CptS 570: Assignment 2

Abrar Akhyer Abir

Question 1:
First part: CLOSE classifier specifies a point as positive if it is close to C_+ and classifies as negative if it is close to C_-. So, we need to calculate the distance of x from C_+ and C_- and whichever distance is close, we will assign the point to that class.

$$distance(x, C_+) = ||C_+x||^2 = x^2 + 2.x.C_+ + C_+^2$$
$$distance(x, C_-) = ||C_-x||^2 = x^2 + 2.x.C_- + C_-^2$$

From the above equation, we can write:

$$f(x) = \begin{cases} 1, & (2x(C_- - C_+) + (C_+^2 - C_-^2)) > 0 \\ -1, & otherwise \end{cases}$$

Now, if we want to compare the above equation in form of $(wx + b)$:

$$w = 2(C_- - C_+)$$
$$b = C_+^2 - C_-^2$$

Second part:
We have:

$$w = \sum_{i=1}^{n^+ + n^-} \alpha_i y_i x_i$$

From the first part of the question, we got:

$$w = 2(C_- - C_+)$$

Now,

$$2(C_- - C_+) = \sum_{i=1}^{n^+ + n^-} \alpha_i y_i x_i$$
$$\Rightarrow 2C_- - 2C_+ = \sum_{i=1}^{n^+} \alpha_i y_i x_i + \sum_{i=1}^{n^-} \alpha_i y_i x_i$$
$$\Rightarrow 2C_- - 2C_+ = \sum_{i=1}^{n^+} \alpha_i y_i x_i + \sum_{i=1}^{n^-} \alpha_i y_i x_i$$
$$\Rightarrow 2C_- - 2C_+ = \sum_{i=1}^{n^+} \alpha_i x_i - \sum_{i=1}^{n^-} \alpha_i x_i$$

From the question, we know that:

$$C_+ = \frac{1}{n_+} \sum_{i:y_i=+1} x_i \text{ and } C_- = \frac{1}{n_-} \sum_{i:y_i=-1} x_i$$

If we compare both equations from above, we will get for positive class:

$$\alpha_i = \frac{2}{n_+}$$

For negative class:

$$\alpha_i = \frac{2}{n_-}$$

Question 2:
First part: We have the redial basis function (RBF) kernel as follows:

$$K(x_i, x_j) = exp(-\frac{1}{2}||x_i - x_j||)^2$$

We will write the equation following way using maclarin series:

$$\begin{aligned} K(x_i, x_j) &= exp(-\frac{1}{2}||x_i - x_j||)^2 \\ &= exp-\frac{1}{2}||x_i||^2.exp-\frac{1}{2}||x_j||^2.exp^{x_i.x_j} \\ &= exp-\frac{1}{2}||x_i||^2.exp-\frac{1}{2}||x_j||^2.\sum_{n=0}^{\infty}\frac{(x_i.x_j)^n}{n!} \end{aligned}$$

Here, we can see that the third term has infinite degree which will project the vector into higher number of dimension. As a result, we can say, RBF kernel has infinite dimensions.
Second part:

$$\begin{aligned} ||\Phi(x_i) - \Phi(x_j)||^2 &= \Phi(x_i)^2 + \Phi(x_j)^2 - 2\Phi(x_i).\Phi(x_j) \\ &= K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j) \\ &= 1 + 1 - 2.exp(-\frac{1}{2}||x_i - x_j||)^2 \\ &= 2 - 2.exp(-\frac{1}{2}||x_i - x_j||)^2 \end{aligned}$$

From the above equation, we can say that:

$$||\Phi(x_i) - \Phi(x_j)||^2 \leq 2$$

Question 3:
We have the RBF kernal as follow:

$$K(x_i, x_j) = exp(-\frac{1}{2}||x_i - x_j||)^2$$

According to the question, we have a testing example x_{far} that is far away from one of the training instance x_i. We can write the distance as:

$$||x_i - x_{far}|| = \infty$$

which leads to :

$$K(x_i, x_j) = exp(-\frac{1}{2}||x_i - x_j||)^2 = 0$$

Now, if we substitute the the above in the decision boundary equation:

$$\begin{aligned} w.\Phi(x) + b = f(x; \alpha, b) &= \sum_{i \in SV} y_i.\alpha_i K(x_i, x) + b \\ &= b \end{aligned}$$

We can say now:

$$f(x; \alpha, b) \approx b$$

Question 4:
According to Mercer's theorem a function K is a kernal function if and only if its corresponding kernal matrix is symmetric and positive semi definite. We are given a Kernal function $K(x_i, x_j) = -<x_i, x_j>$.This kernal function does not follow the positive semi-definite properties. So, this is not a valid kernal.

Question 5:
We know, the soft margin for SVM is:

$$-y^i(w^i.x^i + b) \geq 1 - \xi_i$$

We have two cost C_+ and C_- the cost of misclassifying a positive and negative example respectively. For the positive example $y_i = 1$:

$$-y^i(w^i.x^i + b) \geq C_+ - \xi_i$$

For the negative example $y_i = -1$:

$$-y^i(w^i.x^i + b) \geq C_- - \xi_i$$

Question 6:
a. We can consider each cluster as our training data. Suppose, we have $n1$ as k_+

positive clusters and $n2$ as k_- clusters. So, we have $n1 + n2$ cluster center points as our training data. We will now train our SVM model using this training points. Each of these cluster will then be assigned an α value which will specifies the importance of the cluster.

b. Here, we will eliminate the clusters with no data point or $\alpha = 0$ after calculating the support vectors. The clusters with $\alpha = 0$ have uniform labeling, hence further consideration of them is not necessary. The remaining mixed-label clusters will be re-clustered for a subsequent review since they contain components more likely to develop into support vectors. For instance, we will select a specific cluster C_k and locate the support vector that is closest to it in cluster C_1. The vector from C_1 to C_k will then be moved until all of C has a uniform label ($\alpha = 0$).

c. When we have no other cluster to recompute, we can stop.

Question 7:

a. Tom can choose the support vectors that has the highest alpha value since higher the value of alpha indicates that support vector is close to our decision boundary and it influences output. The algorithm would be the following:

Algorithm 1 An algorithm to select B support vector

1: Sort the support based on their alpha value.
2: Select upto B support vectors from the sorted list.

b. If we want to use at most B support vectors during the training process, at first we initialized a B sized list where we will store our α values. While training whenever we are updating the α value, we will check whether the α value is less than the minimum α value in our stored list. If it is less than the minimum α value from the list, we assign that α value as ZERO. Otherwise, we will store it on our list. By following this procedure only highest α values will be stored and rest of α will be assigned to zero. We can follow this algorithm:

Algorithm 2 An algorithm to select B support vector in training

1: Initialized a B sized array to store alpha values named $bvalues$
2: While updating alpha value in training, we have: $\alpha_i = \alpha_i + y_i$
3: if $\alpha_i < min(bvalues)$ then
4: $\quad \alpha_i = 0$
5: else
6: $\quad bvalues.add(\alpha_i)$

Programming and Empirical Analysis:
Question 1:
a. From the Figure 1, we can see as C-value increases, the number of accuracy get higher in our training data. However, it is not true for validation and testing data. The validation and testing accuracy increased upto $c = 0.1$ then it start decreasing. We get the highest testing and validation accuracy at $C = 0.1$

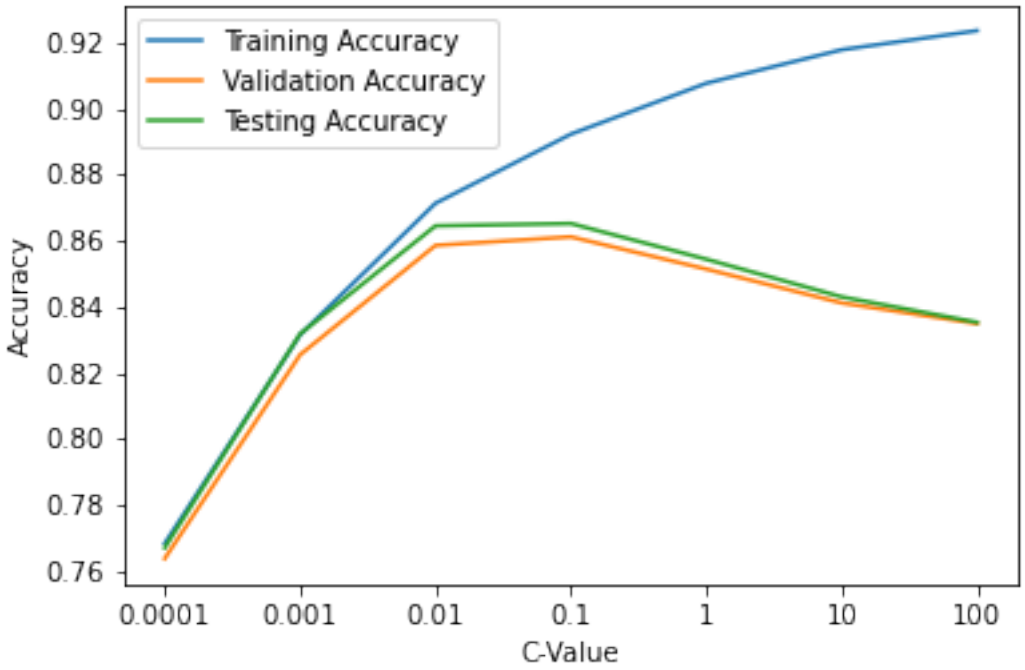

Figure 1. Accuruacy of SVM with different C-values

In case of number of support vectors from the Figure 2 , we see as the C value increases the number of support vectors decreases. While running these experiments, I have also noticed that higher the C value, higher the running time. As a result, I could not able to generate the result for $C = 1000, 10000$ since it was taking longer time and my laptop got slowed down.

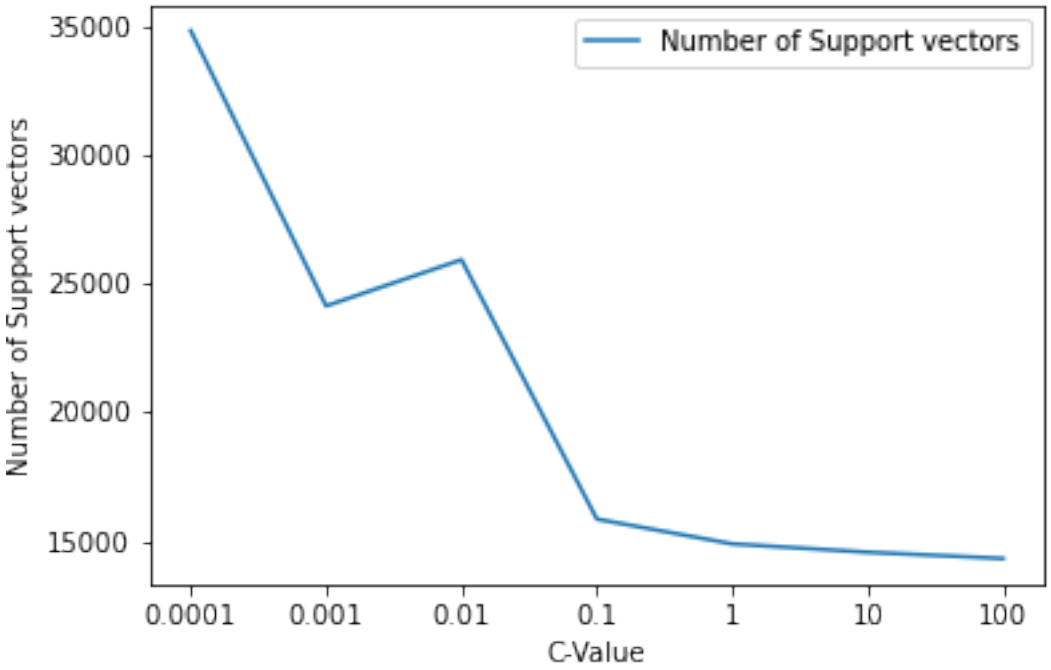

Figure 2. Number of support vectors with different C-values

b. From the Figure 1, we see that the best C-value is 0.1. So, I have run the SVM model with all training and validation data and got testing accuracy of 0.8654 and the confusion matrix is shown in Figure 3 .

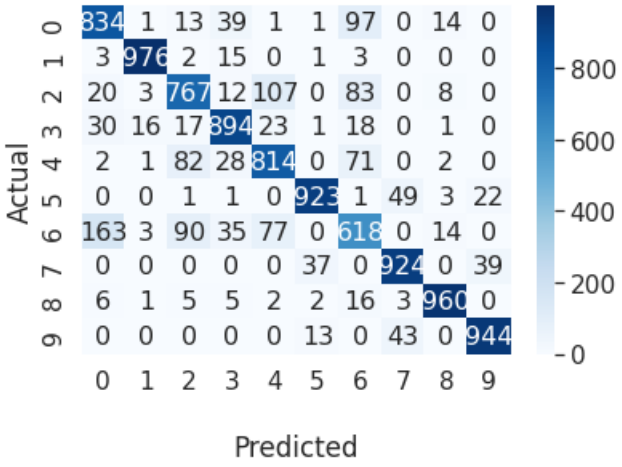

Figure 3. Confusion Matrix for C=0.1

c. From the Figure 4, we can see that as the polynomial degree increases, the training, validation and testing accuracy decreased. We get the best accuracies for linear kernal. While running the experiments, I also noticed that as degree increased the running time to build the model also increased.

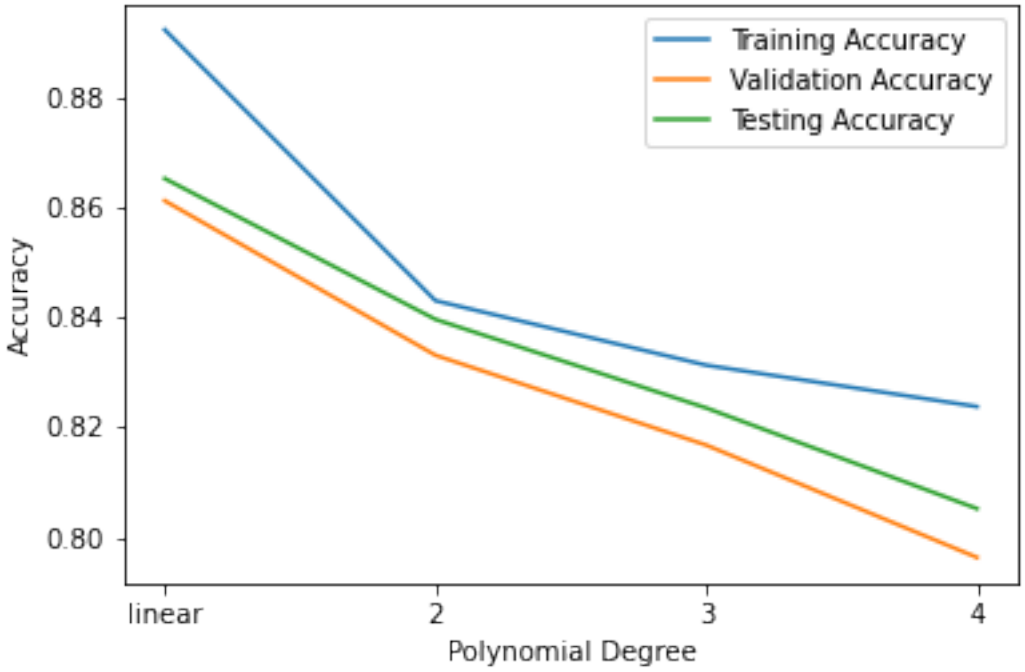

Figure 4. Accuracy for different polynomial degree

In case of number of support vectors (Figure 5), we get the lowest number of support vector in linear kernal. For polynomial kernal, as the degree increased the number of support decreased than previous degree.

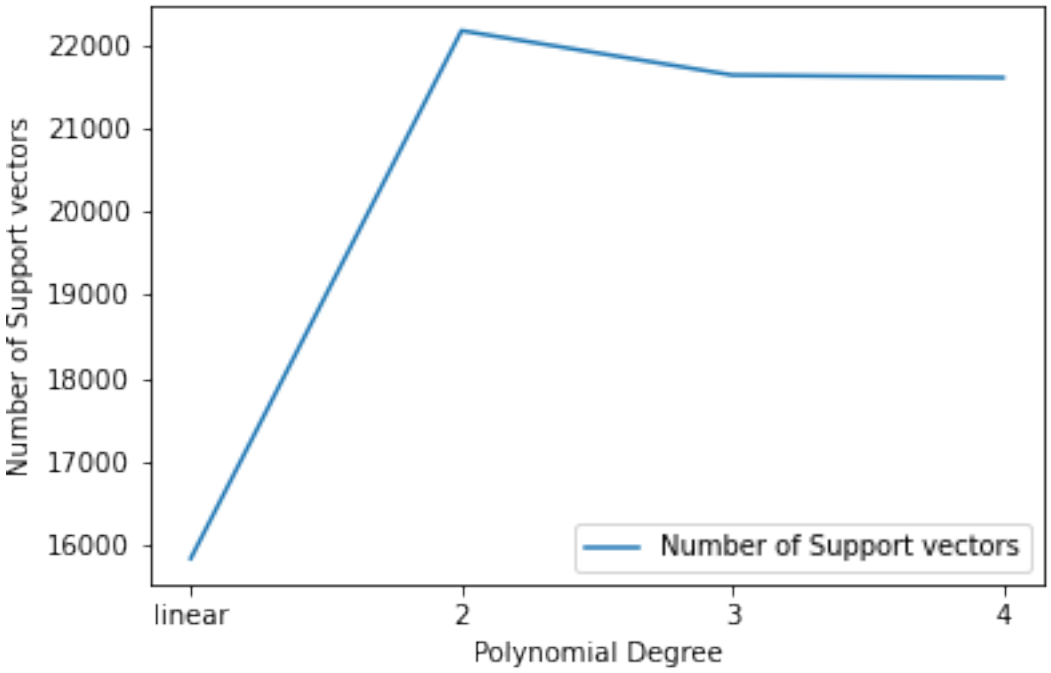

Figure 5. Number of support vectors with different polynomial degree

Question 2:

a. From the Figure 6, as the number number of iteration increases, the mistakes in training data after each iteration also decreased. After running the experiment, I have found that for polynomial degree 2, we get the highest accuracy. Below the training, validation, and testing accuracy at the end of 5 iterations:

Training Accuracy: 0.8866

validation Accuracy: 0.8563

Testing Accuracy: 0.8632

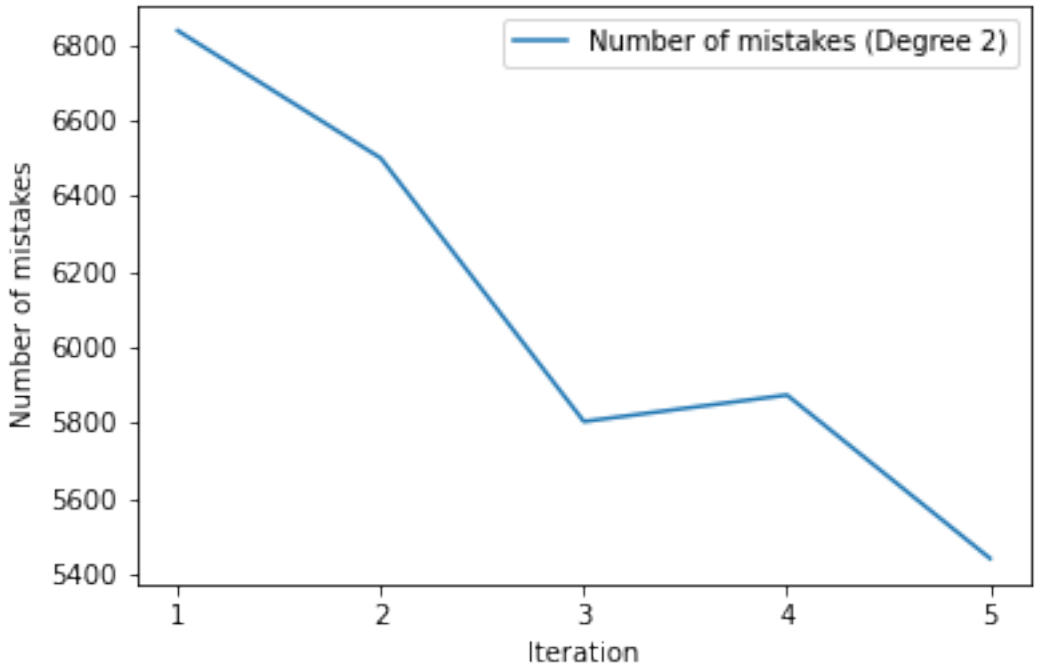

Figure 6. Number of mistake per iteration for polynomial degree=2

Question 3:
a. The tree has been generated using the information heuristic and following the formula to compute candidate feature using the formula from the question.
b. The accuracy on validation examples and testing examples without pruning the tree:
Validation Accuracy: 0.89
Testing Accuracy: 0.92

c. I have built the pruning by limiting the depth of the tree. I run the experiments for depth=10 and the result is shown on Figure 7. From Figure 7, we can see that for depth 5, we get both validation and testing accuracy highest. So, we will take the depth as 5.

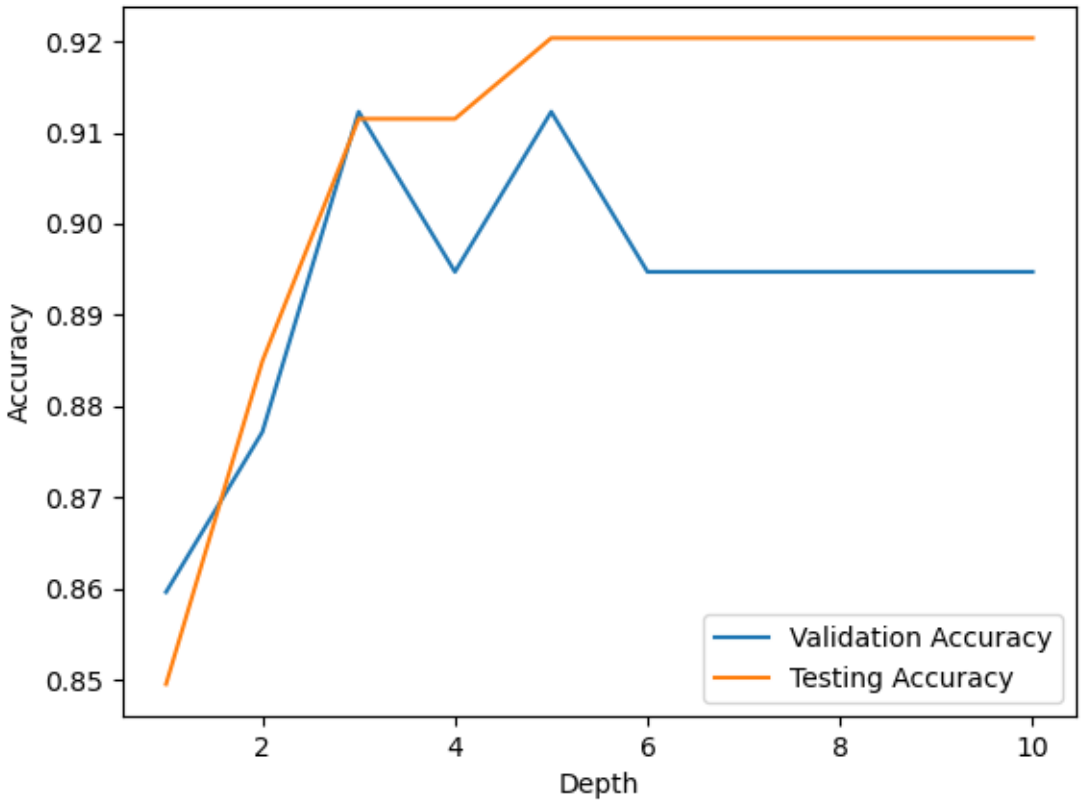

Figure 7. Number of mistake per iteration for polynomial degree=2

d. From the figure 7, we saw that the at depth 5, we get the highest validation and testing accuracy.
Validation Accuracy: 0.9122807017543859
Testing Accuracy: 0.9203539823008849
We also see that at depth=1, the decision tree has lowest accuracy. As the depth increased, we get better accuracy. However, at certain depth the validation accuracy dropped. At depth=5, we get both testing and validation accuracy high. If we kept increasing the depth, the decision tree will completely fit with our training data and it may over generalized and can not perform well on testing data.