

```
# !pip install quandl
# !pip install hmmlearn
# !pip install yfinance

import quandl
import numpy as np
import pandas as pd
import statsmodels.api as sm
from scipy import stats
from matplotlib import cm, pyplot as plt
from hmmlearn.hmm import GaussianHMM
import scipy
import datetime
import json
import seaborn as sns
import joblib

import yfinance as yf
import plotly.express as px

import warnings
warnings.filterwarnings('ignore')
```

```
from scipy.stats import mode

asset = '^RUA'
data_df = yf.download(asset,
                      start='1990-01-01',
                      end='2021-12-31',
                      progress=False)

freq = 'W-MON'
df_W = data_df.resample(freq).first()[['Close']]

df_W['return'] = df_W['Close']/df_W['Close'].shift(1) - 1
df_W['ret_avg'] = df_W['return'].rolling(8).mean()

conditions = [ df_W['ret_avg'] <= -0.004, (df_W['ret_avg'] > -0.004) & (df_W['ret_avg'] <= 0), df_W['ret_avg'] > 0 ]
choices = [ -1, 0, 1 ]

df_W['Indicator'] = np.select(conditions, choices, default=np.nan)
df_W['final_state'] = df_W['Indicator'].rolling(window=10).apply(lambda x: mode(x)[0]).shift(-9)
```

```
df_W_new = df_W[['final_state']]
df_W_new = df_W_new.ffill().bfill()
df_W_new
```

final_state

```
df_W_new[['final_state']].to_pickle('/data/workspace_files/rule_based.pickle')
df_W_new[['final_state']].to_pickle('/data/workspace_files/output/rule_based.pickle')
```

```
px.scatter(df_W[df_W.index<pd.to_datetime('2018-01-01')]['Close'],
          color=df_W_new[df_W_new.index<pd.to_datetime('2018-01-01')]['final_state'],
          ).update_layout(yaxis_title='Close').update_coloraxes(showscale=False)
```

