Predictive Analytics

MPBA G513

Project Report

Submitted to:

Dr.Vamsidhar Ambatipudi

Department of Management, BITS Pilani

Pilani Campus

2021-2023

Submitted by:

Sayantan Paul - 2021H1540807
Karan Balaji S -2021H1540810
Parthasarathi Bera - 2021H1540811
Indika Debnath - 2021H1540818
Aditi Gupta- 2021H1540821P

Date of submission: 05-05-2022

UBER Churn Prediction

Project Report

Customer churn is a major problem of customers leaving UBER service/subscription and moving to another service. Due to the direct effect on profit margins, businesses now are looking to identify customers who are at the risk of churning and retaining them by personalized promotional offers. In order to retain them, they need to identify the customers as well as the reason for churning so that they can provide the customers with personalized offers.

The aim of our project is to solve this problem for the transportation domain (UBER service), by identifying which customers are at risk of churning and what are the reasons for churning with the help of data mining and machine learning algorithms. The project focuses on the below deliverables - Predict customers likely to churn using supervised learning classification algorithms and customer segmentation of customers to validate the similarities in the 'likely to churn' customer subset to come up with different segments. The reasons for a particular customer churn can vary from internal factors as well as external factors but we will try to understand the reasons for churn depending on internal factors using explainable AI, which breaks into the black box of machine learning algorithms and gives a clear explanation of the predictions.

The data is taken from an online source, where it is very important to maintain a healthy supply of cab drivers employed with companies like Uber, and Ola and maintain supply in the marketplace. Being able to predict when a particular driver might churn gives the added advantage of being able to take action accordingly and in a timely fashion.

In this project, we tried to imitate the machine learning algorithms to detect the churn and come up with a solution to mitigate the problem.

Problem Statement

Uber is interested in predicting rider retention in the US. To help explore this problem, they have provided a sample dataset of a cohort of users who have signed up for an uber account and have taken a ride in January 2014.

Dataset Description

The dataset contains 50,000 rows and 14 columns which has been highlighted below which describes what the columns mean.

- city: city this user signed up in
- phone: primary device for this user while registering or booking.
- signup_date: date of account registration; in the form 'YYYY-MM-DD'
- last_trip_date: the last time this user completed a trip; in the form 'YYYY-MM-DD'
- avg_dist: the average distance *(in miles) per trip taken in the first 30 days after signup
- avg_rating_by_driver: the rider's average rating over all of their trips
- avg_rating_of_driver: the rider's average rating of their drivers over all of their trips
- surge_pct: the percent of trips taken with surge multiplier > 1
- avg_surge: The average surge multiplier over all of this user's trips
- trips_in_first_30_days: the number of trips this user took in the first 30 days after signing up
- luxury_car_user: True if the user took an luxury car in their first 30 days; False otherwise
- weekday_pct: the percent of the user's trips occurring during a weekday.

Metrics:

Accuracy is a common metric for binary classifiers; it takes into account both true positives and true negatives with equal weight.

$$accuracy = \frac{true\ positives + true\ negatives}{dataset\ size}$$

This metric was used when evaluating the classifier because false negatives and false positives both erode the predict the possibility:

❖ False negatives result in either a longer delay between the user pointing the camera text and device speaking the text ("processing delay") or in the worst case, completely prevent the application from reading said text

❖ On the other hand, false positives make the application try to extract text from images that don't contain any. This results in unnecessary computations on the remote server, which can be both costly and slow the application down. In the worst case, this might also result in the application reading gibberish.

Processing delay (defined above) is also a metric that has a big effect on the user experience. It can be broken into two components, as the image processing is done in two steps:

processing delay ≈ classification delay + extraction delay

❖ The classification delay is the time it takes for the classifier to detect text; it is important by itself, as the application provides feedback to the user immediately after it detects text.

❖ The extraction delay is the time it takes for the application to start speaking after text was detected by the classifier.

Exploratory Data Analysis

There is an unknown column 'Unnamed: 0', in the dataset, so we can take it as ID of each observation. Renaming it into ID column. Using the info() to get the information of the data.

#	Column	Non-Null Count	Dtype
0	ID	50000 non-null	int64
1	avg_dist	50000 non-null	float64
2	avg_rating_by_driver	49799 non-null	float64
3	avg_rating_of_driver	41878 non-null	float64
4	avg_surge	50000 non-null	float64
5	city	50000 non-null	object
6	last_trip_date	50000 non-null	object
7	phone	49604 non-null	object
8	signup_date	50000 non-null	object
9	surge_pct	50000 non-null	float64
10	trips_in_first_30_days	50000 non-null	int64
11	luxury_car_user	50000 non-null	bool
12	weekday_pct	50000 non-null	float64
13	churn	50000 non-null	int64

As "ID" feature will not add any value to the dataset so we ignore it for now and calculate the 5 point summary! Using describe() function.

1. Count parameter of Five-Point summary let us know that there are few missing/null values present in dataset.
2. Variables namely "avg_dist", "surge_pct" and "trips_in_first_30_days" have huge deviation between mean and max value. This might be the indicator of huge-number of or large outlier(s) present. This will clear out in further processes.

From the description of Categorical Columns:

1. We have null/missing values in "phone" variable.
2. There are 31 unique values in "signup_date" we need to see the anomalies present in it.

Percentage of driver who churned are 53.53 and who did not churned is 46.462.So, it is fairly a balanced dataset.

Null/missing value inferences:

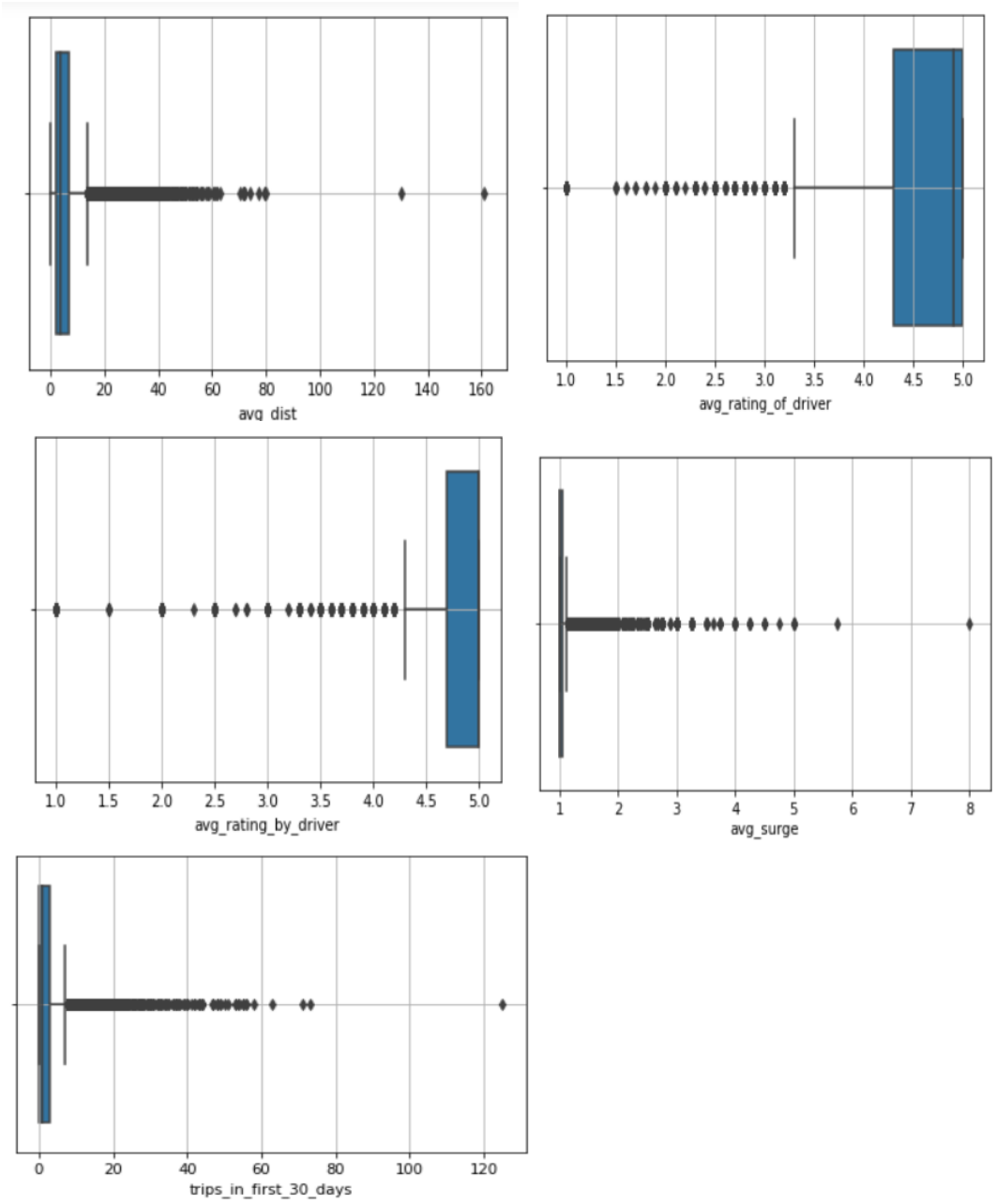
1. In this data-set there is not much null values. Amongst all the possible predictors,"avg_rating_of_driver" contains maximum null value i.e. around 16%. Variables "avg_rating_by_driver" and "phone" are 0.402% and 0.792% which are quite negligible.
2. After giving quite a thought to these variables, we decided not to treat this null/missing values for now. As there null can be declared as IGNORANCE / NEGLIGENCE we might be able infer out few business solutions or restrictions.

Using skewness function, we got:

```
ID                      0.000000
avg_dist                3.464170
avg_rating_by_driver    -4.128909
avg_rating_of_driver    -2.428485
avg_surge               6.821346
surge_pct               3.144124
trips_in_first_30_days  5.167755
luxury_car_user         0.507262
weekday_pct             -0.477788
churn                   -0.141880
dtype: float64
```

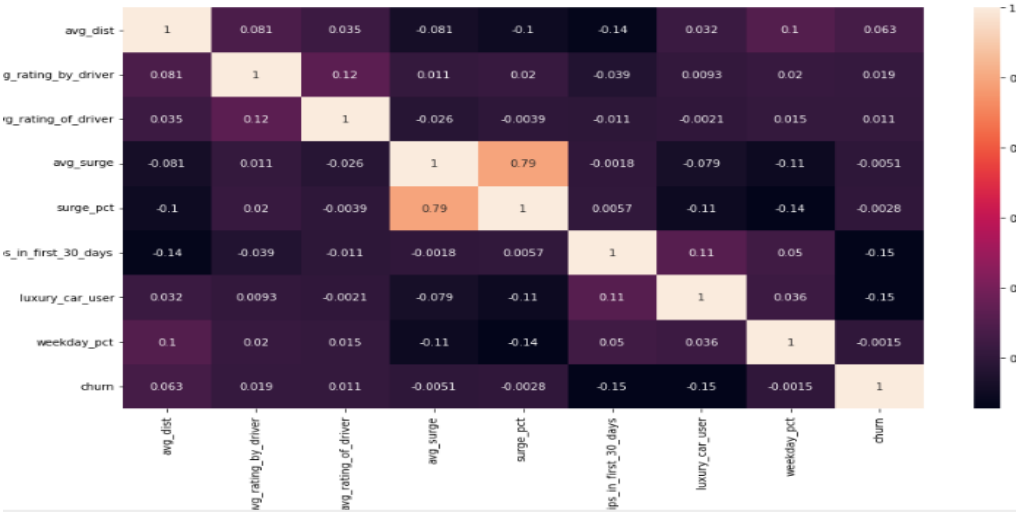
We can only say that many of them are light to heavily skewed. "weekday_pct" is the only variable which is widely distributed. We will be using various transformation techniques to treat this varied variable. "avg_dist", "avg_surge", "surge_pct" and "trips_in_first_30_days" are the features which might belong to Log Normal or might use Box-cox or yeo Johnson.

Outlier analysis:



The boxplot shows us that some of the variables have huge number of outliers. It will bias out prediction and analysis. So we need to process it.

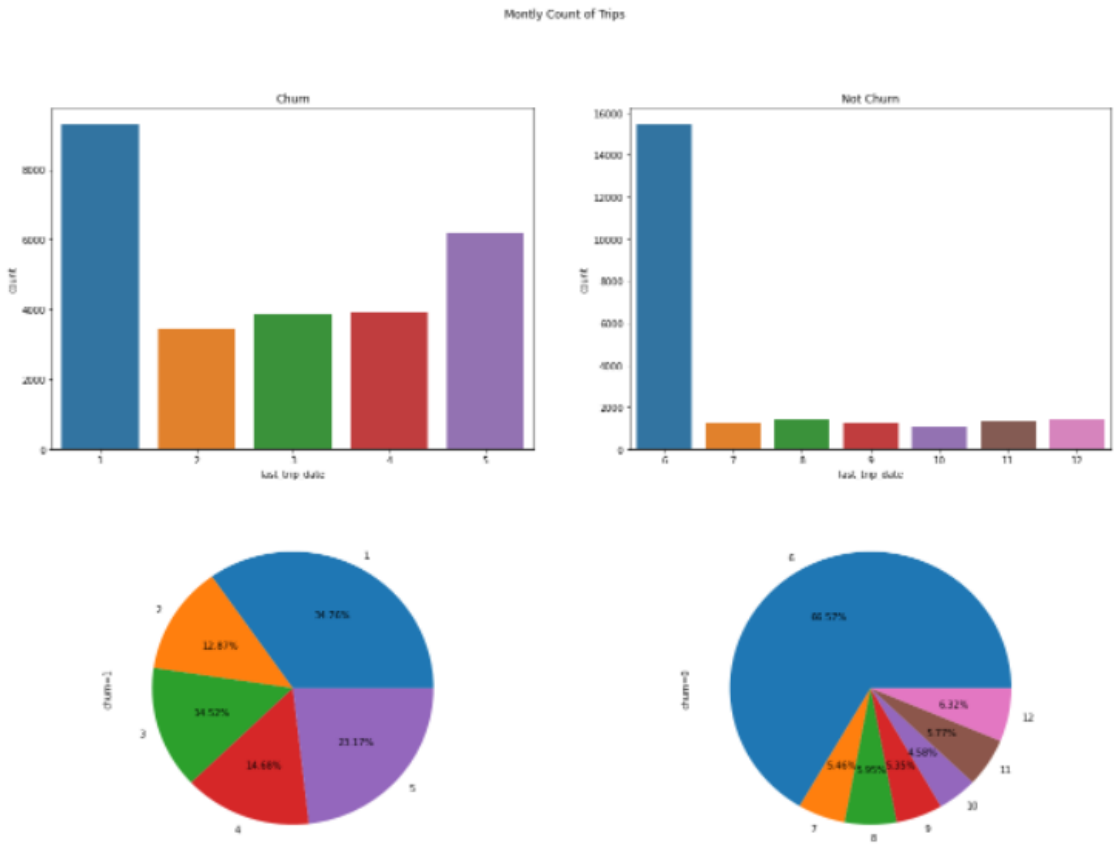
Heat map showing correlation between variables.



From heatmap we get to know that "Surge_pct" and "avg_surge" are highly positively correlated (0.79). This positively high correlation states that these variables are similar to each other. This correlation will lead to multicollinearity and the assumption of NO MULTICOLINEARIY will fail. This will ultimately lead to low accuracy of our models. We can treat this multicollinearity using PCA.

As we know scatter plot is the best plot/graph to visualize the relationship between the variables. We decided to go ahead with joint plot and regplot which is a modified version of scatterplot, these plots are more inferential than scatterplot. From joint plot which is combination of scatterplot and histogram, it is distribution of each feature and relation between them can be readily interpretable.

Reg plot gives us the a straight line or the best fit line of the variables on scatter plot which gives a fairly good idea of the slope of their relation.¶

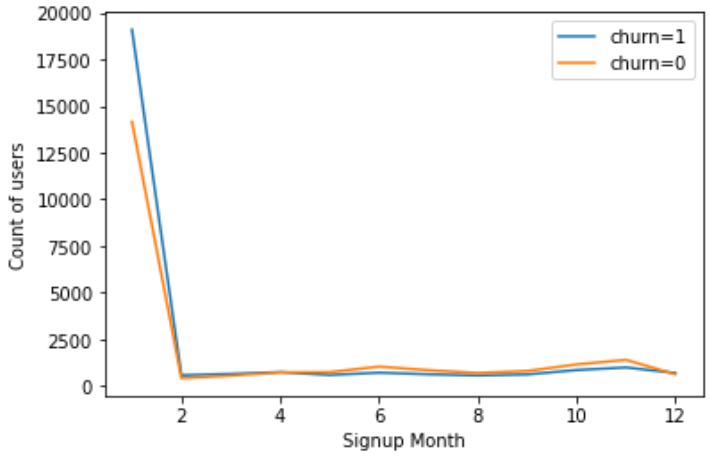


The above graphs shows the sublimation on the basis of customers who are churned and not churned count. It shows count of last trip month wise.

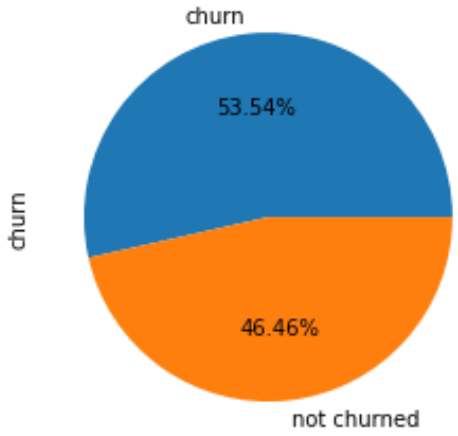
1. We got a very unique inference from this distribution. Customer who are churned and not churned is been split in two halves of a year. In the year 2014 customers have churned in first half of the year i.e., in first five months.
2. We can see there is a peek on the 6th month and on the same month we can see the customers have stopped leaving this cab service.
3. It gave an sight that the campaign/promotion which is done was very successful and after that we got a stable flow of trip each month.
4. We recommend that they should upgrade this promotion/campaign to increase the frequency of last trip.
5. They can even go for same campaign each year as it is touching a peak of around 16000 count in a single month which is an exponential increase.

Pie plot is noted to give the best inference in the percent for of our data. From the above pie plot which is in regard with the Monthly count of trips, we can easily justify the Monthly count of trips graph.

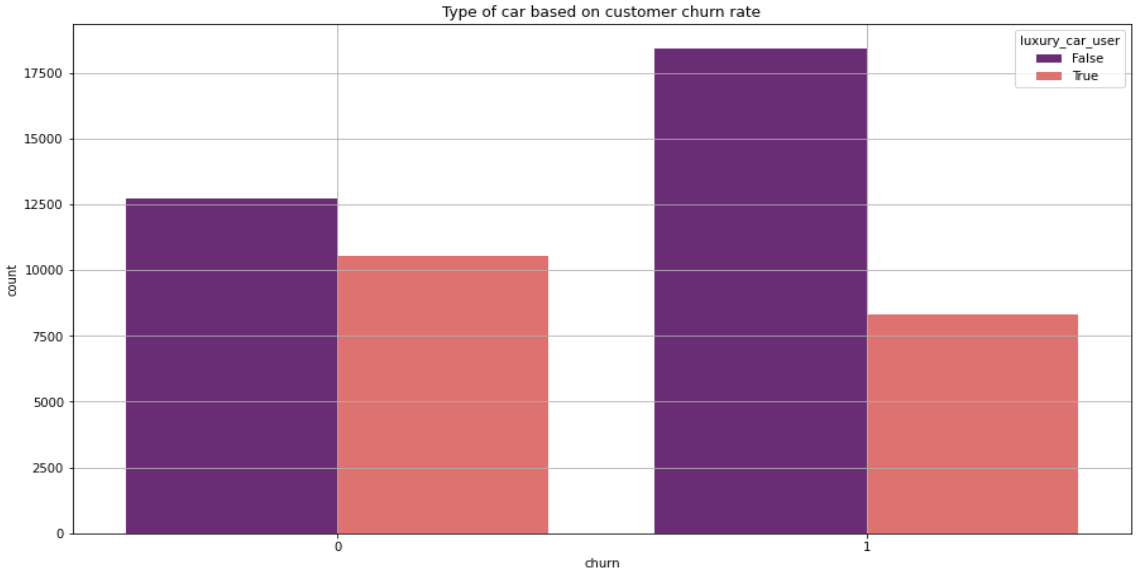
1. This pie plot tells us that customers who are un-subscribing the cab service 2. is taking place only in the first 5 month of 2014. And January has the highest churn percentage 32.12% followed by May 24.71%.
2. We even get a hike in a active customers in the month of June i.e. 63.26% which can be stated as a new all-time peak of active customers in the year of 2014.
3. It shows that this company faced a very successful campaign/promotion. On the later months from July to December we can see a stable active user every month, which shows the stability due to the campaign.



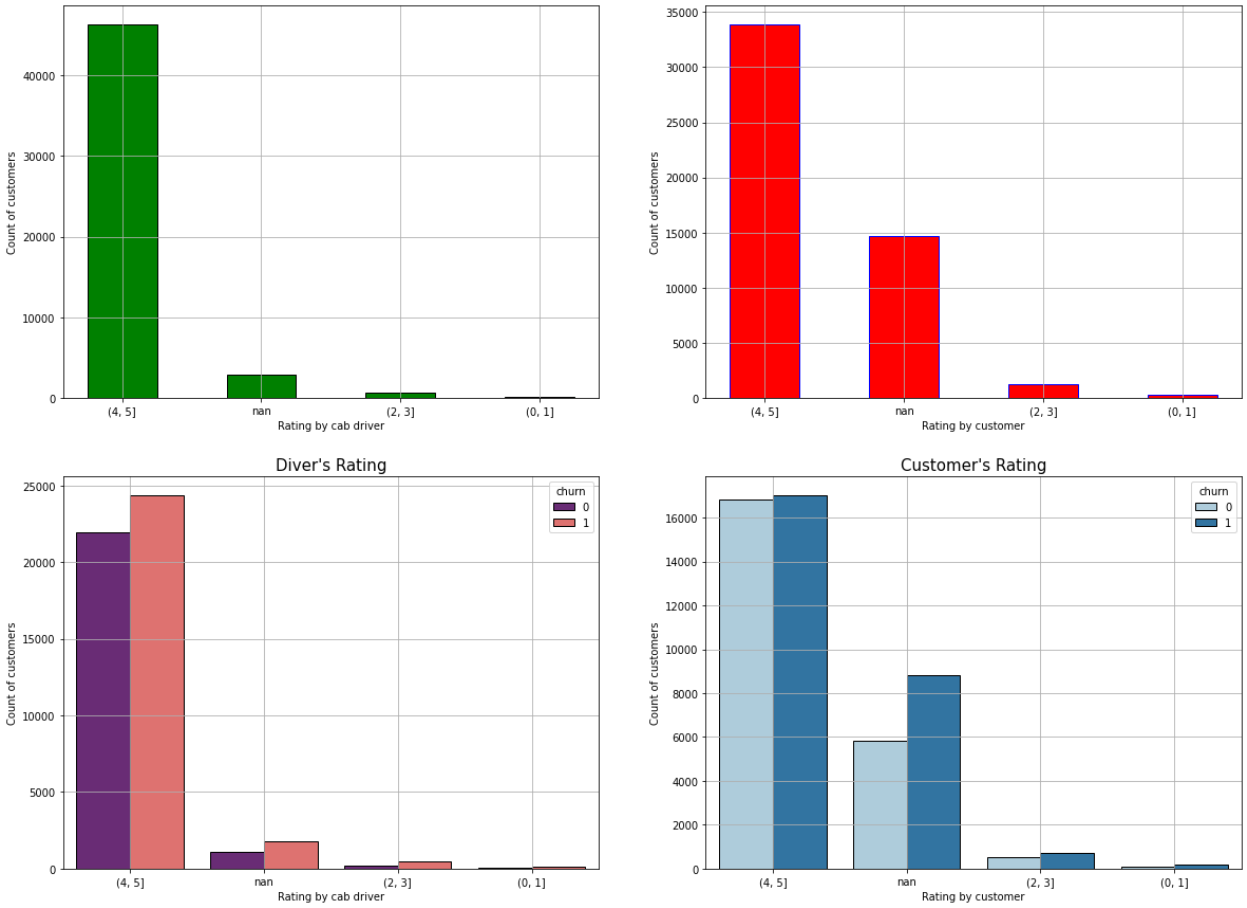
The above line graph shows that Signing up for ride sharing services has been drastically decreased after inital months. Also, there is a small spike for 11th month for users who are still u sing
services.



53.54% of the customers have already been churned. This is not an ideal situation for any type of organization. Due to this the company will face a increased depth if not acted in time. Thus this analysis will prove critical.



This gives us an insight that less customers are preferring luxurious cars than the luxurious ones whether they are leaving the services or not.

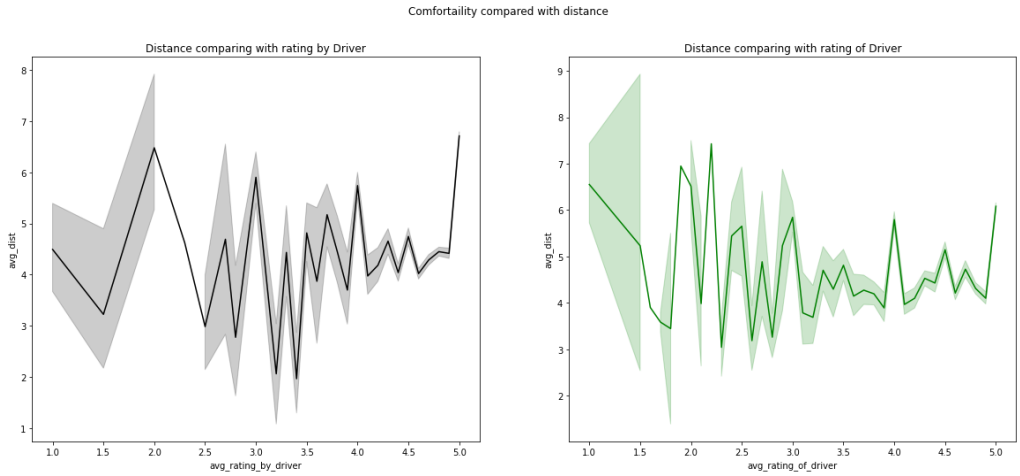


Feedback is the most lethal weapon for a company to find a solution for n number of problem statements.

1. As we can see in the above subplots, reviews from customer have a major missing value around 15,000, which might lead to an un-answered critical task that might need an immediate handling.
2. Company has to come up with unique User Interface such as reward-based feedback/rating system. In which customers and drivers will get reward if they report a grey area or area of improvement.
3. This reward base system will help to further enforce the relation between customer - service provider - driver which in the end reduces the churn rate.

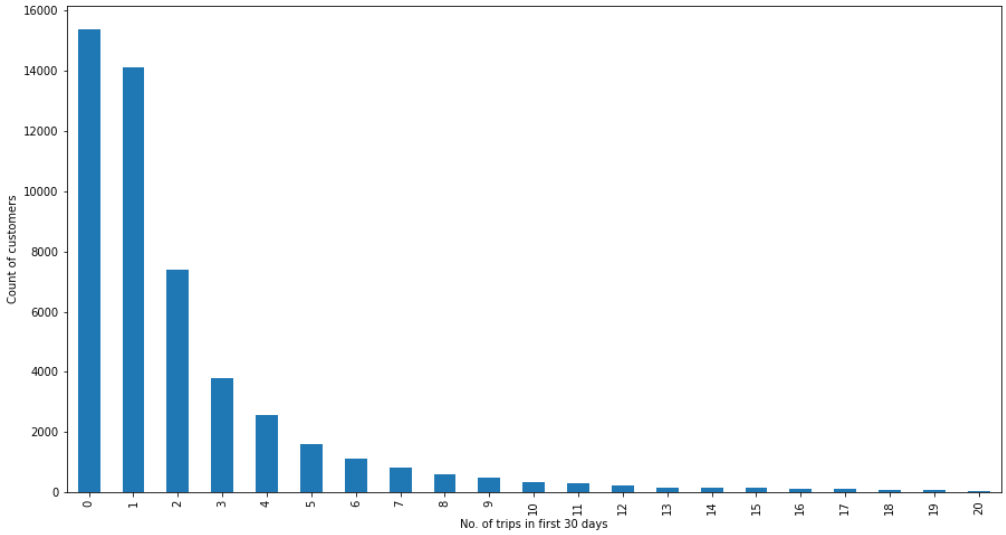
r_by_driver	(0, 1]	(2, 3]	(4, 5]	nan
r_of_driver				
(0, 1]	33.91	90.70	1232.07	320.84
(2, 3]	13.82	286.12	6012.02	623.97
(4, 5]	292.04	1402.96	175811.49	7885.48
nan	473.12	1887.66	86343.31	7131.82

The pivot table showing rating by driver as column and rating of driver as row with average distance as the values.



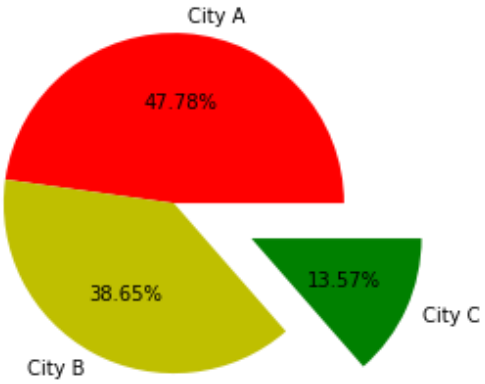
Comfortability compared with distance:

1. When Comfortability is compared with distance, we inferred that the distance increases the comfortability of customers as well as of drivers.
2. When we plot a graph of avg-distance travelled by a customer with its rating we can see a sudden rise in trend in rating between 1.5-3.
3. This indicates us that as average distance of a customer increases, satisfaction of customer decreases.
4. At a certain point it might occur that customer is not all satisfied with services which again gives us a platform to keep entertaining customers so they might not churn in future.

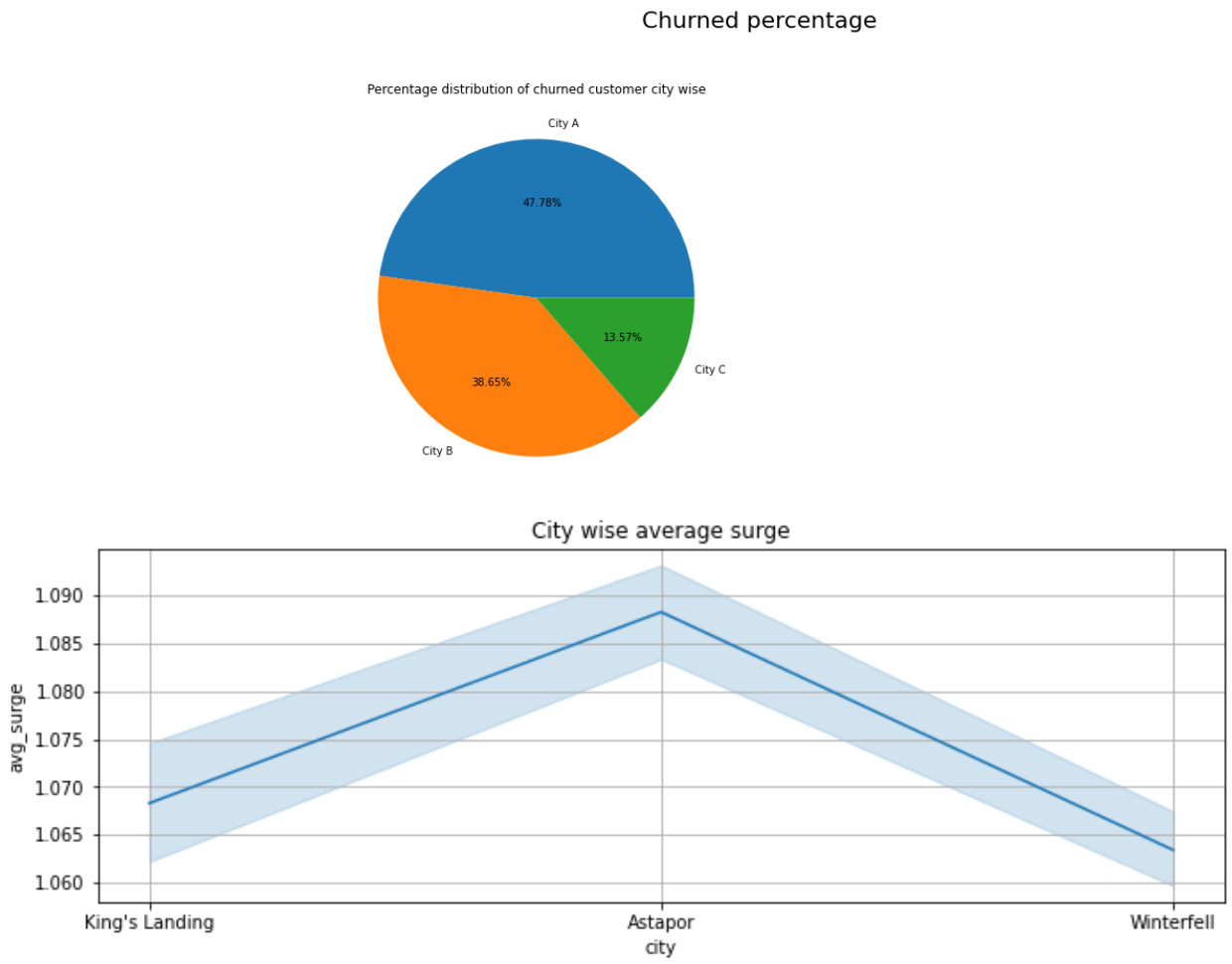


After registering to the ride share services, in the intial 30 days around 15400 number of customers have made 0 trips and around 14100 have booked services only once.

This graph shows the count of customers who have made less number of trips in the initial 30 days only and they are more likely to be churned.

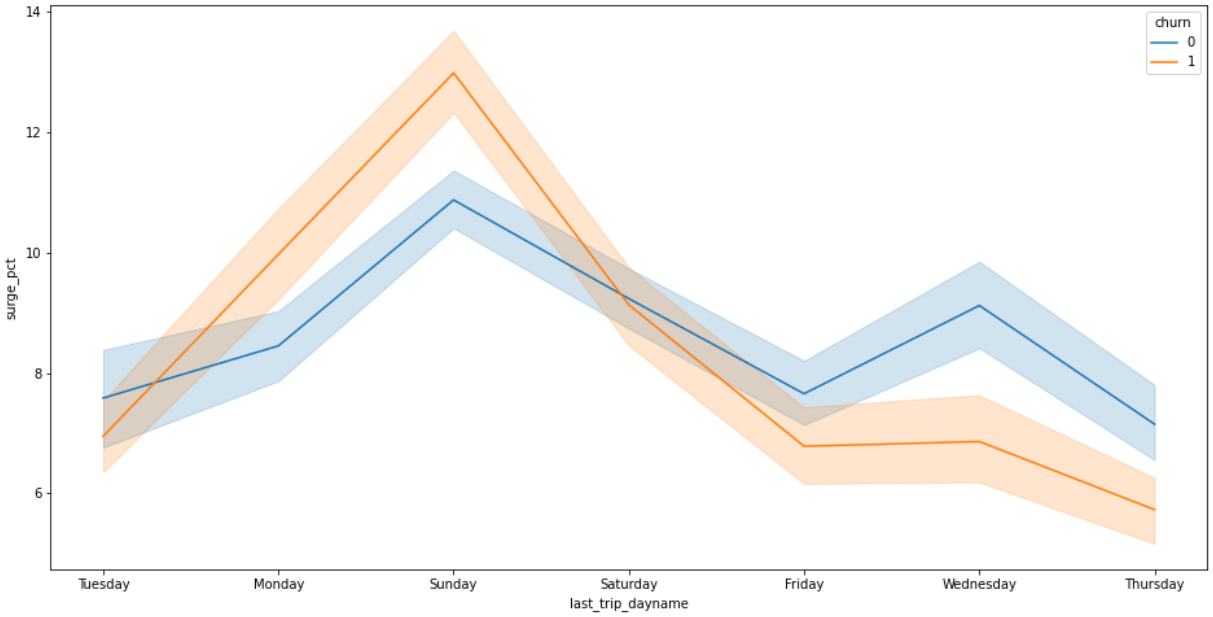


Most of the customers who are not using services are from City A.
i.e. City A should be the target area for promoting ride share services by providing some offers, cashbacks and improving overall customer experience.



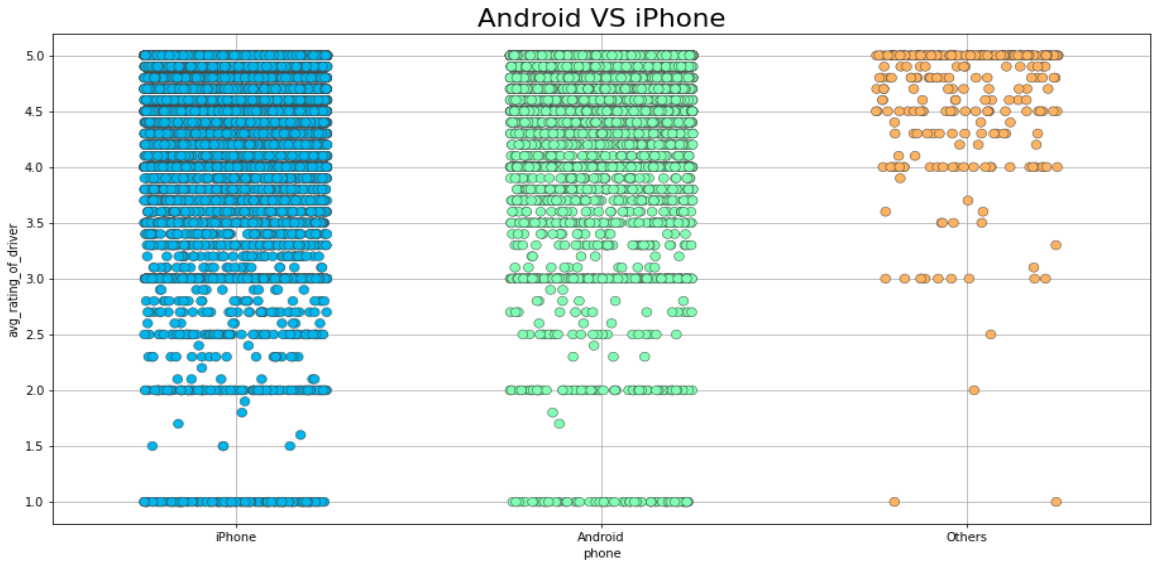
Above subplot gives us inference on percentage wise distribution of churned customers in each city.

1. City A have highest percentage of churn rate which is around 47.78% followed by city B 38.65% and city C 13.57%.
2. According to the data we should focus on city A to decrease churn rate by providing new offers and promotion.
3. While comparing pie chart churned customers and line plot of average surge with respect to city we can say that surge is not the only parameter affecting churn rate of customer.
4. As churn rate of city A is higher we can diverge the drivers of city A to city B as average surge is higher compared to other cities.
5. When drivers are diverged it will reduce the surge on city B and flow of customer will increase and top of it, we can even assure drivers of more rides to earn money.



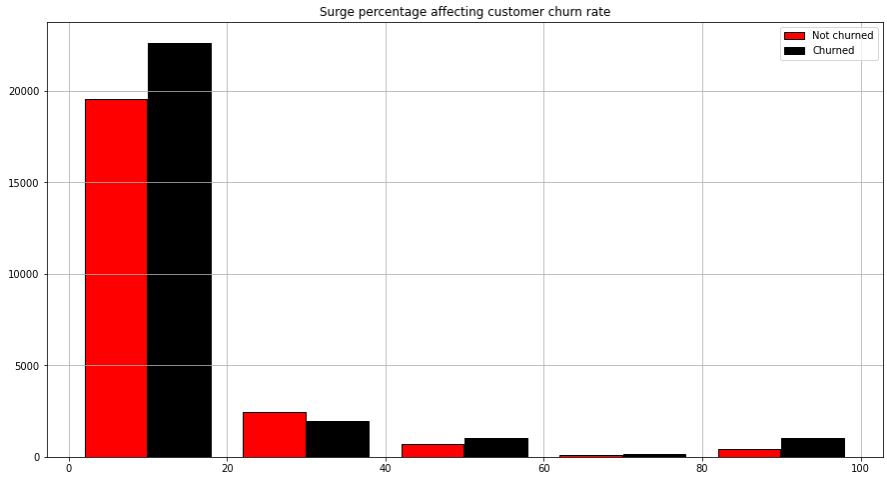
Surge_pct is higher for the customers on weekends is higher than on weekday.

1. This shows the normal trend that the Surge pricing is higher on weekends due to increase in demand of more drivers in an area. i.e., when riders in a given area are more than available drivers.
2. This encourages more drivers to serve the busy area over time and shifts rider demand, to maintain reliability and restore balance.
3. This helps drivers to earn incentive.
4. One of the inferences we can deduce that the customers who are not churned are actually facing higher surge when compared to the customers who have churned.
5. This shows us an area of improvement where we can provide extra efforts for loyal customer so in feature so they have same affinity toward this brand.



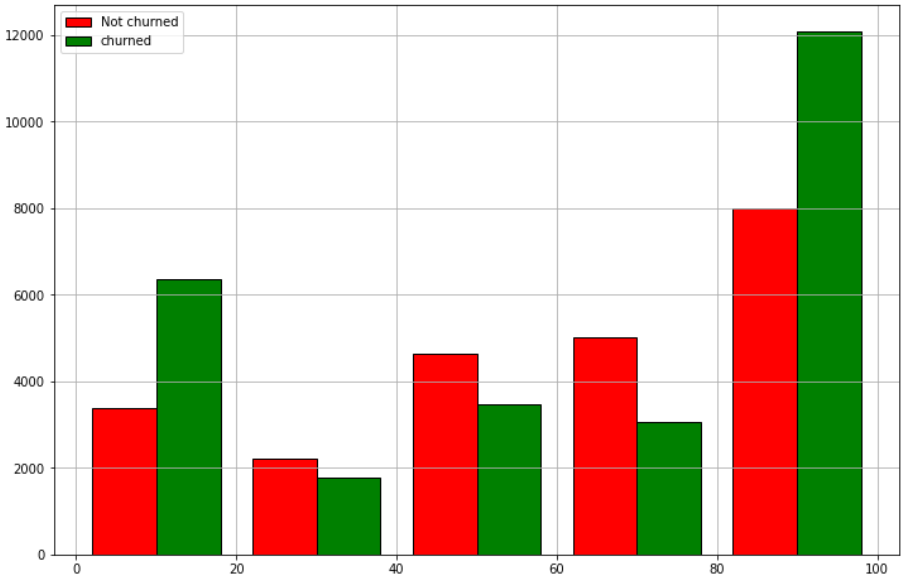
Phone feature in our dataset gives us information about the platform used by customers this company is using.

1. We find many missing values in this independent variable.
2. Initially we decided to fill these null values with the most repetitive platform in the data.
3. At the end of the day after researching about cab company platforms we get to know that, even if customers don't have an android or iPhone or any device/smartphone customers can book a cab as long as they have a internet connection and a browsers.
4. Internet and browser are the basic features provided by and cell phone manufacture.
5. Cab company provides a mini website suitable for all types of browsers usable in mobile phones.
6. This graph let us know there are very less awareness in customer, so company should also focus on this in case of emergency.
7. As we all know iPhone is all about is services and luxury but, as we can see in the above plot that span on iPhone customer is rating lies in a range of 3-5 which is a wider range of rating compared to Android.
8. There might be a possibility that iPhone customers are facing issues or bugs while booking a ride or facing issues while travelling.



This plot revels a static truth about churn behaviour of customers. ['surge_pct'] plays an important role for customers.

1. As we can see in the above graph, major chunk of customers lies in range of 0 to 20 of ['surge_pct'].
2. Customers churning between 0-20 range might have found a beter cab service with low surge percentage.
3. Customer churning rate is higher in range of 80-100.
4. There are also any customers who did not did not churn certain rise in ['surge_pct'] show's that they have brand affinity and stays loyal to this can service. For them we can provide various discounts/offers so we can retain these customers.



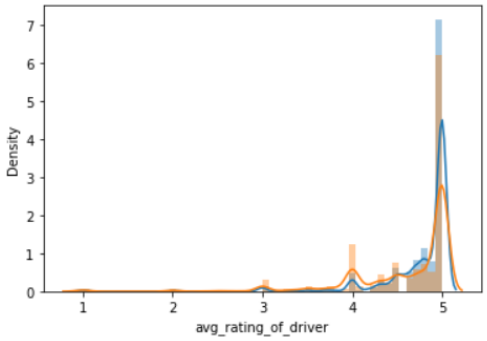
As we have proved that ['surge_pct'] is playing an important task while predicting the behavior of customers. But from the above bar graph we can say that ['weekday_pct'] is also a parameter which is affecting churn rate a lot, ['weekday_pct'] is also prove to be critical to define the churn rate and customer behaviors. as surge increases on the week days customer tends to lose patience and more likely to churn.

Treating Null Values of the Data:

Check the distribution of columns, skewness and kurtosis of it.

```
Skewness of the avg_rating_by_driver:  -4.128909161682118
Kurtosis of the avg_rating_by_driver:  24.228354360460248
/usr/local/lib/python3.7/dist-packages/seaborn/distributior
  warnings.warn(msg, FutureWarning)

Skewness of the avg_rating_of_driver:  -2.4284849281100045
Kurtosis of the avg_rating_of_driver:  8.137954307885723
/usr/local/lib/python3.7/dist-packages/seaborn/distributior
  warnings.warn(msg, FutureWarning)
```



Both of the distributions are left skewed.

Now let's try to impute the missing values in these features without changing the distribution much. The avg_rating_of_driver and avg_rating_by_driver are ordinal categorical variables ranging from 0 to 5.

With Iterative Imputer the avg ratings were going beyond the limit of 5 and also, we noticed -ve values in the result. Therefore, instead of using iterative imputer we are using median.

After EDA, we found the anomaly that there are customers who have signed up with the app even after they have taken last_trip.

Pre-Processing:

Normality Test:

Various statistical methods used for data analysis make assumptions about normality, including correlation, regression, t-tests, and analysis of variance. Central limit theorem states that when sample size has 100 or more observations, violation of the normality is not a major issue. Although for meaningful conclusions, assumption of the normality are still tested.

There are two main methods of assessing normality: Graphical and numerical.

Anderson Darlington Test:

The test is a modified version of a more sophisticated nonparametric goodness-of-fit statistical test called the Kolmogorov-Smirnov test. A feature of the Anderson-Darling test is that it returns a list of critical values rather than a single p-value. This can provide the basis for a more thorough interpretation of the result. By default, the test will check against the Gaussian distribution (dist='norm').

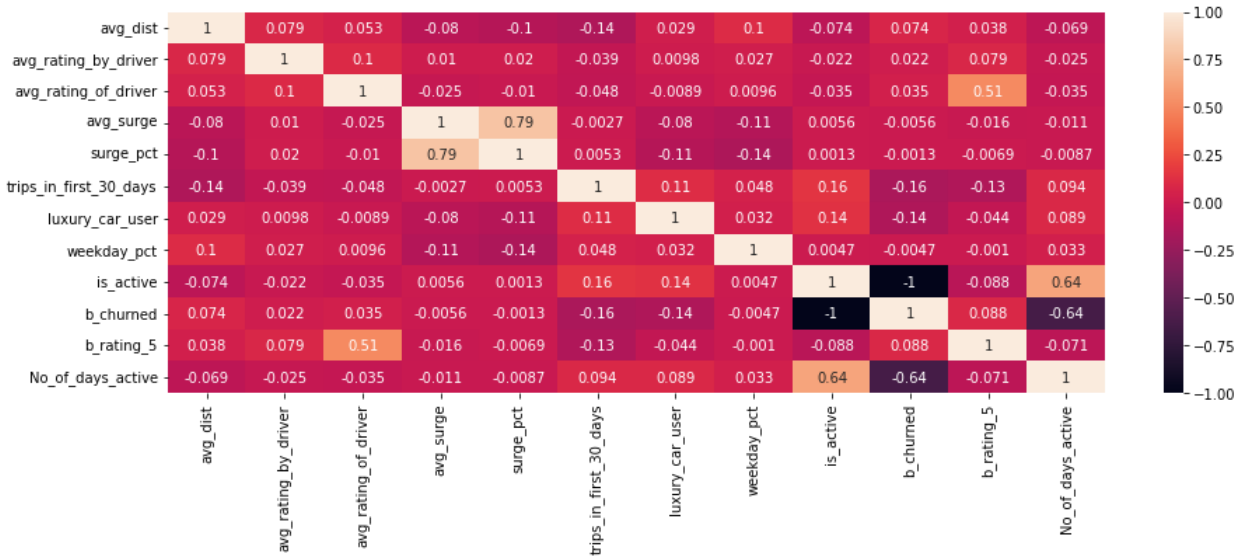
H0: Data is Normally Distributed.

Ha: Data is not Normally Distributed.

Observation:

All the variables in the data frame does not have normality hence rejecting the null hypothesis of Anderson Darlington Test.

Numerical-Numerical Analysis:



Conclusion from Numerical-Numerical feature analysis:

1. The avg_surge and surge_pct are showing maximum positive correlation of 0.79.
2. Others features are not highly correlated to each other.

T-Test (Numerical- Categorical Analysis)

H0: The distributions of churned customers and customers who are continuing services DO NOT differ from each other on basis of the feature selected.

Ha: The distributions of churned customers and customers who are continuing services differ from each other on basis of the feature selected.

Conclusion from t-test independent:

weekday_pct,surge_pct and avg_surge came out to be insignificant for prediction of churn. As with respect to churn,p values are larger and the distributions are overlapping with each other. Therefore, on basis of these three features it is difficult to predict churn. But we will still use RFE and SFS to see the significant no of features.

All the other features are significant for prediction of churn as p value<0.05 and there is more separation between the distributions wrt to churn.

CHI SQUARE:

H0: The groups are independent

Ha: The groups are dependent on each other.

All the three categorical variables like City ,Phone and Luxury_car_User are significant in prediction of Churn.

Preparation of Data:

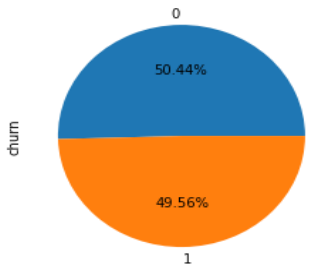
Transformation:

We will be performing transformation like Log, Sqrt, Reciprocal, Square, Exponential and Power Transformation and choose the best performing transformation.

	Initial skewness of data	Log	Sqrt	Reciprocal	Square	Exponential	PowerTransform
avg_dist	3.500755	0.535933	1.328133	NaN	43.256141	197.757427	0.010223
avg_rating_by_driver	-4.149630	-6.107514	-5.367919	11.795030	-2.850634	-1.621381	-0.881958
avg_rating_of_driver	-2.758217	-4.157066	-3.614995	9.062542	-1.896859	-1.113590	-0.786610
avg_surge	6.995161	4.859109	5.204415	-3.054548	18.601447	186.929681	1.257356
surge_pct	3.163954	1.179539	1.702115	NaN	4.784836	5.526723	0.882423
trips_in_first_30_days	5.473589	0.757468	1.071898	-0.903914	56.935132	197.757427	0.094733
weekday_pct	-0.476946	-1.404823	-1.058136	NaN	0.080791	0.666655	-0.740668
churn	0.017696	0.017696	0.017696	0.017696	0.017696	0.017696	0.017696
No_of_days_active	0.517725	-1.213170	-0.419657	NaN	2.256899	NaN	-0.378436

As we are getting better results from Power Transform we will select Power Transformation.

Checking Imbalance in the data:



The target variable is pretty balanced. So we will not use any oversampling or under sampling techniques.

Base Model:

Logit Regression Results

Dep. Variable:	churn	**No. Observations:**	39108
Model:	Logit	**Df Residuals:**	39094
Method:	MLE	**Df Model:**	13
Date:	Sun, 24 Apr 2022	**Pseudo R-squ.:**	0.3587
Time:	12:58:32	**Log-Likelihood:**	-17382.
converged:	True	**LL-Null:**	-27106.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.5706	0.030	19.222	0.000	0.512	0.629
avg_dist	0.0069	0.014	0.495	0.620	-0.021	0.034
avg_rating_by_driver	0.0295	0.014	2.100	0.036	0.002	0.057
avg_rating_of_driver	0.0144	0.014	1.049	0.294	-0.012	0.041
avg_surge	0.2448	0.042	5.889	0.000	0.163	0.326
surge_pct	-0.2999	0.043	-7.013	0.000	-0.384	-0.216
trips_in_first_30_days	-0.3528	0.014	-24.707	0.000	-0.381	-0.325
weekday_pct	0.0861	0.014	6.150	0.000	0.059	0.114
No_of_days_active	-1.7312	0.019	-93.163	0.000	-1.768	-1.695
city_City B	0.4182	0.030	13.725	0.000	0.358	0.478
city_City C	-0.5690	0.037	-15.248	0.000	-0.642	-0.496
phone_Others	-0.6700	0.153	-4.366	0.000	-0.971	-0.369
phone_iPhone	-0.5283	0.029	-17.959	0.000	-0.586	-0.471
luxury_car_user_True	-0.4143	0.028	-14.783	0.000	-0.469	-0.359

Interpretation:

LOG(ODDS)=0.5706+0.0295(avg_rating_by_driver)+0.0144(avg_rating_of_driver)+0.2448(avg_surge)-0.2999(surge_pct)

Checking Multicollinearity

	Vif
const	4.917784
avg_dist	1.046031
avg_rating_by_driver	1.152403
avg_rating_of_driver	1.073180
avg_surge	9.215963
surge_pct	9.799270
trips_in_first_30_days	1.218072
weekday_pct	1.033068
No_of_days_active	1.146664
city_City B	1.172719
city_City C	1.213334
phone_Others	1.022583
phone_iPhone	1.045065
luxury_car_user_True	1.036601

Since, both the VIF values for avg_surge and surge_pct are almost same. We checked value of skewness for avg_surge and surge_pct. The skewness of avg_surge is higher so we will remove avg_surge and check multicollinearity between all the independent variables again. Avg_surge is removed to eliminate multicollinearity. Avg_surge is removed to eliminate multicollinearity.

MODEL FOR CHURN PREDICTION

We have performed various supervised algorithms on our data set they are:

1) Logistic Regression being the base model 2) Decision Trees 3) Random Forest 4) K Neighbors Classifier 5) Light GBM The project then briefly describes the chosen algorithms and illustrates the results for churn.

1. Logistic Regression

Logistic regression is a classification algorithm often used as baseline model to set a benchmark. It suits well where our label is binary and categorical. LR uses predictive analysis to describe the trade-off or relationship between a dependent binary variable and a set of independent variables. One drawback is that it doesn't handle collinearity and requires a large sample. It doesn't need the data to be linear in nature, it handles nonlinear relationships with the use of nonlinear log loss transformations.

Here we get an accuracy of approximately 80% on test data, recall of 77% and precision of 82%. We are more interested in recall than accuracy or precision as we need to correctly identify churners and stop then from churning by offering various marketing offers. We ran LR for different solvers along with regularization parameters to tune the model thus generated. The ROC-AUC score of test data is 0.88.

```
Accuracy of train 0.8014611872146119
Confusion matrix of train
 [[11500  2365]
 [ 3070 10440]]
ROC-AUC score of train 0.8847450099737797

Accuracy of test 0.7968124094434501
Confusion matrix of test
 [[4860 1002]
 [1382 4489]]
ROC-AUC score of test 0.8820545428521468
```

	precision	recall	f1-score	support
0	0.79	0.83	0.81	13865
1	0.82	0.77	0.79	13510
accuracy			0.80	27375
macro avg	0.80	0.80	0.80	27375
weighted avg	0.80	0.80	0.80	27375

2. Decision Trees:

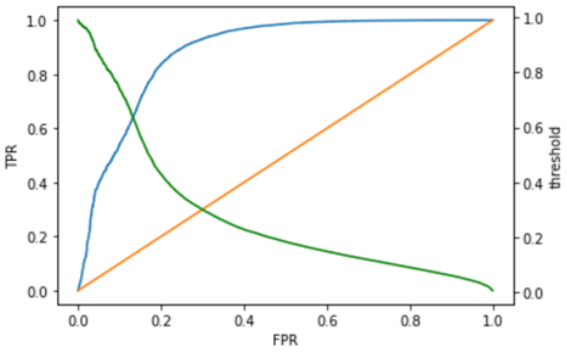
The basic structure of the decision tree consists of internal nodes and leaf nodes, where internal nodes checks certain conditions and splitting points and creates branches to reduce entropy. Leaf nodes that represent label values in our case: Exited status as 0 or 1. It supports categorical as well as continuous data.

Here we get an **accuracy** of approximately **85.24% on test data**, recall of 80% and precision of 83%. We are more interested in recall than accuracy or precision as we need to correctly identify churners and stop then from churning by offering various marketing offers. The ROC-AUC score of test data is 0.869.

```
Accuracy of train 0.9708858447488584
Confusion matrix of train
 [[13459   406]
 [  391 13119]]
ROC-AUC score of train 0.9975463087405971

Accuracy of test 0.8523821699480099
Confusion matrix of test
 [[5005  857]
 [ 875 4996]]
ROC-AUC score of test 0.86934349227137
```

	precision	recall	f1-score	support
0	0.94	0.80	0.87	5862
1	0.83	0.95	0.88	5871
accuracy			0.87	11733
macro avg	0.88	0.87	0.87	11733
weighted avg	0.88	0.87	0.87	11733



3. Random Forest:

This is an Ensemble based model. Ensembles are a divide-and-conquer approach used to improve performance. In RF multiple decision trees are implemented together. The crux is that multiple weak learners together can be a strong learner and give better results. The weak learner in random forest are decision trees and multiple decision trees are formed using sampled data, then RF takes the decision which is supported by majority of the trees. It is used to reduce overfitting of decision trees by not depending on just 1 tree.

The performance increases as **accuracy increased to 87%** in test data and recall increased to 79%. Also precision is 95%. This tree was constructed by taking 100 decision trees. The ROC-AUC score of test data is 0.946.

```
Accuracy of train 0.8842009132420091
Confusion matrix of train
 [[11184  2681]
 [  489 13021]]
ROC-AUC score of train 0.953959111374006

Accuracy of test 0.8759055654990199
Confusion matrix of test
 [[4657 1205]
 [ 251 5620]]
ROC-AUC score of test 0.946506505935849
```

	precision	recall	f1-score	support
0	0.95	0.79	0.86	5862
1	0.82	0.96	0.89	5871
accuracy			0.88	11733
macro avg	0.89	0.88	0.88	11733
weighted avg	0.89	0.88	0.88	11733

4. K Neighbors Classifier:

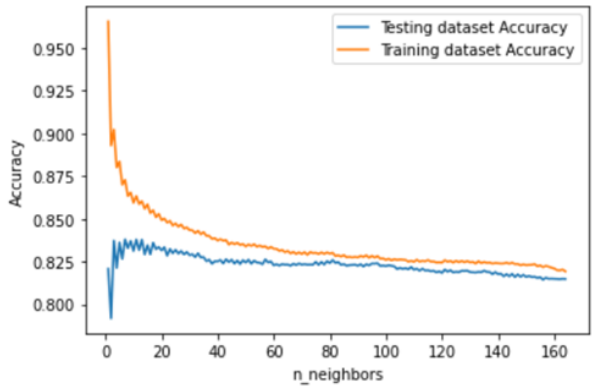
We have used the K Nearest Neighbors algorithm to see how good the model performs for this kind of data and the results are as follows:

```
Accuracy of train 0.8836529680365297
Confusion matrix of train
 [[12097  1768]
 [ 1417 12093]]
ROC-AUC score of train 0.9571084180408362

Accuracy of test 0.8362737577772096
Confusion matrix of test
 [[4905  957]
 [ 964 4907]]
ROC-AUC score of test 0.9067498121938289
```

	precision	recall	f1-score	support
0	0.82	0.85	0.84	5862
1	0.85	0.82	0.83	5871
accuracy			0.84	11733
macro avg	0.84	0.84	0.84	11733
weighted avg	0.84	0.84	0.84	11733

From the above result of **K neighbors,** accuracy **is 84% and recall is 85%.**



5. **Light GBM**:

Light GBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages: Faster training speed and

higher efficiency, Lower memory usage, better accuracy, Support of parallel, distributed, and GPU learning, Capable of handling large-scale data.

```
Accuracy of train 0.9085296803652968
Confusion matrix of train
 [[12017  1848]
 [  656 12854]]
ROC-AUC score of train 0.9710132682099221

Accuracy of test 0.8895423165430836
Confusion matrix of test
 [[4972  890]
 [ 406 5465]]
ROC-AUC score of test 0.9582216767751047
```

```
              precision    recall  f1-score   support

           0       0.93      0.84      0.88      5862
           1       0.86      0.93      0.89      5871

    accuracy                           0.89     11733
   macro avg       0.89      0.89      0.89     11733
weighted avg       0.89      0.89      0.89     11733

Accuracy score of tuned LightGBM model:  0.8892866274610074
```

The performance of **Light GBM** as **accuracy is 88%** in test data and **recall is 84%.** Also precision is 93%.

Also the feature importance is displaying below:

	imp
No_of_days_active	**66741.102761**
trips_in_first_30_days	**3005.468678**
weekday_pct	**1593.632632**
surge_pct	**1345.066890**
avg_rating_by_driver	**981.263748**
phone_iPhone	**545.987000**
luxury_car_user_True	**402.022270**
city_City B	**371.343377**
city_City C	**369.241437**
phone_Others	**35.904980**

The less important features are not contributing much to our entire model as they are not used for splits. So, we can drop these.

Inference:

With the important feature like No_of_days_active, trips_in_first_30_days, weekday_pct, avg_rating_by_driver, surge_pct, phone_iPhone, the model can reach a reasonable prediction possibility. We will select Random Forest since it has a Cohen's Kappa Score of **0.9417627**, Accuracy of Test as **0.85101849** and ROC-AUC score of test **0.9431922**.