**TRƯỜNG ĐẠI HỌC THUỶ LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



Nhóm 20:

HỌ TÊN:

|  |  |
| --- | --- |
| Nguyễn Đình Cường | 2151264648 |
| Trần Vĩ Cầm | 2151260825 |
| Nguyễn Văn Luyện | 2151264669 |
| Trang Công Thành | 2151260840 |

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN TIỀN XỬ LÝ DỮ LIỆU**

**ĐỀ TÀI:ĐÁNH GIÁ ĐỘ RỦI RO RỜI BỎ CỦA NGƯỜI DÙNG**

**NGƯỜI HƯỚNG DẪN: TẠ QUANG CHIỂU**

HÀ NỘI, năm 2024

Mục lục:

[Danh mục bảng biểu : 3](#_Toc5249)

[Phần 1: Giới thiệu bài toán 5](#_Toc19141)

[I, Thông tin dữ liệu 5](#_Toc13751)

[Phần 2: Tiền xử lí dữ liệu 7](#_Toc13975)

1, Các bước tiền xử lí dữ liệu.....................................................................7

1.1, Missing data.................................................................. ....................7

1.2, Mã hoá dữ liệu....................................................................................8

1.3, Loại bỏ cột theo tương quan cao dựa vào threashold..........................8

1.4, Xử lí ngoại lệ.......................................................................................9

1.5, Chuẩn hoá dữ liệu:..............................................................................9

2, Ứng dụng mô hình................................................................................10

[1.1. Chia dữ liệu.......................................................................................11](#_Toc28164)

[1.2. Huấn luyện mô hình..........................................................................12](#_Toc23554)

[1.3. Kiểm tra độ chính xác của mô hình...................................................12](#_Toc19942)

[Phần 3: TỔNG KẾT.................................................................................13](#_Toc6803)

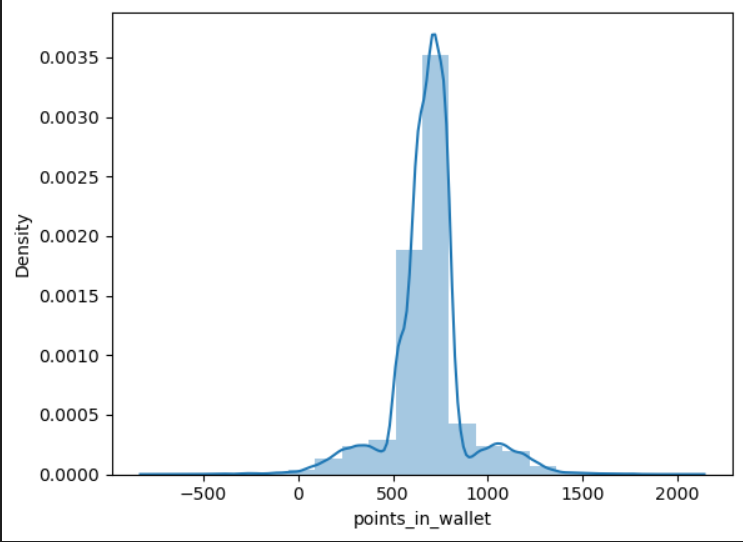
[3.1, Tiền xử lý dữ liệu:.............................................................................13](#_Toc2991)

[3.2, Chạy mô hình học máy......................................................................13](#_Toc14435)

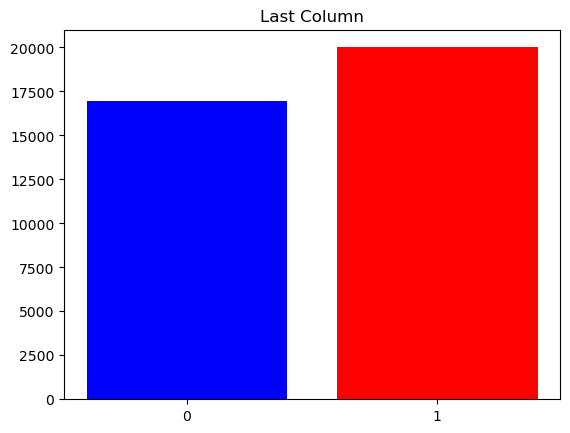
[3](#_Toc14435)[.3, Nhận xét và kết luận..........................................................................13](#_Toc25055)

3.4, Nguồn tài liệu tham khảo..................................................................13

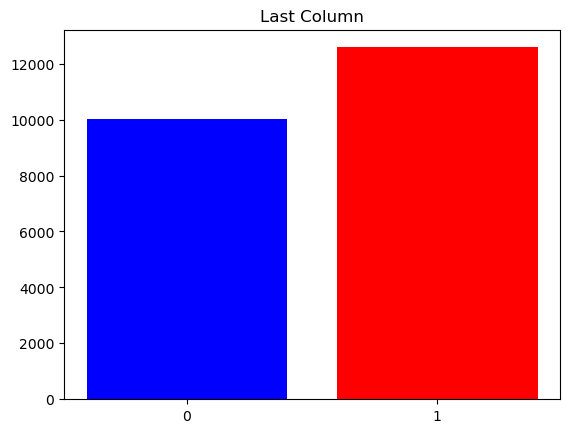
**Danh mục bảng biểu :**



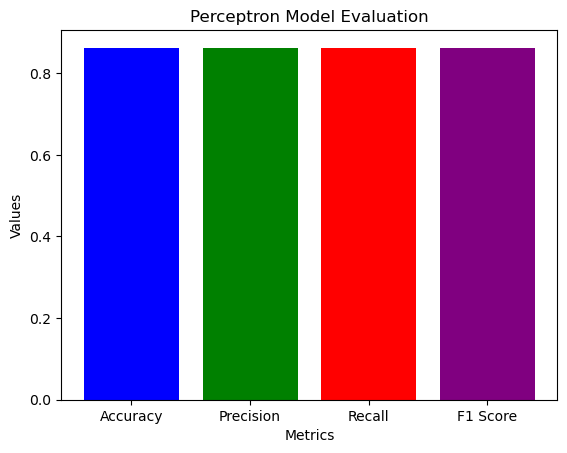
Biểu đồ thể hiện giá trị của cột “points\_in\_wallet”



Biểu đồ số lượng 0/1 của cột churn\_rick\_score



Biểu đồ cột cuối cùng sau khi xử lí điểm ngoại lệ



Biểu đồ thể hiện đánh giá mô hình tỉ lệ đoán đúng theo Perceptron thông qua Accuracy, precisiom, recall, F1 score

**Phần 1: Giới thiệu bài toán**

**I, Thông tin dữ liệu**

Trong lĩnh vực khoa học dữ liệu, tiền xử lý dữ liệu là một bước không thể thiếu và thường chiếm một phần lớn thời gian trong toàn bộ quy trình phân tích dữ liệu. Tiền xử lý dữ liệu, một quá trình quan trọng giúp làm sạch, chuẩn bị và biến đổi dữ liệu thô thành một định dạng phù hợp và sẵn sàng cho việc phân tích. Tiền xử lý dữ liệu không chỉ giúp tăng độ chính xác và hiệu quả của các mô hình phân tích sau này mà còn đảm bảo rằng kết quả phân tích là đáng tin cậy và có thể được diễn giải một cách hợp lý

Tập dữ liệu được sử dụng trong dự án này chứa thông tin về Đánh giá độ rủi ro rời bỏ của người dùng

1. age (tuổi): Độ tuổi của người dùng.

2. gender (giới tính): Giới tính của người dùng.

3. security\_no (số bảo mật): Số bảo mật cá nhân của người dùng.

4. region\_category (danh mục khu vực): Danh mục phân loại khu vực địa lý mà người dùng thuộc về.

5. membership\_category (danh mục thành viên): Phân loại hạng thành viên của người dùng.

6. joining\_date (ngày tham gia): Ngày mà người dùng gia nhập hệ thống.

7. joined\_through\_referral (tham gia qua giới thiệu): Thông tin về việc người dùng có tham gia thông qua việc được giới thiệu từ người khác không.

8. referral\_id (ID giới thiệu): Mã định danh của người giới thiệu (nếu có).

9. preferred\_offer\_types (loại ưu đãi ưa thích): Loại ưu đãi mà người dùng ưa thích.

10. medium\_of\_operation (phương tiện hoạt động): Phương tiện mà người dùng sử dụng để tương tác với hệ thống.

11. internet\_option (lựa chọn internet): Lựa chọn kết nối internet của người dùng.

12. last\_visit\_time (thời gian thăm cuối cùng): Thời gian mà người dùng thăm cuối cùng.

13. days\_since\_last\_login (số ngày kể từ lần đăng nhập cuối cùng): Số ngày kể từ lần đăng nhập cuối cùng của người dùng.

14. avg\_time\_spent (thời gian trung bình dành): Thời gian trung bình mà người dùng dành cho hoạt động trên hệ thống.

15. avg\_transaction\_value (giá trị giao dịch trung bình): Giá trị trung bình của các giao dịch mà người dùng thực hiện.

16. avg\_frequency\_login\_days (số ngày đăng nhập trung bình): Số ngày trung bình giữa các lần đăng nhập của người dùng.

17. points\_in\_wallet (điểm trong ví): Số điểm tích luỹ trong ví của người dùng.

18. used\_special\_discount (sử dụng giảm giá đặc biệt): Thông tin về việc người dùng đã sử dụng giảm giá đặc biệt hay không.

19. offer\_application\_preference (ưu tiên ứng dụng ưu đãi): Sự ưu tiên của người dùng đối với việc ứng dụng các ưu đãi.

20. past\_complaint (khiếu nại trong quá khứ): Thông tin về việc người dùng đã có khiếu nại trong quá khứ hay không.

21. complaint\_status (trạng thái khiếu nại): Trạng thái của các khiếu nại được người dùng đã gửi.

22. feedback (phản hồi): Phản hồi của người dùng về dịch vụ hoặc trải nghiệm của họ.

23. churn\_risk\_score (điểm rủi ro churn): Điểm đánh giá rủi ro rời bỏ của người dùng.

**Phần 2: Tiền xử lí dữ liệu**

**1, Các bước tiền xử lí dữ liệu**

**1.1, Missing data**

Chuyển file dữ liệu về file.csv

df.isnull().sum()

#in ra và tính tổng số lg thiếu trong mỗi cột

Input:

age 0

gender 0

security\_no 0

region\_category 5428

membership\_category 0

joining\_date 0

joined\_through\_referral 0

referral\_id 0

preferred\_offer\_types 288

medium\_of\_operation 0

internet\_option 0

last\_visit\_time 0

days\_since\_last\_login 0

avg\_time\_spent 0

avg\_transaction\_value 0

avg\_frequency\_login\_days 0

points\_in\_wallet 3443

used\_special\_discount 0

offer\_application\_preference 0

past\_complaint 0

complaint\_status 0

feedback 0

churn\_risk\_score 0

dtype: int64

=> Xử lí các cột : region\_category ; preferred\_offer\_types , points\_in\_wallet

Ngoài ra còn xử lí các cột chứa giá trị “?, XXXXXXXX”

Ở cột points\_in\_wallet : xử lí bằng cách thay thế bằng giá trị trung bình (mean), còn các cột khác thì xử lí bằng cách random

Ở cột oined\_through\_referral: thay thế gíatrị ? Bằng “No” tương tự các cột khác

df['points\_in\_wallet']=df['points\_in\_wallet'].fillna(df['points\_in\_wallet'].mean())

df['joined\_through\_referral'] = df['joined\_through\_referral'].replace({'?': 'no', 'yes': 'yes', 'no': 'No'})

print(df['joined\_through\_referral'].head())

df['joined\_through\_referral'] = df['joined\_through\_referral'].replace({'?': 'no', 'yes': 'yes', 'no': 'No'})

print(df['joined\_through\_referral'].head())

df['medium\_of\_operation'] = df['medium\_of\_operation'].replace({'?': 'Both', 'Desktop': 'Desktop', 'Smartphone': 'Smartphone'})

print(df['medium\_of\_operation'].head())

df['referral\_id'].replace('xxxxxxxx', 'CID12313', inplace=True)

print(df['referral\_id'].head())

df['region\_category'].fillna('Town', inplace=True)

df.head(5)

unique\_values1 = df['preferred\_offer\_types'].drop\_duplicates()

print(unique\_values1)

df['preferred\_offer\_types'] = df['preferred\_offer\_types'].replace({pd.NA: 'Without Offers'})

print(df)

**1.2, Mã hoá dữ liệu:**

Sử dụng LabelEncoder để mã hoá các cột chứa dữ liệu object

from sklearn.preprocessing import LabelEncoder

for col in df.columns:

if df[col].dtype == 'object': # Kiểm tra nếu cột chứa kiểu dữ liệu object

label\_encoder = LabelEncoder()

df[col] = label\_encoder.fit\_transform(df[col]) # Mã hóa cột object

print(df)

Sau khi mã hoá xong , các cột chứa dữ liệu object sẽ trở về dữ liệu int , float

**1.3, Loại bỏ cột theo tương quan cao dựa vào threashold:**

def drop\_correlated\_columns(df, threshold):

"""

Loại bỏ các cột có độ tương quan cao với các cột khác trong DataFrame dựa vào ngưỡng threshold.

Parameters:

- df: DataFrame đầu vào

- threshold: Ngưỡng tương quan để quyết định xem cột có nên bị loại bỏ hay không

Returns:

- DataFrame mới sau khi loại bỏ các cột không cần thiết

"""

# Tính toán ma trận tương quan

corr\_matrix = df.corr().abs()

# Tạo một ma trận boolean với True ở vị trí cần loại bỏ

upper\_triangle = corr\_matrix.where(np.triu(np.ones(corr\_matrix.shape), k=1).astype(bool))

# Tìm các cột cần loại bỏ

to\_drop = [column for column in upper\_triangle.columns if any(upper\_triangle[column] > threshold)]

# Loại bỏ các cột không cần thiết

df\_filtered = df.drop(to\_drop, axis=1)

return df\_filtered

# Loại bỏ các cột có độ tương quan > 0.7

df\_filtered = drop\_correlated\_columns(df1, 0.7)

df\_filtered

=> Loại bỏ 1 cột còn 22 cột

**1.4, Xử lí ngoại lệ**

Phương pháp IQR:

def remove\_outliers\_iqr(df, threshold=1.5):

"""

Loại bỏ các điểm ngoại lệ từ DataFrame sử dụng phương pháp IQR (khoảng giữa phần tư thứ nhất và thứ ba).

Parameters:

- df: DataFrame đầu vào

- threshold: Ngưỡng để xác định điểm ngoại lệ. Một giá trị phổ biến là 1.5.

Returns:

- DataFrame mới sau khi loại bỏ điểm ngoại lệ

"""

Q1 = df.quantile(0.25)

Q3 = df.quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - threshold \* IQR

upper\_bound = Q3 + threshold \* IQR

return df[(df >= lower\_bound) & (df <= upper\_bound)].dropna()

# Loại bỏ điểm ngoại lệ từ toàn bộ dữ liệu

filtered\_df = remove\_outliers\_iqr(df\_filtered )

**1.5, Chuẩn hoá dữ liệu:**

PCA:

import numpy as np

import pandas as pd

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

# Chuẩn hóa dữ liệu

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(X)

# Áp dụng PCA với số lượng thành phần chính mong muốn

n\_components = 21

pca = PCA(n\_components=n\_components)

pca.fit(scaled\_data)

# Biến đổi dữ liệu ban đầu thành các thành phần chính

pca\_data = pca.transform(scaled\_data)

# Tạo DataFrame mới từ các thành phần chính

pca\_df = pd.DataFrame(data=pca\_data, columns=[f'PC{i}' for i in range(1, n\_components + 1)])

pca\_df

**2, Ứng dụng mô hình học máy cho dữ liệu bằng thuật toán SVM và MLP**

Sau khi dữ liệu đã được tiền xử lý, ta có thể tiến hành xây dựng mô hình học máy cho bài toán.

Cụ thể, các bước tiến hành như sau:

Chia dữ liệu thành tập train và test để huấn luyện mô hình.

Chạy mô hình bằng tập train.

Kiểm tra độ chính xác của mô hình bằng tập test và các độ đo như F1,recall,…

Đầu vào: tập dữ liệu

Đầu ra:

* 1. **Chia dữ liệu**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(pca\_df, Y, test\_size=0.2, random\_state=42)**

Sử dụng hàm train\_test\_split chia tập dữ liệu thành 2 phần trong đó:

Tập Ttrain chiếm 80%

Tập Test chiếm 20%

**#** In ra kích thước của tập huấn luyện và tập kiểm tra

print("Kích thước tập huấn luyện:", X\_train.shape, y\_train.shape)

print("Kích thước tập kiểm tra:", X\_test.shape, y\_test.shape)

**1.2. Huấn luyện mô hình**

Theo 2 thuật toán MLP và SVM

**1.3. Kiểm tra độ chính xác của mô hình**

**Accuracy (Độ chính xác):**

Accuracy là tỷ lệ phần trăm giữa số lượng dự đoán đúng và tổng số lượng mẫu trong tập dữ liệu kiểm tra.

Công thức tính: Accuracy = (Số lượng dự đoán đúng) / (Tổng số lượng mẫu)

Độ chính xác càng cao thì mô hình càng tốt. Tuy nhiên, độ chính xác không phản ánh được hiệu suất của mô hình trong trường hợp dữ liệu mất cân bằng hoặc khi có các lớp có tỷ lệ khác nhau.

**Classification Report (Báo cáo phân loại):**

Classification report là một báo cáo chi tiết về hiệu suất của mô hình trên các lớp (classes) khác nhau trong bài toán phân loại.

Bao gồm các thông số như precision, recall, F1-score và support cho mỗi lớp.

Precision (độ chính xác): Tỷ lệ giữa số lượng dự đoán đúng của lớp đó và tổng số lượng mẫu được dự đoán là lớp đó.

Recall (độ nhớ lại): Tỷ lệ giữa số lượng dự đoán đúng của lớp đó và tổng số lượng mẫu thuộc lớp đó trong dữ liệu thực tế.

F1-score: Trung bình điều hòa của precision và recall, cung cấp một phép đo tổng thể về hiệu suất của mô hình.

Kết quả :

SVM:

accuracy\_score: 0.8586283185840708

precision\_score: 0.8586283185840708

recall\_score: 0.8586283185840708

f1\_score: 0.858628318584071

Độ chính xác của mô hình: 0.8586283185840708

MLP:Accuracy: 0.9128318584070797

**Phần 3: TỔNG KẾT**

Trong bài báo cáo này, chúng em đã thực hiện một loạt các bước tiền xử lý dữ liệu và chạy mô hình học máy để giải quyết bài toán. Chúng em đã đạt được một số kết quả quan trọng và nhận định sau đây:

**3.1, Tiền xử lý dữ liệu:**

Chúng em đã thu thập và tiền xử lý một tập dữ liệu để chuẩn bị cho quá trình huấn luyện mô hình.

Các phương pháp tiền xử lý đã được áp dụng để giải quyết các vấn đề ban đầu của dữ liệu.

**3.2, Chạy mô hình học máy:**

Chúng em đã chọn và huấn luyện một số mô hình học máy trên tập dữ liệu đã tiền xử lý.

Qua quá trình huấn luyện, mô hình cho kết quả tương đối ổn định trên các thang đo như recall, f1 score,…

**3.3, Nhận xét và kết luận:**

Từ quá trình tiền xử lý dữ liệu và chạy mô hình học máy, chúng em nhận thấy như sau:

**Uu điểm:**

Các phương pháp tiền xử lý khá nhanh và hiệu quả.

Có thể thay đổi nhiều phương pháp để linh hoạt tùy vào bài toán.

**Hạn chế:**

Có thể một số phương pháp gây ra mất thông tin đặc trưng về dữ liệu.

Nhiều phương pháp có thể tốn nhiều thời gian và không hiệu quả khi dữ liệu không phù hợp.

Chúng em cũng xin đề xuất một vài ý tưởng:

Nghiên cứu và thử nghiệm các phương pháp tiền xử lý dữ liệu mới để giảm thiểu mất mát thông tin và tăng cường khả năng tổng quát hóa của mô hình.

Áp dụng các kỹ thuật tiền xử lý dữ liệu tiên tiến như feature engineering để tạo ra các đặc trưng mới có thể cải thiện hiệu suất của mô hình.

Nhìn chung, quá trình này đã cung cấp cái nhìn sâu rộng về quá trình tiền xử lý dữ liệu và chạy mô hình học máy trong việc giải quyết bài toán thực tế. Những kết quả và nhận định từ bài toán này có thể hữu ích cho các nghiên cứu và ứng dụng trong tương lai.

**3.4, Nguồn tài liệu tham khảo:**

**Lý thuyết:**

Slide bài giảng của TS. Tạ Quang Chiểu, đại học Thủy Lợi.

Machinelearningcoban.com

Book data Preprocessing Python master