CAPSTONE PROJECT

LOAN APPROVAL PREDICTION

PRESENTED BY

STUDENT NAME: KUMAR MAHESH KORUPROLU

COLLEGE NAME: RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, SRIKAKULAM

DEPARTMENT: CSE

EMAIL ID: MAHESHKORUPROLU06@GMAIL.COM

AICTE STUDENT ID: STU679C7437C55951738306615

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Financial institutions face the critical challenge of making fast, accurate, and fair decisions regarding loan approvals. Traditionally, loan officers assess an applicant’s eligibility based on income, credit history, employment status, and other demographic factors, which is time-consuming and prone to human bias or error.

To streamline this process, there is a need for an automated system that can predict the likelihood of a loan being approved based on historical data. The objective is to build a machine learning model that analyzes applicant information and accurately predicts whether a loan application should be approved or not. This solution will help lenders make more consistent and data-driven decisions, improve customer experience, and reduce operational time and risk.

PROPOSED SOLUTION

- **Data Collection & Understanding**
  - Use historical loan datasets containing features like gender, income, employment status, credit history, loan amount, and education level.
  - Explore relationships between features and the loan approval outcome.
- **Data Preprocessing**
  - Handle missing values, inconsistent formats, and categorical variables through encoding.
  - Perform feature scaling and outlier handling to ensure model performance.
- **Feature Selection & Engineering**
  - Select relevant features contributing most to the prediction.
  - Engineer new features if needed, such as loan-to-income ratio or combined income
- **Model Building**
  - Train multiple machine learning models (e.g., Logistic Regression, Random Forest, XGBoost).
  - Use cross-validation and hyperparameter tuning to improve model accuracy.
- **Model Evaluation**
  - Evaluate models using accuracy, precision, recall, F1-score, and ROC-AUC.
  - Select the best-performing model based on validation performance.
- **Model Deployment**
  - Save the final model using .pkl for future predictions.
- **Continuous Monitoring & Improvement**
  - Monitor model predictions and retrain periodically with updated data.
  - Incorporate feedback to improve fairness and reduce bias in approvals.

SYSTEM APPROACH

The Loan Approval Prediction System follows a systematic approach to ensure accurate, scalable, and interpretable predictions. This section outlines the tools, libraries, and methodology used in developing the machine learning pipeline.

- **System requirements**
  - Minimum: 4 GB RAM, Intel i3 Processor
  - Recommended: 8+ GB RAM, Intel i5/i7 or equivalent
  - Disk space: 500 MB for dataset and model files
- **Library**
  - pandas
  - numpy
  - matplotlib, seaborn
  - sklearn
  - joblib

ALGORITHM & DEPLOYMENT

- **Algorithm Selection**
  - For this binary classification problem, we experimented with multiple supervised learning algorithms including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN).
  - Among these, Random Forest Classifier performed the best with an accuracy of 80.5%, making it the chosen model for deployment. It was selected due to its ability to handle both categorical and numerical variables, resist overfitting, and deliver consistent performance across imbalanced datasets.
- **Data Input:**
  - Applicant Income, Coapplicant Income,Loan Amount,Loan Amount Term,Credit History,Gender,Marital Status,Education
- **Training Process:**
  - The dataset was split into training and testing sets using an 80/20 split.
  - Cross-validation and grid search were used to fine-tune hyperparameters such as the number of estimators and max depth in the Random Forest model.
  - The model was evaluated using metrics such as Accuracy, Precision, Recall, F1 Score, and ROC AUC to ensure balanced performance.
  - The final model was serialized using the joblib library and saved as a .pkl file for later use.
- **Prediction Process:**
  - Once trained, the model takes new applicant data as input and predicts whether the loan will be approved (Yes) or not (No). The prediction is based on patterns learned from the historical data. Key influencing factors like credit history and income are weighted heavily in the decision-making process.. For real-time predictions, inputs are passed to the saved model either through a command-line script, a web interface (e.g., using Flask or Streamlit), or an API service, which returns the prediction immediately.

RESULT

The Loan Approval Prediction System was evaluated using a range of performance metrics to assess its accuracy and effectiveness. Among the models tested, the Random Forest Classifier achieved the highest accuracy of 80.5%, outperforming Logistic Regression, Decision Tree, SVM, and KNN in overall prediction quality.

To ensure fair evaluation, the dataset was split into training and testing sets. The Random Forest model demonstrated strong generalization on the test data, with a Precision of 31.25%, Recall of 17.91%, and ROC AUC score of 67.17%, indicating a reasonable trade-off between false positives and false negatives.

A confusion matrix was used to visualize correct vs. incorrect classifications, and ROC curves were plotted for all models to compare their ability to distinguish between approved and non-approved loans. The Random Forest model showed the highest area under the ROC curve, confirming its superiority.

The results indicate that the model can serve as a reliable decision-support tool for automating and assisting in the loan approval process. With further tuning and the inclusion of more diverse data, its predictive power can be improved even further.

CONCLUSION

The Loan Approval Prediction system successfully demonstrates how machine learning can assist in automating and improving the efficiency of the loan approval process. Among the various models evaluated, the Random Forest Classifier delivered the best performance with an accuracy of 80.5%, highlighting its effectiveness in capturing complex relationships within the data. The project showed that incorporating features such as applicant income, loan amount, credit history, and employment details plays a crucial role in determining loan approval outcomes.

During implementation, challenges such as handling missing values, class imbalance, and selecting optimal hyperparameters were addressed through data preprocessing, feature engineering, and model tuning. These steps significantly contributed to the improved model performance.

While the results are promising, there is room for improvement. Enhancing the dataset with more diverse and detailed features, integrating external financial indicators, and using advanced ensemble techniques or neural networks could further boost prediction accuracy. Overall, the system has the potential to support financial institutions by providing quick, data-driven decisions, reducing manual workload, and ensuring fairer and more consistent evaluations.

FUTURE SCOPE

In the future, the loan prediction model can be improved by incorporating more detailed user data, such as credit scores, bank statements, and employment history. This will increase the model's ability to make accurate and fair decisions.

Advanced machine learning techniques like XGBoost or deep learning can be explored to further boost prediction accuracy. These models can handle more complex patterns and provide better generalization.

The system can also be deployed in real-time using cloud platforms, allowing instant loan eligibility checks for users. This would improve the speed and efficiency of the loan approval process.

Adding explainability tools like SHAP or LIME can help banks understand why a loan was approved or rejected, which builds trust and transparency.

Finally, the model can be expanded to support multiple regions, languages, and regulatory frameworks, making it suitable for broader, real-world use.

REFERENCES

GitHub repository: Link

Thank you