PROGRAMMING PROJECT REPORT

This project captures the distinction between vanilla linear regression, regularized linear regression, and bayesian linear regression. Here, we conduct two sets of experiments namely two tasks. In the first task, we compare the linear regression and bayesian linear regression for two datasets: CRIME and HOUSING. In the second dataset, we take a third and five-degree polynomial function and then apply both of the methods to see the performance of both of these algorithms. Below, the results are shared for two tasks.

TASK-1

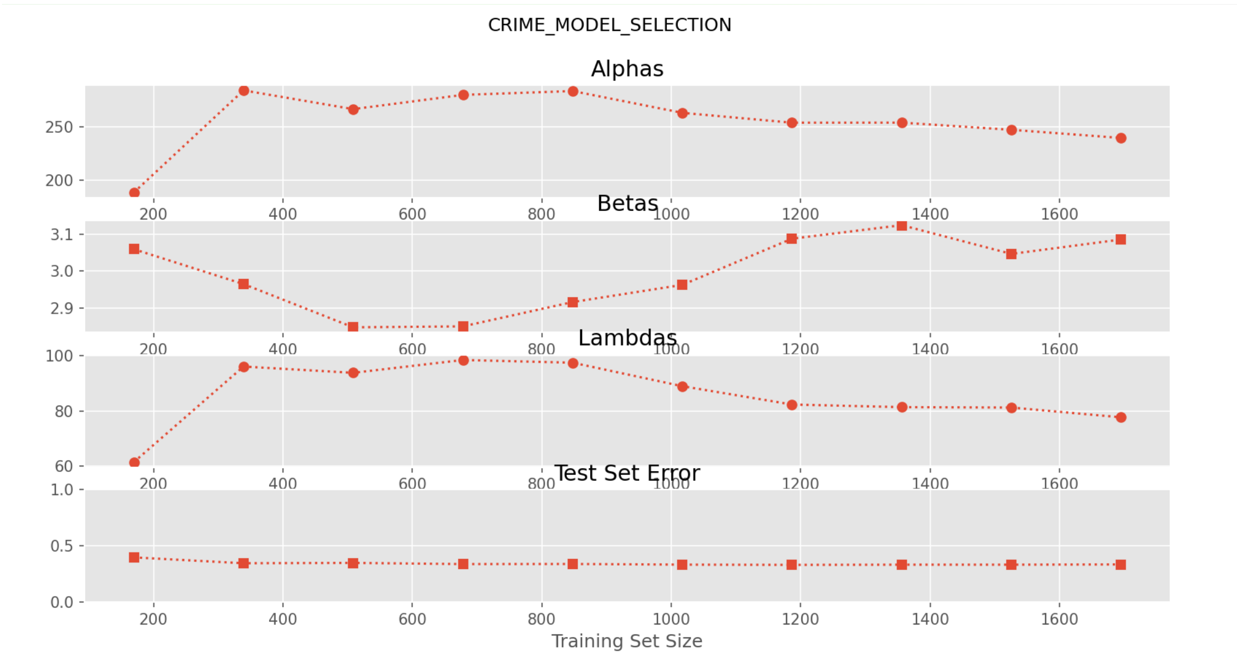
QUESTION -1

- We ran the model selection algorithm using bayesian linear regression and below are the alpha and beta values. (Note alpha was initialized with value 5.0 and beta was initialized with value 1.0)
- For CRIME DATASET

```
For CRIME DATA
For train set size of 169 the alpha is 188.13442292780064
For train set size of 169 the beta is 3.059060286961196
For train set size of 169 the lambda is 61.5007241699997
-----------------------------------------------------------
For train set size of 339 the alpha is 284.72211279203196
For train set size of 339 the beta is 2.9644187319414015
For train set size of 339 the lambda is 96.04652329448987
```

```
----------------------------------------------------------
For train set size of 508 the alpha is 266.9905318051459
For train set size of 508 the beta is 2.847152362092656
For train set size of 508 the lambda is 93.77458521688953
----------------------------------------------------------
For train set size of 678 the alpha is 280.44906045127516
For train set size of 678 the beta is 2.849684394927586
For train set size of 678 the lambda is 98.4140773450113
----------------------------------------------------------
For train set size of 847 the alpha is 284.0879130173372
For train set size of 847 the beta is 2.9156220186722694
For train set size of 847 the lambda is 97.43646851271434
----------------------------------------------------------
For train set size of 1017 the alpha is 263.47594593781855
For train set size of 1017 the beta is 2.962261500501463
For train set size of 1017 the lambda is 88.94418871973872
----------------------------------------------------------
For train set size of 1186 the alpha is 254.16211846839803
For train set size of 1186 the beta is 3.0875608190601413
For train set size of 1186 the lambda is 82.31809294230045
----------------------------------------------------------
For train set size of 1356 the alpha is 254.2072308418734
For train set size of 1356 the beta is 3.124195868816011
For train set size of 1356 the lambda is 81.36725145155874
----------------------------------------------------------
For train set size of 1525 the alpha is 247.4378221546035
For train set size of 1525 the beta is 3.0462555187934286
For train set size of 1525 the lambda is 81.2268769405823
----------------------------------------------------------
For train set size of 1695 the alpha is 239.7167113559968
```

For train set size of 1695 the beta is 3.085298058113479
For train set size of 1695 the lambda is 77.69645163636983



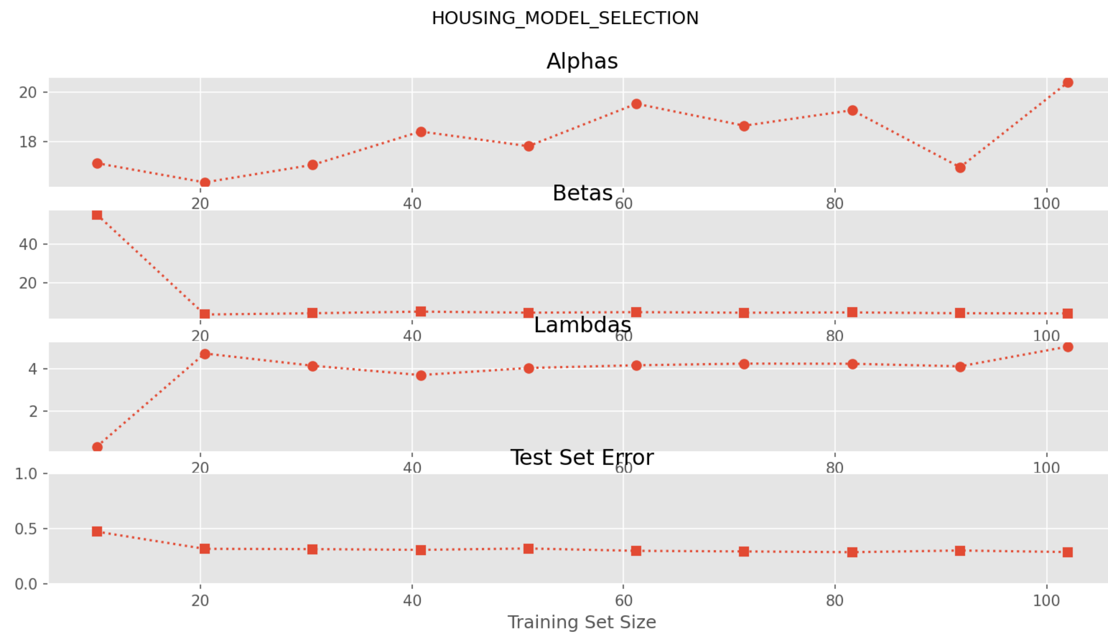
- NOW FOR THE HOUSING DATASET:

For train set size of 10 the alpha is (17.11589411580974+5.443301597589448e-30j)
For train set size of 10 the beta is (55.429956856019174-4.928386388392230 5e-29j)
For train set size of 10 the lambda is (0.3087841861444 84+3.727475280327282e-31j)

For train set size of 20 the alpha is 16.334757054363713
For train set size of 20 the beta is 3.458601661454 6717
For train set size of 20 the lambda is 4.722936797379954

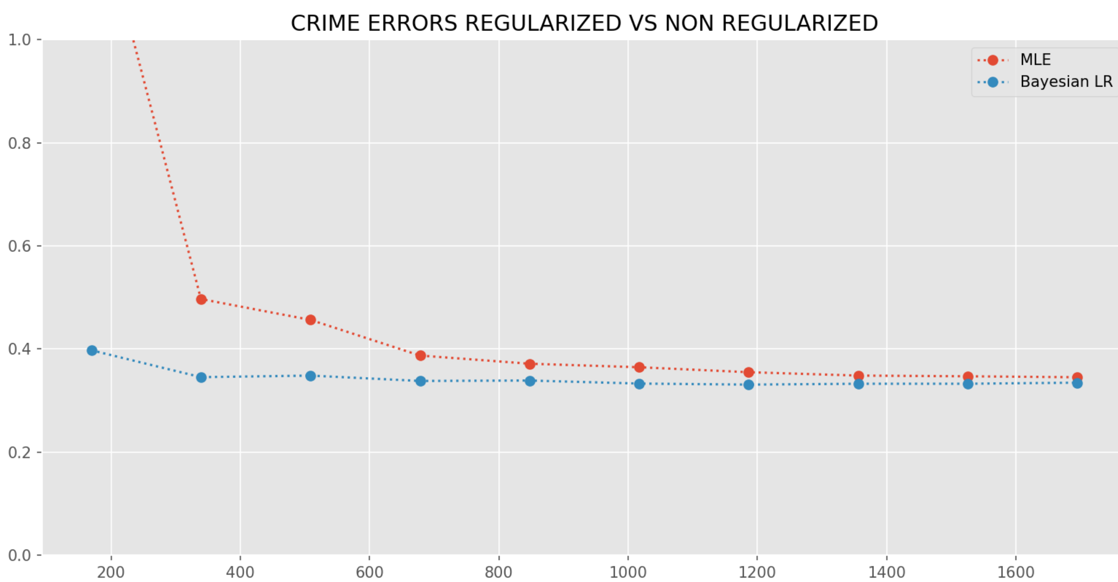
For train set size of 30 the alpha is 17.047127873991485
For train set size of 30 the beta is 4.11444616640 3642
For train set size of 30 the lambda is 4.143237554835248

```
----------------------------------------------------------
For train set size of 40 the alpha is 18.406163735922693
For train set size of 40 the beta is 4.974602720759276
For train set size of 40 the lambda is 3.7000268703092236
----------------------------------------------------------
For train set size of 51 the alpha is 17.808745250198914
For train set size of 51 the beta is 4.411517216877257
For train set size of 51 the lambda is 4.0368753820267385
----------------------------------------------------------
For train set size of 61 the alpha is 19.53649930959937
For train set size of 61 the beta is 4.691273568977769
For train set size of 61 the lambda is 4.164434033177986
----------------------------------------------------------
For train set size of 71 the alpha is 18.64262925093142
For train set size of 71 the beta is 4.393496838777713
For train set size of 71 the lambda is 4.243232653859805
----------------------------------------------------------
For train set size of 81 the alpha is 19.273259592676656
For train set size of 81 the beta is 4.549845214265449
For train set size of 81 the lambda is 4.236025333839457
----------------------------------------------------------
For train set size of 91 the alpha is 16.950879335243208
For train set size of 91 the beta is 4.11960028724407
For train set size of 91 the lambda is 4.114690298408298
----------------------------------------------------------
For train set size of 102 the alpha is 20.41249676487572
For train set size of 102 the beta is 4.041254850735992
For train set size of 102 the lambda is 5.051029325992198
```



QUESTION-2

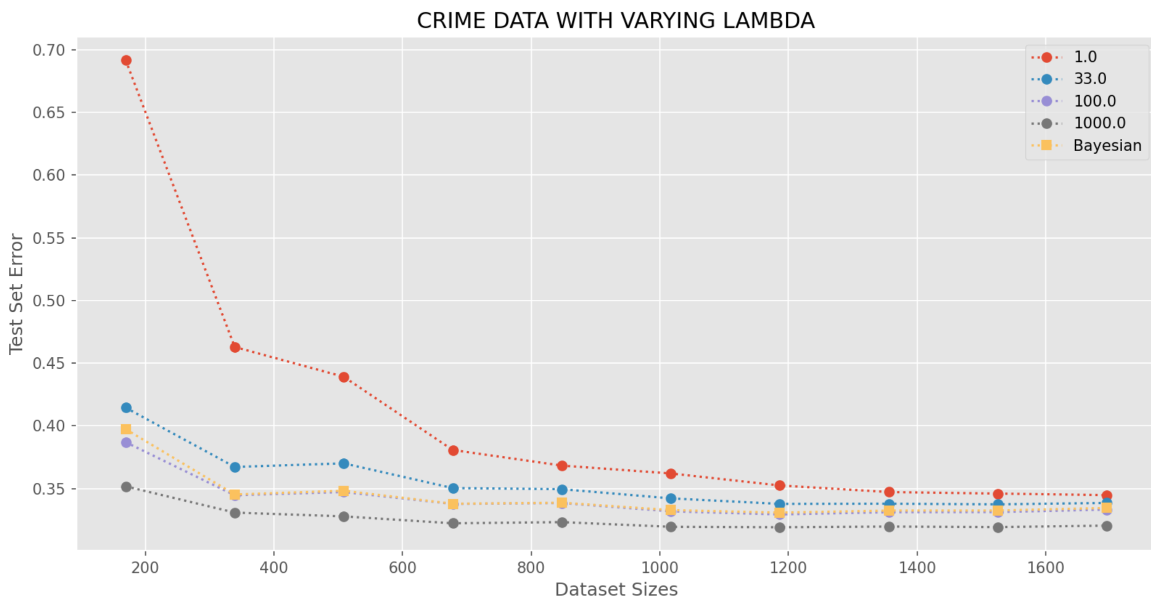
- Here we have to compare the test set error between the non-regularized linear regression and bayesian linear regression. We got the following plots for CRIME and HOUSING

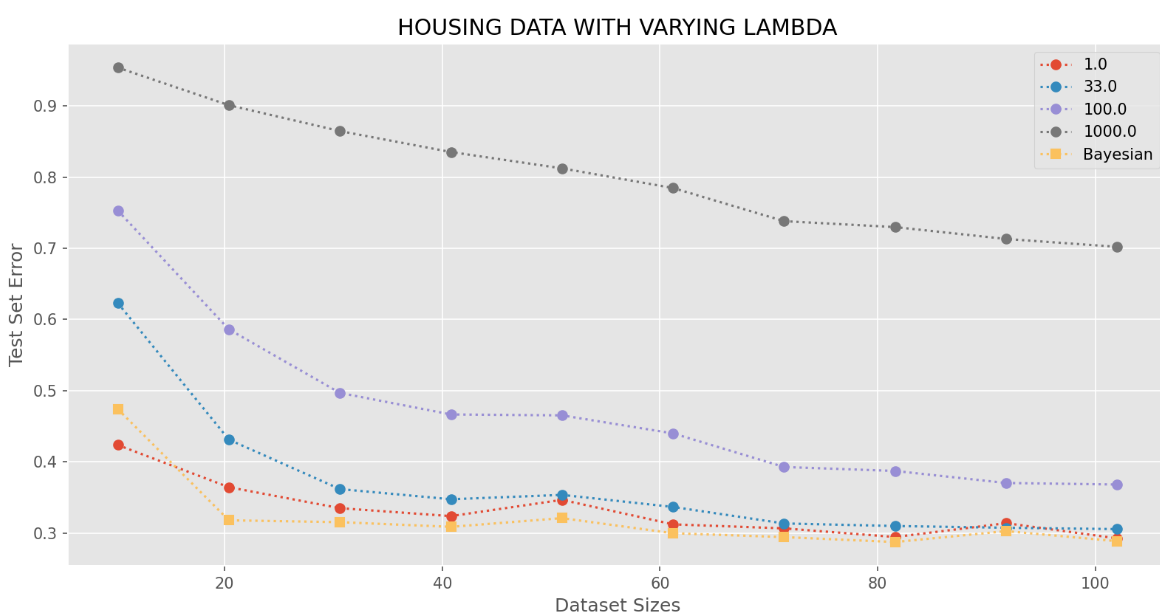


- Note that we have limited the y-axis from [0-1]. We can see the relative performance between these two algorithms. The Bayesian Linear Regression is clearly superior to the non-regularized MLE linear regression. For both of the datasets, bayesian linear regression performs better. The bayesian LR iteratively optimizes to the optimal solution, whereas the MLE without any regularization parameters greedily optimizes. So for a smaller dataset, the MLE model tends to overfit, thus performing poorly on the test set. The Bayesian model due to regularization does not overfit the training data and it performs better on the testing dataset. **But as the dataset size increases, their performance converges in both datasets. This suggests that regularization is not substantially favorable in larger datasets, but for smaller datasets, it fares well. So regularization is favorable when we have a small or a skewed dataset, else there performance converges with MLE as the dataset size increases.**

QUESTION-3

- Here we have to compare different values of lambdas for MLE and the bayesian algorithm. The plots for the question are the following for CRIME and HOUSING.

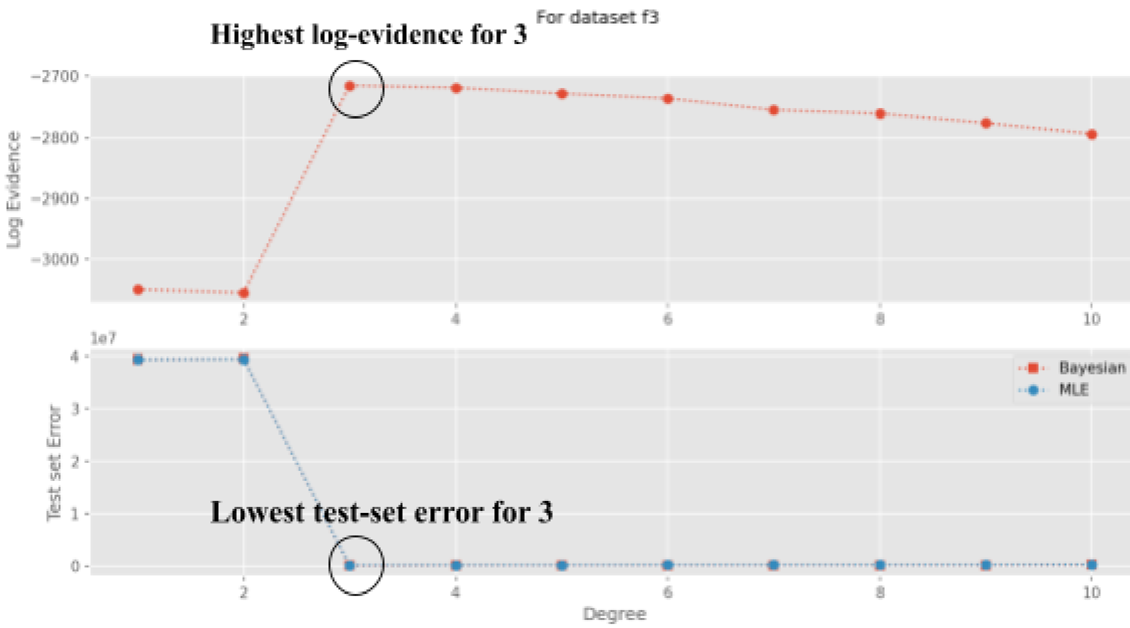




- We can see that the lambda value is specific to the dataset rather than being universal. For the CRIME dataset, the optimal lambda value is 1000.0 and for the HOUSING dataset, it is 1.0. So we cannot pick a universal lambda for all datasets, only after testing it on our testing dataset, we can conclude which lambda performs better. The Bayesian algorithm performed reasonably well and it heuristically works better. For MLE, we may have to go for a brute-force solution of trying out every lambda, but for bayesian, we get very good results without any brute-force solutions. So the bayesian algorithm converges well and it was able to pick relevant lambdas specific to the dataset. For hyperparameter optimization of lambda, one could take a brute force approach of trying out all possible values with a certain range and then test it on the test set. The lambda value that gives the least test set error is the superior one and can be used for further inference.

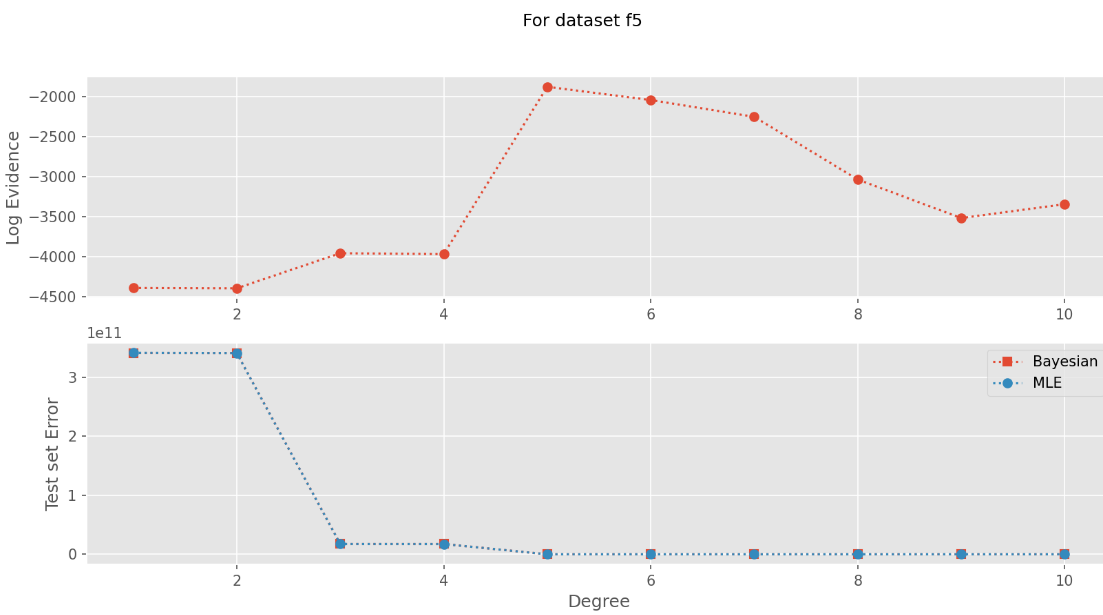
TASK-2

- Here we have the hyperparameter as the polynomial degree and we have to show that the bayesian algorithm can find the optimal hyperparameter using the log evidence metric. **Note that the test set error for both MLE and Bayesian algorithms are nearly the same as we have the entire training dataset.**
- **Now for the F3 dataset**



- **We can see that the log evidence for the third degree is the highest and so is the test set error is the lowest for the third degree. So the bayesian algorithm using log evidence was able to pick an optimal degree hyperparameter as F3 means third-degree polynomial.**

SIMILARLY for F5



- Here we can see that the log evidence is highest for degree 5 and the test set error is lowest for degree 5. So log evidence successfully optimized the degree hyperparameter and we don’t need the test set to optimize the hyperparameter, we can do it from the training set only.
- As discussed earlier, with the increase in dataset size, the non-regularized model performed nearly equal to the bayesian model as it did not overfit and the dataset was not unbalanced. So their performance converged.