# Covid-19 severity prediction

ML phase III report

Moneer Zaki
Computer Engineering
The American University in Cairo
moneerzaki@aucegypt.edu

Mario Ghaly
Computer Engineering
The American University in Cario
mariomamdouh@aucegypt.edu

## ABSTRACT

This study conducts a comparative analysis of various supervised machine-learning models on a cleaned and preprocessed dataset. It starts with small and final modifications to the dataset before. The models then evaluated include K-Nearest Neighbors (KNN) with different distance measures, Logistic Regression, Decision Trees, Random Forest, Naive Bayes, Neural Networks, and Support Vector Machines (SVM). The analysis encompasses experimental setup, parameter selection, performance evaluation, and considerations of model suitability for the problem domain, culminating in a proposal for the most appropriate technique based on performance and problem fit.

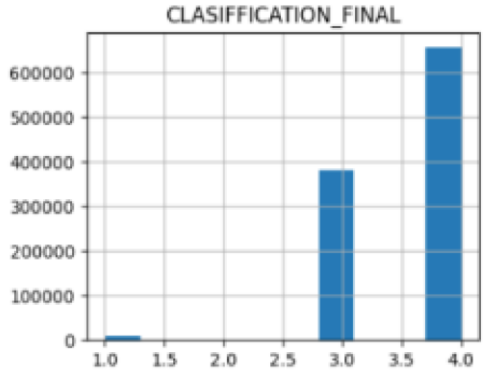
## FINAL DATA PREPARATION

As can be seen in the following 2 figures, Those are all the columns, attributes with their own unique values. Eg: USMER has values of 1 and 2. MEDICAL_UNIT has values from 1 to 13.

```
USMER
[2 1]
MEDICAL_UNIT
[ 1  2  3  4  5  6  7  8  9 10 11 12 13]
SEX
[1 2]
PATIENT_TYPE
[1 2]
DEAD
[1 2]
PNEUMONIA
[1 2]
AGE
[ 65  72  55  53  68  40  64  37  25  38  24  30  48  23  80  61  54  59
  45  26  32  49  39  27  57  20  56  47  50  46  43  28  33  16  62  58
  36  44  66  52  51  35  19  90  34  22  29  14  31  42  15   0  17  41
   2  10   1  12   4   7   6   8  60   5  13  63  75  81  67  18  70  88
  85  92  73  74  78  76  82  77  86  71  95  87  83  84  79  69  89   3
  97  93 100  91  21 103  11   9  94  96 101 107 102  98  99 109 119 116
 105 111 104 114 120 106 110 118 117 121 108 115 113]
PREGNANT
[2 1]
DIABETES
[2 1]
COPD
[2 1]
ASTHMA
[2 1]
INMSUPR
[2 1]
HIPERTENSION
[1 2]
OTHER_DISEASE
[2 1]
CARDIOVASCULAR
[2 1]
OBESITY
[2 1]
RENAL_CHRONIC
[2 1]
TOBACCO
[2 1]
```

Thus, edits needed to be done are:

1. Convert all binary values of range from 1 to 2 to be or range from 0 to 1
2. Use One-hot encoding method for “MEDICAL_UNIT” as its numerical values represent names in origin.
3. In Phase II, the output column “CLASSIFICATION_FINAL” was as the following graph:




The imbalanced classes was a motivation to combine all 3 values (1,2,3) to be only 1 value of 1 which means the patient has covid. And the value of 4 would be of value 0 now, which means the patient does not have covid. Thus, “CLASSIFICATION_FINAL” should now have values 0 and 1, such that “0” means the patient has no covid, while value “1” means that the patient does have covid, making the label classes distributed into around 61% to 39%, which is relatively better.

4. Regarding the “AGE” column, it needed some modifications depending on the model that is being trained on.

Finally, here is the final pre-processed data with their new values:

```
USMER
[0 1]
SEX
[1 0]
PATIENT_TYPE
[1 0]
DEAD
[1 0]
PNEUMONIA
[1 0]
AGE
[ 65  72  55  53  68  40  64  37  25  38  24  30  48  23  80  61  54  59
  45  26  32  49  39  27  57  20  56  47  50  46  43  28  33  16  62  58
  36  44  66  52  51  35  19  90  34  22  29  14  31  42  15   1   0  17
  41   2  10  12  13   4   7   6   8  60   5  63  75  81  67  18  70  88
  85  92  73  74  78  76  82  77  86  71  95  87  83  84  79  69  89   3
  97  93 100  91  21 103  11   9  94  96 101 107 102  98  99 109 119 116
 105 104 111 114 120 106 108 110 118 117 121 115 113]
PREGNANT
[0 1]
DIABETES
[0 1]
COPD
[0 1]
ASTHMA
[0 1]
INMSUPR
[0 1]
HIPERTENSION
[1 0]
OTHER_DISEASE
[0 1]
CARDIOVASCULAR
[0 1]
OBESITY
[0 1]
RENAL_CHRONIC
[0 1]
TOBACCO
[0 1]
MEDICAL_UNIT_1
[1 0]
MEDICAL_UNIT_2
[0 1]
MEDICAL_UNIT_3
[0 1]
MEDICAL_UNIT_4
[0 1]
MEDICAL_UNIT_5
[0 1]
MEDICAL_UNIT_5
[0 1]
MEDICAL_UNIT_6
[0 1]
MEDICAL_UNIT_7
[0 1]
MEDICAL_UNIT_8
[0 1]
MEDICAL_UNIT_9
[0 1]
MEDICAL_UNIT_10
[0 1]
MEDICAL_UNIT_11
[0 1]
MEDICAL_UNIT_12
[0 1]
MEDICAL_UNIT_13
[0 1]
CLASIFFICATION_FINAL
[1 0]
```

## GENERAL EXPERIMENTAL SETUP

All models were imported from sklearn, and implemented using cross-validation with a number of folds = 10. This number of folds was chosen to divide the data as 90% training and 10% testing in each fold. KNN was the only model with folds = 5 since it is computationally expensive and takes a lot of time. Also, only KNN and SVM were implemented using cuML to utilize GPU instead.

For the performance metrics, all models reported accuracy, recall, precision, F1-score, and AUC, except for a very minor cases

in KNN and SVM where the accuracy was not recorded when migrating the code to run on cuML instead of sklearn. However, this would not be of a negative impact on the final decision as accuracy is not a reliable metric for this dataset; what is more important is recall and the F1-score.

## KNN

**Model Description**

K-nearest neighbors is a type of instance-based learning model. It operates by retrieving a set of previously classified instances from the training set that share similar traits to a new instance provided for classification. This retrieved set is then used to determine the class of the new instance based on the most frequent class among its neighbors. The researchers find K-nearest neighbors suitable for the problem at hand because similar items, such as used automobiles, tend to have similar prices, aligning well with the model's principle.

**Experimental Setup & Parameter Choice**

All the data in the dataframe is preprocessed and ready to be used in the KNN model except for the age column which needs to be normalized. Knowing that the range of values of "AGE" column is from 0 to 120, normalization between values 0 and 1 is applied by min/max scaling for the distance measure without the age overcoming other distances for other attributes.

```
AGE
[0.53719008 0.59504132 0.45454545 0.43801653 0.56198347 0.33057851
 0.52892562 0.30578512 0.20661157 0.31404959 0.19834711 0.24793388
 0.39669421 0.19008264 0.66115702 0.50413223 0.44628099 0.48760331
 0.37190083 0.21487603 0.26446281 0.40495868 0.32231405 0.2231405
 0.47107438 0.16528926 0.46280992 0.38842975 0.41322314 0.38016529
 0.3553719  0.23140496 0.27272727 0.1322314  0.51239669 0.47933884
 0.29752066 0.36363636 0.54545455 0.42975207 0.4214876  0.2892562
 0.15702479 0.74380165 0.28099174 0.18181818 0.23966942 0.11570248
 0.25619835 0.34710744 0.12396694 0.00826446 0.         0.14049587
 0.33884298 0.01652893 0.08264463 0.09917355 0.10743802 0.03305785
 0.05785124 0.04958678 0.0661157  0.49586777 0.04132231 0.52066116
 0.61983471 0.66942149 0.55371901 0.14876033 0.5785124  0.72727273
 0.70247934 0.76033058 0.60330579 0.61157025 0.6446281  0.62809917
 0.67768595 0.63636364 0.7107438  0.58677686 0.78512397 0.71900826
 0.68595041 0.69421488 0.65289256 0.57024793 0.73553719 0.02479339
 0.80165289 0.76859504 0.82644628 0.75206612 0.17355372 0.85123967
 0.09090909 0.07438017 0.7768595  0.79338843 0.83471074 0.88429752
 0.84297521 0.80991736 0.81818182 0.90082645 0.98347107 0.95867769
 0.8677686  0.85950413 0.91735537 0.94214876 0.99173554 0.87603306
 0.89256198 0.90909091 0.97520661 0.96694215 1.         0.95041322
 0.9338843 ]
```

At the beginning a value of k = 5 for the KNN model was chosen, and 'Eucledian' and 'Manhattan' metrics were tried.

**Performance Evaluation**

```
Distance Metric: euclidean
True Positive: 325578
True Negative: 62646
False Positive: 593950
False Negative: 66401
Precision: 0.3540707841414206
Recall: 0.8306006189107069
Error Rate: 0.6297603890994922
Accuracy: 0.3702396109050783
Specificity: 0.09541026750086812
False Positive Rate: 0.9045897324991319
True Negative Rate: 0.09541026750086812
F1 Score: 0.4964944906889555
Cross-Validation Time: 7363.786397457123 seconds
```

```
Distance Metric: manhattan
True Positive: 325726
True Negative: 62331
False Positive: 594265
False Negative: 66253
Precision: 0.3540534635664914
Recall: 0.8309781901581462
Error Rate: 0.6299196528622177
Accuracy: 0.3700803471377822
Specificity: 0.09493052044179373
...
True Negative Rate: 0.09493052044179373
F1 Score: 0.4965448905081671
Cross-Validation Time: 19208.221725702286 seconds
```

Since, the most critical case is predicting that a covid holder should be predicted true. Thus, the recall value is the most important among all other performance values. As can be seen from different distance measures, the recall is high around 83%.

Logically, KNN is the best machine learning model to be used in medical cases for the following reason. The patient is more likely to be diagnosed as a Covid carrier if there are many cases much similar to his own case. Thus, it is predicted that we find KNN with high recall value.

## LOGISTIC REGRESSION

**Model Description**

Logistic regression is a statistical and probabilistic method that is convenient for binary classification, which is the target problem as the patient either has or doesn't have COVID. Despite its name, logistic regression is a classification algorithm rather than a regression algorithm.

**Experimental Setup & Parameter Choice**

- Since this model gives weights for every numerical feature, and the only constraint for it is that all features should be numerical, then the dataset was ready for experimentation on logistic regression.
- The essential hyperparameter choices were:
  - *random_state = 0* → The random sequences generated by the algorithm will be the same every time you run the code
  - max_iter = 1500 → Increasing the default value of maximum iterations(100) before convergence to yield better results
  - The optimization solver to find the best weights was set to be the default('lbfgs')

**Performance Evaluation**

```
Final Metrics>>
Accuracy = 0.6145274828507249
Precision = 0.6106977211624736
Recall = 0.2319889892557529
F1-score = 0.256096831250162
AUC = 0.5374435718127113
```

The performance metrics induce that logistic regression is not a very good fit for the problem in hand. Although precision at 61% could be ok, recall = 23% and F1-score = 0.25 is very low. This could be explained as since the logistic regression assigns individual weights for every feature and relies on a linear decision boundary, it may struggle to capture complex relationships present in the data. Additionally, logistic regression assumes that the relationship between the features and the target variable is linear, which might not hold true for this problem. Moreover, the threshold variable was not modified.

## DECISION TREE

**Model Description**

Decision trees employ inductive learning techniques, deriving overarching conclusions from a series of factual observations and feature characteristics. This algorithm stands out not only for its simplicity in comprehension and implementation but also for its widespread success and popularity. Decision trees serve primarily for classification tasks, making them a candidate to solve this problem.

**Experimental Setup & Parameter Choice**

- Instead of letting the model handle the binning of the age column, it was handled before initializing the model since there are already some pre-conceptions about the relation between age and being diagnosed with COVID-19. Also, it gives space for trying different binning ranges. The age ranges were as follows:
    - Infant: 0-2 years
    - Child: 3-12 years
    - Teenager: 13-19 years
    - Young adult: 20-39 years
    - Middle-aged adult: 40-59 years
    - Older adult: 60-79 years
    - Elderly: 80+ years
- Entropy criteria is used: Entropy serves as a crucial metric in different decision tree models due to its ability to quantify the uncertainty or disorder within a dataset. By utilizing entropy, these models can intelligently select features that result in the most informative splits, ultimately leading to the construction of accurate and efficient decision trees.
- In the second trial a variable called, splitter "best"

**Performance Evaluation**

```
Final Metrics>>
Accuracy = 0.616876380580234
Precision = 0.5724013697125163
Recall = 0.21556983665306445
F1-score = 0.27333832793611645
AUC = 0.536010524749739
```

```
Final Metrics>>
Accuracy = 0.6184489837927376
Precision = 0.5747247903965494
Recall = 0.218442433625531
F1-score = 0.27586335068067036
AUC = 0.5378450871814664
```

Since, the recall is the most important Which we got 21% in both trees. Which is not very good relative to other models. While we have a very good accuracy and precision rates, but not very important in such model. Thus, not taken into consideration.

## RANDOM FOREST

**Model Description**

Random forest addresses the issue of high variance encountered in decision trees by leveraging parallel decision trees with different features. By aggregating the outputs of multiple trees, it effectively mitigates variance, leading to more robust and accurate predictions. Regarding the classification problem in hand, Random forest takes the majority vote of the trees, which reduces the problem of overfitting in the normal decision trees.

**Experimental Setup & Parameter Choice**

- The dataset was ready for running; however, as stated earlier, the 'AGE' column was binned manually for the purpose of trying different binning ranges. The new binning was first (0,15) as these are the ages that are most likely to be safe from the coronavirus, then (16, 30) as the relatively healthy youth group, after that (31,50) to indicate old people -- but not very old -- finally, (51+) to indicate older people with health concerns.
- The essential hyperparameters were the following:
    - *n_estimators = 100* → The number of decision trees in the forest was set to 100 to reduce overfitting and get more accurate predictions balanced with computational efficiency
    - max_depth = 10 → The maximum depth of each decision tree in the forest was set to 10 to reduce overfitting and complexity
    - min_samples_split = 5000 → The minimum number of samples required to split an internal node was set to 5000 to prevent splitting too early

**Performance Evaluation**

```
Final Metrics>>
Accuracy = 0.6300732867517357
Precision = 0.6159879649373736
Recall = 0.22246557555316473
F1-score = 0.27648361100741387
AUC = 0.5479377069289924
```

Random forests yielded such a small improvement form decision trees, and its performance is still not very good. A potential reason for these results could be the imbalanced data -- which is going to be explained in detail in the final section. Random Forests may struggle to effectively learn from the minority class. Although Random Forests have mechanisms to handle imbalanced data, they may still produce biased predictions or overlook important minority class instances, which is in this case the more important label value(1). Consequently, recall and F1-score are bad.

## NAÏVE BAYES

**Model Description**

The Naive Bayes classifier is a probabilistic classification approach grounded in Bayes' Theorem, operating under the assumption of independence among predictors. However, its reliance on the assumption of data independence can be a double-edged sword, potentially compromising reliability when input features are correlated. There is also Bayesian Networks as a supervised machine learning model that accommodates this drawback. However, structuring the network of features is very difficult and computational. Thus, it was considered less practical or feasible for this dataset.

**Experimental Setup & Parameter Choice**

There are no specific parameters to choose from in the naive Bayes model, and there are no constraints on the feature types. Hence, the model was ready to run directly.

**Performance Evaluation**

```
Final Metrics>>
Accuracy = 0.6184489837927376
Precision = 0.5747247903965494
Recall = 0.218442433625531
F1-score = 0.27586335068067036
AUC = 0.5378450871814664
```

As could be seen, the Recall was again 21.8% which is not very good. There is arguably ok accuracy and precision. However, for the target of this project, they are not of super importance while viewing the performance on our dataset, as the recall is still of the most importance.

Reasoning for these results could be the fact that Naive Bayes assumes that all features are conditionally independent given the class label. However, in earlier heat maps in the prior report, there were actually some decent correlations between the features and each other. Moreover, the same reason of the dataset nature and feature values imbalance as in Random Forests holds here.

## NEURAL NETWORK

**Model Description**

The Artificial Neural Networks algorithm draws inspiration from the human brain, aiming to predict outcomes and model patterns. Comprising multilayer perceptrons, it operates primarily through two key algorithms: the feedforward algorithm for producing outputs and the backpropagation algorithm for training and adjusting weights. Neural networks' biggest pros is that it can model complex and nonlinear relationships between features and the target variable. This gives an advantage over the aforementioned supervised machine learning models.

**Experimental Setup & Parameter Choice**

- The maximum number of iterations before convergence to set the weights is set to ***300*** for computational complexity. However, if this model was chosen, this parameter shall be more
- The number of hidden layers was set to ***2*** to avoid overfitting in a dataset of 1 million rows
- The number of neurons in the hidden layers was ***128***, ***32*** respectively in relation to the number of features, the number of samples, and that this is a binary label
- The activation function was set to ***RELU***

**Performance Evaluation**

```
Final Metrics>>
Accuracy = 0.6268041044936805
Precision = 0.6065065708624977
Recall = 0.2174653272069845
F1-score = 0.26609367192561956
AUC = 0.5443197019214483
```

Despite predicting to have better performance when utilizing Neural Networks, the results did not support the assumption. Neural Networks' main drawbacks are that they overfit -- which is not the issue here -- and that they take a lot of time. Training this model on CPU with these parameters took around 4 hours, and before it, another NN model with the same parameters, but with hidden layers = [10,10], and the performance metrics were not better. Nevertheless, Neural Networks could yield better results theoretically if given different parameters as making the activation function sigmoid instead of RELU.

## SVM

**Model Description**

Support Vector Machine (SVM) is a supervised learning model used for classification and regression tasks. It works by finding the optimal hyperplane that best separates different classes in the feature space. SVM aims to maximize the margin between classes, effectively identifying the boundary with the greatest separation between them. It can handle linear and non-linear relationships in data through the use of various kernel functions. SVM is particularly effective in high-dimensional spaces and is widely.

**Experimental Setup & Parameter Choice**

There is no experimental setup needed for this model, and the kernel function is the default ‘rbf’

**Performance Evaluation**

```
Precision = 0.4143675841451211
Recall = 0.35535795178207097
F1-score = 0.3680850136731414
AUC = 0.5189801511585547
```

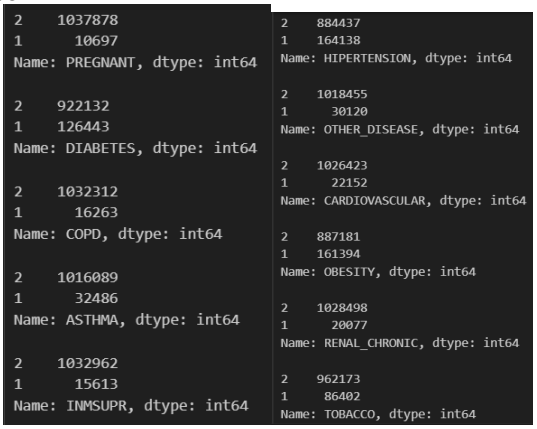
The Recall value is 35% which is the second-highest model after KNN. That means the performance of such model is of high importance in predicting Covid carriers. However, the precision, and F1-scores are not very high, but not of a great deal.
SVM cons are that it is memory intensive and that when its model was run, it took the longest time among all models. This is a major drawback despite the fact that it has the potential, theoretically, to produce better metrics.

## PERFORMANCE COMPARISON

| Model | Accuracy | Precision | Recall | F-score | AUC |
|---|---|---|---|---|---|
| KNN (euclidean, k = 5) | 37% | 35% | 83% | 49% | 0.45 |
| KNN (euclidean, k = 4) | _ | 37% | 93% | 53% | 0.5 |
| KNN (euclidean, k = 3) | _ | 37% | 88% | 52% | 0.49 |
| KNN (euclidean, k = 2) | _ | 37% | 92% | 53% | 0.49 |
| KNN (euclidean, k = 1) | _ | 34% | 84% | 48% | 0.45 |
| KNN (Manhattan) | 37% | 35% | 83% | 49% | 0.3 |
| | | | | | |
| Logistic reg | 61% | 61% | 23% | 25% | 0.537 |
| Decision tree ID3 | 61.7% | 57% | 21% | 27% | 0.536 |
| Decision Tree C4.5 | 61.8% | 57% | 21% | 27% | 0.5378 |
| Random Forest | 63% | 61.5% | 22% | 27% | 0.547 |
| Naive Bayes | 62% | 57% | 22% | 27% | 0.5378 |
| Neural Networks | 62% | 60% | 22% | 27% | 0.544 |
| SVM | _ | 41.4% | 35.5% | 36.8% | 0.51898 |

The performance of all the supervised learning models mentioned beforehand was not very efficient, which is due to multiple reasons including but not limited to:

- All features have low correlations with the label. The maximum correlation was around 18%
- Most features unique values are imbalanced. For example, as attached in the screenshots below, the value distributions of most disease features are around 90% not suffering from the disease and 10% suffering, which makes the problem harder to solve




- Another issue could be the fact that COVID is not a simple disease to predict and there could be patients' records identical to each other, but having different labels. Even after more discretizing of the label to be a binary label, there is still an imbalance in the 2 classes as it is around 61% negative to 39% positive.

Relatively, the top-performing models are KNN, SVM, and NN. Performance metrics of KNN are the best since the most important performance metric relatively is recall. This is simply because the final application/model's purpose is to predict whether a patient has COVID or not, and having higher recall than precision indicates that the model could detect more carriers better despite predicting many non-carriers of COVID as carriers. Moreover, KNN classifier has the highest f-1 score, which is a reliable metric that is unbiased, taking the harmonic mean of both recall and precision.

The reason for picking SVM and NN as candidates for the top 3 models is that SVM has the second highest recall and F1-score, and it has the potential to increase the performance by modifying the parameters. On the other hand, NN has many parameters also that could be modified and significantly increase the performance metrics, and more importantly, it is more fit for this problem than other models since it constructs non-linear relations between the features and the output.

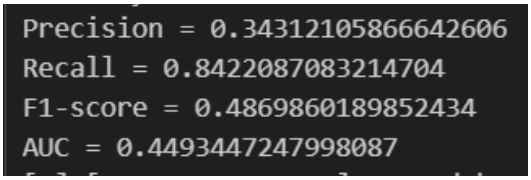
## FINAL MODEL CHOICE

For this project choice, the decision was to choose KNN as the best model because it has the highest Recall by far among other models and the highest f1-score. As stated before the Recall is of super importance because there is a crucial need to predict those who have coronavirus correctly. Additionally, KNN as a concept is very suitable for this problem. COVID-19 in its early discovery was fought by avoiding the common symptoms that are shown among its

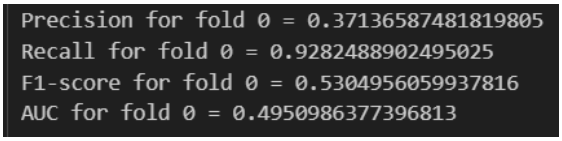
carriers, and indeed most healthcare problems have medical records in common. Thus, using an instance-based learning model is justified for a healthcare problem.

Moreover, cosine similarity has not been tested yet, and might yield better results since most of the features are binary. As a result of the KNN being the best model so far, the parameter choice of the model was changed by trying different K-values from 1 to 5, and also trying the metric to be Manhattan with k = 5. Regarding other parameters, this would be modified in the next step to try achieving the best performance possible for this model on this dataset. There would be more testing for the K value in the upcoming phase.

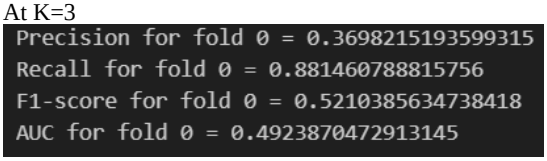
At K=1



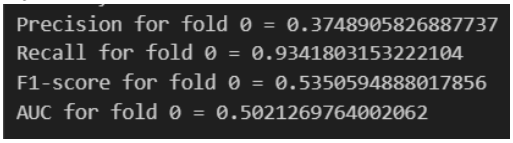

At K=2




At K=3




At K=4




At K=5