

## CHAPTER 5

### IMPLEMENTATION AND RESULTS

#### 5.1. Implementation

This research uses the python programming language. The stock price data that has been collected will later be used by Prophet and WMA methods to predict future stock prices. The steps taken in implementing Prophet and WMA methods will be explained below.

#### 5.2. Prophet Implementation

```
1 df['y_scaled'] = df['y'] / df['y'].max()
2 t_scale = df['ds'].max() - df['ds'].min()
3 df['t'] = (df['ds'] - df['ds'].min()) / t_scale
4 future['t'] = (future['ds'] - future['ds'].min()) / t_scale
```

Line 5 is used to get the y\_scaled value by using the max absolute scaler method. The value of y\_scaled is the value of the stock price that has been scaled. Then in lines 6 to 7 is used to scaled the time data with the min max scaler. Line 8 is used for is scaling the future time data, which is the predicted time period with the min max scaler.

```
5 stop = len(t)
6 start = len(t) - check
7 changepoints = np.linspace(start, stop, 25)
```

Line 9 is used to be the stop requirements, namely the condition of the stop point for the changepoint installation. Line 10 is used to be the start requirements, namely the initial location conditions for the changepoint installation. The purpose of this design is to be able to install the changepoint correctly. Line 11 is used to install changepoints with the same distance from each other according to the designed start and stop points as many as 25.

```

8  deltas = np.random.laplace(0, 0.01, 25)
9  beta = np.random.normal(0, 10)
10 a = (t[:, None] > changepoints) * 1

```

Line 12 is used to make arrangements about how much the trend will change. Line 13 is used to make arrangements about how much seasonality will change. Line 14 is used to make a matrix. This matrix will be useful for recording the point of change in the value of growth due to passing through the changepoint. Later this matrix will be used in calculating the trend value.

```

11 trend = k + a @ deltas
12 trend = trend * t
13 trend = trend + m + a @ gamma
14 trend = trend * df['y'].max()

```

Line 15 to 18 is used to calculate the trend value based on the Prophet method. Line 15, is used to perform matrix multiplication, which is between a and deltas. Then the results will be added to the value of k. The goal is that every time a changepoint is passed, the value of k will be added to the value of deltas. Line 16 is used to multiplied the matrix between the total growth and the value of t. Line 17 is used to add the result of multiplying the total growth matrix and the t value with the total offset value. The total offset value is obtained by multiplying the matrix between a and gamma. Then the result will be added with the value of m. The goal is that when every changepoint is passed, the value of m will be added to the value of gamma. In line 18, the results of the previous calculation will be multiplied by the maximum share price. This is done to restore the scaled stock price data to normal.

```

15 if check == 7 or check <= 30:
16     n = 3
17     p = 7
18 elif check >= 365:
19     n = 10

```

```

20 p = 365.25
21 s = 2 * np.pi * n * t / p
22 seasonality = np.cos(s) + np.sin(s)
23 seasonality = seasonality * beta
24 df2 = np.array(1)
25 df2 = trend + seasonality

```

Line 19 to 27 is used to calculate the seasonality value using the Prophet method. On line 19 to 24 there will be a check for the amount of time that will be predicted. If the number is 7 days or less than 30 days then n will be worth 3 and p will be worth 7. But if the amount of time to be predicted is more or equal to 365 days then n will be worth 10 and p will be worth 365.25. On line 25-26 there will be a process of calculating the seasonality formula. In line 27 the result of calculating the seasonality formula will be multiplied by beta. Line 28 is used to create a new array to store the predicted results. In line 29, the trend values and seasonality values will be added to get the prediction results.

### 5.3. WMA Implementation

```

26 alldata= df['y']
27 weightamount = 30
28 weight = np.array((15))
29 nextweight = np.sum(weight)

```

Line 30 is used to enter all stock price data into the array. Later alldata will be used to store the results of stock price predictions. Line 31 is the place to set the amount of data that will be used to train the WMA method. Line 32 is used to enter the first weight of 15. Line 33 is used to enter the number of weights into the next weight.

```

30 for i in check:
31     wma = data * weight
32     wma = wma / totalweight
33     wma = np.sum(wma)
34     alldata = np.append(alldata,wma)

```

In line 34-37 there is a process of calculating the prediction results using the WMA method. On line 35, there is a multiplication process between stock price data and weight. The latest stock data will be multiplied by the largest weight. In line 36, the result of multiplying stock price data with weight will be divided by the total of all weight. In line 37 all the data in the wma array will be added up to get the prediction result. In line 38 the prediction results will be stored in the alldata array.

**5.4. Result**

**Table 5.1: Prediction Result**

Method	Validation	Test
Prophet	2.73	6.92
Weighted Moving Average	10.47	14.87

The table above is the accuracy value of the WMA and Prophet methods calculated by the MAPE method. Testing is carried out in 2 stages, namely validation and test. After being calculated using the MAPE method, it was found that, Prophet prediction accuracy value in the validation stage was 2.73 and in the test stage was 6.92. And WMA prediction accuracy value in the validation stage was 10.47 and in the test stage was 14.87.