Homework 2

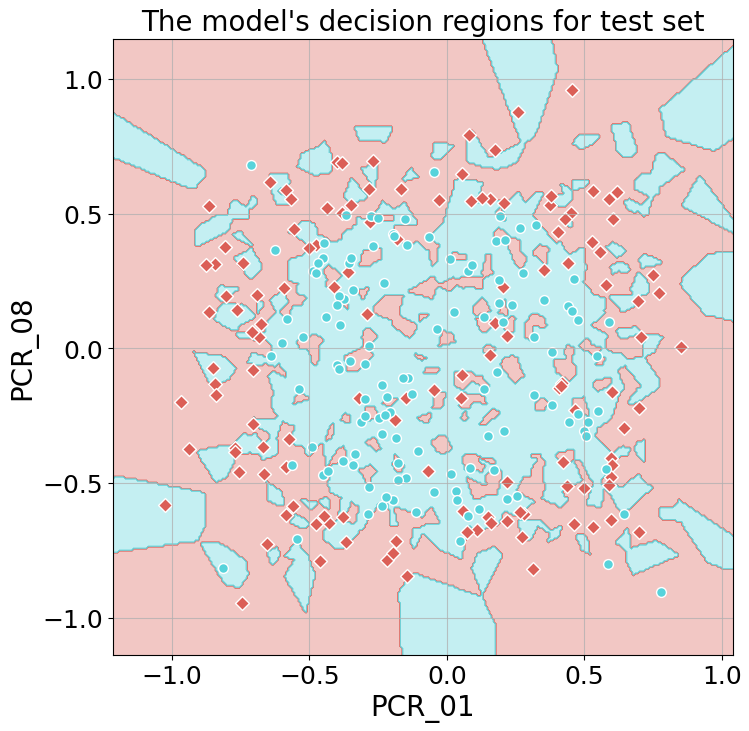
**Introduction to Machine Learning (02360766)**

Submitted by: Eyal Amdur, Gal Porat

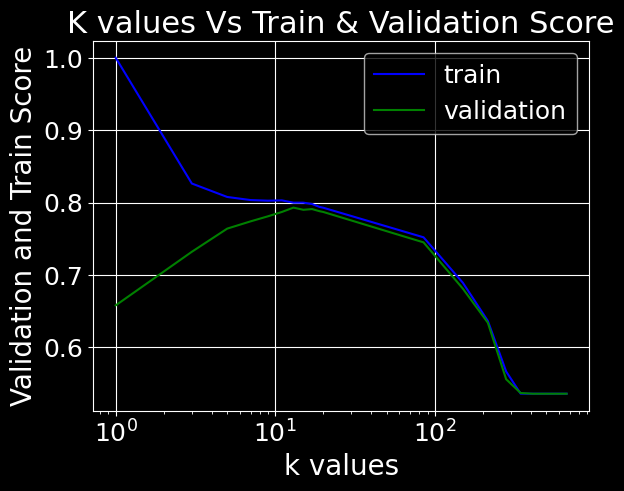
ID: 318849270, 315116095

**Part 1: Basic model selection with k-Nearest Neighbors**

**(Q1)** **Training set score: 1.0**  **Test set score: 0.7**



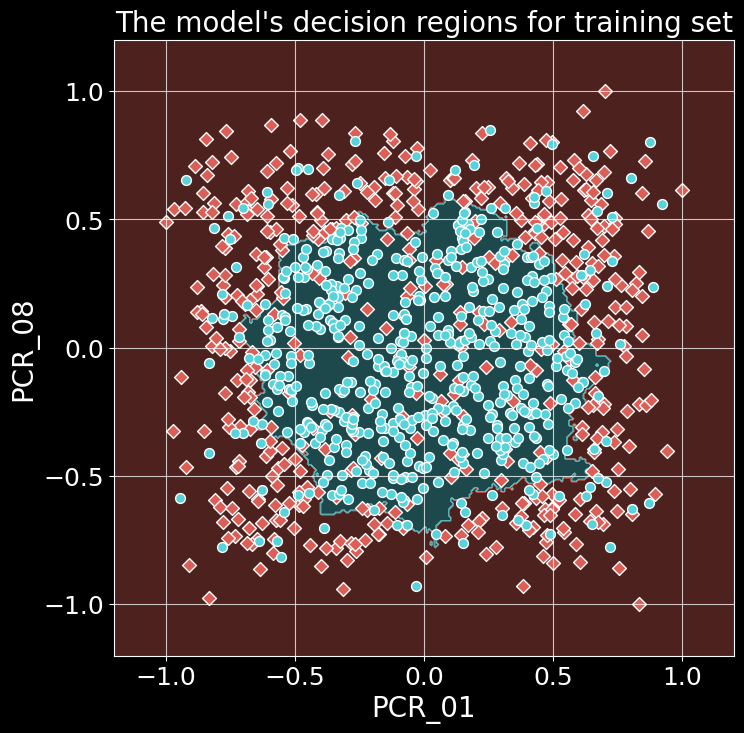
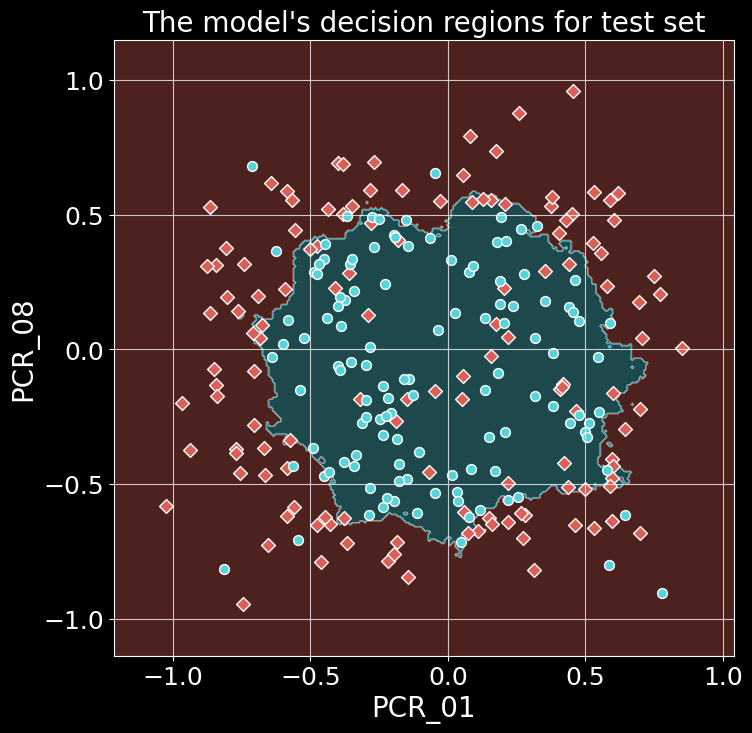
**(Q2)**

From the attached plot we can see that the K that provides the maximal validation is K=13 With:

Setting k = 1 leads to overfitting, as shown in the plot where the training score is perfect at 1. This happens because, with k = 1, each sample is considered as its own nearest neighbour, which gives usperfect fit to the training data without allowing the model to generalize effectively.

On the other hand, using k = 670 results in underfitting due to the considerably low training score. A model with k = 670 considers a large amount of samples as its neighbours, which diminishes the affect of the closest neighbours, ultimately resulting in a failure to capture the patterns in the data.Top of Form

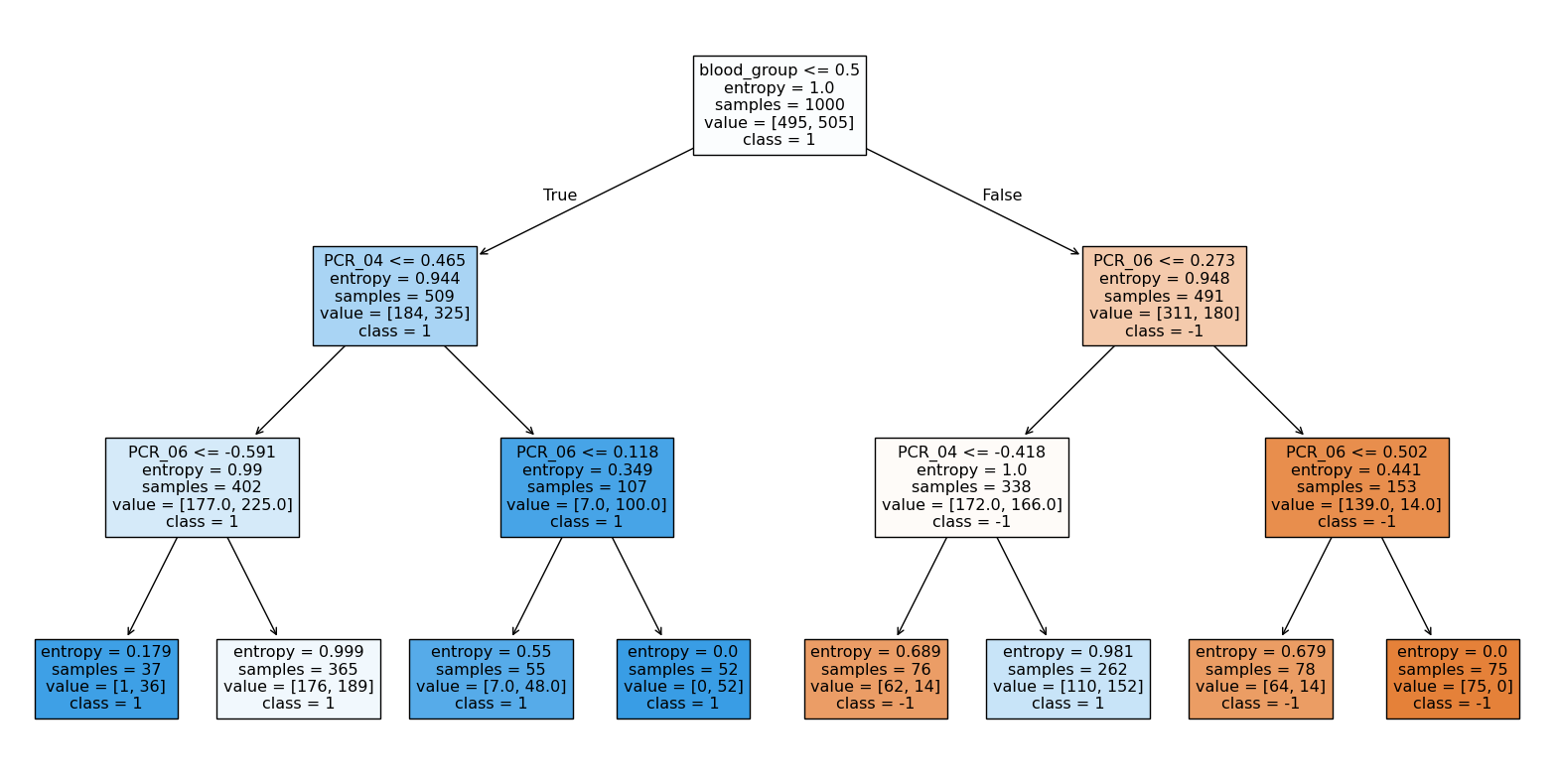
**(Q3)** For K=13 (Best k from Q2) the results are:

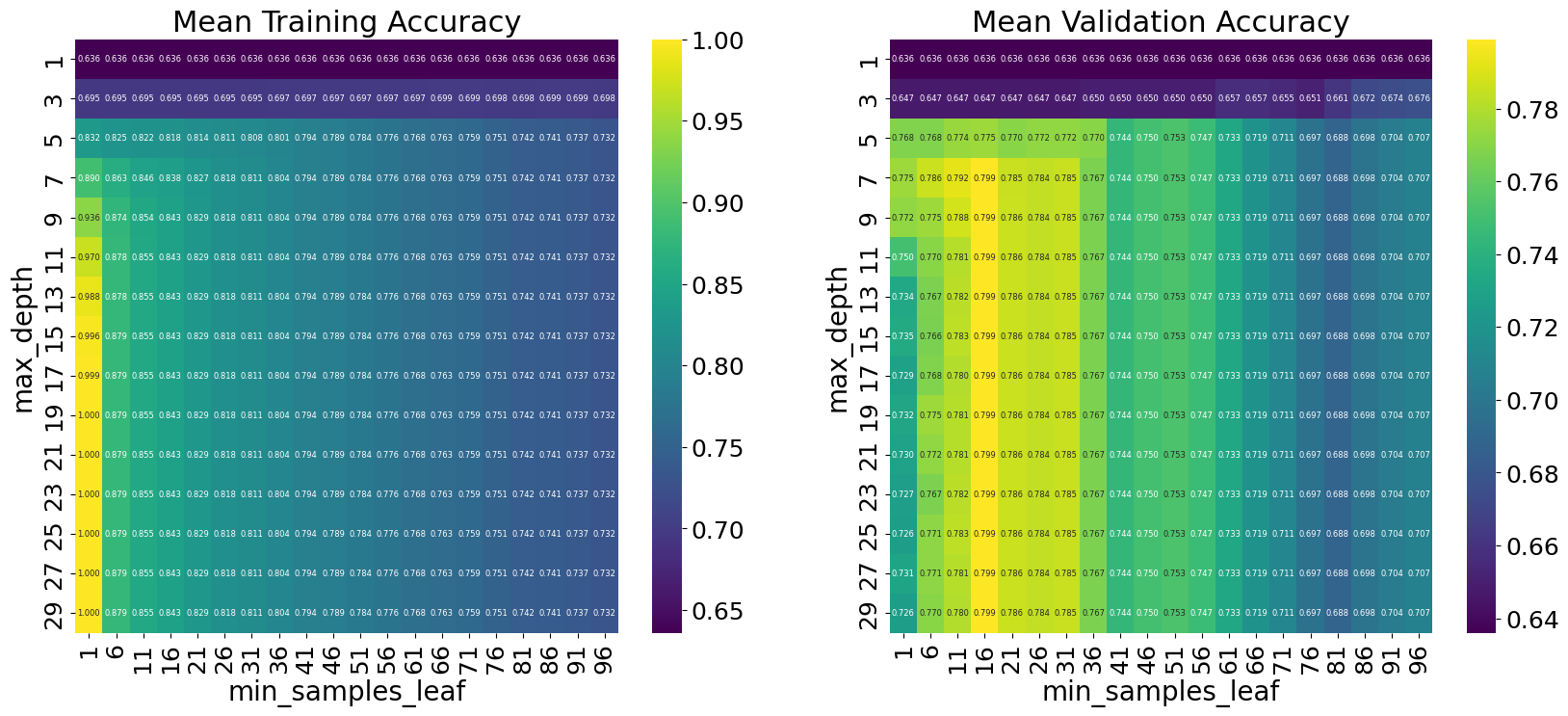
**Training set score: 0.8**  **Test set score: 0.776**

**(Q4)** In Question 1, the test score was 0.7 when k was 1, whereas it increased to 0**.**776 when k is 13. As mentioned before, using k = 1 results in overfitting, which consequently leads to a lower score on the test set.

**Part 2: Decision trees**

**(Q5)**



**(Q6)**

**c)** The optimal combination according to the heatmap is:  
min\_samples\_leaf: 16  
max\_depth: 7

**d)** A hyperparameter combination that causes underfitting is:  
min\_samples\_leaf: 96  
max\_depth: 1

**e)** A hyperparameter combination that causes overfitting is:  
min\_samples\_leaf: 1  
max\_depth: 29

**f)** The hyperparameter in question d leads to underfitting because the tree is restricted to a single split. As a result, it cannot effectively separate non-separable data, such as the data in our case.

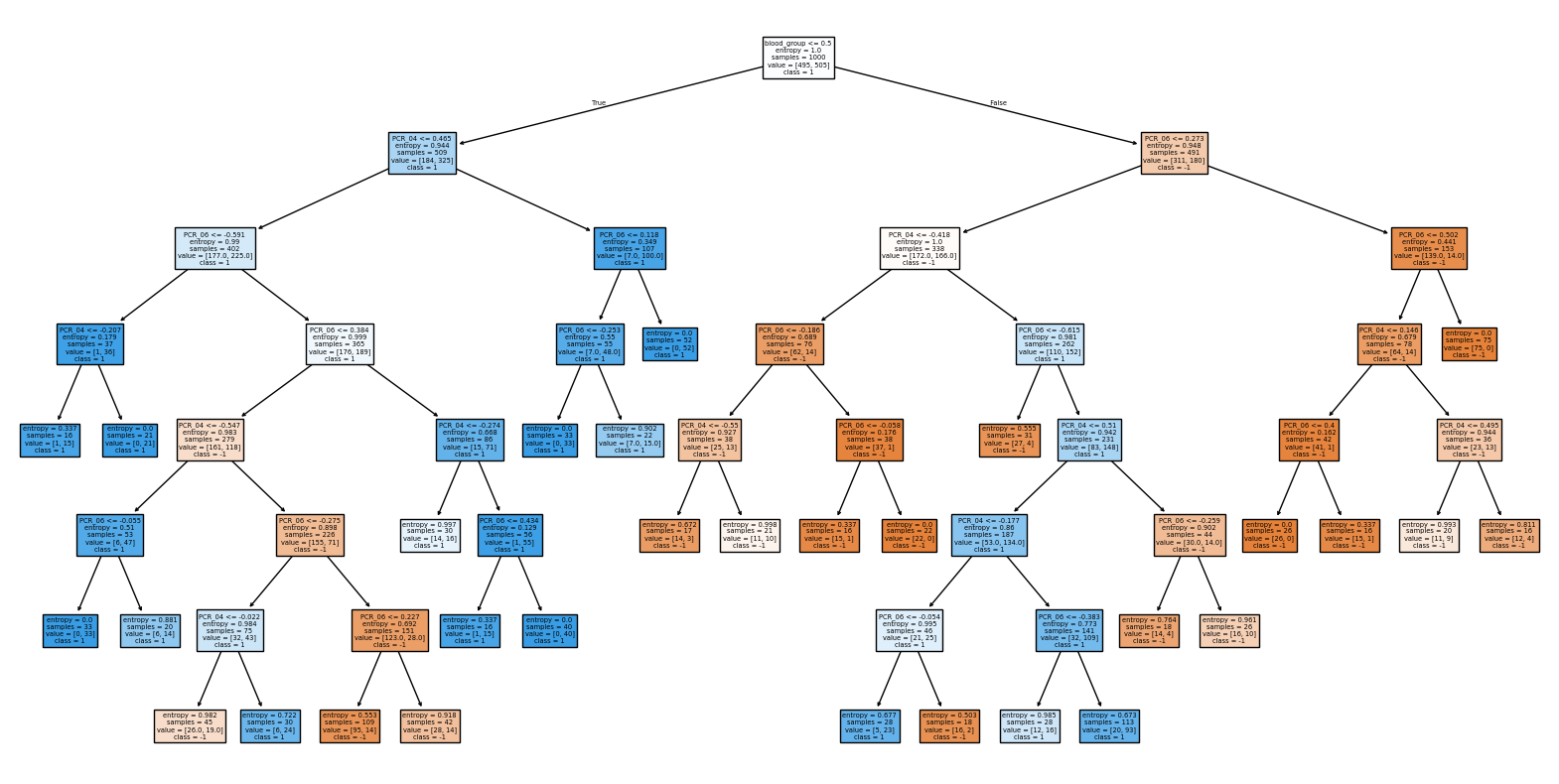
The hyperparameter in question e causes overfitting because the model is allowed to grow a tree with leaves containing just one sample. While this results in highly accurate predictions on the training set, the model becomes too tailored to the training data. Consequently, it performs poorly on the validation set, as it fails to generalize well to unseen data.

**(Q7)** Since our grid consisted of 15 potential values for max\_depth and 20 values for min\_samples\_leaf, we evaluated a total of 15×20=300 hyperparameter combinations.

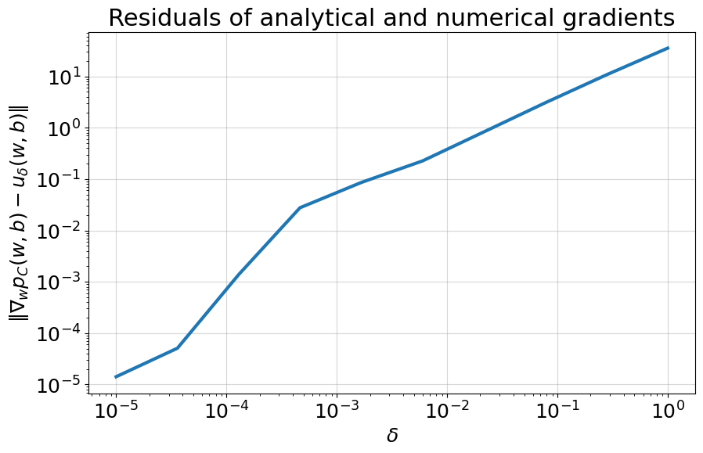
If we had introduced a third hyperparameter, the total number of combinations would increase by multiplying the current total by the number of options for the new parameter. Generally, if each hyperparameter has n possible values and there are k hyperparameters, the total number of combinations is n^k.

While having more hyperparameters to tune can improve model performance by exploring a wider range of configurations, it significantly increases computational cost. Therefore, it’s essential to balance the potential performance gains with the computational expense.

**(Q8)**



**Part 3: Linear SVM and the Polynomial kernel**

**(Q9)**

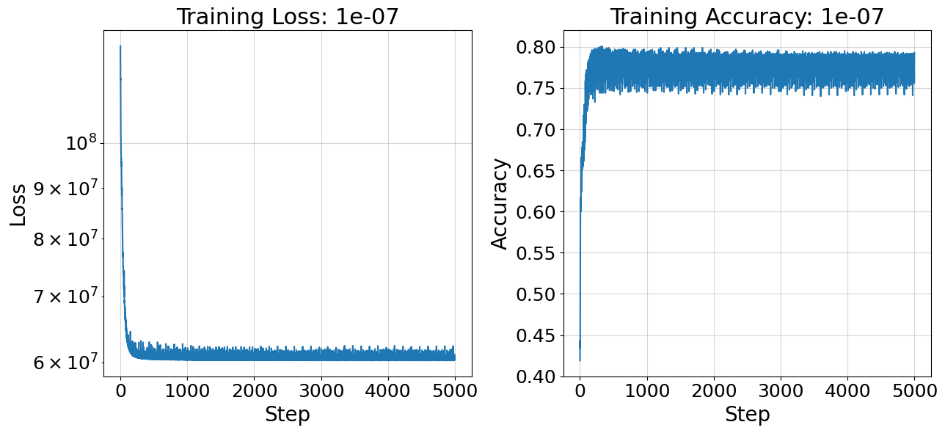
As can be seen in the graph, the smaller the delta, the smaller the difference between the analytical and numerical derivatives. This is because a smaller delta represents a slope at a point that is closer to the computed point, thus providing a value that is closer to the actual slope at that point.

**(Q10)**   
The relationship between the error and accuracy graphs is not as expected. As the number of steps increases, we would anticipate a decrease in error and an increase in accuracy. While the error does decrease, the accuracy varies without a clear trend. This difference arises due to the high value of the regularization parameter C, which emphasizes the loss function and leads to the HardSVM problem.

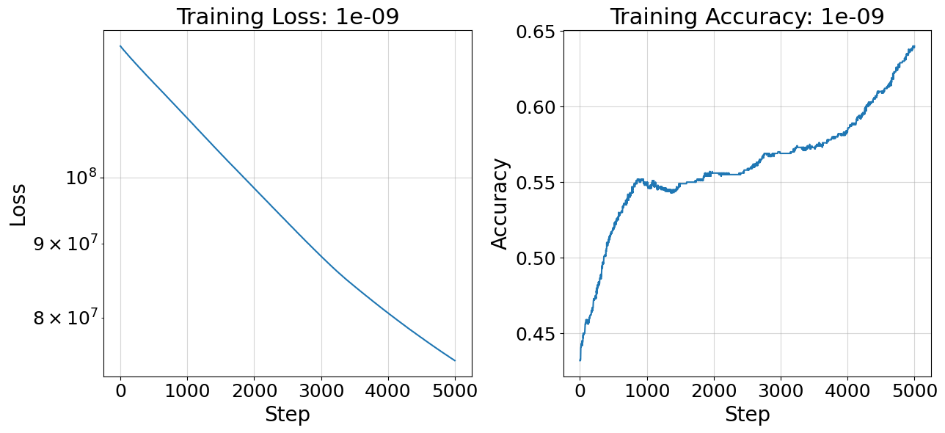
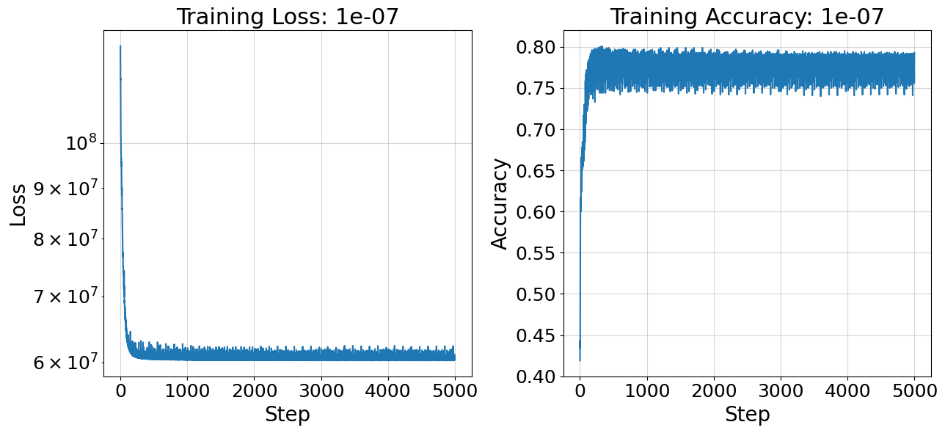
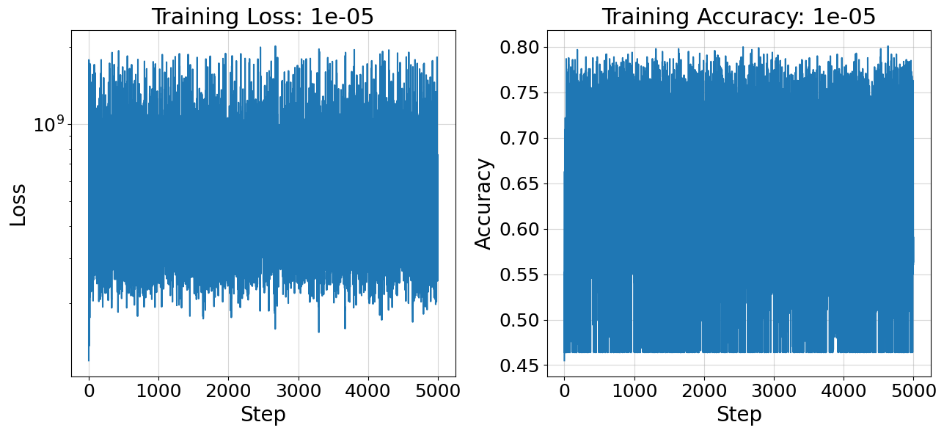
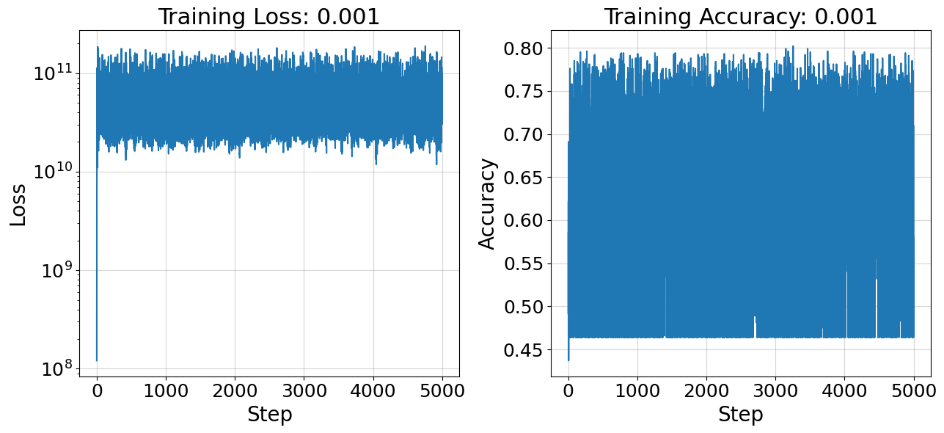
**(Q11)**

We would choose the learning rate of 1e-07.

As shown in the graph below, for this rate, the error decreases rapidly with a steep slope, and accordingly, the accuracy measure rises to a high score, ranging between 0.74 and 0.80.   
In comparison to other measures, we can see that the upper bounds of the accuracy measures are approximately equal to the rate we chose, but the lower bounds are much lower. Additionally, these measures are unstable, making it difficult to clearly understand their scores for each step.  
Our chosen learning rate of 1e-07 graphs:

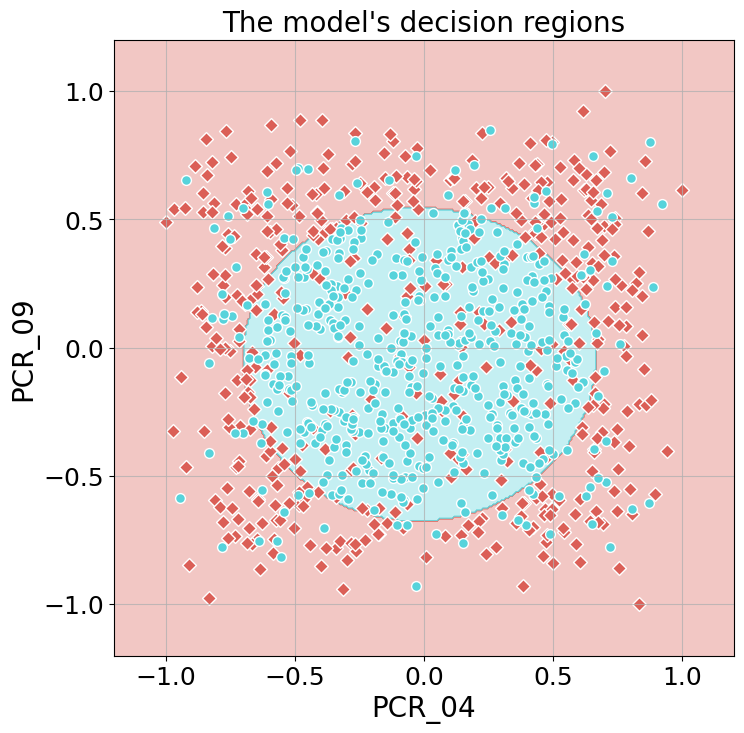
****

**All the learning rates graphs:**

****

**(Q12)** Our model accuracy is:

And the decision model is:



**Part 4: The RBF kernel**

**(Q13)**