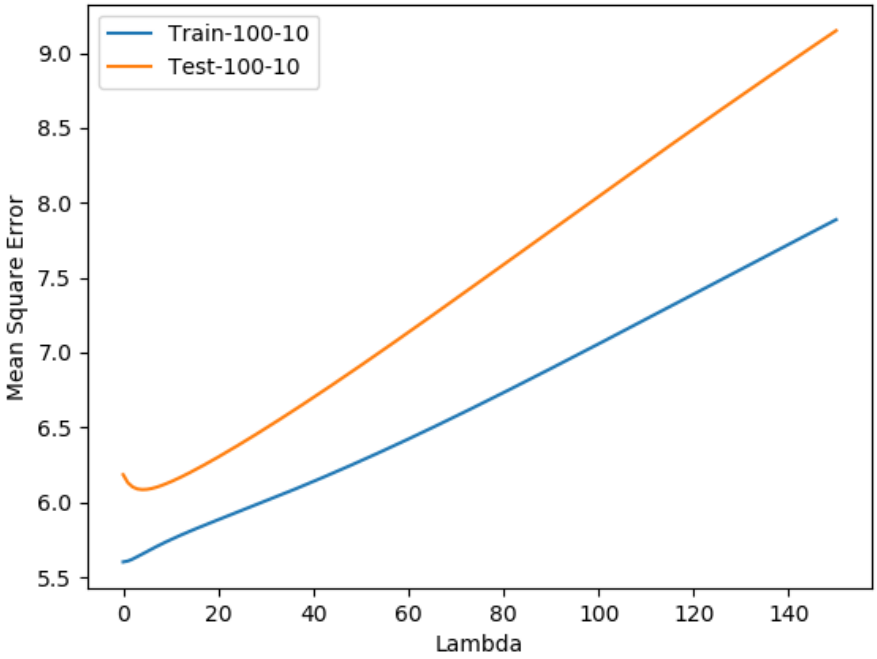
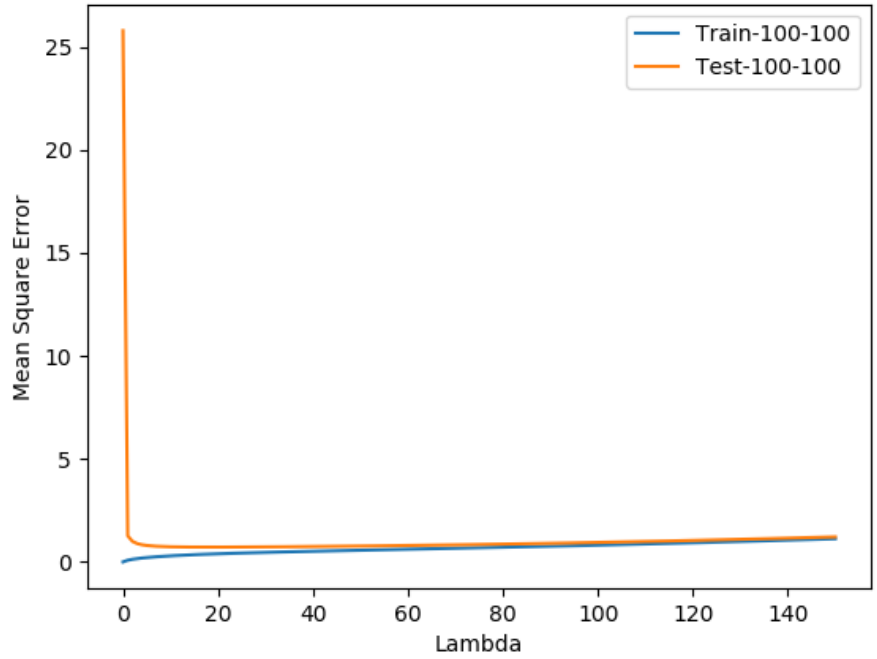
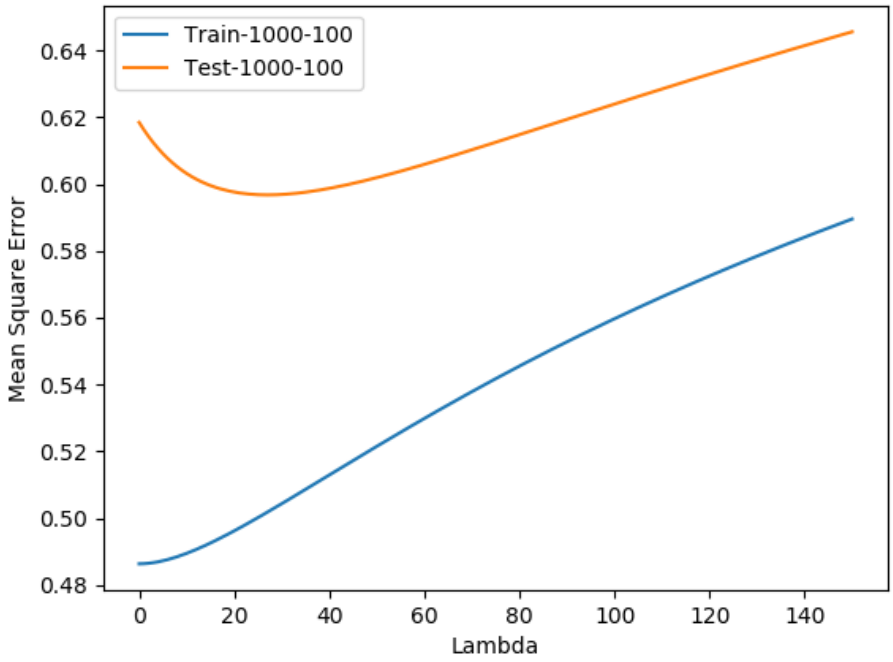
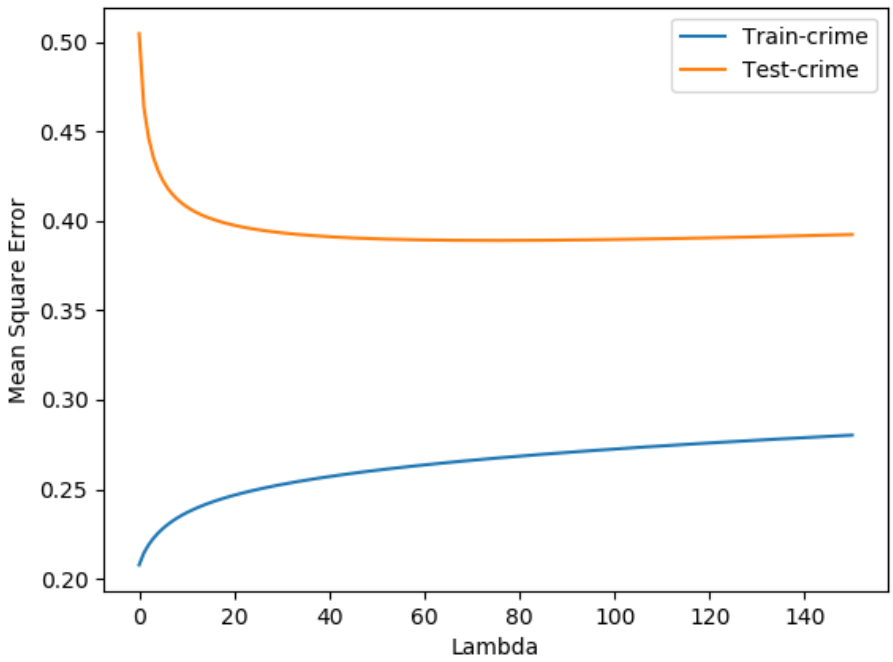
Task 1:

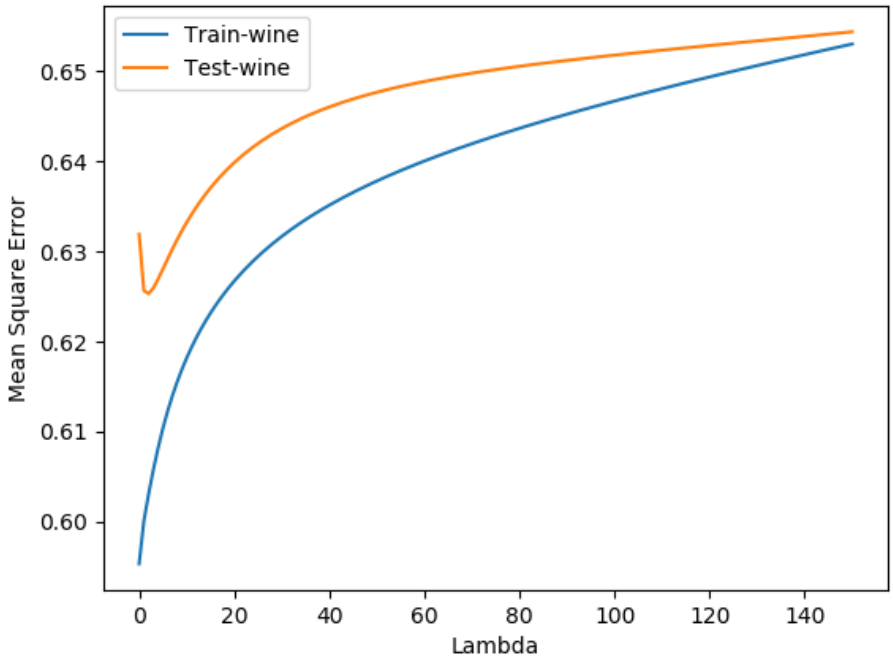
Graph for MSE against Lambda:






Train-1000-100
Test-1000-100
Mean Square Error
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0.50
0.48
0
20
40
60
80
100
120
140
Lambda

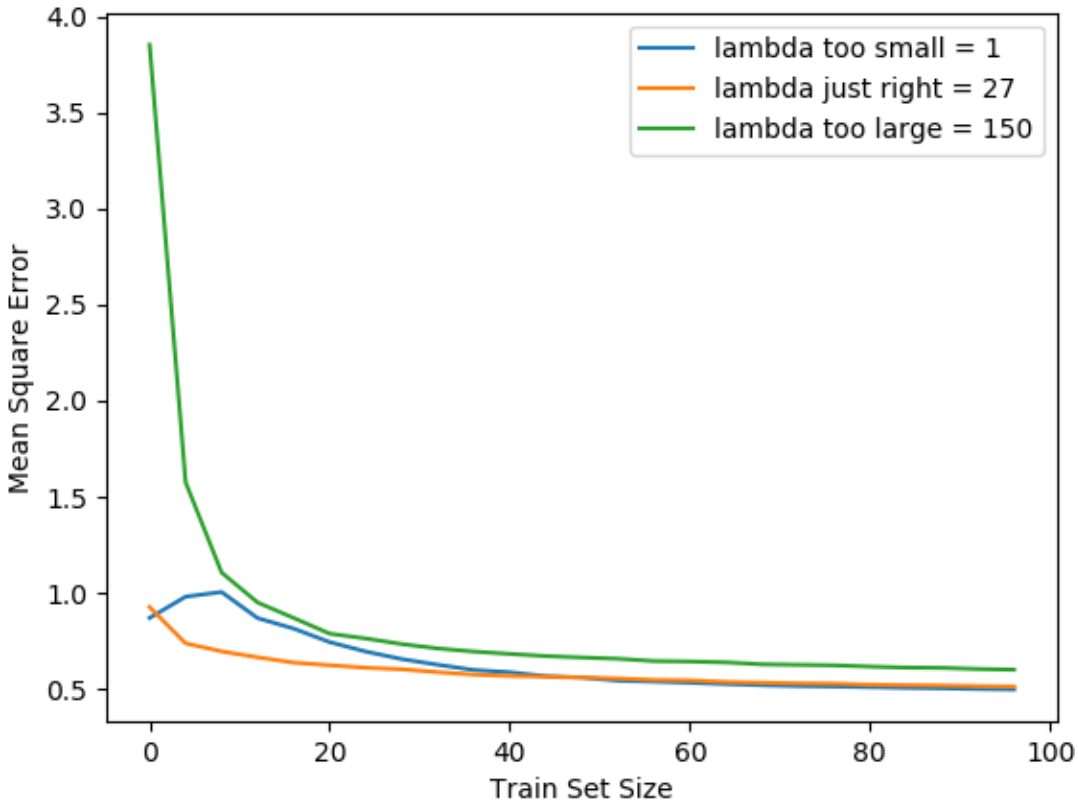

Train-crime
Test-crime
Mean Square Error
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
20
40
60
80
100
120
140
Lambda



1. The MSE of the given true functions is approximately same as MSE calculated
2. We can't use training set MSE to select lambda because MSE is calculated using hidden parameter 'w' which in turn is calculated using lambda, Lambda in training set is not a good representation for lambda needed to calculate MSE as we don't want to predict results for training set but for test set.
3. Lambda's effect on error test set shows that the curve in the above graph who have global minima will give us the best lambda whose corresponding MSE is minimum. This effect differs from dataset to dataset.
4. Variation of Lambda is dependent on dataset size, nature and features of data set.

Task 2:

Graph for combine plotting for 3 cases of Lambda:



1. I have used 4%, 8%, 12%....100% as my train set size
2. too-small = 1, just-right = 27, too-large = 150
3. Accuracy of lambda increases with data set size
4. The just-right lambda gives better and accurate results irrespective of data size
5. The variation is dependent on lambda if the lambda is too small or large the MSE increases

Task 3.1:

Table for comparing Lambda and MSE between task 1 and task 3.1:

	Task 3.1			Task 1	
	Lambda	MSE	Runtime(sec)	Lambda	MSE
Dataset -100-10	15	6.2144	0.60	4	6.0847
Dataset-100-100	19	0.7203	1.36	18	0.7202
Dataset-1000-100	23	0.5970	3.54	27	0.5967
Dataset-crime	150	0.3923	2.08	75	0.3890
Dataset-wine	2	0.6253	2.03	2	0.6253

Observation:

The MSE value for most of the datasets in task 1 and task 3.1 are approximately same and the value of lambda is also almost same for data sets except for dataset 100-10 and crime

Task 3.2:

Table for comparing Lambda and MSE between task 1 and task 3.1:

	Task 3.2					Task 1	
Dataset	Alpha	Beta	Lambda	MSE	Runtime(sec)	Lambda	MSE
100-10	1.5993	0.1569	10.1876	6.9100	0.35	4	6.0847
100-100	1.0640	75.6898	5.9762	0.7776	0.54	18	0.7202
1000-100	10.6773	1.8530	5.8894	0.6078	0.80	27	0.5967
crime	317.8763	3.3504	94.8763	0.4161	0.87	75	0.3890
Wine	5.3688	1.6145	3.3252	0.6418	1.39	2	0.6253

Observation:

The MSE value for most of the datasets in task 1 and task 3.1 are approximately same but the value of lambda differs in task 1 and task 3.2

Task 3.3:

Comparison between 3.1 and 3.2:

3.1 – Cross Validation approach
We train the model on k-1 parts of data set and test on the kth part, which is looped making the test set different for every loop. And this is repeated for n times in this case 150
3.2 – Bayesian Model approach
In this we select a lambda by converging alpha and beta value (I used alpha/beta < 10^-7 for convergence). We do not run the loop for fix n times instead we try to converge the alpha beta value.
From above table we can conclude Bayseian Model approach takes less time making it time efficient in comparison to Cross validation approach. But Cross Validation gives us not only better MSE but also lambda making it more accurate.