

# Machine Learning Algorithms for Predicting the Results of COVID-19 Coronavirus Infection

Yuri Kravchenko, Nataliia Dakhno, Olga Leshchenko, Anastasiia Tolstokorova

Taras Shevchenko National University of Kyiv, Volodymyrs'ka str. 64/13, Kyiv, 01601, Ukraine

## Abstract

The paper analyzes data collected from around the world on patients with COVID-19. The patients studied were of different ages, with different chronic diseases and symptoms, men and women. A binary classifier has been developed that considers data on a person's health, symptoms, patient's age, and other properties and determines the patient's disease outcome by assigning it to one of two categories: fatal or not. The work's practical value is to help hospitals and health facilities decide who needs care in the first place when the system is overcrowded and to eliminate delays in providing the necessary care.

## Keywords <sup>1</sup>

supervised learning, classification problem, model fitting, feature selection, feature engineering, data normalization, model validation, confusion matrix, logistic regression, Naive Bayes, Decision tree, random forest.

## 1. Introduction

In March 2020, the World Health Organization officially declared the Covid-19 coronavirus a global pandemic. COVID-19 coronavirus disease is an infectious disease caused by the recently discovered coronavirus SARS-CoV-2. The danger of this disease lies not so much in its lethality but the rate of its spread after a long incubation period. The infected person does not yet experience any symptoms and continues to contact people, spreading the infection. The best way to prevent and slow down infection transmission is to be well informed about the COVID-19 virus, the disease it causes, and how it spreads. By visualizing the development of the disease in other countries where the outbreak has passed, it is possible to build a truly effective behavior strategy that will save lives while minimally harming the economy, if possible, in these circumstances.

The coronavirus outbreak originated in the Chinese city of Wuhan but has now spread to the rest of the world. Cases of infection continue to grow exponentially. As a result, workers are transferred to telecommuting, pupils and students begin to study at home, conferences are canceled, store shelves are devastated, and the global economy is under serious threat. It is undeniable that coronavirus infection has irreparably affected all spheres of human life - from education to global economic change. It is safe to say that this is one of the most severe health crises in decades, if not centuries.

However, there are currently almost no systematic reviews in the available literature that describe the accumulated data on COVID-19 and suggest methods for their analysis.

Data analysis will help to understand the basic patterns of data behavior better. Furthermore, a more thorough analysis based on data and sound forecasts can be useful for decision-making and policy-making [1,2,3]. To analyze and build the optimal strategy, you need to use a decision support system. (DSS) [4,5,6,7]. DSS is an automated expert-assistant in selecting decisions by the operator at the stages of analysis, forming possible scenarios, selecting the best of them, and evaluating the results of the decision [8,9,10]. There are currently significant advances in the development and

---

*IT&I-2020 Information Technology and Interactions, December 02–03, 2020, KNU Taras Shevchenko, Kyiv, Ukraine*

EMAIL: kr34@ukr.net (Y. Kravchenko); nataly.dakhno@ukr.net (N. Dakhno); lesolga@ukr.net (O. Leshchenko); tlstkr@gmail.com (A. Tolstokorova);

ORCID: 0000-0002-0281-4396 (Y. Kravchenko); 0000-0003-3892-4543 (N. Dakhno); 0000-0002-3997-2785 (O. Leshchenko)



© 2020 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

widespread practical application of mathematical models and methods for different classes of problems [11,12,13,14]. The rapid development of information technology, in particular, advances in data collection, storage, and processing methods, have allowed many organizations to collect vast amounts of data that need to be analyzed [15,16,17]. Also, the development of technology entails increasing the requirements for quality and accuracy of decision-making. It makes it necessary to develop further and improve methods in DSS. In this work, a DSS model was developed to help identify patterns between the characteristics of the patient (sex, age, types of symptoms, and chronic diseases) who contracted COVID-19 and mortality. This study offers a model of artificial intelligence that can provide hospitals and medical facilities with the information they need to address congestion. It will also allow developing a patient sorting strategy to address hospitalization priorities and eliminate delays in providing the necessary care.

## 2. Main part

To get started, we need to form a training set that meets the criteria of the goal. The training set is the data on which the algorithm is trained. The training looks different, depending on which algorithm is used. After training the model it is necessary to determine its effectiveness according to certain metrics.

### 2.1. Input data

Before the study, relevant data were found that meet the criteria for the work [18]. They contain a set of characteristics of patients with coronavirus. This data is sufficient to be able to divide them into training and test kits. They also have information about the disease's outcome, which will be a vector of labels of the binary classifier. This dataset collects data from more than 920,000 patients from around the world of all ages, with various chronic diseases and symptoms, including men and women.

The dataset's information has a lot of gaps and redundant data, so they were cleared before using them [19]. At the stage of data cleansing, all redundant and useless signs of patients are removed. Only gender, age, information on symptoms, chronic diseases, and treatment outcomes are used for the classification. Also, all data that does not contain information about the patient's illness's outcome is deleted from the dataset. In order to use binary classification, all the necessary features were coded. Each symptom was highlighted and marked with the number 1 or 0, if it is present or absent in each patient, respectively. The result was a dataset of the following form (Figure1).

```
data.head()
```

| city           | province  | country   | latitude  | longitude  | geo_resolution | date_onset_symptoms | ... | date_death_or_discharge | notes_for_disc |
|----------------|-----------|-----------|-----------|------------|----------------|---------------------|-----|-------------------------|----------------|
| Shek Lei       | Hong Kong | China     | 22.365019 | 114.133808 | point          | NaN                 | ... | NaN                     |                |
| Vo Euganeo     | Veneto    | Italy     | 45.297748 | 11.658382  | point          | NaN                 | ... | 22.02.2020              |                |
| NaN            | NaN       | Singapore | 1.353460  | 103.815100 | admin0         | NaN                 | ... | 17.02.2020              |                |
| Zhengzhou City | Henan     | China     | 34.629310 | 113.468000 | admin2         | NaN                 | ... | NaN                     |                |
| Pingxiang City | Jiangxi   | China     | 27.513560 | 113.902900 | admin2         | NaN                 | ... | NaN                     |                |

**Figure 1:** The dataset after encoding

In the input data, age differs in its scale. Different scales of data can negatively affect the gradient descent method's convergence because the cost function will be very stretched [20]. Therefore, the minimax method was applied to the data. To do this, use the formula:

$$x_n = \frac{x - x_{\min}}{x_{\max} - x_{\min}};$$

where  $x_n$  – normalized value of the feature,  $x$  – the current value of the feature,  $x_{\min}$  – the minimum value for this characteristic,  $x_{\max}$  – the maximum value for this characteristic.

With the help of trait normalization by the minimax method, the values were set in the range [0,1], because other traits lie in this range. After normalization, the data were obtained as follows (Figure 2).

|       | age      | chronic_disease_binary | if_dead | male | female | malaise | asthma | pneumonia | sore_throat | severe | ... | fev |
|-------|----------|------------------------|---------|------|--------|---------|--------|-----------|-------------|--------|-----|-----|
| 0     | 0.442105 | 0                      | 0       | 1    | 0      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 1     | 0.810526 | 0                      | 1       | 1    | 0      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 2     | 0.631579 | 0                      | 0       | 0    | 1      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 3     | 0.284211 | 0                      | 0       | 1    | 0      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 4     | 0.578947 | 0                      | 0       | 0    | 1      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| ...   | ...      | ...                    | ...     | ...  | ...    | ...     | ...    | ...       | ...         | ...    | ... | ... |
| 15575 | 0.831579 | 0                      | 1       | 0    | 1      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 15576 | 0.668421 | 1                      | 1       | 1    | 0      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 15577 | 0.984211 | 0                      | 1       | 0    | 1      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 15578 | 0.726316 | 0                      | 1       | 1    | 0      | 0       | 0      | 0         | 0           | 0      | 0   | ... |
| 15579 | 0.563158 | 0                      | 0       | 1    | 0      | 0       | 0      | 0         | 0           | 0      | 0   | ... |

Figure 2: The dataset after normalization

## 2.2. Data classification

After obtaining the required data, several algorithms were selected that are most suitable for achieving this goal. In this study, five classification algorithms were selected that had previously proven to be the best in this type of problem [21]. These are the following algorithms: Logistic regression,  $K$ -nearest neighbors algorithm, Decision trees, Reference vector method, Naive Bayesian classifier. For data analysis, we will use the Python programming language along with packages for data visualization and analysis. To accomplish this task, the Anaconda distribution kit was used as a software environment, which allows you to immediately install Python and the necessary libraries. Jupyter Notebook was chosen as the environment for the task execution. The main packages used during the research: Pandas, Numpy, Matplotlib, Seaborn, Scipy, Scikit-Learn.

The results of solving the binary classification problem were presented in the Confusion Matrix, which consists of four cells: TP - True Positive, objects that have been classified as positive and are actually positive (belonging to this class); FP - False Positive, which were classified as positive but in fact negative; FN - False Negative, which were classified as negative but actually positive; TN - True Negative, which were classified as negative and actually negative (do not belong to this class).

*Logistic regression:*

Logistic regression is a well-known statistical method for determining several factors' influence on a logical pair of results. The name "logistic regression" reflects the fact that the data curve is compressed by applying a logistic transformation to reduce the effect of extreme values.

In order to solve the regression problem instead of predicting a binary variable, a continuous variable with values on the interval [0,1] is assumed for any values of independent variables. This is achieved by applying the regression equation:

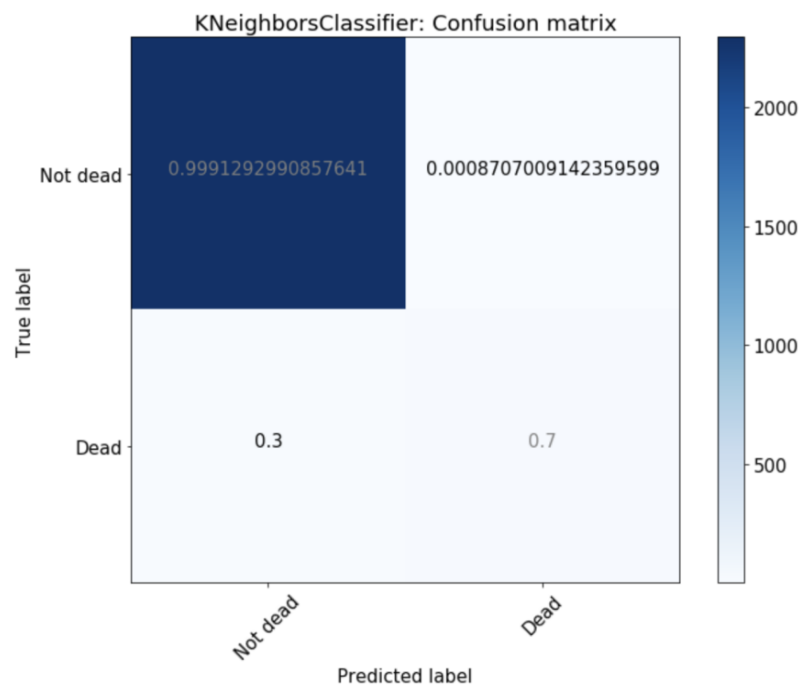
$$P = \frac{1}{1 + e^{-y}},$$

where  $P$  – the probability that an event of interest will occur,  $y$  – standard regression equation.

The normalized error matrix for the implemented logistic regression is presented in Figure 3.

*Algorithm of  $K$ -nearest neighbors:*

The method of  $K$ -nearest neighbors, or  $KNN$ -classification, defines the dividing boundaries locally. In the first variation of the  $1NN$ , each attribute belongs to a particular class depending on its nearest neighbor's information. In the  $KNN$  variant, each attribute belongs to the nearest neighbors' preferred class, where the  $k$  is the parameter of the method.



**Figure 3:** Normalized logistic regression confusion matrix

Consider some of the pros and cons of using the  $KNN$  algorithm. One of the advantages of  $KNN$  is that it is a straightforward algorithm. This makes the  $KNN$  algorithm much faster than other algorithms. One of the disadvantages of  $KNN$  - the algorithm does not work well with extensive data. With a large number of sizes of the algorithm, it becomes difficult to calculate the distance in each dimension. The  $KNN$  algorithm has a high prediction cost for large data sets. The normalized error matrix for the implemented  $K$ -nearest neighbor algorithm is presented in Figure 4.

*Decision trees:*

The decision tree is one of the most common and widely used machine learning algorithms with a teacher who can perform regression and classification tasks. For each attribute in the data set, the decision tree algorithm forms a node where the essential attribute is located in the root node. To evaluate, we start at the root node and work down the tree, following the appropriate node that matches our condition or "solution." This process continues until a sheet node is reached containing the forecast or result of the decision tree. The normalized error matrix for the implemented decision tree method is presented in Figure 5.

*Method of reference vectors:*

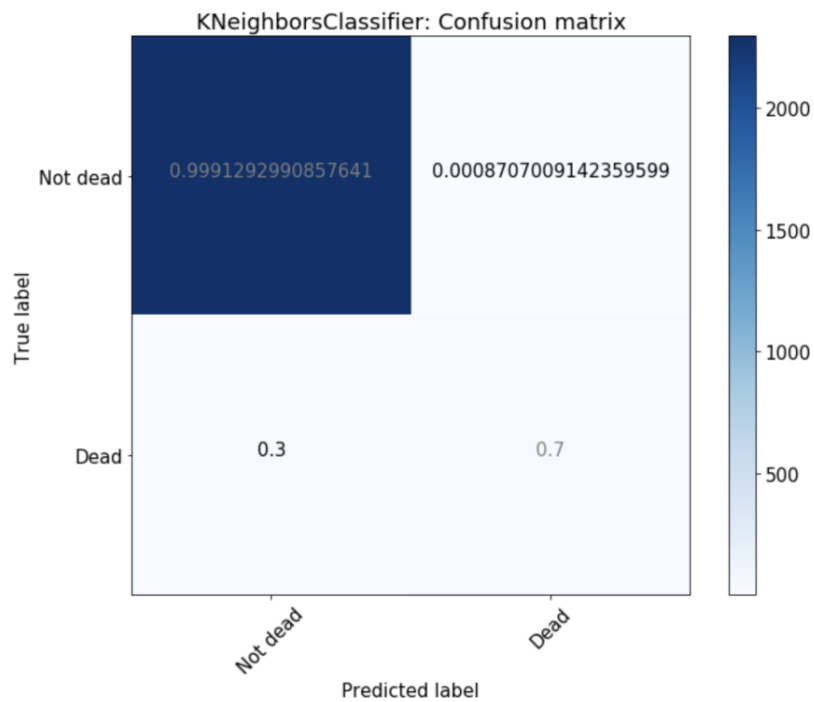
If the training set contains two classes of data that allow linear division, then there are many linear classifiers with which you can divide this data. The reference vectors method looks for a dividing surface (hyperplane) as far as possible from any data points. The dividing hyperplane is given by the offset parameter  $b$  (the point of intersection with the  $x$ -axis) and the normal vector  $\vec{w}$  to the dividing hyperplane. Since the dividing hyperplane is perpendicular to the vector of the normal  $\vec{w}$ , all points  $\vec{x}$  on the hyperplane satisfy the equation:

$$\vec{w}^T \cdot \vec{x} + b = 0.$$

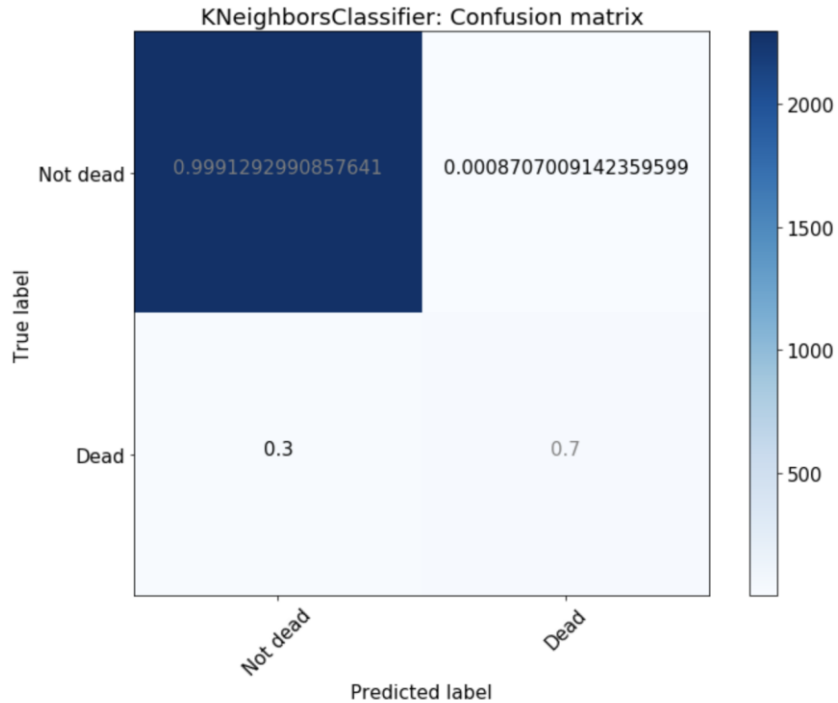
Now suppose we have a learning set  $D = \{(\vec{x}_i, y_i)\}$  in which each element is a pair consisting of a point  $\vec{x}$  and the corresponding class label  $y_i$ . In the method of reference vectors, two classes are

always called +1 and -1 (not 1 and 0). Therefore, the linear classifier is described by the following formula:

$$f(\vec{x}) = \text{sign}(\vec{w}^T \cdot \vec{x} + b)$$



**Figure 4:** Normalized  $K$ -neighbors classifier confusion matrix



**Figure 5:** Normalized decision tree method confusion matrix

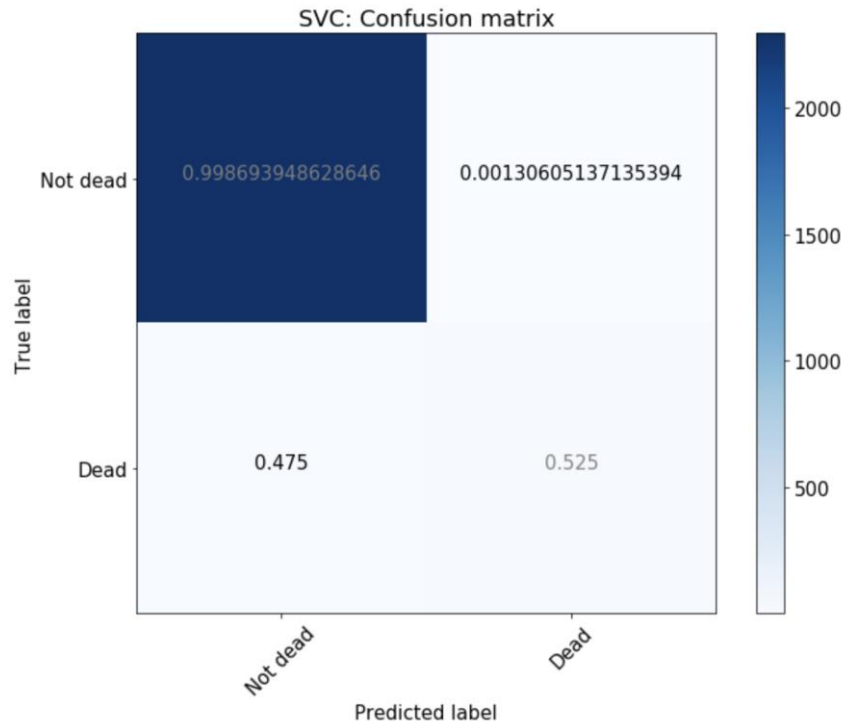
A value of -1 indicates one class and +1 indicates another. Next, we want to choose such  $\vec{w}$  и  $b$  which maximize the distance to each class. We can calculate that this distance is equal  $\frac{1}{\|\vec{w}\|}$ . The

problem of finding the maximum  $\frac{1}{\|\vec{w}\|}$  is equivalent to the problem of finding the minimum  $\|\vec{w}\|^2$ .

Let's write all this in the form of an optimization problem:

$$\begin{cases} \arg \min_{\vec{w}, b} \|\vec{w}\|^2, \\ y_i(\vec{w}^T \cdot \vec{x}_i + b) \geq 1, \quad i = \overline{1, m}. \end{cases}$$

The normalized error matrix for the implemented method of reference vectors is presented in Figure 6.



**Figure 6:** The SVC method confusion matrix

*Naive Bayesian Classifier:*

Naive Bayesian classifiers belong to the family of simple probability classifiers. They are based on Bayes' theorem with naive assumptions about the independence between features.

Bayes' theorem allows us to calculate the conditional probability:

$$P(C | x) = \frac{P(C) \cdot P(x | C)}{P(x)}.$$

If we classify an object that is a vector  $\vec{x} = (x_1, x_2, \dots, x_n)$  with  $n$  properties, then the classifier will find the probability of  $k$  possible classes for this object. If we take into account the "naive" assumptions about the conditional independence of the features, the numerator of the Bayesian formula will take the form:

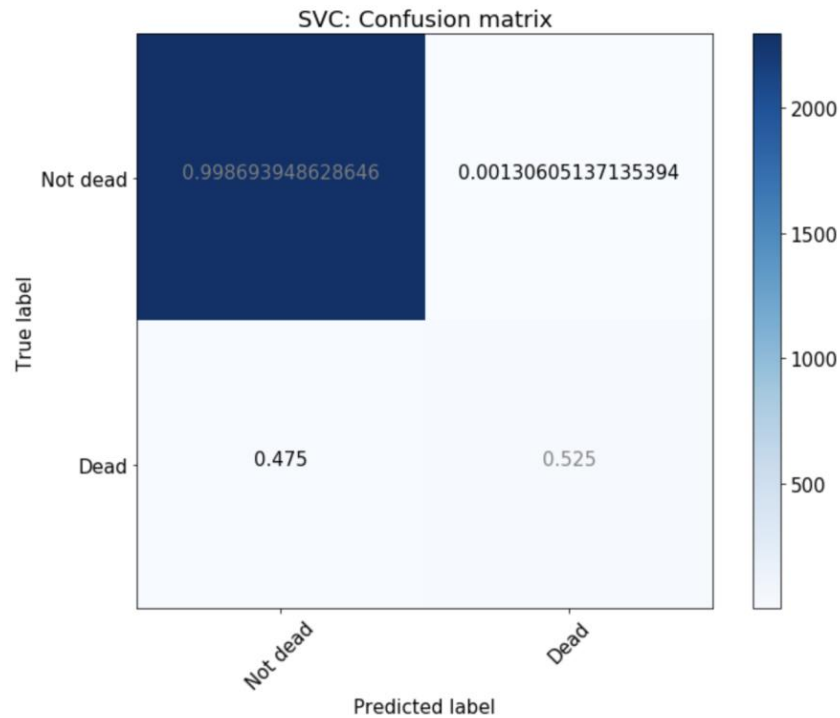
$$P(C_k | x_1, x_2, \dots, x_n) = \frac{P(C_k) \cdot \prod_{i=1}^n P(x_i | C_k)}{\sum_k P(C_k) P(x | C_k)}.$$

The corresponding classifier is a function that assigns a class label  $C_k$  for some  $k$  in the following way:

$$\hat{y} = \arg \max_k P(C_k | x_1, x_2, \dots, x_n) = \arg \max_k P(C_k) \cdot \prod_{i=1}^n P(x_i | C_k).$$

The normalized error matrix for the implemented naive Bayesian classifier is presented in Figure 7. To assess the quality of classification models that will solve this problem, the following metrics were chosen: accuracy, precision, recall, f-measure, logarithmic loss (logloss), area under the ROC curve. After evaluating the models, the classification report was analyzed by each of the metrics (Figure 3–7). The measure of accuracy is intuitive and obvious:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} .$$



**Figure 7:** The Naive Bayes classifier confusion matrix

Since the accuracy metric does not work well on unbalanced data, it was decided to allocate a separate sample for correct evaluation of the algorithms. Table 1 shows how the accuracy result differs in different models. After analyzing the work of different algorithms, we can conclude that the classifier of decision trees was the most accurate in this metric. Precision can be interpreted as the proportion of objects called positive by the classifier and at the same time actually positive [22]:

$$precision = \frac{TP}{TP + FP} .$$

Table 2 shows how the precision of the results differs in different models. According to the precision score metric, the best decision was again made by the classifier of decision trees, having the highest possible accuracy. Recall shows what proportion of positive class objects out of all positive class objects the algorithm found [22]:

$$recall = \frac{TP + TN}{TP + FN} .$$

Table 3 shows how the recall results differ in different models. According to this metric, the most complete was the classifier of decision trees, having a completeness of 0.8.

There are several different ways to combine precision and recall into an aggregate quality criterion. *F* -measure - average harmonic precision and recall [23]:

$$F = \frac{2precision \cdot recall}{precision + recall}$$

Table 4 shows how the *F* -measure of the results differs in different models. According to this metric, the most effective classifier was the classifier by the decision tree method. The logarithmic

loss metric was also chosen to study the efficiency of classification models [24]. The logloss metric assigns the weight of each predicted probability. The farther the probability from the actual value, the greater the weight. The goal is to minimize the total amount of all error weights.

$$\text{logloss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)].$$

**Table 1**

Classifier accuracy score comparison

| Classifier model            | Accuracy score     |
|-----------------------------|--------------------|
| Logistic regression         | 0.8518518518518519 |
| K-nearest neighbors         | 0.8641975308641975 |
| Decision tree               | 0.9012345679012346 |
| Method of reference vectors | 0.7777777777777778 |
| Naive Bayesian              | 0.8148148148148148 |

**Table 2**

Classifier precision score comparison

| Classifier model            | Precision score    |
|-----------------------------|--------------------|
| Logistic regression         | 0.9565217391304348 |
| K-nearest neighbors         | 0.9310344827586207 |
| Decision tree               | 1.00               |
| Method of reference vectors | 0.9565217391304348 |
| Naive Bayesian              | 0.7575757575757576 |

The results of calculations on this metric for each of the studied algorithms are given in Table 5.

**Table 3**

Classifier recall comparison

| Classifier model            | Recall |
|-----------------------------|--------|
| Logistic regression         | 0.55   |
| K-nearest neighbors         | 0.675  |
| Decision tree               | 0.8    |
| Method of reference vectors | 0.55   |
| Naive Bayesian              | 0.625  |

One way to estimate the model as a whole without being tied to a specific threshold is the area under the ROC error curve [25]. This curve is a line from (0,0) to (1,1) in the coordinates True Positive Rate (TPR) and False Positive Rate (FPR):



$$TPR = \frac{TP}{TP + FN}; \quad FPR = \frac{FP}{FP + TN}.$$

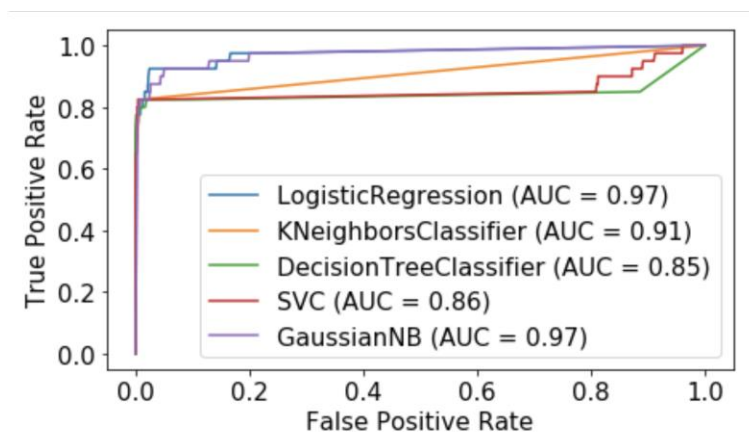
**Table 4**  
Classifier *F* -measure comparison

| Classifier model            | F-measure |
|-----------------------------|-----------|
| Logistic regression         | 0.78      |
| K-nearest neighbors         | 0.58      |
| Decision tree               | 0.89      |
| Method of reference vectors | 0.47      |
| Naive Bayesian              | 0.68      |

**Table 5**  
Classifier logarithmic loss comparison

| Classifier model            | logloss             |
|-----------------------------|---------------------|
| Logistic regression         | 0.22168696855844788 |
| K-nearest neighbors         | 0.3694783949797303  |
| Decision tree               | 0.11823286741946416 |
| Method of reference vectors | 0.4285948286894619  |
| Naive Bayesian              | 0.33992223100658514 |

The area under the curve in this case shows the quality of the algorithm. The closer the area is to 1, the more accurately the model works. Figure 8 shows a graph of ROC-curves for each of the selected models, which can clearly see the efficiency of the algorithms.



**Figure 8:** Graph of ROC-curves for different classifiers

To compare the created classifiers by the area under the ROC curve, it is necessary to calculate it for each of the algorithms. The results of the calculations are given in Table 6. According to the classifiers' evaluation, the decision tree algorithm has the largest share of correctly classified objects with a balanced data set. Besides, this method has the lowest logarithmic loss and the largest area under the ROC curve, which indicates the harmony of sensitivity and specificity.

**Table 6**  
Classifier ROC comparison

| Classifier model            | ROC                |
|-----------------------------|--------------------|
| Logistic regression         | 0.824782324771441  |
| K-nearest neighbors         | 0.7120646495428821 |
| Decision tree               | 0.9                |
| Method of reference vectors | 0.662064649542882  |
| Naive Bayesian              | 0.8107585981715281 |

### 3. Conclusions

After analyzing each of the algorithms and comparing the results, we can say that in this study, among all the proposed methods of machine learning to solve problems of binary classification, the algorithm of decision-making trees best coped. Evaluating it according to the selected metrics and comparing the results with other algorithms, we can talk about its most significant effectiveness in DSS. The developed classifier and its application in DSS can help hospitals and health facilities decide who needs attention in the first place when the system is overcrowded, as well as eliminate delays in providing the necessary care. This study could be scaled up to other diseases to help the health care system respond more effectively to an outbreak or pandemic.

### 4. References

- [1] O. Pysarchuk, A. Gizun, A. Dudnik, T. V. Griga, Domkiv, S. Gnatyuk. "Bifurcation prediction method for the emergence and development dynamics of information conflicts in cybernetic space." CEUR Workshop Proceedings, 2020, 2654, pp. 692–709. <http://ceur-ws.org/Vol-2654/paper54.pdf>.
- [2] O. Barabash, H. Shevchenko, N. Dakhno, Y. Kravchenko and L. Olga. "Effectiveness of Targeting Informational Technology Application." 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC). Conference Proceedings. 05-09 October, 2020, Kyiv, Ukraine. Igor Sikorsky Kyiv Polytechnic Institute. pp. 193 – 196., doi: 10.1109/SAIC51296.2020.9239154.
- [3] S. Toliupa, I. Tereikovskiy, I. Dychka, L. Tereikovska and A. Trush. "The Method of Using Production Rules in Neural Network Recognition of Emotions by Facial Geometry." 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), (2019): 323–327. doi: 10.1109/AIACT.2019.8847847.
- [4] O. Barabash, N. Dakhno, H. Shevchenko and V. Sobchuk. "Unmanned Aerial Vehicles Flight Trajectory Optimisation on the Basis of Variational Inequality Algorithm and Projection Method." 2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD) (2019): 136–139. doi: 10.1109/APUAVD47061.2019.8943869.
- [5] K. Kolesnikova, O. Mezentseva and O. Savielieva. "Modeling of Decision Making Strategies In Management of Steelmaking Processes." 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2019, pp. 455 – 460, doi: 10.1109/ATIT49449.2019.9030524.
- [6] O. Barabash, P. Open'ko, O. Kopiika, H. Shevchenko and N. Dakhno. "Target Programming with Multicriterial Restrictions Application to the Defense Budget Optimization. " *Advances in Military Technology*. 14.2 (2019): 213–229. ISSN 1802-2308, eISSN 2533-4123. doi: 10.3849/aimt.01291, [http://aimt.unob.cz/articles/19\\_02/1291.pdf](http://aimt.unob.cz/articles/19_02/1291.pdf).
- [7] N. Dakhno, O. Barabash, H. Shevchenko, O. Leshchenko and A. Musienko. "Modified Gradient Method for K-positive Operator Models for Unmanned Aerial Vehicle Control." 2020 IEEE 6th

- International Conference on Methods and Systems of Navigation and Motion Control (MSNMC), KYIV, Ukraine, 2020, pp. 81-84, doi: 10.1109/MSNMC50359.2020.9255516.
- [8] V. Tuyrin, O. Barabash, P. Openko, I. Sachuk and A. Dudush. "Informational support system for technical state control of military equipment." 2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD) (2017): 230–232, doi: 10.1109/APUAVD.2017.8308817.
- [9] D. Obidin, V. Ardelyan, N. Lukova-Chuiko and A. Musienko. "Estimation of functional stability of special purpose networks located on vehicles." 2017 IEEE 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD) (2017): 167–170, doi: 10.1109/APUAVD.2017.8308801.
- [10] O. Barabash, N. Lukova-Chuiko, V. Sobchuk and A. Musienko. "Application of Petri Networks for Support of Functional Stability of Information Systems." 2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC) (2018): 1–4, doi: 10.1109/SAIC.2018.8516747.
- [11] M. Prats'ovytyi, O. Svynchuk. "Spread of Values of a Cantor-Type Fractal Continuous Nonmonotone Function." *J Math Sci* 240 (2019): 342–357. <https://doi.org/10.1007/s10958-019-04356-0>.
- [12] A. Rokochinskiy, P. Volk, L. Kuzmych, V. Turcheniuk, L. Volk and A. Dudnik. "Mathematical Model of Meteorological Software for Systematic Flood Control in the Carpathian Region." 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), (2019): 143–148. doi: 10.1109/ATIT49449.2019.9030455.
- [13] V. Mukhin, V. Zavgorodnii, O. Barabash, R. Mykolaichuk, Y. Kornaga, A. Zavgorodnya, V. Statkevych. "Method of restoring parameters of information objects in a unified information space based on computer networks." *International Journal of Computer Network and Information Security*, 12(2) (2020): 11–21. DOI:10.5815/ijcnis.2020.02.02.
- [14] H. Hnatiienko, V. Kudin, A. Onyshchenko, V. Snytyuk and A. Kruhlov, "Greenhouse Gas Emission Determination Based on the Pseudo-Base Matrix Method for Environmental Pollution Quotas Between Countries Allocation Problem," 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, 2020, pp. 1-8, doi: 10.1109/SAIC51296.2020.9239125.
- [15] D. Lukianov, M. Mazeika, V. Gogunskii, K. Kolesnikova. "SWOT analysis as an effective way to obtain primary data for mathematical modeling in project risk management." *CEUR Workshop Proceedings*, 2711, 2020: 79 – 92. <http://ceur-ws.org/Vol-2711/paper7.pdf>.
- [16] Hu Zhenbing, V. Mukhin, Y. Kornaga, O. Herasymenko, Y. Bazaka. "The scheduler for the gridsystem based on the parameters monitoring of the computer components, *Eastern-European Journal of Enterprise Technologies*." 1(2017): 31–39. doi: <https://doi.org/10.15587/1729-4061.2017.91271>.
- [17] Y. Kravchenko, O. Leshchenko, N. Dakhno, O. Trush, O. Makhovych. "Evaluating the Effectiveness of Cloud Services." 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT) (2019): 120–124. doi: 10.1109/ATIT49449.2019.9030430.
- [18] B. Xu, B. Gutierrez, S. Mekaru and et al., "Epidemiological data from the COVID-19 outbreak, real-time case information." *Scientific Data*. 2020 Mar;7(1):106. doi: 10.1038/s41597-020-0448-0.
- [19] C. Sammut, G. I. Webb. "Encyclopedia of Machine Learning and Data Mining." Springer Science+Business Media New York, 2017. doi: <https://doi.org/10.1007/978-1-4899-7687-1>.
- [20] D. T. Larose, C. D. Larose. "Discovering Knowledge in Data: An Introduction to Data Mining." John Wiley & Sons, 2014.
- [21] T.M. Mitchell. "Machine Learning." McGraw Hill, 1997.
- [22] S. Geman, E. Bienenstock and R. Doursat. "Neural Networks and the Bias/Variance Dilemma." *Neural Computation*, 4/1 (1992): 1–58. doi: 10.1162/neco.1992.4.1.1.
- [23] Y. Sasaki. "The truth of the F-measure." School of Computer Science, University of Manchester, 2007.
- [24] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers." *IBM Journal of Research and Development*, 3/3(1959): 210–229. doi: 10.1147/rd.33.0210.
- [25] T. Fawcett. "Introduction to ROC analysis." *Pattern Recognition Letters*, 27/8 (2006): 861–874. doi: 10.1016/j.patrec.2005.10.010.