Assignment 2
FMAN45
Machine Learning

Solving a nonlinear kernel SVM with hard constraints, The Lagrangian dual of the soft margin SVM, Dimensionality reduction on MNIST using PCA, Clustering of unsupervised data using K-means, and Classification of MNIST digits using SVM

Author

Kajsa Hansson Willis

Lund University

Lunds universitet
Lunds Tekniska Högskola

Spring 2025

1 Exercise 1

In the first exercise, the aim was to compute the kernel matrix, K, with the help of given values. The kernel matrix is described by Equation 1, further described by Equation 2. The values can be seen in the x_i row of Table 1.

$$K = [k(x_i, x_j)]_{1 \leq i,j \leq 4} \tag{1}$$

$$k(x, y) = \Phi(x)^T \Phi(y) \text{ where } \Phi(x) = \binom{x}{x^2} \tag{2}$$

i	1	2	3	4
x_i	-2	-1	+1	+2
y_i	+1	-1	-1	+1

Table 1: The values of the data points x_i and y_i for exercise 1, 2 and 3

This gives the following feature maps $\Phi(x_1) = \binom{-2}{4}$, $\Phi(x_2) = \binom{-1}{1}$, $\Phi(x_3) = \binom{1}{1}$, and $\Phi(x_4) = \binom{2}{4}$. The kernel matrix is then calculated to be:

$$K = \begin{bmatrix} 20 & 6 & 2 & 12 \\ 6 & 2 & 0 & 2 \\ 2 & 0 & 2 & 6 \\ 12 & 2 & 6 & 20 \end{bmatrix}$$

2 Exercise 2

The purpose of the second exercise is to solve the maximization problem in Equation 3 for α assuming that $\alpha = \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$ with the given information.

$$\max_{\alpha_1,\alpha_2,\alpha_3,\alpha_4} \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{4} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \text{ subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{4} y_i \alpha_i = 0,\ \forall i \tag{3}$$

The expression in Equation 3 can be rewritten according to Equation 4 because it can be itemized to a single α as it is known that they are all equal. The values from Table 1 can then be inserted.

$$
\begin{aligned}
&\max_\alpha 4\alpha - \frac{1}{2}\sum_{i,j=1}^{4} \alpha^2 y_i y_j k(x_i, x_j) = \\
&= \max_\alpha 4\alpha - \frac{\alpha^2}{2}(1*1*20 + 1*-1*6 + 1*-1*2 + 1*1*12 - 1*1*6 \\
&\quad -1*-1*2 - 1*-1*0 - 1*1*2 - 1*1*2 - 1*-1*0 - 1*-1*2 \\
&\quad -1*1*6 + 1*1*12 + 1*-1*2 + 1*-1*6 + 1*1*20) \\
&= \max_\alpha 4\alpha - \frac{\alpha^2}{2}(20 - 6 - 2 + 12 - 6 + 2 - 2 - 2 + 2 - 6 + 12 - 2 - 6 + 20) \\
&= \max_\alpha 4\alpha - \frac{\alpha^2}{2}(36) = \max_\alpha 4\alpha - 18\alpha^2
\end{aligned}
\tag{4}
$$

In order to get the maximum, the derivative can be taken and set to zero, which yields Equation 5.

$$\frac{d(4\alpha - 18\alpha^2)}{d\alpha} = 4 - 36\alpha = 0 \longrightarrow \alpha = 4/36 = \frac{1}{9} \tag{5}$$

The value of α, and thus $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, is found to be 1/9, or approximately 0,11. As the alpha values are the same and the sum of the y-values is zero (1+1-1-1=0), both conditions are true for $\alpha = \frac{1}{9}$.

3 Exercise 3

The aim of exercise 3 is to obtain the polynomial x from the data-target pairs in Table 1 and the α-value from exercise 2 to reduce the classifier function, $g(x)$, to its simplest possible form.

The obtained value of α is $\frac{1}{9}$. The classifier function is expressed in Equation 6 where b is a bias term. The expression is simplified to Equation 7.

$$g(x) = \sum_{j=1}^{4} \alpha_j y_j k(x_j, x) + b \tag{6}$$

$$
\begin{aligned}
g(x) &= \frac{1}{9}(1*k(x_1,x) - 1*k(x_2,x) - 1*k(x_3,x) + 1*k(x_4,x)) + b \\
&= \frac{1}{9}((-2\ 4)\binom{x}{x^2} - (-1\ 1)\binom{x}{x^2} - (1\ 1)\binom{x}{x^2} + (2\ 4)\binom{x}{x^2}) + b \\
&= \frac{1}{9}(-2x + 4x^2 + x - x^2 - x - x^2 + 2x + 4x^2) + b \\
&= \frac{1}{9}(6x^2) + b = \frac{2}{3}x^2 + b
\end{aligned}
\tag{7}
$$

In order to determine b, the known relationship in Equation 8, where x_s is any support vector, can be leveraged. Inserting $s = 1$ gives Equation 9.

$$y_s(\sum_{j=1}^{4}\alpha_j y_j k(x_j, x_s) + b) = 1 \tag{8}$$

$$1(\sum_{j=1}^{4}\alpha_j y_j k(x_j, x_1) + b) = 1 = g(-2) = \frac{2}{3}(-2)^2 + b = \frac{8}{3} + b \longrightarrow b = 1 - \frac{8}{3} = -\frac{5}{3} \tag{9}$$

Thus, the classifier function can be reduced to $g(x) = \frac{2}{3}x^2 - \frac{5}{3}$.

4 Exercise 4

In the fourth exercise, the aim is to simplify $g(x)$ of the nonlinear kernel SVM with a new set of values of x_i and y_i, obtained from Table 2.

i	1	2	3	4	5	6	7
x_i	-3	-2	-1	0	1	2	4
y_i	+1	+1	-1	-1	-1	+1	+1

Table 2: The values of the data points x_i and y_i for exercise 4

In order to determine the classifier function for A SVM, the support vectors must be determined. These happen when the classification changes, i.e. for $x = \{-2, -1, 1, 2\}$ where -2 and -1 mark the change from +1 to -1 and 1 and 2 mark the change from -1 to +1. These are the values that will decide where the cutoff line, the classifier function, is.

Looking at the support vectors, it is apparent that they are the same as the vectors in the previous exercise, which can be viewed in Table 1. Because the support vectors are exactly the same, ie the new table is just more classifications added on the right side of the classifier function, the classifier function will also be the same. Therefore, the solution to the nonlinear kernel SVM with hard constraint on the dataset in Table 2 is $g(x) = \frac{2}{3}x^2 - \frac{5}{3}$.

5 Exercise 5

In the fifth exercise, the aim is to show that the Lagrangian dual problem for Equation 10 is given by Equation 11, where ξ is the error term for the data points x, w is the parameter, and C a constant.

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \text{ subject to } y_i(w^T x_i + b) \geq 1 - \xi_i,\ \forall_i, \xi \geq 0,\ \forall_i \tag{10}$$

$$\max_{\alpha_1,..,\alpha_n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \text{ subject to } 0 \leq \alpha_i \leq C,\ \forall_i, \sum_{i=1}^{n} \alpha_i y_i = 0,\ \forall_i \tag{11}$$

The corresponding Lagrangian function is represented by Equation 12.

$$L(w, b, \xi) = \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) + \xi_i - 1) - \sum_{i=1}^{n} \lambda_i \xi_i, \ \ \alpha_i, \lambda_i \geq 0 \tag{12}$$

The partial derivatives of the Lagrangian function with respect to w, b, ξ are then taken, which are expressed in Equations 13, 14 and 15.

$$\frac{dL}{dw} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{13}$$

$$\frac{dL}{db} = - \sum_{i=1}^{n} \alpha_i y_i = 0 \rightarrow 0 = \sum_{i=1}^{n} \alpha_i y_i \tag{14}$$

$$\frac{dL}{d\xi_i} = C - \alpha_i - \lambda_i = 0 \rightarrow C = \alpha_i + \lambda_i \tag{15}$$

As $C = \alpha_i + \lambda_i$ and $\lambda_i \geq 0$ according to Equation 15, it must be true that $0 \leq \alpha_i \leq C$. Thus, the conditions described in Equation 11 hold.

Equation 13 can be plugged into Equation 12 which yields Equation 16 after inserting Equations 14 and 15 as well.

$$\begin{aligned} L(w, b, \xi) = \frac{1}{2}||\sum_{i=1}^{n} \alpha_i y_i x_i||^2 + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i (\sum_{i=1}^{n} \alpha_i y_i x_i)^T x_i + b) + \xi_i - 1) - \sum_{i=1}^{n} \lambda_i \xi_i = \\ = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + C \sum_{i=1}^{n} \xi_i (C - \alpha_i - \lambda_i) + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_1 y_i b \\ = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j, \ \ \alpha_i, \lambda_i \geq 0 \end{aligned} \tag{16}$$

It has thus been shown that Equation 11 is the Lagrangian dual problem, since it is equal to Equation 16 and the right conditions hold.

6 Exercise 6

In exercise 6, the goal is to use complementary slackness of the KKT conditions to show that that support vectors with $y_i(w^T x_i + b) < 1$ have coefficient $\alpha_i = C$.

The complementary slackness condition from the KKT says that $\lambda_i \xi_i = 0,\ i = 1, 2, ..., m$. It is known that if strong duality holds and x, λ, α are optimal, then the KKT conditions hold. Inserting that $\lambda_i = C - \alpha_i$ from Equation 15, Equation 17 is obtained. If $y_i(w^T x_i + b) < 1$, there must be an error, ie $\xi \neq 0$ which is known from Equation 10. This means that λ_i, or $C - \alpha_i$, must be zero. This is expressed in Equation 18. Thus, it has been proven that $C = \alpha_i$ if $y_i(w^T x_i + b) < 1$.

$$(C - \alpha_i)\xi_i = 0 \tag{17}$$

$$C - \alpha_i = 0 \rightarrow C = \alpha_i \tag{18}$$

7 Exercise 7

The purpose of the seventh exercise was to compute a linear principal components analysis (PCA) and then visualize the training data in two dimensions, ie with two principal components.

The first step was to ensure that the requirements for PCA were met, which meant removing the mean from the training data, as the data must be zero-mean. The Matlab function *svd* was then utilized to find the principal components, as it performs a singular value decomposition and returns the principal components. The training data was the projected on the two principal components and a scatterplot was created with the result, which can be seen in Figure 1. The different digits can be distinguished quite well in two dimensions, although there is some overlap.

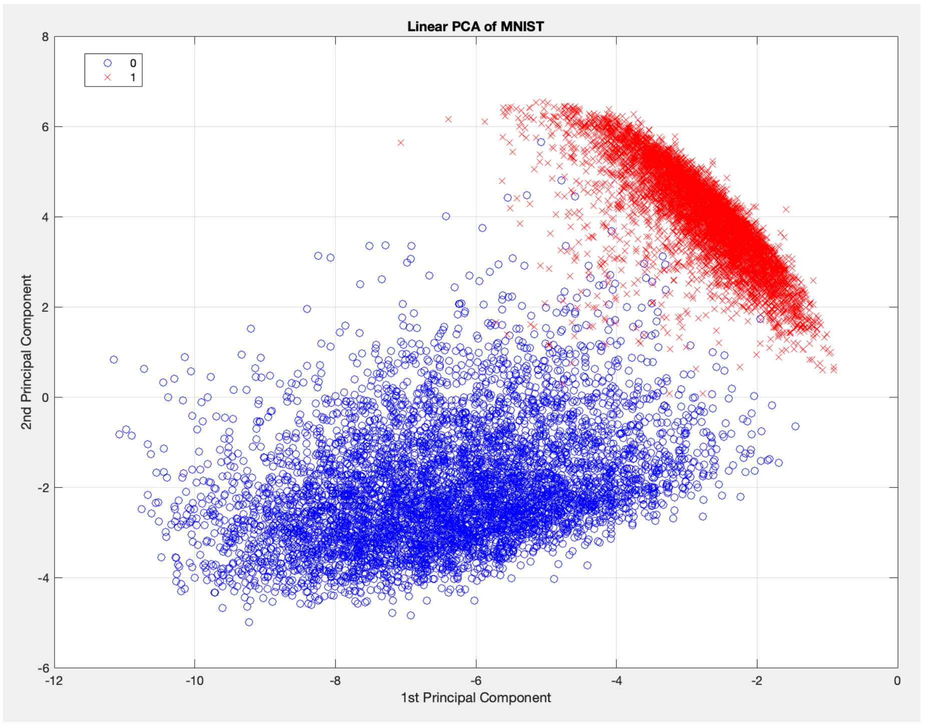

Figure 1: Scatterplot of the principal components analysis in exercise 7 with blue circles for the training data labeled as a zero and red crosses for the training data labeled as a one

8 Exercise 8

In the eight exercise, the purpose was to implement K-means clustering for the training data. An outline of a Matlab function was provided and completed for this task.

The procedure involved first implementing distance functions to calculate the distance between a data point and all of the centroids and the distance between any two centroids. The distance functions were then utilized in functions to assign clusters to individual data points and to compute the centroids of the clusters and how far the centroid has shifted in each iteration.

This function was then applied for both two and five clusters. This was illustrated by applying the PCA technique with two principal components from exercise 7 and using the clusters as labels. The results with two clusters can be observed in Figure 2 and the results with five clusters can be observed in Figure 3.

Small amounts of overlap can be found in the two-cluster plot, but this is quite insignificant compared to the five-cluster plot which has much overlap. It may seem strange that a data point may appear closer to another centroid but

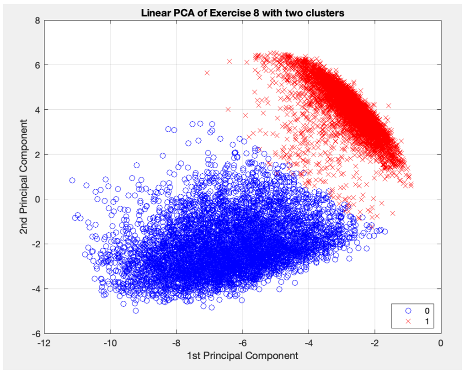

Figure 2: Visualization of PCA for two clusters

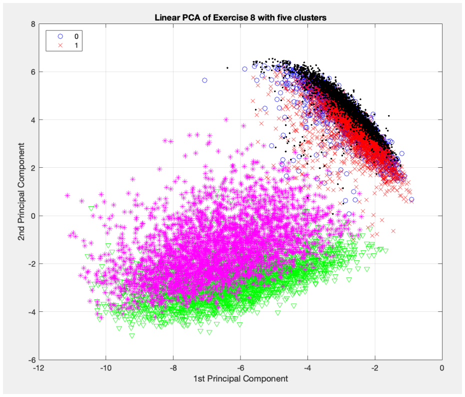

Figure 3: Visualization of PCA for five clusters

still be assigned a different centroid. The explanation for the overlap is because the dimension is reduced from 784, where the classification is done, to 2, in the plot, which is a very large jump so it is quite natural. If one were to add a third dimension, it is likely that much of the cluster overlap would disappear. With fewer clusters it becomes simpler to separate them as well, as can be seen when comparing the plots.

9 Exercise 9

In the ninth exercise, the aim was to visualize the clustering by displaying the centroids as images for both 2 and 5 centroids.

This was done by first stacking the centroids back into the right shape $x_i \in R^{28x28}$ with the Matlab function *reshape*. The Matlab function *imshow* was then used to display the clusters as images. For two clusters, the results can be viewed in Figure 4. For five clusters, the results are displayed in Figure 5.The images for five clusters are just a little bit clearer than the ones for two clusters.

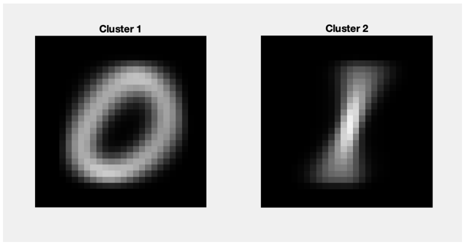

Figure 4: Display of the centroid with two clusters as an image

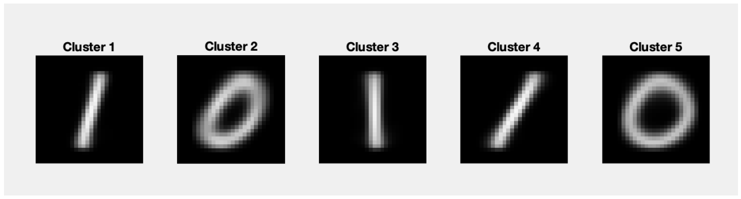

Figure 5: Display of the centroid with five clusters as an image

10 Exercise 10

In exercise 10, the aim was to use the K-means clustering for classification by completing a number of steps. This was done by implementing a new function in Matlab, *K_ means_ classifier* that assigns a data point to its closest centroid by returning its index. This was then used to assign each cluster centroid the label, 0 or 1, that the majority of the data points have using the training data. Using this information, the K-means clustering for classification method was evaluated by comparing the predictions to the true labels for both the training and the testing data. The results from this are found in Table 3. Additionally, a PCA plot, similar to in exercise 8, was created to illustrate the results of the clustering method which can be viewed in Figure 6.

Table 3: K-means classification results with K=2

Training data	Cluster	# '0'	# '1'	Assigned to class	# misclassified
	1	5 809	114	1	114
	2	6	6 736	0	6
$N_{\text{train}} = 12\ 665$				Sum misclassified:	120
				Misclassification rate (%):	0,95
Testing data	**Cluster**	**# '0'**	**# '1'**	**Assigned to class**	**# misclassified**
	1	968	12	1	12
	2	0	1 135	0	0
$N_{\text{test}} = 2\ 115$				Sum misclassified:	12
				Misclassification rate (%):	0,57

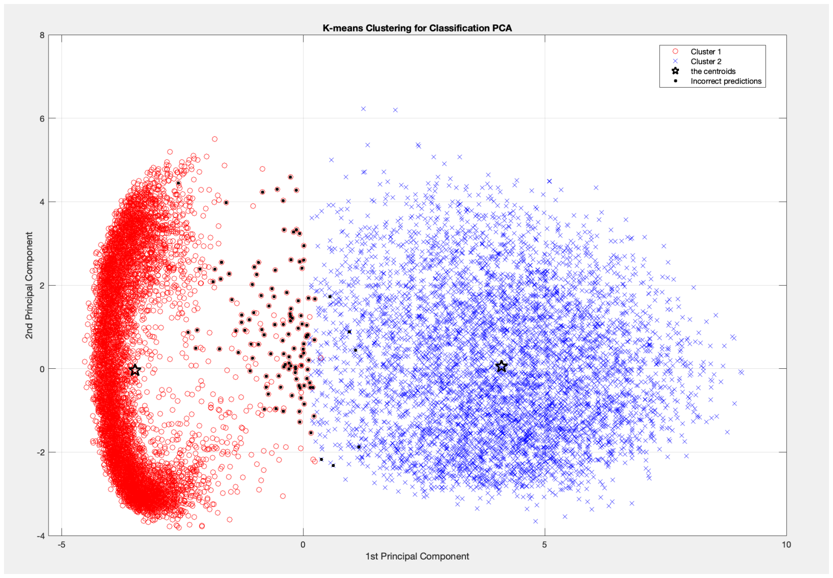

Figure 6: Linear PCA of the two-cluster analysis with the centroids and the incorrect predictions marked out

11 Exercise 11

In exercise 11, the purpose is to explore if the misclassification rate can be lowered further on test data by considering a different amount of clusters, K. As it is obvious that fewer than two clusters will be worse, larger numbers of K were tested. This was done by first calculating the clusters and then applying the same principles as in exercise 10. The number of clusters that were considered

K	*Misclassification rate for training data (%)*	*Misclassification rate for testing data (%)*
2	0,95	0,57
3	0,66	0,52
5	0,37	0,19
10	0,24	0,19
20	0,22	0,09
30	0,18	0,09

Table 4: The misclassification rates for different numbers of clusters, K

were K=3, 5, 10, 20 and 30. A summary of the misclassification rates for the different amounts of clusters can be found in Table 4.

For K=3, new clusters had to be created using the *K_means_clustering*-function previously explained. The clusters were then labeled 0,0,1 respectively. For the training data there were 84 misclassifications, corresponding to a misclassification rate of 0,66%. For the testing data, there were 11 misclassifications corresponding to a rate of 0,52%.

For K=5, the clusters from exercise 8 were utilized. The clusters were assigned the labels 0,0,1,0, and 1, respectively. For the training data there were 47 misclassifications, which corresponds to a misclassification rate of 0,37%. For the testing data there were 4 misclassifications corresponding to a 0,19% misclassification rate. This is a significant improvement from K=2 and K=3.

For K=10, new clusters had to be created using the *K_means_clustering*-function. The clusters were labeled 1, 0, 0, 1, 0, 1, 1, 0, 0, and 0, respectively. For the training data there were 31 misclassifications, corresponding to a misclassification rate of 0,24%. For the testing data, there were 4 misclassifications corresponding to a rate of 0,19%. The misclassification rate for the training data is significantly lower than the previous values, whereas it is the same as for K=5 for the testing data.

For K=20, new clusters were again created with the *K_means_clustering*-function. For the training data there were 28 misclassifications, corresponding to a misclassification rate of 0,22%. For the testing data, there were 2 misclassifications corresponding to a rate of 0,09%. Again, it is apparent that the misclassification rate has decreased compared to with fewer clusters.

For K=30, new clusters were again created with the *K_means_clustering*-function. For the training data, there were 23 misclassifications, corresponding to a misclassification rate of 0,18%. For the testing data, there were 2 misclassifications corresponding to a rate of 0,09%. This is the lowest misclassification rate for the training data from the tested amounts of clusters, and the same as the for K=20 for the testing data, which are both the lowest value.

This shows that it is indeed possible to lower the misclassification rate on test data by considering a different (larger) amount of clusters from K=2. There seems to be a pattern of the misclassification rate decreasing, or staying the same, with increasing K.

12 Exercise 12

In exercise 12, the supervised training data, ie the training data and the training labels, were used to train a linear SVM classifier. This was done with the help of the Matlab function *fitcsvm* which trains a binary soft-margin SVM with the default error parameter $C = 1$. The Matlab function *predict* was then used to obtain class predictions using the model generated by *fitcsvm*. The classification results can be seen in Table 5. For the test data, there are only two misclassifications which corresponds to a misclassification rate of 0,09%.

Table 5: Linear SVM classification results

Training data	Predicted class	True class: # '0'	# '1'
	'0'	5923	0
	'1'	0	6742
$N_{\text{train}} = 12\ 665$		Sum misclassified:	0
		Misclassification rate (%):	0
Testing data	**Predicted class**	**True class: # '0'**	**# '1'**
	'0'	979	1
	'1'	1	1134
$N_{\text{test}} = 2\ 115$		Sum misclassified:	2
		Misclassification rate (%):	0,0946

13 Exercise 13

In exercise 13, the supervised data was used to train a non-linear kernel SVM classifier using a Gaussian kernel. The Matlab function, from exercise 12, *fitcsvm*, was used again, but this time adjusted for the Gaussian kernel.

The β-parameter is a scaling parameter used in the function. For the standard $\beta = 1$, there were 388 misclassified data points among the test data; all of them were ones that were incorrectly predicted as zeros. This was modified in order to obtain the best possible results, so other β-values were tested as well, including 0,5, 1,2 2, 5, 6, 7, 25, and 100. The misclassifications can be viewed in Table 6. It can be noted that $\beta = 5$ and $\beta = 6$ are the best as it gives zero errors. It was further noted that all the errors for β-values lower than the optimal one were ones incorrectly predicted as zeros, whereas for the larger β-values, it was the opposite instead.

The best (perfect) classification results can be observed in Table 7 where $\beta = 5$.

β**-value:**	**Misclassifications for test data:**
0,5	980
1	388
1.2	305
2	76
5	0
6	0
7	1
25	2
100	2

Table 6: The amount of misclassifications for different β-values in exercise 13

Table 7: Gaussian kernel SVM classification results for $\beta = 5$

Training data	Predicted class	True class: # '0'	# '1'
	'0'	5 923	0
	'1'	0	6 743
$N_{\text{train}} = 12\ 665$		Sum misclassified:	0
		Misclassification rate (%):	0
Testing data	Predicted class	True class: # '0'	# '1'
	'0'	980	0
	'1'	0	1135
$N_{\text{test}} = 2\ 115$		Sum misclassified:	0
		Misclassification rate (%):	0

14 Exercise 14

Although it is possible to achieve a very low misclassification rate on both training and testing data with a good choice of parameters for the Gaussian kernel SVM, it is false to expect the same error on new images.

This is because selecting hyperparameters based on them fitting the testing data well and trying to achieve perfection is likely to lead to overfitting. It can be seen as a way of indirectly using the test data as training data, which means the purpose of the test data is lost.

The data is not hypersensitive to the choice of β as the errors are low for what seems to be a quite large range, so it is possible that the beta value does not influence to a large extent. If so, it should be possible to still achieve good results on new images, but this is not certain.