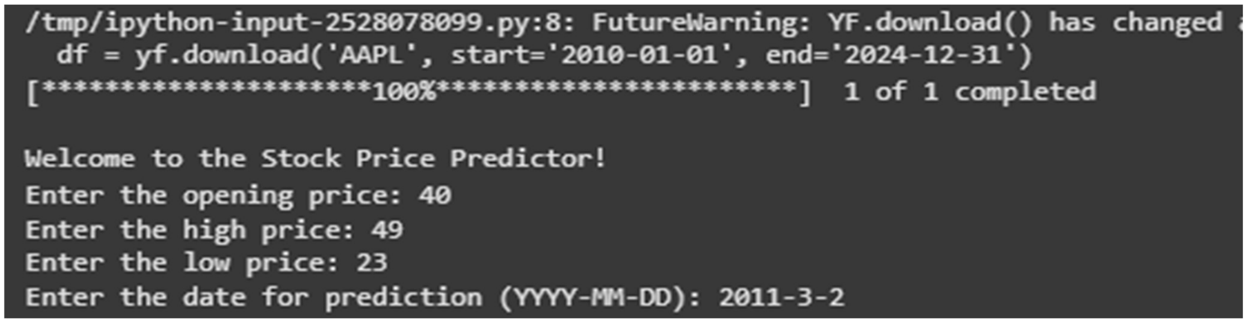
DHC Task 2:(Code)

Predict Future Stock Prices Using Linear Regression (AAPL)

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import yfinance as yf
import matplotlib.dates as mdates
# Step 1: Download Apple stock data from 2010 to 2024
df = yf.download('AAPL', start='2010-01-01', end='2024-12-31')
df.reset_index(inplace=True)
# Step 2: Create target column for next day’s Close price
df['Target'] = df['Close'].shift(-1)
df.dropna(inplace=True)
# Step 3: Select input features and target variable
features = ['Open', 'High', 'Low']
X = df[features]
y = df['Target']
# Step 4: Train Linear Regression model
model = LinearRegression()
model.fit(X, y)
# Step 5: Predict values for entire dataset
df['Predicted_Close'] = model.predict(X)
# Step 6: Take user input
```

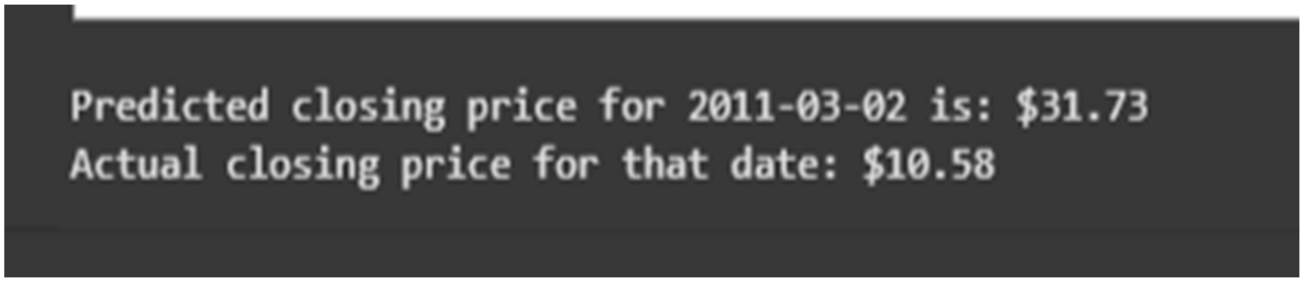
```
print("\nWelcome to the Stock Price Predictor!")
open_price = float(input("Enter the opening price: "))
high_price = float(input("Enter the high price: "))
low_price = float(input("Enter the low price: "))
input_date = input("Enter the date for prediction (YYYY-MM-DD): ")
input_date = pd.to_datetime(input_date)
# Step 7: Make prediction based on input
user_input = pd.DataFrame([[open_price, high_price, low_price]],
columns=features)
user_predicted_price = model.predict(user_input)[0]
# Step 8: Try to find actual price from dataset
actual_price = None
date_match = df[df['Date'] == input_date]
if not date_match.empty:
    actual_price = float(date_match.iloc[0]['Close'])  # 100% guaranteed to
extract just the float
# Step 9: Plotting
plt.figure(figsize=(16, 10))
# Plot actual and predicted prices over time
plt.plot(df['Date'], df['Close'], label='Actual Close Price', color='blue',
linewidth=2)
plt.plot(df['Date'], df['Predicted_Close'], label='Predicted Close Price',
color='red', linestyle='--', linewidth=2)
# Plot user's prediction
plt.scatter([input_date], [user_predicted_price], color='green', label='Your
Prediction', s=120, zorder=5)
# Plot actual close if available
if actual_price is not None:
    plt.scatter([input_date], [actual_price], color='yellow', label='Actual Close
Price', s=120, zorder=5)

else:

    print(f"\nNo actual closing price found for {input_date.date()} (possibly a
weekend or holiday).")

# Format x-axis for better readability

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))

plt.gca().xaxis.set_major_locator(mdates.YearLocator(1))

# Plot formatting

plt.title('Apple Stock: Actual vs Predicted Close Prices (2010–2024)',
fontsize=18)

plt.xlabel('Date', fontsize=14)

plt.ylabel('Close Price (USD)', fontsize=14)

plt.xticks(rotation=45, fontsize=10)

plt.yticks(fontsize=12)

plt.legend(fontsize=12)

plt.grid(True, linestyle='--', alpha=0.6)

plt.tight_layout()

plt.show()

# Step 10: Show results

print(f"\nPredicted closing price for {input_date.date()} is:
${round(user_predicted_price, 2)}")

if actual_price is not None:

    print(f"Actual closing price for that date: ${round(actual_price, 2)}")

else:

    print("Actual closing price not available for that date.")
```

Output:



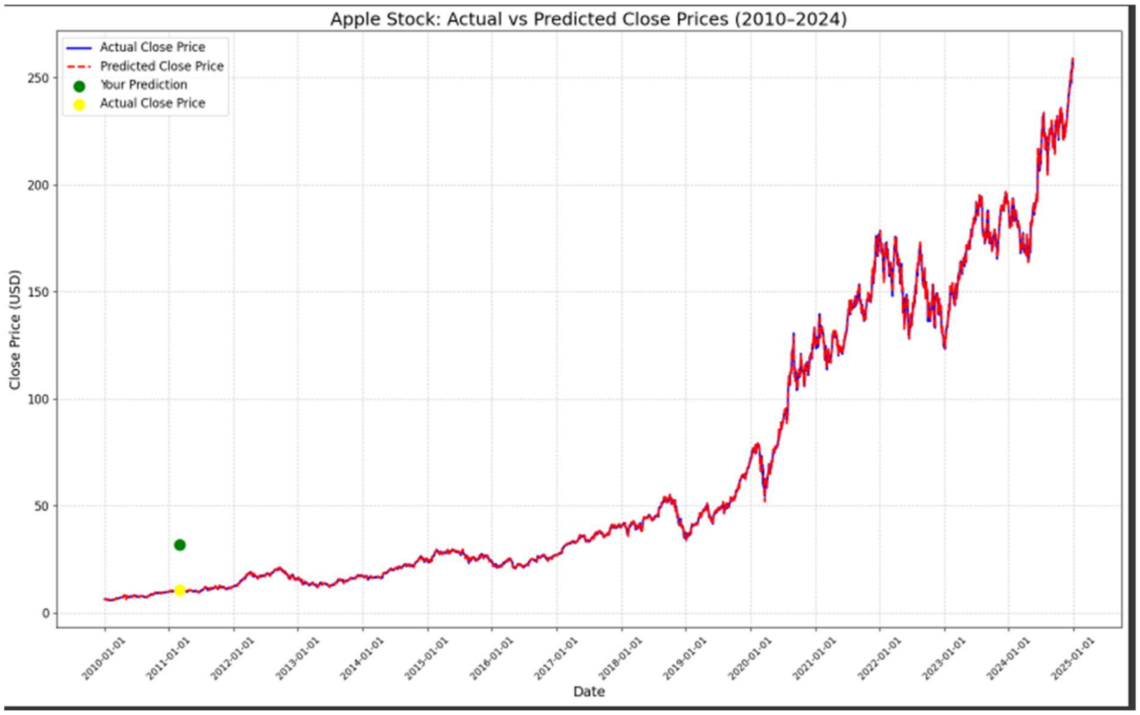
```
/tmp/ipython-input-2528078099.py:8: FutureWarning: YF.download() has changed
  df = yf.download('AAPL', start='2010-01-01', end='2024-12-31')
[*********************100%***********************]  1 of 1 completed

Welcome to the Stock Price Predictor!
Enter the opening price: 40
Enter the high price: 49
Enter the low price: 23
Enter the date for prediction (YYYY-MM-DD): 2011-3-2
```

Visualization (Plotting):



```
Predicted closing price for 2011-03-02 is: $31.73
Actual closing price for that date: $10.58
```

Visualization (Plotting):



Explanation of the graph:

The graph visually represents the historical stock prices of Apple Inc. (AAPL) over the period from 2010 to 2024.

On the x-axis, the timeline (dates) is shown, while the y-axis displays the corresponding closing prices in USD.

This time series plot helps identify trends in the stock's performance—such as periods of growth, decline, or stability.

It provides insights into how Apple's stock has evolved over the years, highlighting major shifts that may correlate with economic events, product launches, or company announcements.

By analysing this graph, one can observe long-term investment patterns and evaluate the stock's overall performance over more than a decade.