# Tài Liệu Hướng Dẫn Huấn Luyện Mô Hình Dự Đoán Bệnh Tim

# Heart Disease Documents

## I. Giới thiệu

Tài liệu này hướng dẫn chi tiết quá trình huấn luyện mô hình dự đoán bệnh tim (Heart Disease Prediction Model) dựa trên bộ dữ liệu 'heart.csv'. Mục tiêu của mô hình là phân loại bệnh nhân có nguy cơ mắc bệnh tim dựa trên các thông số lâm sàng như tuổi, huyết áp, cholesterol, nhịp tim, và các đặc điểm khác.

## II. Môi trường và thư viện cần thiết

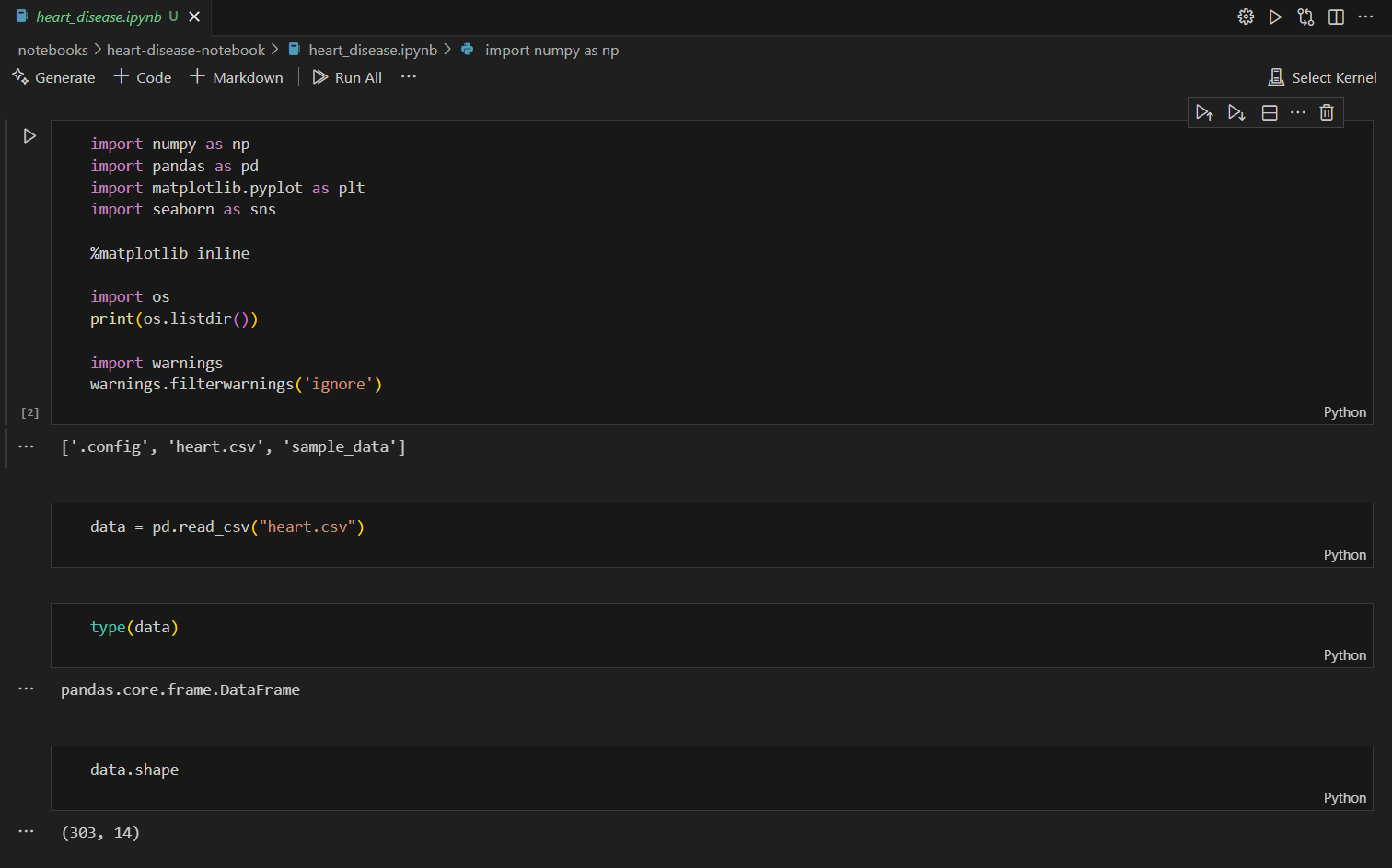
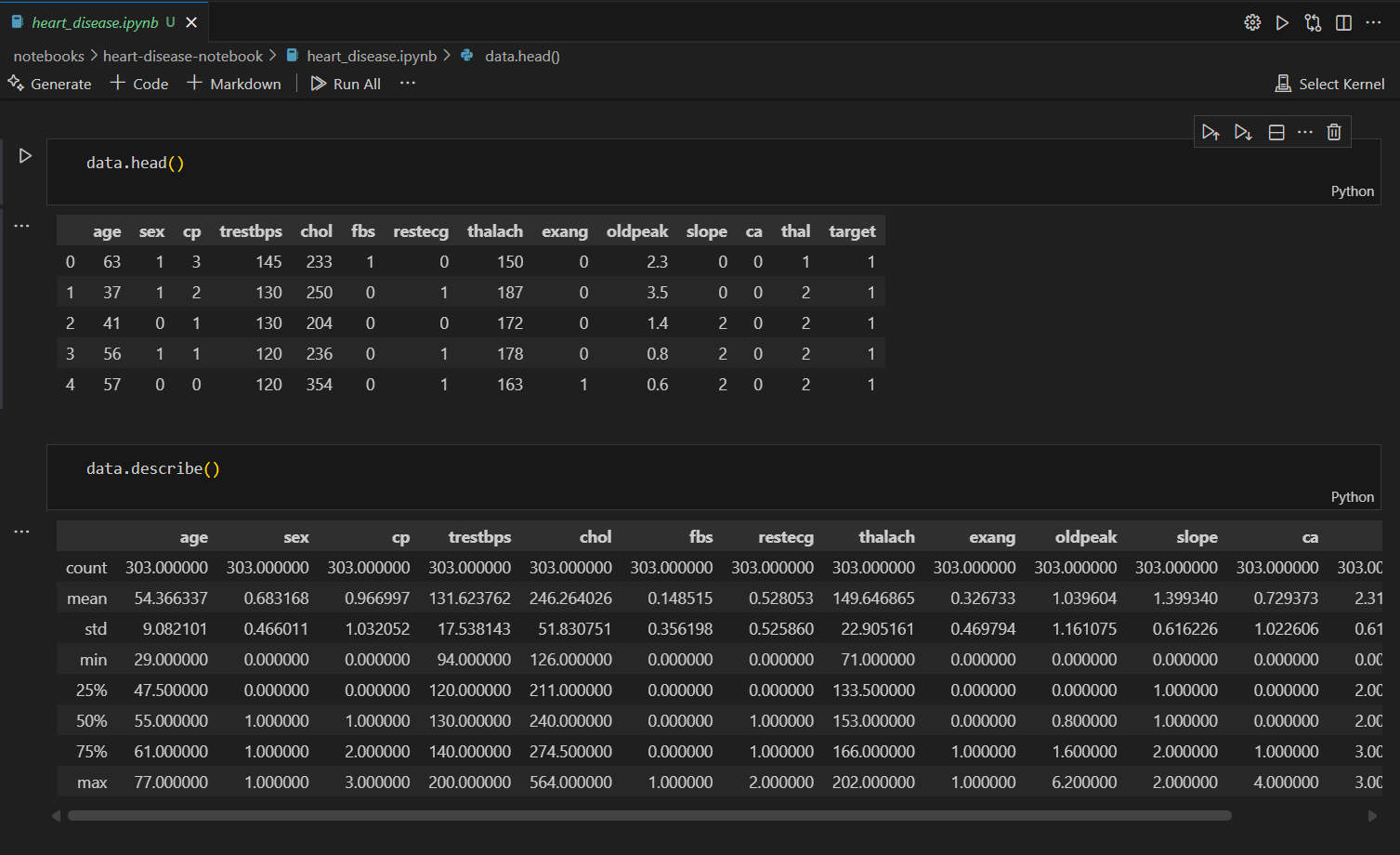
Để huấn luyện mô hình, cần cài đặt Python và các thư viện sau:

* - numpy
* - pandas
* - matplotlib
* - seaborn
* - scikit-learn

Cài đặt nhanh bằng lệnh: ***pip install requirements.txt***

## III. Chuẩn bị dữ liệu

1. Tải file dữ liệu 'heart.csv' và đặt trong cùng thư mục với notebook.  
2. Nạp dữ liệu bằng lệnh:

1. age: The person's age in years

2. sex: The person's sex (1 = male, 0 = female)

3. cp: The chest pain experienced (Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic)

4. trestbps: The person's resting blood pressure (mm Hg on admission to the hospital)

5. chol: The person's cholesterol measurement in mg/dl

6. fbs: The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)

7. restecg: Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)

8. thalach: The person's maximum heart rate achieved

9. exang: Exercise induced angina (1 = yes; 0 = no)

10. oldpeak: ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot. See more here)

11. slope: the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)

12. ca: The number of major vessels (0-3)

13. thal: A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

14. target: Heart disease (0 = no, 1 = yes)

Heart disease risk factors to the following: high cholesterol, high blood pressure, diabetes, weight, family history and smoking .

According to another source , the major factors that can't be changed are: increasing age, male gender and heredity.

Note that thalassemia, one of the variables in this dataset, is heredity.

Major factors that can be modified are: Smoking, high cholesterol, high blood pressure, physical inactivity, and being overweight and having diabetes.

Other factors include stress, alcohol and poor diet/nutrition.

import pandas as pd

data = pd.read\_csv("heart.csv")

data.head()

3. Kiểm tra thông tin dữ liệu, kích thước và giá trị bị thiếu:  
 - data.info()  
 - data.describe()  
 - data.isnull().sum()

## IV. Tiền xử lý dữ liệu

Dữ liệu trong tập 'heart.csv' đã được làm sạch (không có giá trị bị thiếu). Tuy nhiên, cần kiểm tra tương quan giữa các biến với cột mục tiêu 'target' để xác định mức độ ảnh hưởng.

***data.corr()["target"].abs().sort\_values(ascending=False)***

## V. Chia tập huấn luyện và kiểm thử

Sử dụng train\_test\_split để chia dữ liệu thành hai phần: 80% train và 20% test.

from sklearn.model\_selection import train\_test\_split

X = data.drop("target", axis=1)

y = data["target"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

## VI. Huấn luyện mô hình

Huấn luyện và so sánh nhiều mô hình khác nhau để chọn ra mô hình tốt nhất:

* - Logistic Regression
* - Naive Bayes
* - Support Vector Machine (SVM)
* - K-Nearest Neighbors (KNN)
* - Decision Tree

Huấn luyện Logistic Regression và đánh giá độ chính xác:

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

lr = LogisticRegression()

lr.fit(X\_train, y\_train)

y\_pred = lr.predict(X\_test)

acc = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {acc\*100:.2f}%")

## VII. Đánh giá mô hình

Sau khi huấn luyện, so sánh độ chính xác giữa các mô hình bằng chỉ số Accuracy. Mô hình có Accuracy cao nhất sẽ được chọn làm mô hình chính thức.

Ví dụ:  
- Logistic Regression: 85.0%  
- Naive Bayes: 83.0%  
- SVM: 86.0%  
- KNN: 84.0%  
- Decision Tree: 88.0%  
→ Chọn Decision Tree làm mô hình tốt nhất.

## VIII. Lưu và tải mô hình

Sau khi huấn luyện xong, lưu mô hình để sử dụng lại mà không cần huấn luyện lại từ đầu:

import joblib

joblib.dump(dt, "heart\_disease\_model.pkl")

# Tải lại mô hình

model = joblib.load("heart\_disease\_model.pkl")

## IX. Hướng dẫn retrain mô hình

Để huấn luyện lại mô hình trên dữ liệu mới:  
1. Mở Jupyter Notebook hoặc môi trường tương tự.  
2. Tải lại dữ liệu (heart.csv hoặc tập dữ liệu mới).  
3. Lặp lại các bước tiền xử lý và chia train/test.  
4. Gọi lại đoạn code huấn luyện mô hình.  
5. Lưu mô hình mới bằng joblib.dump().  
6. So sánh độ chính xác mới với mô hình cũ để đánh giá cải tiến.

## X. Tóm tắt

Độ chính xác đạt được tốt nhất.