International Journal of Advanced Research and Review

www.ijarr.org

Loan Defaulter Risk Prediction

[1]Shaik Afrin, [2]Reddy Tejaswi, [3]Tatikonda Sathvik, [4]Mr. T. Sesha Sai

[1,2,3]U. G Student, Dept COMPUTER SCIENCE AND ENGINEERING, St.Ann's College Of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

[4]Assistant Professor, COMPUTER SCIENCE AND ENGINEERING, St.Ann's College Of Engineering and Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

ABSTRACT

This project uses a Random Forest Classifier to predict loan default risk, offering a more accurate and flexible alternative to traditional rule-based credit scoring. It analyses borrower specific factors including income, credit score, DTI ratio, and employment details to predict risk more accurately. The predictive analysis is then used to classify applicants into low, medium or high risks categories, helping the loan officers to make more efficient in decision making. A Flask-based web application integrates the trained model, allowing loan officers to input borrower data and receive real-time risk prediction with associated probabilities. Data -preprocessing was performed using pandas and numpy.

KEYWORDS *Loan Default Risk prediction, Machine Learning, Random Forest Classifier, Decision trees, Label Encoding.*

INTRODUCTION

Loan default prediction is an essential task for banks and financial institutions to reduce credit risk and prevent financial losses. Traditionally, loan approvals were based on manual evaluation and credit scoring systems such as Logistic Regression, which often failed to capture complex borrower patterns. To overcome these limitations, this project introduces a Machine Learning-based Loan Defaulter Risk Prediction system that analyses multiple factors such as income, credit score, loan amount, debt-to-income ratio, and employment details to predict whether a borrower is likely

to default. The system uses the Random Forest Classifier for improved accuracy and reliability, supported by Scikit-learn, Pandas, and NumPy for model training and data preprocessing. A Flask-based web application integrates the model, allowing users to input borrower details and receive real-time predictions that classify applicants into low-risk, moderate-risk, or high-risk categories, helping institutions make faster and more reliable lending decisions.

LITERATURE REVIEW

In recent years I explored some related papers to loan defaulter prediction. In that I got several publications. To check papers. I studied recent papers. In that first paper is predicting loan defaulters using logistic regression will be proposed by Serrano- Cinca et al but in this paper, they not explored more about real-time process in based verification. and also taken another paper is credit risk assessment using deep learning developed by Lessman et al but in this paper not used scalable infrastructure for storing of data. finally, I taken paper of loan defaulter prediction using ensemble models by Xia et al but in this paper interpretability process not there for accurate risk assessment. These limitations solved by proposed method.

RELATED WORK

This project uses a Random Forest Classifier to predict loan default risk, overcoming limitations of traditional methods like Logistic Regression and rule- based scoring. It analyzes borrower factors such as income, credit score, DTI ratio, and employment details to classify applicants into low, medium, or high-risk categories. A Flask-based web application integrates the model, allowing real-time risk predictions, Data preprocessing for this project was efficiently handled using libraries like Pandas and NumPy, with Label Encoding used to prepare categorical data for the model. Visual insights, such as credit score distribution charts, were generated using Matplotlib and Seaborn.

EXISTING METHOD

In the existing system, loan default risk is primarily assessed using traditional credit scoring methods and manual evaluations by loan officers. Common approaches include Logistic Regression, Linear Discriminant Analysis (LDA), and rule-based scoring systems, which rely heavily on past credit history and a limited set of financial ratios. These models are generally linear in nature, assuming straightforward relationships between borrower attributes and default risk. As a result, they often fail to capture the complex, non-linear patterns inherent in financial behavior. The evaluation process in these systems is also time-consuming and labour intensive, requiring significant manual intervention. Moreover, such approaches may not adapt well to changing economic conditions or shifts in borrower behavior over time. Their reliance on a few predefined rules can also lead to biased or inaccurate risk assessments.

PROPOSED SYSTEM

The existing system for loan default risk is characterized by traditional methods such as

Logistic Regression and rule-based credit scoring, which are often linear, slow, and inadequate for capturing complex behavioral patterns. The proposed system, conversely, uses a robust Random Forest Classifier model trained with a full array of technologies including NumPy and Pandas for data handling and Label Encoding for categorical features. This model analyses critical factors including income, credit score, DTI ratio, and employment details to accurately classify applicants into low, medium, or high-risk

groups. The entire process is integrated via a Flask web application, utilizing tools like Seaborn for visualizations and Joblib for efficient model loading, ultimately providing instant, faster, and more accurate predictions to streamline and secure the loan approval process.

SYSTEM ARCHITECTURE

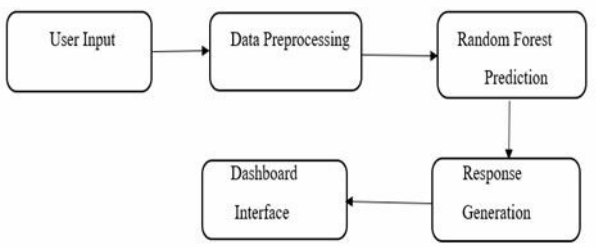

Fig 1: Architecture of the system

METHODOLOGY DESCRIPTION

User Input: The process initiates with User Input, which involves the loan officer utilizing the web- based form to enter the 16 essential applicant features (e.g., Credit Score, DTI Ratio, Employment Type, etc.) for a single borrower. This block, which is the sole source of data for the prediction, captures the information and sends it as a POST request to the Flask backend, triggering the real-time risk assessment process.

Data Preprocessing: Following data capture, the Data Preprocessing block, handled by the Flask application using Pandas, takes over. Its critical function is to clean and transform the raw input into the exact numerical vector required by the machine learning model. This transformation includes applying the fitted encoders (saved during the training phase) to categorical variables, ensuring the data's column order and format perfectly match the model's expected structure for accurate prediction.

Random Forest Prediction: The core engine of the system is the Random Forest Prediction block. It utilizes the pre-trained, loan default model full.pkl model, which is loaded into memory, to receive the pre-processed data vector. The model then executes its complex, ensemble-based logic to calculate the precise probability of default (a score between 0% and 100%), which represents the likelihood that the new applicant will fail to repay the loan.

Response Generation: The Response Generation block performs the crucial step of converting

the raw numerical output from the model into a practical business decision. This involves classifying the default probability into one of the three actionable categories: Low risk, Moderate risk, High risk based on pre- established risk thresholds. Additionally, this stage formulates contextual recommendations or explicit warning messages for the loan officer.

Dashboard Interface: The final block, the Dashboard Interface, represents the output and delivery mechanism. It is responsible for rendering the final result—the Risk Level and Probability—prominently on the web interface. This interface also displays Statistical Visualizations (generated using Matplotlib/Seaborn), such as the Credit Score Distribution chart, to provide contextual data and aid the loan officer in making a swift, well-informed lending decision. This system reduces time and manual effort and leading to make more accurate decisions.

RESULTS & DISCUSSION

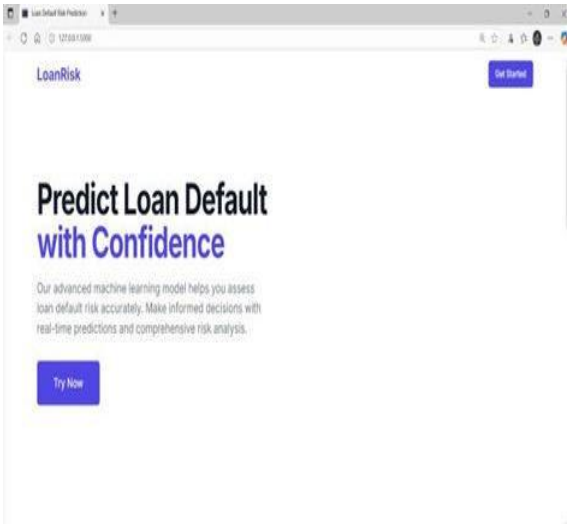

Fig 2: Home Page

This page is a dashboard for a Loan Defaulter Prediction web application. It is designed to provide a user-friendly interface for risk assessment and data visualization.

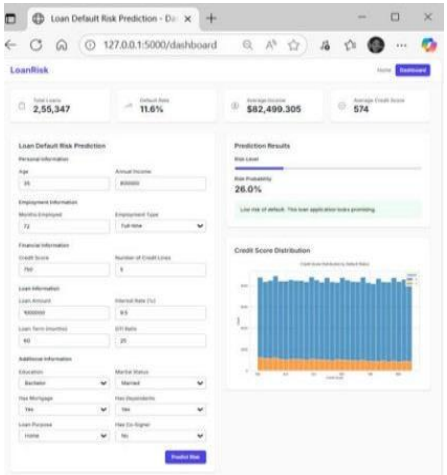

Fig 3: Low-risk profile

The dashboard confirms a highly favorable loan application, predicting a low default risk (26%). The applicant presents a strong profile, being 35 years old with an ₹8,00,000 annual income and a long history of 72 months of full-time employment.

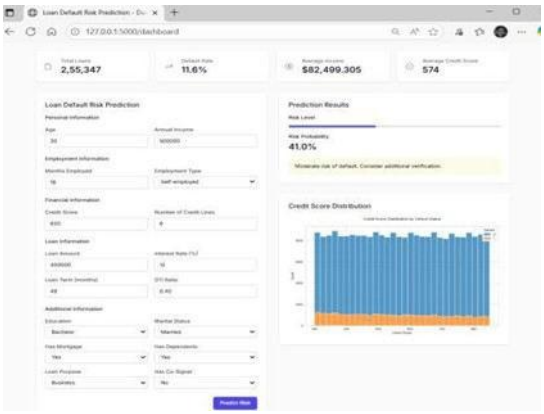

Fig 4: Moderate-risk profile

The model predicts a 41% moderate default risk, strongly recommending that the lender conduct further financial checks and gather supporting documents before granting approval to ensure a safer lending decision.

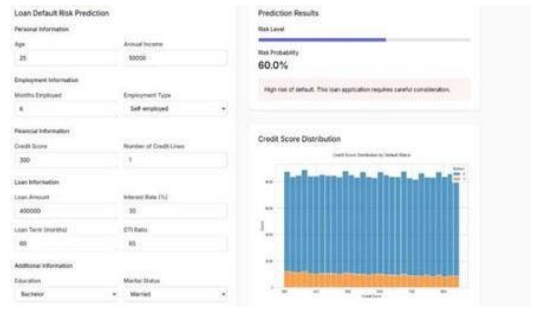

Fig 5: High-risk profile

This loan application presents a severe high-risk scenario due to an array of highly unfavorable factors. The profile is marked by minimal income (₹50,000 annually) and a short 6 months of experience for a 25- year-old self-employed individual.

CONCLSUION

This project successfully developed a robust, data-driven system for loan default risk prediction using a Random Forest

Classifier. The application automates risk assessment, providing lenders with an informed decision-making tool.

FUTURE SCOPE

Future efforts will focus on enhancing the system by adopting Advanced AI Models (XGBoost/Neural Networks) for higher accuracy, integrating Explainable AI (SHAP) for transparency, and ensuring broader access through Cloud & Mobile Deployment.

REFERENCES

[1] Harini, D. P. (2020/7). preliminary scientific research and informative representation of sales data. Juni Khyat ISSN: 2278-4632, 2278-4632.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[3] J. D. Hunter, "Matplotlib: A 2D graphics environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.

[4] M. L. Waskom, "Seaborn: Statistical data visualization," Journal of Open Source Software, vol. 6, no. 60, p. 3021, 2021.

[5] W. McKinney, "Data structures for statistical computing in Python," in Proceedings of the 9th Python in Science Conference, 2010, pp. 51–56.

[6] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau et al., "Array programming with NumPy," Nature, vol. 585, pp. 357– 362, 2020.

[7] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with Python," in Proceedings of the 9th Python in Science Conference, 2010.

[8] M. Grinberg, Flask Web Development: Developing Web Applications with Python. Sebastopol, CA: O'Reilly Media, 2018. [9] Joblib Developers, "Joblib: Running Python functions as pipeline jobs," 2023.

[10] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. Burlington, MA: Morgan Kaufmann, 2011.

[11] T. Fawcett, "An introduction to ROC analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 882–891, 2006.

[12] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[13] Gunicorn Developers, "Gunicorn (Green Unicorn) HTTP for Unix," 2023.

[14] D. J. Hand and W. N. Henley, "Statistical classification methods for credit scoring," Journal of the Royal Statistical Society: Series A, vol. 160, no. 3, pp. 523–541, 1997.

[15] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), 2017.

[16] T. J. Hastie, R. Tibshirani, and J. H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York, NY: Springer, 2009.

[17] S. E. Windecker et al., "Risk stratification using threshold-based classification," European Heart Journal, vol. 37, no. 3, pp. 272–298, 2016.

[18] S. K. Murthy, "Automatic construction of decision trees from data: A multi- disciplinary survey," Data Mining and Knowledge Discovery, vol. 2, pp. 345–389, 1998.

[19] S. R. L. R. K. B. C. Pedregosa et al., "Gunicorn: Production WSGI HTTP server for Python applications," Scientific Programming, 2018.

[20] E. R. Tufte, The Visual Display of Quantitative Information. Cheshire, CT: Graphics Press, 2001.