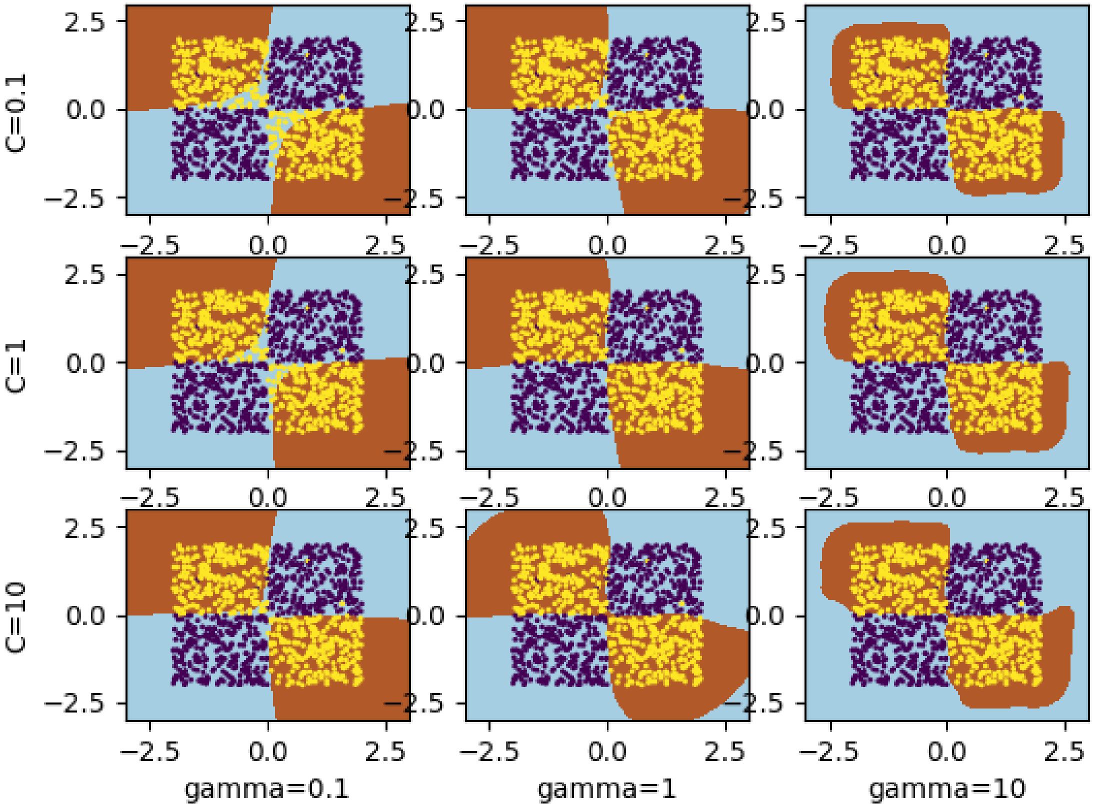
ML Homework 5

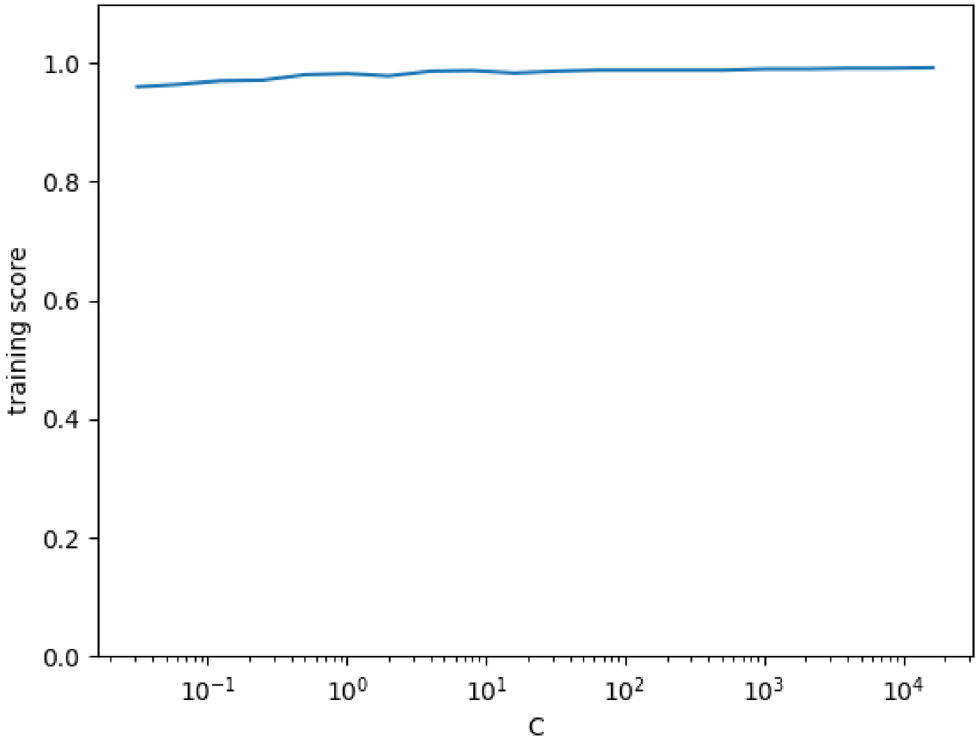
Autor: Marián Kravec (Prosím Vlado nestrácaj s kontrolou tohto veľa času, nevedel som čo k tomu chcem napísať, ale keď už som vygeneroval grafy tak som sa rozhodol to odovzdať)

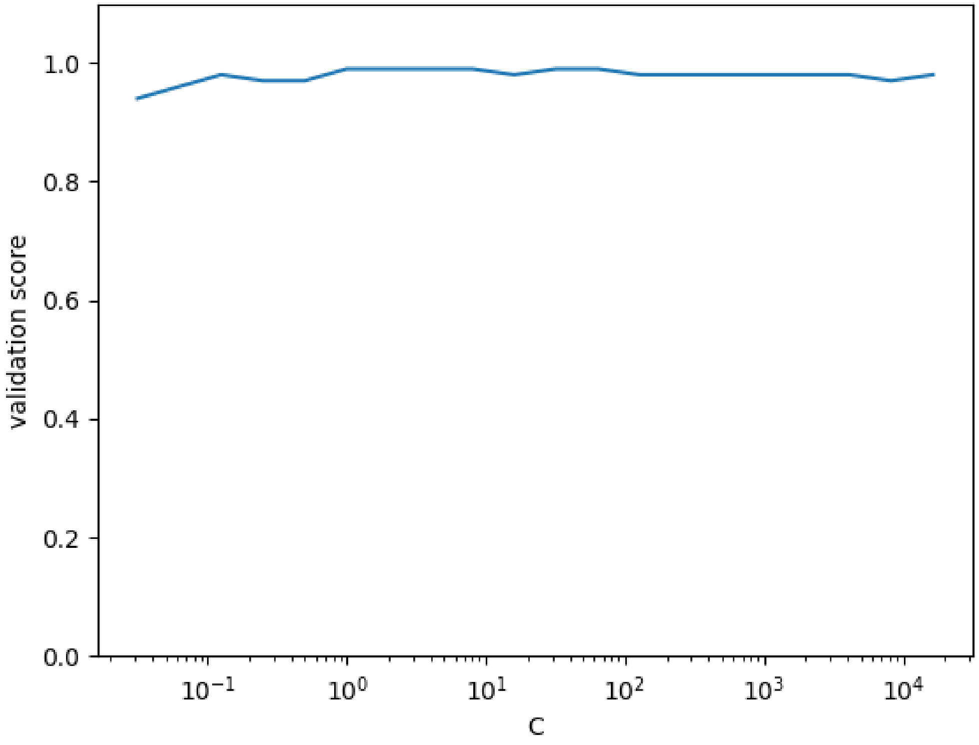
We will create dataset using provided function that create data with XOR pattern, however we add more functionality to generative function. We add possibility to set noise coefficient (fraction of samples that would be incorrectly labeled) and to set center of decision boundary.
We will create 3 training datasets. First will have 1000 samples (will be biggest) with small noise (only 1%) and decision boundary will stay in middle. It can be considered as desired dataset (big, with little noise). Second will opposite of the first with only 100 samples and high noise (25%), it will also have moved decision boundary higher (it will split y-axis into $1/4$ and $3/4$ instead of $1/2$ and $1/2$). Third dataset will be something in-between from size and noise values with 300 samples and 10% noise, however decision boundary will be moved closer to top right corner and will split both axis into $1/4$ and $3/4$.
For each of these datasets we prepare validation dataset with 100 samples and same values noise and decision boundary as respective training set.

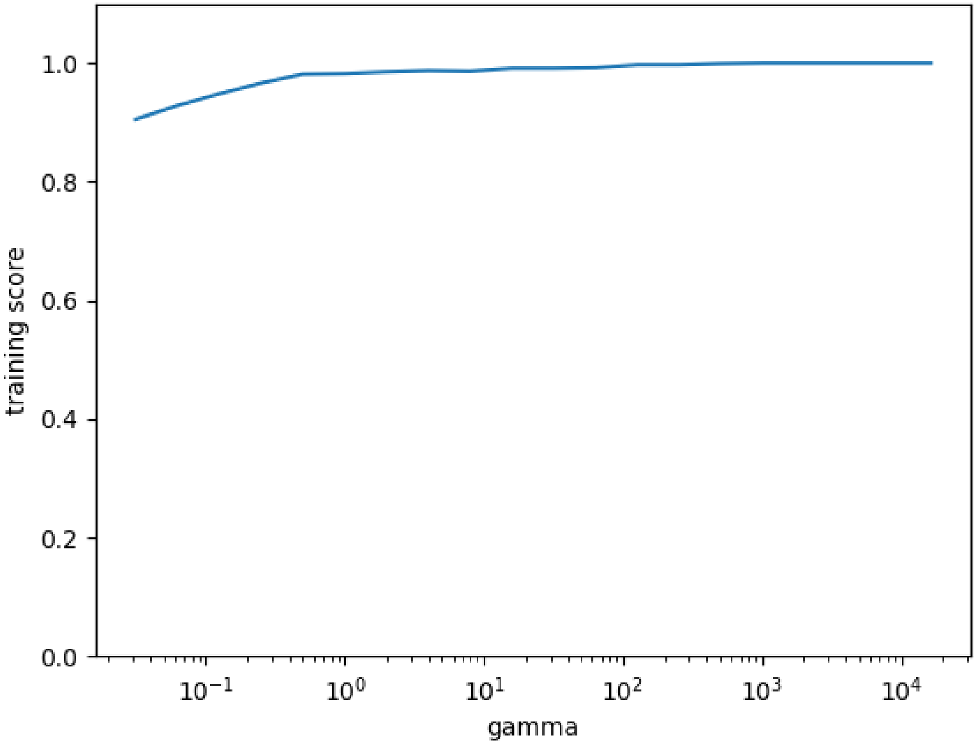
Let's start with first dataset. Firstly let's take a look at how SVD compute boundaries for combination of 3 values of C (0.1, 1, 10) and 3 values of γ (0.1, 1, 10):

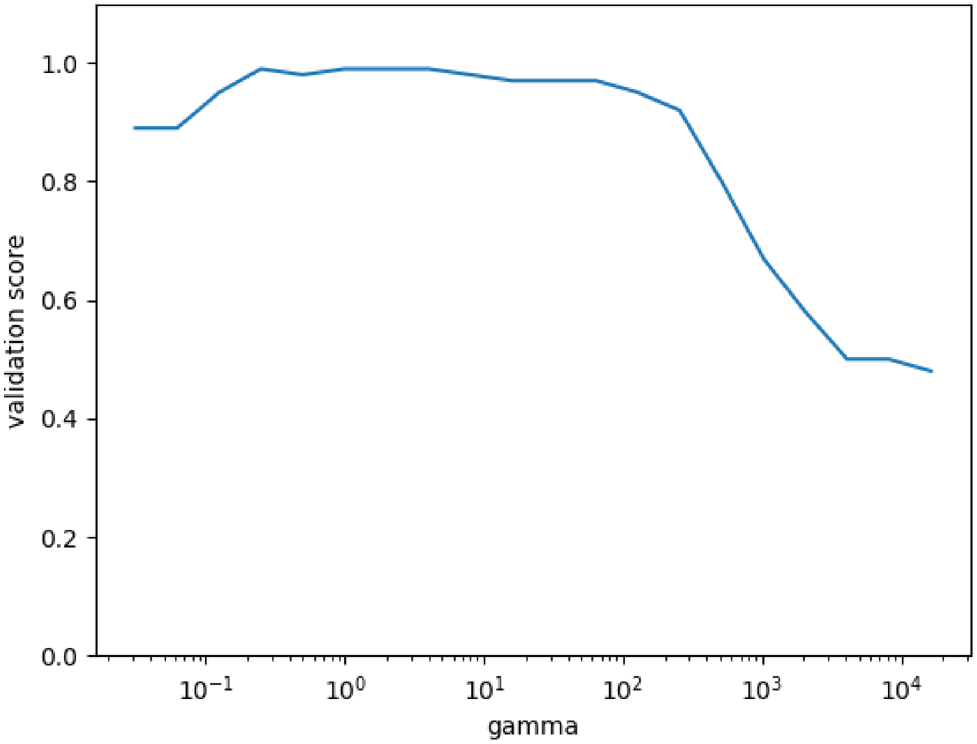


Now let's plot scores of prediction for different values of C and γ for both training and validation dataset (plots with different C values have $\gamma = 1$ and plots with different γ values have $C = 1$):

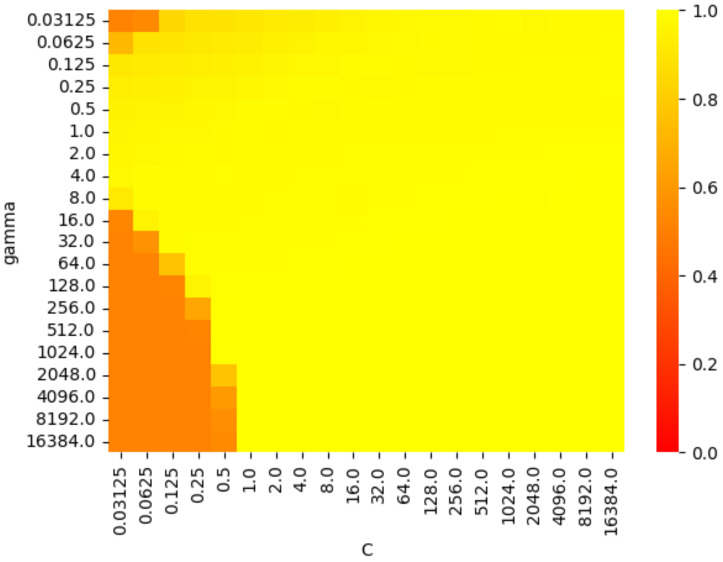

training score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
C

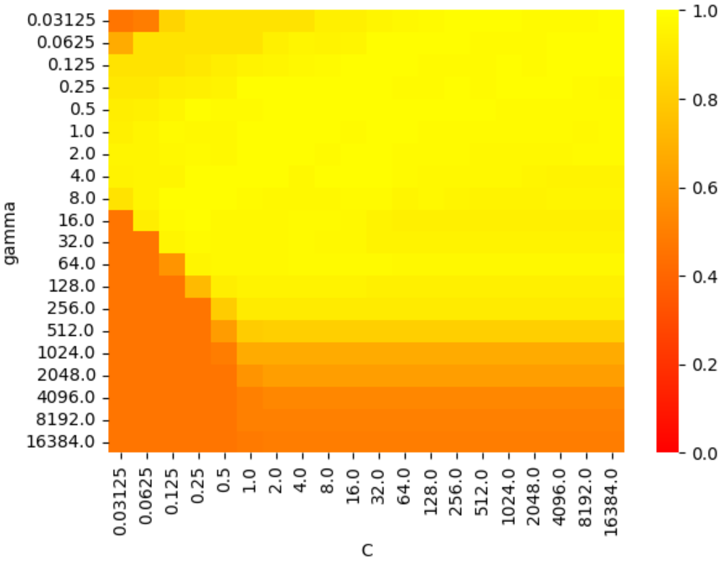

validation score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
C


training score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
gamma


validation score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
gamma

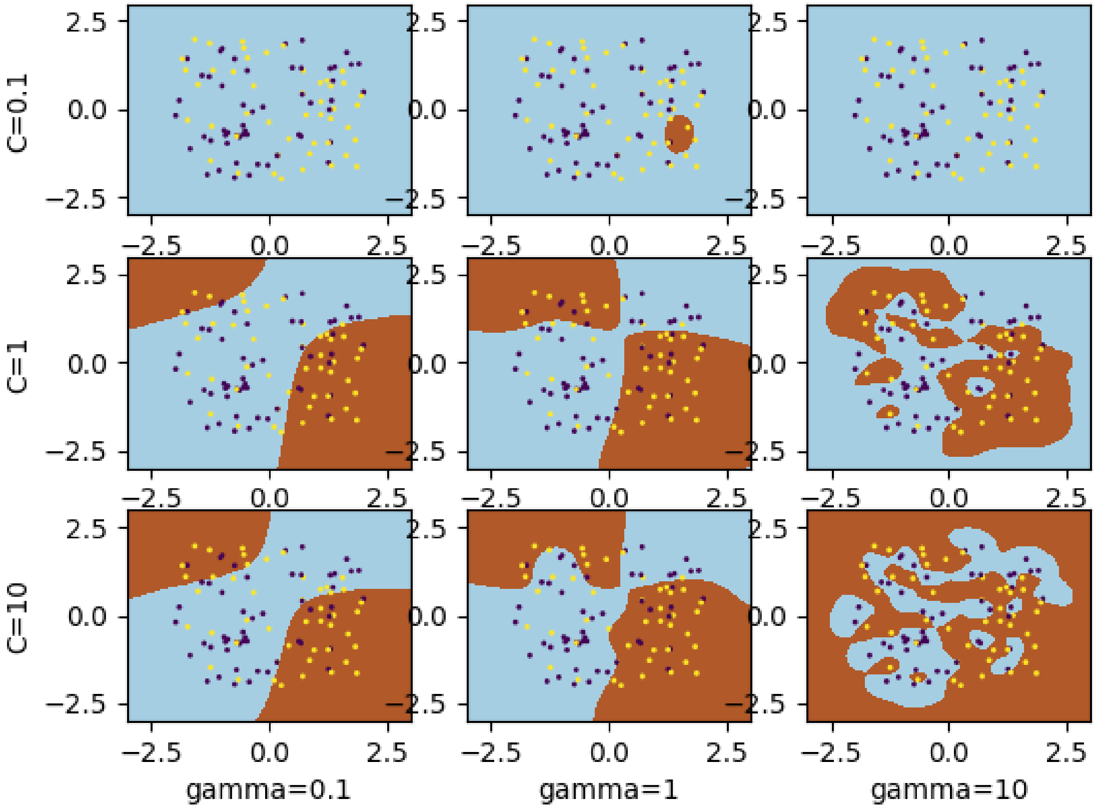
In both cases (both C and γ) we can see that as values of hyperparameter increase training score get closer to 1 (perfect score). How ever in case of validation score we can see that in case of C hyperparameter this score fluctuate close to 1 for all values but in case of γ hyperparameter validation score at first increase (it's lower for very small values) then it fluctuate close to 1 but then it suddenly start dropping which mostly likely mean that those models are overfitted.
As a final plot let's take a look at heatmaps for different values of C and γ for both training and validation data:



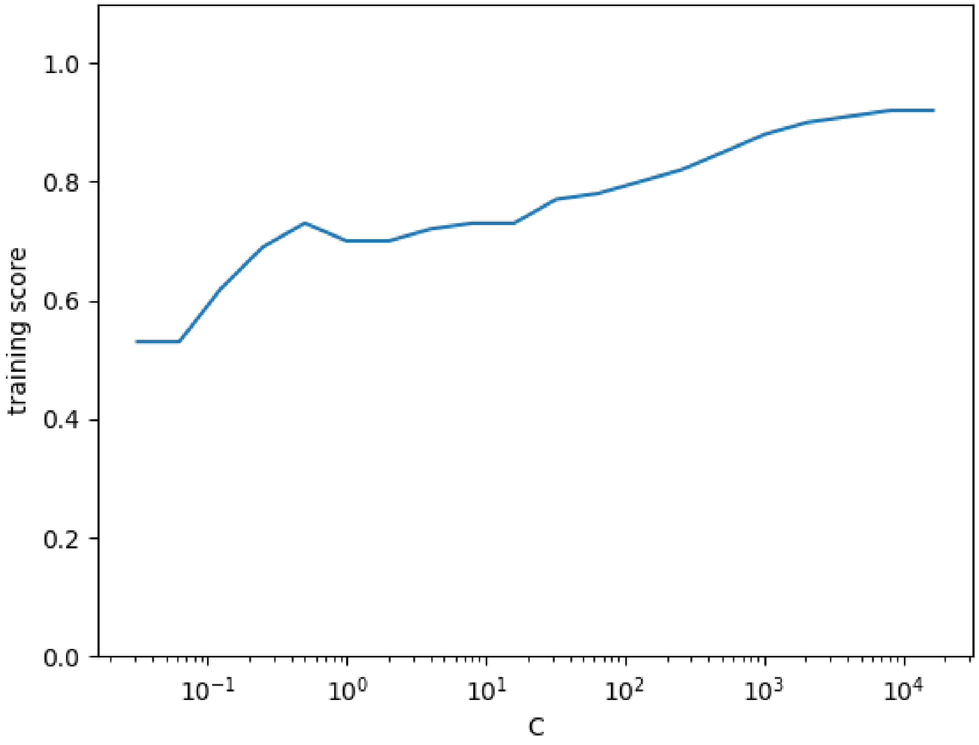


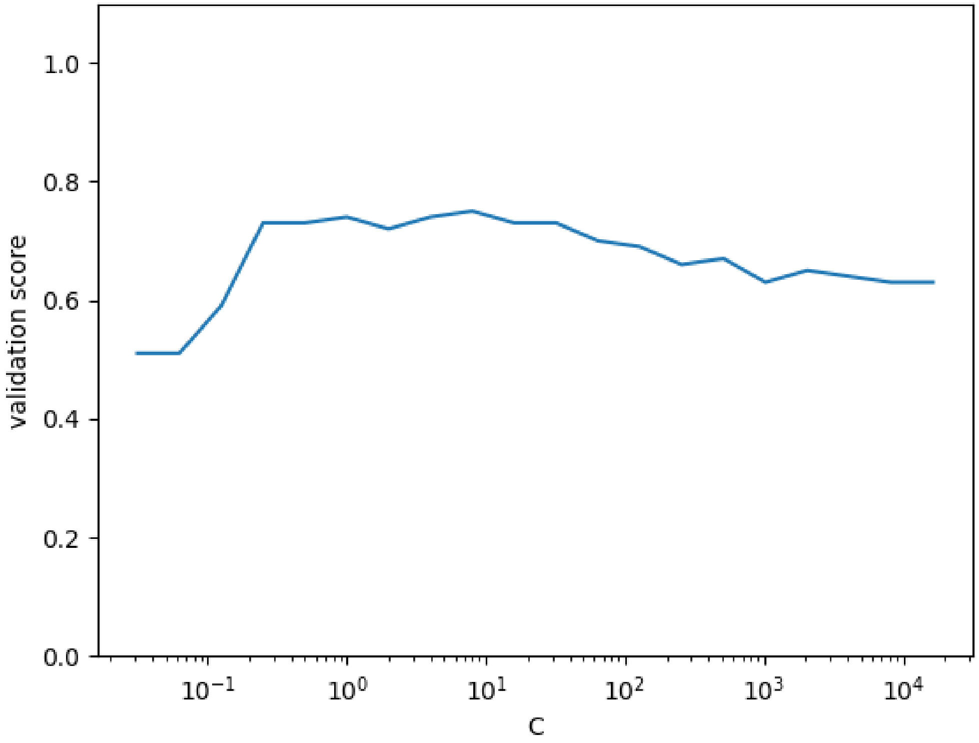
We can see similar behavior as in lineplots however we also see that for small values C and large values of γ we get decreasing even training score.

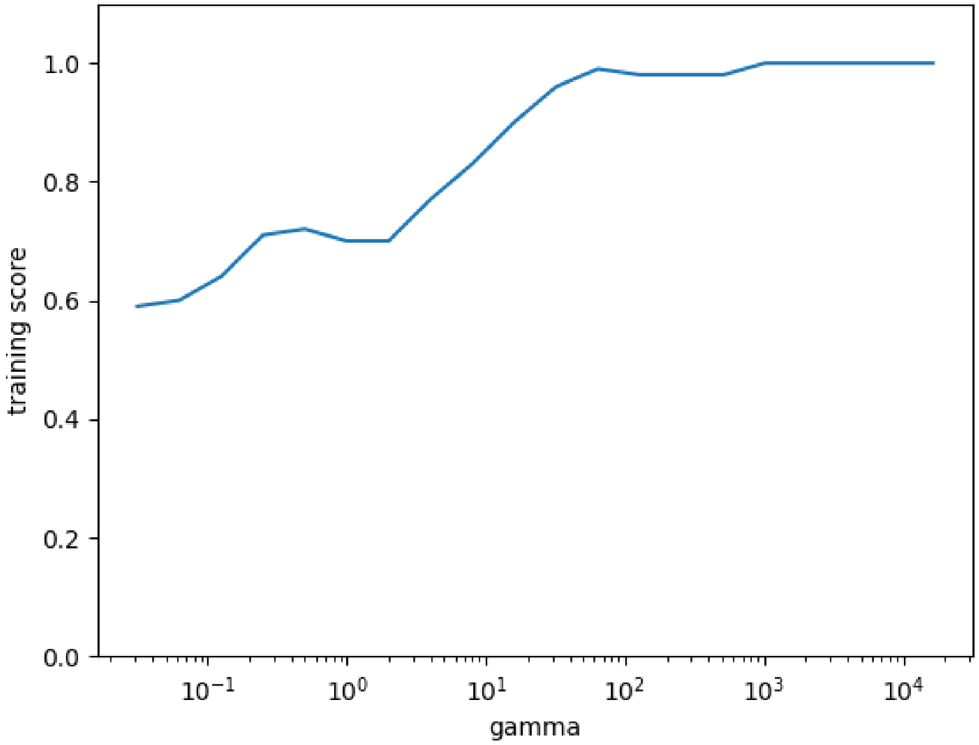
Now let's move to second dataset (we will create same plots).

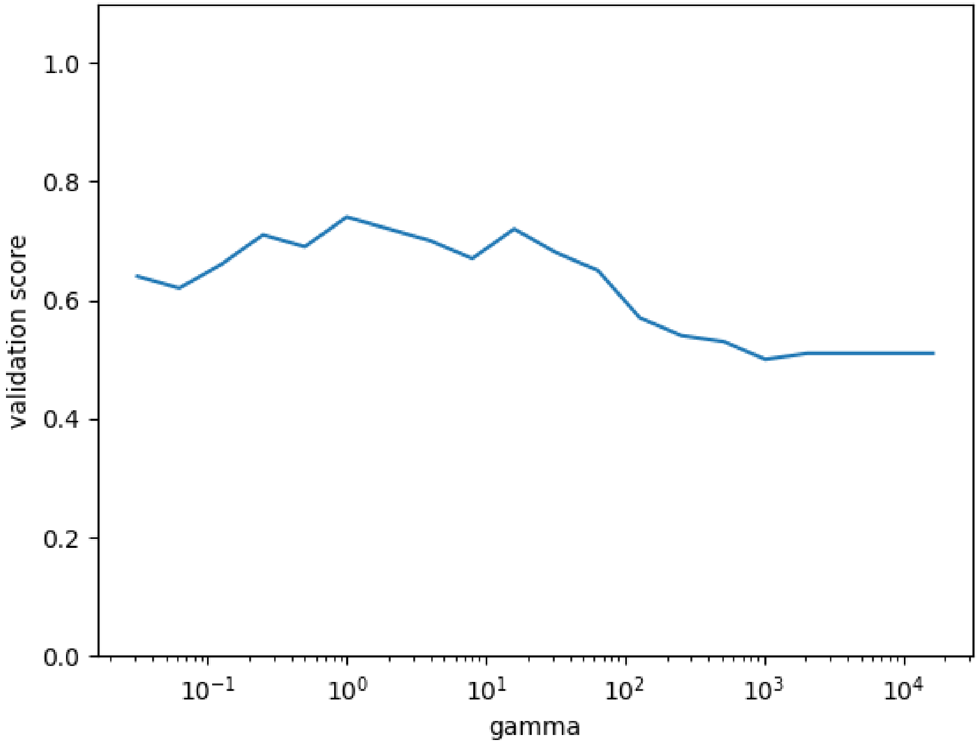


In this case we can see that small values of C have hard time identifying any yellow point in such noisy dataset. We can also see opposite extreme where for large values of both hyperparameters it tries to identify all points perfectly which leads to overfitting.

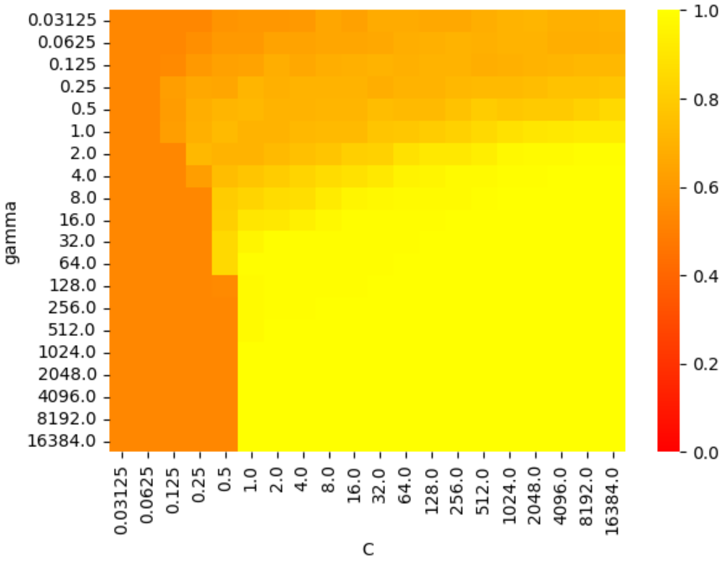

training score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
C

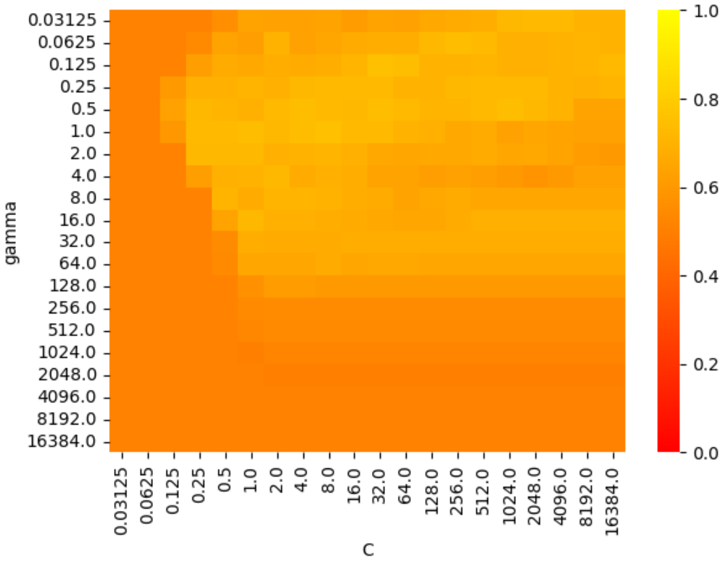

validation score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
C


training score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
gamma

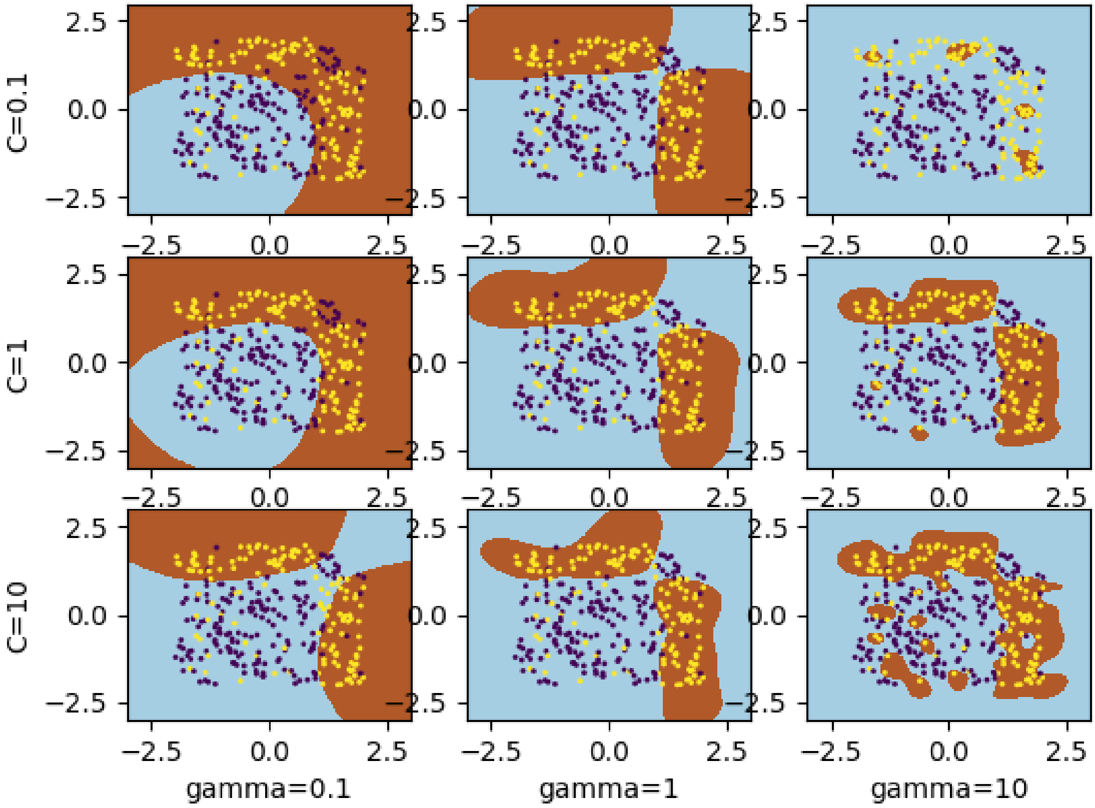

validation score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
gamma

In lineplots we can almost perfectly same behavior as in first dataset (with only different that because of noise validation score is fluctuating around lower numbers)

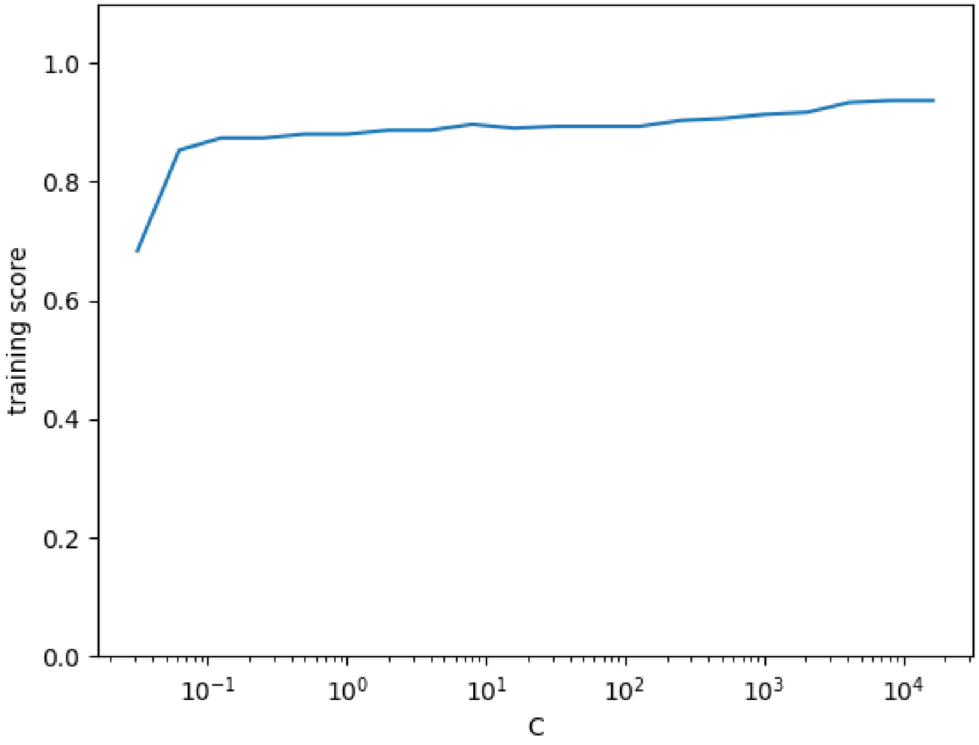




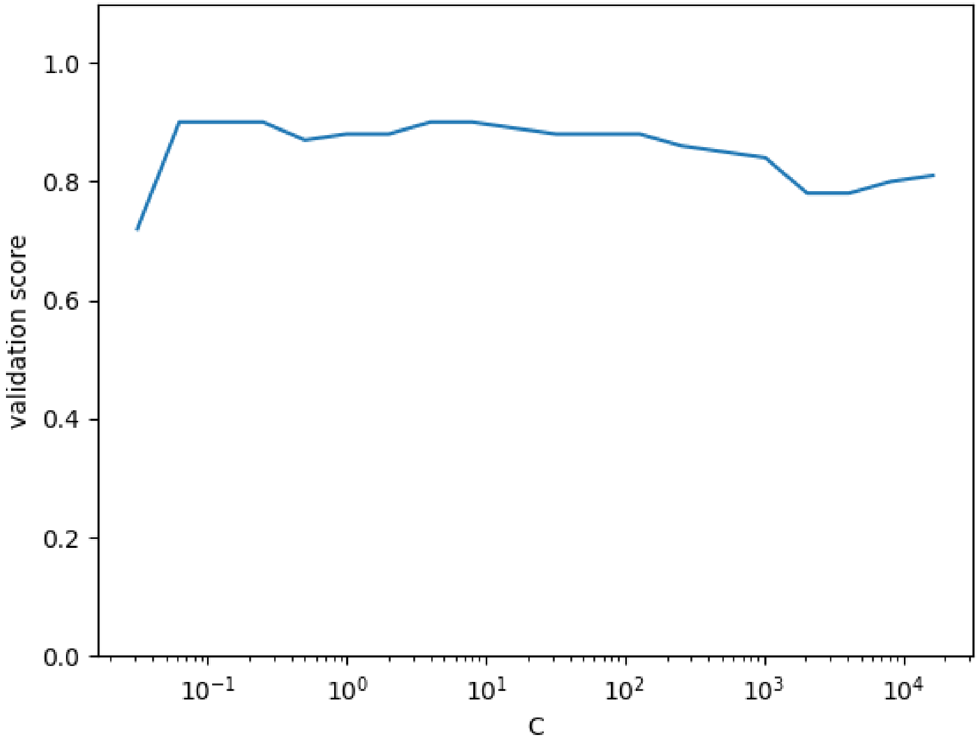
Now we take a look at third dataset:



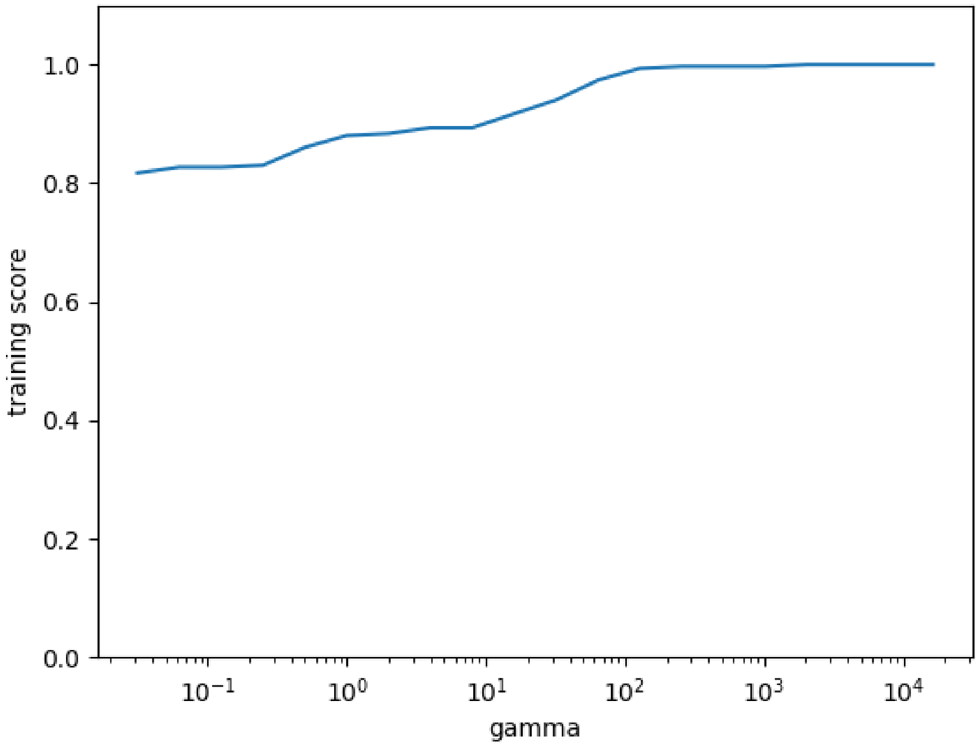
Here is similar behavior to second with only difference that small values of C can identify yellow at least to some extend (at least for smaller values of γ) most likely thanks to lower noise.

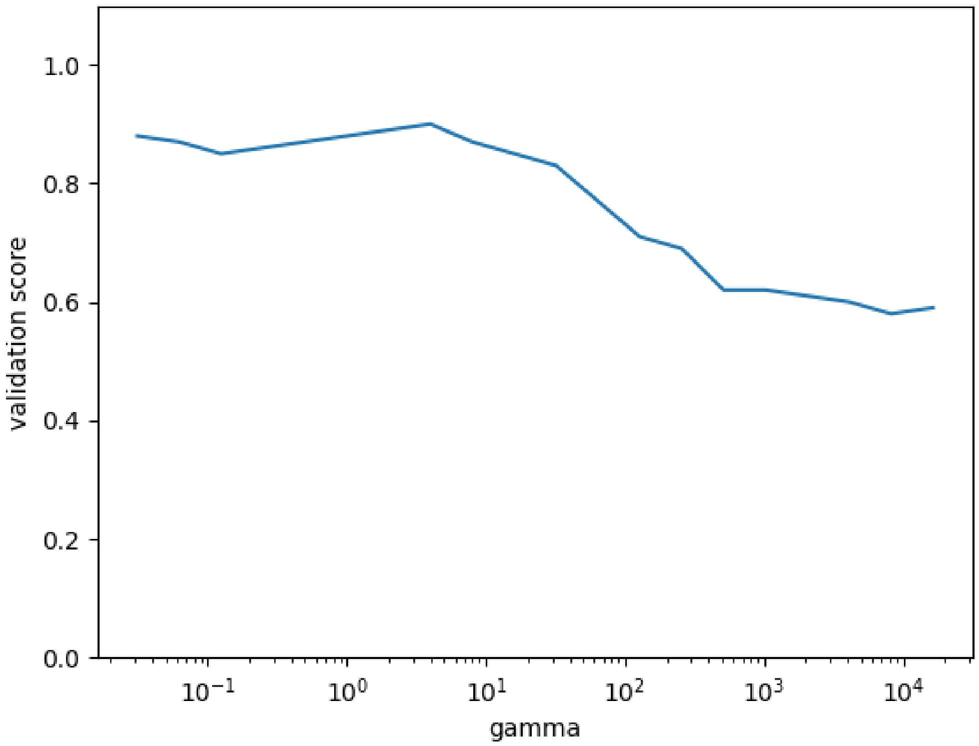


training score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
C

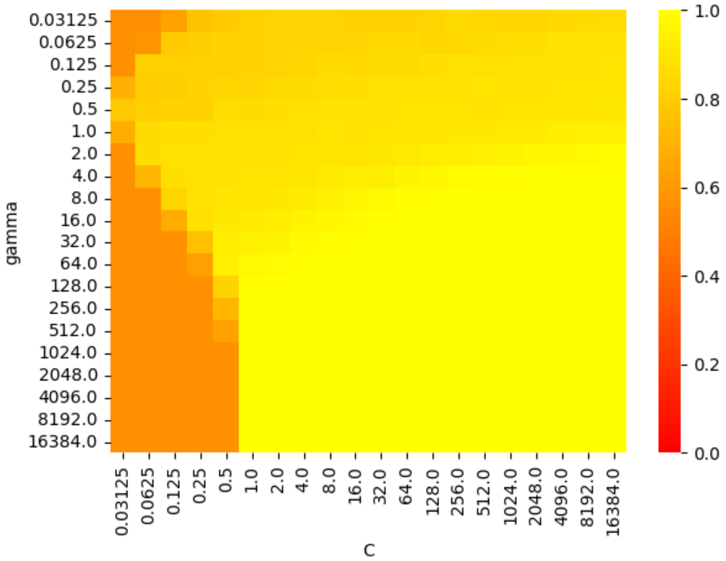


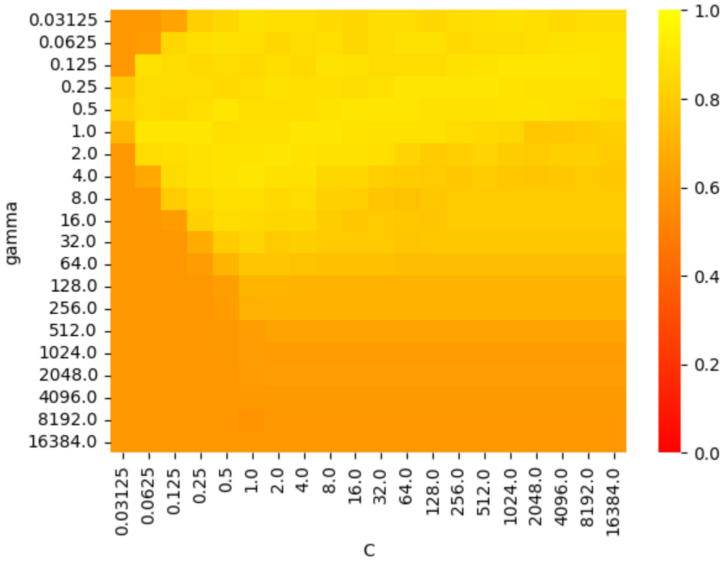
validation score
1.0
0.8
0.6
0.4
0.2
0.0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
C


1.0
0.8
0.6
0.4
0.2
0.0
training score
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
gamma


1.0
0.8
0.6
0.4
0.2
0.0
validation score
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
gamma

Lineplots are more or same as in previous datasets.





All in all we can say that we both hyperparameters behave similarly in a way that as they increase model is trying to fit training data more precisely. Hyperparameter γ seems to bigger influence than C.