University of Houston

Spring 2023

COSC 4337

COVID-19 RISK OF DEATH PREDICTIVE ANALYSIS

Group member:

Han Hoang - 2088807

Johnny Diep - 1336047

Carson Dial - 1155918

Introduction

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. It was originally identified in China in 2019 and became pandemic in 2020. Most people infected with COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. However, older people, and those with underlying health conditions like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop severe symptoms and in some cases death. During waves of the global pandemic, healthcare providers have faced the shortage of medical resources and, at the same time, were required to efficiently distribute it to save the lives of millions of patients. Therefore, being able to predict what kind of resource an individual might require at the time of being tested positive or even before that will be of great help to the authorities and hospitals.

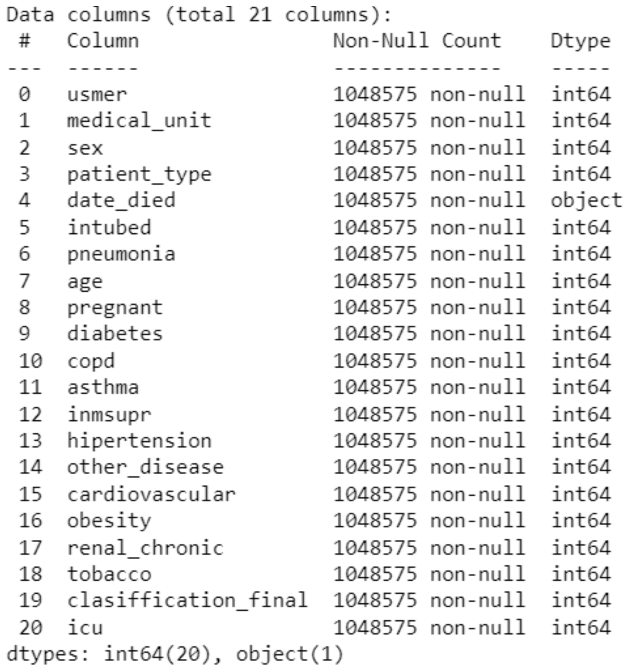
Given a COVID-19 patient's current symptom, status, and medical history, our main goal is to build a machine learning model that will predict whether the patient is in high risk of death from coronavirus. In the first project milestone, we will be working on data insights, exploratory data analysis and feature selection.

Data Description

I. Data Source and Link: https://www.kaggle.com/datasets/meirnizri/covid19-dataset

II. Data Overview

The dataset was provided by the Mexican government, that contains an enormous number of anonymized patient-related information including pre-conditions. The raw dataset consists of 1,048,575 unique patients and 21 unique features with the data type as follows:



```
Data columns (total 21 columns):
 #   Column                Non-Null Count    Dtype 
---  ------                --------------    ----- 
 0   usmer                 1048575 non-null  int64 
 1   medical_unit          1048575 non-null  int64 
 2   sex                   1048575 non-null  int64 
 3   patient_type          1048575 non-null  int64 
 4   date_died             1048575 non-null  object
 5   intubed               1048575 non-null  int64 
 6   pneumonia             1048575 non-null  int64 
 7   age                   1048575 non-null  int64 
 8   pregnant              1048575 non-null  int64 
 9   diabetes              1048575 non-null  int64 
 10  copd                  1048575 non-null  int64 
 11  asthma                1048575 non-null  int64 
 12  inmsupr               1048575 non-null  int64 
 13  hipertension          1048575 non-null  int64 
 14  other_disease         1048575 non-null  int64 
 15  cardiovascular        1048575 non-null  int64 
 16  obesity               1048575 non-null  int64 
 17  renal_chronic         1048575 non-null  int64 
 18  tobacco               1048575 non-null  int64 
 19  clasiffication_final  1048575 non-null  int64 
 20  icu                   1048575 non-null  int64 
dtypes: int64(20), object(1)
```

Among the attributes, *date_died* (object datatype) type and *age* (integer datatype) are numerical attributes. The remaining attributes are all categorical but represented as integer values. The feature details are described as follows:

1. usmer: indicates whether the patient treated medical units of the first or second level.
2. sex: female (sex = 1) and male (sex = 2)

3. age: of the patient.
4. classification: covid test findings. Values 1-3 mean that the patient was diagnosed with covid in different degrees. 4 or higher means that the patient is not a carrier of covid or that the test is inconclusive.
5. patient type: type of care the patient received in the unit. Value 1 means returned home and 2 means hospitalization.
6. date died: the date of death. If the patient recovers, the value will be 9999-99-99.
7. medical unit: type of institution of the National Health System that provided the care (value 1-13)
8. pneumonia: whether the patient already have air sacs inflammation or not.
9. pregnancy: whether the patient is pregnant or not.
10. diabetes: whether the patient has diabetes or not.
11. copd: whether the patient has Chronic obstructive pulmonary disease or not.
12. asthma: whether the patient has asthma or not.
13. inmsupr: whether the patient is immunosuppressed or not.
14. hypertension: whether the patient has hypertension or not.
15. cardiovascular: whether the patient has heart or blood vessels related disease.
16. renal chronic: whether the patient has chronic renal disease or not.
17. other disease: whether the patient has other disease or not.
18. obesity: whether the patient is obese or not.
19. tobacco: whether the patient is a tobacco user.
20. intubed: whether the patient was connected to the ventilator.
21. icu: whether the patient had been admitted to an Intensive Care Unit.

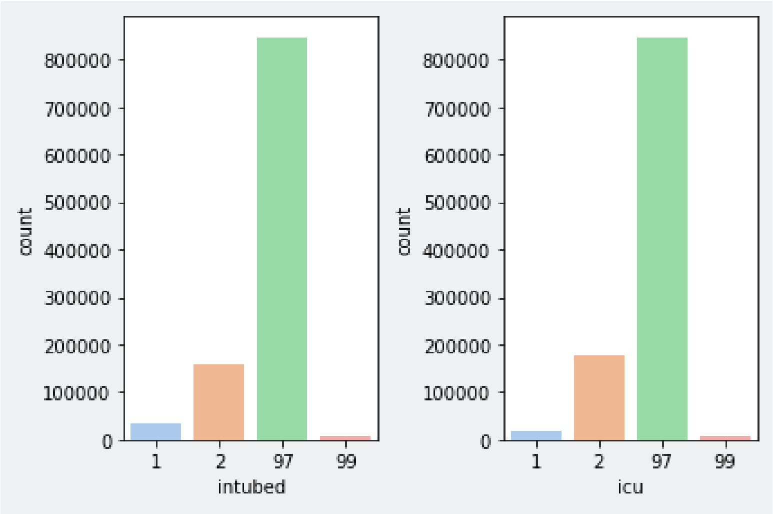
As for features regarding medical conditions and resources (i.e. feature 8-21), they are binary variables with value 1 for "yes" and 2 for "no". In addition, values as 97, 98 and 99 are missing data that will be handled before moving forward to building the predictive model.

Exploratory Data Analysis

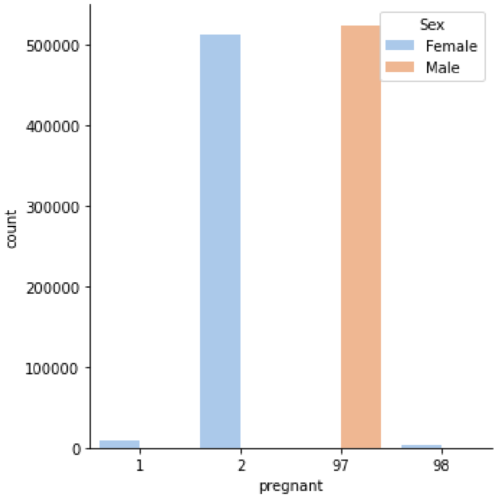
I. Handling Missing Values

Out of 21 features, there are some features we expect to be binary which are **intubed, pneumonia, pregnant, diabetes, copd, asthma, inmsupr, hipertension, other_disease, cardiovascular, obesity, renal_chronic, tobacco, icu.** However, all these columns contain more than 2 values in which values as 97, 98 and 99 are missing data, which we don't want in the dataset.

First, we proceed to visualize unique values of 3 features: **intubed, icu, pregnant.**

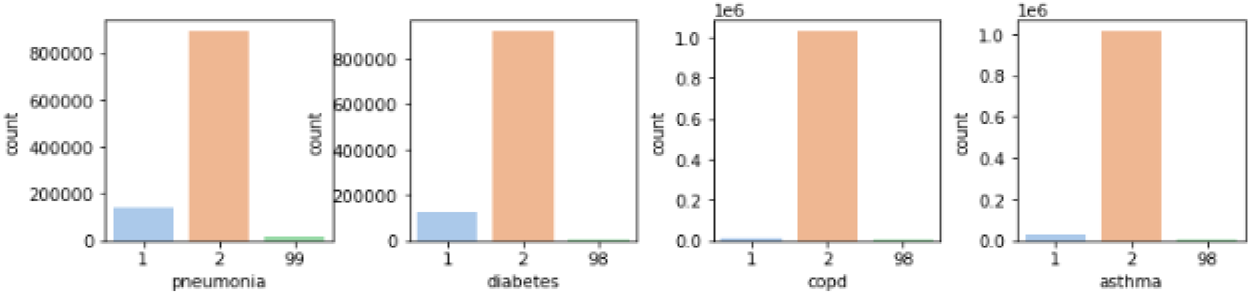


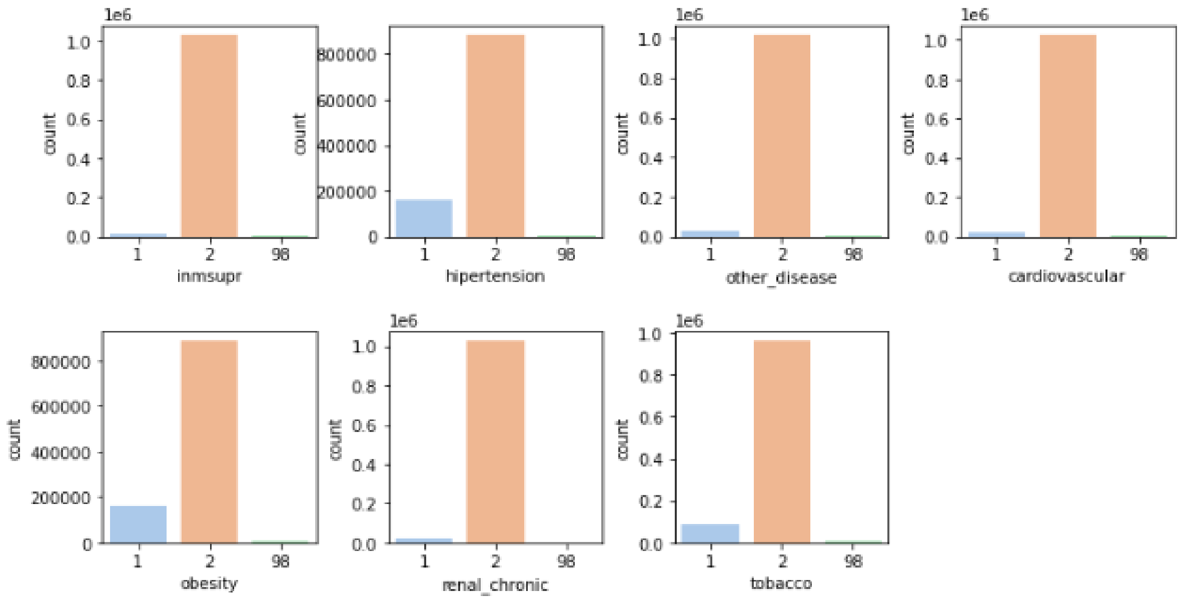
From the plots, we can see that in columns **intubed** and **icu**, there is more than 70% data of the data missing. If we keep the data in the dataset, it will cause problems when trying to analyze the rest of the features.. Therefore, we decided to drop both columns from the dataset.



This plot explains the reason why half data in column **pregnant** is missing. Notably, all values 97 relate to male patients. Therefore, we want to replace 97 with a value of 2 which means 'not pregnant'. In addition, we will remove rows containing value 98 in this column since these are the real missing data for female patients.

Next, we want to look at the remaining binary variables (i.e pneumonia, diabetes, copd, asthma, inmsupr, hipertension, other_disease, cardiovascular, obesity, renal_chronic, tobacco):





It is visible that missing data accounts for very small percent in every countplot, so we will simply remove any rows with missing data (value 98, 99) in these columns.

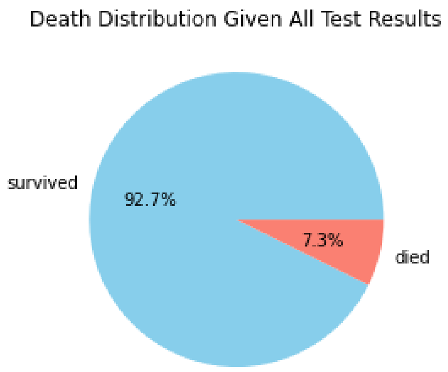
After plotting and analyzing the proportion of missing data in each column, we proceed to clean and preprocess our data following these steps:

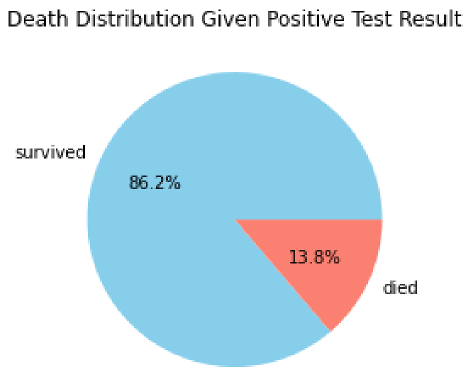
- Remove 2 columns **intubed** and **icu** due to the disproportion of missing data over valid data.
- Remove missing data (labeled as 98) and convert value 97 to 2 in column **pregnant.**
- Remove rows with missing data (labeled as 98, 99) in following columns: **pneumonia, diabetes, copd, asthma, inmsupr, hipertension, other_disease, cardiovascular, obesity, renal_chronic, tobacco**

Lastly, we created a new column **death** to be the response variable. This varible indicates whether the patient died from the coronavirus based on the condition of column **date_died**. If date_died is a specific date, we set *death = 1 (yes)*, otherwise if date_died = 9999-99-99, then *death = 2 (no)*. After handling the missing data, our dataset now has ***19 features*** and ***1,021,977 unique patients.***

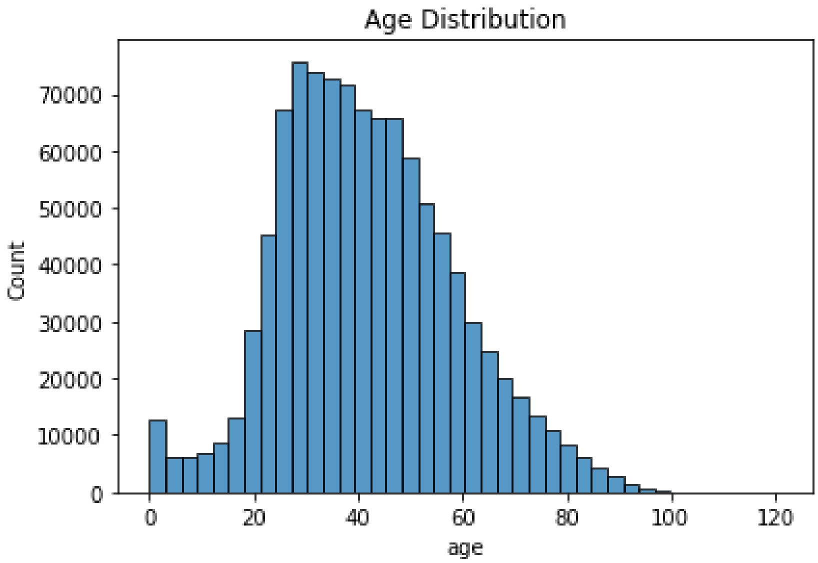
II. Data Visualization

Firstly, we want to see the ratio between number of survivals and deaths in the response variable **death** using pie charts.

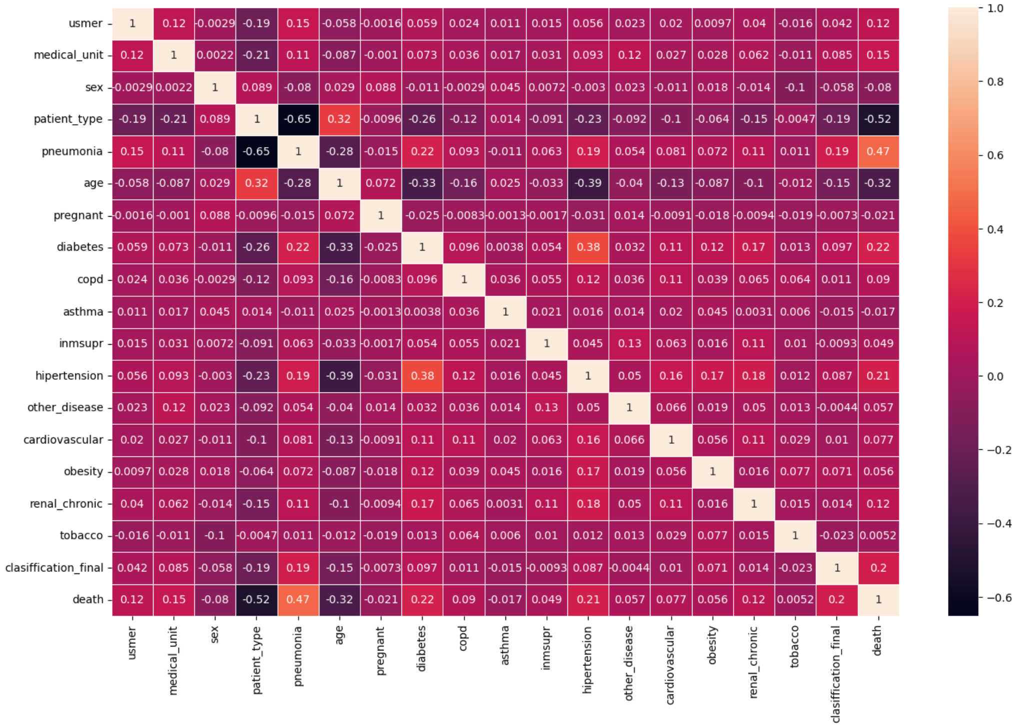


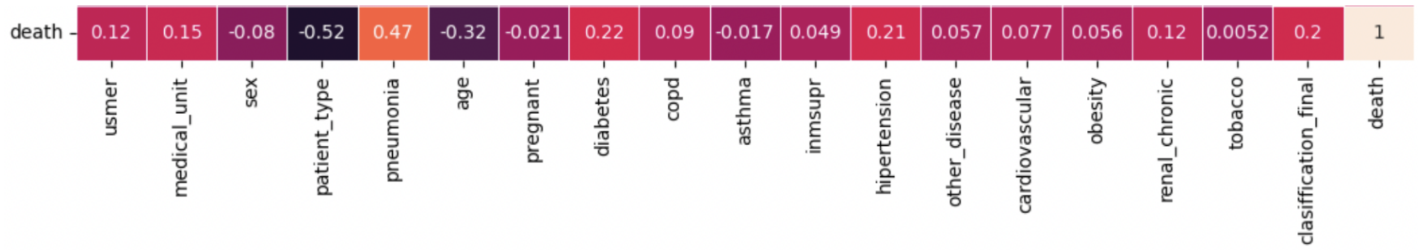


The left pie chart shows the death distribution in the whole dataset. From this plot, we can infer that there is a 92.7% chance of survival and 7.3% chance of death for all COVID-19 test participants. This implies that patients are more likely to survive from the virus. The pie chart on the right displays the death distribution given the positive COVID test results and it yeilds the same results as the first pie chart. Given that the patients get infected with coronavirus, they still have a much higher chance to survive even though the survival rate is slightly lower (86.2%) in this case.

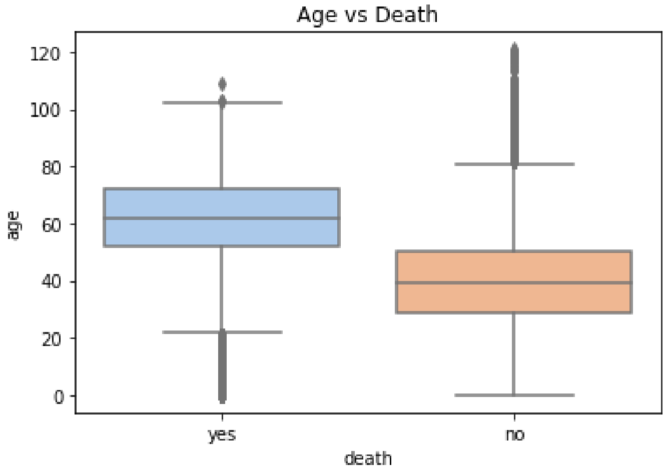


Secondly, we want to take a quick look at the age distribution. This age histogram has a wide range from 0 to 100 years old. It is unimodal with a peak of approximately around 25-30. The majority of age values are distributed between the range of 20 to 60 and there are no outliers detected.

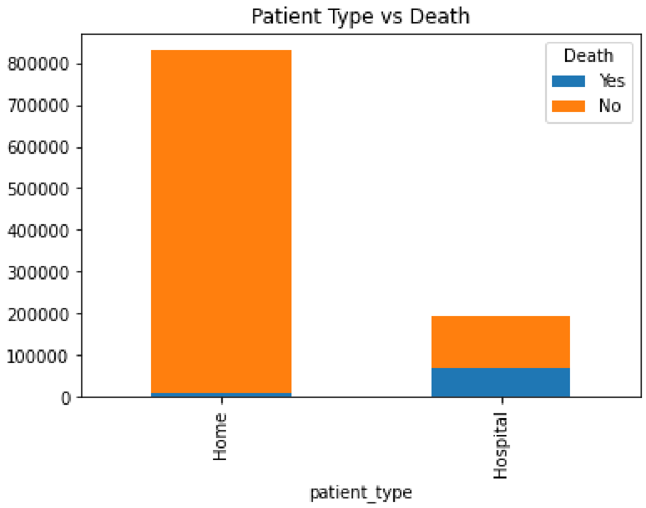




Next, a correlation matrix was created to analyze the relationship between each attribute with death. Among 18 predictors, there are 3 attributes that have the strongest correlation with death which are **patient_type, pneunomia** and **age**. The correlation coefficients are respectively 0.52, 0.47 and 0.32. **Pneumonia** is an attribute that has a relatively strong correlation with death. This does make sense since COVID-19 is a respiratory disease, which enhances the symptoms of pneumonia making the condition much more severe. Regarding **patient_type** and **age**, we will look more into the relationship between them and the response variable **death** using boxplot and barplot.



The boxplot illustrates the relationship between age and death. It indicates that older people are more likely to die from the disease. This can be explained by comparing the mean age between two boxplots. In the blue boxplot, the mean age of patients that died is 60 years old. This is much higher compared to the mean age of survivors in the orange boxplot, which is 40 years old. In addition, the age value for most deaths ranges from 20 to 100 with outliers mostly lying below age 20. On the other hand, the age value for most survivals ranges from 0 to 80 with outliers spreading from age 80 to 120.



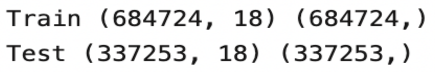
Finally, to illustrate the relationship between the patient type and death, we proceeded to create a barplot with respect to the patient types, namely 'returned home' and 'hospitalization'. The most noticeable characteristic of the plot is that the number of patients returned home outweights the number of hospitalizations. As can be seen in the 'returned home' bar, the number of deaths only makes up a negligible percentage out of all returned home patients. This is reasonable as patients that have mild to moderate symptoms are mostly sent home and can recover without requiring special treatment. Meanwhile, number of deaths in 'hospitalization' bar accounts for more than one third of number of hospitalization. This indicates that hospitalized patients are more likely to die since they have developed severe symptoms which are harder to treat or recover from.

Feature Selection

The correlation chart we saw during the Exploratory Data Analysis gave us a small look at the relationships between all of the attributes. In this section, we will determine the importance of the different variables and how they relate to our response variable: death. A few factors need to be taken into account when deciding what feature selection strategy to employ. Our COVID-19 dataset is almost completely made up of categorical variables. The only quantitative variable, age, can actually be treated as a categorical variable as well because of the size of the dataset. With over 19 variables and over 1 million rows after cleaning, the huge size of the dataset also plays a role on which feature selection method we use.

I. Creating Training and Testing Sets

Before we create our feature selection models, we must create training and testing sets. Splitting the data and using some of it to train the models and the rest to test the models helps to prevent overfitting and improves performance across different datasets. Overfitting occurs when the model learns the training data extremely well but fails to perform on new data. Because of how big the COVID-19 dataset is, we decided to randomly sample one-third of our data for the training set. We did not use replacement, so each data point is unique in the training and testing sets.



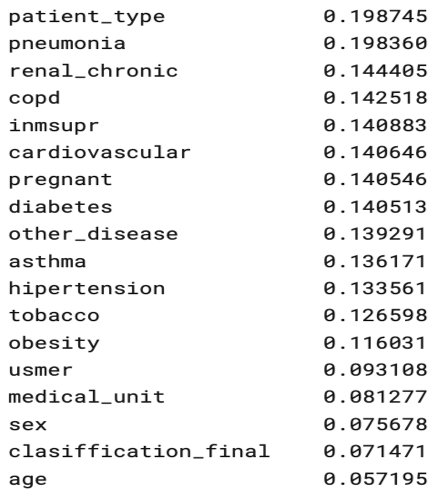
```
Train (684724, 18) (684724,)
Test (337253, 18) (337253,)
```

II. Filter Methods

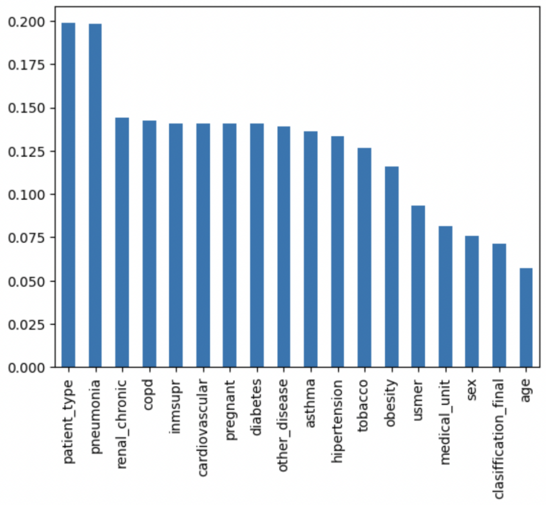
Filter methods use univariate statistics to find the intrinsic properties of features. Filter methods do not involve training models or cross validation, so they are significantly faster and less computationally expensive than some other methods. Because of this, they are often used when dealing with high-dimensional data or extremely large datasets.

1. Mutual Information Feature Selection

Mutual information is the application of information gain to feature selection. Mutual information is calculated between two variables and measures the reduction in uncertainty (entropy) for one variable given a known value of the other variable. The equation is as follows: **MI(feature;target) = Entropy(feature) - Entropy(feature|target).** The MI score will fall in the range from 0 to ∞. The high value of Mi means a closer connection between the feature and the target indicating feature importance for training the model. However, the lower the MI score like 0 indicates a weak connection between the feature and the target



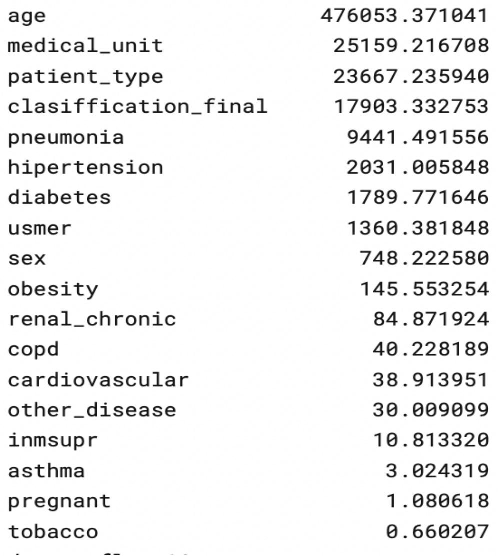
```
patient_type             0.198745
pneumonia                0.198360
renal_chronic            0.144405
copd                     0.142518
inmsupr                  0.140883
cardiovascular           0.140646
pregnant                 0.140546
diabetes                 0.140513
other_disease            0.139291
asthma                   0.136171
hipertension             0.133561
tobacco                  0.126598
obesity                  0.116031
usmer                    0.093108
medical_unit             0.081277
sex                      0.075678
clasiffication_final     0.071471
age                      0.057195
```



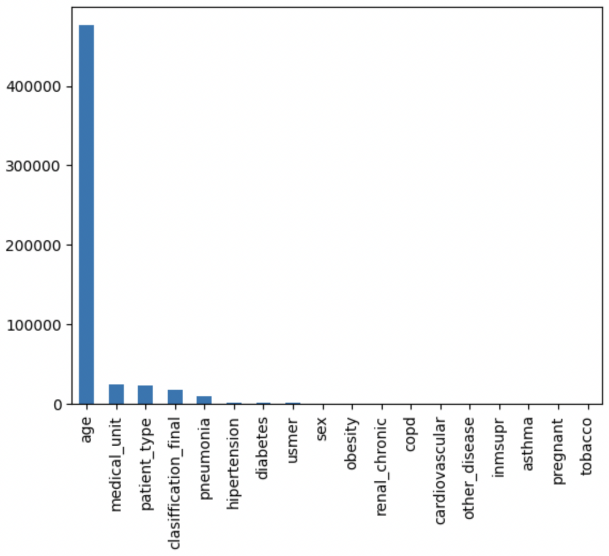
After running the algorithm, the results show that patient_type and pneumonia had the highest MI scores while age and classification_final had the lowest scores. However, I believe that the Mutual Information method is misinterpreting the importance of age because it is counting every age as its own category. We will explore other feature selection methods to see if our assumption is correct.

2. Chi-Squared Feature Selection

A chi-squared test is used in statistics to test the independence of two events. We can apply this test to categorical features in order to determine their dependence. When two features are independent, the observed count is close to the expected count giving us a smaller Chi-Square value. High Chi-Squared values suggest that the feature has a higher dependence on the response and should be selected for model training.



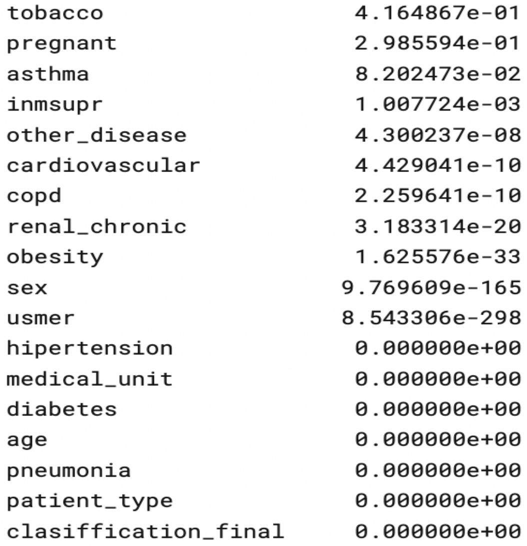
```
age                     476053.371041
medical_unit             25159.216708
patient_type             23667.235940
clasiffication_final     17903.332753
pneumonia                 9441.491556
hipertension              2031.005848
diabetes                  1789.771646
usmer                     1360.381848
sex                        748.222580
obesity                    145.553254
renal_chronic               84.871924
copd                        40.228189
cardiovascular              38.913951
other_disease               30.009099
inmsupr                     10.813320
asthma                       3.024319
pregnant                     1.080618
tobacco                      0.660207
```



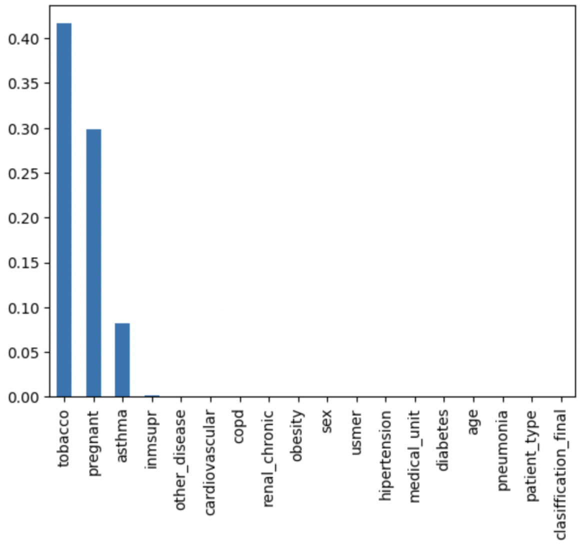
The chi-squared test values that were produced indicate that age and medical unit has the highest significance while tobacco and pregnant had the lowest. Again, I think that the significance of age has been misrepresented by these results. Although I do think age is important, the quantitative nature of the feature is inflating the chi-squared value.

3. P-Values

The P value is derived from the chi-squared value and is essentially the probability of a feature deviating from what was expected purely due to chance. The traditional threshold for significance in a p value is $p < 0{,}05$. However, in extremely large datasets such as this COVID-19 dataset, a small p-value does not necessarily indicate real significance. With so many categorical features compared to the huge number observations, p values will be end up being low regardless of significance.



```
tobacco                    4.164867e-01
pregnant                   2.985594e-01
asthma                     8.202473e-02
inmsupr                    1.007724e-03
other_disease              4.300237e-08
cardiovascular             4.429041e-10
copd                       2.259641e-10
renal_chronic              3.183314e-20
obesity                    1.625576e-33
sex                       9.769609e-165
usmer                     8.543306e-298
hipertension               0.000000e+00
medical_unit               0.000000e+00
diabetes                   0.000000e+00
age                        0.000000e+00
pneumonia                  0.000000e+00
patient_type               0.000000e+00
clasiffication_final       0.000000e+00
```



The bar graph and p values show that while tobacco, pregnant, and asthma come in over .05, the rest of the p values are extremely low. This indicates a high significance in the remaining features. Next we will explore feature selection methods that will better account for the size of the dataset.

III. Wrapper Methods

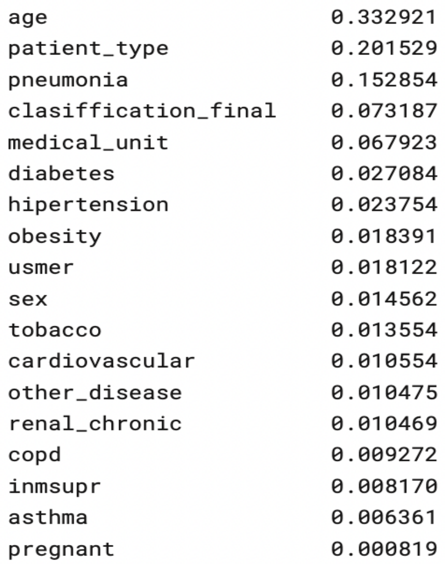
Wrapper feature selection requires some method to search the space of all possible subsets of features. As the subsets are examined, the method assesses their quality and learns and evaluates a classifier with that feature subset. It is considered a greedy search algorithm because it checks every possible feature combination against the evaluation criterion. Wrapper methods normally produce more accurate results than filter methods, but they are considerably more computationally intensive. We attempted to implement Recursive Feature Elimination on our dataset, but the computational requirements of the model were too large because of the size and dimensionality of the dataset.

IV. Embedded Methods

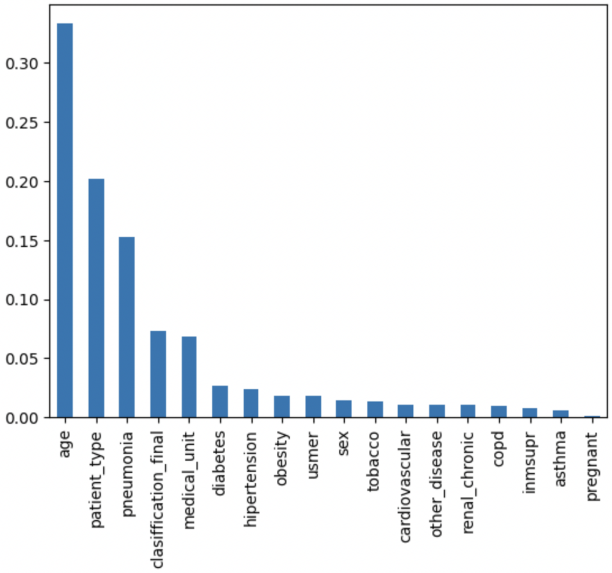
Embedded methods try to incorporate the benefits of both the wrapper and filter methods. They include interactions of features but also maintain reasonable computational costs. Embedded methods tend to give better results than filter methods.

Random Forest Feature Selection

Random Forest Importance Selection is a kind of Bagging Algorithm that aggregates a specified number of decision trees. In bagging, a random sample of data in a training set is selected with replacement, meaning individual data points can be chosen more than one time. Random Forest Importance Selection utilizes this strategy to naturally rank how well each tree improves the purity of the node. Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of the trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features.



```
age                     0.332921
patient_type            0.201529
pneumonia               0.152854
clasiffication_final    0.073187
medical_unit            0.067923
diabetes                0.027084
hipertension            0.023754
obesity                 0.018391
usmer                   0.018122
sex                     0.014562
tobacco                 0.013554
cardiovascular          0.010554
other_disease           0.010475
renal_chronic           0.010469
copd                    0.009272
inmsupr                 0.008170
asthma                  0.006361
pregnant                0.000819
```



The Random Forest Selection Model returned more balanced results and showed that age and patient_type were significant while pregnant and asthma were the least significant.

IIV. Conclusion

Based on the results of the final feature selection we did, we decided to include all of the features except pregnant in our model. This was because pregnant was considered insignificant by all of the feature selection methods except Mutual Information Gain. The filter based models were not very consistent in their rankings of importance, but we believe the Random Forest Importance selection provided a balanced approach between the wrapper and filter methods. Because of the inconsistency, we will most likely test different feature combinations in our prediction model using these findings to determine the optimal combination during future stages of this project.

References

1. Brownlee, J. (2020, August 18). *How to Perform Feature Selection with Categorical Data.* MachineLearningMastery.com. https://machinelearningmastery.com/feature-selection-with-categorical-data/
2. *COVID-19 Dataset.* (2022, November 13). Kaggle. https://www.kaggle.com/datasets/meirnizri/covid19-dataset
3. Gupta, A. (2023, February 15). *Feature Selection Techniques in Machine Learning (Updated 2023).* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/#:~:text=Fisher's%20Score,variables%20as%20per%20the%20case
4. *Seaborn: Statistical data visualization — seaborn 0.12.2 documentation.* (n.d.). https://seaborn.pydata.org/index.html

Contributors

1. Han Hoang: Data Description, Exploratory Data Analysis
2. Carson Dial: Feature Selection
3. Johnny Diep: Data Visualization