Regime detection and prediction in Financial Markets Lesson 2: Application of Gaussian Mixture Model

4 min read · Jul 3, 2022

T-Ballz Finance Follow

Hello guys. Welcome back to the 2nd lesson of Market regime classification/detection in Financial Markets series, presented by T-ballz Finance. Last time I briefly covered K-means clustering method and its application for regime classification in financial markets. This weekend, I am going to walk through an alternative but slightly more conplex method. It's called Gaussian Mixture Model.

When people from academic Finance studied the stock market in the old days, they assumed that the stock market returns follow the Normal distribution. Later on they realized that's certainly not necessarily true. Instead, one can possibly divide the stock returns into different subgroups/regimes but each of them can be normally distributed instead. Gaussian Mixture Model is therefore a useful method for classifying market regimes. Gaussian Mixture model, similar to the family of Markov switching models or finite mixture models, is estimated by maximum likelihood method using the Expectation-Maximization (EM) algorithm. I am not going to cover EM algorithm here as it's not meaningful to go through the math. As a quant trader/investor, most of us are practitioners, we only want to apply these tools to help build our trading strategies so It's enough to just know what these tools are used for and how to apply them, we can skip the underlying theories. (But if you are really interested, you should find some more material and learn the math behind it. There's lots of free material for stats and machine learning online)

For example, for the difference between k-means and GMM, you can check this stackexchange post: clustering — In cluster analysis, how does Gaussian mixture model differ from K Means when we know the clusters are spherical? — Cross Validated (stackexchange.com)

Alright, enough talk, let's get our hands dirty. I'll use the same dataset as last time. Using the yfinance package to download the S&P 500 and VIX data from Yahoo Finance, I calculate the corresponding ATR-Price ratio, relative volume and daily return of S&P 500:

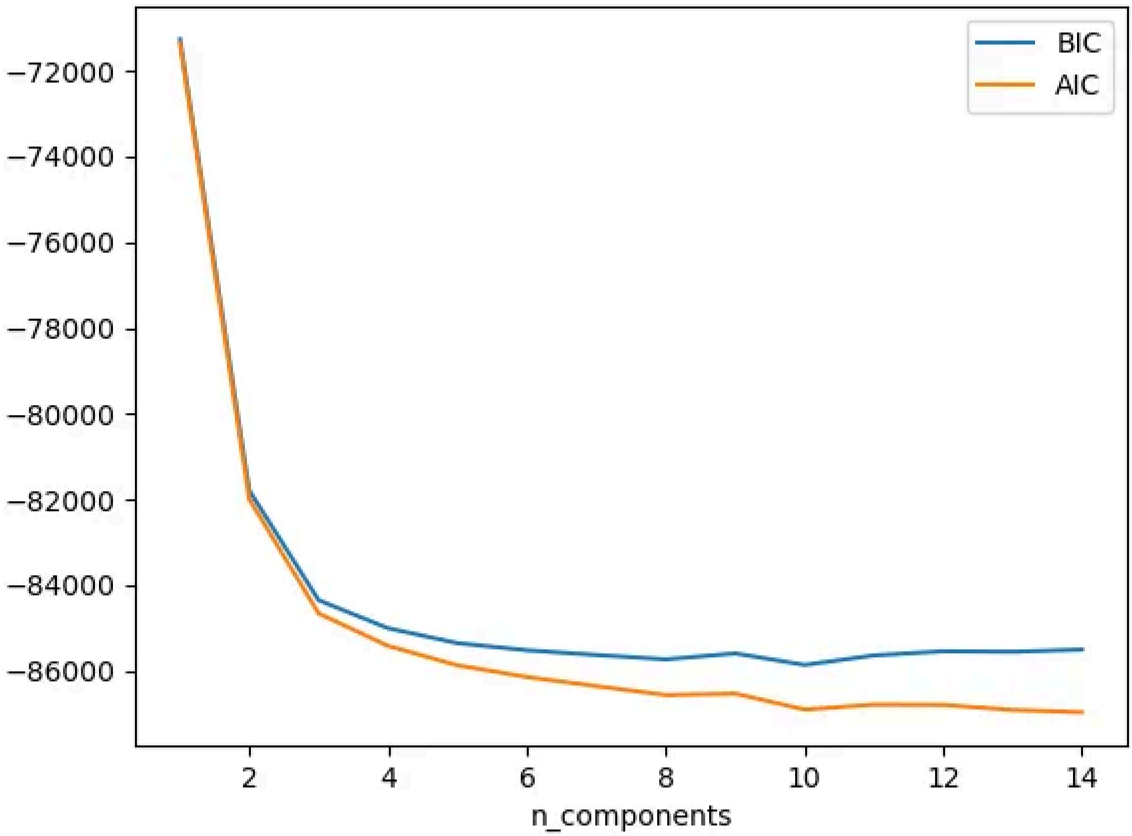
```
import yfinance as yf
import datetime
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import Stock_quant.ATR
import Stock_quant.wwma
from sklearn.mixture import GaussianMixture

pd.set_option('display.max_columns', None)

#Get the data ready
end_time = datetime.date.today()
start_time = end_time - datetime.timedelta(days = 11850)
df = yf.download('^GSPC',start = start_time, end = end_time)
df_replace = yf.download('^GSPC',start = start_time, end = end_time)


VIX = yf.download('^VIX',start = start_time, end = end_time)
df['daily return'] = df['Close']/ df['Close'].shift(1) -1
df['ATR_price'] = Stock_quant.ATR.ATR(df['Low'], df['High'],df['Close'], 14)/df['Close']
df['relative_Volume'] = df['Volume'] / df['Volume'].rolling(40).mean()
df['VIX'] = VIX['Close']
df = df[['daily return',  'ATR_price', 'VIX', 'relative_Volume']]
df = df[40:]
df_replace = df_replace[40:]
```

Just like the previous lesson, fire up python ,import the necessary packages and load the data

First step we need to do is determine the optimal number of 'components' for the GMM model. For simplicity, think of components just as the clusters/regimes. To determine the optimal number, we use the AIC and BIC information criteria. For those who took statistics and econometrics courses in college, you should be familiar with them. For those who have no idea what they are, all you need to know is we pick the number that minimize both the AIC and BIC:

```
#Check how many regimes we should have optimally
n_components = np.arange(1, 15)
models = [GaussianMixture(n, covariance_type='full', random_state=0).fit(df) for n in n_components]
plt.plot(n_components, [m.bic(df) for m in models], label='BIC')
plt.plot(n_components, [m.aic(df) for m in models], label='AIC')
plt.legend(loc='best')
plt.xlabel('n_components')
plt.show()
```

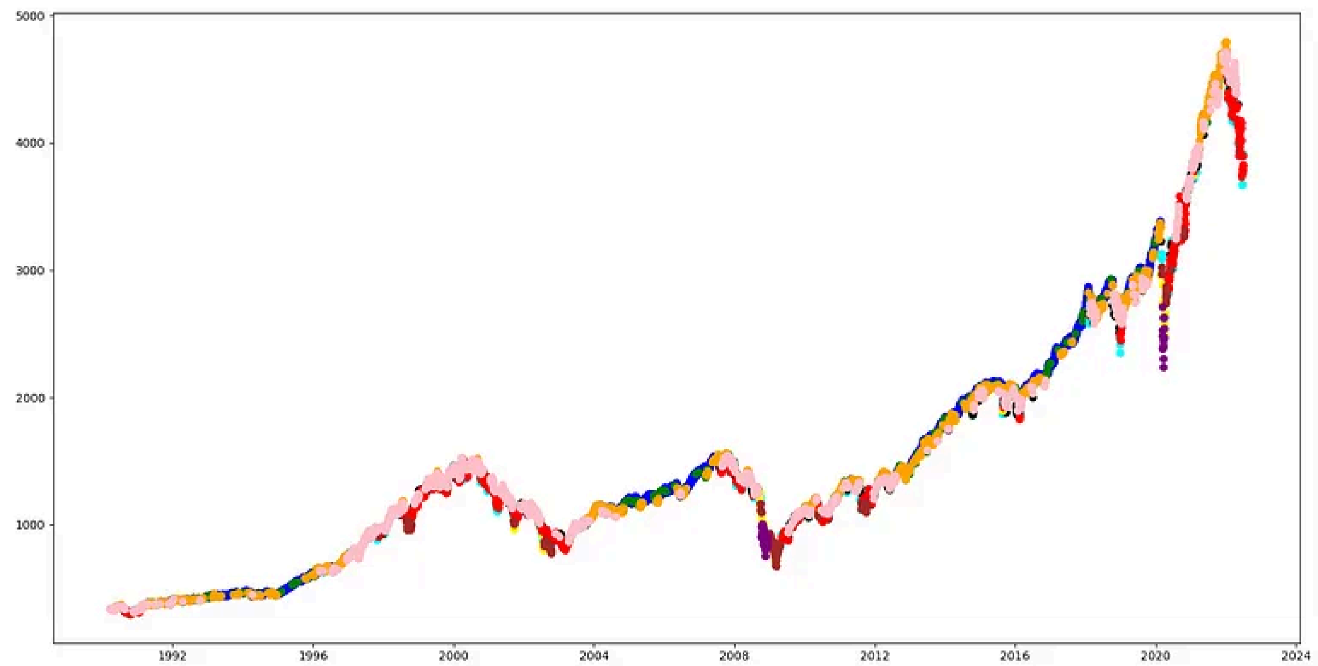
Determining the optimal number of components/regimes



Seems 10 regimes are the optimal…… so let's go ahead with 10 regimes

```
gmm = GaussianMixture(n_components=10)
gmm.fit(df)

#predictions from gmm
labels = gmm.predict(df)
frame = pd.DataFrame(df)
df_replace['regime'] = labels


color=['blue','green','cyan', 'black', 'yellow', 'red','brown','orange','purple','pink']
#color=['blue','green','cyan', 'black']
for k in range(0,10):
    data = df_replace[df_replace["regime"]==k]
    plt.scatter(data.index,data['Close'],c=color[k])

plt.show()
```

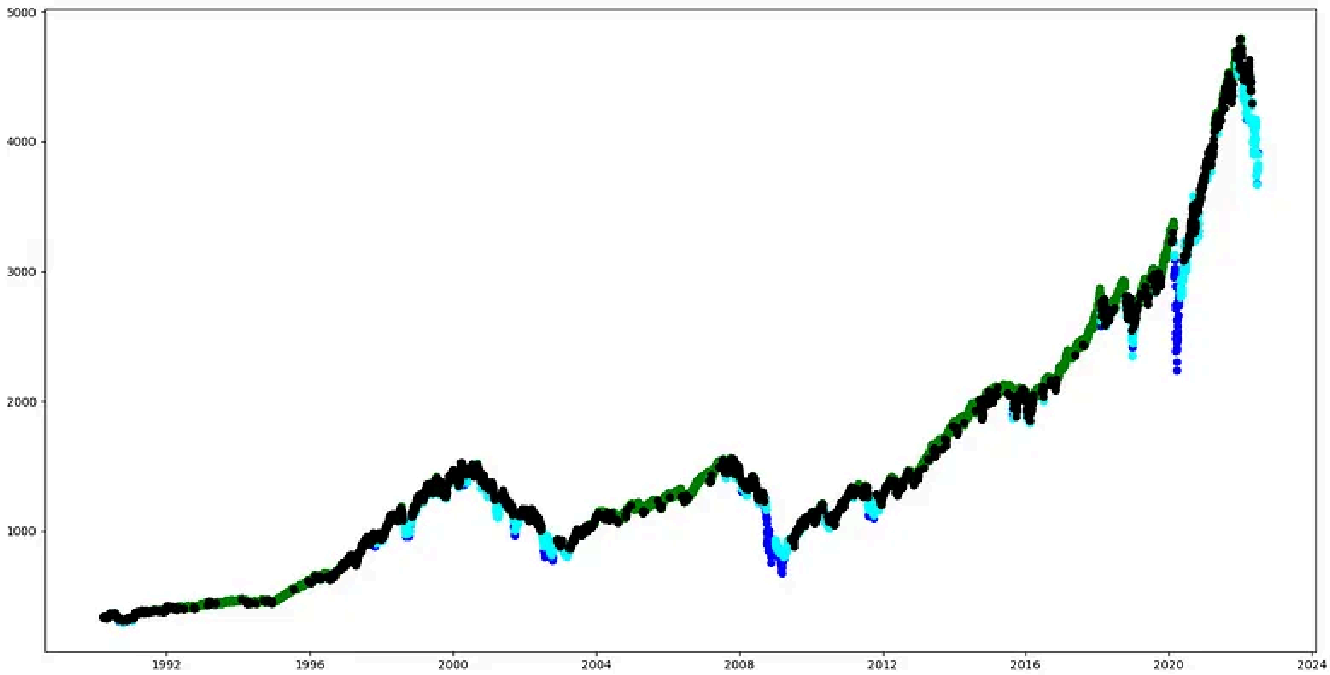
Estimate the model with 10 components and result visualization using Matplotlib



Ok, we can see that whenever the market is in the 'purple' regime, it's close to bottoming. But 10 regimes seem too much even though the model suggest it's the best. What if we just go with 4 regimes, just like K-means suggest? Let's run the code again but specify n_components = 4 only this time:

```
gmm = GaussianMixture(n_components=4)
gmm.fit(df)

#predictions from gmm
labels = gmm.predict(df)
frame = pd.DataFrame(df)
df_replace['regime'] = labels


#color=['blue','green','cyan', 'black', 'yellow', 'red','brown','orange','purple','pink']
color=['blue','green','cyan', 'black']
for k in range(0,4):
    data = df_replace[df_replace["regime"]==k]
    plt.scatter(data.index,data['Close'],c=color[k])

plt.show()
```

Run the code again but n_components = 4 this time



The visualized results look more or less the same as the k-means results from the previous lesson. It seems both the 'buy and hold' and 'buy the dip' are both good strategies to be used in the 'Green regime' . In the cyan regime, you probably should sell and maybe consider shorting the market too to hedge your portfolio. Compared to K-means, the GMM doesn't seem to be better at finding the bottoming regime (blue regime here or brown regime in the previous lesson), market seems always close to bottom if there's consecutive blue dots clustering together.

I plan to make 1 or 2 more lessons about market regime classification (one for trend filtering algorithm, the other one for model based classification such as regime switching/ finite mixture models) depending how many people really like my educational contents or I could move to other areas of quant trading/finance. I hope you enjoy my medium articles and follow me on medium or follow me on twitter:

T-Ballz🔺 (@Clark_SM7) / Twitter

Quantitative Finance | Quantitative Strategies | Trading | Investing | Stock Market

Follow

Written by T-Ballz Finance

368 Followers · 2 Following

I am a PhD dropout who trades stocks, crypto, shitcoins, options, futures, NFTs.

Responses (2)

Wangchuyin

What are your thoughts?

Demid Fedorov
Jul 4, 2022

Awesome article, would love to see part 3!

1 Reply

arbitech
Jul 5, 2022

No predictability powerness. By nature mixtures, and hidden Markova chains are backward calculating.

1 reply Reply

More from T-Ballz Finance

T-Ballz Finance

Retrieving Historical P/E data with Python

Hey guys, sorry for not updating here for a while. I have been pondering what to talk about the topics of market volatility lately. I am...

Aug 22, 2022 29 2

```
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
import ATR
import RSI
from sklearn.cluster import KMeans
sns.set()

pd.set_option('display.max_columns', None)
end_time = datetime.date.today()
start_time = end_time - datetime.timedelta(days = 11850)
df = yf.download('^GSPC',start = start_time, end = end_time)
df_replace =  yf.download('^GSPC',start = start_time, end = end_time)
```

T-Ballz Finance

Regime detection and prediction in Financial Markets Lesson 1: Simple tutorial

Have you guys ever wondered why your trading strategy doesn't work well? It is possible that all simple strategies can work probably well...

Jun 19, 2022 119 1

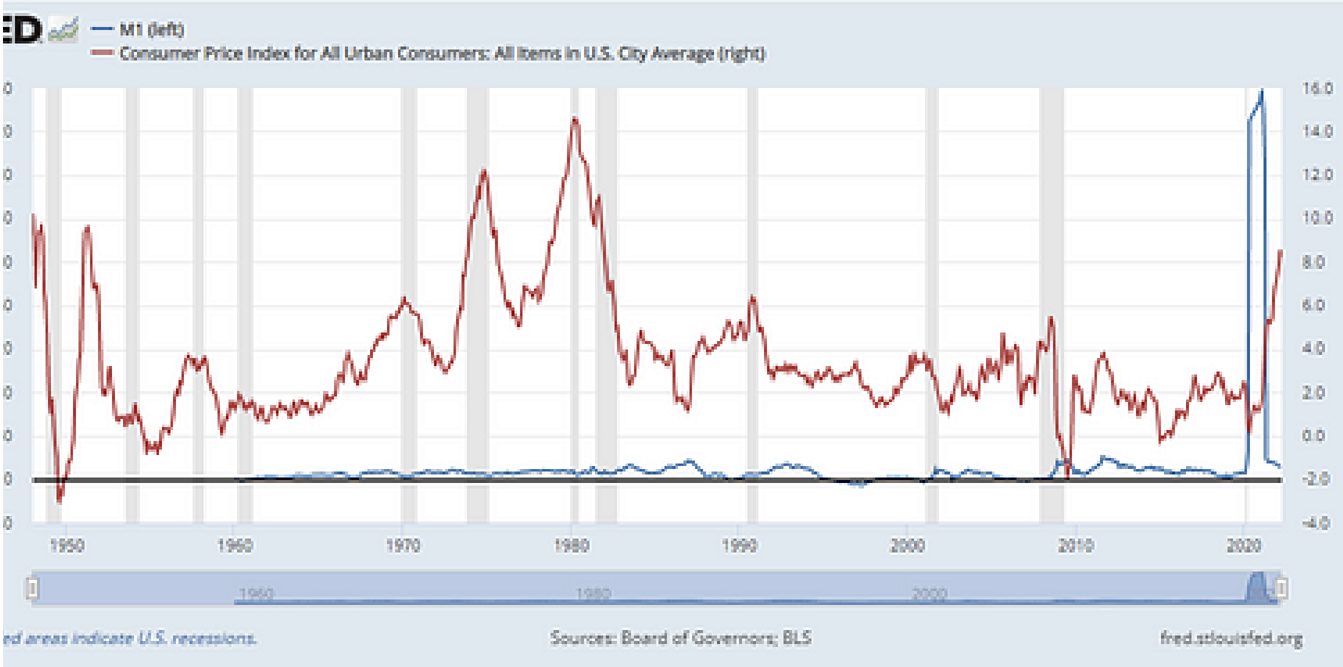
```
for i in range(0,len(df_replace)):
    if df_replace['regime'].iloc[i] == 0:
        col.append('blue')
    elif df_replace['regime'].iloc[i] == 1:
        col.append('red')
    elif df_replace['regime'].iloc[i] ==2:
        col.append('orange')
    else:
        col.append('brown')

print(df.groupby(by=df['regime']).mean())

print(df[df['regime']==2])

plt.scatter(df_replace.index, df_replace['Close'], c= col)
plt.show()


df.to_csv(r'C:\Users\clark\PycharmProjects\Stock_quant\Data\df_regime.csv')
df_replace.to_csv(r'C:\Users\clark\PycharmProjects\Stock_quant\Data\df_regime_replace.csv')
```

T-Ballz Finance

Market Regime Detection and Prediction Lesson 3: Prediction via supervised learning methods

Hey guys, it's been 2 weeks since my previous article. I was busy with something but finally I got some time this weekend. This could very...

Jul 24, 2022 31 3



T-Ballz Finance

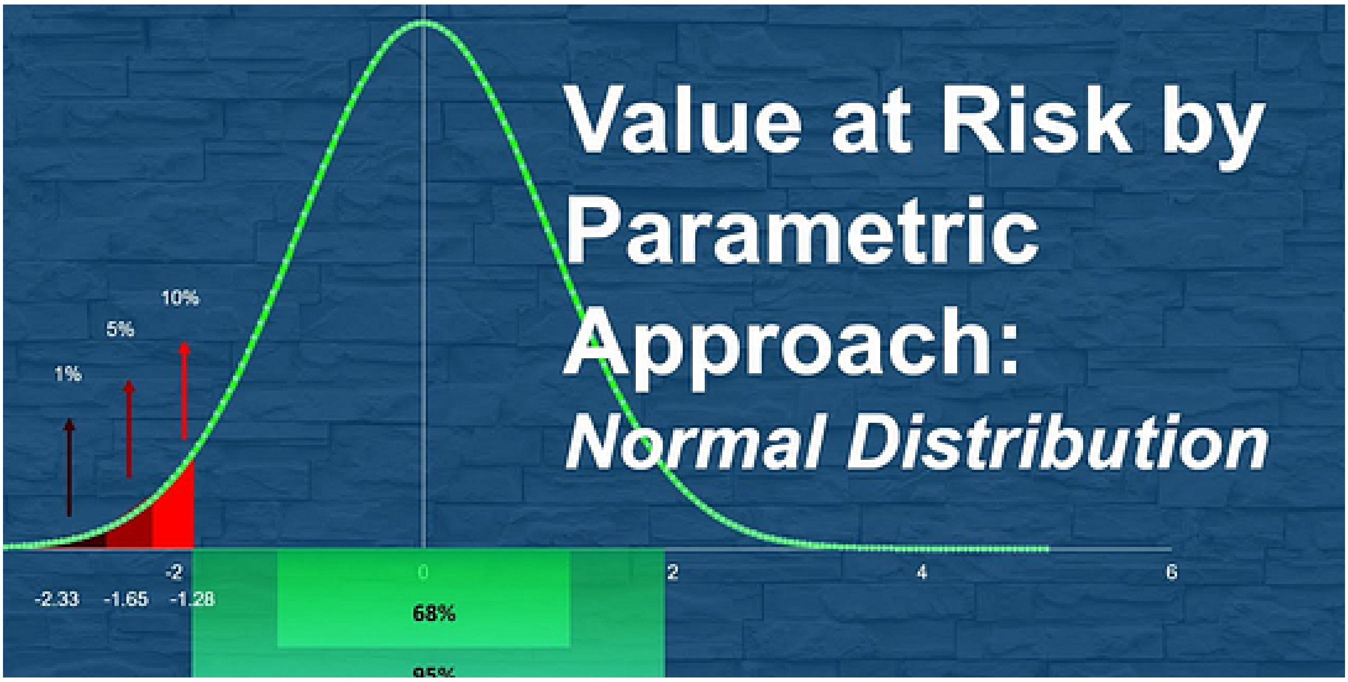
T-ballz Finance: Forecasting Inflation with Vector Autoregressive model using Python

Hey guys, today I am going to present you how I forecast inflation using a time series Econometric model in Python. Macro forecasting is...

May 28, 2022 39

See all from T-Ballz Finance

Recommended from Medium



Roi Polanitzer

Using the VaR Model in the Analytical Approach to Measure the Risks of Investing in the S&P 500...

1. The VaR Model

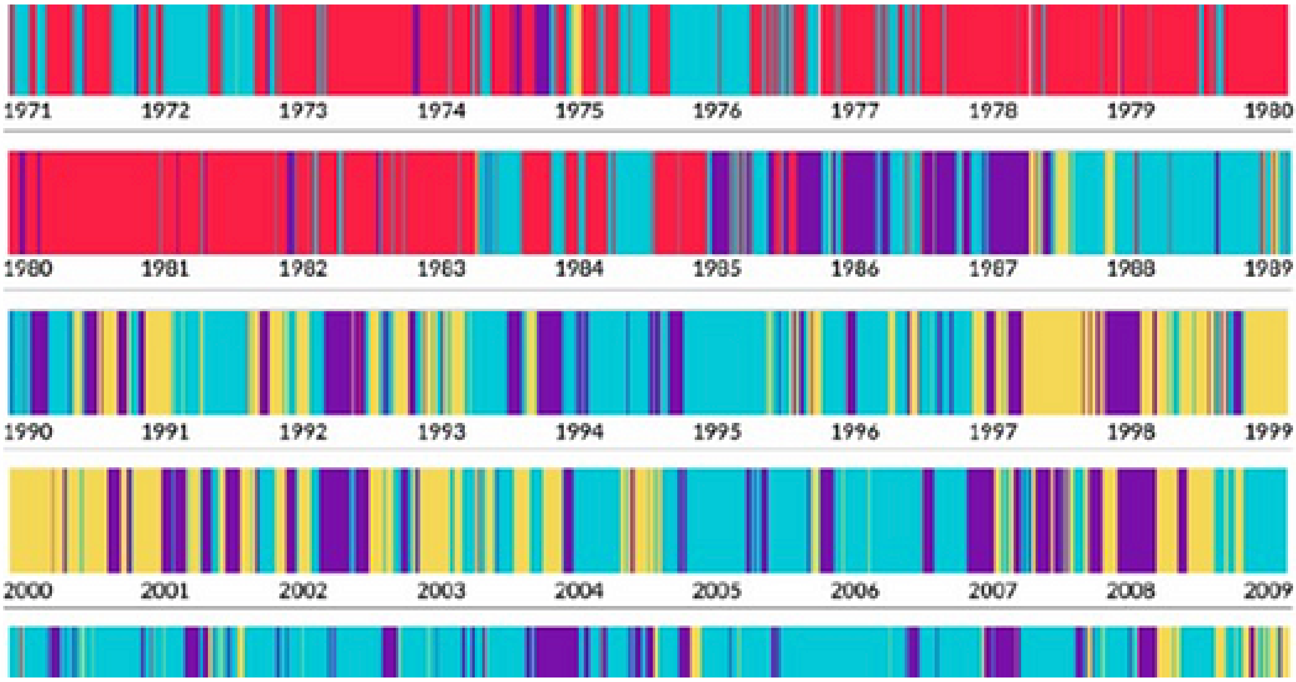
Apr 8

Aakash Chavan Ravindranath, Ph.D

Hidden Markov Model and the Medilian Fund: A Deep Dive into Market Prediction

Introduction In finance, making informed predictions is key to successful trading. One powerful tool that helps in this endeavour is the...

Mar 2 1



Big Universe Little Time

A Hidden Markov Model Analysis for Gold Historical Data

What is “Hidden States” concept? Why are the states “Hidden” and not observable? How can we foresee the states in Candlestick Chart data...

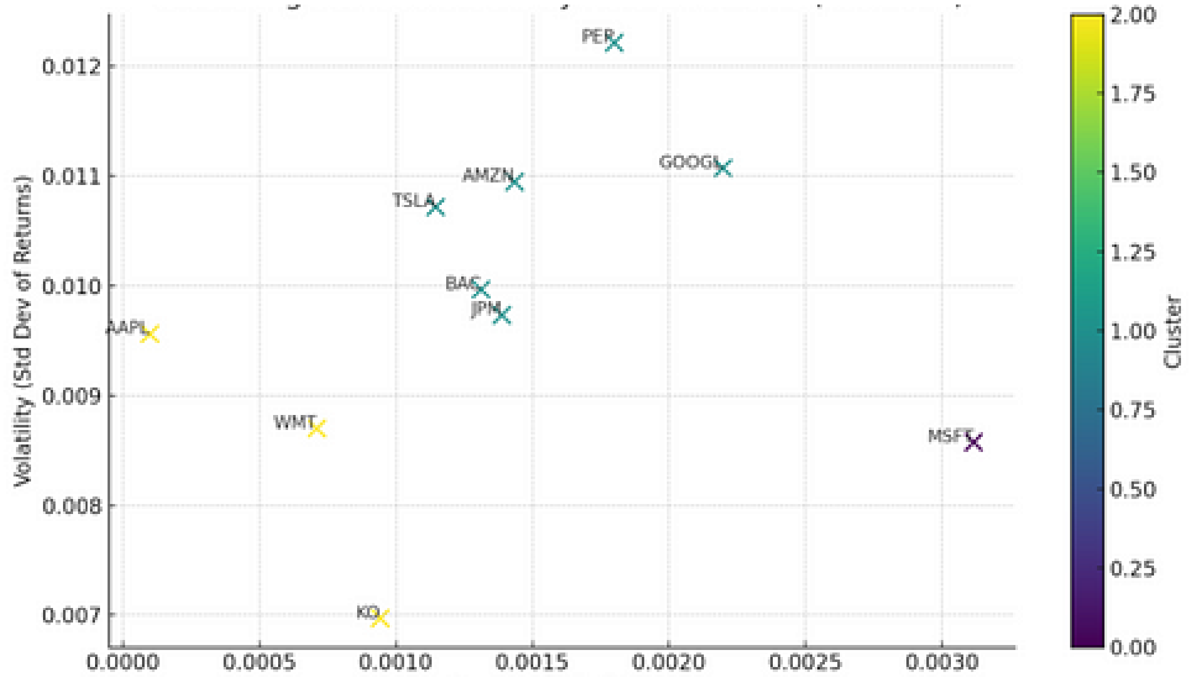
Nov 4, 2024 35

In Funny AI & Quant by Pham The Anh

Unlocking Alpha: Advanced Exploratory Data Analysis Techniques in Quantitative Trading

In the dynamic realm of quantitative trading, the ability to extract meaningful insights from financial data sets the foundation for...

Dec 27, 2024 1



In Coding Nexus by Code Pulse

Detecting Market Regimes: A New Approach Using Clustering Algorithms

Imagine you're a chef trying to figure out the secret ingredients in a stew without a recipe. You taste it, identify patterns, and group...

Feb 21 46 4

Kridtapon P.

Trading Bitcoin with Confidence: A Deep Dive into Indicator-Based Optimization

Building a custom Bitcoin strategy with low drawdown and high consistency using heatmaps, signal triggers, and backtesting

Apr 22 7

See more recommendations