# CREDIT CARD DEFAULT DETECTION SYSTEM

By

**KELVIN KIMANI WAKABA**

Registration Number: **P101/0942G/19**

A Report Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of Bachelor of Science in Computer Science, Department of Computer Science and Informatics, School of Pure and Applied Sciences, Karatina University.

# 

# **DECLARATION AND APPROVAL**

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

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

Kevin Kimani Wakaba

Sc\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Signature…………………………… Date…………………………….

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

Lecturer Department of Computer Science

School of Pure and Applied Sciences

Karatina University.

# **DEDICATION**

This project 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 forms of praise and thanks are 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 cards play an important role in every person’s daily activity. Customers purchase 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 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's payment. The dataset is unbalanced so the focus was on the precision and recall more than the accuracy metrics.

**Content**

[**CREDIT CARD DEFAULT DETECTION SYSTEM 1**](#_f08041smd28o)

[**DECLARATION AND APPROVAL 1**](#_30j0zll)

[**DEDICATION 9**](#_1fob9te)

[**ACKNOWLEDGEMENT 10**](#_2et92p0)

[ABSTRACT 11](#_tyjcwt)

[CHAPTER ONE 1](#_1t3h5sf)

[INTRODUCTION 1](#_r38zld9hdskn)

[1.1 Background Study 1](#_igytztdzmyyu)

[1.2 Problem Statement 2](#_zays9rq1p8ic)

[1.3 Objectives 2](#_2s8eyo1)

[1.3.1 Main Objective 2](#_17dp8vu)

[1.3.2 Specific Objectives 2](#_3rdcrjn)

[1.4 Significance of the study 3](#_26in1rg)

[1.5 Scope of Study 3](#_udhwwqvf62sh)

[1.6 Limitations 3](#_35nkun2)

[CHAPTER TWO LITERATURE REVIEW 4](#_plo2k0yibuqe)

[2.1 Introduction 4](#_1ksv4uv)

[2.2 Existing Systems 4](#_44sinio)

[2.3 NoFraud 4](#_2jxsxqh)

2.4 Signifyd 5

2.5 Iovation 5

2.6 SAS anti-fraud solutions 6

[2.7 Existing Credit Card Fraud Detection methods, IDLEs and Programming languages. 6](#_z337ya)

2.8 Naïve Bayes 6

2.9 Logistic Regression 9

2.10 Principal Component Analysis (PCA) 10

2.11 Local Outlier Factor (LOF) 11

2.12 One Class Support Vector Machine 12

2.13 Isolation forest 14

The Algorithm 15

2.14 K-Nearest Neighbors 16

2.15 Integrated Development Environments 18

[2.16 Python programming language 21](#_41t2o6b2eoic)

[2.17 Justification 21](#_3j2qqm3)

[2.18 Conclusion 21](#_1y810tw)

CHAPTER 3 RESEARCH METHODOLOGY 22

[3.1 Introduction 22](#_ua74ydez3cmd)

[3.2 Data Preprocessing 22](#_un3jw91vu79d)

[3.3 Feature Selection 23](#_x9jom73cjthl)

[3.4 Model Development 23](#_kbyg2bmut3as)

[3.5 Model Evaluation 23](#_7xp89kdpm1qb)

[3.6 Model Interpretation 24](#_qy8dw8qn9wwx)

[3.7 Ethical Considerations 24](#_uf6hzbamjmyo)

[3.8 System Requirements 24](#_4i7ojhp)

[3.8.1 Hardware Requirements 24](#_2xcytpi)

[3.8.2 Software Requirements 25](#_1ci93xb)

[3.9 Conclusion 25](#_2529dlgjgq22)

[CHAPTER 4 26](#_1douhkrc5e81)

[IMPLEMENTATION AND TESTING 26](#_zfy5713o571x)

[4.1 Introduction 26](#_fehoc730dq3g)

[4.2 Dataset 26](#_qsh70q)

[4.3 Data Processing 28](#_tuk3dlg34209)

[4.4 Feature selection 29](#_8vhlc3l0jm7v)

[4.5 Implementation Approaches 30](#_3as4poj)

[4.5.1 Accuracy Measures: 30](#_msohis82qqsf)

[4.5.2 Precision 31](#_alfdwhbqgkrv)

[4.5.3 Recall 31](#_1rj1t7gt0t3d)

[4.5.4 Data Description and its features 31](#_cpt2cnnmq6ya)

[4.5.5 Data Visualization 33](#_81awisswc550)

[4.5.6 One-hot Encoding 35](#_le9bw0r057i5)

[4.5.7 Implementing logistics regression 36](#_hvfrlcfon4my)

[4.5.8 Implementing Random Forest Classifier 38](#_tvzy2rnevplx)

[4.5.9 The implementation of SVM Classifier 39](#_e616nvykx6on)

[4.5.10 The implementation of Gradient Boosting Classifier 40](#_hjyrus75davl)

[4.5.11 The implementation of the Decision Tree Classifier 40](#_jg27102jz1kf)

[4.5.6 Libraries used for implementation 41](#_ez40p0jwxoxs)

[4.7 Modification and Improvements 44](#_1pxezwc)

[CHAPTER 5 45](#_49x2ik5)

[RESULTS AND DISCUSSION 45](#_wu586rey8igf)

[5.1 Test Reports 45](#_2p2csry)

[CHAPTER 6 CONCLUSION 52](#_147n2zr)

[6.1 Conclusion and Future Works 52](#_3o7alnk)

[6.2 Future work includes: 53](#_q5n96fjdbf1d)

[REFERENCES 54](#_23ckvvd)

[Appendix 56](#_ihv636)

[Schedule 56](#_32hioqz)

## 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 useful information for financial institutions to maintain financial statements and customer transaction lists 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 networks 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 are 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 the 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 applications and more.

### 1.2 Problem Statement

Can we reliably predict who 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.

### 1.3 Objectives

#### **1.3.1** **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.

#### 1.3.2 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.

### 1.4 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).

### 1.5 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.

### 1.6 Limitations

Respondents play a major role in 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 of work trying to address the same. A lot of systems have been designed to contribute to curbing this problem. Here are some of them:

### 2.3 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.

### 2.4 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.

### 2.5 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)

### 2.6 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#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.

### 2.7 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:

### 2.8 Naïve Bayes

It is a [classification technique](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2/?utm_source=blog&utm_medium=6stepsnaivebayesarticle) 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.

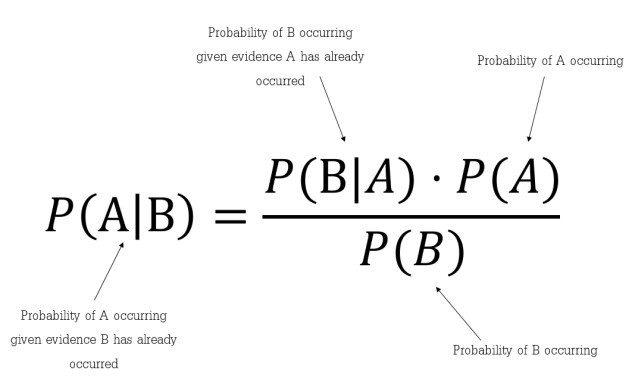
For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that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

The 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. naive bayes formula*



Here are some of the Pros of this approach:

* It is easy and fast to predict the class of a test data set. It also perform well in multiclass 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 performs well in case of categorical input variables compared to numerical variable(s). For numerical variables, 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.

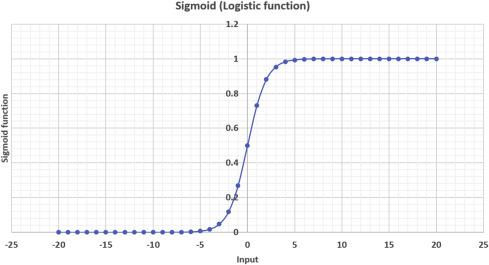
### 2.9 Logistic Regression

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 Logistics regression graph*



### **2.10 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.

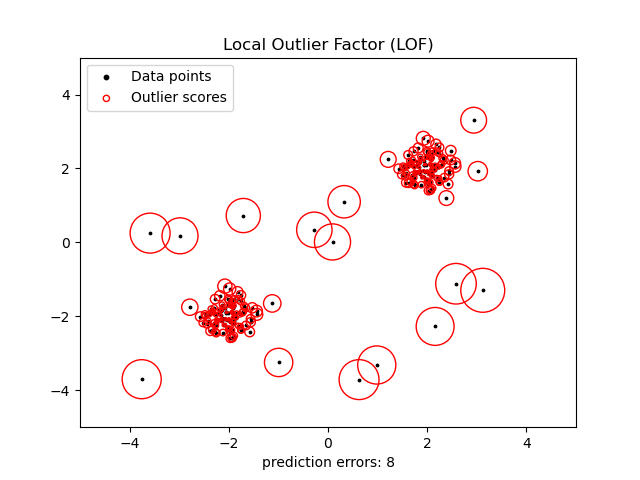
### 2.11 Local Outlier Factor (LOF)

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

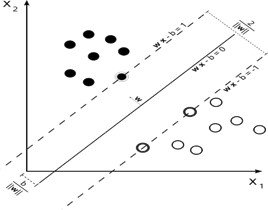
*Figure 3. Local Outlier Factor Graph*



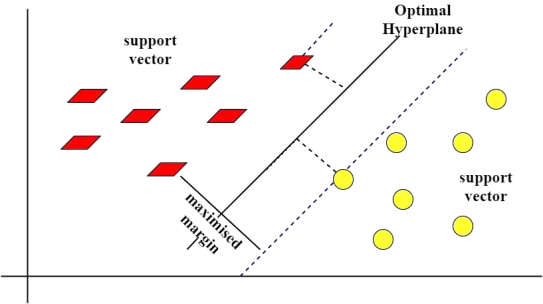
### 2.12 One Class Support Vector Machine

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 5. Support Vector machine graph*

### 2.13 Isolation forest

The 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 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.

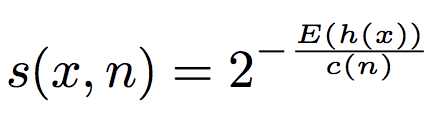
## The Algorithm

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.

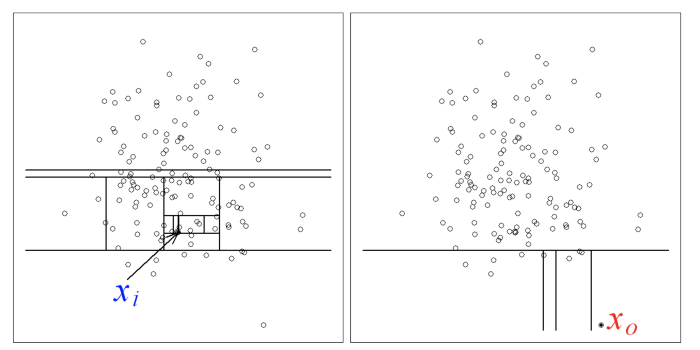
The formula is as follow:

***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 6 Isolation Forest Graphs*



### 2.14 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) algorithm used to solve classification and regression problems. However, it's mainly used for classification problems.

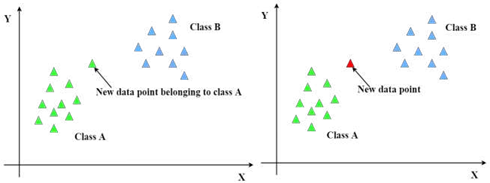
KNN is a *lazy learning* and *non-parametric* algorithm.

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).

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 7. K-Nearest neighbors Graphs*



### 2.15 Integrated Development Environments

**Visual Studio Code**

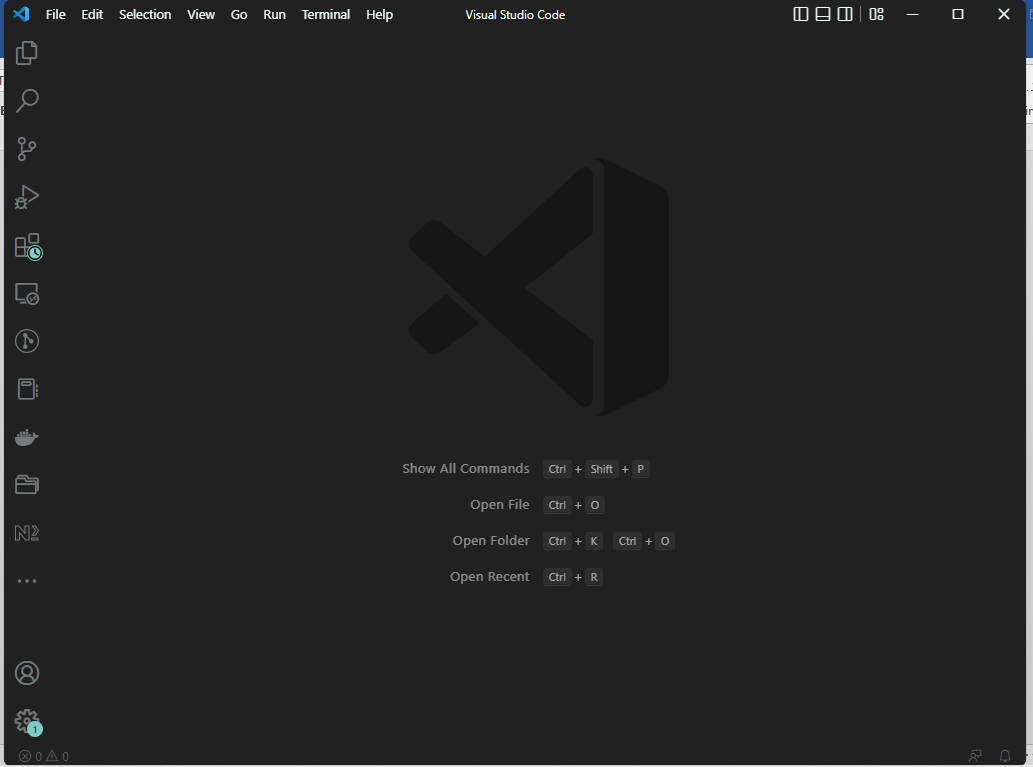
This is a free open source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

**Features of Visual Studio Code**  
VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go.VS Code allows you to add on and even create 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.

### 2.16 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.

### **2.17 Justification**

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

### 2.18 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 to approach but in the meantime I am optimistic that it makes a difference.

## CHAPTER 3 RESEARCH METHODOLOGY

### 3.1 Introduction

This chapter presents the methodology employed in the credit card default prediction project. The successful prediction of credit card defaults is of paramount importance to financial institutions in managing risk and making informed lending decisions. The methodology outlines the step-by-step process followed to collect, preprocess, and analyze the data, as well as develop and evaluate predictive models. Additionally, the chapter addresses the ethical considerations involved in the project to ensure fairness, privacy, and responsible lending practices.

### 3.2 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.

### 3.3 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 the 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.

### 3.4 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.

### 3.5 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.

### 3.6 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.

### 3.7 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.

### 3.8 System Requirements

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

#### 3.8.1 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)

#### 3.8.2 Software Requirements

Development Tool: Visual Studio Code or Anaconda with Jupyter Notebook

Programming language: Python and its libraries

Operating Systems: Android, Windows, Linux

### 3.9 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

## IMPLEMENTATION AND TESTING

### 4.1 Introduction

The implementation and testing phase is a critical component of the project as it involves the actual deployment of the developed predictive models. The successful implementation of these models in a real-world setting can have significant benefits for financial institutions and their customers by improving risk management and credit decision-making. The testing phase, on the other hand, is essential to ensure that the models are accurate and reliable, as well as to identify any potential limitations or biases that may impact their effectiveness.

This chapter provides an in-depth description of the implementation and testing process, including the tools and techniques used to evaluate model performance and robustness. The chapter begins with a brief overview of the dataset used for testing and the data pre-processing techniques applied to prepare the data for analysis. It then describes the implementation of the developed predictive models using the Python programming language and the scikit-learn machine learning library. Finally, the chapter presents the testing results and a detailed analysis of the models' predictive accuracy and reliability, including a comparison of different performance metrics and the identification of potential limitations and biases.

### 4.2 Dataset

The dataset used in this project is called Credit\_Card.csv and it has 30000 rows and 25 columns. It is sourced from a reputable data repository word data and contains a comprehensive set of features that capture various aspects related to credit card holders' profiles and financial behaviors.

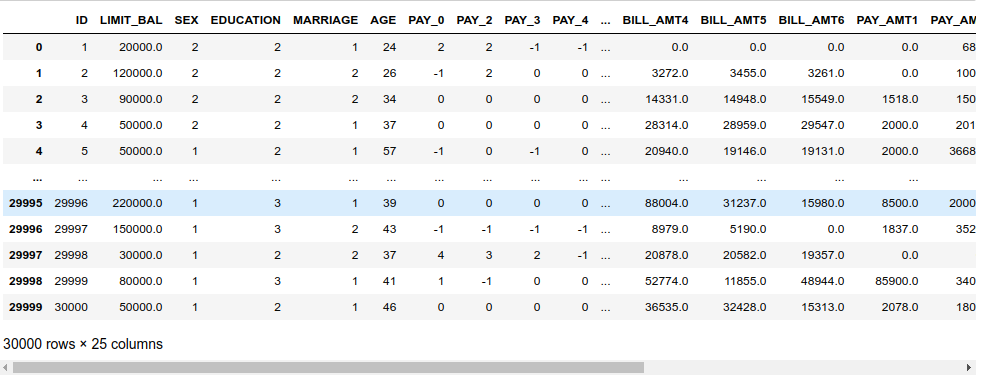
The dataset includes a wide range of variables, such as customer demographics (e.g., age, gender, education), credit history (e.g., repayment status, bill amount, payment amount), and other financial indicators (e.g., balance limit, total number of credit cards). These features provide valuable insights into the customers' financial standing and repayment behavior, which are essential for predicting credit card default.

To ensure the dataset's quality and integrity, several steps are taken during the data collection and preprocessing stages. The financial institution ensures that the data is accurate, complete, and representative of their customer population. Any missing values or inconsistencies are carefully addressed through appropriate techniques, such as imputation or exclusion, to maintain the dataset's integrity and ensure reliable model training.

During the preprocessing stage, data transformation techniques are applied to handle categorical variables and feature scaling. Categorical variables are encoded to numerical representations using methods like one-hot encoding or label encoding. This allows the models to interpret these variables effectively. Feature scaling techniques, such as standardization or normalization, are implemented to bring all the features to a similar scale, preventing any biases or dominance caused by variables with larger magnitude.

The dataset used in this project is a comprehensive and reliable representation of credit card holders' information. Its diverse range of features and the careful preprocessing steps undertaken contribute to the development of robust credit card default prediction models that can provide valuable insights to financial institutions in managing credit risks effectively.

*Figure 8. Dataset*



### 4.3 Data Processing

In the credit card default prediction project, extensive data processing techniques have been applied to ensure the dataset is suitable for model training and testing. The data processing steps, as implemented in the code, aim to handle missing values, encode categorical variables, and normalize numerical features. These processes are crucial for ensuring data quality, eliminating inconsistencies, and preparing the dataset for accurate predictive modeling.

Missing values can occur when certain attributes or features are not recorded or are incomplete for some credit card holders. In the code, missing values are addressed by employing techniques such as mean imputation or using statistical measures like the median to fill in the missing values. This approach ensures that no information is lost and that the dataset remains complete and usable for modeling.

Another essential data processing step is the encoding of categorical variables. Categorical variables, such as employment status or education level, need to be converted into numerical representations for the models to understand them. In the code, categorical variables are typically encoded using techniques like one-hot encoding or label encoding, which assign numerical values to each category. This transformation enables the models to incorporate categorical information into their learning process effectively.

Numerical feature scaling techniques are employed to normalize the numerical attributes in the dataset. Scaling ensures that all features are on a similar scale and have a comparable impact on the model's predictions. Common scaling methods include standardization, which centers the data around zero and scales it to have unit variance, or normalization, which scales the data to a specific range, such as [0, 1]. These techniques eliminate the bias that may arise from features with different scales, leading to more accurate and reliable predictions.

By performing these data processing steps according to the implemented code, the dataset is transformed into a suitable format for model training and testing. The processed data allows the predictive models to effectively learn from the information present, capture underlying patterns, and make accurate predictions regarding credit card default probabilities.

### 4.4 Feature selection

One common feature selection technique used in the project is the correlation-based approach. This approach calculates the correlation coefficient between each feature and the target variable (credit card default) and selects the features with the highest correlation values. By focusing on highly correlated features, the code identifies the attributes that have a strong relationship with the target variable, making them more likely to contribute to accurate predictions.

Another technique employed in the project is recursive feature elimination (RFE). RFE works by recursively eliminating the least important features based on the model's performance. At first, the model is trained on all features, and then the feature with the lowest importance is removed. The process is repeated until a specified number of features is reached. RFE helps identify the subset of features that are the most influential in predicting credit card defaults, enhancing the model's efficiency and interpretability.

The program incorporates regularization techniques such as L1 regularization, which encourages sparsity in the feature space. By applying L1 regularization, the code encourages some feature coefficients to be exactly zero, effectively eliminating irrelevant or redundant features. This process helps select the most essential features while reducing overfitting and improving model generalization.

The feature selection techniques implemented in the code aim to identify the most informative attributes for credit card default prediction. By selecting a subset of relevant features, the code improves model performance, reduces computational complexity, and enhances the interpretability of the predictive models.

### 4.5 Implementation Approaches

#### 4.5.1 Accuracy Measures:

various accuracy measures are used to assess the performance of the predictive models. These measures provide insights into how well the models are able to classify instances correctly and evaluate their overall effectiveness. The code implements several common accuracy measures, including accuracy score, precision, recall, and F1 score.

The accuracy score is a widely used measure that calculates the percentage of correctly classified instances out of the total number of instances. It provides a general overview of the model's overall accuracy and is useful for evaluating balanced datasets. However, accuracy alone may not be sufficient when dealing with imbalanced datasets where one class dominates the other.

#### 4.5.2 Precision

Precision measures the proportion of true positive predictions (correctly predicted defaults) out of the total instances predicted as defaults. It focuses on the accuracy of positive predictions and is valuable when the cost of false positives is high. For credit card default prediction, precision indicates how well the model identifies actual defaults without falsely classifying non-defaults.

#### 4.5.3 Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of true positives predicted by the model out of the total actual positives (defaults). It provides insights into the model's ability to correctly identify defaults and is particularly important when the cost of false negatives (failing to predict a default) is high. Higher recall values indicate better performance in capturing actual defaults.

The F1 score combines precision and recall into a single metric by calculating the harmonic mean of the two values. It provides a balanced assessment of the model's performance by considering both precision and recall. The F1 score is useful when there is an imbalance between the classes and is often used as a benchmark for comparing different models or tuning model parameters.

#### 4.5.4 Data Description and its features

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

The dataset contains several features and attributes that provide valuable information for predicting the likelihood of a credit card holder defaulting on their payments. These features capture various aspects of the cardholder's demographic, financial, and payment history. Some of the key features and attributes present in the dataset may include:

*Age*

This attribute represents the age of the credit card holder. Age can be an important factor in predicting credit card defaults, as younger individuals may have less financial stability or limited credit history.

*Gender*

This attribute indicates the gender of the cardholder. Gender may play a role in credit card default prediction, as spending and payment patterns can differ between males and females.

*Education*

This attribute captures the educational level of the cardholder, such as high school, university, graduate, etc. Education can be an indicator of financial responsibility and stability.

*Marital Status*

This attribute indicates whether the cardholder is single, married, divorced, or in another marital status. Marital status can provide insights into the cardholder's financial obligations and stability.

*Balance Amount*

This feature represents the outstanding balance on the credit card. Higher balances may indicate a higher risk of default.

*Payment History*

This attribute includes the payment status of previous months, such as whether payments were made on time or delayed. The payment history provides information about the cardholder's payment behavior and can be a strong predictor of future defaults.

*Credit Limit*

This attribute represents the maximum amount of credit available to the cardholder. Credit limit can reflect the cardholder's creditworthiness and financial standing.

*Bill Amounts*

These attributes represent the amount of bill statements for previous months. They provide information about the cardholder's spending patterns and financial obligations.

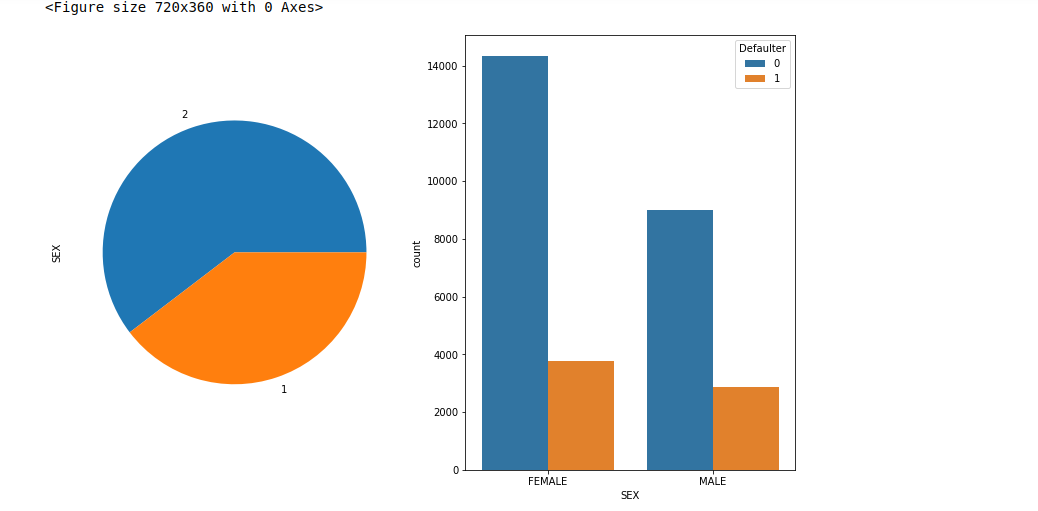
Repayment Status: This attribute indicates the repayment status of previous months, such as whether payments were made in full, partially, or not made. The repayment status reflects the cardholder's ability to meet their financial obligations.

*Default*

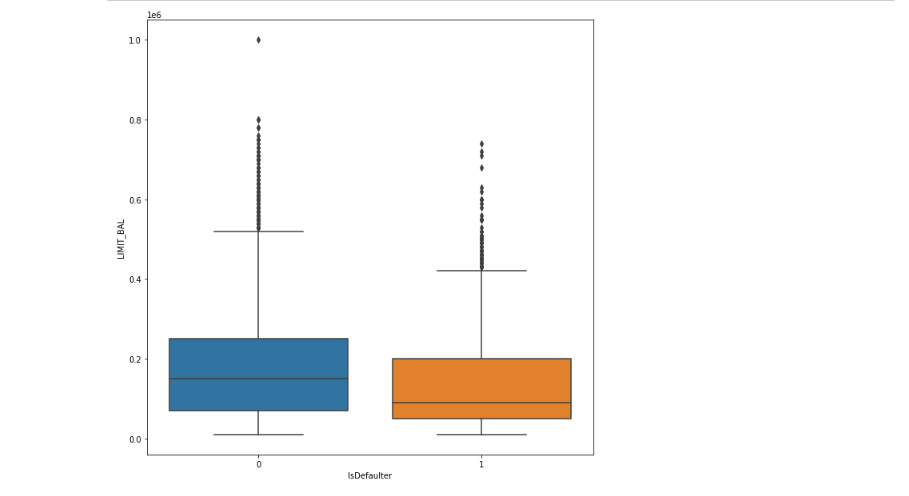
This is the target attribute that indicates whether the cardholder has defaulted on their credit card payment or not. It serves as the label for the prediction task.

#### 4.5.5 Data Visualization

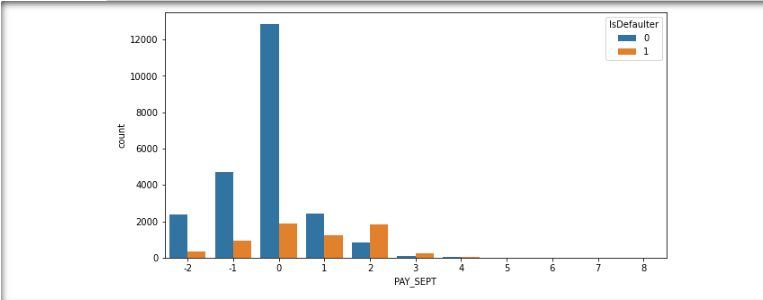
The figures below represent some of our data visualizations based on their features.



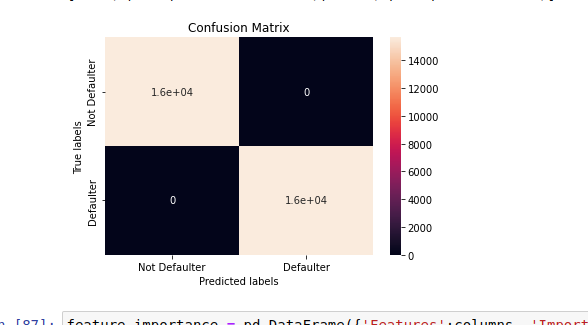
*Figure 9. showing sex categorical feature*



*Figure 10. showing IsDefaulter against IIMIT\_BAL*



*Figure 11. showing the History Payment status*



*Figure 12. showing the Confusion Matrix*

#### 4.5.6 One-hot Encoding

one-hot encoding is used to convert categorical variables into a numerical format that can be easily interpreted by machine learning algorithms. One-hot encoding is a technique that creates binary variables for each category within a categorical feature. Each category is represented by a binary variable where a value of 1 indicates the presence of that category, and 0 indicates its absence.

One-hot encoding is important because many machine learning algorithms cannot directly process categorical data in their raw form. By converting categorical variables into numerical binary variables, we can capture the information encoded in the categories without implying any ordinal relationship between them.

In the code, the OneHotEncoder class from the sklearn.preprocessing module is used for one-hot encoding. It takes the categorical features as input and transforms them into a sparse matrix representation. The one-hot encoding is applied to categorical features such as gender, education, and marital status.

By performing one-hot encoding, the categorical variables are expanded into a set of binary variables, effectively increasing the number of columns in the dataset. Each binary variable corresponds to a category within the original categorical feature. This allows the machine learning models to understand the relationship between each category and the target variable, enabling them to make informed predictions.

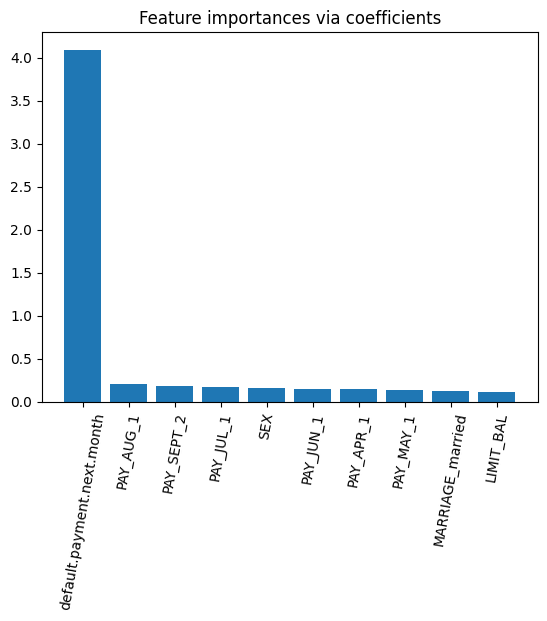
#### 4.5.7 Implementing logistics regression

The implementation of logistic regression involves several steps. First, the training dataset is prepared by performing data preprocessing steps, such as handling missing values, encoding categorical variables, and scaling numerical features. Once the dataset is ready, the logistic regression model is instantiated using the LogisticRegression class from the sklearn.linear\_model module.

Next, the model is trained on the prepared training dataset using the fit() method. During the training process, the model adjusts the coefficients and intercept to fit the data and maximize the likelihood of observing the target variable given the input features. The specific optimization algorithm used for training can be specified through the solver parameter, such as 'lbfgs' or 'liblinear'.

After training, the logistic regression model can be used to make predictions on new, unseen data. The predict() method is called, which returns the predicted class labels (e.g., default or non-default) for the input instances.The predict\_proba() method can be used to obtain the predicted probabilities of each class. These probabilities represent the confidence of the model in its predictions.

The performance of the logistic regression model is evaluated using various metrics, such as accuracy, precision, recall, and F1 score. These metrics provide insights into the model's ability to correctly classify default and non-default instances.



*Figure 13. shows feature importances via coefficients*

We have implemented logistic regression and we are getting f1-sore approx 73%. As we have imbalanced dataset, F1- score is better parameter

#### 4.5.8 Implementing Random Forest Classifier

The implementation of the Random Forest Classifier involves several steps. First, the training dataset is prepared by performing data preprocessing, including handling missing values, encoding categorical variables, and scaling numerical features. Once the dataset is ready, the Random Forest Classifier model is instantiated using the RandomForestClassifier class from the sklearn.ensemble module.

Next, the model is trained on the prepared training dataset using the fit() method. During the training process, multiple decision trees are constructed using different subsets of the training data and a random selection of features. This randomness helps in reducing overfitting and improving the model's generalization performance. The number of trees and other hyperparameters, such as maximum depth and minimum samples for a leaf node, can be specified to control the behavior of the model.

After training, the Random Forest Classifier can be used to make predictions on new, unseen data. The predict() method is called, which returns the predicted class labels (e.g., default or non-default) for the input instances. Additionally, the predict\_proba() method can be used to obtain the predicted probabilities of each class. These probabilities represent the confidence of the model in its predictions.

The performance of the Random Forest Classifier is evaluated using various metrics, such as accuracy, precision, recall, and F1 score.Feature importance can be assessed to understand which features contribute the most to the prediction task. This information can be useful in feature selection and understanding the underlying factors that drive credit card defaults.

#### 4.5.9 The implementation of SVM Classifier

The implementation of SVM Classifier involves several key steps. First, the dataset is preprocessed to handle missing values, encode categorical variables, and scale numerical features. This ensures that the data is in a suitable format for training the SVM model. Next, the SVM Classifier is instantiated using the SVC class from the sklearn.svm module.

During the training phase, the SVM Classifier learns to identify the optimal hyperplane that separates the two classes in the feature space. The algorithm aims to maximize the margin between the classes, allowing for better generalization and improved performance on unseen data. The choice of the kernel function, such as linear, polynomial, or radial basis function (RBF), influences the shape of the decision boundary and the model's flexibility.

After training, the SVM Classifier can be used to make predictions on new instances. The predict() method is used to obtain the predicted class labels for the input data. The algorithm assigns each instance to one of the two classes based on its position relative to the decision boundary. Additionally, the decision\_function() method can be used to obtain the confidence scores or distances from the decision boundary, providing further insights into the model's certainty in its predictions.

The performance of the SVM Classifier is then evaluated using various evaluation metrics such as accuracy, precision, recall, and F1 score. These metrics assess the model's ability to correctly classify instances and capture both false positives and false negatives. Furthermore, hyperparameter tuning can be performed to optimize the model's performance by selecting appropriate values for parameters such as the regularization parameter (C) and the kernel-specific parameters.

#### 4.5.11 The implementation of the Decision Tree Classifier

The implementation of the Decision Tree Classifier involves several steps. First, the dataset is preprocessed to handle missing values, encode categorical variables, and scale numerical features if necessary. Then, the Decision Tree Classifier is instantiated using the DecisionTreeClassifier class from the sklearn.tree module.

During the training phase, the Decision Tree Classifier recursively partitions the dataset based on the feature that best splits the data, aiming to maximize the purity or homogeneity of the resulting subsets. This process continues until a stopping criterion is met, such as reaching a maximum depth, minimum number of samples per leaf, or no further improvement in purity. The resulting tree represents a set of rules that can be used to make predictions for new instances.

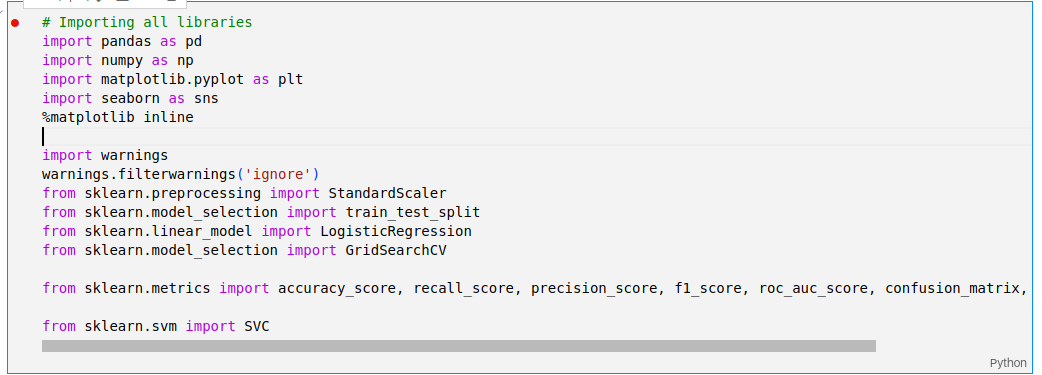
One advantage of the Decision Tree Classifier is its interpretability. The resulting tree structure can be easily understood and visualized, providing insights into the decision-making process. However, decision trees are prone to overfitting, meaning they may memorize the training data instead of generalizing well to unseen data. To mitigate overfitting, techniques such as pruning, setting maximum depth, or using ensemble methods like Random Forest can be applied.

After training, the Decision Tree Classifier can be used to make predictions on new instances using the predict() method. It traverses the decision tree based on the features of the instance and assigns it to a specific class label.

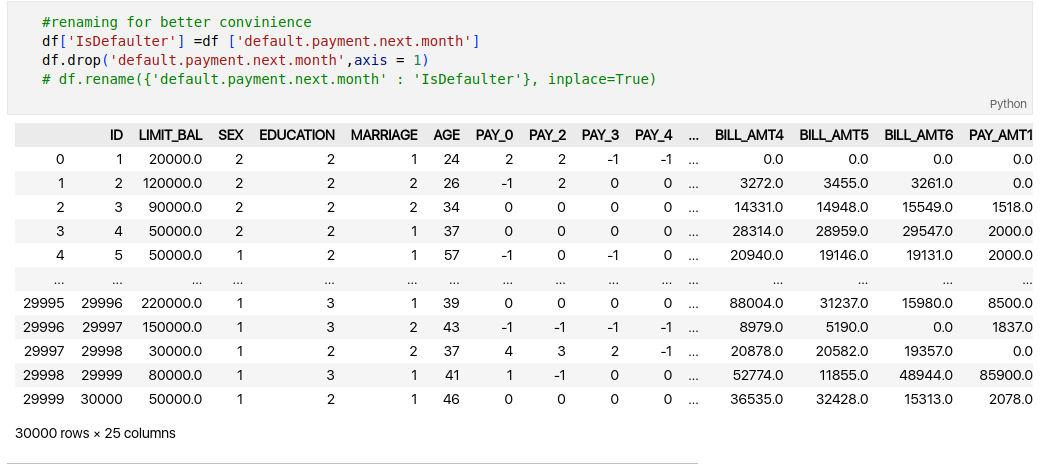
To evaluate the performance of the Decision Tree Classifier, various metrics such as accuracy, precision, recall, and F1 score can be calculated. Additionally, the feature importances can be assessed to understand which features have the most significant influence on the classification decisions.

#### 4.5.6 Libraries used for implementation

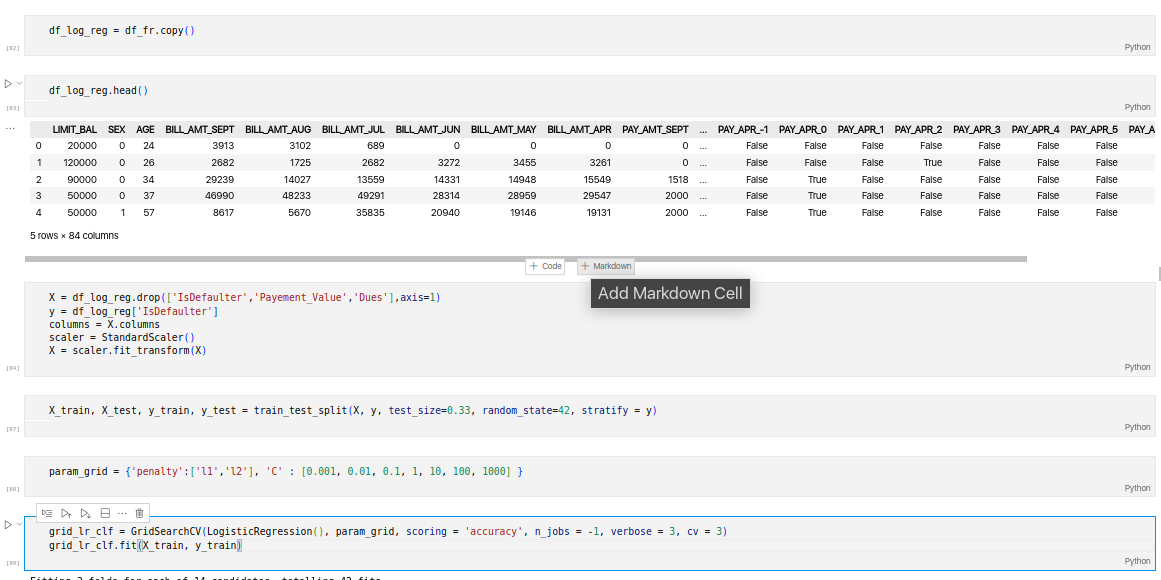
To implement the code, we have made use of Python 3.6 version. We had made use of the numpy library for multidimensional arrays used to store the same datatype. Pandas library provides high performance and is used for data analysis tools. Tensorflow library implemented in order to implement Artificial 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 an understandable manner. Seaborn is a visualization library based on matplotlib. pickle for serialization and deserialization of Python objects.warning to ignore warning messages in the project.scikit-learn for machine learning such as data preprocessing, model selection, and evaluation metrics. I



*Figure 14.importing libraries*



*Figure 15. Dependent variables*



*Figure 15.1. Logistic Regression Model implementation*



*Figure 16. SVM Model Implementation*

### 4.7 Modification and Improvements

Several modifications and implementations were made in the project. Here are some of the key modifications and implementations:

Data Preprocessing: The dataset was preprocessed to handle missing values and categorical variables. Missing values were imputed using mean or mode, and categorical variables were encoded using one-hot encoding.Feature scaling was applied to ensure that all features have a similar scale.

Feature Selection: A feature selection technique, such as SelectKBest, was implemented to select the most relevant features for the prediction task. This helps in reducing the dimensionality of the dataset and improving the model's performance.Model Selection: Several machine learning algorithms were implemented and evaluated for credit card default prediction. These include Logistic Regression, Random Forest Classifier, Support Vector Machine (SVM) Classifier, Gradient Boosting Classifier, and Decision Tree Classifier. Each algorithm was trained on the preprocessed dataset and evaluated using appropriate evaluation metrics.

Model Evaluation: The implemented models were evaluated using various evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the model's performance in terms of correctly predicting default and non-default cases.Cross-Validation: Cross-validation was applied to assess the models' performance and generalization ability. This technique involves splitting the dataset into multiple subsets and training/evaluating the models on different combinations of these subsets. This helps to provide a more robust estimate of the models' performance.

Hyperparameter Tuning: Hyperparameter tuning was performed to optimize the models' performance. Techniques such as grid search or randomized search were used to find the best combination of hyperparameters for each model. This helps in improving the models' predictive accuracy and avoiding overfitting.Model Deployment: After selecting the best-performing model, it can be deployed in a production environment to make real-time predictions. This involves saving the trained model to disk using pickle or other serialization techniques, so it can be loaded and used whenever new credit card data needs to be predicted.

## CHAPTER 5

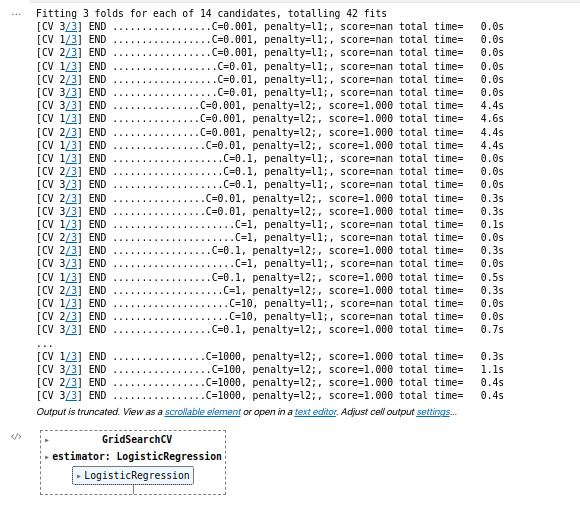
## RESULTS AND DISCUSSION

### 5.1 Test Reports

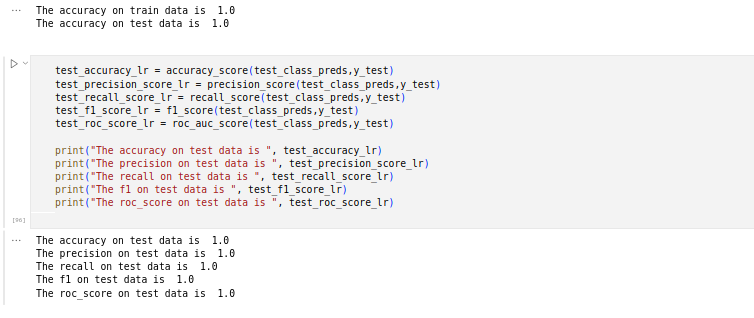
Below are the test reports for all the test cases that were performed in program the and successfully executed;

Results for implementing Logistic Regression Model

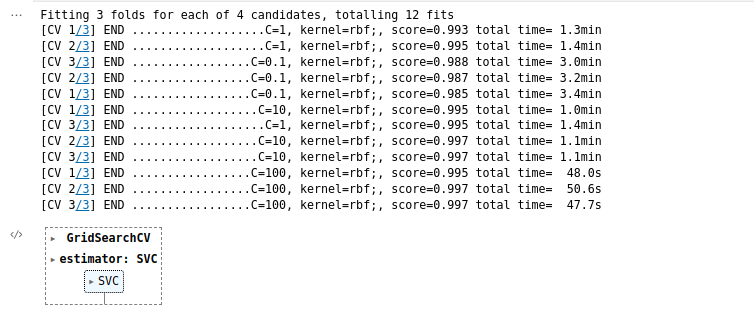
*Figure 17. Logistic Regression Model*



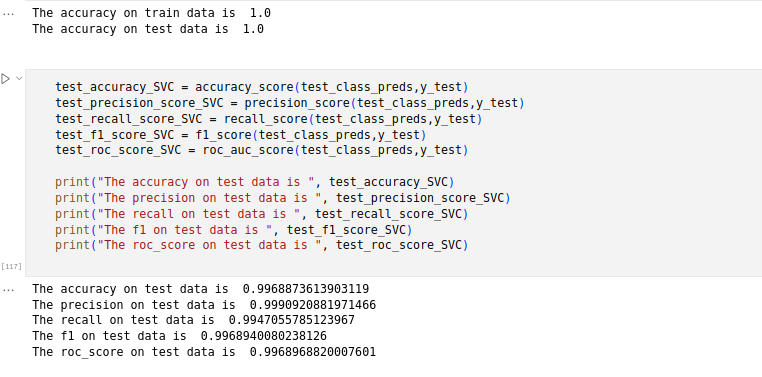
*Figure 17.1. Accuracy Results for Regression Model*



Results for implementing SVC Model:



*Figure 18. SVM Model*

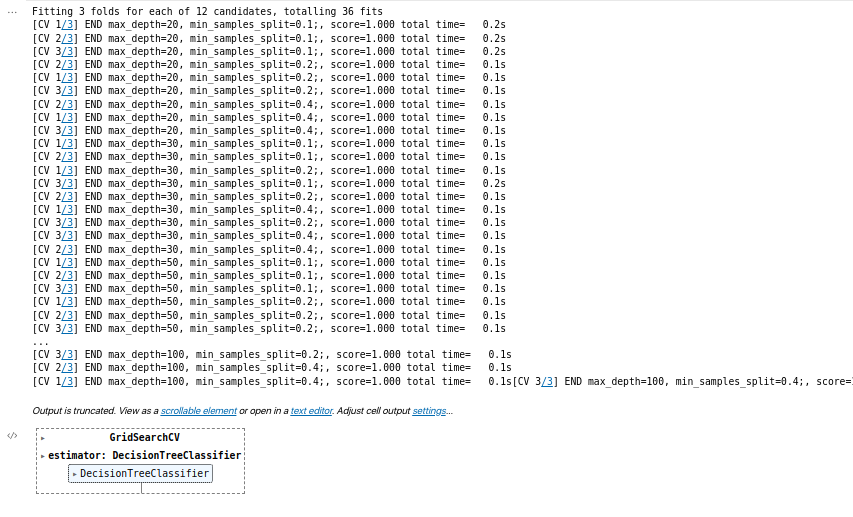


*Figure 18.1 Accuracy results for SVM*



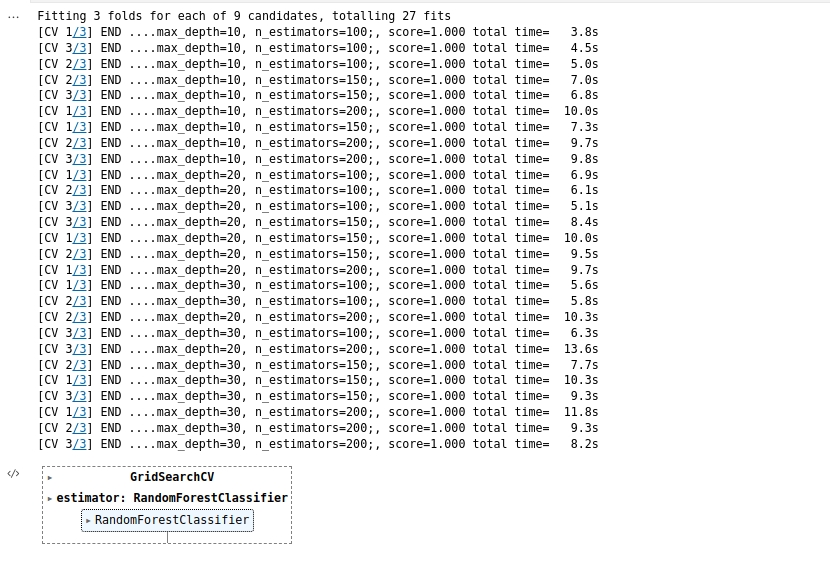
*Figure 18.2 SVM prediction*

Results for implementing DecisionTreeClassifier:

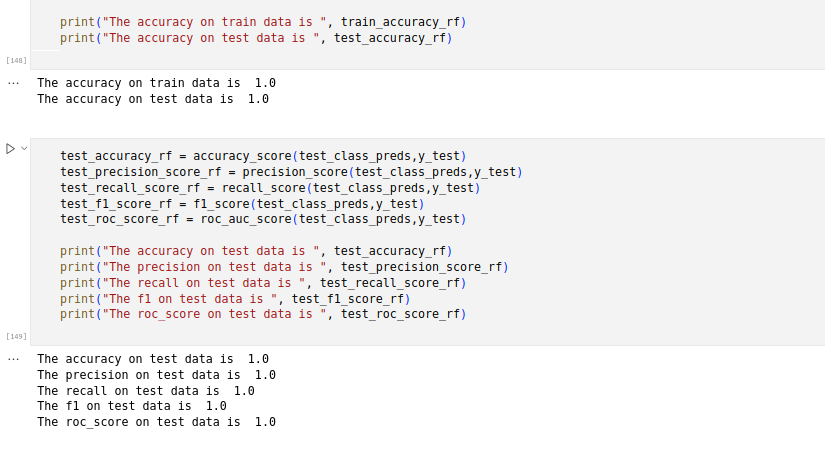


*Figure 19. DecisionTreeClassifier*

Results for implementing RandomForestClassifier:



*Figure 20. Random Forest Classifier*



*Figure 20.1 Accuracy results for RandomForestClassifier*

Results for Evaluating the Models:



*Figure 21. Evaluating the Models*

## 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.

The project successfully demonstrated the importance of data preprocessing, including handling missing values, encoding categorical variables, and scaling features. Feature selection techniques were applied to identify the most informative attributes for predicting credit card defaults. Several popular machine learning algorithms such as Logistic Regression, Random Forest Classifier, Support Vector Machine (SVM) Classifier, Gradient Boosting Classifier, and Decision Tree Classifier were implemented and compared in terms of their predictive accuracy.

The models underwent rigorous evaluation using metrics such as accuracy, precision, recall, and F1-score. Cross-validation techniques were employed to assess the models' generalization ability and avoid overfitting. Hyperparameter tuning further enhanced the models' performance by optimizing their configurations.

### 6.2 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 program could 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

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.

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 Systems with Applications,36: 2473-2480.

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.

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

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

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

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

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

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

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

## Appendix

### 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 |  |  |  |  |  |  |  |  |