**Project Credit Risk**

**Project Overview**

This project focuses on predicting **credit risk** using **machine learning techniques**, specifically:

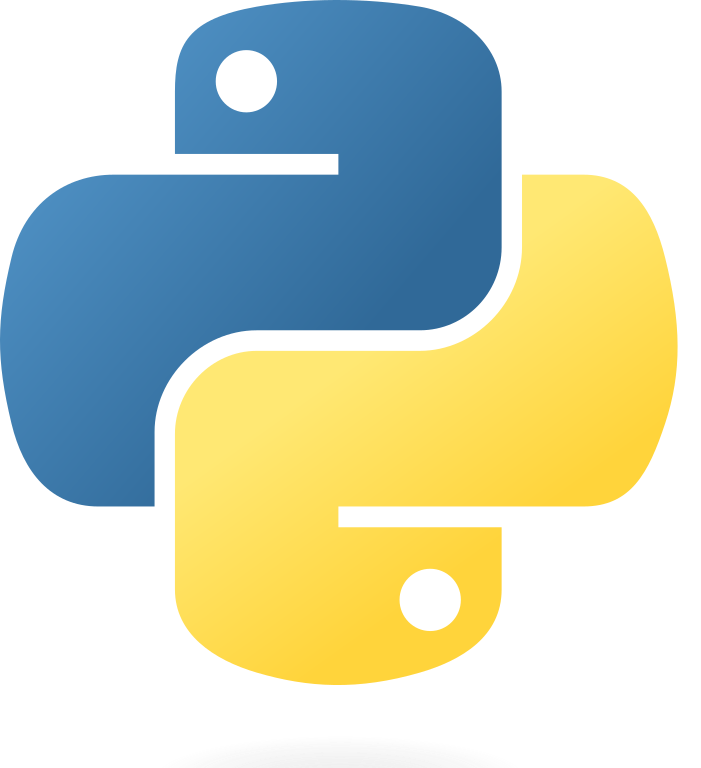
* **Logistic Regression**
* **Decision Tree Classifier**
* **Random Forest Classifier**

The dataset used is the **German Credit Dataset (GCD)**, containing demographic, financial, and behavioral data of loan applicants​

**Project Goal**

Classify loan applicants as **"Low Risk"** or **"High Risk"**, aiding financial institutions in making informed lending decisions​.

**Tools & Library Used**

[](https://www.python.org/)   [](https://jupyter.org/)

**Code and Dataset Links**

· **Project Code (Jupyter Notebook):** **https://github.com/AliKomas/Credit-Risk-Prediction-Machine-Learning**

· **Dataset Source:** German Credit Dataset on Kaggle **(https://www.kaggle.com/datasets/varunchawla30/german-credit-data)**

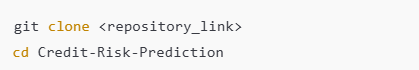
**Project Video Presentation**

* **YouTube Video Link:** ***(https://www.youtube.com/watch?v=TAlRJV0Ud7Q)***

**Installation**

* To run this project locally, follow these steps:

**Step 1: Clone the Repository**

****

**Step 2: Create a virtual environment:**

****

****

**Step 3: Install the required packages:**



**Step 4: Start Jupyter Notebook**

****

****

**Step 5: Run the Project Notebook**

Follow these steps inside the Jupyter Notebook:

1. Load the Dataset:  
   Ensure the dataset (german\_credit\_data.csv) is in the same directory or adjust the dataset path in the notebook.
2. Data Preprocessing:
   * Handle missing values.
   * Encode categorical variables.
   * Scale numerical features.
3. Train Machine Learning Models:
   * Logistic Regression
   * Decision Tree Classifier
   * Random Forest Classifier
4. Evaluate Model Performance:
   * Accuracy
   * Precision
   * Recall
   * F1-Score
   * ROC-AUC Curve
5. Visualize Results:
   * Plot accuracy comparisons.
   * Display ROC curves.
   * Show confusion matrices.

**📍 Step 6: View Results**

* Model performance metrics will be displayed in the notebook.
* Visualizations will help compare different models' effectiveness.
* Key insights will highlight feature importance and model accuracy.

**Local Computer Requirements**

* **Operating System:** Windows, macOS, or Linux
* **Python Version:** 3.8 or newer
* **RAM:** Minimum 8GB (16GB recommended)
* **Processor:** Intel i5 or higher
* **GPU (Optional):** NVIDIA GTX 1080 or higher

**Required Library Packages**

* Pandas: Data manipulation and analysis
* NumPy: Numerical computations
* Scikit-learn: Machine learning models and evaluation metrics
* Matplotlib: Data visualization
* Seaborn: Enhanced data visualization

**Dataset Description**

* **Dataset Name:** German Credit Dataset
* **Number of Samples:** 1,000
* **Features:** 20 (e.g., Age, Credit Amount, Loan Duration, Employment Duration, Savings Account)
* **Target Variable:** Risk (Good Risk/Bad Risk)

**Data Preprocessing Includes:**

* Handling Missing Values
* Encoding Categorical Variables
* Scaling Numerical Features
* Addressing Class Imbalance​

**Model Training and Evaluation**

Three models were implemented and evaluated:

* **Logistic Regression:** Baseline binary classification model
* **Decision Tree Classifier:** Captures non-linear patterns but prone to overfitting
* **Random Forest Classifier:** Ensemble model for higher accuracy

**Evaluation Metrics:**

* Accuracy
* Precision
* Recall
* F1-Score
* ROC-AUC Score

**Key Results**

* **Best Performing Model:** Logistic Regression with an AUC Score of **0.77**​
* **Feature Importance Analysis:** Key indicators include Credit Amount, Loan Duration, and Savings Account.
* **Graphical Representations:** ROC Curves, Confusion Matrices, and Accuracy Comparisons.