Import library

In [1]:

```
from pynse import *
import datetime
import logging
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from itertools import islice
import datetime
from dateutil.relativedelta import relativedelta

# config logging

logging.basicConfig(level=logging.DEBUG)
```

executed in 2.65s, finished 12:56:40 2021-09-06

In [2]:

```
Selec_No_of_years_to_show_data= 1
```

executed in 6ms, finished 12:56:40 2021-09-06

In [3]:

```
nse = Nse()
```

executed in 39ms, finished 12:56:43 2021-09-06

```
INFO:pynse.core:pyNse cache size: (6.07, 'MB').
You may want to run `nse.clear_data()` if running low on disk space.
```

Return Market status

In [4]:

```
nse.market_status()
```

executed in 928ms, finished 12:56:48 2021-09-06

```
INFO:pynse.core:downloading market status
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.nseindia.co
m:443
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET / HTTP/1.1" 200
41318
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET /api/marketStatu
s HTTP/1.1" 200 226
```

Out[4]:

```
{'marketState': [{'market': 'Capital Market',
   'marketStatus': 'Open',
   'tradeDate': '06-Sep-2021',
   'index': 'NIFTY 50',
   'last': 17377.4,
   'variation': 53.80000000000291,
   'percentChange': 0.31,
   'marketStatusMessage': 'Normal Market is Open'},
  {'market': 'Currency',
   'marketStatus': 'Open',
   'tradeDate': '06-Sep-2021',
   'index': '',
   'last': '',
   'variation': '',
   'percentChange': '',
   'marketStatusMessage': 'Market is Open'},
  {'market': 'Commodity',
   'marketStatus': 'Open',
   'tradeDate': '06-Sep-2021',
   'index': '',
   'last': '',
   'variation': '',
   'percentChange': '',
   'marketStatusMessage': 'Market is Open'},
  {'market': 'Debt',
   'marketStatus': 'Open',
   'tradeDate': '06-Sep-2021',
   'index': '',
   'last': '',
   'variation': '',
   'percentChange': '',
   'marketStatusMessage': 'Market is Open'}]}
```

Bhavcopy

The Bhav Copy is a snapshot of the activity that has taken place in the market for the particular day. ... You can use it to observe activity taking place in any particular scrip or across the broad market, as it contains End Of Day (EOD) details of the entire market.

In [5]:

```
nse.bhavcopy()
```

executed in 1.08s, finished 12:56:53 2021-09-06

```
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.nseindia.co
m:443
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET / HTTP/1.1" 200
41317
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET /api/historical/
cm/equity?symbol=SBIN&series=%5B%22EQ%22%5D&from=27-08-2021&to=06-09-2021&csv=t
rue HTTP/1.1" 200 437
DEBUG:pynse.core:read C:\Users\kumar\.pynse/bhavcopy_eq/bhav_2021-09-03.pkl fro
m disk
```

Out[5]:

		DATE1	PREV_CLOSE	OPEN_PRICE	HIGH_PRICE	LOW_PRICE
SYMBOL	**SERIES**					
20MICRONS	**EQ**	2021-09-03	53.85	54.45	54.85	53.1
21STCENMGM	**EQ**	2021-09-03	34.85	35.50	35.50	35.5
3MINDIA	**EQ**	2021-09-03	24543.05	24900.00	25000.00	24400.0
5PAISA	**EQ**	2021-09-03	466.15	463.00	475.10	463.0
63MOONS	**EQ**	2021-09-03	94.80	93.75	99.45	93.7
...	**...**	...	...	...	...	.
ZODIACLOTH	**EQ**	2021-09-03	113.05	115.70	118.90	112.2
ZODJRDMKJ	**EQ**	2021-09-03	30.95	31.65	31.70	30.4
ZOMATO	**EQ**	2021-09-03	137.60	135.95	151.40	135.6
ZOTA	**EQ**	2021-09-03	381.85	386.75	388.75	371.5
ZYDUSWELL	**EQ**	2021-09-03	2319.40	2320.00	2325.00	2287.1

1519 rows × 13 columns

Sort data by average price

In [6]:

```
daily_share_price =nse.bhavcopy()
filt = daily_share_price.sort_values(by= ['AVG_PRICE'],ascending=False).head(5

filt
top_5_share_by_average_price= filt
top_5_share_by_average_price
```

executed in 666ms, finished 12:56:59 2021-09-06

```
DEBUG:pynse.core:read C:\Users\kumar\.pynse/bhavcopy_eq/bhav_2021-09-03.pkl f
rom disk
```

Out[6]:

		DATE1	PREV_CLOSE	OPEN_PRICE	HIGH_PRICE	LOW_PRICE
SYMBOL	SERIES					
MRF	EQ	2021-09-03	81265.15	81949.0	82600.10	81509.
HONAUT	EQ	2021-09-03	40517.35	40599.9	40603.05	40084.
PAGEIND	EQ	2021-09-03	32049.55	32032.0	32949.85	32032.
SHREECEM	EQ	2021-09-03	30323.25	30290.1	30590.80	29962.

Here Symbol and series are index, to proced further, First reset indexes.

In [7]:

```
top_5_share_by_average_price.index
```

executed in 9ms, finished 12:57:02 2021-09-06

Out[7]:

```
MultiIndex([(     'MRF', 'EQ'),
            (  'HONAUT', 'EQ'),
            ( 'PAGEIND', 'EQ'),
            ('SHREECEM', 'EQ'),
            ( '3MINDIA', 'EQ')],
           names=['SYMBOL', 'SERIES'])
```

Reset Index

In [8]:

```
top_5_share_by_average_price= top_5_share_by_average_price.reset_index()
top_5_share_by_average_price
```

executed in 31ms, finished 12:57:06 2021-09-06

Out[8]:

	SYMBOL	SERIES	DATE1	PREV_CLOSE	OPEN_PRICE	HIGH_PRICE	LOW_PRICE
0	MRF	EQ	2021-09-03	81265.15	81949.0	82600.10	81509
1	HONAUT	EQ	2021-09-03	40517.35	40599.9	40603.05	40084
2	PAGEIND	EQ	2021-09-03	32049.55	32032.0	32949.85	32032
3	SHREECEM	EQ	2021-09-03	30323.25	30290.1	30590.80	29962
4	3MINDIA	EQ	2021-09-03	24543.05	24900.0	25000.00	24400

Store top 5 share in list

In [9]:

```
list_of_top_5_share= []
list_of_top_5_share= top_5_share_by_average_price['SYMBOL']
print(list_of_top_5_share)
```

executed in 15ms, finished 12:57:09 2021-09-06

```
0         MRF
1      HONAUT
2     PAGEIND
3    SHREECEM
4     3MINDIA
Name: SYMBOL, dtype: object
```

Create pandas data_frame

In [10]:

```
pd_data_frame = pd.DataFrame(top_5_share_by_average_price)
```

executed in 5ms, finished 12:57:12 2021-09-06

Set time duration to fetch data

In [11]:
```
datetime.date.today()
```
executed in 6ms, finished 12:57:15 2021-09-06

Out[11]:
```
datetime.date(2021, 9, 6)
```

Type *Markdown* and LaTeX: α^2

In [12]:
```
years_ago = datetime.date.today() - relativedelta(years=Selec_No_of_years_to_s

years_ago
```
executed in 6ms, finished 12:57:17 2021-09-06

Out[12]:
```
datetime.date(2020, 9, 6)
```

Testing of getting data from NSE

In [13]:
```
hdfc= nse.get_hist('HDFC', years_ago,datetime.date.today())
hdfc
```
executed in 1.07s, finished 12:57:20 2021-09-06

```
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.nseindia.
com:443
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET / HTTP/1.1" 20
0 41240
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET /api/historica
l/cm/equity?symbol=HDFC&series=%5B%22EQ%22%5D&from=06-09-2020&to=06-09-2021&c
sv=true HTTP/1.1" 200 11094
```

Objective is to pridict MACD , lowEMA, SIGNAL_Line,and longLine of TOP 5 Share by Average Price

In [14]:

```
list_of_top_5_share
```

executed in 16ms, finished 12:57:30 2021-09-06

Out[14]:

```
0          MRF
1       HONAUT
2      PAGEIND
3     SHREECEM
4      3MINDIA
Name: SYMBOL, dtype: object
```

In [16]:

```
list_of_data= []
for i in list_of_top_5_share_share:
    list_of_data.append(nse.get_hist(i, years_ago,datetime.date.today()))
list_of_data
```

executed in 4.50s, finished 12:57:48 2021-09-06

```
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.nseindia.
com:443
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET / HTTP/1.1" 20
0 41199
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET /api/historica
l/cm/equity?symbol=MRF&series=%5B%22EQ%22%5D&from=06-09-2020&to=06-09-2021&cs
v=true HTTP/1.1" 200 11458
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.nseindia.
com:443
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET / HTTP/1.1" 20
0 41285
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET /api/historica
l/cm/equity?symbol=HONAUT&series=%5B%22EQ%22%5D&from=06-09-2020&to=06-09-2021
&csv=true HTTP/1.1" 200 10963
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.nseindia.
com:443
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET / HTTP/1.1" 20
0 41287
DEBUG:urllib3.connectionpool:https://www.nseindia.com:443 "GET /api/historica
l/cm/equity?symbol=PAGEIND&series=%5B%22EQ%22%5D&from=06-09-2020&to=06-09-202
```

Display data stored in list

In [17]:
```
list_of_data
```

executed in 38ms, finished 12:57:55 2021-09-06

Out[17]:
```
[               open      high       low     close  volume
 Date
 2020-09-07  58425.4  58854.95  58000.00  58374.20   12609
 2020-09-08  58500.0  58500.00  58050.00  58212.65    9805
 2020-09-09  58200.0  59054.95  57867.00  58763.55   12863
 2020-09-10  58865.0  59399.00  58700.00  59134.05    9822
 2020-09-11  59100.0  59400.00  58445.90  59226.70   14081
 ...             ...       ...       ...       ...     ...
 2021-08-30  77000.0  78374.95  77000.00  78185.50    9045
 2021-08-31  78300.0  79855.05  78257.80  79742.75   14859
 2021-09-01  80000.0  80559.95  79239.80  79954.65   10789
 2021-09-02  80393.0  81875.55  80041.45  81265.15   14947
 2021-09-03  81949.0  82600.10  81509.90  82397.85   15764

 [248 rows x 5 columns],
                open      high       low     close  volume
 Date
 2020-09-07  32250.0  32699.00  32150.00  32238.20    1048
 2020-09-08  32238.2  32392.00  31150.00  31321.10    2538
 2020-09-09  31280.0  32500.00  30802.00  31303.15    2969
 2020-09-10  31399.0  32666.20  31328.55  32136.85   11437
 2020-09-11  32279.0  32800.00  31705.00  32625.15    3441
 ...             ...       ...       ...       ...     ...
 2021-08-30  40139.0  40350.00  39306.70  39636.30    4555
 2021-08-31  39636.3  40500.00  39501.10  39604.90    5046
 2021-09-01  39899.0  40888.00  39250.00  40299.10    5721
 2021-09-02  40569.0  40690.00  40160.00  40517.35    3795
 2021-09-03  40599.9  40603.05  40084.05  40161.60    3233

 [248 rows x 5 columns],
                open      high       low     close  volume
 Date
 2020-09-07  18550.0  18625.00  17900.00  18305.65   89494
 2020-09-08  18299.9  18575.00  18050.00  18137.45   61865
 2020-09-09  18100.0  18589.95  17905.10  18503.85   67414
 2020-09-10  18511.2  18743.35  18301.50  18495.10   73491
 2020-09-11  18478.0  18777.00  18384.25  18440.75   45656
 ...             ...       ...       ...       ...     ...
 2021-08-30  30950.0  31490.00  30899.95  31424.20   21907
 2021-08-31  31424.2  31629.00  31100.00  31499.10   23477
 2021-09-01  31500.0  32288.95  31279.30  32108.25   21226
 2021-09-02  32116.5  32391.90  31721.00  32049.55   23348
 2021-09-03  32032.0  32949.85  32032.00  32473.30   18937

 [248 rows x 5 columns],
                 open      high       low     close  volume
 Date
 2020-09-07  20100.00  20119.15  19636.05  19782.95   59729
 2020-09-08  19782.95  20199.00  19272.10  19355.85   65858
 2020-09-09  19300.00  19566.00  19068.05  19401.50   83567
 2020-09-10  19500.00  19750.00  19319.00  19643.35   44713
 2020-09-11  19735.50  19735.50  19365.10  19502.50   61264
 ...              ...       ...       ...       ...     ...
2021-08-30  26900.10  27420.00  26830.80  27157.85   41390
2021-08-31  27150.00  28469.00  27048.75  28287.90  102061
2021-09-01  28296.00  28600.00  27948.10  28478.25   75809
2021-09-02  28550.00  30514.85  28480.00  30323.25  157100
2021-09-03  30290.10  30590.80  29962.10  30440.80   81600

[248 rows x 5 columns],
                open      high       low     close  volume
Date
2020-09-07  19700.00  19739.65  19301.45  19353.15    2449
2020-09-08  19400.00  19612.00  19100.00  19130.70    2847
2020-09-09  19145.00  19299.00  18532.10  18807.35    2475
2020-09-10  18851.00  19241.00  18700.00  18726.10    3035
2020-09-11  18999.90  18999.90  18350.00  18569.00    7016
...              ...       ...       ...       ...     ...
2021-08-30  22688.15  23062.45  22618.00  22979.10   21577
2021-08-31  23100.00  24838.30  23100.00  24391.45   16209
2021-09-01  24391.45  25304.00  24164.10  25060.10    7005
2021-09-02  25060.10  25368.00  24400.00  24543.05    4121
2021-09-03  24900.00  25000.00  24400.00  24537.05    4348

[248 rows x 5 columns]]
```

display data of first share, data fetched by index of list

In [18]:

```
list_of_data[0]
```

executed in 29ms, finished 12:57:59 2021-09-06

2020-09-07	58425.4	58854.95	58000.00	58374.20	12809
2020-09-08	58500.0	58500.00	58050.00	58212.65	9805
2020-09-09	58200.0	59054.95	57867.00	58763.55	12863
2020-09-10	58865.0	59399.00	58700.00	59134.05	9822
2020-09-11	59100.0	59400.00	58445.90	59226.70	14081
...	...	...	...	...	...
2021-08-30	77000.0	78374.95	77000.00	78185.50	9045
2021-08-31	78300.0	79855.05	78257.80	79742.75	14859
2021-09-01	80000.0	80559.95	79239.80	79954.65	10789
2021-09-02	80393.0	81875.55	80041.45	81265.15	14947
2021-09-03	81949.0	82600.10	81509.90	82397.85	15764

248 rows × 5 columns

Dump data in pandas dataFrame from list

In [19]:

```
list_of_data_frame_of_share_market =[]

for i in list_of_data:
    list_of_data_frame_of_share_market.append(pd.DataFrame(i))
list_of_data_frame_of_share_market
```

executed in 43ms, finished 12:58:02 2021-09-06

Out[19]:

```
[                open      high       low     close  volume
 Date
 2020-09-07  58425.4  58854.95  58000.00  58374.20   12609
 2020-09-08  58500.0  58500.00  58050.00  58212.65    9805
 2020-09-09  58200.0  59054.95  57867.00  58763.55   12863
 2020-09-10  58865.0  59399.00  58700.00  59134.05    9822
 2020-09-11  59100.0  59400.00  58445.90  59226.70   14081
 ...             ...       ...       ...       ...     ...
 2021-08-30  77000.0  78374.95  77000.00  78185.50    9045
 2021-08-31  78300.0  79855.05  78257.80  79742.75   14859
 2021-09-01  80000.0  80559.95  79239.80  79954.65   10789
 2021-09-02  80393.0  81875.55  80041.45  81265.15   14947
 2021-09-03  81949.0  82600.10  81509.90  82397.85   15764

 [248 rows x 5 columns],
                open      high       low     close  volume
 Date
 2020-09-07  32250.0  32699.00  32150.00  32238.20    1048
 2020-09-08  32238.2  32392.00  31150.00  31321.10    2538
 2020-09-09  31280.0  32500.00  30802.00  31303.15    2969
 2020-09-10  31399.0  32666.20  31328.55  32136.85   11437
 2020-09-11  32279.0  32800.00  31705.00  32625.15    3441
 ...             ...       ...       ...       ...     ...
 2021-08-30  40139.0  40350.00  39306.70  39636.30    4555
 2021-08-31  39636.3  40500.00  39501.10  39604.90    5046
 2021-09-01  39899.0  40888.00  39250.00  40299.10    5721
 2021-09-02  40569.0  40690.00  40160.00  40517.35    3795
 2021-09-03  40599.9  40603.05  40084.05  40161.60    3233

 [248 rows x 5 columns],
                open      high       low     close  volume
 Date
 2020-09-07  18550.0  18625.00  17900.00  18305.65   89494
 2020-09-08  18299.9  18575.00  18050.00  18137.45   61865
 2020-09-09  18100.0  18589.95  17905.10  18503.85   67414
 2020-09-10  18511.2  18743.35  18301.50  18495.10   73491
 2020-09-11  18478.0  18777.00  18384.25  18440.75   45656
 ...             ...       ...       ...       ...     ...
 2021-08-30  30950.0  31490.00  30899.95  31424.20   21907
 2021-08-31  31424.2  31629.00  31100.00  31499.10   23477
 2021-09-01  31500.0  32288.95  31279.30  32108.25   21226
 2021-09-02  32116.5  32391.90  31721.00  32049.55   23348
 2021-09-03  32032.0  32949.85  32032.00  32473.30   18937

 [248 rows x 5 columns],
                 open      high       low     close  volume
 Date
 2020-09-07  20100.00  20119.15  19636.05  19782.95   59729
 2020-09-08  19782.95  20199.00  19272.10  19355.85   65858
2020-09-09  19300.00  19566.00  19068.05  19401.50   83567
2020-09-10  19500.00  19750.00  19319.00  19643.35   44713
2020-09-11  19735.50  19735.50  19365.10  19502.50   61264
...              ...       ...       ...       ...     ...
2021-08-30  26900.10  27420.00  26830.80  27157.85   41390
2021-08-31  27150.00  28469.00  27048.75  28287.90  102061
2021-09-01  28296.00  28600.00  27948.10  28478.25   75809
2021-09-02  28550.00  30514.85  28480.00  30323.25  157100
2021-09-03  30290.10  30590.80  29962.10  30440.80   81600

[248 rows x 5 columns],
                open      high       low     close  volume
Date
2020-09-07  19700.00  19739.65  19301.45  19353.15    2449
2020-09-08  19400.00  19612.00  19100.00  19130.70    2847
2020-09-09  19145.00  19299.00  18532.10  18807.35    2475
2020-09-10  18851.00  19241.00  18700.00  18726.10    3035
2020-09-11  18999.90  18999.90  18350.00  18569.00    7016
...              ...       ...       ...       ...     ...
2021-08-30  22688.15  23062.45  22618.00  22979.10   21577
2021-08-31  23100.00  24838.30  23100.00  24391.45   16209
2021-09-01  24391.45  25304.00  24164.10  25060.10    7005
2021-09-02  25060.10  25368.00  24400.00  24543.05    4121
2021-09-03  24900.00  25000.00  24400.00  24537.05    4348

[248 rows x 5 columns]]
```

In [20]:

```
list_of_data_frame_of_share_market[0]
```

executed in 31ms, finished 12:58:07 2021-09-06

Out[20]:

	open	high	low	close	volume
Date					
2020-09-07	58425.4	58854.95	58000.00	58374.20	12609
2020-09-08	58500.0	58500.00	58050.00	58212.65	9805
2020-09-09	58200.0	59054.95	57867.00	58763.55	12863
2020-09-10	58865.0	59399.00	58700.00	59134.05	9822
2020-09-11	59100.0	59400.00	58445.90	59226.70	14081
...	...	...	...	...	...
2021-08-30	77000.0	78374.95	77000.00	78185.50	9045
2021-08-31	78300.0	79855.05	78257.80	79742.75	14859
2021-09-01	80000.0	80559.95	79239.80	79954.65	10789
2021-09-02	80393.0	81875.55	80041.45	81265.15	14947
2021-09-03	81949.0	82600.10	81509.90	82397.85	15764

248 rows × 5 columns

longEMA= df.Close.ewm(span=21, adjust=False).mean() longEMA

Calculate 3 moving averages according to closig market

In [21]:

```
list_of_data_frame_of_share_market[0]
```

executed in 31ms, finished 12:58:13 2021-09-06

Out[21]:

	open	high	low	close	volume
Date					
2020-09-07	58425.4	58854.95	58000.00	58374.20	12609
2020-09-08	58500.0	58500.00	58050.00	58212.65	9805
2020-09-09	58200.0	59054.95	57867.00	58763.55	12863
2020-09-10	58865.0	59399.00	58700.00	59134.05	9822
2020-09-11	59100.0	59400.00	58445.90	59226.70	14081
...	...	...	...	...	...
2021-08-30	77000.0	78374.95	77000.00	78185.50	9045
2021-08-31	78300.0	79855.05	78257.80	79742.75	14859
2021-09-01	80000.0	80559.95	79239.80	79954.65	10789
2021-09-02	80393.0	81875.55	80041.45	81265.15	14947
2021-09-03	81949.0	82600.10	81509.90	82397.85	15764

248 rows × 5 columns

Find values short Signam and Long 'MACD' Lines.

In [22]:

```
list_of_EMA =[]


# signalEMA= list_of_data_frame_of_share_market[0]['close'].ewm(span=12, adjus
# longEMA= list_of_data_frame_of_share_market[0]['close'].ewm(span=26, adjust=

# for i in range(len(list_of_data_frame_of_share_market)):
#     print(len(list_of_data_frame_of_share_market))
#     print(type(i))
#     print(ord('i'))
#     j= ord('i')- 105

shortEMA= list_of_data_frame_of_share_market[0]['close'].ewm(span=9, adjust=Fa
signalEMA= list_of_data_frame_of_share_market[0]['close'].ewm(span=12, adjust=
longEMA= list_of_data_frame_of_share_market[0]['close'].ewm(span=26, adjust=Fa


shortEMA_1= list_of_data_frame_of_share_market[1]['close'].ewm(span=9, adjust=
signalEMA_1= list_of_data_frame_of_share_market[1]['close'].ewm(span=12, adjus
longEMA_1= list_of_data_frame_of_share_market[1]['close'].ewm(span=26, adjust=


shortEMA_2= list_of_data_frame_of_share_market[2]['close'].ewm(span=9, adjust=
signalEMA_2= list_of_data_frame_of_share_market[2]['close'].ewm(span=12, adjus
longEMA_2= list_of_data_frame_of_share_market[2]['close'].ewm(span=26, adjust=

shortEMA_3= list_of_data_frame_of_share_market[3]['close'].ewm(span=9, adjust=
signalEMA_3= list_of_data_frame_of_share_market[3]['close'].ewm(span=12, adjus
longEMA_3= list_of_data_frame_of_share_market[3]['close'].ewm(span=26, adjust=

short_EMA_4= list_of_data_frame_of_share_market[4]['close'].ewm(span=9, adjust
signalEMA_4= list_of_data_frame_of_share_market[4]['close'].ewm(span=12, adjus
longEMA_4= list_of_data_frame_of_share_market[4]['close'].ewm(span=26, adjust=

# shortEMA_5= list_of_data_frame_of_share_market[5]['close'].ewm(span=9, adjus
# signalEMA_5= list_of_data_frame_of_share_market[5]['close'].ewm(span=12, adj
# longEMA_5= list_of_data_frame_of_share_market[5]['close'].ewm(span=26, adjus

# shortEMA_6= list_of_data_frame_of_share_market[6]['close'].ewm(span=9, adjus
# signalEMA_6= list_of_data_frame_of_share_market[6]['close'].ewm(span=12, adj
# longEMA_6= list_of_data_frame_of_share_market[6]['close'].ewm(span=26, adjus

# shortEMA_7= list_of_data_frame_of_share_market[7]['close'].ewm(span=9, adjus
# signalEMA_7= list_of_data_frame_of_share_market[7]['close'].ewm(span=12, adj
# longEMA_7= list_of_data_frame_of_share_market[7]['close'].ewm(span=26, adjus

# shortEMA_8= list_of_data_frame_of_share_market[8]['close'].ewm(span=9, adjus
# signalEMA_8= list_of_data_frame_of_share_market[8]['close'].ewm(span=12, adj
# longEMA_8= list_of_data_frame_of_share_market[8]['close'].ewm(span=26, adjus

# shortEMA_9= list_of_data_frame_of_share_market[9]['close'].ewm(span=9, adjus
# signalEMA_9= list_of_data_frame_of_share_market[9]['close'].ewm(span=12, adj
# longEMA_9= list_of_data_frame_of_share_market[9]['close'].ewm(span=26, adjus


#     list_of_EMA =[[shortEMA,signalEMA,longEMA],[shortEMA_1,signalEMA_1,longE
#                  [shortEMA_3,signalEMA_3,longEMA_3],[short_EMA_4,signalEMA_4
#                  [shortEMA_6,signalEMA_6,longEMA_6],[shortEMA_7,signalEMA_7,
#                  [shortEMA_9,signalEMA_9,longEMA_9]]

list_of_EMA =[[shortEMA,signalEMA,longEMA],[shortEMA_1,signalEMA_1,longEMA_1],
                 [shortEMA_3,signalEMA_3,longEMA_3],[short_EMA_4,signalEMA_4,l
```

executed in 23ms, finished 12:58:17 2021-09-06

Reference for analysis

```
list_of_EMA[0][0][0]
list_of_EMA[0][1][0]
list_of_EMA[0][2][0]

list_of_EMA[1][0][0]
list_of_EMA[1][1][0]
list_of_EMA[1][2][0]
```

In [23]:

```
dict_of_latest_CDMA= {}
list_of_top_5_share
for i in list_of_EMA:
    for j in i:
        for l in list_of_top_5_share:
            print(j[0])
            dict_of_latest_CDMA[l] = j[0]
            print("========================================")
    print('...........................')
```

executed in 24ms, finished 12:58:23 2021-09-06

```
32238.2
========================================
32238.2
========================================
32238.2
========================================
32238.2
========================================
32238.2
========================================
32238.2
========================================
32238.2
========================================
32238.2
========================================
32238.2
========================================
32238.2
========================================
```

In [24]:

```
list_of_column_data= []
for i in list_of_EMA:
    for j in range(0,len(i)):
        list_of_column_data.append(i[0][0])
print(len(list_of_column_data))
```

executed in 8ms, finished 12:58:27 2021-09-06

```
15
```

In [25]:

```
length_to_split = [5,5,5]

Inputt = iter(list_of_column_data)
Output = [list(islice(Inputt, elem))
          for elem in length_to_split]

# Printing Output

print("Split length list: ", length_to_split)
print("List after splitting", Output)
```

executed in 18ms, finished 12:58:29 2021-09-06

```
Split length list:  [5, 5, 5]
List after splitting [[58374.2, 58374.2, 58374.2, 32238.2, 32238.2], [32238.2,
18305.65, 18305.65, 18305.65, 19782.95], [19782.95, 19782.95, 19353.15, 19353.1
5, 19353.15]]
```

In [26]:

```
numpy_array = np.array(Output)
transpose = numpy_array.T
```

executed in 12ms, finished 12:58:31 2021-09-06

In [27]:

```
transpose
```

executed in 7ms, finished 12:58:33 2021-09-06

Out[27]:

```
array([[58374.2 , 32238.2 , 19782.95],
       [58374.2 , 18305.65, 19782.95],
       [58374.2 , 18305.65, 19353.15],
       [32238.2 , 18305.65, 19353.15],
       [32238.2 , 19782.95, 19353.15]])
```

In [28]:

```
data_frame =pd.DataFrame(transpose,columns=['shortEMA','signalEMA','LongEMA'])

data_frame
```

executed in 20ms, finished 12:58:35 2021-09-06

Out[28]:

	shortEMA	signalEMA	LongEMA
0	58374.2	32238.20	19782.95
1	58374.2	18305.65	19782.95
2	58374.2	18305.65	19353.15
3	32238.2	18305.65	19353.15
4	32238.2	19782.95	19353.15

In [ ]:

In [29]:

```
data_frame['Stock_name'] = list_of_top_5_share
```

executed in 13ms, finished 12:58:39 2021-09-06

Display data in tabular format , this data autometically update from NSE Website and dislay MACD Line , Signal Line with respect to top 5 share (by Avg_price)

and respective values

In [30]:

```
data_frame
```

executed in 20ms, finished 12:58:41 2021-09-06

Out[30]:

	shortEMA	signalEMA	LongEMA	Stock_name
0	58374.2	32238.20	19782.95	MRF
1	58374.2	18305.65	19782.95	HONAUT
2	58374.2	18305.65	19353.15	PAGEIND
3	32238.2	18305.65	19353.15	SHREECEM
4	32238.2	19782.95	19353.15	3MINDIA