TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ MÔN**

**NHẬP MÔN HỌC MÁY**

*Người hướng dẫn:* **PGS.TS Lê Anh Cường**

*Người thực hiện*: **Nguyễn Thanh Triều - 52100495**

**Lương Quang Vinh - 52100865**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ MÔN**

**NHẬP MÔN HỌC MÁY**

*Người hướng dẫn:* **PGS.TS Lê Anh Cường**

*Người thực hiện*: **Nguyễn Thanh Triều - 52100495**

**Lương Quang Vinh - 52100865**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến thầy về những kiến thức quý báu và sự hướng dẫn tận tâm trong môn nhập môn học máy. Thầy đã giúp em hiểu rõ hơn về chủ đề này và trang bị cho em những kỹ năng quan trọng. Em hết sức biết ơn công lao của thầy và cam kết nỗ lực hết mình để hoàn thành tốt môn học. Mong rằng thầy sẽ tiếp tục hỗ trợ và khuyến khích em trong quá trình học tập.

Một lần nữa, em xin bày tỏ lòng biết ơn sâu sắc đến thầy.

Trân trọng,

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của Thầy Lê Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 6 tháng 12 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Thanh Triều*

*Lương Quang Vinh*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Môn Học máy cung cấp cho sinh viên các kiến thức nền tảng về máy học và ứng dụng của nó trong thực tế. Trong khóa học, chúng em được làm quen với các khái niệm cơ bản như training data, models, learning algorithms. Giảng viên cũng giới thiệu cho sinh viên một số thuật toán học máy phổ biến hiện nay như linear regression, logistic regression, decision trees, neural networks, v.v...

Ngoài ra, chúng em cũng được thực hành xây dựng mô hình học máy để giải quyết các bài toán thực tế như dự đoán, phân loại, tóm tắt văn bản, v.v...Thông qua đó, sinh viên có cơ hội ứng dụng lí thuyết để giải quyết các vấn đề trong thực tế với dữ liệu thực. Đồng thời rèn luyện kĩ năng lập trình với Python để xây dựng các mô hình Machine Learning.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc154265242)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc154265243)

[TÓM TẮT iv](#_Toc154265244)

[MỤC LỤC 1](#_Toc154265245)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 2](#_Toc154265246)

[PHẦN 2 : 3](#_Toc154265247)

[TÀI LIỆU THAM KHẢO 33](#_Toc154265248)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1. Import thư viện 4](#_Toc154265249)

[Hình 2. Đọc dữ liệu 5](file:///E:\MC\Final\Bai2_52100865_52100495.docx#_Toc154265250)

[Hình 3. Định dạng và thống kê 5](#_Toc154265251)

[Hình 4. Tính toán dữ liệu còn thiếu 6](#_Toc154265252)

[Hình 5. Thêm cột mục tiêu 6](#_Toc154265253)

[Hình 6. Hiển thị dữ liệu dạng Numerical 7](#_Toc154265254)

[Hình 7. Vẽ và hiển thị phân phối của các biến số trong dữ liệu 7](#_Toc154265255)

[Hình 8. Vẽ biểu đồ KDE 7](#_Toc154265256)

[Hình 9. Biểu đồ về phân phối của các biến số 8](#_Toc154265257)

[Hình 10. Dữ liệu dạng Categorical 9](#_Toc154265258)

[Hình 11. Phân phối đặc trưng 10](#_Toc154265259)

[Hình 12. Biểu đồ boxenplot thể hiện phân phối đặc trưng 11](#_Toc154265260)

[Hình 13. Hiển thị heatmap 12](#_Toc154265261)

[Hình 14. Heatmap 13](#_Toc154265262)

[Hình 15. Kiểm tra mối tương quan giữa các cột 13](#_Toc154265263)

[Hình 16. Bỏ cột "WeightInKilograms" 14](#_Toc154265264)

[Hình 17. Hiển thị dữ liệu dạng Categorical và thực hiện tiền xử lý 14](#_Toc154265265)

[Hình 18. Chia dữ liệu 15](#_Toc154265266)

[Hình 19. Train và đánh giá mô hình 15](#_Toc154265267)

[Hình 20. Đặc trưng và đánh giá mô hình 15](#_Toc154265268)

[Hình 21. Biểu đồ tròn 16](#_Toc154265269)

[Hình 22. Chọn mẫu cho bài toán 16](#_Toc154265270)

[Hình 23. Baseline 17](#_Toc154265271)

[Hình 24. Tạo hàm huấn luyện và đánh giá mô hình 17](#_Toc154265272)

[Hình 25. Sử dụng mô hình Random Forest 18](#_Toc154265273)

[Hình 26. Sử dụng mô hình Naive Bayes 18](#_Toc154265274)

[Hình 27. Sử dụng mô hình Decision Tree 19](#_Toc154265275)

[Hình 28. Định nghĩa một RandomForest 20](#_Toc154265276)

[Hình 29. Huấn luyện Random Foreswst trên tập đã được Resample 20](#_Toc154265277)

[Hình 30. Kết quả trên tập đã được Resample 20](#_Toc154265278)

[Hình 31. Huấn luyện mô hình Random Forest trên tập dữ liệu gốc 21](#_Toc154265279)

[Hình 32. Kết quả trên tập dữ liệu gốc 21](#_Toc154265280)

[Hình 33. Sử dụng Feed Forward Neural Network để giải quyết bài toán 22](#_Toc154265281)

[Hình 34. Kết quả khi sử dụng Feed Forward Neural Network để giải quyết bài toán 22](#_Toc154265282)

[Hình 35. Sử dụng Recurrent Neural Network để giải quyết bài toán 23](#_Toc154265283)

[Hình 36. Kết quả khi sử dụng Feed Forward Neural Network để giải quyết bài toán 23](#_Toc154265284)

[Hình 37. Tránh overfitting trên mô hình FFNN bằng cách thêm dropout layer 24](#_Toc154265285)

[Hình 38. Tránh overfitting trên FFNN 24](#_Toc154265286)

[Hình 39. Tránh overfitting trên FFNN với Early Stopping 25](#_Toc154265287)

[Hình 40. Kết quả 25](#_Toc154265288)

[Hình 41. Tránh overfitting trên mô hình RNN 26](#_Toc154265289)

[Hình 42. Kết quả 26](#_Toc154265290)

[Hình 43. Tránh overfitting trên mô hình RandomForest 27](#_Toc154265291)

[Hình 44. Tránh overfitting trên Naïve Bayes 28](#_Toc154265292)

[Hình 45. Tránh overfitting trên Decision Tree 29](#_Toc154265293)

[Hình 46. Đoạn mã Python 30](#_Toc154265294)

**DANH MỤC BẢNG**

**No table of figures entries found.**

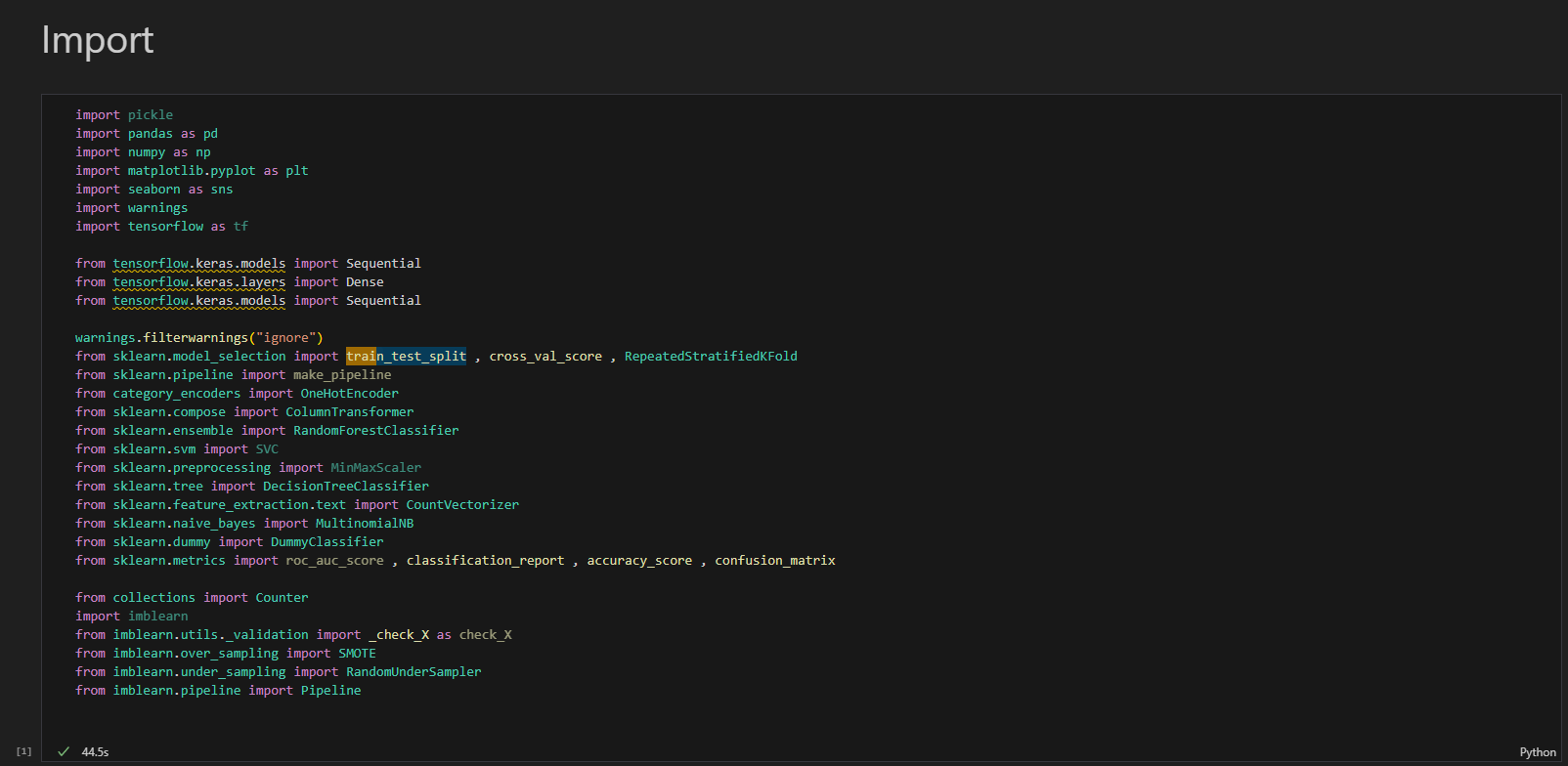
PHẦN 2 :

Đưa ra một bài toán dự đoán có thể giải quyết bằng học máy (machine leanring) với các yêu cầu sau:

* Số Feature/Attribute gồm nhiều kiểu: categorial và numerical;
* Dữ liệu phải chưa được học, thực tập trên lớp và trong bài tập về nhà;

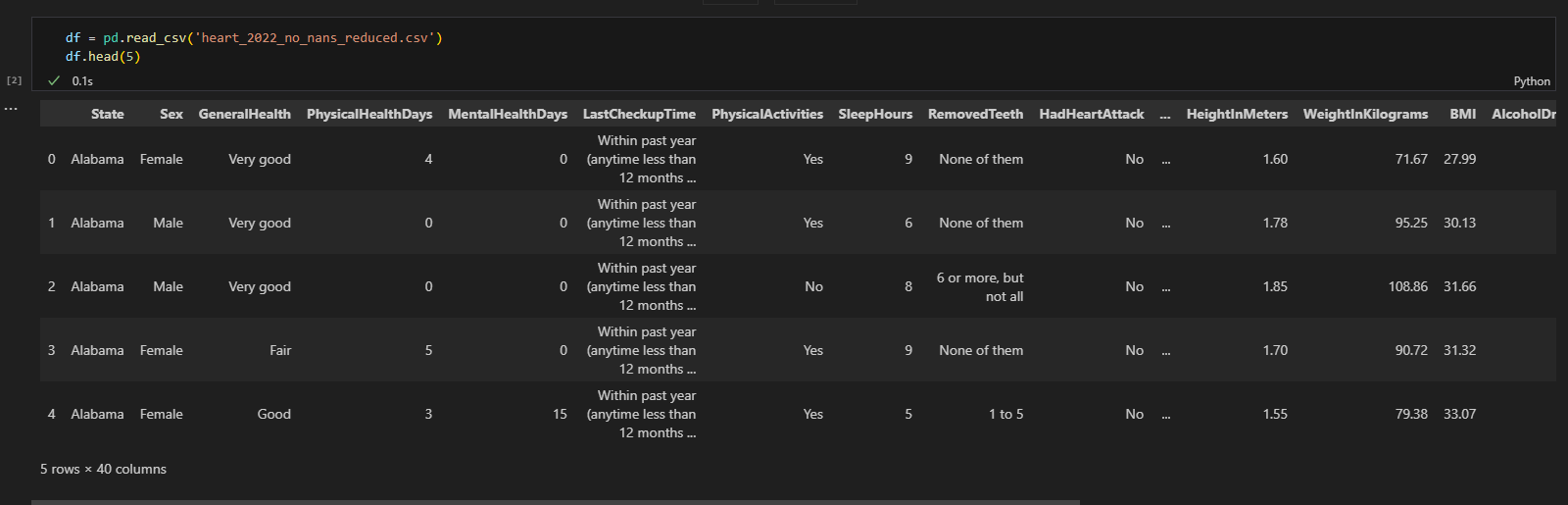
1. Phân tích thống kê trên dữ liệu, vẽ các đồ thị để hiểu bài toán, hiểu dữ liệu. Tìm hiểu các đặc trưng và đánh gía vai trò của các đặc trưng đối với mục tiêu bài toán;

* Đầu tiên, import thư viện sử dụng trong bài toán



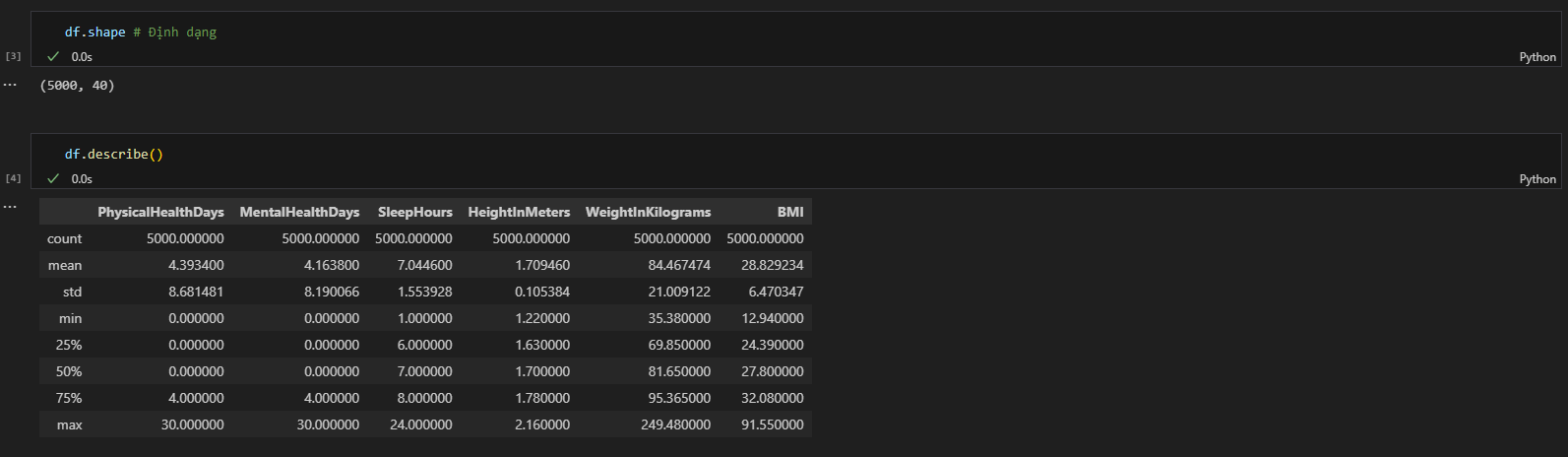
Hình . Import thư viện

* Đọc dữ liệu bằng thư viện pandas và hiển thị 5 dòng đầu tiên của dữ liệu



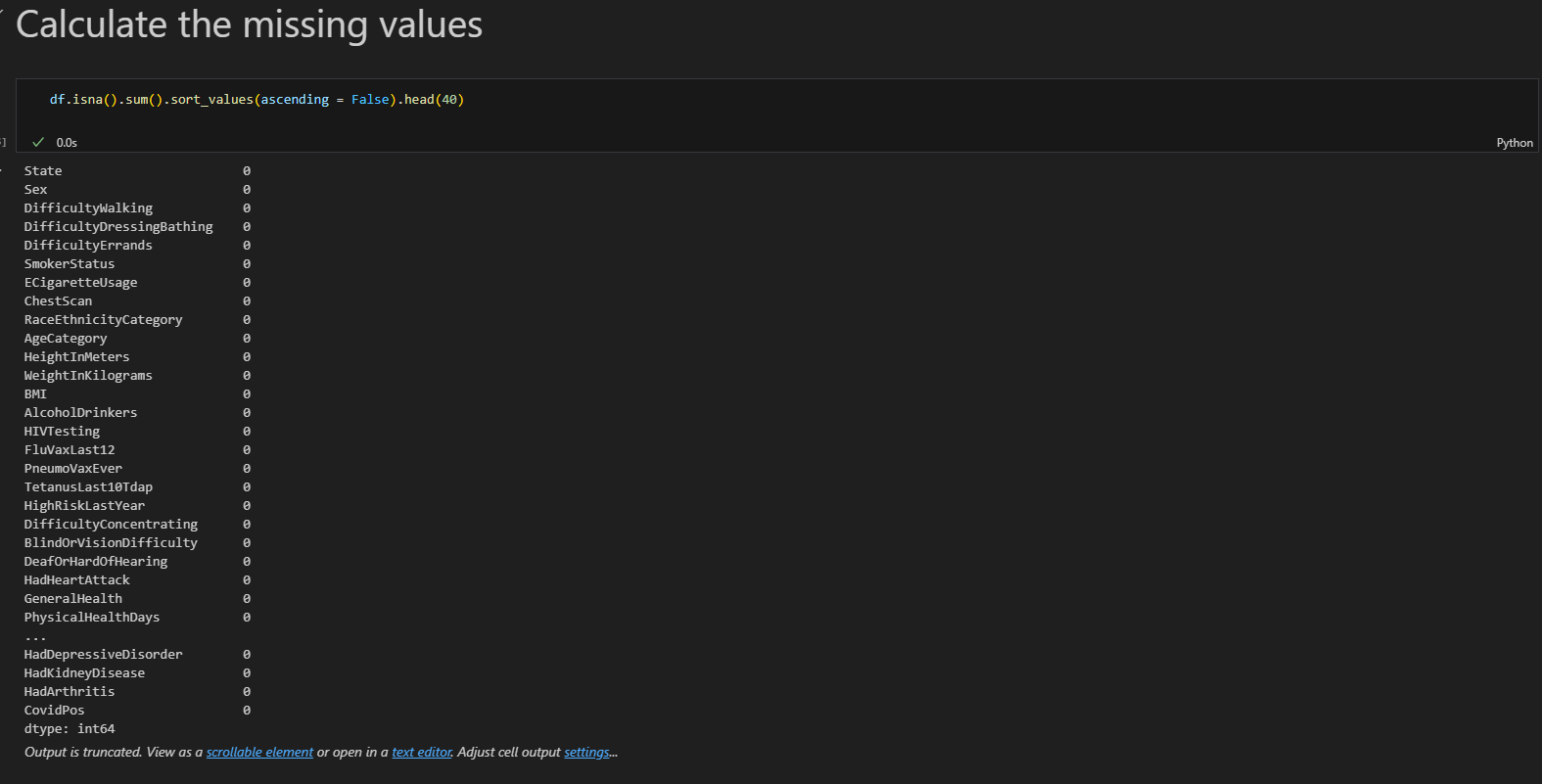
Hình . Đọc dữ liệu

* Định dạng hiển thị số dòng, cột của dữ liệu và thống kê mô tả



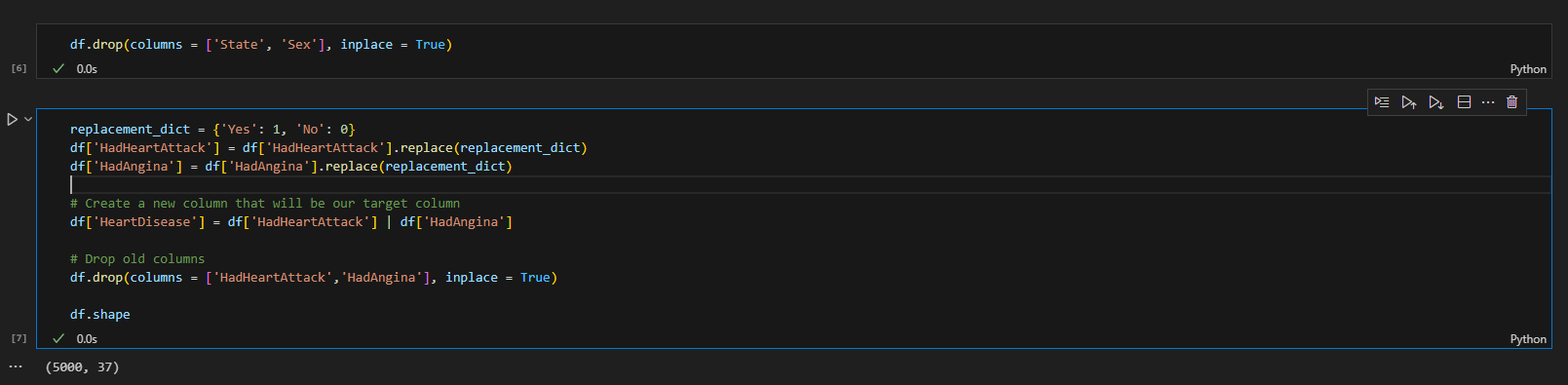
Hình . Định dạng và thống kê

* Tính toán giá trị còn thiếu của dữ liệu. Ở dữ liệu nhóm chúng em chọn thì không có dữ liệu nào bị thiếu



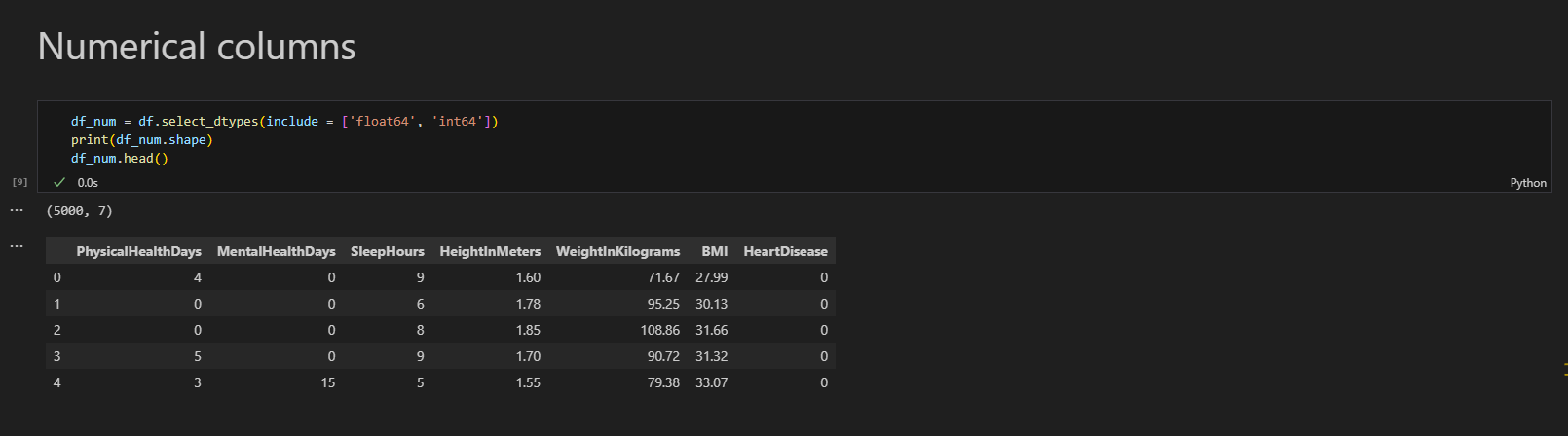
Hình . Tính toán dữ liệu còn thiếu

* Loại bỏ đi những cột không cần thiết, thay vào đó thêm cột dữ liệu “HeartDisease” là cột mục tiêu của bài toán



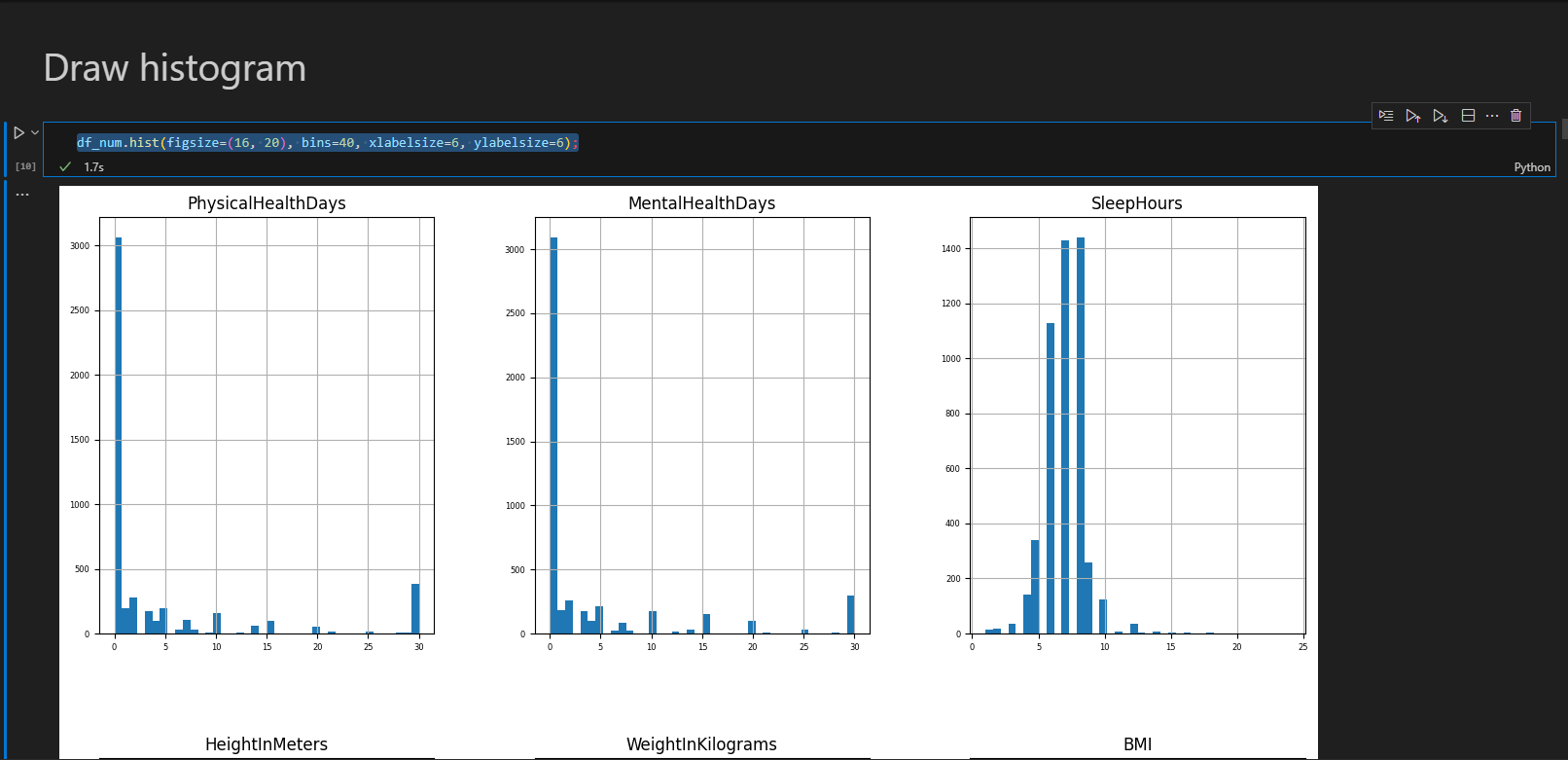
Hình . Thêm cột mục tiêu

* Hiển thị dữ liệu dạng Numerical



Hình . Hiển thị dữ liệu dạng Numerical

* Vẽ biểu đồ cho tổng quan về phân phối của các biến số trong dữ liệu



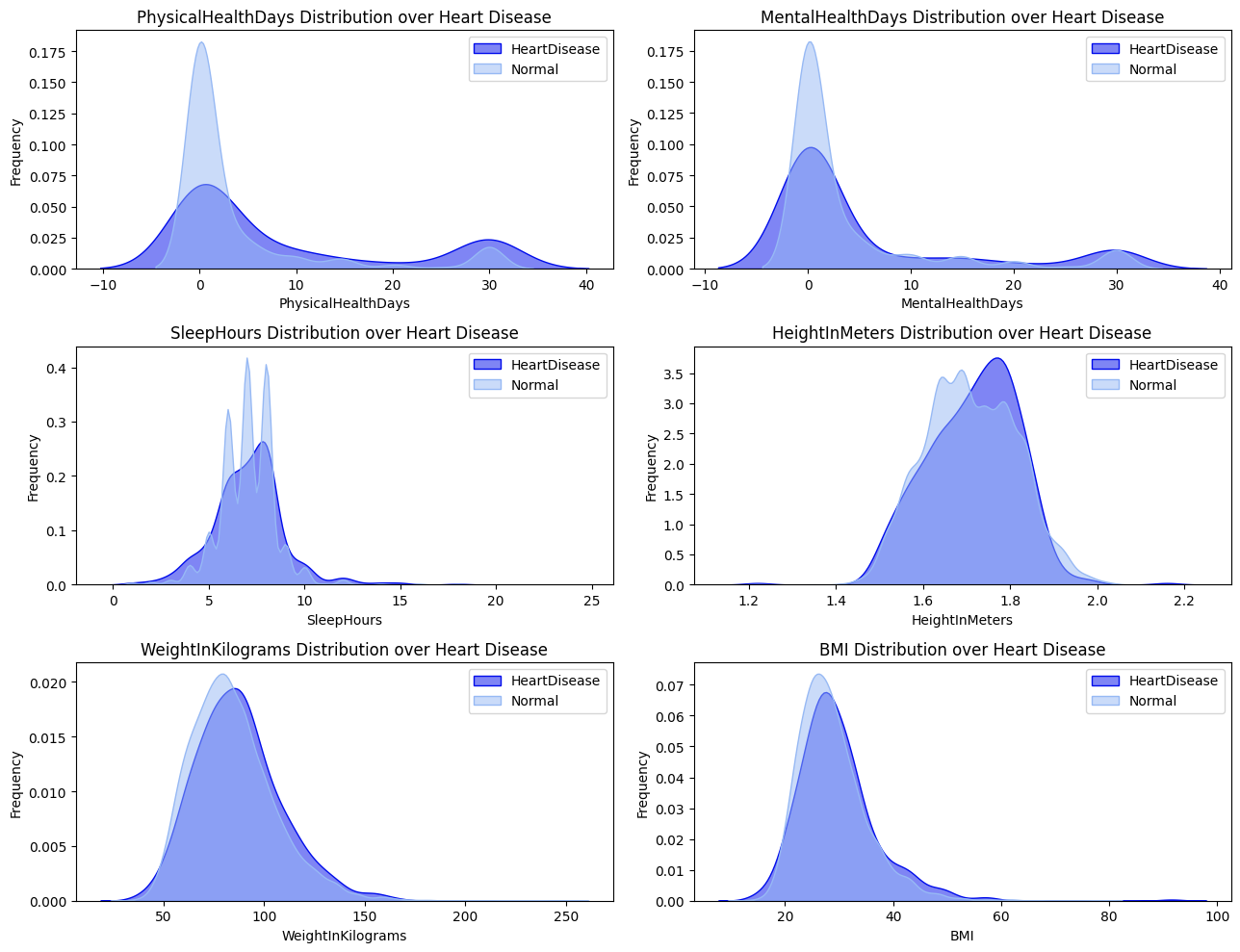
Hình . Vẽ và hiển thị phân phối của các biến số trong dữ liệu

* Vẽ một lưới các biểu đồ KDE để so sánh phân phối của các biến số trong tập dữ liệu theo nhóm HeartDisease ( bị bệnh tim ) và Normal ( không bị bệnh tim )



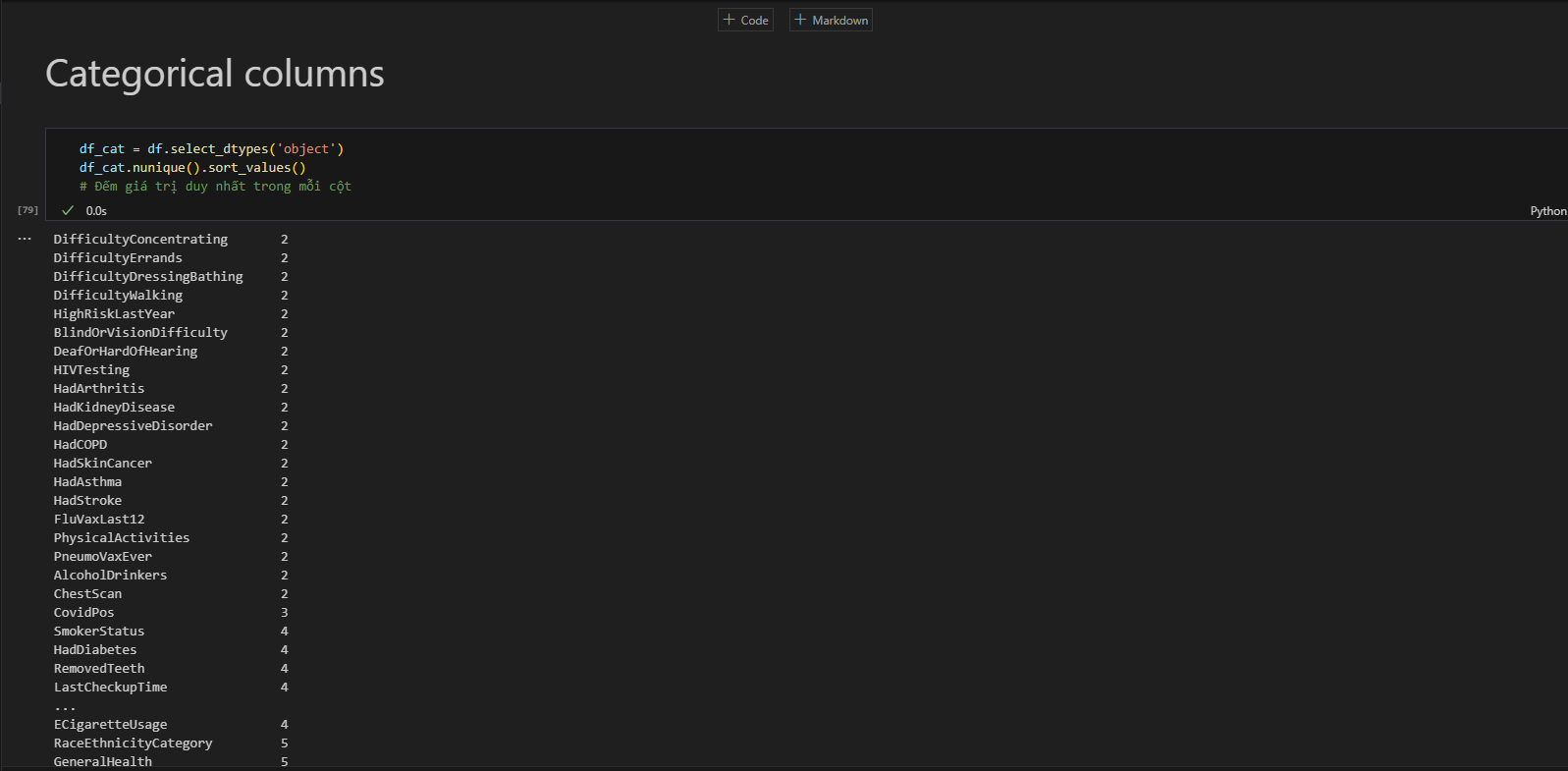
Hình . Vẽ biểu đồ KDE

Kết quả nhận được :



Hình . Biểu đồ về phân phối của các biến số

* Hiển thị dữ liệu dạng Categorical

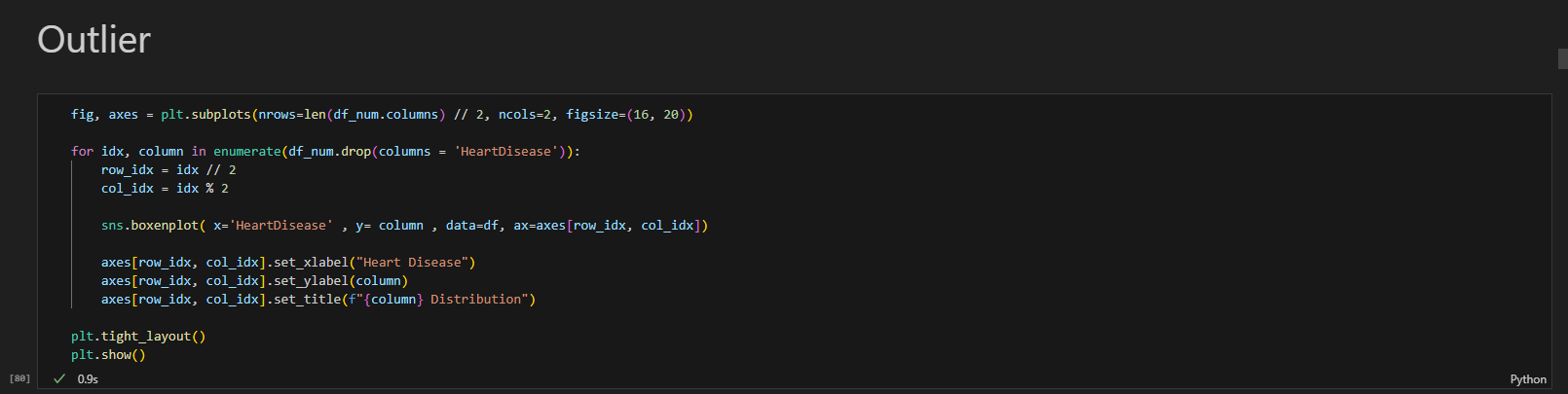


Hình . Dữ liệu dạng Categorical

Đoạn mã dưới đây tạo một lưới các biểu đồ boxenplot (một biến thể của box plot) để thể hiện phân phối của các đặc trưng (cột) dựa trên giá trị của biến mục tiêu HeartDisease.

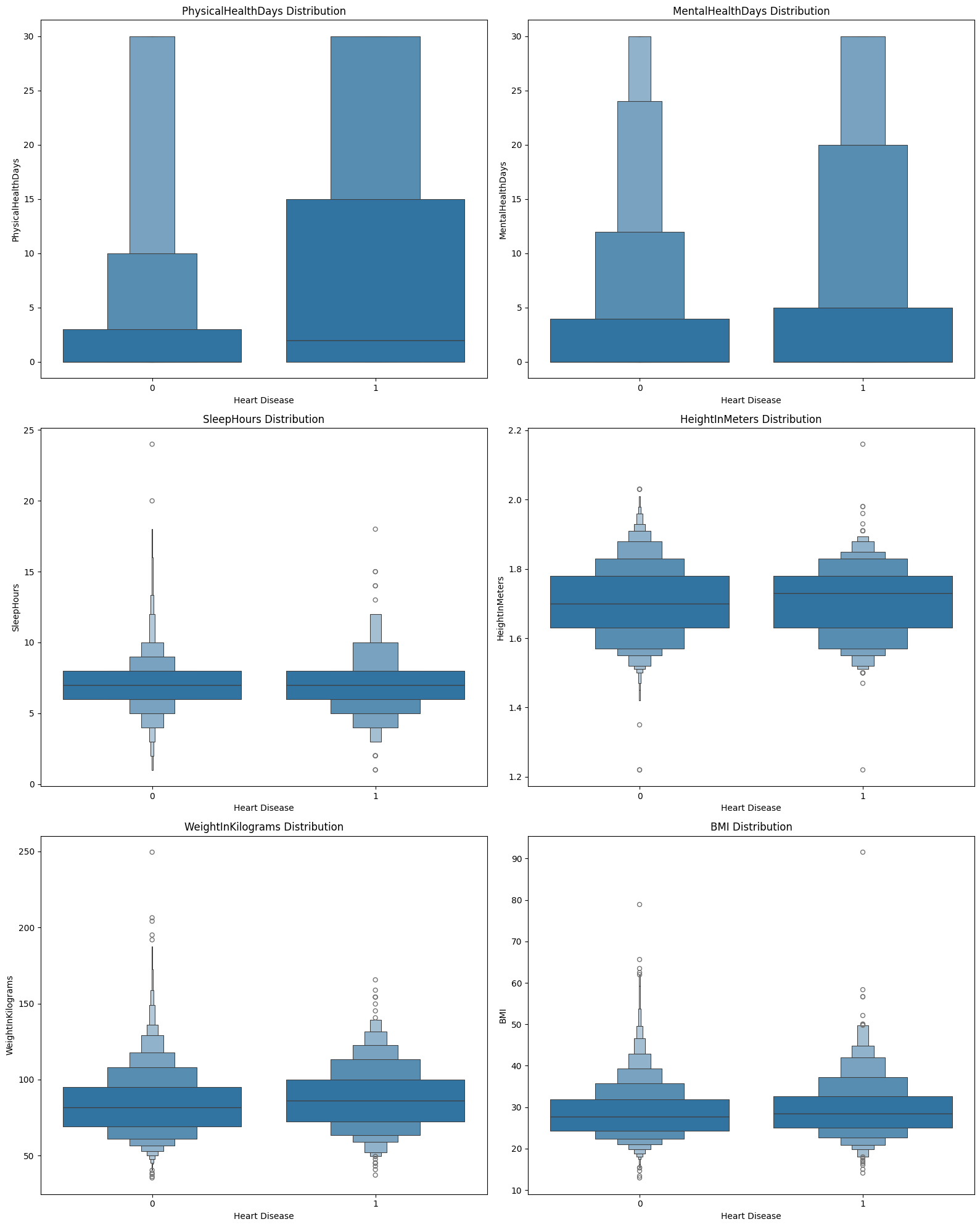
Cụ thể, đối với mỗi đặc trưng (cột) trong df\_num, nó vẽ một boxenplot hiển thị phân phối của giá trị của đặc trưng đó cho hai trường hợp: khi HeartDisease là 1 (có bệnh tim) và khi HeartDisease là 0 (không có bệnh tim). Các boxenplot được sắp xếp thành lưới, với mỗi dòng của lưới chứa hai biểu đồ tương ứng với một đặc trưng.

Các boxenplot giúp bạn nhìn nhận về sự phân tán và sự chênh lệch giữa các giá trị của các đặc trưng dựa trên biến mục tiêu. Bạn có thể đánh giá xem có sự khác biệt đáng kể nào giữa các phân phối khi HeartDisease là 1 và khi HeartDisease là 0 hay không.



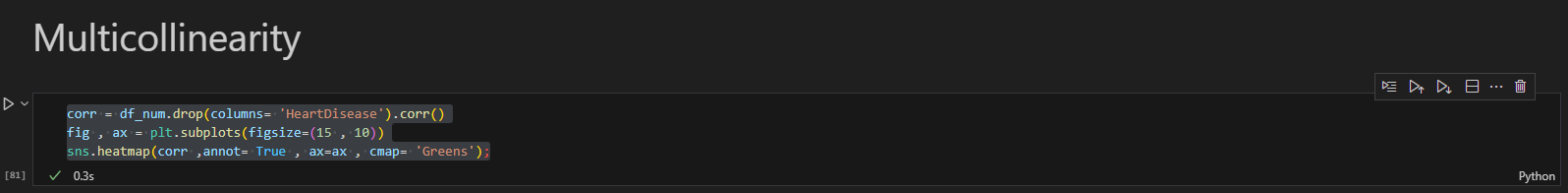
Hình . Phân phối đặc trưng

Kết quả thu được :



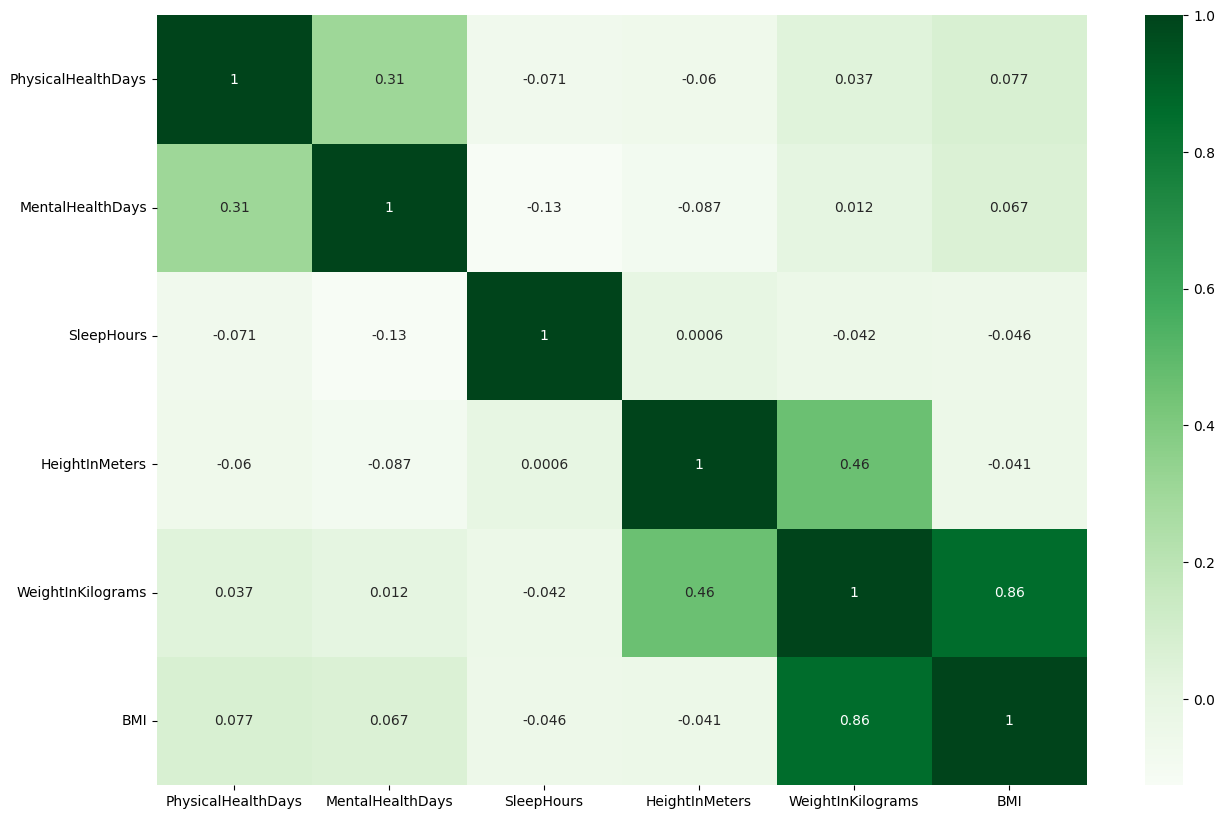
Hình . Biểu đồ boxenplot thể hiện phân phối đặc trưng

* Hiển thị heatmap của ma trận tương quan giữa các đặc trưng biến số trong df\_num, loại bỏ cột “HeartDisease” vì quan tâm đến mối quan hệ giữa các đặc trưng và biến mục tiêu.
* Ý nghĩa của heatmap này là hiển thị mức độ tương quan giữa từng cặp đặc trưng trong DataFrame của bạn. Các giá trị tương quan thường nằm trong khoảng từ -1 đến 1, trong đó
* Giá trị dương: Tương quan hoàn toàn tích cực (nếu một biến tăng, biến kia cũng tăng và ngược lại).
* Giá trị âm: Tương quan hoàn toàn tiêu cực (nếu một biến tăng, biến kia giảm và ngược lại).
* Giá trị 0: Không có tương quan tuyến tính giữa các biến.
* Các ô màu trên heatmap thường được màu sáng hơn nếu tương quan càng cao và màu đậm hơn nếu tương quan càng thấp.



Hình . Hiển thị heatmap

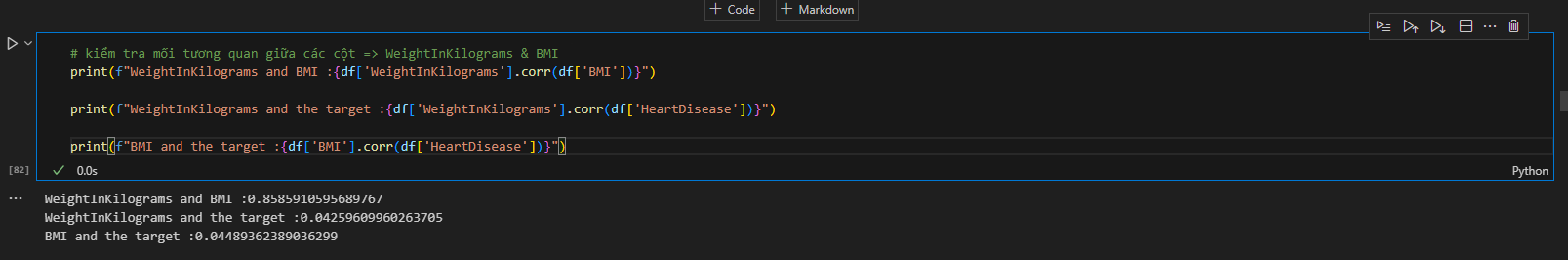
Kết quả thu được :



Hình . Heatmap

Nhìn vào heatmap, bạn có thể xác định được những cặp đặc trưng nào có mức tương quan cao, điều này có thể cung cấp thông tin quan trọng khi chọn đặc trưng cho mô hình hoặc khi phân tích ảnh hưởng của chúng đối với biến mục tiêu.

* Kiểm tra mối tương quan giữa các cột



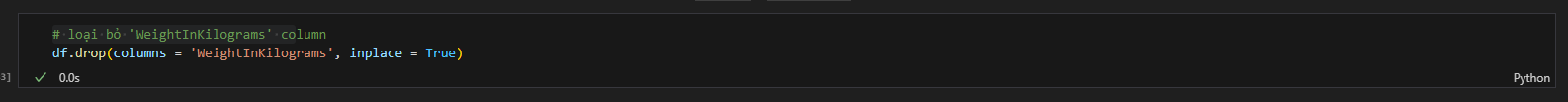
Hình . Kiểm tra mối tương quan giữa các cột

Ta có thể thấy, mối quan hệ giữa “WeightInKilograms” và “BMI” là khá mạnh, nên việc giữ lại cả hai đặc trưng trên có thể dẫn đến hiện tượng đa cộng tuyến.

Hiện tượng đa cộng tuyến xảy ra khi có sự tương quan cao giữa hai hoặc nhiều biến độc lập trong mô hình hồi quy tuyến tính. Điều này có thể tạo ra vấn đề khi ước lượng hệ số của các biến tương quan cao và làm tăng độ chệch của các ước lượng.

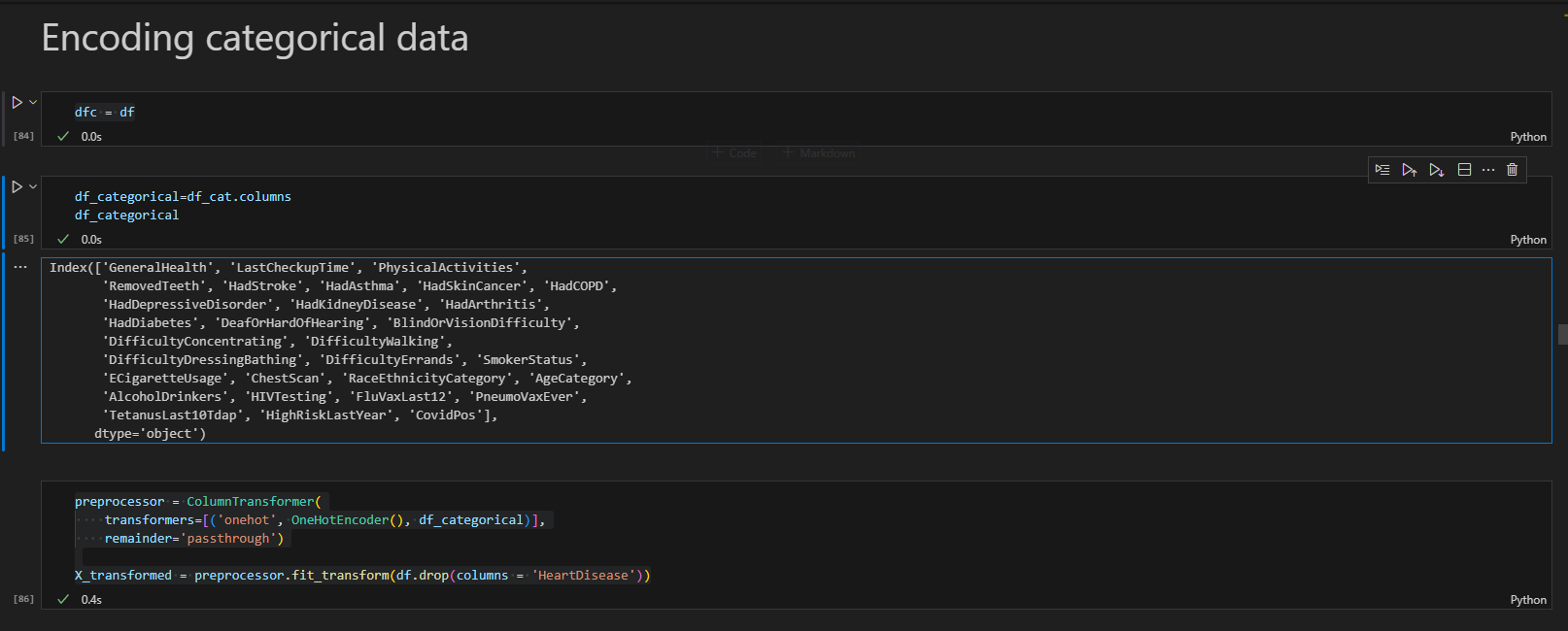
Nếu "WeightInKilograms" và "BMI" chứa thông tin tương tự và cùng đóng góp vào việc giải thích biến mục tiêu ("HeartDisease"), việc giữ lại một trong hai có thể giúp mô hình trở nên đơn giản hơn mà không làm mất đi quá nhiều thông tin. Tuy nhiên, quyết định loại bỏ biến nào nên dựa trên kiểm tra kỹ lưỡng của dữ liệu và hiểu rõ về mối quan hệ giữa các biến.

Nên sẽ loại bỏ cột “WeightInKilograms”

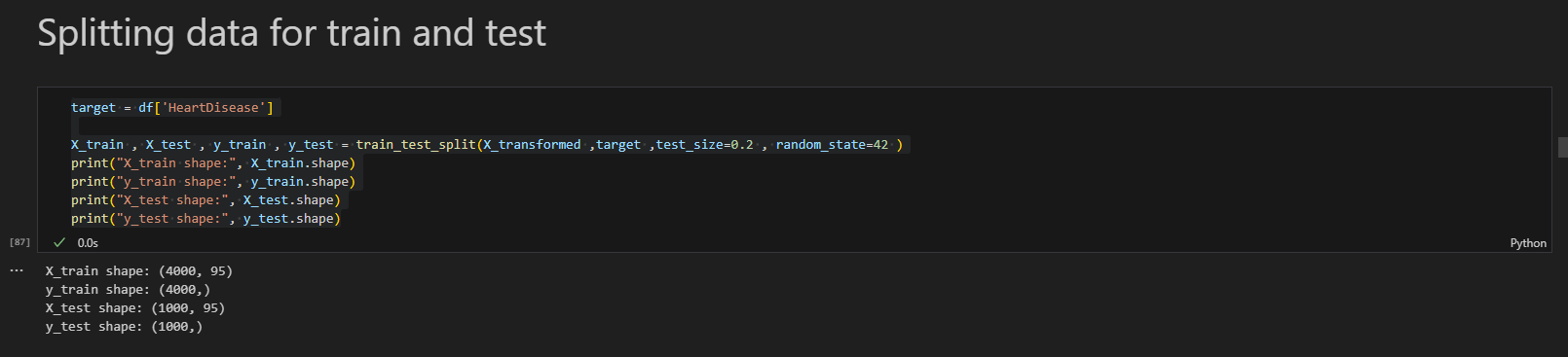


Hình . Bỏ cột "WeightInKilograms"

* Hiển thị dữ liệu dạng Categorical và thực hiện tiền xử lý dữ liệu trên DataFrame ‘df’ trước khi sử dụng chúng để huấn luyện mô hình học máy

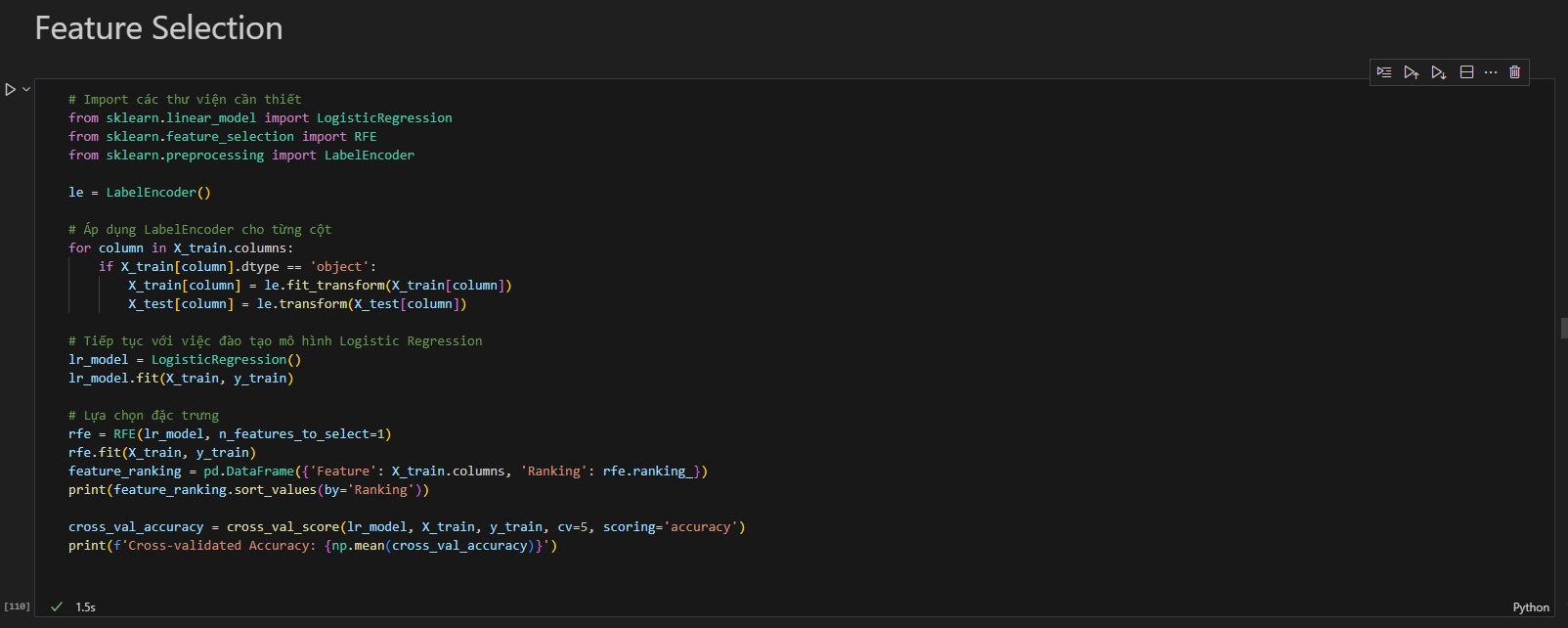


Hình . Hiển thị dữ liệu dạng Categorical và thực hiện tiền xử lý

* Chia dữ liệu thành tập huấn luyện (X\_train, y\_train) và tập kiểm tra (X\_test, y\_test) bằng cách sử dụng hàm train\_test\_split từ scikit-learn 

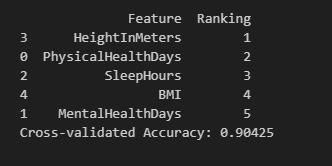
Hình . Chia dữ liệu

* Train và đánh giá mô hình Logistic Regression sau khi đã tiền xử lý dữ liệu



Hình . Train và đánh giá mô hình

Kết quả thu được : Các đặc trưng của dữ liệu và trung bình của mô hình trên tập huấn luyện sau cross-validation



Hình . Đặc trưng và đánh giá mô hình

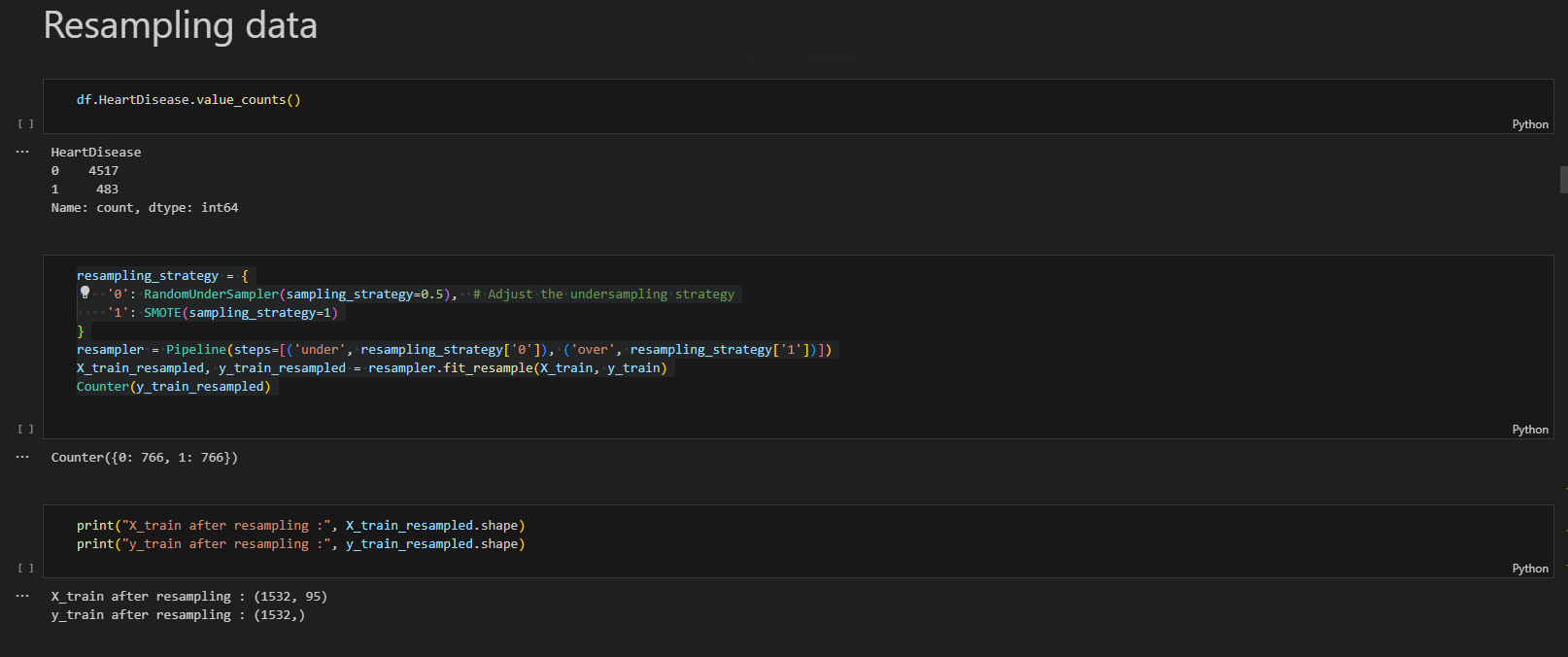
- Vẽ biểu đồ tròn thể hiện phân phối của biến mục tiêu “HeartDisease”



Hình . Biểu đồ tròn

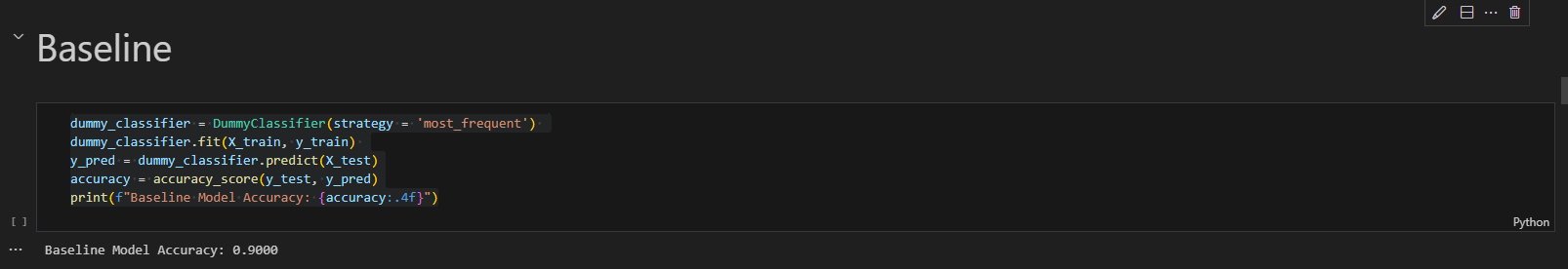
Nhìn vào biểu đồ trên ta có thể thấy, biến mục tiêu “HeartDisease” chiếm 9,7%. Điều đó có nghĩa là các mẫu không có vấn đề về tim mạch chiếm 90,3%, còn lại 9,7% đó là phần trăm mẫu có vấn đề về tim mạch

* Chọn mẫu cho bài toán



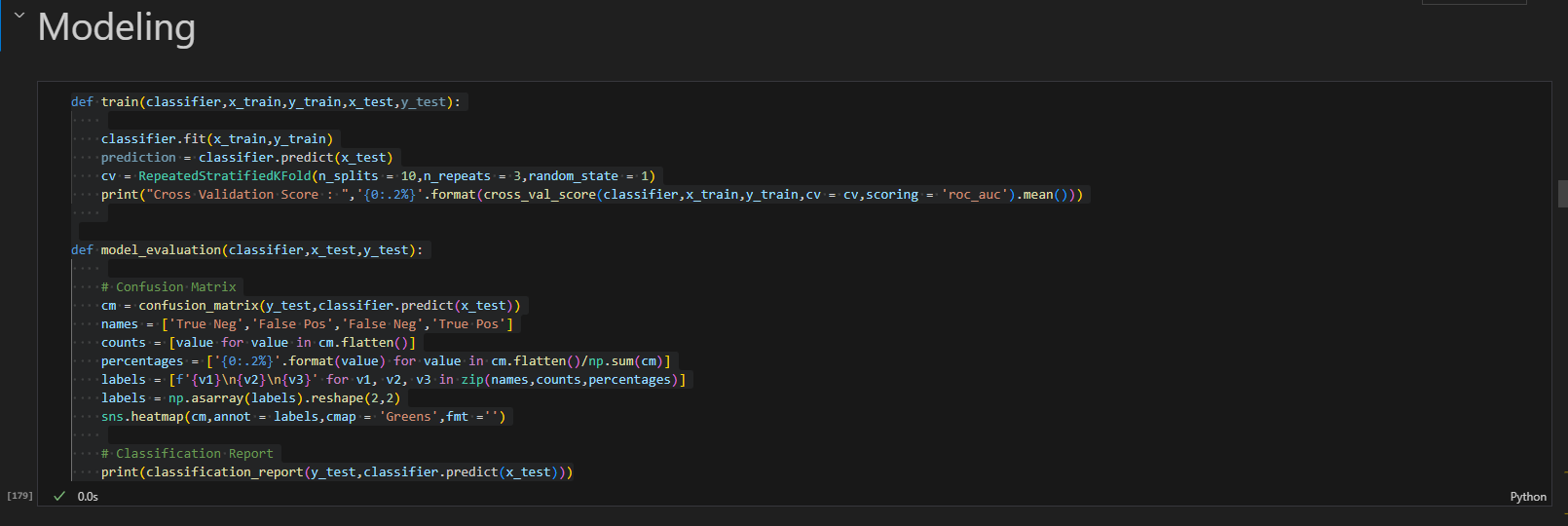
Hình . Chọn mẫu cho bài toán

* Tạo và đánh giá một mô hình cơ sở (baseline) sử dụng Dummy Classifier



Hình . Baseline

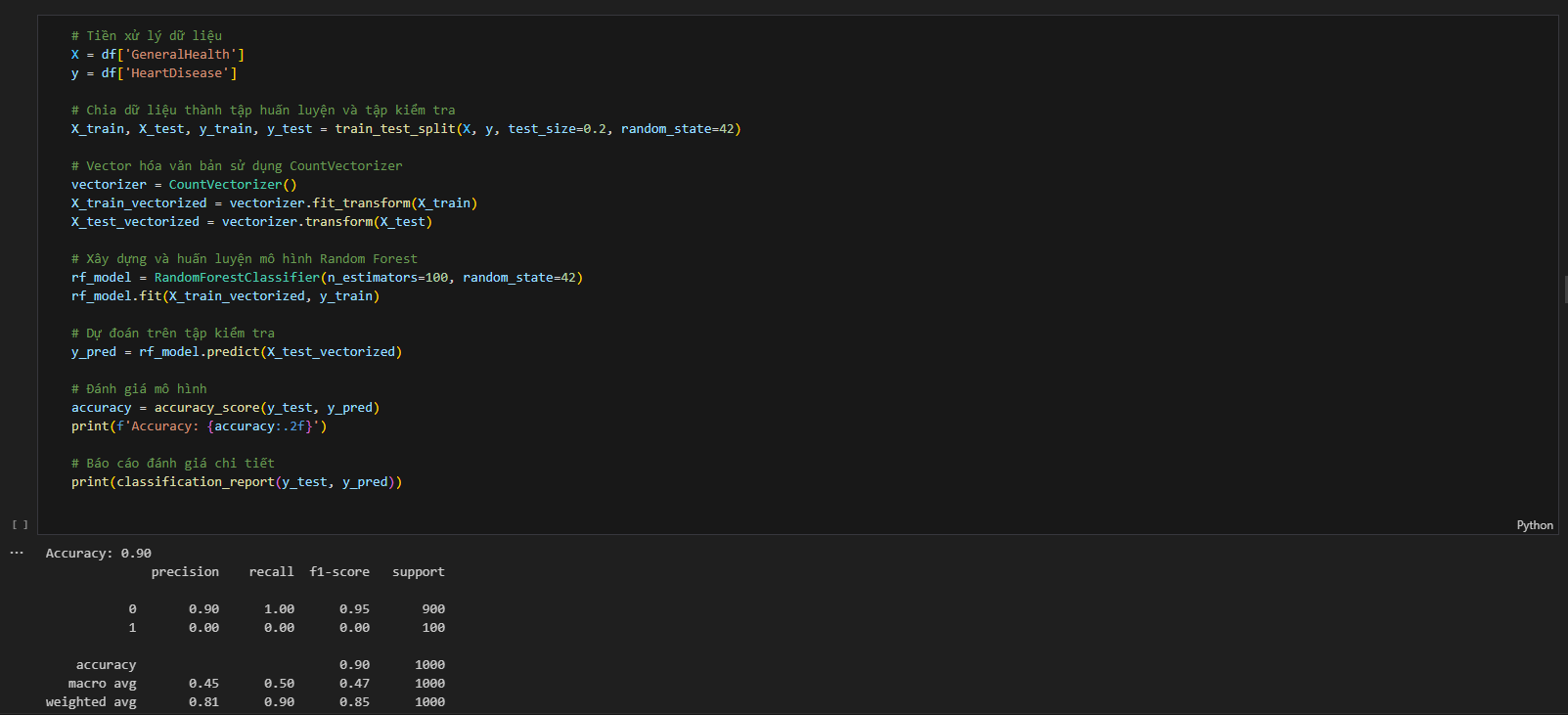
* Tạo hàm huấn luyện mô hình và đánh giá mô hình



Hình . Tạo hàm huấn luyện và đánh giá mô hình

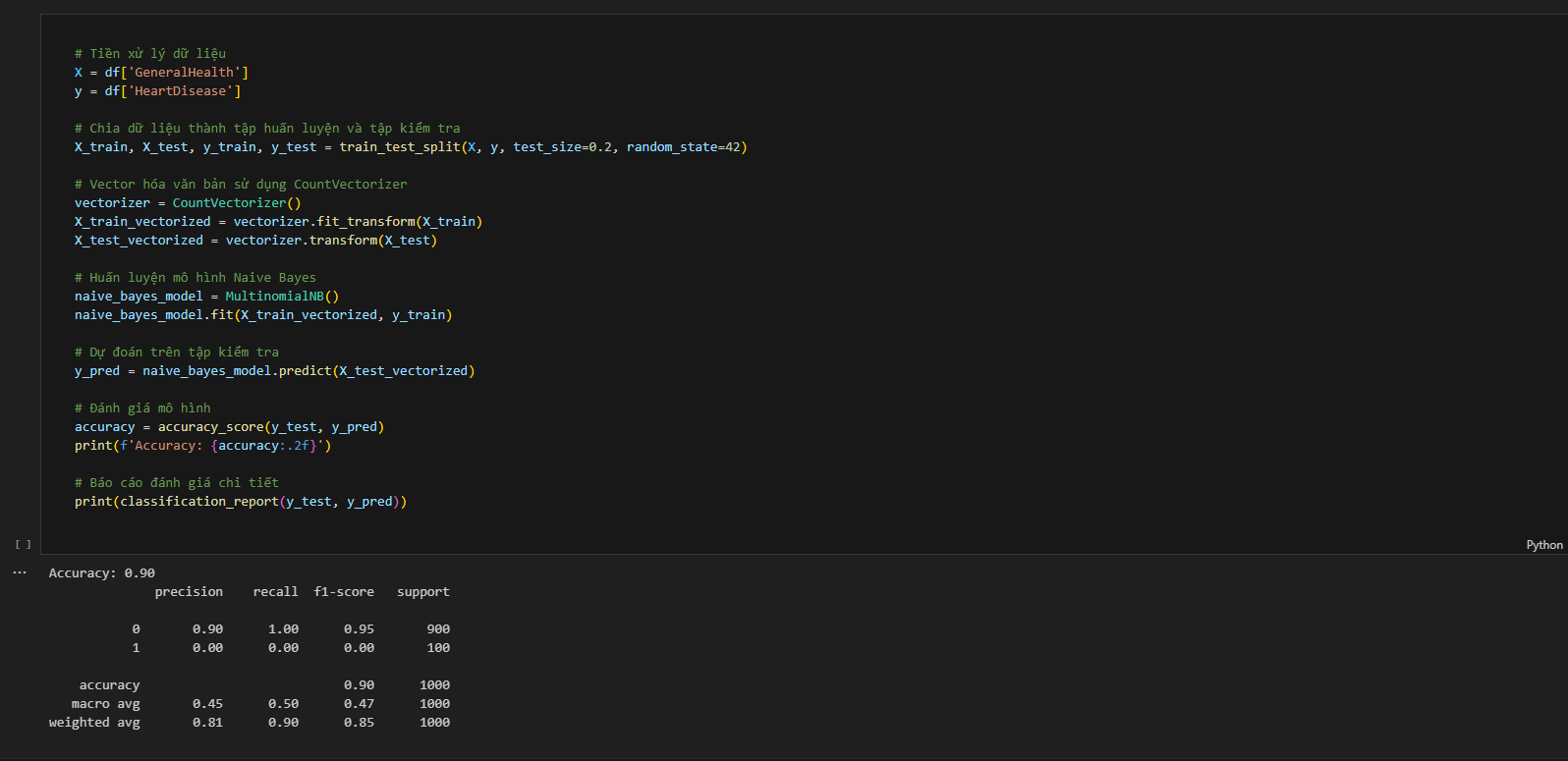
**2)**

- Sử dụng mô hình Random Forest để giải quyết bài toán



Hình . Sử dụng mô hình Random Forest

* Sử dụng mô hình Naïve Bayes để giải quyết bài toán :



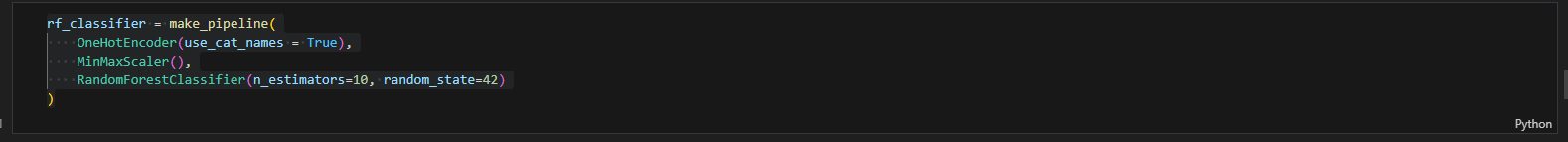
Hình . Sử dụng mô hình Naive Bayes

* Sử dụng mô hình Decision Tree để giải quyết bài toán



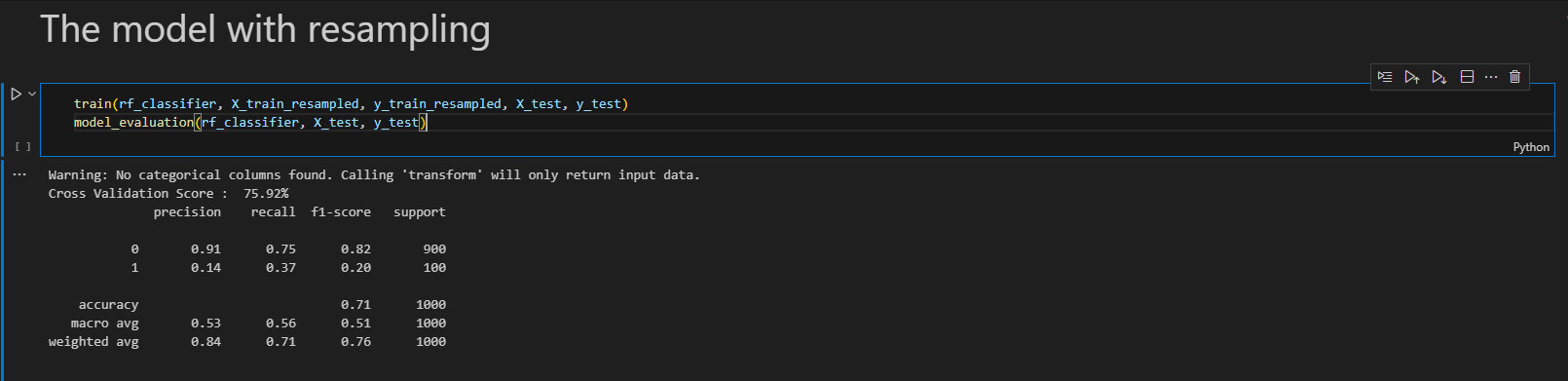
Hình . Sử dụng mô hình Decision Tree

* Định nghĩa một RandomForestClassifier thông qua một Pipeline trong scikit-learn



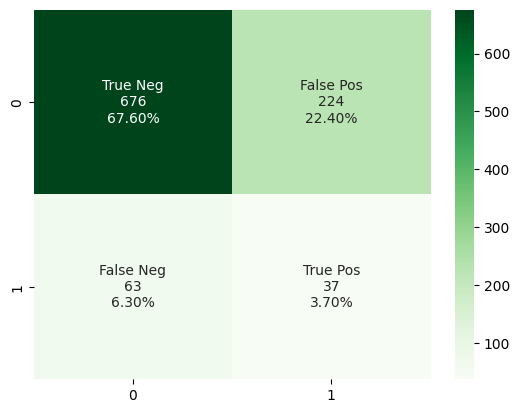
Hình . Định nghĩa một RandomForest

* Huấn luyện mô hình Random Forest trên tập dữ liệu đã được Resample



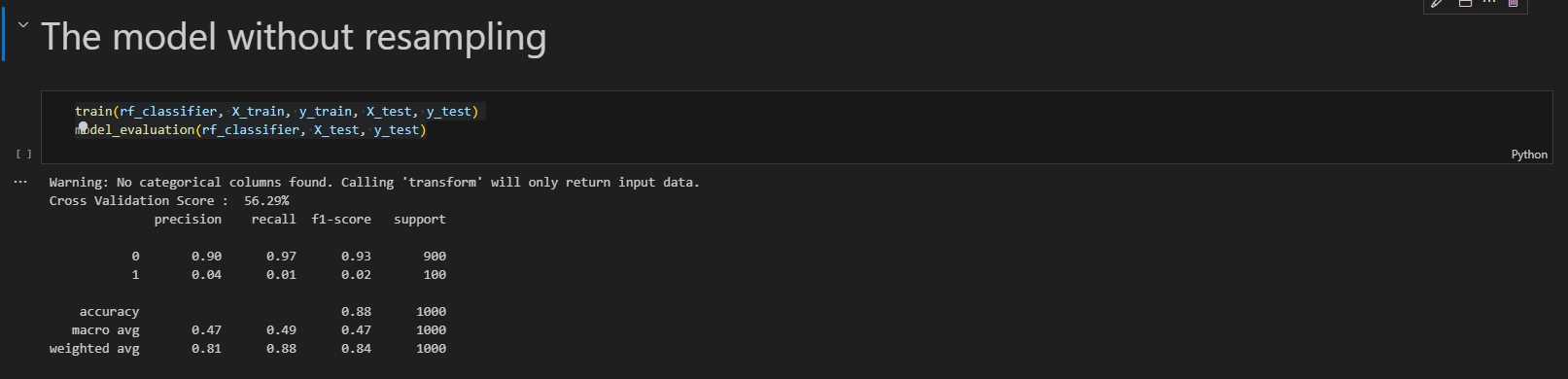
Hình . Huấn luyện Random Foreswst trên tập đã được Resample

* Kết quả :



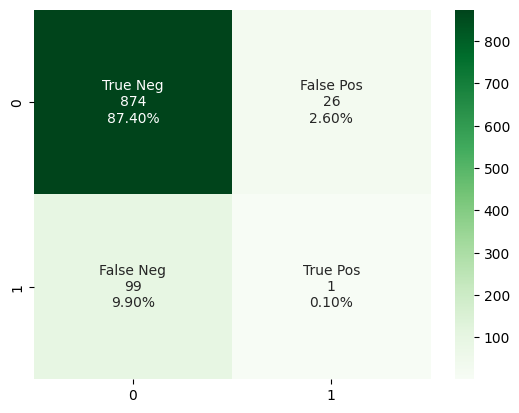
Hình . Kết quả trên tập đã được Resample

* Huấn luyện mô hình Random Forest trên tập dữ liệu gốc



Hình . Huấn luyện mô hình Random Forest trên tập dữ liệu gốc

* Kết quả :

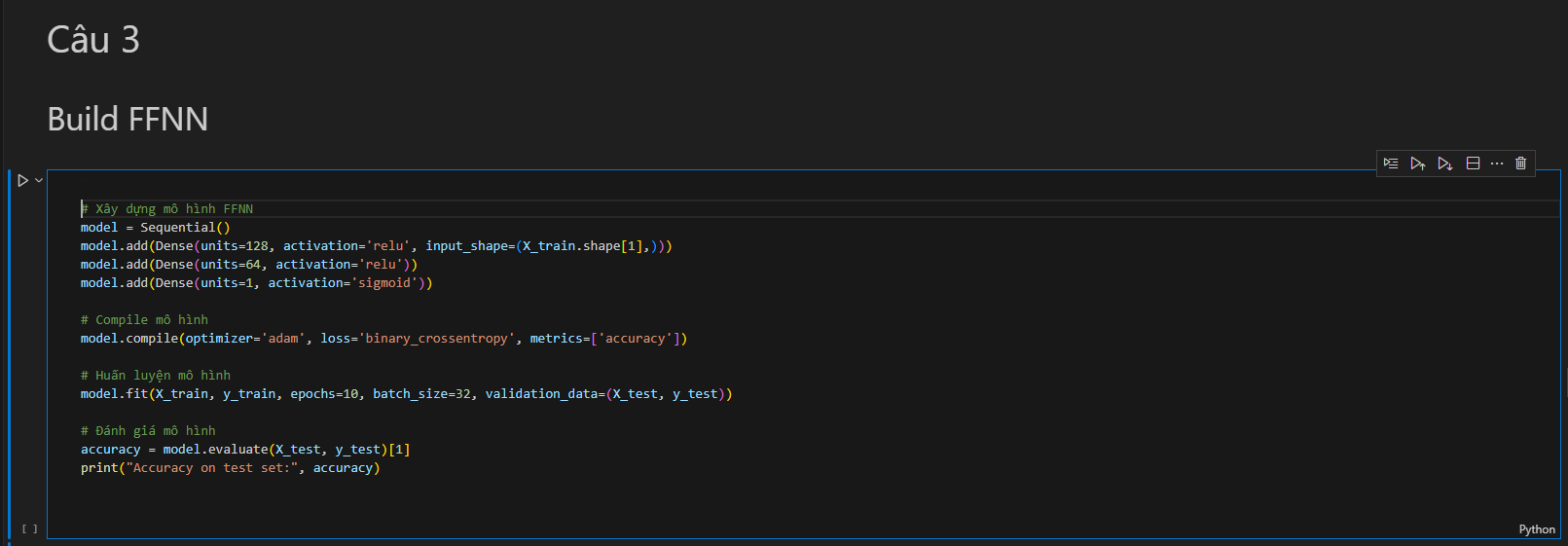


Hình . Kết quả trên tập dữ liệu gốc

Sự khác biệt giữa hai đoạn code này nằm ở việc sử dụng dữ liệu huấn luyện đã được resample hay không. Việc resample có thể được thực hiện để cải thiện khả năng dự đoán của mô hình trên dữ liệu mất cân bằng. Cụ thể, nó giúp giảm ảnh hưởng của lớp đa số (majority class) và tăng khả năng nhận biết lớp thiểu số (minority class).

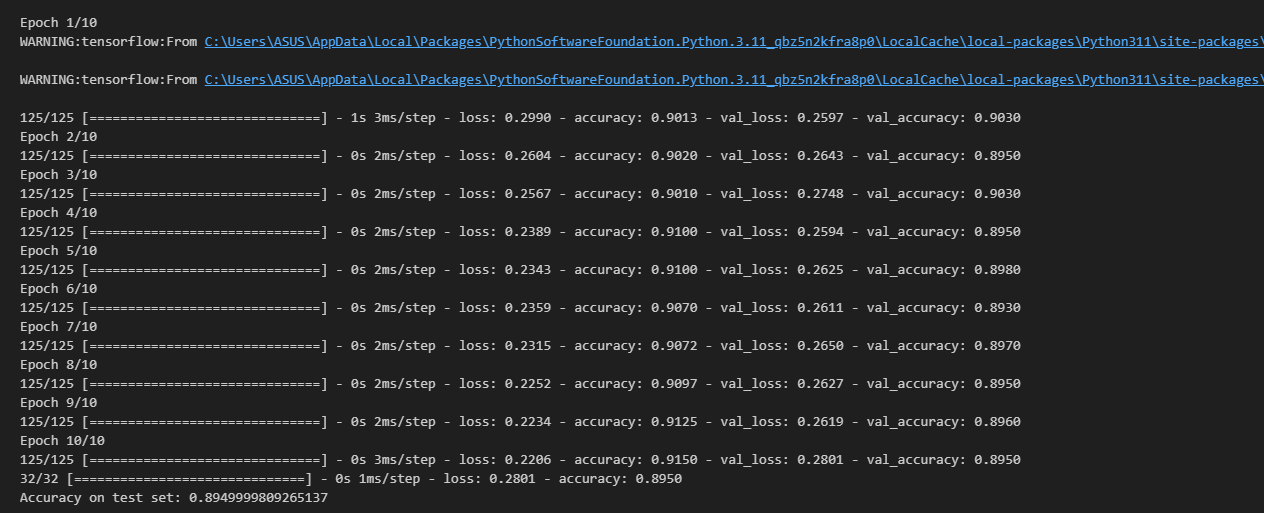
**3)**

- Sử dụng Feed Forward Neural Network để giải quyết bài toán



Hình . Sử dụng Feed Forward Neural Network để giải quyết bài toán

Kết quả:



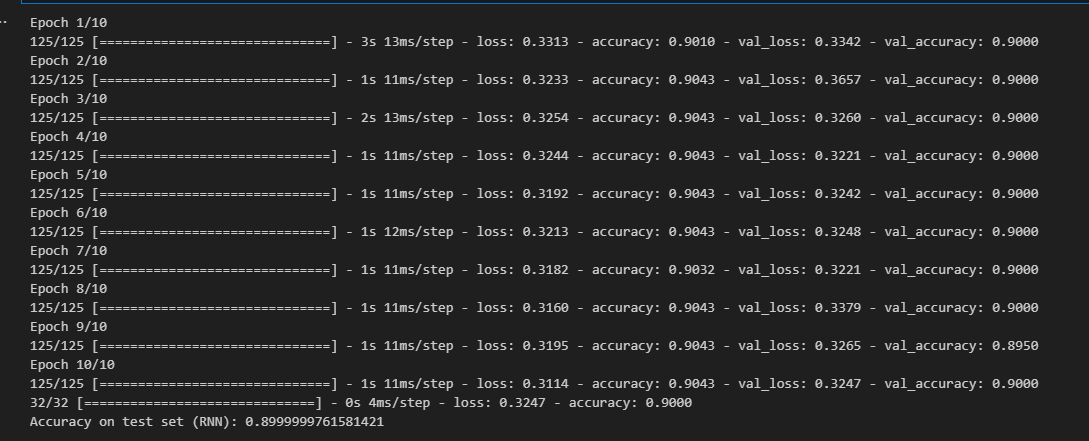
Hình . Kết quả khi sử dụng Feed Forward Neural Network để giải quyết bài toán

* Sử dụng Recurrent Neural Network để giải quyết bài toán



Hình . Sử dụng Recurrent Neural Network để giải quyết bài toán

Kết quả:



Hình . Kết quả khi sử dụng Feed Forward Neural Network để giải quyết bài toán

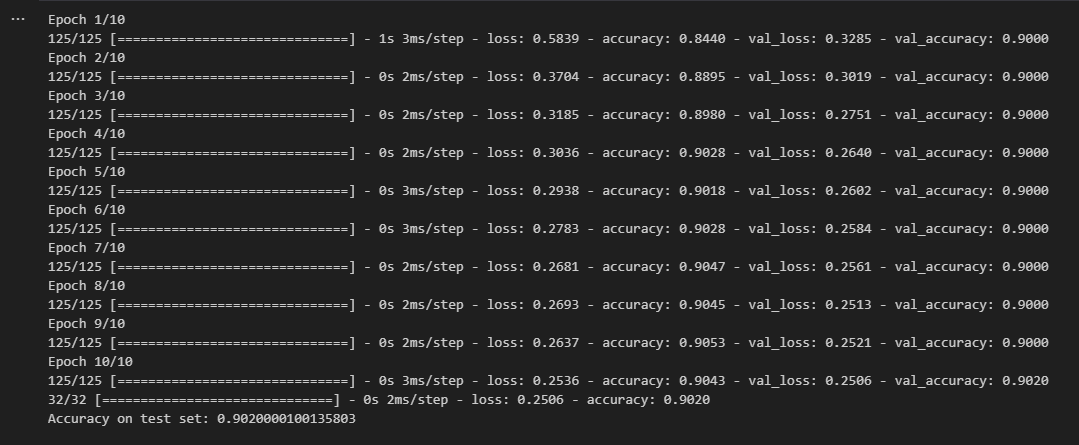
**4)**

* Tránh overfitting trên mô hình FFNN bằng cách thêm dropout layer



Hình . Tránh overfitting trên mô hình FFNN bằng cách thêm dropout layer

Kết quả :



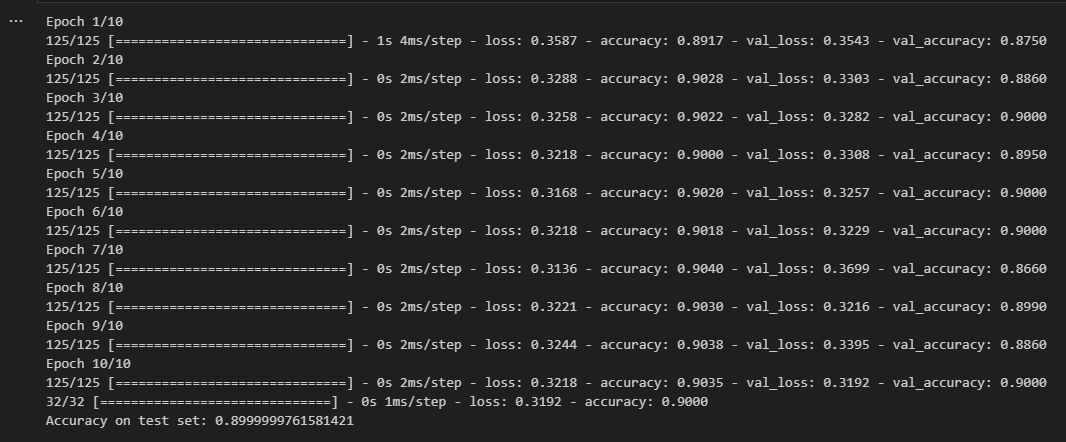
Hình . Tránh overfitting trên FFNN

* Tránh overfitting trên FFNN với Early Stopping



Hình . Tránh overfitting trên FFNN với Early Stopping

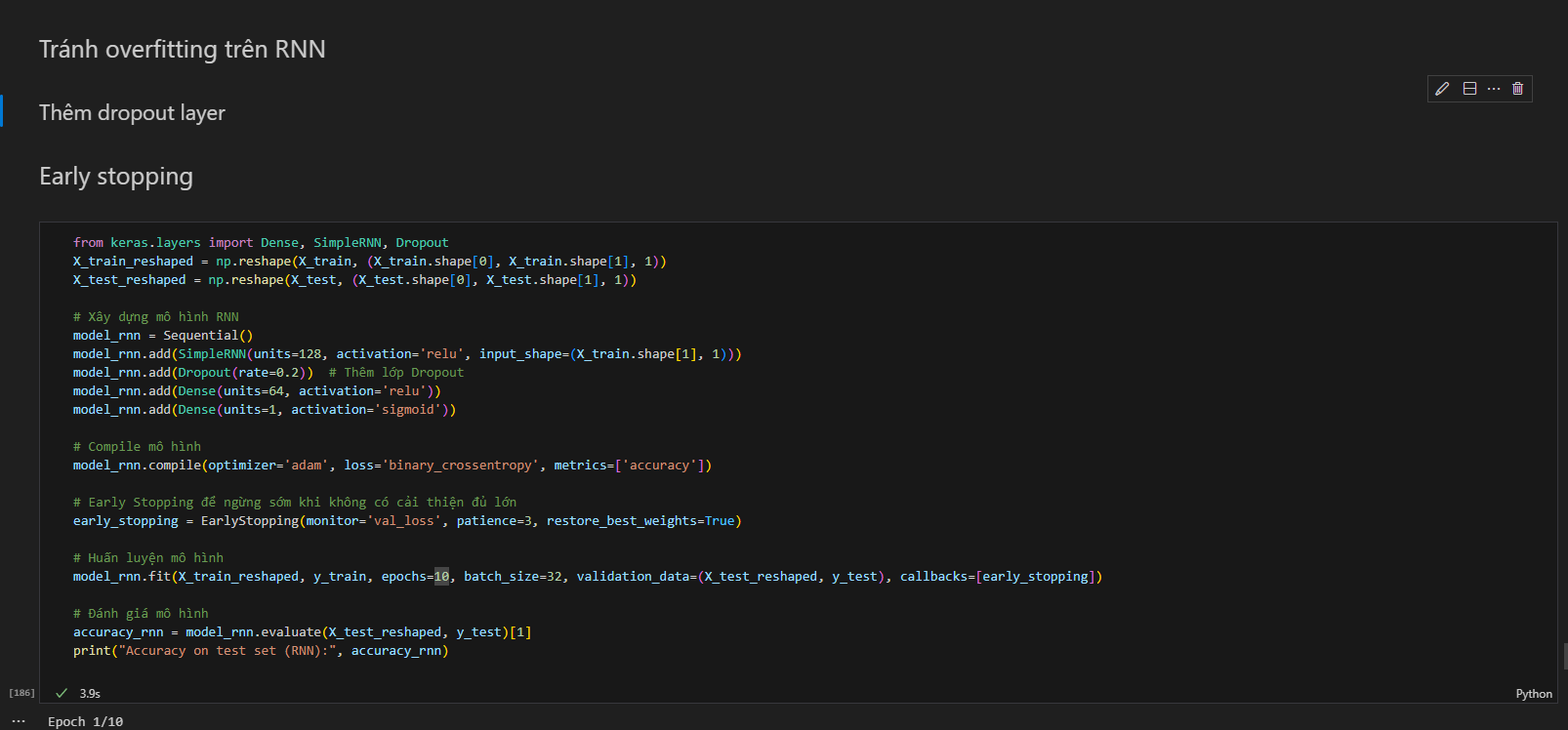
Kết quả :



Hình . Kết quả

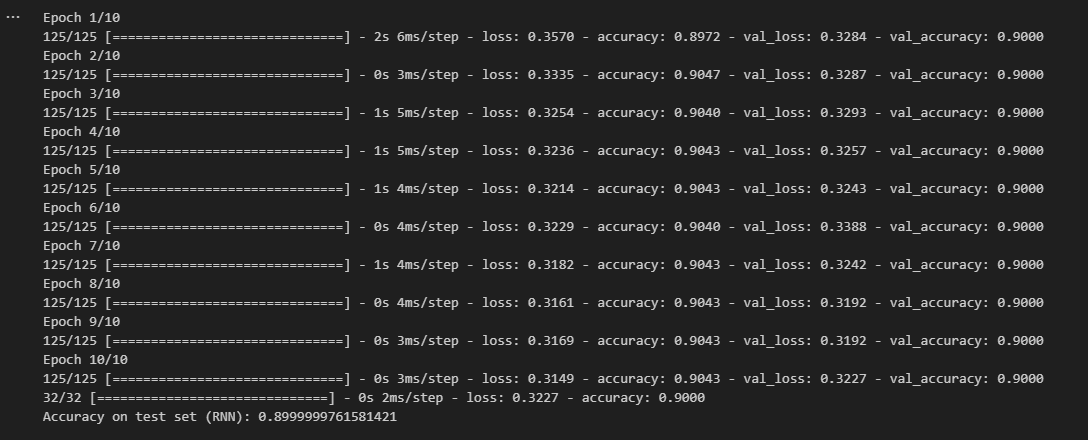
Ta có thể thấy, tỉ lệ đã nhỉnh hơn 1 chút sau khi áp dụng kỹ thuật tránh overfitting trên mô hình FFNN

* Tránh overfitting trên mô hình RNN bằng cách dùng cả 2 cách thêm Dropout Layer và Early Stopping



Hình . Tránh overfitting trên mô hình RNN

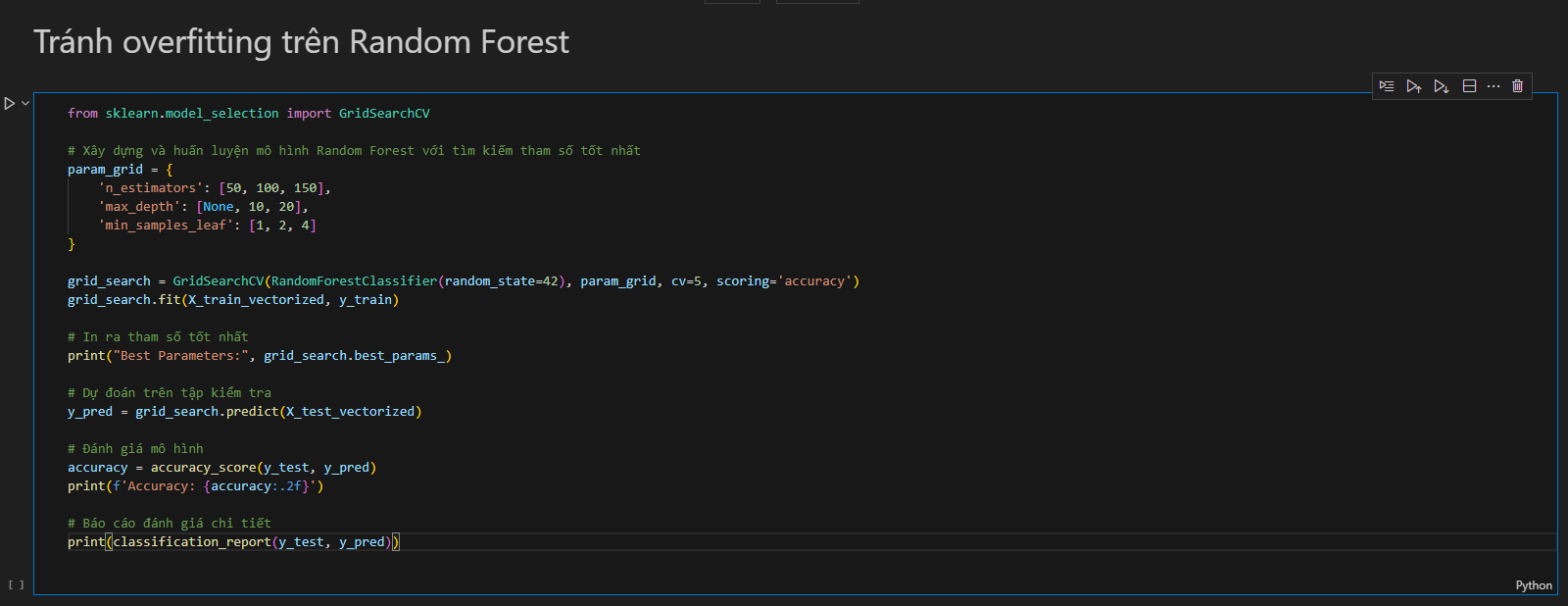
Kết quả :



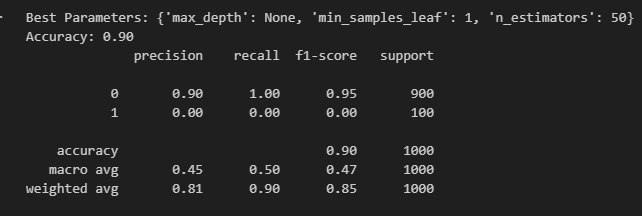
Hình . Kết quả

Ta có thể thấy, tỉ lệ đã nhỉnh hơn 1 chút sau khi áp dụng kỹ thuật tránh overfitting trên mô hình RNN

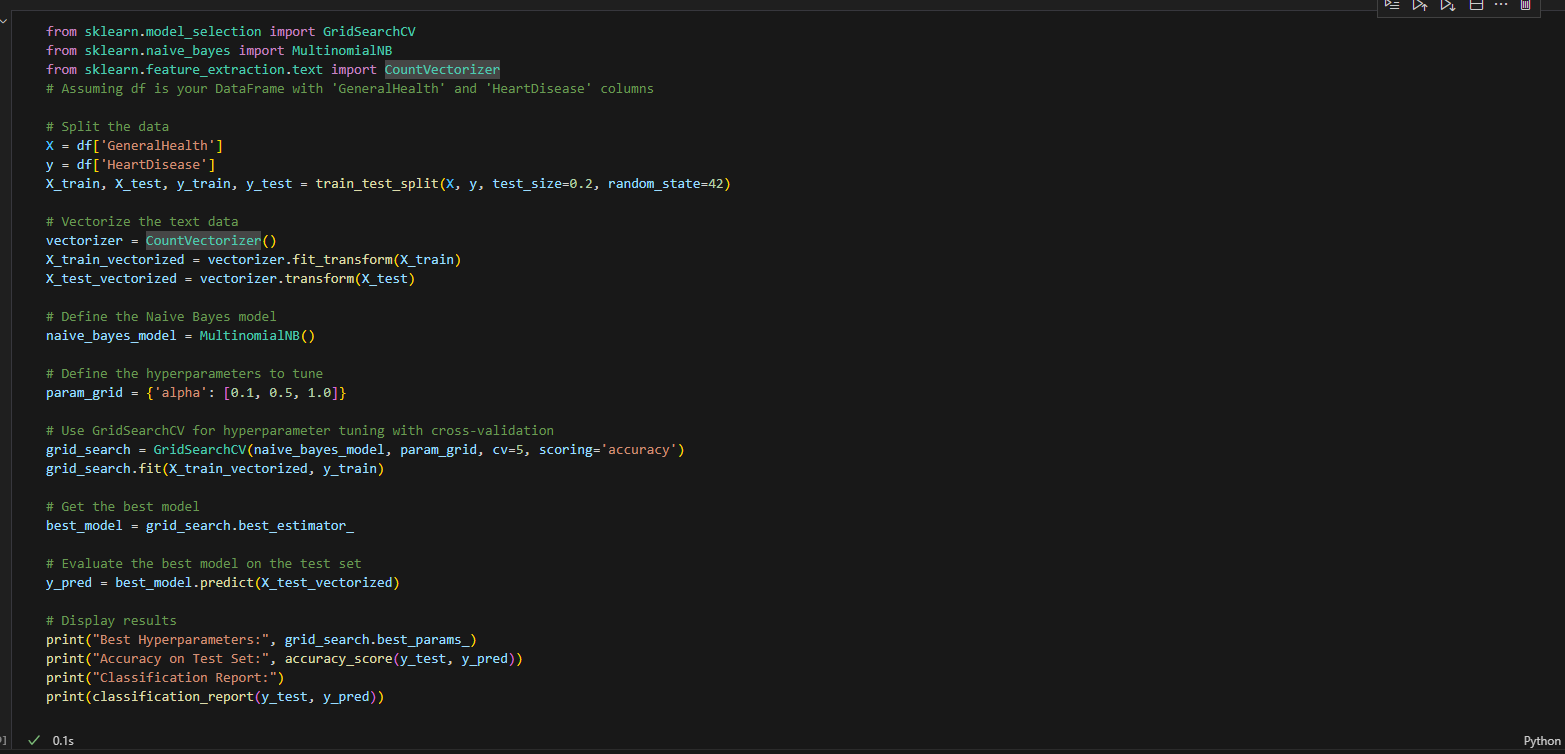
* Tránh overfitting trên mô hình RandomForest



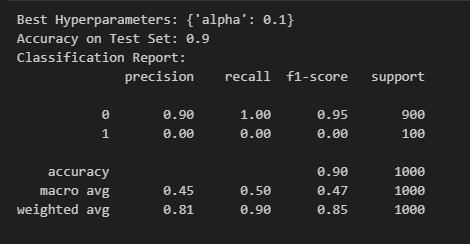
Hình . Tránh overfitting trên mô hình RandomForest

Kết quả : 

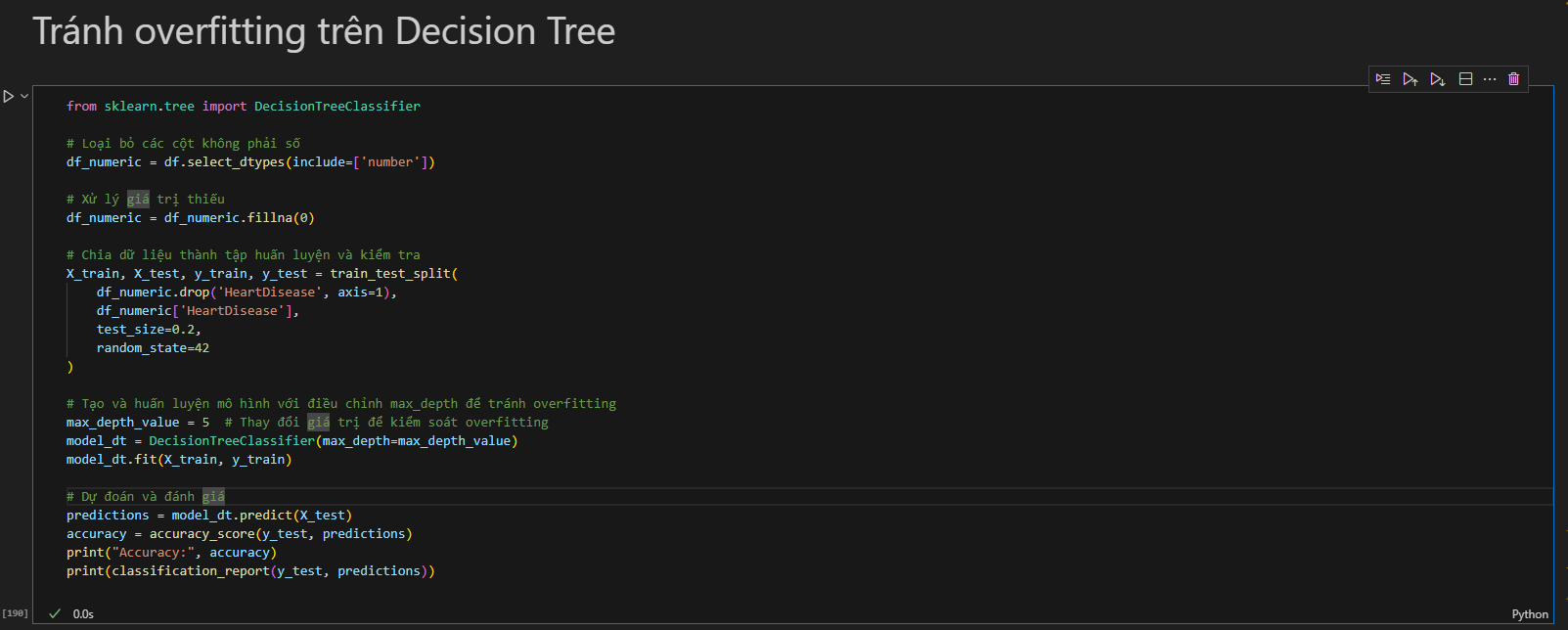
* Tránh overfitting trên Naïve Bayes



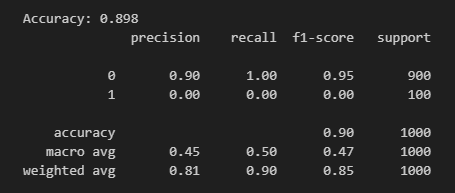
Hình . Tránh overfitting trên Naïve Bayes

Kết quả : 

* Tránh overfitting trên Decision Tree



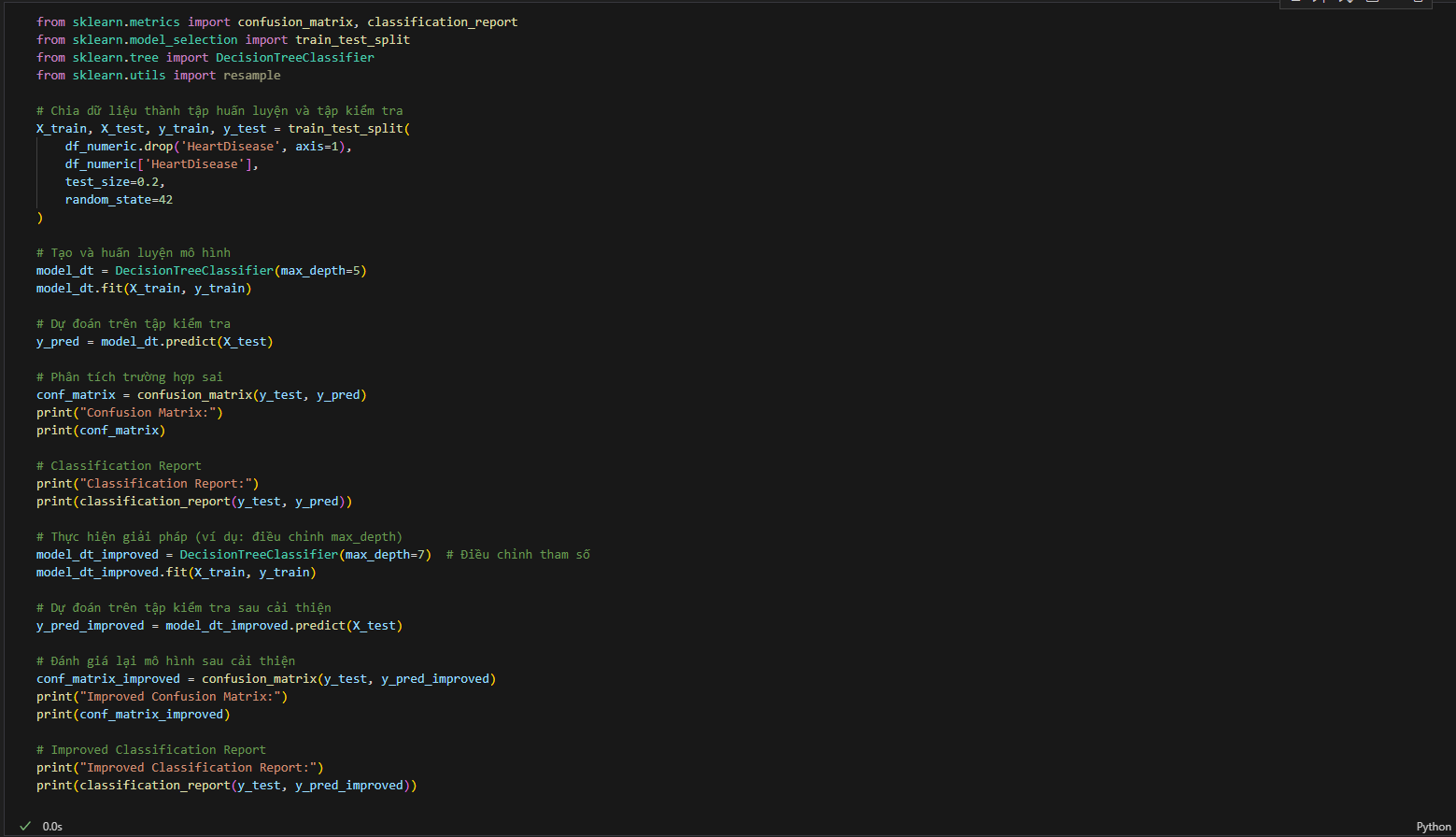
Hình . Tránh overfitting trên Decision Tree

Kết quả : 

**5)**

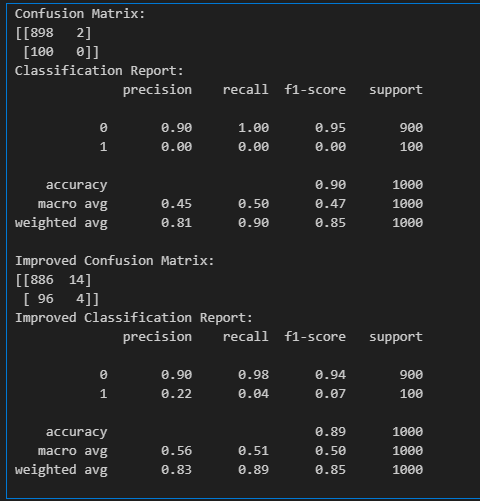
Để cải thiện độ chính xác của mô hình và giải quyết vấn đề trường hợp sai, bạn có thể thực hiện các bước sau:

* Phân tích Trường hợp Sai (False Cases Analysis):
  + Lấy các dự đoán của mô hình trên tập kiểm tra.
  + So sánh dự đoán với nhãn thực tế để xác định các trường hợp sai.
  + Phân tích các trường hợp False Positive (FP) và False Negative (FN) để hiểu tại sao mô hình đưa ra các quyết định không chính xác.
  + Thực hiện Giải pháp:
  + Dựa trên phân tích trường hợp sai, xem xét các biện pháp cụ thể để cải thiện mô hình. Các biện pháp này có thể bao gồm:
  + Tối ưu hóa tham số mô hình: Điều chỉnh các tham số của mô hình (ví dụ: max\_depth, learning\_rate, n\_estimators).
  + Thêm dữ liệu: Nếu có thêm dữ liệu huấn luyện, có thể giúp mô hình học tốt hơn.
  + Xử lý mất cân bằng dữ liệu: Nếu dữ liệu mất cân bằng, sử dụng kỹ thuật resampling hoặc weighting để cải thiện hiệu suất trên lớp thiểu số.
  + Sử dụng mô hình phức tạp hơn hoặc đơn giản hơn: Tùy thuộc vào bản chất của dữ liệu, bạn có thể thử nghiệm các mô hình phức tạp hơn hoặc đơn giản hơn.
* Huấn luyện lại và Đánh giá:
  + Thực hiện lại quá trình huấn luyện với các biện pháp được thực hiện.
  + Đánh giá lại mô hình trên tập kiểm tra để đảm bảo rằng cải thiện đã được đạt được.
* Dưới đây là một đoạn mã Python đơn giản để thực hiện các bước trên:



Hình . Đoạn mã Python

Kết quả :



Ban đầu :

* Confusion Matrix :

[[898 2]

[100 0]]

* Classification Report :

precision recall f1-score support

0 0.90 1.00 0.95 900

1 0.00 0.00 0.00 100

accuracy 0.90 1000

macro avg 0.45 0.50 0.47 1000

weighted avg 0.81 0.90 0.85 1000

Ở đây:

* Accuracy (độ chính xác) là 0.90, tức là mô hình đúng 90% trên tập kiểm tra.
* Precision (độ chính xác của dự đoán positive) cho lớp 0 là 0.90, nhưng cho lớp 1 là 0.00, tức là mô hình không dự đoán đúng positive cho lớp 1.
* Recall (độ chính xác của dự đoán true positive) cho lớp 0 là 1.00, nhưng cho lớp 1 là 0.00, tức là mô hình không đánh giá tốt về true positive cho lớp 1.
* F1-score là một trung bình điều hòa giữa precision và recall.

Sau khi cải thiện :

* Confusion Matrix:

[[886 14]

[ 96 4]]

* Classification Report:

precision recall f1-score support

0 0.90 0.98 0.94 900

1 0.22 0.04 0.07 100

accuracy 0.89 1000

macro avg 0.56 0.51 0.50 1000

weighted avg 0.83 0.89 0.85 1000

Ở đây:

- Accuracy giảm nhẹ xuống 0.89 sau khi cải thiện.

- Precision cho lớp 0 vẫn cao là 0.90, nhưng recall của lớp 1 tăng lên 0.04 so với 0.00, tức là mô hình đang nhận ra một số trường hợp positive cho lớp 1.

- F1-score có sự cải thiện đối với lớp 1, nhưng vẫn thấp do số lượng true positive là ít.

Tóm lại, sau khi cải thiện, mô hình có vẻ nhận ra một số trường hợp positive cho lớp 1 hơn, nhưng vẫn còn nhiều cơ hội để cải thiện độ chính xác và đánh giá chính xác hơn trên cả hai lớp.

TÀI LIỆU THAM KHẢO

[1]. Các phương pháp Optimizer

<https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8>