Analysis and performance of model

Instituto Tecnológico y de Estudios Superiores de Monterrey
Carlos Alberto Hurtado Sánchez
A01700885

***Abstract*—This document explores the performance of three decision tree based classifiers on UCI's Heart Disease Data Set.**

***Index Terms*—classifiers, UCI, decision trees**

I. Train, test, and validation sets

This document is a model and instructions for LaTeX. Please observe the conference page limits.

II. Model selection

The goal of the model predictions is to predict whether a patient suffers from a heart disease or not. So three tree-based classifiers were created: a decision tree model, a random forest model, and a hyperparameter fine-tuned random forest model.

All of the three models were created after 13 features, out of which six are categorical.

III. Bias and variance

Decision tree models exhibit a low bias because the algorithms used to create the tree do not make assumptions about the target value since they only take the decision to choose a node from the tree or not based on the highest information gain.

Unfortunately, decision trees do have high variance because the tree they are constructed on is highly dependent on the provided dataset. The later means that if a small variation is introduced in the data, then a different decision tree will be constructed thus resulting in a high variance.

To address the high variance of decision trees ensemble algorithms are incorporated into the tree. By applying the idea of bagging, fitting several independent models and taking the average of their predictions, on decision trees a lower variance is achieved. So with random forest it is possible to have a model with low variance and less susceptible to overfitting.

IV. Model performance

To test the performance of each model, a train and test split was made for each model as well as the construction of a confusion matrix.

Although the performance on the test set and the result of the confusion matrices was not constant, due to the high variance of decision trees, the random forest models showed to have higher accuracy scores on the test sets as well as better results on the confusion matrices.

V. Regularization

Regularization was applied using hyperparameter tuning through Scikit-Learn so that the best combination of hyperparameters is found and a more optimal random forest is created.

Each hyperparameter cobmination, the model is cross validated and scored in order to get the most optimal model based on a combination of predefined hyperparameters. So for this model, the hyperparameters of interest are:

- Number of trees in forest
- Sampling of features (taking the square or the logarithm of the number of features)
- Maximum number of leaf nodes
- Maximum depth of the trees
- Whether trees are bootstrapped or not

Due to the high variance of decision trees, it is not possible to provide a list of the best combination of hyperparameters, but the code is left available in the repository.