

```
In [1]: #Step 1: Import Required Libraries
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

#Step 2: Download Historical Stock Data
# We're using Tesla (TSLA) as an example, you can replace it with any ticker (e.g. AAPL, MSFT)
data = yf.download('TSLA', start='2022-01-01', end='2024-12-31', auto_adjust=False)

#Step 3: Select Features and Clean Data
data = data[['Open', 'High', 'Low', 'Volume', 'Close']]
data.dropna(inplace=True)

#Step 4: Prepare Feature Set (X) and Target Variable (y)
X = data[['Open', 'High', 'Low', 'Volume']]
y = data['Close']

#Step 5: Split Data into Training and Testing Sets (80/20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

#Step 6: Train the Random Forest Regressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train) #FIX: No .ravel() needed

#Step 7: Make Predictions
predictions = model.predict(X_test)

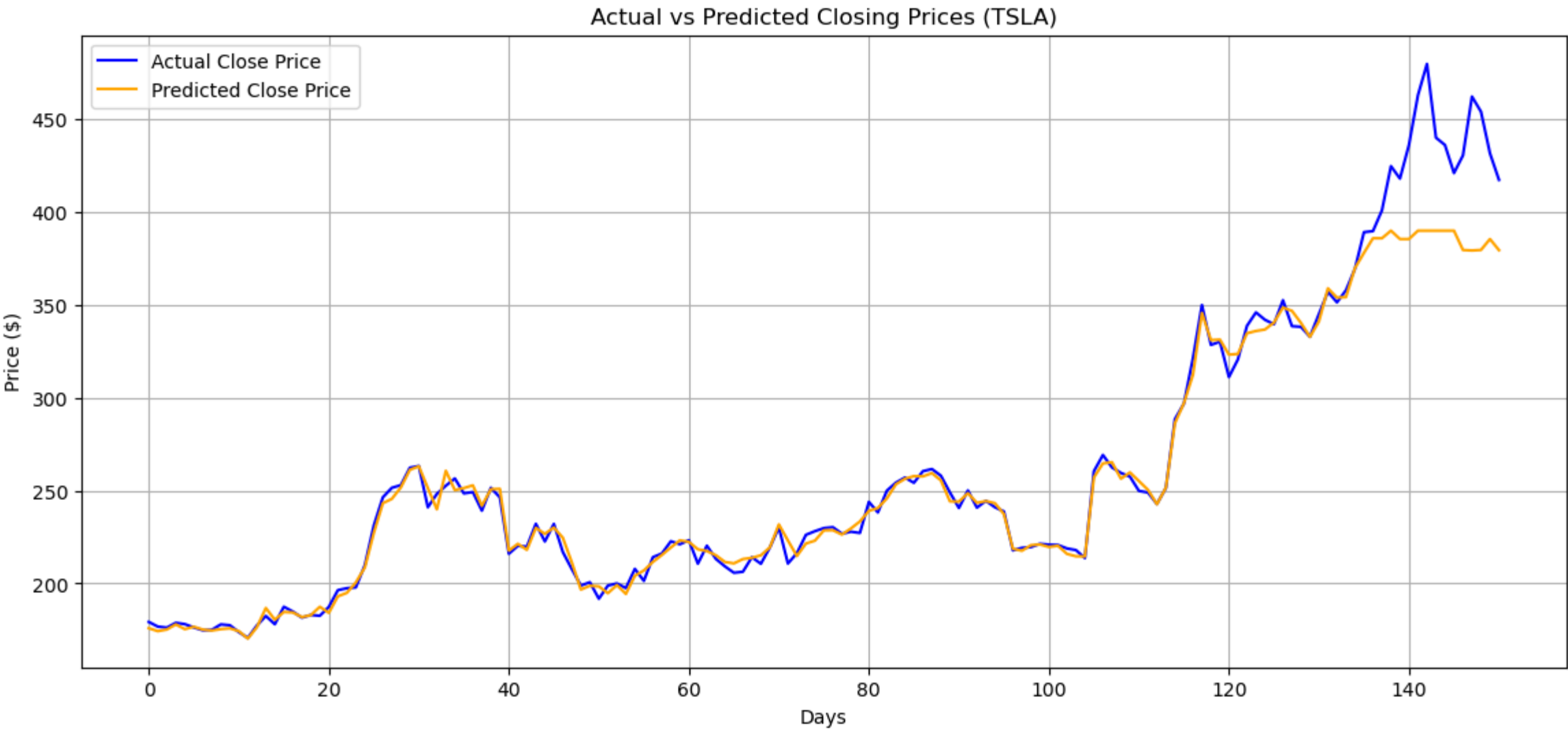
#Step 8: Plot Actual vs Predicted Close Prices
plt.figure(figsize=(14,6))
plt.plot(y_test.values, label='Actual Close Price', color='blue')
plt.plot(predictions, label='Predicted Close Price', color='orange')
plt.title('Actual vs Predicted Closing Prices (TSLA)')
plt.xlabel('Days')
plt.ylabel('Price ($)')
plt.legend()
plt.grid(True)
plt.show()

#Step 9: Predict the Next Day's Closing Price
last_day = X.tail(1)
next_day_price = model.predict(last_day)
print("Predicted Closing Price for Next Day:", round(next_day_price[0], 2))

#(Optional) Linear Regression Comparison
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
lr_predictions = lr_model.predict(X_test)

# Plot for Linear Regression vs Actual
plt.figure(figsize=(14,6))
plt.plot(y_test.values, label='Actual Close Price', color='blue')
plt.plot(lr_predictions, label='Linear Regression Prediction', color='green')
plt.title('Linear Regression: Actual vs Predicted Closing Prices (TSLA)')
plt.xlabel('Days')
plt.ylabel('Price ($)')
plt.legend()
plt.grid(True)
plt.show()

[*****100%*****] 1 of 1 completed
C:\Users\Ommer\anaconda3\Lib\site-packages\sklearn\base.py:1473: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
```



🤖 Predicted Closing Price for Next Day: 379.53

