SIGMAWEDGE

Extracting Data from Quantrocket:

Price data (daily close prices only) for Apple stock (sid='AAPL') for the year 2023 (01-01-2023 to 12-31-2023)data has been extracted from Quantrocket.

Importing Libraries and reading data:

Necessary libraries have been imported and the data read has been converted into a dataframe.

Model

Stock Price Model Inference:

This Python class, named Model, encapsulates a simplistic Stock Price model designed to maximize portfolio value based on a straightforward trading strategy. The key functionalities are outlined below:

- calculate_returns(self, price_history): Computes daily returns from Close prices using the formula: r(d) = p(d) - p(d-1)p(d-1).
- classify_state(self, returns): Classifies daily returns into three states: Bull market (+1), Flat market (0), or Bear market (-1). Classification is based on specified threshold values.
- calculate_transition_distribution(self, states): Derives a transition distribution matrix representing the likelihood of transitioning between market states.
- maximize_portfolio_value(self, price_history): Implements the main strategy to maximize portfolio value. Buy signals occur when transitioning from a flat market to a bull market, and sell signals occur when transitioning to a bear market. The method returns the final portfolio value, indices of buy signals, a list of portfolio values at each step, and the transition distribution matrix.

Fitting the Model for the Given Dataset:

Following model fitting, key insights include:

- maximized portfolio value
- optimal buy indices and dates
- transition distribution matrix

Test For Arima:

The ARIMA (Auto Regressive Integrated Moving Average) model is used for time series forecasting, specifically for predicting future values based on historical observations. It combines autoregressive (AR) and moving average (MA) components, allowing it to capture trends, seasonality, and temporal dependencies in time series data.

Test Type:

- KPSS test is employed to examine the stationarity of a time series.

Hypotheses:

- Null Hypothesis (H0): The time series is stationary around a deterministic trend.
- Alternative Hypothesis (H1): The time series has a unit root and is non-stationary.

Test Results:

```
Results of KPSS Test:
Test Statistic             1.715955
p-value                    0.010000
#Lags Used                10.000000
Critical Value (10%)       0.347000
Critical Value (5%)        0.463000
Critical Value (2.5%)      0.574000
Critical Value (1%)        0.739000
dtype: float64
```

Inference:

- Since the p-value (0.01) is less than the significance level (0.05), we reject the null hypothesis.
- Therefore, the time series is considered non-stationary.
- The test statistic exceeding the critical values further supports the rejection of the null hypothesis.
- The number of lags used in the test is 10.

Conclusion:

- The 'Close' time series is likely non-stationary, indicating the presence of a unit root or a deterministic trend.

Dickey-Fuller test is employed to assess the stationarity of a time series:

Hypotheses:

- Null Hypothesis (H0): The time series has a unit root and is non-stationary.
- Alternative Hypothesis (H1): The time series is stationary.

Test Results:

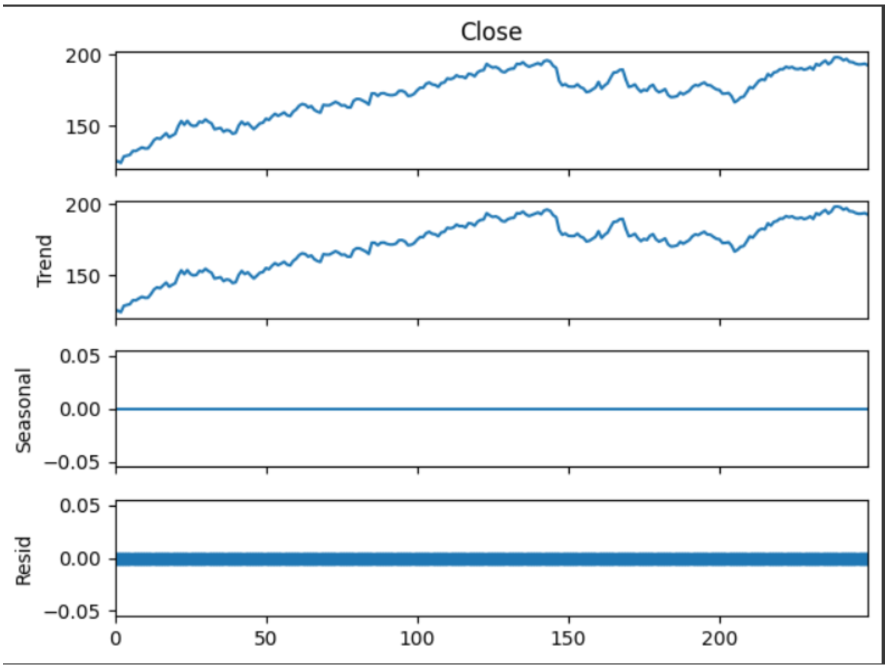
```
Results of Dickey-Fuller Test:
Test Statistic                  -2.586111
p-value                          0.095902
#Lags Used                       0.000000
Number of Observations Used    249.000000
Critical Value (1%)             -3.456888
Critical Value (5%)             -2.873219
Critical Value (10%)            -2.572994
dtype: float64
```

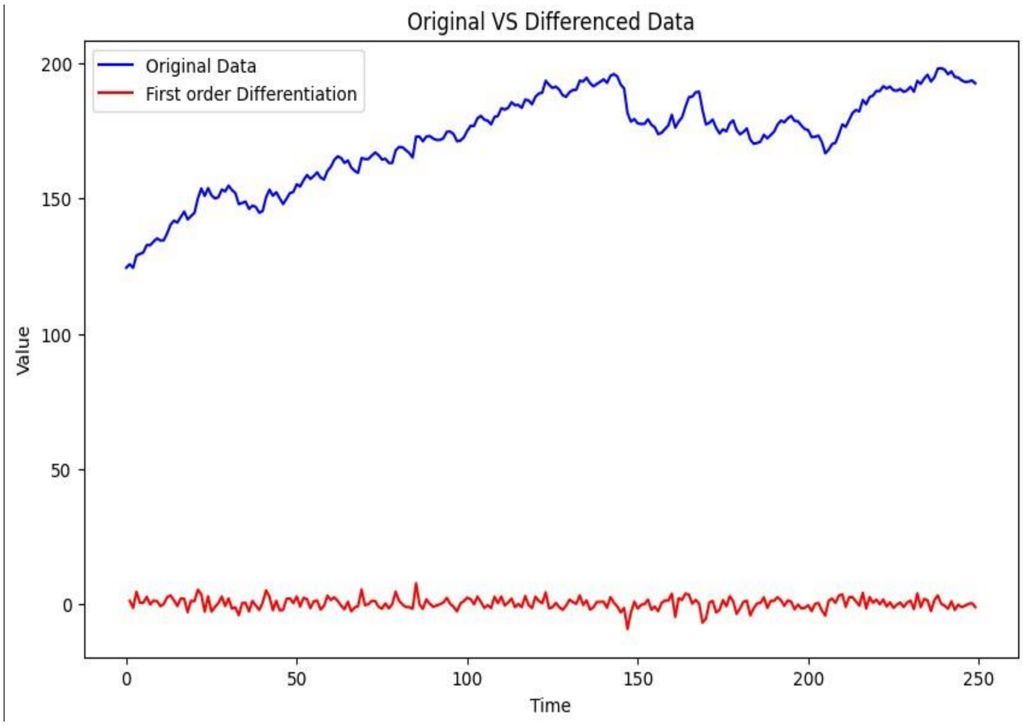
Inference:

- The p-value (0.095902) is greater than the significance level (commonly 0.05).
- We fail to reject the null hypothesis, suggesting insufficient evidence to conclude stationarity.
- The test statistic (-2.586111) is less negative than critical values, further supporting nonrejection.
- Consider additional analysis or differencing to explore stationarity further.

Conclusion:

- The 'Close' time series may be non-stationary based on the Dickey-Fuller test, but further investigation is recommended.





Inference based on Stationarity Tests after First-Order Differencing:

Objective:

- From the two test conducted above, it as resulted that the dataset is non-stationary
- Both KPSS and ADF tests suggest that the dataset becomes stationary after first-order differencing.
- This transformation enhances the suitability of the data for time series analysis.
-

Inference for ACF (AutoCorrelation Function) - Parameter Selection:

Objective:

- ACF is used to analyze the autocorrelation structure in time series data, aiding in parameter selection for time series models.

Interpretation of ACF:

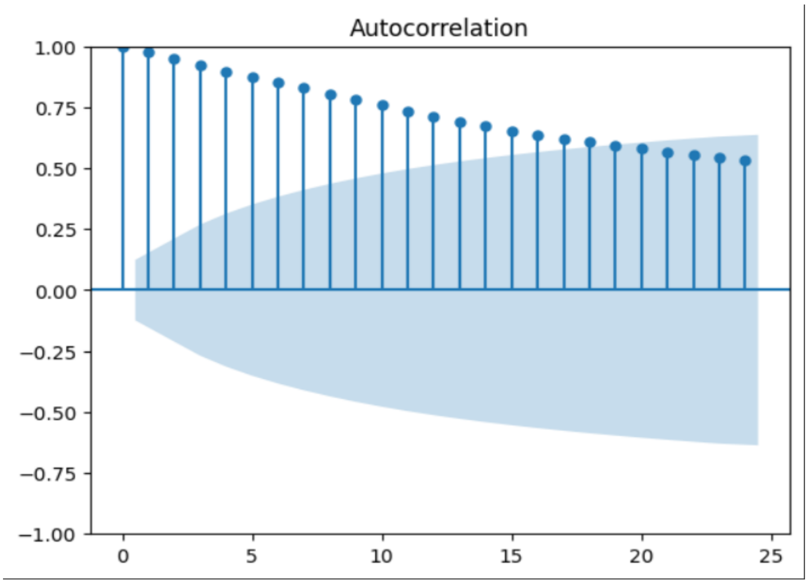
- Peaks and valleys in the ACF plot indicate the strength and direction of autocorrelation at different lags.
- Positive peaks suggest positive autocorrelation and negative peaks suggest negative autocorrelation.
- Lags with values close to 0 indicate weak or no autocorrelation.

Parameter Selection:

- Identify significant lags in the ACF plot based on their heights.
- Significant lags may help determine the order (q) for AutoRegressive (AR) models.

Rules of Thumb:

- Select the lag where the ACF drops to insignificance (within confidence intervals).
- For an AR(q) model, consider lags up to q where the ACF values are significant.



Conclusion:

- From the graph the value of q is 18.

Inference for PACF (Partial Autocorrelation Function):

Purpose:

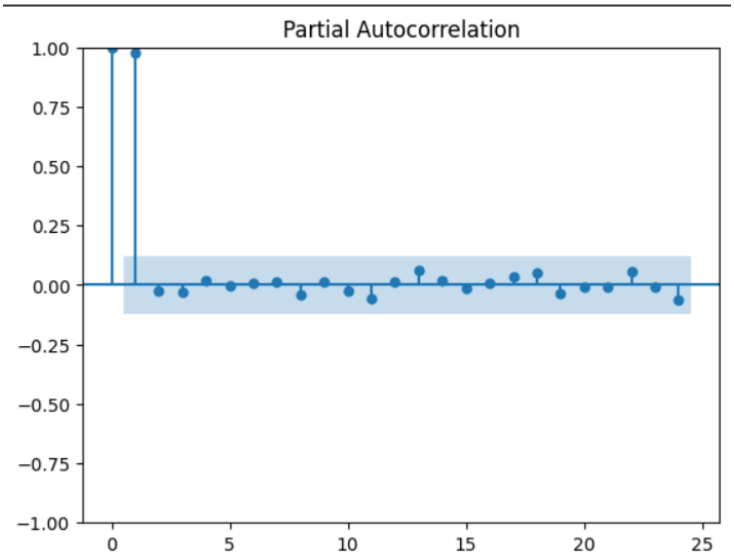
- PACF is used to identify the direct relationship between a time series and its lagged values, removing the influence of intervening observations.

Interpretation:

- Significant positive partial autocorrelation at lag k suggests a direct influence.
- Significant negative partial autocorrelation at lag k indicates a possible over-differencing.

Parameter Selection:

- Lag values with significant spikes guide the choice of the 'p' parameter in an ARIMA model.
- Examine where partial autocorrelation drops off to identify the autoregressive behavior.



Conclusion:

- From the graph value of p is 1.

Arima Model

ARIMA Forecasting Function Inference:

The provided Python function, arima(step, x), performs time series forecasting using the ARIMA (AutoRegressive Integrated Moving Average) model. Here's a breakdown of the key steps:

1. **Data Preparation:**
    - The function takes two parameters: step (the number of days to forecast into the future) and x (a pandas Series containing the time series data).
    - The historical time series data is extracted from the Series x.
2. **Model Training:**
    - The entire dataset is used for training the ARIMA model.
    - The model is initialized with an ARIMA order of (1, 1, 18)(Note: Values are determinded by the test conducted above).
    - The model is fitted to the training data.
3. **Forecasting:**
    - Future values are forecasted for the specified number of days (step).
    - The original stock prices and the forecasted values are combined to create an extended time series.
4. **Date Generation:**
    - Future dates for the forecast are generated based on the last date in the original dataset.
    - Weekday dates are considered, excluding weekends.
5. **Output:**

- The function returns two lists: dates (containing the future dates) and stock_price (containing the historical and forecasted stock prices).

Calculating the No Of Days:

Calling the Function

Stock Portfolio Optimization Inference:

This Python script employs a financial model, represented by the Model class, to optimize stock portfolio values. Here's a breakdown of the key steps:

1. **User Input:**
   - The user is prompted to enter a date in the format (yyyy-mm-dd).
   - The script extracts year, month, and day from the input.
2. **Date Processing:**
   - The last date is calculated based on the user input.
   - The number of weekdays between the last date and the last date of the Close price data available is computed.

3. **ARIMA Forecasting and Portfolio Optimization:**
   - If there are future weekdays for which predictions are needed:
     - The ARIMA model is used to forecast stock prices for upcoming weekdays.
     - The portfolio value is maximized using the forecasted stock prices. ○ Optimal buy indices, dates, and the transition distribution matrix are printed.
     - The script informs whether to buy the stock on the last entered date based on portfolio performance.
   - If there are no upcoming weekdays for which predictions are needed:
     - A message is displayed indicating that stock purchase is not recommended on previous dates.
4. **Output:**
   - The script provides insights into the maximized portfolio value, optimal buy indices, dates, and the transition distribution matrix.
   - The user is informed about the recommendation to buy or not buy the stock on the last entered date.