ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA KHOA HỌC MÁY TÍNH
----------o0o----------

BÁO CÁO MÔN HỌC

Nhập môn Học máy và Khai phá dữ liệu

Đề tài: Phân tích kết quả học tập của sinh viên

GVHD: TS. Nguyễn Nhật Quang

SVTH: Vương Hữu Hưng 20204564
Nguyễn Hoàng Bảo 20204515
Hoàng Quốc Trung 20204613
Đỗ Đức Phương 20204680
Nguyễn Minh Đức 20183713

TP. HÀ NỘI, THÁNG 12 NĂM 2023

Mục lục

1. Phân công công việc

Họ và tên	Công việc
Hoàng Quốc Trung	Phân cụm dữ liệu, Phát hiện ngoại lai, tìm hiểu thuật toán

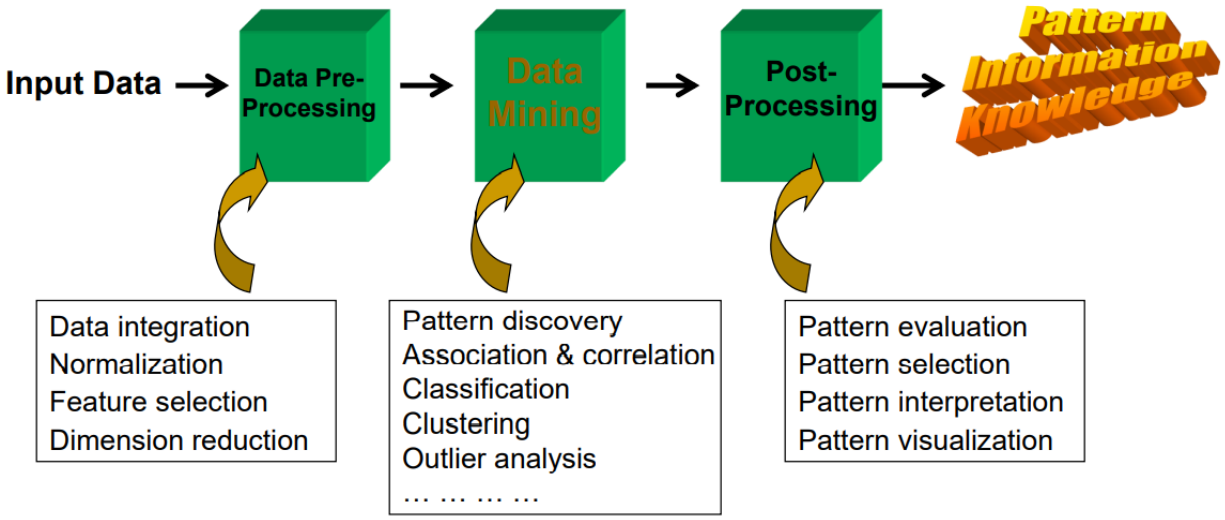
Nguyễn Minh Đức	Tiền xử lý dữ liệu, phân tích dữ liệu sau khi tiền xử lý, tìm hiểu thuật toán
Vương Hữu Hưng	Xây dựng mô hình, Phân cụm dữ liệu, tìm hiểu thuật toán
Nguyễn Hoàng Bảo	Phân tích dữ liệu sau khi tiền xử lý, Đánh giá hiệu năng, tìm hiểu thuật toán
Đỗ Đức Phương	Tiền xử lý dữ liệu, phân tích dữ liệu sau khi tiền xử lý, tìm hiểu thuật toán

2. Giới thiệu bài toán và ứng dụng

   a. Lĩnh vực Khai phá dữ liệu và phân cụm

Trong kỷ nguyên mới, khai phá dữ liệu đang trở thành từ khóa rất hot. Thực vậy, khai phá dữ liệu đã và đang được đẩy mạnh nghiên cứu và ứng dụng nhiều trong các lĩnh vực khác nhau và mang lại những cải thiện rất đáng kể. Những lĩnh vực có ứng dụng của khai phá dữ liệu bao gồm: thiên văn học, tin sinh học, thương mại điện tử, marketing, quản lý quan hệ khách hàng, y tế…. Đặc biệt, khai phá dữ liệu được xem là phương pháp mà đơn vị Able Danger của Quân đội Hoa Kỳ sử dụng để xác định lực lượng khủng bố đứng đầu cuộc tấn công ngày 11 tháng 9 nhằm vào nước này.

Khai phá dữ liệu là quá trình tính toán để tìm ra các mẫu, tìm ra thông tin đáng quý và tri thức từ một bộ dữ liệu để sử dụng cho quá trình đánh giá, phân tích sau này. Nó là một bước của quá trình khai phá tri thức.



Các phương pháp khai phá dữ liệu có thể chia làm các phương pháp chính sau:

- Phân loại (Classification): Là phương pháp dự báo, cho phép phân loại một đối tượng vào một hoặc một số lớp cho trước.
- Hồi qui (Regression): Khám phá chức năng học dự đoán, ánh xạ một mục dữ liệu thành biến dự đoán giá trị thực.
- Phân cụm (Clustering): Một nhiệm vụ mô tả phổ biến trong đó người ta tìm cách xác định một tập hợp hữu hạn các cụm để mô tả dữ liệu.
- Tổng hợp (Summarization): Một nhiệm vụ mô tả bổ sung liên quan đến phương pháp cho việc tìm kiếm một mô tả nhỏ gọn cho một bộ (hoặc tập hợp con) của dữ liệu.
- Mô hình ràng buộc (Dependency modeling): Tìm mô hình cục bộ mô tả các phụ thuộc đáng kể giữa các biến hoặc giữa các giá trị của một tính năng trong tập dữ liệu hoặc trong một phần của tập dữ liệu.

- Dò tìm biến đổi và độ lệch (Change and Deviation Dectection): Khám phá những thay đổi quan trọng nhất trong bộ dữ liệu.

Trong các phương pháp khai phá dữ liệu, phân cụm dữ liệu (data clustering) là quá trình tìm kiếm để phân ra các cụm dữ liệu, các mẫu dữ liệu từ tập dữ liệu lớn sao cho các đối tượng trong cùng một cụm là tương đồng, còn các đối tượng thuộc các cụm khác nhau sẽ có nét khác nhau rõ rệt.

b. Mô tả bài toán

Số học sinh, sinh viên của ngành giáo dục nói chung là một con số rất lớn. Mỗi trường, cơ sở đào tạo có số học sinh, sinh viên lớn; đặc biệt các trường đại học, như Đại học Bách Khoa Hà Nội, con số lên đến 30, 40 nghìn sinh viên. Chỉ riêng việc quản lý số sinh viên khổng lồ đã là một thách thức rất lớn huống chi việc phân tích và đánh giá, vẽ ra các chiến lược cho các sinh viên. Việc này chắc chắn không thể làm thủ công mà phải nhờ vào sự hỗ trợ của công nghệ. Công nghệ sẽ đảm nhận phân tích, tìm ra điểm sai, tìm ra lí do từ đó các nhà lãnh đạo đánh giá, đưa ra các chiến lược phù hợp, nâng cấp chất lượng cho việc giáo dục và các vấn đề xoay quanh để giảng dạy trở nên hiệu quả.

Kết quả học tập của một cá nhân phụ thuộc vào rất nhiều yếu tố. Bài toán này sẽ xem xét chủ yếu dựa trên những tác nhân chủ quan từ phía sinh viên: kết quả trung bình học kỳ, giới tính, quê quán, các đặc điểm gia đinh, số lượng các học phần đăng ký cho kỳ, tỷ lệ tham dự các buổi học trên lớp…. Thực hiện việc phân cụm có thể tìm ra được những tác nhân gây ra kết quả học tập không tốt của học sinh sinh viên, sự liên kết giữa chúng từ đó tìm ra giải pháp. Kết quả học tập của học sinh được xem là tốt, không tốt sẽ dựa trên điểm trung bình của sinh viên.

c. Ứng dụng

Bài toán có trả lời một số những câu hỏi như:

- Giới tính có ảnh hưởng đến khả năng học tập không?
- Trình độ giáo dục của cha mẹ liên quan đến khả năng học tập của con cái không?
- Việc thi các bài test trước kì thi có giúp cải thiện kết quả bài thi không?
- Nên làm thế nào để có thể có được kết quả kì thi tốt nhất?

3. Tổng quan dữ liệu

a. Nguồn dữ liệu

Nguồn dữ liệu sử dụng:
https://www.kaggle.com/datasets/spscientist/students-performance-in-exams

Bộ dữ liệu chứa điểm thi của học sinh trong các môn học khác nhau, với mục tiêu kiểm tra sự ảnh hưởng của phụ huynh, việc luyện thi tới kết quả học tập của học sinh.

```
# Display head of data
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

b. Mô tả dữ liệu

- Dữ liệu được lưu dưới dạng file csv

- Số lượng bản ghi là 1000
- Số cột ứng với số thuộc tính là 8
- Với các thông tin cơ bản về các thuộc tính:

Thuộc tính	Ý nghĩa
gender	Giới tính của sinh viên
race/ethnicity	Dân tộc
parental level of education	Trình độ giáo dục của cha mẹ
lunch	Bữa trưa sinh viên sử dụng
test preparation course	Làm các bài test trước kỳ thi
math score	Điểm toán
reading score	Điểm đọc hiểu
writing score	Điểm viết

c. Tiền xử lý dữ liệu

- Kiểm tra kiểu dữ liệu:

```
#Column data types
df.dtypes
```

```
gender                       object
race/ethnicity               object
parental level of education  object
lunch                        object
test preparation course      object
math score                    int64
reading score                 int64
writing score                 int64
dtype: object
```

- Kiểm tra có giá trị null không:

```
#Check if data has missing value
df.isna().sum()
```

```
gender                       0
race/ethnicity               0
parental level of education  0
lunch                        0
test preparation course      0
math score                   0
reading score                0
writing score                0
dtype: int64
```

- Kiểm tra kết quả trùng lặp:

```
#Check if data has duplicated value
df.duplicated().sum()
```

```
0
```

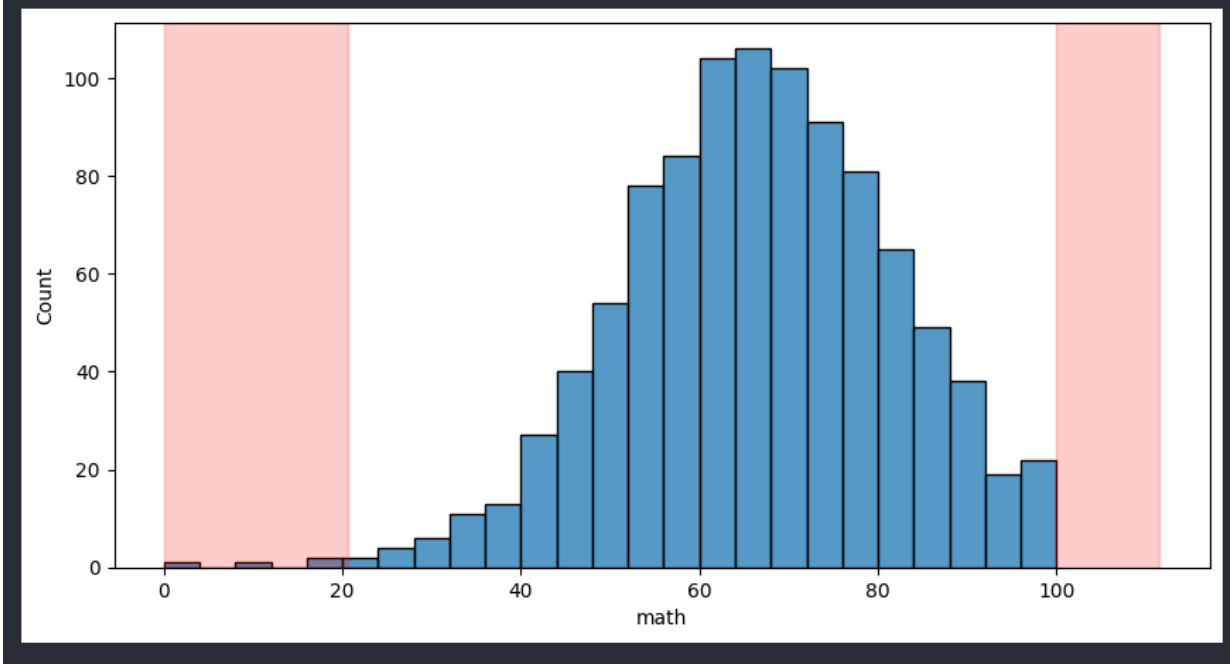
- Phát hiện ngoại lai:
    - Phương pháp sử dụng: Standard Deviation, độ lệch chuẩn là một thước đo của phương sai, tức là mức độ trải rộng của các điểm dữ liệu riêng lẻ so với giá trị trung bình. Trong thống kê, nếu phân phối dữ liệu xấp xỉ bình thường thì khoảng 68% giá trị dữ liệu nằm trong một độ lệch chuẩn của giá trị trung bình và khoảng 95% nằm trong hai độ lệch chuẩn và khoảng 99,7% nằm trong ba độ lệch chuẩn.
    - Khai báo hàm phát hiện ngoại lai:

```python
#Define a outlier check function
def outlier_standard_deviation(df, column):
    data_mean, data_std = df[column].mean(), df[column].std()
    cut_off = data_std * 3
    lower, upper = data_mean - cut_off, data_mean + cut_off
    print('The lower bound value is', lower)
    print('The upper bound value is', upper)
    df1 = df[df[column] > upper]
    df2 = df[df[column] < lower]
    print('Total number of outliers are', df1.shape[0]+ df2.shape[0])
    plt.figure(figsize = (10,5))
    sns.histplot(df['math'], kde=False)
    plt.axvspan(xmin = lower,xmax= df['math'].min(),alpha=0.2, color='red')
    plt.axvspan(xmin = upper,xmax= df['math'].max(),alpha=0.2, color='red')
    return lower, upper
```

- Phát hiện ngoại lai với thuộc tính điểm toán:

```python
#Check ouliers in math columns, filter them
lower, upper = outlier_standard_deviation(df, 'math')
df = df[(df['math'] < upper) & (df['math'] > lower)]
```

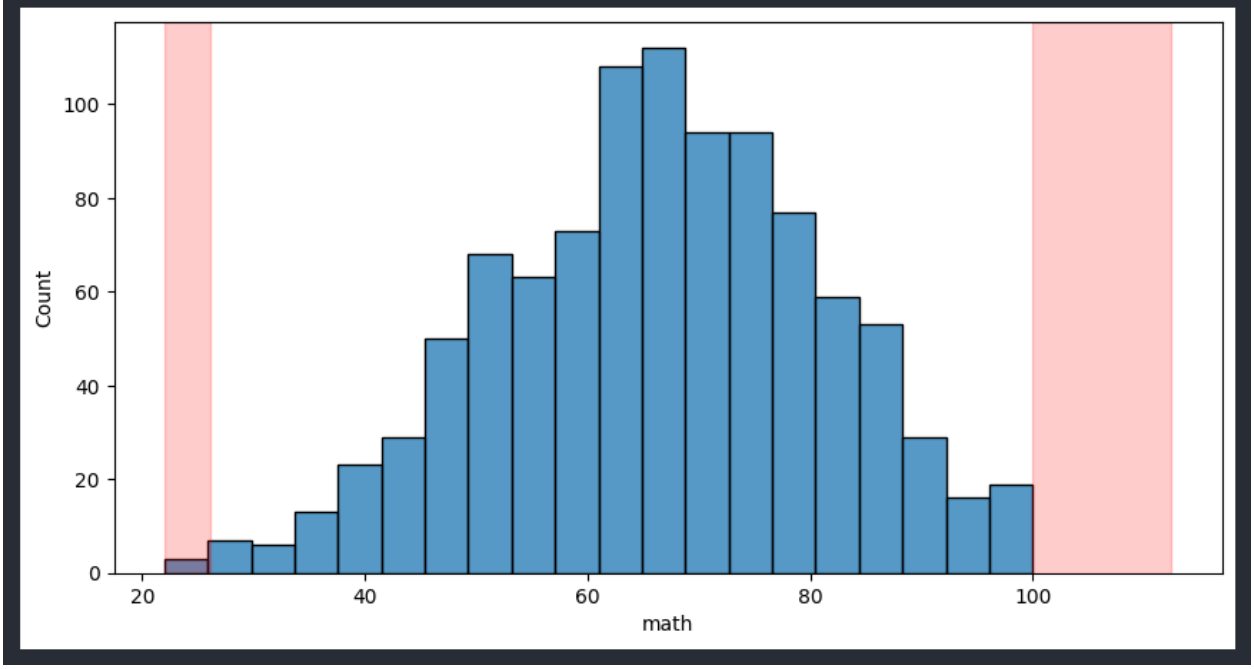
```
The lower bound value is 20.59975971197165
The upper bound value is 111.57824028802835
Total number of outliers are 4
```



- Phát hiện ngoại lai với thuộc tính điểm đọc hiểu:

```
#Check ouliers in reading columns, filter them
lower, upper = outlier_standard_deviation(df, 'reading')
df = df[(df['reading'] < upper) & (df['reading'] > lower)]
```

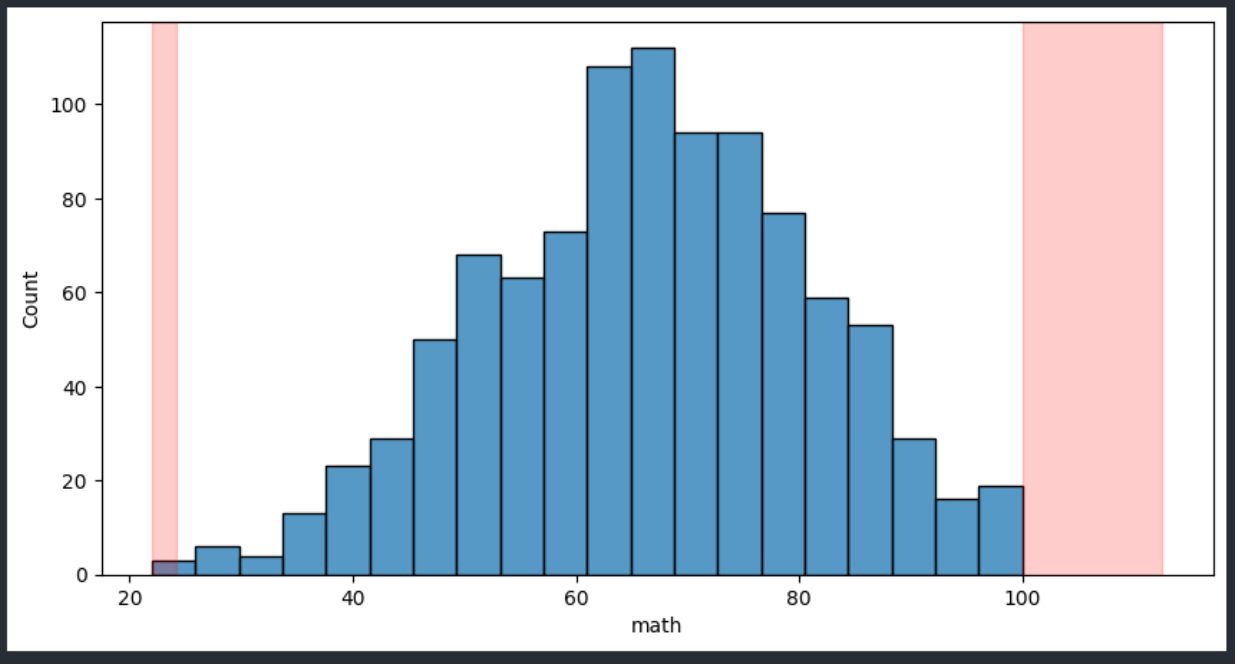
```
The lower bound value is 26.189102205790704
The upper bound value is 112.48158052513298
Total number of outliers are 3
```



- Phát hiện ngoại lai với điểm viết:

```python
#Check ouliers in writing columns, filter them
lower, upper = outlier_standard_deviation(df, 'writing')
df = df[(df['writing'] < upper) & (df['writing'] > lower)]
```

```
The lower bound value is 24.225961923417813
The upper bound value is 112.54140967779064
Total number of outliers are 0
```



- Dữ liệu sau khi lọc bỏ ngoại lai:

```
df
```

	gender	race	parent_education	lunch	test_preparation	math	reading	writing
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

993 rows × 8 columns

- Xử lý các thuộc tính không nguyên:

```
#Encode nominal columns
columns_encode = ['gender','race','parent_education','lunch','test_preparation']
for column in columns_encode:
    encoder = LabelEncoder()
    kmeans_df[column] = encoder.fit_transform(kmeans_df[column])
    print(encoder.classes_)
    print(np.sort(kmeans_df[column].unique()))
```

```
['female' 'male']
[0 1]
['group A' 'group B' 'group C' 'group D' 'group E']
[0 1 2 3 4]
["associate's degree" "bachelor's degree" 'high school' "master's degree"
 'some college' 'some high school']
[0 1 2 3 4 5]
['free/reduced' 'standard']
[0 1]
['completed' 'none']
[0 1]
```

4. Giải quyết bài toán
    a. Phân tích dữ liệu

Tính thêm cột điểm trung bình:

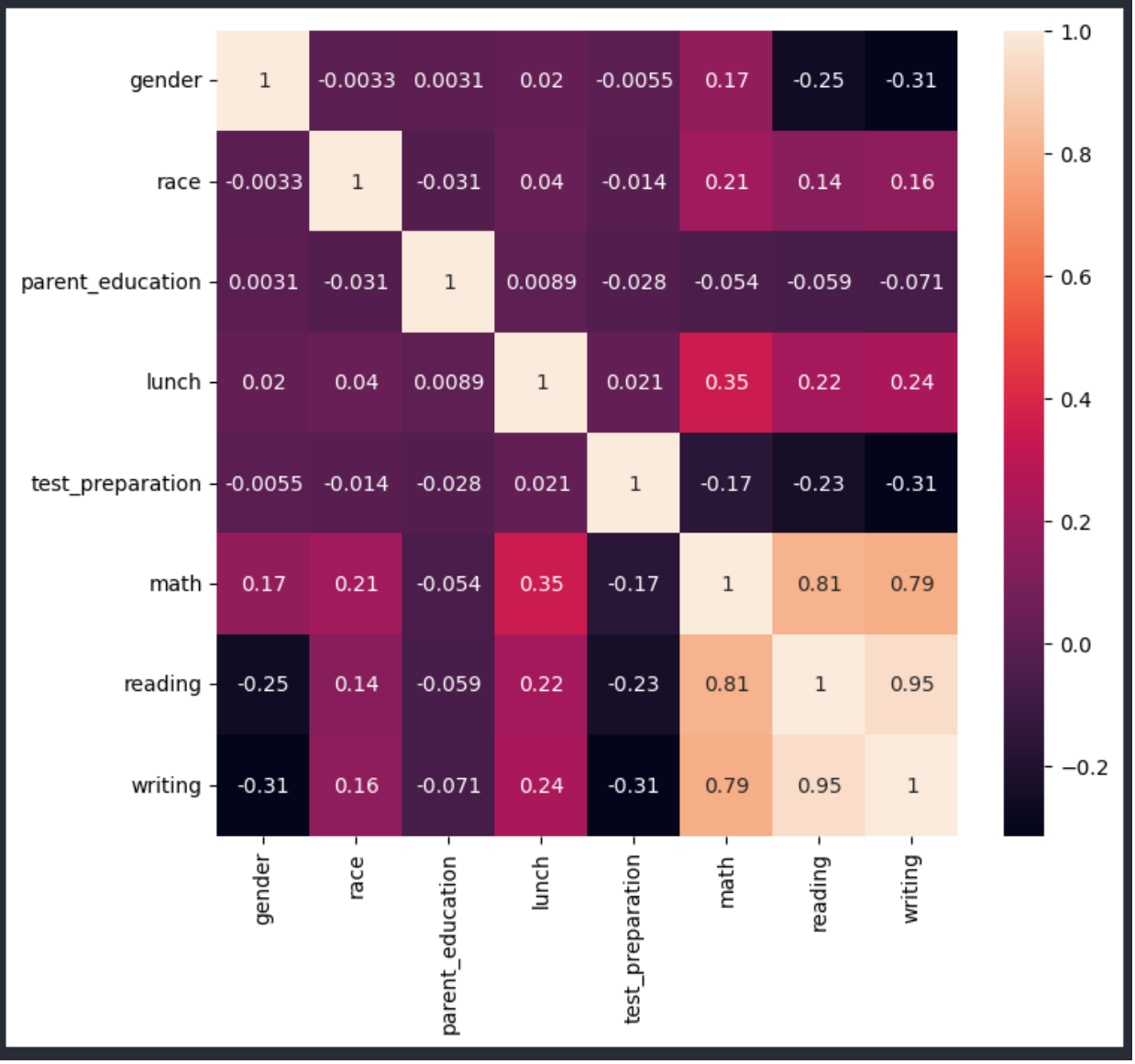
```
#Calculate final score
research_df['final score'] = research_df[['math', 'reading', 'writing']].mean(axis=1)
research_df.head()
```

	gender	race	parent_education	lunch	test_preparation	math	reading	writing	final score
0	female	group B	bachelor's degree	standard	none	72	72	74	72.666667
1	female	group C	some college	standard	completed	69	90	88	82.333333
2	female	group B	master's degree	standard	none	90	95	93	92.666667
3	male	group A	associate's degree	free/reduced	none	47	57	44	49.333333
4	male	group C	some college	standard	none	76	78	75	76.333333

Kiểm tra tương quan giữa các thuộc tính:

```
#Correlations among columns
correlations = kmeans_df.corr()
fig = plt.figure(figsize=(8,7))
sns.heatmap(data=correlations, annot=True)
```

```
<AxesSubplot: >
```



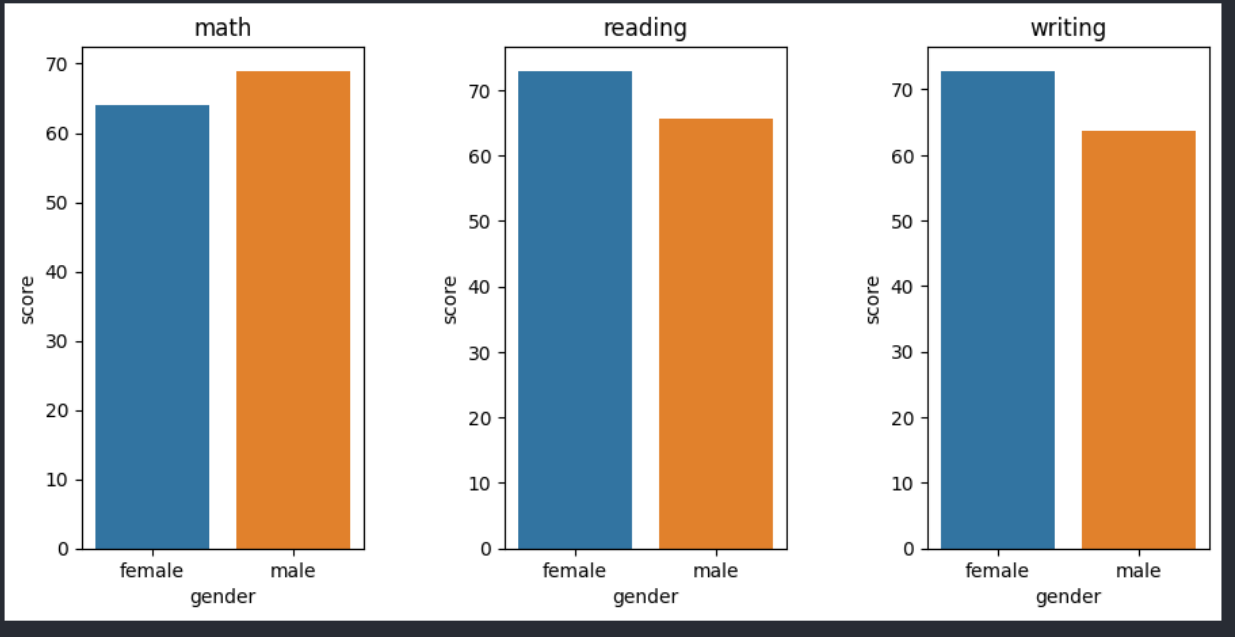
Có thể thấy những học sinh có kết quả tốt sẽ có số điểm ở cả ba môn đều cao, đặc biệt là học sinh có điểm đọc hiểu cao thì điểm viết của học sinh đó cũng cao.

Các thuộc tính còn lại chưa thấy sự liên kết rõ rệt, cần phân tích điểm số trên từng thuộc tính

Xem xét sự phân bố điểm giữa nam và nữ:

```
#Showing scores base on gender
fig, ax = plt.subplots()
fig.subplots_adjust(hspace=0.5, wspace=0.5, left = 0.2, right = 1.5)

for i in range(3):
    plt.subplot(1,3, i+1)
    gender_df = research_df.groupby('gender')[list(research_df.columns[-4:-1])[i]].describe()
    sns.barplot(x=gender_df.index, y=gender_df.loc[:,"mean"].values)
    plt.ylabel("score")
    plt.title(list(research_df.columns[-4:-1])[i])
```

```
C:\Users\dinhh\AppData\Local\Temp\ipykernel_2180\1207430453.py:6: MatplotlibDeprecationWarning:

Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases later; explicitly
```

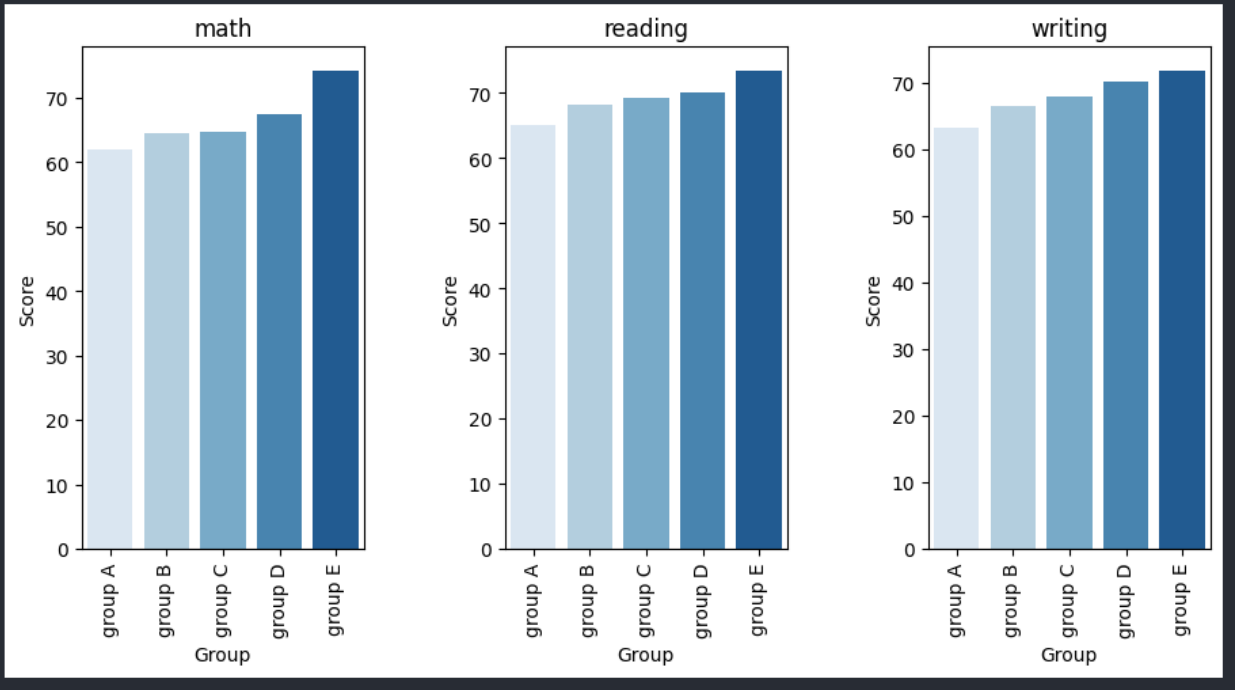


Có thể thấy con trai thường có điểm toán cao hơn nữ nhưng lại có điểm viết và đọc hiểu kém hơn.

Xem xét sự phân bố điểm dựa trên khu vực dân tộc:

```python
#Showing scores base on race
fig, ax = plt.subplots()
fig.subplots_adjust(hspace=0.5, wspace=0.5, left = 0.2, right = 1.5)
for idx in range(3):
    plt.subplot(1,3, idx+1)
    race_df = research_df.groupby("race")[list(research_df.columns[-4:-1])[idx]].mean()
    sns.barplot(x=race_df.index, y = race_df.values, palette = "Blues")
    plt.xlabel("Group")
    plt.ylabel("Score")
    plt.xticks(rotation=90)
    plt.title(list(research_df.columns[-4:-1])[idx])
plt.show()
```

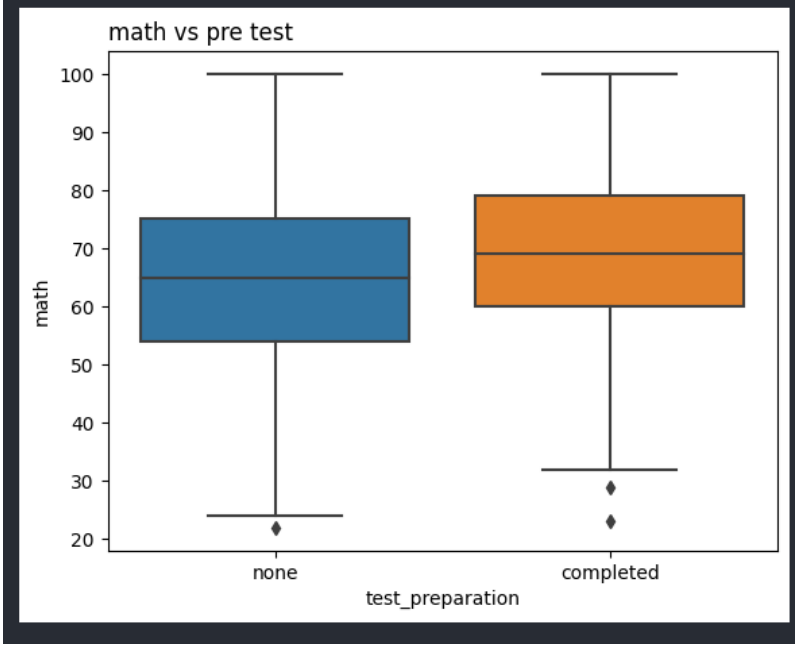
```
C:\Users\dinhh\AppData\Local\Temp\ipykernel_2180\3886928119.py:5: MatplotlibDeprecationWarning:

Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases later; explicitly
```

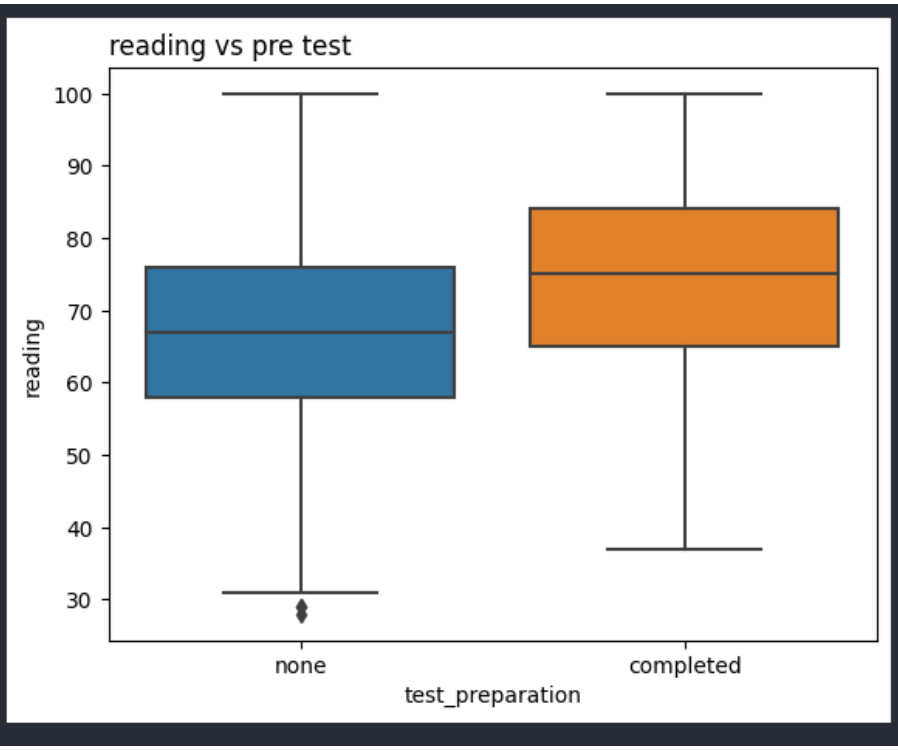


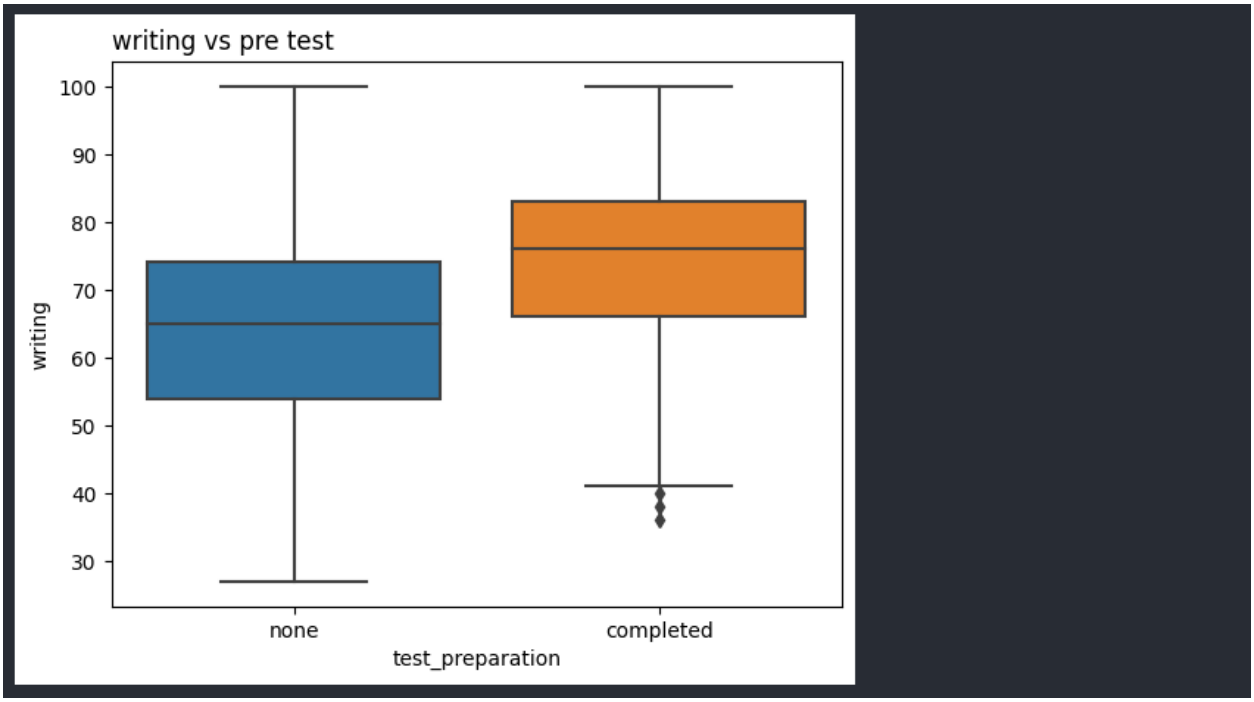
Nhóm E có kết quả học tập cao hơn hẳn so với các nhóm khác đặc biệt là nhóm A

Xem xét sự phân bố điểm dựa trên việc chuẩn bị các bài test trước kỳ thi:

```python
#Showing scores base on preparation before exams
for item in research_df.columns[-4:-1]:
    sns.boxplot(x=research_df["test_preparation"], y=research_df[item])
    plt.title(item+" vs pre test", loc="left")
    plt.show()
```





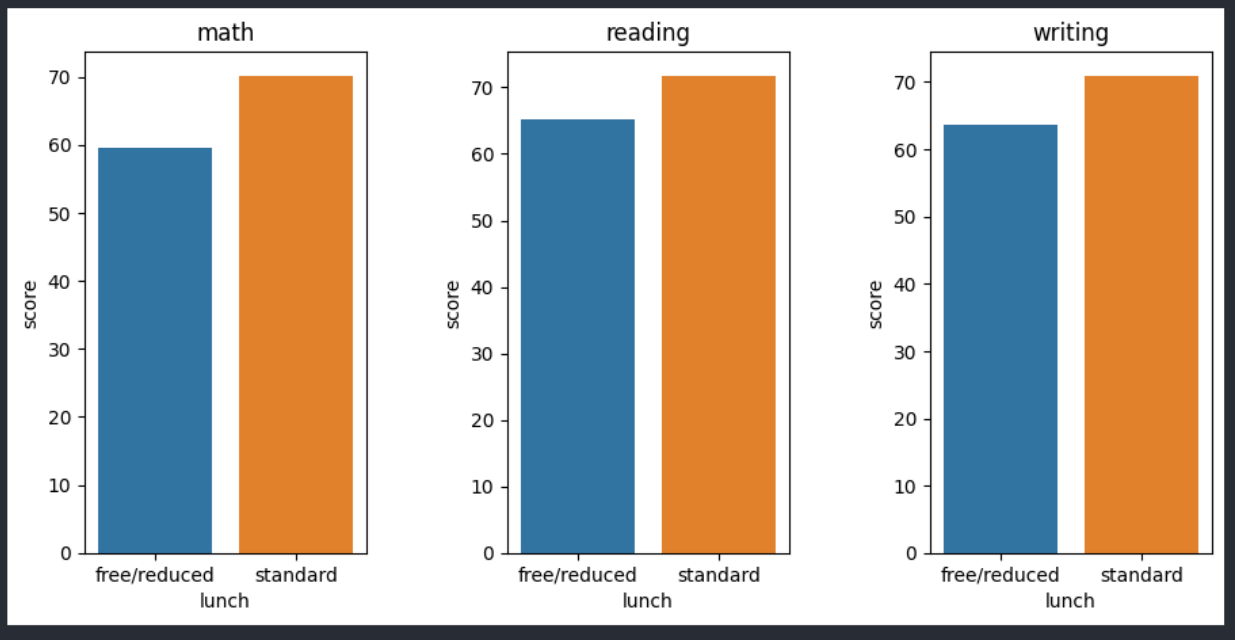


Một điều rõ ràng là những học sinh có sự chuẩn bị kỹ lưỡng hơn sẽ có kết quả thi tốt hơn

Xem xét sự phân bố điểm dựa trên dùng bữa trưa:

```python
#Showing scores base on using lunch
fig, ax = plt.subplots()
fig.subplots_adjust(hspace=0.5, wspace=0.5, left = 0.2, right = 1.5)

for i in range(3):
    plt.subplot(1,3, i+1)
    lunch_df = research_df.groupby('lunch')[list(research_df.columns[-4:-1])[i]].describe()
    sns.barplot(x=lunch_df.index, y=lunch_df.loc[:,"mean"].values)
    plt.ylabel("score")
    plt.title(list(research_df.columns[-4:-1])[i])
```

```
C:\Users\dinhh\AppData\Local\Temp\ipykernel_2180\548904083.py:6: MatplotlibDeprecationWarning:

Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases later; explicitly
```

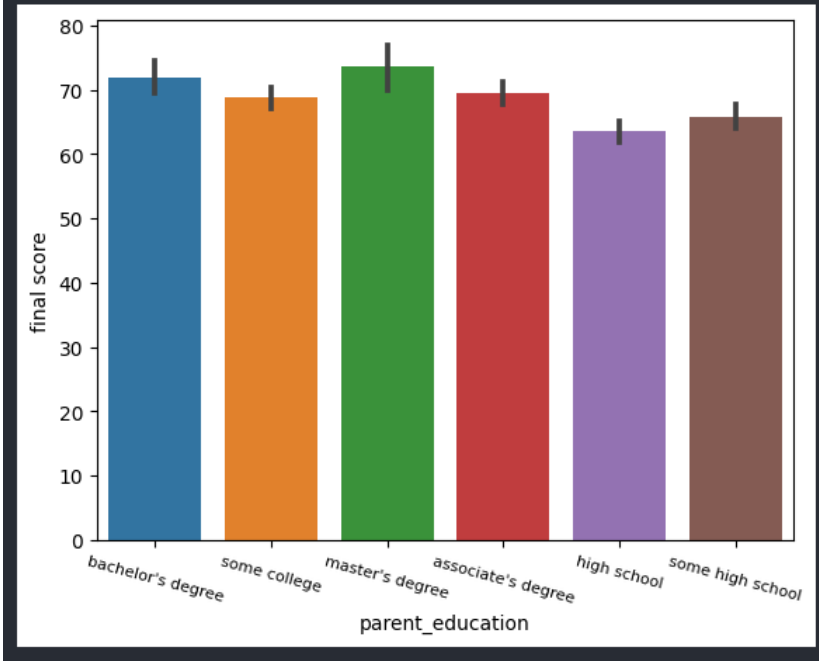


Những học sinh ăn bữa trưa đầy đủ cũng đều có kết quả học tập tốt hơn

Xem xét sự phân bố điểm dựa trên trình độ học vấn của cha mẹ:

```
#Showing scores base on levels on education of parents
plt.figure(2)
sns.barplot(data=research_df, x='parent_education', y='final score')
plt.xticks(rotation=-15, fontsize=8)
```

```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, "bachelor's degree"),
  Text(1, 0, 'some college'),
  Text(2, 0, "master's degree"),
  Text(3, 0, "associate's degree"),
  Text(4, 0, 'high school'),
  Text(5, 0, 'some high school')])
```



Trình độ học vấn của cha mẹ có vẻ không ảnh hưởng thực sự nhiều đến khả năng học tập của học sinh .

- Dựa trên những phân tích cơ bản về dữ liệu có thể nhận thấy rằng:
- Những học sinh thuộc khu vực dân tộc nhóm E có kết quả học tập tốt hơn hẳn
- Những học sinh có làm bài test trước khi thi sẽ có kết quả tốt hơn
- Những học sinh sử dụng bữa trưa chuẩn dinh dưỡng cũng có điểm thi cao hơn

- Trình độ học vấn của cha mẹ không thực sự ảnh hưởng đến kết quả học tập của con

Để có thể có được nhận định chính xác hơn và câu trả lời rõ ràng hơn cho bài toán, chúng ta cần phân cụm sinh viên dựa trên điểm các môn, sau đó khảo sát kỹ hơn trong từng cụm, tỉ lệ các thuộc tính sẽ phân bổ như nào.

b. Mô hình và giải thuật áp dụng

Để áp dụng giải thuật phân cụm cho bài toán này, có 2 đề xuất:

- Thuật toán phân cụm K-Means (Hard-clustering)
- Thuật toán phân cụm mờ Fuzzy C-means (Soft-clustering)

i. KMeans

- Đặc điểm của k-means và giải thuật

Là phương pháp phổ biến nhất trong các phương pháp phân cụm dựa trên chia cắt

Giải thuật k – means phân chia tập dữ liệu thành k cụm (k được xác định trước):

- Mỗi cụm có một điểm trung tâm gọi là centroid
- Các điểm dữ liệu thuộc cùng một cụm có sự tương tự cao với nhau, và khác biệt so với các dữ liệu thuộc cụm khác
- Đầu vào: một tập dữ liệu (các dữ liệu được biểu diễn bằng một dải các thuộc tính)
- Đầu ra: nhóm mà mỗi điểm dữ liệu thuộc vào

Các bước chính của phân cụm k-means:

- Xác định k là số các cụm
- Xác định ngẫu nhiên k ví dụ học (gọi là seeds) để làm điểm trung tâm ban đầu của k cụm đó
- Đối với mỗi ví dụ x, gán nó vào cụm có điểm trung tâm của cụm đó gần với x nhất
- Tính toán lại điểm trung tâm của cụm dựa trên các ví dụ thuộc cụm đó tại thời điểm hiện tại
- Dừng lại nếu điều kiện hội tụ thỏa mãn; nếu không, quay lại bước 3

- Các vấn đề cần chú ý của k-means
- "Gần" có ý nghĩa như thế nào?

Với một bộ dữ liệu, đặc biệt là bộ dữ liệu lớn, chúng ta không thể trực quan hóa chúng và xác định các điểm mình cho là "gần" bằng mắt. Sự "gần" hay tương tự giữa các điểm dữ liệu sẽ được đo bằng các hàm khoảng cách.

Các hàm khoảng cách thường dùng: Euclid, city-block, minkowski… Điển hình sử dụng nhiều nhất chính là hàm khoảng cách Euclid. Trong bài toán này chúng ta sử dụng khoảng cách Euclid.

Với hai điểm x, y trong không gian m chiều, công thức tính khoảng cách Euclid là:

$$d(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

Khi tính bước 3, chúng ta thực hiện tính khoảng cách từ điểm dữ liệu x đến các centroid dựa trên công thức trên.

- Điều kiện hội tụ?

Qua các vòng lặp, làm sao để biết khi nào thuật toán phân cụm k-means dừng lại?

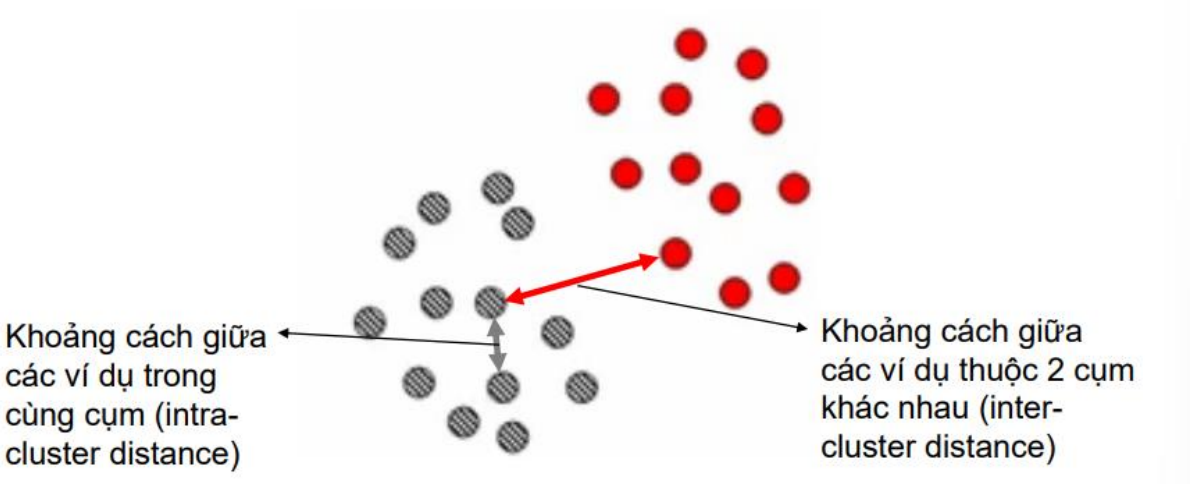
Các điều kiện hội tụ đó là: không có (hoặc rất ít) việc gán lại các ví dụ vào cụm khác; không có (hoặc rất ít) việc thay đổi các điểm trung tâm của các cụm.

Trong bài toán này, điều kiện dừng là: không có hoặc có rất ít sự gán lại các ví dụ vào cụm khác.

- Làm sao xác định được độ hiệu quả của k-means?

Thuật toán phân cụm nói chung là dạng học không có giám sát (learn by observations) vậy nên không có nhãn đầu ra trong tập học để xác minh tính hiệu quả của k-means. Việc đánh giá hiệu quả phân cụm là rất thách thức. Các nguyên tắc của phân cụm sẽ được sử dụng để đánh giá: Sự gắn kết các ví dụ trong cùng cụm là tối đa, và sự tương tự giữa các ví dụ thuộc 2 cụm khác nhau là tối thiểu.

Các phương pháp đo độ chính xác thường dùng là RMSSTD, R-squared, Dunn-index, Davies-Bouldin index.



- Xác định k thế nào là tối ưu

Việc xác định k một cách không có cơ sở sẽ dẫn đến độ chính xác của thuật toán không cao, và việc rút ra tri thức từ các cụm sẽ khó khăn. Mặt khác, k là tham số phải xác định từ ngay bước đầu tiên của thuật toán vậy nên phải xác định một k đủ tốt. Các phương pháp xác định k thường dùng là elbow method và silhouette coefficient.

Muốn biết tập dữ liệu có được phân cụm đủ tốt hay không, chúng ta sử dụng Inertia hay SSE (Sum of squared errors). Công thức tính SSE như sau:

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i_j j)} \, d\left(x^i - \mu^j\right)$$

Trong đó: n là số điểm dữ liệu, k là số cụm, x^i là ví dụ học thứ I,μ^j là centroid của cụm thứ j

$$w^{(i_j j)} = 1 \; nếu \; x^i \text{thuộc cụm j}$$

ii. Fuzzy C-Means

- Đặc điểm của Fuzzy C-means và thuật toán

Trong khi k-means là một thuật toán phân cụm cứng (hard clustering), có nghĩa là mỗi điểm dữ liệu sẽ chỉ được gán cho một cụm duy nhất, thì Fuzzy C-means là một thuật toán phân cụm mềm (soft clustering) và phân cụm mờ (fuzzy clustering)

Kết quả của Fuzzy C-means khác so với k-means: một điểm dữ liệu có thể thuộc nhiều cụm. Đầu ra của một điểm dữ liệu sau khi chạy qua thuật toán FCM sẽ là bộ xác suất.

Để dễ hiểu: giả sử có ba cụm A, B, C và một điểm dữ liệu x. Sau FCM, x được gán một bộ như sau (0,1; 0,3; 0,6) ứng với ba cụm A, B và C. Điều này có nghĩa là 10% khả năng x thuộc cụm A, 30% khả năng x thuộc cụm B và 60% khả năng x thuộc cụm C

Thuật toán FCM rất giống với k-means. Điểm khác biệt duy nhất là ở đầu ra, khi mà k-means cho đầu ra cứng còn FCM cho đầu ra "mềm mại" hơn dựa trên một chỉ số gọi là chỉ số đo mức độ thành viên (membership grade).

Membership grade có giá trị từ 0 đến 1. Nếu nó có giá trị 0 thì điểm dữ liệu đó hoàn toàn không thuộc về cụm đang xét, nếu là 1 thì nó hoàn toàn thuộc về cụm đang xét. Có thể nhìn nhận k-means cũng sử dụng membership grade này, nhưng sẽ chỉ là 0 hoặc là 1 tương ứng với việc điểm dữ liệu thuộc 1 cụm duy nhất. FCM thì membership grade có giá trị từ 0 đến 1, chỉ ra phần trăm khả năng mà nó thuộc về cụm đang xét.

Trong FCM, chúng ta đi cực tiểu hóa hàm sau:

$$\sum_{j=1}^{k} \sum_{x \in cj} u_{ij}^{m} (x_i - \mu_j)^2$$

Trong đó,

- uij chính là membership grade, là độ đo xác định xem liệu điểm dữ liệu x_i có thuộc về cụm C_(j)hay không. Tất nhiên với mỗi điểm xác định thì tổng các giá trị membership grade của nó đối với các cụm sẽ bằng 1
- μj là điểm centeroid của cụm j ; m là tham số mờ (fuzziness parameter hoặc fuzzifier).

Tham số mờ xác định độ mờ của việc phân cụm, m có giá trị thực nằm trong khoảng từ 1 đến vô cùng. Nếu m = 1, FCM trở thành phân cụm cứng như k-means.

- Giá trị uij^m được xác định bởi công thức sau:

$$u_{ij}^{m} = \frac{1}{\sum_{i=1}^{k} \left(\frac{|x_i - c_j|}{|x_i - c_k|} \right)^{\frac{2}{m-1}}}$$

- Tập hợp các giá trị uij^m tạo thành ma trận U

Ma trận U thay đổi qua từng vòng lặp

- Centroid của mỗi cụm được xác định bằng công thức sau:

$$C_j = \frac{\sum_{x \in C_j} u_{ij}^m x}{\sum_{x \in C_j} u_{ij}^m}$$

- Trong đó, Cj là centroid của cụm j
- Thuật toán FCM diễn ra như sau:

  + Chọn số cụm k và chọn tham số m. Khởi tạo ma trận U: U^((0))

  + Tính toán tâm của các cụm cho mỗi bước lặp

$$C_j = \frac{\sum_{x \in C_j} u_{ij}^m x}{\sum_{x \in C_j} u_{ij}^m}$$

  + Cập nhật lại ma trận U theo công thức

$$u_{ij}^m = \frac{1}{\sum_{i=1}^{k} \left(\frac{|x_i - c_j|}{|x_i - c_k|} \right)^{\frac{2}{m-1}}}$$

  + Kiểm tra điều kiện hội tụ: sự thay đổi của ma trận U không đáng kể

+ Nếu chưa hội tụ, quay lại bước 2.

- Các vấn đề cần chú ý của Fuzzy C-means:

- Chọn số cụm k như thế nào

Số cụm k sẽ được chọn như k-means. Chiến lược FCM áp dụng trong bài toán này sẽ là: với một điểm dữ liệu x, ta sẽ gán x cho cụm mà nó có khả năng lớn nhất thuộc vào, hay membership grade của nó với cụm đó là cao nhất. Việc này cho ta một giải thuật phân cụm cứng như k-means, từ đó cách chọn số cụm k sẽ tương tự như k-means đã trình bày ở trên

- Đo độ hiệu quả như thế nào

Như đã nói, với một điểm dữ liệu x, ta sẽ gán x cho cụm mà nó có khả năng lớn nhất thuộc vào, hay membership grade của nó với cụm đó là cao nhất. Việc này cho ta một giải thuật phân cụm cứng như k-means, từ đó cách đánh giá độ hiệu quả sẽ tương tự như k-means đã trình bày ở trên

- Xác định giá trị của siêu tham số m như thế nào

Đây là một vấn đề rất thách thức. Theo một số nghiên cứu, giá trị m nên sử dụng thuộc khoảng [1,5;4]

- Sự thay đổi "không đáng kể" là như thế nào?

Sự thay đổi không đáng kể của ma trận U cũng là vấn đề nan giải. Nó quyết định trực tiếp đến thời gian mà thuật toán chạy, số vòng lặp của thuật toán và độ chính xác. Một giá trị e được xác định, nếu mà ma trận U thay đổi không quá giá trị e thì thuật toán dừng. Vấn đề xảy ra, nếu e quá nhỏ thì thuật toán hội tụ rất chậm; nếu e quá lớn thì độ chính xác của thuật toán giảm.

Giải pháp đặt ra là chọn một e vừa đủ nhỏ, đồng thời giới hạn số vòng lặp tối đa cho thuật toán

c. Nội dung tiến hành trong mã nguồn

Sau khi thực nghiệm, kết quả cho thấy việc phân cụm khi không bao gồm giới tính và dân tộc cho kết quả tốt hơn

i. KMeans

Tối ưu số cụm k bằng phương pháp Elbow

```
#Optimize k parameter in kmeans
kmeans_dis = list()
for i in range(2, 25):
    kmeans = KMeans(init = "k-means++", n_clusters = i, n_init = i)
    kmeans.fit_transform(kmeans_df.iloc[:,2:])
    kmeans_dis.append(kmeans.inertia_)
plt.plot(list(range(2,25)), kmeans_dis, marker = "o")
plt.xlabel("Number of clusters")
plt.ylabel("SSE")
plt.show()
```

Chạy Kmeans:

```
#Apply Kmeans with k=3
X = kmeans_df[["parent_education","lunch","test_preparation","math","reading","writing"]]
kmeans = KMeans(init = "k-means++", n_clusters = 3, n_init=3)
kmeans.fit_predict(X)
kmeans_label = kmeans.labels_
kmeans_df["cluster"] = kmeans_label
centroids = kmeans.cluster_centers_
kmeans_df.head(10)
```

ii. Fuzzy C-Means

Khởi tạo tham số tương tự KMeans và thêm số vòng lặp tối đa, tham số m, các hàm tính toán

```python
#Define parameters and funtions

#number of clusters
k = 3
#maximum number of iterations
MAX_ITER = 100
#number of data points
n = len(fcm_df)
#Fuzzy parameter
m = 1.7

def initializeMembershipMatrix():
  membership_mat = []
  for i in range(n):
    random_num_list = [random.random() for i in range(k)]
    summation = sum(random_num_list)
    temp_list = [x/summation for x in random_num_list]
    flag = temp_list.index(max(temp_list))
    for j in range(0, len(temp_list)):
      if (j == flag):
        temp_list[j] = 1
      else:
        temp_list[j] = 0

    membership_mat.append(temp_list)
  return membership_mat

def calculateClusterCenter(membership_mat):
  cluster_mem_val = list(zip(*membership_mat))
  cluster_centers = []
  for j in range(k):
    x = list(cluster_mem_val[j])
    xraised = [p ** m for p in x]
    denominator = sum(xraised)
    temp_num = []
    for i in range(n):
      data_point = list(fcm_df.iloc[i, 2:])
      prod = [xraised[i] * val for val in data_point]
      temp_num.append(prod)
    numerator = map(sum, list(zip(*temp_num)))
    center = [z/denominator for z in numerator]
    cluster_centers.append(center)
  return cluster_centers
```

```
def updateMembershipValue(membership_mat, cluster_centers):
  p = float(2/(m-1))
  for i in range(n):
    x = list(fcm_df.iloc[i, 2:])
    distances = [np.linalg.norm(np.array(list(map(operator.sub, x, cluster_centers[j])))) for j in range(k)]
    for j in range(k):
      den = sum([math.pow(float(distances[j]/distances[c]), p) for c in range(k)])
      membership_mat[i][j] = float(1/den)
  return membership_mat

def getClusters(membership_mat):
  cluster_labels = list()
  for i in range(n):
    max_val, idx = max((val,idx) for (idx, val) in enumerate(membership_mat[i]))
    cluster_labels.append(idx)
  return cluster_labels

def fuzzyCMeansClustering():
    membership_mat = initializeMembershipMatrix()
    curr = 0
    while curr < MAX_ITER:
        cluster_centers = calculateClusterCenter(membership_mat)
        membership_mat = updateMembershipValue(membership_mat, cluster_centers)
        cluster_labels = getClusters(membership_mat)
        curr += 1
    return cluster_labels, cluster_centers, membership_mat
```

Chạy FCM

```
#Implement FCM
labels, centers, membership_mat = fuzzyCMeansClustering()
```

5. Đánh giá hiệu năng

- Dunn Index

Chỉ số Dunn (DI) (được giới thiệu bởi JC Dunn vào năm 1974), một số liệu để đánh giá các thuật toán phân cụm, là một sơ đồ đánh giá nội bộ, trong đó kết quả dựa trên chính dữ liệu được phân cụm. Giống như tất cả các chỉ số khác như vậy, mục đích của chỉ số Dunn này để xác định các tập hợp các cụm nhỏ gọn, với phương sai nhỏ giữa các thành viên của cụm và được phân tách

tốt, trong đó các phương tiện của các cụm khác nhau đủ xa nhau, so với phương sai trong cụm.

Giá trị chỉ số Dunn càng cao, hiệu quả phân cụm càng tốt. Số lượng cụm tối đa hóa chỉ số Dunn được lấy làm số cụm tối ưu k. Nó cũng có một số nhược điểm. Khi số lượng cụm và tính chiều hướng của dữ liệu tăng lên, chi phí tính toán cũng tăng lên.

- Davies Bouldin Index

Chỉ số Davies–Bouldin (DBI) (được giới thiệu bởi David L. Davies và Donald W. Bouldin vào năm 1979), một số liệu để đánh giá các thuật toán phân cụm, là một sơ đồ đánh giá nội bộ, trong đó việc xác nhận mức độ phân cụm đã được thực hiện bằng cách sử dụng số lượng và tính năng vốn có của tập dữ liệu.

DB càng thấp, hiệu quả phân cụm càng tốt. Nó cũng có một nhược điểm. Một giá trị tốt được báo cáo bằng phương pháp này không ngụ ý truy xuất thông tin tốt nhất.

- Khai báo hàm tính toán hiệu năng, hai độ đo sử dụng là Dunn Index và Davies Bouldin Index

```python
#Define functions

def getInnerDistance(cluslabels,centre,data):
    innerDis = np.full(len(centre),0.0)
    for i in range(len(data)):
        cen = cluslabels[i]
        innerDis[cen] += np.linalg.norm(data[i]-centre[cen])

    for i in range(len(centre)):
        innerDis[i] = innerDis[i]/(cluslabels.count(i))
    return innerDis

def getDunnIndex(cluslabels,centre,data):
    inter = np.inf
    for i in range(len(centre)):
        for j in range(i+1,len(centre)):
            temp = np.linalg.norm(centre[i]-centre[j])
            if inter > temp:
                inter = temp

    intra = 0
    for i in range(len(data)):
        for j in range(i+1,len(data)):
            if cluslabels[i] == cluslabels[j]:
                temp = np.linalg.norm(data[i]-data[j])
                if intra < temp:
                    intra = temp
    dunnIdx = inter/intra
    return dunnIdx

def getDB(cluslabels,centre,data):
    innerDis = getInnerDistance(cluslabels,centre,data)
    max = 0
    for i in range(len(centre)):
        for j in range(i+1,len(centre)):

            score = (innerDis[i]+innerDis[j])/np.linalg.norm(centre[i]-centre[j])
            if max < score:
                max = score
    return max
```

Đánh giá hiệu năng KMeans và FCM

```python
#Check Kmeans
print("DB index =", getDB(list(kmeans_df['cluster']), list(centroids), list(kmeans_df.iloc[:,2:-2].values)))
print("Dunn index =", getDunnIndex(list(kmeans_df['cluster']), list(centroids), list(kmeans_df.iloc[:,2:-2].values)))
```

✓ 0.9s

```
DB index = 0.8229072038328629
Dunn index = 0.5038196727167851
```

```python
#Check FCM
print("DB index =", getDB(list(fcm_df['cluster']), np.array(centers), list(fcm_df.iloc[:,2:-2].values)))
print("Dunn index =", getDunnIndex(list(fcm_df['cluster']), np.array(centers), list(fcm_df.iloc[:,2:-2].values)))
```

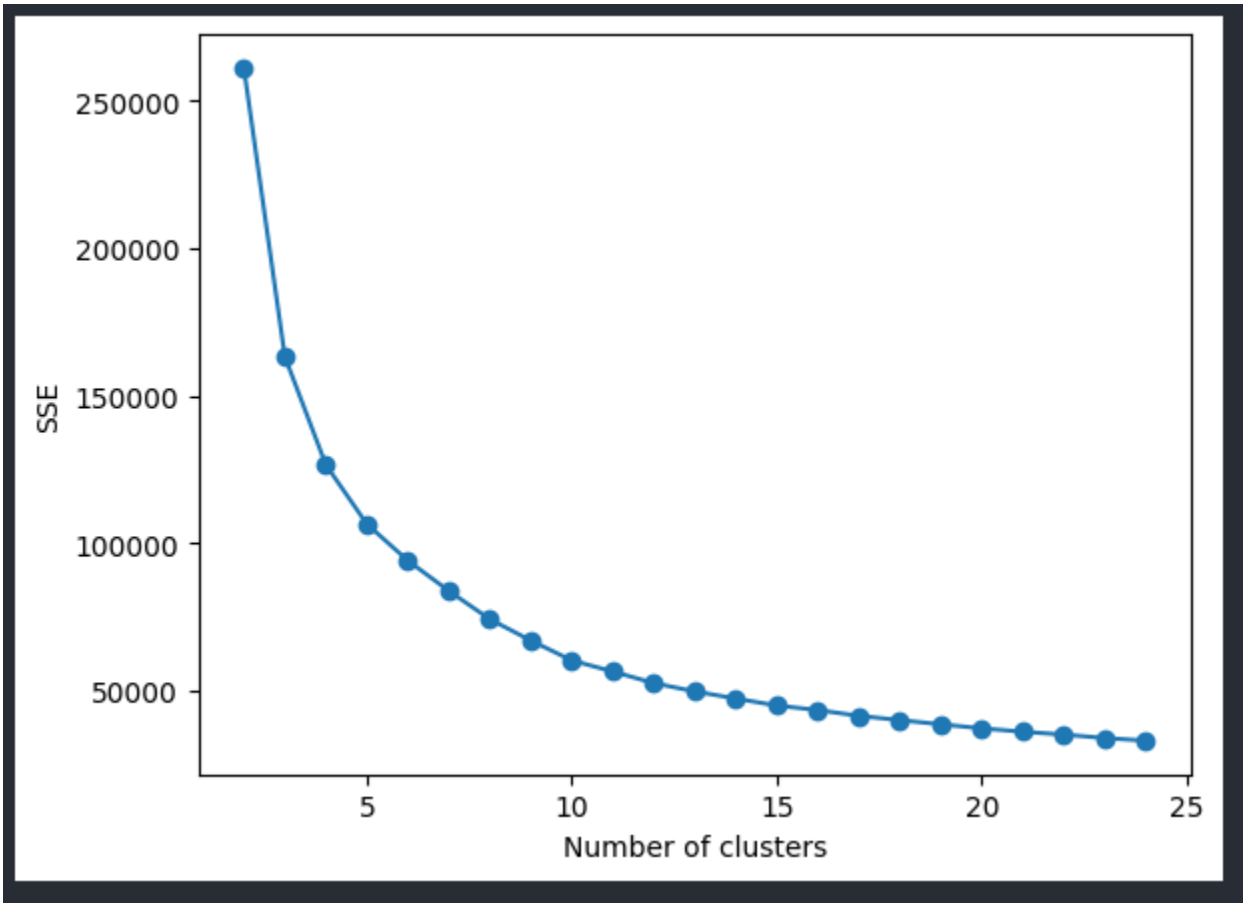
✓ 0.8s

```
DB index = 0.8133023997329732
Dunn index = 0.5099183939594114
```

Có thể thấy nếu sử dụng 2 độ đo hiệu năng này, chất lượng phân cụm của 2 thuật toán là tương đương nhau, vì vậy có thể sử dụng kết quả của một trong hai để phân tích cụm sau này.

6. Kết quả
    a. Các kết quả phân cụm

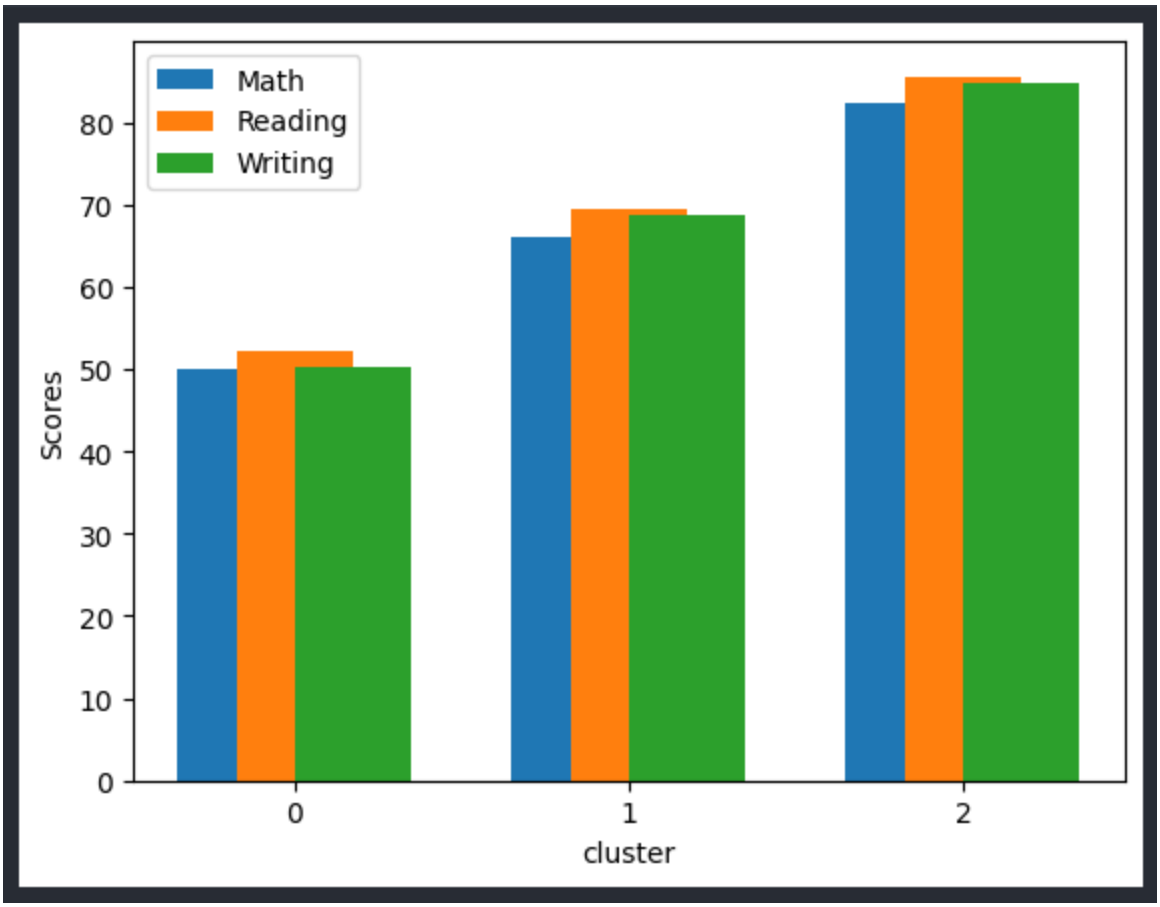
    Số cụm tối ưu là 3

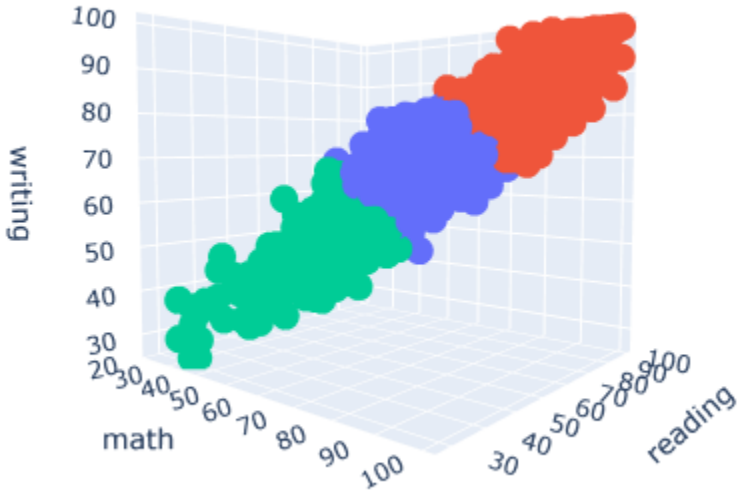


    Điểm trung bình 3 môn của mỗi cụm

cluster	math	reading	writing
0	49.951852	52.185185	50.222222
1	66.080275	69.561927	68.839450
2	82.432056	85.595819	84.777003

Vì tương quan giữa điểm 3 môn là tốt như trên khi phân tích dữ liệu, nên các cụm cũng có tương quan các điểm tốt và tương đương nhau


Math
Reading
Writing
Scores
0
10
20
30
40
50
60
70
80
0
1
2
cluster


writing
100
90
80
70
60
50
40
30
20
30
40
50
60
70
80
90
100
math
30
40
50
60
70
80
90
100
reading

Để có thể kiểm chứng rõ hơn những nhận định đã đưa ra và có câu trả lời chính xác, hãy phân tích dữ liệu từng cụm. Để phân tích dễ dàng, chúng ta sẽ sắp xếp cụm theo điểm trung bình giảm dần từ Rank 0 đến Rank 2

```
#Calculate average scores in each clusters
class_df["total_ave_score"] = (class_df.math + class_df.reading + class_df.writing)/3
rank = class_df["total_ave_score"].sort_values(ascending = False)
rank.index
```

✓ 0.4s

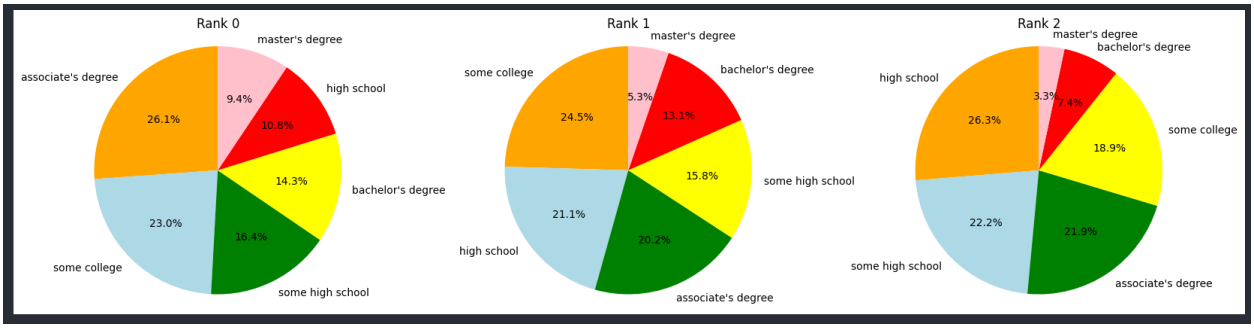
```
Int64Index([2, 1, 0], dtype='int64', name='cluster')
```

```
rank
```

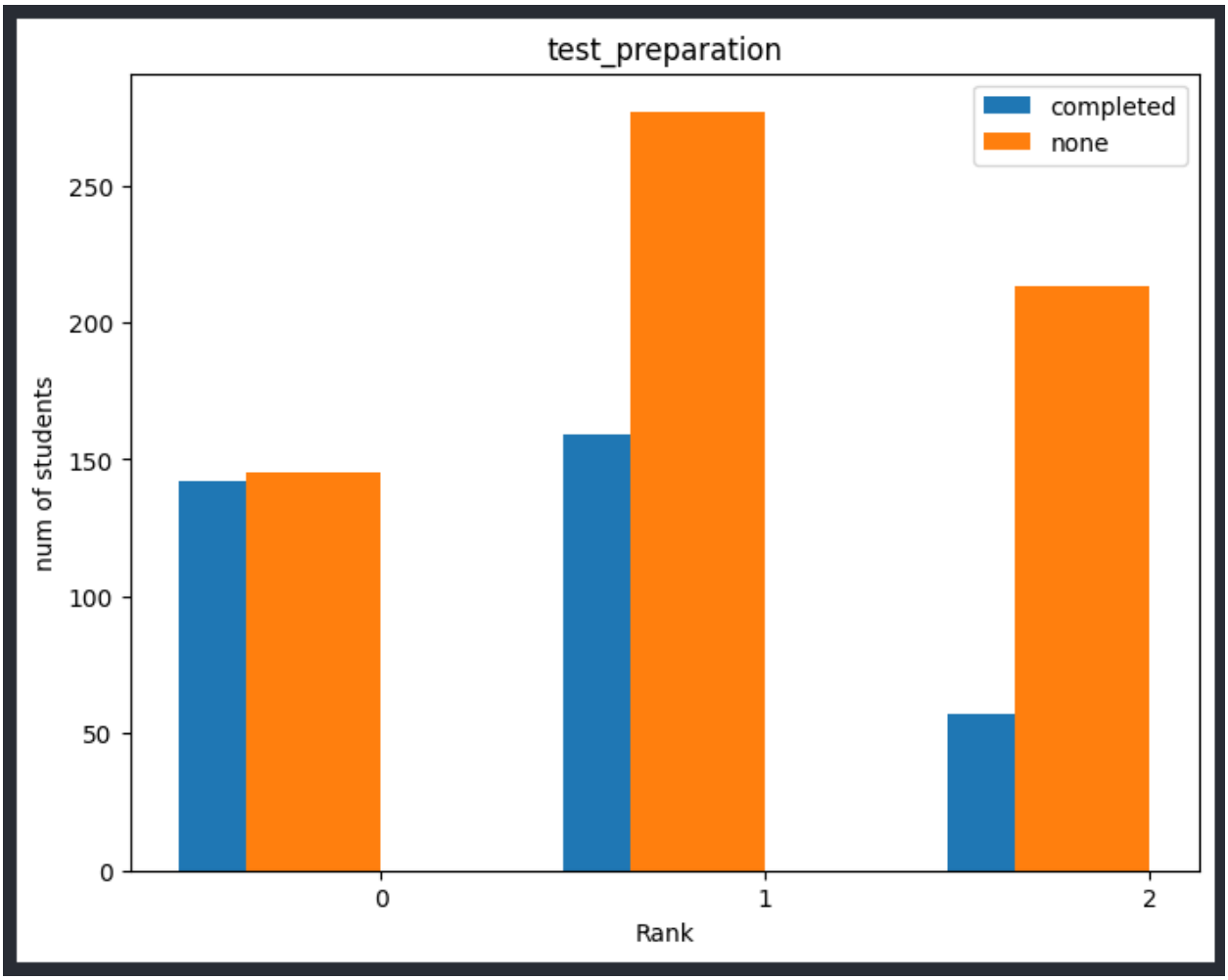
✓ 0.6s

```
cluster
2    84.268293
1    68.160550
0    50.786420
Name: total_ave_score, dtype: float64
```

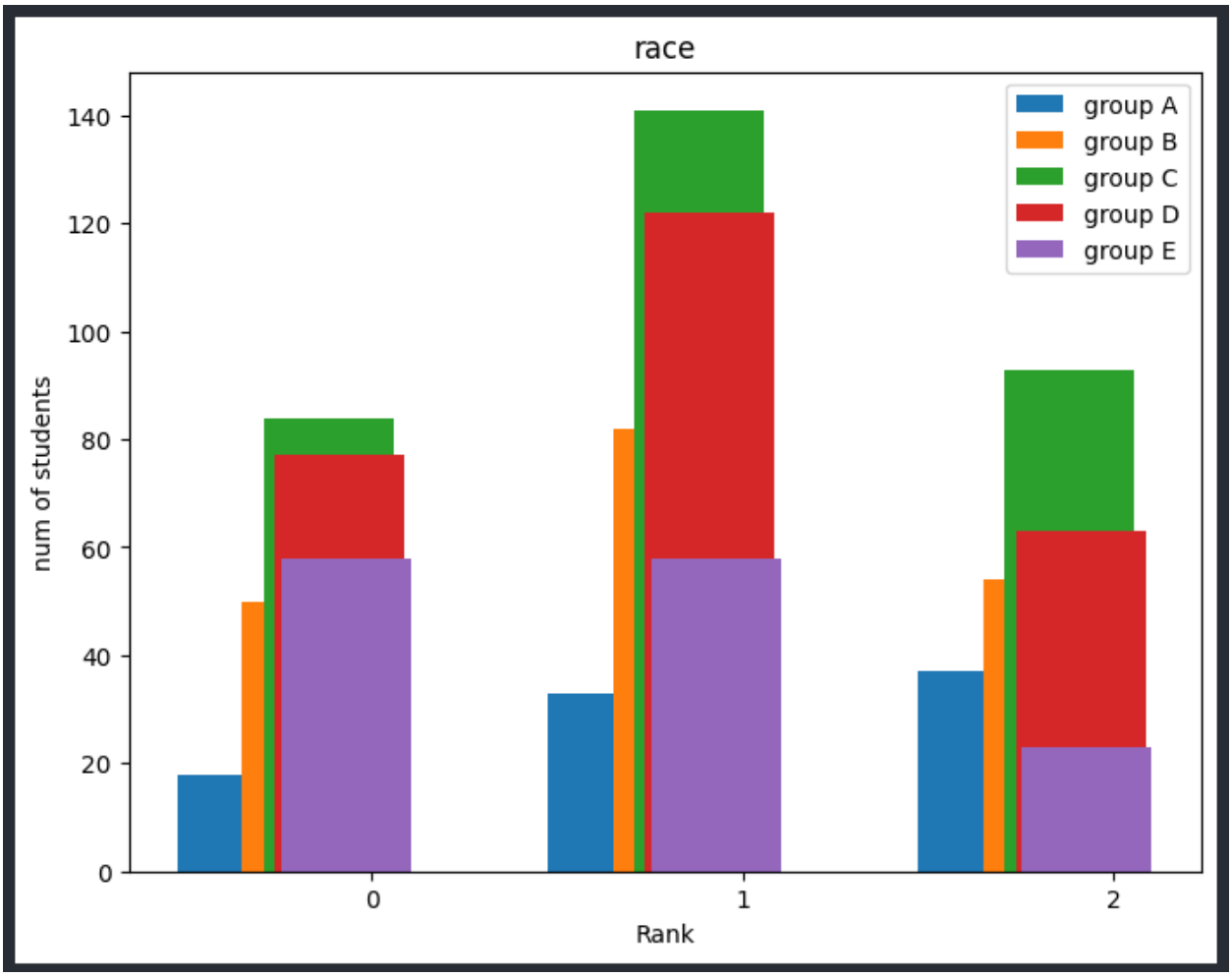
Tỉ lệ trình độ giáo dục của cha mẹ trong từng cụm



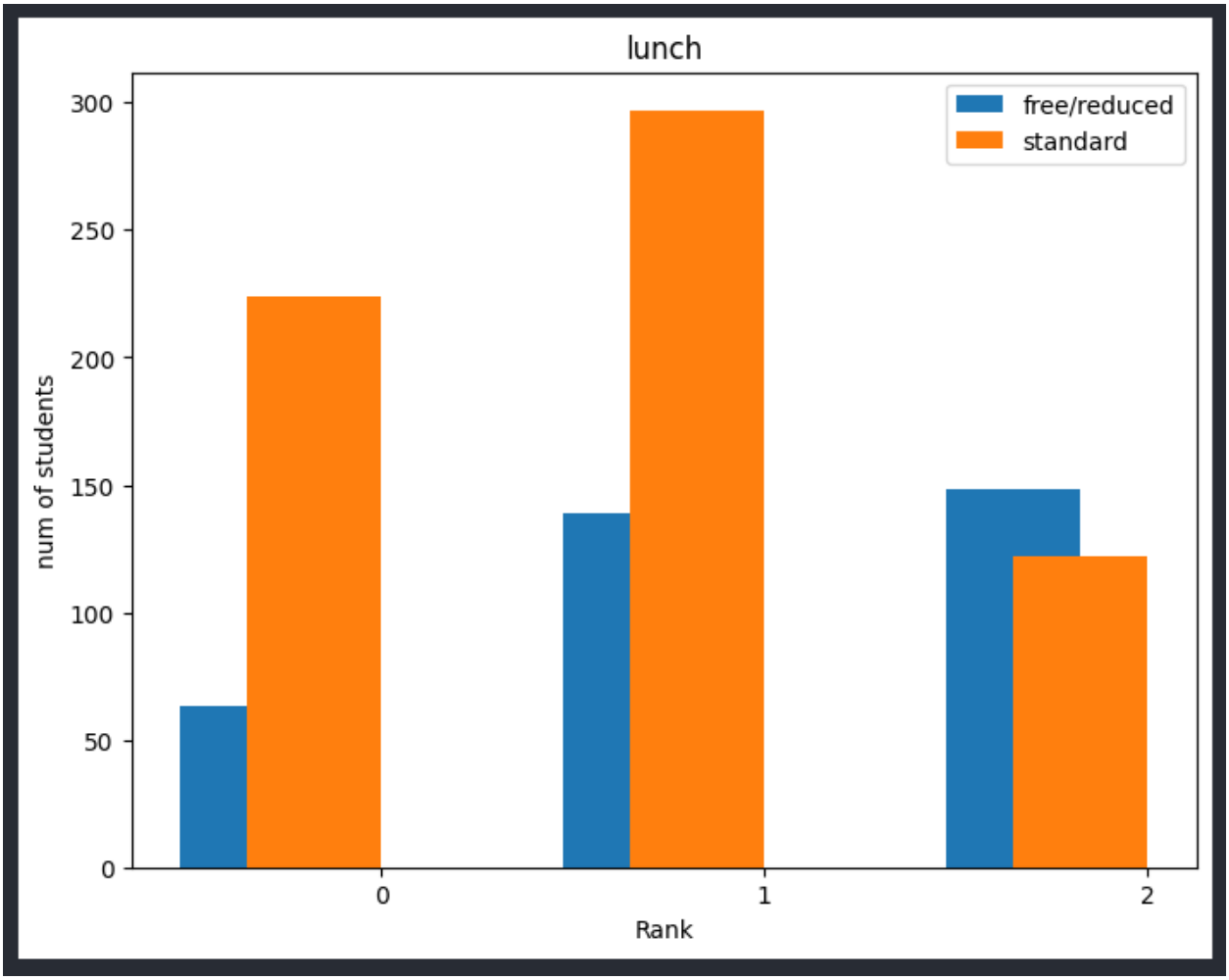
Tỉ lệ chuẩn bị test trước bài kiểm tra



Tỉ lệ dân tộc trong các cụm:



Còn bữa trưa thì sao?



b. Những tri thức rút ra được

Qua phân tích dữ liệu cụm trên có thể đưa ra các kết luận sau:

- Trình độ học vấn của cha mẹ có ảnh hưởng không nhiều đến kết quả học tập của con cái, dễ thấy gần 70% học sinh có cha mẹ ở trình độ giáo dục thấp và trung bình vẫn có kết quả học tập tốt
- Điều này cũng chính xác đối với khu vực dân tộc học sinh đó sinh sống, khi ở 3 cụm phân bố số lượng dân tộc đều tương tự nhau

- Học sinh có sự chuẩn bị làm bài test sẽ có kết quả học tập tốt hơn, 50% số lượng có chuẩn bị bài test thuộc Rank 0 và con số này chênh lệch càng ngày càng lớn ở Rank sau
- Học sinh có dùng bữa trưa chuẩn dinh dưỡng cũng có thể có kết quả tốt hơn, vì tỉ lệ % học sinh dùng bữa trưa tự do tăng dần ở những rank thấp

Và kết luận chung: có thể thấy rằng hoàn cảnh của học sinh không ảnh hưởng lớn đến khả năng học tập của họ mà phần lớn nằm ở nỗ lực, học sinh chuẩn bị bài test trước khi thi và có một chế độ dinh dưỡng phù hợp sẽ có kết quả học tập tốt hơn.

7. Khó khăn và hướng phát triển trong tương lai
    a. Khó khăn

- Khó khăn trong việc tìm kiếm thêm dữ liệu để kiểm chứng kết quả
- Những thuộc tính như chuẩn bị bài hay chế độ ăn trưa có thể không chính xác và làm sai lệch kết quả
- Việc đánh giá hiệu năng của phân cụm là khá khó, nên chưa thể đánh giá tốt độ chính xác của việc phân cụm
- Việc sử dụng các công cụ hay ngôn ngữ chưa thành thạo, cụ thể là Python cũng là một hạn chế lớn

    b. Phát triển trong tương lai

- Kết hợp với thuật toán phân loại để có được các dự đoán và nhận định tốt hơn
- Nâng cao khả năng làm chủ công nghệ, các thư viện về visualize data, các thuật toán để có những góc nhìn trực quan hơn

8. Tài liệu tham khảo

- Tập bài giảng nhập môn học máy và khai phá dữ liệu - PGS.TS. Thân Quang Khoát
- https://scikit-learn.org/stable/modules/clustering.html
- https://www.geeksforgeeks.org/dunn-index-and-db-index-cluster-validity-indices-set-1/
- https://thesai.org/Downloads/SpecialIssueNo3/Paper%2022-Clustering%20Student%20Data%20to%20Characterize%20Performance%20Patterns.pdf
- https://www.kaggle.com/datasets/spscientist/students-performance-in-exams
- https://vitalflux.com/k-means-elbow-point-method-sse-inertia-plot-python/
- https://www.researchgate.net/publication/300483722_On_the_selection_of_m_for_Fuzzy_c-Means