**Nhóm:**

**SV1:**

**SV2:**

**Machine Learing: Thuật toán Cây Quyết định (Decision Tree)**

Cây quyết định là một trong những thuật toán Machine Learning phổ biến nhất hiện nay. Nó được dùng trong cả bài toán phân lớp và hồi quy. Cây quyết định là một mô hình thuộc nhóm thuật toán Học có giám sát (Supervised Learning), thuật toán sẽ đưa ra quyết định dự trên các câu hỏi. Nhóm lựa chọn sử dụng bài toán phân lớp (Classifier) để giải quyết vấn đề.

1. **Vấn đề:**

Cho bảng dữ liệu thời tiết định dạng .csv gồm các trường dữ liệu sau: Outlook, Temp, Humidity, Windy, Play. Dựa vào các yêu tố thời tiết xác định liệu có ra ngoài chơi hay không ?

1. **Giải quyết:**

Để trả lời cho câu hỏi có đi ra ngoài chơi hay không, nhóm sẽ sử dụng thuật toán Dicision Tree Classifier. Trước hết cần import file csv chứa dataset. Do dữ liệu của dataset là dạng text để clear data cần chuyển từ dạng text dang number. Sau đó tiến hành chia dữ liệu làm 2 phần training và predict. Cuối dùng dùng thuật toán Dicision Tree Classifier để train ,predict và so kết qua predict với dữ liệu gốc để đánh giá độ chính xác của thuật toán.

* 1. **Importing Libraries**

Các thư viện cần dùng:

*from* sklearn.tree *import* DecisionTreeClassifier

*from* sklearn.preprocessing *import* LabelEncoder

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

*from* sklearn *import* metrics

*import* pandas *as* pd

* *Sklearn.tree.DecisionTreeClassifier*: thuật toán Cây quyết định phân lớp
* *Sklearn.preprocessing.LabelEncoder:* mã hóa text sang number bằng cách gán các từ giống nhau bằng một con số cụ thể.
* *Sklearn.model\_selection.train\_test\_split:* dùng đẻ chia dataset thành 2 phần train và predict
* *Sklearn metrics:* dùng để tính toán độ chính xác của kết quả dự đoán
* *Pandas:* thư viện dùng để đọc file .csv
  1. **Import Dataset**

Dùng phương thức **read\_csv()** của thư viện **pandas** để đọc file cvs, các tham số cần truyền gồm: ***path*** đường dẫn của file csv, ***header*** xác định hàng sẽ được chọn làm label và hàng đó không lưu và dataset, ***names*** tên của các trường dữ liệu, ***delimiter*** vì dữ liệu đọc vào sẽ được phân cách bởi ký tự “ ; ” delimiter sẽ xác định các cột dựa vào ký tự “ ; ”. Có thể kiểm tra dataset bằng ***head()***.

col\_names=['outlook','temp','humidity','windy','play']

*#import dataset*

dataset=pd.read\_csv('.\data\_e3.csv',header= 0,names=col\_names,delimiter=";")

print(dataset.head())

Kết quả:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| outlook | temp | humidity | windy | play |
| sunny | hot | high | FALSE | no |
| sunny | hot | high | TRUE | no |
| overcast | hot | high | FALSE | yes |
| rainy | mild | high | FALSE | yes |
| rainy | cool | normal | FALSE | yes |
| rainy | cool | normal | TRUE | no |
| overcast | cool | normal | TRUE | yes |
| sunny | mild | high | FALSE | no |
| sunny | cool | normal | FALSE | yes |
| rainy | mild | normal | FALSE | yes |
| sunny | mild | high | TRUE | yes |
| overcast | mild | high | TRUE | yes |
| overcast | hot | normal | FALSE | yes |
| rainy | mild | high | TRUE | no |

* 1. **Data Cleaning**

Chia dataset thành 2 phần ***feature*** và ***target***

feature\_cols=['outlook','temp','humidity','windy']

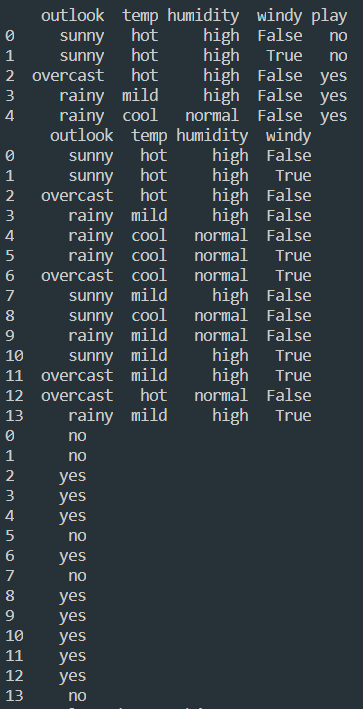
feature=dataset[feature\_cols]

target=dataset.play

print(feature)

print(target)

Kết quả:



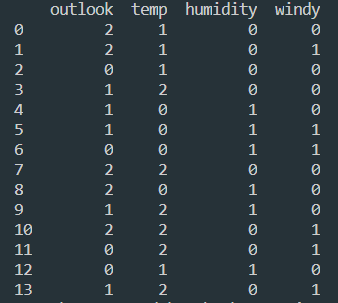
* 1. **Convert Text to number**

Thực hiện convert text sang number bằng phương thức ***LabelEncoder()***

enCode=LabelEncoder()

feature\_en=feature.apply(enCode.fit\_transform)

Kết quả:



* 1. **Split Data into Training and Test Sets**

Sử dụng phương thức ***train\_test\_split****()* để chia data, các tham số cần truyền gồm: mảng feature, mảng label của data, ***test\_size*** tý lệ chia data, ***random\_state*** chia data bằng cách lựa chọn ngẫu nhiên.

x\_train,x\_test , y\_train, y\_test =train\_test\_split(feature\_en,target, test\_size=0.3, random\_state=1)

* 1. **Training and Test**

Tạo đối tượng ***DecisionTreeClassifier()***

clf =DecisionTreeClassifier()

Thực hiện training

clf.fit(x\_train,y\_train)

Tiến hành dự đoán data test và so sánh kết quả dự đoán với kết quả thực tế.

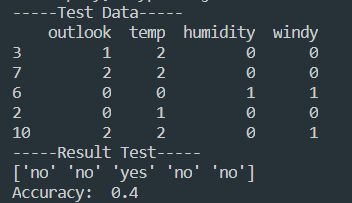
*#Predict*

result= clf.predict(x\_test)

print(result)

print("Accuracy: ",metrics.accuracy\_score(y\_test,result))

Kết quả:



Độ chính xác của kết quả dự đoán của thuật toán ***DecisionTreeClassifier*** so với thực tể là ***0.4.***

1. **Kết luận**

Độ chính xác của DecisionTreeClassifier sau vài lần test luôn là 0.4. Nguyên nhân có thể là do data training quá ít dẫn đến kết quả bị sai lệch với thực tế.