TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH.
PHÒNG QUẢN LÝ KHOA HỌC & HỢP TÁC QUỐC TẾ.

CÔNG TRÌNH DỰ THI
GIẢI THƯỞNG SINH VIÊN NGHIÊN CỨU KHOA HỌC EURÉKA
LẦN THỨ XX NĂM 2022

TÊN CÔNG TRÌNH:

SỬ DỤNG THUẬT TOÁN PHÂN CỤM ĐỂ ĐỀ XUẤT VỊ TRÍ TRẠM XE ĐƯA ĐÓN NHÂN VIÊN.

LĨNH VỰC NGHIÊN CỨU: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: TRÍ TUỆ NHÂN TẠO

Mã số công trình: …….
(Phần này do BTC Giải thưởng ghi)

Mục lục

DANH MỤC HÌNH

SỬ DỤNG THUẬT TOÁN PHÂN CỤM ĐỂ ĐỀ XUẤT VỊ TRÍ TRẠM XE ĐƯA ĐÓN NHÂN VIÊN

Tóm tắt: Việc tổ chức đưa đón nhân viên đi làm mỗi buổi sáng là một hoạt động thiết yếu của nhiều công ty lớn. Với lượng xe có hạn và thông tin khảo sát về chỗ ở của các nhân viên, người quản lý sẽ cân nhắc việc bố trí các trạm xe buýt sao cho hợp lý, giảm thiểu tổng quãng đường di chuyển của nhân viên. Nhằm giải quyết bài toán đó, trong nghiên cứu này, nhóm chúng tôi tìm hiểu sử dụng thuật toán phân cụm K-means để chia nhỏ thành các khu vực mà nhân viên ở gần nhau, dựa trên dữ liệu đầu vào là danh sách địa chỉ nhân viên, sau đó chúng tôi mã hóa địa chỉ bằng ArcGIS API và tính toán ma trận khoảng cách theo đường chim bay dựa trên kinh độ - vĩ độ quy đổi sang tọa độ cực. Trên cơ sở có các cụm, nhóm cũng tìm hiểu triển khai thuật toán Convex Hull tìm bao lồi cho các khu vực và trực quan hóa lên bản đồ thành các miền riêng biệt cho dễ quan sát. Kết quả thu được là một bản đồ trên đó có các điểm tọa độ là trạm xe buýt hoặc chỗ ở nhân viên đã được phân chia một cách trực quan và có thể giúp doanh nghiệp hay công ty dễ dàng tìm được những trạm xe buýt hợp lý và tối ưu.

Từ khóa: Công thức Haversine, Geocode API, thuật toán Convex Hull, thuật toán K-means, thuật toán K-medoids, Trạm xe buýt.

USING CLUSTERING ALGORITHMS TO RECOMMEND THE LOCATIONS OF SHUTTLE BUS

Abstract: Nowadays, the picking up of the staff by shuttle bus every morning is a significant activity of every big company. With a limited number of vehicles and the information from the staff’s residence survey, the manager can consider some fixed bus stations that minimize the staff’s moving distances. To solve this problem, in this research, our group used K-means clustering algorithm to subdivide by area of employees near each other, based on the input data is a list of employee addresses, we then encode the address using the ArcGIS API and calculate distance matrix as crow flies based on longitude - latitude converted to polar coordinates. Based on the clusters, our group also study the convex hull algorithm to find out the boundary of every group of staff and visualize it on the map for viewing purpose. The result is a map on which the coordinate points of bus

stops or employee accommodation are visually divided and can help businesses or companies easily find suitable bus stops, rational and optimal.

Keyword: Bus station, Convex Hull algorithm, Geocode API, Haversine formula, K-means algorithm, K-medoids algorithm

PHẦN MỞ ĐẦU

Lý do chọn đề tài

Hiện nay việc đi làm hàng ngày có thể là một thách thức đối với mỗi người và mất nhiều thời gian hơn bình thường vì các phương tiện giao thông công cộng còn nhiều mặt hạn chế, các vấn đề quấy rối, các quy định an toàn, dịch bệnh, kẹt xe, hình ảnh xe buýt đông đúc và ngột ngạt, tấp nập trên con phố lúc 8 giờ sáng … đối với những nhân viên ở xa nơi làm việc, thì việc đi làm sẽ là rất khó khăn, điều đó làm giảm năng suất và hiệu suất làm việc của nhân viên. Trên thực tế, theo một cuộc khảo sát của Robert Half [1], gần một phần tư số người lao động đã phải rời bỏ việc làm do đường đi làm không tốt. Vì vậy trong các công ty, tập đoàn, doanh nghiệp lớn việc đưa đón nhân viên mỗi bữa sáng là một hoạt động thiết yếu, ví dụ thực tế như Samsung Việt Nam, họ đã có một hệ thống xe buýt với gần 1,000 xe và nhiệm vụ của nó là đưa đón các nhân viên đi làm hàng ngày ở trong phạm vi 60 km xung quanh nhà máy [2]. Một số công ty lớn trong ngành vận tải nhân viên như Arjun Travels, Ambassador Tours & Travels, Safetrax, Vee Vee Bus Services, Wings Travels, Transdev ... Tại Hoa Kỳ, Transdev đã vận chuyển 6.000 người ở San Francisco đến nơi làm việc của họ ở Thung lũng Silicon, cho các công ty như Apple, Google, Microsoft và Genentech [3]. Với số lượng xe có hạn thì doanh nghiệp cần phải giải bài toán cách để quản lí và bố trí các trạm xe buýt hợp lí nhằm giảm thiểu quãng đường di chuyển của nhân viên để đến với nơi làm việc. Nếu bài toán này được giải quyết hiệu quả thì sẽ đem lại một lợi ích to lớn đối với doanh nghiệp. Về mặt kinh tế sẽ tiết kiệm cho doanh nghiệp một chi phí như là giảm chi phí nhiên liệu, chi phí xe đưa đón, chi phí tuyển tài xế, nhân viên sẽ tiết kiệm được thời gian khi đến nơi làm việc do quãng đường đã được giảm thiểu đáng kể và sẽ không phát sinh thêm chi phí đi lại vì đã có doanh nghiệp đưa đón. Về phía công ty, nhân viên đi làm đầy đủ và đúng giờ sẽ là tiền đề để đảm bảo chất lượng công việc cho doanh nghiệp. Về mặt xã hội, nhu cầu cho nhân viên đi làm bằng xe riêng giảm, giảm áp lực lên hạ tầng giao thông, giảm tắc đường. Về mặt môi trường, khi mà số lượng xe giảm thiểu, quãng đường đi được rút gọn thì sẽ góp phần vào việc giảm đáng kể lượng khí thải carbon dioxide bảo vệ môi trường.

Chính vì vậy, nhóm chúng tôi đã nghiên cứu thuật toán phân cụm để giải quyết vấn đề này bằng cách thu thập thông tin địa chỉ của nhân viên để quy đổi sang kinh độ và vĩ độ, trên cơ sở các thông tin đó chúng tôi sử dụng thuật toán phân cụm để trực quan hoá vị trí lên bản đồ và xác định các điểm toạ độ phù hợp với số cụm qua đó đề xuất phương án đặt vị trí đặt xe tốt nhất. Dựa trên những thống kê nghiên cứu và vận dụng các thuật toán để đưa ra giải pháp cho bài toán đặt trạm xe buýt đáp ứng với nhu cầu và phù hợp với chiến lược của doanh nghiệp.

1. Mục tiêu nghiên cứu

- Giải quyết vấn đề đưa đón nhân viên cho các công ty, tập đoàn lớn bằng cách chia nơi ở của nhân viên thành các khu vực và cho xe buýt đưa đón
- Giảm thiểu tình trạng kẹt xe từ phương tiện cá nhân chuyển sang phương tiện công cộng
- Giảm thiểu chi phí đi lại cho nhân viên
- Tiết kiệm thời gian và giảm chi phí cho công ty
- Nâng cao chất lượng lao động

1. Đối tượng nghiên cứu

- Ngôn ngữ lập trình Python và các thư viện của Python
- Machine learning: Thuật toán phân cụm K-mean và K-medoids.
- Thuật toán Convex Hull
- Công thức Haversine tính khoảng cách mặt cầu
- Truy xuất kinh độ, vĩ độ từ thông tin địa chỉ.

2. Phạm vi nghiên cứu

- Bài báo sử dụng dữ liệu địa lý của các thành phố ở Việt Nam năm 2022.
- Nội dung chủ yếu là phân các điểm nơi ở của nhân viên thành các cụm và đặt tâm của cụm đó là trạm xe buýt
- Ứng dụng được xây dựng chủ yếu bằng ngôn ngữ Python

3. Ý nghĩa khoa học và ý nghĩa thực tiễn, quy mô và phạm vi áp dụng

3.1 Ý nghĩa khoa học

- Góp phần vào việc áp dụng học máy vào việc giải quyết các vấn đề thuộc lĩnh vực giao thông vận tải.

3.2 Ý nghĩa thực tiễn

Hỗ trợ và giúp đỡ các công ty doanh nghiệp có số lượng nhân viên lớn có thể tiết kiệm chi phí di chuyển, thời gian và công sức, đảm bảo chất lượng công việc, nâng cao hiệu quả và năng suất làm việc.

3.3 Quy mô và phạm vi áp dụng

Áp dụng cho những tập đoàn, công ty và doanh nghiệp có số lượng nhân viên lớn và có nhu cầu di chuyển tới công ty bằng các phương tiện công cộng, giá rẻ.

PHẦN 1: TỔNG QUAN VỀ TÀI LIỆU

Thuật toán xây dựng trạm xe cho nhân viên là một cấu trúc thuộc thể loại toán học rời rạc và được xây dựng bằng ngôn ngữ lập trình cấp cao thông qua nghiên cứu những kiến thức có trong lĩnh vực công nghệ thông tin và toán học. Mặc dù bài toán còn chưa được tối ưu hoá trong một số trường hợp, tuy nhiên đã tìm ra được một giải pháp gần như giải quyết được các vấn đề về giao thông trong một thành phố đông đúc với nhiều phương tiện đi lại đang làm ảnh hưởng nghiêm trọng đến môi trường.

Bài toán được giải thích đơn giản rằng: Cần nhập số cụm là bao nhiêu để chúng tự động tính khoảng cách và tự sắp xếp lại với nhau trên bản đồ, rồi dùng bao lồi để bao quanh các cụm đã được sắp xếp, tiếp theo chúng dựa vào những cụm đã được bao lồi trước đó để tính trung tâm nằm bên trong từ đó tạo ra trạm xe cho nhân viên di chuyển dễ dàng. Hiện nay cũng có rất nhiều bản đồ điện tử ra đời, nên việc trực quan hoá các dữ liệu bài toán trên cũng không ổn định nhưng cũng sẽ góp phần tạo nên sự tiện lợi cho người dùng. Để giải quyết vấn đề trên nhóm chúng tôi đã đề xuất đề tài "Xây dựng trạm xe đưa đón nhân viên" cùng với mã hoá dữ liệu đầu vào, thuật toán phân cụm K-means và bao lồi các cụm Convex Hull.

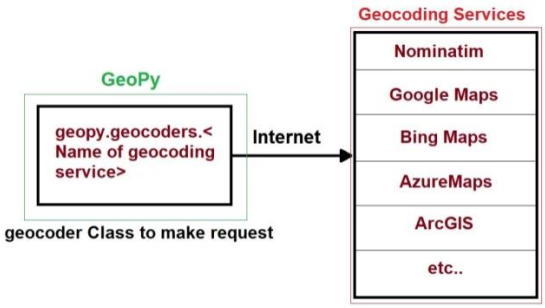
PHẦN 2: PHƯƠNG PHÁP

1. Mã hóa địa chỉ

Như chúng tôi đã nói ở trên, dữ liệu địa chỉ đầu vào là danh sách địa chỉ nhân viên trong đó chứa các thông tin như: số nhà, tên đường, quận/huyện, tỉnh thành và thành phố và từ dữ liệu thô này, chúng tôi sẽ xử lý trả về kinh độ, vĩ độ. Sau một thời gian nghiên cứu, chọn lọc, chúng tôi đã đưa ra hai phương án tối ưu nhất, phù hợp: Dịch vụ mã hóa địa lý Geocode API (Application Programming Interface – API) và Selenium truy vấn lấy kinh độ, vĩ độ.

1.1 Dịch vụ mã hóa địa lý Geocode API

Để lấy được kinh độ vĩ độ, thì GeoPy một thư viện của ngôn ngữ Python được xem như là một ứng dụng khách, cầu nối, liên kết với một số dịch vụ web mã hóa địa lý phổ biến để truy xuất ra kinh độ, vĩ độ. Hình dưới đây cho biết ý tưởng về chức năng của GeoPy:



Hình 1. Ảnh ví dụ các bên thứ ba liên kết với GeoPy.

Như trong hình trên mã hóa địa lý được cung cấp bởi một trong số nhiều dịch vụ khác nhau chúng ta có thể lựa chọn tùy vào nhu cầu hiện tại của khách hàng. Mỗi dịch vụ định vị địa lý, chẳng hạn như Google Maps, Bing Maps hoặc Nominatim, ArcGIS có một lớp riêng

trong geopy.geocoders tóm tắt API của dịch vụ. Mỗi bộ mã hóa địa lý xác định ít nhất một phương thức mã hóa địa lý, để phân giải một vị trí từ một chuỗi từ đó truy xuất thành một cặp tọa độ và có thể xác định một phương pháp, ngược lại, phương pháp này phân giải một cặp tọa độ thành một địa chỉ [4].

Ở đề tài này, chúng tôi sử dụng thư viện GeoPy kết hợp với dịch vụ ArcGIS với ưu điểm là nhanh chóng trong việc truy xuất kinh độ, vĩ độ và họ cung cấp 1 triệu yêu cầu truy vấn miễn phí mỗi tháng. Nhưng có nhược điểm là độ chính xác thấp và độ lệch trung bình chúng tôi nghiên cứu khoảng 50m cho một địa chỉ.

Cách thức lấy kinh độ, vĩ độ:

Đầu tiên, chúng tôi gọi ra dịch vụ mã hóa địa lý ArcGIS tạo ra một phiên bản của lớp ArcGIS. Sau khi tạo lớp, chúng tôi áp dụng phương pháp mã hóa địa lý để truy xuất ra vị trí từ dữ liệu địa chỉ đầu vào. Sử dụng phương thức ‘latitude” và “longtitude” để lấy kinh độ vĩ độ.

1.2 Selenium truy vấn lấy kinh độ, vĩ độ

Phương pháp này được chúng tôi nghĩ ra thông qua nhiều lần tra địa chỉ thực tế trên Google Chrome. Chúng tôi nhận thấy rằng mỗi lần tra cứu một địa chỉ thực tế nào đó thì kinh độ, vĩ độ sẽ được cập nhật thông qua đường dẫn. Và mục tiêu chúng tôi là lấy đường dẫn đó về và sử dụng phương thức Regex trích xuất ra kinh độ, vĩ độ. Ưu điểm của phương pháp này cho kết quả tương đối chính xác, và không tốn kém chi phí nhưng nhược điểm là việc truy vấn lâu.

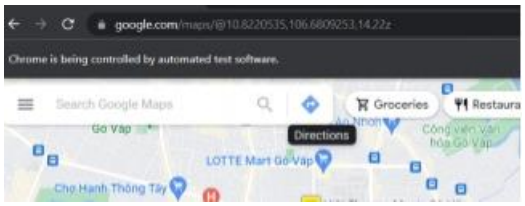
Trước tiên, chúng tôi đã tìm địa chỉ “Trường Đại học Công nghiệp TP HCM”:

Hình 2. Ảnh ví dụ về lấy kinh độ, vĩ độ.

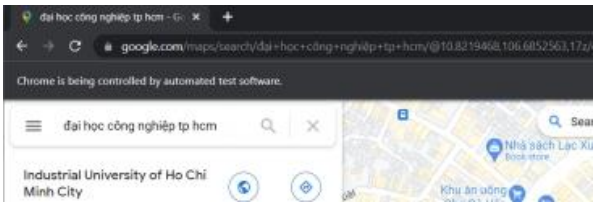
Và để thực hiện ý tưởng này, chúng tôi đã sử dụng thư viện Selenium của Python cho phép tương tác từng bước với trang web của Google Maps. Sau đó, chúng tôi cài đặt ChromeDriver để kiểm tra tự động các ứng dụng web trên trình duyệt Google Chrome, cung cấp khả năng điều hướng đến trang web và nhập dữ liệu địa chỉ đầu vào để tìm kiếm. Nhưng để truy cập tới trang Google Maps thì chúng tôi phải khởi tạo phiên bản WebDriver của Google Chrome truyền vào đường dẫn lưu trữ ChromeDriver.

Sau đó chúng tôi tận dụng phương thức driver.get để điều hướng đến trang web Google Maps thông qua đường dẫn *https://www.google.com/maps/*. WebDriver sẽ đợi đến khi trang được tải đầy đủ trước khi trả lại quyền kiểm soát cho thử nghiệm hoặc tập lệnh của chúng tôi.

Tiếp đến chúng tôi tạo ra một hàm tìm kiếm tự động trên Google Maps bằng cách xác định tên class trên khung kìm kiếm trong trang Google Maps truyền vào phương thức *find_element_by_name* để định vị, vị trí nhập địa chỉ. Tiếp theo, chúng tôi gửi chuỗi địa chỉ cần tìm vào, điều này tương tự như nhập các phím bằng bàn phím. Và sử dụng phương thức send key (Keys.RETURN), tương tự như nút tìm kiếm trên trang. Sau đó đường dẫn mới trên trang Google Maps sẽ được cập nhật mới.



Hình 3. Đường dẫn trước khi tìm kiếm.



Hình 4. Đường dẫn sau khi tìm kiếm.

Rõ ràng sau khi tìm kiếm chúng tôi đã có được đường dẫn và việc sau đó là chúng tôi sẽ lấy đường dẫn đó về bằng phương thức *driver.current_url*. Việc cuối cùng, chúng tôi sử dụng RegEx để trích xuất ra kinh độ vĩ.

2. Tính khoảng cách theo lý thuyết

Để đo khoảng cách vị trí giữa các nhân viên chúng tôi sử dụng công thức Haversine là công thức dùng trong xác định khoảng cách đường tròn lớn giữa hai điểm trên một mặt cầu dựa trên kinh độ và vĩ độ của hai điểm đó[5]. Đường hasrsine có thể được biểu thị dưới dạng hàm lượng giác như sau:

$$Haversine(\theta) = Sin^2\left(\frac{\theta}{2}\right)$$

Đường hasrsine của góc ở tâm (là d / r) được tính theo công thức sau:

$$\left(\frac{d}{r}\right) = havesine(\Phi 2 - \Phi 1) + \cos(\Phi 1)\cos(\Phi 2)\, haversine(\lambda 2 - \lambda 1)$$

Trong đó r là bán kính trái đất (6371 km), d là khoảng cách giữa hai điểm, Φ_1, Φ_2 là vĩ độ của hai điểm, và λ_1, λ_2 là kinh độ của hai điểm tương ứng [6].

Giải d bằng cách sử dụng hàm sin nghịch đảo, chúng ta nhận được:

$$d = 2.r.\arcsin\left(\sqrt{sin^2\left(\frac{\Phi 2 - \Phi 1}{2}\right) + \cos(\Phi 1)\cos(\Phi 2)\, sin^2\left(\frac{\lambda 2 - \lambda 1}{2}\right)}\right)$$

Kết hợp với thư viện math của python để xây dựng chương trình tính khoảng cách và biểu diễn thành ma trận nhiều vị trí khác nhau của các nhân viên với dữ liệu vào ban đầu là số vị trí " n" và các dòng tiếp theo lần lượt là " lat A lon A", " lat B lon B", …, " lat n lon n"

Mô tả: Cách tính ma trận khoảng cách

```
1. import math
2. def Haversine_formula(lat1, long1, lat2, long2):
3.       lat1 = math.radians(lat1)
4.              # Hoặc lat1 = lat1 * math.pi/180
5.       lat2 = math.radians(lat2)
6.              # Hoặc lat2 = lat2 * math.pi/180
7.       long1 = math.radians(long1)
8.              # Hoặc long1 = long1 * math.pi * 180
9.       long2 = math.radians(long2)
10.            # Hoặc long2 = long2 * math.pi * 180
```

```
    temp = math.asin(math.sqrt(math.sin((lat2-
lat1)/2)**2 + math.cos(lat1) * math.cos(lat2) *
math.sin((long2-long1)/2)**2))
        d = 2 * r * temp
        return round(d, 3)
    r = 6371 # bán kính trái đất theo dữ liệu từ
google
    n = int(input())
    lati = []
    longi = []
    for i in range(n):
        x, y = map(float, input().split())
        lati.append(x)
        longi.append(y)
    for i in range(len(lati)):
        for j in range(len(longi)):
            print(Haversine_formula(lati[i],
longi[i], lati[j], longi[j]), end = " ")
    print()
```

3. Thuật toán phân cụm K-means clustering

3.1 Giới thiệu về thuật toán K-means

K-means được ra đời bởi nhà toán học người Ba Lan là Hugo Steinhaus vào năm 1956. Thuật toán phân cụm được giới thiệu trong bài báo bằng tiếng Pháp "Sur la Divison dé Cops Matériels en Parties". Năm 1967, James MacQueen đưa ra thuật ngữ K-means trong bài báo cáo "Some Methods for Classification and Analysis of Multivariate Observations". Cơ bản với cùng phương pháp đó Edward W. Forgy đã công bố vào năm 1965 , vậy nên đôi khi được gọi là Lloyd-Forgy.

Đây là phương pháp phân các dữ liệu điểm đầu vào thành các cụm khác nhau. Thuật toán K-means có nhiều ứng dụng, trong đó trí tuệ nhân tạo và học máy là sử dụng nhiều nhất.

Phương pháp của thuật toán K-means là tạo và cập nhật liên tục điểm trung tâm và phân các điểm dữ liệu vào các nhóm khác nhau. Ban đầu, tạo ra trung điểm ngẫu nhiên và gán mỗi điểm từ tập dữ liệu vào tâm gần nó nhất. Sau đó thuật toán sẽ cập nhật lại điểm trung tâm và lặp lại các bước cho tới khi đạt được kết quả gần như chính xác.[7]

3.2 Độ phức tạp của thuật toán K-means

3.2.1 Mô tả

Cho một tập hợp điểm $(x_1, x_2, \dots, x_n)$, phân cụm K-means giúp chia n điểm thành k $(\leq n)$ cụm với tập hợp cụm $S = \{S_1, S_2, \dots, S_k\}$. Được tuân theo quy tắc:

$$\underset{S}{argsin} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = \underset{S}{argsin} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Với μ_i là giá trị trung bình của các điểm trong S_i. Điều này tương đương với việc giảm thiểu độ lệch bình phương theo cặp của các điểm trong cùng một cụm:

$$\underset{S}{argsin} \sum_{i=1}^{k} \frac{1}{|S_i|} \sum_{x,y \in S_i} \|x - y\|^2$$

Từ đó, ta có thể suy ra công thức tương đương:

$$|S_i| \sum_{x \in S_i} \|x - \mu_i\|^2 = \sum_{x \neq y \in S_i} \|x - y\|^2$$

Vì tổng phương sai không đổi nên với công thức trên có thể tối đa hóa tổng bình phương độ lệch giữa các điểm trong các cụm khác nhau.[7]

3.2.2 Thuật toán tiêu chuẩn

Với tập k điểm trung tâm ban đầu $m_1, m_2, \dots, m_k$ thuật toán được thực hiện liên tục và lặp lại hai bước:

Bước chỉ định: Gán mỗi điểm cho cụm có giá trị trung bình gần nhất với khoảng cách Euclide bình phương nhỏ nhất. Ta có:

$$S_i = \{x_p : \|x_p - m_i\|^2 \leq \|x - m_j\|^2 \ \forall j, 1 \leq j \leq k\}$$

Mỗi điểm x_p được chỉ định cho chính xác một tập S.

Bước cập nhật: Tính toán lại giá trị trung bình (trọng tâm) cho từng cụm:

$$m_i = \frac{1}{|S_i|} \sum_{x, j \epsilon\, S_i} x_j$$

Thuật toán được dừng lại khi giá trị của các phép gán không còn thay đổi. [7]

3.3 Ưu điểm và nhược điểm của thuật toán K-means

Thuật toán này rất phù hợp khi số cụm K cho trước luôn ổn định, có khả năng mở rộng và dễ dàng sửa đổi với các dữ liệu phân cụm mới, trải qua một vài vòng lặp hữu hạn thì vẫn giữ được tính hội tụ của một cụm, các cụm cũng sẽ được tách biệt rõ ràng, không để xảy ra hiện tượng một đối tượng xuất hiện chung với các cụm dữ liệu khác. Đặc biệt hơn là chúng tạo ra cụm chặt chẽ hơn so với các thuật toán phân cụm khác nếu các cụm có dạng hình cầu. Mặc dù áp dụng rất phổ biến nhưng K-means vẫn còn khá nhiều hạn chế. Số điểm khởi tạo ban đầu sẽ quyết định tâm của cụm, các vị trí được khởi tạo khác nhau sẽ ảnh hưởng đến cách phân cụm, dù thuật toán có cùng số cụm. Tóm lại vị trí khởi tạo khác nhau có thể dẫn tới cách phân cụm khác nhau, không thể đảm bảo đạt được tối ưu toàn cục. Phương pháp duy nhất là thử và sai (try and error) và phương pháp xác định số cụm như Elbow. Trước khi huấn luyện thuật toán ta phải đảm bảo loại bỏ outliers (dữ liệu ngoại lai), chỉ khi tính được trọng tâm của cụm mới ứng dụng được thuật toán này. Để tìm được tâm cụm gần nhất, Thuật toán K-means yêu cầu phải tính khoảng cách từ một điểm tới toàn bộ các tâm cụm, nên sẽ rất khó thích nghi với bộ nhớ của máy, đặc biệt là RAM vì nó cần phải load toàn bộ dữ liệu, từ đó nó sẽ vượt quá khả năng lưu trữ khi triển khai dữ liệu lớn, thay vào đó phải sử dụng phương pháp online learning. [8]

3.4 Các bước thực hiện

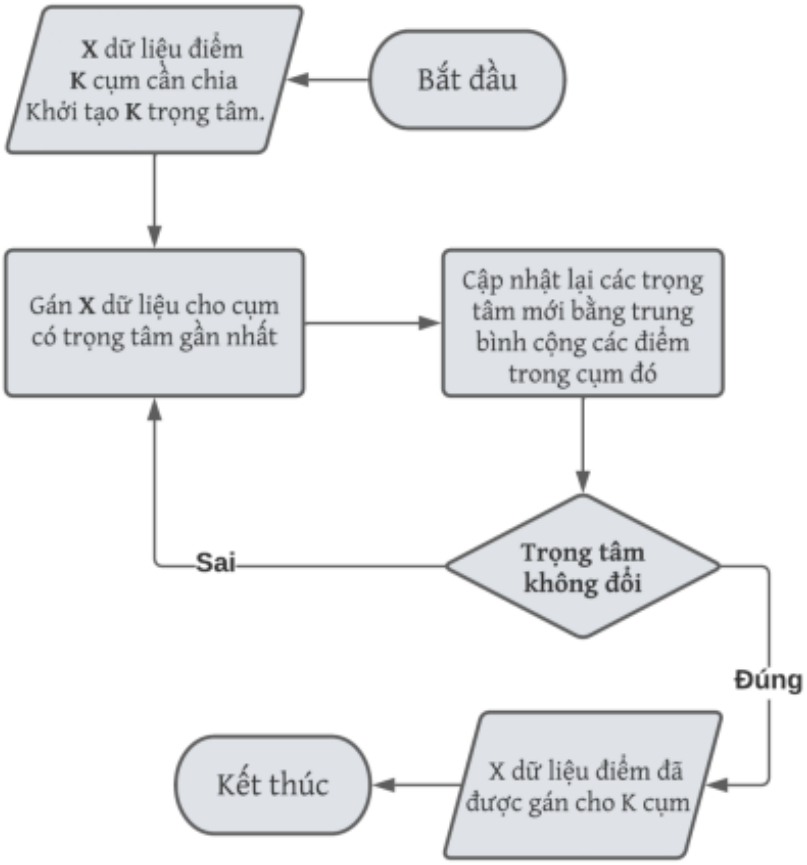
Bước 1:

Chúng tôi nhận đầu vào là một tệp excel (có dạng .xlsx). Sau đó sử dụng thư viện pandas để đọc file lưu vào trong biến dạng DataFrame, và trong file đó ít nhất phải chứa cột "Address" chứa địa chỉ thực. Mỗi phần tử của danh sách là một danh sách khác chứa các giá trị mục cho các đối tượng địa lý.

	G	H	I	J
	Họ đệm	Tên	Giới tính	Address
1	Triệu Quốc	An	Nam	1,Hoàng Cầu,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
1	Nguyễn Đứ	Anh	Nam	3,Thành Công,Khu tập thể Bắc Thành Công,Thành Công,Ba Đình District,Hanoi,Vietnam,
1	Dương Thá	Bảo	Nam	Ngõ 3 Thái Hà,Trung Liệt,Hanoi,Hoàn Kiếm District,Hanoi,Vietnam,

Hình 5. Danh sách chứa các đối tượng địa lý đầu vào.

Bước 2:
Khởi tạo K trọng tâm bất kì.
Bước 3:
Sau đó chúng tôi tính ma trận khoảng cách. Gán các điểm vào cụm có trọng tâm gần nó nhất.
Bước 4:
Nếu việc gán dữ liệu vào cụm ở bước 02 không thay đổi so với vòng lặp trước đó thì ta dừng và đưa ra kết quả.
Bước 5:
Cặp nhật các trọng tâm mới bằng cách lấy trung bình cộng của các điểm được gán ở mỗi cụm trước đó.
Bước 6:
Trở lại bước 02.



Hình 6. Sơ đồ tóm tắt các bước thực hiện của thuật toán.

4. Thuật toán K-medoids

4.1 Giới thiệu về K-medoids

K-medoids là một thuận toán phân cụm tương tự như k-means. Tên gọi này được đặt ra bởi Peter J. Rousseeuw và Leonard Kaufman hay còn gọi là PAM (partition around medoids). Cả thuật toán k-means và k-medoids đều chia nhỏ dữ liệu thành các nhóm nhỏ và làm giảm thiểu khoảng cách giữa các điểm tới điểm trung tâm của cụm đó. Trong đó, điểm trung tâm là một trong các điểm thuộc cụm đó.
K-medoids là một kỹ thuật phân cụm chia tập hợp dữ liệu gồm n điểm thành k cụm, trong đó k là số cho trước.[13]

4.2 Thuật toán

K-medoids sử dụng thuật toán tìm kiếm tham lam, nó có thể không phải giải pháp tối ưu nhưng nhanh hơn so với tìm kiếm vét cạn. Thuật toán hoạt động như sau:

1. (BUILD) khởi tạo: chọn k trong số n điểm dữ liệu làm trung điểm
2. Liên kết mỗi điểm dữ liệu với medoid gần nhất
3. (SWAP) Trong khi giá trị hình thể giảm:
   - Đối với mỗi medoid m và với mỗi điểm không phải medoid O:
     Xem xét sự hoán đổi của m và O và tính toán sự thay đổi giá trị
     Nếu thay đổi giá trị là gần nhất thì ta thay m bằng O
   - Thực hiện liên tục cho tới khi medoid m là chính xác nhất. Thuật toán kết thúc

Độ phức tạp thời gian chạy của thuật toán PAM trên mỗi lần lặp là $O(k(n-k)^2)$, bằng cách chỉ thay tính toán sự thay đổi giá trị. Một triển khai tính toán tổng quát mọi thời điểm $O(n^2k^2)$. Thời gian chạy này có thể giảm xuống $O(n^2)$ bằng cách chia sự thay đổi giá trị thành ba phần để thực các phép tính hoặc liên tục thực hiện hoán đổi FastPAM. Tại thời điểm đó, khởi tạo ngẫu nhiên có thể trở thành giải pháp thay thế cho BUILD.[13]

5. Áp dụng thuật toán vào đề tài

Như đã biết thì thuật toán K-means được dùng để gom dữ liệu đã cho thành các cụm dữ liệu, sao cho dữ liệu đó đến với tâm cụm gần nó nhất. Dựa trên những đặc tính của thuật toán K-means, nhóm nghiên cứu đã áp dụng vào việc phân cụm nơi ở nhân viên thành các cụm các khu vực sao cho đường đi từ nơi ở của nhân viên đến vị trí tâm cụm đó là ngắn nhất. Theo dữ liệu thu thập được là vị trí chỗ ở của các nhân viên cần đưa đón.

Dữ liệu đầu vào theo đúng các trường như nhau:

<Số nhà>, <Tên đường>, <Phường/Xã>, <Quận/Huyện>, <Tỉnh/Thành phố>, <Quốc Gia>.

Ví dụ

12, Nguyễn Văn Bảo, Phường 04, Quận Gò Vấp, Thành phố Hồ Chí Minh, Việt Nam.

Qua thông tin địa chỉ thu thập được, bằng kỹ thuật dữ liệu nhóm nghiên cứu chuyển đổi những dòng địa chỉ đó trở thành toạ độ tương ứng.

Ví dụ với địa chỉ "12, Nguyễn Văn Bảo, Phường 04, Quận Gò Vấp, Thành phố Hồ Chí Minh, Việt Nam". Thì ta sẽ thu được cặp kinh độ, vĩ độ là 10.8222053, 106.6874994.

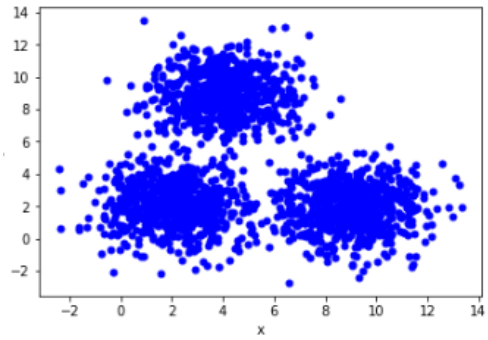
Dựa vào các địa chỉ, chúng tôi đã viết chương trình để chuyển đổi địa chỉ thực thành toạ độ địa lí tương ứng mà máy tính có thể tính toán và xử lí được.

Sau khi chuyển hoá hết toàn bộ thông tin vị trí, đây là lúc mà thuật toán K-means được đưa vào áp dụng, với dữ liệu đầu vào là các điểm, sau đó tính toán khoảng cách địa lý giữa các điểm với nhau. Cuối cùng là phân các điểm đó thành K cụm tương ứng (Với số K phụ thuộc vào yêu cầu của công ty nhưng vẫn đảm bảo số lượng cụm phải nhỏ nhất).

6. Mô phỏng

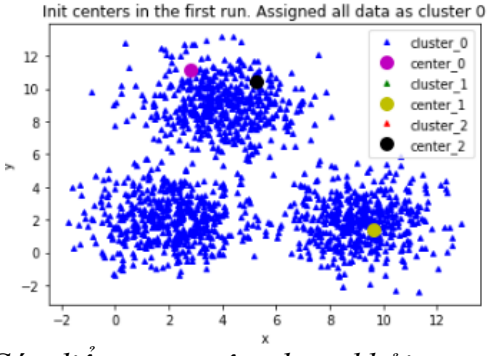
Thuật toán K-means là thuật toán tuân theo một quy trình lặp đi lặp lại trong đó, nó cố gắng giảm thiểu khoảng cách của các điểm dữ liệu so với các điểm trọng tâm [9] nhờ đó mà độ chính xác giữa các điểm được tối đa hóa. Để chứng minh cho điều này chúng tôi đã thực hiện một mô phỏng về quá trình thực hiện của thuật toán K-means.

Đầu tiên, chúng tôi khởi tạo 500 điểm và số trung tâm là 3:



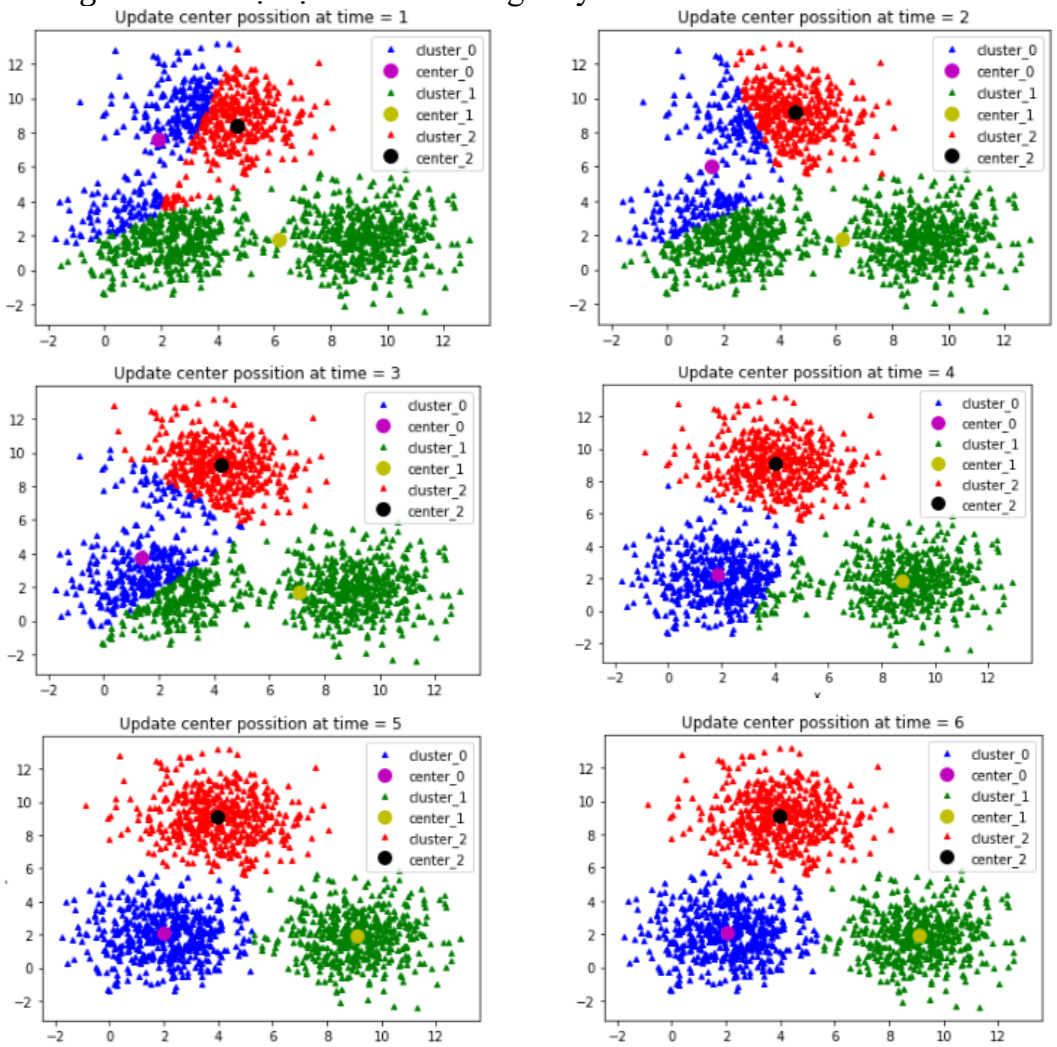
Hình 7. Các điểm được trực quan hóa.

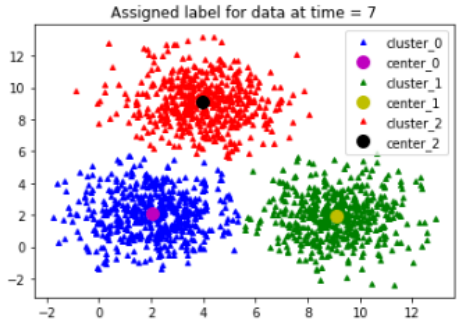
Sau đó chúng tôi sử dụng thuật toán K-means bắt đầu khởi tạo 3 điểm trung tâm, do các điểm trung tâm không thể xác định một cách chính xác. Vì thế chúng tôi sẽ khởi tạo 3 điểm trung tâm một cách ngẫu nhiên và xác định chúng làm trung tâm cho mỗi cụm.



Hình 8. Các điểm trung tâm được khởi tạo ngẫu nhiên.

Khi khởi tạo xong 3 điểm trung tâm, chúng tôi bắt đầu cập nhật tâm cho từng cụm liên tục và chỉ dừng nếu như toạ độ của tâm không thay đổi:

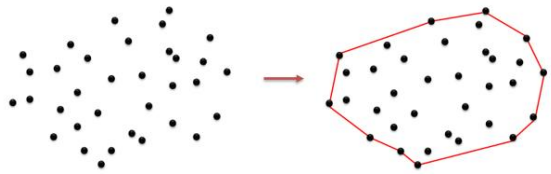




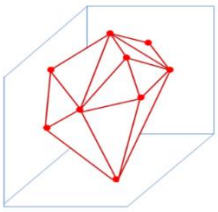
Hình 9. Mô tả quá trình cập nhật 3 cụm.

6. Thuật toán Convex Hull

Trong hình học tính toán (computational geometry), bao lồi (Convex Hull) là phần lồi của một tập hợp các điểm được xem là tập lồi nhỏ nhất sao cho diện tích nhỏ nhất khi ở trong hình học phẳng và thể tích nhỏ nhất trong hình học không gian mà chứa mọi điểm trong tập hợp đó[10].



Hình 10. Bao lồi trong hình học phẳng.



Hình 11. Bao lồi trong hình học không gian.

Hiện nay thuật toán để tìm bao lồi được áp dụng cho nhiều lĩnh vực khác nhau như toán học, thống kê, tối ưu hóa tổ hợp, kinh tế, mô hình hóa hình học, vật lý lượng tử, khoa học vật liệu, khoa học dữ liệu, …

Khi nhìn bằng mắt vào một tập hợp các điểm, con người chúng ta có thể dễ dàng để nhận biết được đâu là vỏ lồi và nó đi qua những điểm nào, nhưng để máy tính làm được việc này thì không đơn giản như vậy, và chúng ta cần viết những đoạn mã để xử lý công việc này. Để tìm tập hợp các điểm thuộc phần lồi chúng ta có thể sử dụng thuật toán sau.

Đầu tiên chúng ta cần tạo một class (python) để lưu tọa độ kinh độ vĩ độ của các địa chỉ

Mô tả: Tạo class lưu kinh độ, vĩ độ.

```
1. class Point:
2.       def __init__(self, x, y):
3.            self.x = x
4.            self.y = y
```

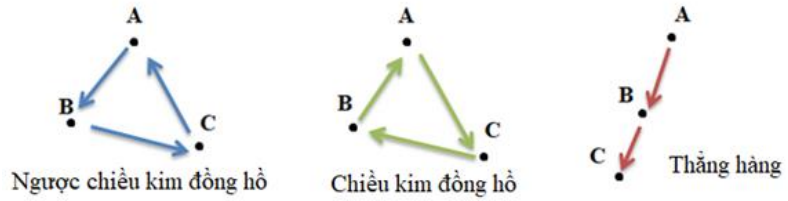
Sau đó tạo ra một hàm để tìm ra một điểm bất kì nằm trên bao lồi ví dụ tọa độ trên cùng bên trái, hàm dưới đây sẽ trả về index của tọa độ ngoài cùng bên trái và nếu có nhiều tọa độ như vậy (x bằng nhau) thì sẽ lấy tọa độ có vĩ độ (y) lớn nhất.

Mô tả: Hàm trả về tọa độ trên cùng bên trái của tập tọa độ.

```
1. def Left_index(points):
2.       #Tìm điểm xa nhất bên trái
3.       minn = 0
4.       for i in range(1,len(points)):
5.            if points[i].x < points[minn].x:
6.                 minn = i
7.            elif points[i].x == points[minn].x:
8.                 if points[i].y > points[minn].y:
9.                      minn = i
10.      return minn
```

Tiếp đó chúng ta viết hàm xác định vị trí của 3 điểm p, q, r
Định hướng của bộ 3 điểm có thứ tự trong mặt phẳng có thể là:

- Ngược chiều kim đồng hồ
- Chiều kim đồng hồ
- Thẳng hàng



Hình 12. Sơ đồ cho thấy các định hướng của 3 điểm A, B, C .

Ý tưởng để tính toán toán định hướng là tính độ dốc của từng đoạn thẳng AB, BC, ví dụ tọa độ các điểm là A(x1, y1), B(x2, y2), C(x3, y3)

Độ dốc của đoạn thẳng AB là : **d1 = (y2 - y1) / (x2 - x1)**

Độ dốc của đoạn thẳng BC là : **d2 = (y3 - y2) / (x3 - x2)**

Nếu d1 > d2 hướng theo kim đồng hồ (rẽ phải)

Từ những giá trị d1, d2 ở trên chúng ta có thể kết luận rằng, hướng phụ thuộc vào dấu hiệu của biểu thức dưới :

(y2 - y1) * (x3 - x2) - (y3 - y2) * (x2 - x1)

Nếu biểu thức dương thì A, B, C theo chiều kim đồng hồ, nếu âm thì ngược chiều kim đồng hồ, còn nếu bằng không thì A, B, C thẳng hàng [11].
Dưới đây là code thực hiện ý tưởng trên :

Mô tả: Hàm kiểm tra định hướng của ba điểm.

```
1. def orientation(p, q, r):
2.       val = (q.y - p.y) * (r.x - q.x) - (q.x - p.x)
   * (r.y - q.y)
3.       if val == 0:
4.            return 0
5.       elif val > 0:
6.            return 1
7.       else:
8.            return 2
```

Hàm sẽ trả về các giá trị sau

0 à p, q và r thẳng hàng

1 à Theo chiều kim đồng hồ

2 à Nếu Ngược chiều kim đồng hồ

Cuối cùng là thực hiện thuật toán Convex Hull [12].

Mô tả: Thuật toán Convex Hull.

```
1. def convexHull(points, n):
2.      if n < 3:
3.           return
4.      l = Left_index(points)
5.      hull = []
6.      p = l
7.      q = 0
8.      while(True):
9.           hull.append(p)
10.               q = (p + 1) % n
11.               for i in range(n):
12.                    if(orientation(points[p],
13.                                   points[i],
  points[q]) == 2):
14.                         q = i
15.               p = q
16.               if(p == l):
17.                    break
18.          ans = []
19.          for each in hull:
20.               ans.append([points[each].x,
  points[each].y])
21.          return ans
```

Khi tìm được điểm chắc chắn nằm trong bao lồi thì xuất phát từ điểm này ta sẽ đi tìm các điểm khác nằm trong bao lồi cho đến khi quay trở lại điểm ta chọn lúc đầu. Độ phức tạp của thuật toán là $O(n^2)$ đủ nhanh để đáp ứng nhu cầu của đề tài.

PHẦN 3: KẾT QUẢ - THẢO LUẬN

Sau quá trình nghiên cứu trên nhóm chúng tôi đã đạt được những kết quả đáng kể
Từ tệp dữ liệu đầu vào là một file excel hoặc csv chứa các địa chỉ của nhân viên như hình bên dưới:

STT	MSSV	Họ đệm	Tên	Giới tính	Địa chỉ
1	21048321	Triệu Quốc	An	Nam	1,Hoàng Cầu,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
2	21009991	Nguyễn Đức	Anh	Nam	3,Thành Công,Khu tập thể Bắc Thành Công,Thành Công,Ba Đình District,Hanoi,Vietnam,
3	21037621	Dương Thái	Bảo	Nam	Ngõ 3 Thái Hà,Trung Liệt,Hanoi,Hoàn Kiếm District,Hanoi,Vietnam,
4	21044391	Nguyễn Duy	Bảo	Nam	2,Trần Quang Diệu,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
5	21078921	Nguyễn Trần Quốc	Bảo	Nam	10,Thành Công,Khu tập thể Bắc Thành Công,Thành Công,Ba Đình District,Hanoi,Vietnam,
6	21045171	Võ Trần Quốc	Bảo	Nam	Ngõ 252 Tây Sơn,Trung Liệt,Hanoi,Hoàn Kiếm District,Hanoi,Vietnam,
7	21057571	Bùi Thanh	Bình	Nam	26,Vũ Thạnh,Ô Chợ Dừa,Hanoi (Ha Noi),Đống Đa,Hanoi,Vietnam,
8	21052051	Trần Chí	Bình	Nam	8,Nguyên Hồng,Khu tập thể Bắc Thành Công,Thành Công,Đống Đa,Hanoi,Vietnam,
9	21057211	Trần Thị Mỹ	Châu	Nữ	2,Trần Quang Diệu,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
10	21057631	Võ Ngọc	Châu	Nam	1010000,Nguyên Hồng,Thành Công,Ba Đình District,Hanoi,Vietnam,
11	21086741	Đỗ Thành	Đạt	Nam	116,Thành Công,Khu tập thể Bắc Thành Công,Thành Công,Ba Đình District,Hanoi,Vietnam,
12	21031381	Giáp Văn	Đạt	Nam	57,Nguyễn Phúc Lai,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
13	21021331	Lê Hữu	Đạt	Nam	11,Nguyễn Phúc Lai,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
14	21053421	Lê Thành	Đạt	Nam	16,Nguyễn Lương Bằng,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
15	21024081	Nguyễn Thanh	Định	Nam	24,Hào Nam,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,
16	21022491	Hà Linh	Duy	Nam	89,Hào Nam,Ô Chợ Dừa,Hanoi,Đống Đa,Hanoi,Vietnam,
17	21047321	Nguyễn Trường	Duy	Nam	109,Trung Liệt,Trung Liệt,Đống Đa,Hanoi,Vietnam,
18	21018551	Nguyễn Hữu Phước	Hải	Nam	19,Nguyên Hồng,Khu tập thể Bắc Thành Công,Thành Công,Ba Đình District,Hanoi,Vietnam,

Hình 13. Danh sách chứa các đối tượng địa lý đầu vào.

Chúng tôi sẽ tiến hành trích xuất các kinh độ, vĩ độ và tính khoảng cách giữa các điểm. Sau đó là áp dụng thuật toán phân cụm K-means để phân chia các điểm tọa độ gần nhau thành một cụm như hình bên dưới:

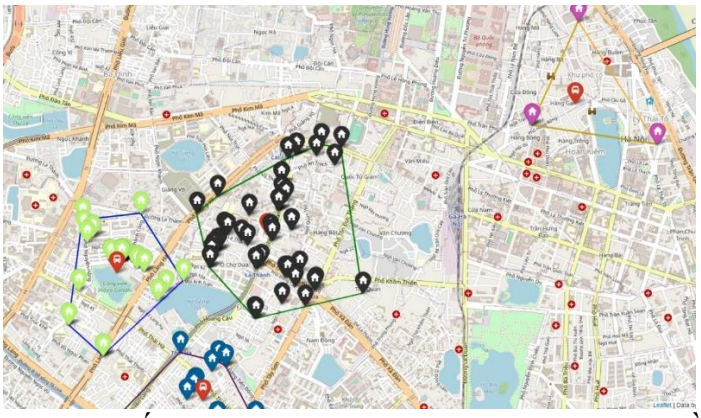
	STT	MSSV	Họ đệm	Tên	Giới tính	Địa chỉ	Latitude	Logtitude	Cluter
0	1	21048321	Triệu Quốc	An	Nam	1,Hoàng Cầu,Ô Chợ Dừa,Đống Đa,Hanoi,Vietnam,	21.020290	105.825570	0
1	2	21009991	Nguyễn Đức	Anh	Nam	3,Thành Công,Khu tập thể Bắc Thành Công,Thành ...	21.020080	105.815970	1
2	3	21037621	Dương Thái	Bảo	Nam	Ngõ 3 Thái Hà,Trung Liệt,Hanoi,Hoàn Kiếm Distr...	21.008996	105.822197	1
3	4	21044391	Nguyễn Duy	Bảo	Nam	2,Trần Quang Diệu,Ô Chợ Dừa,Đống Đa,Hanoi,Viet...	21.016550	105.825090	0
4	5	21078921	Nguyễn Trần Quốc	Bảo	Nam	10,Thành Công,Khu tập thể Bắc Thành Công,Thành...	21.019890	105.816549	1
5	6	21045171	Võ Trần Quốc	Bảo	Nam	Ngõ 252 Tây Sơn,Trung Liệt,Hanoi,Hoàn Kiếm Dis...	21.008997	105.825179	1
6	7	21057571	Bùi Thanh	Bình	Nam	26,Vũ Thạnh,Ô Chợ Dừa,Hanoi (Ha Noi),Đống Đa,H...	21.026209	105.827142	0
7	8	21052051	Trần Chí	Bình	Nam	8,Nguyên Hồng,Khu tập thể Bắc Thành Công,Thành...	21.022750	105.811850	1
8	9	21057211	Trần Thị Mỹ	Châu	Nữ	2,Trần Quang Diệu,Ô Chợ Dừa,Đống Đa,Hanoi,Viet...	21.016550	105.825090	0
9	10	21057631	Võ Ngọc	Châu	Nam	1010000,Nguyên Hồng,Thành Công,Ba Đình Distric...	21.022390	105.812565	1
10	11	21086741	Đỗ Thành	Đạt	Nam	116,Thành Công,Khu tập thể Bắc Thành Công,Thàn...	21.020600	105.813980	1

Hình 14. Danh sách đối tượng địa lý đầu vào đã được phân cụm.

Sau đó trực quan hóa lên bản đồ:

Hình 15. Kết quả của nhóm đạt được sau khi phân cụm.

Bước cuối cùng là chúng tôi sẽ sử dụng thuật toán bao lồi (Convex hull) để bao quanh các cụm lại và đánh dấu bến xe buýt.



Hình 16. Kết quả của nhóm đạt được sau khi bao lồi.

PHẦN 4: KẾT LUẬN – ĐỀ NGHỊ

Mô hình sử dụng thuật toán phân cụm để đề xuất vị trí trạm đưa đón nhân viên với độ chính xác cao trong thực tế đã phần nào giúp giảm tình trạng kẹt xe ở những vùng đô thị đông đúc vì số lượng nhân viên trong một công ty rất lớn, đồng thời giúp giảm chi phí đi lại của mỗi nhân viên so với việc đi bằng phương tiện cá nhân. Song với đó, lượng khí thải từ các phương tiện cá nhân thải ra môi trường được giảm xuống. Đồng thời, sử dụng mô hình này sẽ tính toán được vị trí lắp đặt trạm tối ưu nhất, giúp giảm tình trạng đến công ty muộn, từ đó thúc đẩy năng suất công việc tăng cao. Hiện nay với sự phát triển nhanh chóng của khoa học – công nghệ, thông qua quá trình nghiên cứu người dùng cũng sẽ dễ dàng tiếp cận cho việc lưu thông của mình theo ý muốn. Như kết quả nghiên cứu trên sẽ là bước đệm tiếp theo cho việc đi sâu hơn và phát triển trong lĩnh vực giao thông vận tải hiện nay. Nhằm cải tiến chất lượng dịch vụ, chúng tôi dự định sẽ tạo ra một sản phẩm web, app dựa trên bài nghiên cứu này để người dùng có trải nghiệm tốt hơn với một giao diện đẹp mắt, sáng tạo và các dịch vụ hỗ trợ người dùng có hiệu quả. Mở rộng đối tượng sử dụng là những người đi làm, học sinh, sinh viên, …Và sau đó chúng tôi sẽ cải tiến và thêm một số tính năng như: Dự đoán số cụm cần thiết với dữ liệu đầu vào là danh sách các địa chỉ, tính khoảng cách dựa trên đường đi thực tế, tối ưu thời gian biến đổi địa chỉ về kinh độ, vĩ độ giúp ứng dụng hoạt động hiệu quả hơn, tối thiểu số lần gọi API để tiết kiệm chi phí, có tính năng chỉ đường, tính khoảng cách giữa xe buýt và trạm xe theo thời gian thực để người dùng chủ động thời gian trong việc chờ xe buýt, tài xế xe buýt xác định được con đường đi đến trạm một cách dễ dàng và nhanh nhất. Đó cũng chính là những dự kiến nghiên cứu của nhóm chúng tôi trong tương lai.

Danh mục tài liệu tham khảo

[1] Robert Half. 2018. *Nearly One-Quarter Of Workers Have Left A Job Due To A Bad Commute, According To Robert Half Survey,* xem 30.4.2022. <https://press.roberthalf.com/2018-09-24-Nearly-One-Quarter-Of-Workers-Have-Left-A-Job-Due-To-A-Bad-Commute-According-To-Robert-Half-Survey>.

[2] Phạm Trung Hiền, Trần Sơn Bách, Lê Minh Sơn, Nguyễn Thị Hạnh, Ngọ Xuân Quảng, Phạm Thanh Trà. 2020. *Samsung Việt Nam chăm sóc nhân viên thế nào?,* xem 30.4.2022, <https://www.vietnamplus.vn/samsung/tong-hop-cac-hoat-dong-phuc-loi-cho-nhan-vien-Samsung-VN.html>.

[3] Transdev, *Corporate and industrial transport,* xem 30.4.2022, https://www.transdev.com.au/our-solutions/our-worldwide-solutions/corporate-institutional-and-industrial-site-transport/.

[4] GeoPy, 2016, *Welcome to GeoPy's documentation!,* xem 30.4.2022, <https://geopy.readthedocs.io/en/latest/#nominatim>.

[5] Wikipedia, *haversine formula,* xem 30.4.2022, <https://en.wikipedia.org/wiki/Haversine_formula>.

[6] GeeksforGeeks, 2022, *Haversine formula to find distance between two points on a sphere,* xem 30.4.2022, <https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/>.

[7] Wikipedia, *k-means clustering,* xem 30.4.2022, <https://en.wikipedia.org/wiki/K-means_clustering>.

[8] Bùi Anh Kiệt, *Thuật toán K-means và ứng dụng trong thực tế, báo cáo môn khai phá dữ liệu và kho dữ liệu.*

[9] Hans-Hermann Bock, *Clustering Methods: A History of k-Means Algorithms*, tạp chí Springer Link.

[10] Wikipedia, *Convex-Hull,* xem 30.4.2022, <http://wcipeg.com/wiki/Convex_hull>.

[11] GeeksforGeeks, 2022, *Orientation of 3 ordered points,* xem 30.4.2022, <https://www.geeksforgeeks.org/orientation-3-ordered-points/>.

[12] Wikipedia, *Bao lồi,* xem 30.4.2022, <https://vi.wikipedia.org/wiki/Bao_l%E1%BB%93i>.

[13] Wikipedia, *K-medoids,* xem 30.4.2022, <https://en.wikipedia.org/wiki/K-medoids>.