ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN

TRẦN XUÂN PHÚ – 19520843

TRẦN ĐÌNH NAM – 19520758

ĐỒ ÁN CUỐI KỲ HỌC MÁY THỐNG KÊ

XÂY DỰNG MÔ HÌNH PHÂN KHÚC KHÁCH HÀNG DỰA TRÊN BỘ DỮ LIỆU CÓ SẴN MALL CUSTOMERS

Sinh viên ngành KHOA HỌC DỮ LIỆU

GIẢNG VIÊN HƯỚNG DẪN

TS. Nguyễn Tấn Trần Minh Khang

ThS. Võ Duy Nguyên

TP. HỒ CHÍ MINH, 2021

NHẬN XÉT CỦA GIẢNG VIÊN

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

TP. Hồ Chí Minh, ngày.... tháng.... năm 2021

GIẢNG VIÊN HƯỚNG DẪN

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

CHƯƠNG I: MỞ ĐẦU

1. Lý do chọn đề tài.

Phân khúc khách hàng đóng một vai trò khá quan trọng trong Marketing. Việc hiểu rõ được khách hàng mục tiêu rất quan trọng trong việc xây dựng và phát triển một sản phẩm hoặc một dịch vụ thành công.

Việc phân khúc khách hàng sẽ giúp nhanh chóng tìm ra được nhóm khách hàng có tiềm năng, từ đó ta có thể tập trung vào nhóm khách hàng này một cách tốt hơn mà không phải tốn thời gian và sức lực để quan tâm đến các nhóm khách hàng khác ít tiềm năng hơn. Việc tập trung vào một nhóm khách hàng như vậy sẽ khiến cho chất lượng tiếp thị, quản trị đối với nhóm khách hàng đó trở lên linh hoạt và tốt hơn. Ngoài ra, phân khúc khách hàng còn giúp ta thấu hiểu hơn về khách hàng của mình, ta có thể biết được khách hàng của mình cần gì, muốn gì…Từ đó, ta có thể điều chỉnh một cách linh hoạt để được khách hàng tin tưởng và yêu thích hơn.

Vì vậy, lý do chúng tôi chọn đề tài này nhằm nghiên cứu và tìm hiểu về các loại khách hàng cũng như là quá trình phân khúc khách hàng để chọn ra được những nhóm khách hàng có tiềm năng. Cùng với đó, chúng tôi cũng muốn tìm hiểu và học hỏi thêm về các thuật toán Học không giám sát (Unsupervised Learning) nên đã quyết định chọn đề tài này.

2. Mục tiêu

Mục tiêu mà đề tài đặt ra như sau:

- Phân tích, nắm rõ được ý nghĩa các đặc trưng của bộ dữ liệu.
- Nắm rõ được lý thuyết về phương pháp được sử dụng.
- Phân khúc các khách hàng trong bộ dữ liệu ra thành nhiều nhóm.
- Nắm được các ý nghĩa các tham số mô hình.
- Thực nghiệm phương pháp trên một số điểm dữ liệu.
- Đánh giá kết quả đạt được.

CHƯƠNG II: BỘ DỮ LIỆU

1. Giới thiệu về bộ dữ liệu.

- Tên bộ dữ liệu: Mall Customers Segmentation Dataset
- Nguồn bộ dữ liệu: https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python
- Số điểm dữ liệu: 200 điểm dữ liệu
- Số thuộc tính của mỗi điểm dữ liệu: 5 thuộc tính

2. Các thuộc tính của bộ dữ liệu.

Mỗi điểm dữ liệu có 5 thuộc tính, gồm có:

- CustomerID: Mã ID duy nhất của mỗi khách hàng.
- Gender: Giới tính của khách hàng.
- Age: Độ tuổi của khách hàng.
- Annual Inc: Thu nhập hàng năm của khách hàng.
- Spending Score: Điểm chi tiêu của khách hàng do trung tâm mua sắm đánh giá có phạm vi từ 1 đến 100.

CHƯƠNG III: CÁC PHƯƠNG PHÁP SỬ DỤNG

Trong bài toán phân khúc khách hàng này chúng tôi quyết định sẽ sử dụng thuật toán **Phân cụm K-Means (K-Means Clustering)** để phân khúc khách hàng trong bộ dữ liệu thành nhiều nhóm, cùng với đó sẽ sử dụng các thuật toán như **Cây quyết định (Decision Tree)**, **Rừng ngẫu nhiên (Random Forest)** và **K-neighbor** để tiến hành thực nghiệm. Cùng với đó, chúng tôi sẽ đánh giá mô hình vừa được xây dựng bằng các phương thức đánh giá khác nhau như **chỉ số đánh giá Accuracy** và **Ma trận nhầm lẫn (Confusion Matrix).**

1. Thuật toán phân cụm K-Means (K-Means Clustering).

1.1. Tổng quát về thuật toán phân cụm K-Means.

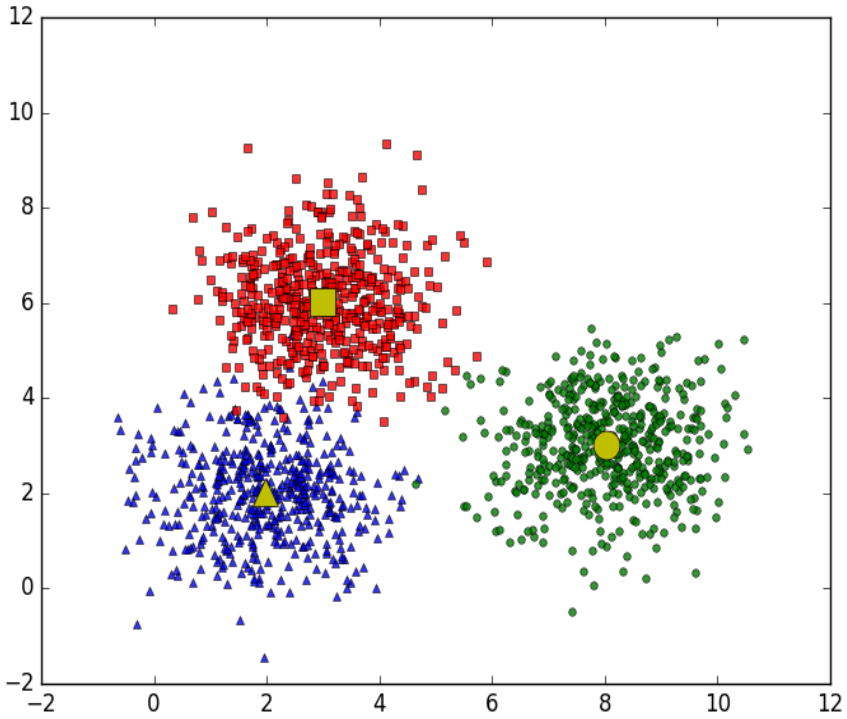
Phân cụm K-Means là một thuật toán phân cụm đơn giản thuộc kỹ thuật Học không giám sát (Unsupervised Learning). Trong thuật toán phân cụm K-Means này, các điểm dữ liệu sẽ không có nhãn và mục đích của thuật toán này làm thế nào để phân các điểm dữ liệu thành các cụm (cluster) khác nhau mà sao cho các điểm dữ liệu trong cùng một cụm phải có cùng một số tính chất nhất định, nói cách khác là giữa các điểm dữ liệu trong cùng một cụm phải có sự liên quan lẫn nhau.

Thuật toán phân cụm K-Means thường được sử dụng trong các bài toán về phân vùng ảnh, phân khúc khách hàng, thống kê dữ liệu, tìm kiếm trong lượng dữ liệu khổng lồ…

1.2. Ý tưởng thuật toán phân cụm K-Means.

Ý tưởng của thuật toán này như sau. Bộ dữ liệu ban đầu sẽ được phân thành k cụm, với mỗi nhóm sẽ có một tâm ngẫu nhiên (centroid) tương ứng. Với mỗi điểm dữ liệu, tiến hành tính khoảng cách từ điểm dữ liệu đó đến các tâm ngẫu nhiên và phân điểm dữ liệu vào cụm có tâm gần với điểm dữ liệu đó nhất. Sau khi đã phân tất cả các điểm dữ liệu vào cụm phù hợp, cập nhật lại tâm mỗi cụm bằng cách lấy trung bình cộng khoảng cách từ các điểm dữ liệu đến tâm của

cụm tương ứng. Thuật toán này sẽ lặp lại các bước phân cụm và cập nhật lại tâm của mỗi cụm cho đến khi tâm của mỗi cụm không thay đổi hoặc tâm của tất cả các điểm dữ liệu không thay đổi.



Hình 1. Bài toán K-means với số cụm là 3

Thuật toán phân cụm K-Means trên được chứng minh là hội tụ và có độ phức tạp tính toán là:

$$O(3nkd)\tau T^{flop}$$

Trong đó:

- n là số lượng điểm dữ liệu.
- k là số cụm dữ liệu.
- d là số chiều dữ liệu.
- τ là số vòng lặp.
- T^{flop} là thời gian để thực hiện một phép tính cơ sở như phép nhân, chia…

Như vậy, do thuật toán K – Means phân tích phân cụm đơn giản nên có thể áp dụng đối với các tập dữ liệu có số điểm dữ liệu lớn.

1.3. Cơ sở toán học.

Phương thức phân cụm dữ liệu được thực hiện dựa trên trên khoảng cách Euclide nhỏ nhất giữa đối tượng đến các phần tử trung tâm của các cụm, khoảng cách giữa 1 điểm dữ liệu đến 1 phần tử trung tâm cụm được tính theo công thức sau:

Gọi:

- $a_i = (a_{i1,} a_{i2}, \dots, a_{im})$ là điểm dữ liệu thứ i cần được phân cụm.

  Với $i = 1 \to n$

- $c_j = (c_{j1}, c_{j2}, .., c_{jk})$ là phần tử trung tâm cụm j .

  Với $j = 1 \to k$

$$\partial_{ji} = \sqrt{\sum_{z=1}^{m} (a_{iz} - c_{jz})^2}$$

Trong đó:

- ∂_{ij} là khoảng cách Euclide giữa tâm cụm c_j và điểm dữ liệu a_i.
- a_{iz} thuộc tính thứ z của điểm dữ liệu i đang xét.
- c_{jz} thuộc tính thứ z của tâm cụm j đang xét.

Ban đầu sẽ có k phần tử trung tâm được chọn ngẫu nhiên cho k nhóm, sau mỗi một lần phân nhóm cho các điểm dữ liệu thì phần tử trung tâm sẽ được chọn lại theo công thức sau:

$$c_{ij} = \frac{\sum_{z=1}^{t} x_{zj}}{t}$$

Trong đó:

- t số điểm dữ liệu hiện có của cụm thứ i.
- x_{zj} thuộc tính thứ j của điểm dữ liệu z.
- c_{ij} toạ độ thứ j của phần tử trung tâm cụm i.

1.4. Ưu và nhược điểm của thuật toán phân cụm K-Means.

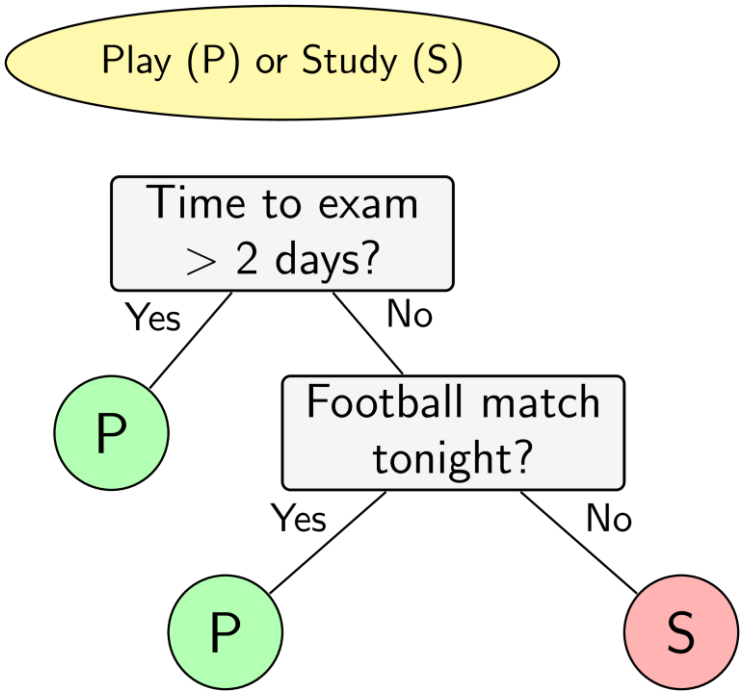
- Ưu điểm của thuật toán K-Means:
  - Dễ dàng cài đặt.
  - Dễ dàng áp dụng với kích thước dữ liệu lớn.
  - Thuật toán đảm bảo hội tụ.
  - Các vị trí trung tâm ban đầu có thể tuỳ biến.
  - Dễ dàng áp dụng với các bài toán mới.
  - Hình dáng và kích thước cụm đa dạng.
- Nhược điểm của thuật toán K-Means:
  - Số cụm k cần được xác định trước. Chúng ta có thể sử dụng phương pháp khuỷu tay (Elbow method) để xác định số cụm k một cách tối ưu hơn.
  - Nghiệm cuối cùng phụ thuộc vào tâm cụm được khởi tạo ban đầu.
  - Các cụm cần có số lượng điểm gần bằng nhau.
  - Các cụm cần có dạng hình tròn (cầu).
  - Phân cụm K-Means sẽ không thực hiện được nếu một cụm nằm phía trong một cụm khác.
  - Thuật toán sẽ chạy không tốt khi dữ liệu có outliers hoặc bộ dữ liệu có số chiều lớn.

2. Phương pháp Cây quyết định (Decision Tree).

2.1. Mô hình Cây quyết định.

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào các dãy luật. Khi cho dữ liệu về các đối tượng

gồm các thuộc tính cùng với lớp của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.



Hình 2. Mô hình cây quyết định.

Trong mô hình Cây quyết định, các ô xám, lục, đỏ trên *Hình 2.1.1* được gọi là các *node*. Các *node* màu lục và đỏ là các *node* thể hiện đầu ra được gọi là các *node lá* (*leaf node* hoặc *terminal node*). Các *node* màu xám thể hiện câu hỏi là các *non-leaf-node*, các *non-leaf-node* trên cùng được gọi là *node gốc* (*root node*). Các *non-leaf-node* thường có hai hoặc nhiều *node con* (*child node*) và các *node con* này có thể là một *node lá* hoặc một *non-leaf-node* khác. Các *node con* có cùng node bố được gọi là *sibling node*. Nếu tất cả các *non-leaf-node* chỉ có 2 *node con*, thì mô hình đó được gọi là mô hình Cây quyết định Nhị phân (Binary Decision Tree). Các câu hỏi trong cây quyết định nhị phân đều có thể được đưa về dạng câu hỏi đúng hay sai. Các cây quyết định không thuộc dạng nhị phân cũng có thể được đưa về dạng cây quyết định nhị phân bởi hầu hết các câu hỏi đều có thể được đưa về dạng câu hỏi đúng sai.

2.2. Tổng quát về thuật toán Cây quyết định.

Cây quyết định là một mô hình học có giám sát (Supervised Learning), có thể được áp dụng vào cả hai bài toán Classification và Regression. Việc xây dựng một mô hình cây quyết định trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của cây quyết định là nó có thể làm việc với các thuộc tính dạng định tính, thường rời rạc và không có thứ tự. Ví dụ xanh, vàng, ngủ, chạy…Ngoài ra cây quyết định cũng làm việc với dữ liệu có vectơ đặc trưng bao gồm cả thuộc tính dạng định tính và định lượng, cùng với đó là cây quyết định ít yêu cầu việc chuẩn hoá dữ liệu.

Như đã nói ở trên, mô hình cây quyết định có thể được áp dụng vào cả hai bài toán Classification và Regression, cho nên mô hình cây quyết định được phân ra thành 2 loại:

- Cây hồi quy (Regression tree): Ước lượng các hàm số có giá trị là số thực thay vì được sử dụng cho các nhiệm vụ phân loại.
- Cây phân loại (Classification tree): Có nhiệm vụ là phân loại với giá trị đầu ra là một biến phân loại.

Thuật toán cây quyết định được ứng dụng vào khá nhiều lĩnh vực để giải quyết các bài toán như Xác định các yếu tố chủ chốt mang đến lợi nhuận tốt nhất cho khách sạn, dự báo ngày đặt chỗ nhiều nhất, dự đoán khối u lành tính hay ác tính, phân loại cấu trúc thứ cấp của protein,…

2.3. Thuật toán xây dựng Cây quyết định.

Có rất nhiều thuật toán được đưa ra để xây dựng nên một cây quyết định, và có một số thuật toán tiêu biểu như:

- ID3 (Iterative Dichotomiser 3): Một thuật toán được áp dụng cho các bài toán Classification mà tất cả các thuộc tính đều ở dạng định tính.

- C4.5 (Successor of ID3): Là thuật toán mở rộng của thuật toán ID3, với khả năng xử lý được cả dữ liệu định lượng dạng liên tục và cả dữ liệu định tính.
- CART (Classification And Regression Tree): CART chủ yếu được dùng để xây dựng cây quyết định chỉ phân theo 2 nhánh mỗi một lần.

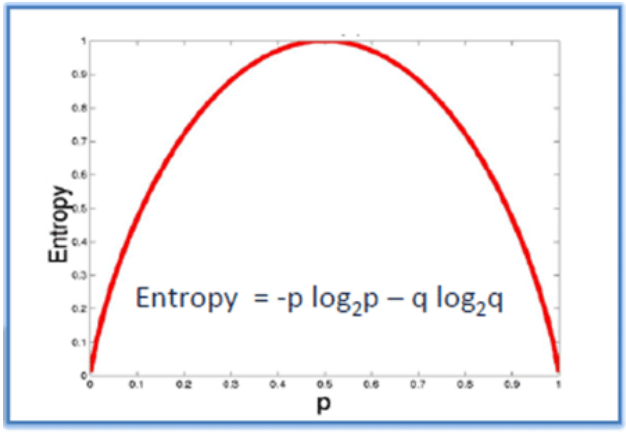
2.4. Bộ tham số thuật toán.

- **Entropy**

Được dùng trong thuật toán xây dựng cây ID3, C4.5, C5.0. Số đo này dựa trên khái niệm entropy trong lý thuyết thông tin, được mở rộng và thống kê với công thức như sau:

- Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $x_1, x_2,\ldots,x_n$.
- Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p(x=x_i)$.
- Ký hiệu phân phối này là $p = (p_1, p_2, \ldots, p_n)$. Entropy của phân phối này được định nghĩa là:

$$H(p) = -\sum_{i=1}^{n} p_i \times \log(p_i)$$



Entropy = -0.5 $\log_2$0.5 – 0.5 $\log_2$0.5 = 1

Hình 3. Hàm Entropy được tính khi tung một đồng xu

Hình 3 biểu diễn sự thay đổi của hàm Entropy. Ta có thể thấy, Entropy đạt tối đa khi xác suất xảy ra của hai lớp bằng nhau.

- **Gini**

Được dùng trong thuật toán xây dựng cây CART. Nó dựa vào việc bình phương các xác suất thành viên cho mỗi thể loại đích trong nút. Giá trị nó tiến đến cực tiểu (bằng 0) khi mọi trường hợp trong nút rơi vào một thể loại nhất định duy nhất.

  - Giả sử y nhận các giá trị từ 1,2,…,m và gọi f(i, j) là tần suất của giá trị j trong nút i.
  - Nghĩa là f(i, j) là tỷ lệ các bản ghi với y = j được xếp vào nhóm i.

$$I_G(i) = 1 - \sum_{j=1}^{m} f(i,j)^2$$

- **Max_depth**

Được dùng để xác định độ sâu tối đa của cây quyết định. Giá trị max_depth được đặt mặc định là none, và trường hợp này thường dẫn đến kết quả bị overfitting. Tham số max_depth được sử dụng là một trong những cách để điều chỉnh cây quyết định, hoặc giới hạn độ sâu của cây nhằm ngăn chặn tình trạng overfitting.

2.5. Ưu và nhược điểm của thuật toán cây quyết định.

- Ưu điểm của thuật toán cây quyết định:
  - Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
  - Dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả.

- Có thể làm việc với cả dữ liệu số và dữ liệu phân loại.
- Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.
- Có khả năng làm việc với dữ liệu lớn.
- Dễ trực quan hoá.

- Nhược điểm của thuật toán cây quyết định:
  - Mô hình cây quyết định hay gặp vấn đề overfitting.
  - Độ chính xác của việc dự báo thấp so với các thuật toán học máy khác.
  - Độ đo Information Gain gặp khó khăn với dữ liệu có miền giá trị là dữ liệu phân loại.
  - Việc tính toán trở nên phức tạp nếu biến phụ thuộc có nhiều lớp (nhãn).

3. Phương pháp Rừng ngẫu nhiên (Random Forest).

3.1. Tổng quát về thuật toán Rừng ngẫu nhiên.

Random Forest là thuật toán xây dựng nên nhiều cây quyết định từ thuật toán Cây quyết định, tuy nhiên mỗi cây quyết định sẽ được xây dựng bằng mỗi cách ngẫu nhiên khác nhau. Sau đó kết quả dự đoán sẽ được tổng hợp từ tất cả các cây quyết định. Trong bài toán phân loại, kết quả của cây quyết định phổ biến nhất sẽ được chọn làm kết quả cuối cùng. Còn trong bài toán hồi quy, mức trung bình của tất cả các kết quả đầu ra của cây được xem là kết quả cuối cùng. Thuật toán Rừng ngẫu nhiên đơn giản và mạnh mẽ hơn so với các thuật toán phân loại phi tuyến tính khác

Thuật toán Rừng ngẫu nhiên được ứng dụng vào các lĩnh vực như Tài chính, Chứng khoán, Dược, Thương mại điện tử để giải các bài toán được đưa ra tìm kiếm khách hàng tiềm năng và khách hàng lừa đảo, dự đoán giá sản phẩm…

3.2. Ý tưởng thuật toán Rừng ngẫu nhiên.

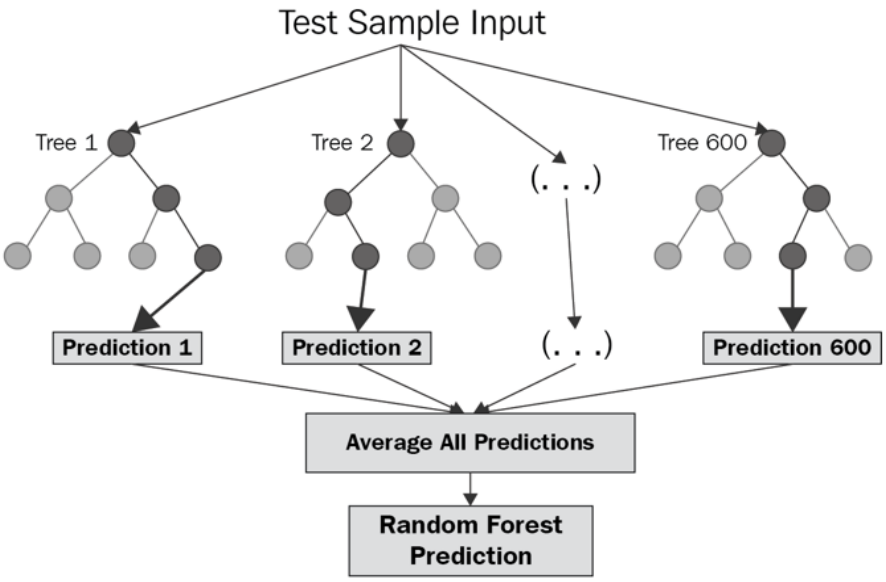
Mô hình Rừng ngẫu nhiên được xây dựng dựa trên 3 bước chính:

- Lấy ngẫu nhiên *n* dữ liệu từ bộ dữ liệu, và chọn ngẫu nhiên *k* thuộc tính (k < số thuộc tính ban đầu).
- Dùng thuật toán Cây quyết định để xây dựng cây quyết định từ bộ dữ liệu mới lấy ra được.
- Lặp lại 2 bước đầu *m* lần để xây dựng được *m* cây quyết định.

Do quá trình xây dựng mỗi cây quyết định đều có yếu tố ngẫu nhiên nên kết quả là các cây quyết định trong thuật toán Rừng ngẫu nhiên có thể khác nhau.

Kết quả của mô hình Rừng ngẫu nhiên ta có được thông qua các bước:

- Lấy kết quả của tất cả các cây quyết định đã tạo ra để dự đoán kết quả và lưu vào một danh sách.
- Tính trung bình cho tất cả các kết quả vừa có được, và lấy đó làm kết quả của mô hình.



Hình 4. Quá trình dự đoán kết quả của mô hình Rừng ngẫu nhiên.

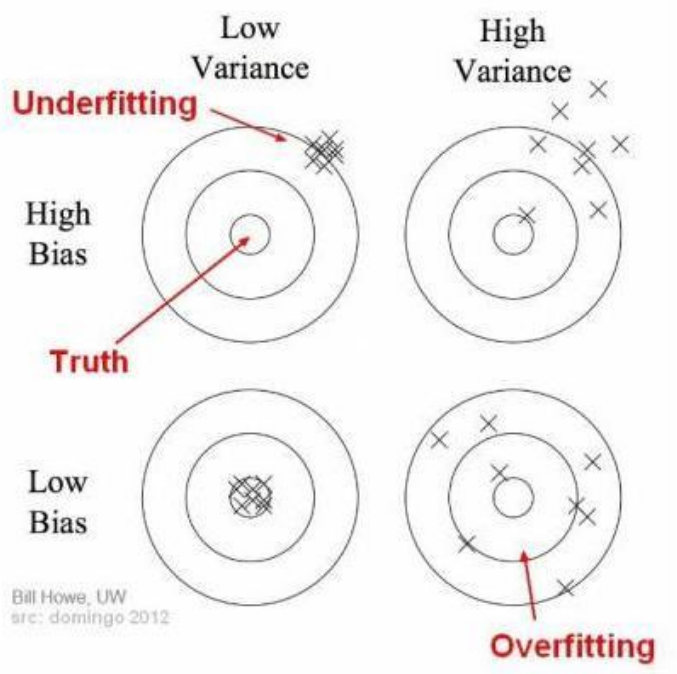
3.3. Ưu điểm của thuật toán Rừng ngẫu nhiên.

Trong thuật toán Cây quyết định, nếu để độ sâu tuỳ ý thì cây sẽ phân loại đúng hết các dữ liệu trong tập training dẫn đến mô hình có khả năng cáo dự đoán tệ trên tập test/validation, khi đó mô hình bị overfitting.

Thuật toán Rừng ngẫu nhiên gồm nhiều cây quyết định, mỗi cây quyết định đều được xây dựng dựa trên những yếu tố ngẫu nhiên:

- Lấy ngẫu nhiên dữ liệu để xây dựng cây quyết định.
- Lấy ngẫu nhiên các thuộc tính để xây dựng cây quyết định.

Do mỗi cây quyết định trong thuật toán Rừng ngẫu nhiên không dùng tất cả dữ liệu training, cũng như không dùng hết tất cả thuộc tính của dữ liệu để xây dựng nên mỗi cây có thể sẽ dự đoán không tốt, khi đó mỗi mô hình cây quyết định không bị overfittingmà có thể bị underfitting. Tuy nhiên, kết quả của mô hình Rừng ngẫu nhiên lại tổng hợp từ nhiều cây quyết định, nên thông tin từ các cây sẽ bổ sung cho nhau dẫn đến mô hình có low bias và low variance, hay mô hình có kết quả dự đoán tốt.



Hình 5. Minh hoạ các trường hợp overfitting, underfitting, balance

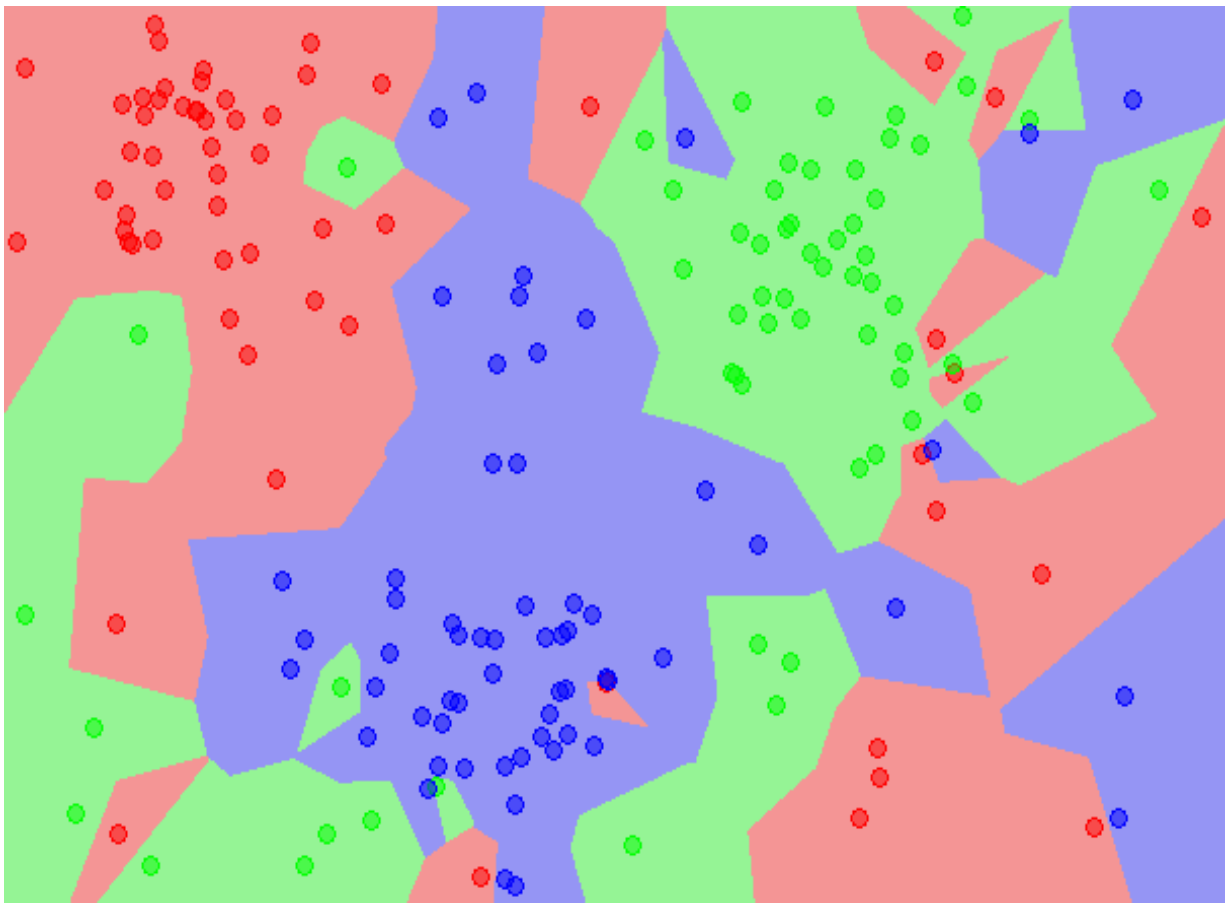
4. Phương pháp KNN (K – nearest neighbor).

4.1. Tổng quát về thuật toán K – nearest neighbor.

K – nearest neighbor là một trong những thuật toán Học có giám sát (Supervised Learning) đơn giản nhất trong Machine Learning. Thuật toán KNN dựa trên giả định là những thứ tương tự hay có tính chất gần giống nhau sẽ nằm ở vị trí gần nhau, với giả định như vậy, KNN được xây dựng trên các công thức toán học phục vụ để tính khoảng cách giữa 2 điểm dữ liệu để xem xét mức độ giống nhau của chúng.

KNN còn hay được gọi với cái tên "Lazy learning method" vì tính đơn giản của nó, bởi khi học thuật toán này không học một điều gì từ dữ liệu đào tạo, có nghĩa là quá trình training không quá phức tạp để hoàn thiện mô hình (tất cả các dữ liệu đào tạo có thể được sử dụng để kiểm tra mô hình KNN). Tuy nhiên, điều này làm cho việc xây dựng mô hình nhanh hơn nhưng giai đoạn thử nghiệm chậm hơn và tốn kém hơn về mặt thời gian và bộ nhớ lưu trữ, đặc biệt khi bộ dữ lớn và phức tạp với nhiều thuộc tính khác nhau. Trong các trường hợp xấu nhất, KNN cần thêm thời gian để quét tất cả các điểm dữ liệu và việc này sẽ cần nhiều không gian bộ nhớ hơn để lưu trữ dữ liệu. Ngoài ra, KNN không cần dựa trên các tham số khác nhau để tiến hành phân loại dữ liệu, không đưa ra bất kỳ kết luận cụ thể nào giữa biến đầu vào và biến mục tiêu, mà chỉ dựa trên khoảng cách giữa điểm dữ liệu cần phân loại với điểm dữ liệu đã phân loại trước đó. Đây là một đặc điểm cực kỳ hữu ích vì hầu hết trong thế giới thực tại không thực sự tuân theo bất kỳ giả định lý thuyết nào, ví dụ như phân phối chuẩn trong thống kê.

Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu.



Hình 6. K – nearest neighbor trong Classification với K = 1.

Thuật toán K – nearest neighbor được sử dụng phổ biến nhất trong các nhiệm vụ giúp các tổ chức tài chính, ngân hàng phân tích rủi ro tín dụng của khách hàng. Ngoài ra, KNN còn được sử dụng trong các lĩnh vực như Y tế về việc xác định loại thuốc phù hợp, lĩnh vực Giáo dục về việc phân loại học sinh, lĩnh vực Thương mại điện tử về việc xây dựng hệ thống khuyến nghị, phân loại khách hàng dựa trên dữ liệu từ website và trong nhiều lĩnh vực khác về các nhiệm vụ dự báo các sự kiện kinh tế trong tương lai, dự báo tình hình thời tiết, xác định xu hướng thị trường chứng khoán…

4.2. Ý tưởng thuật toán.

K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

Trong bài toán Classification, nhãn của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong tập dữ liệu training. Nhãn của một test data có thể được quyết định bằng cách bầu chọn theo số phiếu giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra nhãn.

Trong bài toán Regression, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.

Thuật toán KNN được xây dựng dựa trên các bước chính:

- Từ tập dữ liệu đã được gán nhãn là D và A là tập dữ liệu chưa được gán nhãn.
- Chọn một số K bất kỳ, K là một số nguyên, là số điểm dữ liệu đã phân loại có khoảng cách gần nhất với điểm dữ liệu chưa được phân loại.
- Đo khoảng cách (có thể sử dụng Euclidean, Manhattan, Minkowski hoặc trọng số) từ dữ liệu mới A đến tất cả các điểm dữ liệu khác đã được phân loại trong D.
- Với kết quả có được, sắp xếp theo thứ tự với giá trị khoảng cách từ bé đến lớn nhất.
- Chọn ra các điểm dữ liệu có giá trị khoảng cách bé nhất với điểm dữ liệu cần được phân loại dựa trên K cho trước.
- Xem xét giá trị của biến mục tiêu của các điểm dữ liệu gần nhất, chọn ra giá trị xuất hiện nhiều nhất và gán cho điểm dữ liệu chưa được phân loại.

Bước khó khăn nhất của thuật toán KNN, và cũng là bước đau đầu nhất, cần sự kinh nghiệm của nhà phân tích, đó chính là chọn K bằng bao nhiêu.

4.3. Ưu và nhược điểm của thuật toán KNN.

- Ưu điểm của thuật toán KNN:
  - Thuật toán đơn giản, dễ dạng triển khai.
  - Độ phức tạp tính toán của quá trình training là bằng 0.
  - Việc dự đoán kết quả của dữ liệu mới rất đơn giản.
  - Không cần giả sử gì về phân phối của các class.
  - Xử lý tốt với tập dữ liệu nhiễu.
- Nhược điểm của thuật toán KNN:
  - Với K nhỏ thì KNN rất nhạy cảm với tập dữ liệu nhiễu.
  - Với một thuật toán mà mọi tính toán đều nằm ở khâu test. Trong đó việc tính khoảng cách tới từng điểm dữ liệu trong bộ dữ liệu training sẽ tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.

5. Phương pháp đánh giá.

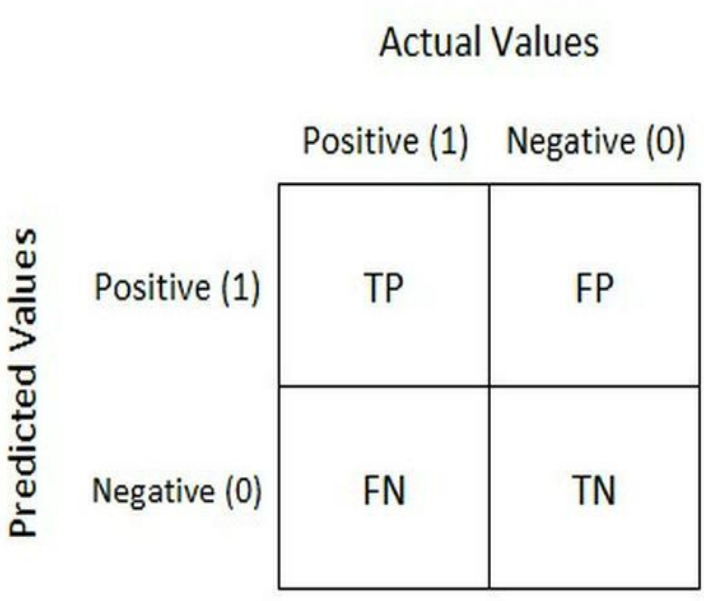
5.1. Chỉ số đánh giá Accuracy.

Phương pháp đánh giá này đơn giản tính tỷmva

lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. Accuracy chỉ phù hợp với các bài toán mà kích thước các lớp dữ liệu là tương đối giống nhau.

5.2. Phương pháp đánh giá Confusion Matrix.

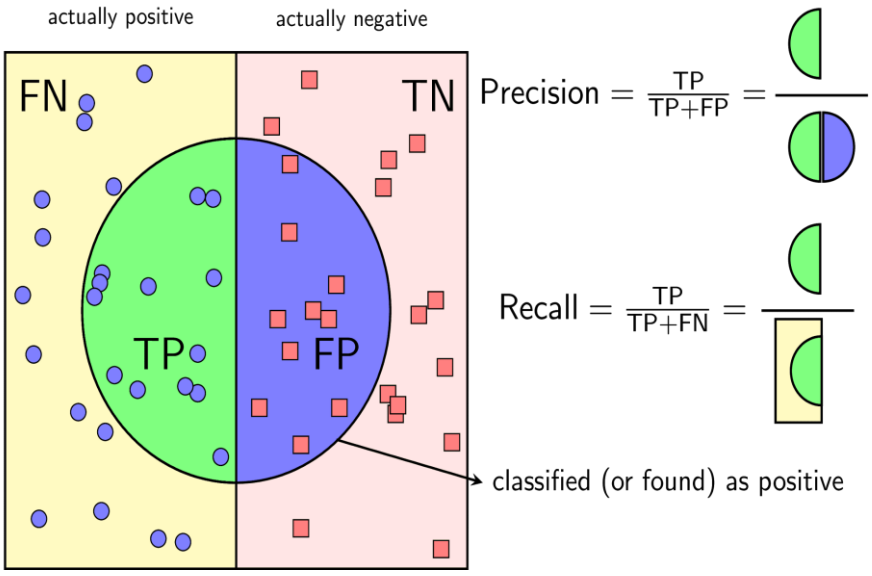
Phương pháp đánh giá Ma trận nhầm lẫn (Confusion Matrix) là một phương pháp đánh giá kết quả của những bài toán phân loại và xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số đối với mỗi lớp phân loại:



Hình 7. Confusion Matrix.

- TP (True Positive): Số lượng dự đoán chính xác.
- TN (True Negative): Số lượng dự đoán chính xác một cách gián tiếp.
- FP (False Positive): Số lượng các dự đoán sai lệch.
- FN (False Negative): Số lượng các dự đoán sai lệch một cách gián tiếp.

Từ 4 chỉ số của một Confusion Matrix, có thể đánh giá mức độ tin cậy của mô hình bằng 2 giá trị:



Hình 8. Chỉ số Precision và Recall.

- Precision: được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive.
- Recall: được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive.

Cả Precision và Recall đều là các số không âm nhỏ hơn hoặc bằng một. Precision cao tương đương với việc độ chính xác của điểm tìm được là cao. Recall cao đồng nghĩa với việc True Positive Rate cao, tức tỉ lệ bỏ sót các điểm thực sự là Positive thấp.

F1-Score là harmonic mean của precision và recall. F1-Score có giá trị nằm trong nửa khoảng (0,1], F1-Score càng cao thì bộ phân lớp càng tốt.

$$F1 = 2 \times \frac{1}{\frac{1}{precision} + \frac{1}{recall}} = 2 \times \frac{precision \times recall}{precision + recall}$$

Macro Average là chỉ số trung bình cộng của các chỉ số theo class.

$$Macro\ Average = \frac{\sum_{c=1}^{C} precision/recall/F1 - score}{c}$$

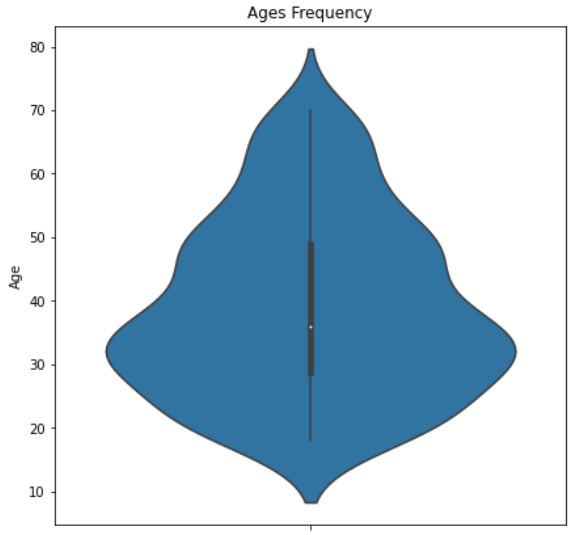
Weighted Average là trung bình có trọng số là một phép tính có tính đến mức độ quan trọng khác nhau của các con số trong bộ dữ liệu. Khi tính chỉ số này, mỗi số trong bộ dữ liệu được nhân với trọng số định trước (w) trước khi thực hiện phép tính cuối cùng. Mức trung bình trọng số có thể chính xác hơn mức trung bình đơn giản, trong đó tất cả các số trong tập dữ liệu được gán một trọng số giống nhau.

$$Weighted\ Average\ = \frac{\sum_{c=1}^{C} w_c \times precision/recall/F1 - score}{\sum_{c=1}^{C} w_c}$$

CHƯƠNG IV: XÂY DỰNG MÔ HÌNH

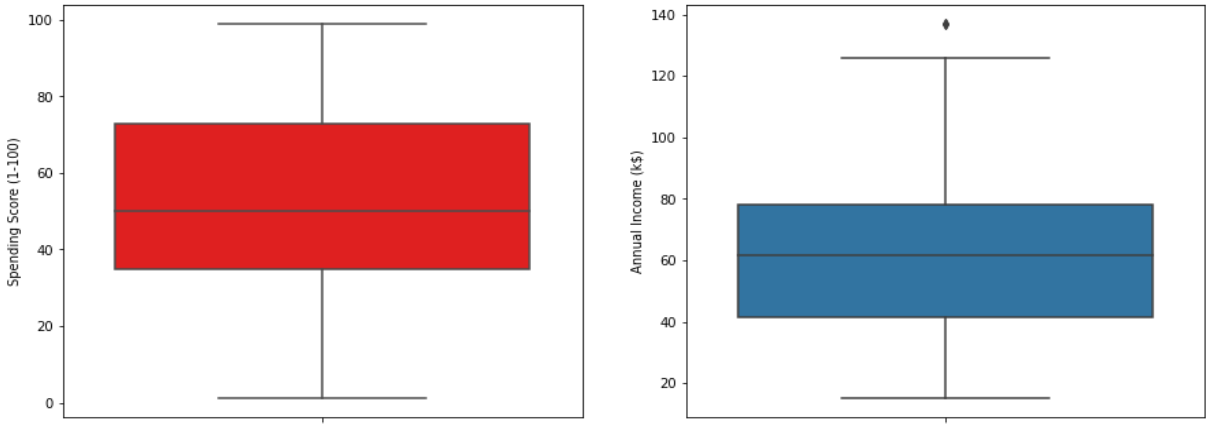
1. Trực quan hoá và phân tích dữ liệu.

Bước đầu tiên, chúng tôi trực quan hoá các điểm dữ liệu nhằm có một góc nhìn rõ ràng hơn về các đặc điểm của bộ dữ liệu và phân tích để có thể chọn ra được các thuộc tính phù hợp cho việc huấn luyện mô hình nhất:



Hình 9. Biểu đồ thuộc tính Age.

Qua biểu đồ về thuộc tính độ tuổi (Age) ta nhận thấy được đa số khách hàng của trung tâm mua sắm đều nằm trong khoảng độ tuổi từ 20 cho đến 40 tuổi.

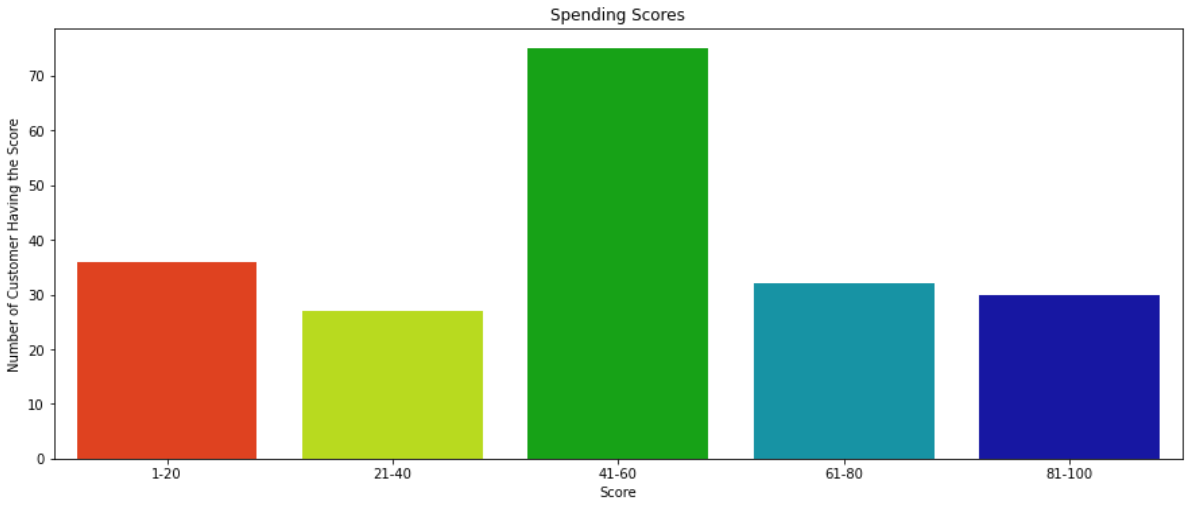


Hình 10. Biểu đồ thuộc tính Spending Score và Annual Income.

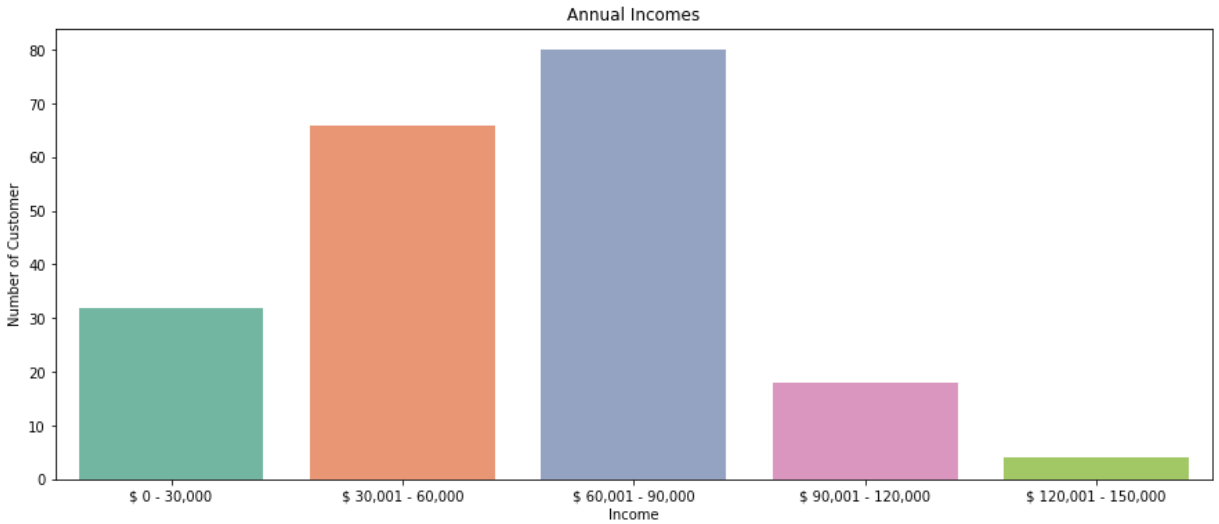
Nhìn vào biểu đồ nén của 2 thuộc tính điểm chi tiêu (Spending Score) và thuộc tính thu nhập hàng năm (Annual Income), chúng tôi rút ra nhận xét về 2

thuộc tính nói trên rằng mức điểm chi tiêu của khách hàng dành cho trung tâm mua sắm trung bình rơi vào khoảng 50 điểm và có số khách hàng đạt điểm trên mức trung bình cao hơn phần còn lại, về phần thu nhập hằng năm thì trung bình mức thu nhập hằng năm của khách hàng đến trung tâm mua sắm nằm ở khoảng 60.000$ nhưng số khách hàng có mức thu nhập trên trung bình lại cao hơn phần còn lại.

Chúng tôi quyết định vẽ thêm 2 biểu đồ cột nữa về thuộc tính Spending Score và Annual Income để có cái nhìn rõ ràng hơn về 2 thuộc tính trên.

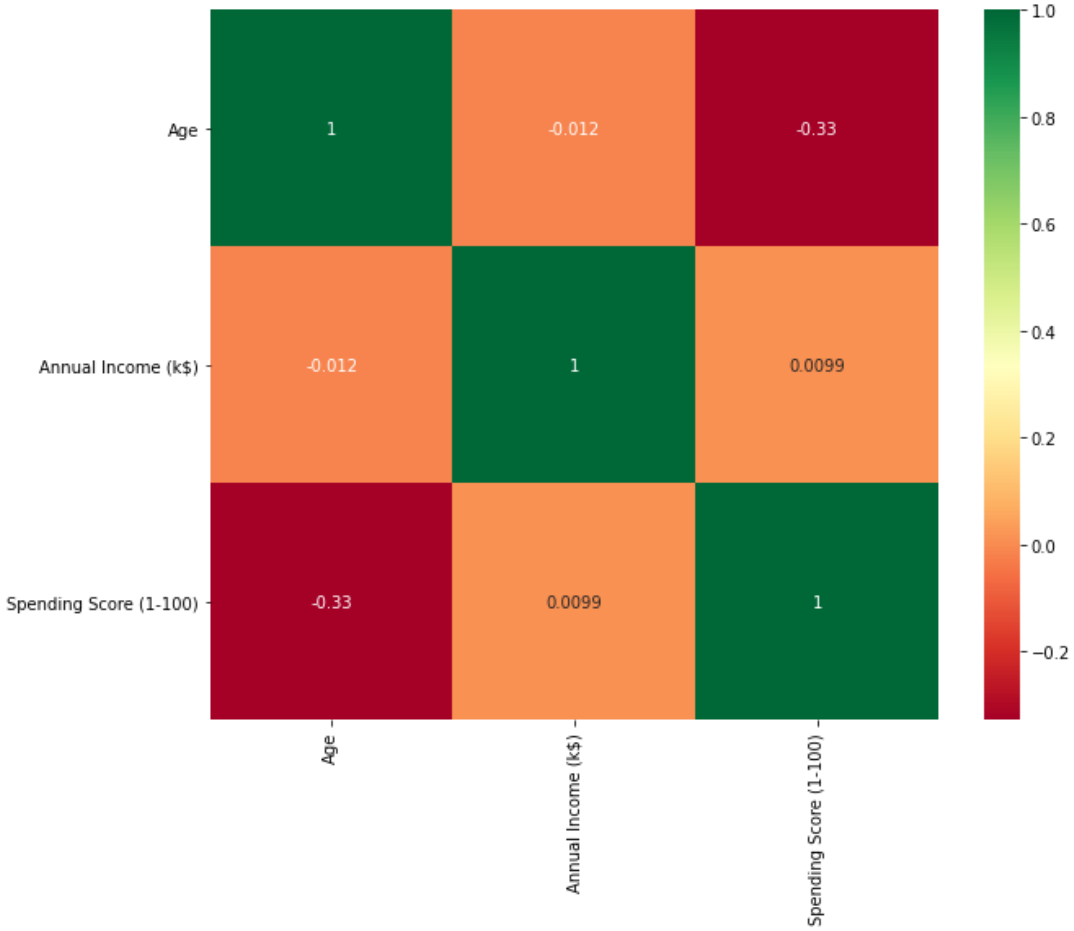


Hình 11. Biểu đồ cột thuộc tính Spending Score.



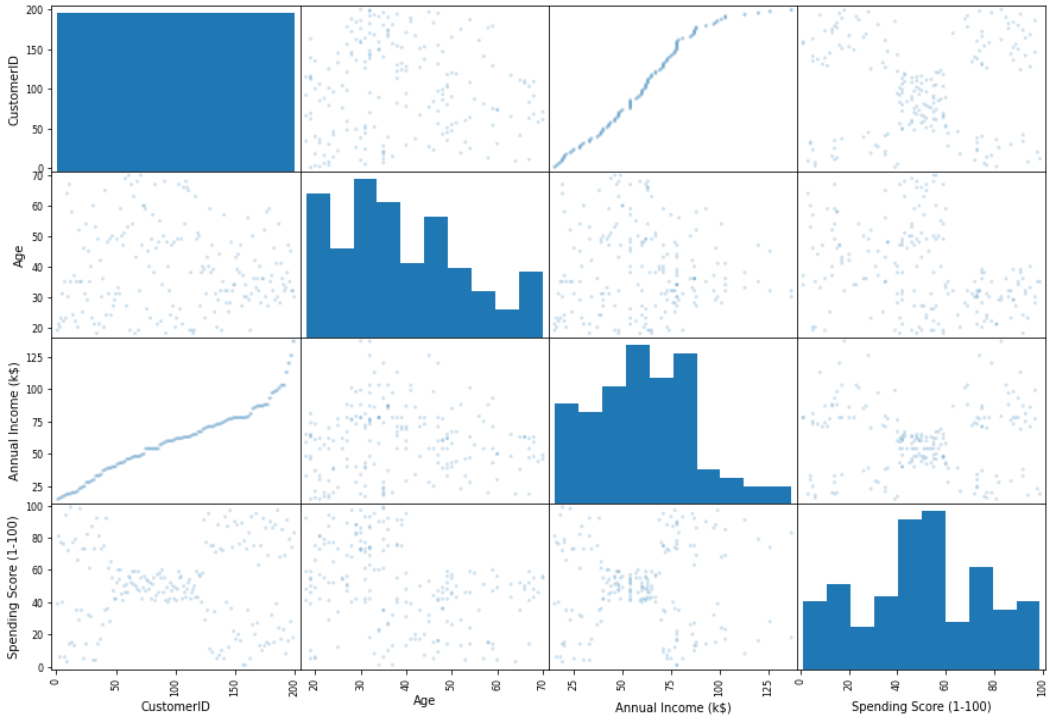
Hình 12. Biểu đồ cột thuộc tính Annual Income.

Để nhìn nhận về mối tương quan giữa các thuộc tính với nhau, chúng tôi đã vẽ ra một biểu đồ thể hiện mối tương quan giữa 3 thuộc tính là Age, Spending Score và Annual Income.



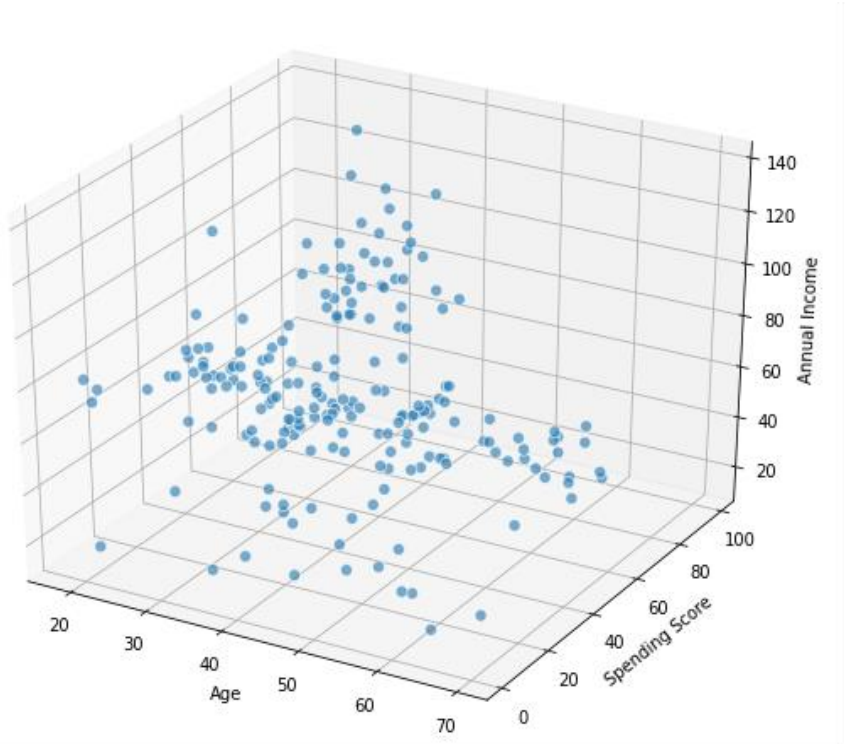
Hình 13. Mối tương quan giữa các thuộc tính.

Điều duy nhất chúng tôi thấy được trong biểu đồ này đó chính là sự tương quan nghịch biến giữa thuộc tính Age và thuộc tính Spending Score ở mức -0.33, nhưng mức điểm đó vẫn chưa thể giúp đưa ra bất kỳ kết luận gì về các thuộc tính trong bộ dữ liệu. Chúng tôi quyết định vẽ thêm các ma trận phân tán giữa các thuộc tính với nhau để có thể đưa ra kết luận cuối cùng.



Hình 14. Các ma trận phân tán giữa các thuộc tính.

Nhìn vào các ma trận phân tán trên chúng tôi thấy được rằng có một số cụm điểm dữ liệu xuất hiện trong biểu đồ giữa 2 thuộc tính Spending Score và Annual Income.



Hình 15. Biểu đồ 3D giữa 3 thuộc tính.

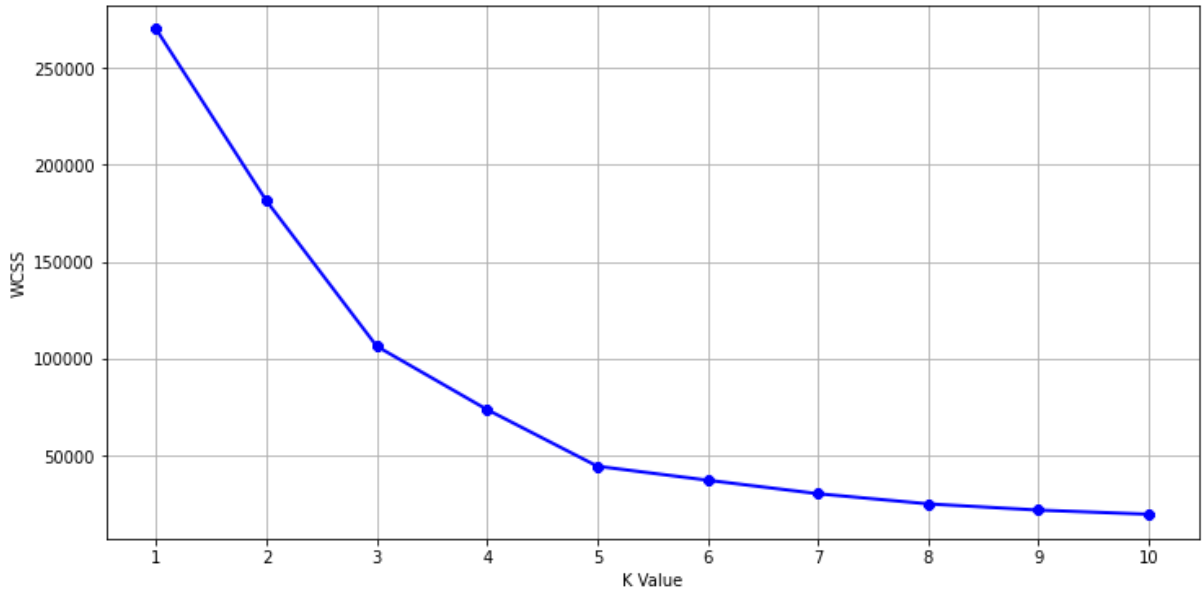
Từ biểu đồ trên chúng tôi có rút ra nhận xét rằng, từ hai thuộc tính Spending Score và Annual Income ta có thể phân ra thành những nhóm khách hàng nhất định với các đặc điểm chi tiêu khác nhau.

2. Huấn luyện mô hình.

2.1. Phân khúc khách hàng.

Từ các bước trực quan hoá dữ liệu và phân tích, chúng tôi quyết định chọn ra 2 thuộc tính để đưa vào mô hình phân khúc khách hàng, đó chính là 2 thuộc tính Spending Score và Annual Income.

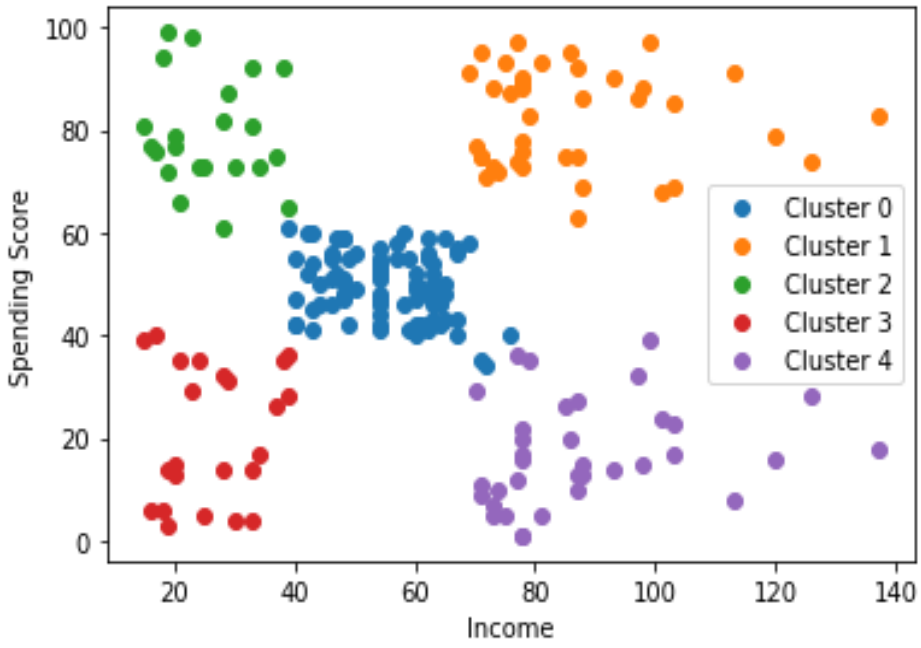
Với dự định sẽ sử dụng phương pháp K – mean clustering để tìm ra các nhóm khách hàng riêng biệt, nên bước đầu tiên chúng tôi thực hiện chính là chọn ra tham số *k* phù hợp cho mô hình. Để chọn ra tham số *k* cho mô hình, chúng tôi sử dụng phương pháp Khuỷu tay (Elbow Method) với giá trị hàm lỗi nhỏ nhất ở mức tối ưu.



Hình 16. Biểu đồ hàm lỗi với k là số cụm.

Qua biểu đồ trên, chúng tôi rút ra được nhận xét rằng, giá trị hàm lỗi giảm mạnh từ $k = 1$ đến $k = 5$ và bắt đầu giảm nhẹ từ $k = 6$ trở đi. Vì vậy, số cụm $k = 5$ là số cụm tối ưu cho mô hình.

Sau khi đã chọn được số lượng cụm tối ưu cho mô hình, chúng tôi tiếp tục bước tiếp theo của việc xây dựng mô hình đó là sử dụng thuật toán K – mean để phân các điểm dữ liệu vào các cụm tương ứng và trực quan hoá mô hình để có góc nhìn toàn vẹn về số lượng cụm và các điểm dữ liệu.



Hình 17. Mô hình phân khúc khách hàng.

Tiếp đến, bộ dữ liệu sẽ được thêm vào một cột thuộc tính *Segmentation,* là thuộc tính sẽ lưu nhãn của cụm mà điểm dữ liệu được phân bổ vào. Cột thuộc tính mới được thêm vào sẽ phục vụ cho việc học và dự đoán ở các bước tiếp theo.

	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Segmentation
0	Male	19	15	39	3
1	Male	21	15	81	4
2	Female	20	16	6	3
3	Female	23	16	77	4
4	Female	31	17	40	3

Hình 18. 5 điểm dữ liệu đầu tiên của bộ dữ liệu.

2.2. Dự đoán cụm khách hàng.

Sau khi đã có thuộc tính *Segmentation* trong bộ dữ liệu, chúng tôi tiến hành chia bộ dữ liệu thành 2 phần train và test với tỉ lệ 7:3. Sử dụng 3 phương pháp dự đoán là Decision Tree, Random Forest và KNN để xây dựng mô hình máy học và dự đoán kết quả.

Decision Tree

	Predicted	Actual
0	4	4
1	3	3
2	1	1
3	1	1
4	2	2

Random Forest

	Predicted	Actual
0	4	4
1	3	3
2	1	1
3	1	1
4	2	2

KNN (Neighbor = 5)

	Predicted	Actual
0	4	4
1	3	3
2	1	1
3	1	1
4	2	2

Bảng 1. Kết quả dự đoán so với thực tế của 3 mô hình

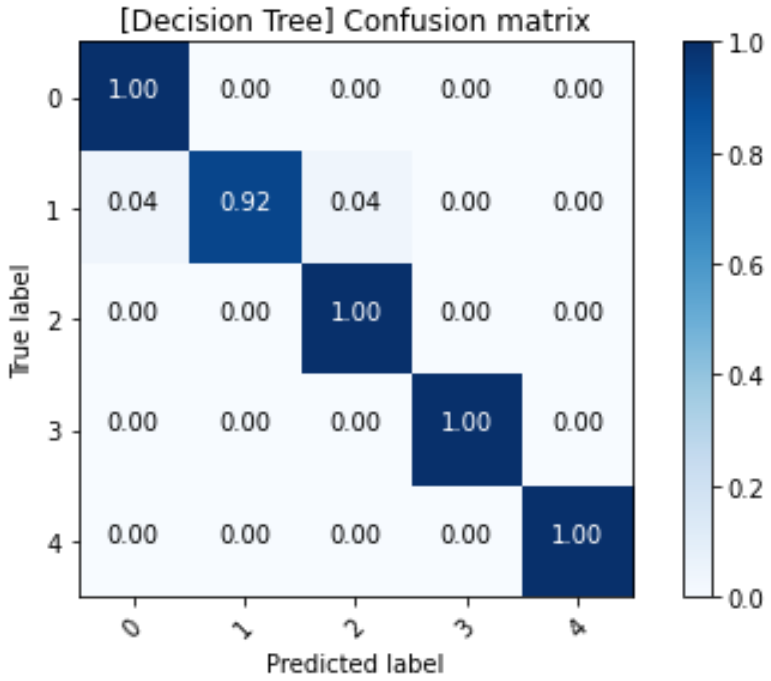
Qua quan sát 5 điểm dữ liệu đầu tiên, chúng tôi thấy được kết quả dự đoán của của 3 mô hình khá tốt, với giá trị dự đoán gần như chính xác với hầu hết các điểm dữ liệu trong bộ dữ liệu test, điều đó cho thấy cả 3 phương pháp đều chạy rất tốt trên bộ dữ liệu này. Sau quá trình xây dựng mô hình, chúng tôi đi đến phần tiếp theo là đánh giá hiệu suất mô hình để biết chính xác mô hình có hoạt động tốt hay không.

CHƯƠNG V: ĐÁNH GIÁ HIỆU SUẤT MÔ HÌNH

Chúng tôi sử dụng 4 chỉ số đánh giá mô hình là Accuracy, Precision, Recall, F1-Score để đánh giá 3 phương pháp được chúng tôi sử dụng là Decision Tree, Random Forest và KNN.

1. Phương pháp Decision Tree.

Confusion Matrix:



Hình 19. Confusion Matrix trên tập test – Decision Tree.

Theo ma trận trên, chúng tôi rút ra được 1 vài nhận xét rằng:

- Số điểm dữ liệu phân loại đúng: 10 + 22 + 12 + 7 + 7 = 58.
- Số điểm dữ liệu phân loại sai: 1 + 1 = 2.
- Tỉ lệ điểm dữ liệu phân loại đúng: 58 / 60 = 0.96667.

⇨ Accuracy: 0.96667.

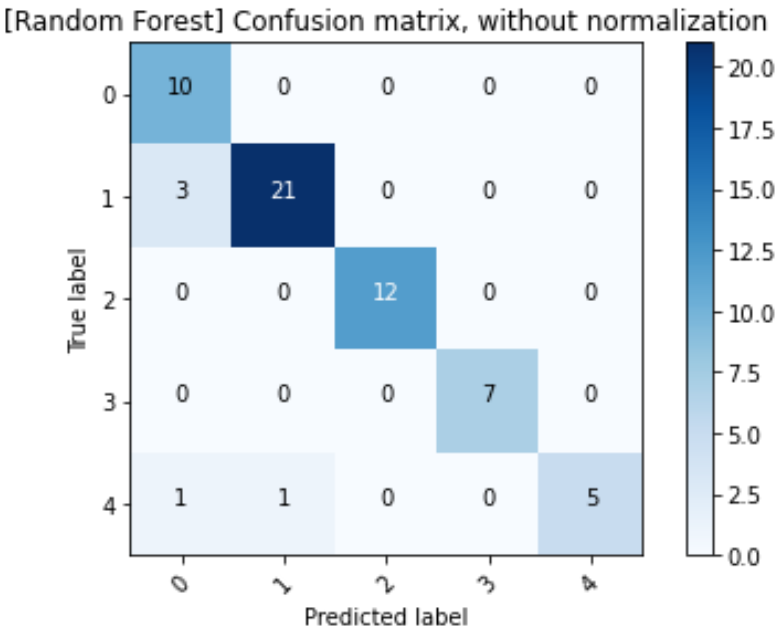
Classification report:

	Precision	Recall	F1-Score	Support
0	0.91	1.00	0.95	10
1	1.00	0.92	0.96	24
2	0.92	1.00	0.96	12
3	1.00	1.00	1.00	7
4	1.00	1.00	1.00	7
Macro Average	0.97	0.98	0.97	60
Weighted Average	0.97	0.97	0.97	60

Bảng 2. Classification report trên tập test - Decision Tree.

2. Phương pháp Random Forest.

Confusion Matrix:



Hình 20. Confusion Matrix trên tập test – Random Forest.

Theo ma trận trên, chúng tôi rút ra được 1 vài nhận xét rằng:

- Số điểm dữ liệu phân loại đúng: 10 + 21 + 12 + 7 + 5 = 55.
- Số điểm dữ liệu phân loại sai: 3 + 1 + 1 = 5.

- Tỉ lệ điểm dữ liệu phân loại đúng: 55 / 60 = 0.91666.

⇨ Accuracy: 0. 91666.

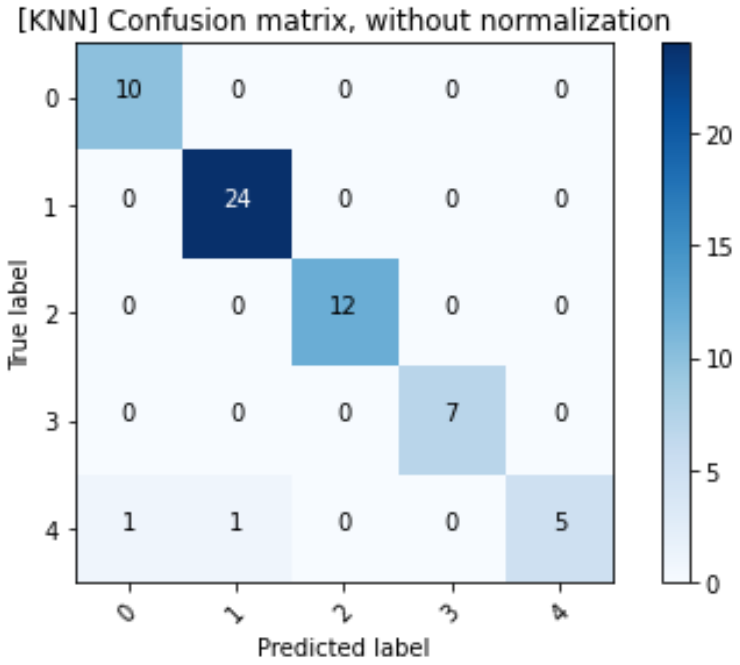
Classification report:

	Precision	Recall	F1-Score	Support
0	0.71	1.00	0.83	10
1	0.95	0.88	0.91	24
2	1.00	1.00	1.00	12
3	1.00	1.00	1.00	7
4	1.00	0.71	0.83	7
Macro Average	0.93	0.92	0.92	60
Weighted Average	0.93	0.92	0.92	60

Bảng 3. Classification report trên tập test - Random Forest.

3. Phương pháp KNN.

Confusion Matrix:



Hình 21. Confusion Matrix trên tập test – KNN.

Theo ma trận trên, chúng tôi rút ra được 1 vài nhận xét rằng:

- Số điểm dữ liệu phân loại đúng: 10 + 24 + 12 + 7 + 5 = 58.
- Số điểm dữ liệu phân loại sai: 1 + 1 = 2.
- Tỉ lệ điểm dữ liệu phân loại đúng: 58 / 60 = 0.96667.

⇨ Accuracy: 0. 96667.

Classification report:

	Precision	**Recall**	**F1-Score**	**Support**
0	0.91	1.00	0.95	10
1	0.96	1.00	0.98	24
2	1.00	1.00	1.00	12
3	1.00	1.00	1.00	7
4	1.00	0.71	0.83	7
Macro Average	0.97	0.94	0.95	60
Weighted Average	0.97	0.97	0.96	60

Bảng 4. Classification report trên tập test – KNN.

CHƯƠNG VI: KẾT LUẬN

Trong bài báo cáo này, chúng tôi đã trình bày về các phương pháp và quá trình nhằm phân cụm cho các khách hàng của một trung tâm mua sắm. Chúng tôi đã tiến hành phân tích chi tiết về bộ dữ liệu bằng cách trực quan hoá từng thuộc tính của bộ dữ liệu và quan sát chúng, từ đó rút ra được các thuộc tính có độ quan trọng cao trong việc xây dựng mô hình. Từ đó chúng tôi đã thực hiện việc xây dựng mô hình máy học bằng cách kết hợp nhiều phương pháp khác nhau nhằm giải quyết bài toán được đặt ra và thấy được các mô hình đều hoạt động tốt trên bộ dữ liệu và bài toán này.

	Accuracy	**Precision**	**Recall**	**F1-score**
Decision Tree	0.96667	0.97	0.97	0.97
Random Forest	0.91666	0.93	0.92	0.92
KNN	0.96667	0.97	0.97	0.96

Bảng 5. Bảng so sánh kết quả giữa các mô hình.

Phân khúc khách hàng ngày càng quan trọng trong thời kỳ nền kinh tế dịch vụ chiếm được ưu thế hơn so với những nền kinh tế còn lại. Việc phân khúc khách hàng sẽ giúp các nhà kinh doanh tìm được những khách hàng tiềm năng của họ, từ đó họ có thể đầu tư một cách tốt nhất cho dịch vụ dành cho những khách hàng họ nhắm đến. Bài toán này còn giúp giảm trừ sự hao phí tài nguyên và những mất mát không đáng có cho các nhà kinh doanh, khi mà họ phải sử dụng các phép thử để có thể tìm ra nhóm khách hàng mà họ muốn nhắm đến rồi từ đó mới quyết định hướng phát triển cho mình, trong khi bài toán có thể giúp họ tìm ra kết quả một cách đơn giản, hiệu quả và ít tốn kém hơn.

Lời cuối cùng, chúng em xin chân thành cảm ơn các thầy đã tận tình chỉ bảo, giúp đỡ chúng em ngay từ những buổi học đầu tiên cho đến nay và trong suốt quá trình hoàn thiện đồ án cuối kỳ này.

DANH MỤC TÀI LIỆU THAM KHẢO

1. Vũ Hữu Tiệp (2016). *Machine Learning Cơ Bản.*
2. Subiz (2017). "*Phân khúc khách hàng để Marketing hiệu quả*". Truy cập ngày 1/6/2021.

   https://subiz.com.vn/blog/phan-khuc-khach-hang.html

3. Google Developer (2020). "*k-Means Advantages and Disadvantages*". Truy cập ngày 20/5/2021.

   https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

4. Data Flair (2020). "*Data Science Project – Customer Segmentation using Machine Learning in R*". Truy cập ngày 20/5/2021.

   https://data-flair.training/blogs/r-data-science-project-customer-segmentation/?fbclid=IwAR1PM8duI_CoRulBys3la9Cmz0OXuYCQMbMPwJ-t_ubib66Q8cVETZpU3YU

5. Business Science (2016). "*Customer Segmentation Part 1: K Means Clustering*". Truy cập ngày 20/5/2021.

   https://www.business-science.io/business/2016/08/07/CustomerSegmentationPt1.html

6. Kaggle (2019). "*Customer Segmentation Using KMeans and Machine Learning using Decision Tree and Random Forest*". Truy cập ngày 20/5/2021.

   https://www.kaggle.com/omodencode/clustering-and-predicting-customer-categories?fbclid=IwAR0Mzt-BsSA6kO_mp7U90FwAB6ZoKbx_giYzGrrhbVLtDPW2Cido5Vk6tJw

7. Machine Learning cơ bản (2017). "*K-means Clustering*". Truy cập ngày 3/6/2021.

   https://machinelearningcoban.com/2017/01/01/kmeans/

8. Towards Datascience (2019). "*How Does k-Means Clustering in Machine Learning Work?*". Truy cập ngày 3/6/2021.

   https://towardsdatascience.com/how-does-k-means-clustering-in-machine-learning-work-fdaaaf5acfa0

9. RStudio. "*Chapter 4: K-Means Clustering (Cluster Analysis in R)*". Truy cập ngày 3/6/2021.

https://rstudio-pubs-static.s3.amazonaws.com/323353_31f9a891bbf24fc0b42ff835344c2d1f.html

10. Lập trình không khó (2018). "*Thuật toán K-Means (K-Means clustering) và ví dụ*". Truy cập ngày 3/6/2021.

    https://nguyenvanhieu.vn/thuat-toan-phan-cum-k-means/#ban-doc-thuc-nghiem

11. Muthukrishnan (2018). "*Mathematics behind K-Mean Clustering algorithm*". Truy cập ngày 4/6/2021.

    https://muthu.co/mathematics-behind-k-mean-clustering-algorithm/

12. Ứng dụng công nghệ thông tin (2016). "*Thuật toán KMeans*". Truy cập ngày 4/6/2021.

    http://ungdung.khoa-hnvd.com/Hoc_thuat/KMeans.html

13. Machine Learning cơ bản (2018). "*Decision Trees (1): Iterative Dichotomiser 3*". Truy cập ngày 4/6/2021.

    https://machinelearningcoban.com/2018/01/14/id3/#-y-tuong

14. Big Data Solution. "*Thuật toán cây quyết định (P.1): Classification & Regression tree (CART)*". Truy cập ngày 4/6/2021.

    https://bigdatauni.com/tin-tuc/thuat-toan-cay-quyet-dinh-classification-regression-tree-cart-p-1.html

15. Trí tuệ nhân tạo (2019). "*Cây Quyết Định (Decision Tree)*". Truy cập ngày 6/6/2021.

    https://trituenhantao.io/kien-thuc/decision-tree/

16. Viblo (2018). "*Phân lớp bằng Random Forests trong Python*". Truy cập ngày 9/6/2021.

    https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz#_cac-tinh-nang-quan-trong-trong-scikit-learn-4

17. Machine Learning cơ bản. "*Random Forest algorithm*". Truy cập ngày 9/6/2021.

    https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html

18. Forum Machine Learning cơ bản (2019). "*Mối quan hệ đánh đổi giữa bias và variance*". Truy cập ngày 9/6/2021.

https://forum.machinelearningcoban.com/t/moi-quan-he-danh-doi-giua-bias-va-variance/4173

19. Machine Learning cơ bản (2017). "*K-nearest neighbors*". Truy cập ngày 10/6/2021.

    https://machinelearningcoban.com/2017/01/08/knn/

20. Big Data Solution. "*Thuật toán KNN và ví dụ đơn giản trong ngành ngân hàng*". Truy cập ngày 10/6/2021.

    https://bigdatauni.com/tin-tuc/thuat-toan-knn-va-vi-du-don-gian-trong-nganh-ngan-hang.html

21. Viblo (2019). "*KNN (K-Nearest Neighbors)*". Truy cập ngày 11/6/2021.

    https://viblo.asia/p/knn-k-nearest-neighbors-1-djeZ14ejKWz

22. Machine Learning cơ bản (2018). "*Các phương pháp đánh giá một hệ thống phân lớp*". Truy cập ngày 11/6/2021.

    https://machinelearningcoban.com/2017/08/31/evaluation/

23. RStudio (2018). "*Đánh giá mô hình hồi quy*". Truy cập ngày 12/6/2021.

    http://rstudio-pubs-static.s3.amazonaws.com/445130_e065fc3cceaf4393ba8011e3d7e106b5.html

24. Noron (2018). "*Tìm hiểu về Confusion matrix trong Machine Learning?*". Truy cập ngày 13/6/2021.

    https://www.noron.vn/post/tim-hieu-ve-confusion-matrix-trong-machine-learning-1fz9nhqo5ux

25. Math2IT (2019). "*Hiểu confusion matrix*". Truy cập ngày 13/6/2021.

    https://math2it.com/hieu-confusion-matrix/