POLITECNICO
DI TORINO

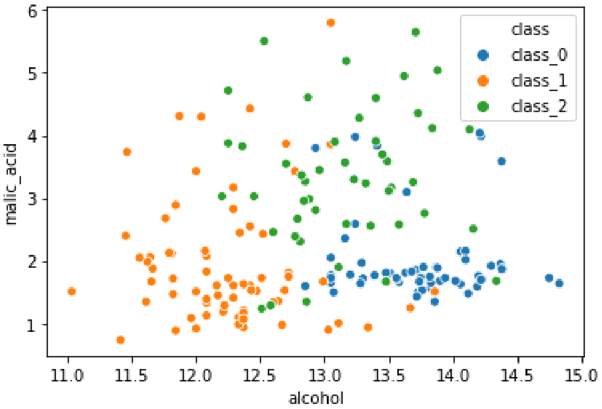
Report Homework 1
Course: Machine learning and deep learning

Davide Bussone

16 May 2020

1 Data exploration

The dataset examined is called "Wine dataset". It could be considered as a multi-class dataset. It is composed of thirteen real and positive features and by a "target" label, useful to identify a different type of wine.
The aim is to study how two attributes of the dataset influence the wine type's classification. As you can see in scatter plot 1, the features considered are "alcohol" and "malic acid". This distribution indicates, for instance, that in presence of small values of both alcohol and malic acid, wine belongs to class 1.



Reading from graph 1, you can infer that data are not distributed in an uniform way. For example, a class counts 71 elements, while, another class is less represented with only 48 items. This disparity may be a factor leading to have a less score when the evaluation of the model is computed on the test set.

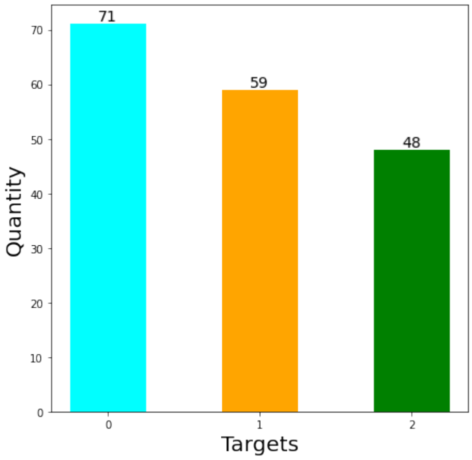


Figure 1: Number of items for each target

As already mentioned, the dataset was split into three subsets, the training set, the validation set and the testing set. Before doing that, data were normalized using the StandardScaler method, provided by the library Scikit learn, into the section preprocessing. This split action follows the proportion represented in graph 2.

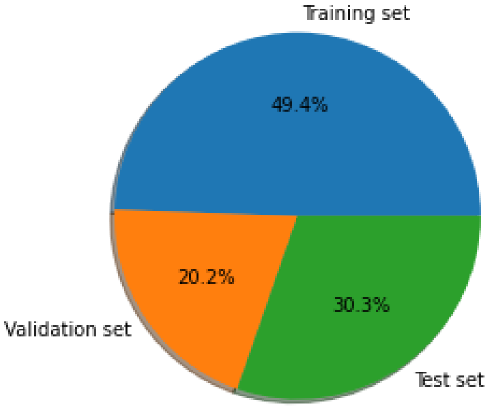


Figure 2: Split into training,validation and test set

2 K-nearest neighbors

The first algorithm to apply is the K-nearest neighbor classifier. Its goal is to find a predefined number of training samples closest in distance to a new point or to a new data, and predict the target label considering this distance.

K values	Accuracy score
1	0.694
3	0.722
5	0.750
7	0.750

Table 1: Table of K values for KNN

The parameter to tune was K. K is a parameter that refers to the number of nearest neighbours to include in the majority of the voting process. As the table 1 shows, the best accuracy, on the validation test, is obtained by applying K=5 or K=7. This, apparently, means that the results are better with a higher number of nearest neighbours (obviously only between the four values we are considering).

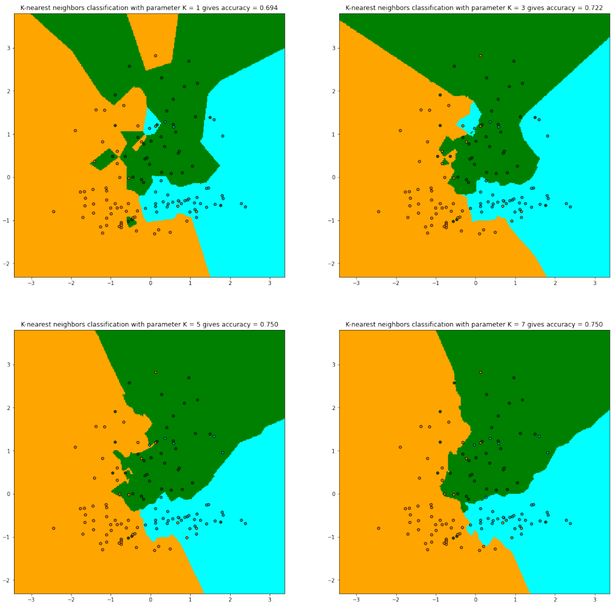


Figure 3: Decision boundaries for all K

In plot 3, you can see the decision boundaries for each K considered, applied on the validation set and also all relative accuracy scores.

Finally, after the tuning, the choice of K was K=5. As a consequence, KNN algorithm was implemented on the test set using this hyperparameter. The accuracy score obtained is higher than the accuracy in the validation set (0.81 against 0.75).

In addition to that, if the choice of K was K=7 (it could be an available choice because the accuracy score on the validation set is the same), the decision boundaries were very similar to graph 4.

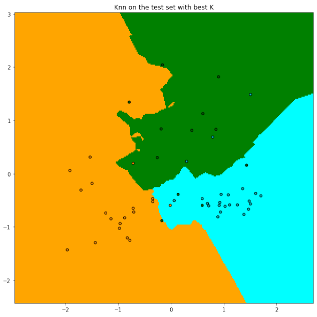


Figure 4: Decision boundaries for the best K

3 Linear support vector machines

In a very simple way, SVM finds a separating hyperplane between data of some classes. So, SVM is an algorithm that takes the data as an input and gives as output a hyperplane that separates those classes if possible. In this section, SVM uses a linear separator (technically, SVM has a linear kernel).

C values	Accuracy score
0.001	0.417
0.01	0.444
0.1	0.750
1	0.750
10	0.778
100	0.778
1000	0.778

Table 2: Table of C values for linear SVM

The parameter to tune was C. C is a parameter that refers to the penalty assigned to the model in case of misclassification. As shown in the board 2, the best accuracy, on the validation test, is obtained by computing C=10, C=100, C=1000. This, apparently, means that increasing the value of the penalty, the score becomes better.

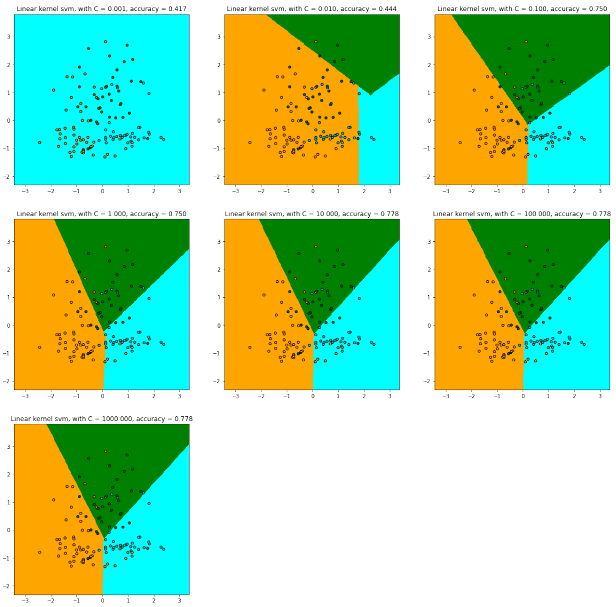


Figure 5: Decision boundaries for all C

In plot 5, you can see the decision boundaries for each C considered, run on the validation test

and all relative accuracy scores. As you can see, while you are increasing the value of C, the decision boundaries become more similar and also more precise in terms of data classification.
To conclude, after the tuning, the choice of C was C=100. As a consequence, linear SVM algorithm was applied on the test set using this hyperparameter. The accuracy score obtained is higher than the accuracy in the validation set (0.81 against 0.78).
In addition to that, if the choice of C was C=10 or C=1000 (it could be an available choice because the accuracy score on the validation set is the same), the decision boundaries were very similar to the image 6.

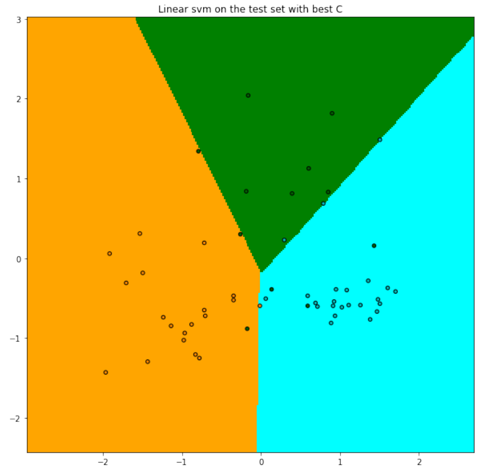


Figure 6: Decision boundaries for the best C

4 Radial support vector machines

In this case, the kernel of SVM is changed from a linear kernel to a radial kernel, that is a non-linear kernel.
The parameter to tune was again C. As you can infer from the bench 3, the best accuracy, on the validation test, is obtained by using C=0.1, C=1, C=10. This, apparently, means that in case you penalize in a weak way misclassifation errors, you get a very low accuracy (underfitting), but also that if you penalize too much these errors, your accuracy score goes down (risk of overfitting). Maybe, this behaviour is an effect of the non linearity.

C values	Accuracy score
0.001	0.417
0.01	0.417
0.1	0.750
1	0.778
10	0.750
100	0.722
1000	0.694

Table 3: Table of C values for radial SVM

In picture 7, you can see the decision boundaries for each C considered applied on the validation test and all relative accuracy scores.

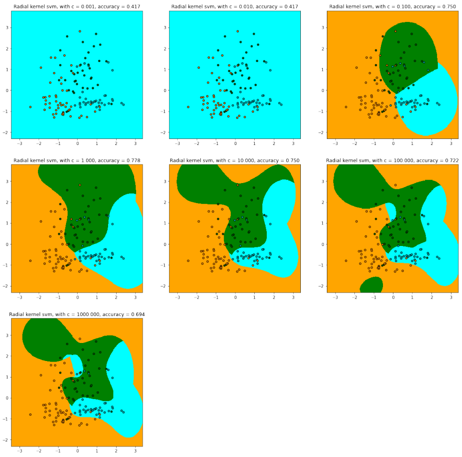


Figure 7: Decision boundaries for all C

At the end, after the tuning, the choice of C was C=1. As a consequence, radial SVM algorithm was applied on the test set using this hyperparameter, as you can see boundaries in figure 8.

The accuracy score obtained is higher than the accuracy in the validation set (0.83 against 0.75). Moreover, this means that machine learns well how to classify the data from the train-validation procedure, because in the test set results become better in a gradual way. Actually, these conclusions are available for all algorithms applied until now.

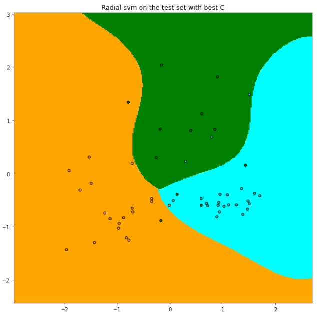


Figure 8: Decision boundaries for the best C

A support vector machine algorithm, with a radial kernel, has another parameter that could be tuned together with C: it is called "gamma". Graphically, the gamma parameter can be said to adjust the curvature of the decision boundary. To find out the best combination of C and gamma, the method chosen is the grid search. Initially, grid search was applied without cross-validation ,then this method is repeated with K-fold cross-validation using five splits.

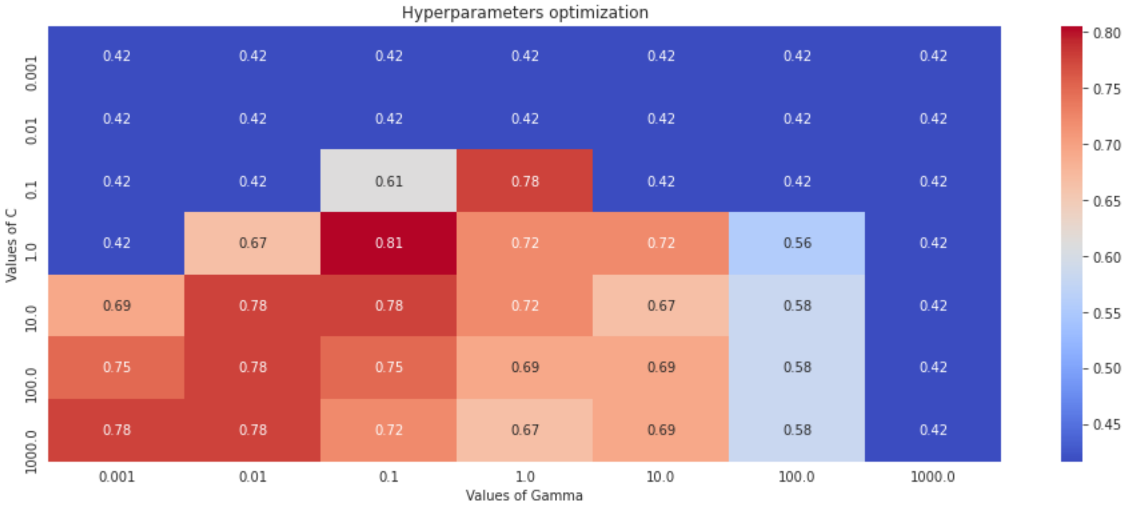


Figure 9: Hyperparameter search accuracy

After the first type of grid search applied on the validation set, the heatmap in image 9 shows

the best combination of hyperparameters: C=1 and gamma=0.1, with accuracy score 0.81.
Then, the hyperparameters found were evaluated on the test set (so C=1and gamma=0.1), as you can see in plot 10. Gamma is not too much great, so the "adjustment" of the curvature of decision boundary is not so significant. The accuracy score obtained increases significantly, from 0.81 on the validation set to 0.8703 on the test set.

Finally, after computing a merge between training set and validation set, a k-fold validation was performed, which is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. In this case k is set to 5. This technique is primarily used in machine learning to estimate the skill of a machine learning model on unseen data. Its aim is to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.So, 5 iterations were computed; at each iteration there was a different portion of the training set that turns into validation set. The final accuracy score is the average of the score obtained in each iteration. The best matching of hyperparameters found, after these five iterations, was C=1 and gamma=1. The accuracy score was around 0.8063 on the validation set. Then, these hyperparameters were applied on the test set and the accuracy increases to 0.8703. Generally k-fold cross validation results in a less biased estimate of the model performance than other methods, like a simple train test split holdout or like a grid search without cross validation. Actually, in this case grid search and k-fold cross validation lead to the same results. This can be a consequence of the fact that the dataset is small.

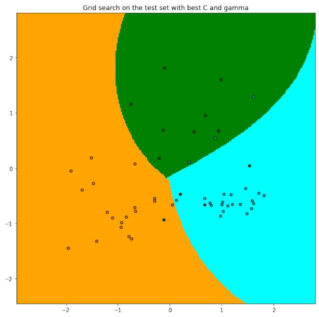


Figure 10: Best combination of C and gamma on the test set

5 Difference between KNN and SVM

K-nearest neighbour classifier makes its decisions on the basis of local information. So, it does not draw decision boundaries across the whole space, but classification task is founded on few local points. It is considered a lazy algorithm because it does not learn a discrimination function from training data, but it memorizes the training dataset. On the other side, support vector machine builds a hyperplane in the entire n-dimensional space and it computes a discrimination function in view to catch the optimal separating hyperplane.

6 Different pairs of attributes

In this section, all the possible combinations of two attributes, in the Wine dataset, were considered. Better results on the test set, compared to the couple "alcohol" and "malic acid", were obtained with four combinations. The only algorithm applied is the radial support vector machine. Moreover, only best hyperparameters were considered, thanks to a grid search applied for this purpose.

The best combination is portrayed in 14, it reaches an accuracy on the test set of 0.92, with best parameters C=1 and gamma=10. In 13 the accuracy grows until 0.90, with best parameters C=10 and gamma=0.1. 0.88 is the value of the score of the couples in figure 11 and 12.
As you can infer from these graphs, only four combinations (total was 77) achieved a greater result than the case study pair, so it could be said that alcohol and malic acid are features that contribute well to classify the different types of wine.

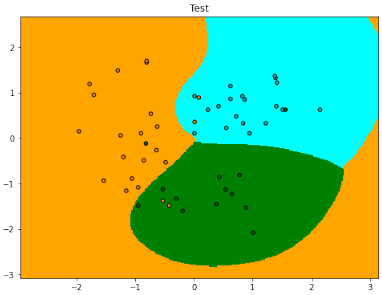


Figure 11: Decision boundaries for ('alcohol', 'total phenols')

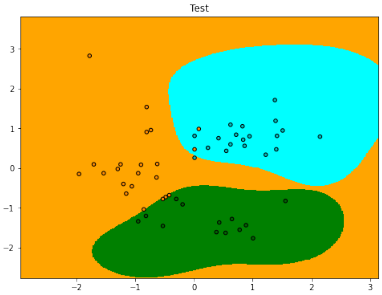


Figure 12: Decision boundaries for ('alcohol', 'flavanoids')

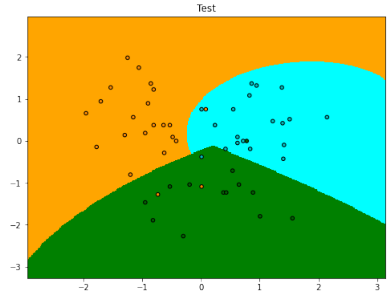


Figure 13: Decision boundaries for ('alcohol', 'hue')

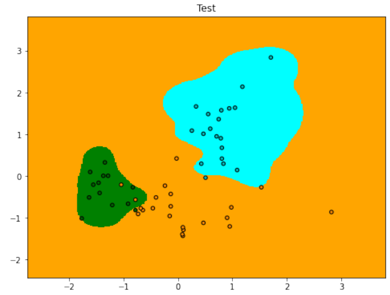


Figure 14: Decision boundaries for ('flavanoids', 'proline')

7 Conclusions

The algorithm best performing is "Support vector machine" with a radial kernel. In the investigated case, actually, all the algorithms proposed work well, maybe because the dataset has a small shape (178,13). In addition to that, a radial kernel is preferable to a linear kernel, because the non-linearity let avoid misclassification errors due to the impossibility to separate some classes,linearly.