Credit Score Prediction

Method Used: Data Visualization

By

Samriddhi Sahu
CSE(AI)
202401100300213

Introduction

Credit score prediction is the process of predicting a person's creditworthiness (Good, Average, or Poor) using machine learning based on factors like age, income, debt, marital status, and education level. The data is preprocessed by converting categorical values into numbers using label encoding and scaling numerical values for better model performance. A Random Forest Classifier is trained on this data, which combines multiple decision trees to improve accuracy and reduce overfitting. The predicted credit scores for new data and is evaluated using accuracy, confusion matrix, and classification report. This helps financial institutions make better loan decisions and low the financial risk.

Here,

Good Score -> Low risk of error in the payments
Average -> Moderate risk
Poor -> High risk of errors

Methodology

In this project we have used the method of **Data Visualization**

Data visualization is the graphical representation of data and information using charts, graphs, maps, and other visual elements. The goal of data visualization is to make complex data easier to understand and interpret by presenting it in a visually appealing and accessible format. By using visuals, trends, patterns, and outliers in data become more obvious, which can help in making data-driven decisions.

Common types of data visualizations include:

Bar charts – To compare quantities across different categories.

Line charts – To show trends over time.

Pie charts – To show the proportion of categories in a whole.

Data visualization helps analysts and scientists in decision-making to communicate findings in a clear and effective way.

Code for this Project

```
#Importing libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Sample Data Generation (replace this with real data)
data = {
    'Age': np.random.randint(20, 65, 1000),
    'Income': np.random.randint(30000, 120000, 1000),
    'Debt': np.random.randint(0, 50000, 1000),
    'Marital_Status': np.random.choice(['Single', 'Married', 'Divorced'], 1000),
    'Education_Level': np.random.choice(['High School', 'Bachelor', 'Master', 'PhD'], 1000),
    'Credit_Score': np.random.choice(['Good', 'Average', 'Poor'], 1000)
}

# Create a DataFrame
df = pd.DataFrame(data)

#  Check the data
print(df.head())

# Encode categorical variables
le = LabelEncoder()
df['Marital_Status'] = le.fit_transform(df['Marital_Status'])
df['Education_Level'] = le.fit_transform(df['Education_Level'])
df['Credit_Score'] = le.fit_transform(df['Credit_Score'])  # Target variable

# Split data into features and target
X = df.drop('Credit_Score', axis=1)
```

```
y = df['Credit_Score']

# Split into train and test sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a
Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Model Evaluation
print("\nAccuracy Score:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Confusion Matrix Heatmap
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, cmap="Blues", fmt="d")
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

The output of this Project