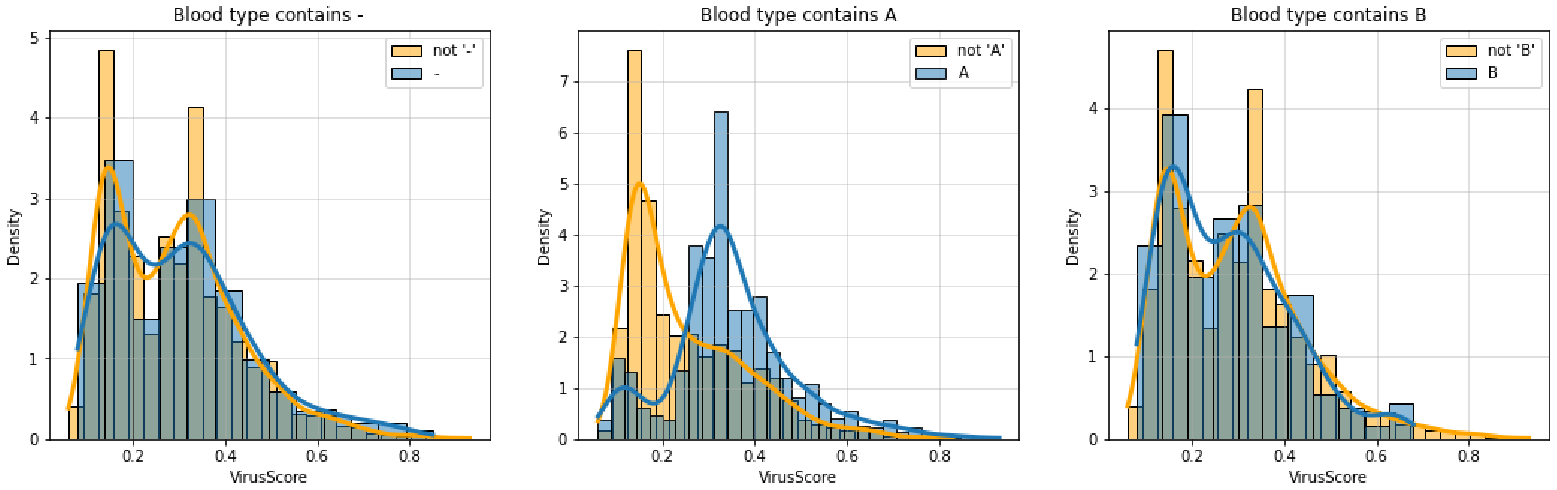
Submitted by: Eden Dembinsky 212227888 & Assaf Lovton 2098444414 - HW3

Section 1: Quick data exploration and preparation

Q1.



Q2.

As we can see “blood type contains -” and “blood type contains B” does not provide us with any interesting information since for both groups we find almost identical VirusScore. But, “blood type contains A” reveals that blood type that does not contain A is more likely to get a low VirusScore while blood types that contain A are likely to get a higher VirusScore. So, to conclude, the most informative way to learn the VirusScore target variable is “blood type contains A”.

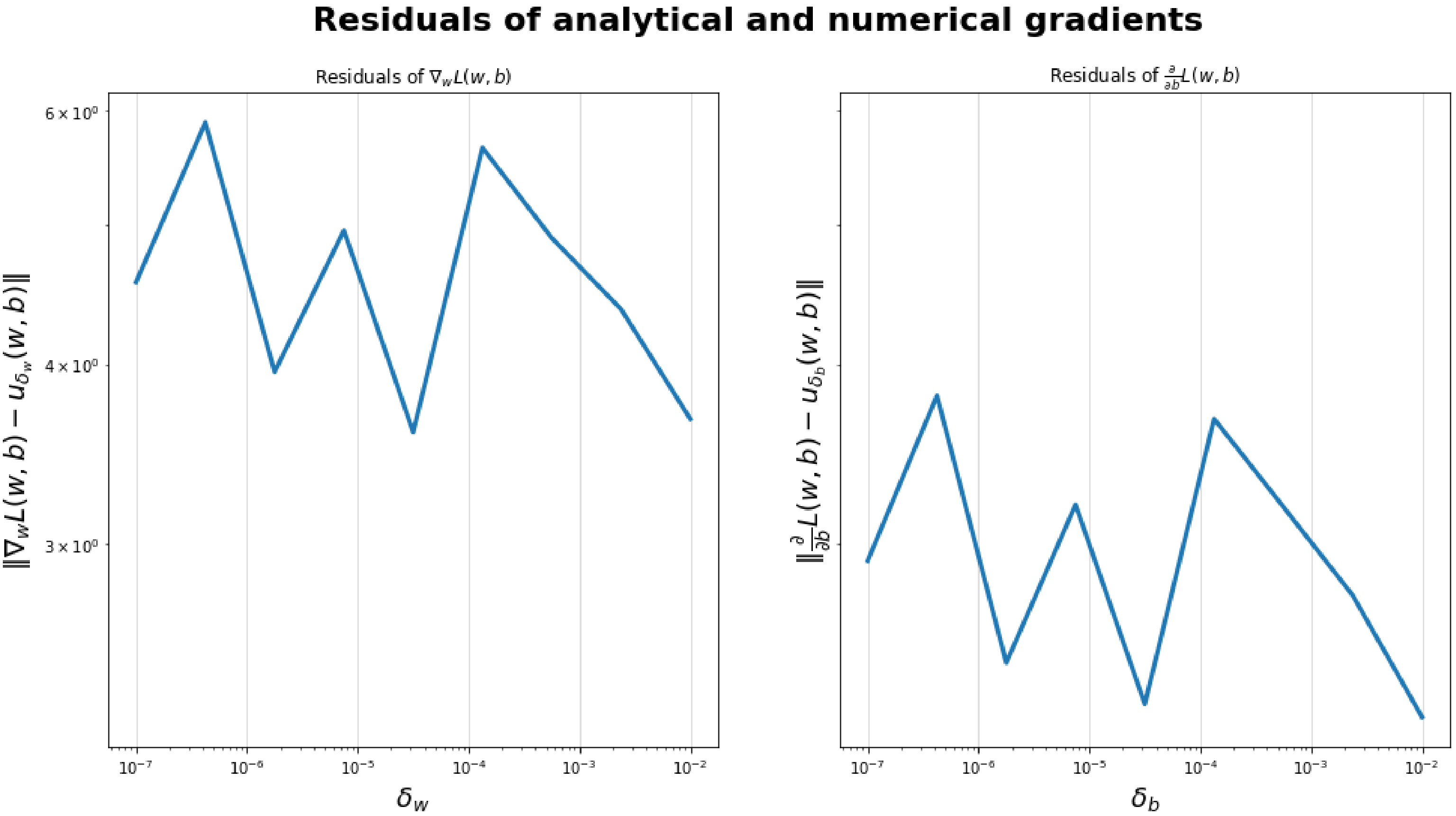
Section 2: Linear regression implementation

Q3.

$$\frac{1}{m}\left|\left|Xw + 1_b - y\right|\right|_2^2 \overset{c=Xw-y}{=} \frac{1}{m}\left|\left|c + 1_b\right|\right|_2^2 = \frac{1}{m}\Sigma(c_i + b)^2 \overset{\frac{\partial}{\partial b}}{\rightarrow} \frac{1}{m}\Sigma_i\, 2(c_i + b) = \frac{1}{m}\Sigma_i(2c_i) +$$

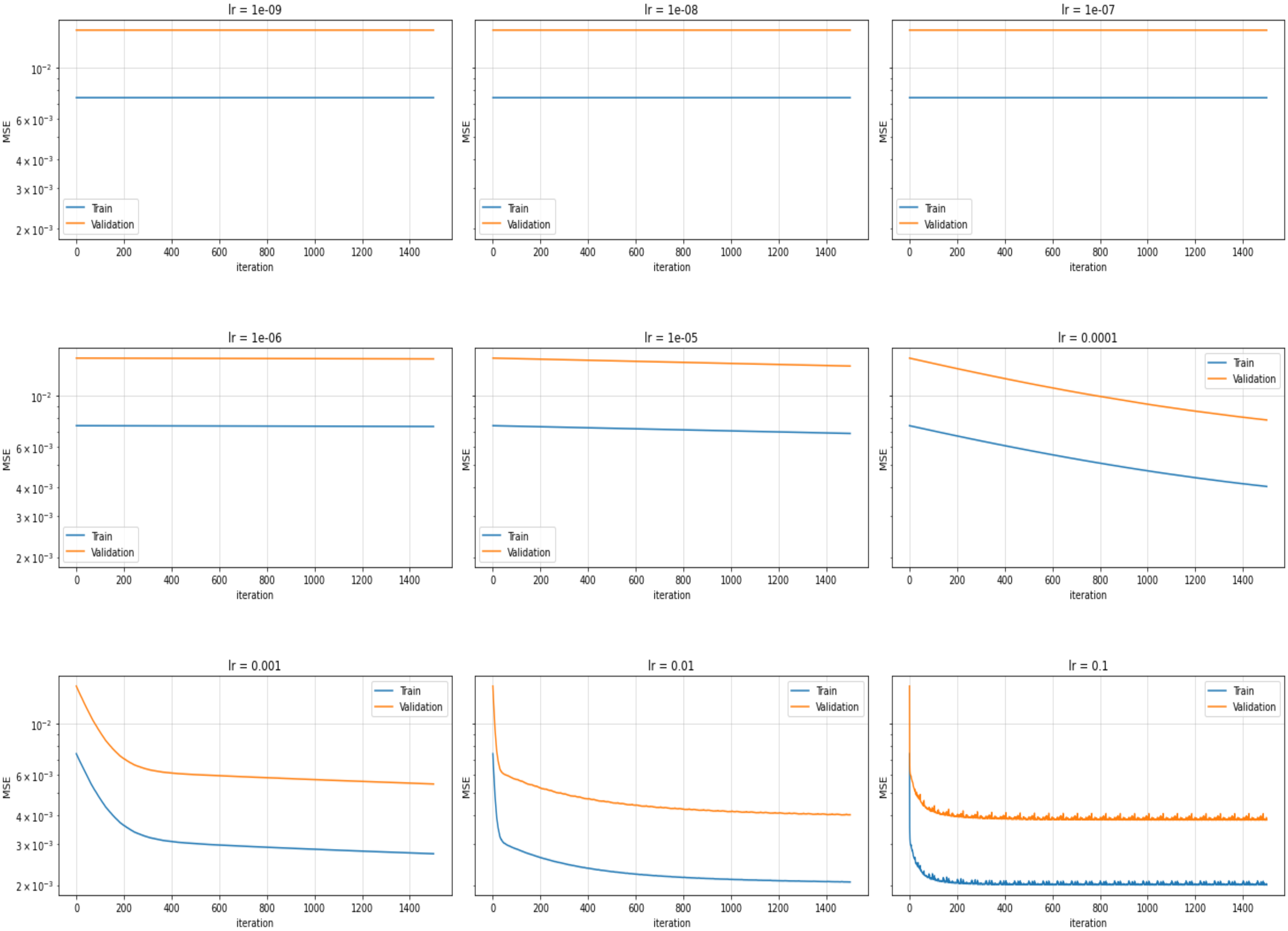
$$\frac{1}{m}\Sigma_i(2b) = \frac{2}{m}\Sigma(c_i) + 2 * m * \frac{b}{m} \overset{\Sigma_i c_i=(Xw-y)*(1_b)}{=} 2b + \frac{2}{\mathrm{m}}\ (\mathrm{Xw} - \mathrm{y}) *\underline{1}$$

Q4.



As we can see in the graphs both graphs are almost identical, only differ by a shift up.
This is happening because the numeric computation has its numeric limitations (accuracy) with no regard to the gradient itself.
We found that there is no specific lambda to minimize (we ran multiple times and got a minimum for different lambdas).

Q5.



As we can see in graphs for learning rates smaller than 1e-05 does not allow any learning procedure, since the steps are so small we barely progress in the negative direction of the gradient each step (since the step size is determined by the learning rate) so that's make sense because small learning rates cause small change in the weights during the learning, that leads to the fact the weights does not change enough (for the specified amount of iterations) resulting an algorithm that cannot learn (or learn too slow)...
For lr = 0.1 we can see that we do mange to learn but we got some jumps in the accuracies, This may be explained due to the fact that large learning rate causes large steps, too large steps may cause to miss the minimum point, and thus is what we see in the graph.
For lr = 0.0001-0.001 , we can see that we are able to learn but there is not enough iteration to converge to the minimum point (we can see that we constantly get better yet we finish too early), yet we can see that lr = 0.01 provided us with the best result the algorithm is able to converge to minimum with a smooth line, moreover we can see that it converged greatly in

the first 100 iteration and from this point and on we progress constantly but slowly to a better solution.

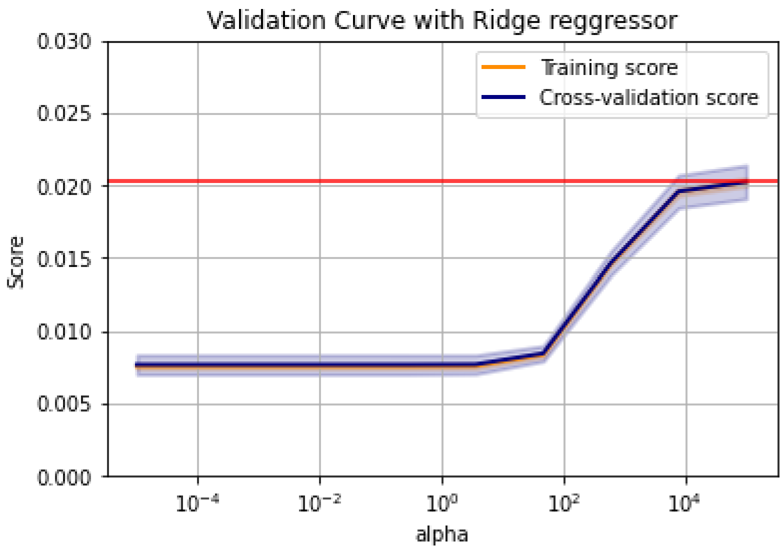
Section 3: Evaluation and Baseline

Q6.

Model	Section	Train MSE	valid MSE
		Cross validated	
Dummy	2	0.020272575509982637	0.020283053049045142

Section 4: Ridge linear regression

Q7.



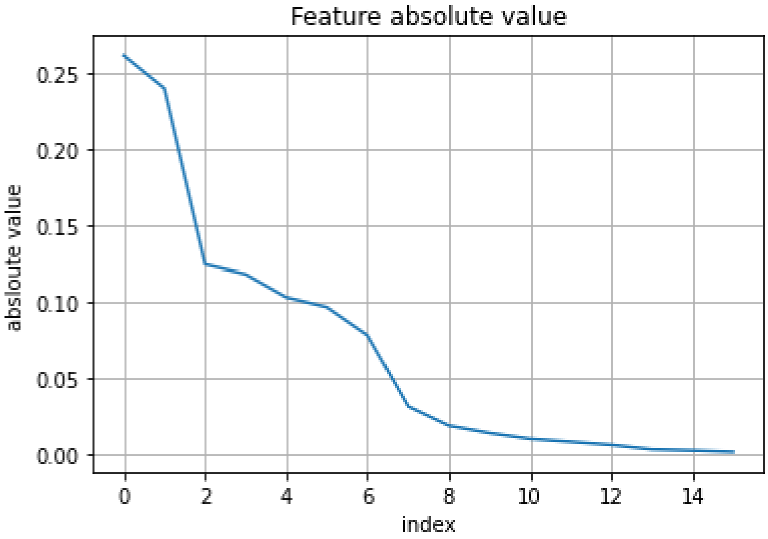
we found out that the best alpha is 0.1with a score of 0.00764982.

Q8.

Model	Section	Train MSE	valid MSE
		Cross validated	
Dummy	3	0.020272575509982637	0.020283053049045142
Ridge linear	4	0.0075355590909086875	0.0076498246770981115

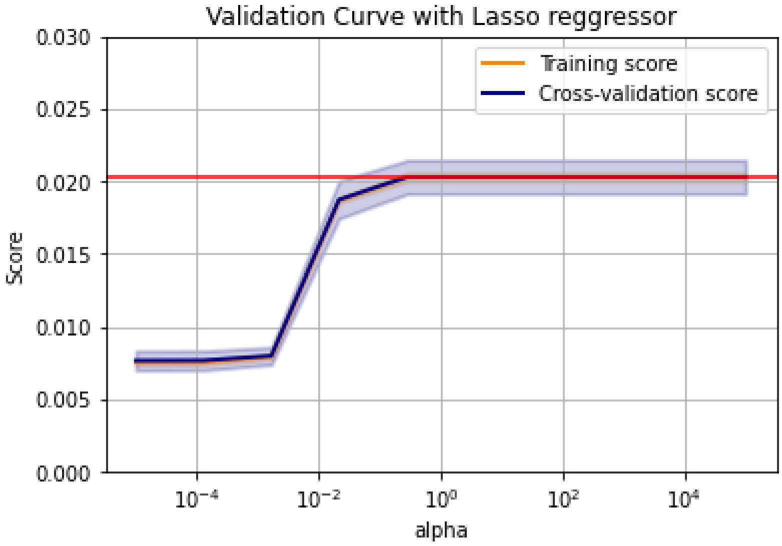
Q9.
symptoms_shortness_of_breath, num_of_siblings, PCR_07, symptoms_fever and PCR_08.

Q10.



Section 5: Lasso linear regression

Q11.



validation error = 0.007650380152048541 ,training error = 0.007535664534436627, best alpha = 1*10^-4

Q12.

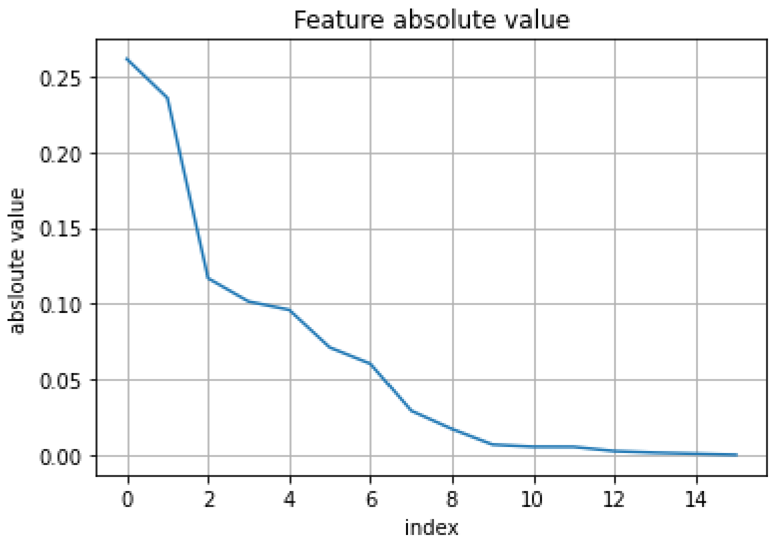
The graphs are similar, yet they're not the same. Lasso regression Fails to learn apporptaily for alpha bigger than 0.1, while for the Ridge regression alpha = 0.1 result with the best model as we saw previously. Still with the wright alpha , both of the regression algorithms succeeded to learn and provide the same mse error: 0.007535.

Q13.

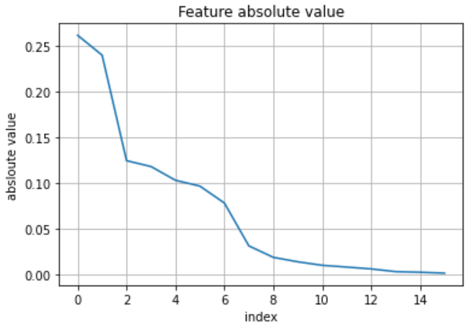
Model	Section	Train MSE	valid MSE
		Cross validated	
Dummy	3	0.020272575509982637	0.020283053049045142
Ridge linear	4	0.007535559090908687	0.007649824677098111
Lasso linear	5	0.007535664534436627	0.007650380152048541

Q14.
symptoms_shortness_of_breath, num_of_siblings, symptoms_fever,
PCR_08 and blood_type_A

Q15.



Q10 recall - Ridge regression:



Ridge: symptoms_shortness_of_breath, num_of_siblings, PCR_07, symptoms_fever and PCR_08.
Lasso: symptoms_shortness_of_breath, num_of_siblings, symptoms_fever,
PCR_08 and blood_type_A.

Let's compare the results for the Ridge(right) and Lasso(left) regression feature's absolute value- we can see that for the highest first two features we get the exact same results, but then we can see a slight change, it comes from the fact that PCR_07 which was the third highest absolute value feature for the Ridge regression disappear from the top 5 features for the Lasso regression. Instead of PCR_07 (which does not appear in the top 5) we get blood_type_A as the fifth value for the Lasso regression.

Section 6: Polynomial fitting

Q16.

When using the poly class, the algorithm regressor can now add coefficients to features like PCR_01XPCR_02. Already in HW1 we saw that these two features together can predict the spread feature with great precision. Linear regressor tries to find the best linear approximation, that is why without the polynomial extension we couldn't use features like PCR_01XPCR_02 before.

We believe that by letting our model use second degree polynomials we are increasing it's complexity so we expect to get better results that will be interpreted as better scores and lower errors. We also think that second degrees are low enough polynoms to avoid overfitting that data.

Q17.

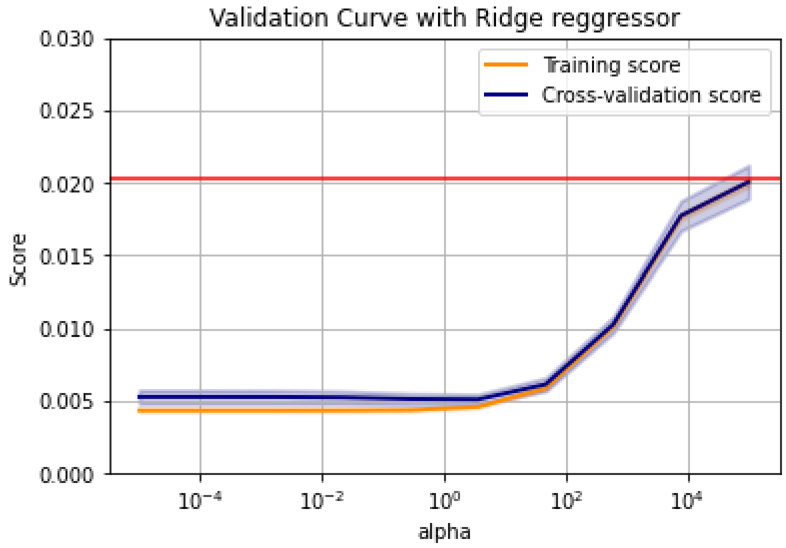
$$\text{We can also write } h_{multi}(x) \; as \; C(w_1^t x + b_1) + (1 - C)(w_2^t x + b_2)$$

$$s.t \; C = \{0{,}1\} \; according \; to \; satisfying \; the \; condtion$$

$$= C(w_1^t x) + C(b_1) + (1 - C)(w_2^t x) + (1 - C)(b_2) = C(w_1^t + 1 - w_2^t)x + C(b_1 + 1 - b_2) \in H_{poly}$$

$$as \; we \; can \; see \; this \; is \; a \; linear \; equation \; therefore \; it \; is \; in \; H_{poly}$$

Q18.



validation error = 0.00507879739710952 , training error = 0.004323837562342335, best alpha = 3.

Q19.

Model	Section	Train MSE	valid MSE
		Cross validated	
…	…	…	…
Ridge polynomial	6	0.00432383756234233	0.00507879739710952

Section 7: Testing your models

Q20.

Model	Section	Test MSE
Dummy	3	0.020367501736111116
Ridge linear	4	0.009105444088794871
Lasso linear	5	0.00877587786126076
Ridge polynomial	6	0.008013763477338776