IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 10, October 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 10, October 2024| DOI: 10.15680/IJIRCCE.2024.1210015

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Prediction of Loan Based System Using Machine Learning

Dr.S.Manohar, Tanvi Kalaskar, Sanjana P

Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India

ABSTRACT: The Loan Prediction System aims to enhance the loan approval process by utilizing machine learning models to predict the likelihood of a loan application being approved. By analyzing historical loan data, the system identifies key factors that influence loan approval or rejection. Models such as Logistic Regression, Decision Trees, Random Forest, and Neural Networks are used to provide accurate predictions, reduce processing time, and mitigate risks. The system promotes fairness and scalability, addressing the limitations of manual processing in traditional loan approval systems.

KEYWORDS: Machine learning, loan approval, credit risk, prediction model, decision trees, fairness, automation.

I. INTRODUCTION

The loan approval process is a vital component of financial institutions' operations, as it directly impacts their ability to manage risk and profitability. Efficiently identifying suitable candidates for loans helps financial institutions mitigate the risk of defaults while promoting customer satisfaction. Traditionally, loan approval decisions have been made through manual processes, relying on loan officers' expertise to evaluate applicants based on credit scores, income levels, employment status, and other factors. While this approach has been effective for decades, it is increasingly facing challenges in today's rapidly digitalizing world.

Manual loan evaluations are time-consuming, often taking days or even weeks to complete. Furthermore, the process is prone to human error and subjective biases, leading to inconsistent decisions and potential discrimination against certain groups. Additionally, the increasing volume of loan applications, fueled by the expansion of online banking and digital lending platforms, has put further pressure on financial institutions to streamline their processes without sacrificing accuracy or fairness.

With the advent of machine learning (ML), there is an opportunity to automate and enhance the loan approval process, reducing delays, improving accuracy, and ensuring fairness. Machine learning algorithms have the ability to analyze large volumes of historical data, learn from patterns, and make predictions based on complex, non-linear relationships that may not be easily identifiable by human evaluators. This approach can significantly reduce the time required to process loan applications, eliminate subjective biases, and ensure that loan decisions are based on objective, data-driven criteria.

In this project, we propose a Loan Prediction System that leverages various machine learning algorithms such as Logistic Regression, Decision Trees, Random Forest, and Neural Networks to predict the likelihood of a loan being approved. By utilizing historical data from past loan applicants, the system identifies key factors influencing loan approval outcomes, such as credit history, income, employment status, and demographic information.

The model's primary objective is to provide accurate, consistent, and fair predictions that enable financial institutions to make informed decisions while minimizing risk and improving operational efficiency.

This system is particularly valuable in today's financial landscape, where institutions are faced with the dual challenge of managing increasing workloads and adhering to stricter regulations regarding fairness and transparency in lending. Machine learning models offer a scalable and robust solution that not only reduces operational costs but also ensures that

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

creditworthy individuals are not unfairly denied loans due to human error or bias. Additionally, by offering real-time loan predictions, the system improves customer satisfaction by significantly reducing application turnaround times.

The implementation of a machine learning-based loan prediction system represents a major shift towards more intelligent and automated decision-making processes within financial institutions. By replacing manual evaluations with data-driven algorithms, the proposed system promises to enhance accuracy, reduce processing times, and promote fairness in the lending process. In the following sections, we detail the machine learning models used in this system, the dataset on which they were trained, and the results of the model's performance in predicting loan approvals.

II. METHODOLOGY

The management of loans is very effective and beneficial in that the bank and the applicants know the suitability of the loan, without affecting the costs that directly affect the bank and prevent customers who cannot repay the loan from taking legal action.

Both customers and banks are benefiting from this use of machine learning, and advanced AI-based methods are being used to train models and extract features that help predict bank loans.

Automated systems help banks around the world offer loans and credit cards to customers by calculating credit scores based on spending history, repayment history, and other credit history.

The working time for the application process to be approved and qualified depends on the satisfaction of the job and the work organization, the type of work (public sector, private sector) and other central forces that ensure firm performance and their value to firms.

This article covers the basic features required for loan calculation. One of the main problems in the peer-to-peer (P2P) lending industry is estimating risk. In this case, data is collected from public repositories. The data is pre-processed to improve overall data quality. and select a valid property.

The dataset is divided into training and testing sections. Train machine learning models using training data and model performance is evaluated using matching data. Maturity Forecast Accuracy of Decision Trees, Random Forests, Support Vector Machines, K Neighborhoods, and Mixed Models Using Decision Trees and Adaboost Classifiers Performance and Performance of Total Credit Benchmark Models.

The results provide an overview of the empirical model. Compare your model to existing models to determine its effectiveness. High education standards with integrated education standards.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

III. SYSTEM ARCHITECTURE:

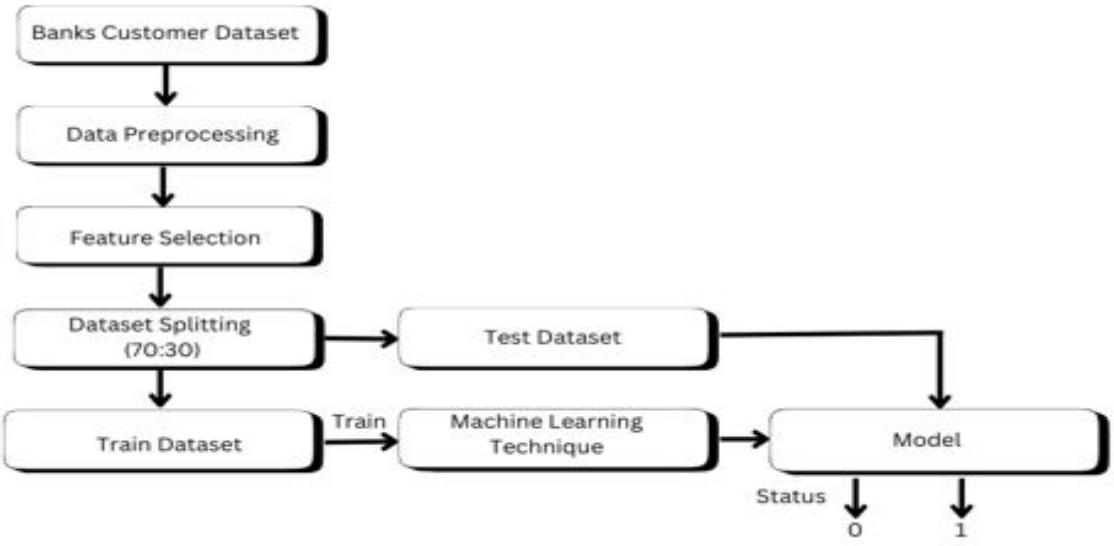

Figure 1: Architecture of Loan Prediction System using Machine Learning

4.1. Elaboration on the Existing Diagram:

- **Banks Customer Dataset:** This is the dataset containing historical information about customers, their financial details, and past loan approvals or denials. The dataset includes features such as income, employment status, credit score, loan amount, loan history, etc.
- **Data Preprocessing:** This step involves cleaning the data by handling missing values, normalizing numeric fields, encoding categorical variables (such as gender, marital status), and removing any outliers. It ensures that the dataset is in a suitable format for training machine learning models.
- **Feature Selection:** In this step, you select the most relevant features (attributes) from the dataset that have the highest impact on loan approval. For example, credit score and income might be more important than other factors. This helps improve model performance by reducing dimensionality and focusing on the most significant variables.
- **Dataset Splitting (70:30):** The dataset is divided into two parts:
- Training Dataset (70%): Used to train the machine learning models.
- Test Dataset (30%): Used to evaluate how well the models perform on unseen data, which helps in assessing the generalization capability of the model.
- **Train Dataset:** This is the portion of the dataset used to build the machine learning models. The training process involves feeding the data to the models and allowing them to learn patterns and relationships.
- **Machine Learning Technique:** This represents the various machine learning models (Random Forest, Naive Bayes, Decision Tree, K-Nearest Neighbors) that are applied to the training data. The models learn from the data and identify patterns to make predictions about loan approval.
- **Model:** This is the final trained model that has learned from the training data and can now predict whether a loan should be approved or rejected based on new input data. The model is then tested using the Test Dataset to evaluate its performance.
- **Status (0 or 1):** The model outputs a binary decision:
- 0: Loan Rejected.
- 1: Loan Approved.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. NOVELTY AND CONTRIBUTIONS OF PROPOSED LOAN PREDICTION SYSTEM

This section outlines the innovative aspects and contributions of the proposed loan prediction system, emphasizing its distinct methodologies and applications compared to existing frameworks.

- **Advanced Data Preprocessing Techniques:** Our system implements advanced techniques for handling missing values, including multiple imputation methods, which enhance the dataset's integrity.
- **Feature Selection**: The model employs an adaptive feature selection algorithm that dynamically adjusts based on data trends, ensuring the most relevant features are prioritized, thus optimizing the predictive accuracy.
- **Machine Learning Model Ensemble:** The proposed system utilizes an ensemble of machine learning algorithms, including Random Forest, Naive Bayes, and Decision Trees, to improve prediction reliability by harnessing the strengths of diverse approaches.
- **Performance Evaluation Metrics:** In addition to standard accuracy measures, the system incorporates advanced metrics like ROC-AUC and precision-recall curves to provide a comprehensive evaluation of model effectiveness.
- **Real-World Applications:** By integrating this model into the loan approval workflow, banks can streamline their decision-making process, reduce default rates, and enhance customer satisfaction by providing quicker responses.
- **Comparison with Existing Systems:** Unlike existing systems that rely solely on traditional statistical methods, our model leverages machine learning techniques that offer better adaptability to evolving customer data patterns.

V. DIFFERENT MACHINE LEARNING ALGORITHMS

5.1 RANDOM FOREST

Using Random Forest Classifier in sklearn. ensemble, the code applies a random forest classifier model to the X_train and y_train datasets. Create a Random Forest Classifier object as rf_clf in this code. The classifier is called using the fitting procedure based on the dataset X_train and the target variable y_train. It then learns patterns and connections between features and target variables by fitting a random forest distribution model to the training data. Running this code will train and prepare the rf_clf object to make predictions on new, never- before-seen data using the prediction method. Always evaluate the performance of the model using test data to determine its generality and make any necessary adjustments. An ensemble learning technique called random forest uses multiple decision trees to produce predictions. It is known for its ability to manage and create complex data, often used to make reliable predictions. Division of labor. Use the training random forest classifier clf model with the provided code to predict the target. Transform and interpret test data X_test Accuracy of prediction. Random Forest Predictors in this code call the rf_clf classifier object. Use the method to pass test data to X_test. this is the Expected value of the target variable using the learned variable Model. Metrics are used to determine how accurate it is. The prediction is accuracy score contrasted with the actual y_test target and expected values y_pred. The proportion of correctly predicted samples is Displayed as an accuracy indicator. The code then displays and prints the desired y_pred value. Actual measurement. Check out Sklearn and its metrics. I imported the module correctly and the dimensions x_test and y_test correspond to the learning model.

5.2 NAÏVE BAYES:

The provided code uses the Gaussian Naive Bayes classifier nb_classifier to predict the target variable for the X_test test data and calculate the accuracy of the prediction. This code calls the prediction method of the Gaussian Naive Bayes classifier object nb_classifier by passing the test data X_test. This uses a learning model to generate the desired value of the target variable. Measurements are used to determine how accurate the prediction is. accuracy_score Compare actual target value y_test with expected value y_pred. The percentage of samples predicted correctly is represented by the correct score.

Finally, the code returns the accuracy estimate, which is a floating point value between 0 and 1, followed by the Gaussian Naive Bayesian Accuracy and accuracy scores. Verify that Sklearn and indicator modules are implemented correctly and that the X_test and y_test dimensions match the learning model. Metrics are used to determine the accuracy of predictions. accuracy_score compares the actual target value y_test with the expected value y_pred. The percentage of correctly predicted samples is displayed as a correct score. The code then specifies the desired y_pred value and returns the correct score.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

5.3 DECISION TREE:

The provided code was placed into a separate decision tree for the X_train and y_train datasets using decision trees from sklearn. tree. However seems that you neglected to provide predicted values for the y_pred variable.

In this code, the Decision Tree Classifier object is created as dt_clf. The classifier is called using the fitting procedure based on the dataset X_train and the variable graph of y_train. This is how a decision tree classifier model is fitted to the data to understand patterns and connections between features and different targets.

The predicted value of the X_test test data is generated using the estimator after the model is trained and stored in the y_pred variable. The estimated value y_pred is printed from the last number. Make x_train and y_train the same size and you have imported the required module (sklearn. tree).

5.4 K-N N (k-Nearest Neighbour):

The given code fits the K-nearest neighbor's classifier to the X_train and y_train datasets using the K Neighbors Classifier from sklearn. neighbors. In this code, the K Neighbors Classifier object is created as kn_clf. The classifier is called using the fitting procedure based on the dataset X_train and the variable graph of y_train. To understand patterns and connections between characteristics and target variables, K-means nearest neighbor classification models are fitted to the data. After running this code, the kn_clf object will be trained and prepared to make predictions on new data, and unpredictable data using the prediction method. Remember to evaluate the model's performance using test data to determine its generality and make any necessary adjustments.

A simple but effective classification technique called K-Nearest Neighbors (KNN) classifies patterns based on neighbors' agreement. The label assigned to the model is determined by the labels of the K nearest neighbors in the training set.
Using the K-Nearest classifier model kn_clf, X_test determine the number you give to predict the target variable for the test data and determine the accuracy of the prediction. This code calls the prediction method of the K nearest neighbor classifier kn_clf to pass the test data X_test. This will use the learning model to generate the desired value of the target variable. Indicators are used to determine the accuracy of the forecast. Accuracy score compares the actual target value y_test with the expected value y_pred. The percentage of correctly predicted samples is expressed as a correct score. The code then specifies the desired y_pred value and returns the correct score. Ensure that the sklearn and metrics modules are implemented correctly and that the X_test and y_test dimensions match the learning model

VI. HARDWARE USED

6.1 Anaconda Navigator:

Anaconda Navigator is a graphical user interface (GUI) included in the Anaconda distribution that allows users to launch applications and manage anaconda packages, environments, and channels without using command-line interface (CLI) commands. It is available for Windows, macOS, and Linux. Navigator provides a number of features that make it easier to work with Anaconda, including:

- A search bar that allows users to find and install packages from the Anaconda Cloud or from local repositories.
- A list of all installed packages, with information about each package, such as its version and dependencies.
- The ability to create and manage environments, which are isolated workspaces that contain specific versions of packages and their dependencies.
- The ability to launch applications, such as Jupyter Notebook, Spyder, and Visual Studio Code, from within Navigator.
- A built-in help system that provides documentation on Anaconda and its features.

Overall, Anaconda Navigator is a powerful tool that can help users to be more productive and efficient when working with Anaconda.
Here are some examples of how Anaconda Navigator can be used:

- A data scientist can use Navigator to create a new environment for a specific project, install the necessary packages, and launch Jupyter Notebook to start working on the project.
- A software developer can use Navigator to manage their Python development environment, install new packages, and launch Visual Studio Code to start coding.
- A machine learning engineer can use Navigator to manage their machine learning environment, install new packages, and launch a terminal to start working on their models.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Anaconda Navigator is a valuable tool for anyone who uses Anaconda, and it is especially useful for beginners and users who are not comfortable using the command line.

6.2 Juypter Notebook:

Jupyter Notebook is an open-source web-based interactive development environment (IDE) for creating and sharing documents that contain live code, equations, visualizations, and narrative text. It is widely used by scientists, data scientists, and machine learning engineers for data cleaning and analysis, data visualization, and model building. Jupyter Notebook documents, also known as Jupyter notebooks, are saved as JSON files with the ipynb extension. They contain markdown cells, code cells, and raw cells. Markdown cells are used for formatted text, code cells are used for Python, R, Julia, or other languages, and raw cells are used for unformatted text. Jupyter notebooks can be run locally or on a remote server. To run a notebook locally,you can use the Jupyter Notebook application. To run a notebook on a remote server, you can use a Jupyter Hub server. Once a notebook is running, you can interact with it by typing code into the code cells and pressing shift+enter to execute the code. The results of the code execution will be displayed below the code cell. Jupyter notebooks are a powerful tool for data science and machine learning. They can be used to explore and clean data, visualize data, build and train machine learning models, and deploy models to production.

Here are some of the benefits of using Jupyter Notebook:

• It is an open-source and free tool.
• It is easy to use and learn.
• It is compatible with many programming languages, including Python, R, Julia, and Scalar
• It supports interactive coding and visualization.
• It can be used to create and share documents that contain live code, equations, visualizations, and narrative text.

Jupyter Notebook is a valuable tool for data scientists and machine learning engineers. It can help them to be more productive and efficient in their work.

VII. SOFTWARE USED

7.1 Python:

Python is a general-purpose programming language that is used for a wide variety of applications, including web development, data science, machine learning, and artificial intelligence. It is one of the most popular programming languages in the world, and is known for its simplicity, readability, and flexibility. Python is a high-level language, which means that it abstracts away many of the low-level details of computer programming, such as memory management and garbage collection. This makes Python easier to learn and use than many other programming languages. Python is also an interpreted language, which means that it does not need to be compiled before it can be run. This makes Python programs easier to develop and debug than programs written in compiled languages Python has a number of features that make it popular for machine learning, including:

• Dynamic computation graphs: Python supports dynamic computation graphs, which means that the model's graph can be constructed and modified while it is running. This makes Python very flexible and allows for a lot of experimentation.
• Pythonic interface: Python has a Pythonic interface, which makes it easy to use for Python developers.
• GPU support: Python supports GPUs, which can significantly speed up the training and inference of deep learning models.
• Large community: Python has a large and active community, which means that there are many resources available to help users get started and troubleshoot problems .Python is used by companies and organizations of all sizes, including Google, Facebook, and Amazon. It is also used by researchers at universities and research labs around the world.

Here are some of the applications of Python:

• Web development: Python can be used to develop web applications using frameworks such as Django and Flask.
• Data science: Python can be used for data science tasks such as data analysis, data visualization, and machine learning.
• Machine learning: Python can be used to develop and train machine learning models using libraries such as TensorFlow and PyTorch.
• Artificial intelligence: Python can be used to develop artificial intelligence applications such as chatbots and virtual assistants.
• Scripting: Python can be used to write scripts to automate tasks.
• Education: Python is a popular language for teaching programming to beginners.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Python is a powerful and versatile programming language that can be used for a wide variety of applications. It is a good choice for beginners due to its simplicity and readability. It is also a good choice for experienced programmers due to its flexibility and performance.

Here are some of the benefits of using Python:

- It is free and open-source software.
- It is easy to learn and use.
- It is versatile and can be used for a wide variety of applications.
- It is scalable and can be used to develop large and complex applications.
- It has a large and active community.

7.2 NumPy:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is a popular library for scientific computing, machine learning, and data science. It is used by researchers, engineers, and students alike to develop and apply scientific and mathematical algorithms**.**

7.3 Pandas:

Pandas is an open-source Python library providing high-performance, easy-to use data structures and data analysis tools for working with structured (tabular, multidimensional, potentially heterogeneous) and time series data. It provides a number of features that make it a popular choice for data scientists. Pandas can be used to prepare data for machine learning models and to evaluate the performance of machine learning models. Pandas is a valuable tool for data scientists and anyone else who works with data in Python. It provides the tools and resources needed to clean, analyze, and visualize data effectively.

7.4 Sklearn:

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project. Scikit-learn is one of the most popular machine learning libraries in Python. It is known for its ease of use, scalability, and performance. Scikit-learn is used by researchers, engineers, and students alike to build and train machine learning models.

7.5 Matplotlib:

Matplotlib is a Python library for data visualization. It provides a wide range of features for creating plots and charts, including line plots, bar charts, scatter plots, histograms, and heatmaps. Matplotlib is also able to create more complex visualizations, such as 3D plots and geographic maps. Matplotlib is a popular choice for data visualization because it is easy to use and produces high-quality results. Matplotlib is also free and open-source software, which makes it accessible to a wide range of users.

7.6 Seaborn:

Seaborn is a Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics. Seaborn is popular among data scientists and machine learning engineers because it makes it easy to create high-quality visualizations with minimal code.

VIII. RESULT (COMPARISION TABLE)

In this research, we got better accuracy than the previous models. Accuracy from different models is listed below:

Table 9.1. Result and accuracy of all the algorithms

S.No	Algorithms	Accuracy

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

1	Random Forest	77.23%
2	Naive Bayes	82.92%
3	Decision Tree	75.6%
4	K-Nearest Neighbour	79.67%

From the above table, we can conclude that the Naive Bayes (NB) Algorithm Gives a Better Accuracy of 82.93%.

IX. CONCLUSION AND FUTURE SCOPE:

We develop and test machine learning (ML) models of open mortgages. To understand the data set and understand the credit approval process we first conduct a data analysis. To solve missing values, we fill them with the appropriate value according to the distribution of the data. We also performed log transformation and scaling to prepare the data for modeling. We then trained and evaluated various classification models, including K-nearest neighbors, decision trees, random forest classifiers, and Gaussian Naive Bayes classifiers. We use accuracy as a metric to evaluate the effectiveness of these models. According to our findings, we found that the naïve bayes classifier outperformed other models and achieved up to 82.93% accuracy on the test set.

Therefore, it can be concluded that the naïve bayes can predict loan approval based on the given features. In future scope the accuracy can be improved more by following methods:

- Adding more data
- Feature selection
- Multiple algorithms
- Algorithm tuning
- Ensemble methods
- Cross validation
- Data preprocessing
- Model selection
- Model assessment

REFERENCES

1) Kumar, Rajiv, et al. (2019). Prediction of loan approval using machine learning. International Journal of Advanced Science and Technology, 28(7), 455-460.
2) Supriya, Pidikiti, et al. (2019). Loan prediction by using machine learning models. International Journal of Engineering and Techniques, 5(2), 144-147.
3) Ashwitha, K., et al. (2022). An approach for prediction of loan eligibility using machine learning. International Conference on Artificial Intelligence and Data Engineering (AIDE). IEEE.
4) Kumari, Ashwini, et al. (2018). Multilevel home security system using Arduino & gsm. Journal for Research, 4.
5) Patibandla, RSM Lakshmi & Naralasetti Veeranjaneyulu. (2018). Survey on clustering algorithms for unstructured data. Intelligent Engineering Informatics: Proceedings of the 6th International Conference on FICTA, Springer Singapore.
6) Tejaswini, J., et al. (2020). Accurate loan approval prediction based on a machine learning approach. Journal of Engineering Science,11(4), 523-532.
7) Sri, K. Santhi & P. R. S. M. Lakshmi. (2017). DDoS attacks, detection parameters, and mitigation in cloud environment. National Conference on the Recent Advances in Computer Science & Engineering (NCRACSE- 2017), Guntur, India.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details