Đề Tài 3: Phân tích dữ liệu cổ phiếu VN theo ngày với Clustering Algorithms, Principal Component Analysis và Gaussian Mixture model.

I. Mở Đầu

Đề xuất một mô hình dự đoán Clustering Algorithms,Principal Component Analysis và Gaussian Mixture model với 2 yêu cầu mà đề tài 3 đề ra như sau:

- Dùng PCA (k-pca) và các thuật toán phân cụm để tìm mối liên hệ giữa các cổ phiếu, tìm cổ phiếu bất thường. Làm các phân tích này theo từng giai đoạn.
- Dùng Gaussian Mixture model (k models) cho return của đường VN-index, dùng KL để tìm tìm trong rổ cổ phiếu VN-index ra k cổ phiếu có thể dùng để tạo ra một danh mục hòng xấp xỉ vn-index.

Mô hình đề xuất được áp dụng dự đoán cho một số mã cổ phiếu của thị trường chứng khoán Việt Nam. Các thông số đánh giá kết quả thực nghiệm sẽ được giới thiệu trong bài báo cáo.

II. Principal Component Analysis(PCA) và K-means Clustering Algorithms

1. Clustering Algorithms
   K-means clustering là thuật toán cơ bản nhất trong Unsupervised Learning. Mục đích của thuật toán để phân dữ liệu thành cụm khác nhau sao cho dữ liệu trong cùng một cụm thể hiện tính chất giống nhau.

   - Ký hiệu toán học:
     Giả sử có **N** điểm dữ liệu là $X = [x_1, x_2, ..., x_n] \epsilon \mathbb{R}^{dx\mathbf{N}}$ và $K < N$ là số cluster chúng ta muốn phân chia. Chúng ta cần tìm các center $m_1, m_2, ..., m_k \epsilon \mathbb{R}^{dx1}$ và label của mỗi điểm dữ liệu.
     Với mỗi điểm dữ liệu x_i đặt $y_i = [y_{i1}, y_{i2}, ..., y_{iK}]$ là label vector của nó, trong đó nếu x_i được phân vào cluster k thì $y_{ik} = 1$ và $y_{ij} = 0, \forall j \neq k$. Điều này có nghĩa là có đúng một phần tử của vector y_i là bằng 1 (tương ứng với cluster của x_i), các phần tử còn lại bằng 0. Ví dụ: nếu một điểm dữ liệu có label vector là $[1, 0, 0, ..., 0]$ thì nó thuộc vào cluster 1, là $[0, 1, 0, ..., 0]$ thì nó thuộc vào cluster 2, ...
     Ràng buộc của y_i có thể viết dưới dạng như sau:

     $$y_i k \epsilon \{0, 1\}, \sum_{k=1}^{K} y_{ik} = 1$$

   - Hàm mất mát và bài toán tối ưu:
     Nếu ta coi center m_k là center (hoặc representative) của mỗi cluster và ước lượng tất cả các điểm đường phân vào cluster này bởi m_k, thì một điểm dữ liệu x_i được phân vào cluster k sẽ bị sai số là $(x_i - m_k)$. Chúng ta mong muốn sai số này có trị tuyệt đối nhỏ nhất nên ta sẽ tìm cách đại lượng sau đây đạt giá trị nhỏ nhất:

     $$||X_i - m_k||_2^2$$

     Hơn nữa, vì x_i được phân vào cluster k nên $y_{ik} = 1, y_{ij} = 0, \forall j \neq k$. Khi đó, biểu thức bên trên sẽ được viết lại là:

     $$y_{ik}||x_i - m_k||_2^2 = \sum_{j=1}^{K} y_{ij}||x_i - m_j||_2^2$$

     Sai số cho toàn bộ dữ liệu sẽ là:

     $$\mathcal{L}(Y, M) = \sum_{i=1}^{N} \sum_{j=1}^{K} y_{ij}||x_i - m_j||_2^2$$

     Trong đó $Y = [y_1; y_2; ...; y_N]$,$M = [m_1, m_2, ..., m_k]$ lần lượt là các ma trận được tạo bởi label vector của mỗi điểm dữ liệu và center của mỗi cluster. Hàm số mất mát trong bài toán K-means clustering của chúng ta là hàm $\mathcal{L}(Y, M)$ với ràng buộc như được nêu trong phương trình.
     Tóm lại, chúng ta cần tối ưu bài toán sau:

     $$Y, M = argmin_{Y,M} \sum_{i=1}^{N} \sum_{j=1}^{K} y_{ij}||x_i - m_j||_2^2$$

     $$\text{subject to: } y_{ij} \epsilon \{0, 1\} \forall i, j; \sum_{j=1}^{K} y_{ij} = 1 \forall i$$

   - Tối ưu Hàm Mất mát:

- *Giả sử đã tìm được các centers, hãy tìm các label vector để hàm mất mát đạt giá trị nhỏ nhất. (Cố định M, tìm Y)* Bài toán tìm Label vector cho từng điểm dữ liệu X_i như sau:

$$y_i = argmin_{yi} \sum_{j=1}^{K} y_{ij} ||x_i - m_j||_2^2$$

$$Subjectto : y_{ij} \epsilon 0, 1 \forall j; \sum_{j=1}^{K} y_{ij} = 1$$

Vì chỉ có một phần tử của label vector y_i bằng 1 nên bài toàn có thể tiếp tục được viết:

$$j = argmin_j y_{ij} ||x_i - m_j||_2^2$$

Vì $||x_i - m_j||_2^2$ là bình phương khoảng cách tính từ điểm x_i tới center m_j,vậy mỗi điểm x_i thuộc vào cụm có gần nó nhất.

- *Giả sử đã tìm được các cluster cho từng điểm, hãy tìm các center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất. (Cố định M, tìm Y)* Bài toán tìm center cho mỗi cluster :

$$m_j = argmin_{mj} \sum_{i=1}^{n} y_{ij} ||x_i - m_j||_2^2$$

Tìm nghiệm bằng phương pháp giải đạo hàm bằng 0, hàm cần tối ưu là hàm liên tục và có đạo hàm xác định tại mọi điểm. Tìm được giá trị nhỏ nhất và điểm tối ưu tương ứng vì hàm lồi theo m_j
Đặt $l(m_j)$ là hàm bên trong dấu $argmin$, ta có đạo hàm:

$$\frac{\partial l(m_j)}{\partial m_j} = 2 \sum_{i=1}^{N} y_{ij} (m_i - x_i||$$

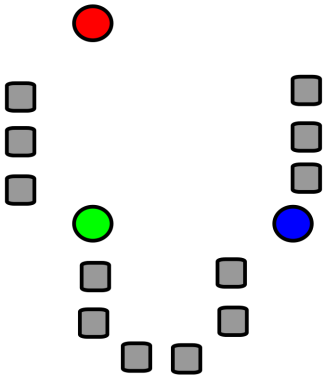
Giải phương trình đạo hàm bằng 0 ta có:

$$m_j \sum_{i=1}^{N} y_{ij} = \sum_{i=1}^{N} y_{ij} x_i$$

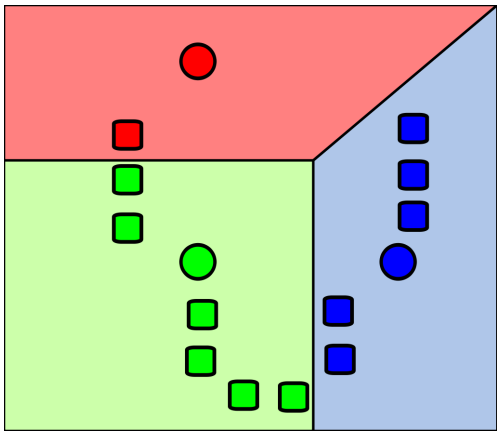
$$\Rightarrow m_j = \frac{\sum_{i=1}^{N} y_{ij} x_i}{\sum_{i=1}^{N} y_{ij}}$$

Nhận thấy m_j là trung bình cộng của các điểm trong cluster j

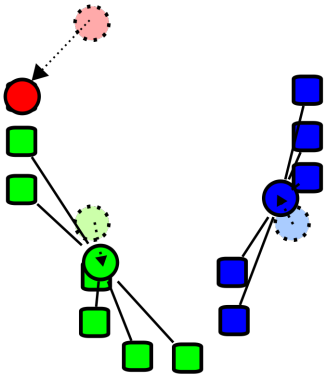
- Thuật Toán:
  - B1: Xác định K ngẫu nhiên
  - B2: Tìm những giá trị gần nhất với giá trị K ngẫu nhiên ban đầu.
  - B3: Nếu giá trị không cò gì thay đổi thì ta dừng ở B2.
  - B4: Bắt đầu tính trung bình thêm lần nữa bằng cách lấy trung bình cộng của tất cả các điểm đã gán ở B2.
  - B5: Quay lại bước 2
  - KQ: Đạt được đến khi hội tụ



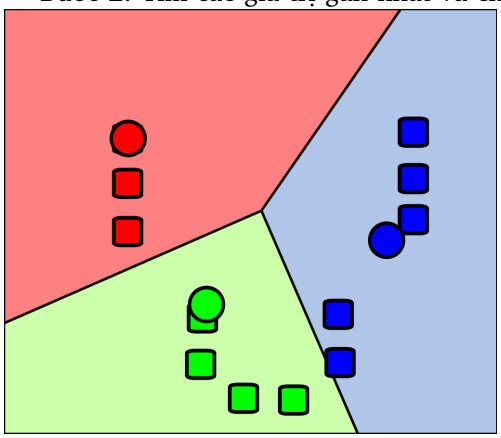
Bước 1: Tạo K ngẫu nhiên (giả sử K = 3)



Bước 2: Tìm các giá trị gần nhất và chia cụm



Bước 3: Tâm cụm K thành giá trị trung bình mới



Bước 4 +5 : Đạt đến khi hội tụ.

2. Principal Component Analysis(PCA)

- Việc giảm chiều dữ liệu bằng cách dùng PCA với cách đơn giản nhất là từ D vè $K < D$ và chỉ giữ lại K là phần tử quan trọng nhất. Tuy nhiên, lượng thông tin mà mỗi thành phần mang như nhau, bỏ đi thành phần nào cũng dẫn đến việc mất một lượng thông tin lớn khác. Biểu diễn các vector dữ liệu ban đầu trong hệ cơ sở mới mà trong hệ cơ sở mới đó, tầm quan trọng giữa các thành phần là khác nhau rõ rệt thì có thể bỏ qua thành phần ít quan trọng.
  PCA chính là phương pháp tìm một hệ cơ sở mới sao cho thông tin dữ liệu chủ yếu tập rung ở một vài toạ độ, phần còn lại chỉ mang một lượng nhỏ thông tin. Và để cho đơn giản trong tính toán, PCA sẽ tìm một hệ trực chuẩn để làm cơ sở mới.
  Gọi Σ là ma trận hiệp phương sai của X. Tiếp đến, đặt $(\lambda_1, v_1), ..., (\lambda_p, v_p)$ là các cặp vecto riêng giá trị riêng đã sắp xếp của Σ sao cho $\lambda_1 \lambda_2 \dots \lambda_p \dots 0$. Gỉa sử rằng chúng tôi chọn r cặp đầu tiên để giảm kích thước. Sau đó, lượng phương sai được giải thích bởi các cặp r này là:

$$\frac{\lambda_1 + \lambda_2 + \ldots + \lambda_r}{\lambda_1 + \lambda_2 + \ldots + \lambda_p}$$

  Ngoài ra, đặt $V = [v_1, v_2, \ldots, v_r]$. Sau đó phiên bản giảm kích thước của X là XV
- Thuật Toán PCA:
  - B1: Tính vector kỳ vọng của toàn bộ dữ liệu:

$$\overline{x} = \frac{1}{N} \sum_{n=1}^{N} X_n$$

  - B2: Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:

$$\widehat{x}_n = x_n - \overline{x}$$

  - B3: Tính ma trận hiệp phương sai:

$$S = \frac{1}{N} \widehat{X} \widehat{X}^T$$

  - B4: Tính các trị riêng và vector riêng có norm bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.

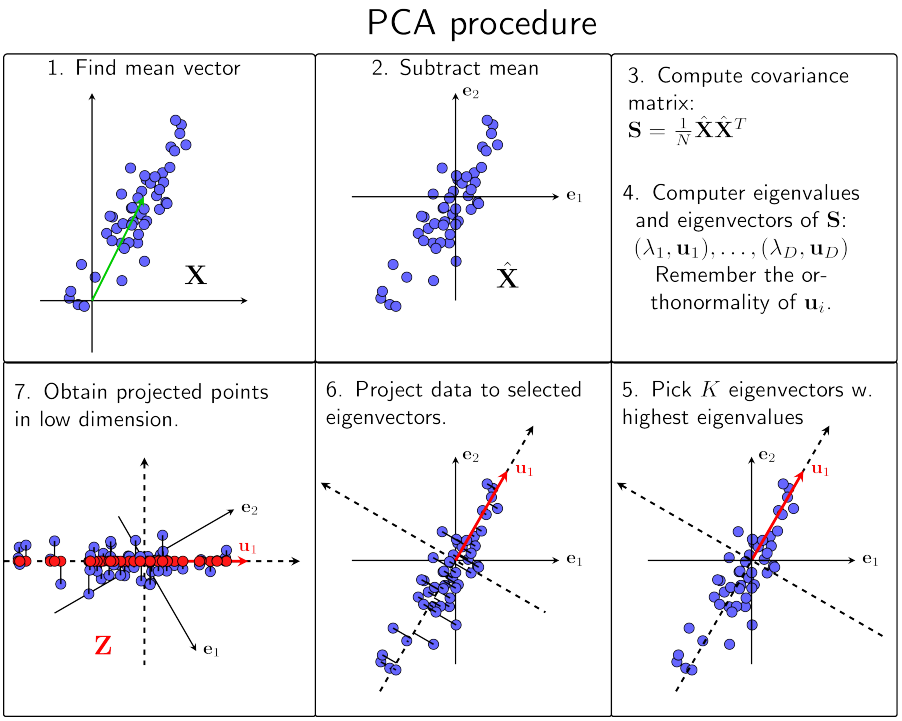
- B5: Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trận U_k có các cột tạo thành một hệ trực giao. K vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hoá.
- B6: Chiếu dữ liệu ban đầu đã chuẩn hóa $\widehat{X}$ xuống không gian con tìm được.
- Dữ liệu mới chính là tọa độ của các điểm dữ liệu trên không gian mới.

$$Z = U_K^T \widehat{X}$$

Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới như sau:

$$x \approx U_K Z + \overline{x}$$

Các bước thực hiện PCA có thể được xem trong hình dưới đây:



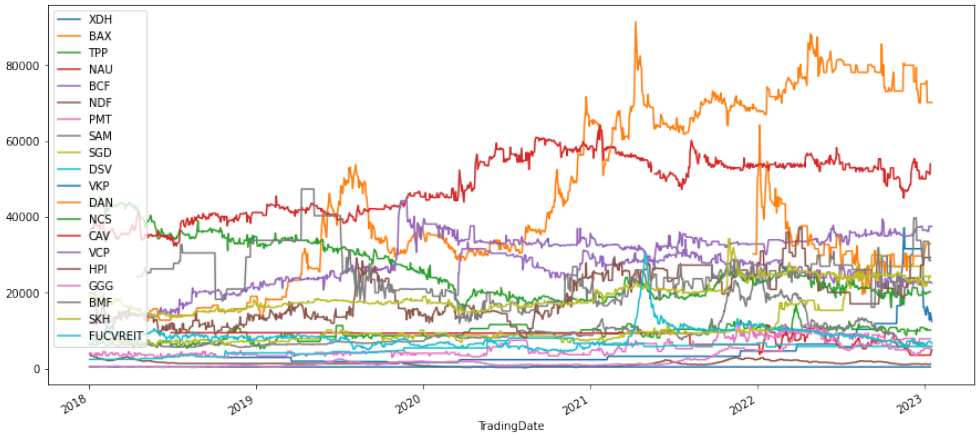
3. Áp Dụng Vào Đề Bài:

- Đặt Vấn Đề: Có rất nhiều mã chứng khoán khác nhân, do quá nhiều yếu tố tác động đến giá hàng hóa và giá dịch vụ nói chung, chỉ số chứng khoán và giá cổ phiếu nói riêng nê có một thời gian dài người ta cho rằng không thể tìm được cổ phiếu bất thường hay phân tích được chúng.
  Hiện nay, có nhiều kỹ thuật ứng dụng trong đó mạnh mẽ nhất là kỹ thuật thống kê và kỹ thuật trí tuệ nhân tạo để chuẩn đoán thị trường chứng khoán.
  Trong bài báo cáo này, ta sử dụng kỹ thuật giảm chiều dữ liệu và kỹ thuật phân cụm để xác định các thuộc tính. Rất nhiều kỹ thuật giảm chiều mà ta từng học như ở bài này sẽ sử dụng *Principal Component Analysis (PCA)* và *Clusterring Algorithms* để phân cụm các cổ phiếu và xác định đánh giá.
  Data : `!pip install vnstock`

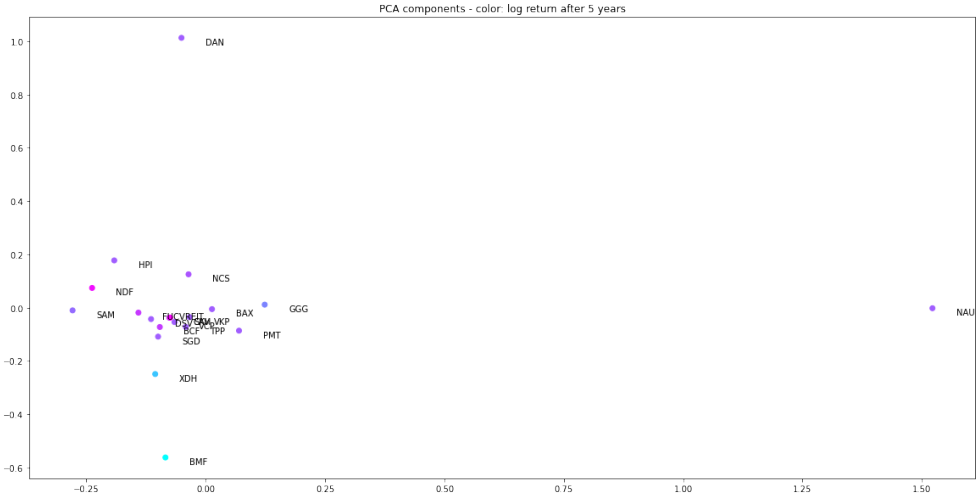
```
Shape: (1631, 4)
  ticker  group_code                                    company_name  \
0    VVS  UpcomIndex  Công ty Cổ phần Đầu tư Phát triển Máy Việt Nam
1    XDC  UpcomIndex   Công ty TNHH MTV Xây dựng Công trình Tân Cảng
2    HSV  UpcomIndex                Công ty Cổ phần Gang Thép Hà Nội
3    CST  UpcomIndex              Công ty Cổ phần Than Cao Sơn - TKV
4    BVL  UpcomIndex                          Công ty Cổ phần BV Land

              company_short_name
0  Đầu tư Phát triển Máy Việt Nam
1    Xây dựng Công trình Tân Cảng
2                Gang Thép Hà Nội
3              Than Cao Sơn - TKV
4                        BV Land
```

  Ngày bắt đầu: `start = "2018-01-01"`

- Giảm Chiều Dữ Liệu và Phân Cụm:
  Ta nhìn tổng quan về dữ liệu để thấy các cổ phiếu đang dao động trên thị trường. Lấy mẫu 20 mã cổ phiếu để visualizing.

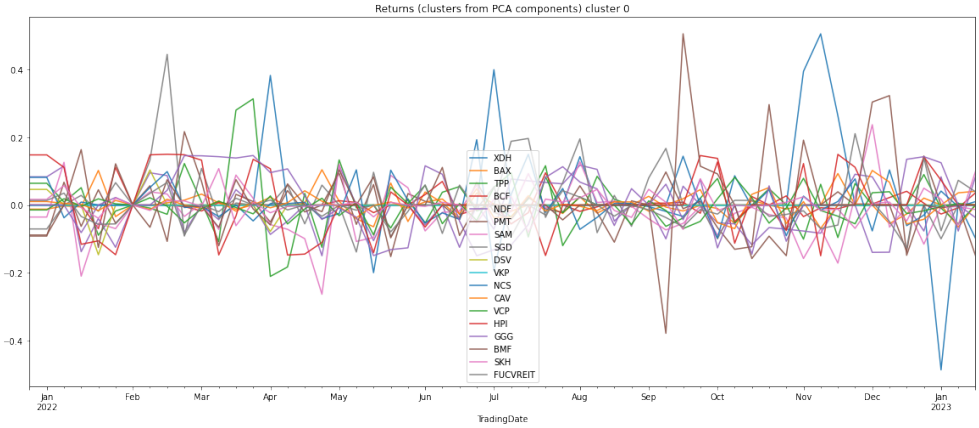


Lúc đầu, Data lúc đầu có số chiều là 100, mỗi chiều thể hiện giá trị *closed* trong ngày và ta dùng PCA giảm chiều các mã cổ phiếu sao cho *explained Variance* 95% ra kết quả `Number of PCA components` 12. Biểu diễn `Number of dimensions for clustering with PCA: 12` như sau để dễ dàng nhận thấy:

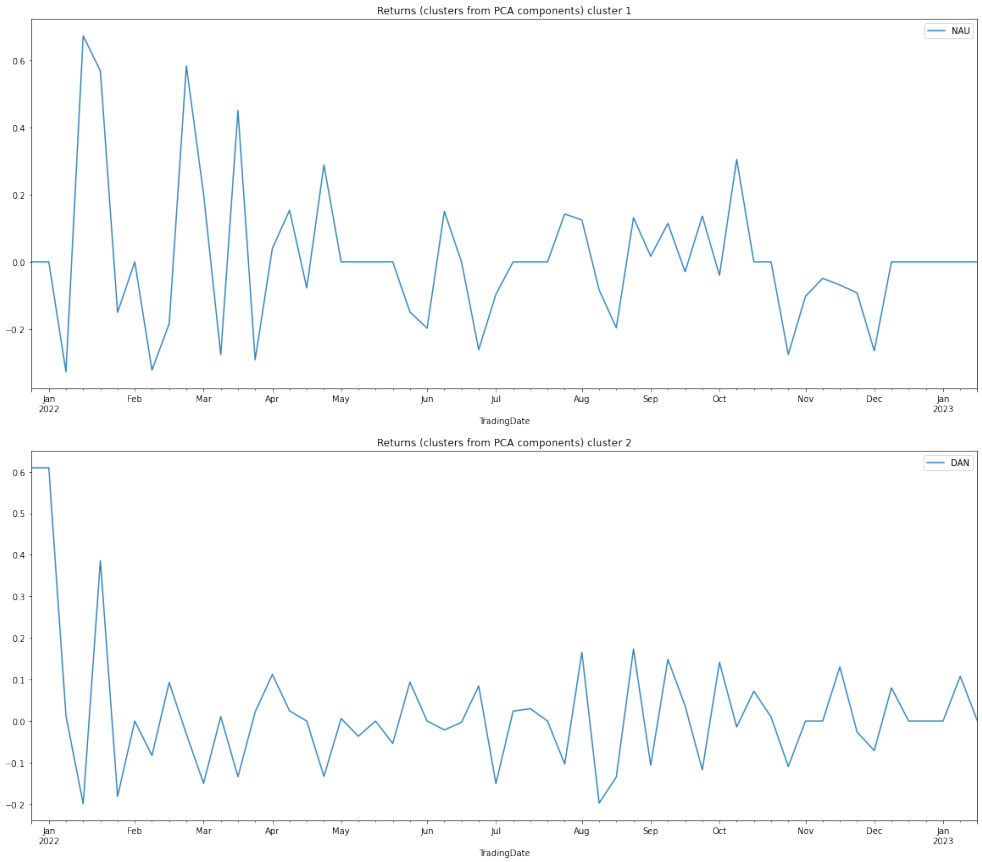


Nhìn vào tổng quan ta đã thấy rõ 2 điểm bất thường đó là : *Công Trình Đô Thị Nghệ An (NAU)* và *Dược Danapha(DAN)*. Tiến hành phân cụm để thấy rõ hơn về dữ liệu sau khi giảm chiều:

- B1: Áp dụng Kmeans bằng thư viện `scikit-learn` với số phâm cụm clusters = 3
- B2: `kmeans.fit()` để phân cụm
- B3: Visualizing data các cổ phiếu từng cụm để nhận biết các cổ phiếu bất thường.



Cluster 0 biểu diễn rõ các mối liên hệ giữa các cổ phiếu trên thị trường.



Cluster 1 & 2 biến động khác hoàn toàn so với các mã cổ phiếu còn lại.

III. Gaussian Mixture model

Phân phối Gaussian, còn được gọi là phân phối chuẩn

$$\mathcal{N}(X|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma|}} exp\{-\frac{(X-\mu)^T\Sigma^{-1}(X\mu)}{2}\}$$

Trong đó μ là vecto trung bình chiều, Σ là ma trận hiệp phương sai DxD, mô tả Gaussian và $|\Sigma|$ biểu thị định thức của Σ.

Phân phối Gaussian đối xứng về giá trị trung bình và mô tả bằng giá trị trung bình, độ lệch chuẩn. Các phân phối phức tạp, đa phương thức có thể được mô hình hóa một cách thích hợp bằng cách sử dụng hỗn hợp phân phối Gaussian.

Gaussian Mixture Model là unsupervised clustering, tạo các cluster hình elip dựa trên mật độ xác suất ước tính bằng cách sử dụng kỳ vọng - tối đa hóa. Mỗi cụm được mô hình hóa dưới dạng phân phối Gaussian. Hiệp phương sai thay vì chỉ có giá trị trung bình như K-means, mang lại cho Gaussian Mixture Model khả năng cung cấp định lượng tốt hơn, thước đo mức độ phù hợp trên một số cụm. Một Gaussian Mixture Model được biểu diên dưới dạng tổ hợp tuyến tính của phân bố xác suất Gaussian cơ bản :

$$p(X) = \sum_{k=1}^{N} \pi_k \mathcal{N}(X|\mu_k,\Sigma_k)$$

Trong đó, K là số thành phần trong mô hình hỗn hợp và π_k được gọi là hệ số trọn, mang lại ước tính mật độ của từng thành phần Gaussian. Mật độ Gaussian cho bởi $\mathcal{N}(X|\mu_k,\Sigma_k)$, được gọi là mật độ thành phần của mô hình hỗn hợp. Mỗi thành phần k được mô tả bằng phân phối Gauss với giá trị trung bình μ_k, hiệp phương sai π_k

1 Gaussian Mixture Model

Khác với K-means, thay vi gán các giá trị cứng cho mỗi quan sát dữ liệu dưới dạng one-hot vector r_ij. Ta sẽ tính xác suất mà mỗi cụm có thể sinh ra quan sát dữ liệu đó. Đầu tiên, ta khởi tạo các tham số vector trung bình μ_k, ma trận covariance Σ_k và hệ số mixing π_k của mỗi cụm k. Gaussian Mixture Model sẽ sử dụng phương pháp Expectation-maximization để tiến hành phân cụm: **E-step:** Với mỗi dữ liệu j, ta tính toán xác suất mà mỗi cụm sinh ra dữ liệu đó bằng cách

$$r_{j,k} = \frac{\pi_k \mathcal{N}(x_j|\mu_k,\sigma_k}{\sum_{i=1}^{K} \pi_i \mathcal{N}(x_j|\mu_i,\sigma_i}$$

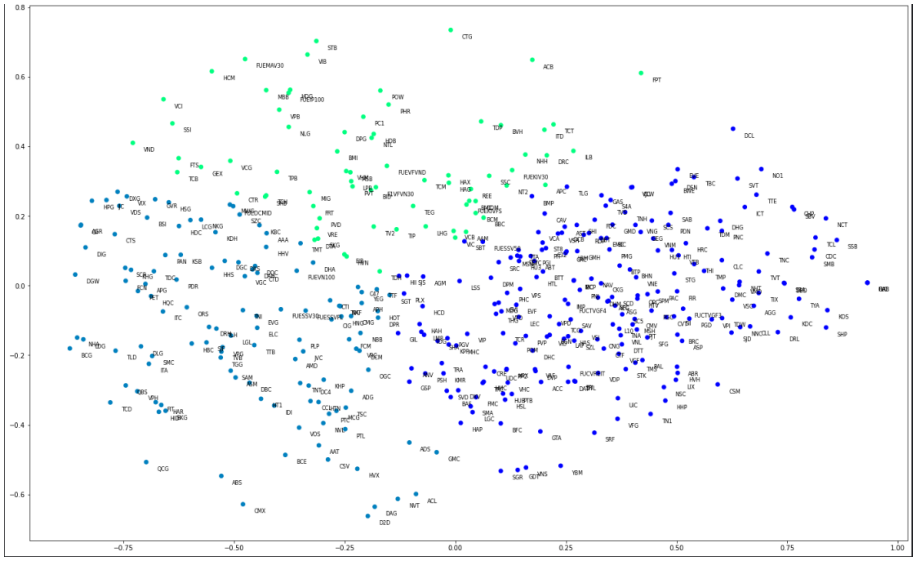
M-step: Với mỗi phân cụm k, ta tính toán lại các tham số μ_k, Σ_k, π_k bằng cách:

$$\mu_k = \frac{1}{\sum_{j=1}^{K} r_{j,k}} \sum_{j=1}^{N} r_{j,k} x_j$$

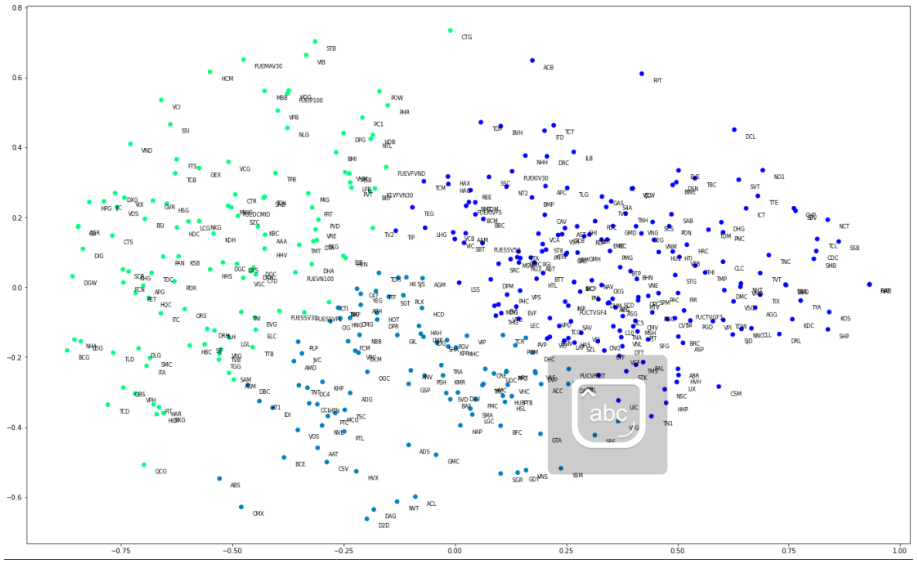
$$\Sigma_k = \frac{1}{\sum_{j=1}^{K} r_{j,k}} \sum_{j=1}^{N} r_{j,k} (x_j - \mu_k)(x_j - \mu_k)^T$$

$$\pi_k = \frac{\sum_{j=1}^{N} r_{j,k}}{N}$$

Ở bài toán này, tụi em sẽ thực hiện việc giảm chiều dữ liệu PCA và tiến hành phân cụm tất cả các cổ phiếu thuộc VN-Index với số cụm là 3.



Gaussian Mixture model



K-means

TÀI LIỆU THAM KHẢO

1. Hoang Thien Ly, PCA based on covariance matrix (PCA-form1) - Principal Component Analysis based frameworks for efficient missing data imputation algorithms https://arxiv.org/pdf/2205.15150.pdf
2. Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model https://www.sciencedirect.com/science/article/pii/S1877050920309820?via%3Dihub
3. https://stats.stackexchange.com/questions/489459/in-cluster-analysis-how-does-gaussian-mixture-model-differ-from-k-means-when-we
4. https://en.wikipedia.org/wiki/Kmeans_clustering
5. `https://machinelearningcoban.com/2017/06/15/pca/#3-principal-component-analysis`