

Honours Project Investigation Report

Machine Learning: Experimental Validation of In-house Advanced Data Classification Techniques. Improvement in Random Forest algorithm implemented in R language.

2015

A report submitted as part of the requirements established by
the Learning Agreement between
The Robert Gordon University and
The Technical University of Valencia included in
the Erasmus Program, for the degree
in Computing science for Information Systems at
The Technical University of Valencia, Spain

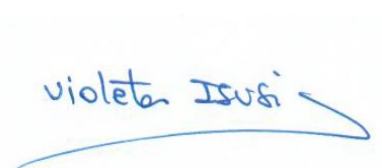
Student author: Violeta Isusi
Supervisors: Dr. Mohamed Gaber
Dr. Cèsar Ferri

Abstract

Nowadays the amount of data generated per day in the world is substantially higher. Therefore, it is crucial to improve the techniques for extracting knowledge in this context. This project research in this line regarding Classification Techniques by means of Decision Trees and specially improving the Random Forest algorithm performance through implementation of Diversified Random Forest algorithm in R language.

Declaration

I confirm that the work contained in this Honours project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

A handwritten signature in blue ink that reads "violeta Isusi". The signature is written in a cursive style and is underlined with a long, sweeping horizontal line.

Signed

Date: 03 May 2015

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr. Mohamed Medhat Gaber, member of Institute for Innovation, Design & Sustainability (IDEAS) at The Robert Gordon University and Dr. Cèsar Ferri Ramírez, member of research groups at Computer Systems and Computation Department (DSIC) at The Technical University of Valencia for their advice, support and guidance along the over months.

I would like also to express my gratitude to Khaled Fawagreh, research student at The Robert Gordon University for his contribution regarding the Diversified Random Forest algorithm implemented in Java.

Furthermore, I would like to show my appreciation for his attention and interest to Dr. Eyad Elyan, lecturer at The Robert Gordon University and Honours Individual Project module leader.

Finally, it has to been said that it was a pleasure for me to have had the opportunity to work with all of them in this interesting and useful project that is at the forefront. It has been a great challenge for me for going in depth in this field of study from now.

Index

1.	Introduction.....	6
2.	Background.....	8
2.1.	Machine Learning and Data Mining.....	8
2.2.	Random Forest Algorithm.....	13
3.	Related work.....	18
3.1.	New Variable Selection Approaches.....	18
3.2.	Modifications in the forest construction and voting mechanism..	19
4.	Diversified Random Forest Algorithm.....	28
5.	Implementation.....	33
5.1.	Java analysis.....	33
5.2.	R implementation.....	38
5.3.	Pseudo-codes.....	40
5.4.	Conclusions: Differences in developing between Java and R.....	41
6.	Experiment.....	43
6.1.	Description.....	43
6.2.	Experiment's details.....	44
6.3.	Testing results.....	45
6.3.1.	Diversified RF implemented in R in comparison with Original RF	
6.3.2.	Comparative between Diversified RF implemented in R and Java	
6.3.3.	Comparing all of them.....	48
6.3.4.	Summary.....	48
7.	Summary and Reflection.....	50
7.1.	Summary.....	50
7.2.	Reflection.....	50
7.3.	Improvements.....	51
7.4.	New approach.....	51
8.	References.....	53
9.	Bibliography.....	56

1. Introduction

About this Thesis

This is the thesis of Violeta Isusi from now on referred to as Isusi's project (2015). This is submitted to fulfill the requirements established by the Learning Agreement between The Robert Gordon University and The Technical University of Valencia included in the Erasmus Program.

Generally, this project is framed within the line of data processing to extract information to apply Data Mining techniques to in order to achieve better results. These better results can be in relation to different areas such as medicine and insurance with early detection of diseases, environment to predict weather, marketing by means of the prediction of the evolution in prices of a specific product, detection of purchasing patterns or determination of groups of clients, banking by detecting fraudulent potential movements in accounts.

It is a reality that the amount of generated data has increased significantly in recent years. In fact, Witten, Hall and Frank (2011 p. 4) noted that "It has been estimated that the amount of data stored in the world's databases doubles every 20 months". Therefore, it is crucial to research in areas as Machine Learning, Data mining and Data Warehouse to improve data processing systems for adapting to this new situation.

Specifically, Isusi's project (2015) follows this trend using one specific technique: Classification by means of the Diversified Random Forest algorithm. Many concepts have yet to be defined and explained but the aim of Isusi's project (2015) is: Improving the Random Forest algorithm's performance through Diversified Random Forest algorithm.

Chapter List

In the first chapter Background main concepts to contextualize Isusi's project (2015) are defined. These major concepts are: Machine Learning, Data Mining, Data Mining Techniques, Classification, Decision Trees and Random Forest Algorithm. It is required to understand this part to be able to continue with the rest.

In the second chapter Related Work is explained. This means, other research that has the same aim as Isusi's project (2015). New Variable Selection Approaches and Modifications in the forest construction and voting mechanism approaches are included in the Related Work.

Diversified Random Forest algorithm is defined and described in depth in the third chapter. In this section, Fawagreh, Medhat and Elyan's (2014b) project is analysed as a starting point of Isusi's project. Furthermore, the Diversified Random Forest's strategy to overtake the Random Forest's performance is clarified as well.

The fourth chapter deals with the implementation task. In this part, Fawagreh, Medhat and Elyan's (2014b) project implementation in Java is found out by reverse engineering and Isusi's project (2015) implementation in R language is illustrated as well. Finally, in this chapter both implementations are compared.

The experiment carried out for Isusi's project (2015) is explained and its results are shown in the sixth chapter. In this section, results from Original Random Forest and Diversified Random Forest implemented in R and Java are compared. Furthermore, comments and conclusions are achieved by analysing the results.

At last, in the seventh chapter a holistic conclusion and reflection is made as a Isusi's project summary.

2. Background

2.1. Machine Learning and Data Mining

Concepts

On the one hand, a first historical and general definition about Machine Learning stated by Samuel (1959 cited by Coursera, 2015) leads to computers being able to learn but they have not to be specifically programmed for this. Learning process is in training with real data that the machine interprets by applying Data Mining techniques. After this process, the machine is capable of making decisions in an autonomous and independent way. Nowadays, Witten (2011, p.8) emphasised two properties in the learning process "... the acquisition of knowledge and the ability to use it".

Mitchell (1997 p. 2, cited by Coursera, 2015) in a more recent and specific Machine Learning definition stated that "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ". In this definition, performance and measure facets are included highlighting the evaluation of the Machine Learning process.

On the other hand, regarding Data Mining, Witten (2011, p.7) noted an operational definition "as the process of discovering patterns, automatically or semi-automatically, in large quantities of data - and the patterns must be useful".

Consequently, by means of Data Mining techniques knowledge from the data is obtained always with some measured performance. Furthermore, Data Mining techniques allows predictions to be made with the actual data (Tan, Steinbach and

Kumar, 2006). Data Mining techniques are needed when the source of data from which the machine has to learn is huge.

Machine Learning and Data Mining are not exclusive concepts but they and other disciplines such as Artificial Intelligence and Statistics are related .

Data Mining Techniques

Going into detail regarding Data Mining techniques, they can be classified by different criteria. Following Tan, Steinbach and Kumar (2006) approach Data Mining techniques can be classified in Descriptive or Predictive according to its aim.

On the one hand, the aim of Descriptive techniques, also named Unsupervised learning techniques is to find patterns from the relationship in data. Depends on how is the pattern, three sub-techniques can be distinguished. Firstly, if the pattern is made from implication rules or feature subsets the sub-technique is Association Analysis. Secondly, if the pattern consists in finding groups with similar characteristics, the sub-technique is Cluster Analysis. Finally, Anomaly Detection is the sub-technique that identifies major differences in data finding atypical cases.

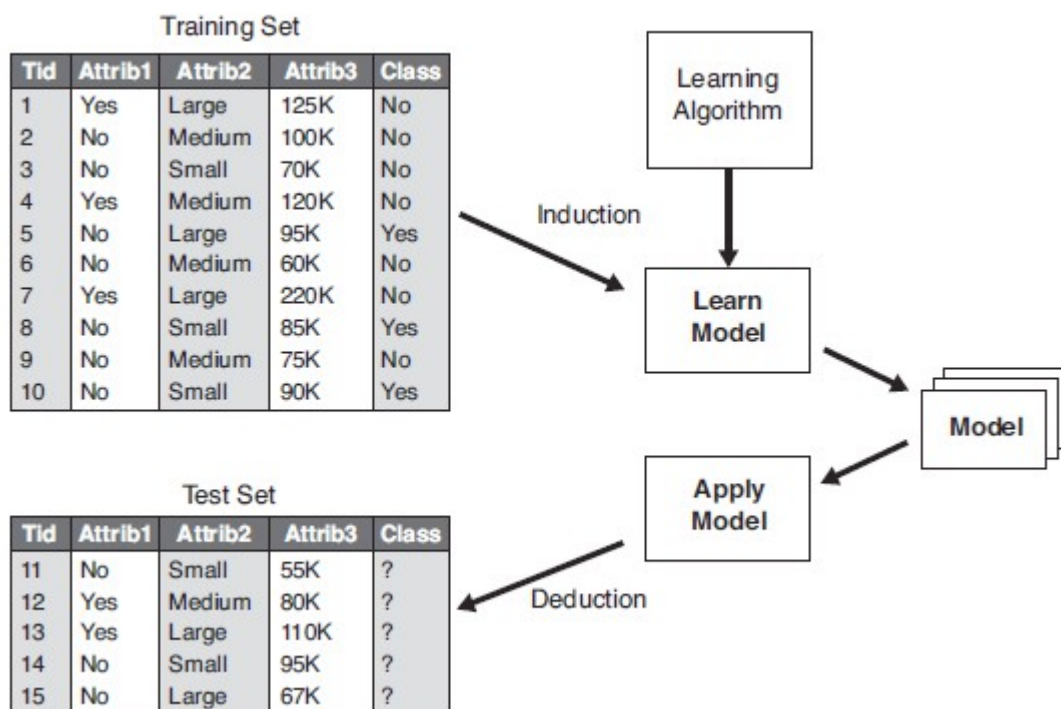
On the another hand, Predictive or Supervised learning techniques starts from real data and attempts to predict one attribute basing on the others. Two sub-techniques are included in this group depends on the attribute to predict. When this attribute is continuous, Regression techniques are applied and in case of nominal attribute Classification techniques are required.

Classification Techniques

As stated Chen, Han, and Yu (1996, p.19) "Data classification has been studied substantially in statistics, machine learning, neural networks, and expert systems and is an important theme in data mining".

Input data in this technique is made of several attributes that inform a nominal Class. The dataset is compound by many instances with different values for each attribute and the Class.

The classification process starts from a dataset with real data. Firstly, this dataset is sampled in two subsets; one for training and another for testing. Secondly, training set is analysed and a model capable to predict the Class value for new data is built. Thirdly, this model is applied to the testing set that contains the values for the attributes but not for the Class. Finally, the model predicts the Class value for each instance in testing set. This last task is named: label unlabelled instances. This process is shown in the follow figure:



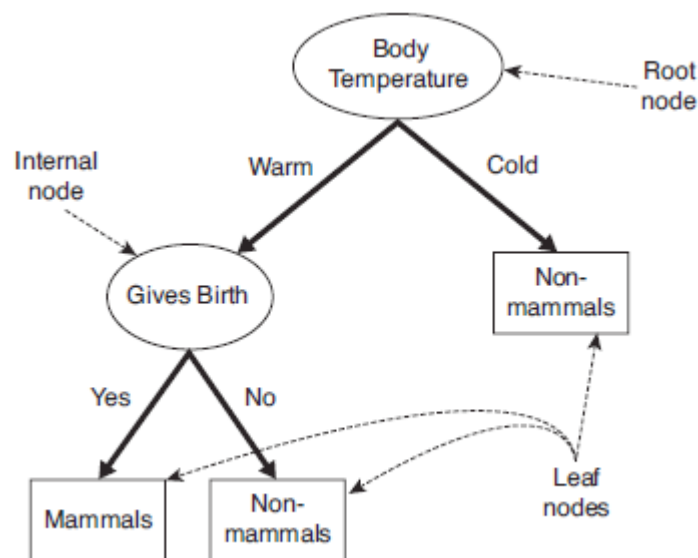
(Tan, Steinbach and Kumar, 2006, p.148, figure 4.3: General approach for building a classification model)

It has to be highlighted again that the model predicts according to a specific and measured performance. Particularly, Accuracy indicator shows the reliability of the model.

There are several techniques for classifying, that means, building the model that label unlabelled instances. Rule-Based classifier, Nearest-Neighbor classifiers, Bayesian classifiers, Artificial Neural Network, Ensemble methods or Decision Trees are included in the main techniques for classifying as noted Tan, Steinbach and Kumar (2006, Contents XV).

Decision trees

Generally, Decision trees are made of three types of nodes: one root node, one or more internal nodes and more than one leaf or terminal node. In Classification techniques Decision trees are usually binary. This means, each time that one node is split two branches are generated. Nodes represent conditions regarding attributes and Class values are in the leafs. An example of decision tree for classification is shown as follows:

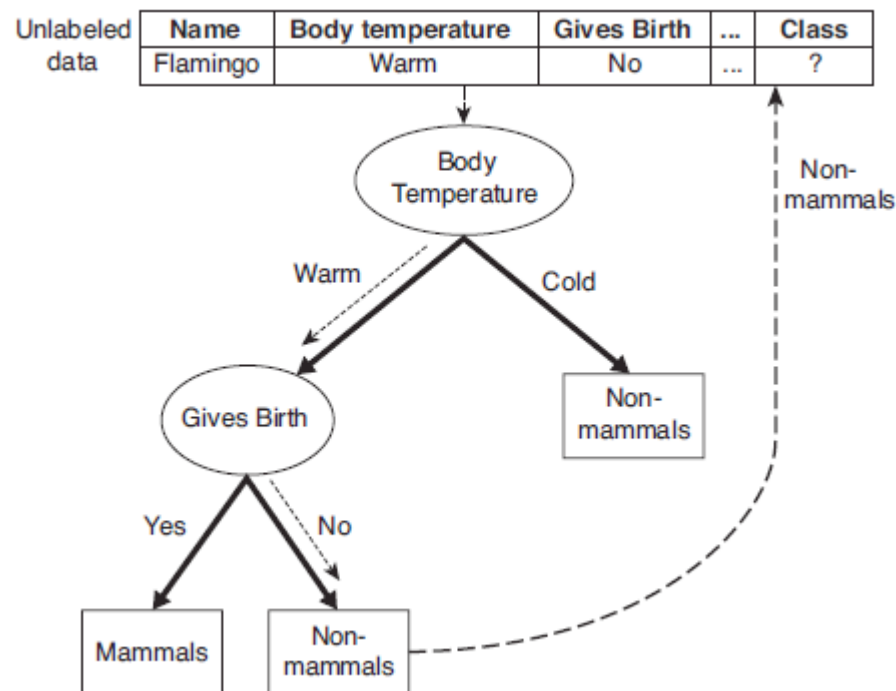


(Tan, Steinbach and Kumar, 2006, p.151, figure 4.4: A decision tree for the mammal classification problem)

"A typical decision tree learning system, adopts a top-down irrevocable strategy" is stated by Chen, Han, and Yu (1996, p.20).

The decision tree starts the top-down strategy by the root node that is split according to a condition regarding one attribute. This process is repeated recursively until Class values are obtained at leafs' tree.

A complete example regarding classification process is shown as follows:



(Tan, Steinbach and Kumar, 2006, p.152, figure 4.5: Classifying an unlabelled vertebrate. "The dashed lines represent the outcomes of applying various attribute test conditions on the unlabelled vertebrate. The vertebrate is eventually assigned to the Non-mammal class")

Different criteria in the splitting process can be applied. Some of them noted by Chen, Han, and Yu (1996, p.20) are Information gain, Gini index and Chi-square test. In Diversified Random Forest section (Chapter 4), Information gain is explained. Regarding Gini index can be defined as follows where T is a data set and " p_i is the relative frequency of class I in T" (Chen, Han, and Yu 1996, p.20) :

$$\text{gini}(T) = 1 - \sum p_i^2$$

Random Tree, Random Forest, Diversified Random Forest, J48, LMT, M5p and DecisionStump are included in the main Decision trees for classification techniques

2.2. Random Forest Algorithm

Random Forest is an ensemble learning that was proposed by Leo Breiman in 2001 who conducted significant amount in research about machine learning in classification and regression trees and Bagging and Boosting methods.(University of California. Department of statistics 2005)

Ensemble learnings

In general, ensemble learnings can be used in classification and in regression as well. From the point of view of classification techniques, an ensemble learning or committee machine is defined as a group of different base classifiers which work separately and join after their results in an only one classifier (Izenman 2013).

Specifically, in Random Forest algorithm "... each tree casts a unit vote for the most popular class ..." (Breiman 2001 p. 6). Therefore Random Forest uses majority voting rule for classification procedure in the follow manner: "Once all the classifiers have been queried, the class that receives the greatest number of votes is returned as the final decision of the ensemble" (Fawagreh, Gaber and Elyan 2014a p. 603)

Generalization error

Over the last years, according to Bousquet and Elisseeff (2002), ensemble learnings are been used to lower the generalization error of learning algorithms. Generalization error or risk is a measure of the performance of an algorithm that depends on the training set, the formula is shown below. Stability and generalization error are related considering that the goal of stability is to achieve bounds on the generalization error as tightly as possible.

Generalization error or Risk of an algorithm or function (A) in a training set (S) is defined as $R(A, S) = E_z [l(AS, z)]$ and hypothesis stability β with respect the loss function l as $\forall i \in \{1, \dots, m\}, E_S, z [|l(AS, z) - l(AS \setminus i, z)|] \leq \beta$ where S is the

training set $S = \{z_1 = (x_1, y_1), \dots, z_m = (x_m, y_m)\}$, m is the size, z is $Z = X \times Y$ drawn independent and identically distributed, where X and $Y \subset \mathbb{R}$, E_z is the expectation when z is sampled, and finally, $l(AS, z)$ is the loss of a function AS with respect z .

Stability

It is important to point out the sensitive analysis that measures the stability of one algorithm to know "how much the variation of the input can influence the output" (Bousquet and Elisseeff 2002 p. 499). An algorithm is unstable if "a small perturbation of the learning set induces major changes in the resulting" (Izenman 2013 p. 505). This knowledge is very useful to design robust algorithms, that means they are stable.

In this context, Random Forest developed by Leo Breiman's hand and is an example of instability so it is not robust. This characteristic of Random Forest is used for achieving better algorithm performance as it is explained above is case of generalization error or risk. Another relevant measure to evaluate algorithm performance is the Accuracy defined as number of correct predictions over total number of predictions.

Bagging and Boosting

Concerning methods for perturbing the training set, there are several but the most important are Bagging and Boosting, where Leo Breiman made a major contribution. On the one hand, the Bagging approach, acronym for bootstrap aggregating, perturbs the training dataset by means of randomization technique and each perturbation is independent of the others. From the learning set using bootstrap replicates new learning sets are achieved making up a set of independent and new learning sets. Statistically speaking, $\{\mathcal{L}^{(B)}\}$ from \mathcal{L} . In detail, each $\mathcal{L}^{(B)}$ is made of " N " cases, drawn at random, but with replacement, from \mathcal{L} . Each (x_n, y_n) may appear repeated times or not at all in any particular $\mathcal{L}^{(B)}$ (Breiman 1996 p.123).

Random Forest is the main representative algorithm in Bagging approach, Breiman (2001 p. 6) defined it as "... a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \dots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors..." "and x is an input vector". The pseudo-code of Random Forest algorithm is shown as follows:

Algorithm 1 RF algorithm

```

{User Settings}
input  $N, S$ 
{Process}
Create an empty vector  $\vec{RF}$ 
for  $i = 1 \rightarrow N$  do
    Create an empty tree  $T_i$ 
    repeat
        Sample  $S$  out of all features  $F$  using Bootstrap sampling
        Create a vector of the  $S$  features  $\vec{F}_S$ 
        Find Best Split Feature  $B(\vec{F}_S)$ 
        Create A New Node using  $B(\vec{F}_S)$  in  $T_i$ 
    until No More Instances To Split On
    Add  $T_i$  to the  $\vec{RF}$ 
end for
{Output}
A vector of trees  $\vec{RF}$ 

```

(Fawagreh, Gaber and Elyan 2014a p. 605, Algorithm 1: RF algorithm)

On the other hand the Boosting approach, is deterministic, each perturbation depends on the previous perturbations. This approach establishes weights for generating perturbations based on the misclassification of the previous perturbations following a history process. AdaBoost is the main representative algorithm in Boosting approach. Going into details, Adaboost calls the algorithm inside of a loop that is repeated T times. Weights are set following a distribution each round, as Freund and Schapire (1999 p. 2) have both stated, "... the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set". The psedocode of Adaboost algorithm is shown as follows where the training set is $\{(x_1, y_1), \dots, (x_m, y_m)\}$, m is the size, $D_t(i)$ is the weight on the distribution, T is the times that the loop is repeated.

Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- Train weak learner using distribution D_t .
- Get weak hypothesis $h_t : X \rightarrow \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right).$$

(Freund and Schapire 1999 p. 3, Figure 1: The boosting algorithm AdaBoost)

Randomization

Regarding randomization characteristics, in Random Forest algorithm is triple. Three layers can be distinguished. The first layer is in the sample (with replacement) from the training data for building the model. The second layer matches with a selection of features from all of them. The size of this selection is always the same, is fixed. At last, the third layer as a consequence of the second, refers to splitting process in each node because the best split is not chosen between all of features but only between selected features.

Pruning

Turning to another issue, algorithms that belong to decision trees technique, in general, can grow with or without prune. On the one hand, pruning means limiting the deepness of the tree, the number of levels the tree has. Therefore, some branches are not built. Pruning is used when quickness is more priority than accuracy. A decision is needed in a short time without consuming too much resources, between some established levels of accuracy. On the other hand, without pruning is used when the maximum level of accuracy that the algorithm is able to achieve, is required. From the point of view of Random Forest approach, the accuracy is the maximum priority follows by the amount of needed resources, such as time or computational efforts. Specifically the main aim in Random Forest algorithm is to improve the accuracy .

Finally, some advantages of Random Forest algorithm are low over-fitting, so rich performance in its predictions, high accuracy level, so low error, quickness, simpleness and easily paralelized. (Fawagreh, Gaber and Elyan 2014a)

3. Related work

3.1. New Variable Selection Approaches

Hapfelmeier and Ulm (2013) stated that several New Variable Selection Approaches achieve higher accuracy and identify informative variables. These approaches are based on the theoretical framework of permutation tests.

A random permutation is applied, several times, to the predictor variable and afterwards, the relation between the predictor variable and the response is evaluated. The measure used for the evaluation is the Permutation Accuracy Importance that "is determined by the mean difference of prediction accuracies observed for each tree (in terms of correct classification rate or mean squared error(MSE)) before and after random permutation of a predictor variable".

Values of Permutation Accuracy Importance close to zero, or negative indeed, indicate that the predictor actually does not predict. Inversely, large values in this measure means that the relation between the predictor and the response is very high.

In the evaluation of the relation between the predictor variable and the response after the permutation, a performance value (p-value) can be applied to reject variables that not satisfy it.

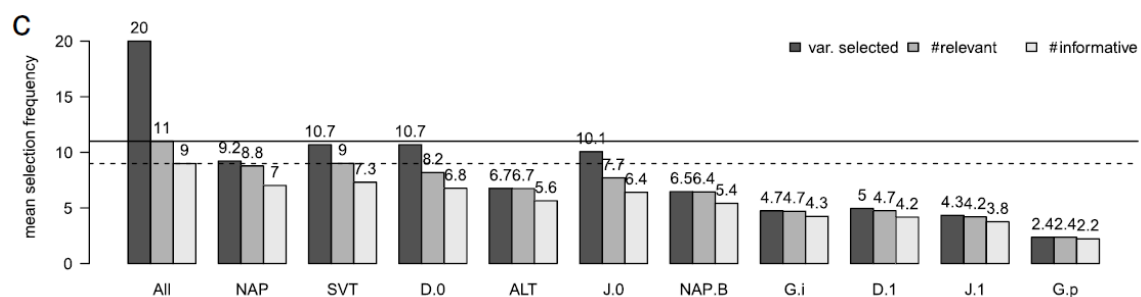
As a consequence of random permutation, associations of the predictor variable with the response and the remaining variable space are broken . Therefore, the accuracy achieved after the random permutation is due to a relevant predictor.

From the point of view of the concept of Relevance, for New Variable Selection Approaches only related variables to the response are Relevant. However, after

random permutations a variable can see its Permutation Accuracy Importance risen due to is related to the response or to another variable.

Summarizing, identifying informative variables is useful to make a decision about what variables should to be included in the model and what not. This can be expressed mathematically as $H_0 : Y \perp X_j$ where Y is the response and X_j , ($j=1,\dots,m$)

Below a graph shown results of the simulation study III made by Hapfelmeier and Ulm (2013) regarding New Variable Selection Approaches. The degree of relevance of each variable is measured by mean selection frequency indicator.



(Hapfelmeier and Ulm, 2013 p.61 Fig.2 (c) "Mean selection frequency of variables and additional information about the amount of relevant and informative variables among them. Reference values are indicated by horizontal lines Relevant Variables: var. 1-11; Informative Variables: var. 1-9.)

3.2. Modifications in the forest construction and voting mechanism

These extensions carried out by Triopoliti et al. (2013) improve the prediction performance of Random Forest algorithm by means of two main strategies. On the one hand, rising the strength, reducing the correlation between trees and introducing diversity in each tree. On the another hand, optimizing the voting mechanism to combine results of each classifier.

In the first group of these modifications, the forest construction is altered in two different approaches named Rotation Forest and RK-RF With Me.

In the second group the voting mechanism is optimized assigning weighted votes or selecting a subset of trees. On the one side, weighted votes are assigned according to the concept of distance explained below. The Nearest neighbour technique is modified in RF With Wv version. On the other side, the aim is to select a subset of trees with better performance by means of feature selection or clustering techniques. Firstly, Modified SFS-RF, Modified SBS-RF and Optimal RF are improvements using feature selection. Secondly, Clustering-RF is the modification in case of building a subset of trees using clustering techniques.

All these modifications are detailed down below.

- FOREST CONSTRUCTION MODIFIED: ROTATION FOREST AND RK-RF WITH ME

History researches focused on that modified only one of the followed aspects. The number of selected features in each split node of each tree or the evaluation process about the impurity. In the first group, RK-RF and Rotation Forest modifications are included. On the contrary, RF with Relieff and RF with me belong to the second group.

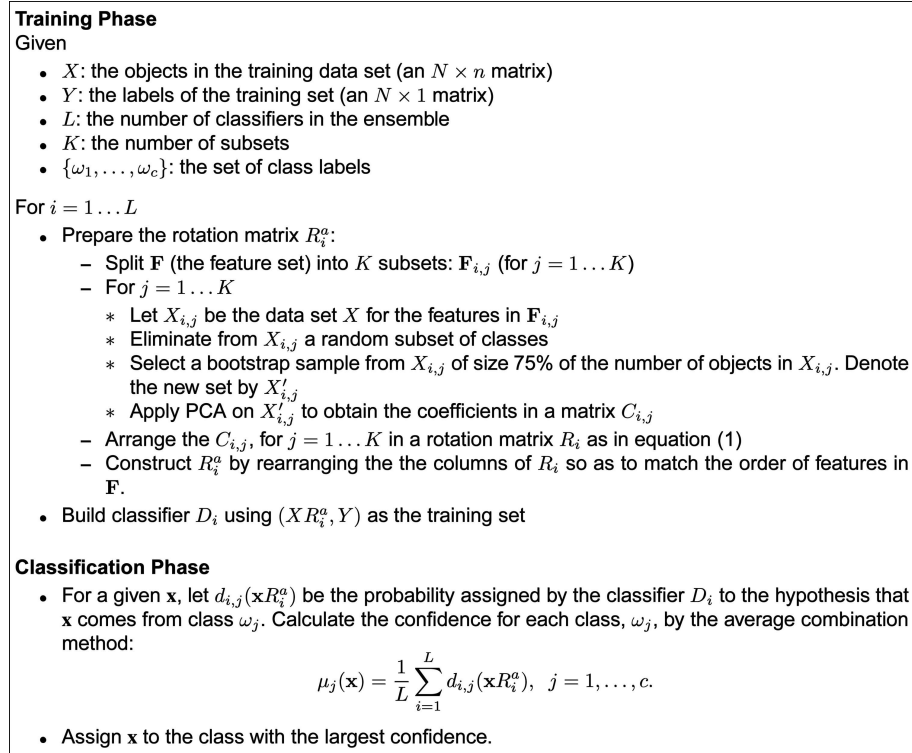
The extensions carried out by Tripoliti et. al (2013) focused mainly on modifying previous Rotation Forest and mixing in only one approach, named RK-RF with me both aspects for modifications indicated above.

- ROTATION FOREST

Firstly, Rotation Forest was proposed by Rodriguez et al. (2006) and Tripoliti et al. (2013 p.46) improved it by two major transformations. On the one hand, "by replacing the induction process of the trees of the forest. More specifically, the feature set is split randomly into r subsets, using Principal Component Analysis (PCA) on each subset." The features are resembled and the process is repeated r times, fact that allows to achieve lower correlation between the trees. On the other hand, in

order to minimize differences with regard to “Classical Random Forest”, the J48 algorithm used by Rodriguez et al. (2006) to build the trees is replaced by the Random Tree.

The Rotation Forest modification pseudo-code is shown as follows:



(Rodriguez et al. 2006, p.1621 Fig.1: Pseudo-code of the Rotation Forest ensemble method.)

○ RK-RF WITH ME

Secondly, RK-RF With Me extension, works on one side, changing into random selection the number of selected features in each split node of each tree. On the other side, altering the evaluation process about how the node impurity is determined using many evaluation measures such as Gini Index, Gain Ratio, Relief, Minimum Description Length and Myopic Relief.

Higher accuracy performance results achieved for this Modifications in the forest construction are shown below where can be compared to Classical Random Forest (named in this project Random Forest algorithm):

Table 3
Classification accuracy of the modifications affecting the construction of the forest (Modifications in italics are proposed in this work).

Classical RF				Modifications affecting the number of features participating on the splitting of the node						Modifications affecting the feature evaluation measure that determines the best split						Modification affecting the number of features and the feature evaluation measure			
				RK-RF			Rotation Forest			RF with ReliefF			RF with me			RK-RF with me			
				T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	
Datasets	D1.	72.80%	41	73.70%	70.00%	20	74.90%	93.10%	45	96.70%	71.10%	85	76.50%	71.10%	65	73.90%	70.00%	27	75.60%
	D2.	86.30%	80	87.20%	87.80%	37	87.80%	98.57%	35	99.00%	86.00%	74	87.50%	86.00%	21	87.80%	87.50%	65	87.50%
	D3.	71.60%	17	71.90%	70.90%	32	73.80%	73.85%	55	74.50%	71.20%	12	74.10%	71.20%	56	73.80%	72.80%	52	74.80%
	D4.	81.30%	36	82.40%	80.20%	26	83.90%	81.27%	25	83.89%	79.00%	71	81.70%	79.40%	74	82.80%	79.80%	14	82.70%
	D5.	94.00%	6	95.30%	93.30%	23	95.30%	98.95%	30	98.97%	94.00%	21	96.00%	93.30%	25	96.70%	93.30%	18	95.30%
	D6.	97.80%	40	98.30%	96.10%	22	98.90%	98.40%	30	99.20%	99.40%	10	99.40%	97.70%	22	98.90%	97.70%	11	98.30%
	D7.	85.10%	18	88.30%	89.20%	13	92.80%	93.33%	25	95.89%	92.30%	54	93.30%	88.70%	27	92.30%	90.20%	64	92.30%
	D8.	96.30%	99	96.80%	96.00%	98	98.60%	97.70%	55	98.20%	95.80%	32	97.20%	97.00%	97	97.00%	96.80%	71	97.00%
	D9.	93.20%	56	94.60%	93.70%	45	94.30%	95.70%	35	96.00%	93.40%	89	94.90%	94.00%	82	95.20%	93.70%	54	94.90%
	D10.	82.80%	60	85.10%	81.30%	96	84.70%	90.38%	30	91.82%	82.30%	45	85.20%	82.70%	86	85.60%	82.30%	73	84.60%
	D11.	75.80%	85	76.00%	76.00%	42	78.30%	78.00%	70	78.40%	76.40%	48	77.10%	75.90%	81	77.60%	77.00%	66	77.60%
	D12.	98.60%	51	99.10%	98.10%	80	98.60%	99.00%	65	99.00%	98.60%	79	98.60%	97.20%	64	98.60%	98.10%	57	99.50%
	D13.	74.60%	20	76.80%	75.90%	11	76.90%	97.70%	75	98.00%	74.90%	76	76.10%	76.50%	30	77.90%	74.70%	18	77.70%
	D14.	82.50%	35	84.40%	82.50%	61	88.00%	82.40%	20	83.30%	81.50%	42	86.10%	87.00%	100	87.00%	81.50%	69	86.10%
	D15.	83.00%	70	83.80%	85.20%	57	85.50%	92.34%	65	92.53%	85.00%	32	85.50%	84.93%	54	85.10%	83.60%	21	85.60%
	D16.	96.50%	51	97.00%	97.30%	20	98.00%	98.50%	70	98.60%	96.75%	11	97.80%	97.50%	9	97.80%	98.00%	23	98.30%
	D17.	62.90%	5	79.20%	85.90%	99	86.00%	87.00%	60	94.88%	85.30%	81	85.30%	85.10%	81	85.70%	85.80%	91	86.10%
	D18.	60.60%	28	78.80%	79.00%	95	79.50%	72.00%	80	79.00%	78.60%	100	78.60%	78.70%	76	80.10%	80.00%	100	80.00%
	D19.	13.30%	42	78.50%	84.70%	26	87.20%	85.44%	65	89.00%	84.60%	65	87.20%	85.50%	52	88.40%	84.60%	15	87.80%
	D20.	81.70%	88	81.90%	82.10%	18	83.10%	81.28%	95	83.14%	81.90%	10	82.20%	82.40%	22	82.80%	81.80%	78	83.30%
	D21.	99.50%	43	99.60%	99.60%	61	99.80%	72.00%	60	73.00%	99.50%	5	100%	99.50%	74	99.80%	99.60%	73	99.80%
	D22.	62.20%	97	64.90%	75.90%	72	78.30%	70.00%	40	78.60%	72.20%	51	73.80%	79.10%	100	79.10%	76.20%	76	78.20%
	D23.	99.40%	15	99.50%	99.50%	5	99.90%	96.65%	75	97.00%	99.50%	5	99.90%	99.40%	8	99.80%	99.80%	5	99.90%
	D24.	94.80%	89	95.70%	97.90%	56	99.40%	98.36%	45	100%	97.10%	55	99.20%	97.10%	7	99.00%	99.20%	100	99.20%

(Tripoliti et al. 2013, p.54, Table3: Classification accuracy of the modifications affecting the construction of the forest)

- VOTING MECHANISM OPTIMIZED

Tripoliti et al. (2013) optimize the voting mechanism in two different ways. On the one hand, weights are assigned to the vote of each tree. On the other hand, only the trees that are most predictive and less correlated are selected in subsets.

- ASSIGNING WEIGHTED VOTES

Firstly, modifications for assigning weights attempt to delimit data similarities between the training instances and the instances to be classified. The major modifications for assigning weights are Nearest neighbour, Dynamic integration and Optimization. Nearest neighbour modifications are explained below.

- Nearest neighbour modifications: RF WITH WV3

In order to measure similarities, Distance functions are applied to determine the degree of similarity. Therefore, this voting process is a Distance Weighted Voting where "... two values are considered to be closer if they have similar classifications"

as noted Tripoliti et al. (2013 p.47)

These modifications examine the schemes, indicated above, of Nearest neighbour, Dynamic integration and Optimization making use of some Distance functions in which the formulas employed by Cunningham (2009) are included.

From the point of view of Nearest neighbour modifications, below can be seen the Distance metric equation between an instance to be classified (q) and each training instance that belong to a training dataset ($x_i \in D$). The distance is work out for each feature in the set of features ($f \in F$) and a weight for each (w_f) is assigned applying a function (δ)

$$d(q, x_i) = \sum_{f \in F} w_f \delta(q_f, x_{if})$$

(Cunningham 2009, p.6 Eq.(1))

Afterwards, according to the result of the distance calculated above, votes are assigned to each classification. Below the equation to work out the vote to assign to y_i class for neighbour x_c is shown. If class labels are equal $1(y_i, y_c)$ returns 1 and 0 in case of being different.

$$Vote(y_j) = \sum_{c=1}^k \frac{1}{d(q, x_c)^n} 1(y_j, y_c)$$

(Cunningham 2009, p.7 Eq.(3))

Tripoliti et al. (2013) in their modification named RF With Wv3 uses a different equation to measure the distance between an instance to be classified (x) and each training instance that belong to a training dataset (x_j). This distance denoted as $d(x, x_j)$ uses two versions of the Value Difference Metric (vdm) named vdm_{af} and vdm_{bf} . Further detailed information about this concept is available in Wilson and Martinez (1997).

$$d(\mathbf{x}, \mathbf{x}_j) = \sqrt{\sum_{f \in F} w_f vdm_f(x_f, x_{jf})^2},$$

$$vdm_{af} = \sum_{i=1}^c |P_{f,x_f,i} - P_{f,x_{jf},i}|^2$$

$$vdm_{bf} = \sqrt{\sum_{i=1}^c |P_{f,x_f,i} - P_{f,x_{jf},i}|^2}$$

(TRIPOLITI et al. 2013, pp.46-47. Eq.(12-14))

Higher accuracy performance results achieved for these Optimized Voting Mechanism Modifications about assigning weighted votes are shown below where can be compared to Classical Random Forest (named in this project Random Forest algorithm):

Table 4_i
Classification accuracy of the modifications affecting the voting mechanism (Modifications in italics are proposed in this study).

Classical RF				Modifications based on nearest neighbors																		Modification based on genetic algorithms																					
				RF with ww-1						RF with ww-2						RF with ww-3						RF with ww-4						RF with ww-5						RF with ww-6									
				T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc										
Datasets	D1.	72.80%	41	73.70%	71.60%	99	75.60%	72.00%	41	73.00%	92.73%	41	97.17%	71.00%	41	72.90%	88.18%	99	91.51%	74.55%	41	76.36%	86.30%	80	87.20%	85.40%	51	87.50%	81.18%	80	81.85%	93.82%	80	96.47%	86.60%	80	87.80%	83.24%	51	86.01%	84.71%	80	86.00%
	D2.	86.30%	80	87.20%	85.40%	51	87.50%	81.18%	80	81.85%	93.82%	80	96.47%	86.60%	80	87.80%	83.24%	51	86.01%	84.71%	80	86.00%	86.30%	17	71.90%	68.30%	25	72.20%	83.87%	17	84.97%	74.51%	17	75.82%	68.30%	17	68.30%	82.26%	25	82.68%	70.97%	17	73.87%
	D3.	71.60%	17	71.90%	68.30%	25	72.20%	83.87%	17	84.97%	74.51%	17	75.82%	68.30%	17	68.30%	82.26%	25	82.68%	70.97%	17	73.87%	81.30%	36	82.40%	79.00%	23	83.50%	85.19%	36	97.41%	92.96%	36	97.78%	80.10%	36	81.70%	94.44%	23	97.04%	80.74%	36	82.59%
	D4.	81.30%	36	82.40%	79.00%	23	83.50%	85.19%	36	97.41%	92.96%	36	97.78%	80.10%	36	81.70%	94.44%	23	97.04%	80.74%	36	82.59%	94.00%	6	95.30%	94.00%	6	96.00%	96.67%	6	97.33%	80.00%	6	83.33%	93.30%	6	93.33%	97.33%	6	100.00%	94.00%	6	94.67%
	D5.	94.00%	6	95.30%	94.00%	6	96.00%	96.67%	6	97.33%	80.00%	6	83.33%	93.30%	6	93.33%	97.33%	6	100.00%	94.00%	6	94.67%	97.80%	40	98.30%	98.30%	77	98.90%	89.44%	40	89.89%	98.33%	40	98.89%	98.30%	40	98.90%	99.44%	77	99.44%	98.89%	40	98.89%
	D6.	97.80%	40	98.30%	98.30%	77	98.90%	89.44%	40	89.89%	98.33%	40	98.89%	98.30%	40	98.90%	99.44%	77	99.44%	98.89%	40	98.89%	98.30%	18	88.30%	89.30%	21	93.40%	85.13%	18	87.37%	89.70%	18	93.40%	85.20%	18	90.30%	96.84%	21	96.84%	90.00%	18	93.68%
	D7.	85.10%	18	88.30%	89.30%	21	93.40%	85.13%	18	87.37%	89.70%	18	93.40%	85.20%	18	90.30%	96.84%	21	96.84%	90.00%	18	93.68%	96.30%	56	94.60%	93.70%	56	94.90%	92.60%	56	92.60%	99.70%	56	99.70%	93.70%	56	93.40%	94.29%	56	94.29%	93.70%	56	94.60%
	D8.	96.30%	56	96.80%	96.80%	78	96.80%	90.70%	99	90.70%	98.94%	99	98.94%	96.00%	99	97.00%	87.54%	78	87.54%	98.97%	99	99.05%	82.80%	60	85.10%	84.60%	80	87.50%	87.10%	60	87.10%	99.00%	60	99.00%	84.20%	60	81.80%	88.10%	80	88.10%	85.24%	60	85.70%
	D9.	93.20%	56	94.60%	93.70%	56	94.90%	92.60%	56	92.60%	99.70%	56	99.70%	93.70%	56	93.40%	94.29%	56	94.29%	93.70%	56	94.60%	75.80%	85	76.00%	75.80%	97	77.20%	88.70%	85	88.70%	88.57%	85	88.57%	76.40%	85	75.50%	77.92%	97	77.92%	76.36%	85	77.92%
	D10.	82.80%	60	85.10%	84.60%	80	87.50%	87.10%	60	87.10%	99.00%	60	99.00%	84.20%	60	81.80%	88.10%	80	88.10%	85.24%	60	85.70%	98.60%	51	99.10%	98.60%	80	99.10%	79.52%	51	79.52%	98.57%	51	98.57%	98.10%	51	99.10%	98.57%	80	98.57%	98.57%	51	99.05%
	D11.	75.80%	85	76.00%	75.80%	97	77.20%	88.70%	85	88.70%	88.57%	85	88.57%	76.40%	85	75.50%	77.92%	97	77.92%	76.36%	85	77.92%	74.60%	20	76.80%	75.30%	61	77.70%	83.29%	20	83.29%	89.17%	20	89.17%	73.50%	20	74.80%	78.71%	61	78.71%	76.35%	20	78.71%
	D12.	98.60%	51	99.10%	98.60%	80	99.10%	79.52%	51	79.52%	98.57%	51	98.57%	98.10%	51	99.10%	98.57%	80	98.57%	98.57%	51	99.05%	82.50%	35	84.40%	84.30%	55	91.00%	87.90%	35	87.90%	97.00%	35	97.00%	85.70%	35	85.70%	86.80%	55	86.80%	88.50%	35	88.50%
	D13.	74.60%	20	76.80%	75.30%	61	77.70%	83.29%	20	83.29%	89.17%	20	89.17%	73.50%	20	74.80%	78.71%	61	78.71%	76.35%	20	78.71%	83.00%	70	83.00%	84.60%	71	85.00%	83.00%	70	83.90%	98.60%	70	98.60%	84.00%	70	84.60%	97.44%	71	97.44%	86.00%	70	86.80%
	D14.	82.50%	35	84.40%	84.30%	55	91.00%	87.90%	35	87.90%	97.00%	35	97.00%	85.70%	35	85.70%	86.80%	55	86.80%	88.50%	35	88.50%	96.50%	51	97.00%	97.90%	52	98.30%	98.70%	51	99.00%	99.00%	51	99.00%	97.70%	51	98.20%	98.80%	52	99.00%	97.60%	51	98.00%
	D15.	83.00%	70	83.80%	84.60%	71	85.00%	83.00%	70	83.90%	98.60%	70	98.60%	84.00%	70	84.60%	97.44%	71	97.44%	86.00%	70	86.80%	62.90%	5	79.20%	85.10%	89	85.30%	86.88%	5	87.58%	89.92%	5	90.64%	85.10%	5	85.30%	85.76%	89	86.45%	85.32%	5	85.81%
	D16.	96.50%	51	97.00%	97.90%	52	98.30%	98.70%	51	99.00%	99.00%	51	99.00%	97.70%	51	98.20%	98.80%	52	99.00%	97.60%	51	98.00%	60.60%	28	78.80%	79.80%	94	80.40%	81.80%	28	83.07%	71.21%	28	72.30%	79.80%	28	80.40%	68.89%	94	81.50%	81.54%	28	84.62%
	D17.	62.90%	5	79.20%	85.10%	89	85.30%	86.88%	5	87.58%	89.92%	5	90.64%	85.10%	5	85.30%	85.76%	89	86.45%	85.32%	5	85.81%	13.30%	42	78.50%	85.90%	78	88.50%	92.25%	42	95.33%	99.35%	42	99.35%	85.80%	42	88.50%	90.32%	78	93.33%	89.33%	42	91.33%
	D18.	60.60%	28	78.80%	79.80%	94	80.40%	81.80%	28	83.07%	71.21%	28	72.30%	79.80%	28	80.40%	68.89%	94	81.50%	81.54%	28	84.62%	81.70%	88	81.90%	86.80%	78	87.50%	92.25%	88	91.56%	91.47%	88	91.56%	81.10%	88	81.40%	84.81%	78	84.89%	81.25%	88	81.25%
	D19.	13.30%	42	78.50%	85.90%	78	88.50%	92.25%	42	95.33%	99.35%	42	99.35%	85.80%	42	88.50%	90.32%	78	93.33%	89.33%	42	91.33%	99.50%	43	99.60%	99.50%	5	100%	73.33%	43	73.33%	73.33%	43	73.33%	99.60%	43	99.60%	70.30%	5	72.20%	99.50%	43	100%
	D20.	81.70%	88	81.90%	86.80%	78	87.50%	92.40%	88	91.56%	91.47%	88	91.56%	81.10%	88	81.40%	84.81%	78	84.89%	81.25%	88	81.25%	62.20%	97	64.90%	79.10%	97	79.10%	87.41%	97	88.00%	81.46%	97	82.00%	62.20%	97	77.90%	62.25%	97	62.66%	63.33%	97	64.67%
	D21.	99.50%	43	99.60%	99.50%	5	100%	73.33%	43	73.33%	73.33%	43	73.33%	99.60%	43	99.60%	70.30%	5	72.20%	99.50%	43	100%	99.40%	15	99.50%	99.30%	8	99.80%	97.01%	15	98.13%	100%	15	100%	99.30%	15	99.50%	98.16%	8	99.30%	99.30%	15	99.53%
	D22.	62.20%	97	64.90%	79.10%	97	79.10%	87.41%	97	88.00%	81.46%	97	82.00%	62.20%	97	77.90%	62.25%	97	62.66%	63.33%	97	64.67%	94.80%	89	95.70%	98.50%	7	99.70%	85.14%	89	86.00%	99.00%	89	99.00%	94.50%	89	94.70%	83.17%	7	84.00%	97.00%	89	96.00%
	D23.	99.40%	15	99.50%	99.30%	8	99.80%	97.01%	15	98.13%	100%	15	100%	99.30%	15	99.50%	98.16%	8	99.30%	99.30%	15	99.53%	94.80%	89	95.70%	98.50%	7	99.70%	85.14%	89	86.00%	99.00%	89	99.00%	94.50%	89	94.70%	83.17%	7	84.00%	97.00%	89	96.00%
	D24.	94.80%	89	95.70%	98.50%	7	99.70%	85.14%	89	86.00%	99.00%	89	99.00%	94.50%	89	94.70%	83.17%	7	84.00%	97.00%	89	96.00%																					

(Tripoliti et al. 2013, p.55, Table 4_i: Classification accuracy of the modifications affecting the voting mechanism)

SELECTING A SUBSET OF TREES

The main idea in these modifications is that a subset of trees will be selected if they achieve better performance than all forests.

The subset of trees can be organised according to two different approaches: Feature selection or Clustering techniques.

- Feature Selection: Modified SFS-RF, Modified SBS-RF, Optimal RF

First of all, a little background about Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) is needed. Both approaches are based on the idea that if the performance of the subset increases as a result of adding (SFS) or removing (SBS) a tree to the subset then this tree is included to the current subset.

Tripoliti et al.'s (2013) research transforms the criteria for evaluating the performance of SFS and SBS creating Modified SFS-RF and Modified SBS-RF that as is shown below increase the accuracy in comparison with "Classical Random Forest".

Some disadvantages of SFS and SBS were overtaken in their Optimal RF modification. This modification is not sequential therefore each iteration not depends on the previous and takes in account all possibilities introducing trees in the subset. In this optimal vision the criteria for including a tree in the subset is double. This means, the accuracy has to be increased and the correlation decreased. By means of this, maiden classifiers are stronger and less correlated, aims for Tripoliti et al. (2013) modifications.

- Clustering: CLUSTERING-RF

Concerning the modifications in clustering techniques to make out a subset of the best trees, Tripoliti et al. (2013) created a Clustering-RF approach. This is characterized by using eight diversity measures, being made up to nine clusters, handling K-means algorithm, stabling achieve the highest accuracy such a criteria for being a member of the cluster and combining the ensemble follow majority voting strategy.

Higher accuracy performance results achieved for these Optimized Voting Mechanism

Modifications about selecting a subset of trees are shown below where can be compared to Classical Random Forest (named in this project Random Forest algorithm):

Classical RF				Modifications based on feature selection techniques												Modification based on clustering techniques						
				SFS-RF			SBS-RF			Modified SFS-RF			Modified SBS-RF						Optimal RF			
				T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc				T100	BT	Acc	
Datasets	D1.	72.80%	41	73.70%	–	29	78.50%	–	52	73.90%	–	19	79.50%	–	37	80.20%	–	71	80.50%	–	20	77.60%
	D2.	86.30%	80	87.20%	–	36	88.10%	–	52	87.80%	–	33	88.10%	–	56	88.10%	–	65	88.40%	–	21	87.20%
	D3.	71.60%	17	71.90%	–	80	73.50%	–	59	71.90%	–	15	73.80%	–	10	74.20%	–	28	74.50%	–	23	74.20%
	D4.	81.30%	36	82.40%	–	10	83.20%	–	50	82.40%	–	23	85.80%	–	42	84.60%	–	43	86.10%	–	26	82.70%
	D5.	94.00%	6	95.30%	–	23	96.70%	–	4	94.70%	–	51	96.70%	–	11	96.70%	–	63	96.70%	–	18	97.30%
	D6.	97.80%	40	98.30%	–	21	98.90%	–	21	98.90%	–	30	98.90%	–	24	98.90%	–	91	98.90%	–	27	98.90%
	D7.	85.10%	18	88.30%	–	24	90.20%	–	32	88.70%	–	39	92.30%	–	46	92.30%	–	70	92.40%	–	24	88.20%
	D8.	96.30%	99	96.80%	–	8	97.20%	–	53	97.00%	–	27	97.90%	–	35	96.80%	–	69	96.80%	–	16	96.80%
	D9.	93.20%	56	94.60%	–	31	95.20%	–	41	94.30%	–	16	95.70%	–	46	94.60%	–	14	95.70%	–	48	94.00%
	D10.	82.80%	60	85.10%	–	18	83.70%	–	86	87.20%	–	15	87.10%	–	56	87.00%	–	69	89.00%	–	29	88.00%
	D11.	75.80%	85	76.00%	–	20	76.80%	–	81	77.00%	–	20	78.30%	–	68	78.50%	–	31	78.80%	–	4	76.80%
	D12.	98.60%	51	99.10%	–	70	98.60%	–	20	98.60%	–	37	99.10%	–	17	99.10%	–	48	99.10%	–	44	98.60%
	D13.	74.60%	20	76.80%	–	17	77.60%	–	79	77.10%	–	13	76.80%	–	17	77.50%	–	20	77.70%	–	49	76.90%
	D14.	82.50%	35	84.40%	–	34	87.30%	–	68	84.40%	–	12	88.20%	–	50	86.20%	–	15	88.70%	–	19	86.20%
	D15.	83.00%	70	83.80%	–	39	85.10%	–	61	85.50%	–	43	86.10%	–	43	87.50%	–	40	87.10%	–	30	87.60%
	D16.	96.50%	51	97.00%	–	28	98.20%	–	26	98.30%	–	25	97.70%	–	26	98.30%	–	27	98.40%	–	35	98.40%
	D17.	62.90%	5	79.20%	–	3	75.50%	–	10	67.20%	–	3	85.10%	–	10	85.10%	–	3	79.50%	–	47	94.90%
	D18.	60.60%	28	78.80%	–	18	78.80%	–	17	73.60%	–	18	79.80%	–	17	79.80%	–	43	79.60%	–	11	79.80%
	D19.	13.30%	42	78.50%	–	35	89.80%	–	28	87.80%	–	28	89.80%	–	28	87.80%	–	91	87.80%	–	35	84.90%
	D20.	81.70%	88	81.90%	–	11	83.80%	–	70	82.20%	–	6	82.40%	–	74	83.00%	–	44	83.70%	–	2	82.60%
	D21.	99.50%	43	99.60%	–	90	100%	–	9	100%	–	90	100%	–	9	100%	–	94	100%	–	42	99.80%
	D22.	62.20%	97	64.90%	–	9	65.60%	–	53	66.20%	–	14	67.60%	–	43	64.90%	–	52	68.90%	–	43	99.50%
	D23.	99.40%	15	99.50%	–	68	99.50%	–	2	99.50%	–	47	99.80%	–	3	99.30%	–	71	99.50%	–	48	99.40%
	D24.	94.80%	89	95.70%	–	56	99.00%	–	14	99.00%	–	36	100%	–	19	98.00%	–	45	99.10%	–	17	99.80%

- APPLYING SIMULTANEOUSLY BOTH PERSPECTIVES: FOREST CONSTRUCTION MODIFIED AND WEIGHTED VOTES ASSIGNED

Going into details, all integrated modifications use combination mechanism factor together with another. This another element can be the evaluation measure to determine the best split in case of RF With Me Wv-1, Wv-3 and Optimal RF With Me. At last the number of features in each splitting node is took in account as another factor by RK-RF With Wv-1 and Optimal RF With RK-RF.

project Random Forest algorithm):

Table 5

Classification accuracy of the modifications affecting the construction and the voting mechanisms (modifications in *italics* are proposed in this study).

Classical RF				Construction and voting mechanism																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
				RF with me and ww-1						RF with me and ww-3						RK-RF with ww-1						RK-RF with me and ww-1						Optimal RK-RF						RF and						Optimal RF and me						Optimal RF with me and RK-RF																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
				T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc	T100	BT	Acc																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
Datasets		D1.	72.80%	41	73.70%	70.00%	62	74.70%	93.64%	41	97.17%	66.10%	82	76.60%	67.10%	96	76.80%	-	58	73.70%	-	61	78.60%	-	53	79.40%	D2.	86.30%	80	87.20%	85.10%	20	86.90%	96.47%	80	96.47%	86.00%	71	88.10%	85.70%	23	87.00%	-	29	89.90%	-	28	88.10%	-	62	87.50%	D3.	71.60%	17	71.90%	70.30%	41	72.20%	76.14%	17	76.14%	69.90%	49	73.50%	70.60%	88	73.20%	-	65	73.20%	-	22	73.50%	-	23	74.50%	D4.	81.30%	36	82.40%	80.10%	53	82.80%	98.88%	36	98.88%	79.40%	51	83.10%	79.80%	29	82.70%	-	31	85.40%	-	21	82.80%	-	20	82.80%	D5.	94.00%	6	95.30%	94.00%	25	96.00%	80.00%	6	80.00%	93.30%	6	95.30%	94.00%	29	96.00%	-	78	95.30%	-	79	95.30%	-	56	95.30%	D6.	97.80%	40	98.30%	97.70%	22	98.90%	98.89%	40	98.89%	96.10%	22	98.90%	97.70%	11	98.30%	-	66	98.30%	-	85	98.30%	-	71	98.30%	D7.	85.10%	18	88.30%	89.70%	42	92.80%	96.50%	18	96.50%	90.80%	64	93.30%	91.20%	21	93.30%	-	74	92.80%	-	67	91.80%	-	24	92.80%	D8.	96.30%	99	96.80%	96.80%	60	97.20%	98.40%	82	98.40%	96.30%	62	97.00%	96.50%	71	97.00%	-	77	96.70%	-	78	97.50%	-	38	97.00%	D9.	93.20%	56	94.60%	94.30%	82	95.20%	95.70%	56	95.70%	93.40%	56	94.30%	94.60%	40	94.90%	-	35	94.90%	-	73	95.40%	-	33	94.30%	D10.	82.80%	60	85.10%	82.70%	93	86.10%	99.04%	60	99.04%	81.80%	94	84.70%	81.80%	73	83.70%	-	30	87.10%	-	35	87.00%	-	19	86.10%	D11.	75.80%	85	76.60%	76.60%	66	77.50%	86.10%	85	86.10%	75.00%	77	77.30%	75.10%	84	77.10%	-	25	80.10%	-	37	78.60%	-	38	79.20%	D12.	98.60%	51	99.10%	97.20%	68	98.60%	99.52%	51	99.52%	98.10%	36	99.50%	98.10%	52	99.10%	-	40	99.00%	-	39	98.60%	-	52	99.50%	D13.	74.60%	20	76.80%	76.60%	76	78.50%	93.80%	20	93.80%	75.90%	33	77.30%	74.90%	11	77.40%	-	13	77.10%	-	7	76.70%	-	9	77.80%	D14.	82.50%	35	84.40%	92.00%	20	96.00%	98.00%	35	98.00%	86.10%	61	90.70%	84.30%	13	89.90%	-	27	89.90%	-	15	92.60%	-	14	92.70%	D15.	83.00%	70	83.80%	85.08%	68	85.10%	99.00%	70	99.80%	84.06%	70	84.60%	85.00%	50	85.40%	-	57	86.60%	-	52	87.30%	-	51	87.50%	D16.	96.50%	51	97.00%	98.40%	50	98.50%	98.70%	51	99.00%	98.00%	46	98.30%	98.50%	52	98.60%	-	45	98.30%	-	54	98.70%	-	53	98.60%	D17.	62.90%	5	79.20%	85.10%	81	85.70%	89.92%	89	89.92%	85.90%	99	86.00%	85.80%	91	86.10%	-	3	80.80%	-	75	86.20%	-	85	86.50%	D18.	60.60%	28	78.80%	78.80%	76	80.10%	71.21%	94	71.21%	79.00%	95	79.50%	68.10%	12	80.40%	-	44	82.00%	-	35	73.70%	-	29	84.90%	D19.	13.30%	42	78.50%	85.30%	89	88.50%	99.35%	58	99.35%	83.40%	26	88.50%	85.90%	25	87.30%	-	70	87.90%	-	46	89.80%	-	35	89.80%	D20.	81.70%	88	81.90%	81.80%	12	82.20%	91.46%	22	91.47%	81.30%	21	81.80%	81.30%	34	83.10%	-	54	83.50%	-	18	83.70%	-	65	83.50%	D21.	99.50%	43	99.60%	99.50%	74	99.80%	73.33%	43	73.33%	99.60%	61	99.80%	99.60%	73	99.80%	-	98	100%	-	99	100%	-	97	100%	D22.	62.20%	97	64.90%	77.60%	90	79.50%	81.45%	97	92.05%	76.30%	53	79.00%	79.80%	52	80.20%	-	68	65.60%	-	47	66.20%	-	45	66.30%	D23.	99.40%	15	99.50%	99.50%	7	99.80%	100%	15	100%	99.50%	5	99.50%	99.90%	9	100%	-	95	99.50%	-	9	99.50%	-	5	99.50%	D24.	94.80%	89	95.70%	98.30%	31	99.40%	99.00%	89	99.00%	97.40%	16	99.70%	98.00%	30	98.10%	-	96	98.00%	-	93	98%	-	90	99.00%

(Tripoliti et al. 2013, p.56, Table4_ii: Classification accuracy of the modifications affecting the voting mechanism)

4. Diversified Random Forest Algorithm

Diversified Random Forest (Diversified RF from now on) is one approach that attempts to increase the performance of Random Forest. Others different approaches that are similar to Diversified RF are explained in the next section named Extensions, Random Forest algorithm related work.

Diversified Random Forest

Diversified RF algorithm is an extension carried out in 2014 by The Institute for Innovation, Design & Sustainability (IDEAS) at the host university Robert Gordon University (RGU). Nowadays, IDEAS is working in more extensions of Random Forest algorithm, specifically regarding pruning.

Fawagreh, Gaber and Elyan (2014b) argued that higher accuracy performance is obtained by Diversified RF, consequently is considered as an improvement in relation to Random Forest algorithm. Results are shown as follows:

Dataset	Number of Features	RF	DRF
soybean	36	77.543106%	75.43103%
eucalyptus	20	20.0%	22.76%
car	7	62.108845%	59.93197%
credit	21	75.97059%	76.147064%
sonar	61	0.7042253%	0.9859154%
white-clover	32	63.333332%	63.809525%
diabetes	9	73.71648%	79.34865%
glass	10	12.328766%	17.123287%
vehicle	19	73.81944%	73.40277%
vote	17	97.97298%	97.36487%
audit	13	96.30882%	96.29411%
breast-cancer	10	71.855675%	75.46391%
pasture	23	41.666668%	40.833336%
squash-stored	25	55.555553%	53.333344%
squash-unstored	24	60.000004%	61.11111%

(Fawagreh, Gaber and Elyan 2014b, p. 90, Table 1: Performance Comparison of RF & DRF)

Regarding Isusi's project (2015) results, they can be seen in a below the Experiment section and confirm a similar success.

Strategy: Diversity and Weighted Voting Technique

The scenario built by Diversified RF to achieve these better performance results is more diverse and uses a weighted voting technique instead of the majority voting to combine the results of different classifiers in an only one, like ensemble learnings do.

On the one hand, based on ensemble learnings strategy, Diversified RF evolves its in the same way. As was explained in previous section, ensemble learnings, where Random Forest algorithm is included, achieve better accuracy as a result of introducing diversity by means of randomization. Likewise, Diversified RF strategy is to introduce more diversity in Random Forest algorithm. By using random Subspaces when the classifier builds the model Diversified RF is injecting more diversity.

As is indicated by Fawagreh, Gaber and Elyan (2014b, p. 87), the number of Subspaces is worked out according to the size of Diversified RF needed. The ratio is as follows, where α is a factor between 0 and 1 ($0 < \alpha \leq 1$) and Z is the size of the Diversified RF algorithm.

$$\text{Number of Subspaces} = \alpha \times Z$$

Each subspace conform a sub-forest with a number of trees calculated in accordance with the ratio between the size of the Diversified RF algorithm over the number of Subspaces, as follows:

$$\text{Number of Trees} = Z / \text{Number of Subspaces}$$

On the other hand, Diversified RF combines the results of each classifier my means of weighted voting technique. This technique lies in establishing a weight for each

classifier vote and combining its results, according to this, in only one that is the final result of the Diversified RF. This weight is calculated for each Subspace using the measures of Information gain of each attribute to predict the class in the training set, the entropy and the number of attributes that there are in this training set.

Absolute Predictive Power

Going into details, following the method of Cuzzocrea et al. (2013) cited by Fawagreh, Gaber and Elyan (2014b, p. 87), this weighted voting technique is named Absolute Predictive Power (APP) and worked out as follows, where S is the training set, $|Att(S)|$ is the Number of attributes in training set, $I(S, Att)$ is the Information gain of each attribute in the training set and $E(S)$ is the Entropy of the training set

$$APP(S) = (1 / |Att(S)|) \times \sum (I(S, Att) / E(S))$$

Information gain and Entropy measures.

After that, some concepts have to be defined. Firstly, Information gain is a measure for evaluating the quality of attributes. Kononenko and Kukar (2007, p.157) stated that "It is defined as the amount of information, obtained from the attribute, for determining the class". Therefore, in any way is indicative of dependence between an attribute and the class. Secondly, Entropy is an impurity measure. Entropy is calculated comparing maiden predictions based on testing set with data in training set. Consequently, is a measure of uncertainty as well. Entropy values may have a range of values between 0 and 1 ($0 \leq E(S) \leq 1$) where $E(S)$ equal to 0 means maximum uncertainty and 1 maximum certainty. If $E(S)$ is equal to 1 all predicted values are equal to values from training set.

As Fawagreh, Gaber and Elyan (2014b, p. 88) indicated, Entropy of one dataset is worked out as follows, where S is the given dataset, K is the number of instances in its, x_i refers to a generic instance of S and $p_i(x_i)$ indicate the probability that x_i occurs in S .

$$E(S) = \sum_{i=1}^k - p_i(x_i) \log_2 p_i(x_i)$$

$D(S)$, the Diversity of one dataset S , is the result of $E(S)$ over its K instances, as follows:

$$D(S) = E(S) / K$$

Concerning the degree of Diversity, a comparative analysis between Random Forest and Diversified RF is explained below in the Experimental section. Furthermore, consequences for the accuracy depending on the degree of Diversity are illustrated in the Conclusion section.

Pseudo-code

Finally, all these theoretic concepts are implemented in source code in any language. In the case of Fawagreh, Gaber and Elyan's research, Diversified RF algorithm was implemented in Java using Weka (2009) library. Moreover, the Isusi's project used R language, Rweka (Hornik, K., Buchta, C., Zeileis, A., 2009) and Weka (2009) libraries to carry out it. Details about the implementation of this project can be seen in The Experimental section and in the CD delivered as well. Lastly, the specification in pseudo-code to develop Diversified RF algorithm in any language is shown below:

Algorithm 1 RF algorithm

```
{User Settings}
input  $N, S$ 
{Process}
Create an empty vector  $\vec{RF}$ 
for  $i = 1 \rightarrow N$  do
  Create an empty tree  $T_i$ 
  repeat
    Sample  $S$  out of all features  $F$  using Bootstrap
    sampling
    Create a vector of the  $S$  features  $\vec{F}_S$ 
    Find Best Split Feature  $B(\vec{F}_S)$ 
    Create A New Node using  $B(\vec{F}_S)$  in  $T_i$ 
  until No More Instances To Split On
  Add  $T_i$  to the  $\vec{RF}$ 
end for
{Output}
A vector of trees  $\vec{RF}$ 
```

(Fawagreh, Gaber and Elyan 2014b, p. 89, Algorithm 1: Diversified Random Forest Algorithm)

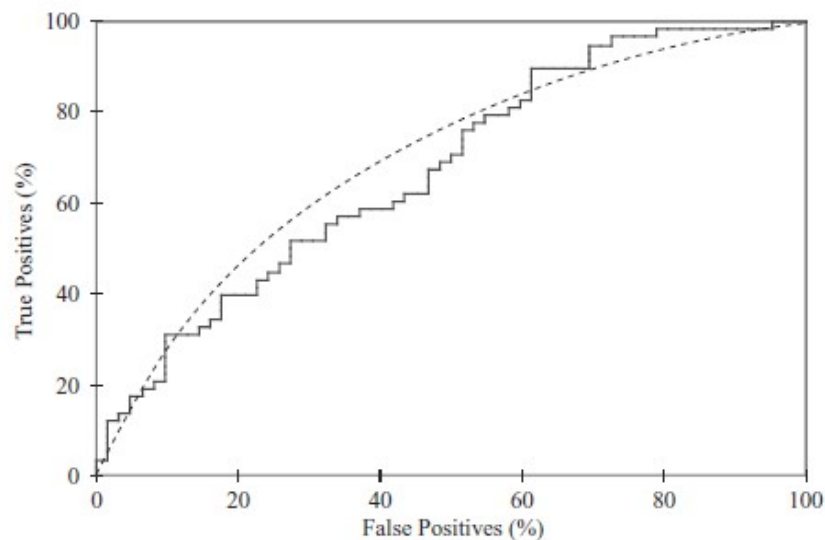
5. Implementation

5.1. Java analysis

Isusi's project (2015) implementation, starts from the study of a Java's implementation of Diversified RF algorithm delivered by Fawagreh, Gaber and Elyan (2014b) . The implementation format provided was a Java project with some input data ready to be imported in any IDE (Integrated Development environment). This implementation has no attached documentation therefore, in the early phases the source code was analysed, the functionality and details regarding the differences in relation to Random Forest algorithm were extracted.

Generally, the code implements the Diversified Random Forest algorithm functionalities and calculates its performance. Accuracy, Entropy and Area Under the Curve are included in the performance indicators. Regarding the Accuracy and Entropy (Diversity), they are defined and explained in previous sections. Concerning Area Under the Curve indicators, Fawagreh, Gaber and Elyan (2014b) chose specifically the Area Under the ROC (Receiver Operating Characteristic).

Going into detail, Witten, Hall and Frank (2011 p.172) stated that "ROC curves depict the performance of a classifier without regard to class distribution or error costs. They plot the true positive rate on the vertical axis against the true negative rate on the horizontal axis". An example of this is shown in the follow graph:



(Witten, Hall and Frank 2011 p.173, Fig. 5.3: A sample ROC curve)

The Area Under the ROC indicator is used to evaluate the model taking in account the ensuing statement, noted by Witten, Hall and Frank (2011 p.177) "The larger area the better the model".

The University of Waikato: Weka

This implementation makes use of Weka (2009) library, that is result of Machine Learning Group's research at The University of Waikato (2015). The Weka (2009) project started in 2005 and is continually being developed. The current version for development is Weka (2009) 3.7.12 and the Fawagreh, Gaber and Elyan's project (2014b) developed the Java code in 3.7.9 version. For further information about Weka (2009) versions or to download the software see The University of Waikato (2015) website (downloads section) for current versions and Sourceforge (2015) for old versions.

Weka (2009) library brings together several Machine Learning algorithms implemented in Java. This library can be used for developing in Java projects or can be executed in stand-alone way as well.

Input data: Arff format

The Java code is implemented for reading datasets as input data in Arff format. The University of Waikato (2008) noted that Arff is an abbreviation of Attribute-Relation File Format that was specifically created by Machine Learning Group's research in The University of Waikato (2008) to be used for Weka (2009) library.

This file format contains instances that have in common a set of attributes. Its structure is divided in two sections: the Header and the Data. Firstly, the Header in turn contains two different parts. On the one hand, the Relation Declaration that is a string with the dataset's name. On the other hand, the Attribute Declaration contains a series of attributes that inform the Class in order to predict it. The Class is located in this Attribute Declaration in the last position.

Secondly, the Data Section is a list of instances. Each instance is in one different row and rows are separated by carriage returns. This data section is related to the attribute section as is indicated by The University of Waikato (2008) "The order the attributes are declared indicates the column position in the data section". The columns are separated by a comma.

These three sections can be seen in a small segment of the well-known Iris dataset shown as follows:

```
% 1. Title: Iris Plants Database
%
@RELATION iris

@ATTRIBUTE sepallength REAL
@ATTRIBUTE sepalwidth REAL
@ATTRIBUTE petallength REAL
@ATTRIBUTE petalwidth REAL
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
6.6,2.9,4.6,1.3,Iris-versicolor
4.7,3.2,1.3,0.2,Iris-setosa
7.7,2.6,6.9,2.3,Iris-virginica
5.0,3.6,1.4,0.2,Iris-setosa
7.4,2.8,6.1,1.9,Iris-virginica
```

Java implementation's details

Random Forest and Random Tree algorithms are included amongst the algorithms implemented in the Weka (2009) library. Obviously, the Random Forest could not be used to implement the Diversified Random Forest version because they have methodology differences.

Consequently, the Java implementation uses some functionalities provided by Weka (2009) library such as following Classes: ArffLoader, Instances, InfoGainAttributeEval and Evaluation; However for other functionalities tailored code is needed.

The major tasks tailored were: building the subspaces of trees, determining the weights for subspace and calculating the performance. From the point of view of building the subspaces of trees, Fawagreh, Gaber and Elyan's strategy used a collection of Random Trees models which were put into groups making subspaces.

Forest's structure

Specifically the built forest is made up of 500 trees clustered in 10 subspaces of 50 trees each. For determining the performance, the Diversified Random Forest is executed 10 times (runs = 10).

Referring to the formulas that were explained in the Diversified Random Forest section about Diversified Random Forest running, to work out it noted by Fawagreh, Gaber and Elyan (2014b, p. 87), the forest's structure is the following:

$$\begin{aligned}\text{Number of Trees} &= Z / \text{Number of } \textit{Subspaces} \\ 500 &= Z / 10\end{aligned}$$

Therefore the size of the Diversified RF algorithm is $Z = 5000$

$$\text{Number of } Subspaces = \alpha \times Z$$

$$10 = \alpha \times 5000$$

Therefore the factor is $\alpha = 0.002$

Random selection of features

Regarding the selection of features for building the subspaces, 75 percent of them will be selected in randomization manner.

Working out the weights for voting

Furthermore, Information gain of each attribute and entropy in the dataset are used for determining the weight for each subspace. Going into details, for each subspace training dataset, Information gain of each attribute is obtained from the Java Class `InfoGainAttributeEval` using the method `evaluateAttribute(attribute)`. This amount is divided by the entropy and accumulated for every attribute. Finally, this quantity is pondered by the number of attributes that are in the training dataset. Pseudo-code regarding this can be seen as follows:

For 10 subspace training datasets

For each attribute

Extract Information gain

For each attribute

Accumulate (Information gain / entropy)

Pondered weight = Accumulated value / number of attributes

Using this weight the maximum weight possible per subspace can be worked out as follows:

Maximum weight per subspace = Number of trees per subspace * Subspace training dataset's Weight

By extension the Maximum weight in total for all subspaces can be calculated accumulating the Maximum weight per subspace for all of them.

This Maximum weight in total for all subspaces divided by 2 is used as a minimum level required to determine the correct classifications when predictions from Random Tree models built with testing dataset are compared to training data.

Testing instances are classified by the Random Tree model using the method `classifyInstance(testing Instance)` of the Java Class `RandomTree`.

Perturbing training set by Bagging method

Training datasets are perturbed before building the Random Tree model using the method `resample(new Random())` of the Java Class `Instances`. This method as the name indicates, makes a re-sample in randomization manner by default with replacement. The resulting dataset is used to build the model with the Random Tree algorithm.

5.2. R implementation

Implementation in R language makes use of `RWeka` (Hornik, Buchta and Zeileis, 2009) library available to download and read documentation in R Foundation for Statistical Computing (2015) website.

Regarding versions used, these are: 3.1.2 (2014-10-31) "Pumkin Helmet" version for R language; 0.4-23 version for `RWeka` (Hornik, Buchta and Zeileis, 2009) library that has dependencies with the `Rwekajars` and `rJava` packages which versions used were 3.7.12-1 in case of `Rwekajars` and 0.9-6 in case of `rJava`. At last the package `stringr` is used for strings treatments and its version is 0.6.2. All this software is available for downloading at R Foundation for Statistical Computing (2015b) website.

From the point of view of developing, the chosen IDE (Integrated Development environment) was RStudio, using the version 0.98.1091 – 2009-2014 This is available for downloading in

RWeka (Hornik, Buchta and Zeileis, 2009) library is an interface to use Weka (2009) library in R code. In fact, the Weka (2009) library is embed in the Rwekajars package. However, unfortunately not all Classes and methods from Weka (2009) library are available in Rweka (Hornik, Buchta and Zeileis, 2009) library. From the point of view of programming, as a developer, the main functionalities that are missed were the Classes Instance and Instances to save built models and instances from testing datasets. This point will be extended below where Java and R implementations are compared.

The main use of Rweka library was for:

- Reading data from Arff files using `read.arff(file)` instruction,
- Evaluating information gain of each attribute using the `InfoGainAttributeEval(formula, data, subset, na.action, control = NULL)` instruction,
- Building the models using the `make_Weka_classifier("weka/classifiers/trees/RandomTree")` instruction,
- Making predictions using `predict(model,newdata=testingDataset)` instruction,
- Evaluating the built model using the `evaluate_Weka_classifier(model,newdata=testingDataset,complexity=TRUE,classification=TRUE)` instruction

Going into detail about the model built, as is noted above, it was used the RandomTree functionality from RWeka (Hornik, Buchta and Zeileis, 2009) library that implements the Class RandomTree from Weka (2009) library rather than RandomTree package from R language (R Foundation for Statistical Computing 2015b). The main reason for this choice was to later better compare the results between Isusi's project (2005) and Fawagreh, Gaber and Elyan's project (2014b) that built the models with

Weka (2009) functionalities.

Concerning tailored code, when Rweka (Hornik, Buchta and Zeileis, 2009) library could not offer help, R language was used. The major tasks had been reading files, treating the resulting dataframes, parsing to extract the information and manage vectors and matrixs to save intermediate calculations for working out model's performance.

5.3. Pseudo-codes

Pseudo-code Java

```
For 10 runs
  For 10 subspaces
    Building training and testing datasets
    //training phase
    For 500 trees
      For 10 subspaces
        Working out entropy and weight
        Re-sampling training dataset
        Building model with Random Tree algorithm
      Working out Maximum weight in total for all subspaces
    //testing phase
    For 500 models
      Working out the Area Under the ROC
    For all instances in testing dataset
      For 10 subspaces
        For 500 models
          Classifying the instance
          Calculating voting weight for this instance
        Working out Entropy and Accuracy

    Showing performance's results for each run
  Showing performance's results for the Diversified RF calculating the average
```


Pseudocode R

```
For 10 runs
  For 10 subspaces
    Building training and testing datasets
  //training phase
  For 500 trees
    For 10 subspaces
      Working out entropy and weight
      Working out Maximum weight in total for all subspaces
      Re-sample training dataset
    Building model with Random Tree algorithm
    Predicting the Class
    Working out the Area Under the ROC
  For N instances in testing dataset
    Saving intermediate calculations
  For N instances in testing dataset
    For 10 subspaces
      For 500 trees
        Saving intermediate calculations
    Working out Entropy and Accuracy
  Showing performance's results for each run
Showing performance's results for the Diversified RF calculating the average
```

5.4. Conclusions: Differences in developing between Java & R

Firstly, implementation in R had to manage strings from the datasets by parsing and matching to locate and extract information from Relation Declaration, Attribute Declaration, Class attribute (always in last position in the dataset) and Data section. These tasks are easier to manage in Java implementation by using Weka (2009) library because they are not available in Rweka (Hornik, Buchta and Zeileis, 2009).

Secondly, the code structure regarding loops is different between Java and R

implementations because Weka (2009) library allows to save the models and instances from datasets as an objects in RandomTree, Instances and Instance Classes.

Therefore, in R implementation calculations for working out Entropy and Accuracy were more complex because intermediate results have to be stored in vectors and matrixs.

Thirdly, regarding the re-sampling, in the training datasets in the Java implementation every RandomTree model is built from a training dataset re-sampled whereas in the R implementation the training dataset is re-sampled only once per subspace. Therefore in Java implementation the training set is more perturbed than in R implementation. Below in the Experimental section where the results are explained this point will be reasoned.

Finally, an improvement in Java implementation can be suggested. That is to remove a loop of 500 models that run all models's instances in order to evaluate each model. This part is located after a loop that build the 500 models, therefore it can be performed immediatly after models are created in the same loop rather than later in another independent loop.

6. Experiment

6.1. Description

The main aim of the experiment is to improve the Original RF's performance, specially the prediction's accuracy. The strategy to succeed in it, as is explained in previous sections, is introducing more diversity by means of randomization and implementing weighted voting technique to combine the results like ensemble learning.

The experiment is implemented in R language starting from a Java implementation of Diversified RF provided by Fawagreh, Medhat and Elyan (2014b). The input datasets for running the algorithm, are obtained from UCI Machine Learning Repository that is available online at Lichman (2013b) website.

Testing phase was done. It is compulsory in Isusi's project (2015) in order to extract performance's results. With this purpose, it is used a sample made of ten datasets and each is run ten times. The Diversified RF's performance is evaluated by means of Entropy, Accuracy and Area under the ROC indicators. For further information about these indicators see previous sections.

In order to provide results to make a comparative analysis, the testing process includes the experiments of Original RF, and Diversified RF implemented in Java and R algorithms.

Design

Several R files are implemented for carrying out different tasks. These tasks are:

- Building the training and testing datasets starting from the complete dataset.
- Implementing Diversified RF algorithm.

- Implementing Original Random Forest algorithm

From the point of view of Random Forest implementation, it was implemented using Rweka (Hornik, K., Buchta, C., Zeileis, A., 2009) functionalities calling the Class RandomForest from Weka (2009) library rather than using the Random Forest package from R language (R Foundation for Statistical Computing 2015b). The main reason for this choice was to be closer to Java implementation to make the comparative analysis more reliable.

6.2. Experiment's Details

Input data

The datasets selected from UCI Machine Learning Repository that are available online at Lichman (2013b) website, are in Arff format. These are as follows:

- Heart Disease Databases (Heart-c)
- Heart Disease Databases (Heart-statlog)
- Pima Indians Diabetes Database
- Lymphography Domain
- Hepatitis Domain
- BUPA liver disorders
- Cardiotocography
- Balance Scale Weight & Distance Database
- Glass Identification Database
- Image Segmentation data

These datasets were selected according to the criteria of Not to have missing values, otherwise this could be a factor to influence in the model's performance in different way for each dataset depends on the quantity of them.

Random selection of features

Diversity is added randomizing and selecting one subsection of features in each dataset. The percentage chose was 75 percentage. The same proportion as in Java implementation.

Re-sample

Regarding the bagging technique for perturbing the training data, this is re-sampled once for each subspace.

6.3. Testing results

6.3.1. Diversified RF implemented in R in comparison with Original RF

R							
DIVERSIFIED RF				ORIGINAL RF			
DATASET	ACCURACY %	AUC %	ENTROPY – DIVERSITY		ACCURACY %	AUC %	ENTROPY – DIVERSITY
Cgt	85.254	85.418	0.204		84.582	89.947	0.394
Hepatitis	82.448	69.358	0.366		82	73	0.125
Heart-statlog	80.227	70.753	0.608		74.157	73.596	0.487
Heart-c	79.393	28.087	0.261		66	NAN	0.496
Diabetes	78.07	63.339	0.671		74.117	68.343	0.388
Lymph	77.872	33.55	0.342		68.75	NAN	0.45
Segment	95.703	95.394	0.084		96.358	99.591	0.401
Balance-scale	68.592	63.9	0.512		79.227	74.243	0.496
Glass	63.043	62.718	0.326		65.714	NAN	0.453
Liver-disorders	55.752	56.994	0.77		66.666	65.507	0.482

Table 1: Diversified RF implemented in R in comparison with Original RF

It can be seen in the result's table above how Diversified RF overtakes Original RF in the majority of cases regarding the Accuracy indicator. Going into depth in the Accuracy analysis, working out the percentage of improvement and deterioration in all cases, the results are shown as follows:

R		
ACCURACY PERFORMANCE		
DIVERSIFIED RF OVER ORIGINAL RANDOM FOREST		
DATASET	IMPROVEMENT %	DETERIORATION %
Cgt	0.672	
Hepatitis	0.448	
Heart-statlog	6.07	
Heart-c	13.393	
Diabetes	3.953	
Lymph	9.122	
Segment		0.655
Balance-scale		10.635
Glass		2.671
Liver-disorders		10.914
TOTAL	33.658	24.875

Table 2: Accuracy performance in Diversified RF implemented in R over Original RF

Above, as the result's table illustrates, the total percentage of improvement in Diversified RF is 33.658 in the ten selected datasets. Therefore, Diversified RF's results demonstrate better Accuracy in its predictions from the point of view of quantity and quality.

Regarding the Entropy indicator, it can be seen in the table below how this is higher in Diversified RF than in Random Forest as well. Therefore, these results are evidence that the Diversified RF implementation achieves its aim of adding Diversity in comparison with the Random Forest algorithm.

R		
ENTROPY		
DIVERSIFIED RF OVER ORIGINAL RANDOM FOREST		
DATASET	IMPROVEMENT %	DETERIORATION %
Cgt		0.19
Hepatitis	0.241	
Heart-statlog		0.121
Heart-c	0.235	
Diabetes	0.283	
Lymph		0.108
Segment		0.317
Balance-scale	0.016	
Glass		0.127
Liver-disorders	0.288	
TOTAL	1.063	0.863

Table 3: Entropy performance in Diversified RF implemented in R over Original RF

Concerning the Area Under the Curve (AUC) indicator, is always better in Original RF rather than Diversified RF. However, this supposed better performance does not translate in better results for the Accuracy indicator. Below, this point is explained in more detail in the subheading Comparing all of them.

6.3.2. Comparative between Diversified RF implemented in R and Java

As is illustrated in the result's table below, Diversity in Java implementation is always greater than in R implementation. This was an expected result because in fact, in the Java implementation each training dataset is re-sampled before building the model and in R implementation only once per subspace.

DIVERSIFIED RF							
R					JAVA		
DATASET	ACCURACY %	AUC %	ENTROPY – DIVERSITY		ACCURACY %	AUC %	ENTROPY – DIVERSITY
Cgt	85.254	85.418	0.204		87.029	85.941	0.271
Hepatitis	82.448	69.358	0.366		79.4	65.388	0.456
Heart-statlog	80.227	70.753	0.608		80.337	70.5	0.691
Heart-c	79.393	28.087	0.261		83.8	70.14	0.292
Diabetes	78.07	63.339	0.671		75.647	62.012	0.772
Lymph	77.872	33.55	0.342		80.833	72.71	0.392
Segment	95.703	95.394	0.084		96.059	94.771	0.124
Balance-scale	68.592	63.9	0.512		78.985	75.275	0.502
Glass	63.043	62.718	0.326		65.857	74.166	0.412
Liver-disorders	55.752	56.994	0.77		62.631	56.378	0.868

Table 4: Comparative between Diversified RF implemented in R and Java

Regarding the Accuracy indicator, eight in ten cases in the Java implementation have higher Accuracy than the R implementation. It is the majority of them. However the 2 cases that achieve more Accuracy in R implementation succeeded in it with lower Diversity.

Therefore, these results confirm that perturbing the training dataset applying bagging techniques rise the model's performance in ensemble learnings.

AUC		
DIVERSIFIED RF IN R OVER DIVERSIFIED RF IN JAVA		
DATASET	IMPROVEMENT %	DETERIORATION %
Cgt		0.523
Hepatitis	3.97	
Heart-statlog	0.253	
Heart-c		42.053
Diabetes	1.327	
Lymph		39.16
Segment	0.623	
Balance-scale		11.375
Glass		11.448
Liver-disorders	0.616	

Table 5: The Area Under the Curve (AUC) indicator in Diversified RF implemented in R over Diversified RF implemented in Java

Regarding the Area Under the Curve (AUC) indicator, seems that its results are a bit arbitrary. However, always that accuracy is better, the Area Under the Curve (AUC) indicator is better as well. Consequently, this indicator can be useful for supporting a better performance but not for determining it. Other indicators such as Accuracy and Entropy have to be taken in account as well.

6.3.3. Comparing all of them

The most interesting to highlight in this holistic comparative is that Java and R versions do not overtake Original RF in almost the same datasets. These are named Segment, Balance-scale and liver disorders as it can be seen in the table below.

R						JAVA			
DIVERSIFIED RF			ORIGINAL RF			DIVERSIFIED RF			
DATASET	ACCURACY %	AUC %	ENTROPY – DIVERSITY	ACCURACY %	AUC %	ENTROPY – DIVERSITY	ACCURACY %	AUC %	ENTROPY – DIVERSITY
Cgt	85.254	85.418	0.204	84.582	89.947	0.394	87.029	85.941	0.271
Hepatitis	82.448	69.358	0.366	82	73	0.125	79.4	65.388	0.456
Heart-statlog	80.227	70.753	0.608	74.157	73.596	0.487	80.337	70.5	0.691
Heart-c	79.393	28.087	0.261	66	NAN	0.496	83.8	70.14	0.292
Diabetes	78.07	63.339	0.671	74.117	68.343	0.388	75.647	62.012	0.772
Lymph	77.872	33.55	0.342	68.75	NAN	0.45	80.833	72.71	0.392
Segment	95.703	95.394	0.084	96.358	99.591	0.401	96.059	94.771	0.124
Balance-scale	68.592	63.9	0.512	79.227	74.243	0.496	78.985	75.275	0.502
Glass	63.043	62.718	0.326	65.714	NAN	0.453	65.857	74.166	0.412
Liver-disorders	55.752	56.994	0.77	66.666	65.507	0.482	62.631	56.378	0.868

Table 6: Comparative between Original RF and Diversified RF algorithms implemented in R and Diversified RF implemented in Java

Focusing in the Glass dataset which R implementation does not achieve overtake, Java implementation only overtakes Original RF in a small quantity. Specifically this quantity is 0.143 %. Probably adding more diversity or applying other techniques it can be overtaken more comfortably.

6.3.4. Summary

Experiment's results confirm theory explained in previous sections about The higher Diversity the better the model's performance.

The Diversified RF implementation in R achieves its aim regarding overtaking the Random Forest algorithm's performance. However, the best results in Accuracy

performance are for Java implementation due to it has the highest diversity as well.

The last conclusion is an observation about the Area Under the Curve (AUC) indicator. This is that always that the Accuracy is high, the AUC is as well but not conversely.

7. Summary and Reflection

7.1. Summary

Isusi's project (2015) started making a preliminary research regarding concepts such as Machine Learning, Data mining, Classification techniques and Decision trees. This is made to contextualize the Random Forest and Diversified RF algorithms.

Afterwards, Random Forest and related work close to Diversified RF research is analysed and shown in depth.

The main tasks of Isusi's project (2015) were the Diversified RF implementation in R and the experiment that tests its performance. Experiment's results achieve the aim of improving Random Forest performance.

Furthermore, a comparative analysis is made regarding Random Forest and Diversified RF implemented in R and in Java.

7.2. Reflection

Isusi's project (2015) is an unusual configuration, original and interesting. It is unusual because models for classification are built by Weka (2009) library inside of implementation in R language. On the one hand, the majority of classification technique's research developed in R uses packages RandomTree and RandomForest from R Foundation for Statistical Computing (2015b) website. On the another hand, research developed in Java, regarding classification techniques, uses Weka (2009) library.

Derived from above, Isusi's project (2015) can be complemented with other implementation following the same functionality but using packages RandomTree and

RandomForest from R Foundation for Statistical Computing (2015b) website. Furthermore, if the same datasets are tested the results obtained could be useful to evaluate Weka (2009) library and R packages (R Foundation for Statistical Computing 2015b) performance.

7.3. Improvements

Firstly, in order to rise the Diversified RF, implemented in R, performance, more diversity can be added in the design. As is noted in Implementation and Experiment sections, Diversified RF, implemented in Java, has more diversity than R version because every training dataset is re-sampled before building the model. However, training datasets are re-sampled only once per subspace in Diversified RF implemented in R.

Consequently, more diversity can be added in R version of Diversified RF by re-sampling in the same manner than Java version. If this is done speed performance has to be evaluated due to execution process in R is so slower than in Java. Then, testing process could be too much time-consuming.

Secondly, R implementation improvement would be to tailor the code for reading datasets in any format. Current version only is capable to read datasets in Arff format but it will be interesting to read in csv format as well.

7.4. New approach

Firstly, If I were to tackle the project again sure that R implementation would be more efficient than currently. Overall regarding dataframes treatment.

Secondly, concerning the datasets for testing, other repositories can be used. Depending on used repository Isusi's project (2015) can be applied to different area for different needs. For example others repositories that can be used are: The GeoNames geographical database, Airport airline and route data and CMU StatLib

Datasets Archive. For further information about datasets' repositories see RDM (2015) website.

8. References

Words 9300

- Bousquet, O., Elisseeff, A., (2002). *Stability and Generalization*. [Online] Journal of Machine Learning Research 2, PP. 499-526. Available from: <http://www.eecs.berkeley.edu/~brecht/cs294docs/week6/02.Bousquet.stability.pdf> [Accessed 06 April 2015]
- Breiman, L., (2001). *Machine Learning: Random Forest*. [Online] The Netherlands: Kluwer Academic Publishers. 45(1):5-32. Available from: <http://link.springer.com/article/10.1023/A:1010933404324> [Accessed 10 April 2015].
- Breiman, L. (1996). *Bagging predictors*. *Machine learning*. [Online] The Netherlands: Kluwer Academic Publishers. 24(2), 123-140 Available from: <http://link.springer.com/article/10.1007/BF00058655> [Accessed 06 April 2015].
- Chen, M. S., Han, J., and Yu, P. S. (1996). *Data mining: an overview from a database perspective*. [Online] Knowledge and data Engineering, IEEE Transactions on, 8(6), 866-883. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=553155&tag=1 [Accessed 30 April 2015]
- Coursera. (2015). *Machine Learning*. [Online] Available from: <https://www.coursera.org/learn/machine-learning> [Accessed 30 April 2015]
- Cuzzocrea, A., Francis, S. L., and Gaber, M., (2014). *An information-theoretic approach for setting the optimal number of decision trees in random forests*. In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on. IEEE. [Online]. October. pp. 1013-1019. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6721930 [Accessed 16 April 2015]
- Cunningham, P., (2009). *A taxonomy of similarity mechanisms for case-based reasoning*. Knowledge and Data Engineering, IEEE Transactions on, [Online] 21(11), 1532-1543. Available from: <https://www.csi.ucd.ie/files/ucd-csi-2008-1.pdf> [Accessed 18 April 2015]
- Fawagreh, K., Gaber, M. and Elyan, E., (2014a). *Random forests: from early developments to recent advancements*. *Journal of Systems Science & Control Engineering*. [Online] 2(1), pp. 602-609. Available from: <http://www.tandfonline.com/doi/full/10.1080/21642583.2014.956265#.VSLUhZOGOT-> [Accessed 06 April 2015].
- Fawagreh, K., Gaber, M. and Elyan, E., (2014b). *Diversified Random Forests Using Random Subspaces*. Intelligent Data Engineering and Automated Learning–IDEAL, LNCS 8669, 2014.

- Springer International Publishing.[Online], pp. 85-92. Available from:
http://link.springer.com.ezproxy.rgu.ac.uk/chapter/10.1007%2F978-3-319-10840-7_11
 [Accessed 16 April 2015]
- Freund, Y, Schapire R.E., and Abe, N., (1999). *A short introduction to Boosting*. [Online] Journal-Japanese Society For Artificial Intelligence. 14(5), pp 771-780. Available from: <http://www.yorku.ca/gisweb/eats4400/boost.pdf> [Accessed 10 April 2015].
 - Hall, M. A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., (2009). *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, Volume 11, Issue 1.
 - Hapfelmeier, A., and Ulm, K., (2013). *A new variable selection approach using random forests*. *Computational Statistics & Data Analysis*. [Online]. 60, pp. 50-69. Available from: <http://www.sciencedirect.com/science/article/pii/S0167947312003490> [Accessed 17 April 2015]
 - Hornik, K., Buchta, C., Zeileis, A., (2009) *Open-Source Machine Learning: R Meets Weka*. [Online] *Computational Statistics*, 24(2), 225-232. Available from: doi:10.1007/s00180-008-0119-7 [Accessed 28 April 2015]
 - Isusi, V., (2015). *Machine Learning: Experimental Validation of In-house Advanced Data Classification Techniques. Improvement in Random Forest algorithm implemented in R language*.
 - Izenman, A.J., (2013). *Modern Multivariate Statistical Techniques: Regression, Classification and Manifold learning*. Corrected 2nd edition. New York: Springer
 - Kononenko, I., and Kukar, M. (2007). *Machine learning and data mining*. 1st published. Chichester, UK: Horwood.
 - Lichman, M. (2013a). *UCI Machine Learning Repository* [online] Irvine, CA: University of California, School of Information and Computer Science. Available from: <http://archive.ics.uci.edu/ml> [Accessed 28 April 2015]
 - Lichman, M. (2013b). *UCI Machine Learning Repository .Iris Dataset*. [online] . Irvine, CA: University of California, School of Information and Computer Science. Available from: <https://archive.ics.uci.edu/ml/datasets/Iris/> [Accessed 28 April 2015]
 - Mitchell, T. (1997). *Machine Learning*, Boston: McGraw Hill.
 - R Foundation for Statistical Computing (2015a) *RWeka: R/Weka interface*. [online]. CRAN. Available from: <http://cran.r-project.org/web/packages/RWeka/index.html> [Accessed 28 April 2015]
 - R Foundation for Statistical Computing (2015b) *The Comprehensive R Archive Network*. [online]. CRAN. Available from: <http://cran.r-project.org/> [Accessed 28 April 2015]
 - Rdm (2015) . *RdataMining: R and Data Mining: Resources: Free Datasets*. [Online]. Available from: <http://www.rdatamining.com/resources/data> [Accessed 01 May 2015]
 - Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J., (2006). *Rotation forest: A new classifier ensemble method*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, [Online]

- 28(10), 1619-1630. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1677518 [Accessed 19 April 2015]
- RStudio, Inc. (2014) *Rstudio products* [online] Available from: <http://www.rstudio.com/products/rstudio/> [Accessed 28 April 2015]
 - Samuel, A.L., (1959) *Some studies in machine learning using the game of checkers*. [Online] IBM Journal of Research and Development, 44(1.2), 206-226. Available from: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5389202> [Accessed 28 April 2015]
 - Sourceforge (2015). *Weka. Machine learning software to solve data mining problems*. 3.7.9 version [online] Available from: <http://sourceforge.net/projects/weka/files/weka-3-7/3.7.9/> [Accessed 26 April 2015]
 - Tan, P., Steinbach, M., Kumar, V., (2006). *Introduction to Data Mining*. Boston: Pearson Addison Wesley
 - The University of California. Department of Statistics. (2005). *Leo Breiman 1928-2005* [online]. Berkeley, CA : University of California. Available from: <https://www.stat.berkeley.edu/~breiman/> [Accessed 06 April 2015].
 - The University of Waikato (2015). *Weka* . [online] Available from: <http://www.cs.waikato.ac.nz/ml/weka/index.html> [Accessed 26 April 2015]
 - The University of Waikato (2008). *Attribute-Relation File Format (ARFF)*. [online] Available from: <http://www.cs.waikato.ac.nz/~ml/weka/arff.html> [Accessed 26 April 2015]
 - Tripoliti, E. E., Fotiadis, D. I., and Manis, G., (2013). *Modifications of the construction and voting mechanisms of the random forests algorithm*. [Online] Data & Knowledge Engineering. 87, 41-65. Available from: <http://www.sciencedirect.com/science/article/pii/S0169023X13000748> [Accessed 18 April 2015]
 - Wilson, D. R., and Martinez, T. R., (1997). *Improved heterogeneous distance functions*. [Online] JAIR, 6, 1-34. Available from: <https://www.jair.org/media/346/live-346-1610-jair.pdf> [Accessed 18 April 2015]
 - Witten, I. H., Hall, M. A., Frank, E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. 3 rd Edition. Morgan Kaufmann

9. Bibliography

- *Computing for Data Analysis*. (2012) Playlist video. Directed by – Roger Peng. [Online] Available from <https://youtu.be/EiKxy5IecUw> [Accessed 02 May 2015]
- edX. (2015) *Courses* [online] Available from: <https://www.edx.org/courses>. [Accesses 02 May 2015]
- Girke, T. UCR .Institute for Integrative Genome Biology. Programming *in R* [online] Available from: <http://manuals.bioinformatics.ucr.edu/home/programming-in-r> [Accessed 02 May 2015]
- Hernández Orallo, J., Ramírez Quintana, M. J., & Ferri Ramírez, C. (2004). *Introducción a la Minería de Datos*. Madrid: Pearson Educación SA.
- Kabacoff, R.I, (2014) Quick-R. *Accessing the power of R*. [online] Available from: <http://www.statmethods.net/> [Accessed 02 May 2015]
- The University of Kansas (2008) *Weka Manual* [online] Available from: http://www.ittc.ku.edu/~nivisid/WEKA_MANUAL.pdf [Accesses 02 May 2015]
- Togaware Pty Ltd (2014) *Hands-On Data Science with R*. [online] Available from: <http://onepager.togaware.com/> [Accessed 02 May 2015]
- Weka Sourceforge. *Weka* [online]. Available from: weka.sourceforge.net/doc.stable/ [Accesses 02 May 2015]
- Williams, G. (2011). *Data mining with Rattle and R: the art of excavating data for knowledge discovery*. Springer Science & Business Media.