COVID-19 Prediction Model

Hrayr Gevorgyan, Elise Hoang, Zizhao Guan

1. Introduction

COVID-19 has been the center of attention for the past few years and most people have felt helpless as a result. We are all relying on the health sector to get us through the pandemic and want to do our best to help. This, along with our general interest in machine learning, led us to try to use our skills and create a model to predict whether or not individuals have COVID-19 or not. By analyzing datasets of symptoms and training a model, we hope to be able to create a platform in which given symptoms from the user will predict whether or not the individual is sick. This can be used practically as a first step to detection. Instead of putting a strain on resources by using testing kits, people could get the likelihood of infection from our tool and make a decision to do further testing given the results. In addition, through this project, we wish to establish and develop our ability to apply machine learning to predict more diseases that are hard to diagnose by traditional means.

In this report, we are going to use five sections to complete. (1)The first section is the introduction which is talking about the problem we are trying to solve and the purpose of doing this project. (2)The second section is the method and materials. For the methods, we are going to talk about the machine learning techniques applied in this project. The machine learning techniques included in our project are Linear SVC, Decision Tree, PCA, Adaboost, and Random Forest. And we also used a confusion matrix, heatmap, and seaborn to visualize our study. For the material section, we are going to credit and introduce the data we found. Also, in this section, we will show the relationship between the covid test results and the symptoms in the data. (3)The third section is the results. In this section, we are about to showcase the results we got. (4)The fourth section is the discussion, we will discuss the results here. (5)The last one is the appendix that includes the figures and the references of the data.

2. Methods and Materials

2.1 Materials

2.1.1 Coronavirus Disease 2019 Clinical Data(Carbon Health and Braid Health, 2019)

This data is obtained from the repository(https://covidclinicaldata.org/). The CSV files are on the GitHub repository(https://github.com/mdcollab/covidclinicaldata/tree/master/data/). It contains 83 rows of features. These features include information about the patients, the test results, and some data on symptoms. It has 93,995 samples. The data is reliable since the authors request it from the test centers and healthcare facilities. For this data, we call it data_1 here.

2.1.2 Symptoms for COVID-19 Prediction Model(nshomron and Zoabi, 2020)

This data can be found on the GitHub repository(https://github.com/nshomron/covidpred). It's a relatively large dataset with 136,294 rows but it has fewer features(symptoms). The authors downloaded and translated the data from the repository(https://data.gov.il/dataset/covid-19). We used the file "corona_tested_individuals_ver_006.english.csv.zip" in the data folder from the GitHub repository for training and prediction purposes. For this data, we call it data_2 here.

2.2 Methods

2.2.1 Features(columns/symptoms) Selection and Dataframe Cleanup

To get the data_1, we need to concatenate several CSV files and reset the index. Since this data has 83 columns but most of them are not measurable or less relevant to the covid test, we decided to drop the columns down to eleven. The remaining columns and the description are shown in Table 1. And we drop the rows with NaN. The result DataFrame is shown in Table 2.

Then we used the bar plot to display the relationship between test results and every feature(Boolean type) that is needed to calculate the number of samples. We also used the catplot to visualize the relationship between the test results and the features with Float/Int value.

For the data_2, we only chose the corona_tested_individuals_ver_006.english.csv to analyze. We kept most of the columns and dropped the columns such as test date, gender, and test indication. The data types and column descriptions are shown in Table 3. And Tabel 4 shows the remaining Dataframe.

2.2 Using Machine Learning Technique to Train and Predict

To find some better models, we first trained models on the training set from data_1 with Decision Tree and Linear SVC and built the models. Then, we used the models to predict the test set with these tree techniques. We also used PCA to reduce the dimensions of the data and use a Logistic Regression model to predict. to find the correlation between the features, We used a Heatmap. And we used the Confusion Matrix to check the accuracy of the Decision Tree model.

For data_2, besides the methods we used for data_1, we used AdaBoost and Random Forest additionally.

2.2.3 Adjusting the Parameters

In this project, to gain a better result and model, we adjusted the parameters such as the size of the samples and the number of components. We adjusted the size of data_2 from 10,000 to 120,000. And in the PCA training, we adjusted the n_components from 2 to 5.

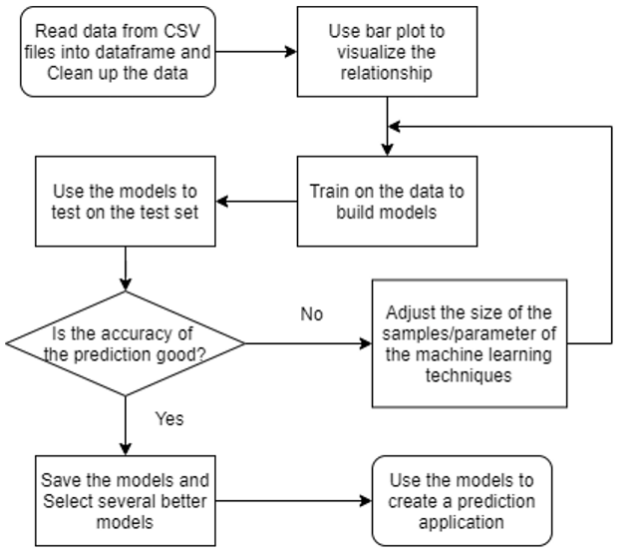
2.2.4 Building an Application for Covid19 Prediction

After we performed the experiments, we used the models that show better predictions to construct an application. Users can use this application to input their information and symptoms then get the result.

Pseudo code for application:

```
Array Symptom_info = INPUT general symptom information
String Model = INPUT which model to use
Dictionary models = <map of model names to generated model pickle files>
IF model needs a dataframe:
      Input = dataframe(Symptom_info)
ELSE IF model needs an array:
      Input = Symptom_info
Loaded_model = OPEN models[model]
print(Loaded_model.predict(input))
```

2.2.5 Flow Diagram



3. Results - you can include screenshots along descriptions.

3.1 The Relationship between Features and Test Results for Data_1

Since data_1 only has a few positive cases, we only can show better results. (See results at Figure 1)

3.2 The Relationship between Features and Test Results for Data_2 (See results at Figure 2)

3.3 The Best Prediction of PCA and Logistic Regression for Data_1 and Data_2

For Data_1 with Size of Negative Cases =31761 (Positive Cases =654):

	precision	recall	f1-score	support
Negative	0.98	1.00	0.99	15894
Positive	0.33	0.01	0.01	314
accuracy			0.98	16208
macro avg	0.66	0.50	0.50	16208
weighted avg	0.97	0.98	0.97	16208

For data_2 with Size of Negative Cases =10626 (Positive Cases =10626):

	precision	recall	f1-score	support
False	0.68	0.92	0.78	2596
True	0.88	0.58	0.70	2717
accuracy			0.75	5313
macro avg	0.78	0.75	0.74	5313
weighted avg	0.78	0.75	0.74	5313

The scores for negative and positive cases of data_2 can check Figure 3.

3.4 The Best Prediction of Adaboost for Data_2 with Size of Negative Cases =10626 (Positive Cases =10626):

	precision	recall	f1-score	support
False	0.68	0.92	0.78	2596
True	0.88	0.58	0.70	2717
accuracy			0.75	5313
macro avg	0.78	0.75	0.74	5313
weighted avg	0.78	0.75	0.74	5313

3.5 The Best Prediction of Decision Tree for Data_1 and Data_2

For Data_1 with Size of Negative Cases =31761 (Positive Cases =654):

	precision	recall	f1-score	support
False	0.98	0.97	0.97	4683
True	0.08	0.12	0.10	107
accuracy			0.95	4790
macro avg	0.53	0.54	0.54	4790
weighted avg	0.96	0.95	0.95	4790

For data_2 with Size of Negative Cases =10626 (Positive Cases =10626):

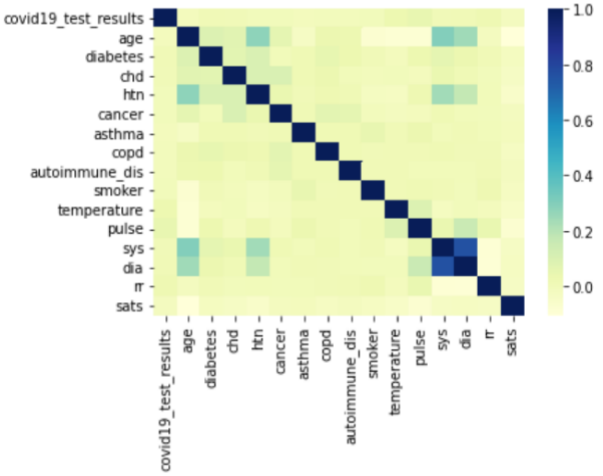
	precision	recall	f1-score	support
False	0.74	0.82	0.78	2142
True	0.80	0.70	0.74	2109
accuracy			0.76	4251
macro avg	0.77	0.76	0.76	4251
weighted avg	0.77	0.76	0.76	4251

3.6 The Best Prediction of Random Forest for Data_2 with Size of Negative Cases =10626 (Positive Cases =10626):

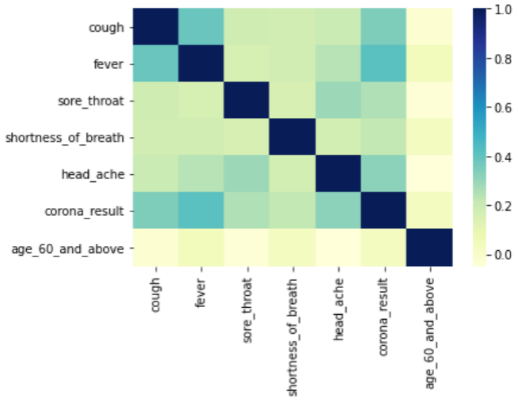
	precision	recall	f1-score	support
False	0.72	0.83	0.77	2596
True	0.81	0.69	0.75	2717
accuracy			0.76	5313
macro avg	0.77	0.76	0.76	5313
weighted avg	0.77	0.76	0.76	5313

3.7 Heatmap (using Spearman Correlation) for Data_1 and Data_2↩(back to previous)

For Data_1 with Size of Negative Cases =31761 (Positive Cases =654):

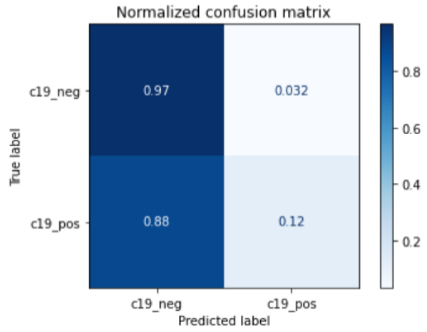


For data_2 with Size of Negative Cases =10626 (Positive Cases =10626):

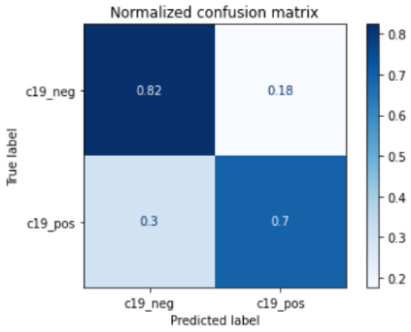


3.8 Confusion Matrix for Data_1 and Data_2 using Decision Tree

For Data_1 with Size of Negative Cases =31761 (Positive Cases =654):



For data_2 with Size of Negative Cases =10626 (Positive Cases =10626):



3.9 The Predictions for Changing the Parameters for Data_2 (See results at Table 5)

3.10 Performance of Changing n_components of PCA

n_components	Overall Accuracy	False Precision	True Precision	Recall(True positive rate)
2	0.73	0.68	0.80	0.62
3	0.73	0.68	0.79	0.63
4	0.74	0.68	0.87	0.58
5	0.75	0.68	0.89	0.58

4. Discussion

The biggest challenge we faced in completing this project had to do with the plain fact that it is not easy to know if someone is carrying the COVID-19 virus. If it were trivial, the spread of the virus wouldn't be so hard to contain. As people can carry the virus without showing symptoms, or can have all the symptoms but be sick with some other virus, it is quite hard to predict if someone is carrying the virus with such few features. However, the perfect dataset, one with many features and balanced amounts of positive and negative cases, doesn't seem to exist. Because of this constraint we had to choose between a dataset with many features but very few COVID-19 positive records (data_1) , or a dataset with fewer features but many more COVID-19 positive records (data_2) . Though we tried with the data_1, our models would always predict negative because they would be right the vast majority of the time. Therefore, we had to use data_2.

From Heatmap, we can see that if some symptoms are true or false, the likelihood that the person has COVID-19 or not are either very likely, or very unlikely. Though this works well for the dataset that we're using, we suspect that the real life application of our models may be skewed as a result. When using our models to carry out predictions, we can easily see that if some symptoms are true, our models will always predict a certain result. This is not very desirable and could be remedied by using a much larger and better balanced dataset with many more features. Now we just have to wait for such a perfect dataset to come into existence!

From the summary (Table 5), we can see that when we were increasing the size of negative cases, the accuracies and the precisions were increasing. Some can reach more than 90%. However, the recalls (true positive rates) were decreasing. The PCA one had only 1% true positive rate. This problem may be caused by the bias of data after we increased the size of negative cases. The models may classify almost all the samples (including positive cases) as negative. Therefore, we had higher precisions but lower true positive rates (true positive rate = true positive/(true positive + false negative)). We can imagine that we put a line of threshold=-10 on the Figure 3 so that the samples (score>-10) are classified as negative, and samples (score<=-10) are classified as positive. Therefore, in the end, we use the 10626 as the size of the negative cases to train the models for our application.

5. Appendix

Table 1. The Remaining Features and Description for Data_1((Carbon Health and Braid Health, 2019) ↩(back to previous)

Name	Type	Description
covid19_test_results	String (Positive, Negative)	Results of covid19 test
high_risk_exposure_occupation	Boolean	Whether patient in a profession with a high risk of exposure.
cough	Boolean	Whether the patient has cough symptoms.
fever	Boolean	Whether the patient has a fever.
fatigue	Boolean	Whether the patient has fatigue.
headache	Boolean	Whether the patient has a headache.
muscle_sore	Boolean	Whether the patient has sore muscles.
sore_throat	Boolean	Whether the patient has a sore throat.
temperature	Float	The temperature of the patient
rr	Int	Respiratory rate measured in breaths per minute.
sats	Int	Oxygen saturation of the patient

Table 2. Part of the Dataframe for Data_1 ↩(back to previous)

	covid19_test_results	high_risk_exposure_occupation	cough	fever	fatigue	headache	muscle_sore	sore_throat	temperature	rr	sats
0	Negative	True	False	True	True	True	True	False	37.10	15.0	98.0
1	Negative	True	False	False	True	True	False	False	37.15	17.0	100.0
2	Negative	False	True	True	True	True	False	True	37.15	17.0	100.0
3	Negative	False	True	False	False	False	False	False	36.90	16.0	97.0
4	Negative	False	False	False	True	False	False	False	36.50	14.0	97.0

Table 3. The Remaining Features and Description for Data_1(nshomron and Zoabi, 2020) ↩(back to previous)

Name	Type	Description
corona_result	Boolean	Results of covid19 test
cough	Int	Whether the patient has cough symptoms (False = 0 / True = 1)
fever	Int	Whether the patient has a fever. (False = 0 / True = 1)
sore_throat	Int	Whether the patient has a sore throat. (False = 0 / True = 1)
shortness_of_breath	Int	Whether the patient has shortness of breath. (False = 0 / True = 1)
head_ache	Int	Whether the patient has a headache. (False = 0 / True = 1)
age	Int	Whether the patient is over 60 years old (False = 0 / True = 1)

Tabel 4. Part of the Dataframe for Data_2 ↩(back to previous)

	cough	fever	sore_throat	shortness_of_breath	head_ache	corona_result	age_60_and_above
122808	1.0	0.0	0.0	0.0	0.0	False	1
122809	1.0	0.0	0.0	0.0	0.0	True	0
122810	0.0	0.0	0.0	0.0	0.0	False	0
122811	0.0	1.0	0.0	0.0	0.0	False	0
122812	1.0	0.0	0.0	0.0	0.0	False	1

Figure 1. The Relationship between Features and Test Results for Data_1. ↩(back to previous)

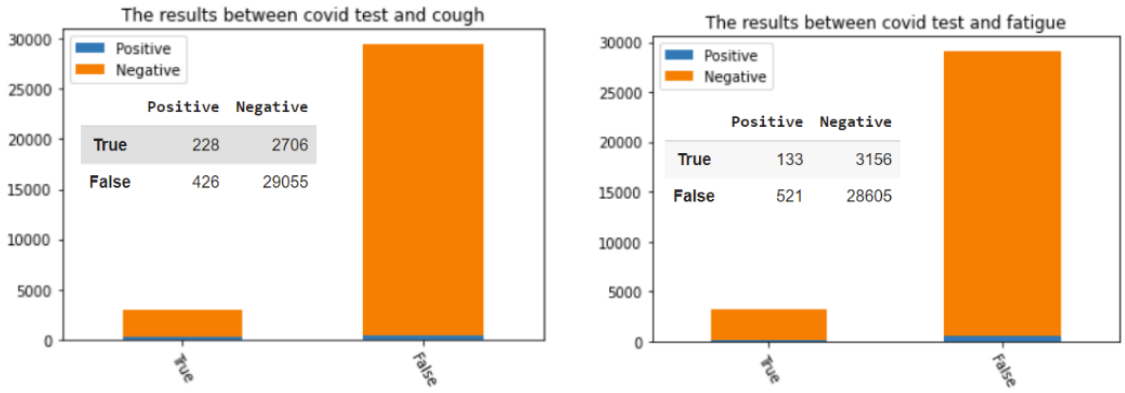


Figure 2. The Relationship between Features and Test Results for Data_2. ↩(back to previous)

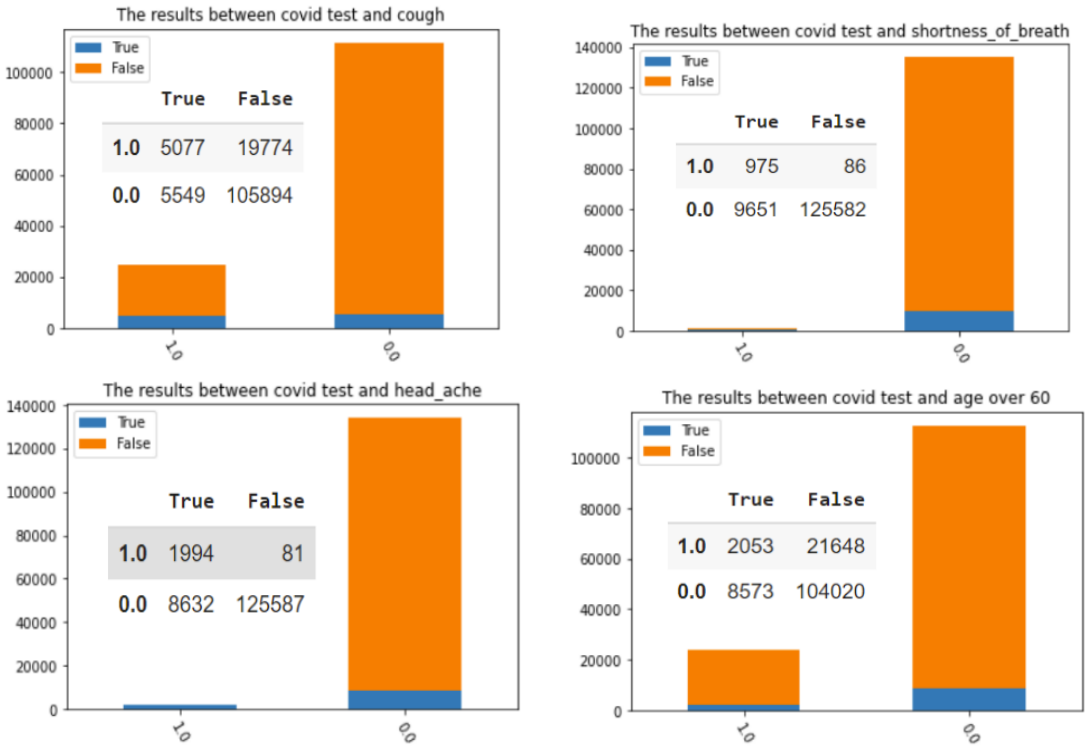


Figure 3. Scores Distribution for Negative and Positive Cases ↩(back to previous)

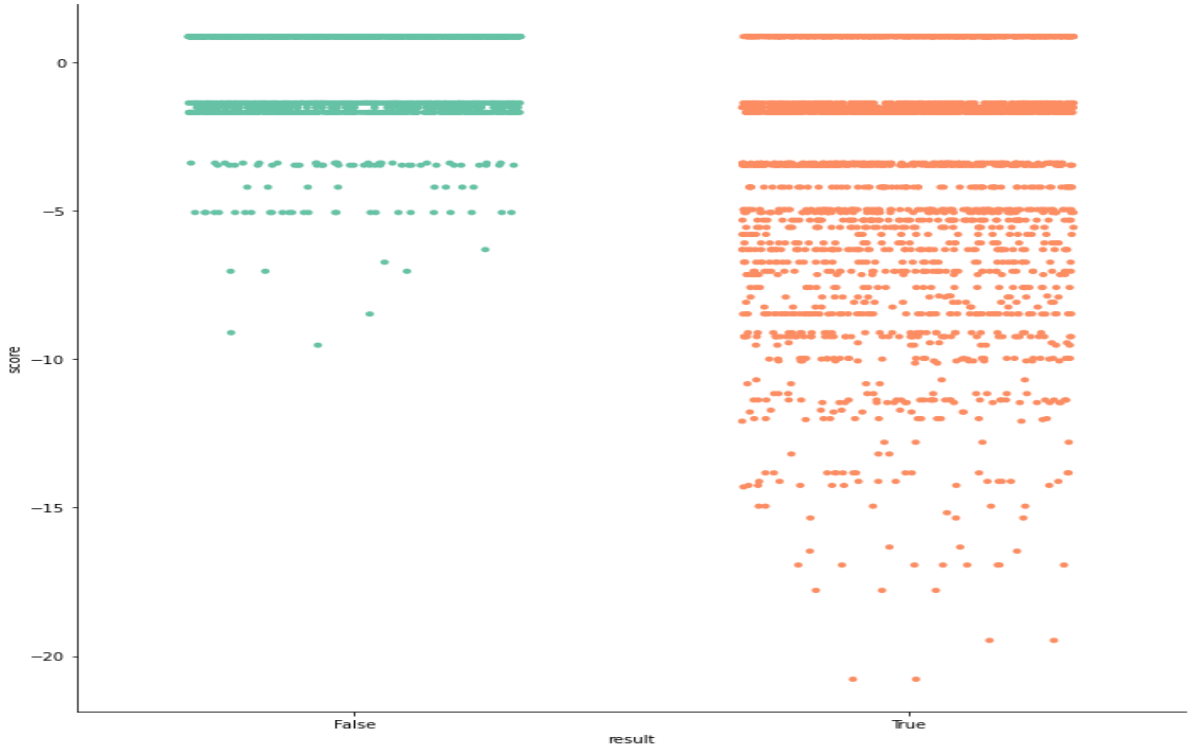


Table 5. The Accuracy for Changing the Parameters for Data_2 ↩(to result) ↩(to discussion)

Method	Size of Negative Cases	Overall Accuracy	False Precision	True Precision	Recall(True positive rate)
PCA and Logistic Regression	10626	0.73	0.68	0.80	0.63
	50000	0.84	0.86	0.55	0.30
	90000	0.91	0.91	0.70	0.15
	125668	0.92	0.92	0.94	0.01
Adaboost	10626	0.75	0.68	0.89	0.56
	50000	0.88	0.87	0.90	0.31
	90000	0.92	0.92	0.97	0.27
	125668	0.94	0.94	0.95	0.25
Decision Tree	10626	0.76	0.74	0.80	0.70
	50000	0.88	0.89	0.81	0.46
	90000	0.93	0.93	0.90	0.34
	125668	0.95	0.95	0.88	0.34
Random Forest	10626	0.76	0.72	0.82	0.69
	50000	0.89	0.89	0.82	0.45
	90000	0.93	0.93	0.91	0.34
	125668	0.94	0.95	0.88	0.34
Confusion Matrix	10626		0.83	0.7	
	50000		0.98	0.46	
	90000		1	0.34	
	125668		1	0.34	

Instructions to run code:

Demo video in case you get stuck: https://www.youtube.com/watch?v=2H1xf9p_tro

1. **To train the data and save the models.**
    a. Open the file named ”analyzing_data_2_and_saving_models.ipynb”.
    b. Run the code. It will read the CSV file from GitHub. The data will be data_2 “corona_tested_individuals_ver_006.english.csv” by default.
    c. Input the size of negative cases. The size should be from 10000 to 120000. If nothing is entered, we will default to what we think is the best value (10626).
    d. The models will be created and saved in the local directory.
    e. It should contain six .pkl files, one for each of the created models.

2. **To run the covid19 prediction application**
    a. Open the file named “using_model_to_predict.ipynb”
    b. Run the code. It will ask for user input. Answer the questions using ‘1’ as yes and ‘0’ as no. Answer the last question by entering the name of the model you want to use to predict.
    c. The chosen model will make a prediction and output it.

References

Carbon Health and Braid Health.(2020).Coronavirus Disease 2019(COVID-19)Clinical Data Repository(10-20-2020).Accessed from https://covidclinicaldata.org/

Zoabi,Yazeed.(2020).A COVID-19 Prediction Model Using Symptoms. Accessed from https://github.com/nshomron/covidpred