Master in Artificial Intelligence

Supervised and Experiential Learning

Ensemble learning: Random Forest and Decision Forest

Romero Antolin, Axel

May 4th, 2023

Contents

1 Introduction

For the second project in the Supervised and Experiential Learning course, we conducted an implementation and validation of two ensemble learning algorithms, the Random Forest and the Decision Forest. In Section 2, we provide an explanation of ensemble learning and the details of both algorithms presented in this project.
The implementation was completed using Python 3, and the corresponding code is accessible within the `source` folder. Additional information regarding the code organization and execution can be found in Section 3. To validate the implementation of the ensemble learning models, we utilized three distinct data sets of varying sizes to assess performance and the ability to classify new instances. The test outcomes can be seen in Section 4.

2 Ensemble learning

Ensemble learning is a machine learning technique that involves combining multiple models to improve the accuracy and performance of predictions. The basic idea behind ensemble learning is that by combining several weaker models, one can create a more accurate and robust model than any individual model. Ensemble learning can be applied to any machine learning algorithm, including decision trees, and can be used for both classification and regression problems.

There are several ways to create an ensemble of models, but two of the most common approaches are bagging and boosting. Bagging involves training multiple models on different subsets of the training data, created by random sampling with replacement. Each model produces its own prediction, and the final prediction is computed by aggregating the predictions of all models. Boosting involves training weak models sequentially, where each model tries to correct the mistakes made by the previous model. This approach focuses on difficult examples that are difficult to classify, which can improve the overall performance of the model. Ensemble learning has several benefits, including improved accuracy, robustness, prevention of overfitting, and interpretability.

In this project, we have to implement two ensemble learning models, which are the Random Forest and Decision Forest classifiers. In a Random Forest, multiple decision trees are trained on bootstrap samples of the data, where each tree is trained on a random subset of the features. The final prediction is made by aggregating the predictions of all the trees, either by majority voting (for classification) or by averaging (for regression). The random selection of features and the use of bootstrap samples help to reduce overfitting and increase the diversity of the trees in the ensemble, which in turn improves the performance of the Random Forest. Overall, the use of the CART algorithm with the Gini index as the impurity measure in Random Forests results in a powerful and flexible machine learning method that can be applied to a wide range of problems. On the other hand, Decision Trees follow the same procedure but the trees are constructed using always the same data, the original data is not bootstrapped. The subset of features is randomly selected for each tree.

The CART algorithm constructs a binary tree recursively by partitioning the input space into subsets based on the values of a feature, so as to minimize the impurity of the resulting subsets. The impurity of a subset is measured by the Gini index, which is a measure of the probability of misclassification if a sample from the subset is randomly labelled according to the class distribution of the subset. The Gini index is defined as:

$$\text{Gini}(X) = 1 - \sum_{i=1}^{k} p_{x \in c_i}^2 \tag{1}$$

where $p_{x \in c_i}$ is the probability of observing a sample from the i-th class in the subset, k is the number of classes and X is the set of all instances to be discriminated at each node. During training the algorithm test different binary splits of the instances of the node for all the attributes and possible

values and calculate the weighted average of the Gini index in order to obtain the partition that minimizes this value. For the categorical variables, the algorithm considers all binary partitions for all the values of each feature and for the numerical values, the algorithm uses the midpoint between each pair of sorted adjacent values to create binary partitions of cases with a value lower or equal to the midpoint and cases with higher value.

3 Methodology

In this section, we will explain the implementation of the Random Forest and Decision Forest algorithms in Python 3 and the experimental approach to analyze the performance of the algorithms and the ability to generate accurate classifications. We also analyze the impact on the performance of different values of the parameters.

3.1 Implementation

Using the theoretical background given in the course lectures and the original papers of Random Forest [1] and Decision Forest [2] we implemented both algorithms in Python 3. All the code is available in the `source` folder and the instructions to execute the code will be given at the end of the section. The implementation of the algorithms is in the `ensemble_learning.py` and the script used to execute the algorithm for the experimental phase is `experiments.py`. Also, there is a `main.py` script with the code to execute both Random Forest and Decision Forest for all the data sets with the optimal parameters found in the experimental phase. We created three Python classes called `RandomForestClassifier`, `DecisionForestClassifier` and `DecisionTreeClassifier` where the use and parameters are:

- `DecisionTreeClassifier`(): creates and fits one decision tree.
  - `cat_variables`: list with the name of the categorical variables of the data set.
  - `max_depth`: maximum depth of the resulting trees.
  - `min_samples_split`: minimum number of samples to create a new split (reduce overfitting).
- `RandomForestClassifier`() and `DecisionForestClassifier`(): define and fit the classifier using the parameters of the class.
  - `dataset_name`: name of the data set to use to train and evaluate the algorithm. This value has to be one valid name of a CSV file in the `data` folder.
  - `target_name`: name of the variable to be predicted.
  - `cat_variables`: list with the name of the categorical variables of the data set.
  - `test_split`: the value of the percentage of instances of the data set separated for the evaluation phase, the test data. The rest of the instances are used for training.
  - `n_estimators`: number of trees.
  - `number_features`: number of features to be selected.
  - `max_depth`: maximum depth of the resulting trees.
  - `min_samples_split`: minimum number of samples to create a new split (reduce overfitting).

There are different important parts in the Random Forest and Decision Forest class of the algorithm that we are going to explain in detail in the following paragraphs. First, we integrated into the class the import and preprocessing of the data. We import the data set using the `dataset_name` parameter where the file has to be in CSV format. For the preprocessing, first, we check if there are missing data and, in the case of missing values, we imputed them with mean or mode depending if the variable is numerical or categorical. Then, for the categorical variables of the data set, we encoded the values to avoid characters and have integers that make it easier to calculate distances. Finally, we performed the train and test split of the data for the training and evaluation using the test size of the parameter

`test_split`.

One important aspect of the implementation of both algorithms is how to store the tree structure. In this case, we decided to use a dictionary for each tree that has as keys the feature name, the value of the condition, the left subtree and the right subtree. A subtree is also a dictionary with the same structure. For example, a small tree will be represented as { `feature: sugar, value: 0.5, left: 0, right: { feature: cholesterol, value: 190.5, left: 0, right: 1 } }` where the root node is the feature sugar and cases with sugar <= 0.5 are classified as 0 and the cases where sugar > 0.5 go to the child node where the feature is cholesterol. The cases where cholesterol <= 190.5 are classified as 0 and cholesterol > 190.5 as 1.

The next step is the implementation of the training phase with the `fit`() method. Here we can find the differences between the Random Forest and the Decision Forest. For both methods, we iterate over the number of trees (`n_estimators` parameter) and, for the Random Forest, we create the bootstrapped sample and randomly select the number of features based on the `number_features` parameter. In Algorithm 2 we can see the steps of the training of the Random Forest. Then, with this data, we use the `DecisionTreeClassifier` class to create and fit the decision tree using the CART method, where Algorithm 1 describes the procedure used to train each decision tree. For the Decision Forest, we use all the original data and only randomly select the features, so the algorithm is the same as 2 but removing the second line.
Also, we created the `predict`() and `evaluate`() methods to produce predictions using the classifier and evaluate the performance. For the predictions, the method iterates over all the trees of the ensemble and, for each one, traverse all the tree until arriving at a leaf node. After obtaining a prediction for all the trees, the method returns the most frequent class as a final prediction. Then, the evaluation method returns the train and test accuracy, the confusion matrix of the test and the classification report also for the test.

Another method that we implemented in `RandomForestClassifier`() and `DecisionForestClassifier`() classes is the `feature_importance`() that returns a python dictionary for each model where the features used are presented alongside the number of times they appear in a decision split. With this procedure, we obtain the feature importance for each model and if we order this list we obtained the most important and less important features of the data set in order to predict the instances.

The following code is an example of how to execute the Random Forest algorithm with this implementation for the Heart Disease data set:

```
cat_variables_heart = [sex, chest, sugar, ecg, angina, slope,
thal, disease]

model = RandomForestClassifier(dataset_name=heart,
        target_name=disease,
        cat_variables=cat_variables_heart,
        test_split=0.2,
        n_estimators=40)
X_train, y_train, X_val, y_val = model.import_data()
model.fit(X_train, y_train)
pred = model.predict(X_val)
acc = model.evaluate(X_val, y_val)
```

Algorithm 1 CART algorithm.

procedure CART($X, y, depth$)
    $num\_samples, num\_features \leftarrow X.shape$
    $num\_labels \leftarrow len(np.unique(y))$
    if $len(np.unique(y)) = 1$ or $depth = max\_depth$ or $num\_samples < min\_samples\_split$ **then**
        return majority class label in y
    end if
    $best\_feature \leftarrow 0$
    $best\_value \leftarrow 0$
    $best\_score \leftarrow -1$
    for $feature$ **in** $num\_features$ **do**
        if $feature$ is categorical **then**
            for $value$ **in** unique values of $X[feature]$ **do**
                $score \leftarrow$ Gini index of split based on $value$
                if $score > best\_score$ **then**
                    $best\_feature \leftarrow feature$
                    $best\_value \leftarrow value$
                    $best\_score \leftarrow score$
                end if
            end for
        else
            sort values of $X[feature]$
            select representative values
            for $value$ **in** representative values **do**
                $score \leftarrow$ Gini index of split based on $value$
                if $score > best\_score$ **then**
                    $best\_feature \leftarrow feature$
                    $best\_value \leftarrow value$
                    $best\_score \leftarrow score$
                end if
            end for
        end if
    end for
    if $best\_score = -1$ **then**
        return majority class label in y
    end if
    partition data based on $best\_feature$ and $best\_value$
    build left and right subtrees using the partitioned data
    return feature: $X.columns[best\_feature]$, value: $best\_value$, left: $CART(left\_data)$,
    right: $CART(right\_data)$
end procedure

Algorithm 2 Random Forest

for $i \leftarrow 1$ to $n\_estimators$ **do**
    Sample rows with replacement: $indices\_row \leftarrow$ random choice of $X.shape[0]$ elements
    Sample columns without replacement: $indices\_col \leftarrow$ random choice of nf elements from $X.shape[1]$
    Select bootstrap sample of data: $X\_bootstrap \leftarrow$ rows of X indexed by $indices\_row$ and columns indexed by $indices\_col$
    Select bootstrap sample of target variable: $y\_bootstrap \leftarrow$ rows of y indexed by $indices\_row$
    Train a decision tree on the bootstrap sample: $tree \leftarrow$ DecisionTreeClassifier with parameters $attr\_names$, $cat\_variables$, max_depth, $min\_samples\_split$, and $min\_samples\_leaf$
    Fit decision tree to bootstrap sample: $tree.fit(X\_bootstrap, y\_bootstrap)$
    Add decision tree to the ensemble: self.trees.append($tree$)
end for

3.2 Evaluation

For the evaluation of the implementation of the algorithms, we first try to test them in three data sets from the UCI Machine Learning Repository of different sizes and with numerical and categorical attributes. We included some data sets with missing data to test the preprocessing part of the algorithm. Also, for each data set, we tested the models with different parameter configurations in order to analyze the effect of them. At the end of the section, we will explain the parameter configuration of the experiments conducted in this work. The data sets from the UCI Machine Learning Repository are:

To measure the performance and ability to classify new instances of the models we used train/test cross-validation where the training split was used to train the model and the test split to evaluate the performance in a new set of instances. In this practical work, we used the train and test accuracy and the confusion matrix to analyze the performance of the models in the different data sets and parameter configurations.

The parameter configurations for the experiments are based on the `n_estimators` and `number_features` parameters. For both algorithms, the number of trees will have the values of 1, 10, 25, 50, 75 and 100. Then, the number of features will depend on the algorithm and the number of features of the data set. Being M the number of features of the data set, for Random Forest we will have values of 1, 2, int($\log_2 M + 1$), int($\sqrt{M}$) and M. For Decision Forest, the number of features will be int($M/4$), int($M/2$), int($3 * M/4$), Runif(1,M) and M. We evaluated the algorithms using all the combinations of the parameters, but for the large data set, we could not test the models for all the features (M) because of the high training time.

4 Results

In this section, we will present the results obtained in the evaluation of the Random Forest and Decision Forest algorithms in the three data sets and the procedure explained in the previous section. All the results are stored in the `results` folder where, for each data set, we generate a CSV file with the accuracies and execution time of each combination of parameters, two txt files with the results and the final trees for each combination, two general plots that summarize the accuracy and execution time for each combination of parameters and, for each combination, a confusion matrix and the classification report with detailed results.

4.1 Heart Disease

In Tables 1 and 2 we can see the results of the Heart Disease data set for the Random Forest and Decision Forest, respectively. For the first algorithm, we can see a clear tendency for higher test accuracy as the number of trees increases and also less variation as the number of features changes. The combination of parameters that obtain a higher test accuracy is `n_estimators` = 75 and `number_features` = 2, with a training accuracy of 0.8935 and a test accuracy of 0.8333. There is another combination of parameters that obtains the same test accuracy but presents more overfitting (NT = 50, F = 5) and one case with higher accuracy (NT = 25, F = 1) but it uses only one feature, making the model less stable and very dependent of the feature selected. In Figure 1a we can visualize the test accuracy for all the combinations of parameters which demonstrates the increasing tendency as the number of trees also increase. Also, in Figure 1b we can see the confusion matrix of the selected model. For the feature importance, we can see some variation in the ordered list but some features are clearly in the first position of importance. For this case, the most important features are cholesterol and max hr and the less important are sugar and sex.

For the Decision Forest, the first thing we can see is a higher overfitting for mostly all the combinations of parameters compared to the Random Forest. The selected combination of parameters that produces the better test accuracy with some overfitting is `n_estimators` = 25 and `number_features` = 6, with a training accuracy of 0.9953 and a test accuracy of 0.8333. In Figure 2a we can see the test accuracies for the combinations of parameters and in Figure 2b the confusion matrix of the selected model. For this model, the features that are more and less important are the same as the previous one.

NT	F	Train	Test	Feature Importance
1	1	0.5416	0.6111	sugar
	2	0.6898	0.7037	cholesterol, max hr
	4	0.7824	0.5740	oldpeack, rbp, slope, sugar
	5	0.8194	0.4444	rbp, cholesterol, ecg, slope, sugar
	13	0.7638	0.6296	chest, age, rbp, ecg, cholesterol, angina, oldpeack, max hr, vessels, sugar, slope, thal, sex
10	1	0.7731	0.6851	rbp, oldpeack, slope, chest, thal, ecg, sugar
	2	0.7962	0.7592	rbp, age, max hr, oldpeack, sex, slope, angina, chest, ecg, vessels, sugar
	4	0.7453	0.6667	cholesterol, max hr, rbp, age, slope, sex, chest, vessels, angina, ecg, oldpeack, sugar, thal
	5	0.8611	0.6851	max hr, cholesterol, oldpeack, age, vessels, slope, rbp, angina, ecg, sex, chest, sugar
	13	0.7824	0.7407	max hr, oldpeack, chest, cholesterol, age, sex, rbp, angina, slope, ecg, vessels, thal, sugar
25	1	0.8703	0.8703	cholesterol, rbp, max hr, oldpeack, age, vessels, chest, slope, angina, thal, sex
	2	0.7916	0.7407	cholesterol, max hr, oldpeack, age, slope, rbp, vessels, ecg, chest, angina, thal, sugar, sex
	4	0.8842	0.7037	cholesterol, rbp, oldpeack, max hr, age, chest, vessels, angina, thal, slope, sex, ecg, sugar
	5	0.8981	0.6296	cholesterol, oldpeack, age, rbp, chest, max hr, vessels, slope, thal, ecg, sex, sugar
	13	0.8704	0.7222	cholesterol, age, oldpeack, max hr, rbp, vessels, slope, chest, angina, thal, ecg, sugar, sex
50	1	0.8565	0.7592	cholesterol, oldpeack, age, max hr, rbp, vessels, slope, chest, angina, ecg, sex, sugar, thal

	2	0.8888	0.7592	max hr, cholesterol, age, oldpeack, rbp, chest, vessels, ecg, thal, sugar, slope, sex, angina
	4	0.8935	0.7407	cholesterol, age, max hr, oldpeack, rbp, vessels, slope, ecg, chest, angina, thal, sex, sugar
	5	0.9351	0.8333	max hr, cholesterol, rbp, age, oldpeack, vessels, thal, chest, sex, slope, angina, ecg, sugar
	13	0.8611	0.6852	max hr, cholesterol, rbp, age, oldpeack, vessels, thal, chest, sex, slope, angina, ecg, sugar
	1	0.8518	0.7592	max hr, cholesterol, rbp, age, oldpeack, chest, vessels, slope, ecg, thal, sugar, angina, sex
75	**2**	**0.8935**	**0.8333**	cholesterol, oldpeack, age, rbp, max hr, slope, chest, vessels, angina, thal, ecg, sex, sugar
	4	0.9352	0.7407	cholesterol, max hr, oldpeack, age, rbp, vessels, thal, chest, slope, angina, ecg, sex, sugar
	5	0.9352	0.7592	cholesterol, max hr, age, rbp, oldpeack, vessels, chest, thal, slope, angina, sex, ecg, sugar
	13	0.8703	0.7778	max hr, oldpeack, rbp, age, cholesterol, vessels, chest, slope, ecg, thal, angina, sex, sugar
	1	0.8565	0.7963	cholesterol, age, rbp, max hr, oldpeack, chest, thal, slope, vessels, ecg, sex, sugar, angina
100	2	0.8935	0.7592	oldpeack, cholesterol, max hr, age, rbp, chest, vessels, angina, thal, ecg, slope, sex, sugar
	4	0.9120	0.7592	cholesterol, age, max hr, oldpeack, rbp, slope, vessels, chest, angina, sex, thal, ecg, sugar
	5	0.9583	0.7963	max hr, age, cholesterol, rbp, oldpeack, vessels, chest, slope, thal, sex, angina, ecg, sugar
	13	0.9074	0.7592	age, cholesterol, oldpeack, rbp, max hr, vessels, slope, chest, thal, angina, ecg, sex, sugar

Table 1: Results of the Random Forest classifier for the different combinations of parameters in the Heart data set with the train accuracy, test accuracy and features ordered by importance.

NT	**F**	**Train**	**Test**	**Feature Importance**
	3	0.6527	0.6667	thal, chest, ecg
	6	0.9815	0.5555	rbp, cholesterol, oldpeack, chest, ecg, angina
1	10	0.8379	0.6296	max hr, cholesterol, rbp, age, chest, vessels, ecg, oldpeack, sex
	13	0.8101	0.6111	max hr, ecg, angina, cholesterol, age, vessels, oldpeack, rbp, slope, sugar, chest, sex, thal
	runif	0.9537	0.5740	oldpeack, cholesterol, rbp, thal
	3	0.9676	0.5926	oldpeack, cholesterol, age, rbp, max hr, chest, thal, vessels, ecg, angina, sex, slope
10	6	1.0	0.6852	age, cholesterol, rbp, max hr, oldpeack, vessels, angina, chest, slope, sex, sugar, thal, ecg
	10	0.9722	0.7222	age, max hr, cholesterol, oldpeack, rbp, thal, chest, vessels, sex, ecg, slope, sugar, angina
	13	0.9722	0.6667	rbp, cholesterol, max hr, age, oldpeack, chest, slope, thal, ecg, angina, sugar, vessels, sex
	runif	0.9167	0.6296	cholesterol, age, oldpeack, rbp, vessels, angina, thal, max hr, sex, slope, chest, ecg, sugar

	3	0.9815	0.7037	cholesterol, age, max hr, rbp, oldpeack, slope, vessels, angina, sex, chest, ecg, thal, sugar
25	**6**	**0.9953**	**0.8333**	age, max hr, rbp, cholesterol, vessels, oldpeack, slope, chest, sex, ecg, angina, thal, sugar
	10	0.9768	0.6296	cholesterol, rbp, age, max hr, oldpeack, slope, vessels, thal, ecg, chest, angina, sex, sugar
	13	0.9444	0.6667	cholesterol, rbp, age, max hr, oldpeack, thal, chest, ecg, vessels, angina, sex, slope, sugar
	runif	0.9953	0.6852	age, cholesterol, max hr, rbp, chest, oldpeack, slope, thal, ecg, sex, vessels, angina, sugar
	3	0.9862	0.7037	cholesterol, max hr, age, rbp, oldpeack, chest, angina, thal, vessels, sex, slope, ecg, sugar
50	6	0.9815	0.7778	max hr, cholesterol, age, oldpeack, rbp, chest, thal, vessels, sex, slope, ecg, angina, sugar
	10	0.9954	0.7037	age, oldpeack, cholesterol, max hr, rbp, chest, vessels, thal, slope, angina, ecg, sex, sugar
	13	0.9861	0.7222	age, cholesterol, max hr, oldpeack, rbp, slope, ecg, chest, thal, vessels, sex, sugar, angina
	runif	1.0	0.7037	rbp, max hr, age, cholesterol, oldpeack, chest, sex, vessels, thal, slope, angina, ecg, sugar
	3	0.9861	0.7407	cholesterol, max hr, oldpeack, age, rbp, vessels, chest, slope, thal, angina, sex, ecg, sugar
75	6	0.9861	0.6481	max hr, age, cholesterol, oldpeack, rbp, chest, vessels, slope, thal, angina, sex, ecg, sugar
	10	0.9907	0.7407	cholesterol, max hr, age, rbp, oldpeack, vessels, chest, slope, thal, angina, ecg, sex, sugar
	13	0.9907	0.6852	age, cholesterol, max hr, rbp, oldpeack, chest, thal, vessels, slope, sex, angina, ecg, sugar
	runif	0.9907	0.6852	cholesterol, age, max hr, oldpeack, rbp, chest, vessels, thal, slope, ecg, angina, sex, sugar
	3	1.0	0.7222	cholesterol, max hr, age, oldpeack, rbp, chest, vessels, thal, slope, sex, angina, ecg, sugar
100	6	0.9953	0.7407	cholesterol, max hr, oldpeack, rbp, age, chest, vessels, slope, thal, angina, sex, ecg, sugar
	10	0.9954	0.7222	age, cholesterol, max hr, rbp, oldpeack, vessels, chest, thal, slope, sex, angina, ecg, sugar
	13	0.9861	0.7407	cholesterol, age, max hr, rbp, oldpeack, vessels, thal, chest, slope, sex, ecg, angina, sugar
	runif	1.0	0.6852	cholesterol, max hr, rbp, age, oldpeack, chest, slope, thal, sex, vessels, angina, ecg, sugar

Table 2: Results of the Decision Forest classifier for the different combinations of parameters in the Heart data set with the train accuracy, test accuracy and features ordered by importance.

4.2 Breast Cancer

For each combination of the parameters, we can see the train accuracy, test accuracy and the list of features used ordered by importance in Table 3 and Table 4 for the Random Forest and Decision Forest models, respectively. Both models produce high accuracy for this data set in both training and test set. For the Random Forest, the selected model is the one that maximizes the test accuracy which in this case is the combination of `n_estimators` = 75 and `number_features` = 4, which produces a training accuracy of 0.9678 and a test accuracy of 0.9857. In Figure 3a we can visualize the results of

the test accuracy for the different combinations of parameters showing similar performance for all the combinations and in Figure 3b we can see the confusion matrix of the selected model. If we analyze the importance of the features we can see a lot of variation but the feature that appears as the most important more times is Bland Chromatin which indicates that it is the one that better classifies the data. Then, the Mitoses feature is clearly the variable that appears more frequently at the end of the ordered list showing that this feature is not very important for the classification of the instances.

For the Decision Forest, the results are very similar and the selected model is the one where the parameters are `n_estimators` = 25 and `number_features` = runif. The visualization of the results, that are very similar to the Random Forest, can be seen in Figure 4a and the confusion matrix of the selected model in 4b. Also, the analysis of the importance of the features is more or less the same and here we can see more clearly that the most important feature is the Bland Chromatin and the less important the Mitoses.

NT	F	Train	Test	Feature Importance
1	1	0.9266	0.9285	Uniformity of Cell Size
	2	0.9284	0.9357	Uniformity of Cell Size, Clump Thickness
	3	0.9374	0.9214	Normal Nucleoli, Marginal Adhesion, Clump Thickness
	4	0.3256	0.3215	Bare Nuclei, Uniformity of Cell Size, Mitoses, Marginal Adhesion
	9	0.3899	0.3214	Clump Thickness, Uniformity of Cell Size, Single Epithelial Cell Size, Bland Chromatin, Marginal Adhesion, Bare Nuclei, Mitoses, Uniformity of Cell Shape, Normal Nucleoli
10	1	0.9624	0.9714	Clump Thickness, Marginal Adhesion, Bland Chromatin, Single Epithelial Cell Size, Bare Nuclei, Normal Nucleoli, Uniformity of Cell Shape
	2	0.9463	0.9643	Normal Nucleoli, Mitoses, Clump Thickness, Marginal Adhesion, Single Epithelial Cell Size, Bland Chromatin, Uniformity of Cell Size, Uniformity of Cell Shape
	3	0.9392	0.8928	Marginal Adhesion, Uniformity of Cell Size, Mitoses, Uniformity of Cell Shape, Single Epithelial Cell Size, Clump Thickness, Normal Nucleoli, Bare Nuclei, Bland Chromatin
	4	0.9535	0.9357	Single Epithelial Cell Size, Uniformity of Cell Shape, Normal Nucleoli, Bland Chromatin, Marginal Adhesion, Clump Thickness, Mitoses, Bare Nuclei, Uniformity of Cell Size
	9	0.9517	0.9357	Bland Chromatin, Clump Thickness, Bare Nuclei, Uniformity of Cell Size, Normal Nucleoli, Marginal Adhesion, Single Epithelial Cell Size, Uniformity of Cell Shape, Mitoses
25	1	0.8604	0.8357	Single Epithelial Cell Size, Mitoses, Uniformity of Cell Size, Marginal Adhesion, Normal Nucleoli, Clump Thickness, Bland Chromatin, Uniformity of Cell Shape, Bare Nuclei
	2	0.9678	0.9571	Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Clump Thickness, Normal Nucleoli, Uniformity of Cell Shape, Mitoses, Uniformity of Cell Size
	3	0.9678	0.9642	Single Epithelial Cell Size, Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Normal Nucleoli, Bland Chromatin, Bare Nuclei, Mitoses, Uniformity of Cell Size
	4	0.9660	0.9571	Bland Chromatin, Uniformity of Cell Shape, Single Epithelial Cell Size, Uniformity of Cell Size, Clump Thickness, Normal Nucleoli, Bare Nuclei, Mitoses, Marginal Adhesion
	9	0.9731	0.9664	Bland Chromatin, Clump Thickness, Marginal Adhesion, Single Epithelial Cell Size, Uniformity of Cell Size, Uniformity of Cell Shape, Normal Nucleoli, Bare Nuclei, Mitoses

50	1	0.9642	0.9714	Marginal Adhesion, Uniformity of Cell Size, Clump Thickness, Uniformity of Cell Shape, Single Epithelial Cell Size, Bland Chromatin, Bare Nuclei, Mitoses, Normal Nucleoli
	2	0.9660	0.9714	Clump Thickness, Bland Chromatin, Single Epithelial Cell Size, Uniformity of Cell Size, Normal Nucleoli, Marginal Adhesion, Mitoses, Uniformity of Cell Shape, Bare Nuclei
	3	0.9732	0.9714	Normal Nucleoli, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Bare Nuclei, Bland Chromatin, Marginal Adhesion, Mitoses, Single Epithelial Cell Size
	4	0.9660	0.9714	Uniformity of Cell Shape, Bland Chromatin, Normal Nucleoli, Single Epithelial Cell Size, Uniformity of Cell Size, Marginal Adhesion, Clump Thickness, Bare Nuclei, Mitoses
	9	0.9767	0.9643	Bland Chromatin, Clump Thickness, Uniformity of Cell Shape, Single Epithelial Cell Size, Marginal Adhesion, Uniformity of Cell Size, Normal Nucleoli, Bare Nuclei, Mitoses
75	1	0.9427	0.9428	Uniformity of Cell Shape, Single Epithelial Cell Size, Mitoses, Marginal Adhesion, Bland Chromatin, Normal Nucleoli, Clump Thickness, Uniformity of Cell Size, Bare Nuclei
	2	0.9552	0.9643	Normal Nucleoli, Bland Chromatin, Single Epithelial Cell Size, Clump Thickness, Uniformity of Cell Size, Bare Nuclei, Uniformity of Cell Shape, Marginal Adhesion, Mitoses
	3	0.9570	0.9643	Uniformity of Cell Shape, Normal Nucleoli, Bland Chromatin, Uniformity of Cell Size, Marginal Adhesion, Clump Thickness, Single Epithelial Cell Size, Mitoses, Bare Nuclei
	4	**0.9678**	**0.9857**	Bland Chromatin, Single Epithelial Cell Size, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Normal Nucleoli, Bare Nuclei, Marginal Adhesion, Mitoses
	9	0.9785	0.9714	Bland Chromatin, Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Normal Nucleoli, Uniformity of Cell Size, Single Epithelial Cell Size, Bare Nuclei, Mitoses
100	1	0.9660	0.9786	Bland Chromatin, Uniformity of Cell Shape, Marginal Adhesion, Normal Nucleoli, Bare Nuclei, Mitoses, Clump Thickness, Single Epithelial Cell Size, Uniformity of Cell Size
	2	0.9606	0.9643	Bland Chromatin, Clump Thickness, Bare Nuclei, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Normal Nucleoli, Mitoses, Single Epithelial Cell Size
	3	0.9606	0.9500	Normal Nucleoli, Single Epithelial Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Uniformity of Cell Size, Bland Chromatin, Clump Thickness, Bare Nuclei, Mitoses
	4	0.9749	0.9643	Bland Chromatin, Single Epithelial Cell Size, Normal Nucleoli, Uniformity of Cell Shape, Uniformity of Cell Size, Clump Thickness, Marginal Adhesion, Bare Nuclei, Mitoses
	9	0.9767	0.9786	Bland Chromatin, Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Uniformity of Cell Size, Single Epithelial Cell Size, Normal Nucleoli, Bare Nuclei, Mitoses

Table 3: Results of the Random Forest classifier for the different combinations of parameters in the Breast Cancer data set with the train accuracy, test accuracy and features ordered by importance.

NT	**F**	**Train**	**Test**	**Feature Importance**
1	2	0.9392	0.9143	Single Epithelial Cell Size, Uniformity of Cell Shape
	4	0.3810	0.3357	Single Epithelial Cell Size, Uniformity of Cell Shape, Bland Chromatin, Mitoses

	7	0.9374	0.9428	Clump Thickness, Marginal Adhesion, Bland Chromatin, Uniformity of Cell Size, Single Epithelial Cell Size, Mitoses, Bare Nuclei
	9	0.8318	0.8643	Bland Chromatin, Mitoses, Single Epithelial Cell Size, Clump Thickness, Bare Nuclei, Normal Nucleoli, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion
	runif	0.9141	0.9286	Normal Nucleoli, Bland Chromatin, Single Epithelial Cell Size, Bare Nuclei, Marginal Adhesion, Uniformity of Cell Shape, Uniformity of Cell Size
10	2	0.9553	0.9643	Bland Chromatin, Normal Nucleoli, Single Epithelial Cell Size, Marginal Adhesion, Uniformity of Cell Size, Mitoses, Clump Thickness, Bare Nuclei
	4	0.9714	0.9643	Bland Chromatin, Uniformity of Cell Size, Normal Nucleoli, Bare Nuclei, Clump Thickness, Single Epithelial Cell Size, Marginal Adhesion, Uniformity of Cell Shape, Mitoses
	7	0.9105	0.9074	Bland Chromatin, Uniformity of Cell Shape, Single Epithelial Cell Size, Normal Nucleoli, Clump Thickness, Marginal Adhesion, Uniformity of Cell Size, Mitoses, Bare Nuclei
	9	0.8336	0.8714	Bland Chromatin, Clump Thickness, Single Epithelial Cell Size, Marginal Adhesion, Mitoses, Uniformity of Cell Size, Bare Nuclei, Normal Nucleoli, Uniformity of Cell Shape
	runif	0.9588	0.9286	Marginal Adhesion, Clump Thickness, Bland Chromatin, Normal Nucleoli, Uniformity of Cell Size, Uniformity of Cell Shape, Mitoses, Single Epithelial Cell Size, Bare Nuclei
25	2	0.9606	0.9714	Bland Chromatin, Uniformity of Cell Shape, Single Epithelial Cell Size, Uniformity of Cell Size, Clump Thickness, Normal Nucleoli, Marginal Adhesion, Mitoses, Bare Nuclei
	4	0.9696	0.9571	Bland Chromatin, Uniformity of Cell Size, Normal Nucleoli, Single Epithelial Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Clump Thickness, Bare Nuclei, Mitoses
	7	0.9713	0.9714	Bland Chromatin, Normal Nucleoli, Single Epithelial Cell Size, Marginal Adhesion, Uniformity of Cell Shape, Bare Nuclei, Clump Thickness, Uniformity of Cell Size, Mitoses
	9	0.8372	0.8571	Bland Chromatin, Marginal Adhesion, Single Epithelial Cell Size, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Normal Nucleoli, Mitoses, Bare Nuclei
	runif	**0.9767**	**0.9786**	Bland Chromatin, Uniformity of Cell Shape, Marginal Adhesion, Clump Thickness, Single Epithelial Cell Size, Mitoses, Uniformity of Cell Size, Normal Nucleoli, Bare Nuclei
50	2	0.9714	0.9714	Single Epithelial Cell Size, Bland Chromatin, Uniformity of Cell Shape, Normal Nucleoli, Uniformity of Cell Size, Bare Nuclei, Mitoses, Clump Thickness, Marginal Adhesion
	4	0.9767	0.9714	Bland Chromatin, Normal Nucleoli, Single Epithelial Cell Size, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei, Mitoses
	7	0.9767	0.9714	Bland Chromatin, Uniformity of Cell Shape, Clump Thickness, Single Epithelial Cell Size, Marginal Adhesion, Uniformity of Cell Size, Normal Nucleoli, Bare Nuclei, Mitoses
	9	0.8533	0.8785	Single Epithelial Cell Size, Clump Thickness, Bland Chromatin, Marginal Adhesion, Normal Nucleoli, Uniformity of Cell Size, Uniformity of Cell Shape, Bare Nuclei, Mitoses
	runif	0.9785	0.9643	Bland Chromatin, Marginal Adhesion, Single Epithelial Cell Size, Uniformity of Cell Size, Clump Thickness, Normal Nucleoli, Uniformity of Cell Shape, Bare Nuclei, Mitoses

75	2	0.9678	0.9786	Bland Chromatin, Marginal Adhesion, Uniformity of Cell Size, Single Epithelial Cell Size, Clump Thickness, Normal Nucleoli, Uniformity of Cell Shape, Bare Nuclei, Mitoses
	4	0.9803	0.9714	Bland Chromatin, Uniformity of Cell Size, Uniformity of Cell Shape, Normal Nucleoli, Single Epithelial Cell Size, Marginal Adhesion, Clump Thickness, Bare Nuclei, Mitoses
	7	0.9678	0.9714	Bland Chromatin, Uniformity of Cell Shape, Single Epithelial Cell Size, Clump Thickness, Uniformity of Cell Size, Marginal Adhesion, Normal Nucleoli, Mitoses, Bare Nuclei
	9	0.8604	0.8714	Bland Chromatin, Clump Thickness, Single Epithelial Cell Size, Marginal Adhesion, Uniformity of Cell Size, Uniformity of Cell Shape, Normal Nucleoli, Bare Nuclei, Mitoses
	runif	0.9767	0.9714	Bland Chromatin, Uniformity of Cell Size, Uniformity of Cell Shape, Single Epithelial Cell Size, Normal Nucleoli, Clump Thickness, Marginal Adhesion, Bare Nuclei, Mitoses
100	2	0.9660	0.9715	Uniformity of Cell Shape, Normal Nucleoli, Bland Chromatin, Single Epithelial Cell Size, Marginal Adhesion, Uniformity of Cell Size, Bare Nuclei, Clump Thickness, Mitoses
	4	0.9803	0.9714	Single Epithelial Cell Size, Uniformity of Cell Shape, Bland Chromatin, Normal Nucleoli, Marginal Adhesion, Clump Thickness, Uniformity of Cell Size, Bare Nuclei, Mitoses
	7	0.9713	0.9714	Bland Chromatin, Uniformity of Cell Size, Clump Thickness, Single Epithelial Cell Size, Marginal Adhesion, Normal Nucleoli, Uniformity of Cell Shape, Bare Nuclei, Mitoses
	9	0.8604	0.8928	Bland Chromatin, Clump Thickness, Single Epithelial Cell Size, Uniformity of Cell Shape, Uniformity of Cell Size, Marginal Adhesion, Normal Nucleoli, Bare Nuclei, Mitoses
	runif	0.9767	0.9643	Bland Chromatin, Single Epithelial Cell Size, Clump Thickness, Uniformity of Cell Size, Normal Nucleoli, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei, Mitoses

Table 4: Results of the Decision Forest classifier for the different combinations of parameters in the Breast Cancer data set with the train accuracy, test accuracy and features ordered by importance.

4.3 Obesity data set

For this data set, the execution was much higher than the previous ones because it has more instances and features so, we could not test the performance of the models using all the features and, for the Decision Tree, we were not able to evaluate the model with 100 trees. Also, we had to do the experiments in two stages which makes impossible to generate a visualization of the results as in the previous ones. The results can only be presented in Tables 6 and 5 for Random Forest and Decision Forest, respectively.

For the Random Forest, in the case of 100 trees, we can see two combinations that produce a decent test accuracy but present high overfitting so, as the selected model we decided to choose the case with `n_estimators` = 10 and `number_features` = 4, which has as train accuracy 0.6848 and for test 0.6004. In Figure 5a we can see the confusion matrix for the selected model. If we analyze the feature importance for the Random Forest model we can clearly see that the more important variable to predict the type of obesity is the Height, Weight and family history of overweight. On the other hand, the less important variable is more or less always the Smoke variable.

For the Decision Forest the test accuracies are lower compared to Random Forest and the selected model is the one with `n_estimators` = 50 and `number_features` = 4, with a train accuracy of 0.8453 and a test accuracy of 0.5626. In this model, we tested the use of more features than the previous

model and we can see that there is no improvement in test accuracy and the execution time increased a lot. In the analysis of the feature importance, we obtained the same conclusion as before, where the important variables are Height, Weight and family history of overweight, and the less important feature is Smoke.

NT	F	Train	Test	Feature Importance
1	1	0.2280	0.2482	family_history_with_overweight
	2	0.2292	0.2198	FAVC, CH2O
	4	0.4786	0.4704	Weight, CAEC, TUE, SCC
	5	0.1558	0.1158	MTRANS, NCP, FAVC, TUE, SMOKE
10	1	0.5728	0.3664	Height, Age, CH2O, CAEC, NCP, FAF, FCVC, TUE, SMOKE
	2	0.3708	0.3073	Height, MTRANS, CALC, NCP, FCVC, FAF, CAEC, TUE, Gender, SCC
	4	**0.6848**	**0.6004**	Age, Height, Weight, Gender, MTRANS, CAEC, FCVC, family_history_with_overweight, FAVC, CALC, CH2O, SCC, NCP, TUE, FAF, SMOKE
	5	0.5894	0.4302	Weight, Age, Height, FCVC, FAF, Gender, family_history_with_overweight, TUE, NCP, CAEC, FAVC, CALC, CH2O, MTRANS, SCC, SMOKE
25	1	0.6718	0.4491	Height, Weight, Age, CAEC, NCP, FAF, TUE, FCVC, SCC, CH2O, family_history_with_overweight, FAVC
	2	0.6949	0.4137	Age, Height, Weight, NCP, TUE, FCVC, CH2O, MTRANS, FAVC, family_history_with_overweight, CALC, SCC, CAEC, SMOKE, Gender
	4	0.6771	0.4515	Weight, Age, Height, NCP, FAF, TUE, FCVC, Gender, family_history_with_overweight, CAEC, MTRANS, FAVC, CALC, CH2O, SCC, SMOKE
	5	0.5373	0.3900	Age, Height, Weight, FCVC, FAF, MTRANS, Gender, NCP, CAEC, family_history_with_overweight, CALC, TUE, FAVC, SCC, CH2O, SMOKE
50	1	0.4828	0.3688	Weight, Height, Age, CAEC, CH2O, MTRANS, Gender, CALC, TUE, SCC, FAVC, FCVC, FAF, NCP, family_history_with_overweight
	2	0.5284	0.4822	Weight, Age, Height, FAF, NCP, FAVC, FCVC, MTRANS, TUE, family_history_with_overweight, CALC, CAEC, SCC, CH2O, Gender, SMOKE
	4	0.8525	0.5413	Height, Age, Weight, MTRANS, FCVC, TUE, FAF, family_history_with_overweight, CAEC, NCP, CH2O, Gender, CALC, FAVC, SCC, SMOKE
	5	0.8227	0.5343	Height, Weight, Age, NCP, FCVC, CAEC, Gender, TUE, CH2O, family_history_with_overweight, MTRANS, FAF, CALC, FAVC, SCC, SMOKE
75	1	0.5497	0.3735	Height, Weight, Age, CAEC, FCVC, NCP, FAF, Gender, CALC, MTRANS, CH2O, family_history_with_overweight, FAVC, TUE, SCC, SMOKE
	2	0.7997	0.5957	Weight, Height, Age, NCP, CALC, Gender, TUE, CAEC, FCVC, MTRANS, family_history_with_overweight, FAVC, SCC, CH2O, FAF, SMOKE
	4	0.9069	0.6122	Weight, Age, Height, TUE, FAF, NCP, Gender, CALC, FCVC, family_history_with_overweight, FAVC, MTRANS, CAEC, CH2O, SCC, SMOKE
	5	0.8601	0.5886	Height, Weight, Age, CALC, Gender, NCP, FAF, MTRANS, family_history_with_overweight, FCVC, CAEC, TUE, FAVC, CH2O, SCC, SMOKE
100	1	0.6155	0.4491	Height, Weight, Age, FAF, CAEC, MTRANS, NCP, FCVC, CH2O, Gender, TUE, SCC, family_history_with_overweight, FAVC, SMOKE, CALC
	2	0.9069	0.6808	Weight, Age, Height, CAEC, NCP, TUE, FCVC, MTRANS, CALC, CH2O, FAVC, SCC, Gender, family_history_with_overweight, FAF, SMOKE

4	0.8791	0.6335	Weight, Age, Height, Gender, CALC, NCP, MTRANS, family_history_with_overweight, FAF, FCVC, CH2O, CAEC, FAVC, TUE, SCC, SMOKE
5	0.8495	0.5721	Age, Height, Weight, FCVC, CALC, NCP, Gender, FAF, family_history_with_overweight, TUE, CAEC, CH2O, FAVC, MTRANS, SCC, SMOKE

Table 5: Results of the Random Forest classifier for the different combinations of parameters in the Obesity data set with the train accuracy, test accuracy and features ordered by importance.

NT	**F**	**Train**	**Test**	**Feature Importance**
1	4	0.3803	0.2624	Age, FAF, family_history_with_overweight, TUE
	8	0.6405	0.4751	Weight, Age, CALC, CAEC, family_history_with_overweight, TUE, Gender, SMOKE
	12	0.3513	0.3309	Weight, FAVC, NCP, CALC, FCVC, MTRANS, CAEC, FAF, CH2O, family_history_with_overweight, TUE, SMOKE
	runif	0.9141	0.9286	Age, Height, NCP, family_history_with_overweight, Gender, CH2O
10	4	0.7109	0.3309	Age, Height, Weight, TUE, CALC, MTRANS, CAEC, Gender, FAVC, family_history_with_overweight, CH2O, SCC, NCP, FAF, SMOKE
	8	0.4786	0.2789	Height, Weight, Age, Gender, CALC, FCVC, NCP, CAEC, FAF, CH2O, FAVC, family_history_with_overweight, TUE, MTRANS, SCC, SMOKE
	12	0.4852	0.2813	Height, Weight, Age, NCP, CALC, FAF, TUE, family_history_with_overweight, CH2O, FAVC, MTRANS, Gender, FCVC, SCC, CAEC, SMOKE
	runif	0.6232	0.4420	Age, Weight, Height, FCVC, TUE, NCP, CAEC, FAF, CALC, Gender, SCC, FAVC, CH2O, MTRANS, family_history_with_overweight, SMOKE
25	4	0.8779	0.5579	Age, Weight, Height, family_history_with_overweight, Gender, NCP, FCVC, CAEC, CH2O, FAVC, CALC, FAF, MTRANS, TUE, SCC, SMOKE
	8	0.6907	0.3971	Height, Age, Weight, FCVC, NCP, CALC, Gender, FAVC, family_history_with_overweight, MTRANS, TUE, FAF, CH2O, CAEC, SCC, SMOKE
	12	0.4662	0.3404	Height, Weight, Age, TUE, FCVC, NCP, CALC, FAF, CAEC, CH2O, MTRANS, Gender, family_history_with_overweight, FAVC, SCC, SMOKE
	runif	0.4816	0.2836	Weight, Age, Height, FCVC, NCP, CAEC, TUE, CALC, FAF, family_history_with_overweight, MTRANS, CH2O, FAVC, Gender, SCC, SMOKE
50	**4**	**0.8453**	**0.5626**	Weight, Height, Age, CAEC, FCVC, Gender, FAF, family_history_with_overweight, MTRANS, TUE, CH2O, NCP, CALC, FAVC, SCC, SMOKE
	8	0.4970	0.3427	Height, Weight, Age, FAF, FCVC, NCP, CALC, Gender, TUE, MTRANS, family_history_with_overweight, CAEC, CH2O, FAVC, SCC, SMOKE
	12	0.4620	0.3144	Height, Age, Weight, FAF, CALC, FCVC, NCP, TUE, Gender, CAEC, MTRANS, family_history_with_overweight, FAVC, CH2O, SCC, SMOKE
	runif	0.5936	0.42008	Height, Weight, Age, NCP, FCVC, CALC, FAF, Gender, TUE, MTRANS, CH2O, family_history_with_overweight, CAEC, FAVC, SCC, SMOKE
75	4	0.8489	0.4609	Height, Age, Weight, TUE, FCVC, CALC, Gender, FAF, CAEC, MTRANS, family_history_with_overweight, CH2O, FAVC, NCP, SCC, SMOKE
	8	0.5776	0.3617	Age, Weight, Height, FCVC, NCP, Gender, CALC, TUE, FAF, MTRANS, family_history_with_overweight, CAEC, CH2O, FAVC, SCC, SMOKE
	12	0.4840	0.3357	Height, Age, Weight, NCP, FAF, Gender, CALC, TUE, MTRANS, FCVC, CAEC, CH2O, family_history_with_overweight, FAVC, SCC, SMOKE

runif	0.5983	0.3995	Weight, Age, Height, NCP, FCVC, CALC, TUE, FAF, MTRANS, CAEC, CH2O, family_history_with_overweight, Gender, FAVC, SCC, SMOKE

Table 6: Results of the Decision Forest classifier for the different combinations of parameters in the Obesity data set with the train accuracy, test accuracy and features ordered by importance.

5 Conclusions

In this practical work, we implemented from scratch two well-known ensemble learning classifiers in Python and evaluate the performance and effect of different parameter configurations in three different data sets. In general, the resulting test accuracy for all the data sets is high, showing the good capability of these types of models to classify correctly new instances. If we compare the two models, the Decision Forest usually presents more overfitting than the Random Forest classifier because of not having the bootstrap step. The results follow the theoretical background since the fitted trees are not pruned. If we analyze the performance of the combination of parameters, we can see that usually, a high number of trees produce a better and more consistent classification model, so as a starting point in defining a new model for a new data set, use 75 or 100 trees could be a good option. About the features, the use of more features usually leads to better performance but sometimes with overfitting or with an increase that is not relevant compared to the use of fewer variables. To define the optimal number of features the best option is to perform a grid search.
The execution time of the two models was very similar under the same conditions and we saw that the number of features is the element that affect the most the execution time whereas especially the number of numerical features has a big impact on the complexity of the model.

One advantage of this type of model is the easy interpretability of the fitted trees since they are a combination of binary rules. This characteristic alongside a good performance in the classification task makes these models very useful for a lot of tasks where the understanding of the decision and the features that are more or less important is key.

As a feature work, it could be interesting to also add to the implementation the capability to solve regression tasks and evaluate the performance in different data sets. Also, more work related to making the implementation more efficient can be done in order to reduce the execution time and be able to test the model with large data sets and a high number of trees and features.

References

[1] Leo Breiman. "Random Forests". English. In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. URL: http://dx.doi.org/10.1023/A%3A1010933404324.

[2] Tin Kam Ho. "The random subspace method for constructing decision forests". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), pp. 832–844. DOI: 10.1109/34.709601.

[3] Fabio Mendoza Palechor and Alexis de la Hoz Manotas. "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico". In: *Data in Brief* 25 (2019), p. 104344. ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2019.104344. URL: https://www.sciencedirect.com/science/article/pii/S2352340919306985.

Appendices

Supplementary information about the data sets and the results that we could not add in the main document.

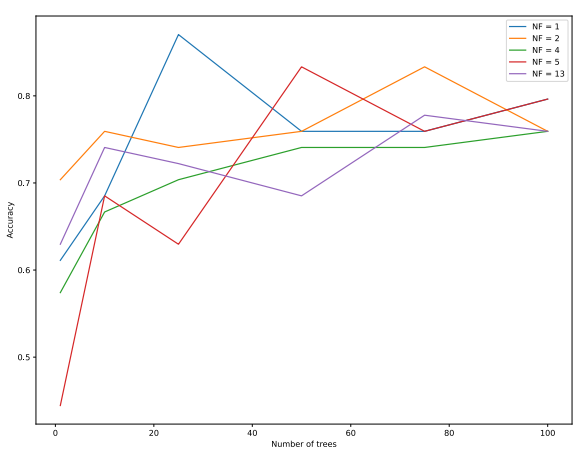
Data set variables

- **Heart Disease**. Data set with 270 instances and 14 attributes where the class variable is `disease`. The set of variables contains numerical and categorical attributes and there is no missing data. The variables are:
  - age: age in years
  - sex: sex (1 = male; 0 = female)
  - chest: chest pain type where 1: typical angina; 2: atypical angina; 3: non-anginal pain; and 4: asymptomatic
  - rbp: resting blood pressure (in mm Hg on admission to the hospital)
  - cholesterol: serum cholesterol in mg/dl
  - sugar: (fasting blood sugar $>$ 120 mg/dl) (1 = true; 0 = false)
  - ecg: resting electrocardiographic results where 0: normal; 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV); and 2: showing probable or definite left ventricular hypertrophy
  - max hr: maximum heart rate achieved
  - angina: exercise-induced angina (1 = yes; 0 = no)
  - oldpeack: ST depression induced by exercise relative to rest
  - slope: the slope of the peak exercise ST segment where 1: upsloping; 2: flat; and 3: downsloping
  - vessels: number of major vessels (0-3) coloured by fluoroscopy
  - thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
  - disease: diagnosis of heart disease (angiographic disease status) where 0: $<$ 50% diameter narrowing; and 1: $>$ 50% diameter narrowing
- **Breast Cancer**. Data set with 699 instances and 10 attributes where the class variable is `Class`. In this case, the set of variables contains only numerical attributes and there are 16 instances that have missing data. The variables are:
  - Clump Thickness: Represents the thickness of the tumour in mm
  - Uniformity of Cell Size: Represents the uniformity in the size of the cells
  - Uniformity of Cell Shape: Represents the uniformity in the shape of the cells
  - Marginal Adhesion: Represents the degree of adhesion of the tumour cells to nearby cells
  - Single Epithelial Cell Size: Represents the size of the epithelial cells. It is a numerical attribute
  - Bare Nuclei: Represents the presence of a nucleus in the tumour cells
  - Bland Chromatin: Represents the degree of chromatin staining in the tumour cells
  - Normal Nucleoli: Represents the size and shape of the nucleoli in the tumour cells
  - Mitoses: Represents the number of mitoses (cell division) in the tumour cells
  - Class: Represents the diagnosis of the tumour (0 for benign, 1 for malignant)
- **Obesity**. Data set with 2111 instances and 17 attributes where the class variable is `NObeyesdad`.

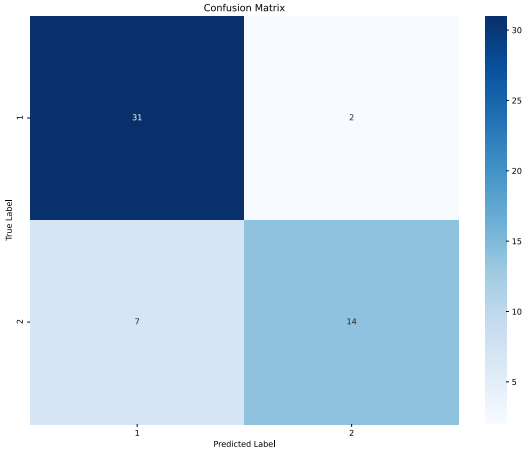
In this case, the set of variables contains only numerical and categorical attributes and there is no missing data. The variables are:

- Gender: Gender of the person (0 = female, 1 = male)
- Age: Age of the person (numerical attribute)
- Height: Height of the person (numerical attribute)
- Width: Width of the person (numerical attribute)
- family_history_with_overweight: ¿Has a family member suffered or suffers from overweight? (0 = no, 1 = yes)
- FAVC: ¿Do you eat high-caloric food frequently? (0 = no, 1 = yes)
- FCVC: ¿Do you usually eat vegetables in your meals? (0 = Always, 1 = Sometimes, 2 = Never)
- NCP: ¿How many main meals do you have daily? (0 = More than three, 1 = three, 2 = Between 1 and 2)
- CAEC: ¿Do you eat any food between meals? (0 = Always, 1 = Frequently, 2 = Sometimes, 3 = No)
- SMOKE: ¿Do you smoke? (0 = no, 1 = yes)
- CH2O: ¿How much water do you drink daily? (0 = More than 2L, 1 = Between 1 and 2L, 2 = Less than a liter)
- SCC: ¿Do you monitor the calories you eat daily? (0 = no, 1 = yes)
- FAF: ¿How often do you have physical activity? (0 = 4 or 5 days, 1 = 2 or 4 days, 2 = 1 or 2 days, 3 = I do not have)
- TUE: ¿How much time do you use technological devices such as cell phones, videogames, television, computer and others? (0 = More than 5 hours, 1 = 3-5 hours, 2 = 0-2 hours)
- CALC: ¿How often do you drink alcohol? (0 = Always, 1 = Frequently, 2 = Sometimes, 3 = No)
- MTRANS: ¿Which transportation do you usually use? (0 = Automobile, 1 = Bike, 2 = Motorbike, 3 = Public transport, 4 = Walking)
- NObeyesdad: mass body index discretised in 7 categories according to the WHO (0 = Insufficient weight, 1 = Normal weight, 2 = Obesity type I, 3 = Obesity type II, 4 = Obesity type III, 5 = Overweight level I, 6 = Overweight level II)

Heart Desease

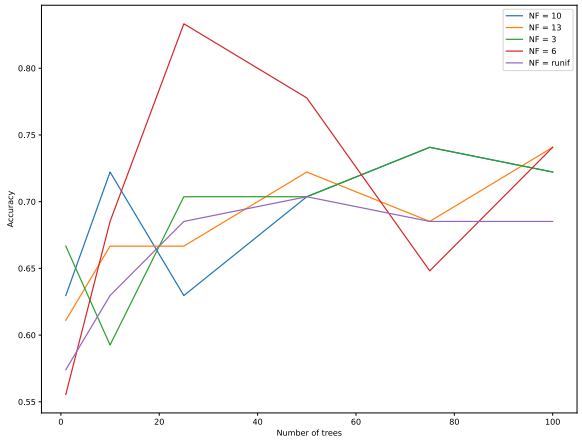


(a)

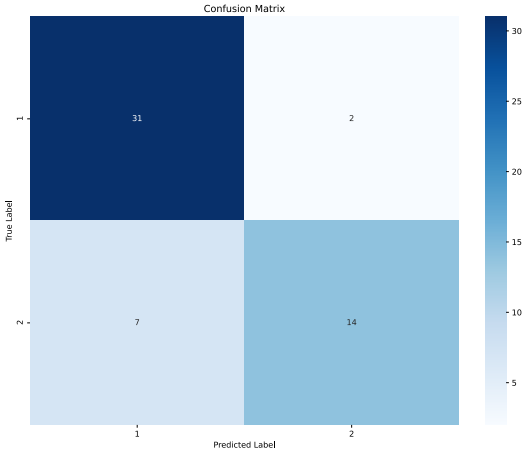


(b)

Figure 1: Visualization of the results of the Random Forest classifier for the Heart data set where (a) shows the accuracy for the different combinations of parameters and (b) the confusion matrix of the selected model.



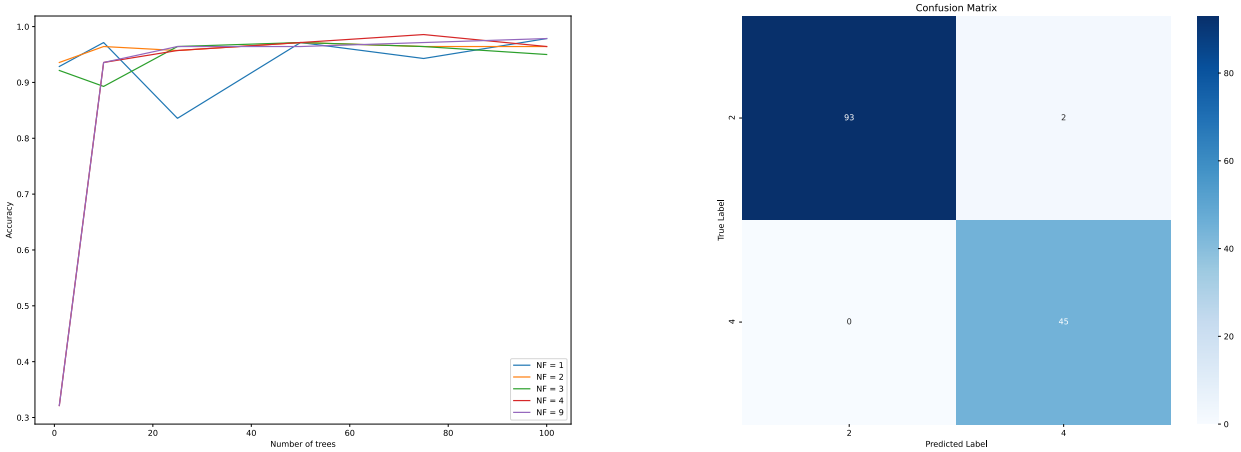
(a)



(b)

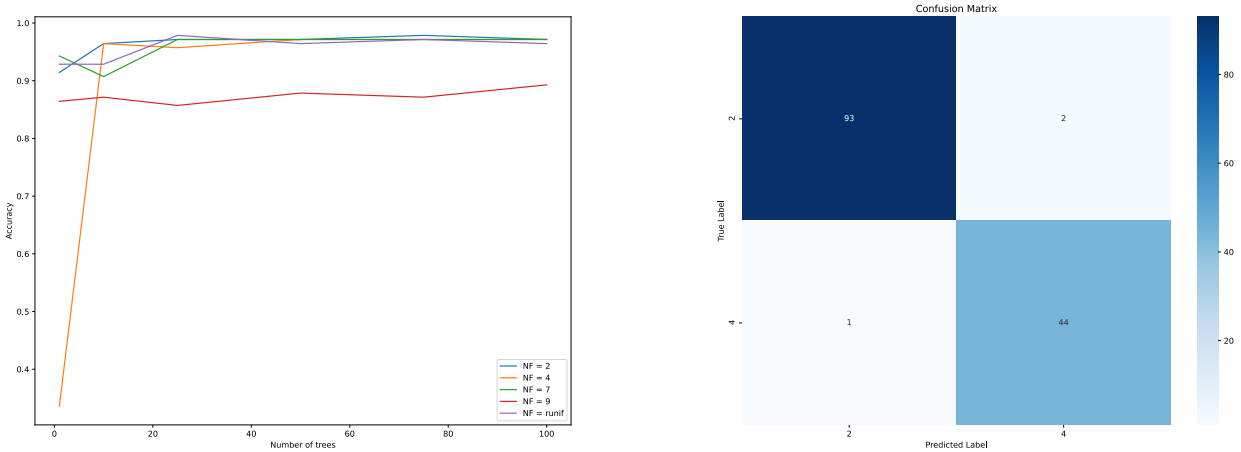
Figure 2: Visualization of the results of the Decision Forest classifier for the Heart data set where (a) shows the accuracy for the different combinations of parameters and (b) the confusion matrix of the selected model.

Breast Cancer



(a) (b)

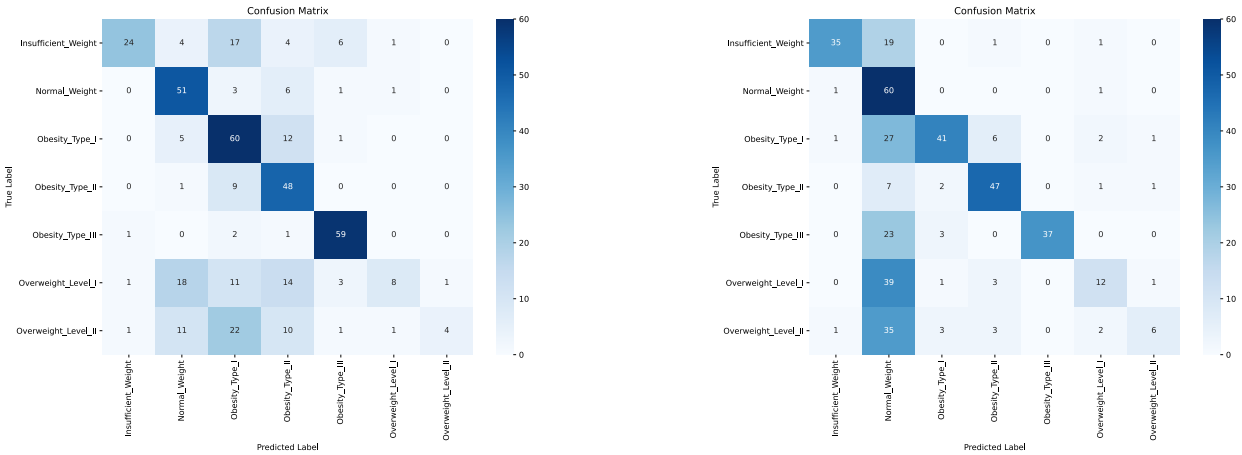
Figure 3: Visualization of the results of the Random Forest classifier for the Breast data set where (a) shows the accuracy for the different combinations of parameters and (b) the confusion matrix of the selected model.



(a) (b)

Figure 4: Visualization of the results of the Decision Forest classifier for the Breast data set where (a) shows the accuracy for the different combinations of parameters and (b) the confusion matrix of the selected model.

Obesity Dataset



(a) (b)

Figure 5: Visualization of the confusion matrix of the selected model for the (a) Random Forest and (b) Decision Forest for the Obesity data set.