CREDIT RISK PREDICTION

PROBLEM STATEMENT: Predicting Credit Risk for Loan Applicants

INTRODUCTION:

Credit risk assessment is a crucial process for financial institutions to determine whether a loan applicant is likely to repay their loan or default. Traditionally, this assessment relied on manual evaluation of an applicant's financial background, but with the growth of data and machine learning techniques, automated and more accurate prediction models have become possible.

This project focuses on building a machine learning-based system to predict the credit risk of loan applicants using the German Credit Data. The goal is to classify applicants into two categories:

- Good Credit Risk (likely to repay)
- Bad Credit Risk (high risk of default)

The project involves:

- Cleaning and preprocessing the raw dataset,
- Engineering a new target variable (Risk),
- Training a Random Forest Classifier with hyperparameter tuning,
- Evaluating model performance with proper metrics,
- And creating a Streamlit web application where users can input applicant details and instantly get a credit risk prediction.

OBJECTIVE:

Develop a machine learning model to predict the credit risk of loan applicants using the German Credit dataset. The model should classify applicants into two categories: good credit risk and bad credit risk. Additionally, provide insights into the key factors influencing credit risk and suggest strategies for improving the credit evaluation process.

METHODOLOGY:

1. Data Collection

- **Dataset Used**: German Credit Data.
- **Description**: The dataset contains information about loan applicants, including attributes like age, job type, housing situation, savings, checking account status, loan amount, duration, and loan purpose.

2. Data Preprocessing

- **Dropping Unnecessary Columns**:
  - The column 'Unnamed: 0' (index column) was removed because it did not add any predictive value.
- **Handling Missing Values**:
  - Missing values in 'Saving accounts' and 'Checking account' columns were filled using the mode (most frequent value) of each column.
- **Creating the Target Variable (Risk)**:
  - Instead of using an existing target, a new binary target was engineered:
    - If Credit Amount > 4000, labeled as Bad Risk (0).
    - If Credit Amount ≤ 4000, labeled as Good Risk (1).
- **Dropping 'Credit Amount'**:
  - After creating the Risk variable, 'Credit amount' was dropped to avoid data leakage.
- **Encoding Categorical Variables**:
  - Applied Label Encoding on categorical columns: Sex, Housing, Saving accounts, Checking account, and Purpose.

- **Feature Scaling**:
    - Used StandardScaler to standardize feature values to have zero mean and unit variance.

3. Model Development

- **Feature and Target Separation**:
    - Features (X) and Target (y) were separated after preprocessing.
- **Train-Test Split**:
    - Dataset split into 80% training and 20% testing.
    - Used stratification to ensure the target variable distribution remains the same in both sets.
- **Handling Imbalanced Classes**:
    - The class_weight='balanced' parameter was used in Random Forest to handle class imbalance automatically.
- **Model Chosen**:
    - Random Forest Classifier was selected due to:
        - Its ability to handle both categorical and numerical features.
        - Robustness to overfitting.
        - Good performance on tabular data.
- **Hyperparameter Tuning**:
    - Used GridSearchCV to find the best combination of hyperparameters:
        - n_estimators (number of trees)
        - max_depth (maximum depth of trees)
        - min_samples_split (minimum samples to split a node)
        - min_samples_leaf (minimum samples at a leaf node)
    - 3-Fold cross-validation was used during Grid Search.

4. Model Evaluation

- **Metrics Used**:
    - **Accuracy**: Percentage of correct predictions.
    - **Precision**: How many predicted "Good Risk" applicants were actually good.
    - **Recall**: How many actual "Good Risk" applicants were correctly predicted.
    - **F1-Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**:
    - Plotted to visually observe True Positives, True Negatives, False Positives, and False Negatives.
- **Feature Importance**:
    - Identified which features contributed most to the model’s decision-making (e.g., Duration, Job Type, Purpose).

5. Model Saving

- The final best model was saved using **Pickle** (credit_risk_model.pkl) so that it could be reused without retraining.

6. Web App Deployment (Streamlit)

- **Frontend**:
    - Built a user-friendly interface using **Streamlit**.
    - Users can select values for Sex, Job, Housing, Savings, Checking account, Credit Amount, Duration, and Purpose.
- **Input Processing**:
    - User inputs are preprocessed in the same way as the training data to maintain consistency (e.g., label encoding, column arrangement).

- **Prediction**:
    - When the user clicks the "Predict Credit Risk" button:
        - The processed input is passed to the trained model.
        - The app displays either:
            - "Good Credit Risk"
            - "Bad Credit Risk"
- **Error Handling**:
    - Ensured feature count matches expected input format to avoid prediction errors.

.

RESULT:

After building and training the Random Forest model on the German Credit Dataset, the model achieved the following performance metrics on the test set:

- Accuracy: 0.8300
- Precision: 0.9333
- Recall: 0.8344
- F1-Score: 0.8811

CONCLUSION:

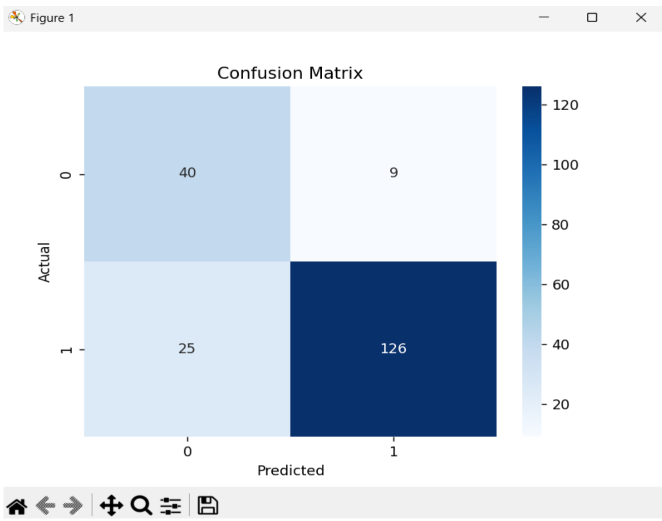
This project successfully developed an AI-powered Credit Risk Prediction System using a Random Forest Classifier.

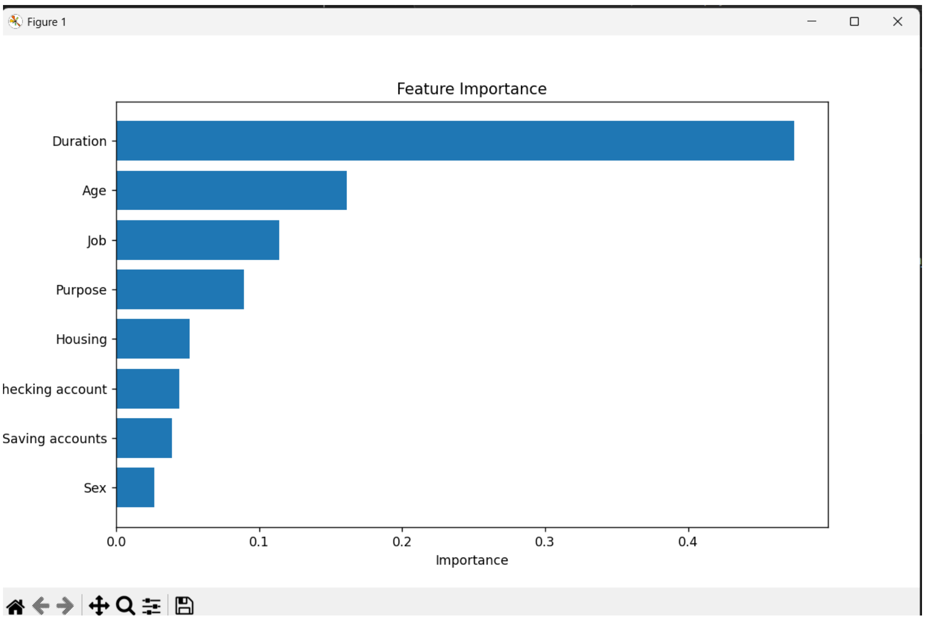
The Implemented approach was selected because:

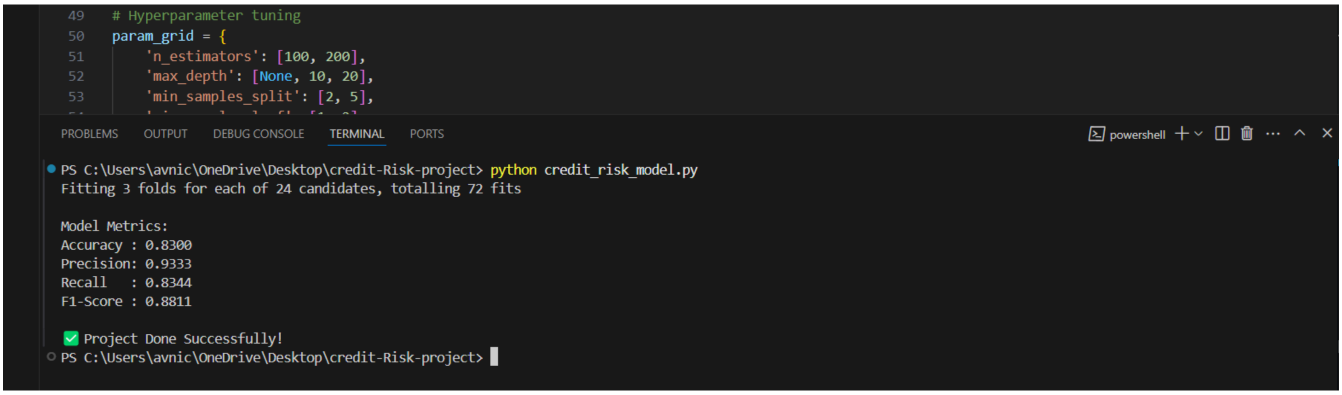
- Random Forest handled mixed data types (categorical + numerical) very well.
- It also managed class imbalance (more Good risk applicants than Bad) by setting class_weight='balanced'.
- Feature Engineering (creating a Risk column from Credit Amount) made the problem more meaningful and focused.

- Hyperparameter Tuning improved the model's performance without overfitting.
- Deployment via Streamlit made the solution user-friendly, allowing easy input and real-time predictions.

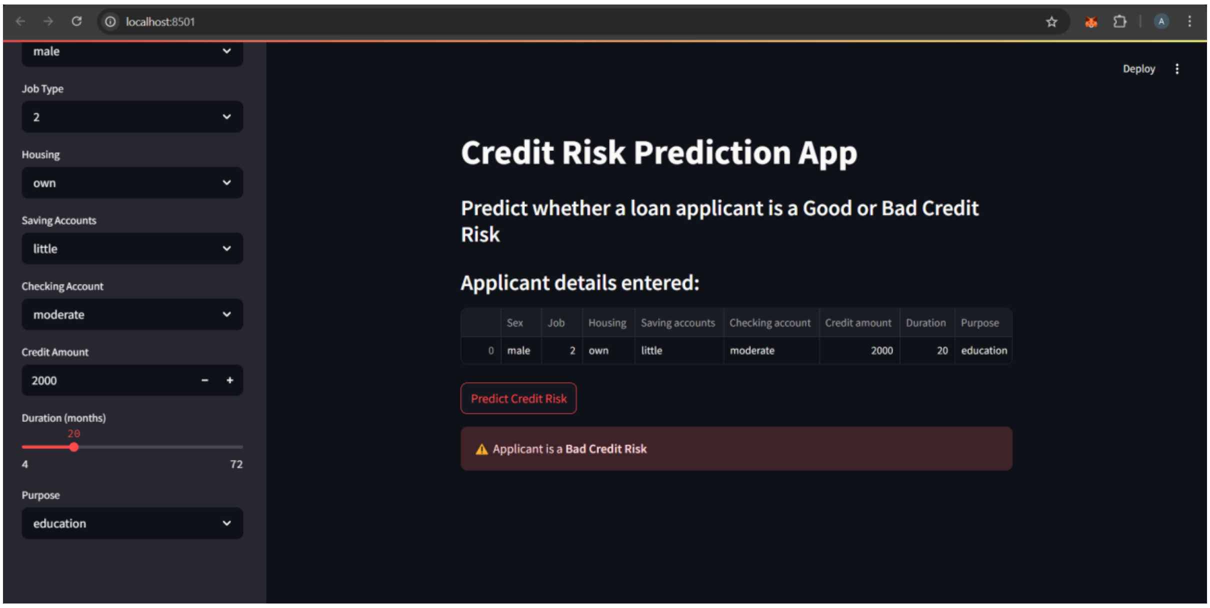
Thus, the selected methodology proved to be accurate, reliable, and practical for real-world credit risk prediction tasks.







# Hyperparameter tuning
param_grid = {
'n_estimators': [100, 200],
'max_depth': [None, 10, 20],
'min_samples_split': [2, 5],
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS C:\Users\avnic\OneDrive\Desktop\credit-Risk-project> python credit_risk_model.py
Fitting 3 folds for each of 24 candidates, totalling 72 fits
Model Metrics:
Accuracy : 0.8300
Precision: 0.9333
Recall : 0.8344
F1-Score : 0.8811
Project Done Successfully!
PS C:\Users\avnic\OneDrive\Desktop\credit-Risk-project>



localhost:8501
male
Job Type
2
Housing
own
Saving Accounts
little
Checking Account
moderate
Credit Amount
2000
Duration (months)
20
4
72
Purpose
education
Deploy
Credit Risk Prediction App
Predict whether a loan applicant is a Good or Bad Credit Risk
Applicant details entered:
Sex Job Housing Saving accounts Checking account Credit amount Duration Purpose
0 male 2 own little moderate 2000 20 education
Predict Credit Risk
Applicant is a Bad Credit Risk