Support Vector Machines (SVMs)

Task 1:

Q1: What is the margin and support vectors?

Margin is the width that the boundary can be increase without hitting a data point

Support Vectors are the data points that the margin boundaries are pushed up against

Q2: How does SVM deal with non-separable data?

SVM uses a slack variable to allow some datapoint to fall into the margin space, or allow some misclassification and push the margin against some support vectors to be able to minimize the overall misclassifications. The process is to use soft margins instead of hard margins

Q3: What is a kernel?

A Kernel is a function that directly obtain the value of inner products and take the datapoints to a higher dimensional space without doing feature mapping

Q4: How does a kernel relate to feature vectors?

SVM uses a Kernel trick which engaging a Kernel function to compute the inner products of all pairs of data in the higher feature space in order to take data points into a higher dimensional feature space without ever computing the coordinates of the data in that space where the training data is linearly separable. The Kernel trick mapping process is a much cheaper computation than having to exclusively map the features into a higher dimensions.


Task 2: Construct a support vector machine that computes the kernel function. Use four values of +1 and -1 for both inputs and outputs:

- **[-1, -1] (negative)**
- **[-1, +1] (positive)**
- **[+1, -1] (positive)**
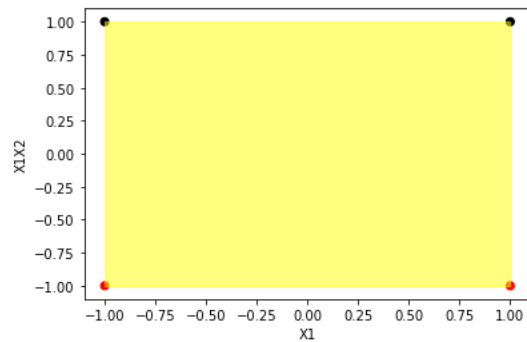- **[+1, +1] (negative).**

Map the input [x1, x2] into a space consisting of x1 and x1x2. Draw the four input points in this space, and the maximal margin separator. What is the margin?

I converted the point into a data frame and plotted the data.

	x1	x2	target	x1x2
0	-1	-1	0	1
1	-1	1	1	-1
2	1	-1	1	-1
3	1	1	0	1
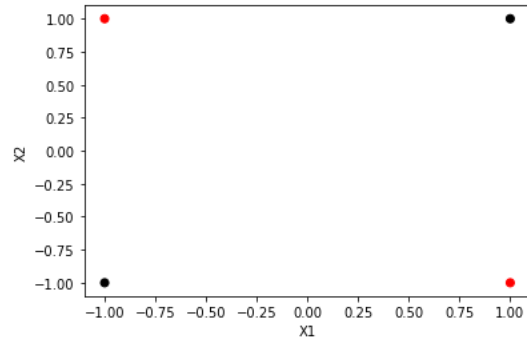
The points that have the target of Positive are Red and the points with Negative target are "Black".

Above we have the 4 points plotted in the [X1,X1X2] space and the maximal margin separators are the four edges of the yellow rectangle that pushed against the support vectors in this space.

The margin is the yellow rectangle that pushes against the support vectors and separate the area between positive target points and the negative target points.

Task 3: Recall that the equation of the circle in the 2-dimensional plane is (x1 - a)2 + (x2 - b)2 - r2 = 0. Please expand out the formula and show that every circular region is linearly separable from the rest of the plane in the feature space (x1, x2, x12, x22).

The circle equation expands into five terms

$$0 = x^2_1 + x^2_2 - 2ax_1 - 2bx_2 + (a^2 + b^2 - r^2)$$

Considering that the slopes of the variables x^2_1, x^2_2, x_1, x_2 are (1, 1, -2a, -2b) respectively.

And the intercept in this equation is $(a^2 + b^2 - r^2)$. Looking at the format of this equation, it looks like a linear regression equation in the four dimensional space of x^2_1, x^2_2, x_1, x_2.

It shows that the circle equation is linearly separable in the four dimensional feature space of $x^2_1, x^2_2, x1, x2$.

Task 4: Recall that the equation of an ellipse in the 2-dimensional plane is c(x1 - a)2 + d(x2 - b)2 - 1 =0. Please show that an SVM using the polynomial kernel of degree 2, K(u, v) = (1 + u ·v)2, is equivalent to a linear SVM in the feature space (1, x1, x2, x12, x22, x1x2) and hence that SVMs with this kernel can separate any elliptic region from the rest of the plane.

The ellipse equation expands into six terms
$0 = cx^2_1 + cx^2_2 - 2acx1 - 2bdx2 + (a^2c + b^2d - 1)$

The feature space for the above equation contains $x^2_1, x^2_2, x1, x2$ and the intercept is
$(a^2c + b^2d - 1)$

Using the polynomial kernel degree 2 it is going to use the kernel of

$K(x1,x2) = (1 + x1^T.x2)^2$ which the dot product of $x^T.x$ will create a matrix below and add an additional dimension of x1x2 to the equation.

$$x^T.x = \begin{bmatrix} x^2_1 & x2x1 \\ x1x2 & x^2_2 \end{bmatrix}$$

And this will confirm that the SVM algorithm with using the polynomial kernel of degree 2 can linearly separate every point on an ellipse in feature space $(x^2_1, x^2_2, x1, x2, x1x2, 1)$
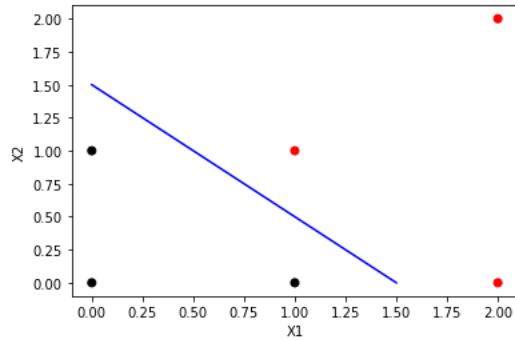
Task 5: Consider the following training data
(a) Plot these six training points. Are the classes {+, -} linearly separable?
Yes by looking at the graph, we can see that the classes are linearly separable
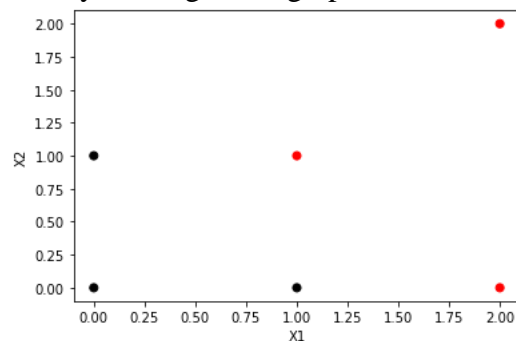
(b) Construct the weight vector of the maximum margin hyperplane by inspection and identify the support vectors.

The maximum margin hyperplane should have a slope of −1 and pass from the points of $(0, 1.5)^{\mathrm{T}}$ and $(1.5,0)^{\mathrm{T}}$ in order to be the maximum margin.

Therefore it's equation is $x_1 + x_2 = 3/2$, and the weight vector is $(1, 1)^{\mathrm{T}}$.

Task 6: Consider a dataset with 3 points in 1-D:

(class)	x
$+$	0
$-$	-1
$-$	$+1$

(a) Are the classes {+, -} linearly separable?
No the points are not linearly separable because the positive class of zero is in between the two negative classes of 1 and -1.

(b) Consider mapping each point to 3-D using new feature vectors f(x) = [1, sqrt(2)x, x^2].
Are the classes now linearly separable?
Taking the points to 3-D feature vectors using the function f(x) = [1, sqrt(2)x, x^2], we are going to have three points of $(1, 0, 0)$, $(1, -\sqrt{2}, 1)$, $(1, \sqrt{2}, 1)$.
Based on the 3D view below, the classes are linearly separable by a hyperplane.

If so, find a separating hyperplane.

A separating hyperplane is given by the weight vector (0,0,1) in the new space.
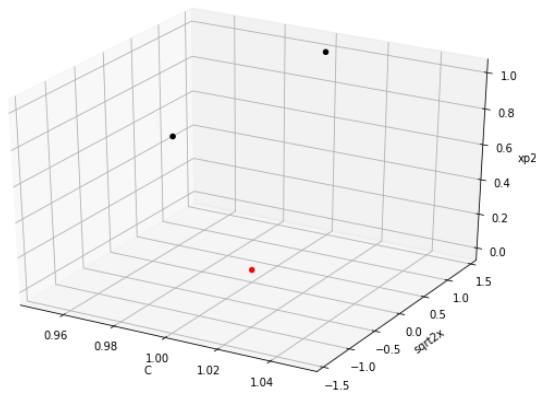
Task 7: Learning SVMs on the Titanic dataset ((https://www.kaggle.com/c/titanic). Please report your fivefold cross validation classification accuracies on Titanic training set, with respect to the linear, quadratic, and RBF kernels. Which kernel is the best in your case?

I ran the Grid Search Cross Validation on the dataset and the 'RBF' kernel was found as the best kernel for this case.

The fivefold cross validation SVM accuracy is as below:

`[0.73913043, 0.83060109, 0.73770492, 0.79234973, 0.74863388]`

The mean of the fivefold cross validation is 0.76

You can find my code uploaded in Github using the link below :

https://github.com/nimaarvin83/CAP5610/blob/main/HW3_Titanic/titanic/titanic.ipynb