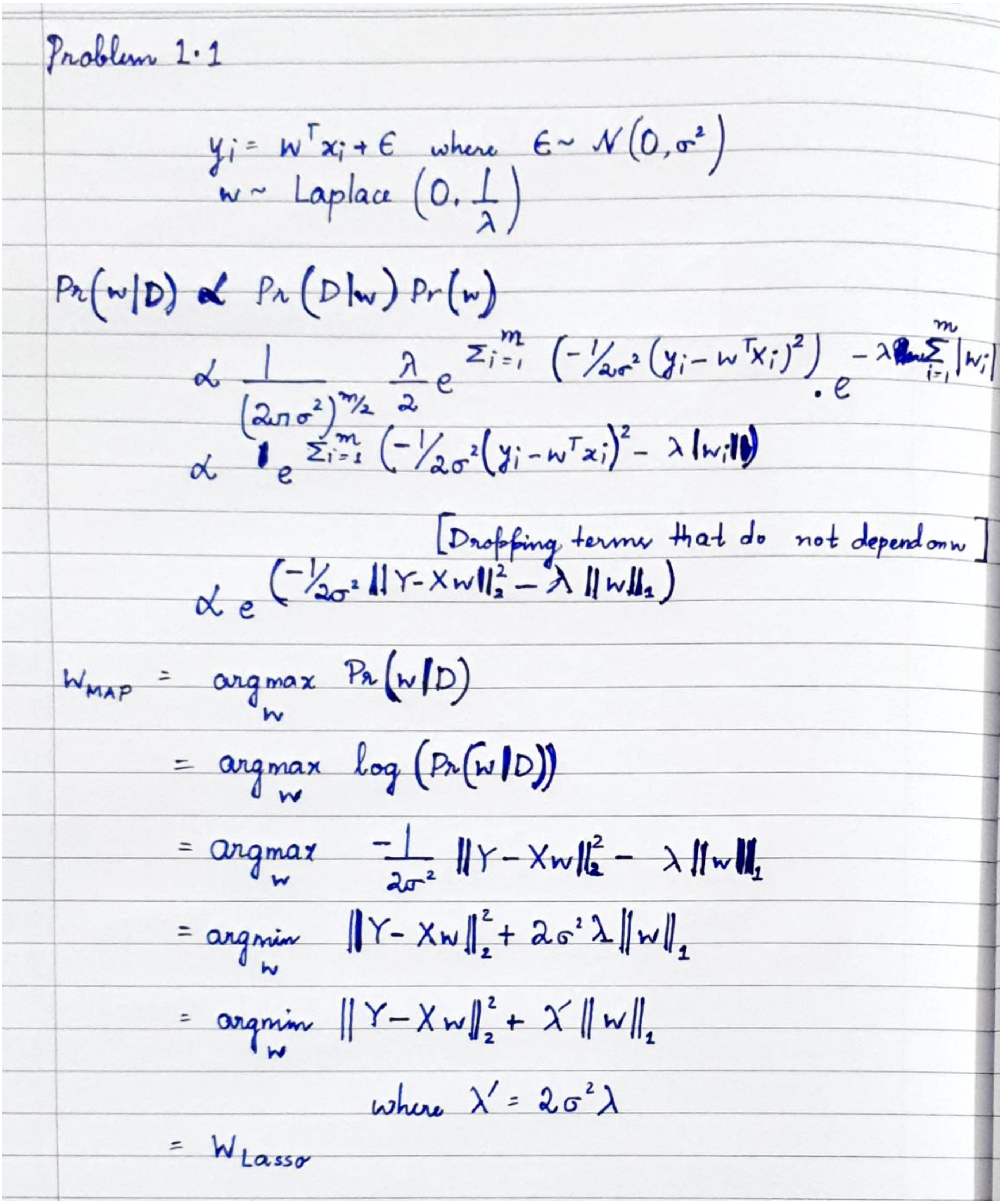
Problem 1



Problem 1·1

$$y_i = w^T x_i + \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

$$w \sim \text{Laplace}\left(0, \frac{1}{\lambda}\right)$$

$$Pr(w|D) \propto Pr(D|w)\, Pr(w)$$

$$\propto \frac{1}{(2\pi\sigma^2)^{m/2}} \frac{\lambda}{2} e^{\Sigma_{i=1}^{m} \left(-\frac{1}{2\sigma^2}(y_i - w^T x_i)^2\right)} \cdot e^{-\lambda \sum_{i=1}^{m} |w_i|}$$

$$\propto 1\, e^{\Sigma_{i=1}^{m} \left(-\frac{1}{2\sigma^2}(y_i - w^T x_i)^2 - \lambda |w_i|\right)}$$

[Dropping terms that do not depend on w]

$$\propto e^{\left(-\frac{1}{2\sigma^2} \|Y - Xw\|_2^2 - \lambda \|w\|_1\right)}$$

$$W_{MAP} = \underset{w}{\text{argmax}}\ Pr(w|D)$$

$$= \underset{w}{\text{argmax}}\ \log(Pr(w|D))$$

$$= \underset{w}{\text{argmax}}\ \frac{-1}{2\sigma^2} \|Y - Xw\|_2^2 - \lambda \|w\|_1$$

$$= \underset{w}{\text{argmin}}\ \|Y - Xw\|_2^2 + 2\sigma^2 \lambda \|w\|_1$$

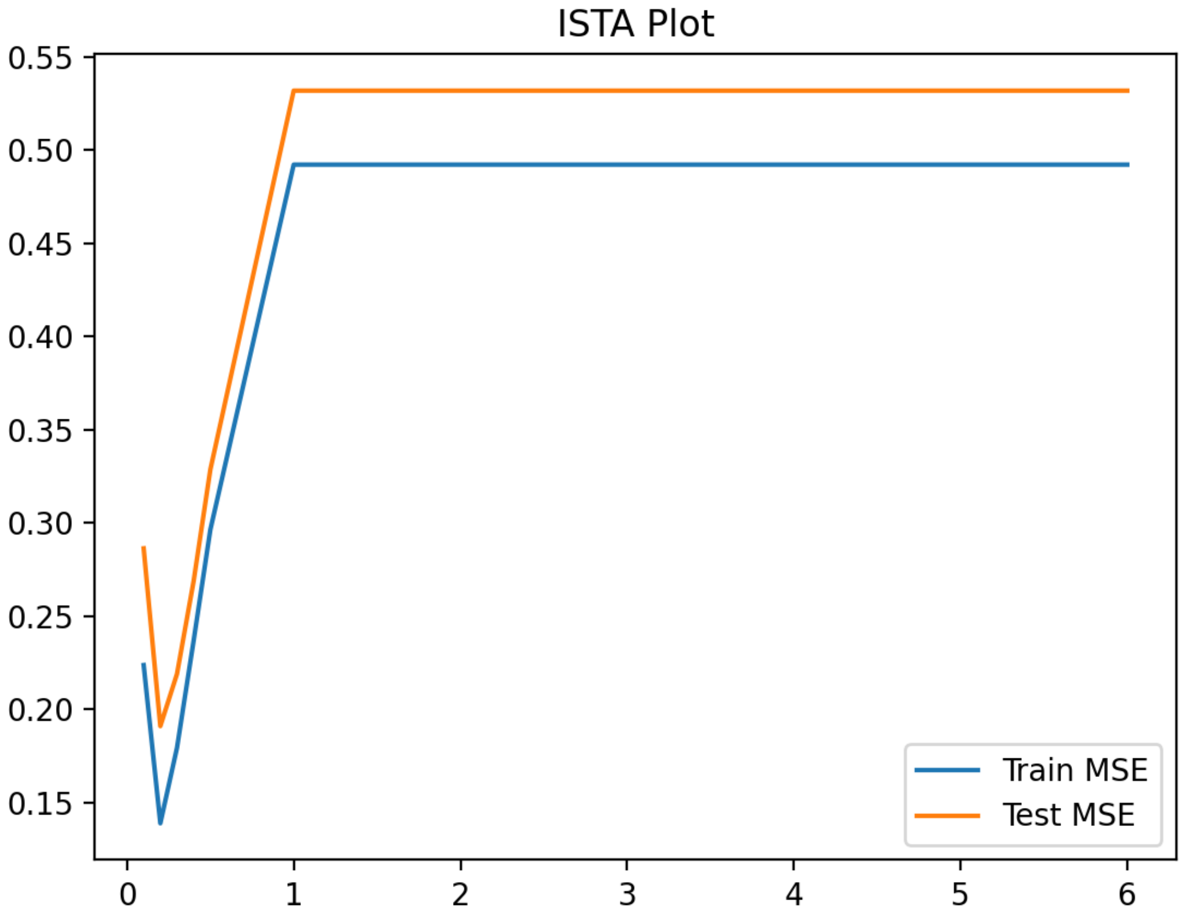
$$= \underset{w}{\text{argmin}}\ \|Y - Xw\|_2^2 + \lambda' \|w\|_1$$

where $\lambda' = 2\sigma^2 \lambda$

$$= W_{Lasso}$$

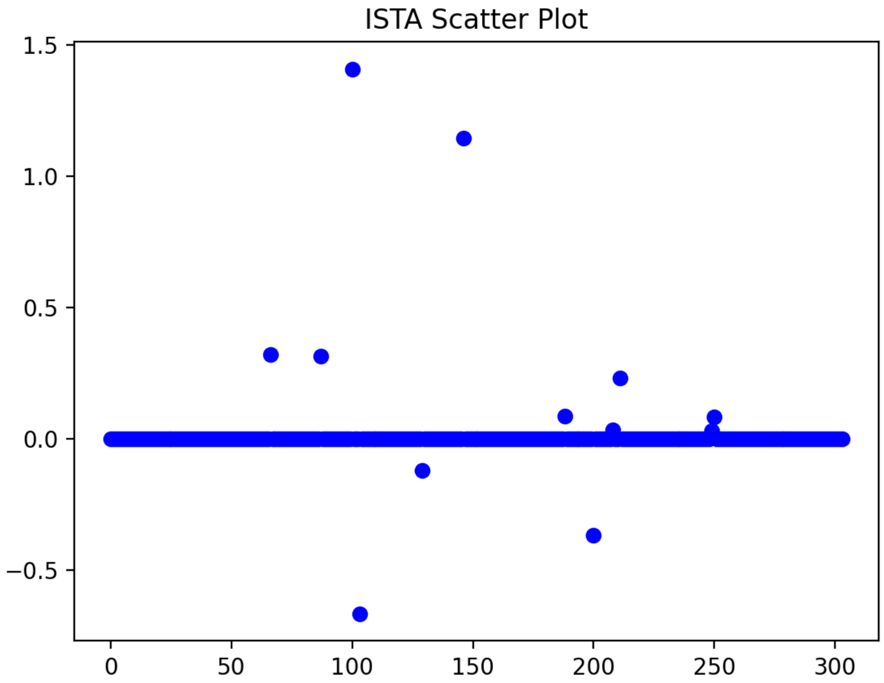
Problem 1.2

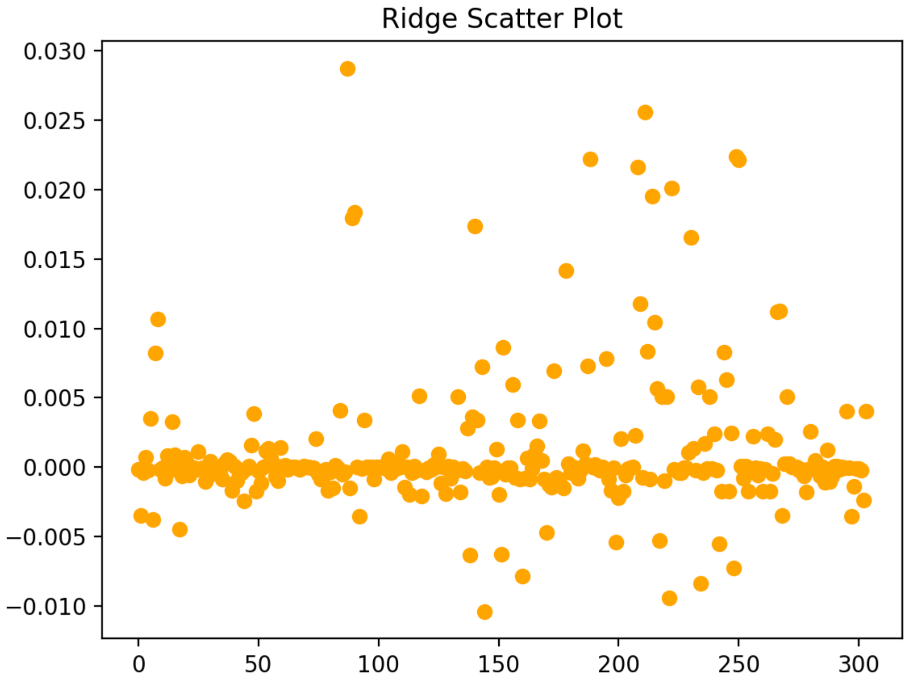
(b) MSE vs lambda



The optimal value of lambda is 0.2 as the test and train MSE are minimum at lambda = 0.2
We observe that the value of MSE decreases till 0.2 and then increases till 1. Further it remains constant. This is because the weights approach zero with higher values of lambda and hence deviate from the optimal values. The value then becomes constant at higher values of lambda as most of the values have been driven to 0. The dip initially is though problem specific.

(c)

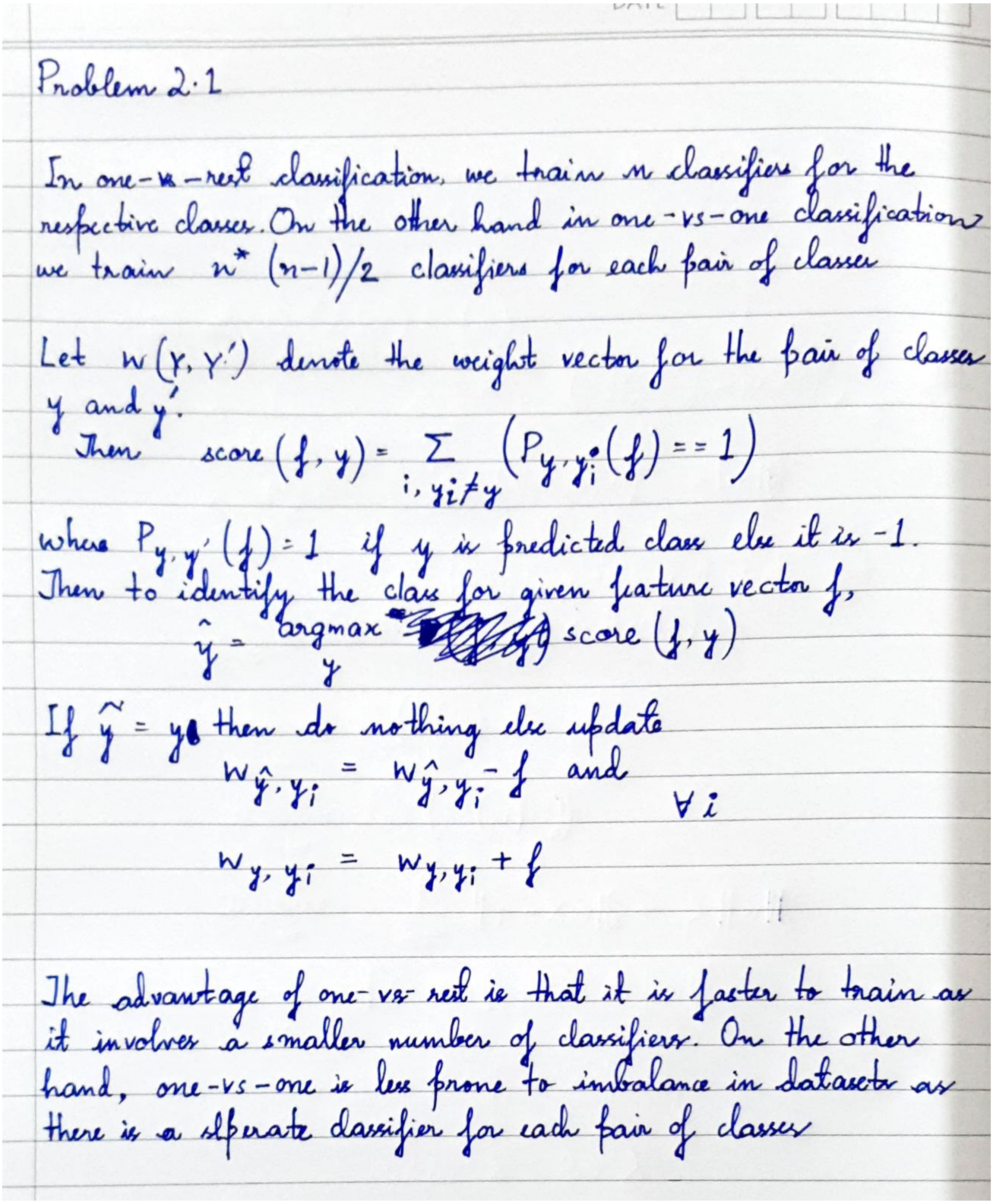




Ridge regression tries to penalize the Euclidean norm and hence reduces the weights with increasing values of regularization parameter. On the other hand, Lasso penalizes L1-norm and hence increases the number of weights with 0 value. This can be seen from the above two plots.

Problem 2

Problem 2.1



Problem 2·1

In one-vs-rest classification, we train n classifiers for the respective classes. On the other hand in one-vs-one classification we train $n^{*}(n-1)/2$ classifiers for each pair of classes

Let $w(y, y')$ denote the weight vector for the pair of classes y and y'.

Then $$\text{score}(f, y) = \sum_{i,\, y_i \neq y} \left(P_{y, y_i}(f) == 1\right)$$

where $P_{y,y'}(f) = 1$ if y is predicted class else it is -1.

Then to identify the class for given feature vector f,

$$\hat{y} = \underset{y}{\text{argmax}}\ \text{score}(f, y)$$

If $\tilde{y} = y$ then do nothing else update

$$w_{\hat{y}, y_i} = w_{\hat{y}, y_i} - f \text{ and} \qquad \forall i$$

$$w_{y, y_i} = w_{y, y_i} + f$$

The advantage of one-vs-rest is that it is faster to train as it involves a smaller number of classifiers. On the other hand, one-vs-one is less prone to imbalance in datasets as there is a seperate classifier for each pair of classes

Problem 3

Problem 3.1

(a) Increasing the value of lambda increases the bias and decreases the variance. The bias is increased as the weights are penalized with increasing values of lambda and hence deviate from original function. Variance decreases as minor changes in data won't affect the weights as the lambda is high enough to penalize such changes.

(b) Increase in number of train examples results in a decrease in variance. This is because changes in some of the data points won't affect the predicted function as it has been derived from a large dataset. We can't say anything about bias as it is problem specific.

(c) The changes to bias and variance will be problem specific as it depends on the coefficients of linear dependence of a given feature with other features. However in case it affects the prediction(some features cannot be selected by the algorithm), it results in high bias and low variance otherwise bias and variance remain unchanged.

Problem 3.2

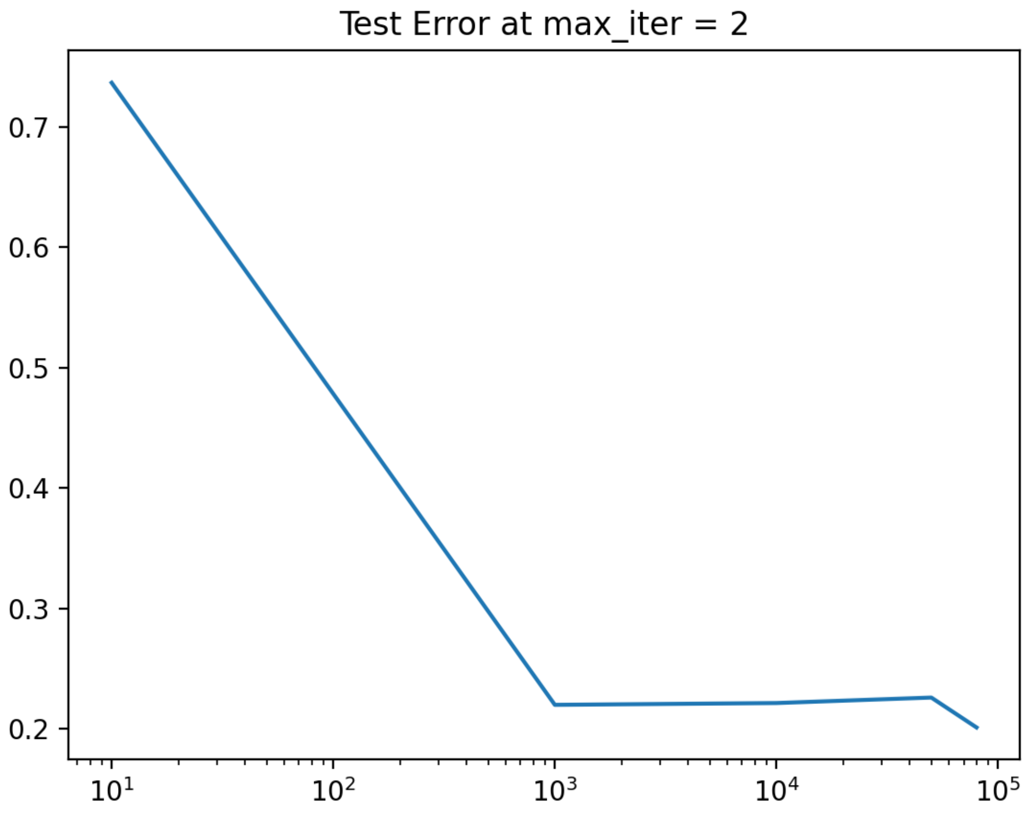
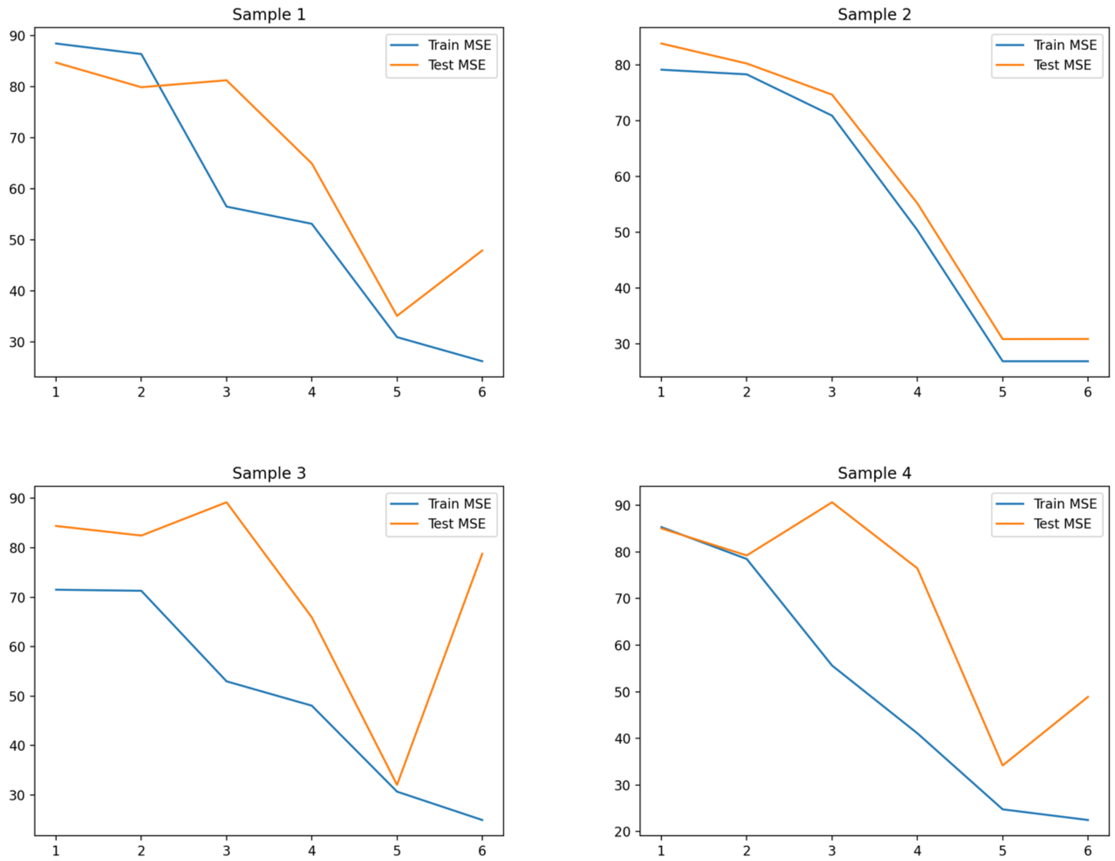


Figure 1: Test Error vs Number of Samples

The test error decreases with increase in number of samples as the bias and variance term decreases.

We observe the lowest value of train error at degree = 6 and lowest value of test error at degree = 5. This is because lambda = 6 overfits the training data and hence doesn't generalise well. Theoretically we expect degree = 6 to obtain the least train error as it has higher number of features than at other degrees.



These are the plots for MSE vs different sample datasets.
We observe the train and test error decrease till degree = 5 as bias decreases and variance increases.
After degree = 6 we observe that train error decreases but test error increases. This is because bias decreases and variance increases much more than bias.