University of Córdoba

Computer engineering
Computer Science Engineering Degree
4th course

Introduction to computational models

Lab Assignment 4: Support Vector Machine

Valentín Gabriel Avram Aenachioei
03524931C
p92avavv@uco.es

Academic year 2022-2023
Córdoba, June 9, 2023

Contents

List of Tables

List of Figures

1 Introduction

In this paper, I will set out how I have done the fourth lab assignment of the subject, the Support Vector Machine. In this lab assignment, I will not explain the model used and will not emphasise on the code implementation.

Instead of emphasizing on the code implementation, I will add a script used to resolve the question, based in the script given as a guideline. This paper will focus on answering the questions asked in the instructions of the lab assignment.

2 Questions

2.1 Question 1

Open this script and explain its contents. You will see that the first dataset is used, and the SVM is graphically represented. Comment on what type of kernel is being used, and what are the training parameters.

The script *libsvm.py* is already given as a guideline, so it is unncessary to show the code.

First of all, the libraries are imported and the datasets are loaded. Then, a SVM model is created and trained. For this, it is used a linear type of kernel and C value of 100, being C the regularization parameter. Then, the results are plotted. The generated plot for the first dataset is:

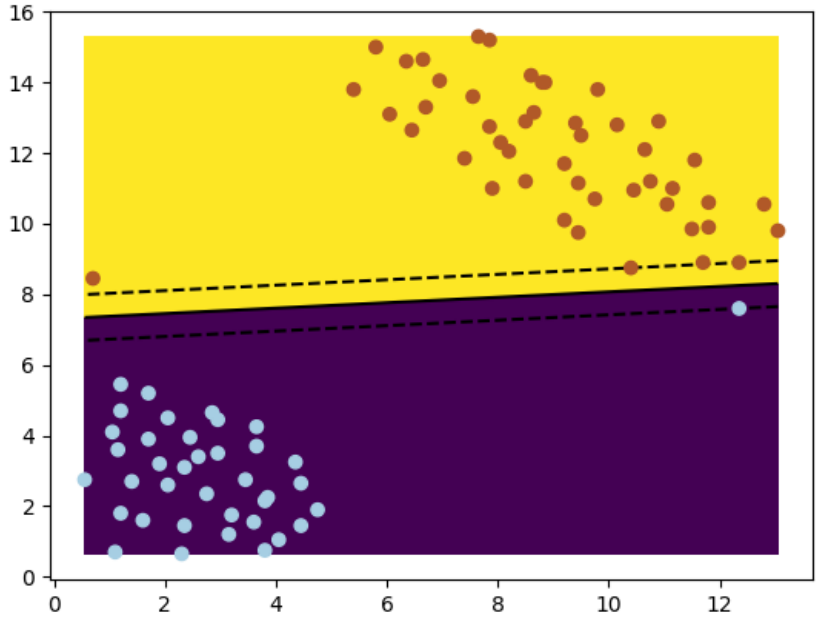

Figure 1: Basic script Dataset 1 Plotting

It only shows the H, H^+ and H^- hyperplanes and the representation of the patterns in the space.

2.2 Question 2

Intuitively, which hyper plane do you think will make the least mistake in the task of separating the two kinds of points?

The image 2 shows the points represented in the space, and in green, there is a hand-drawn possible hyperplane, painted in green color.

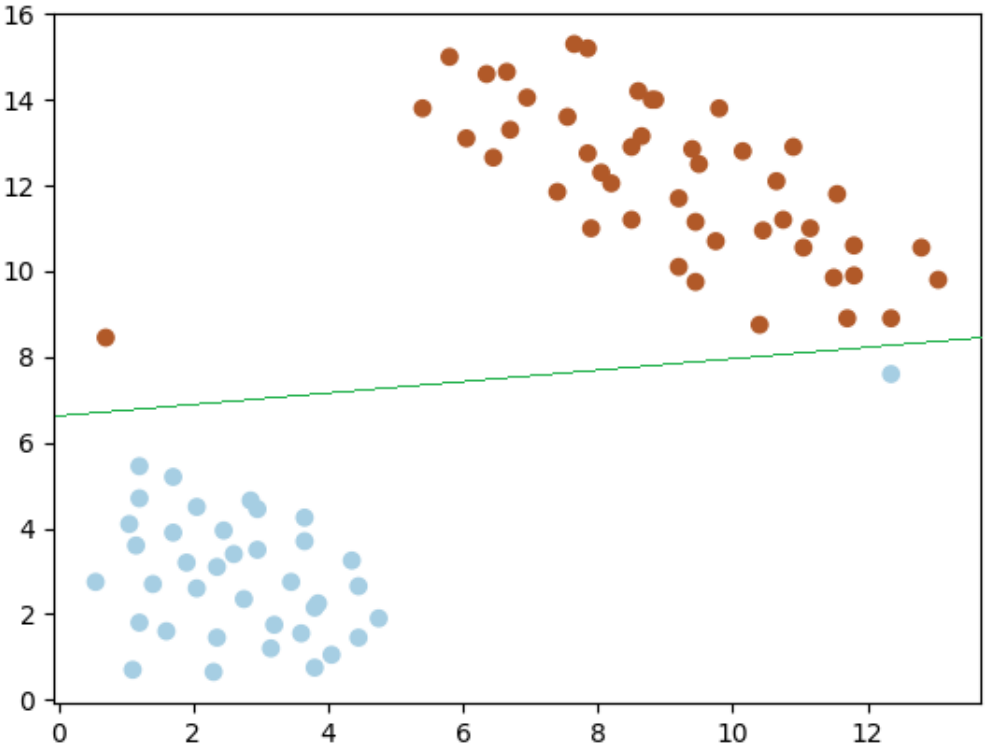

Figure 2: Dataset 1 points plotting

2.3 Question 3

Modify the script trying different values for C, specifically, C $\in$ $(10^2, 10^1, 10^0, 10^1, 10^2, 10^3, 10^4)$. Observe what is happening, explaining why and select the most adequate value for C.

With lower C values, the margin increases, wich can lead to misclassifications, as can be seen in the image 3.

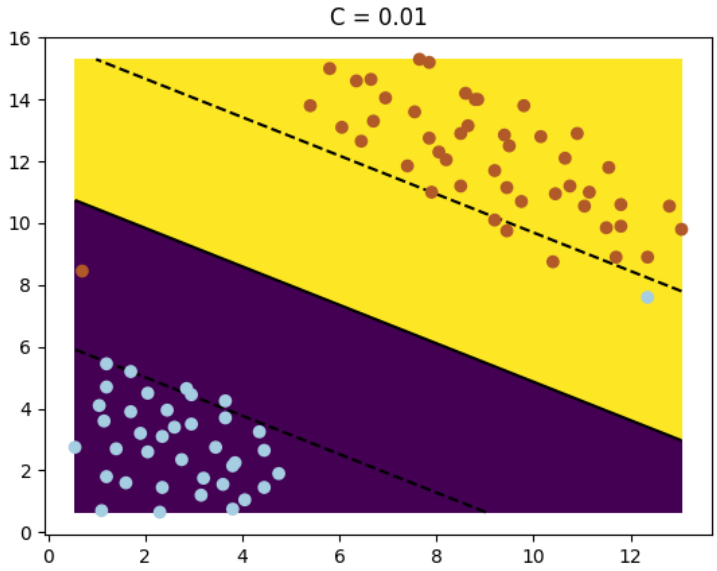

Figure 3: Dataset 1 with C=0.01

For values of C higher than $C = 1$, the SVM is more restrictive, having some of the patterns in the margin itself, as can be seen in the in the image 4.

The most adequate value for C could be $C = 1$, since with bigger values of C the results does not improve.

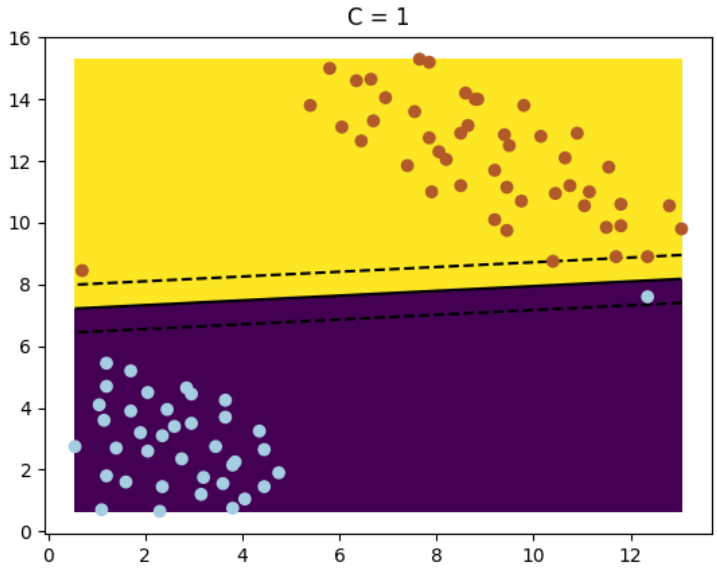

Figure 4: Dataset 1 with C=1

2.4 Question 4

Try running a linear SVM with the values for C used in the previous question. Do you get any satisfactory results in the sense that there are no errors in the training set? Why?

In this case, the dataset is not linearly separable, at least on 2 dimentions. As we can see in the images 5 and 6, the adjusting the C value is complete useless. A possible solution could be using another type of kernel, since a linear hyperplane is not able to difference classes.

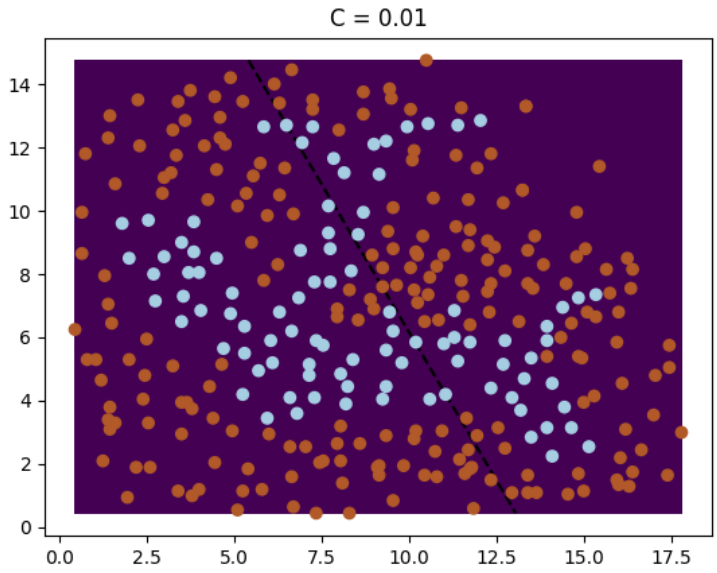

Figure 5: Dataset 2 with C=0.01

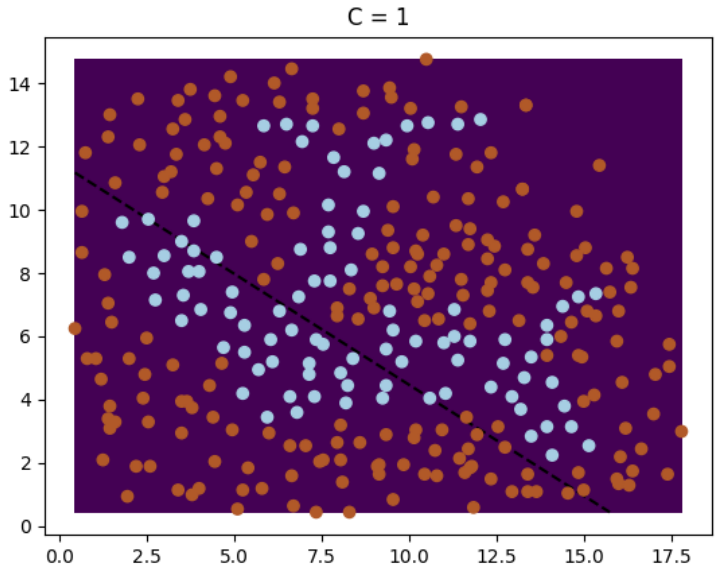

Figure 6: Dataset 2 with C=1

2.5 Question 5

Propose a non-linear SVM configuration (using RBF or Gaussian kernel) that solves the problem. The result should be similar to the one in Figure 3. What values have you considered for C and for γ ?

Also, include an example of a parameter configuration that produces overfitting and another that produces under-fitting.

In this case, a Radial Basis Function has been used. A RBF configuration that solves the problem can be using $C = 100$ and $\gamma = 2$. The result can be seen in the image 7

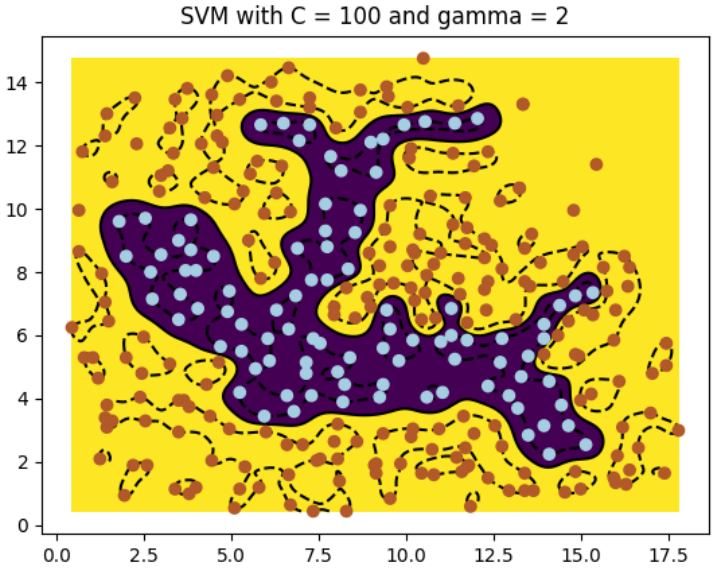

Figure 7: Dataset 3 using C=100 and $\gamma = 2$

The image 8 shows a configuration that leads to underfitting, using $C = 10^3$ and $\gamma = 10^{-3}$

The image 9 shows a configuration that leads to overfitting, using $C = 10^2$ and $\gamma = 20$

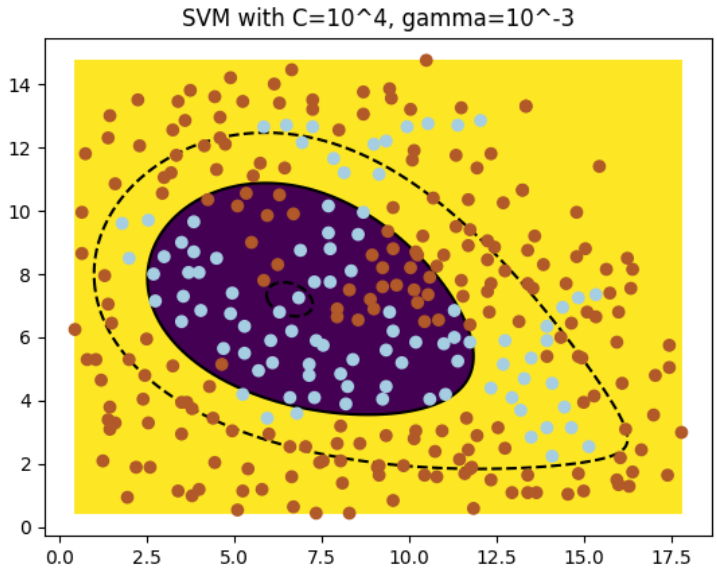

Figure 8: Dataset 2 underfitted

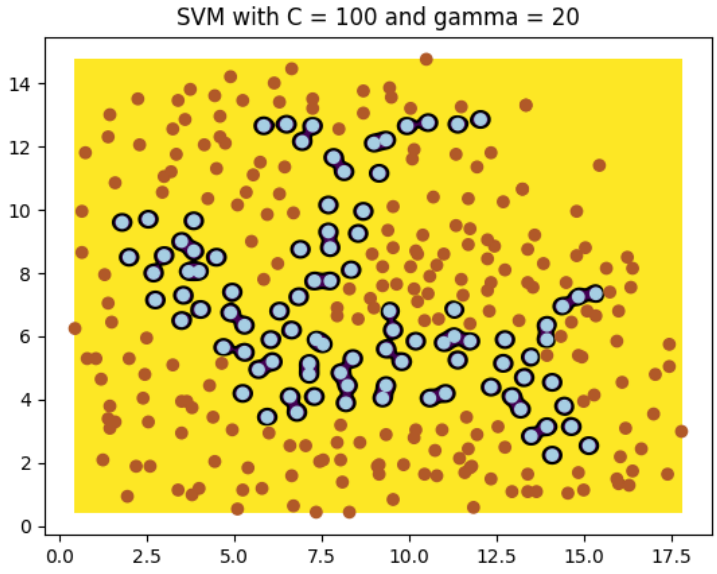

Figure 9: Dataset 2 overfitted

2.6 Question 6

In this case, is the dataset linearly separable?. At first sight, do you detect points that are presumably 'outliers'? Why?

As it can be seen in the image 10, as in the previous dataset, the data is not linearly separable using a linear kernel. There are some otuliers, distanced from the main cluster of its class.

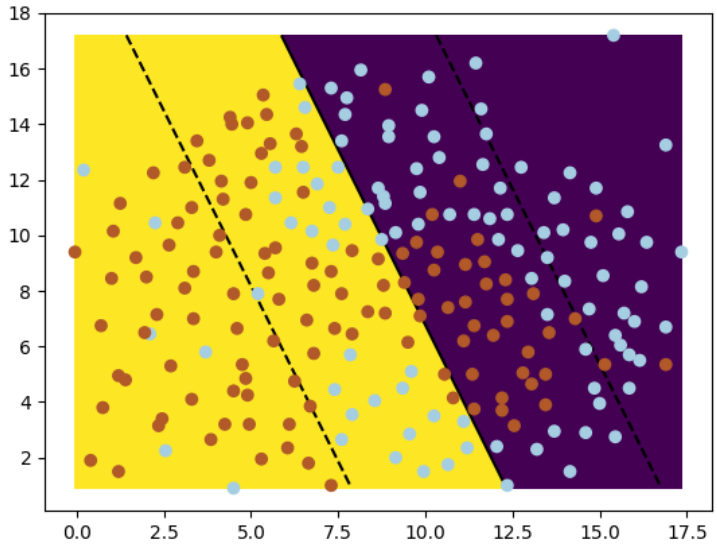

Figure 10: Dataset 3

2.7 Question 7

Run a SVM to classify the data, in order to obtain a result as close as possible to that of Figure 5. Set the value of the optimal parameters. In addition, include an example of a parameter configuration that produces over-fitting and one that produces under-fitting.

In this case.The image 11 shows a configuration that leads to underfitting, using $C = 20$ and $\gamma = 2$

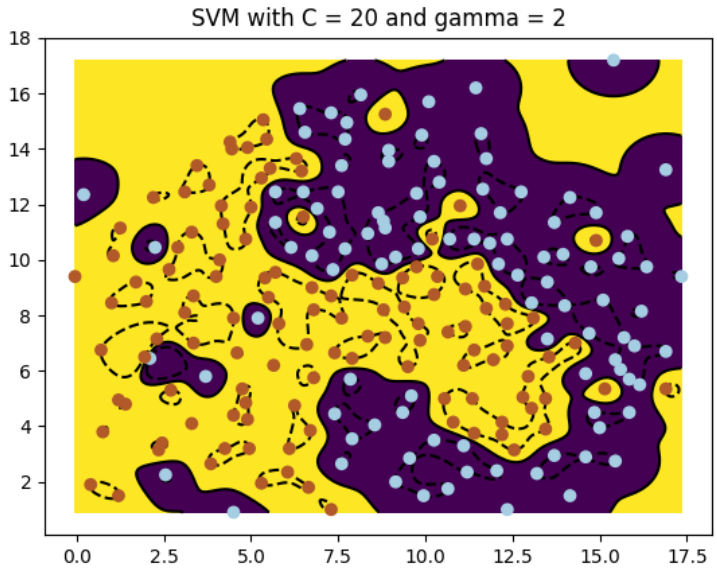

Figure 11: Dataset 3 using C=20 and $\gamma = 2$

The image 12 shows a configuration that leads to underfitting, using $C = 0.2$ and $\gamma = 0.2$

The image 13 shows a configuration that leads to overfitting, using $C = 200$ and $\gamma = 20$

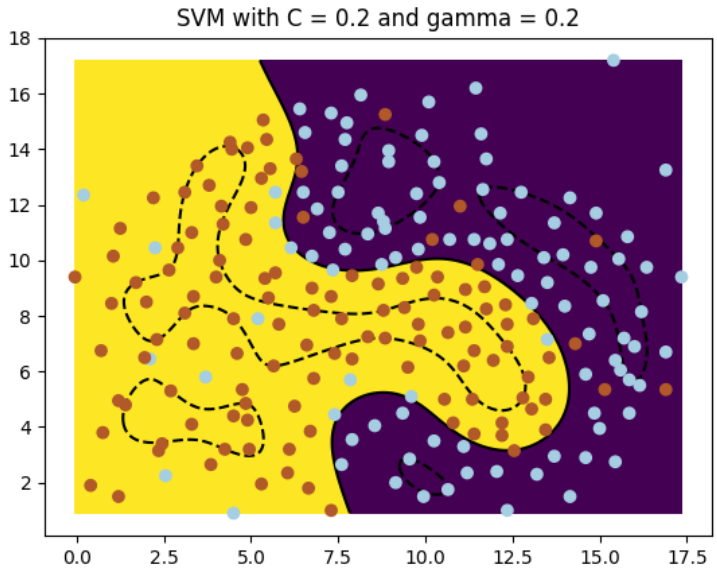

Figure 12: Dataset 3 underfitted

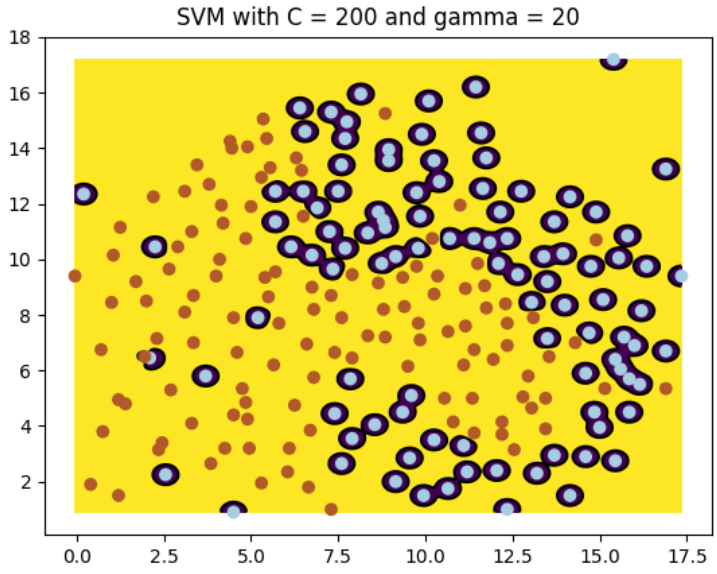

Figure 13: Dataset 3 overfitted

2.8 Question 8

We are going to reproduce this process in Python. Divide the synthetic dataset dataset3.csv into two stratified random subsets, with a 75% of patterns for training and a 25% of patterns for the test set. Make the complete training process (standardization, training and prediction), optimizing again the values of C and γ that you obtained in the last question. Check the accuracy that is obtained for the test set. Repeat the process more than once to check that the results depend a lot on the seed used to make the partition.

Changing the *´random_state´* value while splitting the dataset, we obtain the results of accuracy for testing seen in the table 1

Kernel Used	**random state value**	**C**	γ	**Accuracy**
Linear	1	10^{-2}	-	*66.07 %*
RBF	1	*10*	*1*	*89.28 %*
Linear	5	10^{-2}	-	*66.07%*
RBF	5	*10*	*1*	*85.71%*
Linear	20	10^{-2}	-	*69.64%*
RBF	20	*10*	*1*	*89.28%*

Table 1: Dataset 3 partition and results

2.9 Question 9

Extend the above code to perform the training of question 8 without the need to specify the values of C and γ.

Compare the optimal values obtained for both parameters with those you obtained by hand. Extend the range of values to be explored, if you consider it necessary.

For this case, only one of the random state values will be tested, $ramdon\_state = 20$. For this case, as the previous case, the best parameters obtained using a Grid Search were: $C = 100$ and $\gamma = 1$. The accuracy obtained in testing is 94.54% and 89.28% for testing. This results are similar to the previous obtained.

2.10 Question 10

What drawbacks do you observe in adjusting the value of the parameters "by hand", checking the accuracy in the test set (which was done in question 8)?

We can see that the results obtained in both question 8 and question 9 are the same, wich means that the script coded for the question 8 already was testing the possible values for C and γ. Obviously, adjusting the parameters by hand may let some parameters out of the test, which may lead to an error.

2.11 Question 11

To be sure that you understand how the parameter search is performed, implement manually (without using GridSearchCV) the K-fold nested cross validation explained in this section. You may find useful the use of compression lists and the class StratifiedK-Fold. Compare the results with those you get using GridSearchCV.

With the manully implemented K-fold nested cross validation, the best configuration obtained is $C = 10$ and $\gamma = 1$, a C value ten times lower. Even with this C value, the testing accuracy is the same, 89.28% .

2.12 Question 12

Use the script you developed in question 9 for the training of this database, without doing the split and the standarization. Pay attention to the CCR value obtained for the generalization set and compare it to the one obtained in previous practices. At the end, please, take a look to the optimal values obtained for the parameters.

In this case, the best estimated parameters have been : $C = 10$ and $\gamma = 0.1$. The accuracy obtained on training was 78.76%, worse as it is expected with those parameters.

2.13 Question 13

Find out where the value of K for internal cross validation is specified and the range of values used for the parameters C and γ.

How could you reduce the computational time needed to carry out the experiment? Try to set K = 3, K = 5 and K = 10 and compare, using a table, the computational times obtained and the results for the test set in terms of CCR.

The K-value is a parameter of GridSearch function, named 'cv´. The results of using different K values, with the best estimated parameter are shown in the table 2.

K-value	Time (s)	C	γ	Train Accuracy	Test Accuracy
3	*3.66*	*10000*	*0.001*	*81.53%*	*68.18%*
5	*0.33*	*0.0001*	*0.0001*	*70.76%*	*70.45%*
10	*0.58*	*100*	*0.01*	*82.3%*	*63.63%*

Table 2: K-values testing

2.14 Question 14

Use these methods to calculate the performance of the model for each class and by groups. This is, for the best model, calculate confusion matrices for train and test, and display some given performance metrics for test (for example, accuracy and FNR, but you can use others that you consider valid). Analyse the expected behaviour according to this information and briefly discuss the limitations of the metrics when representing the confusion matrix in this problem.

As an overrall, we have a false positive rate of =91.15% and a false negative rate of =3.25%. If we do the same analysis by classes, we can obtain the result seen in the table 3.

Class	False positive rate	False negative rate
0	*88.57%*	*2.1%*
1	*100%*	*7.14%*

Table 3: Missclassification per class

As we can seen, the results does not make much sense, since we have a 100% false positive rate and a 7.14% false negative rate at the same time.

2.15 Question 15

Optimize the model for classification fairness by changing the cross validation criteria.

Since we are in a binary problem, we can use the generic expression: $GM = \sqrt{S_+ S_-}$. Using the recall_score functions we can obtain the Sensitivity on each class. After this, we have a result of $GM = 0.07588$.

2.16 Question 16

A linear SVM model with the values $C = 10^{-2}$, $C = 10^{-1}$, $C = 1$ and $C = 10$ must be trained. For this, use a script similar to the one used for question 9. Compare the results and stablish the best configuration

For this model, no split has been made and the training has been used. The accuracy obtained with the different C values can be seen in the table 4

C-value	Accuracy
0.01	*98%*
0.1	*98.9%*
1	*97.8%*
10	*97.5%*

Table 4: Linear model with different C values

As can be seen, the best results are obtained while using a $C = 0.1$ value.

2.17 Question 17

For the best configuration, it builds the confusion matrix and establishes the misclassification emails. Check the input variables for the emails incorrectly classified and find out the reason behind it. Note that for each pattern, when x_i is equal to 1 it means that the i-th word in the vocabulary appears, at least once, in the email.

After exectuing the script, we can see that most of the emails are well classified. The missclassifications are caused by some key words, different in each case. The results of the test can be seen in the table 5. For the key words, only some of them have been shown.

Email	True class	Predicted class	Keywords
10	*Spam*	*No spam*	*emailaddr, httpaddr, subscript*
22	*No spam*	*Spam*	*contact, email, sponsor*
59	*No spam*	*Spam*	*advertis, bank, enterpris*
74	*No spam*	*Spam*	*agent, broadcast, compani*
148	*No spam*	*Spam*	*administr, contract, dollar*
329	*No spam*	*Spam*	*announc, review, servic*
408	*Spam*	*No spam*	*account, democrat, govern*
527	*No spam*	*Spam*	*emailaddr, httpaddr, mial*
561	*No spam*	*Spam*	*cours, number, spam*
843	*No spam*	*Spam*	*adquir, commerci, free*
882	*Spam*	*No spam*	*cours, drug, emailaddr*

Table 5: Missclassification on Linear SVM and some keywords causing it

2.18 Question 18

Compare the results obtained with the results achieved using an RBF network. To do this, use the programme developed in the previous practice. Use only one seed (the one with the best results)

Using the script made in the third lab assignment, the best accuraccy obtained is 98.60%. That means that the RBF is generalising as the SVM does, because the difference in negictible. This does not makes much sense, since the RBF should be generalising slightly better than the SVM, although the difference would be negictible.

2.19 Question 19

Train a non-linear SVM and compare the results obtained

For this, we can use again the script made for the question 17, adapting it to use a RBF kernel. The table 6 shows the results of the test. Again, only some of the keywords have been shown.

Email	True class	Predicted class	Keywords
10	*Spam*	*No spam*	*emailaddr, httpaddr, subscript*
22	*No spam*	*Spam*	*contact, email, sponsor*
50	*No spam*	*Spam*	*command, english, internet*
90	*Spam*	*No spam*	*agent, broadcast, compani*
118	*Spam*	*No spam*	*attack, concern, free*
209	*No spam*	*Spam*	*acces, data, pattern*
305	*Spam*	*No spam*	*health, stop, your*
329	*No spam*	*Spam*	*announc, review, servic*
792	*Spam*	*No spam*	*account, enternain, guarante*
882	*Spam*	*No spam*	*cours, drug, emailaddr*

Table 6: Missclassification on RBF and some keywords causing it