## 1.1. Mô tả bài toán

Tim mạch là bệnh lý xuất hiện âm thầm nhưng để lại nhiều hậu quả nghiêm trọng đối với tính mạng. Trong những năm trở lại đây, tỷ lệ tử vong do các bệnh tim mạch ngày một tăng ở các nước đang phát triển. Vì vậy, mục đích là phân tích xem có bị mắc bệnh tim mạch hay không .

Trong báo cáo này, em sử dụng bài toán phân lớp nhị phân để giải quyết bài toán, dự đoán xem người đó có mặc bệnh tim hay không mắc bệnh tim. Kỹ thuật học máy sử dụng là Logistic Regression. Phương pháp đánh giá mô hình học máy được sử dụng là đo độ chính xác Accuracy và Confusion Matrix.

## 1.2 Thu thập và mô tả dữ liệu

Trong bài toán dự đoán bệnh tim, em đã tìm hiểu và thu thập đươc tập dữ liệu các đặc điểm của người dùng Internet. Tập dữ liệu nói trên được em thu thập từ **https://www.kaggle.com/zeeshanmulla/heart-disease-dataset.** Tập dữ liệu bao gồm 304 dòng mẫu với 14 cột đặc điểm của người dùng.

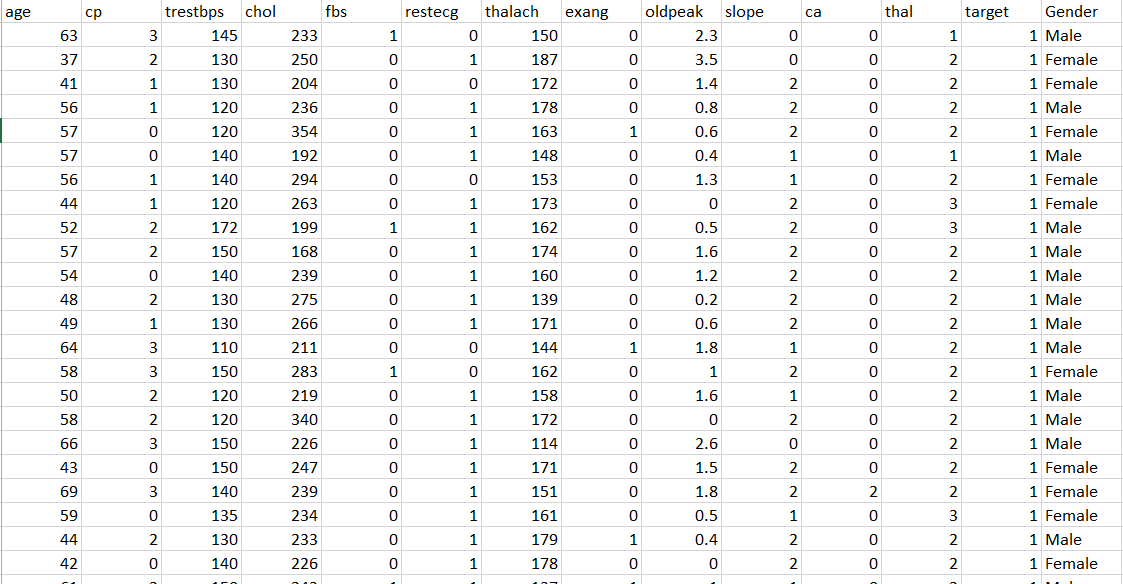
**1.2.1. Mô tả thuộc tính**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuộc tính** | **Mô tả** |
| 1 | Age | Tuổi |
| 2 | Gender | Giới tính |
| 3 | cp | Kiểu đau ngực (1 = đau thắt ngực điển hình; 2 = đau thắt ngực không điển hình; 3 = đau không đau thắt ngực; 4 = không có triệu chứng) |
| 4 | trestbps | Huyết áp lúc nghỉ (tính bằng mm Hg khi nhập viện) |
| 5 | chol | Cholestoral trong huyết thanh tính bằng mg / dl |
| 6 | fbs | Đường huyết lúc đói> 120 mg / dl (1 = true; 0 = false) |
| 7 | restecg | Kết quả điện tâm đồ lúc nghỉ (0 = bình thường; 1 = có ST-T; 2 = phì đại) |
| 8 | thalach | Nhịp tim tối đa đạt được |
| 9 | exang | Đau thắt ngực do tập thể dục (1 = có; 0 = không) |
| 10 | oldpeak | ST trầm cảm gây ra do tập thể dục liên quan đến nghỉ ngơi |
| 11 | slope | độ dốc của đoạn ST bài tập cao điểm (1 = dốc lên; 2 = bằng phẳng; 3 = dốc xuống) |
| 12 | ca | số lượng các mạch chính (0-3) được tô màu bởi bột màu |
| 13 | thal | 3 = bình thường; 6 = khuyết tật cố định; 7 = khiếm khuyết có thể đảo ngược |
| 14 | target | thuộc tính dự đoán - chẩn đoán bệnh tim (tình trạng bệnh mạch máu) (Giá trị 0 = thu hẹp đường kính <50%; Giá trị 1 => thu hẹp đường kính 50%) |

Với 14 đặc điểm của người dùng Internet như trên, để phù hợp với yêu cầu là chuẩn đoán bệnh tim, chúng em chọn thuộc tính **Y là target** ở đây làm biến mục tiêu cần tìm, các thuộc tính còn lại làm biến giải thích.

**1.2.2. Số lượng mẫu**

Tập dữ liệu của em bao gồm 304 dòng tương ứng với số lượng mẫu. Dưới đây là một số mẫu có trong tập dữ liệu:

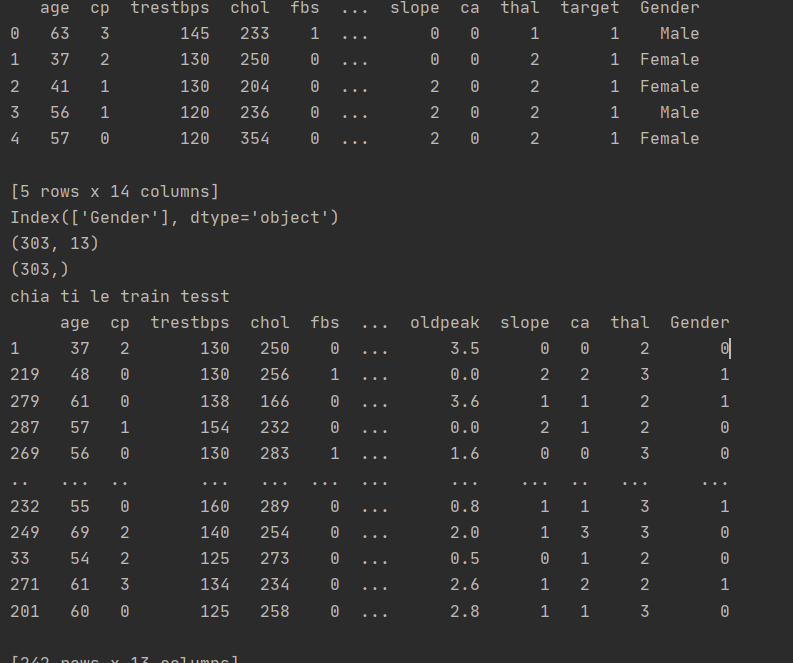


## 1.3. Xử lí dữ liệu

**Bước 1:** đọc file dữ liệu ở dạng csv, xử lí các cột dữ liệu ở dạng object:”Gender”. Chọn biến mục tiêu y = “target”

**Code và kết quả:**

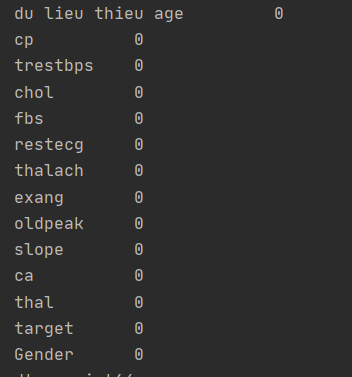
df = pd.read\_csv("heart1.csv")  
data\_frame = df.dropna()  
print (df.head())  
objList = df.select\_dtypes(include = "object").columns #lấy thông tin các cột object để xử lí  
print(objList)  
for cols in objList:  
 df[cols] = preprocessing.LabelEncoder().fit\_transform(df[cols].astype(str))  
X = df.drop(["target"], axis = 1)  
y = df["target"]  
print (X.shape)  
print (y.shape)



**Bước 2:** Kiểm tra dữ liệu thiếu

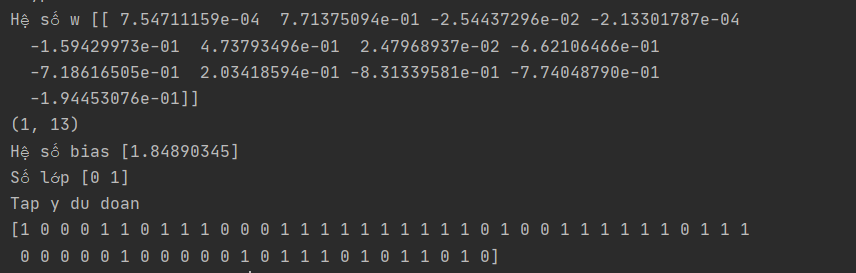
**Code và kết quả:**

print("du lieu thieu",df.isnull().sum())



**Bước 3:** Chia tập dữ liệu thành 2 tập train và test với **Code và kết quả:**

from sklearn.linear\_model import LogisticRegression  
log\_model = LogisticRegression(max\_iter = 1000).fit(X\_train,y\_train)  
  
y\_pred = log\_model.predict(X\_test) #y\_prediction là y dự đoán được  
print ("Hệ số w", log\_model.coef\_)  
print (log\_model.coef\_.shape)  
print ("Hệ số bias", log\_model.intercept\_)  
print ("Số lớp", log\_model.classes\_)  
#print ("Số vòng lặp", log\_model.n\_iter\_)  
print ("Tap y du doan")  
print (y\_pred)



## 1.4. Kỹ thuật học máy sử dụng và kiểm tra có mất cân bằng hay không

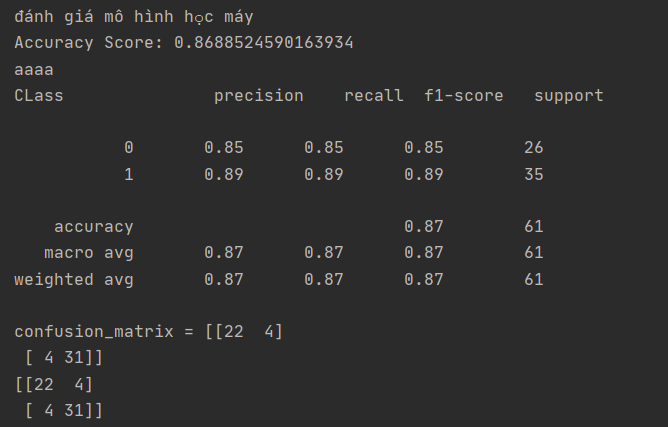
Với bài toán trên, em sử dụng kỹ thuật học máy Logistic Regression. Logistic Regression là 1 thuật toán phân loại được dùng để gán các đối tượng cho 1 tập hợp giá trị rời rạc (như 0, 1, ...). Một ví dụ điển hình là phân loại Email, gồm có email công việc, email gia đình, email spam,... Giao dịch trực tuyến có là an toàn hay không an toàn, khối u lành tính hay ác tình. Thuật toán trên dùng hàm sigmoid logistic để đưa ra đánh giá theo xác suất.

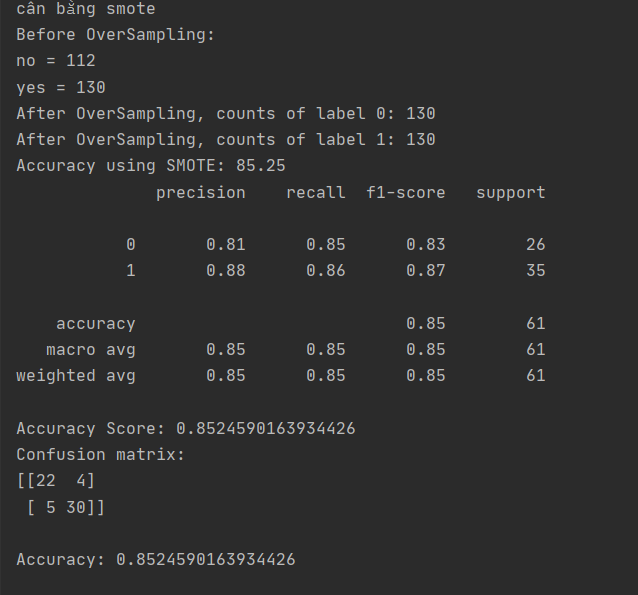
Kỹ thuật Logistic Regression sử dụng hàm Sigmoid làm hàm dự đoán. Hàm này có dạng:

https://images.viblo.asia/43965a36-f47e-4f23-8d05-695a315f7ce5.png

**Code và kết quả:**

print("đánh giá mô hình học máy")  
from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix  
  
print("Accuracy Score:", accuracy\_score(y\_test, y\_pred))  
print("aaaa")  
print("CLass",classification\_report(y\_test,y\_pred))  
print ('confusion\_matrix =',confusion\_matrix(y\_test,y\_pred))  
  
print(confusion\_matrix(y\_test,y\_pred))  
  
#cân bằng smote  
print("cân bằng smote")  
  
print('Before OverSampling:')  
  
print('no =', sum(y\_train == 0))  
  
print('yes =', sum(y\_train == 1))  
  
sm = SMOTE(random\_state=2)  
  
X\_train\_res, y\_train\_res = sm.fit\_sample(X\_train, y\_train)  
  
print("After OverSampling, counts of label 0: {}".format(sum(y\_train\_res == 0)))  
  
print("After OverSampling, counts of label 1: {}".format(sum(y\_train\_res == 1)))  
  
model = LogisticRegression(max\_iter=10000).fit(X\_train\_res, y\_train\_res)  
  
yhat = model.predict(X\_test)  
  
accuracy = accuracy\_score(y\_test, yhat)  
  
print('Accuracy using SMOTE: %.2f' % (accuracy \* 100))  
  
print(classification\_report(y\_test, yhat))  
from sklearn.metrics import accuracy\_score  
print("Accuracy Score:", accuracy\_score(y\_test, yhat))  
from sklearn.metrics import confusion\_matrix  
cnf\_matrix = confusion\_matrix(y\_test, yhat)  
print('Confusion matrix:')  
print(cnf\_matrix)  
print('\nAccuracy:', np.diagonal(cnf\_matrix).sum()/cnf\_matrix.sum())





**1.5. Phương pháp đánh giá độ chính xác**

Phương pháp đánh giá độ chính xác được sử dụng trong bài toán này là Accuracy và Confusion Matrix.

**Accuracy:** Cách đánh giá này tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. Giả sử, ta đếm được 7 điểm dữ liệu được dự đoán đúng trên tổng số 10 điểm. Như vậy, Accuracy Score của bài toán sẽ là 0.7 (hay 70%).

**Kết quả:**



**1.6. Kết luận**

Logistic Regression chạy khá nhanh và chuẩn xác, tuy nhiên nhược điểm của nó cũng như các thuật toán phân lớp khác như SVM, DecisionTreeClassifier, Random Forest là rất nhạy cảm với dữ liệu mất cân bằng. Kết quả khi chạy có thể rất đẹp nhưng mô hình dự đoán thường không đáng tin cậy. Chính vì thế, để đảm bảo dữ liệu được huấn luyện tốt và cho ra dự đoán chính xác, cần tiến hành xử lý mất cân bằng ngay từ khâu ban đầu trước khi chạy mô hình dự đoán kết quả.