COMPUTATIONAL PROBLEMS OF ELECTRICAL ENGINEERING

Vol. 11, No. 2, 2021

RESEARCH INTO MACHINE LEARNING ALGORITHMS FOR THE CONSTRUCTION OF MATHEMATICAL MODELS OF MULTIMODAL DATA CLASSIFICATION PROBLEMS

Nataliya Boyko
Lviv Polytechnic National University, Lviv, Ukraine
nataliya.i.boyko@lpnu.ua
https://doi.org/10.23939/jcpee2021.02.001

Abstract: Currently, machine learning algorithms (ML) are increasingly integrated into everyday life. There are many areas of modern life where classification methods are already used. Methods taking into account previous predictions and errors that are calculated as a result of data integration to obtain forecasts for obtaining the classification result are investigated. A general overview of classification methods is conducted. Experiments on machine learning algorithms for multimodal data are performed. It is important to consider all the characteristics of metrics and features when using ML algorithms to predict multimodal data. The main advantages and disadvantages of Gradient Boosting, Random Forest, Logistic Regression and XGBoost algorithms are analyzed in the work.

Key words: classification, binary classification, gradient boosting, random forest, logistic regression, Xgboost.

1. Introdaction

The problem of classification in machine learning occupies a leading place. After all, in modern life, people constantly face problems where they need to determine whether a person or phenomenon belongs to a certain category, but it can usually take a long time, so the best option is to entrust this work to a computer that can process a lot of information in a short period. There are many areas of modern life where classification methods are already used. However, one of the most global sphears using machine learning classification methods is medicine, for example, to diagnose diseases or classify the population at risk. In this regard, we will apply the methods described below. Namely, basing on existing data, we will diagnose a person as being at risk of coronary heart disease or as not having such misfortune.

2. Classification methods

2.1. Gradient Boosting

In our case, this method will take into account all previous predictions and errors that will be received as a result of the iteration of data and the received forecasts, for receiving the result of classification. The principle of method operation is described below [2, 4, 7, 10].

Gradient Boosting (GB) is a machine learning method that can solve regression, classification, and data prediction problems.

GB forms models of individual "weak students" in an iterative way. The general idea is that the model will learn from previous mistakes. The gradient is used to minimize the loss function, i.e. at each iteration of learning the result of the "weak student" is compared with the correct result that is expected. The distance between prediction and expectation represents the error rate of our model. These errors will be used to calculate the gradient. The gradient is a partial derivative of the loss function, and it can be used to find the direction in which the parameters of the model can be changed to minimize the error in the next iteration, using gradient descent [3, 6, 10].

For initialization, the following formula is used:

$$F_0(x) = \arg\min_\gamma \sum_{i=1}^{n} L(y_i, \gamma) . \quad (1)$$

The following steps must be performed on each iteration.

The pseudo-residuals are calculated using formula (2):

$$r_{im} = \left[\frac{\partial L(y_i, F(x_i))}{\partial L(x_i)} \right] F(x) = F_{m-1}(x) \; for \; i = 1, ..., n \quad (2)$$

Let us define the basic "student" (for example, a tree) for the pseudo-residuals and teach him in the training sample [1, 7, 9].

The one-dimensional optimization problem is solved using formula (3):

$$\gamma = \arg\min_\gamma \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i)) + \alpha h_m(x_i)), \quad (3)$$

where α is the training rate.

The last step is to update our model. To do this, formula (4) is used:

$$F_m = F_{m-1}(x) + \gamma h_m(x) . \quad (4)$$

2.2. Random Forest

In this case, the signs that may affect the development of coronary heart disease are randomly selected

and decision trees are built. To get the classification, all these trees will be combined for obtaining the answer [10, 11].

Random Forest (RF) builds many individual decision trees, and then the forecasts of all trees are combined to obtain the final class forecast.

The classifier using N trees will look as follows:

$$a(x) = \frac{1}{N}\sum_{i=1}^{N} b_i(x) \quad (5)$$

where $b_i(x)$ is the decision tree.

In classification problems, the solution by majority vote is chosen.

2.3. Logistic Regression

Using this method, the approximate importance of each symptom will be deduced which will help to more accurately classify whether a person is at risk of developing coronary heart disease [1, 13].

Logistic Regression (LR) is used when the dependent variable is categorical.

First, a matrix of weights with random initialization is created (in most cases, these are zero arrays) and then it is multiplied by the signs, and an offset is added. Let us calculate the value of the function using the obtained weights and offset according to formula (6).

$$a = (b + w_1x_1 + w_2x_2 + ... + w_nx_n) * \alpha, \quad (6)$$

where α is the training rate (showing how fast we should move to a minimum: it is best to choose a small enough value so that we do not get stuck at some point) [6].

Then the obtained value is transferred to the communication function, using formula (7):

$$\overline{y_i} = \frac{1}{1+e^{-a}}. \quad (7)$$

The last step is to use a gradient and update the weights. Firstly, it is neccessary to calculate gradient for all properties by formula (8). Using the formula (9), updated weights can be obtained.

$$dw_i = \sum_{i=1}^{i=n} (\overline{y_i} - y_i) \quad (8)$$

$$w_i = w_i - dw_i. \quad (9)$$

2.4. XGBoost

In our case, this method will take into account all previous predictions and errors to obtain the classification result, as in the Gradient Boosting method. XGBoost is a modification of the Gradient Boosting method, but differences are observed in its implementation [5, 8].

To begin with, as in GB, we use formula (1) for initialization. The following steps must be performed on each iteration. The pseudo-residuals are calculated using formula (2). Next, the best division of the node is determined by the following formula:

$$G_{ain} = \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_R + G_L)^2}{H_L + H_R + \lambda} - \gamma, \quad (10)$$

where G_L are residuals in the left leaf, G_R are residuals in the right leaf, H_R is the number of residuals in the right leaf, H_L is the number of residuals in the left leaf, γ and λ are hyper parameters.

If G_{ain} is a non-negative number, then the division is suitable, if it is negative, this option is cut off [5].

When the best tree has been built, it is necessary to calculate the output value of each leaf by formula (11).

$$O_{utput_i} = \frac{\sum r_{im}}{H_i + \lambda}. \quad (11)$$

Then all that remains is to update the model.

3. Analysis of the selected dataset

To conduct experiments, it is necessary to prepare and analyze a data set. To do this, the "Replication Dataset for: South African Heart Disease" is used located in free access from Harvard [1].

This data set contains information on bad habits, factors that influence the development of coronary heart disease in South Africa, as well as the classification of people by the presence of the disease.

Let us analyze all the features of the dataset in more detail:

- CLASS – whether a person has ischemic heart disease. If the result is negative, the value is -1, otherwise 1;
- sbp – indicator of systolic blood pressure of the person;
- tobacco – the cumulative effect of smoking;
- ldl – low-density lipoproteins;
- adiposity – an indicator of human obesity;
- famhist – information about the presence of heart disease in the family and their course. 1 – recorded at the moment, 0 – no information.
- typeA – human behavior;
- obesity – body mass index;
- alcohol – the amount of habitual alcohol consumption;
- age – the age of the person at the time of data entry into the sample.

4. Theoretical Algorithms

4.1. Gradient Boosting

To implement GB, an algorithm consisting of the following items is used.

1. The value of the prediction function is initialized with a constant value. The logarithm of the ratio of positive to negative values in our training data set is calculated, as well as the probability of attributing the record to a person who is the at-risk group by a following formula:

$$F_0 = \frac{e^{(true/\,false)}}{1+e^{(true/\,false)}}. \quad (12)$$

This step is to repeat for each iteration $m=1,\ldots,M$.

2. Pseudo-residuals are calculated by formula (13):

$$r_{it} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\overline{f(x)}}, for\ i=1,...,n \quad (13)$$

where n is the number of records in our data set.

3. A new tree of the base solution $h_t(x)$ is built based on the pseudo-residuals $\{(x_i, r_{it})\}\, i=1,...,n$.

4. The output values for each leaf of the current tree are calculated by formula (14):

$$\frac{\sum_{i=1}^{n} r_{im}}{\sum_{i=1}^{n} (F_{m-1} * (1-F_{m-1}))}. \quad (14)$$

5. Current approximation is updated by formula (15):

$$\overline{F(x)} \leftarrow \overline{F(x)} + \overline{F_t(x)} = \sum_{i=0}^{t} \overline{F_i(x)}. \quad (15)$$

6. The final model is composed (formula 16):

$$\overline{F(x)} = \sum_{i=0}^{M} \overline{F_i(x)}. \quad (16)$$

The following diagram illustrates GB operation:

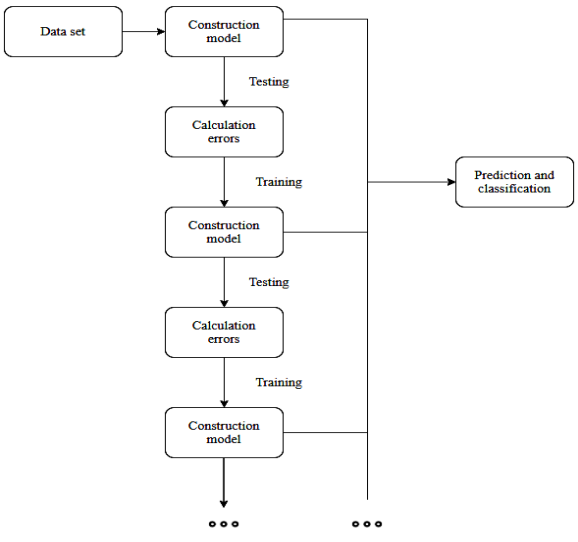

Fig. 1. Illustration of the Gradient Boosting algorithm.

4.2. Random Forest

The operation of the algorithm can be illustrated using the following steps.

1. First, a random set of data from a given one is selected.
2. Next, the algorithm will construct a decision tree for each sample and obtain results for forecasting from each decision tree.
3. At this stage, voting will be conducted for each predicted result.
4. In the end, the result of the forecast is chosen by the largest number of votes as the final result of the forecast.

The following diagram illustrates Random Forest algorithm operation.

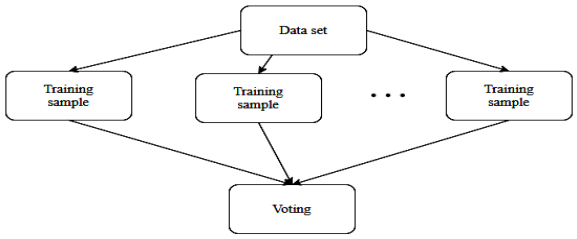

Fig. 2. Illustration of the Random Forest algorithm.

4.3. Logistic Regression

To implement the LR algorithm, the following steps are performed.

1. The sigmoid function is implemented.
2. Model parameters are initialized.
3. The loss function is calculated and the parameters are updated using a gradient.
4. A model for prediction and classification is formed.

4.4. XGBoost

In this algorithm, decision trees are created in sequential form. Scales play an important role in XGBoost. Scales are assigned to all independent variables, which are then fed into the decision tree, which predicts the results [4]. The weight of variables predicted incorrectly by the tree increases, and these variables are fed to the tree of the second decision. These individual classifiers/forecasts are then compiled to provide a stronger and more accurate model. It can work on regression, classification, ranking, and user-defined forecasting problems.

5. Practical application of algorithms

5.1. Gradient Boosting

To depict the operation of the algorithm, we will select several records from our data set, which are listed in Table 1. For ease of visual perception, we replace the class value for people who do not have coronary heart disease with 0 (instead of -1).

Table 1

Table input data

No.	class	sbp	tobacco	ldl	adipo-sity	fam-hist	Type A	obe-sity	alco-hol	age
1	1	160	1.2	5.73	23.11	1	49	25.3	97.2	52
2	1	144	0.01	4.41	28.61	0	55	28.87	2.06	63
3	0	138	0.08	3.48	32.28	1	52	29.14	3.81	46
4	0	132	6.2	6.47	36.21	1	62	30.77	14.14	45
5	1	170	7.5	6.41	38.03	1	51	31.99	24.24	58

The last record will be used to present the classification of the trained model.

First, the value of the ratio of positive to negative responses (log odds) is initialized. In our case it is: $\log(2/2) = \log 1 = 0$, then the probability that the person is at risk group is calculated as follows:

$$\frac{e^0}{1+e^0} = 0.5 \qquad (17)$$

At the next step, the pseudo-residuals are found for each line in the training sample (the first four lines):

$$r_{11} = 1-0.5 = 0.5 \quad r_{21} = 1-0.5 = 0.5$$
$$r_{31} = 0-0.5 = -0.5 \quad r_{41} = 0-0.5 = -0.5 \qquad (18)$$

Now we need to build a decision tree. In this case, the number of leaves will be limited to three. In the leaves, we will write the pseudo-residuals, on their side, for convenience, the record number, and at the bottom – the output value.

The number of trees will be also limited to two for demonstration.

Because we do the calculations ourselves, we can visually determine which properties allow us to best divide the data.

So, the first tree will look like this:

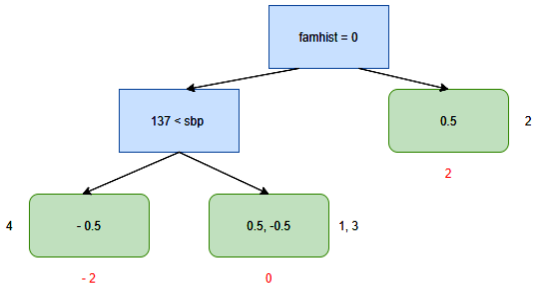

Fig. 3. The first decision tree obtained.

First, the data are taken according to family history. Only the second person does not have it, so the record goes to the right branch. The output value in this branch will be equal:

$$\frac{0.5}{0.5\cdot(1-0.5)} = 2. \qquad (19)$$

The remaining values we can share more. For example, in terms of sbp. To do this, the average sbp is calculated for the remaining people and they are written to the appropriate leaves.

The fourth person gets to the left leaf, and the output value for a tree, in this case, will be equal:

$$\frac{-0.5}{0.5\cdot(1-0.5)} = -2 .. \qquad (20)$$

The first and third entries are included in the right leaf. Its output value is:

$$\frac{0.5+(-0.5)}{0.5\cdot(1-0.5)+0.5\cdot(1-0.5)} = 0. \qquad (21)$$

Now our model should be updated. That is, it will have the form of the sum of the initial probability and the initial value of the obtained tree, which is shown in Fig. 3.

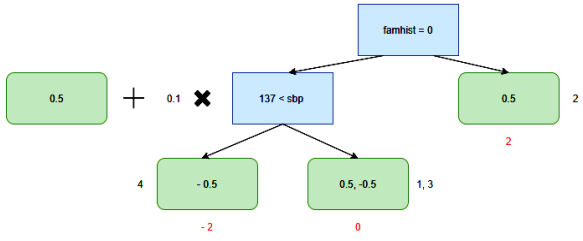

Fig. 4. The obtained model after the first constructed tree.

The probability values need to be updated for each record.

Table 2

Updating the probability value

No.	1	2	3	4
Probability	0.5 + 0.1 · 0 = =0.5	0.5 + 0.1 · 2 = =0.7	0.5 + 0.1 ·0 = =0.5	0.5 + 0.1 ·(-2) = =0.3

Next, similar actions will be performed to build two more trees and get the final model.

The pseudo-residuals are calculated by formula 22:

$$r_{12} = 1-0.5 = 0.5 \quad r_{22} = 1-0.7 = 0.3$$
$$r_{32} = 0-0.5 = -0.5 \quad r_{42} = 0-0.3 = -0.3 \qquad (22)$$

A second decision tree is constructed.

The output value for a leaf that includes the last two people is:

$$\frac{-0.5+(-0.3)}{0.5\cdot(1-0.5)+0.3\cdot(1-0.3)} = -1.74. \qquad (23)$$

For the second one it is: $\frac{0.3}{0.7\cdot(1-0.7)} = 1.43$.

The last one is equal to $\frac{0.5}{0.5\cdot(1-0.5)} = 2$.

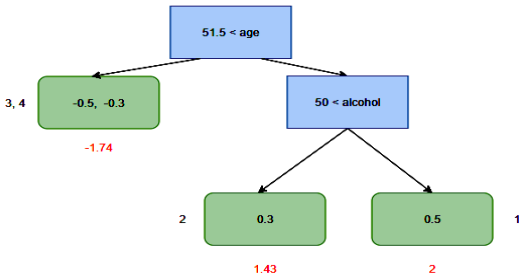

Fig. 5. Second built tree.

Update probabilities are given is Table 3.

Table 3

Updating the probability value

No.	1	2	3	4
Proba-bility	$0.5 + 0.1 \cdot 0 + +0.1 \cdot 2 = 0.7$	$0.5 + 0.1 \cdot 2 + +0.1 \cdot 1.43 = =0.7$	$0.5 + 0.1 \cdot 0 + + 0.1 \cdot (-1.74) = =0.326$	$0.5 + 0.1 \cdot (-2) + + 0.1 \cdot (-1.74) = =0.126$

The updated model will look like this:

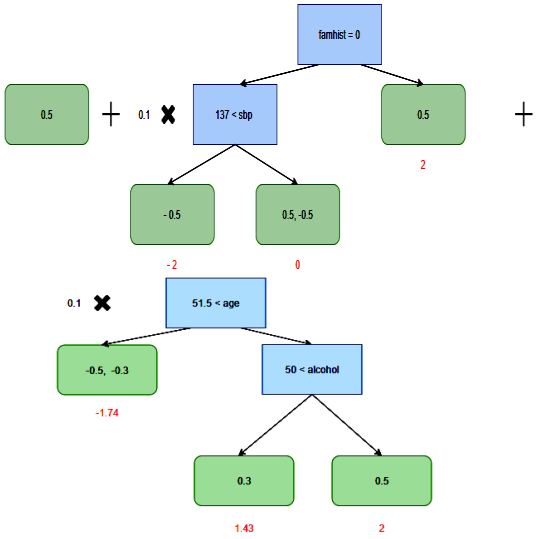

Fig. 6. The obtained model after the second constructed tree.

Our model is used to classify the test record.

So, we calculate whether a person is at risk:

$$F = 0.5 + 0.1 \cdot 0 + 0.1 \cdot 1.43 = 0.643 > 0.5 \qquad (24)$$

It follows that the probability of detecting coronary heart disease for this person is 64.3 %, which means that the person falls into the class marked 1.

5.2. Random Forest

For this method, we will use the same data as in the previous section.

In the first step, we need to randomly generate a subset that we will use to build the tree. For example, it might look like this (See Table 4).

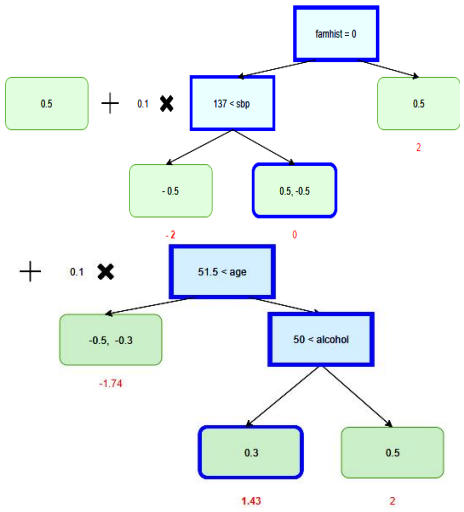

Fig. 7. A model applied to determine the likelihood that a person has a coronary heart disease.

Table 4

A subset used to build the first tree

No.	class	sbp	tobacco	ldl	adi-posity	famhist	typeA	obesity	alcohol	age
1	1	160	1.2	5.73	23.11	1	49	25.3	97.2	52
2	1	144	0.01	4.41	28.61	0	55	28.87	2.06	63
3	1	160	1.2	5.73	23.11	1	49	25.3	97.2	52
4	0	132	6.2	6.47	36.21	1	62	30.77	14.14	45

Let us create the first decision tree. To depict the operation of the algorithm, the number of the trees is limited to three.

Therefore, we will use only two properties at each step.

At the root, two properties are randomly selected. We leave one of them whose values vary the least. In this case, our choice was made between famhist and tobacco. The right sheet includes a person who does not have a family history of the disease, i.e. the result of this leaf will be 1.

Next, the sample using tobacco is divided. The first and third entries with output 1 go to the left sheet, and that with 0 value goes to the right.

The resulting tree is shown in Fig. 8.

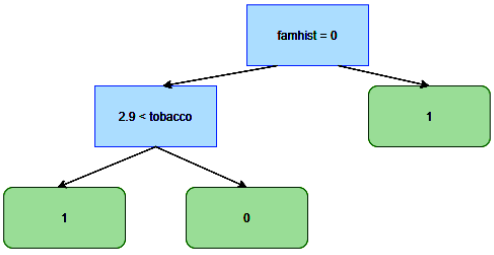

Fig. 8. The first decision tree for the Random Forest algorithm.

As a result, a tree which will be used for classification in the future is obtained.

Then, the construction of the second tree begins. To do this, a new sample is formed (Table 5).

Table 5

A subset used to build the second tree

No.	class	sbp	tobacco	ldl	adipo-sity	famhist	type A	obesity	alcohol	age
1	1	160	1.2	5.73	23.11	1	49	25.3	97.2	52
2	0	132	6.2	6.47	36.21	1	62	30.77	14.14	45
3	0	138	0.08	3.48	32.28	1	52	29.14	3.81	46
4	0	132	6.2	6.47	36.21	1	62	30.77	14.14	45

The tree will look like that in Fig. 9.

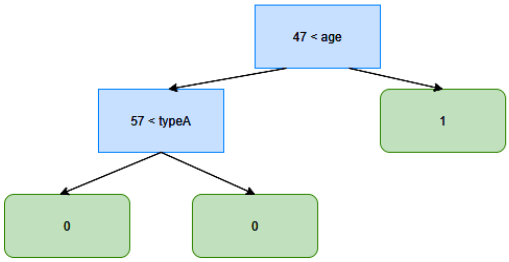

Fig. 9. The second solution tree for the Random Forest algorithm.

Let us do these steps again and build the last tree.

Table 6

A subset used to build the last tree

No.	class	sbp	tobacco	ldl	adipo-sity	fam-hist	Type A	obesity	alco-hol	age
1	1	160	1.2	5.73	23.11	1	49	25.3	97.2	52
2	1	144	0.01	4.41	28.61	0	55	28.87	2.06	63
3	1	144	0.01	4.41	28.61	0	55	28.87	2.06	63
4	0	132	6.2	6.47	36.21	1	62	30.77	14.14	45

The following tree is obtained:

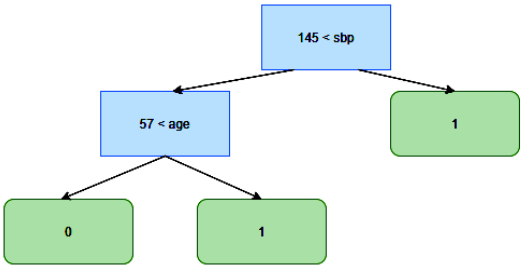

Fig. 10. The third solution tree for the Random Forest algorithm.

We can now proceed to the classification. So we need to check the entry in each tree and vote.

The original value from each tree will be searched as follows:

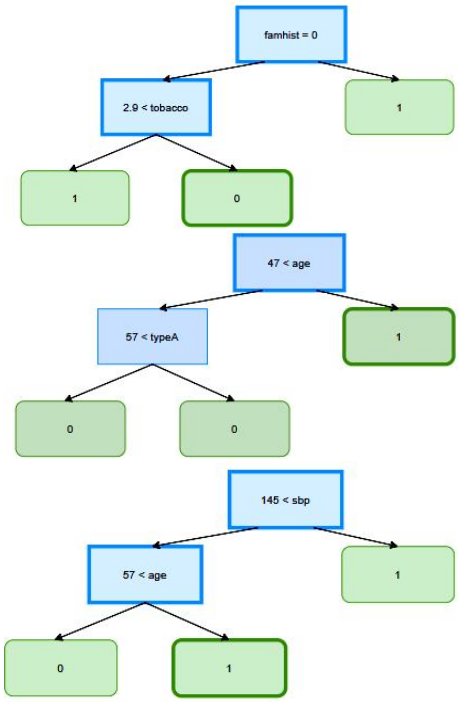

Fig. 11. The voting of the Random Forest algorithm.

As a result, we obtain a negative answer from the first tree and two positive – from the second and third one. Because the classification is performed by majority vote, the person will be classified as being at risk for coronary heart disease.

5.3. Logistic regression

At the first step, the bias and weights with zeros are initialized.

$$b = 0, w_1 = 0,\ w_2 = 0,\ w_3 = 0,\ w_4 = 0,\ w_5 = 0,\ w_6 = 0,$$
$$w_7 = 0,\ w_8 = 0,\ w_9 = 0.$$

Next, the value of the function for each record from the training sample should be calculated.

$$a_1 = (0 + 0 \cdot 160 + 0 \cdot 1.2 + 0 \cdot 5.73 + 0 \cdot 23.11 + 0 \cdot 1 + 0 \cdot 49 + $$
$$+ 0 \cdot 25.3 + 0 \cdot 97.2 + 0 \cdot 52) \cdot 0.0001 = 0$$

$$a_2 = (0 + 0 \cdot 144 + 0 \cdot 0.01 + 0 \cdot 4.41 + 0 \cdot 28.61 + 0 \cdot 0 + 0 \cdot 55 + $$
$$+ 0 \cdot 28.87 + 0 \cdot 2.06 + 0 \cdot 63) \cdot 0.0001 = 0$$

$$a_3 = (0 + 0 \cdot 138 + 0 \cdot 0.08 + 0 \cdot 3.48 + 0 \cdot 32.28 + 0 \cdot 1 + 0 \cdot 52 + $$
$$+ 0 \cdot 29.14 + 0 \cdot 3.81 + 0 \cdot 46) \cdot 0.0001 = 0$$

$$a_4 = (0 + 0 \cdot 132 + 0 \cdot 6.2 + 0 \cdot 6.47 + 0 \cdot 36.21 + 0 \cdot 1 + 0 \cdot 62 + $$
$$+ 0 \cdot 30.77 + 0 \cdot 14.14 + 0 \cdot 45) \cdot 0.0001 = 0$$

The predicted value for each record is also calculated:

$$\overline{y}_1 = \frac{1}{1+0} = 0.5 \qquad \overline{y}_2 = \frac{1}{1+0} = 0.5$$

$$\overline{y}_3 = \frac{1}{1+0} = 0.5 \qquad \overline{y}_4 = \frac{1}{1+0} = 0.5$$

Then, we calculate the gradients for each property.

$$dw_1 = \frac{1}{4}*(0.5*160+0.5*144-0.5*138-0.5*132) = 4.25$$

$$dw_2 = \frac{1}{4}*(0.5*1.2+0.5*0.01-0.5*0.08-0.5*6.2) = -0.63$$

$$dw_3 = \frac{1}{4}*(0.5*5.73+0.5*4.41-0.5*3.48-0.5*6.47) = 0.02$$

$$dw_4 = \frac{1}{4}*(0.5*23.11+0.5*28.61-0.5*32.28-0.5*36.2) = -4.72$$

$$dw_5 = \frac{1}{4}*(0.5*1+0.5*0-0.5*1-0.5*1) = -0.125$$

$$dw_6 = \frac{1}{4}*(0.5*49+0.5*55-0.5*52-0.5*62) = -1.25$$

$$dw_7 = \frac{1}{4}*(0.5*25.3+0.5*28.87-0.5*29.14-0.5*30.77) = -0.94$$

$$dw_8 = \frac{1}{4}*(0.5*97.2+0.5*2.06-0.5*3.81-0.5*14.14) = 10.2$$

$$dw_9 = \frac{1}{4}*(0.5*52+0.5*63-0.5*46-0.5*45) = 3.$$

The last step is to update the scales and the shift.

$w_1 = 0-4.25 = -4.25$ $w_2 = 0-(-0.65) = 0.65$

$w_3 = 0-0.02 = -0.02$ $w_4 = 0-(-4.72) = 4.72$

$w_5 = 0-(-0.125) = 0.125$ $w_6 = 0-(-1.25) = 1.25$

$w_7 = 0-(-0.94) = 0.94$ $w_8 = 0-10.2 = -10.2$

$w_9 = 0-3 = -3$ $b = 0-0 = 0$

To classify the last record, it is substituted in the regression equation and the result is passed to the related function.

$$a = (0+(-4.25)*170+0.65*7.5+(-0.02)*6.41+$$
$$+4.72*38.03+0.125*1+1.25*51+0.94*31.99+$$
$$+(-10.2)*24.26+(-3)*58)*0.0001 = -0.0866$$

$$\overline{y_{pr}} = \frac{1}{1+e^{-(-0.0866)}} = 0.478 < 0.5$$

Because the probability obtained is less than 0.5, the person will be classified as being not prone to the coronary heart disease, which, as we can see, is not true. This behavior can be explained by the fact that only one iteration and small sample size were performed.

5.4. XGBoost

Let us define the initial probability as 0.5, because XGBoost is used to solve the classification problem.

For each sample, the residue is calculated according to the resulting formula (25).

$$r_{12} = 1-0.5 = 0.5 \quad r_{22} = 1-0.5 = 0.5$$
$$r_{32} = 0-0.5 = -0.5 \quad r_{42} = 0-0.5 = -0.5 \qquad (25)$$

All the pseudo-residuals are put to the root and a few trees are built from which the best one is to choose. To begin with, we will build a tree, considering only the sbp indicator. Also, to demonstrate the operation of the algorithm, the number of sheets is limited to three.

So, the first division will look like this (the balances being sorted by sbp growth):

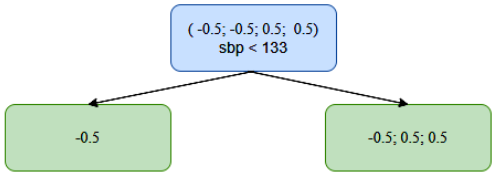

Fig. 12. Image of the first division.

At this stage, the Gain for the root is to be calculated according to formula (26). Arbitrary values γ and λ are chosen equal to 1 and 0, respectively.

$$G_L = -0.5, \quad H_L = 1,$$
$$G_R = -0.5+0.5+0.5 = 0.5, \quad H_R = 4$$

$$G_{ain} = \frac{(-0.5)^2}{1+0}+\frac{0.5^2}{4+0}-\frac{(-0.5+0.5)^2}{4+1+0}-1 = \qquad (26)$$
$$= -0.6875$$

The process continues with calculating the amplifications corresponding to other permutations.

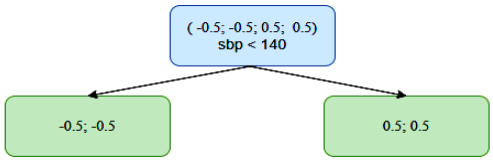

Fig.12. Image of the second division.

$$G_L = -0.5+(-0.5) = -1,$$
$$H_L = 2, \quad G_R = 0.5+0.5 = 1, \quad H_R = 2$$

$$G_{ain} = \frac{(-1)^2}{2+0}+\frac{1^2}{2+0}-\frac{(-1+1)^2}{2+2+0}-1 = 0$$

Similarly, it is done for the remaining records.

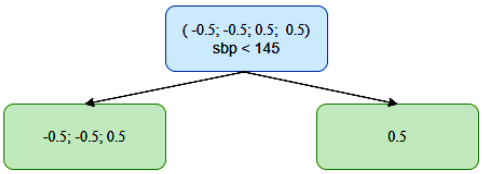

Fig. 14. Image of the third division.

$$G_L = -0.5+(-0.5)+0.5 = -0.5, \quad H_L = 3,$$
$$G_R = 0,5, \quad H_R = 1$$

$$G_{ain} = \frac{(-0.5)^2}{3+0}+\frac{0.5^2}{1+0}-\frac{(-0.5+0.5)^2}{3+1+0}-1 = -0.667\,.$$

As it can be seen, the largest Gain is observed for the third division, so we leave it.

Since the number of leaves is limited to 3, and we have only two leaves, we continue to check which of the vertices is better to divide in the same way.

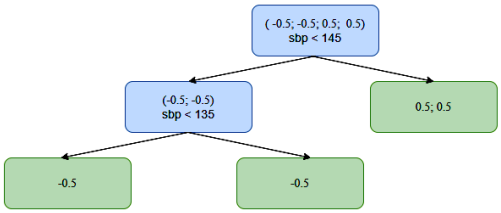

Fig. 15. Picture of the division of the left leaf.

$$G_L = -0.5,\ \ H_L = 1,\ G_R = -0,5,\ H_R = 1,$$

$$G_{ain} = \frac{(-0.5)^2}{1+0} + \frac{(-0.5)^2}{1+0} - \frac{(-0.5+(-0.5))^2}{1+1+0} - 1 = -1$$

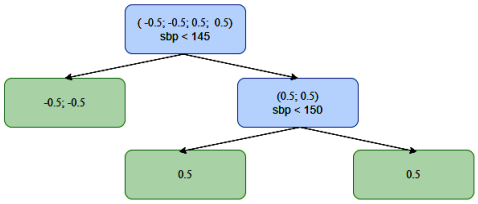

Fig. 16. Picture of the division of the right leaf.

$$G_L = 0.5, \quad H_L = 1, \quad G_R = 0.5, \quad H_R = 1,$$

$$G_{ain} = \frac{0.5^2}{1+0} + \frac{0.5^2}{1+0} - \frac{(0.5+0.5)^2}{1+1+0} - 1 = -1.$$

As we can see, in both cases Gain is a negative value, so it makes no sense to divide any of the leaves.

Therefore, the tree that enters will remain as follows:

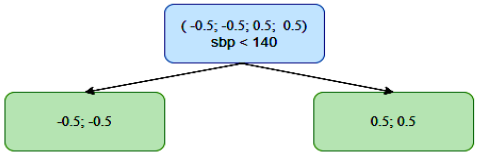

Fig. 17. The tree included in the model for classification.

Since the decision tree must return one scalar, we decide that output values for each leaf are to be calculated by formula (27).

$$O_{output_1} = \frac{-0.5+(-0.5)}{2+0} = -0.5$$
$$O_{output_2} = \frac{0.5+0.5}{2+0} = 0.5 \qquad (27)$$

The first prediction is the sum of the initial probability and prediction made by the tree multiplied by the learning rate.

It is assumed that the degree is 0.4. In this case, if the last record is passed to the model, the following result is obtained:

$$P_{prediction} = 0.5 + 0.4 * 0.5 = 0.7 > 0.5. \qquad (28)$$

That is, even at the stage when only the indicator of systolic blood pressure of the person is taken into account, it is already possible to make predictions (not necessarily accurate, because there is little data for it).

In the future, these residuals should be used to build another decision tree and this process is to be repeated until we reach the maximum number of trees.

Once we have completed the learning model, the predictions made by the XGBoost model as a whole are the sum of the initial predictions and predictions made in each individual decision tree multiplied by the learning factor.

6. Conducting experiments

6.1. Gradient Boosting

Having launched our own implementation of GB, where 80 % of all records are used for training and 20 % for classification respectively, we obtain the following result (Fig. 18).

```
Training Finished
0.5866013383510299
True - True:  0
False - False:  40
True - False:  18
False - True:  35
```

Fig. 18. The results of the Gradient Boosting algorithm.

Thus, we can see that GB classified the records with an accuracy of 59 %. We received the most correct answers when a person is actually healthy, and mistakes if he does not have coronary heart disease, but our algorithm attributed it to people at risk.

6.2. Random Forest

When using the RF method, one should keep in mind that different results will always be obtained, because decision trees are built on randomly selected records and properties. In Fig. 19, there are two different versions of the program given.

```
Accuracy: 0.6451612903225806
True - True:  14
False - False:  46
True - False:  19
False - True:  14
```

```
Accuracy: 0.7204301075268817
True - True:  18
False - False:  49
True - False:  15
False - True:  11
```

Fig. 19. The results of the Random Forest algorithm.

Thus, it can be seen that the classification correctness ranges from 65 % to 72 %. In the case of Random Forest, most incorrect answers can lead to the late detection of the disease.

6.3. Logistic regression

Running the implemented LR algorithm, we obtain the following results:

```
LR classification accuracy: 0.7338129496402878
True - True:  16
False - False:  86
True - False:  31
False - True:  6
```

Fig. 20. The results of the Logistic Regression algorithm.

Therefore, a classification accuracy of 73 % is obtained. We can conclude that this algorithm also cannot be used for medical purposes without improvement.

6.4. XGBoost

Running the implemented XGBoost algorithm, we obtain the following results:

```
-- XGBoost --
Accuracy:  0.7419354838709677
True - True:  14
False - False:  55
True - False:  20
False - True:  4
```

Fig. 21. The results of the XGBoost algorithm.

Therefore, a classification accuracy of 74 % is obtained. We can conclude that this algorithm also cannot be used for medical purposes without improvement.

7. Advantages and disadvantages

7.1. Gradient Boosting

Advantages of the GB algorithm:

- predictive accuracy that cannot be achieved using other methods;
- high flexibility allowing the optimization of loss functions and provides several options for setting hyper parameters, making the function more flexible;
- no preprocessing required since it often works well with categorical and numeric values, without data preparation.
- missing data processing.

Disadvantages:

- continuation of improving error minimization, which can take a long time. Cross-checking should be used to neutralize it.
- high costs of computations: GB often requires a lot of trees (> 1000), process being limited in time and memory.
- high flexibility leading to many parameters that interact and strongly influence the behavior of the method (number of iterations, tree depth, regularization parameters, etc.).
- less interpretive, although this is easily solved with the help of various tools (importance variable, graphs of partial dependence, etc.).

7.2. Random Forest

Advantages of the RF algorithm:

- Predicted efficiency can compete with the best learning algorithms.
- It provides a reliable assessment of the function.
- It offers effective test error estimating without incurring the cost of re-teaching the model associated with cross-validation.

On the other hand, the algorithm has the following disadvantages:

- The ensemble model is inherently less interpreted than a single decision tree.
- Teaching a large number of deep trees can be costly (but can be done in parallel) and uses a lot of memory.
- Forecasts are slower which can cause problems.
- The results will always be different because the data is selected randomly.

7.3. Logistic Regression

Among the advantages of logistic regression are the following points:

- Logistic regression works well when the data set is linearly separated.
- Logistic regression is less prone to over-placement but may predominate in large dimensional datasets. Regularization techniques (L1 and L2) should be considered to avoid readjustment in these scenarios.
- Logistic regression gives not only a measure of how relevant the predictor is (the size of the coefficient) but also the direction of its association (positive or negative).
- Logistic regression is easier to be implemented, interpreted, and provides very efficient teaching.

Among the disadvantages are:

- The main limitation of logistic regression is the assumption of linearity between the dependent variable and the independent variables. In the real world, data is rarely linearly separated. In most cases, data would be confusing.
- If the number of observations is less than the number of features, logistic regression should not be used, otherwise, it may lead to over-observation.
- Logistic regression can only be used to predict discrete functions. Therefore, the dependent variable logistic regression is limited to a set of discrete numbers. This limitation itself is problematic, because it prohibits the prediction of continuous data.

7.4. XGBoost

XGBoost is an efficient algorithm being easy to use and providing high performance and accuracy compared to other algorithms. XGBoost is also known as an adjustable version of GB. Let us consider some of the benefits of the XGBoost algorithm:

- Regularization: XGBoost has built-in regularization of L1 (Lasso Regression) and L2 (Ridge Regression), which does not allow the model to overflow. That is why XGBoost is also called the adjustable GB (Gradient Boosting) form.
- Parallel processing: XGBoost uses the power of parallel processing and, therefore, it is much faster than GB. Several CPU cores are used to run the model.
- Missing value handling: XGBoost has a built-in ability to handle missing values. When XGBoost encounters a missing value in a node, it tries to split both the left and right branches and learns the path that results in greater losses for each node. It then does the same, working on the test data.
- Cross-validation: XGBoost allows the user to cross-check each iteration of the acceleration process, and thus easily obtain the exact optimal number of accelerated iterations per run. This is unlike GBM, where we need to run a grid search and can only check limited values.

Among the disadvantages are: complex interpretation, rigid visualization; excess fulfillment is possible if the settings are not configured properly. It is difficult to configure because of too many hyper settings.

8. Conclusion

In this article, the work of algorithms Gradient Boosting, Random Forest, Logistic Regression, XGBoost was analyzed. The advantages and disadvantages of each algorithm were also analyzed. Each algorithm solves the classification problem (in this case, a binary one).

Also, after analyzing all the results, we can conclude that the algorithms are not good enough to classify the susceptibility of people to the development of coronary heart disease, because they are not accurate enough and require observation by a qualified specialist. Another possible option is to use initial classification algorithms with more in-depth training on a larger training and test sample so that the model can be more accurate, but the possibility of model overflow should also be considered, leading to more significant problems.

References

[1] "Dataset South African Heart Disease", *https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/76SIQD*

[2] "Open Machine Learning Course: Gradient Boosting Machines", *http://uc-r.github.io/gbm_regression*

[3] P. Rathi and A. Sharma, "A review paper on prediction of diabetic retinopathy using data mining techniques", in *International journal of innovative research in technology*, Vol. 4, pp. 292–297, 2017.

[4] N. Boyko and K. Boksho, "Application of the Naive Bayesian Classifier in Work on Sentimental Analysis of Medical Data", in *Proc. 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden*, pp. 230–239, 2020.

[5] C. Maklin, "XGBoost Python Example", *https://towardsdatascience.com/xgboost-python-example-42777d01001e, last accessed 2020/12/21.*

[6] R. M. V. Humphris, *Testing Algorithm Fairness Metrics for Binary Classification Problems by Supervised Machine Learning Algorithms*, Vrije Universiteit Amsterdam, 2020.

[7] R. S. Brid, "Boosting", *https://medium.com/ greyatom / boosting-ce84639a805d, last accessed 2018/11/01.*

[8] J. Brownlee, "A gentle introduction to xgboost for applied machine learning", *https://machine learningmastery.com/ gentle-introduction-xgboost-applied-machine-learning/, last accessed 2019/11/18.*

[9] N. Boyko and R. Hlynka, "Application of Machine Algorithms for Classification and Formation of the Optimal Plan", in *Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021),* Vol. 1, Main Conference Lviv, Ukraine, April 22–23, pp. 1853–1865, 2021.

[10] J. Brownlee, "A gentle introduction to the bootstrap method", *https://machinelearningmastery.com/ a-gentle-introduction-to-the-bootstrap-method/, last accessed 2020/06/29.*

[11] A. Chakure, "Decision tree classification", *https://towardsdatascience. com/decision-tree-classification-de64fc4d5aac, last accessed 2019/11/28.*

[12] C. Cortes and V. N. Vapnik, "Support-vector networks", *Machine Learning*, Vol. 20(3), pp. 273–297, 1995. doi: https://doi.org/10.1023/ A:1022627411411.

[13] N. Boyko, "Information system of catering selection by using clustering *analysis*", in *2018 IEEE Ukraine Student, Young Professional and Women in Engineering Congress (UKRSYW)* October 26, Kyiv, Ukraine, pp. 7–13, 2018.

[14] "DataCamp. Hyperparameter tuning with randomizedsearchcv", *https: //campus.datacamp. com/ courses/supervised-learning-with-scikit-learn/ fine-tuning-your-model?ex=11, last accessed 2020/06/11.*

[15] "DeZyre. Metrics for evaluating machine learning algorithms", *https://www.dezyre.com/data-science-in-python-tutorial/ performance-metrics-for-machine-learning-algorithm, last accessed 2019/11/28.*

ДОСЛІДЖЕННЯ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПОБУДОВИ МАТЕМАТИЧНИХ МОДЕЛЕЙ ЗАДАЧ КЛАСИФІКАЦІЇ МУЛЬТИМОДАЛЬНИХ ДАНИХ

Наталія Бойко

Сьогодні алгоритми машинного навчання (ML) все більше інтегруються у повсякденне життя. Можна навести безліч сфер сучасного життя, де вже застосовуються методи класифікації. Досліджуються методи, які враховують попередні передбачення та помилки, які обчислюються в результаті інтегрування даних задля отримання прогнозів, для отримання результату класифікації. Проведено загальний огляд методів класифікації. Здійснено експерименти над алгоритмами машинного навчання для мультимодальних даних. Важливо враховувати всі характеристики метрик та ознак під час використання алгоритмів ML для прогнозування мультимодальних даних. В роботі проаналізовано основні переваги та недоліки алгоритмів Gradient Boosting, Random Forest, Logistic Regression та XGBoost.

Boyko Nataliya Ivanivna, PhD, Associate Professor of the Artificial Intelligent Systems Department of Lviv Polytechnic National University. Scientific interests: machine learning, data visualization, intellectual data analysis, system analysis.

Received: 04.08.2021. Accepted: 25.09.2021