POLITECNICO DI TORINO

MACHINE LEARNING AND DEEP LEARNING

ALBERTO MARIA FALLETTA - S277971

HOMEWORK 1 - REPORT

NEAREST NEIGHBORS, LINEAR SVM, SVM WITH RBF KERNEL

Introduction

In this assignment we will experiment with two of the most important classification algorithms in machine learning, namely Nearest Neighbors and SVM, and we will visualize results of their implementations along with how these are influenced by changing their parameters.

For the purpose we will be using Scikit Learn's wine recognition dataset, a collection of results of chemical analysis of wines grown in a specific Italian region by three different cultivators. There are thirteen different measurements taken for different constituents found in the three types of wine, these measurements correspond to the numeric 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 dataset is made of 178 samples divided in three different classes (class_0 (59), class_1 (71), class_2 (48)). There are no missing values for features nor for labels.

Assignment

Loading the dataset with the command load_wine( ) the data is returned in the form of a dictionary from which I decided to create a Pandas dataframe of the only two features of interest for a 2D representation (Alcohol and Malic acid).

I randomly divided twice the dataset, the first time between Training + Validation and Test, the second one between Train and Validation, with train_test_split( ) and then plotted the graph of the complete dataset along with its partitions by mean of a custom function plot_dataset_and_partitions( ), with class_0 being green, class_1 orange and class_2 blue.

Figure 1 Visualization of the complete 2D dataset along with a possible split

1. K-Nearest Neighbors

I then applied K-nearest neighbors, training the classifier on the training set and evaluating the accuracy score on the validation set for K= [1, 3, 5, 7], plotting for each value of K the decision boundaries and training data points by mean of np.meshgrid and a custom function plot_training_and_boundaries( )

The second picture shows how decision boundaries change for different values of K. It is possible to see how, for increasing values of the parameter, points far from the centroid of their relative class lose relevance for the classification since K increasing implies a bigger and bigger neighborhood to be considered in order to classify a point.

Furthermore, evaluating these plots for many random partitions of the starting dataset, I noticed areas belonging to a class where I expected they would belong to another one, this is due to ties (es. K=5, closest points: 2 blue, 2 green, 1 orange). Scikit Learn's KNeighborsClassifier, in tie situations, assigns the area according to the order of the class in the training set.

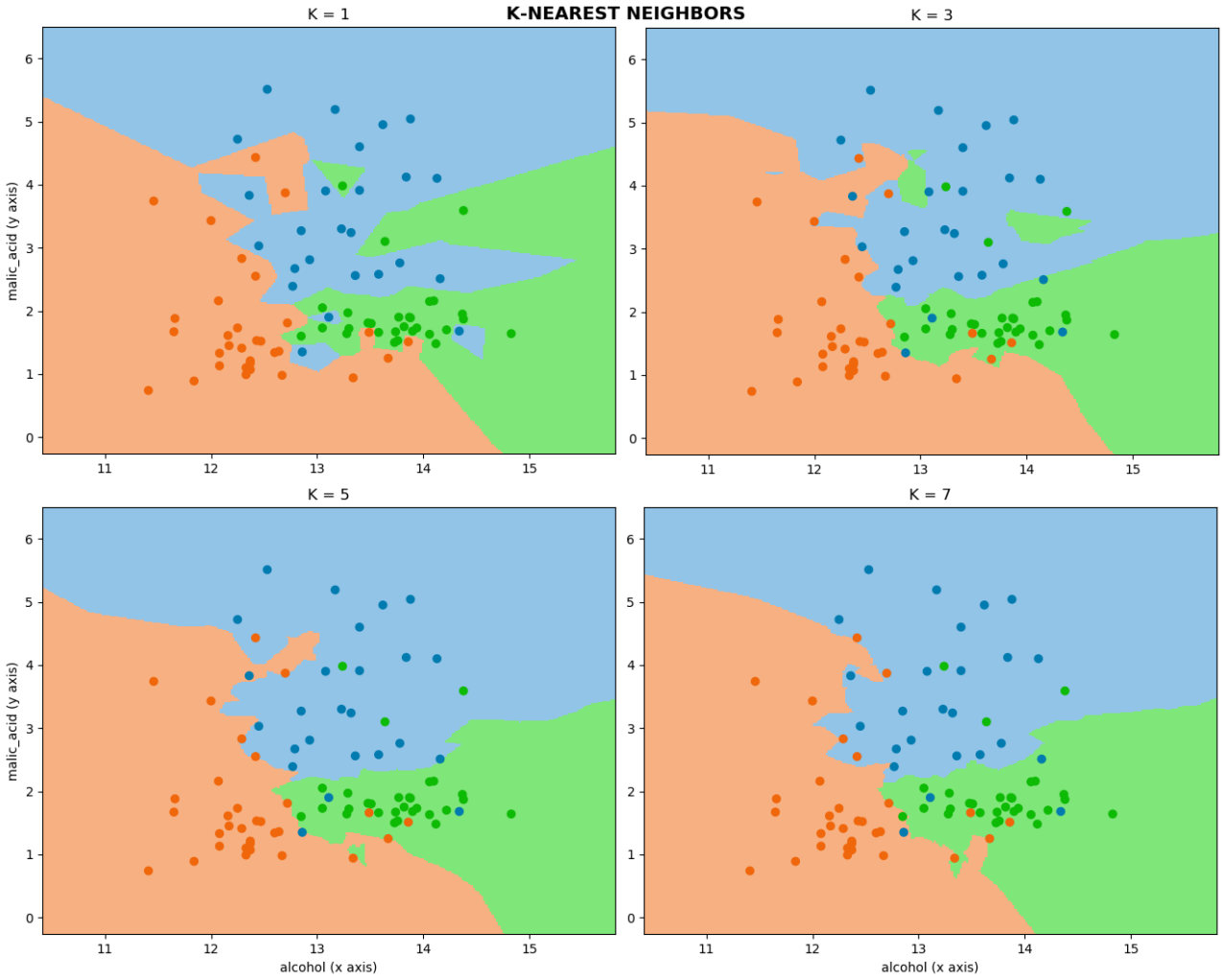

Figure 2 Decision boundaries when K changes

1.1. Accuracy changing K value

With a third custom function plot_accuracy_plot() I plotted the accuracy score achieved on the Validation set with KNeighborsClassifier for different values of parameter K.

The accuracy is highly dependent on the partition split since the dataset is very limited in size. Sometimes for a given training set the accuracy line is increasing monotonic, some other times for other training sets is increasing and decreasing like the one in the picture on the right.

Given these partitions, KNeighborsClassifier with K=3, trained on both training and validation set, has an accuracy score on the test set of 0.778

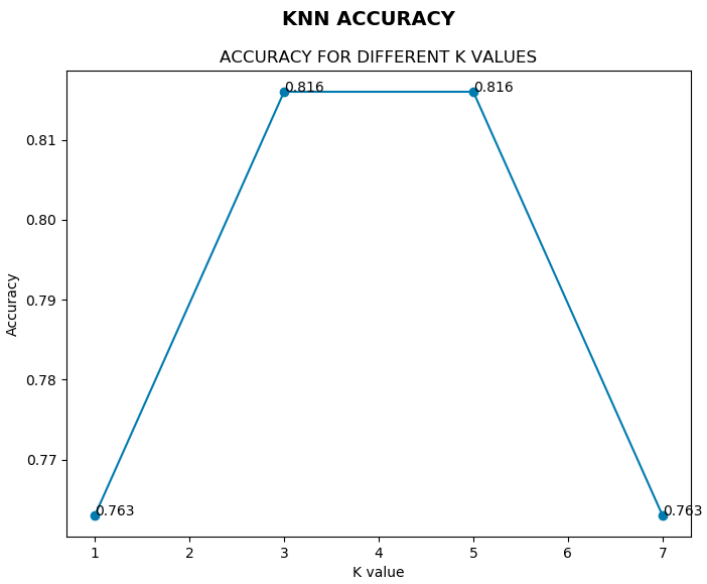

Figure 3 Validation accuracy when K changes

2. SVM

In the next two pages are reported the graphs generated by the application of SVC classifier, trained on the training set and evaluated in accuracy on the validation set for C= [0.001, 0.01, 0.1, 1, 10, 100, 1000] respectively with "linear" kernel first and "rbf" kernel later.

The pictures show how decision boundaries change for different values of C. Since with SVM we are searching for an hyperplane with the largest minimum margin, and a hyperplane that correctly separates as many instances as possible, from the moment our training dataset is not linearly separable, C tells the SVM how much to avoid misclassifying each training point. For large values of C, the SVM chooses a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Otherwise, a very small value of C causes SVM to look for a separating hyperplane with larger margin, even if that hyperplane misclassifies more points.

The difference between the two classifier is the kernel. While in the first one the SVM is linear, which leads to linear separation hyperplanes as shown in the decision boundaries plots, the second one has a Radial basis function kernel "rbf" which is a function whose value depends on the distance from the origin or from some point and for that reason it is able to look for non-linear separation hyperplanes.

Finally, also the plots of the accuracy score achieved on the Validation set with the classifiers for different values of parameter C can be found in the images with, this time, a logarithmic scale to better represent the x axis.

Given these partitions, SVM with "linear" kernel and C=0.1, trained on both training and validation set, has an accuracy score on the test set of 0.704, while SVM with "rbf" kernel and C=10, trained on both training and validation set, has an accuracy score on the test set of 0.722.

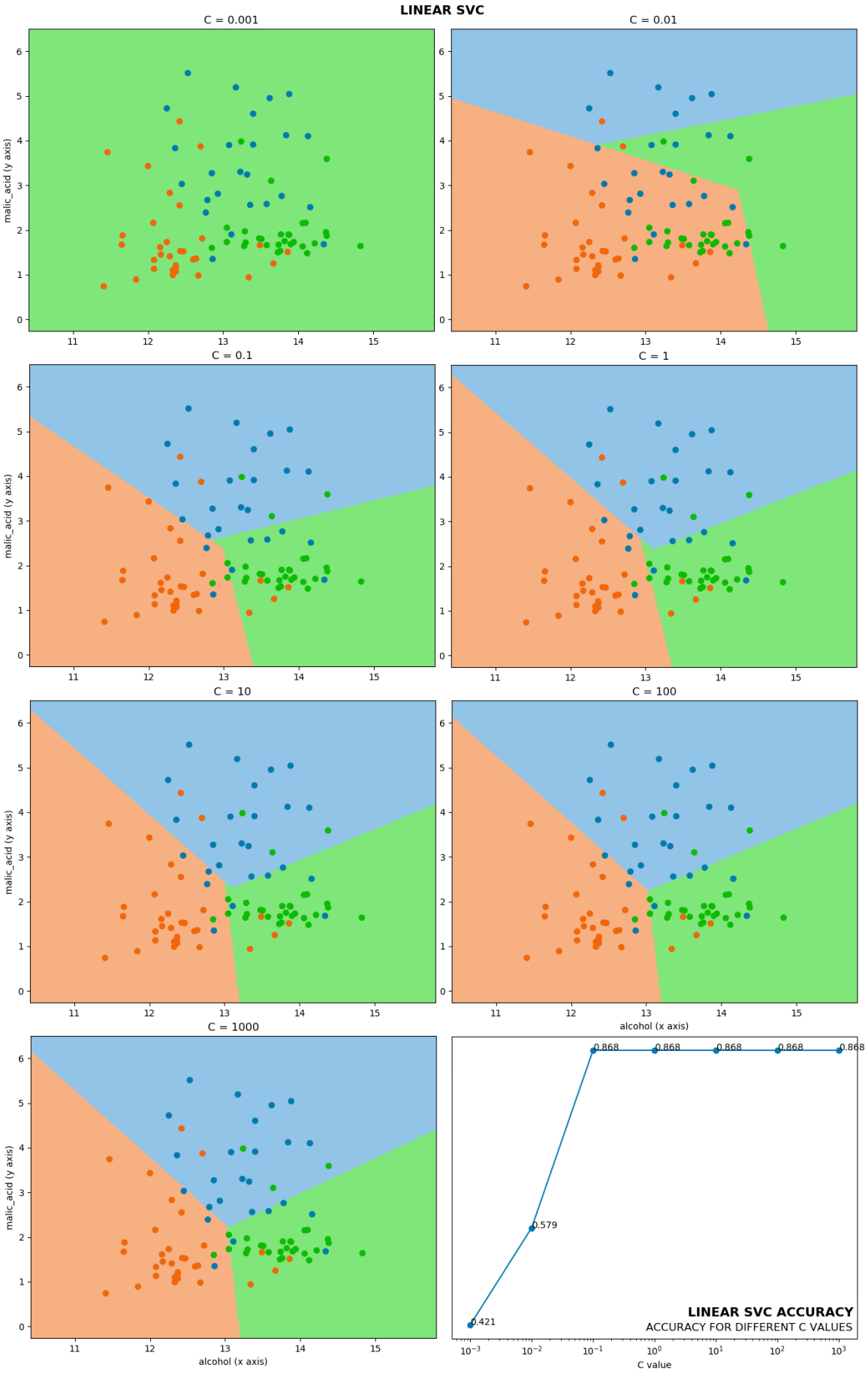

Figure 4 Decision boundaries and validation accuracy when C changes for SVM linear kernel

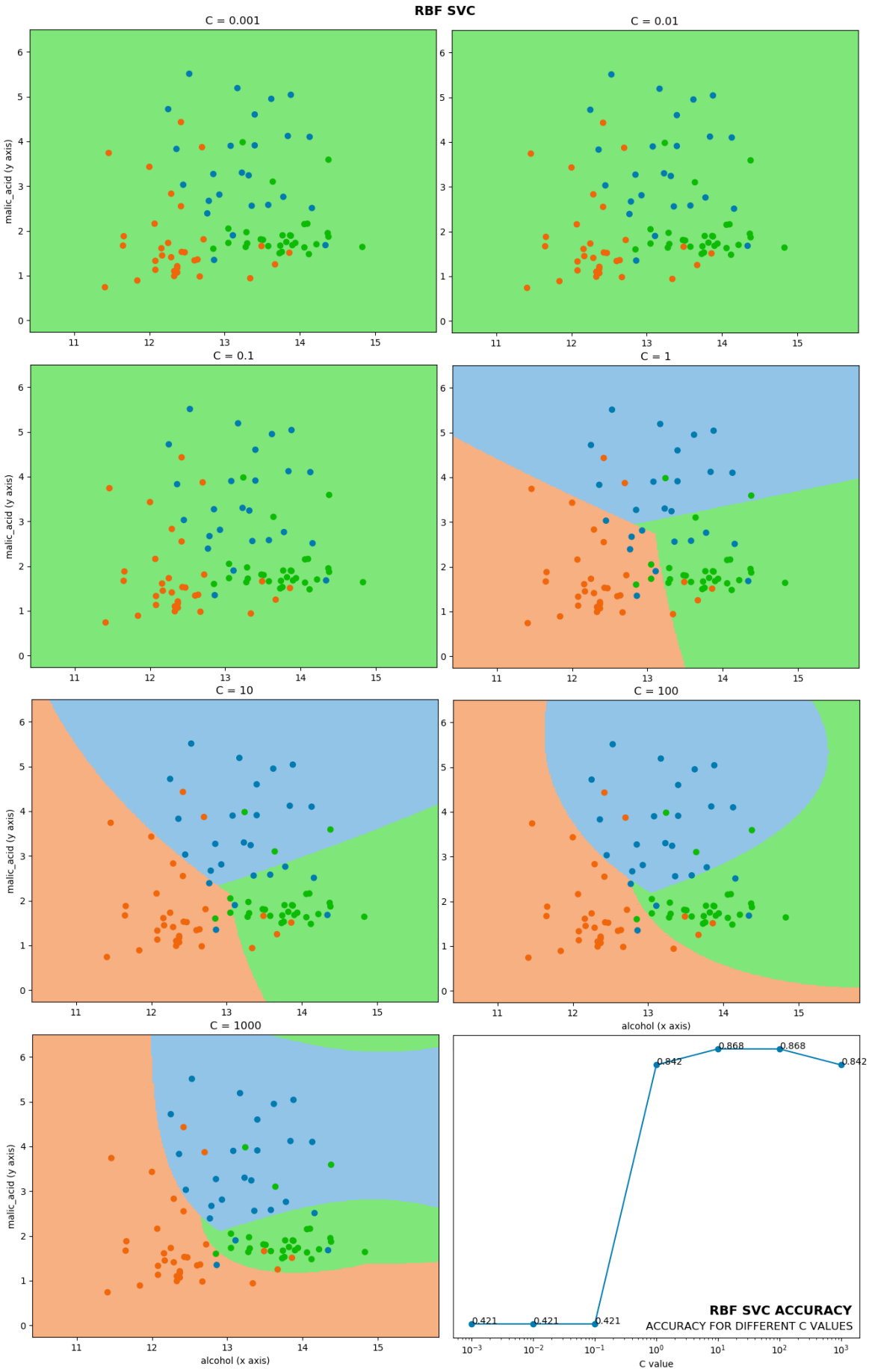

Figure 5 Decision boundaries and validation accuracy when C changes for SVM rbf kernel

2.1. K-Fold Cross Validation

The two following plots are generated after searching the best parameters for C and Gamma using an SVM with “rbf” kernel trained on the training set and evaluated on the validation set.

Gamma parameter defines how far the influence of a single training sample reaches, with low values meaning ‘far’ and high values meaning ‘close’. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

For Gamma I chose the value of “1/n_features” and the same value one order of magnitude higher and on order lower, for C the same values used previously.

The two images differ since in the second case a 5-fold cross validation has been used to search for the best parameters while in the first no cross validation has been used.

While the first method is dependent on how the data is split into train and test sets, cross validation gives the model the opportunity to train on multiple train-test splits that gives better indication of how well the model will perform on unseen data.

Given these partitions, SVM with “rbf” kernel, C=1 and gamma=0.5, trained on both training and validation set, has an accuracy score on the test set of 0.778, while SVM with “rbf” kernel, C=10 and gamma=0.05, trained with cross validation, has an accuracy score on the test set of 0.741

Usually I should expect a more accurate model using cross validation, thing that casually did not happen in this run, probably due to the fact that the validation set used for the first model is very similar to the test set.

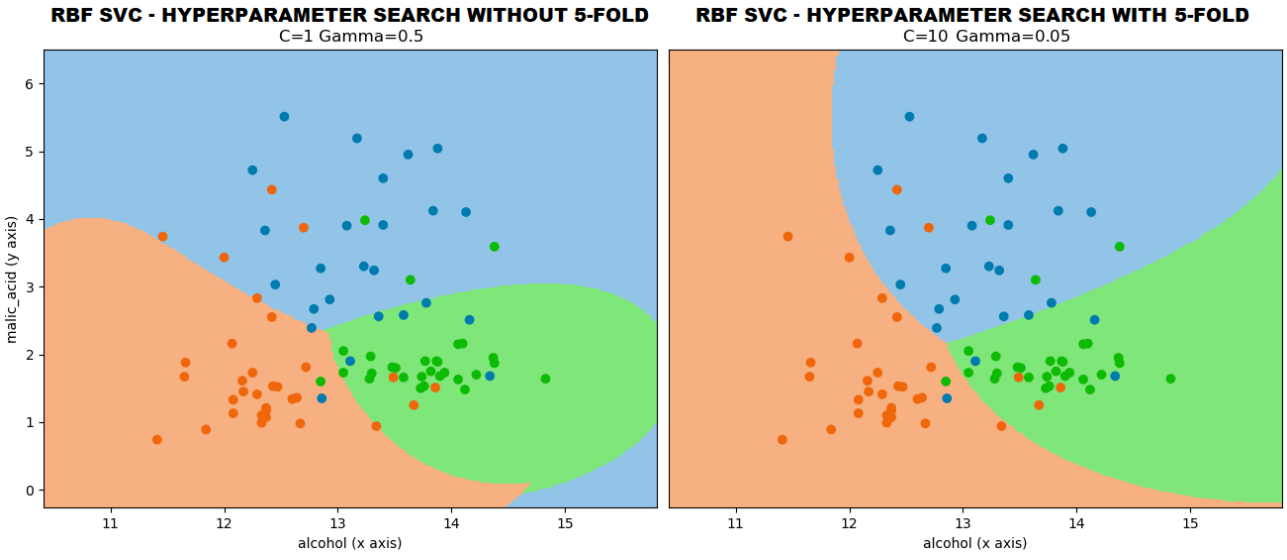

Figure 6 Decision boundaries with parameter search performed respectively without and with Cross Validation

Parameter search for C and Gamma with and without Cross Validation[1]:

No CV		C						
		0.001	0.01	0.1	1	10	100	1000
γ	0.005	μ=0.298 σ=0.135	μ=0.298 σ=0.135	μ=0.298 σ=0.135	μ=0.377 σ=0.080	μ=0.781 σ=0.040	μ=0.798 σ=0.030	μ=0.842 σ=0.026
	0.05	μ=0.298 σ=0.135	μ=0.298 σ=0.135	μ=0.360 σ=0.092	μ=0.816 σ=0.026	μ=0.860 σ=0.015	μ=0.842 σ=0.045	μ=0.816 σ=0.000
	0.5	μ=0.298 σ=0.135	μ=0.298 σ=0.135	μ=0.816 σ=0.026	μ=0.860 σ=0.030	μ=0.833 σ=0.030	μ=0.833 σ=0.030	μ=0.807 σ=0.054
	5	μ=0.298 σ=0.135	μ=0.298 σ=0.135	μ=0.509 σ=0.160	μ=0.816 σ=0.053	μ=0.781 σ=0.040	μ=0.719 σ=0.040	μ=0.719 σ=0.054

With CV		C						
		0.001	0.01	0.1	1	10	100	1000
γ	0.005	μ=0.472 σ=0.151	μ=0.472 σ=0.151	μ=0.472 σ=0.151	μ=0.574 σ=0.105	μ=0.759 σ=0.062	μ=0.779 σ=0.044	μ=0.790 σ=0.032
	0.05	μ=0.472 σ=0.151	μ=0.472 σ=0.151	μ=0.518 σ=0.148	μ=0.769 σ=0.041	μ=0.805 σ=0.015	μ=0.800 σ=0.015	μ=0.800 σ=0.015
	0.5	μ=0.467 σ=0.142	μ=0.467 σ=0.142	μ=0.790 σ=0.023	μ=0.800 σ=0.018	μ=0.795 σ=0.062	μ=0.790 σ=0.058	μ=0.769 σ=0.053
	5	μ=0.456 σ=0.124	μ=0.456 σ=0.124	μ=0.620 σ=0.047	μ=0.805 σ=0.058	μ=0.749 σ=0.049	μ=0.728 σ=0.062	μ=0.718 σ=0.084

3. KNN vs SVM

KNN is one of the simplest classification algorithms available for supervised learning, its idea is to search for closest match of the test data in the feature space. The decision boundaries achieved with this algorithm are much more complex than the ones found with SVM, especially in the case data is not linearly separable, thus obtaining a nice classification. However, outliers can significantly kill the performance and the algorithm is more likely to overfit. As the complexity of the space grows, the accuracy of K-NN comes down and we would need more data.

The SVM is a classifier based on the separation of the data by hyperplanes and due to its optimal nature, it is guaranteed that the separated data would be optimally separated.

The results and observations show that SVMs are more reliable classifiers. However, KNN is less computationally intensive than SVM.

[1] Validation Accuracy is the result of three runs. It is therefore expressed by expected value and standard deviation.

4. Extra

In order to try different features, instead of choosing randomly, I'll take the pair of features with the best silhouette score and the pair with the worst score, to do so I'll calculate the silhouette for all the pairs of features, avoiding repetitions. After that I'll train KNN, LINEAR SVM and RBF SVM on the training set, tuning on the validation set, evaluating on the test set and plotting the decision boundaries along with the training points only for the promising value of the parameter.

4.1. Best Features

Silhouette score is a number between +1 and -1 and tells how clusters are well separated from one another and how close to its centroid each cluster is.

The best pair of features is alcohol and flavonoids with silhouette score of 0.434

In the case of this first pair of features the situation is the best possible therefore the classifiers score high values in accuracy on the test sets.

KNN with K=3 scoring in accuracy 0.944

SVC with “linear” kernel and C=100 scoring in accuracy 0.889

SVC with “rbf” kernel and C=1000 scoring in accuracy 0.889

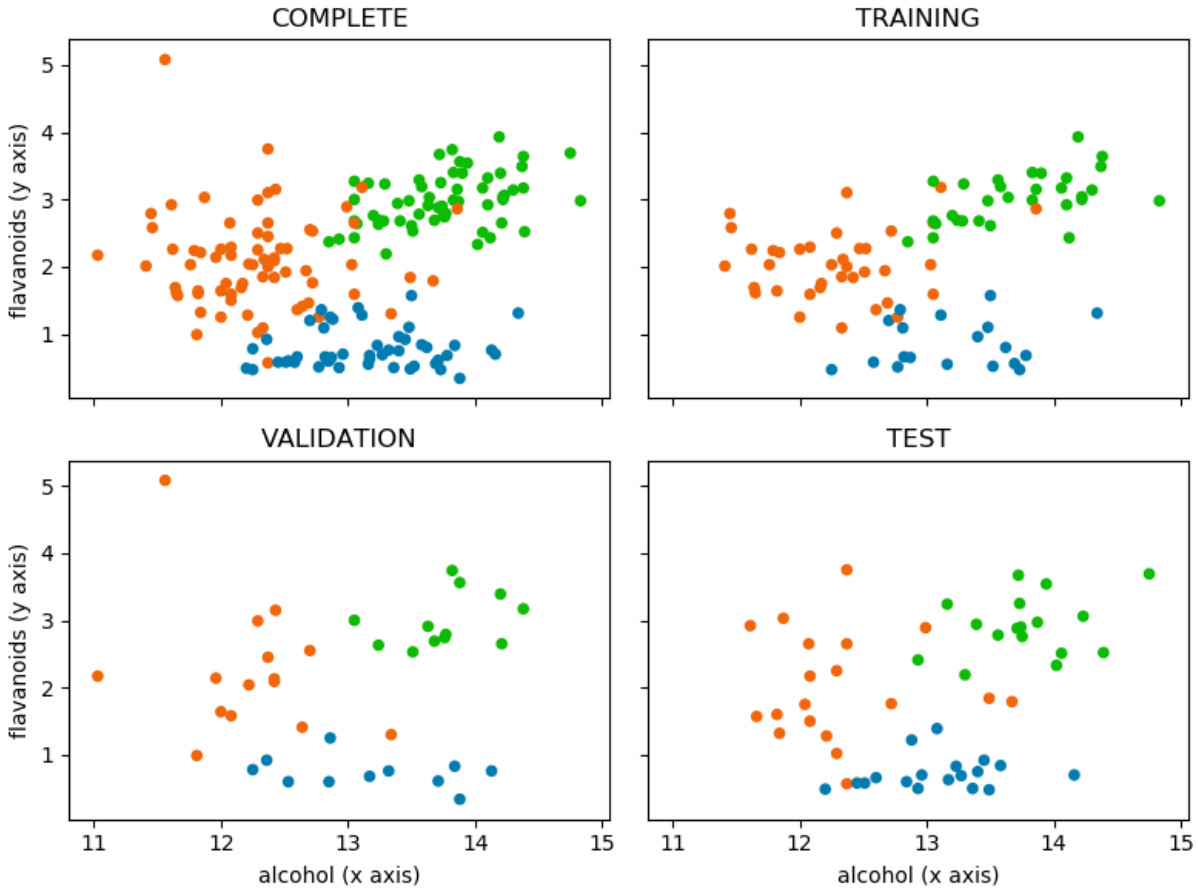

Figure 7 2D dataset using the best features with respect to silhouette score

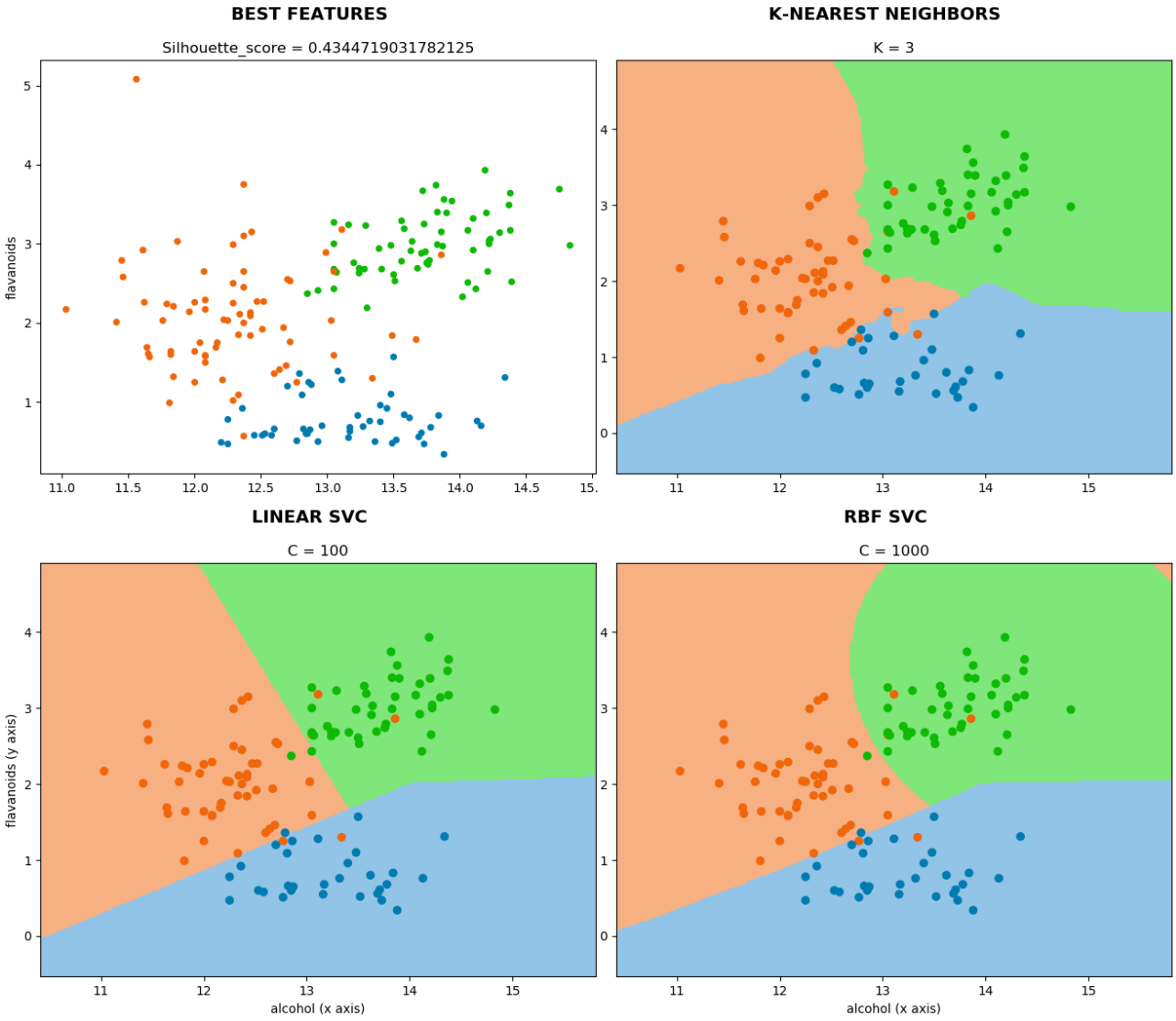

Figure 8 Decision boundaries for best features

4.2. Worst Features

The worst pair of features is ash and magnesium with silhouette score of -0.021

Being the situation the worst possible the classifiers are not able to reach an acceptable.

Test accuracies:

KNN with K=3 scoring in accuracy 0.500

SVC with “linear” kernel and C=100 scoring in accuracy 0.630

SVC with “rbf” kernel and C=1000 scoring in accuracy 0.593

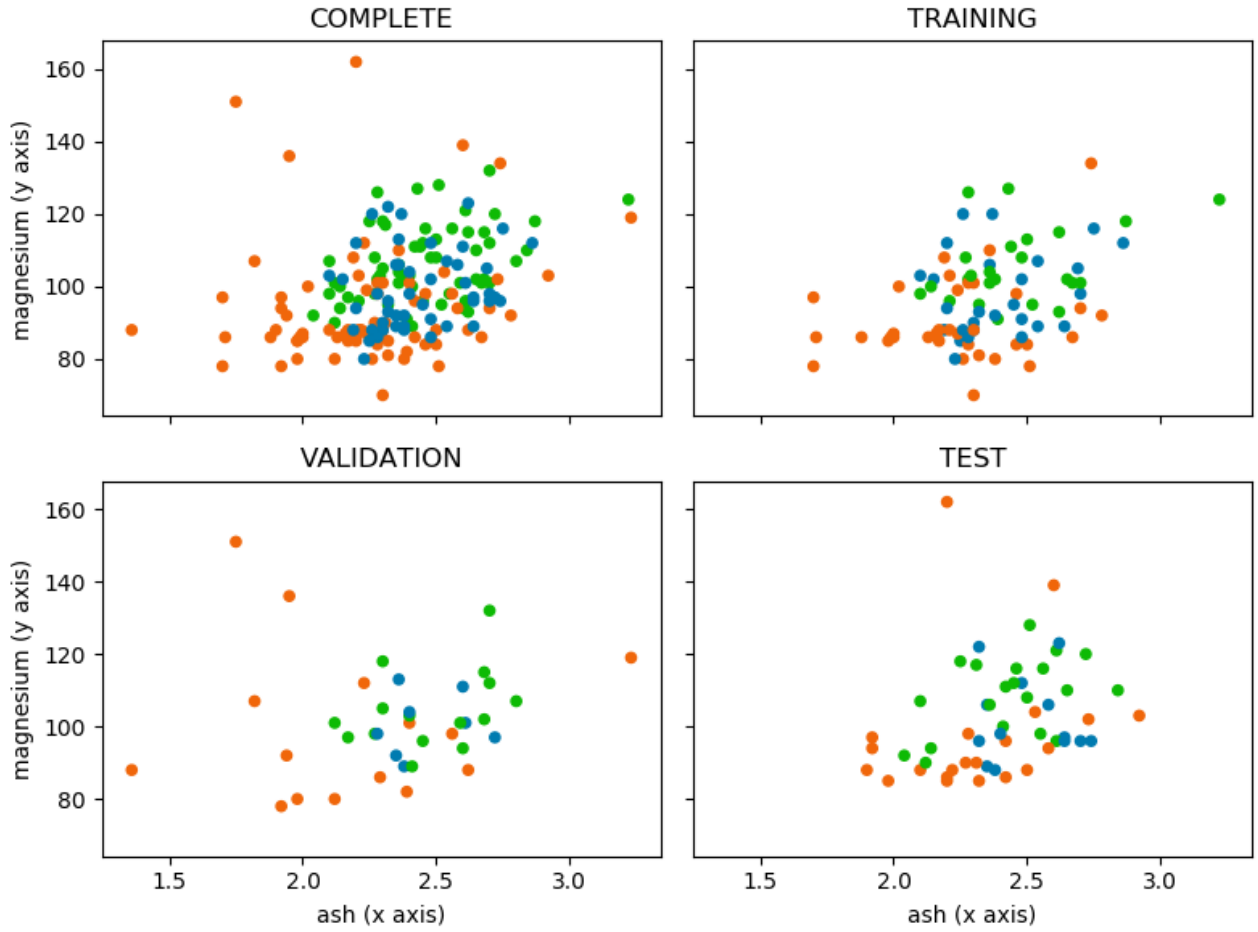

Figure 9 2D dataset using the worst features with respect to silhouette score

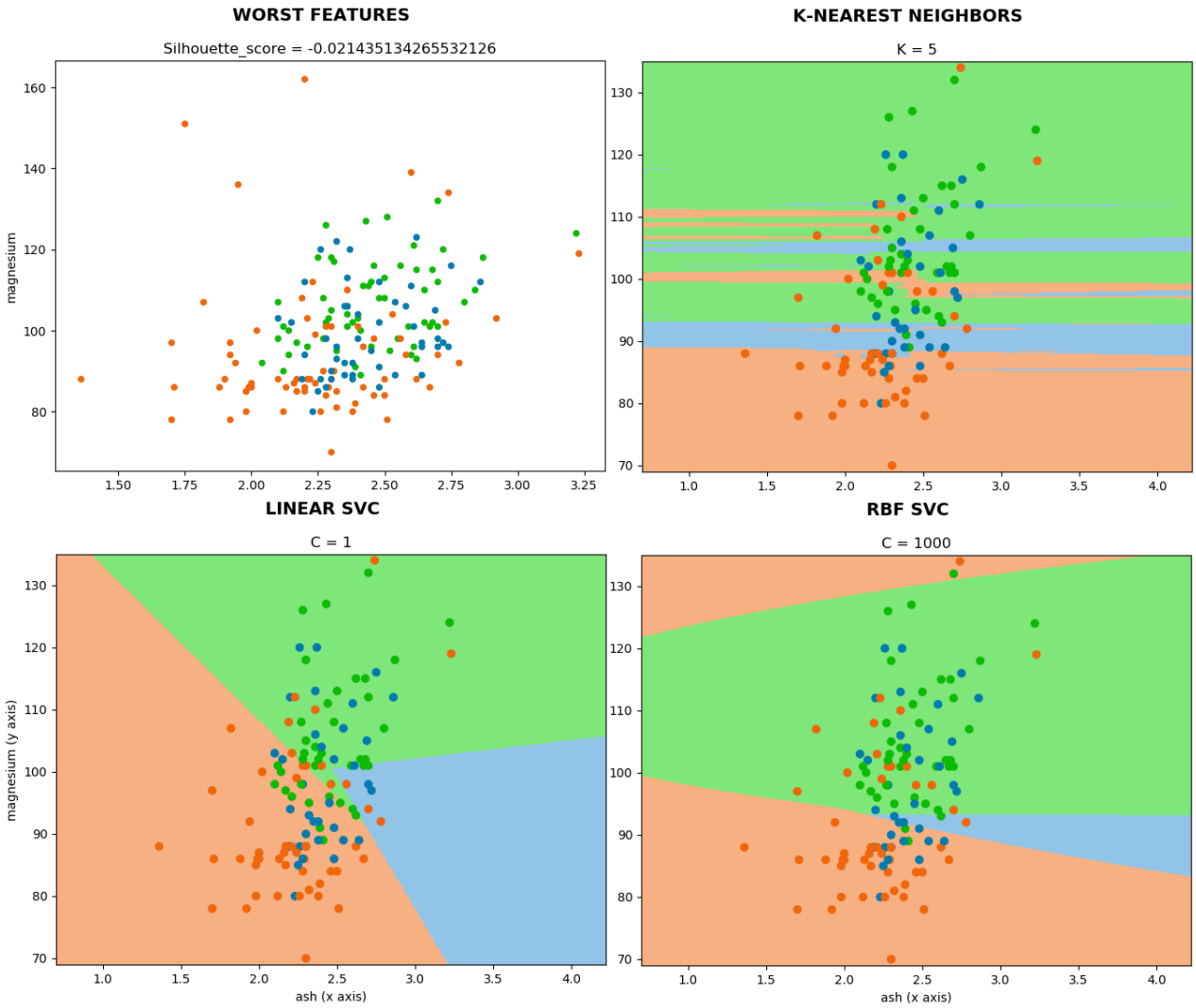

Figure 10 Decision Boundaries for worst features