Cab Surge Price Type Prediction using AI

Submitted By - ABHILASHA LODHA

July 18, 2021

Using Predictive Analytics to predict Surge Price Type

INTRODUCTION

With the upcoming cab aggregators and demand for mobility solutions, the past decade has seen immense growth in data collected from commercial vehicles with major contributors such as Uber and Ola to name a few.

There are loads of innovative data science and machine learning solutions being implemented using such data and that has led to tremendous business value for such organisations.

BUSINESS PROBLEM

XXX Cab Private Limited company is a cab aggregator service company. Their customers can download their app on smartphones and book a cab from any where in the cities the company operates in. They, in turn search for cabs from various service providers and provide the best option to their client across available options. They have been in operation for little less than a year now. During this period, they have captured surge_pricing_type from the service providers.

The business aim is to build a predictive model, which could help them in predicting the surge_pricing_type pro-actively. This would in turn help them in matching the right cabs with the right customers quickly and efficiently.

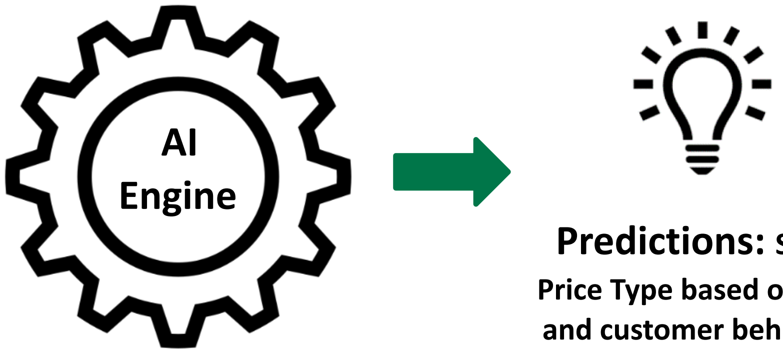
PROPOSED SOLUTION

- Deploy AI engine to predict surge price type
  - Analogous to a typical classification problem
  - Use existing cab trips & customer data to train the AI model
  - Potential algorithms that can be explored:
    - Logistic Regression
    - Tree-based Classification (Decision Trees)
    - Bagging Algorithms (Random Forest)
    - Boosting Algorithms (XGBoost)

Leveraging available input and target variables to make Surge Price Predictions

Input Variables	Definition
Trip_ID	ID for TRIP
Trip_Distance	The distance for the trip requested by the customer
Type_of_Cab	Category of the cab requested by the customer
Customer_Since_Months	Customer using cab services since n months; 0 month means current month
Life_Style_Index	Proprietary index created by XXX Cabs showing lifestyle of the customer based on their behavior
Confidence_Life_Style_Index	Category showing confidence on the index mentioned above
Destination_Type	XXX Cabs divides any destination in one of the 14 categories.
Customer_Rating	Average of life time ratings of the customer till date
Cancellation_Last_1Month	Number of trips cancelled by the customer in last 1 month
Var1, Var2 and Var3	Continuous variables masked by the company
Gender	Gender of the customer

Target Variable	Definition
Surge_Pricing_Type	Predictor variable can be of 3 types



Predictions: Surge Price Type based on trips and customer behaviour

KEY BENEFITS

- Automated solution for predicting surge price
- Accurate matching of right cabs with the right customers quickly and efficiently
- Avoiding loosing customers
- Mitigation of financial and business losses

Machine Learning Modelling Steps

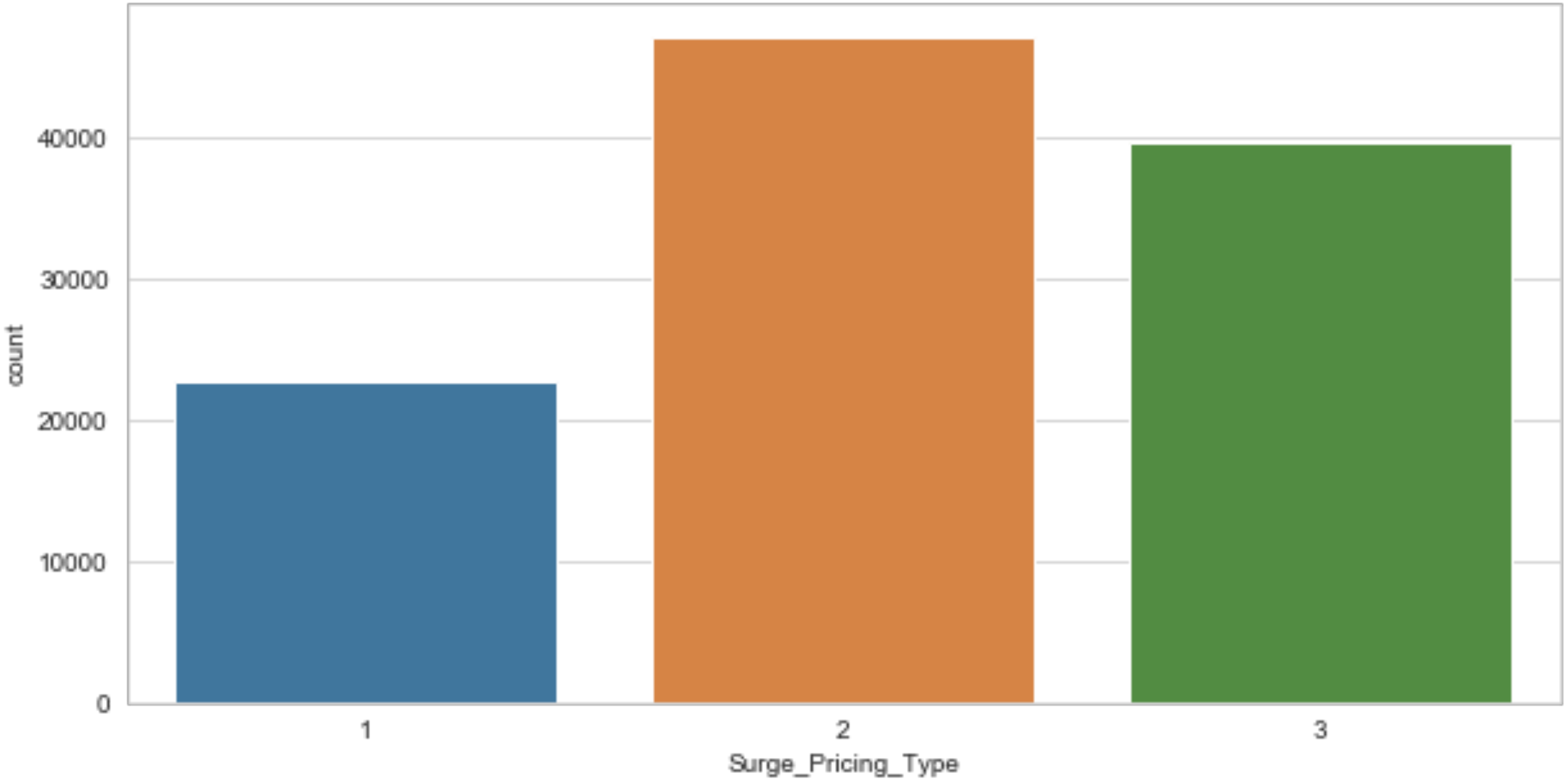
Steps

- Importing libraries and data
- Exploring data
- Checking for missing values
- Treating missing values by EDA
- Converting categorical columns to one-hot encoding
- Checking correlation & dropping highly correlated values
- Outlier detection & treatment
- Checking skewness of data
- Univariate, bivariate, multivariate analysis
- Scaling continuous values by Standard Scaler
- X and y assignment
- Train and test splitting (70-30)
- Training using LR, DT, RF, XGB
- Checking training accuracies
- Determining feature importance
- Prediction on Test data
- Evaluation metrics

Input Data Exploration and Understanding

- Trip_ID is not required in building up the predictive model and so can be dropped
- The dataset has no empty values for input variables Trip_Distance, Destination_Type, Customer_Rating, Cancellation_Last_1Month, Var2, Var3, Gender
- There are no empty values in Target variable Surge_Pricing_Type
- We have some missing values in Type_of_Cab which is a categorical column with values A,B,C,D and E. For the missing values (nan), a new category 'F' can be created
- Customer_Since_Months are the customers using the cab service since n months. Nan values in this column can thus be replaced with 0, indicating that they are the newbies to this cab service
- Life_Style_Index and Confidence_Life_Style_Index are the Proprietary indexes created by XXX Cabs showing lifestyle of the customer based on their behaviour. EDA can be performed to replace nan values
- Var1 is a continuous variable masked by the company. Again, need to perform EDA to replace the nan values

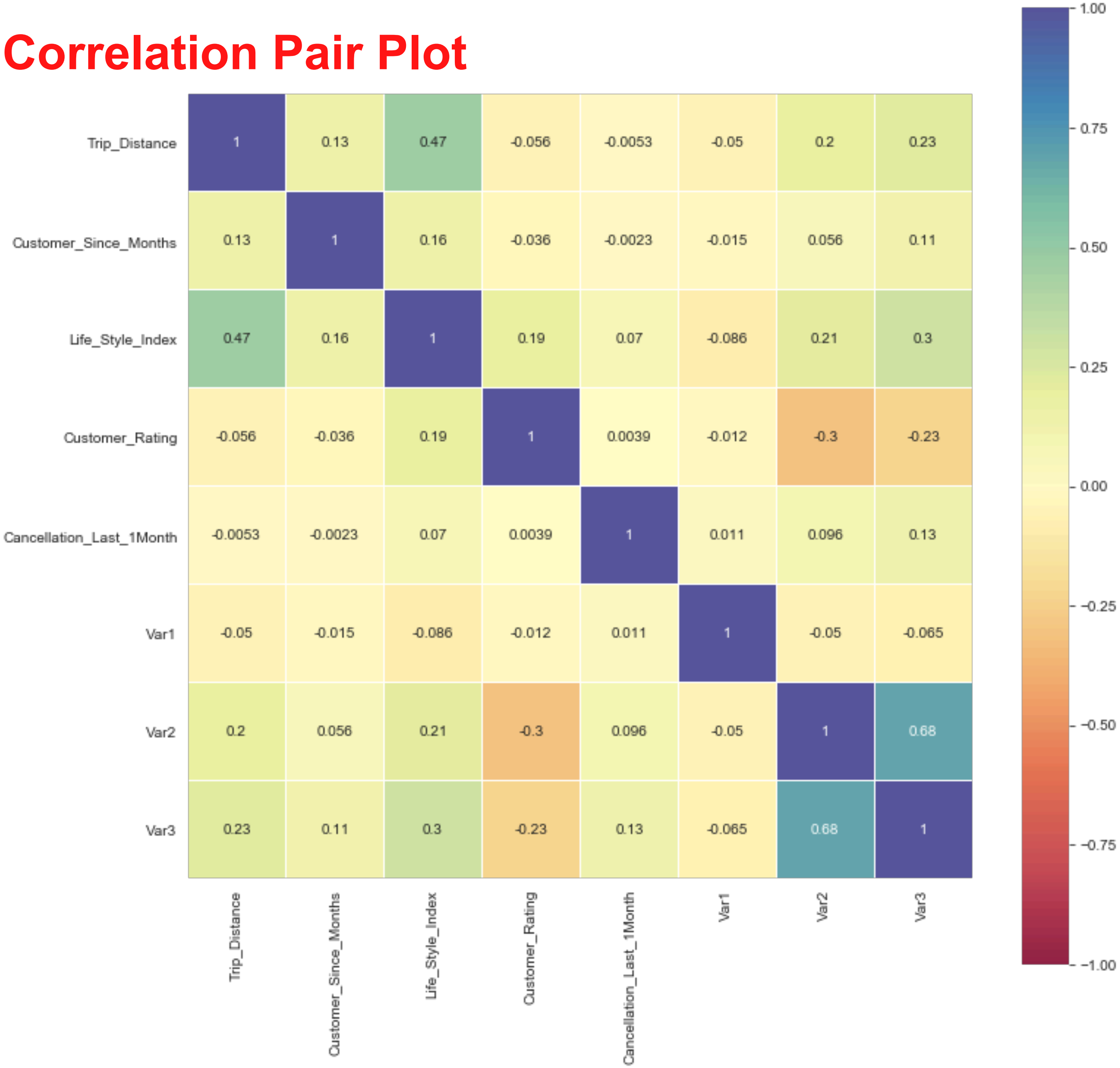
Target Variable (Surge Price Type) Data Exploration



INSIGHTS

Since there is not much difference between the target variable classes, sampling isn't required.

Correlation Pair Plot



	Trip_Distance	Customer_Since_Months	Life_Style_Index	Customer_Rating	Cancellation_Last_1Month	Var1	Var2	Var3
Trip_Distance	1	0.13	0.47	-0.056	-0.0053	-0.05	0.2	0.23
Customer_Since_Months	0.13	1	0.16	-0.036	-0.0023	-0.015	0.056	0.11
Life_Style_Index	0.47	0.16	1	0.19	0.07	-0.086	0.21	0.3
Customer_Rating	-0.056	-0.036	0.19	1	0.0039	-0.012	-0.3	-0.23
Cancellation_Last_1Month	-0.0053	-0.0023	0.07	0.0039	1	0.011	0.096	0.13
Var1	-0.05	-0.015	-0.086	-0.012	0.011	1	-0.05	-0.065
Var2	0.2	0.056	0.21	-0.3	0.096	-0.05	1	0.68
Var3	0.23	0.11	0.3	-0.23	0.13	-0.065	0.68	1

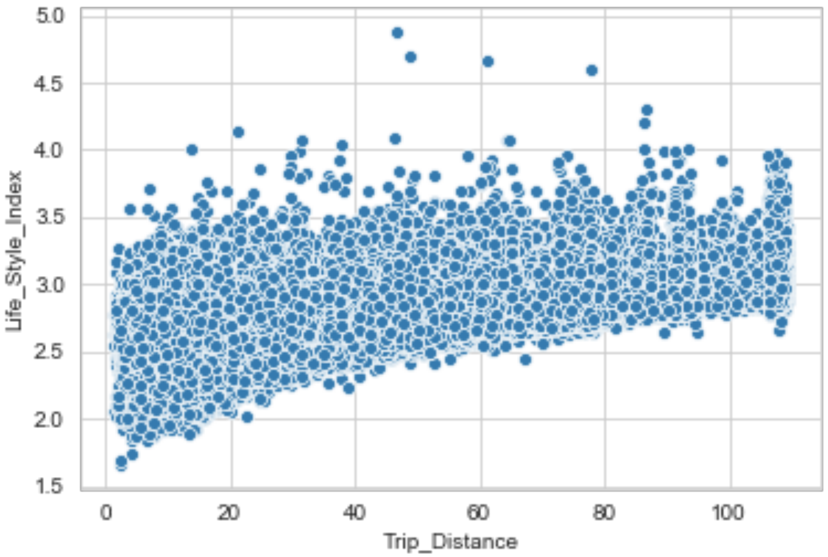
INSIGHTS

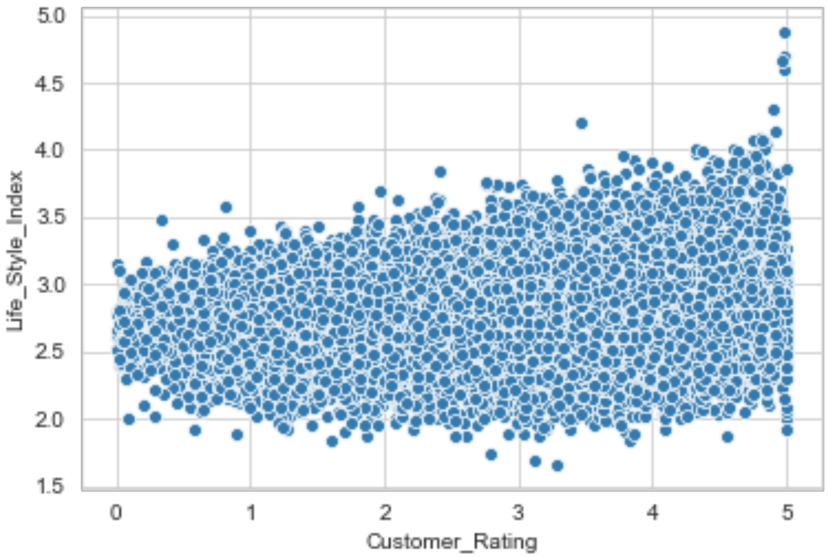
Correlation matrix values > 0.9 symbolises high correlation values between the independent features.

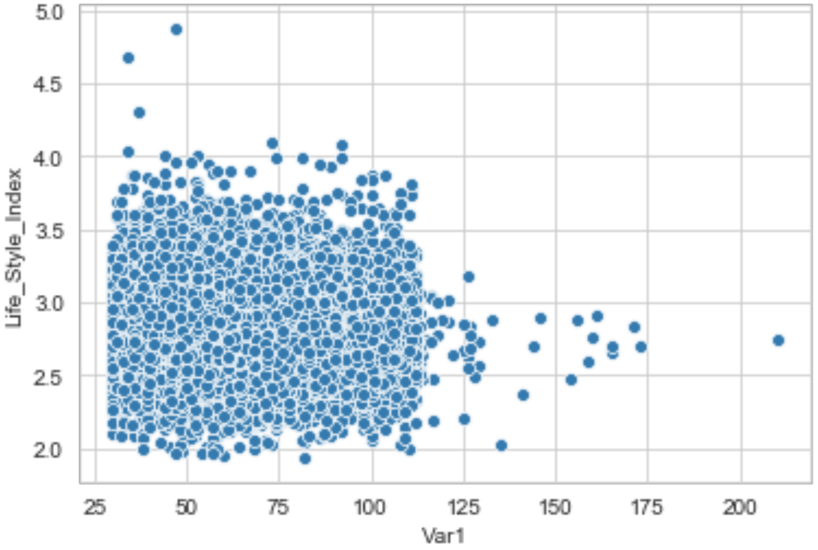
None of the variables are much correlated.

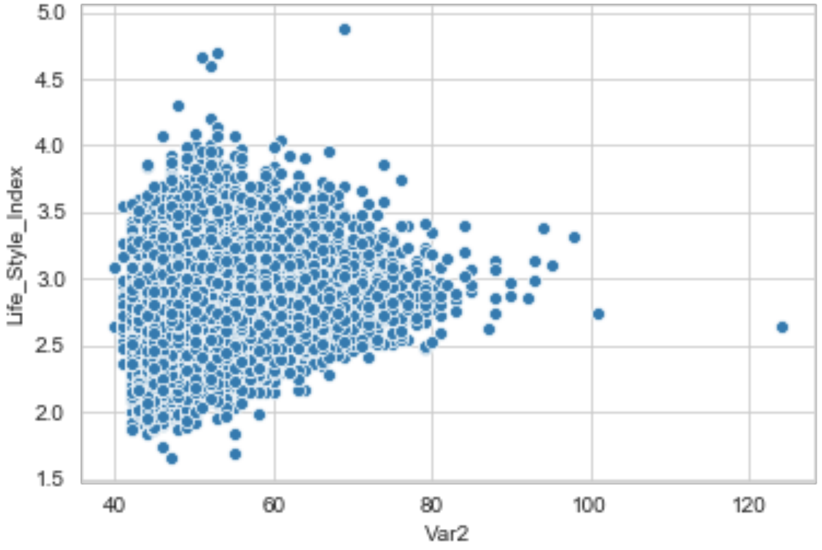
Var2 and Var3 are 68% correlated, no need of dropping any variable.

EDA on Life_Style_Index and Missing value treatment







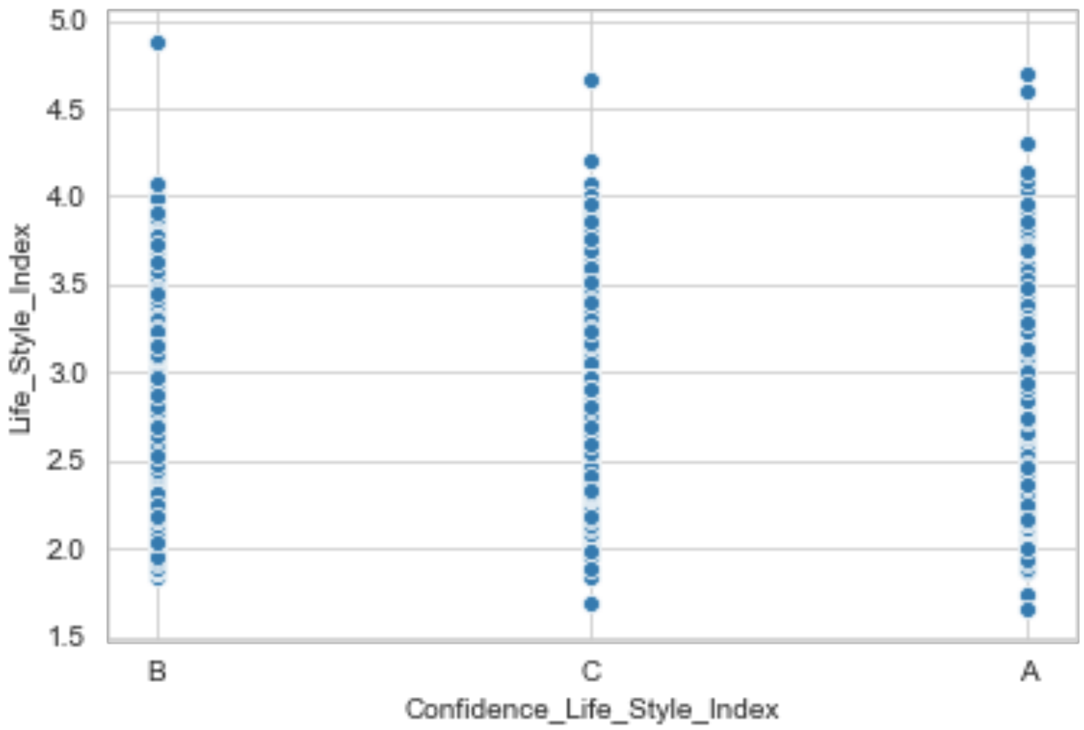


INSIGHTS on Life_Style_index

From these scatter plots, it can be seen that most of the values of Life_style_index are distributed between 2 to 3.5

For simplicity, we can replace nan values with mean i.e. 2.8

EDA on Confidence_Life_Style_Index and Missing value treatment



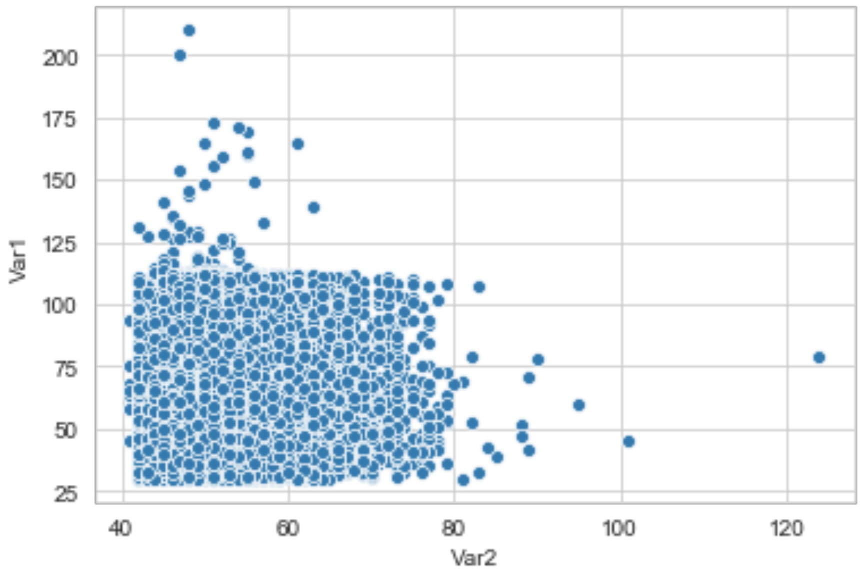
INSIGHTS on Confidence_Life_Style_index

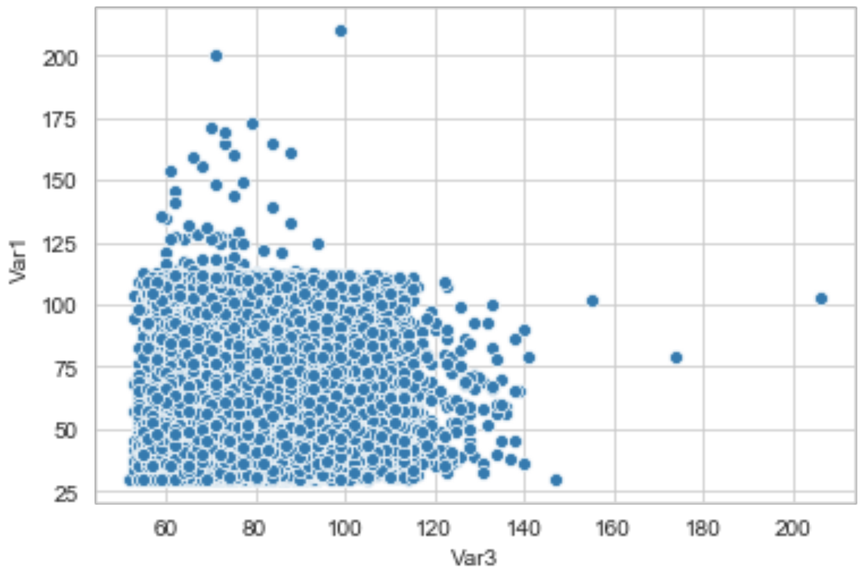
Confidence_Life_Style_index is a category column showing customer confidence based on Life_style_index .

From the scatter plot, it can be seen that most of the values of Confidence_Life_Style_index are randomly distributed with Life_style_index.

For simplicity, we can replace nan values randomly with A,B or C.

EDA on Var1 and Missing value treatment





INSIGHTS on Var1

From these scatter plots, it can be seen that most of the values of Var1 are distributed with Var2 and Var3 between 25 to 112

For simplicity, we can replace nan values with mean i.e. 64

EDA of Continuous Variables

Fig. : Trip_Distance

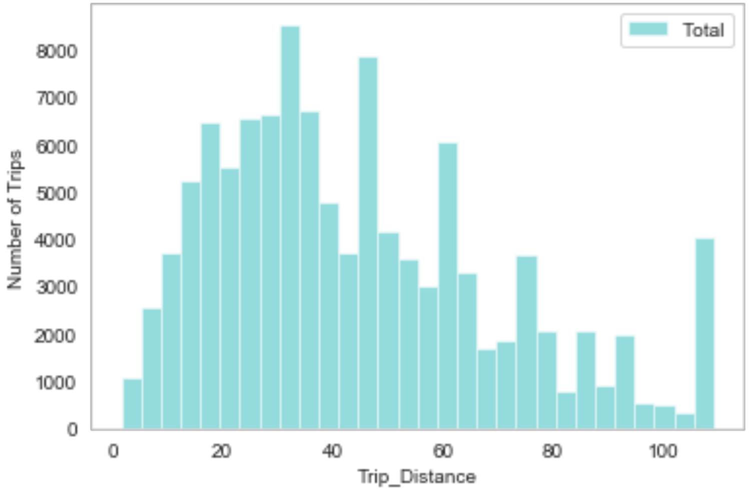


Fig. : Life_Style_Index

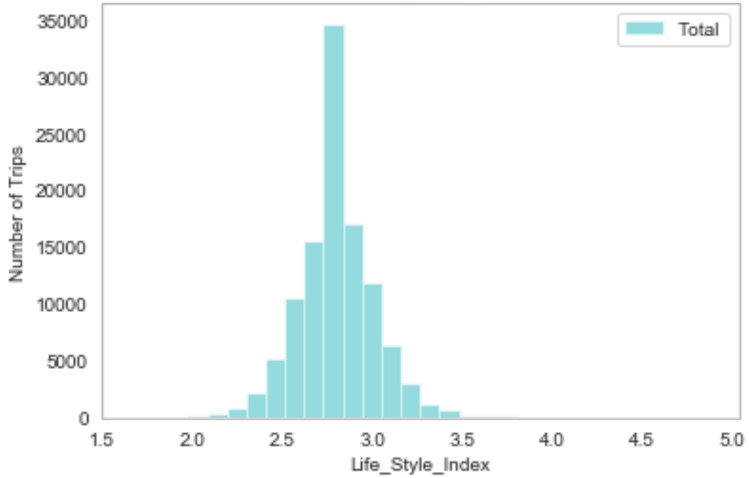


Fig. : Customer_Rating

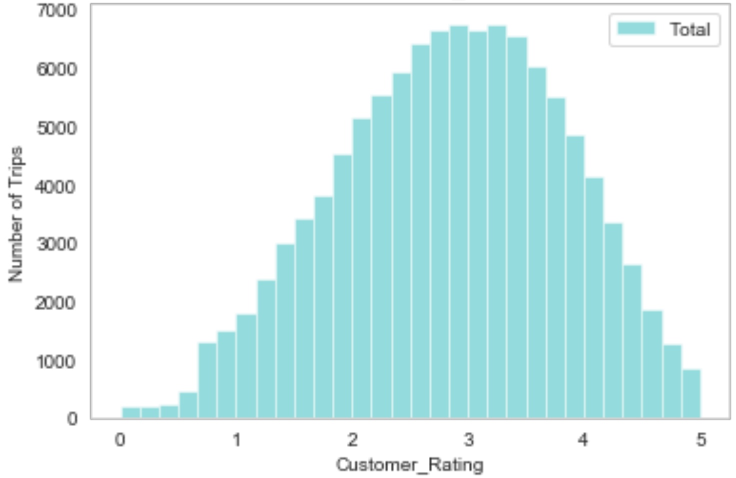


Fig. : Var1

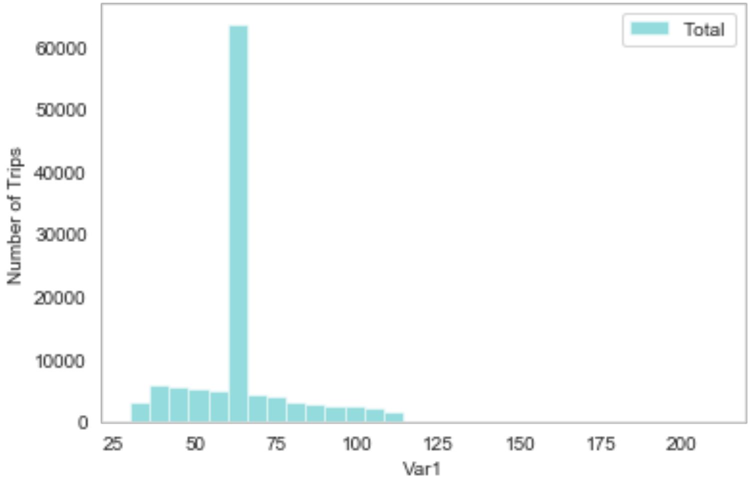


Fig. : Var2

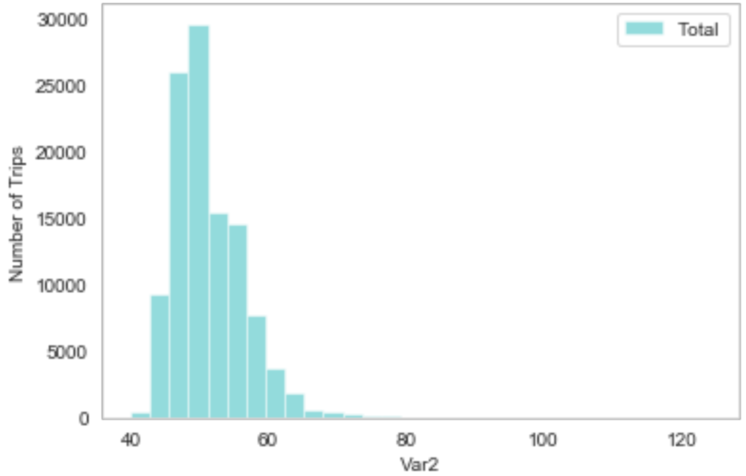
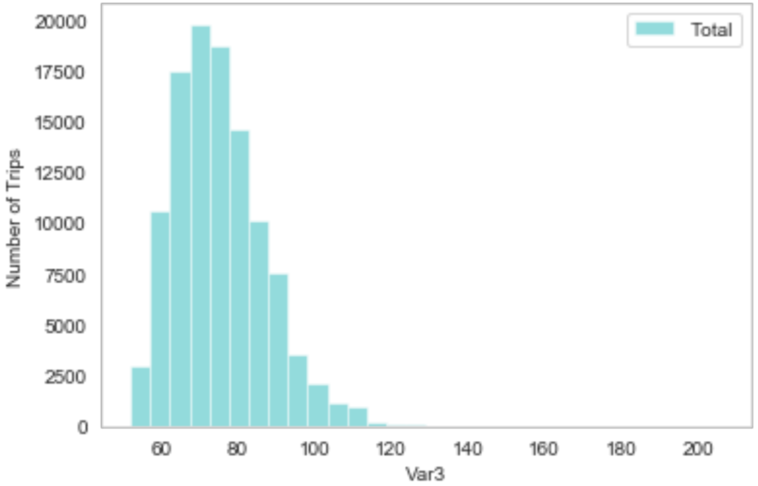
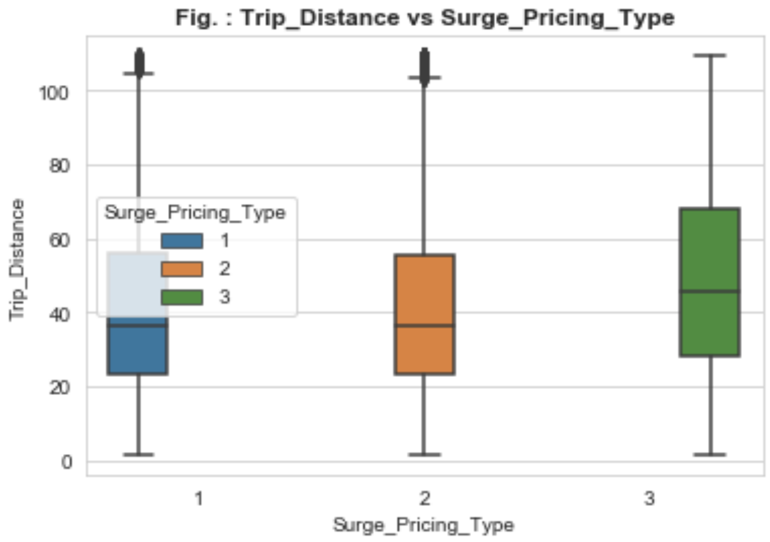
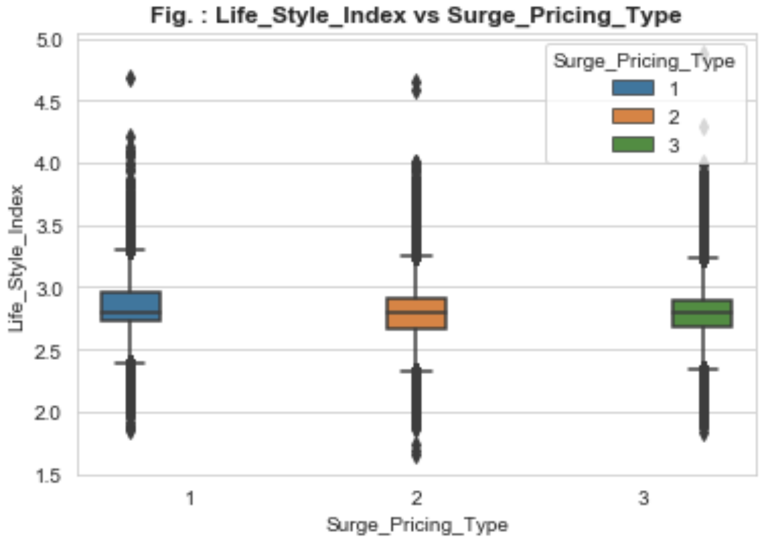


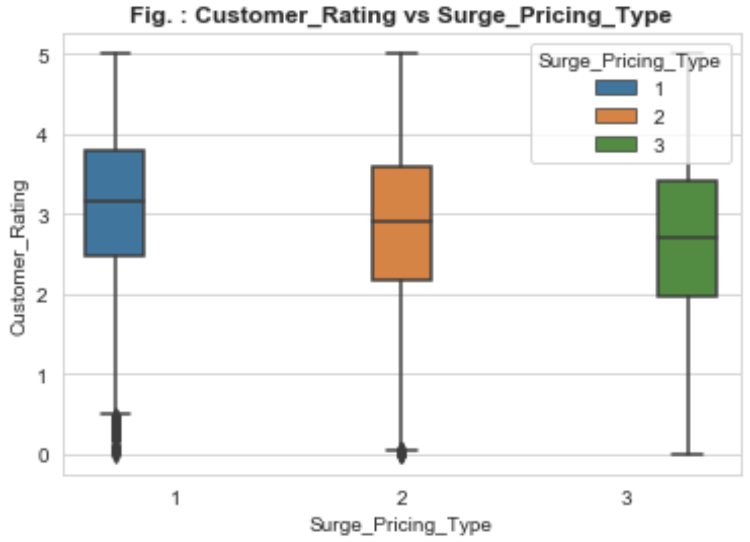
Fig. : Var3



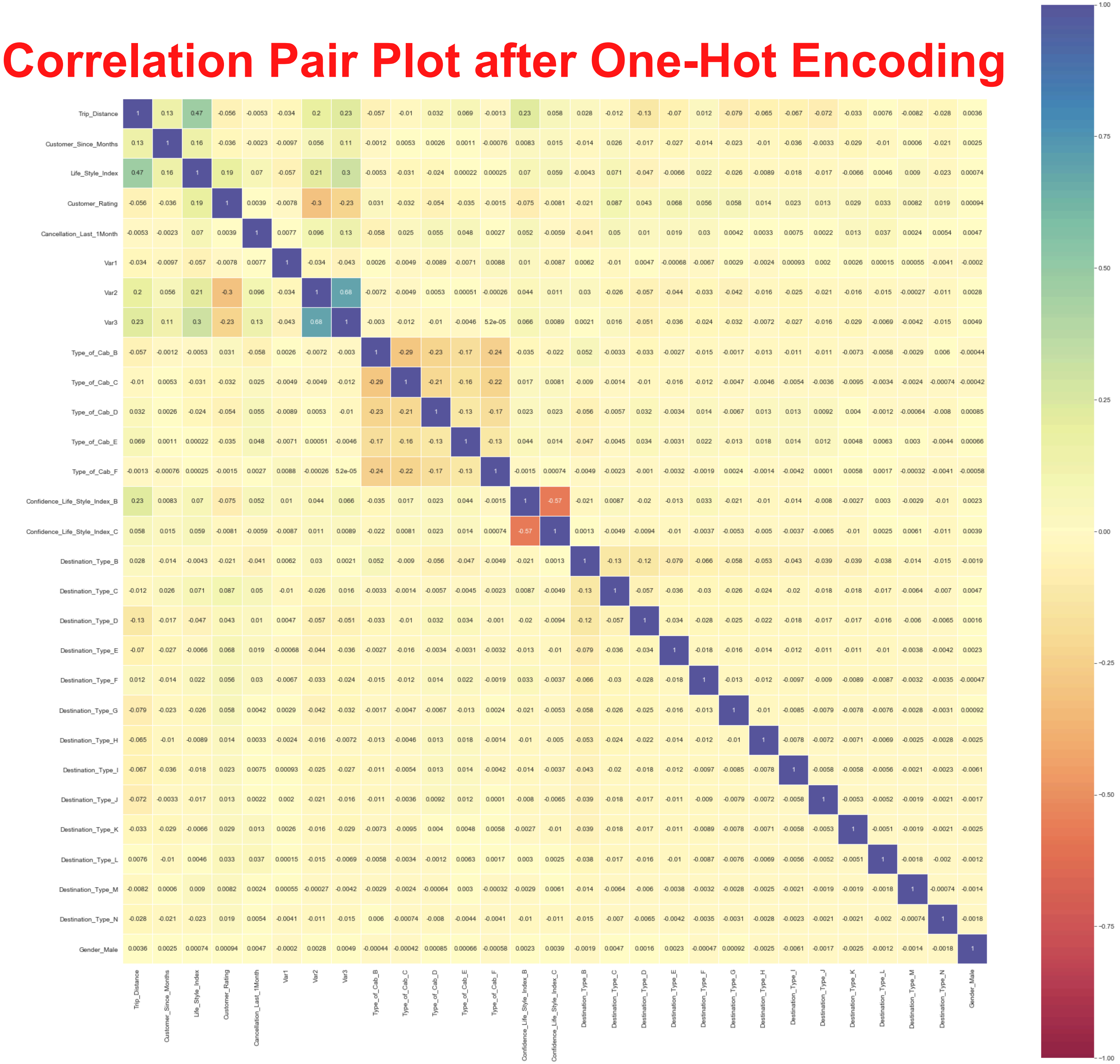
Bivariate Analysis using Box Plots



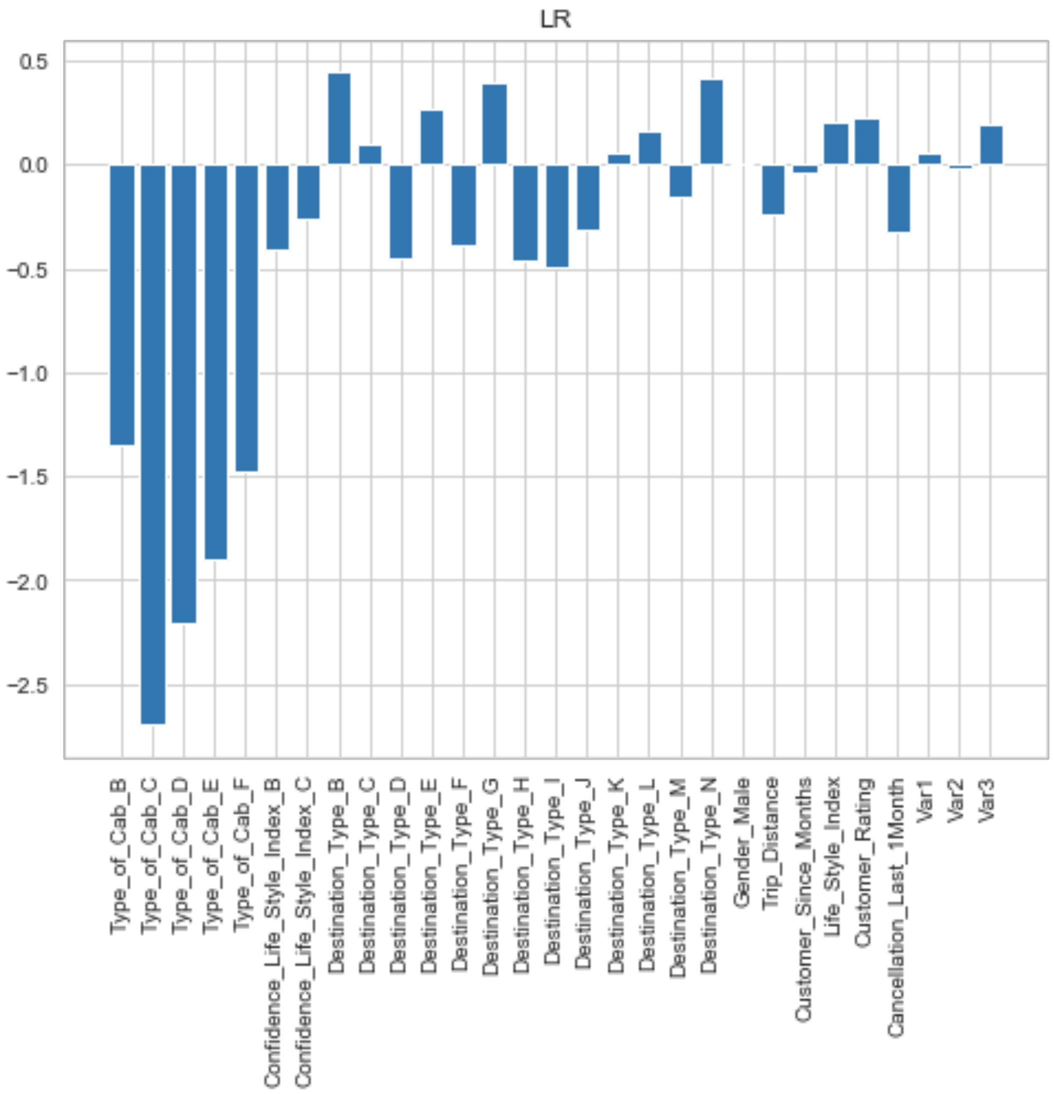


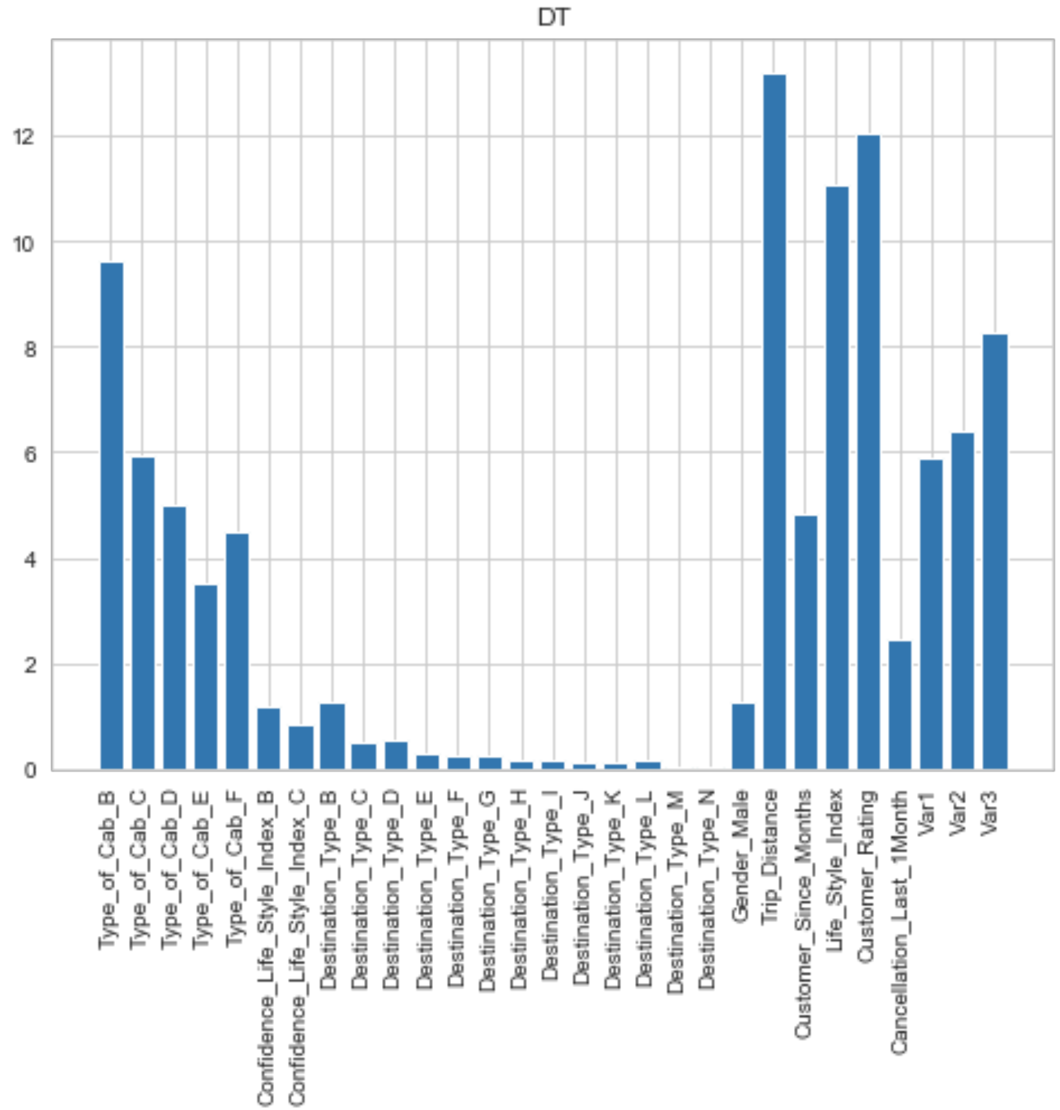


Correlation Pair Plot after One-Hot Encoding

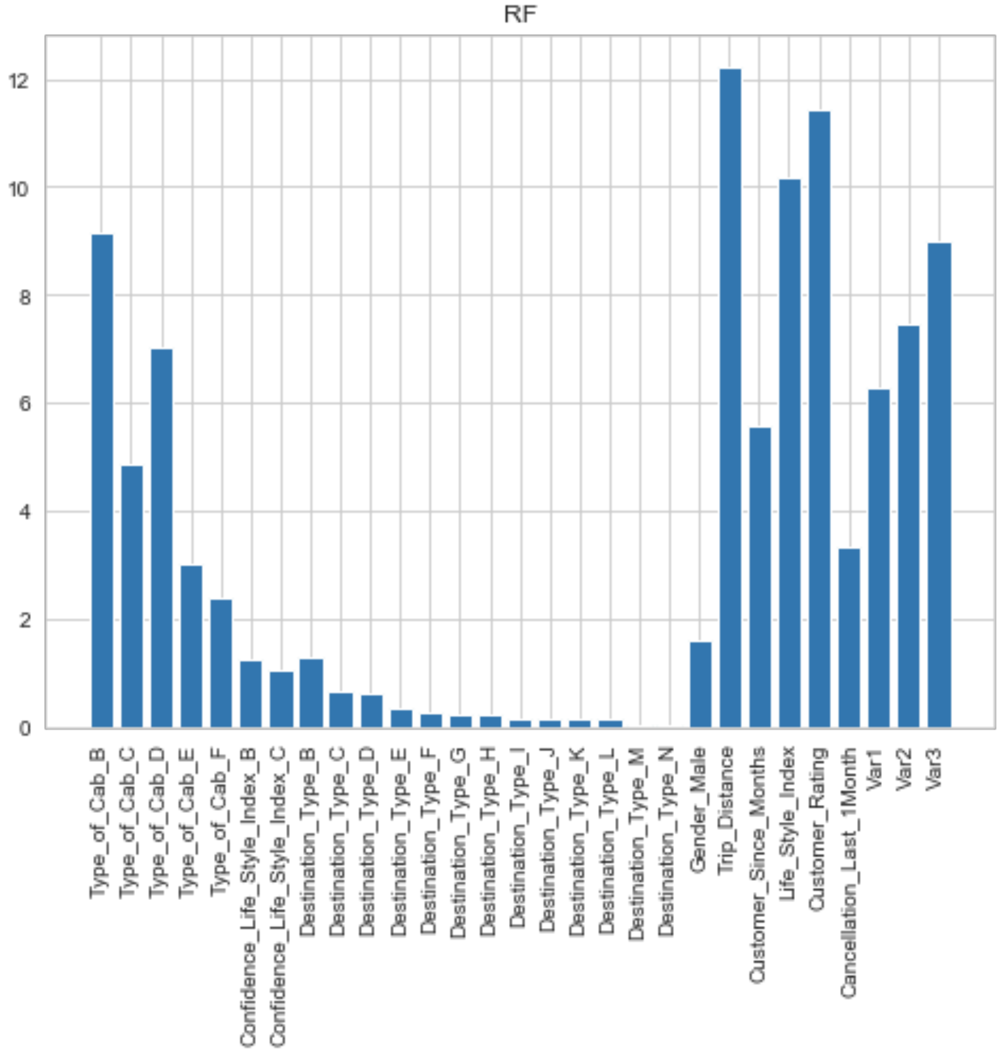


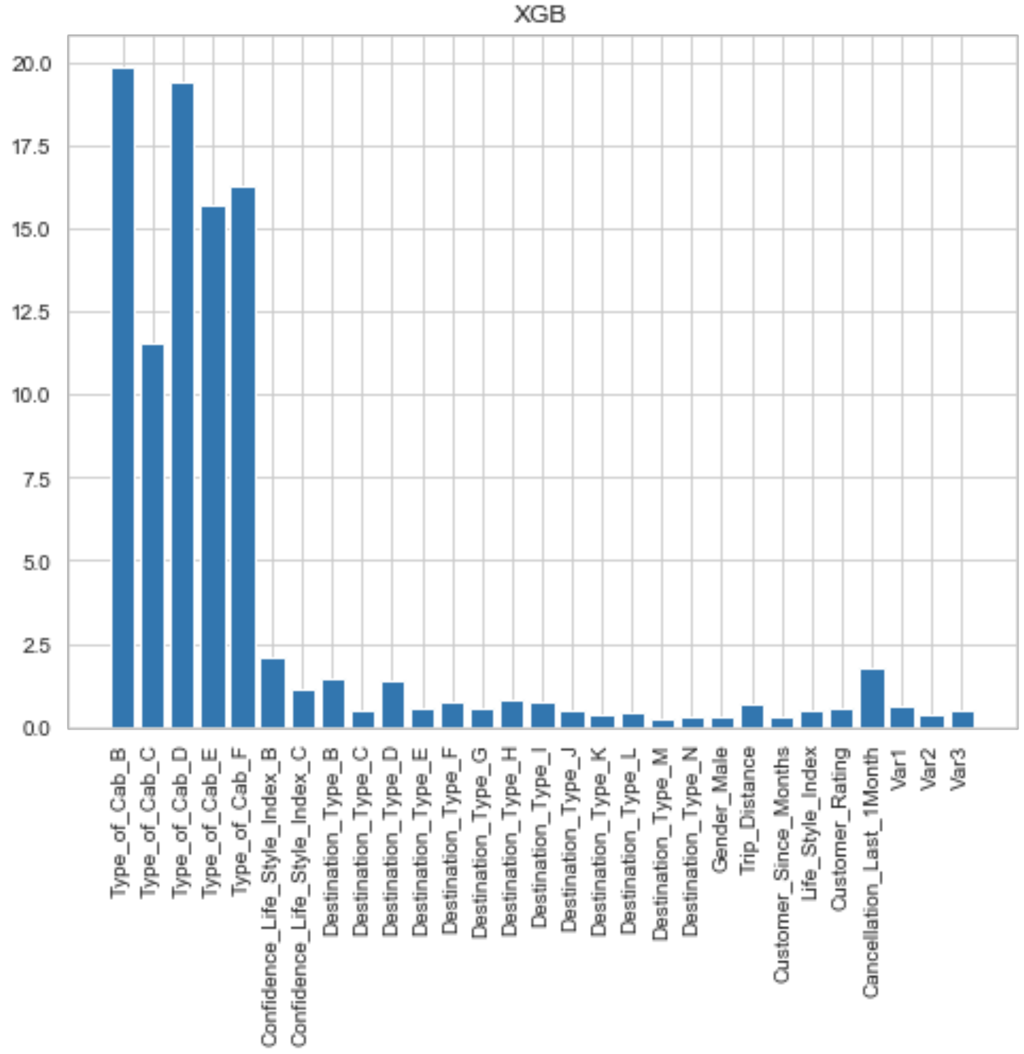
Feature Significance Graphs on Logistic Regression and Decision Trees



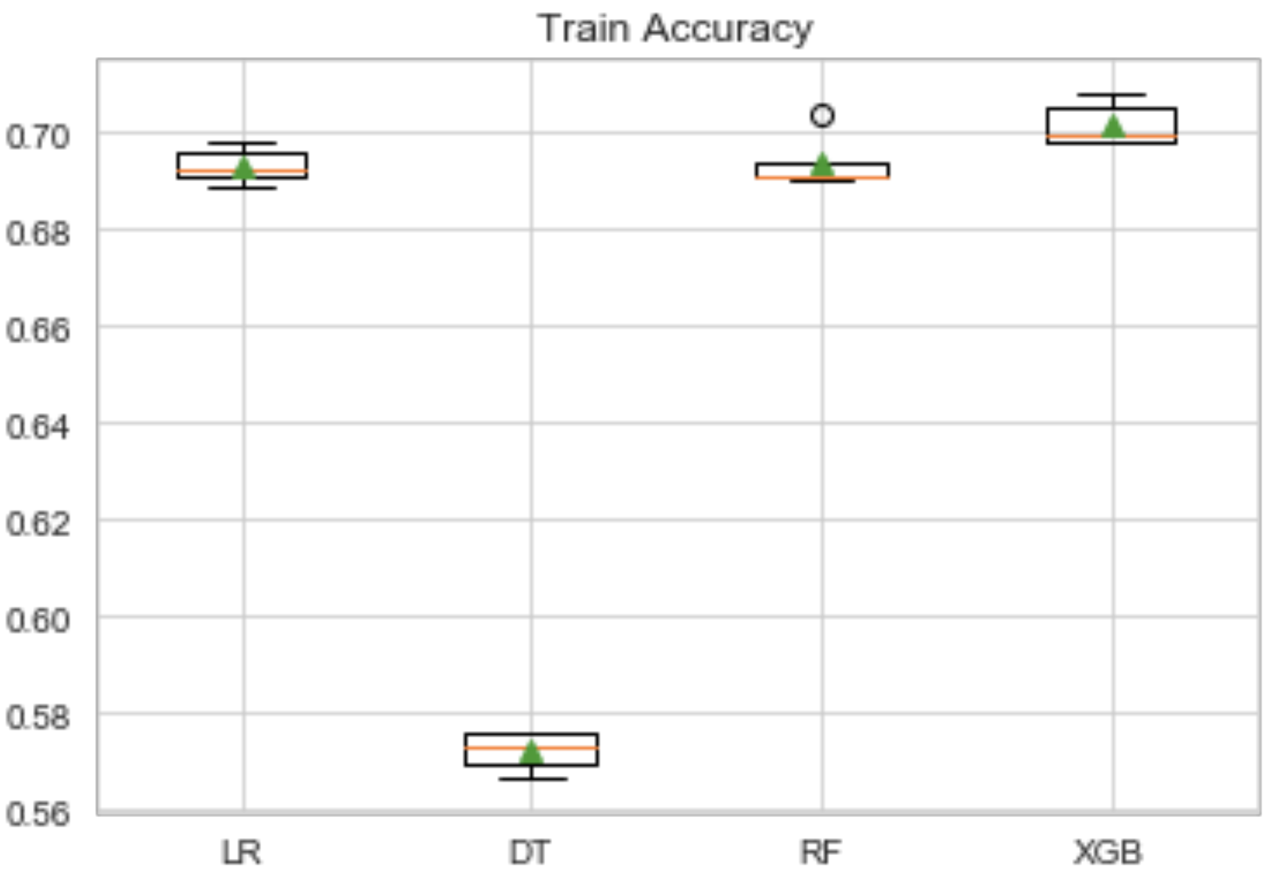


Feature Significance Graphs on Random Forest and XGBoost





Algorithm Comparison Graphs



INSIGHTS

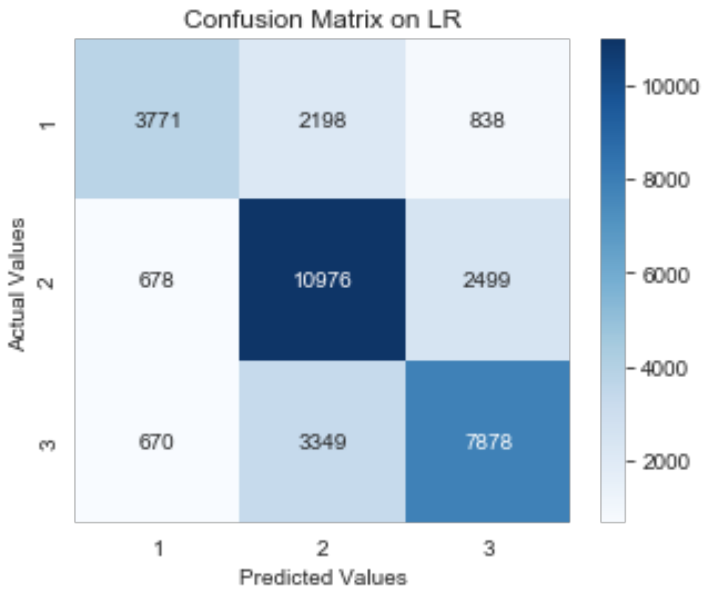
Stratified K-fold Cross Validation is used to avoid changes in train accuracies each time we train the model and also enable some proportion of each target class in train and validation sets.

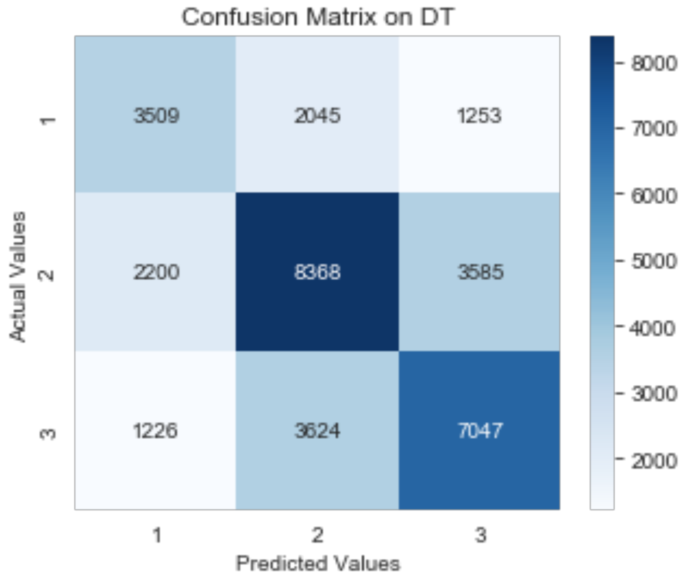
Train Accuracies with different models are -

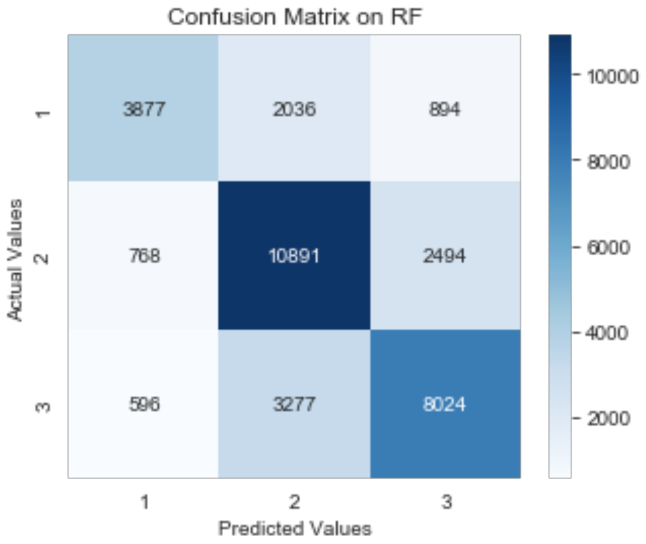
```
LR:  69.34%
DT:  57.23%
RF:  69.42%
XGB:  70.19%
```

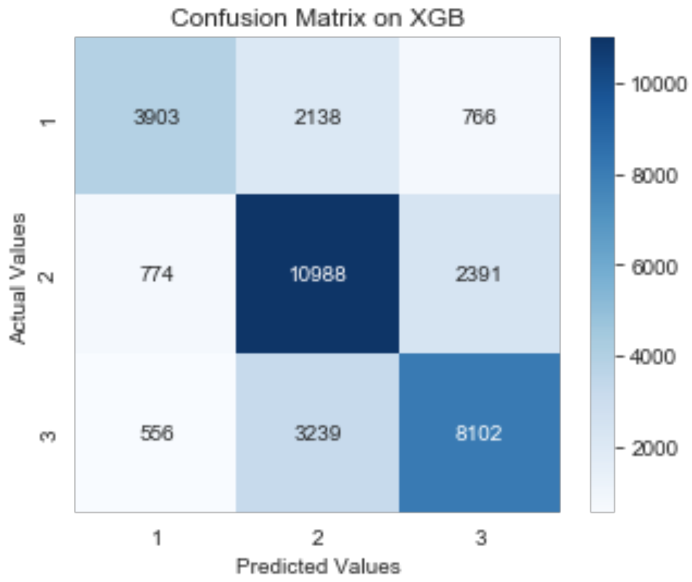
Highest train accuracies are achieved with XGB so we can select XGB as our base model and further try enhancing the accuracies by hyper-parameter optimisation of XGB.

Test Accuracies and Confusion Matrix on Validation Dataset









The Evaluation Matrix that we have chosen are -
- Confusion Matrix
- Precision
- Recall
- F1-score

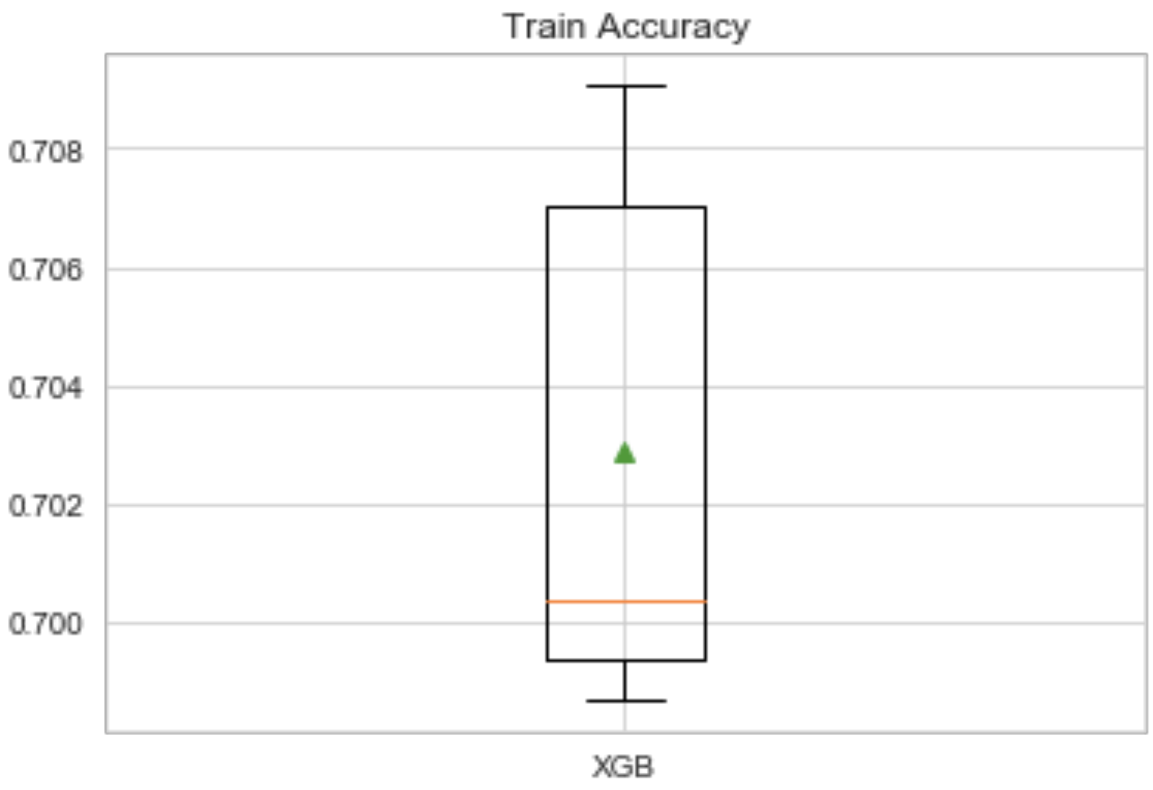
Accuracy is used when there are nearly equal number of samples belonging to each class.

But here, in our case, class 2 of the target variable holds the majority and there are not equal sample distribution in 3 classes, so we use the above mentioned metrics to validate our predictions.

Test Accuracies with different models are -

```
LR: 68.86%
DT: 57.60%
RF: 69.37%
XGB: 69.98%
```

Train Accuracy with XGBoost

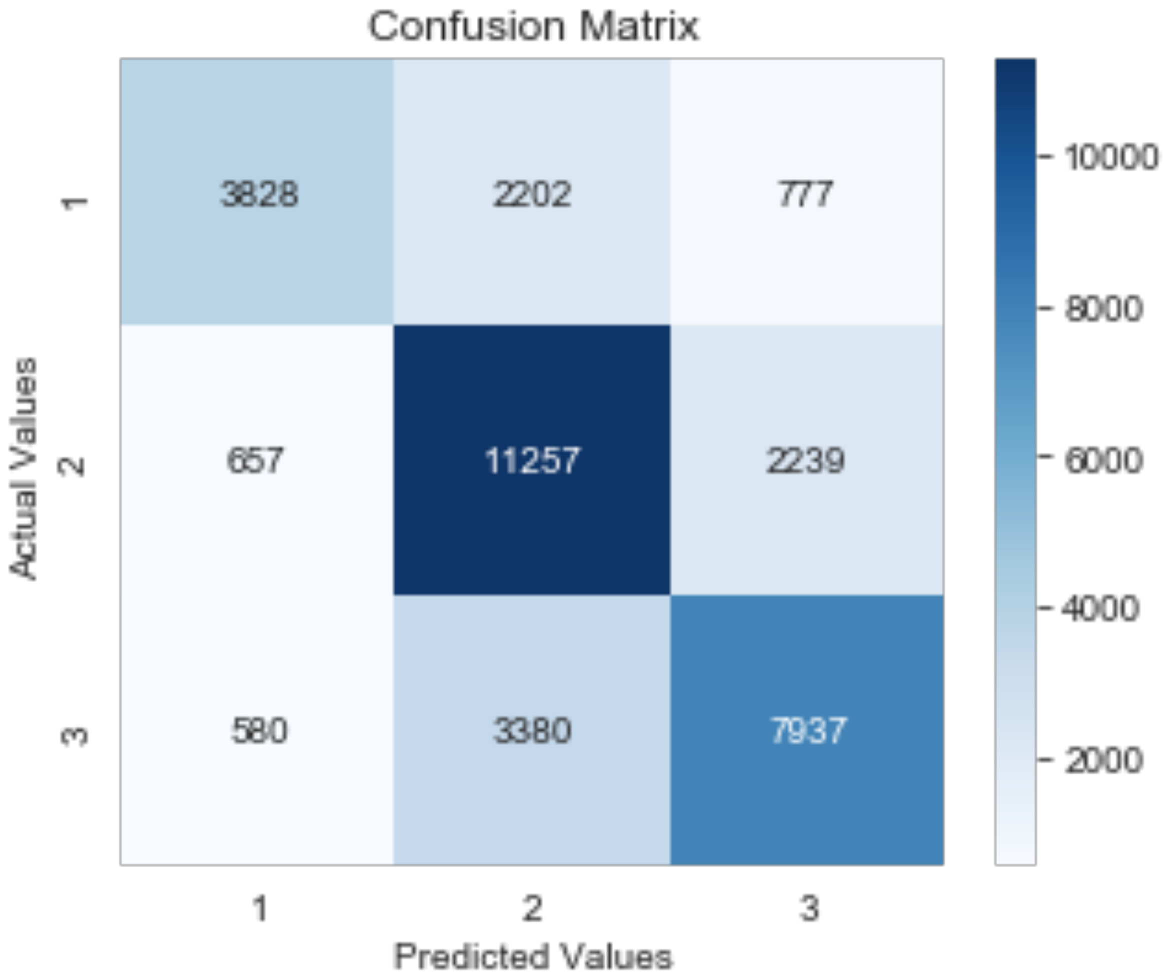


Optimising the hyper parameters, we are able to enhance the train accuracy of XGB to 70.29%.

We have selected XGB as our base model due to the following reasons -

- Gave highest train-test accuracy on our data compared to others
- Ensemble boosting algorithm with base models (DT) added sequentially to reduce the residuals to a great extent
- Fast algorithm

Confusion Matrix on XGBoost



New Test Accuracy of XGB after hyper parameter optimisation -

XGB: 70.07%

THANK YOU