VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"PROFILING BASED CREDIT LIMIT APPROVAL SYSTEM USING MACHINE LEARNING"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

AKANKSHA MISHRA (1CR16CS012)

PAVAN G REDDY (1CR16CS110)

PAVAN KUMAR (1CR16CS111)

PAVAN TEJA B DIWAKAR BABU (1CR16CS113)

Under the Guidance of,

DR. PREM KUMAR RAMESH

Professor and Head, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"Profiling based Credit Limit Approval System using Machine Learning"** carried out by **Ms**. **Akanksha Mishra**, USN **1CR16CS012**, **Mr. Pavan G Reddy**, USN **1CR16CS110**, **Mr**. **Pavan Kumar**, USN **1CR16CS111, Mr**. **Pavan Teja B Diwakar Babu**, USN **1CR16CS113,** bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Dr. Prem Kumar Ramesh	**Dr. Sanjay Jain**
Professor & Head	**Principal**
Dept. of CSE, CMRIT	**CMRIT**

External Viva

Name of the examiners	Signature with date
1.	________________
2.	________________

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "**Profiling based Credit Limit Approval System using Machine Learning**" has been successfully completed under the guidance of Dr. Prem Kumar Ramesh, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: CMR Institute of Technology, Bengaluru

Date: 05th June, 2020

Team members:

Akanksha Mishra (1CR16CS012) ____________________

Pavan G Reddy (1CR16CS110) ____________________

Pavan Kumar (1CR16CS111) ____________________

Pavan Teja B Diwakar Babu (1CR16CS113) ____________________

ABSTRACT

Machine Learning has become an indispensable tool in decision support system and finds use cases across various industries. The banking sector is no exception to this. Banks generate a large amount of data on a daily basis. If this data is stored, processed and analyzed well, it can be used to extract meaningful knowledge, which can help increase business profits. Banks play an important role in the economy. Presently banks are facing challenges such as customer retention, fraud detection, risk management and profiling. In this project, a system has been proposed to improve the profiling of bank customer's behavior to identify a potential customer. The profile of a customer serves as a basis to request for a new credit limit, which may or maybe be approved. This is achieved through the use of two different Convolution Neural Networks to perform each of these tasks. The model for predicting the profile of customers has an accuracy of 99.37%, and the model to new credit limit approval attains an accuracy of 79.43%. These models are trained and deployed as a web application using the Flask framework. This project gives a robust solution for profiling bank customers and approving the credit limit based on profiling.

ACKNOWLEDGEMENT

The satisfaction of successfully completing a task feels incomplete without mentioning and thanking those people who helped and aided us in our journey. Their guidance and support gave direction to our hard work and efforts which was important in the completion of this project.

We take this opportunity to express our sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing us a platform to pursue our studies and carry out our final year project.

We have a great pleasure in expressing our deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

It is a privilege and honor to express our eternal thanks to our project guide, **Dr. Prem Kumar Ramesh**, Professor and HOD, Department of Computer Science and Engineering, CMRT, Bangalore, for providing us with his invaluable experience, support and guidance throughout the tenure of this project, which was pivotal in the successful completion of our project.

We also extend our thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged us.

Finally, we would like to thank my parents and friends for all their moral support they have given us during the completion of this work.

Akanksha Mishra (1CR16CS012)

Pavan G Reddy (1CR16CS110)

Pavan Kumar (1CR16CS111)

Pavan Teja B Diwakar Babu (1CR16CS113)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Relevance of the Project

The evolution of science and technology has influenced progress in various industries including the banking industry, which forms the life and blood of any economy. The modern banking industry is a network of financial institutions licensed by the state to supply banking services. The principal services offered relate to storing, transferring, extending a credit against, or managing the risks associated with holding various forms of wealth. The precise bundle of financial services offered at any given time has varied considerably across institutions, across time, and across jurisdictions, evolving in step with changes in the regulation of the industry, the development of the economy, and advances in information and communications technologies.

Banks deal with a large amount of data which can be used to extract useful knowledge that can impact business profits. For instance, banks can increase their profitability by focusing on their valuable customers. Profiling a bank customer's behaviour by taking relevant data into account will help the bank to determine what credit limit should be provided. The task is efficiently performed using machine learning techniques, which removes manual intervention without compromising on accuracy. Performing the profiling using Machine Learning algorithms helps to remove any manual intervention without compromising on the accuracy of the results. Profiling bank customer's behaviour by taking relevant data into account will help the bank to determine potential customers and focus on those customers, who will contribute to the bank's profitability. Typically, systems take either transaction or demographic information of the user into account to profile a customer whereas the proposed system takes both transaction and demographic details to predict the profile of a customer.

Credit allowance and credit approval are the most important services provided by the banks and there is a high risk in managing this task. The project also helps the banks to approve the credit based on the profiling done on the customers using their demographic information.

1.2 Problem Statement

Presently, banks are facing issues with profiling which plays a significant role in customer retention and customer satisfaction, which is why continuous and extensive research and development are being done in this direction. A system with good performance helps banks attain higher profitability by customer satisfaction through a focus on valuable customers, which are considered as the main engine in the bank's profitability.

Machine Learning techniques are used to perform this task efficiently. The aim is to design and implement a system that will efficiently profile customers based on their transaction and demographic information of the last six months to identify potential (valuable) customers. The facility must be extended to customers with a good profile, to request a new credit limit for which customers are required to furnish certain information including the number of dependants, applicable income and so on along with the desired credit limit. The inputs provided by the user will be processed to output whether or not the new credit limit can be granted by the bank.

Such a system must be accessible to users, bank customers and bank staff, in the form of a web application with a smooth user experience. These decisions taken by the proposed system, play an important role in the bank's profitability, and there is a need to develop powerful prediction algorithms that achieve higher and more accurate results.

1.3 Objective

The aim is to design and implement machine learning models that will aid in taking important decisions such as predicting the profile of a customer and approving the new credit limit requested by a potential customer. The profile of the customer proves his or her creditworthiness.

Clustering operation is performed on a dataset containing transaction and demographic information of bank customers described for a period of six months. This clustering segregates potential customers from the others and updates the target values in the dataset accordingly. This dataset will be used to train a machine learning model to predict the profile of customers on providing transaction and demographic information of a bank customer. If a customer is identified as a potential customer, then that customer is considered as eligible to request a new credit limit. This is achieved by training a model that predicts whether the requested credit limit must be approved based on certain demographics of the user.

The objective was to design a solution that performs the above tasks seamlessly while attaining high accuracy.

1.4 Scope of the Project

The scope of the project is to deliver a solution that solves the major challenges faced by the banking industry. Profiling is carried out to identify valuable customers. A potential customer is the one who has very good repayment history and dues filed on time. These customers are the major asset for the banks as they help banks to have consistent turnover. The project identifies such customers and extends an additional functionality of requesting a new credit limit, which is feasible for the bank to approve. These decisions play an important role in business and measures must be taken to ensure that the outcome of these decisions leads to profitability.

Research had been carried out to understand the domain and audience of such a system to be able to deliver an efficient solution. The system's usability of the solution was improved by providing an interface with the goal to simulate a good user experience rather than restricting this solution only to those who understand the tools and technologies that were used to implement it. This was achieved by creating a web application to simulate these services provided by a bank that gives accurate results in real-time to both customers of the bank as well as to employees of the bank who are required to track requests made by the customers on the platform.

The models for making decisions were trained using standard Machine Learning techniques such as Improved K-Means [5] and Convolution Neural Networks [4], that were trained and tested using suitable datasets. These models were integrated in a web application implemented using Flask Framework. The project was developed using Python technology stack.

1.5 Methodology

The proposed system aims to improve the profiling of bank customer's behaviour to identify valuable customers. These customers are provided with an additional facility to request for a new credit limit, which may or may not be approved. The users access these features on a web application by logging into their accounts. The task of performing profiling and determining whether the requested credit limit should be approved is done using Machine Learning techniques.

It has been found that Convolution Neural Network (CNN) is a suitable technique to handle bank data as it preserves a complex spatial data structure that the algorithm can exploit. Moreover, it handles the feature engineering part. Dataset has been obtained from the UCI Machine Learning Repository, which has twenty-three attributes and contains transactional and demographic details of thirty thousand bank customers. This dataset is clustered using Improved K-Means to generate the target attribute (profile of the customers). This dataset is now used to train a CNN model to predict the profile of a customer. Another dataset contains certain demographic details such as the number of dependants, gender, applicable income and so on. This dataset is used to train another Convolution Neural Network model to predict whether a potential customer's requested credit limit must be approved. The model for predicting the profile of customers has an accuracy of 99.37% with a loss of 4.6%, and the model to new credit limit approval attains an accuracy of 79.43%.

CHAPTER 2

LITERATURE SURVEY

Literature survey is mainly concerned with reading the research papers based on existing systems and the technologies that can be used to develop the system to solve the problem. This phase involves collection of knowledge and collating the obtained knowledge into useful information and trying to find various methodologies that can be used. Once this knowledge is obtained, it is evaluated and only the information that is required to develop the system is retained and the rest is discarded. The following section give many results on analysis of banking and financial data which was carried out by different methods, techniques. Many researchers have developed and implemented various analysis and prediction models using different data mining techniques

In 2017, Priyanka S. Patil and Nagaraj V. Dharwadkar [1] explained how bank data can be analyzed using machine learning. Banks have data that contains valuable information and it is very important to store, process, manage and analyze this data in order to extract knowledge from it. This knowledge can be used to increase business profit. Here, the authors have presented work on customer retention and fraud detection. Machine learning helps to handle large data in the most intelligent fashion by developing algorithms in order to generate insights from it. In this work authors have used supervised techniques to perform classification in order to identify customer’s behaviour and retain them. The proposed system uses Artificial Neural Network as a machine learning algorithm for classification and prediction as it can process multiple inputs efficiently and also it handles large, complex data easily. Two datasets have been used: German credit dataset is used for fraud detection problem, obtained from UCI machine Learning Repository and another dataset, used for customer retention problem which is prepared under the guidance of the bank. This algorithm gives 72% and 98% accuracy for the first and second dataset respectively.

In 2017, N. Malini and Dr. M. Pushpa [2] explained how credit card fraud can be identified using KNN and outliers Detection. Credit card fraud is also growing along with the development in technology and loss due to these fraudulent acts is billions of dollars. The characteristics of a good fraud detection include: It must be able to identify the frauds quickly and accurately and, in any case, a genuine transaction should not be considered as fraud. The advantage of using unsupervised method over supervised data is that it need not be trained to discriminate between a legal and illegal transaction. K-nearest neighbour algorithm is used largely in detection systems. The results of KNN depend on the following factors: The distance metric used to decide the nearest neighbours, distance rule that is used for the classification from K-nearest neighbour and number of neighbours considered to classify the new sample. The authors claim that by performing over sampling and extracting the principal direction of the data we can use our KNN method to determine the anomaly of the target instance. Hence the KNN method is suitable for detecting fraud with the limitation of memory. The proposed methodology had been compared with power methods and other known anomaly detection methods, experimental results prove that the KNN method is accurate and efficient.

Majid Sharahi and Mansoureh Aligholi [3] explained how data of bank customers is clustered using K-means. The aim of this research is to perform segmentation of bank customers using clustering techniques and provide suitable marketing strategies for each cluster of customers. Banks deal with large amount of customer data entities and the analysis of data obtained from the databases of bank customers can provide useful information to detect hidden patterns in the data and can improve the level of banking services to each group of customers. For experimentation, the authors use dataset containing the information of 60 companies from customer of Bank Sepah in database of this bank and analyse it. Seven fields were selected as a final variable to enter K-means algorithm. These fields are: “Type of Company”, “Lifetime of Company”, “Type of Activity”, “Time of Collaboration with Bank”, “Credit history”, “Type of Credit” and “Amount of credit”. On applying K-Means, 2 clusters are obtained. On analysing information about each cluster, banks can offer some suggestions about marketing strategy for each cluster. Applying suitable strategies can lead to increased customer satisfaction.

In 2019, Waseem Ahmad Chishti and Shahid Mahmood Awan [4] explained how Artificial neural network (ANN) can be used to classify credit card default customer. The purpose of this study and research to classify and predict the credit card default customers payment through the approach of artificial neural network (ANN) known as deep neural network. This paper explains the dataset which signifies Taiwan credit card defaults in 2005 and their previous payment histories taken from popular machine learning dataset resource, UCI Machine Learning Repository. The paper throws light on each and every concept and step required to build, train, validate and test a deep neural network model for classification tasks. Moreover, authors have also tried to elaborate the relevant and important concepts associated with deep neural network models that must be kept in mind during model building. This paper tries to classify the default payment customer with more than 82% accuracy. For implementation, various deep neural network techniques with different libraries are used to attain maximum accuracy to support the authors goal to build a model which can be used for future prediction. This study proves deep neural networks are the only one that can accurately estimate the real probability of default. So, by using this network model, which is more complex, sophisticated and more widely used than a simple neural network and logistic regression model, the classification will deliver a better performance and accuracy.

In 2019, Emad Abd Elaziz Dawood, Essamedean Elfakhrany and Fahima A. Maghraby [5] explained how profiling bank customer's behaviour can be improved using machine learning. The evolution of credit card evolution has caused noticeable changes in the banking industry. Banks have a huge dataset for customer's transactions of their credit cards. Banks are facing challenges such as default prediction, risk management, customer retention, and customer profiling for different purposes to achieve higher profitability and reduce the risk. It is important to identify customers well, to overcome such challenges. Machine learning is the science of enabling computers to act without being programmed and can be employed to design robust solutions for the banking industry. The paper presents a system that is trained using transaction and demographic details of a customer in order to perform profiling of the customer as it gives better results than taking only the demographic or transaction details into account. The data set 'default of credit card clients' is obtained from the archive of UCI (the University of California, Irvine) Machine Learning Repository. In the proposed framework the machine learning techniques are applied to build the customer profile. The profiling phase recognizes the items in a group and places them under target categories and the accuracy rate of the unsupervised techniques is evaluated through Gini co-efficient. The paper compares k-mean, improved k-mean and fuzzy c-means for unsupervised technique and it found that improved k-means gives the best result. The result of unsupervised technique is used as input for supervised technique. The unsupervised technique used is Artificial Neural Network (ANN) which evaluates the results to compare them. On scanning the confusion matrix of the neural network, it achieves an accuracy rate for the neural network in Matlab equal to 98.08%. The paper proposes a model and technique to improve profiling of its customer, get high profitability, and reduce the risk.

In 2017, G. Arunjothi and Dr. C. Senthamarai [7] explained the prediction of loan status in a commercial bank using KNN. The Banking Industry always needs a more accurate predictive modelling system for many issues. Predicting credit defaulters is a difficult task for the banking industry. The loan status is one of the quality indicators of the loan. It doesn't show everything immediately, but it is the first step of the loan lending process. The loan status is used for creating a credit scoring model. The credit scoring model is used for accurate analysis of credit data to find defaulters and valid customers. The objective of this paper is to create a credit scoring model for credit data. Various machine learning techniques are used to develop the financial credit scoring model. In this paper, we propose a machine learning classifier-based analysis model for credit data. In this paper the algorithm used is the combination of Min-Max normalization and K- Nearest Neighbour (K-NN) classifier. This proposed model in this paper provides the important information with the highest accuracy. It is used to predict the loan status in commercial banks using machine learning classifier. In this paper, authors have proposed a loan status model to predict the loan applicant as a valid customer or default customer. The proposed model shows 75.08% accuracy result in classifying credit applicants using the R package. The credit lenders can use this model to make a loan decision on loan proposals. Further, the comparison study has been made with different levels of iterations. The iteration level is 30 based k-NN model gives significant accuracy than other levels. This model can be used to avoid the huge loss of commercial banks.

In 2015, Hui Sun and Mingyan Guo [8] explained how logistic regression is better for credit risk assessments model for small and medium sized enterprise compared to Hosmer- Lemeshow test and Hosmer-Lemeshow randomness test. Small and medium-sized enterprises play a very important role in China's economic and social development. Their development is inseparable from the financial support, and the development of credit business is also an important aspect of a bank's operation. At the time of solving the problem of financing difficulties of small and medium-sized enterprises, how to manage credit risk will become an important subject to study. In this paper, 50 money-borrowing enterprises' financial data in a commercial bank is chosen as samples to construct a logistic risk assessment model. Regression analysis, Hosmer Lemeshow test and Hosmer-Lemeshow randomness test is made. In this paper, first financial indicators are selected which affect credit risk of small and medium-sized enterprises in Tangshan, China. Secondly, based on financial data and the performance of money-borrowing companies in 2012, a logistic model is used to do the analysis. The main conclusions of this paper are as follows: Factor analysis is used to extract principal components based on sample data. Among financial indexes of money-borrowing companies, debt paying ability index, future growth ability index, cost index, operation ability index and profitability index have great impact on credit risk. Logistic regression model is performed with Hosmer-Lemeshow test and Hosmer-Lemeshow randomness test. It is found that the logistic regression model has good fitness of the data. To sum up, with increasing numbers of commercial banks, competition will become increasingly fierce. So, credit risk management is important for commercial banks to achieve great development. It requires commercial banks to establish a unified database and information management system and use modern risk management techniques for effective management.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

System requirements are also known as minimum system requirements. System requirements play major roles in systems engineering, as they form the basis of system architecture, design activities, system integration and verification. System requirements often indicate the minimum and the recommended configuration. The former is the most basic requirement, enough for a product to install or run, but performance is not guaranteed to be optimal. The latter ensures a smooth operation.

The following subsections describe the functional, non-functional, hardware and software requirements of the project.

3.1 Functional Requirements

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly the system services in detail.

Functional requirements of the project are as follows:

- Provide a dedicated dashboard to customers of the bank, which is accessible to the customers on authenticating using their login credentials.
- Provide results of the profile prediction for a user in real-time when he or she is logged in to their account by fetching the transaction and demographic details of the user from the database.
- Render a form to fill certain demographic details (example: Number of dependants) along with the desired credit limit. Some details such as gender, marital status and educational qualification are already present in the database and must be fetched from there. This facility must be extended to a requesting user on his or her account only if they are deemed fit for a potential customer.
- Provide results regarding the approval of the requested credit limit for a potential customer in real-time to when he or she is logged in to their account.

- Provide a tabular display of the request results of a particular user on his or her dashboard, showing only the outcome. An option must be provided to view further details such as values of all the information of the user that were taken as parameters.
- Provide an admin dashboard, that allows the admin to search and view information about the users, their transaction and request history for the purpose of tracking usage of the web application.
- Provide results of the profile prediction for a particular user in real-time to the admin on entering the Bank ID of the user under interest. The transaction and demographic details of the users are fetched from the database.
- Provide a tabular display with search option on the admin dashboard, showing the request results of various users, showing Bank ID, request date and only the outcome. An option must be provided to view further details such as values of all the information of the user that were taken as parameters.

3.2 Non-Functional Requirements

Non-functional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them. Non-Functional Requirements are as follows:

- **Usability**

  Usability measures characteristics such as consistency and aesthetics in the user interface. Consistency is the constant use of mechanisms employed in the user interface while aesthetics refers to the artistic, visual quality of the user interface. It is the ease at which the users operate the system and make productive use of it. This addresses the factors that establish the ability of the software to be understood, used, and learned by its intended users. The application interfaces must be designed with end users in mind so that they are intuitive to use, are localized, and provide access.

- **Performance**

  Performance defines how fast a software system or its particular piece responds to certain users' actions under certain workload. In most cases, this metric explains how much a user must wait before the target operation happens (the login time, credit approval result, etc.) given the overall number of users at the moment. But it's not always like that, performance requirements may describe background processes invisible to users, e.g. backup, background server pages, etc.

- **Security**

  This non-functional requirement assures that all data inside the system or its part will be protected against unauthorized access. But there's a catch. The lion's share of security non-functional requirements can be translated into concrete functional counterparts. If you want to protect the admin panel from unauthorized access, you would define the login flow and different user roles as system behaviour or user actions.

3.3 Hardware Requirements

Hardware system requirements often specify the operating system version, processor type, memory size, available disk space and additional peripherals, if any, needed. The hardware requirements of this system include:

- Processor: Intel Core
- Hard Disk: 50 GB and above
- RAM: 8.00 GB and above
- System type: 64-bit operating system, x64-based processor

3.4 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. The requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. The software requirements for the development of the project include:

- Coding Language: Python 3.8 and above
- Libraries: Keras, Pandas, NumPy, SKLearn, Matplotlib
- Front-end: Bootstrap 4
- Back-end: Flask framework
- Database Management System: MySQL
- Web Server: XAMPP
- Integrated Development Environment: Jupyter Notebook / Spyder
- Text Editor: VS Code / Sublime
- Operating System: Windows 10

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Overview

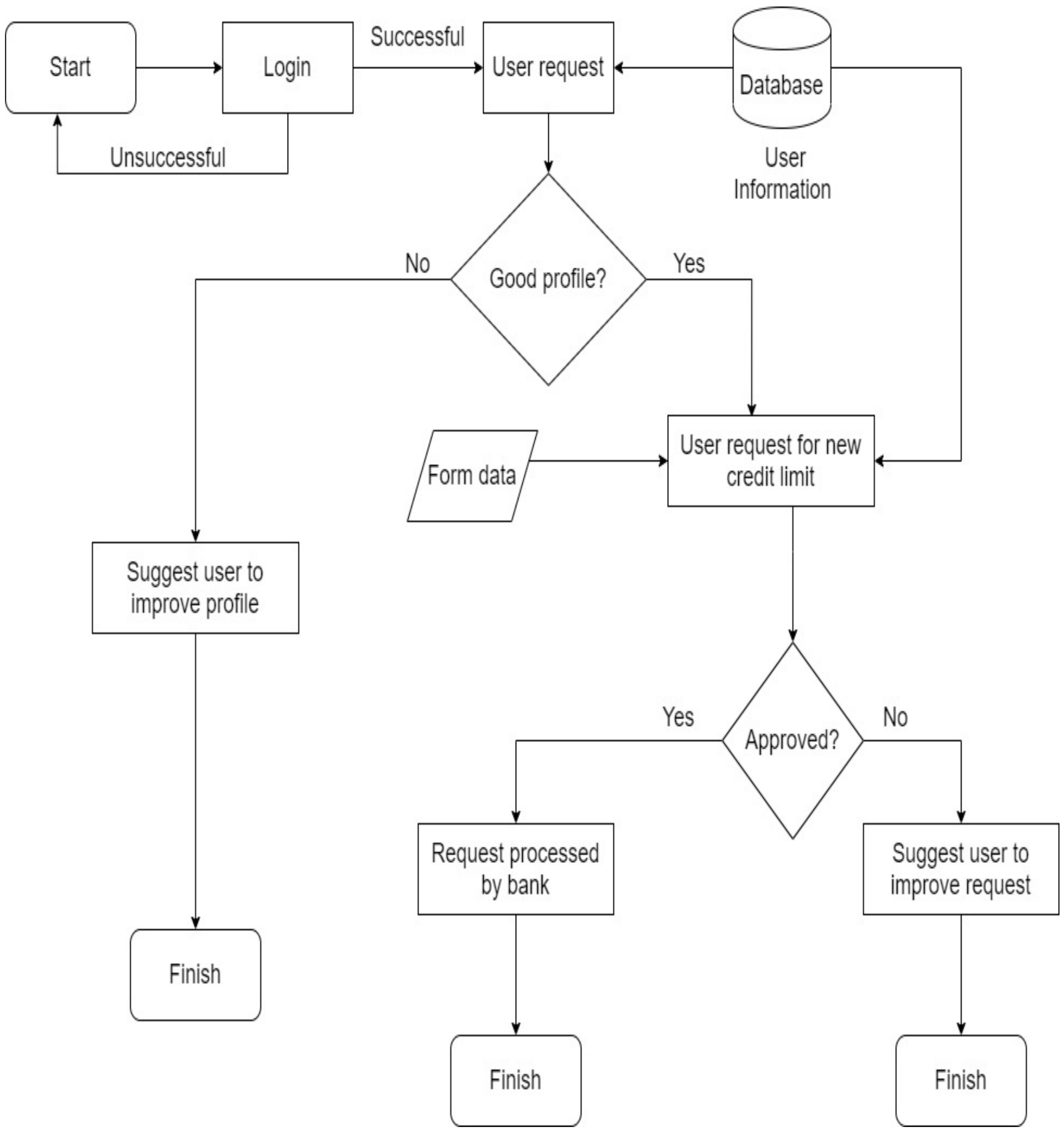

Fig 4.1: System flowchart

The system flowchart illustrated in Fig 4.1 aims to provide an overview of the main task of the project, which is to provide information to the logged-in customer about his or her profile eligibility which in turn determines whether or not the customer is allowed to request for a new credit limit.

The profile prediction technique is improved by taking both transaction and demographic details of the customer as parameters for the underlying neural network. This model is triggered when a customer attempts to request. Only if the customer maintains a good profile, they are given a form to provide some data while some information stored in the database is fetched and fed to the second neural network which determines whether or not the request should be approved.

4.2 Users of the System

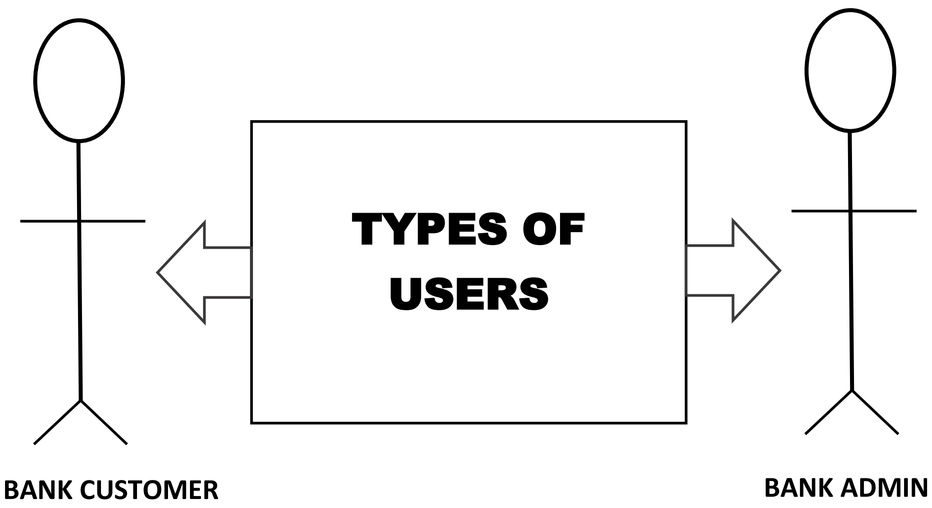

Fig 4.2: Types of system users

The audience of any software solution is the users for whom the website is designed specifically. The users for this web application include the bank customers and the bank admin, who is an authorized employee of the bank. The web application provides a separate and dedicated dashboard for the two types of users that differ in both functionalities and user interface. The two users log in through a common login page where they have to enter their credentials that are their registered email address and corresponding password; and they will then be redirected to the appropriate dashboards.

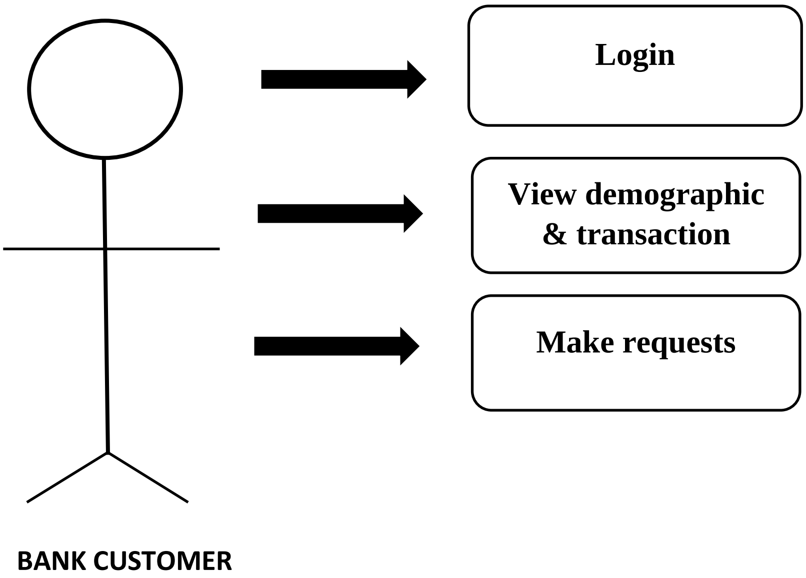

Fig 4.3: Use case diagram for bank customer

Bank customers are the primary users of the web application and who are provided access to the web application by the bank. The use case diagram for a bank customer is shown in Fig 4.3. Each customer has a unique Bank ID, which is issued by the bank. Their demographic and transactional details are verified by the bank and stored in the database so that it can be fetched as and when required. Customers can use this application to view their demographic and transaction details recorded by the bank, to check if the bank considers them as a potential customer. The data to perform profiling of the bank customer (user) logged-in is fetched from the database. If a user does not maintain a good profile and they try to make a request, then a message is rendered suggesting the user to maintain a good profile. In case, they maintain a good profile then the web application will provide them with an additional facility to request a new credit limit. For this purpose, their dashboard renders a form to fill a few demographic details along with the desired credit limit, that will be approved and processed by the bank only if it is considered feasible. In case, the request for a new limit is not considered feasible, the user will be suggested to request a lower credit limit. The website is designed to ensure that the user's parameters are gauged appropriately and give the users a good experience while using the application.

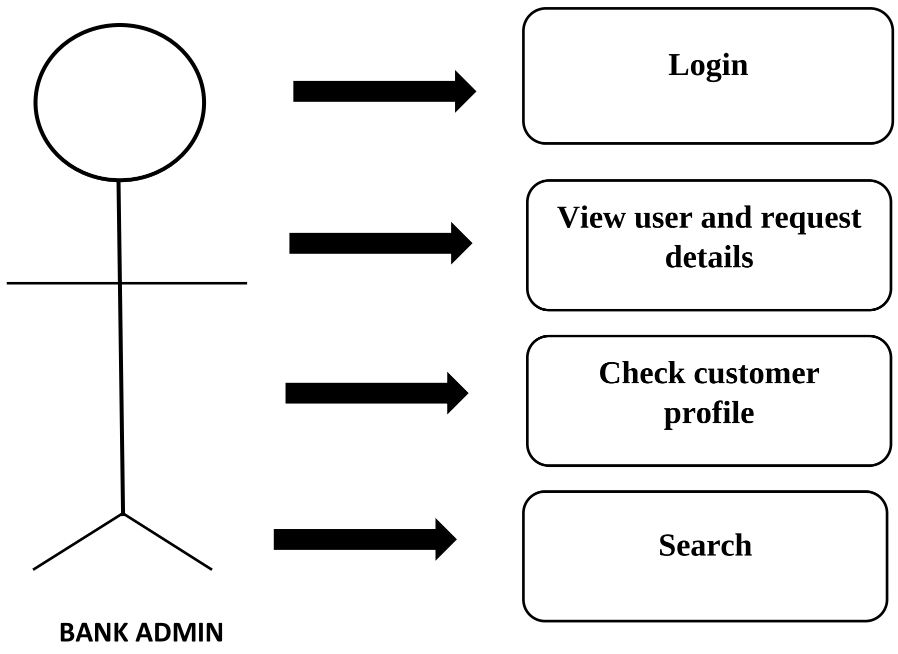

Fig 4.4: Use case diagram for bank admin

The other user of this bank application is the bank admin, who is an authorised employee of the bank. The use case diagram for the bank admin is shown in Fig 4.4. This user is responsible for tracking the usage of this facility and reporting the same to the bank. The admin can view the details of the various customers (transaction and demographic). This user can also view the outcome and details of the requests made by the various customers on the application. Search facility is provided to the admin to allow the performance of his administrative tasks. The admin can also check if a particular user has a good profile, by triggering a task by entering the Bank ID of the customer under focus. The purpose of providing a dedicated dashboard to the bank admin on the website is to facilitate smooth auditing from the bank's end. It is essential to ensure that the bank admin's dashboard is user-friendly to ensure that the admin does not face difficulties in performing his or her duties and proceed with it efficiently.

4.3 Datasets

Using Weka, the attributes that are least contributing towards the class value were identified. Initially the data was loaded, which was made free from extreme values and outliers and by clicking on the select attribute tab on Weka and observing the InfoGainAttributeEval along with the Ranker. This was the approach used to decide which features to be passed to the machine learning model. There are two distinct datasets used to train and test the system. They are:

4.3.1 Dataset 1:

Table 4.1: Attributes in Dataset 1

Attribute no	Attribute name	Description
X1	Limit_BAL	Amount of the given credit
X2	Sex	Gender (1- male; 2- female)
X3	Education	Education (1 = graduate school; 2= university; 3=High school; 4 =others).
X4	Marital status	Marital status (1 = married; 2 =single; 3=others)
X5	Age	Age (years)
X6-X11	Pay_0 to Pay_6	Repayment status
X12-X17	Bill_AMT1 to Bill_AMT6	Amount of bill statement
X18-X23	Pay_AMT1 to PayAMT6	Amount of previous payment
X24	Y	Default Payment

The dataset titled ‘Default of credit card clients' is obtained from the archive of the UCI (the University of California, Irvine) Machine Learning Repository.[5] It is a recently published dataset (obtained in 2015). The attribute details in the dataset are given in Table 4.1. The data set contains 30000 observations and 23 variables. All explanatory variables were normalized. Standardizing data is a data pre-processing step applied to variables to scale these variables to a similar range.[5]

This dataset is clustered to segregate customers with good profiles from those customers who do not maintain a good profile, which is used as target value. The dataset is then used for training and testing the model for predicting the profile of a bank customer using their transaction and demographic data.

4.3.2 Dataset 2:

The dataset used to train and test the model for approving the new credit limit is obtained from UCI Machine Learning Repository[1] and the dataset has attributes which are as follows:

- Gender
- Marital status
- Dependents
- Education
- Self Employed (yes or no)
- Applicant income (annual)
- Co-applicant income (annual)
- Credit Limit
- Credit History

Some key takeaways from the dataset[2] are:

- Applicants who are male and married tend to have more applicant income whereas applicants who are female and married have least applicant income.
- Applicants who are male and are graduated have more income over the applicants who have not graduated.

[1] https://www.kaggle.com/altruistdelhite04/credit-limit-approval-problem-dataset
[2] https://github.com/Architectshwet/Credit-limit-approval-using-Machine-Learning-and-Python

- The applicants who are married and graduated have the more applicant income.
- Applicants who are not self-employed have more applicant income than the applicants who are self-employed.
- Applicants who have more dependents have least applicant income and applicants which have no dependents have maximum applicant income.
- Applicants who have property in urban and have credit history have maximum applicant income
- Applicants who are graduate and have credit history have more applicant income.
- Loan Amount is linearly dependent on Applicant income
- From heatmaps, applicant income and loan amount are highly positively correlated.
- No of applicants who are married are more than no of applicants who are not married.
- Applicants with no dependents are maximum.
- Applicants with graduation are more than applicants with no graduation.

CHAPTER 5

IMPLEMENTATION

The project aims to create a web application that gives results for the profile of a customer and approval of the request for a new credit limit in case of a potential customer. The underlying decisions are taken using Machine Learning techniques. Convolution Neural Network models have been found to perform this task efficiently. The model is saved in .h5 extension file and deployed as a web application built using the Flask framework. The project has been built on Python technology stack.

5.1 Clustering Bank Customers

The task at hand is to cluster the dataset titled ‘Default of credit card clients’ obtained from the UCI Machine Learning repository after it has been pre-processed. This clustering is performed to segregate customers with good profiles from those customers who have not maintained a good profile. The objective of this clustering is to generate target value which forms the twenty-fourth attribute of the dataset, which is used to train the model for predicting the profile of a bank customer. The clustering generates a value of 0 for a customer with a good profile and 1 for a customer who has not maintained a good profile.

The clustering of the bank customers is achieved using the unsupervised technique, Improved K-means. The number of clusters (say k) has been chosen as two, that is to say, k = 2. The algorithm is as follows:

STEP 1: Clusters the data into k groups where k is predefined, where its value is two.

STEP 2: Select k points at random as cluster centres.

STEP 3: Assign objects to their closest cluster centre according to the Euclidean distance function.

STEP 4: Calculate the centroid or mean of all objects in each cluster.

STEP 5: Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.

The number of clusters has been chosen as two after determining the suitable number of clusters using the Elbow method. The optimum value for k can be found using an Elbow point graph. The algorithm is randomly initialised for a range of k values. Within-Cluster-Sum-of-Squares (WCSS) is the sum of squares of the distances of each data point in all clusters to their respective centroids. The resulting graph shown in Fig 5.1 takes WCSS in the y-axis and each value of k on the x-axis.

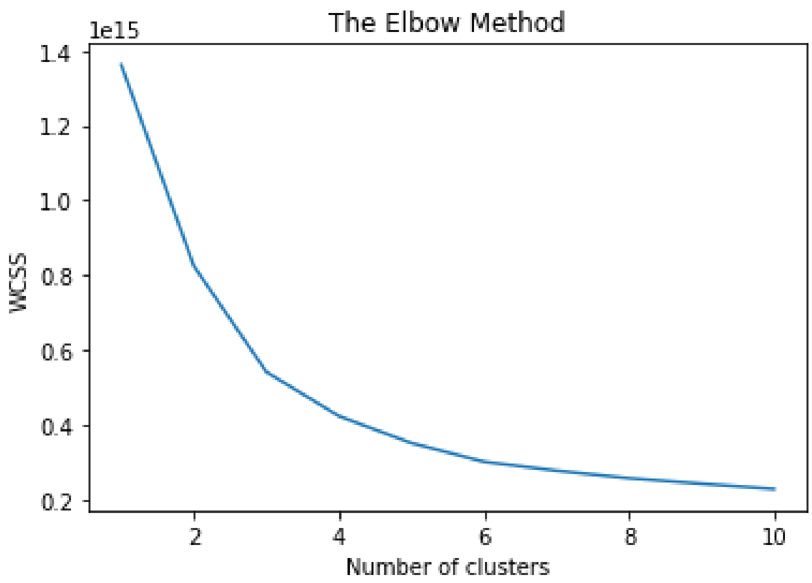

Fig 5.1: Plot of Elbow Method

Improved K-means algorithm was chosen as the unsupervised technique for clustering after comparing its performance against K-means. K-means has the following drawbacks:

- First, it has been shown that the worst-case running time of the algorithm is super-polynomial in the input size.
- Second, the approximation found can be arbitrarily bad with respect to the objective function compared to the optimal clustering.

The Improved K-means algorithm addresses the second of these obstacles by specifying a procedure to initialize the cluster centres before proceeding with the standard k-means optimization iterations. With the K-means ++ initialization, the algorithm is guaranteed to find a solution that is O (log k) competitive to the optimal k-means approach.

The Gini coefficient can identify a more refined collection of non-overlapping clusters to report. For example, it is able to determine when it makes more sense to report a collection of smaller non-overlapping clusters versus a single large cluster containing all of them. It also fulfils a set of desirable theoretical properties, such as being invariant under a uniform multiplication of the population numbers by the same constant.[6]

The error rate measures the error propensity of a model based on a learning sample and applied to the whole population.

A comparison between the performance between the two algorithms on the chosen dataset has been shown in Table 5.1.

Table 5.1: Comparison of Gini coefficients

Clustering algorithm	**Gini coefficient**	**Error rate**
K-Means Clustering	26.37	27.4%
Improved K-Means	37.61	16.44%

```
#load dataset
dataset = pd.read_csv(r'customer_profile.csv')
X  = df.iloc[1:30001,1:24].values
#elbow method
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
#clustering
kmeans = KMeans(n_clusters=2, init='k-means++', max_iter=300, n_init=10, random_state=0)
y_kmeans = kmeans.fit_predict(X)
y_kmeans1 = array(range(30001),dtype='str').reshape(30001)
y_kmeans1.shape
y_kmeans1[0]="kmeans"
for i in range(30000):

        y_kmeans1[i+1]=y_kmeans[i]
        i=i+1
dataset['Y0']=y_kmeans1
dataset.to_csv(r'customer_profile.csv', index=False)

```

Fig 5.2: Improved K-means code for clustering customers

Fig 5.2 shows the code snippet which clusters the bank customers in the dataset into two clusters using the Improved K-means. The Improved K-means is a clustering algorithm, which has more accuracy compared to other clustering algorithms such as hierarchical clustering, density-based clustering or model-based clustering. Fig 5.2 illustrates the code used to implement Improved K-means clustering algorithm on dataset 1. The algorithm considers the 'default of next month' column for clustering into two clusters that are, cluster 1 and cluster 0. The algorithm uses k-means++ as an initialisation method which is used to overcome the random centroid initialization problem. The Elbow method is employed to find the number of clusters. The algorithm gives a new column with data converted into two clusters which are used further as input for CNN built for profiling the customers as potential and non-potential customers.

```python
#load dataset
dataset = pd.read_csv(r'customer_profile.csv')
X  = df.iloc[1:30001,1:24].values
#elbow method
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='random', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
#clustering
kmeans = KMeans(n_clusters=2, init='k-means++', max_iter=300, n_init=10, random_state=0)
y_kmeans = kmeans.fit_predict(X)
y_kmeans1 = array(range(30001),dtype='str').reshape(30001)
y_kmeans1.shape
y_kmeans1[0]="kmeans"
for i in range(30000):
        y_kmeans1[i+1]=y_kmeans[i]
        i=i+1
dataset['Y0']=y_kmeans1
dataset.to_csv(r'customer_profile.csv', index=False)

```

Fig 5.3: K-means code for clustering customers

Fig 5.3 illustrates a code snippet for the implementation of the K-means algorithm on dataset 1. The 'default of next month' column of dataset 1 is passed as a parameter in form of list to this algorithm and clustering is done. The algorithm uses the elbow method to decide the number of clusters. WCSS is evaluated in order to compute the sum of squares of the distances of each data point in all clusters to their respective centroids. The idea is to minimise the sum. The KMeans function takes into account a few parameters, one of which is the init. Init decides the number of cluster formation, we use k-means++ as initialization method to avoid initial cluster initialization for centroids. The final output after implementing this algorithm is that we get a new column segregated into 2 clusters, namely-cluster 0 and cluster 1.

5.2 Profiling of Bank Customers

Banks profile customers in order to get a better understanding of their present and potential customers. Potential customers are valuable customers as they are the ones who have a very good repayment history and dues are filed on time. These customers are important to the banks as they help banks to have consistent turnover. The task is to be able to predict the profile of a customer given twenty-three parameters that include the demographic and transaction details of the customer for the last six months.

It has been found that Convolution Neural Network models work well for non-image data as they can maintain spatial relations for complex information and perform the task of feature engineering.

The model is trained and tested using the 'Default of credit card clients' dataset obtained from the UCI Machine Learning repository after it has been pre-processed and target values have been generated using Improved K-means.

The model considers the following parameters for predicting the profile of a customer:

- Age
- Education qualification
- Gender
- Marital status
- Limit balance
- Bill statement of six months
- Amount of previous months for six months
- Repayment status of six months

The model is implemented using Keras. Keras is a neural network library written in python that runs on top of Theano and TensorFlow. Keras is a high-level wrapper API that wraps the lower-level API. It follows best practices to reduce the cognitive load: it offers simple and consistent APIs. It is required to install this in order to implement the neural network model.

The model uses Rectified Linear Unit (ReLU) as the activation function as it computationally efficient, has fast learning time and does not give in to the problem of vanishing gradient.

Implementation of Convolution Neural Network model is as follows:

- The model used takes all the twenty-three parameters as input and uses the sigmoid activation function.
- It has eight hidden layers deeply connected and uses ReLU as an activation function.
- The output layer uses the sigmoid function to predict 0 or 1.
- 'Adam' is used as an optimizer to optimize the weights and reduce the loss rate.

Why Convolution Neural Network over other techniques?

K-nearest neighbours (KNN) algorithm [7] uses 'feature similarity' to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. can be used for both classifications as well as regression predictive problems. However, it is mainly used for classification predictive problems in the industry. Fig 5.4 shows a code snippet for profiling bank customers using K Nearest Neighbours.

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression [8] is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Fig 5.5 shows a code snippet for profiling bank customers using Logistic Regression.

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

For binary classification [5], accuracy can also be calculated in terms of positives and negatives as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where,

True Positive (TP) is an outcome where the model correctly predicts the positive class.

True Negative (TN) is an outcome where the model correctly predicts the negative class.

False Positive (FP) is an outcome where the model incorrectly predicts the positive class.

False Negative (FN) is an outcome where the model incorrectly predicts the negative class.

In classification, error rate represents the proportion of patterns that have been incorrectly classified by a decision model.

A comparative study had been held by implementing other models, training and testing them using the same dataset and it was found that the Convolution Neural Network model gave the best result, and hence it was used in the final system. The comparison of the results in Table 5.2.

Table 5.2: Comparison of classification techniques

Classification	**Accuracy**	**Error rate**
K Nearest Neighbours (KNN)	75.6	32.4
Logistic Regression	77.3	36.6
Convolution Neural Network	99.6	4.8

```python
import numpy as np
import numpy
import pandas as pd
from pandas import read_csv

dataset = pd.read_csv(r'customer_profile.csv')
dataset.dtypes
df = dataset
df.dtypes

X  = df.iloc[1:30001,1:24].values
Y = df.iloc[1:,24].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.5)
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Fig 5.4: KNN code for profiling

```python
import numpy as np
import numpy
import pandas as pd
from pandas import read_csv
from sklearn import metrics
#load dataset
dataset = pd.read_csv(r'customer_profile.csv')
df = dataset
df.dtypes
X  = df.iloc[1:30001,1:24].values
Y = df.iloc[1:,24].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3)

# import the class
from sklearn.linear_model import LogisticRegression
# instantiate the model (using the default parameters)
logreg = LogisticRegression()
# fit the model with data
logreg.fit(X_train,y_train)
#prediction
y_pred=logreg.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Fig 5.5: Logistic Regression code for profiling

```
seed = 10
numpy.random.seed(seed)
#load dataset
dataset = pd.read_csv(r'customer_profile.csv')
dataset.dtypes
df = dataset
df.dtypes
X  = df.iloc[1:30001,1:24].values
Y = df.iloc[1:,25].values
X1  =X[17]
#neural network
model = Sequential()
model.add(Dense(12, input_dim=23, init='uniform', activation='relu'))
model.add(Dense(8, init='uniform', activation='relu'))
model.add(Dense(1, init='uniform', activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(X, Y, epochs=50, batch_size=100)

scores = model.evaluate(X, Y)
X1
q = model.predict( np.array( [X1,] )  )
```

Fig 5.6: CNN code for profiling

CNN is an unsupervised learning algorithm that works in feed-forward mode in this project. CNN has the seed value as ten, where seed value determines the number of instances to be considered for feeding the neural network. Fig 5.6 shows a code snippet for the CNN architecture used in the project, it has three layers, namely - input layer, middle layer and output layer. The CNN designed has the following architecture: twenty-three input layers since there are twenty-three input attributes, eight dense hidden layers with ReLU as activation function, one output layer for value to be either one or zero. The model uses binary_crossentropy as a loss function in order to minimise the loss by assigning weights to the CNN in a very precise manner. The number of epochs defines the number of iterations the model has to train over the entire dataset. Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data. The algorithm is called Adam. It is not an acronym and is not written as "ADAM" the name Adam is derived from adaptive moment estimation.

5.3 Credit Limit Approval

A good profile indicates creditworthiness hence these customers are given the facility to request a new credit limit. The term credit limit[3] refers to the maximum amount of credit a financial institution extends to a client. A lending institution extends a credit limit on a credit card or a line of credit. Lenders usually set credit limits based on the information given by the credit-seeking applicant.

Convolution Neural Network (CNN) model has been implemented to accept certain demographic information about a customer with a good profile to determine if the requested credit limit is feasible to approve. The model is trained and tested using a dataset obtained from the UCI Machine Learning Repository, which is pre-processed and normalized.

CNN model is composed of two parts: Feature extraction and Classification. In the feature extraction part, the network will perform a series of convolutions and pooling operations during which the features are detected. In the classification part, the fully connected layers will serve as a classifier on top of these extracted features.

The model has eight hidden layers and uses ReLU as the activation function. This is because it is computationally efficient, has fast learning time and does not give in to the problem of vanishing gradient.

The model takes the following parameters as input:

- Number of dependants
- Gender
- Education qualification
- Self-employed (Yes or No)
- Income
- Co-applicable income
- Requested credit limit
- Credit history

[3] https://www.investopedia.com/terms

```
seed = 10
numpy.random.seed(seed)

#neural network
model = Sequential()
model.add(Dense(12, input_dim=10, activation='relu', kernel_initializer='uniform'))
model.add(Dense(8, activation='relu', kernel_initializer='uniform'))
model.add(Dense(1, activation='sigmoid', kernel_initializer='uniform'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X, y, epochs=10, batch_size=50)

#evaluating the model
scores = model.evaluate(X, y)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

Fig 5.7: CNN code for credit limit approval

Fig 5.7 illustrates a code snippet for the CNN architecture for credit limit approval, which is similar to that used for profiling customers but the number of layers is different. CNN takes seed value as ten. It has twelve input layers since the number of input attributes is twelve, eight dense-layered in the middle and one output layer. CNN uses Adam as its optimizer and binary_crossentropy for minimising the loss. The feature analysis on dataset 2 gave a clear visualization of all the twelve features contributing in the evaluation of credit approval. The call to trigger this model is invoked only if the customer is found to be a potential customer.

5.4 Saving the Models

Given that deep learning models can take hours, days and even weeks to train, it is important to know how to save and load them from disk in order to be able to perform prediction in real-time. An H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data. Files saved in the HDF5 version are saved as an H5 or HDF5 file. H5 file is a binary file that holds the weights. Keras also supports a simpler interface to save both the model weights and model architecture together into a single H5 file. Saving the model in this way includes everything required about the model, including:

- Model weights
- Model architecture
- Model compilation details (loss and metrics)
- Model optimizer state.

This means that the model is loaded and used directly, without having to re-compile it. The model can be saved by calling the *save()* function on the model and specifying the filename. Fig 5.8 shows a code snippet to save a model. The saved model can then be loaded later by calling the *load_model()* function and passing the filename. The function returns the model with the same architecture and weights. Fig 5.9 shows a code snippet to load the model from the H5 file.

```
model.fit(X, y, epochs=10, batch_size=50)
scores = model.evaluate(X, y)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
#saving model to disk
model.save('credit_approval.h5')
print ('Model saved to disk')

```

Fig 5.8: Save model to H5 file

```python

#function for  loading model from disk
def get_model():
    global model
    model = keras.models.load_model('model/credit_approval.h5')
    print("model loaded")


```

Fig 5.9: Load model from H5 file

5.5 Web Application

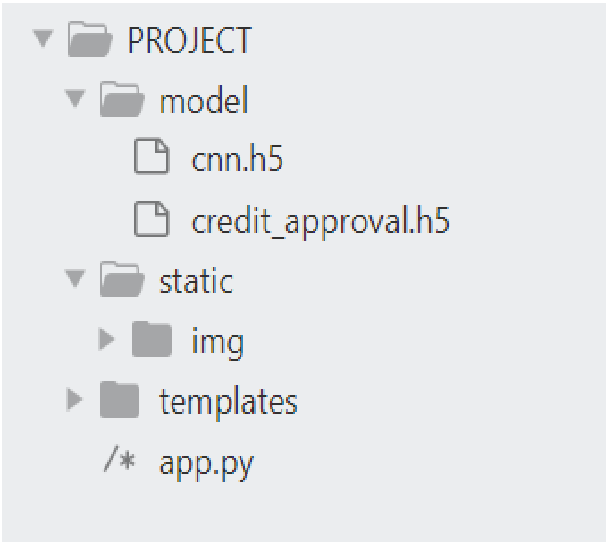

Fig 5.10: Project directory structure

The outcome of models for customer profiling and request for a new credit limit can be accessed by those who do not understand the working of neural network models, by deploying it as a web application with a user-friendly interface. Additional functionalities are also added to support these tasks such as showing customer details, their usage of the portals, request history and so on. The H5 file that saved the model is loaded by the web application framework every time a request is made.

Web Application is built using Flask, which is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application framework. Flask offers suggestions but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that makes adding new functionality easy. The directory structure of the project is shown in Fig 5.10. The typical Flask directory structure has been implemented in this project:

- The folder ‘model’ contains that H5 file of the two models created.
- The folder ‘templates’ contains the .html files used for rendering the user interface on the web application.
- The folder ‘static’ contains the ‘img’ folder which has all the image files used to make the user interface look better.
- The file ‘app.py’ contains Flask APIs that receives customer details through GUI or API calls, computes the predicted value based on the model and returns it. It also makes all the required calls to the database to fetch data and store data.

```

@app.route('/fetch_transactions')
@is_logged_in
def transactions_display():
    cur = mysql.connection.cursor()
    result = cur.execute(
    "SELECT * from transaction_details as t,user_details as u  where t.bank_id = u.bank_id ;")
    if result > 0:
        transactions = cur.fetchall()
        return render_template('fetch_transactions.html', transactions = transactions)
    else:
        msg = 'No transactions found!'
        return render_template('fetch_transactions.html', msg=msg)
    # Close connection
    cur.close()

```

Fig 5.11: Flask code to fetch transactions

Fig 5.11 shows code snippet from app.py in which the route '/fetch_transactions' is used to list transactions of all the customers. The route triggers a MySQL call to the database to fetch transaction details of all customers. The rows obtained from the database are passed to the front-end code to be displayed on the admin dashboard when the admin clicks on the "Transactions" tab on the navigation bar.

```
{% extends 'base.html' %}
{% block head %}
    <title>REQUEST HISTORY</title>
{% endblock %}
{% block body %}
<<div class="mycover">
  <h1>REQUEST HISTORY<br>
    <small></small></h1>
  <hr>
  <table class="table table-striped">
    <tr>
      <th>REQUEST_ID</th>
      <th>TRANSACTION ID</th>
      <th>CREDIBILITY</th>
      <th></th>
    </tr>
    {% for his in history %}
      <tr>
        <td>{{his.request_id}}</td>
        <td>{{his.txn_id}}</td>
        {% if his.credibility == 0  %}
        <td>Eligible</td>
        {% else %}
        <td>Not Eligible</td>
        {% endif %}
        {% if msg == 'record not found' %}
        <td>Credit approval details not found</td>
        {% else %}
        <td><a href="request_history_details/{{his.request_id}}" class="btn btn-success">MORE INFO</a></td>
        {% endif %}

      </tr>
    {% endfor %}
  </table>
</div>
{% endblock %}
```

Fig 5.12: UI code

Fig 5.12 shows a code snippet from the front-end portion of the project. The code snippet shown is used to render the web page for displaying the list of customer requests fetched from the database. This web page is rendered when the admin clicks on the "User Requests" tab on the navigation bar.

5.6 Screenshots from Web Application

This section shows how some of the pages implemented in the web application appear and aims to give an understanding of the purpose of creating certain pages.

Fig 5.13: About Us page

The screenshot shown in Fig 5.13 is the "About" page which gives an idea about the relevance of the web application to any visitor. This page can be viewed by the bank customers and bank admin before and after logging in to their respective dashboards.

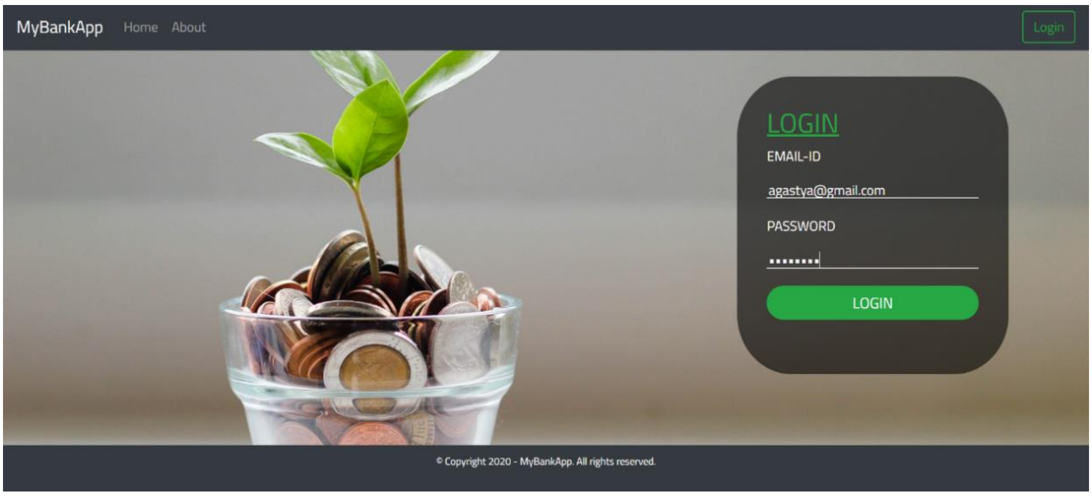

Fig 5.14: Login page

The screenshot shown in Fig 5.14 is the Login page, which is common to all users of the web application, that is, bank customers and bank admins. A login form is presented which takes the user's email address and password for authentications and redirects the user to the authorised dashboard if the correct email address and corresponding password are entered.

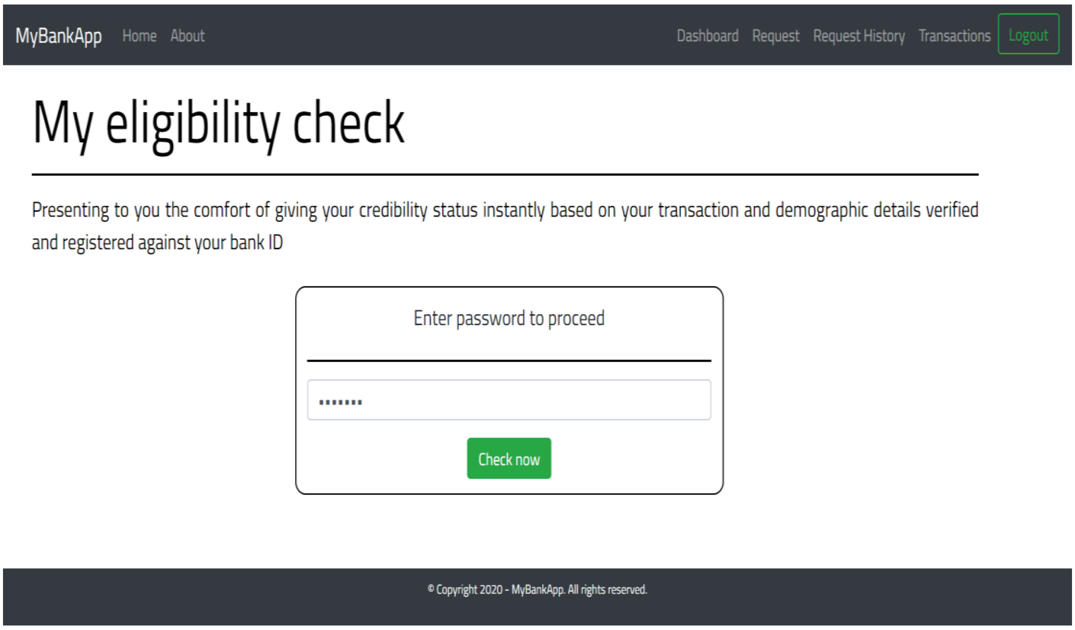

Fig 5.15: Customer eligibility check page

Bank customers logged-in to their dashboard can click on the Request tab in the navigation tab, which will lead them to the page shown in Fig 5.15. Once on this page, the customer needs to enter his or her password. The password entered on this page is verified against that of the customer logged in and only if it is found to be correct then, a call is triggered to the model for determining the profile of the customer.

If the logged in customer has a good profile then a form is rendered as shown in Fig 5.16, where the user (customer) can make a request for a new credit limit as the bank considers him or her as a potential customer. The user has to enter the certain demographic details as shown in the form along with the desired credit limit. However, these are not the only parameters required. When the user clicks on the submit button, other parameters such as gender, marital status and educational qualification are fetched from the database and a call is triggered to the model for approving the credit limit.

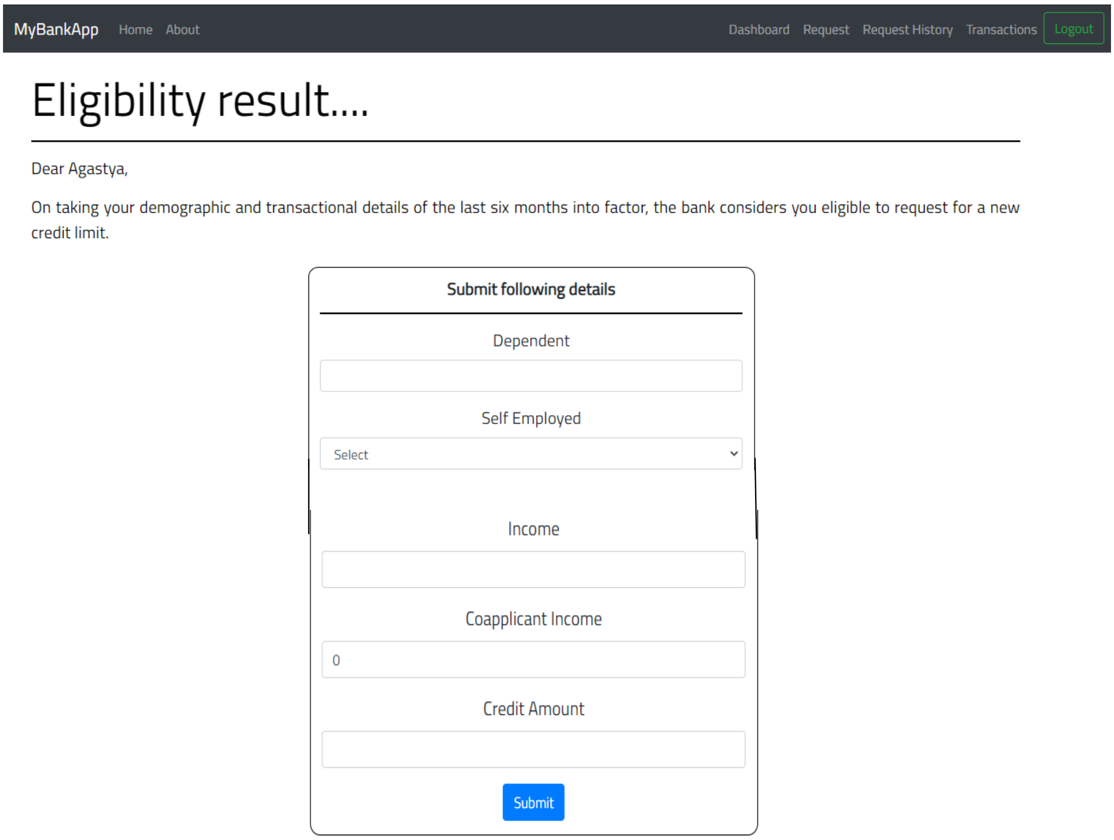

Fig 5.16: Request form for a new credit limit

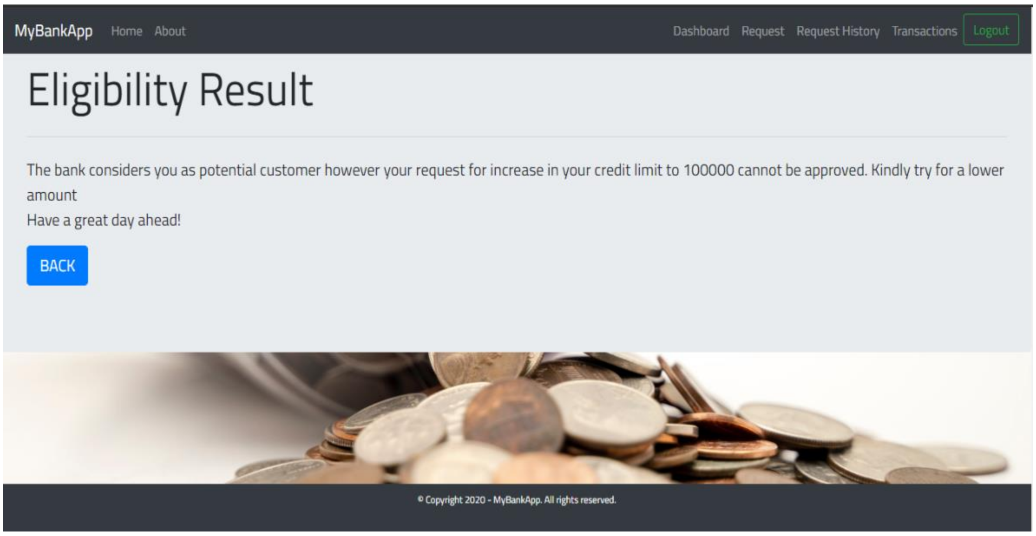

Fig 5.17: Page when credit limit request is rejected

When a potential customer fills the form to put across his or her request for a new credit limit, there are two possibilities, either the request is approved and processed by the bank or the request is rejected and the customer is suggested to request a lower credit amount. Fig 5.17 shows the message displayed when the request is rejected and Fig 5.18 shows the message displayed if the request is approved.

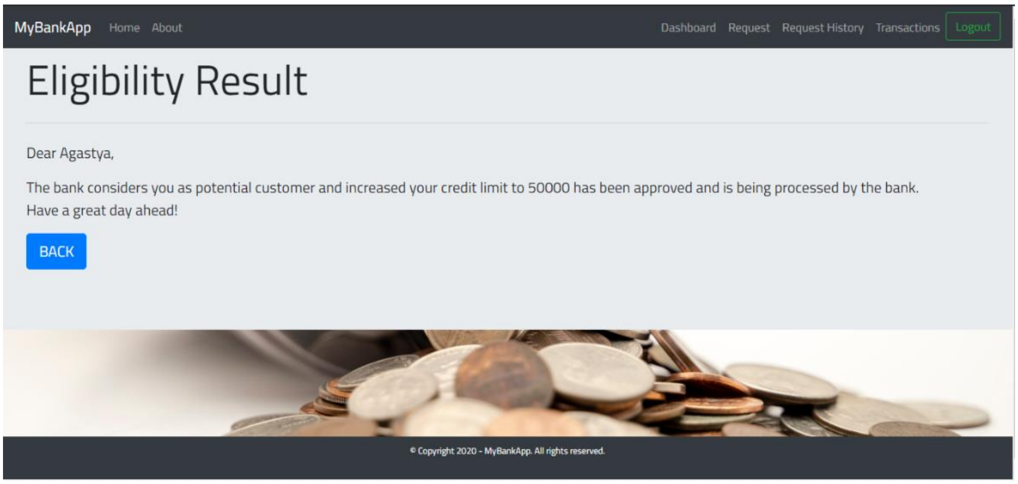

Fig 5.18: Page when credit limit request is approved

Customers can check the requests made by them after logging to their accounts and clicking on the "Request History" tab of the navigation bar. They can view details of the request such as the date and time of the request and values of the parameters by clicking on the "MORE INFO" button. Fig 5.19 shows the request history page for a bank customer.

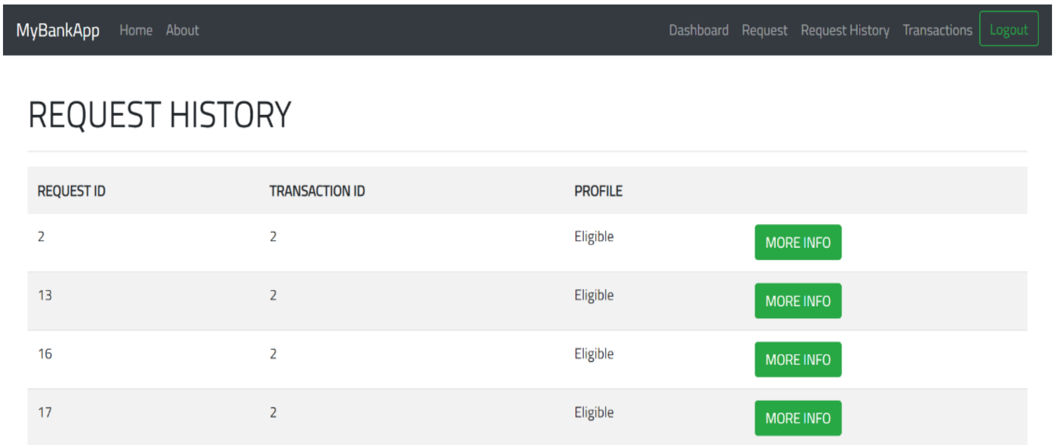

Fig 5.19: Request history (customer)

The bank admin has a separate dashboard and Fig 5.20 shows the landover page, where the admin is redirected to on entering the correct email address and the corresponding password.

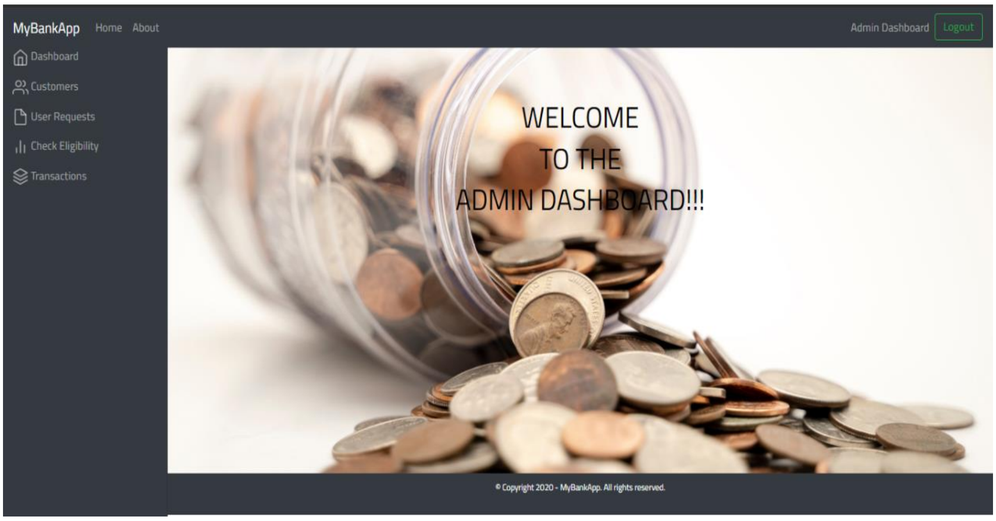

Fig 5.20: Admin dashboard landover page

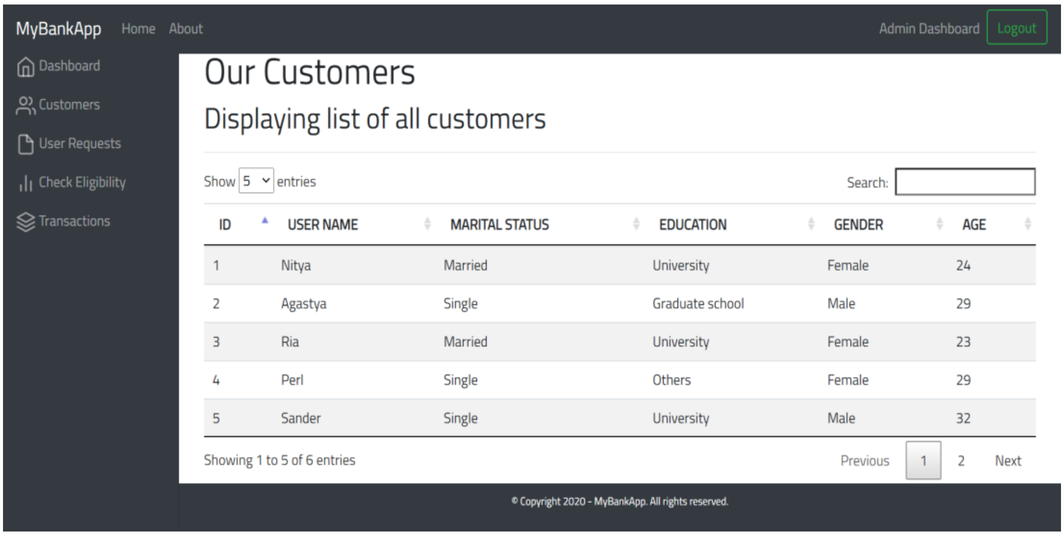

Fig 5.21: List of customers

The admin can look at the list of bank customers which shows their Bank ID and their demographic details by clicking on the “Customers” on the vertical navigation bar on the side. Pagination is provided so that the admin can choose to view a list of five, ten or twenty-five customers at a time and a search box has been provided to aid in filtering results. This is shown in Fig 5.21.

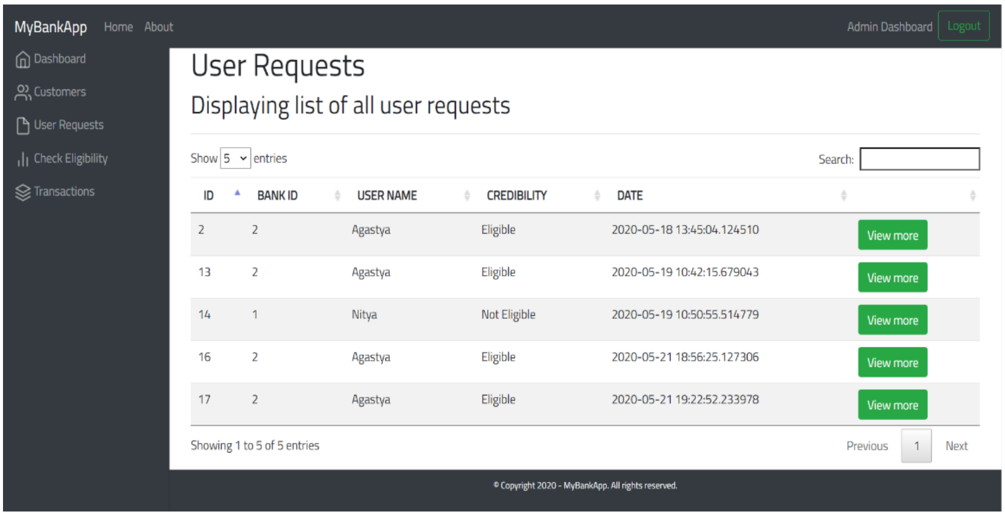

Fig 5.22: List of customer requests

The bank admin can view requests made by the bank customers and this page is shown in Fig 5.22. Pagination and search box have been provided for the admin’s convenience in using the web application. The admin can view details of any request by clicking on the “View more” button for that request.

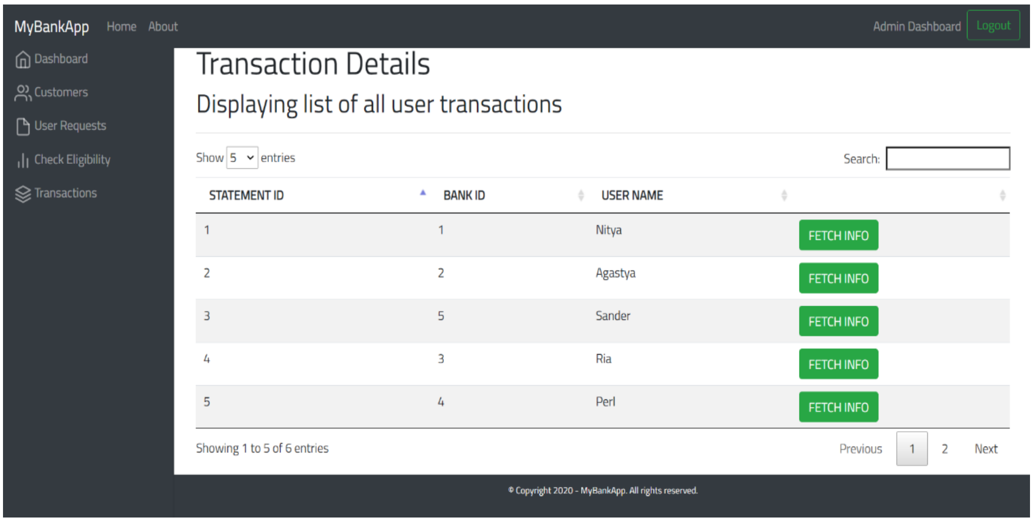

Fig 5.23: List of customer transactions

The admin can view the transaction details of customers by clicking on the “Transactions” tab of the navigation tab. This is illustrated in Fig 5.23.

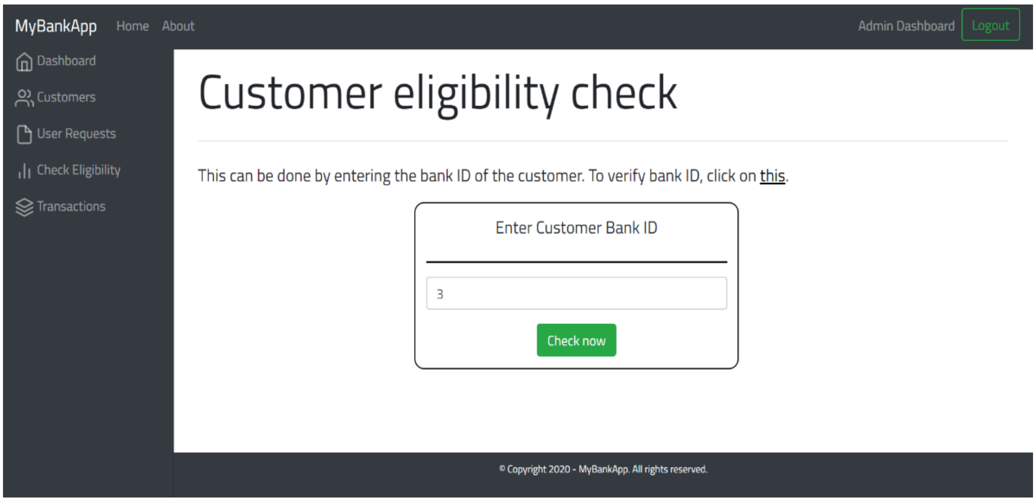

Fig 5.24: Customer eligibility check (admin)

The admin can check whether a customer is a potential customer or not, by entering the Bank ID of a customer, which is used to uniquely identify a customer on the web page shown in Fig 5.24. On clicking on the “Check now” button, transaction and demographic details of the customer (whose Bank ID is entered in the form) from the database and a call is triggered to the model for predicting the profile of the customer.

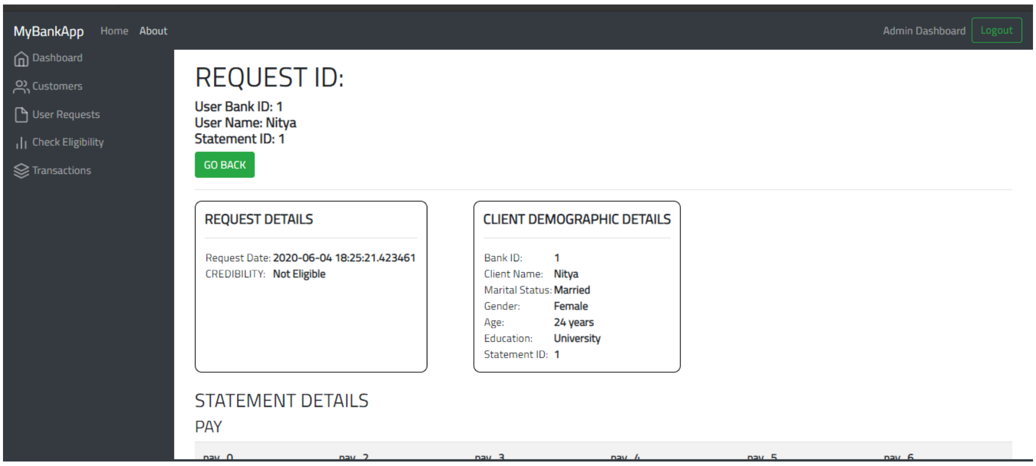

Fig 5.25: Request details of an ineligible customer

The admin can view details of customers by clicking on the “View more” button. Fig 5.25 shows some of the details for an ineligible customer’s request and Fig 5.26 shows some of the details of an eligible user’s request.

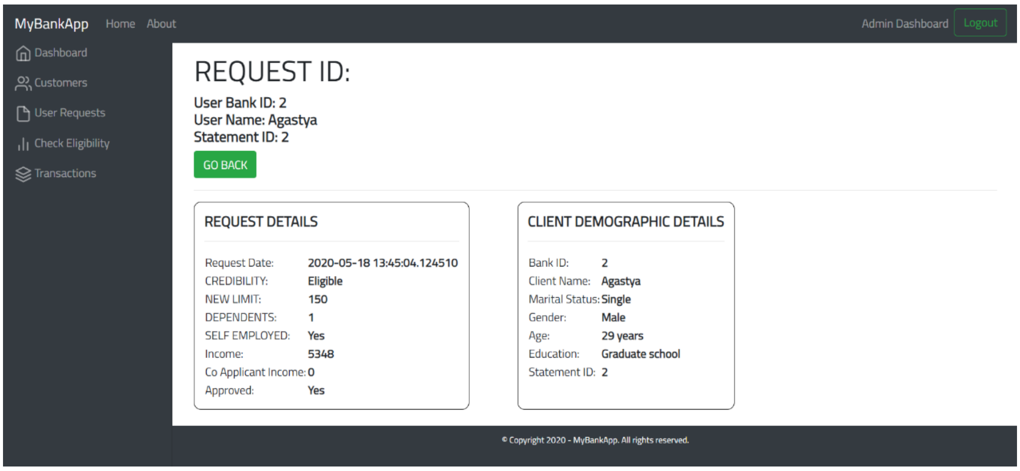

Fig 5.26: Request details of an eligible customer

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Results

The project has been designed and implemented to improve the profiling of bank customer's behaviour and extend the facility of requesting a new credit limit to customers with good profiles. These tasks were performed by training and testing Convolution Neural Network models using suitable datasets obtained from the UCI Machine Learning repository. Their model for predicting the profile of a customer has an accuracy of 99.36% (shown in Fig 6.1) and the model for determining the approval of a new credit limit attains an accuracy of 79.43% (shown in Fig 6.2). These efficient models are designed to cater to the banking sector to solve the problem of profiling which has an impact on the creditworthiness of bank customers. This in turn will allow banks to focus on valuable customers that can help increase business profits.

```
Epoch 45/50
30000/30000 [==============================] - 1s 26us/step - loss: 0.0511 - accuracy: 0.9879
Epoch 46/50
30000/30000 [==============================] - 1s 22us/step - loss: 0.0514 - accuracy: 0.9868
Epoch 47/50
30000/30000 [==============================] - 1s 25us/step - loss: 0.0502 - accuracy: 0.9879
Epoch 48/50
30000/30000 [==============================] - 1s 23us/step - loss: 0.0478 - accuracy: 0.9894
Epoch 49/50
30000/30000 [==============================] - 1s 24us/step - loss: 0.0469 - accuracy: 0.9903
Epoch 50/50
30000/30000 [==============================] - 1s 24us/step - loss: 0.0464 - accuracy: 0.9897
30000/30000 [==============================] - 1s 25us/step
[0.00227163]
accuracy: 99.36%
loss: 4.37%
```

Fig 6.1: Accuracy of profile prediction

```
Epoch 7/10
525/525 [==============================] - 0s 30us/step - loss: 0.5472 - accuracy: 0.7943
Epoch 8/10
525/525 [==============================] - 0s 31us/step - loss: 0.5450 - accuracy: 0.7943
Epoch 9/10
525/525 [==============================] - 0s 30us/step - loss: 0.5453 - accuracy: 0.7943
Epoch 10/10
525/525 [==============================] - 0s 28us/step - loss: 0.5462 - accuracy: 0.7962
525/525 [==============================] - 0s 61us/step
accuracy: 79.43%
```

Fig 6.2: Accuracy of credit limit approval

6.2 Contributions

The project at hand has been proposed and implemented to simplify customer profiling, which is currently one of the challenges faced by the banking sector and has been designed to automate the task of profiling, which determines the creditworthiness of customers. A customer with a good profile is a source of profitability and is considered eligible for credit products such as an increase in credit limit. This software solution delivers the above two requirements in the form of a user-friendly web application with additional features to support the two main tasks. This is done using advanced techniques and using efficient tools and technologies.

The dataset used for profiling is labelled and is clustered to create a new label as a target for neural network classification is the main aspect of this study, which helps to reduce the clustering execution time and get the best accuracy results.

Banks are confronting many challenges like default prediction, risk management, customer retention, and customer profiling for different purposes to achieve higher profitability and reduce the risk. Hence it is necessary to identify customers well, to solve such challenges.

The convolution neural network showed the highest accuracy. So that any bank in the future can use this model and technique to improve the profiling of its customers, get high profitability, and reduce the risk and offer beneficial services to customers that contribute to the bank’s profitability.

CHAPTER 7

TESTING

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect-free. Testing plays an important role in the development of any project as it helps to identify errors, gaps or missing requirements contrary to the actual requirements.

7.1 Types of Testing

- Unit Testing
  It focuses on the smallest unit of software design. It is used to test an individual unit or group of interrelated units. It was done by using sample input and observing its corresponding outputs. Test cases were prepared as shown in section 7.2 of the report and the outputs of the model were verified to ensure that they matched the expected results for given input parameters.
- Integration Testing
  The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output. For instance, after the models were trained and tested, they were integrated into the web application using H5 files and the outputs were verified against the expected behaviour.
- Functionality Testing
  This is a type of software testing whereby the system is tested against the functional requirements. It included checking the flow of the application from one page to another, displaying relevant messages based on the result of the models and fetching appropriate data for the correct user from the database.
- Usability Testing
  It is the practice of testing how easy design is to use on a group of representative users. This involved checking if users are able to navigate smoothly on the web applications and there is no ambiguity as to how to perform a task.

7.2 Test Cases

Test cases were prepared and the outputs produced were verified against the expected output for the given input parameters. Some of the test cases are shown in this section.

Table 7.1 illustrates the test case for a customer who does not maintain a good profile and it found that the model prepared is able to give the expected output for such a customer.

Table 7.1: Test Case 1

Test ID	1
Test Description	Predicting the profile of a customer
Input	Demographic Data: - age: 24 years marital_status: Married sex: Female education: University Transaction Data from January to June (in Rupees): - Limit balance: 20000 Bill Statement: 3913, 3102, 689, 0, 0, 0 Previous payment 0, 689, 0, 0, 0, 0 Repayment Status: 2, 2, -1, -1, -2, -2
Actual Output	Not a good profile
Expected Output	Not a good profile
Test Result	PASSED

The test case for a customer who maintains a good profile, that is, a customer who is viewed as a potential customer is illustrated in Table 7.2. The model prepared is able to give the expected output for such a customer.

Table 7.2: Test Case 2

Test ID	2
Test Description	Predicting the profile of a customer
Input	Demographic Data: - age: 29 years marital_status: Single sex: Male education: Graduate School Transaction Data from January to June (in Rupees): - Limit balance: 500000 Bill Statement: 367965, 412023, 445007, 542653, 483003, 473944 Previous payment 55000, 40000, 38000, 20239, 20239, 13750, 13770 Repayment Status: 0, 0, 0, 0, 0, 0
Actual Output	Good profile implying a potential customer
Expected Output	Good profile implying a potential customer
Test Result	PASSED

Test cases were also prepared for new credit limit approval which is a facility extended to those bank customers who are viewed as potential customers. Table 7.3 illustrates a test case where the requested credit limit is not approved and Table 7.4 illustrated a test case where the requested credit limit is approved.

Table 7.3: Test Case 3

Test ID	3
Test Description	Approve credit limit requested by potential customer
Input	Number of Dependants: 4 Gender: Male Education qualification: Not Graduate Self Employed: Yes Income: Rs. 100000 Co-applicable Income: 0 Requested Credit Limit: Rs. 50000
Actual Output	Request is not approved
Expected Output	Request is not approved
Test Result	PASSED

Table 7.4: Test case 4

Test ID	4
Test Description	Approve credit limit requested by potential customer
Input	Number of Dependants: 0 Gender: Male Education qualification: Graduate Self Employed: No Income: Rs. 1000000 Co-applicable Income: 0 Requested Credit Limit: Rs. 45000
Actual Output	Request for new credit limit is approved
Expected Output	Request for new credit limit is approved
Test Result	PASSED

CHAPTER 8

CONCLUSION

8.1 Conclusion

In the 21st century, science and technology have influenced the growth and functioning of all the industries including the banking industry. The banking sector is one of the oldest businesses in the world and is the foundation of any country's economy. The invention and evolution of credit cards have changed how people spend their money and consume products and services. Although this sector has progressed and grown in recent years, it still faces many challenges such as default prediction, risk management, credit extensions, customer profiling and retention.

The project was carried out during the stipulated period with the aim of designing and implementing a software solution for the banking sector using standard machine learning techniques.

Bank customers can log into their respective accounts on the web application, to make requests for a new credit limit. The user will be given a form to fill for this purpose and it will be rendered only if the bank considers the user as a valuable customer based on the profile of the logged in user. The outcome of profiling bank customers depends on various demographic details (such as gender, education qualifications and so on) and transaction information of the last six months. The parameters analyzed for approving requested credit limit include the number of dependents, marital status and so on along with the desired credit limit.

The objective of the project is achieved by training two Convolution Neural Network models, where each model was trained using a different dataset. The model for predicting the profile of customers has an accuracy of 99.37% and the model to new credit limit approval attains an accuracy of 79.43%. These models are trained and deployed as a web application using the Flask framework. The use of machine learning for customer profiling increases the accuracy of results and improves the efficiency of the entire decision-making process. This model also eliminates human bias and errors.

8.2 Future Scope

The scope of the project implemented can be further enhanced through the following suggestions:

- CIBIL score is a 3-digit number that ranges between 300 and 900, measuring the ability to repay the borrowed amount. A good CIBIL score is required to get better access to credit products and hence this concept must be integrated into the existing system.
- The parameters required to determine the profile of a bank customer can be extended to include Medical Informatics.
- The user interface can be further enhanced to ensure a smooth and pleasant user experience on the web applications.
- Functionalities can be extended on the admin side such generating PDF of user's transaction statements or requests with the bank's letterhead.

REFERENCES

[1] Priyanka S. Patil and Nagaraj V. Dharwadkar, “Analysis of Banking Data Using Machine Learning”, International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2017

[2] N. Malini and Dr. M. Pushpa, “Analysis on Credit Card Fraud Identification Techniques based on KNN and Outlier Detection”, 3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEEICB17)

[3] Majid Sharahi and Mansoureh Aligholi, “Classify the Data of Bank Customers Using Data Mining and Clustering Techniques”, Journal of Applied Environmental and Biological Sciences

[4] Waseem Ahmad Chishti and Shahid Mahmood Awan, “Neural Network a Step by Step Approach to Classify Credit Card Default Customer”**,** 2019 International Conference on Innovative Computing (ICIC)

[5] Emad Abd Elaziz Dawood, Essamedean Elfakhrany and Fahima A. Maghraby, “Improve Profiling Bank Customer’s Behavior Using Machine Learning”, IEEE Access

[6] Han, Junhee, Li Zhu, Martin Kulldorff, Scott Hostovich, David G. Stinchcomb, Zaria Tatalovich, Denise Riedel Lewis, and Eric J. Feuer, “Using Gini coefficient to determining optimal cluster reporting sizes for spatial scan statistics”, International Journal of Health Geographics, 2016

[7] G. Arutjothi, Dr. C. Senthamarai “Prediction of Loan Status in Commercial Bank using Machine Learning Classifier”, International Conference on Intelligent Sustainable Systems (ICISS 2017)

[8] Hui Sun, Mingyuan Guo “Credit Risk Assessment Model of Small and Medium-Sized Enterprise Based on Logistic Regression”, 2015 IEEE IEEM