SIGMA WEDGE HACKATHON

-P. KAMAL NITHISH(21PD17)

Installed Quantrocket and took the 2023 daily close prices of Apple stock (sid='AAPL') from Quantrocket's publicly accessible us-stock price data.

Modelling:

This Python class is designed to model the behavior of a financial market based on daily closing prices.

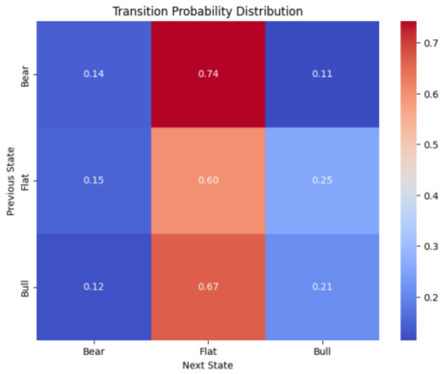
Attributes

- price (DataFrame): Contains financial data with 'Date' and 'Close' columns.
- returns (list): Stores daily returns.
- state (list): Captures market states (1 for Bull, 0 for Flat, -1 for Bear).
- portfolio_values (list): Keeps track of portfolio values over time.
- optimal_buys (list): Stores indices of optimal buying opportunities.

Function Definitions

- State_calculation() :This function calculates daily returns using the formula: . It then classifies each day into market states: Bear (-1), Flat (0), and Bull (1
- portfolio_values():Calculates the portfolio value for each day based on the determined market states. It also identifies optimal buying opportunities and stores their indices.
- visualize_transition_probability(): Generates a visual representation of the transition probability distribution between different market states. This visualization aids in understanding the dynamics of state changes over time.
- decide_buy(): This function determines whether to place a buy order trade based on the change in portfolio value. It evaluates the portfolio value change and decides whether initiating a trade on certain dates is advisable.

Output:



PORTFOLIO VALUE V(N): 17

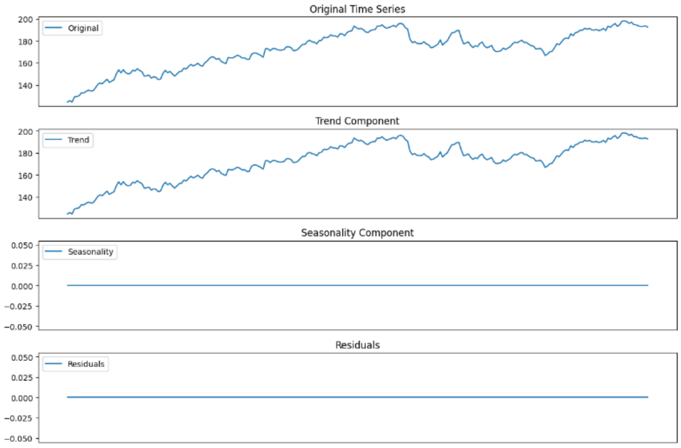
OPTIMAL BUY INDICES: [6, 8, 12, 16, 21, 28, 30, 41, 50, 52, 59, 61, 69, 79, 85, 88, 94, 100, 103, 108, 110, 113, 117, 120, 123, 133, 142, 160, 164, 177, 187, 191, 207, 209, 212, 216, 218, 232, 234, 238]

BUY DATES : ['2023-01-11', '2023-01-13', '2023-01-20', '2023-01-26', '2023-02-02', '2023-02-13', '2023-02-15', '2023-03-03', '2023-03-16', '2023-03-20', '2023-03-29', '2023-03-31', '2023-04-13', '2023-04-27', '2023-05-05', '2023-05-10', '2023-05-18', '2023-05-26', '2023-06-01', '2023-06-08', '2023-06-12', '2023-06-15', '2023-06-22', '2023-06-27', '2023-06-30', '2023-07-17', '2023-07-28', '2023-08-23', '2023-08-29', '2023-09-18', '2023-10-02', '2023-10-06', '2023-10-30', '2023-11-01', '2023-11-06', '2023-11-10', '2023-11-14', '2023-12-05', '2023-12-07', '2023-12-13']

Time Series Forecasting:

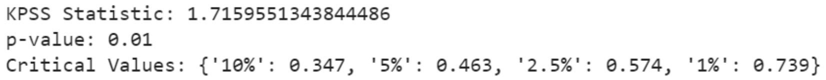
The ARIMA (Auto Regressive Integrated Moving Average) model is used for time series forecasting, specifically for predicting future values based on historical observations. It combines autoregressive (AR) and moving average (MA) components, allowing it to capture trends, seasonality, and temporal dependencies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

- **Components**: **Result**: Increasing Trend, No Seasonality, No Residuals.

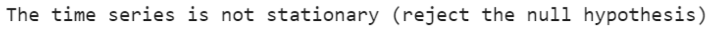


- **Testing Stationarity**

KPSS Test:

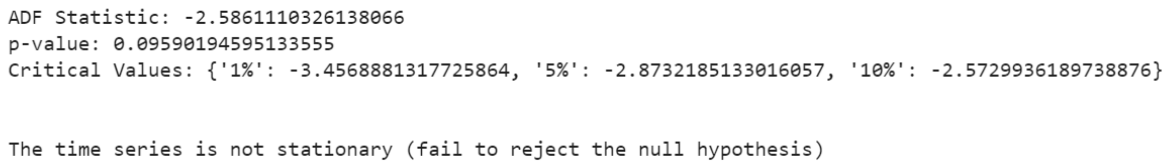


```
KPSS Statistic: 1.7159551343844486
p-value: 0.01
Critical Values: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}
```



```
The time series is not stationary (reject the null hypothesis)
```

Dickey Fuller Test:



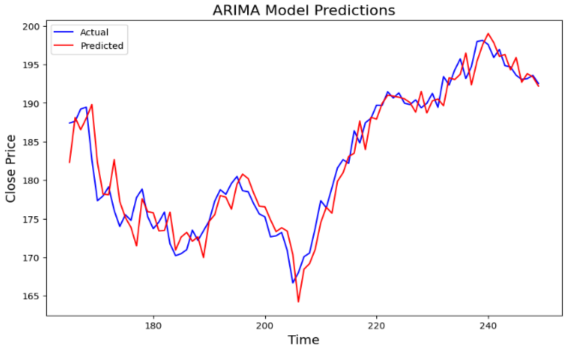
```
ADF Statistic: -2.5861110326138066
p-value: 0.09590194595133555
Critical Values: {'1%': -3.4568881317725864, '5%': -2.8732185133016057, '10%': -2.5729936189738876}


The time series is not stationary (fail to reject the null hypothesis)
```

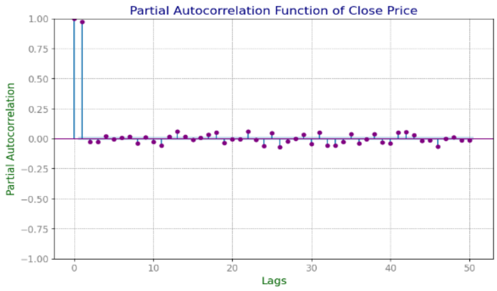
Conclusion:

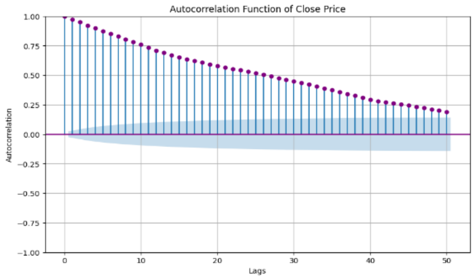
The 'Close' time series is likely non-stationary, indicating the presence of a unit root or a deterministic trend.

ARIMA:

- **Differenced Data:** To make non stationary as stationary data. Stationarity (constant mean and variance) achieved after first order differencing used KPSS test for testing.
- **ACF Plot:** To calculate q (Moving average) parameter for model.
- **PACF Plot:** To calculate p (Auto Regression) parameter for the model.







Forecasting:

Input:Specify a date for prediction.

Method:

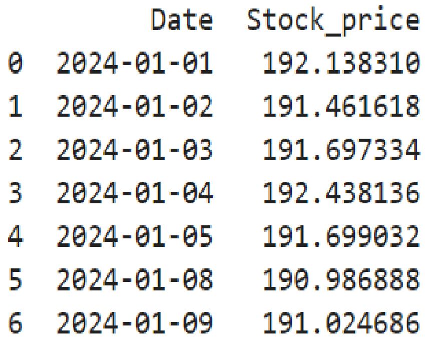
1. Determine the time gap by calculating the number of days between the last date in the dataset and the specified date.
2. Utilize this temporal information as steps for forecasting, employing the ARIMA model.

3. The Modelling model incorporates the forecasted dataframe to assess market conditions, executing essential computations.

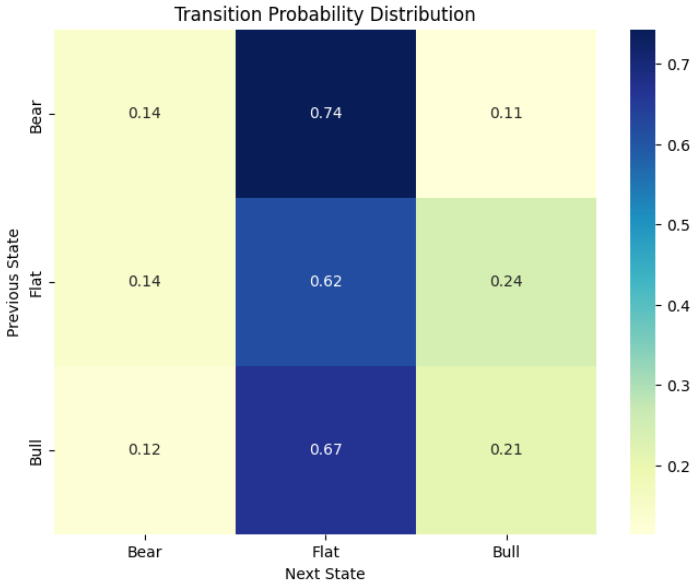
Output:Transition probability distribution, Optimal Buy Indices, Buy Dates.

Additionally, the model predicts the feasibility of executing a stock trade on the given day

Result:



```
         Date  Stock_price
0  2024-01-01   192.138310
1  2024-01-02   191.461618
2  2024-01-03   191.697334
3  2024-01-04   192.438136
4  2024-01-05   191.699032
5  2024-01-08   190.986888
6  2024-01-09   191.024686
```



PORTFOLIO VALUE V(N): 17

OPTIMAL BUY INDICES: [6, 8, 12, 16, 21, 28, 30, 41, 50, 52, 59, 61, 69, 79, 85, 88, 94, 100, 103, 108, 110, 113, 117, 120, 123, 133, 142, 160, 164, 177, 187, 191, 207, 209, 212, 216, 218, 232, 234, 238]

BUY DATES : ['2023-01-11', '2023-01-13', '2023-01-20', '2023-01-26', '2023-02-02', '2023-02-13', '2023-02-15', '2023-03-03', '2023-03-16', '2023-03-20', '2023-03-29', '2023-03-31', '2023-04-13', '2023-04-27', '2023-05-05', '2023-05-10', '2023-05-18', '2023-05-26', '2023-06-01', '2023-06-08', '2023-06-12', '2023-06-15', '2023-06-22', '2023-06-27', '2023-06-30', '2023-07-17', '2023-07-28', '2023-08-23', '2023-08-29',

```
'2023-09-18', '2023-10-02', '2023-10-06', '2023-10-30', '2023-11-01', '2023-11-06', '2023-11-10', '2023-11-14', '2023-12-05', '2023-12-07', '2023-12-13']
You can't buy the stock on 2024-01-10.

***********************************************************************
```