**Họ và tên: Nguyễn Văn Linh – 18020762 – Nhóm 1**

**Titanic classification – report**

1. **Github - Folder titanic classification gồm:**

* Report.docx (report task report)
* Titanic\_classification.ipynb (code implementation)
* File predict\_data\_submit.csv (file .csv output label dữ liệu test.csv)

1. **Quá trình xây dựng classification model đối với bộ dữ liệu**

* Data pre-processing

Lấy dữ liệu, import thư viện

Xử lý dữ liệu thiếu (missing data)

Mã hóa dữ liệu phân loại (encode categorical data)

Loại bỏ dữ liệu ít hay không cần thiết

Feature scaling

Chia tập dữu liệu thành 2 tập training set và test test

* Training process

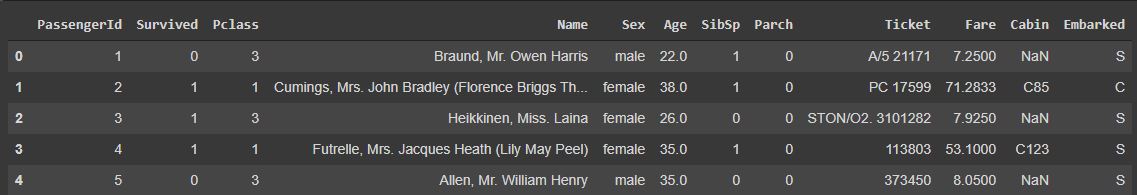
Train model với logistic regression

* testing process
* evaluate precision

**2.1 data pre-processing**

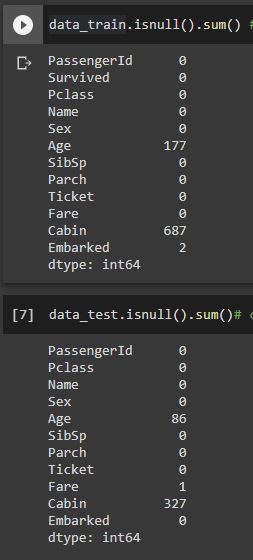
* import thư viện load dữ liệu từ file

load train.csv



tương tự với test.csv

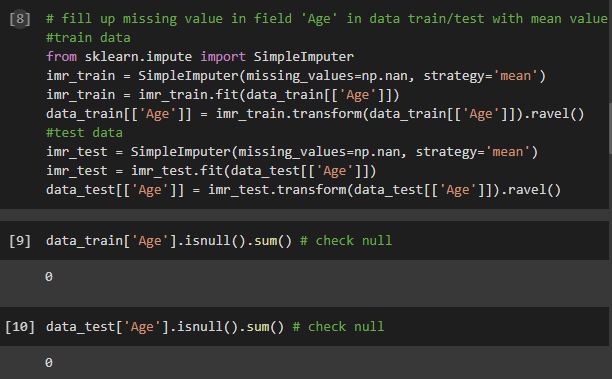
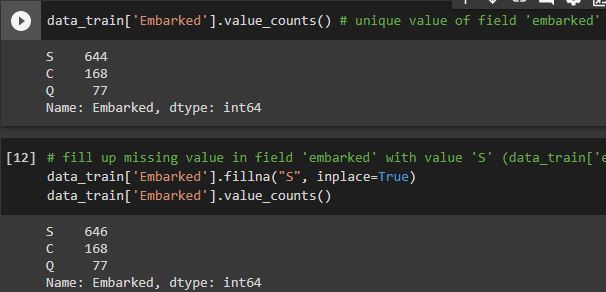
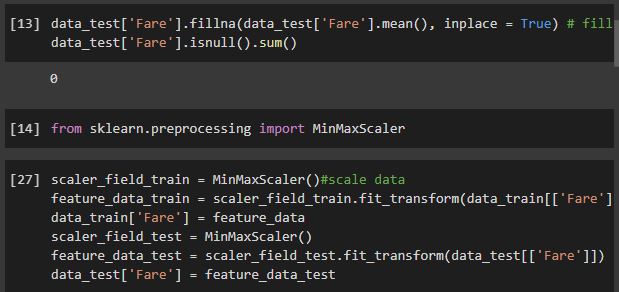
* Từ đó ta có thông tin các features từ bộ dữ liệu titanic
* PassengerId – id của hành khách
* Survived – sống sót hay không (0 – không, 1 - có)
* Pclass – loại vé (1 – hạng cao, 2 – hạng trung, 3 – hạng thấp)
* Sex – giới tính (male - female)
* Age – tuổi
* SibSp – số lượng anh em, hoặc vợ chồng đi cùng
* Parch – số lượng phụ huynh, hoặc con cái đi cùng
* Ticket – số tiền vé
* Cabin – số cabin
* Embarked – bến khởi hành (C, Q, S)
* Kiểm tra dữ liệu thiếu (missing value)



* Ta thấy ở

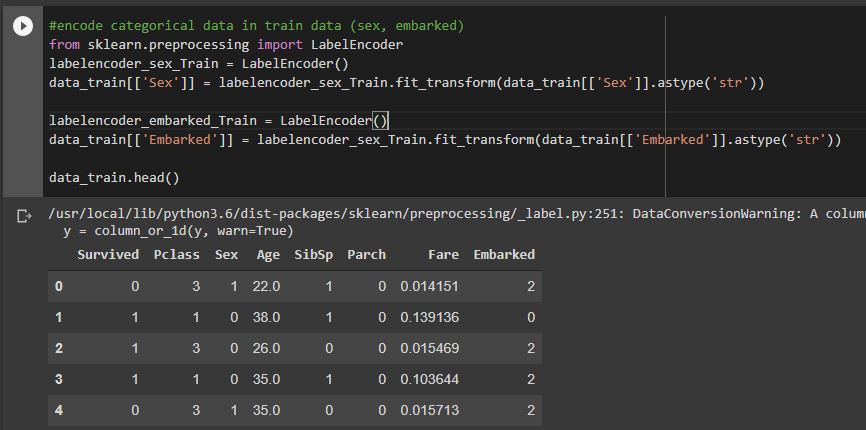
dữ liệu train: item ‘Age’ thiếu 177 giá trị, ‘Embarked’ thiếu 2 giá trị, ‘Cabin’ thiếu 687 giá trị.

dữ liệu test: item ‘Age’ thiếu 86 giá trị, ‘Fare’ là 1 còn ‘Cabin’ thiếu 327 giá trị.

* Để xử lý missing value với item ‘Age’ ta sẽ fill up các feature null bằng giá trị trung bình (mean) của cả item ‘Age’ cho cả 2 tập dữ liệu train và test
* Tiếp đến item ‘Embarked’, chỉ có 3 giá trị cho tất cả các feature (S – C - Q) với giá trị “S” chiếm nhiều nhất trong các feature, ta sẽ fill up các giá trị null bằng giá trị “S” cho tập dữ liệu train.
* Tiếp theo là item ‘Fare’ thiếu 1 giá trị trong tập dữ liệu test, ta cũng sẽ fill up bằng giá trị mean của item ‘Fare’. Và sau đó ta sẽ là scaling data.
* Loại bỏ dữ liệu có ít feature, hay không cần thiết

Với các item không hay ít ảnh hưởng đến quá trình training của bộ dữ liệu, ta nêm xem xét loại bỏ chúng. ở đây là item ‘PassengerId’ ‘Ticket’, ‘Name’, ‘Cabin’.

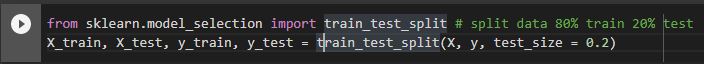
* Mã hóa dữ liệu phân loại (encode categorical data)

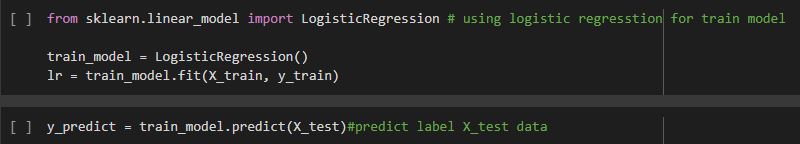
Với item ‘Sex’, ‘Embarked’, từ string sang numberic

**2.2 . Traning processing**

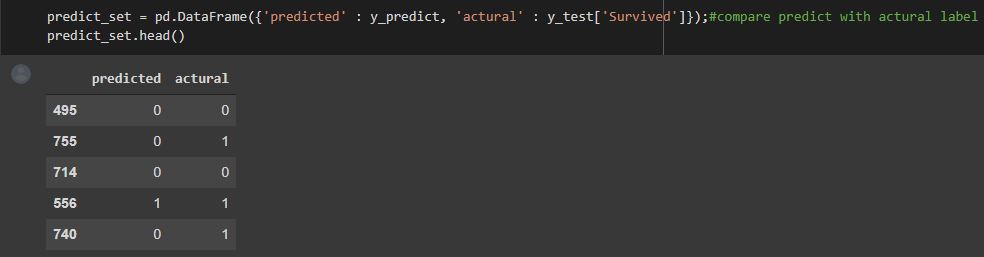
- Sau khi dữ liệu được xử lý

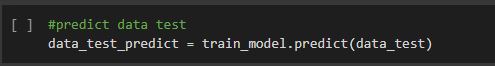
- Ta chia tập train thành 2 phần: training và testing (test\_size = 0.2) để kiểm tra đánh giá độ chính xác (precision) của mô hình.



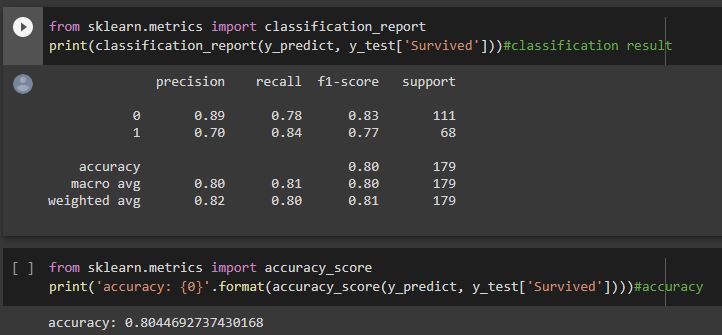
- sử dụng logistic regression để train model

**2.3 Testing processing**

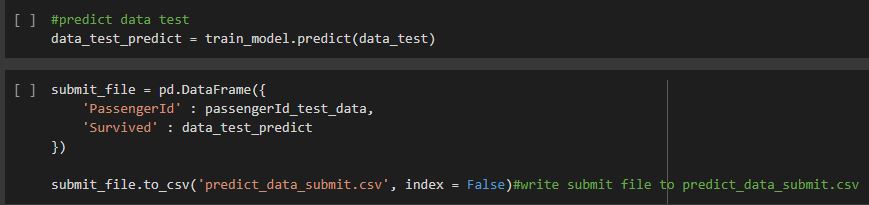
 Kiểm tra dữ liệu dự đoán so với thự tế

predict với bộ dữ liệu test

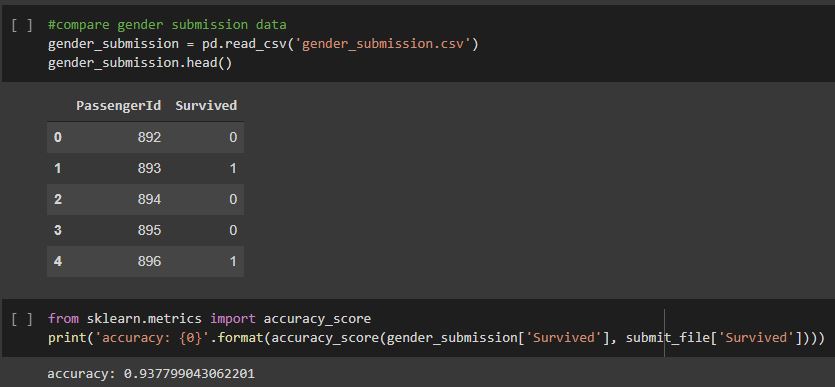
**2.4 Evaluate precision**

 Đánh giá độ chính xác khi dự đoán của model với bộ dữ liệu train (test\_size = 0.2)

* Kết quả với độ chính xác : 0,8

 Đánh giá độ chính xác khi dự đoán với bộ dữ liệu test.csv

* Kết quả là output label của bộ dữ liệu test được ghi vào file predict\_data\_submit.csv. với 418 features là số hành khách với kết quả dự đoán survived (1 - 0)

 Kết quả này so sánh với bộ dữ liệu gender\_submission với kết quả thực tế - label là của bộ dữ liệu test.csv chưa được gắn label. Với độ chính xác 0.93

Submit file predict\_data\_submit.csv trên kaggle với score : 0,77033 ( bad :)) )

