Major HW 3 – Regression

Orad Barel, 311288203, oradbarel@campus.technion.ac.il

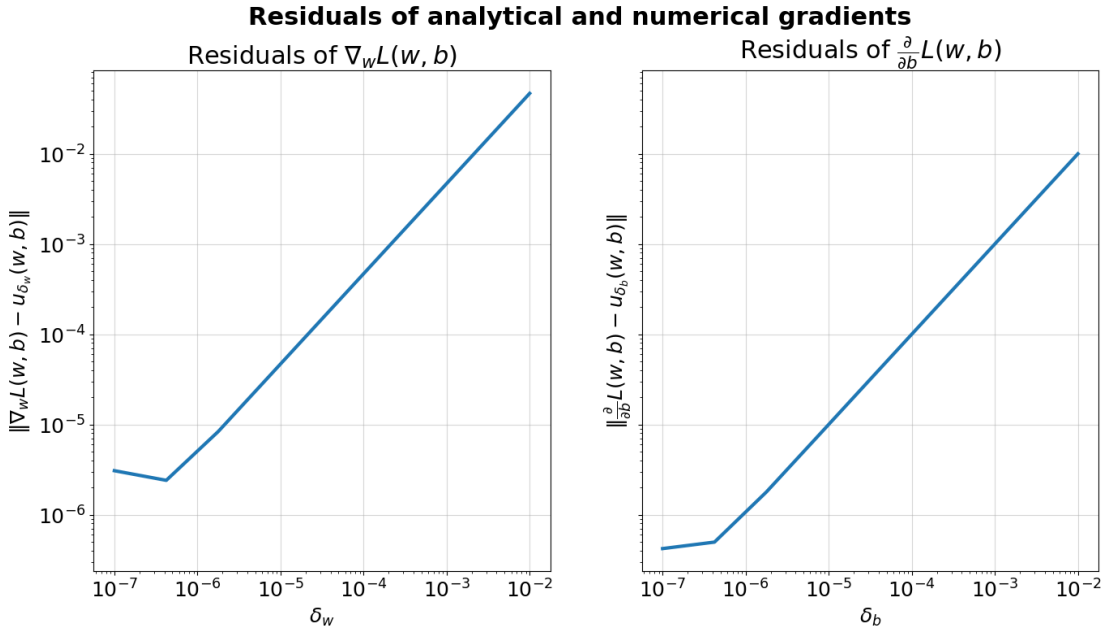
Ofir Manor, 316084623, ofir.manor@campus.technion.ac.il

Q1.

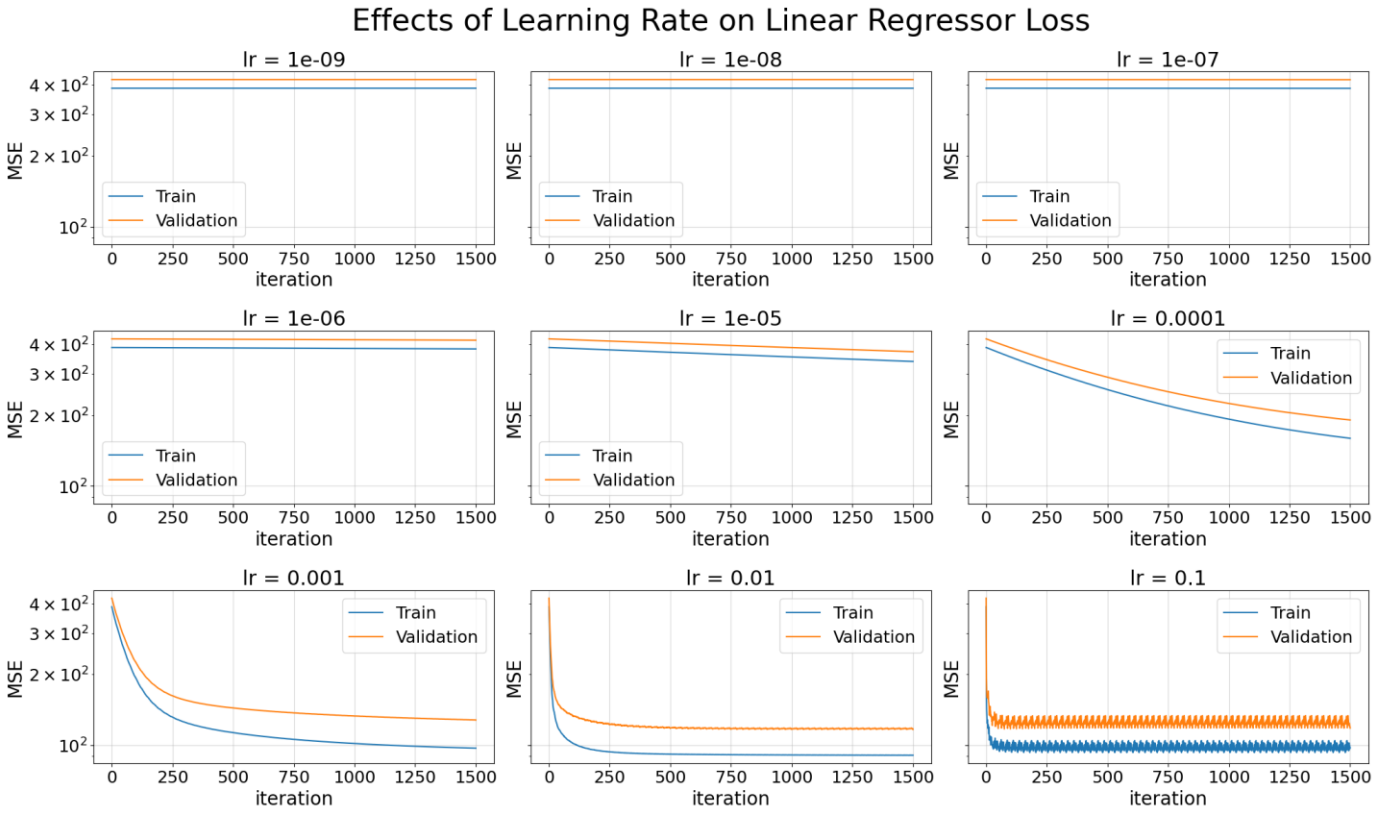
$$\frac{\partial}{\partial b}L(\underline{\boldsymbol{w}},b)=\frac{\partial}{\partial b}\frac{1}{m}\left\|X\underline{\boldsymbol{w}}+\underline{\mathbf{1}}_m\cdot b-\underline{\boldsymbol{y}}\right\|_2^2 \underset{(f^2)'=2f'f}{=}$$

$$\frac{2}{m}\underline{\mathbf{1}}_m^{\boldsymbol{T}}\left(X\underline{\boldsymbol{w}}+\underline{\mathbf{1}}_m\cdot b-\underline{\boldsymbol{y}}\right)=\frac{2}{m}\sum_{i=1}^{m}(\underline{\boldsymbol{w}}^T\underline{\boldsymbol{x_i}}+b-y_i)$$

Q2.



Q3.



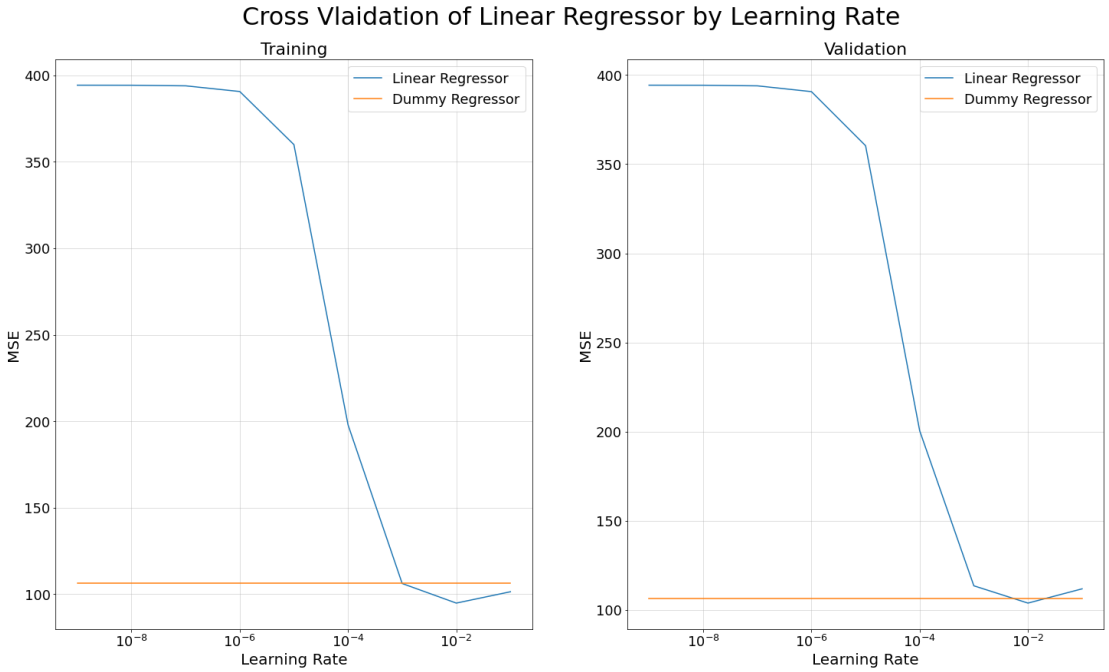
We see here that from the most part (excluding $lr = 0.1$), the higher the learning rate, the faster the descent of the loss is. This makes sense because small learning rates create a slow gradient descent, so that we arrive at the minimum of the loss only after a great number of iterations. As for $lr = 0.1$ does not converge, 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 $lr = 0.01$. It does not make sense to increase the number of gradient steps as it seems to achieve this minimum before step 1500.

Q4.

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

Q5.



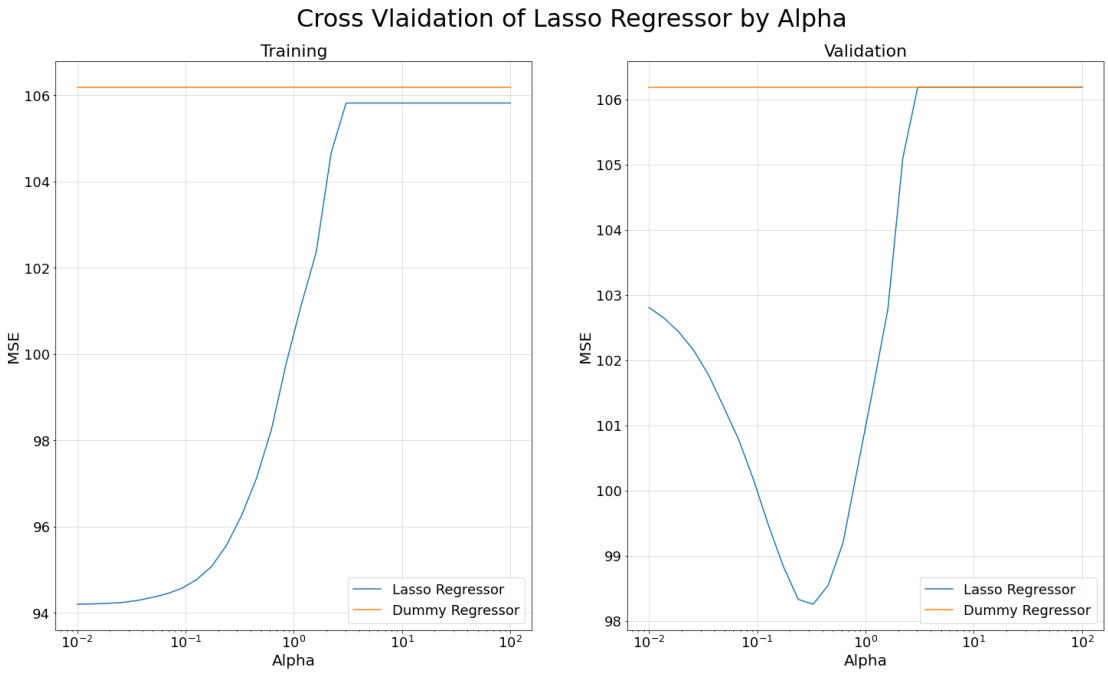
Model	Section	Train MSE	Valid MSE
		Cross Validated	
Dummy	2	105.82	106.19
Linear	2	94.77	103.8

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 would also not change. This model uses SGD in order to find the minimal loss. The direction of the SGD would not change as the gradient direction and the learning rate would remain the same. The size of the step would be affected by the normalization of the features, but so would the feature space so that the step would remain relative to the feature space. Hence, we would arrive at the same MSE.

Q7.



We found that the optimal $\alpha = 0.32$ and it achieves a validation loss of 98.25.

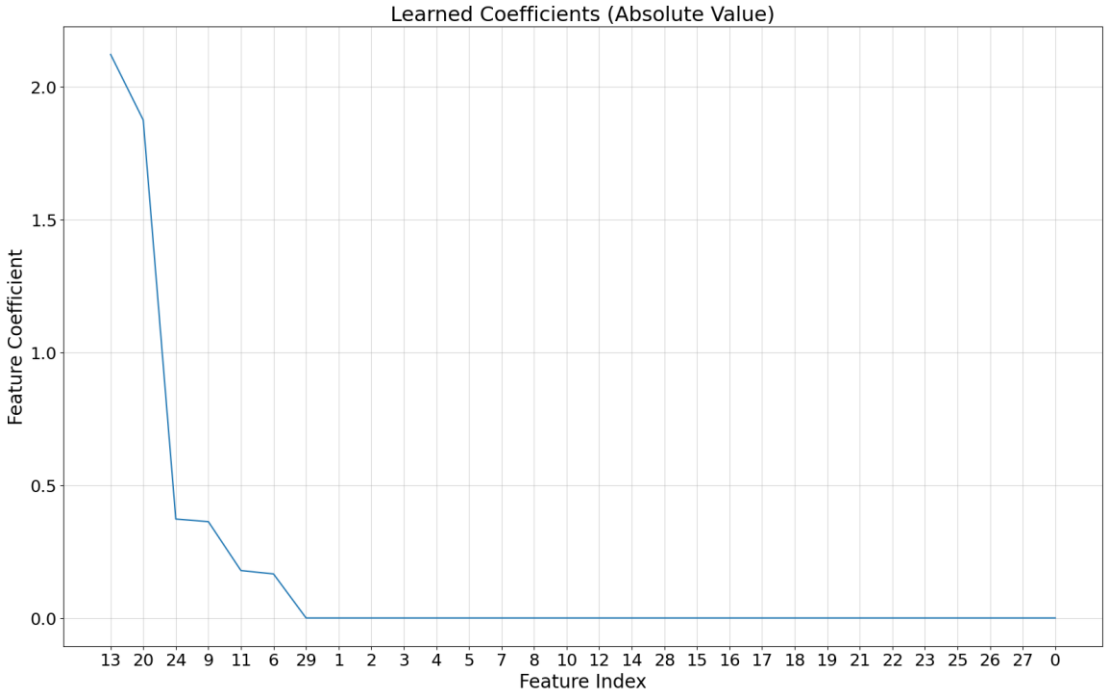
Q8.

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.



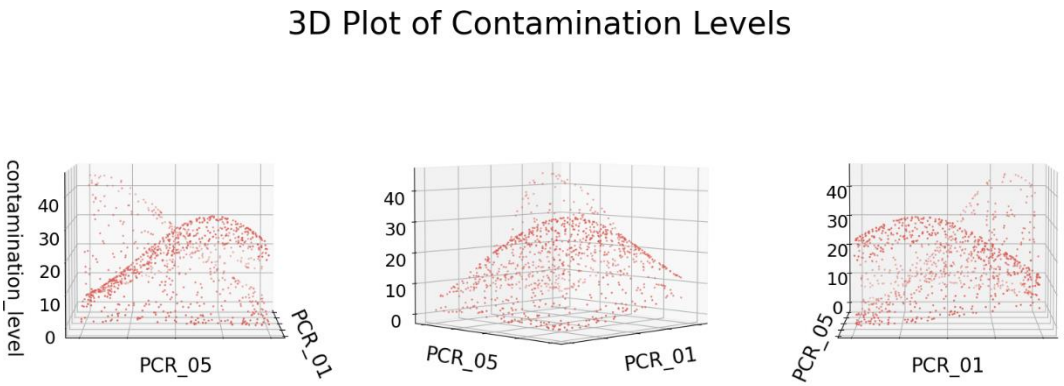
Q11.

As we saw in the tutorial, lasso (which uses ℓ_1 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.

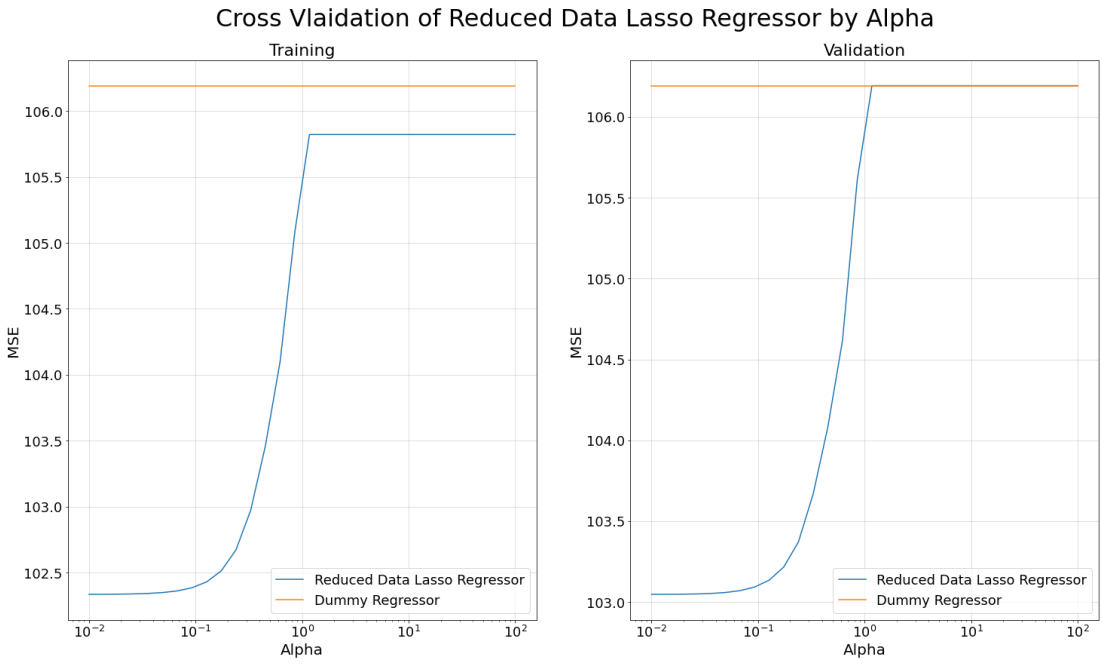
Yes, had we chosen not to normalize the features the training performance of Lasso would have changed. Features who in their original scale is smaller from the others would have far greater regularization in order to compensate. This would mean that for said features the regularization has greater importance 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 most contamination levels can be found on some parabolic line. This means our model should be able to hypothesize the target on a parabolic plane, unlike the "flat" planes our Linear regressor and Lasso modeled.

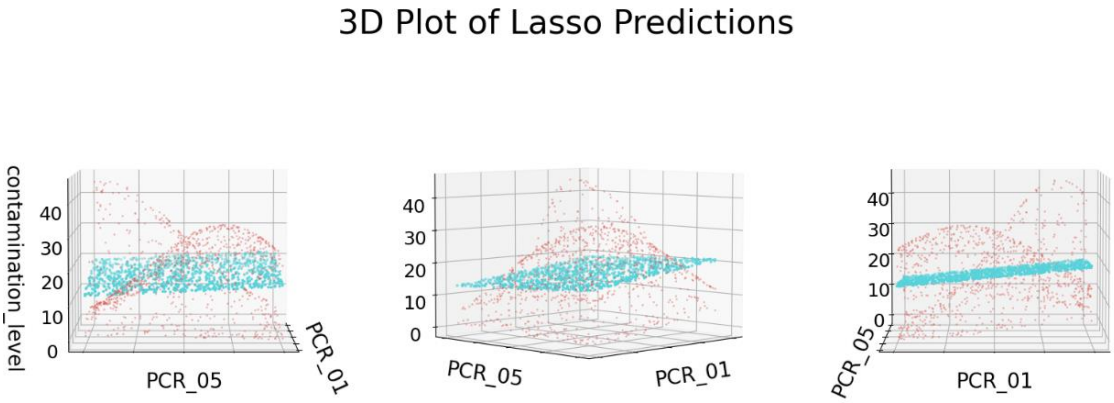
Q14.



Using only PCR_01 and PCR_05 as features we found that the optimal $\alpha = 0.01$.

It achieves a validation loss of 103.04.

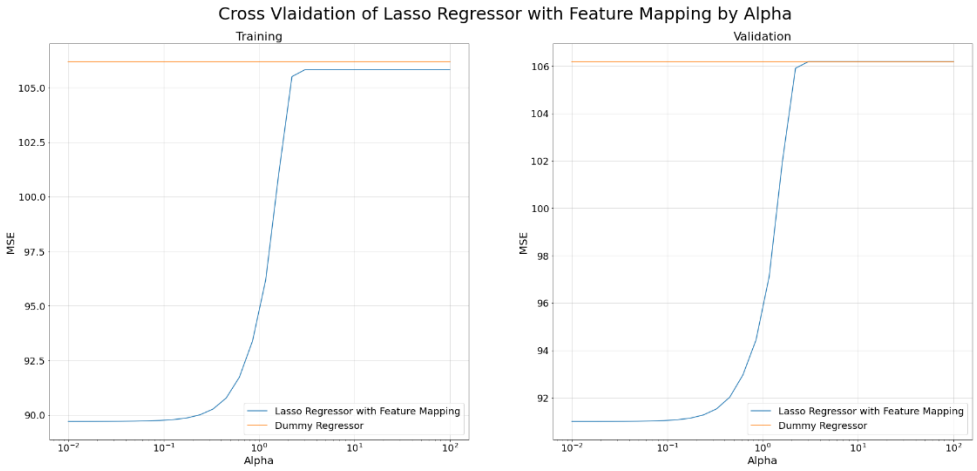
Q15.



Q16.

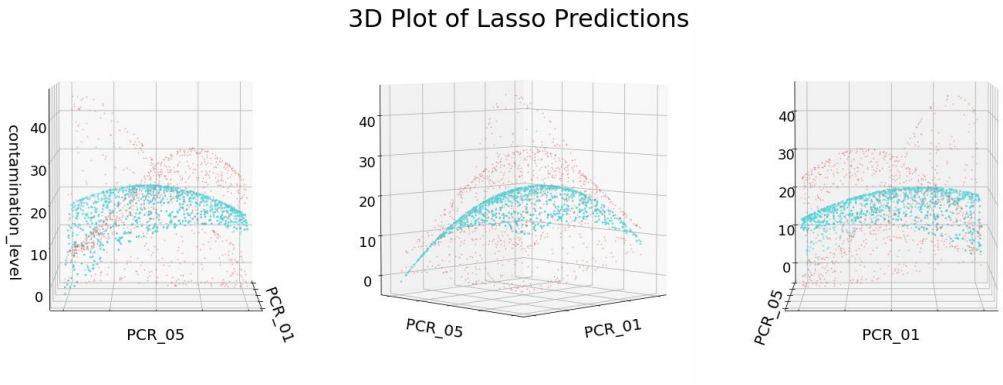
After 2nd degree polynomial mapping, many of our "new" features are of a second degree ($either\ x_i^2\ or\ x_i \cdot x_j$) and seeing as how these features have already been normalized earlier, most of them lie between 1 and -1. Thus, after mapping them, all those that did not equal 1 or -1 will now 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.



We found that the optimal $\alpha = 0.013$ and it achieves a validation loss of 91.

Q18.



19.

We have found that after feature mapping, out model was able to better hypothesize our target variable. We can see from the plot, that our predictions are placed on a parabolic plane, which (from a human eye perspective) is far closer to the actual distribution of the contamination level with respect to of PCR_01 and PCR_05. We have also seen that the error has reduced by over 10% after our feature mapping (while the regularization strength remained similar). All of this goes to show us that regularization is not enough to improve our model, but feature mapping adds necessary data to allow our models to predict with greater accuracy.

20.

For our Random Forest model, we decided to transform feature sugar_levels with the 3rd degree polynomial transformation and the features PCR_01 and PCR_05 with the RBF transformation with $\gamma = 10^{-2}$ transformation. To decide on these features and their respective transformations we first looked at a univariate analysis of our features with respect to the contamination levels. We found that these three features showed some curvature, so we decided to test different feature mappings on then (using cross validation and taking the transformations that led to the best validation score).

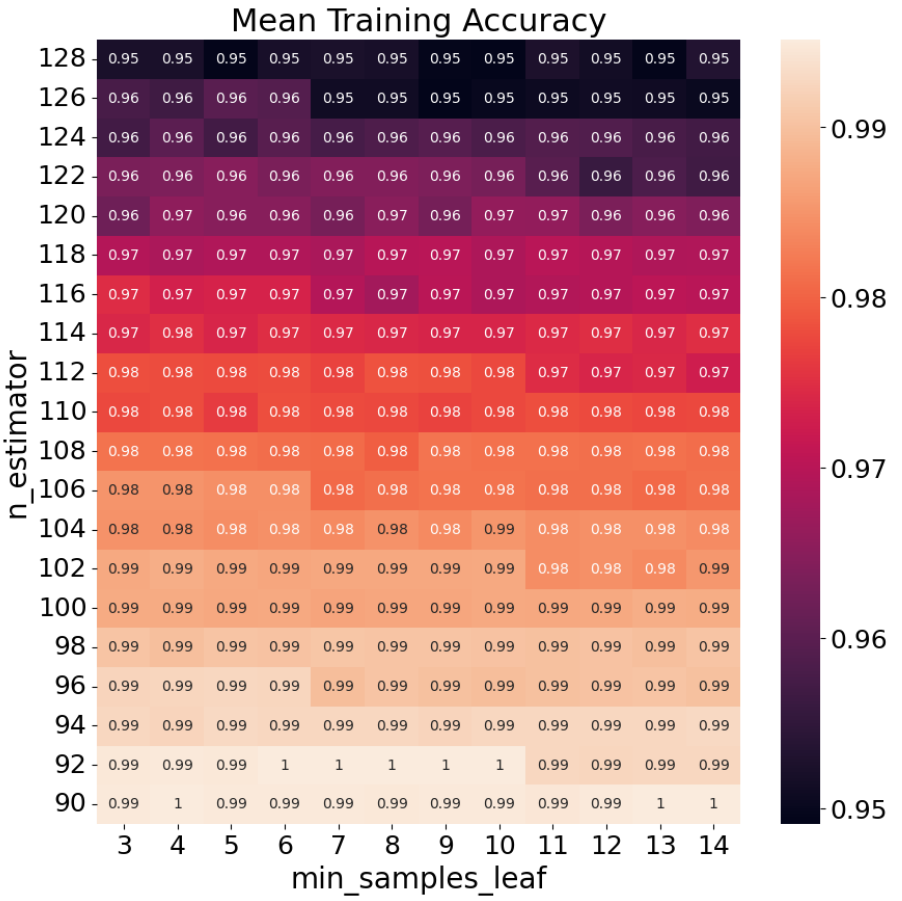
21.

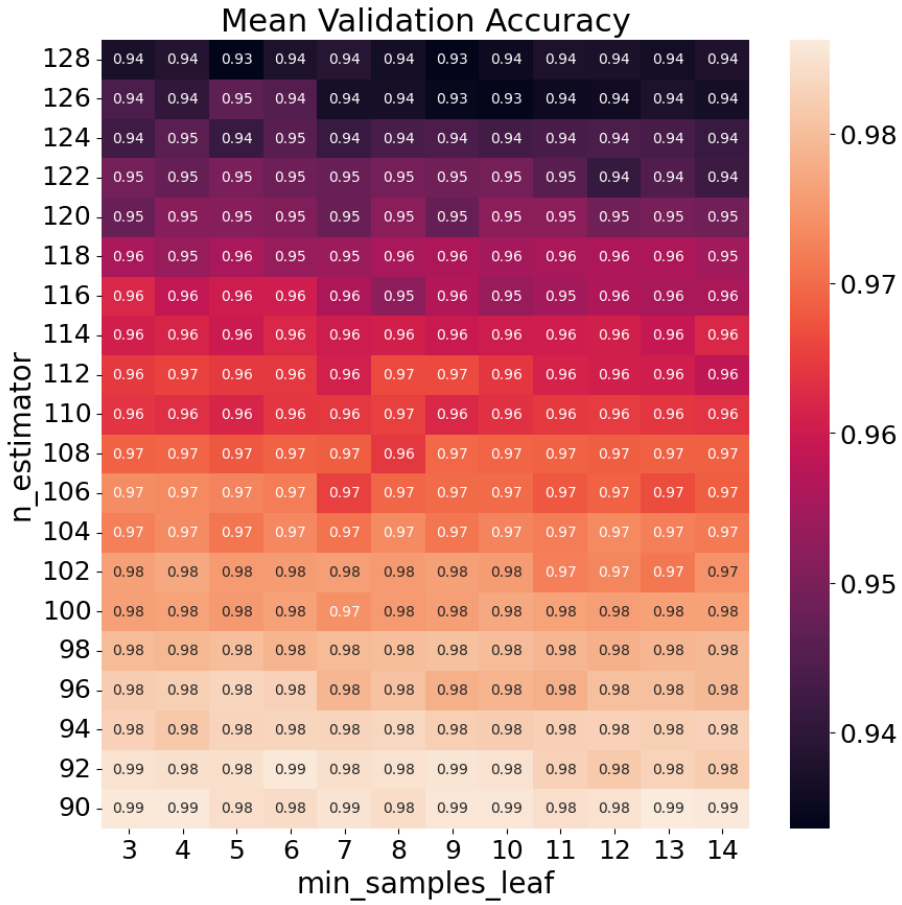
Each node in each tree in the Random Forest acts as a linear separator, yet many of our features are not distributed in a way that would allow for much information gain using a linear model. RBF allows adding some non-linearity by creating two new "areas" in the feature space, one inside the "bubbles" created by RBF and one outside. This can create overfitting on the training data, so for untuned RBF we'd expect smaller training error and higher validation error, but as we are tuning the γ we expect both to go down.

22.

A Random Forest selects only a few features each time and finds how well their combination can predict our target. This means that we do not have to do manual feature selection, as the random forest will explore on its own which combinations provide the most data and give that selection the most weight. It also combines several models, allowing for greater flexibility.

23.


Mean Training Accuracy
n_estimator
128 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95
126 0.96 0.96 0.96 0.96 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95
124 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96
122 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96
120 0.96 0.97 0.96 0.96 0.96 0.97 0.96 0.97 0.97 0.96 0.96 0.96
118 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97
116 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97
114 0.97 0.98 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97
112 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.97 0.97 0.97 0.97
110 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98
108 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98
106 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98
104 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.99 0.98 0.98 0.98 0.98
102 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.98 0.98 0.98 0.99
100 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99
98 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99
96 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99
94 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99
92 0.99 0.99 0.99 1 1 1 1 1 0.99 0.99 0.99 0.99
90 0.99 1 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.99 1 1
3 4 5 6 7 8 9 10 11 12 13 14
min_samples_leaf
0.99
0.98
0.97
0.96
0.95


Mean Validation Accuracy
n_estimator
128 0.94 0.94 0.93 0.94 0.94 0.94 0.93 0.94 0.94 0.94 0.94 0.94
126 0.94 0.94 0.95 0.94 0.94 0.94 0.93 0.93 0.94 0.94 0.94 0.94
124 0.94 0.95 0.94 0.95 0.94 0.94 0.94 0.94 0.94 0.94 0.94 0.94
122 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.94 0.94 0.94
120 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95 0.95
118 0.96 0.95 0.96 0.95 0.95 0.96 0.96 0.96 0.96 0.96 0.96 0.95
116 0.96 0.96 0.96 0.96 0.96 0.95 0.96 0.95 0.95 0.96 0.96 0.96
114 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96 0.96
112 0.96 0.97 0.96 0.96 0.96 0.97 0.97 0.96 0.96 0.96 0.96 0.96
110 0.96 0.96 0.96 0.96 0.96 0.97 0.96 0.96 0.96 0.96 0.96 0.96
108 0.97 0.97 0.97 0.97 0.97 0.96 0.97 0.97 0.97 0.97 0.97 0.97
106 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97
104 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97 0.97
102 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.97 0.97 0.97 0.97
100 0.98 0.98 0.98 0.98 0.97 0.98 0.98 0.98 0.98 0.98 0.98 0.98
98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98
96 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98
94 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98 0.98
92 0.99 0.98 0.98 0.99 0.98 0.98 0.99 0.98 0.98 0.98 0.98 0.98
90 0.99 0.99 0.98 0.98 0.99 0.98 0.99 0.99 0.98 0.98 0.99 0.99
3 4 5 6 7 8 9 10 11 12 13 14
min_samples_leaf
0.98
0.97
0.96
0.95
0.94

We found that the optimal hyper-parameters are 90 estimators and a minimum of 13 samples per leaf. This led to a training score of 4.58 and a validation score of 6.01

24.

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
RF Regressor	5	4.58	6.01

25.

Model	Section	Train MSE	Valid MSE	Test MSE
		Cross Validated		Retrained
Dummy	2	105.82	106.19	115.31
Linear	2	94.77	103.8	107.02
Lasso Linear	3	96.25	98.25	107.82
RF Regressor	5	4.58	6.01	4.17

We have found that the Random Forest Regressor performed best on the test set. This leads us to believe that any linear approach to the data is an essentially flawed one. All the other regressors failed to capture the complexity of our feature space by relying on linear functions for the hypothesis space. This led to large underfitting as the model could not fit the data well either in training, validation, or the actual test results.