THỰC HÀNH HỌC MÁY CÓ GIÁM SÁT VỚI K-NN

Yêu cầu: Xây dựng chương trình dự báo bệnh tim của bệnh nhân dựa trên các chỉ số xét nghiệm y khoa , bao gồm: 'age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal' .Biết rằng bệnh viện đã thu thập dữ liệu đầu vào và đánh nhãn chẩn đoán dựa vào các bác sỹ đầu ngành trong suốt hai năm, thông qua thăm khám bệnh nhân tại bệnh viện.

1. Thêm các thư viện cần thiết

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

2. Tải dữ liệu

```
df = pd.read_csv("heart.csv")
df
```

3. Tiến hành phân tích EDA các cột dữ liệu input
   a. Hiển thị cách đánh index của dữ liệu
   b. Danh sách các cột input

```
columns_eda = df.columns[:-1]
columns_eda
```

   c. Phân tích phân phối các biến số input

```
fig = plt.figure(figsize=(16,10))

for i in range(len(columns_eda)):
    colname = columns_eda[i]
    sub = fig.add_subplot(3,5,i+1)
    sns.histplot(data=df,x=colname, kde=True)
```

   d. Đếm xem số lượng các biến input (nhóm định tính) theo biến mục tiêu

```
categories_list = ['sex', 'cp', 'fbs', 'restecg','exang','slope', 'ca', 'thal']
fig = plt.figure(figsize=(20,8))

for i in range(len(categories_list)):
    colname = categories_list[i]
    sub = fig.add_subplot(2,4,i+1)
    sns.countplot(data=df,x=colname,hue="target")
```

   e. Phân tích biểu đồ Box-plot các biến input (nhóm định lượng) theo biến mục tiêu

```
numeric_list = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
fig = plt.figure(figsize=(20,8))

for i in range(len(numeric_list)):
    colname = numeric_list[i]
    sub = fig.add_subplot(2,4,i+1)
    sns.boxplot(data=df,y=colname, x="target")
```

4. Xây dựng mô hình
    a. Chuẩn bị dữ liệu

```
X = df.iloc[:,:-1].values
y = df[['target']].values
X = X.astype(float)
y = y.astype(float)
```

    b. Phân chia tập dữ liệu thành hai phần train và test tỉ lệ 80:20 và hệ số random là 42
    c. Hiển thị danh sách index của các sample trong tập X_train, X_test
    d. Huấn luyện mô hình

```
knn = KNeighborsClassifier(n_neighbors=8, algorithm="ball_tree")

knn.fit(X_train, y_train)
```

    e. Đánh giá mô hình trên tập test bằng độ đo accuracy

```
knn.score(X_test, y_test)
```

    f. Sử dụng độ đo đánh giá accuracy trên tập train và test để so sánh và chọn lựa K bao nhiêu là tốt nhất cho mô hình K-NN. Sinh viên nhìn biểu đồ để trả lời

```
import numpy as np
neighbors = np.arange(1, 20)
train_accuracy = np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

# Loop over K values
for i, k in enumerate(neighbors):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)

    # Compute training and test data accuracy
    train_accuracy[i] = knn.score(X_train, y_train)
    test_accuracy[i] = knn.score(X_test, y_test)

# Generate plot
plt.plot(neighbors, test_accuracy, label = 'Testing dataset Accuracy')
plt.plot(neighbors, train_accuracy, label = 'Training dataset Accuracy')

plt.legend()
plt.xlabel('n_neighbors')
plt.ylabel('Accuracy')
plt.show()
```

5. Xây dựng lại mô hình với K là tốt nhất dựa trên câu 4f. Sau đó, sử dụng kết quả đó để giải quyết từ câu 6 trở đi.
6. In ra danh sách khoảng cách và các hàng xóm từ các mẫu dữ liệu trong tập test từ mô hình ở câu 5

```
distances,indices = knn.kneighbors(X_test)
```

7. Cho biết sample input đầu tiên trong tập test sẽ lần lượt gần K (hàng xóm) dòng nào trong tập train tính theo index và khoảng cách tương đương theo độ đo Euclidean là bao nhiêu?
8. Cho biết với một sample input tương ứng ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal'] = [47,0,0,150,250,0,1,140,1,0.5,1,0,2] thì sẽ

gần K hàng xóm nào trong tập huấn luyện, với tương ứng lần lượt khoảng cách là bao nhiêu và cho biết giá trị Age của hàng xóm gần nhất.

9. Xem danh sách các giá trị của lớp đánh nhãn của biến output (target) nhằm xác định bệnh nhân có bệnh hay không
10. Xem danh sách xác suất tiền dự báo của các mẫu dữ liệu tập test theo các nhãn đầu ra
11. Cho biết mẫu input thứ hai trong tập test khi qua mô hình dự báo sẽ cho xác xuất dự báo nhãn nào cao hơn và giá trị là bao nhiêu, còn nhãn thấp là nhãn nào có xác suất bao nhiêu
12. Liệt kê danh sách các nhãn dự báo thông qua mô hình của các mẫu dữ liệu input trong tập test. Hãy cho biết kết quả dự báo của mẫu input thứ 5 trong tập test sẽ được dự báo là bao nhiêu?

```
y_test_predicted = knn.predict(X_test)
y_test_predicted
```

13. Hãy cho biết nếu điều chỉnh ngưỡng xác suất (threshold) đưa ra quyết định là 0.65 thì kết quả dự báo của mẫu input số hai và mẫu số năm và mẫu số sáu là bao nhiêu
14. Hãy đánh giá mô hình trên tập test qua Confussion – Matrix

```
from sklearn.metrics import confusion_matrix
cfmx = confusion_matrix(y_test, y_test_predicted)
cfmx
```

15. Dựa vào confusion matrix, hãy cho biết giá trị quan trọng nhất trong bài toán dự báo này.
16. Hãy đánh giá mô hình thông qua các giá trị Precision, Recall và F1 tổng quát. Trong bài toán dự báo này thì đại lượng nào quan trọng.
17. Hãy in ra bảng báo cáo các đại lượng đánh giá theo từng nhóm giá trị nhãn đầu ra

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_test_predicted))
```

18. Đánh giá mô hình thông qua đồ thị AUC & ROC

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics

y_pred_proba = knn.predict_proba(X_test)[:,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)
plt.plot(fpr,tpr,'go-',label="AUC="+str(auc))
plt.plot([0,1],[0,1],'r--')
plt.title("AUC & ROC Curve")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend(loc=4)
plt.fill_between(fpr, tpr, facecolor='lightgreen', alpha=0.7)
plt.show()
```

19. Hãy in ra màn hình accuracy, confusion matrix, [tn, fp, fn, tp], [TPR, FNR, FPR, TNR], [precision, recal, F1] và đồ thị AUC & ROC với ngưỡng xác xuất phân lớp là 0.65. Đưa ra kết luận khi thiết lập threshold tăng hoặc threshold giảm sẽ tác động như thế nào đến kết quả đánh giá mô hình.

20. Viết chương trình dự báo bệnh tim dưới dạng Console Application, cho phép người dùng nhập các đặc trưng đầu vào (feature input) và ngưỡng xác suất (threshold, nếu không nhập ngưỡng thì mặc định ngưỡng là 0.5). Sau đó, in ra kết quả chẩn đoán bệnh tim.

BÀI TẬP VỀ NHÀ

Sinh viên làm nghiên cứu tương tự cho tập dữ liệu iris. Mô tả dữ liệu iris: *The Iris dataset consists of 150 samples of iris flowers from three different species: Setosa, Versicolor, and Virginica. Each sample includes four features: sepal length, sepal width, petal length, and petal width. It was introduced by the British biologist and statistician Ronald Fisher in 1936 as an example of discriminant analysis.*