TRƯỜNG ĐẠI HỌC HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC

NHẬP MÔN HỌC MÁY

NGÀNH: CÔNG NGHỆ THÔNG TIN

Đề tài:

ÁP DỤNG THUẬT TOÁN DECISION TREE ĐỂ DỰ ĐOÁN NGUY CƠ RỚT TỐT NGHIỆP CỦA SINH VIÊN

SVTH: Trần Tấn Phát
MSSV: 2274802010644

GVHD: TRẦN NGỌC VIỆT

THÀNH PHỐ HỒ CHÍ MINH – NĂM 2025

LỜI CẢM ƠN

Lời nói đầu tiên, chúng em xin chân thành cảm ơn sự hướng dẫn của thầy Trần Ngọc Việt, người đã luôn dành thời gian và tâm huyết để hỗ trợ chúng em trong suốt quá trình học tập và nghiên cứu. Trong quá trình thực hiện nghiên cứu đề tài, chúng em đã gặp không ít khó khăn nhưng nhờ có sự hướng dẫn tận tình của thầy nên nhóm em đã có thể hoàn thành tốt bài tiểu luận.

Tiếp theo, chúng em xin gửi lời chân thành cảm ơn đến Khoa Công nghệ thông tin- Đại Học Văn Lang thành phố Hồ Chí Minh đã tạo điều kiện thuận lợi cho chúng em học tập và hoàn thành đề tài tiểu luận này.

Mặc dù nhóm đã rất cố gắng vận dụng những kiến thức đã học được trong thời gian qua để hoàn thành bài tiểu luận nhưng do không có nhiều kinh nghiệm thực tiễn nên khó tránh khỏi những thiếu sót trong quá trình nghiên cứu và làm bài. Nhưng với tinh thần cầu tiến và mong muốn tiến bộ, chúng em tin rằng những ý kiến và đóng góp của quý thầy cô và các bạn đọc giả sẽ góp phần giúp chúng em hoàn thiện bản thân hơn.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT – THUẬT TOÁN DECISION TREE

1.1. Lý do chọn đề tài

Với đề tài Decision Tree là một dễ hiểu và dễ giải thích cho những người không cùng chuyên ngành, phù hợp để tính toán và áp dụng vào các bài toán thực tế. Decision Tree có cấu trúc giống như cách con người quyết định theo kiểu "nếu…thì".

1.2. Cơ sở lý thuyết Decision Tree

A. Decision Tree là gì?

Decision Tree là 1 mô hình học máy dùng để giải quyết các bài toán phân loại và hồi quy. Mô hình hoạt động giống như cách con người ra quyết định theo các dạng chuỗi câu hỏi "có…không" và "nếu…thì".

Cấu trúc của Decision Tree bao gồm:

- Nút gốc (Root Node): Bắt đầu của cây, chứa toàn bộ dữ liệu.
- Nút trong (Internal Node): Đại diện cho các câu hỏi hoặc điều kiện phân chia dữ liệu.
- Nhánh (Branches): Thể hiện kết qủa của nút đó.
- Nút lá (Leaf Node): Kết quả cuối cùng của cây.

Ví dụ:

Nếu GPA < 2.0 => Sinh viên có nguy cơ rớt tốt nghiệp.

Nếu GPA >= 2.0 và số buổi tham gia học > 15 => Sinh viên đủ điều kiện tốt nghiệp.

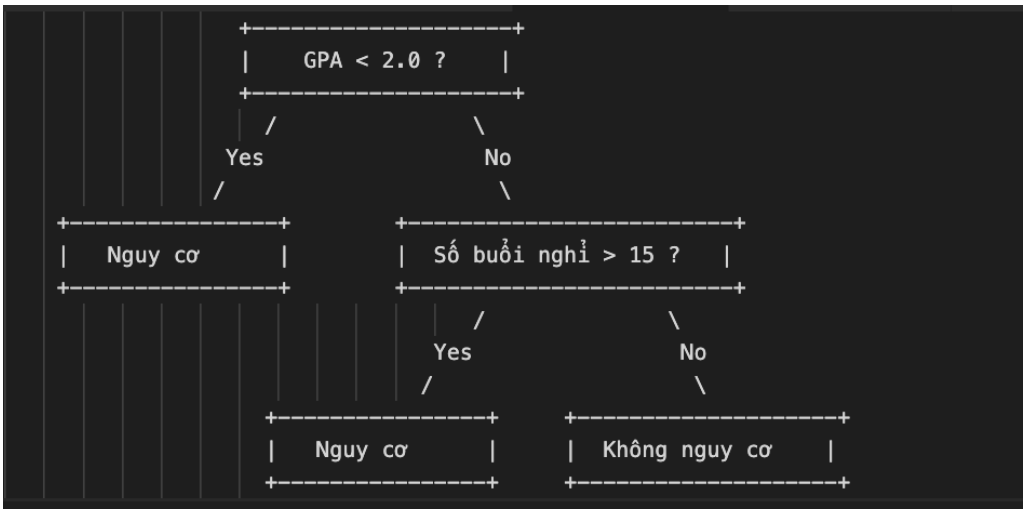
❖ **Định nghĩa**

- Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào các dãy luật.
- Các thuộc tính của đối tượng n thuộc tính các kiểu dữ liệu khác nhau như Nhị phân (Binary) , Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.
- Dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

B. Một số kiểu mô hình Decision Tree

❖ Cây phân loại (Classification Tree)

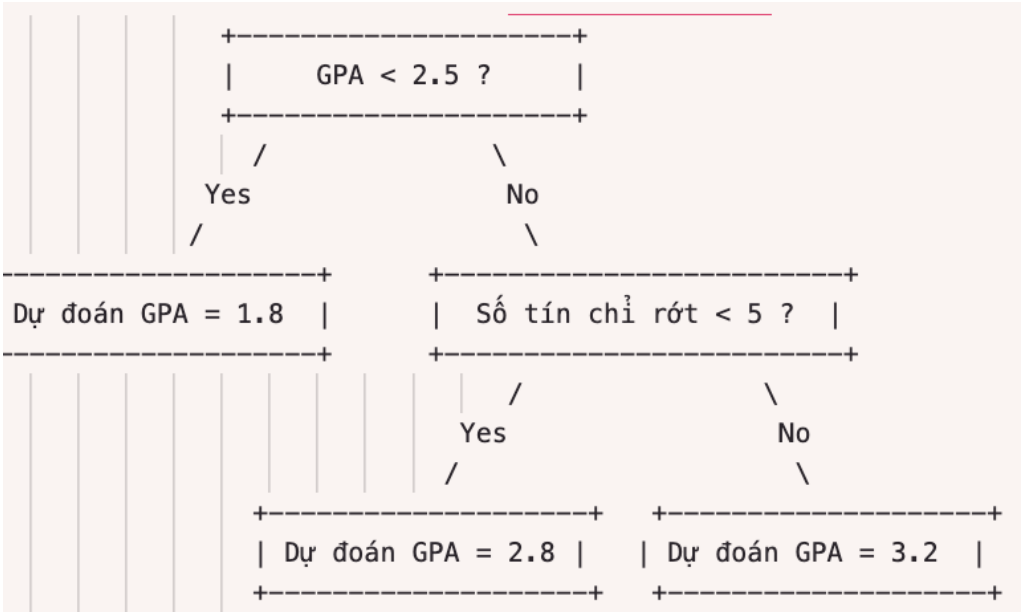
- Cây phân loại (Classification Tree) là một thuật toán thuộc nhóm **Decision Tree**, dùng để giải quyết các bài toán phân loại (Classification).
- Mục tiêu chính của cây phân loại là dựa vào các đặc điểm (thuộc tính) của dữ liệu để phân chia thành các nhóm rõ ràng (các lớp nhãn).
- ;/'?Kết quả tại các nút lá cuối cùng của cây là nhãn phân loại (ví dụ như "Nguy cơ" – "Không nguy cơ", "Đạt" – "Không đạt", "Có" – "Không").
- Thuật toán sẽ lựa chọn thuộc tính nào giúp phân tách dữ liệu tốt nhất tại mỗi bước dựa trên các tiêu chí đo lường độ thuần khiết của dữ liệu.
- Khi cây đạt đến điều kiện dừng (như độ sâu tối đa, số mẫu tối thiểu, hoặc mức độ thuần khiết đủ cao), mô hình sẽ dừng lại và đưa ra kết quả phân loại ở các nút lá.



Hình 1. Hình minh hoạ cho mô hình cây phân loại

❖ **Cây hồi quy (Regression Tree)**

- Cây hồi quy là một dạng của mô hình Decision Tree được sử dụng để giải quyết các bài toán dự đoán giá trị liên tục (continuous value prediction), thay vì phân loại ra các nhóm như cây phân loại.
- Cây hồi quy cho ra các giá trị số thực tại các nút (vd: Thời gian tốt nghiệp, số điểm GPA,..).
- Dữ liệu được chia nhỏ dần dựa trên các điều kiện ngưỡng (threshold) của các thuộc tính để giảm sai số dự đoán.
- Quá trình chia tiếp tục cho đến khi đạt điều kiện dừng (số lượng mẫu nhỏ, độ sâu tối đa, hoặc sai số đủ nhỏ).



Hình 2. Hình mô phỏng mô hình cây hồi quy

C. Ứng dụng thuật toán Decision Tree

❖ **Real-time Decision Support:**

- Decision Tree giúp đưa ra khả năng quyết định rõ ràng, phù hợp vào việc áp dung các hệ thống hỗ trợ trong thời gian thực.

❖ **Multi-class Classification:**

- Thuật toán Decision Tree có thể xử lý tốt các bài toán phân loại đa lớp nhờ vào cấu trúc phân tách theo nhiều nhánh dựa trên các thuộc tính khác nhau.

- Điều này giúp Decision Tree được ứng dụng rộng rãi trong các hệ thống đánh giá, phân loại nhiều mức độ hoặc trong các hệ thống phân nhóm khách hàng dựa trên hành vi tiêu dùng.

❖ **Risk Analysis/Failure Detection:**

- Decision Tree rất hiệu quả trong các bài toán phân tích rủi ro hoặc phát hiện lỗi, nhờ khả năng xác định rõ nguyên nhân dẫn đến các tình huống xấu thông qua các nút quyết định.
- Chẳng hạn, trong môi trường giáo dục, Decision Tree có thể phát hiện nguyên nhân chính khiến sinh viên có nguy cơ bỏ học dựa trên GPA thấp, số tín chỉ trượt cao, hoặc số buổi nghỉ nhiều.

1.3 Kết luận

❖ Thuật toán Decision Tree là một trong những phương pháp học máy trực quan, dễ hiểu và dễ hiểu nhất hiện nay. Với cấu trúc nhánh rõ ràng. Decision Tree giúp mô hình hoá dữ liệu và các quyết định phức tạp thành các bước đi có trình tự rõ ràng, cụ thể, từ đó người dùng có thể dễ dàng phân tích nguyên nhân và đưa ra kết quả đầu ra.

❖ Decision Tree đặc biệt phù hợp cho cả bài toán phân loại (Classification) và hồi quy (Regression), đồng thời xử lý tốt dữ liệu dạng bảng với nhiều thuộc tính khác nhau. Nhờ khả năng xác định các yếu tố quan trọng ảnh hưởng đến quyết định cuối cùng, thuật toán này được ứng dụng rộng rãi trong các lĩnh vực

❖ Tóm lại, với ưu điểm dễ xây dựng, dễ giải thích và phù hợp với nhiều dạng bài toán thực tế, Decision Tree là một công cụ mạnh mẽ và đáng tin cậy trong lĩnh vực trí tuệ nhân tạo và khai phá dữ liệu.

CHƯƠNG 2. ÁP DỤNG THUẬT TOÁN DECISION TREE CHO BÀI TOÁN DỰ ĐOÁN NGUY CƠ BỎ HỌC CỦA SINH VIÊN

2.1 Phát biểu bài toán:

Xây dựng mô hình học máy sử dụng thuật toán Decision Tree để dự đoán nguy cơ rớt tốt nghiệp của sinh viên dựa trên các thông tin cá nhân, học tập và hành vi của sinh viên.

2.2 Minh họa bài toán

Tập dữ liệu đã chuẩn bị:

- Các đặc trưng: ["Tuoi", "Gioi_tinh", "GPA", "So_tin_chi_truot", "So_buoi_nghi", ...]
- Nhãn: "Nguy_co_bo_hoc" (1 hoặc 0).

Decision Tree sẽ duyệt qua các đặc trưng để tìm:

- Đặc trưng nào giúp phân chia tốt nhất (tức Information Gain lớn nhất hoặc Gini nhỏ nhất).
- Sau đó tiếp tục chia nhỏ cho đến khi đạt đến độ sâu tối đa (ở đây bạn dùng max_depth=4) hoặc các tiêu chí dừng khác.

Ví dụ tại gốc cây:

- 60% sinh viên không có nguy cơ bỏ học, 40% có nguy cơ.
- Entropy= $-(0{,}6 \times \log_2(0{,}6) + 0{,}4 \times \log_2(0{,}4)) = 0{,}971$

Cây quyết định sẽ kiểm tra từng đặc trưng:

- Nếu chia theo GPA:
  - GPA < 2.0 → phần lớn có nguy cơ bỏ học.
  - GPA >= 2.0 → phần lớn không bỏ học.

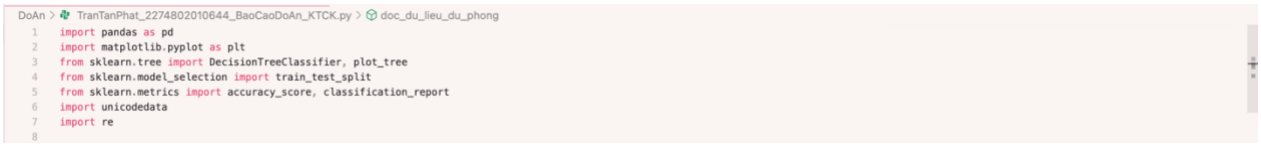
Nếu Information Gain của việc chia theo GPA là lớn nhất, nó sẽ chọn GPA làm nút gốc, và tiếp tục chia nhỏ các nhánh con dựa trên các đặc trưng khác như "So_buoi_nghi", "So_tin_chi_truot",…

2.3 Mã nguồn

❖ **Import thư viện (Hình 3)**

Các thư viện được sử dụng để thực hiện:

- Pandas: Xử lý dũ liệu dạng bảng.
- Matplotlib: Vẽ đồ hoạ.
- Sklearn: Dùng mô hình Decision Tree để phân loại và đánh giá.
- Unicodedata, re: Xử lý và làm sạch văn bản.



Hình 3

❖ **Đọc dữ liệu CSV với các encoding dự phòng (Hình 4)**

- Hàm doc_du_lieu_du_phong(): Dùng để tránh lỗi khi gặp dữ liệu bị mã hoá lạ.
- Đọc dữ liệu file CSV: Đọc file từ đường dẫn cụ thể.
- In thông tin dữ liệu: Đếm số dòng dữ liệu đọc được, hiển thị thông tin tổng quát, hiển thị 5 dòng đầu tiên của bảng dữ liệu để quan sát cấu trúc.

```
Codeium: Refactor | Explain | Generate Docstring | ×
def doc_du_lieu_du_phong(duong_dan):
    encodings = ['utf-8', 'utf-8-sig', 'latin1', 'cp1252', 'cp1258']
    for enc in encodings:
        try:
            data = pd.read_csv(duong_dan, sep='\t', encoding=enc)
            print(f"Đọc thành công với encoding: {enc}")
            return data
        except Exception as e:
            print(f"Lỗi với encoding {enc}: {e}")
    raise Exception("Không thể đọc dữ liệu với các encoding đã thử!")

du_lieu = doc_du_lieu_du_phong('/Users/trantanphat/Documents/Python/NMHM/DoAn/SinhVien.csv')

print(f"Số dòng dữ liệu ban đầu: {len(du_lieu)}")
print("\nThông tin dữ liệu:")
print(du_lieu.info())
print("\n5 dòng đầu tiên:")
print(du_lieu.head())
```

Hình 4

- **Làm sạch dữ liệu văn bản(Hình 5):** Chuẩn hoá Unicode, xoá các ký tự đặc biệt.

- **Loại bỏ bản bản ghi không hợp lệ (Hình 6)**
  - Xử lý trùng lặp và giá trị khuyết bằng các hàm như xoá bản ghi trùng (drop_duplicates), xoá bản ghi giá trị thiếu (dropna), loại bỏ bản ghi có Noi_song = Khac.
  - Lọc dữ liệu theo điều kiện hợp lý:
  - GPA: 0-4.0
  - Tuoi: 17-30
  - So_tin_chi_truot >= 0
  - So_buoi_nghi >= 0

- **Thêm điều kiện đặc trưng phụ trợ**
  - Xếp loại học lực dựa trên GPA.
  - Điều kiện tốt nghiệp hoàn thiện nếu đạt các điều kiện như: GPA>= 2.0, Số tín chỉ trượt <= 10, Số buổi nghỉ <= 15.

```
DoAn > TranTanPhat_2274802010644_BaoCaoDoAn_KTCK.py > ...
Codeium: Refactor | Explain | Generate Docstring | X
def lam_sach_text(text):
    if isinstance(text, str):
        text = unicodedata.normalize('NFKD', text.strip())
        text = re.sub(r'[^\w\s]', '', text)
        text = re.sub(r'_', ' ', text)
    return text

cac_cot_text = ['Gioi_tinh', 'Chuyen_nganh', 'Ton_giao', 'Noi_song', 'Quoc_tich']
for cot in cac_cot_text:
    du_lieu[cot] = du_lieu[cot].apply(lam_sach_text)

du_lieu.drop_duplicates(inplace=True)
du_lieu.dropna(inplace=True)
du_lieu = du_lieu[du_lieu['Noi_song'] != 'Khac']
du_lieu = du_lieu[
    (du_lieu['GPA'].between(0, 4.0)) &
    (du_lieu['Tuoi'].between(17, 30)) &
    (du_lieu['So_tin_chi_truot'] >= 0) &
    (du_lieu['So_buoi_nghi'] >= 0)
]
```

```
# Thêm cột xếp loại học lực và điều kiện tốt nghiệp
du_lieu['Hoc_luc'] = du_lieu['GPA'].apply(lambda gpa: (
    'Xuat sac' if gpa >= 3.6 else
    'Gioi' if gpa >= 3.2 else
    'Kha' if gpa >= 2.5 else
    'Trung binh' if gpa >= 2.0 else
    'Yeu'
))

du_lieu['Dieu_kien_tot_nghiep'] = du_lieu.apply(
    lambda row: 'Dat' if row['GPA'] >= 2.0 and row['So_tin_chi_truot'] <= 10 and row['So_buoi_nghi'] <= 15 else 'Khong Dat',
    axis=1
)
```

❖ Mã hoá dữ liệu danh mục (Hình 7)

- Biến giá trị văn bản thành số để phục vụ trong việc huấn luyện mô hình.
- Các cột được mã hoá để huấn luyện gồm: Gioi_tinh, Chuyen_nganh, Ton_giao, Noi_song, Quoc_tich, Hoc_luc, Dieu_kien_tot_nghiep.

```
# Mã hóa dữ liệu
bang_ma_hoa = {
    'Gioi_tinh': {'Nam': 0, 'Nu': 1},
    'Chuyen_nganh': {'Cong nghe thong tin': 0, 'Ke toan': 1, 'Quan tri kinh doanh': 2},
    'Ton_giao': {'Khong': 0, 'Phat giao': 1, 'Thien chua giao': 2},
    'Noi_song': {'TP.HCM': 0, 'Ha Noi': 1, 'Da Nang': 2},
    'Quoc_tich': {'Viet Nam': 0, 'Lao': 1, 'Campuchia': 2},
    'Hoc_luc': {'Xuat sac': 0, 'Gioi': 1, 'Kha': 2, 'Trung binh': 3, 'Yeu': 4},
    'Dieu_kien_tot_nghiep': {'Dat': 1, 'Khong Dat': 0}
}
for cot, ma_hoa in bang_ma_hoa.items():
    du_lieu[cot] = du_lieu[cot].map(ma_hoa)
```

Hình 7

❖ Kiểm tra dữ liệu rỗng và chuẩn bị dữ liệu mô hình huấn luyện (Hình 8)

- Tạo cột Nguy_co_rot_tot_nghiep (biến mục tiêu) với 1 và 0:
    - 1: Là Nguy_co_rot_tot_nghiep nếu GPA<2.0 và số buổi nghỉ >15.

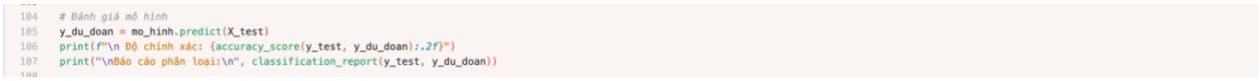
- 0: Là ngược lại với điều kiện.

- Kiểm tra dữ liệu rỗng: Tạo ra dữ liệu bị xoá sau khi xử lý, chương trình sẽ dừng lại và cảnh báo lỗi.
- Chuẩn bị dữ liệu cho mô hình huấn luyện: Biến đầu vào gồm 12 đặc trưng như: “Tuoi”, “Gioi_tinh”, “GPA”, “Chuyen_nganh”,…

Hình 8

❖ Đánh giá mô hình (Hình 9)

- Dự đoán tập tin Test bằng mô hình huấn luyện (X_test).
- In độ chính xác (accuracy_score).
- In báo cáo phân loại bao gồm:
  - Precision: Độ chính xác của từng lớp.
  - Recall: Khả năng tìm đúng của mô hình.
  - Support: Số lượng mẫu của từng lớp.



Hình 9

❖ Vẽ cây quyết định (Decision Tree) (Hình 10)

- Vẽ trực quan hoá cây quyết định của mỗi mô hình kích thước (30,20).
- Các tham số gồm:
  - Feature_name: tên của các đặc trưng đầu vào.
  - Class_names: tên của các lớp đầu ra (0: “Khong rot tot nghiep”, 1: “Nguy co rot tot nghiep”)
  - Filled= True: Tô màu các xác suất dự đoán.
  - Rounded= True: bo tròn góc các nút.
  - Fontsize= 14: Kích thước chữ trong cây.
- Tiêu đề biểu đồ: “Cay quyet dinh nguy co rot tot nghiep cua sinh vien”.

```
# Vẽ cây quyết định
plt.figure(figsize=(30, 20))
plot_tree(
    mo_hinh,
    feature_names=dac_trung,
    class_names=["Khong rot tot nghiep", "Nguy co rot tot nghiep"],
    filled=True,
    rounded=True,
    fontsize=14
)
plt.title("Cay quyet dinh nguy co rot tot nghiep cua sinh vien", fontsize=22)
plt.show()
```

Hình 10

❖ In kết quả dự đoán (Hình 11)

- Tạo bản sao của tập dữ liệu kiểm tra (X_test) để lưu kết quả dự đoán, tránh làm thay đổi dữ liệu gốc.
- Lấy tên sinh viên từ dữ liệu theo đúng chỉ số của dòng X_test để đảm bảo thông tin chính xác.
- Gán cột “Ten_sinh_vien” vào DataFrame.
- Lấy giá trị thực tế của nhãn từ tập kiểm tra (y_test) thông qua dữ liệu “Thuc_te”.
- Tạo ra kết quả dự đoán mô hình (y_du_doan) từ tập dữ liệu của “Du_doan”.
- In tiêu đề và 3 cột mốc quan trọng gồm: “Ten_sinh_vien”, “Thuc_te”, “Du_doan”.

```
# In kết quả dự đoán kèm tên sinh viên
ket_qua = X_test.copy()
ket_qua['Ten_sinh_vien'] = du_lieu.loc[X_test.index, 'Ten_sinh_vien']
ket_qua['Thuc_te'] = y_test.values
ket_qua['Du_doan'] = y_du_doan

print("\n Kết quả dự đoán:")
print(ket_qua[['Ten_sinh_vien', 'Thuc_te', 'Du_doan']])
```

2.4 Ưu/ Nhược điểm

❖ Ưu điểm

- Xử lý dữ liệu đầy đủ và kỹ lưỡng: Có các bước là sạch dữ liệu, nhiều đặc trưng hữu ích, thực tế để tăng chất lượng dự đoán.
- Mã hoá rõ ràng: Áp dụng bảng mã cụ thể, đảm bảo mô hình được số hoá chính xác.
- Cây được phân tích dễ hiểu và trực quan: Cây được vẽ đơn giản, rõ ràng, giúp người dùng dễ dàng phân tích nguyên nhân gây rớt tốt nghiệp.
- Kết quả phân tích chi tiết: In kết quả so sánh sinh viên, so sánh thực tế và dự đoán giúp kiểm tra độ chính xác.

❖ Nhược điểm

- Chưa đánh giá được mô hình chuyên sâu: Chỉ dùng các hàm đơn giản, chưa thêm được các chỉ số như ROC-AUC, hay đánh giá chéo.
- Chưa xử lý mất cân bằng dữ liệu: Nếu tỉ sinh viên có nguy cơ rớt và không rớt quá chênh lệch, mô hình sẽ bị lệch và độ chính xác cao nhưng dự đoán kém.
- Mô hình đơn giản: Decision Tree có độ sâu nhất định sẽ bị underfitting nếu thêm vào dữ liệu phức tạp, không khai thác tối ưu.

2.5Kết quả trả về

❖ Kết quả xuất ra từ dữ liệu csv: 10 dữ liệu sinh viên

❖ **Kiểm tra DataFrame chứa tất cả các info bên trong file**

```
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Ten_sinh_vien         100 non-null    object
 1   Ma_so                 100 non-null    object
 2   Tuoi                  100 non-null    int64
 3   Gioi_tinh             100 non-null    object
 4   GPA                   100 non-null    float64
 5   So_tin_chi_truot      100 non-null    int64
 6   So_buoi_nghi          100 non-null    int64
 7   Chuyen_nganh          100 non-null    object
 8   Ton_giao              100 non-null    object
 9   Nien_khoa             100 non-null    int64
 10  Noi_song              100 non-null    object
 11  Quoc_tich             100 non-null    object
 12  Dieu_kien_tot_nghiep  100 non-null    object
dtypes: float64(1), int64(4), object(8)
```

❖ **Lấy 5 sinh viên đầu tiên để thử độ chính xác của ứng dụng**

5 dòng đầu tiên:

	Ten_sinh_vien	Ma_so	Tuoi	Gioi_tinh	GPA	So_tin_chi_truot	So_buoi_nghi	Chuyen_nganh	Ton_giao	Nien_khoa	Noi_song	Quoc_tich	Dieu_kien_tot_nghiep
0	Sinh vien 1	SV001	21	Nu	1.6	1	1	Ke toan	Phat giao	2021	Ha Noi	Viet Nam	Khong Dat
1	Sinh vien 2	SV002	22	Nam	1.7	2	2	Quan tri kinh doanh	Thien chua giao	2022	Da Nang	Viet Nam	Khong Dat
2	Sinh vien 3	SV003	23	Nu	1.8	3	3	Cong nghe thong tin	Thien chua giao	2023	TP.HCM	Viet Nam	Khong Dat
3	Sinh vien 4	SV004	24	Nam	1.9	4	4	Ke toan	Khong	2020	Ha Noi	Viet Nam	Khong Dat
4	Sinh vien 5	SV005	20	Nu	2.0	5	5	Quan tri kinh doanh	Phat giao	2021	Da Nang	Campuchia	Dat

Độ chính xác: 1.00

❖ **Báo cáo hiệu suất của phân loại, đánh giá hiệu suất**

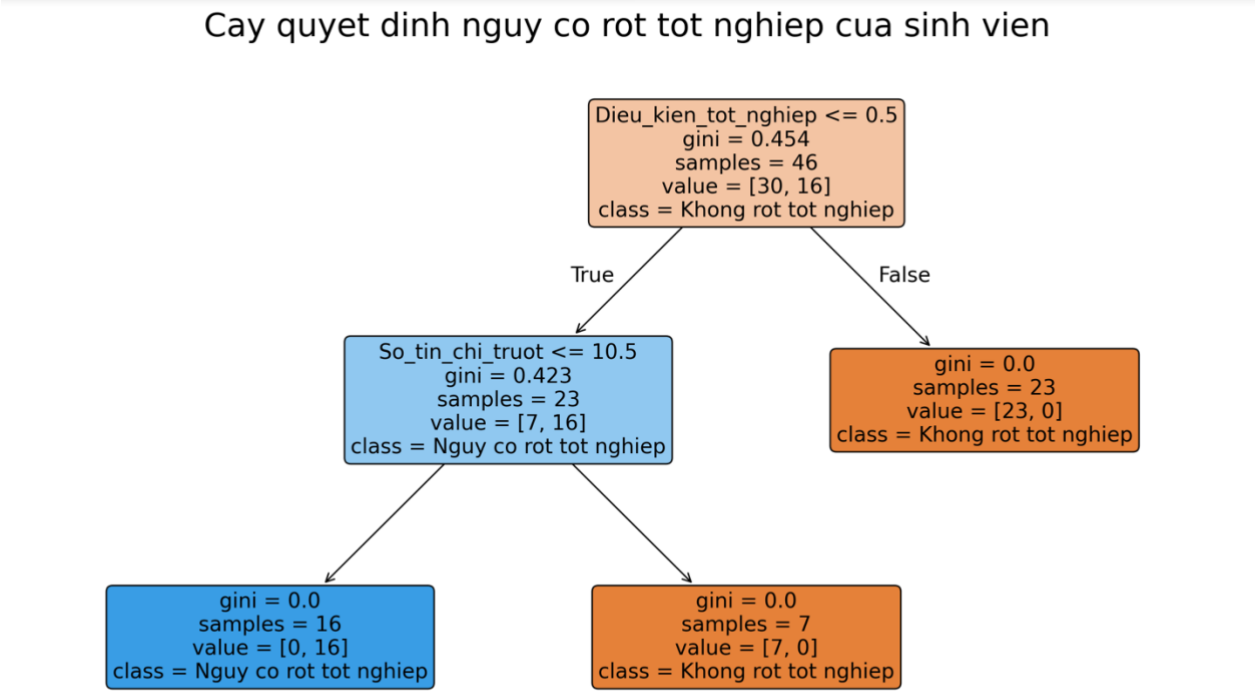
Báo cáo phân loại:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6
1	1.00	1.00	1.00	6
accuracy			1.00	12
macro avg	1.00	1.00	1.00	12
weighted avg	1.00	1.00	1.00	12

❖ **Kết quả dự đoán mô hình dựa trên tập dữ liệu**

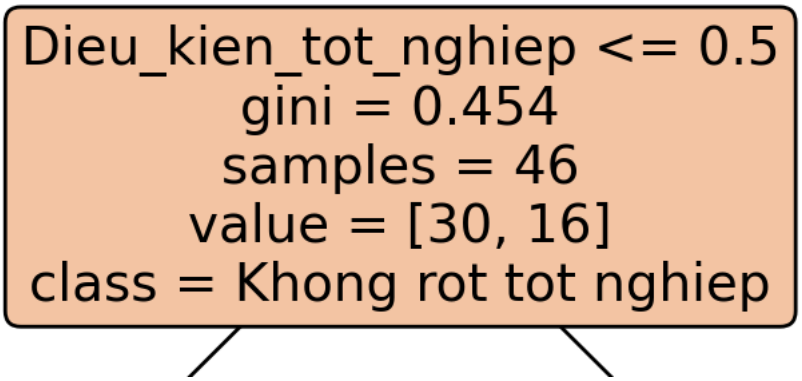
```
Kết quả dự đoán :
   Ten_sinh_vien  Thuc_te  Du_doan
0     Sinh vien 1        1        1
7     Sinh vien 8        0        0
60  Sinh vien 61         1        1
19  Sinh vien 20         0        0
75  Sinh vien 76         1        1
91  Sinh vien 92         1        1
64  Sinh vien 65         0        0
42  Sinh vien 43         0        0
78  Sinh vien 79         1        1
18  Sinh vien 19         1        1
```

❖ **Xuất ra Decision Tree Classification**



Giải thích Gini:

1. Nút gốc



Tính xác suất mỗi lớp:

p1 = 30 / 46 = 0.6522 p2 = 16 / 46 =0.3478

Tính Gini:

Gini = 1 - (p1^2 + p2^2)

Thay số vào:

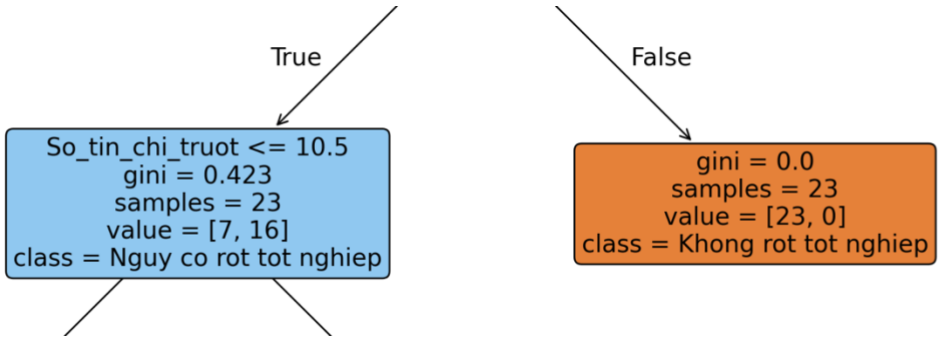
Gini = 1 - (0.6522^2 + 0.3478^2)

= 1 - (0.4253 + 0.1210)

= 1 - 0.5463

= 0.454

2. Nút trái và nút phải



Tính xác suất mỗi lớp:

p1 = 7 / 23 = 0.3043 p2 = 16 / 23 =0.6957

Tính Gini:

Gini = 1 - (p1^2 + p2^2)

Thay số vào:

Gini = 1 - (0.3043^2 + 0.6957^2)

= 1 - (0.0926 + 0.4840)

= 1 - 0.5766

= 0.423

Tính xác suất mỗi lớp:

p1 = 23 / 23 = 1 p2 = 0 / 23 = 0

Tính Gini:

Gini = 1 - (p1^2 + p2^2)

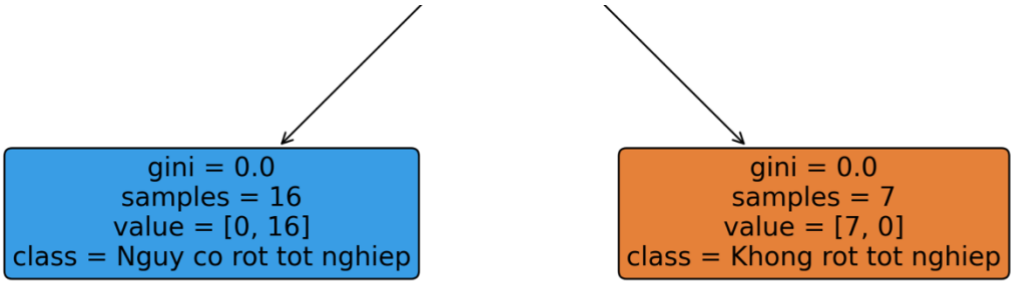
Thay số vào:

Gini = 1 - (1^2 + 0^2)

= 1 - 1

= 0

3. Nút con trái và nút con phải



Tính xác suất mỗi lớp:

p1 = 0 / 16 = 0 p2 = 16 / 16 = 1

Tính Gini:

Gini = 1 - (p1^2 + p2^2)

Thay số vào:

Gini = 1 - (0^2 + 1^2)

= 1 - 1

= 0

Tính xác suất mỗi lớp:

p1 = 7 / 7 = 1 p2 = 0 / 7 = 0

Tính Gini:

Gini = 1 - (p1^2 + p2^2)

Thay số vào:

Gini = 1 - (1^2 + 0^2)

= 1 - 1

= 0

CHƯƠNG 3. KẾT LUẬN

Toàn bộ đoạn code trên xây dựng một quy trình đầy đủ và rõ ràng để xử lý dữ liệu sinh viên và dự đoán nguy cơ rớt tốt nghiệp bằng mô hình Decision Tree Classifier. Quy trình bao gồm các bước từ đọc dữ liệu với nhiều encoding khác nhau, làm sạch và chuẩn hóa dữ liệu, thêm các cột đặc trưng như học lực và điều kiện tốt nghiệp, mã hóa các giá trị phân loại, đến việc xác định nhãn mục tiêu là nguy cơ rớt tốt nghiệp dựa trên GPA và số buổi nghỉ. Sau khi chuẩn bị dữ liệu, mô hình cây quyết định được huấn luyện với độ sâu cố định và đánh giá bằng độ chính xác cùng báo cáo phân loại.

Ngoài ra, mô hình còn được trực quan hóa thông qua biểu đồ cây quyết định và hiển thị kết quả dự đoán kèm tên sinh viên để dễ dàng kiểm tra. Nhìn chung, đoạn code đã xử lý tốt toàn bộ quy trình từ tiền xử lý dữ liệu đến huấn luyện và đánh giá mô hình, tuy nhiên có thể cải tiến thêm ở các bước như phân tích dữ liệu, tối ưu tham số mô hình và lưu trữ kết quả để tăng tính ứng dụng thực tế và độ chính xác của mô hình.

TÀI LIỆU THAM KHẢO

Kaggle: https://www.kaggle.com/code/kareemabdelhamed/100-automated-process-cleaning-eda-modeling-ai

Youtube: https://www.youtube.com/watch?v=_L39rN6gz7Y&t=18s

Github: https://github.com/apache/hadoop

E-learning: https://byvn.net/9PGE

Phần mềm thứ 3: ChatGPT, Coursera, Copilot, Deepseek,…

LINK GITHUB

https://github.com/trantanphat0811/Introducting_To_Machine_Learning/tree/main/DoAn