Machine Learning project report

Francesco Casciola, Nicola Landolfi

July 2019

Contents

Chapter 1

Problem introduction

Through this project we intend to become familiar with imbalanced multi-class supervised classification problems. To serve this purpose, we decided to explore the Forest Cover type dataset in the UCI Machine Learning Repository. The dataset comprises observations taken from 30m by 30m patches of the Roosevelt National Forest (in northern Colorado) that are labelled upon the main cover type of that patch.

1.1 Dataset overview

The training set counts 15 120 observations while the test set 565 892. The data comes from the US Geological Survey (USGS) and the US Forest Service (USFS). For each patch the following 12 variables (and their units of measurement) plus the labels are provided:

1. Elevation (m),
2. Aspect (azimuth from true north),
3. Slope (°),
4. Horizontal distance to nearest surface water feature (m),
5. Vertical distance to nearest surface water feature (m),

6. Horizontal distance to nearest roadway (m),
7. Hillshade 9am: a relative measure of incident sunlight at 09:00 h on the summer solstice (index),
8. Hillshade Noon: a relative measure of incident sunlight at noon on the summer solstice (index),
9. Hillshade 3pm: a relative measure of incident sunlight at 15:00 h on the summer solstice (index),
10. Horizontal distance to nearest historic wildfire ignition point (m),
11. Wilderness area: the macro-area the patch belongs to (four binary values, one for each wilderness area),
12. Soil type: the principal soil type in the patch (40 binary values, one for each soil type),
13. Cover type: forest cover type (classes from 1 to 7, one for each patch).

Table 1.1 shows that the dataset is imbalanced. For an effective classifier all classes should be equally represented in the training set, thus it is necessary to reduce its size. According to [1] the training set (which includes the validation data) should follow the approach of Figure 1.1. The relative sizes of the training and testing sets make the classification a challenging problem since the training set is $\sim 2.6\%$ of the overall dataset.

1.2 Feature engineering

It is important to notice that the analysis, from now on, has been done on the training set only (to avoid data leakage [2]). The total number of features is 54, thus, simple feature selection is done: for example, Figure 1.2 shows the elevation feature grouped by each cover type and it is quite clear that classes 4 (Cottonwood/Willow), 5 (Aspen) and 7 (Krummholz) are easily separable; focusing on the numerical features, Figure 1.3 shows the

Cover type	Occurrences
1	211 840
2	283 301
3	35 754
4	2 747
5	9 493
6	17 367
7	20 510
Total	581 012

Table 1.1: Number of observations within each forest cover type class

Forest cover type class	Training data set observations	Validation data set observations	Test data set observations	Total observations per cover type
Spruce/fir	1620	540	209 680	211 840
Lodgepole pine	1620	540	281 141	283 301
Ponderosa pine	1620	540	33 594	35 754
Cottonwood/ willow	1620	540	587	2747
Aspen	1620	540	7333	9493
Douglas-fir	1620	540	15 207	17 367
Krummholz	1620	540	18 350	20 510
Total observations per data set	11 340	3780	565 892	581 012

Figure 1.1: Number of observations within each forest cover type class for each dataset. The validation set is 25% the size of the training set.

correlation plots for each of them and we noticed that the couples «Vertical_Distance_To_Hydrology - Horizontal_Distance_To_Hydrology» and «Hillshade_Noon - Hillshade_3pm» are both highly correlated (positive correlation coefficients are 0.6521 and 0.6145, respectively). Furthermore, the couple «Hillshade_9am - Hillshade_3pm» has a negative correlation coefficient equal to -0.7800. From these observations, two new features could substitute the ones used for their computation allowing us to reduce the dimension of the feature space:

- Distance_To_Hydrology: the Euclidean distance computed on Vertical_Distance_To_Hydrology and Horizontal_Distance_To_Hydrology,
- Mean_Hillshade: the mean value of the three Hillshade features.

Moreover, Hillshade_3pm missing values (88 zeros) have been imputed with the median over all the training samples and one-hot encoding has been applied to the target variable yielding 7 binary variables. Finally, the numerical features were scaled to zero mean and unit variance (standardization [4]).

1.2.1 Circular quantities scaling

Scaling «Aspect» and «Slope» features via standardization can't be done in the usual way [3]: being circular quantities it's necessary to recur to *directional statistics* theory to compute the mean value and standard deviation of the two. Given $x \in (-\pi, \pi]$ we obtain

$$\begin{aligned} \mu &= \operatorname{atan2}\left(\underbrace{\frac{1}{N}\sum_{j=1}^{N} \sin x}_{\mathcal{S}}, \underbrace{\frac{1}{N}\sum_{j=1}^{N} \cos x}_{\mathcal{C}} \right), \\ \sigma &= \sqrt{-2\ln\left(\sqrt{\mathcal{C}^2 + \mathcal{S}^2}\right)}, \end{aligned} \tag{1.1}$$

where μ is the mean value of the set and σ is the standard deviation.

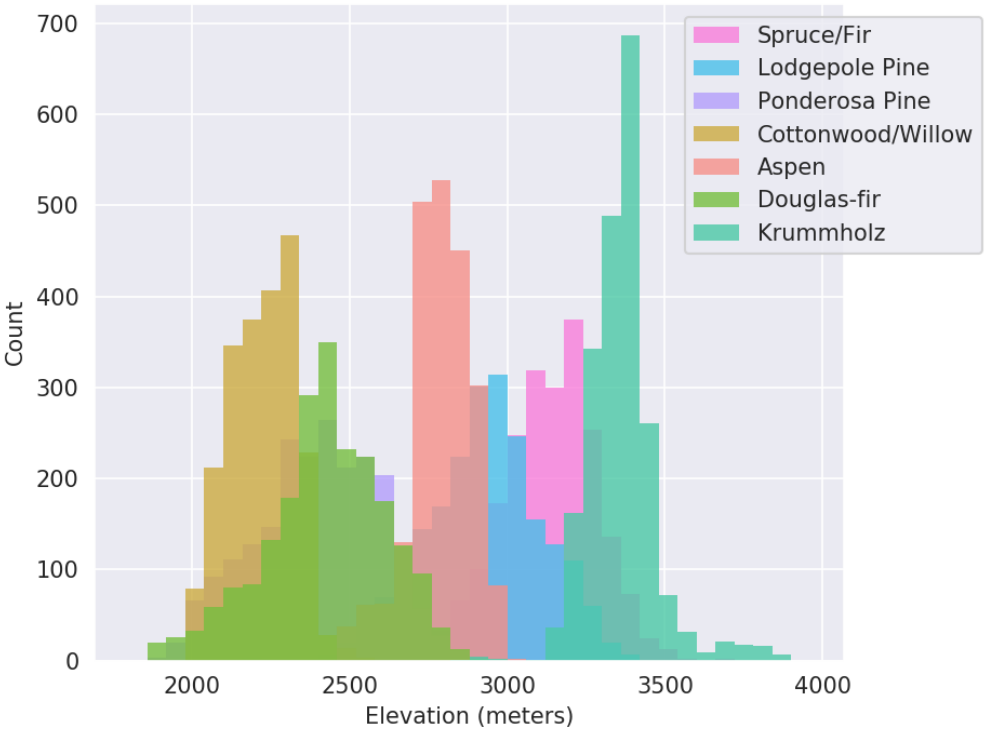

Figure 1.2: Elevation histogram for each cover type. At a first glance classes 4 (Cottonwood/Willow), 5 (Aspen) and 7 (Krummholz) are easily separable.

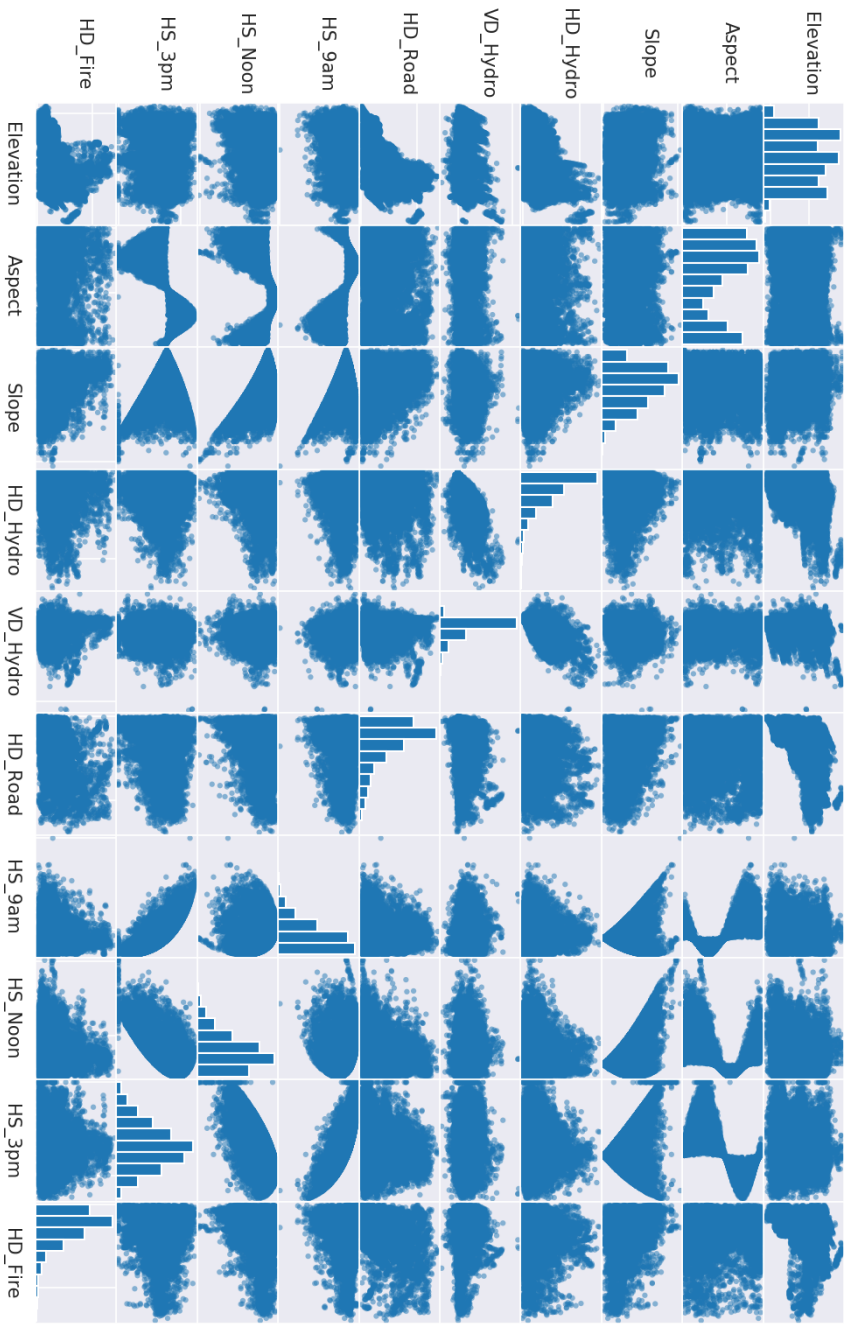

Figure 1.3: Correlation matrix of the numerical features.

Chapter 2

The models

In this chapter we discuss several neural network classifiers and their results on the Forest Cover Type dataset.

2.1 Baseline model

We decided to start with an adaptation of the *optimal* model's architecture outlined in [1] and reported in Figure 2.1. The model is made of 51 input nodes, 120 hidden nodes and seven output nodes (symbolized as baseline-51-120-7) where each layer is fully connected. Both the hidden and output layers utilized the logistic (sigmoid) activation function:

$$\sigma(a) = \frac{1}{1 + e^{-a}}.$$

The 51 input features are obtained from the 54 initial ones by using «Distance_To_Hydrology» feature as described in Section 1.2 and by dropping «Aspect» and «Slope» features since the Hillshade ones, according to their definition, vary depending on a factor

$$\begin{aligned} \alpha = &\cos(Slope) \cdot \cos(90 - Altitude) + \\ &+ \sin(Slope) \cdot \sin(90 - Altitude) \cdot \cos(Azimuth - Aspect). \end{aligned} \tag{2.1}$$

Given the training set size, we picked SGD (Stochastic Gradient Descent) as loss function optimizer, while the loss function is the classic MSE (Mean Squared Error, as in [1]):

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Y_i^*)^2 \tag{2.2}$$

where Y_i is the vector of the true values of the target, Y_i^* is the model's prediction and N is the number of samples. After running a grid-search algorithm over a set of hyperparameters, the best model was found (learning rate (LR) to 0.05, batch size to 128 and number of epochs to 200): the classifier has then been trained via a 10-fold cross-validation. The accuracy on the training set and validation set is shown in Figure 2.2 while the losses in Figure 2.3: to smooth the plots, we plot the metric value at the last epoch of each fold. Given the multi-class nature of the classification task, Figure 2.4 represent the heat-map rendering of the classification matrix to better convey the information regarding the correctly classified/misclassified samples: it is easy to notice that 532 samples, of the total 587 in the test set, belonging to the minority class (Cottonwood/Willow) are correctly classified; 5 285 are misclassified as the minority class but they belong to Ponderosa Pine. Moreover, a good amount of Spruce/Fir samples are correctly classified (this should be expected since this class is one of the majority classes). Other useful performance metrics for imbalanced class problems are precision (PRE), recall (REC) and F1-score (F1):

$$\begin{aligned} \text{PRE} &= \frac{TP}{TP + FP}, \\ \text{REC} &= \frac{TP}{FN + TP}, \\ \text{F1} &= 2 \cdot \frac{\text{PRE} \cdot \text{REC}}{\text{PRE} + \text{REC}}, \end{aligned} \tag{2.3}$$

where TP stands for True Positive, FP for False Positive and FN for False Negative. Table 2.1 summarises each metric for all 7 classes: as expected,

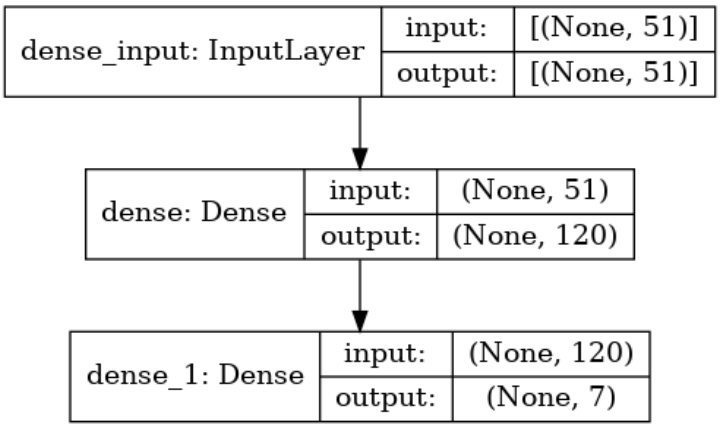

Figure 2.1: Baseline model diagram.

the first minority class Cottonwood/Willow exhibits high recall ($\sim 90.63\%$) and very low precision ($\sim 6.95\%$) which means that the classifier is giving a lot of positive predictions while being wrong most of the times (the classifier confuses quite often samples belonging to other classes as samples belonging to this class); the second minority class Aspen is pretty much the same, high recall ($\sim 71.96\%$) and low precision ($\sim 6.63\%$). For the majority class Spruce/Fir the classification looks quite reasonable (recall $\sim 64.75\%$, precision $\sim 66.37\%$), mainly because the class represents a huge part of the dataset. In fact, during the validation phase, the model having highest accuracy is picked and this will necessarily involve a high recall value for the aforementioned class. Finally, the weighted avg column represent a weighted average of each precision metric where the weights are the support value for each class. For example, the weighted avg of the precision column is calculated as Eq. 2.4:

$$\begin{aligned} &\frac{(209679)0.6637 + (281141)0.7898 + (33594)0.6497 + (587)0.0695}{565892} + \\ &+ \frac{(7333)0.0663 + (15207)0.2611 + (18350)0.3569}{565892} \approx 0.6964. \end{aligned} \tag{2.4}$$

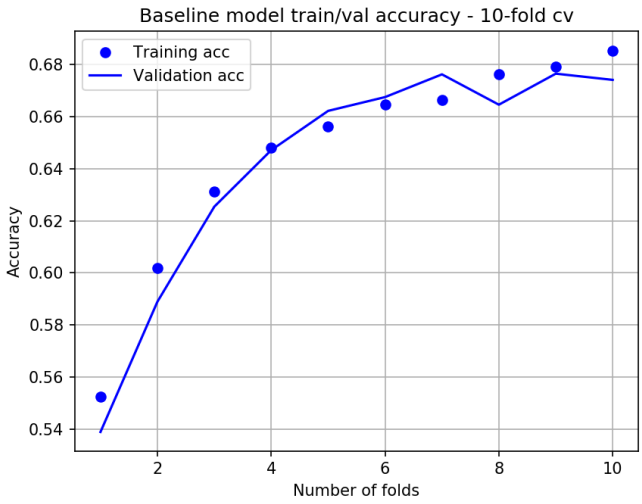

Figure 2.2: Baseline model accuracy.

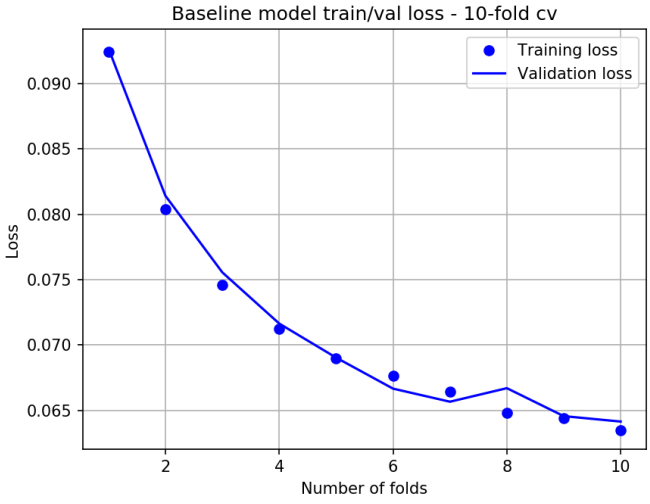

Figure 2.3: Baseline model losses.

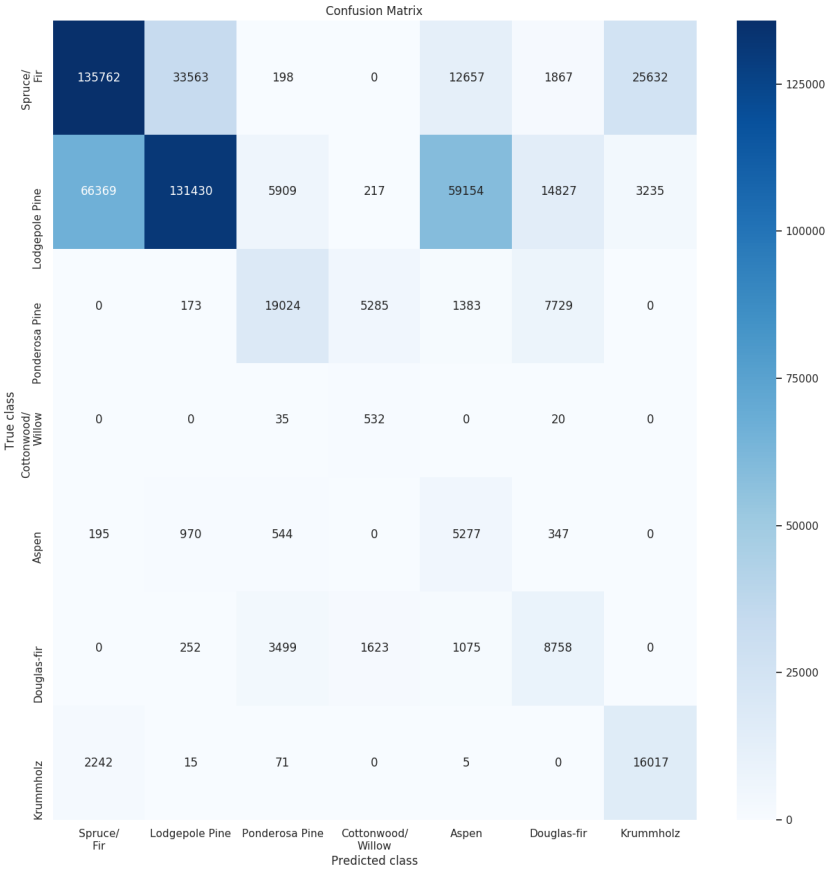

Figure 2.4: Heat-map rendering of the confusion matrix for the baseline model.

	precision	recall	f1-score	support
Spruce/Fir	0.6637	0.6475	0.6555	209680
Lodgepole Pine	0.7898	0.4675	0.5873	281141
Ponderosa Pine	0.6497	0.5663	0.6051	33594
Cottonwood/Willow	0.0695	0.9063	0.1291	587
Aspen	0.0663	0.7196	0.1215	7333
Douglas-fir	0.2611	0.5759	0.3593	15207
Krummholz	0.3569	0.8729	0.5066	18350
weighted avg	0.6964	0.5598	0.5984	565892
test set accuracy	0.5598	0.5598	0.5598	0.5598

Table 2.1: Precision, recall, f1-score summary table for the baseline model. Support indicates the number of occurrences of each particular class in the true responses (for the test set). Weighted avg is the per metric weighted average where the weights correspond to the support of that class.

2.2 Proposed models

Inspired by what we have seen so far, we tried to come up with possible good classifiers while trying to *keep it simple*. Section 2.2.1 goes through a model similar to the baseline of Section 2.1. Section 2.2.2 present a completely different model based on our intuition of neural networks.

2.2.1 One hidden layer network

The model is made of 51 input nodes, 120 hidden nodes and seven output nodes (symbolized as onehidden-51-120-7) where each layer is fully connected (same architecture as Figure 2.1). The hidden layer activation function is the ReLU:

$$\text{ReLU}(a) = (a)_+ = a \cdot [a > 0]. \tag{2.5}$$

The output layer activation function is the softmax:

$$\text{Softmax}(a)_j = \frac{\exp(a_j)}{\sum_{c=1}^{C} \exp(a_c)},\ j = 1, 2, \ldots, C. \tag{2.6}$$

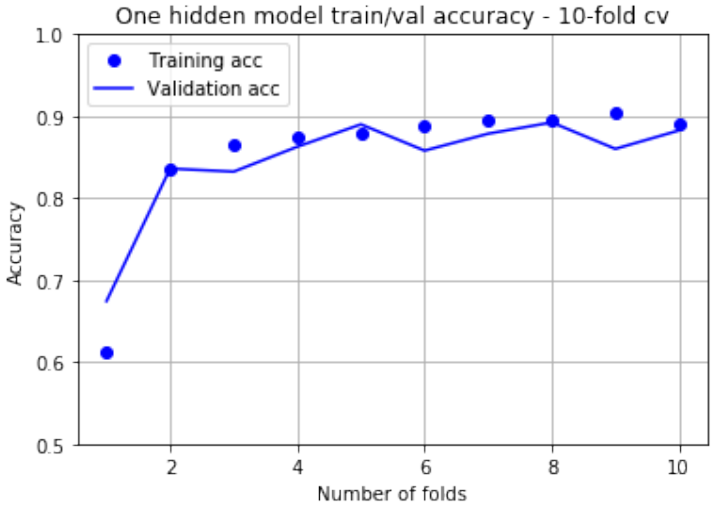

Figure 2.5: One hidden model accuracy.

where C is the number of classes ($C = 7$ for our classification task). The loss function optimizer is again SGD and the loss function is the Cross Entropy:

$$\text{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}[y_i = c]\log p_{model}(y_i = c) \tag{2.7}$$

where N is the number of samples, C the number of classes, $p_{model}(y_i = c)$ the probability predicted by the model for the i-th sample of belonging to the c-th class. After running a grid-search algorithm over a set of hyperparameters, the best model was found (learning rate to 0.5, batch size to 128 and number of epochs to 100): the classifier has then been trained via a 10-fold cross-validation. The accuracy on the training set and the validation set is shown in Figure 2.5 while the losses in Figure 2.6. It is easy to notice a slight improvement compared to the baseline. Figure 2.7 and Table 2.2 confirm that the learning process has improved: only 32 over 587 samples of Cottonwood/Willow are misclassified; 15 774 samples are classified as Aspen but they belong to Lodgepole Pine which, compared to the baseline, is an improvement of ~ 3.8 times (namely, 15 774 against 59 154). It is quite clear that this model trains faster than the baseline and performs way better.

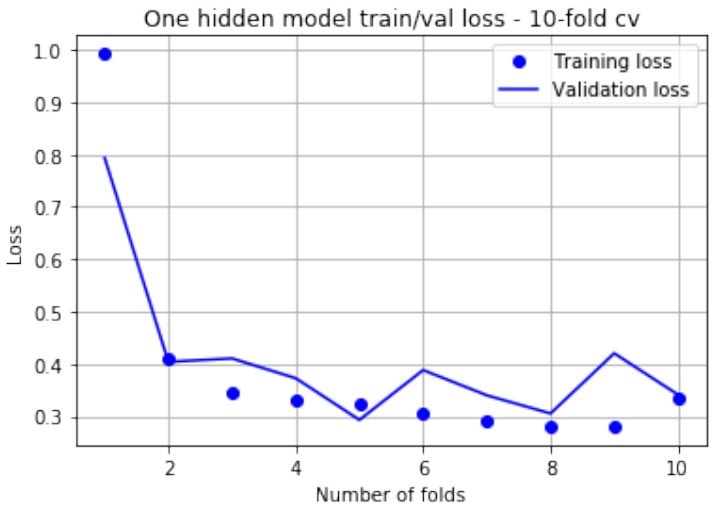

Figure 2.6: One hidden model losses.

	precision	recall	f1-score	support
Spruce/Fir	0.7569	0.7092	0.7323	209680
Lodgepole Pine	0.8110	0.7207	0.7632	281141
Ponderosa Pine	0.7813	0.7343	0.7571	33594
Cottonwood/Willow	0.2498	0.9455	0.3952	587
Aspen	0.2582	0.8805	0.3993	7333
Douglas-fir	0.4571	0.8973	0.6057	15207
Krummholz	0.5728	0.9663	0.7192	18350
weighted avg	0.7642	0.7323	0.7406	565892
test set accuracy	0.7323	0.7323	0.7323	0.7323

Table 2.2: Precision, recall, f1-score summary table for the one hidden layer model.

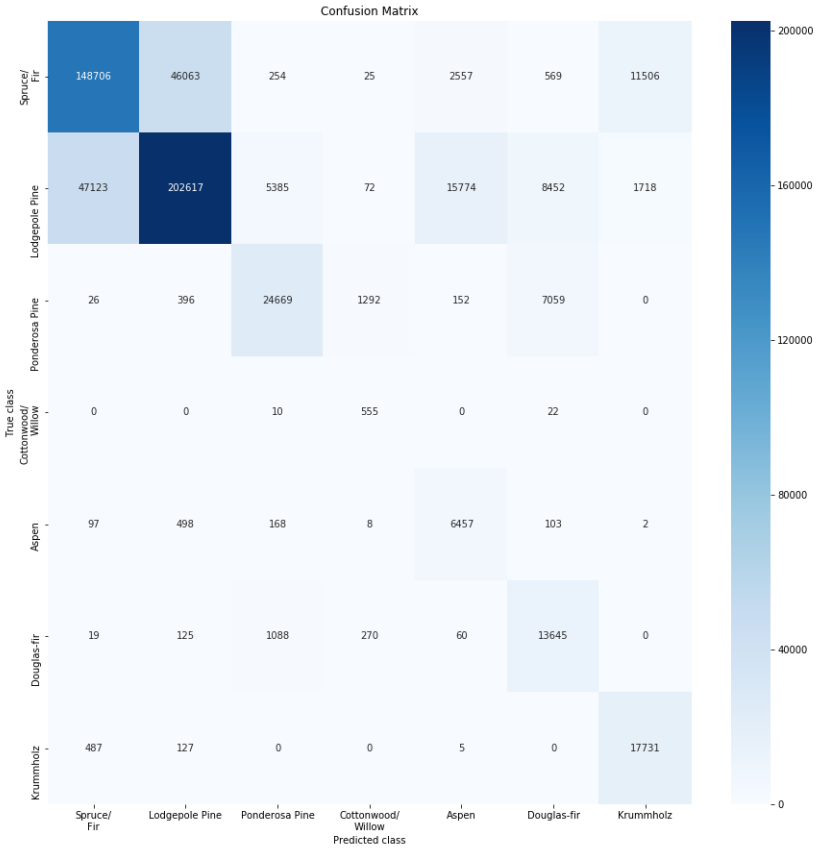

Figure 2.7: Heat-map rendering of the confusion matrix for the one hidden layer model.

2.2.2 Two hidden layers network

Based on our experience with the Forest Cover Type dataset we came up with a different architecture shown in Figure 2.8. This model consists of 51 input nodes, 2 hidden layers (the first one made of 30 nodes while the second one of 15 nodes) and seven nodes for the output layer; each layer is fully connected, as usual. Both hidden layer activation functions are ReLU and the output layer activation function is, again, the softmax. The loss function optimizer and the loss function are equal to the one hidden layer model. After running a grid-search algorithm over a set of hyperparameters, the best model was found (learning rate to 0.3, batch size to 128 and number of epochs to 100): the classifier has then been trained via a 10-fold cross-validation. The accuracy on the training set and the validation set is shown in Figure 2.9 while the losses in Figure 2.10. Even though the network performs slightly worse than onehidden-51-120-7, it has 2085 edges (vs 6960) thus the computational burden is less heavy (*Parsimony principle*).

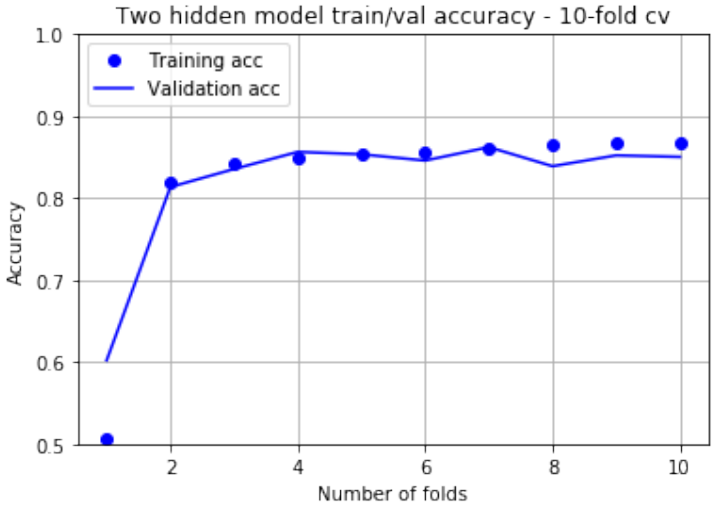

Figure 2.9: Two hidden layers model accuracy.

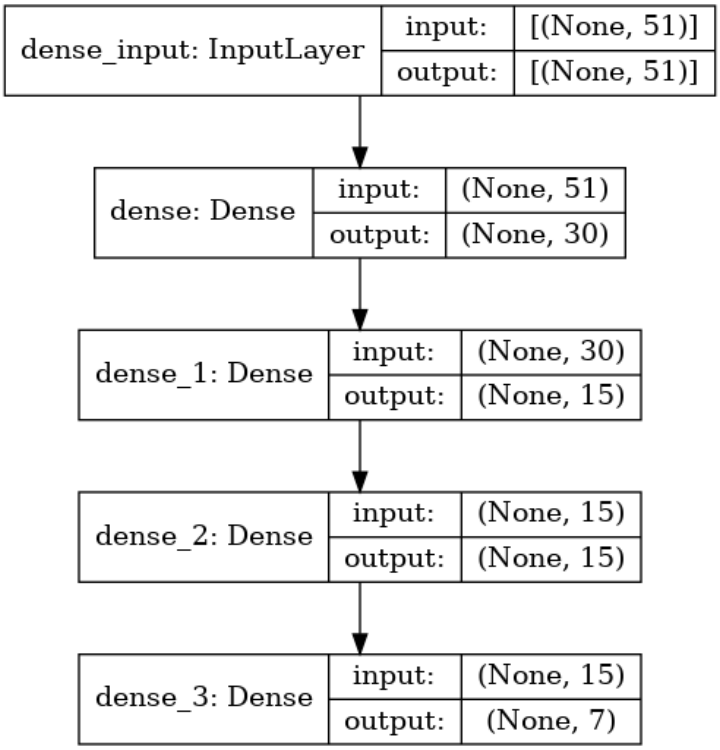

Figure 2.8: Two hidden layers model diagram.

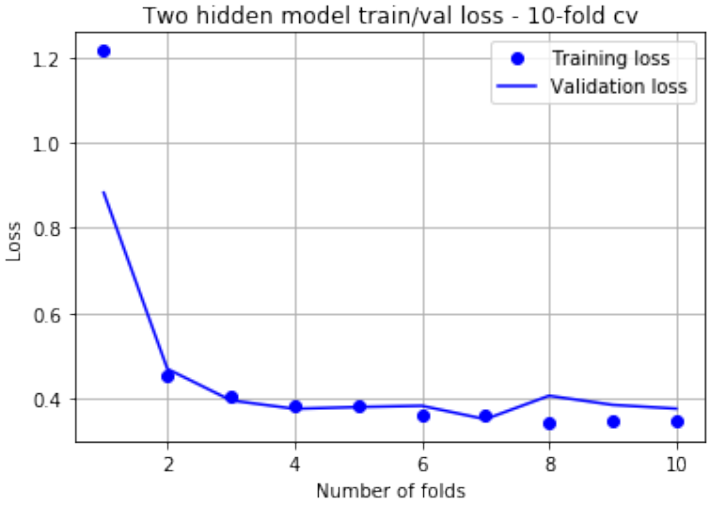

Figure 2.10: Two hidden layers model losses.

	precision	recall	f1-score	support
Spruce/Fir	0.6801	0.7700	0.7223	209680
Lodgepole Pine	0.8374	0.6226	0.7142	281141
Ponderosa Pine	0.7666	0.7118	0.7382	33594
Cottonwood/Willow	0.1316	0.9813	0.2321	587
Aspen	0.2351	0.8773	0.3708	7333
Douglas-fir	0.5062	0.8131	0.6240	15207
Krummholz	0.5418	0.9478	0.6894	18350
weighted avg	0.7479	0.7019	0.7104	565892
test set accuracy	0.7019	0.7019	0.7019	0.7019

Table 2.3: Precision, recall, f1-score summary table for the two hidden layers model.

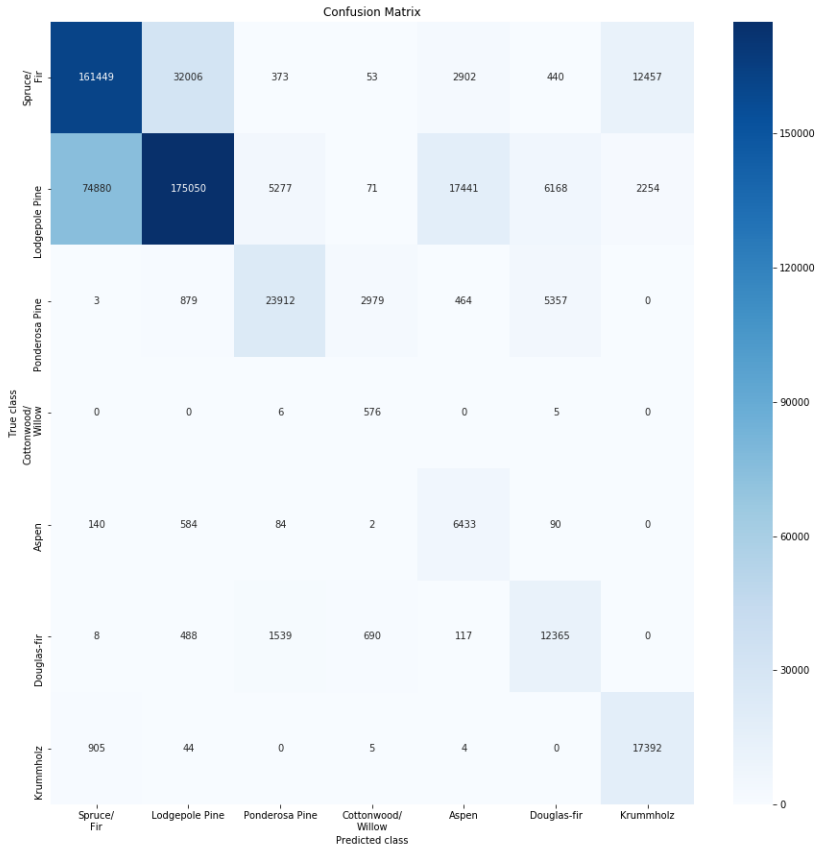

Figure 2.11: Heat-map rendering of the confusion matrix for the two hidden layers model.

2.2.3 Varying the previous models

Since removing redundant features allowed the machine to perform better, we decided to accomplish feature selection in a more automatized way. We modified the models seen in Section 2.2.1 and 2.2.2 by adding two hidden layers before the first hidden layer of each of the previous models. The first of two new hidden layers has $\nu < 51$ units, while the second has 51 inspired by a generic auto-encoder architecture in Figure 2.12.
Auto-encoders is a class of neural networks generally trained using as labels the training set itself, with the hope that the machine will learn an underlying set of rules that allows it to create a simplified representation of the data. The hidden layers bottleneck of an auto-encoder allows the data to be represented in a low dimensionality space that can then be used to reconstruct the input data after the bottleneck. The reason why we tried these schemes is that we hope to get a similar effect, except for the fact that we aren't interested in reconstructing the input data, but we want to use the bottleneck for an automated feature reduction. Running a grid-search algorithm over a set of possible ν values we picked the candidate substitutes for the previous networks. Moreover, we also noticed that, when using these networks, the loss in training and validation phase reaches rapidly a good value and then starts oscillating around it. Thus, we tried to change the split ratio between training and validation set for the 10-fold cross validation algorithm to 60%/40% to avoid overfitting. Ultimately two good networks were found, symbolised as: onehidden_red-51-49-51-120-7 and twohidden_red-51-46-51-30-15-7.
During the training of onehidden_red-51-49-51-120-7 the same learning rate as for onehidden-51-120-7 was used. While for twohidden_red-51-46-51-30-15-7 the learning rate is 0.5 and this network performs better ($\sim +3.4\%$ of accuracy) than the twohidden-51-30-15-7 seen in Section 2.2.2.

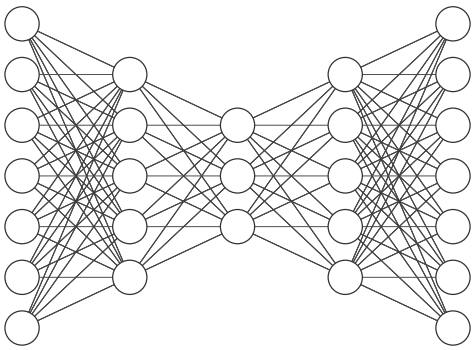

Figure 2.12: Generic auto-encoder layout

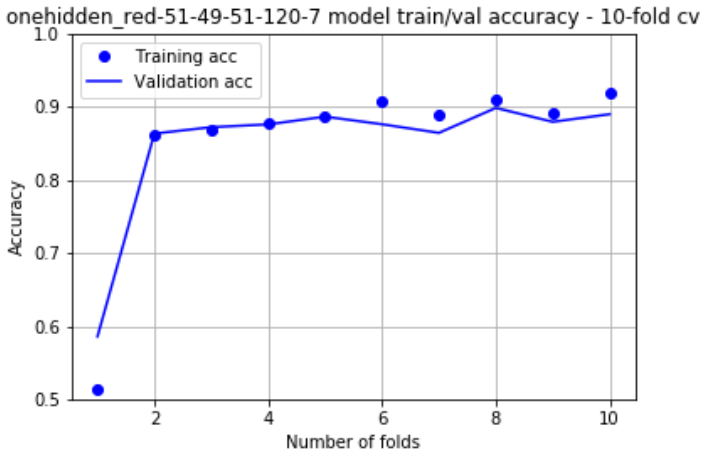

Figure 2.13: onehidden_red-51-49-51-120-7 model accuracy.

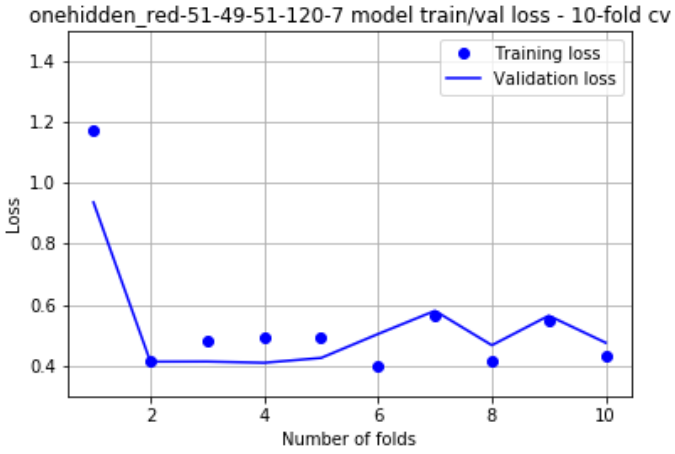

Figure 2.14: onehidden_red-51-49-51-120-7 model losses.

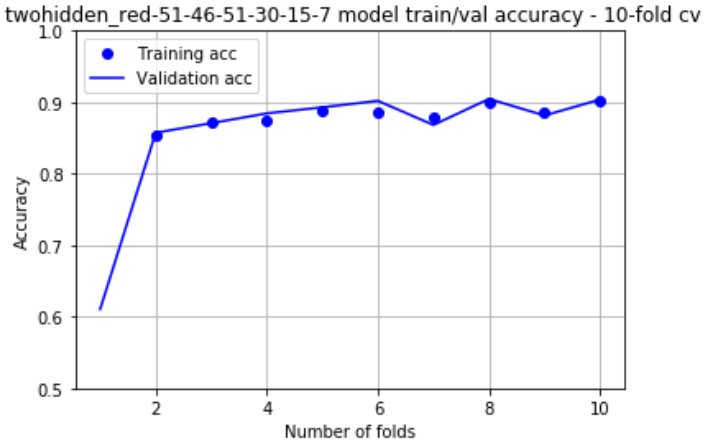

Figure 2.15: twohidden_red-51-46-51-30-15-7 model accuracy.

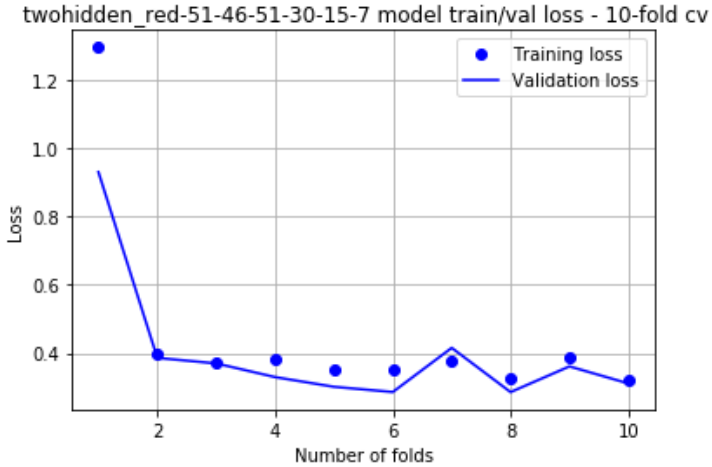

Figure 2.16: twohidden_red-51-46-51-30-15-7 model losses.

	precision	recall	f1-score	support
Spruce/Fir	0.7503	0.7708	0.7604	209680
Lodgepole Pine	0.8482	0.6837	0.7571	281141
Ponderosa Pine	0.7880	0.8498	0.8177	33594
Cottonwood/Willow	0.2351	0.9625	0.3779	587
Aspen	0.2199	0.9399	0.3564	7333
Douglas-fir	0.5564	0.8712	0.6791	15207
Krummholz	0.5867	0.9617	0.7288	18350
weighted avg	0.7833	0.7435	0.7533	565892
test set accuracy	0.7435	0.7435	0.7435	0.7435

Table 2.4: Precision, recall, f1-score summary table for onehidden_red-51-46-51-120-7.

	precision	recall	f1-score	support
Spruce/Fir	0.7606	0.7098	0.7343	209680
Lodgepole Pine	0.8092	0.7182	0.7610	281141
Ponderosa Pine	0.7843	0.8386	0.8105	33594
Cottonwood/Willow	0.2228	0.9642	0.3620	587
Aspen	0.2455	0.9183	0.3874	7333
Douglas-fir	0.5107	0.8369	0.6343	15207
Krummholz	0.5834	0.9495	0.7227	18350
weighted avg	0.7664	0.7358	0.7441	565892
test set accuracy	0.7358	0.7358	0.7358	0.7358

Table 2.5: Precision, recall, f1-score summary table for twohidden_red-51-46-51-30-15-7.

Chapter 3

Conclusion

Table 3.1 and 3.2 show that the model yielding highest results both for precision and accuracy is onehidden_red. Considering the second best (twohidden_red), we notice it achieves the best result only when it comes to the precision for predictions about samples coming from the first class. In the majority of the other cases it is beaten by either onehidden or onehidden_red. Being the second best means that the model manages to keep high average precision and recall scores, but it does not stand out in the classification of any specific class. In addition to this its average accuracy is positively influenced by its recall (0.7182) in predictions related to samples belonging to the majority class. The onehidden model's scores are often really good both in precision and recall, but its average accuracy suffers from a low recall in the predictions inherent to the first class (which causes a decrease of the precision scores of all the other classes for said model). Similar observations can be done for the twohidden model, which is weak in recognising the samples belonging to the majority class. This model, though, has a really high recall value for recognising the samples belonging to the minority class, but this result is not really impressive because of the low precision score on the same class (which means that many samples belonging to other classes are recognised as belonging to the minority one). Finally, the baseline model does not obtain any good result, with an average accuracy of 0.5598 (our best model scores 0.7435) and an average precision of 0.6964 (our best model scores 0.7833).

	baseline	onehidden	onehidden_red	twohidden	twohidden_red	Best model
Spruce/Fir	0.6475	0.7092	0.7708	0.7700	0.7098	onehidden_red
Lodgepole Pine	0.4675	0.7207	0.6837	0.6226	0.7182	onehidden
Ponderosa Pine	0.5663	0.7343	0.8498	0.7118	0.8386	onehidden_red
Cottonwood/Willow	0.9063	0.9455	0.9625	0.9813	0.9642	twohidden
Aspen	0.7196	0.8805	0.9399	0.8773	0.9183	onehidden_red
Douglas-fir	0.5759	0.8973	0.8712	0.8131	0.8369	onehidden
Krummholz	0.8729	0.9663	0.9617	0.9478	0.9495	onehidden
Weighted avg recall	0.5598	0.7323	0.7435	0.7019	0.7358	**onehidden_red**

Table 3.1: Weighted avg recall (accuracy) comparison among the discussed model. The best model is onehidden_red-51-46-51-120-7.

	baseline	onehidden	onehidden_red	twohidden	twohidden_red	Best model
Spruce/Fir	0.6637	0.7569	0.7503	0.6801	0.7606	twohidden_red
Lodgepole Pine	0.7898	0.8110	0.8482	0.8374	0.8092	onehidden_red
Ponderosa Pine	0.6497	0.7813	0.7880	0.7666	0.7843	onehidden_red
Cottonwood/Willow	0.0695	0.2498	0.2351	0.1316	0.2228	onehidden
Aspen	0.0663	0.2582	0.2199	0.2351	0.2455	onehidden
Douglas-fir	0.2611	0.4571	0.5564	0.5062	0.5107	onehidden_red
Krummholz	0.3569	0.5728	0.5867	0.5418	0.5834	onehidden_red
Weighted avg precision	0.6964	0.7642	0.7833	0.7479	0.7664	**onehidden_red**

Table 3.2: Weighted avg precision comparison among the discussed model. The best model is onehidden_red-51-46-51-120-7.

Bibliography

[1] Jock A. Blackard and Denis Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 1999.

[2] Pawel Smialowski, Dmitrij Frishman, and Stefan Kramer. Pitfalls of supervised feature selection. *Bioinformatics*, 26(3):440–443, 2009.

[3] Wikipedia. Directional statistics, 2019. Online; accessed July-2019.

[4] Wikipedia. Feature scaling, standardization (z-score normalization), 2019. Online; accessed July-2019.