

Classification methods in the diagnosis of breast cancer

Anna Magdalena Ogłoszka, Łukasz Smaga

Faculty of Mathematics and Computer Science, Adam Mickiewicz University,
Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland,
e-mail: a.m.ogloszka@gmail.com, ls@amu.edu.pl

SUMMARY

This paper concerns the problem of diagnosing the type of cancer with the use of machine learning and statistical methods. Nowadays, the problem of neoplasms, in particular breast cancer, is one of humanity's greatest challenges. The identification of cancer and its type is extremely important. In solving this problem, classification methods can be used as objective tools that may be helpful for doctors making a diagnosis. For this reason, we discuss many efficient classifiers in the context of cancer detection. In addition, we consider the topic of data set transformations to deal with the problem of data unbalance, as well as measures of classification quality. In the experimental part, an attempt will be made to find the best classifier and to improve the quality of the original data set to obtain the highest values of classification quality measures for a particular data set.

Key words: cancer, breast cancer, classification, classifier, unbalanced data, quality measurements of classification

1. Introduction

1.1. Cancers

It is an indisputable fact that cancer has become a disease of civilization. The number of cases of illness and death is steadily increasing and will continue to increase. This is why prevention is so important, in terms of conducting research, raising public awareness, and reducing exposure to factors that increase the risk of disease.

In the human body, cells constantly die and constantly divide – new tissues replace old ones. Neoplasms are a group of diseases in which the process of cell division takes place uncontrollably, and the resulting cells do not resemble typical cells in their structure. Errors in writing and reading

DNA – mutations – cause these cells to react inappropriately to signals sent from the body. This leads to the immortalizing of groups of cells, and these divide in an unlimited and unstoppable way. As soon as cells lose contact with the medium on which they are located, metastases are formed.

According to estimates by the International Agency for Research on Cancer, 12.7 million people worldwide developed cancer in 2008 and 7.5 million died of cancer. In Poland, an estimated 150,000 people are diagnosed with cancer each year (Wojciechowska and Didkowska, 2013).

Breast cancer is a tumor that arises from the tissue of the breast gland and develops locally in the breast. This type produces metastases in lymph nodes and internal organs, most commonly in the lungs, liver, bone, and brain. Malignant tumor of the nipple is the most common of all malignant tumors in women, accounting for an estimated 23% of all cancers in women and 14% of deaths. It affects mainly women in highly developed countries. The fewest cases are observed in Asia and Africa.

Although breast cancer is associated mainly with women, there are also cases of breast cancer among men. They account for 1% of all breast cancers and less than 1% of male malignancies.

The risk depends among others on the following factors: sex, age, genetic stress, reproductive and hormonal factors, lifestyle, other cancers in the past, and occurrence of benign lesions (Wojciechowska and Didkowska, 2013).

1.2. Machine learning

Nowadays, due to the power of computers, data analysis is a powerful tool to improve many aspects of people's lives. In particular, machine learning algorithms can train a machine (e.g., computer, robot) to do a task even more efficiently and faster than humans. They can detect imperceptible relationships between variables. Machine learning methods (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008) can be divided into three main kinds: supervised learning, unsupervised learning, and reinforcement learning. The first can be divided into classification and regression. Based on datasets with known classes, classifiers can learn to recognize which class (group) an observation belongs to. Regression methods model the relationship between different variables. The main task of supervised learning methods is to predict the value of the dependent variable based on a learned model for new values of independent variables (e.g., a new patient after a series of tests). Unsupervised learning tries to find a particular structure of the data. The

best-known example is cluster analysis, which finds homogeneous groups of objects without prior knowledge of labels. Reinforcement learning concerns how intelligent agents ought to take actions in an environment to maximize the notion of cumulative reward (Kaelbling et al., 1996). It does not need labeled input/output pairs like supervised learning. Instead, it focuses on finding a balance between the exploration of uncharted territory and the exploitation of current knowledge.

Machine learning has a variety of applications in different fields of science (see, for example, Mezouar and Afia, 2022 or Li et al., 2021), beginning with academic research, business analysis, and medical applications. It has achieved many spectacular successes. Perhaps one of the most famous is the victory of the AlphaGo algorithm over the human world go champion Lee Sedol. There are also more practical applications. One of the most common applications of machine learning is image recognition. The algorithms can identify objects, persons, places, digital images, etc. For example, “Deep Face” is a Facebook project which recognizes the face and identifies a person in a picture, automatically suggesting a friend. Modern speech recognition is based on machine learning methods. It is used for instance in “search by voice” in Google Assistant, Siri, Cortana, and Alexa. Traffic prediction applications such as Google Maps show us the correct path with the shortest route and predict the traffic conditions. In automatic language translation, artificial neural networks translate a text into our familiar language. Data analysis in medicine can quickly analyze data sets, to show connections that are not detectable to humans at first glance. Of course, there is nothing to replace a doctor’s diagnosis, but analyzing the data can help speed up the diagnostic process. One such medical application of machine learning is considered in this paper, and is described in the next section.

1.3. Machine learning for cancer detection

In this paper, we review the known supervised machine learning algorithms which can be useful in the detection of cancer or its type. There are many methods of supervised learning, and it is well known that there is no method which is always the best. For this reason, one has to choose between many methods for a particular problem and for the available data set. Here, we describe how the whole analysis can be performed. We consider a particular data set containing data about patients with malignant and benign tumors, and our task is to construct a classifier which will be the best for this data set. We consider the following methods: logistic regression, naive

Bayes classifier, K-nearest neighbors, linear discriminant analysis, quadratic discriminant analysis, decision trees, random forests, XGBoost, and support vector machine. Moreover, as the data set is unbalanced, we combine these algorithms with different methods that deal with this problem. Sometimes reduction of the number of variables can improve the results, so we try to test this using principal component analysis. After briefly describing those methods, we present different measures of the quality of classification results. There are many measures, which do not have to give the same recommendation for a classifier, and so the choice of the measure can be important. Finally, we consider an experiment in which different combinations of the above methods are compared. The results are discussed and the best and the worst methods are established for this particular data set. The implementation used the Python programming language (Agresti and Kateri, 2021; Müller and Guido, 2018; VanderPlas, 2016). The code is available from the author. For readers interested in similar research, we refer to the papers (Kourou et al., 2015; Singh et al., 2019; Wernick et al., 2010) and the references therein, which contain different applications of machine learning methods in the medical field.

The paper is organized as follows: Section 2 presents a description of the data set considered as well as some details of its preparation. Section 3 describes the classification methods which are promising methods of machine learning and will be considered in the experiment. The classification quality measures are also presented there. Section 4 presents and discusses the results of the experiment. Section 5 summarizes the paper.

This paper is based on a degree thesis by Ogłoszka (2022).

2. Data

We consider the Breast Cancer Wisconsin Data Set (2022), which is freely available at UC Irvine Machine Learning Repository (Dua and Graff, 2019). It was created by Dr. William H. Wolberg, W. Nick Street, and Olvi L. Mangasarian at the University of Wisconsin (see also Bennett, 1992, and Bennett and Mangasarian, 1992). Data were collected by fine needle aspiration (FNA; Diamantis et al., 2009) of a breast mass. They describe the characteristics of the cell nuclei present in the image. In the following section, more details about this data set are given.

2.1. Classes and variables

The first 569 patients (212 malignant and 357 benign tumors) submitted data for the study, and the results of the other 130 patients were samples for the test (Wolberg et al., 1994). Malignant tumors were histopathologically confirmed and subjects with benign tumors were re-examined after one year of follow-up if the appearance/size of the cancer cells changed. In addition to the visual evaluation of the images obtained after the appropriate preparation of the material, scans of these samples were processed and machine learning was used to assess the malignancy of breast cancer.

Thus we have two classes and we want to predict to which class a new patient belongs. To construct a classifier we use the preserving characteristics of tumor nuclei by digital imaging as described by the following nine variables (Salama et al., 2012):

1. Clump thickness (*thickness*) – we know that benign cells tend to be grouped in monolayers and malignant ones tend to be grouped in multilayers.
2. Uniformity of cell size/shape (*u_cellSize* and *u_cellShape*) – malignant tumors vary in size and shape.
3. Marginal adhesion (*adhesion*) – cancer cells often lose the ability to be adhesive, when the normal cells stick together.
4. Single epithelial cell size (*epith_cellSize*) – this value is related to uniformity of cell size, because if the epithelial cell is enlarged it could mean that the cell is cancerous.
5. Bare nuclei (*nuclei*) – are usually found with benign tumors and mean nuclei that are not surrounded by cytoplasm.
6. Bland chromatin (*chromatin*) – this term concerns the uniform texture of the nucleus. In malignant cells chromatin tends to be coarser.
7. Normal nucleoli (*nucleoli*) – these are small structures seen in the nucleus. In a normal cell the nucleoli are rather small, but in a cancerous one the nucleoli are larger and more prominent, and there may be more than one.
8. Mitoses (*mitoses*) – this means the process when a cell divides and replicates; by counting them, a pathologist can determine the grade of cancer.

It is recognized that the higher the value of a feature, the greater the likelihood of malignancy of a neoplasm (Wolberg et al., 1994). In the next sections, we present some transformations of the data set.

2.2. Unbalanced data

Data sets usually have some inconveniences that make it difficult to work on them. The considered data set has the problem of imbalanced classes (see Figure 1). The benign class is much greater, which is natural as the most common type of tumor is benign. This may affect the classification results. To solve this problem, algorithms were used to improve the balance of the data set by equalizing the size of the samples (Jaworski, 2015).

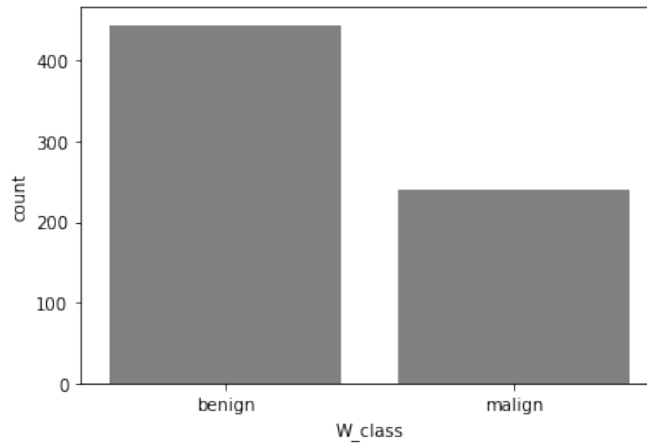


Figure 1. Representation of unbalanced classes

The idea of *Under Sampling* is to randomly reduce the size of classes. Namely, the size of the classes is equal to the size of the one with fewer observations. In *Over Sampling*, we do the opposite, i.e., the size of both classes is increased by copying the record of the minority class. Therefore the sizes of classes are equal to the number of classes with a larger number of observations. The *Synthetic Minority Over-Sampling Technique* (SMOTE) algorithm works similarly to *Over Sampling*, but is slightly different. It selects a random sample from a class with a lower number of observations, and then selects an example from the same class using the K-Nearest Neighbor algorithm and adds it to a smaller class. We will test all of these methods in the experiment described in section 4.

2.3. Dimension reduction

The number of features may impair the performance of machine learning methods. This can occur especially when the number of observations is small, possibly even smaller than the number of variables. In such cases, one of the possible transformations is a dimension reduction. Then, the machine learning methods are applied to the new variables instead of the original ones. There are different methods to reduce the number of features. Here, we consider the classical *principal component analysis* (PCA) (James et al., 2017; Krzyśko et al., 2008). This is a technique to reduce the data dimension by replacing correlated features with a smaller number of uncorrelated variables. In the case of uncorrelated original variables, PCA does not make sense.

The results of PCA are principal components, which are new variables that are linear combinations of the original variables. They are designed in such a way that the first component explains the most of the variability of the original features, and the next one explains as much as possible of the remaining variability of the original set, and so on. The direction of the first principal component is the one along which the observations differ the most, and at the same time, this vector determines the line closest to the data. The successive components are not correlated with the previous ones, and this condition also ensures the orthogonality of the variables to each other.

The principal components can be constructed based on the covariance matrix and on the correlation matrix. However, the results are different. The general recommendation is to use the covariance matrix. However, in case of large differences in the variances of the original variables, it is better to use the correlation matrix.

The number of principal components is the same as the number of original features, but we do not have to use all of the new variables in further analysis. A small number of them usually explain almost the entire variability of the original data set. By data variability, we mean explaining as much of the variance as possible. Typically it is enough to select the first principal components that explain between 80% and 90% of the variance. For our specific data set (section 2), five principal components were selected from the original nine characteristics; these explain 90% of the variance, but three of them could also be used, since they explain 80% of the variance. There is also another method to select principal components – all we have

to do is look at the scree plot (Figure 2). The feature for which the slope of the curve levels off indicates the number of factors that should be generated by the analysis. For our data the number of principal components fulfilling this condition is equal to one.

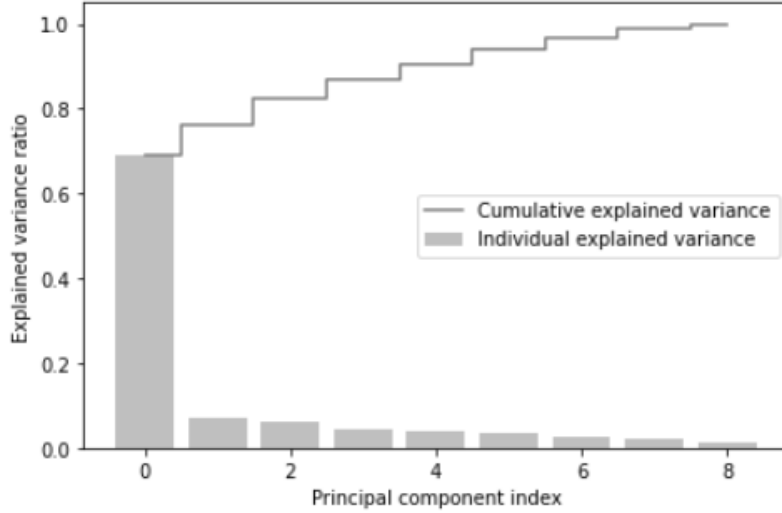


Figure 2. Scree plot for principal component analysis

We will use all three sets with one, three, and five principal components in the experiments in section 4.

2.4. Missing data

In the original data set, there were missing values. More precisely, there were 16 “?” characters in the column “*nuclei*”. We consider two main approaches to dealing with missing values. First, we delete all 16 observations corresponding to missing data. In such a case, it is possible to lose some amount of important information. The second approach is to impute the missing values. There are many methods for this. We have chosen the K-Nearest Neighbors algorithm described in section 3.1.3 to fill these gaps.

2.5. Data sets

Combining the above data set and the methods of its transformation, we consider the following data sets, to which we will apply the classifiers presented in the next section:

1. The base data without 16 rows corresponding to missing data;
2. The base data with missing values filled by the K-Nearest Neighbors algorithm;
3. The data with balanced samples obtained by the *Under Sampling* method;
4. The data with balanced samples obtained by the *Over Sampling* method;
5. The data with balanced samples obtained by the *SMOTE* method;
6. The data mapped to the first principal component (decision taken by scree plot);
7. The data mapped to the first three principal components (explained 80% of variance);
8. The data mapped to the first five principal components (explained 90% of variance).

3. Methodology

Classification is the assignment of an object with specific features to a corresponding class from a predefined set of classes. Classes are mostly denoted by labels and are assigned observations with specific characteristics (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008). The classifier is trained to identify appropriate classes based on the training data set. Then, its performance can be tested using quality measures. Some of them are described in section 3.2. When the performance is satisfactory, we can use the trained method to predict a class for a new observation.

For the considered data set, we are especially interested in binary classification. In this case, there are only two classes. The corresponding labels are usually 0 and 1 for the normal and abnormal states respectively. In medical problems, it is assumed that the normal condition corresponds to a healthy patient and that an abnormal condition represents a patient with a disease.

3.1. Selected classification methods

In this section, we review several popular supervised machine learning methods, which we apply to the considered data set in the experimental part.

3.1.1. Regression

Regression is a statistical method that searches for a function that describes the relationships between variables in the best manner. On this basis, we can predict the unknown values of one quantity based on the known values of another. Typically, a relationship between the explanatory (independent) variables X and the dependent variable Y is sought to describe the expected value of the variable Y depending on the value of the variable X . Nevertheless, one can use the obtained regression model to classify objects. Below, we present the two main regression models which can be used for classification.

Multiple regression The first described regression model is multiple regression. This is the primary regression model that considers the influence of several independent variables $X_1, X_2, \dots, X_n$ on the dependent variable Y (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008). The model of such a regression then assumes the linear relationship between variables:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon = \beta_0 + \beta_1\mathbf{X}_1 + \beta_2\mathbf{X}_2 + \dots + \beta_n\mathbf{X}_n + \epsilon,$$

where:

- $\mathbf{Y}$ – a vector of observations for the dependent variable,
- $\mathbf{X}$ – a matrix of values of explanatory variables,
- $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ – a vector of unknown parameters,
- ϵ – a vector of random errors, which are independent random variables with zero expected value and constant non-zero variance.

Estimating the vector β using the least squares estimator, say $\hat{\beta}$, we obtain a prediction of Y for new data $\mathbf{x}_p$ is as follows:

$$\mathbf{x}_p^\top \hat{\beta}.$$

Then the classifier is of the form $1(\mathbf{x}_p^\top \hat{\beta} > 1/2)$, where $1(\cdot)$ is the indicator function.

Logistic regression In logistic regression, the variable Y has a binomial distribution that has two values: 0 and 1 (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008). The link function is the logit function given as follows:

$$\text{logit}(p) = \ln \frac{p}{1-p},$$

where p is the probability of success, i.e., attaining the value 1. Then this probability is expressed by the following formula:

$$p = \frac{1}{1 + \exp(-\mathbf{X}\beta)},$$

where $\mathbf{X}$ and $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ are the same as in the multiple regression model. After estimating vector β by $\hat{\beta}$, we obtain the predicted value of probability p :

$$\hat{p} = \frac{1}{1 + \exp(-\mathbf{x}_p^\top \hat{\beta})},$$

and the classification rule is $1(\hat{p} > 1/2)$.

3.1.2. Naive Bayes classifier

This is based on the assumption that the features described by the experimental units are mutually independent. In practice, this is very rarely the case, which explains the name “naive”. This classifier is based on the Bayes formula, which expresses a conditional probability as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where $P(A|B)$ expresses the probability of an event A when an event B occurs, and $P(A)$, $P(B)$ are the probabilities of the events A and B respectively (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008).

Translating the above formula into the language of the random variables X and Y , we have:

$$P(Y = y_k | X = x) = \frac{P(X = x | Y = y_k)P(Y = y_k)}{P(X = x)} \quad (1)$$

where $P(Y = y_k)$ is the *a priori* probability, $P(X = x | Y = y_k)$ is the probability of the event $X = x$ in the class y_k , $P(Y = y_k | X = x)$ – the *a posteriori* probability of belonging to a class y_k .

In practice, these probabilities are not known, so the density of each feature in each class is estimated. Then, due to the independence assumption, the product of these densities is used. We assign the observation to the class for which the estimated a posteriori probability is the largest.

3.1.3. K-Nearest Neighbors

The naive Bayes classifier requires an estimation of the edge density function, which is not always easy. The K-Nearest Neighbors (KNN or K-NN) method does not require estimation of the conditional density functions. This classifier assigns the observation to the class to which most of its K neighbors belong. The main issue is the choice of distance. Different distances usually give different results for the KNN method.

3.1.4. Linear and quadratic discriminant analyses

Assuming that the data in all classes have a multidimensional normal distribution, Bayesian methods based on the formula (1) are most appropriate for classification (Górecki, 2011; Krzyśko et al., 2008). When the covariance matrices in all classes are the same, we should use a method called linear discriminant analysis (LDA). When the a priori probabilities of class membership are equal and we are dealing with only two classes, this method is called Fisher's linear discrimination. In such a situation, the boundary surfaces are hyperplanes. Otherwise, when the covariance matrices differ, we use quadratic discriminant analysis (QDA). This gives second-degree hypersurfaces as separating surfaces. In practice, these methods are very often used without checking the normality of the distribution, let alone the equality of the covariance matrix.

Linear discriminant analysis results in discriminant coordinates. These are new variables similar to the principal components known from PCA. They can also be used similarly as principal components, since they are not correlated. An application is, for example, dimension reduction. Unlike the principal components, they are not orthogonal.

3.1.5. Decision trees

Classification trees are special cases of decision trees (DT) (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008). These methods use diagrams being tree-like structures to classify data. This classifier divides all observations

into separate and complementary subsets such that the subsets are as homogeneous as possible. A tree consists of a root and branches leading from the root to the subsequent nodes. At each node, a certain condition related to a given observation is checked. Based on this, one of the branches leading to the next node is selected. At the bottom, there are leaves which indicate to which class the given observation should be assigned.

The advantages of the DT method include the possibility of use for quantitative and qualitative features, efficient classification of objects, a simple form of presentation, and the resistance of this classifier to outliers.

Its extension increases the complexity of the operation of such a model and also makes the interpretation and legibility of the procedure difficult. Moreover, there is a high probability of overfitting. For this reason, first, the trees are most often constructed to be as complex as possible, and then a technique called pruning is used to shrink the tree.

3.1.6. Random forests

Random forests (Breiman, 2001) combine the decision tree method and *bagging*. The method of random forests joins multiple classification trees. First, we draw B bootstrap samples, i.e., B times N elements are drawn with replacement from the N -element data set. For each of the bootstrap samples, we construct a classification tree in such a way that at each node we draw m features that will participate in choosing the best partition. The number m must be less than the number of all features. The trees are constructed without pruning. Finally, the observation is assigned to a class by the voting method, i.e., a vector of feature values is analyzed by all trees and assigned to the class in which it was placed most often. The only parameter of the method is the number m , which should be much smaller than the data dimension p .

3.1.7. XGBoost

The *extreme gradient boosting* (XGBoost) method is derived from decision trees, but it uses *boosting* instead of *bagging* as in random forests (Chen and Guestrin, 2016). *Boosting* improves the classifier with each pass of the algorithm taking into account the errors of the previous transition. For example, it corrects the approach by giving higher weights to cases which were wrongly assigned previously. The word *gradient* in the method name is derived from the use of the *gradient descent* algorithm to determine the

minimum of the loss function.

XGBoost combines everything mentioned above: it uses weak predictors to improve the algorithm by *boosting*, decision trees, and a simple gradient algorithm to correct the loss function.

3.1.8. Support Vector Machine

The Support Vector Machine (SVM; Krzyśko et al., 2008) method consists of converting observations that cannot be subdivided into classes by hyperplane into higher dimensional spaces where they are linearly subdivided. This method selects the observations that are most difficult to separate, and these are determined by the support vectors. The main objective of the classifier is to draw the widest boundary between observations that are difficult to separate. In transforming the observations to higher-dimensional space, kernel functions are used. Thanks to the kernel trick, there is no need to transform the observations. One only needs to measure distances in a higher dimension.

3.2. Classification quality measures

In this section, we present different methods to estimate the classification error in the case of binary classification.

3.2.1. Confusion matrix

When comparing our classification results with real class labels, we have the following four options. Here, label 1 (respectively 0) is called the “good” (respectively “bad”) condition.

- TP (*true positive*) – the model predicted the “good” condition and the real observation was the “good” condition;
- TN (*true negative*) – the model predicted a “bad” condition and the real observation was a “bad” condition;
- FP (*false positive*) – the model predicted the “good” condition and the real observation was the “bad” condition (the first type of error);
- FN (*false negative*) – the model predicted a “bad” condition and the real observation was a “good” condition (the second type of error).

A matrix that has values on the diagonal TP and TN, and beyond it values FP and FN we call a *confusion matrix* (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008). The confusion matrix created based on the data of patients with or without breast cancer is presented in Table 1.

Table 1. Confusion matrix

		Predicted class	
		Predicted positive	Predicted negative
True class	Actually positive	people who have breast cancer and are correctly diagnosed with it	people who have breast cancer and are incorrectly diagnosed as healthy
	Actually negative	people who do not have breast cancer and are incorrectly diagnosed as having breast cancer	people who do not have breast cancer and are correctly diagnosed as healthy

3.2.2. Classification quality measures based on the confusion matrix

The following quality measures can be used to evaluate the classifiers for two-class problem (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008):

Accuracy measures how many of all observations were correctly classified:

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

Recall expresses the probability that the classification will be correct as long as the case is positive:

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

Specificity – the probability that the classification is correct provided that the case is negative:

$$Specificity = \frac{TN}{FP + TN}.$$

Precision – the probability that the classification is correct (positive) provided that the predicted case was positive:

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

F1-score – harmonic mean of precision and sensitivity as a balance between the two:

$$\frac{2TP}{2TP + FP + FN}.$$

3.2.3. ROC curve and AUC

The classification quality measures defined above depend on the threshold used, which is usually equal to a half. To take into account different thresholds, one can use the receiver operating characteristic (*ROC*) curve. Let

$$TPR = \frac{TP}{n_g}, FPR = \frac{FP}{n_b},$$

where $n_g = TP + FN$ denotes the number of observed “good” states, and $n_b = FP + TN$ “bad” states. The *ROC* curve is a plot of *TPR* on the vertical axis against *FPR*. It shows how good the model is at distinguishing between positive and negative cases for different model parameters. The curves, however, are difficult to compare. To overcome this problem, the goodness of the model can be expressed numerically by calculating the area under the *ROC* curve *AUC*, as the following integral:

$$AUC = \int_0^1 ROC(t)dt.$$

The closer the *AUC* value is to one, the greater the model’s ability to correctly predict the “good” state (Górecki, 2011; James et al., 2017; Krzyśko et al., 2008). In other words, when one positive and one negative case are selected, the *AUC* value indicates a value for the probability that the model will assign a greater score to the positive than the negative case.

4. Results

In this section, we present and discuss the results of experiments in which we tried to compare the presented classification methods. The data were preprocessed and visualized. The imbalance problem was solved and the classifiers were applied. Finally, the quality measures were calculated and reported.

First of all, let us notice that the histograms show that benign neoplasms are characterized by lower values of features (Table 2). At the same time, malignant tumors show higher values, and the characteristics are more diverse and characterized by a larger spread. Thus, we can see that the feature *Class* separates variables.

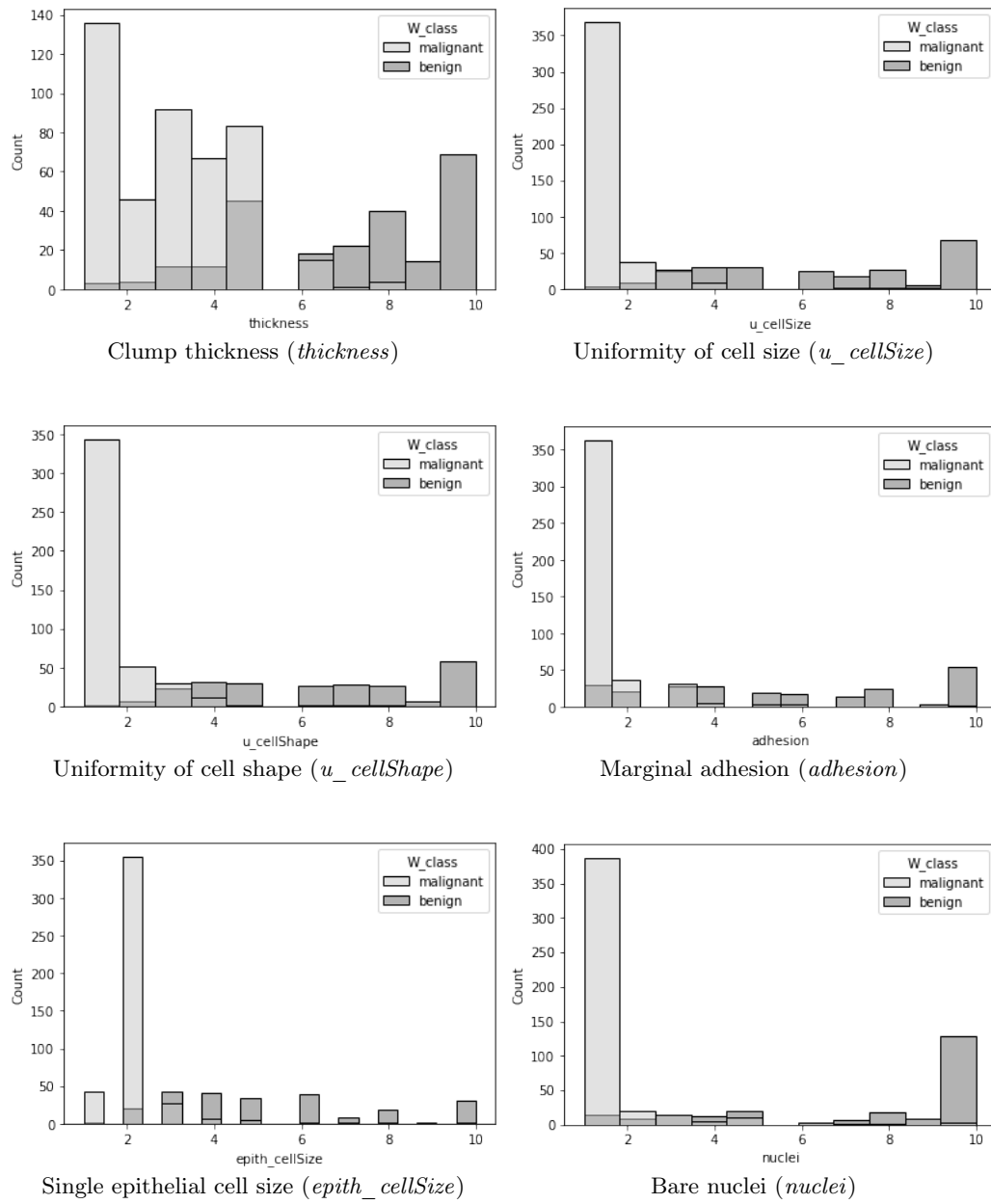
Due to the character of cancer data, the accuracy and recall are the most important classification quality measures. The results are presented in Tables 3–5. Therefore, the best methods under this condition are:

- XGBoost for the set Under Sampling;
- SVM for the set with completed *NaNs*;
- Naive Bayes classifier for the base dataset;
- Naive Bayes classifier for the set Over Sampling;
- KNN for the set Over Sampling;
- KNN for the set SMOTE;
- QDA for the set Over Sampling;
- Random Forest for the set Over Sampling;
- Random forest for the set SMOTE.

The classifier XGBoost for the set *Under Sampling* reached the highest of all measurement values of the classification quality. The worst method cannot be specified, but the multiple regression and the classifier LDA for base data had the lowest values for AUC, recall, and F1. The minimum precision was achieved with the QDA classifier for the base set, and the same method, but the set SMOTE had the lowest accuracy.

Thus, the best combination is XGBoost and the *Under Sampling* set, as it maximized every measure of classification quality. However, it is worth noting that the *Over Sampling* set more often maximized the most important measure here – recall.

The *SMOTE* and the base data set often gave the worst results. The PCA set did not optimize the algorithm's work to obtain large values for the required parameters and reduce the dimensionality of the dataset. We consider also different numbers of combinations of principal components due to different conditions, and we do not have a straight answer to the question of how many principal components is the best. However, this might be because principal component analysis on correlated data works well, by replacing correlated columns with an appropriate combination of uncorrelated ones. The considered set did not have large correlation values.

Table 2. Histograms for all features (data by class of cancer)

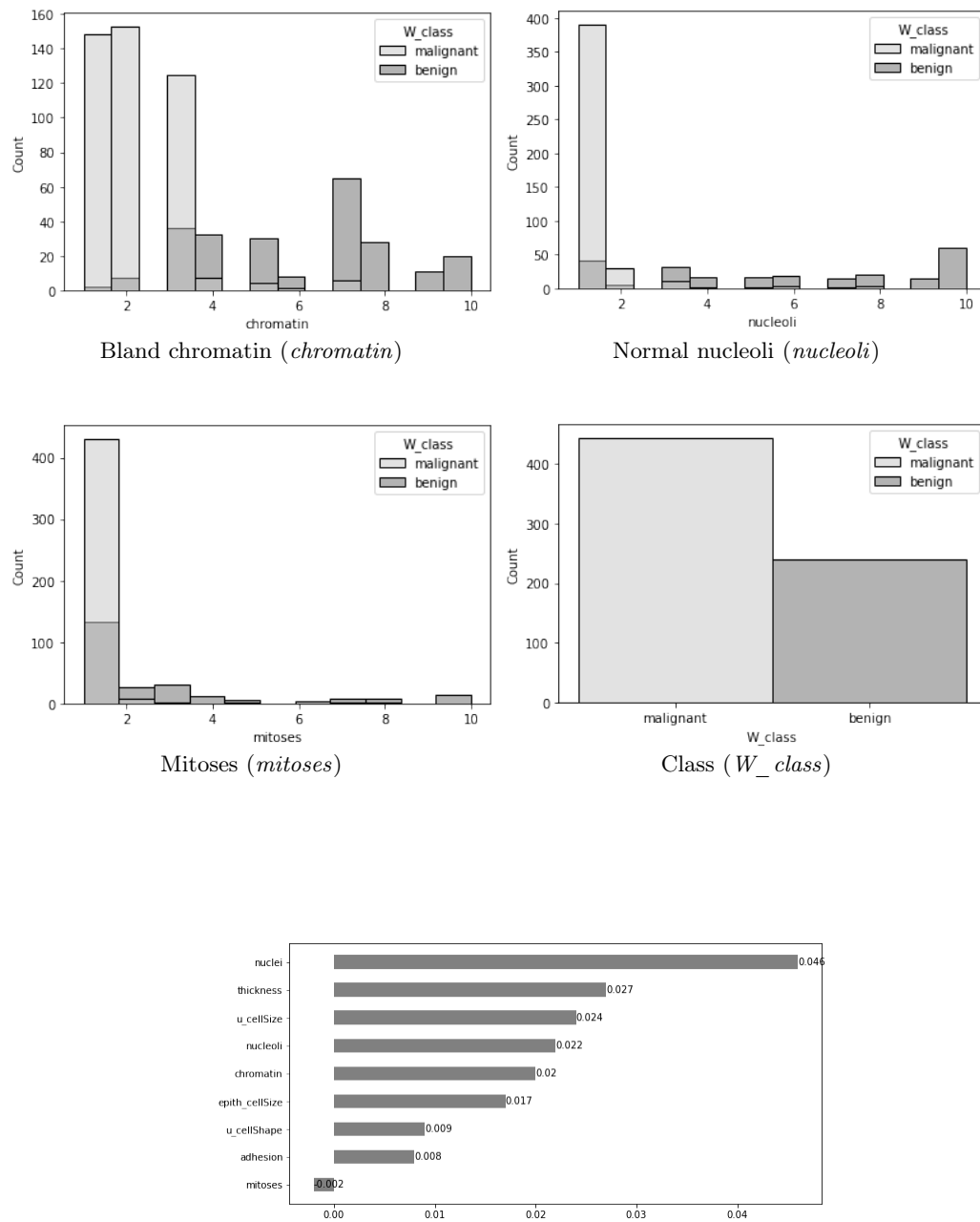


Figure 3. Graphical presentation of the feature importance of variables for the multiple regression method – the maximum importance for feature *nuclei*, the minimum for *mitoses*

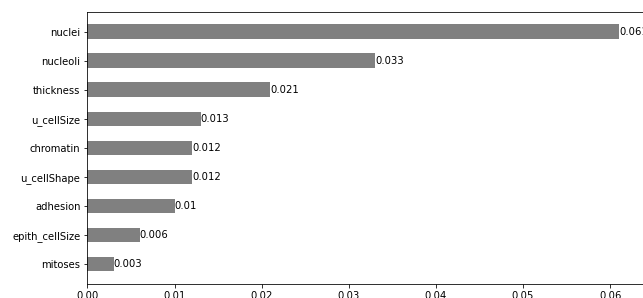


Figure 4. Graphical presentation of the feature importance of variables for the KNN method – the maximum importance for feature *nuclei*, the minimum for *mitoses*

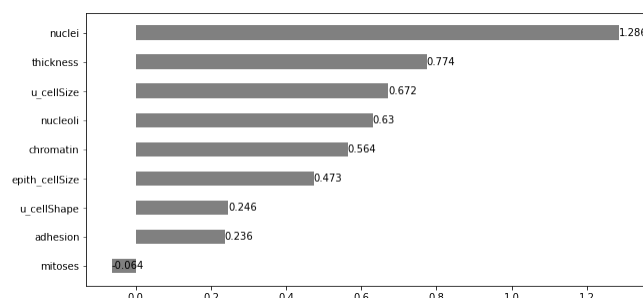


Figure 5. Graphical presentation of the feature importance of variables for the LDA method – the maximum importance for feature *nuclei*, the minimum for *mitoses*

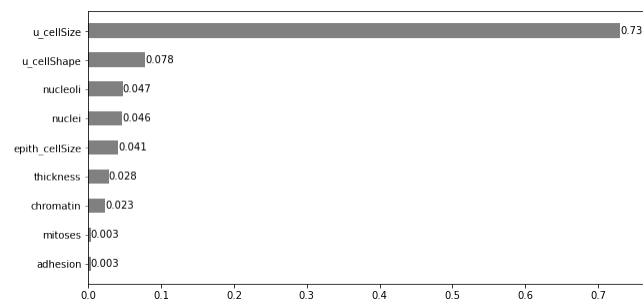


Figure 6. Graphical presentation of the feature importance of variables for the XGBoost method – the best classifier for the data set considered

Finally, between the minimum and maximum value of each measure, improvements were made:

- for AUC by 0.076 (almost 8 percentage points);
- for recall by 0.14 (14 percentage points);
- for precision by 0.106 (almost 11 percentage points);
- for the variable F1 by 0.094 (more than 9 percentage points);
- for accuracy by 0.063 (more than 6 percentage points).

Accuracy was improved by almost 6 percentage points, reaching 99%, and recall by up to 12 percentage points to a maximum value of 100%.

The importance of the variables indicates which trait had the greatest impact on the prediction. This value can be positive or negative, and the greater it is, the more a specific feature will impact the prediction model used (see Saarela and Jauhiainen, 2021, and the references therein). When analyzing the implemented plot feature importance for each classifier, we can see that the most common value with the maximum importance is the *nuclei* feature, and the *mitoses* property usually reaches the minimum of that value (Figures 3, 4, and 5). However, for the best XGBoost method, the best variable is different (Figure 6).

Combining different classifiers and data sets finally produced satisfactory results in terms of recall and accuracy – achieving 100% and almost 100% respectively with such a small dataset is a particularly good result.

Table 3. The classification quality measures (in %) of the classifiers (sorted descending by accuracy; the five best results are given in **bold**, and while the five worst results are presented in *italics*)

Method and set	Accuracy	Recall	Precision	F1 score	AUC
XGBoost, set: Under Sampling	98.96	100.00	98.08	99.03	98.89
Naive Bayes classifier, set: Under Sampling	97.92	98.04	98.04	98.04	97.91
KNN, set: Under Sampling	97.92	98.04	98.04	98.04	97.91
QDA, set: Under Sampling	97.92	98.04	98.04	98.04	97.91
Random Forest, set: Under Sampling	97.92	98.04	98.04	98.04	97.91
SVC, set: Under Sampling	97.92	98.04	98.04	98.04	97.91
KNN, set: PC1-PC5	97.81	100.00	94.34	97.09	98.28
XGBoost, set: PC1-PC5	97.81	100.00	94.34	97.09	98.28

Method and set	Accuracy	Recall	Precision	F1 score	AUC
KNN, set: based	97.81	98.00	96.08	97.03	97.85
KNN, set: all	97.81	98.00	96.08	97.03	97.85
KNN, set: PC1-3	97.81	98.00	96.08	97.03	97.85
KNN, set: Over Sampling	97.75	100.00	95.55	97.72	97.82
KNN, set: SMOTE	97.19	100.00	94.50	97.17	97.28
XGBoost, set: SMOTE	97.19	100.00	94.50	97.17	97.28
Random Forest, set: SMOTE	97.19	100.00	94.50	97.17	97.28
KNN, set: PC1	97.08	100.00	94.50	97.17	97.28
XGBoost, set: PC1-5	97.08	100.00	92.59	96.15	97.70
SVC, set: PC1-3	97.08	100.00	92.59	96.15	97.70
XGboost, set: based	97.08	98.00	94.23	96.08	97.28
XGBoost, set: all	97.08	98.00	94.23	96.08	97.28
Decision Tree, set: PC1-5	97.08	98.00	94.23	96.08	97.28
SVC, set: PC1	97.08	98.00	94.23	96.08	97.28
Multiple Regression, set: Under Sampling	96.88	96.08	98.00	97.03	96.93
LDA, set: Under Sampling	96.88	96.08	98.00	97.03	96.93
SVC, set: Over Sampling	96.63	98.84	94.44	96.59	96.70
Naive Bayes classifier, set: PC1	96.35	98.00	92.45	95.15	96.70
QDA, set: PC1	96.35	98.00	92.45	95.15	96.70
Random Forest, set: PC1-5	96.35	98.00	92.45	95.15	96.70
Random Forest, set: based	96.35	96.00	94.12	95.05	96.28
Random Forest, set: all	96.35	96.00	94.12	95.05	96.28
SVC, set: based	96.35	96.00	94.12	95.05	96.28
SVC, set: all	96.35	96.00	94.12	95.05	96.28
SVC, set: PC1-5	96.35	96.00	94.12	95.05	96.28
Logistic Regression, set: PC1-5	96.35	94.00	95.92	94.95	95.85
Logistic Regression, set: Over Sampling	96.07	98.84	93.41	96.05	96.16
Random Forest, set: Over Sampling	96.07	98.84	93.41	96.05	96.16
SVC, set: SMOTE	96.07	98.84	93.41	96.05	96.16
Logistic Regression, set: SMOTE	96.07	97.67	94.38	96.00	96.12
Logistic Regression, set: Under Sampling	95.83	94.12	97.96	96.00	95.95
Naive Bayes classifier, set: PC1-3	95.62	98.00	90.74	94.23	96.13
QDA, set: PC1-3	95.62	98.00	90.74	94.23	96.13
Decision Tree, set: based	95.62	98.00	90.74	94.23	96.13
Logistic Regression, set: based	95.62	94.00	94.00	94.00	95.28
Logistic Regression, set: all	95.62	94.00	94.00	94.00	95.28
Logistic Regression, set: PC1	95.62	94.00	94.00	94.00	95.28
Logistic Regression, set: PC1-3	95.62	94.00	94.00	94.00	95.28

Method and set	Accuracy	Recall	Precision	F1 score	AUC
Naive Bayes classifier, set: Over Sampling	95.51	100.00	91.49	95.56	95.65
XGBoost, set: Over Sampling	95.51	96.51	94.32	95.40	95.54
Decision Tree, set: Over Sampling	95.51	96.51	94.32	95.40	95.54
Naive Bayes classifier, set: SMOTE	94.94	98.84	91.40	94.97	95.07
Multiple Regression, set: Over Sampling	94.94	94.19	95.29	94.74	94.92
Multiple Regression, set: SMOTE	94.94	94.19	95.29	94.74	94.92
LDA, set: Over Sampling	94.94	94.19	95.29	94.74	94.92
LDA, set: SMOTE	94.94	94.19	95.29	94.74	94.92
Naive Bayes classifier, set: based	94.89	100.00	87.72	93.46	95.98
Naive Bayes classifier, set: all	94.89	100.00	87.72	93.46	95.98
QDA, set: PC1-5	94.89	98.00	89.09	93.33	95.98
Random Forest, set: PC1	94.89	96.00	90.57	93.20	95.13
XGBoost, set: PC1	94.89	94.00	92.16	93.07	94.70
Decision Tree, set: PC1	94.89	94.00	92.16	93.07	94.70
Random Forest, set: PC1-3	94.89	92.00	93.88	92.93	94.28
Multiple Regression, set: PC1-3	94.89	90.00	95.74	92.78	93.85
Decision Tree, set: Under Sampling	94.79	92.16	97.92	94.95	94.97
QDA, set: Over Sampling	94.38	100.00	89.58	94.51	94.57
Naive Bayes classifier, set: PC1-5	94.16	98.00	87.50	92.45	94.98
QDA, set: based	94.16	98.00	87.50	92.45	94.98
QDA, set: all	94.16	98.00	87.50	92.45	94.98
Decision Tree, set: all	94.16	96.00	88.89	92.31	94.55
Multiple Regression, set: based	94.16	88.00	95.65	91.67	92.85
Multiple Regression, set: all	94.16	88.00	95.65	91.67	92.85
Multiple Regression, set: PC1-5	94.16	88.00	95.65	91.67	92.85
LDA, set: based	94.16	88.00	95.65	91.67	92.85
LDA, set: all	94.16	88.00	95.65	91.67	92.85
LDA, set: PC1-5	94.16	88.00	95.65	91.67	92.85
QDA, set: SMOTE	93.82	98.84	89.48	93.92	93.98
Decision Tree, set: SMOTE	93.82	93.02	94.12	93.57	93.79
LDA, set: PC1	93.43	88.00	93.62	90.72	92.28
Multiple Regression, set: PC1	92.70	86.00	93.48	89.58	91.28

Table 4. The results of the classifiers with the best quality measures, sorted descending by accuracy (best 5 scores for each measure)

Method and set	Confusion matrix	
	Predicted positive	Predicted negative
Method: XGBoost		
Dataset: Under Sampling	Actually positive 44	1
	Actually negative 0	51
Method: Naive Bayes Classifier		
Dataset: Under Sampling	Actually positive 44	1
	Actually negative 1	50
Method: KNN		
Dataset: Under Sampling	Actually positive 44	1
	Actually negative 1	50
Method: QDA		
Dataset: Under Sampling	Actually positive 44	1
	Actually negative 1	50
Method: Random Forest		
Dataset: Under Sampling	Actually positive 44	1
	Actually negative 0	51
Method: SVC		
Dataset: Under Sampling	Actually positive 44	1
	Actually negative 1	50
Method: KNN		
Dataset: PC1-5	Actually positive 84	3
	Actually negative 0	50
Method: XGBoost		
Dataset: PC1-3	Actually positive 84	3
	Actually negative 0	50
Method: KNN		
Dataset: Over Sampling	Actually positive 88	4
	Actually negative 0	86
Method: KNN		
Dataset: SMOTE	Actually positive 88	4
	Actually negative 0	86
Method: XGBoost		
Dataset: SMOTE	Actually positive 87	5
	Actually negative 1	85
Method: Random Forest		
Dataset: SMOTE	Actually positive 87	5
	Actually negative 1	85
Method: KNN		
Dataset: PC1	Actually positive 83	4
	Actually negative 0	50
Method: XGBoost		
Dataset: PC1-5	Actually positive 83	4
	Actually negative 0	50
Method: SVC		
Dataset: PC1-5	Actually positive 84	3
	Actually negative 2	48
Method: Naive Bayes Classifier		
Dataset: Over Sampling	Actually positive 84	8
	Actually negative 0	86
Method: Naive Bayes Classifier		
Dataset: base	Actually positive 80	7
	Actually negative 0	50
Method: Naive Bayes Classifier		
Dataset: all	Actually positive 80	7
	Actually negative 0	50
Method: QDA		
Dataset: Over Sampling	Actually positive 82	10
	Actually negative 0	86

Table 5. The results of the classifiers with the worst quality measures, sorted ascending by accuracy (worst 5 scores for each measure)

Method and set	Confusion matrix	
	Predicted positive	Predicted negative
Method: Naive Bayes Classifier	Actually positive	80 7
Dataset: base	Actually negative	0 50
Method: Naive Bayes Classifier	Predicted positive	Predicted negative
Dataset: all	Actually positive	80 7
	Actually negative	0 50
Method: Naive Bayes Classifier	Predicted positive	Predicted negative
Dataset: PC1-5	Actually positive	80 7
	Actually negative	1 49
Method: QDA	Predicted positive	Predicted negative
Dataset: base	Actually positive	80 7
	Actually negative	1 49
Method: QDA	Predicted positive	Predicted negative
Dataset: all	Actually positive	80 7
	Actually negative	1 49
Method: Decision Tree	Predicted positive	Predicted negative
Dataset: all	Actually positive	81 6
	Actually negative	5 45
Method: Multiple regression	Predicted positive	Predicted negative
Dataset: base	Actually positive	85 2
	Actually negative	6 44
Method: Multiple regression	Predicted positive	Predicted negative
Dataset: all	Actually positive	85 2
	Actually negative	6 44
Method: Multiple regression	Predicted positive	Predicted negative
Dataset: PC1-5	Actually positive	85 2
	Actually negative	6 44
Method: LDA	Predicted positive	Predicted negative
Dataset: base	Actually positive	85 2
	Actually negative	6 44
Method: LDA	Predicted positive	Predicted negative
Dataset: all	Actually positive	85 2
	Actually negative	6 44
Method: LDA	Predicted positive	Predicted negative
Dataset: PC1-5	Actually positive	85 2
	Actually negative	6 44
Method: QDA	Predicted positive	Predicted negative
Dataset: SMOTE	Actually positive	81 6
	Actually negative	1 49
Method: Decision Tree	Predicted positive	Predicted negative
Dataset: SMOTE	Actually positive	87 5
	Actually negative	1 85
Method: LDA	Predicted positive	Predicted negative
Dataset: PC1	Actually positive	84 3
	Actually negative	6 44
Method: Multiple regression	Predicted positive	Predicted negative
Dataset: PC1	Actually positive	84 3
	Actually negative	7 43

5. Conclusion

In this study, the problem of diagnosing the type of breast cancer (malignant or benign) with the use of machine learning methods was considered.

In the theoretical part, the problem of classification was discussed along with a description of many efficient classifiers. The topic of data set transformations, to deal with the problem of data unbalance, was also discussed. The different classification methods and their quality were described. In the experimental part, an attempt was made to find the best classifier and improve the quality of the original data set to obtain the highest values of classification quality measures.

In the course of the work, it turned out not only that the selection of the best classifier can improve the diagnosis process, but also that the work on the data set can give good results.

Further research may include other, perhaps more recent, classification methods, such as the ensemble classifier (Datta et al., 2010) or artificial neural networks. It would be a good idea to expand research on the topic with an analysis of the original photo-based data set and to extend the code to provide the ability to classify based on images.

Acknowledgements

The authors are grateful to the anonymous referees for the many valuable comments and constructive suggestions which resulted in the present version.

REFERENCES

- Agresti A., Kateri M. (2021): Foundations of Statistics for Data Scientists: With R and Python. CRC Press.
- Bennett K.P. (1992): Decision tree construction via linear programming. Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, 97–101.
- Bennett K.P., Mangasarian O.L. (1992): Robust linear programming discrimination of two linearly inseparable sets. Optimization Methods and Software 1: 23–34.
- Breast Cancer Wisconsin Data Set <https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28original%29> [accessed: 21.10.2022].
- Breiman L. (2001): Random forests. Machine Learning 45: 5–32.
- Chen T., Guestrin C. (2016): XGBoost: A scalable tree boosting system. arXiv: 1603.02754 [accessed: 21.10.2022].

- Datta S., Pihur V., Datta S. (2010): An adaptive optimal ensemble classifier via bagging and rank aggregation with applications to high dimensional data. *BMC Bioinformatics* 11: 427.
- Diamantis A., Magiorkinis E., Koutselini H. (2009): Fine-needle aspiration (FNA) biopsy: historical aspects. *Folia Histochemica Et Cytobiologica* 47: 191–197.
- Dua D., Graff C. (2019): UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science <http://archive.ics.uci.edu/ml> [accessed: 21.10.2022].
- Górecki T. (2011): Basics of statistics with examples in R. BTC (in Polish).
- James G., Witten D., Hastie T., Tibshirani R. (2017): An Introduction to Statistical Learning with Applications in R. Springer.
- Jaworski B. (2015): Cost-optimal sampling of examples for imbalanced data. Master's thesis defended in Computer Science. Poznań University of Technology, Poznań (in Polish) https://kofeina.net/~benek/studia/praca_magisterska_benedykt_jaworski.pdf [accessed: 21.10.2022].
- Kaelbling L.P., Littman M.L., Moore A.W. (1996): Reinforcement learning: A Survey. *Journal of Artificial Intelligence Research* 4: 237–285.
- Kourou K., Exarchos T.P., Exarchos K.P., Karamouzis M.V., Fotiadis D.I. (2015): Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal* 13: 8–17.
- Krzyśko M., Wołyński W., Górecki T., Skorzybut M. (2008). Machine learning – pattern recognition, cluster analysis and dimensional reduction. WNT, Warsaw (in Polish).
- Li Y., Shan B., Li B., Liu X., Pu Y. (2021): Literature review on the applications of machine learning and blockchain technology in smart healthcare industry: A bibliometric analysis. *Journal of Healthcare Engineering* 9739219.
- Mezouar H., Afia A.E. (2022): A systematic literature review of machine learning applications in software engineering. In: Lazaar, M., Duvallet, C., Touhafi, A., Al Achhab, M. (eds) *Proceedings of the 5th International Conference on Big Data and Internet of Things. BDIoT 2021. Lecture Notes in Networks and Systems*, vol 489. Springer, Cham.
- Müller A.C., Guido S. (2018): Introduction to Machine Learning with Python. O'REILLY.
- Ogłoszka A.M. (2022): Classification methods in the diagnosis of breast cancer. Master's thesis defended in Data Science. Adam Mickiewicz University, Poznań (in Polish).
- Saarela M., Jauhiainen S. (2021): Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences* 3, article number 272.
- Salama G.I., Abdelhalim M.B., Zeid M. (2012): Breast cancer diagnosis on three different datasets using multi-classifiers. *International Journal of Computer and Information Technology* 1: 36–43.

- Singh P., Singh S.P., Singh D.S. (2019): An introduction and review on machine learning applications in medicine and healthcare. 2019 IEEE Conference on Information and Communication Technology, pages 1–6, doi: 10.1109/CICT48419.2019.9066250.
- Wernick M.N., Yang Y., Brankov J.G., Yourganov G., Strother S.C. (2010): Machine learning in medical imaging. IEEE Signal Processing Magazine 27: 25–38.
- Wojciechowska U., Didkowska J. (2013): Illnesses and deaths from malignant neoplasms in Poland. National Cancer Registry from National Oncology Institute Maria Skłodowska-Curie – National Research Institute. Available at <http://onkologia.org.pl/raporty/> [accessed on 06.01.2022].
- Wolberg W.H., Street W.N., Mangasarian O.L. (1994): Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77: 163–171.
- VanderPlas J. (2016): Python Data Science Handbook. O'REILLY.