# EX:No.3 221501015

18/02/25

Program to perform Linear Regression on Bitcoin price data by importing, loading, preprocessing, and visualizing data.

## Aim:

To implement Linear Regression on Bitcoin price data by importing, loading, preprocessing, and visualizing the dataset.

## Implementation:

### Importing necessary libraries

import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

### Loading the dataset

file\_path = 'BTC-USD.csv'  
try:  
 df = pd.read\_csv(file\_path, parse\_dates=['Date'])  
 display(df.head())  
 display(df.info())  
except FileNotFoundError:  
 print("Error: 'BTC-USD.csv' not found.")  
 df = None  
except pd.errors.ParserError:  
 print("Error: Could not parse the CSV file correctly.")  
 df = None  
except Exception as e:  
 print(f"An unexpected error occurred: {e}")  
 df = None

### Data Preprocessing

# Check for missing values and data types  
missing\_values = df.isnull().sum()  
print("Missing values per column:\n", missing\_values)  
print("\nData types per column:\n", df.dtypes)  
  
# Set 'Date' column as index and handle missing values  
df = df.set\_index('Date')  
df.fillna(method='ffill', inplace=True)

### Visualizing the dataset

# Plot the distribution of Bitcoin closing price  
plt.figure(figsize=(10, 6))  
plt.hist(df['Close'], bins=50, color='skyblue', edgecolor='black')  
plt.title('Distribution of Bitcoin Close Price')  
plt.xlabel('Close Price')  
plt.ylabel('Frequency')  
plt.show()

### Linear Regression Model Implementation

# Define input and output variables  
X = df[['Open']]  
y = df['Close']  
  
# Split data into training and testing sets  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Train the Linear Regression model  
model = LinearRegression()  
model.fit(X\_train, y\_train)  
  
# Predict Bitcoin prices on the test data  
y\_pred = model.predict(X\_test)

### Model Evaluapasted-movie.pngtion

mae = mean\_absolute\_error(y\_test, y\_pred)  
mse = mean\_squared\_error(y\_test, y\_pred)  
print(f'Mean Absolute Error: {mae}')  
print(f'Mean Squared Error: {mse}')

### Plot Actual vs Predicted Prices

plt.figure(figsize=(10, 6))  
plt.scatter(X\_test, y\_test, color='blue', label='Actual Prices')  
plt.plot(X\_test, y\_pred, color='red', label='Predicted Prices')  
plt.title('Actual vs Predicted Bitcoin Prices')  
plt.xlabel('Open Price')  
plt.ylabel('Close Price')  
plt.legend()  
plt.show()

## Output:

## pasted-movie.png

## Result:

Thus, importing, loading, preprocessing, and applying Linear Regression on Bitcoin price data has been successfully implemented.