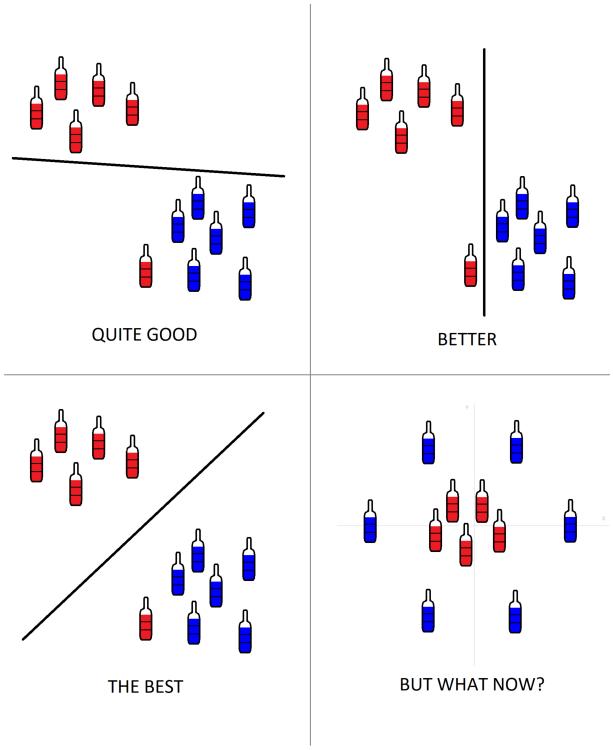
POLITECNICO DI TORINO

Homework 1

Student name: *Erich Malan*
s267475

Course: *Machine Learning and Deep Learning* – Professor: *Barbara Caputo*
Due date: *June, 2020*



Wine Dataset Classification

Introduction

The aim of this assignment is the wine classification of the Wine Dataset (sklearn) through KNN and SVM and their hyperparameters finetuning. Along this report are presented and discussed the approaches, their related results and comparisons.

Data

0.1. Data exploration. The dataset proposed is about different types of wine and is composed by 178 records characterized by 13 numerical features and a discrete multi-class target label. In particular each record is described by three possible label values {0,1,2} denoting a different wine type.
Features:
alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, od280/od315 of diluted wines, proline
The target's distribution is not properly uniform, but does not even present a tremendous disparity (as depicted in *figure 1*). As we may notice in *figure 2*, these two classes show different domains per each wine but common conflicting areas are also present.

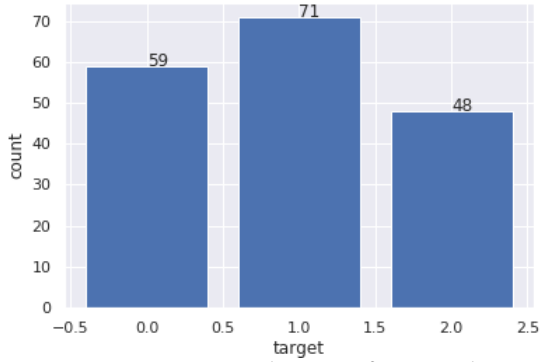


Figure 1: *Distribution of target class*

Figure 2: *Wine dataset's scatterplot of malic acid over alcohol*

Data preparation. Accordingly to the assignment requirements only *alcohol* and *malic acid* features are selected. Then in order to perform the analysis, I splitted the dataset in 3 chunks to complete their respective roles: *training, validation, test*, with ratio of [5,2,3] (fig. 3). The split is performed considering the distribution of features, as it is probed through the scatterplots posted below. Their objectives are respectively for: training the algorithm, check the accuracy on the validation split, according to its value finetune the hyperparameters and finally test the performance on the test set. Using a separate set for validation and testing is designed to prevent overfitting.

KNN - K Nearest Neighbors

Training & Validating. The KNN is a classification algorithm that considers the nearest neighbors to classify a new record. This attempt starts from just one neighbor up to seven with an increment of 2-step policy. As shown below in the decision boundaries plots, the number of neighbors highly affects the result, by a first sight it can be seen

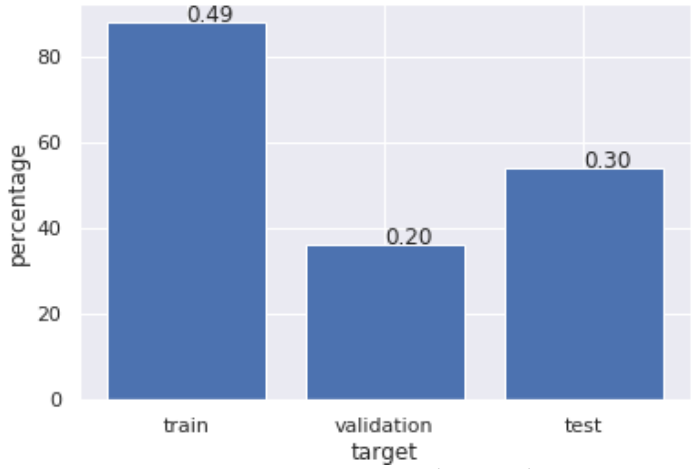


Figure 3: *Train,eval,test split*

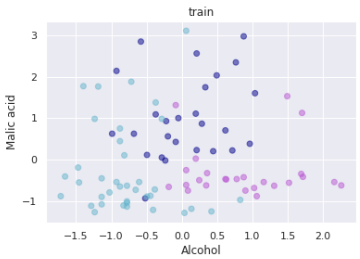


Figure 4: *Train Scatterplot*

Figure 5: *Validation Scatterplot*

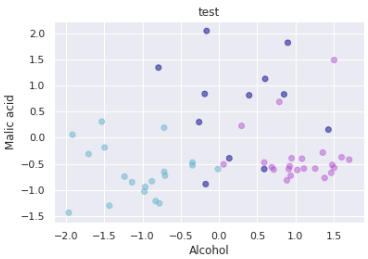


Figure 6: *Test Scatterplot*

that initially the blue small area (top-mid of 1NN) and then the green one (mid-mid of 3NN) are initially separated from their main domains and then along with the increase of neighbors will disappear. The boundaries of 5NN/7NN seem to be clearly more steady and surely less noisy. This aptitude of algorithm might be lead to a 'more neighbors -> better model' law, hypothesis corroborated by the accuracies table presented. Obviously this law has a turning point, as depicted in *figure 11*, too high Ks would lead to a 1 label only classification, near 80 for example the accuracy starve near 40% and every record would be labeled as '1'.

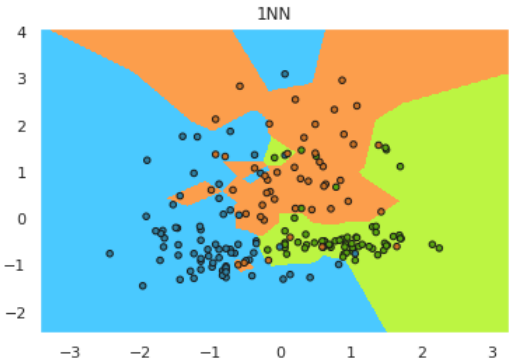


Figure 7: *1NN decision boundary*

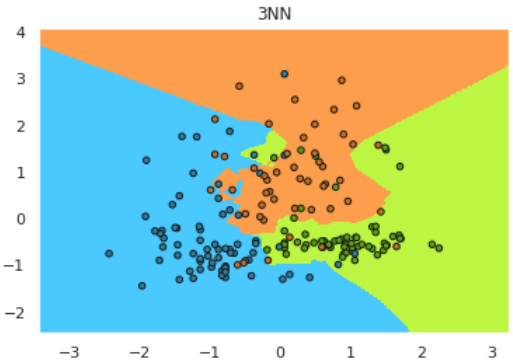


Figure 8: *3NN decision boundary*

Testing. The first higher accuracy should be fine and reasonable to be tested on the test set to check if the model is overfitted on the validation set. The accuracy obtained

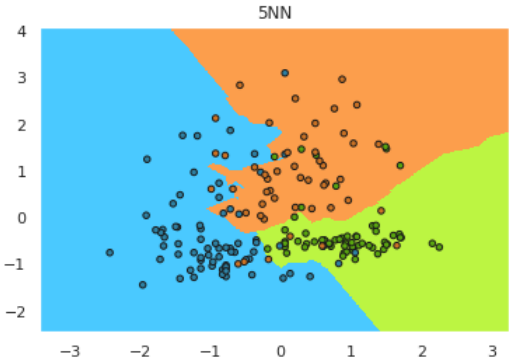


Figure 9: *5NN decision boundary*

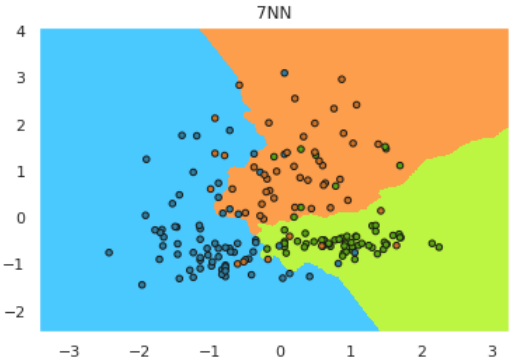


Figure 10: *7NN decision boundary*

Table 1: Validation accuracy over K	
K	Acc.
1	0.69
3	0.72
5	0.75
7	0.75

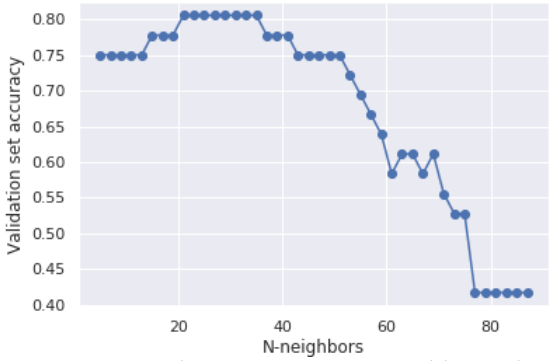


Figure 11: *val accuracies over neighbours lineplot*

is ≈ 0.81. *figure 12* and *figure 13* shows ground truth labels and predicted ones, as we could imagine by deduction from the decision boundary plots, the misclassification errors lie near edges over common areas.

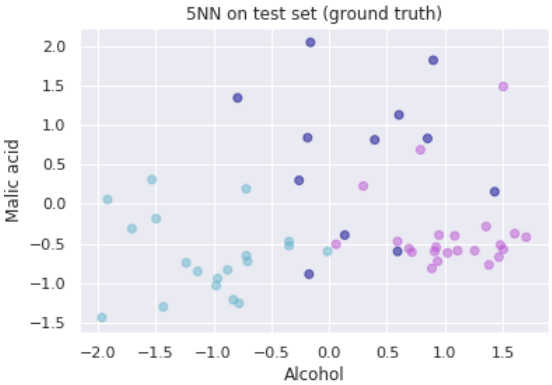


Figure 12: *5NN scatterplot ground truth*

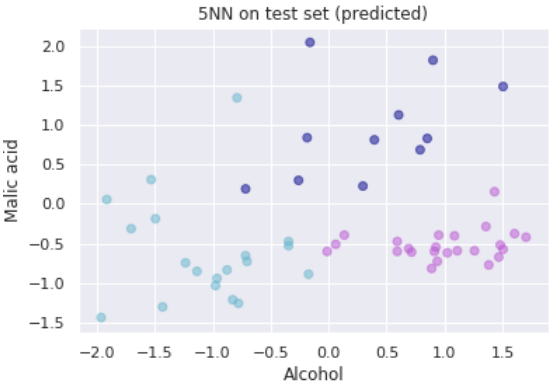


Figure 13: *5NN scatterplot predicted*

SVM

Linear SVM. The second part of the assignments shares the same aim but the task will now be achieved by the Support Vector Classifier model. In particular the various experiments pertain to the kernel type (linear vs rbf) and hyperparameters such as C, and γ. C represent the cost of misclassification (in the case of a soft margin problem)

while γ influence the radius of the model (whenever the model is based on a RBF kernel). The figures below show the behaviour of the Linear SVM along to the increase of C.

Figure 14: *linear SVM plots*

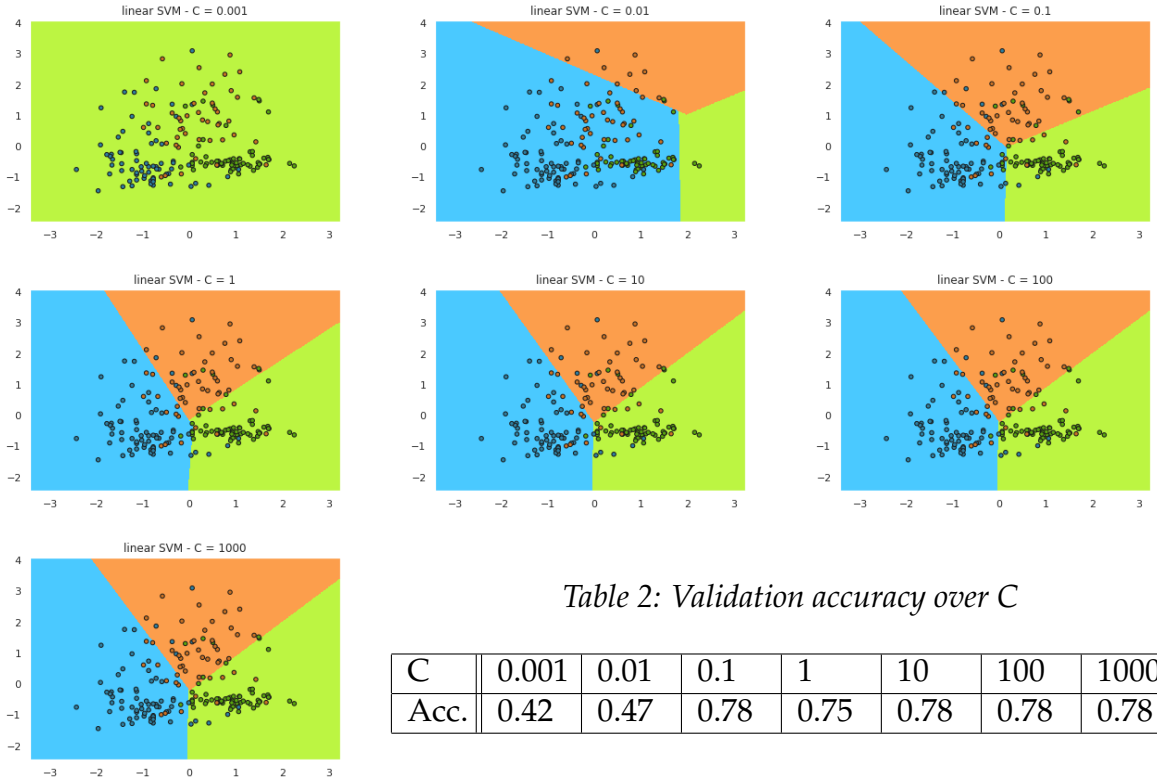


Table 2: Validation accuracy over C

C	0.001	0.01	0.1	1	10	100	1000
Acc.	0.42	0.47	0.78	0.75	0.78	0.78	0.78

As explained lower C means lower penalty for misclassification, and C = 0.001 is representative in this case because is shown as a single label classification (the same as KNN for 80 neighbors, the class with higher probability will be chosen). So theoretically by incrementing it, the cost increase and the model try to be more precise, but this has a turning point too as reported in the validation accuracies (*table 2*). The best accuracy is 0.78 and we found it for C = {0.1, 10, 100, 1000}. This time I chose C = 100 to evaluate the model on the test set because is the middle point of higher accuracies (while for example C = 0.1 could be just a lucky glory split that decrease for C = 1 for values of 10,100,1000 the decision boundaries seem to be pretty similar).

Accuracy obtained on test set for C = 100: 0.815, even higher than validation set.

Radial SVM. As enounced before next experiments are related to the use of the RBF kernel (Radial Basis Function), as we can imagine now the boundaries are no more going to be linear. Let's perform the same C search as we did with LSVC.

The best accuracy is 0.75 and the middle point is for C = 1, and the respective result on the test set is 0.83, even in this case is higher than validation set's one. As we may notice in fig. 15 the decision boundaries differs a lot for C > 0.001 by showing curved edges. Moreover for C = 0.01 the RBF is still performing a single label classification differently from the linear one.

Now the next parameter to be tested is γ, an hyperparameter that defines the kernel function $e^{-\gamma|x_i - x_j|^2}$, which grew practically is inversely proportional to the variance of

Figure 15: *radial SVM plots*

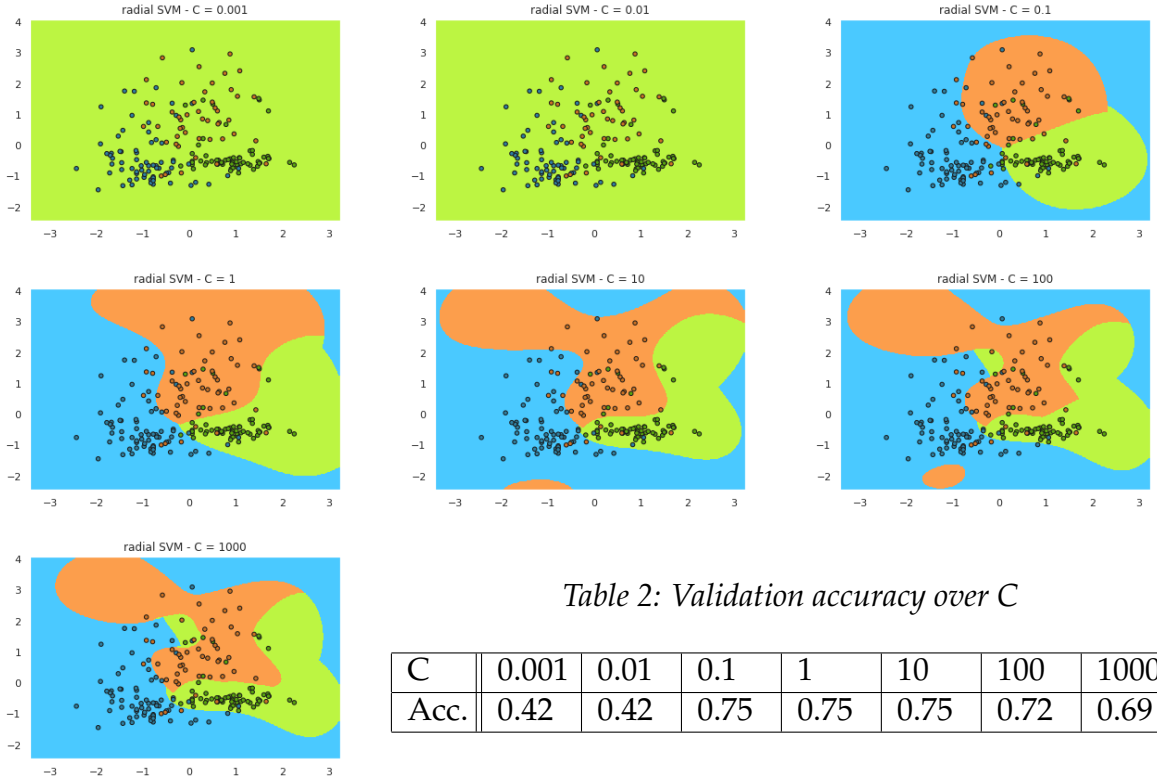


Table 2: Validation accuracy over C

C	0.001	0.01	0.1	1	10	100	1000
Acc.	0.42	0.42	0.75	0.75	0.75	0.72	0.69

the function. Higher gammas values mean smaller variance and so two points would be considered similar if they are close between them (small radius of influence). Here is a proposal of γ's values and the grid search performed. (*figure 16*)

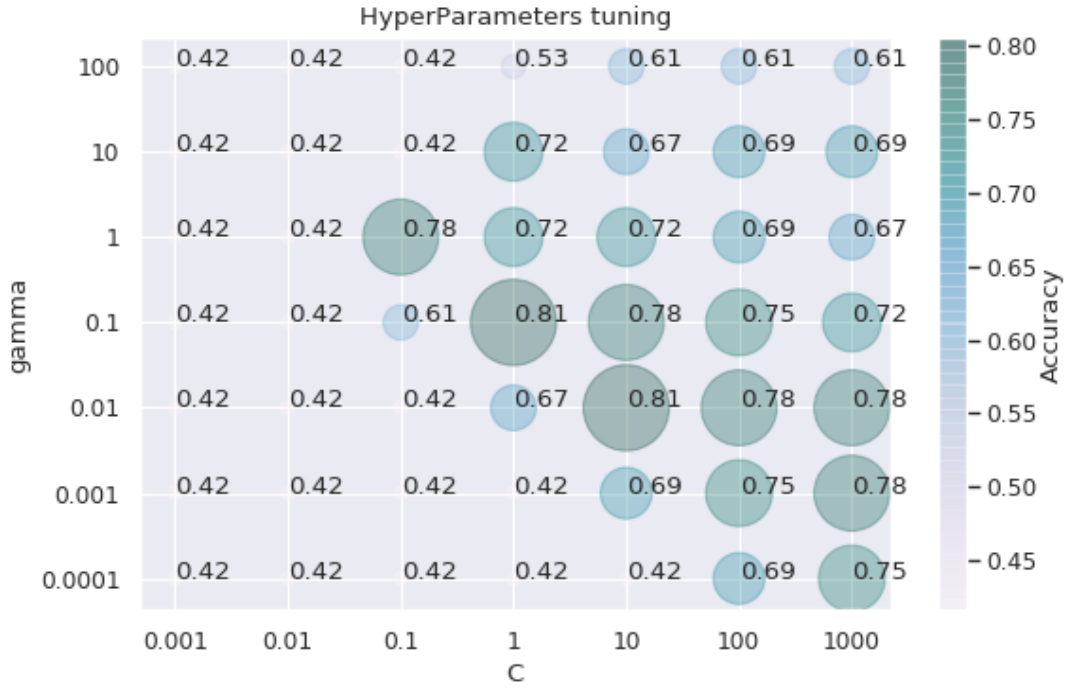


Figure 16: Grid search on validation accuracy with RBF kernel

By a first look it appears that there is a sort of triangular prism almost centered in the grid. I evaluated as best parameters on test set C = 10 and γ = 0.01 and obtained an accuracy of 0.87 which is pretty good (best one between every model finetuned in this paper). Here it is the scatterplot of test set with the decision boundaries.

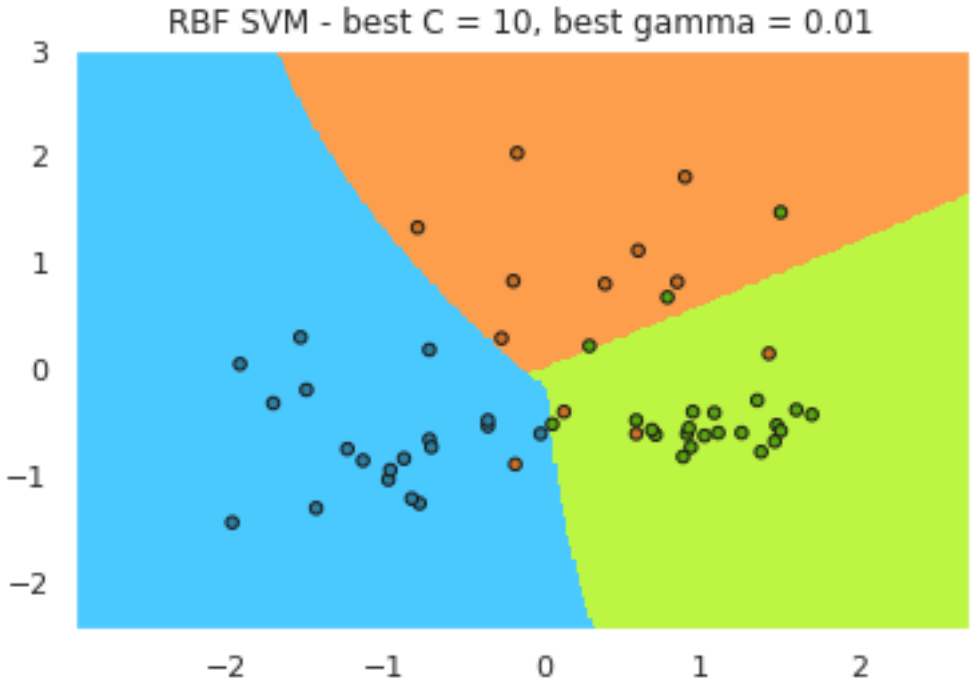


Figure 17: Decision boundary plot RBF test (the points in this case are part of test set, in order to better visualize misclassified records)

The next step is merging the test set and the validation set in order to get a train set of 70% record and test set of 30%, perform a grid search on training, this time with a 5-fold cross validation, and finally test it on the test set with best parameters found.

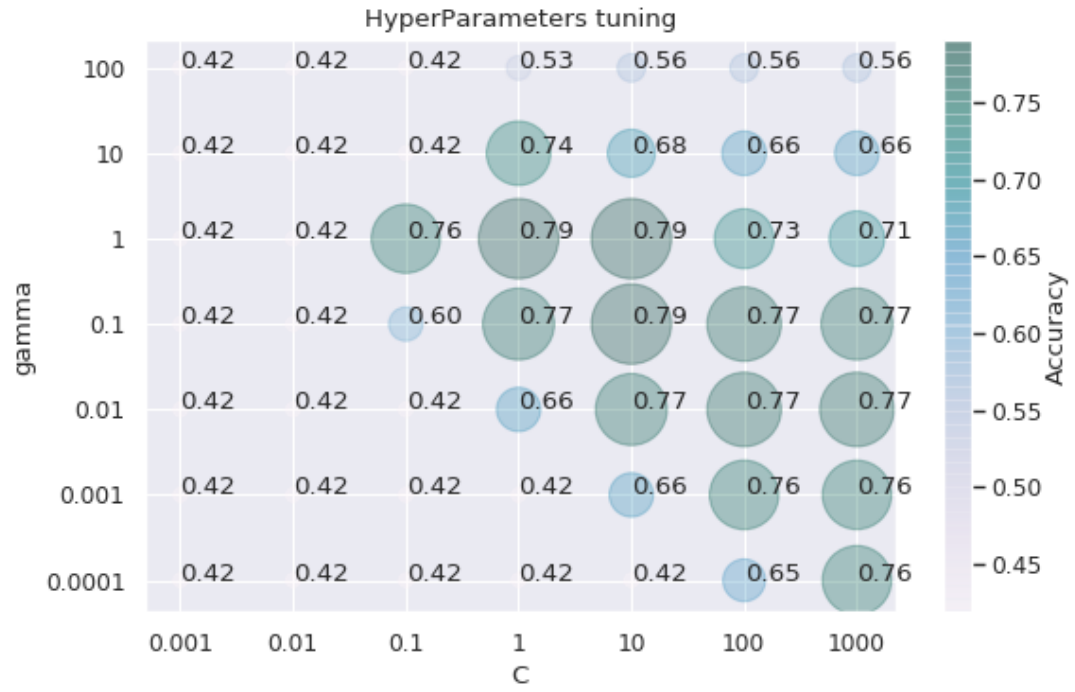


Figure 18: Grid search on validation accuracy with RBF kernel (CV)

In fig 18 is represented the new grid search, while in fig 19 are represented the decision boundaries of the new training set. Typically more training data means a better model, and in this case the accuracy is the same but theoretically it should be more steady versus overfitting, and it must be additionally pointed out that the validation set should have decreased the model accuracy accordingly to previous results but it did not. In fact the 5-fold cross validations is a deeper search because it performs 5 different splits on training set, everyone of them is used separately as validation set while the others are used for training, this means that each part is used both for training and validating separately. For each combination of parameters 5 iteration are performed to check the average and variance of accuracy results to checking the model stability over the different splits. In the end the best couple of parameter is C = 1, γ = 1 and the accuracy of the model on the test set is 0.870.

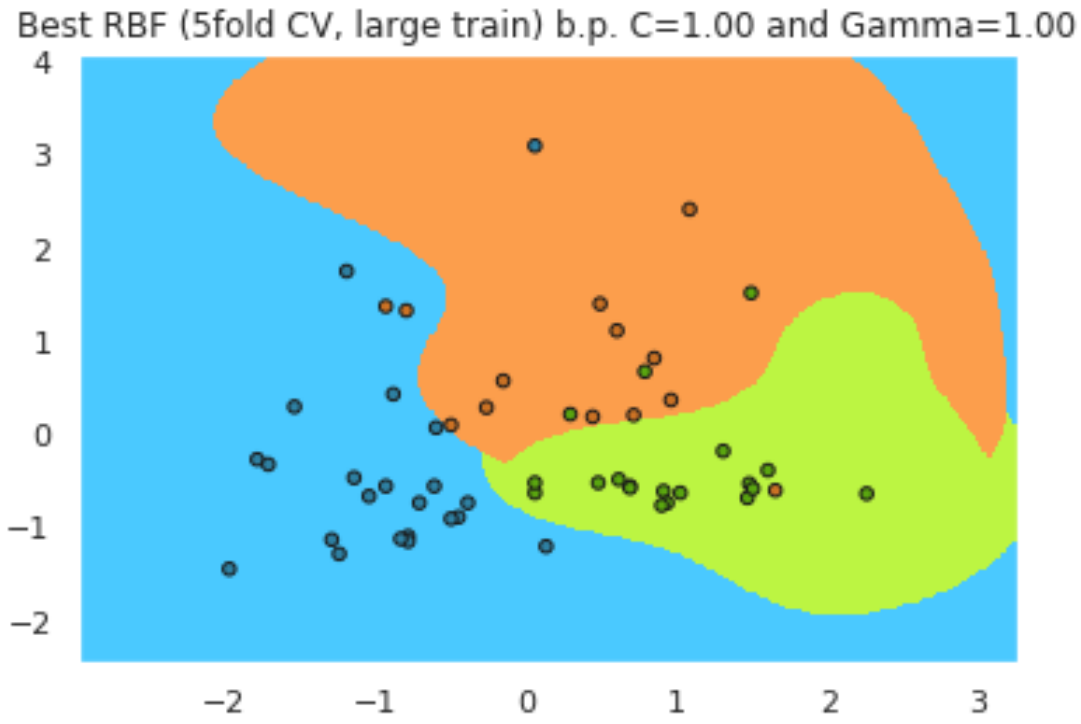


Figure 19: Decision boundary plot RBF test (CV) (the points in this case are part of test set, in order to better visualize misclassified records)

KNN vs SVM

In comparison KNN already with few K neighbors behave as a reliable and sufficient performing model, with a worst case of 0.69 for K = 1. It handles non-linear problems with a simple algorithm just by ranking the distances beetween the points but this is also one of his main problem because on larger dataset for each record it requires lots of computation. On the other hand the simplicity of the algorithm make almost easy to preview a possible range of interesting K. SVM behaves completely different, it aims to build the best hyperplane between the values of features that better splits the targets, that simultaneously means higher training's cost but cheaper classification's one. It also needs a deeper search for finetuning hyperparameters which in my opinion are less transparent to imagine a priori. Moreover, for bad sets of them it performs really bad for example in fig. 16 and fig. 18 there are many 0.42 accuracy or for example even for K = 1 KNN is barely working while first attempt of C values produced 1-label only classification (*fig. 7, fig. 14, fig.15*). But in the end, especially for rbf kernel (that is costlier) the final result is much more precise (Just out of curiosity I tested higher values of K neighbours for KNN and even that the accuracy gap lies near 7 percentage points below the svm-rbf result).

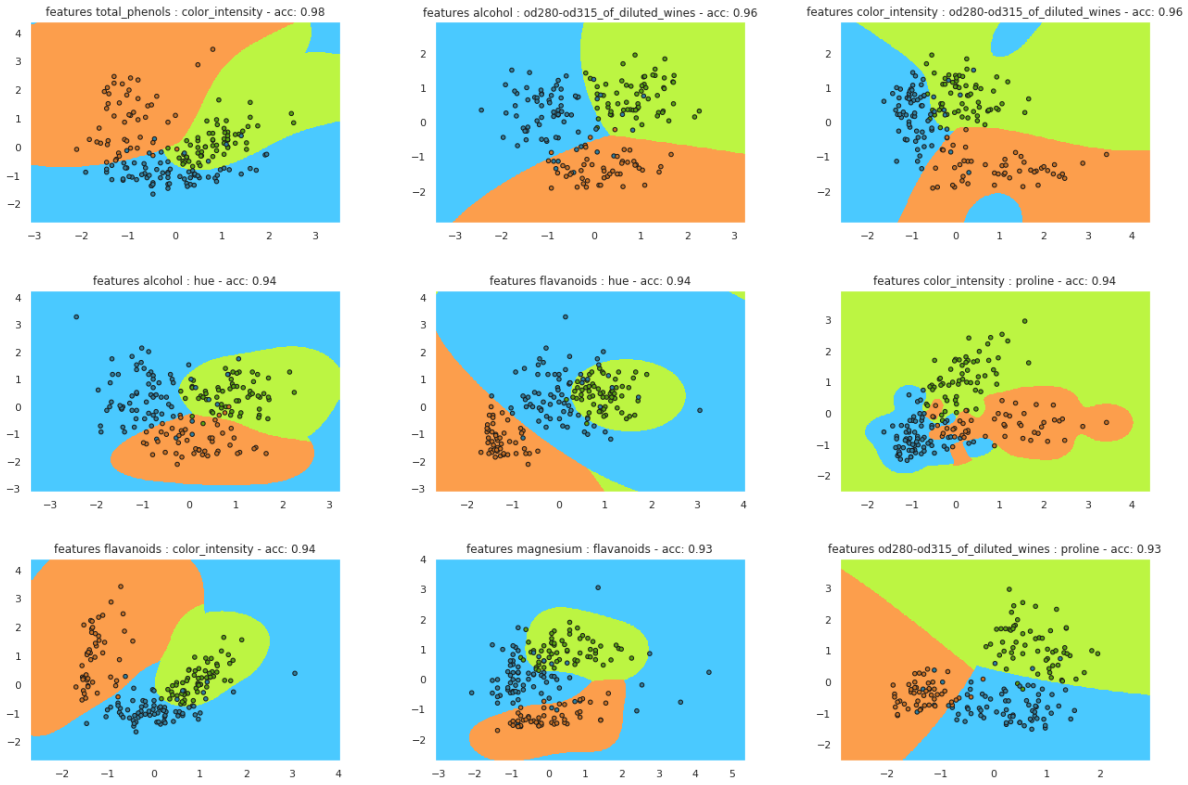
Other couple of attributes

Given that the dataset comprehends 13 attributes plus the target, they produce 78 possible couples of attributes. With an extensive search that tested each couple of them performing a 5fold CV with SVM classification based on the RBF kernel to search best

	Params	Test Acc.
KNN	K = 5	0.81%
LSVC	C = 100	0.815%
RBF-SVC	C = 1, γ = 1	0.83%
RBF-SVC (5 fold sv)	C = 1, γ = 1	0.87%

parameters then applied to evaluate the accuracy on the test set, it appears that 19 of them allow better splits of the one chosen. In particular should be mentioned the couple 'total phenol' & 'color intensity' that produces an accuracy over the test set of 0.98 which is a very impressive result. Following are presented the decision boundary of the 9 best possible couples.

Figure 20: *Best test accuracies obtained by checking other couples of attributes*



It could be stated that probably given the figures above the couple composed by 'od280/od315 of diluted wines' should perform well with the linear SVC too while other more agglomerated / globular shapes such as the ones regarding the 'flavanoids' should be better classified with KNN rather than linear SVC.

feature 1	feature 2	Test Acc.
total phenols	color intensity	0.98%
alcohol	od280/od315 of diluted wines	0.96%
color intensity	od280/od315 of diluted wines	0.96%
alcohol	hue	0.94%
flavanoids	hue	0.94%
color intensity	proline	0.94%
flavanoids	magnesium	0.93%
proline	od280/od315 of diluted wines	0.93%