School of Computer Science

UPES

Dehradun, Uttarakhand

MAJOR PROJECT 1

FinEase - Simplifying Financial Decisions using optimized ML and NLP Techniques

Submitted By –

S.No.	Name	Roll No.	SAP Id
1.	Anusha Tripathi	R2142210133	500091457
2.	Khushi Srivastava	R2142210421	500091444
3.	Pratyaksh Kumar Singh	R2142210584	500091004
4.	Yutika Mishra	R2142210880	500091483

Under the guidance of

Dr. Anand Kumar

School Computer Science (SOCS), UPES

Bidholi Campus, Dehradun – 248007

INDEX

Synopsis Report

Abstract

Through interactive tools and tailored recommendations, the suggested intelligent banking platform uses machine learning (ML) and natural language processing (NLP) to make financial decision-making easier for consumers. To deliver personalized financial advice, the system analyses user inputs such as bank account information, EMI plans, loan choices, and user attitude analyses. The loan and account modules allow users compare choices side by side by presenting recommendations in a tabular style. Sentiment analysis provides information on banking products by using natural language processing (NLP) to categorize customer input as neutral, negative, or positive.

Users may see the entire amount paid and the length of payback by using the EMI module, which visually depicts loan repayment over time. Predictive analytics and real-time market trend analysis improve investment decision-making by predicting changes in the financial landscape. This user-focused platform seeks to improve decision-making, raise financial literacy, and offer data-driven, user-friendly solutions that help users match their financial behaviour with their long-term objectives.

1. Introduction

In this rapidly emerging digital world, personal finance management has been a very complex issue for the individual with an ever-growing, often bewildering scope of varied banking services, loan products, investment options, and financial planning tools. This project will therefore try to solve the complexity of today's increasingly intricate task by suggesting an advanced intelligent banking platform that uses Machine Learning (ML) algorithms and Natural Language Processing (NLP) techniques to provide specific financial solutions for the users.

It shall provide customized banking, loan, and investment guidance to its users. These recommendations will be on the basis of user-specific data, for example, salary, credit score, and financial goals. In case no bank account exists, the system shall provide them with banks suitable for their needs and fee structures accordingly. For those with a bank account, the platform will deliver customized loan recommendations according to their income, credit history, and scoring to provide them with the best lending options. The smart investment advisory feature will also be integrated, which can help users figure out their own personalized and diversified investment strategy for adjusting their risk profiles with long-term financial goals and market trends in mind. The investment product would be pretty diversified, which means it would cover most of the stock products, bonds, and mutual funds. The principal features include a loan eligibility and sanction prediction model. The model uses machine learning to predict users' chances of getting sanctioned for the loan basis of such factors as credit score, debt-to-income ratio, and salary trends. In addition, users will be availed of predictive analytics for market trend forecasting. A completely new module for sentiment analysis will gather user reviews and online sentiment data to give insights into the products of banking and investments so that users may refine their choices given real-world feedback and market perceptions.

This will bridge the individual with the chaotic world of finance through smart, intuitive, user-friendly financial solutions.

2. Literature Review

In [1], It has been exhibited the dynamic landscape of Artificial Intelligence (AI), Machine Learning(ML) stands out as one of the most exhilarating recent technologies. an approach that leverages SHAP(SHapley Additive exPlanations) to enhance the interpretability of XGBoost models, focusing on loan prediction have been introduced. Further, In [2], the AIML technologies has been used precisely for the banking sector. The two computational intelligences are aiding in rewriting how interaction with money is done. AI and ML have given the banking sector a new way of meeting their customers' demands, who are looking for smarter, convenient, and safer ways to access, save, spend, and invest their money. It is, therefore, up to the financial institutions to heed the call of the new era. Customers are a smart lot nowadays. They have realized that technology is not expensive or complicated to learn, everything is bundled together in a smartphone that an ordinary man can easily operate.

In [2], A widely acknowledged definition of RI from the *policy discourse* is provided by **Von Schomberg** as "a transparent, interactive process by which societal actors and innovators become mutually responsive to each other with a view to the (ethical) acceptability, sustainability and societal desirability of the innovation process and its marketable products (to allow a proper embedding of scientific and technological advances in our society)." Sustainability here means meeting present needs without compromising the ability of future generations to meet their own needs. It includes 'grand challenges' such as inequality, hunger and access to water, pollution, loss of biodiversity, violent conflicts, deforestation, acidification of the oceans, global warming, infectious diseases, and pandemics.

In [3], It has been shown that customer retention is one of the most important operation goals in the telecommunication industry because it is directly related to companies' revenue. Therefore, companies perform a variety of activities to establish a long-term relationship with their customers and to minimize consumer defections. To successfully accomplish customer retention activities, it is important to correctly understand customer switching behaviour and identify the reasons behind the churning.

In [4], It addresses the problem of customer churn in the banking sector. This study proposed four machine learning algorithms namely Logistic Regression (LR), Support Vector Machine (SVM), Neural Network (NN), and Random Forest (RF) for examining their efficiency in the process of predicting customer churn. Whereas, the RF algorithm was the best regarding accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC). This study then developed a mathematical model for further understanding RF and issued a new version of it that was named Enhanced Random Forest Algorithm (ERFA), which provided better results than the default Random Forest algorithm results by adjusting some variables related to its implementation using a real dataset, which was the U.S bank customers and with imbalance and balance scenarios.

In [5], It is stated that as the number of Internet users and businesses grows, many clusters of e-commerce applications appear not to be physically linked to each other in the system but are inter-related in business. Online banking has been in practice since the 1980s, when it was first introduced by four major banks in New York. The study of commercial financial transactions made a short-term expectation using a logistic regressive model and a SVM model. The comparison between them concluded that the SVM model prediction was better than 100% of the logistic regressive model and 97.67%, respectively.

In [6], it is stated that the adaptation of AI and ML has been noticeable in the banking sector for reasons well known now. After doing an extensive literature view of the selected research papers, researchers have been able to conclude the most common areas where these two technologies have been able to contribute. Most of it has been in activities around Customers, including acquisition, retention, churning, enhanced experience and ease, and loyalty. The second most researched area is automated credit scoring techniques, where many scholars did find the application of AI and ML on a very advanced level.

3. Problem Statement

In today’s financial landscape, individuals face several challenges when managing their finances and making informed decisions about banking, loans, and investments. The current financial sector presents problems such as:

i. Overwhelming Financial Choices: With numerous banking products, loan offers, and investment options available, users struggle to navigate and select solutions tailored to their unique financial circumstances.[6]

ii. Difficulty in Assessing Loan Eligibility: Individuals often face uncertainty in determining which loans they are eligible for, leading to frustration and confusion during the loan application process.[6]

iii. Lack of Personalized Investment Guidance**:** Without personalized financial advice, users frequently make suboptimal investment choices, resulting in missed opportunities for financial growth or unnecessary risks.[1]

iv. Complexity in Understanding Financial Markets: The ever-changing nature of financial markets makes it difficult for users to predict trends and make informed decisions about where to invest.[4]

The lack of an efficient, personalized solution highlights the need for an intelligent banking platform. This project aims to fill these gaps by offering users personalized recommendations and insights, streamlining their financial decision-making process across banking, loans, and investments.

4. Objective

The objective of this project is to develop a user-centric banking website that simplifies financial decision-making through:

(i) To Analyse the user data for Personalized Bank Recommendations using keyword-based filtering or text-based search filtering algorithm.
(ii) To enhance the sentiment analysis for Personalized Bank Recommendations using NLP.
(iii) To record and analysis of financial marketing trends using ML in real-time scenario.

5. Methodology

The methodology follows as Fig. 1 and Fig. 2.

i. Personalized Bank Recommendations

Comprehensive data collecting is the first step in the algorithm's development, with an emphasis on customers' financial profiles (such as age, income, banking preferences, and credit scores) and specific bank information (such as interest rates, fees, and features). This data is transformed into a format appropriate for machine learning models by feature engineering and analysis. Among the tasks include managing missing numbers, standardizing data, and developing features like savings potential. Individualized bank suggestions Use text-based search engines or keyword-based filtering to match customer preferences with pertinent banking goods. Advanced methods that learn from past user data and improve recommendations over time, such as collaborative or content-based filtering, can further improve accuracy.

Based on the debt, interest rate, and loan term, a user-friendly interface shows customized banking alternatives and tools to determine total interest payments. For improved decision-making, the platform might provide interactive features and side-by-side comparisons. Conversational user interaction is made possible by integrating NLP, increasing accessibility and engagement. Feedback systems aid in model improvement, ensuring that it changes to better suit user needs.[1]

ii. Sentiment analysis for Personalized Bank Recommendations using NLP

Sentiment analysis examines user reviews to improve tailored bank recommendations. To guarantee uniformity, the procedure starts with data cleansing, which includes standardizing language, eliminating unusual characters, and fixing mistakes. Tokenization, stop word removal, and lemmatization are handled by NLP tools such as NLTK or spaCy to get data ready for analysis.

Sentiment analysis is used to categorize reviews as neutral, negative, or positive using the pre-processed data. This assists users in making smarter banking decisions by identifying trends, such as favourable comments about low interest rates or unfavourable comments about hidden expenses. By comprehending context, deep learning methods like RNNs or models like Naive Bayes can improve accuracy.

By adding sentiment analysis to the recommendation system, users may make better decisions by receiving enhanced information like sentiment summaries and highlighted reviews. By keeping suggestions up to date, this integration enables users to choose banks that best suit their requirements and tastes.[5]

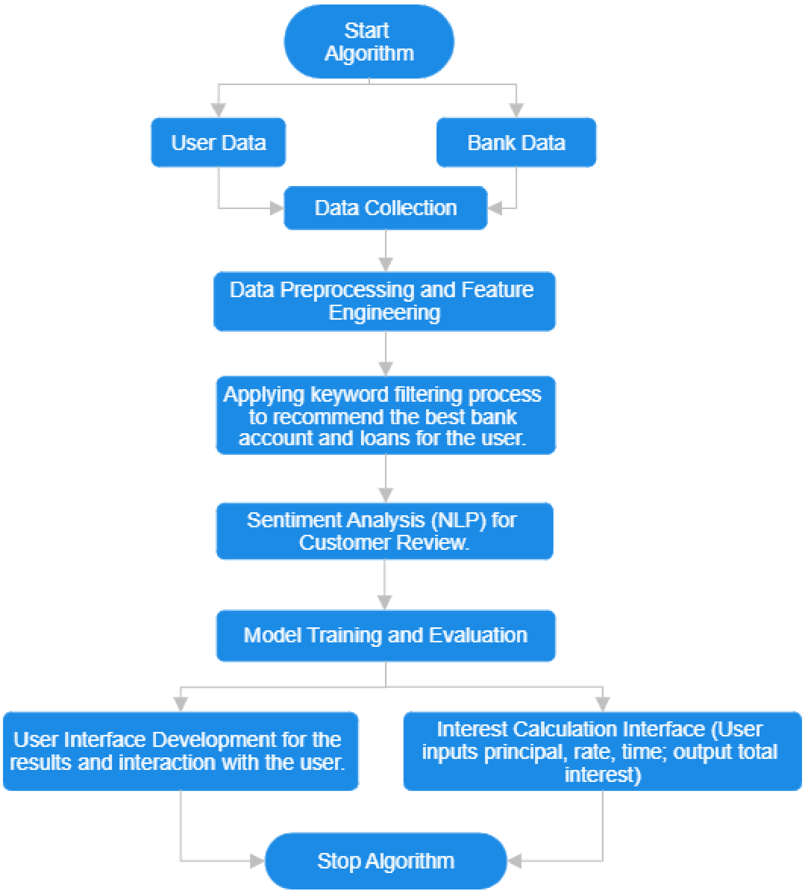

Start Algorithm
User Data
Bank Data
Data Collection
Data Preprocessing and Feature Engineering
Applying keyword filtering process to recommend the best bank account and loans for the user.
Sentiment Analysis (NLP) for Customer Review.
Model Training and Evaluation
User Interface Development for the results and interaction with the user.
Interest Calculation Interface (User inputs principal, rate, time; output total interest)
Stop Algorithm

Fig 1: Design diagram for the working of the project for personalized recommendation and sentiment analysis.

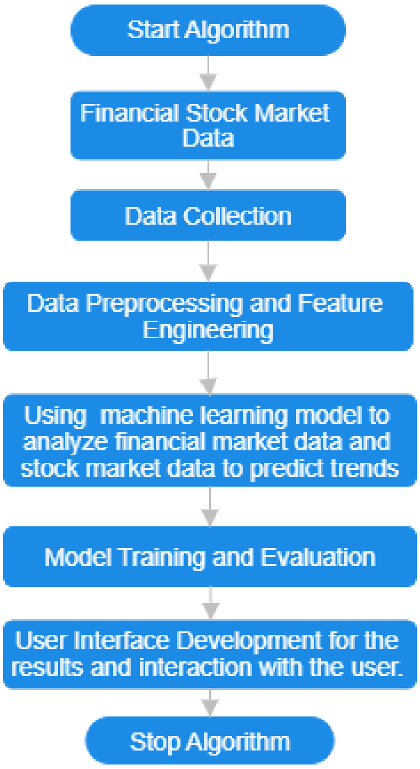

Start Algorithm
Financial Stock Market Data
Data Collection
Data Preprocessing and Feature Engineering
Using machine learning model to analyze financial market data and stock market data to predict trends
Model Training and Evaluation
User Interface Development for the results and interaction with the user.
Stop Algorithm

Fig 2: Design diagram for the working of the project for predictive analytics of financial market trend.

iii. Real Time Financial Market Dashboard

For the Financial Market Dashboard, the methodology first starts from the integration of comprehensive financial market data through its API widgets on the TradingView. These widgets contain detailed insights on the financial markets, including real-time 'top gainers,' 'losers,' and 'most actively traded' stocks of selected or available indices.

This leads to a well-designed User Interface and Responsiveness of the dashboard. The dashboard is powered with real-time market data through the TradingView API Integration that offers interactive dashboard which display wholesome insights such as stock indices, forex, top gainers and losers, etc.

6. Architecture of the proposed model

6.1 Data Collection Layer

- Bank and Loan Data: With the help of data available online on the official websites of the 4 banks which are – ICICI Bank, HDFC Bank, Axis Bank and SBI Bank; wide range of data is collected of all the types of Accounts (both saving and current accounts) and Loans. The data collected consisted of variety of bank offerings which included services offered, eligibility criteria, key features, type of loan and account, Loan amount, Income of the user, Purpose of loan, and various other attributes. The combined data of Loan Types of all 4 banks formed a table of 37 rows and 8 columns. Whereas for saving account, there were 30 rows and 6 columns; and 29 rows and 6 columns for Current Account data. All the column in the table were categorical data.
- User data – For understanding the preferences of the user, various inputs were taken from them. For Loan preferences – Type of loans, User age, Eligibility, and two key features of loan were taken as user input. All the 5 inputs are taken as integer values which are later mapped to the actual data referencing to a relevant meaning in the data. Similarly, for Account recommendation; type of account(saving/current), type of accounts, User age, Eligibility, and two key features of accounts were taken as user input. All the 6 inputs are taken as integer values which are later mapped to the actual data referencing to a relevant meaning in the data.

6.2. Data Processing and Preprocessing Layer

- Data Cleaning: After the collection of data, the entire dataset is checked for any inconsistencies, missing data in the rows and then handling them in case any missing data is found, and ensures data uniformity across multiple sources. An additional column titled as ‘Bank’ is added in the individual dataset of all the banks before combining the dataset into a single dataset. The column ‘Bank’ consists of the name of the bank of which the data belongs. This addition of column is done for both Loan and Account dataset.

- Feature Engineering: At each step of filtration of relevant data for user, the data needs to be transformed into a form from which data could be easily mapped with the preferences of the user. Extracts relevant features from user and bank data for personalized recommendations. For example, the age criteria is a column with categorical datatype and it consist of a range of age. We are taking input of age from user as an integer value, so the data from the table needs to first extract the range from the string and then convert the number into integer to check if the user age falls in the given range of age criteria.

6.3. Personalized Recommendation Engine

- Text-based Search Filtering Algorithm: For recommendation of the most suitable account and loans from the best 4 banks, keyword – based search filtration was applied on the dataset which helped in the selection of the most suitable accounts or loans according to the user preferences specified by them. It is a filtering algorithm used to recommend suitable loans and accounts to the user. Majorly 5 to 6 times the filtering process was applied, at each step user preference was taken and then according to that preference the suitable rows were filtered out. At each step if any feature engineering was required then it was done to provide the best results.

6.4. Sentiment Analysis Module (NLP)

- Natural Language Processing (NLP) Techniques: Using the standard Natural Language Toolkit (nltk) offered in python for NLP, sentiment analysis is applied on the feedbacks given by the user. Using nltk, the feedback first undergoes Tokenization, it is a fundamental step involved in NLP for understanding the language. It involves division of the entire statements into units called tokens.

- SentimentIntensityAnalyzer is initialized. It is a part of the nltk module which is used for Sentiment Analysis. After tokenization, sentiment analysis is applied to customer reviews using NLP models. A sentiment score is calculated for the entire statement. The score consists of probability values for all the cases – positive, negative, and neutral. Text data is processed to classify reviews as positive, negative, or neutral, aiding decision-making. If the compound score is greater than or equal to 0.05, then it is classified as positive; else if the compound score is smaller than or equal to -0.05, it is classified as negative; otherwise, it is classified as a neutral statement.

- User Feedback Analysis: Reviews are analysed in real-time to display the feedbacks given by the users along with its classification as positive, negative or neutral.

6.5. Financial Market Dashboard

- TradingView API Integration: The core part of the dashboard is the TradingView API, allowing a visualization of real-time market data. The API comprises the interactive widgets like Market Overview and Top Gainers & Losers to display the indices, futures, forex, and bonds. The TradingView API also provides real-time updates, trends analysis at high levels of detail, and performance metrics, enabling the trading user to analyze market conditions effectively.
- User Interface and Responsiveness: The user interface of the dashboard to the financial market is very clean and professional for the users. Related sections like the Market Overview and Top Gainers & Losers are shown side by side in panels to allow for coherent orchestration and easy access with instant views. Extensive use of CSS lends consistency everywhere, with responsive design techniques applied to ensure that elements such as charts and data tables are adjusted elegantly for optimal readability on any device.

6.6. Frontend User Interface

- Interactive Dashboard: The frontend is developed using HTML, CSS and Javascript. An interactive interface is constructed which would accept various user preferences for loans and accounts at run time and would dynamically work of filtering out the best results for the user according to their preferences. The interface at each filtering step would provide with a large number of options for the user to select from and then the preferences would be matched from the dataset in the backend to suggest results.

- Custom Calculators: The interface would also provide the user with the option to calculate the Equated Monthly Instalment (EMI). The interface would accept loan amount, rate of interest and the duration of the loan from the user. In the backend these inputs would be used to calculate EMI and then provide the user with a brief detail of how much amount they would be paying each month for the given years. The result would be displayed in the form of a graph plotted between the loan paid and the number of years elapsed.

7. System Requirements (Software/Hardware)

Hardware Interface:

The Processor required for the project should be of Intel Core i5 or equivalent. RAM is required to be of 8GB or higher. Minimum of 256GB SSD for fast data access and storage. A Stable broadband connection for accessing data sources and online resources. The GPU which can be used for faster processing, should be NVIDIA GTX 1060 or equivalent.

Software Interface:

The Operating System required for this project should be Windows 10, macOS, or Linux. For the backend, python will be used. Python version 3.7 or higher will be installed. Python Libraries - NumPy, pandas, scikit-learn, TensorFlow, NLTK for NLP tasks will be used. Integrated Development Environment (IDE) required for this project is Visual Studio Code and Jupyter Notebook.

Web Interface:

For an interactive interface, frontend Technologies like HTML, CSS and JavaScript will be used.

8. Work Done till Now

Using the official website data of 4 largest banks of India – HDFC Bank, ICICI Bank, SBI Bank and Axis Bank, data of accounts and loans are extracted.

Loan Dataset – It consists of 5 columns: 'Account Type', 'Eligibility', 'Age Criteria', 'Minimum Balance Requirement' and 'Key Features'.

- 'Account Type' – Name of the account type.
- 'Eligibility' – Eligibility criteria for account enrolment
- 'Age Criteria' – Age Criteria for account enrolment
- 'Minimum Balance Requirement' – Total amount of minimum balance to have in the account.
- 'Key Features' – All the necessary details about the account type.

Account Dataset – It consists of 7 columns: 'Loan Type', 'Purpose', 'Eligibility', 'Age Criteria', 'Income Criteria', 'Loan Amount', 'Features'.

- 'Loan Type' – Name of the Loan type.
- 'Purpose' – Key purpose of the Loan type.
- 'Eligibility' – Main criteria to be fulfilled by the user for loan approval.
- 'Age Criteria' – Age eligibility criteria for loan category.
- 'Income Criteria' – Income range of the user.
- 'Loan Amount' – Loan amount that could be sanctioned for the given loan type.
- 'Features' – Main important features of the loan type.

Data Analysis –

(i) Distribution of Age Criteria in the Saving Account in Banks

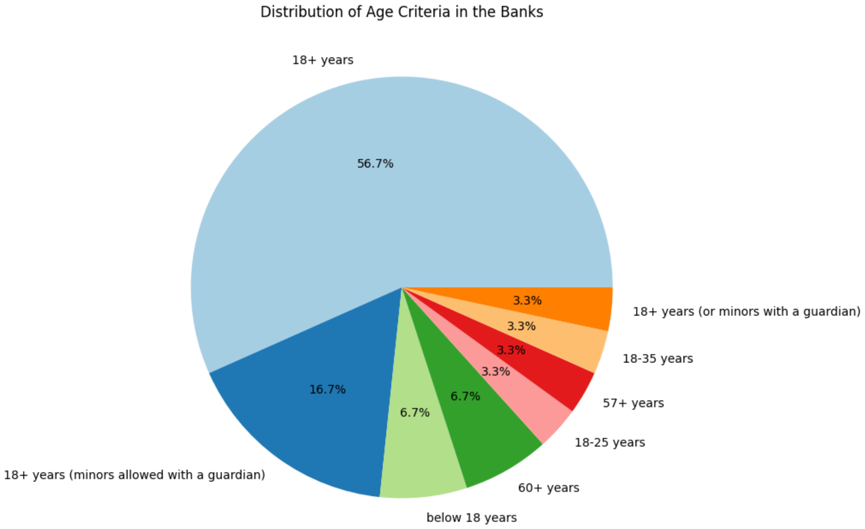


Fig. 3: Distribution of Age Criteria in the Saving Account in Banks using pie chart.

Figure 3 displays a pie chart representation of all the age criteria offered by the saving accounts from all the 4 banks. The results are as follows –

- 18 + years – 56.7%
- 18+ years (or minors with a guardian) and 18+ years (minors with guardians) – 20%
- 18 – 35 years – 3.3%
- 57+ years – 3.3%
- 18-25 years – 3.3%
- 60+ years – 6.7%
- Below 18 years – 6.7%

(ii) Distribution of Saving Accounts in Banks

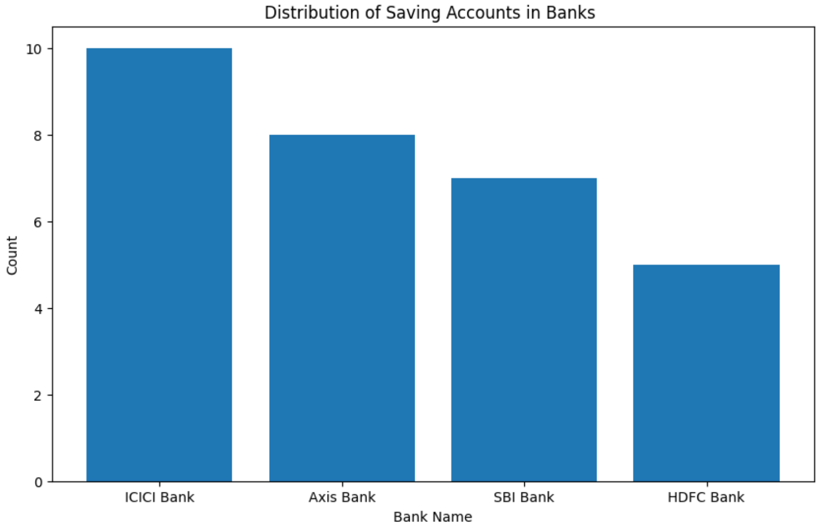


Fig 4: Distribution of Saving Accounts in Banks using bar graph.

Figure 4 displays a bar chart representation of the distribution of the number of saving accounts in all 4 banks. The results are as follows –

- ICICI Bank – 9
- Axis Bank – 8
- SBI Bank – 7
- HDFC Bank – 6

(iii) Distribution of Age Criteria in the Current Account in Banks

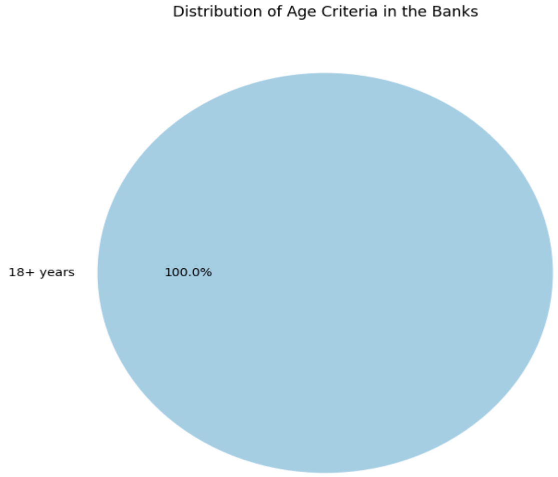


Fig 5: Distribution of Age Criteria in the Current Account in Banks using pie chart.

Figure 5 displays a pie chart representation of all the age criteria offered by the current accounts from all the 4 banks. The results are as follows –

- 18+ years – 100%

(iv) Distribution of Current Accounts in Banks

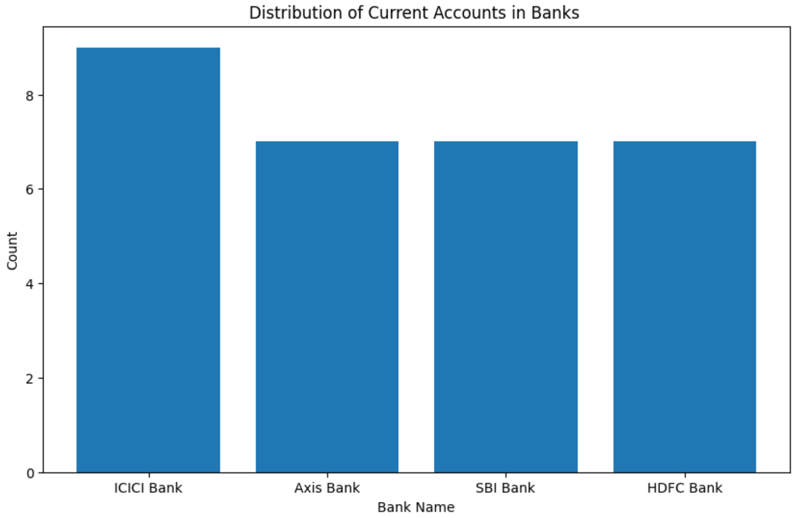


Fig 6: Distribution of Current Accounts in Banks using bar graph.

Figure 6 displays a bar chart representation of the distribution of the number of current accounts in all 4 banks. The results are as follows –

- ICICI Bank – 8
- Axis Bank – 7
- SBI Bank – 7
- HDFC Bank – 7

(v) Distribution of Age Criteria for Loans in the Banks

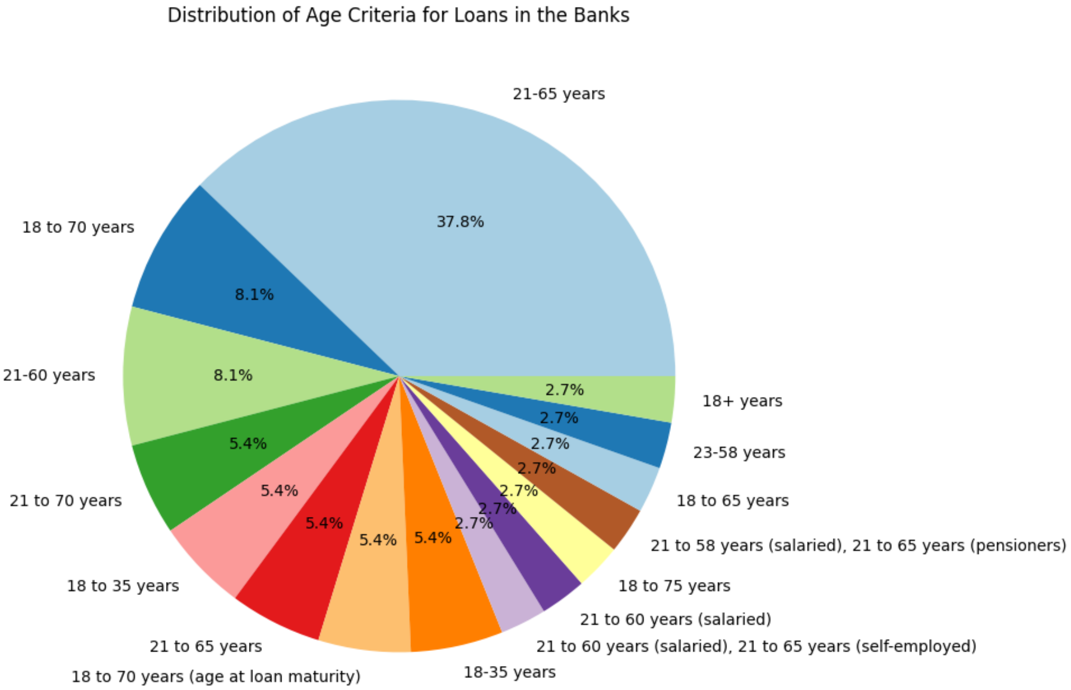


Fig 7: Distribution of Age Criteria for loans in Banks using pie chart.

Figure 7 displays a pie chart representation of all the age criteria offered by loans from all the 4 banks. The results are as follows –

- 21-65 years – 16
- 21-60 years – 5
- 18-70 years – 5
- 18-35 years – 4
- 21-70 years – 2
- 18-75 years – 1
- 21-58 years – 1
- 18-65 years – 1
- 23-58 years – 1
- 18+ years – 1

(vi) Distribution of Number of loans in Banks

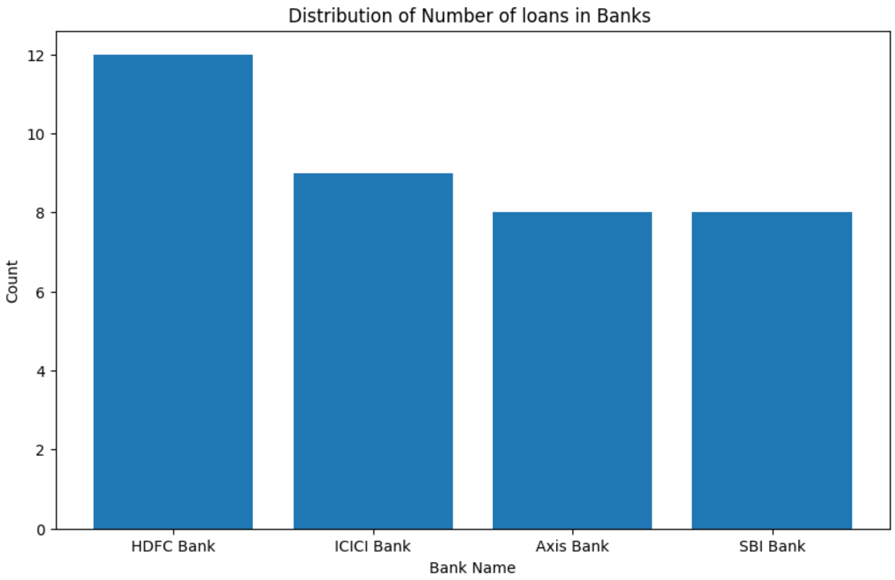


Fig 8: Distribution of Number of loans in Banks using bar graph.

Figure 8 displays a bar chart representation of the distribution of the number of loans in all 4 banks. The results are as follows –

- ICICI Bank – 9
- Axis Bank – 8
- SBI Bank – 8
- HDFC Bank – 12

(vii) Numerical Distribution of Loan types

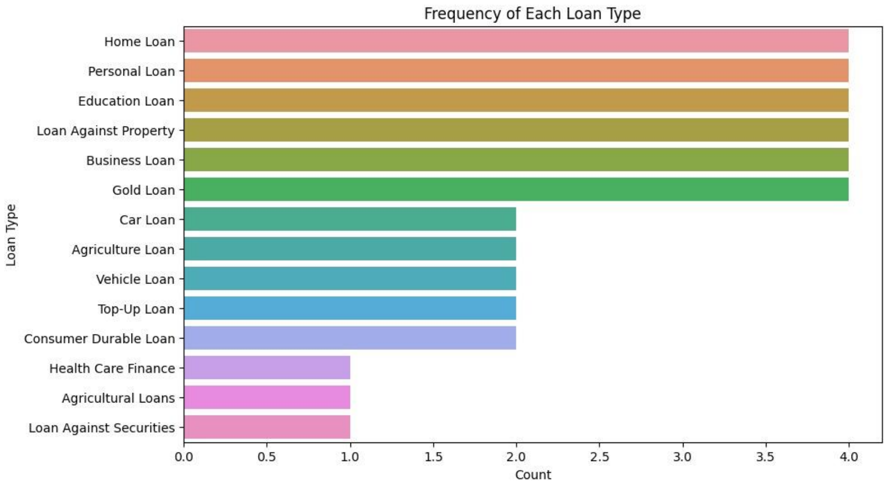


Fig 9: Distribution of Loan Types.

Figure 9 represents a graphical representation of the numerical distribution of the loan types of all the 4 banks. The results are as follows –

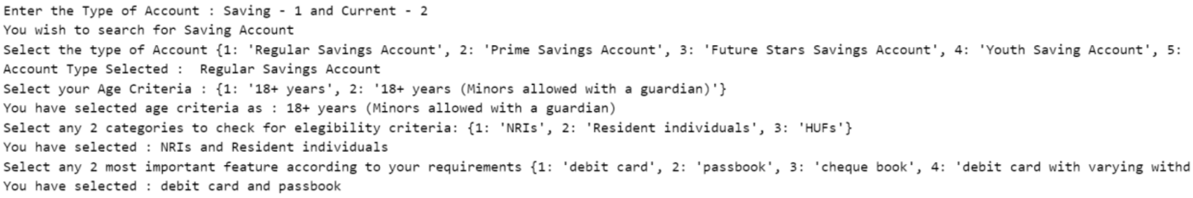
- Home Loan – 4
- Personal Loan – 4
- Education Loan – 4
- Loan Against Property – 4
- Business Loan – 4
- Gold Loan – 4
- Car Loan– 2
- Agriculture Loan – 2
- Vehicle Loan – 2
- Top-Up Loan – 2
- Consumer Durable Loan – 2
- Health Care Finance – 1
- Agricultural Loans – 1
- Loan Against Securities – 1

Recommendation of Bank Accounts –

Inputs taken from user –

- Saving or Current Account
- Type of Account according to the selected preference
- Age Criteria of the user
- Eligibility Criteria selection by the user
- Feature preferences of the user

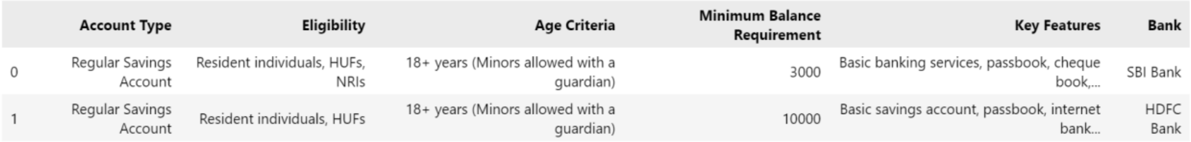
Input/Output –



```
Enter the Type of Account : Saving - 1 and Current - 2
You wish to search for Saving Account
Select the type of Account {1: 'Regular Savings Account', 2: 'Prime Savings Account', 3: 'Future Stars Savings Account', 4: 'Youth Saving Account', 5:
Account Type Selected :  Regular Savings Account
Select your Age Criteria : {1: '18+ years', 2: '18+ years (Minors allowed with a guardian)'}
You have selected age criteria as : 18+ years (Minors allowed with a guardian)
Select any 2 categories to check for elegibility criteria: {1: 'NRIs', 2: 'Resident individuals', 3: 'HUFs'}
You have selected : NRIs and Resident individuals
Select any 2 most important feature according to your requirements {1: 'debit card', 2: 'passbook', 3: 'cheque book', 4: 'debit card with varying withd
You have selected : debit card and passbook
```

Fig. 10: Output snapshot of the input/output results of the algorithm. It displays all the inputs taken from the user and their choice for each question.

Recommended Banks –



	Account Type	Eligibility	Age Criteria	Minimum Balance Requirement	Key Features	Bank
0	Regular Savings Account	Resident individuals, HUFs, NRIs	18+ years (Minors allowed with a guardian)	3000	Basic banking services, passbook, cheque book,...	SBI Bank
1	Regular Savings Account	Resident individuals, HUFs	18+ years (Minors allowed with a guardian)	10000	Basic savings account, passbook, internet bank...	HDFC Bank

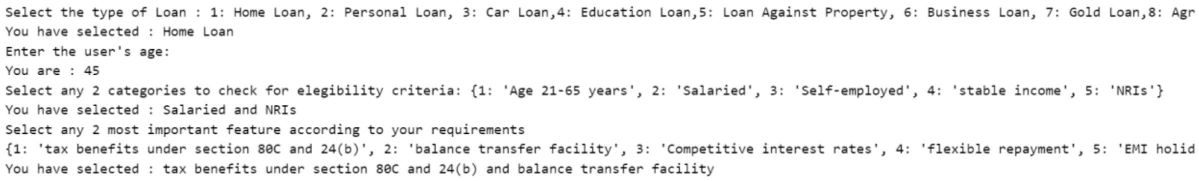
Fig. 11: List of Saving accounts as the final output of the algorithm recommended to the user according to their preferences.

Recommendation of Loan Accounts –

Inputs taken from user –

- Type of Loan according to the user
- Age Criteria of the user
- Eligibility Criteria selection by the user
- Feature preferences of the user

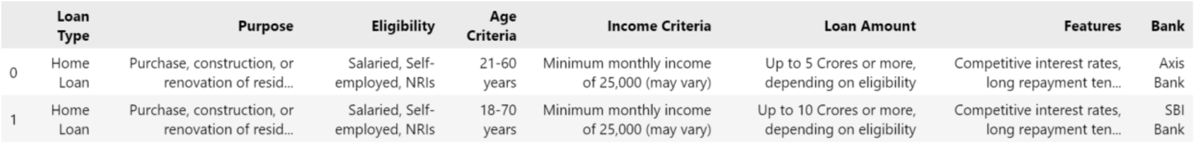
Input/Output –



```
Select the type of Loan : 1: Home Loan, 2: Personal Loan, 3: Car Loan,4: Education Loan,5: Loan Against Property, 6: Business Loan, 7: Gold Loan,8: Agr
You have selected : Home Loan
Enter the user's age:
You are : 45
Select any 2 categories to check for elegibility criteria: {1: 'Age 21-65 years', 2: 'Salaried', 3: 'Self-employed', 4: 'stable income', 5: 'NRIs'}
You have selected : Salaried and NRIs
Select any 2 most important feature according to your requirements
{1: 'tax benefits under section 80C and 24(b)', 2: 'balance transfer facility', 3: 'Competitive interest rates', 4: 'flexible repayment', 5: 'EMI holid
You have selected : tax benefits under section 80C and 24(b) and balance transfer facility
```

Fig. 12: Output snapshot of the input/output results of the algorithm. It displays all the inputs taken from the user and their choice for each question.

Recommended Loans –



	Loan Type	Purpose	Eligibility	Age Criteria	Income Criteria	Loan Amount	Features	Bank
0	Home Loan	Purchase, construction, or renovation of resid...	Salaried, Self-employed, NRIs	21-60 years	Minimum monthly income of 25,000 (may vary)	Up to 5 Crores or more, depending on eligibility	Competitive interest rates, long repayment ten...	Axis Bank
1	Home Loan	Purchase, construction, or renovation of resid...	Salaried, Self-employed, NRIs	18-70 years	Minimum monthly income of 25,000 (may vary)	Up to 10 Crores or more, depending on eligibility	Competitive interest rates, long repayment ten...	SBI Bank

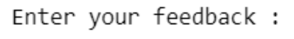
Fig. 13: List of Loan as the final output of the algorithm recommended to the user according to their preferences.

Sentiment Analyzer –

Input taken from user –

- Feedback of the product

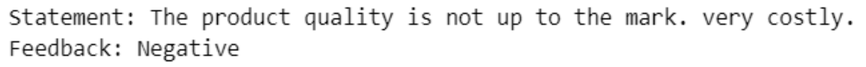
Input/Output –



```
Enter your feedback :
```

Fig. 14: Output snapshot the input/output results of the algorithm. It accepts the feedback from the user as a string for feedback analysis.

Sentiment Analysis for Product Feedback –



```
Statement: The product quality is not up to the mark. very costly.
Feedback: Negative
```

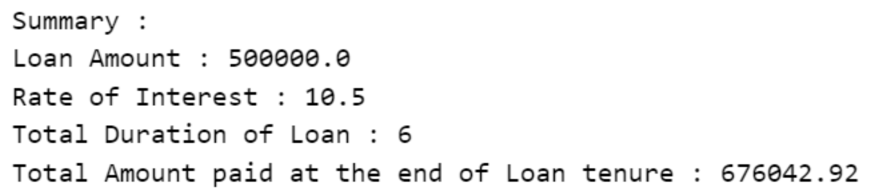
Fig. 15: Output snapshot of the final analysis and classification of the sentiment of the feedback given by the user. The classification is done as positive, negative or neutral.

EMI Calculator –

Input from user –

- Principal Amount
- Duration of Loan
- Rate of Interest

Input/Output –



```
Summary :
Loan Amount : 500000.0
Rate of Interest : 10.5
Total Duration of Loan : 6
Total Amount paid at the end of Loan tenure : 676042.92
```

Fig. 16: Summary snapshot provided as a result of the algorithm to calculate the EMI to be paid by the user after calculation of EMI from the amount, rate of interest and the duration of loan.

Data Visualization of result –

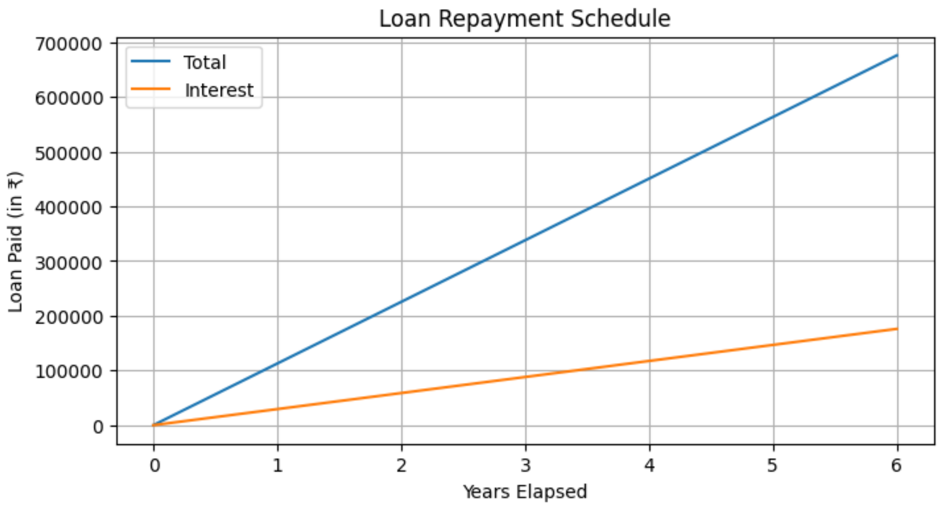
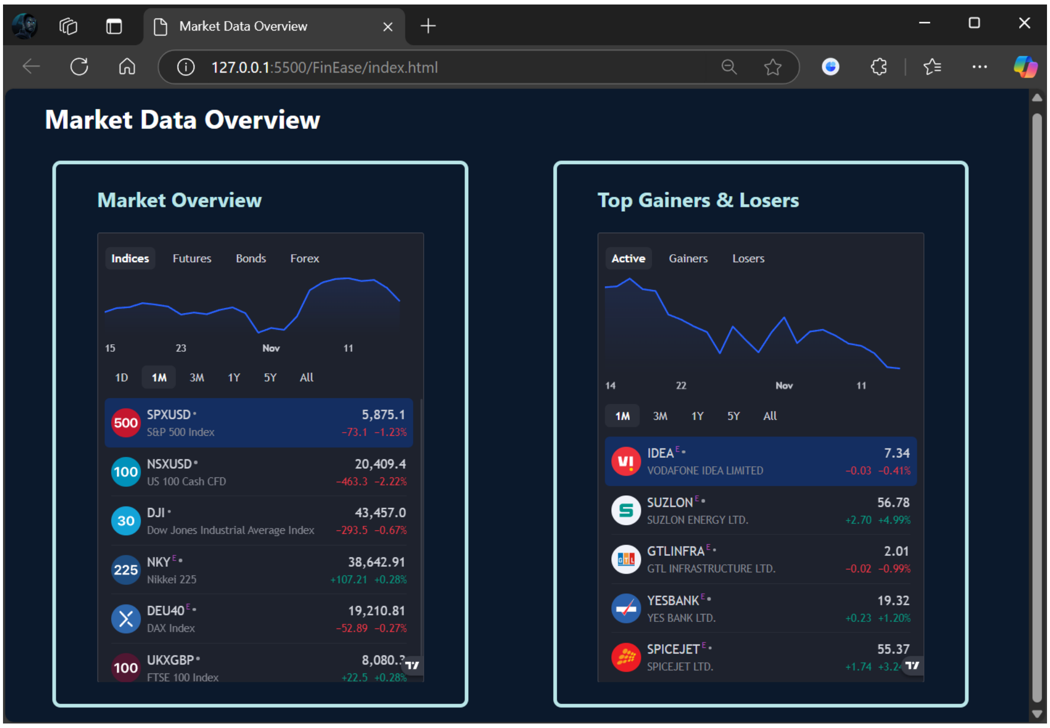


Fig. 17: Graphical representation of the graph formed to display the amount to be paid by the user as EMI at the end of each month. The graph is plotted between Loan paid and years elapsed. Two lines are drawn in the graph, the blue line shows the total amount paid after each year which the yellow line displays the interest amount paid at the end of each year.

Financial Market Trend –

Input/Output



500	SPXUSD S&P 500 Index	5,875.1 −73.1 −1.23%
100	NSXUSD US 100 Cash CFD	20,409.4 −463.3 −2.22%
30	DJI Dow Jones Industrial Average Index	43,457.0 −293.5 −0.67%
225	NKY E Nikkei 225	38,642.91 +107.21 +0.28%
	DEU40 E DAX Index	19,210.81 −52.89 −0.27%
100	UKXGBP FTSE 100 Index	8,080.3 +22.5 +0.28%

IDEA E VODAFONE IDEA LIMITED	7.34 −0.03 −0.41%
SUZLON E SUZLON ENERGY LTD.	56.78 +2.70 +4.99%
GTLINFRA E GTL INFRASTRUCTURE LTD.	2.01 −0.02 −0.99%
YESBANK E YES BANK LTD.	19.32 +0.23 +1.20%
SPICEJET E SPICEJET LTD.	55.37 +1.74 +3.2

Fig. 18: An interactive financial dashboard showing real-time information of markets involved. The Left Panel is an extensive Market Overview with the tabs for Futures, Bonds, and Forex. To its right is Top Gainers & Losers dynamically updated about how the stocks, trends, and percent change.

9. Key Technical Aspects of Project Implementation

(i) Personalized Account Recommendations

a) Technical Diagram & Methodology

The methodology follows as Fig. 19.

Using pandas library, load the 4 dataset of Account Types from 4 different Banks. Combine all the 4 separate dataset into 1 big dataset.

Perform preprocessing on the combined dataset like addition of additional columns, analysing the data types of columns, finding missing data and dealing with it if missing data is found.

Performing data analysis and data visualization for better understanding of the data used for modelling. Using various graphs like bar graph and pie chart to represent the distribution of saving and current account in each bank and distribution of age criteria for accounts (both saving and current) respectively.

Accept user choice for Saving or Current Account, which ever they wish to explore.

Identifying the unique account types from the dataset to help user select the best type of account to initiate the filtering process. Then accepting the account type from user.

Using Keyword Filtering Algorithm to recommend the best suitable account for the user.

Accepting user's age to further filter out the accounts according to their age.

Listing down all the Eligibility criteria for the user from the already filtered out accounts according to their preference and then accepting any 2 eligibility criteria that best fits the user for filtering.

Listing down all the Features for the user from the already filtered out accounts according to their preference and then accepting any 2 features according to the preference of the user for filtering.

Finally displaying all the possible accounts from various banks that best fits the choices of the user and then recommends them.

Mathematical formula used for key word filtration –

Match(a,b) = 1 if a = b else 0 if a!=b - Equation 1

- a and b are string used for matching

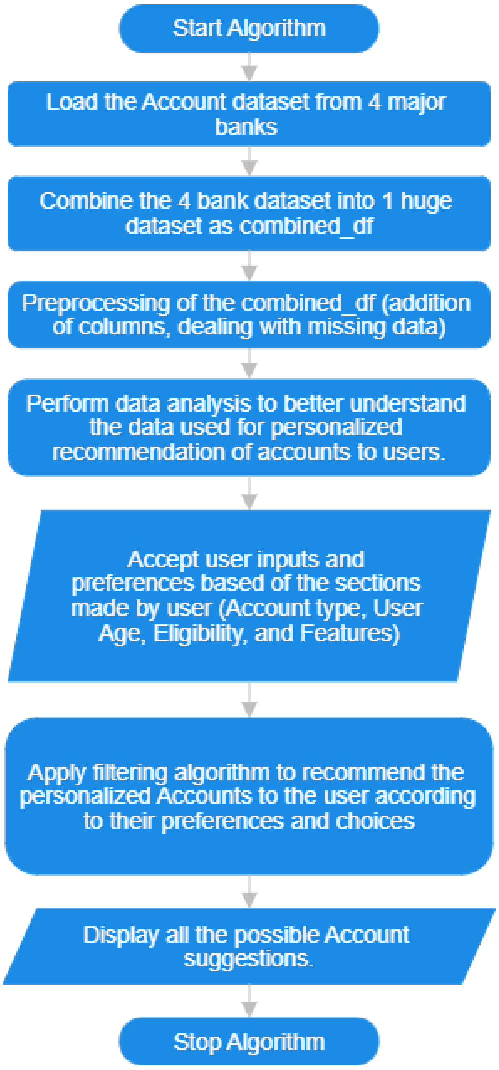

Start Algorithm
Load the Account dataset from 4 major banks
Combine the 4 bank dataset into 1 huge dataset as combined_df
Preprocessing of the combined_df (addition of columns, dealing with missing data)
Perform data analysis to better understand the data used for personalized recommendation of accounts to users.
Accept user inputs and preferences based of the sections made by user (Account type, User Age, Eligibility, and Features)
Apply filtering algorithm to recommend the personalized Accounts to the user according to their preferences and choices
Display all the possible Account suggestions.
Stop Algorithm

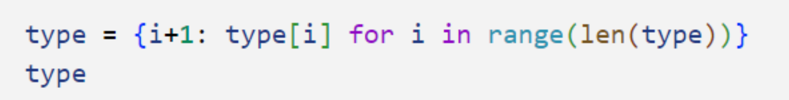
Fig. 19: Flow chart of the methodology of recommendation of personalized account types.

b) Programming Concepts & Functionality

I. Variables Used –

- hdfc_bank – Account dataset of HDFC Bank
- icici_bank – Account dataset of HDFC Bank
- sbi_bank – Account dataset of SBI Bank
- axis_bank – Account dataset of Axis Bank
- combined_df – Combined Account Dataset from 4 Banks
- Account – Final Recommended Banks to the user according to their preferences.

II. Looping Conditions –

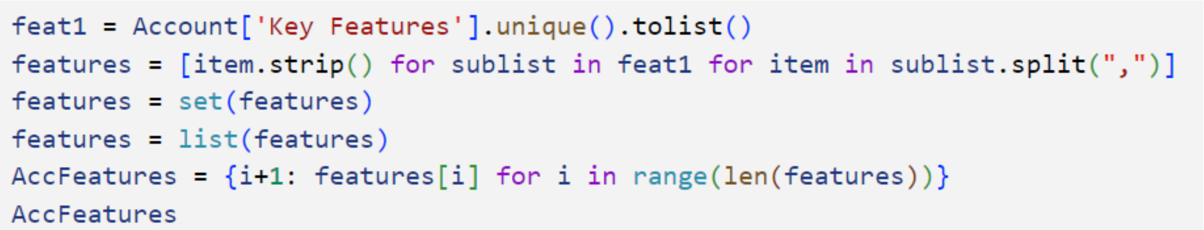


```
type = {i+1: type[i] for i in range(len(type))}
type
```

Fig. 20: Code snapshot for the creation a dictionary of all the type of accounts in the dataset. Mapping each type with an index.

- type – consists of a dictionary of account types from the dataset.

For loop is used as demonstrated in the above code snippet to make a dictionary of all the Account types from the dataset as a key value pair in which key is indexing and value in the account type. This dictionary will be further used to map user input with the actual account type.

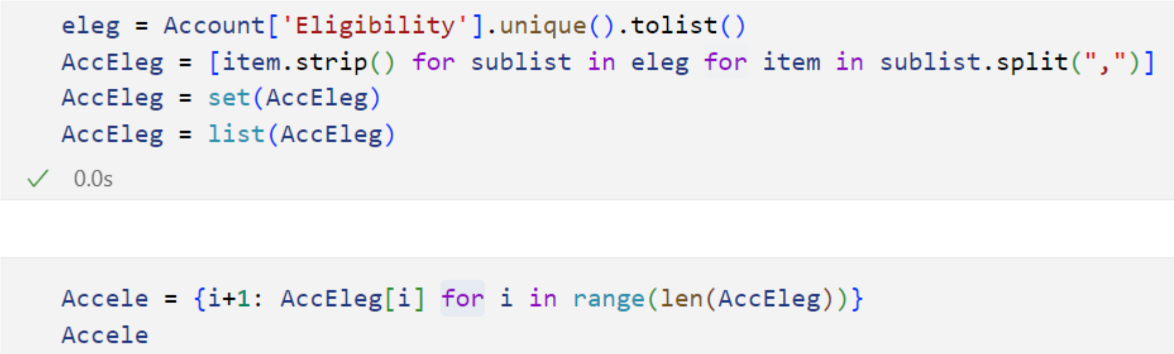


```
feat1 = Account['Key Features'].unique().tolist()
features = [item.strip() for sublist in feat1 for item in sublist.split(",")]
features = set(features)
features = list(features)
AccFeatures = {i+1: features[i] for i in range(len(features))}
AccFeatures
```

Fig. 21: Code snapshot for creation of a dictionary of all the 'Key Features' of the rows selected for the user according to their preferences.

- feat1 – combined list of lists of key features of the entire dataset.
- features – a combined single list of all the features of the dataset.
- AccFeatures – used to map a dictionary of all the features with the given index to easily map users' choice with the actual feature.

For loop is used as demonstrated in the above code snippet to make a list of all the Key Features of the Accounts, combine all the features and treat each feature as a separate functionality which are separated by a comma(,) and then making a dictionary of the list by making a key value pair.



```
eleg = Account['Eligibility'].unique().tolist()
AccEleg = [item.strip() for sublist in eleg for item in sublist.split(",")]
AccEleg = set(AccEleg)
AccEleg = list(AccEleg)
```

✓ 0.0s

```
Accele = {i+1: AccEleg[i] for i in range(len(AccEleg))}
Accele
```

Fig. 22: Code snapshot for creation of a dictionary of all the 'Eligibility' of the rows selected for the user according to their preferences.

- eleg – combined list of lists of eligibility criteria of the entire dataset.
- AccEleg – a combined single list of all the distinct eligibility of the dataset.
- Accele – used to map a dictionary of all the eligibility with the given index to easily map users' choice with the actual eligibility.

Similar to the above process, a list and then dictionary is made for the Eligibility criteria of the accounts.

c) Libraries for Enhanced Functionality

I. Pandas

The library is used in python for loading and working with data, data analysis and data manipulation.

The library is imported as –

import pandas as pd

II. Matplotlib

The library is used in python for visualization and better understanding of the data used for modelling.

The library is imported as –

import matplotlib.pyplot as plt

(ii) Personalized Loan Recommendations

a) Technical Diagram & Methodology

The methodology follows as Fig. 23.

Using pandas library, load the 4 dataset of Loan Types from 4 different Banks. Combine all the 4 separate dataset into 1 big dataset.

Perform preprocessing on the combined dataset like addition of additional columns, analysing the data types of columns, finding missing data and dealing with it if missing data is found.

Performing data analysis and data visualization for better understanding of the data used for modelling. Using various graphs like bar graph and pie chart to represent the distribution of number of loans in each bank and distribution of age criteria for loans respectively.

Identifying the unique loan types from the dataset to help user select the best type of loan to initiate the filtering process. Then accepting the loan type from user.

Using Keyword Filtering Algorithm to recommend the best suitable loans for the user.

Accepting users age to further filter out the loan types according to their age.

Listing down all the Eligibility criteria for the user from the already filtered out loans according to their preference and then accepting any 2 eligibility criteria that best fits the user for filtering.

Listing down all the Features for the user from the already filtered out loans according to their preference and then accepting any 2 features according to the preference of the user for filtering.

Finally displaying all the possible loans from various banks that best fits the choices of the user and then recommends them.

The comparison is based on input from users regarding the type of loan, filtering from a dataset of loans, and matching criteria for each loan. This structured data approach allows for efficient comparison and filtering.

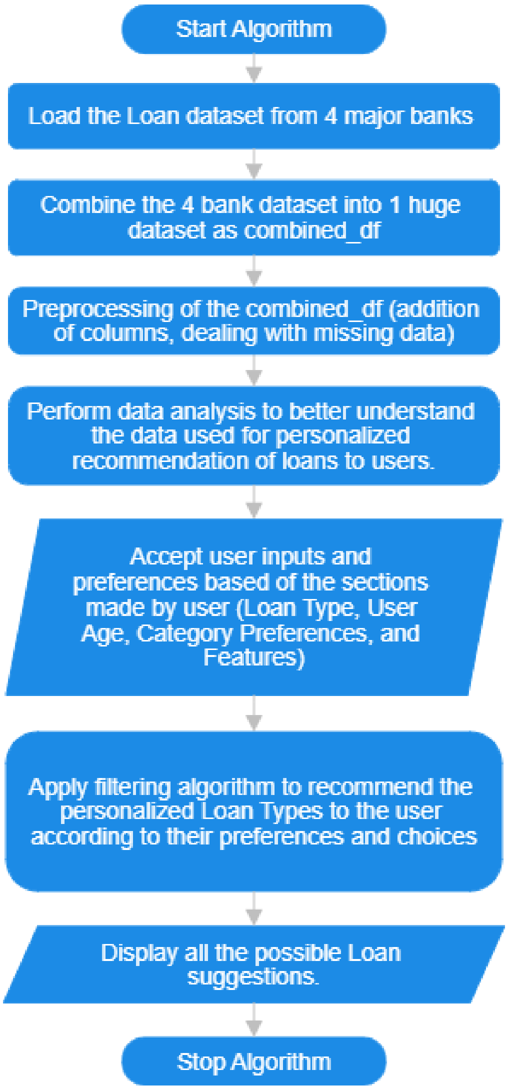

Start Algorithm
Load the Loan dataset from 4 major banks
Combine the 4 bank dataset into 1 huge dataset as combined_df
Preprocessing of the combined_df (addition of columns, dealing with missing data)
Perform data analysis to better understand the data used for personalized recommendation of loans to users.
Accept user inputs and preferences based of the sections made by user (Loan Type, User Age, Category Preferences, and Features)
Apply filtering algorithm to recommend the personalized Loan Types to the user according to their preferences and choices
Display all the possible Loan suggestions.
Stop Algorithm

Fig. 23: Flow chart of the methodology of recommendation of personalized loan types.

b) Programming Concepts & Functionality

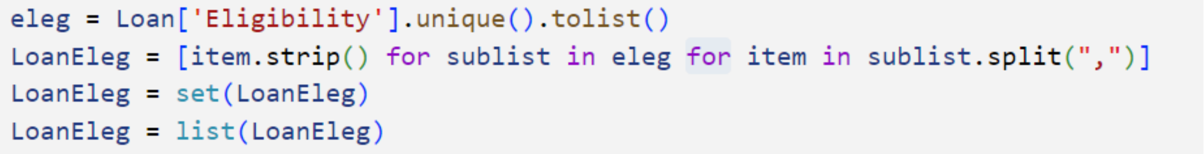
I. Variables Used –

- hdfc_loan – Loan dataset of HDFC Bank
- icici_loan – Loan dataset of HDFC Bank
- sbi_loan – Loan dataset of SBI Bank
- axis_loan – Loan dataset of Axis Bank
- combined_df – Combined Loan Dataset from 4 Banks
- Loan – Final Recommended Banks to the user according to their preferences.

II. Functions –

- age_in_range(row, user_age)
  row – Accept row from the dataset
  user_age – Accept age of the user which is further used for analysing if the age lies in the loans age criteria.

III. Looping Condition –

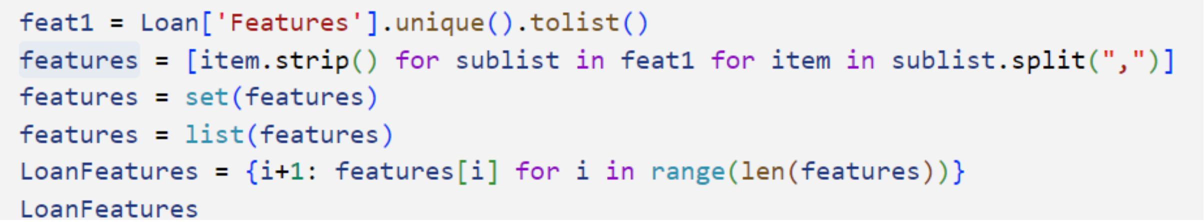


```
eleg = Loan['Eligibility'].unique().tolist()
LoanEleg = [item.strip() for sublist in eleg for item in sublist.split(",")]
LoanEleg = set(LoanEleg)
LoanEleg = list(LoanEleg)
```

Fig. 24: Code snapshot for creation of a dictionary of all the 'Eligibility' criteria of the rows selected for the user according to their preferences.

- eleg – combined list of lists of eligibility criteria of the entire dataset.
- LoanEleg – combined single list of all the distinct eligibility criteria of loans.

For loop is used as demonstrated in the above code snippet to make a list of all the eligibility criteria of the Loans, combine all the criteria and treat each criteria as a separate functionality which are separated by a comma(,).



```
feat1 = Loan['Features'].unique().tolist()
features = [item.strip() for sublist in feat1 for item in sublist.split(",")]
features = set(features)
features = list(features)
LoanFeatures = {i+1: features[i] for i in range(len(features))}
LoanFeatures
```

Fig. 25: Code snapshot for creation of a dictionary of all the 'Features' of the rows selected for the user according to their preferences.

- feat1 – combined list of lists of key features of the entire dataset.
- features – a combined single list of all the features of the dataset.
- LoanFeatures – used to map a dictionary of all the features with the given index to easily map users' choice with the actual feature.

For loop is used as demonstrated in the above code snippet to make a list of all the Key Features of the Loans, combine all the features and treat each feature as a separate functionality which are separated by a comma(,) and then making a dictionary of the list by making a key value pair.

c) Libraries for Enhanced Functionality

I. Pandas

The library is used in python for loading and working with data, data analysis and data manipulation.

The library is imported as –

```
import pandas as pd
```

II. Matplotlib

The library is used in python for visualization and better understanding of the data used for modelling.

The library is imported as –

```
import matplotlib.pyplot as plt
```

(iii) EMI Calculator

a) Technical Diagram & Methodology

The methodology follows as Fig. 26.

Input amount, rate of interest and the duration of loan from the user to calculate Equated Monthly Instalment (EMI).

Two functions are created: calculate_emi() and plot_repayment_schedule().

Each function is called separately to create visual representation of the EMI to be paid per year.

In the calculate_emi() function, predefined formula is used to convert the annual interest rate to a monthly rate, perform the mathematical operations, and calculate the EMI to be paid by the user per month.

In the plot_repayment_schedule(), the interest to be paid each year along with the total amount paid by the end of each year is represented.

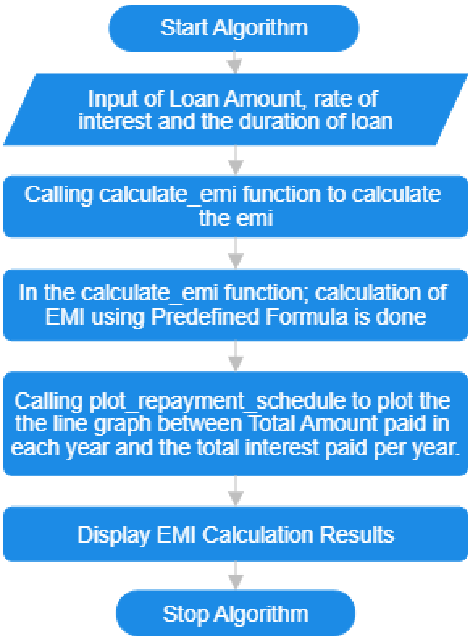

Start Algorithm
Input of Loan Amount, rate of interest and the duration of loan
Calling calculate_emi function to calculate the emi
In the calculate_emi function; calculation of EMI using Predefined Formula is done
Calling plot_repayment_schedule to plot the the line graph between Total Amount paid in each year and the total interest paid per year.
Display EMI Calculation Results
Stop Algorithm

Fig 26: Flow chart of the methodology of EMI Calculator.

Mathematical formula used for emi calculation –

EMI = $(P * r * (1 + r)^n)/((1 + r)^n - 1)$ - Equation 2

- P – principal amount
- r – rate of interest
- n – number of years for the loan

b) Programming Concepts & Functionality

I. Functions used for calculation are –

- calculate_emi(principal, rate, years)
  - principal – Total Loan amount sanctioned to the user.
  - rate – Rate of interest given to the user on loan.
  - years – Total Duration for the loan amount.
  - emi – Total EMI to be paid each month

- plot_repayment_schedule(principal, emi, total_payment, interest_paid, years)
  - principal – Total principal amount
  - emi – EMI amount
  - total_payement – Total sum amount to be paid after the complete duration of loan
  - interest_paid – Total interest paid at the end of each year
  - years – Duration of loan
  - interest_component – Total interest to be paid at the end of each year

II. Guard Conditions –

The principal amount, rate of interest and year duration cannot be less than 0, otherwise the input would be consider invalid.

c) Libraries for Enhanced Functionality

I. Matplotlib

The library is imported in python for better visualization and representation of the result of the EMI Calculator.

The library is imported as –

```
import matplotlib.pyplot as plt
```

plt is an alias for the pyplot submodule of the Matplotlib visualization package

(iv) Sentiment Analyzer

a) Technical Diagram & Methodology

The methodology follows as Fig. 27.

SentimentIntensityAnalyzer is used to Analyze the user feedback as Positive, Negative, or Neutral.

Input is accepted from the user as feedback.

Tokenization of the statement is done to split the statements into its smallest form which is tokens.

SentimentIntensityAnalyzer is initialized and the sentiment score of the statement is calculated.

The compound score is used to classify the sentiment as Positive, Negative, or Neutral:

- Compound ≥ 0.05 → Positive
- Compound ≤ -0.05 → Negative
- Else → Neutral

Mathematical formula used for tokenization –

$T = \{w1, w2, \ldots.., w3\}$ - Equation 3

Mathematical formula used to classify the user feedback as positive/negative/neutral –

Positive Score

$P = \Sigma S'(w_i)$ (for positive words) / Total Tokens - Equation 4

Negative Score

$N = \Sigma S'(w_i)$ (for negative words) / Total Tokens - Equation 5

Neutral Score

U = Count of Neutral Tokens / Total Tokens - Equation 6

- $\Sigma S'(w_i)$: This represents the sum of sentiment scores (S') for all positive or negative words (w_i) in the text.
- Total Tokens: The total number of words or tokens in the text.
- Count of Neutral Tokens: The number of neutral words in the text.

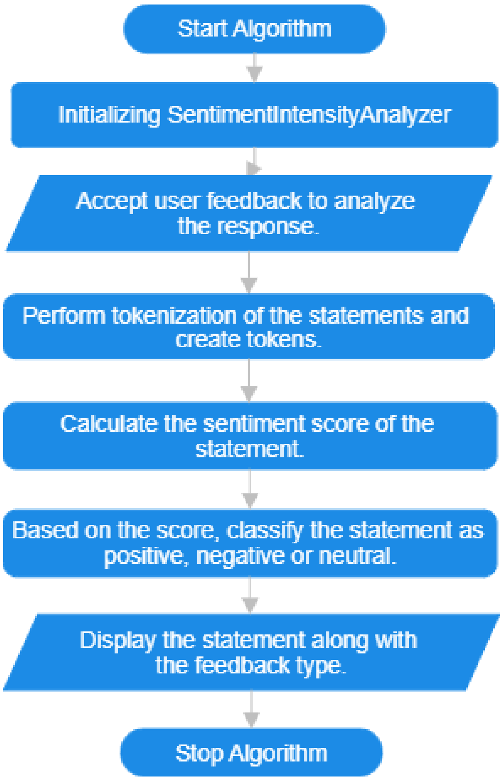

Start Algorithm
Initializing SentimentIntensityAnalyzer
Accept user feedback to analyze the response.
Perform tokenization of the statements and create tokens.
Calculate the sentiment score of the statement.
Based on the score, classify the statement as positive, negative or neutral.
Display the statement along with the feedback type.
Stop Algorithm

Fig 27: Flow chart of the methodology of Sentiment Analyzer.

b) Programming Concepts & Functionality

I. Tokenization –

Tokenization is the process of breaking down a piece of text, like a sentence or paragraph, into smaller units called "tokens." These tokens can be individual words, phrases, or even characters, depending on the approach. It helps computers understand and analyze text by isolating meaningful parts.

II. SentimentIntensityAnalyzer –

The SentimentIntensityAnalyzer() is a tool from the VADER (Valence Aware Dictionary and sentiment Reasoner) library used to analyze the sentiment of text. It assigns scores to text for three sentiment categories: positive, negative, and neutral, and provides an overall compound score (ranging from -1 to +1) to represent the sentiment's intensity.

c) Libraries for Enhanced Functionality

I. nltk toolkit –

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for statistical natural language processing (NLP) for English written in the Python programming language.

II. ntlk.sentiment –

This module in Python is a powerful tool for performing sentiment analysis on text data. It helps assess whether the sentiment of a given text is positive, negative, or neutral.

(v) Real time financial trend

a) Functionality

Fig. 18

- Integrates real-time financial data using TradingView widgets and API.
- Displays:
  - A hotlist widget for top-performing stocks in the BSE (Bombay Stock Exchange).
  - A market overview widget categorizing financial data into Indices, Futures, Bonds, and Forex.
- Provides users with a visual representation of market trends.

b) Use of TradingView API

- Embeds pre-configured TradingView widgets for seamless integration of financial charts.
- Fetches live, up-to-date market data directly from TradingView servers.
- Utilizes asynchronous API script loading for efficient data retrieval without affecting page load speed.

10. SWOT Analysis

Strengths

Advanced Technology: The platform's utilization of optimized ML and NLP ensures accurate and efficient data analysis, enabling personalized financial recommendations.

Scalability: FinEase can handle a large volume of data and users, making it suitable for a wide range of clientele, from individuals to large corporations.

Customization: The platform's ability to tailor financial advice to individual needs and preferences enhances user satisfaction and engagement.

Data Security: Robust security measures can protect sensitive financial data, building.

Weaknesses

Complexity: Implementing and maintaining advanced ML and NLP models can be technically challenging and resource-intensive.

Data Quality: The accuracy of financial recommendations depends on the quality and completeness of the data used, which can be a challenge.

User Adoption: Educating users about the benefits of using AI-powered financial advice may be necessary to drive adoption.

Regulatory Compliance: Adhering to evolving financial regulations can be complex and costly.

Opportunities

Growing Demand: The increasing complexity of financial markets and the need for personalized advice create a significant market opportunity for FinEase.

Integration with Financial Institutions: Partnerships with banks, insurance companies, and investment firms can expand the platform's reach and capabilities.

New Applications: FinEase can be applied to various financial domains, such as wealth management, retirement planning, and risk assessment.

Data Partnerships: Collaborations with data providers can enhance the platform's data quality and insights.

Threats

Competition: The financial technology industry is highly competitive, with established players and emerging startups.

Technological Disruption: Rapid advancements in technology could render existing ML and NLP techniques obsolete.

Economic Downturns: Economic instability can impact user demand for financial advice and investment products.

Regulatory Challenges: Changes in financial regulations can create uncertainties and potential barriers to entry.

12. Conclusion

Through interactive tools and tailored recommendations, the suggested intelligent banking platform uses machine learning (ML) and natural language processing (NLP) to make financial decision-making easier for consumers. To deliver personalized financial advice, the system analyzes user inputs such as bank account information, EMI plans, loan choices, and user attitude analyses. The loan and account modules allow users compare choices side by side by presenting recommendations in a tabular style. Sentiment analysis provides information on banking products by using natural language processing (NLP) to categorize customer input as neutral, negative, or positive.

Users may see the entire amount paid and the length of payback by using the EMI module, which visually depicts loan repayment over time. Predictive analytics and real-time market trend analysis improve investment decision-making by predicting changes in the financial landscape. This user-focused platform seeks to improve decision-making, raise financial literacy, and offer data-driven, user-friendly solutions that help users match their financial behavior with their long-term objectives.

13. References

[1] Anbalagan, G. (2017). New Technological Changes in Indian Banking Sector. *International Journal of Scientific Research and Management*, 5(9), 7015-7021. DOI: 10.18535/ijsrm/v5i9.11

[2] Herrmann, H. (2023). What's next for responsible artificial intelligence: A way forward through responsible innovation. *Heliyon*, 9. https://doi.org/10.1016/j.heliyon.2023.e14379

[3] Al-Mashraie, M., Chung, S.H., & Jeon, H.W. (2020). Customer switching behavior analysis in the telecommunication industry via push-pull-mooring framework: A machine learning approach. *Computers & Industrial Engineering*, 144, 106476. https://doi.org/10.1016/j.cie.2020.106476

[4] Prajwala, T. (2015). A comparative study on decision tree and random forest using R tool. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(1), 196-199.

[5] Zhang, B. (2021). Tactical decision system of table tennis match based on C4.5 decision tree. In *2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 632-635. https://doi.org/10.1109/ICMTMA52658.2021.00146

[6] Singh, T., & Pathak, N. (2000). Emerging role of artificial intelligence in the Indian banking sector. *Journal of Critical Reviews*. https://doi.org/10.55399/WGQO2090