TRƯỜNG ĐẠI HỌC XÂY DỰNG

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC MÁY

ĐỀ TÀI:

THUẬT TOÁN PHÂN LỚP NAÏVE BAYES VÀ THUẬT TOÁN PHÂN CỤM K-MEANS

Sinh viên thực hiện: **Vũ Xuân Hoàn MSSV 85164**

Trần Quốc Thái MSSV 178264

Triệu Việt Hùng MSSV 1523864

Lớp (nhóm): 64CS1 (nhóm 5)

Hà Nội 09-2021

MỤC LỤC

PHẦN 1: LÝ THUYẾT 1
I. Thuật toán phân lớp Naïve Bayes: 1
1. Đặt vấn đề:.....1
2. Khái niệm: 1
II. Thuật toán phân cụm K-means:.....2
1. Đặt vấn đề:.....2
2. Thuật toán phân cụm K-means: 2
PHẦN 2: THỰC NGHIỆM VÀ KẾT QUẢ 5
I. Thuật toán phân lớp Naïve Bayes: 5
1. Chuẩn bị dữ liệu: 5
2. Mô tả dữ liệu ảnh:.....5
3. Dữ liệu với Naïve Bayes: 6
4. Huấn luyện: 6
5. Phân loại: 7
6. Đánh giá: 8
7. Thực nghiệm:.....8
II. Thuật toán phân cụm K-Means: 13
1. Chuẩn bị dữ liệu: 13
2. Thực nghiệm:.....13

PHẦN 1: LÝ THUYẾT

I. THUẬT TOÁN PHÂN LỚP NAÏVE BAYES:

1. Đặt vấn đề:

Trong học máy, phân loại Naïve Bayes là một thành viên trong nhóm các phân loại có xác suất dựa trên việc áp dụng định lý Bayes khai thác mạnh giả định độc lập giữa các hàm hay đặc trưng. Phân loại Naïve Bayes được đánh giá cao khả năng mở rộng, đòi hỏi một số thông số tuyến tính trong số lượng các biến (các feature) trong nhiều lĩnh vực khác nhau.

2. Khái niệm:

Một phân loại Naïve Bayes dựa trên ý tưởng nó là một lớp được dự đoán bằng các giá trị của đặc trưng cho các thành viên của lớp đó. Các đối tượng là một nhóm trong các lớp nếu chúng có cùng các đặc trưng chung. Có thể có nhiều lớp rời rạc hoặc lớp nhị phân.

Các luật Bayes dựa trên xác suất để dự đoán chúng về các lớp có sẵn dựa trên các đặc trưng được trích xuất. Trong phân loại Bayes, việc học được coi như xây dựng một mô hình xác suất của các đặc trưng và sử dụng mô hình này để dự đoán phân loại cho một ví dụ mới.

Biến chưa biết hay còn gọi là biến ẩn là một biến xác suất chưa được quan sát trước đó. Phân loại Bayes sử dụng mô hình xác suất trong đó phân loại là một biến ẩn có liên quan tới các biến đã được quan sát. Quá trình phân loại lúc này trở thành suy diễn trên mô hình xác suất.

Định lý Bayes: Giả sử A và B là hai sự kiện đã xảy ra. Xác suất có điều kiện A khi biết trước điều kiện B được cho bởi công thức:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Trong đó:

- P(A): Xác suất của sự kiện A xảy ra.
- P(B): Xác suất của sự kiện B xảy ra.
- P(B|A): Xác suất (có điều kiện) của sự kiện B xảy ra, nếu biết rằng sự kiện A đã xảy ra.

- P(A|B): Xác suất (có điều kiện) của sự kiện A xảy ra, nếu biết rằng sự kiện B đã xảy ra.

II. THUẬT TOÁN PHÂN CỤM K-MEANS:

1. Đặt vấn đề:

Phân khúc khách hàng là một kĩ thuật để xác định nhu cầu của khách hàng không được thoả mãn. Phân khúc khách hàng là việc chia nhỏ thị trường thành các nhóm khách hàng rời rạc có cùng đặc điểm. Kĩ thuật này có thể được sử dung bởi các công ty để vượt qua đối thủ cạnh tranh bằng cách phát triển các sản phẩm và dịch vụ hấp dẫn độc đáo.

Các dữ liệu mà các doanh nghiệp sử dụng để phân loại khách hàng của họ là:

- Thông tin nhân khẩu học, như là giới tính, tuổi, tình trạng gia đình và hôn nhân, thu nhập, giáo dục và nghề nghiệp.
- Thông tin địa lý, khác nhau tuỳ thuộc vào phạm vi của doanh nghiệp. Đối với các doanh nghiệp bản địa, thông tin này có thể liên quan đến các thị trấn hoặc quận cụ thể. Còn đối với các doanh nghiệp lớn hơn, nó có thể là một thành phố, tiểu bang hoặc thậm chí là cả quốc gia cư trú của khách hàng.
- Tâm lí học, như tầng lớp xã hội, lối sống và đặc điểm tính cách.
- Dữ liệu về hành vi, chẳng hạn như thói quen chi tiêu và tiêu dùng, việc sử dụng sản phẩm / dịch vụ và các lợi ích mong muốn.

2. Thuật toán phân cụm K-means:

Để thực hiện thuật toán K-means, ta thực hiện các bước sau đây:

- Xác định số lượng cụm K.
- Khởi tạo centroids (tâm) bằng cách xáo trộn tập dữ liệu và sau đó chọn ngẫu nhiên K điểm dữ liệu cho centroid mà không cần thay thế.
- Tiếp tục lặp lại cho đến khi không có thay đổi đối với các centroid, tức là việc gán các điểm dữ liệu cho các cụm không thay đổi.

2.1. Tìm số lượng tối ưu của cụm K:

Để tìm số lượng của cụm K, ta có thể sử dụng giá trị WCSS. WCSS đo lường tổng khoảng cách của các quan sát từ trung tâm cụm của chúng, được đưa ra ở công thức dưới đây.

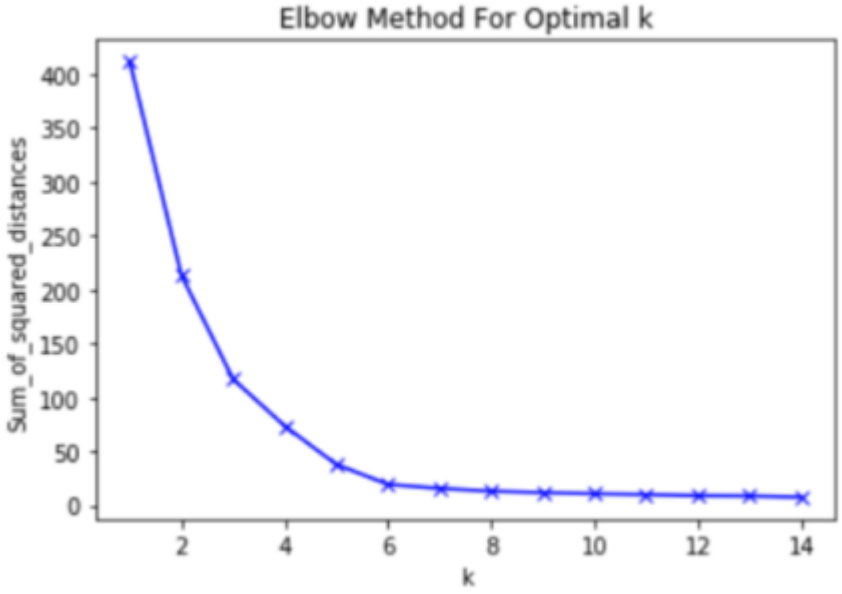
$$WCSS = \sum_{i \in n} (X_i - y_i)^2$$

y_i là trung tâm để quan sát y_i. Mục tiêu chính là tối đa hoá số lượng cụm và hạn chế trường hợp mỗi điểm dữ liệu trở thành trung tâm cụm của riêng nó.

Cụ thể, với mỗi giá trị k, ta sẽ khởi tạo K-means và sử dụng thuộc tính quán tính để xác định tổng bình phương khoảng cách của các mẫu đến trung tâm cụm gần nhất.

Khi k tăng, tổng bình phương khoảng cách có xu hướng bằng 0. Hãy tưởng tượng ta đặt k thành giá trị lớn nhất của nó là n (với n là số mẫu) mỗi mẫu sẽ tạo thành một cụm riêng có nghĩa là tổng các khoảng cách bình phương bằng 0.

Dưới đây là biểu đồ tổng các khoảng cách bình phương cho k trong phạm vi được chỉ định. Nếu biểu đồ trông giống như 1 cánh tay, thì khuỷu tay trên cánh tay là k tối ưu.



Trong biểu đồ ở trên phần khuỷu tay nằm ở k=5 cho thấy k tối ưu cho tập dữ liệu này là 5.

2.2. Thuật toán:

Thuật toán phân cụm K-means như sau:

1. Khởi tạo trung tâm cụm $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$ một cách ngẫu nhiên.
2. Lặp lại cho tới khi hội tụ:

{

Với mọi i, thiết lập:

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

Với mọi j, thiết lập:

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}\}}$$

}

PHẦN 2: THỰC NGHIỆM VÀ KẾT QUẢ

I. THUẬT TOÁN PHÂN LỚP NAÏVE BAYES:

1. Chuẩn bị dữ liệu:

Nhận dạng chữ viết tay là chức năng của máy tính dùng để dự đoán văn bản viết tay dưới dạng các kí tự.

Chúng ta có các tập dữ liệu đại diện cho hình ảnh chữ viết tay cùng với sự phân loại chính xác của những hình ảnh này.

File zip sau khi giải nén gồm những file sau:

- Readme.txt: mô tả những file trong tệp zip.
- Testimages: 1000 ảnh để test.
- Testlabels: Các lớp đúng cho từng ảnh để test.
- Trainingimages: 5000 ảnh để train.
- Traininglabels: các lớp đúng cho từng ảnh để train.

2. Mô tả dữ liệu ảnh:

Dữ liệu hình ảnh bao gồm ảnh với kích thước 28x28 pixel được lưu trữ dưới dạng văn bản. Mỗi ảnh có 28 dòng văn bản với mỗi dòng chứa 28 kí tự. Giá trị của các pixel được mã hoá là ‘ ’ cho màu trắng, ‘+’ cho màu xám và ‘#’ cho màu đen. Ta có thể xem hình ảnh bằng cách sử dụng text editor. Ta sẽ phân loại ảnh của các chữ số từ 0-9, có nghĩa là với hình ảnh của 1 số, ta có thể gắn nhãn chính xác đó là số nào.

Vì máy tính không thể biết hình ảnh là gì nên ta sẽ phải mô tả dữ liệu đầu vào cho nó theo cách mà nó có thể hiểu được. Ta thực hiện điều này bằng cách lấy một hình ảnh đầu vào và trích xuất một tập hợp các feature từ nó. Một feature chỉ đơn giản là mô tả điều gì đó về dữ liệu của chúng ta. Trong trường hợp hình ảnh của chúng ta, ta có thể chia hình ảnh thành dạng lưới được tạo thành bởi các pixel và mỗi pixel có feature mô tả giá trị của nó.

Ta sẽ sử dụng những hình ảnh này như một tập hợp các feature nhị phân sẽ được bộ phân loại sử dụng. Vì hình ảnh có kích thước 28x28 pixel nên ta sẽ có 28x28=784 features cho mỗi ảnh đầu vào. Để tạo ra các feature nhị phân từ dữ liệu này, ta sẽ coi pixel i, j là một feature và nói rằng $F_{i,j}$ có giá trị 0 nếu nó là pixel nền – màu trắng

và 1 nếu nó là nền trước – xám hoặc đen. Trong trường hợp này, ta tổng quát hoá hai giá trị tiền cảnh là xám và đen giống nhau để đơn giản hoá công việc.

3. Dữ liệu với Naïve Bayes:

Áp dụng dữ liệu này cho thuật toán Naïve Bayes, ta muốn tính xác suất cho mỗi đối tượng $F_{i,j}$ có giá trị $f \epsilon \{0,1\}$ để ảnh đó thuộc 1 lớp cho trước $c \in \{0,1, \dots ,9\}$. Để làm điều này, ta sẽ train mô hình trên một tập hợp lớn các hình ảnh mà ta biết lớp mà chúng thuộc về. Một khi ta đã tính toán các xác suất này, ta có thể phân loại ảnh bằng cách tính lớp có xác suất cao nhất cho tất cả các feature đã biết của ảnh và gán nó cho lớp đó. Từ điều này, ta nhận được 2 giai đoạn của thuật toán như sau:

- Training (huấn luyện) – tính toán xác suất có điều kiện mà đối với mỗi feature mà một hình ảnh là một phần của mỗi lớp.
- Classìying (phân loại) - đối với mỗi hình ảnh, tính toán lớp mà nó có nhiều khả năng là thành viên của các xác suất độc lập giả định được tính toán trong giai đoạn huấn luyện.

4. Huấn luyện:

Mục tiêu của giai đoạn huấn luyện là dạy cho máy tính phát hiện khả năng mà một feature $F_{i,j}$ có giá trị $f \in \{0,1\}$ khi ta có các feature nhị phân, cho rằng hình ảnh hiện tại thuộc lớp $c \in \{0,1, \dots ,9\}$, mà ta có thể viết là $P(F_{i,j} = f|class = c)$. Điều này có thể được tính đơn giản bằng công thức:

$$P\left(F_{i,j} = f \middle| class = c\right) = \frac{\#\ lần\ mà\ F_{i,j} = f\ khi\ class = c}{Tổng\ số\ dữ\ liệu\ huấn\ luyện\ ở\ class = c}$$

Với cách sử dụng những xác suất này, ta cần đảm bảo rằng chúng không bằng 0 vì nếu có thì sẽ khiến chúng loại bỏ mọi xác suất khác 0 từ các feature khác. Để làm điều này, ta sẽ sử dụng một kĩ thuật gọi là Laplace Smoothing (làm mịn Laplace). Kĩ thuật này hoạt động bằng cách thêm một giá trị dương k nhỏ vào tử số ở trên, và k. V thành mẫu số (trong đó V là số giá trị có thể mà feature có thể nhận trong trường hợp). Giá trị k càng cao thì độ mịn càng mạnh. Vì vậy, đối với các feature nhị phân thì ta có công thức:

$$P\left(F_{i,j} = f \middle| class = c\right) = \frac{k + \#\ lần\ mà\ F_{i,j} = f\ khi\ class = c}{2k + tổng\ số\ dữ\ liệu\ huấn\ luyện\ ở\ class = c}$$

Ta có thể thử nghiệm với các giá trị khác nhau của k (từ 0,1 đến 10) và tìm được một số k cho ra độ chính xác phân lớp cao nhất. Ta cũng phải ước tính giá trị P (class c) hoặc xác suất của mỗi lớp độc lập với các feature bằng tần số thực nghiệm của các lớp khác nhau trong tập huấn luyện.

$$P(class = c) = \frac{số\ lượng\ điểm\ dữ\ liệu\ trong\ từng\ lớp}{tổng\ số\ điểm\ dữ\ liệu}$$

Với việc huấn luyện được thực hiện xong, ta sẽ có một tập hợp xác suất được tạo theo kinh nghiệm được coi là mô hình mà ta sẽ sử dụng để phân loại trên dữ liệu chưa biết. Mô hình này sau khi được tạo có thể được sử dụng trong tương lai mà không cần tạo lại.

5. Phân loại:

Để phân loại các hình ảnh không xác định bằng cách sử dụng mô hình đã được train, ta phải thực hiện tối đa phân loại sau thử nghiệm (MAP) của dữ liệu thử nghiệm bằng mô hình đã được train. Kỹ thuật này tính toán xác suất sau của mỗi lớp cho hình ảnh cụ thể. Sau đó, phân loại hình ảnh thuộc lớp có xác suất ảnh sau cao nhất.

Vì chúng ta giả định rằng tất cả các xác suất đều độc lập với nhau nên chúng ta có thể tính xác suất mà hình ảnh đó thuộc về lớp bằng cách nhân tất cả các xác suất với nhau và chia cho xác suất của tập feature. Cuối cùng, vì ta thực sự không cần xác suất mà chỉ đơn thuần là để có thể so sánh các giá trị, ta có thể bỏ qua xác suất của tập feature và tính giá trị như công thức sau:

$$P(class) * P(f_{1,1}|class) * P(f_{1,2}|class) * \dots * P(f_{28,28}|class)$$

Nhưng có vấn đề phát sinh là khi máy tính thực hiện tính toán số học với dấu phẩy động, chúng không hoàn toàn tính toán được như cách con người tính toán và chúng có thể tạo ra một loại lỗi gọi là underflow (Đây là hiện tượng xảy ra khi các giá trị số tiến tới giá trị 0 và được coi xấp xỉ là 0. Điều này dẫn đến rất nhiều lỗi ví dụ như chia cho 0 hay là lấy log của 0 thường được coi là $-\infty$).

Để tránh lỗi này, ta sẽ tính toán bằng các sử dụng log của từng xác suất thay vì giá trị thực. Công thức sẽ như sau:

$$\log(P(class)) + \log(P(f_{1,1}|class)) + \log(P(f_{1,2}|class)) + \cdots + \log(P(f_{28,28}|class))$$

Trong đó $f_{i,j}$ là các giá trị feature của dữ liệu đầu vào. Ta có thể sử dụng các xác suất trước và xác suất đã tính được trong giai đoạn train để tính các xác suất sau này.

6. Đánh giá:

Sử dụng nhãn lớp true của tập ảnh test từ tập nhãn test để kiểm tra tính đúng sai của mô hình. Confusion matrix (ma trận nhầm lẫn) là ma trận kích thước 10x10 có các giá trị trong hàng r và cột là tỉ lệ phần trăm của tập ảnh test từ lớp r được phân loại là lớp c.

Trong trường hợp lý tưởng, ma trận này sẽ có giá trị 1 trên đường chéo và giá trị 0 ở những chỗ khác. Điều đó xảy ra khi mọi hình ảnh đều được xác định chính xác, nhưng trong bộ phân loại thì không tốt lắm.

7. Thực nghiệm:

7.1. Đọc dữ liệu và encode:

Đối với bài này thì file đầu vào chứa các 3 loại kí tự " ", "+" và "#" tương ứng như sau:

- " " là màu trắng nền.
- "+" là màu xám.
- "#" là màu đen nền.

Encode:

- Chuyển " " thành kí tự 0.
- Chuyển "+" và "#" thành kí tự 1.

7.2. Implement class NaiveBayes sử dụng Laplace Smoothing để tránh lỗi chia cho 0:

Chạy thử NaïveBayes với smoothing $\in \{0,10\}$, step = 0.1 ta được:

Smoothing	Accuracy	Smoothing	Accuracy	Smoothing	Accuracy	Smoothing	Accuracy
0	0.09	2.5	0.108	5	0.108	7.5	0.108
0.1	0.771	2.6	0.108	5.1	0.108	7.6	0.108
0.2	0.738	2.7	0.108	5.2	0.108	7.7	0.108
0.3	0.624	2.8	0.108	5.3	0.108	7.8	0.108
0.4	0.437	2.9	0.108	5.4	0.108	7.9	0.108
0.5	0.265	3	0.108	5.5	0.108	8	0.108
0.6	0.147	3.1	0.108	5.6	0.108	8.1	0.108
0.7	0.115	3.2	0.108	5.7	0.108	8.2	0.108
0.8	0.108	3.3	0.108	5.8	0.108	8.3	0.108
0.9	0.108	3.4	0.108	5.9	0.108	8.4	0.108
1	0.108	3.5	0.108	6	0.108	8.5	0.108
1.1	0.108	3.6	0.108	6.1	0.108	8.6	0.108
1.2	0.108	3.7	0.108	6.2	0.108	8.7	0.108
1.3	0.108	3.8	0.108	6.3	0.108	8.8	0.108
1.4	0.108	3.9	0.108	6.4	0.108	8.9	0.108
1.5	0.108	4	0.108	6.5	0.108	9	0.108
1.6	0.108	4.1	0.108	6.6	0.108	9.1	0.108
1.7	0.108	4.2	0.108	6.7	0.108	9.2	0.108
1.8	0.108	4.3	0.108	6.8	0.108	9.3	0.108
1.9	0.108	4.4	0.108	6.9	0.108	9.4	0.108
2	0.108	4.5	0.108	7	0.108	9.5	0.108
2.1	0.108	4.6	0.108	7.1	0.108	9.6	0.108
2.2	0.108	4.7	0.108	7.2	0.108	9.7	0.108
2.3	0.108	4.8	0.108	7.3	0.108	9.8	0.108
2.4	0.108	4.9	0.108	7.4	0.108	9.9	0.108

Từ bảng trên ta thấy smoothing $\in \{0,0.4\}$ có các giá trị khá lớn, nên chạy NaiveBayes với step = 0.01 để xem rõ tương quan giữa smoothing với accuracy.

Smoothing	Accuracy	Smoothing	Accuracy
0	0.09	0.2	0.738
0.01	0.684	0.21	0.731
0.02	0.73	0.22	0.724
0.03	0.751	0.23	0.717
0.04	0.755	0.24	0.706
0.05	0.762	0.25	0.695
0.06	0.763	0.26	0.678
0.07	0.769	0.27	0.668
0.08	0.77	0.28	0.653
0.09	0.768	0.29	0.64
0.1	0.771	0.3	0.624
0.11	0.77	0.31	0.606
0.12	0.77	0.32	0.59
0.13	0.77	0.33	0.576
0.14	0.771	0.34	0.557
0.15	0.77	0.35	0.543
0.16	0.764	0.36	0.528
0.17	0.757	0.37	0.506
0.18	0.75	0.38	0.484
0.19	0.749	0.39	0.462

➔ *Kết luận được rằng với smoothing $\in\{0.10, 0.14\}$ thì accuracy của mô hình đạt được cao nhất ở mức 0.771 tức là 77.1%.*

7.3. Viết hàm để tạo ra confusion matrix 10x10 và tính thử Recall và Precision:

Input: nhãn tập test thực tế và dự đoán.

Output: Ma trận đã chuẩn hóa 10x10.

Ví dụ về confusion matrix với smoothing = 0.1 có accuracy = 0.771:

Recall: 0.9771368733026286

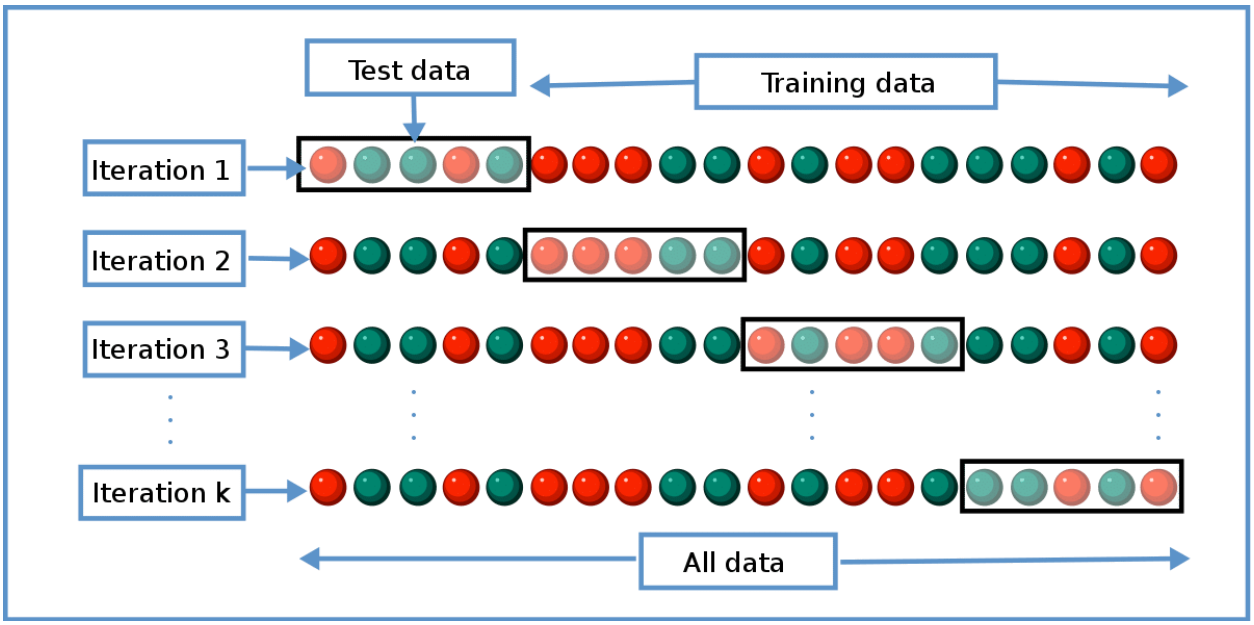
Precision: 0.9106969119628684

confusionMatrix:

```
[[0.789 0.    0.01  0.    0.    0.011 0.    0.    0.01  0.01  ]
 [0.    0.944 0.01  0.    0.    0.    0.022 0.038 0.01  0.    ]
 [0.011 0.009 0.825 0.02  0.019 0.011 0.066 0.038 0.029 0.01  ]
 [0.    0.    0.039 0.75  0.    0.054 0.    0.    0.097 0.02  ]
 [0.011 0.    0.019 0.    0.794 0.022 0.044 0.047 0.029 0.1   ]
 [0.167 0.028 0.029 0.1   0.037 0.826 0.132 0.019 0.194 0.02  ]
 [0.022 0.009 0.039 0.01  0.028 0.011 0.736 0.    0.    0.    ]
 [0.    0.    0.    0.04  0.009 0.011 0.    0.698 0.01  0.03  ]
 [0.    0.009 0.029 0.02  0.019 0.011 0.    0.019 0.553 0.02  ]
 [0.    0.    0.    0.06  0.093 0.043 0.    0.142 0.068 0.79  ]]
```

7.4. Cross Validation:

– Input toàn bộ data random.



– Ta chia nhỏ data thành K phần và chọn test data và train data như hình sau đó lấy trung bình.
– Chạy K cross validation trong khoảng 1 - 10 kèm theo đó smoothing thay đổi từ 0.01 đến 0.1 (step = 0.01) và xem thay đổi.

Chạy xong ta được một số kết quả khá tốt ở mỗi K ví dụ:

- K = 7 và smoothing = 0.13 thì accuracy = 0.8399
- K = 8 và smoothing = 0.12 thì accuracy = 0.8415
- K = 9 và smoothing = 0.13 thì accuracy = 0.8425

7.5. Sử dụng pooling:

– Chạy qua hàm pooling với filter 2x2, S = 2.
– Với chiều của tập train 5000x28x28 thì đầu ra sẽ là 5000x14x14.
– Với chiều của tập test 1000x28x28 thì đầu ra sẽ là 1000x14x14.

7.6. Sử dụng maxpooling:

– Đối với NaiveBayes thì accuracy = 0.754.
– Đối với K-cross validation thì kết quả ở 1 số điểm tiêu biểu có giá trị cao như:
- K = 3 và smoothing = 0.1 thì accuracy = 0.8138.
- K = 6 và smoothing = 0.1 thì accuracy = 0.8115.
- K = 9 và smoothing = 0.14 thì accuracy = 0.8113.

i. Sử dụng meanpooling:

– Đối với NaiveBayes thì accuracy = 0.767.
– Đối với K-cross validation thì kết quả ở 1 số điểm tiêu biểu có giá trị cao như:
- K = 3 và smoothing = 0.1 thì accuracy = 0.8138.
- K = 6 và smoothing = 0.1 thì accuracy = 0.8115.

➔ ***Sau khi pooling thì kết quả thấp hơn một chút.***

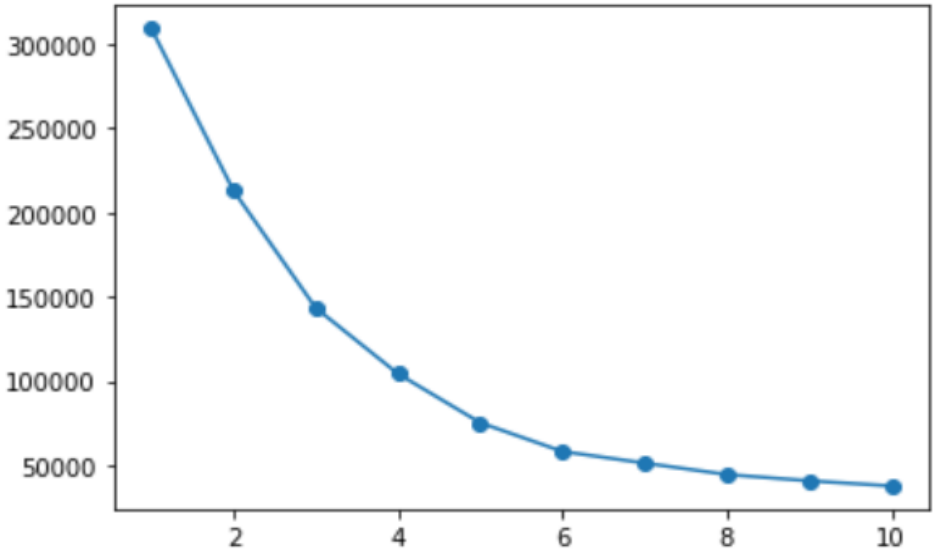
II. THUẬT TOÁN PHÂN CỤM K-MEANS:

1. Chuẩn bị dữ liệu:

Ta có một trung tâm mua sắm và thông qua thẻ thành viên, ta có một số dữ liệu cơ bản về khách hàng như ID Khách hàng, tuổi, giới tính, thu nhập hàng năm và điểm chi tiêu. Tập tin bao gồm thông tin từ 200 khách hàng.

2. Thực nghiệm:

Sau khi tìm hiểu về dữ liệu, áp dụng thuật toán Kmeans và phương pháp Elbow.



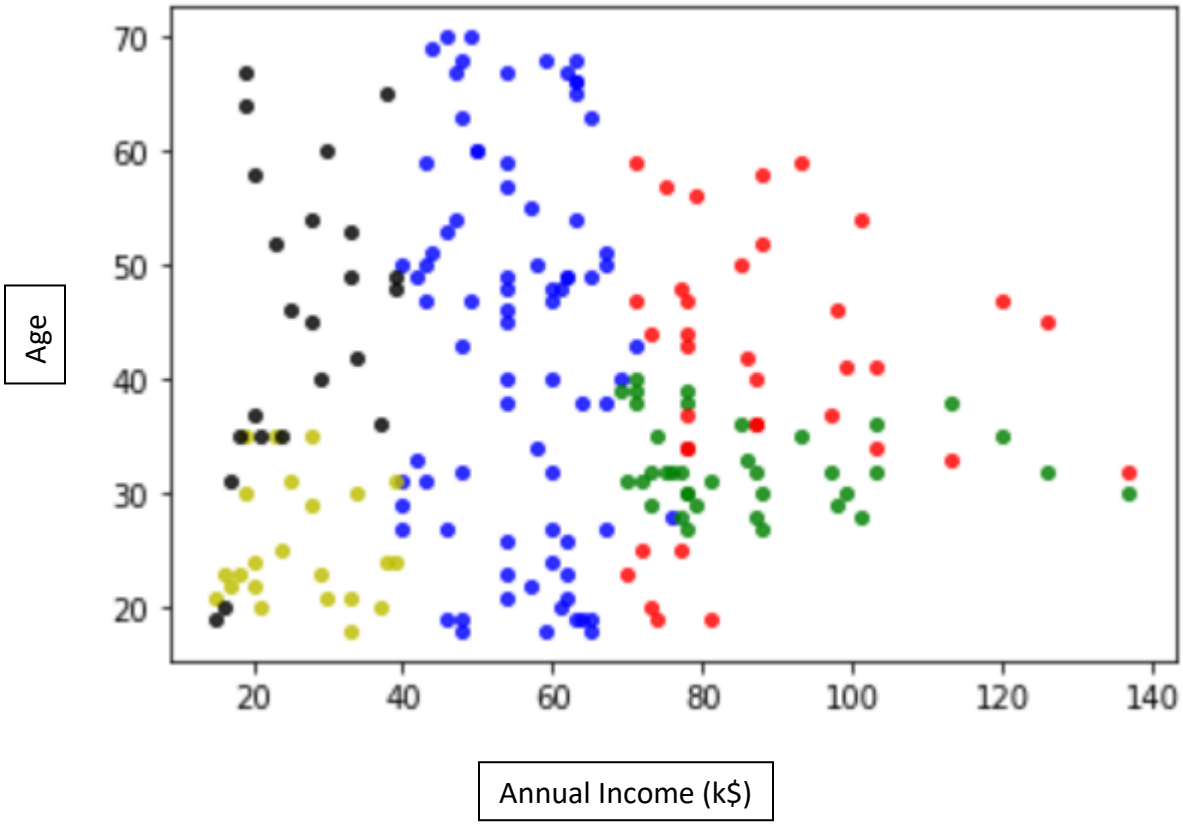
Chúng ta phân nhóm dữ liệu khách hàng ra làm 5 nhóm:

- Nhóm 1: Có 79 người, kí hiệu màu xanh nước biển. (Blue)

- Nhóm 2: Có 39 người, kí hiệu màu xanh lá cây. (Green)
- Nhóm 3: Có 36 người, kí hiệu màu đỏ. (Red)
- Nhóm 4: Có 23 người, kí hiệu màu đen. (Black)
- Nhóm 5: Có 23 người, kí hiệu màu vàng. (Yellow)

Ta sẽ cùng tìm hiểu các đặc điểm của 5 nhóm ấy.

2.1. Trực quan hóa theo tuổi và thu nhập hằng năm (k$):



Biểu đồ 1

Ta có thể thấy:

– Nhóm 1 (Blue): Là nhóm lớn nhất cũng là nhóm gồm tất cả mọi lứa tuổi (18-70 tuổi). Thu nhập của họ cũng ở mức tương đối bằng nhau ở mọi lứa tuổi. Cho thấy nhóm này đại diện cho đại đa số người mua tiêu dùng.

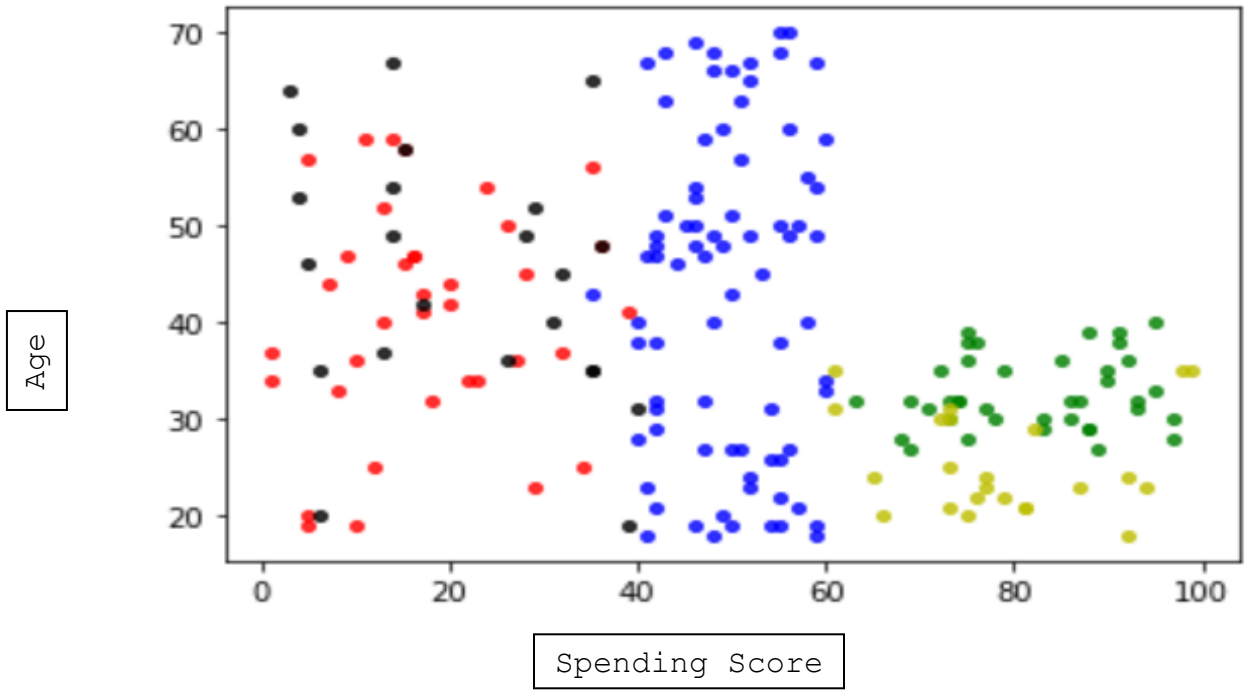
- Nhóm 2 (Green): Là nhóm có thu nhập rất cao (70-140k$) và họ cũng nằm trong tầm tuổi (30-40) đang đi làm với kinh nghiệm và trình độ ở mức tốt nhất. Nên dễ hiểu tại sao họ lại có thu nhập cao.

- Nhóm 3 (Red): Cũng là một nhóm thu nhập cao (70-140k$) nhưng nhóm này lại có những lứa tuổi khác với nhóm 2 là từ 20-25 tuổi và 35-60 tuổi.

- Nhóm 4 (Black): Nhóm này là nhóm có thu nhập thấp (20-40k$), độ tuổi của họ cũng đa phần trên 30 tuổi, chỉ có 2 người khoảng 20 tuổi.

- Nhóm 5 (Yellow): Chúng ta có thể thấy cả nhóm đều có độ tuổi rất trẻ. Chỉ từ 18-35 tuổi, là độ tuổi rất thích mua sắm và các hoạt động vui chơi.

2.2. Trực quan hóa theo tuổi và điểm chi tiêu (1-100):



Biểu đồ 2

Ta có thể thấy:

- Nhóm 1 (Blue): Nhóm có điểm chi tiêu ở mức trung bình (40-60 điểm) và có đầy đủ các lứa tuổi (18-70 tuổi). Có thể thấy rõ hơn nhóm đại diện cho đại đa số người tiêu dùng.

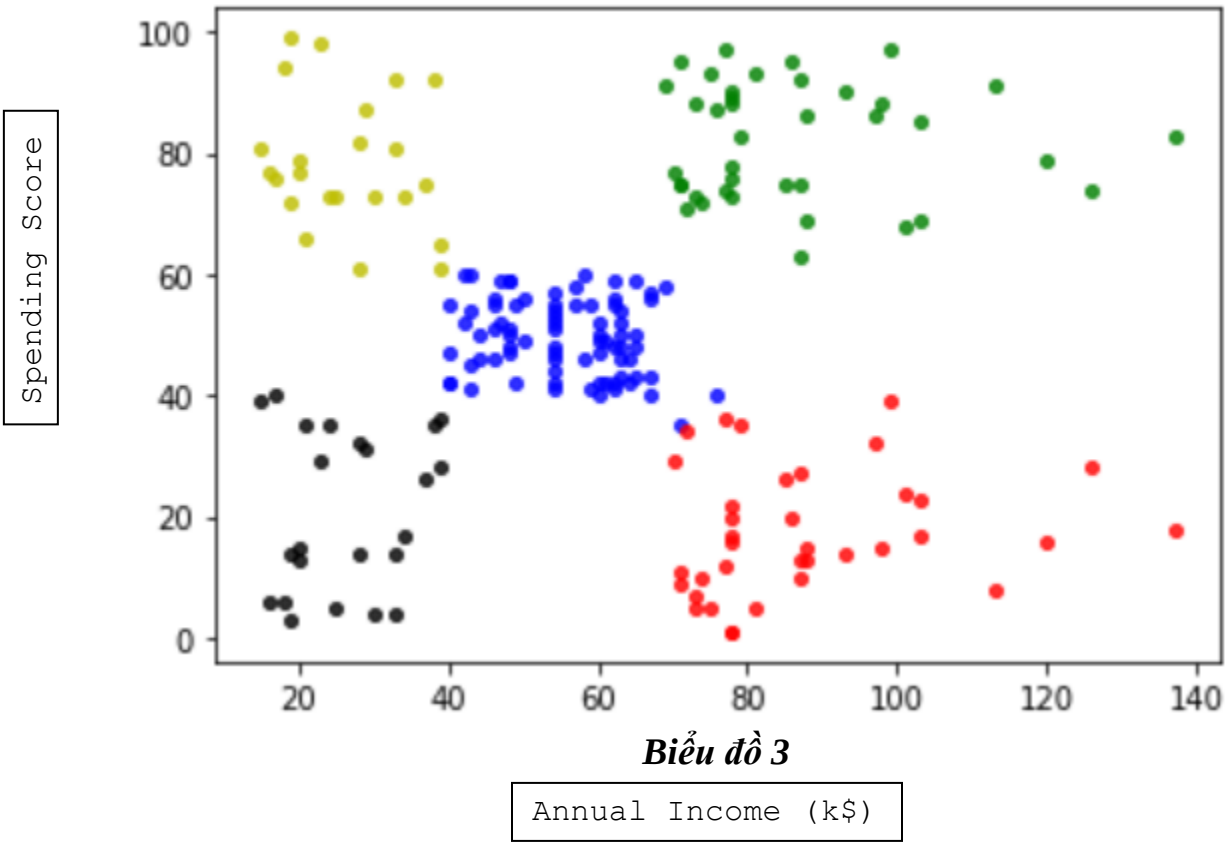
- Nhóm 2 (Green): Nhóm ở độ tuổi chín muồi trong công việc (30-40 tuổi), có thu nhập cao (70-140k$) nên họ tiêu dùng nhiều. Cho thấy đây là nhóm khách hàng cao cấp.

- Nhóm 3 (Red): Là nhóm có độ tuổi phân bố đều (20-60 tuổi) , nhưng tiêu dùng rất ít (0-40 điểm). Phải đẩy mạnh khai thác nhiều hơn ở nhóm này.

- Nhóm 4 (Black): Cũng là nhóm tiêu dùng ít (0-40 điểm) và độ tuổi cũng đa số từ 30-70 tuổi. Cũng là một nhóm phải khai thác thêm.

- Nhóm 5 (Yellow): Là nhóm khách hàng trẻ (18-35t) và có số điểm chi tiêu cao (60-100 điểm). Là nhóm khách hàng thích mua sắm, nên quan tâm và có chính sách tốt để giữ chân họ.

2.3. Trực quan hóa theo thu nhập hằng năm (đơn vị: k$) và điểm chi tiêu (1-100):



Biểu đồ 3

Ta có thể thấy:

- Nhóm 1 (Blue) : Là nhóm chiếm phần đông. Họ có thu nhập ở mức trung bình trong tập dữ liệu (40-70k$), điểm chi tiêu của họ cũng ở mức trung bình (40-60 điểm).

- Nhận định: Nhóm này là nhóm khách hàng lớn, đều đặn và là đại điện cho đại đa số người tiêu dùng của Trung tâm thương mại.Vậy nên ta cần có các chính sách tốt, có đội ngũ tư vấn chăm sóc nhiệt tình để giữ được nhóm khách hàng này.

- Nhóm 2 (Green): Nhóm này tập trung những người có thu nhật rất cao và cao nhất trong tập dữ liệu (70-140 k$) và điểm chi tiêu của họ cũng rất cao (60-100 điểm).

- Nhận định: Đây nhóm khách hàng cao cấp, chi tiêu rất nhiều và nằm trong độ tuổi 30-40 tuổi. Với nhóm khách hàng này khả năng cao đã có gia đình, ta có thể giới thiệu, gửi tin nhắn quảng cáo những mặt hàng cao cấp, tốt cho sức khỏe và tiện ích trong gia đình.

- Nhóm 3 (Red) : Đây là nhóm khách hàng cũng có những người thu nhật rất cao, ngang với nhóm 2 (70-140 k$) nhưng điểm chi tiêu của họ lại rất thấp (0-40 điểm), thấp hơn những người thu nhập trung bình ở nhóm 1.

- Nhận định: Ta thấy đây là nhóm khách hàng có thể khai thác thêm nhiều, có thu nhập tốt và có đầy đủ các lứa tuổi. Ta cần khai thác bằng cách gửi những chính sách đãi ngộ riêng cho nhóm khách hàng này qua tin nhắn, hoặc gửi những quảng cáo sản phẩm tốt, có ích cho sức khỏe, công việc và cuộc sống để có thể tối ưu được nhóm khách hàng này.

- Nhóm 4 (Black): Nhóm này là nhóm khách hàng có thu nhập thấp nhất trong tập dữ liệu (10-40k$) và số điểm chi tiêu của họ cũng nhất nhất (0-40 điểm).

- Nhận định: Là nhóm khách hàng lớn tuổi (đa số >30 tuổi). Với nhóm khách hàng này sẽ thích những sản phẩm tốt, rẻ và có nhiều tiện ích trong gia đình nên chúng ta giới thiệu cho họ nhưng phân khúc sản phẩm tầm trung nhưng lại có nhiều tác dụng, hoặc gửi họ những voucher giảm giá cho lần mua tới.

- Nhóm 5 (Yellow): Nhóm khách hàng này rất đặc biệt, họ có thu nhập thấp, ngang với những người nhóm 4 (10-40 k$) nhưng họ mua sắm rất nhiều, điểm chi tiêu ngang với những người ở nhóm 2 (60-100).
- Nhận định: Họ là những khách hàng trẻ (<35 tuổi) rất thích mua sắm, sẵn sàng chi tiêu cho những sản phẩm họ muốn. Ta nên gửi cho họ nhiều chính sách ưu đãi, voucher tăng dần sau các lần mua, thường xuyên gửi các tin nhắn gợi ý có giảm giá về các mặt hàng họ đã mua hoặc đã xem.