**PROJECT SYNOPSIS**

Problem Statement:

**Definition:** Delinquency is a condition that arises when an activity or situation does not occur at its scheduled (or expected) date i.e., it occurs later than expected.

**Use Case:** Many donors, experts, and microfinance institutions (MFI) have become convinced that using mobile financial services (MFS) is more convenient and efficient, and less costly, than the traditional high-touch model for delivering microfinance services. MFS becomes especially useful when targeting the unbanked poor living in remote areas. The implementation of MFS, though, has been uneven with both significant challenges and successes.

Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

One of our Client in Telecom collaborates with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be delinquent if he deviates from the path of paying back the loaned amount within 5 days.

The sample data from our client database is hereby given to you for the exercise.

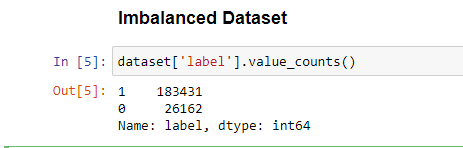
**Exercise:** Create a delinquency model which can predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan   
(Label ‘1’ & ’0’)

Find Enclosed the Data Description File and The Sample Data for the Modeling Exercise.

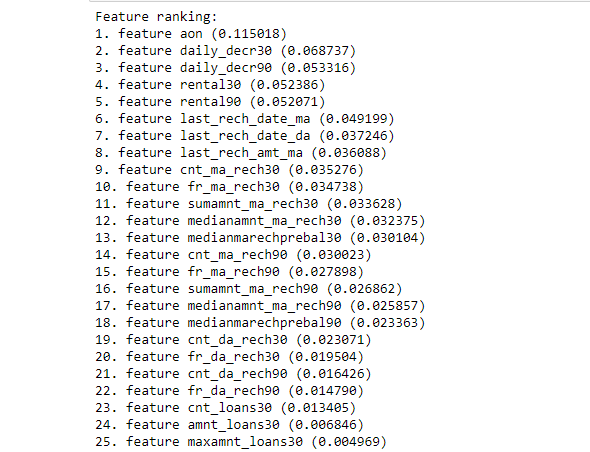
**DATA ANALYSIS AND FEATURE ENGINEERING**

Findings:

1. Number of records: 209593, Number of features: 35
2. Dataset is imbalanced: The dataset has unequal class distribution of target label. Target class zero is outnumbered by class one.

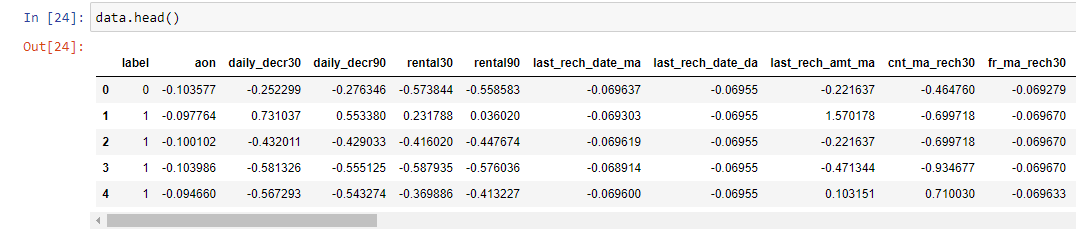


1. Number of features with Null records: None
2. Number of Numerical Features: 35
3. Number of Non Numerical Feature: 1
4. Number of Discrete Features: 4 ('medianamnt\_loans30', 'maxamnt\_loans90', 'medianamnt\_loans90', 'month')
5. Number of Continuous Features: 30
6. Feature Importance:



1. Splitting pdate feature into two different features which are Month and Day.
2. Dropping columns (‘msisdn’,’pcircle’)
3. Feature Scaling: Performing Standard Scaling on all the columns.

Output after feature scaling -



**MODEL TRAINING AND EVALUATION**

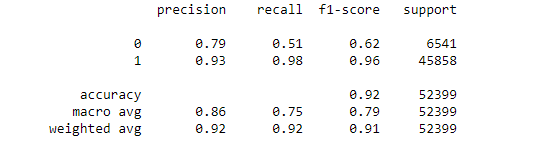
Training Model on following algorithms:

* Random Forest
* Ada Boost
* Gradient Boosting Classifier
* SVM with RBF Kernel
* SVM with Linear Kernel
* Logistic Regression
* XGBoost

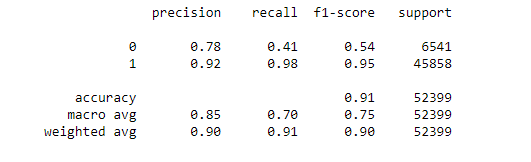
Stratified Train Test Spilt: 80% Training Data and 20% Test Data Stratified on label because the classes are Imbalanced

Evaluation Metric: Class 0 - F1 Score calculated on Test Data.

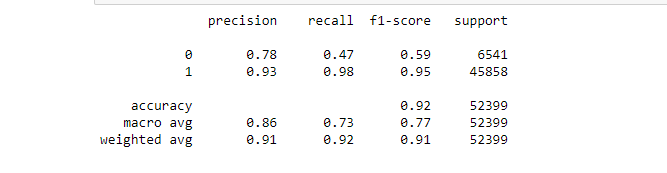
1. Random Forest: Using Random Forest without any parameter tuning we achieved a F1-Score of 62% on class label 0 of Test data.



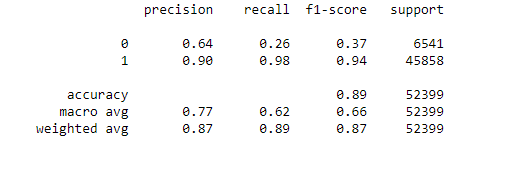
1. Ada Boost Forest: Using Ada Boost Forest without any parameter tuning we achieved a F1-Score of 54% on class label 0 of Test data.



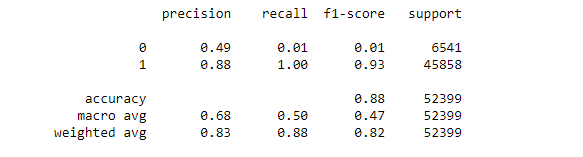
1. Gradient Boosting Classifier: Using Gradient Boosting Classifier without any parameter tuning we achieved a F1-Score of 59% on class label 0 of Test data.



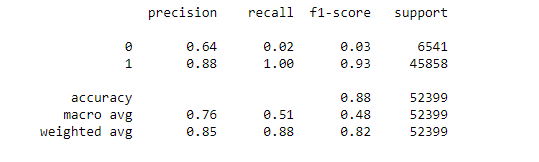
1. SVM Rbf Classifier: Using Gradient Boosting Classifier without any parameter tuning we achieved a F1-Score of 37% on class label 0 of Test data.



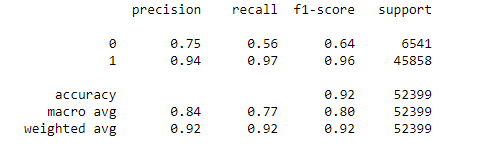
1. SVM Linear Classifier: Using Gradient Boosting Classifier without any parameter tuning we achieved a F1-Score of 0.01% on class label 0 of Test data.



1. Logistic Regression Classifier: Using Gradient Boosting Classifier without any parameter tuning we achieved a F1-Score of 0.03% on class label 0 of Test data.



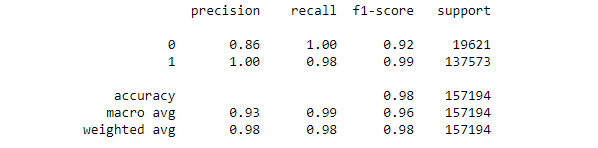
1. XGBoost Classifier: Using Gradient Boosting Classifier with parameter tuning on scale\_pos\_weight we achieved a F1-Score of 64% on class label 0 of Test data.



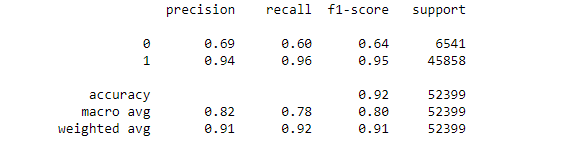
**Using Class Weight in Random Forest Classifier:**

Since we got high F1 Score using Random Forest Classifier we performed Hyper parameter Tuning and got the optimum parameters, now we are going to use class weight to further help the algorithm to distinguish between the imbalanced classes.

On training data:



On Test Data:



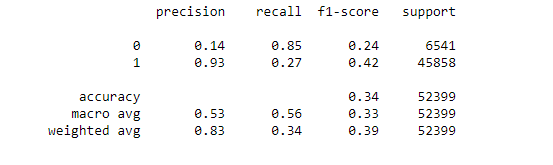
As we can see we increased our F1 Score by 2% on Test data.

Now the Benchmark F1 Score for us is 64% which was achieved using XGBoost and Weighted Random Forest Classifier.

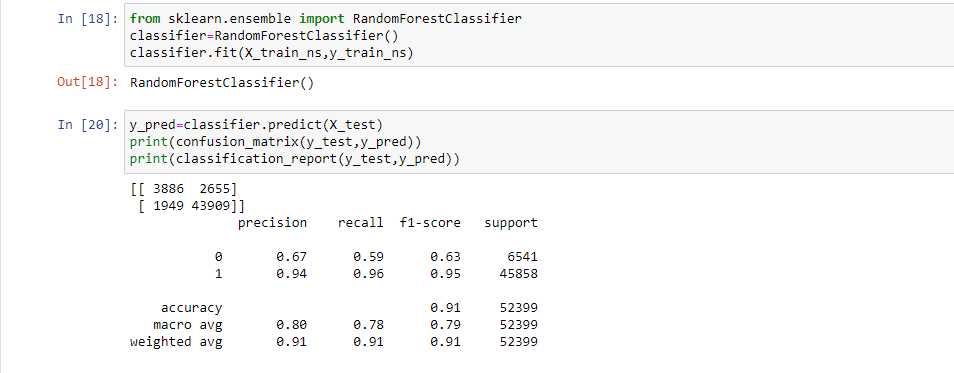
We are now going to use ImbLearn which is an open source library which is used to deal with Imbalanced data. We are going to perform following techniques on the data –

* Under Sampling
* Over Sampling
* SMOTETomek

1. Under Sampling: F1 Score using Under Sampling used on Random Forest Classifier is 24%



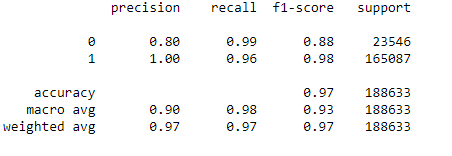
1. SMOTETomek: F1 Score using Under Sampling used on Random Forest Classifier is 63%.



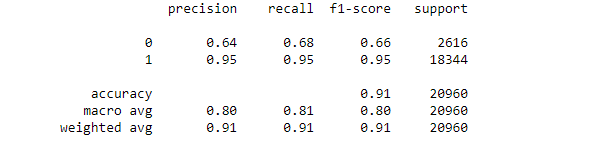
**Over Sampling**

**F1 Score using Under Sampling used on Random Forest Classifier was the highest among all the algorithms performed.**

**Train Data: We achieved F1 Score of 88 % on Train data and Accuracy of 97%**



**Test Data: We achieved F1 Score of 66 % on Train data and Accuracy of 91%**



Preforming StratifiedKFold Cross Validation on Random Forest using Over Sampling Algorithm

Average F1 Score Accuracy: 95%

Average Accuracy : 96%

