Hiệu suất và ứng dụng của Decision Tree, Naive Bayes và Linear Regression trong phân loại khách hàng thẻ tín dụng

1st Lê Nguyễn Minh Trung
Trường Đại học Công Nghệ Thông Tin
19521061@gm.uit.edu.vn

2nd Võ Nữ Diễm Trang
Trường Đại học Công Nghệ Thông Tin
20521013@gm.uit.edu.vn

3rd Trần Gia Phong
Trường Đại học Công Nghệ Thông Tin
20521748@gm.uit.edu.vn

Tóm tắt

Đồ án tập trung nghiên cứu và so sánh hiệu suất của các thuật toán phân loại trong Data Mining. Chúng em áp dụng Decision Tree, Naive Bayes và Linear Regression để phân loại khách hàng thẻ tín dụng. Sử dụng một tập dữ liệu từ Kaggle, chúng em đánh giá các thuật toán dựa trên độ chính xác và khả năng dự đoán.

Mục tiêu là so sánh hiệu suất của các thuật toán trên bài toán phân loại khách hàng thẻ tín dụng. Kết quả giúp hiểu rõ hơn về tính hiệu quả và ứng dụng của từng thuật toán, cũng như lựa chọn phù hợp cho các tập dữ liệu khác nhau.

Đồ án cung cấp kiến thức và kinh nghiệm trong Data Mining cho sinh viên. Kết quả có thể áp dụng vào giải quyết các vấn đề thực tế.

Từ khóa: *Data Mining, Thuật toán phân loại, Phân loại khách hàng thẻ tín dụng, Decision Tree, Naive Bayes, Linear Regression.*

I. GIỚI THIỆU

Trong lĩnh vực Data Mining, việc khai thác tri thức từ dữ liệu là một công việc quan trọng, đóng vai trò then chốt trong việc phân tích và rút trích thông tin hữu ích từ các nguồn dữ liệu đa dạng. Trong bối cảnh ấy, đồ án này tập trung vào nghiên cứu và so sánh hiệu suất của ba giải thuật phân loại quan trọng: Decision Tree, Naive Bayes và Linear Regression. Đây là một nghiên cứu mang tính ứng dụng, nhằm tạo ra sự thực nghiệm và hiểu rõ hơn về khả năng áp dụng của các giải thuật trong lĩnh vực Data Mining.

Lý do chúng em quan tâm đến đề tài này xuất phát từ sự tò mò và nhu cầu hiểu biết sâu hơn về Data Mining, đặc biệt là việc áp dụng các giải thuật phân loại lên các tập dữ liệu thực tế. Đồ án này cung cấp cho chúng em cơ hội để thực nghiệm các phương pháp và công cụ trong lĩnh vực này, và giúp xây dựng một nền tảng kiến thức rộng và sâu hơn về Data Mining cho tương lai.

Mục tiêu của đồ án là nghiên cứu và so sánh hiệu suất của ba giải thuật quan trọng trong việc phân loại. Để đạt được mục tiêu này, chúng em đã chọn một bài toán phân lớp cụ thể là dự đoán trạng thái của khách hàng về việc rời bỏ hoặc tiếp tục sử dụng dịch vụ thẻ tín dụng. Bài toán này được nghiên cứu trên một dataset thu thập từ nguồn Kaggle là "Predicting Credit Card Customer Segmentation". Chúng em đã áp dụng ba giải thuật Decision Tree, Naive Bayes và Linear Regression để xây dựng các mô hình phân loại và đánh giá hiệu suất của chúng.

Một thách thức quan trọng trong nghiên cứu này là lựa chọn giải thuật phù hợp cho từng bài toán trong dataset. Quá trình lựa chọn đúng giải thuật có thể ảnh hưởng đáng kể đến kết quả và hiệu suất của hệ thống phân loại. Bằng cách nghiên cứu và so sánh hiệu suất của các giải thuật Decision Tree, Naive Bayes và Linear Regression trong bài toán phân loại khách hàng, chúng em hy vọng sẽ đưa ra được những kết

luận về sự hiệu quả và ứng dụng của từng giải thuật, từ đó định hướng lựa chọn giải thuật phù hợp trong các tình huống và dataset khác nhau.

Kết quả dự kiến từ nghiên cứu này sẽ mang lại sự hiểu biết sâu sắc hơn về việc áp dụng các giải thuật phân loại trong lĩnh vực Data Mining, đồng thời cung cấp một bối cảnh rõ ràng về quá trình xử lý dữ liệu và phân tích thông tin từ các dataset thực tế. Kết quả này có thể hữu ích cho sinh viên và các nhà nghiên cứu muốn nắm bắt và ứng dụng các kỹ thuật Data Mining trong công việc của mình.

Phần tiếp theo của bài báo cáo này sẽ trình bày chi tiết về cơ sở lý thuyết, các phương pháp và quá trình thực hiện đồ án, bao gồm mô tả về dataset, tiền xử lý dữ liệu, triển khai giải thuật và đánh giá hiệu suất. Cuối cùng, chúng em sẽ tổng kết kết quả và đưa ra nhận xét quan trọng về việc áp dụng các giải thuật phân loại trong lĩnh vực Data Mining.

II. CƠ SỞ LÝ THUYẾT

A. Decision Tree [1]

Cấu trúc của cây quyết định

Khi xây dựng cây quyết định, tốc độ giảm dần của entropy thông tin được sử dụng để xác định biến nhánh tốt nhất và ngưỡng phân đoạn. Entropy thông tin đại diện cho mức độ tạp chất của một tập dữ liệu và được xác định dựa trên Mitchell (1997)[2] là:

$$Entropy(D) = -\sum_{k=1}^{m} P_k log_2 P_k$$

D là tập dữ liệu huấn luyện với kích thước mẫu m và Pk là xác suất của từng loại mẫu. Các tỷ lệ khuếch đại thông tin được sử dụng để đo lường sự khác biệt về entropy thông tin của các tập dữ liệu theo các phương pháp phân loại. Nếu ta chọn biến C để chia tập dữ liệu D thành n tập con thì thông tin tỷ lệ khuếch đại được xác định dựa trên Quinlan (1996)[3] là:

$$Gain\ ratio\ (D, C) = \frac{Entropy(D) - Entropy(D|C)}{Entropy(C)}$$

Thuật toán C5.0 chọn thuộc tính có tỷ lệ thu được thông tin tối đa là điểm phân tách, thiết lập một số nhánh theo giá trị của thuộc tính này và thu được một số tập hợp con. Quá trình lựa chọn này được lặp lại cho đến khi tập hợp con cuối cùng chỉ chứa dữ liệu của cùng một loại, để thực hiện phân loại quy nạp cho dữ liệu (Che, Liu, Rasheed, & Tao, 2011)[4].

Cắt tỉa cây quyết định

Thuật toán C5.0 sử dụng phương pháp cắt tỉa sau để tỉa lá từng lớp và từ các nút lá. Sau khi xây dựng cây quyết định, tập dữ liệu được đệ quy tới từng nút lá của cây theo mô hình cây quyết định đã huấn luyện. Sai số bình phương trung bình của các nút dữ liệu có và không có lá được tính toán. Nếu sai số bình phương trung bình giảm sau khi cắt tỉa thì nút đó đã bị cắt, nếu không thì nó đã được giữ lại (Quinlan, 2019)[5].

Đánh giá cây quyết định

Chúng em lấy 80% dữ liệu mẫu (n = 8101) làm dữ liệu huấn luyện và 20% còn lại (n = 2026) như dữ liệu thử nghiệm. Liệu mô hình được xây dựng bởi dữ liệu huấn luyện có phù hợp với dữ liệu mới hay không. được phản ánh bởi các dữ liệu thử nghiệm. Chất lượng mô hình được đánh giá dựa trên precision, accuracy và recall (Han etal., 2019)[6].

Accuracy đề cập đến tỷ lệ các trường hợp được phân loại chính xác liên quan đến tổng số mẫu kích cỡ. Precision đề cập đến kết quả dự đoán, cho biết có bao nhiêu mẫu xác thực dự đoán là các mẫu xác thực thực sự. Recall áp dụng cho mẫu thực tế, cho biết có bao nhiêu mẫu xác thực trong mẫu được dự đoán chính xác.

B. Naïve Bayes [7]

Đây là thuật toán điều tra dữ liệu số học và không giám sát. Đồng thời, đây là phương pháp phân loại thông tin trùng lặp thành cụm đồng nhất. Nó được sử dụng để vận hành tập dữ liệu lớn thành khám phá sự liên kết và mô hình ẩn đã giúp tạo ra hiệu quả quyết định một cách nhanh chóng. Mỗi cụm là nhóm thông

tin các nhãn có liên quan đến nhau được đặt bên trong cụm tương ứng nhưng không liên quan đến các mục trong cụm khác cụm.

Nó thường được áp dụng phương pháp trong khai thác dữ liệu hàn lâm dự đoán nhóm tồn tại trong tập dữ liệu. Phân loại cũ là phương pháp thường được áp dụng trong khai thác dữ liệu học thuật dự đoán nhóm tồn tại trong tập dữ liệu[8].

Một bộ phân loại Naïve Bayes là phân loại xác suất đơn giản được thành lập trên công thức Bayes. NBC (Naïve Bayes classifiers) đã được đào tạo cực kỳ nhanh chóng. Thật dễ hiểu, dữ liệu đào tạo cần thiết để ước tính tham số, không đáp ứng với các tính năng không liên quan, xử lý tốt dữ liệu thực và riêng biệt[9, 10].

C. Linear Regression [11]

Hồi quy tuyến tính là một trong những thuật toán học máy. Nó dựa trên kỹ thuật học tập có giám sát, là một trong những thuật toán được biết đến rộng rãi và dễ hiểu ngay cả bởi người không quen thuộc với thuật toán máy học.

Ngay trong tên đã cho thấy hồi quy tuyến tính thực hiện hồi quy. Nó xác định mối quan hệ giữa hai biến bằng cách khớp đường hồi quy với dữ liệu.

Một trong hai biến là biến phụ thuộc và phụ thuộc vào biến còn lại, biến đó được gọi là biến độc lập. Ta nên đảm bảo rằng có tồn tại mối quan hệ giữa các biến phụ thuộc và độc lập trước khi triển khai mô hình. Sức mạnh của mối quan hệ giữa các biến có thể thể hiện bằng cách sử dụng biểu đồ phân tán (scatterplot).

Đường hồi quy tuyến tính được biểu diễn dưới dạng:

$$Y=a*X+b$$

- Biến phụ thuộc Y
- a-slope
- Biến độc lập X
- b-Intercept

Với đường hồi quy phù hợp nhất với dữ liệu, tỷ lệ lỗi giữa giá trị dự đoán và giá trị thực có thể được giảm thiểu. Hồi quy tuyến tính được phân thành hai loại. một trong số đó là Hồi quy tuyến tính đơn giản, trong đó chỉ có một biến độc lập được sử dụng và loại hồi quy thứ hai là Hồi quy tuyến tính bội.

III. MÔ TẢ TẬP DỮ LIỆU

Chúng em lấy bộ dữ liệu từ Kaggle https://www.kaggle.com/datasets/thedevastator/predicting-credit-card-customer-attrition-with-m?resource=download có tên là Predicting Credit Card Customer Segmentation. Bộ dữ liệu gồm 10127 dòng và 23 cột thuộc tính được mô tả trong bảng như sau:

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	Clientnum	Int	Số định danh của khách hàng
2	Attrition_Flag	Boolean	Cho biết khách hàng có rời bỏ hay không.
3	Customer_Age	Int	Tuổi của khách hàng
4	Gender	String	Giới tính của khách hàng
5	Dependent_count	Int	Số người phụ thuộc mà khách hàng
6	Education_level	String	Trình độ học vấn của khách hàng
7	Marital_Status	String	Tình trạng hôn nhân của khách hàng
8	Income_Category	String	Loại thu nhập của khách hàng
9	Card_Category	String	Loại thẻ của khách hàng
10	Months_on_book	Int	Số tháng khách hàng đã sử dụng dịch vụ thẻ tín dụng tính từ khi khách hàng mở tài khoản
11	Total_Relationship_Count	Int	Tổng số mối quan hệ giữa khách hàng và nhà cung cấp thẻ tín dụng
12	Months_Inactive_12_mon	Int	Số tháng khách hàng không hoạt động trong 12 tháng qua
13	Contacts_Count_12_mon	Int	Số lượng liên lạc của khách hàng trong 12 tháng qua
14	Credit_Limit	Int	Hạn mức tín dụng của khách hàng
15	Total_Revolving_Bal	Int	Tổng số dư quay vòng của khách hàng
16	Avg_Open_To_Buy	Int	Tỷ lệ mua trung bình của khách hàng
17	Total_Amt_Chng_Q4_Q1	Int	Tổng số tiền thay đổi từ quý 4 sang quý 1
18	Total_Trans_Amt	Int	Tổng số tiền giao dịch
19	Total_Trans_Ct	Int	Tổng số lượng giao dịch
20	Total_Ct_Chng_Q4_Q1	Int	Tổng số thay đổi từ quý 4 sang quý 1
21	Avg_Utilization_Ratio	Int	Tỷ lệ sử dụng trung bình của khách hàng
22	Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1	Float	Sử dụng Naïve Bayes để dự đoán liệu ai đó có rời đi hay không dựa trên các đặc điểm.
23	Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2	Float	Sử dụng Naïve Bayes phân loại loại thẻ của khách hàng dựa vào số lượng liên lạc của khách hàng trong 12 tháng qua và trình độ học vấn.

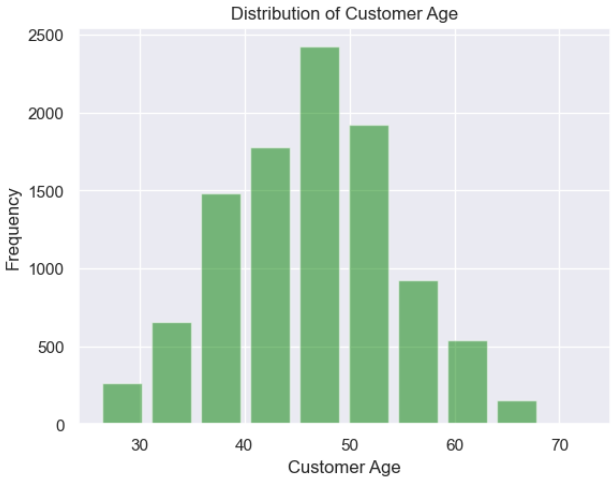
Bảng 1 Mô tả tập dữ liệu

Trong đồ án này, chúng em chỉ sử dụng 21 thuộc tính đầu tiên để phân tích, và Attrition_Flag là thuộc tính quyết định trong bài toán phân loại này.

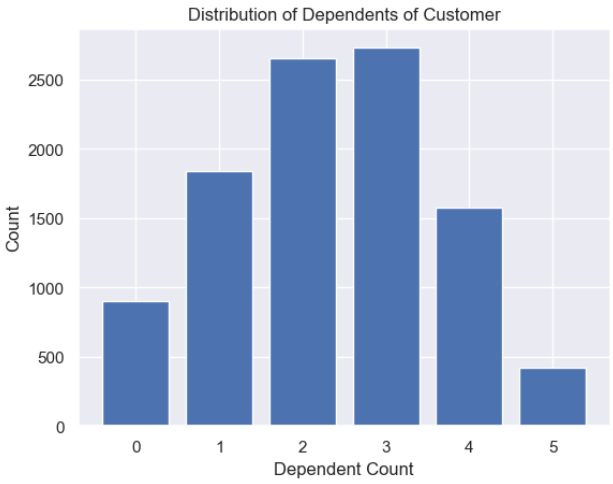
Phân bố dữ liệu

Phân bố dữ liệu (data distribution) là một khía cạnh quan trọng trong mô tả tập dữ liệu. Nó cho biết sự phân phối và phân loại của các mẫu dữ liệu trong tập dữ liệu. Thông qua phân bố dữ liệu, chúng ta có thể hiểu được tỷ lệ các lớp hoặc nhãn trong tập dữ liệu, tỷ lệ các giá trị thuộc tính và các thông số thống kê khác. Việc hiểu rõ phân bố dữ liệu giúp chúng ta đánh giá khả năng áp dụng các giải thuật phân loại và xác định các thực hành phù hợp cho quá trình huấn luyện và đánh giá mô hình.

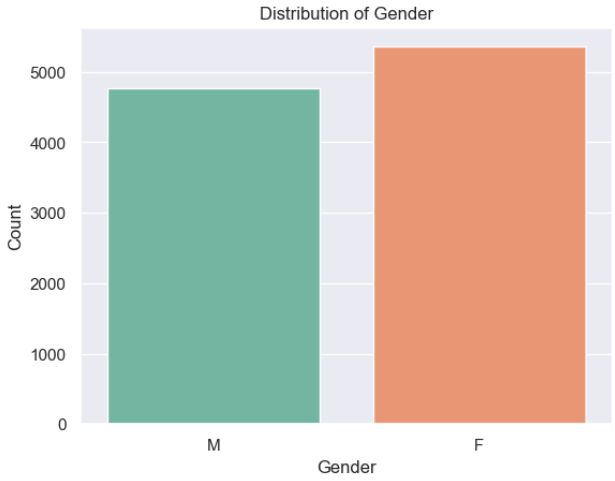
Dưới đây là các biểu đồ biểu diễn sự phân bố dữ liệu của từng thuộc tính được phân tích:



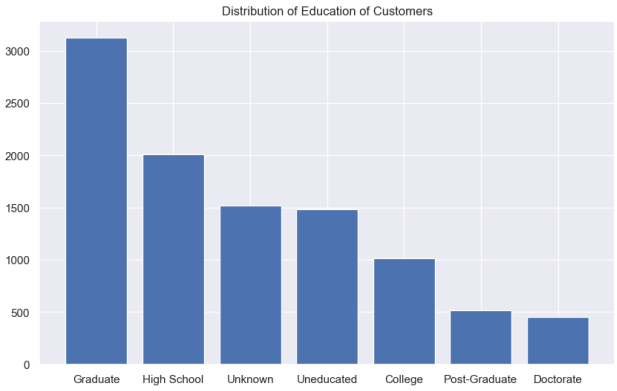
Hình 1 Biểu đồ phân bố thuộc tính Customer_Age



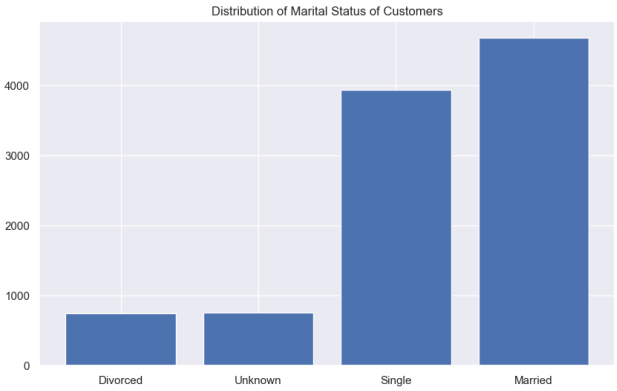
Hình 2 Biểu đồ phân bố thuộc tính Dependent_count



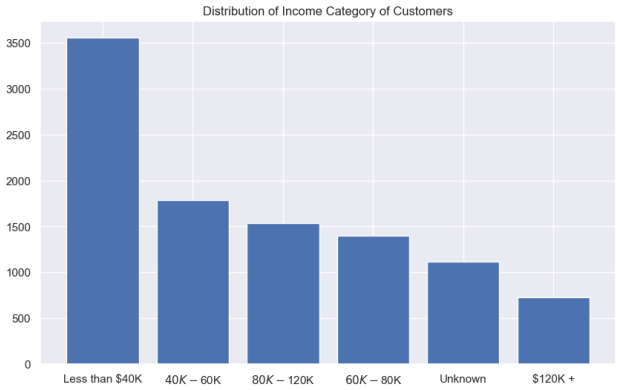
Hình 3 Biểu đồ phân bố thuộc tính Gender



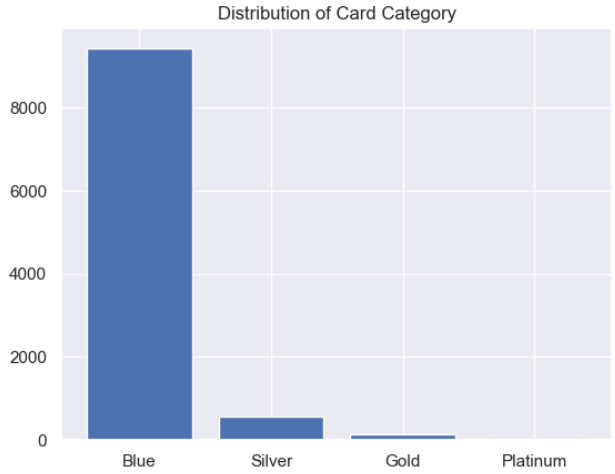
Hình 4 Biểu đồ phân bố thuộc tính Education_Level



Hình 5 Biểu đồ phân bố thuộc tính Marital_Status

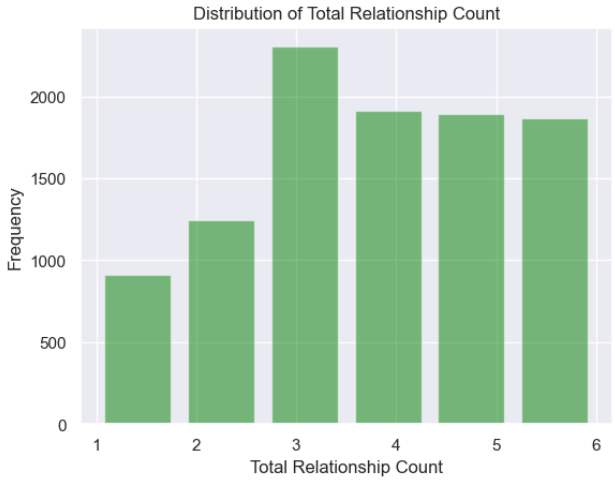


Hình 6 Biểu đồ phân bố thuộc tính Income_Category

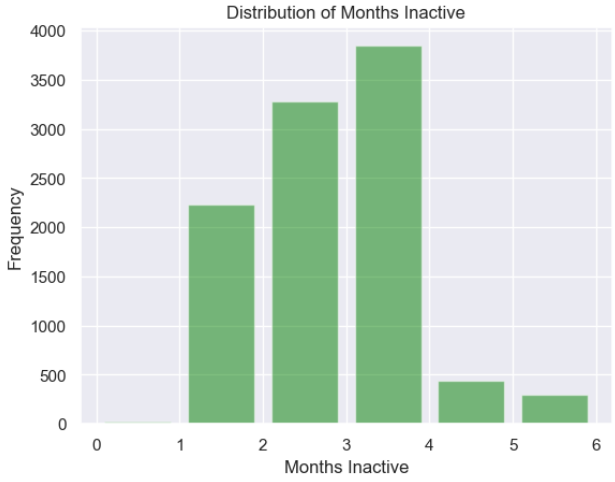


Hình 7 Biểu đồ phân bố thuộc tính Card_Category

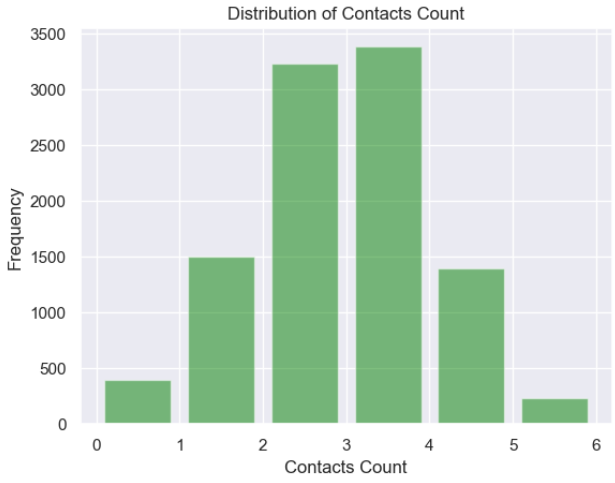
Hình 8 Biểu đồ phân bố thuộc tính Months_on_book



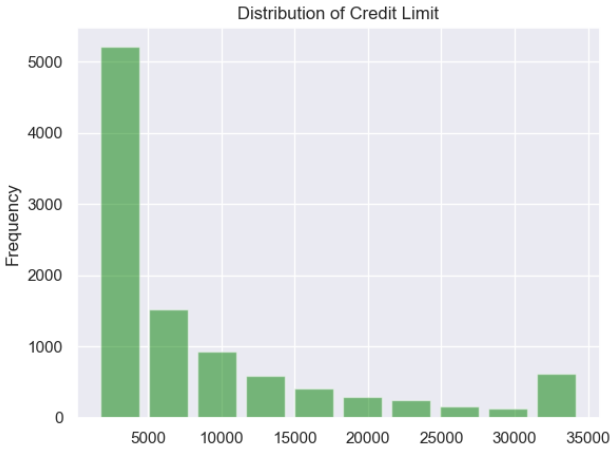
Hình 9 Biểu đồ phân bố thuộc tính Total_Relationship_Count



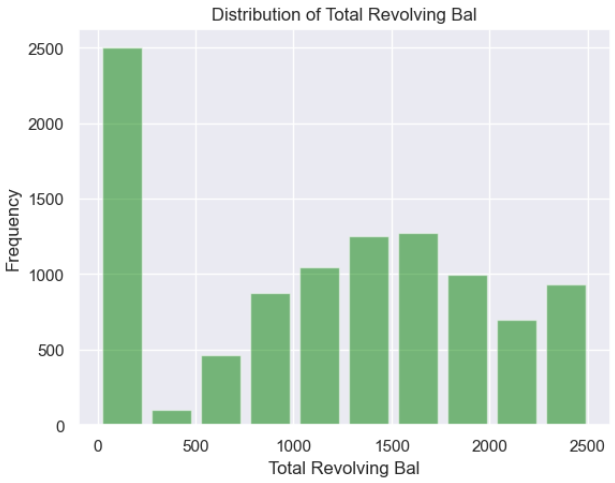
Hình 10 Biểu đồ phân bố thuộc tính Months_Inactive_12_mon



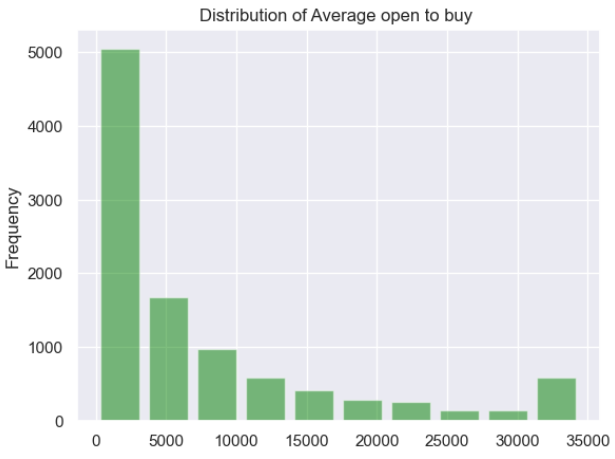
Hình 11 Biểu đồ phân bố thuộc tính Contacts_Count_12_mon



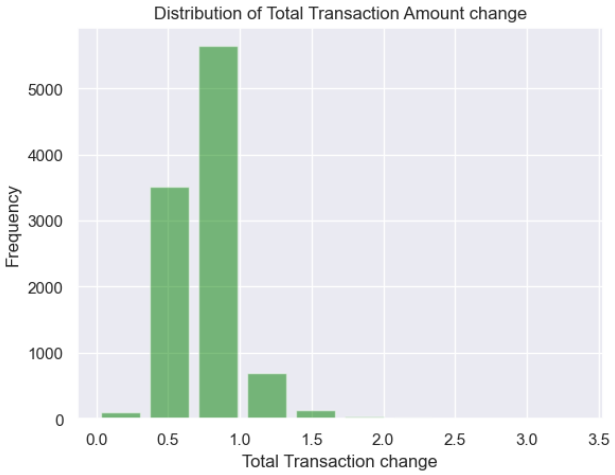
Hình 12 Biểu đồ phân bố thuộc tính Credit_Limit



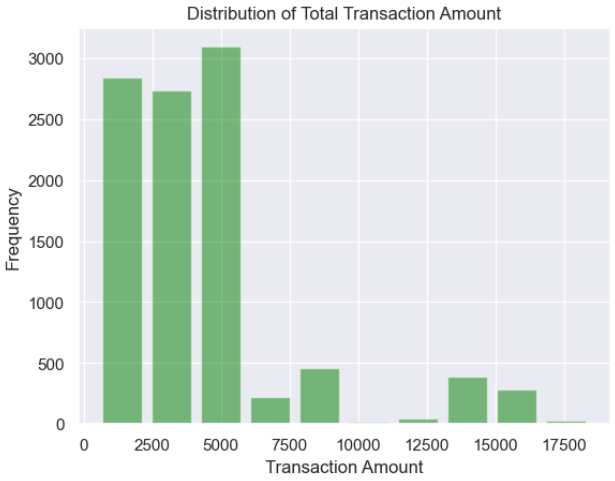
Hình 13 Biểu đồ phân bố thuộc tính Total_Revolving_Bal



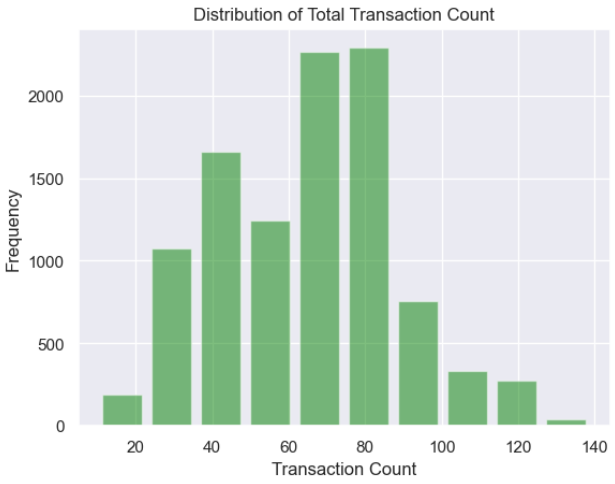
Hình 14 Biểu đồ phân bố thuộc tính Avg_Open_To_Buy



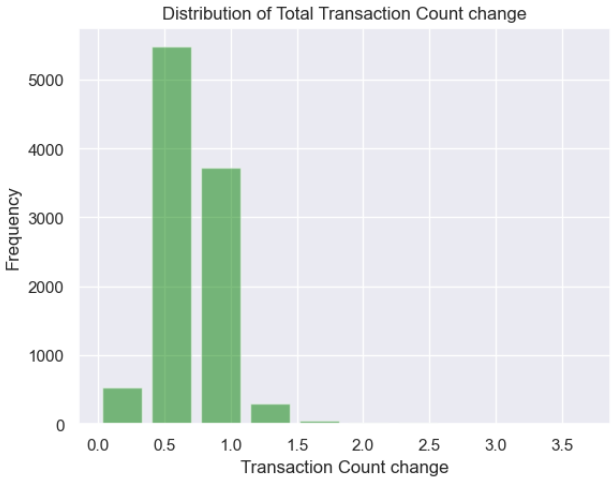
Hình 15 Biểu đồ phân bố thuộc tính Total_Amt_Chng_Q4_Q1



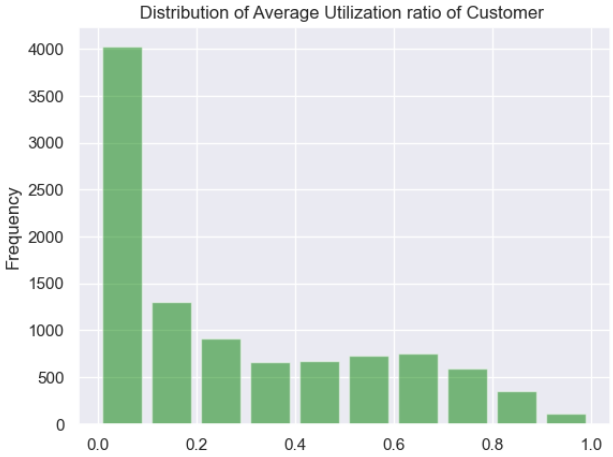
Hình 16 Biểu đồ phân bố thuộc tính Total_Trans_Amt



Hình 17 Biểu đồ phân bố thuộc tính Total_Trans_Ct



Hình 18 Biểu đồ phân bố thuộc tính Total_Ct_Chng_Q4_Q1



Hình 19 Biểu đồ phân bố thuộc tính Avg_Utlization_Ratio

IV. THỰC NGHIỆM

A. *Mô hình thực nghiệm*

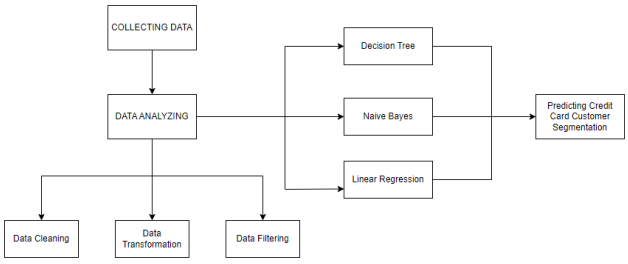
Bước quan trọng nhất trong quá trình thực hiện là thu thập các bộ dữ liệu cần thiết cho công việc nghiên cứu. Phương pháp luận là áp dụng cho tập dữ liệu chứa thông tin của khách hàng. Để giảm khó khăn trong quá trình phân tích, chúng em có thể xác định các thuộc tính duy nhất và loại bỏ các thuộc tính không thể được sử dụng để phân tích. Sau khi thu thập dữ liệu, các dữ liệu được chuyển thành dạng mong muốn.

Đó là bước quan trọng nhất để có được dữ liệu mong muốn cụ thể từ dữ liệu thô. Thêm tỷ lệ độ chính xác của tiền xử lý thô dữ liệu, tỷ lệ chính xác hơn của dữ liệu phù hợp.

Bước tiếp theo sau khi tiền xử lý dữ liệu là tìm dữ liệu không đầy đủ, không liên quan trong tập dữ liệu và loại bỏ nó trong để có được kết quả chính xác của công việc. Loại bỏ giai đoạn dữ liệu không mong muốn được gọi là làm sạch dữ liệu.

Sau đó, chúng em lựa chọn thuật toán như tuyến tính hồi quy, Máy vector hỗ trợ, Tiêu chuẩn Naïve Bayes Phân loại, thuật toán cây quyết định tốt hơn phân loại.

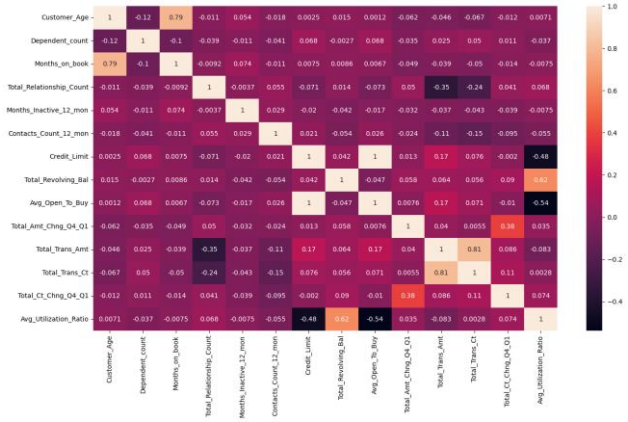
Ở đây, chúng em sử dụng thuật toán Decision Tree, Naïve Bayes và Linear Regission để tiến hành xác định thuộc tính quyết định và phân loại.



Hình 20 Mô hình thực nghiệm

B. *Tiền xử lý dữ liệu*

Ngoài 2 thuộc tính không liên quan mà chúng em đã loại bỏ từ trước, nhóm sẽ dựa vào Ma trận tương quan (Correlation matrix) để xem xét việc loại bỏ (những) thuộc tính nào có mức độ tương quan cao.



Hình 21 Ma trận tương quan

Trong phương pháp 'pearson' để tính toán hệ số tương quan, mức tương quan được đo bằng giá trị nằm trong khoảng từ -1 đến 1:

- Giá trị tuyệt đối gần 1 (0.8 - 1.0) cho thấy một mức tương quan cao, cho thấy sự tương quan mạnh mẽ giữa hai biến (dẫn tới việc loại bỏ thuộc tính).
- Giá trị tuyệt đối gần 0 (0.0 - 0.2) cho thấy một mức tương quan thấp, cho thấy sự tương quan yếu giữa hai biến.
- Giá trị tuyệt đối nằm giữa 0.2 và 0.8 cho thấy một mức tương quan trung bình, không quá mạnh hoặc yếu.

Kết quả từ Ma trận tương quan cho thấy: Không có thuộc tính nào có độ tương quan quá cao để tiến hành loại bỏ. Điều đó có nghĩa, chúng em vẫn giữ 21 thuộc tính trên để phân tích.

Một số thuộc tính định tính chứa giá trị "Unknown" có thể khiến dữ liệu trở nên mơ hồ. Để xử lý điều này, chúng em tiến hành chia bộ dữ liệu thành 2 nhóm (dựa vào 2 giá trị của thuộc tính quyết định), sau đó thay thế của giá trị "Unkhown" thành giá trị xuất hiện nhiều nhất trong mỗi nhóm.

```
# Xác định các thuộc tính chứa giá trị "unknown"
unknown_columns = ['Marital_Status', 'Education_Level', 'Income_Category']

# Xác định thuộc tính quyết định
decision_column = 'Attrition_Flag'

# Lặp qua từng thuộc tính chứa giá trị "unknown"
for column in unknown_columns:
    # Tạo một Dataframe tạm thời chứa giá trị không rỗng và không phải "unknown"
    temp_df = df[df[column] != 'Unknown']

    # Tạo một Series với giá trị xuất hiện nhiều nhất trong thuộc tính, dựa trên thuộc tính quyết định
    most_frequent_value = temp_df.groupby(decision_column)[column].apply(lambda x: x.mode()[0])

    # Tạo một dictionary để ánh xạ giá trị "unknown" thành giá trị xuất hiện nhiều nhất
    replace_dict = dict(zip(most_frequent_value.index, most_frequent_value.values))

    # Thay thế giá trị "unknown" bằng giá trị xuất hiện nhiều nhất
    df[column] = np.where(df[column] == 'Unknown', df[decision_column].map(replace_dict), df[column])
```

Hình 22 Đoạn mã lệnh Python xử lý các thuộc tính chứa giá trị "Unknown"

Tiếp đến, ngoại trừ thuộc tính quyết định là Attrition_Flag, có một số thuộc tính thuộc kiểu dữ liệu định tính – phân loại (Gender, Card_Category, Income_Category, Education_Level, Marital_Status) chưa phù hợp để phân tích trong các mô hình thuật toán đã nêu. Do đó, chúng em tiến hình chuyển những thuộc tính này về dạng one-hot vector để thuận xử lý.

```
one_hot_encoded_data = pd.get_dummies(df, columns = ['Gender','Card_Category','Income_Category','Education_Level','Marital_Status'])
one_hot_encoded_data
```

Hình 23 Đoạn mã lệnh Python chuyển đổi về dạng one-hot vector

C. *Chạy các mô hình và dự đoán*

Đoạn code Python mà chúng em thực thi được công khai và up tại link: https://drive.google.com/drive/folders/1Nzop-r37_0KJArbOpJLg8rzef2GxON1r?usp=sharing

Trong bài toán này, chúng em sử dụng kỹ thuật K-fold Cross-Validation chia tập dữ liệu thành 5 phần. Mỗi phần sẽ lần lượt làm tập test để kiểm thử, các phần còn lại sẽ kết hợp làm tập train để huấn luyện mô hình.

Đầu tiên, nhóm import các thư viện và bộ dữ liệu đã nêu trên để tiến hành thực nghiệm. Một số thông tin và tóm tắt về các thuộc tính như sau:

```
df.info()
[7] ✓ 0.1s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 20 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Attrition_Flag            10127 non-null  object
 1   Customer_Age              10127 non-null  int64
 2   Gender                    10127 non-null  object
 3   Dependent_count           10127 non-null  int64
 4   Education_Level           10127 non-null  object
 5   Marital_Status            10127 non-null  object
 6   Income_Category           10127 non-null  object
 7   Card_Category             10127 non-null  object
 8   Months_on_book            10127 non-null  int64
 9   Total_Relationship_Count  10127 non-null  int64
 10  Months_Inactive_12_mon    10127 non-null  int64
 11  Contacts_Count_12_mon     10127 non-null  int64
 12  Credit_Limit              10127 non-null  float64
 13  Total_Revolving_Bal       10127 non-null  int64
 14  Avg_Open_To_Buy           10127 non-null  float64
 15  Total_Amt_Chng_Q4_Q1      10127 non-null  float64
 16  Total_Trans_Amt           10127 non-null  int64
 17  Total_Trans_Ct            10127 non-null  int64
 18  Total_Ct_Chng_Q4_Q1       10127 non-null  float64
 19  Avg_Utilization_Ratio     10127 non-null  float64
dtypes: float64(5), int64(9), object(6)
memory usage: 1.5+ MB
```

```
df.describe()
```

	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Revolving_Bal
count	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000
mean	46.325960	2.346203	35.928409	3.812580	2.341167	2.455317	8631.953698	1162.814061
std	8.016814	1.298908	7.986416	1.554408	1.010622	1.106225	9088.776650	814.987335
min	26.000000	0.000000	13.000000	1.000000	0.000000	0.000000	1438.300000	0.000000
25%	41.000000	1.000000	31.000000	3.000000	2.000000	2.000000	2555.000000	359.000000
50%	46.000000	2.000000	36.000000	4.000000	2.000000	2.000000	4549.000000	1276.000000
75%	52.000000	3.000000	40.000000	5.000000	3.000000	3.000000	11067.500000	1784.000000
max	73.000000	5.000000	56.000000	6.000000	6.000000	6.000000	34516.000000	2517.000000

Avg_Open_To_Buy	Total_Amt_Chng_Q4_Q1	Total_Trans_Amt	Total_Trans_Ct	Total_Ct_Chng_Q4_Q1	Avg_Utilization_Ratio
10127.000000	10127.000000	10127.000000	10127.000000	10127.000000	10127.000000
7469.139637	0.759941	4404.086304	64.858695	0.712222	0.274894
9090.685324	0.219207	3397.129254	23.472570	0.238086	0.275691
3.000000	0.000000	510.000000	10.000000	0.000000	0.000000
1324.500000	0.631000	2155.500000	45.000000	0.582000	0.023000
3474.000000	0.736000	3899.000000	67.000000	0.702000	0.176000
9859.000000	0.859000	4741.000000	81.000000	0.818000	0.503000
34516.000000	3.397000	18484.000000	139.000000	3.714000	0.999000

Hình 24 Một số thông tin và các chỉ số của các thuộc tính

Xác định biến x và biến y trước khi áp dụng vào từng mô hình:

```
x = one_hot_encoded_data.drop('Attrition_Flag', axis='columns')
y = one_hot_encoded_data['Attrition_Flag']
```

Hình 25 Xác định biến x và biến y

Kỹ thuật K-fold Cross-Validation:

```
# Định nghĩa số lượng nhóm (số lượt Cross-Validation)
num_folds = 5

# Khởi tạo KFold object
kfold = KFold(n_splits=num_folds, random_state=5, shuffle=True)

# Tạo list để lưu các điểm dữ liệu train và test trong mỗi lượt Cross-Validation
train_indices = []
test_indices = []

# Chia dữ liệu thành các nhóm train và test
for train_index, test_index in kfold.split(x):
    train_indices.append(train_index)
    test_indices.append(test_index)

# Lặp qua từng lượt Cross-Validation và train/test mô hình
for fold in range(num_folds):
    print(f"Fold {fold+1}:")

    # Lấy chỉ mục train và test tương ứng với lượt Cross-Validation hiện tại
    train_index = train_indices[fold]
    test_index = test_indices[fold]

    # Lấy dữ liệu train và test từ chỉ mục
    x_train, x_test = x.iloc[train_index], x.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
```

Hình 26 Quá trình thực hiện kỹ thuật K-fold Cross-Validation

Thuật toán Decision Tree:

```
# Train Decision Tree Classifier
clf.fit(x_train, y_train)

# Dự đoán nhãn cho dữ liệu test
y_pred = clf.predict(x_test)

# Đánh giá độ chính xác
tree_score = metrics.accuracy_score(y_test, y_pred)
print("Accuracy:", tree_score)
print("Report:", metrics.classification_report(y_test, y_pred))
```

Hình 27 Tiến hành huấn luyện, dự đoán và tính độ chính xác trên mô hình Decision Tree

```
#Tính ma trận nhầm lẫn
tree_cm = metrics.confusion_matrix(y_test, y_pred)

#Biểu diễn lên đồ thị heatmap
plt.figure(figsize=(12,12))
sns.heatmap(tree_cm, annot=True, fmt=".3f", linewidths=.5, square=True, cmap='Blues_r')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
title = 'Decision Tree Accuracy Score: {0}'.format(tree_score)
plt.title(title, size=15)
```

Hình 28 Tính ma trận nhầm lẫn cho Decision Tree

Thuật toán Naïve Bayes:

```
# Tiếp tục quá trình train và test mô hình tại đây
gnb = GaussianNB()

# Train Decision Tree Classifier
bayes_pred = gnb.fit(x_train, y_train)

# Dự đoán nhãn cho dữ liệu test
y_pred = gnb.predict(x_test)

# Đánh giá độ chính xác
bayes_score = metrics.accuracy_score(y_test, y_pred)
print("Accuracy:", bayes_score)
print("Report:", metrics.classification_report(y_test, y_pred))
```

Hình 29 Tiến hành huấn luyện, dự đoán và tính độ chính xác trên mô hình trên mô hình Naive Bayes

Thuật toán Linear Regression:

```
model = LinearRegression()
model.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Hình 30 Tiến hành chạy và dự đoán trên mô hình Linear

V. KẾT QUẢ VÀ SO SÁNH

A. So sánh Naïve Bayes và Decision Tree

Với 2 mô hình Tree và Bayes, đây là 2 mô hình phân lớp. Do đó, chúng đều được đánh giá từ Ma trận nhầm lẫn (Confusion matrix) và các chỉ số đánh giá hiệu suất liên quan như sau:

- **Accuracy**: Tỷ lệ dự đoán chính xác được thực hiện bởi mô hình về một vấn đề phân loại.
- **Precision**: Tỷ lệ các dự đoán tích cực thực sự trong số tất cả các dự đoán tích cực do mô hình đưa ra về một vấn đề phân loại.
- **Recall**: Tỷ lệ dự đoán tích cực thực sự trong số tất cả các trường hợp tích cực thực tế trong dữ liệu về vấn đề phân loại.
- **F1-Score**: Trung bình điều hòa của độ chính xác và thu hồi, cân bằng cả hai số liệu và thường sử dụng để đánh giá các mô hình phân loại nhị phân.

Fold 1	**Accuracy**	**93.880%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.84	0.80	0.82
	Existing Customer	0.96	0.97	0.96
Fold 2	**Accuracy**	**95.015%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.85	0.84	0.84
	Existing Customer	0.97	0.97	0.97
Fold 3	**Accuracy**	**93.086%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.78	0.82	0.80
	Existing Customer	0.96	0.95	0.96
Fold 4	**Accuracy**	**93.728%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.79	0.83	0.81
	Existing Customer	0.97	0.96	0.96
Fold 5	**Accuracy**	**93.432%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.76	0.80	0.78
	Existing Customer	0.97	0.96	0.96

Bảng 2 Các chỉ số đánh giá của Decision Tree

Fold 1	**Accuracy**	**89.339%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.73	0.62	0.67
	Existing Customer	0.92	0.95	0.94
Fold 2	**Accuracy**	**89.388%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.70	0.58	0.64
	Existing Customer	0.92	0.95	0.94
Fold 3	**Accuracy**	**90.765%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.77	0.66	0.71
	Existing Customer	0.93	0.96	0.95
Fold 4	**Accuracy**	**89.037%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.66	0.61	0.64
	Existing Customer	0.93	0.94	0.94
Fold 5	**Accuracy**	**89.630%**		
	Report	**Precision**	**Recall**	**F1-score**
	Attrited Customer	0.65	0.61	0.63
	Existing Customer	0.93	0.95	0.94

Bảng 3 Các chỉ số đánh giá của Naive Bayes

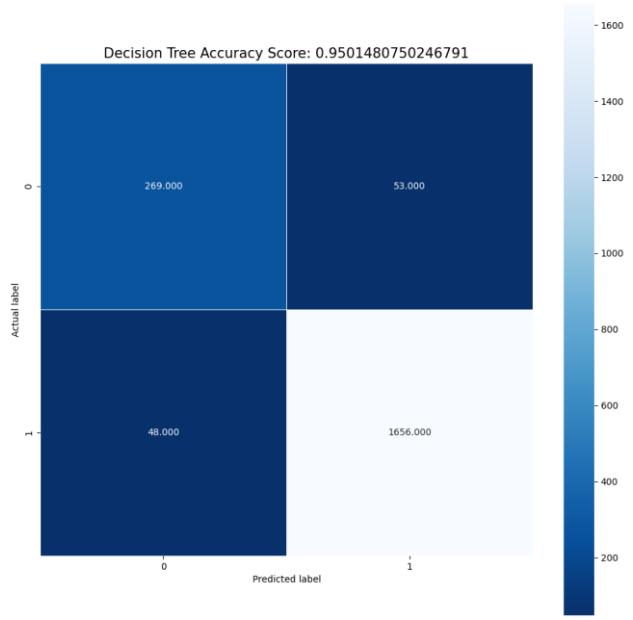
Để đánh giá độ tin cậy chung của mô hình, sử dụng kết hợp cả hai chỉ số precision và recall thành một chỉ số duy nhất F-score, được tính theo công thức:

$$f - measure = \frac{2 * recall * precision}{recall + precision}$$

Trong bài toán này, các chỉ số precision và recall của mô hình đều cao nên chỉ số f-score cũng cao. Qua đó, cho thấy hiệu năng của việc sử dụng mô hình Naïve Bayes và Decision Tree trong bài toán phân loại này là đáng tin cậy.

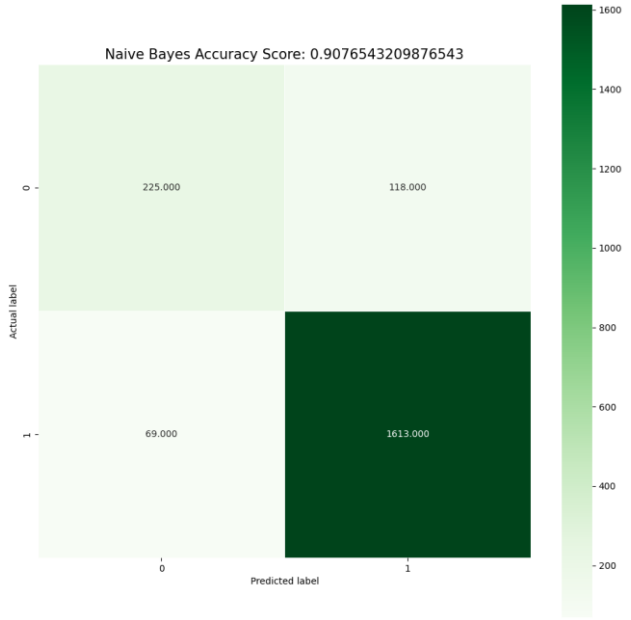
Mô hình đạt độ chính xác cao nhất của Decision Tree khi train, test ở fold 2 với 95.015%. Trong khi đó, fold 3 của Naïve Bayes với 90.765% (fold 3) là mô hình có độ chính xác cao nhất. Từ đây, ta có thể thấy, Decision Tree dự đoán kết quả có phần tốt hơn so với dự đoán áp dụng từ Naïve Bayes.

Kết quả ma trận nhầm lẫn của 2 mô hình có độ chính xác cao nhất (mô hình fold 2 của Decision Tree và mô hình fold 3 của Naïve Bayes):



Hình 31 Ma trận nhầm lẫn mô hình Decision Tree (fold 2)

- TP (True Positive): Số lượng dự đoán chính xác. Là khi mô hình dự đoán đúng khách hàng hủy đăng ký sử dụng thẻ tín dụng (269 khách hàng).
- TN (True Negative): Số lượng dự đoán chính xác một cách gián tiếp. Là khi mô hình dự đoán đúng khách hàng không hủy thẻ tín dụng, tức là việc không chọn trường hợp rời bỏ dịch vụ là chính xác (1656 khách hàng).
- FP (False Positive - Type 1 Error): Số lượng các dự đoán sai lệch. Là khi mô hình dự đoán khách hàng sẽ rời bỏ dịch vụ thẻ tín dụng nhưng người đó vẫn còn tiếp tục (53 khách hàng).
- FN (False Negative - Type 2 Error): Số lượng các dự đoán sai lệch một cách gián tiếp. Là khi mô hình dự đoán khách hàng vẫn tiếp tục dùng dịch vụ thẻ tín dụng nhưng người đó đã hủy đăng ký dịch vụ này, tức là việc không chọn trường hợp không hủy dịch vụ thẻ tín dụng là sai (48 khách hàng).



Hình 32 Ma trận nhầm lẫn mô hình Naïve Bayes (fold 3)

- TP (True Positive): Số lượng dự đoán chính xác là 225 khách hàng.
- TN (True Negative): Số lượng dự đoán chính xác một cách gián tiếp là 1613 khách hàng.
- FP (False Positive): Số lượng các dự đoán sai lệch là 118 khách hàng.
- FN (False Negative): Số lượng các dự đoán sai lệch một cách gián tiếp là 69 khách hàng.

B. Đánh giá Linear Regression

Khác với Tree và Bayess, Linear Regression là một mô hình hồi quy. Do đó, Linear được đánh giá bằng các chỉ số metrics như sau:

- Mean Squared Error (MSE): Độ chênh lệch bình phương trung bình giữa giá trị dự đoán và giá trị thực tế của biến mục tiêu liên tục.
- Root Mean Squared Error (RMSE): Căn bậc hai của MSE và được sử dụng để giải thích chỉ số lỗi trong cùng một đơn vị với biến mục tiêu.
- Mean Absolute Error (MAE): Độ chênh lệch tuyệt đối trung bình giữa giá trị dự đoán và giá trị thực tế của một biến mục tiêu liên tục.
- R-squared (R2): Tỷ lệ phương sai trong biến mục tiêu có thể được giải thích bằng mô hình.

MSE	RMSE	MAE	R2
0.12513324607705423	0.4041718592912714	0.2502664921541084	0.023092464240389243

Bảng 4 Các chỉ số đánh giá hiệu suất Linear Regression

Dựa trên các chỉ số trên, ta có thể kết luận rằng mô hình Linear Regression không phù hợp với dữ liệu và không dự đoán tốt giá trị thực tế. Thực chất, như mục tiêu ban đầu, chúng em đã xác định bài toán thuộc loại phân lớp. Linear là một mô hình hồi quy dự đoán các giá trị số theo thời gian. Điều chúng em muốn thấy là "Sẽ thế nào nếu áp dụng mô hình hồi quy lên một bài toán phân loại" và đây là kết quả mà nhóm đã nhận được. Từ đó khẳng định rõ ràng về việc lựa chọn thuật toán cho bộ dữ liệu là điều rất quan trọng trong quá trình Data Mining.

VI. KẾT LUẬN

Trong đồ án này, chúng em đã tiến hành một nghiên cứu thực hiện việc phân loại khách hàng thẻ tín dụng sử dụng ba thuật toán phân loại chính: Decision Tree, Naive Bayes và Linear Regression. Qua đó, chúng em đã xác định được hiệu suất và ứng dụng của từng thuật toán trong lĩnh vực Data Mining.

Kết quả của đồ án cho thấy Decision Tree và Naive Bayes đạt được độ chính xác cao với tỷ lệ trên 93% và 88% tương ứng. Trong khi đó, việc áp dụng Linear Regression cho bài toán phân loại không đạt được kết quả mong muốn. Điều này làm chúng em nhận thấy rằng việc lựa chọn mô hình phù hợp với bài toán và tập dữ liệu là vô cùng quan trọng.

Kết quả này không chỉ đáng chú ý từ góc độ thực nghiệm mà còn cung cấp những thông tin hữu ích cho sinh viên mới bắt đầu học về Data Mining. Đồ án này đã giúp chúng em hiểu rõ hơn về quá trình lựa chọn và đánh giá hiệu suất của các thuật toán phân loại.

Trong tương lai, chúng em định hướng tiếp tục nghiên cứu và áp dụng các thuật toán khác trong lĩnh vực Data Mining. Chúng em sẽ tìm hiểu về các thuật toán gom cụm (clustering) và hồi quy (regression) để phát triển sự hiểu biết của mình về hiệu suất và ứng dụng của chúng trong việc xử lý dữ liệu và dự đoán trên các tập dữ liệu khác nhau.

Đồ án này là bước đầu trong hành trình học tập và nghiên cứu của chúng em và tạo đà cho các nghiên cứu tiếp theo. Qua đó, chúng tôi mong muốn đóng góp vào sự phát triển và ứng dụng của Data Mining trong thực tế.

NHÌN NHẬN – ACKNOWLEDGMENT

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn đến Trường Đại học Công Nghệ Thông Tin – Đại học Quốc gia Thành phố Hồ Chí Minh và Khoa Hệ thống Thông tin đã tạo điều kiện cho nhóm có cơ hội học tập và nghiên cứu với môn học này, luôn tạo điều kiện tốt nhất để sinh viên có thể hoàn thành tốt quá trình học tại trường nói chung và trong môn học này nói riêng.

Tiếp theo, nhóm em xin gửi lời cảm ơn chân thành tới thầy Hà Lê Hoài Trung, giảng viên trực tiếp phụ trách giảng dạy lớp môn Khai thác dữ liệu. Thầy đã tận tình hướng dẫn, chỉ bảo với những phân tích định hướng rõ ràng cho nhóm trong suốt quá trình thực

hiện đề tài, là tiền đề để nhóm có thể hoàn thành đề tài đúng hạn. Thầy cũng tạo điều kiện thuận lợi nhất có thể với các tài liệu cần thiết liên quan, giải đáp thắc mắc tại lớp khi các nhóm gặp khó khăn.

Và cuối cùng, xin cảm ơn tất cả các bạn trong nhóm đã cùng nhau chia sẻ công việc, hoàn thành tốt trách nhiệm của cá nhân trong suốt quá trình thực hiện với sự hướng dẫn của thầy và phân công của nhóm trưởng. Đó là những yếu tố quan trọng nhất để hoàn thành tốt mục tiêu đã đặt ra.

Nhóm vẫn rất mong nhận sự góp ý từ phía Thầy nhằm giúp nhóm rút ra những kinh nghiệm quý báu, hoàn thiện vốn kiến thức đã học tập để là hành trang cho nhóm có thể tiếp tục hoàn thành những đồ án, nghiên cứu tiếp theo trong tương lai.

Xin chân thành cảm ơn Thầy!

TÀI LIỆU THAM KHẢO

[1] Shumin Xie, Siying Zeng, Lu Liu, Huimin Wei, Yanhua Xu, Xiaoxu Lu (2022) “Predicting Geospatial Thinking Ability for Secondary School Students

Based on the Decision Tree Algorithm in Mainland China”.

[2] Mitchell, M. T. (1997).Machine learning. New York: The McGraw-Hill Companies.

[3] Quinlan, J. R. (1986). Induction of decision trees.Machine Learning,1(1), 81-106.

[4] Quinlan. J. R. (2019, April). C5.0: An informal tutorial. Retrieved from https://www.rulequest.com/see5-unix.html

[5] Che, D., Liu, Q., Rasheed, K., & Tao, X. (2011). Decision tree and ensemble learning algorithms with their applications in bioinformatics. In H. Arabnia & Q. N. Tran (Eds.),Software tools and algorithms for biological systems(pp. 191-199). New York, NY: Springer.doi:10.1007/978-1-4419-7046-6_19

[6] Han, J., Fang, M.,Ye, S., Chen, C., Wan, Q., & Qian, X. (2019). Using decision tree to predict response rates of consumer satisfaction, attitude, and loyalty surveys.Sustainability,11(8), 2306.doi:10.3390/su11082306

[7] Indika Wickramasinghe, Harsha Kalutarage (2020), “Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation”

[8]. Singh, S., & Kumar, V. (2013).Performance Analysis of Engineering Students for Recruitment Using Classification Data Mining Techniques. International Journal of Computer Science Engineering and Technology (IJCSET), 3(2), 31–37.

[9]. UmadeviD.Sundar, Dr.P.Alli,”An Optimized Approach to Predict the Stock Market Behavior and Investment Decision Making using Benchmark Algorithms for Naive Investors”, Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on (IEEE Xplore Digital Library), pg1 -5.,2013

[10]. Gholap, J. (2012). Performance Tuning Of J48 Algorithm For Prediction Of Soil Fertility. Asian Journal of Computer Science and Information Technology, 2(8).

[11] B. Sravani, M. M. Bala (2020), “Prediction of Student Performance Using Linear Regression”. 2020 International Conference for Emerging Technology (INCET)