**CREDIT CARD DEFAULT DETECTION SYSTEM**

By

**MWUOHI JUSTINS MIANO**

Registration Number: **SC200/0146/2018**

A Report Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of Bachelor of Science in Mathematics and Computer Science, Department of Computer Science, School of Computing and Information Technology, Murang’a University of Technology.

# 

# **DECLARATION AND APPROVAL**

I declare that this research is my original work and has not been submitted for any academic award in any other university.

Signature………………….…………. Date……………………….……

Mwuohi Justins Miano.

SC\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This report has been submitted for examination with my approval as the university supervisor.

Signature…………………………… Date…………………………….

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lecturer Department of Computer Science

School of Computing and Information Technology

Murang’a University of Technology.

# **DEDICATION**

This report is dedicated to my parents for their love, support, and inspiration all through my academic journey. I cannot forget about my siblings for their tremendous support which helped me sail though this academic journey.

# 

# **ACKNOWLEDGEMENT**

All form of praise and thanks is due to Almighty God, the creator of mankind, the most merciful and gracious for his blessings, protections, courage and guidance.

I would like to acknowledge and thank my lecturer, who stood by me and made sure I did everything regarding this project efficiently and effectively, my gratitude to him is unmatched. I also thank my parents for their immense support, guidance and encouragement to make sure that I complete this program successfully. I would also like to acknowledge my lecturers and all who have contributed to this point in the success of my academic pursuit. I also thank my colleagues for their great support.

# **ABSTRACT**

In modern day’s credit card plays an important role in every person’s daily activity. Customer purchases their needs with their credit cards and online transitions. Banks and financial institutes consider denying the credit applications of customers to avoid the risk of defaulters. Credit risk is the rise of debt on the customer who fails to make the billing payment for some period. The purpose of the project is how to reduce the defaulters among the list of customers, and make a background check on whether to provide the loan or not and to find the promising customers. These predictive models would benefit the lending institutions and to the customers as it would make them more aware of their potential defaulting rate. The problem is a binary classification problem whether a customer will be defaulting to pay next month payment. The dataset is unbalanced so the focus was on the precision and recall more than the accuracy metrics. After comparison with precision-recall curve, logistic regression is the best model based on the False Negative value of confusion metrics. Moreover, after changing the threshold value of the logistic regression, GUI (Graphical user interface) implemented and predicted whether a customer is defaulter or not-defaulter.

**Table of Contents**

**[DECLARATION AND APPROVAL](#_Toc83294827)** [i](#_Toc83294827)

**[DEDICATION](#_Toc83294828)** [ii](#_Toc83294828)

**[ACKNOWLEDGEMENT](#_Toc83294829)** [iii](#_Toc83294829)

**[ABSTRACT](#_Toc83294830)** [iv](#_Toc83294830)

**[List of Figure](#_Toc83294831)** [vii](#_Toc83294831)

**[CHAPTER ONE](#_Toc83294832)** [1](#_Toc83294832)

**[1.1](#_Toc83294833)****[Introduction](#_Toc83294833)** [1](#_Toc83294833)

**[1.2](#_Toc83294834)****[Problem Statement](#_Toc83294834)** [2](#_Toc83294834)

**[1.3](#_Toc83294835)****[Objectives](#_Toc83294835)** [3](#_Toc83294835)

**[1.3.1](#_Toc83294836)****[Main Objective](#_Toc83294836)** [3](#_Toc83294836)

**[1.3.2](#_Toc83294837)****[Specific Objectives](#_Toc83294837)** [3](#_Toc83294837)

**[1.4](#_Toc83294838)****[Significance of the study](#_Toc83294838)** [3](#_Toc83294838)

**[1.5](#_Toc83294839)****[Scope of Study](#_Toc83294839)** [4](#_Toc83294839)

**[1.6](#_Toc83294840)****[Limitations](#_Toc83294840)** [4](#_Toc83294840)

**[CHAPTER TWO](#_Toc83294841)** [5](#_Toc83294841)

**[2.1 Introduction](#_Toc83294842)** [5](#_Toc83294842)

**[2.2 Existing Systems](#_Toc83294843)** [5](#_Toc83294843)

**[2.2.1](#_Toc83294844)****[NoFraud](#_Toc83294844)** [5](#_Toc83294844)

**[2.2.2](#_Toc83294845)****[Signifyd](#_Toc83294845)** [6](#_Toc83294845)

**[2.2.3](#_Toc83294846)****[Iovation](#_Toc83294846)** [6](#_Toc83294846)

**[2.2.4](#_Toc83294847)****[SAS anti-fraud solutions](#_Toc83294847)** [6](#_Toc83294847)

**[2.3](#_Toc83294848)****[Existing Credit card fraud detection methods, IDE & Languages](#_Toc83294848)** [7](#_Toc83294848)

**[2.3.1](#_Toc83294849)****[Naïve Bayes](#_Toc83294849)**[. 7](#_Toc83294849)

**[2.3.2](#_Toc83294850)****[Logistic Regression](#_Toc83294850)** [10](#_Toc83294850)

**[2.3.3](#_Toc83294851)****[Principal Component Analysis (PCA)](#_Toc83294851)** [11](#_Toc83294851)

**[2.3.4](#_Toc83294852)****[Local Outlier Factor (LOF)](#_Toc83294852)** [12](#_Toc83294852)

**[2.3.5](#_Toc83294853)****[One Class Support Vector Machine](#_Toc83294853)** [13](#_Toc83294853)

**[2.3.6](#_Toc83294854)****[Isolation Forest](#_Toc83294854)** [14](#_Toc83294854)

**[2.3.7](#_Toc83294855)****[K-Nearest Neighbours](#_Toc83294855)** [14](#_Toc83294855)

**[2.3.8](#_Toc83294856)****[IDE and Language](#_Toc83294856)** [15](#_Toc83294856)

**[2.3.9](#_Toc83294857)****[Justification](#_Toc83294857)** [15](#_Toc83294857)

**[2.4](#_Toc83294858)****[Conclusion](#_Toc83294858)** [16](#_Toc83294858)

**[CHAPTER THREE](#_Toc83294859)** [17](#_Toc83294859)

**[3.1 Introduction](#_Toc83294860)** [17](#_Toc83294860)

**[3.2 Data Collection Techniques](#_Toc83294861)** [17](#_Toc83294861)

**[3.2.1 Questionnaires](#_Toc83294862)** [17](#_Toc83294862)

**[3.2.2 Interview Schedules](#_Toc83294863)** [17](#_Toc83294863)

**[3.2.3](#_Toc83294864)****[Observation Method](#_Toc83294864)** [18](#_Toc83294864)

**[3.2.4](#_Toc83294865)****[Secondary Data Collection](#_Toc83294865)** [18](#_Toc83294865)

**[3.2.5](#_Toc83294866)****[Justification](#_Toc83294866)** [18](#_Toc83294866)

**[3.3](#_Toc83294867)****[Software Development Techniques](#_Toc83294867)** [19](#_Toc83294867)

**[3.3.1](#_Toc83294868)****[Waterfall Model](#_Toc83294868)** [19](#_Toc83294868)

**[3.3.2](#_Toc83294869)****[Prototyping](#_Toc83294869)** [21](#_Toc83294869)

**[3.3.3](#_Toc83294870)****[Spiral model](#_Toc83294870)** [22](#_Toc83294870)

**[3.3.4](#_Toc83294871)****[Agile Development](#_Toc83294871)** [23](#_Toc83294871)

**[3.3.5](#_Toc83294872)****[Choice of the model](#_Toc83294872)** [24](#_Toc83294872)

**[3.4](#_Toc83294873)****[System Requirements](#_Toc83294873)** [25](#_Toc83294873)

**[3.4.1](#_Toc83294874)****[Hardware Requirements](#_Toc83294874)** [25](#_Toc83294874)

**[3.4.2](#_Toc83294875)****[Software Requirements](#_Toc83294875)** [25](#_Toc83294875)

**[3.4.3](#_Toc83294876)****[Conclusion](#_Toc83294876)** [26](#_Toc83294876)

**[CHAPTER 4](#_Toc83294877)** [27](#_Toc83294877)

**[4.1. Introduction](#_Toc83294878)** [27](#_Toc83294878)

**[4.2 System Design](#_Toc83294879)** [27](#_Toc83294879)

**[4.2.1 Dataset](#_Toc83294880)** [28](#_Toc83294880)

**[4.2.2 Data Processing](#_Toc83294881)** [33](#_Toc83294881)

**[4.2.3 Feature Selection](#_Toc83294881)** [33](#_Toc83294881)

**[4.3 Implementation Approaches](#_Toc83294882)** [33](#_Toc83294882)

**[4.3.1 Parallel Implementation](#_Toc83294883)** [34](#_Toc83294883)

**[4.3.2 Accuracy Measures](#_Toc83294884)** [34](#_Toc83294884)

**[4.3.3 Precision](#_Toc83294885)** [34](#_Toc83294885)

**[4.3.4 Recall](#_Toc83294886)** [35](#_Toc83294886)

**[4.4 Cross Validation](#_Toc83294888)** [35](#_Toc83294888)

**[4.5 Libraries Used for implementation](#_Toc83294889)** [42](#_Toc83294889)

**[4.6 Modification and Improvements](#_Toc83294896)** [46](#_Toc83294896)

**[CHAPTER 5](#_Toc83294897)** [47](#_Toc83294897)

**[5.1 Test Reports](#_Toc83294898)** [47](#_Toc83294898)

**[5.2 User Documentation](#_Toc83294899)** [50](#_Toc83294899)

**[CHAPTER 6](#_Toc83294900)** [53](#_Toc83294900)

**[6.1 Conclusion and Future Works](#_Toc83294901)** [53](#_Toc83294901)

**[References](#_Toc83294902)** [55](#_Toc83294902)

**[Appendix 1: Budget](#_Toc83294903)** [60](#_Toc83294903)

**[Appendix 2: Schedule](#_Toc83294904)** [6](#_Toc83294904)3

# **List of Figure**

[Figure 1: Naïve Bayes Formula](#_Toc83198417) 9

[Figure 2: Logistic Regression Graph 11](#_Toc83198418)

[Figure 3: Local Outlier Factor Graph 13](#_Toc83198419)

[Figure 4: Support Vector Machine Graph. 14](#_Toc83198420)

[Figure 4.1: Support Vector Machine Graph](#_Toc83198421) 14

[Figure 5: Isolation Forest Formula 16](#_Toc83198422)

[Figure 5.1: Isolation Forest Graph 16](#_Toc83198423)

[Figure 6: K-Nearest Neigbor Graphs 16](#_Toc83198424)

[Figure 7: Visual Studio Code Editor 18](#_Toc83198425)

[Figure 8: Dataset 32](#_Toc83198426)

[Figure 9: Precision Recall Table 34](#_Toc83198427)

[Figure 10: Number of Defaulter and non-defaulters 36](#_Toc83198428)

[Figure 11: One hot-encoding categorical columns education 37](#_Toc83198429)

[Figure 12: Accuracy precision and recall for various alogrithms 40](#_Toc83198430)

[Figure 13: PRC comparison 41](file:/C:/Users/Seen%20Technologies/Desktop/m16/Project%20Report/Odhiambo-Atieno-Project-Proposal-Final.docx" \l "_Toc83198431)

[Figure 13.1: LR Classifier 41](file:/C:/Users/Seen%20Technologies/Desktop/m16/Project%20Report/Odhiambo-Atieno-Project-Proposal-Final.docx" \l "_Toc83198432)

[Figure 14: Python classes and libraries import 42](file:/C:/Users/Seen%20Technologies/Desktop/m16/Project%20Report/Odhiambo-Atieno-Project-Proposal-Final.docx" \l "_Toc83198433)

[Figure 15: Lables design with Tkinter 43](file:/C:/Users/Seen%20Technologies/Desktop/m16/Project%20Report/Odhiambo-Atieno-Project-Proposal-Final.docx" \l "_Toc83198434)

[Figure 15.1: Models implementation 43](file:/C:/Users/Seen%20Technologies/Desktop/m16/Project%20Report/Odhiambo-Atieno-Project-Proposal-Final.docx" \l "_Toc83198435)

[Figure 16: Logical Computations 44](#_Toc83198436)

[Figure 17: Input Form 43](#_Toc83198437)

[Figure 18: Filled Input Form 46](#_Toc83198437)

[Figure 17: Filled Input Form With results 47](#_Toc83198437)

**List of Tables**

[Table 1: Attributes of the selected dateset 34](#_Toc83198415)

[Table 2: One hot encoding 36](#_Toc83198416)

[Table 3: Tabulation of accuracy, precision, recall for various algorithms 39](#_Toc83198415)

[Table 4: Budget 52](#_Toc83198416)

[Table 5: Schedue 53](#_Toc83198416)

# **CHAPTER ONE**

**Introduction**

## **Background Study**

Credit card is a physical card used for paying our bills easily. The cardholder could use it to give a paying promise as a requital to the cost of service and goods. There is a brief explanation of algorithms to define term credit scoring, which determines the relation between defaulters and loan characteristics. It is a useful information for financial institution to maintain financial statement and customer transaction list to reduce the uncertainty. Yeh and Lien (2009) compared the predictive accuracy of probability of default among six data mining methods (specifically, K-nearest neighbor classifier, logistic regression, discriminant analysis, naive Bayesian classifier, artificial neural networks, and classification trees) using customers default payments data in Taiwan. Their experimental results indicated that only artificial neural network could accurately estimate default probability. The use of Taiwan data is beneficial for this project because the sample size of the default payment data in Taiwan is 30,000. Currently, a variety of Machine Learning approaches used to detect fraud and predict payment defaults. Some of the more common techniques include Logistic Regression, K Nearest Neighbor, Decision Tree, Naive Bayes, Support Vector Machine, Feed Forward Neural Networks and Ensemble approaches like Voting Classifier. The dataset contains information on 24 variables, obtained from the UCI Machine Learning Repository. Here we categorized the dataset based on independent variables such as credit amount, age, sex, education, marital status, and their past loan repayment history of last 6 months, History of their past payments made (April to September), amount of bill statement, amount of previous payment. The dependent variable is default, which means whether the customer will pay their next month payment, or not. We can reduce the cost, make a good decision for a potential customer and help in reducing the time consumption for processing loan application and more.

**Problem Statement**

Can we reliably predict who has is likely to default? If so, the bank may be able to prevent the loss by providing the customer with alternative options (such as forbearance or debt consolidation, etc.). I will use various machine learning classification techniques to perform my analysis.

## **Objectives**

### **Main Objective**

To find whether the customer could pay back their next credit amount or not and Identify some potential customers for the bank who can settle their credit balance.

### **Specific Objectives**

The following are the steps followed to manage these goals:

1. Selection of dataset
2. Display some graphical information and visualize the features.
3. Check Null values in the dataset
4. Data pre-processing using one-hot encoding and remove extra parameters
5. Train with classifiers
6. Evaluate the model with test data
7. Compare the accuracy, precision and recall finding the optimal model.

## **Significance of the study**

## This project is helpful for solving a real problem facing major banking institutions by using various classification techniques. Moreover, any user can access GUI and add their gender, education, marital status and payment details to check next month in which category they fall (defaulter or non-defaulter).

## **Scope of Study**

The major purpose of risk prediction is to use information, such as financial statement, customer transaction and repayment records to predict individual customer’s credit risk and to reduce the damage and uncertainty. Many methods, including Logistic Regression and SVM has been used to develop models.

## **Limitations**

Respondents play a major role of providing the data that is needed to facilitate the completion of this project. Therefore, respondents who are uncooperative in providing the needed data will hinder the progress of the project. The data collection method used may not capture all the data that may be required to develop the system. Also, the lack of data collection method that will be embraced by all respondents will highly affect the respondents’ attitude and response depending on how the data collection method in use suits them.

# 

# **CHAPTER TWO**

**LITERATURE REVIEW**

## **2.1 Introduction**

Here I will talk about some of the already existing technologies in place to address a similar problem that is, (credit card fraud detection system) and the methods used. I will also enlist some of the languages and the integrated development environment used.

## **2.2 Existing Systems**

Credit card fraud being a major problem facing banks and financial institutions at large, organizations and engineering bodies have done a significant amount job trying to address the same. A lot of systems have been designed to contribute to curbing this problem. Here are some of them:

### **NoFraud**

This is an ecommerce fraud prevention system that combines machine learning and human intelligence. The tool screens transactions in real time using advanced machine learning algorithms, allowing merchants to concentrate on their primary tasks and goals which are fulfilling orders, interacting with customers, and expanding their business in general.

NoFraud uses thousands of data points in its decision-making process. The system takes into account historical customer data, current transaction data, and also analyzes customer behavior. For instance, it tracks a customer’s device with its activity history, location, tracks and validates IP address, as well as ensures that transaction data doesn’t match with one from global and merchant-specific fraud blacklists. NoFraud also checks transaction velocity – the number of payments made with a credit card, from a specific account, device and IP address during a certain timespan. Transaction security is complemented with bank identification number (BIN) checker service, address verification service (AVS), and card verification number (CVN) service.

### **Signifyd**

This is a system that provides a cloud-based fraud protection platform for eCommerce businesses. The solution automates real-time order screening and approval using machine-learning – all completed in a review that generally takes milliseconds. While Signifyd determines which orders are safe to ship and which are suspicious, merchants make the final decision on whether to decline or approve a transaction. An expert manual review is used only for complicated cases.

The system scores every transaction based on such parameters as location, address, historical purchase data, recent credit score, IP address, etc. To collect the historical profile information, the solution takes into account user activity on both a merchant website and all other marketplaces they visit.

Signifyd comes with numerous features for seamless order processing. It allows businesses to create client blacklists and whitelists, automate order fulfillment, and cancel guarantees on orders that have been canceled by shoppers. Signifyd refunds chargebacks on approved orders that turned out to be fraudulent. A reimbursement is carried out within 48 hours and includes chargeback fees and delivery costs. In addition, businesses can submit a claim by simply filling in a form and providing an order tracking number with the chargeback notice.

### **Iovation**

This solution provides a suite of device-based fraud protection and dynamic authentication solutions. Its products are developed for various industries. eCommerce businesses, insurance companies, financial and ticketing service providers, banks, gaming, and gambling companies, as well as online communities can protect themselves from fraudsters with iovation.

The company has four solutions, two of which – FraudForce and SureScore – ensure fraud protection. Both ML products learn from historical data that contains over 55 million fraud reports and 5 billion known devices gathered by more than 4000 iovation fraud experts. Users can also customize the review with their own business rules for any customer touchpoints (stages of customer contact with your brand.) The other two are (LaunchKey and ClearKey)

### **SAS anti-fraud solutions**

The creators of this system develops and provides analytics software suites for numerous industries, such as banking, healthcare, insurance, media, retail, government, travel and transportation, e.t.c. It also addresses problems of fraud and digital and financial assets security with its Fraud, AML (anti-money laundering) Security Intelligence.  
And so generally, [SAS anti-fraud solutions](https://www.sas.com/en_us/solutions/fraud-security-intelligence.html" \l "view-all-products) are based on a hybrid approach that combines expert rules, mathematical models, analysis of the subject’s social setting, text analytics, anomaly analysis, and other methods. A mix of technology solutions that helps businesses in their battle against any types of fraud and threats. These can be classical attacks on remote banking systems, e.g. termination of unauthorized access to mobile bank app or customer’s personal account in the online banking system. In the case of internal fraud, SAS solutions identify a complex scheme with numerous people involved, as well as various information resources and organization’s IT infrastructure nodes.

## **Existing Credit Card Fraud Detection methods, IDLEs and Programming languages.**

There are currently, a variety of Machine Learning approaches used to detect fraud and predict payment defaults. Some of the more common techniques and methods include:

### **Naïve Bayes**

It is a [classification technique](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=6stepsnaivebayesarticle" \t "_blank) based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

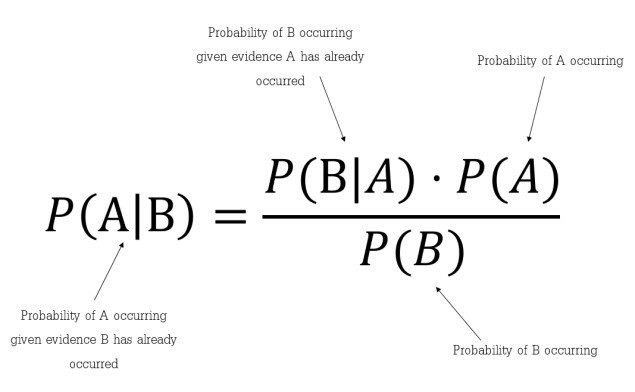
For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

P(yi | x1, x2, …, xn) = P(x1, x2, …, xn | yi) \* P(yi) / P(x1, x2, …, xn)

***Figure 1. naïve bayes formula***



**Here are some of the Pros of this approach:**

* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

**The approach also has the following Cons:**

* If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
* On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict proba are not to be taken too seriously.
* Another limitation of [Naive Bayes](https://courses.analyticsvidhya.com/courses/naive-bayes?utm_source=blog&utm_medium=naive-bayes-explained) is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

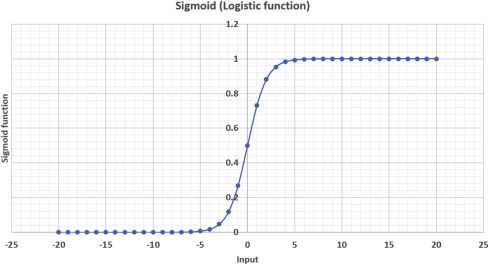
### **Logistic Regression**

Regression Analysis is a statistical tool for investigating the relationship between variables. Regression model has been vastly used in every aspect of research sciences [13-15] Regression Analysis has been discovered independently by both mathematicians Carl Friedrich and Adrien Marie Legendre in the 18th century. The regression models consist of unknown parameters (coefficients), independent variables and dependent variables. The most essential feature of regression is the use of least squares method. Least Square method is a way to use data to make quantitative predictions. Estimates for the parameters are obtained by minimizing the sum of squares of differences between the observed values and the predicted values under the model [16]. Linear regression, where the dependent variable is a linear combination of parameters (coefficients). In simple linear regression, there is one dependent variables and one independent variable, equation 1.1. In Multiple linear regression (MLR) there can be several independent variables of functions if independent variables, equation 1.2.

𝑦 = 𝛽0 + 𝛽1𝑥1 + 𝜀   
𝑦 = 𝛽0 + 𝛽1𝑥1 + 𝛽2𝑥2 + ⋯ + 𝛽𝑘𝑥𝑘 + 𝜀

Here is an example of an output:

***Figure 2. Logistic Regression graph***



### **Principal Component Analysis (PCA)**

Principal Component Analysis (PCA) is a standard statistical tool used in analyzing multidimensional data. It is widely used in almost all areas of research where manipulation of large numbers of attributes is necessary. It is a non-parametric method useful for obtaining relevant information from a complex data set. PCA is used to reduce the dimensionality of a data set, which consists of a large number of interrelated attributes, while retaining as much of the variation present in the original data set as possible. This process is done by linear transformation of the original set of attributes into a smaller set of attributes called principal components (PCs). Principal components are uncorrelated and ordered so that the first few retain most of the variation present in all of the original attributes.

The basic steps of PCA are as follows:

**tep 1:** Standardize the dataset.

**Step 2:**Calculate the covariance matrix for the features in the dataset.

**Step 3:**Calculate the eigenvalues and eigenvectors for the covariance matrix.

**Step 4:**Sort eigenvalues and their corresponding eigenvectors.

**Step 5:**Pick k eigenvalues and form a matrix of eigenvectors.

**Step 6:** Transform the original matrix.

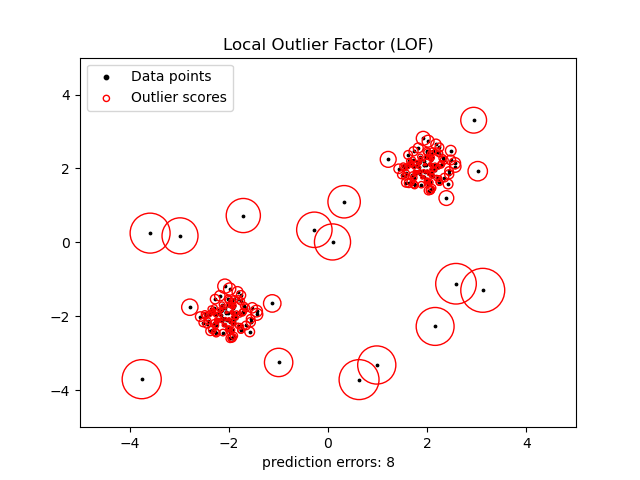
### **Local Outlier Factor (LOF)**

Local Outlier Factor (LOF) is an algorithm used for finding anomalous data points given their deviation in density with respect to their neighbors. If a point has a much lower density than the density of its neighbors, then it has a high (»1) LOF score and can be considered an outlier. For our clothing store example, this might look like a store located on the edge of town where the majority are clustered in its core. However, if all stores in a town were more sparsely located across it, the LOF score would be lower as its neighbors are not exhibiting any real clustering behavior. In calculating the LOF, users must define *k* which specifies the *k-distance* which is the distance of each point to its kth neighbor. A k-value of 5 would take the k-distance as the distance from the point to its 5th nearest neighbor. Smaller k-values produce more localized results, but are more sensitive to noise in the data.

Local outlier factor algorithm can be divided into four parts:

* K-Distance and K-Neighbors
* Reachability Distance
* Local Reachability Density
* Local Outlier Factor Calculation

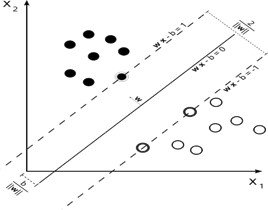
***Figure 3. Local Outlier Factor Graph***



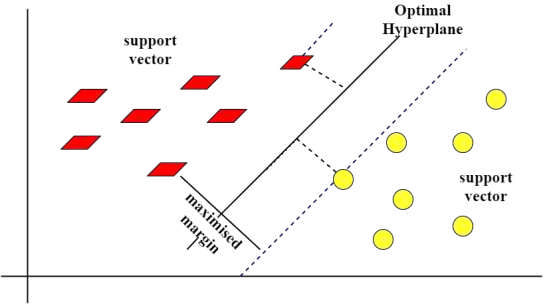
### **One Class Support Vector Machine**

Basically, the support vector machine is a machine learning model that can be used for classification and regression analysis. Mostly it is used with classification problems. One of the most important qualities of SVM is that it creates nonlinear decision boundaries by projecting the data with higher dimensions in the space using its nonlinear function. It uses its function to lift the feature space F of the observations of the I space which cannot be separated by a linear function or straight line. lifted feature space can be separated by the straight hyperplane. This hyperplane is used to separate the data of one class from the other class data. This hyperplane can be the form of a nonlinear curve.

So the one class support vector machine is where the detection of novelty is done by separating the data points from the feature space and maximizing the distance from the hyperplane to the feature space. Resulting in functions that focus on the space where the density of the is maximum so that function can retire +1 if the observation is in a dense region and -1 if the observation belongs to the low dense space.



***Figure 4. Support Vector machine graph***



***Figure 4.1. Support Vector machine graph***

### **Isolation forest**

Isolation Forest algorithm utilizes the fact that anomalous observations are few and significantly different from ‘normal’ observations. The forest is built on the basis of decision trees, each of the trees having access to a sub-sample of the training data. In order to create a branch in the tree, first, a random feature is selected. Afterward, a random split value (between min and max value) is chosen for that feature. If the given observation has a lower value of this feature then the one selected it follows the left branch, otherwise the right one. This process is continued until a single point is isolated or specified maximum depth is reached.

In principle, outliers are less frequent than regular observations and are different from them in terms of values (they lie further away from the regular observations in the feature space). That is why by using such random partitioning they should be identified closer to the root of the tree (shorter average path length, i.e., the number of edges an observation must pass in the tree going from the root to the terminal node), with fewer splits necessary.

The anomaly score is created on the basis of all trees in the forest and the depth the point reaches in these trees.

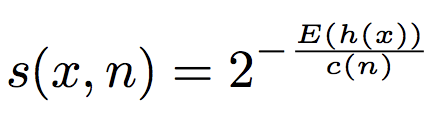
## The Algorithm

Given a sample of data points X, the Isolation Forest algorithm builds an Isolation Tree (iTree), T, using the following steps.

1. **Randomly select an attribute**q **and a split value**p.
2. **Divide**X **into two subsets by using the rule**q < p**. The subsets will correspond to a left subtree and a right subtree in**T.
3. **Repeat steps 1–2 recursively until either the current node has only one sample or all the values at the current node have the same values.**

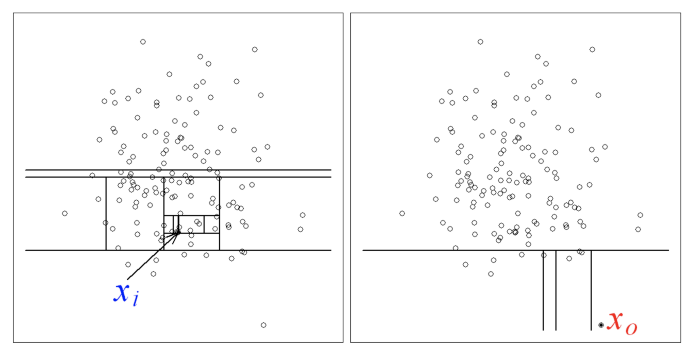
The formula is as follow:

***Figure 5. Isolation Forest Formula***



Where h(x) is the path length of observation x, c(n) is the average path length of unsuccessful search in a Binary Search Tree and n is the number of external nodes.

***Figure 5.1 Isolation Forest Graphs***



### **K-Nearest Neighbors**

The **k-nearest neighbors (KNN) algorithm** is a data classification method for estimating the likelihood that a data point will become a member of one group or another based on what group the data points nearest to it belong to.

The k-nearest neighbor algorithm is a type of [supervised machine learning](https://learn.g2.com/supervised-learning" \t "_blank) algorithm used to solve classification and regression problems. However, it's mainly used for classification problems.

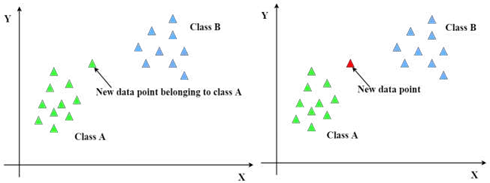
KNN is a lazy learning and non-parametric algorithm.

It's called a lazy learning algorithm or lazy learner because it doesn't perform any training when you supply the training data. Instead, it just stores the data during the training time and doesn't perform any calculations. It doesn't build a model until a query is performed on the dataset. This makes KNN ideal for [data mining](https://www.g2.com/articles/data-mining" \t "_blank).

It's considered a non-parametric method because it doesn’t make any assumptions about the underlying data distribution. Simply put, KNN tries to determine what group a data point belongs to by looking at the data points around it.

This method is not vulnerable to noise and missing data points, which means composing larger datasets in less time. Moreover, it is quite accurate and requires less work from a developer in order to tune the model.

***Figure 6. K-Nearest neighbors Graphs***



### **Integrated Development Environments**

**Visual Studio Code**

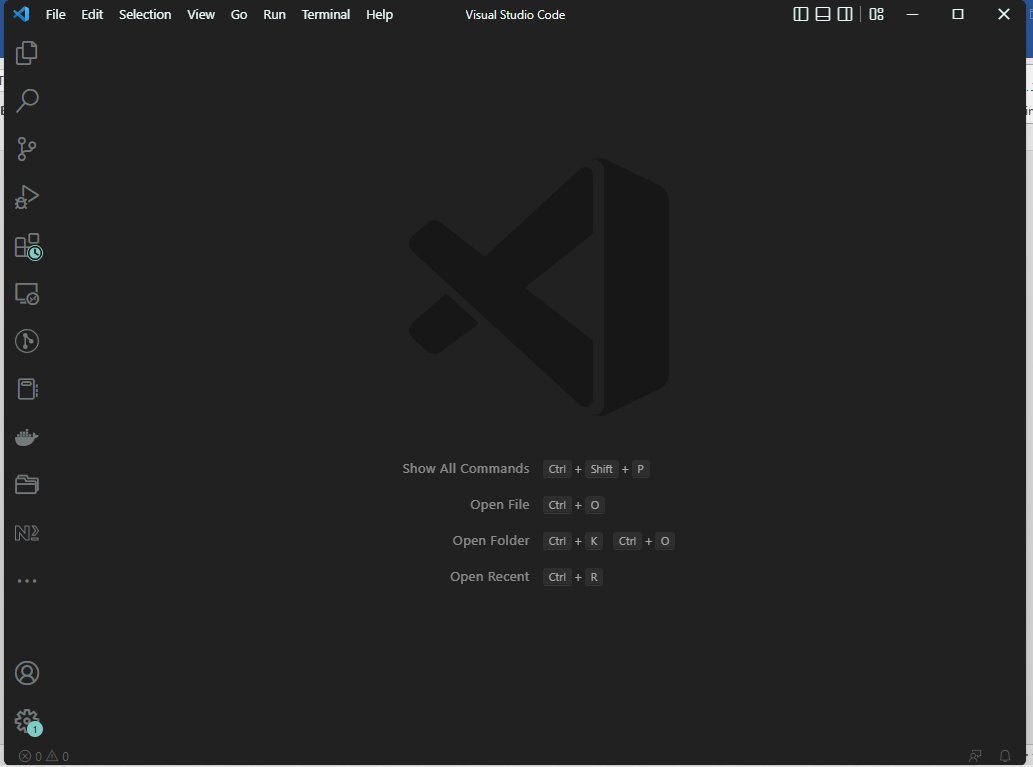
This is a free open source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

**Features of Visual Studio Code**  
VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go, and Dockerfile. Moreover, VS Code allows you to add on and even creating new extensions including code linters, debuggers, and cloud and web development support.

The VS Code user interface allows for a lot of interaction compared to other text editors. To simplify user experience, VS Code is divided into five main regions:

* The activity bar
* The side bar

***Figure 7. Visual Studio Code Editor***



* Editor groups
* The panel
* The status bar
* Minimap: previewing entire file in a sidebar,
* Selecting and editing in several sections of code in parallel (multi-cursor),
* Bookmark even within files,
* Automatic backup,
* Search and replace with plain text or regular expressions,
* Support for macros and plug-ins written in TypeScript ou Javascript,
* Customizing keyboard shortcuts.

There are many advantages over any other IDE; they are as follow:

1. Cross-platform support :

* Windows
* Linux
* Mac

2. Light-weight

3. Robust Architecture

4. Intelli-Sense

5. Freeware: Free of Cost- probably the best feature of all for all the programmers out there, even more for the organizations.

6. Many users will use it or might have used it for desktop applications only, but it also provides great tool support for Web Technologies like; HTML, CSS, JSON.

**Python programming language**

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Python programming has the capabilities for a programmer to design and implement predictive analytic technologies using machine learning, Artificial intelligence, Big Data etc. Python is a cross-functional, maximally interpreted language that has lots of advantages to offer. The object-oriented programming language is commonly used to streamline large complex data sets. Over and above, having a dynamic semantics plus unmeasured capacities of **RAD** (rapid application development), Python is heavily utilized to script as well. There is one more way to apply Python – as a coupling language.

### **Justification**

Having studied the above existing software design and development tools. I used Python as my preferred programming languages to implement the code. I also made use of *numpy* library for multidimensional array used to store of same datatype. *Pandas* library provide high performance and used for data analysis tools. Tensorflow library implemented in order to implement Artificial Neural Network. Sklearn library is used which has various features of classification, regression including SVM, KNN, gradient boosting etc. I also made use of matplotlib, pyplot for comprehensive 2D/3D plotting and displaying in understandable manner. Keras is a high-level API to build and train deep learning model. It is user friendly and composable. Seaborn is a visualization library based on matplotlib. Graphviz is not a python package; it simply put the graphviz files into our virtual directory. Tkinter is used to create a Graphical User Interface.

## **Conclusion**

Based on the study and analysis of the currently available techniques, libraries and software system available for the purpose of curbing, credit card fraud cases, my combination is in place to make things even better, in a way of improving these existent technologies. I might not be the best as there are yet to come new ways approach but in the meant time I am optimistic that it makes a difference.

## ****CHAPTER 3 RESEARCH METHODOLOGY****

## ****Data Preprocessing****

## **Before building the predictive models, the dataset underwent a thorough preprocessing phase. This involved several steps to clean and transform the data into a suitable format for analysis. Missing values were handled by either imputing them using appropriate techniques or removing the corresponding instances. Categorical variables were encoded into numerical representations using techniques like one-hot encoding or label encoding. Feature scaling was applied to ensure that variables with different scales did not unduly influence the models.**

## ****Feature Selection****

## **To enhance the predictive performance and reduce the computational complexity, feature selection techniques were employed. Initially, a correlation analysis was conducted to identify highly correlated variables, as they may provide redundant information. Then, statistical tests such as chi-square test or ANOVA were applied to select features that had a significant impact on the target variable (credit card default). Additionally, domain knowledge and expert input were considered to prioritize relevant features for inclusion in the models.**

## ****Model Development****

## **Two machine learning algorithms, logistic regression, and support vector machine (SVM), were implemented to predict credit card default. Logistic regression was chosen for its simplicity and interpretability, while SVM was selected for its ability to handle complex decision boundaries. The dataset was divided into training and testing sets using a stratified approach to maintain the class distribution. The models were trained on the training set using the selected features and fine-tuned using appropriate hyperparameter optimization techniques, such as grid search or random search.**

## ****Model Evaluation****

## **The performance of the developed models was evaluated using several metrics, including accuracy, precision, recall, F1-score, and receiver operating characteristic area under the curve (ROC-AUC). The models were tested on the held-out testing set, and the results were compared to assess their predictive capabilities. Additionally, performance curves, such as precision-recall curves and ROC curves, were plotted to provide a visual representation of the models' performance. The evaluation metrics and curves were used to determine which model achieved better predictive accuracy and to assess their generalization capabilities.**

## ****Model Interpretation****

## **To gain insights into the factors contributing to credit card default, model interpretation techniques were employed. For logistic regression, the coefficients of the selected features were examined to understand their impact on the probability of default. Feature importance was analyzed for the SVM model using techniques like permutation importance or SHAP values. These interpretation methods helped in understanding the key drivers of credit card default and provided actionable insights for risk management and decision-making.**

## ****Ethical Considerations****

## **Throughout the project, ethical considerations were given due importance. Data privacy and security measures were strictly followed, ensuring the anonymity of individuals and compliance with legal regulations. Additionally, efforts were made to mitigate bias and unfair discrimination by carefully selecting and evaluating features that do not violate ethical standards. The project's findings and recommendations were formulated with the goal of promoting responsible lending practices and risk management while treating customers fairly and transparently.**

## **System Requirements**

This are requirements that are needed in order to design and develop the system. They are categorized as either software Requirements or Hardware requirements.

### **Hardware Requirements**

Hardware Specification:

Server:

Processor: Intel P-IV (or above)

RAM: 2GB (or above)

Hard disk: 10 GB (or above)

Client

Processor: Intel Core i5 1.8GHz or more, Intel Pentium (or above)

RAM: 2 GB RAM (or above)

Hard disk: 20 GB (or above)

### **Software Requirements**

Development Tool: Visual Studio Code or Anaconda with Jupyter Notebook

Programming language: Python and its libraries

Operating Systems: Android, Windows, Linux

## ****conclusion****

This chapter outlined the methodology followed in the project, which involved data collection, preprocessing, feature selection, model development, evaluation, interpretation, and ethical considerations. These steps were designed to ensure a robust and reliable credit card default prediction system while upholding ethical standards and addressing the needs of the financial industry.

**CHAPTER 4**

**SYSTEM DESIGN, IMPLEMENTATION AND TESTING**

## **4.1. Introduction**

Systems design is the process of defining the architecture, product design, modules, interfaces, and data for a system to satisfy specified requirements. [27]Systems design could be seen as the application of systems theory to product development. In systems design the design functions and operations are described in detail, including screen layouts, business rules, process diagrams and other documentation. The output of this stage will describe the new system as a collection of modules or subsystems. Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. Assuch, implementation is the action that must follow any preliminary thinking in order for something to actually happen. In an information technology content, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes. The word deployment is sometimes used to mean the same thing.

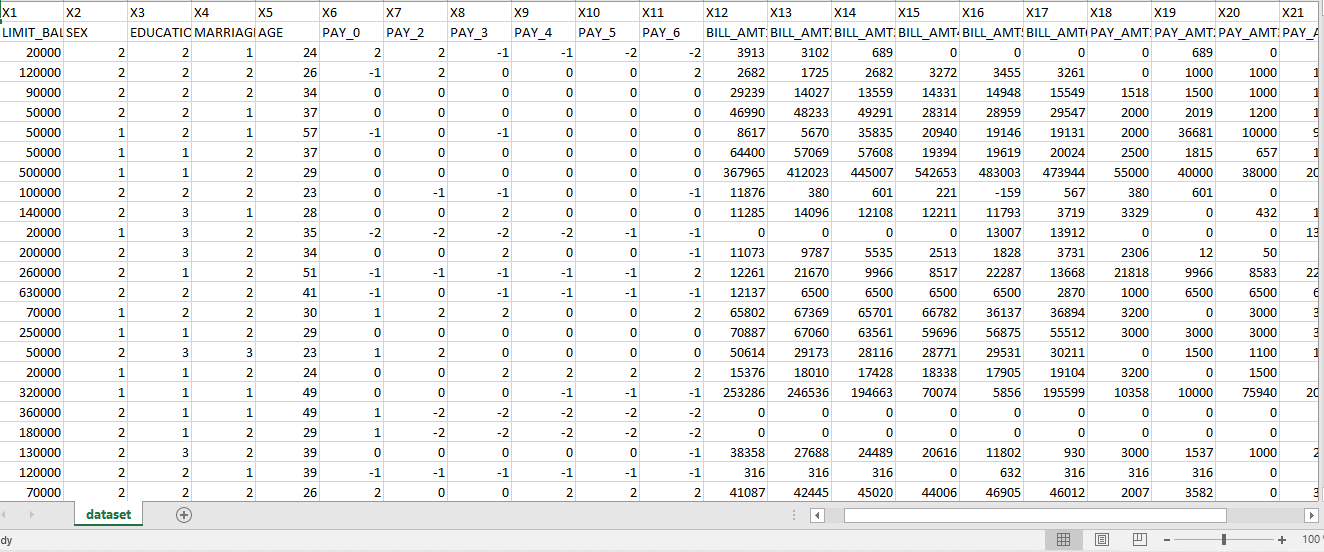
## **4.2 System Design**

To achieve the development of this system I organized everything as follows: Dataset is presented in section II. Data Pre-processing is presented in section III. Section IV describes Feature Selection. In section V training models are explained. Finally, the result and conclusion are presented in sections VI and VII, respectively

### **4.2.1 Dataset**

The system is implemented using customers default payments data in Taiwan. Their experimental results indicated that only artificial neural network could accurately estimate default probability. The use of Taiwan data is beneficial for us because the sample size of the default payment data in Taiwan is 30,000. Yeh and Lien (2009) compared the predictive accuracy of probability of default among six data mining methods (specifically, K-nearest neighbor classifier, logistic regression, discriminant analysis, naive Bayesian classifier, artificial neural networks, and classification trees) and that yield the data used in this system implementation.

***Figure 8. Dataset***



##### **4.2.2.1 Data Processing**

For efficient implementation of the classification algorithm, data preprocessing is performed before feature selection. Under-sampling is performed to make the dataset balanced to avoid the biasing of the classification algorithm towards the majority class. Feature Selection is implemented on a balanced dataset.

##### **4.2.1.2 Feature selection**

Feature selection methods are used to remove unnecessary, irrelevant, and redundant attributes from a dataset that do not contribute to the accuracy of a predictive model or which might reduce the accuracy of the model. In this paper seven feature selection techniques namely Select-K-best, Feature Importance, Extra tress classifier, Person’s correlation, Mutual Information, Step forward selection and Recursive feature elimination are used.

## **4.3 Implementation Approaches**

**4.3.1 Accuracy Measures:**

The accuracy of a model is usually determined after the model parameters learned and fixed and no learning is taking place. Then the test samples fed to the model and the number of mistakes (zero-one loss) the model makes recorded, after comparison to the true targets. Then the percentage of misclassification is calculated. In our dataset, accuracy determine how often the model predicts defaulters and non-defaulters correctly.

**4.3.2 Precision**

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. If the precision is high, then there will be low false positive rate. Here precision tells us that whenever our models predicts it is defaulter how often it is correct.

* + 1. **Recall**

Recall is the ratio of correctly predicted positive observations to the all observations in actual class. In other words, out of all positive class how much we have predicted correctly. When we apply this in our dataset, it shows the actual defaulters that the model will actually predict.

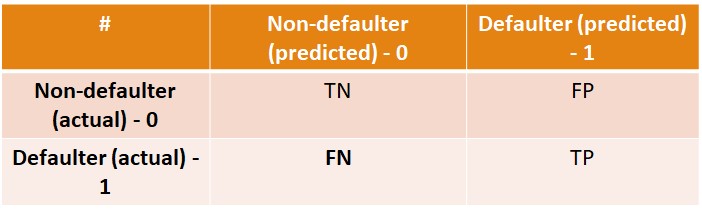
**4.3.4 Precision Recall Curve:**

It will measure the success of prediction, when classes are imbalanced. It will show the tradeoff between precision and recall threshold.

**Loss:** Loss functions let the optimization function know how well it is doing. Loss functions used in the output layer, Layers that support unsupervised layer wise pre-training. a. Cross Entropy loss: Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. b. Binary Cross Entropy: In binary classification, where the number of classes equals 2 either 0 or 1, then it is known as binary cross entropy. [9] Binary cross-entropy calculated as: −(log(𝑝) + (1 − 𝑦)log(1 − 𝑝) III. METHODOLOGY In methodology, data description, independent variable and dependent variable described with scale of variables. Moreover, in the process data preprocessing and feature engineering described as below. A. Data Description: This dataset consists of 30000 total instances and 25 features including

*Precision Recall Curve*

***Figure 9. Precision Recall table***



*Loss:* Loss functions let the optimization function know how well it is doing. Loss functions used in the output layer, Layers that support unsupervised layer wise pre-training.

1. *Cross Entropy loss:*Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.

1. *Binary Cross Entropy:* In binary classification, where the number of classes equals 2 either 0 or 1, then it is known as binary cross entropy. [9] Binary cross-entropy calculated as:

−(log(𝑝) + (1 − 𝑦)log⁡(1 − 𝑝)

*III. METHODOLOGY*

In methodology, data description, independent variable and dependent variable described with scale of variables. Moreover, in the process data preprocessing and feature engineering described as below.

1. *Data Description:*

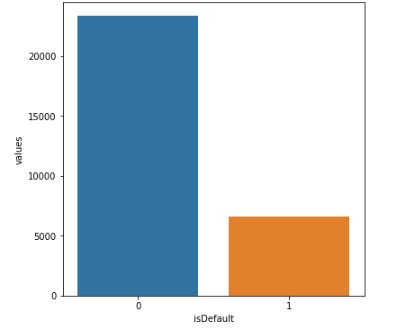
This dataset consists of 30000 total instances and 25 features including-

Table (1) attributes of the selected dataset

|  |  |  |
| --- | --- | --- |
| Independent Variable | Description | Scale of variable |
| Limit\_ BAL | Amount of the given credit (NT dollar) | Continuous  Interval |
| Sex | Gender  (1 = male,  2 = female) | Categorical Nominal |
| Education | Education (1 = graduate school,   1. = university, 2. = high school, 3. = others) | Categorical Nominal |
| Marital Status | Marital status  (1 = married,   1. = single, 2. = others) | Categorical Nominal |
| Age | Age (year) | Continuous  Interval |
| PAY\_0 to PAY\_6 | April to  September | Categorical |
| Bill\_AMT1 to Bill\_AMT6 | Amount of bill statement (NT dollar) | Continuous  Interval |
| Pay\_AMT1 to Pay\_AMT6 | Amount of previous payment (NT dollar) | Continuous  Interval |

|  |  |  |
| --- | --- | --- |
| Dependent Variable | Description | Scale of variable |
| is default | Default payment  (Yes = 1,  No = 0) | Binary |

The total number of customer based on defaulter and non-defaulter from a dataset.



***Figure 10. Number of defaulters and non-defaulters***

1. *Process:*

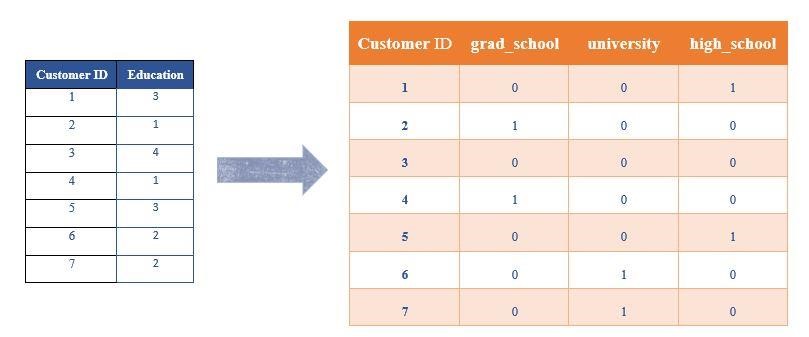
The first step is data preprocessing. Data preprocessing used to convert the raw data into a clean data set.

* ID column dropped as its unnecessary for our modeling.
* The attribute name ‘PAY\_0’ converted to ‘PAY\_1’ for naming convenience.
* Numeric attributes converted to nominal.
* One hot encoding which is a process by which categorical variables converted into a dummy form that provided to algorithms to do a better job in prediction. One hot encoder used to perform linearization of data. For instance, value in the ‘EDUCATION’ variables were grouped such that the values ‘0, 4, 5, 6’ was combined to one value and assigned a value ‘4’.

Table (2) one hot encoding

|  |  |
| --- | --- |
| EDUCATION | ENCODING |
| Grad School | 1 |
| University | 2 |
| High School | 3 |
| Others | 4 |

Converting categorical features into (n-1) features. Customer ID 1 has education value 3 which is converted to 0,0,1 as 1 is assigned to high school. Likewise, for gender Male and Female respectively 0 and 1. For Marital status, there are 4 categorical values as 1 means married, 2 means single and 3 means others. As in the dataset, there is no description about value 0, so we converted to value 3 as others. So, One-hot encoding is applied to education, gender and marital status.



***Figure 11. One-Hot encoding for categorical column education***

* Robust Scaler is used which converts all the variables in the same scale so if the data contains many outliers, scaling using the mean and variance of the data is likely to not work very well then in such cases Robust Scaler is used. For example, in Limit Balance column there are different range of values, which are converted, in proper scale.
* For all classification tasks, target variable converted to numeric.
* Next step is data preparation or feature selection where features selected by declaring the independent and target variable. Different graphs like count plots and pair plots are plotted with the reference to the target variable to check the default (=0) and non-default (=1).
* Before applying algorithms on train data, dataset is split into a ratio of 60:40, which is 60% train data and 40% is test data
* Next step is to train data by applying different algorithms as Support Vector Machine, K-Neighbors Classifier, logistic regression, Gaussian Naïve Bayes and artificial neural network.

**4.4 Cross-validation**

Cross-Validation used to assess the predictive performance of the models and to judge how they perform outside the sample to a new dataset also known as test datathe reason to use cross-validation techniques is that when we fit a model, we are fitting it to a training dataset. Without cross-validation, we only have information on how our model performs in-sample data. Ideally, we would like to see how the model performs when we have new data of customers. [10]

In cross-validation process, K-fold cross validation is used. In K-fold cross validation all observations are used for both training and validation process. Normally 10-fold cross validations process is used. (Step 10). The general process of K-fold validations is to Shuffle the dataset randomly and Split the dataset into k groups (k=10)

For Neural Network, the following are tuning parameters:

**Epochs:** One epoch is when an entire dataset passed forward and backward via NN once. Here epoch value is set to 100.

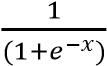
Activation function:

**ReLU:** ReLU is commonly used activation function for deep learning. This has value range from zero to infinity.

(𝑥) = max⁡(0, 𝑥)

**Sigmoid:** A sigmoid function is a differentiable, real function that defined for all real input values and has a non-negative derivative at each point.

𝐹(𝑥) =



**SGD:** Stochastic gradient descent is an iterative method for optimizing a differentiable objective function. Adam optimizer used in this project.

**Input layer:** Input layer is the very beginning of the workflow for neural network. 26 neurons used in input layer.

**Hidden Layer:** Hidden layer is in between Input Layer and Output Layer. 2 hidden layers are used after applying 1,2,3 hidden layer and found overfitting issue as we increased hidden layers. **Output Layer:** It is a predicted feature value or output variables. It is an outcome. In this dataset, there are 2 neurons in an output layer.

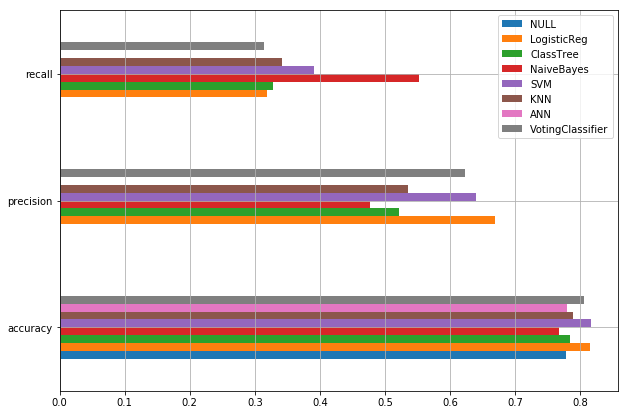
**IV. EVALUATION**

We have applied various supervised algorithm techniques for the dataset; we have tabulated the value of accuracy, precision, recall, and confusion matrix for every algorithm respectively shown below:

***Table (3) Tabulation for accuracy, precision, recall for various algorithms***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Algorithms** | **Accuracy** | **Precision** | **Recall** | **Confusion Metrix** | |
| - | Null | 78 | - | - | - |  |
| 1 | Logistic Regression | 81.45 | 66.92 | 35.9  5 | 8927  [  1806 | 419  ]  848 |
| 2 | KNN | 78.86 | 53.47 | 34.1 | 8557  [  1747 | 789  ]  907 |
| 3 | Naïve Byes | 76.68 | 47.65 |  | 7736  [  1188 | 1610  1466 |
| 4 | Classification Tree | 78.46 | 52.09 |  | 8545  [  1783 | 801  ]  871 |
| 5 | SVM | 81.66 | 63.99 | 39.1  1 | 8762  [  1616 | 584  1038 |
| 6 | Feed  Forward  NN | 75.65 | 33.91 | 40.7  4 | 8927  [  1806 | 419  ]  848 |
| 7 | Voting  Classifier | 83.95 | 67.49 | 32.8  3 | 8842  [  1822 | 504  ]  832 |

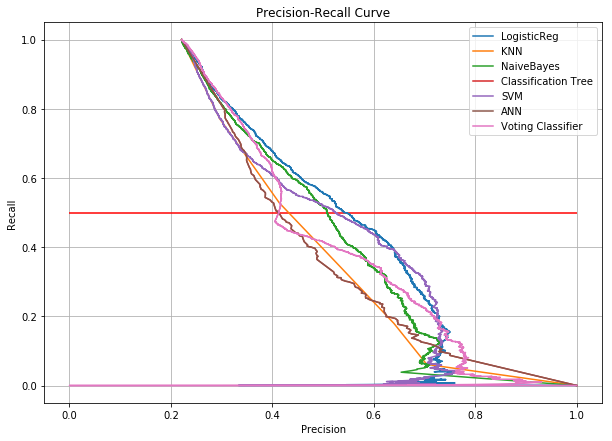
The graphical representation shown below to have a better understanding of the accuracy, precision and recall we have achieved using various algorithms.



***Figure 12. Accuracy, precision and recall for various algorithms***

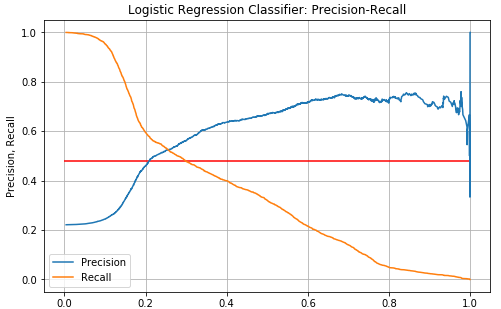
*A. Precision-Recall Curve comparison:*

The below graphical representation PRC comparison of various algorithms. By comparing algorithms, a Voting classifier has good accuracy but when we draw PRC, it shows that Logistic regression has good Precision-Recall value at threshold 0.5. So, while changing threshold values, it improves the Precision and Recall values.



Logistic Regression Classifier to check threshold value: To check threshold value and Precision, recall values at different threshold, we draw Logistic Regression classifier diagram. Here, we shown good precision and recall value at threshold 0.2. So, updated a model with threshold value 0.2 and the improvement was approx. 44% in precision and recall value of a model. As, it decreases False Negative value which means defaulters are predicted as non-defaulter. False Negative value is changed approximately 1800 to 1000 and the confusion matrix was:

[7487 1859]. 1074 1580

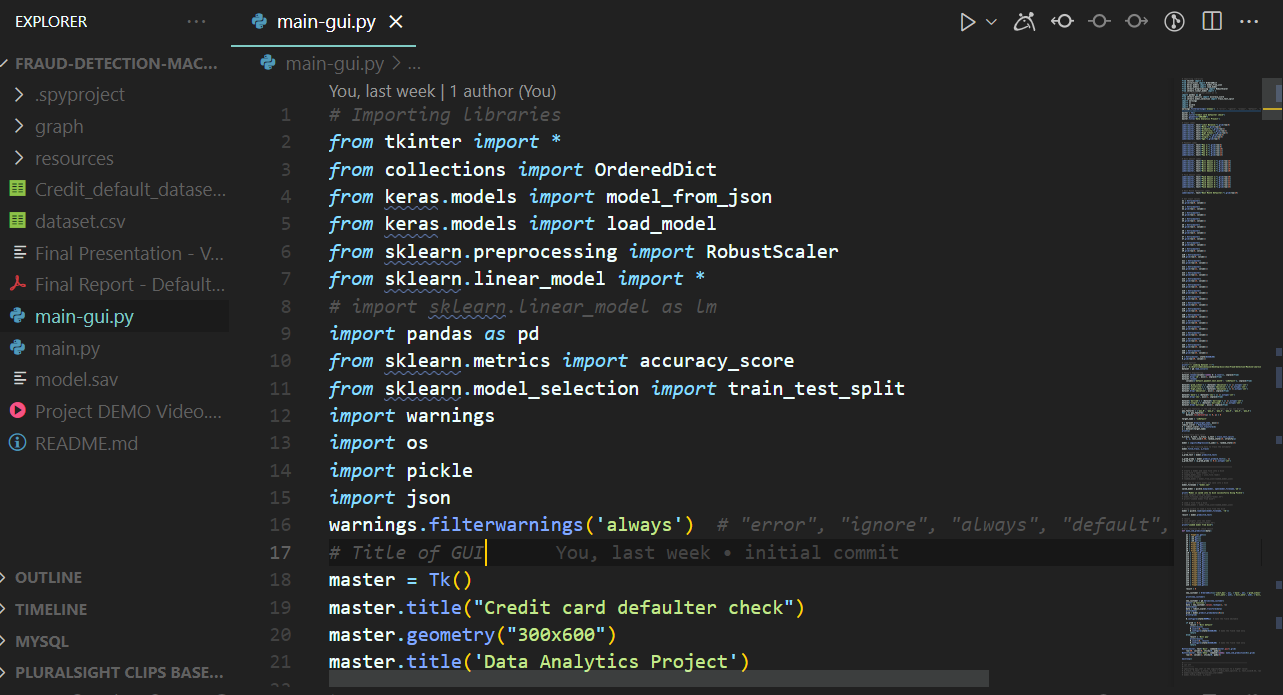


*B. Graphical User Interface:*

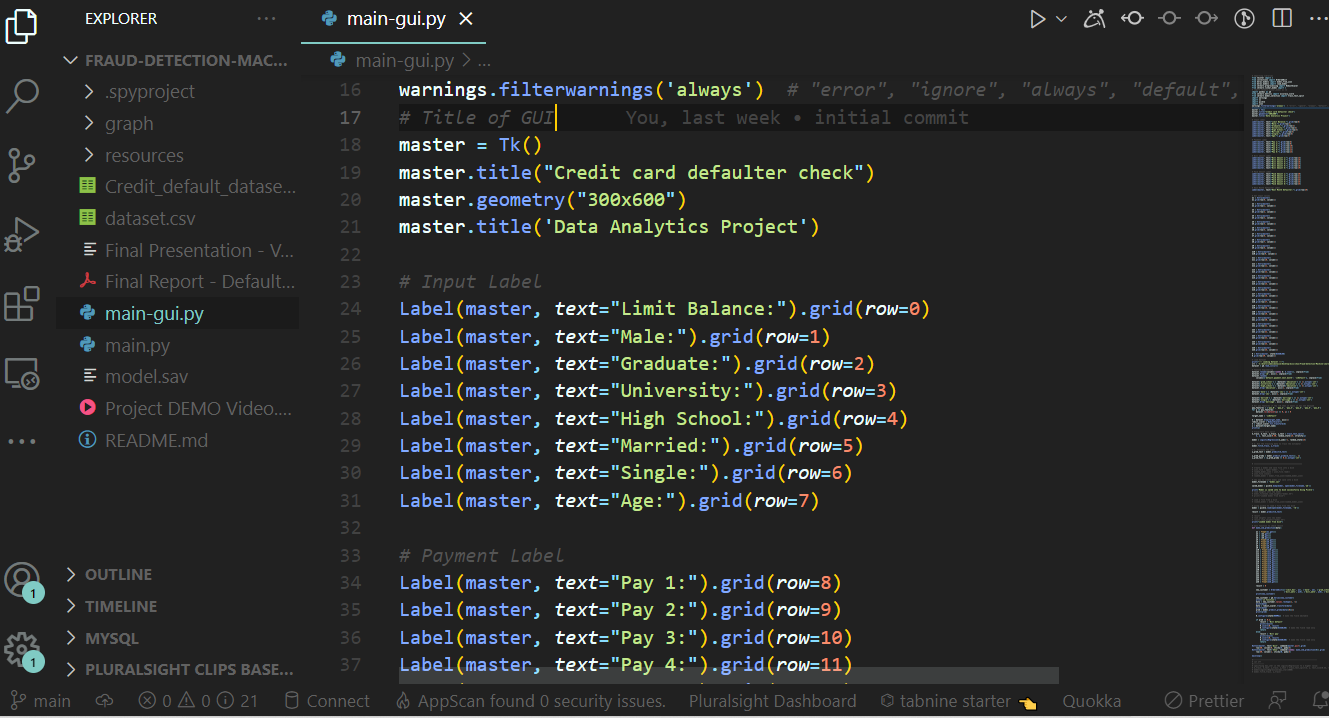
I created Graphical User Interface using python and tkinter, we trained a model and set threshold value at 0.2 in logistic regression. When user will submit below mentioned parameters value, model will predict whether a user will be defaulter or non-defaulter next month in payment.

**4.5 Libraries used for implementation**

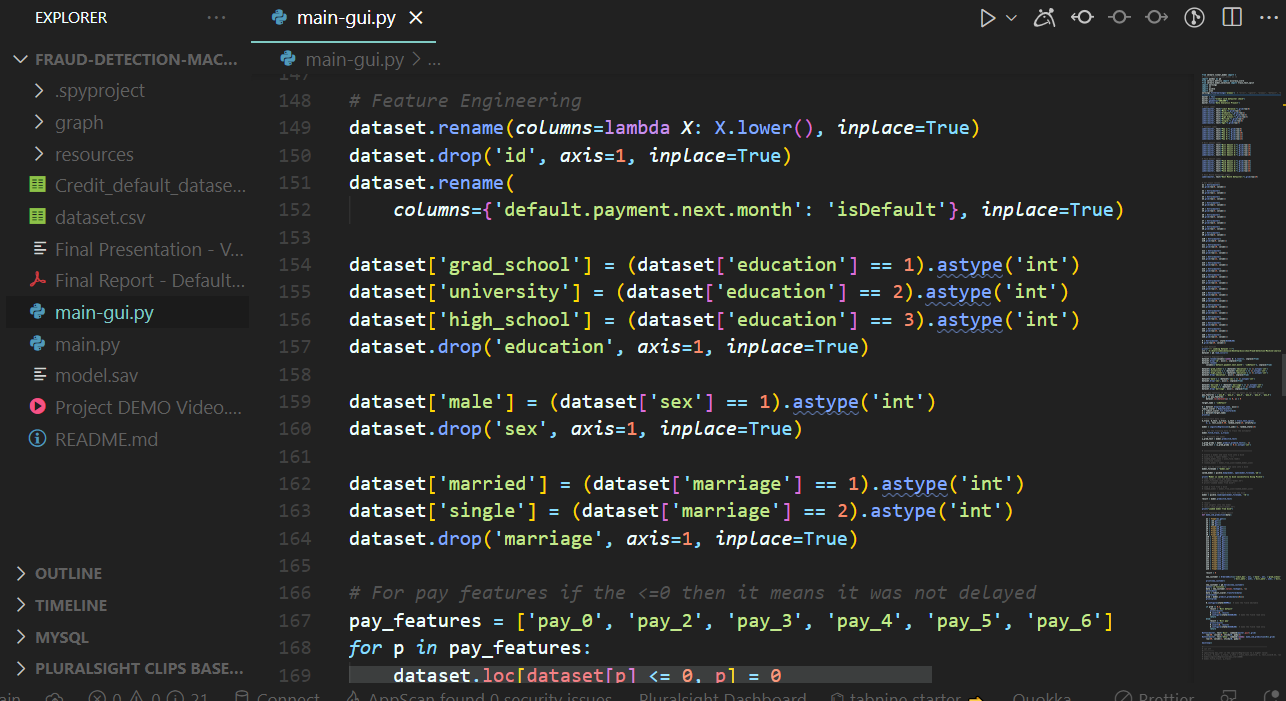
To implement the code, we have made use of Python 3.6 version. We had made use of numpy library for multidimensional array used to store of same datatype. Pandas library provide high performance and used for data analysis tools. Tensorflow library implemented in order to implement Artificial Neural Network. Sklearn library is used which has various features of classification, regression including SVM, KNN, gradient boosting etc. We make use of matplotlib.pyplot for comprehensive 2D/3D plotting and displaying in understandable manner. Keras is a high-level API to build and train deep learning model. It is user friendly and composable. Seaborn is a visualization library based on matplotlib. Graphviz is not a python package; it simply put the graphviz files into our virtual directory. Tkinter is used to create a GUI.



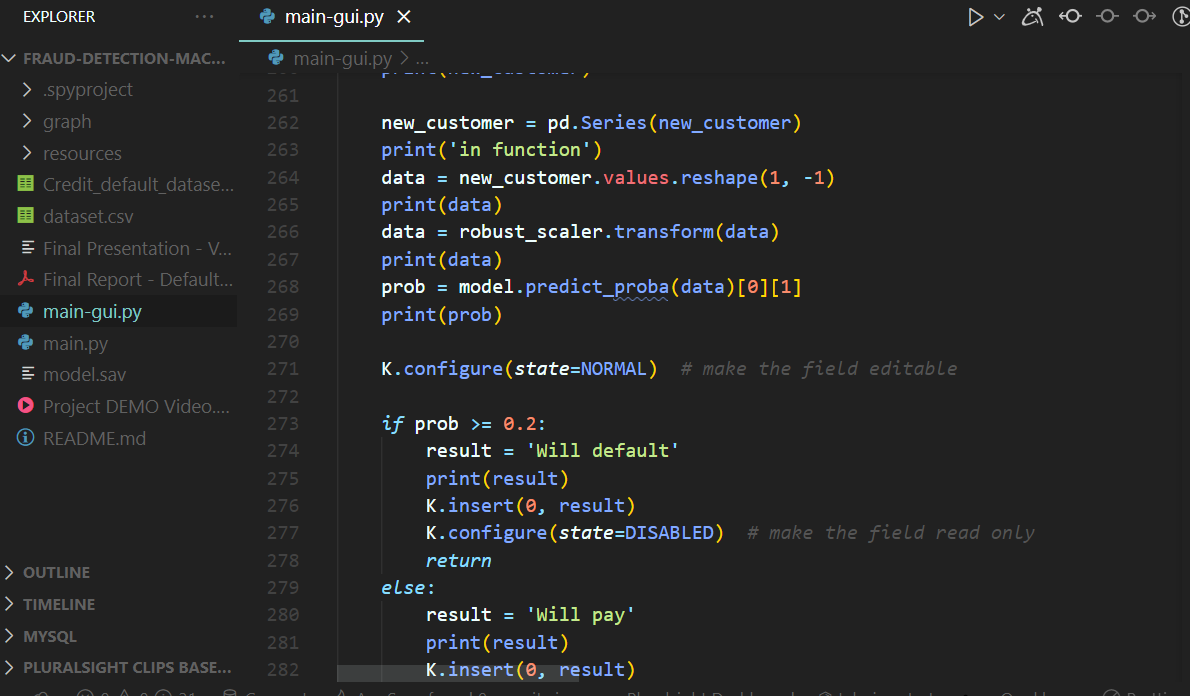
***Figure 14. Python classes and libraries import***



***Figure 15. Labels design with TKinter***



***Figure 15.1 Model implementations***



***Figure 16. Logical computations***

## **4.6 Modification and Improvements**

The model underwent the following modifications and improvements during the process of data collection and system design.

Development of a simple user interface to facilitate easier use by the blind and the visually impaired as covered in the system requirement specification.

Incorporation of audio output for the recognized currency value.

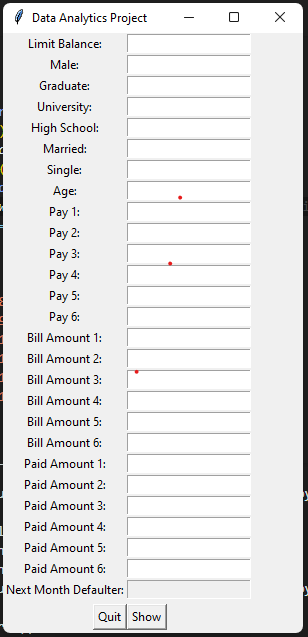
**CHAPTER 5**

**RESULTS AND DISCUSSION**

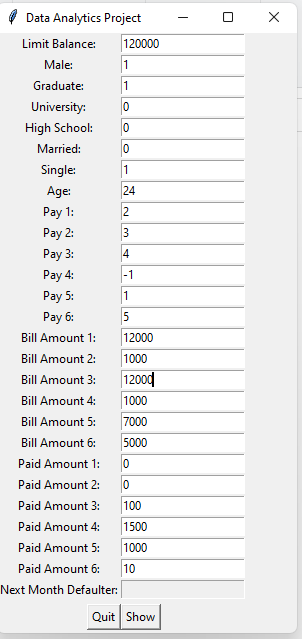
## **5.1 Test Reports**

Below are the test reports for all the test cases that were performed in the and successfully executed;

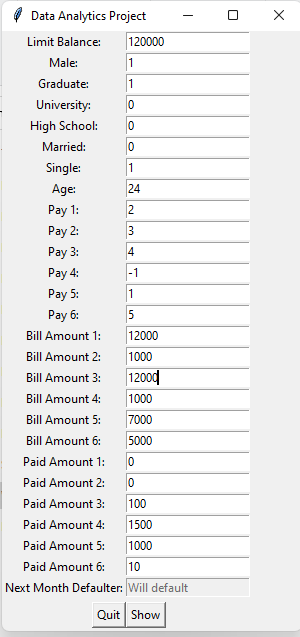
***Figure 17. Input form***



***Figure 18. Filled Input form***



***Figure 19. Filled Input form with test results***



## **5.2 User Documentation**

**Launching the application**

The general steps are mentioned below:

1. Choose the best model and parameters
2. Save to .json file
3. Load a file from disk to predict data screen to get the value.
4. Call a function on button submit and load data to models.
5. Check the probability and the result in the last row in the form.

# **CHAPTER 6**

**CONCLUSION**

## **6.1 Conclusion and Future Works**

This would inform the issuer’s decisions on who to give a credit card to and what credit limit to provide. We investigated the data, checking for data unbalancing, visualizing the features and understanding the relationship between different features. We used both train-validation split and cross-validation to evaluate the model effectiveness to predict the target value, i.e. detecting if a credit card client will default next month. We then investigated five predictive models: We started with Logistic Regression, Naïve bayes, SVM, KNN, Classification Tree and Feed-forward NN and Voting classifier accuracy is almost same. We choose based model Logistic regression based on minimum value of False Negative from confusion metrix.

**Future work includes:**

This project has a very huge potential of growth. It could be an answer to many of the questions troubling money lenders, banks and financial institutions at large. To achieve this there are a number of adjustments that this project needs to incorporate.

* + The Graphical User Interface could be improved so that it looks more intuitive, and user friendly with labels and instructions properly indication what to do at what point
  + The program could also be packaged into a single file that runs smoothly without the headache of having to interact with the code-base during execution, which is likely to scare some users who fear technicalities.
  + The application could also be simplified further so that users from a wide range of professional backgrounds and not necessarily data scientist and programmers could understand and operate the application but rather anyone who needs answers from it.
  + The future development and improvement also heavily relies on the feedback received from the users who have always been at the at the center of its development.

## REFERENCES

[1]. Li, Xiao-Lin, and Yu Zhong. An overview of personal credit scoring: techniques and future work. Journal: International Journal of Intelligence Science ISSN 2163-0283. 2012.

[2]. Yeh, I-C. and C-H. Lien, 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systerm with Applications,36: 2473-2480.

[3]. Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.

[4]. Taiwo Oladipupo Ayodele. (2010) “Types of Machine Learning Algorithms”, New Advances In Machine Learning, Yagang Zhang (Ed.), Intech

[5]. NH Niloy, MAI Navid. Naïve Bayesian Classifier and Classification Trees for the Predictive Accuracy of Probability of Default Credit Card Clients. American Journal of Data Mining and Knowledge Discovery. Vol. 3, No. 1, 2018, pp. 1-12. doi: 10.11648/j.ajdmkd.20180301.11

[6]. Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. Expert Systems with Applications, 36, 3302–3308.

[7]. Davis J., Goadrich M. The Relationship Between PrecisionRecall and ROC Curves.ACM New York, NY, USA 2006. ISBN:1-59593-383-2.

[8]. Christopher M. Fraser(2000), “Neural Networks: A Review from a Statistical Perspective”, Hayward Statistics.

[9]. Shie Mannor, Dori Peleg and Reuven Rubinstein. ICML '05 Proceedings of the 22nd international conference on Machine learning. ACM New York, NY, USA 2005. ISBN:

1-59593-180-5

[10]. Arlot, Sylvain, and Alain Celisse. A survey of crossvalidation procedures for model selection. eprint arXiv:0907.4728. DOI:10.1214/09-SS054.

**Appendices**

### **Appendix 1: Budget**

Table 4: Budget

|  |  |  |  |
| --- | --- | --- | --- |
| **Items** | **Quantity** | **Unit Price (Ksh)** | **Total (Ksh)** |
| **Printing and binding** |  | **3000** | **3000** |
| **Laptop** | **1** | **60,000** | **50,000** |
| **Software** | **2** | **6000** | **12,000** |
| **Internet and Airtime** |  | **5,000** | **10,000** |
| **Travel** | **8** | **1000** | **8,000** |
| **Subsistence** | **10** | **2000** | **20,000** |
| **TOTAL** |  |  | **93,000** |

### **Appendix 2: Schedule**

Table 5: Schedule

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | Jan  2022 | Feb 2022 | March 2022 | Apr 2022 | May 2022 | June 2022 | July 2022 | Aug 2022 |
| Project Title |  |  |  |  |  |  |  |  |
| Introduction |  |  |  |  |  |  |  |  |
| Literature Review |  |  |  |  |  |  |  |  |
| Methodology |  |  |  |  |  |  |  |  |
| Requirement Gathering |  |  |  |  |  |  |  |  |
| Analysis & Design |  |  |  |  |  |  |  |  |
| Development of Prototype |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |
| Conclusion |  |  |  |  |  |  |  |  |