**Credit Card Default Risk Analysis**

**Divyaranjan Das**

**Data science trainees,**

**Alma Better, Bangalore**

**Abstract:**

Lending is one of the key business areas in the banking industry, credit cards as of late have seen huge success over the course of years. In pursuit to increase their market share, banks often issue credit cards to ineligible customers without adequate background checks. Also, many customers used their credit card beyond their repayment capabilities leading to high debt accumulation. Identifying the risky and non-risky customers is the biggest challenge for banks. So, the problem we are trying to analyze is how to identify the risky and non-risky customers, helping the bank to decide if a customer has the potential to repay the used credit of the bank

***Keywords: Credit card,*** ***Debt accumulation***

**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](https://www.listendata.com/2019/07/KS-Statistics-Python.html) to evaluate which customers will default on their credit card payments.

### **Attribute Information**

Below there are the description of the attributes that will be used in our model for better understanding of the data:

* 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, 4=others, 5=unknown, 6=unknown)
* MARRIAGE: Marital status (1=married, 2=single, 3=others)
* AGE: Age in years
* 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)
* 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)

**2. Introduction**

The report is organized to present the problems faced by the banks when a credit card is being issued, considering a simulated data, from a reliable data source we explain the characteristics of the data. The report follows to explain the various methods to adopt to deal with anomalies such as extreme values and outliers, removal of which increases the quality of the data. The data processing is followed by refining the data to improve the performance of the various models available, this is done by applying methods such as aggregation and discretization, which aims at the removal of irrelevant attributes which do not contribute towards the class values and discretizing the continuous values into discreet intervals known as bins. We then did the various test methods such as Training set, n fold Cross validation, SMOTE technique and Percentage split for every classification algorithms used to test the data. Then we did classification algorithms such as Logistic Regression, Random Forest and KNN. By the performance report we decide the best algorithm for the data set we have chosen. Then we move on to the logic of problems where we discuss about the inferences that we find from different ML models and summarizes the project.

**4. Steps involved:**

* **Exploratory Data Analysis**

After loading the dataset we performed this method by comparing our target variable that is “default.payment.next.month” with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

* **Null values Treatment**

Thankfully our dataset contains no missing values and duplicate values. so there was no need to remove or impute them.

* **Encoding of categorical columns**

We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

* **Feature Selection**

In these steps we used algorithms like ExtraTree classifier to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

Next we used ANOVA to select the best feature which we will be using further in our model.

* **Treating class imbalance (SMOTE)**

To compensate for the rare classes in the imbalance dataset, we use SMOTE(Synthetic Minority Over-Sampling Technique) method to over sample the minority class and ensure the sampling is not biased. What this technique does under the hood is simply duplicating examples from the minority class in the training dataset prior to fitting a mode. After SMOTE sampling, the dataset has almost equal size of 0s and 1s.

* **Fitting different models**

For modelling we tried various classification algorithms like:

1. **Logistic Regression**
2. **Decision Tree Classifier**
3. **Random Forest Classifier**
4. **KNN Classifier**

* **Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models like Random Forest Classifier and XGBoost classifier.

**7.1. Algorithms:**

1. **Logistic Regression:**

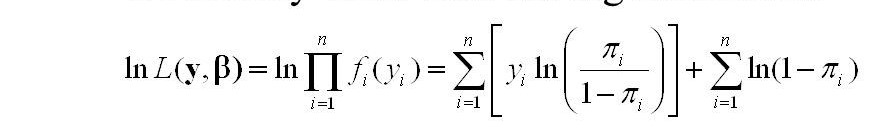
Logistic Regression is a classification algorithm that has been given the name regression due to its mathematical resemblance to linear regression. It used to find the probability of event success and event failure.

The function used in Logistic Regression is sigmoid function or the logistic function given by:

f(x)= 1/1+e ^(-x)



The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:



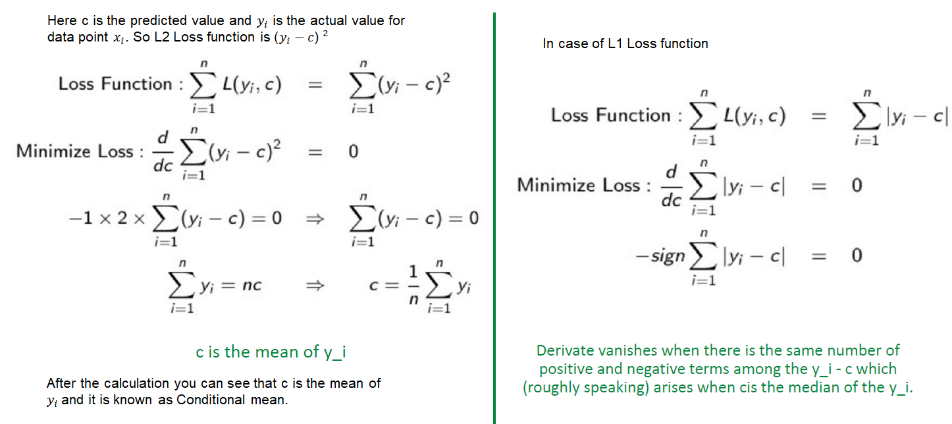
Its major drawback is, it also follows linearity assumptions and could only be used to predict discrete functions.

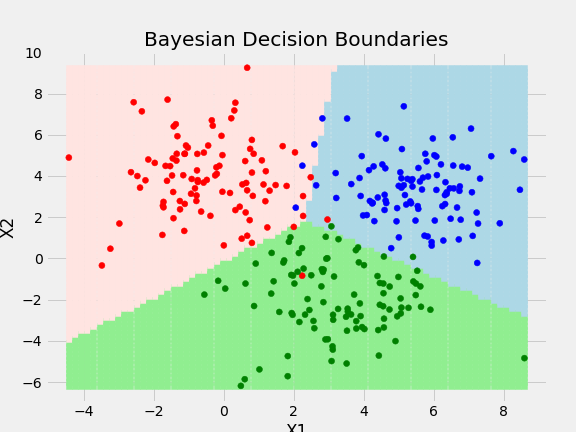
.

1. **K Nearest Neighbor Classifier:**

KNN algorithm is a lazy learning algorithm involving retrieving the K datapoints that are nearest in distance to the original point by comparing the distance from its neighbors.

Loss function of KNN is given by :

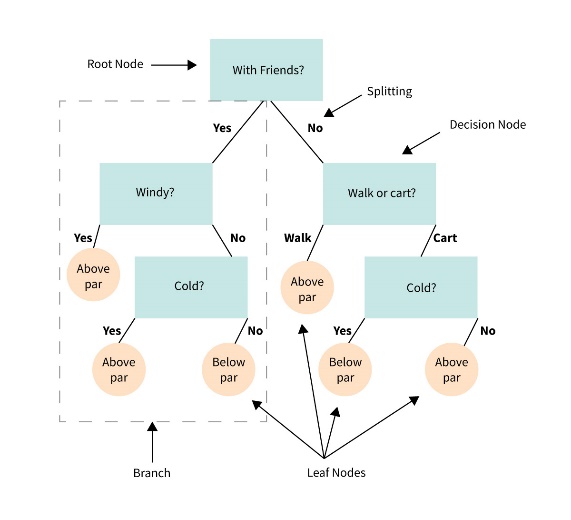




Major drawback of KNN is does not works well for large or high dimensional datasets. It also needs feature scaling and is very sensitive to noisy data and missing data.

1. **Decision Tree:**

A decision tree is a support tool with a tree-like structure that models probable outcomes, cost of resources, utilities, and possible consequences. Decision trees provide a way to present algorithms with conditional control statements.



The drawback is, it is prone to overfitting and unstable algorithm.

1. **Random Forest Classifier:**

The Random Forest Decision Tree Algorithm creates multiple decision trees from a randomly chosen subset of the training set, collects the labels from each subset, and then averages the final prediction according to the most number of times each label has been predicted.

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.



**7.2. Model performance:**

Following are most common evaluation metric for classification algorithms:

1. **Confusion Matrix**-

A confusion matrix summarizes how well a classification model is able to classify examples into specific classes. One axis of the confusion matrix is the predicted label, and the other axis is the actual label. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

1. **Precision/Recall**-

Precision is the ratio of correct positive predictions to the overall number of positive predictions:

TP/TP+FP

Precision is choice of metric when we specifically want to focus on false positives.

Recall is the ratio of correct positive predictions to the overall number of positive examples in the set:

TP/FN+TP

Recall is evaluation choice of metric when want to keep an eye on False negatives.

1. **Accuracy**-

Accuracy is given by the number of correctly classified examples divided by the total number of classified examples. It is choice of metric for balanced dataset. In terms of the confusion matrix, it is given by:

TP+TN/TP+TN+FP+FN

1. **Area under ROC Curve(AUC)**-

An ROC curve combines the true positive rate (the proportion of positive examples correctly predicted, defined exactly as recall) and false positive rate (the proportion of negative examples incorrectly predicted) to illustrate the classification performance.

**7.3. Hyper parameter tuning:**

A hyperparameter is a set of information that gives an algorithm some control over how it learns. In order to learn from these hyperparameters, their definitions impact the parameters of the models. This set of values impacts performance, stability, and interpretation of a model. Each algorithm requires a specific grid of hyperparameters that can be customized to fit the business problem. The hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate output.

We used Randomized Search CV for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds.

**Randomized Search CV**

Random search is a technique where random combinations of the hyperparameters are used to find the best solution for the built model The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The drawback of random search is that it yields high variance during computing. Since the selection of parameters is completely random; and since no intelligence is used to sample these combinations, luck plays its part.

**8. Conclusion:**

Lets take a look at what we done in this particular project. First of all we load the dataset and performed data cleaning and null value treatment. Thankfully there were no null values and duplicate values are present in the dataset.

We did outlier treatment by using IQR caping method and did some EDA and data visualization. We plot different graphs of univariate analysis and bivariate analysis and make different inference from our dataset.

At the end part of EDA we plot the correlation heatmap and find the correlation of each independent variable with our target variable.

There was class imbalance in our target variable ie. More than 50% difference between both the classes .To treat class imbalance we used SMOTE technique while doing modeling.

In feature engineering part we used ExtraTree classifier to check the feature importance of each variable i.e which feature is more important compared to our model and which is of less importance. Then we used ANOVA to select the best feature which we will be using further in our model.

In modeling part we tried four different classification models which are logistic regression, Decision tree classifier, Random Forest classifier and KNN. And we got the following results:

1. Logistic Regression, Decision Trees, Random Forest algorithms were implemented. The important metric to compare all the algorithms in this case is ‘Recall’. As the company can’t afford to predict False negative i.e. predict defaulter as a non defaulter. Since, company is one, who will give to money to the customers,if, for any reason giving money to defaulter is gaining more risk to getting the investment back. Hence, here identifying false negative is important.
2. Logistic Regression had an imbalance in the recall score of about 83% for class 0 and 56% for class 1.
3. Performance on Decision Tree and Random Forest is comparatively better. Decision Trees and Random Forest have recall scores of 75%(class 0) , 49%(class 1) and 65%(class 0), 66%(class 1) respectively.
4. KNN classifier could be a good model but it needs further hypertuning.

**References-**

1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya