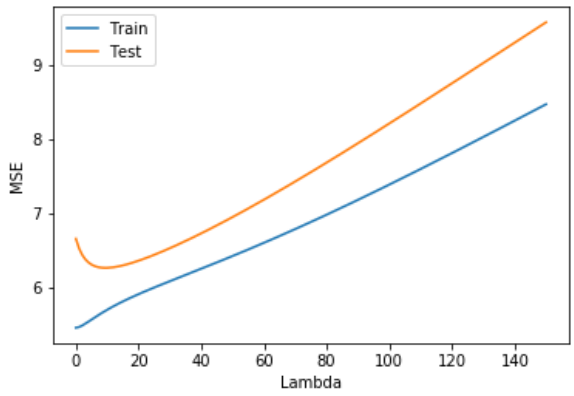
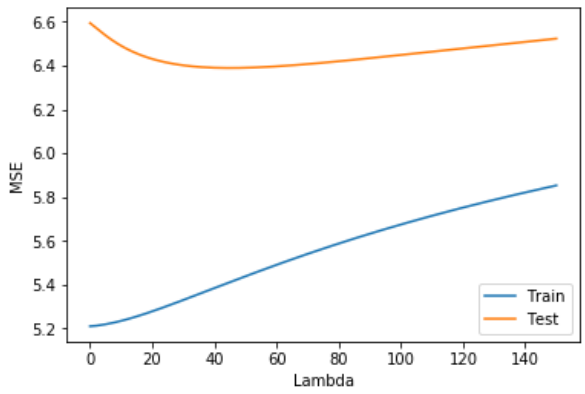
1. Implement L2 regularized linear regression algorithm with λ ranging from 0 to 150 (integers only). For each of the 6 dataset, plot both the training set MSE and the test set MSE as a function of λ (x-axis) in one graph.

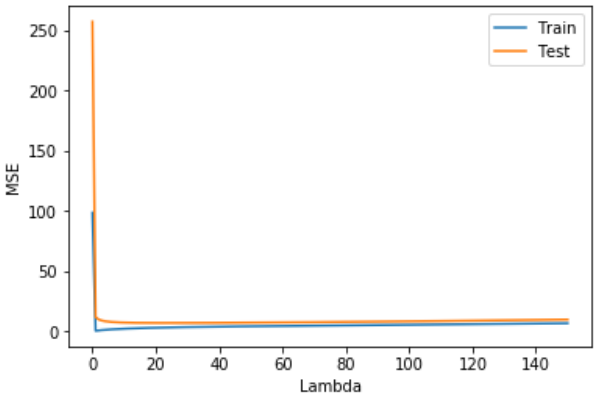
Train_100_10, Test_100_10



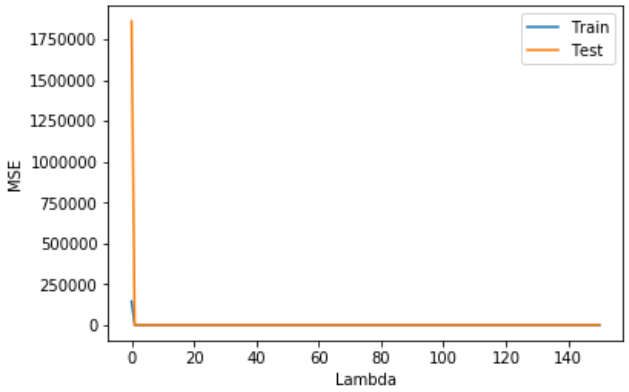
Train_1000_100, Test_1000_100



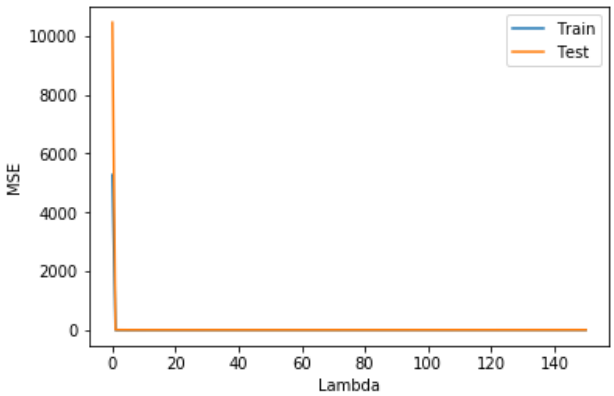
Train_100_100, Test_100_100



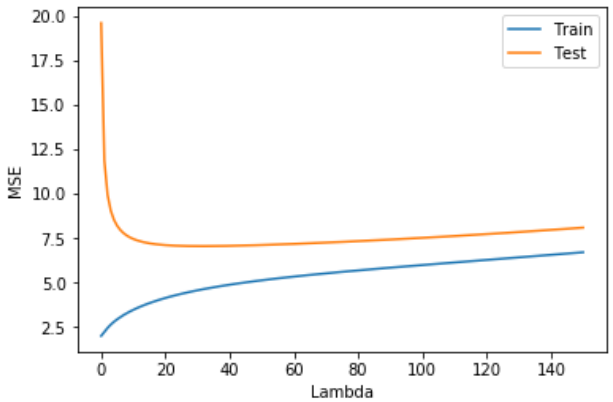
Train_50_1000_100, Test_1000_100



Train_100_1000_100, Test_1000_100



Train_150_1000_100, Test_1000_100

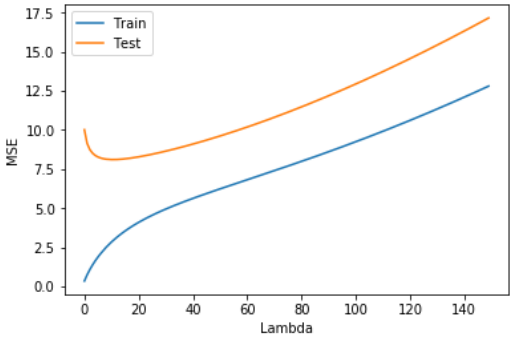


(a) For each dataset, which λ value gives the least test set MSE?

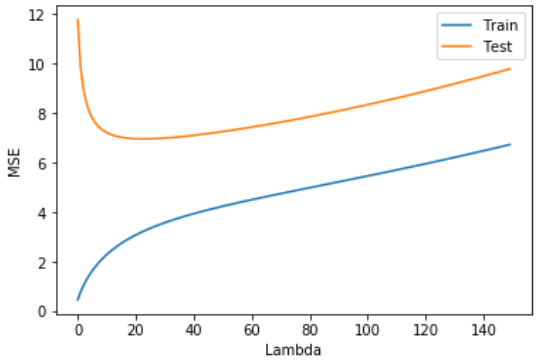
Dataset	λ	MSE
Train_100_10, Test_100_10	9	6.2580310771653576
Train_100_100, Test_100_100	24	6.9654379970459113
Train_1000_100, Test_1000_100	45	6.3891388833018778
Train_50_1000_100, Test_1000_100	12	8.1123841412191293
Train_100_1000_100, Test_1000_100	25	7.6993546513700455
Train_150_1000_100, Test_1000_100	31	7.0520239932745339

(b) For each of datasets 100-100, 50(1000)-100, 100(1000)-100, provide an additional graph with λ ranging from 1 to 150.

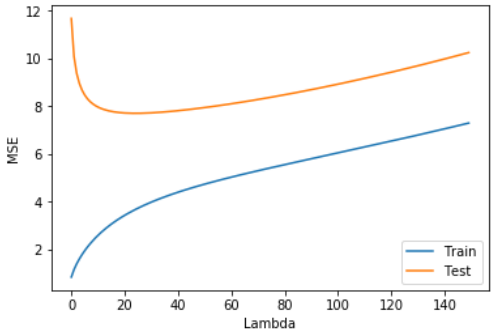
50(1000)-100



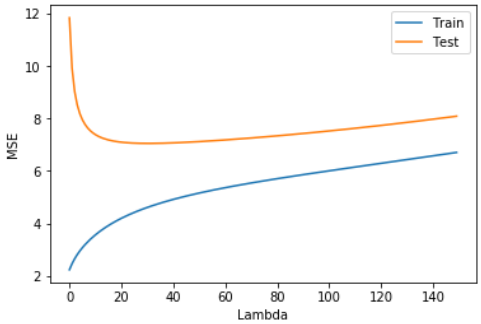
100-100



100(1000)-100



150(1000)-100



(c) Explain why λ = 0 (i.e., no regularization) gives abnormally large MSEs for those three datasets in (b).

When λ = 0, there's no penalty for model complexity. Therefore, the models can be very complex and are overfitting. As a result, the MSEs for test dataset are very large due to the fact that they don't fit them very well.

2. From the plots in question 1, we can tell which value of λ is best for each dataset once we know the test data and its labels. This is not realistic in real world applications. In this part, we use cross validation (CV) to set the value for λ. Implement the 10-fold CV technique discussed in class (pseudo code given in Appendix A) to select the best λ value from the training set.
   (a) Using CV technique, what is the best choice of λ value and the corresponding test set MSE for each of the six datasets?

Dataset	λ	MSE
train-100-10, test-100-10	11	6.2610022477825602
train-100-100, test-100-100	6	7.8088700533841786
train-1000-100, test-1000-100	45	6.3891388833018778
train-50(1000)-100, test-1000-100	15	8.1418617406333738
train-100(1000)-100, test-1000-100	13	7.8663094542669434
train-150(1000)-100, test-1000-100	51	7.1210930541252706

(b) How do the values for λ and MSE obtained from CV compare to the choice of λ and MSE in question 1(a)?

Dataset	λ(CV)	MSE(CV)	λ	MSE
train-100-10, test-100-10	11	6.2610022477825602	9	6.2580310771653576
train-100-100, test-100-100	6	7.8088700533841786	24	6.9654379970459113
train-1000-100, test-1000-100	45	6.3891388833018778	45	6.3891388833018778
train-50(1000)-100, test-1000-100	15	8.1418617406333738	12	8.1123841412191293
train-100(1000)-100, test-1000-100	13	7.8663094542669434	25	7.6993546513700455
train-150(1000)-100, test-1000-100	51	7.1210930541252706	31	7.0520239932745339

Some are larger while some are smaller. MSE and λ obtained from CV improve model accuracy and therefore is more trustworthy.

(c) What are the drawbacks of CV?

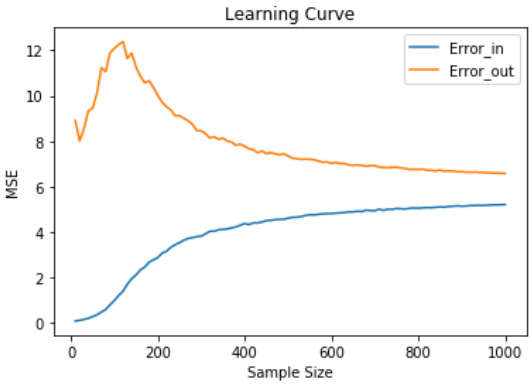
CV is computationally expensive. If the data size are large, then overfitting is less likely to happen and therefore CV will not be necessary. Also, CV can't not assign different weights to the folds. If your data spread across time, then you need to weight the folds that's closer to current time with more weights.

(d) What are the factors affecting the performance of CV?

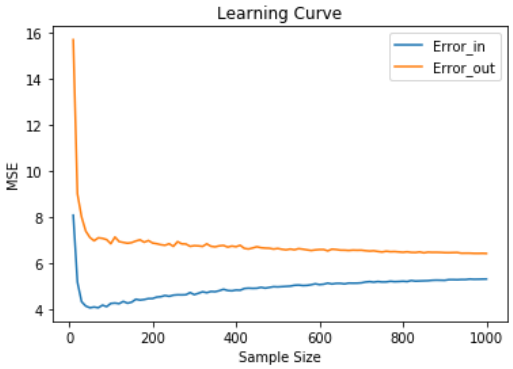
How many folds you want to split your test set in. when K is smaller, you have more bias and less variance. When K is larger, it takes more time to run and you have less bias and more variance. If you are using N folds (N is the entire data size), then the test accuracy would either be 100% or 0%.

3. Fix λ = 1, 25, 150. For each of these values, plot a learning curve for the algorithm using the dataset 1000-100.csv.

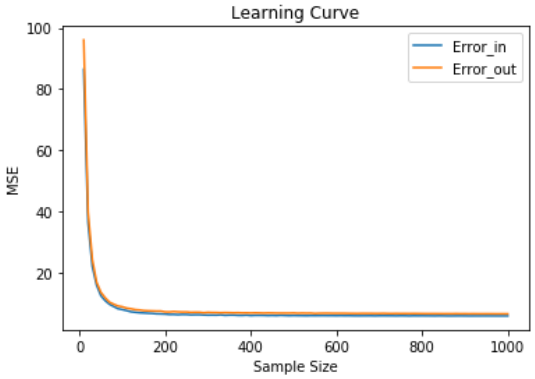
λ = 1



λ = 25



$\lambda = 150$


Learning Curve
Error_in
Error_out
MSE
Sample Size
100
80
60
40
20
0
200
400
600
800
1000