**LOAN APPLICATION STATUS PREDICTION**



**INTRODUCTION**

# One of the most important factors which affects our country’s economy and the financial condition is the credit system governed by the banks. The process of bank credit risk evaluation is recognising at banks across the globe. “As we know credit risk evaluation is very crucial, there are variety of techniques are used for risk level calculation. In addition, credit risk is one of the main functions of the banking community”. “In this paper we have taken the information of past clients of different banks to whom on a bunch of boundaries advance were endorsed. So, the AI model is prepared on that record to get precise outcomes as it is known to all that circulation of credits is the fundamental business of each bank.

# Loan Approval Prediction is extremely handy for employee of banks as well as for the applicant also. “The purpose of this Paper is to offer a quick, immediate, and simple method for selecting deserving applicants. “It is the responsibility of bank to ensure their assets is in the right hand. By implementation of this system we will be able to predict and ensure that applicant for the loan is safe or not by this automatic loan approval prediction system.

# Build predictive models to automate the process of predicting, if the loan of the applicant will be approved or not.

The above problem statement clearly explains that the target variable is categorical & it’s a classification problem as we need to predict if the loan of the applicant would be approved or not. So, this can be solved by any of the below Classification Machine learning algorithms:

The data in the datasets are handled by certain rules which are as follows:.

**1. Problem Definition.**

**2. Data Analysis.**

**3. EDA Concluding Remark.**

**4. Pre-Processing Pipeline.**

**5. Building Machine Learning Models.**

**6. Concluding Remarks.**

**PROBLEM DEFINITION:**

Sometimes it’s hard to get a loan approved after going through many manual documentations processes & it consumes a lot of time to apply for a loan. Here, we have a data set that includes details of applicants who have applied for loans. The dataset includes details like credit history, loan amount, income, dependents, etc.

**FEATURES:**

**Loan\_ID**: Loan ID for the Applicant applying for a loan

**Gender**: Gender of the Applicant

**Married**: Applicant’s marital status

**Dependents**: Number of dependents of the Applicant

**Education**: Applicant’s education status (Graduate/Under Graduate)

**Self\_Employed**: Applicant is self-employed or not

**ApplicantIncome**: Applicant’s Income

**CoapplicantIncome**: Co-applicant’s Income

**LoanAmount**: Loan Amount taken

**Loan\_Amount\_Term**: Term of the loan in months

**Credit\_History**: Applicant’s previous credit history meeting guidelines

**Property\_Area**: Urban, Semi-Urban, or Rural Areas

**Loan\_Status**: Loan Approval status (Target Variable)

* We have 614 rows & 13 columns in the dataset.
* We have maximum columns as categorical including labels & few are integer.

**Objective :**

Here, with the help of the applicant’s details, we would be able to predict whether the loan of the applicant will be approved or not on the basis of the details provided in the dataset.

The techniques in the machine learning is to improve the accuracy of detection on various imbalanced datasets. With the goal of generating higher predictive performance, the impact of feature engineering, feature selection, and parameter tweaking is also analyzed.

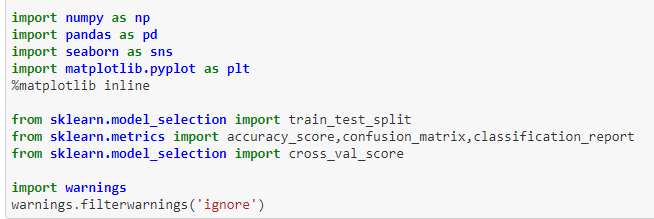
Predictive analysis involves certain steps to obtain accuracy detection

* **Training**
* **Testing**
* **Validation**

All this steps are involved along with the algorithm on the partial training datasets and then it is tested on some random splits .The data in the datasets are handled by certain rules.

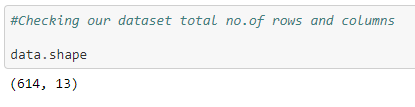
**DATA ANALYSIS**

**Importing Libraries:**



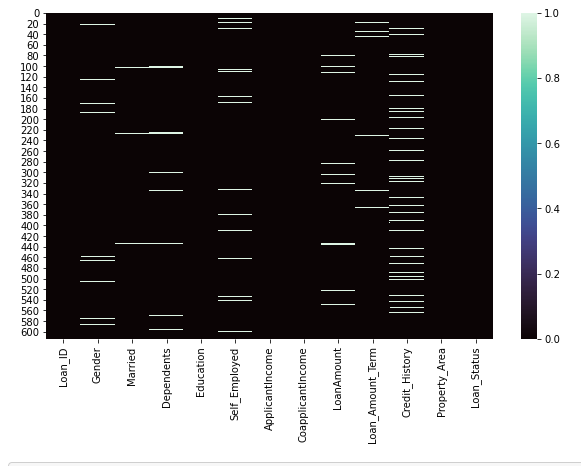
**Extracting dataset**



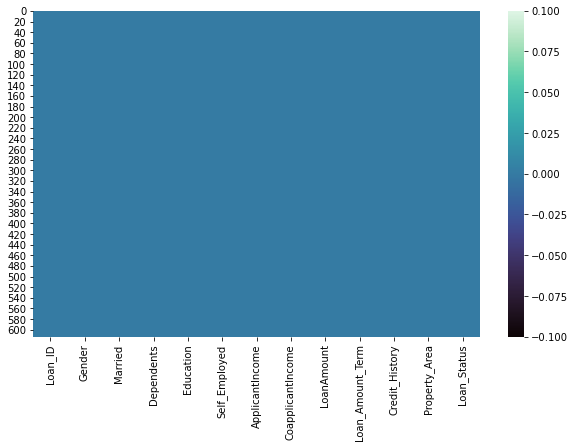


Dataset contains 614 rows and 13 columns. Using this dataset we will be training the Machine Learning models on 80% of the data and the models will be tested on 20% data.

***To check the total null values in all the columns individually***

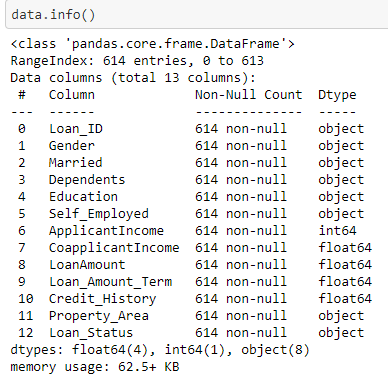


We have now replaced the column data null values with it’s respective “Modes”



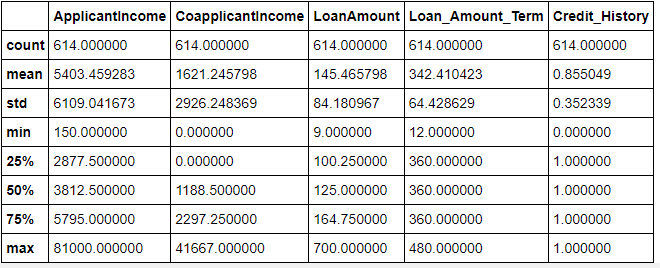
#### **Our dataset contains no null values. Hence, let's proceed further**

**Let’s check the datatypes of our dataset columns**



Our dataset contains both Categorical and Numerical features.

**Statistical information of each feature:**



* Columns: "ApplicantIncome", "CoapplicantIncome" has large standard deviation than their respective column mean And
* "LoanAmount" has also some outliers
* We would need to remove the outliers and check for the skewness presence

# **Exploratory Data Analysis (EDA)**

Exploratory Data Analysis is evidently one of the most important steps during the entire process of extracting insights out of data, even before the actual analysis or modeling begins.

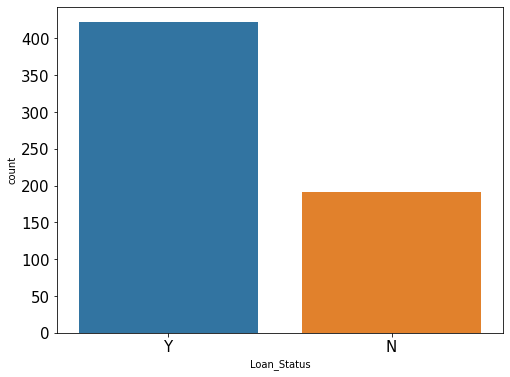
EDA techniques allow for effective manipulation of data sources, enabling data scientists to find the answers they need by discovering data patterns, spotting anomalies, checking assumptions, or testing a hypothesis.

Let’s explore the features in our dataset and know more about the data requirement , it’s importance and proceed further with the outliers and skeness removal and then, our data modeling.

### **Univariate Analysis**

Exploring a single column seperatly and analysing the data structure

**Target column: (loan\_Status: Yes/No)**



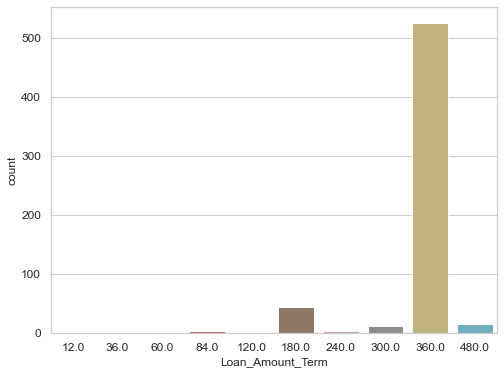
#### **Observation:**

- In our Dataset more than half of the loan applications are approved

- There 2:1 chance of the loans getting approved and rejected

Our label column is Imbanced and we would need to balance it by using "Under sampling/Over Sampling method"

**Loan\_Amount\_Term:**



#### **Observation:**

- "Loan\_Amount\_Term : 360.0" has the highest loan claims

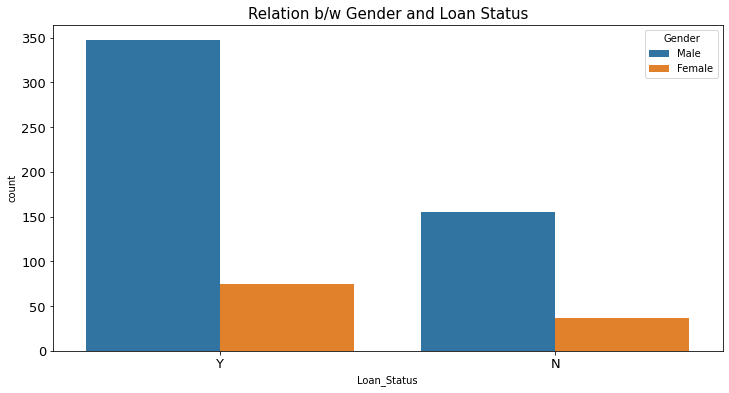
- Then we have “180.0” loan\_amount\_term standing at second maximum count of loan applications.

- Least we have “12.0”, “36.0”, “60.0”, “120.0” loan\_amount terms with least loan applications.

**Bivariate Analysis**

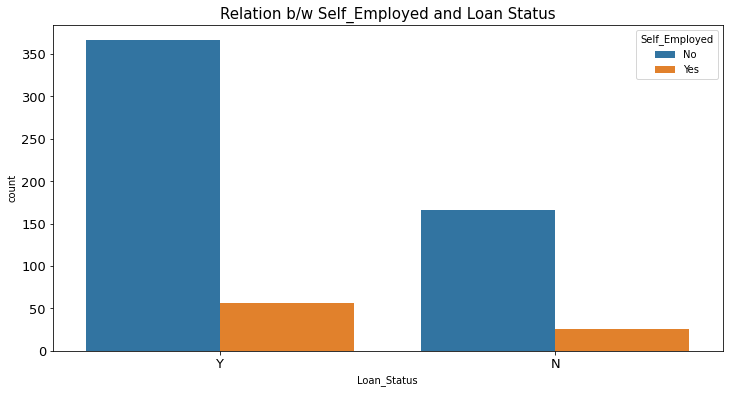
EDA methods illustrate relationships between 2 features using different plots.

**Relation b/w Gender and Loan Status :**

 **Observation:**

- Male Candidates have higher chance of getting the loan approval and rejection when compared to female candidates

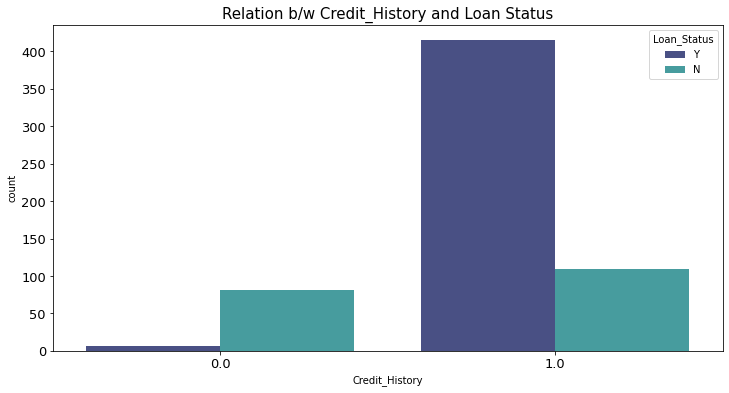
**Relation b/w Self\_Employed and Loan Status :**



#### **Observation:**

- We see that there are still high chances that a non self employed candidate also gets their loan applications approved

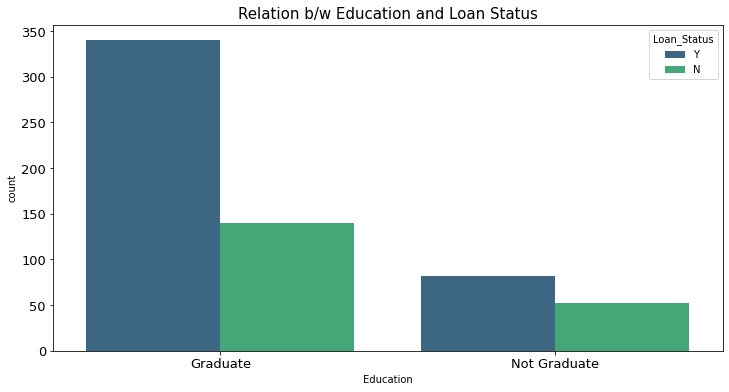
**Relation b/w Credit\_History and Loan Status :**



#### **Observation:**

- Loan Applications are still approved for the candidates who have previous Credit History

**Relation b/w Education and Loan Status :**

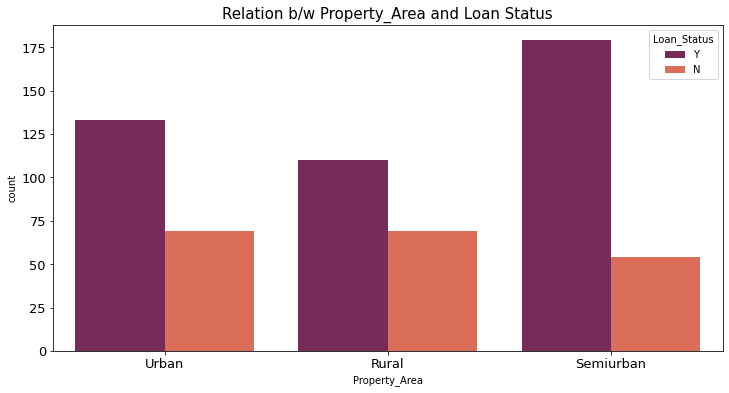


#### **Observation:**

- High Chances of Loan Applications approval of the Graduated candidates when compared to Not Graduated candidates

Hence, the Graduated populates gets the more loans facility

**Relation b/w Property\_Area and Loan Status :**

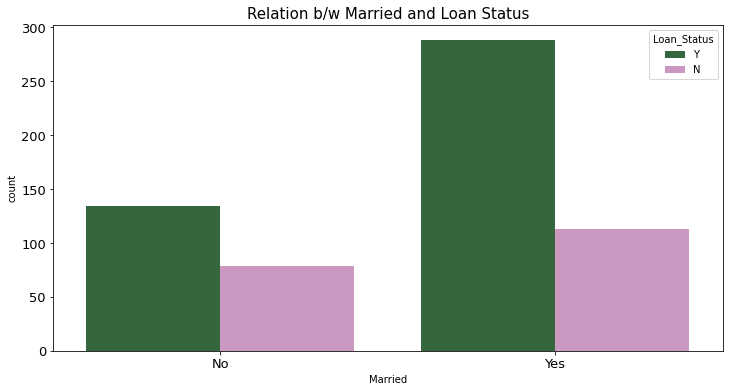


#### **Observation:**

- Semiurban area property candidates have high chances of loan approval and Urban property candidates also gets more number of loans approved

- Rural Area candidates have high chances of Loans rejected and less chances of loans approvals when compared to the Semiurban and Urban area property candidates

**Relation b/w Married and Loan Status :**



#### **Observation:**

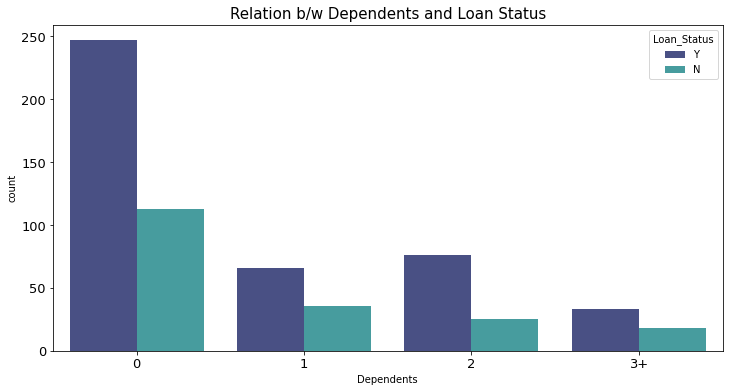
- Married Candidate's Loan Applications are largely approved when compared to unmarried candidates

- And they are rejected as well highly. But are mostly approved.

- Unmarried Candidate’s Loan Approval possibilities are almost

like 50-50 %

**Relation b/w Dependents and Loan Status :**

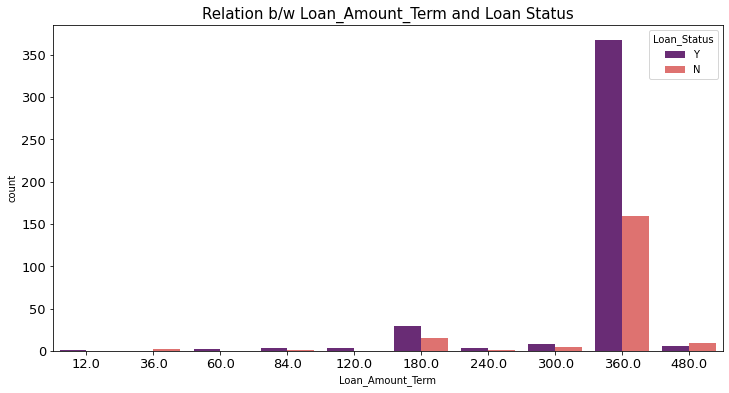


#### **Observation:**

- Candidates with no dependents, high chances of loan approvals

- Candidates with more than 3 dependents, least chances of loan approvals

**Relation b/w Loan\_Amount\_Term and Loan Status :**



#### **Observation:**

- Loan\_Amount\_Term "360.0" has the highest rate of loan approvals and rejections, when compared to other Loan\_Amount\_Terms. But they are mostly approved loans

It means the "Loan\_Amount\_Term" of "360.0" is issued and approved the most

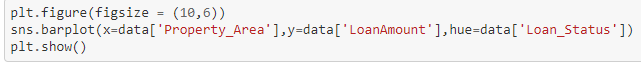
### **Multivariate Analysis**

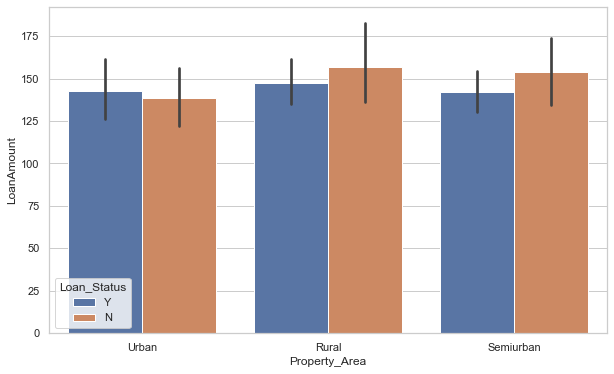
This EDA technique makes use of graphics to show relationships

between 2 or more datasets. The widely-used multivariate graphics

include bar chart, bar plot, heat map, bubble chart, run chart,

multivariate chart, and scatter plot.





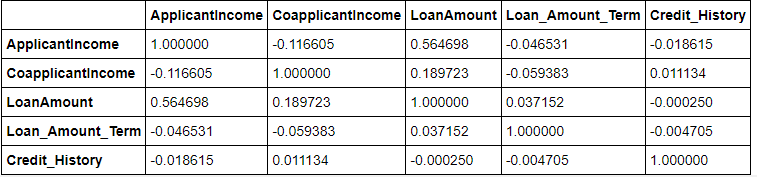
#### **Observation:**

- Rural Property Area gets the highest loan amount approved

**Checking the correlation between features using Heatmap**



**CORRELATION TABLE**



#### **Observation:**

- CoapplicantIncome has the least correlation with

Loan\_Amoun\_Term value: "-0.059383"

- LoanAmount and ApplicantIncome have large correlation of

value: "0.564698"

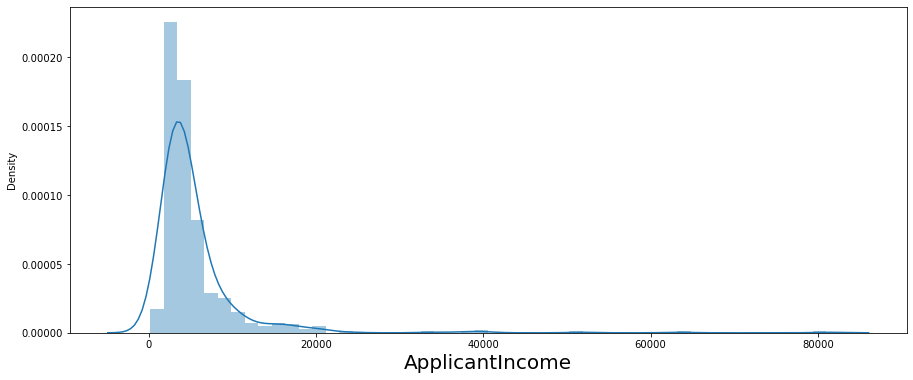
**DATA PRE-PROCESSING**

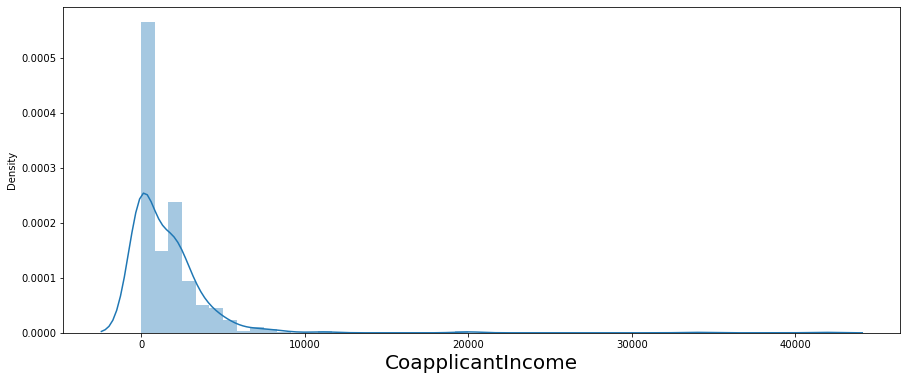
*#Column with Continuous data which ay contain the outliers*

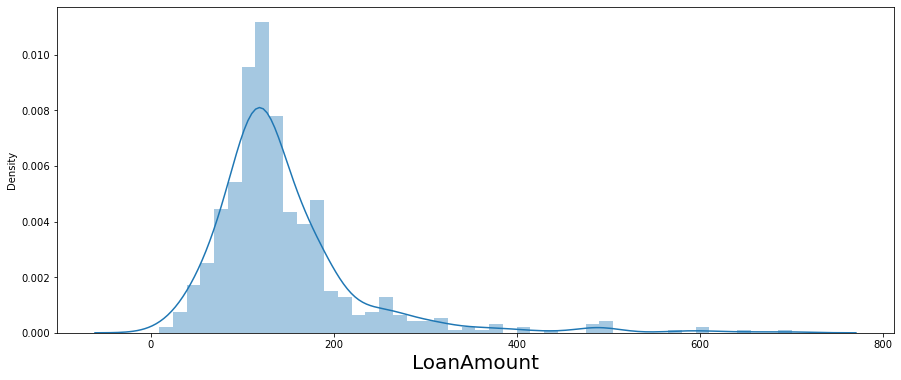
features = ['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount']

One of the reasons we want to check for outliers is to confirm the quality of our data. One of the potential sources for outliers in our data are values that are not correct. There are different potential sources for these “incorrect values”. Two potential sources are missing data and errors in data entry or recording.

**Let’s explore the Normal Distribution to check the skewness in our data columns**







* We see that our data set columns ['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount'] are mostly skewed towards right

### **Removing the outliers**

Let's apply some techniques to remove the outliers from the above 3 columns

##### **Applying the IQR Method**

Shape - Before and After:

Shape Before : (614, 13)

Shape After : (564, 13)

Percentage Loss : 8.143322475570033

##### **Applying the Z-Score Method**

Shape - Before and After:

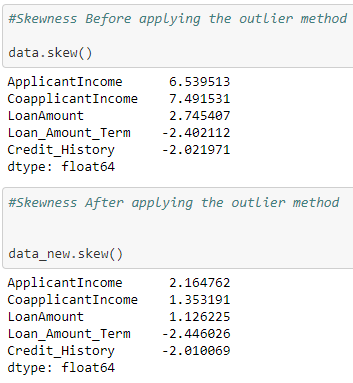
Shape Before : (614, 13)

Shape After : (589, 13)

Percentage Loss : 4.071661237785016

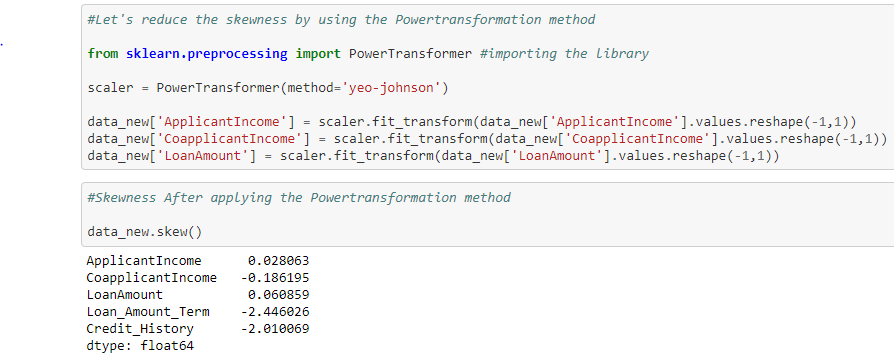
Percentage Loss is very less by applying z-score method. let's proceed with the Z-Score method.

### **Skewness**



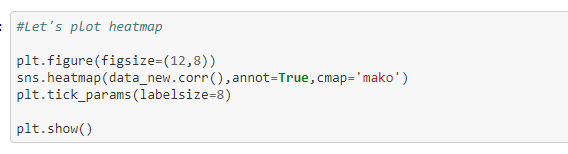
* "Loan\_Amount\_Term" looks like a descrete data so we will ignore the skewness in it
* "Credit\_History" looks like a nominal data with value 0 or 1 so we will ignore the skewness in it

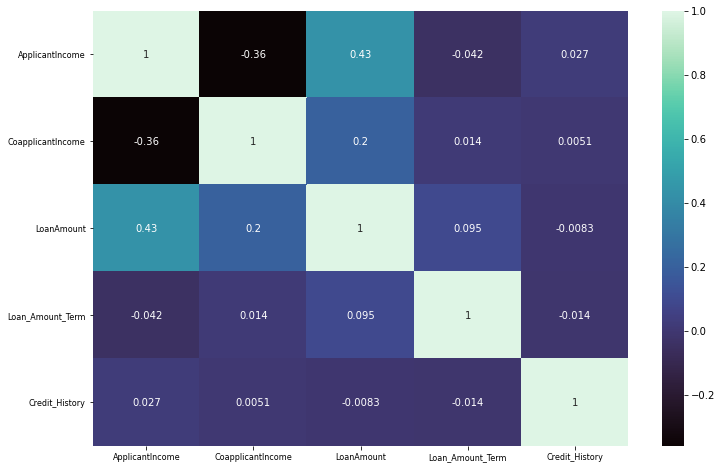
**Let’s apply “Power\_Transformation method to remove the skewness for the rest columns**



**We could see that the skewness are removed**

**Let’s check the correlation between features by plotting Heatmap**





**Here, No features has the value closed to “1”. There is no high correlated features.**

**- “ApplicantIncome” has more correlation with “LoanAmount” with value “0.43”**

## **Encoding Data**

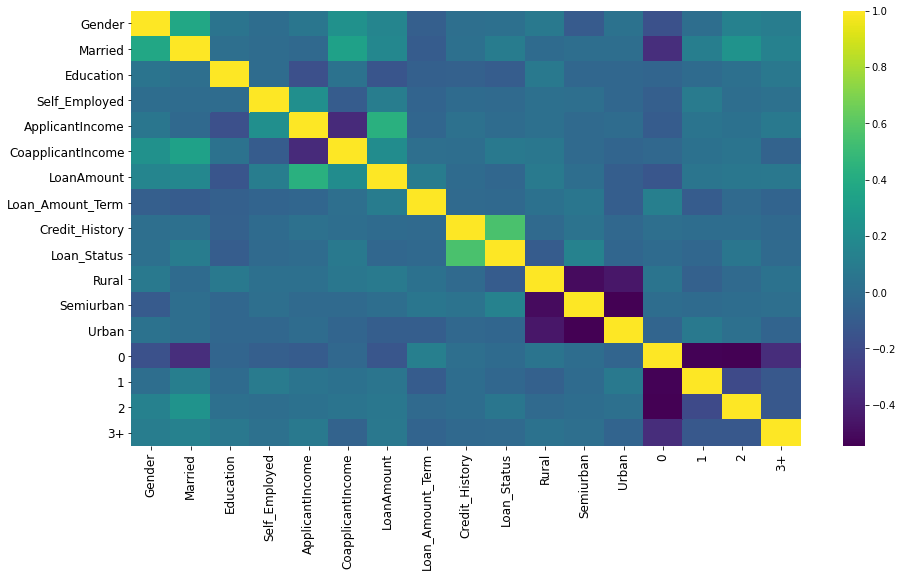
****Encoding**** Categorical****data**** in****Machine Learning**** Most of the****Machine Learning**** Algorithms accepts only Numerical****data**** as input.

For the below Categorical Columns, Label encoding method has been used to convert the data into Numerical data.

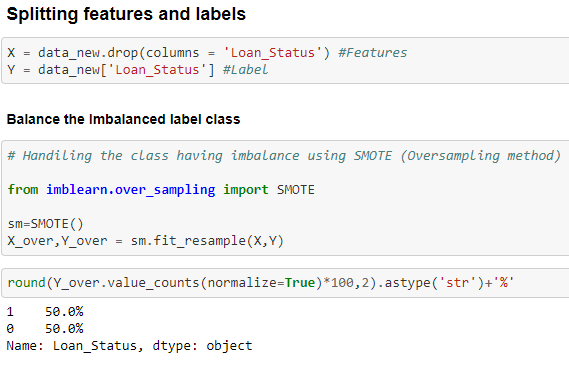
l1 = ['Gender', 'Married','Education','Loan\_Status','Self\_Employed']

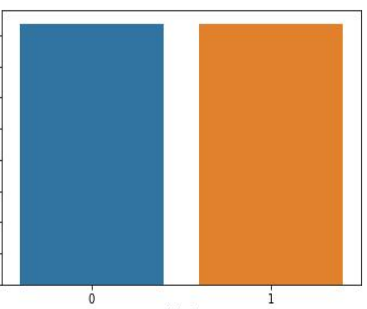
For “Property\_Area” and “Dependents”, the dummies have been created.

**Let’s plot Heatmap for the whole data columns**

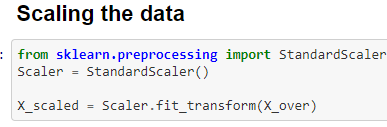


We found that Correlation between features v/s features are not large, so we can now proceed further with our data modelling.





Our Target column is now balanced. So, we can now start building the model by standardizing the data values and splitting the training and testing data



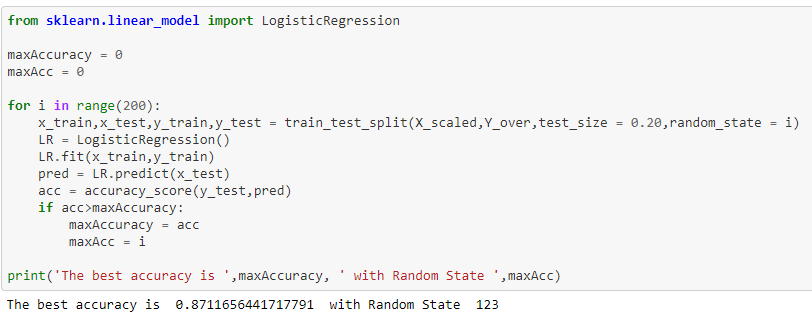
We have applied Standard Scaler method so that all our data columns values are normally distributed/equal.

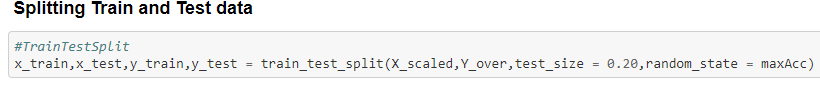
Further, before building the model we will have to split the data to test and train. The best possible way to split the data is by finding the best random state to split and the benefit is that we can control over fitting up to certain extent before even building the model.

We are trying to match the accuracy score of the training data set and the test dataset, which ever split (**random state**) satisfies the condition (**accuracy score of training dataset = accuracy score of testing dataset**).

We’ll take the same random state to split the dataset and build the model.

### **Finding the Best Random State**





MACHINE LEARNING MODELS

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

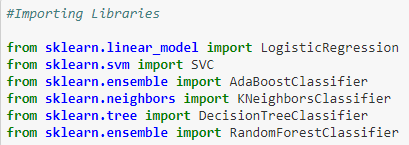
Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

In the machine learning world, model training refers to the process of allowing a machine learning algorithm to automatically learn patterns based on data.

Let’s build some machine learning models and then, choose the best model for our dataset target predictions.

**Importing Libraries**

Let’s import the required libraries for model building



Since the dataset is large to my system configurations, ensemble techniques will be efficient although I’m testing the results with the below algorithms.

1. Logistic Regression

2.Random Forest Classifier

3. Decision Tree Classifier

4. KNeighbors Classifer

5. AdaBoost Classifier

6. SVC

7.BernoulliNB

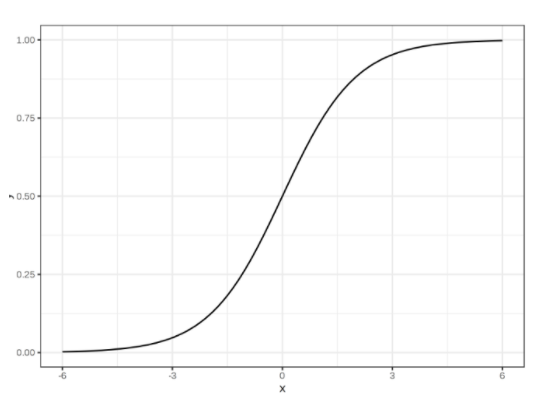
In order to test the model, I’m using accuracy score, F1 score, further in order to verify the model’s fit, I’m using cross validation score to identify the best model.

Let’s start building the models

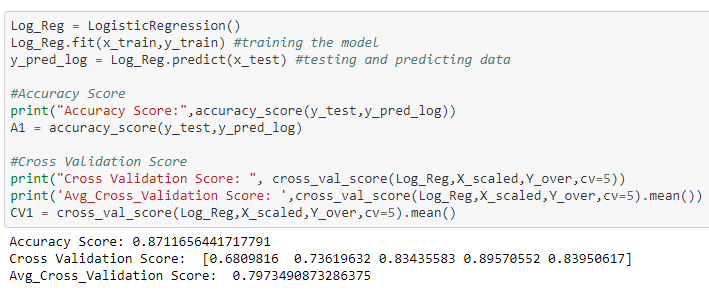
### **Logistic Regression Model:**

****Logistic regression**** is a statistical model that in its basic form uses a **logistic** function to model a binary dependent variable, although many more complex extensions exist. In **regression** analysis, **logistic regression** (or logit **regression**) is estimating the parameters of a **logistic** model (a form of binary **regression**)

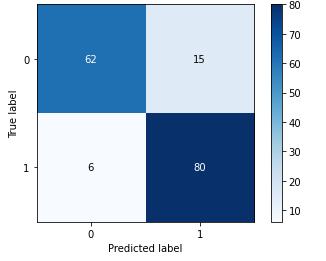
A logistic equation is created in such a way that the output values can only be between 0 and 1 (see below).



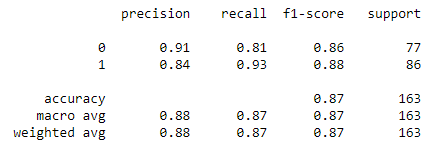
**Let’s Build the **Logistic regression** Model**



**CONFUSION MATRIX**



**CLASSIFICATION REPORT**



By  **Logistic Regression Model,** we were able to get the accuracy score of 0.8711

Cross\_Validation\_Score: 0.7973

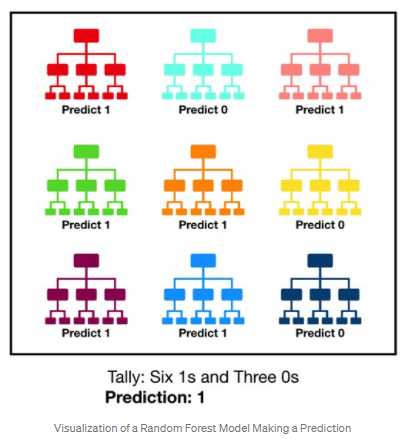
Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting.

F1 score for Logistic Regression Model is more than 86%

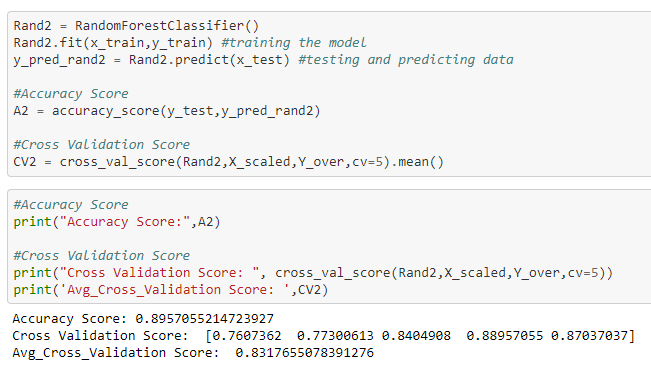
### **Random Forest Classifier Model:**

**Random forests** are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) technique that builds off of decision trees. Random forests involve creating multiple decision trees using [bootstrapped datasets](https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/) of the original data and randomly selecting a subset of variables at each step of the decision tree. The model then selects the mode of all of the predictions of each decision tree.

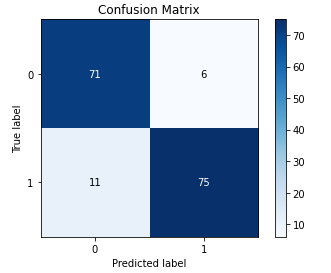
In short, Each individual tree in the random forest spits out a ****class prediction**** and the class with the most votes becomes our model’s prediction.



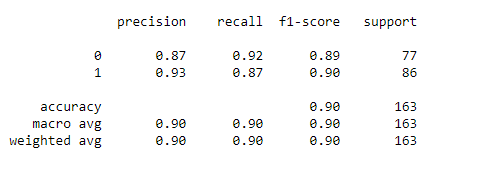
**Let’s Build the Random Forest Classifier Model**



**CONFUSION MATRIX**



**CLASSIFICATION REPORT**



By **Random Forest Classifier model,** we were able to get the accuracy score of 0.8957.

Cross\_Validation\_Score: 0.8317

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting.

**F1 score** for **Random Forest Classifier model** is more than 89%

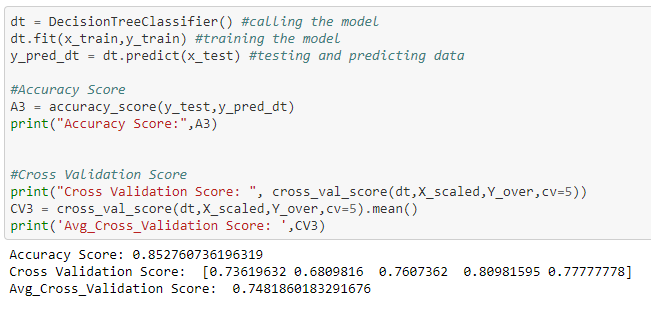
### **Decision Tree Classification Model:**

****Decision trees**** are a popular model, used in operations research, strategic planning, and machine learning. Each square above is called a ****node****, and the more nodes you have, the more accurate your decision tree will be (generally). The last nodes of the decision tree, where a decision is made, are called the ****leaves**** of the tree. Decision trees are intuitive and easy to build but fall short when it comes to accuracy.

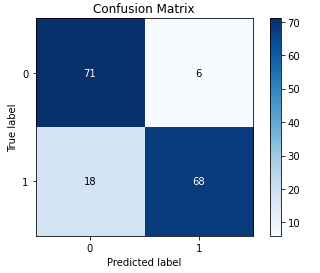
**For an example, Decision Tree Classification Model looks like:**



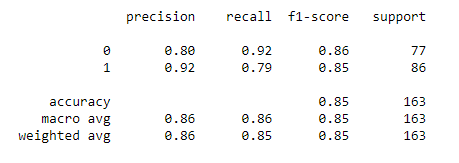
**Let’s Build the Decision Tree Classification Model**



**CONFUSION MATRIX**



**CLASSIFICATION REPORT**



By **Decision Tree Classifier model,** we were able to get the accuracy score of 0.8527.

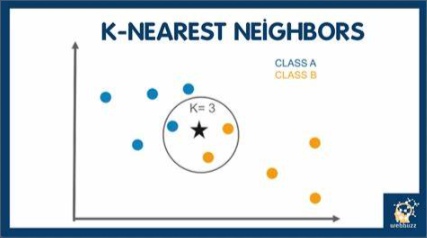
Cross\_Validation\_Score: 0.7481

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

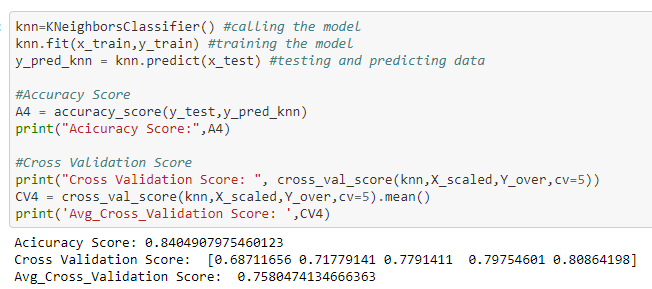
**F1 score** for **Decision Tree Classifier model** is more than 85%

### **KNeighbors Classification Model:**

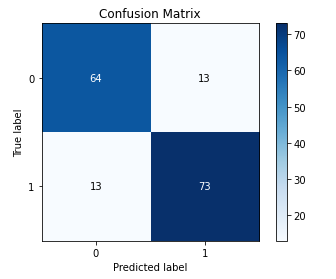
The k-neighbors is commonly used and easy to apply classification method which implements the k neighbors queries to classify data. It is an instant-based and non-parametric learning method. In this method, the classifier learns from the instances in the training dataset and classifies new input by using the previously measured scores.



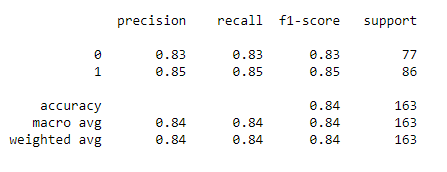
**Let’s Build the KNeighbors Classification Model**



**CONFUSION MATRIX**



**CLASSIFICATION REPORT**



By **KNeighbors Classification model,** we were able to get the accuracy score of 0.8404.

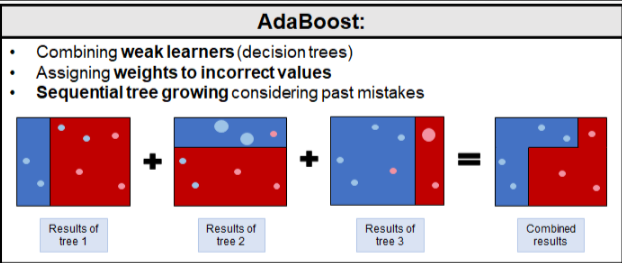
Cross\_Validation\_Score: 0.7580

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

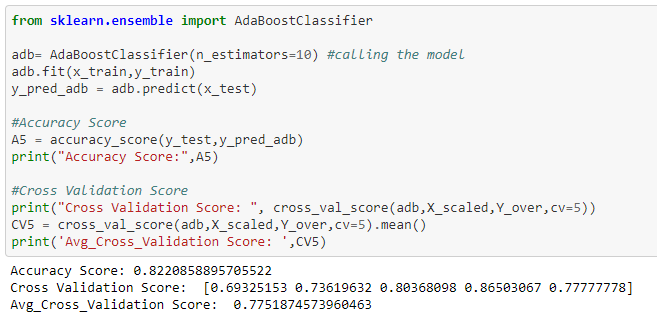
**F1 score** for **KNeighbors Classification model** is more than 83%

### **AdaBoost Classification Model:**

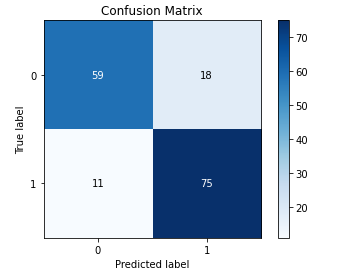
An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.



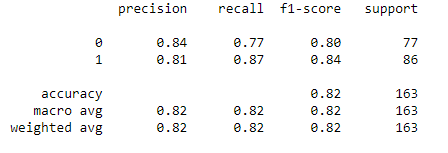
**Let’s Build the AdaBoost Classification Model**



**CONFUSION MATRIX**



**CLASSIFICATION REPORT**



By **AdaBoost Classification model,** we were able to get the accuracy score of 0.8220.

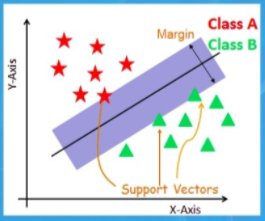
Cross\_Validation\_Score: 0.7751

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

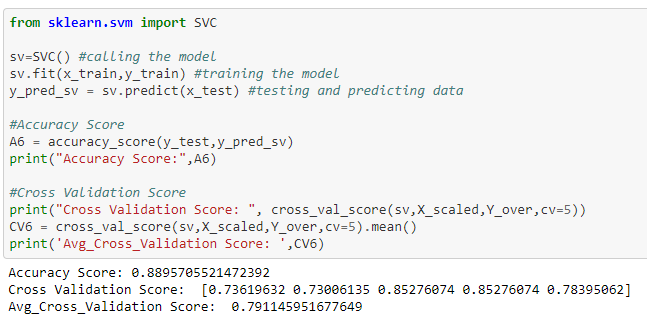
**F1 score** for **AdaBoost Classification model** is more than 80%

### **SVC Model:**

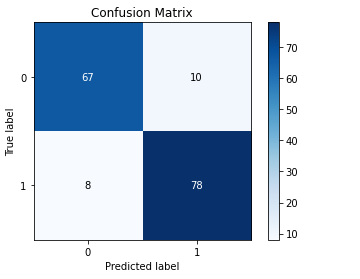
Support Vector Machine algorithm is a simple yet powerful Supervised Machine Learning algorithmthat can be used for building****both regression and classification models.**** Support Vector Machine algorithm can perform really well with both linearly separable and non-linearly separable datasets. Even with a limited amount of data, the support vector machine algorithm does not fail to show its magic.



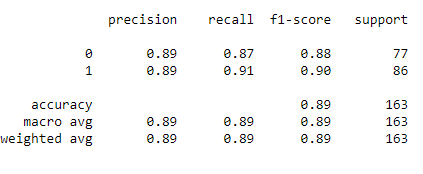
**Let’s Build the SVC Classification Model**



**CONFUSION MATRIX**



**CLASSIFICATION REPORT**



By **SVC Classification model,** we were able to get the accuracy score of 0.8895.

Cross\_Validation\_Score: 0.7911

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

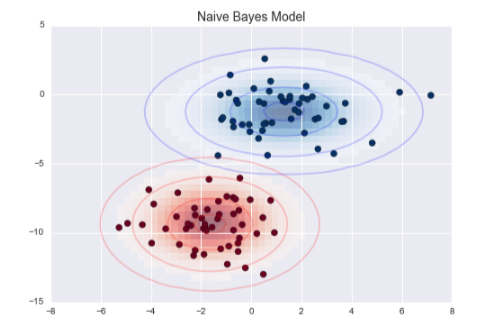
**F1 score** for **SVC Classification model** is more than 88%

### **Bernouli NB Classification Model:**

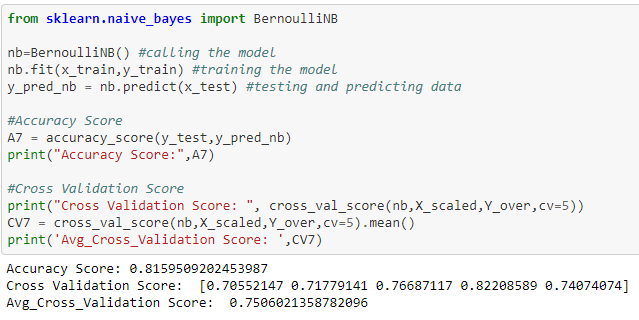
Naive Bayes classifier for multivariate Bernoulli models.

BernoulliNB is designed for binary/boolean features.

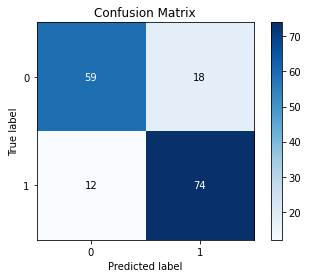
In ****Bernouli NB (BNB) Classifier****, the word when appears in the document, the value of the attribute equivalent to that word is written either one otherwise zero



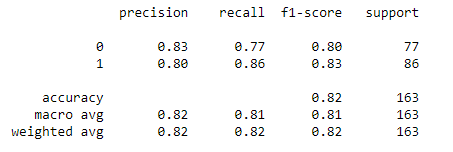
**Let’s Build the **Bernouli NB (BNB)** Classification Model**



**CONFUSION MATRIX**



**CLASSIFICATION REPORT**



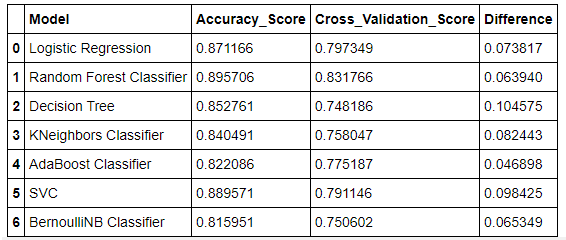
By ****Bernouli NB** Classification Model,** we were able to get the accuracy score of 0.8159.

Cross\_Validation\_Score: 0.7506

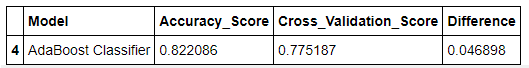
Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

**F1 score** for ****Bernouli NB** Classification Model** is more than 80%

**Let's check for our Overall scores of our models:**



Find the row with least difference



We have built many models,

But to choose the best model we checked the difference of the accuracy score and Cross Validation score

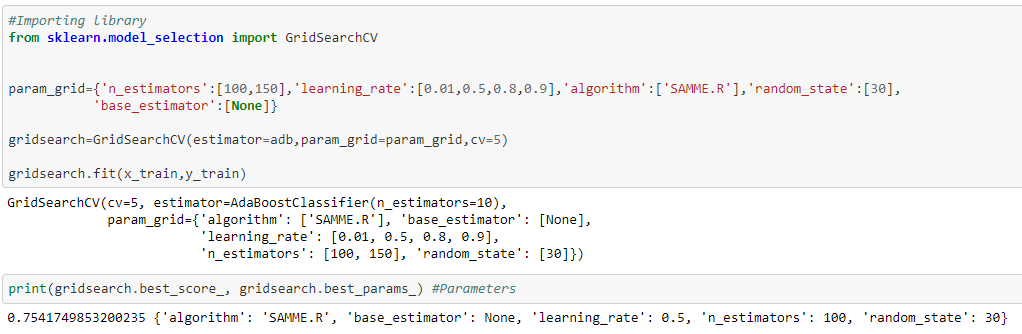
And we found that the "AdaBoost Classifier" has the minimum difference

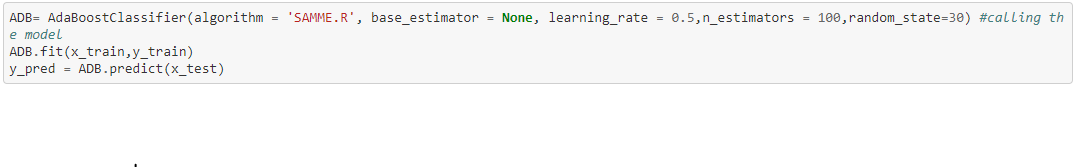
Hence, "AdaBoost Classifier" is our best model with **“ 82.20% ”** accuracy score

## **Hyper Parameter Tuning**

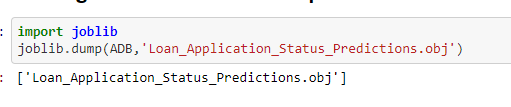
Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are: GridSearchCV. RandomizedSearchCV. GridSearchCV. In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values.

Let's Hyper tune our model to increase the accuracy score. And here are the best parameters that we have found





## **Saving the model for future prediction:**



### **Performance Metrics**

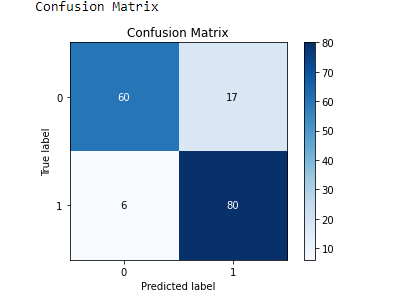
**Accuracy Score: 0.8588957055214724**

**Precision: 0.8247422680412371**

**Recall: 0.9302325581395349**

**F1 score: 0.8743169398907104**

**ROC\_AUC\_SCORE : 0.8547266686801571**



**Classification Report:**

**precision recall f1-score support**

**0 0.91 0.78 0.84 77**

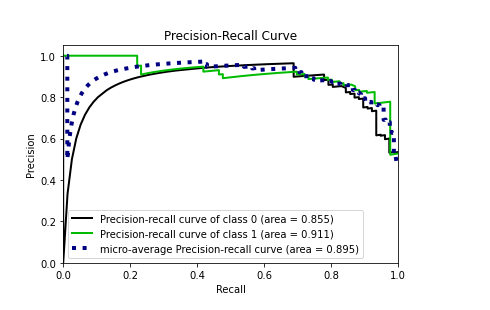
**1 0.82 0.93 0.87 86**

**accuracy 0.86 163**

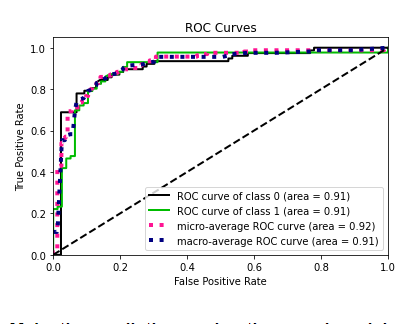
**macro avg 0.87 0.85 0.86 163**

**weighted avg 0.86 0.86 0.86 163**

**Precision Recall Curve**

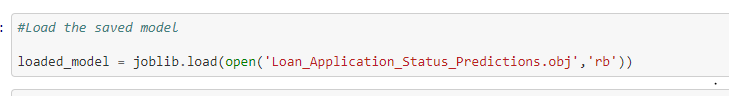


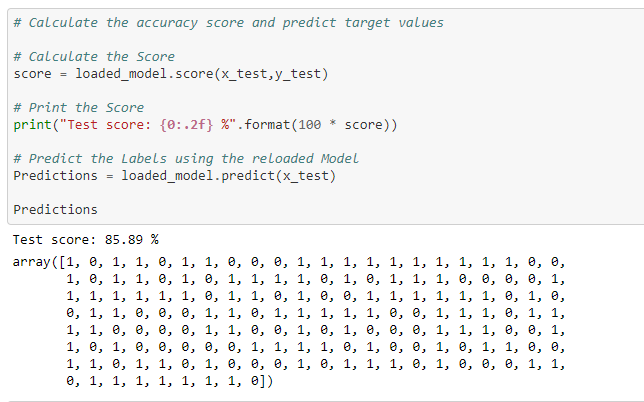
ROC Curve



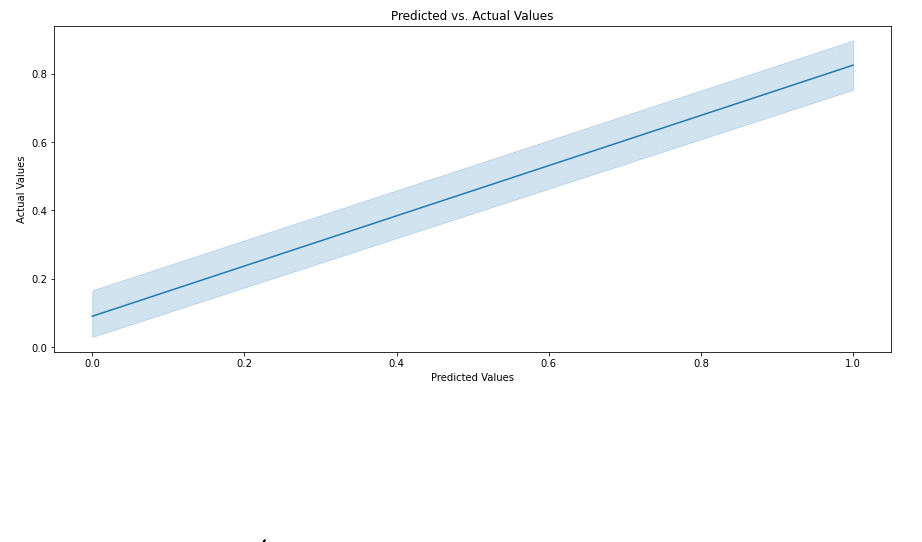
### **Make the predictions using the saved model**

Let’s load the saved model and get the predicted values for the test data and then compare them with the real/actual target value





***#plot predicted vs. actual values***



## **Our model is now ready to predict Loan Claim Status**

## **"accuracy score: 85.89 %”**

## **“F1 score: 87.43%"**

**CONCLUSION**

As we have seen, the prediction is showing a similar relationship with the actual loan status from the train data set, which means the model predicted correctly & this could help banks to save time & predict which all customers could get the approval based on these features. It could also help customers to predict if they would be able to apply for a loan or not based on the bank requirements and accordingly, they would be able to apply for a loan, and by using these model both banks & customers can save time as this process of applying loan is hectic & time-consuming. Hence by using Machine learning techniques we can solve this problem & reduce manual efforts.

**Requirement**:I have prepared a prediction model for predicting loans for banks. The accuracy can be improved by adding more data into the training model. If further worked on this project, it can be converted into a tool with some user interface which could be beneficial, time saving and more accurate therefore removing human errors.

For more and clear steps please use the below links to access the codes and results:

<https://github.com/PoonamRajput16/DataTrained_Evaluation_Projects/blob/main/Loan%20Application%20Status_Predictions.ipynb>