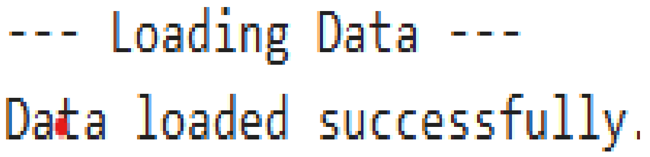
TCS Stock Data- Live and Latest

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
print("--- Loading Data ---")

try:
    # Load the datasets
    history_df = pd.read_csv('TCS_stock_history.csv')
    info_df = pd.read_csv('TCS_stock_info.csv')
    action_df = pd.read_csv('TCS_stock_action.csv')
    print("Data loaded successfully.")

except FileNotFoundError as e:
    print(f"Error: One of the files not found. Please ensure all three CSV files are in the same directory as the script. {e}")
    exit()
```

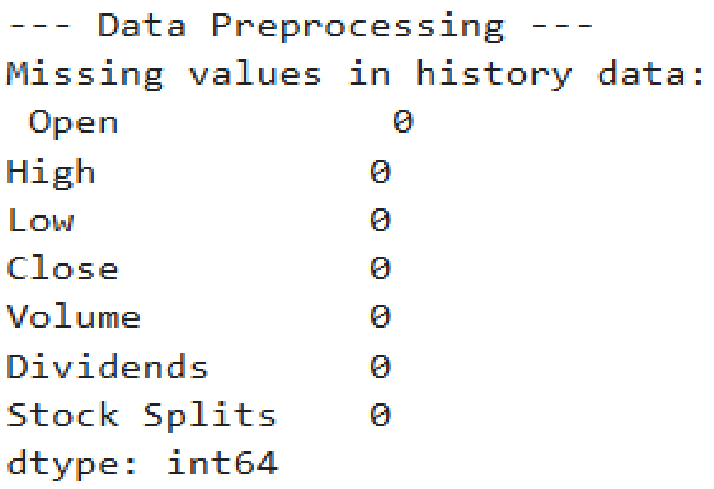
Output:



```
--- Loading Data ---
Data loaded successfully.
```

```
print("\n--- Data Preprocessing ---")
history_df['Date'] = pd.to_datetime(history_df['Date'])
action_df['Date'] = pd.to_datetime(action_df['Date'])
history_df.set_index('Date', inplace=True)
action_df.set_index('Date', inplace=True)
print("Missing values in history data:\n", history_df.isnull().sum())
```

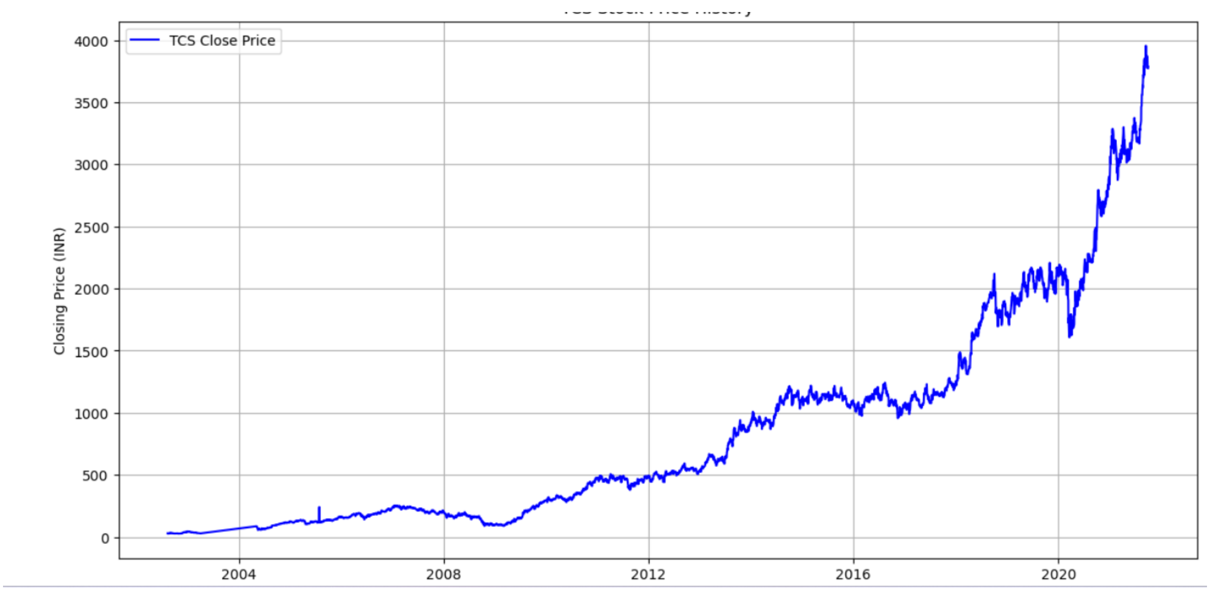
Output:



```
--- Data Preprocessing ---
Missing values in history data:
 Open            0
High            0
Low             0
Close           0
Volume          0
Dividends       0
Stock Splits    0
dtype: int64
```

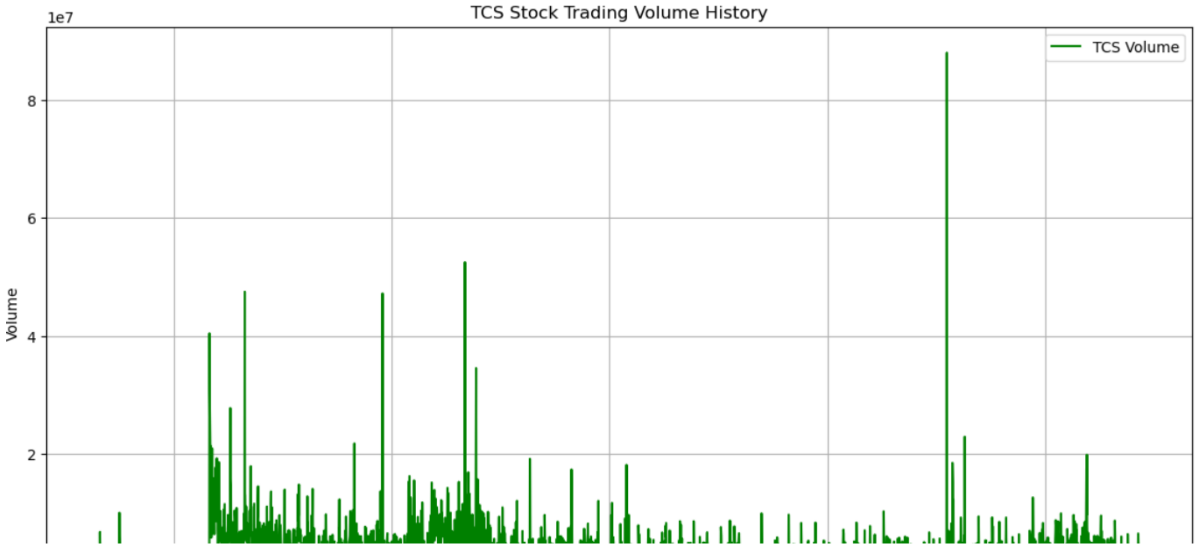
```
print("\n--- Performing EDA and Visualization ---")
plt.figure(figsize=(14, 7))
plt.plot(history_df['Close'], label='TCS Close Price', color='b')
plt.title('TCS Stock Price History')
plt.xlabel('Date')
plt.ylabel('Closing Price (INR)')
plt.legend()
plt.grid(True)
plt.show()
```

Output:



```
plt.figure(figsize=(14, 7))
plt.plot(history_df['Volume'], label='TCS Volume', color='g')
plt.title('TCS Stock Trading Volume History')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.legend()
plt.grid(True)
plt.show()
```

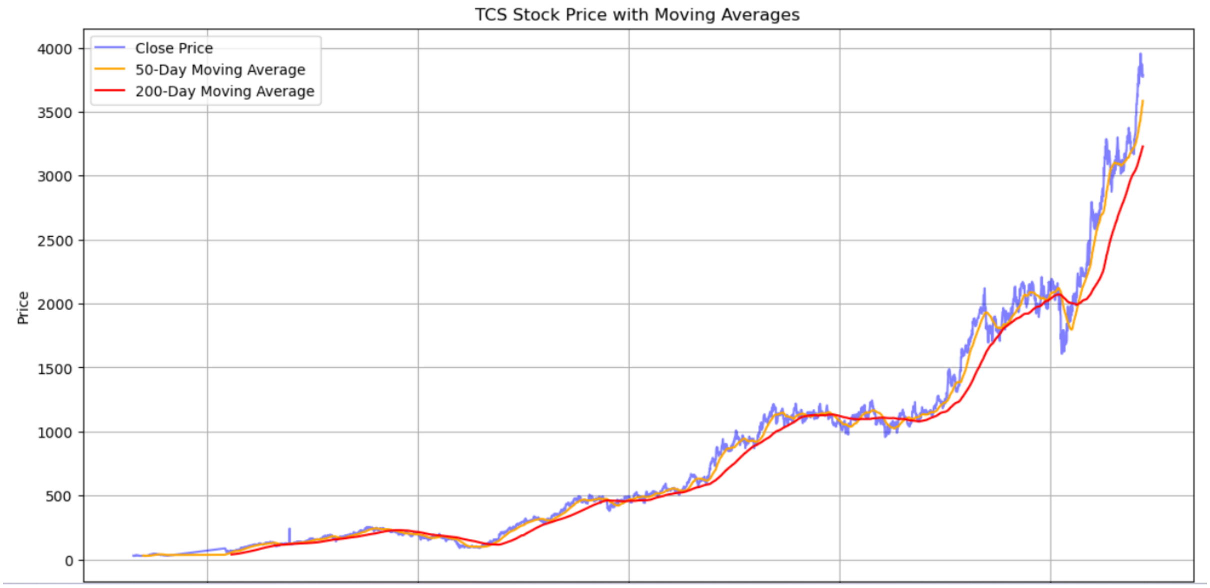
Output:



```
history_df['50_MA'] = history_df['Close'].rolling(window=50).mean()
history_df['200_MA'] = history_df['Close'].rolling(window=200).mean()


plt.figure(figsize=(14, 7))
plt.plot(history_df['Close'], label='Close Price', color='b', alpha=0.5)
plt.plot(history_df['50_MA'], label='50-Day Moving Average', color='orange')
plt.plot(history_df['200_MA'], label='200-Day Moving Average', color='red')
plt.title('TCS Stock Price with Moving Averages')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid(True)
plt.show()
```

Output:

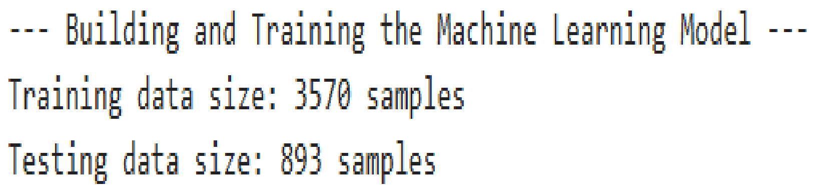


```
print("\n--- Building and Training the Machine Learning Model ---")
features = ['Open', 'High', 'Low', 'Volume']
target = 'Close'

X = history_df[features].dropna()
y = history_df.loc[X.index, target]
split_index = int(0.8 * len(X))
X_train, X_test = X[:split_index], X[split_index:]
y_train, y_test = y[:split_index], y[split_index:]

print(f"Training data size: {len(X_train)} samples")
print(f"Testing data size: {len(X_test)} samples")
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Output:



```
--- Building and Training the Machine Learning Model ---
Training data size: 3570 samples
Testing data size: 893 samples
```

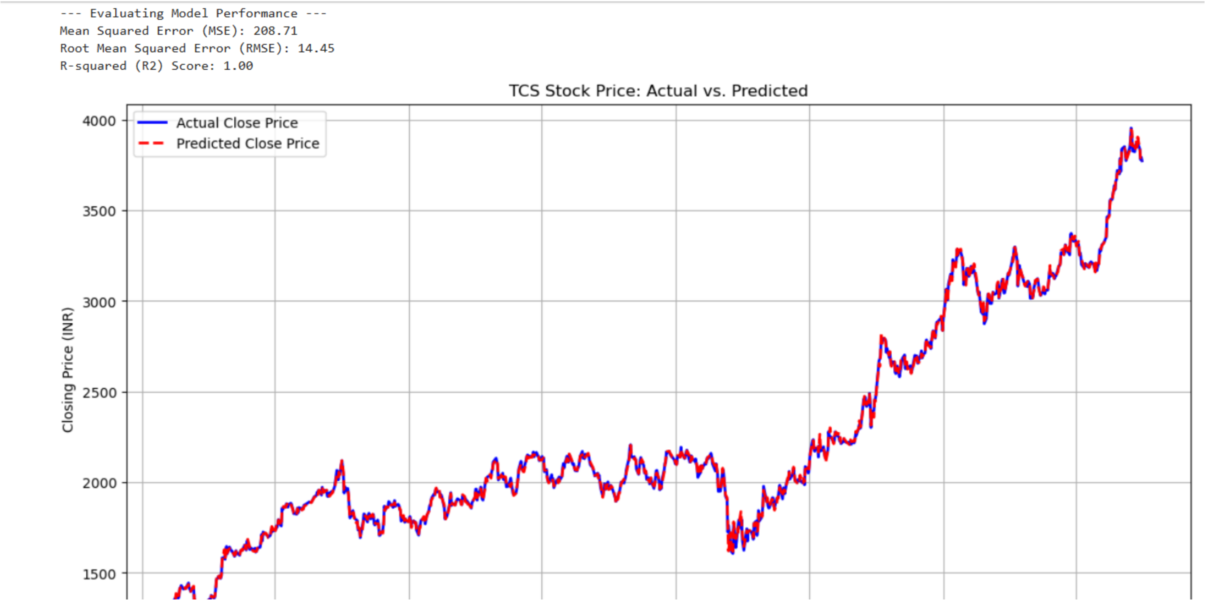
```
print("\n--- Evaluating Model Performance ---")
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")

plt.figure(figsize=(14, 7))
plt.plot(y_test.index, y_test.values, label='Actual Close Price', color='blue',
linewidth=2)
plt.plot(y_test.index, y_pred, label='Predicted Close Price', color='red',
linestyle='--', linewidth=2)
plt.title('TCS Stock Price: Actual vs. Predicted')
plt.xlabel('Date')
plt.ylabel('Closing Price (INR)')
plt.legend()
plt.grid(True)
plt.show()

print("\nProject complete. The script has performed data analysis, visualization,
and a predictive model.")
```

Output:



```
--- Evaluating Model Performance ---
Mean Squared Error (MSE): 208.71
Root Mean Squared Error (RMSE): 14.45
R-squared (R2) Score: 1.00
```