# Import necessary libraries

import numpy as np

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

import matplotlib.pyplot as plt

# Generate synthetic stock data

np.random.seed(42)

dates = pd.date\_range(start="2023-01-01", end="2023-12-31", freq="B")

prices = np.cumsum(np.random.normal(0, 1, len(dates)))

data = pd.DataFrame({"Date": dates, "Close": prices})

# Select relevant features (e.g., closing prices)

features = data["Close"].values.reshape(-1, 1)

# Create a new column for the target variable (next day's closing price)

data["Target"] = data["Close"].shift(-1)

target = data["Target"].dropna().values

# Split the data into training and testing sets

features\_train, features\_test, target\_train, target\_test = train\_test\_split(

features[:-1], target, test\_size=0.2, random\_state=42

)

# Create and train a linear regression model

model = LinearRegression()

model.fit(features\_train, target\_train)

# Make predictions on the test set

predictions = model.predict(features\_test)

# Evaluate the model

mse = mean\_squared\_error(target\_test, predictions)

print(f"Mean Squared Error: {mse}")

# Visualize the results

plt.scatter(features\_test, target\_test, color="black")

plt.plot(features\_test, predictions, color="blue", linewidth=3)

plt.xlabel("Closing Price")

plt.ylabel("Next Day's Closing Price")

plt.title("Stock Market Prediction")

plt.show()

Mean Squared Error: 0.9034454764577216

