

Stock_Columns

September 29, 2021

Create Features for stock analysis

```
[ ]: # Libraries
import pandas as pd
import numpy as np

import warnings
warnings.filterwarnings("ignore")

import fix_yahoo_finance as yf
from pandas_datareader import data as pdr
yf.pdr_override()
```

```
[ ]: stock = 'MSFT'
start = '2014-01-01'
end = '2018-01-01'
df = pdr.get_data_yahoo(stock, start, end)
```

```
[ ]: df = df.reset_index()
df.head()
```

```
[ ]: # Drop date variable
# df = df.drop(['Date'], 1)

# Daily Returns
df['Returns'] = round(df['Adj Close'].pct_change(), 4)
df.head()
```

```
[ ]: # Log Returns
df['Log Returns'] = np.log(df['Adj Close']) - np.log(df['Adj Close'].shift(1))
df.head()
```

```
[ ]: # Calculate in Rows using axis=1
df['Risk'] = round(df[['Open', 'High', 'Low', 'Adj Close']].std(axis=1), 4)
df.head()
```

```
[ ]: # VWAP
```

```
df['VWAP'] = round(np.cumsum(df['Volume']*(df['High']+df['Low'])/2) / np.
    ↳cumsum(df['Volume']), 2)
df.head()
```

```
[ ]: df['Mean'] = round(df[['Open', 'High', 'Low', 'Adj Close']].mean(axis=1), 2)
df.head()
```

```
[ ]: df['Median'] = round(df[['Open', 'High', 'Low', 'Adj Close']].median(axis=1), 2)
df.head()
```

```
[ ]: df['Mode'] = round(df[['Open', 'High', 'Low', 'Adj Close']].mode(axis=1), 2)
df.head()
```

```
[ ]: df['Variance'] = round(df[['Open', 'High', 'Low', 'Adj Close']].var(axis=1), 4)
df.head()
```

```
[ ]: df['Skew'] = round(df[['Open', 'High', 'Low', 'Adj Close']].skew(axis=1), 4)
df.head()
```

```
[ ]: df['Skew'] = round(df[['Open', 'High', 'Low', 'Adj Close']].skew(axis=1), 4)
df.head()
```

```
[ ]: df['Kurt'] = round(df[['Open', 'High', 'Low', 'Adj Close']].kurt(axis=1), 4)
df.head()
```

```
[ ]: # Standard error of the mean
df['Error'] = df[['Open', 'High', 'Low', 'Adj Close']].sem(axis=1)
df.head()
```

```
[ ]: import talib as ta
```

```
[ ]: # Creating Indicators
n=5
df['RSI']=ta.RSI(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['SMA']= pd.rolling_mean(df['Adj Close'].shift(1),window=n)
df['Corr']= pd.rolling_corr(df['SMA'],df['Adj Close'].shift(1),window=n)
df['SAR']=ta.SAR(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),0.
    ↳2,0.2)
```

```
[ ]: # Momentum Indicator Functions
df['ADX']=ta.ADX(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)), np.
    ↳array(df['Open'].shift(1)), timeperiod=n)
df['ADXR']=ta.ADXR(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),np.array(df['Adj Close']), timeperiod=n)
# df['APO']=ta.APO(np.array(df['Adj Close'].shift(1), fastperiod=12,
    ↳slowperiod=26, matype=0))
```

```

df['AROON_DOWN'], df['AROON_UP']=ta.AROON(np.array(df['High'].shift(1)),np.
    ↳array(df['Low'].shift(1)), timeperiod=n)
df['AROONOSC']=ta.AROONOSC(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),timeperiod=n)
df['BOP']=ta.BOP(np.array(df['Open'].shift(1)),np.array(df['High'].shift(1)),\
    np.array(df['Low']),np.array(df['Adj Close'].shift(1)))
df['CCI']=ta.CCI(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),\
    np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['CMO']=ta.CMO(np.array(df['Adj Close'].shift(1)),\
    ↳timeperiod=n)
df['DX']=ta.DX(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),\
    np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['MACD'], df['MACD_SIGNAL'], df['MACD_HIST'] =ta.MACD(np.array(df['High'].
    ↳shift(1)),fastperiod=12, slowperiod=26, signalperiod=9)
# df['MACDEXT'], df['MACD_SIGNAL'], df['MACD_HIST'] =ta.MACDEXT(np.
    ↳array(df['Adj Close'].shift(1)), fastperiod=12, fastmatype=0, slowperiod=26,\
    ↳slowmatype=0, signalperiod=9, signalmatype=0)
# df['MACDFIX'], df['MACD_SIGNAL'], df['MACD_HIST'] =ta.MACDFIX(np.
    ↳array(df['Adj Close'].shift(1)), signalperiod=9)
df['MFI']=ta.MFI(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),\
    np.array(df['Adj Close'].shift(1)),np.array(df['Volume'].
    ↳shift(1)), timeperiod=n)
df['MINUS_DI']=ta.MINUS_DI(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),\
    np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['MINUS_DM']=ta.MINUS_DM(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)), timeperiod=n)
df['MOM']=ta.MOM(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['PLUS_DI']=ta.PLUS_DI(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),\
    np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['PLUS_DM']=ta.PLUS_DM(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)), timeperiod=n)
df['PPO']=ta.PPO(np.array(df['Adj Close'].shift(1)), fastperiod=12,\
    ↳slowperiod=26, matype=0)
df['ROC']=ta.ROC(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['ROCP']=ta.ROCP(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['ROCR']=ta.ROCR(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['ROCR100']=ta.ROCR100(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['slowk'], df['slowd'] =ta.STOCH(np.array(df['High'].shift(1)),np.
    ↳array(df['Low'].shift(1)),np.array(df['Adj Close']), fastk_period=5,\
    ↳slowk_period=3, slowk_matype=0, slowd_period=3, slowd_matype=0)
df['fastk'], df['fastd'] =ta.STOCHF(np.array(df['High'].shift(1)),np.
    ↳array(df['Low'].shift(1)),np.array(df['Adj Close']), fastk_period=5,\
    ↳fastd_period=3, fastd_matype=0)

```

```
# df['fastk'], df['fastd'] =ta.STOCHRIS(np.array(df['Adj Close'].shift(1)),
    ↳timeperiod=N, fastk_period=5, fastd_period=3, fastd_matype=0)
df['TRIX']=ta.TRIX(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['ULTOSC']=ta.ULTOSC(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),\
        np.array(df['Adj Close']), timeperiod1=7, timeperiod2=14,
    ↳timeperiod3=28)
df['WILLR']=ta.WILLR(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),\
        np.array(df['Adj Close'].shift(1)),
    ↳timeperiod=n)
```

```
[ ]: # Volatility Indicator Functions
df['ATR']=ta.ATR(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),np.
    ↳array(df['Adj Close'].shift(1)), timeperiod=n)
df['NATR']=ta.NATR(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['TRANGE']=ta.TRANGE(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),np.array(df['Adj Close'].shift(1)))
```

```
[ ]: # Volume Indicator Functions
df['AD']=ta.AD(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),np.
    ↳array(df['Adj Close'].shift(1)),np.array(df['Volume'].shift(1)))
df['ADOSC']=ta.ADOSC(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),np.array(df['Adj Close'].shift(1)),np.array(df['Volume'].
    ↳shift(1)),fastperiod=3, slowperiod=10)
df['OBV']=ta.OBV(np.array(df['Adj Close'].shift(1)),np.array(df['Volume'].
    ↳shift(1)))
```

```
[ ]: # Price Transform Functions
df['AVGPRICE']=ta.AVGPRICE(np.array(df['Open'].shift(1)),np.array(df['High'].
    ↳shift(1)),np.array(df['Low'].shift(1)), np.array(df['Adj Close'].shift(1)))
df['MEDPRICE']=ta.MEDPRICE(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)))
df['TYPPRICE']=ta.TYPPRICE(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),np.array(df['Adj Close'].shift(1)))
df['WCLPRICE']=ta.WCLPRICE(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)),np.array(df['Adj Close'].shift(1)))
```

```
[ ]: # Pattern Recognition Fuction
df['Two_Crows'] = ta.CDL2CROWS(np.array(df['Open']), np.array(df['High']), np.
    ↳array(df['Low']), np.array(df['Adj Close']))
df['Three_Crows'] = ta.CDL3BLACKCROWS(np.array(df['Open']), np.
    ↳array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Three_Inside_Up_Down'] = ta.CDL3INSIDE(np.array(df['Open']), np.
    ↳array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
```

```

df['Three_Line_Strike'] = ta.CDL3LINESTRIKE(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Thre_Outside_Up_Down'] = ta.CDL3OUTSIDE(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Thre_Stars_In_The_South'] = ta.CDL3STARSINSOUTH(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Three_Advancing_White_Soldiers'] = ta.CDL3WHITESOLDIERS(np.
    →array(df['Open']), np.array(df['High']), np.array(df['Low']), np.
    →array(df['Adj Close']))
df['Abandoned_Baby'] = ta.CDLABANDONEDBABY(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']),
    →penetration=0)
df['Advanced_Block'] = ta.CDLADVANCEBLOCK(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Belt_hold'] = ta.CDLBELTHOLD(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Breakaway'] = ta.CDLBREAKAWAY(np.array(df['Open']), np.array(df['High']),
    →np.array(df['Low']), np.array(df['Adj Close']))
df['Closing_Marubozu'] = ta.CDLCLOSINGMARUBOZU(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Concealing_Baby_Swallow'] = ta.CDLCONCEALBABYSWALL(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Counterattack'] = ta.CDLCOUNTERATTACK(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Dark_Cloud_Cover'] = ta.CDLDARKCLOUDCOVER(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']),
    →penetration=0)
df['Doji'] = ta.CDLDOJI(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Doji_Star'] = ta.CDLDOJISTAR(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Dragonfly_Doji'] = ta.CDLDRAGONFLYDOJI(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Engulfing_Pattern'] = ta.CDLENGULFING(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Evening_Doji_Star'] = ta.CDLEVENINGDOJISTAR(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']),
    →penetration=0)
df['Evening_Star'] = ta.CDLEVENINGSTAR(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']),
    →penetration=0)
df['Up_Down_gap_side_by_side_white_lines'] = ta.CDLGAPSIDESIDEWHITE(np.
    →array(df['Open']), np.array(df['High']), np.array(df['Low']), np.
    →array(df['Adj Close']))
df['Gravestone_Doji'] = ta.CDLGRAVESTONEDOJI(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))

```

```

df['Hammer'] = ta.CDLHAMMER(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Hanging_Man'] = ta.CDLHANGINGMAN(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Harami_Pattern'] = ta.CDLHARAMI(np.array(df['Open']), np.array(df['High']),
    →np.array(df['Low']), np.array(df['Adj Close']))
df['Harami_Cross_Pattern'] = ta.CDLHARAMICROSS(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['High_Wave_Candle'] = ta.CDLHIGHWAVE(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Hikkake_Pattern'] = ta.CDLHIKKAKE(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Modified_Hikkake_Pattern'] = ta.CDLHIKKAKEMOD(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Homing_Pigeon'] = ta.CDLHOMINGPIGEON(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Identical_Three_Crows'] = ta.CDLIDENTICAL3CROWS(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['In_Neck_Pattern'] = ta.CDLINNECK(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Inverted_Hammer'] = ta.CDLINVERTEDHAMMER(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Kicking'] = ta.CDLKICKING(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Kicking_Bull_Bear'] = ta.CDLKICKINGBYLENGTH(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Ladder_Bottom'] = ta.CDLLADDERBOTTOM(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Long_Legged_Doji'] = ta.CDLLONGLEGGEDDOJI(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Long_line_Candle'] = ta.CDLLONGLINE(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Marubozu'] = ta.CDLMARUBOZU(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Matching_Low'] = ta.CDLMATCHINGLOW(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Mat_Hold'] = ta.CDLMATHOLD(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Morning_Star'] = ta.CDLMORNINGSTAR(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['On_Neck_Pattern'] = ta.CDLONNECK(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Piercing_Pattern'] = ta.CDLPIERCING(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))

```

```

df['Rickshaw_Man'] = ta.CDLRICKSHAWMAN(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Rising_Falling_Three_Method'] = ta.CDLRISEFALL3METHODS(np.
    →array(df['Open']), np.array(df['High']), np.array(df['Low']), np.
    →array(df['Adj Close']))
df['Separating_Lines'] = ta.CDLSEPARATINGLINES(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Shooting_Star'] = ta.CDLSHOOTINGSTAR(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Short_Line_Candle'] = ta.CDLSHORTLINE(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Spinning_Top'] = ta.CDLSPINNINGTOP(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Stalled_Pattern'] = ta.CDLSTALLEDPATTERN(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Stick_Sandwich'] = ta.CDLSTICKSANDWICH(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Takuri'] = ta.CDLTAKURI(np.array(df['Open']), np.array(df['High']), np.
    →array(df['Low']), np.array(df['Adj Close']))
df['Tasuki_Gap'] = ta.CDLTASUKIGAP(np.array(df['Open']), np.array(df['High']),
    →np.array(df['Low']), np.array(df['Adj Close']))
df['Thrusting_Pattern'] = ta.CDLTHRUSTING(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Tristar_Pattern'] = ta.CDLTRISTAR(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Unique_3_River'] = ta.CDLUNIQUE3RIVER(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Upside_Gap_Two_Crows'] = ta.CDLUPSIDEGAP2CROWS(np.array(df['Open']), np.
    →array(df['High']), np.array(df['Low']), np.array(df['Adj Close']))
df['Upside_Downside_Gap_Three_Methods'] = ta.CDLXSIDEGAP3METHODS(np.
    →array(df['Open']), np.array(df['High']), np.array(df['Low']), np.
    →array(df['Adj Close']))

```

```

[ ]: # Cycle Indicator Functions
df['HT_DCPERIOD']=ta.HT_DCPERIOD(np.array(df['Adj Close'].shift(1)))
df['HT_DCPHASE']=ta.HT_DCPHASE(np.array(df['Adj Close'].shift(1)))
df['inphase'], df['quadrature']=ta.HT_PHASOR(np.array(df['Adj Close'].shift(1)))
df['sine'], df['leadsine']=ta.HT_SINE(np.array(df['Adj Close'].
    →shift(1)))
df['HT_TRENDMODE']=ta.HT_TRENDMODE(np.array(df['Adj Close'].
    →shift(1)))

df['ATR1']=abs(np.array(df['High'].shift(1)) - np.array(df['Low'].shift(1)))
df['ATR2']=abs(np.array(df['High'].shift(1)) - np.array(df['Adj Close'].
    →shift(1)))

```



```

df['ATR3']=abs(np.array(df['Low'].shift(1)) - np.array(df['Adj Close'].
    ↳shift(1)))
df['AverageTrueRange'] = df[['ATR1', 'ATR2', 'ATR3']].max(axis=1)

df['EMA']=pd.Series(pd.ewma(df['Adj Close'], span = n, min_periods = n - 1))

```

[]: *# Statistic Functions*

```

df['Beta']=ta.BETA(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),↳
    ↳timeperiod=n)
df['CORREL']=ta.CORREL(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)), timeperiod=n)
df['LINEARREG']=ta.LINEARREG(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['LINEARREG_ANGLE']=ta.LINEARREG_ANGLE(np.array(df['Adj Close'].shift(1)),↳
    ↳timeperiod=n)
df['LINEARREG_INTERCEPT']=ta.LINEARREG_INTERCEPT(np.array(df['Adj Close'].
    ↳shift(1)), timeperiod=n)
df['LINEARREG_SLOPE']=ta.LINEARREG_SLOPE(np.array(df['Adj Close'].shift(1)),↳
    ↳timeperiod=n)
df['STDDEV']=ta.STDDEV(np.array(df['Adj Close'].shift(1)), timeperiod=n,↳
    ↳nbdev=1)
df['Time Series Forecast']=ta.TSF(np.array(df['Adj Close'].shift(1)),↳
    ↳timeperiod=n)
df['VAR']=ta.VAR(np.array(df['Adj Close'].shift(1)), timeperiod=n, nbdev=1)

```

[]: *# Overlap Studies Functions*

```

df['upperband'], df['middleband'], df['lowerband']=ta.BBANDS(np.array(df['Adj↳
    ↳Close'].shift(1)), timeperiod=n, nbdevup=2, nbdevdn=2, matype=0)
df['DEMA']=ta.DEMA(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['EMA']=ta.EMA(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['HT_TRENDLINE']=ta.HT_TRENDLINE(np.array(df['Adj Close'].shift(1)))
df['KAMA']=ta.KAMA(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['MA']=ta.MA(np.array(df['Adj Close'].shift(1)), timeperiod=n, matype=0)
df['mama'],df['fama'] = ta.MAMA(np.array(df['Adj Close'].shift(1)),↳
    ↳fastlimit=0, slowlimit=0)
df['MAVP'] =ta.MAVP(np.array(df['Adj Close'].shift(1)),periods, minperiod=2,↳
    ↳maxperiod=30, matype=0)
df['MIDPOINT']=ta.MIDPOINT(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['MIDPRICE']=ta.MIDPRICE(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)), timeperiod=n)
df['SAR']=ta.SAR(np.array(df['High'].shift(1)),np.array(df['Low'].shift(1)),↳
    ↳acceleration=0, maximum=0)

df['SAREXT']=ta.SAREXT(np.array(df['High'].shift(1)),np.array(df['Low'].
    ↳shift(1)), startvalue=0, offsetonreverse=0, accelerationinitlong=0,↳
    ↳accelerationlong=0, accelerationmaxlong=0, accelerationinitshort=0,↳
    ↳accelerationshort=0, accelerationmaxshort=0)

```



```

df['SMA']=ta.SMA(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['T3']=ta.T3(np.array(df['Adj Close'].shift(1)), timeperiod=n, vfactor=0)
df['TEMA']=ta.TEMA(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['TRIMA']=ta.TRIMA(np.array(df['Adj Close'].shift(1)), timeperiod=n)
df['WMA']=ta.WMA(np.array(df['Adj Close'].shift(1)), timeperiod=n)

```

```

[ ]: df['20d_ma'] = df['Adj Close'].shift(1).rolling(window=20).mean()
df['50d_ma'] = df['Adj Close'].shift(1).rolling(window=50).mean()
df['Bol_upper'] = df['Adj Close'].shift(1).rolling(window=20).mean() + 2*
↳df['Adj Close'].shift(1).rolling(window=20).std()
df['Bol_lower'] = df['Adj Close'].shift(1).rolling(window=20).mean() - 2*
↳df['Adj Close'].shift(1).rolling(window=20).std()
df['Bol_BW'] = ((df['Bol_upper'] - df['Bol_lower'])/df['20d_ma'])*100
df['Bol_BW_200MA'] = df['Bol_BW'].shift(1).rolling(window=50).mean()
df['Bol_BW_200MA'] = df['Bol_BW_200MA'].fillna(method='backfill')
df['20d_ewma'] = df['Adj Close'].shift(1).ewm(span=20).mean()
df['50d_ewma'] = df['Adj Close'].shift(1).ewm(span=50).mean()

```