

AI-Powered Credit Risk Scoring Engine Using Random Forest for Preliminary Loan Applicant Assessment

Mrs. V. Sreevani¹, Ms. K. Saisri², Mr. M. Rajesh³, Mr. K. Pranav Kumar⁴, Mr. M. Sujan Roy⁵

¹Associate Professor, Dept. of Computer Science and Engineering, HITAM, Telangana, India

²³⁴⁵Student, Dept. of Computer Science and Engineering, HITAM, Telangana, India

Abstract - Preliminary credit assessment requires many applicant attributes to be examined together, which is time-consuming and can vary between assessors. This study presents an approach for classifying loan applicants into Low, Medium and High credit risk using a Random Forest model integrated into a complete web application. Applicant attributes covering income, employment, credit history, existing debt, the requested loan, payment behavior and savings are collected through a form, and two financial indicators, the debt-to-income ratio and the installment-to-income ratio, are derived from them. Numerical features are standardized and employment status is one-hot encoded inside a Scikit-learn pipeline. The model is trained on a synthetic dataset of 3,000 records so that no sensitive financial data is used. The system reports class probabilities as confidence and maps each risk category to an Approve, Manual Review or Reject recommendation. A Flask application provides registration, session-based login, customer and administrator roles, assessment history and an administrator dashboard, with data stored in SQLite. The model is evaluated using accuracy, precision, recall and F1-score. On the held-out test set of 600 records, the Random Forest model achieved an accuracy of 69.83 percent and a macro F1-score of 70.32 percent. Because the training data is synthetic, the work is an academic prototype and not a validated credit decision tool.

Key Words: Credit Risk Assessment, Credit Scoring, Machine Learning, Random Forest, Feature Engineering, Debt-to-Income Ratio, Synthetic Dataset, Decision Support.

1. INTRODUCTION

Credit allows individuals and organizations to obtain funds for personal, educational and business needs. Before approving a loan, a lender must judge whether the applicant is likely to repay it within the agreed period. This judgment, called credit risk assessment, normally depends on income, employment, existing debt, repayment behavior, credit history and the requested loan. Checking these factors by hand for every application takes time, and two assessors may reach different conclusions on the same file. A consistent way of giving an initial indication of risk is therefore useful to lending institutions.

Machine learning can support this task because it learns relationships between applicant characteristics and historical outcomes and then applies them to new applicants. A trained model on its own, however, is not a usable tool. Someone has to enter the applicant data, see the result, and have the assessment recorded for later review.

This study presents a credit risk scoring engine that connects a Random Forest classifier to a web application. The system classifies an applicant as Low, Medium or High risk, reports a confidence value and suggests a preliminary action. It is built with Python, Flask, Scikit-learn and SQLite, and is intended as an academic prototype that supports, and does not replace, human underwriting.

1.1 Background and Credit Risk Scoring

Financial institutions process many loan applications containing different kinds of financial and personal information. Credit scoring tries to make this process more systematic by assigning a score or category to each applicant. Traditional methods rely on fixed rules or statistical models. Machine learning offers a different approach in which the model learns the relationships from data.

In this study the applicant is described by fourteen attributes: age, income, employment status, years employed, credit history length, existing loans, existing debt, loan amount, loan term, late payments, credit utilization, bankruptcy history, savings balance and number of dependents. Two derived indicators, the debt-to-income ratio and the installment-to-income ratio, add information about the applicant's financial position. Together these are the inputs to the prediction model.

1.2 Need for an Integrated Credit Risk Scoring System

The need for this system arises from the requirement for efficient, consistent and data-driven preliminary assessment. Specifically, the system aims to reduce the time needed for initial assessment, process several attributes in the same way for every applicant, give early warning of potentially high-risk applications, keep a record of assessments, and give administrators an

overview of submitted assessments.

Financial datasets are also sensitive, which makes academic experimentation difficult. This work therefore uses a synthetic dataset, so the project stays self-contained and does not depend on real personal records.

Instead of making the final lending decision, the proposed system provides a preliminary risk category with probabilities. The prediction supports the assessor, and the final decision still requires further financial, regulatory and human evaluation.

2. LITERATURE REVIEW

Credit risk is the possibility that a borrower fails to meet financial obligations on the agreed terms, so incorrect lending decisions can cause financial loss. Previous work on credit assessment has used manual judgment, fixed rules, statistical scoring models and machine learning classifiers.

2.1 Traditional Credit Scoring Methods

Traditional assessment uses predefined rules, financial ratios and statistical techniques. A lender may set thresholds for income, debt and repayment history and use them to decide whether an application meets the requirements. Altman [1] applied multivariate statistical analysis to financial ratios to predict corporate failure, a well-known example of statistical scoring from financial ratios. Such methods are systematic and mathematically established, but depend on modeling assumptions and data quality. Fixed rules, in turn, have limited ability to represent interactions between variables. Textbooks on statistical credit scoring [2] describe these approaches in detail.

2.2 Machine Learning Approaches for Credit Risk Prediction

Machine learning methods learn patterns from historical data [3] and can capture more complex relationships. Logistic Regression estimates class probability from input variables and is simple and interpretable, particularly when relationships are roughly linear [4], [5]. A single Decision Tree is easy to understand and models non-linear relationships, but may become overly complex and overfit [4], [5]. Support Vector Machines separate classes with a decision boundary and can be effective, but need careful parameter selection and feature scaling [4], [5].

Benchmark studies have compared many classification algorithms on credit scoring datasets [6], [7], which shows that selecting a suitable algorithm and evaluation procedure is an important part of credit scoring research.

2.3 Random Forest for Credit Risk Classification

Breiman [8] introduced Random Forest, an ensemble method that combines many decision trees built on varied samples of the training data and features. The final class is determined from the combined votes of the trees, which generally gives more stable predictions than a single tree. This suits credit risk, because risk depends on several interacting factors. For example, the effect of a given loan amount differs with the applicant's income, existing debt and employment situation.

Random Forest also provides feature importance values, which show which variables the model relies on most. These values describe the model and should not be read as causal relationships [4].

2.4 Review of Existing Approaches

Table -1 summarizes the main approaches.

Table -1: Existing Approaches to Credit Risk Assessment

Approach	Major Advantage	Major Limitation
Manual assessment	Human judgment and contextual analysis	Time-consuming and potentially inconsistent
Rule-based system	Simple and easy to understand	Limited ability to model complex relationships
Statistical scoring	Systematic and mathematically established	Depends on assumptions and data quality
Machine learning	Can model complex relationships	Requires suitable data, validation and monitoring

2.5 Research Gap

Existing work shows that statistical and machine learning methods can be used for credit scoring. However, building a usable prototype needs more than training a classifier. The model must be connected to an application where users submit information, receive predictions and keep records.

The gap addressed here is one of integration. Applicant data collection, financial feature engineering, risk classification, confidence reporting, recommendation mapping, user authentication, prediction history and administrative monitoring each exist separately, but are rarely presented together in one small and reproducible system. The proposed research focuses on this end-to-end workflow. It does not claim a new classification algorithm.

3. METHODOLOGY

The proposed research follows a quantitative and experimental methodology. Applicant records are represented as numerical and categorical features, derived ratios are added, and a Random Forest model is trained and evaluated on them. The prediction pipeline is then integrated into a web application.

The methodology consists of dataset preparation, feature engineering, data preprocessing, model training, risk classification and performance evaluation. The overall workflow of the proposed system is as follows:

Applicant Data → Input Validation → Derived Feature Calculation → Preprocessing → Random Forest Prediction → Risk Category and Recommendation → Database Storage → Result Display

3.1 Research Method

The study uses a quantitative approach because the inputs and the evaluation are measurable. A Random Forest classifier is trained on the processed dataset, and its predictions on held-out data are compared with the known class labels using standard classification metrics. The objective is to determine whether the selected applicant attributes and derived ratios can be used to classify applicants into Low, Medium and High risk, and to show that the model can be used inside a working application.

3.2 System Architecture

The system has four logical layers: User Interface Layer → Flask Application Layer → Prediction Layer → Database Layer. The interface provides registration, login, assessment and result pages. The Flask layer [9] handles requests, authentication, validation and routing. The prediction layer contains the trained pipeline and the feature preprocessing logic. The database layer stores user accounts and assessment records. Keeping these responsibilities separate makes the system easier to maintain and extend.

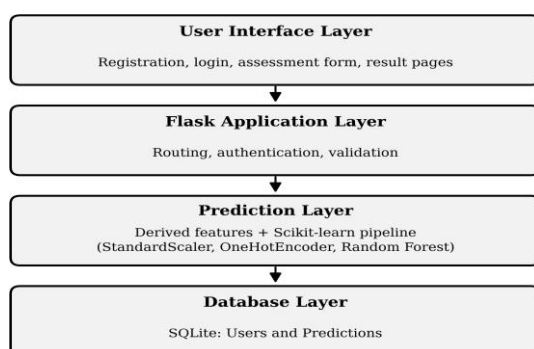


Fig -1: Architecture of the Credit Risk Scoring Engine

3.3 Dataset Preparation

The study uses a synthetic dataset of 3,000 applicant records generated by a script. Synthetic data is used because real credit records contain sensitive personal information. The dataset is divided into training and testing portions. The training portion is used to learn the relationships between applicant features and risk categories, and the testing portion is reserved for evaluating the trained model.

The Low, Medium and High risk labels were assigned using a latent risk score calculated from income, debt-to-income ratio, late payments, bankruptcy history, credit utilization, existing loans, credit history, savings balance, installment-to-income ratio and employment status, together with a random noise component. The 33rd and 70th percentiles of the resulting risk score were used as thresholds: scores below the 33rd percentile were labeled Low, scores between the 33rd and 70th percentiles Medium, and scores above the 70th percentile High. The dataset was generated using a fixed random seed of 42. The final dataset contains 990 Low-risk records (33%), 1,110 Medium-risk records (37%) and 900 High-risk records (30%).

3.4 Input Features

The system uses the applicant attributes listed in Table 2

Table -2: Input Features Used for Credit Risk Classification

Feature	Description
Age	Age of the applicant
Income	Annual income of the applicant
Employment Status	Employment category of the applicant
Years Employed	Number of years employed
Credit History Years	Length of credit history
Existing Loans	Number of existing loans
Existing Debt	Current debt amount
Loan Amount	Requested loan amount
Loan Term	Requested repayment period in months
Late Payments	Number of late payments in the previous year
Credit Utilization	Credit utilization ratio (0 to 1)
Bankruptcy History	Whether a bankruptcy history exists
Savings Balance	Applicant savings

Dependents	Number of dependents
Debt-to-Income Ratio	Derived feature
Installment-to-Income Ratio	Derived feature
Feature	Description
Risk Category	Target class: Low, Medium or High

Savings Balance and Dependents are included as model inputs. The complete feature set consists of fourteen raw applicant attributes and two derived financial ratios, resulting in sixteen input features. Employment Status is treated as the categorical feature, while the remaining fifteen features are processed as numerical inputs.

3.5 Debt-to-Income and Installment-to-Income Calculation

Two ratios are derived from the raw inputs. The debt-to-income ratio (DTI) relates the existing debt and the requested loan to income:

$$DTI = (\text{Existing Debt} + \text{Loan Amount}) / (\text{Income} + 1)$$

The constant 1 in the denominator prevents division by zero. A higher value generally indicates a greater debt burden relative to income. The installment-to-income ratio relates the estimated monthly installment to monthly income:

$$\text{Monthly Income} = \text{Annual Income} / 12$$

$$\text{Monthly Installment} = \text{Loan Amount} / \text{Loan Term}$$

$$\text{Installment-to-Income} = \text{Monthly Installment} / (\text{Monthly Income} + 1)$$

The monthly installment divides the loan evenly over the term and does not include interest. These ratios provide additional information about the applicant's repayment burden.

3.6 Data Preprocessing

The machine learning pipeline uses separate preprocessing for numerical and categorical attributes. Numerical features are processed with StandardScaler, which transforms them to a common scale so that features with large numerical values do not dominate. The employment status attribute is categorical and is converted with OneHotEncoder. Both steps are part of the Scikit-learn pipeline [10], so the same transformation is applied during training and during prediction.

3.7 Machine Learning Model

The proposed system uses the Random Forest classifier as the primary algorithm.

3.7.1 Random Forest

Random Forest is an ensemble algorithm consisting of multiple decision trees. Each tree produces a prediction, and the final class is determined from the combined results. It can handle multiple input features and capture non-linear relationships, and it reduces the risk of overfitting compared with a single tree [8]. In the proposed system it classifies each applicant into Low, Medium or High risk and produces class probabilities.

The Random Forest classifier was configured with 300 decision trees ($n_{\text{estimators}} = 300$), a maximum tree depth of 12 ($\text{max_depth} = 12$), a minimum of 3 samples per leaf ($\text{min_samples_leaf} = 3$), and a fixed random state of 42 ($\text{random_state} = 42$) for reproducibility. Balanced class weights were also used to account for differences in class frequencies.

3.8 Risk Classification and Recommendation

The predicted class is mapped to a preliminary recommendation, as shown in Table -3.

Table -3: Risk Categories and Recommendations

Risk Category	Action	Meaning
Low	Approve	Characteristics are associated with the lower-risk class
Medium	Manual Review	The application needs additional assessment
High	Reject	Preliminary recommendation based on the model prediction

These recommendations are intended for demonstration and decision support. In an actual lending environment, final decisions would need additional financial, regulatory and human evaluation.

3.9 Model Training and Testing

The dataset is divided into training and testing portions. The training portion is used to fit the pipeline, and the testing portion is used to evaluate it. After training, the pipeline is saved with Joblib and loaded by the application for prediction.

The dataset was divided into 80% training data and 20% testing data using a stratified train-test split. Thus, 2,400 records were used for training and 600 records were

reserved for testing. Stratification was applied to preserve the relative distribution of the Low, Medium and High risk classes in both subsets. A random state of 42 was used to ensure reproducibility.

3.10 Performance Evaluation Metrics

The model is evaluated using Accuracy, Precision, Recall and F1-score.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

where TP, TN, FP and FN are true positives, true negatives, false positives and false negatives. Because there are three classes, precision, recall and F1-score were calculated for each risk category and then averaged. Macro averaging was used as the primary aggregate measure, giving equal importance to the Low, Medium and High risk classes regardless of their support. Weighted averages were also obtained from the classification report for reference.

3.11 Web Application, Database and Authentication

The application workflow is as follows. The user registers or logs in, opens the assessment page and enters applicant and loan information. The application validates the values, calculates the derived features and passes the data to the pipeline. The model predicts the class and probabilities, a recommendation is generated, the assessment is stored, and the result is displayed. An administrator can review all assessments on the dashboard.

The application uses a local SQLite database with two entities. The Users entity stores user ID, username, email, password hash, role and account creation time. The Predictions entity stores prediction ID, user ID, applicant attributes, risk category, confidence, recommendation and assessment creation time. One user can have many assessment records.

Authentication is session-based, and passwords are stored as hashes using Werkzeug. A Customer can register, log in, submit assessments and view personal history. An Administrator can access the dashboard and review all assessment records and risk statistics.

4. RESULTS

The Random Forest model achieved an accuracy of

69.83% on the held-out test set of 600 records. The model obtained macro precision of approximately 70.19%, macro recall of approximately 70.46% and macro F1-score of approximately 70.32%, indicating moderate classification performance across the three risk categories.

4.1 Dataset Description

The dataset used in this study is a synthetic dataset generated by a script, containing applicant records for model development.

- Total number of records: 3,000
- Dataset source: synthetic, generated by the dataset generation script
- Input attributes: fourteen raw attributes plus two derived ratios
- Target classes: Low, Medium and High risk
- The dataset contains 990 Low-risk records (33%), 1,110 Medium-risk records (37%) and 900 High-risk records (30%).
- Following the stratified 80:20 split, 2,400 records were used for training and 600 records were used for testing.

4.2 Dataset Feature Analysis

The dataset contains attributes that represent the applicant's income and employment, credit history, existing obligations, the requested loan and payment behavior. The Debt-to-Income Ratio shows the combined burden of existing debt and the requested loan relative to income. The Installment-to-Income Ratio shows the burden of the monthly installment relative to monthly income. Late Payments and Credit Utilization represent payment behavior, and Bankruptcy History indicates a past serious financial event.

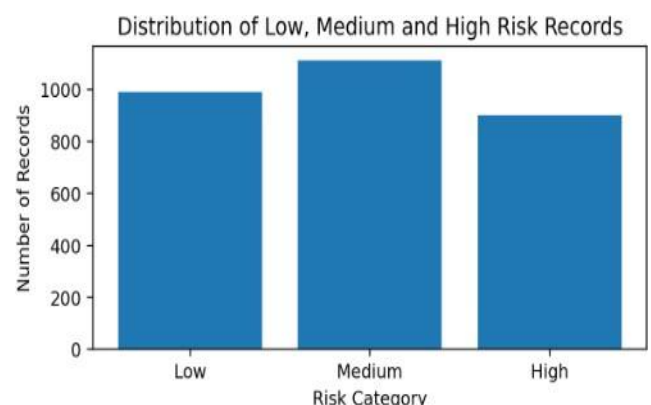


Fig -2: Distribution of Low, Medium and High Risk Records

The dataset has a relatively balanced distribution across the three risk categories, with Medium risk representing

the largest class at 37%, followed by Low risk at 33% and High risk at 30%. This distribution provides representation of all three categories for model training and evaluation.

4.3 Random Forest Results

The Random Forest model was trained on the processed features and evaluated on the test portion. Table -4 gives the overall results, Table -5 the per-class results and Fig -3 the confusion matrix.

Table -4: Random Forest Performance

Evaluation Metric	Result
Accuracy	69.83%
Macro Precision	70.19%
Macro Recall	70.46%
Macro F1-Score	70.32%

Table -5: Per-Class Classification Performance

Risk Category	Precision	Recall	F1-Score	Support
Low	75%	75%	75%	198
Medium	61%	59%	60%	222
High	75%	77%	76%	180

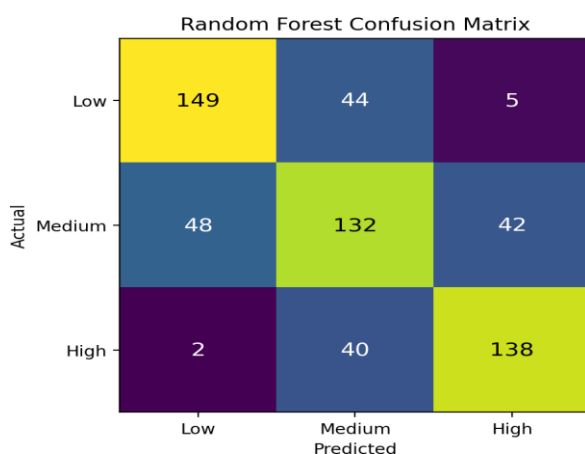
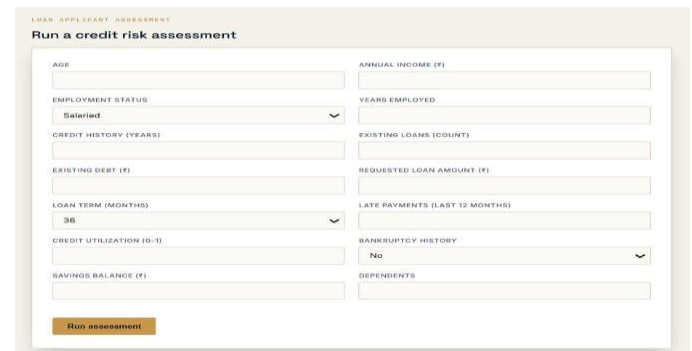


Fig -3: Confusion Matrix of the Random Forest Model

4.4 System Output

The assessment page (Fig -4) collects the applicant attributes. After submission, the result panel (Fig -5) shows the risk category, confidence, recommendation and a probability bar for each class. The administrator dashboard (Fig -6) shows the total assessments,

registered customers, high-risk cases and a table of all assessments.



Form fields include: AGE, ANNUAL INCOME (₹), EMPLOYMENT STATUS (Salaried), YEARS EMPLOYED, CREDIT HISTORY (YEARS), EXISTING LOANS (COUNT), EXISTING DEBT (₹), REQUESTED LOAN AMOUNT (₹), LOAN TERM (MONTHS), LATE PAYMENTS (LAST 12 MONTHS), CREDIT UTILIZATION (0-1), BANKRUPTCY HISTORY, SAVINGS BALANCE (₹), and DEPENDENTS. A 'Run assessment' button is at the bottom.

Fig -4: Credit Risk Assessment Form



Fig -5: Prediction Output with Class Probabilities and Assessment History

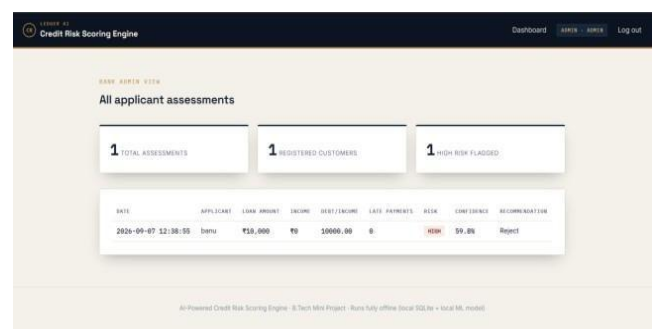


Fig -6: Administrator Dashboard

In the example run shown in Fig -5, an applicant with zero income and a requested loan of ₹10,000 was classified as High risk with a confidence of 59.8%. The class probabilities were High 59.8%, Medium 31.1% and Low 9.1%, and the recommendation was Reject. The administrator dashboard recorded a debt-to-income value of 10,000, which equals the loan amount divided by (0 + 1), so the zero-income case did not cause a division error.

This is a single extreme example that shows the workflow running end to end. It is not an evaluation of

model accuracy.

5. DISCUSSION

The results show that the system can take applicant data through preprocessing, prediction, storage and display in one workflow. The example run confirms that the derived ratios are calculated, the class probabilities are displayed and the record appears in both the customer history and the administrator view.

The evaluation results indicate that the Random Forest model provides moderate performance across the three risk categories, achieving an overall accuracy of 69.83% and a macro F1-score of approximately 70.32%. The Low and High risk classes achieved F1-scores of approximately 75% and 76%, respectively, while the Medium risk class achieved a lower F1-score of approximately 60%. The confusion matrix shows that Medium-risk applicants were more frequently confused with both Low-risk and High-risk applicants, with 48 Medium cases predicted as Low and 42 predicted as High. Similarly, 44 Low-risk cases were classified as Medium. These results indicate that distinguishing the intermediate Medium-risk category is more challenging for the model than identifying the Low and High categories.

The confidence value of 59.8% in the example suggests that the model was not certain even for an extreme input. This shows why a Manual Review path exists, since a prediction with low confidence should be checked by a person.

5.1 Performance Analysis

The Random Forest model achieved an accuracy of 69.83%, with macro precision, recall and F1-score of approximately 70.19%, 70.46% and 70.32%, respectively. The per-class results show stronger performance for the Low and High risk categories than for the Medium category. In particular, the Medium class obtained a recall of approximately 59%, indicating that a considerable number of Medium-risk applicants were assigned to another risk category. Since the dataset is synthetically generated, these results primarily demonstrate the performance of the implemented classification pipeline on the generated data and should not be interpreted as evidence of performance on real-world lending data.

Because the labels come from a synthetic generator, the model has learned the relationships built into that generator and not real lending behavior. High scores on such data would therefore show that the pipeline works, and would not show that it predicts real defaults.

5.2 Importance of the Selected Features

The debt-to-income and installment-to-income ratios describe repayment burden relative to income, which is central to credit assessment. Late payments and credit utilization describe payment behavior, and credit history length and years employed describe stability. Together these describe the applicant's obligations, behavior and stability, which are the factors the model uses to estimate risk. The Random Forest feature importance analysis identified late payments, debt-to-income ratio, income and installment-to-income ratio as the most influential features in the trained model. Their respective importance values were approximately 0.1571, 0.1269, 0.1101 and 0.0926. Credit history length and credit utilization also contributed substantially, with importance values of approximately 0.0764 and 0.0610, respectively. Savings balance and loan amount had importance values of approximately 0.0575 and 0.0511 (Table -6). These importance values describe the contribution of features to the trained Random Forest's predictions and should not be interpreted as causal relationships.

Table -6: Random Forest Feature Importance

Feature	Importance
Late Payments	0.1571
Debt-to-Income Ratio	0.1269
Income	0.1101
Installment-to-Income Ratio	0.0926
Credit History Years	0.0764
Credit Utilization	0.0610
Savings Balance	0.0575
Loan Amount	0.0511

5.3 Practical Significance of the Proposed System

The system can give an initial, consistent indication of risk and can flag potentially high-risk applications early. It can also keep assessment records and provide an administrator overview. It runs locally without external services or sensitive data. In practice it supports the assessor and does not replace full underwriting.

5.4 Limitations

- The training data is synthetic, so the results do not show real-world predictive performance.
- Only one model is used.
- The installment calculation does not include interest.
- The prototype includes a seeded demonstration

administrator account, which must be removed or changed before real use.

- Fairness and bias across applicant groups have not been examined.

6. CONCLUSION

Credit decisions carry financial risk, and checking each application by hand is slow and can be inconsistent. A system that gives an early and consistent indication of risk can support this process.

This study presented an artificial intelligence (AI)-powered credit risk scoring engine that combines a Random Forest classifier with a web application. Applicant attributes and two derived financial ratios were used to classify applicants as Low, Medium or High risk. The model was trained on a synthetic dataset of 3,000 records, and predictions were reported with confidence and a recommendation. The application provided authentication, customer and administrator roles, assessment history and an administrator dashboard.

On the held-out test set of 600 records, the Random Forest classifier achieved an accuracy of 69.83%, with macro precision of approximately 70.19%, macro recall of approximately 70.46% and macro F1-score of approximately 70.32%. The results demonstrate the feasibility of integrating the Random Forest model into an end-to-end credit risk assessment application, while the synthetic nature of the dataset limits conclusions about real-world lending performance.

The system estimates risk for preliminary assessment and does not make the final lending decision. Because the data is synthetic, the work shows a working workflow, and further validation on real data is required before any real-world use.

6.1 Future Scope

The proposed work can be extended in several ways. The system can be trained and evaluated on larger, more diverse and real-world datasets, with cross-validation and comparison with other classifiers. Hyperparameter tuning and more advanced algorithms can be explored. A fairness analysis across applicant groups can be added, together with explainability methods that show the factors behind each prediction. The system can also be extended with graphical dashboards and reports, continuous model monitoring and updating, and integration with banking or financial databases.

REFERENCES

- 1) E. I. Altman, "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy," The

- Journal of Finance, vol. 23, no. 4, pp. 589–609, 1968.
- 2) L. C. Thomas, D. B. Edelman, and J. N. Crook, Credit Scoring and Its Applications. SIAM, 2002.
- 3) J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2011.
- 4) T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2009.
- 5) G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: With Applications in R, 2nd ed. Springer, 2021.
- 6) B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring," Journal of the Operational Research Society, vol. 54, no. 6, pp. 627–635, 2003.
- 7) S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring: An Update of Research," European Journal of Operational Research, vol. 247, no. 1, pp. 124–136, 2015.
- 8) L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- 9) Pallets Projects, "Flask Documentation." [Online]. Available: <https://flask.palletsprojects.com/>. Accessed: Sep. 30, 2026.
- 10) F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.