

MICRO-CREDIT DEFAULTER MODEL

Submitted by:

HIMANSHU SHARMA

**ACKNOWLEDGMENT**

One of the pleasant aspects of preparing a project report is the opportunity to thank those who have contributed to make this project possible.

We are extremely thankful to Mrs.Astha Mishra, whose active interest in the project & insight helped us to formulate, redefine implement our approach to the project.

We are also thankful to our institute & Other seen unseen hands which have given us direct & indirect help in completion of this project.

* **Himanshu Sharma**

**INTRODUCTION**

* **Business Problem Framing**

This project is related to a Microfinance Institution (MFI) is an organization that offers financial services to low income populations. Microfinance services (MFS) becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The MFS provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on.

In given dataset, about Telecom Industry from Indonesia. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber.

Problem in this dataset, build a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of issuance of loan. In this case, Label ‘1’ indicates that the loan has been payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed i.e. defaulter.

* **Conceptual Background of the Domain Problem**

Conceptual Background in this dataset, i.e.We will use Statistics and machine learning algorithms to solve the dataset. Since the target value is classified, we will use the classification model algorithms in it. To visualize the data, we will use matplotlib and seaborn library, which based on python language.

* **Review of Literature**

Review of Literature is basically related to comprehensive summary of dataset as well as descriptions of input variables and output variable.



* **Motivation for the Problem Undertaken**

Motivation for the problem undertaken because project scenario basically related to predict label in team of defaulter and non-defaulters in telecom company. through this project we can easily find the defaulter, which is cause of loss of telecom company.

**Analytical Problem Framing**

* **Mathematical/ Analytical Modelling of the Problem**

In this section we understand and describe the mathematical, statistical and analytics modelling done during this project along with the proper justification. First of all for better understanding about dataset and given attributes information we use **Dataframe.info()** command, which tell me, The total number of attributes, what is name of attributes, datatype of attributes and how many Non-null values are present in dataset.

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 209593 entries, 0 to 209592

Data columns (total 36 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 label 209593 non-null int64

1 msisdn 209593 non-null object

2 aon 209593 non-null float64

3 daily\_decr30 209593 non-null float64

4 daily\_decr90 209593 non-null float64

5 rental30 209593 non-null float64

6 rental90 209593 non-null float64

7 last\_rech\_date\_ma 209593 non-null float64

8 last\_rech\_date\_da 209593 non-null float64

9 last\_rech\_amt\_ma 209593 non-null int64

10 cnt\_ma\_rech30 209593 non-null int64

11 fr\_ma\_rech30 209593 non-null float64

12 sumamnt\_ma\_rech30 209593 non-null float64

13 medianamnt\_ma\_rech30 209593 non-null float64

14 medianmarechprebal30 209593 non-null float64

15 cnt\_ma\_rech90 209593 non-null int64

16 fr\_ma\_rech90 209593 non-null int64

17 sumamnt\_ma\_rech90 209593 non-null int64

18 medianamnt\_ma\_rech90 209593 non-null float64

19 medianmarechprebal90 209593 non-null float64

20 cnt\_da\_rech30 209593 non-null float64

21 fr\_da\_rech30 209593 non-null float64

22 cnt\_da\_rech90 209593 non-null int64

23 fr\_da\_rech90 209593 non-null int64

24 cnt\_loans30 209593 non-null int64

25 amnt\_loans30 209593 non-null int64

26 maxamnt\_loans30 209593 non-null float64

27 medianamnt\_loans30 209593 non-null float64

28 cnt\_loans90 209593 non-null float64

29 amnt\_loans90 209593 non-null int64

30 maxamnt\_loans90 209593 non-null int64

31 medianamnt\_loans90 209593 non-null float64

32 payback30 209593 non-null float64

33 payback90 209593 non-null float64

34 pcircle 209593 non-null object

35 pdate 209593 non-null datetime64[ns]

dtypes: datetime64[ns](1), float64(21), int64(12), object(2)

memory usage: 57.6+ MB

After understanding the dataset values, we take the statistical descriptions of dataset using **Dataframe.describe()** command in python, which tells the following statistical descriptions:



After know about statistical description we move forward in the way of finding co-relation of variables, then move to data-cleaning, then move to visualized the data set and their relationship. After this we move to data pre-processing and modelling as well as testing the accuracy of model.

* **Data Sources and their formats**

Machine learning algorithms are almost always optimized for raw, detailed source data. Thus, the data environment must provision large quantities of raw data for discovery-oriented analytics practices such as data exploration, data mining, statistics, and machine learning.

Tabular data for machine learning is typically found is .csv files. Csv files are text-based files containing comma separated values (csv). Csv files are popular for ML as they are easy to view/debug and easy to read/write from programs (no compression/indexing).

* **Data Pre-processing Done**

Data pre-processing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In simple words, data pre-processing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format.

**Steps in Data Pre-processing in Machine Learning:**

* **Acquire the dataset**

To build and develop Machine Learning models, we must first acquire the relevant dataset. We use download data using given link through the company.

### Import all the crucial libraries

### The core Python libraries used for this data pre-processing in Machine Learning are:

### 

* **Import the dataset**

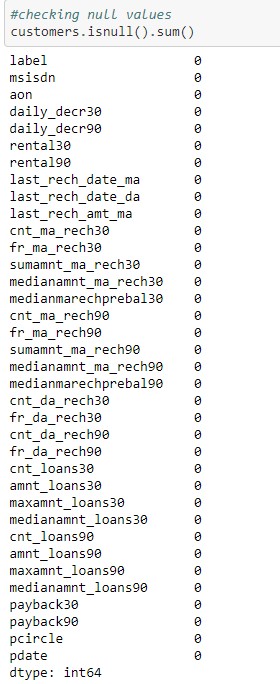
In this step, you need to import the dataset/s that you have gathered for the ML project at hand.



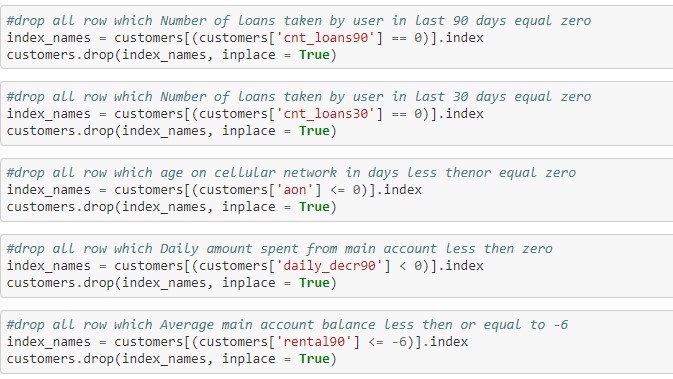
* **Identifying and handling the missing values**

In data pre-processing, it is pivotal to identify and correctly handle the missing values, failing to do this, you might draw inaccurate and faulty conclusions and inferences from the data.

In this dataset, there is no null value, but some faulty data is present.



Handing faulty data in dataset:



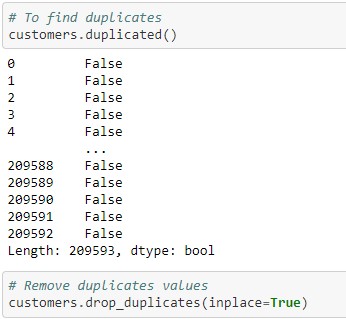
Handing Imbalance data in dataset:



Handing dimensionality reduction :



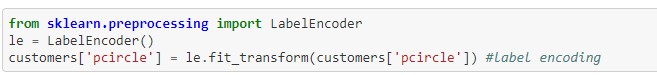
Handing duplicate data in dataset:



* **Encoding the categorical data**

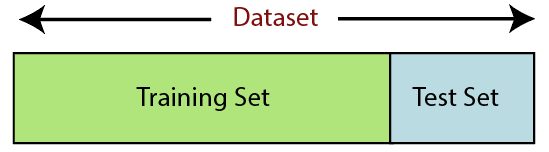
Machine Learning models are primarily based on mathematical equations. Thus, you can intuitively understand that keeping the categorical data in the equation will cause certain issues since you would only need numbers in the equations.

you can use the LabelEncoder() class from the sci-kit learn library. The code will be as follows –

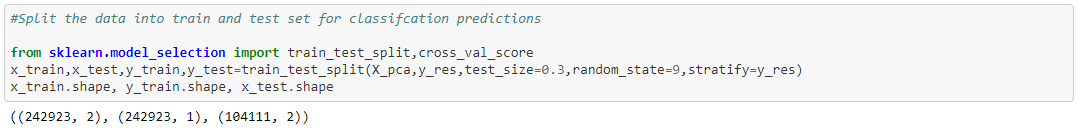


* **Splitting the dataset**

Every dataset for Machine Learning model must be split into two separate sets – training set and test set.

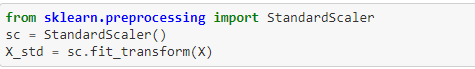


In the same way, we using below code:

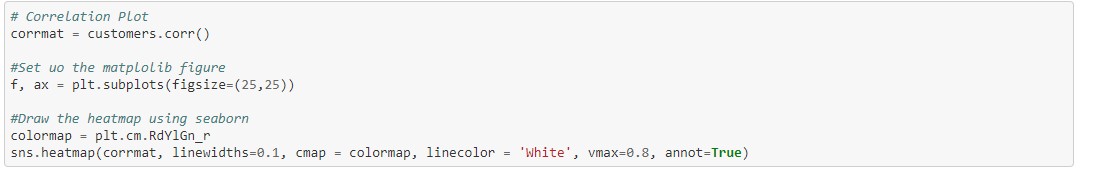


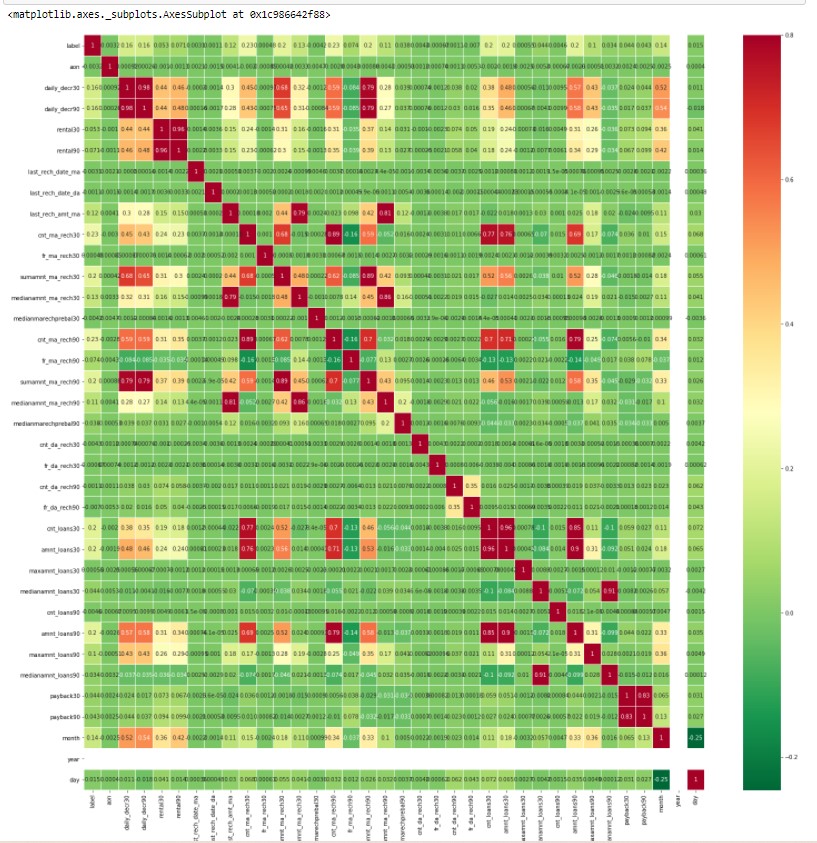
* **Feature scaling**

Feature scaling marks the end of the data pre-processing in Machine Learning. It is a method to standardize the independent variables of a dataset within a specific range. In other words, feature scaling limits the range of variables so that you can compare them on common grounds.



* Data Inputs- Logic- Output Relationships





* **Hardware and Software Requirements and Tools Used**

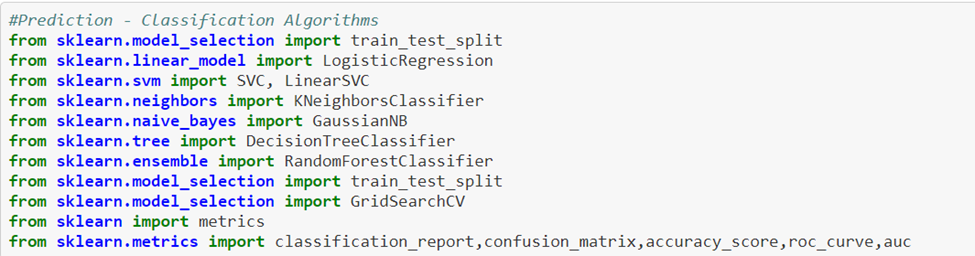
In this project dataset is too large for processing or modelling, that’s why we use good hardware configuration like as above 4GB RAM, above or equal core i3 processor and also need good storage HDD. In way of software, we use any operating system which support python language for coding.

**Model/s Development and Evaluation**

* **Identification of possible problem-solving approaches (methods)**

In this dataset we build a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of issuance of loan. In this case, Label ‘1’ indicates that the loan has been payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed i.e. defaulter. In this case Label has two outcomes ‘1’ and ‘0’, that’s why we use classification model algorithms, because A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. For example, when filtering emails “spam” or “not spam”.

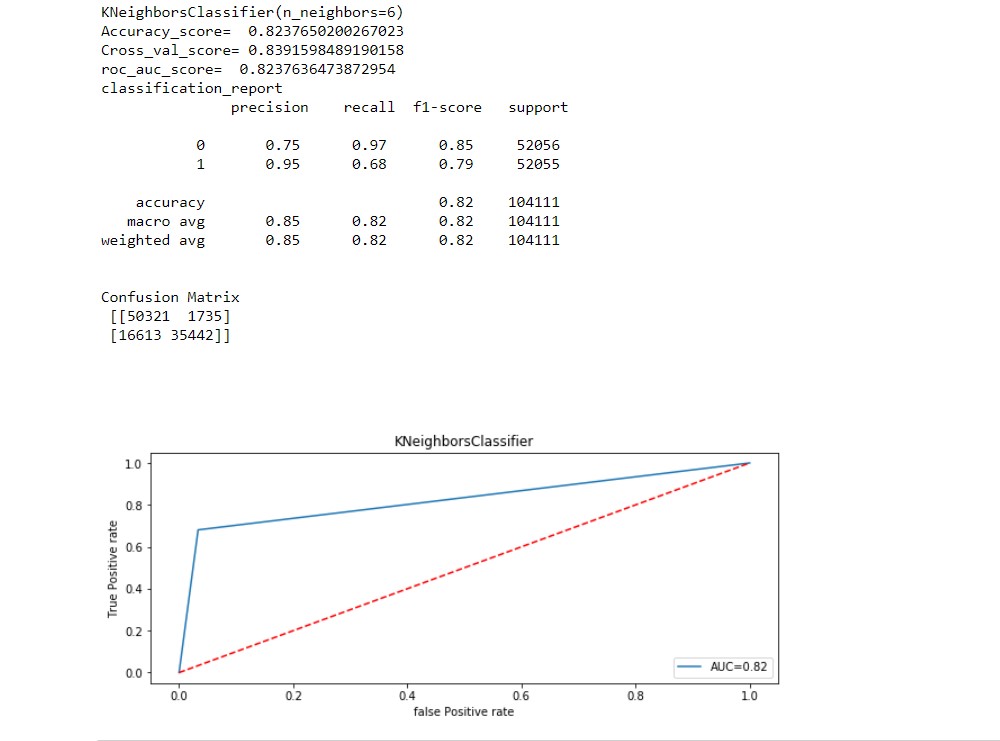
* **Testing of Identified Approaches (Algorithms)**

****

* **Run and Evaluate selected models**

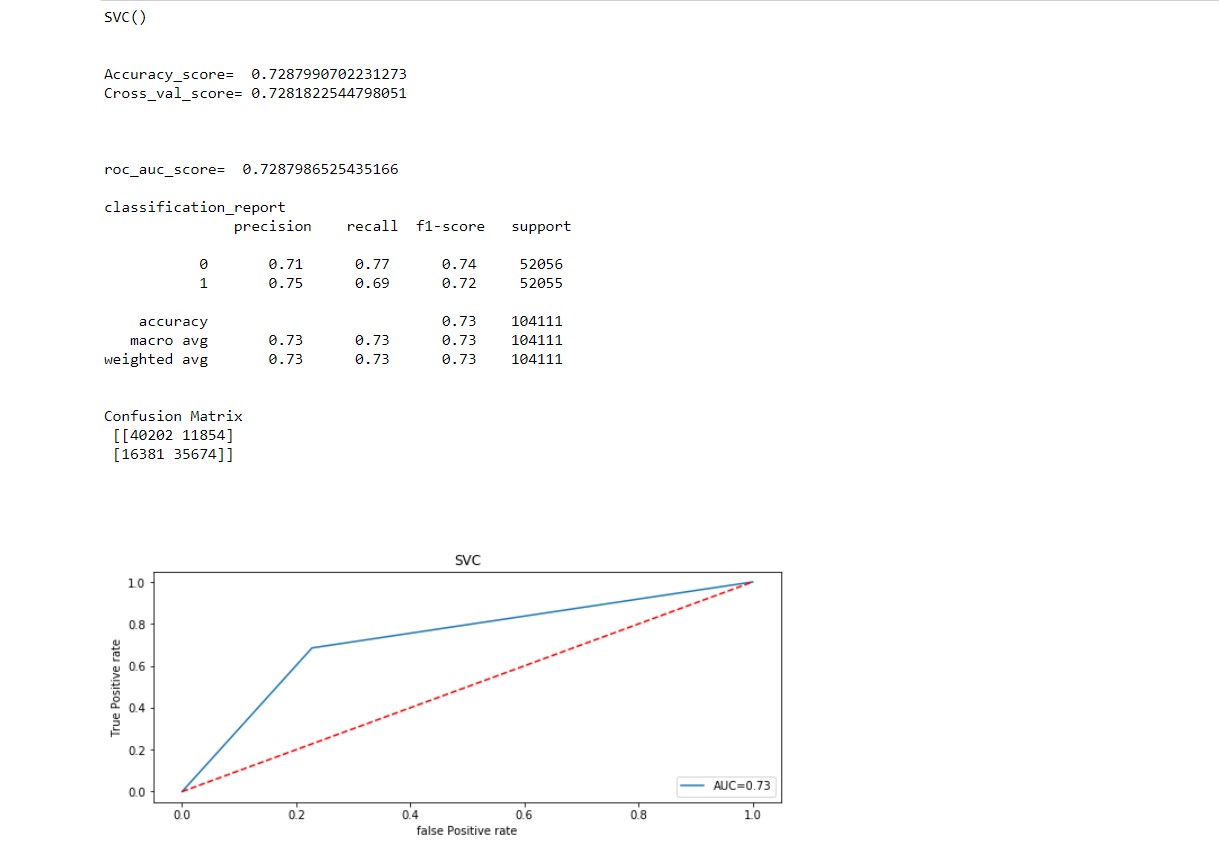
1. **kneighborsclassifier algorithm**





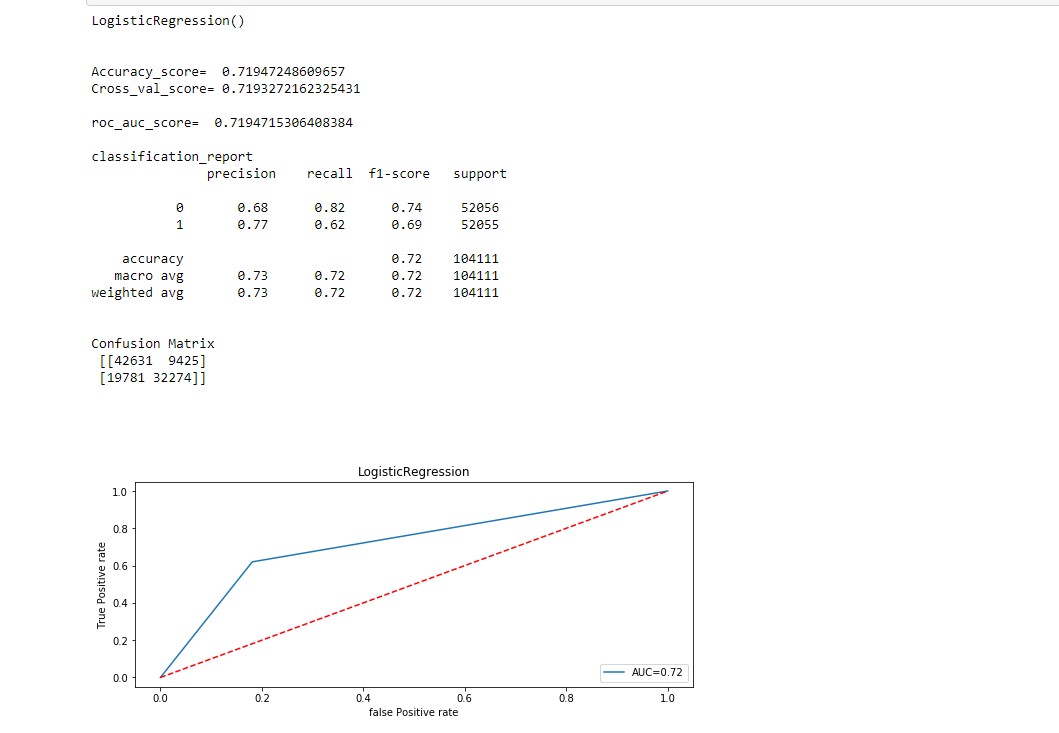
1. **svc algorithm**

****

****

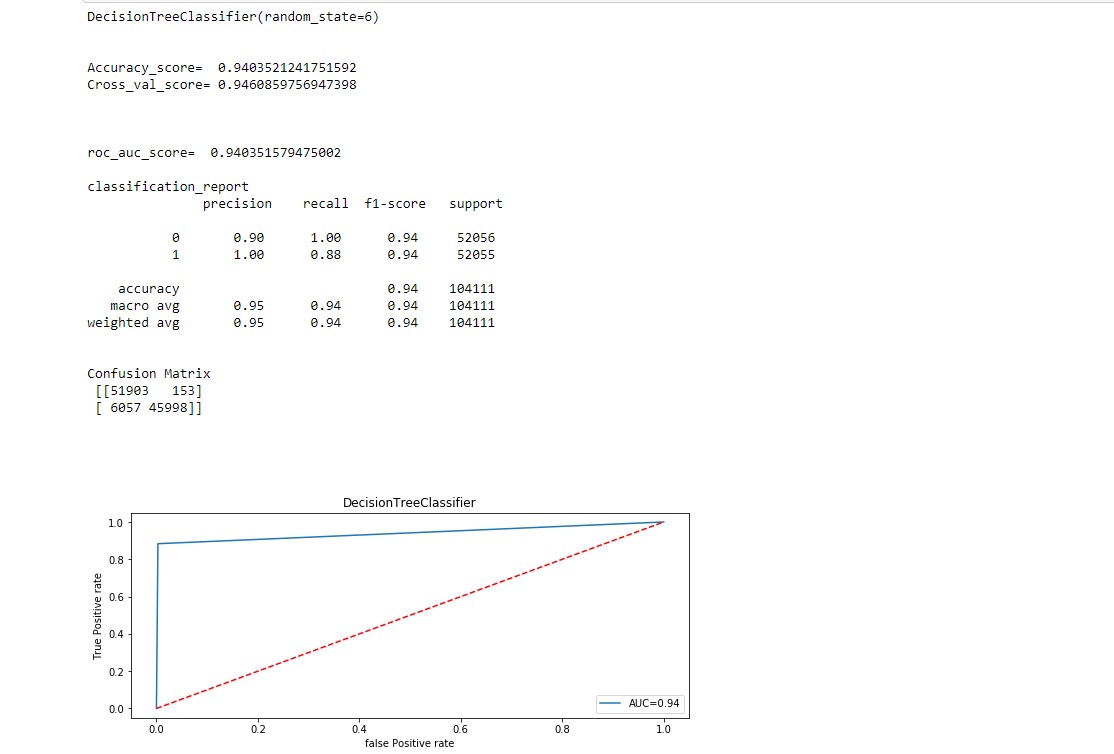
1. **Logistic regression**

****

****

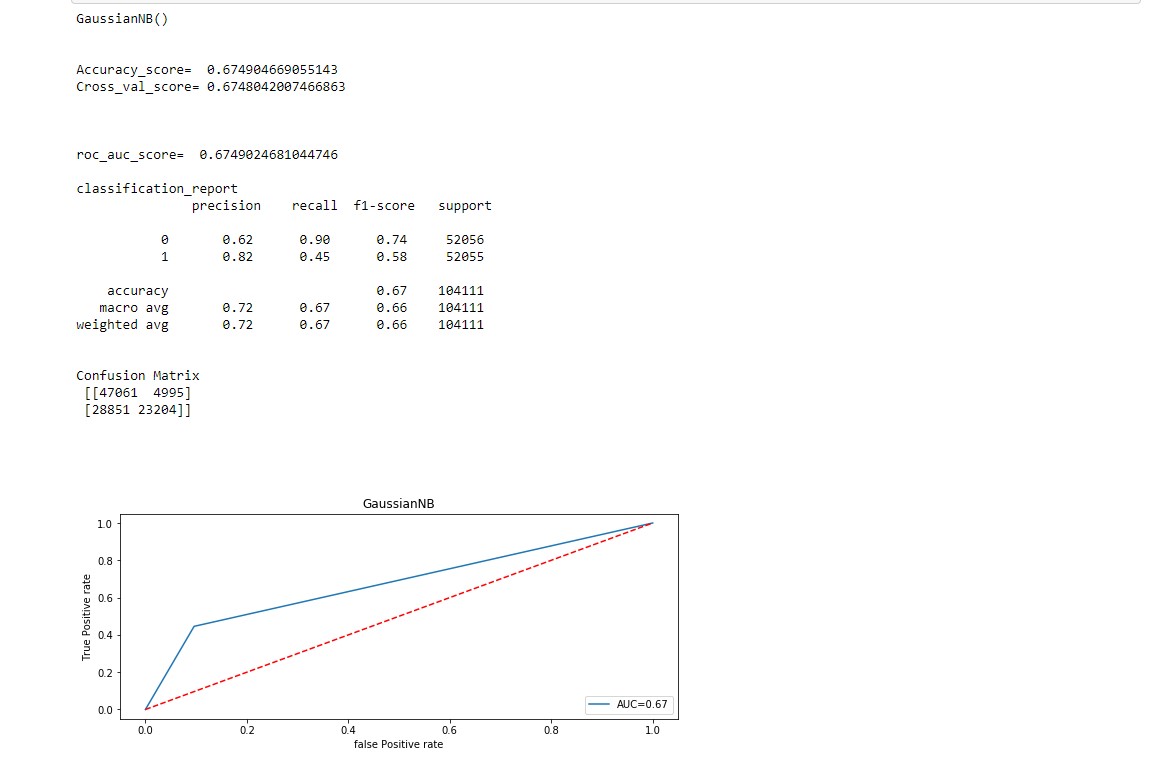
1. **Decision tree classifier**

****

****

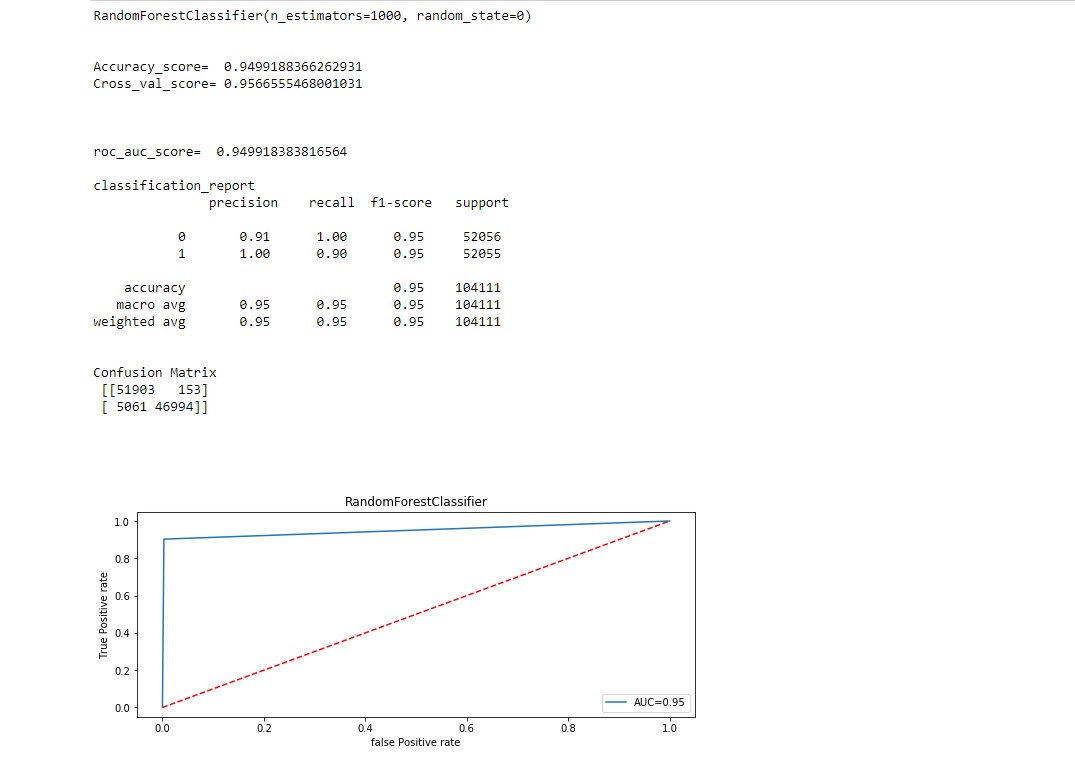
1. **Gaussian Naive Bayes (GaussianNB)**

****

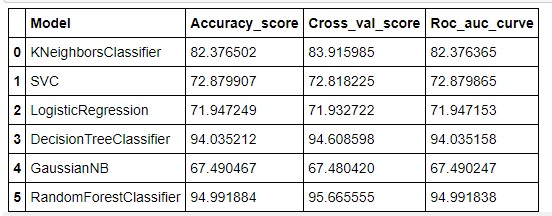
****

1. **RandomForestClassifier**

****

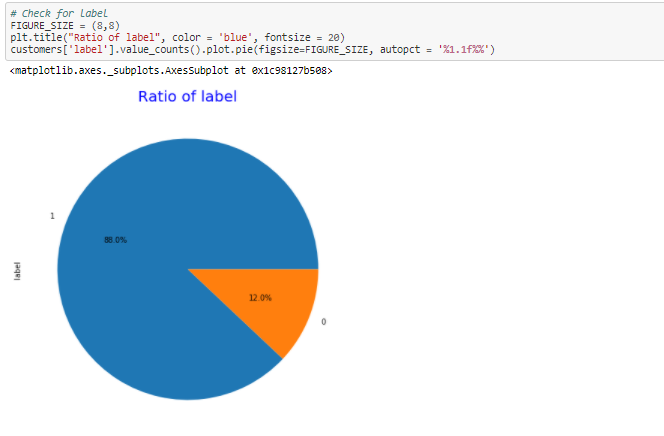
****

* **Final Result of models**

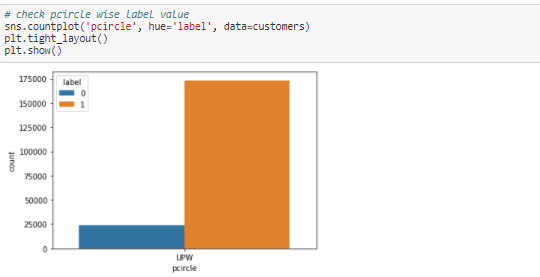
****

* **Visualizations**

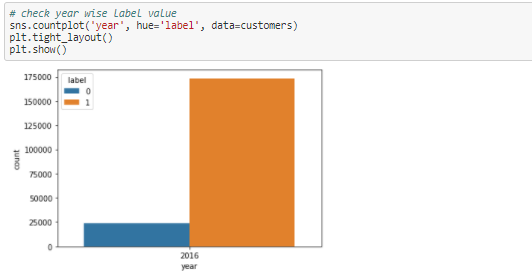
1. **Visualize the Label variable percentage**

****

1. **Visualize the Label variable over pcircle**

****

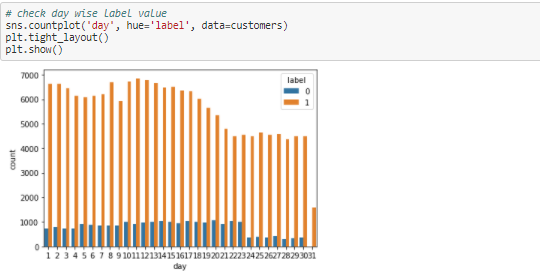
1. **Visualize the Label variable over Year**

****

1. **Visualize the Label variable over Month**

****

1. **Visualize the Label variable over days**

**e) Visualize the Label variable over days**

**CONCLUSION**

After analysing data, visualization and modelling, we come to the conclusion that using the Random Forest Classifier algorithm is suitable for modelling of label's prediction.