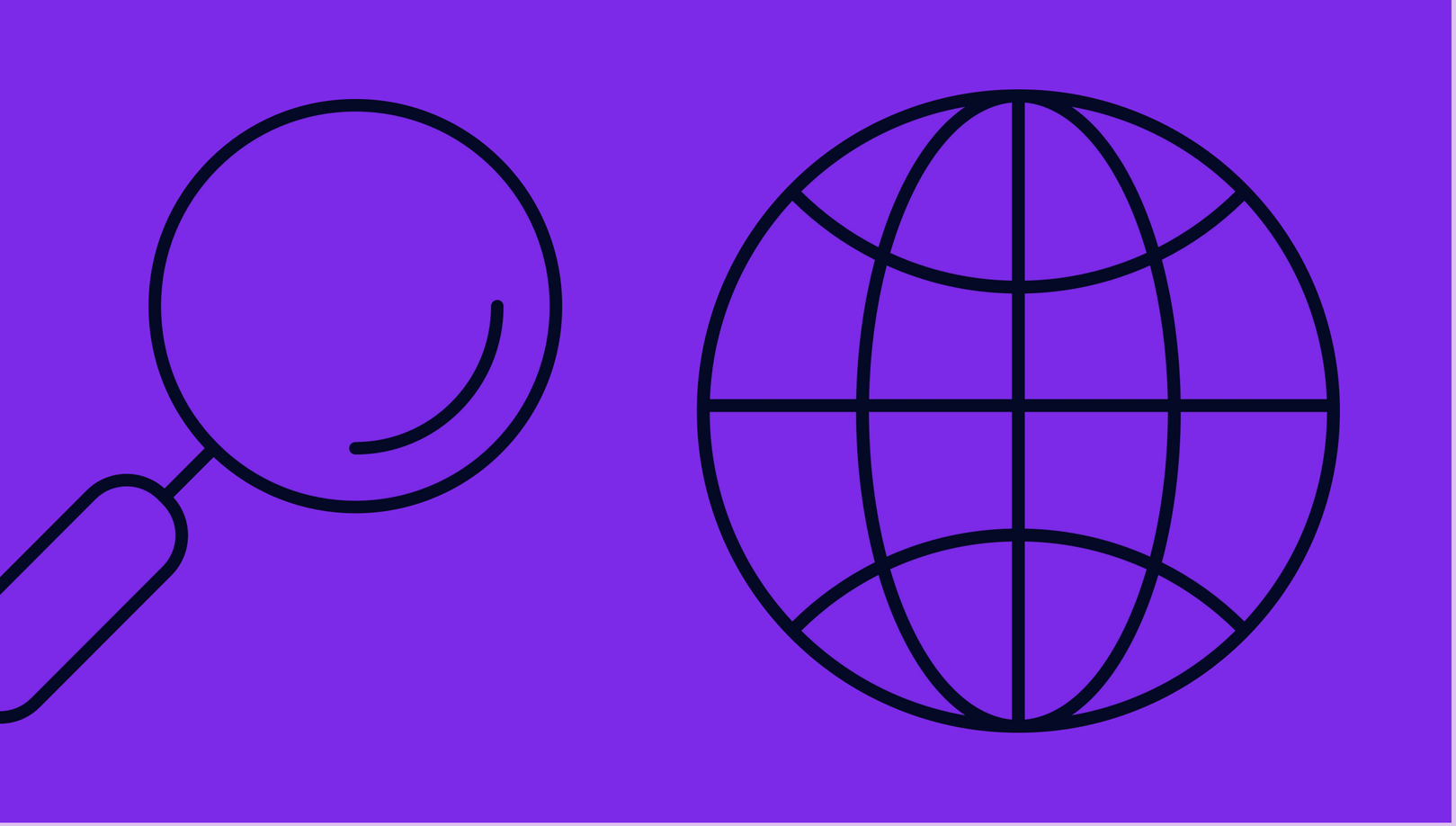
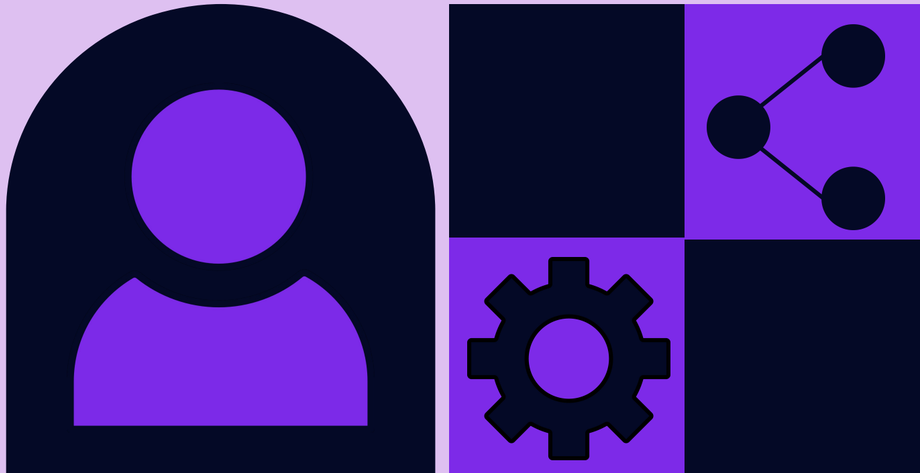
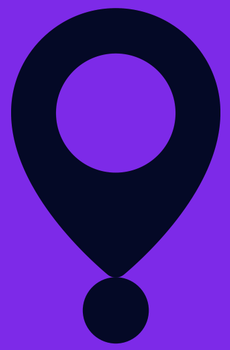


Group 11

Analyze stock market using Clustering methods



Thành viên nhóm 11



Phạm Thanh Tùng

2053571

Phùng Lê Khánh Trình

2053535

Nguyễn Thành Vinh

2053592

Lại Đức Trung

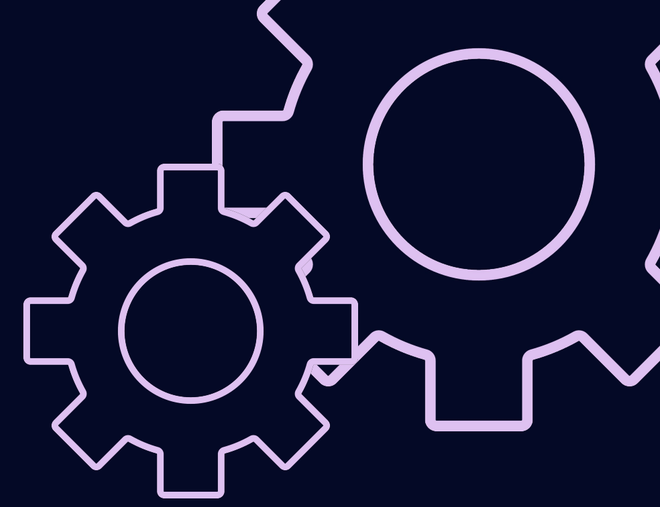
2053538

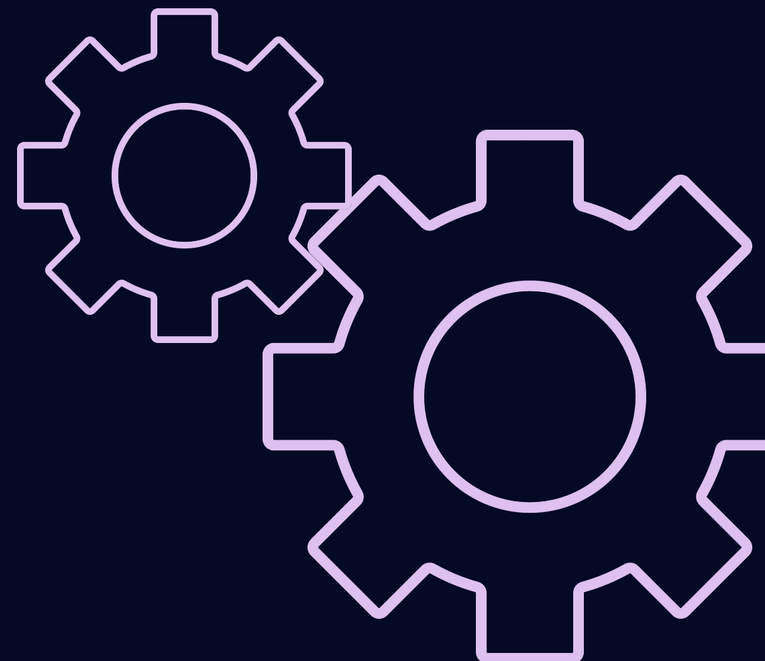
Nội Dung Trình Bày

I. Giới Thiệu Vấn Đề

II. Giải Pháp

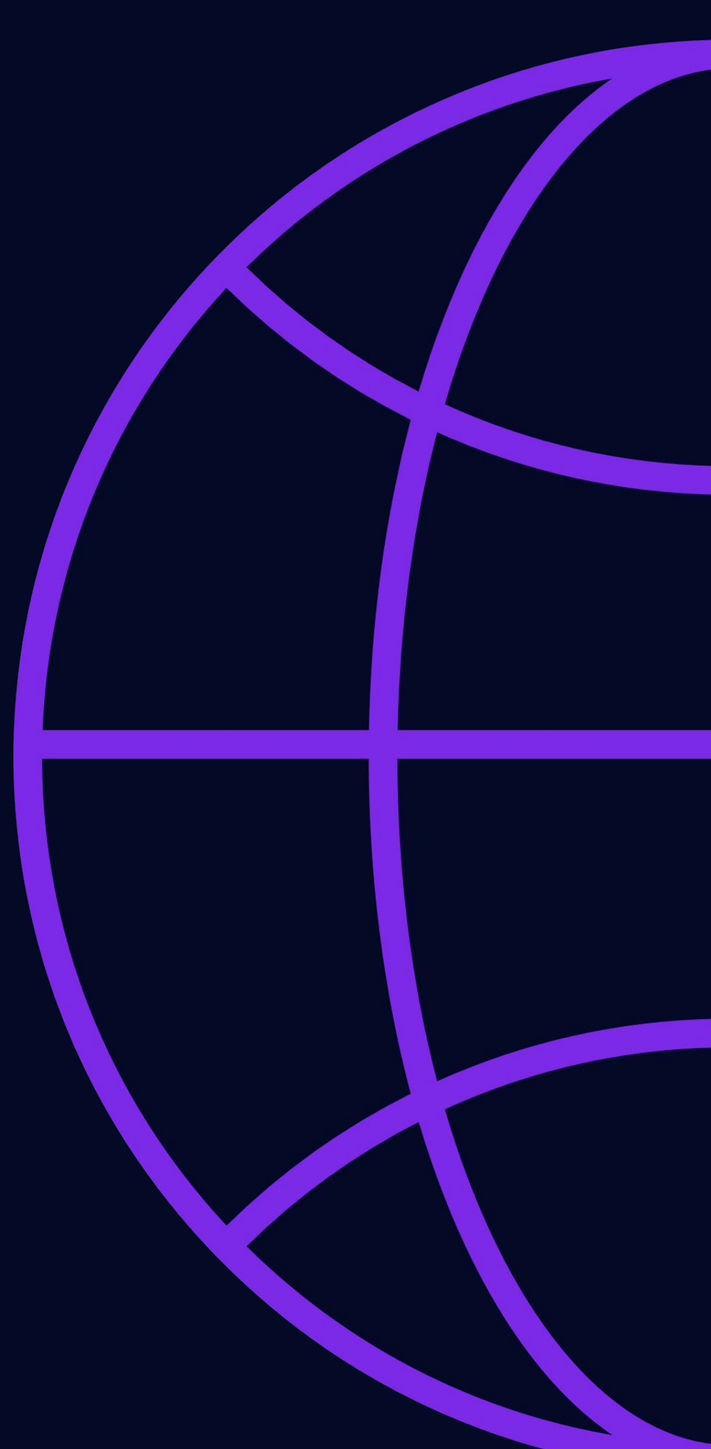
III. Hiện Thực

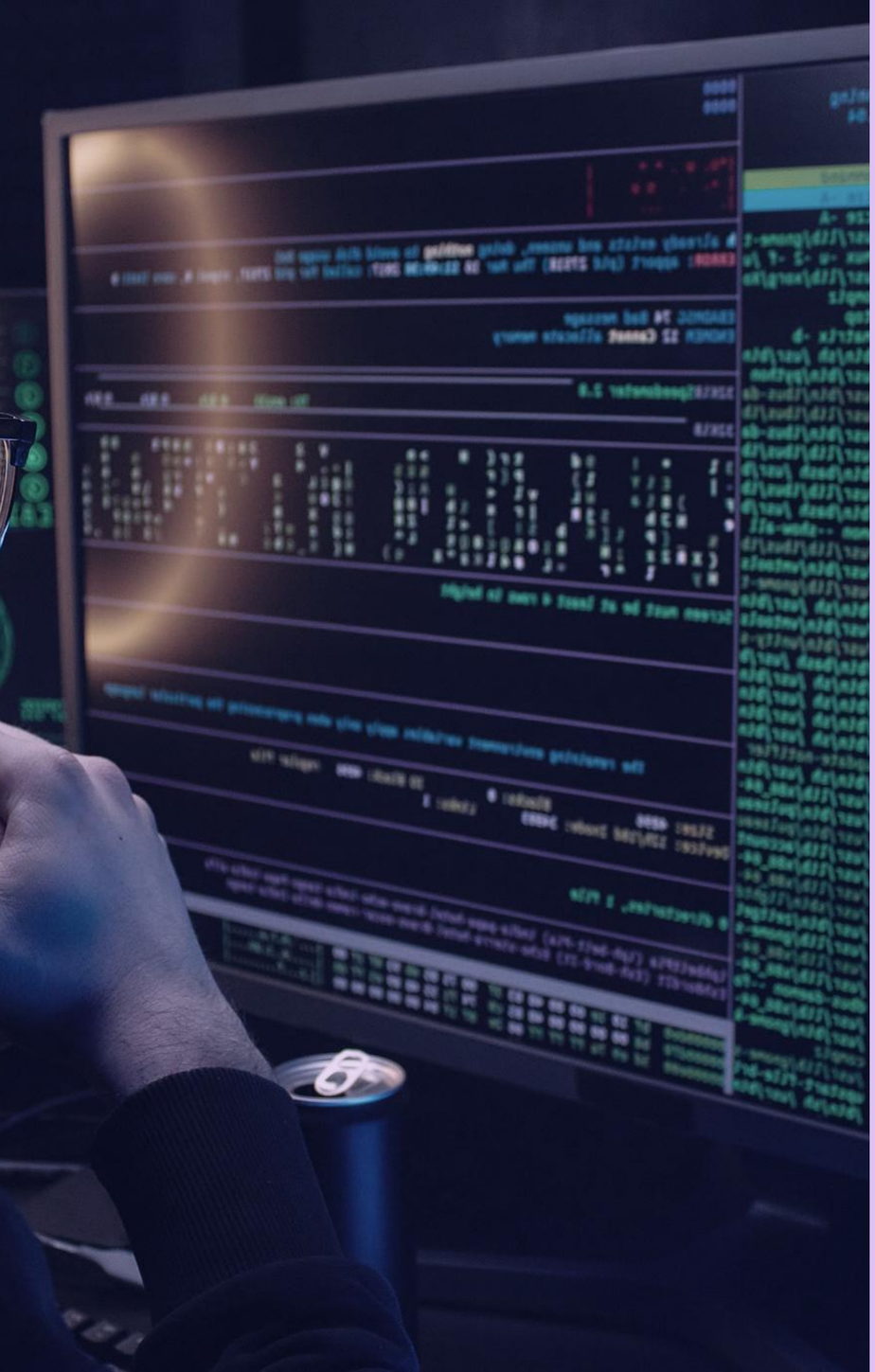




I. Giới Thiệu Vấn Đề

- Phân biệt công ty đó thuộc nhóm nào
- Giúp các nhà đầu tư lựa chọn
- Tìm ra sự tương đồng giữa các công ty
- Đề xuất lời khuyên đầu tư trong một ngành cụ thể
- Giúp nhà đầu tư hiểu rõ hơn về cấu trúc và sự phát triển của thị trường





II. Giải Pháp

- K-Means Clustering
- Hierarchical Clustering
- DBSCAN Clustering

III. Hiện Thực

1. Imports & Data	4. Hierarchical Clustering
2. Exploratory Data Analysis	5. DBSCAN CLustering
3. K-Means Clustering	6. Improvement

1.Import & Data

Import thư viện

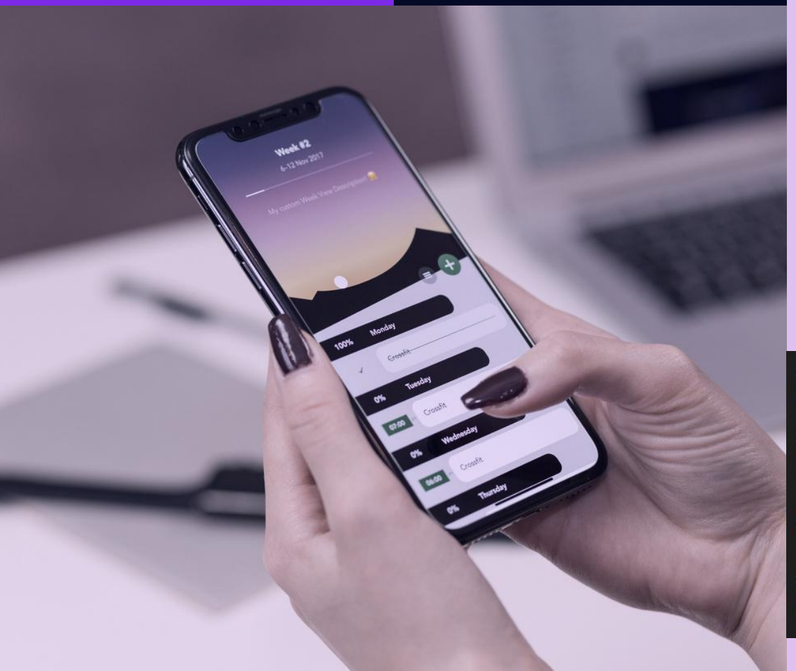
```
import yfinance as yf
```

Định nghĩa tên Công ty

```
'Amazon': 'AMZN',
'Apple': 'AAPL',
```

Lấy dữ liệu

```
data_source = 'yahoo'
start_date = '2017-01-01'
end_date = '2022-01-01'
panel_data = pdr.get_data_yahoo(list(companies_dict.values()),start_date, end_date)
```



U.S. markets closed

S&P 500	Dow 30	Nasdaq	Russell 2000	Crude Oil	Gold
4,137.64	33,886.47	12,123.47	1,781.15	82.68	2,017.70
-8.58 (-0.21%)	-143.22 (-0.42%)	-42.81 (-0.35%)	-15.53 (-0.86%)	+0.52 (+0.63%)	-37.60 (-1.83%)

Netflix, Tesla, Goldman Sachs, Bank of America headline earnings rush in week ahead

First quarter earnings season will be in full flight this week as tech, banks, and Tesla all report results.

Yahoo Finance

Rubenstein on investing: 'I've had a lot of regrets'

Yahoo Finance

Houston, Dallas lead the country in empty office space

Yahoo Finance

Lucid Air Grand Touring: The future of EVs is here

Yahoo Finance

Tesla, Netflix earnings, your taxes: what you need to know

Yahoo Finance Video

IRA options dependent on 'how much time you have on your side' before...

Yahoo Finance Video

Quote Lookup

My Portfolio & Markets Customize

Recently Viewed ›

Symbol	Last Price	Change	% Change
BTC-USD Bitcoin USD	30,345.69	-83.17	-0.27%
AMZN Amazon.com, Inc.	102.51	+0.11	+0.11%
0P0000... AMM Finance SICAV-Amazone Euro Fund	147.45	+0.85	+0.58%
AAPL Apple Inc.	165.21	-0.35	-0.21%

AdChoices

AdRoll

1.Import & Data

```
panel_data.isna().sum()
```

```
Adj Close  AAPL     0
           AMC      0
           AMZN     0
           AXP      0
           BA       0
                   ..
Volume     TM       0
           TXN      0
           VLO      0
           WBA      0
           XOM      0
Length: 162, dtype: int64
```

```
print(stock_close.iloc[0])
```

```
AAPL     29.037500
AMC      34.750000
AMZN     37.683498
AXP      75.349998
BA      156.970001
BAC      22.530001
CVX     117.849998
EVO       4.070000
F        12.590000
GE      190.296036
HMC      29.610001
IBM     159.837479
INTC     36.599998
JNJ     115.839996
```

```
print(stock_open.iloc[0])
```

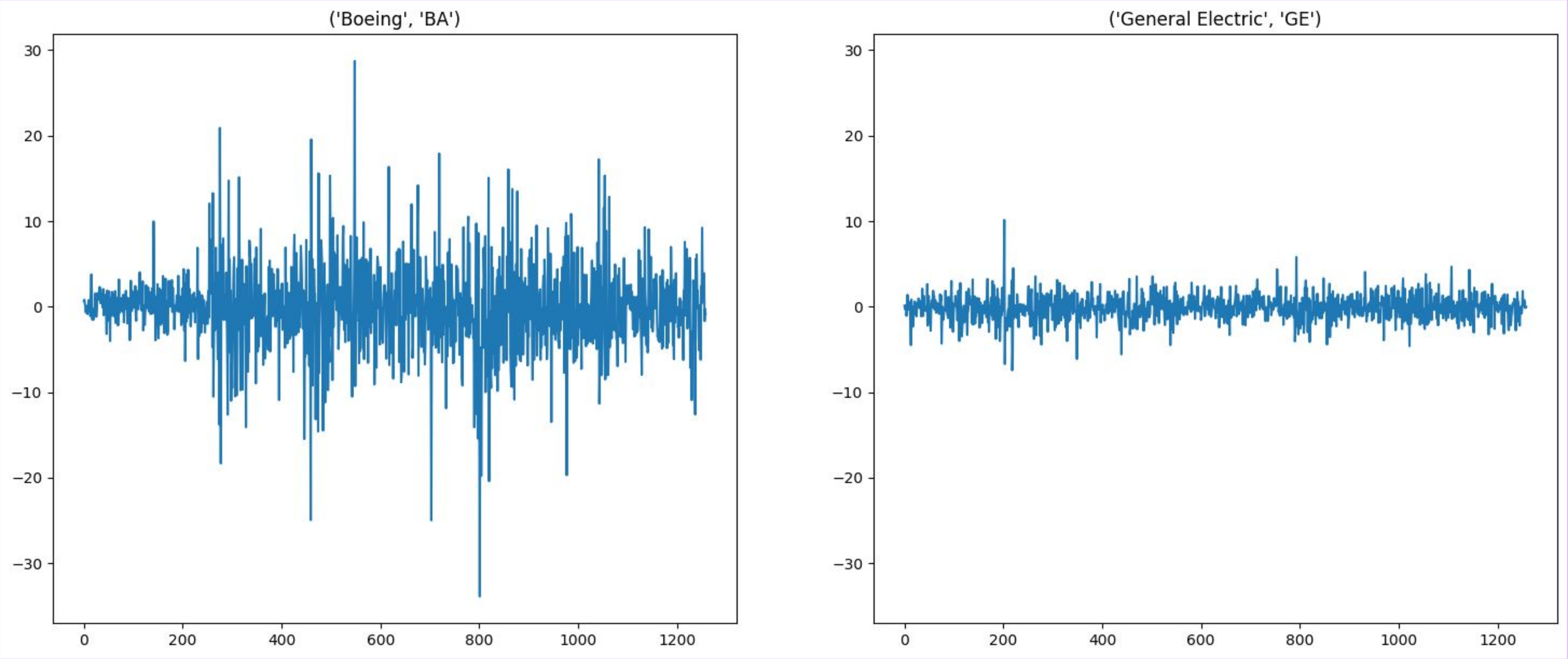
```
AAPL     28.950001
AMC      34.049999
AMZN     37.896000
AXP      74.889999
BA      156.300003
BAC      22.600000
CVX     118.379997
EVO       3.987500
F        12.200000
GE      190.175949
HMC      29.480000
IBM     159.655838
INTC     36.610001
JNJ     115.779999
```

1.Import & Data

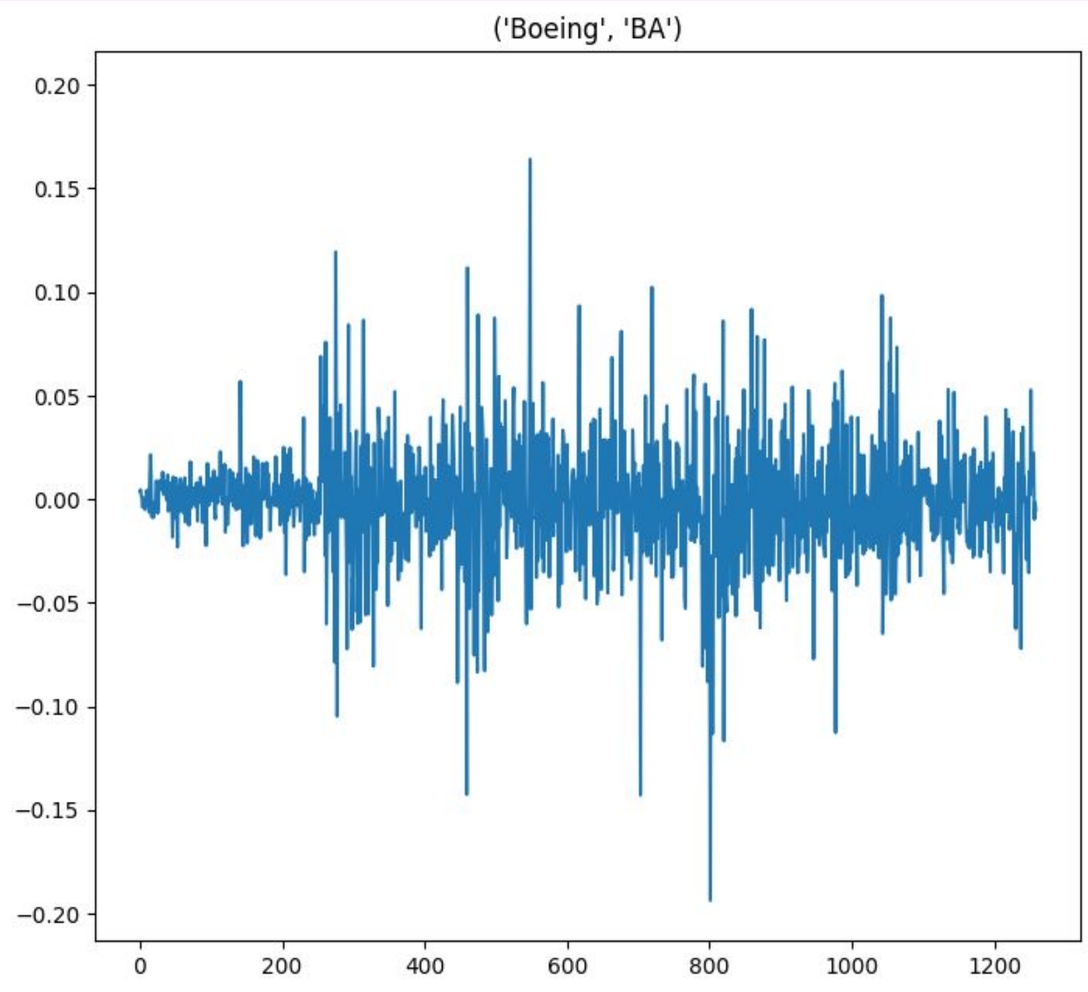
Movement = stock_closed - stock_open

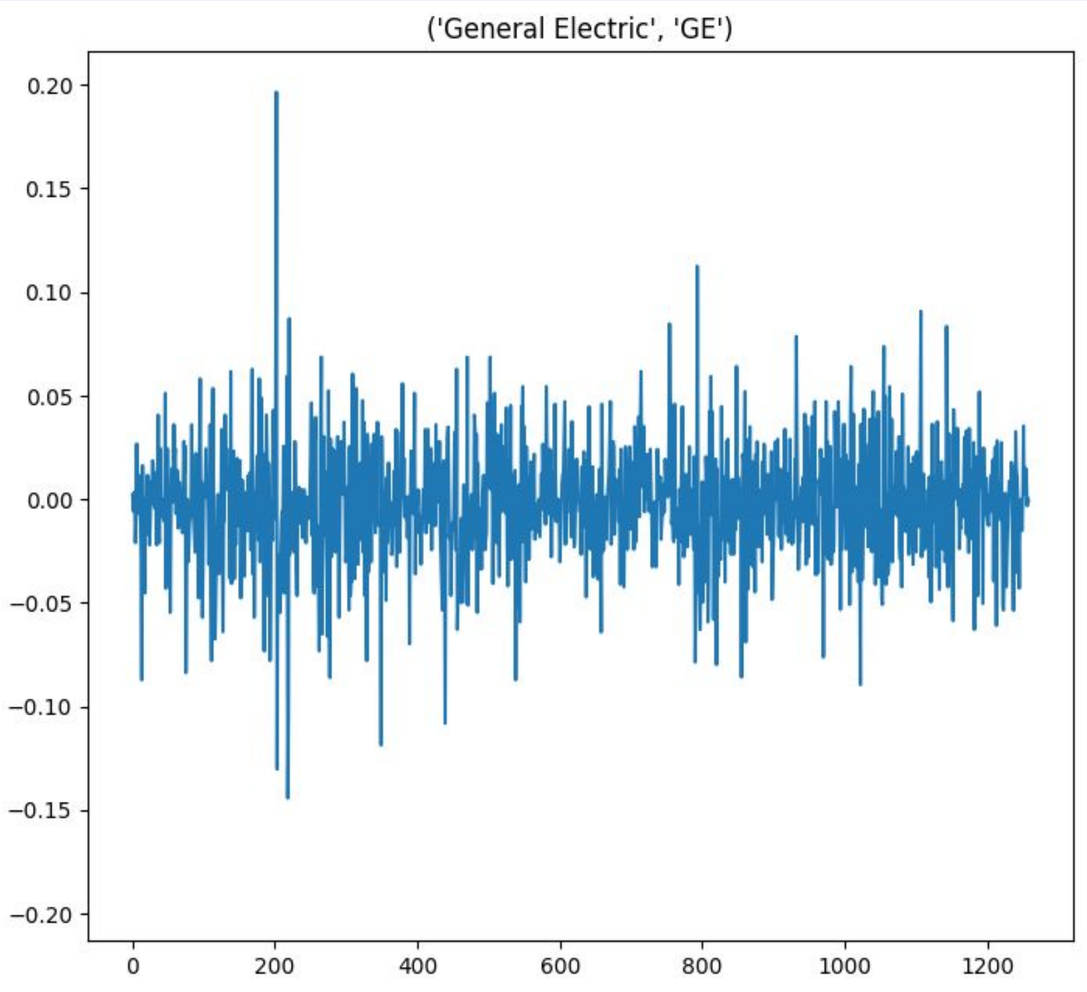
```
Company: Apple, Change: 0.08749961853027344
Company: AMC  Entertainment, Change: 0.7000007629394531
Company: Amazon, Change: -0.21250152587890625
Company: American Express, Change: 0.45999908447265625
Company: Boeing, Change: 0.6699981689453125
Company: Bank of America, Change: -0.06999969482421875
Company: Chevron, Change: -0.529998779296875
Company: Evotec SE, Change: 0.08250021934509277
Company: Ford, Change: 0.3900003433227539
Company: General Electric, Change: 0.120086669921875
Company: Honda, Change: 0.13000106811523438
Company: IBM, Change: 0.181640625
Company: Intel, Change: -0.01000213623046875
Company: Johnson & Johnson, Change: 0.05999755859375
Company: Coca Cola, Change: 0.2999992370605469
Company: Lockheed Martin, Change: 2.1100006103515625
```

2. Exploratory Data Analysis (EDA)



2. Exploratory Data Analysis (EDA)

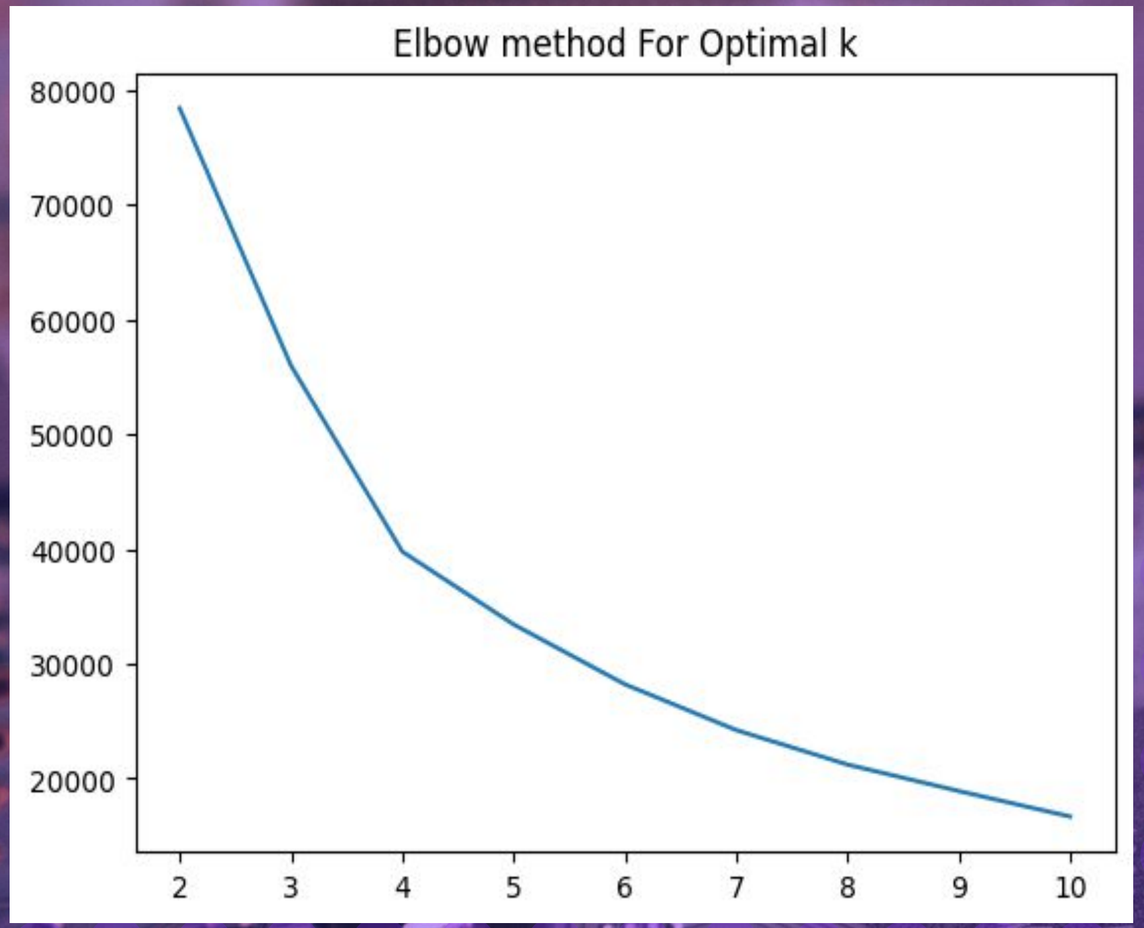


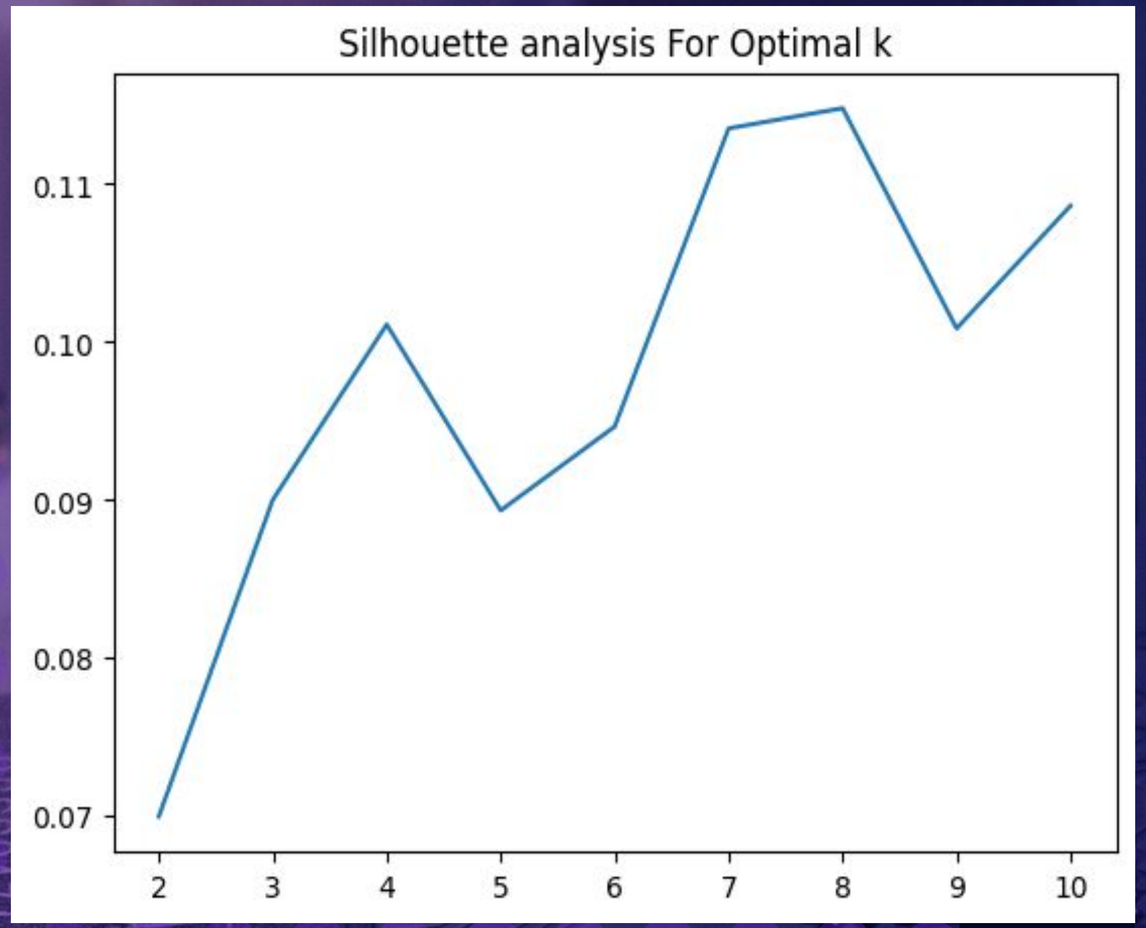


3. Kmeans Clustering

1. Chọn số cụm (k) bạn muốn tạo ra.	4. Phân loại mỗi điểm dữ liệu vào cụm gần nhất với nó.
2. Chọn ngẫu nhiên k điểm từ dữ liệu làm điểm trung tâm của các cụm ban đầu.	5. Tính toán lại trung tâm của mỗi nhóm sau khi phân loại các điểm dữ liệu.
3. Tính khoảng cách của từng điểm dữ liệu đã chọn đến k điểm trung tâm.	6. Lặp lại các bước 3 đến 5 cho đến khi việc phân cụm không thay đổi nữa hoặc đạt được tiêu chí dừng được đặt ra.

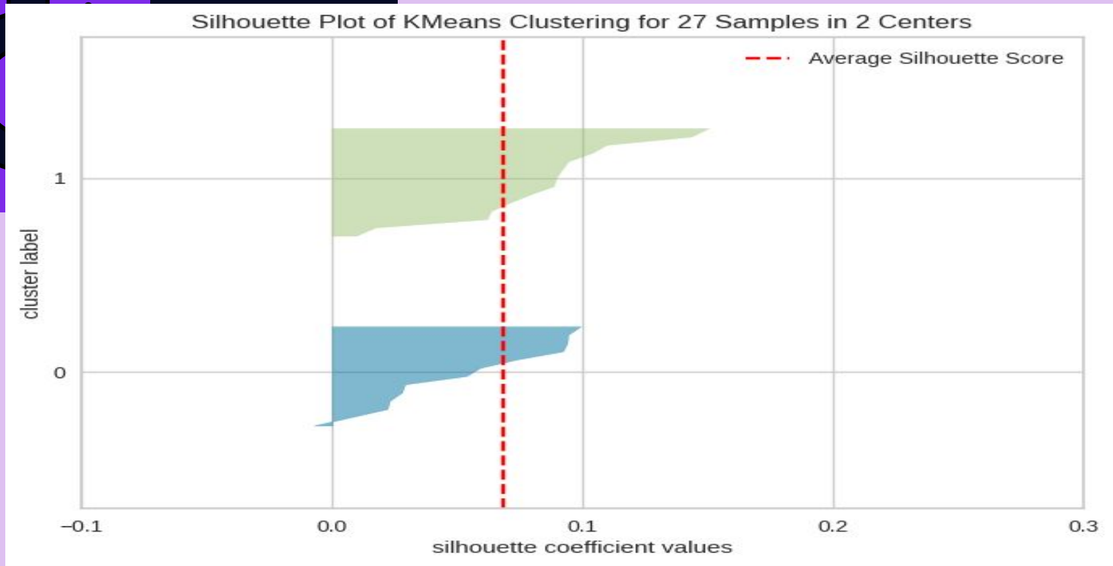
3. Kmeans Clustering

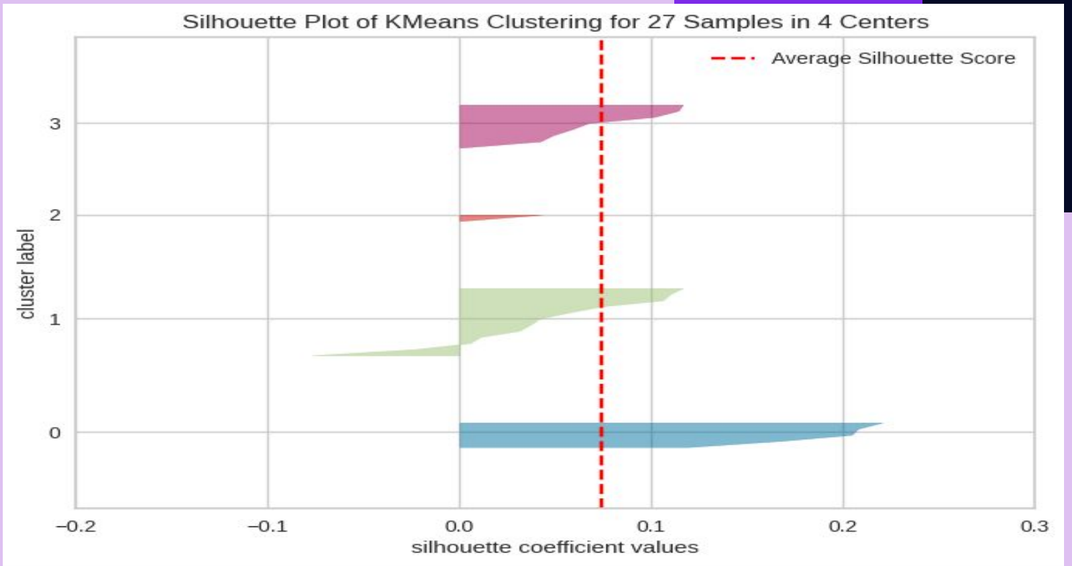


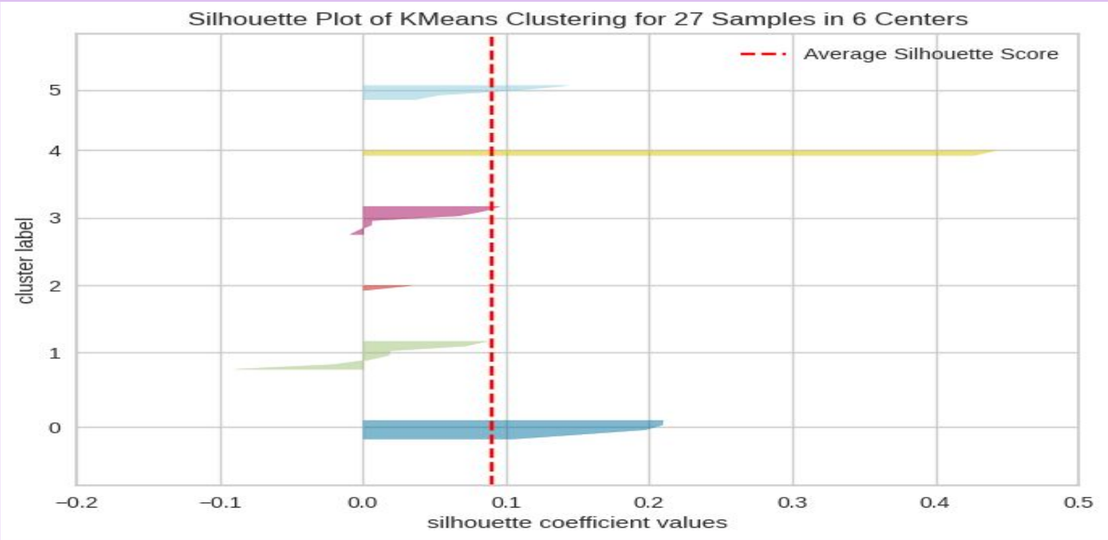


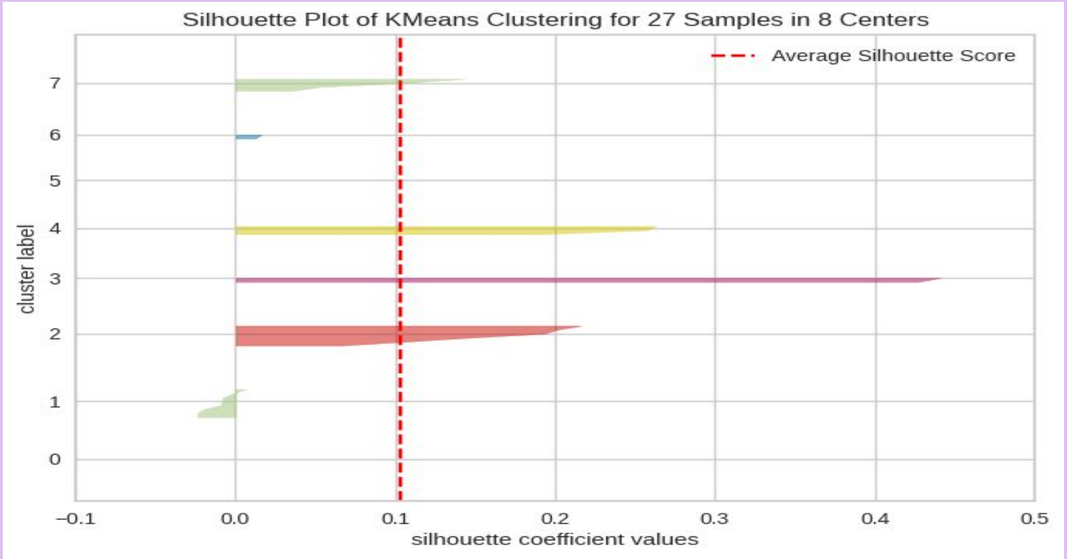
3.Kmeans Clustering

Silhouette Visualizer









3.Kmeans Clustering

Pipeline

```
# import machine learning libraries
from sklearn.pipeline import make_pipeline
from sklearn.cluster import KMeans

# define normalizer
normalizer = Normalizer()

# create a K-means model with 8 clusters
kmeans = KMeans(n_clusters=8 , max_iter=1000)

# make a pipeline chaining normalizer and kmeans
pipeline = make_pipeline(normalizer,kmeans)
```

Silhouette-score

```
from sklearn.metrics import silhouette_score
silhouette_score(new,labels)
```

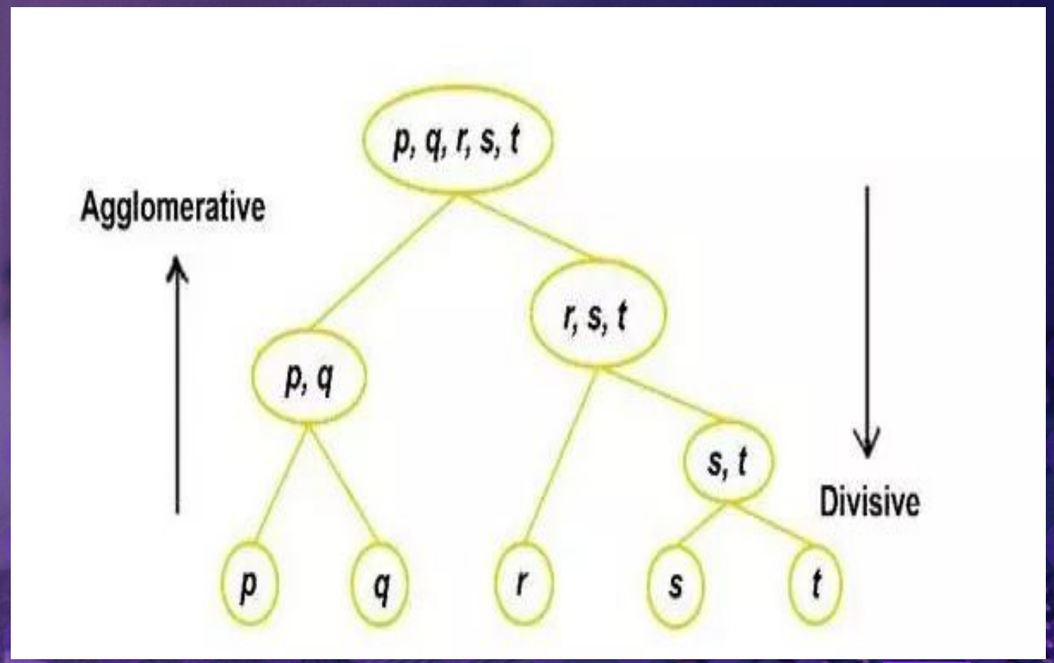
```
0.09501040904415513
```

Predict

	labels	companies
17	0	(McDonalds, MCD)
26	1	(Exxon, XOM)
24	1	(Valero Energy, VLO)
5	1	(Bank of America, BAC)
6	1	(Chevron, CVX)
9	1	(General Electric, GE)
23	2	(Texas Instruments, TXN)
18	2	(Microsoft, MSFT)
16	2	(MasterCard, MA)
12	2	(Intel, INTC)
0	2	(Apple, AAPL)
2	2	(Amazon, AMZN)
21	3	(Plug Power, PLUG)
1	3	(AMC Entertainment, AMC)
15	4	(Lockheed Martin, LMT)
19	4	(Northrop Grumman, NOC)
14	5	(Coca Cola, KO)
20	5	(Pepsi, PEP)
13	5	(Johnson & Johnson, JNJ)
7	6	(Evotec SE, EVO)
11	7	(IBM, IBM)
8	7	(Ford, F)
25	7	(Walgreen, WBA)
4	7	(Boeing, BA)
3	7	(American Express, AXP)
22	7	(Toyota, TM)
10	7	(Honda, HMC)

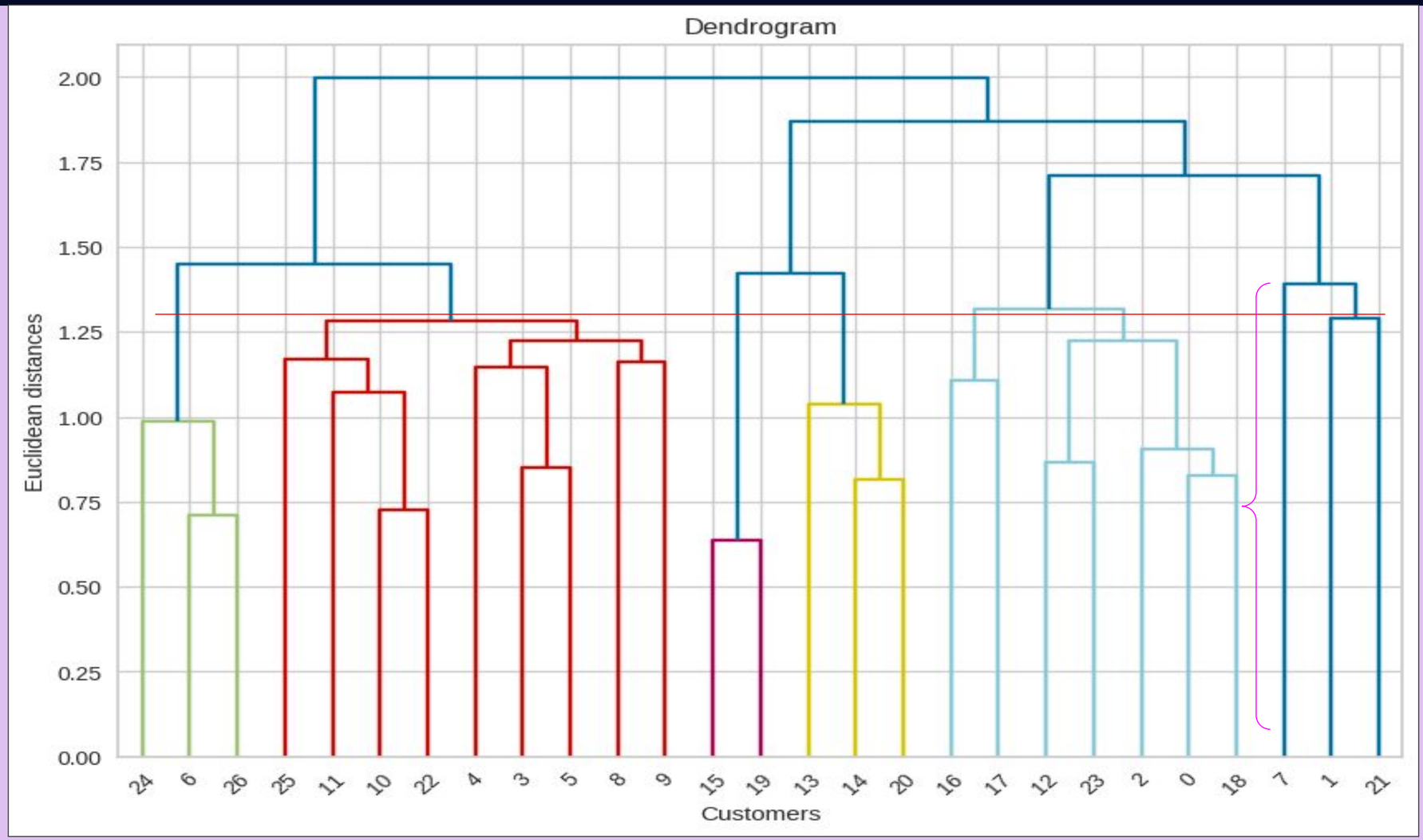
4.Hierarchical Clustering

Phương pháp này xây dựng một cây phân cấp, trong đó mỗi mức độ thể hiện một cụm tăng dần độ hoàn chỉnh. Cụm được xây dựng trên cơ sở khoảng cách giữa các điểm dữ liệu. Các phương pháp đo khoảng cách thường được sử dụng như Euclidean, Manhattan, Mahalanobis, Jaccard, Cosine, và nhiều hơn nữa. Hierarchical clustering có thể được sử dụng trong các lĩnh vực như khai phá dữ liệu, thông tin sinh học, và thị trường tài chính.



4.Hierarchical Clustering

Dendogram



4.Hierarchical Clustering

Predict

```
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=7, metric='euclidean', linkage='ward')
cluster.fit_predict(new)
```

```
array([0, 1, 0, 2, 2, 2, 4, 6, 2, 2, 2, 2, 0, 3, 3, 5, 0, 0, 0, 5, 3, 1,
       2, 0, 4, 2, 4])
```

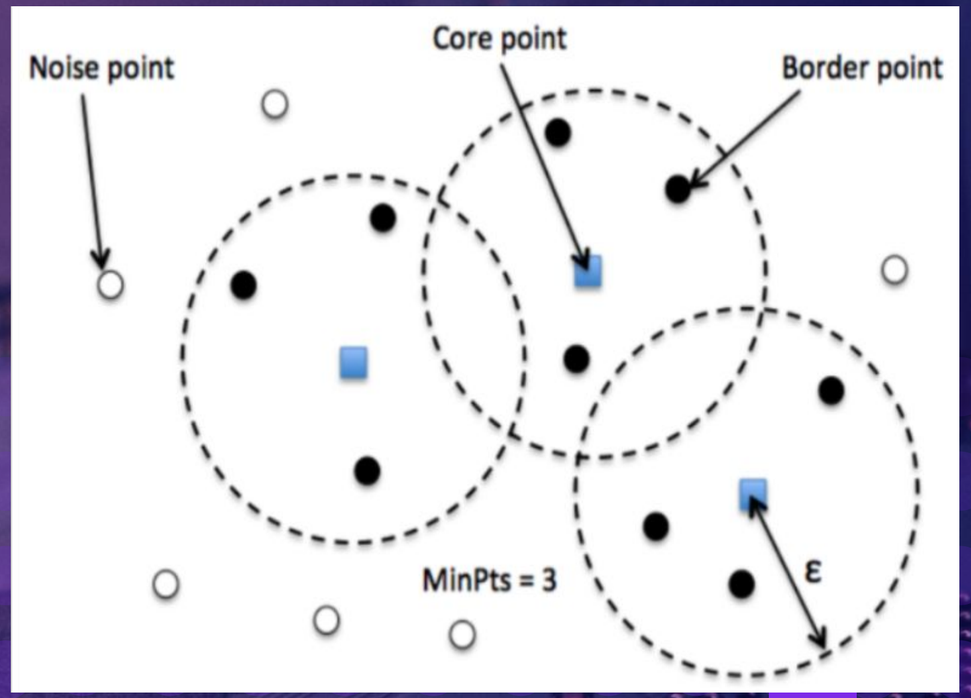
Silhouette-score

```
[61] silhouette_score(new,cl)
```

```
0.11150985480093355
```

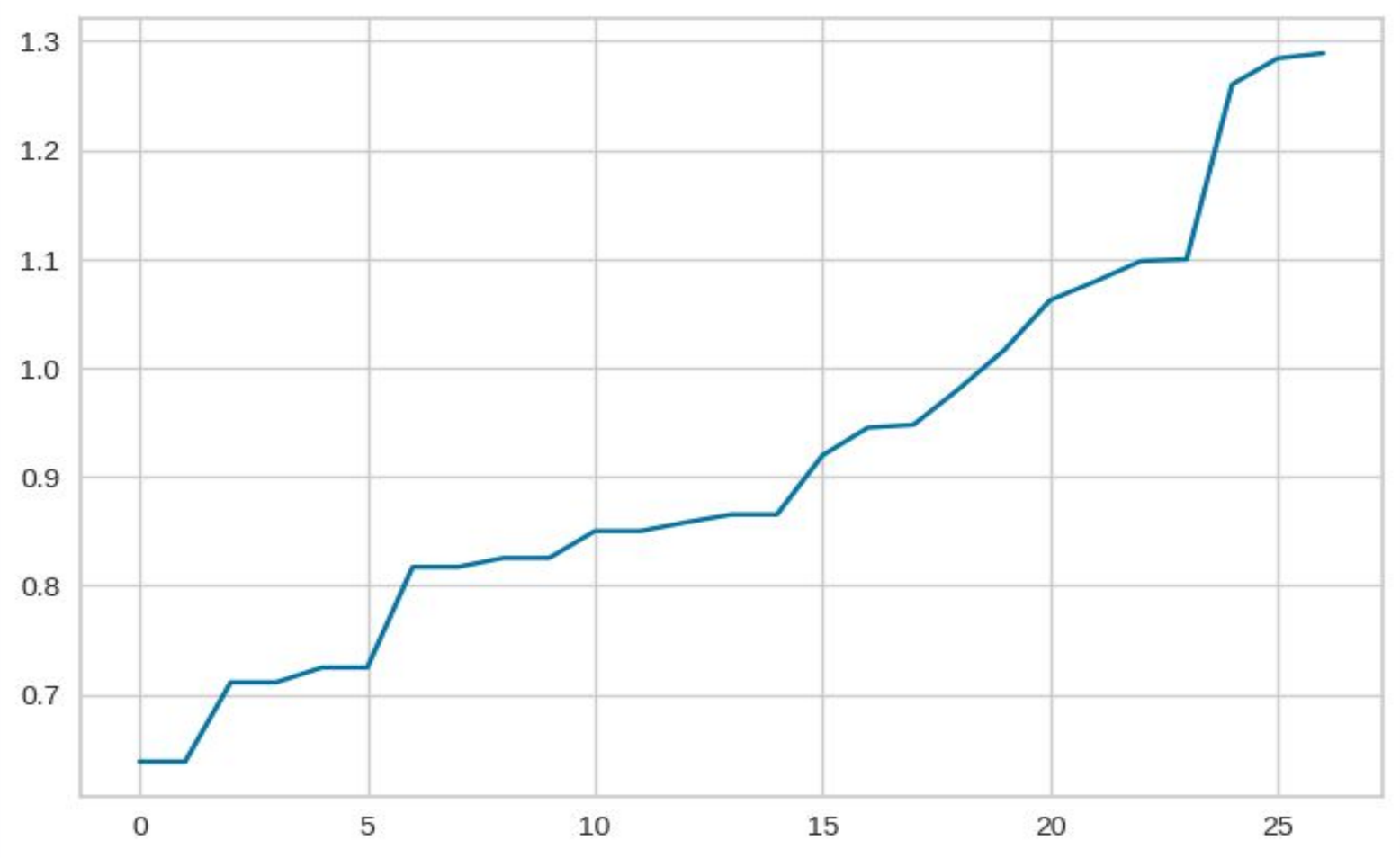
5. DBSCAN CLustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là một thuật toán phân cụm trong Machine Learning được sử dụng để phân loại các điểm dữ liệu thành các cụm dựa trên mật độ của chúng. Thuật toán này xác định các điểm dữ liệu gần nhau bằng cách tính khoảng cách giữa chúng. Các điểm có khoảng cách nhỏ hơn một ngưỡng được xem là thuộc cùng một cụm, và các điểm còn lại được xem là nhiễu. Khác với các thuật toán phân cụm khác, DBSCAN không cần biết số lượng và kích thước của các cụm trước khi chạy thuật toán. Nó được sử dụng rộng rãi trong các ứng dụng khoa học dữ liệu, như phân tích ảnh, nhận dạng giọng nói, phân loại tài liệu và phân tích cảm xúc.



5.DBSCAN CLustering

min_distance



5.DBSCAN Clustering

method

```
from sklearn.cluster import DBSCAN
dbscancluster = DBSCAN(eps=1.2, min_samples=2)
dbscancluster.fit(new)
clusters = dbscancluster.labels_
```

```
len(set(clusters))
```

```
2
```

Silhouette-score

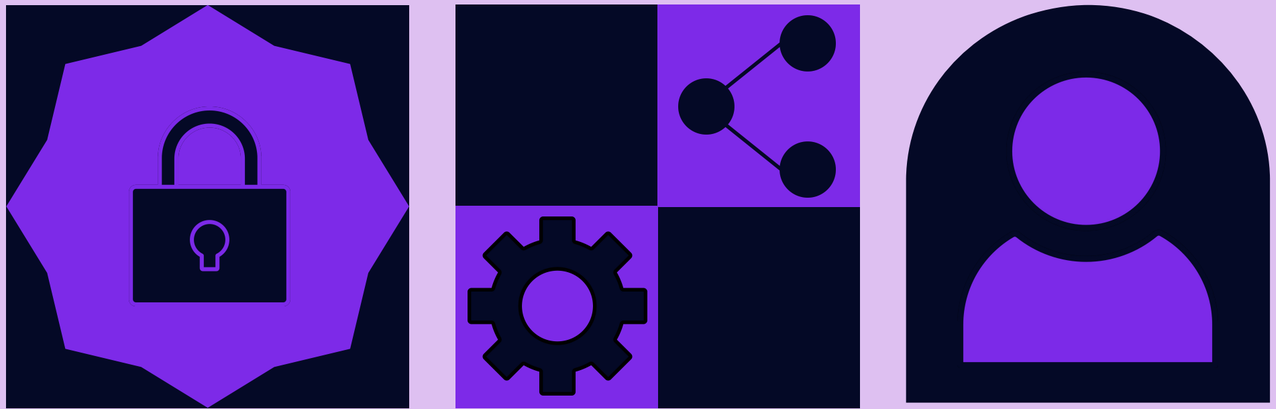
```
silhouette_score(new,clusters)
```

```
0.12412106415253238
```

6.Improvement

Kmeans ++

Principal Component Analysis



6.Improvement

k-means++

```
# import machine learning libraries
from sklearn.pipeline import make_pipeline
from sklearn.cluster import KMeans

# define normalizer
normalizer = Normalizer()

# create a K-means model with 10 clusters
kmeans = KMeans(n_clusters=8 , init = 'k-means++', max_iter=1000)

# make a pipeline chaining normalizer and kmeans
pipeline = make_pipeline(normalizer,kmeans)
```

Silhouette-score

```
from sklearn.metrics import silhouette_score
silhouette_score(new,labels)
```

```
0.11347233626591217
```

Principal Component Analysis
Principal Component Analysis
Principal Component Analysis

Principal Component Analysis(PCA)

PCA (Principal Component Analysis) trong clustering là một phương pháp được sử dụng để giảm số chiều dữ liệu trước khi thực hiện các thuật toán clustering. Phương pháp này cho phép lựa chọn các đặc trưng quan trọng nhất của dữ liệu trong quá trình xử lý và phân tích.

6.Improvement

PCA for kmeans

```
# PCA
from sklearn.decomposition import PCA

# visualize the results
reduced_data = PCA(n_components = 2).fit_transform(new)

# run kmeans on reduced data
kmeans = KMeans(n_clusters=8, init = 'k-means++',max_iter=1000)
kmeans.fit(reduced_data)
labels = kmeans.predict(reduced_data)

# create DataFrame aligning labels & companies
df = pd.DataFrame({'labels': labels, 'companies': companies})

# Display df sorted by cluster labels
print(df.sort_values('labels'))
```

Silhouette-score

```
silhouette_score(reduced_data,labels)
```

```
0.4510321845523287
```

```
new.shape
```

```
(27, 1259)
```

```
reduced_data.shape
```

```
(27, 2)
```

6.Improvement

BEFORE PCA

Kmeans++

```
from sklearn.metrics import silhouette_score
silhouette_score(new,labels)
```

```
0.11347233626591217
```

Hierarchical

```
[61]  silhouette_score(new,cl)
```

```
0.11150985480093355
```

DBSCAN

```
silhouette_score(new,clusters)
```

```
0.12412106415253238
```

AFTER PCA

Kmeans++

```
silhouette_score(reduced_data,labels)
```

```
0.4510321845523287
```

Hierarchical

```
silhouette_score(reduced_data,cl)
```

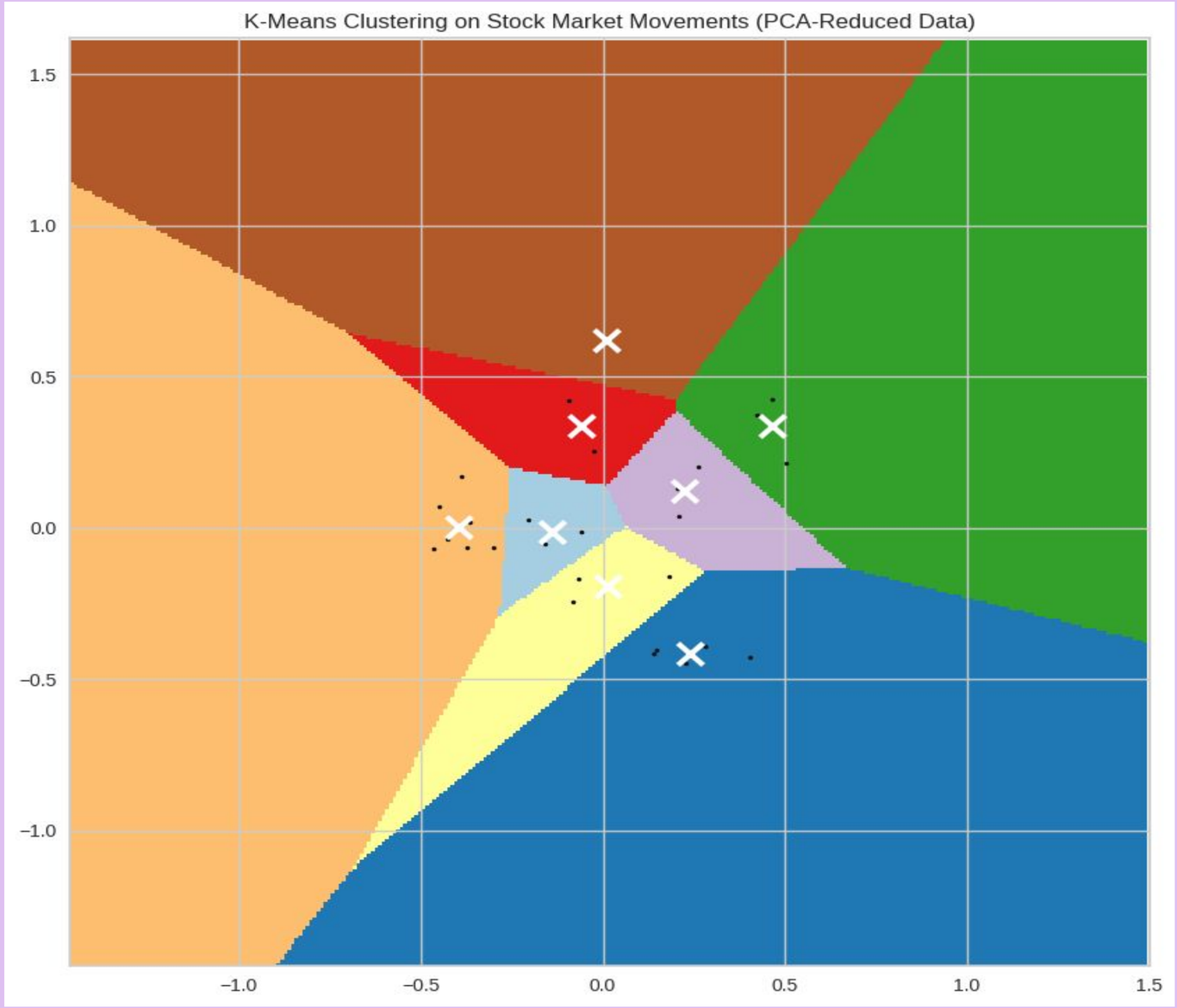
```
0.46935909196649206
```

DBSCAN

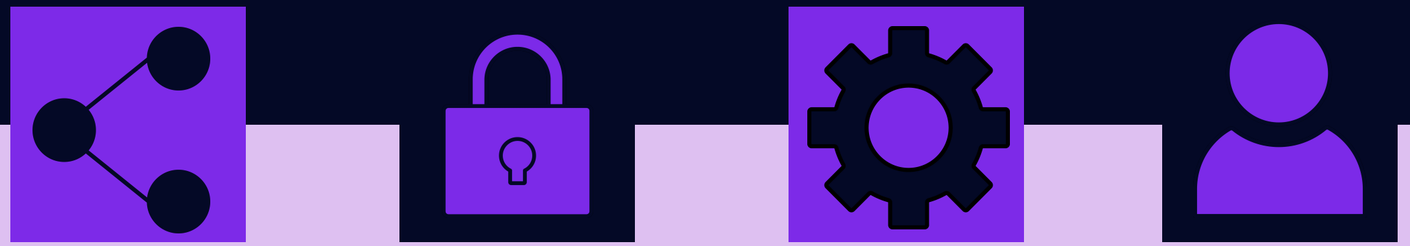
```
silhouette_score(reduced_data,clusters)
```

```
0.49923663304081756
```

Kmeans Visualizer



Thank you!



Cảm ơn thầy và các bạn

đã quan tâm theo dõi