KHWAJA FAREED
UEIT
RAHIM YAR KHAN

Faculty of
Computer Sciences &
Information Technology

Loan Approval System

Group Members:

THE PROJECT IS DEVELOPED & PRESENTED BY:

1. FAWAD AHMED BAIG (COSC221101049)
2. SHAHBAZ AHMAD MAZARI (COSC221101037)
3. MUHAMMAD TAYYAB (COSC221101033)
4. RAHEEL ADEEL (COSC221101013)
5. ABDULLAH TAHIR (COSC221101083)

Submitted to:

Sir Shahzad Hussain

December 12, 2024
COSC-5A
Machine Learning

Loan Approval System

1. Introduction The Loan Approval System is a machine learning (ML) project designed to automate and streamline the process of evaluating loan applications. By leveraging historical data and advanced ML algorithms, this system predicts whether a loan should be approved or rejected, improving decision-making efficiency and reducing manual errors.

2. Objective The primary objective of the Loan Approval System is to assist financial institutions in assessing the risk of loan applicants and automating the decision-making process. The system aims to:

- Reduce manual effort and biases in loan approval.
- Enhance the accuracy of approval predictions.
- Improve the overall efficiency of the loan processing workflow.

3. Features

- **Data Preprocessing**: Handles missing values, encodes categorical variables, and scales numerical features.
- **Model Training**: Trains multiple ML models to identify the best-performing algorithm.
- **Prediction**: Provides binary outcomes (approve/reject) or probability scores for loan approval.
- **Explainability**: Uses feature importance techniques to explain model predictions.

- **User Interface**: An intuitive dashboard for data input, predictions, and insights.

4. Dataset

- **Source**: A publicly available dataset or proprietary data from financial institutions.
- **Features**: Includes attributes such as applicant income, credit history, loan amount, loan term, and property area.
- **Target Variable**: Loan approval status (Approved/Rejected).

5. Technologies Used

- **Programming Language**: Python
- **Libraries**: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- **Framework**: Flask for web interface
- **Tools**: Jupyter Notebook, Visual Studio Code

6. System Architecture

- **Data Collection**: Importing and cleaning the dataset.
- **Data Preprocessing**: Handling missing values, encoding categorical variables, and feature scaling.
- **Model Building**: Experimenting with multiple algorithms such as Logistic Regression, Decision Tree Classifier, Random Forest Classifier, K-Neighbours Classifier etc.
- **Evaluation**: Measuring model performance using metrics like accuracy, precision, recall, and F1 score.

- **Deployment**: Deploying the final model using a web based interface.

7. Model Performance

- The selected model achieved an accuracy of 89%, with precision and recall scores of 93%% and 86%, respectively.
- Confusion matrix and ROC-AUC curve visualizations wer used to assess performance.

8. Deployment The Loan Approval System can be deployed on:

- Local machines for testing purposes.
- Cloud platforms (AWS, Azure, or Google Cloud) for scalability and accessibility.
- Web interface or API for real-time predictions.

9. Challenges Faced

- **Data Quality**: Managing missing or inconsistent data entries.
- **Data Imbalance**: Addressed using techniques like random over sampler.
- **Model Overfitting**: Mitigated using cross-validation and regularization techniques.

10. Future Enhancements

- Incorporating additional features such as credit scores and employment details.
- Extending the system to handle multi-class predictions (e.g., high, medium, low risk).
- Adding a recommendation engine to suggest alternative loan terms for rejected applications.
- Enhancing the user interface for better usability and visualization.

11. Conclusion The Loan Approval System demonstrates how machine learning can improve traditional financial workflows by automating decision-making and enhancing prediction accuracy. With additional data and refinements, the system has the potential to become a valuable tool for financial institutions.

Experiment 1: (Imbalanced Data)

Logislic Regression:

Accuracy = 77%

Decision Tree Classifier

Accuracy = 70%

Random Forest Classifier

Accuracy = 77%

KNeighbors Classifier

Accuracy = 71%

Exp 2: (Balanced Data)

Logistic Regression:

Accuracy = 69%

Decision Tree Classifier

Accuracy = 79%

Random Forest Classifier

Accuracy = 89%

KNeighbors Classifier

Accuracy = 72%

Selected Model:-

Random Forest Classifier with 89% accuracy.

Reason:-

We analyzed that the training data were imbalanced because the recall for 1 was 96% but for 0 was 43% this indicates that the data was imbalance and the model trained in exp1 was biased towards predicting '1' as compared to 0.

Solution: (Conclusion Report)

We used RandomOverSampler to overcome the imbalacement of the data.

After balancing the data, we achieved 89% accuracy with Random Forest Classifier model.