2.360  0.01827 *
## signup_osios web        6.818e-02  6.476e-02   1.053  0.29238
## signup_osmac            4.717e-01  8.362e-02   5.641 1.69e-08 ***
## signup_osother          3.272e-01  1.025e-01   3.192  0.00141 **
## signup_oswindows        4.179e-01  8.447e-02   4.947 7.52e-07 ***
## signup_channelPaid     -1.632e-01  7.157e-02  -2.280  0.02262 *
## signup_channelReferral  5.247e-01  6.729e-02   7.797 6.32e-15 ***
## bgc_duration           -3.278e-02  5.116e-03  -6.407 1.48e-10 ***
## vehicle_added_duration -1.367e-01  4.062e-03 -33.639  < 2e-16 ***
## have_vehicle_yearY     -1.310e+02  4.014e+00 -32.640  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27063  on 35890  degrees of freedom
## Residual deviance: 10280  on 35879  degrees of freedom
##   (5120 observations deleted due to missingness)
## AIC: 10304
##
## Number of Fisher Scoring iterations: 13
```

Variable Selection

From the model summary we could tell that some variables are not important to the model, let's first explore 3 binary independent variables `have_vehicle_make` , `have_vehicle_model` , `have_vehicle_year` :

```
table(subset(new_signup,new_signup$have_vehicle_make == 'N')$have_vehicle_model)
```

```
##
##     N     Y
## 41458     0
```

```
table(subset(new_signup,new_signup$have_vehicle_make == 'N')$have_vehicle_year)
```

```
##
##     N     Y
## 41458     0
```

From above we know that if `have_vehicle_make` is `N` , the 2 others are `N` as well, bacically we just need to use one of these three binary variables, we will use `have_vehicle_make` in the model later on.

```
cor(new_signup$bgc_duration,new_signup$vehicle_added_duration)
```

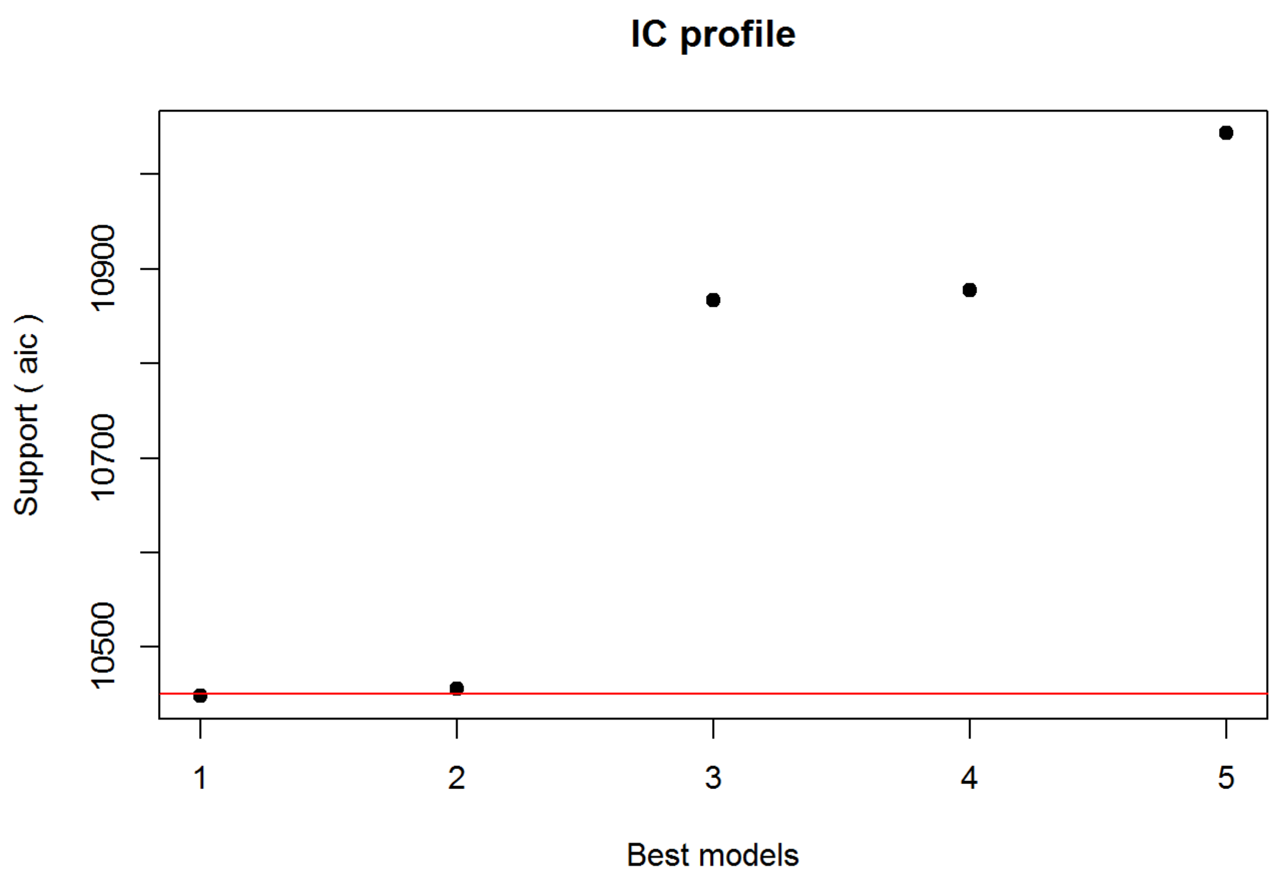
```
## [1] 0.4378981
```

Above we check if there are correlation between 2 numeric variables: `bgc_duration` and `vehicle_added_duration` to avoid multicollinearity, fortunately they are not highly correlated.

So now for the rest of the variables, since the size of variables and dataset are small, let's use **All Subset Variable Selection** method to select best variables for this model.

```
glmulti.logistic.out <-
    glmulti(first_trip~city_name+signup_os+signup_channel+bgc_duration+vehicle_added_duration+ha
ve_vehicle_make,
            data = train,
            level = 1,               # No interaction considered
            method = "h",            # Exhaustive approach
            crit = "aic",            # AIC as criteria
            confsetsize = 5,         # Keep 5 best models
            #plotty = F, report = F,  # No plot or interim reports
            fitfunction = "glm",     # glm function
            family = binomial)       # binomial family for logistic regression
```

```
## Initialization...
## TASK: Exhaustive screening of candidate set.
## Fitting...
##
## After 50 models:
## Best model: first_trip~1+city_name+signup_os+have_vehicle_make+bgc_duration+vehicle_added_dur
ation
## Crit= 10448.3725150049
## Mean crit= 10738.5350071129
```




```
## Completed.
```

The picture above shows AIC for top 5 performance models, let’s take a look on them:

```
glmulti.logistic.out@formulas
```

```
## [[1]]
## first_trip ~ 1 + city_name + signup_os + signup_channel + have_vehicle_make +
##     bgc_duration + vehicle_added_duration
## <environment: 0x0000000056940ba8>
##
## [[2]]
## first_trip ~ 1 + signup_os + signup_channel + have_vehicle_make +
##     bgc_duration + vehicle_added_duration
## <environment: 0x0000000056940ba8>
##
## [[3]]
## first_trip ~ 1 + city_name + signup_os + have_vehicle_make +
##     bgc_duration + vehicle_added_duration
## <environment: 0x0000000056940ba8>
##
## [[4]]
## first_trip ~ 1 + signup_os + have_vehicle_make + bgc_duration +
##     vehicle_added_duration
## <environment: 0x0000000056940ba8>
##
## [[5]]
## first_trip ~ 1 + city_name + signup_os + signup_channel + have_vehicle_make +
##     vehicle_added_duration
## <environment: 0x0000000056940ba8>
```

In terms of **AIC**, since No.1 model is almost the same as No.2, so `city_name` is not important to the model, for simplicity, we will use No.2 model.

Evaluating the model

```
best_model <- glmulti.logistic.out@objects[[2]]
# Evaluating model on test dataset and setting up decision boundary to be 0.5
fitted.results <- predict(best_model,newdata=test,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
fitted.results <- na.omit(fitted.results)

misClasificError <- mean(fitted.results != test$first_trip)
```

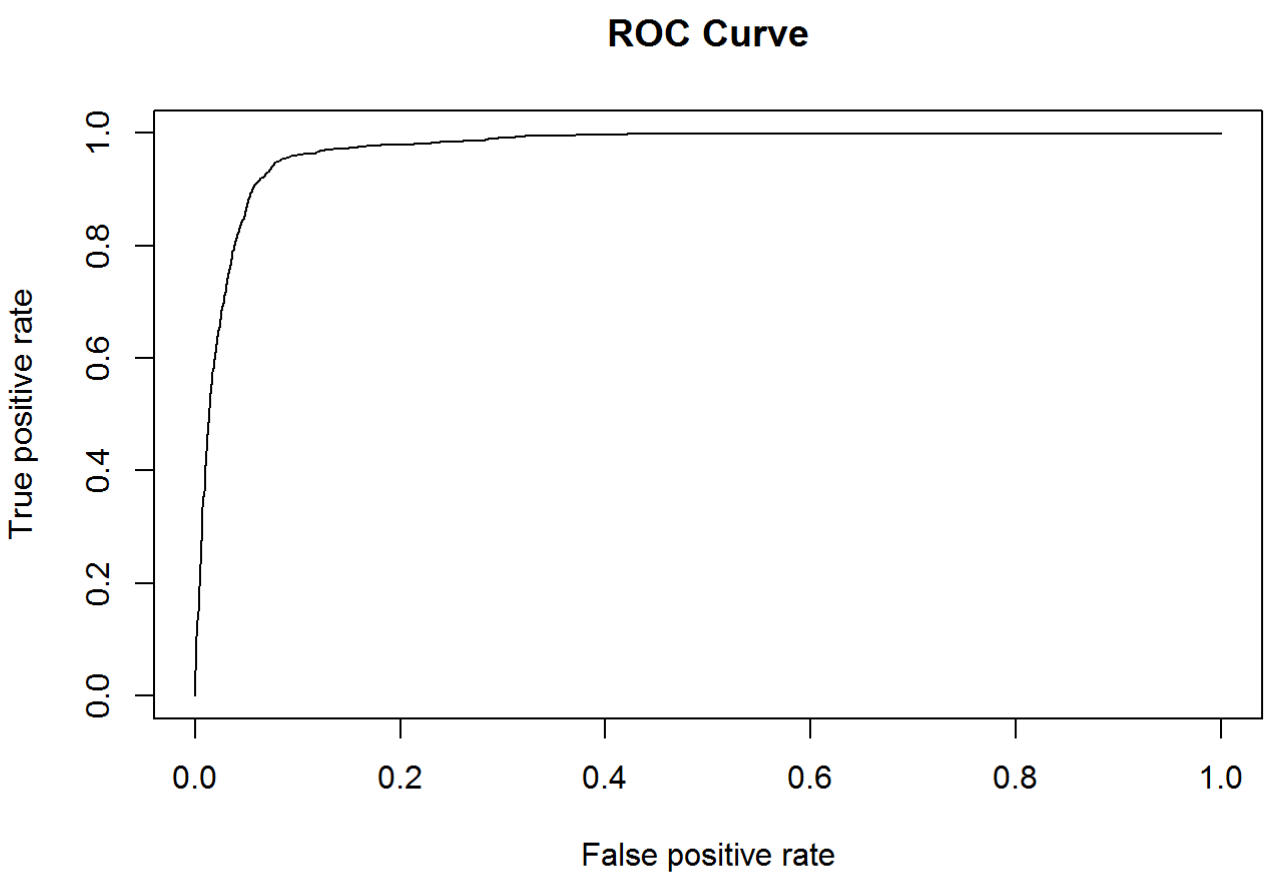
```
## Warning in fitted.results != test$first_trip: longer object length is not a
## multiple of shorter object length
```

```
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.785442574981712"
```

We setup threshhold to be *0.5* and get accuracy 0.785 (the classifier could label 78.5% of the test data correctly), which is pretty well in terms of logistic model, I could change the threshold or even running **cross validation** to improve the accuracy.
Now let's double check model performance by looking at its **ROC curve** and calculate **AUC**:

```
p <- predict(best_model, newdata=test, type="response")
pr <- prediction(p, test$first_trip)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf,main="ROC Curve")
```




```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
print(paste('AUC',auc))
```

```
## [1] "AUC 0.971845194647015"
```

For **ROC Curve**, X axis is **False Positive Rate** and Y axis is **Ture Positive Rate** We could see AUC (0.97) is pretty close to 1 which means the performance is really well.

Interpreting the model

Now let's take a closer look at No.2 model and its table of deviance (Chi Square Test)

```
summary(best_model)
```

```
##
## Call:
## fitfunc(formula = as.formula(x), family = ..1, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1985  -0.1655  -0.0920   0.0000   3.4055
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.322e+02  4.038e+00  32.735  < 2e-16 ***
## signup_osios web        7.568e-02  6.464e-02   1.171  0.24172
## signup_osmac            4.733e-01  8.358e-02   5.663 1.49e-08 ***
## signup_osother          3.245e-01  1.023e-01   3.171  0.00152 **
## signup_oswindows        4.190e-01  8.440e-02   4.965 6.88e-07 ***
## signup_channelPaid     -1.678e-01  7.154e-02  -2.346  0.01899 *
## signup_channelReferral  5.298e-01  6.723e-02   7.880 3.26e-15 ***
## have_vehicle_makeY     -1.309e+02  4.014e+00 -32.606  < 2e-16 ***
## bgc_duration           -3.300e-02  5.119e-03  -6.447 1.14e-10 ***
## vehicle_added_duration -1.365e-01  4.062e-03 -33.605  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27063  on 35890  degrees of freedom
## Residual deviance: 10288  on 35881  degrees of freedom
##   (5120 observations deleted due to missingness)
## AIC: 10308
##
## Number of Fisher Scoring iterations: 13
```

```
anova(best_model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: first_trip
##
## Terms added sequentially (first to last)
##
##
##                        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                  35890      27063
## signup_os              4    133.8     35886      26930 < 2.2e-16 ***
## signup_channel         2   2083.4     35884      24846 < 2.2e-16 ***
## have_vehicle_make      1   9932.5     35883      14914 < 2.2e-16 ***
## bgc_duration           1   2922.0     35882      11992 < 2.2e-16 ***
## vehicle_added_duration  1   1703.8     35881      10288 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From above we could learn:

Signup OS

The deviance table shows `Signup_OS` is an important variable which reduce residual deviance 133.8 by adding it.
Summary table tell us that drive signup from **desktop (Windows and Mac)** will have more log odds (0.419 and 0.473, respectively) to convert, which match the conclusion from exploratory analysis (Mac and Windows are among the top in terms of first trip rate), though we acquired more signups from **mobile/tablet (IOS and Andriod)**

Signup Channel

The deviance table shows `Signup_channel` is an highly important variable which reduce residual deviance 2083.4 by adding it.
Summary table tell us that driver coming from **Referral** will have more log odds (0.529) to convert while **Paid** will have less log odds (-0.168), which also match our conclusion from exploratory analysis (**Referral** is the most efficient channel and **Paid** is the worst)

BGC Duration and Vehicle Added Duration

These two variables represent the consideration time after signup, both of them are pretty significant from the deviance table.
Summary table tell us that both of them have negtive influence on first trip (-0.136 and -0.033), which make sense. Since the more time you consider, the less possible you will try it.

Have Year Make

This one could be the most confused variable, we could see it is super important variable(reduce residual deviance 9932.5 by adding it), the summary said “have year make” has negtive effect on first trip which does not make sense, we need to take a closer look at both value equals *Y* and *N*:

```
table(subset(new_signup,new_signup$have_vehicle_make == 'Y')$first_trip)
```

```
##
##    0    1
## 7350 5873
```

```
table(subset(new_signup,new_signup$have_vehicle_make == 'N')$first_trip)
```

```
##
##     0     1
## 41194   264
```

The first table is first trip distribution when a driver have vehicle make information while second table is is first trip distribution when a driver does have, clearly we could see that if a dirver does not have vehicle make information most likely (99.4%) mean he would not have first trip, there is no doubt that having vehicle make information will significantly increase the possibility of first trip,I think the reason why `have_vehicle_makeY` have a negtive parameter is due to *Perfect Separation*. Since most of values are *N* in `have_vehicle_make` and most of the values are *0* in `first_trip` (the dataset is *too sparse*).
So we could conclude that driver with vehicle information (make, model and year) are more likely to have first trip, which make sense since driver will provide more information is he is interested in Uber.

Insights and Actions

1. **Uber should re-allocate some money from other channel (like Paid) to referral** since it is the most efficient channel in term of getting first trip. Possible methods could be **increasing referral bonus and decrease Paid cost**. However, we also need to think about the **Cost per First Visit** from each channel which we cannot get from current dataset.

2. Desktop user have higher first trip rate but lower signups mean desktop user have higher quality than mobile/tablet user. Maybe mobile/tablet UI is easier for signup but desktop not, or maybe Uber is more advertising on mobile/tablet than desktop. Both reasons means **Uber need to focus on desktop, either UI design or Advertising, to get more signups on desktop.**

3. Since conderation time matters,**Uber need to continually keep drivers' attention by**
   A. Regularly send out email reminder after signup about background check and add vehicle
   B. give small bonus to rewarding the driver who take action (background check/add vehicle or even first trip) within a short period (like one week, two weeks or a month).