(AI/ML) Assignment 2- Anish Deshpande-180100013

Theory for question 1:

Date :

Assignment 2

1.1) $W_{lasso} = \arg\min_{w} \|\phi\vec{\omega} - \vec{y}\|_2^2 + \lambda\|w\|_1$

To prove: ω_{lasso} = MAP estimate of linear regression subject to a Laplace prior on $\omega \sim \text{Laplace}(0, \theta)$

$$\text{Laplace}(\omega_i \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|w_i - \mu|}{b}\right)$$

$$\therefore \text{Laplace}(\omega_i \mid 0, b) = \frac{1}{2b} \exp\left(\frac{-|w_i|}{b}\right) = p(\omega_i \mid 0, b).$$

$\Phi(x)$: matrix : $\phi(x_j)$: feature vector for j^{th} sample.

By Bayes Theorem : (Y is the data)

$$P_{os}(w \mid Y) = \frac{P(Y \mid w)\, P_{rior}(w)}{P(Y)}$$

Here $\text{prior}(w) = \prod_{i=1}^{p} \frac{1}{2b} \exp\left(\frac{-|w_i|}{b}\right)$

(assuming 'p' features).

$P(Y|\omega) =$

We know $Y = \omega^T \Phi(x) + \mathcal{E}$

$(\mathcal{E} \sim \mathcal{N}(0, \sigma^2))$

$$\therefore P(y_j \mid x_j, w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_j - w^T\phi(x_j))^2}{2\sigma^2}\right)$$

$$\therefore P(y_j|x,w) = \prod_{j=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_j - w^T\phi(x_j))^2}{2\sigma^2}\right).$$

$$\because P(W|Y) = \frac{P(Y|W)\,P(W)}{P(Y)}$$

$$\hat{W}_{MAP} = \underset{w}{\operatorname{argmax}}\ P(W|Y) = \underset{w}{\operatorname{argmax}}\ \frac{P(Y|W)\,P(W)}{P(Y)}$$

$$= \underset{w}{\operatorname{argmax}}\ \log(P(Y|w)) + \log(P(W)) + \text{constant w.r.t } w$$

$$(\log(P(Y)).$$

$$\therefore \hat{W}_{MAP} = \underset{w}{\operatorname{argmax}}\left[\log\left(\prod_{j=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_j - w^T\phi(x_j))^2}{2\sigma^2}\right)\right)\right.$$

$$\left. + \log\left(\prod_{i=1}^{p} \frac{1}{2b} \exp\left(-\frac{|w_i|}{b}\right)\right)\right]$$

$$\therefore \hat{W}_{MAP} = \underset{w}{\operatorname{argmax}}\left[\sum_{j=1}^{m}\left[\frac{-[y_j - w^T\phi(x_j)]^2}{2\sigma^2} + \frac{1}{b}\sum_{i=1}^{p} -|w_i|\right]\right]$$

$$(\operatorname{Log}\left(1/\sqrt{2\pi\sigma^2}\right) \text{ and } \log(1/2b) \text{ are constant.}$$

$$\therefore \hat{W}_{MAP} = \underset{w}{\operatorname{argmin}}\left(\sum_{j=1}^{m} \frac{(y_j - w^T\phi(x_j))^2}{2\sigma^2} + \frac{1}{b}\sum_{i=1}^{p} |w_i|\right)$$

(removing negative sign)

$$\sum_{i=1}^{p} |w_i| = \|w\|_1$$

Also $\sum_{j=1}^{m} (y_j - w^T \phi(x_j))^2$

In matrix form is : $\|y - \phi w\|_2^2$

$= \|\phi w - y\|_2^2$

$$\therefore \hat{w}_{MAP} = \underset{w}{\operatorname{argmin}} \left(\frac{\|\phi w - y\|_2^2}{2\sigma^2} + \frac{1}{b} \|w\|_1 \right)$$

$$\therefore \hat{w}_{MAP} = \frac{1}{2\sigma^2} \underset{w}{\operatorname{argmin}} \left(\|\phi w - y\|_2^2 + \frac{2\sigma^2}{b} \|w\|_1 \right)$$

But

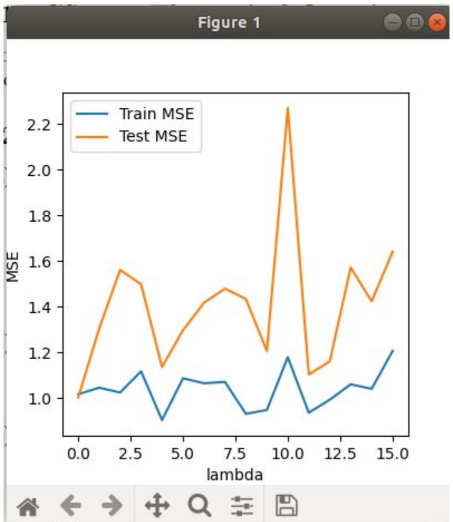
$$w_{LASSO} = \underset{w}{\operatorname{argmin}} \left(\|\phi w - y\|_2^2 + \lambda \|w\|_1 \right)$$

$\therefore$ Taking $\lambda = \frac{2\sigma^2}{b}$, we see that the solution for lasso is the same as the MAP estimate of linear regression with a ~~Gaussian Pric~~ Laplacian Prior.

Hence Proved.

Q1)

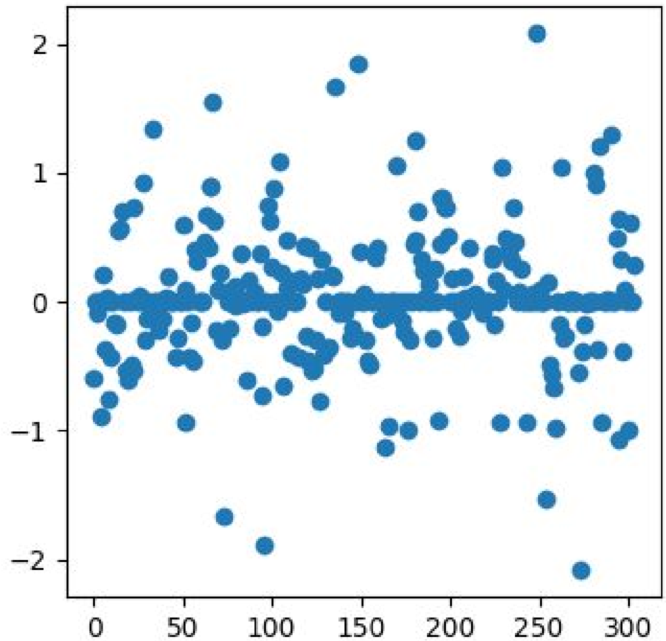
1b)



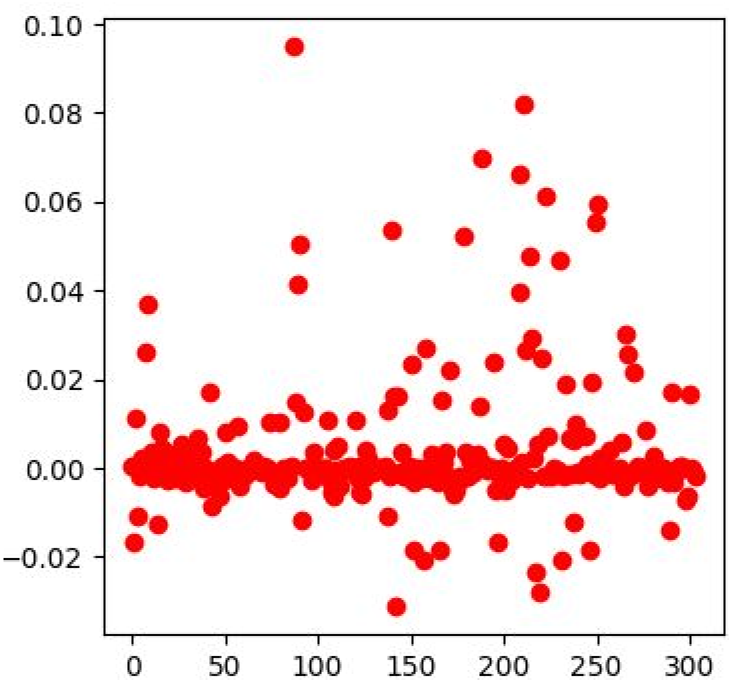
I understand that this is not how the plot is supposed to look, but it is the only way it was plotted, despite numerous attempts. However, lambda = 0.1 did give a very low value of MSE, and in my individual trials, the MSE increased for larger values of lambda.

Lambda = 0.1 is optimal because any lower and we do not consign enough variables to zero, and any higher we start observing underfitting effects with an increase in the error.

Q1.c) **Scatter plot of weights for lasso regression:**
Y-axis: value of each weight vector element
X-axis: index of each weight vector element



Scatter plot of weights for ridge regression:
Y-axis: value of each weight vector element
X-axis: index of each weight vector element



For ridge regression, we see a lot of weights close to zero, but in the case of lasso regression, a lot of the weights in the neighbourhood of zero are exactly zero. Also, lasso regression looks sparser than ridge regression.

Q2) Theory for the 2 different types of perceptron:

Date : ___________

2.1 Say that we have 'C' classes in our dataset.

* In the 'one-vs-rest' model, we have a binary classification between one class and the rest of the classes taken together. Hence, we have 'C' binary classification problems, one per class.
In the 'one-vs-one' model, we have a binary classification problem per pair of classes.
$\therefore \binom{C}{2}$ binary classification problems. $\left(\frac{C(C-1)}{2}\right)$

* Another difference lies in prediction. For
'One-vs-rest': The class whose weighted score $w \cdot$ is the highest for a given sample is the predicted class.
'one-vs-one': Each classifier 'provides' (outputs) a label for the prediction, and the final predicted class is the class label that received the most number of 'votes' among the $\binom{C}{2}$ binary classifiers.

For large number of classes, 1vsRest is better (1vs1 is not scalable).

for a very large dataset and a moderate number of classes, 1vs1 is good, (1vsRest: each model trains on the whole dataset), as each classifier doesn't need the whole dataset.

Question3) Bias Variance tradeoff.
Theory:

3.1 $\text{Bias} = E[\hat{f}(x)] - f(x)$

predicted true.

$\text{Variance} = E(\hat{f}(x) - E[\hat{f}(x)])^2$

predicted average predicted.

(on the same value, different realisations of the model.

a) Increasing 'λ' in lasso regression:

→ This is done to prevent overfitting.

→ Even a higher degree fit looks like a lower degree fit for sufficiently high λ

* Increasing 'λ' increases the bias of the model, but decreases the variance.

→ This is because, higher the lamba, more the selective our model (more coefficients are tended to zero), and the model effectively has less variables.

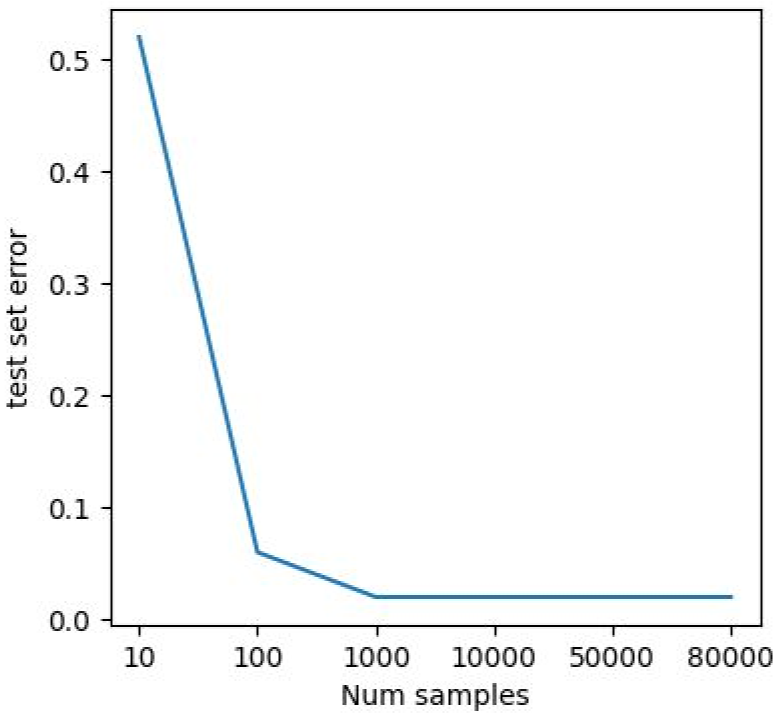
3.1b) Adding more training examples for a perceptron will initially lower both the bias and variance, as each class gets appropriately weighted. If the number of iterations are constant then more the training examples, the better. After a while,

3.2

3.1 c) Adding more features (linearly dependent) will not improve the performance of the lasso. It will, however, increase the bias a little as these new features may go to zero along with the old features, reducing the number of variables the lasso regression model selects.
There will be ~~unstable selections~~ inconsistent selections of linearly dependent variables also, at times. This may increase the variance of the model.

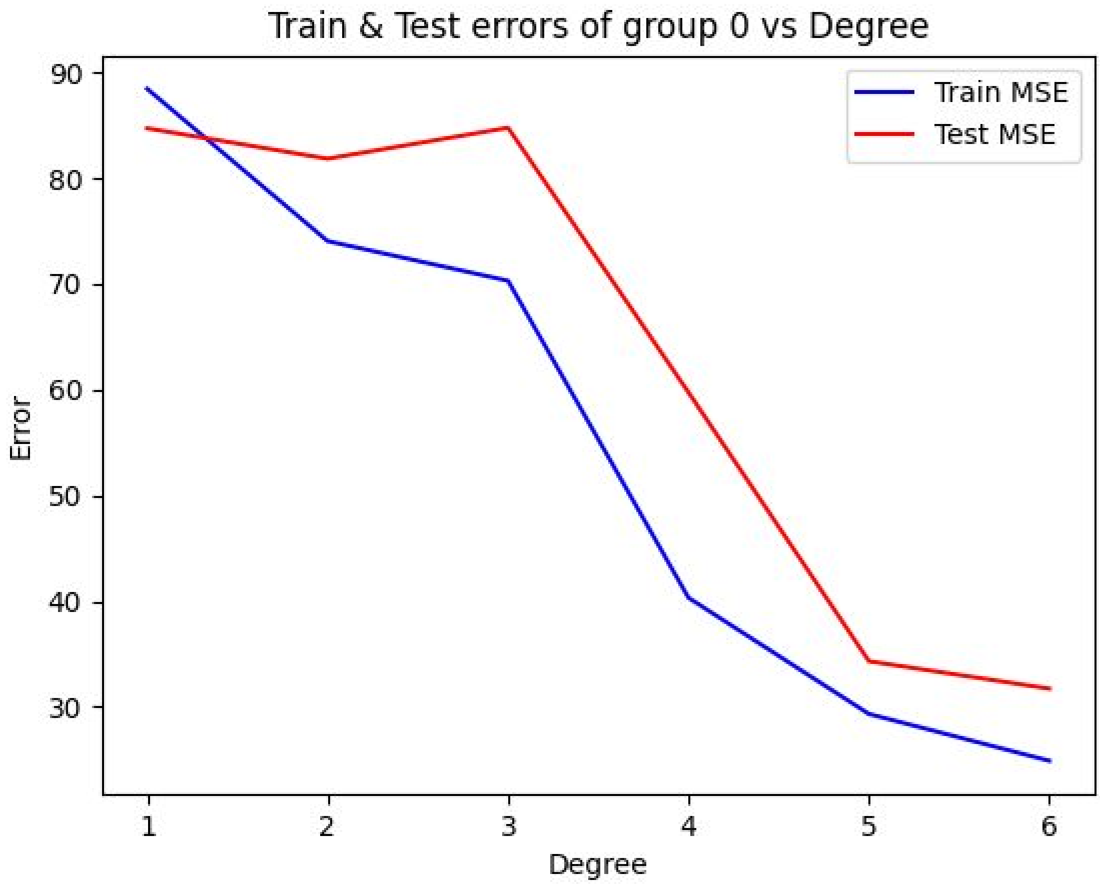
Q3.2)LAB:

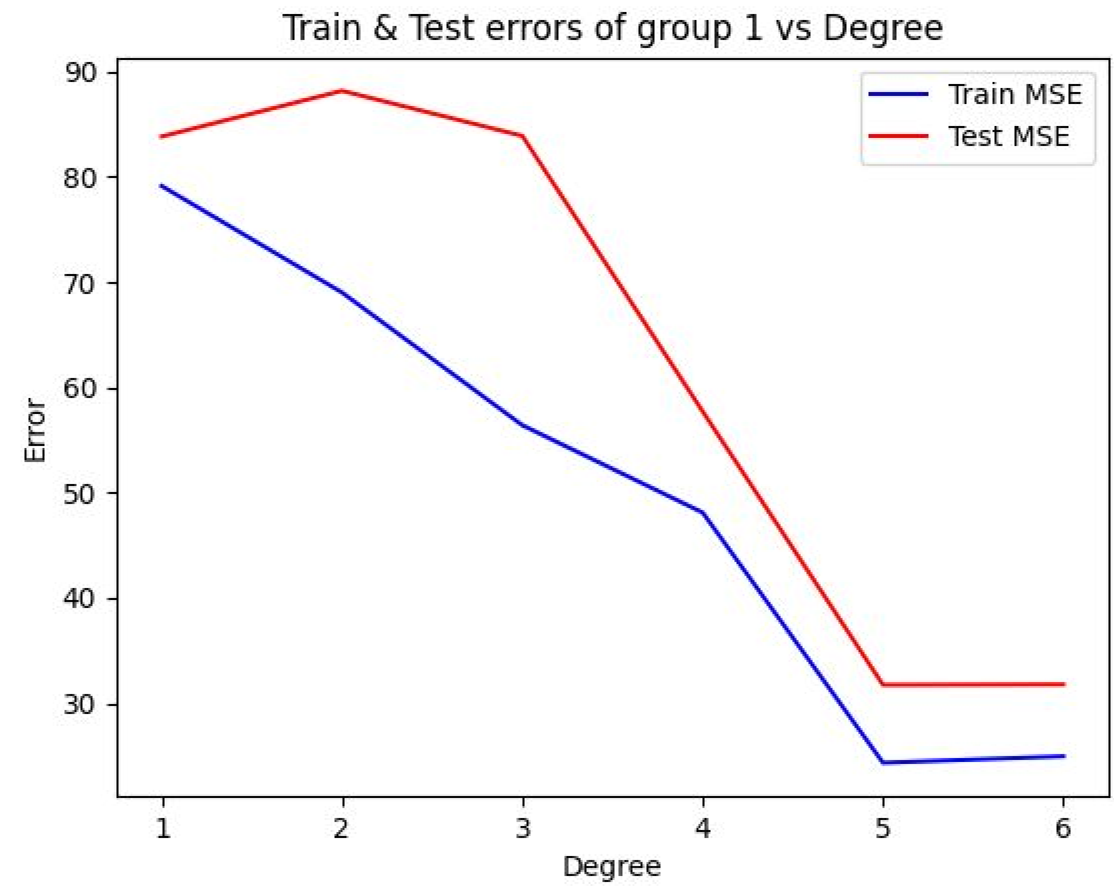
This plot shows the error in the test set for the 1-vs-Rest perceptron, as a function of the number of training samples.

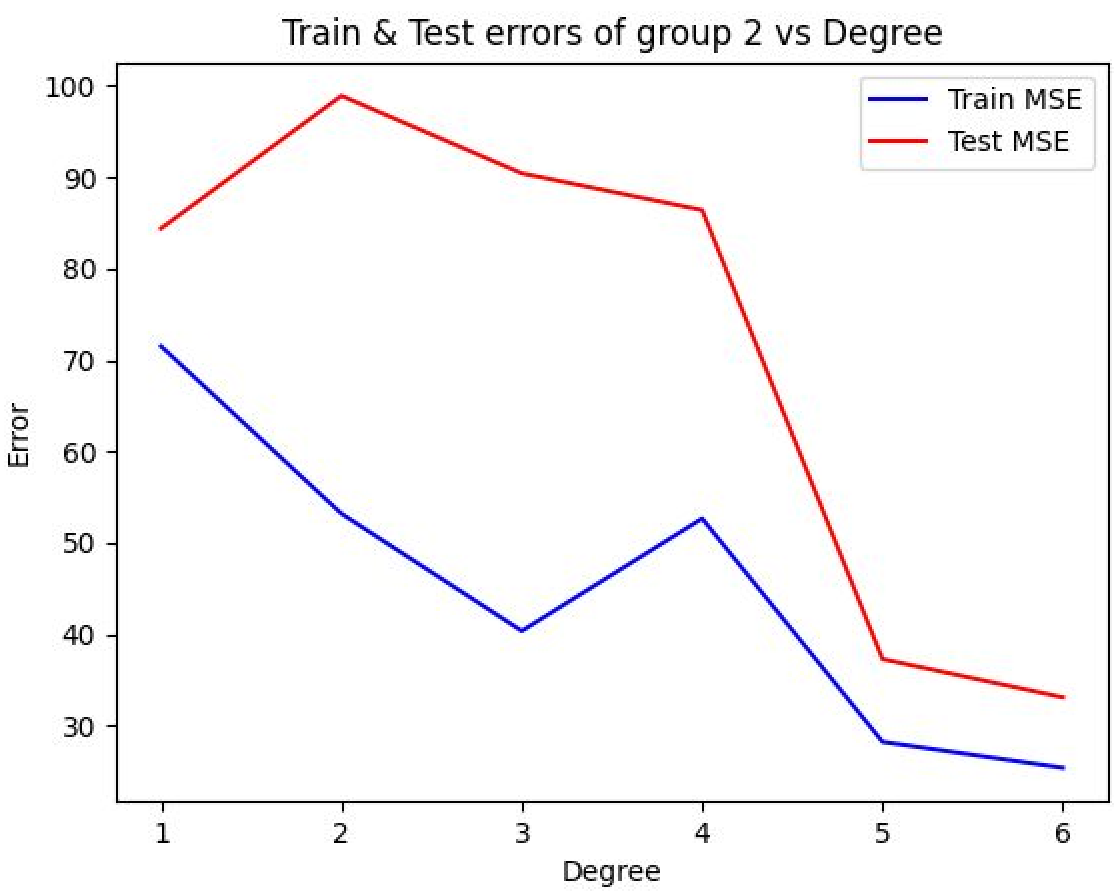


The bias decreases a little as the error stagnates at 0.02. It does not go below this value. The variance in the test set reduces and stagnates. It does not reach 0.

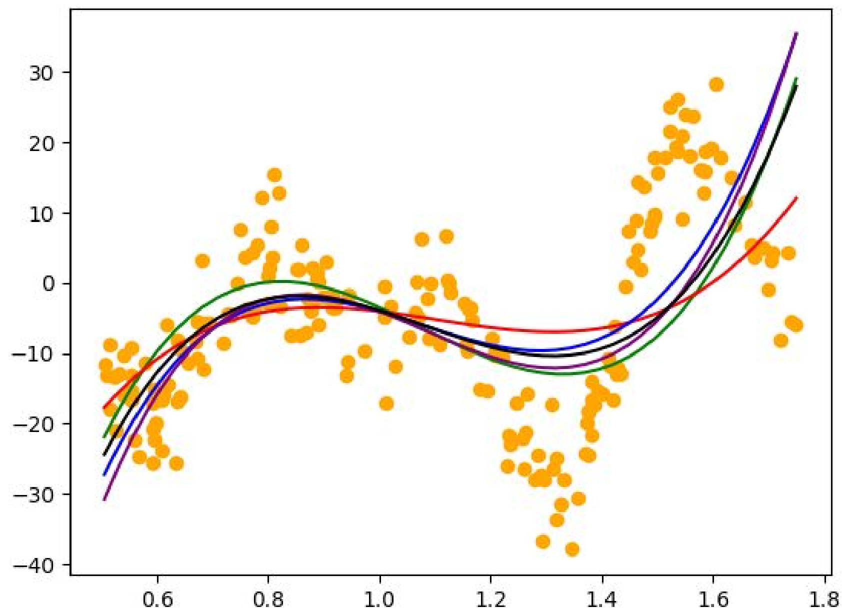
For the question p3.py:
Here we have the 4 plots (for each subset of the indices) plotting train and test erros versus the degree of the polynomial fit .

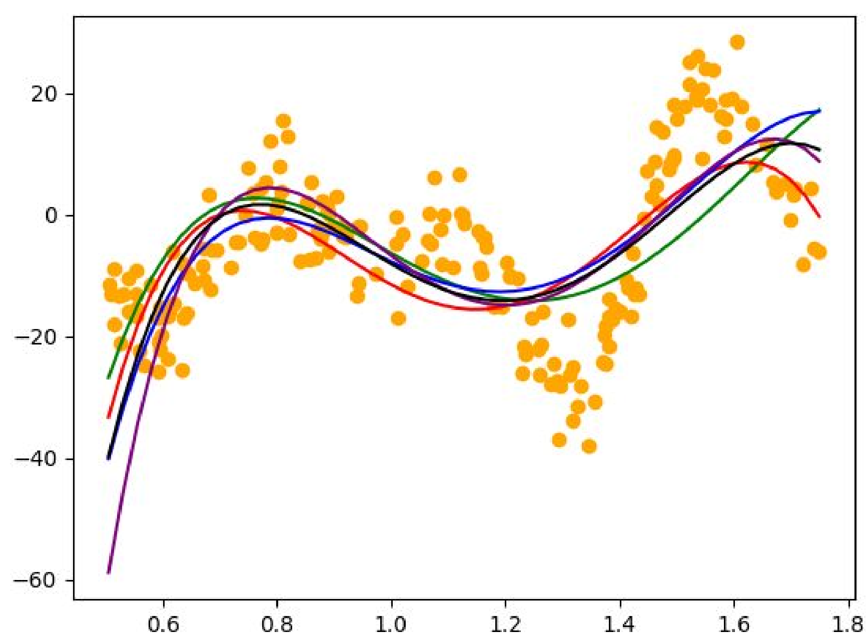


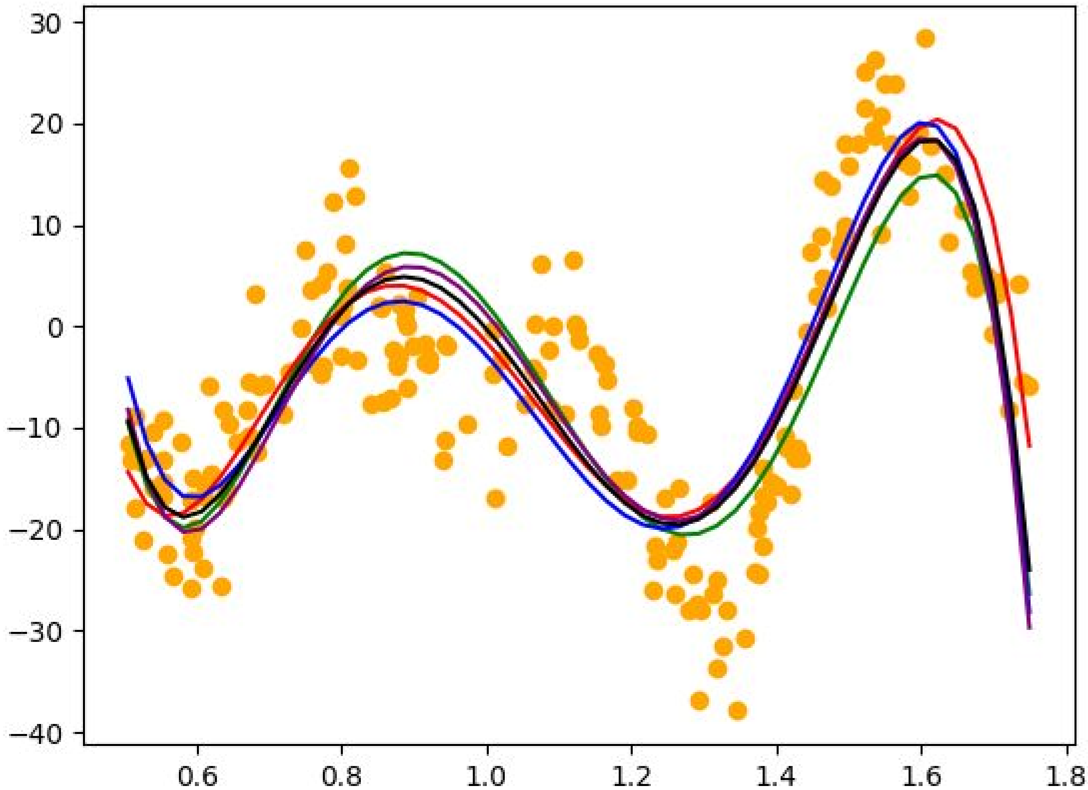

Train & Test errors of group 1 vs Degree
Train MSE
Test MSE
Error
Degree
90
80
70
60
50
40
30
1
2
3
4
5
6

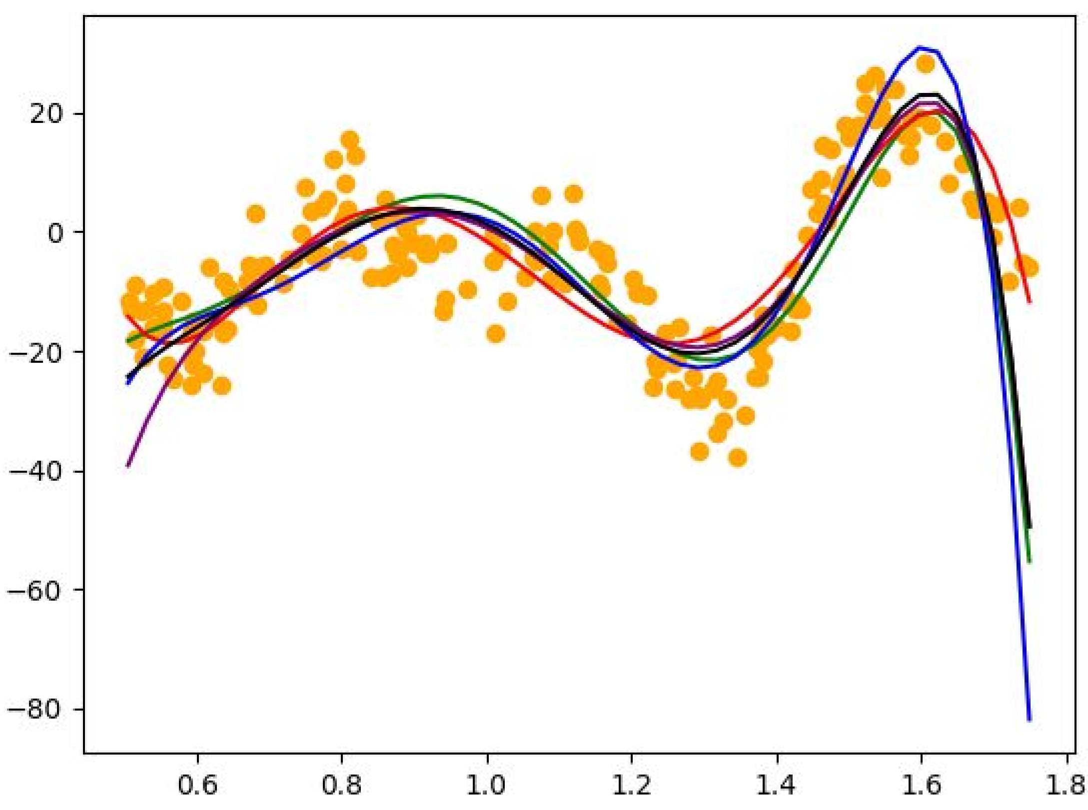

Train & Test errors of group 2 vs Degree
Train MSE
Test MSE
Error
Degree
100
90
80
70
60
50
40
30
1
2
3
4
5
6

Here are the plots for the polynomial fit for all four subsets for degree 3,4,5 and 6. Theoretically, the data is observed to have 5 points of extrema, (derivative 0 at 5 points), and hence suggests that a polynomial of degree 6 (at least) must have the minimum training error.
With the increase in degree of polynomial, we expect overfitting to show. But this has not happened yet, as the maximum degree (6) is close to the actual degree of the polynomial. The bias decreases with the increase in degree, while the variance is set to increase. An appropriate tradeoff between the two is obtained at degree = 6.






30
20
10
0
−10
−20
−30
−40
0.6
0.8
1.0
1.2
1.4
1.6
1.8


20
0
−20
−40
−60
−80
0.6
0.8
1.0
1.2
1.4
1.6
1.8