

Báo cáo đồ án

DỰ ĐOÁN KẾT QUẢ HỌC TẬP HỌC KÌ TIẾP THEO CỦA SINH VIÊN UIT

Giảng viên: Nguyễn Thị Anh Thơ - IS353.P12

NHÓM 3

Ngô Thùy Yến Nhi - 21521230

Ngô Kỳ Anh - 21521825

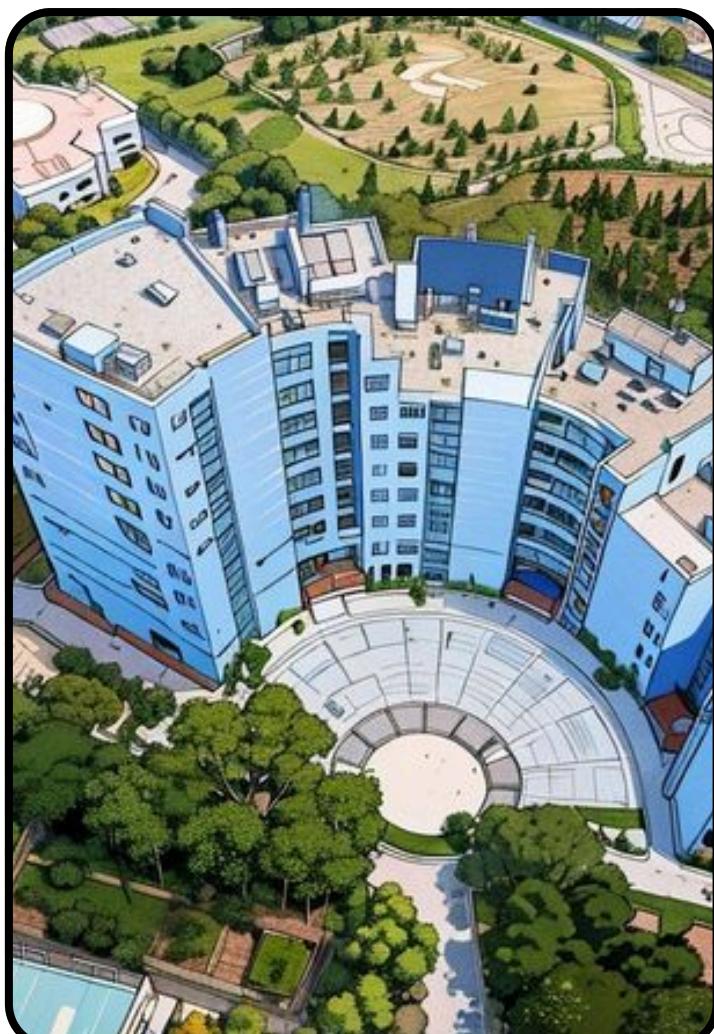
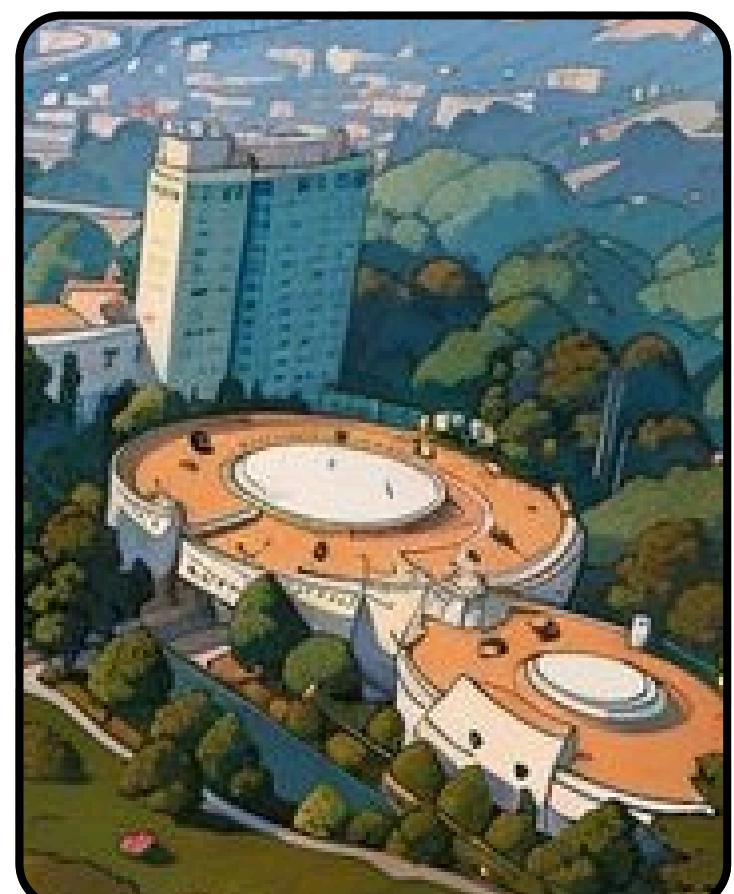
Mai Quốc Bảo - 21521850

Võ Thị Bích Ly - 21522317

Trần Kim Thanh - 21522605

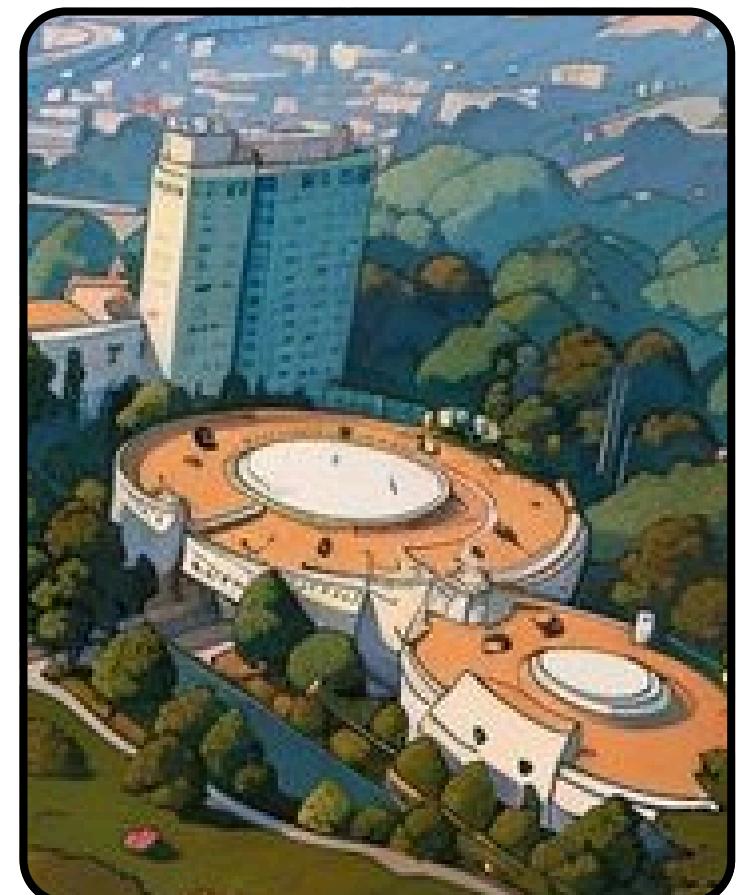
Hoàng Xuân Lộc - 22520788

Nguyễn Ngọc Gia Khiêm - 21520287



NỘI DUNG

- 1 Tổng quan
- 2 Mô hình giải bài toán
- 3 Kết luận và hướng phát triển



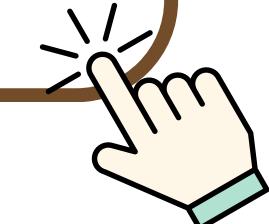


1.TỔNG QUAN

1. Tổng quan

Mục tiêu đề tài: Dự đoán điểm trung bình các kỳ học tiếp theo của sinh viên thông qua phân tích dữ liệu và thử nghiệm các mô hình hiệu quả. Nhóm tiến hành **tiền xử lý và thử nghiệm toàn diện để đánh giá cách tiếp cận tối ưu.**

- **Nhà trường:** Dự đoán kết quả học tập giúp cải thiện đào tạo và kịp thời xử lý các vấn đề học lực.
- **Doanh nghiệp:** Hỗ trợ nhà tuyển dụng chọn ứng viên tiềm năng, đáp ứng yêu cầu công việc.
- **Cố vấn và sinh viên:** Phát hiện nguy cơ giảm sút, giúp xây dựng kế hoạch học tập hiệu quả.

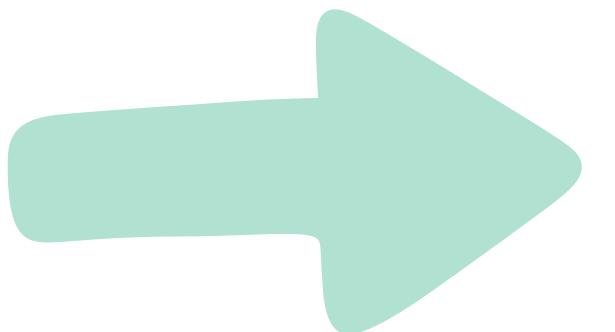


1. Tổng quan



Phát biểu bài toán

Đầu vào: Gồm thông tin về học kỳ, sinh viên và môn học, bao gồm danh sách môn học đã đăng ký, điểm số các kỳ trước và đặc điểm môn học.



Đầu ra: **Dự đoán điểm của sinh viên cho các môn trong kỳ học tiếp theo.**

1. Tổng quan



Thách thức bài toán

Xử lý dữ liệu:

- Dữ liệu sinh viên, môn học và điểm số thiếu, nhiễu hoặc bất thường.
- Thách thức: xử lý dữ liệu thiếu, thay thế giá trị không hợp lệ mà vẫn đảm bảo tính chính xác.



1. Tổng quan



Thách thức bài toán

Sự phụ thuộc vào ngữ cảnh:

- Điểm số sinh viên không chỉ phản ánh năng lực mà còn chịu ảnh hưởng bởi yếu tố ngữ cảnh như độ khó, nội dung môn học.
- Đánh giá cần xem xét cả đặc điểm sinh viên và môn học để xây dựng mô hình dự đoán hiệu quả, phản ánh đúng năng lực trong các ngữ cảnh học tập..



1. Tổng quan



Đối tượng phạm vi

- **Đối tượng:** Sinh viên trong hệ thống giáo dục với dữ liệu học tập chi tiết đủ để mô hình dự đoán.
- **Phạm vi:** Xây dựng mô hình dự đoán điểm số môn học, đánh giá độ chính xác và phân tích yếu tố ảnh hưởng, hỗ trợ nhà trường và sinh viên cải thiện kết quả học tập.
- **Giới hạn:** Chỉ tập trung vào dự đoán điểm môn học, không bao gồm quản lý chương trình hay thay đổi môn học.

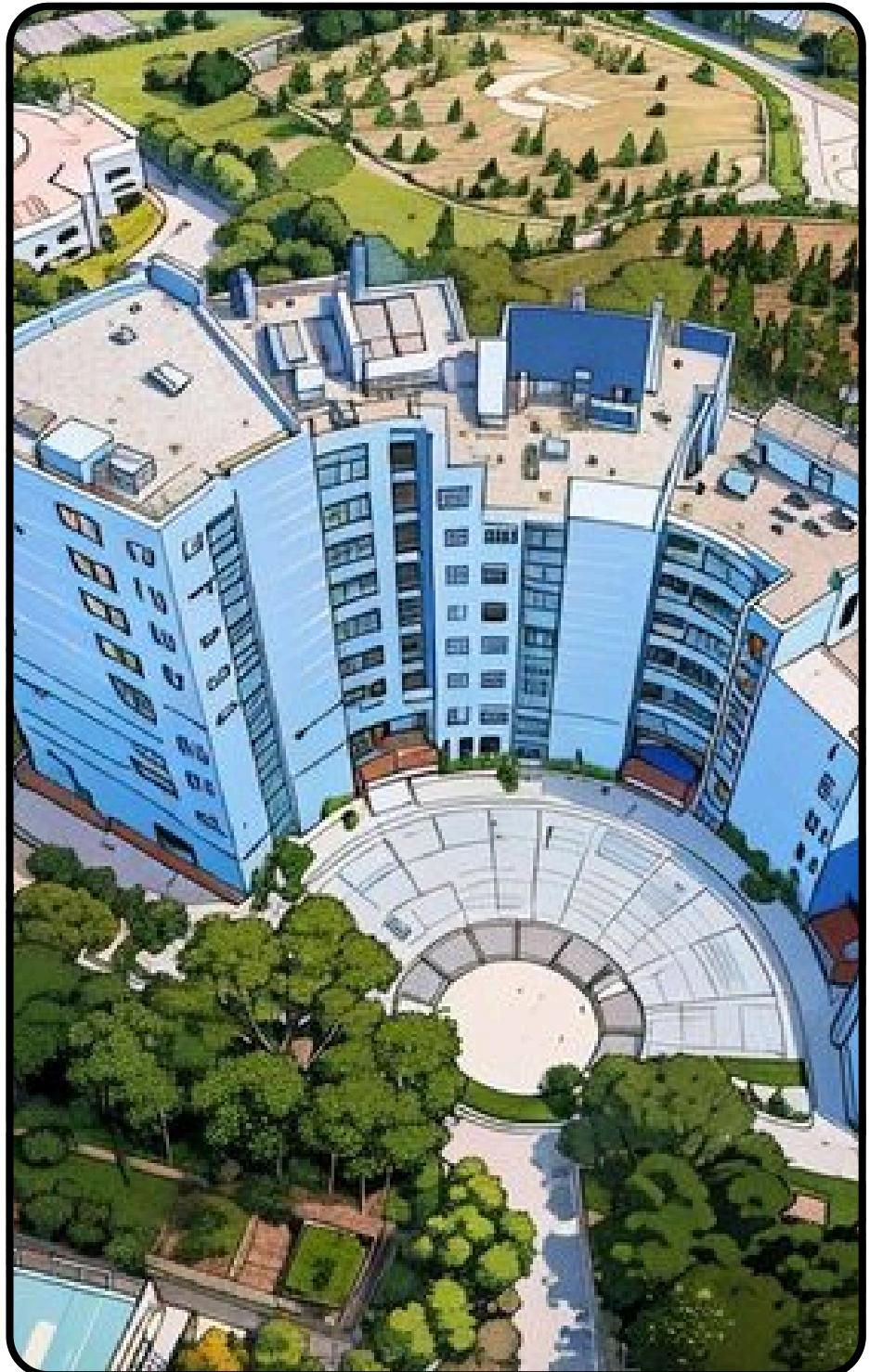
1. Tổng quan



Mục tiêu



- Nghiên cứu và áp dụng kỹ thuật Machine Learning, Deep Learning, và hệ thống gợi ý để dự đoán kết quả học tập.
- Phân tích ưu, nhược điểm và tối ưu hóa sự kết hợp giữa các kỹ thuật.
- Áp dụng trọng số cho đặc trưng học tập để xác định yếu tố ảnh hưởng lớn nhất và tăng độ chính xác mô hình.



2. MÔ HÌNH GIẢI BÀI TOÁN



2.1 Tiền xử lý dữ liệu

Tiền xử lý tổng quát

Đối với mỗi tệp dữ liệu định dạng Excel trong , thực hiện lần lượt các bước sau để làm sạch và cải thiện chất lượng dữ liệu:

- Xóa các khoảng trắng ở tên cột (header).
- Loại bỏ các dữ liệu trùng lặp bằng phương pháp drop_duplicates - Xóa các cột không cần thiết
- Loại bỏ các kí tự đặc biệt và giá trị rỗng

Mô hình giải bài toán

2.1 Tiền xử lý dữ liệu

Bảng sinhvien.xlsx

Các bước tiền xử lý bảng sinh viên

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['id', 'namsinh', 'noisinh', 'diachi_tinhtp', 'tinhtrang']

```
for file in files:  
    if file in '01.sinhvien.xlsx':  
        df = pd.read_excel(file)  
        tmp = ['id', 'namsinh', 'noisinh', 'diachi_tinhtp', 'tinhtrang']  
        df.drop(columns=tmp, inplace=True, errors='ignore')  
        df.info()  
        df.to_excel(file, index=False)  
        break
```

Mô hình giải bài toán

2.1 Tiềng xử lý dữ liệu

Bảng sinhvien.xlsx

Xây dựng hàm làm sạch tên và chuẩn hóa về định dạng chung

- Kiểm tra giá trị rỗng: Nếu name là NaN hoặc chuỗi rỗng, hàm trả về "".
- Loại bỏ khoảng trắng ở đầu và cuối chuỗi.
- Loại bỏ dấu và chuẩn hóa chữ cái đầu.
- Loại bỏ dấu ngoặc đơn và ngoặc kép.

```
def cleaning(name):  
    if pd.isna(name) or name.strip() == "":  
        return ""  
    name = name.strip()  
    name = unidecode.unidecode(name).title()  
    name = name.replace("'", "").replace("’", '')  
    return name
```

Mô hình giải bài toán

2.1 Tiền xử lý dữ liệu

Bảng sinhvien.xlsx

- **Làm sạch tên:** Sử dụng cleaning(name) để chuẩn hóa tên.
- **Kiểm tra tên rỗng:** Nếu kết quả là chuỗi rỗng, trả về None.
- **Tìm tên gần khớp:** Dùng process.extractOne(cleaned_name, mapp_flat) để tìm tên có độ tương đồng cao nhất, trả về match và score.
- **Lọc theo điểm khớp:** Nếu score ≥ 50 , trả về match; ngược lại, trả về None

```
def get_closest_match(name):  
    cleaned_name = cleaning(name)  
    if not cleaned_name:  
        return None  
    match, score = process.extractOne(cleaned_name, mapp_flat)  
    return match if score >= 50 else None
```

Mô hình giải bài toán

2.1 Tiền xử lý dữ liệu

Bảng sinhvien.xlsx

Bước	Mô tả
Đọc dữ liệu	Đọc tệp Excel trong danh sách <code>files</code> vào DataFrame <code>df</code> .
One-hot encoding	Tạo cột mới với tiền tố <code>hedt_</code> và <code>khoa_</code> nếu có cột <code>hedt</code> và <code>khoa</code> .
Làm sạch tên	Áp dụng hàm <code>cleaning</code> để làm sạch giá trị cột <code>noisinh</code> .
Xác định khu vực	Sử dụng hàm <code>get_closest_match</code> để gán khu vực (<code>khuvuc</code>) cho <code>noisinh</code> .
Ánh xạ ngược	Dùng <code>reverse_mapping</code> (từ điển) để chuyển giá trị khu vực về mã số.
Gán giá trị mặc định	Thay <code>NaN</code> trong <code>khuvuc</code> bằng <code>-1</code> , đổi kiểu thành <code>Int64</code> .
Kiểm tra lỗi	In các giá trị <code>noisinh</code> không xác định được <code>khuvuc</code> (<code>khuvuc = -1</code>).
Lưu kết quả	Lưu DataFrame <code>df</code> đã xử lý trở lại tệp Excel ban đầu.

```
for file in files:
    df = pd.read_excel(file)

    # One hot encoding

    if 'hedt' in df.columns:
        df = pd.get_dummies(df, columns=['hedt'], prefix='hedt')
    if 'khoa' in df.columns:
        df = pd.get_dummies(df, columns=['khoa'], prefix='khoa')

    if 'noisinh' in df.columns:
        df['noisinh'] = df['noisinh'].apply(cleaning)
        df['khuvuc'] = df['noisinh'].apply(get_closest_match)
        df['khuvuc'] = df['khuvuc'].map(reverse_mapping)
        df['khuvuc'] = df['khuvuc'].fillna(-1).astype('Int64')
        unexpected_values = df[df['khuvuc'] == -1]['noisinh'].unique()
        if unexpected_values.size > 0:
            print(f"Test Error: {unexpected_values}")
    df.to_excel(file, index=False)
```

Mô hình giải bài toán

2.1 Tiền xử lý dữ liệu

Bảng xeploaiav.xlsx

Các bước tiền xử lý bảng xếp loại anh văn:

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['id', 'listening', 'reading', 'mamh', 'ghichu']

```
for file in files:  
    if file in '04.xeploaiav.xlsx':  
        df = pd.read_excel(file)  
        tmp = ['id', 'listening', 'reading', 'mamh', 'ghichu']  
        df.drop(columns=tmp, inplace=True, errors='ignore')  
        df.info()  
        df.to_excel(file, index=False)  
    break
```

2.1 Tiền xử lý dữ liệu

Bảng xeploaiav.xlsx

Tạo cột phân loại xl_av dựa trên tổng điểm như sau:

- Dưới 40: gán xl_av là 0.
- Từ 40 đến dưới 60: gán xl_av là 1.
- Từ 60 đến dưới 70: gán xl_av là 2.
- Từ 70 đến dưới 80: gán xl_av là 3.
- Từ 80 đến dưới 90: gán xl_av là 4.
- Từ 90 trở lên: gán xl_av là 5

```
for index, row in df.iterrows():
    total = row['total']
    if total < 40:
        df.at[index, 'xl_av'] = 0
    elif 40 <= total < 60:
        df.at[index, 'xl_av'] = 1
    elif 60 <= total < 70:
        df.at[index, 'xl_av'] = 2
    elif 70 <= total < 80:
        df.at[index, 'xl_av'] = 3
    elif 80 <= total < 90:
        df.at[index, 'xl_av'] = 4
    elif total >= 90:
        df.at[index, 'xl_av'] = 5
```

Mô hình giải bài toán

2.1 Tiền xử lý dữ liệu

Bảng thisinh.xlsx

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['lop12_matinh', 'lop12_matruong', 'TEN_TRUONG', 'dien_tt', 'diem_tt']
- Tạo thêm một cột xeploai_tt dựa trên điểm thi DGNL hoặc THPT

```
for file in files:  
    if file in '05.Thisinh.xlsx':  
        df = pd.read_excel(file)  
        tmp = ['lop12_matinh', 'lop12_matruong', 'TEN_TRUONG', 'dien_tt', 'diem_tt']  
        df.drop(columns=tmp, inplace=True, errors='ignore')  
        df.info()  
        df.to_excel(file, index=False)  
        break
```

```
if 'dien_tt' in df.columns and 'diem_tt' in df.columns:  
    df['xeploai_tt'] = None  
  
for index, row in df.iterrows():  
    if row['dien_tt'] == 'DGNL':  
        dgnl_score = row['diem_tt']  
        if dgnl_score < 600:  
            df.at[index, 'xeploai_tt'] = 0  
        elif 600 <= dgnl_score < 750:  
            df.at[index, 'xeploai_tt'] = 1  
        elif 750 <= dgnl_score < 900:  
            df.at[index, 'xeploai_tt'] = 2  
        elif 900 <= dgnl_score < 1000:  
            df.at[index, 'xeploai_tt'] = 3  
        elif dgnl_score >= 1000:  
            df.at[index, 'xeploai_tt'] = 4  
    else:  
        thpt_score = row['diem_tt']  
        if thpt_score < 20:  
            df.at[index, 'xeploai_tt'] = 0  
        elif 20 <= thpt_score < 22:  
            df.at[index, 'xeploai_tt'] = 1  
        elif 22 <= thpt_score < 24:  
            df.at[index, 'xeploai_tt'] = 2  
        elif 24 <= thpt_score < 26:  
            df.at[index, 'xeploai_tt'] = 3  
        elif thpt_score >= 26:  
            df.at[index, 'xeploai_tt'] = 4
```

2.1 Tiền xử lý dữ liệu

Bảng diemrl.xlsx

- Load dữ liệu: Tải dữ liệu Excel vào một DataFrame
- Xóa các cột không cần thiết ['id', 'ghichu', 'drl']

```
for file in files:  
    if file in '10.diemrl.xlsx':  
        df = pd.read_excel(file)  
        tmp = ['id','ghichu','drl']  
        df.drop(columns=tmp, inplace=True, errors='ignore')  
        df.info()  
        df.to_excel(file, index=False)  
        break
```

2.1 Tiền xử lý dữ liệu

Bảng diemrl.xlsx

Tạo cột phân loại điểm rèn luyện drltl dựa trên tổng điểm như sau:

- Dưới 35: gán drltl là 0.
- Từ 35 đến dưới 50: gán drltl là 1.
- Từ 50 đến dưới 65: gán drltl là 2.
- Từ 65 đến dưới 80: gán drltl là 3.
- Từ 80 đến dưới 90: gán drltl là 4.
- Từ 90 trở lên: gán drltl là 5.

```
for index, row in df.iterrows():
    drl_score = row['drl']
    if drl_score < 35:
        df.at[index, 'drltl'] = 0
    elif 35 <= drl_score < 50:
        df.at[index, 'drltl'] = 1
    elif 50 <= drl_score < 65:
        df.at[index, 'drltl'] = 2
    elif 65 <= drl_score < 80:
        df.at[index, 'drltl'] = 3
    elif 80 <= drl_score < 90:
        df.at[index, 'drltl'] = 4
    elif drl_score >= 90:
        df.at[index, 'drltl'] = 5
```

2.1 Tiền xử lý dữ liệu

Bảng diemrl.xlsx

Kết hợp 2 bảng drl

```
diemrl = pd.read_excel('diemrl.xlsx')
diemrl2 = pd.read_excel('10.diemrl.xlsx')

if 'mssv' in diemrl.columns and 'mssv' in diemrl2.columns:
    combined_df = pd.concat([diemrl, diemrl2], ignore_index=True)

combined_df.info()

combined_df.to_excel('combined_diemrl.xlsx', index=False)
```

Mô hình giải bài toán

2.1 Tiền xử lý dữ liệu

Bảng dữ liệu sau xử lý

STT	Tên thuộc tính	Ý nghĩa thuộc tính	Kiểu dữ liệu	Ghi chú
1	mssv	Mã số sinh viên	object	
2	gioitinh	Giới tính	float64	1: nam 0: nữ
3	lopsh	Lớp sinh hoạt	object	
4	khoahoc	Khóa học	float64	
5	chuyennganh2	Chuyên ngành 2	object	
6	hedt_CLC	Hệ đào tạo CLC	bool	
7	hedt_CNTN	Hệ đào tạo CNTN	bool	
8	hedt_CQUI	Hệ đào tạo chính quy	bool	
9	hedt_CTTT	Hệ đào tạo CTTT	bool	
10	hedt_KSTN	Hệ đào tạo KSTN	bool	

11	khoa_CNPM	Khoa CNPM	bool	
12	khoa_HTTT	Khoa HTTT	bool	
13	khoa_KHMT	Khoa KHMT	bool	
14	khoa_KTMT	Khoa KTPM	bool	
15	khoa_KTTT	Khoa KTTT	bool	
16	khoa_MMT&TT	Khoa MMT&TT	bool	
17	khuvuc	Khu vực sinh sống của sinh viên	float64	
18	xl_av	Xếp loại anh văn đầu vào của mỗi sinh viên	float64	Sinh viên được xếp loại anh văn đầu vào dựa vào kết quả bài thi anh văn đầu vào ở trường hoặc các chứng chỉ tiếng Anh khác. Mức độ tiếng Anh đầu vào được đánh giá theo mức độ tăng dần từ 0 → 5

Mô hình giải bài toán

2.1 Tiền xử lý dữ liệu

Bảng dữ liệu sau xử lý

19	xeploai_tt	Xếp loại trúng tuyển của sinh viên	float64	Sinh viên được xếp loại dựa trên diện trúng tuyển và kết quả đầu vào, được chia theo giá trị từ 0 \rightarrow 4
20	hocky	Học kỳ	float64	
21	namhoc	Năm học	float64	
22	drltl	Điểm rèn luyện tích lũy	float64	Đánh giá theo 6 mức tăng dần từ 0 \rightarrow 5.
23	dtbhk	Điểm trung bình học kỳ	float64	
24	sotchk	Số tín chỉ học kỳ	float64	
25	dtbhk_truoc	Điểm trung bình học kỳ trước	float64	

2.2 Khám phá dữ liệu

Chuyển thuộc tính dtbhk thành thuộc tính đầu ra ‘xeploai’ như sau:

- Dưới 5: gán xeploai là 0.
- Từ 5 đến dưới 6.5: gán xeploai là 1.
- Từ 6.5 đến dưới 8: gán xeploai là 2.
- Từ 8 đến dưới 9: gán xeploai là 3.
- Từ 9 trở lên: gán drltl là 4.

```
def danh_gia_diem(score):
    if score < 5:
        return 0
    elif 5 <= score < 6.5:
        return 1
    elif 6.5 <= score < 8:
        return 2
    elif 8 <= score < 9:
        return 3
    else:
        return 4
```

```
df['xeploai'] = df['dtbhk'].apply(danh_gia_diem)
```

2.2 Khám phá dữ liệu

Gộp các thuộc tính hệ đào tạo và khoa thành 1 thuộc tính:

- Thuộc tính hệ đào tạo:
- Thuộc tính khoa:

```
def determine_hedt(row):  
    if row['hedt_CLC']:  
        return 'CLC'  
    elif row['hedt_CNTN']:  
        return 'CNTN'  
    elif row['hedt_CQUI']:  
        return 'CQUI'  
    elif row['hedt_CTTT']:  
        return 'CTTT'  
    else:  
        return 'other'
```

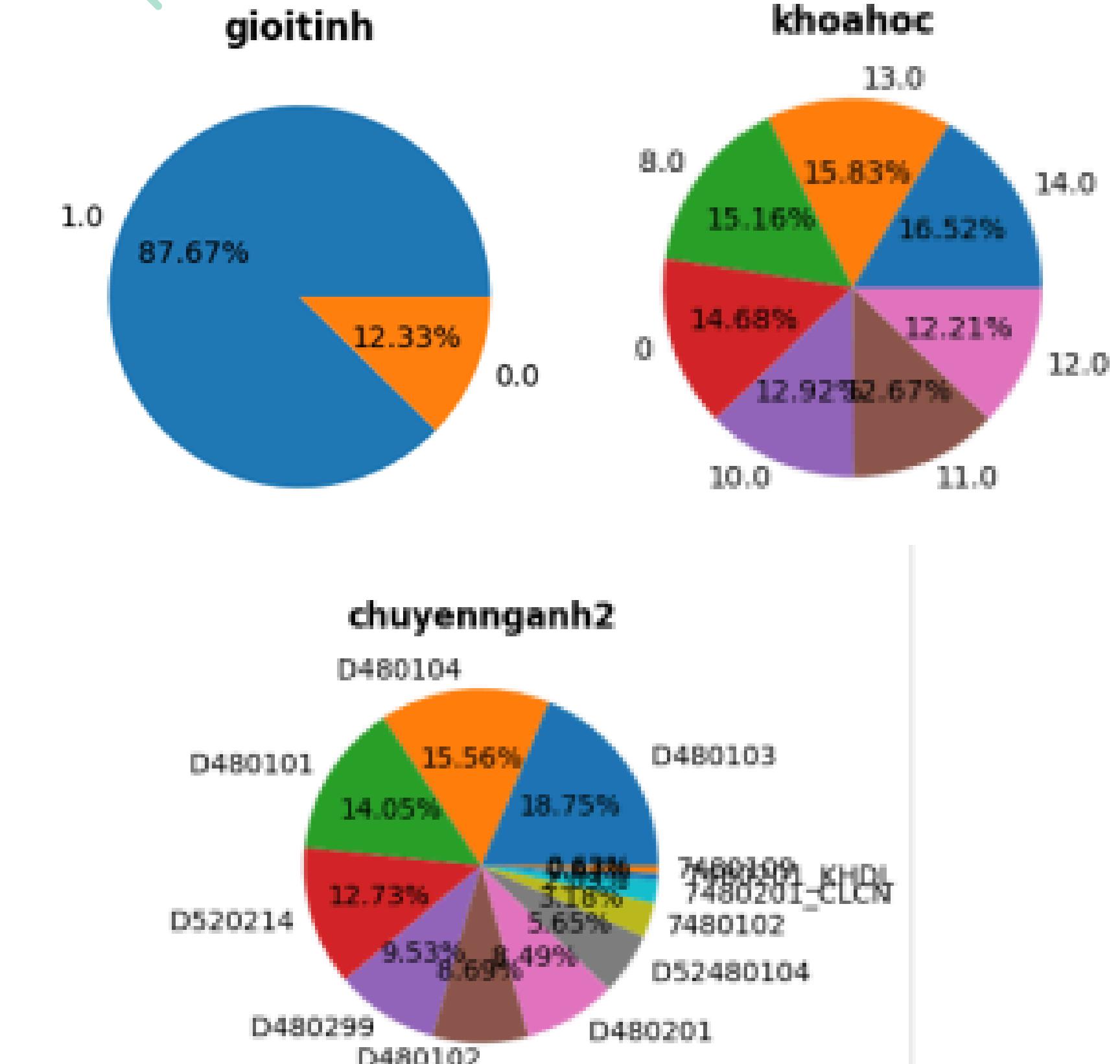
```
def determine_khoa(row):  
    for col in row.index:  
        if 'khoa_' in col and row[col] == True:  
            return col.replace('khoa_', '').strip()  
    return 'Other'
```

2.2 Khám phá dữ liệu

Phân tích đơn biến

a. Thuộc tính phân loại

- Giới tính:** 87.67% một giới tính, 12.33% còn lại, cho thấy sự mất cân bằng trong dữ liệu.
- Khóa học:** Phân bổ khá đồng đều, một số khóa chiếm tỷ lệ cao hơn.
- Chuyên ngành 2:** Nhiều nhãn nhỏ, phản ánh sự đa dạng trong chuyên ngành sinh viên chọn.



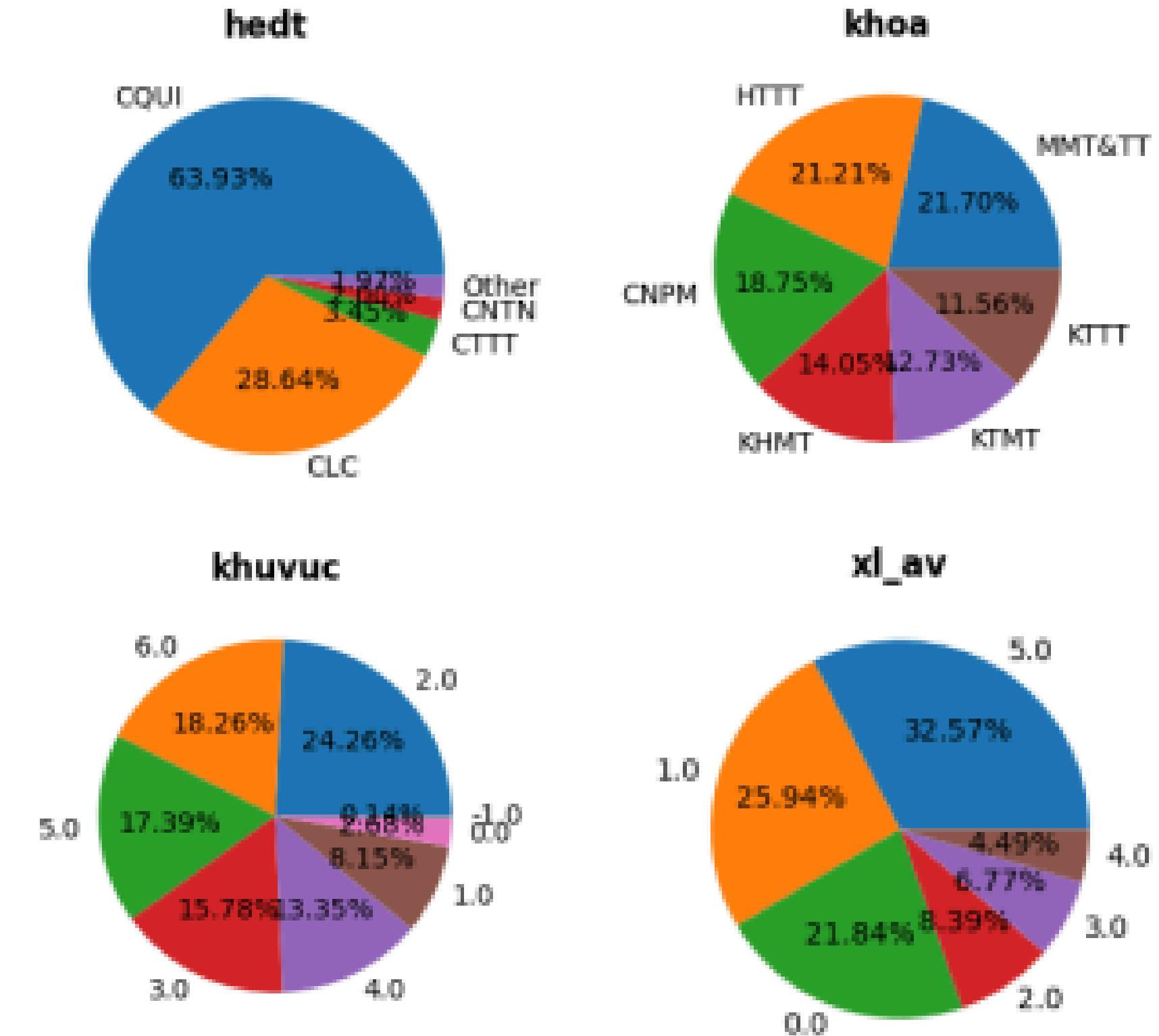
Mô hình giải bài toán

2.2 Khám phá dữ liệu

Phân tích đơn biến

a. Thuộc tính phân loại

- **Hệ đào tạo:** CQUI chiếm 63.93%, CLC 28.64%, phần lớn sinh viên thuộc hai hệ chính.
- **Khoa:** Phân bổ đồng đều nhưng HHT, MMT&TT, CNPM chiếm tỷ lệ cao hơn.
- **Khu vực:** Phân bố đều, khu vực 1 và 2 chiếm tỷ lệ lớn, phản ánh sự đa dạng địa lý.
- **Xếp loại AV:** Đồng đều, tập trung vào mức cao (5.0, 4.5), phản ánh trình độ ngoại ngữ tốt.

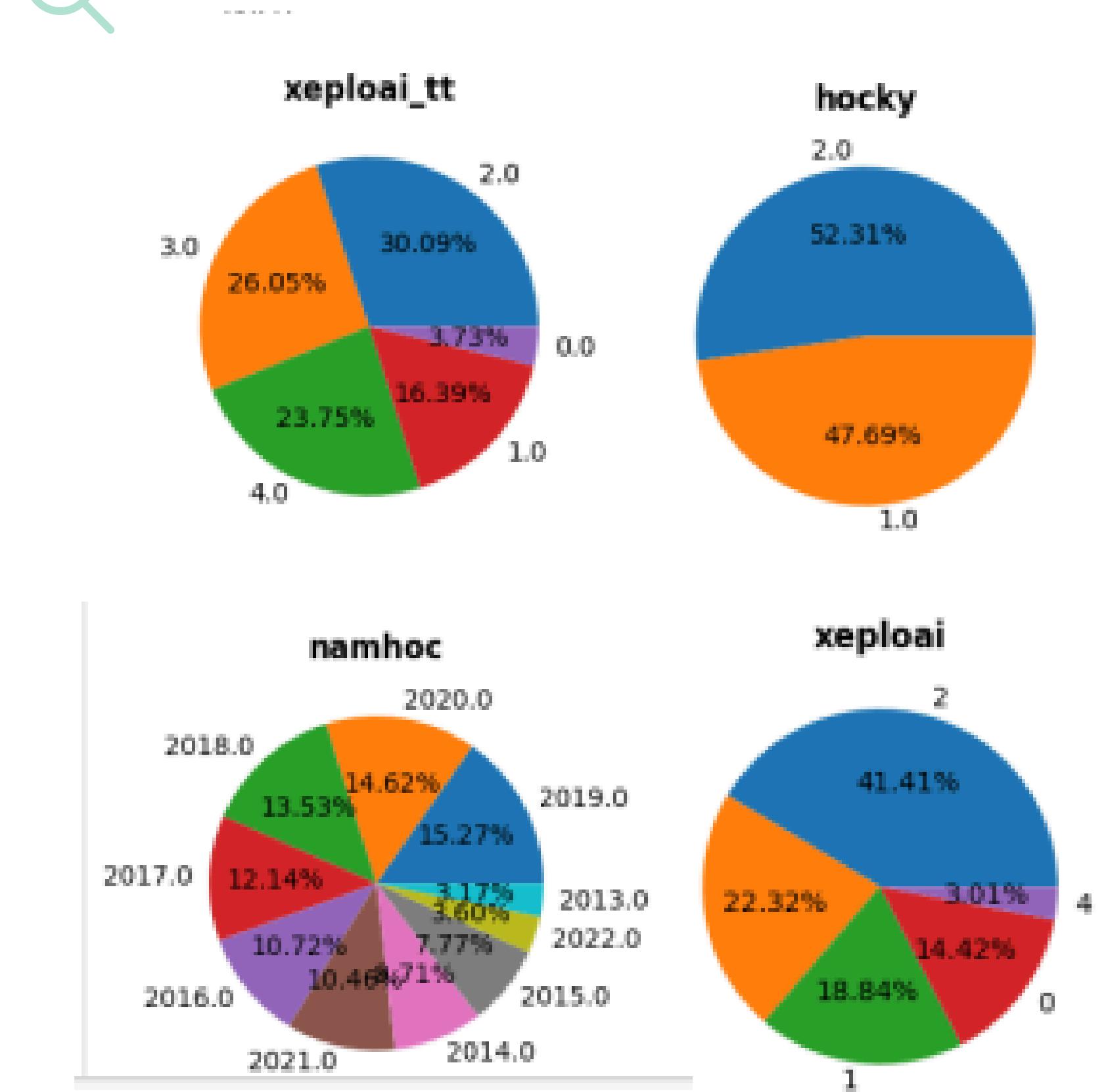


2.2 Khám phá dữ liệu

Phân tích đơn biến

a. Thuộc tính phân loại

- Xếp loại TT: Đồng đều, mức 3.0 và 2.0 chiếm tỷ lệ lớn nhất, thể hiện sự đa dạng thành tích.
- Học kỳ: Học kỳ 2 chiếm 52.31%, do khảo sát diễn ra vào học kỳ 1.
- Năm học: Từ 2017-2022, các năm gần đây có tỷ lệ cao hơn do gia tăng sinh viên mới.
- Xếp loại: Đa dạng, mức 0 (khá) chiếm 41.41%, mức 2 (xuất sắc) chỉ 3.01%, gây mất cân bằng dữ liệu.



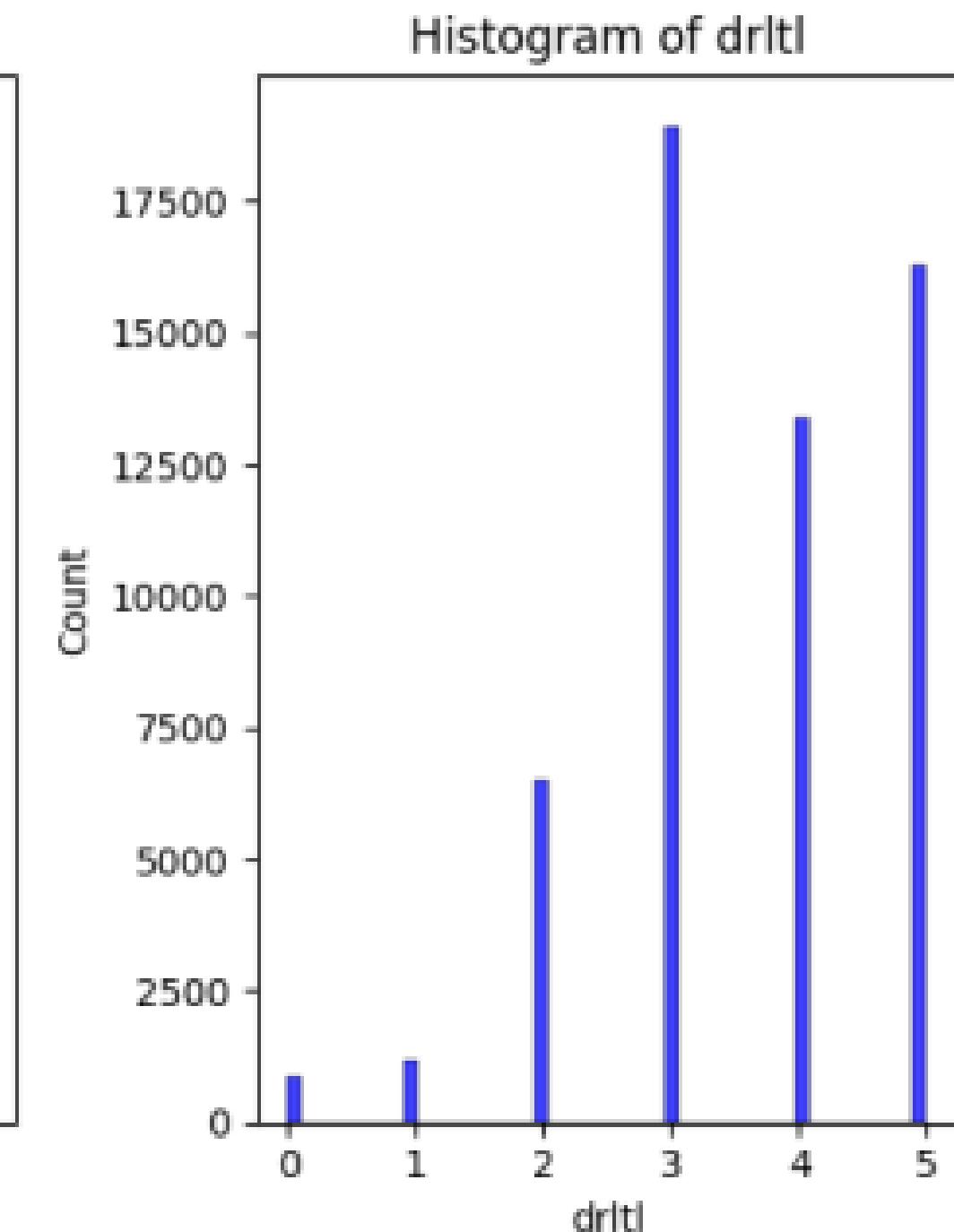
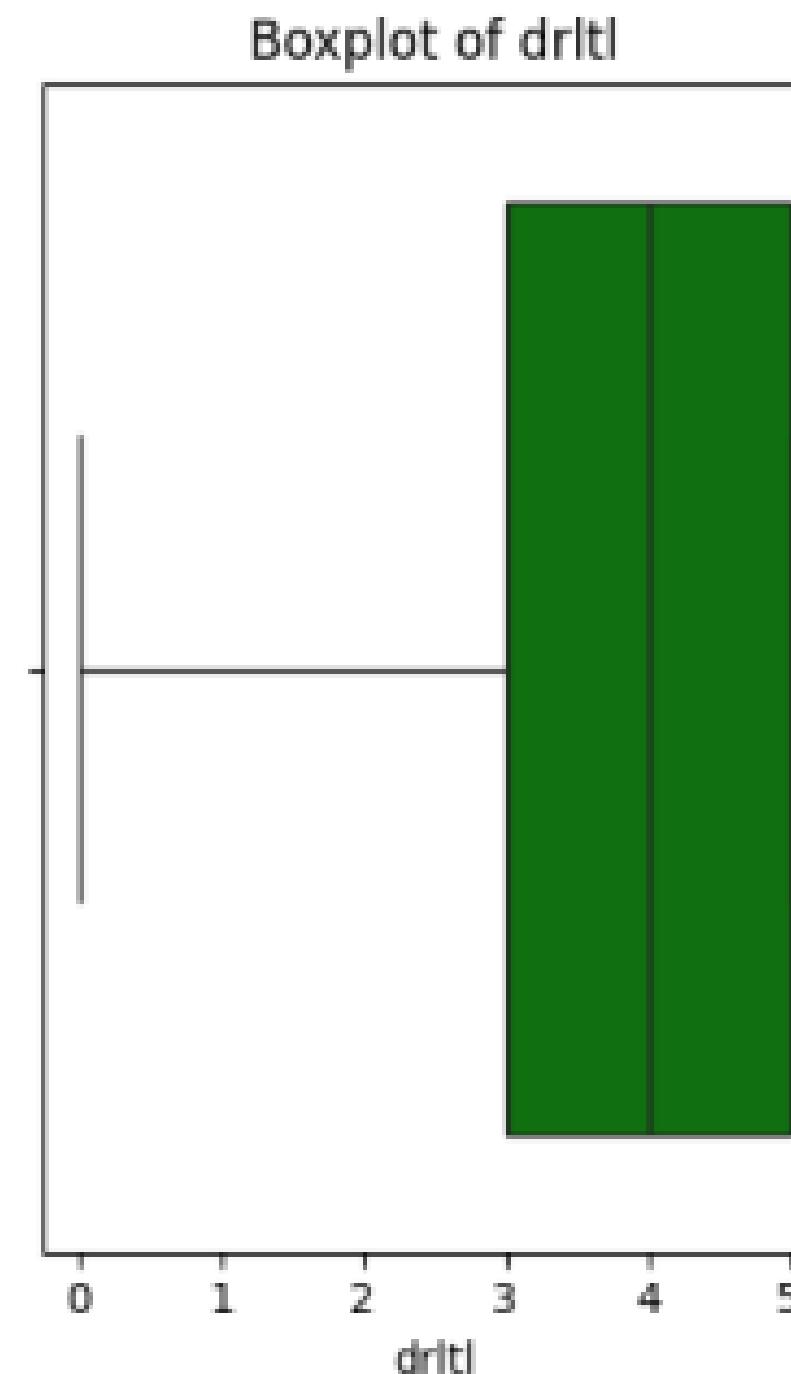
2.2 Khám phá dữ liệu

Phân tích đơn biến

b. Thuộc tính tuyển tính

Tiến hành vẽ boxplot và histogram: - **Thuộc tính drltl**

- **Boxplot:** Điểm rèn luyện tích lũy từ 3-5, phần lớn sinh viên đạt kết quả tốt, không có giá trị bất thường.
- **Histogram:** Điểm cao (gần 5) phổ biến, sinh viên giảm dần ở điểm thấp, phản ánh nỗ lực và kỷ luật.



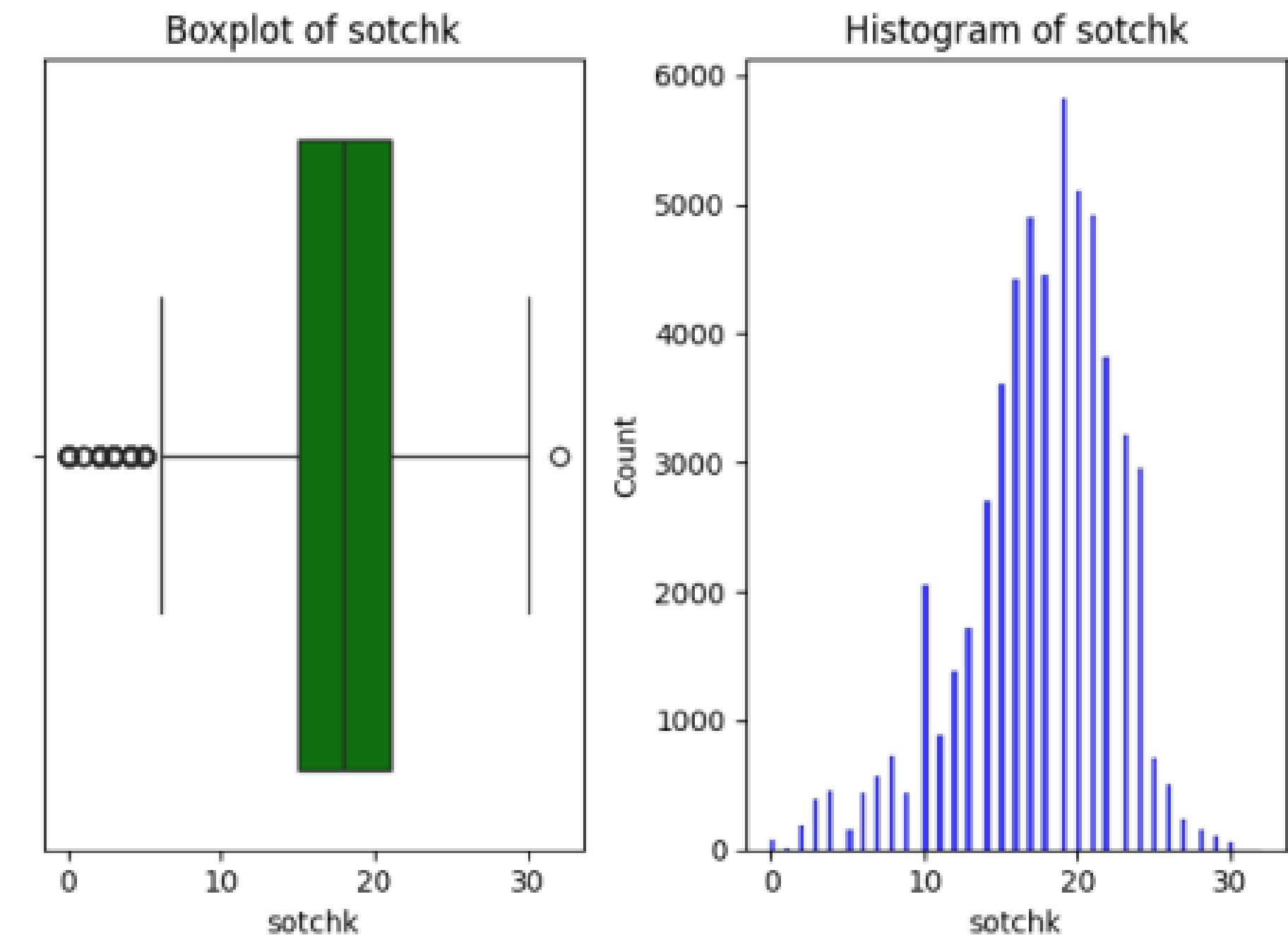
2.2 Khám phá dữ liệu

Phân tích đơn biến

b. Thuộc tính tuyến tính

Tiến hành vẽ boxplot và histogram: **Thuộc tính sotchk**

- **Boxplot:** Một số ngoại lệ dưới mức tín chỉ bình thường, phần lớn đăng ký 10-30 tín chỉ, đảm bảo tiến độ học tập.
- **Histogram:** Phân bố hình chuông, sinh viên chủ yếu đăng ký tín chỉ trung bình, giảm dần ở mức quá cao/thấp.



2.2 Khám phá dữ liệu

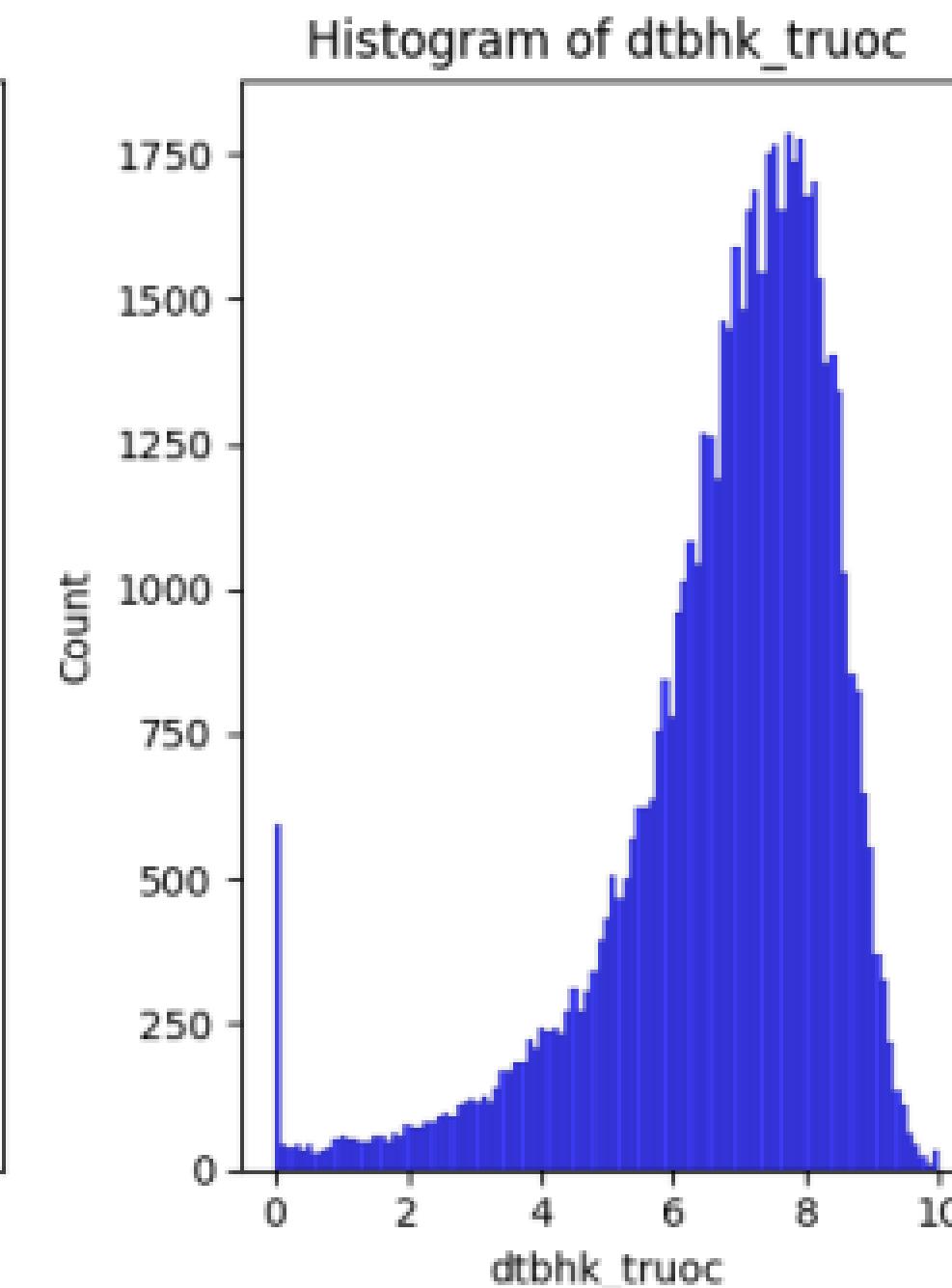
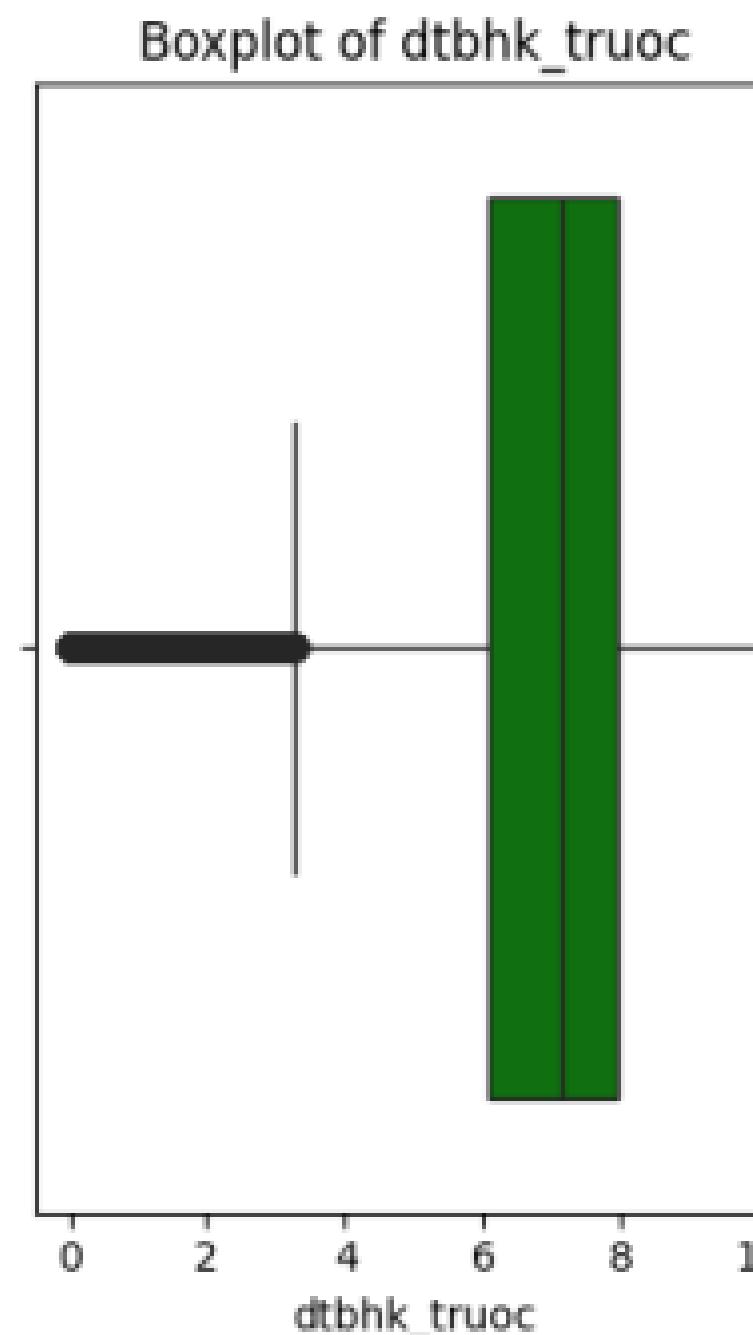
Phân tích đơn biến

b. Thuộc tính tuyến tính

Tiến hành vẽ boxplot và histogram:

Thuộc tính dtbhk_truoc

- **Boxplot:** Một số điểm ngoại lệ dưới trung bình, phần lớn điểm từ 5-8, đảm bảo tiến bộ học tập.
- **Histogram:** Lệch phải, điểm 6-8 phổ biến, giảm dần ngoài khoảng này, phản ánh nỗ lực ổn định.



2.2 Khám phá dữ liệu

Trích chọn đặc trưng

- **Filter Method:** Dựa trên các chỉ số thống kê hoặc độ liên kết để đánh giá tầm quan trọng của đặc trưng, bao gồm:
 - Correlation Matrix
 - Mutual Information
 - Chi-Square Test
- **Wrapper Method:** Chọn đặc trưng dựa trên hiệu suất của mô hình học máy thông qua sử dụng SelectKBest để chọn k đặc trưng tốt nhất.

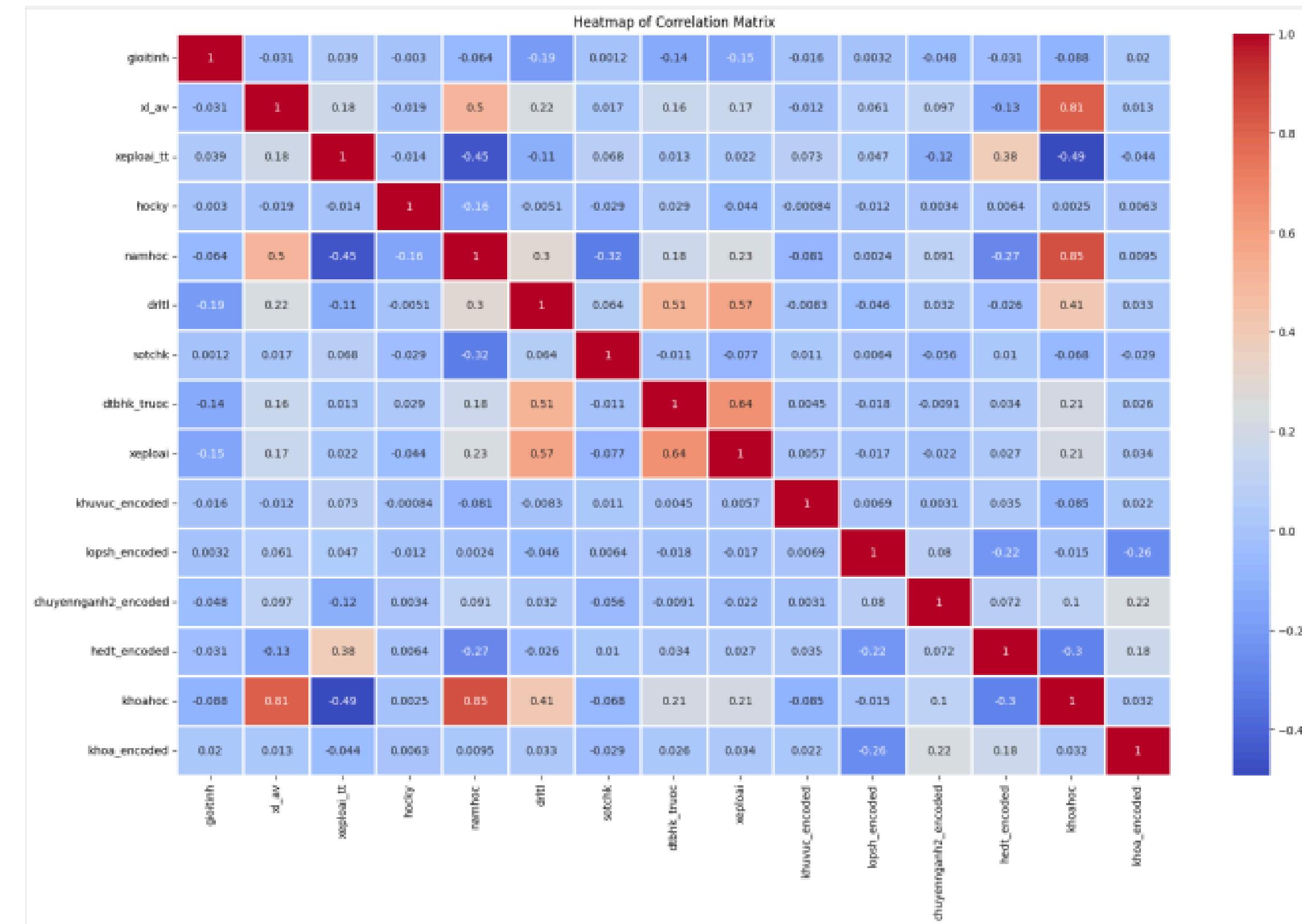
Mô hình giải bài toán

2.2 Khám phá dữ liệu

Trích chọn đặc trưng

Quan sát các thuộc tính qua ma trận tương quan (Correlation Matrix)

- Thuộc tính khuvuc (mức độ tương quan 0.0057) => loại bỏ cột khuvuc
 - Các thuộc tính đầu vào: namhoc, khoahoc, xlav có độ tương quan cao (>0.5), chọn thuộc tính namhoc do các thuộc tính còn lại dư thừa và không ảnh hưởng đáng kể đến dự đoán.



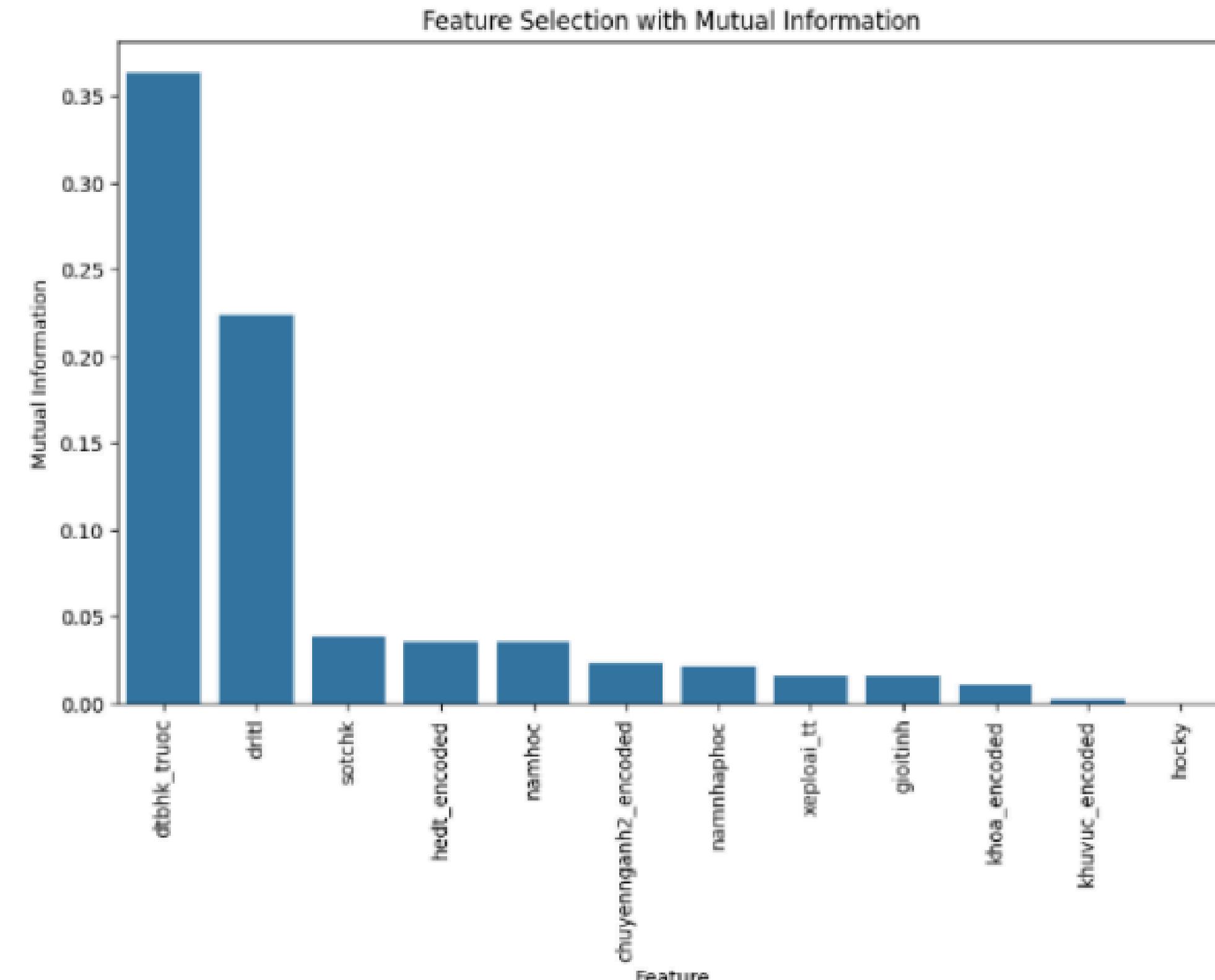
2.2 Khám phá dữ liệu

Trích chọn đặc trưng

Dùng **Mutual Information** chọn đặc trưng liên quan mạnh với nhãn và loại bỏ thuộc tính dư thừa.

- dtbhk_truoc: Có mối quan hệ mạnh mẽ với nhãn xeploai, là yếu tố quyết định.
- Các thuộc tính khác: drltl, sotchk, hedt, namhoc có MI > 0.2, có ảnh hưởng nhất định nhưng yếu hơn.

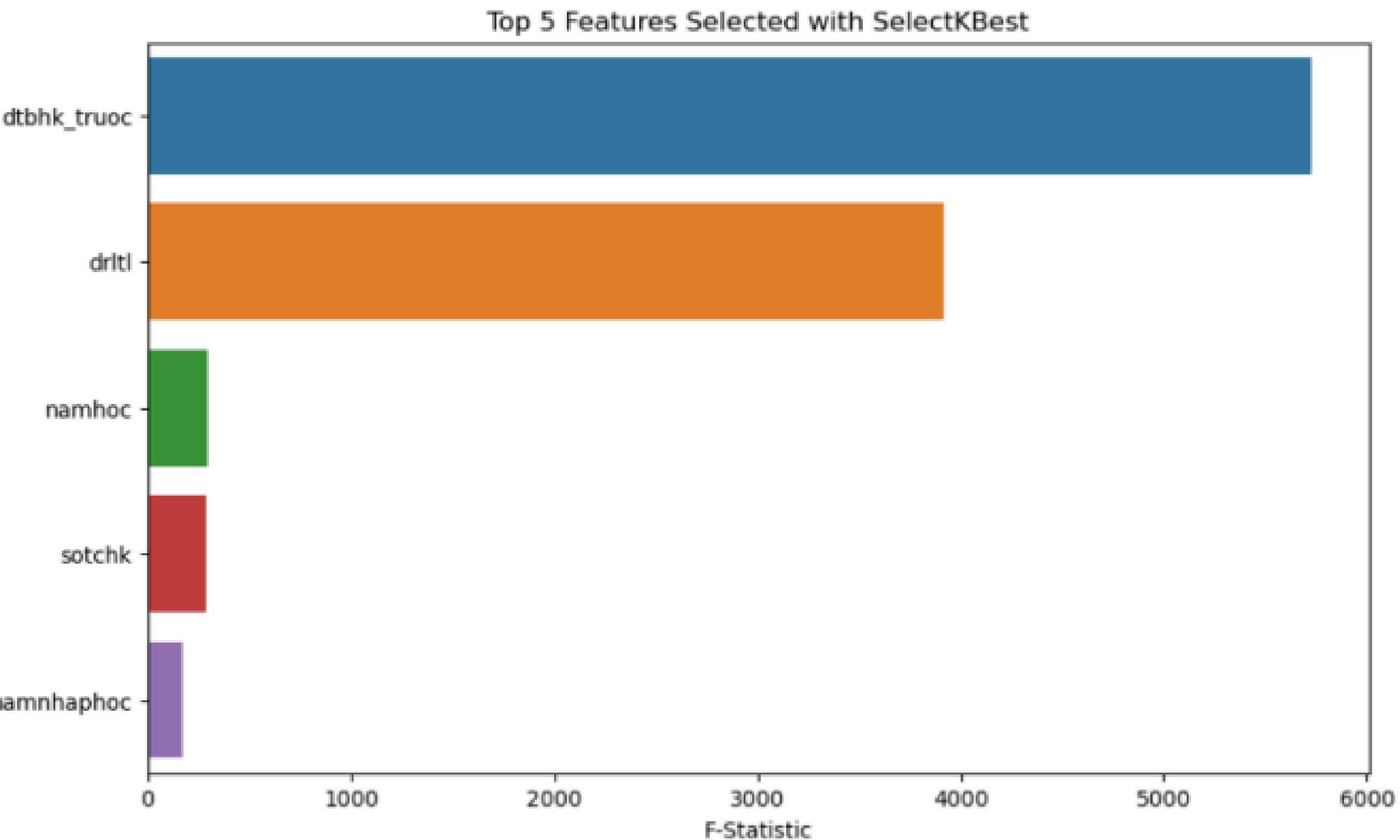
=> **Kết luận:** Các thuộc tính dư thừa có MI thấp có thể bị loại bỏ để cải thiện hiệu quả mô hình.



2.2 Khám phá dữ liệu

Trích chọn đặc trưng

Nhóm sử dụng phương pháp
trích chọn đặc trưng Select K
Best cũng cho ra kết quả
tương tự

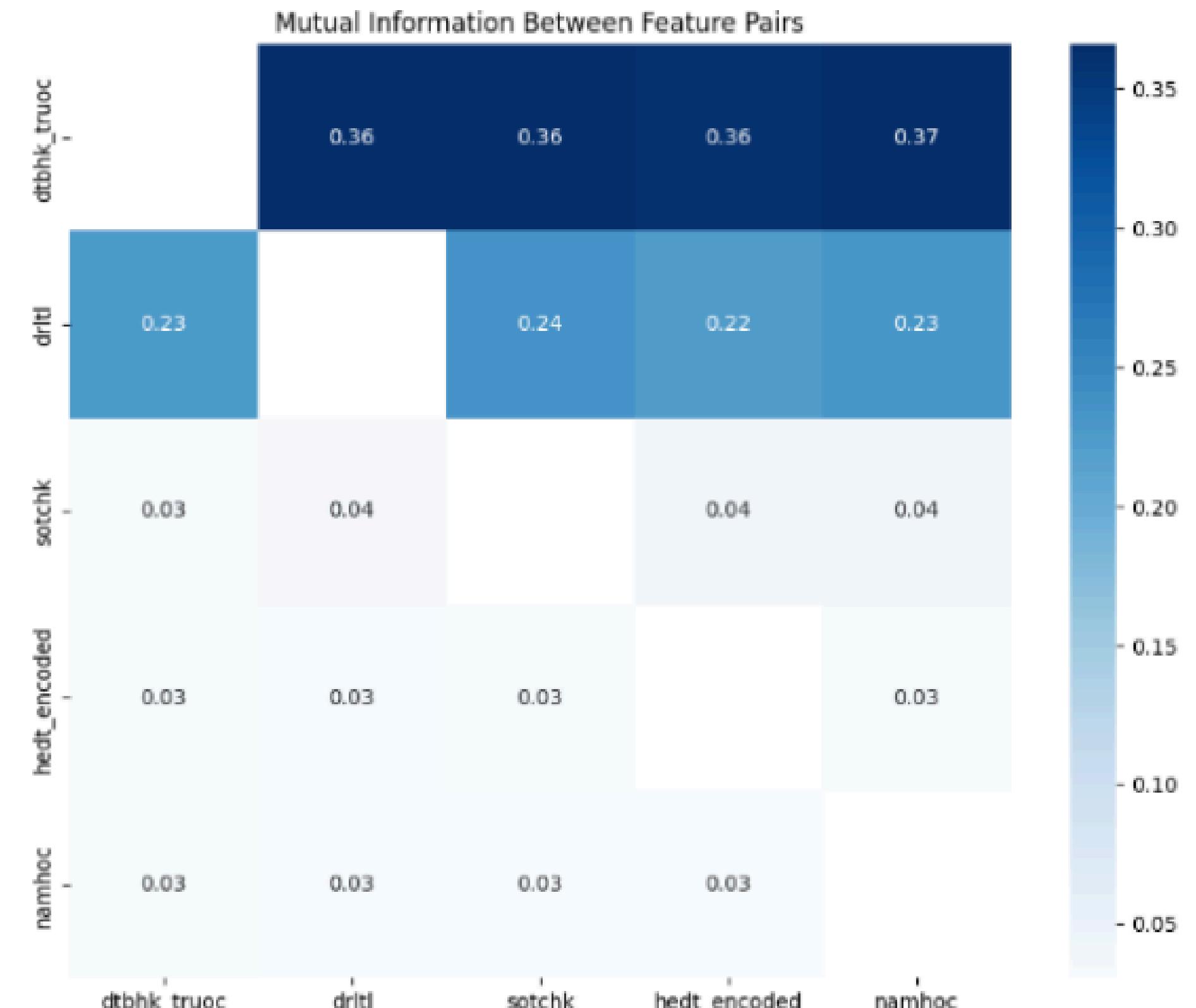


Mô hình giải bài toán

2.3 Mối quan hệ giữa các thuộc tính

Xây dựng đồ thị mạng

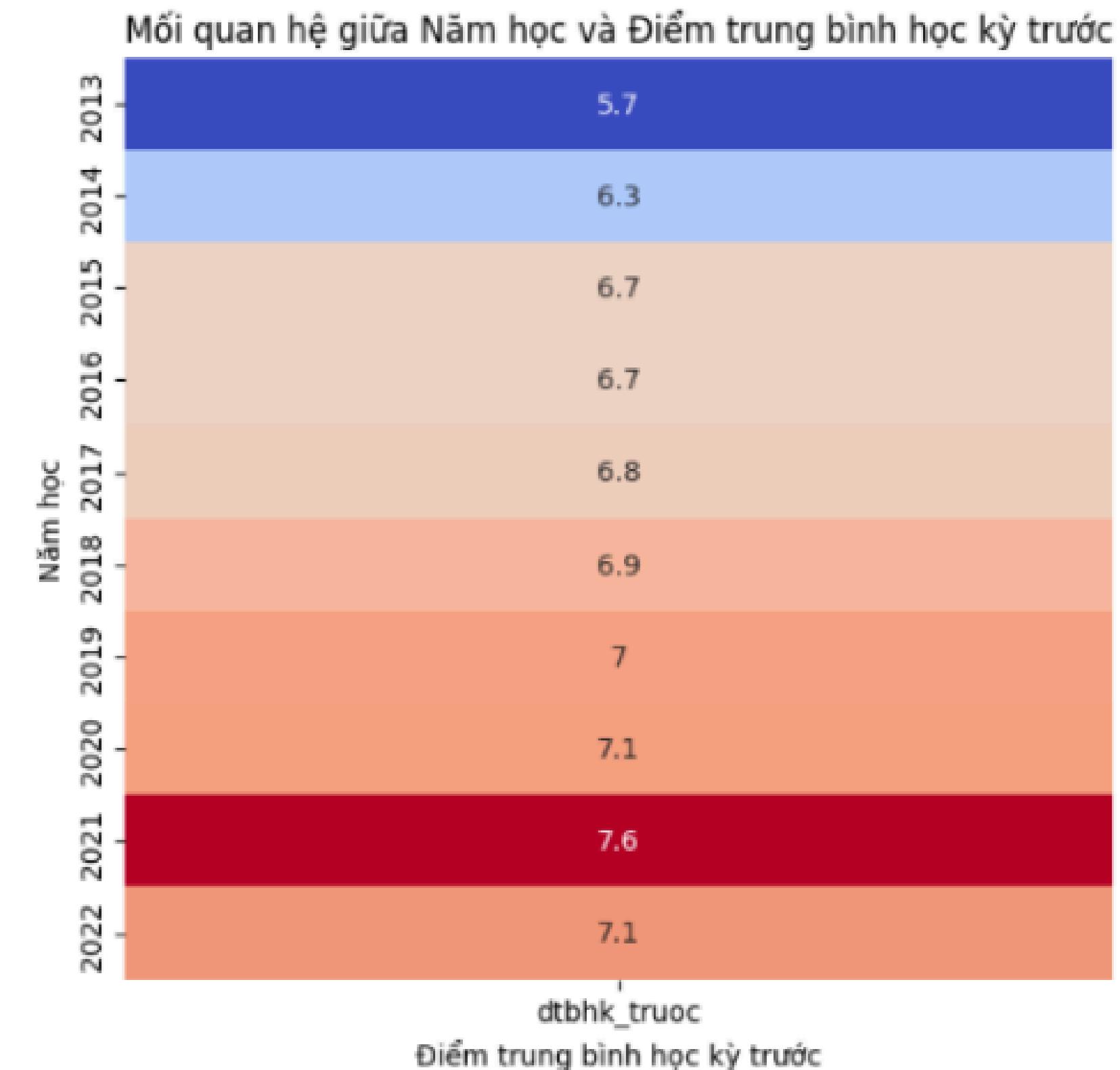
Biểu đồ cho thấy dtbhk_truoc và namhoc có MI cao (0.37), thể hiện mối quan hệ chặt chẽ, trong khi sotchk và drltl có MI cao hơn các cặp khác (0.04).



2.3 Mối quan hệ giữa các thuộc tính

dtbhk_truoc và namhoc

Xu hướng quan hệ tuyến tính giữa hai biến này cho thấy **dtbhktruoc tăng dần trong những năm gần đây**, phản ánh sự cải thiện chất lượng học tập của sinh viên.

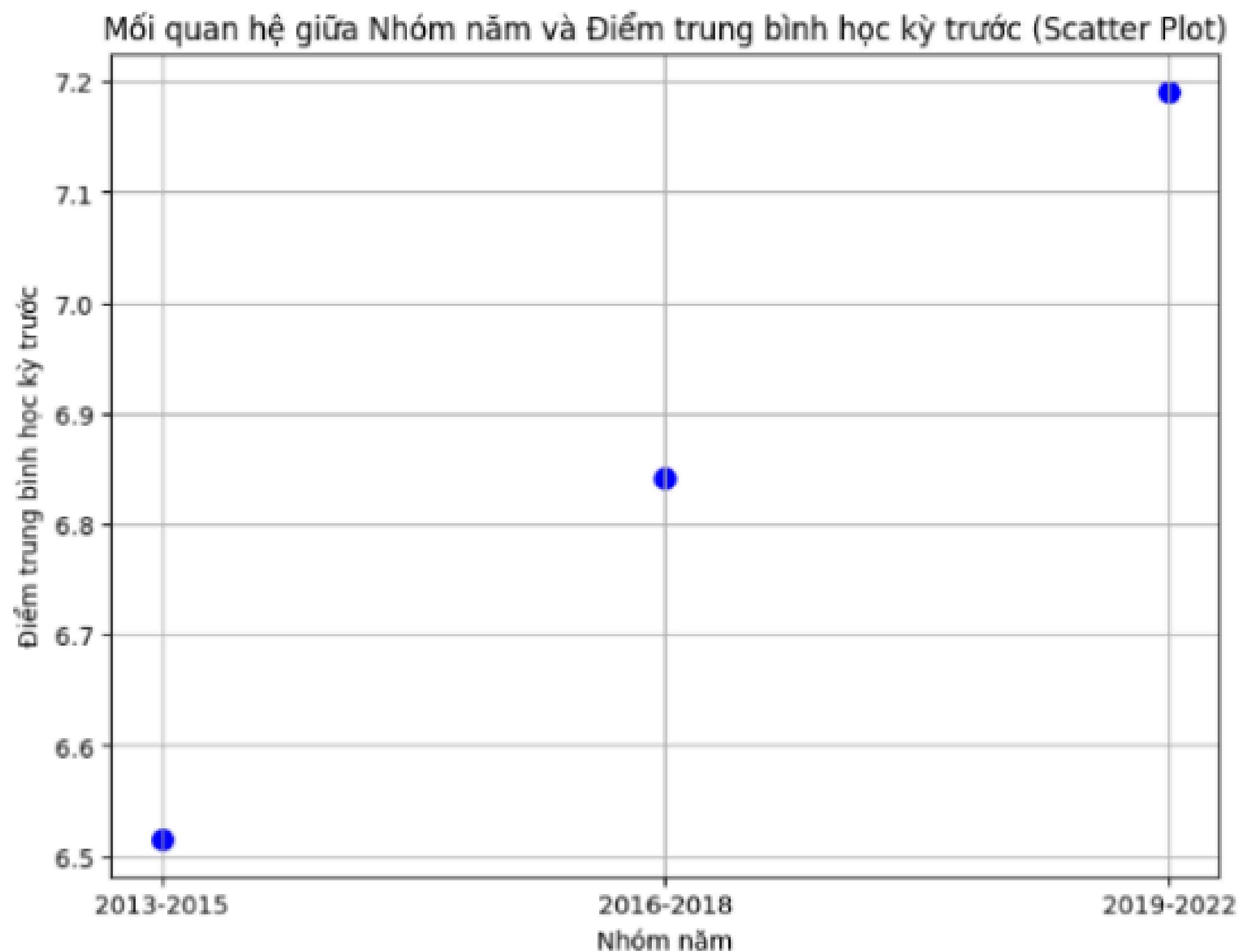


Mô hình giải bài toán

2.3 Mối quan hệ giữa các thuộc tính

dtbhk_truoc và namhoc

BIỂU ĐỒ PHÂN TÁN

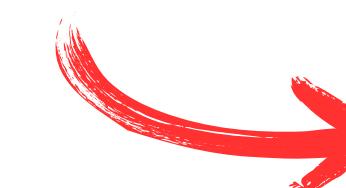


Mô hình giải bài toán

2.3 Mối quan hệ giữa các thuộc tính

dtbhk_truoc và namhoc

Nhóm thực hiện kiểm định **Chi-square**, với p_value < 0.5 cho thấy namhoc ảnh hưởng đến dtbhktroc. Chỉ số Cramér's V (0.02) cho thấy mối liên hệ khá mạnh giữa hai biến.



```
dtb_contingency = pd.crosstab(df['dtbhk_truoc'], df['namhoc'])

dtb_chi2, dtb_p, _, _ = chi2_contingency(dtb_contingency)

{
    "chi2_statistic": dtb_chi2,
    "p_value": dtb_p
}
```

```
In [13]: {'chi2_statistic': 14129.254850515426, 'p_value': 3.6682185101371154e-267}
```

```
In [14]:
```

```
def cramers_v(chi2, n, dof):
    return np.sqrt(chi2 / (n * dof))

n = df.shape[0]
dtb_dof = dtb_contingency.shape[0] - 1

dtb_cramers_v = cramers_v(dtb_chi2, n, dtb_dof)

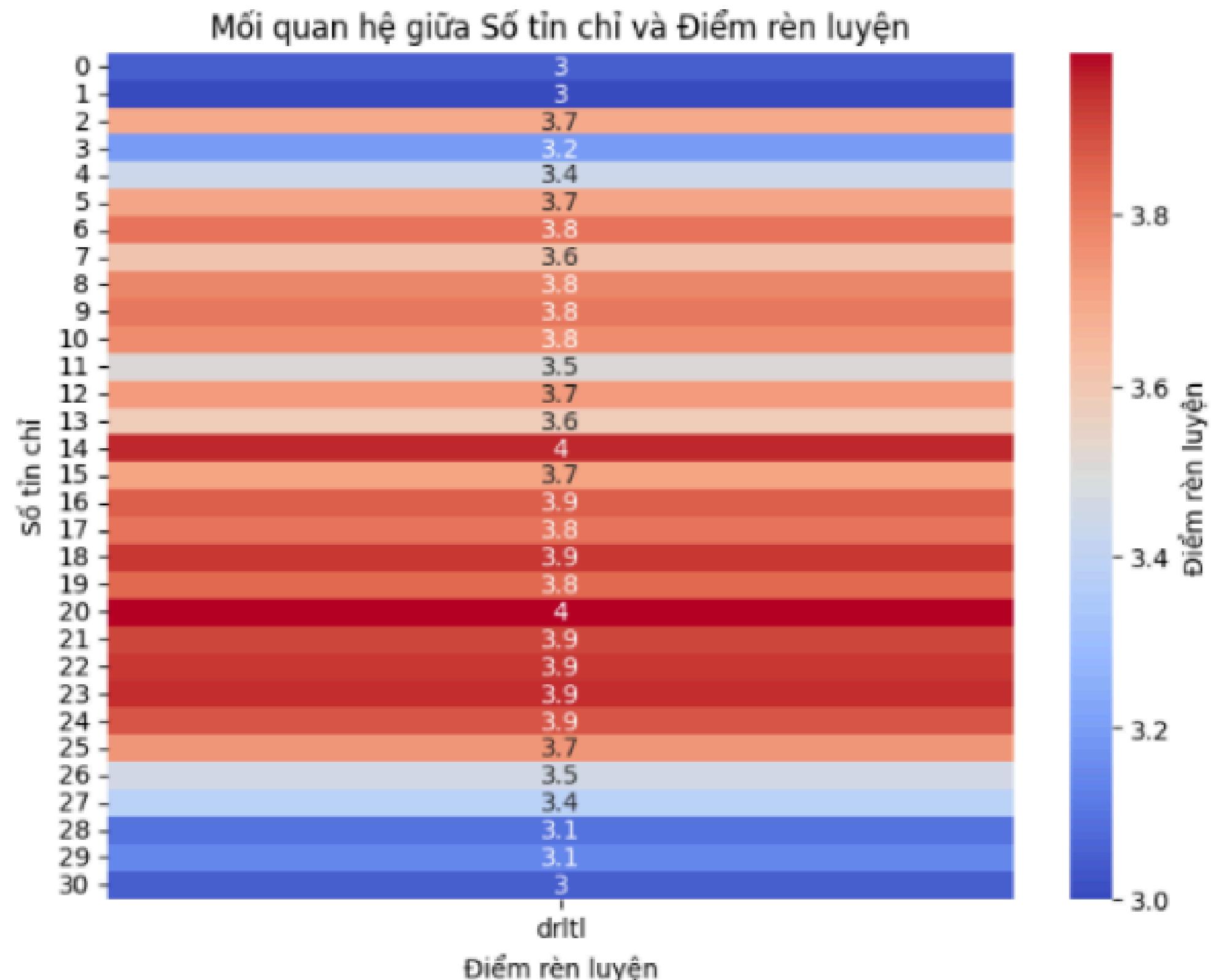
{
    "Hedt và xeploai": dtb_cramers_v
}
```

```
In [14]: {'Hedt và xeploai': 0.020209587529910384}
```

2.3 Mối quan hệ giữa các thuộc tính

sotckhk và drltl

- Tín chỉ từ 14-24 có điểm rèn luyện cao, chủ yếu xếp loại tốt.
- Tín chỉ >24 hoặc <14 có điểm thấp, do thiếu môn hoặc quá tải ảnh hưởng đến kết quả học tập.



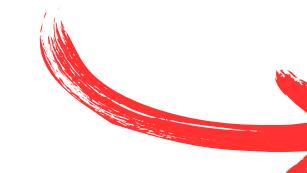
Mô hình giải bài toán

2.3 Mối quan hệ giữa các thuộc tính

sotckhk và drltl

- Chỉ số p_value < 0.5 cho thấy namhoc ảnh hưởng đến dtbhktroc, được củng cố bởi chỉ số Cramér's V (0.07), cho thấy mối liên hệ rất mạnh giữa hai biến.

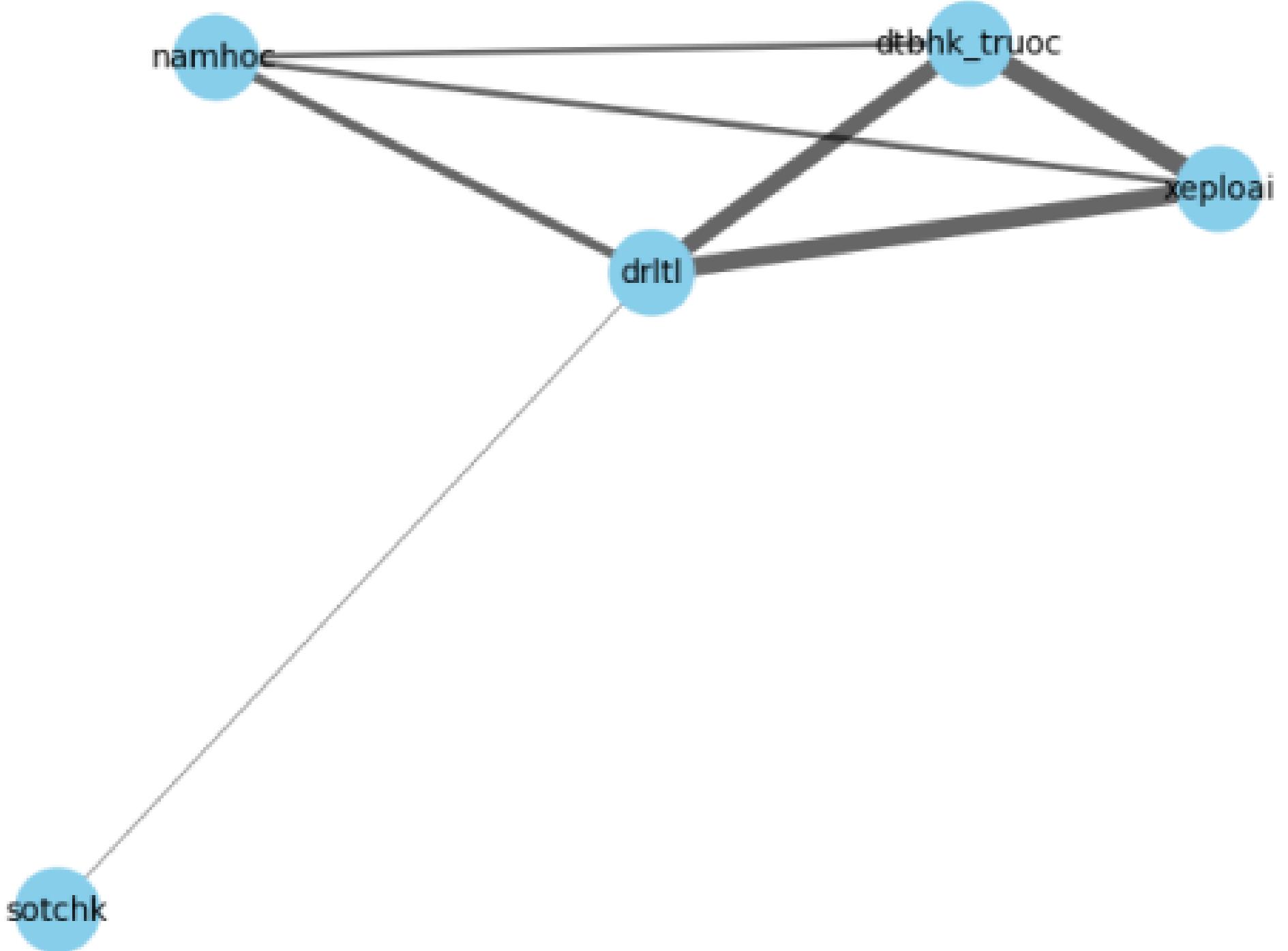
```
In [17]:  
dtb_contingency = pd.crosstab(df['drltl'], df['sotchk'])  
  
dtb_chi2, dtb_p, _, _ = chi2_contingency(dtb_contingency)  
  
{  
    "chi2_statistic": dtb_chi2,  
    "p_value": dtb_p  
}  
  
Out[17]:  
{'chi2_statistic': 1189.5411669457355, 'p_value': 4.7106086336993525e-146}  
  
In [18]:  
def cramers_v(chi2, n, dof):  
    return np.sqrt(chi2 / (n * dof))  
  
n = df.shape[0]  
dtb_dof = dtb_contingency.shape[0] - 1  
  
dtb_cramers_v = cramers_v(dtb_chi2, n, dtb_dof)  
  
{  
    "Sotckhk và drltl": dtb_cramers_v  
}  
  
Out[18]:  
{'Sotckhk và drltl': 0.07867707129335524}
```



2.4 Xây dựng đồ thị mạng

Biến đổi -> đưa vào ML/DL

- **Vector hóa dữ liệu:** Sử dụng DictVectorizer để chuyển đổi dữ liệu thành ma trận thưa (sparse matrix) từ danh sách các cặp thuộc tính liên kết mạnh.
- **Tạo ma trận kề:** Dựa trên mối quan hệ giữa các đối tượng lân cận (tổng các nút kết nối, các cạnh liền kề).
- **Phân cụm phân cấp:** Kết hợp với average pooling để xây dựng cấu trúc mạng đa tầng từ dữ liệu ban đầu, gom cụm đặc trưng và tính giá trị trung bình trong mỗi cụm.



2.4 Xây dựng đồ thị mạng

Đánh giá mô hình

- Ma trận kề và Average Pooling hiệu quả nhất trong việc giữ thông tin cấu trúc dữ liệu mạng.
- DictVectorizer cho kết quả thấp hơn do mất mối quan hệ giữa các nút trong quá trình "nén" vector.

Chọn hướng tiếp cận Ma trận kề và Hierarchical clustering.

Phương pháp	Mô hình	Average Accuracy	Mean Std
DictVectorizer	XGBoost	0.9058	0.0347
	Random Forest	0.9274	0.0292
Ma trận kề	XGBoost	0.9487	0.0035
	Random Forest	0.965	0.0032
Hierarchical clustering	XGBoost	0.9487	0.0035
	Random Forest	0.965	0.0032

2.5 Cân bằng dữ liệu

Synthetic Minority Over-sampling Technique - Edited Nearest Neighbors

SMOTENN kết hợp over-sampling (SMOTE) và under-sampling (ENN) để xử lý dữ liệu không cân bằng.

- **SMOTE** tạo mẫu tổng hợp cho lớp thiểu số.
 - **ENN** làm sạch dữ liệu bằng cách loại bỏ các mẫu không phù hợp.
 - ENN loại bỏ mẫu có nhãn khác với hầu hết hàng xóm gần nhất, giữ lại các mẫu rõ ràng và giảm nhiễu.
- > Phương pháp giúp cải thiện chất lượng mẫu lớp thiểu số và hiệu suất mô hình học máy.



2.5 Khai thác dữ liệu mạng

Dataset

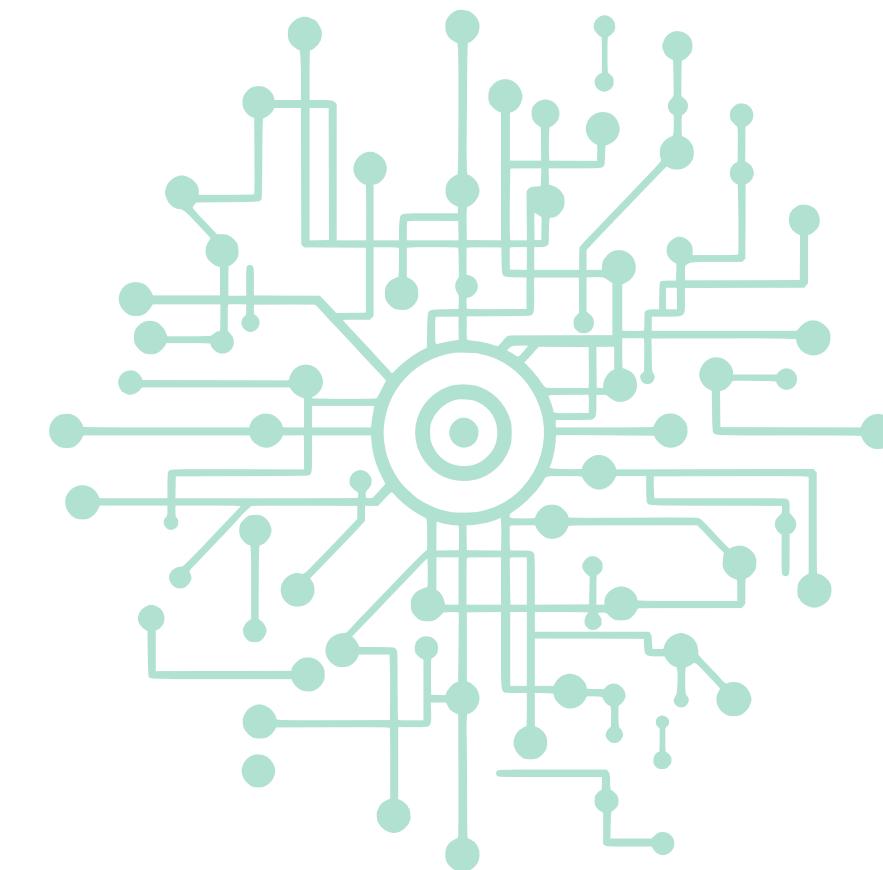
- Dữ liệu gốc có 35,849 dòng, sau khi áp dụng SMOTEENN, số mẫu giảm còn 24,496 do giảm mất cân bằng.
- Chia dữ liệu thành 10 phần (folds) và sử dụng 10-fold cross validation để đánh giá.



2.5 Khai thác dữ liệu mạng

Các hướng tiếp cận Machine Learning

- **XGBoost:** Thuật toán học tăng cường dựa trên cây quyết định, sử dụng gradient descent để giảm thiểu sai số dự đoán qua từng cây.
- **Random Forest:** Tạo nhiều cây quyết định độc lập bằng cách bootstrap dữ liệu và đặc trưng ngẫu nhiên, sau đó dự đoán qua bầu chọn đa số.
- **AdaBoost:** Kết hợp các bộ phân loại yếu bằng cách gán trọng số mẫu, tăng trọng số cho các mẫu khó và giảm cho mẫu dễ, tạo thành bộ phân loại mạnh.
- **SVM:** Phân loại tuyến tính và phi tuyến bằng cách tìm siêu phẳng tối ưu với lề rộng nhất trong không gian đặc trưng.



2.5 Khai thác dữ liệu mạng

Các hướng tiếp cận Deep Learning

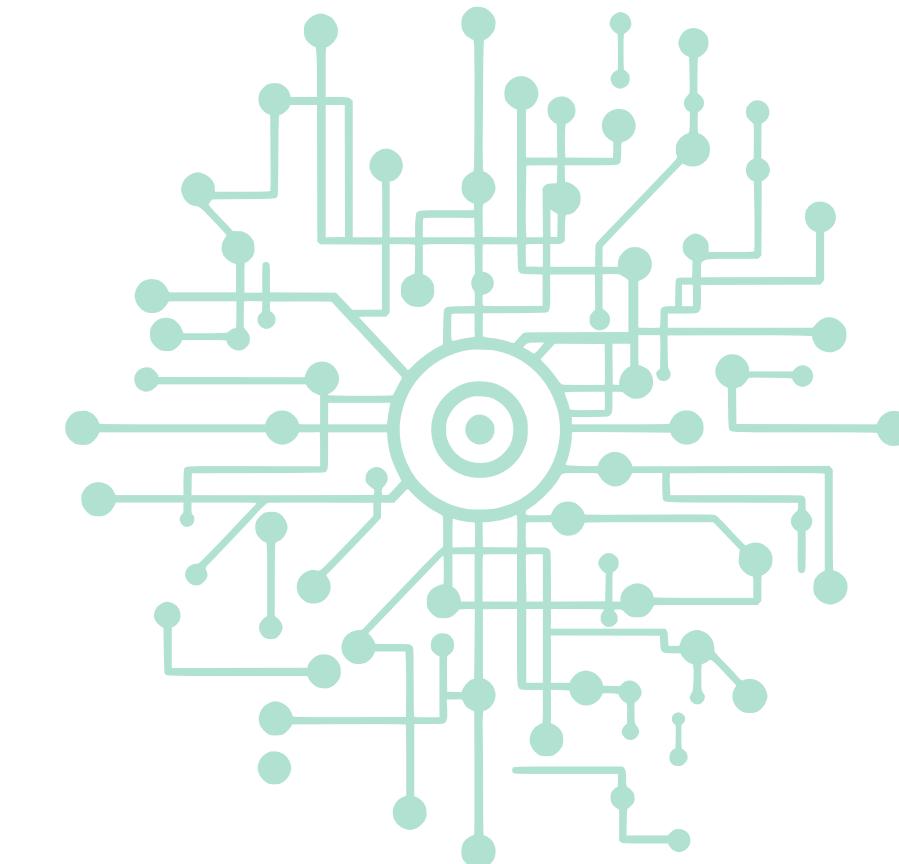
- Feed Forward Neural Network với cơ chế Attention Layer để xử lý và kết hợp hai cặp đặc trưng (drltl, sotchk) và (dtbhktroc, namhoc) từ dữ liệu mạng.
- Graph Convolutional Network với đầu vào là ma trận kề gồm các nút và danh sách cạnh liền kề, mô hình này sẽ lan truyền thông tin giữa các nút, qua đó, học được các đặc trưng từ cấu trúc đồ thị và dữ liệu liên quan.



2.5 Khai thác dữ liệu mạng

Kịch bản thực nghiệm

- Nhóm đề xuất kịch bản thực nghiệm tập trung vào việc phân chia dữ liệu theo thời gian,
- Dữ liệu học kỳ 1 được sử dụng làm đặc trưng đầu vào (features), trong khi dữ liệu học kỳ 2 được sử dụng làm nhãn mục tiêu (labels).
- Nhóm thực hiện mô phỏng tình huống thực tế, bằng cách sử dụng kết quả của học kỳ 1 là cơ sở để dự đoán kết quả học kỳ 2.
- Dữ liệu sau đó được chia thành hai phần: dữ liệu huấn luyện bao gồm các dòng có năm học nhỏ hơn 2021, và dữ liệu kiểm tra bao gồm các dòng có năm học từ 2021 trở đi.



Mô hình giải bài toán

2.5 Khai thác dữ liệu mạng

Kết quả thực nghiệm - Machine Learning

- Dữ liệu sau khi combine giúp cải thiện hiệu suất các mô hình, đặc biệt là Random Forest và XGBoost.
- Vấn đề mất cân bằng dữ liệu vẫn tồn tại, Class 2 có Precision thấp nhất.
- XGBoost Precision giảm từ 88% xuống 82%.

Thuật toán	Trước khi combine		Sau khi combine			
	Accuracy	Precision	Accuracy	Precision		
XGBoost	95%	Class 0	97%	95%	Class 0	97%
		Class 1	91%		Class 1	91%
		Class 2	86%		Class 2	86%
		Class 3	93%		Class 3	93%
		Class 4	97%		Class 4	97%
RandomForest	96%	Class 0	97%	98%	Class 0	98%
		Class 1	93%		Class 1	96%
		Class 2	88%		Class 2	93%
		Class 3	93%		Class 3	96%
		Class 4	97%		Class 4	99%
AdaBoost	93%	Class 0	97%	94%	Class 0	96%
		Class 1	89%		Class 1	89%
		Class 2	85%		Class 2	85%
		Class 3	93%		Class 3	93%
		Class 4	97%		Class 4	97%
SVM	94%	Class 0	97%	86%	Class 0	93%
		Class 1	93%		Class 1	77%
		Class 2	87%		Class 2	64%
		Class 3	87%		Class 3	78%
		Class 4	95%		Class 4	89%

Mô hình giải bài toán

2.5 Khai thác dữ liệu mạng

Kiểm tra overfitting -
Machine Learning

Thuật toán	Trước khi combine		Sau khi combine	
	Average Accuracy	Mean Std	Average Accuracy	Mean Std
XGBoost	94.87%	0.0035	96.20%	0.0041
Random Forest	96.19%	0.0025	98.04%	0.0033
AdaBoost	93.46%	0.0055	94.92%	0.0042
SVM	94.29%	0.0046	94.03%	0.0037

- Kết quả **k-fold cross validation** không thấy overfitting, độ lệch chuẩn thấp và Average Accuracy ổn định.
- Random Forest và XGBoost hiệu suất cao, ổn định.
- AdaBoost cải thiện ít hơn.
- SVM giảm độ chính xác, không khai thác tốt cấu trúc phi tuyến của dữ liệu.

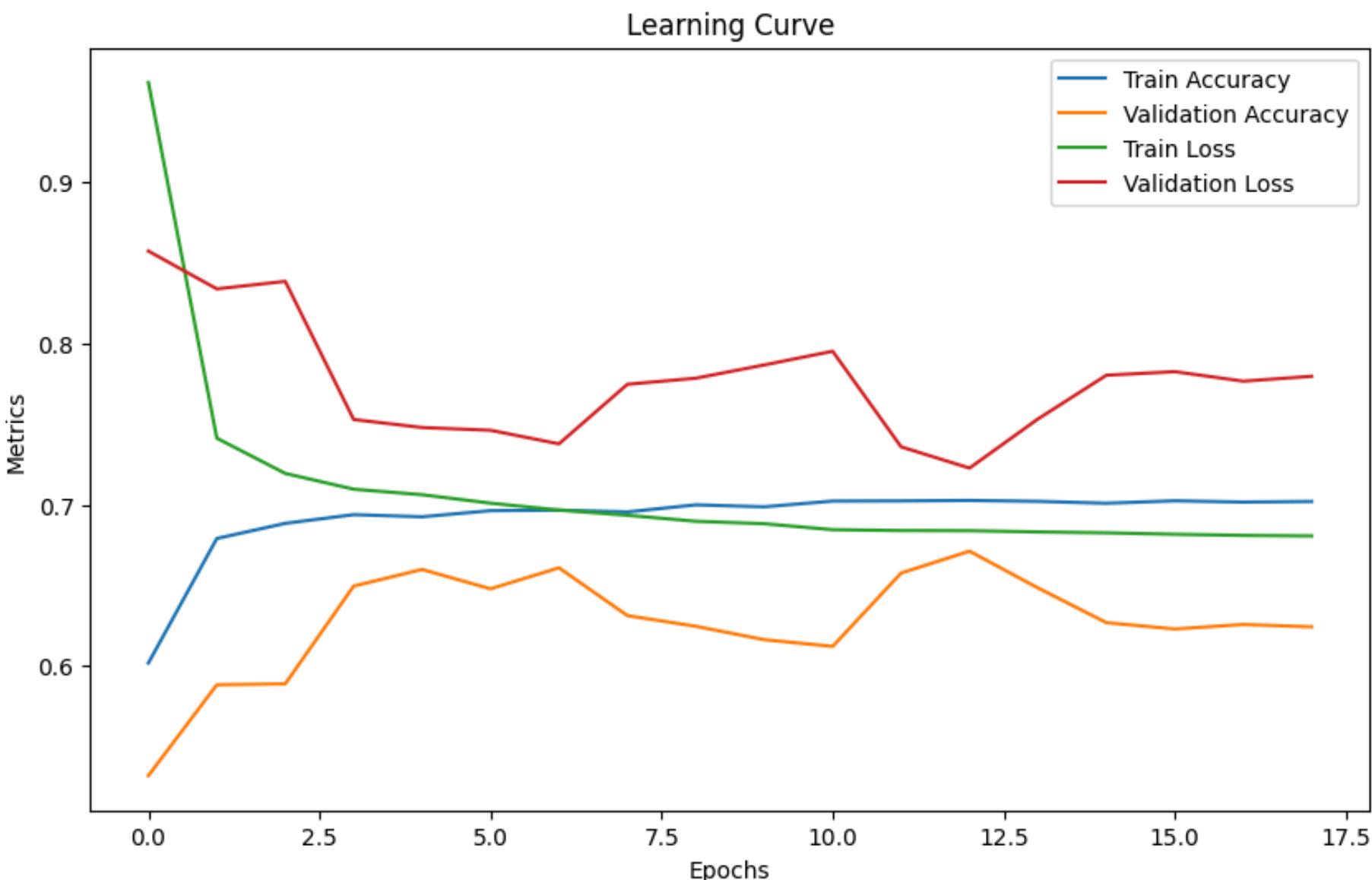
Mô hình giải bài toán

2.5 Khai thác dữ liệu mạng

Kết quả thực nghiệm Deep Learning

Feed Forward Neural Network

- Tập train: accuracy tăng từ 51.66% lên 70.14%
- Tập validation: accuracy tăng từ 53.2% lên 65.98%
- Biến động nhẹ giữa các epoch (ví dụ: giảm xuống 61.21% ở epoch 11).
- Dấu hiệu overfitting nhẹ khi chênh lệch giữa train và validation chỉ ~5-10%.



Mô hình giải bài toán

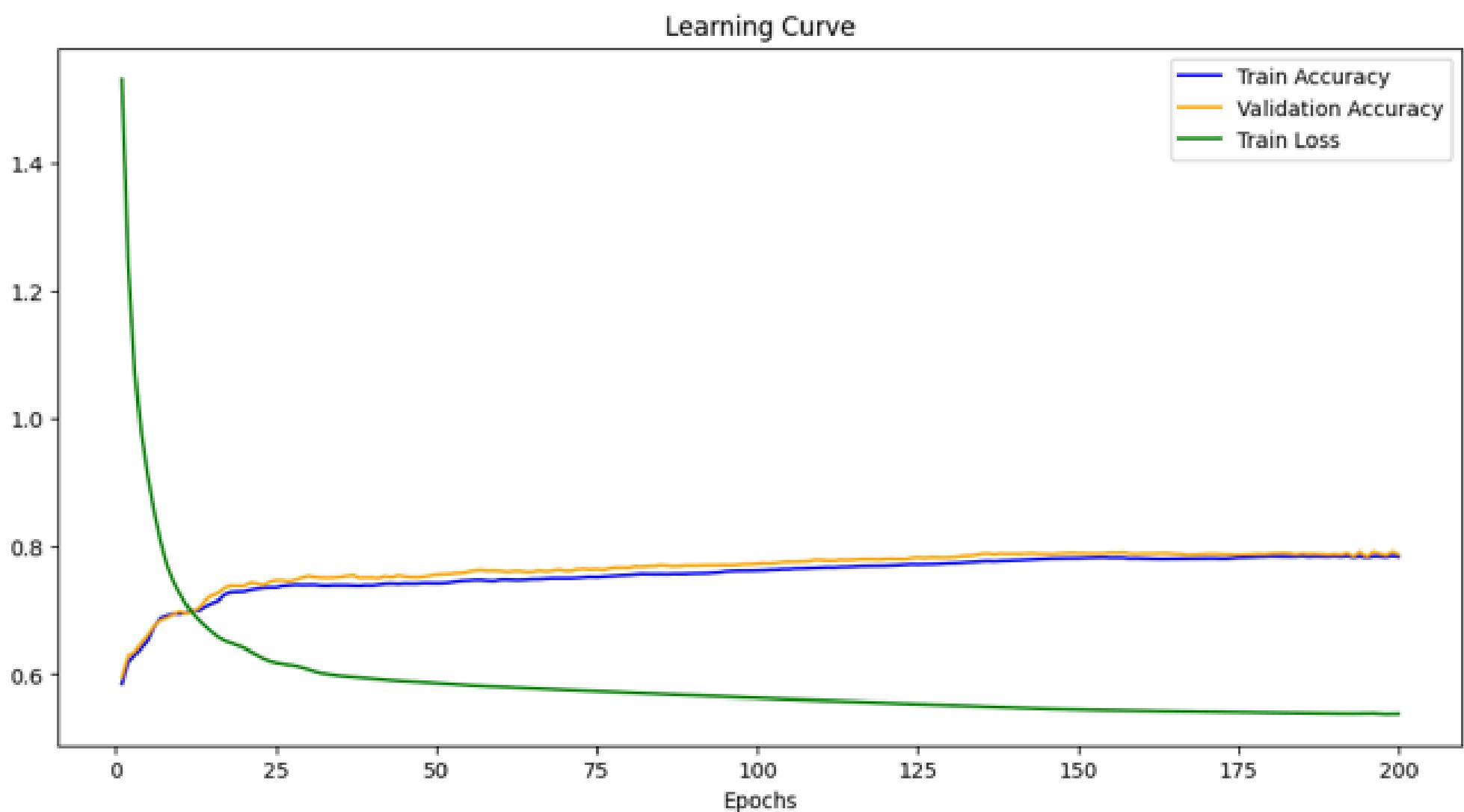
2.5 Khai thác dữ liệu mạng

Kết quả thực nghiệm Deep Learning

Graph Convolutional Network

- Train Accuracy và Validation Accuracy đều tăng nhanh ở các Epoch đầu và duy trì ổn định sau khoảng 50 epoch (79%).
- Khoảng cách giữa 2 độ chính xác rất nhỏ, không có hiện tượng overfitting.
- Đường Validation Accuracy gần song song với Train Accuracy, không giảm ở các epoch sau.
- Điều này chứng tỏ mô hình không overfitting.

=> Kết luận: GCN tốt hơn FNN



Mô hình giải bài toán

2.5 Khai thác dữ liệu mạng

Chương trình Demo

Kết quả dự đoán xếp loại học kỳ tiếp theo của bạn là **Giỏi**

Bạn đang làm rất tốt! Hãy giữ vững phong độ và luôn sẵn sàng học hỏi thêm. Thành công lớn đang chờ bạn phía trước!

Điểm trung bình học kỳ của bạn:

8

Điểm rèn luyện tích lũy:

40

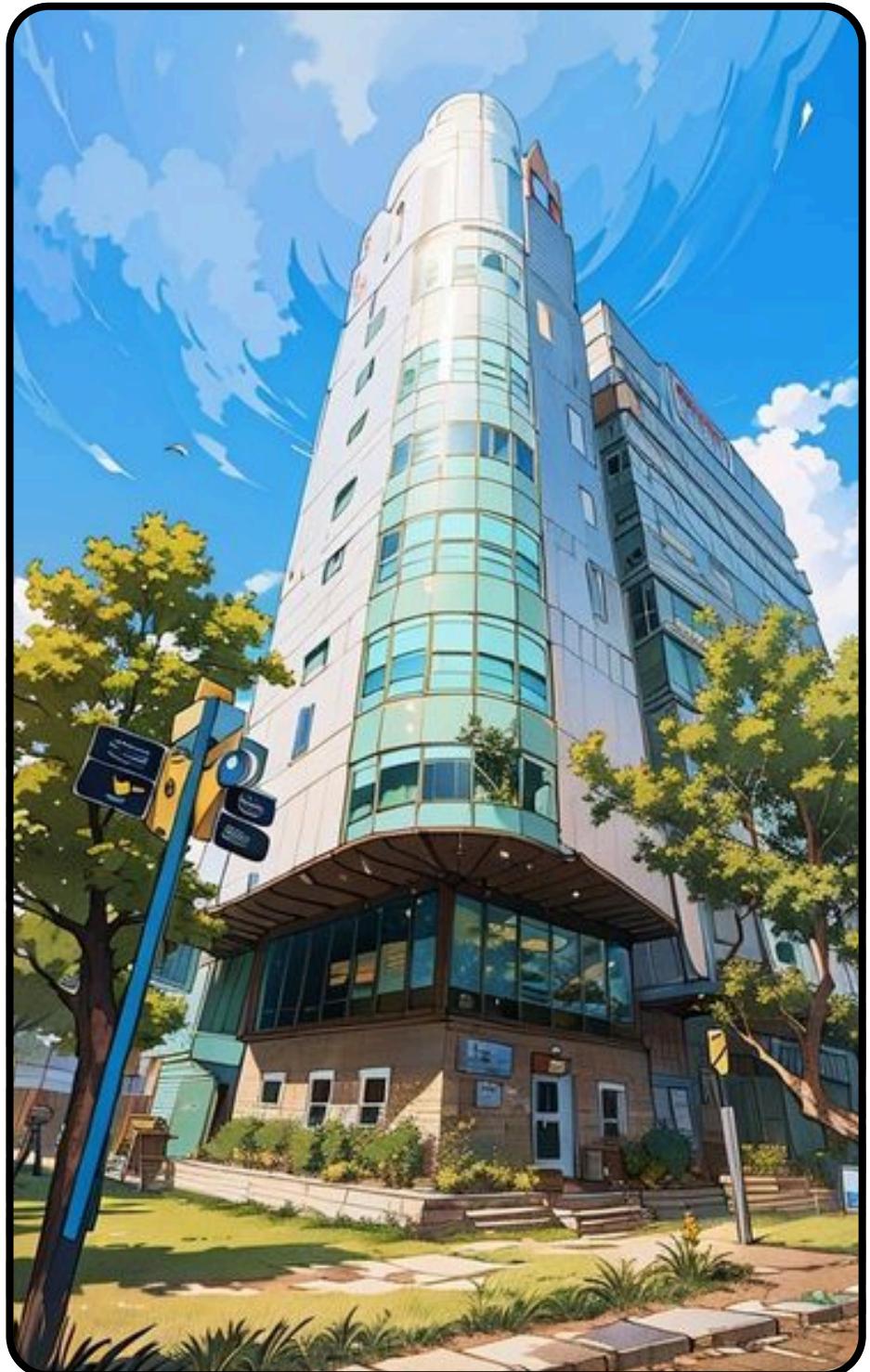
Số tín chỉ:

20

Năm học:

2024

Dự đoán



3. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

3. KẾT LUẬN

Mô hình dự đoán đạt độ chính xác cao, giúp hỗ trợ sinh viên qua các yếu tố như điểm số và tín chỉ.

3.2 Hướng phát triển:

- **Cải thiện dữ liệu:** Thu thập thêm nguồn dữ liệu để nâng cao độ chính xác.
- **Cải tiến mô hình:** Thử nghiệm thuật toán mới và tối ưu mô hình.
- **Ứng dụng thực tiễn:** Mở rộng dự đoán các yếu tố như khả năng tốt nghiệp.
- **Khả năng mở rộng:** Áp dụng cho nhiều trường và ngành học.

NHÓM 3

CHÂN THÀNH CẢM ƠN CÔ VÀ MỌI
NGƯỜI ĐÃ LẮNG NGHE!

