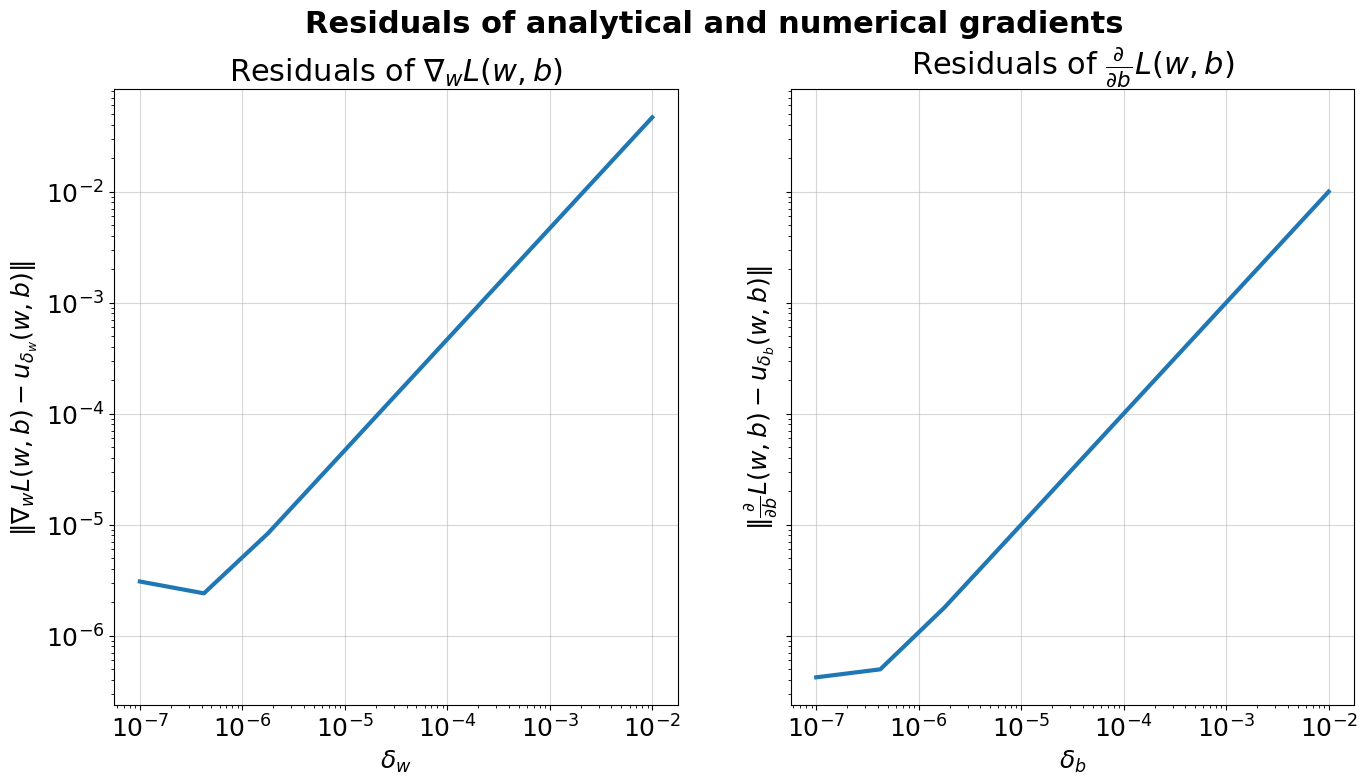
Major HW 3 – Regression

Orad Barel, 311288203, [oradbarel@campus.technion.ac.il](mailto:oradbarel@campus.technion.ac.il)

Ofir Manor, 316084623, [ofir.manor@campus.technion.ac.il](mailto:ofir.manor@campus.technion.ac.il)

**Q1.**

**Q2.**

****

Diagram

Description automatically generated**Q3.**

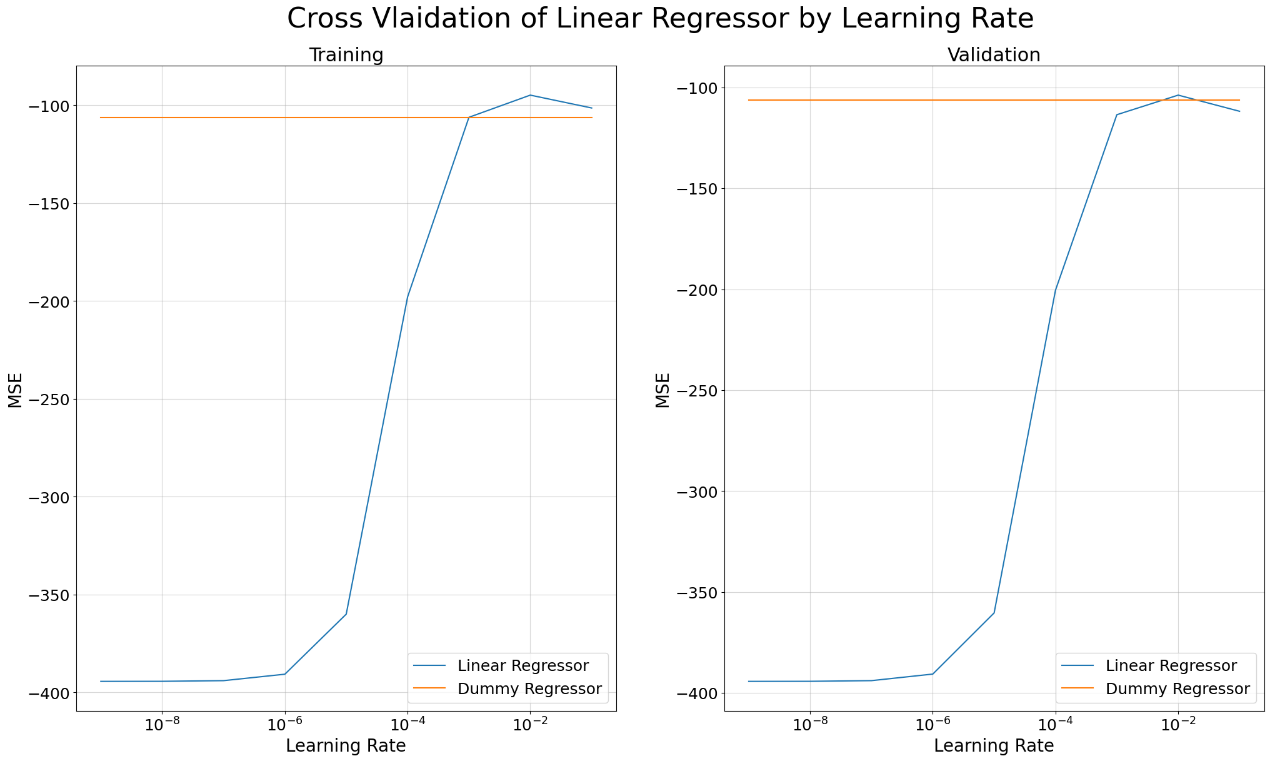
We see here that from the most part (excluding ), the higher the learning rate, the faster the descent of the loss is. This makes sense because small learning rates create a slow gradient descent, so that we arrive at the minimum of the loss only after a great number of iterations. As for we can see that the loss is divergent after a few iterations, the large “jumps” that occur do not allow the gradient descent to arrive at a minimal loss.

Our best learning rate (the one that achieved the minimal validation loss) is . It does not make sense to increase the number of gradient steps as it seems to achieve this minimum before step 1500 and slightly diverge around the minimum.

**Q4.**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Section | Train MSE | Valid MSE |
|  |  | Cross Validated | |
| Dummy | 2 | -105.82 | -106.19 |

**Q5.**



|  |  |  |  |
| --- | --- | --- | --- |
| Model | Section | Train MSE | Valid MSE |
|  |  | Cross Validated | |
| Dummy | 2 | -105.82 | -106.19 |
| Linear | 2 | -94.77 | -103.8 |

**Q6.**

Depends on the model. The dummy model, that always uses the average contamination level, would not change. The average stays the same regardless of any normalization we did to the features.

The Linear Regressor will change. This model uses SGD in order to find the minimal loss. SGD is sensitive to the values of the features. The partial derivative of a feature which holds “larger” (as in far greater or far lesser) data points than other features will pull the SGD stronger in its direction, thus making it more prominent, even if it’s data is less indicative of our target.

**Q7.**

Chart, line chart

Description automatically generated

We found that the optimal and it achieves a validation loss of -98.25

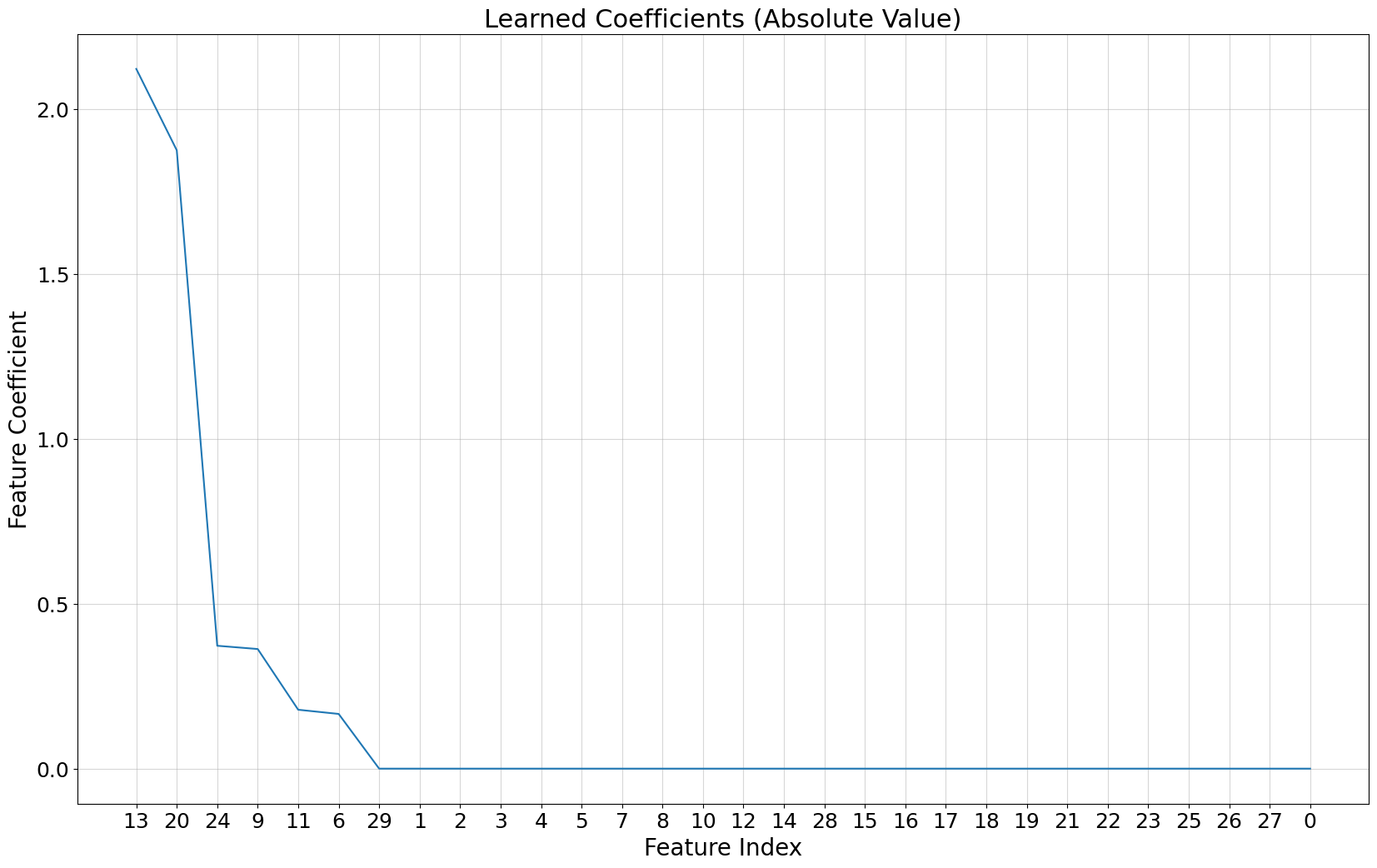
**Q8.**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Section | Train MSE | Valid MSE |
|  |  | Cross Validated | |
| Dummy | 2 | -105.82 | -106.19 |
| Linear | 2 | -94.77 | -103.8 |
| Lasso Linear | 3 | -96.25 | -98.25 |

**Q9.**

|  |  |
| --- | --- |
| Feature | Coefficient |
| sugar\_levels | 2.121 |
| PCR\_01 | 1.875 |
| PCR\_05 | 0.372 |
| num\_of\_siblings | 0.362 |
| household\_income | 0.178 |

**Q10.**

****

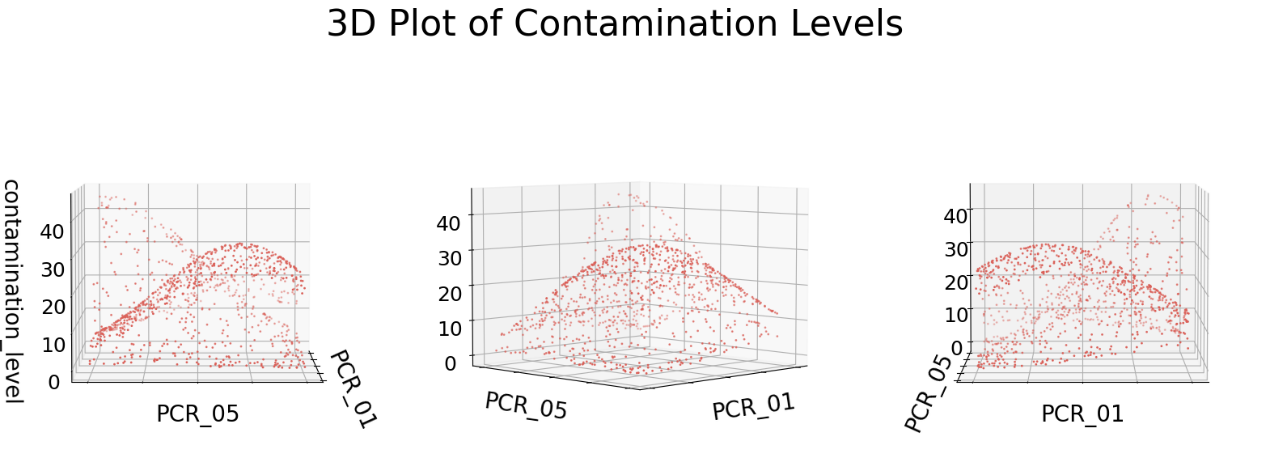
**Q11.**

As we saw in the tutorial, lasso (which uses regularization) causes a sort of “variable selection”. The magnitude of the coefficient gives us an idea of the effect the feature has on finding our target, the features with larger (absolute) coefficient will be more prominent in our model for predicting the target.

**Q12.**

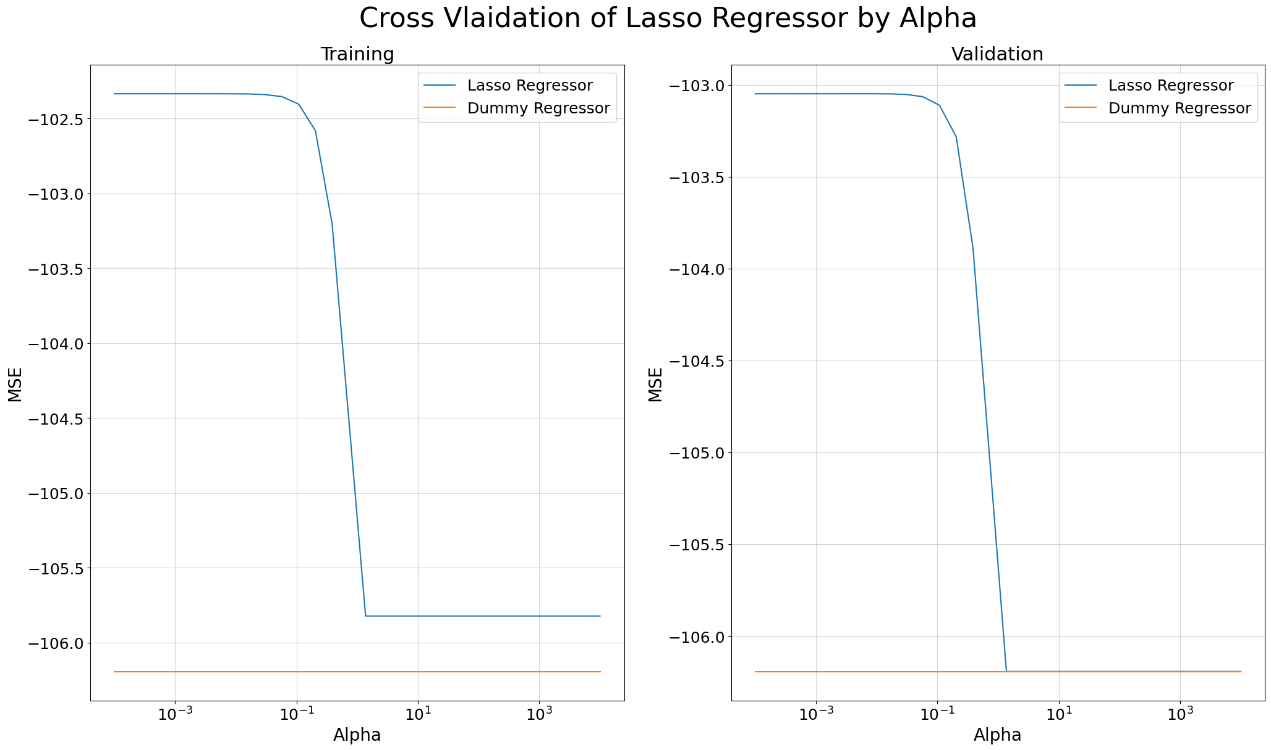
Yes, had we chosen not to normalize the features the training performance of Lasso would have changed. Features who in their original scale is smaller from the others would have far greater regularization in order to compensate. This would mean that for said features the regularization has greater importance than the data, while for others their regularization would remain minimal. This changes the training performance of the model.

**Q13.**



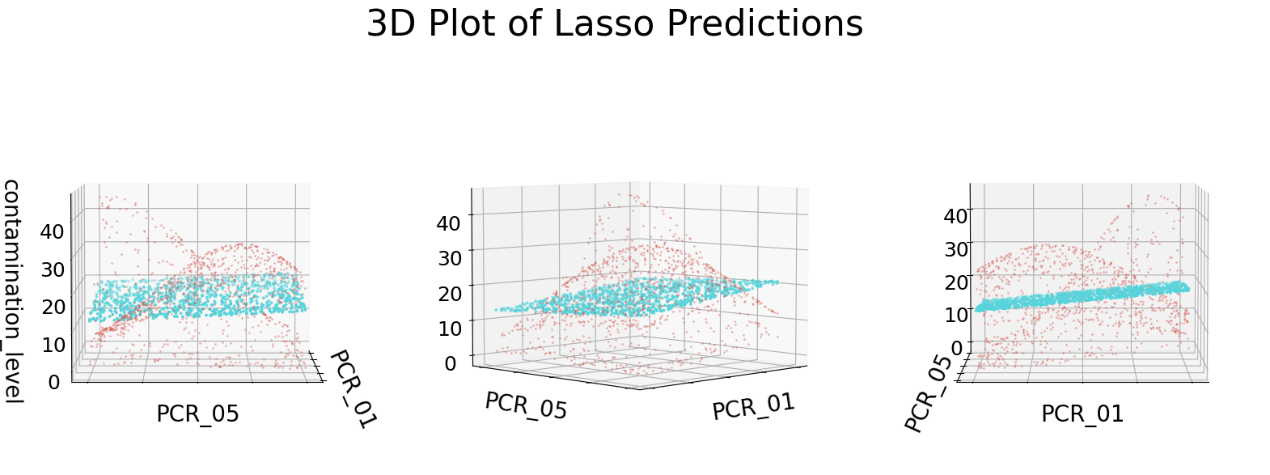
From this visualization, we can understand that the contamination levels are distributed in a mostly of parabolic manner through PCR\_01 and PCR\_05, but with a large amount of noise. When we look at the bivariate analysis of them, we see that still a certain parabolic nature returns, where the large majority of contamination levels can be found below some parabolic line, which only a few datapoints which seem “noisy” not under said “line”. This means that we would need some model that can handle the noise, one with some regularization (which in this assignment is Lasso).

**Q14.**



Using only PCR\_01 and PCR\_05 as features we found that the optimal and it achieves a validation loss of -103.04, though this is slightly misleading, as further experimentation showed that all smaller values of also achieve the same validation loss.

**Q15.**

****

**Q16.**

After 2nd degree polynomial mapping, many of our “new” features are of a second degree () and seeing as how these features have already been normalized earlier, most of them lie between 1 and -1. Thus, after mapping them, all those that did not equal 1 or -1 will now lie much closer to 0. This will create a non-evenly distributed scattering of datapoint as there will be small patched at 1 and -1, and a larger patch near 0. This new distribution could harm the learning process as it creates artificial “islands” that could falsely give more information.

**Q17.**

**Chart, line chart

Description automatically generated**

We found that the optimal and it achieves a validation loss of -91.