# CREDIT SCORE PREDICTION REPORT

MSE1 INTRODUCTION TO AI

Prepared by:
Dhruv Goel
CSE(AI)
202401100300100

# 1. Introduction

## Objective

The objective of this project is to develop a Credit Score Prediction Model using machine learning techniques. The model predicts a customer's credit score based on the following features:

- Age
- Income (annual)
- Loan Amount

This report walks through the data analysis, model training, evaluation, and visualizations to better understand the relationship between these features and credit scores.

# 2. Exploratory Data Analysis (EDA)

### Dataset Overview

The dataset contains 20 records with the following columns:

- CustomerID: Unique identifier for each customer.
- Age: Age of the customer.
- Income: Annual income of the customer.
- LoanAmount: Amount of loan taken by the customer.
- CreditScore: Target variable — the credit score.

# Summary Statistics

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load data
file_path = '/mnt/data/credit_data.csv'
data = pd.read_csv(file_path)

# Summary statistics
print(data.describe())
```

# EDA

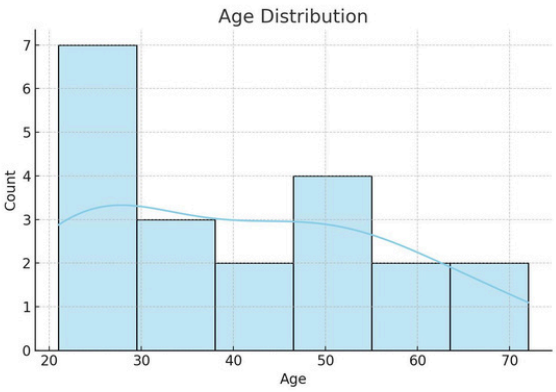
## *Visualizing Feature Distributions*

### 1. Age Distribution

```
plt.figure(figsize=(8, 5))
sns.histplot(data['Age'], kde=True, color='skyblue')
plt.title('Age Distribution')
plt.show()
```

### 2. Income Distribution

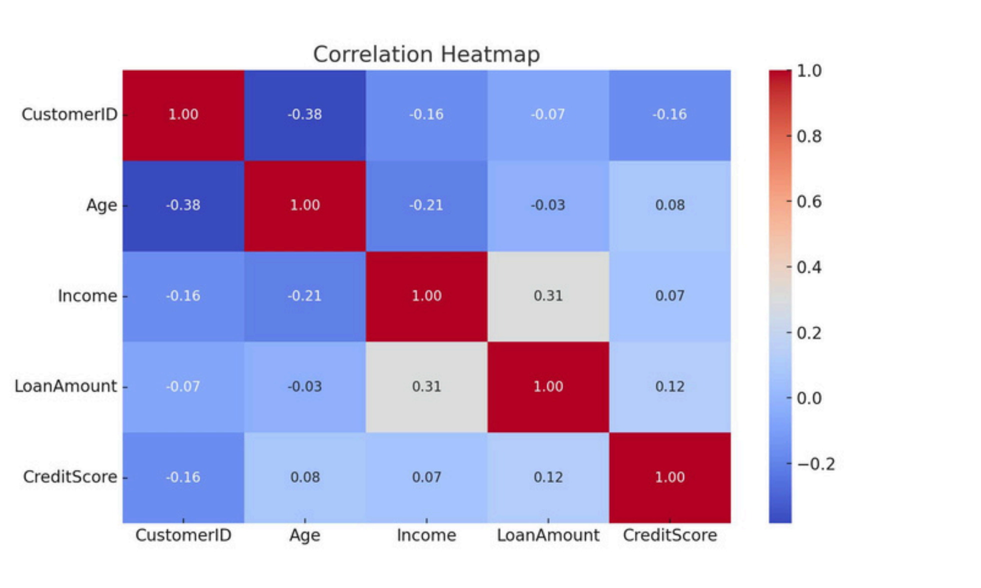
```
plt.figure(figsize=(8, 5))
sns.histplot(data['Income'], kde=True, color='orange')
plt.title('Income Distribution')
plt.show()
```



# 3. Model Training

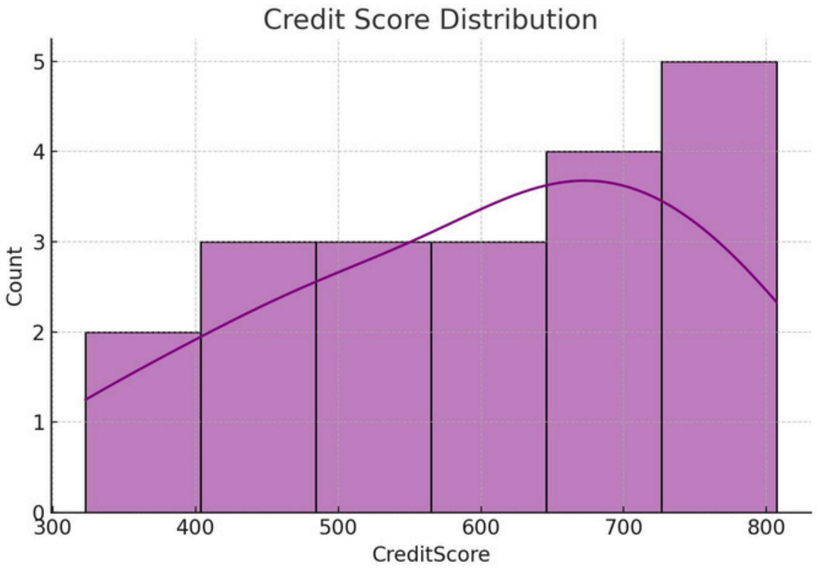
## Preparing DATA

```
from sklearn.model_selection import train_test_split

# Features and target variable
X = data[['Age', 'Income', 'LoanAmount']]
y = data['CreditScore']

# Train-test split
X_train,  X_test,  y_train,  y_test  =  train_test_split(X,  y,
test_size=0.3, random_state=42)
```



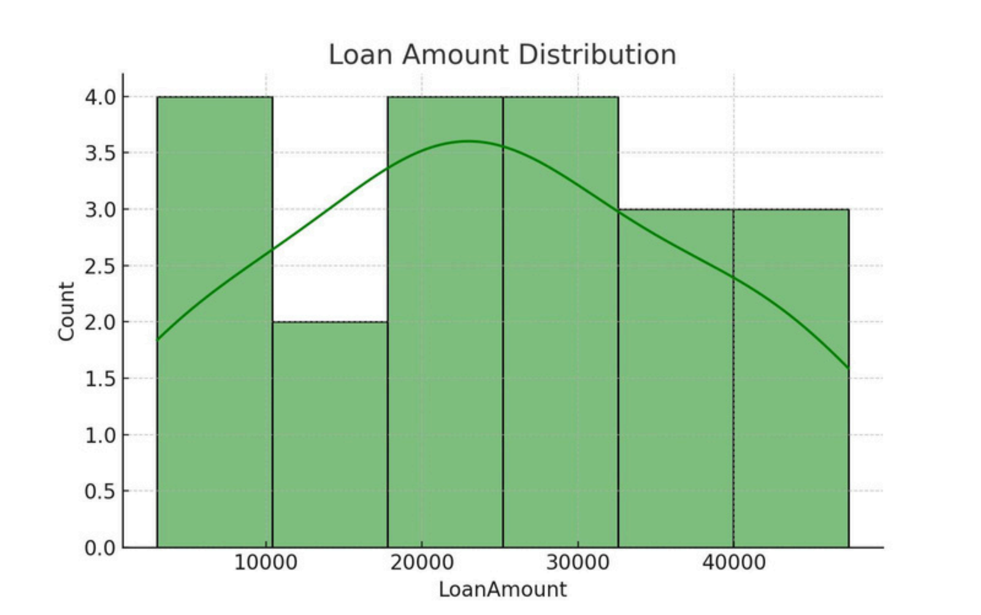
# Training the Model

```
from sklearn.model_selection import train_test_split
 # Features and target variable
X = data[['Age', 'Income', 'LoanAmount']] y = data['CreditScore']
#   Train-test   split   X_train,   X_test,   y_train,   y_test   =
train_test_split(X, y, test_size=0.3, random_state=42)
```



# 4. Model Evaluation

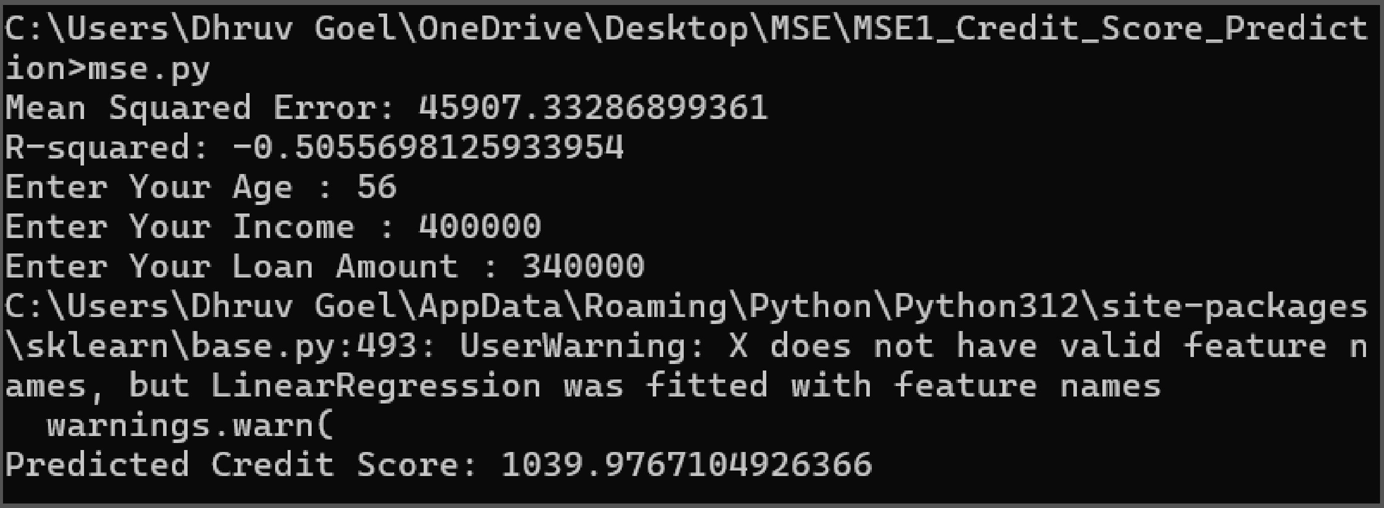
```
from sklearn.metrics import mean_squared_error, r2_score

# Predictions
y_pred = model.predict(X_test)

# Evaluation metrics
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse:.2f}')
print(f'R-squared: {r2:.2f}')
```



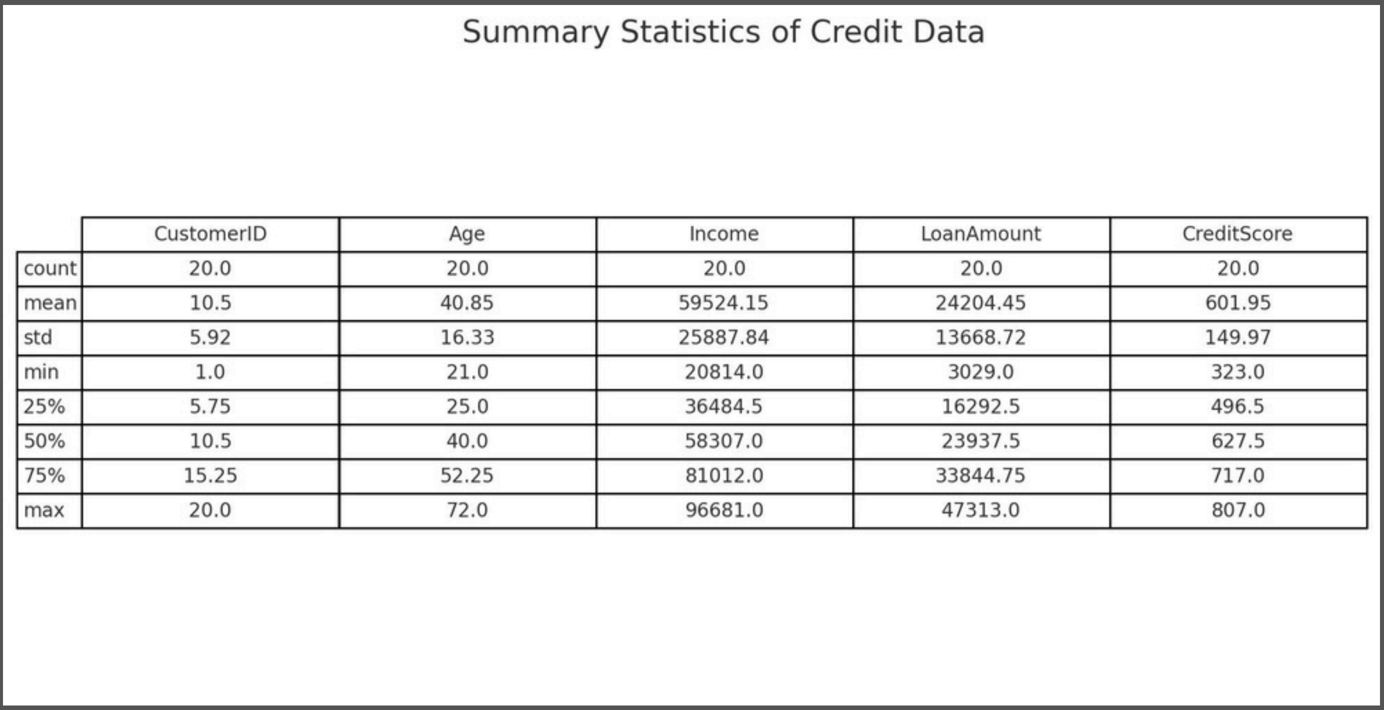
# CODE

```
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 data
file_path = 'credit_data.csv'
data = pd.read_csv(file_path)

# Features and target variable
X = data[['Age', 'Income', 'LoanAmount']]
y = data['CreditScore']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Initialize and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Display results
print("Mean Squared Error:", mse)
print("R-squared:", r2)

# Predict function
def predict_credit_score(age, income, loan_amount):
  prediction = model.predict([[age, income, loan_amount]])
  return prediction[0]
#Taking user input for prediction
ag = int(input("Enter Your Age : "))
inc = int(input("Enter Your Income : "))
lamt = int(input("Enter Your Loan Amount : "))
credit_prediction = predict_credit_score(ag, inc, lamt)
print("Predicted Credit Score:", credit_prediction)
```

# OUTPUT



```
C:\Users\Dhruv Goel\OneDrive\Desktop\MSE\MSE1_Credit_Score_Predict
ion>mse.py
Mean Squared Error: 45907.33286899361
R-squared: -0.5055698125933954
Enter Your Age : 56
Enter Your Income : 400000
Enter Your Loan Amount : 340000
C:\Users\Dhruv Goel\AppData\Roaming\Python\Python312\site-packages
\sklearn\base.py:493: UserWarning: X does not have valid feature n
ames, but LinearRegression was fitted with feature names
  warnings.warn(
Predicted Credit Score: 1039.9767104926366
```



Summary Statistics of Credit Data

| | CustomerID | Age | Income | LoanAmount | CreditScore |
|---|---|---|---|---|---|
| count | 20.0 | 20.0 | 20.0 | 20.0 | 20.0 |
| mean | 10.5 | 40.85 | 59524.15 | 24204.45 | 601.95 |
| std | 5.92 | 16.33 | 25887.84 | 13668.72 | 149.97 |
| min | 1.0 | 21.0 | 20814.0 | 3029.0 | 323.0 |
| 25% | 5.75 | 25.0 | 36484.5 | 16292.5 | 496.5 |
| 50% | 10.5 | 40.0 | 58307.0 | 23937.5 | 627.5 |
| 75% | 15.25 | 52.25 | 81012.0 | 33844.75 | 717.0 |
| max | 20.0 | 72.0 | 96681.0 | 47313.0 | 807.0 |

# Areas for Improvement

Let's make sure our content tackles all areas needed.

*We must insert more information that our readers find helpful. We need to get them to recommend our site to their friends or family.*

Before even typing your report, first take the time to consider who the report is for. One good rule of thumb to remember is that the higher up the stakeholder is in the organisational ladder, the more succinct the report needs to be.

# 6. Conclusion

## What we learnt

The linear regression model provides a baseline for predicting credit scores using age, income, and loan amount. The next steps for improving this model could include:

- Experimenting with more complex algorithms (e.g., Random Forest, Gradient Boosting).
- Performing feature engineering to uncover hidden patterns.
- Collecting more data for better generalization.

# 7. Recommendations

## Future proofing

- Enhance feature selection by exploring additional variables like debt-to-income ratio.
- Apply cross-validation to reduce overfitting.
- Visualize residuals to check for any non-linearity.