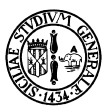
****

**UNIVERSITÀ DEGLI STUDI DI CATANIA**

DIPARTIMENTO DI ECONOMIA E IMPRESA

**CORSO DI LAUREA IN DATA SCIENCE FOR MANAGEMENT**

|  |
| --- |
|  |

**Report on “wine” dataset**

**Casella Bruno**

**1000014143**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
| **Advanced Machine Learning** |

**ACADEMIC YEAR 2020 – 2021**

CLASSIFICATION USING SCIKIT-LEARN

I made a comparison of different classifiers using the “wine” dataset contained in Scikit-Learn.

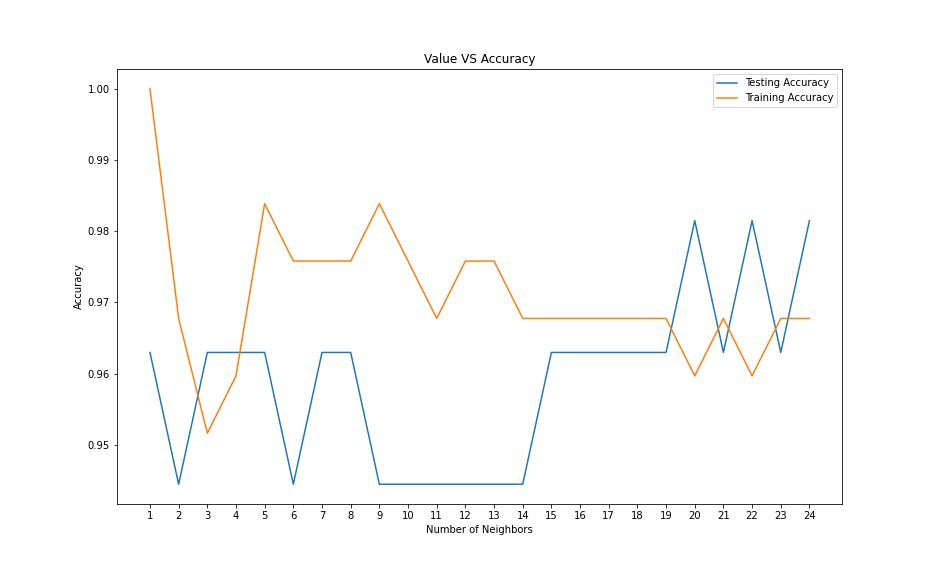
The dataset contains 13 attributes, that are the constituents found in each of the three types of wine. There are 178 instances, no missing values, so in a classification context, this is a well posed problem with well-behaved class structures, indeed the class\_0 contains 59 instances, class\_1 71 and class\_2 48.

Firstly, I putted the data into a dataframe, in order to organize by columns and then I added the target variable (the classes). So, the shape of the dataframe is 178 rows and 14 columns.

I splitted the set into training set and test set. I gave 70% of data to the training set and 30% to the test set. After the splitting, I scaled the data because many algorithms assume that all features are centered around zero and have approximately the same variance. I scaled after the splitting to avoid data leakage (when information from outside the training set is used to create the model).

After these preliminary steps, I started to fit the models.

The first family I analyzed was the K-Nearest Neighbors Classifiers. Firstly I used the KNeighborsClassifier() function without changing any parameter. By default, it has 5 as number of neighbors and weight equal to ‘uniform’. I obtained a score around to 0.962. Using the classification\_report() function, I printed out the precision (the ratio tp/(tp + fp)), the recall (the ratio tp/(tp+fn)), the f-score (that depends on both precision and recall) and the support (the number of occurrences of each class in the true values of the test set). I repeated the same process using n\_neighbors = 3, obtaining the same results. Then I showed the testing and training accuracies in a graph, varying the number of neighbors from 1 to 25.



The best result was in correspondence of 20 neighbors, so again, I made the same steps of before, using n\_neighbors=20, and I saw that the score increased to around 0.981.

For this best case, I showed also the decision boundaries for each class. I took only the first 2 features, Alcohol and Malic\_Acid. I showed the boundaries using the weights parameter ‘uniform’ and then ‘distance’.

I tried also with the RadiusNeighborsClassifier() with a radius of 4.0, and I obtained a score of 0.944.

Next family of Classifiers: Random Forests. I used the default number of estimators (the number of trees in the forest) that is 100. The criterion (the function to measure the quality of a split.) I used is the Gini (by default). Another option is "entropy". I didn't specify the maximum depth of the tree, then the nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples. I obtained an accuracy of 1.0 because I used too many estimators. The accuracy remains 1.0 also lowering the number of estimators of a lot. A first decrease in the accuracy is when n\_estimators=41 or lower.

Then I fitted the DecisionTreeClassifier() using the same parameters of the Random Forest Classifier, but obviously in this case I have only a tree. The score obtained was about 0.944.

For the Decision Tree I plotted also the decision surfaces using paired features and then I plotted the tree.



The next classifier is the ExtraTreesClassifier. Also here, I used the same parameters of the Random Forest. The score obtained was 1.0.

Now, I showed the decision boundaries of different classifiers. The plot compares the decision surfaces learned by a decision tree classifier (first column), by a random forest classifier (second column) and by an extra-tree classifier (third column).

In the first row, the classifiers are built using the Alcohol and the Malic Acid features only,

on the second row using the Alcohol and Ash only, and on the third row using the Ash and the Alcalinity of Ash only.

I use parameters to obtain the best results (max\_depth=None for DecisionTrees and n\_estimators = 100 for Random Forests and ExtraTrees).

I obtained perfect results (score=1.0) in all the previous cases. So, I varied the max\_depth for the DecisionTreeClassifier (max\_depth=5), and the n\_estimators for Random Forest and Extra Trees (n\_estimators=30). Obviously, in some cases, the score was lower.

At the end, I tried the family of Naive Bayes Classifiers, starting with the Gaussian Naive Bayes. I obtained a score of 1.0. Then I tried with the Bernoulli Naive Bayes, obtaining a score of about 0.962.

Finally, a table that summarizes the scores of the different classifiers. In blue the best results.

