ỦY BAN NHÂN DÂN TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KỲ
KHAI THÁC DỮ LIỆU

Tên Đề Tài:
Ứng Dụng Khai Phá Dữ Liệu để Phân Cụm Công Việc Theo Đặc Tính Kỹ Năng,Tiêu Đề,Vị Trí và Địa Điểm

Giảng Viên Hướng Dẫn:
Thầy Nguyễn Thanh Phước

Sinh viên thực hiện
3122410004 - Nguyễn Văn An

TP. Hồ Chí Minh, Tháng 12/2024

LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cám ơn các thầy cô trong khoa Công nghệ thông tin của trường đại học Sài Gòn, những người đã trực tiếp giảng dạy cung cấp kiến thức và phương pháp trong những năm qua, đó là những nền tảng cơ bản, là những hành trang vô cùng quý giá để em có thể bước vào sự nghiệp trong tương lai. Để có được kết quả này em xin đặc biệt gửi lời cảm ơn chân thành nhất tới thầy Nguyễn Thanh Phước đã quan tâm giúp đỡ hướng dẫn em hoàn thành một cách tốt nhất đồ án ngành trong thời gian qua. Trong quá trình hoàn thành đồ án, vì chưa có kinh nghiêm thực tế chỉ dựa vào lý thuyết đã học, cùng với thời gian có hạn nên đồ án sẽ không tránh khỏi những sai sót. Kính mong nhận được sự góp ý, nhận xét từ các thầy để kiến thức của em ngày càng hoàn thiện hơn và rút ra được nhiều kinh nghiệm bổ ích có thể áp dụng vào thực tiễn một cách hiệu quả trong tương lai. Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, Tháng 5 năm 2025

Mục lục

DANH MỤC HÌNH ẢNH

Danh sách bảng

CHƯƠNG I: GIỚI THIỆU

1.1.GIỚI THIỆU VỀ BÀI TOÁN

- Trong thời đại bùng nổ dữ liệu, lượng thông tin về người dùng và việc làm ngày càng trở nên phong phú. Tuy nhiên, thực tế cho thấy vẫn còn rất nhiều công việc chưa tìm được ứng viên phù hợp, và ngược lại, nhiều người lao động cũng chưa tìm được công việc tương xứng với năng lực và sở thích của mình. Điều này gây ra tình trạng lãng phí nguồn lực và ảnh hưởng tiêu cực đến sự phát triển xã hội.
- Một trong những kỹ thuật quan trọng trong khai phá dữ liệu nhằm hỗ trợ giải quyết vấn đề này là phân cụm (clustering), phương pháp cho phép nhóm các đối tượng (chẳng hạn như ứng viên hoặc công việc) có đặc điểm tương đồng vào cùng một cụm mà không cần đến nhãn (label) có sẵn. Việc áp dụng phân cụm vào phân tích dữ liệu việc làm sẽ giúp khám phá ra các nhóm công việc có yêu cầu kỹ năng tương tự nhau, cũng như tiêu đề ,vị trí và địa điểm tương đồng nhau. Kết quả phân cụm có thể hỗ trợ nhiều bài toán thực tế như: tìm hiểu đặc điểm từng nhóm đối tượng, tối ưu hóa quá trình tuyển dụng, xây dựng lộ trình đào tạo phù hợp, và là nền tảng cho các hệ thống khuyến nghị trong các bước tiếp theo.

1.2.MỤC TIÊU GIẢI QUYẾT BÀI TOÁN

- Trong báo cáo này, dữ liệu được thu thập thông qua kỹ thuật web scraping từ nền tảng LinkedIn, một mạng xã hội chuyên biệt trong lĩnh vực nghề nghiệp. LinkedIn mang lại nhiều lợi ích cho người dùng, đặc biệt là trong việc tìm kiếm cơ hội việc làm, tuyển dụng, và xây dựng mạng lưới kết nối nghề nghiệp. Người dùng có thể chia sẻ hồ sơ cá nhân, kinh nghiệm làm việc, và kết nối với các nhà tuyển dụng hoặc các chuyên gia trong cùng lĩnh vực.
- Dữ liệu thu thập bao gồm thông tin về các công việc và kỹ năng yêu cầu tương ứng. Mục tiêu của bài toán là ứng dụng kỹ thuật phân cụm để nhóm các công việc có yêu cầu kỹ năng, tiêu đề, vị trí và địa điểm tương đồng lại với nhau. Đây là bước nền tảng ban đầu cho việc xây dựng hệ thống khuyến nghị việc làm trong các nghiên cứu tiếp theo.

1.3.PHẠM VI CỦA ĐỀ TÀI

- Đề tài tập trung vào việc ứng dụng kỹ thuật phân cụm không giám sát để nhóm các công việc có yêu cầu kỹ năng, tiêu đề, vị trí và địa điểm tương đồng dựa trên dữ liệu được thu thập từ nền tảng LinkedIn. Các bước chính bao gồm: tiền xử lý dữ liệu văn bản, biểu diễn kỹ năng công việc dưới dạng vector số, và phân cụm các công việc sử dụng các thuật toán MiniBatch-Kmeans và DBSCAN.
- Do bộ dữ liệu gốc có kích thước lớn với khoảng 1,3 triệu bản ghi, để đảm bảo hiệu quả xử lý và phù hợp với giới hạn tài nguyên tính toán trong khuôn khổ đề tài, em đã lấy mẫu ngẫu nhiên 50.000 bản ghi từ tập dữ liệu ban đầu để tiến hành tiền xử lý, phân tích và xây dựng mô hình. Việc chọn mẫu vẫn đảm bảo tính đại diện vì dữ liệu được lấy một cách ngẫu nhiên và giữ nguyên phân bố tổng thể.
- Phạm vi đề tài không bao gồm việc triển khai hệ thống khuyến nghị hoàn chỉnh, mà chỉ dừng lại ở việc khám phá cấu trúc dữ liệu công việc thông qua phân cụm. Kết quả này đóng vai trò như một tiền đề cho việc phát triển các hệ thống hỗ trợ tuyển dụng hoặc gợi ý việc làm trong tương lai.

1.4.CẤU TRÚC CỦA BÀI BÁO CÁO

- Cấu trúc báo cáo được tổ chức như sau:

- **CHƯƠNG I: GIỚI THIỆU** - Chương này giới thiệu về đề tài , bài toán giải quyết, mục tiêu và phạm vi đề tài
- **CHƯƠNG II: MÔ TẢ VỀ BỘ DỮ LIỆU** - Chương này mô tả về nguồn gốc , kích thước , số lượng mẫu và thuộc tính, các kiểu dữ liệu,... của bộ dữ liệu sử dụng
- **CHƯƠNG III: TIỀN XỬ LÍ DỮ LIỆU** - Chương này mô tả về các bước tiền xử lí dữ liệu
- **CHƯƠNG IV: KHAI PHÁ DỮ LIỆU** - Chương này mô tả về quá trình khai phá dữ liệu
- **CHƯƠNG V: ĐÁNH GIÁ VÀ CHỌN THUẬT TOÁN**- Chương này nêu các yêu cầu về chương trình và đề xuất các thuật toán phù hợp để khai phá dữ liệu, các bước cài đặt thuật toán được đề xuất, ...
- **CHƯƠNG VI: KẾT QUẢ VÀ THẢO LUẬN** - Chương này đánh giá kết quả của các thuật toán được áp dụng trong quá trình khai phá dữ liệu.
- **CHƯƠNG VII: KẾT LUẬN** - Chương này trình bày các kết quả của quá trình khai phá dữ liệu của từng mô hình và so sánh chỉ số đánh giá này giữa các mô hình được huấn luyện. Từ đó chọn thuật toán tốt nhất để giải quyết các vấn đề được đặt ra.
- **TÀI LIỆU THAM KHẢO** - Liệt kê các tài liệu đã sử dụng để tham khảo trong quá trình thực hiện báo cáo.

CHƯƠNG II: MÔ TẢ VỀ BỘ DỮ LIỆU

2.1.NGUỒN GỐC CỦA BỘ DỮ LIỆU SỬ DỤNG

- Trong báo cáo này, bộ dữ liệu "1.3M LinkedIn Jobs & Skills (2024)" được sử dụng. Đây là bộ dữ liệu được thu thập thông qua kỹ thuật web scraping từ nền tảng nghề nghiệp nổi tiếng LinkedIn, bao gồm hơn 1,3 triệu tin tuyển dụng cùng với thông tin chi tiết về các kỹ năng yêu cầu cho từng công việc.
- Bộ dữ liệu được công bố nhằm phục vụ cho nhiều bài toán thực tiễn như: phân tích thị trường lao động, xây dựng bản đồ kỹ năng, và phát triển hệ thống gợi ý việc làm. Ngoài ra, nó cũng hỗ trợ một số tác vụ như:

- **Thực hiện tiền xử lý và làm sạch dữ liệu thô.**
- **Phân tích các chức danh công việc hoặc ngành nghề đang có nhu cầu cao nhất ở các thành phố hoặc quốc gia khác nhau.**
- **Xác định các công ty hàng đầu đang tuyển dụng cho các vị trí cụ thể**
- **Khai thác dữ liệu kỹ năng để xác định những kỹ năng được yêu cầu nhiều nhất trong từng nhóm công việc.**
- **Xây dựng hệ thống gợi ý việc làm dựa trên hồ sơ người dùng và dữ liệu tin tuyển dụng**
- **Xác định khoảng cách kỹ năng trên thị trường lao động để phục vụ xây dựng các chương trình giáo dục hoặc đào tạo phù hợp**

2.2.KÍCH THƯỚC BỘ DỮ LIỆU

- Bộ dữ liệu ban đầu bao gồm 3 file csv riêng biệt cụ thể là:

- **job_skills.csv** : là bộ dữ liệu gồm khoảng 1.3 triệu bản ghi chứa các kỹ năng (job_skills) được trích xuất từ phần mô tả công việc (job_summary) bằng kỹ thuật NER (Named Entity Recognition). Cụ thể là file job_skills.csv có 2 thuộc tính như sau:
  - **job_link** : Đây là liên kết (URL) đến bài đăng công việc cụ thể trên LinkedIn và là FOREIGN KEY (khóa ngoại) dùng để liên kết với các bảng dữ liệu khác
  - **job_summary.csv** : là bộ dữ liệu này bao gồm phần mô tả công việc (job_summary) được lấy từ LinkedIn...
- **job_summary.csv** : là bộ dữ liệu này bao gồm phần mô tả công việc (job_summary) được lấy từ LinkedIn. Cụ thể là file job_summary.csv có 2 thuộc tính như sau:
  - **job_link** : Đây là liên kết (URL) đến bài đăng công việc cụ thể trên LinkedIn và là FOREIGN KEY (khóa ngoại) dùng để liên kết với các bảng dữ liệu khác
  - **job_summary (kiểu str)** : là chuỗi văn bản mô tả công việc nội dung quan trọng giúp người lao động hiểu công việc đó yêu cầu gì.

- **linkedin_job_postings.csv** : là bộ dữ liệu được sử dụng trong bài toán là một tập dữ liệu lớn, bao gồm khoảng 1,3 triệu bài đăng tuyển dụng được thu thập từ nền tảng LinkedIn trong năm 2024 thông qua kỹ thuật web scraping. Bao gồm: job_skill, job_links, job_titles, job_location, company,.. Cụ thể là file linkedin_job_postings.csv có 14 thuộc tính như sau:
  - **job_link** : Đây là liên kết (URL) đến bài đăng công việc cụ thể trên LinkedIn và là FOREIGN KEY (khóa ngoại) dùng để liên kết với các bảng dữ liệu khác
  - **last_processed_time (kiểu datetime)** : Dấu thời gian cho biết lần cuối cùng tin tuyển dụng được xử lý
  - **got_summary (kiểu bool)** : Cho biết hệ thống có trích xuất thành công phần mô tả công việc (job summary) hay không.
  - **job_summary (kiểu str)** : là chuỗi văn bản mô tả công việc nội dung quan trọng giúp người lao động hiểu công việc đó yêu cầu gì.
  - **got_ner (kiểu bool)** : Cho biết hệ thống có thực hiện phân tích NER (Named Entity Recognition) để trích xuất kỹ năng từ bài đăng hay không.
  - **is_being_worked (kiểu bool)** : Cho biết bài đăng này hiện có đang được xử lý bởi hệ thống hay không (hữu ích trong các hệ thống xử lý song song).
  - **job_title (kiểu str)** : Tiêu đề (chức danh) công việc được tuyển dụng.
  - **company (kiểu str)** : Tên công ty đăng tuyển công việc.
  - **job_location (kiểu str)** : Địa điểm làm việc.
  - **first_seen (kiểu datetime)** : Thời điểm hệ thống lần đầu phát hiện bài đăng này.
  - **search_city (kiểu str)** : Thành phố được sử dụng làm tiêu chí tìm kiếm khi thu thập dữ liệu.
  - **search_country (kiểu str)** : Quốc gia được sử dụng làm tiêu chí tìm kiếm khi thu thập bài đăng công việc. Cột này giúp phân loại các công việc theo quốc gia và có thể hỗ trợ phân tích xu hướng tuyển dụng theo khu vực.
  - **search_position (kiểu str)** : Chức danh (vị trí) công việc được sử dụng làm tiêu chí tìm kiếm trong quá trình thu thập.
  - **job_level (kiểu str)** : Trình độ/cấp bậc của vị trí công việc.
  - **job_type (kiểu str)** : Loại hình công việc.

2.3.CÁC KIỂU DỮ LIỆU

- Bộ dữ liệu sử dụng trong đề tài gồm ba tệp CSV, trong đó hai tệp quan trọng là **linkedin_job_postings.csv** và **job_skills.csv**. Hai tệp này được gộp lại bằng thao tác nối (merge) trên khóa job_link, nhằm tạo ra một bảng dữ liệu đầy đủ kết hợp cả thông tin công việc và kỹ năng yêu cầu. **Phần này sẽ được giải thích kĩ ở tiền xử lí dữ liệu**
- Sau khi gộp, dữ liệu có nhiều cột được pandas nhận diện là object, bao gồm cả các cột thực tế thuộc kiểu bool hoặc datetime như hình 1 dưới đây.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1296381 entries, 0 to 1296380
Data columns (total 15 columns):
 #   Column               Non-Null Count    Dtype
---  ------               --------------    -----
 0   job_link             1296381 non-null  object
 1   job_skills           1294296 non-null  object
 2   last_processed_time  1296381 non-null  object
 3   got_summary          1296381 non-null  object
 4   got_ner              1296381 non-null  object
 5   is_being_worked      1296381 non-null  object
 6   job_title            1296381 non-null  object
 7   company              1296372 non-null  object
 8   job_location         1296362 non-null  object
 9   first_seen           1296381 non-null  object
 10  search_city          1296381 non-null  object
 11  search_country       1296381 non-null  object
 12  search_position      1296381 non-null  object
 13  job_level            1296381 non-null  object
 14  job_type             1296381 non-null  object
dtypes: object(15)
memory usage: 148.4+ MB
```

Hình 1: Thông tin tổng quan về dataset sau khi gộp.

2.4.DỮ LIỆU BỊ THIẾU

- Ở đây bộ dataset sau khi gộp **linkedin_job_postings.csv** và **job_skills.csv**. Em quan sát thấy có tất cả 1296381 mẫu . Đặc biệt ở 3 cột job_skills, company và job_location có dữ liệu bị thiếu

CHƯƠNG III:TIỀN XỬ LÍ DỮ LIỆU

3.1.GỘP DỮ LIỆU TỪ 2 FILE

- 2 file dữ liệu ban đầu là job_skills.csv và linkedin_job_postings.csv. được chia thành hai bảng riêng: df chứa job_skills.csv và df2 chứa linkedin_job_postings.csv, mỗi bảng chứa các thuộc tính khác nhau nhưng cùng mô tả về các bài đăng tuyển dụng trên LinkedIn. Để tạo thành một tập dữ liệu hoàn chỉnh, em tiến hành gộp hai bảng này bằng phép inner join dựa trên cột khóa chung job_link, sử dụng hàm pd.merge() của thư viện pandas.
- Vì sao lại làm dùng inner join? Việc sử dụng inner join giúp đảm bảo chỉ những bài đăng có mặt ở cả hai bảng mới được giữ lại, từ đó loại bỏ các bản ghi không đầy đủ hoặc không đồng bộ. Sau bước này, tập dữ liệu thu được là bảng linkedin_job_posting, được sử dụng cho các bước tiền xử lý tiếp theo.

3.2.XỬ LÍ DỮ LIỆU BỊ THIẾU

- Với dataset sau khi gộp bị thiếu dữ liệu ở 3 cột là job_skills, company và job_location. Em quyết định loại bỏ các mẫu có dữ liệu bị thiếu , chỉ giữ lại các mẫu đầy đủ dữ liệu.

3.3. LỌC VÀ LẤY MẪU

- Khó khăn gặp phải khi bộ dữ liệu quá lớn dù đã gộp hay loại bỏ dữ liệu bị thiếu nhưng vẫn còn quá lớn, khiến cho việc xử lí, phân tích hay huấn luyện mô hình phân cụm gặp rất nhiều khó khăn. Khi tiến hành xử lý trên môi trường Google Colab với tài nguyên hạn chế, hệ thống thường xuyên gặp tình trạng quá tải CPU, dẫn đến việc huấn luyện toàn bộ tập dữ liệu trở nên bất khả thi. Với vấn đề này, em đề xuất một cách phổ biến đó là lấy một số lượng mẫu nhất định từ tập dataset đã gộp (50000 mẫu), để tiếp tục nghiên cứu và khám phá dữ liệu. Nhưng khi quan sát dataset thì xuất hiện rất nhiều ngôn ngữ khác nhau như là Trung Quốc , Hàn Quốc ,... Điều này gây cản trở đáng kể cho quá trình phân tích và mô hình hóa dữ liệu, đặc biệt là trong các bài toán phân cụm..
- Do tập dữ liệu gốc chứa các bài đăng tuyển dụng từ nhiều quốc gia và ngôn ngữ khác nhau có kích thước quá lớn, nên em quyết định lấy mẫu 50,000 dòng dữ liệu có kỹ năng đầu tiên bằng tiếng Anh để đảm bảo tính đồng nhất về ngôn ngữ trong quá trình phân tích và xây dựng mô hình. Vừa tiết kiệm thời gian nhanh gọn vừa đảm bảo được các mẫu được lấy đúng.
- Em sử dụng thư viện langdetect để kiểm tra ngôn ngữ của kỹ năng đầu tiên trong cột job_skills. Những kỹ năng viết bằng tiếng Anh hoặc là từ viết tắt phổ biến sẽ được giữ lại. Dữ liệu được duyệt tuần tự và thu thập đến khi đủ 50,000 dòng thỏa mãn điều kiện về ngôn ngữ. Việc này giúp tiết kiệm đáng kể chi phí xử lý, rút ngắn thời gian huấn luyện, đồng thời đảm bảo tập dữ liệu đầu vào đủ sạch và nhất quán để tiến hành các bước phân tích tiếp theo.
- Quá trình được thực hiện bằng vòng lặp có kiểm soát với thư viện tqdm để theo

dõi tiến trình. Sau khi hoàn tất, tập mẫu được lưu vào để phục vụ các bước xử lý tiếp theo.

3.4. CHUẨN HÓA VÀ MÃ HÓA CÁC THUỘC TÍNH

- Sau khi lọc và lấy mẫu, tập dữ liệu vẫn chứa nhiều thuộc tính dạng phân loại (categorical) có kiểu dữ liệu object. Để đảm bảo mô hình có thể xử lý các biến này hiệu quả,em tiến hành mã hóa các biến phân loại thành dạng số bằng kỹ thuật Label Encoding thông qua thư viện sklearn.preprocessing.LabelEncoder.
- Cụ thể, em xác định các cột có kiểu dữ liệu là object, sau đó áp dụng LabelEncoder để biến đổi từng cột về dạng số nguyên. Quá trình này giúp đảm bảo rằng toàn bộ tập dữ liệu đều ở dạng số học (numerical), phù hợp với yêu cầu của các thuật toán phân tích và học máy.
- Ngoài ra, một số cột không phục vụ trực tiếp cho mô hình phân cụm như got_summary, got_ner, is_being_worked, search_city, search_country, và cuối cùng là search_position đã được loại bỏ để giảm nhiễu và tăng tính tập trung vào các đặc trưng quan trọng hơn.
- Cuối cùng, để chuẩn hóa dữ liệu và loại bỏ sự khác biệt về thang đo giữa các thuộc tính, em sử dụng StandardScaler để đưa toàn bộ các đặc trưng về trung bình 0 và độ lệch chuẩn 1. Em chọn 4 đặc trưng là kỹ năng công việc , tiêu đề , vị trí và địa điểm công việc để làm đầu vào chính thức cho mô hình phân cụm . Dưới đây là bảng dataframe được chuẩn bị cho mô hình phân cụm.

Đây là dataframe được sử dụng cho model

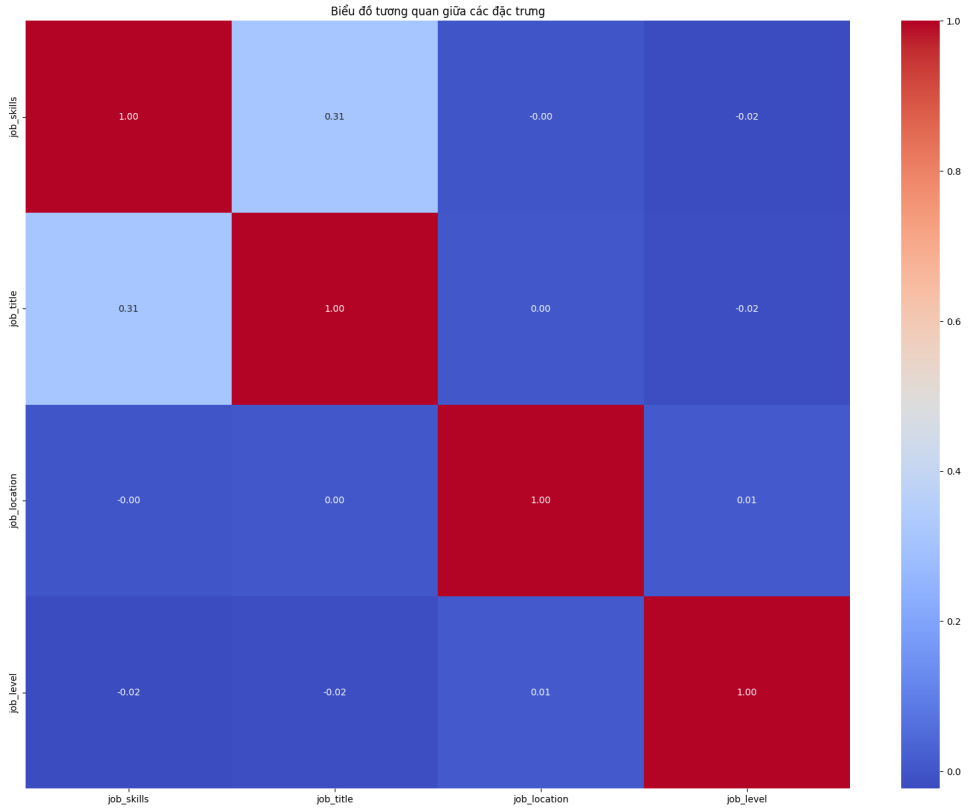
	job_skills	job_title	job_location	job_level
0	-1.351778	-0.226998	0.254600	0.477765
1	-0.362655	-1.447806	-1.577203	0.477765
2	-1.494143	0.992020	-1.401551	0.477765
3	-0.143235	-0.692425	0.585638	0.477765
4	-0.159424	-0.693739	0.761290	0.477765

Hình 2: Thông tin dataframe được chuẩn bị cho phân cụm.

3.5.ĐỘ TƯƠNG QUAN CỦA CÁC THUỘC TÍNH

- Ta sẽ tìm hiểu xem các thuộc tính sau khi được chuẩn hóa và mã hóa có mối tương quan như thế nào với nhau, sau đó bằng hiểu biết về sự tương quan ta sẽ chọn có nên giảm chiều dữ liệu hay không. Hình dưới đây mô tả Correlation matrix biểu thị độ tương quan giữa các thuộc tính. Không có cặp đặc trưng nào có hệ số tương quan tuyệt đối gần 1. Hầu hết các thuộc tính đều có tương quan rất thấp

hoặc gần bằng 0 cho thấy ít bị trùng lặp thông tin giữa các đặc trưng, nên ta sẽ không cần phải giảm chiều dữ liệu vì việc giảm chiều lúc này có thể làm mất thông tin thay vì tối ưu hóa mô hình.



Hình 3: Mô tả Correlation matrix biểu thị độ tương quan giữa các thuộc tính.

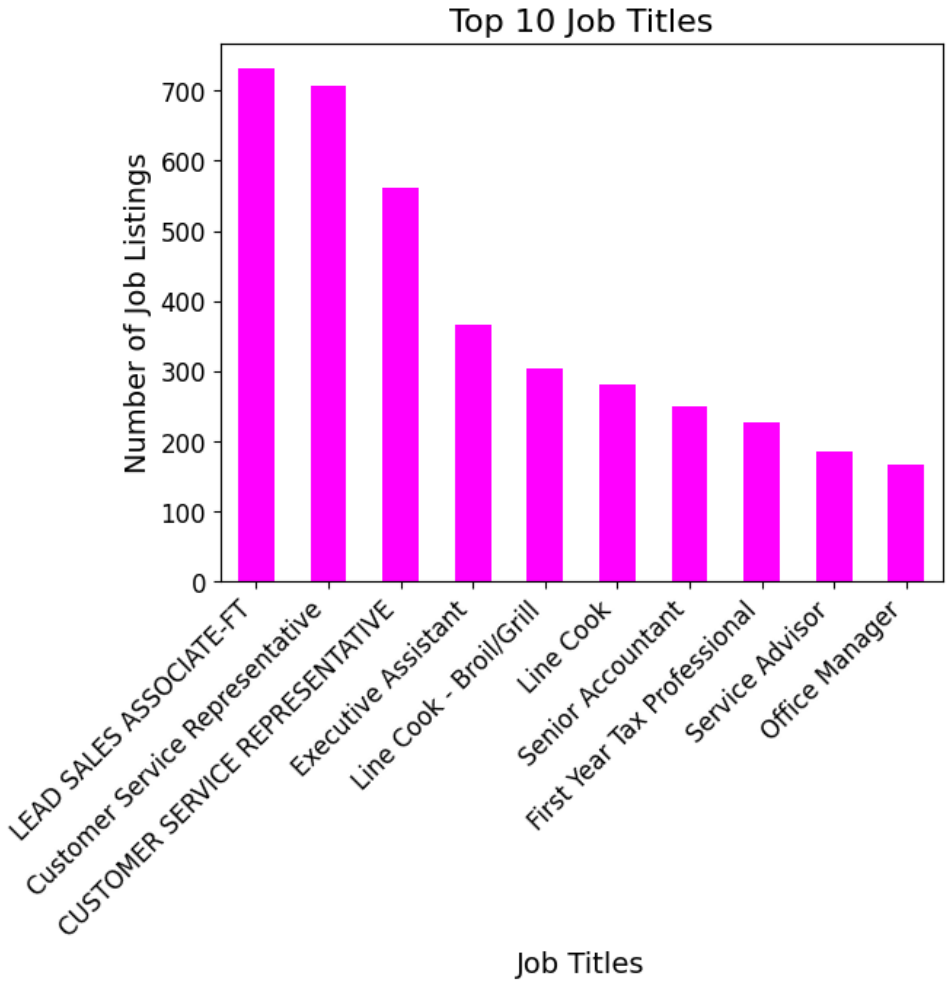
3.6. LOẠI BỎ OUTLIERS

- Loại bỏ outlier sử dụng phương pháp dựa trên Interquartile Range (IQR). Tính Q1 (quartile 1), Q3 (quartile 3) và IQR (Q3 - Q1), sau đó xác định ngưỡng: Dữ liệu ngoài khoảng [Q1 - 1.5IQR, Q3 + 1.5IQR] được coi là outlier, cuối cùng là lọc dữ liệu để loại bỏ outliers. Cuối cùng bộ dữ liệu bao gồm 40708 bản ghi và 4 thuộc tính.

CHƯƠNG IV:KHAI PHÁ DỮ LIỆU

4.1. TOP 10 DỮ LIỆU PHỔ BIẾN CỦA TỪNG ĐẶC TRƯNG

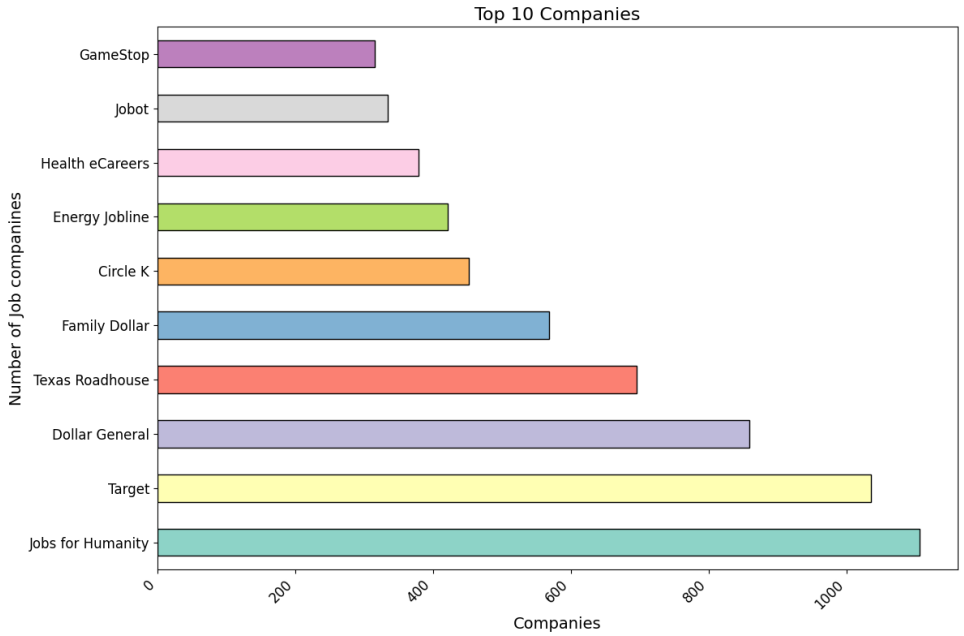
- Trước khi tiến hành phân cụm, việcquan sát các thống kê mô tả của 10 giá trị phổ biến nhất trong từng thuộc tính giúp chúng ta hiểu rõ hơn về đặc điểm của dữ liệu. Những thông tin này cung cấp cái nhìn tổng quan về phạm vi và xu hướng phân bố, đồng thời hỗ trợ việc phát hiện các giá trị bất thường có thể ảnh hưởng đến kết quả phân cụm.



Hình 4: Biểu đồ mô tả top 10 tiêu đề công việc phổ biến trong tập.

- Vị trí "Lead Sales Associate-FT"đứng đầu với hơn 700 công việc, cho thấy đây là vị trí phổ biến nhất. Tiếp theo là "Customer Service Representative"với khoảng 600 công việc. Các vị trí khác như "Executive Assistant", "Line Cook - Broil/Grill", "First Line Cook", "Senior Accountant", "Tax Professional", "Service Advisor", và "Office Manager"có số lượng giảm dần từ khoảng 400 xuống dưới 200. Biểu đồ cho thấy sự chênh lệch rõ rệt: 2 vị trí đầu có số lượng vượt trội, trong khi các vị trí còn lại giảm dần đều.
- Dữ liệu không đồng đều. "Lead Sales Associate-FT"và "Customer Service Representative"chiếm ưu thế lớn, trong khi các vị trí khác có số lượng ít hơn nhiều.

Các vị trí phổ biến nhất thuộc lĩnh vực bán hàng (Sales) và dịch vụ khách hàng (Customer Service), cho thấy đây là các ngành nghề có nhu cầu cao. Các vị trí khác như "Senior Accountant", "Tax Professional"thuộc lĩnh vực tài chính, còn "Line Cook"liên quan đến nhà hàng.
- Chính vì vậy , em chọn "job_titles"làm đặc trưng để huấn luyện phân cụm. Vì "job_title"phản ánh rõ ràng đặc điểm nghề nghiệp trong tập dữ liệu, là một đặc trưng quan trọng để phân cụm phân nhóm kết hợp với với các đặc trưng khác để phân cụm toàn diện. Dữ liệu cho thấy sự khác biệt rõ rệt giữa các giá trị, giúp thuật toán phân cụm (MiniBatch K-Means và DBSCAN) dễ dàng nhận diện các nhóm.



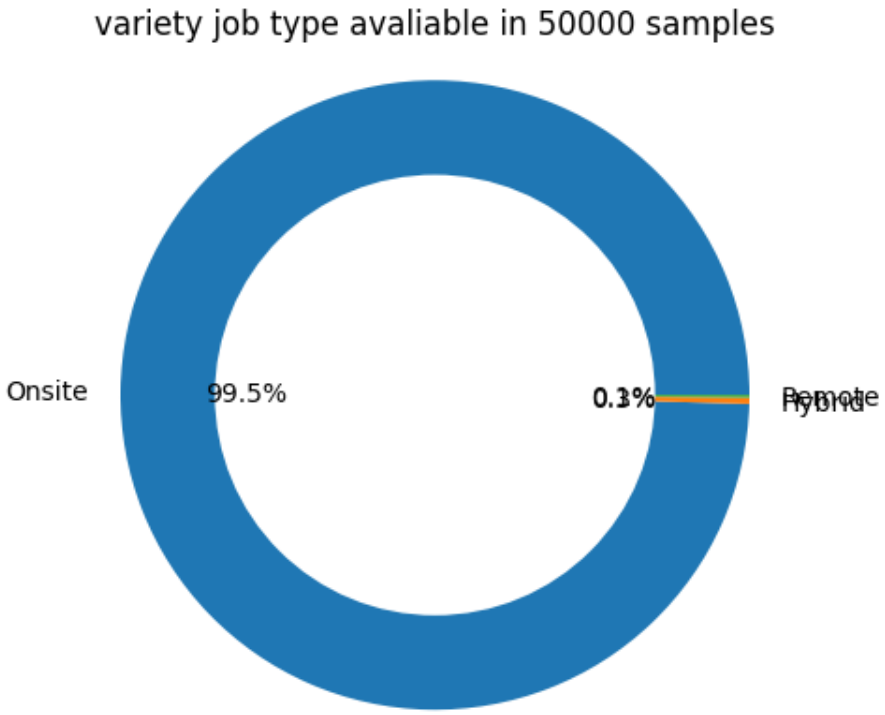
Hình 5: Biểu đồ mô tả top 10 công ty tuyển phổ biến trong tập.

- Biểu đồ này hiển thị top 10 công ty (Companies) có số lượng công việc (Number of Job Listings) cao nhất. Về phân bố trên biểu đồ, "Jobs for Humanity"dẫn đầu với số lượng công việc vượt quá 1000, cho thấy đây là công ty có nhu cầu tuyển dụng cao nhất. "Target"đứng thứ hai với khoảng 900 công việc, gần đạt mức 1000. "Dollar General"xếp thứ ba với khoảng 800 công việc. Các công ty khác như "Texas Roadhouse", "Family Dollar", "Circle K", "Energy Jobline", "Health eCareers", "Jobot", và "GameStop"có số lượng giảm dần từ khoảng 600 xuống dưới 400. Biểu đồ cho thấy sự mất cân bằng: 3 công ty đầu có số lượng vượt trội, trong khi các công ty còn lại có số lượng thấp hơn đáng kể.
- Em quyết định không chọn đặc trưng này, vì mục tiêu phân cụm không liên quan đến công ty , "company"không cần thiết và làm phức tạp dữ liệu.

4.2. SỰ PHÂN BỐ CỦA DỮ LIỆU

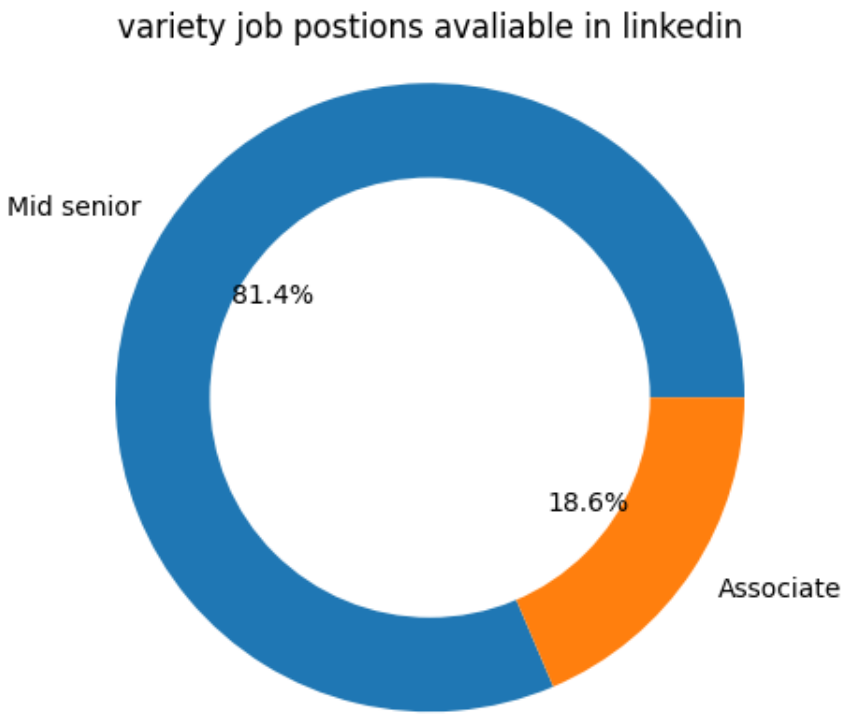
- Việc hiểu rõ sự phân bố của dữ liệu là yếu tố quan trọng trong bài toán phân cụm. Khi dữ liệu phân bố thành các nhóm rõ ràng, ta có thể ước lượng sơ bộ số lượng cụm cần tìm. Bên cạnh đó, việc quan sát phân bố còn giúp phát hiện dữ liệu bị lệch, mất cân bằng hoặc có nhiễu, từ đó đưa ra các phương pháp xử lý phù hợp.

Đặc biệt, sự hiểu biết này đóng vai trò định hướng trong việc lựa chọn thuật toán phân cụm phù hợp và hỗ trợ quá trình tiền xử lý diễn ra nhanh chóng, hiệu quả hơn.



Hình 6: Biểu đồ mô tả phân bố về các loại hình làm việc.

- Biểu đồ này là một biểu đồ dạng donut (hoặc pie chart) thể hiện phân bố các loại công việc (job types) trong một mẫu dữ liệu gồm 50,000 mẫu. Dữ liệu cột job_type có hai loại chính: "Onsite", "Remote"và "Hybrid". Onsite: Chiếm 99.5% (tương đương với khoảng 49,750 mẫu trong tổng số 50,000). Chỉ chiếm 0.3% (tương đương 161 mẫu), với phần còn lại Remote: (0.2%) chứa 65 mẫu. Sự chênh lệch rất lớn giữa "Onsite"với "Hybrid"và "Remote"cho thấy tập dữ liệu này gần như hoàn toàn bao gồm các công việc làm tại chỗ (onsite).
- Em quyết định không chọn thuộc tính làm đặc trưng đầu vào cho phân cụm. Vì Với 99.5% là "Onsite", "Hybrid"(0.3%) và "Remote"(0.2%) gần như không đáng kể, dễ khiến MiniBatch K-Means tạo ra một cụm lớn "Onsite"và bỏ qua "Hybrid"/"Remote". Còn DBSCAN coi "Hybrid"và "Remote"là nhiễu do mật độ thấp.



Hình 7: Biểu đồ mô tả phân bố về vị trí công việc.

- Biểu đồ này là một biểu đồ dạng donut (hoặc pie chart) thể hiện phân bố các cấp bậc công việc (job positions) được lấy từ đặc trưng job_level. Phân tích về biểu đồ: Mid senior: Chiếm 81.4% (tương ứng với phần lớn các vị trí công việc). Associate: Chiếm 18.6% (tương ứng với phần còn lại). Tổng cộng là 100%, cho thấy dữ liệu bao quát toàn bộ các cấp bậc công việc trong mẫu được phân tích.
- Mất cân bằng: Dù không quá nghiêm trọng như trường hợp "job_type"trước đó, vẫn có sự mất cân bằng rõ rệt, với "Mid senior"chiếm ưu thế áp đảo (81.4%) so với "Associate"(18.6%). Phân bố này cho thấy xu hướng công việc tập trung vào các vị trí cấp trung và cao hơn ("Mid senior"), trong khi vị trí "Associate"(thường là cấp thấp hơn hoặc mới vào nghề) ít phổ biến hơn.
- Em quyết định chọn đặc trưng này làm đầu vào mô hình phân cụm. Vì "job_position"(Mid senior vs. Associate) phản ánh cấp bậc công việc, là một đặc trưng quan trọng khi mục tiêu là kết hợp phân nhóm theo kỹ năng công việc và địa điểm. Kết hợp với "job_title"hoặc "job_skills", "job_level"có thể giúp phân tích yêu cầu kỹ năng hoặc kinh nghiệm cần thiết cho từng cấp bậc (ví dụ: "Mid senior"có thể yêu cầu kỹ năng phức tạp hơn "Associate"). Với tỷ lệ 81.4% và 18.6 %, sự mất cân bằng không quá lớn như "job_type"(99.5% vs. 0.3%), nên có thể xử lý được trong phân cụm.

CHƯƠNG V:ĐÁNH GIÁ VÀ CHỌN THUẬT TOÁN

5.1. MÔ HÌNH PHÂN CỤM

- Để xây dựng các mô hình phân cụm trong bài toán nhóm các công việc, em đã lựa chọn sử dụng một số thuật toán phân cụm phổ biến như một hướng tiếp cận ban đầu, bao gồm:
+ MiniBatch K-Means: một biến thể của thuật toán K-Means tiêu chuẩn, tối ưu hơn cho các tập dữ liệu lớn do sử dụng mini-batch để giảm thời gian tính toán.
+ DBSCAN (Density-Based Spatial Clustering of Applications with Noise): một thuật toán phân cụm dựa trên mật độ, phù hợp với các tập dữ liệu có hình dạng phân cụm phức tạp và có khả năng xử lý nhiễu tốt.
- Việc huấn luyện các mô hình trên được thực hiện thông qua các hàm triển khai sẵn có trong thư viện Scikit-learn (sklearn) của Python – một thư viện mạnh mẽ và phổ biến trong lĩnh vực học máy và khai phá dữ liệu.

5.2. SƠ LƯỢC VỀ QUY TRÌNH HUẤN LUYỆN VÀ ĐÁNH GIÁ

5.2.1. CHUẨN BỊ DỮ LIỆU

Để chuẩn bị dữ liệu đầu vào cho các mô hình phân cụm, em tiến hành các bước xử lý (với dữ liệu đã được lấy mẫu và lọc) như sau:

- Áp dụng kỹ thuật mã hóa nhãn (Label Encoding) để chuyển đổi các biến dạng chuỗi (object) thành dạng số.
- Loại bỏ một số cột không cần thiết như `got_summary`, `got_ner`, `is_being_worked`, `search_city`, `search_country`, và `search_position` nhằm giảm nhiễu và tập trung vào các đặc trưng quan trọng hơn.
- Chuẩn hóa toàn bộ dữ liệu bằng `StandardScaler` để đưa các giá trị về cùng thang đo, đảm bảo các đặc trưng không bị ảnh hưởng bởi sự chênh lệch về đơn vị đo.
- Cuối cùng là chọn ra 4 đặc trưng quan trọng và cần thiết là `job_skills`, `job_title`, `job_level`, và `job_location` để làm đầu vào cho mô hình phân cụm và lưu vào Data Frame để được sử dụng làm đầu vào.

Sau các bước xử lý trên, em thu được một **DataFrame dạng số đã được chuẩn hóa**, được sử dụng làm đầu vào cho các mô hình phân cụm

5.2.2. XÂY DỰNG MÔ HÌNH

- Các mô hình MiniBatch-Kmeans và DBSCAN được xây dựng bằng cách sử dụng các triển khai từ thư viện Scikit-learn.

5.2.3. ĐÁNH GIÁ MÔ HÌNH

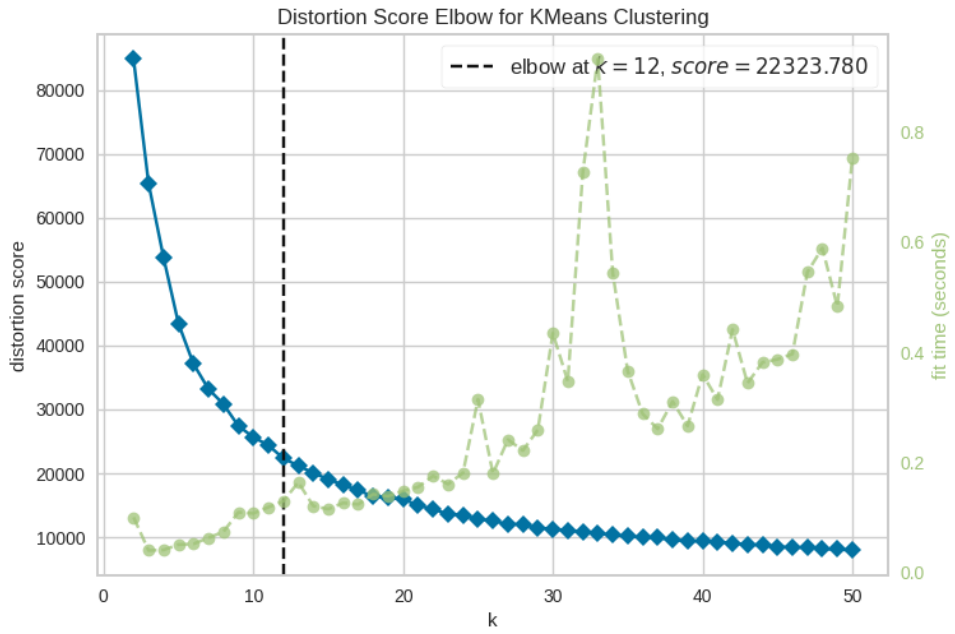
Để đánh giá chất lượng của các mô hình phân cụm, em sử dụng ba chỉ số đánh giá phổ biến như sau:

- **Silhouette Score**: đo độ gắn kết (cohesion) giữa các điểm trong cùng một cụm và độ tách biệt (separation) giữa các cụm khác nhau. Giá trị nằm trong khoảng từ -1 đến 1; càng gần 1 thì cụm càng rõ ràng.
- **Davies-Bouldin Index**: đánh giá mức độ tương đồng giữa các cụm. Giá trị càng nhỏ thể hiện các cụm càng khác biệt và phân tách tốt.
- **Calinski-Harabasz Index (Variance Ratio Criterion)**: đánh giá dựa trên tỷ lệ giữa phương sai giữa các cụm và phương sai trong cụm. Giá trị càng cao càng cho thấy phân cụm hiệu quả.

Các chỉ số này được tính toán cho từng mô hình và được sử dụng để so sánh, đưa ra lựa chọn mô hình phân cụm tối ưu

5.2.4. TINH CHỈNH THAM SỐ

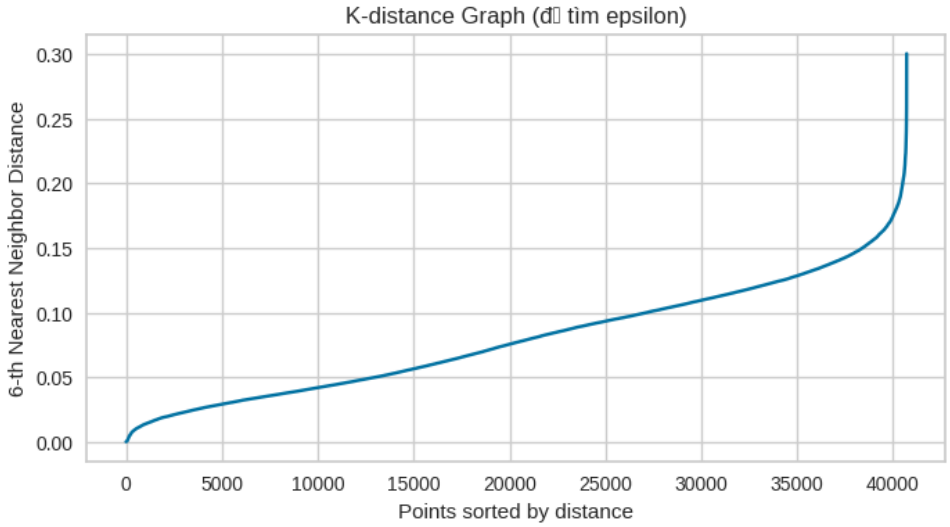
- Đối với **MiniBatchKMeans** (hoặc KMeans), việc tinh chỉnh siêu tham số `n_clusters` (số lượng cụm) là rất quan trọng. Tham số này quyết định số lượng cụm được hình thành trong quá trình phân cụm, ảnh hưởng trực tiếp đến độ hiệu quả của mô hình. Để xác định giá trị tối ưu, em sử dụng **phương pháp Elbow Method** để tìm điểm gãy (elbow point), tại đó sự cải thiện của tổng sai số bình phương nội cụm không còn đáng kể.



Hình 8: Biểu đồ phương pháp ELBOW.

- Đối với **DBSCAN**, hai siêu tham số quan trọng là `eps` (bán kính lân cận) và `min_samples` (số lượng điểm tối thiểu trong vùng lân cận để một điểm được coi là

"core point"). Việc lựa chọn các tham số này ảnh hưởng lớn đến khả năng phát hiện các cụm có hình dạng bất kỳ và xử lý nhiễu. Để chọn giá trị `eps` phù hợp, em sử dụng **biểu đồ k-distance plot** và thử nghiệm nhiều giá trị khác nhau của `min_samples`.



Hình 9: Biểu đồ k-distance plot

- Các bước trên đảm bảo rằng mỗi mô hình được tối ưu hóa và đánh giá một cách cẩn thận, đồng thời đảm bảo rằng các chỉ số đánh giá được sử dụng phù hợp với từng thuật toán phân cụm.

CHƯƠNG VI: KẾT QUẢ VÀ THẢO LUẬN

6.1. CHỈ SỐ ĐÁNH GIÁ

Vì đây là mô hình phân cụm thuộc loại học không giám sát (unsupervised learning), nên không có sẵn nhãn (label) để sử dụng trong quá trình đánh giá. Do đó, việc đánh giá chất lượng của mô hình phân cụm sẽ khác biệt so với các mô hình học có giám sát (supervised learning). Trong báo cáo này, em sử dụng các chỉ số đánh giá nội tại (intrinsic evaluation metrics), bao gồm:

- **Silhouette Score**: đo độ gắn kết (cohesion) giữa các điểm trong cùng một cụm và độ tách biệt (separation) giữa các cụm khác nhau. Chỉ số này có giá trị trong khoảng từ -1 đến 1; càng gần 1 thì các cụm càng rõ ràng và dễ phân biệt.
- **Davies-Bouldin Index**: đánh giá mức độ tương đồng giữa các cụm. Giá trị của chỉ số này càng nhỏ cho thấy các cụm càng tách biệt tốt và không chồng lấn lẫn nhau.
- **Calinski-Harabasz Index (Variance Ratio Criterion)**: dựa trên tỷ lệ giữa phương sai giữa các cụm (inter-cluster variance) và phương sai trong cụm (intra-cluster variance). Giá trị càng lớn chứng tỏ mô hình phân cụm càng hiệu quả.

Các chỉ số này đều không yêu cầu nhãn thật và được sử dụng rộng rãi trong đánh giá mô hình phân cụm.

6.1. DANH SÁCH THAM SỐ

Hầu hết các mô hình phân cụm đều yêu cầu lựa chọn các siêu tham số phù hợp để đạt hiệu quả tối ưu. Cụ thể:

- **Đối với KMeans và MiniBatchKMeans**, tham số quan trọng nhất là số lượng cụm k. Em sử dụng phương pháp **Elbow Method** để xác định giá trị k tối ưu. Bằng cách vẽ biểu đồ biểu diễn tổng sai số bình phương nội cụm (inertia) theo từng giá trị k, điểm gãy (elbow point) được lựa chọn là nơi mà việc tăng thêm số cụm không còn giúp giảm đáng kể sai số.
- **Đối với DBSCAN**, hai siêu tham số quan trọng là `eps` (bán kính lân cận) và `min_samples` (số lượng điểm tối thiểu trong vùng lân cận để một điểm được coi là "core point"). Việc lựa chọn các giá trị phù hợp cho hai tham số này ảnh hưởng trực tiếp đến khả năng phát hiện các cụm có hình dạng bất kỳ cũng như xử lý nhiễu hiệu quả. Em sử dụng **biểu đồ k-distance plot** để xác định giá trị `eps` hợp lý, đồng thời thử nghiệm nhiều giá trị khác nhau cho `min_samples` để tìm được tổ hợp tối ưu.

Tên mô hình	Tham số	Giá trị đã chọn
MiniBatchKMeans	n_clusters	4,5,6,7,8,9,10,11,12,15

Bảng 1: Danh sách các tham số được sử dụng thử nghiệm cho mô hình MiniBatchKmeans

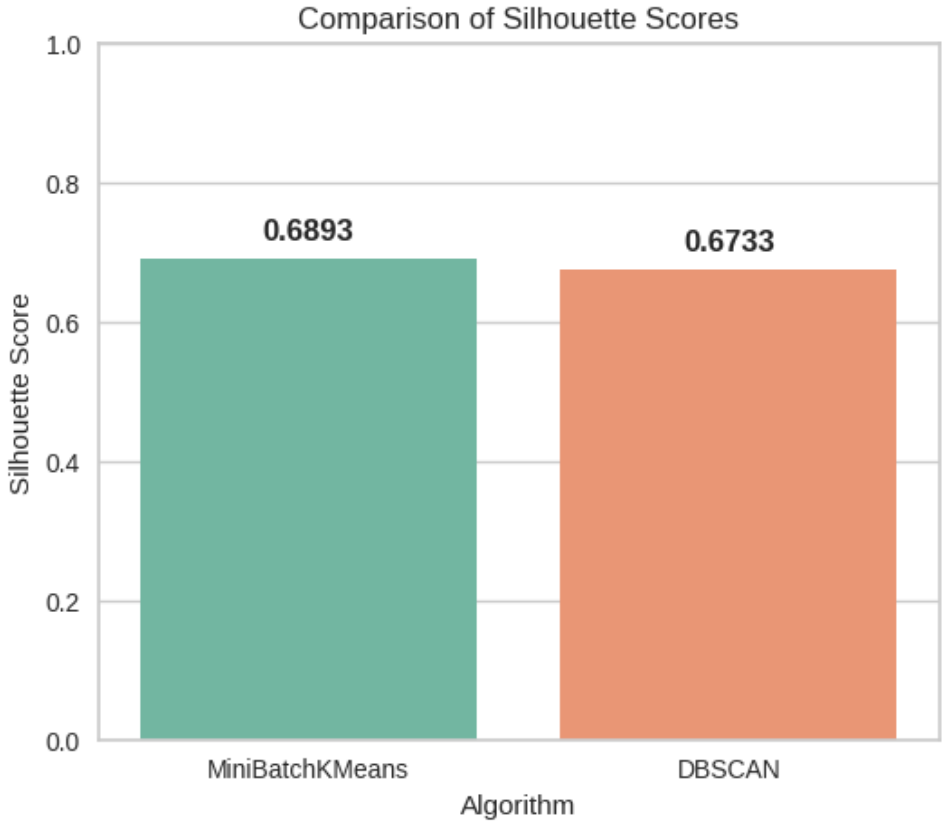
Tên mô hình	**Tham số**	**Giá trị đã chọn**
DBSCAN	eps	0.15, 0.16, 0.17, 0.18, 0.19, 0.20, 0.5, 0.6
DBSCAN	min_samples	4,5,6,7,8

Bảng 2: Danh sách các tham số được sử dụng cho mô hình DBSCAN

CHƯƠNG VII: KẾT QUẢ VÀ KẾT LUẬN

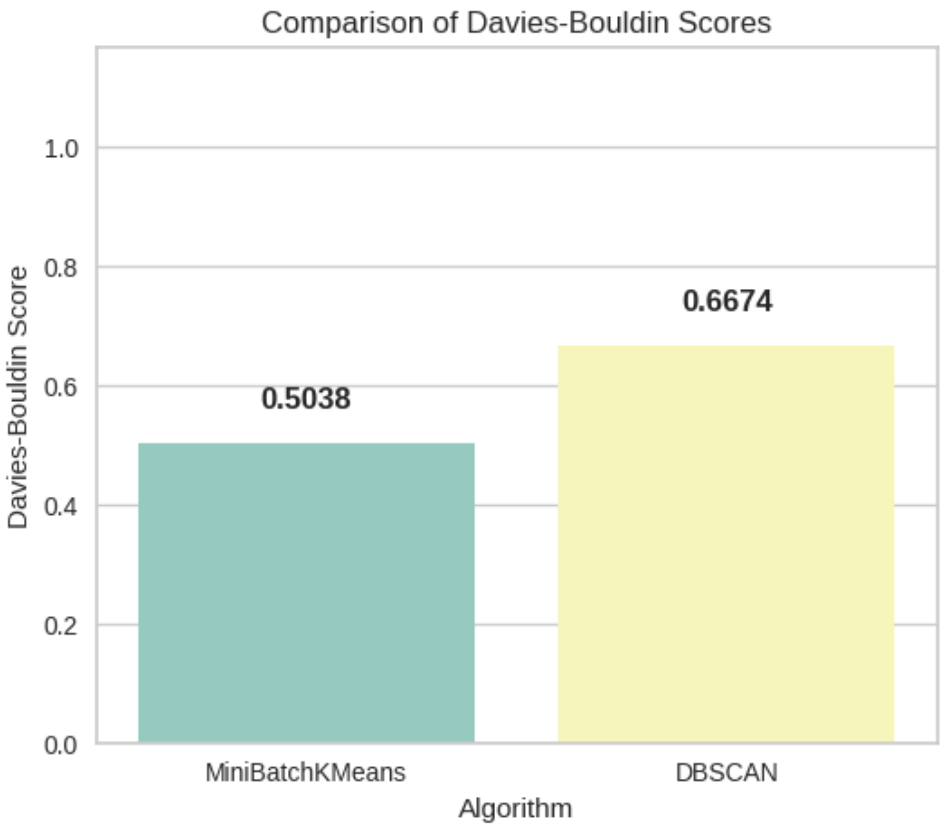
7.1. KẾT QUẢ

- Huấn luyện mô hình MiniBatchKmeans với n_clusters = 12 cho ra kết quả tối ưu nhất trong các lần thử nghiệm , và DBSCAN sử dụng tham số eps = 0.18 và MinPts = 5 cho ra kết quả tối ưu nhất trong các lần thử nghiệm. Dưới đây là các biểu đồ về 3 chỉ số đánh giá khi so sánh 2 mô hình này.



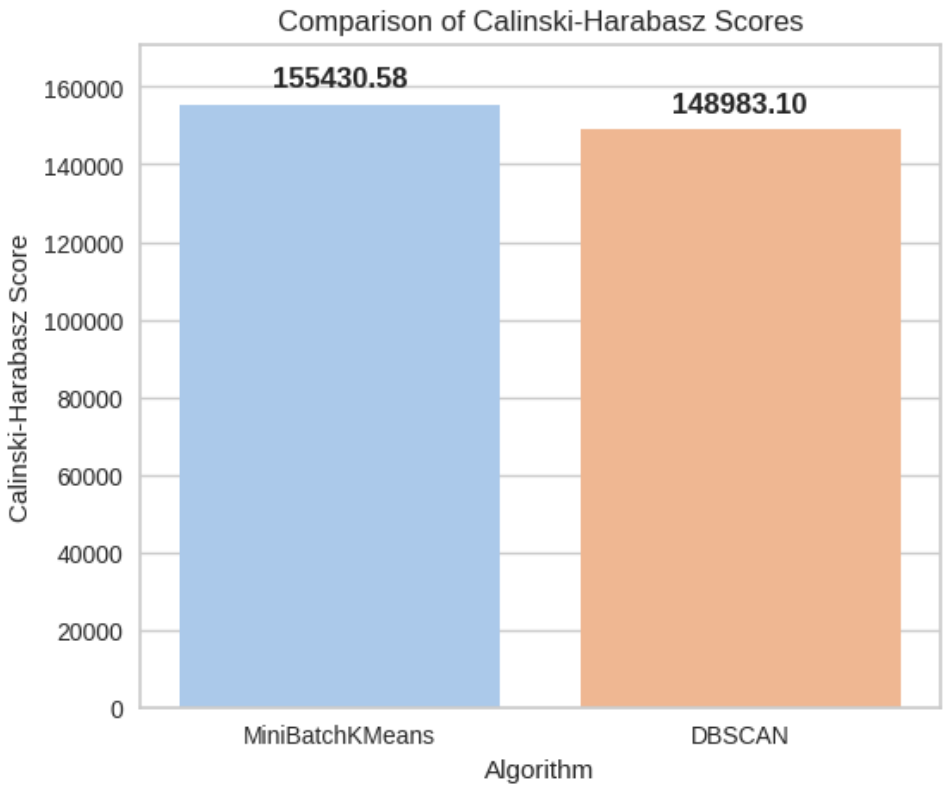
Hình 10: So sánh chỉ số Silhouette Score giữa 2 mô hình.

Xét theo chỉ số Silhouette Score, mô hình MiniBatchKMeans có điểm số cao hơn một chút (0.6893 so với 0.6733 của DBSCAN), cho thấy độ gắn kết trong cụm và mức độ tách biệt giữa các cụm tốt hơn. Cả hai giá trị này đều nằm trên ngưỡng 0.5, phản ánh rằng chất lượng phân cụm của cả hai mô hình khá tốt và tối ưu



Hình 11: So sánh chỉ số Davies-Bouldin Index giữa 2 mô hình.

Nhìn tổng thể, chỉ số Davies-Bouldin Index (DBI) – đại diện cho độ giống nhau giữa các cụm – càng thấp càng tốt, vì điều này cho thấy các cụm phân tách rõ ràng và khác biệt với nhau. Dựa trên biểu đồ kết quả, mô hình MiniBatchKMeans đạt chỉ số DBI là 0.5038, thấp hơn đáng kể so với DBSCAN với giá trị 0.6674. Điều này cho thấy MiniBatchKMeans có khả năng tạo ra các cụm tách biệt và ít chồng lấn hơn, do đó phù hợp hơn trong bối cảnh dữ liệu hiện tại.



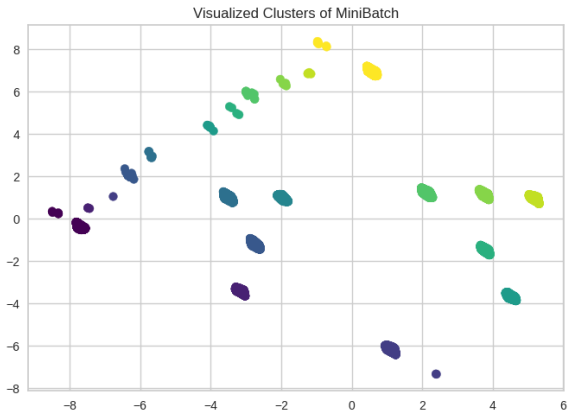
Hình 12: So sánh chỉ số Calinski-Harabasz Index giữa 2 mô hình.

Biểu đồ so sánh chỉ số Calinski-Harabasz Index (Variance Ratio Criterion) cho thấy mô hình MiniBatchKMeans đạt giá trị 155430.58, cao hơn rất nhiều so với DBSCAN chỉ đạt 148983.10. Điều này cho thấy MiniBatchKMeans có khả năng phân tách cụm tốt hơn, với khoảng cách giữa các cụm lớn và phương sai trong cụm nhỏ – là dấu hiệu của một phân cụm hiệu quả.

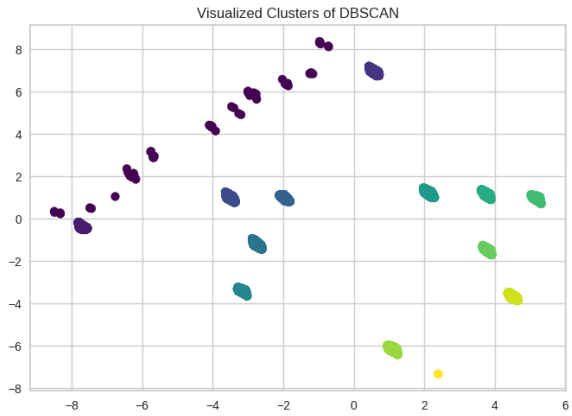
Chỉ số đánh giá	**MiniBatchKMeans**	**DBSCAN**
Silhouette Score	0.6893	0.6733
Davies-Bouldin Index	0.5038	0.6674
Calinski-Harabasz Index	155430.58	148983.10

Bảng 3: So sánh các chỉ số đánh giá giữa hai mô hình phân cụm

- Dựa trên bảng tổng hợp các chỉ số đánh giá, có thể thấy rằng MiniBatchKMeans đều đạt kết quả tốt hơn so với DBSCAN ở cả ba tiêu chí: Silhouette Score cao hơn, Davies-Bouldin Index thấp hơn và Calinski-Harabasz Index cao vượt trội. Điều này cho thấy MiniBatchKMeans tạo ra các cụm có độ gắn kết nội bộ tốt hơn, sự phân tách rõ ràng hơn và cấu trúc cụm ổn định hơn. Chúng ta cùng xem trực quan về sự phân bố các cụm của 2 mô hình MiniBatchKmeans và DBSCAN sau khi sử dụng PCA để giảm số chiều xuống còn 3 để thuận tiện quan sát



Hình 13: Trực quan về sự phân bố cụm của MiniBatchKmeans



Hình 14: Trực quan về sự phân bố cụm của DBSCAN

- Ta thấy cả 2 mô hình đều phâm cụm khá đồng đều và tốt ,nhưng với những chỉ số được đánh giá vượt trội hơn xíu nên có thể kết luận rằng MiniBatchKMeans là mô hình phân cụm hiệu quả hơn trong ngữ cảnh của bộ dữ liệu này, và do đó được lựa chọn làm mô hình tốt nhất trong bài báo cáo này,

7.2. KẾT LUẬN

- Trong báo cáo này,em thực hiện đã sử dụng tập dữ liệu tuyển dụng thu thập từ nền tảng LinkedIn, với quy mô lớn và nhiều ngôn ngữ khác nhau, nhằm mục tiêu khám phá và phân cụm các bài đăng tuyển dụng dựa trên nội dung kỹ năng, tiêu đề, vị trí và địa điểm công việc. Sau quá trình tiền xử lý kỹ càng gồm việc gộp dữ liệu, lấy mẫu kết hợp với lọc dữ liệu tiếng Anh và mã hóa, em đã tiến hành phân cụm bằng hai thuật toán phổ biến là MiniBatchKMeans và DBSCAN.
- Quá trình đánh giá mô hình sử dụng ba chỉ số không giám sát gồm Silhouette Score, Davies-Bouldin Index và Calinski-Harabasz Index. Kết quả cho thấy MiniBatchKMeans consistently outperforms DBSCAN ở tất cả các chỉ số, thể hiện khả năng phân cụm tốt hơn trong trường hợp dữ liệu đã được chuẩn hóa như trong báo cáo này. Do đó, MiniBatchKMeans được lựa chọn là mô hình phân cụm chính.
- Báo cáo này không chỉ mang lại cái nhìn trực quan về việc nhóm các bài đăng tuyển dụng theo kỹ năng mà còn mở ra tiềm năng lớn trong việc phân tích xu hướng nghề nghiệp, hỗ trợ hệ thống gợi ý việc làm, hoặc khám phá các phân khúc thị trường lao động.
- Hướng phát triển của em đề xuất là thứ nhất, áp dụng các kỹ thuật giảm chiều dữ liệu như PCA hoặc t-SNE để cải thiện trực quan hóa và tối ưu hóa hiệu năng phân cụm. Thứ hai, kết hợp nhiều thuật toán phân cụm trong mô hình tổ hợp (clustering ensemble) để tăng độ ổn định của phân cụm. Thứ ba, tiền xử lí dữ liệu đổi các cột đặc trưng thành vector số bằng phương pháp khác là TF-IDF.
- Hạn chế chưa ứng dụng được kết quả phân cụm để sử dụng cho mục đích cụ thể.

TÀI LIỆU THAM KHẢO

Tài liệu

[1] O. Kurasova, V. Marcinkevicius, V. Medvedev, A. Rapecka and P. Stefanovic, "Strategies for Big Data Clustering,"in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI)*, Limassol, Cyprus, 2014, pp. 740-747, doi: 10.1109/ICTAI.2014.115.

[2] Kunchamanjula, *LinkedIn Jobs and Skills EDA Project*, Kaggle Notebook, 2024. Available at: `https://www.kaggle.com/code/kunchamanjula/linkedin-jobs-and-skills-eda-project-6`

[3] Divya Pandove, Shivan Goel, Rinkl Rani, *Systematic Review of Clustering High-Dimensional and Large Datasets*, ACM Trans. Knowl. Discov. Data, vol. 12, no. 2, Article 16, 68 pages, Jan. 2018. DOI: 10.1145/3132088.

[4] Haixun Wang, Wei Wang, Jiong Yang, Philip S. Yu, *Clustering by pattern similarity in large data sets*, In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data (SIGMOD '02)*, Madison, Wisconsin, pp. 394–405, 2002. DOI: 10.1145/564691.564737.