Department of Computer Science Engineering (AI)

ARTIFICAL INTELLIGENCE
Project Report

On

CREDIT SCORE PREDICTION

By

ANISH CHAUDHARY

Date: 09/01/2025

1. Introduction

Credit Score Prediction is a crucial task in financial institutions to determine an individual's creditworthiness. A credit score is a numerical representation of a person's financial history and ability to repay loans. Machine learning techniques can help automate and improve the accuracy of credit score assessments, reducing manual errors and biases.

The objective of this project is to build a machine learning model that can predict an individual's credit score based on features such as income, age, loan history, and other financial parameters.

---

2. Methodology

2.1 Dataset

The dataset used for this project contains various features related to individuals' financial history. The dataset includes columns such as:

- Age
- Income
- Loan Amount
- Debt-to-Income Ratio
- Number of Previous Loans
- Credit History
- Default History
- Target Variable: Credit Score (Categorized as Good, Average, or Poor)

2.2 Data Preprocessing

- Handling Missing Values: Missing data was removed using df.dropna().

- **Encoding Categorical Variables: Non-numeric values were converted to numeric using LabelEncoder().**
- **Feature Scaling: Standardization was applied using StandardScaler().**
- **Splitting Data: The dataset was split into training (80%) and testing (20%) sets using train_test_split().**

2.3 Model Selection

The Random Forest Classifier was used for training the model. It is an ensemble learning method that combines multiple decision trees to improve prediction accuracy and reduce overfitting.

2.4 Model Training & Evaluation

- **The model was trained using RandomForestClassifier with 100 trees (n_estimators=100).**
- **Model performance was evaluated using Accuracy Score, Confusion Matrix, and Classification Report.**

---

3. Results & Outputs

3.1 Model Performance Metrics

- **Accuracy Score: (Actual value will be displayed here after execution)**
- **Confusion Matrix: (Shows how well the model classifies credit scores)**
- **Classification Report: (Displays precision, recall, and F1-score for each class)**

3.2 Feature Importance Visualization

The top 10 most important features affecting the credit score were plotted using a bar chart.

---

4. Code Implementation

Below is the Python code used for the project:

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,
LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,
classification_report, confusion_matrix

# Load the dataset
df = pd.read_csv("/content/credit_data.csv")

# Display basic info
print(df.head())
df.info()

# Handling missing values
df.dropna(inplace=True)

# Encoding categorical columns (if any)
for col in df.select_dtypes(include=['object']).columns:
    df[col] = LabelEncoder().fit_transform(df[col])

# Identify the target column dynamically
target_column = 'Credit_Score'
if target_column not in df.columns:
    potential_targets = [col for col in df.columns if 'credit' in
col.lower() and 'score' in col.lower()]
```

```
    if potential_targets:
        target_column = potential_targets[0]
        print(f"Using '{target_column}' as the target column instead
of 'Credit_Score'")
    else:
        raise KeyError("'Credit_Score' or similar column not found
in the dataset.")

# Splitting features and target variable
X = df.drop(columns=[target_column])
y = df[target_column]

# Splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Model training (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("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test,
y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Feature importance visualization
feature_importances = pd.Series(model.feature_importances_,
index=X.columns)
feature_importances.nlargest(10).plot(kind='barh')
plt.show()
```

---

5. Conclusion

This project successfully implemented a machine learning model to predict credit scores based on financial data. The Random Forest Classifier provided a robust and interpretable solution for this task.

Future improvements can include:

- Trying other models like XGBoost, Logistic Regression, or Neural Networks.
- Using SMOTE to balance dataset classes if there is an imbalance.
- Collecting more features for better prediction accuracy.

---

6. References

- Pandas Documentation: https://pandas.pydata.org/
- Scikit-learn Documentation: https://scikit-learn.org/
- Matplotlib for Visualization: https://matplotlib.org/