

HW3 Report

Section 1

Q1:

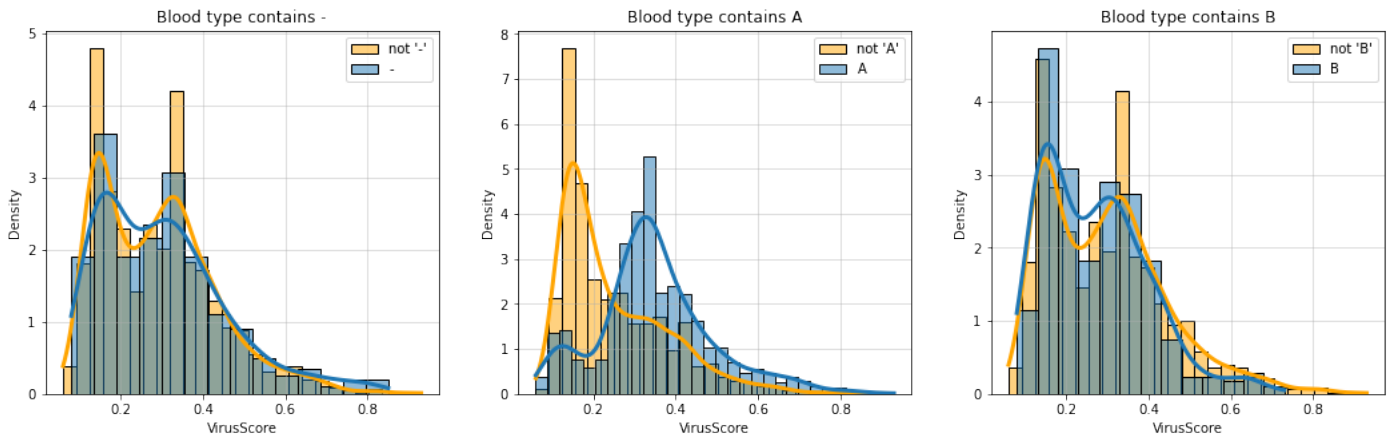


Figure 1: KDE plots of `VirusScore` conditioned different conditions of `blood_type`

Q2:

In figure 1 in the plot of A versus not A, we observe that the groups of patients with and without "A" in their blood types are mostly separable along a boundary that is approximately the `VirusScore` of 0.225.

Therefore, the condition of contains/does not contain A would be most informative for learning `VirusScore`. As it turns out, we decided already in hw1 to create this feature.

Q3:

$$\begin{aligned}
\frac{\partial}{\partial b} \mathcal{L}(\mathbf{w}, b) &\stackrel{a}{=} \frac{\partial}{\partial b} \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 \stackrel{b}{=} \frac{1}{m} \frac{\partial}{\partial b} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 \stackrel{c}{=} \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial b} (\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2 \stackrel{d}{=} \\
&\stackrel{d}{=} \frac{1}{m} \sum_{i=1}^m 2 (\mathbf{w}^\top \mathbf{x}_i + b - y_i) \cdot (1) \stackrel{e}{=} \frac{2}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + b - y_i) \stackrel{f}{=} \frac{2}{m} \left[mb + \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i - y_i) \right] \stackrel{g}{=} \\
&\stackrel{g}{=} 2b + \frac{2}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i - y_i) \Rightarrow \frac{\partial}{\partial b} \mathcal{L}(\mathbf{w}, b) = 2b + \frac{2}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i - y_i)
\end{aligned}$$

a : Definition of $\mathcal{L}(\mathbf{w}, b)$

b : $\frac{1}{m}$ is scalar

c : Derivative of a sum is the sum of derivatives

d : Derivative of $(\mathbf{w}^\top \mathbf{x}_i + b - y_i)^2$ w.r.t b

e : 2 is scalar

f : Sum of b

g : Removing b from the sum

Q4:

Residuals of analytical and numerical gradients

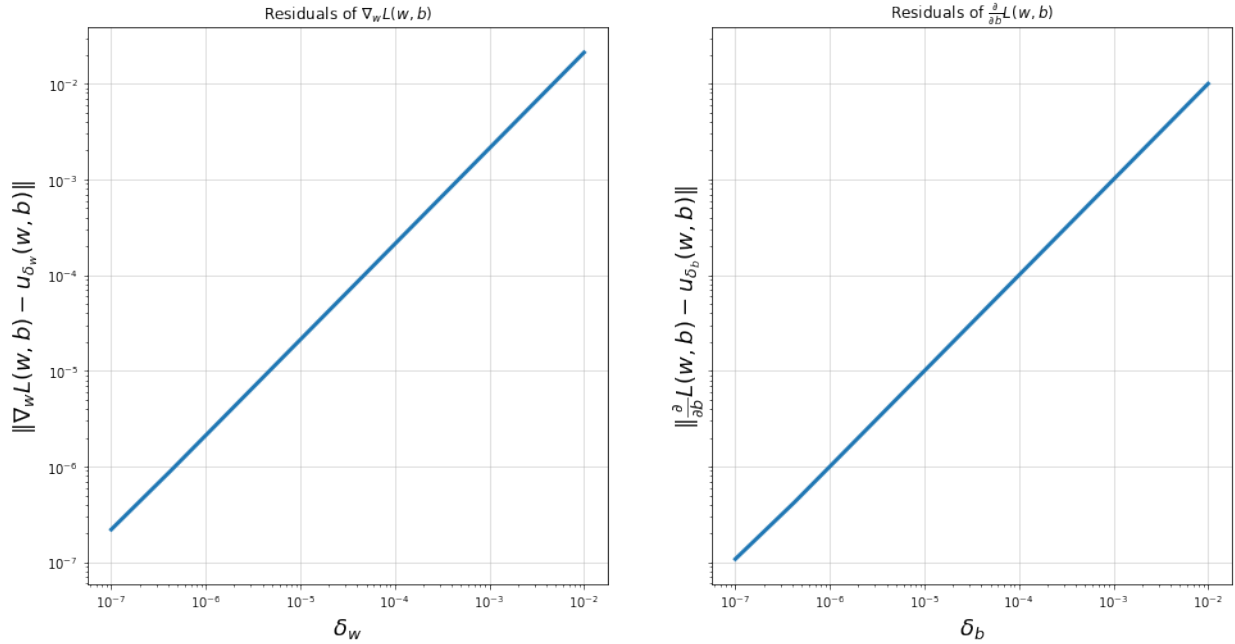


Figure 2: Plot of Residuals of analytical and numerical gradients

As we can see in figure 2, the difference between the analytic and numerical gradients increases in a monotonic fashion as the value of δ increases. This is logical, as δ is the differential size used in the definition of the numerical gradient, and therefore a smaller δ equates to a more precise estimation of the analytic gradient by the numerical gradient.

Q5:

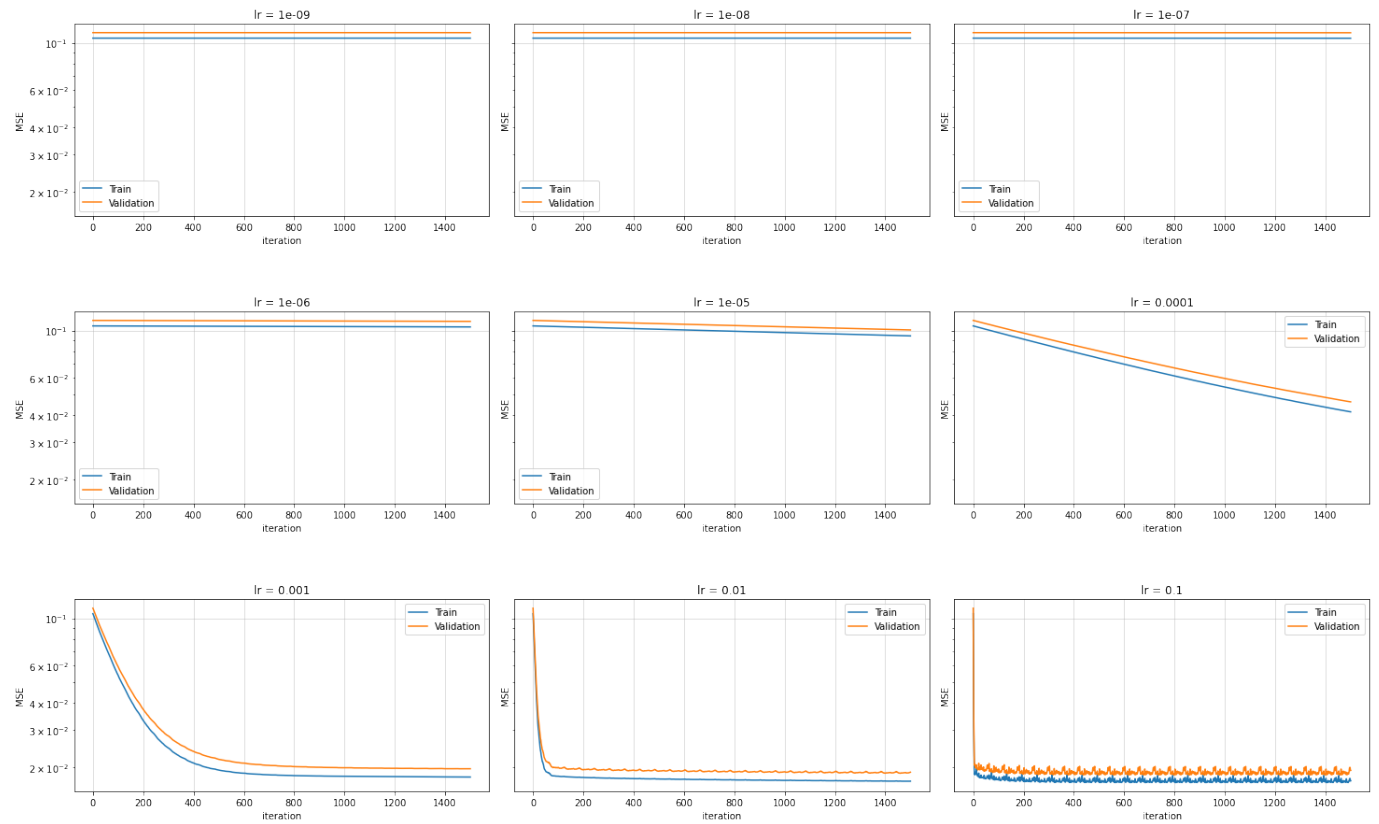


Figure 3: Graphs of Training and Validation Losses as Functions of Iteration Number for Different Learning Rates

We can see in figure 3 that for smaller lr , for those that converge, the convergence is at a higher loss for both the training and validation. This matches the theory, since if the lr is too small, the SGD algorithm is likely to converge to a sub-optimal solution. In addition we observe that for the lr equal to 0.1 the graphs do not converge for both training and validation losses and for lr equal to 0.01 the validation loss does not converge, which fits the theory that says that learning rates that are too high are likely to cause the SGD algorithm to take steps that are too large and thus repeatedly skip-over the optimal solution. This points to 0.001 as being the optimal lr , as both the validation and training losses converge to values that are substantially lower than the next smaller lr .

Q6:

Model	Section	Train MSE	Valid MSE
		Cross Validated	
Dummy	3	0.0204	0.0205

Figure 4: Training and Validation MSE Errors for Dummy Regressor

Section 4

Q7:

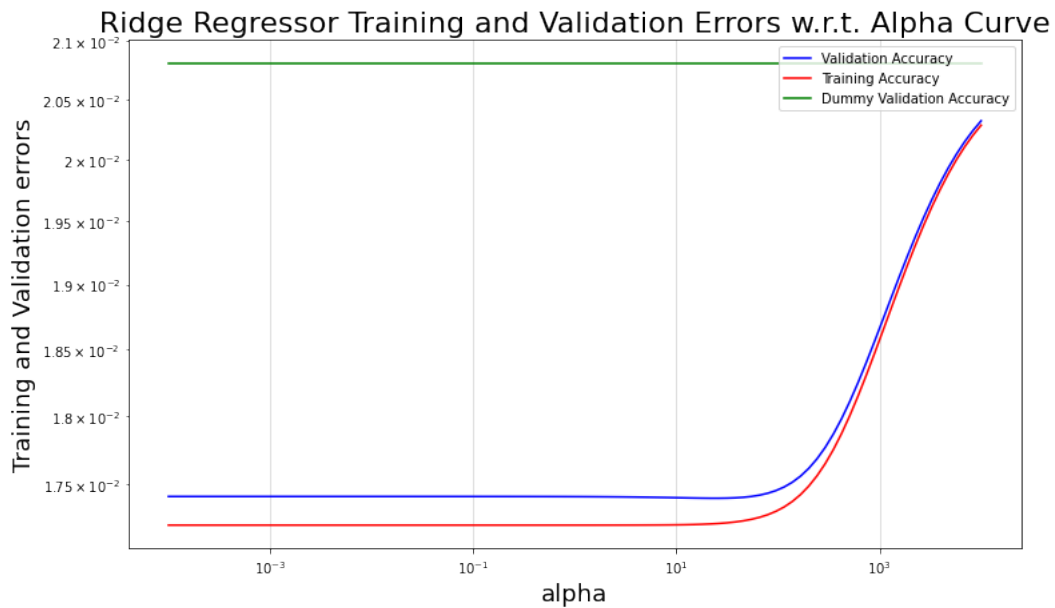


Figure 5: Graph of Training and Validation Errors of Ridge Regressor as a Function of α

Optimal alpha for validation: 25.950242113997373

Validation error for optimal alpha: 0.017402908351734445

Q8:

Model	Section	Train MSE	Valid MSE
		Cross Validated	
Dummy	3	0.0204	0.0205
Ridge linear	4	0.0172	0.0174

Figure 6: Training and Validation MSE Errors for Dummy and Ridge Regressors

Q9:

Feature	Coefficient
blood_A_AB	0.0970
num_of_siblings	0.0298
household_income	-0.0275
PCR_02	-0.0141
PCR_01	-0.0113

Figure 7: Five Features From Ridge Regressor With Largest Coefficients in Terms of Absolute Value, From Largest to Smallest

Q10:

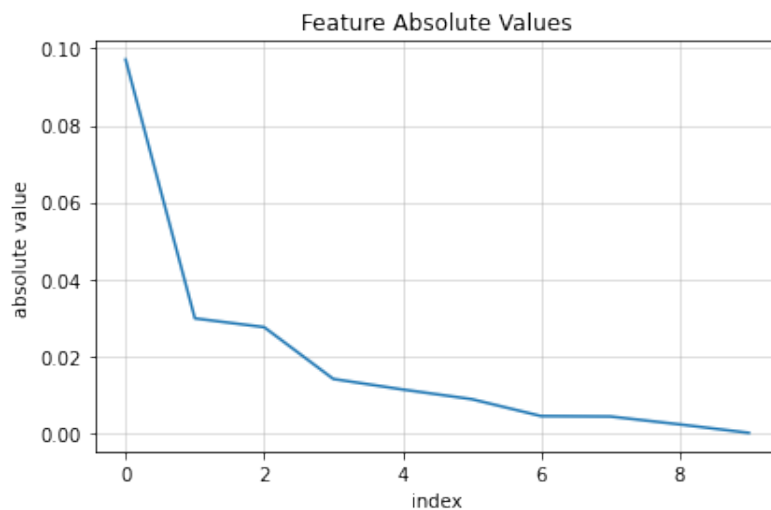


Figure 8: Graph of Absolute Values of Learned Coefficients of Ridge Regressor

Section 5

Q11:

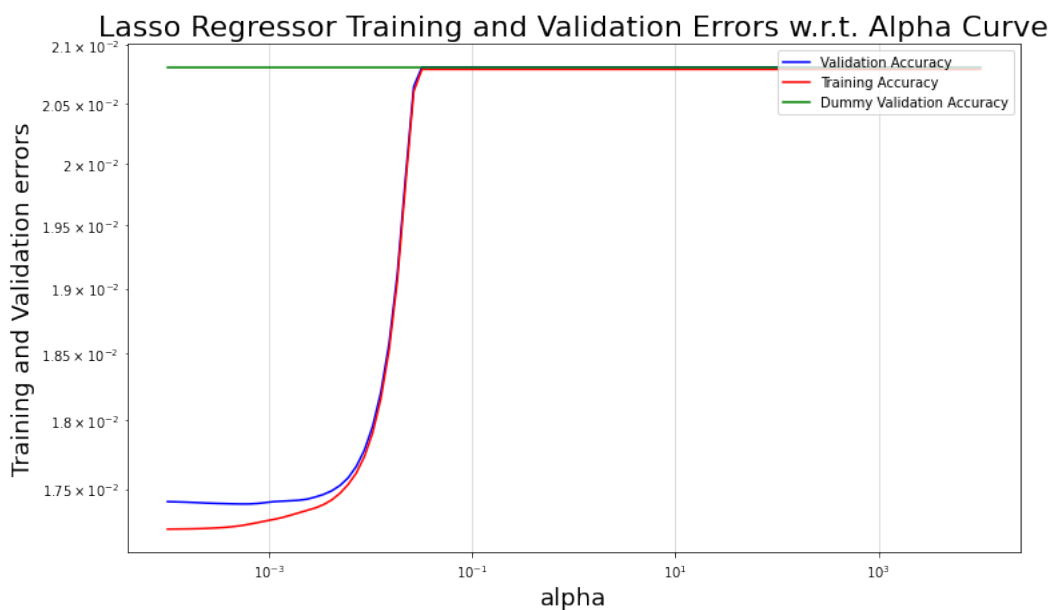


Figure 9: Graph of Training and Validation Errors of Lasso Regressor as a Function of α

Optimal alpha for validation: 0.0005336699231206312

Validation error for optimal alpha: 0.01739141826673015

Q12:

Yes, the graph for tuning the λ of the lasso regressor as can be seen in figure 9 is different from the graph obtained for the ridge regressor in figure 5 in a number of aspects.

Firstly, the steepness of the upwards curve is much more abrupt and significant in the case of the lasso regressor than it is for the ridge. This can be explained by the way in which lasso differs than ridge: Whereas a ridge regressor utilizes ℓ^2 regularization, lasso is implemented with ℓ^1 . This means that as λ (**alpha**) is increased, the weight coefficients for the two regressors decrease in different manners. In the case of ridge, the coefficients experience weight decay, meaning they gradually tend to zero, whereas lasso tends to perform variable selection, meaning that the coefficients are more abruptly forced to zero one-by-one. This is reflected in the graphs, as the abruptness seen in figure 9 could be explained by a certain significant coefficient abruptly forced to zero, whereas in the ridge graph of figure 5, coefficients are more gradually lowered and therefore the curve is more gradual.

Furthermore, the upwards trend in the graph of lasso begins at a much lower **alpha** approximately equal to 0.025 than it does for ridge (approximately **alpha** equal to 70). This can be again explained by the behavior of ℓ^1 as opposed to ℓ^2 : The decision of the lasso regressor to decrease a significant coefficient would be more abrupt and harsher than in the case of ridge, and therefore its effect on the accuracy would be seen almost immediately, and hence the earlier upwards trend in the lasso graph.

Lastly, the optimal **alpha** for minimal validation error in the case of lasso, equal to 0.0005, is much smaller than that of ridge (25.95).

Q13:

Model	Section	Train MSE	Valid MSE
		Cross Validated	
Dummy	3	0.0204	0.0205
Ridge linear	4	0.0172	0.0174
Lasso linear	5	0.0172	0.0173

Figure 10: Training and Validation MSE Errors for Dummy, Ridge, and Lasso Regressors

Q14:

Feature	Coefficient
blood_A_AB	0.0993
num_of_siblings	0.0297
household_income	-0.0223
PCR_02	-0.0064
PCR_10	0.0038

Figure 11: Five Features From Lasso Regressor With Largest Coefficients in Terms of Absolute Value, From Largest to Smallest

Q15:

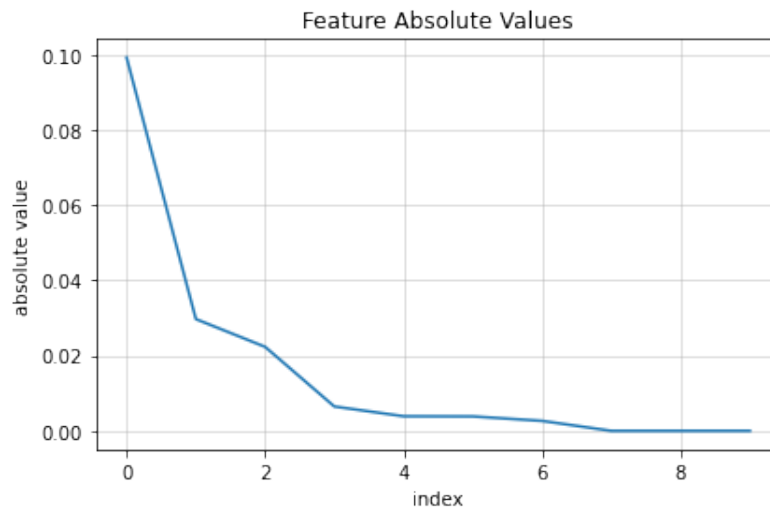


Figure 12: Graph of Absolute Values of Learned Coefficients of Lasso Regressor

Section 6

Q16:

When using a polynomial feature mapping, we can expect the training error to decrease and the validation error to increase. This is because the polynomial mapping will give more flexibility to the regression model to more closely try to fit the training data during training in a polynomial way, thereby leading to lower training error, but also causing overfitting and therefore leading to a higher validation error.

Q18:

Polynomially Mapped Ridge Regressor Training and Validation Errors w.r.t. Alpha Curve

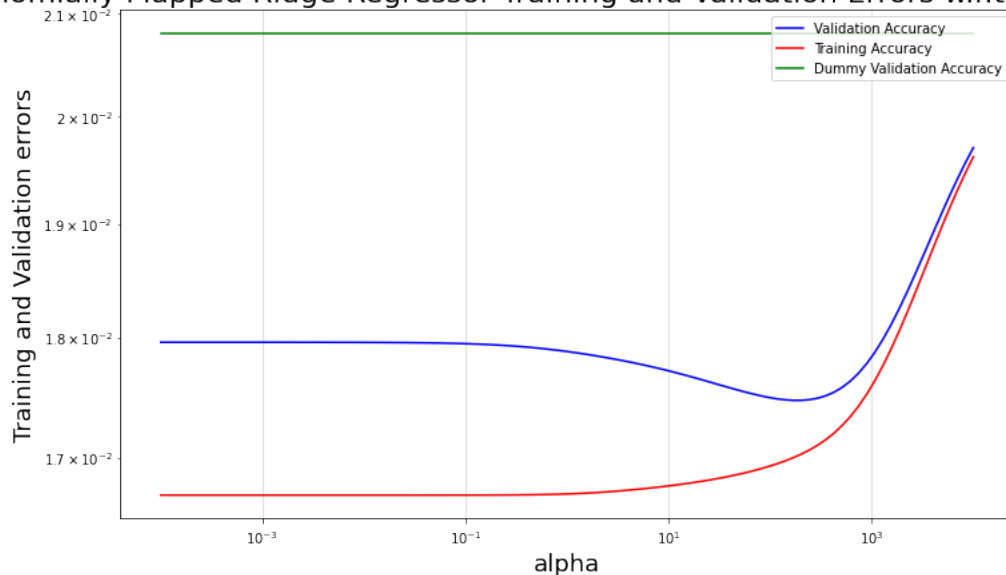


Figure 13: Graph of Training and Validation Errors of Polynomially-Mapped Ridge Regressor as a Function of alpha

Optimal alpha for validation: 200.923300256505

Validation error for optimal alpha: 0.01747015013288799

Q19:

Model	Section	Train MSE	Valid MSE
		Cross Validated	
Dummy	3	0.0204	0.0205
Ridge linear	4	0.0172	0.0174
Lasso linear	5	0.0172	0.0173
Ridge polynomial	6	0.0166	0.0174

Figure 14: Training and Validation MSE Errors for Dummy, Ridge, Lasso, and Polynomial Ridge Regressors

Section 7

Q20:

Model	Section	Train MSE	Valid MSE	Test MSE
		Cross Validated		Retrained
Dummy	3	0.0204	0.0205	0.0182
Ridge linear	4	0.0172	0.0174	0.0163
Lasso linear	5	0.0172	0.0173	0.0162
Ridge polynomial	6	0.0166	0.0174	0.0164

Figure 15: Training, Validation, and Test MSE Errors for Dummy, Ridge, Lasso, and Polynomial Ridge Regressors

As can be seen in figure 15, the lasso regressor performed the best on the test set, as it achieved the lowest MSE of 0.0162. Additionally, surprisingly enough, the test scores were better for all the models than their respective scores for both the training and validation sets. This possibly can be explained by the occurrence of underfitting, as perhaps there was enough variance in the training set, such that the validation sets chosen during the tuning process of all the models was varied enough, therefore causing a high optimal λ to be chosen in all cases, thereby creating models with high regularization.