

```
In [ ]: NAME : SHINDE SHUBHAM DNYANDEV, ROLL NO. : EN23107121, BATCH : C
```

```
In [1]: import pandas as pd
```

```
In [3]: df = pd.read_csv("/home/admin1/RELIANCE.NS_1973-05-08_2025-03-01.csv")  
df
```

```
Out[3]:
```

	date	open	high	low	close	adj_close
0	1996-01-01 00:00:00+05:30	7.319124	7.358397	7.270925	7.345901	3.353593
1	1996-01-02 00:00:00+05:30	7.328050	7.363753	7.235222	7.288776	3.327513
2	1996-01-03 00:00:00+05:30	7.408381	7.745775	7.328050	7.344116	3.352778
3	1996-01-04 00:00:00+05:30	7.274495	7.297702	7.178097	7.276280	3.321809
4	1996-01-05 00:00:00+05:30	7.247718	7.247718	7.163816	7.226296	3.298990
...
7319	2025-02-21 00:00:00+05:30	1228.699951	1240.000000	1222.150024	1228.150024	1228.150024
7320	2025-02-24 00:00:00+05:30	1216.550049	1223.250000	1210.500000	1214.550049	1214.550049
7321	2025-02-25 00:00:00+05:30	1211.000000	1221.000000	1201.500000	1204.000000	1204.000000
7322	2025-02-27 00:00:00+05:30	1212.800049	1215.000000	1200.650024	1207.099976	1207.099976
7323	2025-02-28 00:00:00+05:30	1202.000000	1217.349976	1193.300049	1200.099976	1200.099976

7324 rows × 7 columns

```
In [5]: df.describe()
```

```
Out[5]:
```

	open	high	low	close	adj_close	volume
count	7324.000000	7324.000000	7324.000000	7324.000000	7324.000000	7.324000e+03
mean	326.804309	330.415930	322.919029	326.516915	312.722616	5.938802e+07
std	402.088964	405.818972	398.104232	401.798753	402.816495	1.020567e+08
min	5.462565	5.487558	5.428648	5.487558	2.505212	0.000000e+00
25%	24.984990	25.477694	24.527992	24.952858	16.965307	1.366736e+07
50%	197.244431	199.758835	194.650009	197.067268	176.555252	2.429788e+07
75%	384.945511	389.145729	381.762489	384.688339	370.944550	6.205445e+07
max	1604.449951	1608.800049	1585.500000	1600.900024	1595.484985	1.448889e+09

```
In [7]: df.isnull().sum()
```

```
Out[7]: date      0  
open      0  
high      0  
low       0  
close     0  
adj_close 0  
volume    0  
dtype: int64
```

```
In [9]: df.dtypes
```

```
Out[9]: date        object  
open       float64  
high       float64  
low        float64  
close      float64  
adj_close   float64  
volume     int64  
dtype: object
```

```
In [11]: df["date"] = pd.to_datetime(df["date"])  
df
```

```
Out[11]:
```

	date	open	high	low	close	adj_close
0	1996-01-01 00:00:00+05:30	7.319124	7.358397	7.270925	7.345901	3.353593
1	1996-01-02 00:00:00+05:30	7.328050	7.363753	7.235222	7.288776	3.327513
2	1996-01-03 00:00:00+05:30	7.408381	7.745775	7.328050	7.344116	3.352778
3	1996-01-04 00:00:00+05:30	7.274495	7.297702	7.178097	7.276280	3.321809
4	1996-01-05 00:00:00+05:30	7.247718	7.247718	7.163816	7.226296	3.298990
...
7319	2025-02-21 00:00:00+05:30	1228.699951	1240.000000	1222.150024	1228.150024	1228.150024
7320	2025-02-24 00:00:00+05:30	1216.550049	1223.250000	1210.500000	1214.550049	1214.550049
7321	2025-02-25 00:00:00+05:30	1211.000000	1221.000000	1201.500000	1204.000000	1204.000000
7322	2025-02-27 00:00:00+05:30	1212.800049	1215.000000	1200.650024	1207.099976	1207.099976
7323	2025-02-28 00:00:00+05:30	1202.000000	1217.349976	1193.300049	1200.099976	1200.099976

7324 rows × 7 columns

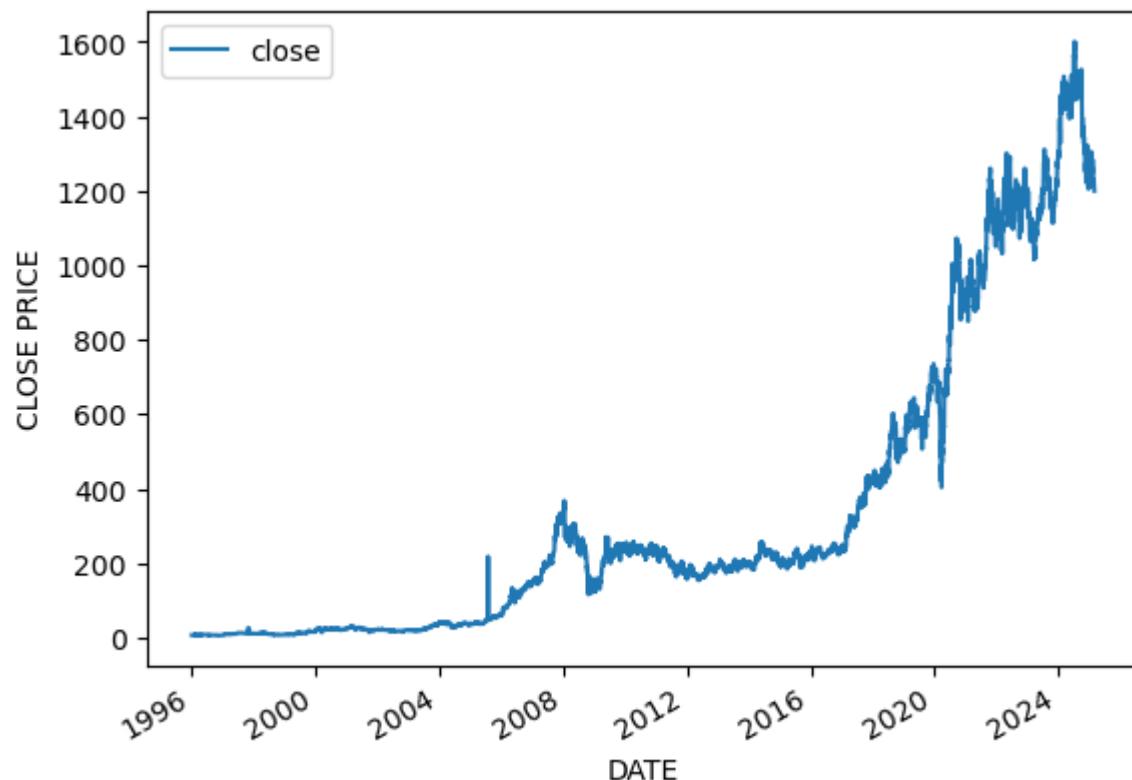
```
In [13]: df.dtypes
```

```
Out[13]: date        datetime64[ns, UTC+05:30]  
open       float64  
high       float64  
low        float64  
close      float64  
adj_close   float64  
volume     int64  
dtype: object
```

```
In [15]: import matplotlib.pyplot as plt
```

```
In [17]: df.plot(x = "date" , y = "close" , xlabel = "DATE" , ylabel = "CLOSE PRICE")
```

```
Out[17]: <Axes: xlabel='DATE', ylabel='CLOSE PRICE'>
```



```
In [19]: df["SMA_50"] = df["close"].rolling(window = 50).mean()  
df["SMA_50"]
```

```
Out[19]: 0           NaN  
1           NaN  
2           NaN  
3           NaN  
4           NaN  
...  
7319    1244.585000  
7320    1243.510000  
7321    1242.683999  
7322    1241.760999  
7323    1241.153999  
Name: SMA_50, Length: 7324, dtype: float64
```

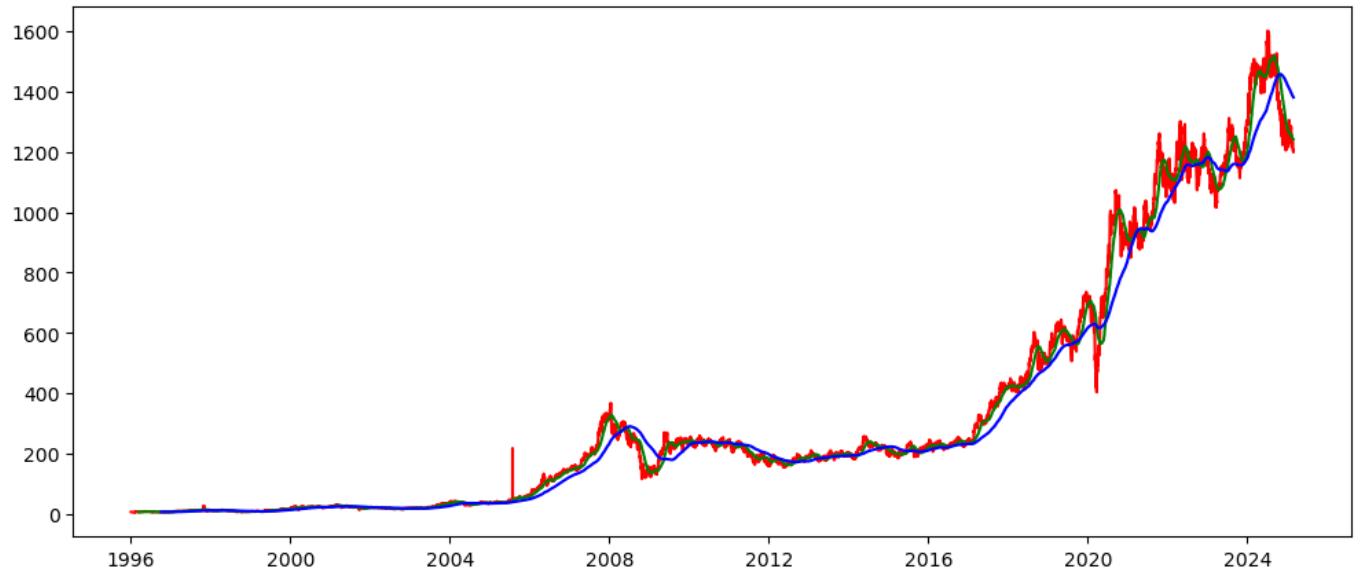
```
In [21]: df["SMA_200"] = df["close"].rolling(window = 200).mean()  
df["SMA_200"]
```

```
Out[21]: 0           NaN  
1           NaN  
2           NaN  
3           NaN  
4           NaN  
...  
7319    1384.737502  
7320    1383.839627  
7321    1382.822502  
7322    1381.844502  
7323    1380.744627  
Name: SMA_200, Length: 7324, dtype: float64
```

```
In [23]: plt.figure(figsize = (12, 5))  
plt.plot(df["date"], df["close"], color="Red")
```

```
plt.plot(df["date"], df["SMA_50"], color="Green")
plt.plot(df["date"], df["SMA_200"], color="Blue")
```

Out[23]: [`<matplotlib.lines.Line2D at 0x7f7e087361f0>`]



In [25]: `from statsmodels.tsa.ar_model import AutoReg`

In [27]: `train_size = int(len(df)*0.8)
train, test = df['close'][:train_size], df['close'][train_size:]`

In [37]: `model = AutoReg(train, lags = 52).fit()`

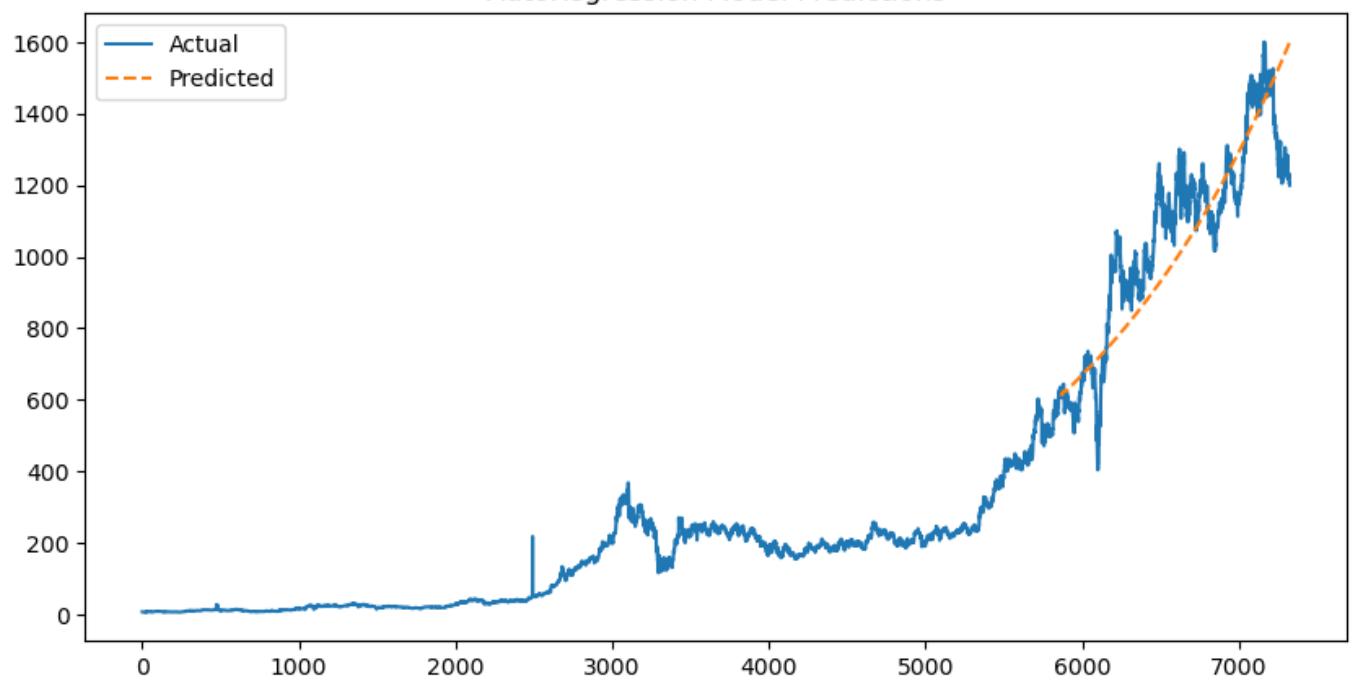
In [39]: `predictions = model.predict(start = len(train), end = len(df)-1, dynamic = False)`

In [41]: `predictions`

```
Out[41]: 5859      617.260907
5860      617.280732
5861      617.252425
5862      617.233863
5863      616.437216
...
7319      1595.857372
7320      1596.902456
7321      1597.948218
7322      1598.994659
7323      1600.041779
Length: 1465, dtype: float64
```

In [43]: `plt.figure(figsize = (10, 5))
plt.plot(df.index, df['close'], label="Actual")
plt.plot(test.index, predictions, label="Predicted", linestyle="dashed")
plt.legend()
plt.title("AutoRegression Model Predictions")
plt.show()`

AutoRegression Model Predictions



In []: