**🏦 Loan Approval Prediction Using Machine Learning**

**Aim:**

**To develop a machine learning model that predicts whether a loan application should be approved based on applicant financial and credit-related features.**

**Description:**

**This mini-project uses a synthetic dataset to simulate loan applications. The objective is to analyze financial features such as CIBIL score, income, loan amount, tenure, and existing loans to determine the likelihood of loan approval. A Random Forest Classifier is trained on the data, and predictions are made for new applicants. The project demonstrates a complete machine learning pipeline, including data generation, preprocessing, training, evaluation, and prediction.**

**Python Program:**

**python**

**CopyEdit**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.preprocessing import StandardScaler**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score**

**# Step 1: Create Synthetic Dataset**

**np.random.seed(42)**

**data = pd.DataFrame({**

**'CIBIL\_Score': np.random.randint(300, 900, 1000),**

**'Annual\_Income': np.random.randint(100000, 2000000, 1000),**

**'Loan\_Amount': np.random.randint(50000, 1000000, 1000),**

**'Tenure\_Years': np.random.randint(1, 30, 1000),**

**'Existing\_Loans': np.random.randint(0, 5, 1000),**

**'Loan\_Approved': np.random.choice([0, 1], 1000, p=[0.4, 0.6])**

**})**

**# Step 2: Exploratory Data Analysis**

**print("Dataset Overview:")**

**print(data.head())**

**plt.figure(figsize=(8, 6))**

**sns.heatmap(data.corr(), annot=True, cmap='coolwarm')**

**plt.title("Feature Correlation Heatmap")**

**plt.show()**

**sns.boxplot(x='Loan\_Approved', y='CIBIL\_Score', data=data)**

**plt.title('CIBIL Score vs Loan Approval')**

**plt.show()**

**# Step 3: Preprocessing**

**X = data.drop('Loan\_Approved', axis=1)**

**y = data['Loan\_Approved']**

**scaler = StandardScaler()**

**X\_scaled = scaler.fit\_transform(X)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(**

**X\_scaled, y, test\_size=0.2, random\_state=42**

**)**

**# Step 4: Train Model**

**model = RandomForestClassifier(n\_estimators=100, random\_state=42)**

**model.fit(X\_train, y\_train)**

**# Step 5: Evaluate Model**

**y\_pred = model.predict(X\_test)**

**print("\nModel Accuracy:", accuracy\_score(y\_test, y\_pred))**

**print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))**

**print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))**

**# Step 6: Predict New Application**

**new\_applicant = pd.DataFrame({**

**'CIBIL\_Score': [750],**

**'Annual\_Income': [1200000],**

**'Loan\_Amount': [500000],**

**'Tenure\_Years': [10],**

**'Existing\_Loans': [1]**

**})**

**new\_applicant\_scaled = scaler.transform(new\_applicant)**

**loan\_status = model.predict(new\_applicant\_scaled)**

**print("\nNew Applicant Prediction:")**

**print("✅ Loan Approved" if loan\_status[0] == 1 else "❌ Loan Rejected")**

**Dataset:**

**A synthetic dataset generated using NumPy, consisting of 1000 loan applicant records with the following fields:**

* **CIBIL\_Score (300–900)**
* **Annual\_Income (in INR)**
* **Loan\_Amount (in INR)**
* **Tenure\_Years (1–30 years)**
* **Existing\_Loans (0–4)**
* **Loan\_Approved (0 = Rejected, 1 = Approved)**

**Input / Output:**

**Input:  
Details of a new loan applicant:**

**python**

**CopyEdit**

**{**

**'CIBIL\_Score': 750,**

**'Annual\_Income': 1200000,**

**'Loan\_Amount': 500000,**

**'Tenure\_Years': 10,**

**'Existing\_Loans': 1**

**}**

**Output:**

**plaintext**

**CopyEdit**

**✅ Loan Approved**

**Result:**

**The model achieved high accuracy on the test data and successfully predicted the loan approval status for a new applicant. This project demonstrates how machine learning can assist in automated financial decision-making processes such as loan approvals.**