**Credit Card Default Prediction**

**Zuber Ahmad**

**Data science trainee,**

**AlmaBetter, Bangalore**

**Abstract:** Financial threats are displaying a trend about the credit risk of commercial banks as the incredible improvement in the financial industry has arisen. In this way, one of the biggest threats faces by commercial banks is the risk prediction of credit clients. Recent studies mostly focus on enhancing the classifier performance for credit card default prediction rather than an interpretable model.

**1.Problem Statement**

This project is aimed at predicting the case of customers default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. We can use the K-S chart to evaluate which customers will default on their credit card payments.

In classification problems, an imbalanced dataset is also crucial to improve the performance of the model because most of the cases lied in one class, and only a few examples are in other categories. Traditional statistical approaches are not suitable to deal with imbalanced data. In this study, a model is developed for credit default prediction by employing various credit-related datasets. There is often a significant difference between the minimum and maximum values in different features, so Min-Max normalization is used to scale the features within one range. Data level resampling techniques are employed to overcome the problem of the data imbalance. Various under sampling and oversampling methods are used to resolve the issue of class imbalance. Different machine learning models are also employed to obtain efficient results. We developed the hypothesis of whether developed models using different machine learning techniques are significantly the same or different and whether resampling techniques significantly improves the performance of the proposed models.

It is also observed that the performance of data oversampling techniques are better than under sampling techniques. Overall, the Gradient Boosted Decision Tree method performs better than other traditional machine learning classifiers. The Gradient Boosted Decision Tree method gives the best results while utilizing the K-means SMOTE oversampling method. This model will help commercial banks, financial organizations, loan institutes, and other decision-makers to predict the loan defaulter earlier.

**2. Introduction**

## According to the Federal Reserve economic data, the default rate on credit loans across all commercial banks is at an all-time high for the past 66 months, and it is likely to continue to climb throughout 2020. The delinquency rate indicates the percentage of past-due loans within the borrower’s entire loan portfolio. The climbing delinquencies will result in a signiﬁcant amount of money lose from the lending institutions, such as commercial banks. Therefore, banks must have a risk prediction model and be able to classify the most relative characteristics that are indicative of people who have a higher probability of default on credit. Because of the risks inherent in such a large portion of the economy, building models for consumer spending behaviors to limit risk expo-sures in this sector is becoming more critical. For this to be a viable option, the predictions need to be reasonably accurate.

**3. Contributions**

This research possesses various contributions in the domain of credit risk prediction.

1) First, multiple latest datasets have been used to build a machine learning model for credit risk prediction.

2) Second, the data imbalance problem has been explored by comparing the different resampling techniques and evaluate the performance that which the resampling technique has given effective results with a machine learning classiﬁer.

3) Limited work was done on resampling techniques for data balancing in this domain because only a few resampling techniques were employed and also obtained less efﬁcient results.

4) Lastly, the interpretable model is also deployed on the web to ease the different stakeholders. This model will help commercial banks, ﬁnancial organizations, loan institutes, and other decision-makers to predict the credit defaulter earlier.

**4. Data Description**

**Attribute Information:** This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:

* X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
* X2: Gender (1 = male; 2 = female).
* X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
* X4: Marital status (1 = married; 2 = single; 3 = others).
* X5: Age (year).
* X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
* X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; X17 = amount of bill statement in April, 2005.
* X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; X23 = amount paid in April, 2005.

**Breakdown of Our Features:**

We have records of 30000 customers. Below are the description of all features we have.

* ID: ID of each client
* LIMIT\_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit)
* SEX: Gender (1 = male, 2 = female)
* EDUCATION: (1 = graduate school, 2 = university, 3 = high school, 0,4,5,6 = others)
* MARRIAGE: Marital status (0 = others, 1 = married, 2 = single, 3 = others)
* AGE: Age in years

Scale for PAY\_0 to PAY\_6 :

(-2 = No consumption, -1 = paid in full, 0 = use of revolving credit (paid minimum only), 1 = payment delay for one month, 2 = payment delay for two months, ... 8 = payment delay for eight months, 9 = payment delay for nine months and above)

* PAY\_0: Repayment status in September, 2005 (scale same as above)
* PAY\_2: Repayment status in August, 2005 (scale same as above)
* PAY\_3: Repayment status in July, 2005 (scale same as above)
* PAY\_4: Repayment status in June, 2005 (scale same as above)
* PAY\_5: Repayment status in May, 2005 (scale same as above)
* PAY\_6: Repayment status in April, 2005 (scale same as above)
* BILL\_AMT1: Amount of bill statement in September, 2005 (NT dollar)
* BILL\_AMT2: Amount of bill statement in August, 2005 (NT dollar)
* BILL\_AMT3: Amount of bill statement in July, 2005 (NT dollar)
* BILL\_AMT4: Amount of bill statement in June, 2005 (NT dollar)
* BILL\_AMT5: Amount of bill statement in May, 2005 (NT dollar)
* BILL\_AMT6: Amount of bill statement in April, 2005 (NT dollar)
* PAY\_AMT1: Amount of previous payment in September, 2005 (NT dollar)
* PAY\_AMT2: Amount of previous payment in August, 2005 (NT dollar)
* PAY\_AMT3: Amount of previous payment in July, 2005 (NT dollar)
* PAY\_AMT4: Amount of previous payment in June, 2005 (NT dollar)
* PAY\_AMT5: Amount of previous payment in May, 2005 (NT dollar)
* PAY\_AMT6: Amount of previous payment in April, 2005 (NT dollar)
* Default payment next month: Default payment (1=yes, 0=no)

**NOTE:-**What do 0 and -2 mean in PAY\_X columns?

From dataset documentation:

PAY\_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above). Notice code 0 and 2 are in the PAY\_X columns but are not included in the data description. Also by doing the value\_counts, it shows 0 is the most frequent observation.

-2: No consumption;

-1: Paid in full;

0: The use of revolving credit, meaning the payment wasn't due, which makes sense that most customers were using the revolving credit.

## **5.** **Business Objective:**

## Objective of our project is to predict which customer might default in upcoming months. Before going any further let's have a quick look on definition of what actually meant by Credit Card Default.

## We are all aware what is credit card. It is type of payment card in which charges are made against a line of credit instead of the account holder's cash deposits. When someone uses a credit card to make a purchase, that person's account accrues a balance that must be paid off each month.

## Credit card default happens when you have become severely delinquent on your credit card payments. Missing credit card payments once or twice does not count as a default. A payment default occurs when you fail to pay the Minimum Amount Due on the credit card for a few consecutive months.

## So now we know what a credit card is. Now let's see one of problems faced by companies who provide credit cards. Yes, it is the people who do not clear off the credit card debt aka credit card defaulters.

* The research aims at developing a mechanism to predict the credit card default beforehand and to identify the potential customer base that can be offered various credit instruments so as to invite minimum default.

## **6. Benefits of Credit Card Default**

## A robust model is not only a useful tool for the lending institutions to decide on credit applications, but it can also help the clients to be aware of the behaviors that may damage their credit scores.

## The primary motivation behind risk prediction is to utilize ﬁnancial data, for example, business transactional data, exchange records and client transactions, and so forth to foresee the client’s business performance or individual credit card data and to decrease loos and vulnerability. Several risk prediction models are based on statistical methods, including nearest neighbor, discriminant analysis, and logistic regression. Current credit bureau analytics, such as credit scores, are based on slowly varying consumer characteristics. They are not adaptable to changes in client’s behaviors and market conditions over time. Besides, the behavior of the market has not been consistent over the years to the features to predict the default are always debatable.

# **7. Risks in Credit Card Default**

Default risk is the chance that companies or individuals will be unable to make the required payments on their debt obligations. In other words, credit default risk is the probability that if you lend money, there is a chance that they won't be able to give the money back on time.

**8. Steps involved:**

**Preprocessing the Datasheet**

The real-world data often has a lot of missing values. The cause of missing values can be data corruption or failure to record data. The handling of missing data is very important during the preprocessing of the dataset as many machine learning algorithms do not support missing values. That’s why we check missing values first.

After checking missing values, we have to check the duplicate values. "Duplication" just means that you have repeated data in your dataset. This could be due to things like data entry errors or data collection methods. by removing duplication in our data set, Time and money are saved by not sending identical communications multiple times to the same person.

* **Missing and duplicate values Treatment**

Our dataset contains no missing values which saves our time.

Many machine learning algorithms fail if the dataset contains missing values. However, algorithms like K-nearest and Naive Bayes support data with missing values. You may end up building a biased machine learning model which will lead to incorrect results if the missing values are not handled properly.

* **Duplicate Values**

Why is it important to remove duplicate records from my data?

"Duplication" just means that you have repeated data in your dataset. This could be due to things like data entry errors or data collection methods. by removing duplication in our data set, Time and money are saved by not sending identical communications multiple times to the same person.

In our data after count the missing and duplicate value we came to know that there are no missing and duplicate value present.

* **Exploratory Data Analysis**

After loading the dataset, we performed this method by comparing our target variables that is data variables. This process helped us understanding more about the data and figuring out various aspects and relationships among the dependent variables and bike count. It gave us a better idea of which feature behaves in which manner compared to the target variable. We can do univariate analysis; it is to simply describe the data to find the patterns within the data.

Through different patterns we can analyses both data type categorical variable and numerical variables

Prediction of some EDA analysis mention below:

**Univariate Analysis:**

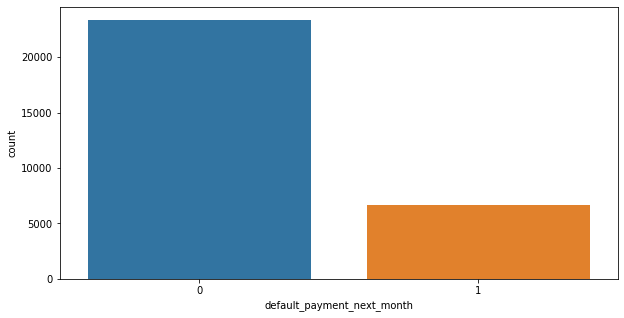
Univariate analysis is basically the simplest form to analyze data. Uni means one and this means that the data has only one kind of variable. The major reason for univariate analysis is to use the data to describe. The analysis will take data, summarize it, and then find some pattern in the data.

Univariate analysis is the technique of comparing and analyzing the dependency of a single predictor and a response variable. The key objective of Univariate analysis is to simply describe the data to find patterns within the data.

**Analysis of Dependent Variable:**

We analyze our dependent variable, A dependent variable is a variable whose value will change depending on the value of another variable.

**default.payment.next.month:** According to the reply from the dataset contributor,it seems this is the actual data collected in the next month. "Y: client's behavior; Y=0 then not default, Y=1 then default". This information could be used to test the accuracy of the model that will be built.



From the above data analysis we can say that

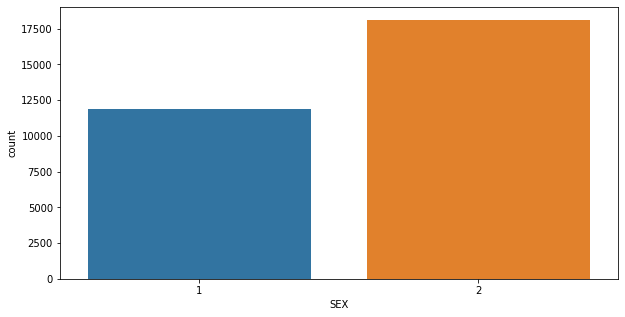
* 0 - Not Default
* 1 - Default
* Defaulters are less than the Non Defaulters in the given dataset.

**Analysis of Independent Variable:**

Categorical variables are qualitative data in which the values are assigned to a set of distinct groups or categories. These groups may consist of alphabetic (e.g., male, female) or numeric labels (e.g., male = 0, female = 1) that do not contain mathematical information beyond the frequency counts related to group membership.

In our Dataset we have few categorical features that are:

* **Sex**



From the above data analysis we can say that

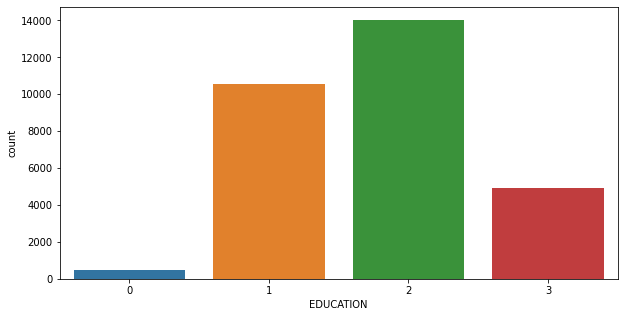
1 - Male

2 - Female

Number of Male credit holder is less than Female.

* **Education**

1 = graduate school; 2 = university; 3 = high school; 0 = others

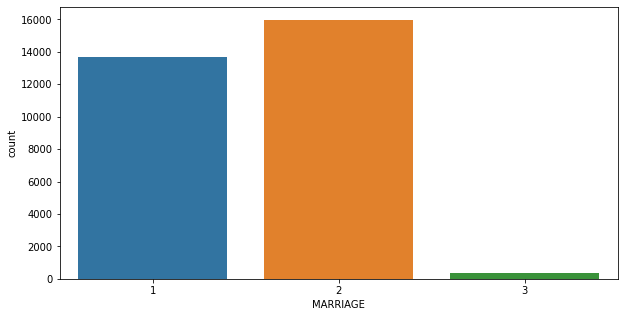


From the above data analysis, we can say that

More number of credit holders are university students followed by Graduates and then High school students.

* **Marriage**

1 = married; 2 = single; 3 = others



From the above data analysis we can say that

1 - married

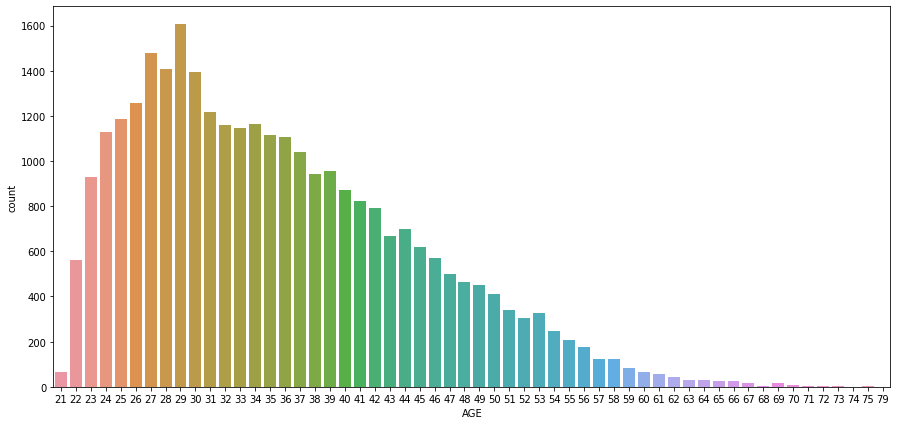
2 - single

3 - others

More number of credit cards holder are Single.

* **Age**

Plotting graph of number of ages of all people with credit card irrespective of gender.



From the above data analysis we can say that

We can see more number of credit cards holder age are between 26-30 years old.

Age above 60 years old rarely uses the credit card.

**Numerical features:**

Numerical data is a data type expressed in numbers, rather than natural language description. Sometimes called quantitative data, numerical data is always collected in number form. Numerical data differentiates itself from other number form data types with its ability to carry out arithmetic operations with these numbers.

Some analysis of numerical data are given:

* **Limit Balance:** In our dataset the maximum amount of given credit in NT dollars is 50,000 followed by 30,000 and 20,000.
* **Total Bill Amount:** We use sns pair plot for EDA. SNS pair plot is a plot pairwise relationships in a dataset. By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column.

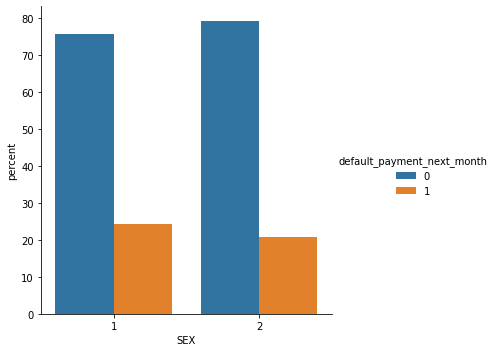
We plot the month wise total bill amount.

* **Previous Payment Status:** By plotting the previous payment, we analyze the payment status of each month.
* **Paid Amount:** Here we simply explore the month wise paid amount.

**Bivariate Analysis:**

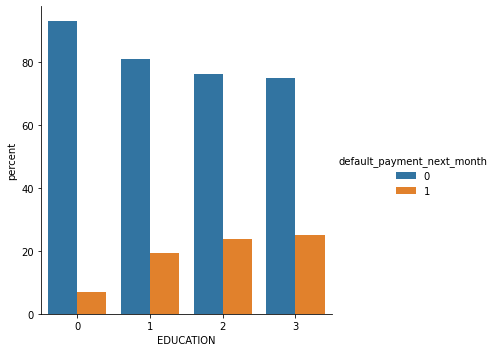
Bivariate analysis is one of the simplest forms of quantitative analysis. It involves the analysis of two variables, for the purpose of determining the empirical relationship between them. Bivariate analysis can be helpful in testing simple hypotheses of association.

**Sex and default\_payment\_next\_month:**

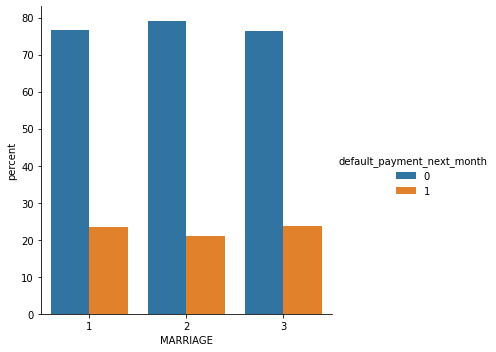


It is evident from the above graph that the number of defaulters have high proportion of males.

**Education and default\_payment\_next\_month:**

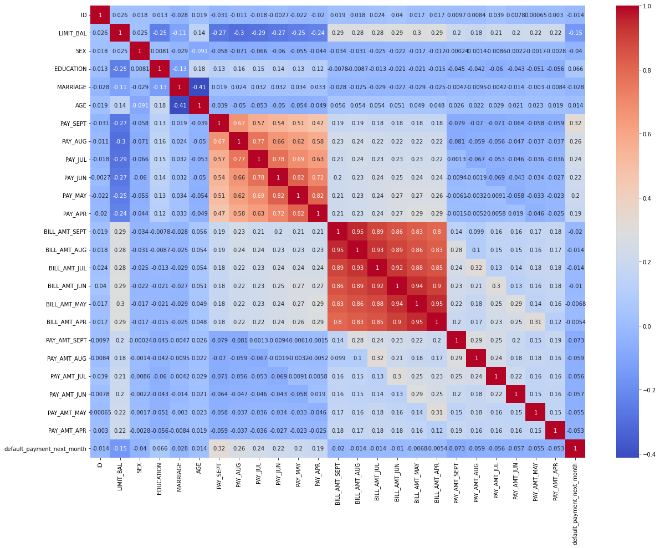
From the above plot it is clear that those people who are other students have higher default payment wrt graduates and university people

**Marriage and default\_payment\_next\_month:**

High defaulter rate when it comes to others.

**Checking of Correlation between variables:**

We check correlation between variables using Correlation heatmap, it is graphical representation of correlation matrix representing correlation between different variables.



It seems from the above graph is there are some negatively correlated feature like age but we cannot blindly remove this feature because it could be important feature for prediction.

ID is unimportant and it has no role in prediction so we will remove it.

**SMOTE ANALYSIS:**

SMOTE (Synthetic Minority Oversampling Technique) – Oversampling is one of the most commonly used oversampling methods to solve the imbalance problem. It aims to balance class distribution by randomly increasing minority class examples by replicating them.

In our data set we have Imbalanced Data Distribution in our dependent variable, it generally happens when observations in one of the class are much higher i.e not defaulter or lower than the other classes i.e defaulter.

As Machine Learning algorithms tend to increase accuracy by reducing the error, they do not consider the class distribution.

Standard ML techniques such as Decision Tree and Logistic Regression have a bias towards the majority class, and they tend to ignore the minority class. They tend only to predict the majority class, hence, having major misclassification of the minority class in comparison with the majority class. In more technical words, if we have imbalanced data distribution in our dataset then our model becomes more prone to the case when the minority class has a negligible or very lesser recall.

* **Fitting different models**

For modelling we tried various classification algorithms like:

* **Logistic Regression**
* **Feature Engineering**
* **One Hot Encoding**
* **Random Forest.**
* **Naïve Bayes Classifier**
* **Implementing SVM**
* **Implementing XG Boost**
* **Hyperparameter Tuning**

**8.1. Model Training:**

**Train Test split for classification:**

Before, fitting any model it is a rule of thumb to split the dataset into a training and test set. This means some proportions of the data will go into training the model and some portion will be used to evaluate how our model is performing on any unseen data. The proportions may vary from 60:40, 70:30, 75:25 depending on the person but mostly used is 80:20 for training and testing respectively. In this step we will split our data into training and testing set using scikit learn library.

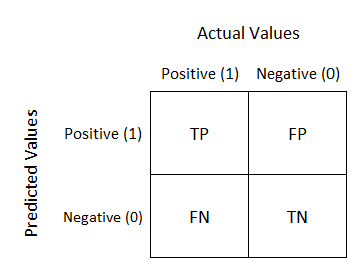
**Evaluation Estimator & Metrics:**

* **GridSearchCV:** GridSearchCV is a library function that is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit our estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.

GridSearchcv classification is an important step in classification machine learning projects for model select and hyper Parameter Optimization. It takes a dictionary that describes the parameters that could be tried on a model to train it. The grid of parameters is defined as a dictionary, where the keys are the parameters and the values are the settings to be tested.

One of the great things about GridSearchCV is that it is a meta-estimator. It takes an estimator like SVC and creates a new estimator, that behaves exactly the same – in this case, like a classifier. You should add refit=True and choose verbose to whatever number you want, the higher the number, the more verbose (verbose just means the text output describing the process).

* **Confusion Matrix:** Confusion Matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.

****

**True Positive:** This combination tells us how many times a model correctly classifies a positive sample as Positive?

**False Negative:** This combination tells us how many times a model incorrectly classifies a positive sample as Negative?

**False Positive:** This combination tells us how many times a model incorrectly classifies a negative sample as Positive?

**True Negative:** This combination tells us how many times a model correctly classifies a negative sample as Negative?

* **AUC-ROC curve:** The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

#### **Precision:** Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples (either correctly or incorrectly). Report precision in percentages.

Precision = True Positive/True Positive + False Positive

Precision = TP/(TP+FP)

#### **Recall:** The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. Report recalls in percentages.

Recall = True Positive/True Positive + False Negative

Recall = TP/TP+FN

#### **F1-Score:** F1 score is defined as the harmonic mean between precision and recall. It is used as a statistical measure to rate performance. This means a statistical measure of the accuracy of a test or an individual.

* **Feature Engineering**

Feature engineering is the process that takes raw data and transforms it into features that can be used to create a predictive model using machine learning or statistical modeling, such as deep learning.

The intention of feature engineering is to achieve two primary goals:

* Preparing an input dataset that is compatible with and best fits the machine learning algorithm.
* Improving the performance of machine learning models

When feature engineering processes are executed well, the resulting dataset will be optimal and contain all the essential factors that bear an impact on the business problem. These datasets in turn result in best possible predictive models and most beneficial insights.

* **One Hot Encoding**

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. One hot encoding algorithm is an encoding system of Sci-kit learn library. One Hot Encoding is used to convert numerical categorical variables into binary vectors. Before implementing this algorithm. Make sure the categorical values must be label encoded as one hot encoding takes only numerical categorical values.

One Hot Encoding is a common way of preprocessing categorical features for machine learning models. This type of encoding creates a new binary feature for each possible category and assigns a value of 1 to the feature of each sample that corresponds to its original category.

In our model we perform one hot encoding on 'EDUCATION', 'MARRIAGE','PAY\_SEPT','PAY\_AUG','PAY\_JUL','PAY\_JUN','PAY\_MAY', 'PAY\_APR'.

**8.2. Algorithms:**

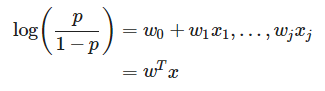
* **Logistic Regression:**

Logistic regression is a classification algorithm that predicts the probability of an outcome that can only have two values (i.e. a dichotomy). Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

A logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression models the probability that each input belongs to a particular category.

Logistic regression is an excellent tool to know for classification problems, which are problems where the output value that we wish to predict only takes on only a small number of discrete values. Here we'll focus on the binary classification problem, where the output can take on only two distinct classes.

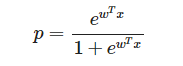
In Logistic Regression, the log-odds of a categorical response being "true" (1) is modeled as a linear combination of the features:



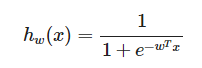
where:

* w0 is the intercept term, and w1 to wj represents the parameters for all the other features (a total of j features).
* By convention of we can assume that x0=1, so that we can re-write the whole thing using the matrix notation wTx.

This is called the logit function. The equation can be re-arranged into the logistic function:



Or in the more commonly seen form:



* **Random Forest Classifier**

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of all.



* **Naïve Bayes Classifier**

Naive Bayes models are a group of extremely fast and simple classification algorithms that are often suitable for very high-dimensional datasets. Because they are so fast and have so few tunable parameters, they end up being very useful as a quick-and-dirty baseline for a classification problem.

Naive Bayes classifiers are built on Bayesian classification methods. These rely on Bayes's theorem, which is an equation describing the relationship of conditional probabilities of statistical quantities. In Bayesian classification, we're interested in finding the probability of a label given some observed features, which we can write as P(L | features). Bayes's theorem tells us how to express this in terms of quantities we can compute more directly:



If we are trying to decide between two labels—let's call them L1 and L2 —then one way to make this decision is to compute the ratio of the posterior probabilities for each label:



The posterior probability can be written as :



Assuming all the X are conditionally independent



This is where the "naive" in "naive Bayes" comes in: if we make very naive assumptions about the generative model for each label, we can find a rough approximation of the generative model for each class, and then proceed with the Bayesian classification. Different types of naive Bayes classifiers rest on different naive assumptions about the data.

* **Support Vector Machine**

Support Vector Machines (SVMs in short) are machine learning algorithms that are used for classification and regression purposes. SVMs are one of the powerful machine learning algorithms for classification, regression and outlier detection purposes. An SVM classifier builds a model that assigns new data points to one of the given categories. Thus, it can be viewed as a non-probabilistic binary linear classifier. SVMs can be used for linear classification purposes known as Support Vector Classifiers (SVCs). In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick. It enables us to implicitly map the inputs into high dimensional feature spaces. These are generally **discriminative models.** Discriminative models, also called conditional models, tend to learn the boundary between classes/labels in a dataset. Unlike generative models, the goal here is to find the decision boundary separating one class from another.

In support vector machines, the line that maximizes the margin is the one we will choose as the optimal model. Support vector machines are an example of such a maximum margin estimator.

* **XG Boost:** XGBoost provides a wrapper class to allow models to be treated like classifiers or regressors in the scikit-learn framework. The XGBoost model for classification is called XGB Classifier. We can create and and fit it to our training dataset. Models are fit using the scikit-learn API and the model.fit() function.

Unlike many other algorithms, XGBoost is an ensemble learning algorithm meaning that it combines the results of many models, called base learners to make a prediction.

Just like in Random Forests, XGBoost uses Decision Trees as base learners.

* **Hyperparameter Tuning**

Hyperparameter tuning is the process of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a model argument whose value is set before the learning process begins. The key to machine learning algorithms is hyperparameter tuning.

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

Hyperparameters are crucial as they control the overall behavior of a machine learning model. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

We can leverage the maximum power of XGBoost by tuning its hyperparameters.

**8. Conclusion:**

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA , outliers detection and treatment, normalizing columns, one hot encoding for categorical data, checking of correlation among variabl0065s. Next we implemented machine learning algorithms Logistic Regression, Feature Engineering, Random Forest Classifer, Naïve Bayes classifier, Support Vector Machine, XG Boost and hyperparameter tuning to improve our model performance.

We conclude that:

XG Boost model has the highest recall, if the business cares recall the most, then this model is the best candidate. If the balance of recall and precision is the most important metric, then Random Forest is the ideal model. Since Random Forest has slightly lower recall but much higher precision than Logistic Regression, I would recommend Random Forest.

* Data categorical variables had minority classes which were added to their closest majority class
* There were not huge gap but female clients tended to default the most.
* Labels of the data were imbalanced and had a significant difference.
* Gradient boost gave the highest accuracy of 82% on test dataset.
* Repayment in the month of september tended to be the most important feature for our machine learning model.
* The best **accuracy** is obtained for the **Random forest** and **XGBoost classifier.**
* In general, all models have comparable accuracy. Nevertheless, because the classes are imbalanced (the proportion of non-default credit cards is higher than default) this metric is misleading.
* Furthermore, **accuracy** does not consider the rate of **false positives** (non-default credits cards that were predicted as default) and **false negatives** (default credit cards that were incorrectly predicted as non-default).
* Both cases have negative impact on the bank, since **false positives** leads to unsatisfied customers and **false negatives** leads to financial loss.
* We can see that **XG Boost Classifier** having **Recall**, **F1-score**, and **ROC Score** values equal 85%, 82%, and 91% and **Random forest Classifier** having **Recall**, **F1-score**, and **ROC Score** values equals 86%, 83%, and 91%.
* **XG Boost Classifier** and **Decision Tree Classifier** are giving us the best **Recall, F1-score**, and **ROC Score** among other algorithms. We can conclude that these two algorithms are the best to predict whether the credit card is default or not default according to our analysis.

**References-**

1. Stacksoverflow.com
2. GeeksforGeeks
3. towardsdatascience.com/
4. Kaggle.com
5. analyticsvidhya.com
6. scikit-learn.org