SUPPORT VECTOR MACHINES

MILAN KRESOVIĆ – S266915 ERASMUS STUDENT

INTRODUCTION

Support vector machines (SVMs) is a class of supervised algorithms for both classification and regression. In last homework we did **Naïve Bayes Classification** to do the classification of the data. This type of classification is called **generative classification**. In this homework another type of classification will be emphasized. **Discriminative classification** rather to model each class, this classification tries to find a line/curve (2D) or a manifold (Multiple dimension) to divide data in different classes.

Dataset used for this homework is **Iris** dataset. Dataset will be split into training, validation and testing sets. In first part of homework, training and validation sets will be used to train **linear SVM** for different parameters of C, and then parameter C from the model with best accuracy will be used to train linear SVM onto the testing set.

In second part of the homework, the same steps will be applied, only this time for **SVM with RBF kernel**. The best parameter C from training and validation set will be used to train SVM with RBF kernel onto the testing set. In third part, the same will be done, but this time model will be trained with different values of Gamma parameter also. The result of the third part will be grid of values of accuracy for models with different C and Gamma parameters. The values of these two parameters that gives the best accuracy on training and validation set will be used to train SVM with RBF kernel onto the testing set.

At last, in fourth part the same will be done as in the third part, only this time using **K-Fold cross-validation** and with training and validation sets being concatenated.

PREPARTION OF THE DATASET

In this exercise, the Iris dataset was used. The data sets consist of 3 different types of irises' petal and sepal length, stored in a 150x4 array. Only the first two components were used, that is **Sepal length** and **Sepal width**.

The data was then split into training, validation and testing set in proportion **5:2:3**. An array of parameter C value has also been made, ranging from 10^-3 to 10^3 where at each step value was multiplied by 10.

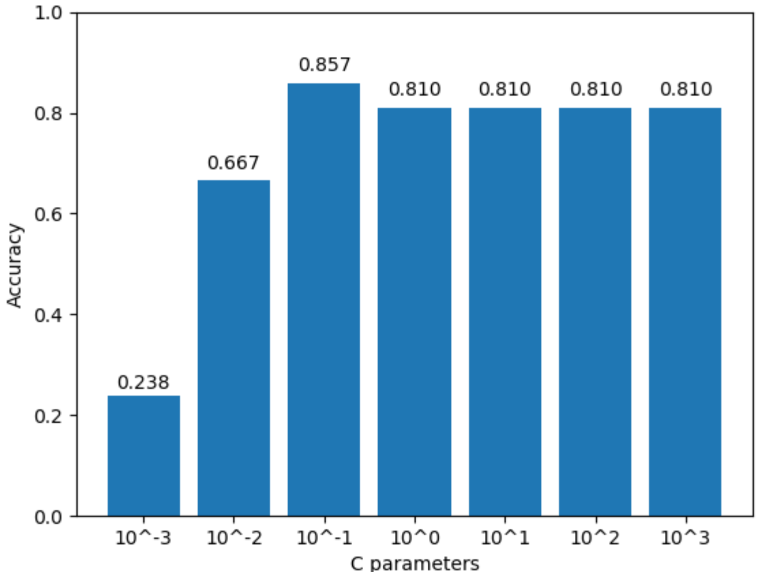
LINEAR SVM

A linear discriminative classifier attempts to draw a straight line (in 2D) to separate two sets of data. To do the optimal job, this type of classifier will try to maximize the margin. That is, the line that will be decided as the best option for separating the data should have the maximum margin between two classes of points. The nearest points of classes that touch that margin are known as the support vectors, thus this algorithm's name.

Depending on how the data overlapped, the tuning of the SVM is allowed. This means that it is possible to change values of some of the parameters to find the best fit. In first part of the homework different values of the parameter C will be used to see how does it changes the model.

Parameter C controls tradeoff between smooth decision boundary and classifying training point correctly. Simply put, it softens the margin and allows some of the points to get into the margin if that results in a better fit. The margin width is proportional to C^-1: so the bigger the C (C >> 1) – the margin is hard and points cannot lie in it; the smaller the C (0 < C < 1) – the margin is softer and can contain some of the points.

The model was used on training and validation set with linear kernel, and for values of parameter C between [10^-3, 10^3].



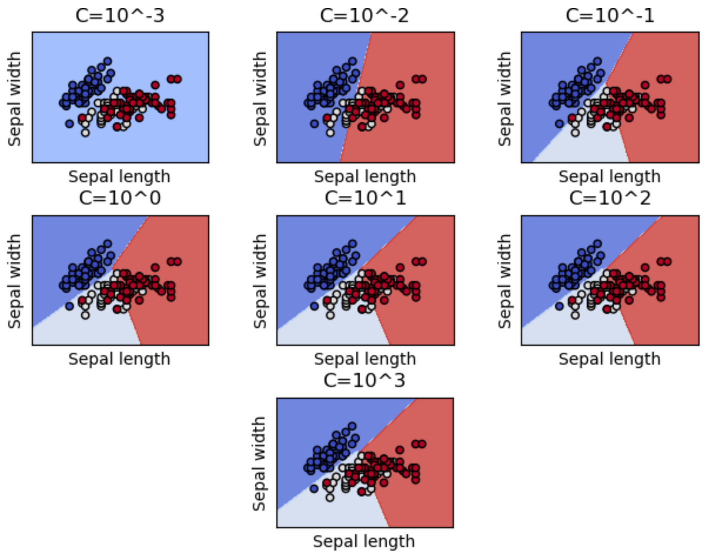


Figure 1. Linear SVM, accuracy for different values of parameter C

Figure 2. Linear SVM, decision boundaries for different values of parameter C

What can be seen on these two graphs is that having bigger parameter C, accuracy is getting better and therefore the data is better separated. As it was said, having bigger parameter C means that margins are harder and that points cannot lie in it. This is especially expected when we bear in mind how overlapped the data from different classes is. SVM is trying to get boundaries clearer and accuracy higher by having harder margin.

Showing only the decision boundaries visualization of the model applied onto the testing set with parameter C being from the model with the biggest accuracy:

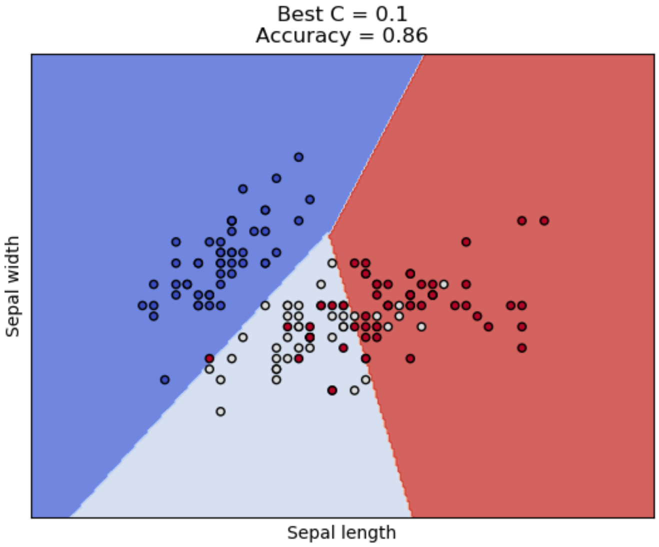


Figure 3. Linear SVM, decision boundaries for model applied onto the testing set with best parameter C

SVM WITH RBF KERNEL AND DIFFERENT VALUES OF PARAMETER C

In the second part of the homework, instead of using linear kernel, SVM was trained with **Radial basis function kernel (RBF) kernel**. The results are following:

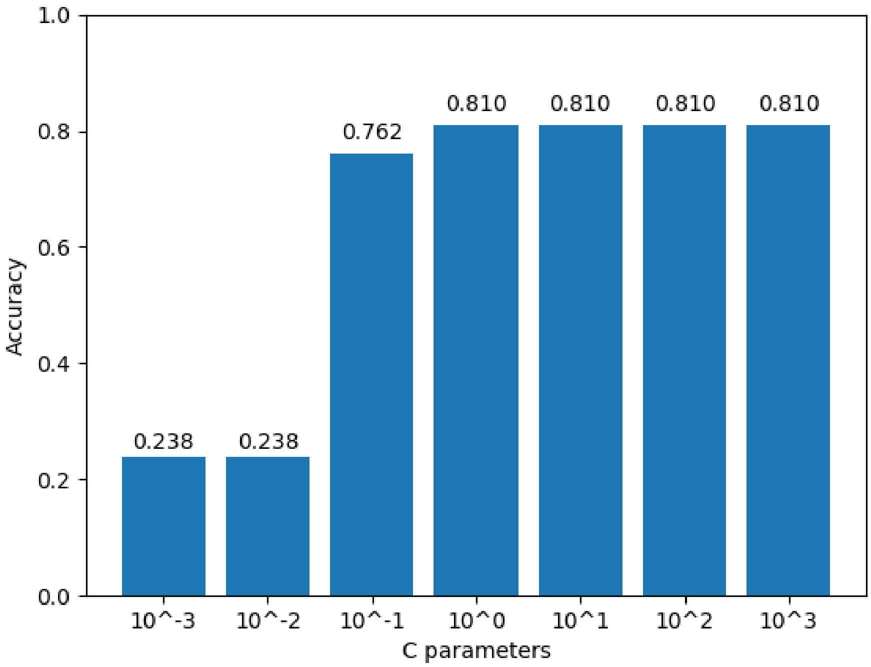


Figure 4. SVM with RBF kernel, accuracy for different values of parameter C

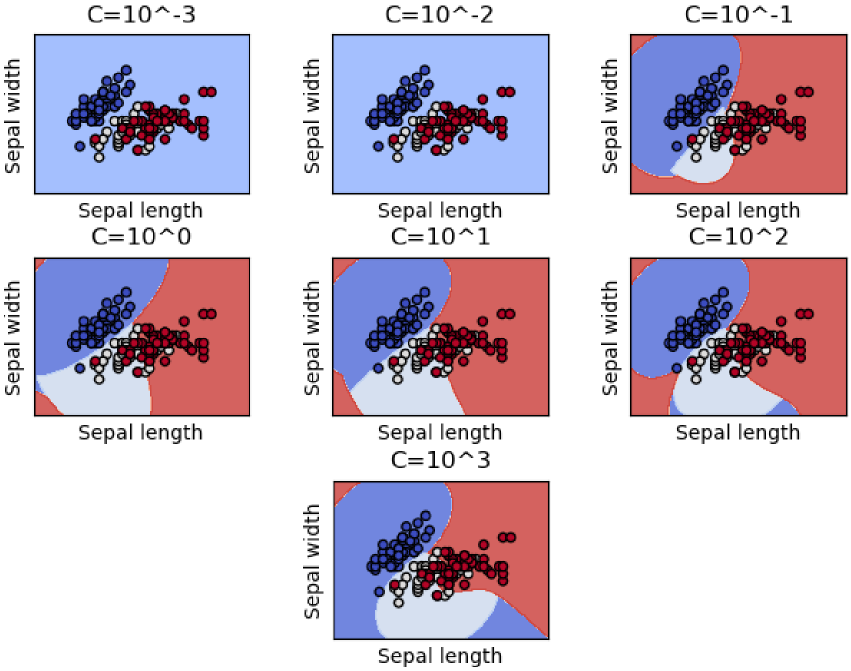


Figure 5. SVM with RBF kernel, decision boundaries for different values of parameter C

Graphs show similar results as the one from using linear kernel. Using the parameter C from the model with the best accuracy, the following decision boundaries are obtained:

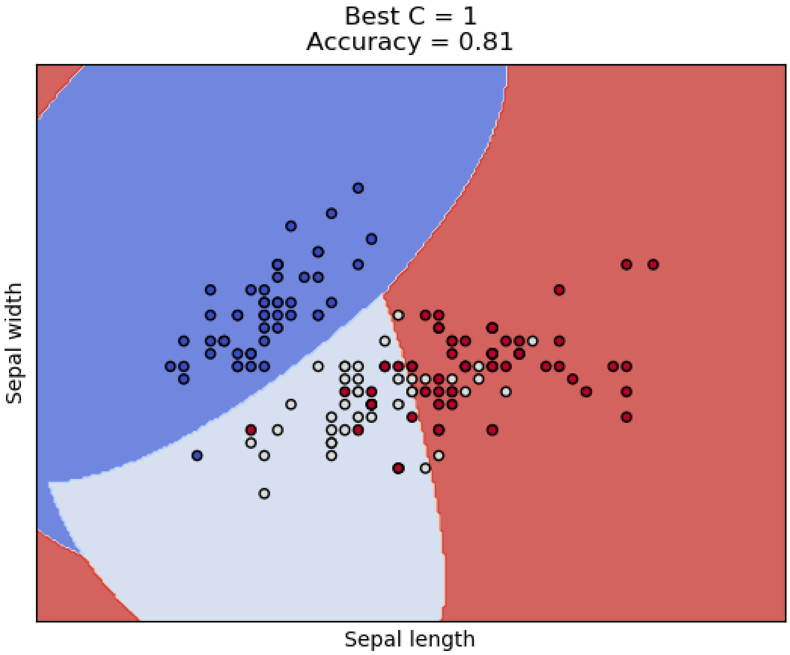


Figure 6. SVM with RBF kernel, decision boundaries for model applied onto the testing set with best parameter C

In figure 6, it can be seen that the results are similar as the one done with using linear SVM. Accuracy is lower and this can indicate that the training dataset is too small, and that possibility of using different validation methods, as K-fold cross-validation, should be considered.

What is also observable in the decision boundaries of the models of SVM with RBF kernel is that boundaries are now not linear as in the first part of the homework, which is expected.

SVM WITH RBF KERNEL AND DIFFERENT VALUES OF PARAMETER C AND GAMMA

In the previous parts of the homework only the hyperparameter C was tuned. When applying SVM with RBF kernel **Gamma hyperparameter** can also be considered for tuning. Gamma parameter defines how far the influence of a single training example reaches. This parameter can be thought of as the 'spread' of the kernel and decision region. That is, when gamma is low – the curve of the decision boundary is lower and the decision region is broader; when gamma is high – the curve of the decision boundary is higher and it created "islands" of decision-boundaries around data points.

Heatmap of the accuracy values acquired from different values of parameter C and Gamma when using SVM with RBF kernel is following:

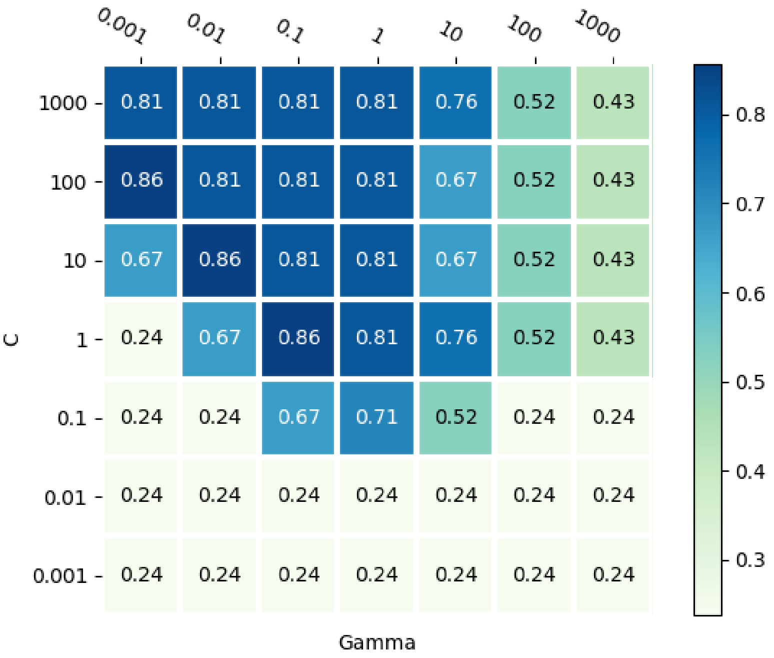


Figure 7. Heatmap of the accuracy values obtained for different values of parameters C and Gamma

From the figure 7 it can be seen that having higher parameter C values and lower parameter Gamma values creates the best results, that is, the accuracy values are the best.

More thoroughly explained, that means that because the different classes are so overlapped, having harder margins (points cannot lie in the margin) and taking even the more distant points into consideration when it's decided where to draw the decision boundary (i.e. having lower curve of the decision boundary) gives the best results of separation of the classes. This can be seen on the plot showing decision boundaries of the model, with best parameter C and Gamma, applied onto the testing set.

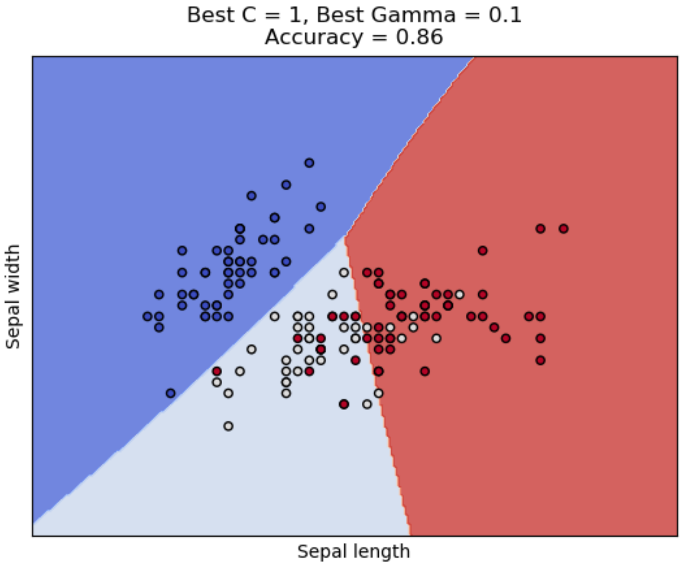


Figure 8. SVM with RBF kernel, decision boundaries for model applied onto the testing set with best parameter C and Gamma

K-FOLD CROSS-VALIDATION

Until now, all the data was split into three sets: training, validation and testing set. Problem with this partitioning is that the number of samples which can be used for learning the model is drastically reduced, but also the results can depend on a particular random choice for the pair of sets. To resolve this problem, it is possible to perform a procedure called **K-fold cross-validation** which would hopefully result in having better tuned parameter C and Gamma and overall better accuracy value.

The training set will be concatenated with the validation set and it will become new training set. K-fold cross-validation is used to divide the new training set into k distinct subsets (folds). Then, in each iteration one subset is used for validation while others k-1 are used for training the model. When all the iterations are finished, accuracy values from all the iterations are then averaged and that is the final accuracy of the trained model.

In this, fourth part of the homework, it was used 5-fold cross-validation. That is, the new training set was divided into 5 folds.

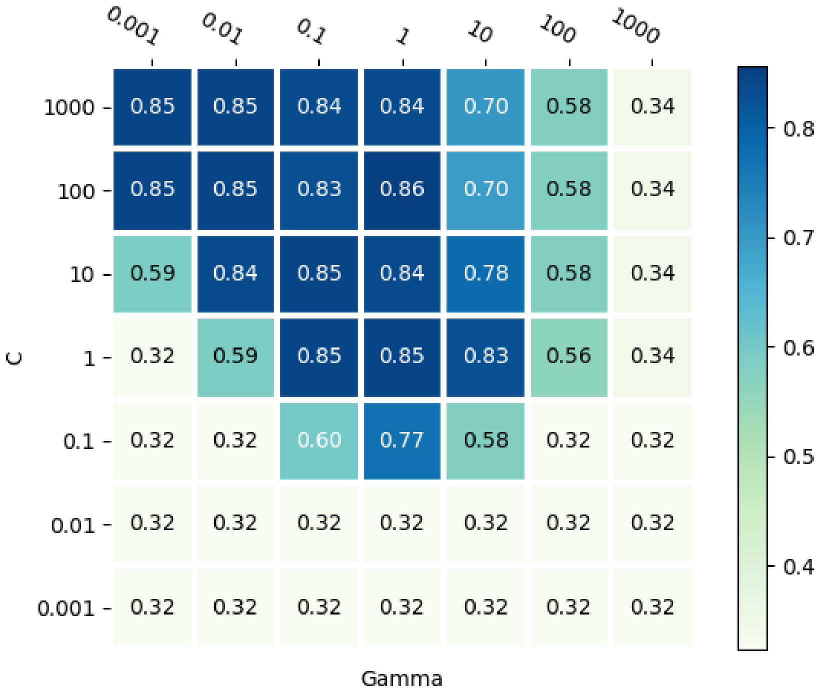


Figure 9. Heatmap of the accuracy values obtained for different values of parameters C and Gamma after the K-fold cross-validation

In figure 9 it can be seen that as in the previous part getting bigger value for parameter C and smaller for parameter Gamma is resulting in better accuracy values.

The next plot is visualized decision boundaries for the best values of parameter C and Gamma using SVM with RBF kernel and K-fold cross-validation:

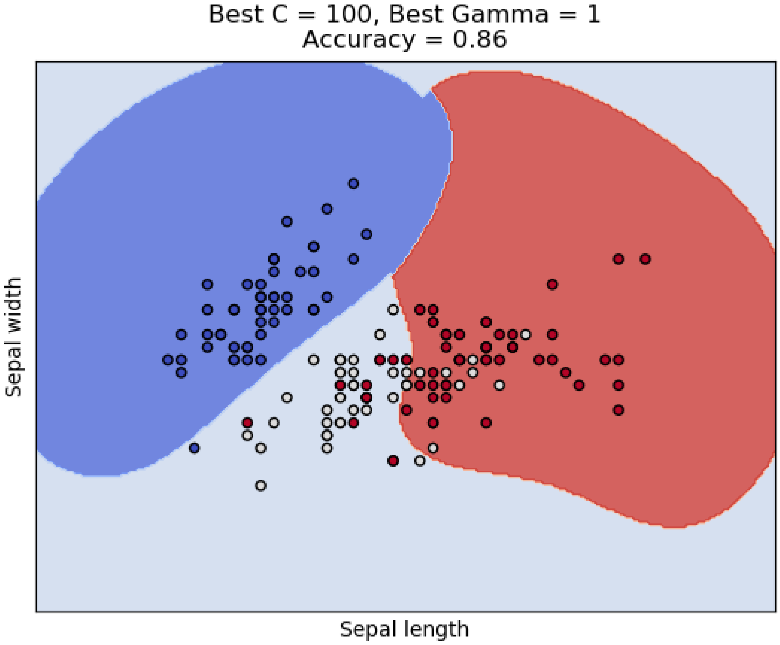


Figure 10. SVM with RBF kernel, decision boundaries for model applied onto the testing set with best parameter C and Gamma with K-fold cross-validation

In figure 10 it can be seen that by using K-fold cross-validation and therefore multiplying volume of the data with which SVM with RBF kernel is trained, accuracy value is the biggest for parameter C being 100 and parameter Gamma being 1 for this dataset.