Death/Recovery prediction for Covid-19 Patients using Machine learning

Omar Mohamed Atef Mohamed
Department of Electrical and Computer Engineering
University of Sharjah
U16104886@sharjah.ac.ae

Ali Bou Nassif
Department of Electrical and Computer Engineering
University of Sharjah
anassif@sharjah.ac.ae

Maha Alaa Eddin
Department of Electrical and Computer Engineering
University of Sharjah
malaaeddin@sharjah.ac.ae

Manar Wasif Abu Talib
Department of Computer science
University of Sharjah
mtalib@sharjah.ac.ae

Qassim Nassir
Department of Electrical and Computer engineering
University of Sharjah
nasir@sharjah.ac.ae

***Abstract*—Covid19 is a newly discovered corona virus that has been officially announced as a pandemic by the World Health Organization in March 2020. It is a new virus in the medical field that has no specific treatment and no vaccines till this moment. In addition, they have not discovered all the symptoms but only some of them. Covid19 is spreading very fast as the medical systems over the world are not able to hospitalize all the patients which lead into a significant increase in the number of the virus death. This work will use the power of machine learning using Python3 to predict which patient has higher probability of death according to a dataset on Kaggle which has over 10k entries and using 9 attributes. 3 different algorithms multilayer perceptron, support vector machine, K nearest neighbor with splitting the data at 70% and 30% were used in this work. The accuracies achieved was between 92% to 100% with MLP, SVM and KNN. SVM achieved the highest accuracy. The models evaluated through precision, accuracy, recall and f measure.**

Keywords—Corona virus, Covid19, Multilayer perceptron, support vector machine, K nearest neighbor.

I. INTRODUCTION

Covid-19 deaths now are more than 200K people and more than 3.5M case around the world [1]. The main reason of that is because the doctors are not able to hospitalize all the patients, so they mostly choose the patients that has higher probability of surviving. Many people think that only old people have higher probability of death, but this is not always the case. the higher probability of risk depends on several factors such as the symptoms, age, dates of symptoms, date of hospitalization and date of confirmation of covid-19.

In this work. We are utilizing Machine learning to predict which patients have higher priority for hospitalization as they have higher probability of death. We are building some classification algorithms to predict that, and we will evaluate the performance through some evaluation metrics.

The "Novel Corona Virus 2019 Dataset" available on Kaggle has over 10k of data from different patients in different countries with different attributes. The main goal of this work is to predict the probability of death, recovery, stable or severe. Using 3 algorithms with different parameters.

The dataset has many data missing, dealing with the missing data took us a lot of time trying to use different techniques.

II. RELATED WORK

After we inspect about the latest work done in Covid19 and Artificial intelligence, we found that Predicting Death/Recovery rate was not done before. Despite having many researches on covid19 detection using chest CT scan.

In [2], They developed an automatic framework to detect Covid19 from the CT scan of the patient's chest. They used deep learning model, it was a Covid19 detection neural network. They used a dataset contains 4352 chest CT scans from 3322 patients. Their model achieved accuracy higher than 90%.

III. TECHNICAL BACKGROUND

A. *Corona Virus Covid-19*

Covid-19 is a new corona virus. It was firstly discovered in China in December 2019 and then it spreads all over the world. The cause of covid-19 is still not clear until now.

Coivd19 has no specific treatment or vaccines but in hospitals they are trying to treat only the symptoms. The virus infects the respiratory system and in the critical cases the virus can damage the lungs and the patient dies. In 11 March 2020 World Health Organization "WHO" has characterized Covid19 as pandemic [3].

Covid19 symptoms are not yet all known but some common symptoms are like normal flu symptoms e.g. fever, headache, fatigue, etc.

B. *Machine Learning*

Machine learning is a subset of Artificial intelligence that works on developing systems that can be improved through learning experience [4].

It is working by training the mathematical models on the training set and as a result it gains experience and can predict and take decisions with being explicitly programmed.

It can be divided into supervised and non-supervised learning. Supervised is when you train the model on a specific dataset giving him the input and the expected outputs. Through some mathematical process it will be able to adjust some parameters to be able to predict and take decisions in the testing phase. 2 types of supervised learning which are regression problems where the output is numeric e.g. temperature or salary. While the other type is classification where it classifies different classes e.g. low, medium, and high.

Non-supervised learning works in opposite way. The models are not given the expected output, so it is used in other applications such as clustering.

C. *Articifial Neural Network*

Artificial Neural Network "ANN" is a supervised machine learning algorithm that used mainly for classification problems as a classifier.

The simplest form is composed of input layer and output layer. This model is called single perceptron and it is used for easy classification problems.

Neural Networks mostly used architecture is composed of 3 parts and it is called Multilayer perceptron "MLP". Input, hidden and output layers. It operates by some process that mimics the way of how human brain works. The input layer contains the inputs of the system, these inputs will be multiplied by the weights which are randomly initially and then they will be adjusted in the training stage. The hidden layer takes the results which are the inputs multiplied by the weights. In the hidden layer there are several neurons each neuron contains a function e.g. sigmoid, tanh, etc. then the output of the hidden layers will be multiplied by some weights then they will be the input to the output layer. The output layer maps the result to the closest class according to the activation function in it.

An example of MLP Neural Networks is shown in Fig.1.

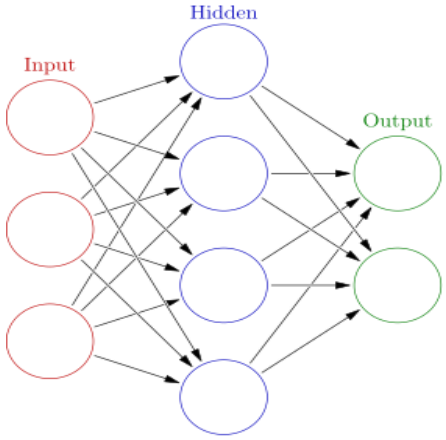

Figure 1: Neural Network Architecture

D. *Support vector machine*

Support vector machine "SVM" is another supervised machine learning algorithm that mainly used in classifications and can also be used in regression through some modifications.

The mechanism of SVMs is by separating the classes from each other using hyperplanes. The classes are represented as data points that are n dimensional feature vector and the hyperplane has a geometric shape that occupies n-1 dimensions [4].

The simplest form of the SVMs is where the data is linearly separable. Where 2 parallel hyperplanes are used to separate the classes such that the distance between the hyperplanes is maximum to minimize the error of classification. The observation that lies on the hyperplanes is known as support vectors. In Fig. 2 showing an example of SVM Hyperplane.

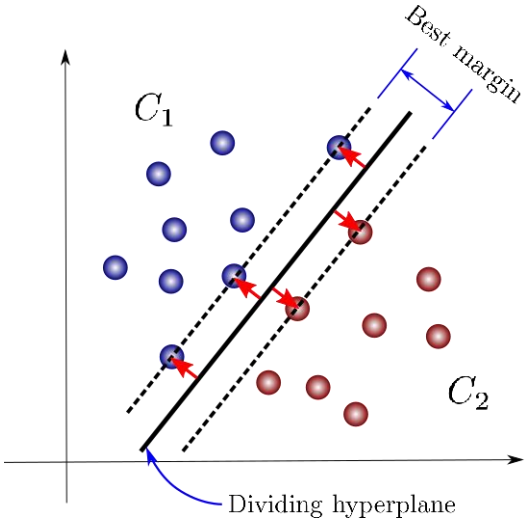

Figure 2: SVM Hyperplane

E. *K-Nearest Neighbours*

K nearest neighbour "KNN" is a supervised machine learning algorithm that can be used in classification and regression. It classifies new cases based on the similarity measure e.g. distance functions.

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results. In Fig.3 showing K nearest neighbour algorithm and how it works according to the value of K [5].

Identify applicable funding agency here. If none, delete this text box.

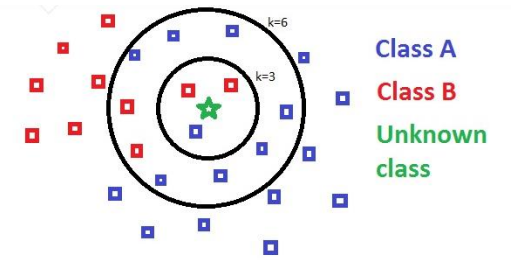

Figure 3: KNN

IV. DATASET

Data pre-processing and filtering is the most consuming time stage in any data science project.

We used a dataset from Kaggle which is "Novel Corona Virus 2019 Dataset", the data has over 10k of entries, but it contains many missing values and many non-necessary attributes. Dealing with this data took a lot of time.

To begin with, we removed ID of each patient, the city were the patient lives, longitude and latitude of his city, all travel history data, death or discharge and some other references columns. We leaved the most important attributes for our system which are age, gender, symptoms, date on set symptoms, date hospitalization, date confirmation and outcome. The outcome is the dependent variable "outputs" that contains 4 classes which are "Discharge", "Death", "Stable", "Severe" and the rest of the attributes are independent variables "inputs".

Secondly all the data converted to numeric data to be easy to handle missing data using different techniques for instance: mean, median and mode. We used to change the gender into 1 for female and 2 for male then using the imputer function in Sklearn in Python3 using PyCharm we used to fill the missing gender data with the mean of the available data and take the ceil of the results to make them either 1 or 2. Also, another imputer used the mean to fill the missing age data. Then for the missing dates, mean or median method will not work so all missing dates were removed from the data and unfortunately the data has been reduced a lot to be only 256 rows. Then we extracted from these dates the difference between the days using DATEDIF function in Excel.

Finally, symptoms column contains for each patient several symptoms or no symptoms. Each symptom has been extracted to be in 1 column individually and the attribute will be 1 is true symptom or 0 false symptom e.g. column fever for patient 2 is 1 indicating that he has fever. Symptoms used in this work are 5 symptoms which are fever, malaise, chills, cough, fatigue.

A. *Features*

1. Age: age of the patient.
2. Sex:
    - 1: Female.
    - 2: Male
3. Fever:
    - 1: Positive
    - 0: Negative
4. Malaise:
    - 1: Positive
    - 0: Negative
5. Chills:
    - 1: Positive
    - 0: Negative
6. Cough:
    - 1: Positive
    - 0: Negative
7. Fatigue:
    - 1: Positive
    - 0: Negative
8. Syms hosp: days between symptoms appearance and hospital admission.
9. Syms conf: days between symptoms appearance and confirmation of covid19.
10. Outcome: dependent attribute consists of 4 classes:
    - Discharge: Patient is recovered.
    - Death: Patient Died
    - Stable: Patient is in stable condition.
    - Severe: patient in critical condition.

The output data are distributed as shown in Fig. 4 the classes are not equally distributed most of the data are discharge and stable while death and severe are small.

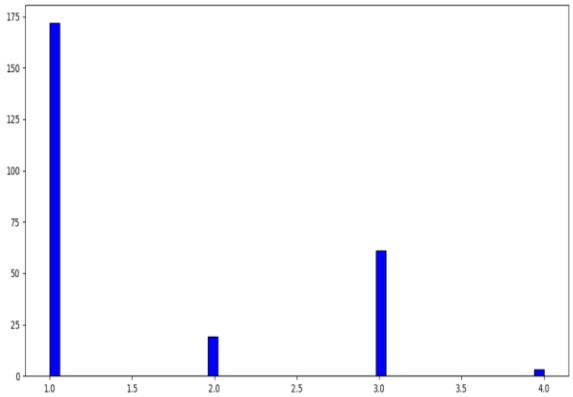

Figure 4: Outcome data

V. METHODOLGY

A. Feature selection

Since we do not have too many attributes and the data are not large enough after filtering. We will have to remove any redundant variables to improve training and reduce overfitting.

In this paper feature selection was done through Python using sklearn to get the correlation between the variables. A CSV file was generated that contain the correlation results between the variables.

Fig.5 is showing the correlation between the variables.

	age	sex	outcome	fever	malaise	headache	chills	cough	fatigue	syms_hosp	syms_conf
age	1	-0.0063	0.273058	0.009971	-0.02842	0.023482	0.009038	-0.0106	0.118847	0.10963373	0.19901924
sex	-0.0063	1	0.007838	-0.03812	0.05553	-0.1127	-0.09312	0.050502	-0.23992	0.05666018	-0.0606727
outcome	0.273058	0.007838	1	-0.19939	-0.04149	-0.11901	0.050789	-0.09719	-0.07531	-0.2172752	0.03800864
fever	0.009971	-0.03812	-0.19939	1	-0.03784	0.298444	0.278605	0.506934	0.221155	0.0493542	0.04076844
malaise	-0.02842	0.05553	-0.04149	-0.03784	1	-0.01129	-0.01054	0.152304	-0.01332	0.04119694	0.01815539
headache	0.023482	-0.1127	-0.11901	0.298444	-0.01129	1	-0.03024	0.053603	0.293971	-0.1627585	-0.0202903
chills	0.009038	-0.09312	0.050789	0.278605	-0.01054	-0.03024	1	0.407803	0.200596	0.10475728	-0.009291
cough	-0.0106	0.050502	-0.09719	0.506934	0.152304	0.053603	0.407803	1	0.022137	0.04647882	0.06771842
fatigue	0.118847	-0.23992	-0.07531	0.221155	-0.01332	0.293971	0.200596	0.022137	1	-0.0614586	0.16233318
syms_hosp	0.109634	0.05666	-0.21728	0.049354	0.041197	-0.16276	0.104757	0.046479	-0.06146	1	0.78945674
syms_conf	0.199019	-0.06067	0.038009	0.040768	0.018155	-0.02029	-0.00929	0.067718	0.162333	0.78945674	1

Figure 5: Correlation between variables

There is high correlation between the fever and cough. In addition, highly correlation between symptoms to hospital and symptoms confirmation.

B. Model Design

In this work Python3 was used to build 3 machine learning models and to test them on the data. Pandas library were used to read the data and to filter it, while Sklearn were used to build the models and to evaluate the results. The data was split randomly into 70% training and 30% testing.

In table.1 shown the 3 models with the best parameters for each of them after simulating these parameters in Python using PyCharm.

	Parameters tested	Best	Best Testing Acc
MLP	1-tanh-sgd-5 Acc = 74% 2-tanh-sgd-10, Acc = 65% 3-relu-sgd-10, Acc = 64% 4-relu-lbfgs-10, Acc= 97% Alpha in all of them = 1e-5	4	98.7%
SVM	1-Kernel: Linear Acc = 100% 2-Kernel: Polynomial Acc = 71.5% 3-Kernel: Sigmoid Acc = 68%	1	100%
KNN	1-K=1, Acc = 95% 2-K=2, Acc = 79% 3-K=3, Acc = 76%	1	95%

As shown in the table, very high testing accuracy achieved between 92% to 100%.

C. Performance

The parameters used to evaluate each model was through accuracy, recall, f-measure, precision. These metrics extracted from the confusion matrix generated from each model by using confusion_matrix function in sklearn.

The equation of each metric is represented as:

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

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

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

$$Fmeasure = \frac{2 * Precision * Recall}{Precision + Recall}$$

Where TP are the true positives which are the correctly classified patients. While TN are the true negatives which are patients that does not belong to a specific class and classified correctly.

The FP are the false positives which are data wrongly classified as positives of a specific class. While FN are false negatives which are patients belongs to a specific class and classified as does not belong to that class.

VI. RESULTS AND DISCUSSIONS

a) Multilayer precptron:

MLP performance was very high. As shown in confusion matrix and the performance metrics. The precision recall and f measure of class 1,2 is 1 supporting 54,4 samples from classes 1,2 respectively which is excellent. In addition, class 3 achieved 0.95 precision, 1 recall and 0.97 for f1 measure supporting 18 samples. The 4th class achieved 0 supporting only 1 class. This is expected because in the whole dataset class 4 counts are available only 4 times. The overall accuracy was very high as shown here it reached 0.99.

Confusion Matrix MLP:

```
[[54  0  0  0]
 [ 0  4  0  0]
 [ 0  0 18  1]
 [ 0 0  0 0]]
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	54
2	1.00	1.00	1.00	4
3	0.95	1.00	0.97	18
4	0.00	0.00	0.00	1
accuracy			0.99	77
macro avg	0.74	0.75	0.74	77
weighted avg	0.97	0.99	0.98	77

In Fig.6 showing the MLP Results in Python3.

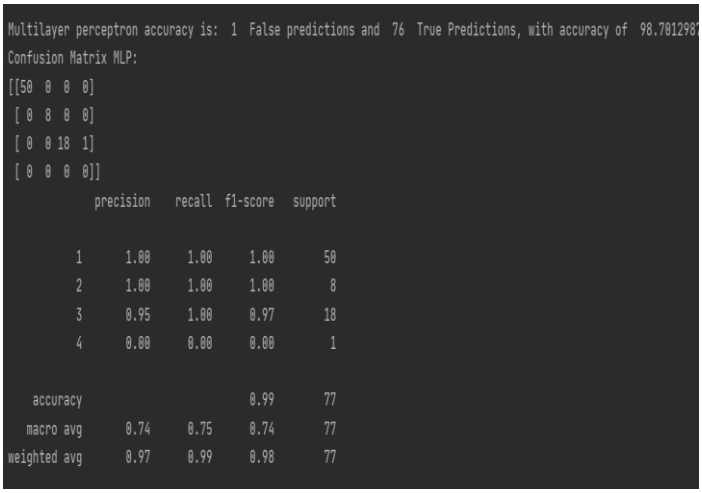

Figure 6: MLP Results

MLP Applied using the following 2 line of codes in Python3

```
ANN = MLPClassifier(solver= 'lbfgs', alpha= 1e-5,
hidden_layer_sizes=(10), random_state=1, activation= 'relu')
ANN.fit(x_train, y_train)
```

b) Support Vector Machine:

SVM performance was the highest among the 3 models. As mentioned before using linear kernel, the model reached testing

Confusion Matrix SVM:

```
[[54  0  0  0]
 [ 0  4  0  0]
 [ 0  0 18  0]
 [ 0  0  0  1]]
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	54
2	1.00	1.00	1.00	4
3	1.00	1.00	1.00	18
4	1.00	1.00	1.00	1
accuracy			1.00	77
macro avg	1.00	1.00	1.00	77
weighted avg	1.00	1.00	1.00	77

classification accuracy 100%. The model evaluation metrics are shown here.

precision, recall and f1-measure are 1 for all classes and accuracy is 100%. The reason of that is that the data after filtering and removing missing data was < 300 row which is very low amount of data.

The SVM implemented through the following 2 line of code

```
SVM = svm.SVC(kernel="linear")
SVM.fit(x_train, y_train)
```

The SVM results are shown here in Fig. 7

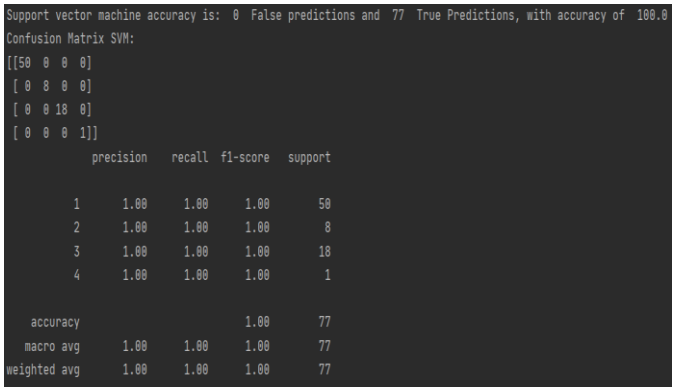

Figure 7: SVM Results

c) K nearest neighbour:

The algorithm achieved its highest accuracy at K=1. The evaluation metrics are shown here below

Confusion Matrix KNN:

```
[[51  0  1  0]
 [ 2  4  0  0]
 [ 1  0 17  0]
 [ 0  0  0  1]]
```

	precision	recall	f1-score	support
1	0.98	0.94	0.96	54
2	0.67	1.00	0.80	4
3	0.94	0.94	0.94	18
4	1.00	1.00	1.00	1
accuracy			0.95	77
macro avg	0.90	0.97	0.93	77
weighted avg	0.96	0.95	0.95	77

KNN was done by the following 2 line of code.

```
knn = KNeighborsClassifier(n_neighbors= 1)
knn.fit(x_train,y_train)
```

KNN results are shown here in Python in Fig. 8.

```
K-Nearest neighbors accuracy is:  4  False predictions and  73  True Predictions, with accuracy of  94.80
Confusion Matrix KNN:
[[50  3  1  0]
 [ 0  3  0  0]
 [ 0  0 18  0]
 [ 0  0  0  2]]
              precision    recall  f1-score   support

           1       0.93      1.00      0.96        50
           2       1.00      0.50      0.67         6
           3       1.00      0.95      0.97        19
           4       1.00      1.00      1.00         2

    accuracy                           0.95        77
   macro avg       0.98      0.86      0.90        77
weighted avg       0.95      0.95      0.94        77
```

Figure 8: KNN Results

It was the lowest among the 3 models. The precision is 0.98 for class 1 and the recall was 0.94 while the f1measure was 0.96. This is considered good among the 54 counts. However, in the 2nd class it achieved 0.67 precision and this is very low due to the low amount of the 2nd class among the testing data while the recall was 1 and the f-measure was 0.80. in the 4th class it achieved 1 in the 3 metrics. The 3rd class it achieved 0.94 in the 3 metrics since it classified 18 counts.

d) Overall Metrics evaluation:

The overall metrics are represented in Fig. 9 using the average precision, recall and f measure of each class in addition to the accuracy of each model to be represented here. It is clear that SVM was the best among the 3 models but since the data are very small so we can not depend on these results 100%.

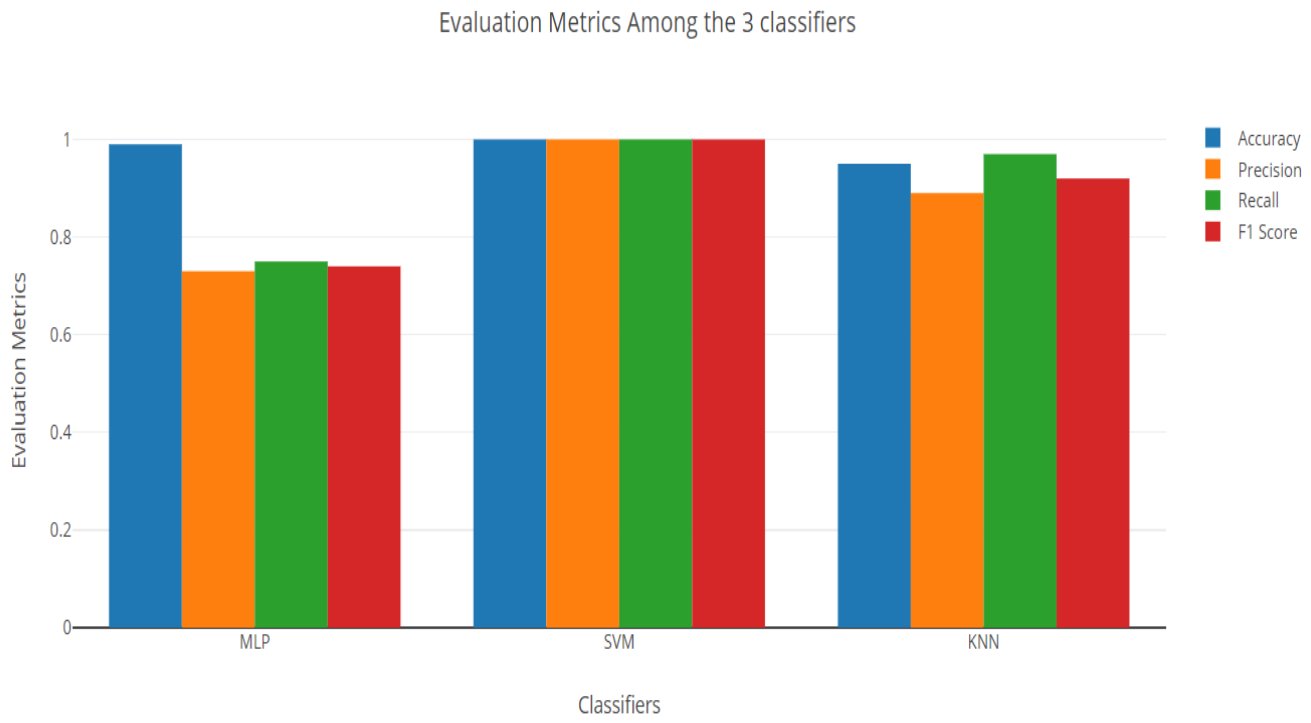

Figure 9: Evaluation Metrics

VII. Conclusion

To sum up, this work compares 3 different classification algorithms which are MLP, SVM and KNN to classify 4 classes of Covid19 patients. The Performance was very high, and this is due to the small amount of data used. The data was very large but with a lot of missing data and after filtering the data it becomes <300 row.

This work was able to predict the patients that has higher risk of death or critical condition according to several patient's data from different countries.

For the Future work we are planning to search for another dataset that is larger and does not have much missing data and to test it on the same models.

References

[1] World Health Organization, Coronavirus disease (COVID-19) Pandemic available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[2] Li L, Qin L, Xu Z, et al. Using Artificial Intelligence to Detect COVID-19 and Community-acquired Pneumonia Based on Pulmonary CT: Evaluation of the Diagnostic Accuracy. Radiology. 2020;296(2):E65-E71. doi:10.1148/radiol.2020200905

[3] World Health Organization, Coronavirus disease (COVID-19) Pandemic available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[4] Machine Learning from wikipedia the free encylopedia, available: https://en.wikipedia.org/wiki/Machine_learning

[5] B. Ali, M. Omar, N. Qassim, A. Manar and A. Mohammad, "Machine Learning Classifications of Coronary Artery Disease", IEEE 2018 November 2018 [2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)]

[6] KNN how it Works, available at: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761