PREDICTING DRIVER CHURN WITH LOGISTIC REGRESSION

A case-study of drivers in the ride-hailing industry

[1]Enioluwamo Ireoluwa Obatoki

Abuja, Nigeria.

Abstract : The ride-hailing industry is no longer a novel phenomenon. Since 2009, when the pioneer ride-hailing giant, Uber, was founded, other similar companies have sprung up around the world, bringing on-demand transportation to millions of people at the tap of a button. Uber is market leader in many of the countries it operates in. However, the company has seen fierce competition from other regional giants - In Asia, Grab, Ola and Didi-Chuxing have given it a run for its money and in Europe and Africa, Bolt(Formerly Taxify) and Yango are fierce contenders. As the global ride-hailing scene continues to take shape, one thing is clear, that the ecosystem is heavily reliant on the real service providers - "Drivers", who are independent contractors providing the vehicles and services to the consumer market.

This paper is a study of drivers in the ride-hailing industry; what are the key drivers of driver churn? and subsequently, how can this knowledge be used to reduce churn and increase driver activity?

***IndexTerms* - Driver, Active driver, Quality, Hours, RPH, Earnings, Churned driver, Fully churned driver.**

I.INTRODUCTION

Drivers in the ride-hailing industry are independent contractors - This means they decide their own hours, when to go online or offline unlike regular employees who must stick to a fixed schedule. However, there are other terms that must be adhered to in the relationship with the ride-hailing app; One of these is the quality of service given to users. For many apps, the minimum average rating a driver must have is 4.4 - 4.5 stars. This figure is an average of the ratings given after completed trips. Depending on the app, the algorithm may check the last 10, 20 or 50 rated trips.

How much money does a driver make weekly? This is dependent on a couple of factors;

1. the total number of hours the driver is online and available to take requests.
2. the total number of trips completed - a factor of the demand(or orders) in that period, the percentage of accepted orders
3. the commission collected by the company per trip - Uber charges 25% fee in many of the countries it operates in. Other players, like Bolt, have identified an opportunity to take market share by offering drivers lower commissions - 15%
4. how much is spent on vehicle maintenance, insurance and fuel.

This paper seeks to identify the relationship between Hours, Quality, RPH, Earnings and driver churn.

II. THEORETICAL FRAMEWORK

Python 3 was used to run the regressions and tests. The regression is run on 4 key categories of driver data from October 2017 till May 2019.

Four independent variables are used in this analysis: Hours, Quality, RPH and Earnings with each classified as "Great", "OK", "Needs Attention" depending on the threshold set for each variable.

Great - the driver is performing excellently on that metric. e.g if Quality is "Great", it means, he/she offers good service and their average rating is 4.7 stars or higher.

OK - the driver's performance is fair on that metric. e.g if Quality is "OK", he/she's average rating is less than 4.7 but greater than or equal to 4.5.

Needs attention - the driver's performance on the metric is poor.. e.g if Quality "Needs Attention", he/she's average rating is less than 4.5.

Keywords:

Driver - An independent contractor who has been accepted into the ride-hailing ecosystem to provide transportation services with his/her vehicle to app users on-demand.

Active driver - a driver who has gone online on the ride-hailing app in the past week

Quality - the average rating a driver has. This rating is assigned by the drivers customers after a trip is completed.

Hours - the number of hours the driver is active(or online) and available to take ride requests.

RPH - Rides per Hour

Earnings - the amount of money the driver makes in trip fares

Churned driver - a driver who has not gone online on the ride-hailing app in the past week

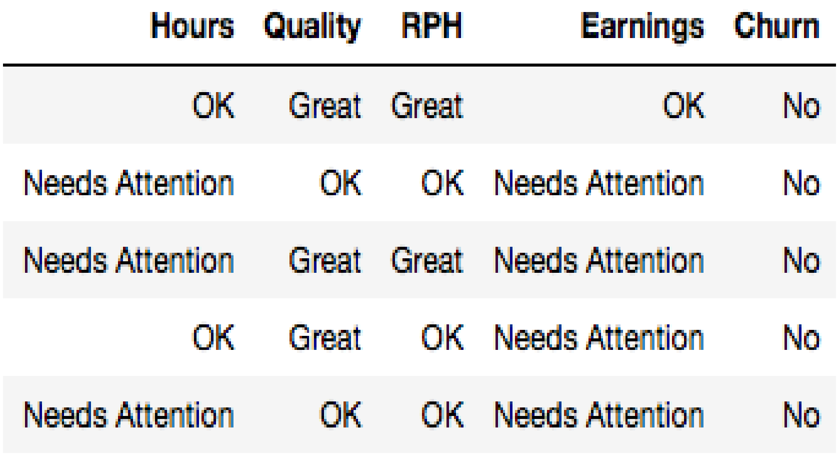
Fully churned driver - a driver who has not gone online on the ride-hailing app in the past 4 weeks

III. RESEARCH METHODOLOGY

3.1 Data Importation

First step is to import useful libraries and read in the data in the csv file:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline

data = pd.read_csv('/Users/ire/Documents/Python/Predicting_Churn_Train_Test.csv')
data.head()
```



Hours	Quality	RPH	Earnings	Churn
OK	Great	Great	OK	No
Needs Attention	OK	OK	Needs Attention	No
Needs Attention	Great	Great	Needs Attention	No
OK	Great	OK	Needs Attention	No
Needs Attention	OK	OK	Needs Attention	No

3.2 Data Preparation

Next, the data which is in string format is converted to numbers as the logistic regression can only read encoded values:

```
data['Hours'].replace(['Needs Attention','OK','Great'],[0,1,2],inplace=True)
data['Quality'].replace(['Needs Attention','OK','Great'],[0,1,2],inplace=True)
data['RPH'].replace(['Needs Attention','OK','Great'],[0,1,2],inplace=True)
data['Earnings'].replace(['Needs Attention','OK','Great'],[0,1,2],inplace=True)
data['Churn'].replace(['Yes','No'],[1,0],inplace=True)
data.info()
data.pop('ID')
```

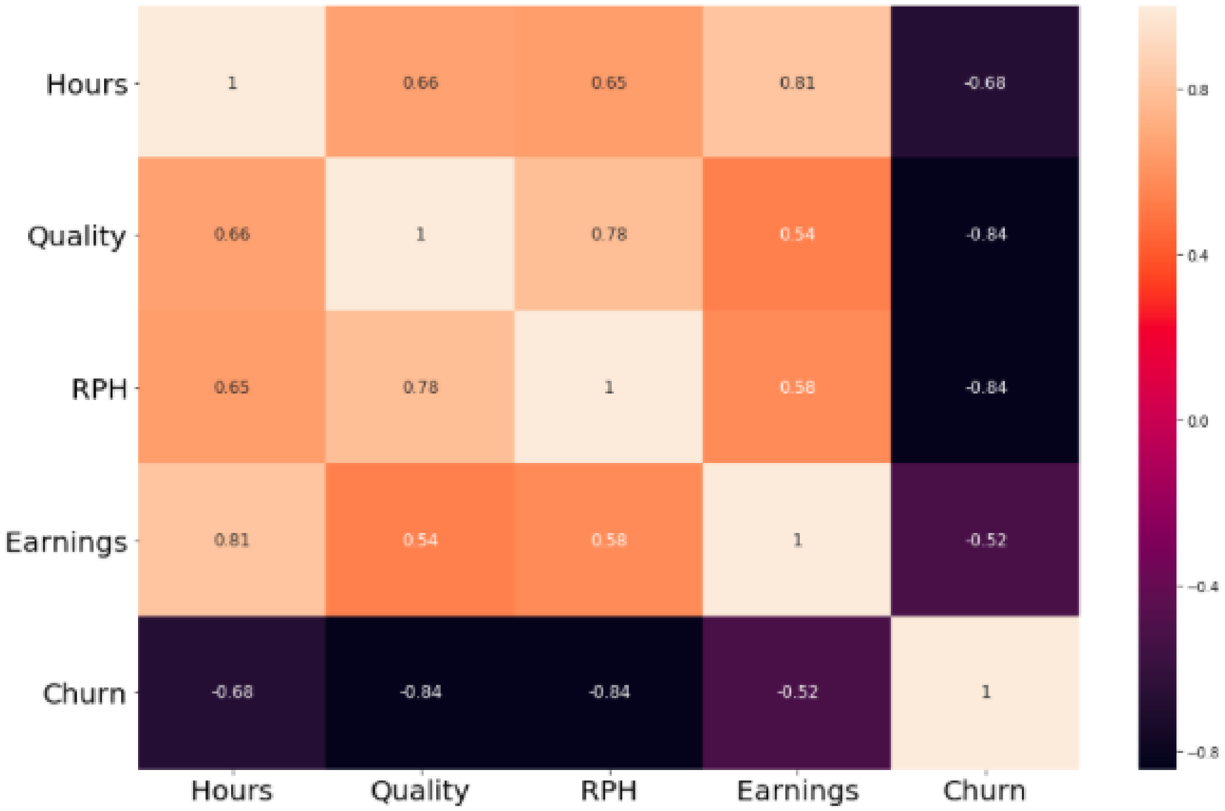
In the last line, the ID is “popped” out of the data because it is not relevant. All other strings are replaced with 0,1,2.

3.3 Data Visualisation

Next, we plot a correlation matrix:

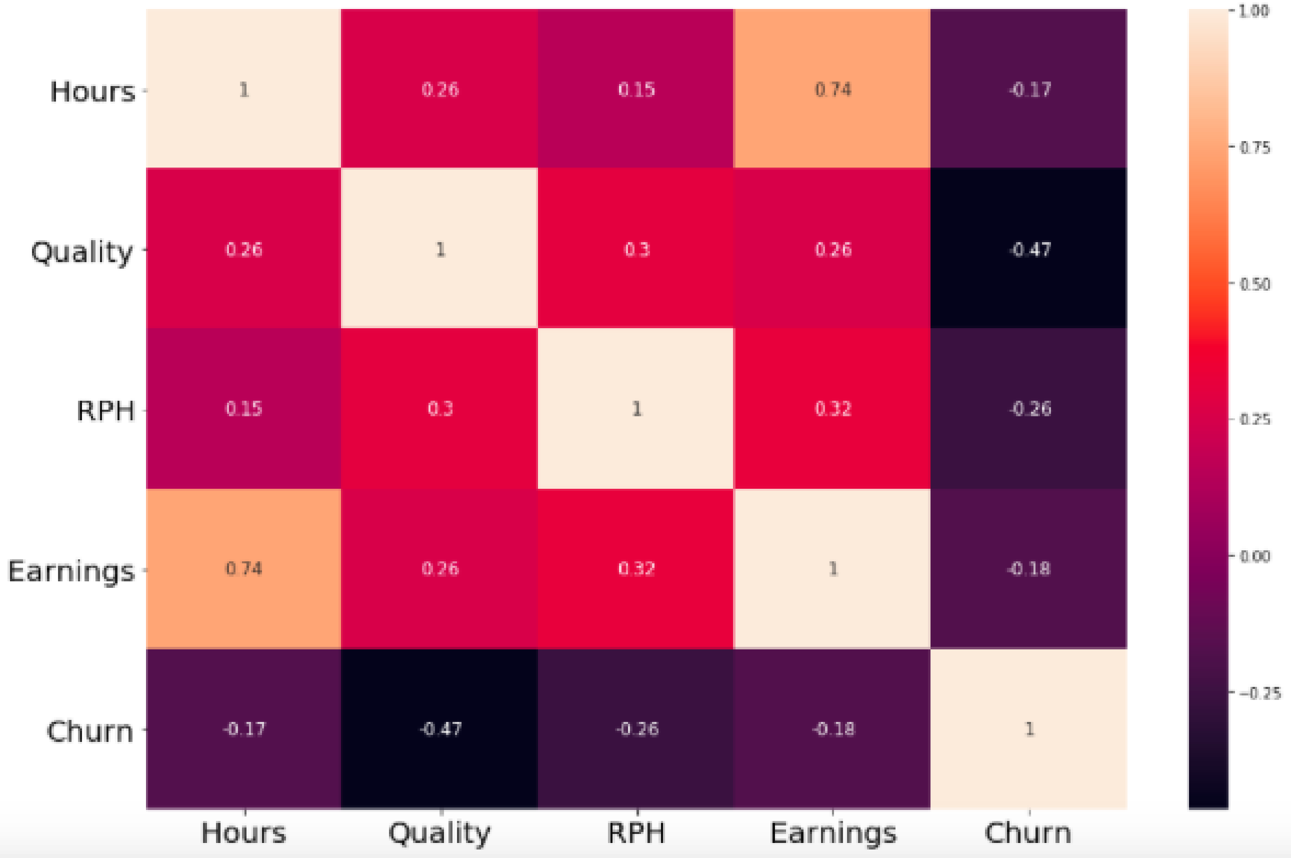
```
corr = data.corr()
sns.heatmap(corr, xticklabels=corr.columns.values, yticklabels=corr.columns.values, annot = True, annot_kws={'size':12})
heat_map=plt.gcf()
heat_map.set_size_inches(15,10)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.show()
```

[1]



From the first iteration, Quality and RPH are most strongly correlated with Churn.
Data contained all drivers' 4-week data, including those who have already fully churned.

When all fully churned drivers were removed, correlation became:



3.4 Model Fitting

Using the train_test_split module in scikit learn, we decide to train our model on 80% of the available data and use 20% to test its accuracy:

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(data, test_size = 0.20)

train_y = train['Churn']
test_y = test['Churn']

train_x = train
train_x.pop('Churn')
test_x = test
test_x.pop('Churn')
```

Using the Logistic Regression module in scikit learn, we run our model to predict dependent variable, Churn, based on each of the 4 independent variables:

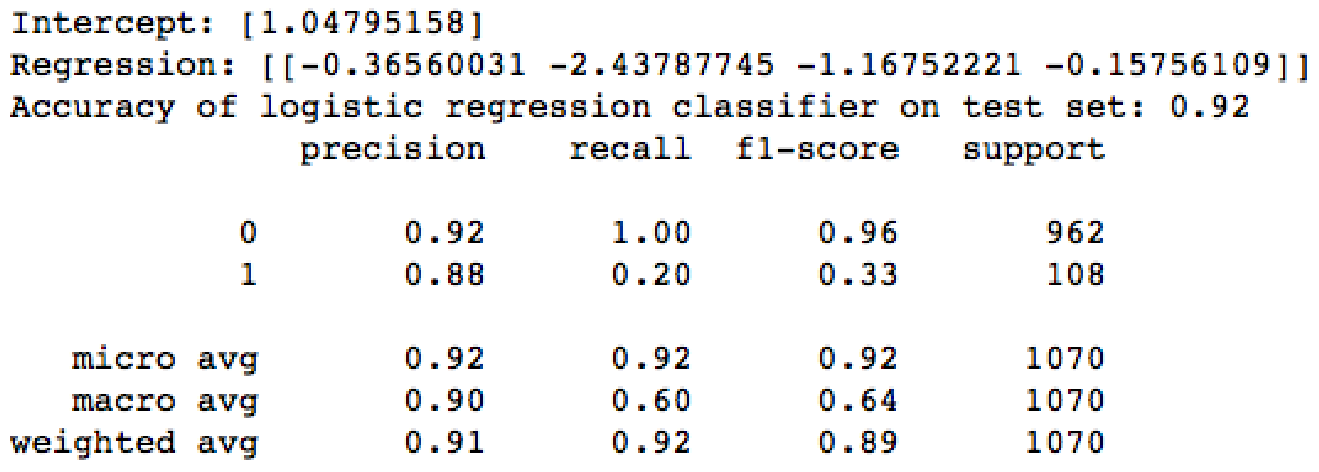
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report

logisticRegr = LogisticRegression()
logisticRegr.fit(X=train_x, y=train_y)

test_y_pred = logisticRegr.predict(test_x)
confusion_matrix = confusion_matrix(test_y, test_y_pred)
print('Intercept: ' + str(logisticRegr.intercept_))
print('Regression: ' + str(logisticRegr.coef_))
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logisticRegr.score(test_x, test_y)))
print(classification_report(test_y, test_y_pred))

confusion_matrix_df = pd.DataFrame(confusion_matrix, ('No churn', 'Churn'), ('No churn', 'Churn'))
heatmap = sns.heatmap(confusion_matrix_df, annot=True, annot_kws={"size": 20}, fmt="d")
heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0, ha='right', fontsize = 14)
heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=45, ha='right', fontsize = 14)
plt.ylabel('True label', fontsize = 14)
plt.xlabel('Predicted label', fontsize = 14)
```

[1]

IV. RESULTS AND DISCUSSION

4.1 Interpretation of the model

The model gave these regression values:



```
Intercept: [1.04795158]
Regression: [[-0.36560031 -2.43787745 -1.16752221 -0.15756109]]
Accuracy of logistic regression classifier on test set: 0.92
              precision    recall  f1-score   support

           0       0.92      1.00      0.96       962
           1       0.88      0.20      0.33       108

   micro avg       0.92      0.92      0.92      1070
   macro avg       0.90      0.60      0.64      1070
weighted avg       0.91      0.92      0.89      1070
```

4.1.1 Precision:

0. Of those predicted to remain active, what proportion actually remained active? Our model says 92% of those predicted to remain active were actually active.

1. Of those predicted to Churn, what proportion actually churned? Our model says 88% of those predicted to churn, churned.

4.1.2 Recall:

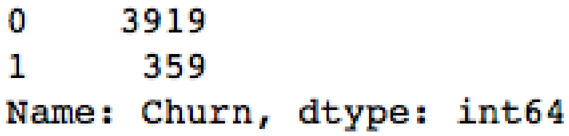
0. Of those who actually did not churn, what proportion

1. Of those who actually churned, what proportion were predicted to? Our model says only 20% of those who churned were predicted to do so

Our model isn’t so reliable so we ‘upsample’ the minority group: Churn, so that the number of churned is equal to the number of active in the data set:

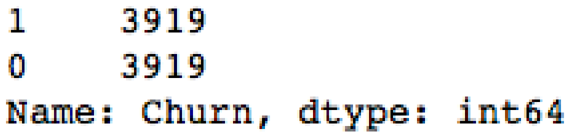
```
from sklearn.utils import resample
#to save the final model to disk, import pickle
import pickle
data_majority = data[data['Churn']==0]
data_minority = data[data['Churn']==1]
data_minority_upsampled = resample(data_minority,
replace=True,
n_samples=3919, #same number of samples as majority classe
random_state=1) #set the seed for random resampling
# Combine resampled results
data_upsampled = pd.concat([data_majority, data_minority_upsampled])
data_upsampled['Churn'].value_counts() [1 - 3]
```

Before



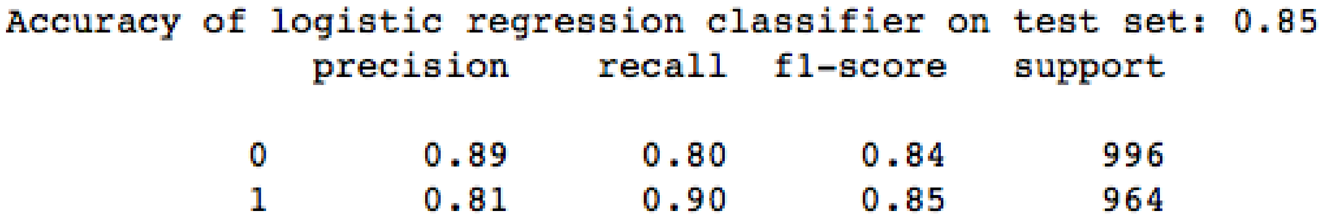
```
0    3919
1     359
Name: Churn, dtype: int64
```

After



```
1    3919
0    3919
Name: Churn, dtype: int64
```

Now, our new model gives:



```
Accuracy of logistic regression classifier on test set: 0.85
              precision    recall  f1-score   support

           0       0.89      0.80      0.84       996
           1       0.81      0.90      0.85       964
```

4.1.3 Precision: 1. 81% of those predicted to churn, churned.
4.1.4 Recall: 1. 90% of those who churned were predicted to do so.
This is a more accurate model than the first with a 20% recall for churned drivers

V. CONCLUSIONS

Of the four variables, Quality and RPH are the most correlated to churn.

1. The lower the average rating of the driver, the more likely it is that he/she will stop driving. This can be due to either of 2 reasons: a. the driver was suspended temporarily or permanently due to the company’s quality checks. Drivers with poorer ratings are more likely to be suspended. b. the driver is generally disgruntled and is giving poor customer service for a while before finally deciding to quit driving.
2. The lower the number of rides completed per hour, the higher the likelihood of churn.This can be due to either of 2 reasons: a. the driver is not in high demand areas and hence is completing fewer trips per online hour which means he is earning less but most importantly is poorly utilized.
3. This model predicts driver churn with 81% precision and 90% recall.

VI. ACKNOWLEDGEMENT

I want to recognize the contribution of all who have gone before to create these modules, methods and packages which are extremely useful for predictive analytics in the present day.

REFERENCES

[1] Predicting customer churn with Python: Logistic regression, decision trees and random forests Thomas June 5 2018, http://dataskunkworks.com/2018/06/05/predicting-customer-churn-with-python-logistic-regression-decision-trees-and-random-forests/

[2] Making Predictions with Data and Python, Predicting Credit Card Default, Alvaro Fuentes, August 2017

[3] Logistic Regression in Python Anish Singh Walia Mar 9 2019, https://medium.com/@anishsingh20/logistic-regression-in-python-423c8d32838b