**Loan Prediction**

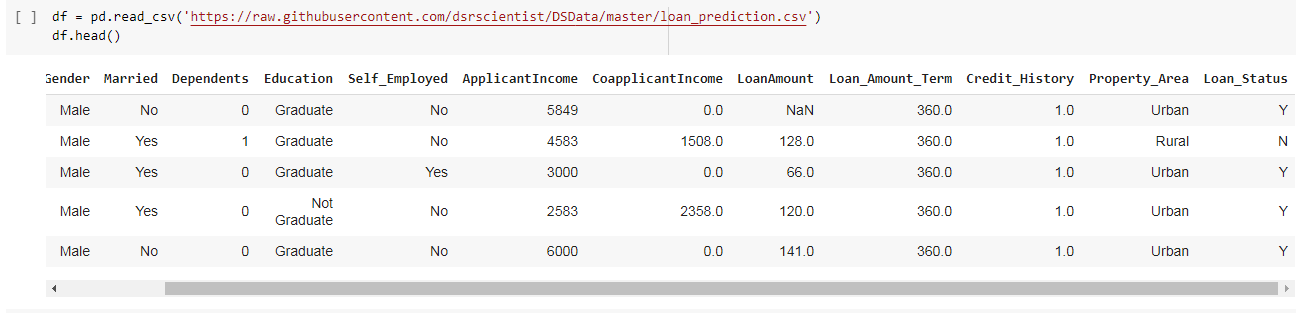
The Project is to predict whether the candidates applied for loan will be approved or not

The independent features included in the dataset are : - Loan\_ID, Gender, Married, Dependents, Education, Self\_Employed, ApplicantIncome, CoapplicantIncome, Loan\_Amount, Loan\_Amount\_Term, Credit History, Property\_Area

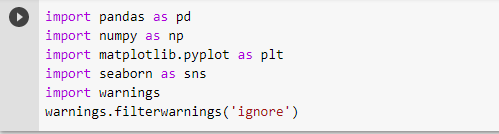
The Dependent feature is Loan\_Status.

This dataset includes details of applicants who have applied for loan.

The Dataset is attached below.



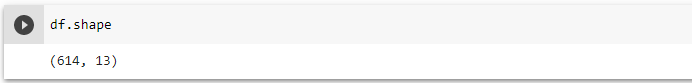
1. Firstly, we need to import necessary libraries like pandas, numpy for reading the data and performing various operations in dataframe, matplotlib.pyplot and seaborn for pictorial representations and warnings library is imported for not displaying warnings signs in the notebook.



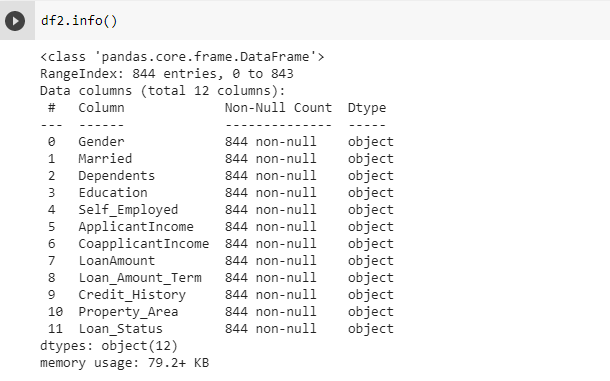
1. **After importing libraries, we need to import the datasets**



1. **Checking the shape of the datasets**

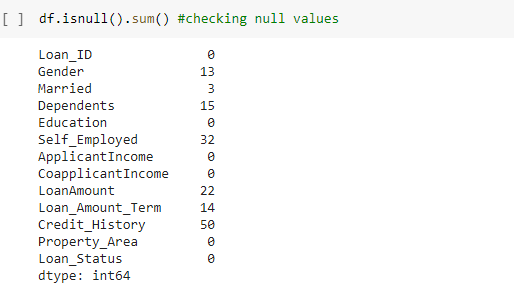


1. **Checking the datatypes of the dataset**



1. **Missing Values**

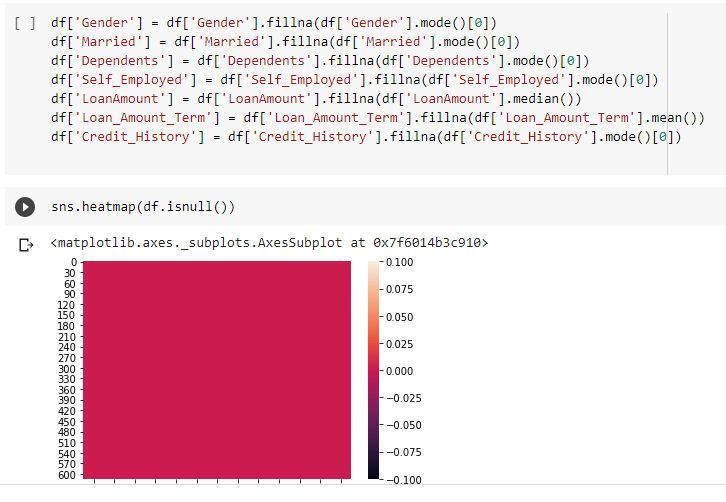
We need to check for missing values in the datasets.



There are 13 null values in gender , 3 in married, 15 dependents , 32 self employed,

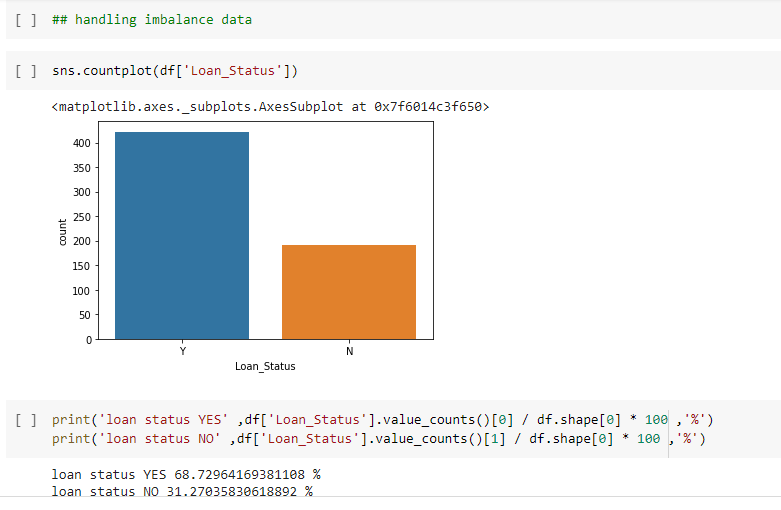
22 loan amount 14 loan amount term and 50 credit history data points , which are null values

1. **Fillling the NAN values**



Here all the nan values are filled with median and categorical variables are filled with mode

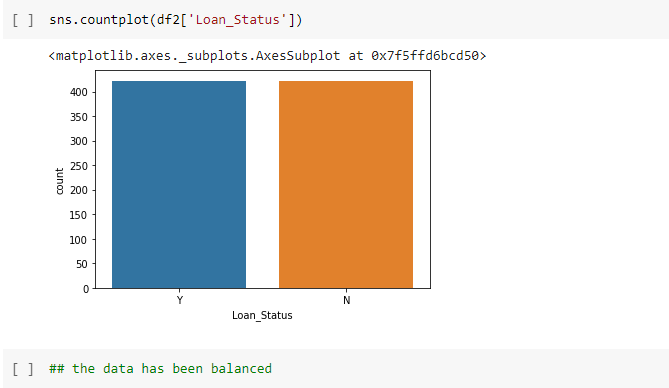
1. **Checking imbalance dataset**



It is clearly seen that ..the target variable with loan approved is 69% .. whereas those loans not approved are 31% . So the dataset should be make balanced ..here I used oversampler technique from random over sampler library

1. **Handling imbalaned dataset**

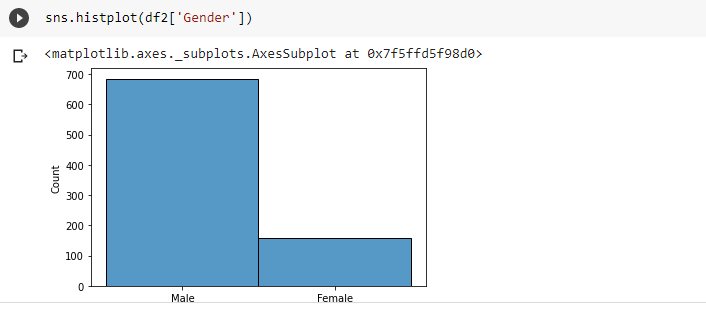




The data is balanced

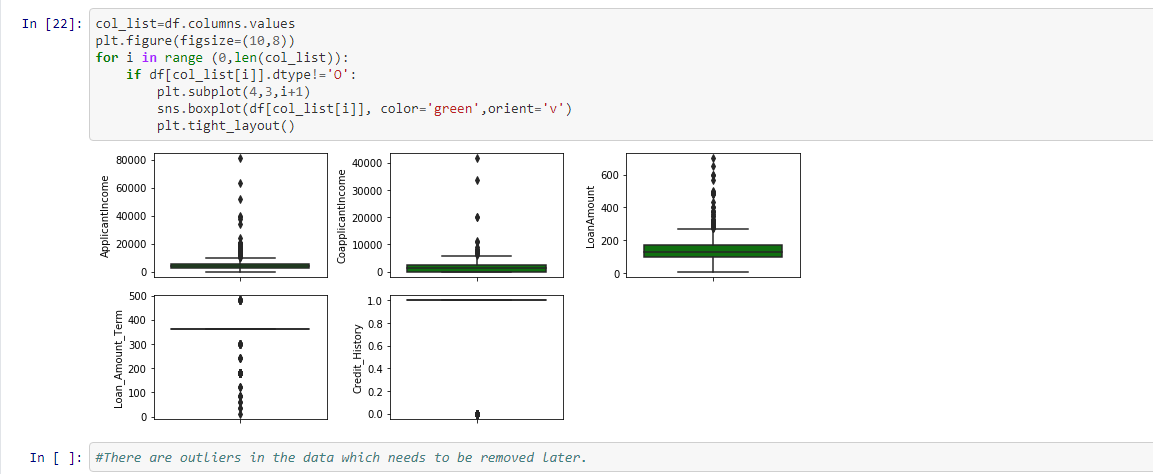
1. **Distribution of the data**

We can check how the data columns are distributed using hist plots. We can observe that many columns are not in symmetric distribution.



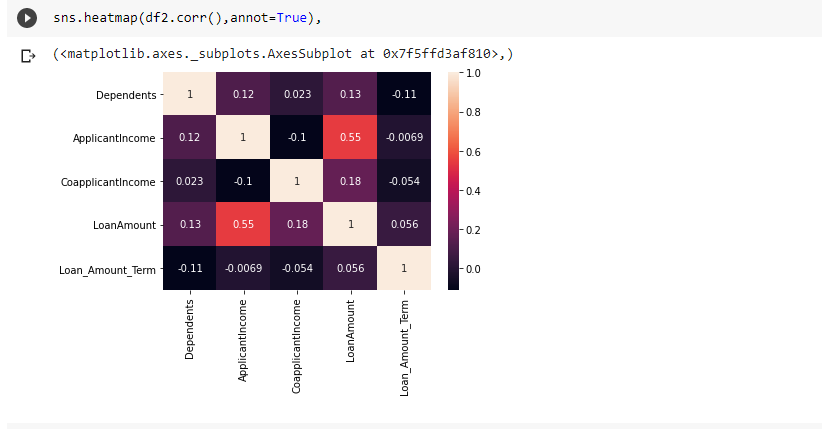
Observations:-

1. There are 489 male and 112 female applicants.
2. The married applicants are 398 and 213 are single
3. The graduate applicants are 480 and non graduate are 134.
4. 345 applicants have no dependents, 102 have one, 101 have two and 51 have 3+ dependents.
5. 475 have credit history and 89 have no credit history.
6. 233 have property in semi urban area, 202 in urban and 179 in rural area
7. 422 loan applications have been approved and 192 have been rejected
8. **Checking outliers**



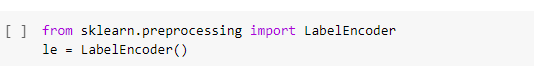
There are outliers in applicant income, capital income , loan amount ,and laon amount term

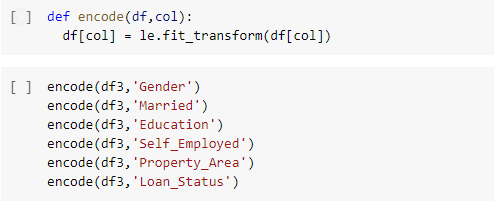
1. **Checking correlation**



none of the independent features are highly correlated with each other . hence no need to drop any features

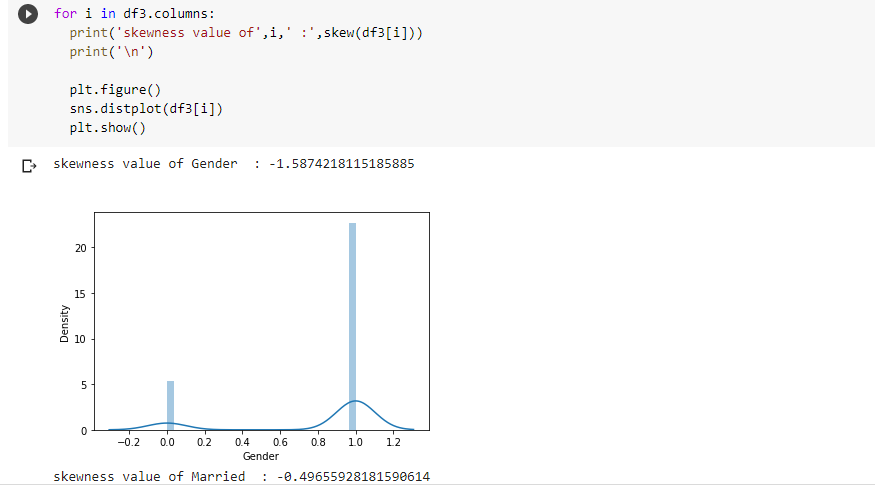
1. **Encoding the categorical features using LabelEncoder**





Categorical features are encoded

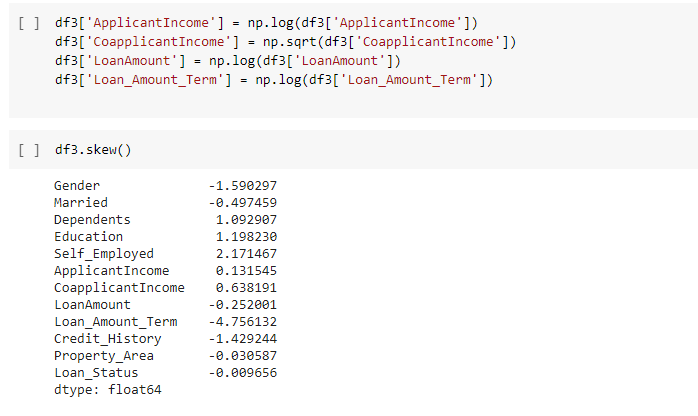
1. **Checking skewness**



there are skewness are most features …we should make it standard distributed to get a better

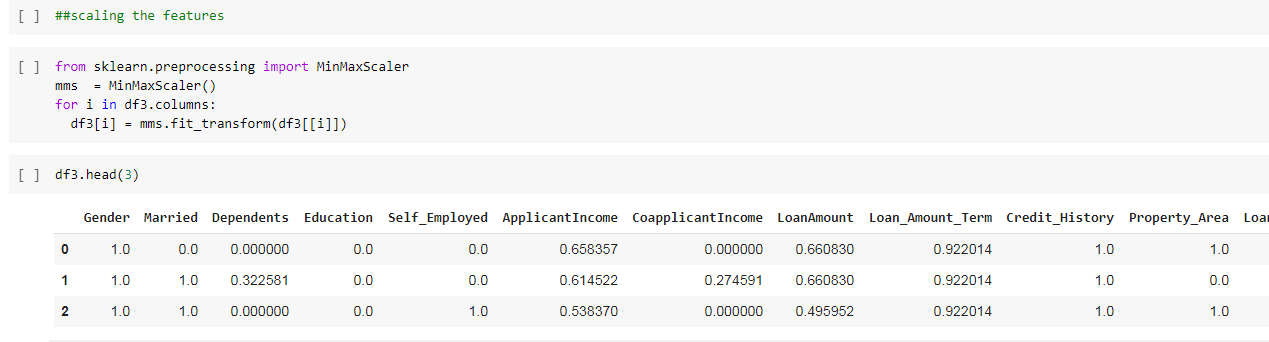
performing model

1. **Handling skewness**.



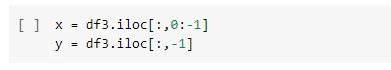
1. **Scaling the features.**

Before training the model with data, we need to scale the data and bring down the values within a normal range. There are two types of scalers:- Standard Scaler and Minmax scaler .Here we are using minmax scaler. This brings the values within the range of 0 to 1.



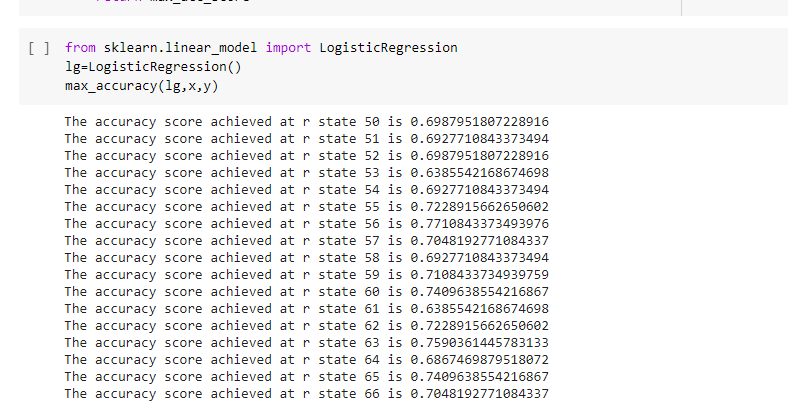
1. **Splitting the data into dependent and independent features.**

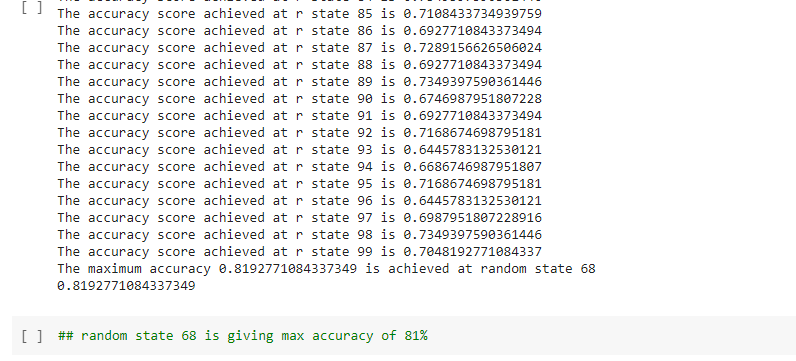
The data needs to be splitted into input and output variables. The input and output variables will be further split into training and test data. The training input and output will be used for training the model and testing input will be used to make predictions. Test output is only used to compare predictions of the model and is not fed to the model.



1. **Selecting best random state.**







The best random state is 68 ..which gives 81% accuracy

1. **Using GridSearchCV for best parameters**

We need to check the performances of other models. Before that we can use GridsearchCV to finalise the best parameters for the models.

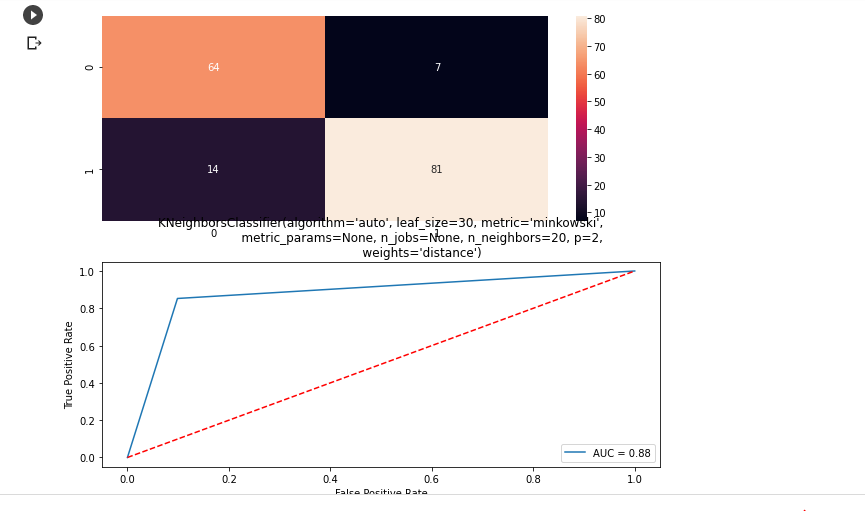




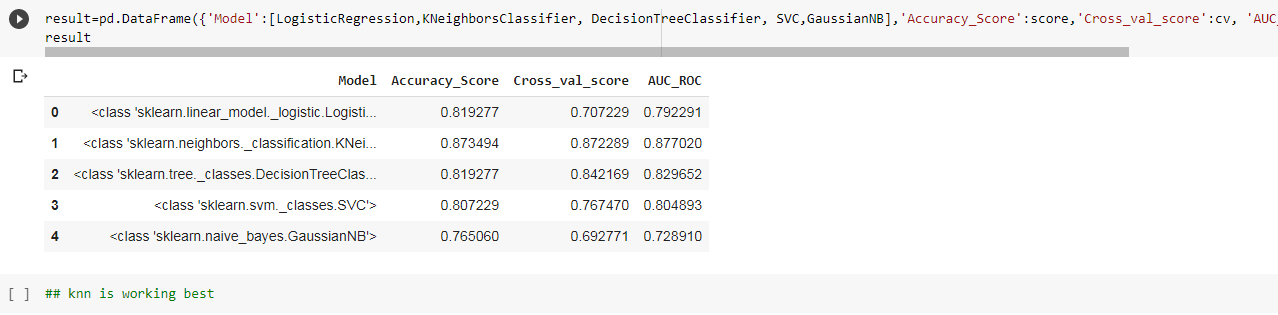
1. **Checking performance of all models**

After obtaining best parameters, we apply the same and obtain the accuracy score ,confusion matrix ,classification report and cross validation scores and other metrics of other models.



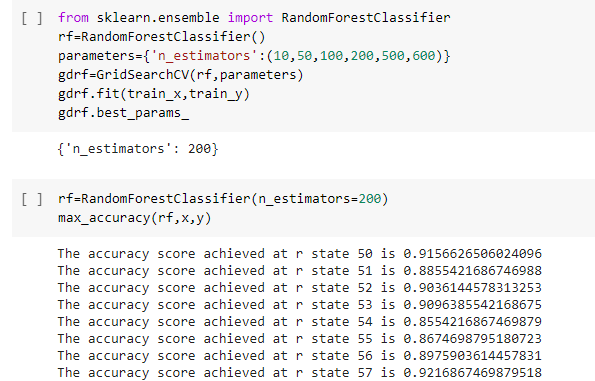


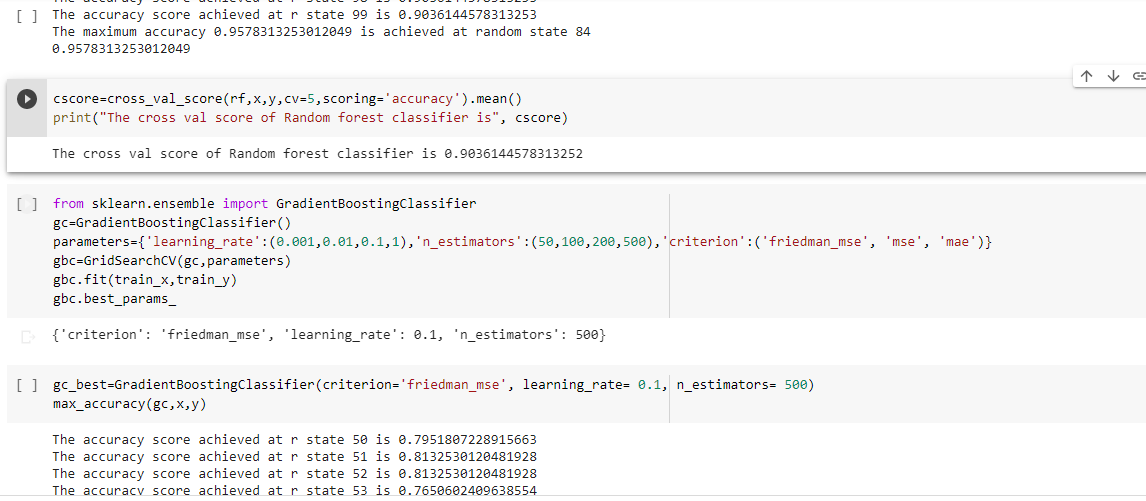
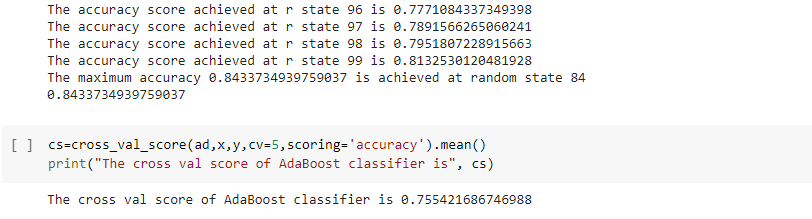
1. **Comparing the results of diffenrent models.**



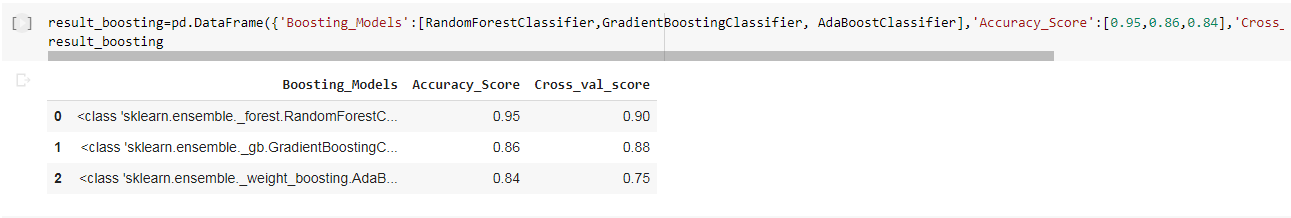
KNN is giving best results .. I will try to improve the accuracy using boosting algorithms

1. **Ensemble Techniques**



1. **Comparing boosting algorithms.**



From the above observation Random forest classifier is the best model .

**21. Finalizing the model and saving it**

After comparing results, we finalize Random forest classifier which is a good model and also is giving better accuracy score and cross validation score. And later we import the joblib library and save the model for future use.

