***A***

***Report on***

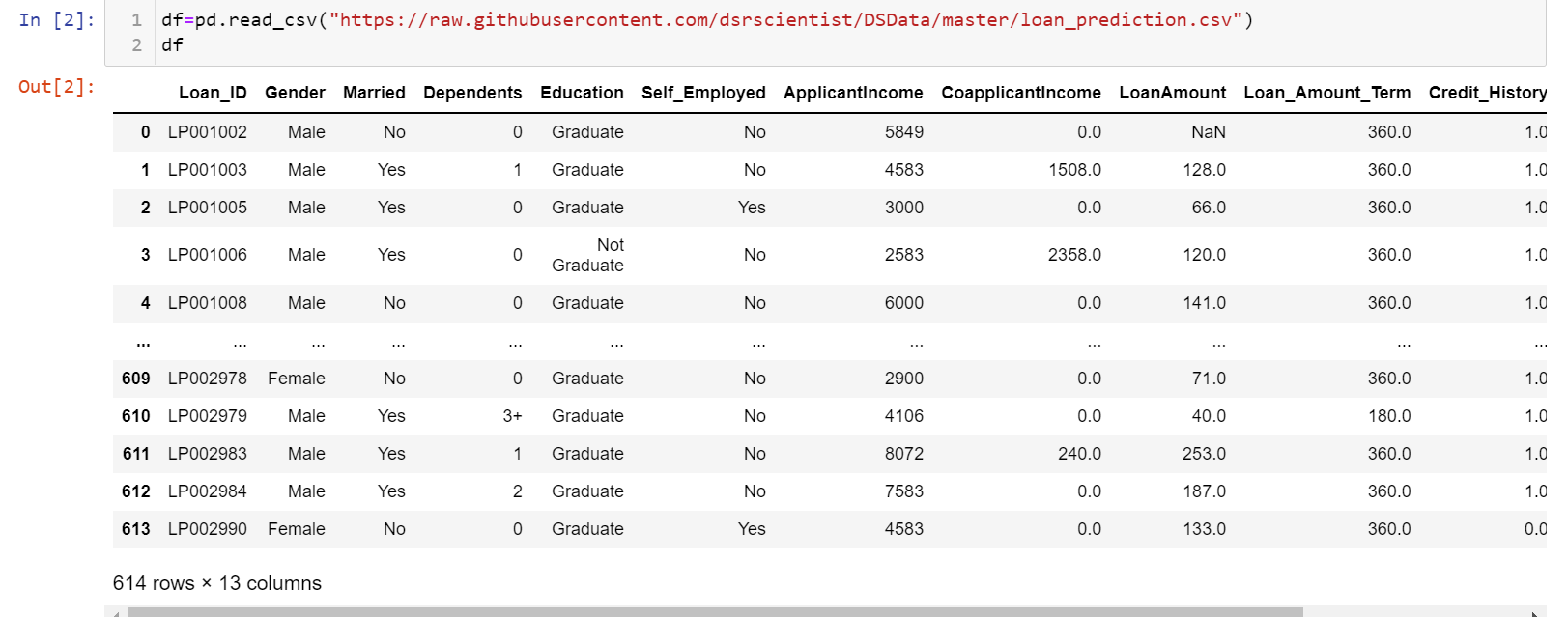
***Loan Status Prediction Project***

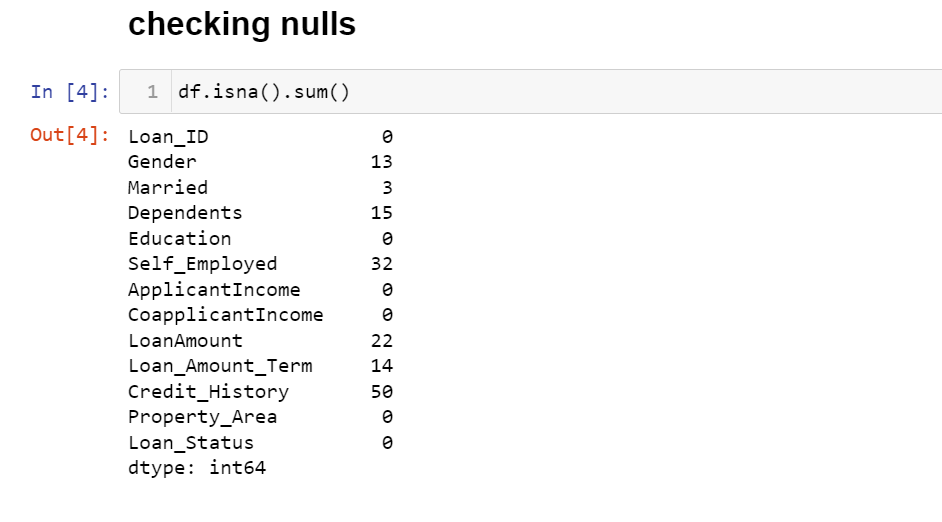
**Problem Definition**

We are given certain details about bank customers such as Loan\_ID, customer employment, customer income, loan amount, loan term etc and we need to predict whether the loan that they applied for will be approved or not.

**Data Analysis**

1. This dataset has 614 rows and 13 columns.

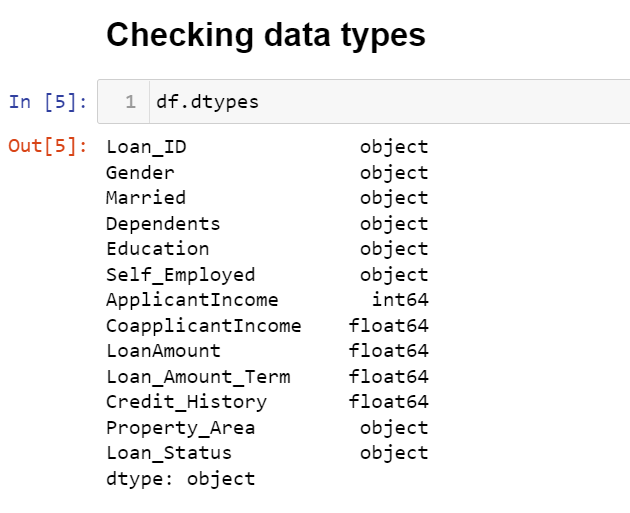
****

1. Checking whether the dataset has null values or not.****

Null values are present in following columns:

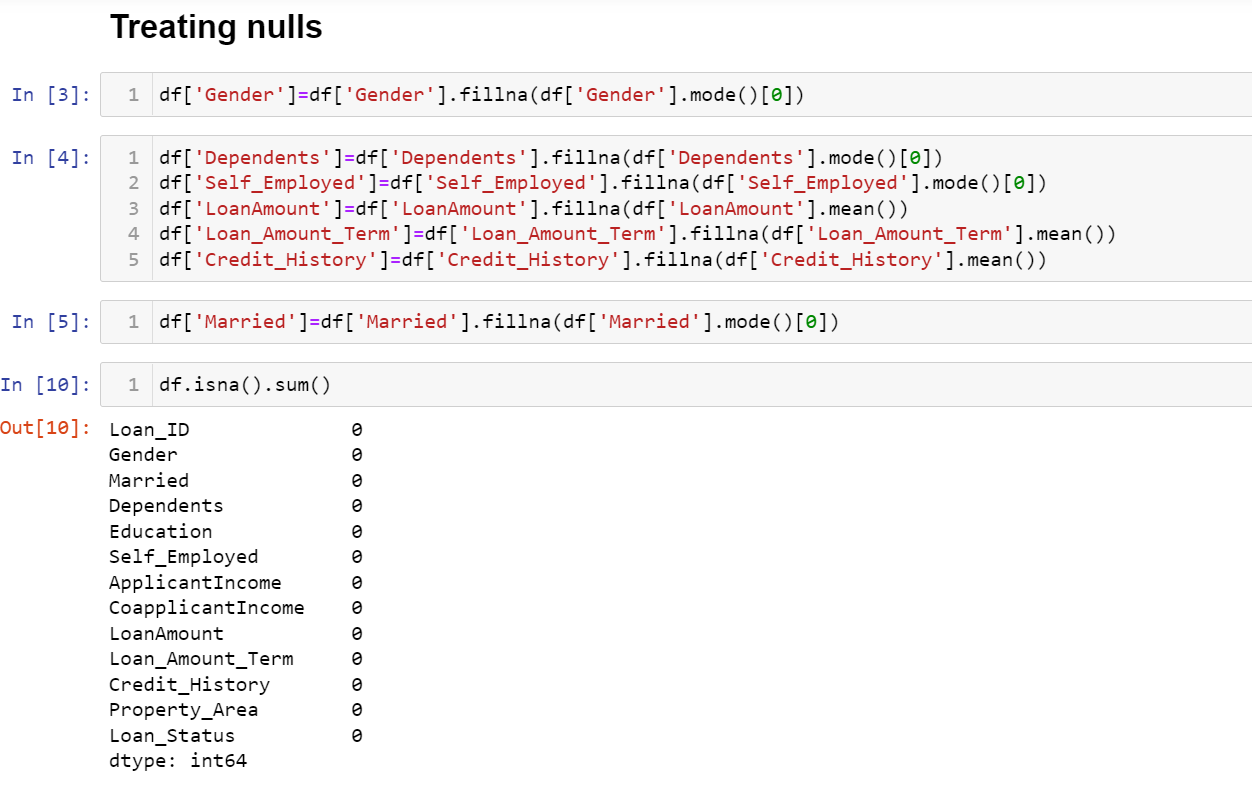
Gender,Married,Dependents,Self\_Employed,LoanAmount,Loan\_Amount\_Term and Credit\_History.

1. Checking datatypes of each feature.



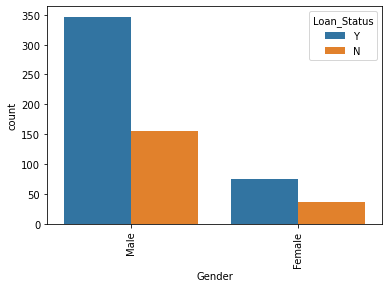
There are both continuous as well as categorical data present so first we will treat null values and then do encoding of categorical data.

1. We will do imputation for treating null values. For continuous data we will replace NaN using mean and for categorical data we will impute NaN using mode.

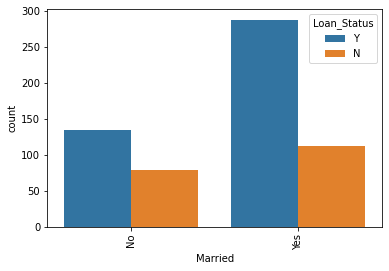


Gender,Married,Dependents and Self\_Employed are the categorical data thus these features are imputed using mode value whereas LoanAmount, Loan\_Amount\_Term and Credit\_History are continuous data thus imputed using mean values.

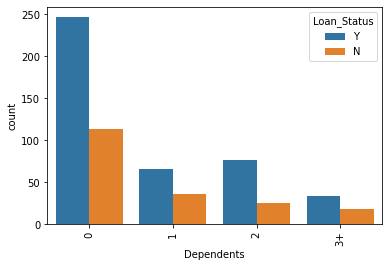
1. Analysing Relation Between categorical independent variables and label Loan\_Status.



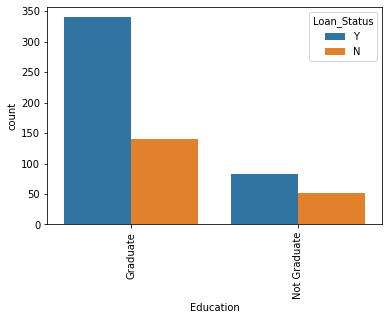
1)Gender: There are more male applicants then female but the probability of loan getting approved is more for female applicants.



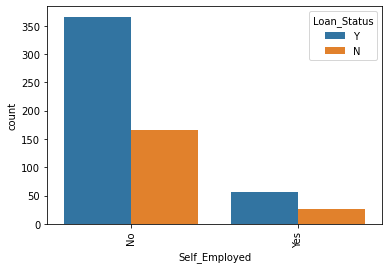
2)Married: More loan applications are from married people and these loan application has higher chance of getting approved because there are less chances that the family man being fraudulent.



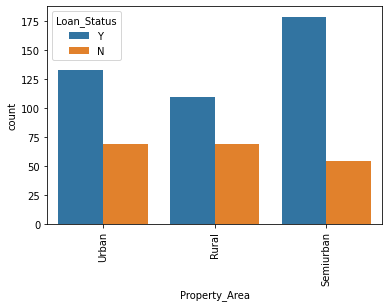
3)Dependents: Higher chance of loan approval for no dependents people because mostly dependents are senior citizens, women’s and children’s and they can be less fraudulent.



4)Education: Higher chance of loan approval for Graduates then non graduates.

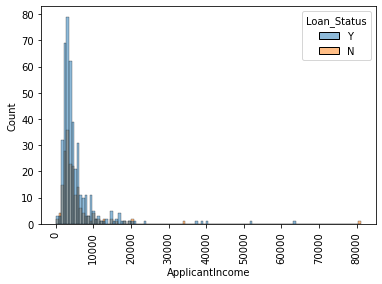


5)Self\_Employed:People who are not self employed means those people who are into any kind of service and not business has higher chance of loan approval.

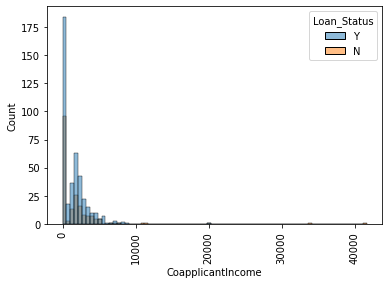


6)Property\_Area:Loan approval change is higher for Semiurban property area and least in Rural property area.

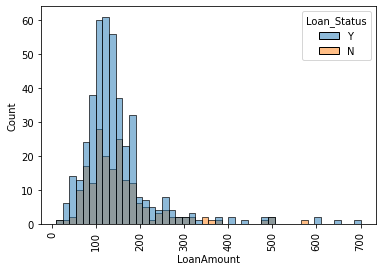
1. Analysing Relation Between continuous independent variables and label Loan\_Status.



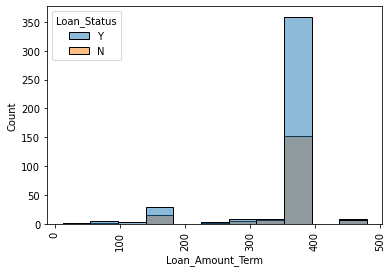
1)ApplicantIncome: More chances of loan approval if income is less than 10000.



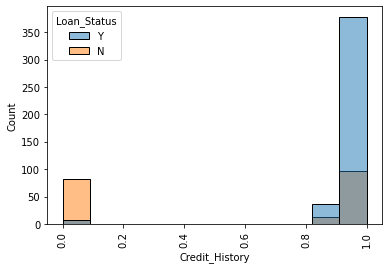
2)CoapplicantIncome: Higher the chances of loan approval if co-applicant Income is 0.



3)LoanAmount: Higher chances of Loan approval if the loan amount is in between 50 and 200. As the loan amount increases the chances of loan approval reduces.



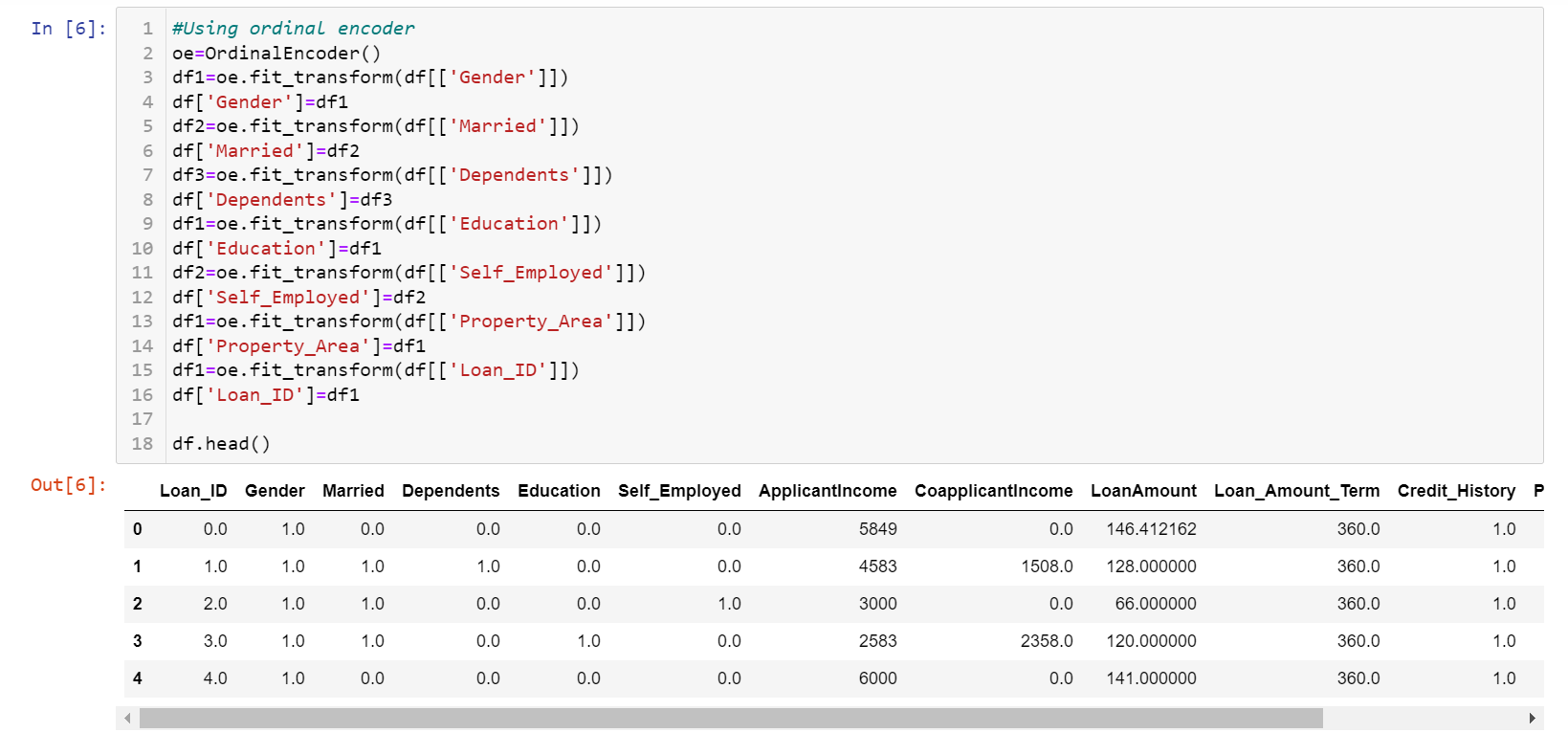
4) Loan\_Amount\_Term: If loan amount return term is in between 350 and 400 more are the chances of loan approval.



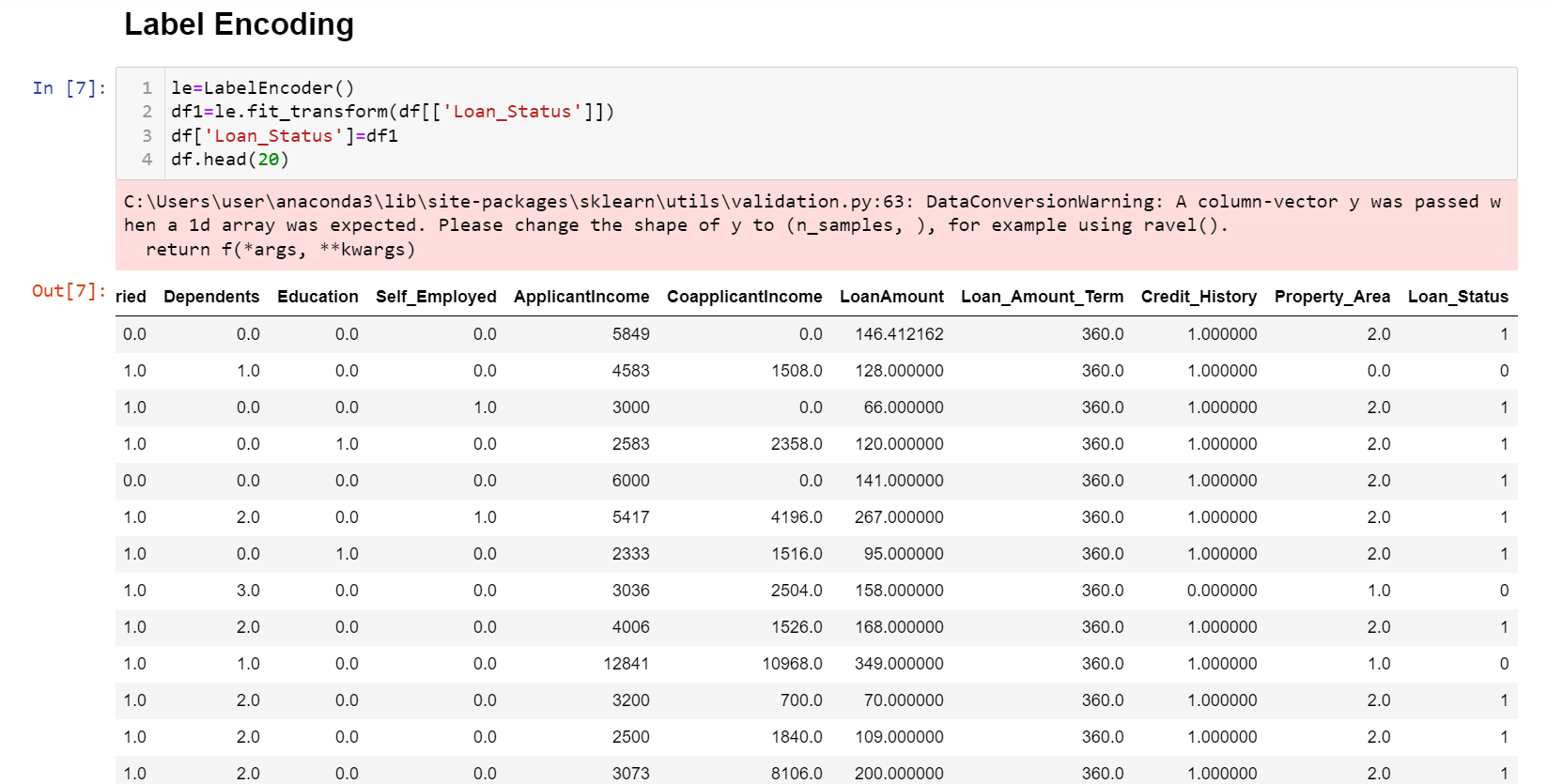
5) Credit\_History: If the applicant has a history of timely depositing the loan amount then there are higher chances of loan approval else there are higher chance of not approving the loan applied.

7) Encoding: As ML Models cannot processes categorical data thus we need to convert all the categorical data into continuous data. Here we can observe that our predictive variable i.e. label itself is a categorical data thus for this we will use Label Encoding and for other independent categorical variables we will use Ordinal Encoder.

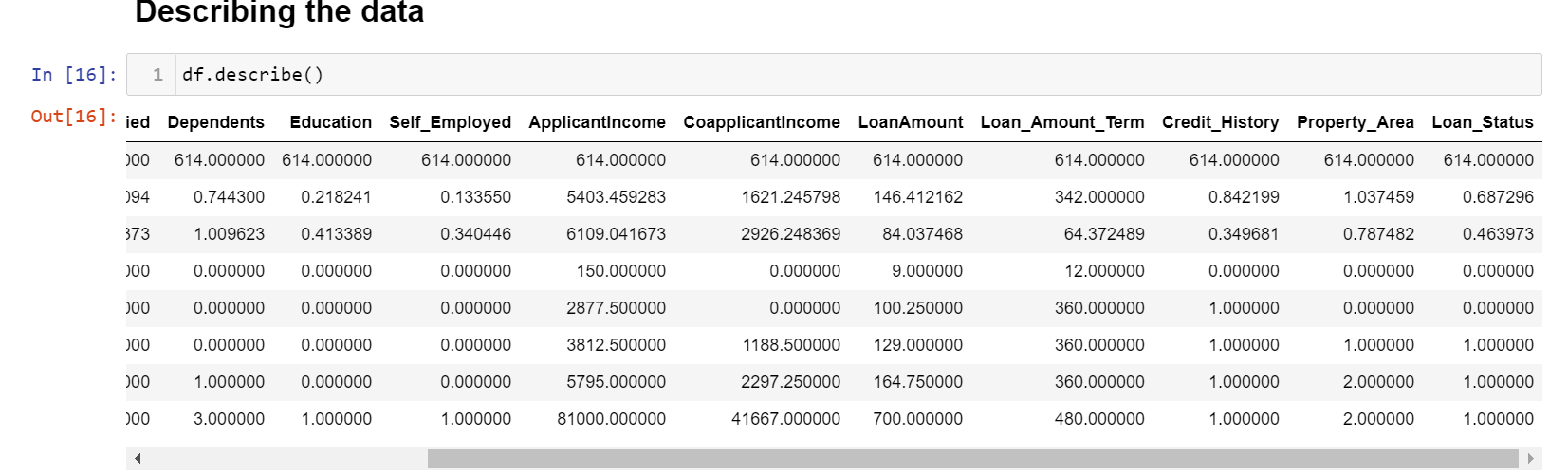
For Independent categorical variables:



For dependent variable Loan\_Status:



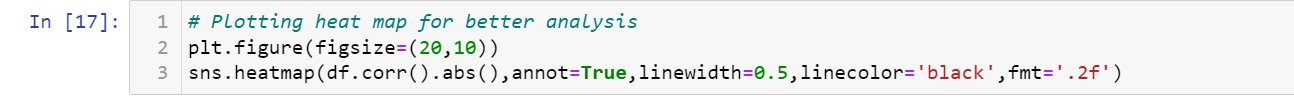
1. Describing the Dataset:



We can observe that there is skewness and outliers in

| Applicant Income | CoapplicantIncome | Loan Amount | Loan\_Amount\_Term |
| --- | --- | --- | --- |
|  |  |  |  |

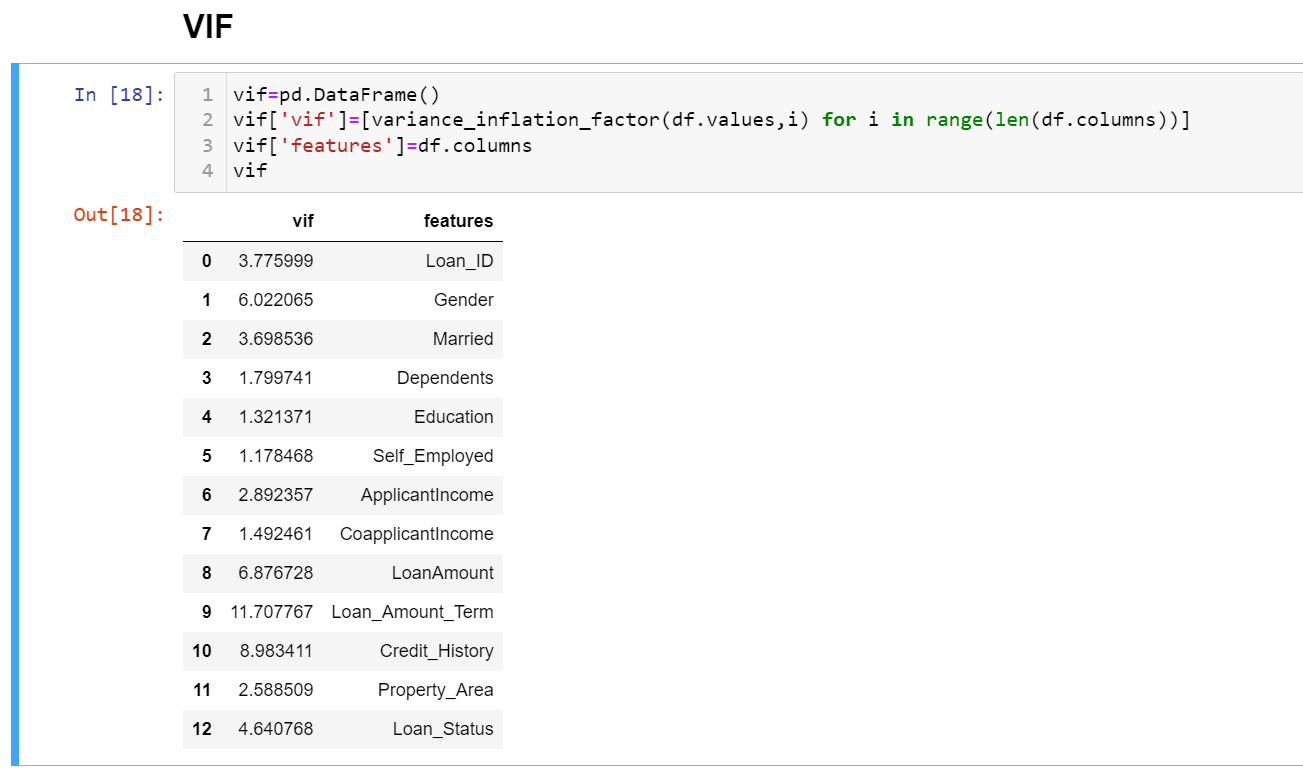
1. Finding Correlation between features: We will plot heatmap of correlations between features and label so that we could see by how much percent each feature are related and by how much percent they are related with the label.





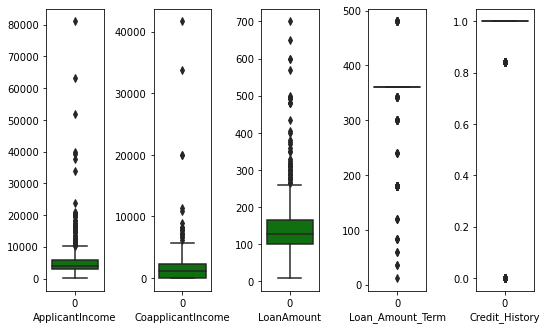
     We can see that multicollinearity could exists between ApplicantIncome and LoanAmount . We can also observe that Credit\_Histroy is highly related with the label Loan\_Status and Self\_Employed and ApplicantIncome is least related.

1. In order to be sure of multicollinearity we calculated VIF score of each feature and those feature having VIF score greater than 10 are considered to be correlated with another feature.

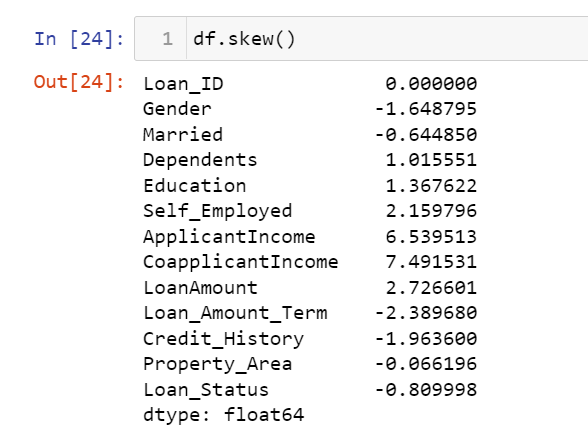


Loan\_Amount\_Term has vif>10 and is only 2% related with Loan\_Status so we have dropped this feature.

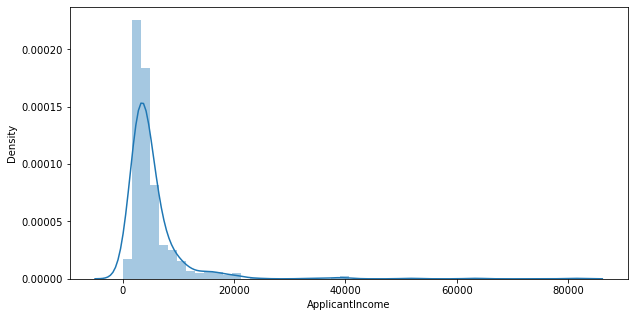
1. We have observed outliers in all the continuous feature and treated then also later.



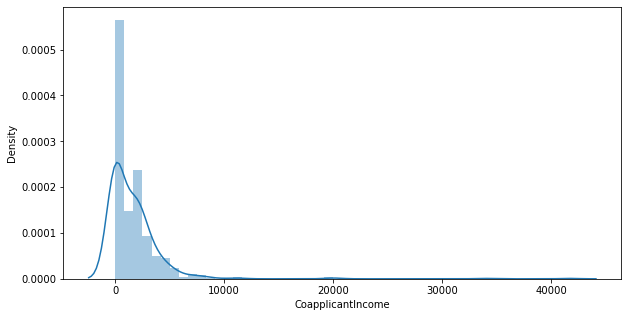
1. Considering the best range of skewness as -0.5 to 0.5, we could observe that ApplicantIncome,CoapplicantIncome, LoanAmount,Loan\_Amount\_Term and credit\_History are skewed. So we treated it later as well.



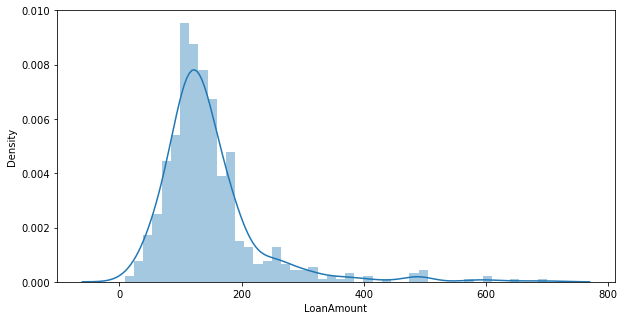
1. In order to be sure of skewness we also plotted distribution graph for ApplicantIncome,CoapplicantIncome, LoanAmount,Loan\_Amount\_Term and credit\_History.



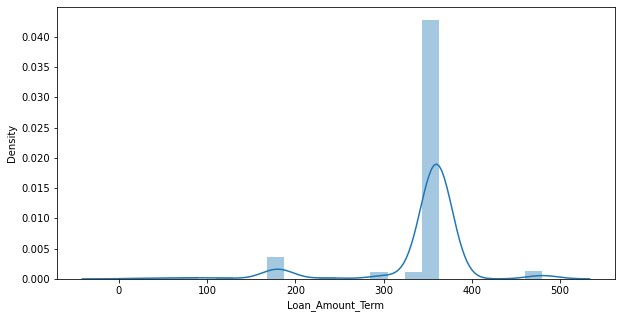
ApplicantIncome is right skewed and should be treated.



CoapplicantIncome is also right skewed and should be treated.



Loan Amount is also right skewed and should be treated.

****

Loan\_Amount\_Term is both left and right skewed.

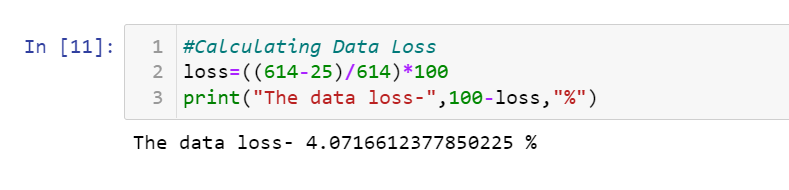
**EDA Concluding Remarks**

We treated the null values using Imputation technique. We observe that if applicant is married graduate male with no dependents and are in some kind of service then he has a higher chance of loan approval.

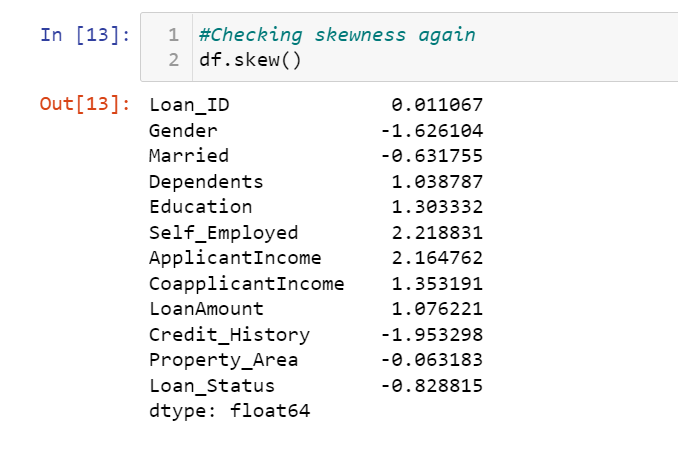
The dataset has to be further processed as it has multicollinearity, outliers and skewness.

**Pre-processing Pipeline**

1. We treated Outliers by using Z-score method. The rows of those features having outliers if z-score is greater than 3 then we are dropping those rows. After applying z-score method we found that there was only 4% of overall data loss so we can bear this much of data loss.

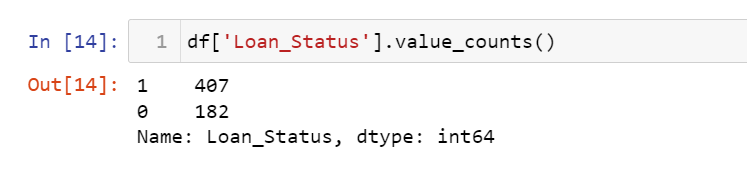


1. After treating outliers we checked the skewness again and found that it still exists.

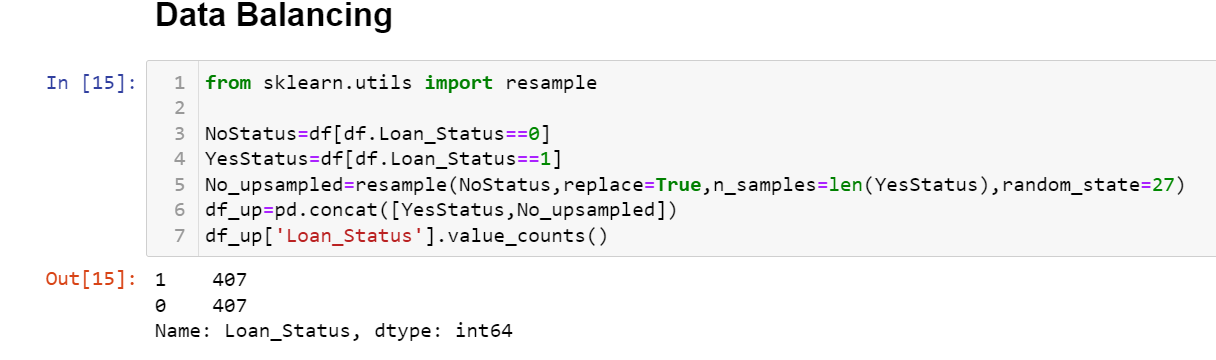


1. Sine this problem is a classification problem so we need to check whether the data is balanced between yes and no class of Loan\_Status so that the model is not biased towards a particular class.

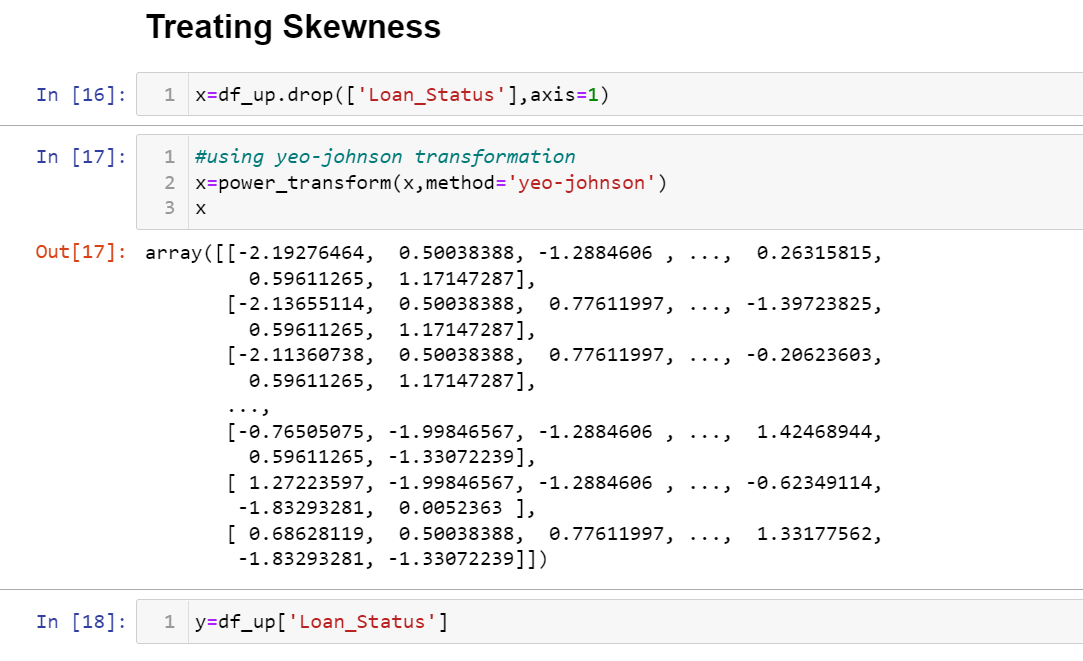
We found that there are more data of class 1 than class 0.



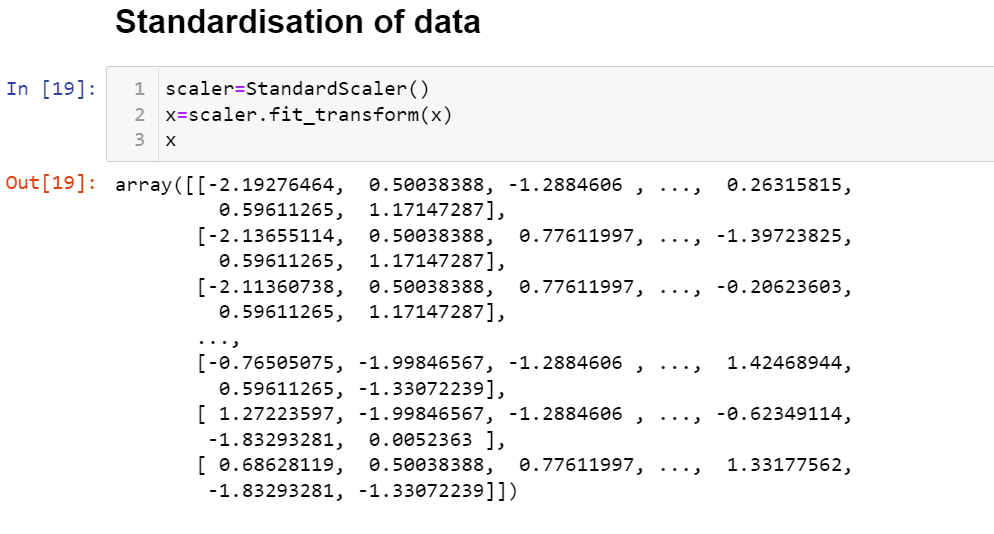
So we need to upsample class 0 data using resampling technique.



1. After balancing the data we treated skewness using yeo-johnson transformation method.



1. Before traing the model with this data we need to standardise the data so that model gives its best performance.



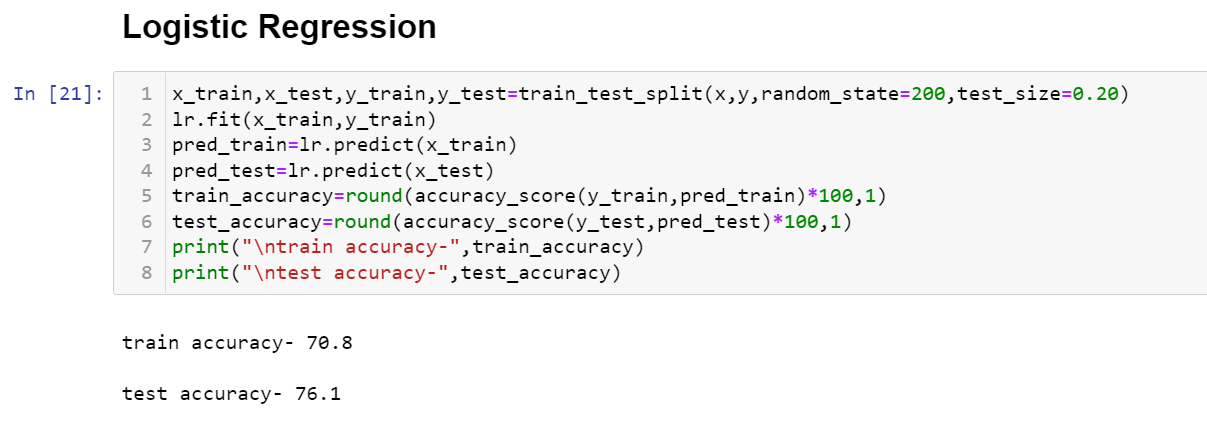
**Building Machine Learning Models**

20) Before building the model I split the data into training data and testing data using train test and split technique.

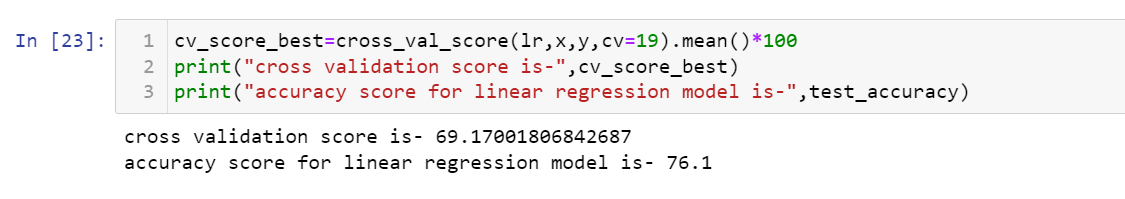
I found best accuracy at random state 200.

**1) Applied Logistic Regression:**

We applied this algorithm and found the rain accuracy to be 71% and test accuracy to be 76%.



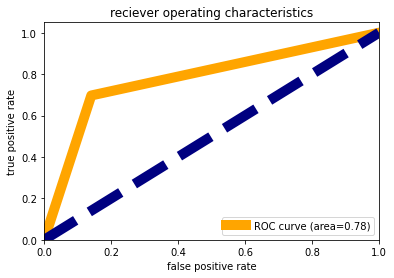
To cross validate this accuracy we used cross validation technique.



And found out that the model is not overfitted.

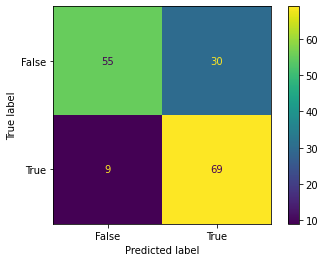
We further checked the model performance on other metrics such as AUC-ROC Curve, Confusion Matrix and Classification Report.

**AUC-ROC Curve:**



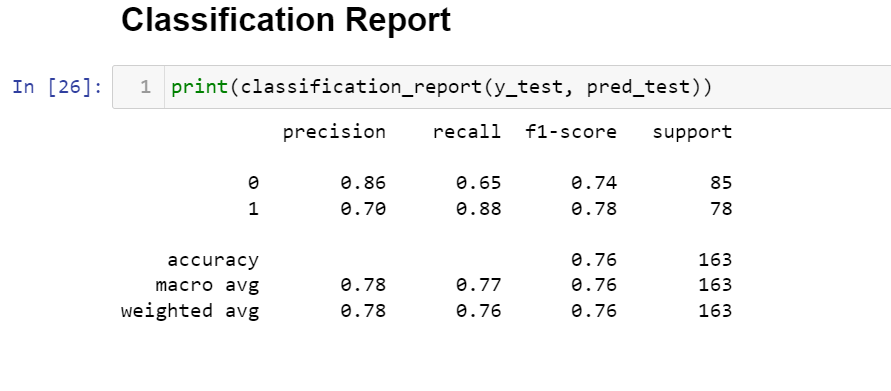
We observed that the area under the curve is just 78% that means 78% times model is predicting accurately and rest all other time it gives wrong prediction.

**Confusion Matrix:**



Model is good in predicting true classes but bad in predicting false class.

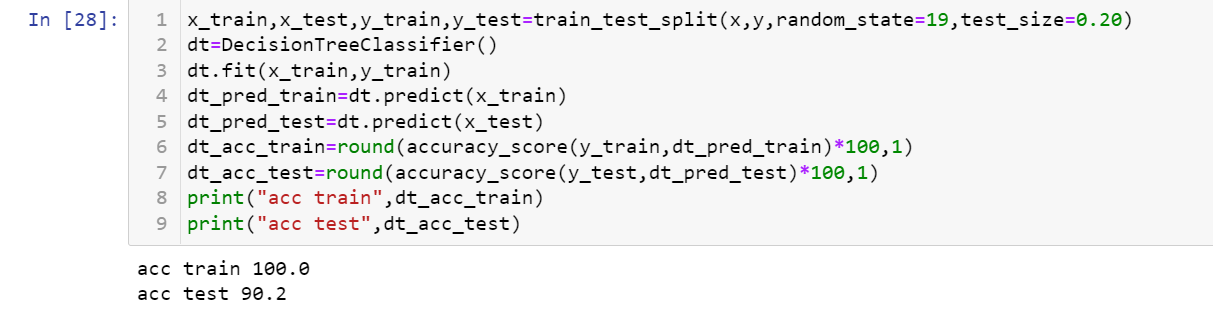
**Classification Report:**

****

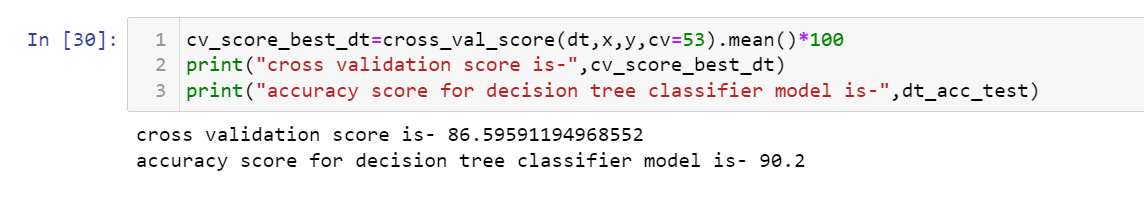
Model is poor in recalling class 0 i.e. False than class 1 i.e. True. F1-score of class 0 is also less than that of class1.

**2) Applying Decision Tree Classifier:**

We applied this algorithm and found the rain accuracy to be 100% and test accuracy to be 90.2%.



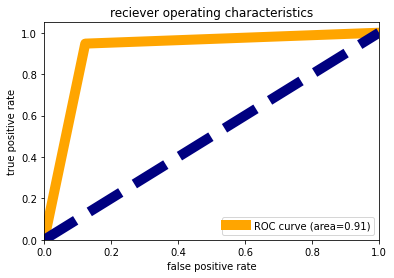
To cross validate this accuracy we used cross validation technique.



And found out that the model is not overfitted.

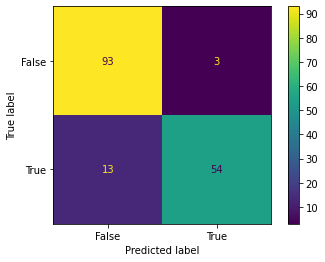
We further checked the model performance on other metrics such as AUC-ROC Curve, Confusion Matrix and Classification Report.

**AUC-ROC Curve:**



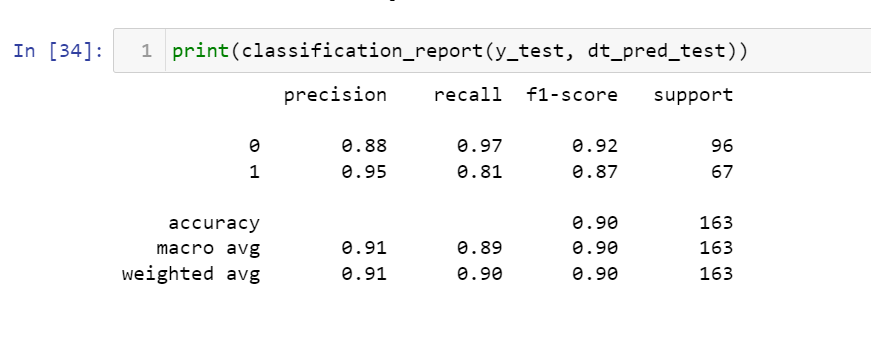
We observed that the area under the curve is 91% that means 91% times model is predicting accurately and rest all other time it gives wrong prediction. It is giving good accuracy but we need to check other parameters as well.

**Confusion Matrix:**

****

Model is good in predicting false classes but bad in predicting true classes.

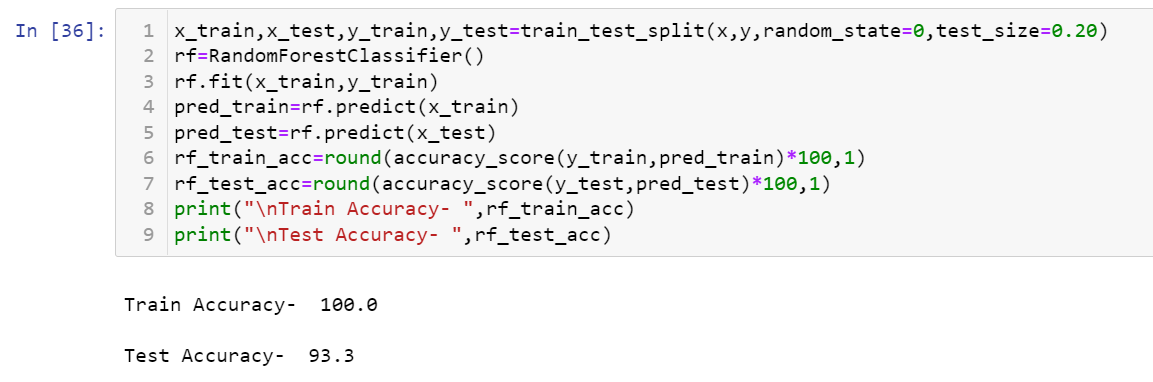
**Classification Report:**

****

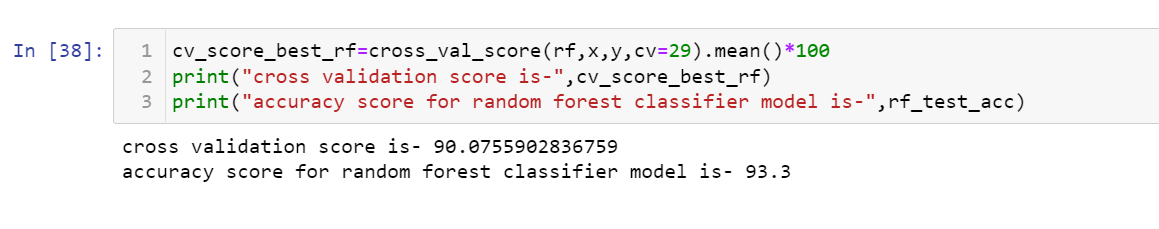
Model is poor in recalling class 1 i.e. True than class 0 i.e. False. F1-score of class 1 is also less than that of class0. So the average accuracy for predicting both the classes is 90%.

**3) Applying Random Forest Classifier:**

We applied this algorithm and found the rain accuracy to be 100% and test accuracy to be 93.3%.



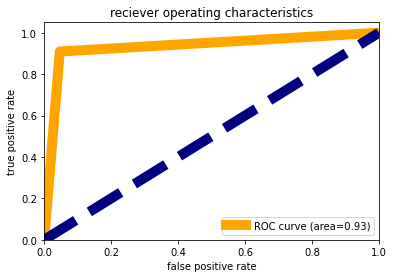
To cross validate this accuracy we used cross validation technique.



And found out that the model is not overfitted as its cv score is 90.0% which is close enough to 93%.

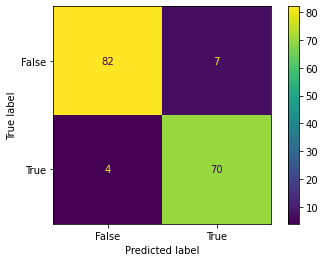
We further checked the model performance on other metrics such as AUC-ROC Curve, Confusion Matrix and Classification Report.

**AUC-ROC Curve:**



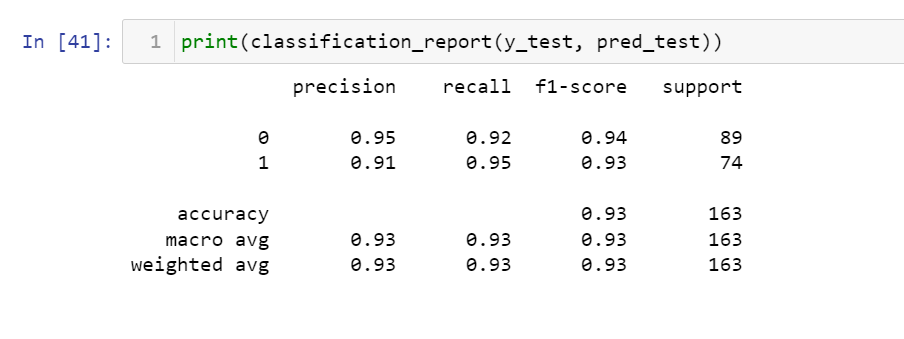
We observed that the area under the curve is 93% that means 93% times model is predicting accurately and rest all other time it gives wrong prediction. It is giving good accuracy but we need to check other parameters as well.

**Confusion Matrix:**

****

Model is good in predicting false classes but bad in predicting true classes. Although we can also see that the samples for true classes on which it is tested is less than samples for false classes.

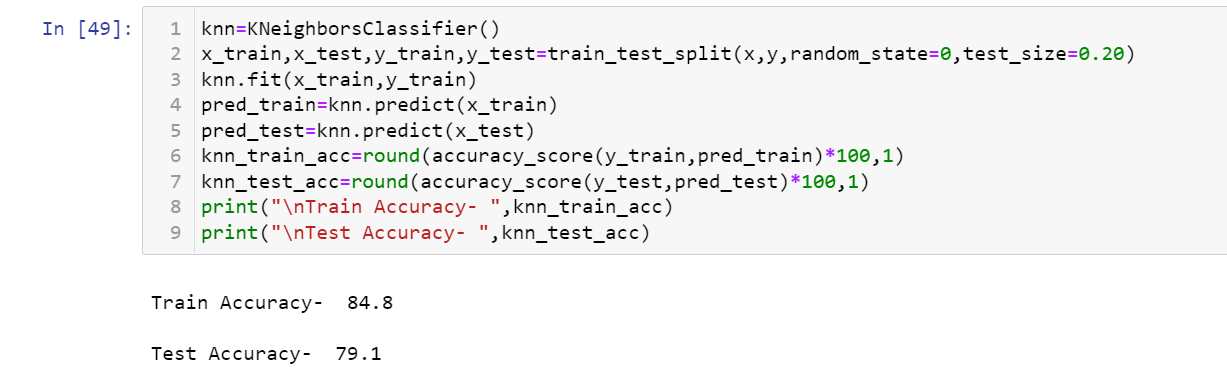
**Classification Report:**

****

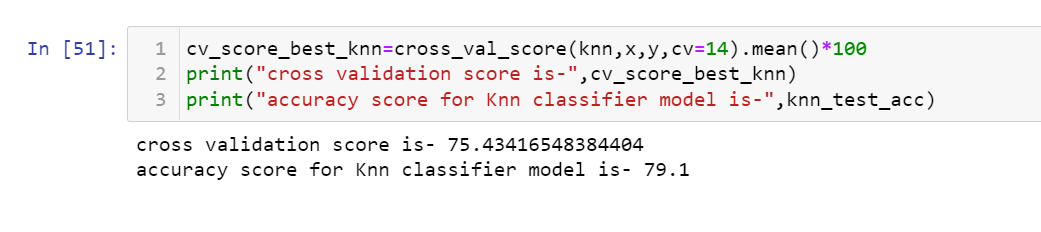
Model is 0.95 times precise in predicting class 0 but it could only recall 92% times class 0 where as it is only 0.91 times precise in predicting class 1 but more number of times that is 95% times it could recall class 1. So this is leading there average accuracy to 93%.

**4) Applying Knn Classifier:**

We applied this algorithm and found the rain accuracy to be 84.2% and test accuracy to be 79.1%.



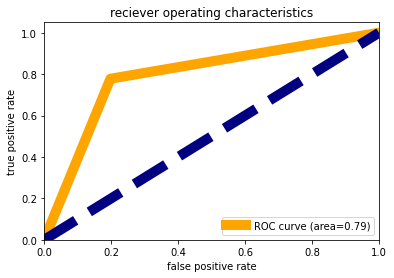
To cross validate this accuracy we used cross validation technique.



And found out that the model is not overfitted as its cv score is 75.4% which close to model accuracy i.e. 79.1%.

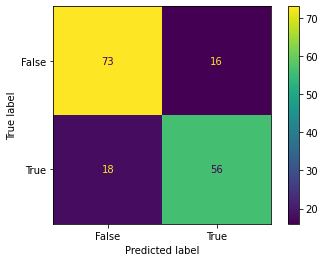
We further checked the model performance on other metrics such as AUC-ROC Curve, Confusion Matrix and Classification Report.

**AUC-ROC Curve:**



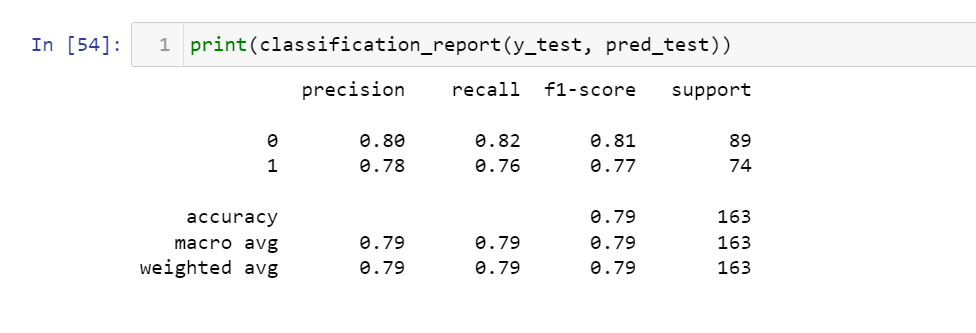
We observed that the area under the curve is just 79% that means just 79% times model is predicting accurately and rest all other time it gives wrong prediction. This accuracy is not satisfying as other models are giving higher accuracy.

**Confusion Matrix:**

****

Model is good in predicting false classes but bad in predicting true classes. Although we can also observe that number of testing samples for class 0 is more than that of class1.

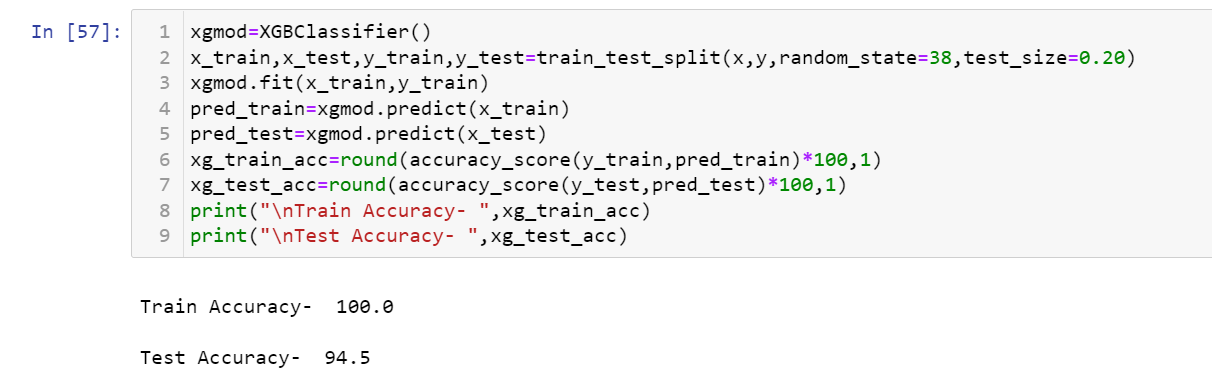
**Classification Report:**

****

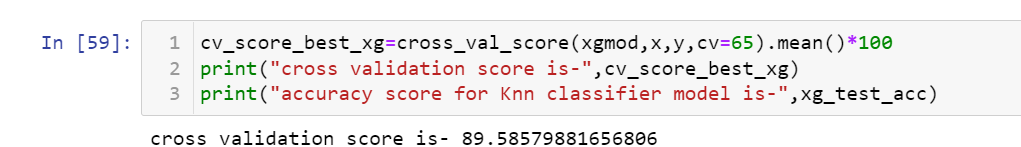
Model is 0.80 times precise in predicting class 0 but it could only recall 82% times class 0 where as it is only 0.78 times precise in predicting class 1 but less number of times that is only 76% times it could recall class 1. F1-score for class 0 is 81% and for class 1 is 77%, so this is leading there average accuracy to 79%.

**5) Applying XGBoost Classifier:**

We applied this algorithm and found the rain accuracy to be 100% and test accuracy to be 94.5%, which is the best accuracy till now that we have achieved.



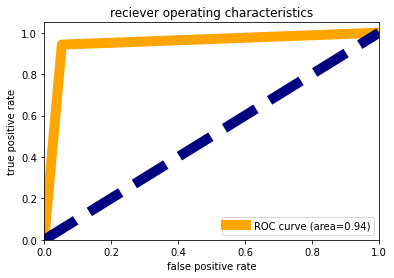
To cross validate this accuracy we used cross validation technique.



And found out that the model is not overfitted as its cv score is 89.6% which can be considered close to model accuracy i.e. 94.5%.

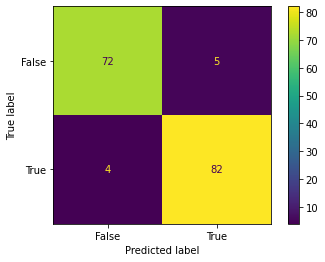
We further checked the model performance on other metrics such as AUC-ROC Curve, Confusion Matrix and Classification Report.

**AUC-ROC Curve:**



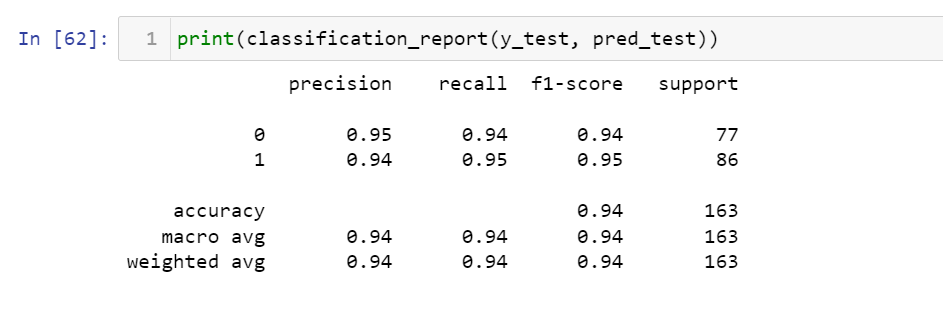
We observed that the area under the curve is 94% that means just 94% times model is predicting accurately and rest all other time it gives wrong prediction. It is giving good accuracy but we need to check other parameters as well.

**Confusion Matrix:**

****

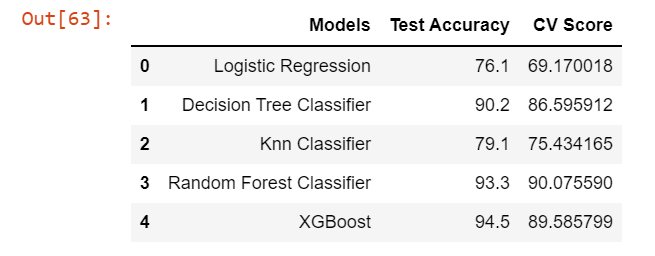
Model is good in predicting true classes but bad in predicting false classes. Although we can also observe that number of testing samples for class 1 is more than that of class 0.

**Classification Report:**

****

Model is 0.95 times precise in predicting class 0 but it could only recall 94% times class 0 where as it is 0.95 times precise in predicting class 1 but more number of times that is 95% times it could recall class 1. F1-score for class 0 is 94% and for class 1 is 95%, so this is leading there average accuracy to 94%, which could be considered as one of the best accuracy.

**Model Summary:**



After considering all the parameters above we finalized Random Forest classifier as our final model because it is giving best accuracy and there **is least difference between accuracy score and cv score,** the area under AUC-ROC Curve is also highest among all other models and its classification report is better then others. So lets try to improve its accuracy by performing hyperparameter tuning on it.

**Hyperparameter tuning on Random Forest Classifier:**

Now we tried to improve the accuracy of Random forest classifier so we considered following parameters for the model and out of these we will find out the best parameter using which the model will perform its best.

The parameters on which we tunned is:

'bootstrap': [True, False],

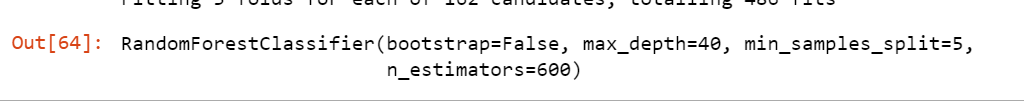
'max\_depth': [40, 50, 60],

'min\_samples\_leaf': [1, 2, 4],

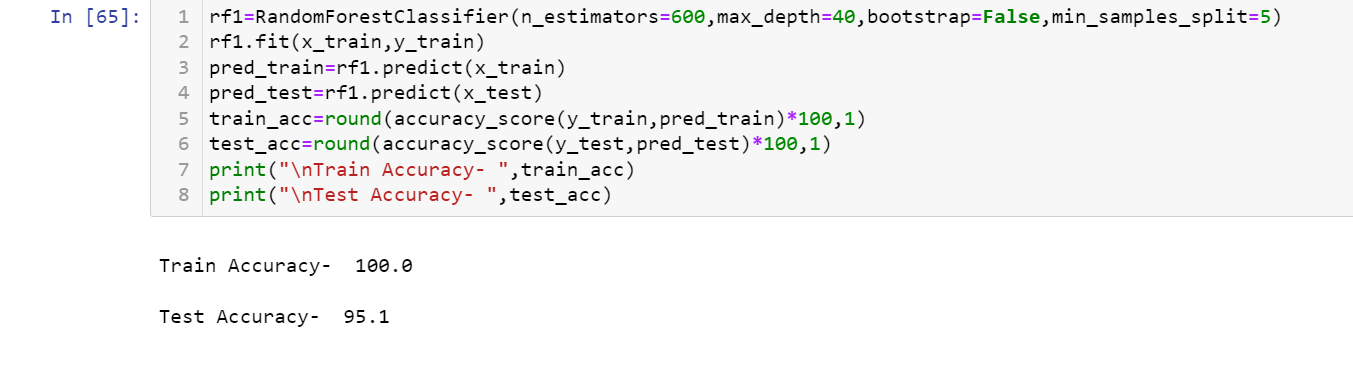
'min\_samples\_split': [2, 5, 10],

'n\_estimators': [200, 400, 600]

These are the best parameters that we got:



We applied these parameters on the model again and got 95% test accuracy.



Finally we saved this tunned model.

**Conclusion**

We got a dataset of size 614 rows and 13 columns with lot many null values. We cleaned this data by imputing null values, encoding, treating multicollinearity, treating outliers and skewness and standardizing the data.

Further we split our data into train and test samples and applied various algorithms:

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. Knn classifier
5. XGBoost Classifier

And we found that the best performing model is Random Forest Classifier so we tunned it to its best parameters there by increasing its accuracy to 95%.

So if we subject this model to actual testing data it would give correct predictions 95% of the time.