Universidad de Córdoba
Escuela Politécnica Superior de Córdoba

Ingeniería Informática
Especialidad: Computación
Cuarto curso. Primer cuatrimestre

Introduction to computational models.

Lab Assignment 4: Support Vector Machines (SVMs).

Javier Herrero Porras
i72hepoj@uco.es

Academic Course 2020-2021
Córdoba, December 1, 2020

Contents

List of Figures

List of Tables

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. Explain the image generated, including all the different elements and colours.

In the script we train the SVM model with a linear kernel (that means that SVM model separate patterns with a straight line). Also we are using parameter C, which indicates the strength of the regularization, with the objective of controlling the error. If C is large that means that model accept less error and it will be a complex model. In our case C is large, so the model will accept less errors.

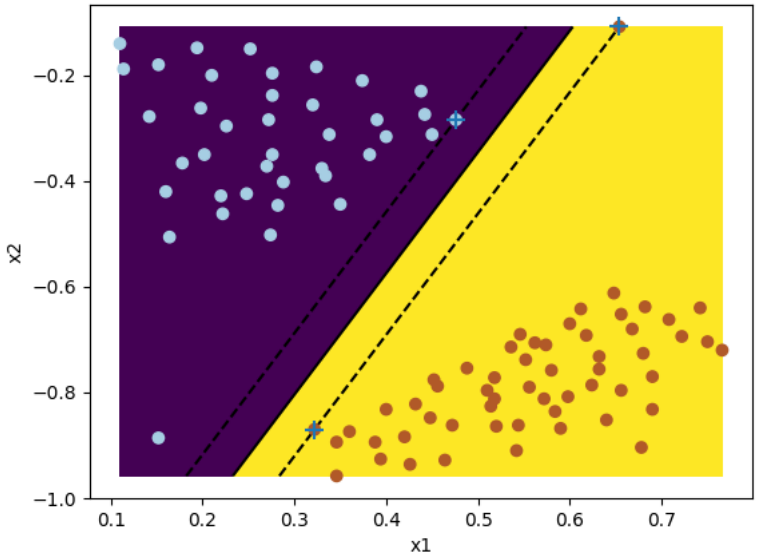

Figure 1: Question 1

In image 1 we can check the result of the model generated for the points. As we said before the model is very simple because we chose a linear kernel and C was large, so the model is able to accept some errors. We check that purple points are for example the blue ones and the other class is the red one. We can also verify the hyperplane obtained (continuous line) and the support vectors for this dataset (discontinuous lines).

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?

In the image obtained, the simpler hyperplane which make the least mistake in separating the two kinds of points is the one which separates both classes without making mistakes. Also there is another option, which is the one that was made by SVM model in Question 1, which makes 2 mistakes in patterns.

3 Question 3

Modify the script trying different values for C, specifically, C belonging to {10e2, 10e1, 10e0, 10e1, 10e2, 10e3, 10e4}. Observe what is happening, explaining why and select the most adequate value for C.

In this case, I have changed the value of C by hand. In the next questions I will change the code to accept it by command line. With different values of C I have obtained these results:

- C = 10e-2: the model is trying to maximize the margin, so it is accepting more errors. This lead to over-fitting, because the model is learning also noisy or wrong patterns.
- C = 10e-1: the model has a small margin, so the errors made are less than model with C=10e-02. Now the model is more complex.
- C = 10e0: the margin is smaller. The number of mistakes is reduced (10 now).
- C = 10e1: the margin is smaller. The number of mistakes is reduced (5 now).
- C = 10e2: the margin is smaller. The number of mistakes is reduced (3 now).
- C = 10e3: the obtained model is the same as the previous value of C.

- C = 10e4: the obtained model is the same as C = 10e2 (mistakes = 3). The model cant make margin smaller. Maybe is good to reconsider changing other model parameters to get a better model (kernel, gamma, etc).

For example, the model obtained with the value of C = 10e-0 is shown in figure 2. In this case the most adequate value of C is 10e2, because with higher values of C the model does not change.

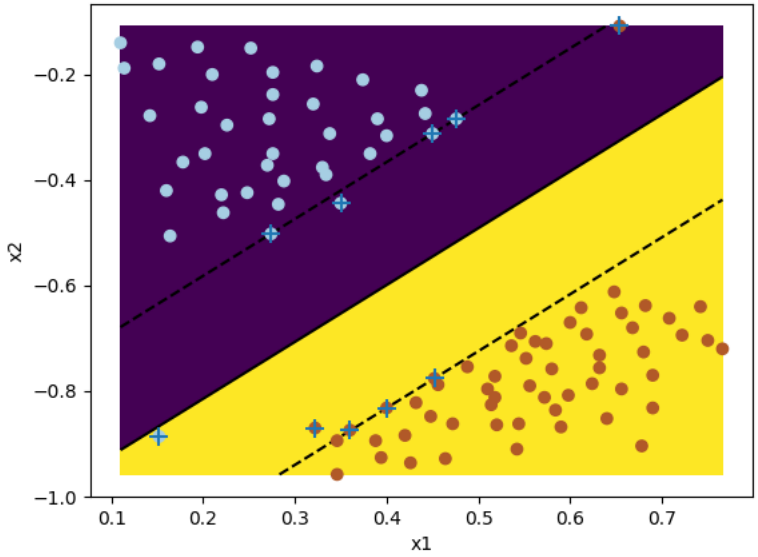

Figure 2: Question 3

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, data is not linearly separable (as we can see in figure 3), so that this model is not able to distinguish correctly between both classes. We will need kernel trick, to consider more dimensions in feature space and separate them. Then, we map the results into the input space, which is easier to compute in.

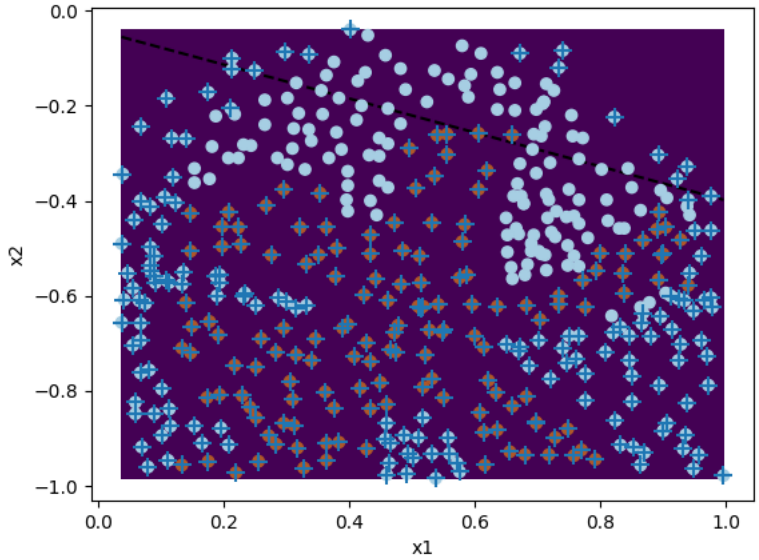

Figure 3: Question 4

5 Question 5

Propose a non-linear SVM configuration (using RBF or Gaussian kernel) that solves the problem. Try to use values for the parameters in the range C,γ belonging to $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$. What values have you considered for C and for γ? Also, include an example of a parameter configuration that produces over-fitting and another that produces under-fitting.

For this question, I have chosen values for C = 1 and $\gamma = 10^2$, obtaining the model shown in figure 4.

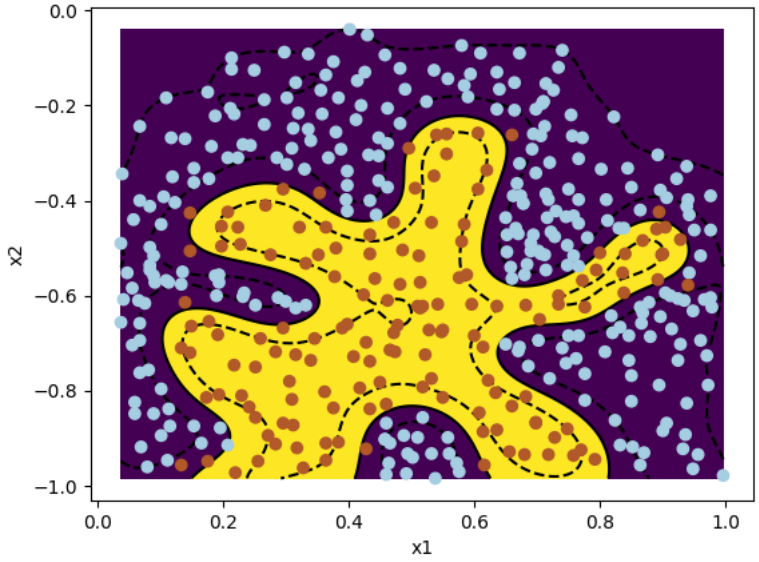

Figure 4: Question 5

One example of parameter configuration that produces over-fitting is $C = 10^3$ and $\gamma = 10^3$ and under-fitting is C = 1 and $\gamma = 1$. The results are shown in figure 5.

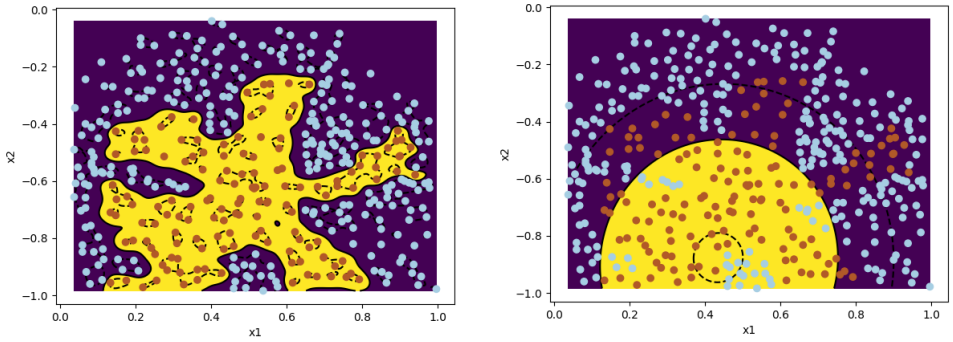

Figure 5: Over-fitting and under-fitting

6 Question 6

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

In this case, dataset is not linearly separable, because the points have the shape of two moons. Yes, I detect 2 red class points which are outliers, because its coordinates are equal to those ones which belong to the blue class.

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. Adjust the parameters in the range C,γ belonging to $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$. 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.

The optimal parameters obtained are C = 10 and $\gamma = 1$ which produces the model shown in figure 6.

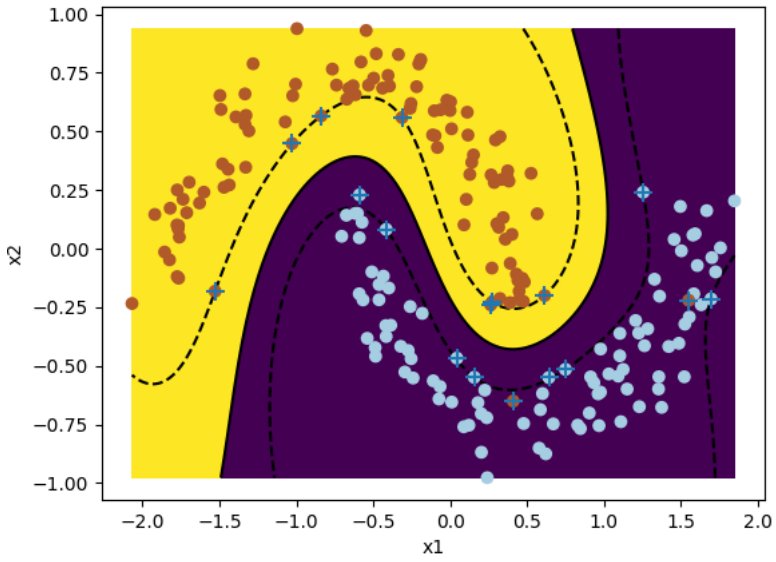

Figure 6: Question 7

One example of parameter configuration that produces over-fitting is $C = 10^2$ and $\gamma = 10^2$ and under-fitting is $C = 10^{-1}$ and $\gamma = 1$. The results are shown in figure 7.

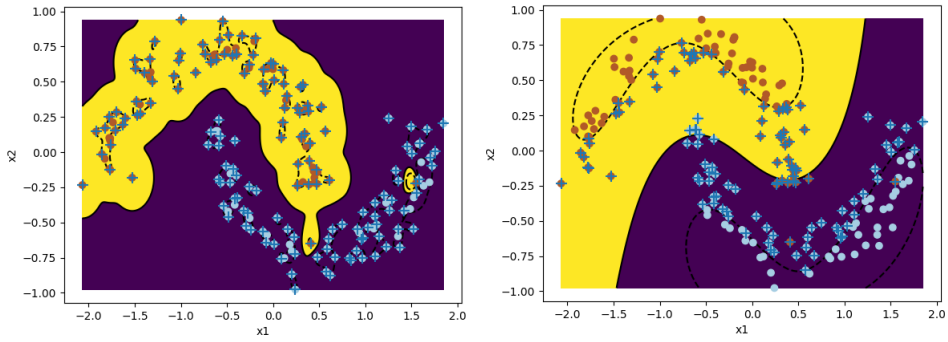

Figure 7: Question 7: Over-fitting and under-fitting

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. 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.

In this question, I have obtained these results (with different executions, I choose one of the different parameter combinations which got a perfect accuracy). We can check that results depend on the random subsets taken by *train_test_split* function:

Execution	C	γ	Accuracy
1	100	1.0	1.0
2	100	1.0	0.96078
3	100	1.0	0.98039
4	100	1.0	1.0
5	100	1.0	1.0

Table 1: Accuracy dataset3.csv

9 Question 9

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

I remember that in question 8, I selected values $C = 10^2$ and $\gamma = 1$. In this case, the optimal values obtained by *GridSearchCV* are $C = 1$ and $\gamma = 1$, with an accuracy = 0.98039 (but with other execution the accuracy = 1, as happened in the previous exercise). These values simplify the model and also get a very good accuracy, so this method is working correctly.

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)?

The first difference observed is that the grid search process is very fast in comparison with "by hand" method. Also, grid search is more secure than introducing parameter "by hand" because this is susceptible of errors made by the user.

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 StratifiedKFold. Compare the results with those you get using GridSearchCV.

The implementation made for this question is in file *Question11.py*. In this exercise, I have obtained an accuracy = 0.9868 (but it depends on the seed, because in other executions I have obtained an accuracy = 1). The optimal parameters are $C = 1$ and $\gamma = 1$, so this demonstrates that this implementations is implemented properly.

12 Question 12

Use the script you developed in question 9 for the training of this database. Pay attention to the CCR value obtained for the generalization set and compare it to the one obtained in previous lab assignments. The process can take a long time. At the end, please, take a look to the optimal values obtained for the parameters.

In the execution, I have obtained an accuracy = 0.8866 and the optimal values for C and gamma are $C = 100$ and $\gamma = 0.001$. This dataset is more complex than others studied before, so the model will be also more complex (due to the value of C) and the value of gamma (very low) implies that the model is not able to generalize new patterns which are far away from the training ones, because the model is constrained.

13 Question 13

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

To set the value of K for the internal cross validation we have to use parameter *cv* in GridSearchCV. Also, for setting C and γ we have to set parameters C and gamma in GridSearchCV (the range of values is establish in *Cs* and *Gs*).

To reduce computational time needed for the experiment, we can change the parameter *n_jobs* in GridSearchCV. In my case I set it to n_jobs = -1, which indicates the maximum of n_jobs.

K	Accuracy	Time
3	0.92	93.6937
5	0.8866	162.45
10	0.8866	374.32

Table 2: Accuracy and time Question 13

The result of the computational times and CCR is shown in table 2. We can see that accuracy is higher when we consider less K, that indicates us that the model works better with a lower number of training patterns (With K = 3, we consider 2/3 for training and 1/3 for validation), and also, the time increases when we increase K because it has to make more iterations and the process is also more complex.

14 Question 14

A linear SVM model with the values C = $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. For this, use a script similar to the one used for question 9. Compare the results and establish the best configuration.

The best configuration obtained with this script is a $C = 0.01$ and $\gamma = 0.001$ with a lineal SVM model. The accuracy obtained was 0.983. Although the dataset is more complex than others studied before, by the value of C we can say that the model will be simple and the value of gamma (very low) implies that the model is not able to generalize new patterns which are far away from the training ones, because the model is constrained.

15 Question 15

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

The implementation of the confusion matrix is done in file *Question15.py*. For the best configuration, we obtain this confusion matrix:

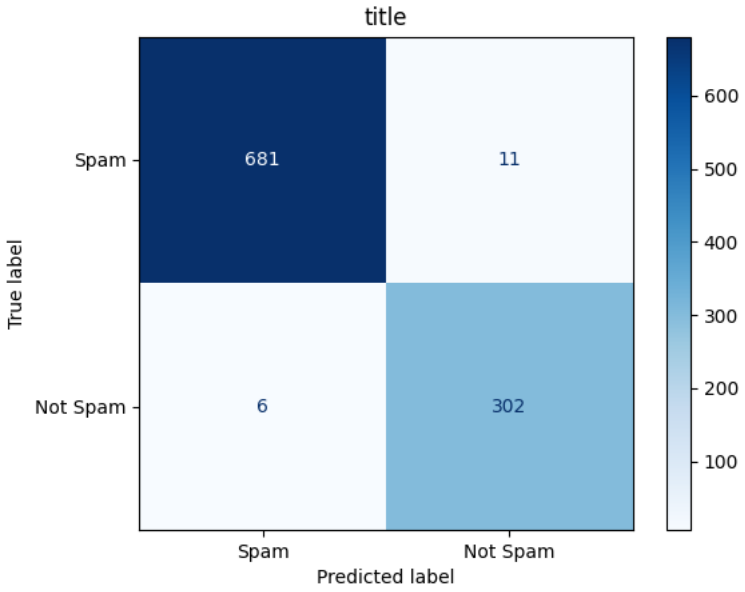

Figure 8: Question 15: Confusion matrix

In image 8 we check that there are 11 emails which are "spam" but the model classifies them as "not spam". Also there are 6 "not spam" emails that are classified as "spam". We are going to analyse these emails in the next subsection. We conclude that this model is able to classify correctly the "Not spam" emails with a high accuracy (98.05%), and the "Spam" emails (98.41%).

15.1 Emails wrong classified as Spam

The emails that are wrong classified as spam are 9, 90, 95, 150, 352 and 881. The content is shown in the following images:

```
9
['emailaddr', 'for', 'group', 'httpaddr', 'inform', 'irish', 'linux', 'list', 'maintain', 'subscript', 'un', 'user']

90
['aa', 'abil', 'abov', 'accord', 'account', 'action', 'add', 'administr', 'all', 'almost', 'alreadi', 'am', 'american', 'an', 'and', 'ar', 'ar
en', 'as', 'associ', 'at', 'averag', 'bar', 'be', 'been', 'best', 'better', 'black', 'board', 'busi', 'by', 'california', 'case', 'ceo', 'chan
c', 'chang', 'check', 'citizen', 'co', 'commun', 'consult', 'corpor', 'creat', 'current', 'dc', 'deliv', 'develop', 'differ', 'direct', 'down
', 'educ', 'email', 'emailaddr', 'emerg', 'engin', 'enjoi', 'equal', 'everi', 'exclus', 'exist', 'far', 'feder', 'fight', 'for', 'former', 'fou
nd', 'from', 'geek', 'get', 'go', 'govern', 'ha', 'have', 'heaven', 'help', 'high', 'how', 'howev', 'httpaddr', 'id', 'immedi', 'in', 'indian
', 'interest', 'is', 'it', 'let', 'list', 'lo', 'look', 'lot', 'mail', 'major', 'make', 'man', 'mean', 'measur', 'member', 'michael', 'monei',
'more', 'mostli', 'name', 'nation', 'net', 'new', 'no', 'non', 'not', 'number', 'numberf', 'of', 'on', 'onli', 'opportun', 'org', 'organ', 'ot
her', 'our', 'out', 'pacif', 'parti', 'past', 'peopl', 'perfect', 'perform', 'pictur', 'polit', 'potenti', 'present', 'presid', 'profession',
'progress', 'purpos', 'ratio', 'remov', 'repli', 'repres', 'return', 'seem', 'sf', 'show', 'sincer', 'situat', 'sourc', 'sponsor', 'state', 's
trategi', 'studi', 'support', 'tabl', 'take', 'templ', 'that', 'the', 'their', 'them', 'there', 'thi', 'thinkgeek', 'through', 'titl', 'to', '
toni', 'top', 'two', 'unfortun', 'univers', 'unsubscrib', 'us', 'vote', 'wa', 'washington', 'we', 'web', 'welcom', 'what', 'white', 'who', 'wi
ll', 'win', 'with', 'woman', 'women', 'word', 'world', 'yourself']
```

Figure 9: Emails wrong classified as Spam (I)

Figure 10: Emails wrong classified as Spam (II)

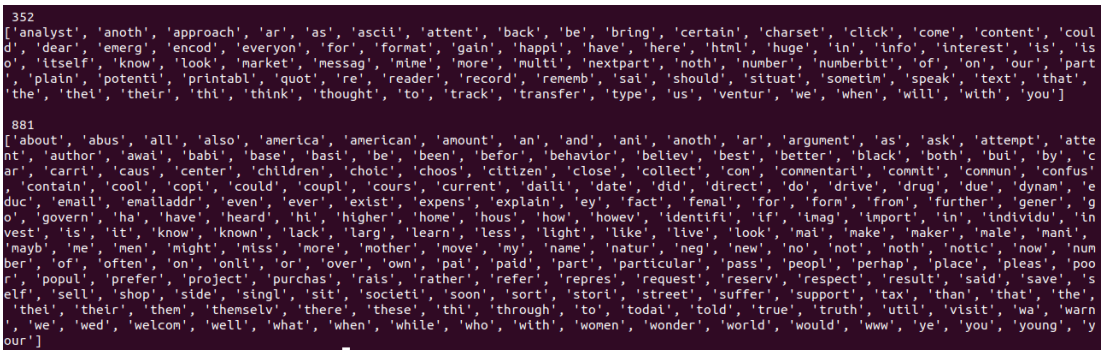

Figure 11: Emails wrong classified as Spam (III)

In general terms, it seems that our model detects spam on emails that have a few of words that begin by the same letter (for example a, t, etc). Also we can often find lots of pronouns, adverbs, prepositions, etc. In addition, the majority of emails have a lot of words.

15.2 Emails wrong classified as Not Spam

The emails that are wrong classified as spam are 21, 49, 73, 147, 208, 267, 328, 391, 526, 594 and 842. The content is shown in the following images:

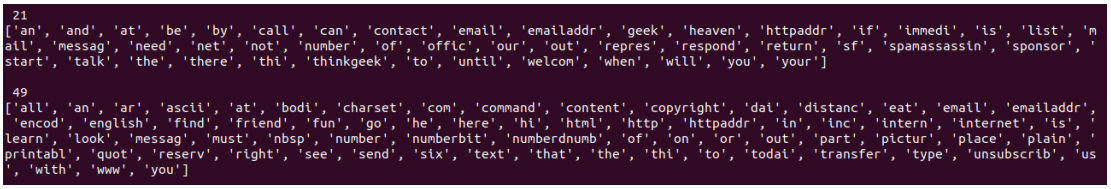

Figure 12: Emails wrong classified as Not Spam (I)

Figure 13: Emails wrong classified as Not Spam (II)

Figure 14: Emails wrong classified as Not Spam (III)

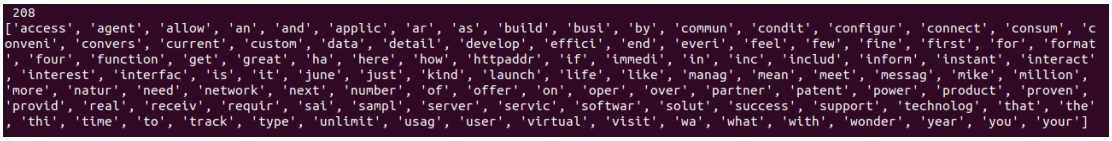

Figure 15: Emails wrong classified as Not Spam (IV)

Figure 16: Emails wrong classified as Not Spam (V)

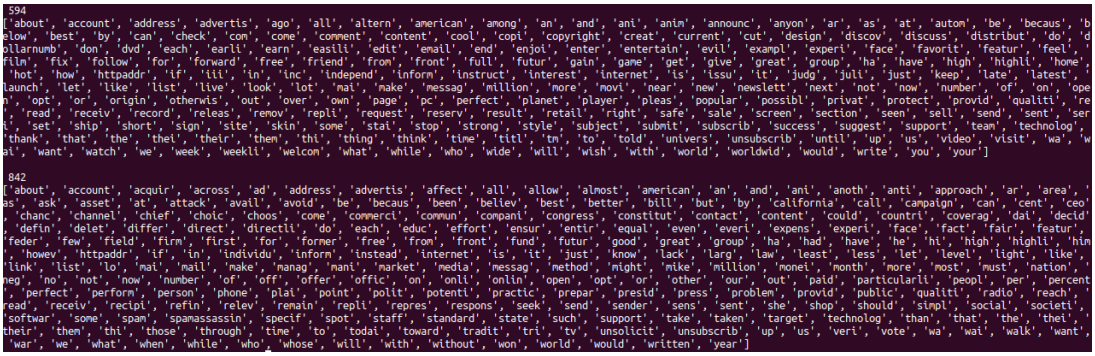

Figure 17: Emails wrong classified as Not Spam (VI)

Figure 18: Emails wrong classified as Not Spam (VII))

In general terms, our model does not detect a spam email when it has few or a huge number of different words (for example emails 21,49 or 73,147). Also, the wrong classified models have some key words like account, address, advertis (emails 73,147,554,267) that maybe confuse our model and make it to classify the emails wrong.

16 Question 16

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

NOTE: For this Question, I have used the programme developed in lab assignment 3, so I did not include anything about it in the folder.

In this case, I have repeated all the experiments I have made for the previous lab assignment. I have obtained that the better ratio of RBF neurons with respect the training patterns is 0.5. It is a high value with respect the results obtained in the previous lab assignment, but it can be caused by the fact that these patterns has 1900 inputs, so the dataset is very complex, so we need a high number of neurons to work better with it (the number of RBFs used is 2000, so the model we are going to work with will be complex). Also, η is 0.01 and the regularization used is L1.

The test CCR obtained is 99.4% which is the best result obtained for the dataset. We can say that RBF network works better with this type of problem (but SVM has a lot of tuning parameters, and in this practice I have worked with a few of them, so maybe it would be a good idea to make more experiments with more tuning parameters). The confusion matrix obtained is:

	Spam	Not Spam
Spam	689	3
Not Spam	3	305

Table 3: Confusion matrix

From table 3 we can conclude that RBF model classify with a very high accuracy Spam e-mails (99,56%) and also is very good classifying "Not Spam" e-mails, because the accuracy is very high (99.02%).

17 Question 17

Train a non-linear SVM and compare the results obtained

In this case, I have considered an 'rbf' kernel. After applying Grid Search, I have obtained the values $C = 10$ and $\gamma = 0.001$, with a final accuracy of 0,968. The final confusion matrix is shown in figure 19. We conclude that this model is able to classify very well the "Spam" emails (accuracy = 99,56%) but it is not able to work correctly with predicting "Not spam" emails, because its accuracy decreases to 90.58%, so the model I propose in this Question would not be good for this dataset.

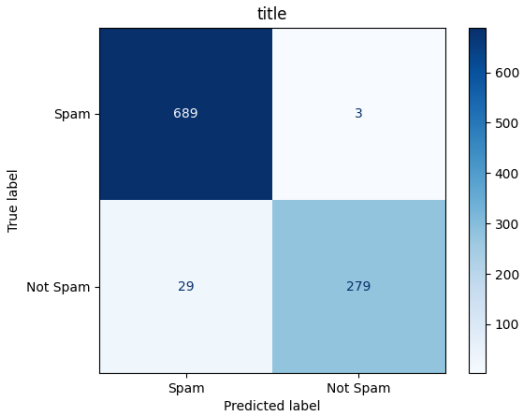

Figure 19: Confusion matrix non-linear SVM

We conclude that for "Spam" problem, a linear SVM model is going to classify better new parameters than for example an Gaussian SVM model, and for this problem the *kernel trick* is not useful at all because the data are lineally separable and if we use it, the accuracy decreases.