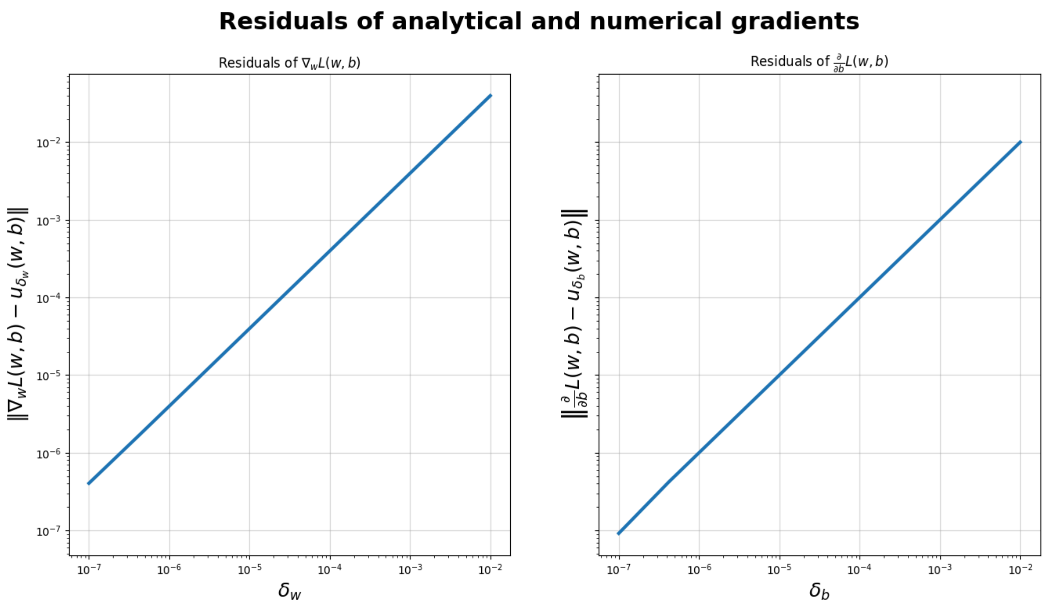
HW3 Regression

Section 1: Linear regression implementation

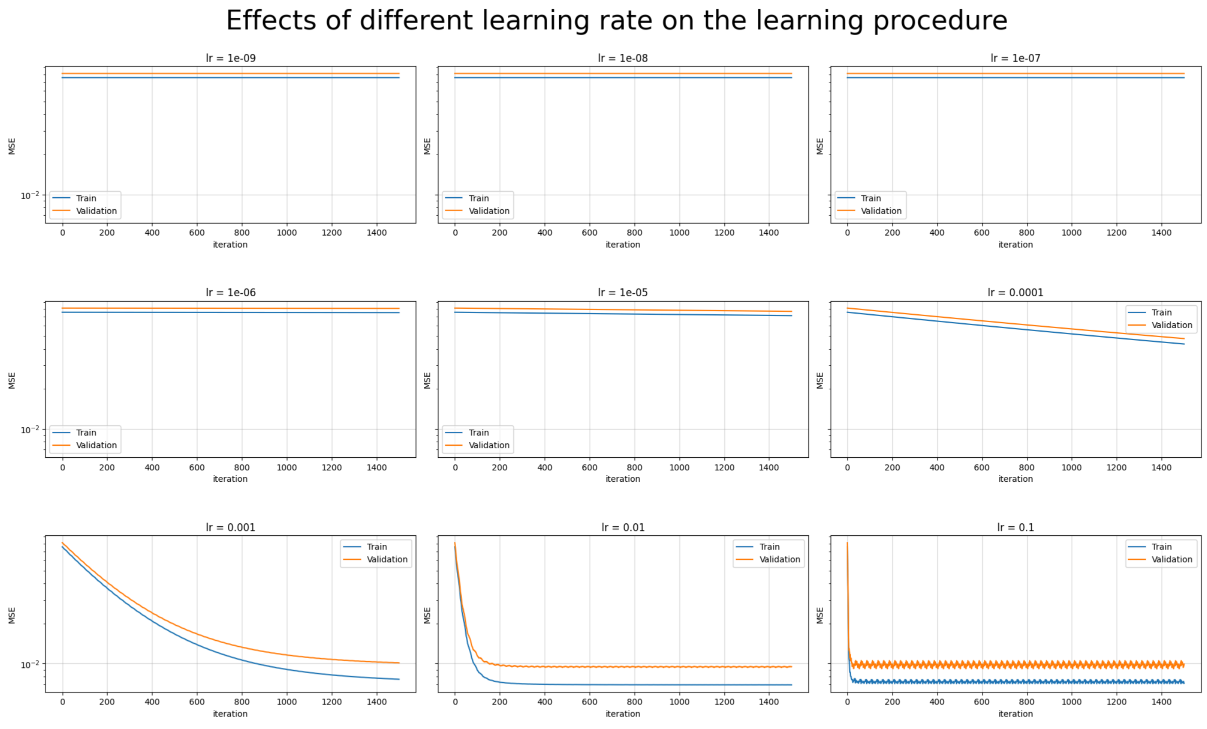
(Q1) Derivation of the analytical partial derivative $\frac{\partial}{\partial b}L(\underline{w}, b) \in R$:

$$\frac{\partial}{\partial b}L(\underline{w}, b) = \frac{1}{m}2\left(\underline{1}_m\right)^T(X\underline{w} + \underline{1}_m \cdot b - \underline{y}) = \frac{\partial}{\partial b}L(\underline{w}, b) = \frac{1}{m}2(X\underline{w} + \underline{1}_m \cdot b - \underline{y})$$

(Q2)



(Q3)



We can see in the lower (small) learning rates that the graphs are straight / plateau, in the graph for learning rate $1e^{-5}$ we can see the start of a very slow convergence. From that, we can assume that there is a need of numerous more iterations to get to the best learning rate.

For the higher learning rate, we can see that there are “jumps” between values around the minimum. This behavior indicates that the learning rate is too high and misses the minimum.

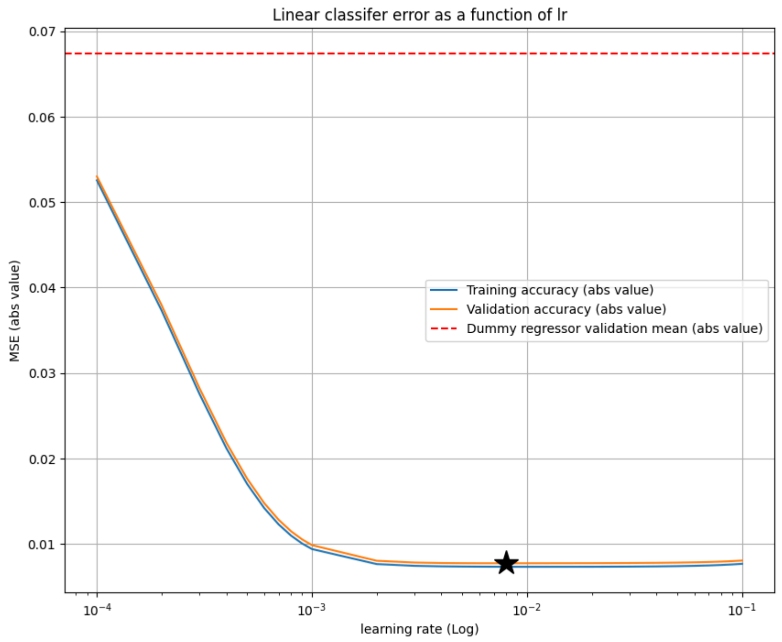
Based on the graphs above we believe that **the optimal learning rate is 0.01** it reaches a stable minimum very quickly, after around 150 iterations. It doesn’t make sense to increase the number of gradient steps – the model is already converging at around 100 steps and increasing the number of steps will not significantly improve the results.

Section 2: Evaluation and Baseline

(Q4)

Model	Section	Train MSE	Valid MSE
		Cross validated	
Dummy	2	0.066927	0.067375

(Q5)



Model	Section	Train MSE	Valid MSE
		Cross validated	
Dummy	2	0.066927	0.067375
Linear	2	0.007322	0.007732

The best learning rate: 0.008
The train score: 0.007322869218734882
The validation score:0.007732264401351307

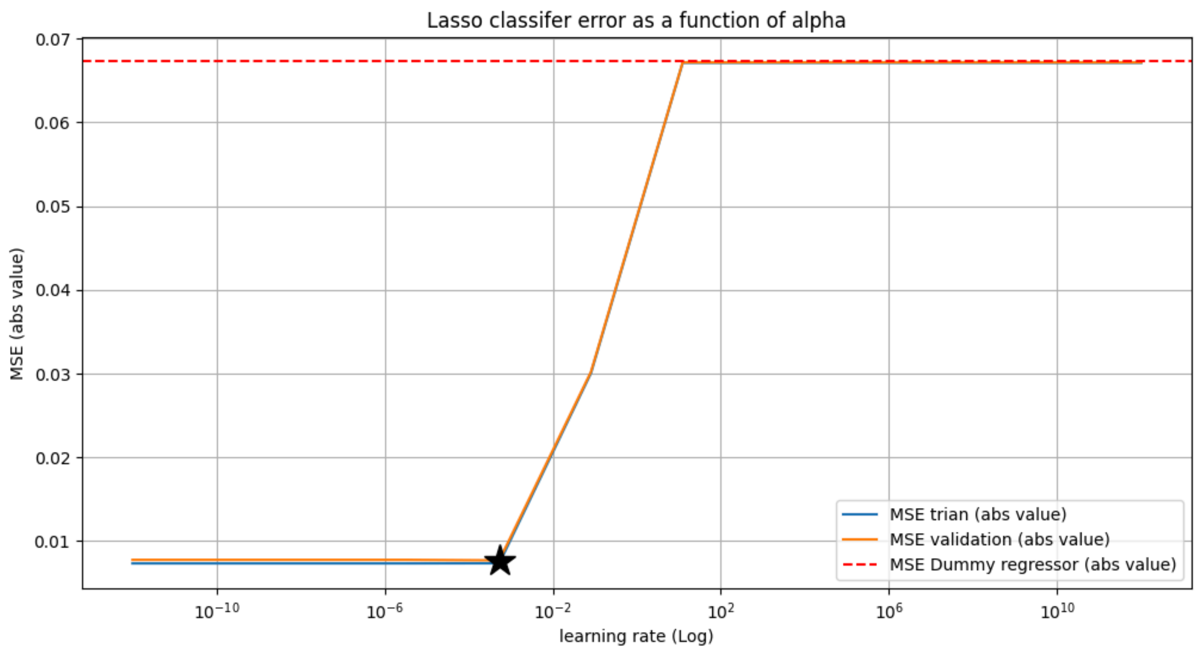
(Q6)

For the Linear Regressor, the normalization is crucial. If we hadn't normalized the features beforehand, the training performance for the linear model might have changed, and perhaps not converged. The reason is we're using a gradient-based training procedure, which is affected by the different scales of the feature. Meaning, that a feature with larger values will be more dominant in the model prediction since its coefficient is dominantly larger than the rest, thus leading to an unbalanced behavior of the model. By normalizing the data, we enable "smoother" convergence in the loss of the model (fewer oscillations on the way to the local minimum), thus improving the training process.

For the dummy model, the results will not change whether we normalize or not. The reason for this phenomenon is that the model uses only the target feature (y values) mean for prediction, hence, the normalization of features doesn't have any effect on the model.

Section 3: Lasso Linear Regression

(Q7)



The best alpha value: 0.0005336699231206301
The training score: 0.007321486823643697
The validation score: 0.007675139292402417

(Q8)

Model	Section	Train MSE	Valid MSE
		Cross validated	
Dummy	2	0.066927	0.067375

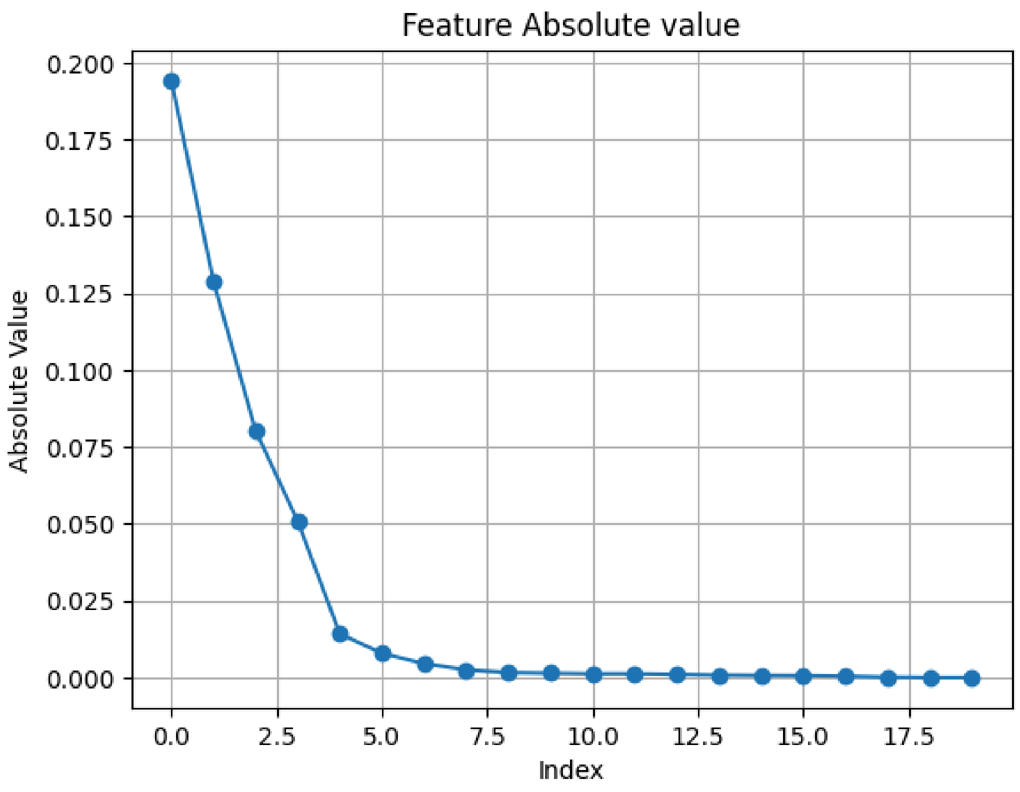
Linear	2	0.007322	0.007732
Lasso-Linear	3	0.007321	0.007651

(Q9)

The 5 features with the 5 largest coefficients (in absolute value) are (in descending order):

```
PCR_04:  0.194193
sugar_levels:  0.128961
PCR_02: -0.080314
PCR_06: -0.050900
age: -0.014185
```

(Q10)



(Q11)

The magnitude of the coefficients is interesting because it reflects the "**importance**" of each feature to the regression (a **relationship between a feature and the target variable**). The higher the coefficient value is, the more the feature is "important" to the model prediction. Accordingly, features with a lower coefficient value are less important, and those with a coefficient value of zero do not impact the model at all in the prediction.
We can see that there is a **major decline** in the coefficients, where besides the first 4 (or 5) features, the rest barely (to not at all) affect the model.

(Q12)

The training performance of the Lasso model would probably change without the feature normalization in pretraining. Earlier, we explained how the training process is affected by the

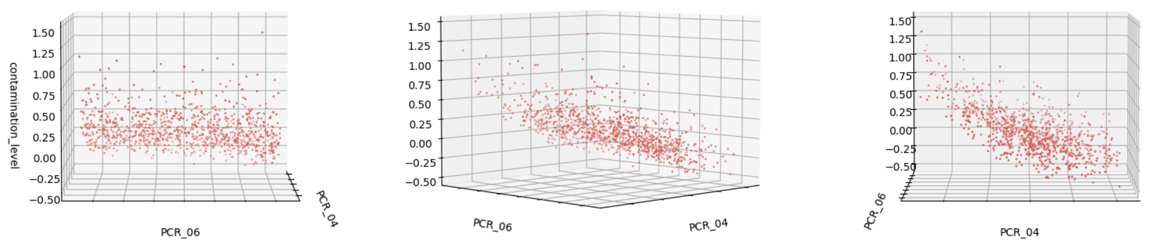
different scales and magnitudes of the features. This effect might be even bigger with Lasso models, as they use L1 regularization which produces sparser weights (reduces more coefficient value to zero for a subset of features).
If the features are not normalized, the model might reduce to zero the coefficient of very important features based on their scale, while emphasizing others that aren't important in the prediction. In other words, the normalization of the data helps us to maintain and ensure that each feature receives a weight in the regularization based on its importance (relation to the predicted value) rather than on its scale/magnitude..

(Q13)

As we learned in class, a lasso regularization produces a sparser model, which means more features are zero (or close to it). The Ridge model uses the L2 normalization and shrinks less coefficients to zero. **Meaning, that with a ridge regressor, we would expect to see less zero coefficients.**
[Hence, the Ridge regressor might be a more stable model but will not be as good as the Lasso model in emphasizing (picking) the most important features for prediction]

Section 4: Polynomial fitting (visualization)

contamination as a function of PCR_04 and PCR_06

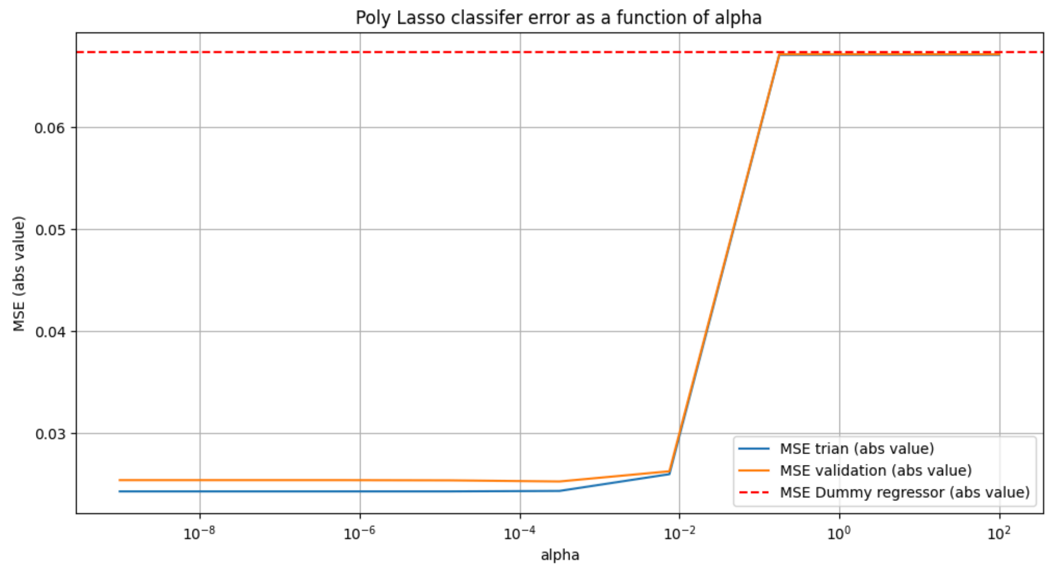


(Q14)

After applying a polynomial mapping, re-normalization is important for two reasons:
1. Different powers create **different scales** for the features (even power:[0,1], odd power:[-1,1]).
2. Looking at the absolute value of the new features - raising to higher powers create **features with more "extreme" magnitude** (close to 0 or 1).

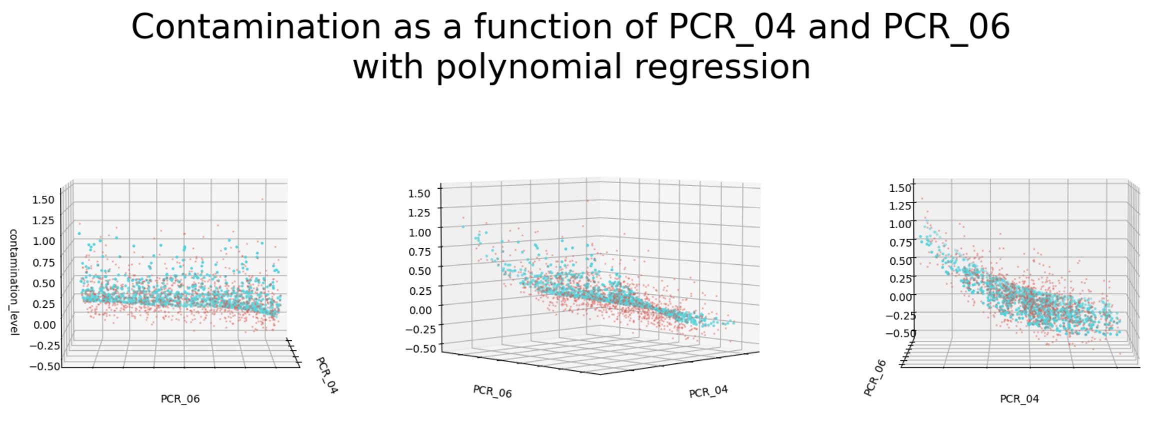
From an optimization point of view, we use SGD to train the model. As described earlier, SGD is sensitive to the differences in the magnitude and the scale of the features. This means the major differences can slow down the convergence rate (and depending on the iterations number even prevent it), and lead to higher error.
(Additional possible effects include practical and statistical aspects such as numerical instability of the model).

(Q15)



The optimal alpha value: 0.00031622776601683794
The optimal training error: 0.024339720701781482
The optimal validation error: 0.02526782046802236

(Q16)

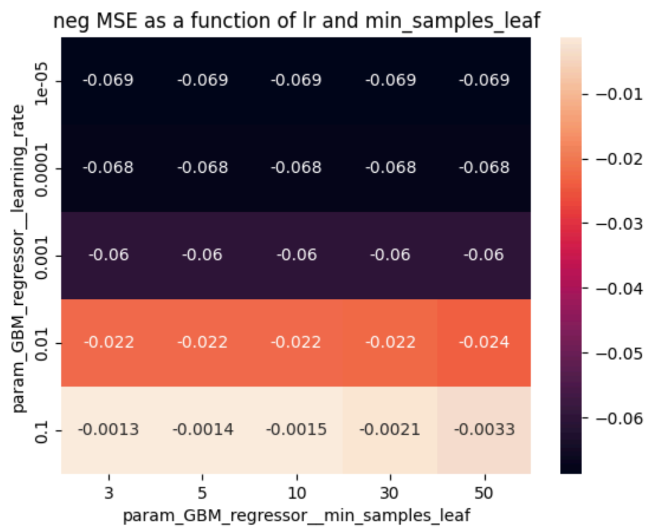


(Q17)

Model	Section	Train MSE	Valid MSE
		Cross validated	
Dummy	2	0.066927	0.067375
Linear	2	0.007322	0.007732
Lasso-Linear	3	0.007321	0.007651
Polynomial lasso	4	0.024339	0.025267

Section 5: Fitting Gradient Boosted Machines (GBM) of the CovidScore

(Q18)



The best parameters are {'GBM_regressor__learning_rate': 0.1, GBM_regressor__min_samples_leaf': 3}

The best score is -0.0013414602212152352

(Q19)

Model	Section	Train MSE	Valid MSE
		Cross validated	
Dummy	2	0.066927	0.067375
Linear	2	0.007322	0.007732
Lasso-Linear	3	0.007321	0.007651
Polynomial lasso	4	0.024339	0.025267
GBM	5	0.000294	0.001341

Section 6: Testing your models

(Q20)

Model	Section	Train MSE	Valid MSE	Test MSE
		Cross validated		Retained
Dummy	2	0.066927	0.067375	0.072493
Linear	2	0.007322	0.007732	0.007632
Lasso-Linear	3	0.007321	0.007651	0.007599

Polynomial lasso	4	0.024339	0.025267	0.028232
GBM	5	0.000294	0.001341	0.001396

Based on the table above we can see that the model with the best performances on the test set is the GBM. It is also the model that performed best on the train and validation set.

From the similar values of train MSE and validation MSE per model and the test MSE (excluding Dummy), we can say that all models performed well with probably very small overfitting on the training set. We can also see that for the Linear and Lasso-Linear models, the test is better than the validation. These findings indicate that the features we chose to predict the contamination levels fitted the task well.

We can also see in the table above that the Ploynomial-Lasso performed the worst out of the models (excluding the Dummy model). A possible explanation for this is the fact that we used only 2 features ‘PCR_04’ and ‘PCR_06’ to predict the contaminations labels. It is possible that these 2 features are not sufficient to predict the contamination labels by themselves and we need the contribution of more features for better predictions.