Report
LSMF2013 Project
Quantitative Data Analysis

S. Kurin, L. Parmentier, and M. Goudjil
Professor M. Saerens

Abstract—In this study, different classification models including logistic regression, k-nearest neighbors (k-NN), support vector machines (SVMs), decision tree, random forest, artificial neural networks (ANNs) as well as feature selection/extraction techniques (stepwise logistic regression, linear discriminant analysis (LDA), principal component analysis (PCA)) are studied to find out those that demonstrate high classification accuracy for "Forest type mapping" dataset in UCI database. Based on the results, it is useful to perform LDA since it creates a parsimonious model (with only 3 components instead of 27 features). In combination with SVM-RBF model it reveales the classification accuracy of 96.97% that is the highest result among all studied models. Alternatively, stepwise logistic regression (with 7 features selected) that demonstrates the accuracy of 94.47% can be used. The classification accuracy after doing PCA is much lower compared to other feature selection/extraction techniques that might be explained by unsupervised nature of the PCA algorithm.
Furthermore, we examine how cost of classification error can affect the solution. The optimal classification rule is deduced theoretically and verified empirically. The theoretical threshold θ_{th} avobe which we categorize the observation as belonging to positive class depends on the non-diagonal elements of the cost matrix.

Index Terms—Classification accuracy, logistc regression, k-NN, random forest, artificial neural networks.

1 Introduction

The dataset ("Forest type mapping" in UCI database) we analyzed contains training and testing data from a remote study which mapped different forest types based on their spectral characteristics at visible-to-near infrared wavelengths, using ASTER satellite imagery. The obtained forest type map can be used to identify and/or quantify the ecosystem services provided by the forest.
Our report consists of 5 parts. In "Data exploration" we explain variables and classes presented in the dataset. In "Variable selection, transformation and recoding" part we analyze feature selection/extraction techniques including stepwise logistic regression, linear discriminant analysis (LDA), principal component analysis (PCA). In the part "Modelling" different classification models are considered. In the part "Cost scenario" we study how cost of classification error can affect the solution. In "Reject option" part we show how to deal with a situation when making decision is too risky.

2 Data exploration

The training and testing datasets we studied contains geographically weighted variables calculated for two tree species, (Cryptomeria japonica)"Sugi" and (Chamaecyparis obtusa)"Hinoki", that were used in addition to spectral information to classify the two species and one mixed forest class "Mixed Deciduous broadleaf" (i.e. this class groups different deciduous tree species). The fourth class represents the different types of land covered and is named Other. Thus, we have 4 different classes s, h, d, o that we also denote as w_1, w_2, w_3, w_4 for a total of 523 observations (198 in the training set and 325 in the testing set).
In total, 27 features x_1, x_2,..., x_{27} were used for image classification (9 spectral bands $b1$, $b2$,...,$b9$ and 18 similarity measures $pred\_minus\_obs\_H\_b1$,.., $pred\_minus\_obs\_H\_b9$, $pred\_minus\_obs\_S\_b1$,..., $pred\_minus\_obs\_S\_b9$ calculated from the inverse distance weighting

(IDW) interpolated values).
In order to visualize more precisely the data, different plot can be produced: barplot, correlation matrix, boxplot.

A bar plot (Fig. 1) allows us to see the proportion of data that each class contains. We can see that the class grouping the more data (37.3%) is the tree specie Sugi (s) while the tree specie Hinoki (h) and the class Other (o) (i.e. the different types of land covered) seems to gather the same proportion of data (16.4% and 15.9% respectively). The second class collecting the more data is the class which groups different deciduous tree species, or the Mixed Deciduous broadleaf class (d) (30.4%).

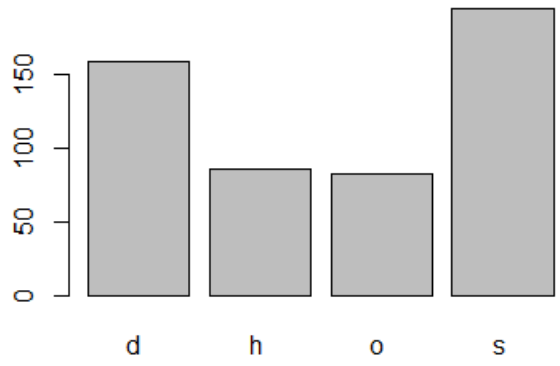

Fig. 1. Number of observations belonging to classes d, h, o, s

With the help of a correlation matrix (Fig. 2 shows 9 spectral bands), we can observe there is a strong evidence of correlation between some variables in the dataset (e.g., $b2$ and $b3$, $b5$ and $b6$, $b8$ and $b9$). It means these variables provide redundant information and some of them could be deleted. However, we use feature selection technique (stepwise logistic regression) to reduce the dimension of feature space. Alternatively, different feature extraction techniques (principal component PCA and LDA) can be used to recode and replace variables with the new ones that contain all relevant information about the dataset. All these techniques are presented and discussed in the parts "Variable selection, transformation and recoding" and "Modelling".

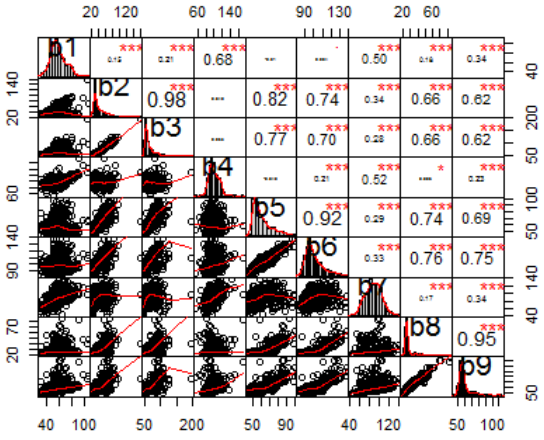

Fig. 2. Correlation matrix of 9 spectral bands

After producing boxplot of the features (Fig. 3), one can observe the data is dispersed. In order to minimize this dispersion, we proceeded to a standardization of the data. This will be presented in the next section.

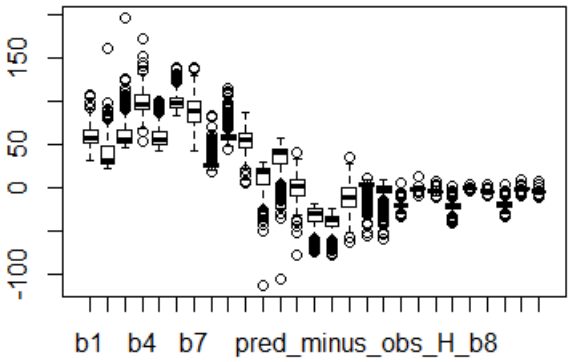

Fig. 3. Boxplot of the features

3 Variable selection, transformation and recoding

In this part, we will discuss different feature extraction/selection techniques that we used for analysis. Before applying any of those techniques, the standardization of the data was performed. It means the data was scaled and centered to have mean zero. The result of this operation is shown in Fig. 4.

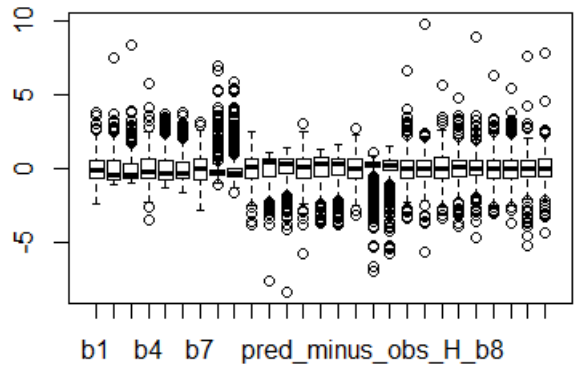

Fig. 4. Scaled and centered data

The first technique we consider is a stepwise logistic regression. It is a method of fitting logistic regression models (see also section "Modelling") in which the choice of predictive variables is carried out by an automatic procedure [1]. In each step, a variable is considered for addition to or subtraction from the set of explanatory variables based on some prespecified criterion. In our case, Akaike information criterion (AIC) was used. A bidirectional elimination of variables was performed that is a combination of forward selection and backward elimination. This took the form of a sequence of AIC values. The final model with reduced number of variables corresponded to the minimum value of AIC. Thus, only 7 variables out of 27 initial variables were selected, and regression had the following final form:

$class \sim b2 + pred\_minus\_obs\_H\_b1 + b8 + pred\_minus\_obs\_S\_b1 + pred\_minus\_obs\_H\_b7 + b9 + b3$

The classification accuracy of stepwise logistic regression and its comparison with that of logistic regression on the full feature space will be given in the part Modelling.

Next, we will consider two feature extraction techniques: LDA and PCA. LDA works when the variables are continuous quantities such as the variables in the datasets we analyze. In LDA, the multivariate Gaussian (or multivariate normal) distribution of the predictors x_1, x_2,..., x_p (p denotes the dimension of feature space) with a class-specific vector and a common covariance matrix is modeled, and then Bayes theorem is used to obtain estimates for a-posteriori probabilities of belonging to class w_i[1]. Before doing LDA, the outliers were identified by Tukey's method [2] which uses interquartile (IQR) range approach (an observation is identified as an outlier if it ranges above and below the $1.5 \cdot IQR$) and then they were made equal to the corresponding mean values (Fig. 5).

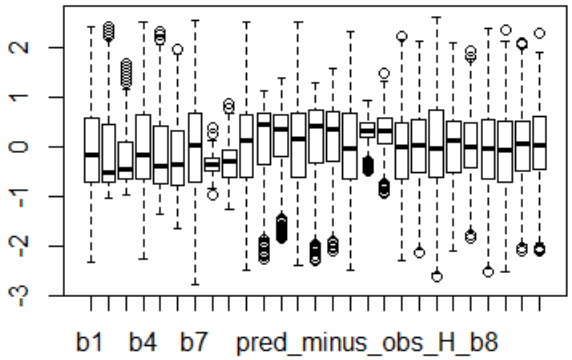

Fig. 5. Data after dealing with outliers

Fig. 6 shows the results of LDA with 3 components that were subsequently used instead of full feature space for different classification models.

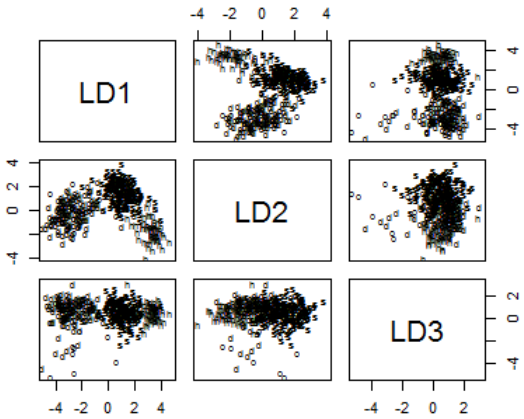

Fig. 6. Linear discriminant analysis

Unlike LDA, PCA is an unsupervised approach that refers to the process by which principal components are computed, and the subsequent use of these components in understanding the data [1]. It finds a low-dimensional representation of a dataset that contains as much as possible of the variation. The idea is that each of the n

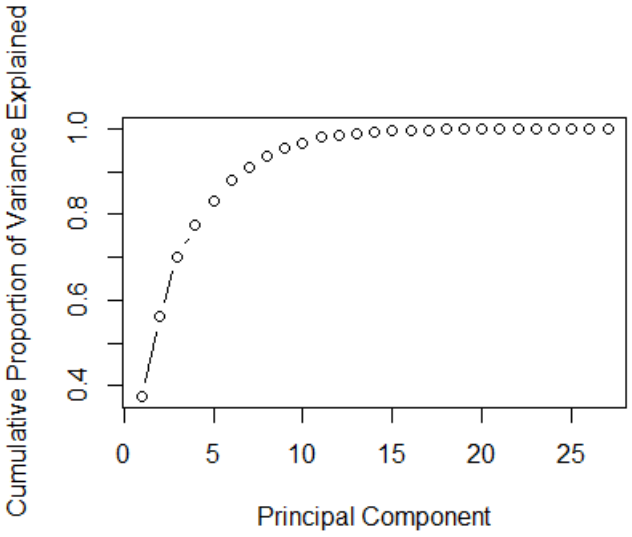

Fig. 7. Cumulative proportion of variance explained

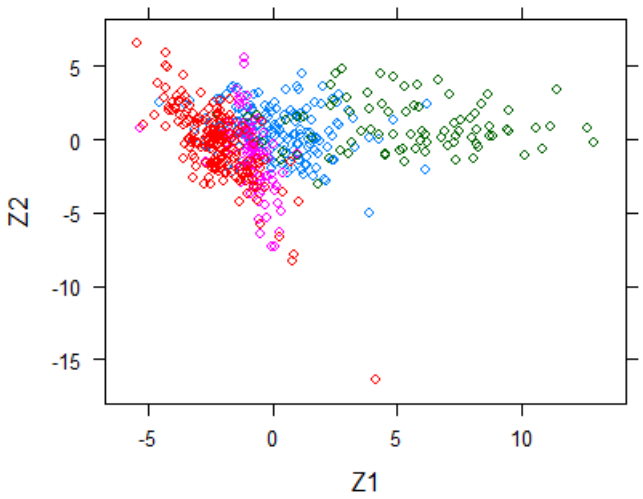

Fig. 8. The first two principal components

observations lives in p-dimensional space, but not all of these dimensions are equally interesting.
The first principal component of a set of features x_1, x_2,..., x_p is the normalized linear combination of the features

$$z_1 = \phi_{11}x_1 + \phi_{21}x_2 + ... + \phi_{p1}x_p$$

that has the largest variance. We refer to the elements ϕ_{11},...,ϕ_{p1} as the loadings of the first principal loading component. By normalized, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
After the first principal component z_1 of the features has been determined, we can find the second principal component z_2 which is the linear combination of x_1,..., x_p that has maximal variance out of all linear combinations that are uncorrelated with z_1.
After identifying all 27 principal components we plotted cumulative proportion of variance explained against principal components (Fig. 7) in order to select only those that explain at least 80% of variance. From Fig. 7, one can conclude that the first 6 principal components are responsible for 87.9% of variance. Therefore, only these components were used for further analysis.
Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization. Once we had computed the principal components, we plotted them against each other in order to produce low-dimensional views of the data. Fig. 8 shows the first z_1 and the second z_2 principal components for our dataset. Four classes are shown in different colors.

4 Modelling

4.1 Logistic regression

In order to directly estimates the a posteriori probabilities of K classes, a logistic regression model is the most appropriate. This model indeed ensures the probabilities sum to one and remain in $[0, 1]$ range [3]. It separates the data by hyperplanes: if the classes are linearly separable, it will separate them and if the classes are not linearly separable, it will find the best linear separation according to the maximum likelihood criterion.

Mathematically it gives:

$$\mathbb{P}(w_K|\mathbf{x}) \approx \hat{y}_K(\mathbf{x}) = \frac{exp(\mathbf{w}_K^T \mathbf{x}')}{\sum_{j=1}^{q} exp(\mathbf{w}_K^T \mathbf{x}')}$$

where $\mathbf{x}' = [1, x_1, x_2, ..., x_p]^T$ is the vector x of p features belonging to one of the K possible classes $w_1, w_2, ..., w_K$. Noted also that the membership values $(\hat{y}_K)$ are distributed according to a multinomial logistic distribution and, therefore,

$$0 \leq \hat{y}_K(\mathbf{x}) \leq 1 \text{ and } \sum_{K=1}^{q} \hat{y}_K(\mathbf{x}) = 1$$

In other words, the model takes the form

$$\log\left(\frac{\mathbb{P}(w_K|\mathbf{x})}{\mathbb{P}(w_{K'}|\mathbf{x})}\right)\approx(\mathbf{w}_K^T-\mathbf{w}_{K'}^T)\mathbf{x}^{'}$$

That is, the log-odds of the posterior probabilities is linear in $\mathbf{x}$.

The error rate is estimated by cross-validation (CV). This approach involves randomly m-fold CV dividing the set of observations into m groups, or folds, of approximately equal size [1]. In our study, we perform 10-fold CV in most of cases. The first fold is treated as a validation set, and the method is fit on the remaining m-1 folds. The ratio E_1 of number of misclassified observations to the total number of observations, is then computed on the observations in the held-out fold. This procedure is repeated m times; each time, a different group of observations is treated as a validation set. This process results in m estimates of the test error, E_1, E_2,..., E_m. The m-fold CV estimate is computed by averaging these values:

$$CV_{(m)}=\frac{1}{m}\sum_{i=1}^{m}E_i$$

The classification accuracy can be calculated as $1-CV_{(m)}$. Correct and misclassified observations can be considered as the elements of confusion matrix. Table 1 shows the confusion matrix for the classification performed by logistic regression for combined training and testing datasets. The classification accuracy is 88.91% (on full feature space).

TABLE 1
Confusion matrix for logistic regression as a classifier for combined training and testing datasets

Logistic regression model		Predicted classes			
		D	H	O	S
True classes	D	144	0	8	7
	H	0	76	1	9
	O	13	0	68	2
	S	6	9	3	177

4.2 k-Nearest neighbors classification

In k-NN classification, a point is classified by a majority vote of its neighbors, which means the point will be assigned to the class most common among its *k* nearest neighbors [4]. In other words, given a query point x_0, we find the k training points $x_{(r)}$, $r = 1,...,k$ closest in distance to x_0, and then classify using majority vote among the k neighbors. In order to compute the distance between two point, Euclidean distance can be used: $d(x,y)=\sqrt{\sum_i(x_i-y_i)^2}$.

For k-NN model we set k equal to 10 by cross-validation. Table 2 shows the confusion matrix for the classification performed by k-NN classifier for combined training and testing datasets. The classification accuracy is 87.95% (on full feature space).

TABLE 2
Confusion matrix for the k-NN classification for combined training and testing datasets

k-NN model		Predicted classes			
		D	H	O	S
True classes	D	138	1	4	16
	H	0	76	0	10
	O	18	0	63	2
	S	2	10	0	183

4.3 Support vector machines

Support vector machines (SVMs) were also used to perform image classification. SVM is a statistical learning algorithm that aims to identify the optimal decision boundary between classes to minimize misclassification. Prior to classification, a kernel is typically applied to the input feature space to increase separability between classes.

In this study, we used SVM with a radial basis function (RBF) kernel for classification because it has achieved higher classification accuracy than other classifiers in previous remote sensing studies [5]. With the SVM-RBF classifier, one can adjust the cost parameter (c) and the kernel spread function (γ) prior to classification. We tested a wide range of paired combinations of c values and γ values, and optimal parameters for each of the SVM classifications were selected by cross-validation. The optimal c value was 100, and the optimal γ value was 0.001.

Table 3 shows the confusion matrix for the classification performed by SVM-RBF classifier for combined training and testing datasets. The classificatoin accuracy is 89.67% (on full feature space).

TABLE 3
Confusion matrix for the SVM-RBF classification for combined training and testing datasets

SVM-RBF model		Predicted classes			
		D	H	O	S
True classes	D	142	2	9	6
	H	0	76	0	10
	O	12	0	69	2
	S	5	8	0	182

We also built SVM-RBF model on the training dataset only in order to compare the classification accuracy with the results of previous study [5]. Table 4 shows the confusion matrix for the classification of observations belonging to the testing dataset. The overall classification accuracy is 85.23% that is close to 85.9% obtained in [5].

TABLE 4
Confusion matrix for the SVM-RBF classification of observations belonging to the testing dataset

SVM-RBF model		Predicted classes			
		D	H	O	S
True classes	D	89	3	7	6
	H	0	34	0	4
	O	10	0	35	1
	S	4	13	0	119

4.4 Decision tree

Decision tree is a non-parametric supervised learning method, which can be used for classification and regression. For our classification problem, this widely used technique allow us to predict the value of our target variable by learning simple decision rules.
The decision tree classifier poses a series of carefully crafted questions about the attributes of the test record [6]. Then, after receiving the answer, a follow-up question is asked until a conclusion about the terminal node is reached. The series of questions and their possible answers can be presented in the form of a hierarchical structure (Fig. 9) consisting of nodes and directed edges tree.

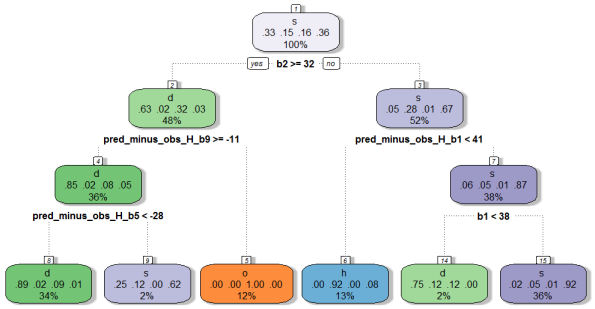

Fig. 9. Decision tree output based on our dataset

Here our algorithm splits the dataset recursively. In this way, the subsets that arise from a split are further split until a predetermined termination criterion is reached. At each step, the split is performed based on the independent variable that results in the largest possible reduction in heterogeneity of the dependent variable. Based on the notion of impurity, which is a measure of heterogeneity of the leaf nodes, splitting rules can be designed in several ways. In our case, the Gini index is used as the impurity quantification method.

Then, in order to evaluate the performance of our model, random samples of the training and testing data had been generated to get an empirical distribution of the accuracy for our decision tree model. In other words, we run 1000 simulations where at each iteration the training/testing datasets were randomly created in the proportion of 70%/30%.

Let acc_i be the model accuracy for the i^{th} iteration. The average accuracy is given by $\mathbb{E}[acc] \approx \sum_{i=1}^{N} acc_i/N$, where N = 1000 in this simulation. According to the law of large numbers, the average of the results converge to the expected value when N tends to infinity. The results of this simulation is shown in Fig. 10. The classification accuracy is 85.69% (on full feature space).

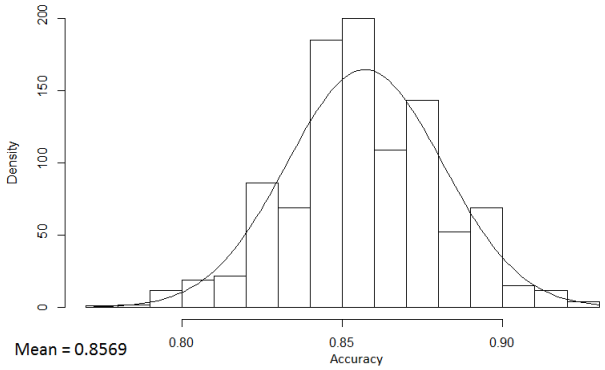

Fig. 10. Decision Tree accuracy distribution

4.4.1 Random forest

The random forest approach starts with a standard decision tree technique but it takes this notion to the next level by creating a large number of decision trees. In ensemble terms, we can consider decision tree as a weak learner and random forest as a strong learner [7].
In this approach illustrated in Fig. 11, every observation is fed into every decision tree [7]. Then, the most common outcome for each observation is calculated and used as the final output.

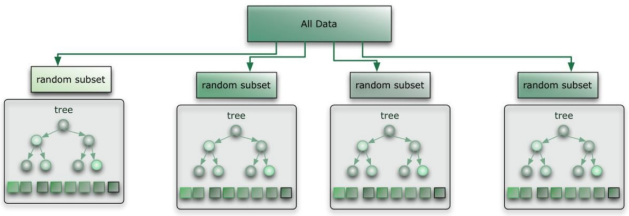

Fig. 11. Random forest model

In this model, there are some sources of randomness in order to make these trees different from one another. Therefore, this approach provides a group of unique trees which all make their classifications differently.

Similarly to the previous section, random samples had been generated in order to find the accuracy distribution for our random forest model (Fig. 12) and therefore to approximate its mean value (88.77%). The classification accuracy calculated via 10-fold cross-validation revealed apprximately the same result (88.55%).

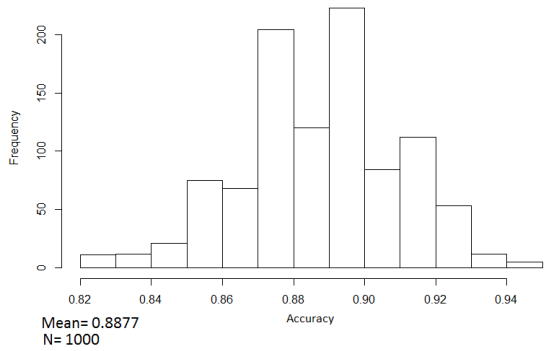

Fig. 12. Random forest classification accuracy distribution

4.5 Artificial neural networks

An artificial neural network (ANN) can learn patterns in data by mimicking the structure and learning process of neurons in the brain. This machine learning model is commonly used for classification in data science. One particularity of this model is that it can be difficult to design perfectly a network since there are many parameters that can affect the output. Moreover, there is no general design methodology for picking the best values for the latter.

Let us take a closer look at the standard ANN architecture which is shown in Fig. 13. It has three layers: an input layer, an output layer with one neuron per class, and a hidden layer of neurons [8]. The neurons have an identical structure and its inputs are summed to give a single value x.

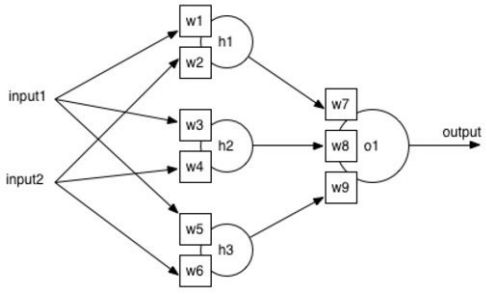

Fig. 13. Neural Network with 2 inputs, 3 hidden neurons and 1 output

Then, the output of the neuron is the output of the defined function, $f(x)$ as described in Fig. 14. A lot of functions can be used in the artificial neuron. In our case, the package nnet used in R gave us the possibility to construct standard ANNs with one

hidden layer of logistic function, which is the most common function used for ANN.

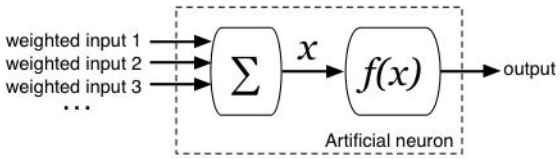

Fig. 14. Artificial Neuron

Regarding the modelling part, we optimized the model by adjusting three main parameters. The first one allows us to change the maximum number of iterations before training halts. In this model, we increased it from 100 (=default value) to 250. It can be beneficial when we have a larger network than may be expected, and therefore it can take more iterations to settle a good state. The second one allows us to add a small decay to the weights. By doing so, they decrease over time unless strengthen by new data.
Finally, we changed the number of hidden neurons to use. Although there are some rules to adjust this parameter such as 1/30 of the training samples, it remains difficult to estimate it. In our case, in order to be accurate when tuning this parameter, every number between 3 and 25 was tried. We kept 16 for this parameter.

Due to the randomness character of this model, similarly to what has been shown in the previous section, a Monte Carlo simulation was used in order to approximate the distribution of the accuracy given the defined parameters. As explained before, we tend towards the true mean when the number of simulation increases.

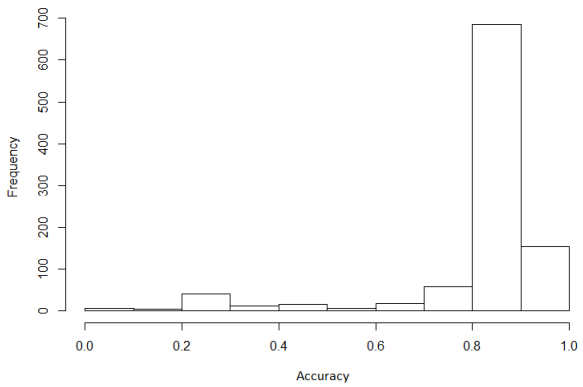

Fig. 15. Accuracy distribution of the ANN non scaled-centered model, number of simulations = 1000

It is seen from the simulation (Fig. 15) that this model can be sporadic. This is why after adjusting the main settings, there is still a point to consider. By analyzing our initial database (before scaling and centering), we can notice that some variables have very different ranges. Therefore, all data were scaled to fall within the range $[-1, 1]$. Thanks to this standardization, all the inputs were settled to a comparable range. Fig. 16 shows that this approach provides better and less volatile results.

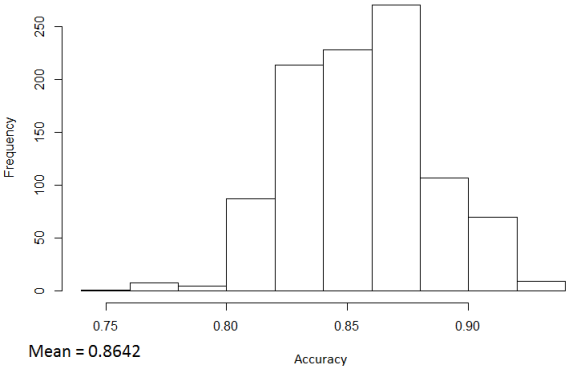

Fig. 16. Accuracy distribution of the ANN scaled-centered model, number of simulations = 1000

The comparison of classification accuracy for different models is given in Table 5 (after LDA), Table 6 (after PCA), Table 7 (on full feature space).

TABLE 5
Comparison of classification accuracy after LDA

Model	Accuracy, %
k-NN	95.8
SVM-RBF	96.97
Random forest	96.95

TABLE 6
Comparison of classification accuracy after PCA

Model	Accuracy, %
Logistic regression	74.42
k-NN	74.4
SVM-RBF	80.9
Random forest	78.79

TABLE 7
Comparison of classification accuracy on full feature space

Model	Accuracy, %
Logistic regression	88.91
k-NN	87.95
SVM-RBF	89.67
Decision tree	85.69
Random forest	88.77
ANN	86.42

Note that stepwise logistic regression (with 7 variables selected) revealed the classification accuracy of 94.47% that outperformed all classification models on full feature space. The classification accuracy after doing PCA was much lower compared to other feature selection/extraction techniques that might be explained by unsupervised nature of the PCA algorithm. In contrast, LDA, which is a supervised method, provided a parsimonious model (only 3 components for each observation) that in combination with SVM-RBF led to the highest classification accuracy (96.97%) among all studied models.

5 Scenario with cost

In order to study how the cost of classification error can affect the solution, we created a two-classes scenario in which the relative cost of each classification error was different. The two classes were defined in the following way: initial classes d, h, o were merged into one class "0" and "s" was labeled as class "1". The cost c_{12} of misleading classification of class "0" as class "1" was equal to 1 whereas the corresponding cost c_{21} of labeling class "1" as class "0" was equal to 15. Correct classifications had zero costs ($c_{11} = c_{22} = 0$). Thus, a cost matrix had the following form:

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 15 & 0 \end{bmatrix}$$

Next, the optimal classification rule (the a-posteriori classification threshold θ above which we categorized the observation as belonging to class 1) was deduced theoretically by combining the following conditions:

$$\begin{cases} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}^T \times \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix}^T \\ p_0 = 1 - p_1 \\ c_{11} = c_{22} = 0 \\ l_1 > l_2 \end{cases}$$

Here, p_0 and p_1 denote the a-posteriori probabilities of belonging to class "0" and "1" respectively; l_1 and l_2 are the expected costs if we decide a given observation belongs to class "0" and class "1" respectively. For minimizing the expected cost, we have to make sure l_1 is higher then l_2 if we decide class "1" (the last condition). We found out that theoretical threshold θ_{th} depends on the non-diagonal elements of the cost matrix and could be expressed as follows:

$$\theta_{th} = \frac{c_{12}}{c_{21}+c_{12}} = 1/16$$

Then, the optimal classification rule with costs ($p_1 > \theta_{th}$) was verified empirically on the data. We showed that the total cost c_{total} obtained on the entire population was lower ($c_{total} = 101$) than if we relied on a standard "Bayesian" classifier ($c_{total} = 258$), without integrating the costs. The a-posteriori probabilities were determined by running logistic regression.
By modifying the decision threshold θ we found out that actual minimal total cost was around the optimal theoretical threshold θ_{th} (see Fig. 17).

6 Study of the "reject option"

The reject option is to be used for those observations for which the conditional class probabilities are close and as such are hard to classify. We defined the reject option λ so that when making decision was difficult, the observation would be ignored. The cost matrix

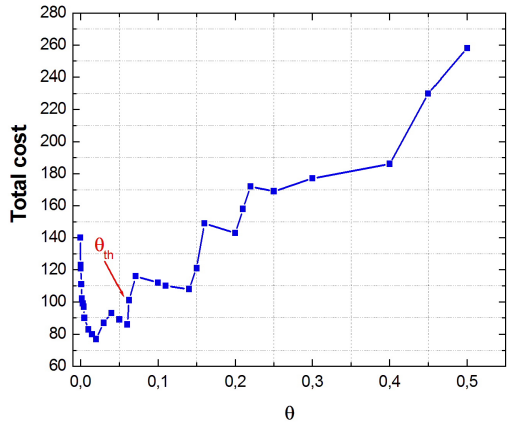

Fig. 17. Total cost depending on θ

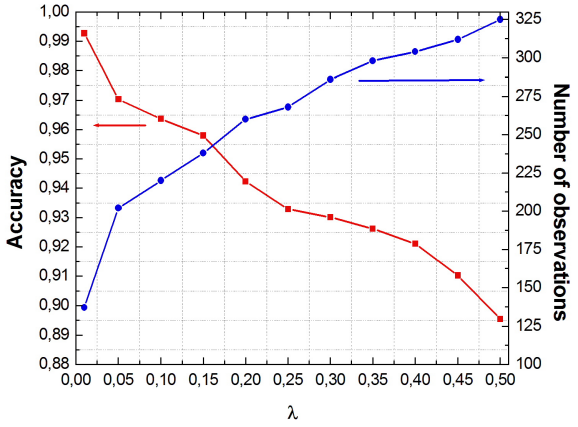

Fig. 18. Accuracy of the classifier and number of observations depending on λ

had the following form:

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Fig. 18 shows how the accuracy of the classifier and number of observations evolve with the change in λ. One can observe that the accuracy increases with decreasing λ. This can be explained by a decrease in the number of observations: λ decrease results in exclusion of a larger number of too risky observations from consideration.

7 Conclusion

In this study, different classification models including logistic regression, k-NN, SVM-RBF, decision tree, random forest, ANNs as well as different feature selection/extraction techniques were studied to find out those that would demonstrate high classification accuracy. Based on the results for "Forest type mapping" dataset, it may be useful to perform LDA since it creates a parsimonious model (with only 3 components instead of 27 features). Together with SVM-RBF model it revealed the classification accuracy of 96.97% that is the highest result among all studied models. Alternatively, stepwise logistic regression (with 7 features selected) that demonstrated the accuracy of 94.47% can be used. The classification accuracy after doing PCA was much lower compared to other feature selection/extraction techniques that might be explained by unsupervised nature of the PCA algorithm.

Furthermore, we studied how cost of classification error could affect the solution. The optimal classification rule was deduced theoretically and verified empirically. We found out that theoretical threshold θ_{th} avobe which we categorized the observation as belonging to positive class depends on the non-diagonal elements of the cost matrix.

References

[1] James, G., Witten, D., Haste, T., & Tibshirani, R. (2015). An Introduction to Statistical Learning, 6th ed. Springer.

[2] Dhana, K. (2016, April 30). Identify, describe, plot, and remove the outliers from the dataset [Web log post]. Retrieved April 24, 2017, from https://www.r-bloggers.com

[3] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference and Prediction 2nd ed. Springer.

[4] Barber, D. (2012). Bayesian Reasoning and Machine Learning, 1st ed. Cambridge University Press.

[5] Johnson, B., Tateishi, R., & Xie, Z. (2012). Using geographically weighted variables for image classification. *Remote Sensing Letters*, 3(6), 491-499

[6] Han, J., Kamber, M., & Pei, J. (2012). Classification: basic concepts. *Data mining Concepts and techniques, Amsterdam: Elsevier.*

[7] Benyamin, D. (2012, November 9). A Gentle Introduction to Random Forests, Ensembles, and Performance Metrics in a Commercial System [Web log post]. Retrieved May 2, 2017, from https://datascienceplus.com

[8] Catterson, V. (2014, January 12). Understanding data science: classification with neural networks in R [Web log post]. Retrieved April 24, 2017, from http://cowlet.org