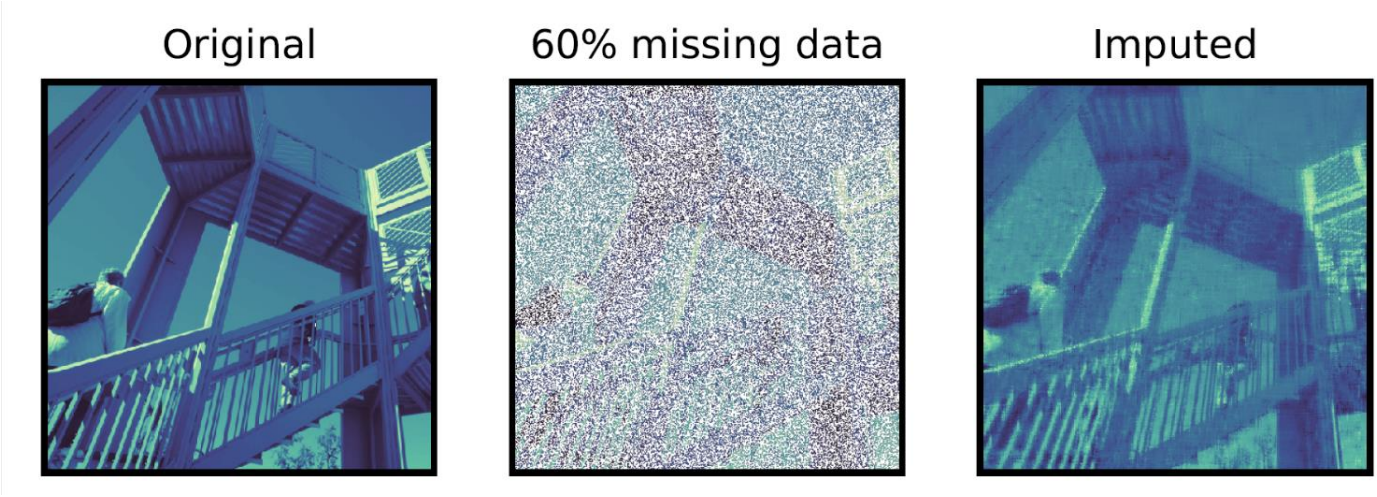
Homework 3 in Data Mining 2

Anja Lieberherr

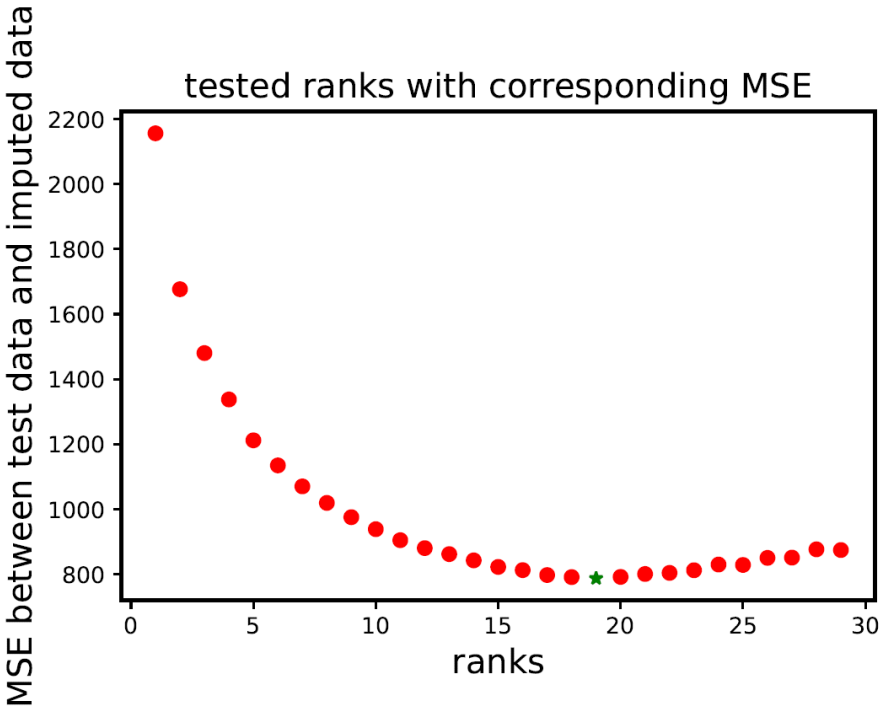
21.04.2021

Exercise 1: Data Imputation via SVD

a) The rank r with the minimum mean squared error was used to impute the original data matrix. The results of the imputation are shown below: the original image and the imputed image.

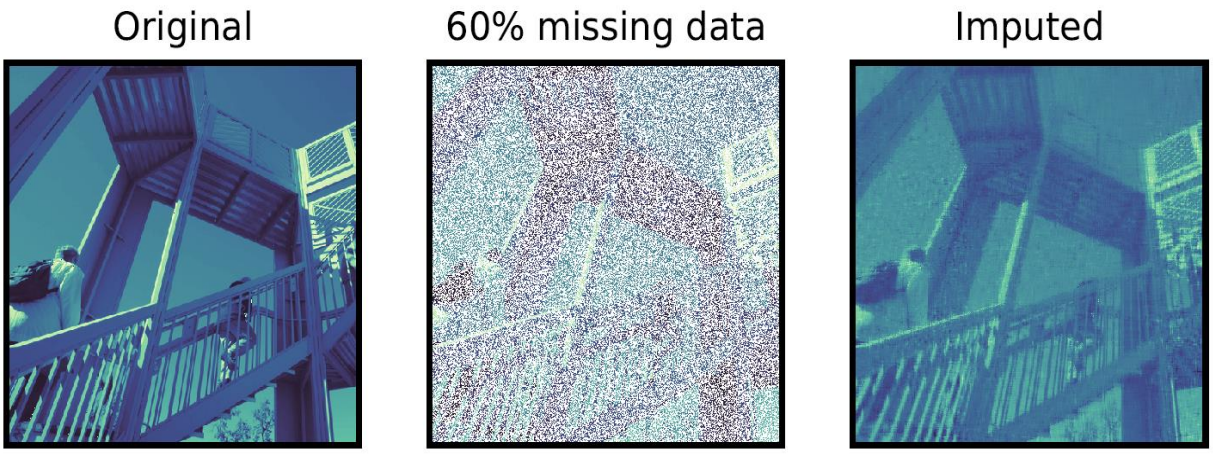


b) A plot is shown below with all the tested ranks r and the corresponding mean squared errors. The optimal rank r is highlighted in green. The optimal rank r is 19.000000 and the corresponding Mean-Squared Error is 787.04.

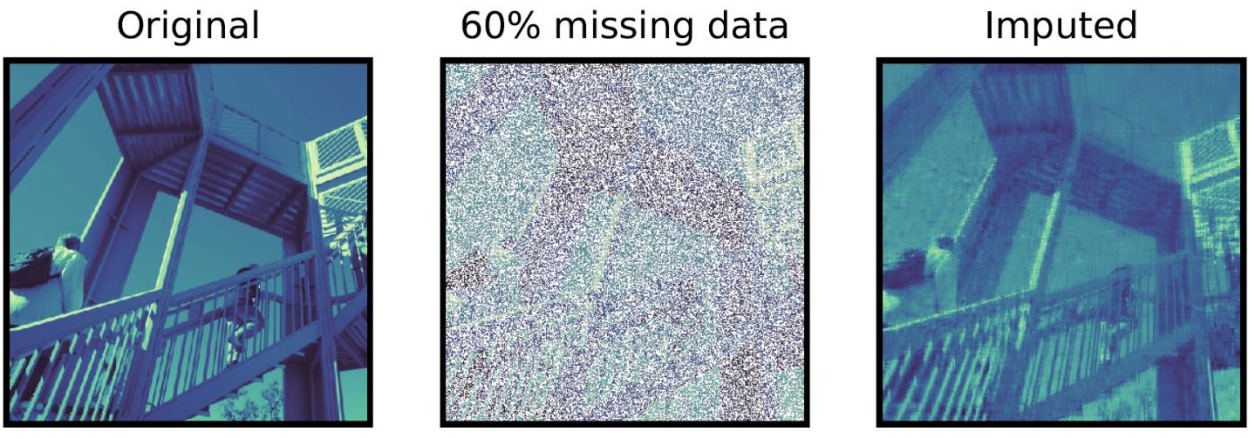


c) What happens if a too large or a too small value for rank r is chosen?
One can expect the imputed image to be a worse approximation of the original image when a too large or too small rank is used for the imputation. On the plot above, on can see that the MSE between the true test points and the imputed test points is high for small ranks and increasing for too large ranks.
The following three plots show the imputed image using a too high rank = 28, the optimal rank = 19 for comparison and a too small rank = 8.
As one would expect, the imputed image with rank 8 is really a bad approximation. This shows that finding the optimal rank is crucial. All in all, one can say that choosing a too high rank leads to an increased MSE due to of overfitting.

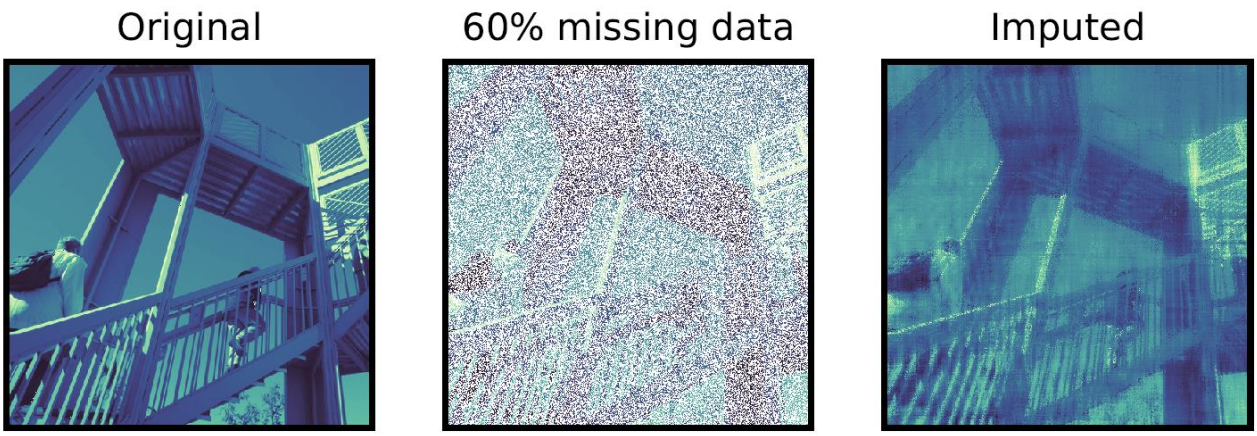
Rank = 28



Optimal Rank = 19

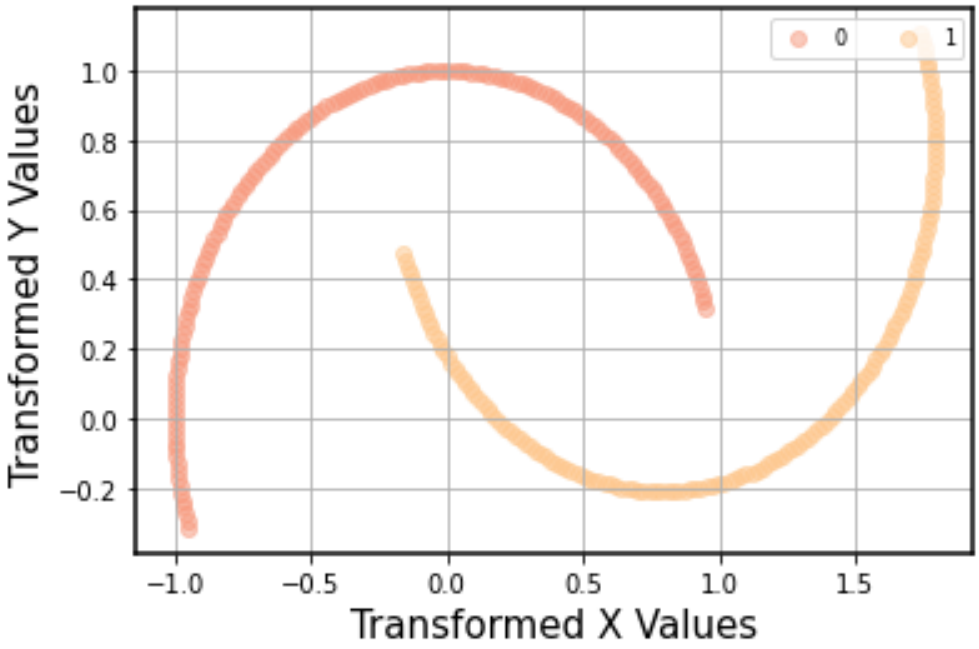


Rank = 8



Exercise 2: Kernel PCA

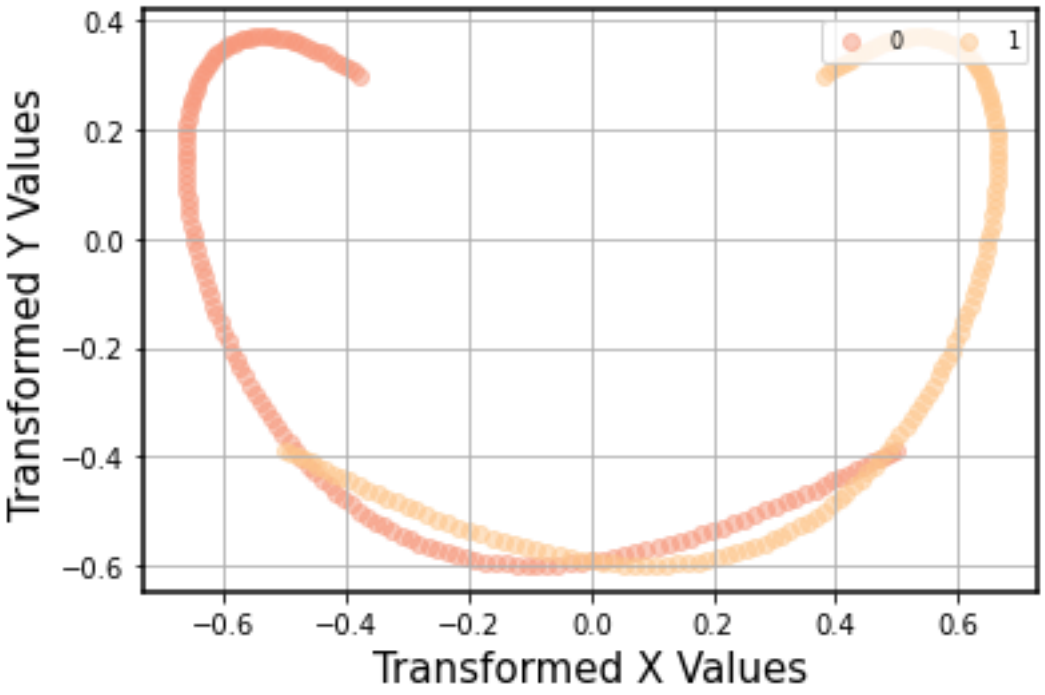
a) Plot of PCA like in HW1 on the Moon data
Variance Explained Exercise 2a:
PC 1: 0.82
PC 2: 0.18
One can see that the first two PCs explain the complete variance (0.82 + 0.18 = 1).
Clearly, one can see in the resulting figure that a linear classifier is unable to perform well on the moon dataset transformed via standard PCA



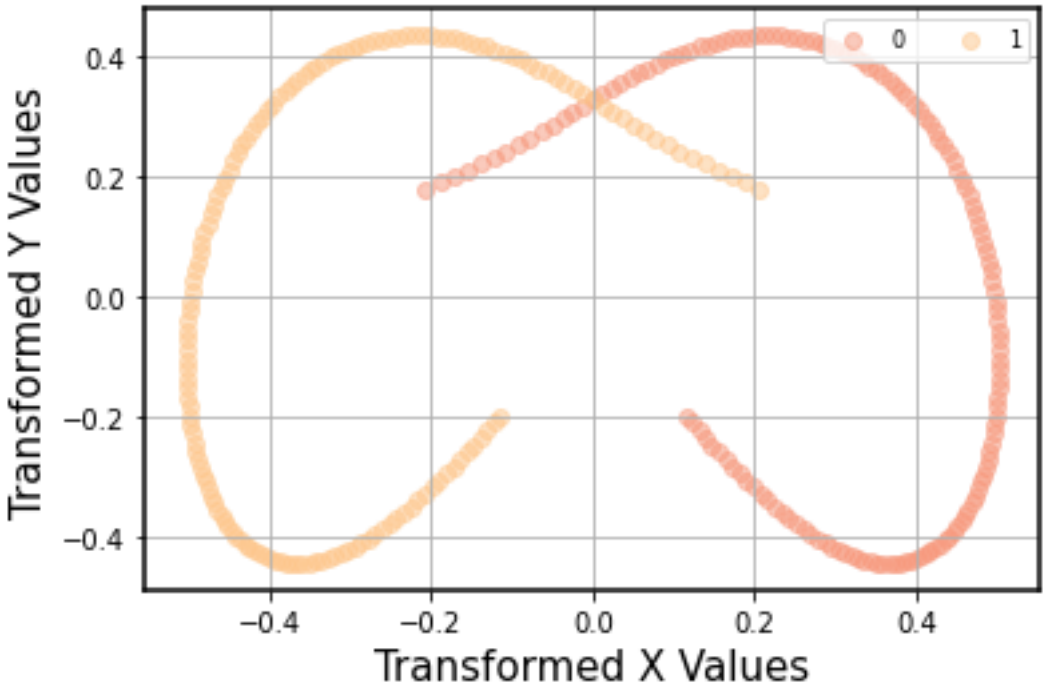
b) Kernel PCA with a Gaussian radial basis function (RBF) with hyperparameter γ. The function RBFKernelPCA() is implemented in the file pca.py.

c) The function RBFKernelPCA() is applied for the following γ = [1,5,10,20] and the resulting plots are listed below.
The plots for γ = 1,5,10 show that the two classes are overlapping and are not very well separated. However, using RBFKernelPCA with the appropriate value for gamma, separates the two classes linearly. The best separation of the two classes is obtained for γ = 20. RBFKernelPCA enabled to compress datasets consisting of nonlinear features onto a lower-dimensional subspace where the classes became linearly separable.
How to find the optimal value for γ? There is no universal value for the tuning parameter gamma that works well for different datasets. Finding a value that is appropriate for a given problem requires experimentation. One can use cross-validation or grid search to find the best gamma.
What does gamma? Technically, the gamma parameter is the inverse of the standard deviation of the RBF kernel (Gaussian function), which is used as similarity measure between two points.

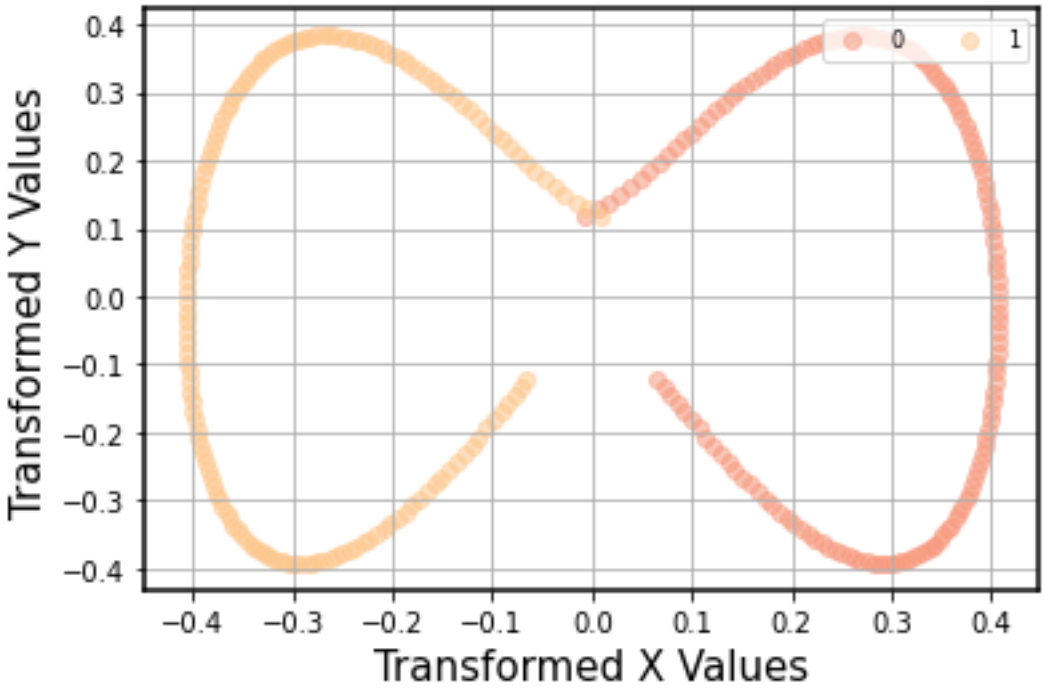
Gamma = 1



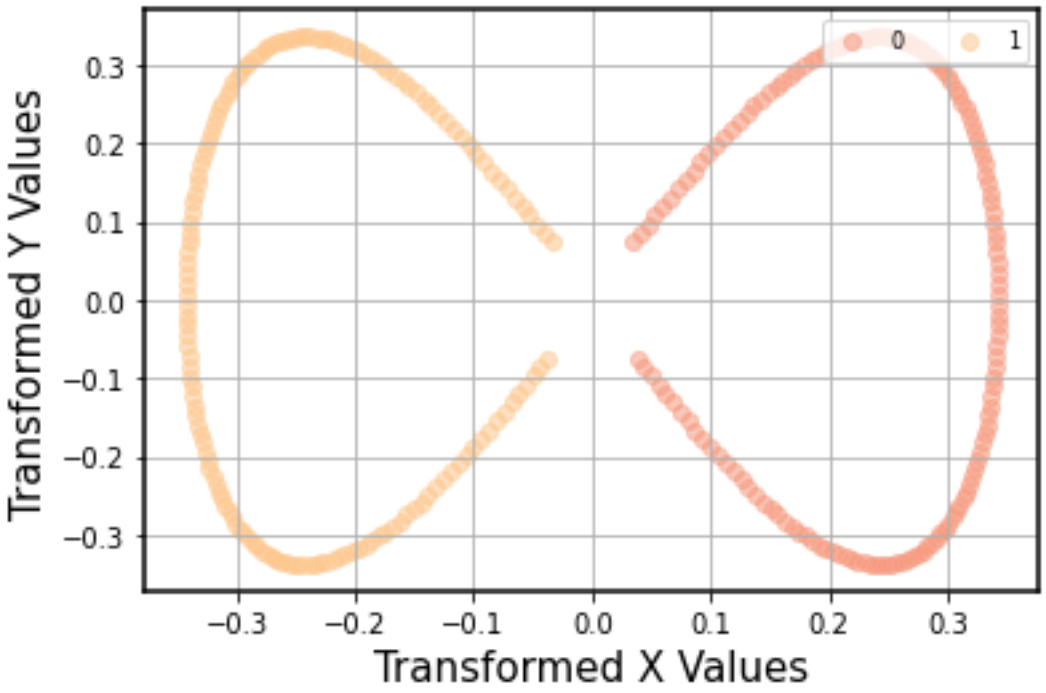
Gamma = 5



Gamma = 10



Gamma = 20


0
1
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
Transformed Y Values
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
Transformed X Values