ELE 489: Fundamentals of Machine Learning
Prof. Seniha Esen Yuksel
Department of Electrical and Electronics Engineering
Hacettepe University

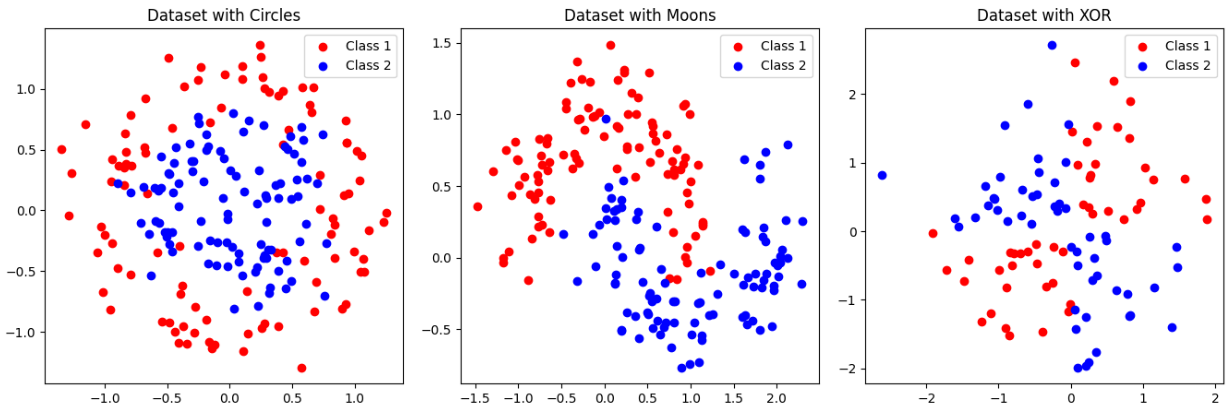
Gaye AKSU
2200357047

HW-5

In this homework, we will work with Support Vector Machines (SVM). First, we will create a toy dataset that cannot be separated by a line. Then, I will try to classify the data using a linear SVM and a non-linear SVM with a different kernel and check whether it works or not and compare both results. Finally, I will visualize the decision boundaries and discuss which method is better.

Q1) Generate your own 2D toy data that you think will not work with linear SVMs, but will work with non-linear SVMs. Generate and scatter-plot two classes of 2D points that are not linearly separable.

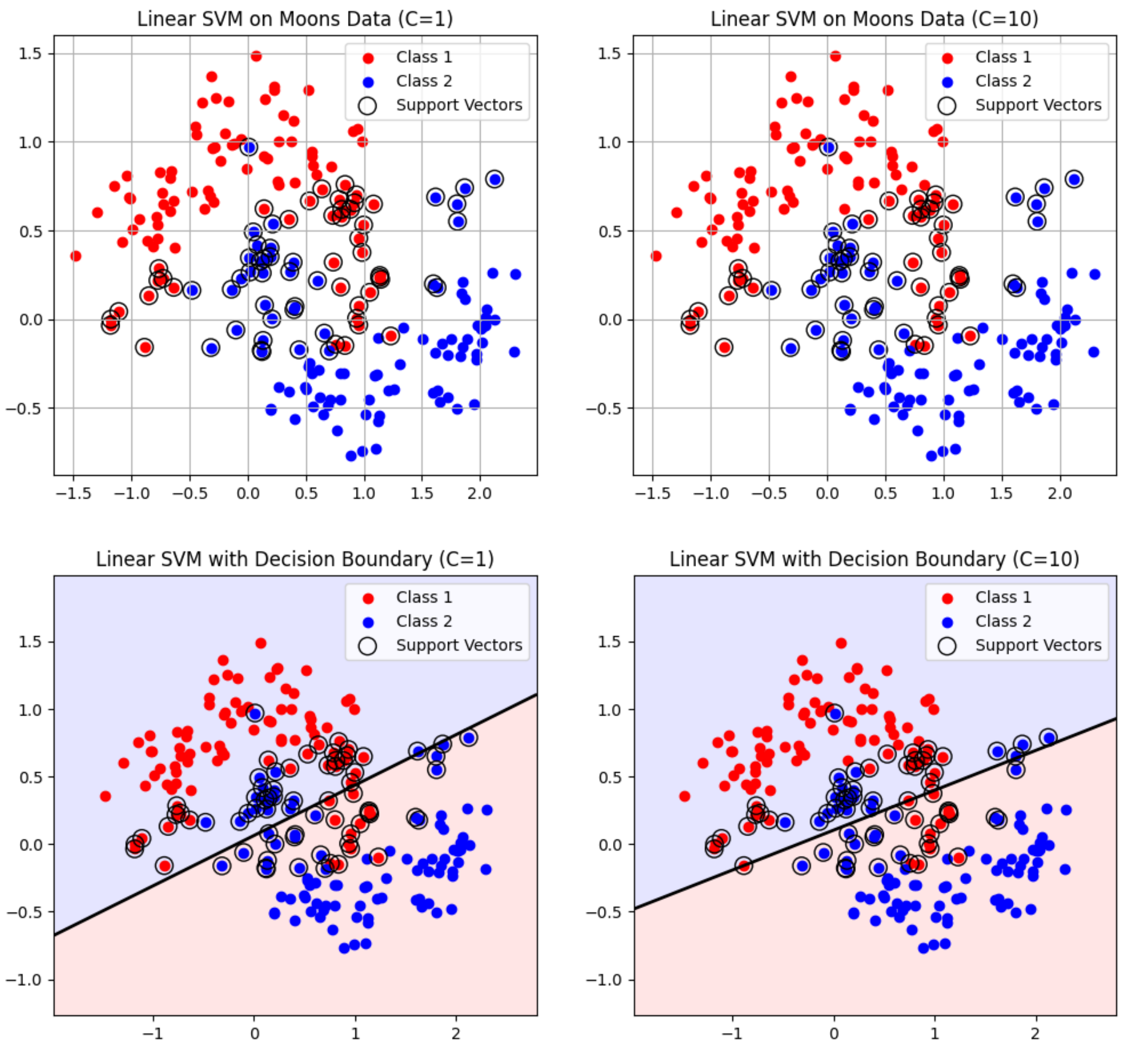
In this question, we are asked to create toy data. A toy dataset is a simple, artificial dataset used for testing and seeing how machine learning models work. In this question, the toy datasets that we generate cannot be separated by a linear SVMs, but can be separated by a non-linear SVMs, such as circles, moons, spirals, and XOR shapes. The graphs below show the circle, moon, and XOR datasets. For the next parts of this homework, I will use the moons dataset.



As seen in the figures above, these datasets cannot be separated by a linear SVM. However, a non-linear SVM can classify them correctly.

Q2) Run a linear SVM and show your results in 2D, on the data itself. Show the support vectors. Evaluate the linear SVM's performance (e.g., margin size, misclassifications) and briefly discuss why it succeeded or failed.

Like I said, for the rest of the homework, I will be using the moons dataset. The reason I chose the moons dataset is that it's a classic example of non-linear classification problems, and it's easy to visualize. To see the effect of margin size, I used linear SVMs with different C values. The value of C controls the margin size; a small C gives a wider margin and allows more mistakes, while a large C makes the margin narrower and tries to reduce mistakes. You can see the 2D plots with the support vectors and decision boundaries for different C values in below.

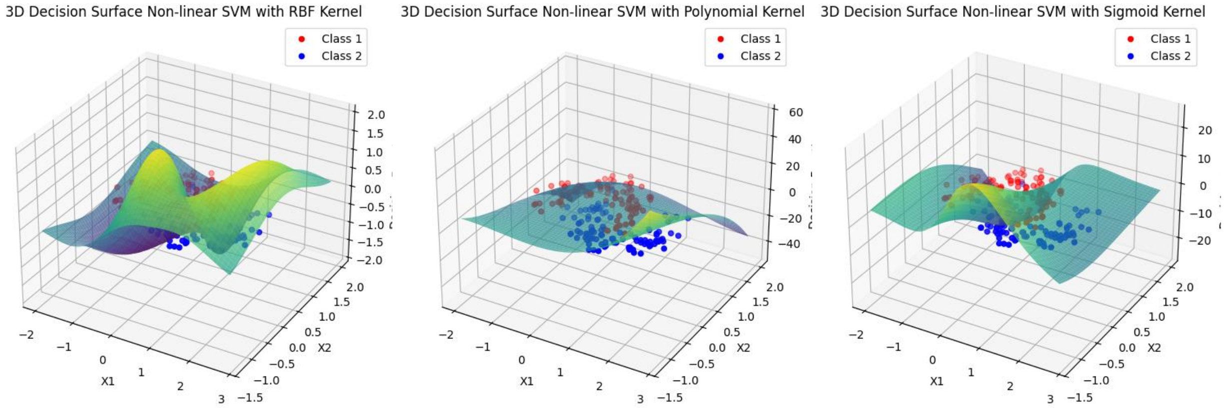


When we look at these plots, we can see that there are only small changes when the C value changes from 1 to 10. The support vectors and the decision boundary are almost the same. This is because the data is not linearly separable, so changing the C value does not make a big difference. Some points near the boundary may change, but the overall result stays similar.

```
Accuracy and Misclaasification calculations for C=1
Linear SVM Accuracy: 0.85
Number of misclassified points: 30
Accuracy and Misclaasification calculations for C=10
Linear SVM Accuracy: 0.83
Number of misclassified points: 33
```

The results show that increasing the C value does not always reduce the number of misclassifications for the moons dataset. This is because the data is not linearly separable, and even though the C value is increased, the linear SVM still cannot achieve a perfect boundary for this dataset. Many points are still classified incorrectly, and sometimes the accuracy can even decrease a little when C increases. This happens because the model tries too hard to fit the data, known as overfitting. Overall, the linear SVM could not separate the two classes effectively. For this dataset, a linear boundary is not enough, so we need to try a non-linear SVM for better results.

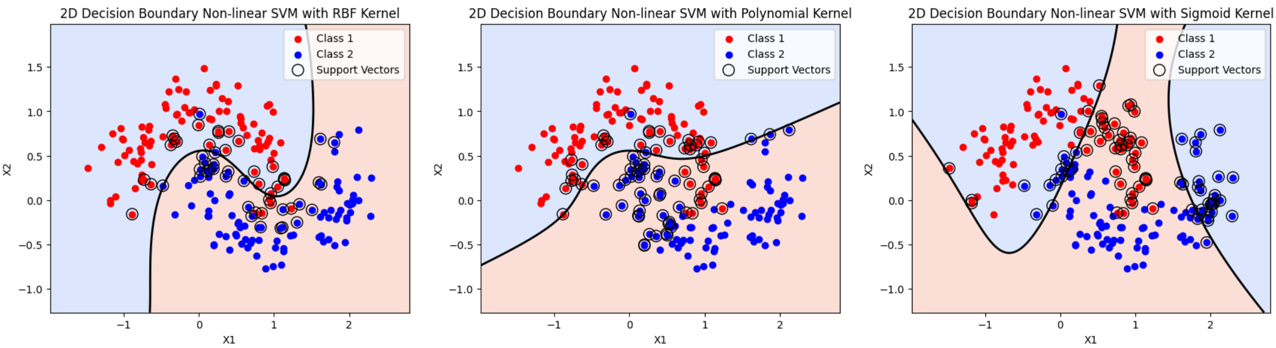
Q3) Now try to classify this toy data with non-linear SVMs. Which kernel did you choose? Plot the decision surface along with the data as a 3D plot.



Kernel Type	Accuracy	Misclassifications
RBF Kernel	0.96	8
Polynomial Kernel	0.92	17
Sigmoid Kernel	0.66	68

When I compare the non-linear SVM kernels with the linear SVM, I see that the non-linear kernels give much better results for this dataset. The RBF kernel has the highest accuracy and the fewest misclassifications, while the linear SVM has more mistakes and a lower accuracy. The sigmoid kernel has the lowest accuracy. Because of these results, I choose the RBF kernel as the best for this dataset.

Q4) Plot the resulting decision boundary as a 2D plot. Mark the support vectors and show the classification results.



The plots above show the 2D decision boundaries of the RBF, polynomial, and sigmoid kernels. The RBF kernel gives the best boundary for this dataset, following the shape of the two classes very well. The polynomial kernel also fits the data but is not as smooth as RBF. The sigmoid kernel does not correctly separate the classes and makes more mistakes. In conclusion, the RBF Kernel Works best for this dataset.

Appendix:

You can see the code that I wrote for this homework in this GitHub link.

https://github.com/AksuGaye/ELE489_HW5.git