AI VIET NAM – COURSE 2023

M04 - Final Project

(Decision Tree and Its Variances)

Ngày 16 tháng 9 năm 2023

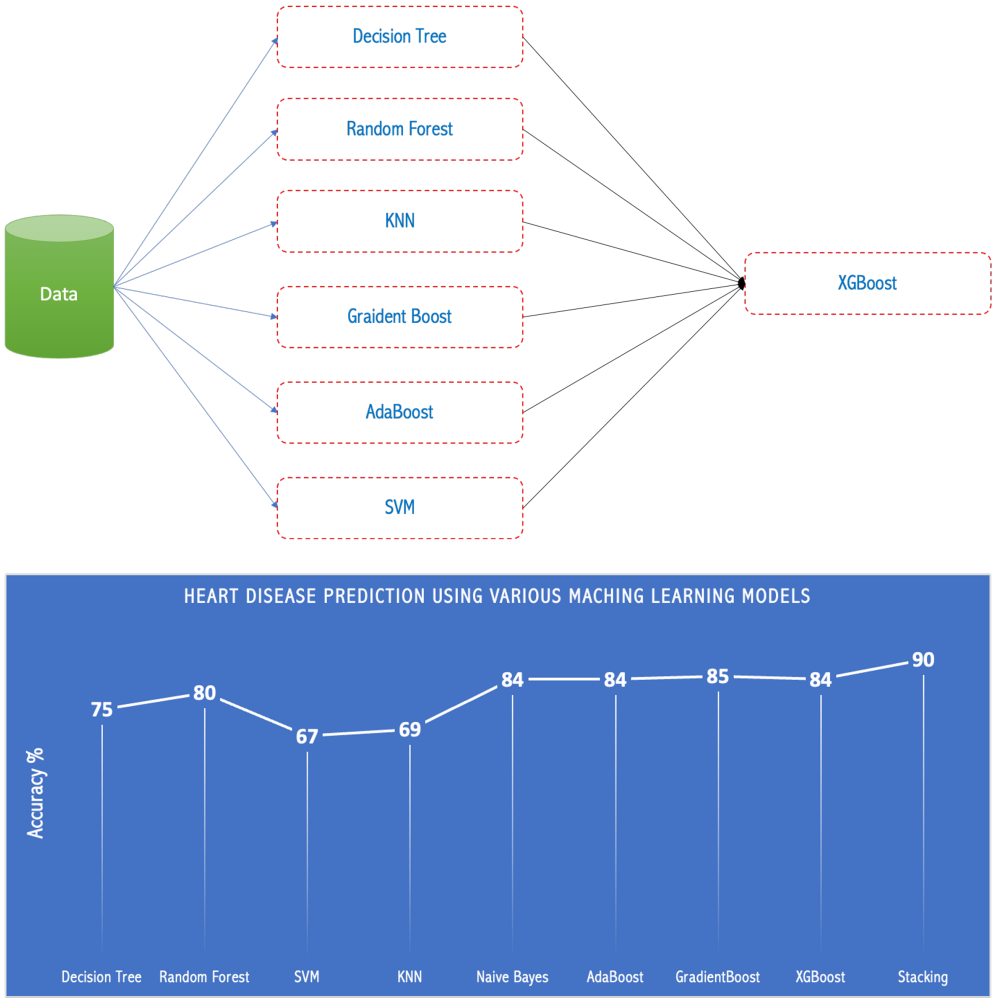
Giới thiệu về project dự đoán khả năng bị bệnh tim của bệnh nhân :

Ngày nay, số lượng các bệnh lý liên quan đến tim (heart disease) ngày càng gia tăng không phân biệt về giới tính hay lứa tuổi. Theo số liệu của WHO, bệnh tim là nguyên nhân gây tử vong lớn nhất thế giới. Trong đó, bệnh tim thiếu máu cục bộ chiếm 16% và đột quỵ chiếm 11% số ca tử vong trên toàn cầu. Kể từ năm 2000, số ca tử vong do căn bệnh này gia tăng nhiều nhất, tăng hơn 2 triệu đến 8,9 triệu ca tử vong vào năm 2019. Các bệnh lý tim thường gặp bào gồm các bệnh liên quan mạch máu (blood vessel diseases) như là bênh động mạch vành (coronary artery disease), các vấn đề loạn nhịp tim (arrhythmias) và dị tật tim bẩm sinh (ongenital heart defects), cùng nhiều bệnh lý khác.

Vì vậy việc dự đoán sớm bệnh tim mạch được coi là một trong những vấn đề quan trọng trong phân tích dữ liệu lâm sàn (clinical data analysis). Nhưng rất khó để xác định chính xác cũng như dự đoán sớm triệu chứng của bệnh tim vì có nhiều yếu tố nhứ tiểu đường, huyết áp cao, cholesterol cao, nhịp tim bất thường và nhiều yếu tố khác. Ngày nay, số lượng dữ liệu trong ngành chăm sóc sức khỏe là rất lớn (big data). Do đó, việc khai thác dữ liệu (data mining) và trích xuất thông tin từ bộ dữ liệu lớn này là cần thiết để hỗ trợ giúp chẩn đoán và phòng ngừa sớm các biến chứng liên quan tim mạch có thể xảy ra. Vì vậy, các nhà khoa học bắt đầu nghiên cứu các phương pháp hiện đại như Khai thác dữ liệu (data mining) và Học máy (machine learing) để dự đoán sớm khả năng bị bệnh tim dựa vào tiền sử sức khoẻ của bệnh nhân.

Trong project này, chúng ta sẽ áp dụng các phương pháp máy học cơ bản để dự đoán xem một người có khả năng mắc bệnh tim hay không dựa trên tập dữ liệu về Bệnh tim Cleveland dataset từ UCI Machine Learning Repository . Tập dữ liệu Cleveland bao gồm 14 thông tin như sau: tuổi (Age), giới tính (sex), trạng thái đau ngực (Chest-pain type), huyết áp khi nghỉ ngơi (Resting Blood Pressure), nồng độ cholesterol trong huyết thanh (Serum Cholestrol), chỉ số đường nhanh trong máu (Fasting Blood Sugar), kết quả điện tâm đồ khi nghỉ ngơi (Resting ECG), nhịp tim tối đa (Max heart rate achieved), có bị đau thắt ngực khi tập thể dụng hay không (Exercise induced angina), chỉ số ST lúc tập thể thao so với lúc thư giãn (ST depression induced by exercise relative to res), chỉ số ST trong lúc hoạt động gắng sức (Peak exercise ST segment), Số lượng mạch chính (gồm động mạch, mao mạch và tĩnh mạch) được phát sáng thông qua nội soi huỳnh quang (Number of major vessels (0–3) colored by flourosopy), thiếu máu tán huyết bẩm sinh (displays the thalassemia), và thông tin có bị biên tim hay không (Diagnosis of heart disease, 0 đại điện cho bệnh nhân không có bệnh, và 1,2,3,4 đại diện cho bệnh nhân có bệnh). Tập dữ liệu Cleveland bao gồm 303 mẫu với 14 thông tin trên được thể hiện thông qua hình 2.

Trong project này chúng ta sẽ sử dụng các giải thuật máy học khác nhau để dự đoán xem bệnh nhân có khả năng bị bệnh tim hay không. Để hoàn thành được project này, AIVN thừa nhận rằng readers đã nắm vững và biết cách sử dụng thư viện sklearn để hiện thực các giải thuật máy học thông dụng cho bài toán classification như: **naive bayes, k nearest neigbors (KNN), decision tree, random forest, Adaboost, gradient boost, XGBoost và support vector machine (SVM)**. Cũng như hiểu rõ mô hình máy học tích hợp ensemple theo bagging, boosting và stack. Hình 1 thể hiện mô hình



Hình 1: Độ chính xác của các giải thuật máy học trên tập dữ liệu Cleveland

Age	Sex	CP	Restbps	Chol	Fbs	restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	Target
63	**1**	**1**	**145**	**233**	**1**	**2**	**150**	**0**	**2.3**	**3**	**0**	**6**	**0**
67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
41	0	2	130	204	0	2	172	0	1.4	1	0	3	0
56	1	2	120	236	0	0	178	0	0.8	1	0	3	0
62	0	4	140	268	0	2	160	0	3.6	3	2	3	3
57	0	4	120	354	0	0	163	1	0.6	1	0	3	0
63	1	4	130	254	0	2	147	0	1.4	2	1	7	2
53	1	4	140	203	1	2	155	1	3.1	3	0	7	1
57	1	4	140	192	0	0	148	0	0.4	2	0	6	0
56	0	2	140	294	0	2	153	0	1.3	2	0	3	0
56	1	3	130	256	1	2	142	1	0.6	2	1	6	2
44	1	2	120	263	0	0	173	0	0	1	0	7	0

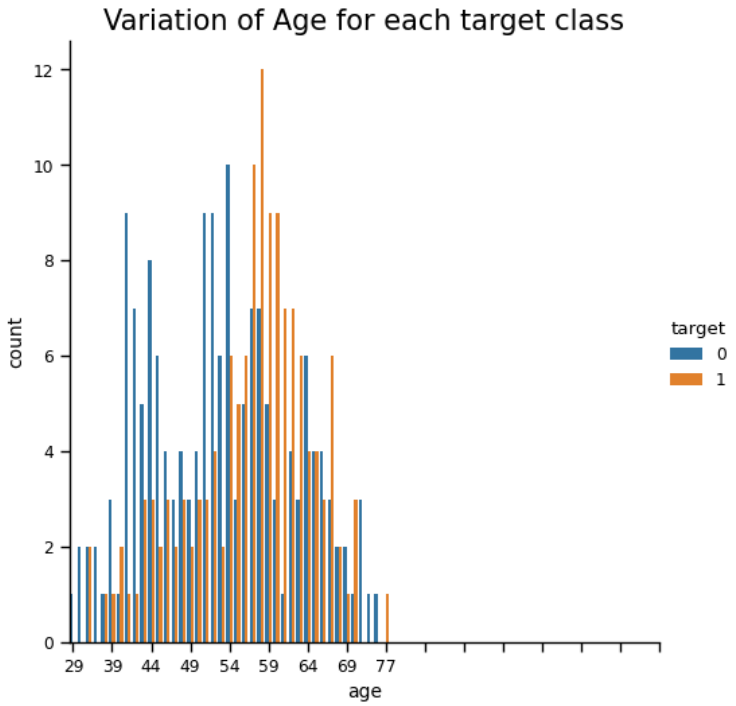
Hình 2: Một vài sample data từ tập dữ liệu y khoa Cleveland

huấn luyện theo phương pháp stacking và độ chính xác của các giải thuật máy học trên tập dữ liệu Cleveland.

Bài tập 1: (Data Analysis) hãy hoàn thiện chương trình sau sử dụng thư viện seaborn để hiện thị mối

quan hệ giữa độ tuổi và khả năng bị bệnh tim. Ở đây, target = 1 ngụ ý rằng người đó đang bị bệnh tim và target = 0 ngụ ý rằng người đó không bị bệnh tim. Hình 3 thể hiện kết quả sau khi thực hiện đoạn code visualization bên dưới:

```

import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import matplotlib.pyplot as plt
import seaborn as sns

# Bai tap 1
df = pd.read_csv('cleveland.csv', header = None)
df.columns = ['age', 'sex', 'cp', 'trestbps', 'chol',
              'fbs', 'restecg', 'thalach', 'exang',
              'oldpeak', 'slope', 'ca', 'thal', 'target']
df['target'] = df.target.map({0: 0, 1: 1, 2: 1, 3: 1, 4: 1})
df['thal'] = df.thal.fillna(df.thal.mean())
df['ca'] = df.ca.fillna(df.ca.mean())

# distribution of target vs age

# Your code here ******************************

plt.show()
```

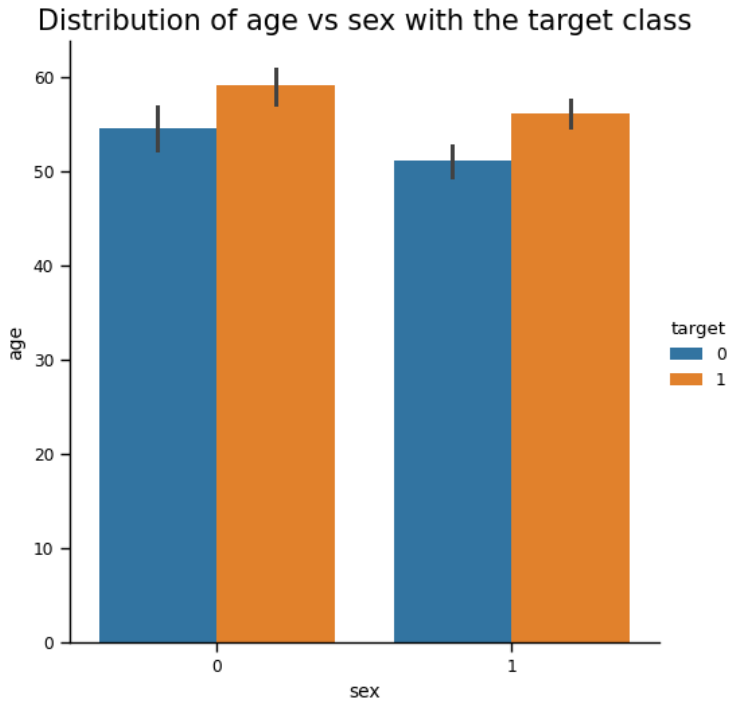


Hình 3: Đồ thị thể hiện mỗi quan hệ giữa độ tuổi và khả năng bị bệnh tim

Bài tập 2: (Data Analysis) hãy hoàn thiện chương trình sau sử dụng thư viện seaborn để hiện thị mối quan hệ giữa độ tuổi, giới tính và khả năng bị bệnh tim. Hình 4 thể hiện kết quả sau khi thực hiện đoạn code visualization bên dưới:

```
# bai tap 2
```

```
# barplot of age vs sex with hue = target

#Your code here **********************************




plt.show()
```



Hình 4: Đồ thị thể hiện mỗi quan hệ giữa độ tuổi, giới tính và khả năng bị bệnh tim

Bài tập 3: (sử dụng **KNN** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng giải thuật KNN để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski'

```
# bai tap 3
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************

```

```
print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for KNeighborsClassifier = {}'.format(
    accuracy_for_train))
print('Accuracy for test set for KNeighborsClassifier = {}'.format(accuracy_for_test))
```

Question 1: Hãy cho biết kết độ chính xác của giải thuật KNN trên tập dữ liệu train và test ở bài tập 3.

```
a) accuracy for train = 0.76 and acccuracy for test = 0.69
b) accuracy for train = 1.76 and acccuracy for test = 0.69
c) accuracy for train = 2.76 and acccuracy for test = 0.69
d) accuracy for train = 3.76 and acccuracy for test = 0.69
```

Bài tập 4: (sử dụng **SVM** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng giải thuật SVM để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: kernel = 'rbf', random_state=42

```
# bai tap 4
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************

print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for SVM = {}'.format(accuracy_for_train))
print('Accuracy for test set for SVM = {}'.format(accuracy_for_test))
```

Question 2: Hãy cho biết kết độ chính xác của giải thuật SVM trên tập dữ liệu train và test ở bài tập 4.

```
a) accuracy for train = 0.76 and acccuracy for test = 0.69
b) accuracy for train = 0.66 and acccuracy for test = 0.67
c) accuracy for train = 2.76 and acccuracy for test = 0.69
d) accuracy for train = 3.76 and acccuracy for test = 0.69
```

Bài tập 5: (sử dụng **Naive Bayes** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng giải thuật Naive Bayes để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: kernel = 'rbf', random_state=42

```
# bai tap 5
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


```

```
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************

print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for Naive Bayes = {}'.format(accuracy_for_train))
print('Accuracy for test set for Naive Bayes = {}'.format(accuracy_for_test))
```

Question 3: Hãy cho biết kết độ chính xác của giải thuật Naive Bayes trên tập dữ liệu train và test ở bài tập 5.

```
a) accuracy for train = 0.76 and acccuracy for test = 0.69
b) accuracy for train = 0.66 and acccuracy for test = 0.67
c) accuracy for train = 0.85 and acccuracy for test = 0.84
d) accuracy for train = 3.76 and acccuracy for test = 0.69
```

Bài tập 6: (sử dụng **Decision Tree** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng giải thuật Decision Tree để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: criterion='gini', max_depth=10, min_samples_split=2

```
# bai tap 6
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************

print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for Decision Tree  = {}'.format(accuracy_for_train))
print('Accuracy for test set for Decision Tree  = {}'.format(accuracy_for_test))
```

Question 4: Hãy cho biết kết độ chính xác của giải thuật Decision tree trên tập dữ liệu train và test ở bài tập 6.

```
a) accuracy for train = 0.76 and acccuracy for test = 0.69
b) accuracy for train = 0.66 and acccuracy for test = 0.67
c) accuracy for train = 0.85 and acccuracy for test = 0.84
d) accuracy for train = 1.0 and acccuracy for test = 0.75
```

Bài tập 7: (sử dụng **Random Forest** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng

giải thuật Random Forest để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: criterion='gini', max_depth=10, min_samples_split=2, n_estimators = 10, random_state=42

```
# bai tap 7
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************

print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for Random Forest = {}'.format(accuracy_for_train))
print('Accuracy for test set for Random Forest = {}'.format(accuracy_for_test))
```

Question 5: Hãy cho biết kết độ chính xác của giải thuật Random Forest trên tập dữ liệu train và test ở bài tập 7.

```
a) accuracy for train = 0.98 and acccuracy for test = 0.8
b) accuracy for train = 0.66 and acccuracy for test = 0.67
c) accuracy for train = 0.85 and acccuracy for test = 0.84
d) accuracy for train = 1.0 and acccuracy for test = 0.75
```

Bài tập 8: (sử dụng **Adaboost** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng giải thuật Adaboost để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: n_estimators=50, learning_rate=1.0

```
# bai tap 8
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************

print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for Adaboost = {}'.format(accuracy_for_train))
print('Accuracy for test set for Adaboost = {}'.format(accuracy_for_test))
```

Question 6: Hãy cho biết kết độ chính xác của giải thuật Adaboost trên tập dữ liệu train và test ở bài tập 8.

```
a) accuracy for train = 0.98 and acccuracy for test = 0.8
b) accuracy for train = 0.91 and acccuracy for test = 0.84
c) accuracy for train = 0.85 and acccuracy for test = 0.84
d) accuracy for train = 1.0 and acccuracy for test = 0.75
```

Bài tập 9: (sử dụng **GradientBoost** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng giải thuật GradientBoost để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: learning_rate=0.1, n_estimators=100, subsample=1.0, min_samples_split=2, max_depth=3, random_state=42

```
# bai tap 9
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************

print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for GradientBoost = {}'.format(accuracy_for_train))
print('Accuracy for test set for GradientBoost = {}'.format(accuracy_for_test))
```

Question 7: Hãy cho biết kết độ chính xác của giải thuật GradientBoost trên tập dữ liệu train và test ở bài tập 9.

```
a) accuracy for train = 0.98 and acccuracy for test = 0.8
b) accuracy for train = 0.91 and acccuracy for test = 0.84
c) accuracy for train = 1.0 and acccuracy for test = 0.85
d) accuracy for train = 1.0 and acccuracy for test = 0.75
```

Bài tập 10: (sử dụng **XGboost** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng giải thuật XGboost để dự đoán bệnh nhân có khả bị bệnh tim hay không sử dụng các tham số sau: objective="binary:logistic", random_state=42, n_estimators = 100

```
# bai tap 10
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)


X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

# your code here **************************
```

```

print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for XGboost = {}'.format(accuracy_for_train))
print('Accuracy for test set for XGboost = {}'.format(accuracy_for_test))
```

Question 8: Hãy cho biết kết độ chính xác của giải thuật XGboosttrên tập dữ liệu train và test ở bài tập 10.

```
a) accuracy for train = 0.98 and acccuracy for test = 0.8
b) accuracy for train = 0.91 and acccuracy for test = 0.84
c) accuracy for train = 1.0 and acccuracy for test = 0.85
d) accuracy for train = 0.92 and acccuracy for test = 0.84
```

noindent

Bài tập 11: (sử dụng kỷ thuật **Stacking** cho dự đoán bệnh tim) hãy hoàn thiện chương trình sau sử dụng kỹ thuật Stacking để dự đoán bệnh nhân có khả bị bệnh tim hay bằng cách dùng phương pháp stacking sử dụng heterogeneous approach với mô hình giải thuật được thể hiện ở hình 1.

```
# bai tap 11
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 42)

dtc =  DecisionTreeClassifier(random_state=42)
rfc = RandomForestClassifier(random_state=42)
knn =  KNeighborsClassifier()
xgb = XGBClassifier(XGBClassifier)
gc = GradientBoostingClassifier(random_state=42)
svc = SVC(kernel = 'rbf', random_state=42)
ad = AdaBoostClassifier(random_state=42)


# your code here **********************


print()
accuracy_for_train = np.round((cm_train[0][0] + cm_train[1][1])/len(y_train),2)
accuracy_for_test = np.round((cm_test[0][0] + cm_test[1][1])/len(y_test),2)
print('Accuracy for training set for Stacking = {}'.format(accuracy_for_train))
print('Accuracy for test set for Stacking = {}'.format(accuracy_for_test))
```

Question 9: Hãy cho biết kết độ chính xác của giải thuật Stacking trên tập dữ liệu train và test ở bài tập 11.

```
a) accuracy for train = 0.92 and acccuracy for test = 0.9
b) accuracy for train = 0.91 and acccuracy for test = 0.84
c) accuracy for train = 1.0 and acccuracy for test = 0.85
d) accuracy for train = 1.0 and acccuracy for test = 0.84
```