MLAI - HOMEWORK 1

Nearest Neighbors, Linear SVM, SVM with RBF Kernel

Teaching assistants: Frattin Fabio and Lorenzo Bonasera

STEP 0 - QUICK RECAP

KNN

- Given a training set $S = (x_1, y_1),...,(x_m, y_m)$, k-NN generates a classifier $h_{k\text{-}NN}$ such that $h_{k\text{-}NN}(x)$ is the label y appearing in the majority of the k points $x_t \in S$ which are closest to x
- k-NN is a family of algorithms, one for each value of K

SVM

- Hard margin SVM for linearly separable problems
  - maximization of the margin
- Soft margin when not linearly separable
  - addition of slack variables and a penalty parameter ***C***
  - mapping to an higher dimensional space through a **Φ** function
- **Kernel** trick to easily compute inner products into higher dimensional space
  - in this homework you will use the **RBF** kernel (**gamma** parameter)
  - https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

$$k(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}$$

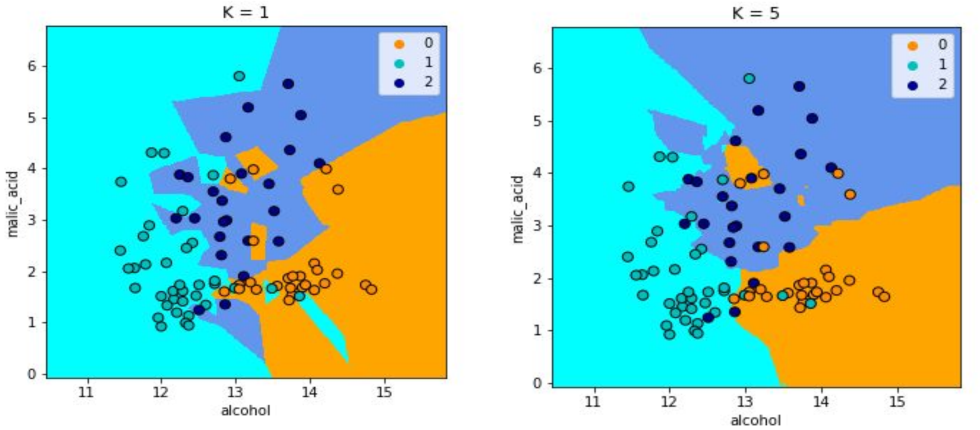
STEP 1 - KNN

WHAT YOU SHOULD DO:

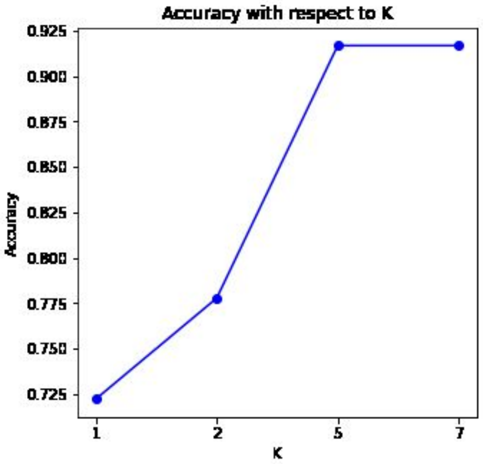
1. Load **Wine** dataset (scikit library)
2. Select **ONLY 2** attributes (the first 2, for example, but feel free to try with different pairs)
   a. **extra**: understand, by looking at the distribution of the data in the chosen 2D, which classification method could have good performances and why.
3. Split into train, validation and test sets (suggested proportion 5:2:3)
4. For different values of K (example: [1,3,5,7]):
   a. apply K-NN
   b. plot data and **decision boundaries**
   c. evaluate on validation set
5. Inspect the results:
   a. plot a graph showing **how the accuracy varies for different value of K**
   b. plot the boundaries for each value of K. How do they change and why?
6. Use the best value of K on the test set and evaluate the accuracy.

WHAT YOU SHOULD GET:

Decision boundaries



Evaluating K

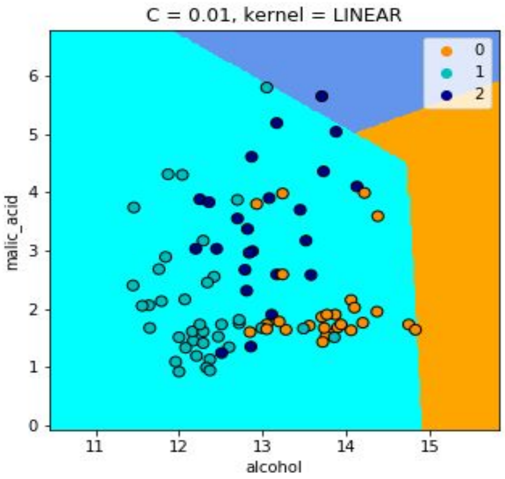


STEP 2 - LINEAR SVM

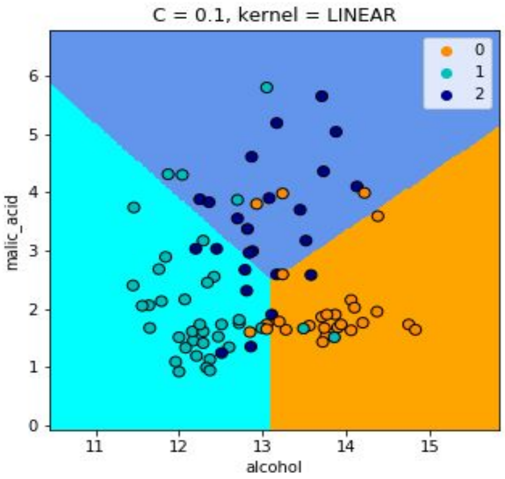
WHAT YOU SHOULD DO:

1. Keep the same data you used before (same features, same split)
2. Repeat the same steps you did before, this time varying the penalty parameter **C** of the **SVM** with **linear kernel**:
    a. example values: C = [0.001, 0.01, 0.1, 1, 10, 100, 1000]
3. Carefully inspect the decision boundaries while varying C, keeping in mind the idea of **soft-margin**:
    a. how does the value of C affects the boundaries?
    b. what happens when C is very low? What about when it is very high?
4. Inspect the *decision_function_shape* parameter
    a. what is its default value? Is it consistent to the results you have obtained?
    b. Try also with the **one-versus-one** policy: **what happens "behind the scenes"**? Are the results different? Why?

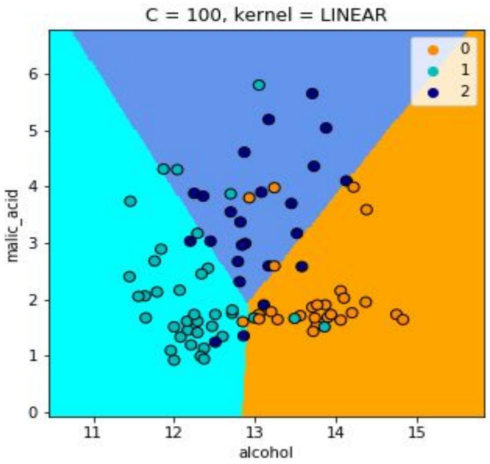
WHAT YOU SHOULD GET:



C = 0.01, kernel = LINEAR



C = 0.1, kernel = LINEAR



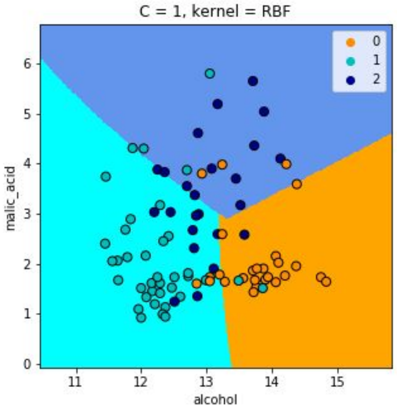
C = 100, kernel = LINEAR

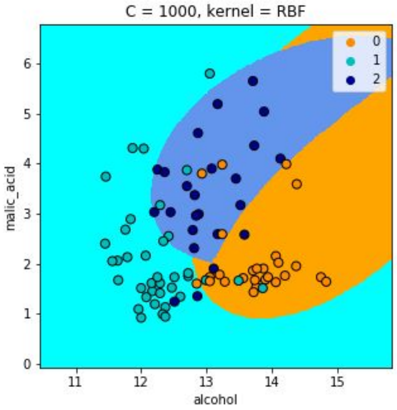
STEP 3 - RBF KERNEL

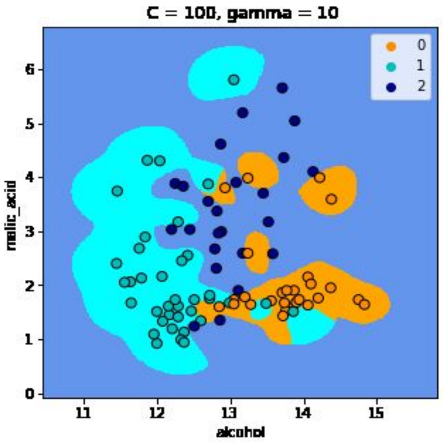
WHAT YOU SHOULD DO:

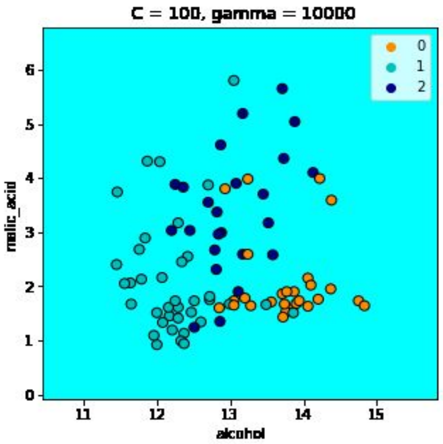
1. Keep the same data you used before (same features, same split)
2. Repeat the same steps you did before, this time use a SVM with an **RBF kernel:**
   a. for this first step, keep **gamma** fixed to its default value, vary only the **C parameter** (choose the values you think are the most suitable)
   b. are the decision boundaries different? why?
3. Perform a **grid search** over both **gamma** and **C** at the same time:
   a. for each of them, select an appropriate range
   b. plot decision boundaries
   c. choose the best parameter according to the performances on the evaluation set
   d. evaluate the model on the test set
4. Inspect the performance scores and the decision boundaries: what is the effect of **gamma?**
5. Does this model perform better than the previous one? Why?

WHAT YOU SHOULD GET:

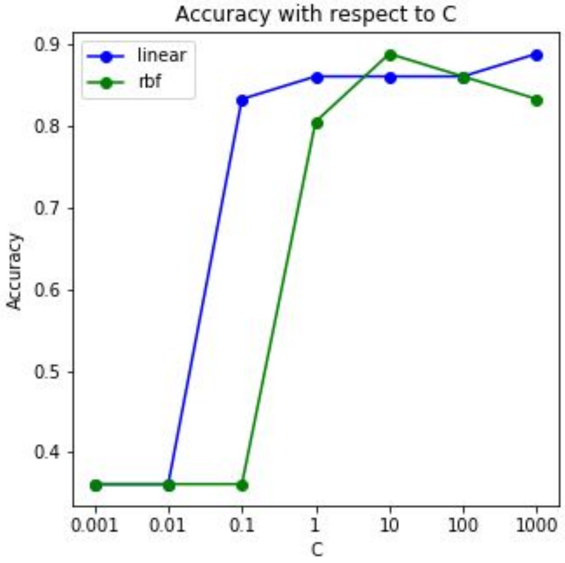


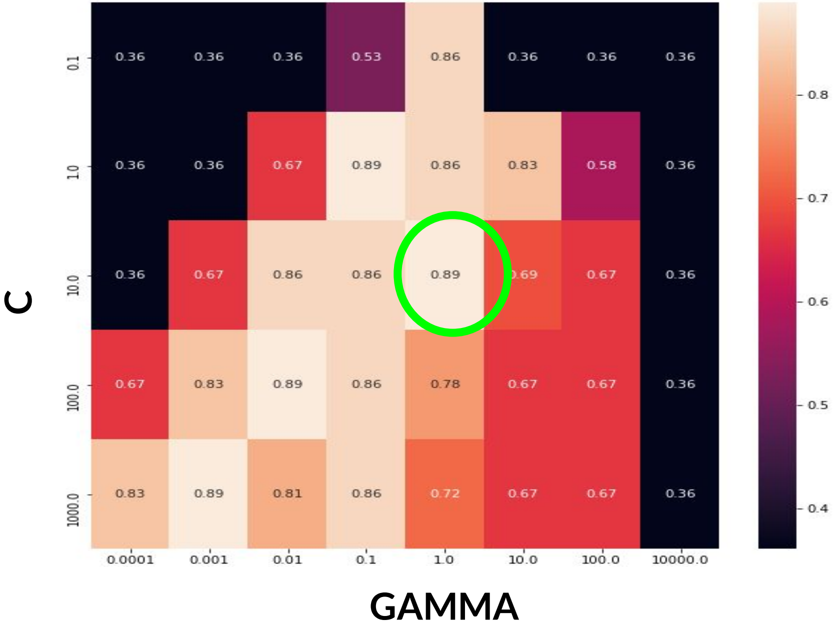






WHAT YOU SHOULD GET:





	0.0001	0.001	0.01	0.1	1.0	10.0	100.0	10000.0
0.1	0.36	0.36	0.36	0.53	0.86	0.36	0.36	0.36
1.0	0.36	0.36	0.67	0.89	0.86	0.83	0.58	0.36
10.0	0.36	0.67	0.86	0.86	0.89	0.69	0.67	0.36
100.0	0.67	0.83	0.89	0.86	0.78	0.67	0.67	0.36
1000.0	0.83	0.89	0.81	0.86	0.72	0.67	0.67	0.36

STEP 4 - K-FOLD

WHAT YOU SHOULD DO:

1. Keep the same data you used before (same features, same split)
2. Merge train and validation set.
3. Repeat the grid search for **gamma** and **C** but this time perform 5-fold cross validation
4. Evaluate on the test set. Is the final score different? Why?

NEED HELP?

USEFUL LINKS

- scikit-learn library: https://scikit-learn.org/stable/
    - wine dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html
    - model selection: https://scikit-learn.org/stable/model_selection.html
    - **knn:** https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
    - **svm**: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
- online you can also find some useful functions to easily plot decision boundaries

CONTACT US!

- Live assistance
  - **WHEN**: LUN 19 OCT 2020, 9.30 - 11.30
  - **WHERE**: Virtual Classroom
  - **WHO**: assistant Lorenzo Bonasera
- **Slack** channel
  - invitation link will be provided soon
  - 2 groups based on the surname (A-M and N-Z)
  - UNTIL **2 NOV 2020** (15 days from today). On that date a possible solution will be uploaded.
- **DO NOT WRITE EMAILS (please),** keep the discussion on Slack so that also other students can see and maybe help before we do

LET'S DO IT!