Mục lục

1. Sơ lược về Machine Learning 2

1.1. Khái niệm 2

1.2. Ứng dụng 2

1.3. Quy trình của Machine Learning 3

1.4. Các kiểu học máy 4

1.5. Deep learning 5

2. Ứng dụng bài toán 5

2.1. Mô tả bài toán 5

2.2. Xử lý dữ liệu 6

2.3. Huấn luyện và đánh giá mô hình 13

2.4. Kết luận 18

3. Tổng kết 19

4. Tài liệu tham khảo 19

1. Sơ lược về Machine Learning

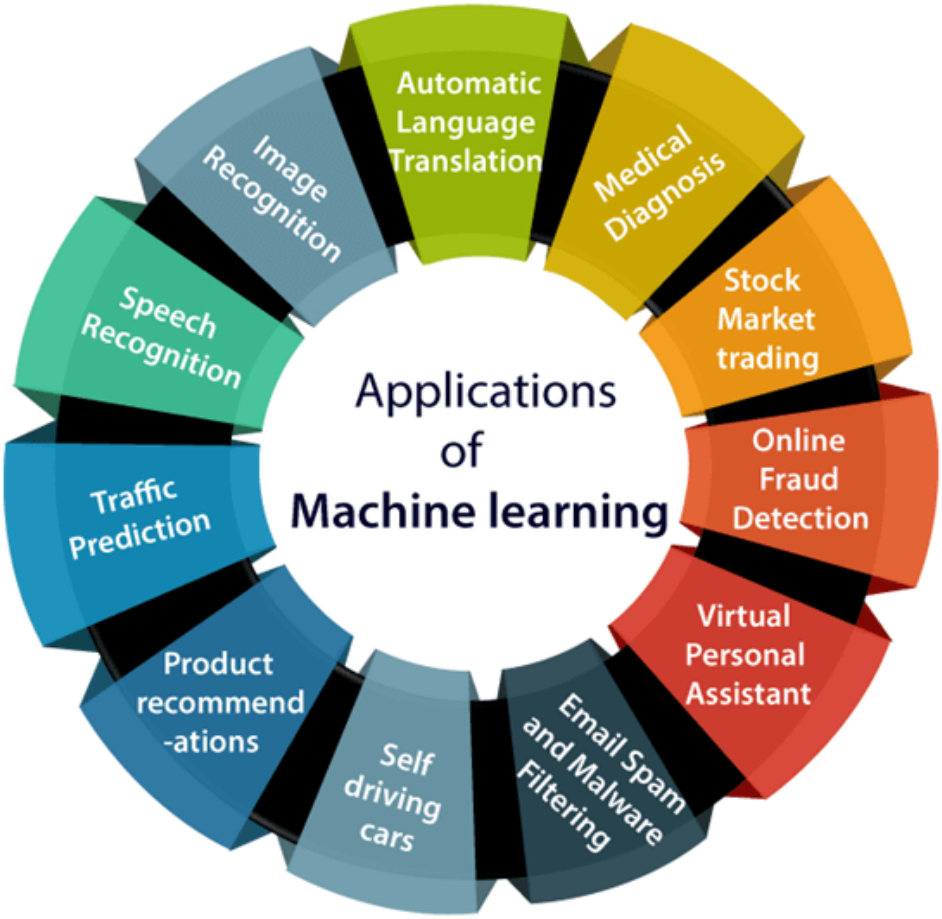
1.1. Khái niệm

Machine Learning (ML) là một lĩnh vực của trí tuệ nhân tạo, nghiên cứu và phát triển các thuật toán để các hệ thống có thể sử dụng dữ liệu trong quá khứ, bắt chước hành vi "học" của con người và giải quyết các bài toán cụ thể.

Theo wiki, ML là thuật ngữ chung, nói về cách máy tính tự giải quyết các vấn đề thông qua các thuật toán tạo ra bởi con người mà không cần hướng dẫn chi tiết.

1.2. Ứng dụng

Machine Learning đang được ứng dụng trong lĩnh vực thị giác máy tính, nhận diện giọng nói, lọc thư điện tử, hệ thống gợi ý, tránh gian lận, dự đoán,…



Hình 1.1: Ứng dụng của Machine Learning

1.3. Quy trình của Machine Learning

Hình 1.2: Các bước thực hiện bài toàn Machine Learning

1.3.1. Thu thập dữ liệu

Dữ liệu tốt cần phải liên quan với bài toán, ít mất mát và ít trùng lặp.

Dữ liệu càng lớn, xây dựng mô hình càng hiệu quả.

Dữ liệu có thể thu thập từ nhiều nguồn và tích hợp với nhau.

1.3.2. Chuẩn bị dữ liệu

Quá trình chuẩn bị dữ liệu bao gồm:

- Trực quan hóa dữ liệu
- Làm sạch dữ liệu với các giá trị thiếu sót, trùng lặp và các giá trị ngoại lại, bất thường
- Chuẩn hóa các giá trị số và số hóa giá trị phân loại
- Lựa chọn thuộc tính
- Tách dữ liệu thành các bộ huấn luyện và kiểm thử

Việc chuẩn bị cần hiểu rõ yêu cầu bài toán, phân tích kĩ dữ liệu, trực quan hóa dữ liệu.

1.3.3. Xây dựng mô hình

Lựa chọn thuật toán phù hợp mục đích bài toán để huấn luyện mô hình.

Tinh chỉnh các tham số trong thuật toán để cải thiện mô hình

1.3.4. Đánh giá mô hình

Tính toán, đánh giá kết quả, độ chính xác của mô hình, tính quan trọng của thuộc tính trong mô hình, chi phí vận hành để từ đó quyết định xây dựng lại và cải thiện mô hình với các bước trên hay triển khai mô hình

1.3.5. Triển khai mô hình

Đưa mô hình vào thực tế, đánh giá lại mô hình liên tục để xây dựng và cải thiện mô hình.

1.4. Các kiểu học máy

Machine learning có thể được phân loại dựa trên mục đích của các thuật toán:

1.4.1. Học có giám sát

Mô hình học có giám sát được huấn luyện bởi bộ dữ liệu được đánh nhãn và sử dụng để dự đoán nhãn của những dữ liệu chưa biết. Dựa vào nhãn là các giá trị cụ thể tính toán hay các lớp, mô hình học có giám sát chia các thuật toán ra làm hai kiểu là mô hình hồi quy và phân lớp.

Các thuật toán chính trong mô hình hồi quy là thuật toán hồi quy tuyến tính (Linear regression), và hồi quy đa thức (Polynominal regression), với mục đích là xây dựng hàm số mô tả tốt nhất quan hệ của biến độc lập với giá trị nhãn, trong đó nhãn là biến có miền giá trị liên tục.

Mô hình phân lớp sử dụng các thuật toán với hai cách tiếp cận chính là:

- Mô hình hồi quy: xây dựng một siêu phẳng chia vùng các mẫu dữ liệu, với các thuật toán đặc trưng là Logistic regression, Support Vector Machine, K-nearest-neighbor,…
- Mô hình cây, xây dựng một sơ đồ cây có thể dự đoán lớp của mẫu dựa trên các mẫu đã biết. với các thuật toán đặc trưng là Decision tree, Random Forest.

Ngoài ra, mô hình phân lớp cũng có các thuật toán kết hợp như Gradient Boosting, hoặc các thuật toán sử dụng mạng nơ ron như Neural Network hay Perceptron, là cơ sở cho mô hình học sâu (Deep learning).

1.4.2. Học không giám sát

Mô hình học không giám sát được huấn luyện bởi những dữ liệu chưa được gắn nhãn, với mục đích để máy có thể tự tìm ra đặc trưng tương đồng giữa các dữ liệu hoặc cấu trúc lại dữ liệu.

Các thuật toán trong mô hình học không giám sát dựa vào các mục đích khác nhau để phân ra hai loại:

- Phân cụm (Clustering): mô hình tìm ra các mẫu tương đồng nhau và chia vào các lớp, sử dụng các thuật toán như K-means-clustering, Mean-shift-clustering,…

- Giảm chiều dữ liệu (Demensionality reduction): mô hình giảm độ phức tạp của dữ liệu bằng cách đưa các thuộc tính vào không gian mới với ít chiều hơn, giữ được ý nghĩa của dữ liệu và có thể tái tạo, nhằm giúp việc tính toán mô hình không bị overftting hay quá phức tạp. Một số thuật toán được sử dụng là Singular value decompostion, principal component analysis, Autoencoders,…

1.4.3. Học bán giám sát và học cải thiện

Mô hình học bán giám sát được huấn luyện trên bộ dữ liệu chứa một phần dữ liệu nhỏ gán nhãn và còn lại không gán nhãn. Mô hình tính toán bằng cách giả định những mẫu không gán nhãn có nhãn bằng nhiều cách và thuật toán như giả định liên tục (Continuity Assumption), giả định cụm (Cluster Assumption), giả định đa điểm (Manifold Asumption).

Mô hình học cải thiện sử dụng các phương pháp kết hợp thuật toán dể cải thiện mô hình.

1.5. Deep learning

Deep Learning là một bộ phận, cải tiến của Machine Learning. Deep Learning là phương thức học máy phức tạp hơn với mạng nơ ron (Neural Network), mô phỏng giống với cách bộ não con người tư duy và kết luận mà không cần đến nhiều sự can thiệp của con người trong quá trình học như Machine Learning.

Deep learning được ứng dụng trong thị giác máy tính, phân tích giọng nói, sinh văn bản, hệ thống lái xe tự động,...

2. Ứng dụng bài toán

2.1. Mô tả bài toán

Một quản lý ngân hàng đang đối mặt với việc nhiều khách hàng rời bỏ dịch vụ thẻ tín dụng, mong muốn có một mô hình có thể dự đoán những khách hàng có ý định rời bỏ dịch vụ để đưa ra chương trình và dịch vụ phù hợp nhằm thay đổi quyết định của nhóm khách hàng này.

Bài toán được cung cấp dữ liệu cá nhân của 10000 khách hàng về thông tin giới tính, tuổi, mức lương, hạng thẻ, hạn mức, số tiền giao dịch,… và tình trạng sử dụng thẻ tín dụng, tổng cộng 18 thuộc tính.

Đây là bài toán phân lớp với nhãn là tình trạng sử dụng thẻ tín dụng của khách hàng với hai lớp là *Đang sử dụng (Existing Customer)* và *Đã ngừng sử dụng (Attrition Customer)*. Tuy nhiên, dữ liệu chỉ có 16.1% khách hàng ngừng sử dụng thẻ nên sẽ gây khó khăn cho việc huấn luyện mô hình.

2.2. Xử lý dữ liệu

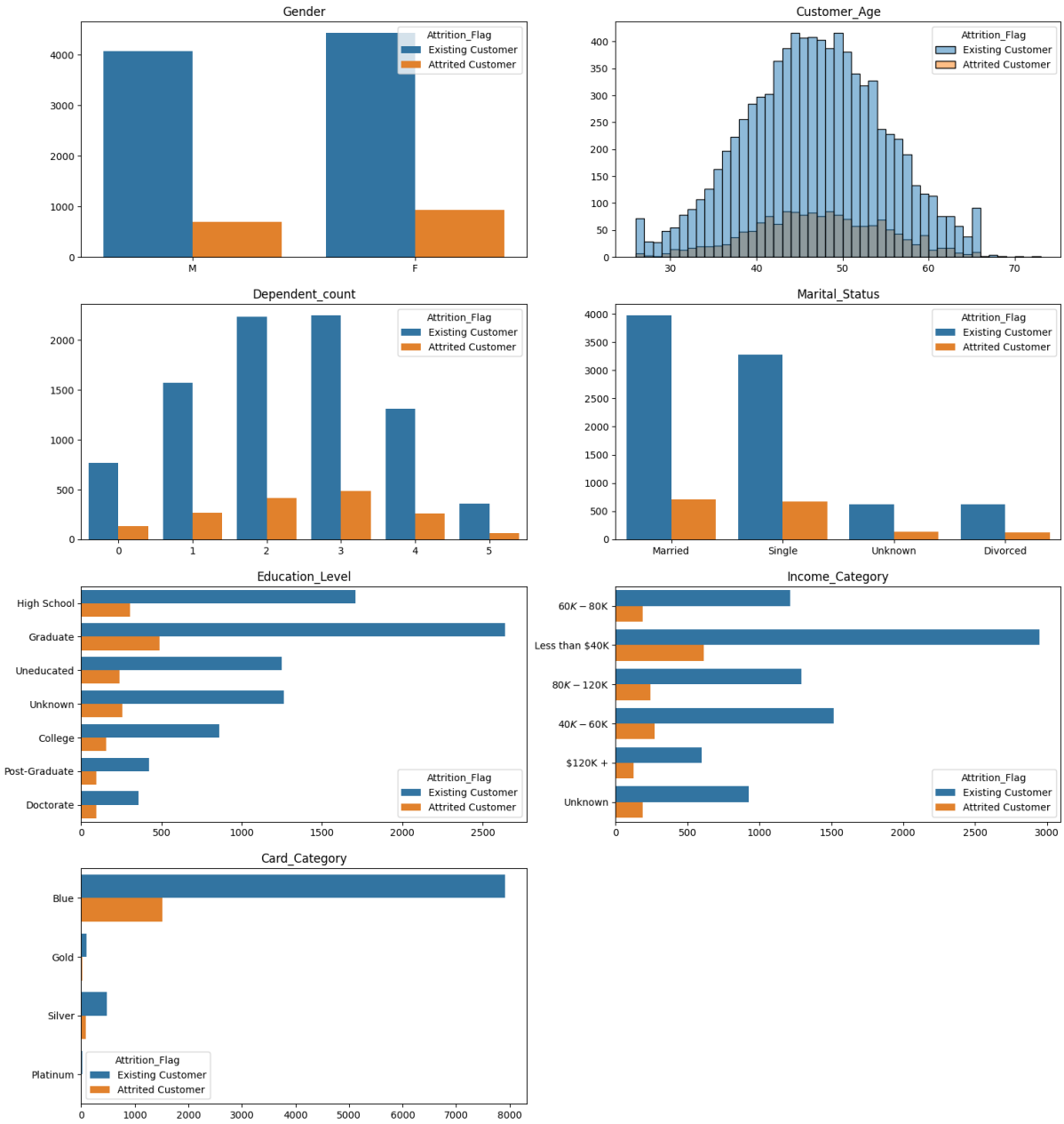
2.2.1. Trực quan hóa dữ liệu

Trước khi thực hiện trực quan, ta khái quát các thuộc tính được đề cập đến trong bài toán:

Thuộc tính	**Ý nghĩa**
Customer age	Tuổi của khách hàng
Gender	Giới tính của khách hàng
Dependent_count	Số người phụ thuộc với khách hàng (con cái, cha mẹ, v.v.)
Education_Level	Trình độ học vấn của khách hàng
Marital_Status	Tình trạng hôn nhân của khách hàng
Income_Category	Mức thu nhập của khách hàng
Card_Category	Loại thẻ mà khách hàng sử dụng
Months_on_book	Thời gian khách hàng sử dụng ngân hàng
Total_Relationship_Count	Số sản phẩm khách hàng sử dụng
Months_Inactive_12_mon	Thời gian không sử dụng của khách hàng
Contacts_Count_12_mon	Số lần khách hàng được ngân hàng liên lạc trong 12 tháng gần nhất
Credit_Limit	Hạn mức trên thẻ tín dụng của khách hàng
Total_Revolving_Bal	Tổng nợ của khách hàng
Avg_Open_To_Buy	Khả năng vay của khách hàng
Total_Trans_Amt	Tổng tiền giao dịch trong 12 tháng
Total_Trans_Ct	Tổng số lần giao dịch trong 12 tháng
Total_Amt_Chng_Q4_Q1	Thay đổi về số tiền giao dịch của Q4 so với Q1
Total_Ct_Chng_Q4_Q1	Thay đổi về số lần giao dịch của Q4 so với Q1
Avg_Utilization_Ratio	Tỷ lệ sử dụng thẻ trung bình

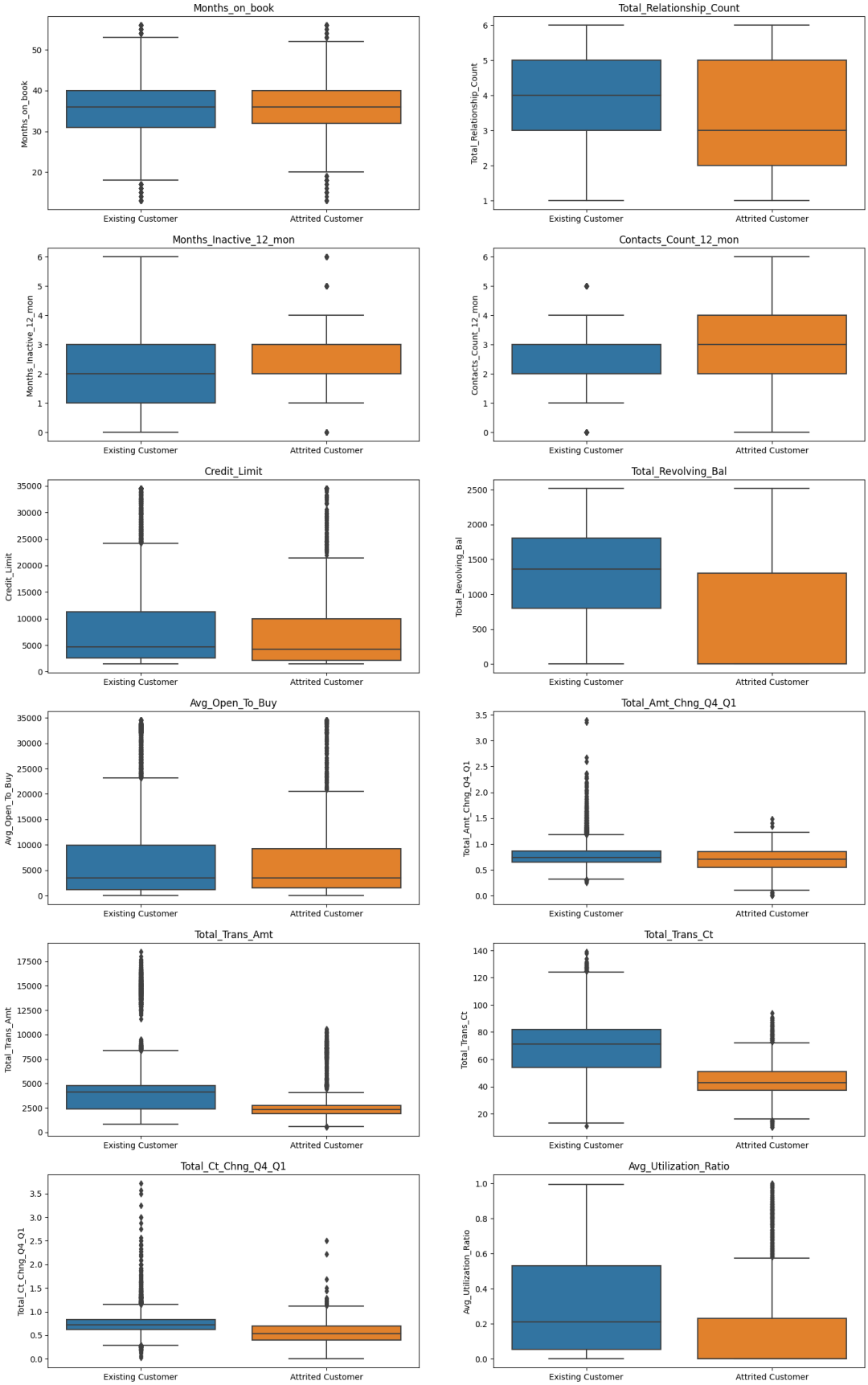
Bảng 2.1: Ý nghĩa của các thuộc tính

Phân bố của dữ liệu mỗi lớp dựa trên từng thuộc tính được trực quan hóa như sau:



Hình 2.1: Phân bố của các thuộc tính phân lớp

Các thuộc tính phân lớp được trực quan bằng tần suất quan sát được trong mỗi lớp phân loại với biểu đồ cột, với một cột thể hiện các giá trị độc nhất của thuộc tính và một cột thể hiện số lần xuất hiện của chúng, chia ra trong hai lớp Existing Customer và Attrited Customer. Từ trực quan, thuộc tính Card_Category thể hiện sự phân bố bất thường với đa số dữ liệu nằm trong giá trị lớp Blue.



Hình 2.2: Phân bố của các thuộc tính ước lượng

Các thuộc tính định lượng được biểu diễn sự phân bố và phân tán của dữ liệu với biểu đồ hộp (box plot). Trực quan của dữ liệu cho thấy các khoảng phân bố, các điểm ngoại lai và bất thường trong dữ liệu, từ đó xem xét các phương pháp loại bỏ dữ liệu ngoại lai.

2.2.2. Chuẩn hóa dữ liệu

Nhằm đảm bảo cho thuật toán tính toán hiệu quả và chính xác, các dữ liệu đầu vào cần được chuẩn hóa dữ liệu.

Đối với những dữ liệu định lượng, phương pháp chuẩn hóa là đặt giá trị lớn nhất và nhỏ nhất là các cận của khoảng và đưa những giá trị trong dữ liệu về cùng một khoảng. Dựa trên công thức sau, dữ liệu được tính toán thông qua MinMaxScaler của thư viện scikit-learn.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Các giá trị ngoài khoảng dữ liệu huấn luyện khi triển khai thực tế vẫn sẽ được tính toán với công thức này, nghĩa là khi mô hình huấn luyện với khoảng [0, 5], chuẩn hóa thành [0, 1], những dữ liệu [5, 10] sẽ nằm trong khoảng [1, 2].

Các giá trị của thuộc tính thể hiện tuổi khách hàng (Customer_Age) không được chuẩn hóa theo dữ liệu định lượng, mà được nhóm theo từng độ tuổi, thể hiện trong thuộc tính Customer_Age_Group, và được chuẩn hóa theo thuộc tính phân lớp.

Đối với những thuộc tính phân lớp, mô hình lựa chọn hai phương pháp tiếp cận là đánh số và one-hot, đặt các giá trị của các lớp là các số nguyên.

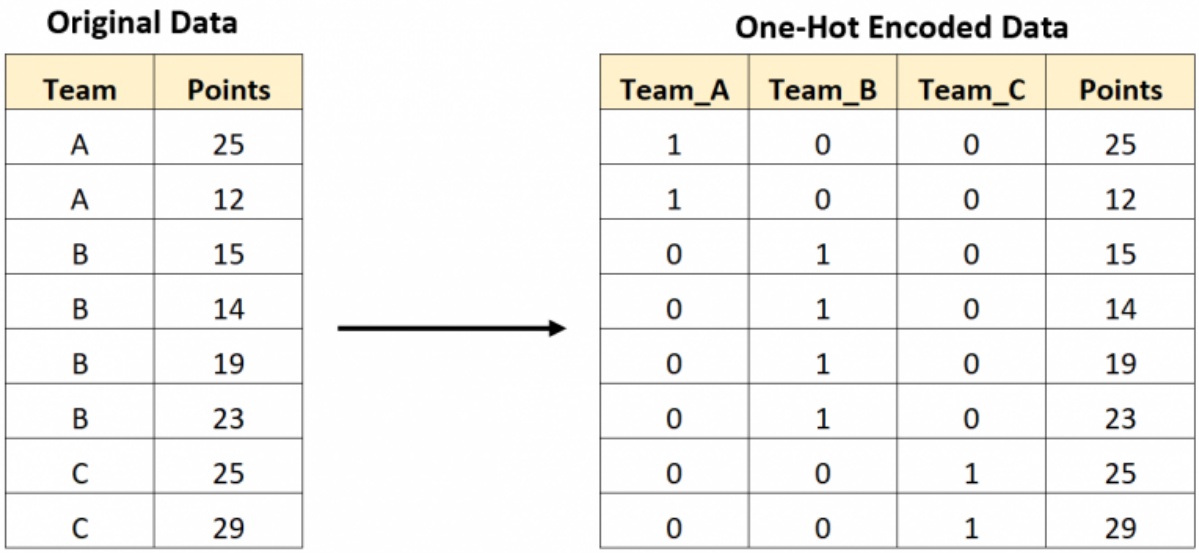
Original Data

Team	Points
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

Label Encoded Data

Team	Points
0	25
0	12
1	15
1	14
1	19
1	23
2	25
2	29

Hình 2.4: Dữ liệu phân lớp được đánh số theo nhãn



Hình 2.5: Dữ liệu phân lớp được mã hóa One-hot

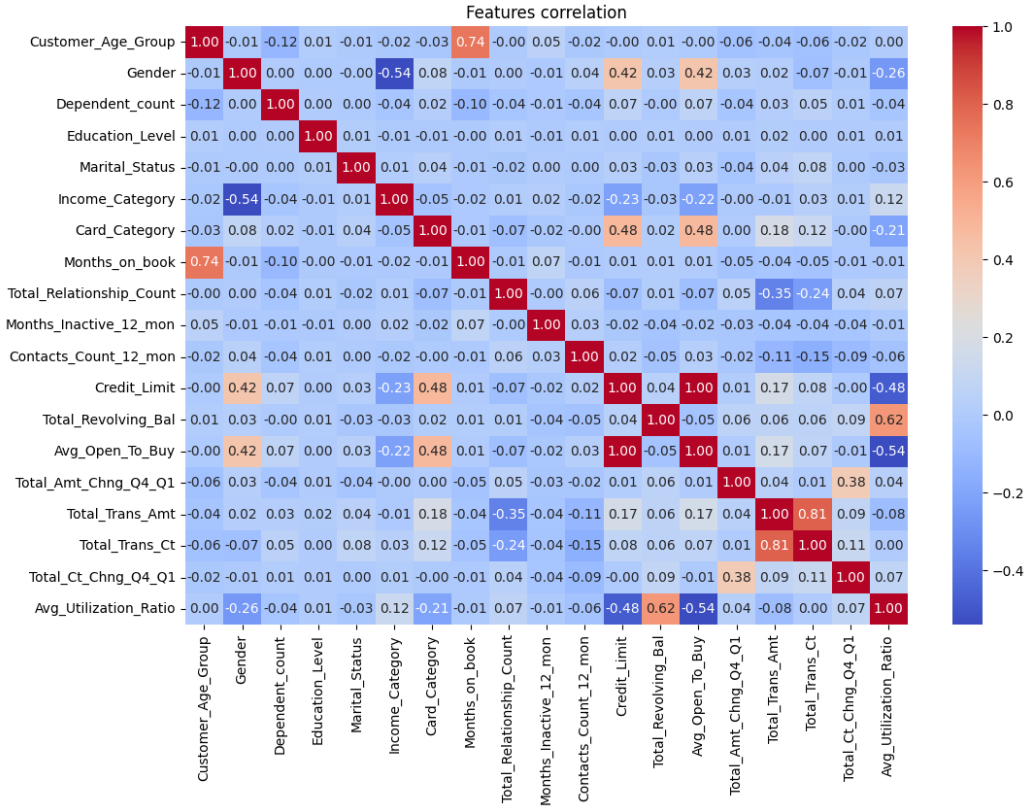
Các giá trị phân lớp được đưa về giá trị nguyên từ 0 đến (n-1) (n là số lớp của thuộc tính trong dữ liệu) thông qua LabelEncoder, hoặc đưa về vector chứa 0 và 1 thể hiện sự xuất hiện của lớp của mỗi mẫu dữ liệu, biểu diễn bằng các cột là lớp của thuộc tính trong dữ liệu thông qua OneHotEncoder, đều là công cụ trong thư viện scikit-learn.

Khi triển khai thực tế, những dữ liệu nằm trong lớp chưa xác định so với dữ liệu được huấn luyện, được đưa về lớp tương đồng với lớp có sẵn, hoặc sử dụng lớp "Unknown", đã xác định trong mô hình. Tuy nhiên, việc thực hiện lại và cập nhật bước chuẩn hóa cũng được xem xét nếu dữ liệu huấn luyện không còn phù hợp với thực tế.

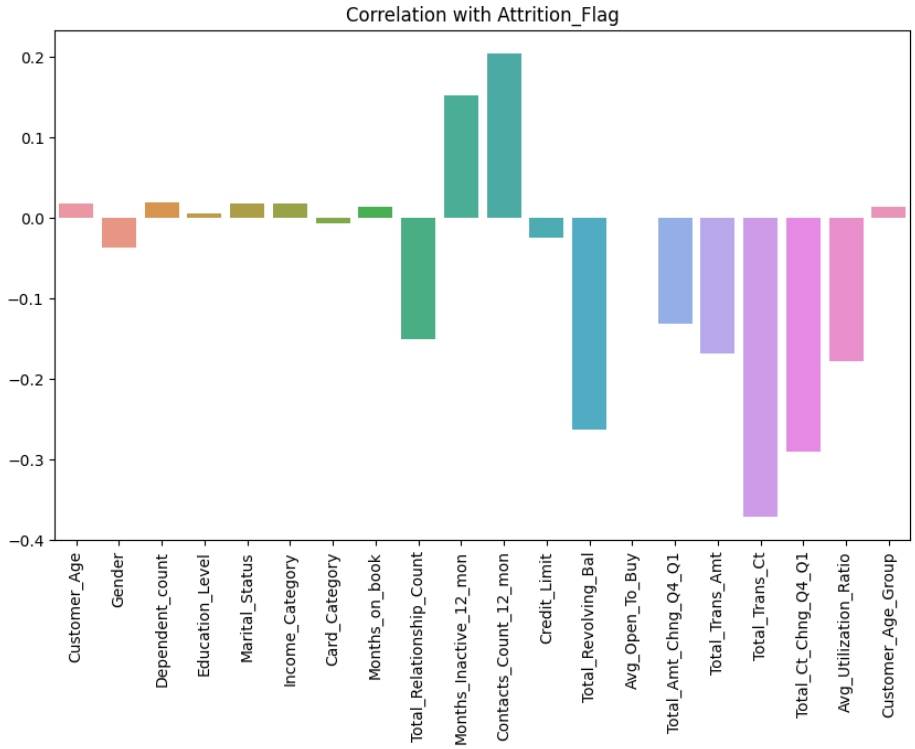
2.2.3. Lựa chọn thuộc tính

Các thuộc tính được chọn lọc nhằm đảm bảo các thuật toán thực hiện hiệu quả. Mô hình xét đến tính tương quan của thuộc tính với nhau, đảm bảo không có tương quan quá cao giữa các thuộc tính.

Tính tương quan (correlation) giữa các biến thuộc tính và với biến nhãn được trực quan hóa với heatmap và biểu đồ cột như sau:



Hình 2.6: Tương quan giữa các thuộc tính



Hình 2.7: Tương quan của các thuộc tính với nhãn bài toán

Dựa trên ý nghĩa và tính tương quan được nêu trên, thuộc tính Credit_Limit và Avg_Utilization_Ration được tính toán bằng tổng và thương của hai thuộc tính độc lập Total_Revolving_Bal và Avg_Open_To_Buy, nên việc loại bỏ hai thuộc tính này cần được

xem xét khi huấn luyện mô hình. Thuộc tính Card_Category với tính tương quan thấp với lớp nhãn và phân bố tập trung vào một lớp nên cũng được xem xét loại bỏ.

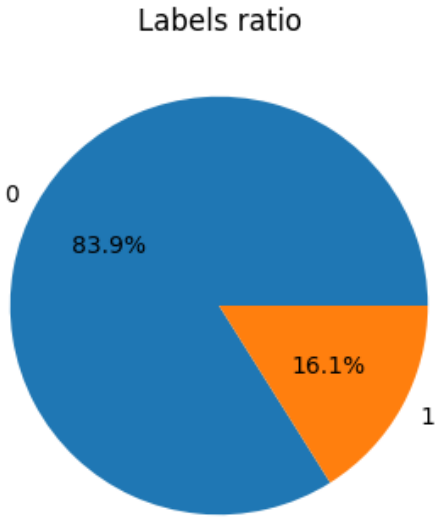
Biểu diễn tương quan của thuộc tính với nhãn cũng cho thấy các thuộc tính độc lập và phi tuyến tính với nhãn của dữ liệu, việc lựa chọn các thuật toán dựa trên mô hình tree-based (mô hình dựa trên cây quyết định, giải thích rõ hơn ở bước huấn luyện mô hình) được ưu tiên khi huấn luyện mô hình.

2.2.4. Phân tách bộ dữ liệu huấn luyện và kiểm thử

Dữ liệu được tách thành bộ dữ liệu dùng để huấn luyện mô hình và một bộ nhằm đánh giá độ chính xác của mô hình.

2.2.4.1. Cân bằng dữ liệu

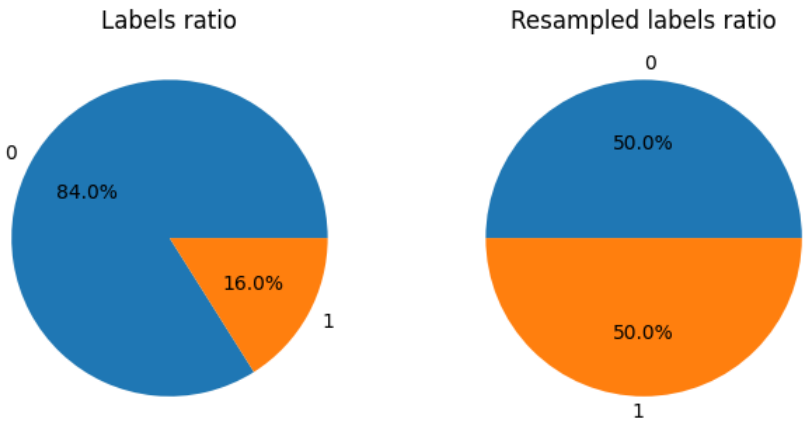
Như đã đề cập trong mô tả bài toán, dữ liệu được cung cấp có sự thiếu cân bằng giữa hai lớp nhãn, mô tả bằng biểu bồ tròn sau:



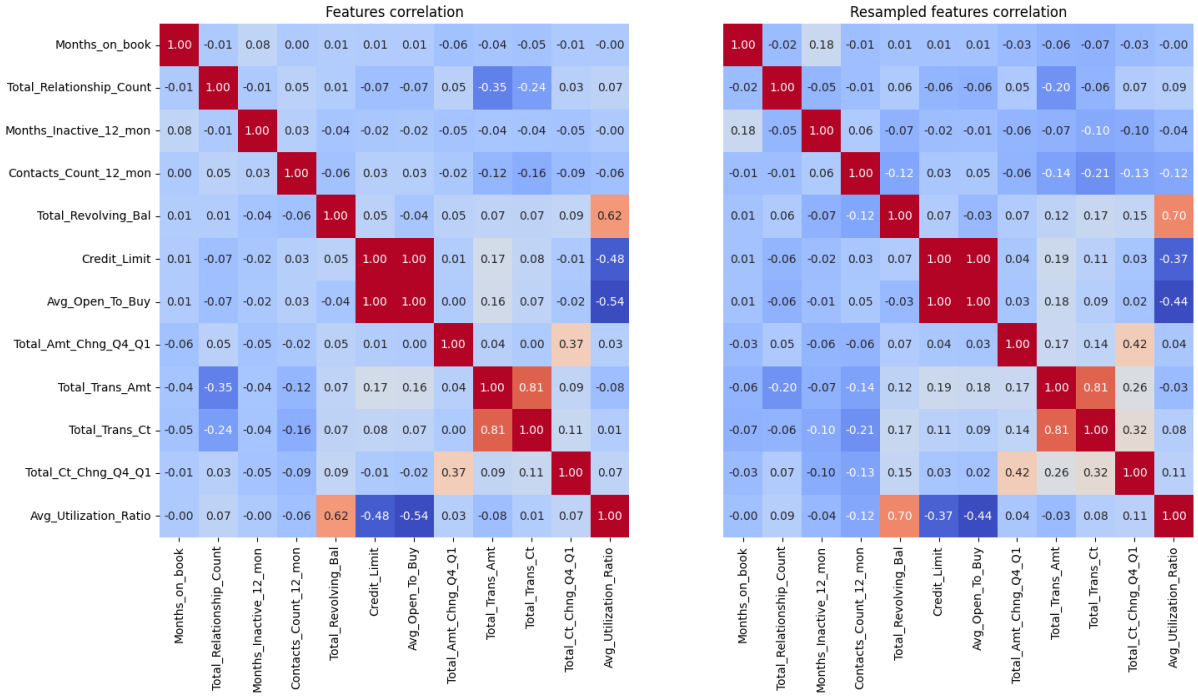
Hình 2.8: Phân bố theo lớp của nhãn bài toán
0: Khách hàng vẫn sử dụng, 1: Khách hàng ngừng sử dụng

Nhằm cho việc huấn luyện mô hình hiệu quả, bộ dữ liệu huấn luyện có thể xem xét đến sử dụng các phương pháp resample, giúp cân bằng lại các lớp nhãn. Trong mô hình này, phương pháp được lựa chọn là oversample với SMOTE.

SMOTE (Synthetic Minority Over-sampling Technique) tổng hợp các thuộc tính của mẫu nhân tạo từ các thuộc tính của mẫu thiểu số trong dữ liệu với các mẫu khác gần nhất, dựa trên thuật toán k-nearest (thuật toán tìm những thực thể có khoảng cách ngắn nhất tới nhau), đảm bảo tính tự nhiên và đa dạng của dữ liệu.



Hình 2.9: Phân bố dữ liệu trước và sau khi resample
0: Khách hàng vẫn sử dụng, 1: Khách hàng ngừng sử dụng



Hình 2.10: Tương quan của thuộc tính dữ liệu trước và sau khi resample

Thực hiện SMOTE lên dữ liệu huấn luyện bên cạnh cân bằng các lớp, cũng gây ra thay đổi nhỏ trong tương quan của các thuộc tính, có thể thấy thông qua bản đồ nhiệt thể hiện tương quan của hai bộ dữ liệu, vậy nên việc ứng dụng vào mô hình cũng cần được xem xét để mô hình được chính xác.

Trong thực tế khi triển khai mô hình, những dữ liệu sinh ra từ SMOTE có thể mẫu thuẫn với những giá trị mới do nhiều nguyên nhân. Để đảm bảo mô hình có thể tiếp tục triển khai, việc kiểm tra thường xuyên là cần thiết.

2.3. Huấn luyện và đánh giá mô hình

Đối với bài toán phân lớp phi tuyến tính, các thuật toán tree-based được ưu tiên ứng dụng như Random Forest, XGB (Extreme Gradient Boosting), tuy nhiên các thuật toán khác như Logistic

Regression, Support Vector Machine, K-Nearest và Naive Bayes cũng được sử dụng để đánh giá mô hình một cách khách quan.

2.3.1. Mô hình Random forest và XGB

Hai thuật toán được sử dụng đều là thuật toán kết hợp dựa trên mô hình cây là Decision tree, với hai cách tiệp cận khác nhau.

Về thuật toán Decision tree, đây là thuật toán xây dựng cây dự đoán bằng cách chọn lọc thuộc tính phù hợp nhất ở gốc và tiếp tục các thuộc tính phù hợp sau đó ở các cây con. Lựa chọn điều kiện dừng của cây có thể ảnh hưởng tới tính hiệu quả và chính xác của mô hình, ở đây có thể là ngưỡng giá trị mất mát, độ sâu, số lượng lá,… Các tham số khác như lambda, alpha trong công thức chính quy cũng được tinh chỉnh để giới hạn sự phát triển của cây.

2.3.1.1. Random forest

Random forest cải thiện mô hình bằng cách xây dựng nhiều Decison tree trên các tập dữ liệu con ngẫu nhiên từ dữ liệu gốc và lấy kết quả đa số. Bằng cách lựa chọn các mẫu dữ liệu ngẫu nhiên và bộ đặc trung con ngẫu nhiên, với số lương cây đủ nhiều, mô hình Random forest không chỉ tạo ra mô hình dự đoán tự nhiên mà còn xác định được những thuộc tính có ảnh hưởng quan trọng tới mô hình nhất, giúp ứng dụng nhiều trong các bài toán thực tế.

Mô hình có thể được tinh chỉnh các tham số tương tự như tham số của Decision tree, hay các số lượng cây, phương thức lấy mẫu để cải thiện hiệu suất mô hình.

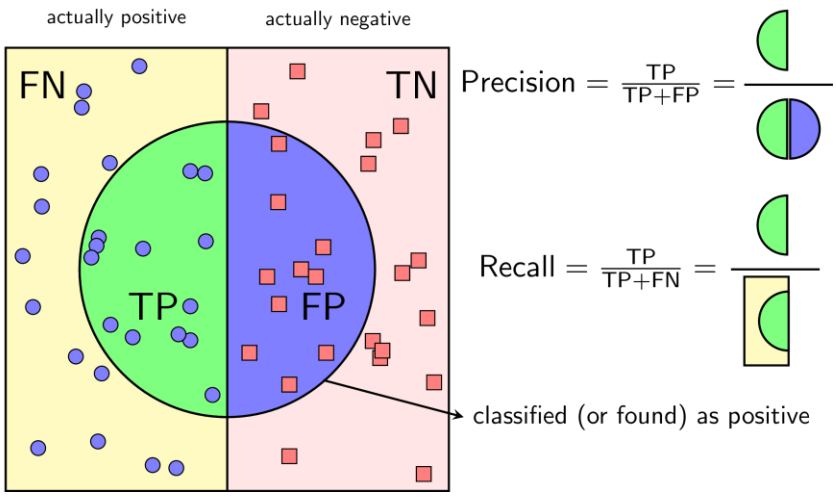
2.3.1.2. Extreme Gradient Boosting (XGB)

Khác với Decision tree xây dựng các cây ngẫu nhiện độc lập với nhau, XGB tiếp cận mô hình cây bằng cách tự động cải thiện một cây dần dần, hay được coi là tỉa cây nhằm xây dựng một cây chính xác và hiệu quả nhất. Cũng như Random forest, mô hình XGB qua nhiều cây được xây dựng, cũng xác định được những thuộc tính có ảnh hưởng quan trọng nhất tới cây.

Mô hình cũng được tinh chỉnh các tham số của Decision tree và tham số learning-rate để cải thiện hiệu suất mô hình.

2.3.1.3. Kết quả mô hình

Mô hình được đánh giá thông qua các giá trị sau:



Hình 2.11: Công thức tính toán Precison và Recall

- Precision: được tính bằng tỉ lệ những điểm gán nhãn đúng so với những điểm được gán nhãn đó, khi giá trị càng cao thì độ chính xác của mô hình càng lớn.
- Recall: được tính bằng tỉ lệ những điểm gán nhãn đúng so với những điểm thực sự có nhãn đó, khi giá trị càng cao thì tỉ lệ bỏ sót càng thấp.
- F1: được tính bằng nghịch đảo của tổng nghịch đảo giá trị Precision và Recall, khi cả hai giá trị Precision và Recall càng cùng cao, F1 cũng sẽ càng cao và mô hình sẽ càng tối ưu.
- Accuracy: được tính bằng tỉ lệ đoán đúng trong các dự đoán, mô hình càng tối ưu khi giá trị càng cao.

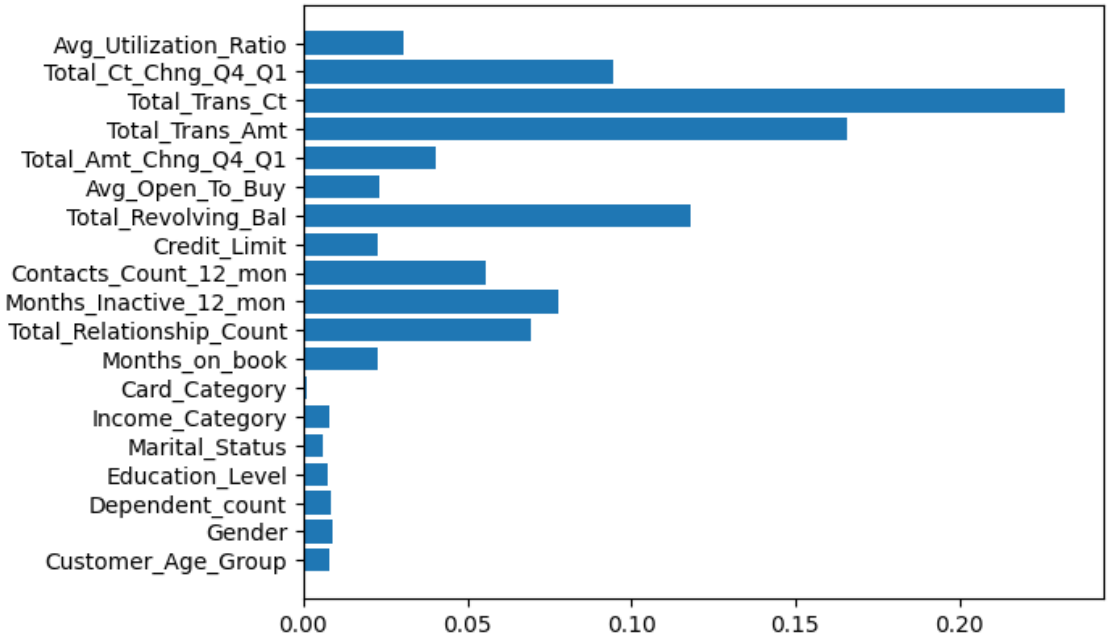
Kết quả được đánh giá trong các trường hợp dữ liệu gốc với dữ liệu resample và lược bỏ thuộc tính như sau:

	Random forest				**XGB**			
	Prec.	Recall	F1	Acc.	Prec.	Recall	F1	Acc.
Original data with full features	0.92	0.78	0.78	0.95	0.93	0.85	0.89	0.97
Resampled data with full features	0.85	0.87	0.86	0.95	0.85	0.89	0.87	0.96
Original data with high correlation features removed	0.91	0.79	0.85	0.95	0.93	0.84	0.88	0.96

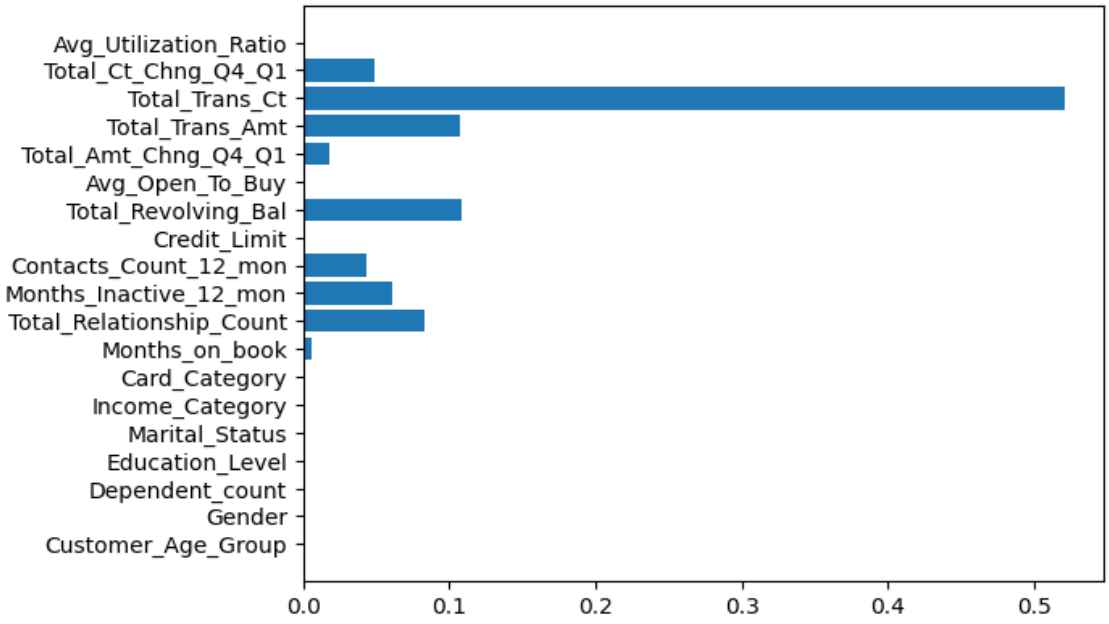
Resampled data with high correlation features removed	0.85	0.87	0.86	0.95	0.85	0.89	0.87	0.96
Original data with high correlation and low importance features removed	0.93	0.83	0.88	0.96	0.93	0.84	0.88	0.96
Resampled data with high correlation and low importance features removed	0.86	0.87	0.86	0.96	0.85	0.89	0.87	0.96

Bảng 2.2: Kết quả của mô hình RF và XGB

Mô hình Random Forest và XGB tính toán khả năng ảnh hưởng của thuộc tính lên kết quả phân lớp của bài toán, được biểu diễn dưới bảng sau:



Hình 2.11: Khả năng ảnh hưởng của thuộc tính, tính toán bởi Random Forest.



Hình 2.12: Khả năng ảnh hưởng của thuộc tính, tính toán bởi XGradient Boosting.

Việc lựa chọn thuộc tính và sinh mẫu không ảnh hưởng nhiều đến kết quả của mô hình XGB do khả năng học chọn lọc các thuộc tính không cần thiết trong mô hình từ dữ liệu thô. Trong khi đó, mô hình Random Forest được cải thiện tính chính xác khi được cung cấp dữ liệu cân bằng và độc lập hơn.

Khi sử dụng dữ liệu được cân bằng với SMOTE, giá trị Precision của cả hai mô hình giảm trong khi giá trị Recall lại tăng. Điều này xảy ra do việc cân bằng tăng số lượng giá trị đích hay giá trị Positive, dẫn đến khả năng tìm kiếm điểm Positive rộng hơn, khiến việc đoán nhầm tăng và bỏ sót giảm.

Cả hai mô hình đều đánh giá ảnh hưởng của từng thuộc tính lên mô hình tương đồng nhau. Những thuộc tính liên quan đến hành vi giao dịch và tương tác với ngân hàng, như số tiền giao dịch hay số lần liên lạc giữa ngân hàng với khách hàng, có ảnh hưởng tới kết quả bài toán nhiều hơn những thuộc tính liên quan đến cá nhân khách hàng như nhóm tuổi hay thu nhập.

2.3.2. Các mô hình khác

Ngoài hai thuật toán trên, mô hình tiếp cận thêm một số thuật toán khác, trong đó có các thuật toán phân lớp tuyến tính, như SVC hay NBC, được sử dụng khi các mẫu dữ liệu được phân tách tuyến tính theo lớp, có thể xây dựng đường thẳng hay mặt phẳng phân chia các lớp:

- Support Vector Machine: một thuật toán giám sát xây dựng một siêu phẳng tốt nhất phân chia các lớp dữ liệu trong không gian bằng cách tính toán margin, khoảng cách của điểm gần nhất tới mặt phẳng. Không chỉ xây dựng một siêu phẳng tuyến tính, SVM có thể xử lý các ngoại lệ, cũng như thêm, biến đổi thuộc tính nhằm thuận lợi xây dựng siêu phẳng phù hợp với mô hình.

- Naive Bayes Classifier: thuật toán coi các thuộc tính là biến ngẫu nhiên độc lập và tính xác suất của mỗi lớp. Thuật toán này được sử dụng nhiều trong phân lớp văn bản, do xử lý tốt xác suất hay tần suất của các biến độc lập. Trong mô hình này, phân phối Gaussian được sử dụng, phù hợp với các biến dữ liệu liên tục.

- K-nearest-neighbors: thuật toán tính toán khoảng cách với các dữ liệu xung quanh và phân loại dựa trên lớp của đa số k mẫu dữ liệu gần nó nhất. Đây là một thuật toán đơn giản nhưng tốn nhiều thời gian và dữ liệu để tính toán khi có nhiều thuộc tính hay chiều dữ liệu.

Các mô hình trên cho kết quả F1 và Accuracy như sau:

	SVM		NBC		KNN	
	F1	Acc.	F1	Acc.	F1	Acc.
Original data with full features	0.47	0.88	0.61	0.87	0.08	0.82
Resampled data with full features	0.63	0.85	0.52	0.79	0.36	0.70
Original data with high correlation and low importance features removed	0.46	0.88	0.62	0.88	0.04	0.85
Resampled data with high correlation and low importance features removed	0.64	0.86	0.54	0.79	0.37	0.62

Bảng 2.3: Kết quả một số mô hình

Các mô hình với các tinh chỉnh cơ bản nhưng vẫn cho ra kết quả yếu hơn hai mô hình Random Forest và XGB. Độ chính xác của những mô hình này chưa đến 90% và đoán sai cũng như xót nhiều hơn mô hình RF và XGB.

Những thuật toán này cần tinh chỉnh và chuẩn bị dữ liệu vào phức tạp hơn, cũng như mẫu dữ liệu mà bài toán cung cấp không phù hợp với cách tiếp cận của thuật toán. Thuật toán K-nearest-neighbors đã không đủ mẫu với dữ liệu gốc để phân lớp hiệu quả. Các biến của bài toán cũng không thể coi là độc lập để áp dụng Naive Bayes và tính toán chính xác xác suất.

2.4. Kết luận

Thông qua việc phân tích dữ liệu và quá trình tinh chỉnh, mô hình đã xác định rằng các thuộc tính liên quan đến giao dịch về số tiền và số lần giao dịch, cũng như sự thay đổi của chúng từ quý 1 đến quý 4, có ảnh hưởng lớn nhất đến khả năng dự đoán hành vi ngừng sử dụng thẻ tín dụng.

Mô hình XGB đã cho thấy khả năng xử lý tốt với dữ liệu phức tạp và độc lập tuyến tính của các thuộc tính, giúp nắm bắt mối quan hệ phức tạp giữa chúng. XGB đã hiển thị hiệu suất dự

đoán tốt hơn so với Random Forest và các thuật toán khác được thử nghiệm, như Naive Bayes và Support Vector Machine. Bài toán hướng đến tìm ra được nhiều nhất có thể khách hàng có khả năng ngừng sử dụng thẻ, vậy nên mô hình chạy trên dữ liệu được cân bằng với giá trị Recall tính được cao hơn được lựa chọn, dữ liệu với các thuộc tính được chọn lọc tuy không khác biệt về kết quả với dữ liệu thô, nhưng có thể tính toán nhanh hơn nên cũng được lựa chọn.

Kết quả của nghiên cứu cung cấp một cơ sở cho việc đưa ra quyết định chiến lược trong việc quản lý khách hàng và giảm nguy cơ hành vi ngừng sử dụng thẻ tín dụng. Các thuộc tính số tiền và số lần giao dịch và sự thay đổi của chúng từ quý 1 đến quý 4 có thể được sử dụng để tạo ra các chiến lược cụ thể nhằm tối ưu hóa trải nghiệm của khách hàng và duy trì sự tương tác tích cực với dịch vụ thẻ tín dụng.

3. Tổng kết

Báo cáo trên cung cấp một tổng quan về Machine Learning và quy trình của một mô hình Machine Learning, được áp dụng vào bài toán dự đoán hành vi bỏ thẻ tín dụng của khách hàng tại một ngân hàng.

Đầu tiên, báo cáo đề cập đến các bước chuẩn bị dữ liệu, bao gồm việc chuẩn bị và tiền xử lý dữ liệu, cân bằng dữ liệu và tách dữ liệu huấn luyện và kiểm tra. Sau đó, báo cáo trình bày về lựa chọn thuộc tính, để cải thiện hiệu suất của mô hình.

Báo cáo cũng trình bày về hai thuật toán được sử dụng để đánh giá mô hình, bao gồm Random Forest và Extreme Gradient Boosting (XGB). Các kết quả và hiệu suất của mô hình được trình bày bằng bảng và biểu đồ.

4. Tài liệu tham khảo