RESEARCH

Exploaratory data analysis on the Cleaveland heart disease dataset and training 4 classifiers to detect coronary artery disease

Yunus Emre Kurnaz, Aakash Nepal, Cuong Gia Pham and Evelina Gudauskayte — Group 10

Full list of author information is available at the end of the article

Abstract

The goal of this first project was to get a general overview of the large field of data science. Important terminologies, plots as well as methods from statistics should be explained and brought closer.

However, the main focus was on the selection and application of a suitable classifiers based on the research question.

Linear regression, Decision trees, k-Nearest-Neighbor, Gaussian Naive Bayes and Random Forest are a set of methods, that can be used to identify and assign unknown variables (observations) to a given dataset.

In this project we have learned, that it is not possible to use one classifier for all problems, but that there are multiple approaches, depending on the given data set and the research question. In addition, we learned the ability to deal critically with a confusion matrix and its concepts such as the importance of specificity, accuracy and sensitivity, as well as the interpretation of ROC curves and their implementation.

A total of 8-10 hours were needed to determine the final results.

1 Scientific Backround

When using a classifier, data automatically gets categorized into one of many classes. These classifications are created by algorithms. This procedure is called machine learning. Predictive classification models approximate a function (f) that relates input variables (X) to discrete output variables (Y). These methods are also used in many biochemical and diagnostic medical applications, as they can be used to make fairly accurate predictions. However, this requires quite a lot of clinical data to achieve a suitable prediction. Since this cannot always be guaranteed, the current incentive is to categorize new patient data based on old existing data. The Cleveland database offers one possibility to get familiar with machine learning. In this study, the collected data were used to determine coronary artery disease (CAD) using a logistic regression. CAD is characterized by narrowing of the arteries that supply oxygen to the heart muscle. CAD can be acute or chronic. In the acute form, a heart attack occurs because a blood clot blocks one or more coronary arteries. As a result, part of the heart muscle does not receive oxygen. In chronic CAD, a coronary vessel is permanently narrowed. As a result, less blood flows to the heart muscle than normal. During physical exertion, the heart is then unable to beat more forcefully because it receives less oxygen. This can lead to symptoms such as shortness of breath and a feeling of tightness in the chest.

2 Goal of the project

The goal of this first project is to become familiar with statistical terminology, to perform an exploratory data analysis for a the Cleveland dataset and to train at least three classifiers to diagnose heart diseases. This includes standard procedures such as creating and interpreting boxplots, pairplots, etc. to get a general overview of the data. Here, we take a practical approach to get a basic overview of machine learning, rather than a theoretical (mathematical) approach.

It is important to familiarize with the topic of classifiers. It is needed for the prediction of a qualitative response (class) for an observation. However, not every classifier is equally applicable to every question; they may have different limitations based on the research question.

In this first project, we practiced and programmed different approaches to classifiers and their interpretation on a ”real life problem” in order to identify possible differences within these classes, while also performing a simple machine learning algorithm to train the dataset, which can determine whether or not a heart disease can be diagnosed.

3 Data and Preprocessing

The multivariate dataset used here, named ”Cleveland heart disease,” contains clinical and biomedical features of more than 300 patients. It was published on the Donald Bren School of Information and Computer Sciences website as a primary source of machine learning datasets. Of the original 76 attributes, only 14 attributes were used for further analysis in most cases. However, for this project, only the ”goal” field was important, because it refers to the presence of a heart disease in a patient. An integer value of 0 indicated no presence, whereas integers of 1 to 4 represented the respective severity of an existing heart disease.

The missing values were removed either in python or in R. Another option was to impute the missing values for example with the scikit function: impute.SimpleImputer() or impute.IterativeImputer(). Because the column of a presence or an absence of a heart disease is important, care was taken in the creation of a training and testing data set to ensure that the division of the data sets took this into account. However, the ”target” column shows an imbalance between patients who do not have heart disease and those who have been categorized by severity. In further steps, this column is restructured so that instead of the severity of heart disease, a distinction is made only between existing heart disease (1) and no heart disease (0). Thus, this problem has been changed into a binary problem to compensate for the inequality of the test and training data sizes.

4 Methods

As described in the previous chapters, the analysis steps were performed using two python libraries. The first one is seaborn, a data visualization library based on matplotlib and the second is scikit-learn, an efficient tools for predictive data analysis. The missing values in the dataset were manually removed as decribed in the previous chapter and prepared for further analysis.

The seaborn library offers a number of functions to get a general overview of the data, in order to do a explorative data analysis. From pairplots and correlation matrices to heatmaps; all necessary statistical functions are included. For this purpose, simply concatenate the pandas.dataframe with the respective functions.

However, the actual analysis was performed twice. First, the dataset and the respective features are split into a training and a test dataset, which can be executed with the provided function of the scikit-learn package (”train_test_split())”. Finding a suitable test size is not always guaranteed, so it is often necessary to split the data again with other test sizes.

Afterwards, the generated test and training sets are passed to one of the many classifiers. In this project, Linear Regression, K-nearest Neighbor, Gaussian Naive Bayes, Decision trees and Random Forest algorithms were used as classifiers. The respective confusion matrices can be plottet with the function: ”confusion_matrix()”. It is possible to determine the sensitivity, specificity and accuracy of the selected model and compare it with previously calculated matrices.

Second, the same data set was executed again, but this time with the difference, that instead of the severity of a heart disease, only a binary distinction is made whether a heart disease is present (1) or not (0). This procedure was performed because the size of the ”goal” column is unbalanced. To compensate for this imbalance, we changed the ”goal” column into a binary classification problem, to overcome the imbalance. On the basis of this confusion matrix, again, sensitivity, specificity and accuracy could also be printed.

Finally, the results of the second run were generated using the function ”roc_curve()”, a function to generate a resulting ROC-curve for a given model. The individual results of the classifiers were plotted as bar plots in a separate file.

4.1 Linear Regression

Linear Regression fits a linear model with coefficients x = (x1, . . . , xp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

4.2 K-Nearest Neighbor

The goal of the K-nearest neighbor algorithm is to determine the nearest neighbors of a given query point, so that we can assign a class label to a point.

4.3 Gaussian Naive Bayes

Naive Bayes classifiers are based on the Bayes theorem. One of the assumptions made is the belief of strong independence between features. These classifiers assume that the value of a particular feature is independent of the value of another feature according to a gaussian distribution.

4.4 Decision Trees

It is a mathematical model that can be used to determine decisions. The hierarchical diagram has a tree-like structure and represents a directed decision path. It consists of root, nodes, branches and leaves. The number of decision levels can vary from decision tree to decision tree.

4.5 Random Forest

It combines the results of many different decision trees to make the best possible decision.

5 Results

To make this section more reader-friendly, only the final results are presented below, as well as the ROC curves of the best classifier.
For this, we divide the analysis results into a multi-class and binary part.

5.1 Multiclass metrics

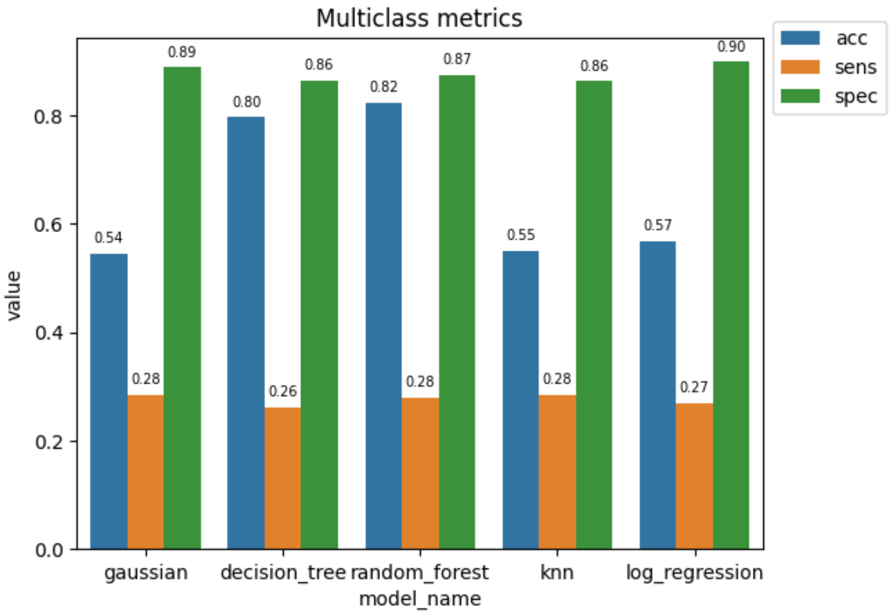

Figure 1 A grouped bar chart of used classifiers (multiclass) regards their accuracy (blue), sensitivity (orange) and specificity (green).

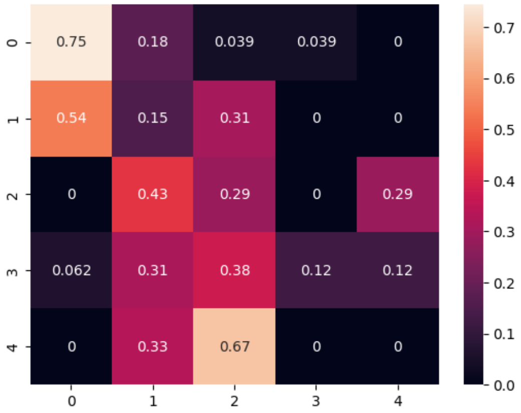

Figure 2 Confusion matrix of the Random Forest classifier (multi-class).

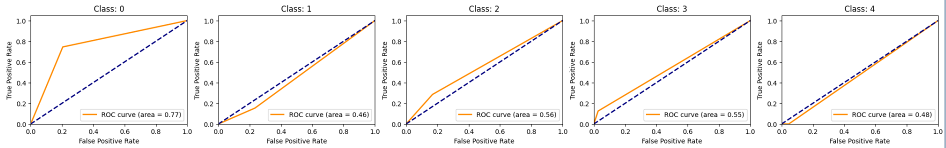

Figure 3 ROC curve of the Random Forest classifier.

Figure 1 shows the different classifiers (x-axis) and their respective values in percent on the y-axis. It is clear that the respective classifiers for the multiclass metric, that distinguish between the severity of heart disease, all produce very similar results. In all cases we have an approximate sensitivity (orange) of 27%. Looking at the results of the accuracy (blue) of the individual classifiers, we get a rather ambiguous result. Decision Tree and Random Forest achieve the highest values of around 80-82% accuracy, this value stagnates at about 55% for the remaining models. A more unambiguous and unanimous result is provided by the specificity (green). All models show an approximately equal result of about 88%.

The ROC curves (Fig. 3) are the results obtained from the confusion matrix (Fig. 2). The confusion matrix (Fig. 2) is a 4x4 matrix that represents the presence of heart disease by its severity or no heart disease. Each row of the matrix represents the instances of an actual class, while each column represents the instances of a predicted class, or vice versa. Looking at the individual ROC curves, which are intended to represent the informative value of this model, it becomes clear that here too, depending on the ”class”, a different result is achieved. In it, the false positive rate and the true positive rate are compared and a curve is drawn using the confusion matrix. In 4 out of 5 cases (Class 1-4), this curve is a diagonal straight line with area under the curve (AUC) scores of 0.46 to 0.56. Only the class ”0” achieved a better score of 0.77, which means, that its curve is located above the original straight line (blue dashed line).

5.2 Binary metrics

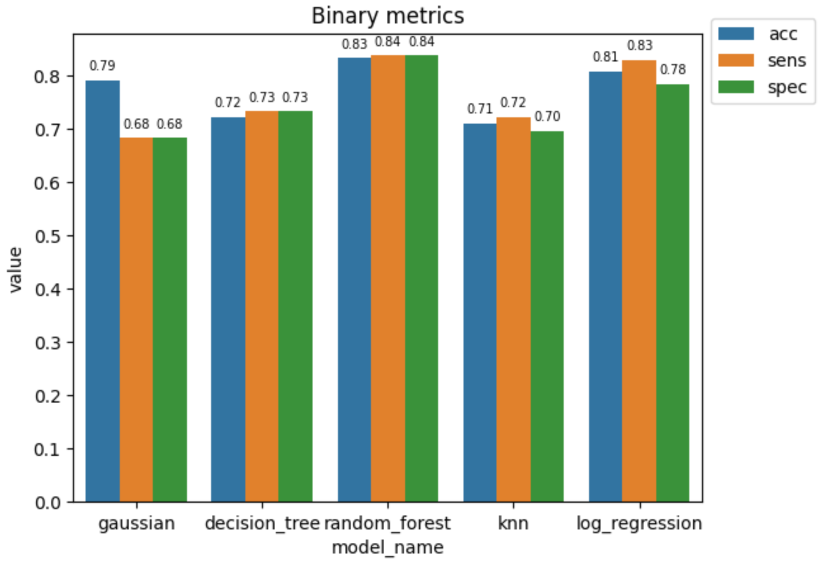

Figure 4 A grouped bar chart used classifiers (binary) regards their accuracy (blue), sensitivity (orange) and specificity (green).

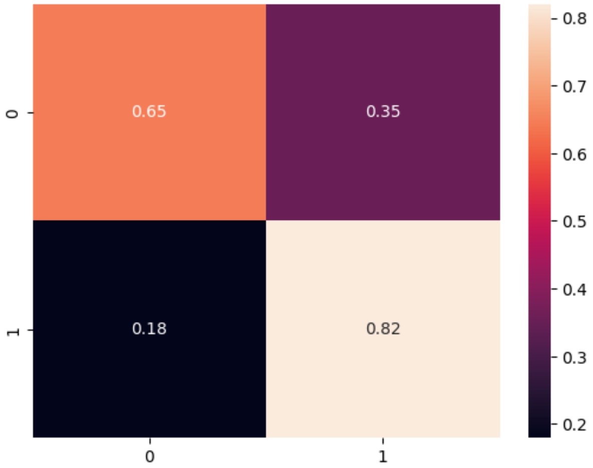

Figure 5 Confusion matrix of the Random Forest classifier (binary).

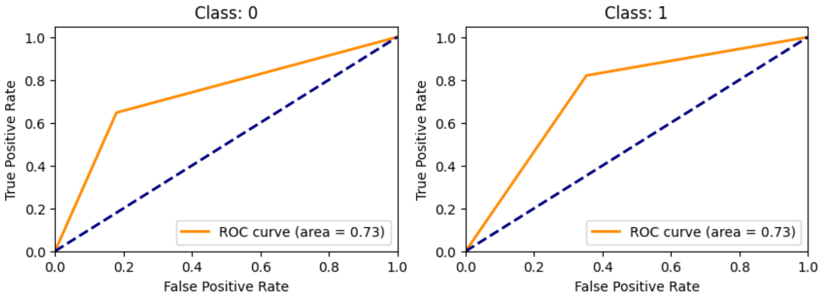

Figure 6 ROC curve of the Random Forest classifier.

By converting the metric into a binary metric (”0”, ”1”), so that only patients with a heart disease and healthy patients can be distinguished, a significantly different result is obtained than in the previous results (Fig. 4). In the grouped bar chart plot (Fig. 4), it is clear that all classifier models now produce a more consistent result. All have approximately the same accuracy (blue), sensitivity (orange) and specificity (green). In all cases, values above 70% are assumed. The best classifier in this case is the Random Forest, where all values are above 83%. It is also clear from the confusion matrix of the Random Forest (Fig. 5), that this is now only a 2x2 matrix, since the values have been adapted to a binary metric. The results from the confusion matrix, shown as ROC curves (Fig. 6), also show a different result. Here, the false positive rate is plotted against the true positive rate and it is made clear that the ROC curve is clearly above the original straight line (blue dashed line) and that the area under the curve assumes values of 0.73.

6 Discussion

The Random Forest performs best when the outcome is binary. This is because the correlation between the different non-zero levels of the outcome variable and the selected features is weaker than the correlation between the zero outcome and the selected features. If we start considering zero and non-zero outcomes, we can separate the data better than if we have to additionally distinguish each non-zero outcome. This effect is best illustrated in figure 1 and 4. This means, that the Random Forest is best adapted to the data.

It is a simple and but quite important topic in statistics or machine learning in general. It illustrates that one classifier cannot always be applied to many problems and that a a one-vs-rest strategy, which converts a multi-class problem into a set of binary tasks for each class in the objective can achieve a better outcome based on the research question.

7 Appendix

Yunus Emre Kurnaz (Bioinformatics) programmed the linear regression classifier and wrote the report.

Aakash Nepal (Bioinformatics) programmed the K-nearest-Neighbor classifier.

Cuong Gia Pham (Bioinformatics) programmed the Gaussian Naive Bayes classifier.

Evelina Gudauskayte (Data Science) programmed the Decision Tree and Random Forest classifiers.