

Loan_prediction

Loan Repayment Prediction

Introduction

The goal of this project is to predict loan repayment outcomes using machine learning. This project specifically implements a Random Forest model, trained on user attributes such as age and recent financial inflows, to determine whether a user will default or repay a loan.

The project also includes a graphical user interface (GUI) that allows for easy interaction, enabling the user to input User ID and view the predicted loan outcome.

Data Overview

The data used for this project was sourced from a PostgreSQL database, consisting of the following key tables:

user_attributes: Contains user-specific features such as age and cash inflows over the last 30 days.

loan_outcomes: Captures the loan outcome for each user, indicating whether the loan was repaid or defaulted.

Key Features Used:

Age: The user's age at the time of loan application.

Cash Inflow (30 days): The sum of financial inflows (in KES) from SMS transactions for the user in the last 30 days.

Model Selection

The model used for this project is the Random Forest Classifier. Random Forest was chosen for its ability to handle complex datasets and perform well in classification tasks.

Model Training:

The data was split into training and testing sets using an 80/20 ratio.

Features such as age and cash_incoming_30days were used for training.

The target variable was the loan outcome, which is binary: "repaid" (1) or "defaulted" (0).

Performance Metrics:

Accuracy: The model was evaluated on its accuracy using the test set.

Confusion Matrix: A confusion matrix was used to visually inspect the model's classification performance.

Graphical User Interface (GUI)

A Tkinter-based GUI was developed to allow easy interaction with the model. The user can input a User ID and the system will:

- Display the user's age and cash inflow for the last 30 days.

- Predict whether the loan will be repaid or defaulted.

- Show the probabilities for both outcomes.

- Allow the user to reset the input fields for subsequent predictions.

Key GUI Features:

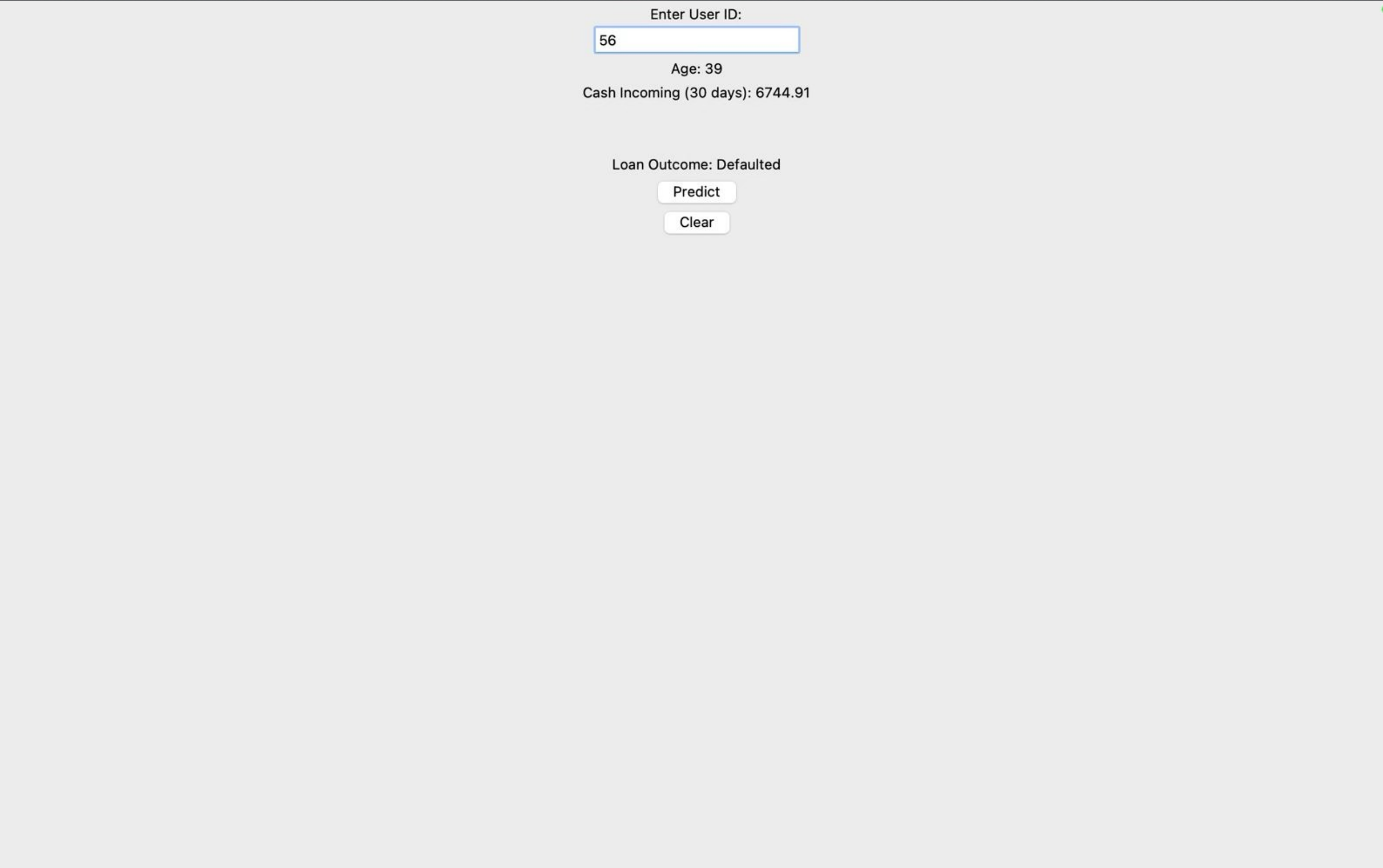
User-Friendly Input: Users can enter a valid User ID and receive a prediction in real-time.

Outcome Probabilities: Probabilities for both “repaid” and “defaulted” outcomes are displayed.

Reset Functionality: A clear button resets the interface for fresh inputs.

Model Evaluation

To evaluate the model, we calculated the accuracy score and used a confusion matrix to compare predicted outcomes against the actual outcomes in the test dataset.



The screenshot shows a web application interface for loan outcome prediction. It features a light gray background with a white rectangular area containing the input fields and buttons. At the top, there is a label "Enter User ID:" followed by a text input field containing the number "56". Below this, the user's "Age: 39" and "Cash Incoming (30 days): 6744.91" are displayed. The predicted "Loan Outcome: Defaulted" is shown in a light blue box. At the bottom, there are two buttons: "Predict" and "Clear".

Enter User ID:
56
Age: 39
Cash Incoming (30 days): 6744.91
Loan Outcome: Defaulted
Predict
Clear

Accuracy:

The model achieved an accuracy of [Insert Accuracy Score] on the test set, showing that it performs well in predicting loan outcomes.

Confusion Matrix:

Below is the confusion matrix representing the model's performance:

True Positives: Correctly predicted repaid loans.

True Negatives: Correctly predicted defaulted loans.

False Positives: Predicted loans as repaid when they were defaulted.

False Negatives: Predicted loans as defaulted when they were actually repaid.

[Insert Confusion Matrix or Graph Here]

Conclusion

This loan prediction project successfully implements a Random Forest classifier to predict loan outcomes based on user attributes. The project also includes a graphical interface to make the system user-friendly. The model's high accuracy demonstrates that it can effectively predict whether a loan will be repaid or defaulted, helping lenders make informed decisions.

Future Work:

Feature Expansion: Incorporate additional features like GPS data to improve prediction accuracy.

Model Optimization: Experiment with hyperparameter tuning to further improve model performance.

Advanced Visualizations: Integrate more advanced visualizations in the GUI to present user data and model outputs.

References

Scikit-learn: Machine Learning in Python

Tkinter: Python GUI Development

Thank you