Credit Score Predictor Report

1.Title page

Project Tittle: Credit score predictor

Prepared by: Shubham Mishra

Roll no: 202401100300245

Institution: KIET Group of Institution

2.Introduction

A **Credit Score Predictor** is a financial tool designed to estimate an individual's credit score based on various financial and behaviour factors. This predictive model assists financial institutions, fintech companies, and consumers in evaluating creditworthiness without directly accessing official credit bureau data.

3.Objectives

The primary objectives of a Credit Score Predictor include:

- Providing an estimate of a user's credit score.
- Helping financial institutions assess loan risks.
- Offering consumers insights into their credit health.
- Streamlining the credit approval process.

4.Data Sources

The accuracy of a Credit Score Predictor depends on the quality of data used. Common data sources include:

- **Payment History**: Timeliness of previous credit payments.
- **Credit Utilization**: Ratio of used credit to available credit.
- **Length of Credit History**: Duration of active credit accounts.
- **Credit Mix**: Types of credit accounts held (credit cards, loans, mortgages, etc.).
- **Recent Credit Inquiries**: Frequency of new credit applications.

5.Methodology

The process of predicting a credit score involves several steps:

1. **Data Collection**: Gathering financial data from sources like bank statements, credit card reports, and transactional history.
2. **Data Preprocessing**: Cleaning and transforming raw data into a structured format.
3. **Feature Engineering**: Selecting and refining relevant attributes such as payment history, outstanding debt, and credit inquiries.
4. **Model Selection**: Choosing an appropriate machine learning model such as:
    - Logistic Regression
    - Decision Trees
    - Random Forest
    - Neural Networks
5. **Training and Testing**: Using historical credit data to train the model and validate its accuracy.
6. **Prediction Generation**: Assigning a credit score range based on input parameters.

6.Applications

A Credit Score Predictor has various applications, including:

- **Banking & Lending**: Evaluating loan applicants' creditworthiness.
- **Fintech Services**: Providing real-time credit insights to users.
- **Consumer Credit Monitoring**: Helping individuals improve financial habits.
- **Risk Management**: Identifying high-risk customers for lenders.

7.Limitations

Despite its usefulness, a Credit Score Predictor has some limitations:

- **Data Dependency**: Accuracy is contingent on the quality and completeness of the data.
- **Lack of Official Credit Data**: Cannot fully replace bureau-generated credit scores.
- **Model Bias**: Potential biases in the training data may affect predictions.
- **Regulatory Compliance**: Must comply with data privacy laws such as GDPR and FCRA.

8.Conclusion

A Credit Score Predictor is a valuable tool for financial decision-making, offering insights into an individual's credit health. While it enhances credit accessibility and risk assessment, it should be used alongside official credit reports for a comprehensive evaluation.

9.Future Enhancements

To improve accuracy and reliability, future developments may include:

- Integration of alternative credit data such as utility payments and rental history.
- Advanced machine learning techniques to enhance predictive accuracy.
- Real-time updates based on user financial behavior.
- Compliance with evolving financial regulations to ensure fair lending practices.

10.Code Implementation

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
data = pd.read_csv('credit_data.csv')

# Display the first few rows of the dataset
print(data.head())

# Define features (X) and target (y)
# Features: Age, Income, LoanAmount
# Target: CreditScore
X = data[['Age', 'Income', 'LoanAmount']]
y = data['CreditScore']

# Split the data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize the Linear Regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, y_train)

# Make predictions on the test data
```

```
y_pred = model.predict(X_test)


# Evaluate the model
# Mean Squared Error (MSE) measures the average squared difference between predicted and actual values
mse = mean_squared_error(y_test, y_pred)
# R-squared (R2) measures how well the model explains the variance in the target variable
r2 = r2_score(y_test, y_pred)


print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")


# Predict a credit score for a new data point
# Replace the values below with actual data for prediction
new_data = pd.DataFrame({
    'Age': [30],  # Age of the customer
    'Income': [50000],  # Annual income
    'LoanAmount': [20000]  # Loan amount
})


predicted_score = model.predict(new_data)
print(f"Predicted Credit Score: {predicted_score[0]}")
```

11.Screenshorts and Output Results

Visual Output

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
from sklearn.metrics import mean_squared_error, r2_score

# Load the dataset
data = pd.read_csv('credit_data.csv')

# Display the first few rows of the dataset
print(data.head())

# Define features (X) and target (y)
# Features: Age, Income, LoanAmount
# Target: CreditScore
X = data[['Age', 'Income', 'LoanAmount']]
y = data['CreditScore']

# Split the data into training and testing sets (80% training, 20%
testing)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Initialize the Linear Regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Evaluate the model
# Mean Squared Error (MSE) measures the average squared difference
between predicted and actual values
mse = mean_squared_error(y_test, y_pred)
# R-squared (R2) measures how well the model explains the variance in
the target variable
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")

# Predict a credit score for a new data point
# Replace the values below with actual data for prediction
new_data = pd.DataFrame({
    'Age': [30],  # Age of the customer
    'Income': [50000],  # Annual income
    'LoanAmount': [20000]  # Loan amount
})

predicted_score = model.predict(new_data)
```

```
print(f"Predicted Credit Score: {predicted_score[0]}")
```

```
   CustomerID  Age  Income  LoanAmount  CreditScore
0           1   34   63961       13576          531
1           2   50   34158       19395          323
2           3   67   92702       37663          416
3           4   26   84846       44572          747
4           5   58   40787       25598          805
Mean Squared Error: 77039.2842810213
R-squared: -9.44644729434587
Predicted Credit Score: 651.0666706564242
```

- **Data set used-credit-Data**

CustomerI	Age	Income	LoanAmou	CreditScore
1	34	63961	13576	531
2	50	34158	19395	323
3	67	92702	37663	416
4	26	84846	44572	747
5	58	40787	25598	805
6	25	94896	20109	807
7	72	21190	4529	624
8	35	75969	47313	776
9	61	46994	42386	664
10	49	49169	3029	631
11	22	37260	22277	661
12	53	80245	18377	756
13	43	74120	27327	503
14	43	52653	17198	673
15	37	80522	32572	707
16	25	96681	4679	445
17	21	28686	7481	542
18	22	82482	39671	332
19	52	32348	28831	619
20	22	20814	27506	477