Heart disease prediction using Machine Learning

Student: Yi Qiang Ji Zhang
Professor: Alex Ferrer Ferre

Aerospace Engineering
Numerical Tools in Machine Learning for Aeronautical Engineering
Polytechnic University of Catalonia — BarcelonaTech

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Date: 29 April 2021

I. Introduction

Heart disease prediction is one of the most complicated tasks in medical field. Today, approximately one person per minute dies due to heart diseases [1]. Thus, it is crucial to research further in this field. This is why Machine Learning and Data Science plays a critical role in the field of healthcare. Good data-driven systems for predicting diseases improve the prevention process, making sure people can live serenely lives. Though, this task is rather laborious and complex due to the amount of attributes that has to take into account as well as the huge amount of data needed. The main goal now is to use Machine Learning to help automate the prediction process to avoid the risks associated with it and alert the patient in advance.

This report makes use of heart disease dataset available in UCI machine learning repository [2]. The proposed work shows the prediction capabilities of the chances of Heart Disease by implementing different Classification Learner techniques. Afterwards, different models were trained and predictions are made with different algorithms. Consequently, the performance of different classification learner algorithms will be studied using MATLAB.

This problem is a classification problem, input features are a variety of parameters such as age, blood pressure, etc., and the target variable as a binary variable, predicting whether heart disease is present 1 or not 0.

The trial results shows that *Gaussian Naive Bayes*, *Weighted KNN* and *Ensemble Bagged Trees* have achieved the highest accuracy of 86.7% compared to other Machine Learning algorithms assayed.

Keywords — Heart Disease Prediction, Machine Learning, Classification Learner, Naive Bayes, KNN, Ensemble Bagged Trees.

II. Attribute Information and Classes

14 different attributes were takin into account when prediction the heart disease, below are listed the attributes and their meaning.

1) Age (years)
2) Sex (1 = male; 0 = female)
3) Chest pain type (4 values): label=
   - Value 1: Typical angina
   - Value 2: Atypical angina
   - Value 3: Non-anginal pain
   - Value 4: Asymptomatic
4) Resting blood pressure (mmHg)
5) Serum cholestoral (mg/dl)
6) Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
7) Resting electrocardiographic results (values 0,1,2) label=
   - Value 0: normal
   - Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
   - Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8) Maximum heart rate achieved
9) Exercise induced angina (1 = yes; 0 = no)
10) Oldpeak = ST depression induced by exercise relative to rest
11) The slope of the peak exercise ST segment label=
    - Value 1: Unsloping
    - Value 2: Flat
    - Value 3: Downsloping
12) Number of major vessels (0-3) colored by flourosopy
13) Thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

The class or "goal" field refers to the presence of heart disease in the patient. It is integer valued 0 (no presence) 1 (presence).

III. Classification

Below is presented the accuracy of the different models analyzed:

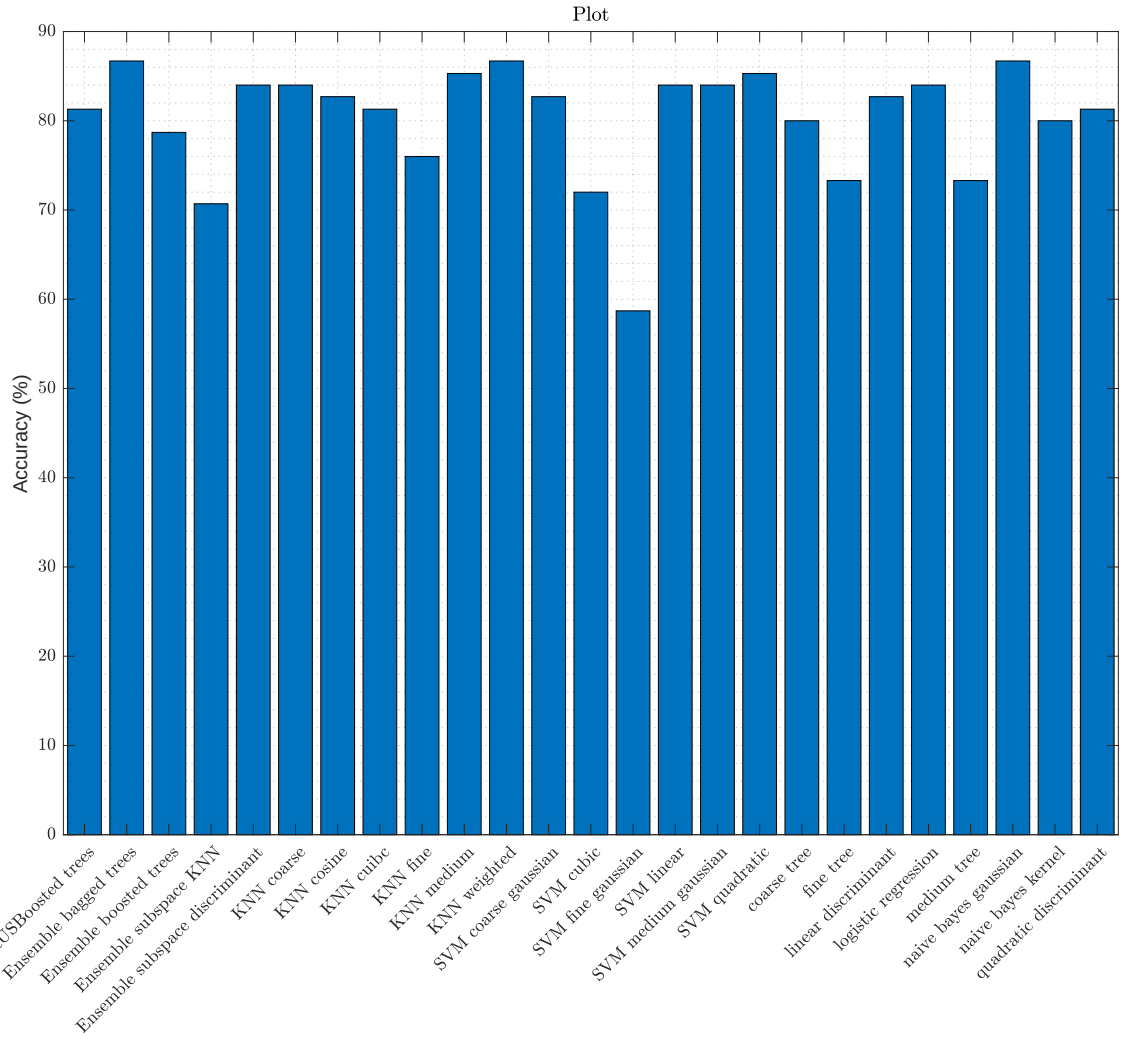

Fig. 1. Accuracy of the classification algorithms. Source: Own.

The accuracy values of each algorithm is:

- fine tree = 73.3;
- medium tree = 73.3;
- coarse tree = 80.0;
- linear discriminant = 82.7;
- quadratic discriminant = 81.3;
- logistic regression = 84.0;
- naive bayes gaussian = 86.7;
- naive bayes kernel = 80.0;
- SVM linear = 84.0;
- SVM quadratic = 85.3;
- SVM cubic = 72.0;
- SVM fine gaussian = 58.7;
- SVM medium gaussian = 84.0;
- SVM coarse gaussian = 82.7;
- KNN fine = 76.0;
- KNN medium = 85.3;
- KNN coarse = 84.0;
- KNN cosine = 82.7;
- KNN cuibc = 81.3;
- KNN weighted = 86.7;
- Ensemble boosted trees = 78.7;
- Ensemble bagged trees = 86.7;
- Ensemble subspace discriminant = 84.0;
- Ensemble subspace KNN = 70.7;
- Ensemble RUSBoosted trees = 81.3;

IV. Results

Below is presented the confusion matrices for 3 of the best classification algorithms found:

A. Gaussian Naive Bayes

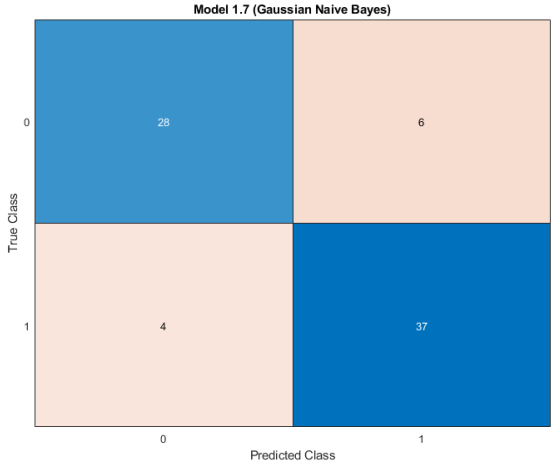

Fig. 2. Confusion matrix of Gaussian Naive Bayes algorithm. Source: Own.

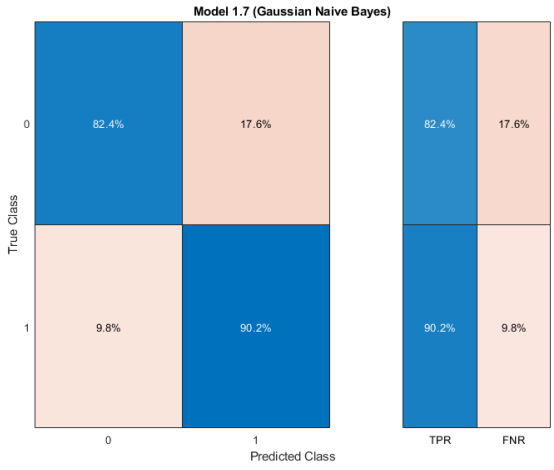

Fig. 3. True Positive Rates and False Negative Rates of Gaussian Naive Bayes algorithm. Source: Own.

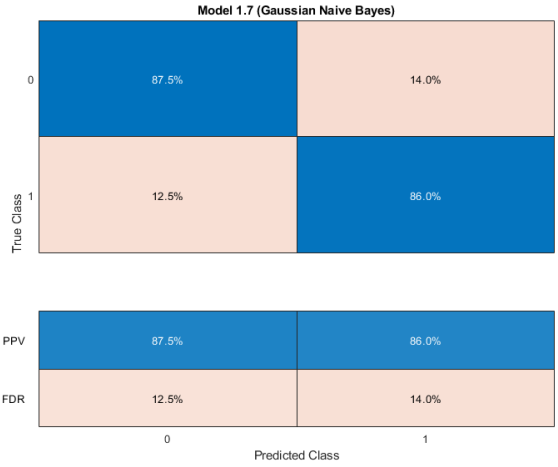

Fig. 4. Positive Predictive Values and False Discovery rates of Gaussian Naive Bayes algorithm. Source: Own.

B. Weighted KNN

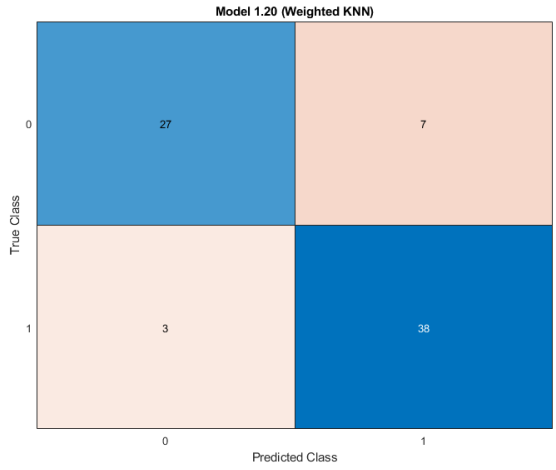

Fig. 5. Confusion matrix of Weighted KNN algorithm. Source: Own.

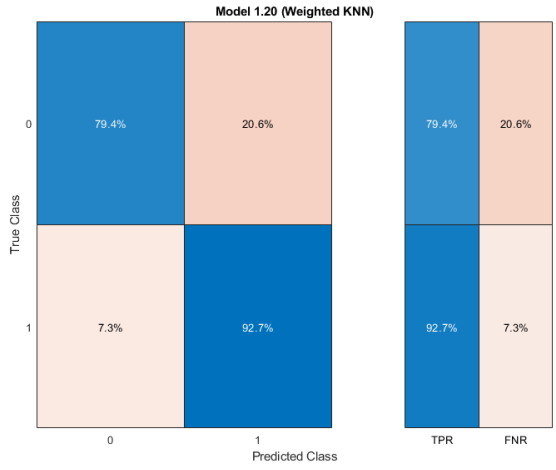

Fig. 6. True Positive Rates and False Negative Rates of Weighted KNN algorithm. Source: Own.

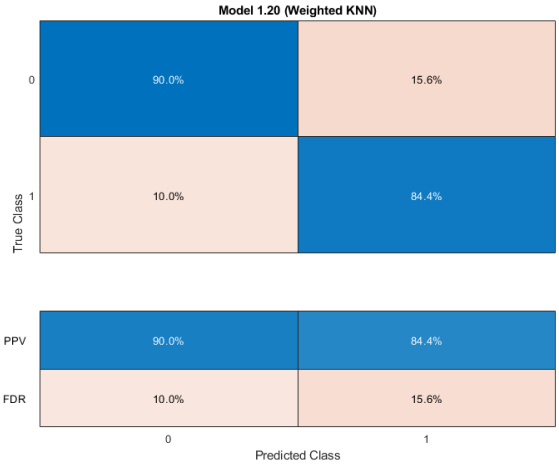

Fig. 7. Positive Predictive Values and False Discovery rates of Weighted KNN algorithm. Source: Own.

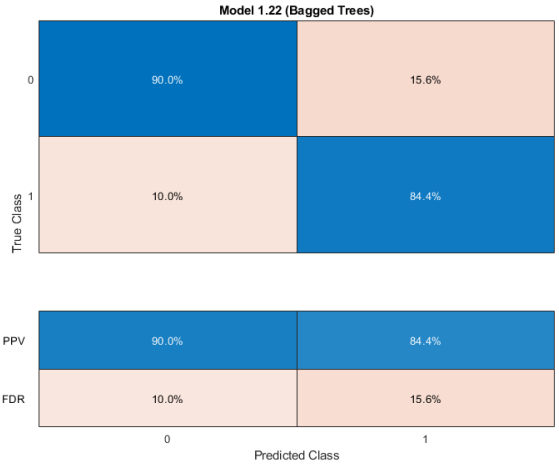

Fig. 10. Positive Predictive Values and False Discovery rates of Bagged Trees algorithm. Source: Own.

C. Bagged Trees

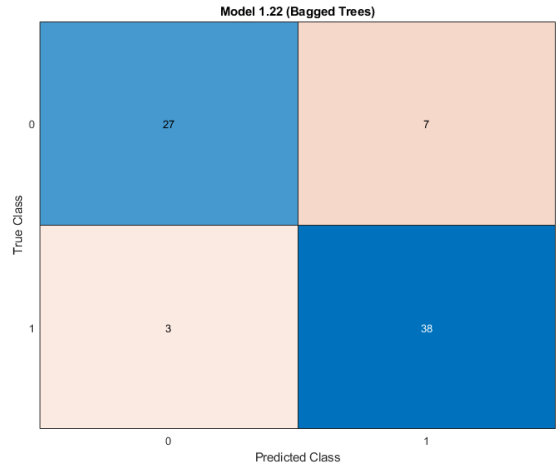

Fig. 8. Confusion matrix of Bagged Trees algorithm. Source: Own.

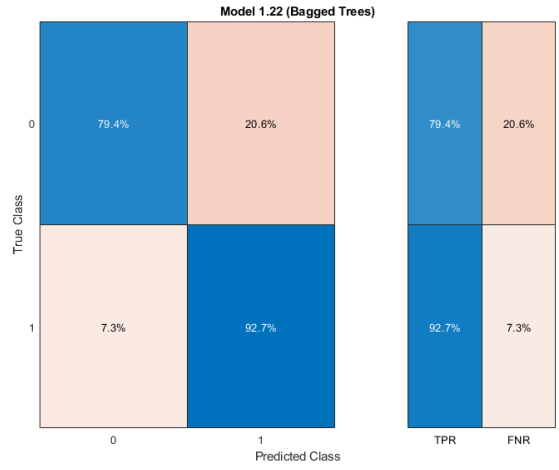

Fig. 9. True Positive Rates and False Negative Rates of Bagged Trees algorithm. Source: Own.

In order to use the classification learner, of the 303 instances, 25% of the data were used to validate the algorithm. The percentage held out is 25 %.

A *confusion matrix* is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. Hence, it allows the visualization of the performance of an algorithm.

For instance, take a look at Figure 2, there are two possible predicted classes: 1 and 0 ('yes' or 'no'). In the case of predicting the disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.

The model used made a total of 75 predictions, and 10 of them failed to predict it in the correct way. The same applies for the other algorithms, where out of 75 predictions, 10 were not correct. This is possible with the leftover 25% data that was set for validation. The validation is extremely important as if there were no data for validating the model, one cannot imply the models accuracy, even if all data of the set were used for it.

Out of those 75 cases, the classifier predicted "yes" 45 times, and "no" 30 times. In reality, 41 patients in the sample have the disease, and 34 patients do not.

Let's define some basic concepts [3],

- **True Positives (TP)**: These are cases in which the classifier predicted the patients have the disease, and they truly do have the disease.
- **True Negatives (TN)**: The classifier predicted no, and they don't have the disease.
- **False Positives (FP)**: The classifier predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **False Negatives (FN)**: The classifier predicted no, but they actually do have the disease. (Also known as a "Type II error.")
- **Positive predictive value (PPV)** is the probability that subjects with a positive screening test truly have the disease.
- **Positive predictive value (NPV)** is the probability that subjects with a negative screening test truly don't have the disease.

Now, in any scenario, specially in diseases, despite

all the above algorithms have the same accuracy, it is always better to have a False Positive rather than a False Negative. This way, it will ensure that the most grievous scenario is that the classifier predicts the patient to have a heart disease but in really they are healthy. Analogously, this information can be retrieved from the PPV and NPV plots.

Consequently, out the 3 most accurate predictors, the most suitable for this case is the Bagged Trees or the KNN algorithm since they share the same minimum False Positive values.

For further analysis, there are other parameters that can be taken into account [3]:

- **Null Error Rate**: This is how often you would be wrong if you always predicted the majority class. This can be a useful baseline metric to compare the classifier against. However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate, as demonstrated by the Accuracy Paradox.
- **Cohen's Kappa**: This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.
- **F Score**: This is a weighted average of the true positive rate (recall) and precision.
- **ROC Curve**: This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class.

References

[1] Apurb Rajdhan et al. "Heart Disease Prediction using Machine Learning". In: *International Journal of Engineering Research and* V9 (May 2020). DOI: 10.17577/IJERTV9IS040614.

[2] UCI Center for Machine Learning and Intelligent Systems. *UCI Machine Learning Repository: Heart Disease Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/Heart+Disease (visited on 04/29/2021).

[3] Dataschool. *Simple guide to confusion matrix terminology*. 2014. URL: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ (visited on 04/29/2021).

V. Code

```
%% Machine Learning
%
%--------------------------------------------------------------------------%
% Heart Disease Data Set
%
% This database contains 76 attributes, but all published experiments refer to using
% a subset of 14 of them. In particular, the Cleveland database is the only one that
% has been used by ML researchers to this date.
%
% The "goal" field refers to the presence of heart disease in the patient.
% It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland
% database have concentrated on simply attempting to distinguish presence
% (values 1,2,3,4) from absence (value 0).
%
% https://archive.ics.uci.edu/ml/datasets/Heart+Disease
% url: https://www.kaggle.com/ronitf/heart-disease-uci
%--------------------------------------------------------------------------%

% Date: 29/04/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: NUMERICAL TOOLS IN MACHINE LEARNING FOR AERONAUTICAL ENGINEERING
% Professor: Alex Ferrer Ferre

% Clear workspace, command window and close windows
clear;
close all;
clc;

% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Analyse data

fine_tree = 73.3;
medium_tree = 73.3;
coarse_tree = 80.0;
linear_discriminant = 82.7;
quadratic_discriminant = 81.3;
logistic_regression = 84.0;
naive_bayes_gaussian = 86.7;
naive_bayes_kernel = 80.0;
SVM_linear = 84.0;
SVM_quadratic = 85.3;
SVM_cubic = 72.0;
SVM_fine_gaussian = 58.7;
SVM_medium_gaussian = 84.0;
SVM_coarse_gaussian = 82.7;
KNN_fine = 76.0;
KNN_medium = 85.3;
KNN_coarse = 84.0;
KNN_cosine = 82.7;
KNN_cuibc = 81.3;
KNN_weighted = 86.7;
Ensemble_boosted_trees = 78.7;
Ensemble_bagged_trees = 86.7;
Ensemble_subspace_discriminant = 84.0;
Ensemble_subspace_KNN = 70.7;
Ensemble_RUSBoosted_trees = 81.3;

x = categorical({'fine tree', 'medium tree', 'coarse tree', 'linear discriminant', ...
    'quadratic discriminant', 'logistic regression', 'naive bayes gaussian', ...
    'naive bayes kernel', 'SVM linear', 'SVM quadratic', 'SVM cubic', ...
    'SVM fine gaussian', 'SVM medium gaussian', 'SVM coarse gaussian', ...
    'KNN fine', 'KNN medium', 'KNN coarse', 'KNN cosine', 'KNN cuibc', ...
    'KNN weighted', 'Ensemble boosted trees', 'Ensemble bagged trees', ...
    'Ensemble subspace discriminant', 'Ensemble subspace KNN', ...
    'Ensemble RUSBoosted trees'});

y = [fine_tree;
medium_tree;
coarse_tree;
linear_discriminant;
quadratic_discriminant;
logistic_regression;
naive_bayes_gaussian;
naive_bayes_kernel;
SVM_linear;
SVM_quadratic;
```

```
SVM_cubic;
SVM_fine_gaussian;
SVM_medium_gaussian;
SVM_coarse_gaussian;
KNN_fine;
KNN_medium;
KNN_coarse;
KNN_cosine;
KNN_cuibc;
KNN_weighted;
Ensemble_boosted_trees;
Ensemble_bagged_trees;
Ensemble_subspace_discriminant;
Ensemble_subspace_KNN;
Ensemble_RUSBoosted_trees]';

% Plot
plot1 = figure(1);
bar(x,y)
% text(1:length(y),y,num2str(y'),'HorizontalAlignment','center','VerticalAlignment','bottom');
set(plot1,'Position',[475 150 1000 800])
ylabel('Accuracy (%)')
title('Plot')
box on
grid minor
hold off;

% Save pdf
set(plot1, 'Units', 'Centimeters');
pos = get(plot1, 'Position');
set(plot1, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
   'PaperSize',[pos(3), pos(4)]);
print(plot1, 'accuracy.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'accuracy.png','-dpng','-r2000');
```