HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN KHOA HỌC DỮ LIỆU

BÁO CÁO

ĐỀ TÀI: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN TỈ LỆ TỬ VONG CỦA BỆNH NHÂN MẮC COVID-19

Nhóm lớp học: 03

Giảng viên giảng dạy: TS. VŨ HOÀI NAM

Thành viên: Hà Hải Nam – B20DCCN028

Hà Nội – 2023

I. LÝ DO CHỌN ĐỀ TÀI

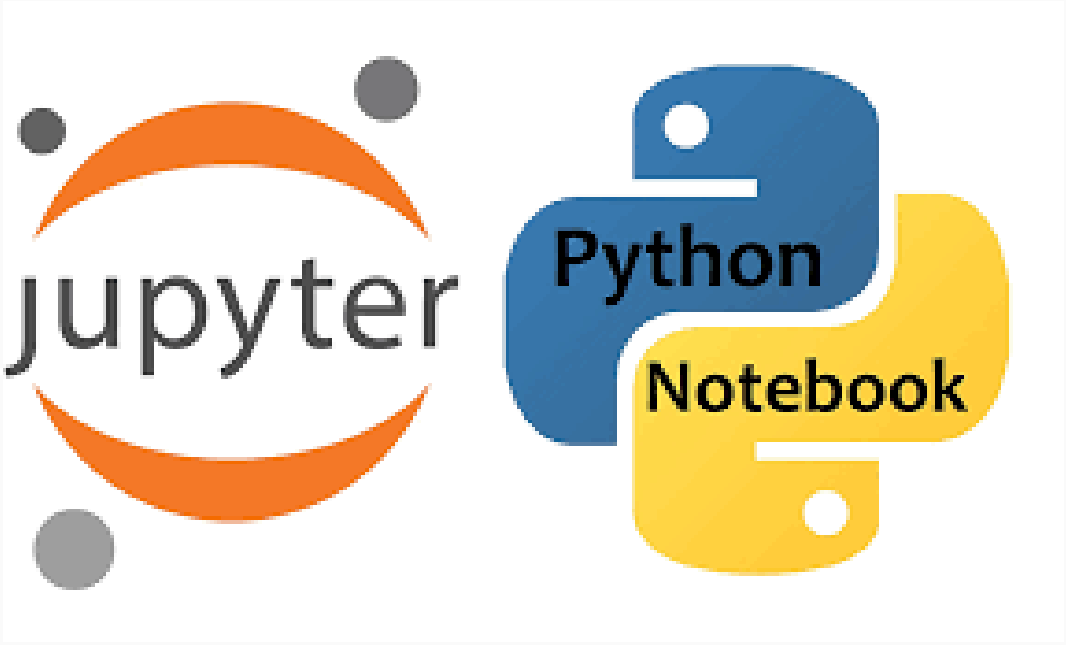
- **Ứng dụng thực tế**: Dự đoán tỉ lệ tử vong của bệnh nhân có thể giúp các bệnh viện có thể sớm cứu chữa kịp thời bệnh nhân có tỉ lệ tử vong cao từ đó giảm thiểu rủi ro không đáng có.
- **Dữ liệu lớn và đa dạng**: Các Dataset về các bệnh nhân mắc COVID-19 là dữ liệu lớn và có kiểu dữ liệu đơn giản phù hợp cho việc làm quen và học tập.
- **Quản lý tài nguyên y tế**: Dự án này sẽ giúp cho chúng ta có thể dự đoán trước các bệnh nhân có tỉ lệ tử vong cao hơn để tiến hành điều trị tập trung để có thể cứu chữa kịp thời.
- **Học máy và Khoa học dữ liệu**: Dự đoán tỉ lệ tử vong của bệnh nhân có liên quan đến nhiều khía cạnh của học máy và khoa học dữ liệu, bao gồm tiền xử lý dữ liệu, chọn thuật toán, đào tạo mô hình, và đánh giá hiệu suất. Đây là cơ hội để học và thực hành các kỹ năng quan trọng trong lĩnh vực này.

II. CÁC CÔNG NGHỆ SỬ DỤNG

1. Ngôn ngữ lập trình Python

- Lý do lựa chọn Python:
  - Python là ngôn ngữ phổ biến, dễ sử dụng và được dùng rộng rãi trong các lĩnh vực khác nhau ví dụ như trí tuệ nhân tạo, khoa học dữ liệu, tự động hóa,…
  - Python có nhiều thư viện khác nhau hỗ trợ cho việc phân tích dữ liệu, tạo modul học máy dễ dàng.
- Các thư viện được sử dụng trong dự án:
  - Scikit-learn: Thư viện học máy.
  - Pandas: Thư viện hỗ trợ lưu trữ dữ liệu
  - Seaborn: Xây dựng dựa trên Matplotlib dùng để vẽ biểu đồ thống kê.
  - Matplotlib: Thư viện hỗ trợ vẽ biểu đồ.

2. Jupyter Notebook



Jupyter Notebook là một nền tảng tính toán khoa học mã nguồn mở, bạn có thể sử dụng nó để tạo và chia sẻ các tài liệu có chứa code trực tiếp, phương trình, trực quan hóa dữ liệu và văn bản tường thuật

- Lí do sử dụng Jupyter Notebook:
    - Jupyter Notebook hỗ trợ người dùng xem kết quả của từng cell mà không phụ thuộc vào phần khác của code.
    - Jupyter Notebook lưu trữ kết quả của mọi ô đang chạy
    - Jupyter Notebook hỗ trợ trực quan hóa dữ liệu và hiển thị các loại biểu đồ giúp người dùng dễ dàng hình dung hơn về dữ liệu.

III. MÔ HÌNH HỌC MÁY ĐƯỢC SỬ DỤNG

Dự án sử dụng mô hình học máy Random Forest.

Random Forest là thuật toán được phát triển lên tiếp dựa trên thuật toán Decision Tree có thể giúp chúng ta giải quyết cả 2 bài toán hồi quy và phân loại.

Ý tưởng của thuật toán: Thuật toán sẽ tạo dựng random các cây quyết định khác nhau và sao đó kết quả cuối cùng sẽ được tổng hợp lại thông qua kết quả của các cây quyết định đó.

Ưu điểm:

- Có độ chính xác cao.
- Có khả năng xử lý dữ liệu lớn.
- Có khả năng xử lý cho cả 2 bài toán hồi quy và phân loại.
- Làm việc tốt trên các dữ liệu bị thiếu.

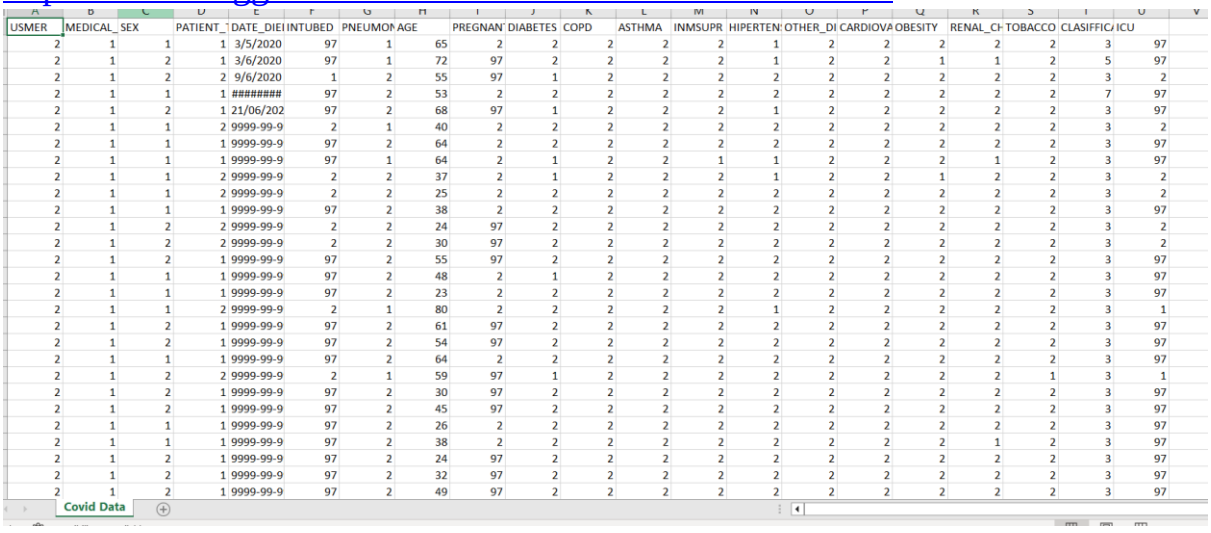
Nhược điểm:

- Tốc độ tính toán chậm.
- Khó để giải thích.

IV. BẮT ĐẦU DỰ ÁN

1. Lấy Dataset

Chọn Dataset ở trên trang web Kaggel ở đường link sau đây:
https://www.kaggle.com/datasets/meirnizri/covid19-dataset



USMER	MEDICAL_	SEX	PATIENT_	DATE_DIE	INTUBED	PNEUMON	AGE	PREGNAN	DIABETES	COPD	ASTHMA	INMSUPR	HIPERTEN	OTHER_DI	CARDIOVA	OBESITY	RENAL_CH	TOBACCO	CLASIFFIC	ICU
2	1	1	1	3/5/2020	97	1	65	2	2	2	2	2	1	2	2	2	2	2	3	97
2	1	2	1	3/6/2020	97	1	72	97	2	2	2	2	1	2	2	1	1	2	5	97
2	1	2	2	9/6/2020	1	2	55	97	1	2	2	2	2	2	2	2	2	2	3	2
2	1	1	1	########	97	2	53	2	2	2	2	2	2	2	2	2	2	2	7	97
2	1	2	1	21/06/202	97	2	68	97	1	2	2	2	1	2	2	2	2	2	3	97
2	1	1	2	9999-99-9	2	1	40	2	2	2	2	2	2	2	2	2	2	2	3	2
2	1	1	1	9999-99-9	97	2	64	2	2	2	2	2	2	2	2	2	2	2	3	97
2	1	1	1	9999-99-9	97	1	64	2	1	2	2	1	1	2	2	2	1	2	3	97
2	1	1	2	9999-99-9	2	2	37	2	1	2	2	2	1	2	2	1	2	2	3	2
2	1	1	2	9999-99-9	2	2	25	2	2	2	2	2	2	2	2	2	2	2	3	2
2	1	1	1	9999-99-9	97	2	38	2	2	2	2	2	2	2	2	2	2	2	3	97
2	1	2	2	9999-99-9	2	2	24	97	2	2	2	2	2	2	2	2	2	2	3	2
2	1	2	2	9999-99-9	2	2	30	97	2	2	2	2	2	2	2	2	2	2	3	2
2	1	2	1	9999-99-9	97	2	55	97	2	2	2	2	2	2	2	2	2	2	3	97
2	1	1	1	9999-99-9	97	2	48	2	1	2	2	2	2	2	2	2	2	2	3	97
2	1	1	1	9999-99-9	97	2	23	2	2	2	2	2	2	2	2	2	2	2	3	97
2	1	1	2	9999-99-9	2	1	80	2	2	2	2	2	1	2	2	2	2	2	3	1
2	1	2	1	9999-99-9	97	2	61	97	2	2	2	2	2	2	2	2	2	2	3	97
2	1	2	1	9999-99-9	97	2	54	97	2	2	2	2	2	2	2	2	2	2	3	97
2	1	1	1	9999-99-9	97	2	64	2	2	2	2	2	2	2	2	2	2	2	3	97
2	1	2	2	9999-99-9	2	1	59	97	1	2	2	2	2	2	2	2	2	1	3	1
2	1	2	1	9999-99-9	97	2	30	97	2	2	2	2	2	2	2	2	2	2	3	97
2	1	2	1	9999-99-9	97	2	45	97	2	2	2	2	2	2	2	2	2	2	3	97
2	1	1	1	9999-99-9	97	2	26	2	2	2	2	2	2	2	2	2	2	2	3	97
2	1	1	1	9999-99-9	97	2	38	2	2	2	2	2	2	2	2	2	1	2	3	97
2	1	2	1	9999-99-9	97	2	24	97	2	2	2	2	2	2	2	2	2	2	3	97
2	1	2	1	9999-99-9	97	2	32	97	2	2	2	2	2	2	2	2	2	2	3	97
2	1	2	1	9999-99-9	97	2	49	97	2	2	2	2	2	2	2	2	2	2	3	97

Covid Data

Đây là dữ liệu được cung cấp bởi chính phủ Mexico về các bệnh nhân bị nhiễm COVID-19 tại đó. Dữ liệu gồm 21 cột với 1048576 hàng trong đó chủ yếu là dữ liệu dưới dạng số tự nhiên.

2. Xử lý tiền dữ liệu và phân tích dữ liệu

a) Tổng quan về dữ liệu

Theo như thông tin được cung cấp các giá trị 97, 98, 99 là các dữ liệu bị thiếu. Trong đó các cột có giá trị Boolean thì 1 có nghĩa là có, 2 có nghĩa là không:

- Sex: 1 là nữ , 2 là nam.
- Age: tuổi của bệnh nhân.
- Classification: Kết quả xét nghiệm Covid với các mức độ từ 1 đến 3.
- Patient type: Loại hình nhận chăm sóc. 1 là ở nhà, 2 là nhập viện.
- Pneumonia: Bệnh nhân có bị viêm túi khí hay không.
- Pregnancy: Bệnh nhân có mang thai hay không.
- Diabetes: Bệnh nhân có bị tiểu đường hay không.
- Copd: Bệnh nhân có bị bệnh phổi tắc nghẽn mãn hính hay không.
- Asthma: Bệnh nhân có bị hen suyễn hay không.
- Inmsupr: Bệnh nhân có bị ức chế miễn dịch hay không.
- Hypertension: Bệnh nhân có bị tăng huyết áp hay không.
- Cardiovascular: Bệnh nhân có bị bệnh liên quan đến tim và mạch máu hay không.
- Renal chronic: Bệnh nhân có bị bệnh thận mãn tính hay không.
- Other disease: Bệnh nhân có bị các bệnh khác hay không.
- Obesity: Bệnh nhân có bị bệnh béo phì hay không.
- Tabacco: Bệnh nhân có hút thuốc lá hay không.
- Usmr: Cấp độ điều trị mà bệnh nhân được sử dụng.
- Medical unit: Cấp độ của tổ chức y tế đang tiến hành chăm sóc.
- Intubed: Bệnh nhân có được sử dụng máy thở hay không.
- ICU: Bệnh nhân có được đưa vào ICU hay không.
- Date died: Ghi 9999-99-99 nếu bệnh nhân còn sống.

b) Xử lý các dữ liệu bị thiếu

```python
for col in df.columns:
    print(f'Cột {col} có số giá trị bị thiếu là')
    print(df.loc[df[col] == 97, col].count() + df.loc[df[col] == 98, col].count() + df.loc[df[col] == 99, col].count())
```
✓ 0.1s

Để kiểm tra các dữ liệu bị thiếu trong Dataset, tôi sử dụng hàm df.loc[df[col] == a, col].count() để đếm giá trị trong cột col có giá trị bằng a. Từ đó ta sẽ có được số các giá trị bị biếu ở trong các cột dữ liệu.

Cột USMER có số giá trị bị thiếu là
0
Cột MEDICAL_UNIT có số giá trị bị thiếu là
0
Cột SEX có số giá trị bị thiếu là
0
Cột PATIENT_TYPE có số giá trị bị thiếu là
0
Cột DATE_DIED có số giá trị bị thiếu là
0
Cột INTUBED có số giá trị bị thiếu là
855869
Cột PNEUMONIA có số giá trị bị thiếu là
16003
Cột AGE có số giá trị bị thiếu là
345
Cột PREGNANT có số giá trị bị thiếu là
527265
Cột DIABETES có số giá trị bị thiếu là
3338
Cột COPD có số giá trị bị thiếu là
3003
Cột ASTHMA có số giá trị bị thiếu là
2979
Cột INMSUPR có số giá trị bị thiếu là
3404
Cột HIPERTENSION có số giá trị bị thiếu là
3104
Cột OTHER_DISEASE có số giá trị bị thiếu là
5045
Cột CARDIOVASCULAR có số giá trị bị thiếu là

3076
Cột OBESITY có số giá trị bị thiếu là
3032
Cột RENAL_CHRONIC có số giá trị bị thiếu là
3006
Cột TOBACCO có số giá trị bị thiếu là
3220
Cột CLASIFFICATION_FINAL có số giá trị bị thiếu là
0
Cột ICU có số giá trị bị thiếu là
856032

Nhận thấy có 3 cột bị thiếu dữ liệu nhiều nhất nên chúng ta sẽ tiến hành xử lý 3 cột dữ liệu đó trước tiên.
Các cột INTUBED và cột ICU bị thiếu gần 80% dữ liệu nên tôi sử dụng hàm drop 2 cột dữ liệu đó.

```
df.drop(['ICU', 'INTUBED'], axis=1, inplace=True)
df.shape
```
✓ 0.1s

Cột dữ liệu PREGNANT có thể được điền dựa trên các dữ liệu ở cột giới tính. Bằng cách thay đổi các dữ liệu ở cột PREGNANT bị thiếu mà ở hàng có cột SEX là 2 bằng 2 bằng hàm sau:

```
df.PREGNANT = df.PREGNANT.replace(97, 2)
```

Tiếp đó xét thấy số lượng các dữ liệu bị thiếu ở các dòng quá nhỏ so với tổng số dữ liệu ta có nên chúng ta sẽ loại bỏ các dữ liệu đó bằng hàm drop như sau:

```
for i in df.columns:
    df.drop(df[df[i]==97].index,inplace=True)
    df.drop(df[df[i]==98].index,inplace=True)
    df.drop(df[df[i]==99].index,inplace=True)
```

Để xác định bệnh nhân đã tử vong hay chưa chúng ta cần thêm 1 cột DEATH có 2 giá trị 0 và 1 trong đó: 0 là biểu thị bệnh

nhân đó còn sống và 1 là biểu thị bệnh nhân đã tử vong. Chúng ta sử dụng hàm sau:

```
def isD(x):
    if x == '9999-99-99':
        return 0
    else :
        return 1

df['DEATH'] = df['DATE_DIED'].apply(isD)
df.drop(['DATE_DIED'], axis=1, inplace=True)
```

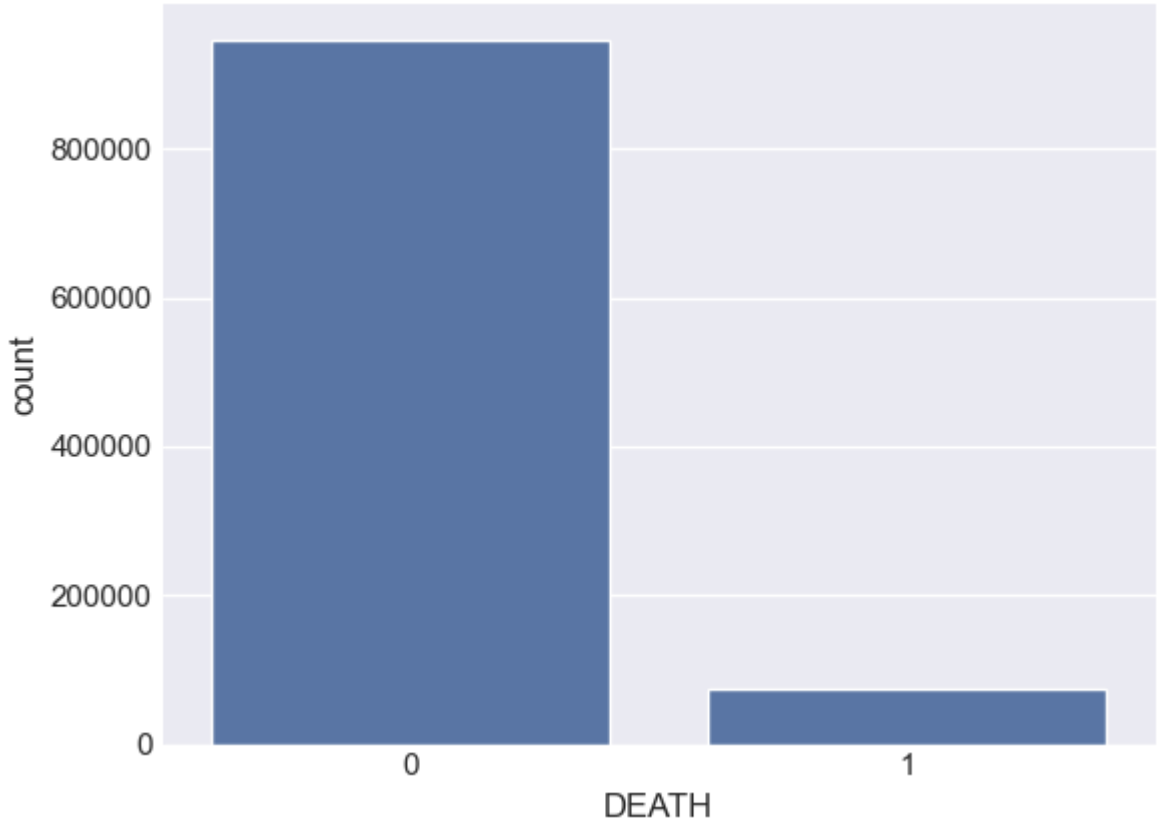
Trong đó isD là để tạo giá trị cho cột DEATH dựa trên cột DATE_DIED.

c) Các biểu đồ

Do các dữ liệu hầu hết được biểu diễn dưới dạng Boolean và có AGE là số tự nhiên nên chúng ta sẽ sử dụng 2 hàm sau để vẽ biểu đồ:

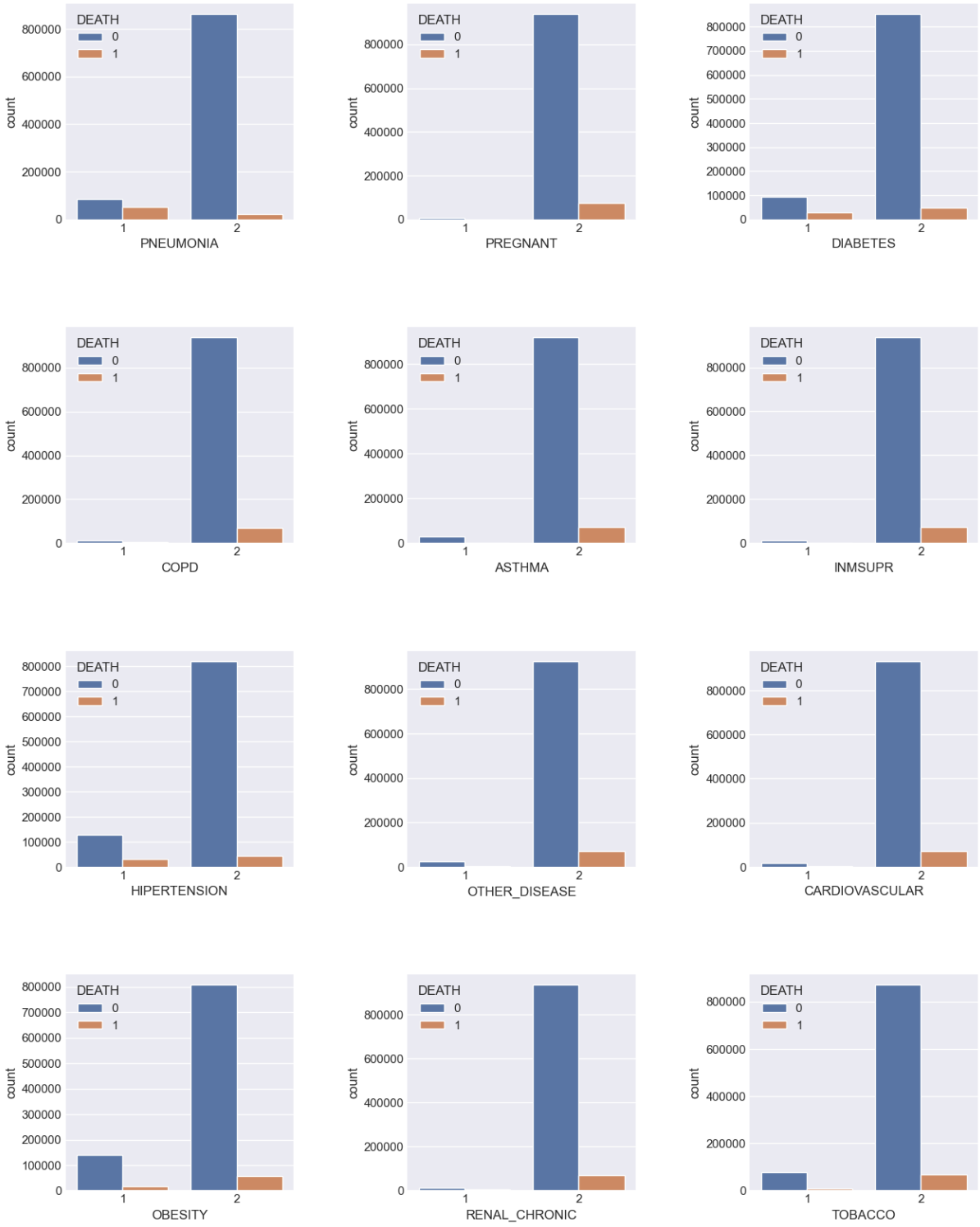
- `sns.countplot()` #hàm được sử dụng để vẽ biểu đồ dành cho biến phân loại
- `sns.histplot()` #hàm được sử dụng để vẽ biểu đồ hiển thị phân phối của 1 biến liên tục

a. Biểu đồ tỉ lệ tử vong của bệnh nhân mắc COVID-19:



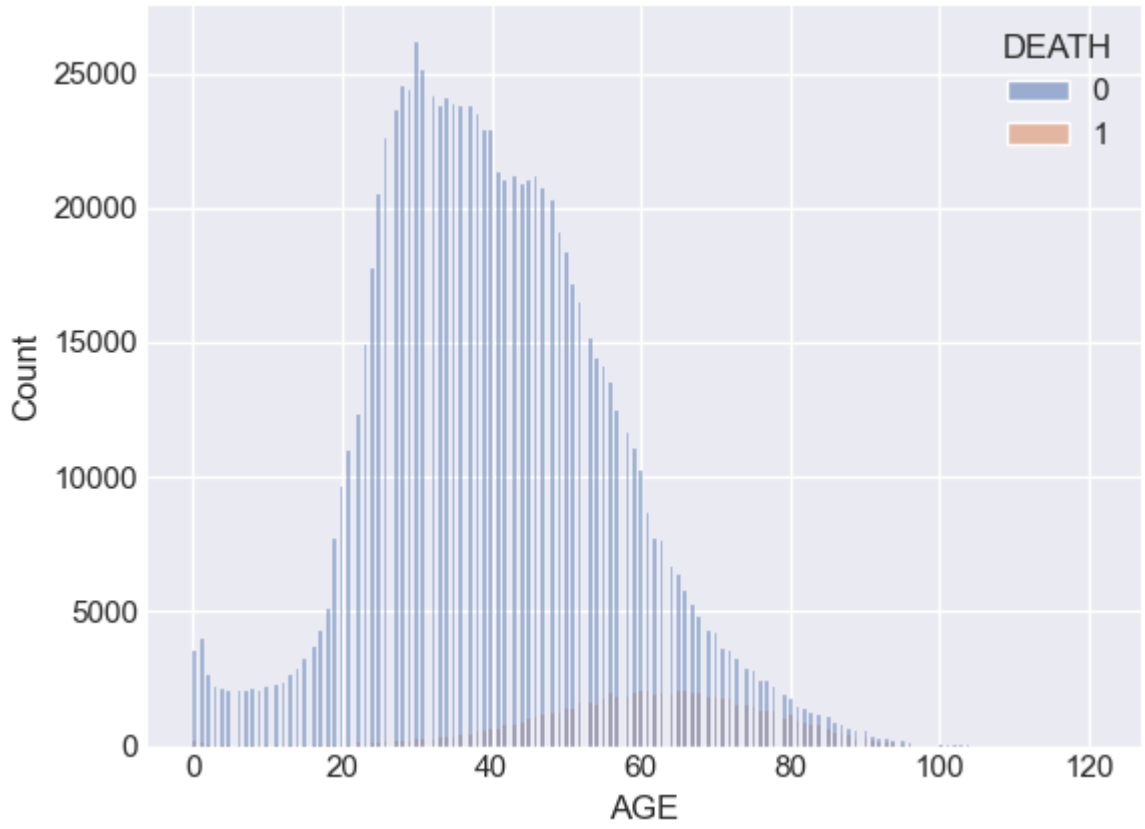
Biểu đồ cho thấy tỉ lệ tử vong của bệnh nhân mắc bệnh COVID-19 ở nơi khảo sát tại thời điểm đó là rất thấp.

b. Các biểu đồ thể hiện tỉ lệ tử vong của các bệnh nhân mắc COVID-19 cùng 1 số bệnh khác:



Các biểu đồ trên cho ta thấy được tỉ lệ tử vong của các bệnh nhân mắc các bệnh khác và bệnh COVID-19 có tỉ lệ tử vong rất cao.

c. Biểu đồ thể hiện tỉ lệ tử vong của các bệnh nhân theo số tuổi:



Biểu đồ cho thấy tỉ lệ tử vong của bệnh nhân lớn tuổi cao hơn hẳn so với tỉ lệ tử vong của bệnh nhân tuổi thanh thiếu niên.

3. Xây dựng mô hình và đánh giá kết quả của mô hình

Đầu tiên, chúng ta cần chia dữ liệu thành nhãn và tập dữ liệu như sau:

```
x = df.drop('DEATH', axis=1)
y = df['DEATH']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

Và sau đó chia thành 2 tập test và train với tỉ lệ là 8 : 2.

Cuối cùng, cho mô hình Random Forest huấn luyện với tệp dữ liệu train đã tạo và chạy tệp dữ liệu test. Từ đó ta có thể tính được độ chính xác của mô hình trên như sau:

Nhận xét: Mô hình cho ra độ chính xác khá cao là 0,9346.

4. Phương hướng phát triển

- Nâng cao hiệu suất của mô hình học máy bằng cách kết hợp nhiều mô hình học máy khác nhau
- Tìm thêm dữ liệu khác để có thể tạo ra dữ liệu đầu vào tốt hơn.