MAI272 - Advanced Machine Learning

Register Number : 2448503

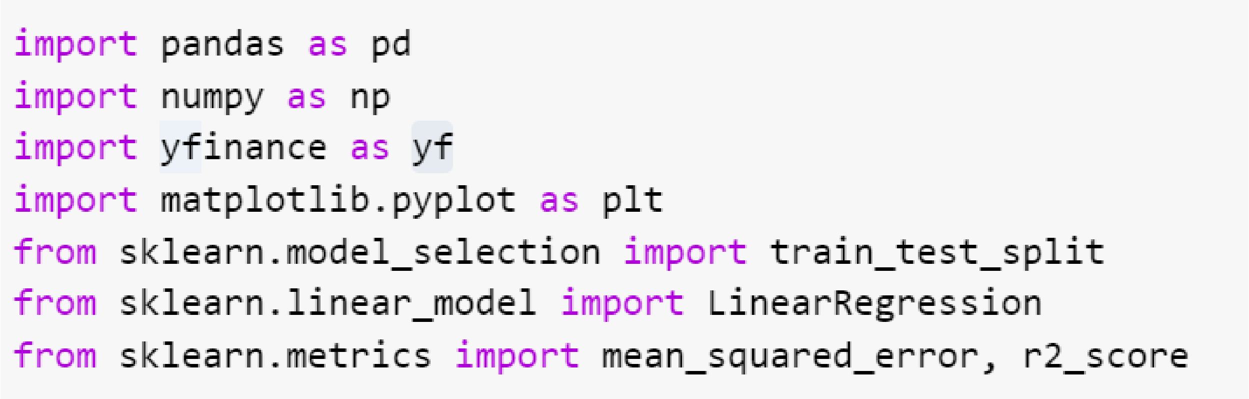
Name : Abhijith E

Experiment Number : 1

Date : 18/10/2024

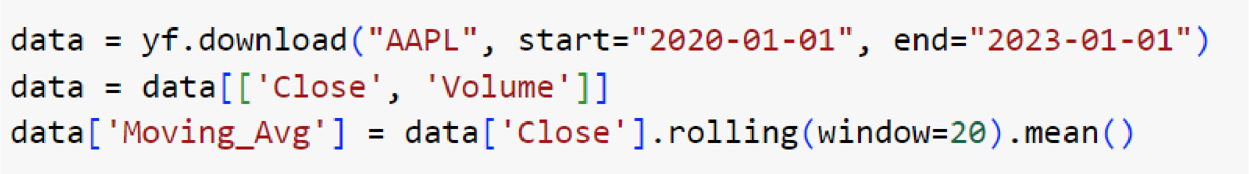
I. .Data Collection and Preprocessing:

1. Importing Required Libraries:



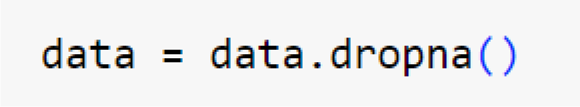
```python
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

2. Collecting Stock Data and Find Moving Average:



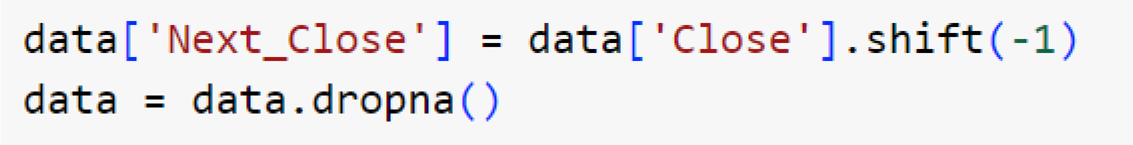
```python
data = yf.download("AAPL", start="2020-01-01", end="2023-01-01")
data = data[['Close', 'Volume']]
data['Moving_Avg'] = data['Close'].rolling(window=20).mean()
```

3. Removing Missing Data:



```python
data = data.dropna()
```

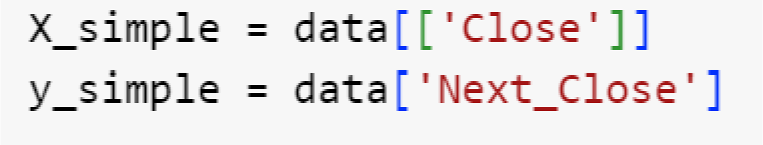
4. Preparing Data for Simple Prediction:



```python
data['Next_Close'] = data['Close'].shift(-1)
data = data.dropna()
```

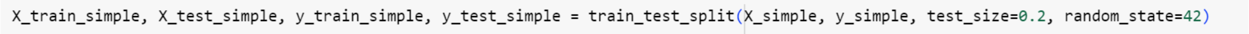
II. .Simple Linear Regression Implementation:

5. Setting Features and Target for Simple Prediction:



```
X_simple = data[['Close']]
y_simple = data['Next_Close']
```

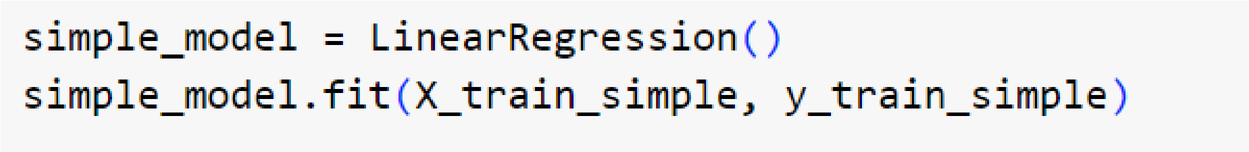
6. Splitting Data into Training and Testing:



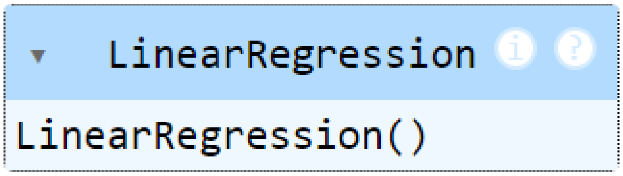
```
X_train_simple, X_test_simple, y_train_simple, y_test_simple = train_test_split(X_simple, y_simple, test_size=0.2, random_state=42)
```

X_train_simple, X_test_simple, y_train_simple, y_test_simple = train_test_split(X_simple, y_simple, test_size=0.2, random_state=42)

7. Training the Model:

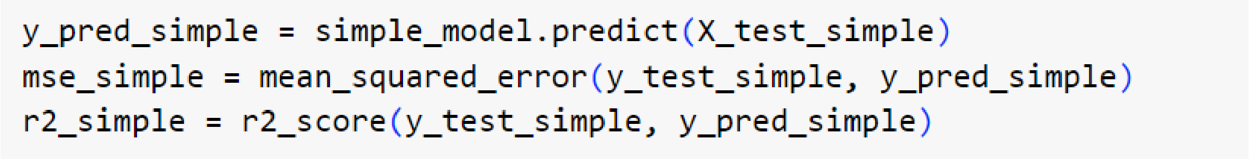


```
simple_model = LinearRegression()
simple_model.fit(X_train_simple, y_train_simple)
```

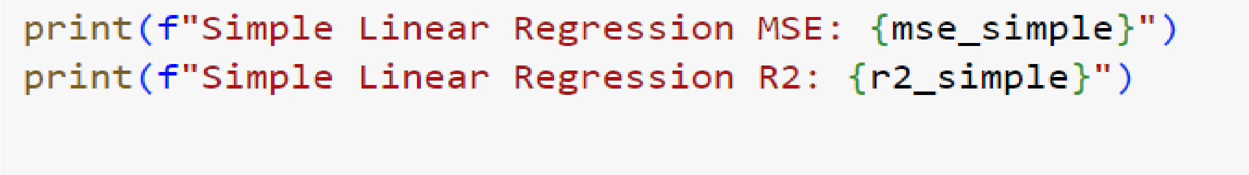


```
▾  LinearRegression
LinearRegression()
```

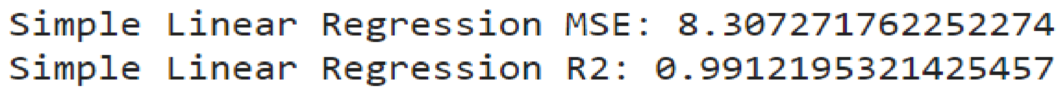
8. Making Predictions and Checking Accuracy:



```
y_pred_simple = simple_model.predict(X_test_simple)
mse_simple = mean_squared_error(y_test_simple, y_pred_simple)
r2_simple = r2_score(y_test_simple, y_pred_simple)
```

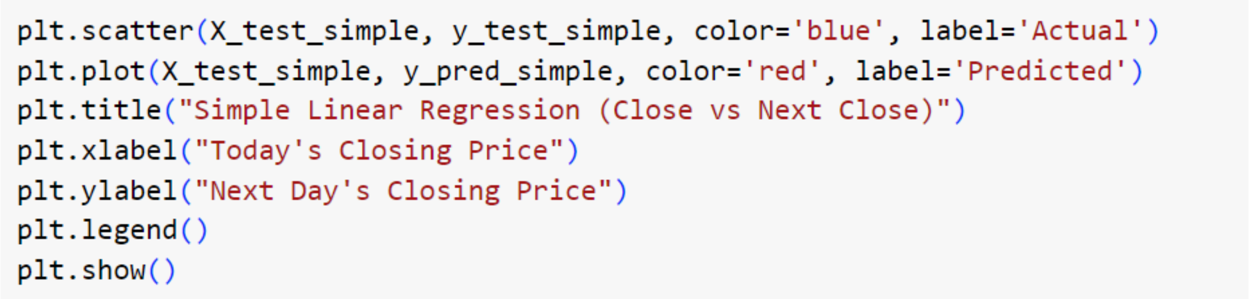


```
print(f"Simple Linear Regression MSE: {mse_simple}")
print(f"Simple Linear Regression R2: {r2_simple}")
```

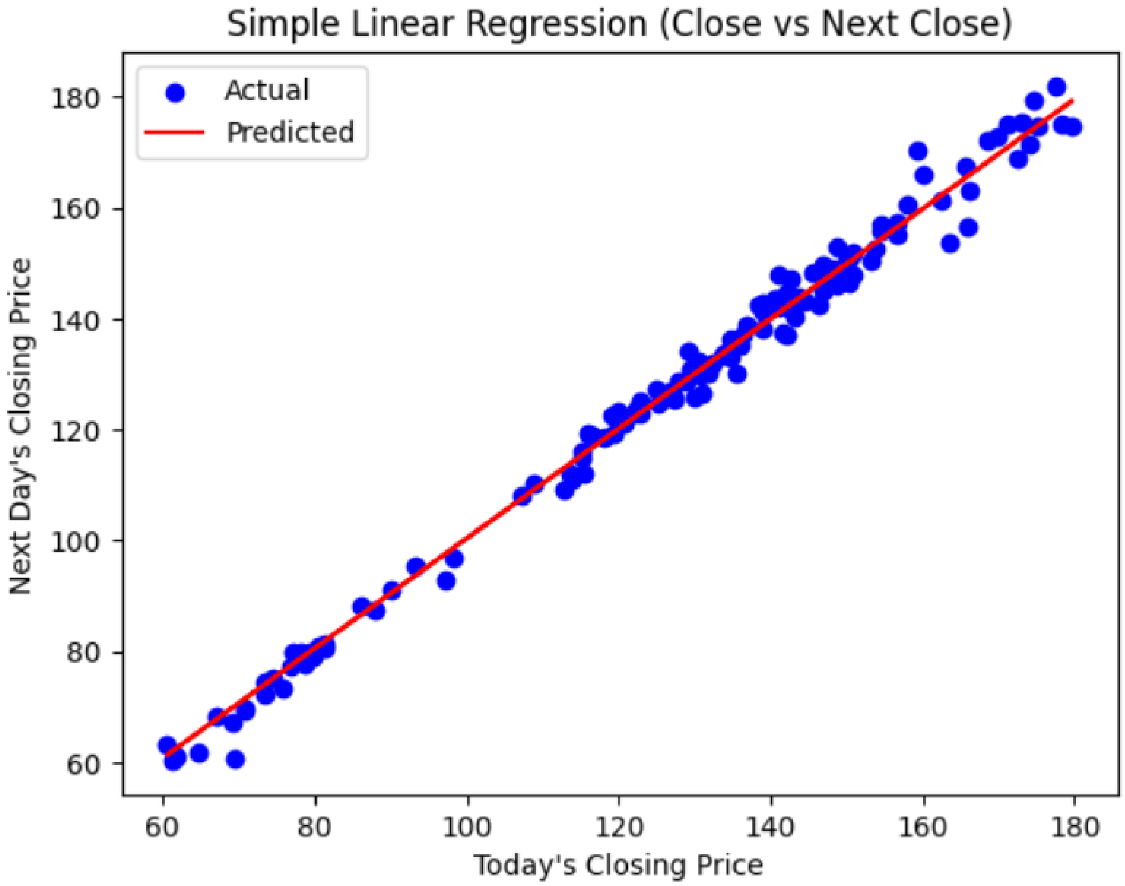


```
Simple Linear Regression MSE: 8.307271762252274
Simple Linear Regression R2: 0.9912195321425457
```

9. Plotting the Simple Linear Regression:

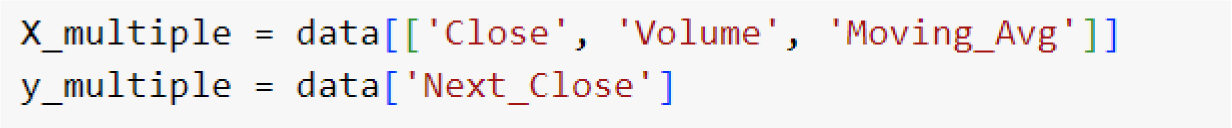


```
plt.scatter(X_test_simple, y_test_simple, color='blue', label='Actual')
plt.plot(X_test_simple, y_pred_simple, color='red', label='Predicted')
plt.title("Simple Linear Regression (Close vs Next Close)")
plt.xlabel("Today's Closing Price")
plt.ylabel("Next Day's Closing Price")
plt.legend()
plt.show()
```



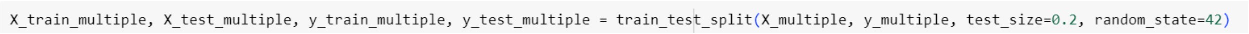
III. .Multiple Linear Regression Implementation:

10. Preparing Data for Multiple Prediction:



```
X_multiple = data[['Close', 'Volume', 'Moving_Avg']]
y_multiple = data['Next_Close']
```

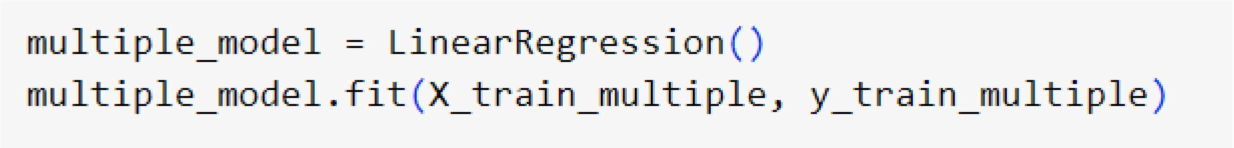
11. Splitting Data Again:



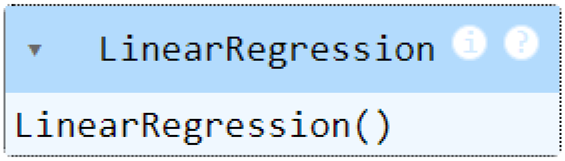
```
X_train_multiple, X_test_multiple, y_train_multiple, y_test_multiple = train_test_split(X_multiple, y_multiple, test_size=0.2, random_state=42)
```

X_train_multiple, X_test_multiple, y_train_multiple, y_test_multiple = train_test_split(X_multiple, y_multiple, test_size=0.2, random_state=42)

12. Training the Multiple Linear Regression Model:

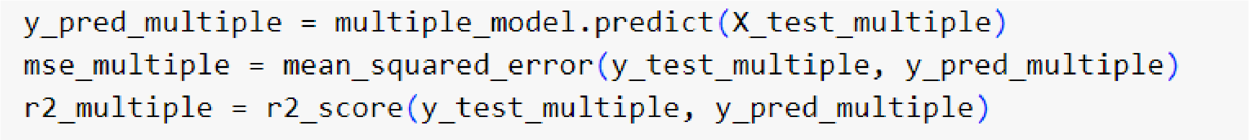


```
multiple_model = LinearRegression()
multiple_model.fit(X_train_multiple, y_train_multiple)
```

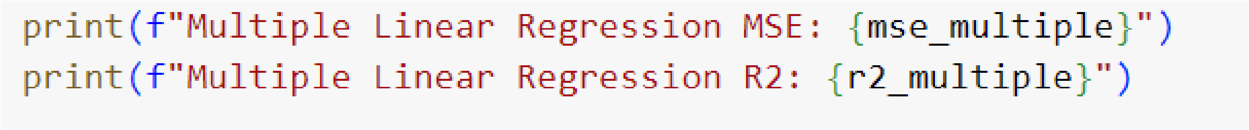


```
LinearRegression()
```

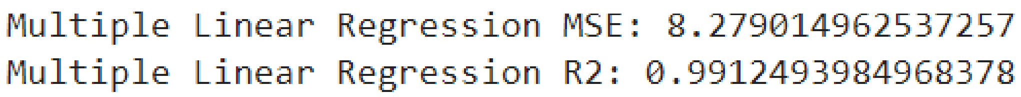
13. Making Predictions and Checking Accuracy (Multiple Features):



```
y_pred_multiple = multiple_model.predict(X_test_multiple)
mse_multiple = mean_squared_error(y_test_multiple, y_pred_multiple)
r2_multiple = r2_score(y_test_multiple, y_pred_multiple)
```

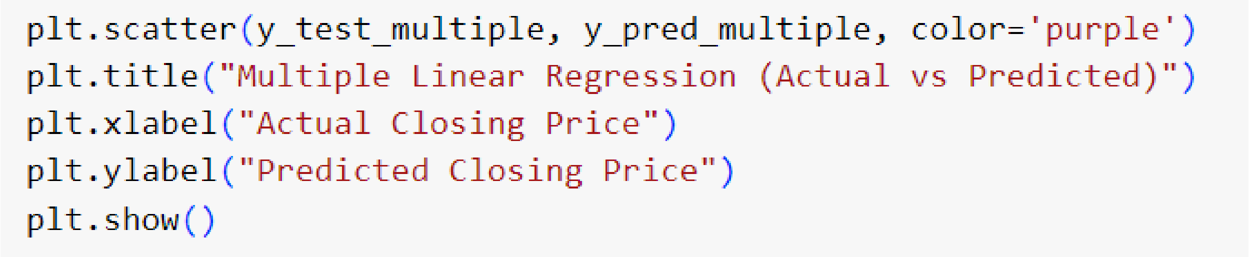


```
print(f"Multiple Linear Regression MSE: {mse_multiple}")
print(f"Multiple Linear Regression R2: {r2_multiple}")
```

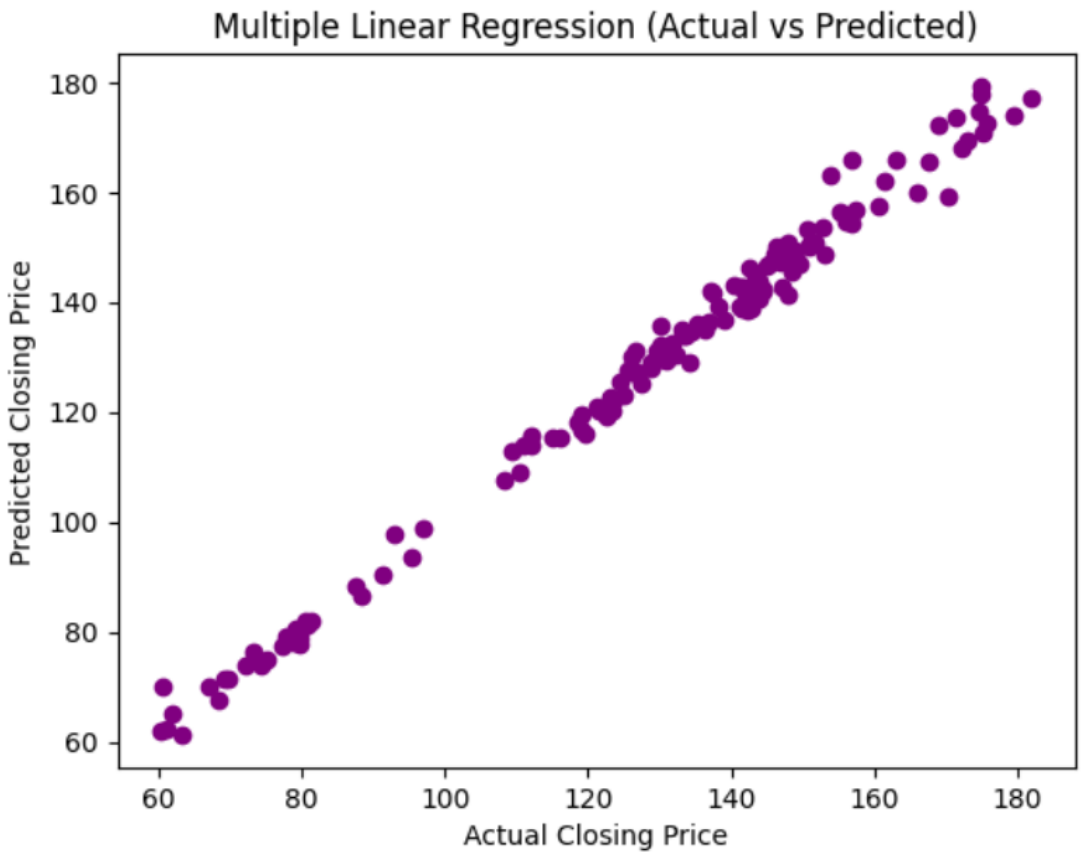


```
Multiple Linear Regression MSE: 8.279014962537257
Multiple Linear Regression R2: 0.9912493984968378
```

14. Plotting the Results for Multiple Linear Regression:



```
plt.scatter(y_test_multiple, y_pred_multiple, color='purple')
plt.title("Multiple Linear Regression (Actual vs Predicted)")
plt.xlabel("Actual Closing Price")
plt.ylabel("Predicted Closing Price")
plt.show()
```


Multiple Linear Regression (Actual vs Predicted)
Predicted Closing Price
Actual Closing Price