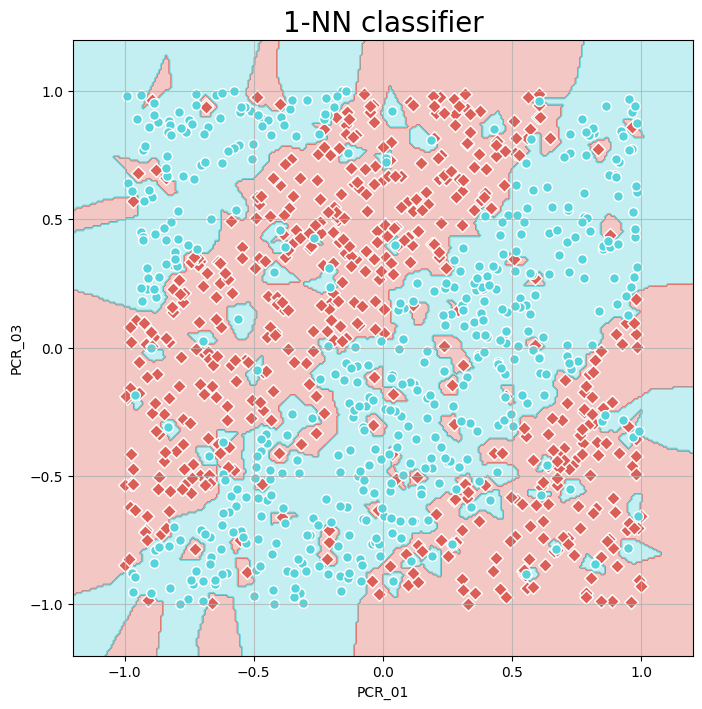
**Wet 2**

**Yagel Maimon**

**Saar Gablinger**

**Q1.** 

Q2. A graph with lines and numbers

Description automatically generated

**The best k value is 9**

**Mean train score: 1.0**

**mean Validation score: 0.869**

**by the figure The k values that cause overfitting is about from 1 to 5 because this is the lowest k values that make the training accuracy be the much highest and we can see that in those numbers the difference between the validation accuracy and the training accuracy is the biggest. and the k values that cause underfitting is about 130 and above that’s happends because we look at too many neighbors and from 130 k we can see that the accuracy is lower then 0.7 and get lower exponential when we go forther.**

Q3.

A map of a map with red and blue dots

Description automatically generated

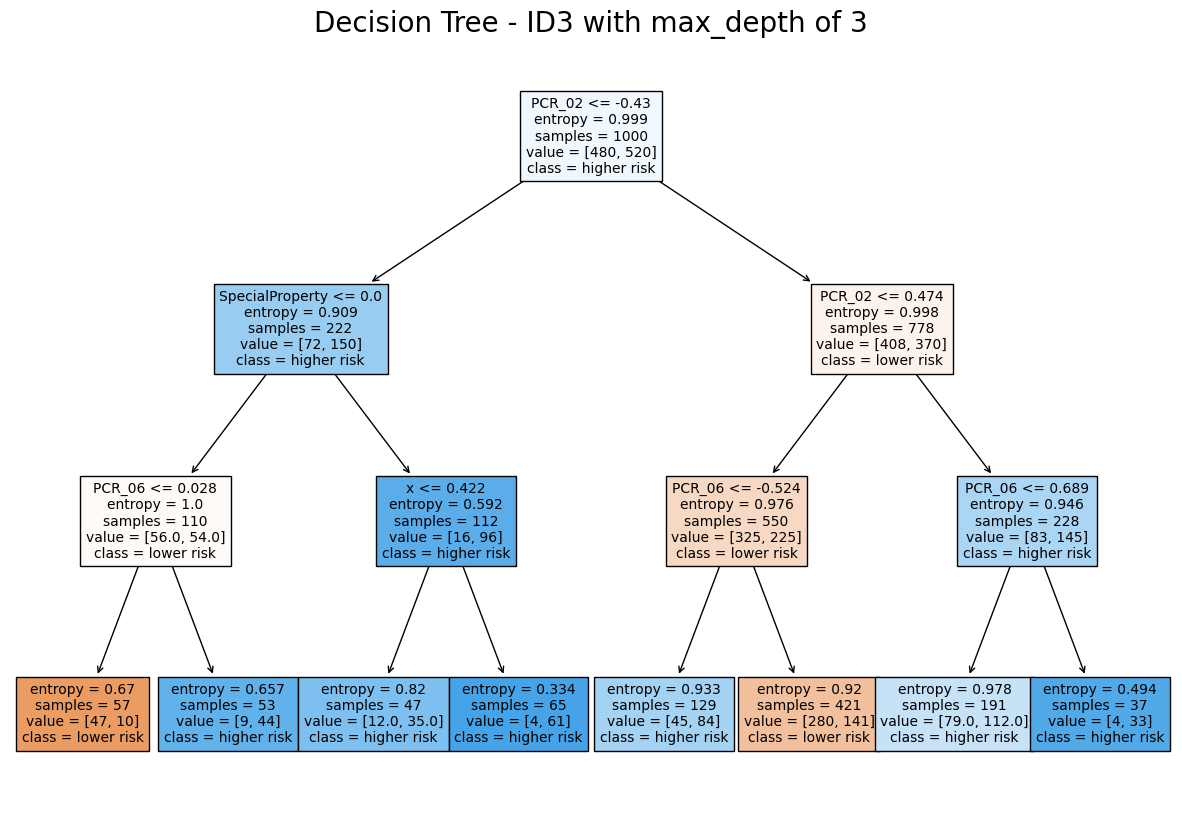
And the tests accuracy is 0.8

Q4.

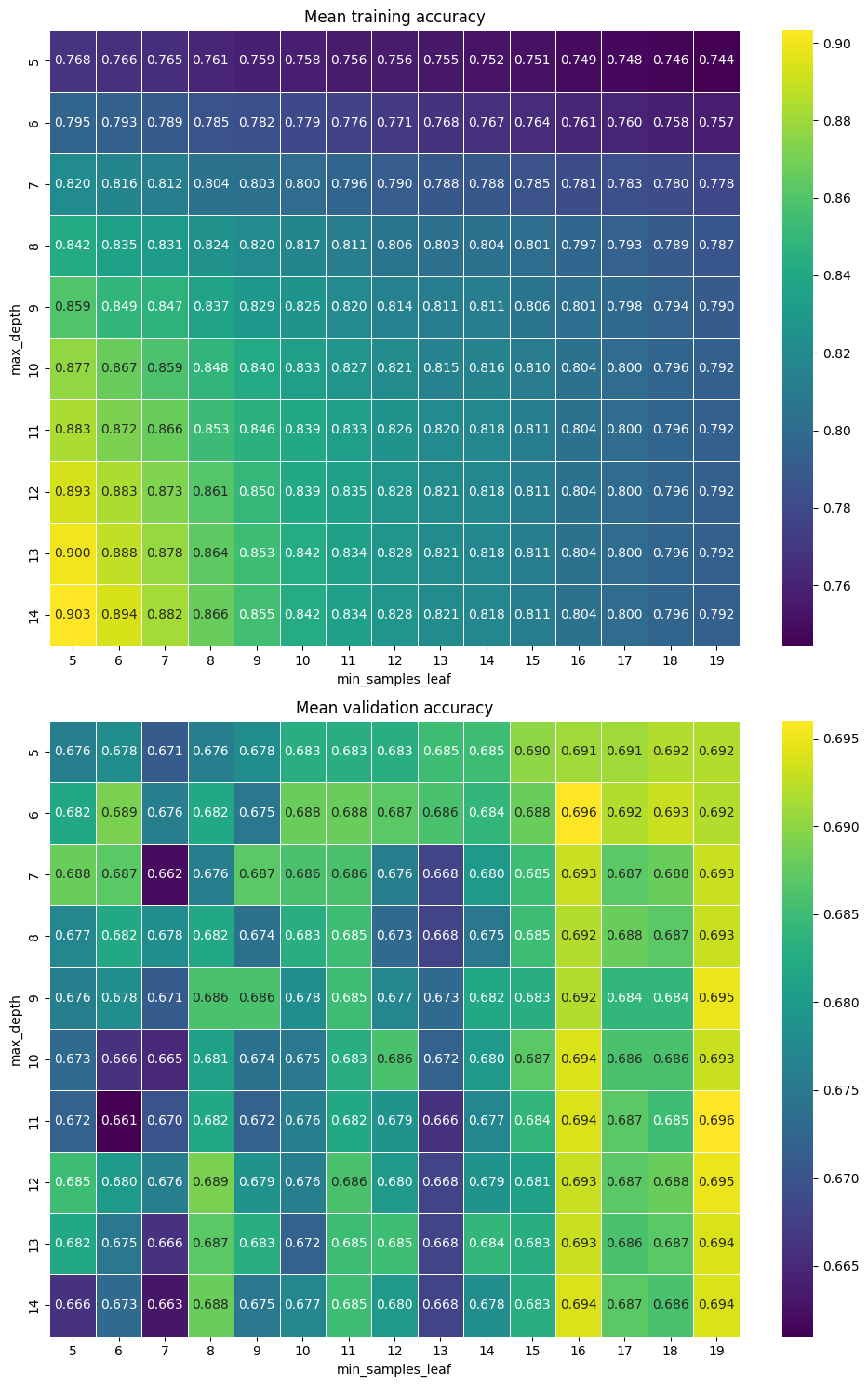
In Q1 we can see that because of the overfitting the bounderies looks dissaster and its not smooth like the bounderies in k=9 because the in k=9 we look at the best number of neighbors that will make the training look good.

Q5.

Accuracy : 0.696



Q6.



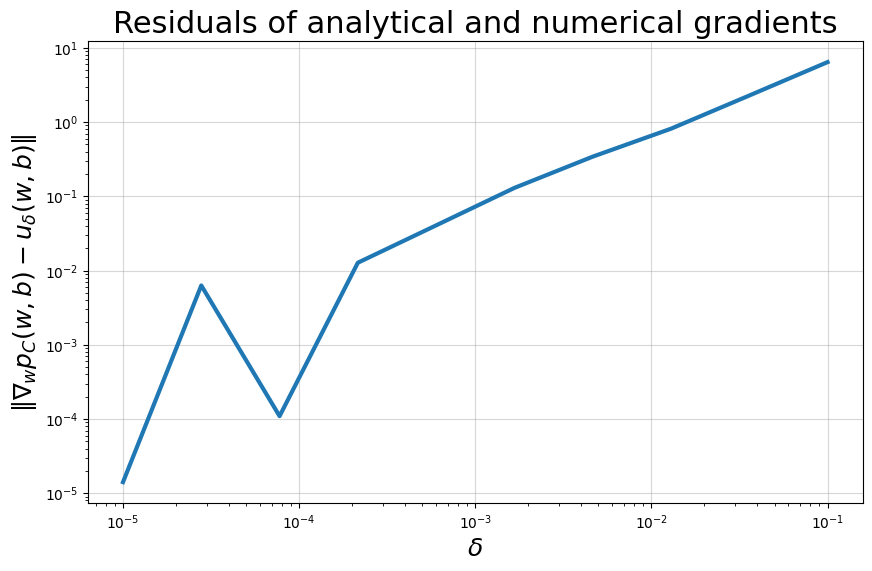
According to the "Mean validation accuracy" plot, the optimal hypermeter combination is: min\_samples\_leaf = 16 or 19, max\_depth = 6 or 11.

A hyperparameter-combination that causes underfitting, according to the "mean validation accuracy" plot, is: min\_samples\_leaf = 7, regardless of the value of max\_depth. This happens because for the above value of min\_samples\_leaf, it's hard to discern the classification.

A hyperpameter-combination that causes overfitting, according to both plots, is: min\_samples\_leaf = 5, max\_depth = 14. This is because the training accuracy is very high, but the validation accuracy is very low.

Q7.The number of hyperparameter combinations that were evaluated in our grid search is 15\* 10 = 150. Had we wished to tune a third hyperparameter, that would affect the number of combinations by multiplying it by the number of the values of this third hyperparameter. Searching over additional hyperparameters affects the total number of possible combinations by multiplying them by the number of values of each additional hypermeter.

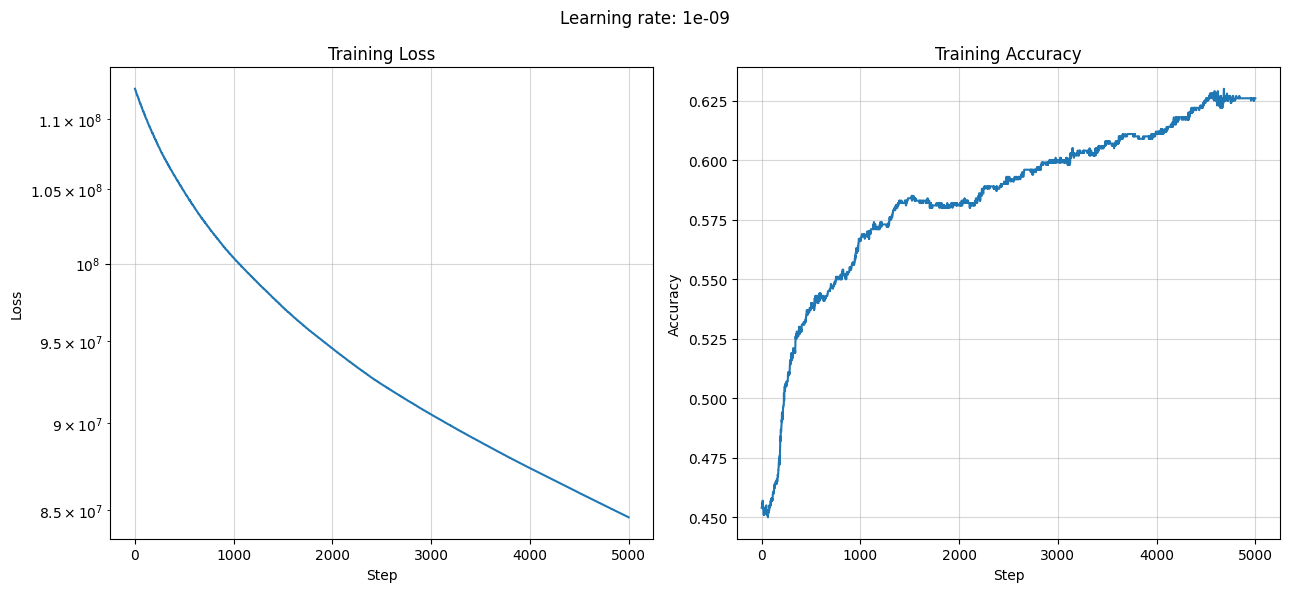
Q8. The test accuracy of this model is: 0.736.

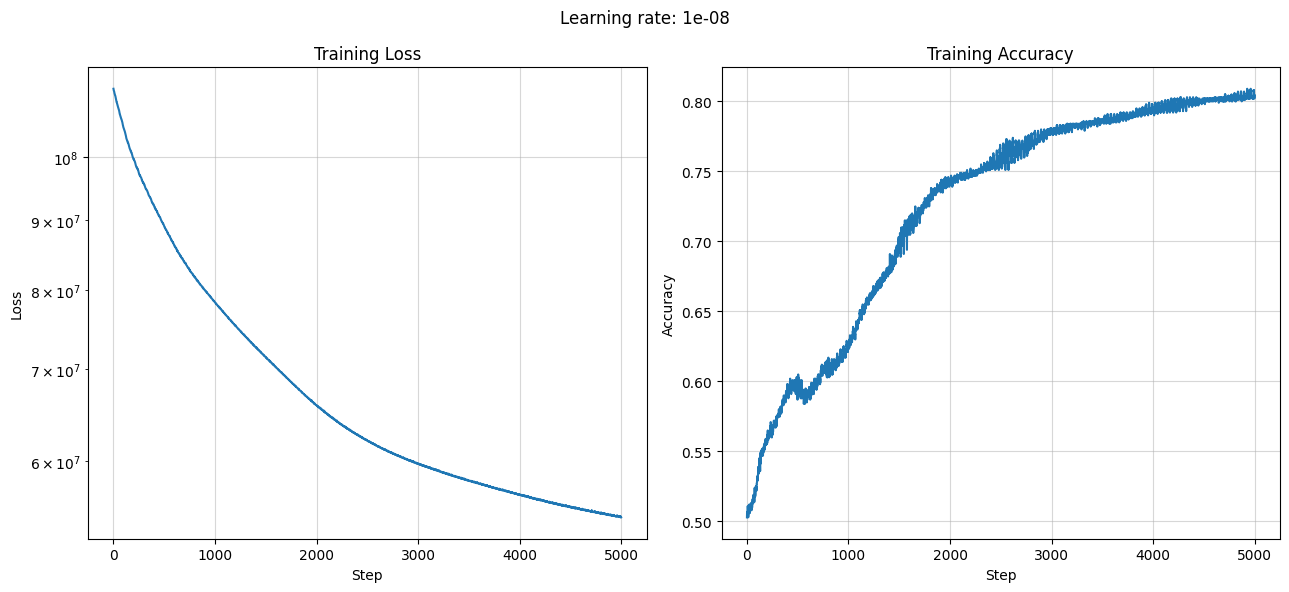
Q9. 

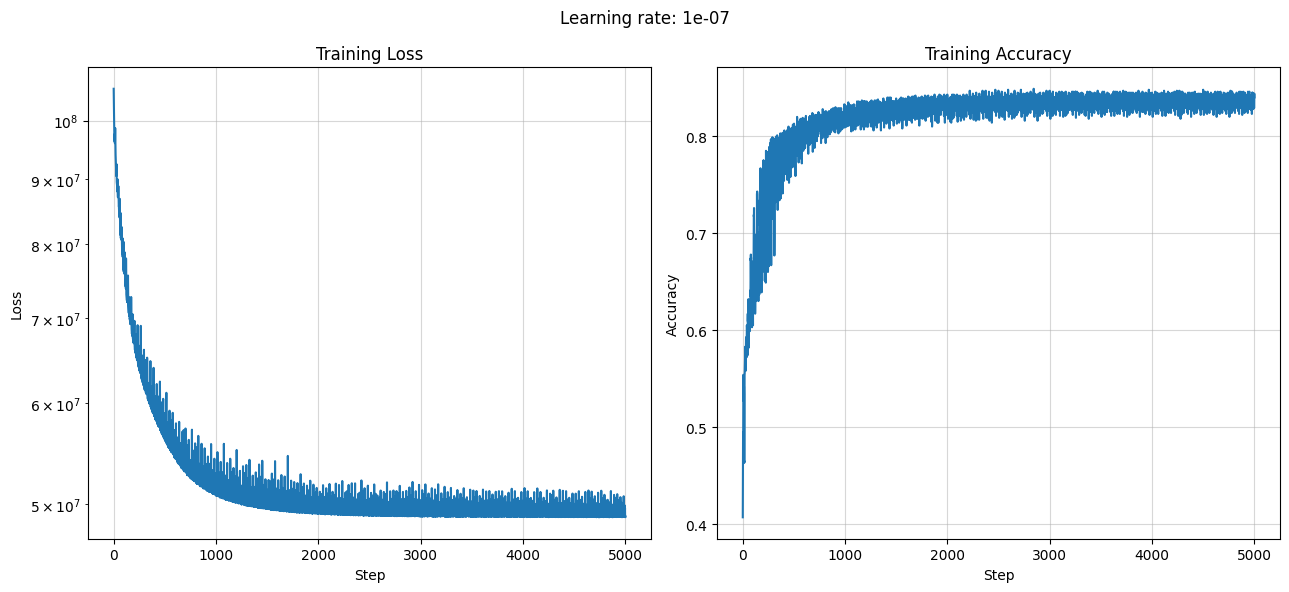
The demonstrated behavior of the plot is that from large δ's, the residuals grow. This is justified because when δ is large, the numeric approximation is less accurate, as we can see in the given formula of the derivative. But when δ is small, the numeric approximation behaves more like the derivative. On the other hand, when δ's value is very small, the numeric approximation behaves less like the derivative. This happens because of computational errors that are caused by δ's extremely small value.

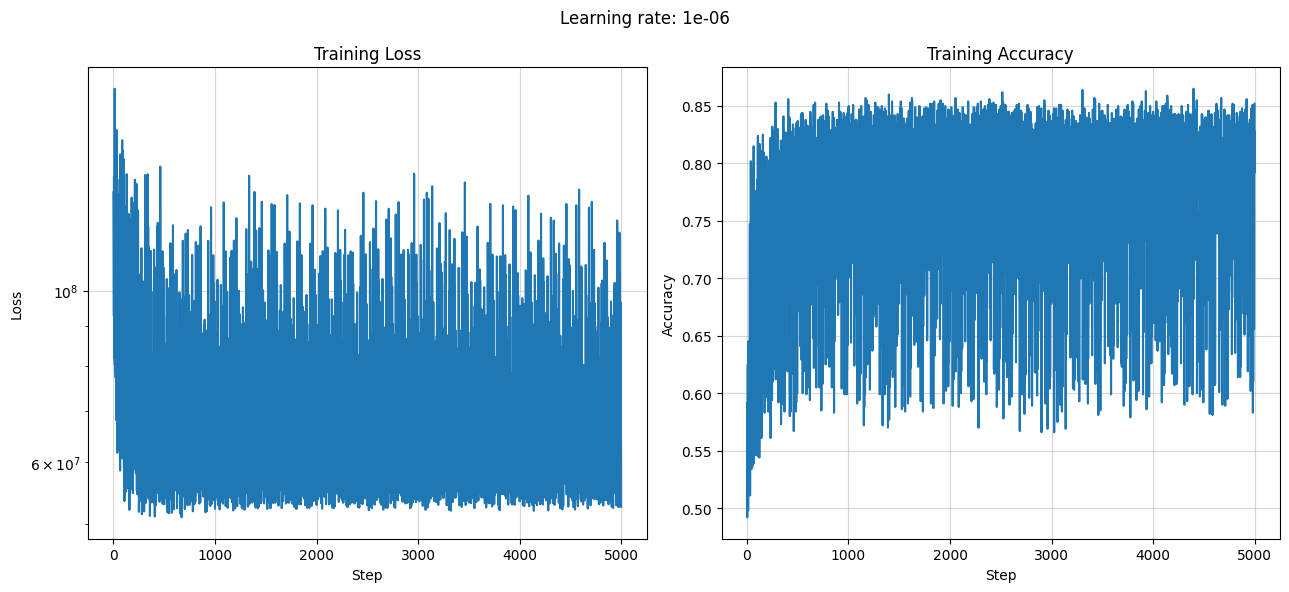
Q10. The interaction matches our expectations: We expect that for large values of C (i.e., a very regularized classifier), we get overfitting. For very small values of lr (i.e, a very regularized classifier), we also get overfitting.

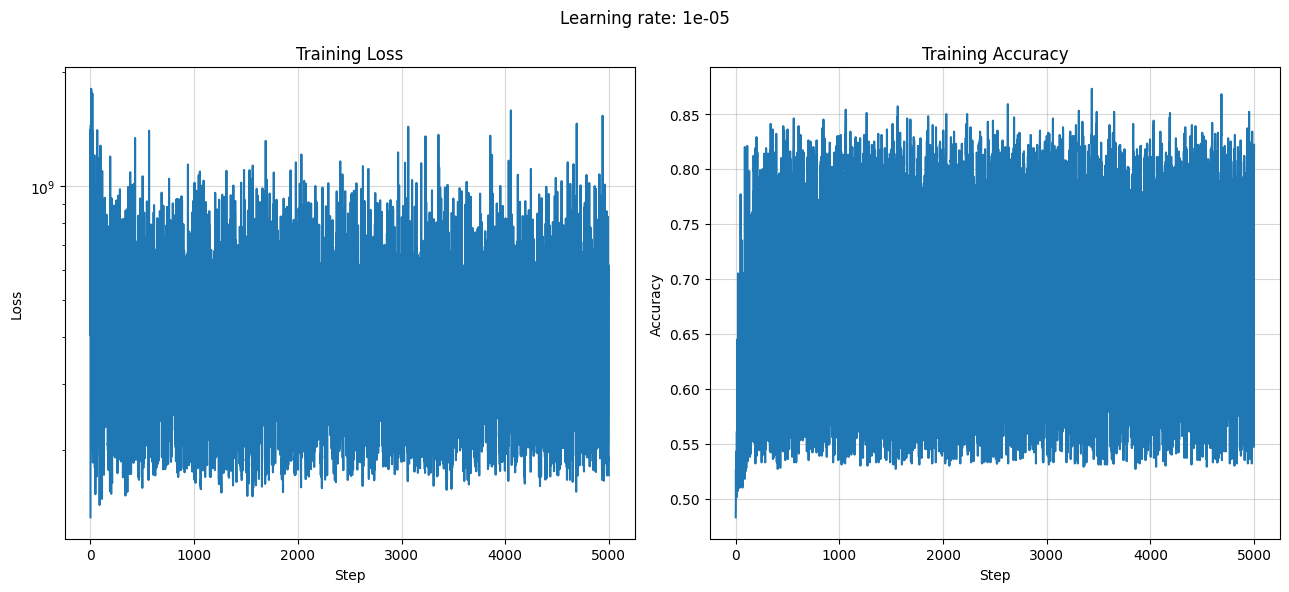
Q11.





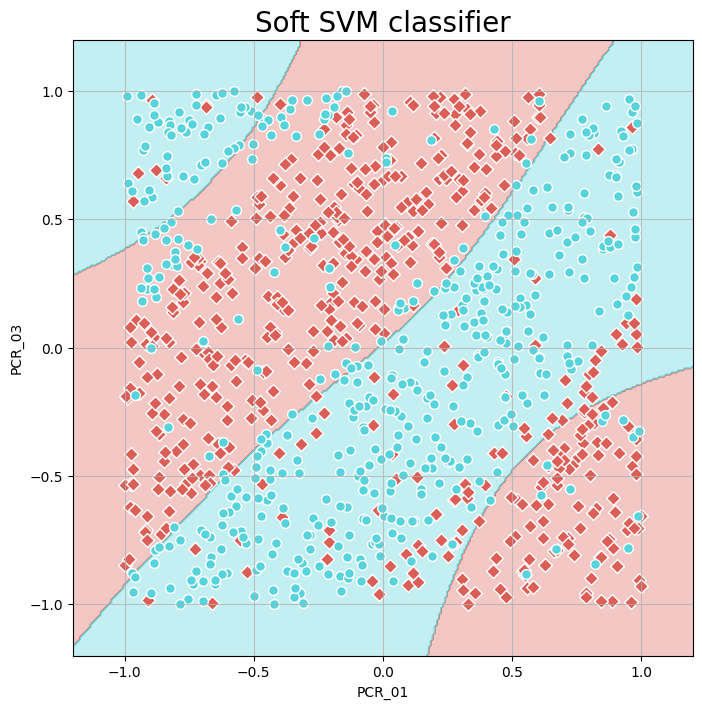






According to the above plots, we would choose the learning rate of 1e-07. We would do so because when the learning rate is 1e-05 or 1e-06, the plot diverges. When the learning rate is 1e-08 or 1e-09, the plot doesn't converge fast enough. Therefore, the learning rate of 1e-07 is the perfect "in between" value – it converges quicker than the lr values of 1e-08, 1e-09 and does not diverge like in lr values of 1e-05, 1e-06.

Q12.

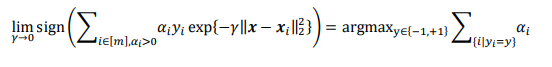


Train score: 0.803

Test score: 0.712

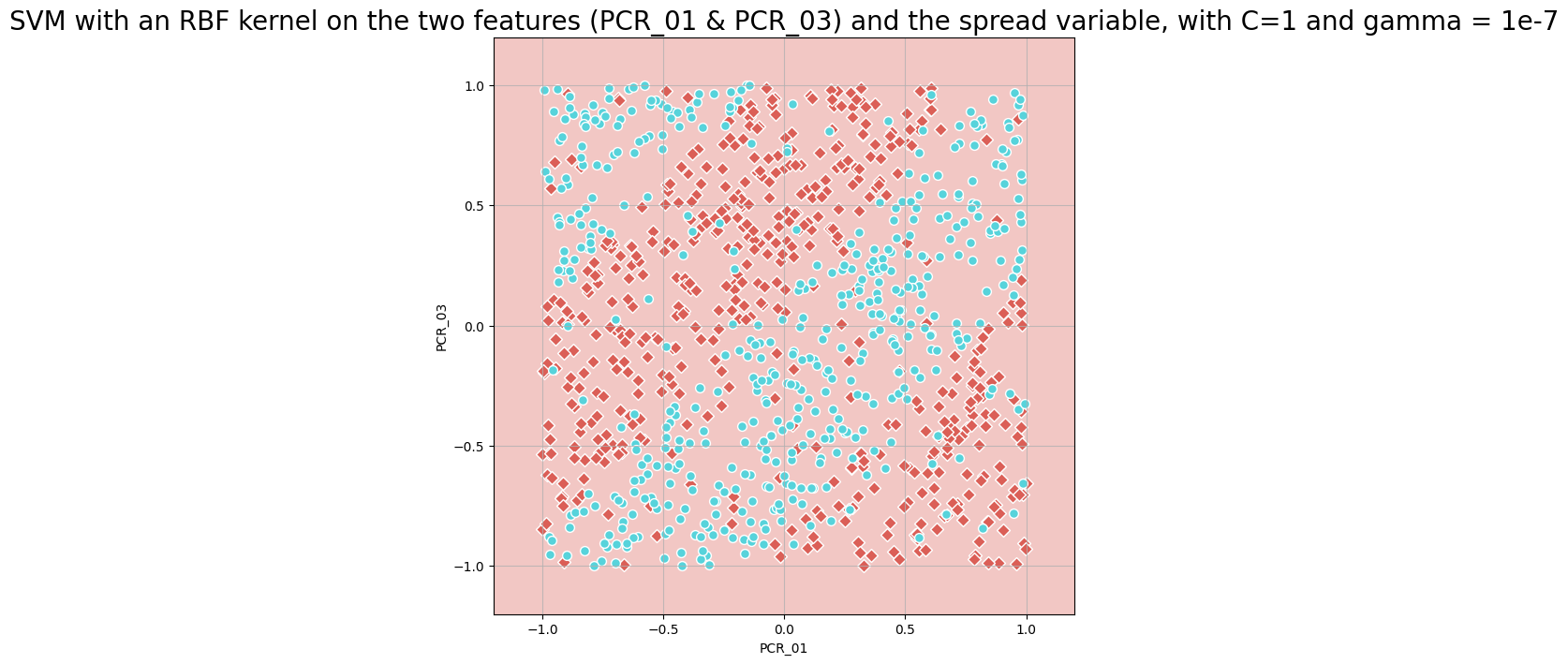
Q13.

Section a.

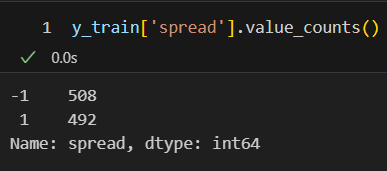
We wiil prove: we can assume that

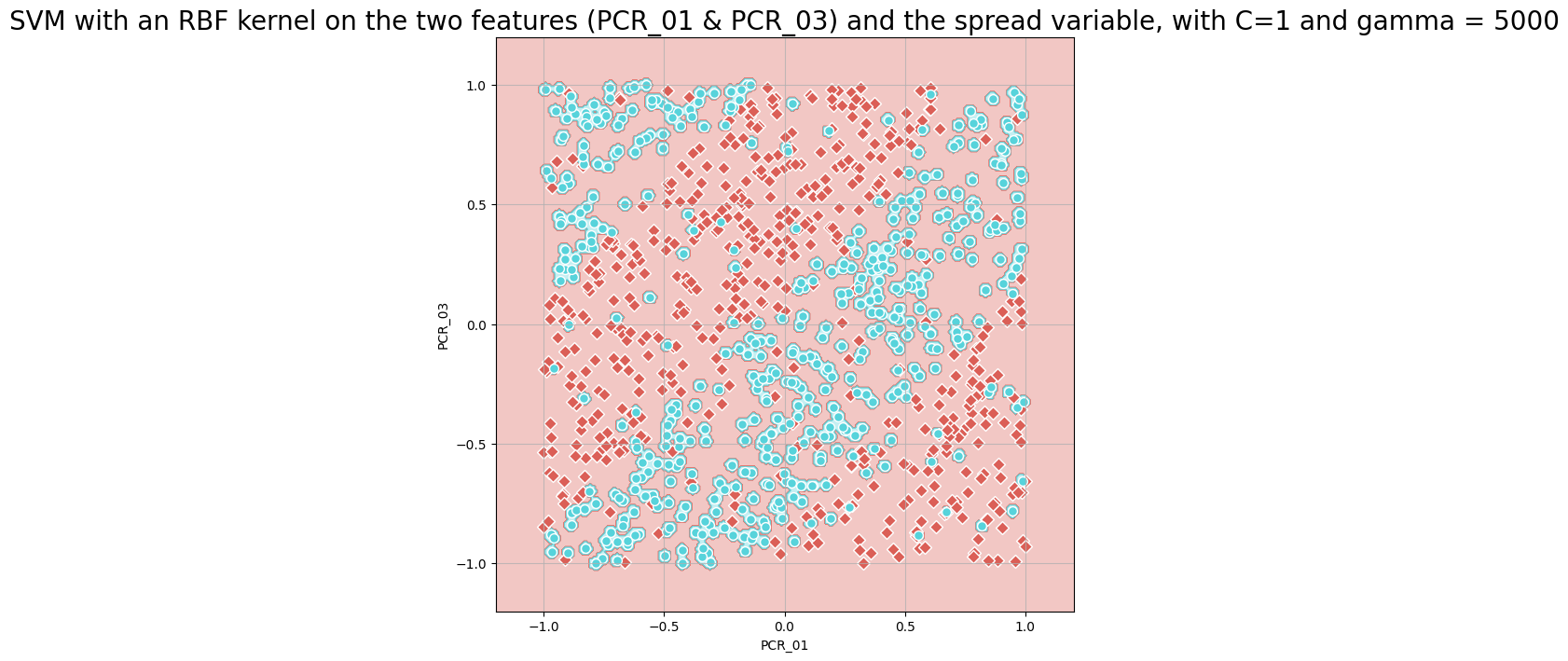
For each I:

Fixed 𝑖 value. It holds that:   
• 𝛼𝑖 is a bounded constant by assumption (iii)   
• 𝑦𝑖 ∈{±1}   
• By assumption (ii) ∃𝑐1 such that all features are bounded by 𝑐1,   
therefore: ‖𝑥−𝑥𝑖‖22 ≤‖𝑥‖22 +‖𝑥1‖22 ≤2𝑐12 ⇒−|𝛾|∙2𝑐12 ≤−𝛾∙‖𝑥−𝑥𝑖‖22 ≤|𝛾|∙2𝑐12   
𝑒𝑥 is monotonic, therefore ∀𝛾∈ℝ: 𝑒−|𝛾|∙2𝑐12 ≤𝑒−𝛾‖𝑥−𝑥𝑖‖22 ≤𝑒|𝛾|∙2𝑐12  
   
𝑒𝑥 is continuous, therefore lim𝑥→𝑥0𝑒𝑥 =𝑒𝑥0, thus:   
lim𝛾→0𝛼𝑖 ∙𝑦𝑖 ∙𝑒−|𝛾|∙2𝑐12= lim𝛾→0+ 𝛼𝑖 ∙𝑦𝑖 ∙𝑒−𝛾∙2𝑐12 =𝛼𝑖 ∙𝑦𝑖 ∙𝑒−0∙𝑐12= ∙𝑦𝑖 ∙1   
lim𝛾→0𝛼𝑖 ∙𝑦𝑖 ∙𝑒|𝛾|∙2𝑐12= lim𝛾→0− 𝛼𝑖 ∙𝑦𝑖 ∙𝑒−𝛾∙2𝑐12=𝛼𝑖 ∙𝑦𝑖 ∙𝑒−0∙𝑐12=𝛼𝑖 ∙𝑦𝑖 ∙1   
   
By the squeeze theorem, we’ll get that lim𝛾→0𝛼𝑖 ∙𝑦𝑖 ∙𝑒−𝛾∙‖𝑥−𝑥𝑖‖22=𝛼𝑖𝑦𝑖.   
   
Since the limit exists for each 𝑖, the limit of the sum of all 𝑖 ∈[𝑚] also exists,   
and is equal to the sum of the limits, therefore:   
lim𝛾→0( ∑ 𝛼𝑖 ∙𝑦𝑖 ∙𝑒−𝛾‖𝑥−𝑥𝑖‖22)= ∑ 𝛼𝑖 ∙𝑦𝑖 ∙1  
 𝑖∈[𝑚],𝛼𝑖>0 𝑖∈[𝑚],𝛼𝑖>0  
  
And it holds that:   
∑ 𝛼𝑖 ∙𝑦𝑖= ∑ 𝛼𝑖 -   
𝑖∈[𝑚],𝛼𝑖>0 𝑖∈[𝑚], 𝛼𝑖>0  
 𝑦𝑖=+1  
  
  
   
And so, the above is greater than 0 ⇔∑ 𝛼𝑖𝑖∈[𝑚],  
𝛼𝑖>0  
𝑦𝑖=+1  
>∑ 𝛼𝑖𝑖∈[𝑚],  
𝛼𝑖>0  
𝑦𝑖=−1  
, so the sign value of the above can be rewritten as: 𝑎𝑟𝑔𝑚𝑎𝑥𝑦∈{±1}  
∑ 𝛼𝑖{𝑖|𝑦𝑖 =𝑦}   
And therefore:   
𝑠𝑖𝑔𝑛(lim𝛾→0𝛼𝑖 ∙𝑦𝑖 ∙𝑒−𝛾‖𝑥−𝑥𝑖‖22  
)=𝑠𝑖𝑔𝑛( ∑ 𝛼𝑖 ∙𝑦𝑖  
𝑖∈[𝑚],𝛼𝑖>0  
)=𝑎𝑟𝑔𝑚𝑎𝑥𝑦∈{±1}  
∑ 𝛼𝑖  
{𝑖|𝑦𝑖 =𝑦}  
   
Section b.   
The decision rule becomes rule by majority, or something like “∞ nearest   
neighbors”, since, as shown in the previous section, the classification rule is a   
constant label determined by the limit of the sign of the sums, which is equal   
to 𝑎𝑟𝑔𝑚𝑎𝑥𝑦∈{±1}  
(∑ 𝛼𝑖{𝑖|𝑦𝑖 = 𝑦} ), and since 𝛼𝑖 =1 it holds that:   
𝑎𝑟𝑔𝑚𝑎𝑥𝑦∈{±1}  
∑ 𝛼𝑖  
{𝑖|𝑦𝑖 =𝑦}  
=𝑎𝑟𝑔𝑚𝑎𝑥𝑦∈{±1}  
∑ 1  
{𝑖|𝑦𝑖 =𝑦}  
=𝑎𝑟𝑔𝑚𝑎𝑥𝑦∈{±1}  
(|{𝑦𝑖 =𝑦|𝑖 ∈[𝑚]}|)   
Which is equal to the majority label.

Q14.

The decision boundary of the above plot matches the decision rule discussed in Q13 section b, because the classification that the classifier gave is the same as the classification of the training points that uphold that their 'spread' variable is -1. This value of the 'spread' variable is in fact the majority label of the training points:



Q15.