

Experiment 8: ARIMA Model for Time Series Forecasting

Aim:

To create an ARIMA model for univariate time series forecasting using Microsoft stock dataset.

1. Importing Required Libraries

Python

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
```

2. Loading the Dataset

Python

```
msft = yf.download("MSFT", start="2015-01-01", end="2024-12-31")
data = msft[['Close']].dropna()
```

3. Data Preparation

- Used only the 'Close' column for univariate time series.
- Split the dataset into **training (90%)** and **testing (10%)**:

Python

```
train_size = int(len(data) * 0.9)
train, test = data[:train_size], data[train_size:]
```

4. Building and Fitting ARIMA Model

Python

```
model = ARIMA(train['Close'], order=(5, 1, 2)) # p=5, d=1, q=2
model_fit = model.fit()
```

5. Forecasting

Python

```
forecast = model_fit.forecast(steps=len(test))
```

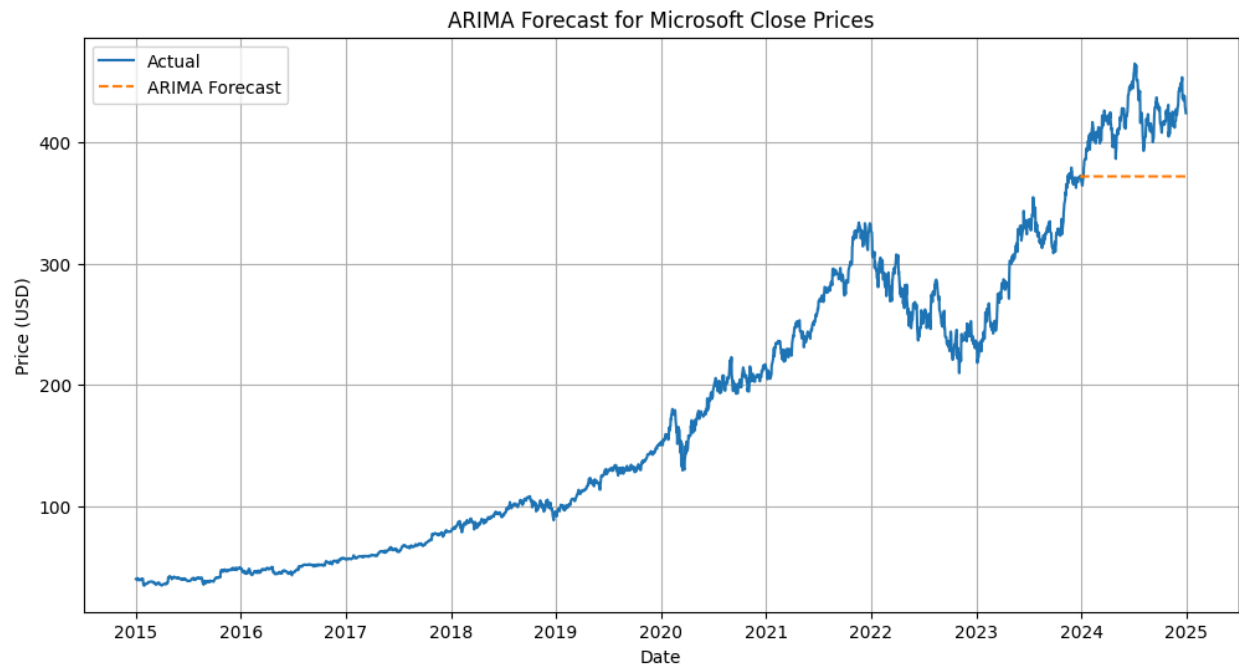
- Forecasted values were plotted against actual test values.
 - This helped evaluate the accuracy of the ARIMA model visually.
-

Forecast Visualization:

You can use a plot like this:

Python

```
plt.figure(figsize=(12,6))
plt.plot(data.index, data['Close'], label='Actual')
plt.plot(test.index, forecast, label='ARIMA Forecast',
         linestyle='dashed')
plt.title("ARIMA Forecast - Microsoft Close Prices")
plt.xlabel("Date")
plt.ylabel("Price (USD)")
plt.legend()
plt.grid(True)
plt.show()
```



Result:

Thus, an ARIMA model was successfully created and used to forecast future values in the Microsoft stock dataset.