Exp no: 3 Implement linear regression model for forecasting time series data.

Date: 20/03/25

**Objectives:**

The primary objective of this experiment is to analyze and predict stock market prices using machine learning techniques. By preprocessing the dataset, handling missing values, and training a Linear Regression model, we aim to forecast stock closing prices based on historical data.

**Background/Scope:**

Stock market data is inherently time series-based and consists of variables like opening price, highest price, lowest price, closing price, and trading volume. Predicting stock prices helps investors and analysts make informed financial decisions. This experiment focuses on preprocessing the dataset and applying a machine learning model to predict future prices.

**Steps for Stock Market Prediction:**

Step 1: Load the Dataset

Load the dataset from a local CSV file and display the first few rows.

import pandas as pd

# Load the dataset

df = pd.read\_csv("market.csv")

# Display the first few rows

print(df.head())

Step 2: Data Cleaning and Preprocessing

Convert the date column to a datetime object and remove missing values.

# Convert 'Date' to datetime format

df['Date'] = pd.to\_datetime(df['Date'])

# Remove missing values

df.dropna(subset=['Date', 'Close', 'Adj Close', 'Open', 'High', 'Low', 'Volume'], inplace=True)

Step 3: Handle Duplicates

Remove duplicate records based on date and ticker to ensure data integrity.

df.drop\_duplicates(subset=['Date', 'Ticker'], inplace=True)

Step 4: Outlier Removal

Use the interquartile range (IQR) method to detect and remove outliers from the closing price.

Q1 = df['Close'].quantile(0.25)

Q3 = df['Close'].quantile(0.75)

IQR = Q3 - Q1

df = df[(df['Close'] >= (Q1 - 1.5 \* IQR)) & (df['Close'] <= (Q3 + 1.5 \* IQR))]

Step 5: Feature Selection and Data Splitting

Select relevant features and split the data into training and testing sets.

from sklearn.model\_selection import train\_test\_split

X = df[['High', 'Low', 'Open', 'Volume']].values

y = df['Close'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1)

Step 6: Train a Linear Regression Model

Train a Linear Regression model to predict the stock closing price.

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, y\_train)

predicted = regressor.predict(X\_test)

Step 7: Compare Actual vs. Predicted Prices

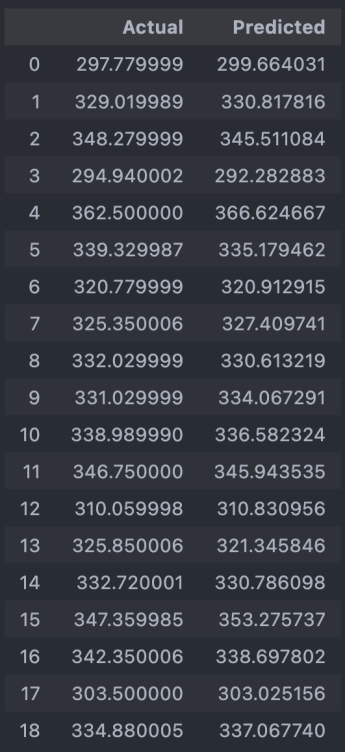
Compare actual and predicted stock prices to evaluate model performance.

import pandas as pd

# Create a DataFrame for comparison

data1 = pd.DataFrame({'Actual': y\_test.flatten(), 'Predicted': predicted.flatten()})

print(data1.head(20))



Step 8: Visualizing Predictions

Plot the actual vs. predicted values to assess the model's accuracy.

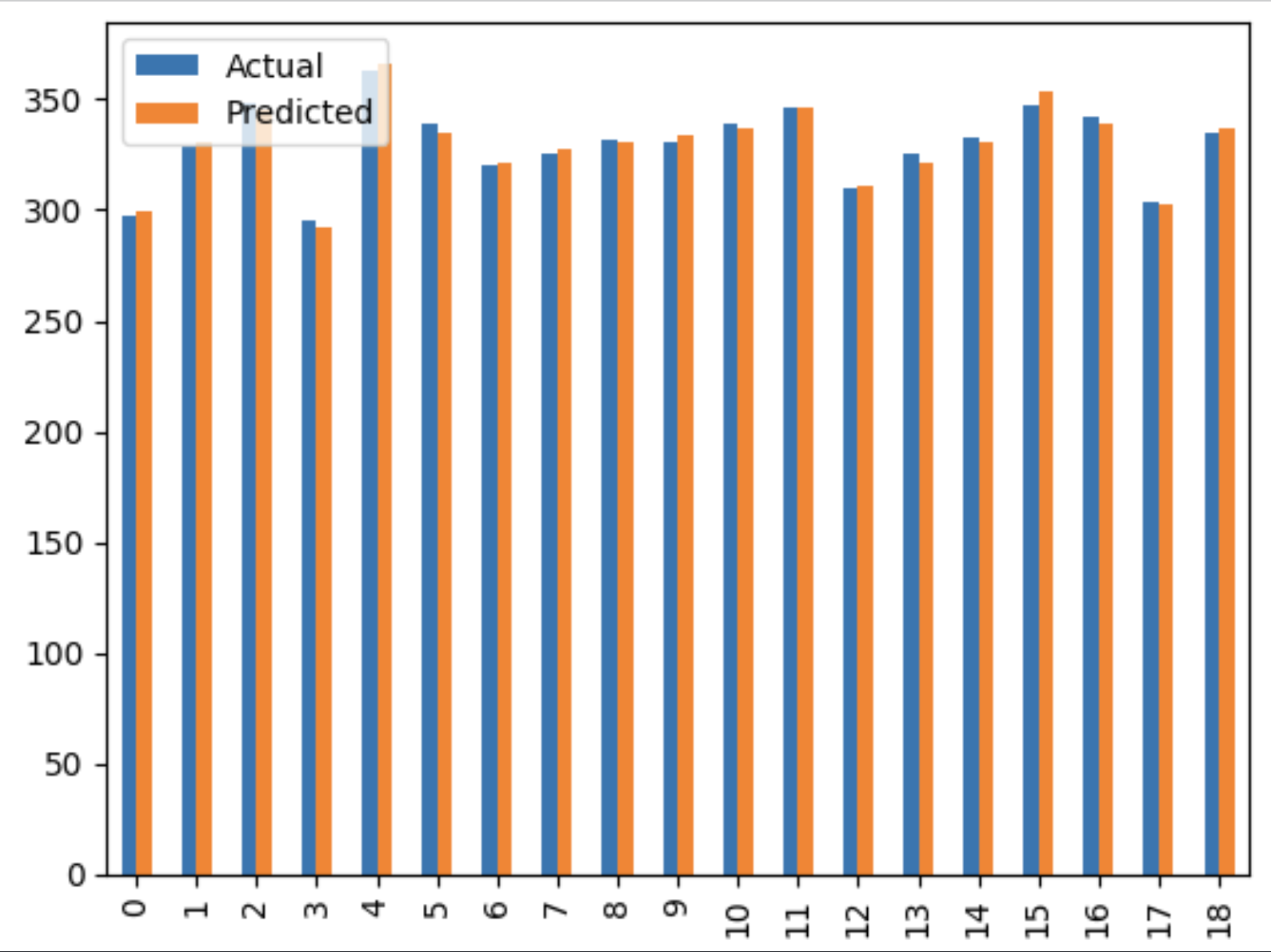
import matplotlib.pyplot as plt

data1.head(20).plot(kind='bar', figsize=(12, 6))

plt.title('Actual vs. Predicted Stock Prices')

plt.xlabel('Data Points')

plt.ylabel('Stock Price')

plt.show()

**Result:**

Thus, the stock market dataset is successfully processed, and a Linear Regression model is trained to predict stock closing prices with reasonable accuracy.