HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP LỚN:

HỌC PHẦN KHAI PHÁ DỮ LIỆU

Đề tài:

Khai phá dữ liệu bằng phương pháp phân cụm và phân lớp để dự báo giá điện thoại

Nhóm bài tập lớn **: 03**

Lớp: **: E22HTTT**

Giảng viên giảng dạy **: Phan Thị Hà**

Thành viên:

Nguyễn Đức Trí – B22DCAT302 (Nhóm trưởng)
Khuất Quang Đông – B22DCVT146
Nguyễn Đại Phát – B22DCVT393
Trần Văn Hoàng – B22DCAT129
Phạm Anh Minh – B22DCAT192
Phạm Việt Bách – B22DCVT043
Nguyễn Ngọc Long – B22DCVT319
Nguyễn Công Minh – B22DCVT343

Hà Nội, 11/2025

Mục lục

LỜI CẢM ƠN

Trong những năm gần đây cùng với phát triển nhanh chóng của khoa học kỹ thuật là sự bùng nổ về tri thức. Kho dữ liệu, nguồn tri thức của nhân loại cũng trở nên đồ sộ, vô tận làm cho vấn đề khai thác các nguồn tri thức đó ngày càng trở nên nóng bỏng và đặt ra thách thức lớn cho nền công nghệ thông tin thế giới.

Nhu cầu về tìm kiếm và xử lý thông tin, cùng với yêu cầu về khả năng kịp thời khai thác chúng để mạng lại những năng suất và chất lượng cho công tác quản lý, hoạt động kinh doanh... đã trở nên cấp thiết trong xã hội hiện đại. Để đáp ứng phần nào yêu cầu này, người ta đã xây dựng các công cụ tìm kiếm và xử lý thông tin nhằm giúp cho người dùng tìm kiếm được các thông tin cần thiết cho mình.

Với các phương pháp khai thác cơ sở dữ liệu truyền thống chưa đáp ứng được các yêu cầu đó. Để giải quyết vấn đề này, một hướng đi mới đó là nghiên cứu và áp dụng kỹ thuật khai phá dữ liệu và khám phá tri thức trong môi trường Web. Do đó, việc nghiên cứu các mô hình dữ liệu mới và áp dụng các phương pháp khai phá dữ liệu trong khai phá tài nguyên Web là một xu thế tất yếu vừa có ý nghĩa khoa học vừa mang ý nghĩa thực tiễn cao.

Vì vậy chúng em chọn đề tài: “Khai phá dữ liệu bằng phương pháp phân cụm và phân lớp để dự báo giá điện thoại”, để làm báo cáo kết thúc môn học của mình.

Báo cáo gồm 5 chương:

Chương 1: Tổng quan về khai phá dữ liệu

Chương 2: Tiền xử lý dữ liệu

Chương 3: Ứng dụng thuật toán K-means để phân cụm sản phẩm điện thoại

Chương 4: Khai phá dữ liệu bằng thuật toán phân lớp

Chương 5: Kết luận và hướng phát triển

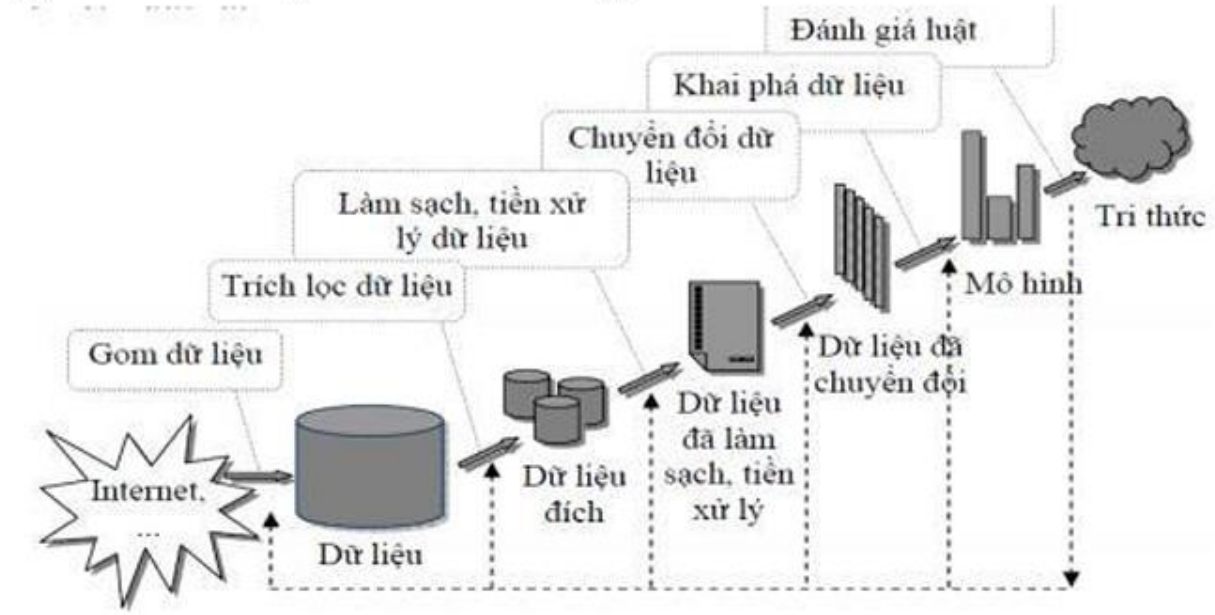
Nhóm 3

CHƯƠNG 1: TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU

1.1. Phát hiện tri thức và khai phá dữ liệu.

- Phát hiện tri thức (Knowledge Discovery) trong các cơ sở dữ liệu là một qui trình nhận biết các mẫu hoặc các mô hình trong dữ liệu với các tính năng: hợp thức, mới, khả ích, và có thể hiểu được.
- Khai phá dữ liệu (Data mining) được định nghĩa như sau: "Data mining là một quá trình tìm kiếm, phát hiện các tri thức mới, tiềm ẩn, hữu dụng trong CSDL lớn".
- Khai phá dữ liệu có thể được sử dụng cho các lĩnh vực y tế, phân tích thị trường, xây dựng ... có thể được xem như là kết quả của sự tiến triển tự nhiên của công nghệ thông tin.

1.2. Quy trình khám phá tri thức trong CSDL.



Hình 1.1 Quá trình khai phá dữ liệu từ cơ sở dữ liệu.

1.3. Mô tả bài toán dự báo giá điện thoại.

1.3.1. Tổng quan bài toán.

- Dataset gồm các mô tả về các thuộc tính tương ứng với dự đoán giá điện thoại.
- Áp dụng các thuật toán để xác định giá của điện thoại.

1.3.2. Phân tích dữ liệu thô

- Nguồn dữ liệu thô: https://www.kaggle.com/datasets/abdulmalik1518/mobiles-dataset-2025/

- Hiểu dữ liệu: Dữ liệu gồm các thông số liên quan đến cấu hình phần cứng của điện thoại và hãng điện thoại. Phân loại giá dựa trên các giá trị của từng thuộc tính.
- Dữ liệu gồm: Dữ liệu bao gồm 930 bản ghi cùng 11 thuộc tính về các đặc trưng của điện thoại .

Company	Model Na	Mobile We	RAM	Front Cam	Back Cam	Processor	Battery Ca	Screen Siz	Launched	Launched Year
Apple	iPhone 16	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh	6.1 inches	USD 799	2024
Apple	iPhone 16	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh	6.1 inches	USD 849	2024
Apple	iPhone 16	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh	6.1 inches	USD 899	2024
Apple	iPhone 16	203g	6GB	12MP	48MP	A17 Bionic	4,200mAh	6.7 inches	USD 899	2024
Apple	iPhone 16	203g	6GB	12MP	48MP	A17 Bionic	4,200mAh	6.7 inches	USD 949	2024
Apple	iPhone 16	203g	6GB	12MP	48MP	A17 Bionic	4,200mAh	6.7 inches	USD 999	2024
Apple	iPhone 16	206g	6GB	12MP / 4K	50MP + 12	A17 Pro	4,400mAh	6.1 inches	USD 999	2024
Apple	iPhone 16	206g	8GB	12MP / 4K	50MP + 12	A17 Pro	4,400mAh	6.1 inches	USD 1,049	2024
Apple	iPhone 16	206g	8GB	12MP / 4K	50MP + 12	A17 Pro	4,400mAh	6.1 inches	USD 1,099	2024
Apple	iPhone 16	221g	6GB	12MP / 4K	48MP + 12	A17 Pro	4,500mAh	6.7 inches	USD 1,099	2024
Apple	iPhone 16	221g	8GB	12MP / 4K	48MP + 12	A17 Pro	4,500mAh	6.7 inches	USD 1,199	2024
Apple	iPhone 16	221g	8GB	12MP / 4K	48MP + 12	A17 Pro	4,500mAh	6.7 inches	USD 1,299	2024
Apple	iPhone 15	171g	6GB	12MP	48MP	A16 Bionic	3,200mAh	6.1 inches	USD 799	2023
Apple	iPhone 15	171g	6GB	12MP	48MP	A16 Bionic	3,200mAh	6.1 inches	USD 849	2023
Apple	iPhone 15	171g	6GB	12MP	48MP	A16 Bionic	3,200mAh	6.1 inches	USD 949	2023
Apple	iPhone 15	203g	6GB	12MP	48MP	A16 Bionic	4,300mAh	6.7 inches	USD 899	2023
Apple	iPhone 15	203g	6GB	12MP	48MP	A16 Bionic	4,300mAh	6.7 inches	USD 999	2023
Apple	iPhone 15	203g	6GB	12MP	48MP	A16 Bionic	4,300mAh	6.7 inches	USD 1,049	2023
Apple	iPhone 15	206g	6GB	12MP / 4K	48MP + 12	A16 Bionic	4,400mAh	6.1 inches	USD 1,099	2023
Apple	iPhone 15	206g	8GB	12MP / 4K	48MP + 12	A16 Bionic	4,400mAh	6.1 inches	USD 1,199	2023

Hình 1.2 Dữ liệu ban đầu.

- Hiểu các thuộc tính:

STT	Thuộc tính	Ý nghĩa
1	Company Name	Tên công ty (Có thể tác động đến giá do thương hiệu)
2	Model Name	Mẫu máy (Có thể kèm theo bộ nhớ trong(ROM))
3	Mobile Weight	Trọng lượng
4	RAM	RAM
5	Front Camera	Thông tin về độ phân giải camera trước
6	Back Camera	Thông tin về độ phân giải camera sau
7	Processor	Tên vi xử lý
8	Battery Capacity	Dung lượng pin
9	Screen Size	Kích thước màn hình
10	Launched Price (USA)	Giá tại thị trường Mỹ tại thời điểm ra mắt
11	Launched Year	Năm ra mắt

CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU

2.1. Làm sạch dữ liệu

- Là quá trình nhận dạng dữ liệu có thể có để tiến hành xử lý các dữ liệu bị nhiễu (noisy data), không nhất quán (inconsistent data) và các dữ liệu bị thiếu (missing data).
  - ■ Xử lý các dữ liệu nhiễu (noisy data), không nhất quán (Inconsistent Data).

Cột	Quy trình làm sạch
Launched Price (USA)	Loại bỏ ký tự tiền tệ ($, USD) và dấu phân cách hàng nghìn (,). Chuyển đổi sang kiểu số thực (float).
RAM, Front Camera, Back Camera	Loại bỏ đơn vị (GB, MP, GHz) và các ký tự không phải số. Trích xuất chỉ giá trị số đầu tiên để xử lý dữ liệu kép/đa (ví dụ: 48MP + 8MP -> 48).
Battery Capacity, Screen Size	Loại bỏ đơn vị đo lường (mAh, inches) và chuyển đổi sang kiểu số.
ROM (Bộ nhớ trong)	Trích xuất thông tin từ cột Model Name. Thực hiện chuyển đổi đơn vị GB sang TB (chia 1024) để đảm bảo tính nhất quán của đơn vị.

RAM	Front Cam	Back Cam	Battery Ca	Screen Siz	Launched	ROM	C
6	12	48	3.6	6.1	799	0.125	
6	12	48	3.6	6.1	849	0.25	
6	12	48	3.6	6.1	899	0.5	
6	12	48	4.2	6.7	899	0.125	
6	12	48	4.2	6.7	949	0.25	
6	12	48	4.2	6.7	999	0.5	
6	12	50	4.4	6.1	999	0.125	
8	12	50	4.4	6.1	1049	0.25	
8	12	50	4.4	6.1	1099	0.5	
6	12	48	4.5	6.7	1099	0.125	
8	12	48	4.5	6.7	1199	0.25	
8	12	48	4.5	6.7	1299	0.5	
6	12	48	3.2	6.1	799	0.125	
6	12	48	3.2	6.1	849	0.25	
6	12	48	3.2	6.1	949	0.5	
6	12	48	4.3	6.7	899	0.125	
6	12	48	4.3	6.7	999	0.25	
6	12	48	4.3	6.7	1049	0.5	
6	12	48	4.4	6.1	1099	0.125	
8	12	48	4.4	6.1	1199	0.25	
8	12	48	4.4	6.1	1299	0.5	
6	12	48	4.5	6.7	1199	0.125	
8	12	48	4.5	6.7	1299	0.25	
8	12	48	4.5	6.7	1399	0.5	
6	12	12	3.2	6.1	799	0.125	

mobiles_dataset_2025_processed

- Xử lý các dữ liệu bị thiếu (Missing Data).
  - Sau quá trình làm sạch và trích xuất, các giá trị thiếu (NaN) còn lại được xử lý như sau:
    +) Thuộc tính ROM: Các giá trị thiếu được điền bằng trung vị (Median) của cột, một phương pháp phù hợp cho dữ liệu liên tục.

Screen Siz	Launched	ROM	C
6.1	799	0.125	
6.1	849	0.25	
6.1	899	0.5	
6.7	899	0.125	
6.7	949	0.25	
6.7	999	0.5	
6.1	999	0.125	
6.1	1049	0.25	
6.1	1099	0.5	
6.7	1099	0.125	
6.7	1199	0.25	
6.7	1299	0.5	
6.1	799	0.125	
6.1	849	0.25	
6.1	949	0.5	
6.7	899	0.125	
6.7	999	0.25	
6.7	1049	0.5	
6.1	1099	0.125	
6.1	1199	0.25	
6.1	1299	0.5	
6.7	1199	0.125	
6.7	1299	0.25	
6.7	1399	0.5	
6.1	799	0.125	

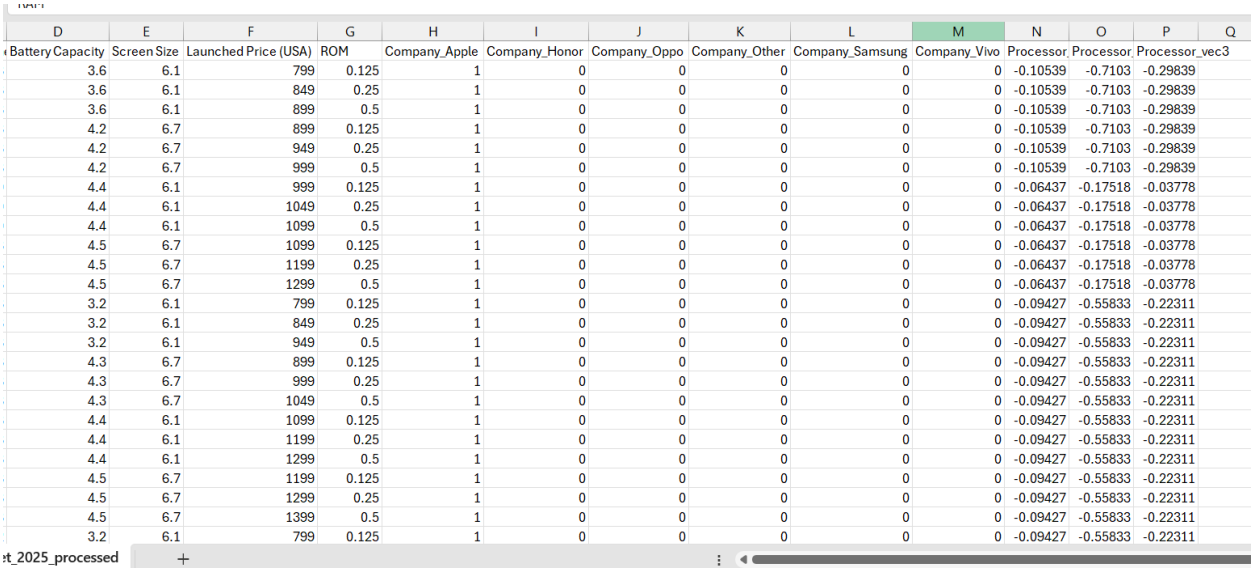
2.2. Tích hợp dữ liệu (Data Integration)

- Là quá trình trộn dữ liệu từ các nguồn khác nhau vào một kho dữ liệu sẵn sàng cho quá trình khai phá dữ liệu.
  - Trong dự án này, dữ liệu được lấy từ một nguồn duy nhất (tập tin CSV) nên không cần thực hiện quá trình tích hợp dữ liệu phức tạp.

2.3. Biến đổi dữ liệu (Data Transformation)

- Giai đoạn Biến đổi dữ liệu nhằm mục đích chuyển đổi các thuộc tính đã được làm sạch thành định dạng số phù hợp với mô hình học máy, đồng thời thực hiện chuẩn hóa để tối ưu hóa hiệu suất mô hình.
- Company Name (Tên hãng):

    - Các hãng được chia thành 6 nhóm bao gồm: apple, honor, oppo, samsung, vivo và các hãng khác (other).
    - Áp dụng kỹ thuật Mã hóa One-Hot (One-Hot Encoding) để chuyển các hãng sản xuất chính (ví dụ: Apple, Samsung, Xiaomi) thành các biến nhị phân (0 hoặc 1). Điều này giúp mô hình nhận biết sự khác biệt giữa các hãng mà không áp đặt mối quan hệ thứ tự.
- Processor (Chip xử lý):
    - Áp dụng TF-IDF Vectorizer để chuyển đổi mô tả chip thành vector, phản ánh tầm quan trọng của các từ khóa.
    - Sử dụng Phân tích Thành phần Chính (PCA) để giảm chiều dữ liệu của vector TF-IDF. Kỹ thuật này trích xuất 3 thành phần chính (Processor_vec1, Processor_vec2, Processor_vec3) để giữ lại phần lớn thông tin cốt lõi của tên chip, đồng thời giảm thiểu số lượng đặc trưng đầu vào.
- ROM (Bộ nhớ trong):
    - Giá trị GB được chuyển đổi thành TB (chia 1024) để đảm bảo tính đồng nhất đơn vị sau đó điền khuyết bằng Median.
- Battery Capacity:
    - Đổi từ mAh thành Ah, mục đích là làm gọn số tránh sai số dữ liệu sau này.
- Launched Price (USA):
    - Chuẩn hóa dấu phẩy (,) để xử lý cả định dạng hàng nghìn và thập phân. Sau đó, nó xác thực giá trị đó phải nằm trong phạm vi hợp lý ($99 đến $2000), nếu không sẽ trả về NaN (giá trị thiếu).



D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Battery Capacity	Screen Size	Launched Price (USA)	ROM	Company_Apple	Company_Honor	Company_Oppo	Company_Other	Company_Samsung	Company_Vivo	Processor	Processor	Processor_vec3	
3.6	6.1	799	0.125	1	0	0	0	0	0	-0.10539	-0.7103	-0.29839	
3.6	6.1	849	0.25	1	0	0	0	0	0	-0.10539	-0.7103	-0.29839	
3.6	6.1	899	0.5	1	0	0	0	0	0	-0.10539	-0.7103	-0.29839	
4.2	6.7	899	0.125	1	0	0	0	0	0	-0.10539	-0.7103	-0.29839	
4.2	6.7	949	0.25	1	0	0	0	0	0	-0.10539	-0.7103	-0.29839	
4.2	6.7	999	0.5	1	0	0	0	0	0	-0.10539	-0.7103	-0.29839	
4.4	6.1	999	0.125	1	0	0	0	0	0	-0.06437	-0.17518	-0.03778	
4.4	6.1	1049	0.25	1	0	0	0	0	0	-0.06437	-0.17518	-0.03778	
4.4	6.1	1099	0.5	1	0	0	0	0	0	-0.06437	-0.17518	-0.03778	
4.5	6.7	1099	0.125	1	0	0	0	0	0	-0.06437	-0.17518	-0.03778	
4.5	6.7	1199	0.25	1	0	0	0	0	0	-0.06437	-0.17518	-0.03778	
4.5	6.7	1299	0.5	1	0	0	0	0	0	-0.06437	-0.17518	-0.03778	
3.2	6.1	799	0.125	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
3.2	6.1	849	0.25	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
3.2	6.1	949	0.5	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.3	6.7	899	0.125	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.3	6.7	999	0.25	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.3	6.7	1049	0.5	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.4	6.1	1099	0.125	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.4	6.1	1199	0.25	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.4	6.1	1299	0.5	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.5	6.7	1199	0.125	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.5	6.7	1299	0.25	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
4.5	6.7	1399	0.5	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	
3.2	6.1	799	0.125	1	0	0	0	0	0	-0.09427	-0.55833	-0.22311	

t_2025_processed

CHƯƠNG 3: ỨNG DỤNG THUẬT TOÁN K-MEANS ĐỂ PHÂN CỤM SẢN PHẨM ĐIỆN THOẠI

3.1. Giới thiệu về Thuật toán K-means

- **K-means** là một trong những thuật toán **phân cụm (Clustering)** phổ biến và cơ bản nhất thuộc nhóm **Học không giám sát (Unsupervised Learning).**
  - **Mục tiêu:** Để phân chia một tập hợp gồm N điểm dữ liệu vào K cụm (cluster) khác nhau.
  - **Nguyên tắc:** Thuật toán sẽ gán mỗi điểm dữ liệu vào cụm có **tâm (centroid)** gần nó nhất. Mục tiêu cuối cùng là tối ưu hóa sao cho tổng bình phương khoảng cách từ các điểm đến tâm cụm của chúng là nhỏ nhất (gọi là **Inertia** hoặc **WCSS).**

3.2. Cách thức Hoạt động của K-means

- Thuật toán K-means hoạt động theo một quy trình lặp đi lặp lại để tìm ra các tâm cụm tối ưu.
  1. **Bước 1: Khởi tạo (Initialize).**
     - Bạn (người dùng) chọn một số K (số cụm bạn muốn).
     - Thuật toán sẽ chọn ra K điểm dữ liệu ban đầu làm K tâm cụm (centroid). Cách chọn thông minh nhất hiện nay là **K-means++** (đảm bảo các tâm ban đầu được trải rộng, không co cụm một chỗ).
  2. **Bước 2: Gán nhãn (Assign).**
     - Với *mọi* điểm dữ liệu, thuật toán sẽ tính khoảng cách (thường là Euclide) từ điểm đó đến K tâm cụm.
     - Mỗi điểm sẽ được "gán" vào cụm của tâm mà nó **gần nhất.**
  3. **Bước 3: Cập nhật (Update)**
     - Sau khi tất cả các điểm đã được gán, vị trí của K tâm cụm sẽ được tính toán lại.
     - Tâm cụm mới của mỗi cụm sẽ được *di chuyển* đến vị trí **trung bình cộng** (mean) của tất cả các điểm dữ liệu nằm trong cụm đó.
  4. **Bước 4: Lặp lại**
     - Thuật toán lặp lại **Bước 2** và **Bước 3.**
     - (Gán nhãn lại): Vì tâm cụm đã di chuyển, một số điểm bây giờ có thể gần tâm cụm khác hơn -> chúng đổi cụm.
- Quá trình này cứ thế tiếp diễn cho đến khi các tâm cụm không còn di chuyển đáng kể nữa, nghĩa là các cụm đã "hội tụ" (ổn định).

3.3. Phương pháp Elbow: Chọn K hợp lý

- Một trong những thách thức lớn nhất của K-means là bạn phải tự chọn K. Nếu chọn K quá nhỏ, các cụm sẽ quá chung chung (under-clustering); nếu chọn K quá lớn, các cụm sẽ bị phân mảnh vô nghĩa (over-clustering
- **Phương pháp Elbow (Khuỷu tay)** giúp chúng ta tìm ra K tối ưu bằng cách trực quan hóa chỉ số **Inertia.**
  - **Inertia (WCSS):** Là tổng bình phương khoảng cách từ các điểm đến tâm cụm của chúng. Inertia thấp nghĩa là các cụm "chặt chẽ" -> Tốt.
  - **Cách hoạt động:**
    1. Chúng ta chạy thuật toán K-means nhiều lần với các giá trị K khác nhau (ví dụ: K chạy từ 1 đến 10).
    2. Với mỗi K, chúng ta ghi lại giá trị Inertia (Inertia luôn giảm khi K tăng).
    3. Chúng ta vẽ đồ thị của Inertia (trục Y) theo K (trục X).
  - **Cách đọc đồ thị:**
    1. Ban đầu, khi tăng K, Inertia sẽ **giảm rất nhanh** (dốc đứng).
    2. Đến một lúc nào đó, đồ thị sẽ bắt đầu "là" ra, gần như nằm ngang. Điều này có nghĩa là việc thêm một cụm mới không còn mang lại lợi ích giảm Inertia nhiều nữa.
    3. Điểm mà đồ thị **bị gãy** (thay đổi độ dốc rõ rệt nhất), trông giống như một **khuỷu tay**, chính là giá trị K tốt nhất được gợi ý.
  - **Trong code (ví dụ với sklearn).**
    1. Tạo một list rỗng: inertia_list = []
    2. Viết vòng lặp for k in range(1, 11):
    3. Bên trong vòng lặp:
       - Tạo model: model = KMeans(n_clusters=k)
       - Fit model: model.fit(data_scaled)
       - Lấy Inertia: inertia_list.append(model.inertia_)
    4. Bên ngoài vòng lặp: Vẽ đồ thị plt.plot(range(1, 11), inertia_list)

***Chú ý**: Ngoài phương pháp Elbow giúp tìm được K hợp lí, việc pick centroid cũng là 1 vấn đề, Áp dụng vào bài toán thực tế "Phân cụm giá điện thoại". Nếu như pick bừa 5 chiếc điện thoại đầu tiên và vô tình 5 chiếc này đều thuộc loại phân khúc "thấp"(ROM,RAM,... đều phế vật) thì sẽ có ảnh hưởng lớn đến việc phân cụm. Vì:

- **Vòng 1 (Gán nhãn):** Cả 5 centroid ban đầu đều ở khu vực "thấp". Các điểm "thấp" sẽ tự chia nhau gán vào 5 centroid này. Tuy nhiên, *tất cả* các điểm "cận cao cấp" và "cao cấp" đều ở rất xa cả 5 centroid. Chúng sẽ cùng bị gán vào *một* centroid duy nhất (cái centroid "thấp" mà gần chúng nhất, dù vẫn là rất xa).
- **Vòng 2 (Cập nhật):**
  - 4 centroid "thấp" sẽ di chuyển loanh quanh trong khu vực "thấp".
  - 1 centroid "thấp" còn lại (cái đã vô tình "bắt" được tất cả các điểm "cao cấp") sẽ đột ngột di chuyển đến vị trí trung bình của *toàn bộ* khu vực "cao cấp" và "cận cao cấp".

Kết quả: Sẽ nhận được một kết quả phân cụm rất tệ: 4 cụm nhỏ chen chúc nhau trong phân khúc thấp, và 1 cụm khổng lồ bao gồm tất cả các phân khúc còn lại.

3.4. Giải pháp lựa chọn centroid thông minh

- Chúng ta sẽ sử dụng vòng lặp. Có 2 cách nhưng chủ yếu là đều áp dụng thư viện hiện đại(như skitlearn trong python).

1. Chạy nhiều lần với khởi tạo ngẫu nhiên (n_init)

- Thay vì chỉ chạy thuật toán 1 lần, chúng ta sẽ chạy nó (ví dụ) 10 lần.
- Mỗi lần, chúng ta "pick bừa" 5 centroid ở 5 vị trí ngẫu nhiên khác nhau.
- Sau 10 lần chạy, chúng ta sẽ có 10 kết quả phân cụm khác nhau.
- Chúng ta sẽ chọn ra kết quả **tốt nhất**—là kết quả có **Inertia** (tổng bình phương khoảng cách từ các điểm đến tâm cụm của chúng) **thấp nhất.**
- Trong thư viện Scikit-learn, tham số này tên là n_init (mặc định thường là 10).

2. Khởi tạo thông minh (K-means++)

- Đây là giải pháp phổ biến và hiệu quả hơn, và hiện là **mặc định** trong hầu hết các thư viện. Thay vì "pick bừa" 5 centroid cùng lúc, K-means++ chọn chúng một cách có chiến lược:
  1. **Centroid đầu tiên:** Chọn 1 điểm dữ liệu *bất kỳ* làm centroid đầu tiên (C1).
  2. **Centroid thứ hai:** Tính khoảng cách từ *mọi* điểm khác đến C1. Chọn điểm **xa nhất** (hoặc chọn ngẫu nhiên với xác suất tỉ lệ với bình phương khoảng cách) làm centroid thứ hai (C2).

3. **Centroid thứ ba:** Với *mọi* điểm, tìm khoảng cách *ngắn nhất* của nó đến *bất kỳ* centroid nào đã được chọn (tức là khoảng cách đến C1 hoặc C2, tùy cái nào gần hơn).
4. Chọn điểm có "khoảng cách ngắn nhất" này là **lớn nhất** làm centroid thứ ba (C3).
5. **Lặp lại:** Cứ tiếp tục chọn điểm tiếp theo sao cho nó *ở xa* tất cả các centroid đã được chọn trước đó, cho đến khi đủ K centroid.

3.5. Ứng dụng: Dự án "Phân loại Giá Điện thoại"

- **Làm rõ thuật ngữ:** Điều quan trọng cần lưu ý: K-means là **Phân cụm (Clustering)**, không phải **Phân loại (Classification).**
  - **Phân loại (bạn dùng):** Là *Học có giám sát*. Bạn cần dữ liệu có nhãn từ trước (ví dụ: "giá rẻ", "cao cấp") để huấn luyện mô hình dự đoán nhãn cho điện thoại mới.
  - **Phân cụm (chúng ta làm):** Là *Học không giám sát*. Chúng ta *không biết* có bao nhiêu phân khúc. Chúng ta yêu cầu K-means: "Hãy nhìn vào dữ liệu Pin, Giá, Hãng... và **tự khám phá** xem có bao nhiêu nhóm tự nhiên trong đó".
- **Quy trình ứng dụng:**
  1. **Thu thập dữ liệu:** 'RAM','Front Camera', 'Back Camera', 'Battery Capacity', 'Screen Size', 'ROM','Launched Price (USA)','Company_Apple', 'Company_Honor', 'Company_Oppo', 'Company_Other','Company_Samsung', 'Company_Vivo', 'Processor_vec1', 'Processor_vec2','Processor_vec3'.
  2. **Tiền xử lý:**
     - Áp dụng StandardScaler cho Launched Price (USA)
  3. **Tìm K:** Chạy phương pháp Elbow trên dữ liệu đã xử lý để tìm K tối ưu (ví dụ: K=3).
  4. **Chạy K-means:** Chạy mô hình K-means cuối cùng với K=3.

```
--- CHẠY MÔ HÌNH CUỐI CÙNG ---
Vui lòng nhìn vào đồ thị Elbow vừa hiển thị và quyết định số K tốt nhất.
Nhập số K bạn chọn (từ 2 đến 10):  3
Đã chọn K = 3. Đang chạy mô hình K-means cuối cùng...
```

```
--- ĐẶC ĐIỂM TRUNG BÌNH CỦA 3 CỤM ---
                RAM  Front Camera  Back Camera  Battery Capacity  Screen Size  \
Cluster
0         10.463104     26.861323    62.687023          4.941216     6.868779
1          5.796651     11.826316    36.650239          5.255452     7.278589
2          5.329897     10.762887    23.030928          4.442577     7.143299

                ROM  Launched Price (USA)  Company_Apple  Company_Honor  \
Cluster
0         0.244275             743.309567            0.0       0.096692
1         0.125486             308.713445            0.0       0.126794
2         0.283183            1028.484536            1.0       0.000000

          Company_Oppo  Company_Other  Company_Samsung  Company_Vivo  \
Cluster
0             0.229008       0.458015         0.101781      0.114504
1             0.059809       0.602871         0.112440      0.098086
2             0.000000       0.000000         0.000000      0.000000

          Processor_vec1  Processor_vec2  Processor_vec3
Cluster
0               0.105787        0.120831       -0.073143
1              -0.077284        0.022345        0.126008
2              -0.096795       -0.594561       -0.241362
```

5. **Diễn giải kết quả:**
   - Gán nhãn cụm (0, 1, 2) vào lại bảng dữ liệu gốc.
   - Sử dụng df.groupby('Cluster').mean() để xem giá trị trung bình của Pin, GiaTien... của từng cụm.
   - **Ví dụ diễn giải:**
     - **Cụm 0:** Giá (TB) vừa phải, Pin (TB) Khỏe -> Nhãn: **"Giá Trung bình".**
     - **Cụm 1:** Giá (TB) cao , Chụp ảnh nét -> Nhãn: **"Giá thấp"**
     - **Cụm 2:** Giá (TB) rất cao, Nhà Apple-> Nhãn: **"Giá Cao"**.

```
---------- TÓM TẮT SỐ LƯỢSNG PHÂN KHÚC ----------
PhanKhuc
Giá thấp          418
Giá trung binh    393
Giá cao            97
```

CHƯƠNG 4: KHAI PHÁ DỮ LIỆU BẰNG THUẬT TOÁN PHÂN LỚP

4.1. Giới thiệu về bài toán phân lớp

- Phân lớp dữ liệu là quá trình gán nhãn cho các đối tượng dữ liệu vào một trong các lớp đã được xác định trước.
- Mục tiêu của bài toán phân lớp là **xây dựng một mô hình học** có khả năng **dự đoán chính xác nhãn lớp của các đối tượng mới** dựa trên thông tin từ tập dữ liệu huấn luyện.
- Quá trình phân lớp bao gồm hai bước chính:
  - **Bước 1: Xây dựng mô hình (Training):**
    - Mô tả tập các lớp đã xác định trước.
    - Chuẩn bị tập dữ liệu huấn luyện (training set), trong đó mỗi mẫu đã có nhãn lớp.
    - Dựa vào tập huấn luyện, tiến hành huấn luyện mô hình bằng các thuật toán: cây quyết định, rừng ngẫu nhiên hoặc KNN.
    - Kết quả là mô hình (hoặc luật phân lớp) có thể mô tả mối quan hệ giữa thuộc tính và lớp.
  - **Bước 2: Vận hành mô hình (Testing/Prediction):**
    - Sử dụng mô hình để dự đoán nhãn lớp cho các đối tượng chưa biết.
    - Đánh giá độ chính xác của mô hình bằng tập dữ liệu kiểm tra (test set) độc lập.
    - Nếu độ chính xác đạt yêu cầu, mô hình được sử dụng để phân lớp các mẫu mới trong thực tế.

4.2. Thuật toán phân lớp

4.2.1. Thuật toán Cây quyết định (Decision Tree)

- Thuật toán Cây quyết định là một trong những phương pháp phân lớp phổ biến nhất.
- Mô hình có cấu trúc dạng cây, trong đó:
  - **Nút trong (internal node)** biểu diễn điều kiện trên một thuộc tính.
  - **Cành (branch)** tương ứng với kết quả của điều kiện.
  - **Lá (leaf)** biểu diễn nhãn lớp.
- Ưu điểm:
  - Dễ hiểu, dễ diễn giải.
  - Xử lý tốt dữ liệu dạng rời rạc.

- Nhược điểm:
  - Có thể bị **overfitting** nếu cây quá sâu.

4.2.2. Thuật toán Rừng ngẫu nhiên (Random Forest)

- Rừng ngẫu nhiên là một **tập hợp của nhiều cây quyết định** (ensemble learning).
- Mỗi cây được huấn luyện trên một phần ngẫu nhiên của dữ liệu và kết quả cuối cùng được lấy bằng **bỏ phiếu đa số (majority voting)**.
- Ưu điểm:
  - Giảm hiện tượng overfitting so với một cây đơn lẻ.
  - Độ chính xác cao, ổn định.
- Nhược điểm:
  - Thời gian huấn luyện lâu hơn.
  - Khó giải thích hơn so với một cây đơn.

4.2.3. Thuật toán KNN (K-Nearest Neighbors)

- Thuật toán KNN là phương pháp **phân lớp dựa trên khoảng cách.**
- Khi cần phân lớp một mẫu mới, thuật toán tìm ra **K mẫu gần nhất** trong tập huấn luyện (dựa theo khoảng cách Euclidean, Manhattan, v.v.), sau đó gán nhãn lớp theo đa số.
- Ưu điểm:
  - Dễ cài đặt, trực quan.
  - Hiệu quả với dữ liệu nhỏ.
- Nhược điểm:
  - Tốn thời gian khi dữ liệu lớn.
  - Phụ thuộc mạnh vào cách chuẩn hóa dữ liệu và giá trị K.

4.3. Các bước thực hiện

Bước 1: Mở file dữ liệu .CSV đã tiền xử lý

- Dữ liệu sau khi đã được làm sạch và mã hóa (ví dụ xử lý giá trị thiếu, chuyển đổi nhãn văn bản sang số, chuẩn hóa thuộc tính) được lưu trong file .csv.
- File được nạp vào chương trình bằng thư viện **pandas** trong Python:

```
# Load data
FILE_PATH = "mobiles_dataset_2025_processed.csv"
df = pd.read_csv(FILE_PATH)
print(f"Loaded {len(df)} rows and {len(df.columns)} columns")
```

Bước 2: Chia tập dữ liệu

- Dữ liệu được chia thành hai phần:
  - **Tập huấn luyện (train)**: 80% dữ liệu.
  - **Tập kiểm tra (test)**: 20% dữ liệu.

```
# Train/test split
RANDOM_STATE = 42
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=RANDOM_STATE
)

# Preprocessing pipeline: impute then scale (fit on train)
numeric_pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

X_train_prep = numeric_pipeline.fit_transform(X_train)
X_test_prep = numeric_pipeline.transform(X_test)

print('Shapes:', X_train_prep.shape, X_test_prep.shape)
```

✓ 0.0s

```
Shapes: (726, 15) (182, 15)
```

Bước 3: Huấn luyện mô hình

```python
# Ensure estimator objects exist
# train with knn
if 'best_knn' in globals() and isinstance(best_knn, int):
    best_knn = KNeighborsRegressor(n_neighbors=best_knn)
#train with default knn
elif 'best_knn' not in globals():
    best_knn = KNeighborsRegressor(n_neighbors=5)
# train with decision tree
if 'best_dt' not in globals():
    best_dt = DecisionTreeRegressor(random_state=RANDOM_STATE)
# train with random forest
if 'rf' not in globals():
    rf = RandomForestRegressor(n_estimators=100, random_state=RANDOM_STATE, n_jobs=-1)
```

Bước 4: Dự đoán và đánh giá hiệu quả mô hình
Bước 4.1: dự đoán bằng mô hình

```python
# Evaluate on test set (use X_test_prep which is preprocessed)
results = {}
knn_pred = best_knn.predict(X_test_prep)
results['KNN'] = print_eval('KNN', y_test, knn_pred)

dt_pred = best_dt.predict(X_test_prep)
results['DecisionTree'] = print_eval('DecisionTree', y_test, dt_pred)

rf_pred = rf.predict(X_test_prep)
results['RandomForest'] = print_eval('RandomForest', y_test, rf_pred)

summary = pd.DataFrame(results).T
display(summary)
```

4.4. Đánh giá mô hình

- Hiệu quả mô hình được đánh giá bằng ba chỉ số chính:

Chỉ số	Ý nghĩa	Công thức
MAE (Mean Absolute Error)	Sai số tuyệt đối trung bình	MAE = (1/n) ∑

RMSE (Root Mean Square Error)	Căn bậc hai sai số bình phương trung bình	$RMSE = \sqrt{(1/n) \sum(y_i - \hat{y}_i)^2}$
R^2 (R-squared)	Hệ số xác định, phản ánh mức độ phù hợp của mô hình	$R^2 = 1 - (SSres/SStot)$

- **MAE và RMSE** càng nhỏ → mô hình càng chính xác
- **R^2** càng gần 1 → mô hình càng tốt.

- Kết quả thu được:

	MAE	RMSE	R2
KNN	1.100837	1.640172	0.838063
DecisionTree	1.035423	1.574037	0.850859
RandomForest	0.956592	1.356167	0.889288

- Từ kết quả trên, có thể thấy **Rừng ngẫu nhiên (Random Forest)** cho độ chính xác cao nhất, sai số thấp nhất. Do đó, mô hình này được lựa chọn cho bước dự đoán cuối cùng.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

- Phân cụm và phân lớp là 2 lĩnh vực khá quan trọng trong khai phá dữ liệu xu hướng trong tương lai, nó được ứng dụng trong nhiều ngành như y tế, thương mại... Hoàn thành đề tài "Khai phá dữ liệu bằng phương pháp phân cụm và phân lớp để dự báo giá điện thoại", nhóm em đã đạt được một số kết quả như sau:
  - Tìm hiểu tổng quan về khai phá dữ liệu, bài toán phân lớp, phân cụm, phương pháp Bags of word để từ đó xây dựng mô hình phân cụm và mô hình phân lớp hỗ trợ dự đoán giá điện thoại.
  - Thu thập dữ liệu điện thoại tiền xử lý dữ liệu bằng Python.
  - So sánh kết quả tỷ lệ train/test để lựa chọn tỷ lệ đánh giá mô hình tốt nhất.
  - Đánh giá mô hình phân lớp so sánh giữa 3 thuật toán Cây quyết định,KNN và random forrest.
- Tuy nhiên bài tập nhóm vẫn còn một số hạn chế:
  - Việc đánh giá chất lượng camera chỉ dựa trên độ phân giải khá phiến diện, cụ thể như sau:
    - Một số thiết bị sử dụng kĩ thuật nội suy để tăng độ phân giải
    - Kích thước cảm biến quan trọng hơn(Tuy nhiên nhiều mẫu máy thường không tiết lộ điều này)
    - Ngoài ra cần những thông tin sau để đánh giá camera:
      +) Tốc độ màn chụp
      +) Khẩu độ
      +) ISO
      +) Chất lượng thấu kính
      +) Khả năng tối ưu phần mềm + AI
  - Nếu có dữ liệu thời gian sử dụng, sẽ tốt hơn dung lượng pin vì khả năng tiêu thụ và tối ưu mỗi máy khác nhau.
  - Kết quả dự đoán tương đối cao nhưng vẫn chưa được tốt nhất.

5.2. Hướng phát triển

- Áp dụng mô hình học máy vào một số lĩnh vực khác và đi vào áp dụng thực tế.

- Xây dựng, cải tiến mô hình dự đoán giá điện thoại với phương pháp học máy khác như Naives Bayes…
- Trong quá trình hoàn thành bài tập lớn, nhóm em đã cố gắng tìm hiểu và tham khảo các tài liệu liên quan. Tuy nhiên, thời gian có hạn nên chúng em sẽ không tránh khỏi những thiếu sót, rất mong nhận được sự đóng góp ý kiến của quý thầy cô và các bạn để báo cáo và kỹ năng của chúng em ngày được hoàn thiện hơn và có thế áp dụng được trong thực tiễn.

TÀI LIỆU THAM KHẢO

[1] https://en.wikipedia.org/wiki/Random_forest
[2] https://cuongndh.blogspot.com/p/khai-pha-du-lieu.html
[3] https://en.wikipedia.org/wiki/K-means_clustering
[4] https://vi.wikipedia.org/wiki/C%C3%A2y_quy%E1%BA%BFt_%C4%91%E1%BB%8Bnh
[5] https://machinelearningcoban.com/
[6] TS.Đặng Thị Thu Hiền, (2019), Bài giảng Khai Phá dữ liệu.
[7] https://www.gsmarena.com/