ĐẠI HỌC UEH

TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ UEH

KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH

ĐỒ ÁN KẾT THÚC HỌC PHẦN

MÔN BIỂU DIỄN TRỰC QUAN DỮ LIỆU

"Lung Cancer Predict"

Giảng viên: Ts. Nguyễn An Tế

Mã lớp học phần: 23C1INF50908202

Nhóm: 6

Danh sách sinh viên:

1. Trương Vũ Phương Quỳnh – 31211027668
2. Đinh Công Thành – 31211027670
3. Nguyễn Thị Thơm – 31211027673
4. Đào Bùi Hương Thùy - 31211027675

TP Hồ Chí Minh, ngày 25 tháng 11 năm 2023

MỤC LỤC

I. Giới thiệu đề tài

Đề tài "Lung Cancer Predict" cho đồ án biểu diễn trực quan dữ liệu sử dụng ngôn ngữ Python nhằm nghiên cứu và phân tích về bệnh ung thư phổi. Dữ liệu được sử dụng trong đồ án này được thu thập từ Kaggle, chứa các thông tin về các yếu tố có thể liên quan đến nguy cơ mắc ung thư phổi và dự đoán nguyên nhân mắc bệnh của bệnh nhân.

Nội dung của đề tài bao gồm các bước sau:

1. Thu thập dữ liệu: Dữ liệu được tải xuống từ nguồn Kaggle: Lung Cancer Prediction
2. Tiền xử lý dữ liệu: Dữ liệu thu thập được sẽ được xử lý để loại bỏ các thông tin không cần thiết, kiểm tra tính đúng đắn và chuẩn hóa định dạng. Nhóm sử dụng các thư viện như pandas và numpy để hỗ trợ.
3. Phân tích và biểu diễn trực quan dữ liệu: Dữ liệu sau khi được xử lý sẽ được phân tích để đưa ra các nhận định và kết luận về bệnh nhân ung thư. Các phân tích thống kê, biểu đồ và mô hình dữ liệu có thể được tạo ra để trực quan hóa và phân tích dữ liệu. Nhóm sử dụng các thư viện như matplotlib và seaborn có thể hỗ trợ trong việc này.
4. Nhận định và kết luận: Kết quả phân tích sẽ được sử dụng để dự đoán khả năng bệnh nhân mắc ung thư phổi, tình trạng bệnh của bệnh nhân, xác định các yếu tố nguy cơ ung thư phổi.

Phân tích biểu diễn trực quan dữ liệu sẽ giúp hiểu rõ hơn về bệnh ung thư phổi. Kết quả phân tích thu được sẽ sử dụng mục tiêu giúp người sử dụng hiểu rõ hơn về tập dữ liệu và tìm ra các mẫu, xu hướng hoặc liên hệ tiềm năng giữa các yếu tố khác nhau để dự đoán ung thư phổi. Hy vọng rằng đồ án này sẽ mang lại kiến thức và cung cấp một cách tiếp cận mới trong việc biểu diễn trực quan dữ liệu y tế.

II. Phương pháp thực hiện

A. Mô tả bộ dữ liệu

Bộ dữ liệu mà nhóm sử dụng chứa thông tin về các bệnh nhân ung thư phổi, được chia thành 3 mức bệnh là Low, Medium và High. Các 23 biến mà bộ dữ liệu thu thập được như sau:

Tên biến	Loại dữ liệu	Giá trị	Mô tả
Age	Định lượng	Tuổi từ 14-73, chia thành 4 nhóm 1,2,3,4 là 14-28, 29-43, 44-58, 59-73	Tuổi của bệnh nhân.
Gender	Định tính	Nam- 1, Nữ -2	Giới tính của bệnh nhân.
Air Pollution	Định tính	Có 8 mức từ 1 đến 8.	Mức độ tiếp xúc với ô nhiễm không khí của bệnh nhân.
Alcohol use	Định tính	Có 8 mức từ 1 đến 8.	Mức độ sử dụng rượu của bệnh nhân.
Dust Allergy	Định tính	Có 8 mức từ 1 đến 8.	Mức độ dị ứng bụi của người bệnh.
OccuPational Hazards	Định tính	Có 8 mức từ 1 đến 8.	Mức độ nguy hiểm nghề nghiệp của bệnh nhân.
Genetic Risk	Định tính	Có 8 mức từ 1 đến 8.	Mức độ rủi ro di truyền của bệnh nhân.
chronic Lung Disease	Định tính	Có 8 mức từ 1 đến 8.	Mức độ bệnh phổi mãn tính của bệnh nhân.
Balanced Diet	Định tính	Có 8 mức từ 1 đến 8.	Mức độ ăn uống cân bằng của bệnh nhân.
Obesity	Định tính	Có 8 mức từ 1 đến 8.	Mức độ béo phì của bệnh nhân.

Smoking	Định tính	Có 8 mức từ 1 đến 8.	Mức độ hút thuốc của bệnh nhân.
Passive Smoker	Định tính	Có 8 mức từ 1 đến 8.	Mức độ hút thuốc thụ động của bệnh nhân.
Chest Pain	Định tính	Có 8 mức từ 1 đến 8.	Mức độ đau ngực của bệnh nhân.
Coughing of Blood	Định tính	Có 9 mức từ 1 đến 9.	Mức độ ho ra máu của người bệnh.
Fatigue	Định tính	Có 9 mức từ 1 đến 9.	Mức độ mệt mỏi của bệnh nhân.
Weight Loss	Định tính	Có 8 mức từ 1 đến 8.	Mức độ giảm cân của bệnh nhân.
Shortness of Breath	Định tính	Có 9 mức từ 1 đến 9.	Mức độ khó thở của bệnh nhân.
Wheezing	Định tính	Có 8 mức từ 1 đến 8.	Mức độ thở khò khè của bệnh nhân.
Swallowing Difficulty	Định tính	Có 8 mức từ 1 đến 8.	Mức độ khó nuốt của bệnh nhân.
Clubbing of Finger Nails	Định tính	Có 9 mức từ 1 đến 9.	Mức độ ngón tay bị dùi trống của người bệnh
Frequent Cold	Định tính	Có 8 mức từ 1 đến 8.	Mức độ thường xuyên cảm lạnh của bệnh nhân.
Dry Cough	Định tính	Có 8 mức từ 1 đến 8.	Mức độ ho khan của bệnh nhân.

Snoring	Định tính	Có 8 mức từ 1 đến 8.	Mức độ ngủ ngáy của bệnh nhân.
Level	Định tính	Low - 1, Medium - 2, High - 3.	Mức độ bệnh của bệnh nhân

Bảng 1: Mô tả bộ dữ liệu

Trong bộ dữ thu được 1000 dòng tương ứng với từng người mắc bệnh ở Trung Quốc.

B. Phương pháp thực hiện

Trước khi biểu diễn dữ liệu, việc tiền xử lý dữ liệu, xử lý các dữ liệu bị thiếu và các bất thường, bộ dữ liệu không bị thiếu hay có outliers.

Để biểu diễn bộ dữ liệu trên nhóm đã sử dụng ngôn ngữ Python và các thư viện matplotlib, seaborn,... . Từ đó thể hiện tính chất của các biến, mối tương quan giữa các biến với nhau và tương quan giữa các biến với biến mục tiêu. Những thông tin được biểu diễn bằng các loại biểu đồ: bar chart, scatter plot, heat map, pie chart,.....

Thực hiện phân lớp, dùng 4 phương pháp KNN classification, Decision Tree, Support Vector Machine, Naive Bayes trong thư viện Sklearn.

III. Tiền xử lý dữ liệu

A. Xử lý dữ liệu bị thiếu và outliers

1. Kiểm tra dữ liệu bị thiếu

Sử dụng hàm info() để kiểm tra các biến của dữ liệu

```
data.info()
```

#	Column	Non-Null Count	Dtype
0	index	1000 non-null	int64
1	Patient Id	1000 non-null	object
2	Age	1000 non-null	int64
3	Gender	1000 non-null	int64
4	Air Pollution	1000 non-null	int64
5	Alcohol use	1000 non-null	int64
6	Dust Allergy	1000 non-null	int64
7	OccuPational Hazards	1000 non-null	int64
8	Genetic Risk	1000 non-null	int64
9	chronic Lung Disease	1000 non-null	int64
10	Balanced Diet	1000 non-null	int64
11	Obesity	1000 non-null	int64
12	Smoking	1000 non-null	int64
13	Passive Smoker	1000 non-null	int64
14	Chest Pain	1000 non-null	int64
15	Coughing of Blood	1000 non-null	int64
16	Fatigue	1000 non-null	int64
17	Weight Loss	1000 non-null	int64
18	Shortness of Breath	1000 non-null	int64
19	Wheezing	1000 non-null	int64
20	Swallowing Difficulty	1000 non-null	int64
21	Clubbing of Finger Nails	1000 non-null	int64
22	Frequent Cold	1000 non-null	int64
23	Dry Cough	1000 non-null	int64
24	Snoring	1000 non-null	int64
25	Level	1000 non-null	object

Hình 1: Các biến của dữ liệu

Tất cả các cột của bộ dữ liệu thu thập đều chứa đầy đủ các giá trị, không chứa giá trị bị thiếu.

2. Kiểm tra dữ liệu có tồn tại outlier hay không

```
a = data.drop(['Age','index','Patient Id'],axis = 1)
for i in a:
  j = data[i].unique().tolist()
  print('Danh sách ',i,' trong data', j)
```

```
Danh sách  Gender  trong data [1, 2]
Danh sách  Air Pollution  trong data [2, 3, 4, 7, 6, 5, 1, 8]
Danh sách  Alcohol use  trong data [4, 1, 5, 7, 8, 3, 6, 2]
Danh sách  Dust Allergy  trong data [5, 6, 7, 4, 2, 8, 1, 3]
Danh sách  OccuPational Hazards  trong data [4, 3, 5, 7, 2, 6, 8, 1]
Danh sách  Genetic Risk  trong data [3, 4, 5, 6, 7, 2, 1]
Danh sách  chronic Lung Disease  trong data [2, 4, 7, 6, 3, 5, 1]
Danh sách  Balanced Diet  trong data [2, 6, 7, 4, 5, 3, 1]
Danh sách  Obesity  trong data [4, 2, 7, 3, 5, 6, 1]
Danh sách  Smoking  trong data [3, 2, 7, 8, 1, 6, 5, 4]
Danh sách  Passive Smoker  trong data [2, 4, 3, 7, 6, 8, 5, 1]
Danh sách  Chest Pain  trong data [2, 4, 7, 3, 6, 5, 9, 8, 1]
Danh sách  Coughing of Blood  trong data [4, 3, 8, 9, 1, 5, 7, 6, 2]
Danh sách  Fatigue  trong data [3, 1, 8, 4, 5, 9, 2, 6]
Danh sách  Weight Loss  trong data [4, 3, 7, 2, 6, 5, 1, 8]
Danh sách  Shortness of Breath  trong data [2, 7, 9, 3, 4, 5, 6, 1]
Danh sách  Wheezing  trong data [2, 8, 1, 4, 6, 7, 5, 3]
Danh sách  Swallowing Difficulty  trong data [3, 6, 1, 4, 2, 5, 8, 7]
Danh sách  Clubbing of Finger Nails  trong data [1, 2, 4, 5, 6, 8, 7, 9, 3]
Danh sách  Frequent Cold  trong data [2, 1, 6, 4, 3, 7, 5]
Danh sách  Dry Cough  trong data [3, 7, 2, 4, 1, 5, 6]
Danh sách  Snoring  trong data [4, 2, 5, 3, 1, 6, 7]
Danh sách  Level  trong data ['Low', 'Medium', 'High']
```

Hình 2: Kiểm tra dữ liệu

Danh sách các dữ liệu ở các cột đều liên tục không tồn tại các giá trị quá cao hay quá thấp đột biến so với tổng thể data. Nhóm kết luận dữ liệu không có các dữ liệu outlier cần phải xử lý.

B. Chuyển dạng dữ liệu

1. Xoá các cột định danh

Các cột định danh không có ý nghĩa trong phân tích xây dựng mô hình dự đoán, nhóm quyết định xóa đi hai cột ‘index’ và ‘Patient Id’

```
data = data.drop(['index'], axis = 1)
data = data.drop(['Patient Id'], axis = 1)
```

2. Chuyển dạng dữ liệu

Các cột dữ liệu features đều đã ở định dạng int64, chỉ có cột dữ liệu target ‘Level’ ở định dạng object chứa các giá trị ‘Low’, ‘Medium’ và ‘High’ nhóm quyết định chuyển dữ liệu cột ‘Level’ đưa các giá trị ‘Low’ -> 1, ‘Medium’ -> 2, ‘High’ ->3

```
map = {'Low': 1,'Medium': 2,'High': 3}
data['Level'] = data['Level'].map(map)
```

Cột ‘Age’ có các giá trị chạy từ 14 đến 73 để thuận tiện cho việc xây dựng mô hình và phân tích, nhóm quyết định rời rạc hóa dữ liệu cột ‘Age’ thành 4 lớp ‘14-28’, ‘29-43’, ‘44-58’, ‘59-73’ sau đó chuyển thành các nhóm tuổi lần lượt thành 1, 2, 3, 4 để thuận tiện cho việc xây dựng mô hình.

```
a = pd.qcut(data['Age'], 4 , labels = ["14-28", "29-43", "44-58", "59-73"])
data['Age'] = a
data.rename(columns = {'Age': 'Age Group'}, inplace=True)
map = {'14-28': 1,'29-43': 2,'44-58': 3, '59-73': 4}
data['Age Group'] = data['Age Group'].map(map)
```

IV. Biểu diễn trực quan dữ liệu

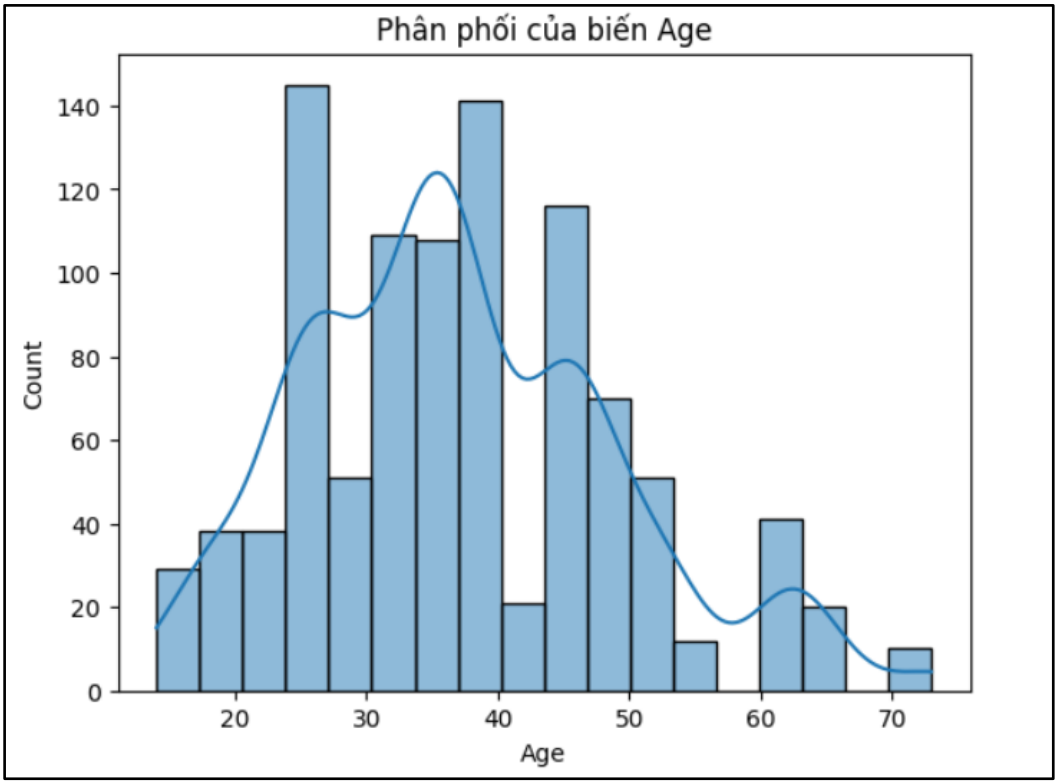
A. Trực quan đơn biến

1. Trực quan dữ liệu biến định lượng

Dùng biểu đồ histogram để theo dõi sự phân bổ của biến định lượng Age trong tập dữ liệu:

```
sbn.histplot(data['Age Group'], kde = True)
plt.title('Phân phối của biến Age')
plt.show()
```

Code biểu đồ phân phối biến định lượng Age

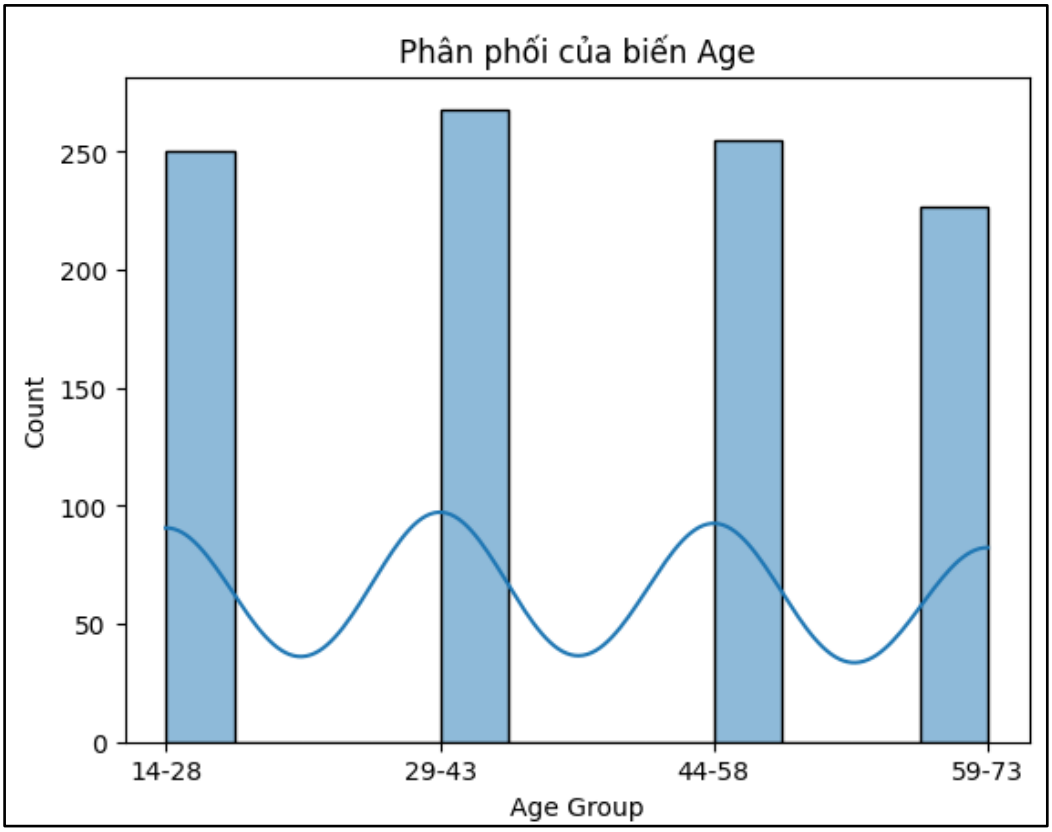


Hình 3: Biểu đồ phân phối biến định lượng Age

Theo biểu đồ phân phối biến Age, đây là biểu đồ trước khi biến thực hiện rời rạc hóa dữ liệu, đa số bệnh nhân mắc bệnh ung thư nằm trong độ tuổi từ 25 đến 45 (trên 80 bệnh nhân/ tuổi). Tiếp theo là độ tuổi từ 14 đến 24 và 45 đến 65. Và thấp nhất là nhóm từ 66 - 73. Nhìn chung, biểu đồ cho thấy bệnh nhân trong độ tuổi còn khá trẻ, đây cũng là khi con người tham gia vào các hoạt động xã hội và nghề nghiệp. Vì vậy, xác định các nhân tố gây bệnh là vấn đề cần hướng đến.

Để thực hiện phân tích thuận lợi hơn, nhóm đã thực hiện rời rạc hóa dữ liệu và vẽ được biểu đồ Age Group sau:

```
tick_labels = ['14-28', '29-43', '44-58', '59-73']

# Vẽ histogram và
sbn.histplot(data['Age Group'], kde=True)
plt.title('Phân phối của biến Age')
# Thay đổi tên các cột xuất hiện trên trục x
plt.gca().set_xticks([1, 2, 3, 4])  # Đặt vị trí của ticks
plt.gca().set_xticklabels(tick_labels)  # Đặt nhãn cho ticks

plt.show()
```



Hình 4: Biểu đồ phân phối biến định lượng Age Group

Theo biểu đồ phân phối biến định lượng Age Group, sau khi thực hiện rời rạc hóa dữ liệu, số lượng bệnh nhân ở mỗi nhóm tuổi được phân phối khá đồng đều. Từ đây, ta có thể phân tích các yếu tố gây ung thư đạt hiệu quả tốt hơn.

2. Trực quan biến định tính

```
cat_col = data.select_dtypes(['int', 'float']).columns.drop("Age
Group", "Level")

cat_plots = len(cat_col)
rows = int(cat_plots / 3) + (cat_plots % 3 > 0)
cols = 3 if cat_plots > 3 else cat_plots

fig, axes = plt.subplots(rows, cols, figsize=(22, 24))
axes = axes.reshape(-1)
for i, ax in enumerate(axes):
    if i < len(cat_col):
        sbn.histplot(data[cat_col[i]], kde=True, ax=ax)
        ax.set_title(f'Phân phối của biến {cat_col[i]}')
axes[-2].remove()
axes[-1].remove()
```

```
plt.tight_layout()
plt.show()
```

Hình 5: Phân phối các biến

Dựa vào biểu đồ phân phối, nhận thấy rằng biến Gender là biến định danh duy nhất chỉ có 2 cột vì đây là biến giới tính 1: Nam và 2: Nữ. Từ biểu đồ biến Gender thấy rằng sự chênh lệch nhóm bệnh nhân nam cao hơn so với bệnh nhân nữ khá rõ ràng (Nam khoảng 600 bệnh nhân, nữ khoảng 400 bệnh nhân, chênh lệch khoảng 200). Từ đây, có thể nhận xét rằng giới tính có ảnh hưởng tới mức độ mắc bệnh của bệnh nhân.

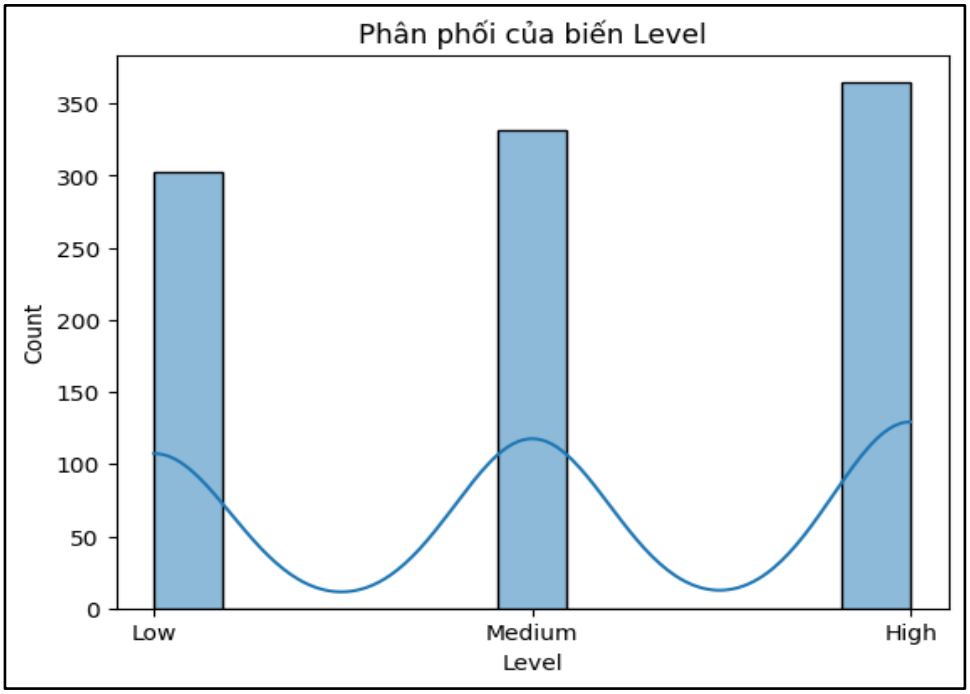
Ngoại trừ biến Gender, từ biểu đồ trên, ta thấy được số lượng bệnh nhân theo từng biến có mức độ tương đồng khá cao. Các biểu đồ có sự biến thiên tương đối rõ ràng có thể nhận xét rằng các biến có ảnh hưởng tới số lượng bệnh nhân. Đa số các biến đều giảm xuống dưới 50 khi ở khoảng mức điểm từ 4 đến 6.

Ngoài ra, các biểu đồ cũng thể hiện xu hướng giảm khi chạm mức điểm tác động cao nhất điển hình như các biến: Air Pollution (ở mức 5 - 6 có khoảng 350 bệnh nhân, mức 7 và 8 chỉ còn khoảng dưới 25), Dust Allergy (ở mức 7 có khoảng 400 bệnh nhân, mức 8 còn khoảng dưới 10), Chest Pain, Weight Loss, Wheezing, Snoring, ,.. có thể cho thấy rằng số lượng bệnh nhân có mức độ điểm đánh giá tác động cao ở các biến này sẽ tương đối ít bị ảnh hưởng bởi hơn.

Tuy nhiên, một số biểu đồ thể hiện các biến yếu tố có mức độ ảnh hưởng ở mức điểm cao nhất với số lượng bệnh nhân khá lớn như là: Genetic Risk (khoảng trên 350 bệnh nhân), Balanced Diet (khoảng 350 bệnh nhân), Obesity (trên 350 bệnh nhân), Alcohol use (khoảng 200 bệnh nhân),... Điều này cho thấy rằng các biến này có thể đang là các yếu tố tác động mạnh đến nguyên nhân gây ra bệnh ung thư phổi…

3. Trực quan biến mục tiêu

```
tick_labels = ['Low', 'Medium', 'High']
# Vẽ histogram và
sbn.histplot(data['Level'], kde=True)
plt.title('Phân phối của biến Level')
# Thay đổi tên các cột xuất hiện trên trục x
plt.gca().set_xticks([1, 2, 3])  # Đặt vị trí của ticks
plt.gca().set_xticklabels(tick_labels)  # Đặt nhãn cho ticks
plt.show()
```



Hình 6: Phân phối biến Level

Nhận xét: Qua biểu đồ phân phối biến mục tiêu (Level), có thể thấy rằng mức độ Level càng cao thì số lượng bệnh nhân càng cao. Số lượng bệnh nhân ở level 1 (Low) có khoảng trên 300 với 30.3% , level 2 (Medium) có khoảng trên 330 với 33.2% và mức 3 (High) có khoảng trên 360 bệnh nhân với 36.5%.

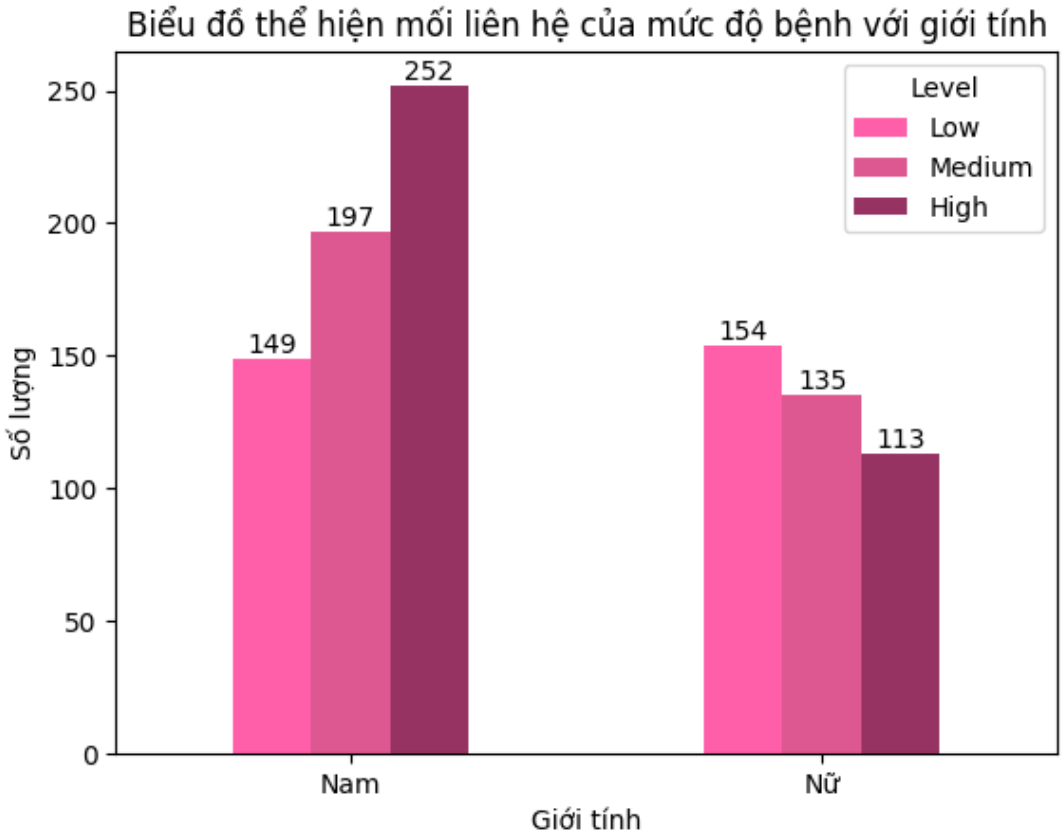
B. Trực quan mối quan hệ các biến với biến mục tiêu

1. Biểu đồ mối liên hệ của mức độ bệnh Level với giới tính Gender

Đoạn mã code để biểu diễn mối liên hệ của mức độ bệnh Level với giới tính Gender:

```
df['Gender'] = df['Gender'].replace({1: 'Nam', 2: 'Nữ'})
df['Level'] = df['Level'].replace({1: 'Low', 2: 'Medium', 3: 'High'})
a = df['Gender']
b = df['Level']
cross = pd.crosstab(a, b)
cross = cross[['Low', 'Medium', 'High']]
print(cross)
plt.figure(figsize=(14, 12))

barplot = cross.plot.bar(color=[(238/255, 106/255, 167/255), (205/255, 96/255, 144/255), (139/255, 58/255, 98/255)], rot=0, legend=False)

for p in barplot.patches:
```

```
    barplot.annotate(format(p.get_height(), '.0f'),
                     (p.get_x() + p.get_width() / 2, p.get_height()),
                     ha='center',
                     va='bottom',
                     fontsize=10,
                     color='black')
plt.xlabel('Giới tính')
plt.ylabel('Số lượng')
level_order = df['Level'].unique()
plt.legend(title='Level', labels=level_order)

plt.title("Biểu đồ thể hiện mối liên hệ của mức độ bệnh với giới tính")

plt.show()
```



Hình 7: Biểu đồ thể hiện mối liên hệ của mức độ bệnh với giới tính

Dựa vào biểu đồ, ta có thể dễ dàng nhận thấy những đặc điểm sau:

Ở mức độ bệnh "Low":

Thể hiện sự phân bố gần như đồng đều giữa giới tính "Nam" và "Nữ." Với 154 trường hợp cho giới tính "Nữ" và 149 trường hợp cho giới tính "Nam," rõ ràng ta có thể thấy sự khác biệt không đáng kể.

⇒ Mức độ bệnh "Low" không thể hiện sự chênh lệch đáng kể giữa giới tính Nam và Nữ, và tần suất của mức độ bệnh này gần như tương tự cho cả hai giới tính.

Ở mức độ bệnh "Medium":

Thể hiện một sự chênh lệch rõ rệt trong phân bố giữa giới tính "Nam" và "Nữ." Với 197 trường hợp cho giới tính "Nam" và 135 trường hợp cho giới tính "Nữ," số lượng người mắc bệnh ở giới tính "Nam" cao hơn nhiều so với giới tính "Nữ"

⇒ Sự khác biệt đáng kể này cho thấy mức độ bệnh "Medium" ảnh hưởng nhiều hơn đối với giới tính Nam, và có thể có các yếu tố gây ra mức độ bệnh này ảnh hưởng khác biệt đối với các nhóm giới tính.

Ở mức độ bệnh "High":

Thể hiện một sự chênh lệch rõ rệt nhất trong phân bố giữa giới tính "Nam" và "Nữ." Với 252 trường hợp cho giới tính "Nam" và 113 trường hợp cho giới tính "Nữ," tần suất cho giới tính "Nam" cao hơn đáng kể so với giới tính "Nữ."

⇒ Sự khác biệt đáng kể này cho thấy mức độ bệnh "High" ảnh hưởng nhiều hơn đối với giới tính Nam, và có thể có các yếu tố gây ra mức độ bệnh này ảnh hưởng khác biệt đối với các nhóm giới tính.

Ngoài ra nhóm cũng xin đưa ra nhận xét chung dựa vào từng nhóm giới tính như sau: Giới tính "Nam" thể hiện một mức độ bệnh trung bình cao hơn so với giới tính "Nữ," với sự ảnh hưởng lớn hơn tại mức độ bệnh "Medium" và "High". Giới tính "Nữ" thể hiện một mức độ bệnh trung bình thấp hơn so với giới tính "Nam," với sự ảnh hưởng lớn nhất tại mức độ bệnh "Low."

⇒ Mức độ bệnh có sự chênh lệch trong phân phối giữa giới tính "Nam" và "Nữ," với mức độ bệnh "Medium" và "High" ảnh hưởng nhiều hơn đối với giới tính "Nam," trong khi mức độ bệnh "Low" ảnh hưởng nhiều hơn đối với giới tính "Nữ."

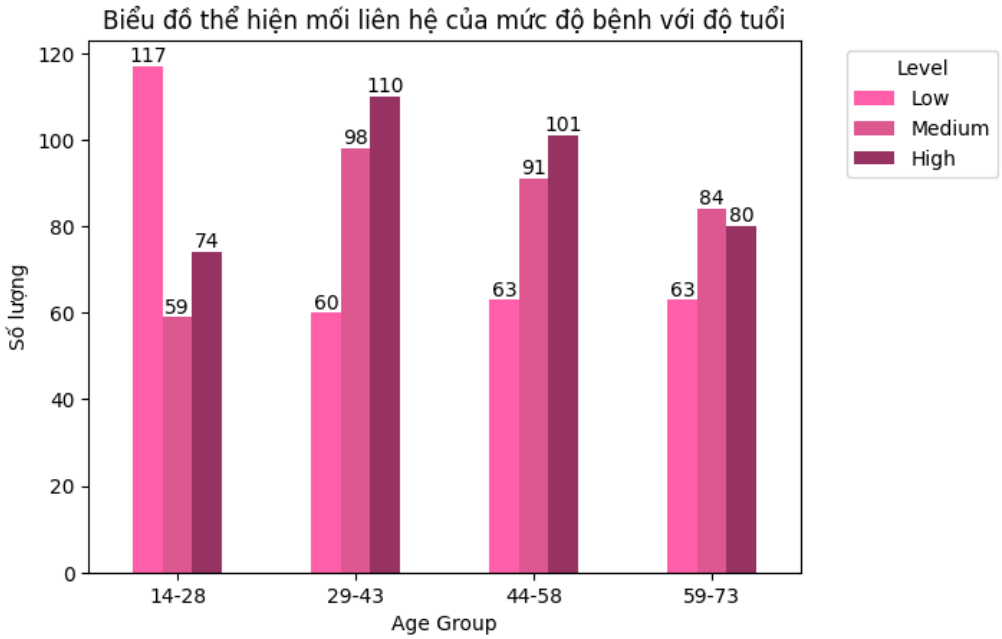
2. Biểu đồ thể hiện mối liên hệ của mức độ bệnh với độ tuổi

```
df['Age Group'] = df['Age Group'].replace({1: '14-28', 2: '29-43', 3: '44-58', 4: '59-73'})
```

```
c = df['Age Group']
d = df['Level']

cross = pd.crosstab(c, d)
new_order = ['Low', 'Medium', 'High']
cross = cross[new_order]
print(cross)
plt.figure(figsize=(14, 20))

barplot = cross.plot.bar(color=[(238/255, 106/255, 167/255), (205/255,
96/255, 144/255), (139/255, 58/255, 98/255)], rot=0)

for p in barplot.patches:
    barplot.annotate(format(p.get_height(), '.0f'),
                     (p.get_x() + p.get_width() / 2, p.get_height()),
                     ha='center',
                     va='bottom',
                     fontsize=10,
                     color='black')

level_order = df['Level'].unique()
plt.xlabel('Age Group')
plt.ylabel('Số lượng')
plt.legend(title='Level', labels=level_order , bbox_to_anchor=(1.05, 1),
loc='upper left')
plt.title("Biểu đồ thể hiện mối liên hệ của mức độ bệnh với độ tuổi")
plt.show()
```

Level / Age Group	Low	Medium	High
14-28	117	59	74
29-43	60	98	110
44-58	63	91	101
59-73	63	84	80

Bảng 2: Mối liên hệ của mức độ bệnh với độ tuổi



Hình 8: Biểu đồ liên hệ giữa mức độ bệnh Level với độ tuổi Age Group

Kết luận 1: Dựa vào kết quả trên ta có thể thấy có 3 mức độ bệnh là Low, Medium và High và giữa các mức độ bệnh có sự biến động theo các nhóm tuổi.

- **Ở nhóm tuổi "14-28":**

Với 117 trường hợp, mức độ bệnh "Low" có số lượng người mắc bệnh gặp phải cao nhất ở nhóm tuổi này. Mức độ bệnh "Medium" và "High" có giá trị thấp hơn so với Low, lần lượt là 59 và 74 trường hợp.

⇒ Ở nhóm tuổi "14-28", người bệnh có xu hướng bị bệnh nhẹ nhiều hơn 2 mức độ bệnh còn lại.

- **Ở nhóm tuổi "29-43":**

Với 110 và 98 trường hợp mắc bệnh lần lượt cho mức độ bệnh "High" và "Medium" thì đây là nhóm tuổi có số lượng người mắc bệnh gặp phải cao nhất không chỉ ở nhóm tuổi này mà còn cao nhất so với các nhóm tuổi khác. Mức độ bệnh "Low" có 60 người mắc bệnh gặp phải, ngược lại với 2 mức độ trên thì đây mức độ bệnh có số lượng người mắc thấp nhất so với các nhóm tuổi còn lại.

⇒ Ở nhóm tuổi "29-43", có sự chênh lệch rõ ràng giữa mức độ bệnh "Low" và 2 mức độ bệnh còn lại. Người mắc bệnh ở nhóm tuổi này có xu hướng bị bệnh nặng hoặc trung bình, khi so sánh với các nhóm tuổi khác thì ta có thể thấy số lượng của 2 mức độ này cao hơn hẳn.

- **Ở nhóm tuổi "44-58":**

Tương tự với nhóm tuổi "29-43" thì ở nhóm tuổi này số lượng người mắc bệnh cao cho 2 mức độ bệnh là "High" và "Medium" với số liệu lần lượt là 101 và 91. Mức độ bệnh "Low" cũng là mức độ có số lượng người mắc phải thấp nhất trong nhóm tuổi này.

⇒ Nhìn chung, nhóm tuổi 44-58 có xu hướng có số lượng người mắc bệnh thấp hơn so với nhóm tuổi 29-43, nhưng số lượng người mắc bệnh ở mức độ "High" và "Medium" vẫn còn quá cao so với các nhóm tuổi còn lại.

- **Ở nhóm tuổi "59-73":**

Ở nhóm tuổi này sự chênh lệch giữa các mức độ bệnh giảm đi đáng kể, các trường hợp mắc bệnh ở từng mức độ khá tương đồng nhau. Cụ thể: High: 80 trường hợp; Medium: 84 trường hợp và Low: 63 trường hợp.

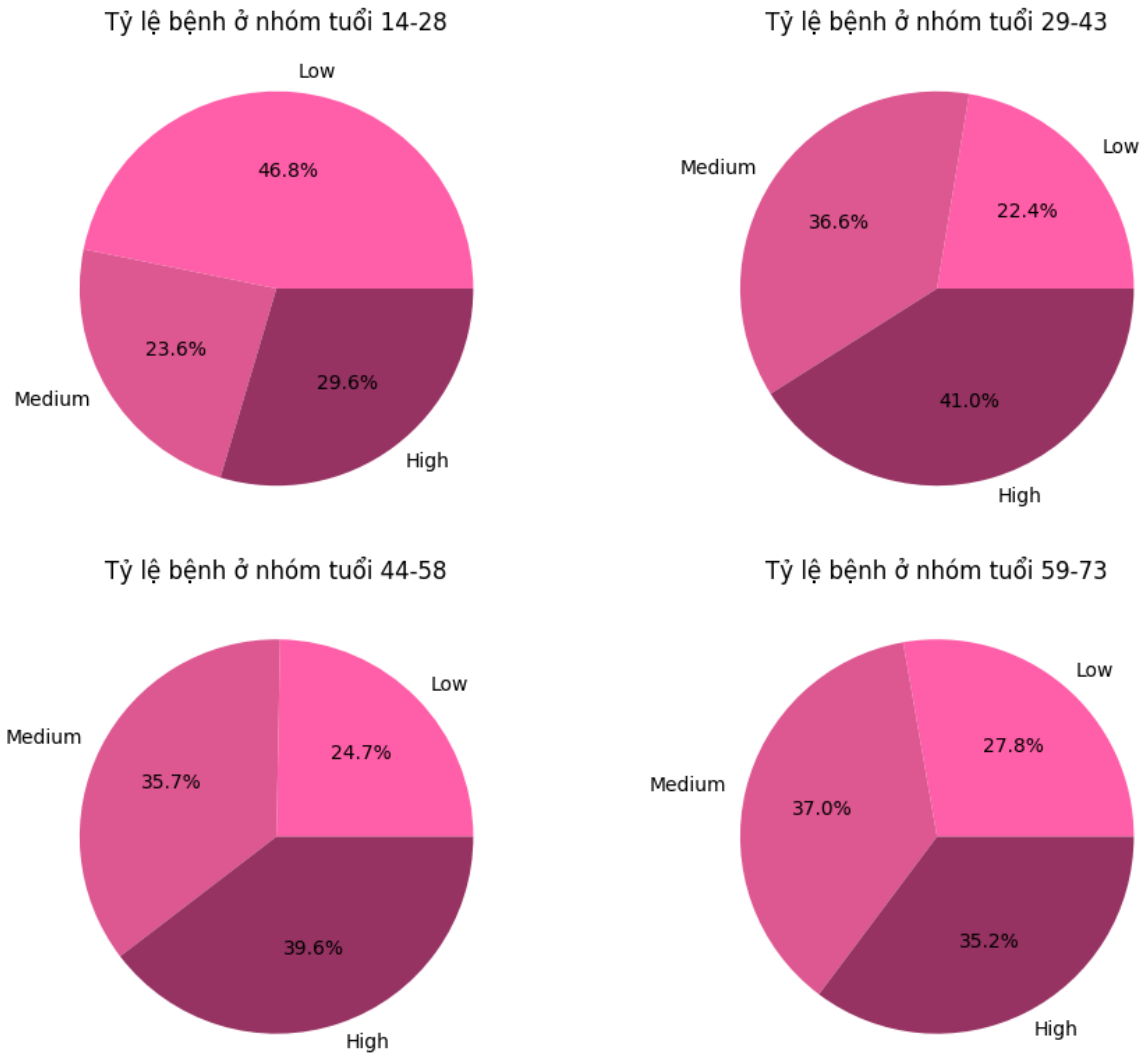
⇒ Trong nhóm tuổi "59-73", có vẻ có sự cân bằng hơn giữa các mức độ bệnh, với số lượng người mắc giảm đáng kể ở mức độ "High".

Nhận xét mở rộng: Ta có thể thấy tổng số ca bệnh được khảo sát trong từng nhóm tuổi là khác nhau, vì thế để có thể so sánh tốt nhất giữa mức độ bệnh của từng nhóm tuổi ta cần tính phần trăm số lượng từng mức độ bệnh ở mỗi nhóm tuổi và nhóm xin trình bày đoạn code sau:

```
cross['Total'] = cross.sum(axis=1)

colors = [(238/255, 106/255, 167/255), (205/255, 96/255, 144/255),
(139/255, 58/255, 98/255)]

fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 8))

for i, ax in enumerate(axes.flatten()):
   age_group = cross.index[i]

   low_percentage = cross.loc[age_group, 'Low'] / cross.loc[age_group,
'Total'] * 100
   medium_percentage = cross.loc[age_group, 'Medium'] /
cross.loc[age_group, 'Total'] * 100
   high_percentage = cross.loc[age_group, 'High'] /
cross.loc[age_group, 'Total'] * 100
```

```
   ax.pie([low_percentage, medium_percentage, high_percentage],
labels=['Low', 'Medium', 'High'], autopct='%1.1f%%', colors=colors)
   ax.set_title(f'Tỷ lệ bệnh ở nhóm tuổi {age_group}')

plt.tight_layout()
plt.show()
```

Kết quả thu được:



Hình 9: Các biểu độ bệnh ở từng nhóm tuổi

Kết luận 2:

Mức độ bệnh "Low": Nhóm tuổi "14-28" có tỷ lệ mắc bệnh cao nhất với 46,8%. Đây là nhóm tuổi đang trong giai đoạn phát triển, có sức đề kháng cao nên tỷ lệ mắc bệnh thấp.

Mức độ bệnh "Medium": Ba nhóm tuổi "29-43", "44-58" và "59-73" có tỷ lệ mắc bệnh tương đương nhau, lần lượt là 36,6%, 35,7% và 37%. Đây là những nhóm tuổi có nguy cơ mắc bệnh cao hơn nhóm tuổi "14-28" do sức đề kháng bắt đầu suy giảm.

Mức độ bệnh "High": Nhóm tuổi "29-43" có tỷ lệ mắc bệnh cao nhất với 41%. Đây là nhóm tuổi có nguy cơ mắc bệnh cao nhất do sức đề kháng suy giảm và có nhiều yếu tố nguy cơ khác như: lối sống, chế độ ăn uống, môi trường sống,...

3. Mối liên hệ giữa các biến với mức độ bệnh

Đoạn mã code để biểu diễn phần trăm của các biến với mức độ bệnh:

```
columns2 = list(data.columns)
columns2.pop(0)
columns2.pop(0)
num_plots = len(columns2)
rows = int(num_plots / 3) + (num_plots % 3 > 0)
cols = 3 if num_plots > 3 else num_plots

fig, axes = plt.subplots(rows, cols, figsize=(20, 28))
axes = axes.reshape(-1)

for i, ax in enumerate(axes):
    if i < num_plots:
        df                                                              =
data.groupby([columns2[i],'Level']).size().unstack(0).apply(lambda    x:
np.round(x*100/x.sum(), 2))
        df.T.plot(kind= 'bar',ax=ax,stacked= True)
        ax.set_title(f'Phần trăm của từng mức độ bệnh Level với biến
{columns2[i]}')
        ax.legend(title='Level', labels=['Low', 'Medium','High'])
axes[-3].remove()
axes[-2].remove()
axes[-1].remove()
plt.tight_layout()
plt.show()
```

Hình 10: Phần trăm của các biến đặc trưng với mức độ bệnh Level

Dựa vào biểu đồ tổng hợp trên, ta thấy được tỷ lệ mức độ bệnh với các biến theo từng điểm có nhiều tương đồng. Phần màu xanh dương thể hiện cho mức Low chiếm phần trăm lớn ở các mức điểm thấp từ 1-3 trong tất cả các biến. Ngược lại, phần High màu xanh lá lại có xu hướng chiếm nhiều ở mức điểm cao 7-9 ở hầu hết biểu đồ. Phần màu cam cho mức Medium hầu như phân bố rộng trên toàn bộ thang điểm.

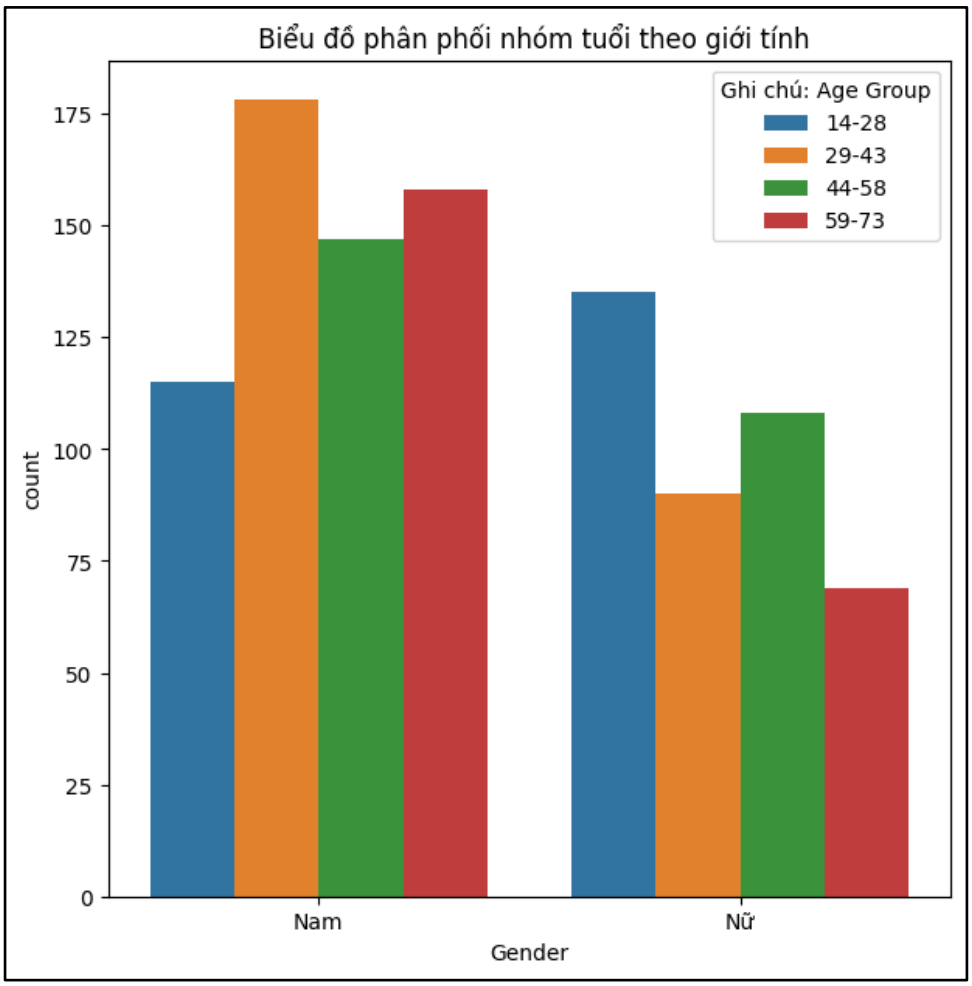
Tuy nhiên có thể nhìn thấy những điểm đặc biệt, trong biến Dust Allergy mức điểm cao nhất có 100% là Low; ở Shortness of Breath mức bệnh High chiếm tỷ lệ lớn ở mức điểm giữa 4 và 5, ít hơn ở hai đầu; biến Wheezing ở mức điểm thấp có mức bệnh High lại chiếm nhiều thứ 2, thay vì mức Low như các biểu đồ khác.

C. Trực quan tương quan của các biến

1. Biểu đồ mối liên hệ nhóm tuổi và giới tính

Để biểu diễn mối quan hệ giữa nhóm tuổi và giới tính, ta sử dụng biểu đồ barplot sau đó thực hiện đổi nhãn để biểu đồ trực quan hơn thông qua đoạn code sau

```
df_gb = data.groupby(["Age
Group" ,"Gender"])["Level"].count().reset_index(name ="count")
fig, ax = plt.subplots(figsize = (7,7))
sbn.barplot(data =df_gb, x ="Gender", y ="count", hue = "Age Group")
ax.set(title =" Biểu đồ phân phối nhóm tuổi theo giới tính",
       xlabel = "Gender",
       ylabel = "count")
# Lấy danh sách handles và labels từ chú thích hiện tại
handles, labels = ax.get_legend_handles_labels()

# Thay đổi nhãn của chú thích
new_labels = ['14-28', '29-43', '44-58', '59-73']
plt.legend(handles, new_labels, title='Ghi chú: Age Group')
# Thay đổi tên các cột xuất hiện trên trục x
tick_labels = ['Nam', 'Nữ']
plt.xticks(ticks=[0, 1], labels=tick_labels)
plt.show
```



Hình 11: Biểu đồ phân phối nhóm tuổi theo giới tính

Biểu đồ có hai trục: trục tung thể hiện số lượng người và trục hoành thể hiện nhóm tuổi theo giới tính. Theo biểu đồ phân phối có thể nhận xét rằng số lượng bệnh nhân thuộc giới tính nam nhiều hơn nữ. Trong đó, nam giới có nhóm tuổi mắc bệnh cao nhất là 29 - 43 tuổi (khoảng hơn 175 người). Nữ giới có độ tuổi mắc bệnh cao nhất là 14 - 28 tuổi (khoảng hơn 130 người) nhưng đây lại là độ tuổi mà nam giới có số lượng bệnh nhân ít nhất (khoảng 115 người).

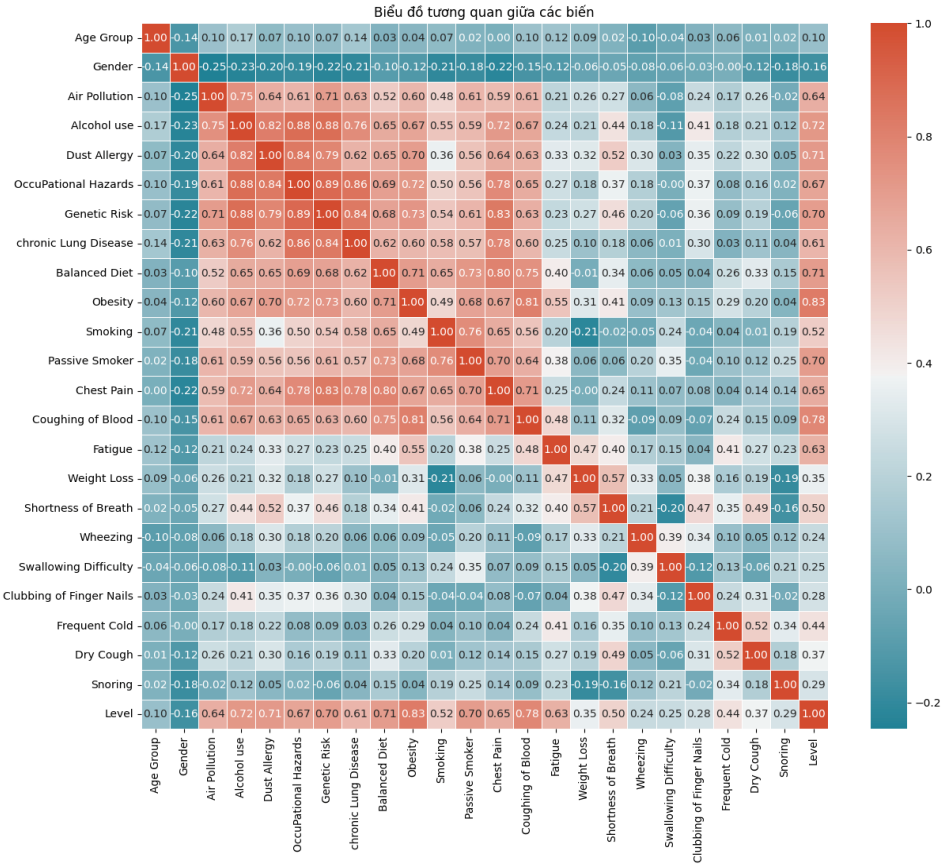
2. Biểu đồ tương quan của các biến

Nhóm sử dụng thư viện seaborn và matplotlib để tạo và hiển thị một biểu đồ thể hiện sự tương quan giữa các biến dưới dạng heatmap.

```
# Ma trận tương quan
correlation_matrix = data.corr()
print(correlation_matrix)

cmap = sns.diverging_palette(220, 20, s=75, l=50, as_cmap=True)
```

```
plt.figure(figsize=(14, 12))
sns.heatmap(correlation_matrix, annot=True, cmap=cmap, fmt=".2f",
linewidths=0.5)
plt.title('Biểu đồ tương quan giữa các biến')
plt.show()
```

Kết quả thu được:



Hình 12: Tương quan các biến Feature

Từ biểu đồ tương quan trên, nhóm xin đưa ra các nhận xét như sau:

- Nhận xét về tương quan của các biến:

+ Biến Air Pollution, Alcohol use và Dust Allergy là các biến có mức tương quan cao, điều này phản ánh mối liên hệ giữa ô nhiễm không khí, sử dụng rượu, và dị ứng bụi trong việc gây ra các vấn đề về sức khỏe cụ thể là ảnh hưởng đến đường hô hấp.
+ Smoking và Passive Smoker có mức tương quan đáng kể, làm nổi bật tác hại của việc hút thuốc và tiếp xúc với khói thuốc từ người khác đối với sức khỏe của hệ thống hô hấp.

+ Obesity có mức tương quan cao với nhiều yếu tố khác, có thể là một yếu tố nguy cơ cho nhiều vấn đề sức khỏe, bao gồm cả bệnh ung thư phổi. Sự liên kết này có thể phản ánh vai trò quan trọng của việc duy trì cân nặng lành mạnh trong việc bảo vệ sức khỏe của hệ thống hô hấp.

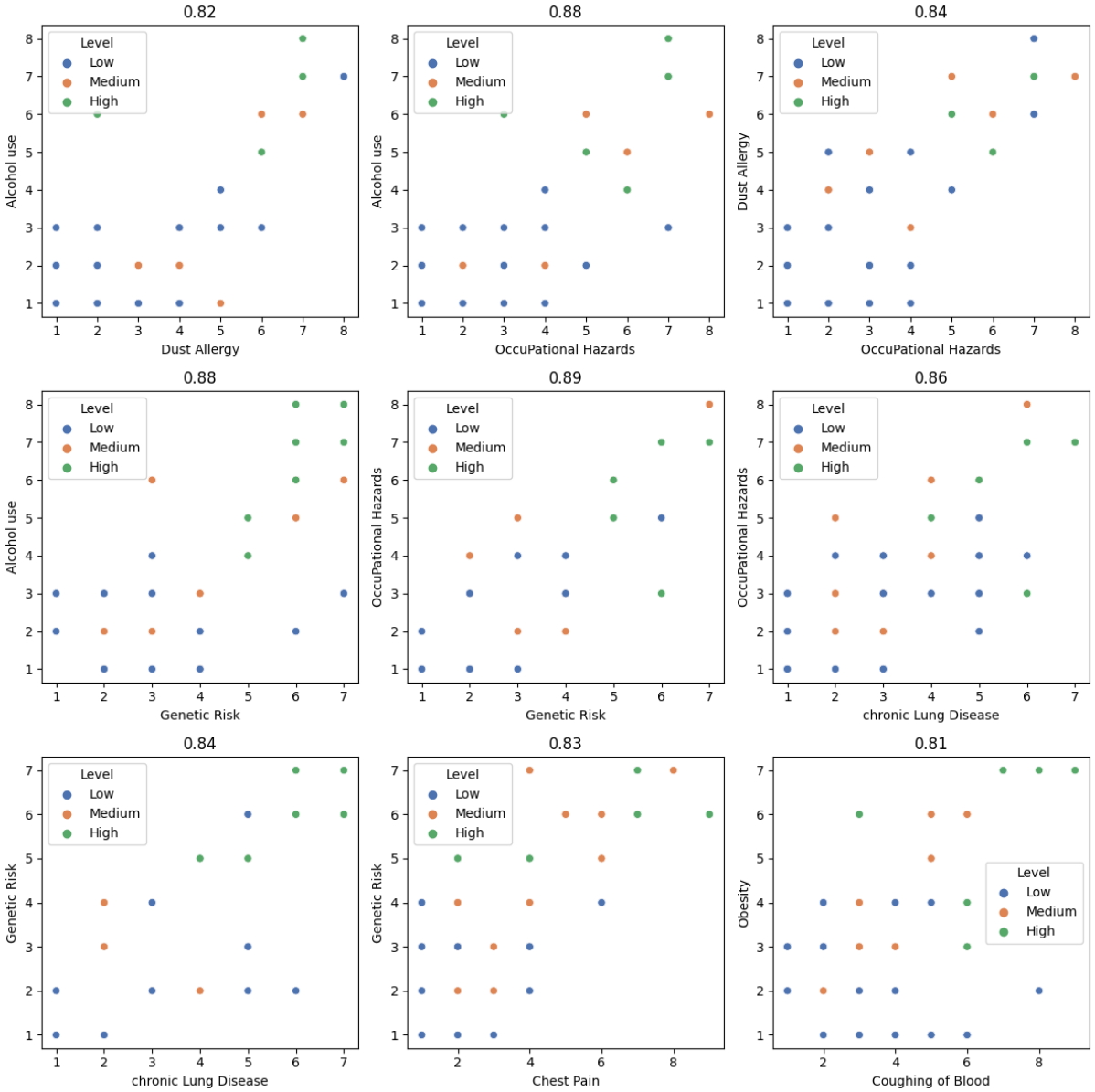
- Nhận xét về tương quan của các biến với biến mục tiêu:
  + Tất cả các biến độc lập trong ma trận tương quan đều có mức độ tương quan cao với biến mục tiêu "Level", với mức tương quan từ 0,6 đến 0,8. Điều này cho thấy rằng các biến này có thể có ảnh hưởng đáng kể đến mức độ nghiêm trọng của bệnh ung thư phổi.
  + Trong số các biến này, biến "Obesity" có mức độ tương quan cao nhất, với hệ số tương quan là 0,82. Điều này cho thấy rằng tình trạng thừa cân có thể là một yếu tố quan trọng đối với sự nghiêm trọng của bệnh ung thư phổi.
  + Các biến khác có mức độ tương quan cao với biến mục tiêu cũng bao gồm:
    - Mức độ ô nhiễm không khí (Air Pollution)
    - Sử dụng rượu (Alcohol use)
    - Dị ứng bụi (Dust Allerg)
    - Rủi ro nghề nghiệp (OccuPational Hazards)
    - Yếu tố di truyền (Genetic Risk)
    - Bệnh phổi mạn tính (chronic Lung Disease)

3. Mối liên hệ giữa các biến tương quan trên 0,8

Đoạn mã code để biểu diễn mối liên hệ của các biến có tương quan trên 0,8:

```
correlation_matrix = data.corr()
lst_corr=[]
for col in correlation_matrix:
  row=0
  for j in correlation_matrix[col]:
    if j==1 :
      break
    if j >0.8:
      lst= [col,correlation_matrix.index[row],j]
      lst_corr.append(lst)
    row +=1
lst_corr.pop()
```

```
num_plots = len(lst_corr)
rows = int(num_plots / 3) + (num_plots % 3 > 0)
cols = 3 if num_plots > 3 else num_plots

fig, axes = plt.subplots(rows, cols, figsize=(11, 11))

for i, ax in enumerate(axes.flatten()):
  if i < len(lst_corr):
    sns.scatterplot(data=data,  x=lst_corr[i][0],y=lst_corr[i][1],ax=ax,
hue='Level', palette="deep")
    ax.set_title("%.2f" % lst_corr[i][2])
plt.tight_layout()
plt.show()
```



Hình 13: Mối liên hệ của các biến có tương quan trên 0,8

Từ biểu đồ trên, thấy được sự tương quan tương đối rõ ràng của 9 cặp biến đặc trưng có hệ số tương quan trên 0,8 gồm : Alcohol use với Dust Allergy; Alcohol use với OccuPational Hazards; Dust Allergy với OccuPational Hazards; Alcohol use với Genetic Risk; OccuPational Hazards với Genetic Risk;OccuPational Hazards với chronic Lung Disease; Genetic Risk với chronic Lung Disease; Genetic Risk với Chest Pain; Obesity với Coughing of Blood. Quan sát để hỗ trợ giảm chiều dữ liệu trước khi xây dựng mô hình học máy khi loại bỏ 1 trong 2 biến có tương quan mạnh.

Bên cạnh đó, nhìn thấy sự phân bổ của 3 mức độ bệnh Low, Medium, High theo màu sắc, thay đổi tương ứng từ thấp đến cao so với mức độ của các biến. Tuy nhiên sự

phân hóa không quá rõ ràng, vẫn có nhiều sự chồng lắp giữa 3 mức, đặc biệt là mức Medium thường phân bổ rộng trên thang đánh giá.

V. Phân lớp

A. Tạo bộ dữ liệu train, test

Các bộ dữ liệu huấn luyện và kiểm thử là điều cần thiết để phát triển một mô hình học máy hiệu quả và đáng tin cậy. Bộ dữ liệu train cung cấp cho thuật toán học máy các thông tin cần thiết để "học" cách thực hiện nhiệm vụ. Để đánh giá hiệu suất và xác minh tính khả thi của mô hình, chúng ta cần kiểm tra mô hình trên tập dữ liệu test là bộ dữ liệu mà nó chưa từng nhìn thấy trước đây.

Để bắt đầu, chúng ta cần tạo ra hai biến: biến x chứa các thuộc tính đầu vào và biến y chứa các giá trị mục tiêu. Trong ví dụ này, các thuộc tính đầu vào là tất cả các cột ngoại trừ 'Level', và giá trị mục tiêu là 'Level'.

```
X = data.drop('Level', axis =1)
y = data['Level']
```

Chúng ta đã sử dụng thư viện sklearn và hàm train_test_split để thực hiện việc chia tách này.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Sử dụng phương thức .shape để in ra kích thước của tập dữ liệu huấn luyện, bao gồm số lượng mẫu và số lượng đặc trưng trong mỗi mẫu. Dòng thứ hai thực hiện tương tự cho tập dữ liệu kiểm thử.

```
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

Ta có được kết quả như sau:

(800, 23)	(800,)
(200, 23)	(200,)

Bảng 3: Bộ dữ liệu train, test

⇒ Tập X_train có 800 dòng dữ liệu đào tạo, 23 cột ứng với 23 thuộc tính

Tập y_train có 800 dòng dữ liệu đào tạo, 1 cột ứng với 1 biến mục tiêu

Tập X_test có 200 dòng dữ liệu kiểm thử, 23 cột ứng với 23 thuộc tính

Tập y_test có 200 dòng dữ liệu kiểm thử, 1 cột ứng với 1 biến mục tiêu

B. Xây dựng mô hình

Với các tập X_train, X_test, Y_train, Y_test đã chia, ta phân lớp bằng các phương pháp và đánh giá mỗi phương pháp bằng các giá trị Accuracy, precision, recall, f1-score.

1. Phân lớp bằng phương pháp KNN classification

Chúng ta đã sử dụng dạng neighbors thư viện sklearn và sử dụng mô hình KNeighborsClassifier để thực hiện phân lớp theo phương pháp KNN.Đầu tiên, chúng ta tạo một đối tượng KNN mới. Sau đó, chúng ta sử dụng phương thức fit() để huấn luyện mô hình trên tập train set. Cuối cùng, chúng ta sử dụng phương thức predict() để dự đoán nhãn của các mẫu trong tập dữ liệu test.

Đoạn code Python tạo ra bộ phân lớp KNN:

```
## Xây dựng mô hình kNN Classification
from sklearn.neighbors       import KNeighborsClassifier
k    = int(pow(X_train.shape[0], 1/2) / 2)
knn = KNeighborsClassifier(n_neighbors = k)
knn.fit(X_train, y_train)  # huấn luyện để tạo  mô hình
y_pred = knn.predict(X_test)
```

Với bài toán phân lớp theo phương pháp KNN, ta chọn số k láng giềng gần nhất tùy ý, tuy nhiên cách phổ biến để chọn số k là k = căn bậc hai của số lượng mẫu trong dữ liệu huấn luyện chia cho 2 (Làm tròn xuống số nguyên gần nhất). Tức là với mỗi điểm trong tệp X_test, ta chỉ xét k điểm trong tập X_train gần nhất và lấy label của điểm đó để dự đoán cho điểm test này.

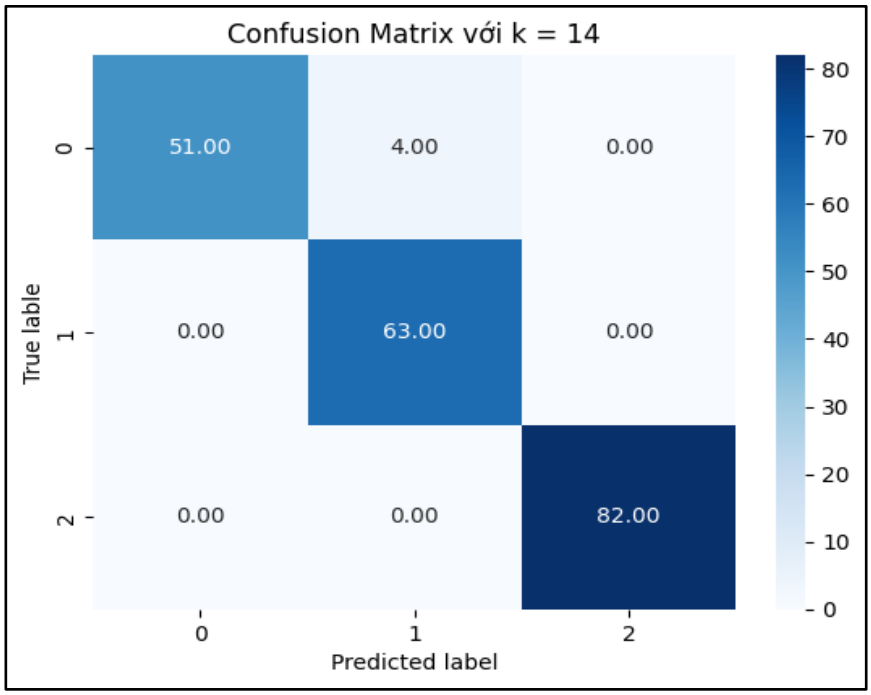
Sử dụng các phương thức để tính các điểm chỉ số (KNN) và ma trận nhầm lẫn được thực hiện bằng confusion_matrix được lấy từ thư viện sklearn :

```
#Ma trận nhầm lẫn
from sklearn.metrics import confusion_matrix
print(f'Confusion Matrix:\n {confusion_matrix(y_test, y_pred)}')
# Đánh giá độ chính xác của mô hình
print(f'Accuracy ={accuracy_score(y_test, y_pred)*100:.2f}%')
print(f'Precision ={precision_score(y_test,
y_pred,average="micro")*100:.2f}%')
print(f'recall ={recall_score(y_test,
y_pred,average="micro")*100:.2f}%')
```

```
print(f'f1-score ={f1_score(y_test, y_pred,average="micro")*100:.2f}%')
```

Kết quả phân lớp của mô hình phân lớp này là:

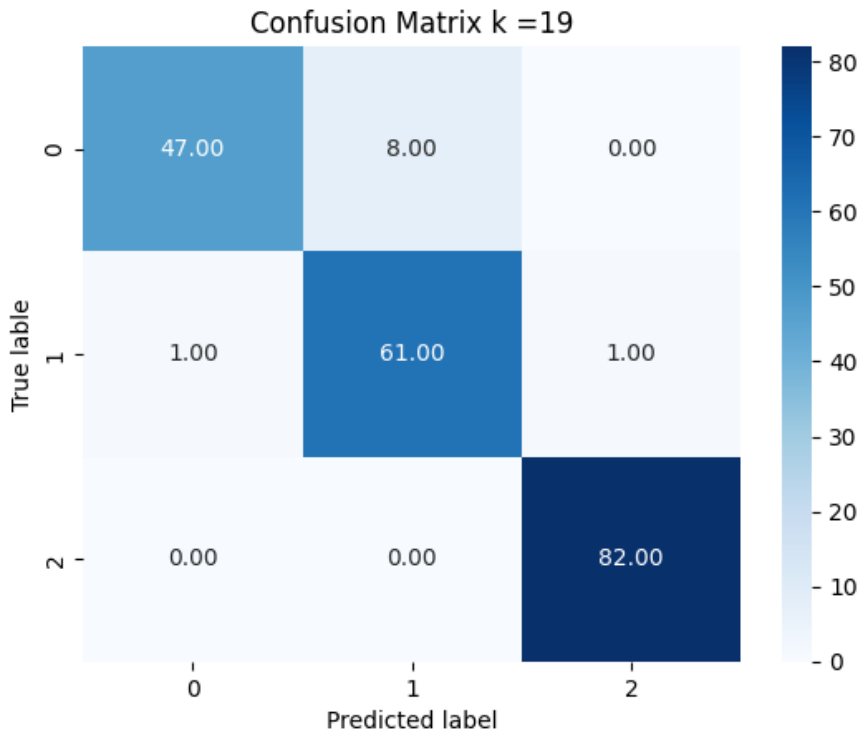
Với k = `int(pow(X_train.shape[0], 1/2) / 2)`: =14 : tại cùng 1 bộ dữ liệu train, test ta thu được ma trận nhầm lẫn như sau :



Hình 14: Confusion matrix với k = 14

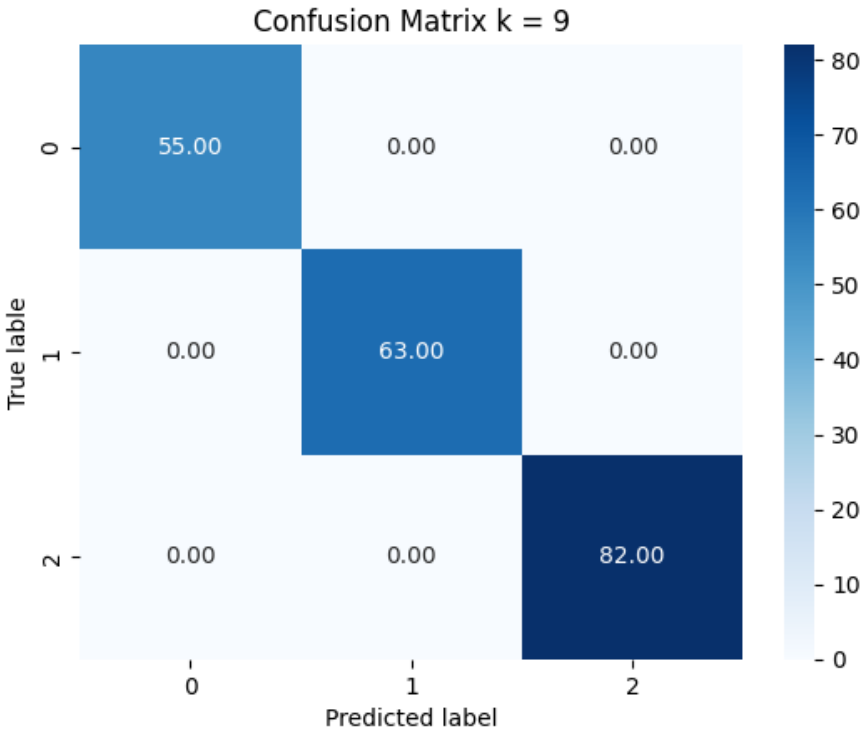
Chọn thêm 1 vài k tùy ý:

Với k = `int(pow(X_train.shape[0], 1/2) / 2) + 5` = 19: tại cùng 1 bộ dữ liệu train, test ta thu được kết quả:



Hình 15: Confusion matrix với k = 19

Với k = `int(pow(X_train.shape[0], 1/2) / 2) - 5` = 9: tại cùng 1 bộ dữ liệu train, test ta thu được chỉ số :



Hình 16: Confusion matrix với k = 9

Sử dụng các phương thức để tính các điểm chỉ số thu được kết quả:

Phương pháp	k	Accuracy	Prediction	Recall	f1-score
KNN	14	98%	98%	98%	98%
	19	95%	95%	95%	95%
	9	100%	100%	100%	100%

Bảng 4: Kết quả phân lớp với phương pháp KNN

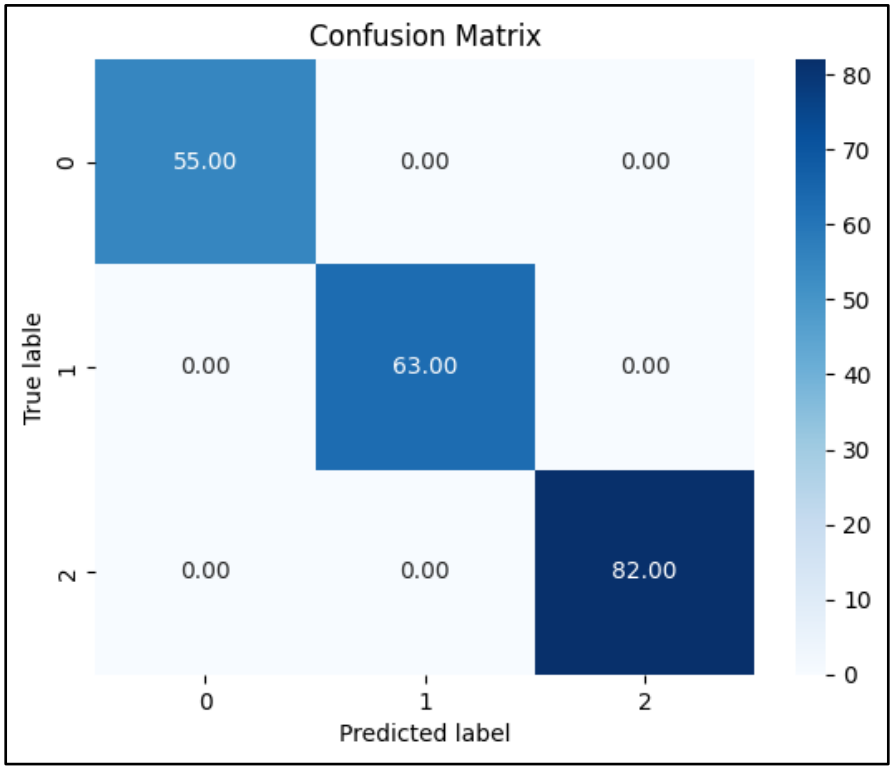
2. Phân lớp bằng phương pháp Decision Tree

Chúng ta đã sử dụng loại tree từ thư viện sklearn và mô hình Decision Tree Classifier để thực hiện phân lớp theo phương pháp KNN.Khởi tạo mô hình phân lớp bằng phương pháp cây quyết định (Decision tree) với thang đo entropy. Sau đó, chúng ta sử dụng phương thức fit() để huấn luyện mô hình trên tập train set. Cuối cùng, chúng ta sử dụng phương thức predict() để dự đoán nhãn của các mẫu trong tập dữ liệu test.

```
## Xây dựng mô hình Decision Tree
clf   = DecisionTreeClassifier(criterion='entropy')
model = clf.fit(X, y) # huấn luyện để tạo  mô hình
# Dự đoán nhãn cho tập kiểm tra
y_pred = model.predict(X_test)
```

Hình: Code Python tạo ra bộ phân lớp Decision Tree

Sử dụng các phương thức để tính các điểm chỉ số và ma trận nhầm lẫn được thực hiện bằng confusion_matrix được lấy từ thư viện sklearn :



Hình 17: Confusion matrix của Decision Tree

Sử dụng các phương thức để tính các điểm chỉ số thu được kết quả:

Phương pháp	Accuracy	Prediction	Recall	f1-score
Decision Tree	100%	100%	100%	100%

Bảng 5: Kết quả phân lớp với phương pháp Decision Tree

3. Phân lớp bằng phương pháp Support Vector Machine

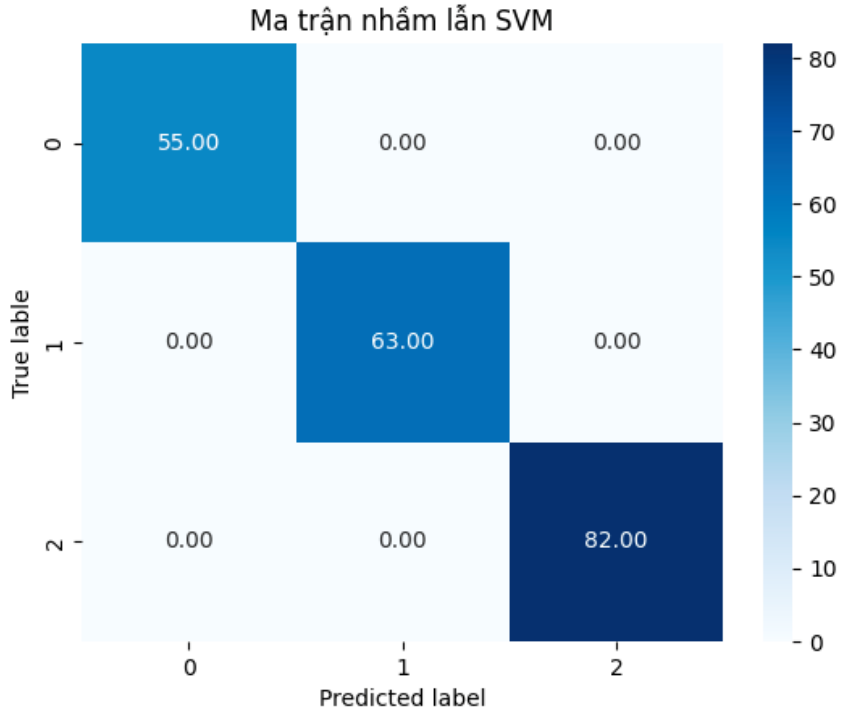
Để xây dựng mô hình phân lớp bằng phương pháp Support Vector Machine, nhóm sử dụng thư viện sklearn và mô hình SVC. Đầu tiên, chúng ta tạo một đối tượng SVM mới. Sau đó, chúng ta sử dụng phương thức fit() để huấn luyện mô hình trên tập train set. Cuối cùng, chúng ta sử dụng phương thức predict() để dự đoán nhãn của các mẫu trong tập dữ liệu test.

```
from sklearn.svm import SVC
svm=SVC()
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)
```

Ma trận nhầm lẫn:

```
# Vẽ ma trận nhầm lẫn
sbn.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='.2f',
cmap='Blues')
```

```
plt.xlabel('Predicted label')
plt.ylabel('True lable')
plt.title('Ma trận nhầm lẫn SVM')
plt.show()
```



Hình 18: Confusion matrix của Support Vector Machine

Nhóm sử dụng các phương thức từ thư viện sklearn để tính toán các chỉ số đánh giá như độ chính xác (Accuracy), độ chính xác trong dự đoán đúng (Precision), độ nhạy (Recall), và F1-score.

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
print(f'Accuracy = {accuracy_score(y_test, y_pred)*100:.2f}%')
print(f'Precision = {(precision_score(y_test,y_pred,average="micro"))
*100:.2f}%')
print(f'recall = {recall_score(y_test,y_pred,average="micro")
*100:.2f}%')
print(f'f1-score = {f1_score(y_test,y_pred,average="micro")*100:.2f}%')
```

Ta thu được kết quả như sau:

Phương pháp	Accuracy	Precision	recall	f1-score

SVM	100.00%	100.00%	100.00%	100.00%

Bảng 6: Kết quả phân lớp với phương pháp Support Vector Machine

Nhận xét: kết quả thu được với độ chính xác là 100% ở tất cả các chỉ số, cho thấy mô hình hoạt động rất tốt trên tập dữ liệu kiểm thử.

4. Phân lớp bằng phương pháp Naive Bayes

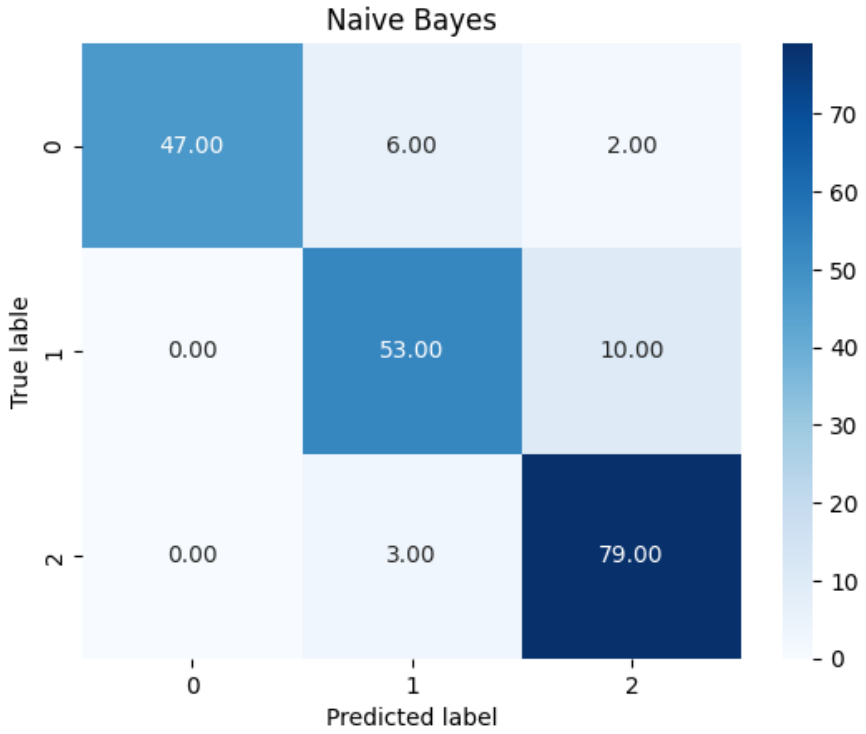
Để xây dựng mô hình phân lớp bằng phương pháp Support Vector Machine, nhóm sử dụng loại naive_bayes từ thư viện sklearn và mô hình GaussianNB. Đầu tiên, chúng ta tạo một đối tượng SVM mới. Sau đó, chúng ta sử dụng phương thức fit() để huấn luyện mô hình trên tập train set. Cuối cùng, chúng ta sử dụng phương thức predict() để dự đoán nhãn của các mẫu trong tập dữ liệu test.

```
from sklearn.naive_bayes import GaussianNB
nbayes=GaussianNB()
nbayes.fit(X_train, y_train)
y_pred = nbayes.predict(X_test)
```

Sử dụng các phương thức để tính các điểm chỉ số

```
#Ma trận nhầm lẫn
from sklearn.metrics import confusion_matrix
print(f'Confusion Matrix:\n {confusion_matrix(y_test, y_pred)}')

# Đánh giá độ chính xác của mô hình
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
print(f'Accuracy ={accuracy_score(y_test, y_pred)*100:.2f}%')
print(f'Precision                                ={(precision_score(y_test,
y_pred,average="micro"))*100:.2f}%')
print(f'recall ={recall_score(y_test, y_pred,average="micro")*100:.2f}%')
print(f'f1-score ={f1_score(y_test, y_pred,average="micro")*100:.2f}%')
```

Kết quả phân lớp của mô hình bằng phương pháp Naive Bayes



Hình 19: Confusion matrix của Naive Bayes

Phương pháp	Accuracy	Precision	Recall	F1-score
Naive Bayes	89.5%	89.5%	89.5%	89.5%

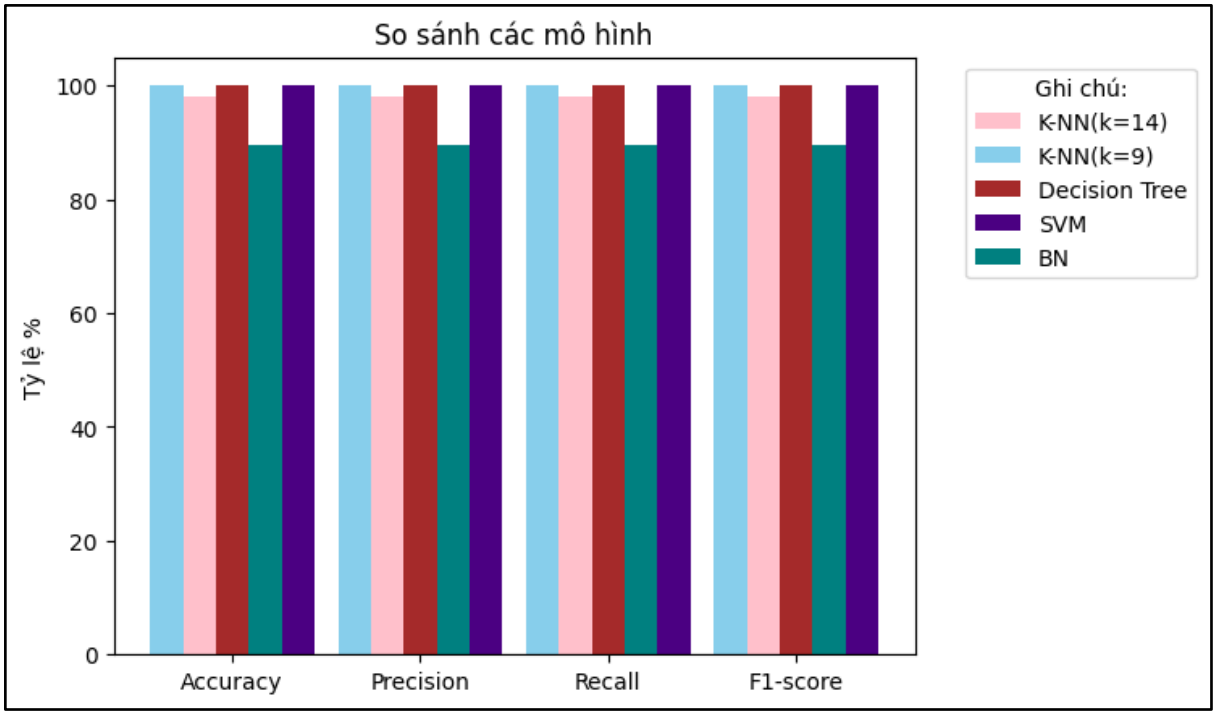
Bảng 7: Kết quả phân lớp với phương pháp Naive Bayes

C. Đánh giá mô hình phân lớp

Bảng tổng hợp các chỉ số:

Phương pháp	Accuracy	Prediction	Recall	f1-score
KNN k=14	98%	98%	98%	98%
KNN k=19	95%	95%	95%	95%
KNN k=9	100%	100%	100%	100%
Decision Tree	100%	100%	100%	100%
SVM	100%	100%	100%	100%
Naive Bayes	89.5%	89.5%	89.5%	89.5%

Bảng 8: Tổng hợp kết quả phân với các phương pháp



Hình 20: Biểu đồ so sánh kết quả các mô hình

Đánh giá: Các mô hình đều đưa ra các kết quả cực kì tốt, các số liệu đánh giá đều rất cao (100%) . Đặc biệt là ba mô hình: Decision Tree, K-NN với k = 9, Support Vector Machine (SVM) có độ chính xác tận 100%, các giá trị được dữ đoán hoàn toàn chính xác với bộ dữ liệu hiện tại. Bên cạnh đó, các mô hình phân lớp còn lại cũng có các kết quả rất tốt: K-NN với k = 14 có độ chính xác là 98% và Naive Bayes có độ chính xác là 89.5%

Kết luận: Mô hình Decision Tree, mô hình Support Vector Machine (SVM) và mô hình K-NN với k = 9 là ba mô hình phân lớp tốt nhất dựa vào kết quả trên. Ba mô hình này có thể được sử dụng để phân loại các mẫu dữ liệu trong tập dữ liệu kiểm tra với độ chính xác cao.

Cũng có thể thấy rằng bộ dữ liệu đã được thu thập tốt, đầy đủ các biến nên có thể huấn luyện hiệu quả trên nhiều mô hình. Naive Bayes kém hiểu quá nhất có thể do các biến của bộ dữ liệu không độc lập với nhau.

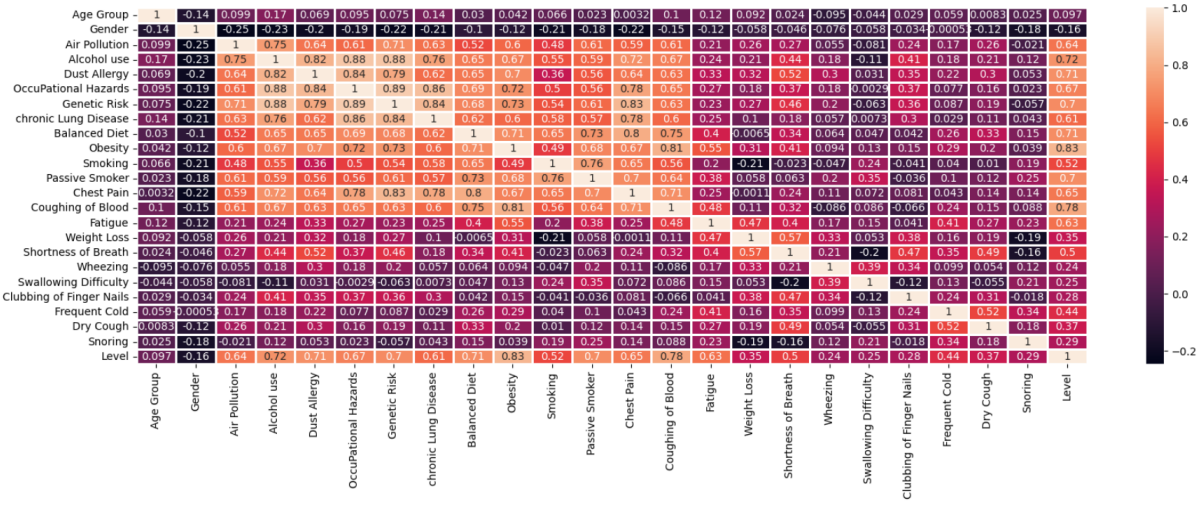
VI. Mô hình phân lớp đã giảm chiều dữ liệu.

A. Giảm chiều dữ liệu

Nhóm thực hiện giảm chiều dữ liệu dựa vào độ tương quan giữa các biến theo biểu đồ heatmap nhằm giảm kích thước dữ liệu, thời gian xử lý để thực hiện mô hình.

Sau đó ra so sánh với mô hình cũ để có thể đánh giá việc giảm chiều dữ liệu có thể đưa ra kết quả tốt hơn hay không.

Nhóm đặt ra mốc độ tương quan giữa các biến để lượt bỏ sẽ là 0,8.



Hình 21: Heatmap các biến ban đầu

Dựa vào biểu đồ Heatmap nhóm đưa ra các biến có độ tương quan cao hơn mốc nhóm đề ra:

Alcohol use - Dust Allergy

Alcohol use - OccuPational Hazards

Alcohol use - Genetic Risk

Dust Allergy - OccuPational Hazards

OccuPational Hazards - Genetic Risk

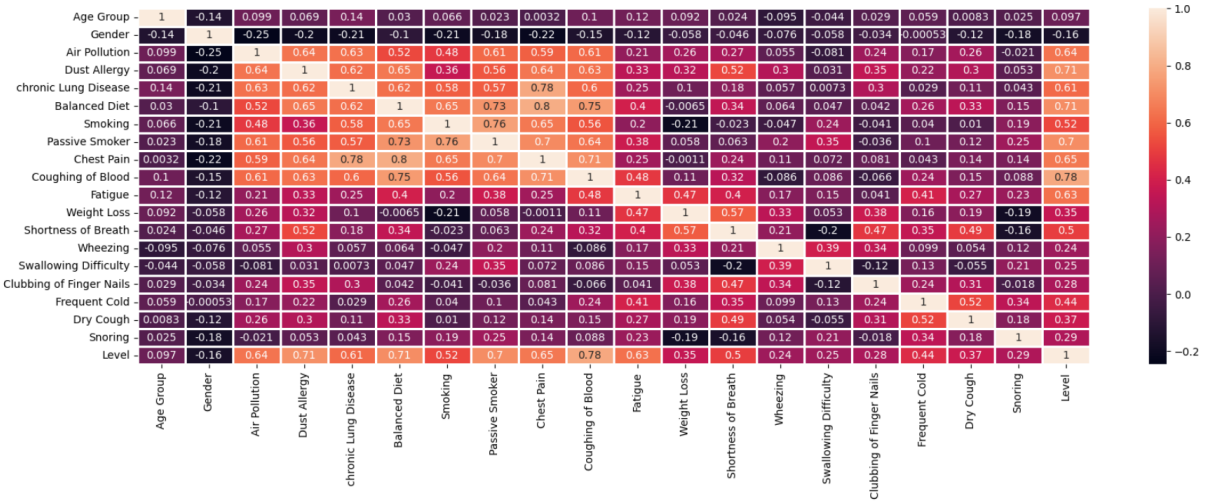
OccuPational Hazards - chronic Lung Disease

Genetic Risk - chronic Lung Disease

Genetic Risk - Chest Pain

Obesity - Coughing of Blood

Nhóm quyết định sẽ lược bỏ các biến Alcohol use, OccuPational Hazards, Genetic Risk, Obesity. Biểu đồ heatmap sau khi đã lược bỏ đi các biến.



Hình 22: Heatmap đã lược bỏ 4 biến

Nhóm thực hiện chạy thử bộ dữ liệu đã rút gọn với mô hình Decision Tree với thang đo entropy để so sánh hiệu quả dự đoán của mô hình có bị ảnh hưởng nhiều hay không.

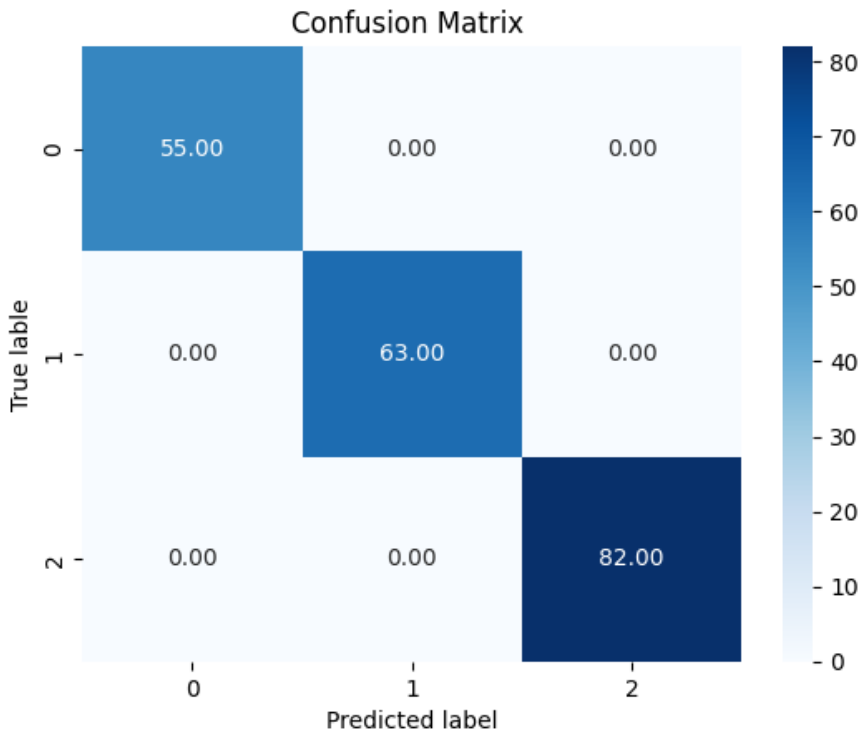
```
# Áp dụng mô hình Decision Tree
X = data.drop('Level', axis = 1)
y = data.Level

## Chia tập dữ liệu thành training, test sets theo tỷ lệ 80:20
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .2,
random_state = 42)
```

```
clf   = DecisionTreeClassifier(criterion='entropy')
model = clf.fit(X, y)
y_pred = model.predict(X_test)
```

Sử dụng các phương thức để tính các điểm chỉ số

```
print(f'Accuracy ={accuracy_score(y_test, y_pred)*100:.2f}%')
print(f'Precision                                ={(precision_score(y_test,
y_pred,average="micro"))*100:.2f}%')
print(f'recall ={recall_score(y_test, y_pred,average="micro")*100:.2f}%')
print(f'f1-score ={f1_score(y_test, y_pred,average="micro")*100:.2f}%')
```

Kết quả phân lớp của mô hình mới bằng phương pháp Decision Tree



Hình 23: Confusion matrix của Decision Tree mới

Phương pháp	Accuracy	Precision	Recall	F1-score
Decision Tree	100%	100%	100%	100%

Bảng 9: Kết quả phân lớp của mô hình mới bằng phương pháp Decision Tree

B. Đánh giá mô hình mới và hiệu quả giảm chiều dữ liệu.

Với mô hình mới, nhóm đã lựa chọn giảm đi 4 biến Alcohol use, OccuPational Hazards, Genetic Risk, Obesity, số lượng biến giảm từ 23 biến đặc trưng xuống 19 biến, giảm 17% dữ liệu. Những biến bị giảm đi có tương quan cao với 1 hoặc nhiều biến khác, được thể hiện qua phân phối của biến, tương quan của biến với biến mục tiêu và đặc biệt là tương quan của 2 biến với nhau.

Lựa chọn mô hình tốt nhất trước khi giảm chiều là Decision Tree với độ chính xác 100%. Thu được mô hình mới với bộ dữ liệu đã giảm, kết quả nhận được là mô hình Decision Tree mới vẫn đạt độ chính xác 100%.

Từ đây, chúng ta có thể thấy rằng việc giảm bớt các biến có tương quan mạnh cho bộ dữ liệu là rất hiệu quả, độ chính xác vẫn là 100% và hỗ trợ việc xây dựng mô hình nhanh hơn.

PHỤ LỤC

A. DANH MỤC HÌNH ẢNH

B. DANH MỤC BẢNG BIỂU

TÀI LIỆU THAM KHẢO

1. Các file Colab của thầy Tế đăng tải trên trang LMS môn LTPTDL, DM, DV
2. *Lung Cancer Predict*. (n.d.). Kaggle. Retrieved November 23, 2022, from https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link
3. Đồ án lập trình phân tích dữ liệu Airlines Customer satisfaction. Nhóm sinh viên đại học Kinh tế TP.HCM.
4. Đề tài biểu diễn trực quan dữ liệu Dự đoán giá Kim cương. Nhóm sinh viên đại học Kinh tế TP.HCM.

Phân công công việc:

STT	Họ và tên	MSSV	Công việc	Mức độ hoàn thành
1	Trương Vũ Phương Quỳnh	31211027668	- Tìm kiếm bộ dữ liệu - Biểu đồ thể hiện mối liên hệ của mức độ bệnh với giới tính. - Biểu đồ thể hiện mối liên hệ của mức độ bệnh với độ tuổi. - Mở rộng. - Biểu đồ tương quan của các biến. - Tạo bộ dữ liệu train, test. - Phân lớp bằng phương pháp Support Vector Machine.	100%
2	Đinh Công Thành	31211027670	- Tìm kiếm bộ dữ liệu. - Xử lý dữ liệu bị thiếu và outliers. - Chuyển dạng dữ liệu. - Phân lớp bằng phương pháp Naive Bayes. - Đánh giá mô hình phân lớp bằng biểu đồ. - Giảm chiều. - Đánh giá mô hình mới.	100%
3	Nguyễn Thị Thơm	31211027673	- Tìm kiếm bộ dữ liệu. - Tóm tắt đề tài. - Trực quan biến thuộc tính. - Trực quan biến mục tiêu. - Biểu đồ mối liên hệ nhóm tuổi và giới tính.	100%

			- Phân lớp bằng phương pháp K-NN classification. - Phân lớp bằng phương pháp Decision Tree.	
4	Đào Bùi Hương Thuỳ	31211027675	- Nhóm trưởng. - Tìm kiếm bộ dữ liệu. - Mô tả bộ dữ liệu. - Phương pháp thực hiện. - Biểu đồ thể hiện mối liên hệ giữa các biến với mức độ bệnh. - Biểu đồ thể hiện mối liên hệ giữa các biến tương quan trên 0,8. - Bảng tổng các chỉ số. - Tổng hợp code.	100%