# **Performance Analysis of Supervised Classification Models on Heart Disease Prediction**

**Ezekiel Adebayo Ogundepo1; Waheed Babatunde Yahya2; Olubayo Adekanmbi1**

1Data Science Nigeria (DSN), Lagos, Nigeria

2Department of Statistics, University of Ilorin, Ilorin, Nigeria.

e-mail: ezekiel@datasciencenigeria.ai, ogundepoezekiel@gmail.com1; wbyahya@unilorin.edu.ng2; olubayo@datasciencenigeria.ai1

**Abstract**

**Purpose**

The application of artificial intelligence in predictive medicine helps to flag the risk factors so that early preclinical diagnosis of health conditions can be achieved to reduce the chances of future problems. This paper presents a detailed predictive analysis of heart disease conditions of some patients to determine the possible risk factors that are suggestive of whether a patient in the sample has heart disease or not. The predictive strength of the identified risk factors was further determined using suitable classification methods.

**Design and Methodology**

Two independent (but similar) published heart disease data, the Cleveland (training) and the Statlog (validation), were considered in this paper. Detail exploratory analysis using the Chi-square test of independence was performed to get more insight about the features that are useful in determining whether a patient has heart disease or not before employing ten standard machine learning techniques for the class prediction. We randomly partitioned the Cleveland data into 70% (208) training samples for building the models and 30% (89) test samples for evaluating their predictive performances over 200 replications. These results were validated on the Statlog data in 10-fold cross-validation.

**Findings**

Results from this study indicated that the Support Vector Machine (SVM) yielded the best predictive performances with 85% Accuracy, 82% Sensitivity, 87% Specificity, 87% Precision, 91% Area under the ROC curve (AUC), and 38.5% LogLoss value. We validated the performance results of the SVM on the Statlog dataset, and appreciable similar predictive results were obtained. Results of the Chi-square test of association showed that some bio-clinical categorical variables are strongly associated with the heart disease conditions of the patients in the two data sets (p < 0.001).

**Practical implications**

This study provides a practical clue to the crops of bio-clinical variables to watch in the clinical diagnosis and treatment of heart disease patients.

**Originality/value**

The result presented in this work is original. More importantly, the prediction results provided by the best-chosen classifier, SVM, on the Cleveland (training) data were all consistent with the validation results obtained from the Statlog data. Therefore, the strength of this work lies in the ability of the SVM classifier to reproduce the prediction results obtained when applied on similar independent heart disease data sets.

**Keywords:** Machine learning, Classifiers, Feature Selection, Heart Disease Classification, Exploratory Data Analysis (EDA), Performance measures.

# **Introduction**

Heart disease refers to several types of heart conditions, and it is one of the dominant and severe illnesses that affect people worldwide and often results in death [1, 2]. The World Health Organization (WHO) estimated that about 17 million lives are lost yearly due to heart-related diseases, which are relatively more prevalent in Asia, India, and the United States of America [3, 4]. Heart diseases are usually dominant in males than females and mostly affects middle-aged and older people [5]. Medical research has identified that lifestyles, obesity, eating habits and physical inactivity are significant factors that can lead to heart-related diseases [6]. Other factors like smoking, high blood pressure, ECG rate, cholesterol level, height, and hypertension can increase heart disease chances.

Heart disease prevention is essential. A sound data-driven system for predicting heart disease will enable us to learn how to reliably detect it and improve the entire research and prevention process, making sure that more people can live healthy lives. Many studies have shown that machine learning techniques effectively predict heart disease given a set of prognostic and bio-clinical variables. [7] proposed a decision support system developed in MATLAB’s GUI using backpropagation neural network for diagnosis of congenital heart disease in children, which gives a better performance compared with the manual diagnosis. [8] used Naive Bayes, classification by clustering, and Decision Tree with a reduced number of features to predict heart disease risk.

Several state-of-the-art classifiers are available in the literature to predict heart disease patients' health conditions. [9] developed a sequential-based feature selection and classification method to efficiently select the core gene biomarkers to predict the histopathological responses of 43 patients with locally advanced rectal carcinomas. [10] investigated and compared the accuracy of seven classifiers for predicting heart disease in the Cleveland data. The experiments' results indicate that the SVM method using the boosting technique outperformed other methods. [11] used an ensemble technique to predict the risk of heart disease, and the result showed ensemble methods could increase accuracy by 7%. A study by [12] showed that decision tree classifiers are simpler and more accurate than most algorithms.

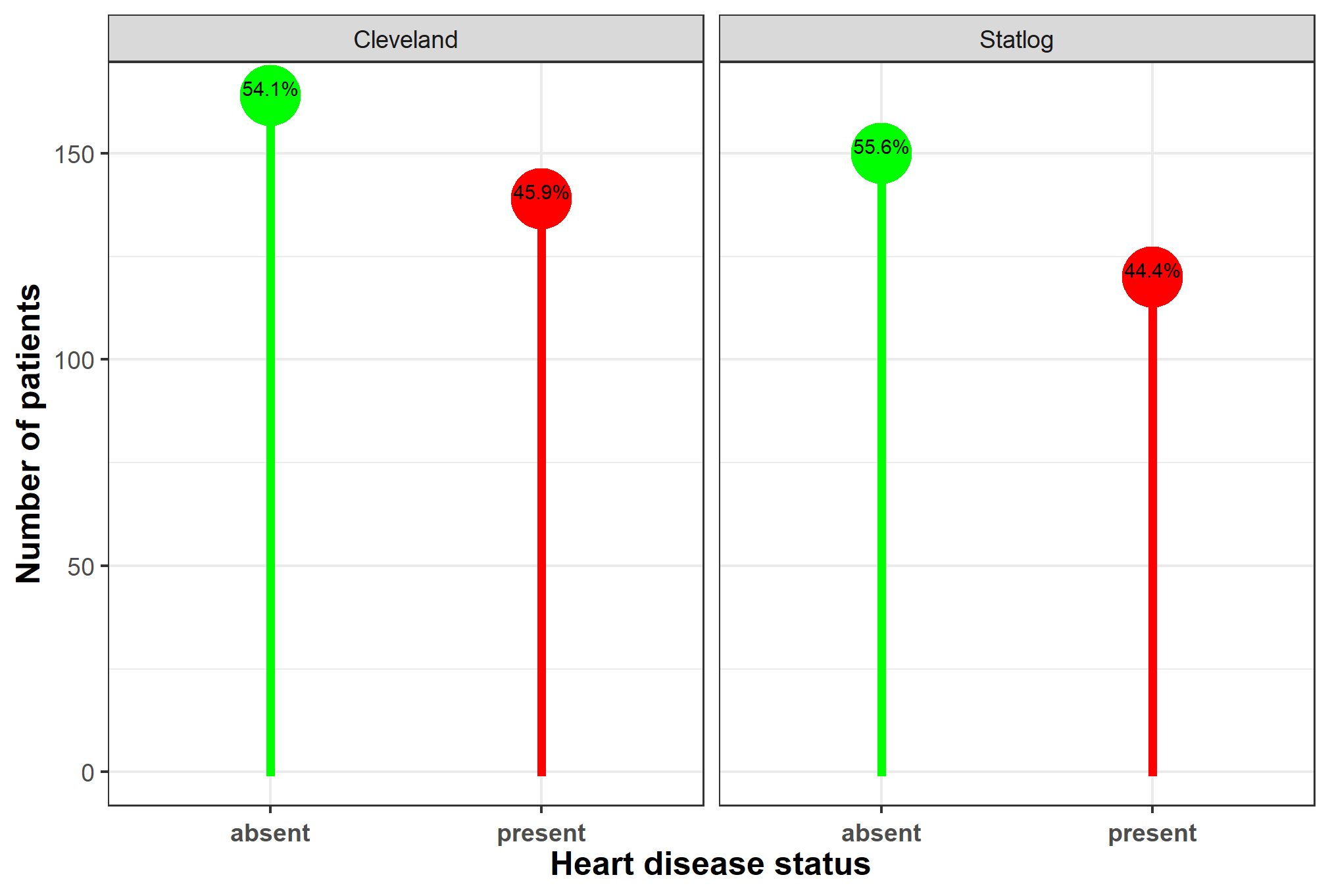
This work aims to evaluate the performances of ten classifiers such as Logistic regression, Random Forest, Decision Tree, Naive Bayes, k-Nearest Neighbour, Extreme Gradient Tree, Conditional Random Forests, Linear Discriminant Analysis, Artificial Neural Network and Support Vector Machine in predicting heart disease. The results form this work would serve as a useful guide in a future study in choosing the appropriate machine learning technique(s) to analyse the kind of data presented in this work as there is no free lunch of statistical machine learning models [13].

**2.0 Materials and methods**

# **2.1 Materials**

This study employed two heart disease datasets open to the public at the UCI (University of California, Irvine C.A) Machine Learning Repository. The first data, which are hereafter called the Cleveland data, were obtained from a heart disease study in the Cleveland database [14] and is available at <https://bit.ly/cleveland-heart-disease-database>. The second data, which are hereafter called the Statlog, were obtained from the Statlog heart disease database [46] and is available at <https://bit.ly/statlog-heart-disease-database>. The two datasets have 13 features, representing the bio-clinical variables obtained on heart disease patients and their cardiovascular statistics. These prognostic variables were used in the various models in this study to predict whether a patient has heart disease or not.

The Cleveland dataset has 303 patients, out of which 139 (45.9%) patients had heart disease, and 164 (54.1%) patients did not have the condition. However, during the preprocessing of the Cleveland dataset, we removed six (6) patients with incomplete information, which reduced the total samples to 297 patients, out of which 137 (46%) patients had heart disease and 160 (54%) did not have the condition. The Statlog data contained 270 patients, out of which 120 (44.4%) patients had heart disease, and 150 (55.6%) patients did not have the condition.



**Figure 1:** A lollypop chart showing the distribution of heart disease status between Cleveland and Statlog heart disease datasets.

***The Training and Validation Data***

In this study, the Cleveland dataset was used as the training data to build all the ten classification models. The Statlog, although entirely independent of the Cleveland data, are nearly similar in structure to the Cleveland data. Therefore, the Statlog heart disease dataset is used as the validation data. By this, all the results obtained on the training (Cleveland) data are validated on the Statlog for efficiency and assessment of the extent of reproducibility of the results.

Table 1 shows the descriptive statistics of the continuous variables in Cleveland heart disease data. The patients' mean age was 54.54 years with a standard deviation of 9.05 years, while the minimum and maximum ages were 29 years and 77 years, respectively. Table 2 and 4 show a detailed exploratory data analysis using the Chi-square test of independence for Cleveland and Statlog heart disease datasets; to get more insight about the features that are useful in determining whether a patient has heart disease or not before applying standard machine learning models for the class prediction.

**Table 1:** Summary statistics of the continuous features in the Cleveland heart disease dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Determinants | Minimum | Maximum | Mean | Median | Standard deviation |
| Age (in years) | 29 | 77 | 54.5421 | 56 | 9.0497 |
| Resting blood pressure in mm Hg | 94 | 200 | 131.6936 | 130 | 17.7628 |
| Serum cholestoral in mg/dl | 126 | 564 | 247.3502 | 243 | 51.9976 |
| Maximum heart rate achieved | 71 | 202 | 149.5993 | 153 | 22.9416 |
| ST depression | 0 | 6.2 | 1.0556 | 0.8 | 1.1661 |

**Table 2:** Frequency (percentage in parenthesis) distribution of determinants of heart disease across the 297 patients in the Cleveland data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Determinants (factors) | Factor levels | Heart disease status | | Total  297 (100%) | P-value |
| **Absent (0)**  **160 (54%)** | **Present (1)**  **137 (46%)** |
| Sex of the patients | Male (ref) | 89 (44.3%) | 112 (55.7%) | 201 (67.7%) | <0.001 |
| Female | 71 (74.0%) | 25 (26.0%) | 96 (32.3%) |
| Chest pain type | Typical angina (ref) | 16 (69.6%) | 7 (30.4%) | 23 (7.7%) | <0.001 |
| Asymptomatic | 39 (27.5%) | 103 (72.5%) | 142 (47.8%) |
| Non-anginal pain | 65 (78.3%) | 18 (21.7%) | 83 (27.9%) |
| Atypical angina | 40 (81.6%) | 9 (18.4%) | 49 (16.5%) |
| FBS > 120 mg/dl | No (ref) | 137 (53.9%) | 117 (46.1%) | 254 (85.5%) | 0.999 \* |
| Yes | 23 (53.5%) | 20 (46.5%) | 43 (14.5%) |
| REC | 2 (ref) | 67 (45.9%) | 79 (54.1%) | 146 (49.2%) | 0.008 |
| 0 | 92 (62.6%) | 55 (37.4%) | 147 (49.5%) |
| 1 | 1 (25.0%) | 3 (75.0%) | 4 (1.3%) |
| Exercise induced angina | No (ref) | 137 (68.5%) | 63 (31.5%) | 200 (67.3%) | <0.001 |
| Yes | 23 (23.7%) | 74 (76.3%) | 97 (32.7%) |
| SST | Downsloping (ref) | 9 (42.9%) | 12 (57.1%) | 21 (7.1%) | <0.001 |
| Flat | 48 (35.0%) | 89 (65.0%) | 137 (46.1%) |
| Upsloping | 103 (74.1%) | 36 (25.9%) | 139 (46.8%) |
| Major vessel | 0 (ref) | 129 (74.1%) | 45 (25.9%) | 174 (58.6%) | <0.001 |
| 3 | 3 (15.0%) | 17 (85.0%) | 20 (6.7%) |
| 2 | 7 (18.4%) | 31 (81.6%) | 38 (12.8%) |
| 1 | 21 (32.3%) | 44 (67.7%) | 65 (21.9%) |
| Thalliumstress test | Fixed defect (ref) | 6 (33.3%) | 12 (66.7%) | 18 (6.1%) | <0.001 |
| Normal | 127 (77.4%) | 37 (22.6%) | 164 (55.2%) |
| Reversable defect | 27 (23.5%) | 88 (76.5%) | 115 (38.7%) |

N.B.: The 𝑃-value is from Pearson chi-square test of independence. The symbol (∗) indicates that the chi-square test of independence is not significant at 5% level. (ref) indicates the reference category of a factor as used in fitting our various models. FBS = Fasting blood sugar; REC = Resting electrocardiographic results; SST = Slope of the peak exercise ST segment, Major vessel = Number of major vessels colored by fluoroscopy.

For the Statlog heart disease data, the patients' mean age was 54.53 years with a standard deviation of 9.11 years, while the minimum and maximum ages were 29 years and 77 years, respectively, are shown in Table 3.

**Table 3:** Summary statistics of the continuous features in the Statlog heart disease dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Determinants | Minimum | Maximum | Mean | Median | Standard deviation |
| Age (in years) | 29 | 77 | 54.433 | 55 | 9.109 |
| Resting blood pressure in mm Hg | 94 | 200 | 131.344 | 130 | 17.862 |
| Serum cholesterol in mg/dl | 126 | 564 | 249.659 | 245 | 51.686 |
| Maximum heart rate achieved | 71 | 202 | 149.678 | 153.5 | 23.166 |
| ST depression | 0 | 6.2 | 1.05 | 0.8 | 1.145 |

**Table 4:** Frequency (percentage in parenthesis) distribution of determinants of heart disease across the 270 patients in the Statlog data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Determinants (factors) | Factor levels | Heart disease status | | Total  270 (100%) | P-value |
| **Absent (0)**  **150 (55.6%)** | **Present (1)**  **120 (44.4%)** |
| Sex of the patients | Male (ref) | 83 (45.4%) | 100 (54.6% ) | 183 (67.8%) | <0.001 |
| Female | 67 (77.0%) | 20 (23.0%) | 87 (32.2%) |
| Chest pain type | Typical angina (ref) | 15 (75.0%) | 5 (25.0%) | 20 (7.4%) | <0.001 |
| Asymptomatic | 38 (29.5%) | 91 (70.5%) | 129 (47.8%) |
| Non-anginal pain | 62 (78.5%) | 17 (21.5%) | 79 (29.3%) |
| Atypical angina | 35 (83.3%) | 7 (16.7%) | 42 (15.6%) |
| FBS > 120 mg/dl | No (ref) | 127 (55.2%) | 103 (44.8%) | 230 (85.2%) | 0.999 \* |
| Yes | 23 (57.5%) | 17 (42.5%) | 40 (14.8%) |
| REC | 2 (ref) | 64 (46 .7%) | 73 (53.3%) | 137 (50.7%) | 0.008 |
| 0 | 85 (64.9%) | 46 (35.1%) | 131 (48.5%) |
| 1 | 1 (50.0%) | 1 (50.0%) | 2 (0.7%) |
| Exercise induced angina | No (ref) | 127 (70.2%) | 54 (29.8%) | 181 (67%) | <0.001 |
| Yes | 23 (25.8%) | 66 (74.2%) | 89 (33%) |
| SST | Downsloping (ref) | 8 (44.4%) | 10 (55.6%) | 18 (6.7%) | <0.001 |
| Flat | 44 (36.1%) | 78 (63.9%) | 122 (45.2%) |
| Upsloping | 98 (75.4%) | 32 (24.6%) | 130 (48.1%) |
| Major vessel | 0 (ref) | 120 (75.0%) | 40 (25.0%) | 160 (59.3%) | <0.001 |
| 3 | 3 (15.8%) | 16 (84.2%) | 19 (7%) |
| 2 | 7 (21.2%) | 26 (78.8%) | 33 (12.2%) |
| 1 | 20 (34.5%) | 38 (65.5%) | 58 (21.5%) |
| Thalliumstress test | Fixed defect (ref) | 6 (42.9%) | 8 (57.1%) | 14 (5.2%) | <0.001 |
| Normal | 119 (78.3%) | 33 (21.7%) | 152 (56.3%) |
| Reversable defect | 25 (24.0%) | 79 (76.0%) | 104 (38.5%) |

N.B.: The 𝑃-value is from Pearson chi-square test of independence. The symbol (∗) indicates that the chi-square test of independence is not significant at 5% level. (ref) indicates the reference category of a factor as used in fitting our various models. FBS = Fasting blood sugar; REC = Resting electrocardiographic results; SST = Slope of the peak exercise ST segment, Major vessel = Number of major vessels colored by fluoroscopy.

**2.2 Methods**

We can implement machine learning (ML) techniques in different forms. Notable among these include supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning methods [15, 16]. The supervised learning algorithms consist of a label and a set of features. The task is to learn a function that maps an input to an output based on example, input-output pairs [17]. A supervised learning algorithm analyzes the training data and produces an inferred function for mapping test data or new input (X) to predict the output or label (Y). The two examples of supervised learning techniques are classification and regression. The main difference between these two methods is that regression has the label of the data continuous while classification has the categorical label (or discrete). This paper applied ten classifiers or classification learning methods on two real-life data to predict whether a patient has heart disease or not.

**2.2.1 *Classification machine learning methods***

A classifier is a supervised ML method that learns from the training data and uses that knowledge to predict unseen or future (test) data. This section presents a brief definition of the most widely used classification learning methods as employed in this work.

***Decision Tree (DT)***

A decision tree (DT) is a tree-like structure that consists of a root node, branches, and leaf nodes [10]. It is a non-parametric model that can efficiently deal with large and complex datasets without imposing complicated distributional assumptions. We can implement DT in both classification and regression tasks. It is easy to interpret, robust to outliers and can also work in the presence of missing values without needing to resort to imputation. A decision tree model's main disadvantage is that it can be subject to overfitting and underfitting when using a small data set [18].

***Extreme Gradient Tree (XGBTree)***

Extreme Gradient Tree (XGBTree), which is also known as Extreme Gradient Boosting (XGBoost) method, is an efficient and scalable implementation of gradient boosted decision trees that are designed for execution speed and model performance [19]. XGBTree offers state-of-the-art results on many challenges and can automatically handle missing data and support tree construction's parallelisation. Other features that make XGBTree efficient is the cache access patterns, data compression capabilities and sharding to build a scalable tree boosting system [20].

***Conditional Random Forests (Cforest)***

Conditional Random Forests (Cforest) is a bagging tree ensemble technique similar to the random forest. The main difference between Cforest and the classical random forest is how the trees are aggregated during the training phase. Cforest uses conditional inference trees as base learners, and this puts more weight on terminal nodes where there is a higher cost. The Cforest approach is better than the classical tree algorithms because the trees are unbiased and do not artificially favour splits in variables with many categories or continuous variables [21].

***Logit***

Logit, also known as logistic regression, is a statistical model that uses a logistic curve to model the probability of a particular class or event existing. Logistic regression studies the association between a categorical dependent variable and a set of independent (explanatory) variables [22, 23]. Logistic regression can be binomial or multinomial. Binomial or binary logistic regression [24] refer to the instance in which the observed outcome can have only two possible outcomes (e.g., “dead” or “alive”, “success” or “failure”, or “yes” or “no”). Multinomial logistic regression refers to cases where the outcome can have three or more possible types (e.g., “better”, “no change”, and “worse”).

***Random Forest (Rforest)***

Random forest (Rforest) is a popular machine learning model used for classification or regression tasks. Random forest falls under a class of algorithms called bagging, which from numerous experiments have shown to outperform single tree models. It constructs multiple decision trees trained on a bootstrap dataset and aggregates the result. The idea behind this is the belief that multiple decision trees will optimally converge to the perfect decision. One significant advantage of the random forest is that it drastically decreases the model's variance without increasing the bias [25].

***Support Vector Machine (SVM)***

Support Vector Machine (SVM) is a supervised machine learning algorithm that uses a robust margin to separate instances into different classes [26]. Given a set of training samples, an SVM model represents the samples as points in space so that there is a clear separation margin between them. New test samples are mapped into that same space and are classified based on the margin's side where they fall.

***Artificial Neural Network (ANNs)***

Artificial Neural Network (ANNs) are computational networks mainly inspired by biological neurons [27, 28]. It represents a collection of computation units that are connected where each unit provides the input to the next unit in the chain. ANNs mainly used an unstructured supervised learning task and have shown great success compared to traditional tree algorithms. One of the reasons for its success is that artificial neural network can accurately extract features from unstructured data without humans' intervention and that they can accurately learn and simulate high dimensional, non-linear data without prior knowledge. Generally, ANNs are arranged in layers: an input layer, one or more hidden layers, and the output layer [29].

***The k-Nearest Neighbour (kNN)***

The k-Nearest Neighbour (kNN) is perhaps the simplest, most popular, highly efficient, and intuitive algorithm for pattern recognition [24]. The strategy for predicting the class of an observation is to identify the k closest neighbours from among the training dataset and then assign the class with the most prevalent class among its nearest neighbours. The kNN works well with a small number of input variables (p) but struggles when the number of inputs (features) is large [30]. The kNN does not have a training phase; hence it is referred to as a lazy learner. We can implement kNN for both the classification and regression problems.

***Linear Discriminant Analysis (LDA)***

Linear Discriminant Analysis (LDA) is a statistical predictive classification method that assigns an unknown class label to one of the classes based on a multivariate observation [31]. LDA finds a linear combination of features that separates two or more classes of observations and makes some simplifying assumptions about the data—normally distributed data, statistically independent features, and identical covariance matrices for every class. LDA makes predictions by estimating the probability that a new set of inputs belongs to each class, and the class that gets the highest probability is the output class. The model uses Bayes Theorem to estimate the probabilities [32].

***Naive Bayes (NB)***

Naive Bayes is a probabilistic machine learning classifier that is based on the Bayes Theorem and can be used for a wide variety of classification tasks [24]. The naive Bayes assumption is that the features that go into the model are independent of each other. It is a simple yet powerful algorithm that is so popular in classification methods because of its predictions that it can quickly make in real-time.

**2.2.2 *Evaluation of Classification Learning Methods***

In this section, we provided a brief discussion of the different metrics for evaluating the performance of various classification models used in this work. The performance measures employed here include the Accuracy, Sensitivity, Specificity, Precision, LogLoss and the Area under Receiver Operating Characteristics curve (AUC).

***Accuracy or Correct Classification Rate***: The accuracy of any given classifier is the ratio of response class labels that the classifier predicted correctly over the total number of predictions.

For a binary classifier, the classification rate is given by:

where TP = True Possitive; TN = True Nagative; FP = False Positive and FN = False Nagative.

In a class-imbalanced dataset, accuracy may not be an excellent method to evaluate classifiers; instead, we may use recall, precision, or the F1 score.

***True Positive Rate* (TPR)**: The True Positive Rate (TPR), often called the Sensitivity or Recall, is the ratio of the total number of samples that are correctly classified as having the response of interest (disease present) divided by the total number of samples that have the response of interest in the test data. Sensitivity is obtained as:

Sensitivity (TPR)

***True Negative Rate* (TNR)**: The True Negative (TNR), often called the Specificity, is the ratio of the total number of samples that are correctly classified as not having the response of interest (disease absent) divided by the total number of samples that do not have the response of interest in the test data. Specificity is obtained as:

Specificity (TNR)

***Precision***: The precision measures how often a classifier correctly predicts the response of interest (disease present). For instance, when the classifier predicts the heart disease to be present in a set of samples, how often is it correct? Precision can be expressed as follows:

***Log-Loss***: The Logarithmic Loss simply shortened as the Log-Loss is another metric to assess the goodness of a classifier. To calculate the Log-Loss, the classifier must predict each class's probability rather than merely predicting the class labels. The Log-Loss is calculated for each predicted probability and the average loss is then reported as the Log-Loss of the classifier. Mathematically, Log-Loss is expressed as:

Log-Loss

where: is the sample size; is the true class label which is a binary indicator ( or ) that assumes indicator 0 if sample does not have the response of interest (disease absent) and assumes indicator 1 if sample has the response of interest (disease present); is the model’s predicted probability that sample is of class in .

Logarithmic loss provides a steep penalty for predictions that are both confident and wrong [35]. That is, it takes into account the uncertainty of our model prediction based on how much it varies from the actual label. LogLoss has no upper bound and it exists on the range [36]. A LogLoss nearer to indicates higher accuracy, whereas LogLoss that is away from indicates lower accuracy. In general, the least LogLoss gives greater accuracy for the classifier. Therefore, the goal is to minimize the LogLoss, and a perfect classifier would have a Log-Loss near zero while less ideal classifiers would have larger values of Log-Loss.

***Area under a ROC curve* (AUC)**:The Area Under the ROC Curve (AUC) metric is used to calculate the Area under the Receiver Operating Characteristics (ROC) curve. The AUC score is always bounded between zero and one, and a very poor classifier has an AUC of around 0.5 [37]. The AUC of a classifier represents the probability that a classifier will rank a randomly chosen positive observation higher than a randomly chosen negative observation. Thus it is a useful metric even for datasets with highly unbalanced classes [37, 38]. Comparing the performance of different classifiers with the ROC curve is not easy [39. 40]. This is because no scalar value represents the expected performance [41].

**3.0 Analysis and Results**

The predictive performance of machine learning models depends on the structure of the dataset, and proper data preparation will ensure the models work optimally. In this section, we carried out an exploratory data analysis on the Cleveland dataset before training different machine learning models on it.

### **3.0.1 *Data and script availability***

We ran statistical analyses in RStudio with R version 4.0.2. Packages used included the rmarkdown for running the codes chunk by chunk [42], tidyverse for data analysis and visualization [43], and caret for classification training [44, 45]. The raw data and scripts employed in this study are all available on GitHub via <https://github.com/gbganalyst/Heart-disease-paper>.

**3.1 *Exploratory Data Analysis***

The Exploratory Data Analysis (EDA) summarises the main characteristics of the heart disease data by using tables and visuals. We performed several EDA to get more insight about the features that are useful in separating the classes of whether or not a patient will have heart disease before applying machine learning models to do the prediction task.

## 3.1.1  *Visualizing class separation by categorical features*

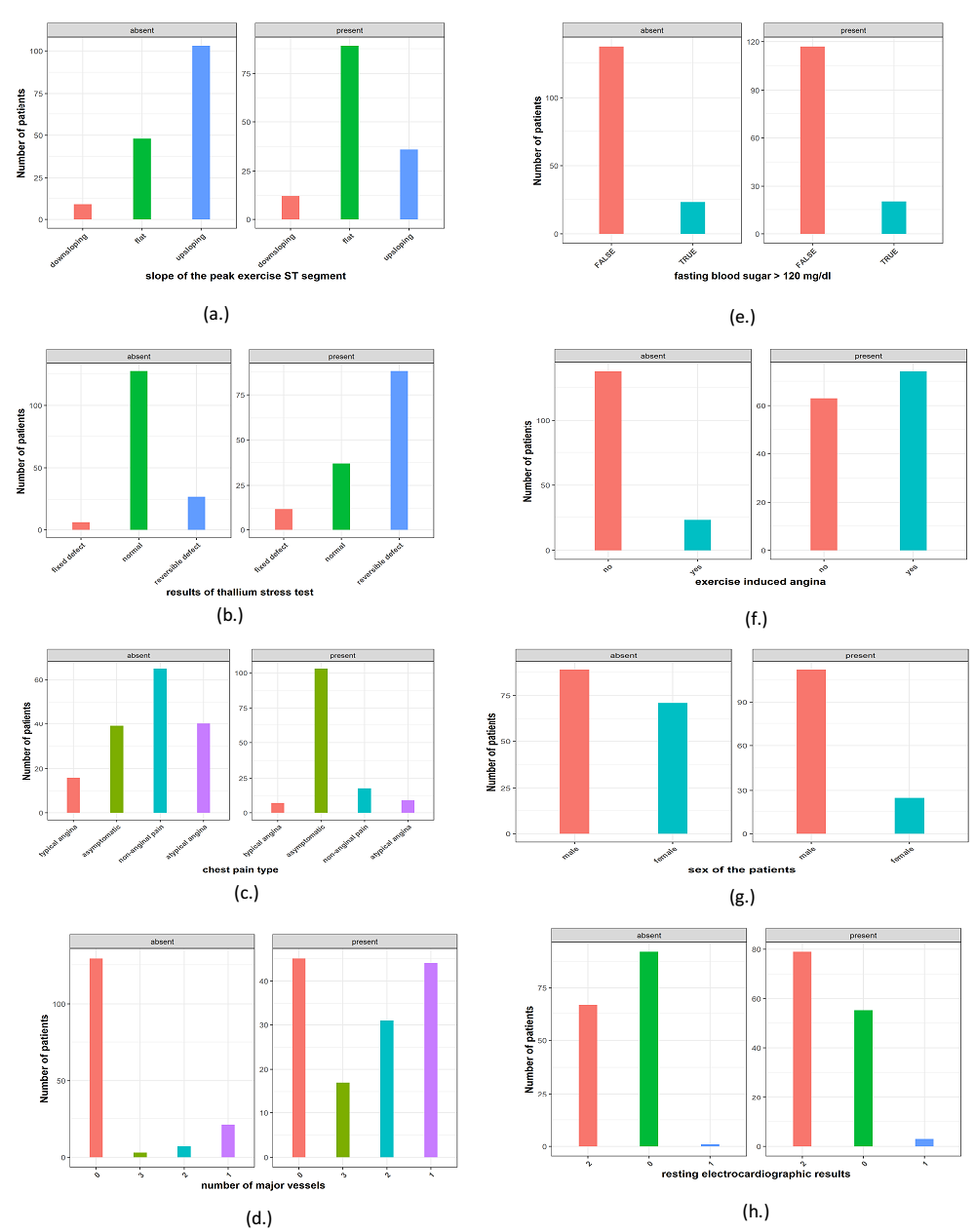
Now we turn to the problem of visualizing categorical features' ability to separate classes of the label. Ideally, a categorical feature will have very different counts for categories in each class of the label. An excellent way to visualize these relationships is with the bar plots faceted by heart disease status of each of the categorical variables, as shown in Figure 2 (a.) to (h.).

The bar plots in Figure 2 contains a lot of information. The key to interpreting these plots is to compare the proportion of the categories for each of the factor levels. If these proportions are distinctly different within each class label, such a feature will likely help separate the class labels.

There are several cases evident in the barplots in Figure 2:

1. Some features such as the slope of the peak exercise ST segment, thallium stress test, chest pain type, and the number of major vessels colored by fluoroscopy showed significant differences in the number of samples that belong to the various factor levels or groups. That is evidence that those factors might be discriminatory of the response class in the data.
2. Other features such as fasting blood sugar, exercise induced angina and sex of the patients show small differences which are unlikely to be significant.
3. Feature like resting electrocardiographic results has dominant categories with very few case of other categories. This feature will likely have very little power to separate the cases.

It is important to note that only a few of these categorical features might help separate the response cases.



**Figure 2 (**(a.) – (h.)**):** Bar plots of features faceted by the class label in the Cleveland data.

**3.1.2 *Visualize class separation by numerical features***

The primary goal of visualization for classification problems is to understand which features are useful for class separation. This section starts by visualizing the discriminatory quality of the metrical (continuous) features in the Cleveland data.



Figure 3. Box plots of the metrical (continuoius) features by class labels in the Cleveland data.

As displayed in Figure 3, the box plots are very useful, and by construction, we focused on the overlap of the quartiles of the distribution. In this case, we might ask the question: is there sufficient differences in the quartiles for the feature to be useful in separating the label classes? All the numerical features in the Cleveland data are useful in separating between the present and absent of heart disease cases. As one might expect, older people tend to have heart disease compared to the younger ones.

**3.2 *Training and evaluation of the classification models***

As remarked earlier in section two, the Cleveland data were used as the training data to build the various classification models in this study. The models’ efficiency and goodness were validated and evaluated on the second independent dataset, the Statlog data.

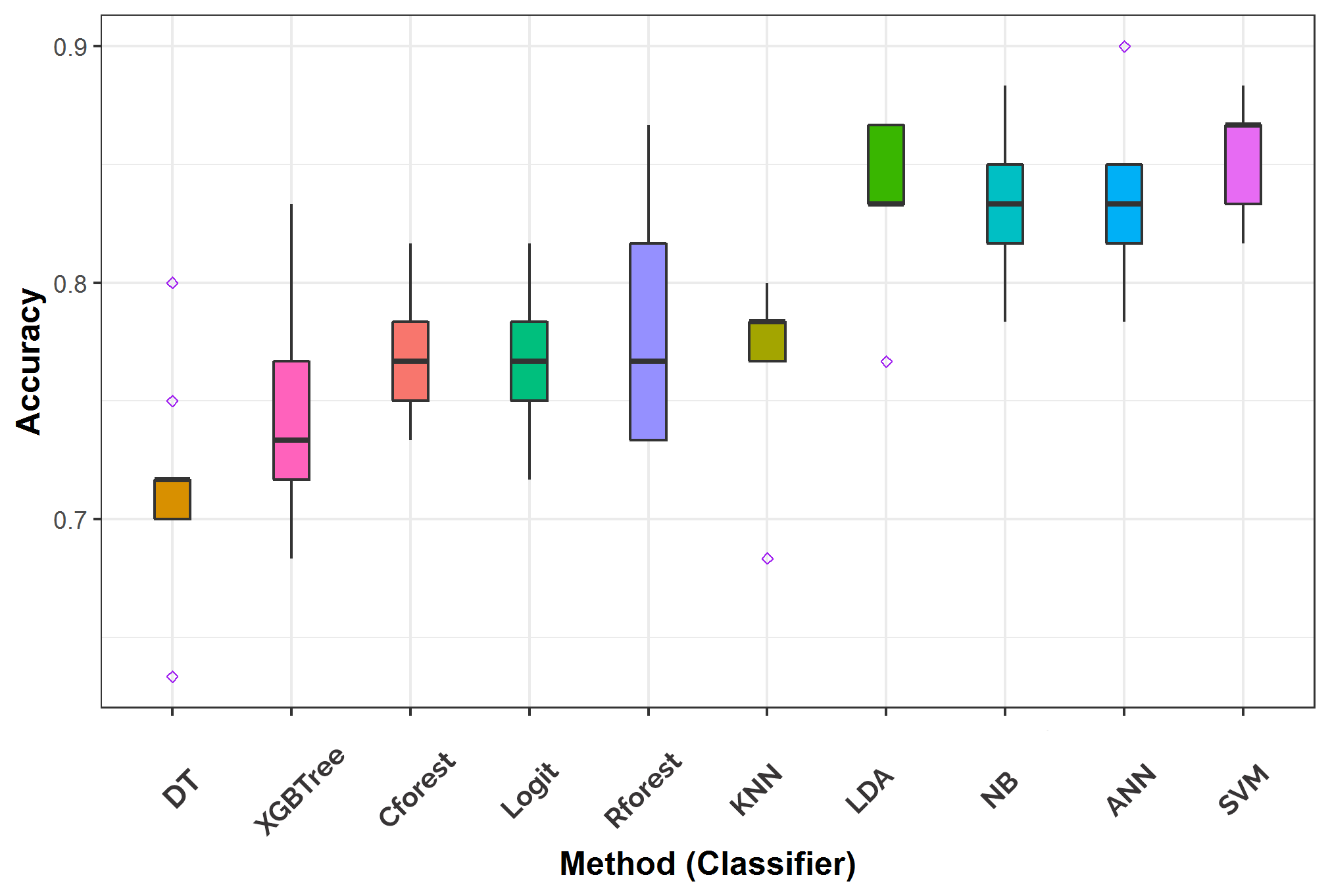
Before applying or fitting any model to the data, it is imperative to perform various data preprocessing. This led us to create a one-hot encoding of our categorical features and rescale numeric features in the dataset so that they can have a similar range of values. Rescaling prevents features from undue influence on the model’s training, especially when the dataset is plagued with a more extensive range of numeric values. Models such as KNN, SVM and neural network required a transformation of features to be centred and scaled before being used [44].

After applying a simple backwards feature selection, also known as recursive feature elimination using random forest on the Cleveland dataset, fifteen (15) out of the nineteen (19) features in the data were selected as the best crop of features for the final models. These selected features include ST depression, maximum heart rate achieved, age of the patients, thallium stress test (normal), serum cholesterol in mg/dl, resting blood pressure in mm Hg, thallium stress test (reversible defect), number of major vessels colored by fluoroscopy (1), exercise induced angina (yes), chest pain type (non-anginal pain), sex of the patients (male), number of major vessels colored by flouroscopy (2), slope of the peak exercise ST segment (upsloping), the slope of the peak exercise ST segment (flat), and resting electrocardiographic results (2) with high predictive values.

During the models’ constructions, 80% of the Cleveland heart disease data were randomly selected and employed for training each model, while the remaining 20% was used as the test data for models’ evaluation over 200 replications. Each model’s performance, averaged over the 200 replications, was determined for each of model’s assessment criteria discussed earlier. However, several tuning of the models’ hyperparameters were performed with the **caret** package in R over 10-fold cross-validation to ensure models stability and efficiency. Without loss of generality, the results of the ten classifiers’ performances analysed here on the test data over 200 replications are shown in Tables 5 through 11. The results shown in various tables were graphically presented in Figures 4 through 10, respectively.

**Table 5:** The results of the prediction performances of all the ten classifiers on the test data over 200 replications based on accuracy. The performances of these classifiers by the mean and median ranking are equally reported. The model with the best prediction performance in mean and/or median ranks are asterisked.

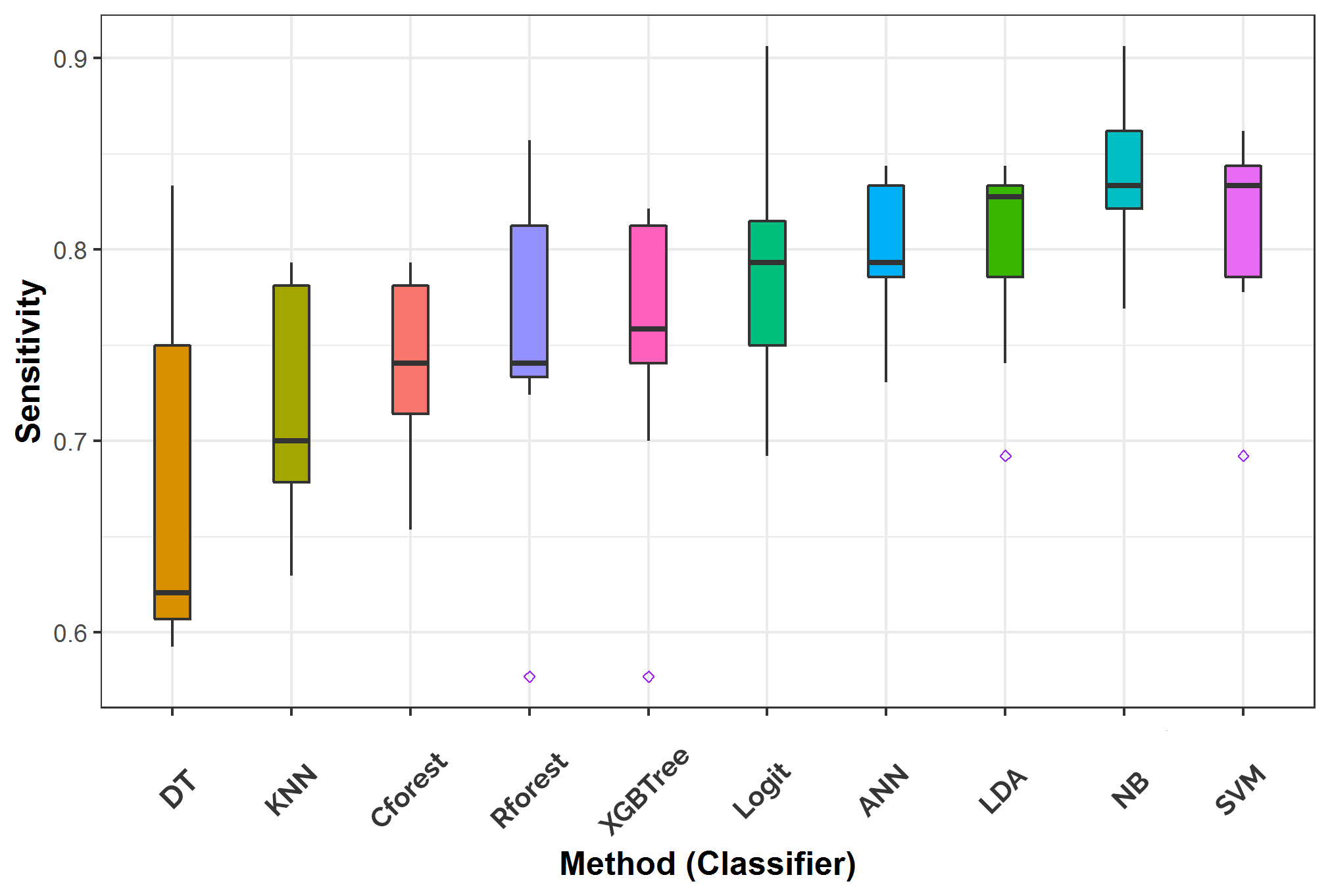
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ML Methods | Accuracy | | | Ranking | | |
| **average** | **median** | **std. dev.** | **average** | **median** | **std. dev.** |
| SVM | 0.8532 | 0.8667 | 0.0246 | 1\* | 1\* | 1\* |
| LDA | 0.8332 | 0.8333 | 0.0365 | 3.5 | 3 | 5 |
| NB | 0.8332 | 0.8333 | 0.0333 | 3.5 | 3 | 4 |
| ANN | 0.8367 | 0.8333 | 0.0387 | 2 | 3 | 6 |
| KNN | 0.7633 | 0.7833 | 0.0415 | 8 | 5 | 8 |
| Cforest | 0.7698 | 0.7667 | 0.0286 | 6 | 7 | 2 |
| Logit | 0.7668 | 0.7667 | 0.0332 | 7 | 7 | 3 |
| Rforest | 0.7827 | 0.7667 | 0.0515 | 5 | 7 | 10 |
| XGBTree | 0.7463 | 0.7333 | 0.0508 | 9 | 9 | 9 |
| DT | 0.7038 | 0.7167 | 0.0393 | 10 | 10 | 7 |



**Figure 4:** The box plots of the performances of all the ten classifiers on the test data over 200 replications based on accuracy.

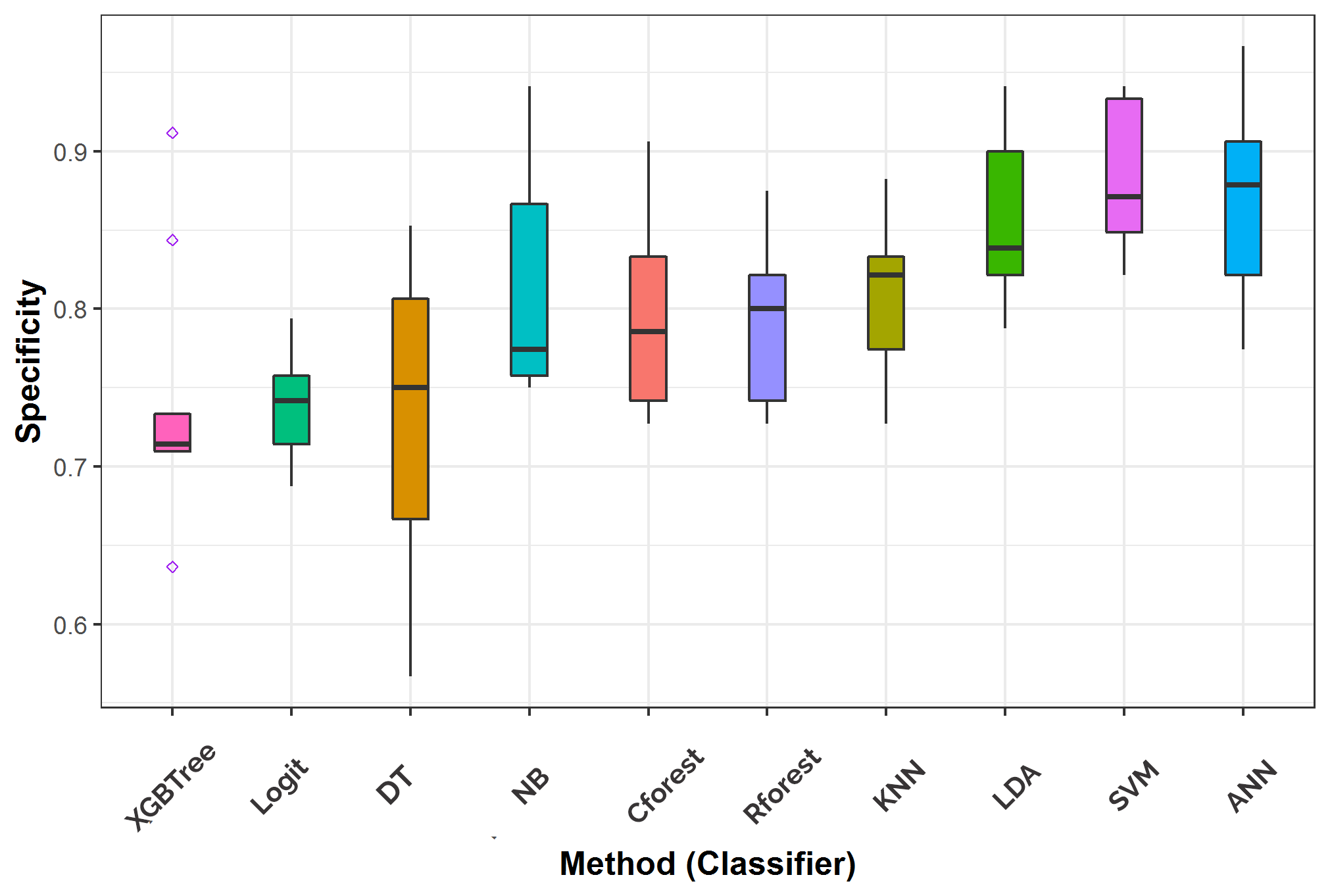
**Table 6:** The results of the prediction performances of all the ten classifiers on the test data over 200 replications based on sensitivity. The performances of these classifiers by the mean and median ranking are equally reported. The model with the best prediction performance in mean and/or median ranks are asterisked.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ML methods | Sensitivity | | | Ranking | | |
| **average** | **median** | **std. dev.** | **average** | **median** | **std. dev.** |
| NB | 0.8473 | 0.8333 | 0.0340 | 1\* | 1.5\* | 1\* |
| SVM | 0.8201 | 0.8333 | 0.0343 | 2 | 1.5\* | 2 |
| LDA | 0.8058 | 0.8276 | 0.0391 | 3 | 3 | 4 |
| Logit | 0.7992 | 0.7931 | 0.0613 | 4 | 4.5 | 8 |
| ANN | 0.7991 | 0.7931 | 0.0372 | 5 | 4.5 | 3 |
| XGBTree | 0.7654 | 0.7586 | 0.0472 | 7 | 6 | 6 |
| Cforest | 0.7389 | 0.7407 | 0.0464 | 8 | 7.5 | 5 |
| Rforest | 0.7722  0.7164 | 0.7407 | 0.0538 | 6 | 7.5 | 7 |
| KNN | 0.7000 | 0.0624 | 9 | 9 | 9 |
| DT | 0.6814 | 0.6207 | 0.0949 | 10 | 10 | 10 |

**Figure 5:** The box plots of the performances of all the ten classifiers on the test data over 200 replications based on sensitivity.

**Table 7:** The results of the prediction performances of all the ten classifiers on the test data over 200 replications based on specificity. The performances of these classifiers by the mean and median ranking are equally reported. The model with the best prediction performance in mean and/or median ranks are asterisked.

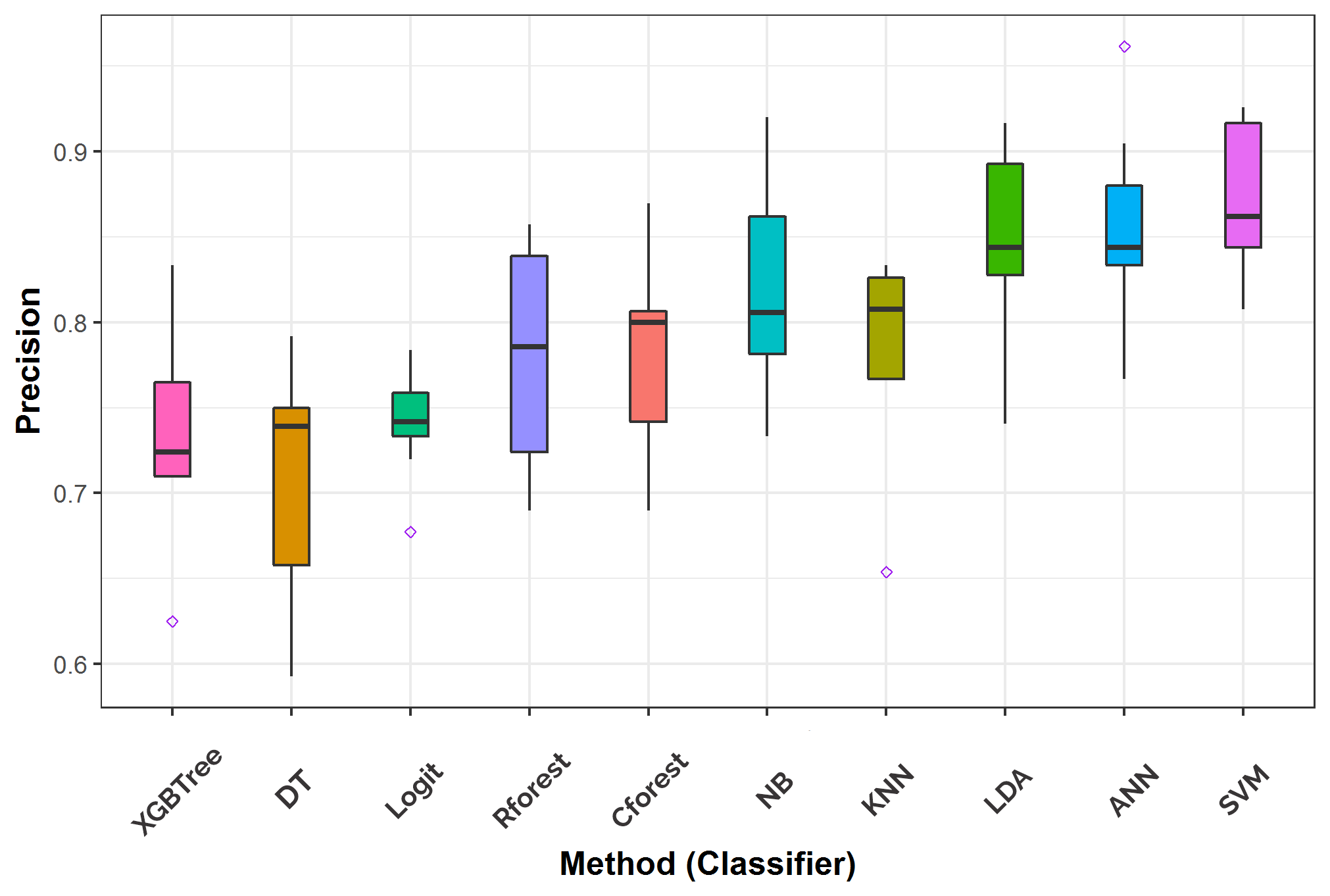
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ML methods | Specificity | | | Ranking | | |
| **average** | **median** | **std. dev.** | **average** | **median** | **std. dev.** |
| ANN | 0.8696 | 0.8788 | 0.0671 | 2 | 1\* | 7 |
| SVM | 0.8824 | 0.8710 | 0.0462 | 1\* | 2 | 2 |
| LDA | 0.8571 | 0.8387 | 0.0544 | 3 | 3 | 5 |
| KNN | 0.8063 | 0.8214 | 0.0511 | 5 | 4 | 3 |
| Rforest | 0.7930 | 0.8000 | 0.0539 | 7 | 5 | 4 |
| Cforest | 0.7988 | 0.7857 | 0.0652 | 6 | 6 | 6 |
| NaiveBayes | 0.8172 | 0.7742 | 0.0735 | 4 | 7 | 9 |
| DT | 0.7207 | 0.7500 | 0.0935 | 10 | 8 | 10 |
| Logit | 0.7341 | 0.7419 | 0.0293 | 8 | 9 | 1\* |
| XGBTree | 0.7278 | 0.7143 | 0.0678 | 9 | 10 | 8 |



**Figure 6:** The box plots of the performances of all the ten classifiers on the test data over 200 replications based on specificity.

**Table 8:** The results of the prediction performances of all the ten classifiers on the test data over 200 replications based on precision. The performances of these classifiers by the mean and median ranking are equally reported. The model with the best prediction performance in mean and/or median ranks are asterisked.

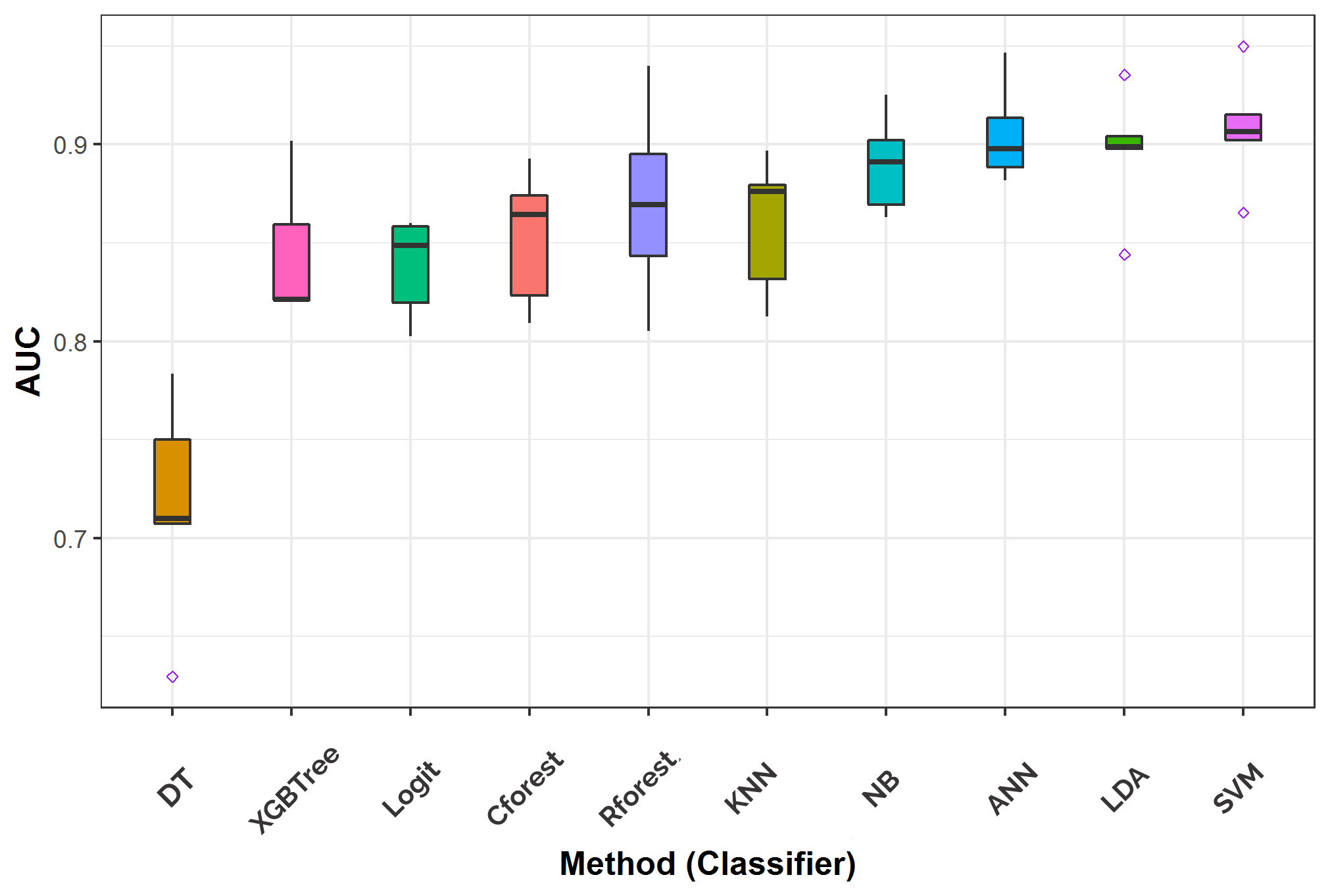
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ML methods | Precision | | | | Ranking | | | | | |
| **average** | **median** | **std. dev.** | **average** | | | **median** | | **std. dev.** | |
| SVM | 0.8711 | 0.8621 | 0.0446 | 1\* | | 1\* | | 2 | |
| LDA | 0.8442 | 0.8438 | 0.0611 | 3 | | 2.5 | | 3.5 | |
| ANN | 0.8572 | 0.8438 | 0.0640 | 2 | | 2.5 | | 5 | |
| KNN | 0.7774 | 0.8077 | 0.0661 | 7 | | 4 | | 9 | |
| NB | 0.8204 | 0.8056 | 0.0649 | 4 | | 5 | | 7 | |
| Cforest | 0.7812 | 0.8000 | 0.0611 | 5 | | 6 | | 3.5 | |
| Rforest | 0.7785 | 0.7857 | 0.0643 | 6 | | 7 | | 6 | |
| Logit | 0.7392 | 0.7419 | 0.0351 | 8 | | 8 | | 1\* | |
| DT | 0.7030 | 0.7391 | 0.0680 | 10 | | 9 | | 10 | |
| XGBTree | 0.7290 | 0.7241 | 0.0651 | 9 | | 10 | | 8 | |



**Figure 7:** The box plots of the performances of all the ten classifiers on the test data over 200 replications based on precision.

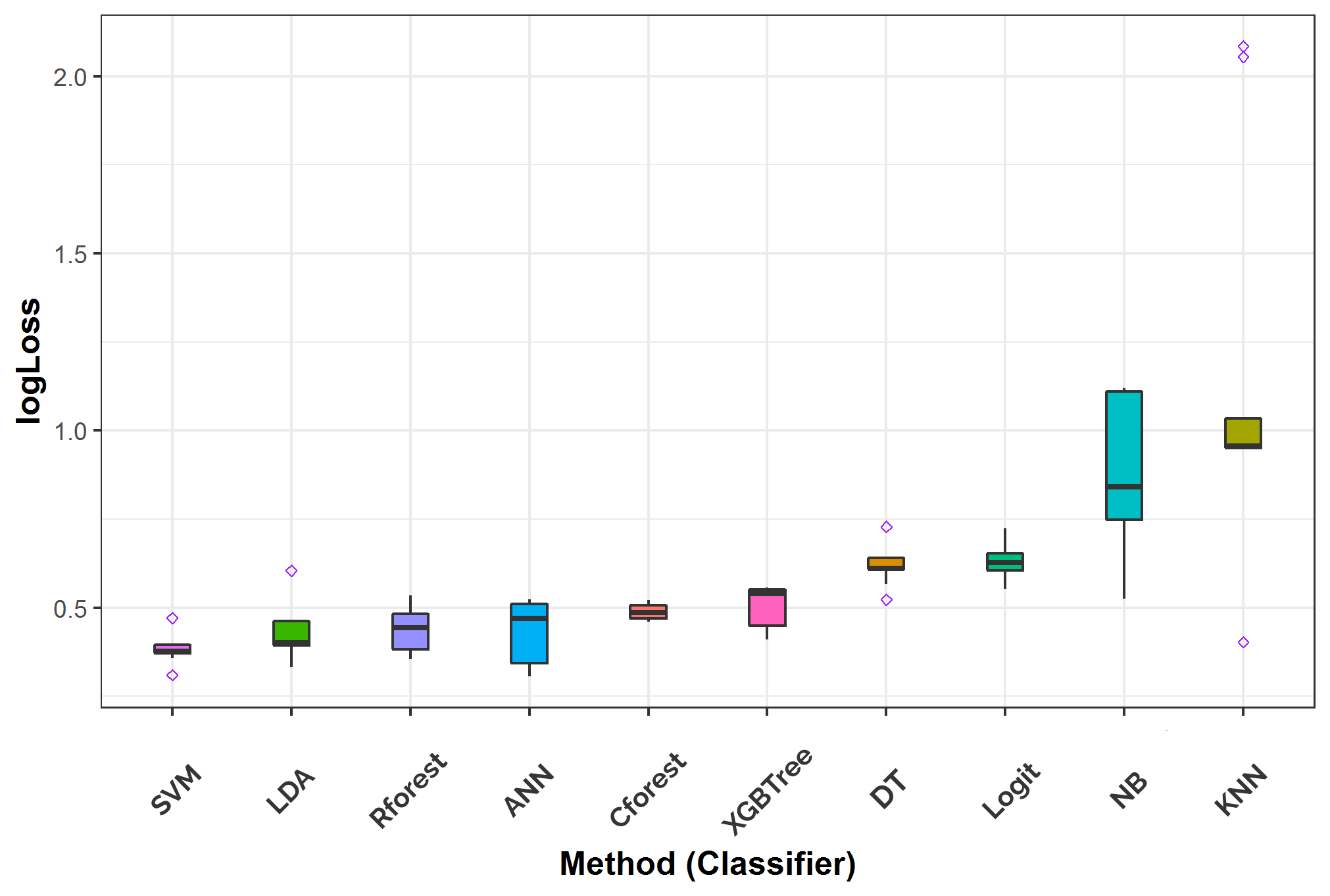
**Table 9:** The results of the prediction performances of all the ten classifiers on the test data over 200 replications based on AUC. The performances of these classifiers by the mean and median ranking are equally reported. The model with the best prediction performance in mean and/or median ranks are asterisked.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ML Methods | AUC | | | Ranking | | |
| **average** | **median** | **std. dev.** | **average** | **median** | **std. dev.** |
| SVM | 0.9076 | 0.9066 | 0.0269 | 1\* | 1\* | 4 |
| LDA | 0.8958 | 0.8989 | 0.0293 | 3 | 2 | 5 |
| ANN | 0.9054 | 0.8977 | 0.0230 | 2 | 3 | 3 |
| NB | 0.8901 | 0.8910 | 0.0225 | 4 | 4 | 1\* |
| KNN | 0.8590 | 0.8761 | 0.0318 | 6 | 5 | 7 |
| Rforest | 0.8702 | 0.8694 | 0.0455 | 5 | 6 | 9 |
| Cforest | 0.8525 | 0.8644 | 0.0315 | 7 | 7 | 6 |
| Logit | 0.8379 | 0.8485 | 0.0227 | 9 | 8 | 2 |
| XGBTree | 0.8446 | 0.8211 | 0.0321 | 8 | 9 | 8 |
| DT | 0.7164 | 0.7098 | 0.0518 | 10 | 10 | 10 |

**Figure 8:** The box plots of the performances of all the ten classifiers on the test data over 200 replications based on AUC.

**Table 10:** The results of the prediction performances of all the ten classifiers on the test data over 200 replications based on Logloss. The performances of these classifiers by the mean and median ranking are equally reported. The model with the best prediction performance in mean and/or median ranks are asterisked.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ML Methods | LogLoss | | | Ranking | | |
| **average** | **median** | **std. dev.** | **average** | **median** | **std. dev.** |
| SVM | 0.3848 | 0.3767 | 0.0516 | 1\* | 1\* | 2 |
| LDA | 0.4390 | 0.4017 | 0.0927 | 3 | 2 | 8 |
| Rforest | 0.4398 | 0.4430 | 0.0651 | 4 | 3 | 6 |
| ANN | 0.4308 | 0.4693 | 0.0890 | 2 | 4 | 7 |
| Cforest | 0.4894 | 0.4868 | 0.0231 | 5 | 5 | 1\* |
| XGBTree | 0.5022 | 0.5401 | 0.0598 | 6 | 6 | 5 |
| DT | 0.6304 | 0.6103 | 0.0545 | 7 | 7 | 3 |
| Logit | 0.6330 | 0.6282 | 0.0561 | 8 | 8 | 4 |
| NB | 0.8700 | 0.8418 | 0.2252 | 9 | 9 | 9 |
| KNN | 1.0935 | 0.9559 | 0.5515 | 10 | 10 | 10 |



**Figure 9:** The box plots of the performances of all the ten classifiers on the test data over 200 replications based on Log-Loss.

**Table 11:** The summary of the prediction performances of all the classifiers on the test data by their median ranks. Models with the best predictive performance in ranks for each model’s assessment metric are asterisked.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ML Methods | Median rank | | | | | |
| **Accuracy** | **Sensitivity** | **Specificity** | **Precision** | **AUC** | **Log-Loss** |
| Cforest | 7 | 7.5 | 6 | 6 | 7 | 5 |
| DT | 10 | 10 | 8 | 9 | 10 | 7 |
| KNN | 5 | 9 | 4 | 4 | 5 | 10 |
| LDA | 3 | 3 | 3 | 2.5 | 2 | 2 |
| Logit | 7 | 4.5 | 9 | 8 | 8 | 8 |
| NB | 3 | 1.5\* | 7 | 5 | 4 | 9 |
| ANN | 3 | 4.5 | 1\* | 2.5 | 3 | 4 |
| Rforest | 7 | 7.5 | 5 | 7 | 6 | 3 |
| SVM | 1\* | 1.5\* | 2 | 1\* | 1\* | 1\* |
| XGBTree | 9 | 6 | 10 | 10 | 9 | 6 |

From the various results obtained as presented in Tables 5 to 11 and Figures 4 to 9, it is quite clear that the best model that provided the best prediction performance among all the ten classifiers considered is the SVM. The SVM outperformed other classifiers, as evident by the results from all the assessment metrics except for the Specificity where SVM came 2nd, as shown in Table 11.

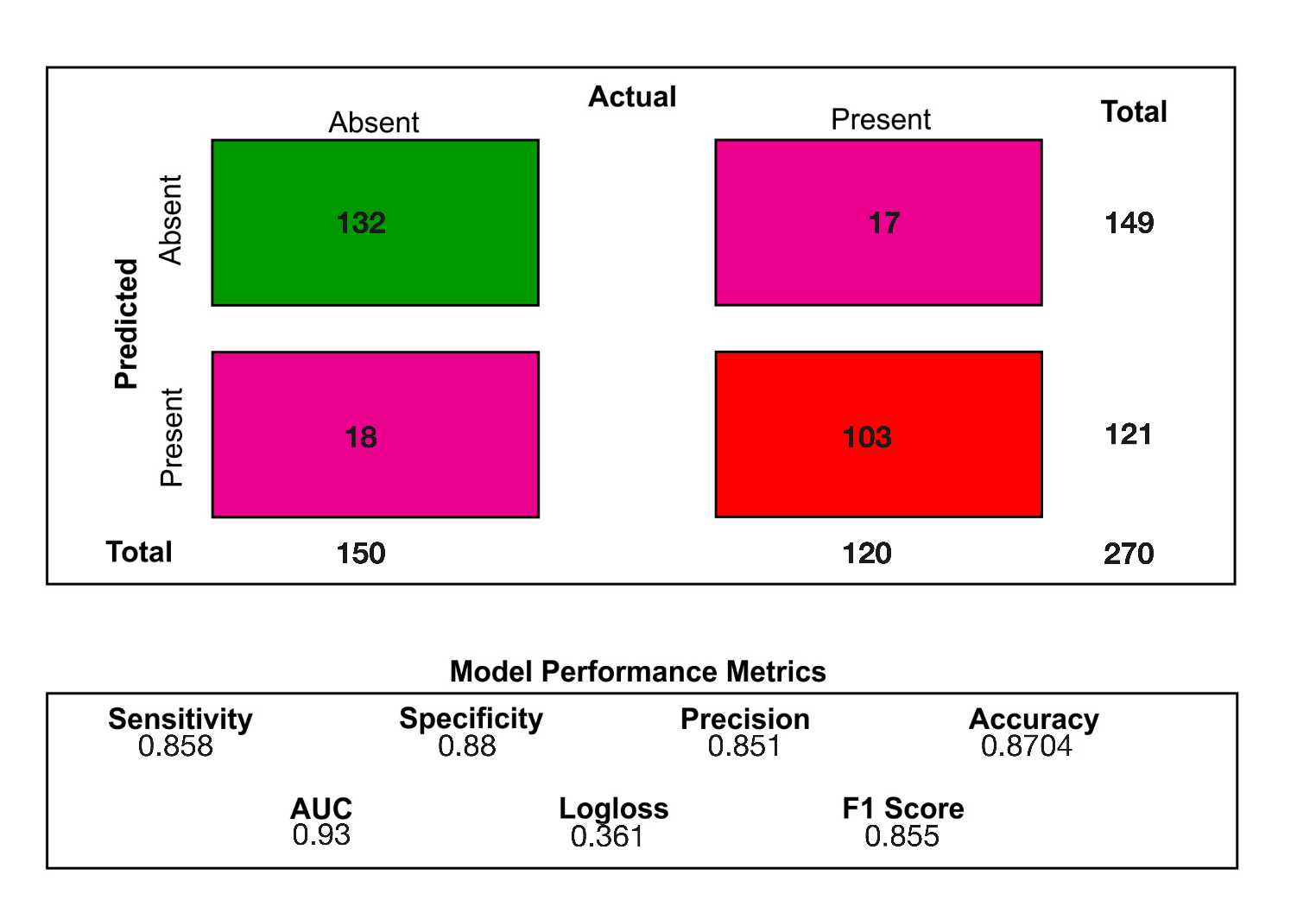
***Validation Results***

As remarked in Section 2, we validated the classification results obtained from the ten models fitted to the Cleveland data using the Statlog data.

The results obtained from the test data showed that the best classifier as chosen by all the models' assessment criteria adopted in this study is the SVM. This final chosen model (SVM) 's prediction performances, as reported in Tables 5 to 11, revealed that the SVM yielded good predictive validation results with 85% Accuracy, 87% Precision, 82% Sensitivity, 88% Specificity, 91% AUC, and 38% Log-Loss value.

The results of the final model (SVM) chosen through the training-testing model’s evaluation procedures are validated on Statlog heart disease data, and the results obtained are presented in pictorial form, as shown in Figure 10.

Prediction results from the validation data showed that the best-chosen model (SVM) yielded an accuracy of 87.04%, Precision of 85.1%, Sensitivity of 85.8%, Specificity of 88%, AUC of 93%, Logloss of 36.1%, and F1-score of 85.5%. These prediction results are quite similar to the SVM model results on the test data as earlier reported.



**Figure 10:** Prediction performance of the best model (SVM) on validation data (Statlog heart disease data).

**4.0 Discussion of results**

The prediction performances of selected ten state-of-the-art machine learning methods are examined in this work for predicting the heart disease status (present or absent) of groups of patients from two real-life publicly available data sets. Evaluation metrics are fundamental in assessing the quality and performance of machine learning models. From the results in Figure 4 to 9 and Table 5 to 11, we observed that choices made by one metric in evaluating the performance of a model is quite different from the choices made by another, but in overall, the Support Vector Machine (SVM) always perform better than all other classifiers under consideration both in the test and validation datasets. And as encouraged by both [13, 32] to try different models before resorting to select the best model in learning the task of heart disease classification, we compared ten models based on assumptions and according to seven evaluation metrics.

Our decision to choose SVM as the best model also confirmed the results of [10] in using SVM to predict heart disease. Results of Chi-square test of association showed that the following bio-clinical categorical variables: Chest pain type, Exercise Induced Angina, Slope of the peak exercise ST segment, Number of major vessels colored by fluoroscopy, and Thalliumstress test are all strongly associated with the heart disease conditions of the patients in the two data sets (p < 0.001).

**5.0 Conclusion**

We have investigated the possibility of using machine learning to predict an instance of heart disease in this paper. Out of the ten classification models evaluated on the heart disease datasets, we found the SVM method to be most suitable to predict the health condition (present or absent) of heart disease patients given a set of bio-clinical variables. We validated the performance level of the SVM on the Statlog data set and obtained appreciable similar predictive results. The SVM model could be adopted in future to analyse data sets with a similar structure for better efficiency. The application of artificial intelligence in predictive medicine will help us flag risk factors so that physicians can work together with patients to reduce the chances of future problems. For instance, patients with a greater risk of heart attacks and irregularities could receive more regular EKGs and cardiologist appointments to ensure the best possible quality of life.

**References**

[1] Kochanek KD, Murphy SL, Xu JQ, Arias E. Deaths: Final data for 2017. *National Vital Statistics Reports*. 2019; 68(9). Available at: https://stacks.cdc.gov/view/cdc/79486 [Accessed 2020 February 2020].

[2] Hannah R, Max R. *Causes of Death*. [internet] 2018. Available at: https://ourworldindata.org/causes-of-death [Accessed 2020 February 2020].

[3] Patil SB, Kumaraswamy YS. Extraction of significant patterns from heart disease warehouses for heart attack prediction. *IJCSNS*. 2009; 9(2):228-35.

[4] Fida B, Nazir M, Naveed N, Akram S. Heart disease classification ensemble optimization using genetic algorithm. In 2011 IEEE 14th International Multitopic Conference 2011 Dec 22 (pp. 19-24).

[5] Anderson RN, Smith BL. Deaths: leading causes for 2001. National Vital Statistics Reports. 2003; 52 (9): 1-86.

[6] Nahar J, Imam T, Tickle KS, Chen YP. Computational intelligence for heart disease diagnosis: A medical knowledge driven approach. *Expert Systems with Applications*. 2013; 40(1):96-104.

[7] Vanisree K, Singaraju J. Decision support system for congenital heart disease diagnosis based on signs and symptoms using neural networks. International Journal of Computer Applications. 2011; 19(6):6-12.

[8] Patel SB, Yadav PK, Shukla DP. Predict the diagnosis of heart disease patients using classification mining techniques. IOSR Journal of Agriculture and Veterinary Science (IOSR-JAVS). 2013;4(2):61-4.

[9] Yahya WB, Rosenberg R, Ulm K. Microarray-based Classification of Histopathologic Responses of Locally Advanced Rectal Carcinomas to Neoadjuvant Radiochemotherapy Treatment. Turkiye Klinikleri Journal of Biostatistics. 2014; 6(1): 8-23.

[10] Pouriyeh S, Vahid S, Sannino G, De Pietro G, Arabnia H, Gutierrez J. A comprehensive investigation and comparison of Machine Learning Techniques in the domain of heart disease. In 2017 IEEE Symposium on Computers and Communications (ISCC) 2017 (pp. 204-207).

[11] Latha CB, Jeeva SC. Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. Informatics in Medicine Unlocked. 2019; 16:100203.

[12] Thenmozhi K, Deepika P. Heart disease prediction using classification with different decision tree techniques. International Journal of Engineering Research and General Science. 2014; 2(6):6-11.

[13] Ogundepo EA, Fokoué E. An empirical demonstration of the no free lunch theorem. Math. Appl. 2019; 8:173-88.

[14] Janosi A, Steinbrunn W, Pfisterer M, Detrano R. Heart disease data set. The UCI KDD Archive. 1988.

[15] Ayodele TO. Types of machine learning algorithms. New advances in machine learning. 2010; 3:19-48.

[16] Simeone O. A very brief introduction to machine learning with applications to communication systems. IEEE Transactions on Cognitive Communications and Networking. 2018; 4(4):648-64.

[17] Russell SJ, Norvig P. Artificial intelligence: a modern approach. Kuala Lumpur, Malaysia; Pearson Education. 2016.

[18] Song YY, Ying LU. Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry. 2015; 27(2):130.

[19] Friedman JH. Greedy function approximation: a gradient boosting machine. Annals of statistics. 2001: 1189-232.

[20] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016 (pp. 785-794).

[21] Strobl C, Zeileis A. Party on!–A new, conditional variable importance measure for random forests available in party. [Internet] 2009. Available at: https://www.r-project.org/conferences/useR-2009/slides/Strobl+Zeileis.pdf [Accessed 2020 February 2020].

[22] Hapfelmeier A, Yahya WB, Rosenberg R, Ulm K. Predictive modelling of gene Expression data. Handbook of Statistics in Clinical Oncology, 3rd ed., Edited by Crowley, J. and A. Hoering. Chapman and Hall/CRC, New York. 2012:463-75.

[23] Manly BF, Alberto JA. Multivariate statistical methods: a primer. CRC press. 2016

[24] Yahya WB, Ulm K, Ludwig F, Hapflemeir A. k-SS: A sequential feature selection and prediction method in microarray study. International Journal of artificial intelligence. 2011; 6(S11):19-47.

[25] Breiman L. Random forests. Machine learning. 2001; 45(1):5-32.

[26] Nisbet R, Elder J, Miner G. Handbook of statistical analysis and data mining applications. Academic Press; 2009.

[27] O'Shea K, Nash R. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458. 2015.

[28] Zou J, Han Y, So SS. Overview of artificial neural networks. In Artificial Neural Networks 2008 (pp. 14-22). Humana Press.

[29] Yahya WB, Oladiipo MO, Jolayemi ET. A fast algorithm to construct neural networks classification models with high-dimensional genomic data. Annals. Computer Science Series. 2012; 10(1):39-58.

[30] Kouiroukidis N, Evangelidis G. The effects of dimensionality curse in high dimensional knn search. In 2011 15th Panhellenic Conference on Informatics 2011 (pp. 41-45). IEEE.

[31] McLachlan GJ. Discriminant analysis and statistical pattern recognition. John Wiley & Sons; 2004

[32] Brownlee J. Master Machine Learning Algorithms: discover how they work and implement them from scratch. Machine Learning Mastery. 2016

[33] Liu H, Roeder K, Wasserman L. Stability approach to regularization selection (stars) for high dimensional graphical models. In Advances in neural information processing systems. 2010 (pp. 1432-1440).

[34] Baeza-Yates R, Ribeiro-Neto B. Modern information retrieval. New York: ACM press; 1999.

[35] Ferri C, Hernández-Orallo J, Modroiu R. An experimental comparison of performance measures for classification. Pattern Recognition Letters. 2009; 30(1):27-38.

[36] Buja A, Stuetzle W, Shen Y. Loss functions for binary class probability estimation and classification: Structure and applications. Working draft, 2005; 3.

[37] Fawcett T. An introduction to ROC analysis. Pattern recognition letters. 2006. 27(8):861-74.

[38] Prati RC, Batista GE, Monard MC. Class imbalances versus class overlapping: an analysis of a learning system behavior. InMexican international conference on artificial intelligence 2004 Apr 26 (pp. 312-321). Springer, Berlin, Heidelberg.

[39] Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition. 1997; 30(7):1145-59.

[40] Drummond C, Holte RC. What ROC Curves Can't Do (and Cost Curves Can). In ROCAI 2004. (pp. 19-26).

[41] Tharwat A. Classification assessment methods. Applied Computing and Informatics. 2020.

[42] Allaire JJ, Xie Y, Jonathan M, Javier L, Kevin U, Aron A, et al. rmarkdown: Dynamic Documents for R, 2019. R package version 1.15.

[43] Wickham H. tidyverse: Easily install and load the ‘tidyverse’.  
R Core Team: Vienna, Austria. 2017. R package version 1.2.1

[44] Kuhn M. Building predictive models in R using the caret package. Journal of statistical software. 2008; 28(5):1-26.

[45] Kuhn M. caret: Classification and Regression Training. 2019. R package version 6.0-84.

[46] Dua D, Graff C. UCI machine learning repository. 2017.