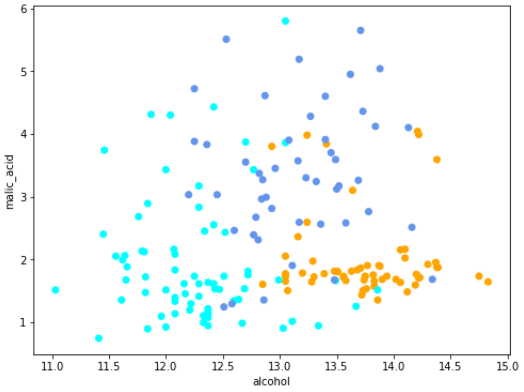
Report

DATASET

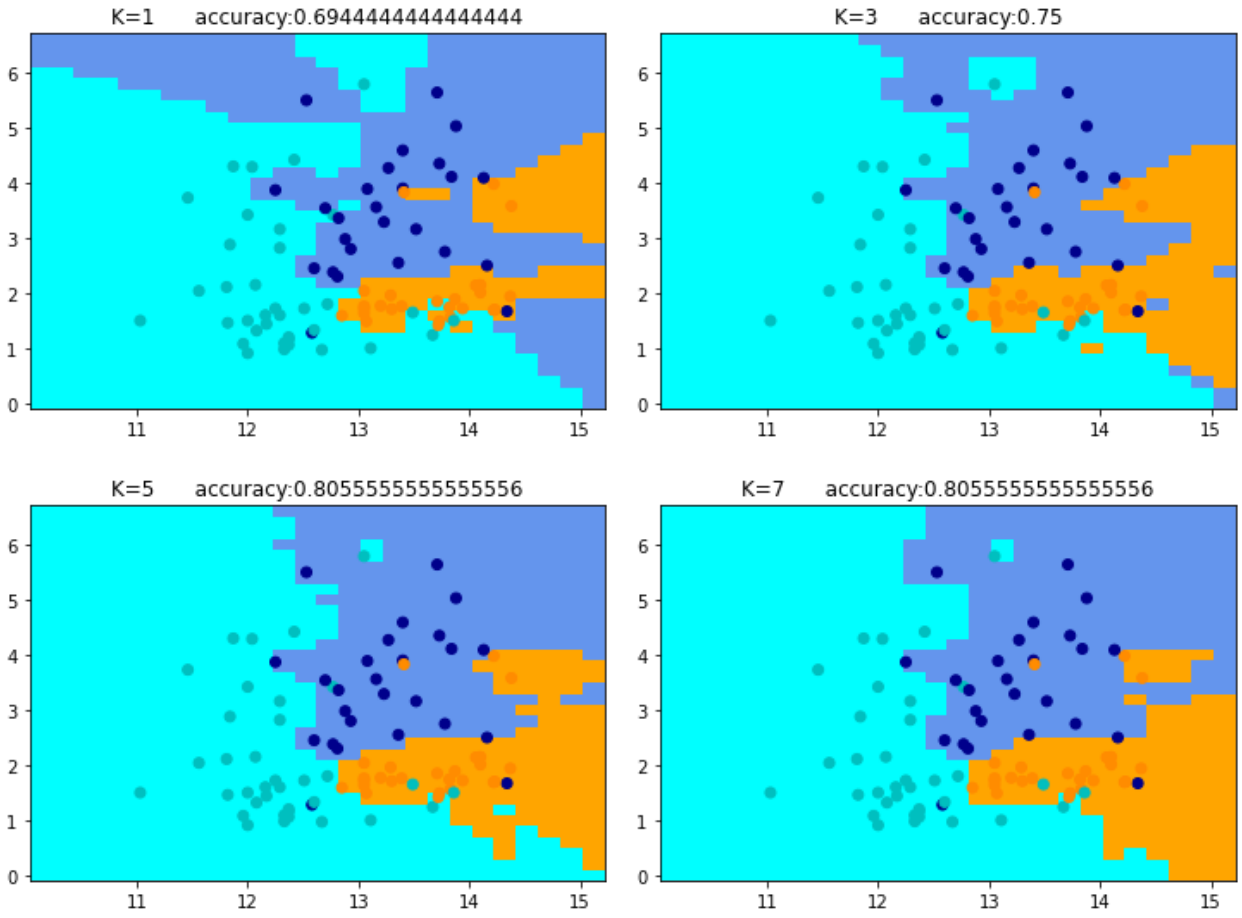
For this project the wine dataset will be analyzed and used to train multiple classification models.
After importing the dataset is possible to see that in addition to data there is a section named "description" which allows to obtain many informations about the content.
There are 178 instances in total and each instance is defined by 13 numeric attributes, representing chemical characteristics of the sampled wine.
The number of possible classes is equal to 3 and the distribution is: 59 for the first, 71 for the second and 48 for the third.
No missing values are present so it's possible to take all the 178 instances without deleting anything.
For the task only the first two attributes will be used (alcohol and malic acid) so, in order to simplify the manipulation of data, from the dataset a dataframe is build and only those attributes are taken into account. Plotting the training set in a two dimensional space allows to realize how the data are distributed and 3 big clusters can be easily spotted.



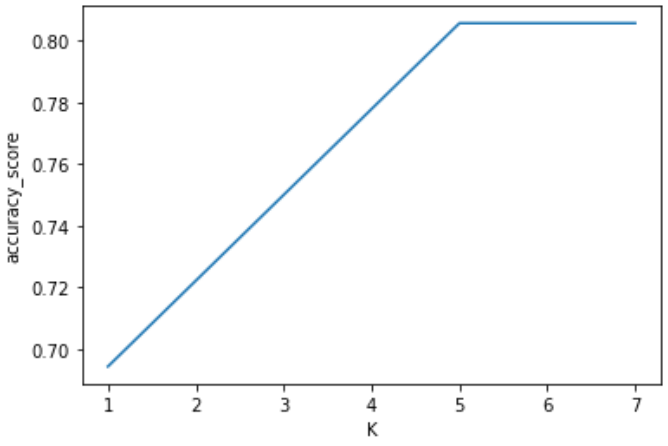
Now the dataframe is divided in train,validation and test with the following percentages: 50, 20, 30.
Obviously is not possible to reach those values (178 instances) so the obtained numbers are an approximation:
%train= 0.4943820224719101, %val= 0.20224719101123595, %test= 0.30337078651685395.

K-NN

This is the first model that will be used. In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor in fact, when the model have to classify a point it looks for k closest point and then classify points by majority vote of its k neighbors.
Using K equals to [1, 3, 5, 7] the obtained boundaries have those shapes:



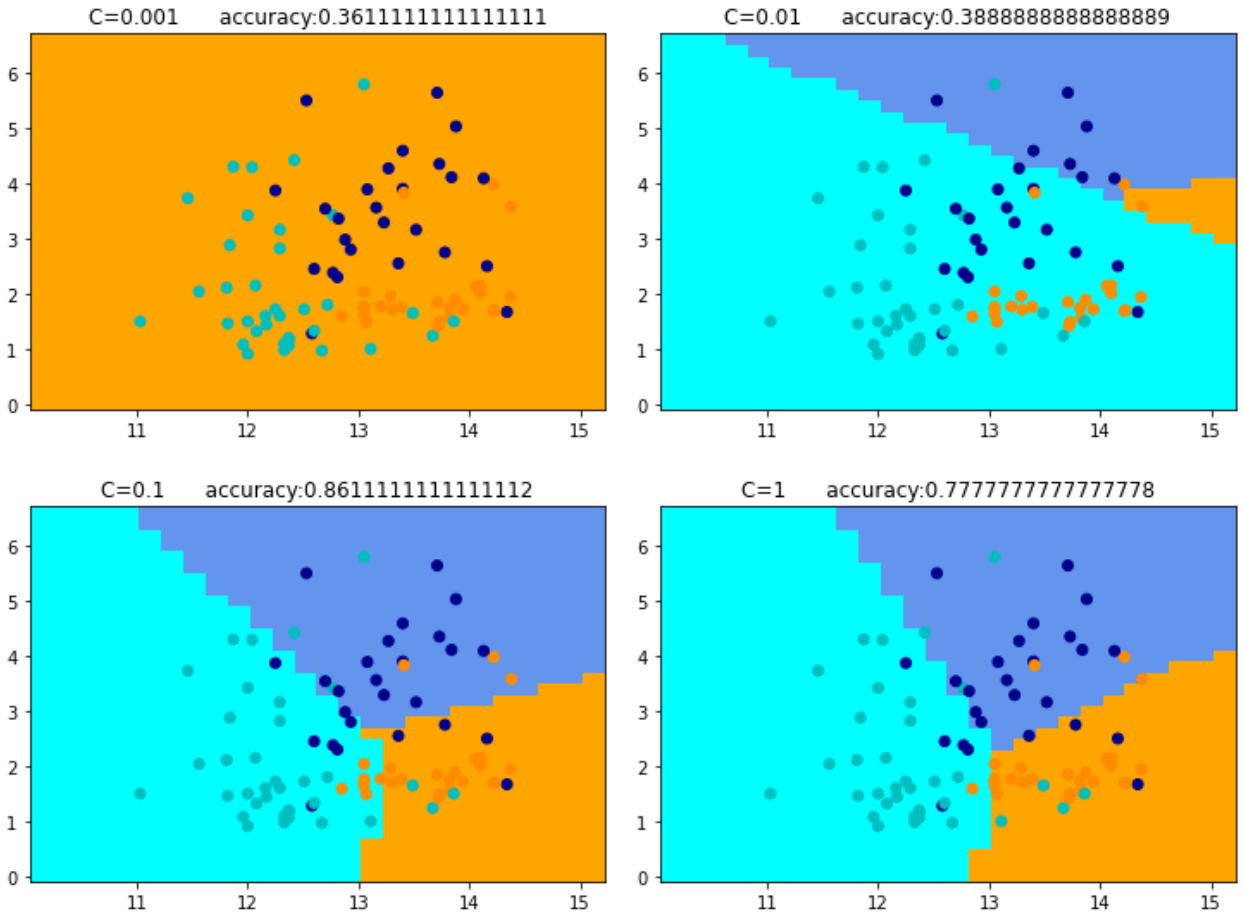
Generally, with a low K the noise will have an higher influence on the model and a big K leads to high computational cost. In this case the clusters don't overlap too much but the presence, in a certain cluster's region, of samples belonging to other clusters is not so low. If those points are considered as noisy points, a better accuracy is predictable when the K is high.

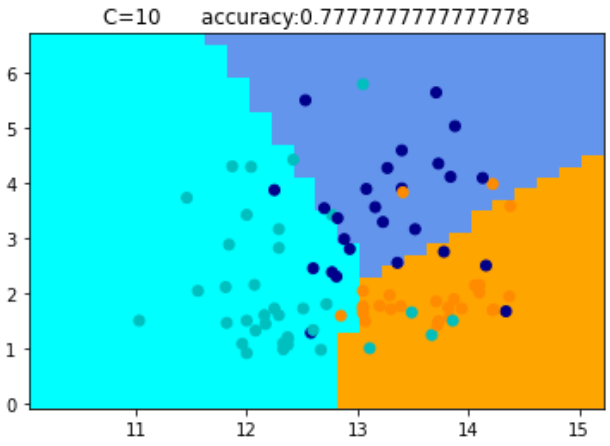


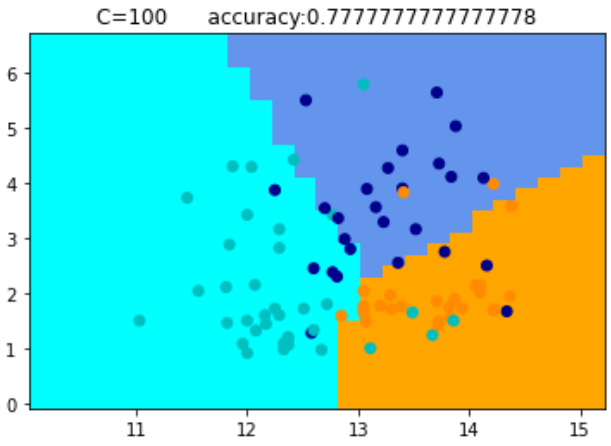
This graph confirms what has been previously said in fact, the more K increases the better accuracy is.
At some point (K=5, K=7) accuracy saturate and there's no need to take an higher K.
Taking the classifier with K=7 and evaluating it on the test set, the achieved accuracy is 81.4%.
The difference of accuracy between test and validation set can be quite big but those obtained differ only for 0.09%.

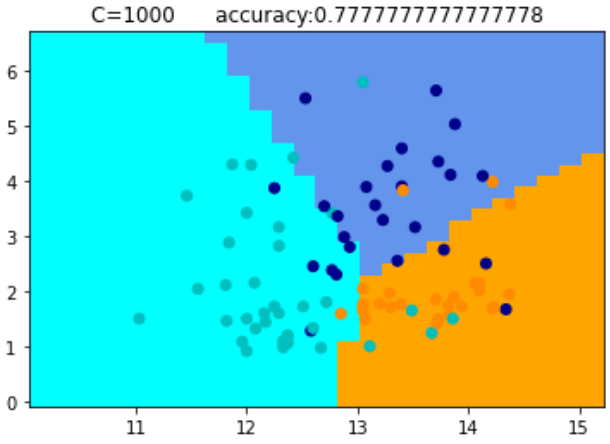
LINEAR SVM

The Support Vector Machine is a supervised learning algorithm usually used for classification.
The distance between the hyperplane and the nearest data points (support vectors) is known as the margin. The goal is to choose a hyperplane with the greatest possible margin, giving a greater chance of new data being classified correctly.
The algorithm works if the points are linearly separable, this is because the constructed hyperplane is a linear separator.
If the data are not linearly separable SVM works by mapping data to a high-dimensional feature space so that data points can be categorized using a linear separator.
The mathematical function used for the transformation is known as the kernel function and SVM supports the following type: linear, polynomial, radial basis function (RBF), sigmoid.
In addition to the kernel is possible to set the C.
It controls the trade off between smooth decision boundary and classifying training points correctly.
A large value of C means you will get more training points correctly. A smaller C will allow more errors and margin errors and usually produce a larger margin. When C goes to Infinity, SVM becomes a hard-margin.
Using the linear SVM from sklearn the kernel is set to “linear” and it cannot be modified.
The standard parameters used for the penalty is “l2” and “hinge” for the loss function.
For the experiment the range of values tried for C is [0.001, 0.01, 0.1, 1, 10, 100, 1000].
As it is possible to see the data don’t look linearly separable and on the frontier the clusters are a little bit mixed. Given this, a relatively large margin (low C) can lead to better accuracy.

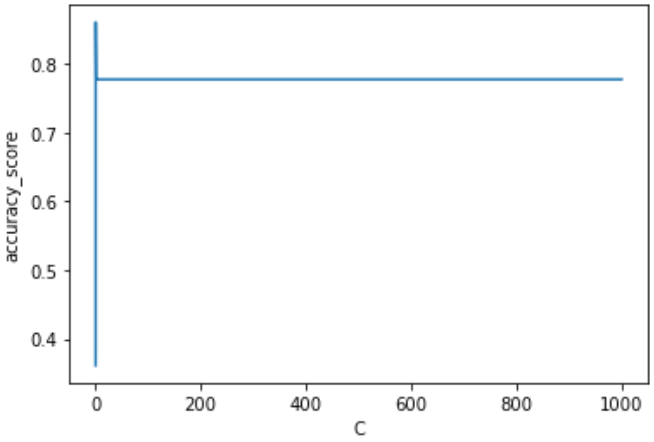








The graphs clearly shows how C determines the decision boundaries. In the first one all validation points are predicted as orange (first class), this is due to the fact that C is too low and allows too many errors. This leads to reserve all the 2D dimensional space to the orange class.
As C increase there's an improvement and an optimum is reached for C=0.1 with an accuracy of 0.86%. Higher values of C are not so efficient because, as mentioned before, set an high tolerance and really few points are allowed to be misclassified.

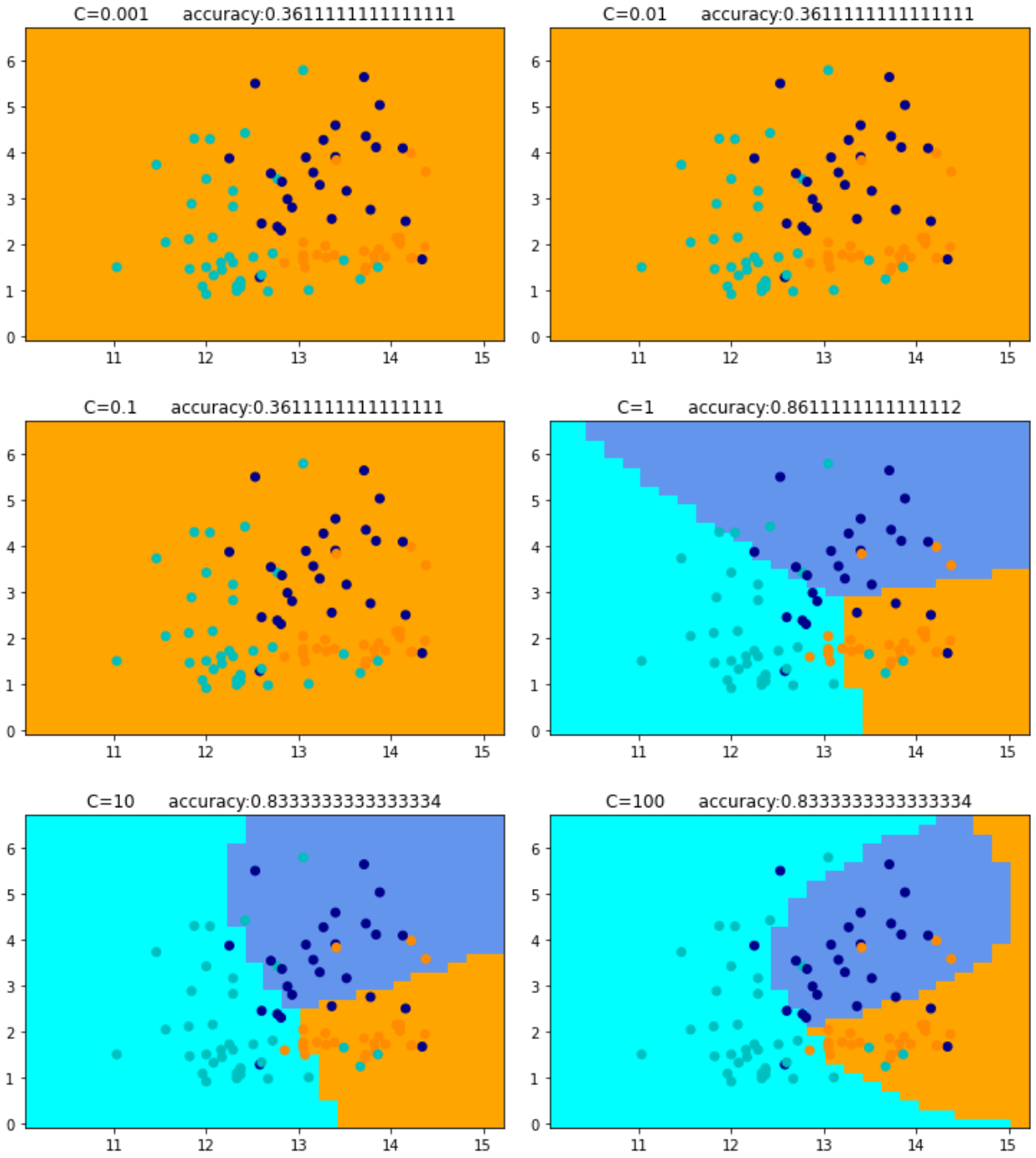


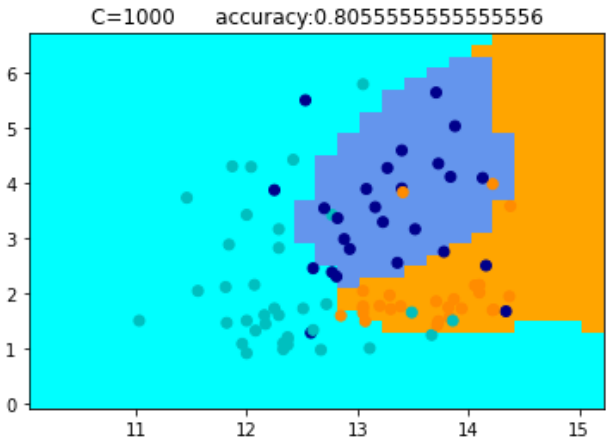
On the test set the achieved accuracy for the model with C=0.1 is 79%.
Given the nature of the data this level of accuracy was expected.

RBF KERNEL

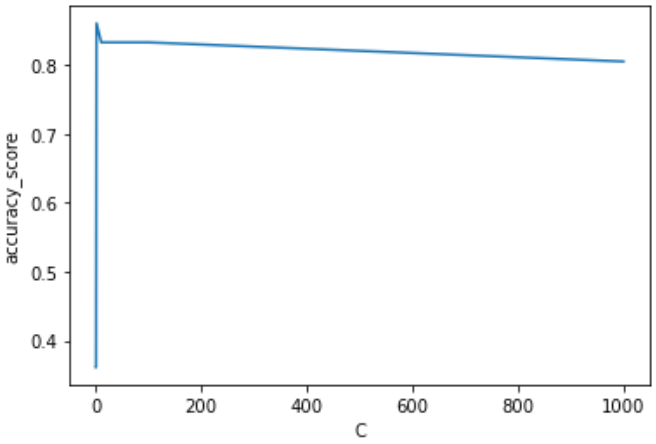
Now the kernel parameter is set to 'RBF', with this function the training data will be mapped in a different dimensional space which allows to obtain a linear hyperplane. When the decision boundaries are plotted to the original dimensional space the shape will be totally different form the

linear case. In this case this type of kernel is preferred over linear in fact the shape of data recommend the usage of a non-linear separator.
It is time to introduce another hyperparameter that characterizes the SVM method, the gamma. Gamma decides how much curvature we want in a decision boundary. High gamma means more curvature.
For this first test the hyperparameter will be set to the default value (the framework will decide for itself which value to keep).





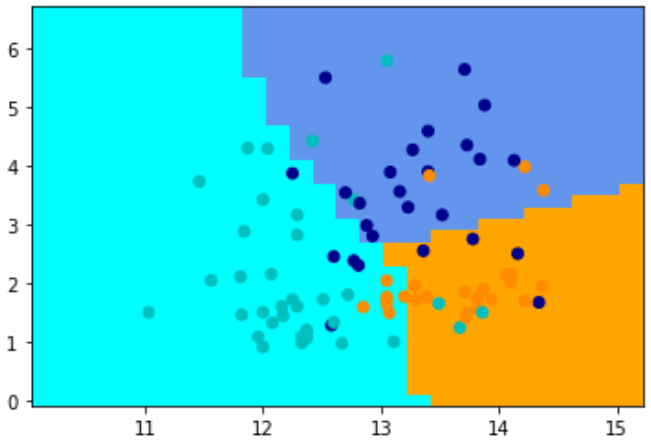
The situation is similar to the linear kernel for low values of C but suddenly the accuracy increase reaching an optimum with C=1. It looks to better than the linear case. This is due to the curvature of the boundaries that are best suited to the shape of the data.



Even if the accuracy on validation is quite big, the accuracy on the test set is 75.9%.
This leads to the following conclusion: gamma has to be optimized for the specific case.

GRID SEARCH

In order to improve the performance with the “rbf” kernel a grid search over gamma and C is applied. After deciding which range to use for both gamma and C the grid search will try all the possible combination between those hyperparameters to get the best model.
The dataset contains very few instances so is allowed to perform a grid search using a wide range for the hyperparameters, the computational cost of this operation is reasonable.
Using C=[0.001, 0.01, 0.1, 1, 10, 100, 1000] and gamma=[1, 0.1, 0.01, 0.001] the best model obtained has an accuracy of 86.1%, the best parameters use C= 10 and gamma= 0.01.



The accuracy on the test set should be at least equal to the previous test and in fact the achieved value is 79.6%.
A better performance on the test set was expected but the two solutions don't differ a lot.

K-FOLD

To further improve the accuracy on the test set a K-fold will be applied.
The training set and the validation set are now merged together with a 70% of samples for the training set and a 30% for the test set.
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. As such, the procedure is often called k-fold cross-validation.
This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. One fold is treated as a validation set, and the method is fit on the remaining k−1 folds. The method is repeated until all the k split have been used as validation.
After that, only the k-1 training fold which performed better on the validation will be used for training the model.
For this last experiment a grid search over the gamma and C will be performed using the SVM algorithm with the 'RGB' kernel. In addition to this, a K-fold with K=5 is applied over the training set.
Now, since the training set is different from the previous case, the best hyperparameter can be different. The performance of the obtained model are expected to be better respect to the non-k-fold approach.
The best parameters are C= 0.1 and gamma= 1, the accuracy on the test-set is 83.3%.

CONCLUSIONS

The first important thing to say before to draw conclusions is that this dataset is a toy dataset, it is hard to evaluate the performance of a model on a specific field using a little dataset having only 178 instances.

	accuracy on test-set
k-nn	81.4
svm	79.6
rbf kernel	75.9
grid-search	79.6
k-fold	83.3

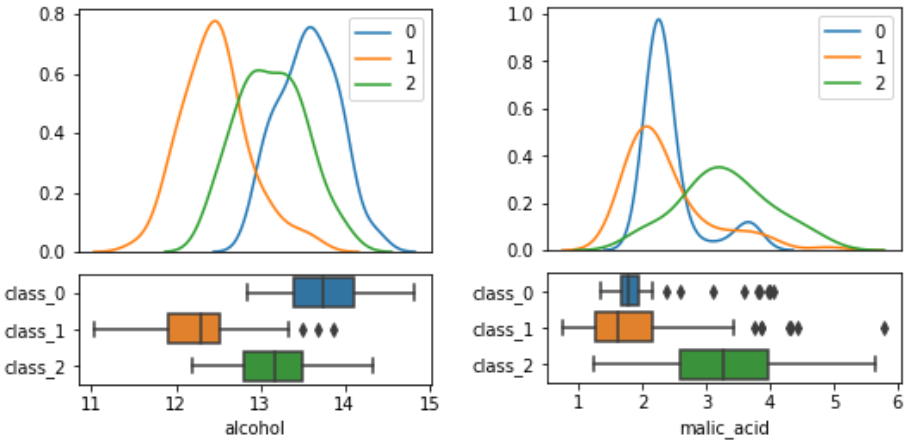
As shown in the figure above, although simple, k-nn has performed well. Probably this is due to the fact that the number of dimensions is low and the clusters have different means and high variances, in those conditions k-nn method is a good choice for the classification. The linear SVM, as expected, reach a lower accuracy because the dataset is clearly not linearly separable.
Using SVM with 'RBF' kernel, without optimizing gamma, the accuracy is the lowest but after a grid search on gamma and C and a K-fold with K=5 the accuracy grows achieving the highest value, 83.3%.

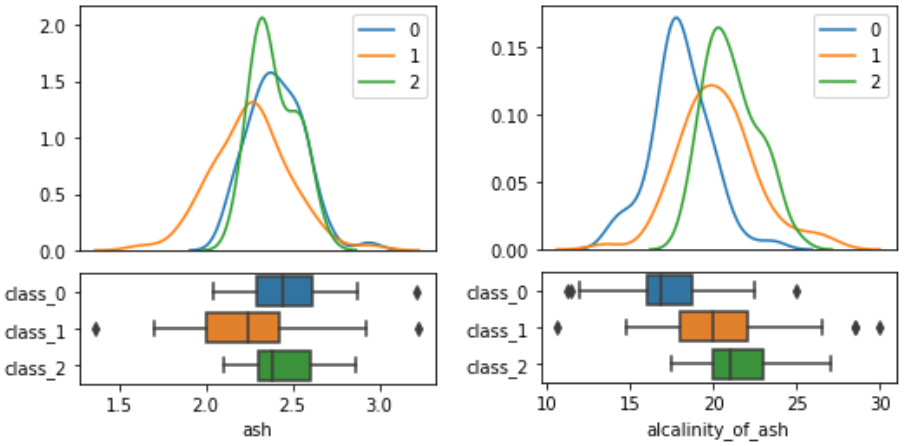
SVM VS K-NN (extra)

The KNN algorithm can compete with the most accurate models because it makes highly accurate predictions.
It does not explicitly build any model, it simply tags the new data entry based learning from historical data. Although this method increases the costs of computation compared to other algorithms, KNN is still the better choice for applications where predictions are not requested frequently but where accuracy is important.
KNN works well with small number of input variables but as the numbers of variables grow K-NN algorithm struggles to predict the output of new data point.
On the other hand SVM is accurate in high dimensional space and, for the training, only a subset of the training set is used.
The algorithm is prone for over-fitting, if the number of features is much greater than the number of samples.
Anyway, SVM is good for image analysis tasks, such as image classification and handwritten digit recognition.
Also SVM is very effective in text-mining tasks, particularly due to its effectiveness in dealing with high-dimensional data.
SVM can also be used for other types of machine learning problems, such as regression, outlier detection, and clustering.
In conclusion those two model have a different approach for the classification of data and, because of this, it is not possible to say which is better, it always depend on data.

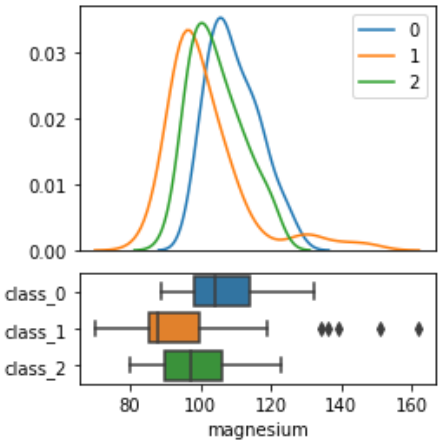
DIFFERENT ATTRIBUTES (extra)

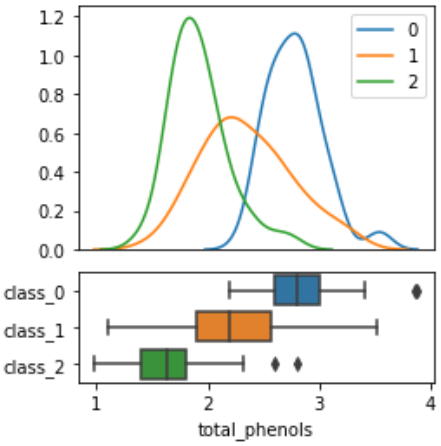
For all the precedent experiments only the first two attributes were used.
To understand if there are other couple of attributes that can distinguish data in a better way it is useful to plot graphs showing the means and the variance of the attributes for each class.

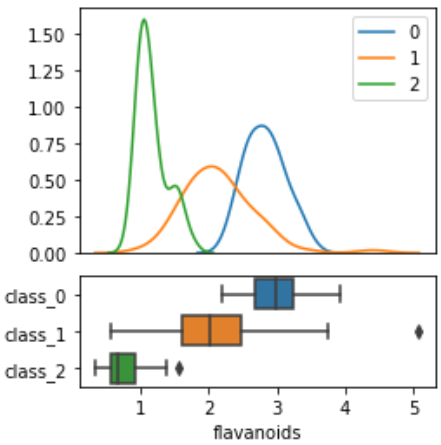


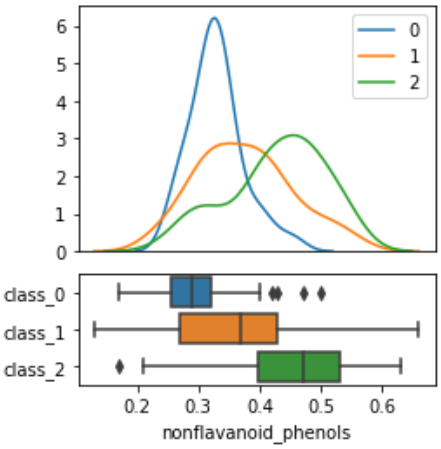

0
1
2
class_0
class_1
class_2
ash

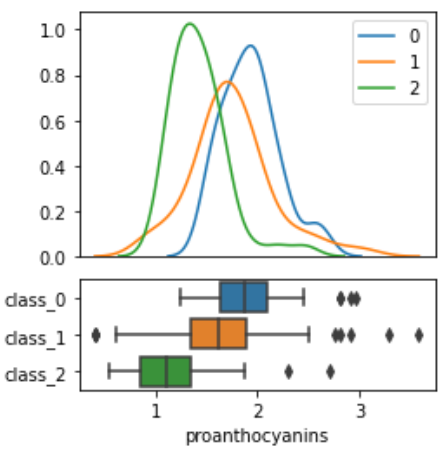
0
1
2
class_0
class_1
class_2
alcalinity_of_ash

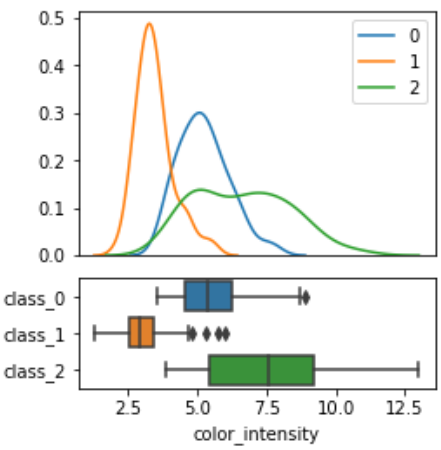

0
1
2
class_0
class_1
class_2
magnesium

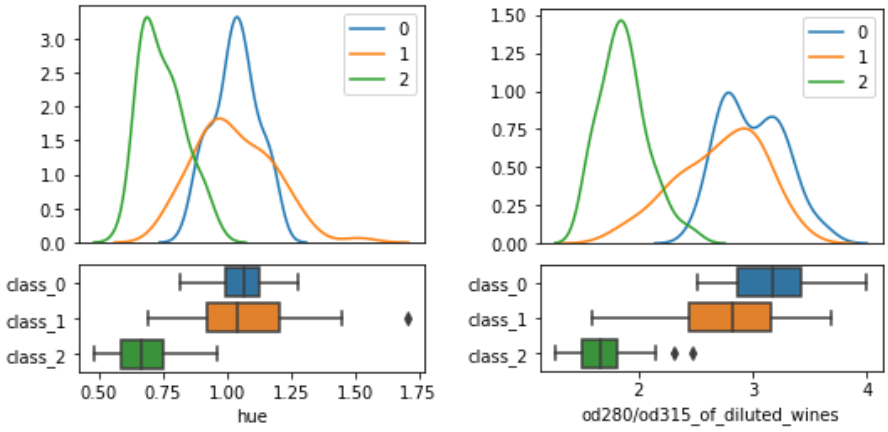

0
1
2
class_0
class_1
class_2
total_phenols

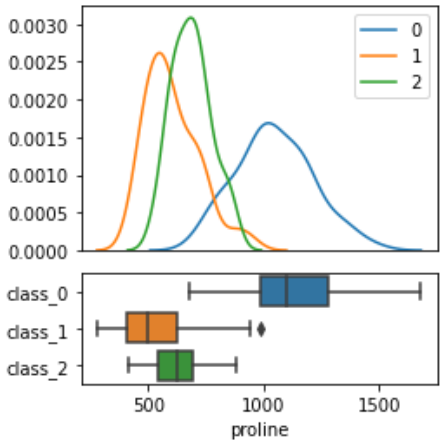

0
1
2
class_0
class_1
class_2
flavanoids


0
1
2
class_0
class_1
class_2
nonflavanoid_phenols

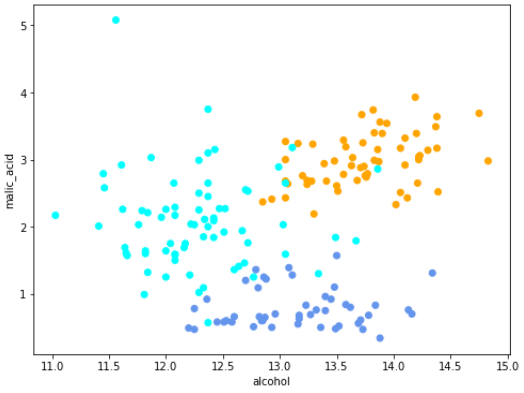

0
1
2
class_0
class_1
class_2
proanthocyanins


0
1
2
class_0
class_1
class_2
color_intensity





For some of the features such as flavanoids and total alcohol, it is clear the class distributions have quite different means.



The clusters are now widely separated and the number of noisy points is low.
Thus we could expect that if now we use alcohol and flavanoids the accuracy can be improved of the previous models.

	accuracy on test-set
k-nn	0.925926
svm	0.944444
rbf kernel	0.944444
grid-search	0.944444
k-fold	0.962963

As shown, the average improvement is 10%.
This result is important because asserts how a deeper study of the data can lead to a big improvement of the model's accuracy.