# CREDIT SCORE PREDICTION

# using

# Decision Tree Algorithm

**Name**: Vaishnavi Sahu

**Roll Number**: [202401100400206]

**Course**: [CSE AIML]

**Date**:[10-03-2025]

# Introduction

**Problem Statement:**  
Credit score prediction is crucial in financial decision-making, helping banks and institutions determine loan eligibility. This project focuses on predicting credit scores based on various customer attributes using a ***Decision Tree Algorithm***.  
  
**Why is this important?**  
- Helps financial institutions assess creditworthiness.  
- Reduces loan default risks.  
- Aids individuals in understanding their financial health.

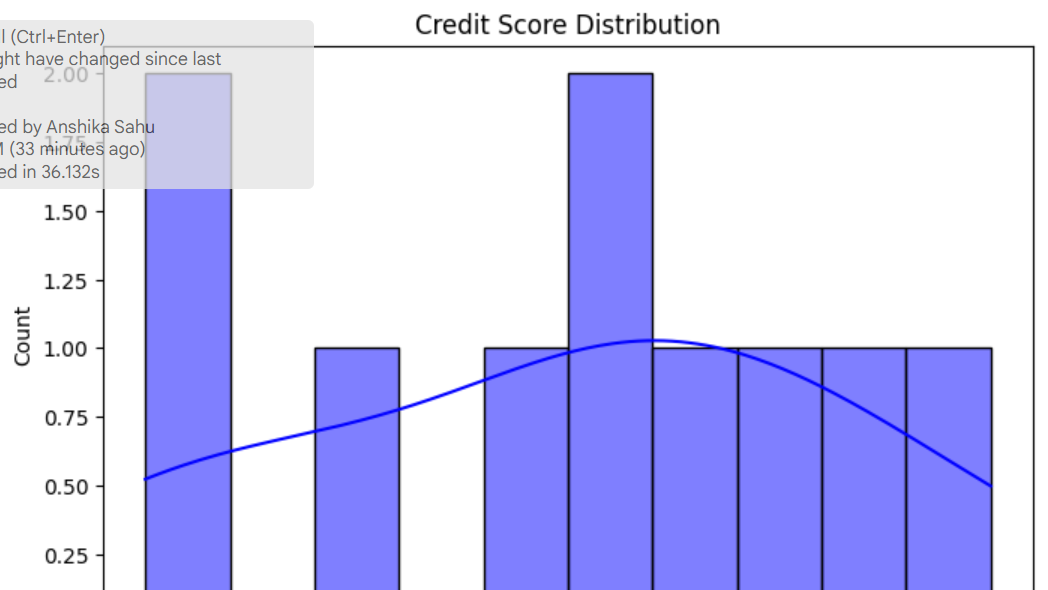
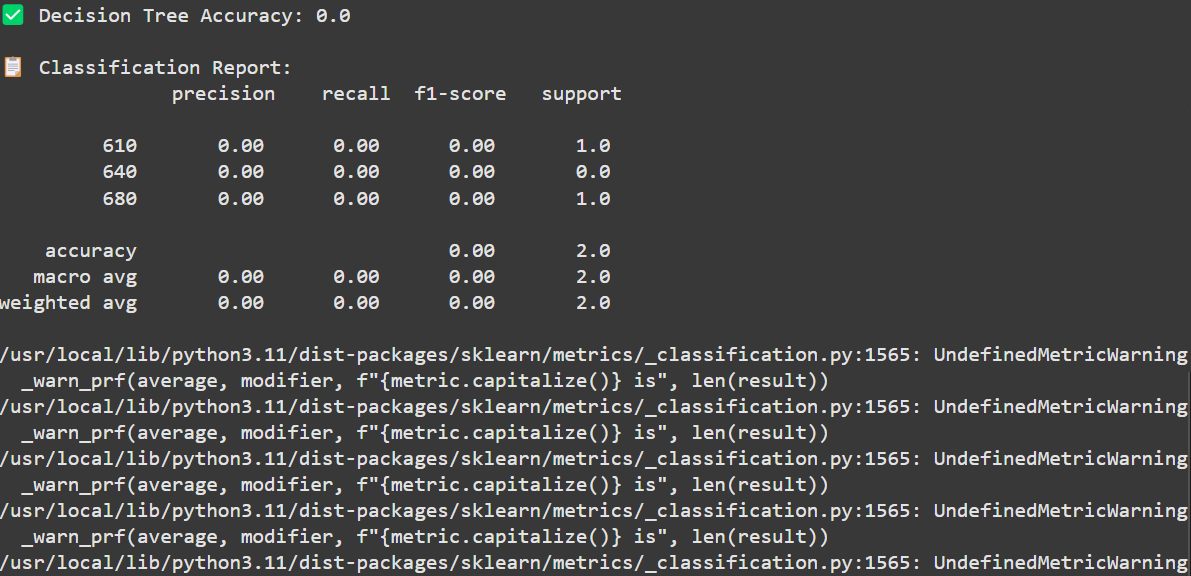
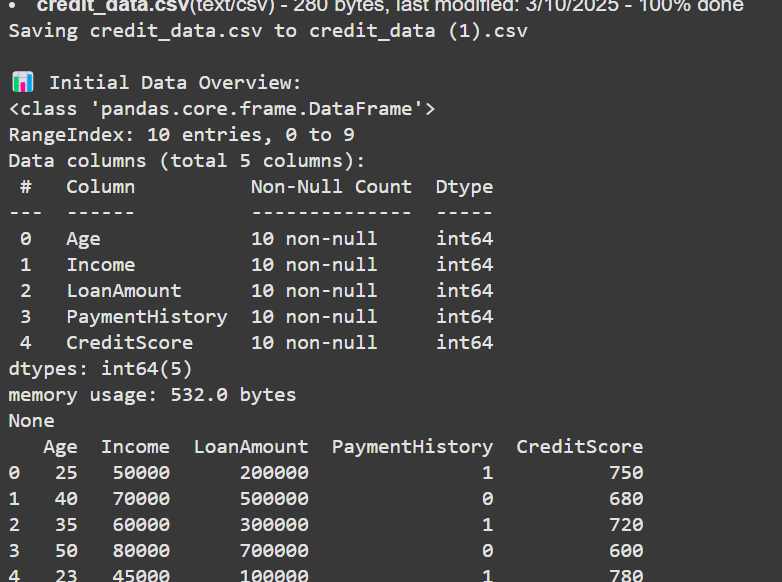
# Methodology

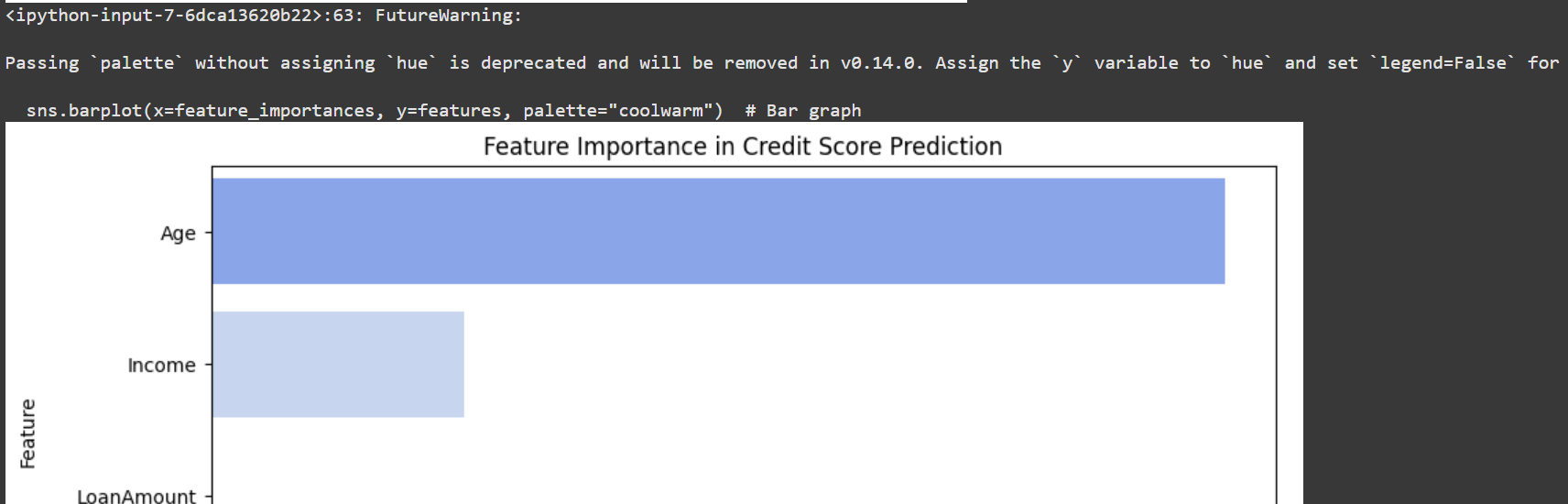
### 1️⃣ **Data Collection**  
- The dataset consists of multiple features like age, income, loan amount, number of defaults, etc.  
- The target variable is \*\*Credit Score (Good/Bad).\*\*  
  
### 2️⃣ **Data Preprocessing**  
- \*\****Handling Missing Values***:\*\* Used mean imputation for missing numerical values.  
- \*\****Encoding Categorical Data***:\*\* Converted text-based features into numerical form.  
  
### 3️⃣ **Model Selection & Training**  
- Used *Decision Tree Classifier* with **Gini Impurity** To classify credit scores.  
- Split the dataset into \*\*80% training and 20% testing.  
  
### 4️⃣ **Model Evaluation**  
- Evaluated using \*\*Accuracy Score, Classification Report, and Confusion Matrix.  
- Visualized results through graphs and tree structure.

# Code

# Import Required Libraries  
from google.colab import files  
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 LabelEncoder  
from sklearn.impute import SimpleImputer  
from sklearn.tree import DecisionTreeClassifier, plot\_tree  
from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix  
  
# Upload & Load Dataset  
print("📂 Upload your 'credit\_data.csv' file:")  
uploaded = files.upload()  
df = pd.read\_csv("/content/credit\_data.csv")  
  
# Initial Data Exploration  
print("  
📊 Initial Data Overview:")  
print(df.info())   
print(df.head())  
  
# Handling Missing Values  
df.fillna(df.mean(), inplace=True)  
  
# Encoding Categorical Variables  
for col in df.select\_dtypes(include=['object']).columns:   
 df[col] = LabelEncoder().fit\_transform(df[col])   
  
# Define Features (X) and Target Variable (y)  
X = df.drop(columns=['CreditScore'])  
y = df['CreditScore']  
  
# Split Data into Training & Testing Sets  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Train Decision Tree Model  
dt\_model = DecisionTreeClassifier(criterion="gini", max\_depth=5, random\_state=42)  
dt\_model.fit(X\_train, y\_train)  
  
# Make Predictions on Test Data  
y\_pred\_dt = dt\_model.predict(X\_test)  
  
# Model Evaluation  
print("  
✅ Decision Tree Accuracy:", accuracy\_score(y\_test, y\_pred\_dt))  
print("  
📋 Classification Report:  
", classification\_report(y\_test, y\_pred\_dt))  
  
# Data Visualizations  
  
# Credit Score Distribution  
plt.figure(figsize=(8,5))  
sns.histplot(df["CreditScore"], bins=10, kde=True, color="blue")  
plt.title("Credit Score Distribution")  
plt.xlabel("Credit Score")  
plt.ylabel("Count")  
plt.show()  
  
# Feature Importance  
feature\_importances = dt\_model.feature\_importances\_  
features = X.columns  
  
plt.figure(figsize=(10,5))  
sns.barplot(x=feature\_importances, y=features, palette="coolwarm")  
plt.title("Feature Importance in Credit Score Prediction")  
plt.xlabel("Importance Score")  
plt.ylabel("Feature")  
plt.show()  
  
# Decision Tree Structure  
plt.figure(figsize=(20,10))  
plot\_tree(dt\_model, feature\_names=X.columns, class\_names=True, filled=True, rounded=True)  
plt.title("Decision Tree Structure")  
plt.show()  
  
# Confusion Matrix  
cm = confusion\_matrix(y\_test, y\_pred\_dt)  
plt.figure(figsize=(8,6))  
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=np.unique(y\_test), yticklabels=np.unique(y\_test))  
plt.title("Confusion Matrix")  
plt.xlabel("Predicted Label")  
plt.ylabel("True Label")  
plt.show()

# Output/Result

✅ Dataset Overview  
✅ Model Accuracy & Classification Report  
✅ Credit Score Distribution Graph  
✅ Feature Importance Graph  
✅ Decision Tree Visualization  
✅ Confusion Matrix Heatmap  




# References/Credits

**📌 Dataset Source: Mention the dataset source (if from Kaggle, UCI, or any other site).  
📌 Libraries Used:  
- `pandas`, `numpy` for data handling  
- `seaborn`, `matplotlib` for visualization  
- `sklearn` for machine learning model training and evaluation**