Assignment -2

Question 1
How is Soft Margin Classifier different from Maximum Margin Classifier?

Maximum Margin Classifier, as we know is the best line- which maintains the largest possible equal distance from the nearest points of both the classes. So for the separator to be optimal, the margin or the distance of the nearest point to the separator should be maximum.

Although the maximal margin line (hyperplane), separates the two classes perfectly, is very sensitive to the training data. This means that the Maximal Margin Classifier will perform perfectly on the training data set. But on the unseen data, it may perform poorly. Also, there are cases where the classes cannot be perfectly separated. This is where soft margin classifier.

The soft margin classifier essentially allows certain points to be deliberately misclassified. By doing this, it is able to classify most of the points correctly in the unseen data and is also more robust.

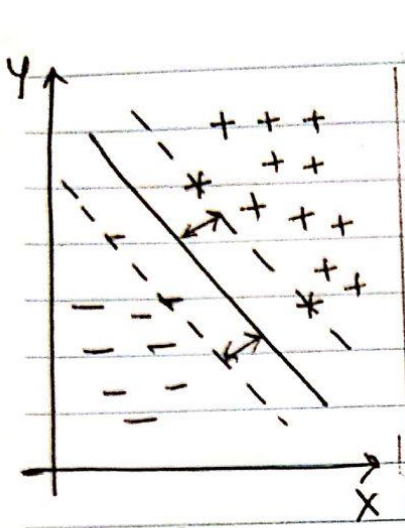
So the differences are - The allowance of softness in margins allows for errors to be made while fitting the model to the training/discovery data set. Conversely, hard margins will result in fitting of a model that allows zero errors. Sometimes it can be helpful to allow for errors in the training set, because it may produce a more generalizable model when applied to new datasets. Forcing rigid margins can result in a model that performs perfectly in the training set, but is possibly over-fit / less generalizable when applied to a new dataset.

Question 2
What does the slack variable Epsilon (ε) represent?

Like the Maximal Margin Classifier, the Support Vector Classifier also maximises the margin; but the margin, here, will allow some points to be misclassified.
A slack variable Epsilon (ε) is used to control the misclassifications. It tells you where an observation is located relative to the margin and hyperplane.

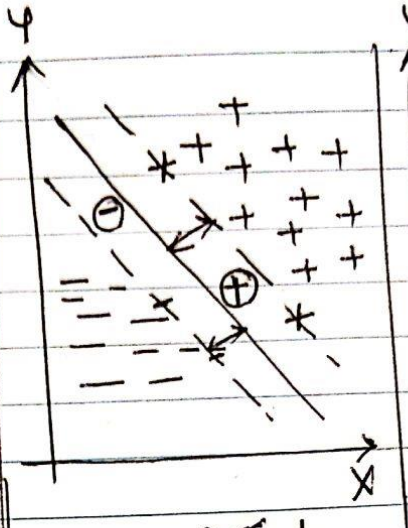
There are three different conditions applied if any new data point comes into play. They are:

Assignment -2



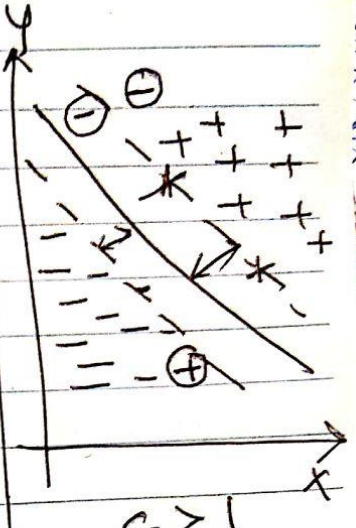
$\epsilon = 0$

Suppose if we define a support vector classifier in such way that it doesn't allow for any mis-classification i.e epsilon(ϵ)=0 then each observation is on the correct side of the margin.



$0 < \epsilon < 1$

If a data point is correctly classified but falls inside the margin (or violates the margin), then the value of its slack ϵ is b/w 0 & 1.



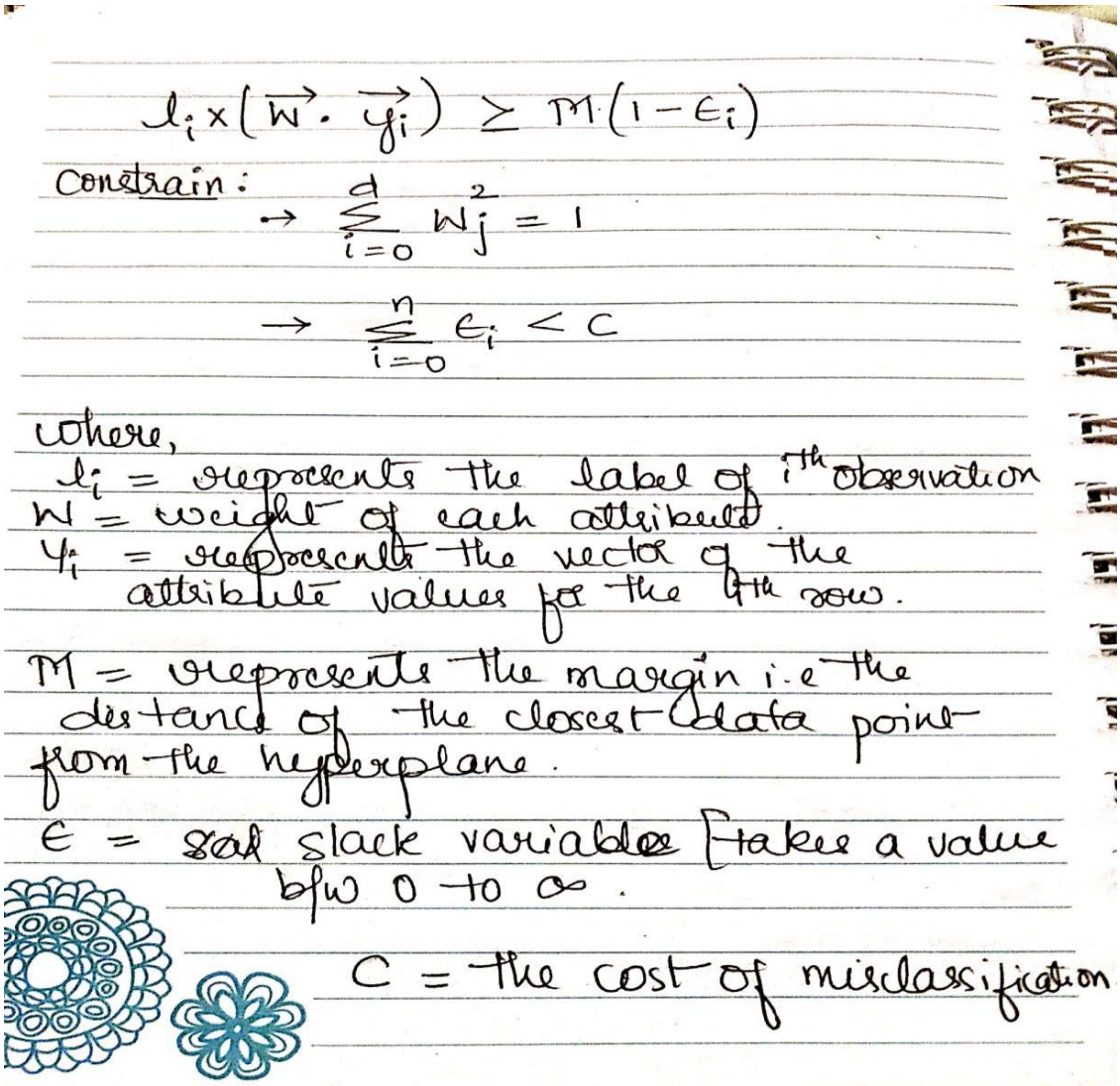
$\epsilon > 1$

If the data point is incorrectly classified (i.e it violates the hyperplane) the value of epsilon (ϵ) > 1..

Assignment -2

Question 3

How do you measure the cost function in SVM? What does the value of C signify?



The summation of all the epsilons of each data point is denoted by cost or 'C', i.e. $\sum \epsilon i \leq C$.

When C is large, the slack variables can be large. In this case, the model is flexible, more generalizable, and less likely to overfit. In other words, it has a high bias. That is because we are allowing a larger number of data points to be misclassified or to violate the margin. With this we get a hyperplane where the margin is wide and misclassifications are allowed.

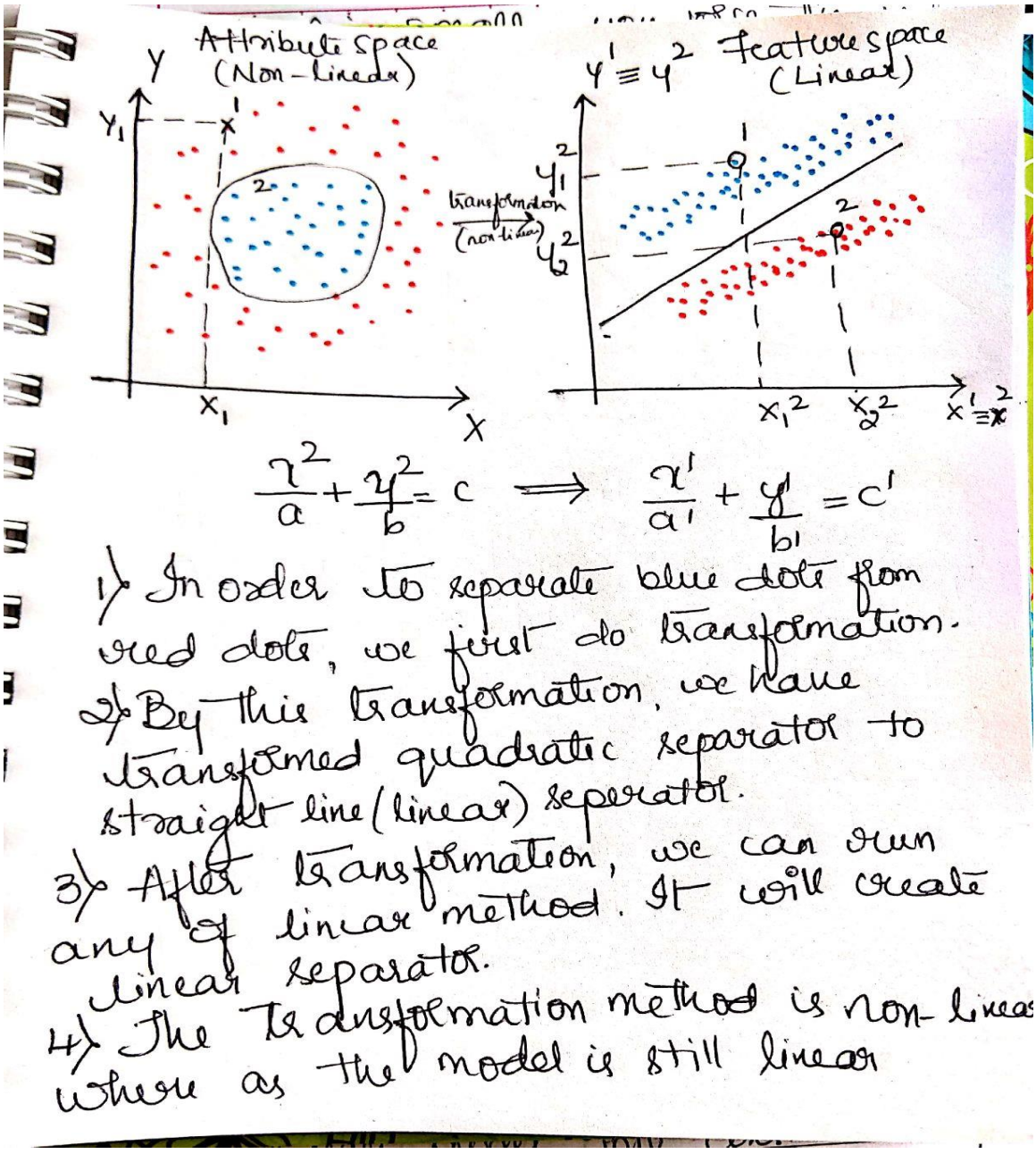
When C is small, we force the individual slack variables to be small. In this case, the model is less flexible, less generalizable, and more likely to overfit. In other words, it has a high variance. That is because we do not allow many data points to fall on the wrong side of the margin or the hyperplane. So the margin is narrow and there are few misclassifications.

Assignment -2

Question 4

Given the above dataset where red and blue points represent the two classes, how will you use SVM to classify the data?

It is evident from the given figure that- is not possible to imagine a linear hyperplane that separates the red and blue points reasonably well. Thus, we tweak the linear SVM model and enable it to incorporate nonlinearity in some way. Kernels serve this purpose — they enable the linear SVM model to separate nonlinearly separable data points.



As shown in the picture above, we do the transformation of every point from attribute space (non-linear) to feature space (linear). Once the transformation of data points is done, we can use our SVM model – which will give us linear separator between red and blue points.

Assignment -2

Question 5
What do you mean by feature transformation?

We can transform nonlinear boundaries to linear boundaries by applying certain functions to the original attributes. The original space (X, Y) is called the attribute space, and the transformed space (X', Y') is called the feature space. The process of transforming the original attributes into a new feature space is called 'feature transformation'.

However as the number of attributes increases, there is an exponential increase in the number of dimensions in the transformed feature space. This creates a huge computational cost. This is where kernels comes into picture to solve the problem of complex feature transformation.