CSAI 801 Artificial Intel & Mach Learn W22

COVID-19 Outcome Prediction

Kariman Karm Mohamed Mousaa

February 8, 2022

I. Project overview

COVID-19 is the disease caused by the emerging coronavirus called SARS-CoV-2. There are many symptoms that accompany the disease, but they differ from one person to another according to several factors such as age. The most common symptoms of COVID-19 are:

- Fever
- dry cough
- stress

Other less common symptoms that may affect some patients include:

- loss of taste and smell,
- Nasal congestion,
- Conjunctivitis (also known as red eyes), etc.

What determines whether a person will die from COVID-19 or not, there are many factors such as age or severity of symptoms and whether or not he has a chronic disease or not. It turned out that the death rate among those over 80 years old reached 25%. Doctors discovered that the main cause of death for people infected with the Corona virus is pneumonia that affects the lungs together, which prevents the body from obtaining the necessary amount of oxygen, and as a result, the body's organs stop performing their functions, and thus the death of the infected person.

Problem Statement

We want to Classification COVID-19 Outcome (if the disease will die or not), we found that COVID-19 Outcome according to different factors like location, country, age and many symptoms. This

work will help us to know Covid-19 Outcome (if the disease will die or not) by K-Nearest Neighbors,

Logistic Regression, Naïve Bayes, Decision Trees, Support Vector Machines.

Evaluation Metrics

To evaluate the model, we will use the F1 score (it is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, recall (It calculates the data that we expected to be co rrect out of all the data that we expected to be correct), precision (It determines the percentage of how ac curate the classifier is in its correct predictions) and accuracy (It determines the number of times the class ifier's answer was correct).

.The F1 Formula

$$F_1 = \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$
$$= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

.The Recall Formula

$$\text{Recall} = \frac{TP}{TP + FN}$$

The precision Formula

$$\text{Precision} = \frac{TP}{TP + FP}$$

The accuracy Formula

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

II. Analysis

Data Exploration and Data Preprocessing & Feature Engineering

Before exploring the data, we need to import the basic libraries (numpy, pandas, matplotlib, seaborn, sklearn) and the dataset

1- We show part of data, all information about the data but when we used data.describe() we found there are outliers in **diff_sym_hos** column, we get all this values and drop it.

country	gender	age	vis_wuhan	from_wuhan	symptom1	symptom2	symptom3	symptom4	symptom5	symptom6	diff_sym_hos	result
863.000000	863.000000	863.000000	863.000000	863.000000	863.00000	863.000000	863.000000	863.000000	863.000000	863.000000	863.000000	863.000000
16.995365	0.849363	49.400000	0.181924	0.107764	12.13905	28.002317	18.298957	11.840093	2.993048	0.998841	0.995365	0.125145
7.809951	0.726062	15.079203	0.386005	0.310261	3.99787	7.473231	2.864064	1.183771	0.127251	0.034040	2.358767	0.331075
0.000000	0.000000	2.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	-5.000000	0.000000
11.000000	0.000000	40.000000	0.000000	0.000000	14.00000	31.000000	19.000000	12.000000	3.000000	1.000000	0.000000	0.000000
18.000000	1.000000	49.400000	0.000000	0.000000	14.00000	31.000000	19.000000	12.000000	3.000000	1.000000	0.000000	0.000000
24.000000	1.000000	57.000000	0.000000	0.000000	14.00000	31.000000	19.000000	12.000000	3.000000	1.000000	1.000000	0.000000
33.000000	2.000000	96.000000	1.000000	1.000000	24.00000	31.000000	19.000000	12.000000	3.000000	1.000000	15.000000	1.000000

Data Describe

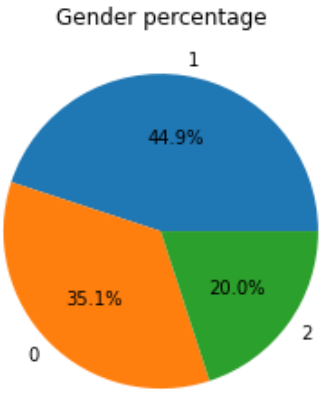
	location	country	gender	age	vis_wuhan	from_wuhan	symptom1	symptom2	symptom3	symptom4	symptom5	symptom6	diff_sym_hos	result
223	56	18	1	35.0	0	0	6	31	19	12	3	1	-1	0
240	128	18	1	65.0	0	0	6	31	19	12	3	1	-5	0

Outlier

```
#remove the outlier
data.drop([223,240],axis = 0, inplace=True)
```

Drop the outlier

2- When we represent gender column, we found, it have 3 genders so we decided to remove, it don't affect the data results.

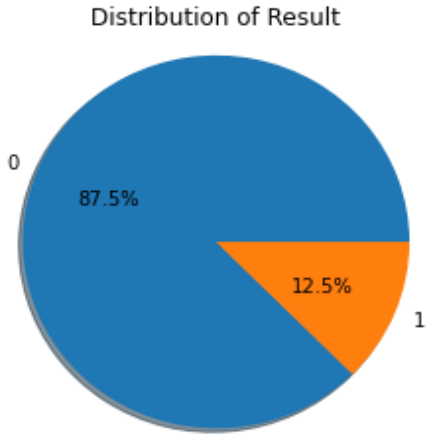


Data columns: become 13 columns. Data rows: from 0 to 861 rows. The Features:

1. Country: where the person resides.
2. Location: which part in the Country.
3. Age: Classification of the age group for each person, based on WHO Age Group Standard.

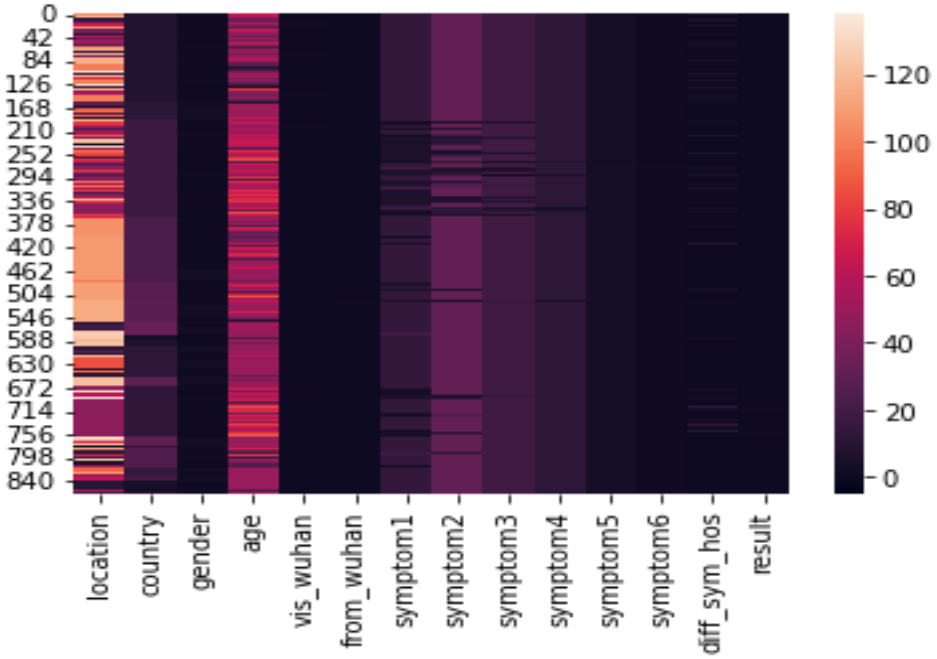
4. Visited_Wuhan: whether the person has visited Wuhan, China or not.
5. From_Wuhan: whether the person is from Wuhan, China or not.
6. Symptoms: there are six families of symptoms that are coded in six fields.
12. Time_before_symptoms_appear.
13. Result: death (1) or recovered (0).

3- We made one hot encoding and merging of the features "location" and "country" because that not numerical, they are categorical.
4- When we represent result column, we found it, the data is imbalance (the number of class (0) greater than class (1) there are many solution to solve this problem but no time but the best metrics for that it will be recall and f1.



Visualization

The heatmap below shows the relationship between all numerical columns, each other and target value.



III. Methodology

Before choosing the algorithms, we first import libraries (we put it in first cell) required and split the data to training and testing then normalizing and selecting data.4
we import from sklearn.model_selection train_test_split to spilt data and sklearn.preprocessing normalize StandardScaler to transform the data.

X_train= data except for the target value" result".
X_valid= sample from test data.
X_test= target value "price".
We split the data to 80 train and 10 validation and 10 test.

IV. Results

Model Evaluation:

1-Benchmark Model: Naïve Bayes

For training

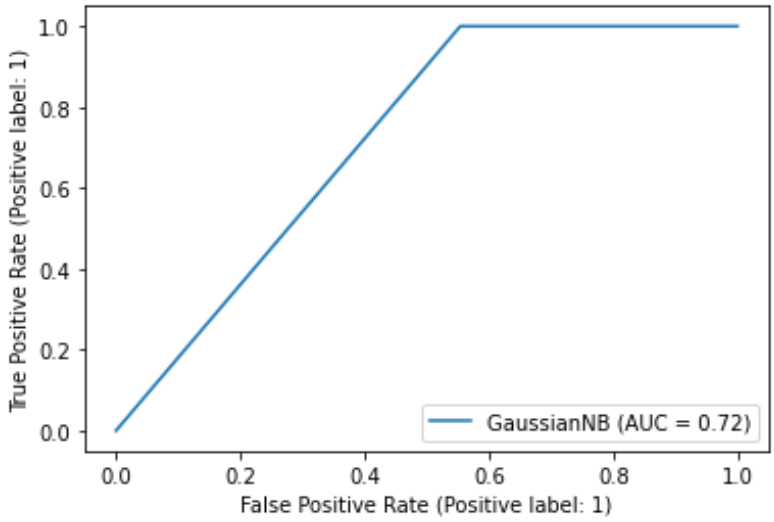
The True Positive is **40** out of **81** and the True Negative is **6** out of **6**, results in an accuracy of 0.53,recall for class(0): 0.49 and class(1):1 and f1: 0.226.

After find the optimal hyper parameters:

The True Positive is **42** out of **81** and the True Negative is **6** out of **6**, results in an accuracy of 0.55, Recall for class (0): 0.52 and class(1):1 and f1: 0.235.

For test

The True Positive is **33** out of **74** and the True Negative is **12** out of **12**, results in an accuracy of 0.52, Recall for class (0): 0.45 and class(1):1, f1: 0.369 and roc_auc_score=0.723.



2- **K-Nearest Neighbors**

For training

The True Positive is **77** out of **81** and the True Negative is **5** out of **6**, results in an accuracy of 0.94, Recall for class (0): 0.95 and class (1): 0.83 and f1: 0.667.

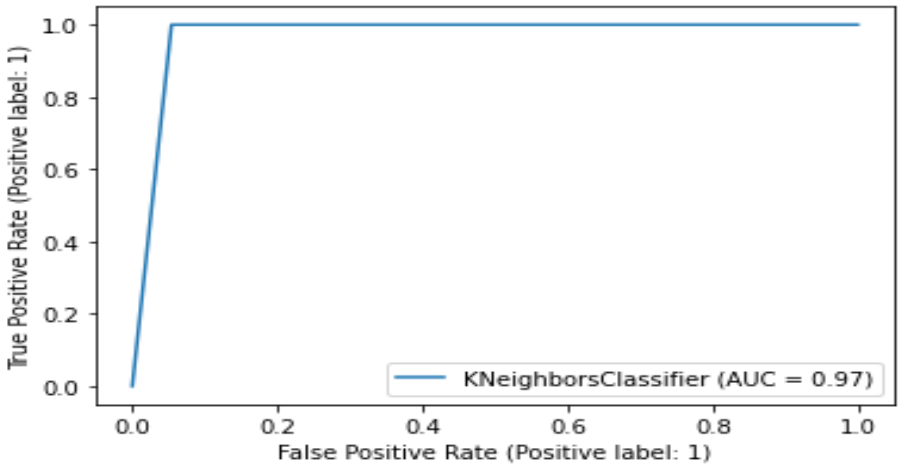
After find the optimal hyper parameters:

The True Positive is **80** out of **81** and the True Negative is **5** out of **6**, results in an accuracy of 0.98,

Recall for class (0): 0.99 and class (1): 0.83 and f1: 0.833.

For test

The True Positive is **70** out of **74** and the True Negative is **12** out of **12**, results in an accuracy of 0.95, Recall for class (0): 0.95 and class (1): 1, f1: 0.857 and roc_auc_score=0.97.



3- Logistic Regression

For training

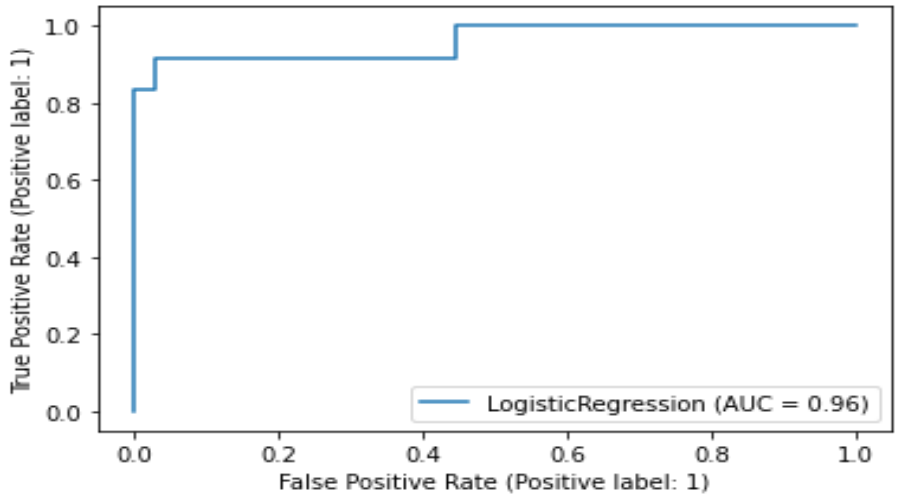
The True Positive is **81** out of **81** and the True Negative is **6** out of **6**, results in an accuracy of 1 ,recall for class(0): 1 and class(1):1 and f1: 1.

After find the optimal hyper parameters:

The True Positive is **78** out of **81** and the True Negative is **6** out of **6**, results in an accuracy of 0.97, Recall for class (0): 0.96 and class (1):1 and f1: 0.8.

For test

The True Positive is **74** out of **74** and the True Negative is **10** out of **12**, results in an accuracy of 0.98, Recall for class (0): 0.97 and class (1): 1, f1: 0.91 and roc_auc_score=0.92.



4- Decision Trees

For training

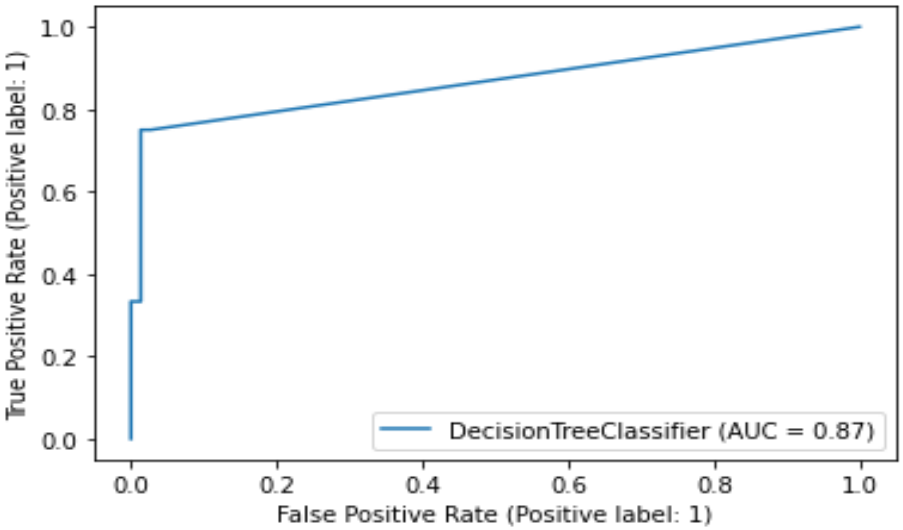
The True Positive is **79** out of **81** and the True Negative is **5** out of **6**, results in an accuracy of 0.97, Recall for class (0): 0.99 and class (1): 0.71 and f1: 0.77.

After find the optimal hyper parameters:

The True Positive is **79** out of **81** and the True Negative is **5** out of **6**, results in an accuracy of 0.97, Recall for class (0): 0.99 and class (1): 0.71 and f1: 0.77.

For test

The True Positive is **73** out of **74** and the True Negative is **8** out of **12**, results in an accuracy of 0.94, Recall for class (0): 0.99 and class (1): 0.67, f1: 0.76 and roc_auc_score= 0.83.



5- Support Vector Machines

For training

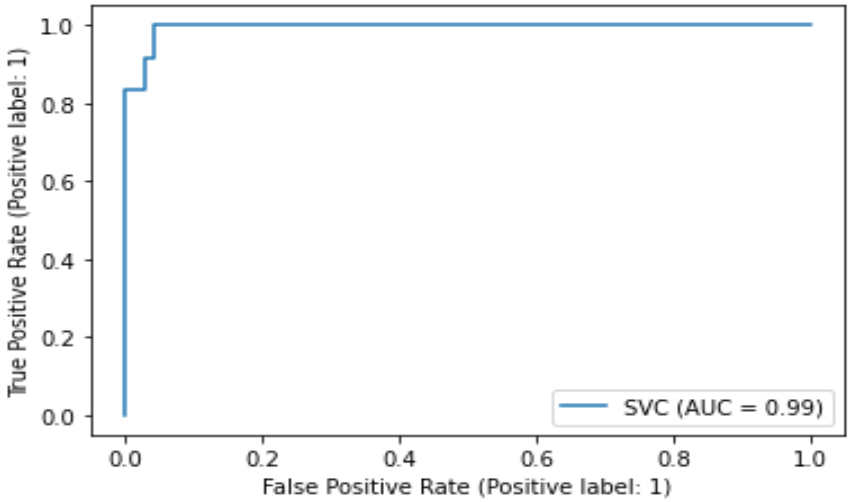
The True Positive is **81** out of **81** and the True Negative is **4** out of **6**, results in an accuracy of 0.98, Recall for class (0):0.98 and class (1):1 and f1: 0.667.

After find the optimal hyper parameters:

The True Positive is **81** out of **81** and the True Negative is **4** out of **6**, results in an accuracy of 0.98, Recall for class (0): 0.98 and class (1): 1 and f1: 0.8.

For test

The True Positive is **74** out of **74** and the True Negative is **6** out of **12**, results in an accuracy of 0.93, Recall for class (0): 0.93 and class (1): 1, f1: 0.67 and roc_auc_score= 0.75.



We can see the highest measure is Logistic Regression then K-Nearest Neighbors, the worse is Naïve Bayes and the worse in speed in training is SVM .

V. Conclusion

This problem was very difficult especially with the imbalance data and not cleaning but the hard part was choosing the model to deal with it and find the optimal hyper parameters , we have tried many different solutions and many different models, but It was very bad speed such as SVM and bad in training and test such as Naïve Bayes. The interested in this project, they were Exploring data and Visualization, that help me more in understanding the nature of data and their problems. Also Pre-processing, it helps the scoring model.

Improvement

To improve this model in the future, we can use the neural network CNN, we can used over sample or more solution to solve imbalance data problem.

References

- **https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall.**
- **https://deepai.org/machine-learning-glossary-and-terms/f-score.**
- **https://en.wikipedia.org/wiki/Accuracy_and_precision.**
- **https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html.**
- **https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.**
- **https://scikit-learn.org/stable/modules/tree.html.**
- **https://scikit-learn.org/stable/modules/naive_bayes.html.**
- **https://scikit-learn.org/stable/modules/svm.html.**
- **https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.**