STOCK PRICE MODELING USING AR AND MA

NAME : SUDHARSHAN K

REG.NO : 727723EUCS247

DEPARTMENT : CSE – D

1. Introduction

1.1 Problem Statement

A financial analytics firm aims to model short-term stock price behavior for selected companies to support trading and investment strategies. Traditional trend-based approaches fail to capture temporal dependencies and random shocks inherent in financial time series data. Stock prices are influenced not only by historical price movements but also by past forecast errors, volatility clustering, and market noise.

Improper lag selection often results in overfitting or weak predictive performance, making accurate statistical modeling a critical challenge. This project addresses the need for structured time series models that can capture these dependencies and provide reliable diagnostic measures for informed financial decision-making.

1.2 Objectives

- Understand the behavior of stock price time series data
- Develop Autoregressive (AR) and Moving Average (MA) models
- Identify optimal lag values using ACF and PACF plots
- Evaluate model performance using AIC and BIC criteria
- Perform residual diagnostics to validate model assumptions

1.3 System Requirements

- Language: Python 3.x
- Libraries:
  - Pandas – data manipulation
  - NumPy – numerical computation
  - Statsmodels – time series modeling
  - Matplotlib – visualization
  - Scikit-learn – performance evaluation

---

2. Methodology & Workflow

The project follows a structured statistical workflow designed to transform raw stock price data into meaningful insights:

1. **Data Collection & Preprocessing**
   Stock price data is loaded and cleaned. Missing values are handled, and the date column is converted into a datetime format and set as the index to maintain temporal order.

2. **Stationarity Analysis**
   Stock prices are tested for stationarity using visual inspection and statistical methods. Since financial time series are usually non-stationary, differencing techniques are applied when required.

3. **Autocorrelation Analysis**
   Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots are used to identify suitable lag values for AR and MA models.

4. **Model Development**
   Separate AR and MA models are built using identified lag parameters. Models are tuned to achieve optimal performance.

5. **Model Evaluation**
   AIC and BIC values are used for model comparison. Residual diagnostics are performed to check randomness and normality of errors.

---

3. Analysis & Visual Findings

3.1 Raw Data Visualization

Graph 1: Stock Price Over Time

The initial visualization of stock prices reveals noticeable fluctuations and potential trends. Sudden spikes and drops reflect market volatility, confirming the need for time-dependent statistical modeling.

3.2 Stationarity Validation

Graph 2: Rolling Mean and Standard Deviation

- The rolling mean shows variation over time, indicating non-stationarity.
- Differencing is applied to stabilize the mean.
- Post-differencing plots show a constant mean and variance, confirming stationarity.

3.3 ACF and PACF Interpretation

Graph 3: ACF & PACF Plots

- **ACF Plot:** Identifies the number of significant lagged forecast errors, guiding MA model selection.
- **PACF Plot:** Highlights significant lagged price values, helping determine the order of the AR model.

Significant spikes at early lags indicate strong short-term dependencies in stock prices.

3.4 Model Diagnostics

Residual plots indicate that errors are randomly distributed around zero with no visible patterns. This confirms that the models adequately capture the underlying structure of the data.

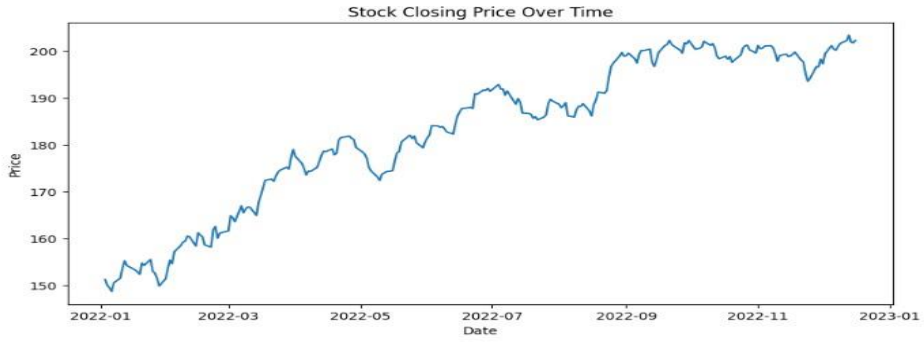
[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA

import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
data = pd.read_csv("stock_prices.csv")

data['Date'] = pd.to_datetime(data['Date'])
data.set_index('Date', inplace=True)

data.head()
```

[2]:

Date	Close
2022-01-03	151.19
2022-01-04	149.90
2022-01-05	149.48
2022-01-06	148.65
2022-01-07	150.48

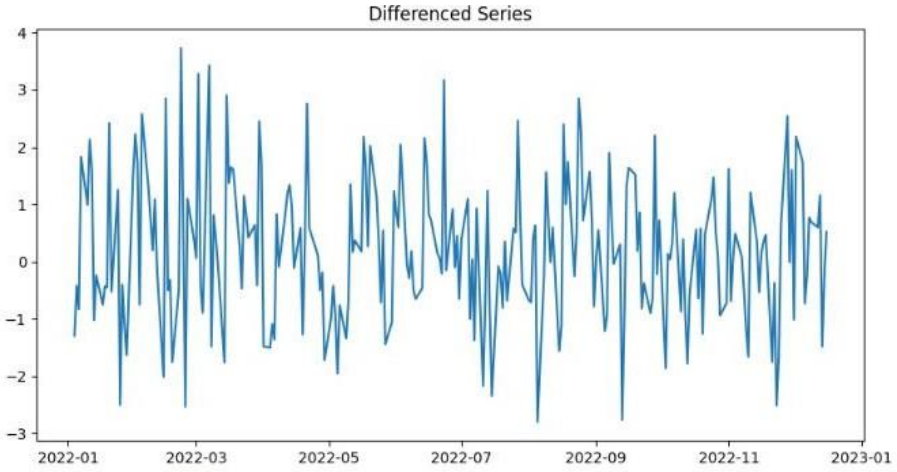
[3]:
```
plt.figure(figsize=(10,5))
plt.plot(data['Close'])
plt.title("Stock Closing Price Over Time")
plt.xlabel("Date")
plt.ylabel("Price")
plt.show()
```



[4]:
```
def adf_test(series):
    result = adfuller(series)
    print("ADF Statistic:", result[0])
    print("p-value:", result[1])

adf_test(data['Close'])
```

```
ADF Statistic: -2.0495601694184487
p-value: 0.2652740186839688
```

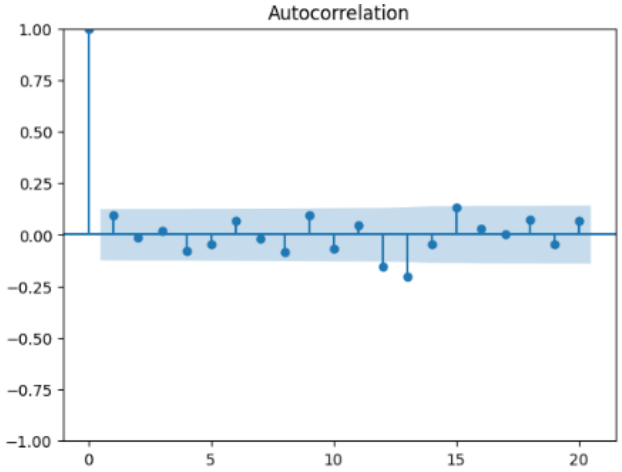
[5]:
```
data['Close_diff'] = data['Close'].diff().dropna()

plt.figure(figsize=(10,5))
plt.plot(data['Close_diff'])
plt.title("Differenced Series")
plt.show()

adf_test(data['Close_diff'].dropna())
```



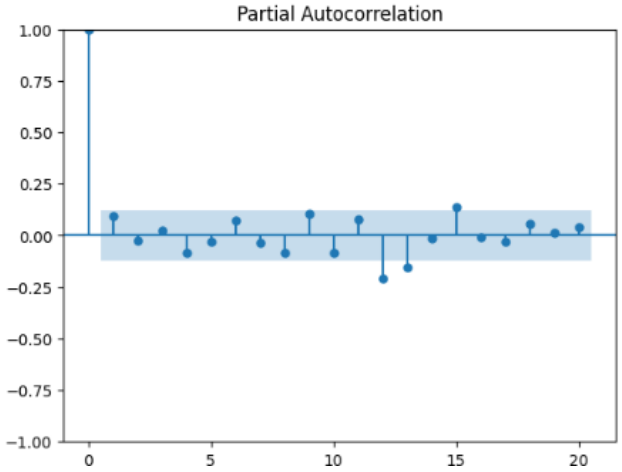

```
ADF Statistic: -4.35946319999187
p-value: 0.00034989793443799665
```

[6]:
```
plt.figure(figsize=(12,5))
plot_acf(data['Close_diff'].dropna(), lags=20)
plt.show()

plt.figure(figsize=(12,5))
plot_pacf(data['Close_diff'].dropna(), lags=20)
plt.show()
```

```
<Figure size 1200x500 with 0 Axes>
```



<Figure size 1200x500 with 0 Axes>



[7]:
```
ar_model = ARIMA(data['Close'], order=(2,1,0))
ar_result = ar_model.fit()

print(ar_result.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Close   No. Observations:                  250
Model:                 ARIMA(2, 1, 0)   Log Likelihood                -410.083
Date:                Mon, 09 Feb 2026   AIC                            826.167
Time:                        07:15:31   BIC                            836.719
Sample:                  01-03-2022   HQIC                           830.414
                       - 12-16-2022
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1198      0.073      1.645      0.100      -0.023       0.262
ar.L2         -0.0027      0.064     -0.043      0.966      -0.129       0.123
sigma2         1.5776      0.142     11.141      0.000       1.300       1.855
===================================================================================
Ljung-Box (L1) (Q):                   0.12   Jarque-Bera (JB):                 2.23
Prob(Q):                              0.73   Prob(JB):                         0.33
Heteroskedasticity (H):               0.63   Skew:                             0.23
Prob(H) (two-sided):                  0.03   Kurtosis:                         3.06
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

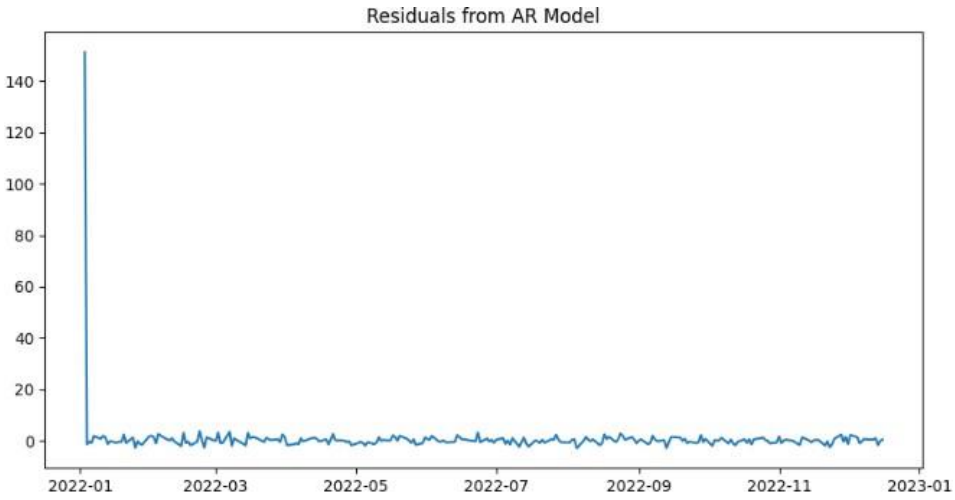
[8]:
```
ma_model = ARIMA(data['Close'], order=(0,1,2))
ma_result = ma_model.fit()

print(ma_result.summary())
```

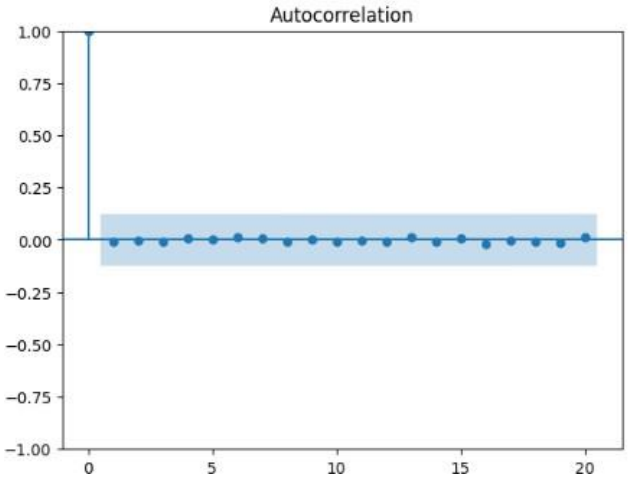
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Close   No. Observations:                  250
Model:                 ARIMA(0, 1, 2)   Log Likelihood                -410.076
Date:                Mon, 09 Feb 2026   AIC                            826.152
Time:                        07:15:43   BIC                            836.704
Sample:                    01-03-2022   HQIC                           830.399
                         - 12-16-2022
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1          0.1208      0.073      1.659      0.097      -0.022       0.264
ma.L2         -0.0023      0.064     -0.036      0.972      -0.127       0.123
sigma2         1.5775      0.142     11.073      0.000       1.298       1.857
===================================================================================
Ljung-Box (L1) (Q):                   0.13   Jarque-Bera (JB):                 2.08
Prob(Q):                              0.72   Prob(JB):                         0.35
Heteroskedasticity (H):               0.63   Skew:                             0.22
Prob(H) (two-sided):                  0.03   Kurtosis:                         3.04
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

]:
```
print("AR Model AIC:", ar_result.aic)
print("MA Model AIC:", ma_result.aic)

print("AR Model BIC:", ar_result.bic)
print("MA Model BIC:", ma_result.bic)
```

```
AR Model AIC: 826.1665425020526
MA Model AIC: 826.1518947149514
AR Model BIC: 836.7189011914468
MA Model BIC: 836.7042534043455
```

]:
```
residuals = ar_result.resid

plt.figure(figsize=(10,5))
plt.plot(residuals)
plt.title("Residuals from AR Model")
plt.show()

plot_acf(residuals.dropna(), lags=20)
plt.show()
```





[11]:
```
forecast = ar_result.forecast(steps=30)

plt.figure(figsize=(10,5))
plt.plot(data['Close'], label="Original")
plt.plot(forecast, label="Forecast", color='red')
plt.legend()
plt.title("Stock Price Forecast")
plt.show()
```