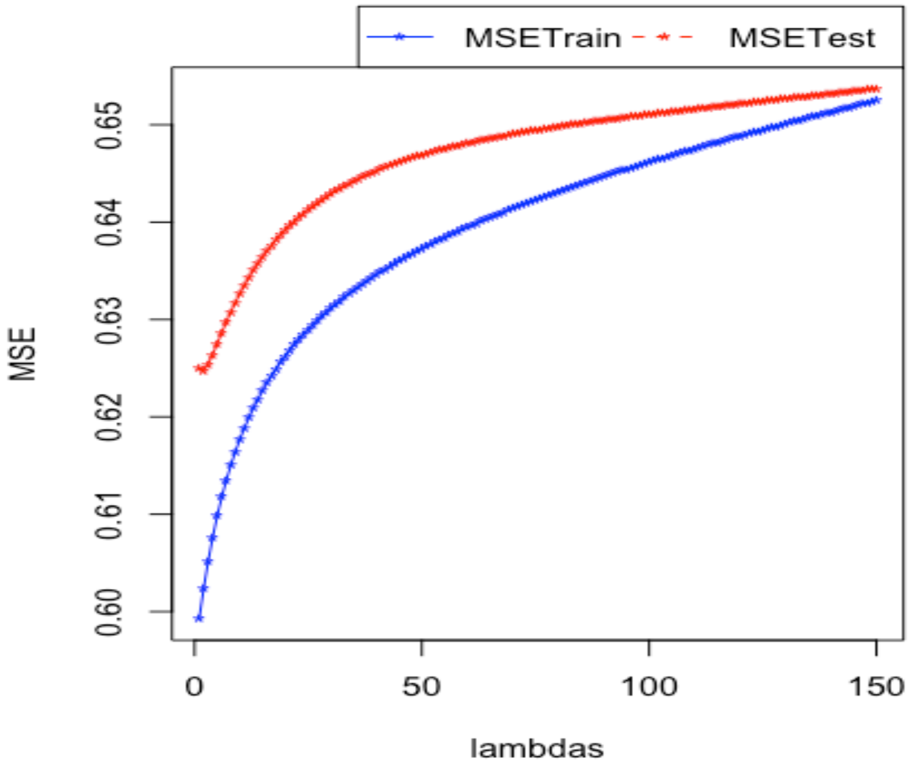
MACHINE LEARNING PROGRAMMING PROJECT 2:

This Project was completed in RStudio.
We were given 5 datasets for training and testing each.

Task 1: REGULARIZATION
We are given the equation for regularization by:

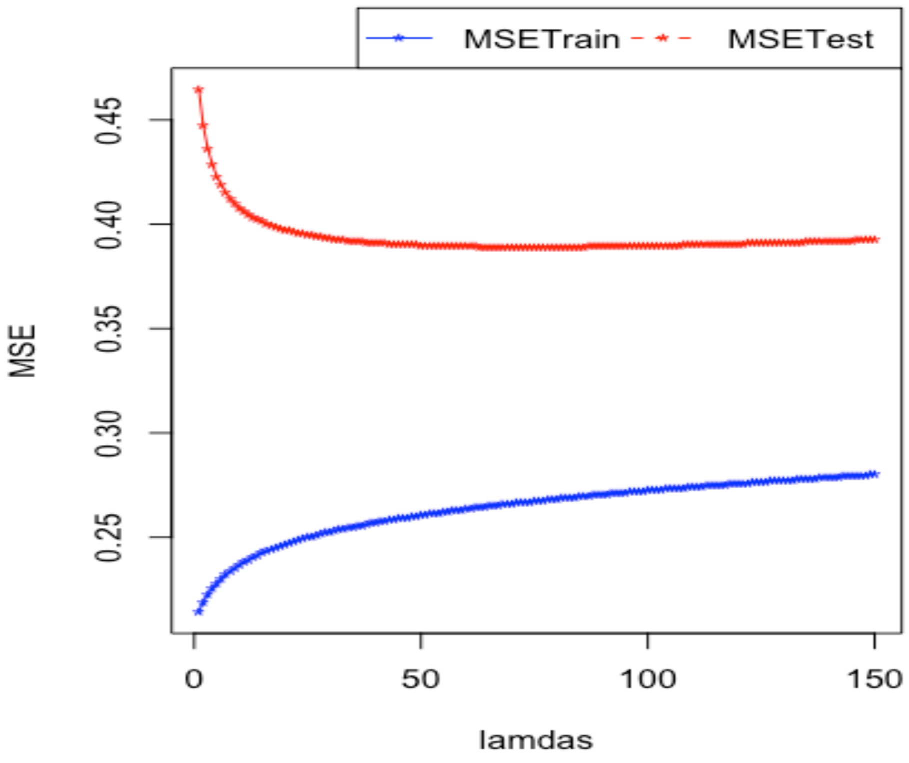
$$\mathbf{w} = \left(\lambda \mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}. \tag{3.28}$$

The main function assigned for implementing this equation on the training and testing datasets and thus calculating the Mean Squared Error(MSE) Values for each is given in the code by linear_model1().
Plot 1: Wine Dataset(MSE vs lamdas for training and testing dataset)



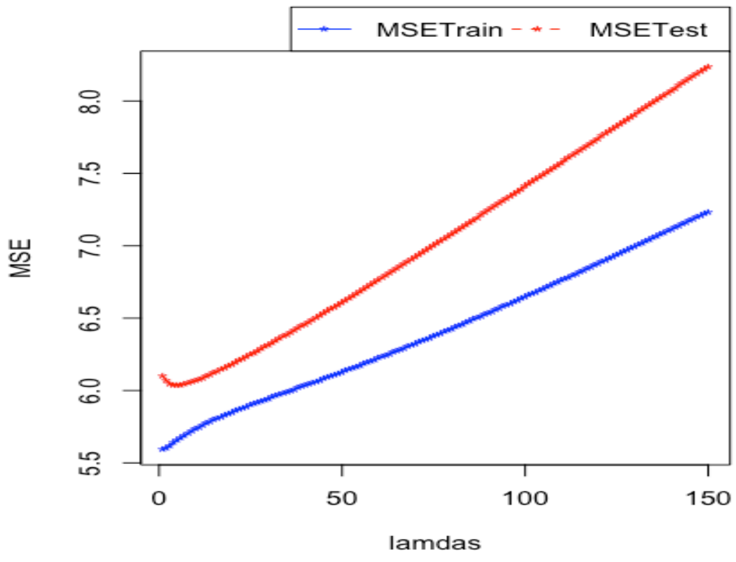
As the value of lambda increases the test set error first increases and then converges towards training set error.

Plot 2: Crime Dataset(MSE vs lamdas for training and testing dataset)



As the value of lambda increases the testing set error decreases and then becomes constant.

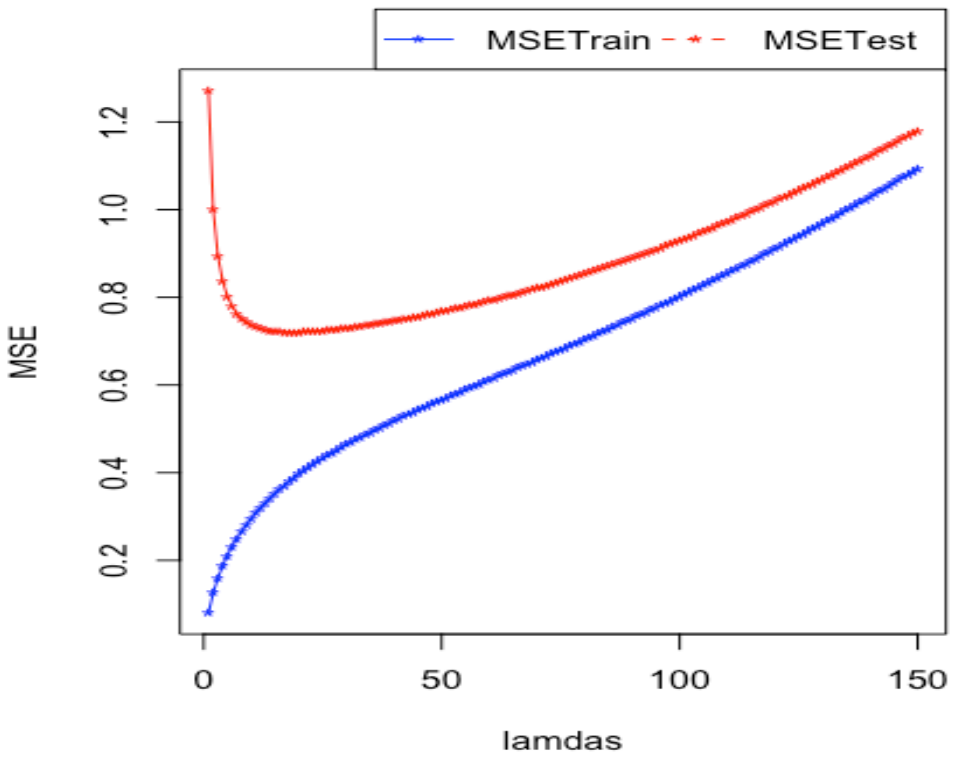
Plot 3: 100_10 Dataset(MSE vs lamdas for training and testing dataset)



MSE of Hidden True function generating data is given by: 5.714
Mean MSE for Training & Testing data for our model: 6.40 & 7.04 resp. which is comparable to MSE of Hidden True Function Generating the data

As the value of lambda increases the testing set MSE first reaches a local minima and then increases.

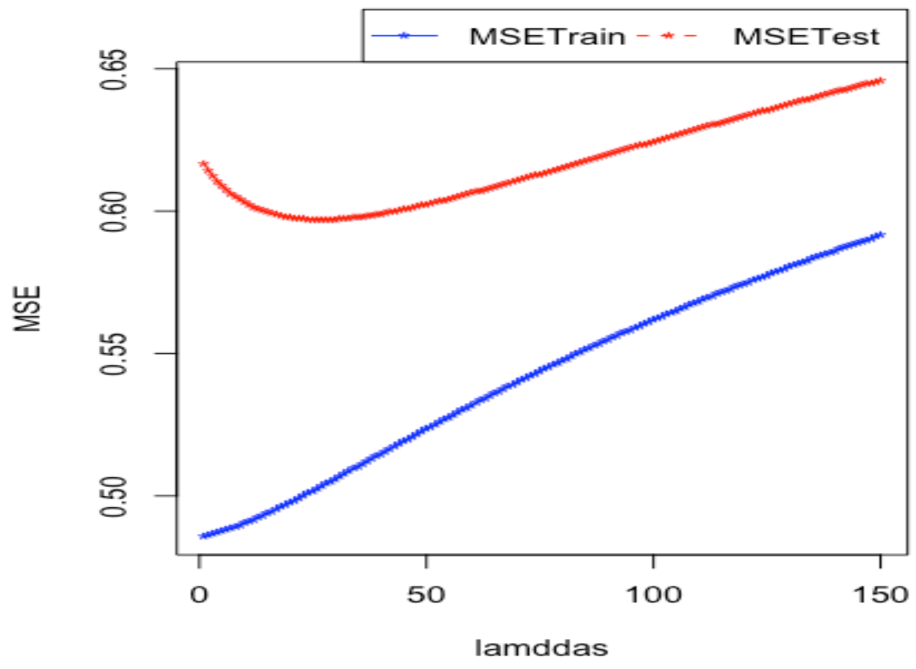
Plot 4: 100_100 Dataset(MSE vs lamdas for training and testing dataset)



MSE of Hidden True function generating data is given by: 0.533
Mean MSE for Training & Testing data for our model: 0.67 & 0.88 resp. which is slightly comparable to MSE of Hidden True Function Generating the data

As the value of lambda increases, the testing set MSE first reaches a local minima and then converges towards the training set MSE.

Plot 5: 1000_100 Dataset(MSE vs lamdas for training and testing dataset)



MSE of Hidden True function generating data is given by: 0.557
Mean MSE for Training & Testing data for our model: 0.541 & 0.61 resp. which is comparable to MSE of Hidden True Function Generating the data

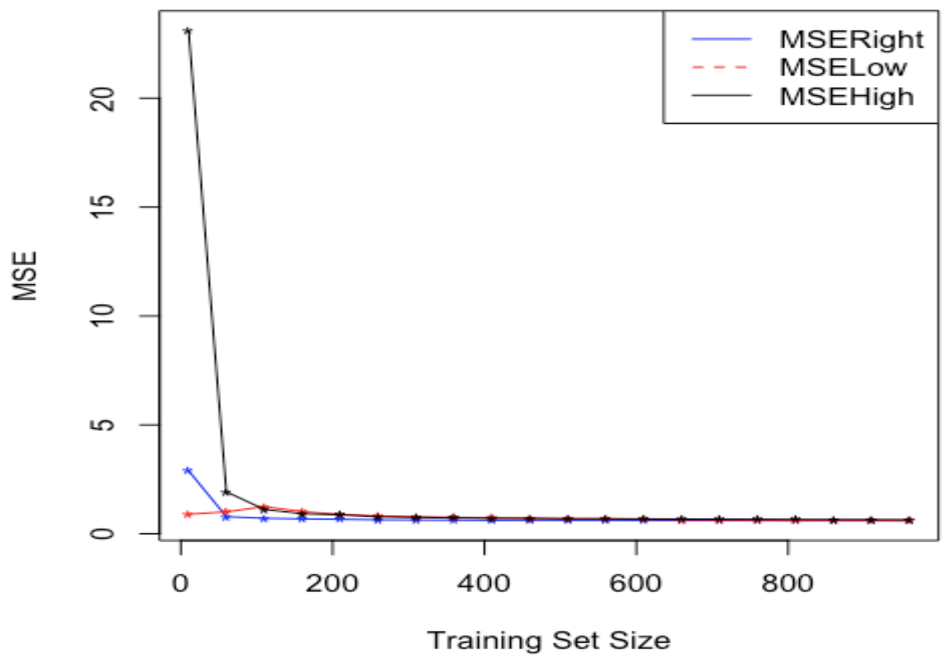
Discussion on the plots: Train sets cannot be used to select lambda because the weights are calculated using the train set. Now we have to use those weights to calculate y hat of the test set. That is how we will learn about the efficiency of our algorithm. Calculating y hat for the training dataset using the weights calculated by training set won't tell us much about the efficiency. That is why we need to select lambda based on testing set. Also as value of lambda increases the model complexity decreases and hence we see that with very large values of lambda, the MSE increases for all datasets except Crime dataset which would be very complex data in the beginning but with high regularization constant, the model complexity decreases and hence the MSE also decreases finally plateauing.

Task 2: LEARNING CURVES

The 3 representative values of lambda for the 1000-100 dataset according to Plot 5 of Task 1 are:

Just Right=27, Too Low=4, Too High=92.

We plotted the MSE for the test set of the 1000-100 dataset for these 3 values of lambda.



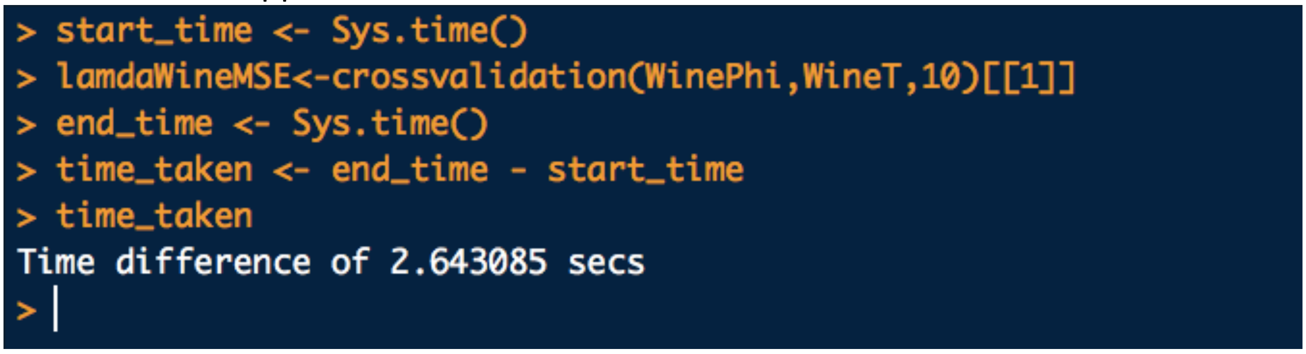
The test Set Error is least for the optimal value of lambda. Also as the training set size increases, for all values of lambdas, the MSE decreases

TASK 3.1: MODEL SELECTION USING CROSS VALIDATION

In this task I applied cross validation on the training dataset for the 5 datasets given. Split the training data into training and testing data. Calculated the weight vector for this training data and evaluated on this testing data to calculate MSE for 150 values of lambda. These are the reported value of MSE & lambdas for the given datasets:

S No	Dataset	MSE Value	Lambda	RunTime
1.	Wine	0.659	1	2.64 secs
2.	Crime	0.338	150	18.53 secs
3.	100_10	6.666	19	2.02 secs
4.	100_100	0.641	9	13.60 secs
5.	1000_10	0.586	20	33.12 secs

Here's a code snippet with results of runtime for the Wine Dataset:



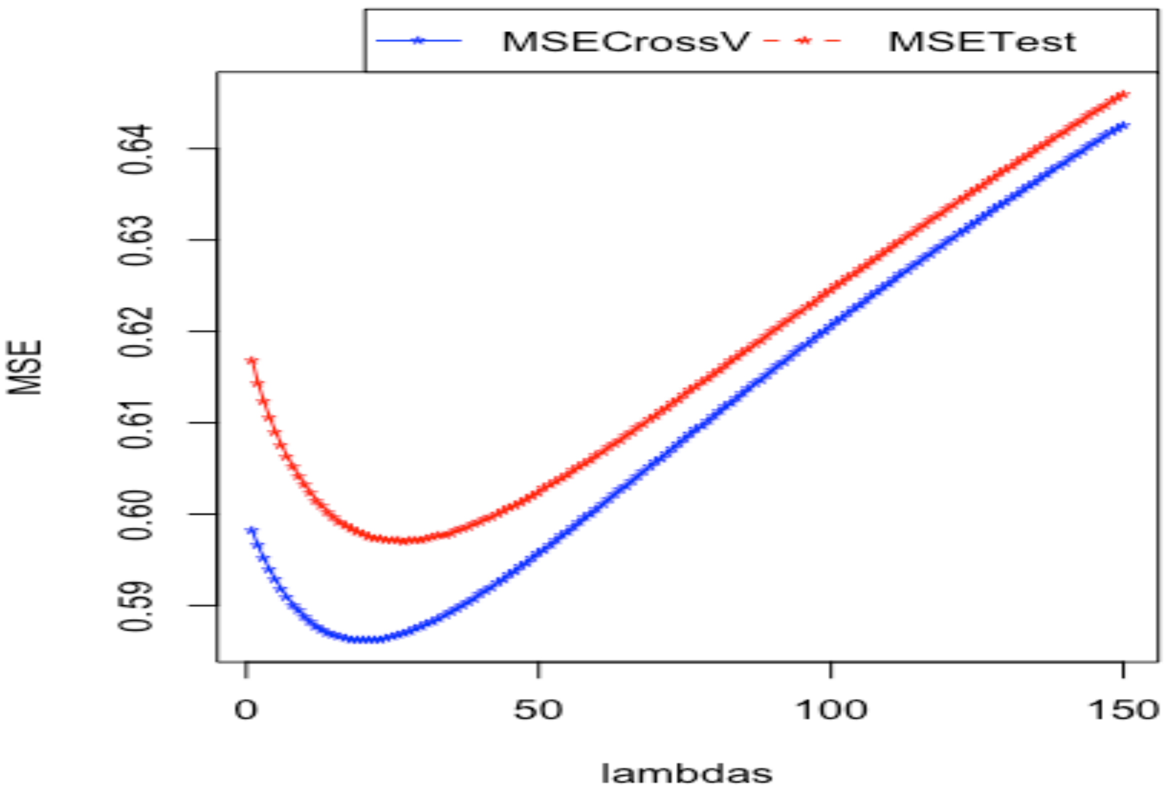
```
> start_time <- Sys.time()
> lamdaWineMSE<-crossvalidation(WinePhi,WineT,10)[[1]]
> end_time <- Sys.time()
> time_taken <- end_time - start_time
> time_taken
Time difference of 2.643085 secs
> |
```

Comparison of BEST MSE AND LAMBDA From Task 1

S No	Dataset	MSE Value	Lambda
1.	Wine	0.624	2
2.	Crime	0.389	74
3.	100_10	6.039	5
4.	100_100	0.720	19
5.	1000_10	0.597	27

We can see that the Cross Validation MSE Values at the top outperform MSE Values from Task 1 for the test datasets with the exception of the Wine and 100_100 datasets. Also the values of lambda differ with the least MSE value for the 1000_100 dataset.

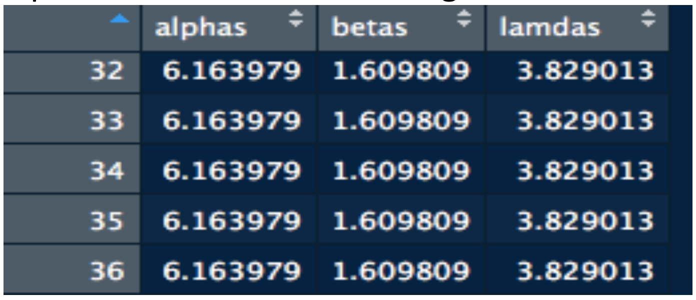
Here is a graph comparing the cross validation test set MSE to the test set MSE in task 1 for the Thousand_100 dataset clearly showing it's doing better.



TASK 3.2: BAYESIAN MODEL SELECTION:

Here we had to Bayesian linear regression on the training data of all the 5 datasets given and calculate value of alpha,beta and hence lambda=alpha/beta. We also calculated the value of Mn for each dataset which was then used as weight vector on the test datasets along with optimum value of lambda obtained from above to calculate MSE for each dataset. The main function developed for these tasks were Bayesian_function() and linear_model().

Here's a printable result from the code for the wine dataset for the value of lambda alpha and beta which converges after more than 30 iterations.



	alphas	betas	lamdas
32	6.163979	1.609809	3.829013
33	6.163979	1.609809	3.829013
34	6.163979	1.609809	3.829013
35	6.163979	1.609809	3.829013
36	6.163979	1.609809	3.829013

These are the reported values of alpha,beta lambda, MSE & Runtime of all the 5 datasets using Bayesian Modelling.

Sno	Dataset	Alpha	Beta	Lambda	MSE	Run-Time
1	Wine	6.16	1.61	3.83	0.627	0.27 secs
2	Crime	425.64	3.25	130.95	0.391	1.46 secs
3	100_10	0.88	0.16	5.34	6.09	0.16 secs
4	100_100	5.15	3.15	1.63	1.063	1.08 secs
5	1000_100	10.29	1.86	5.53	0.608	2.90 secs

TASK 3.3: Comparison:

The MSE Value with the method of Cross Validation is better off as compared to Bayesian Modelling. However, comparing them on runtime, Bayesian modelling fairs better than anyone.