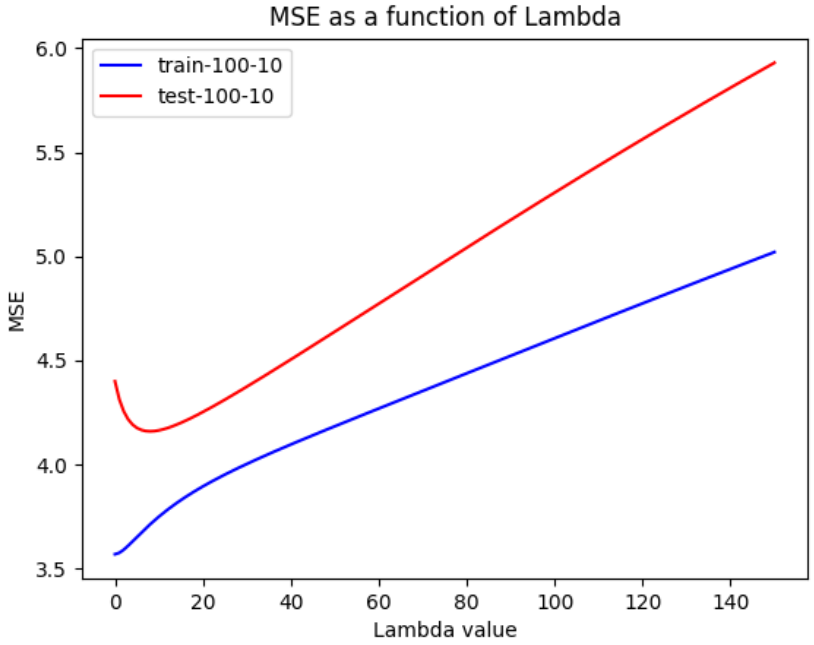
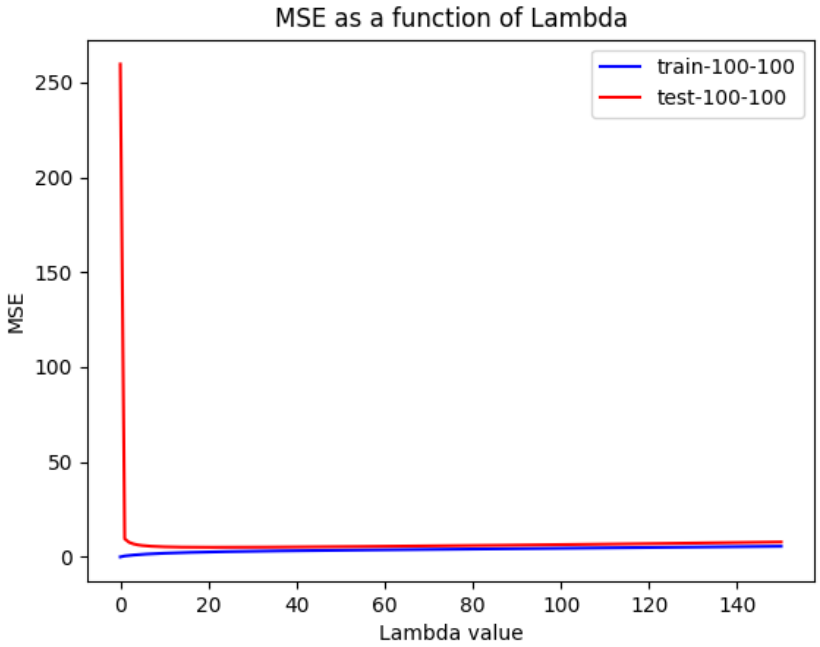
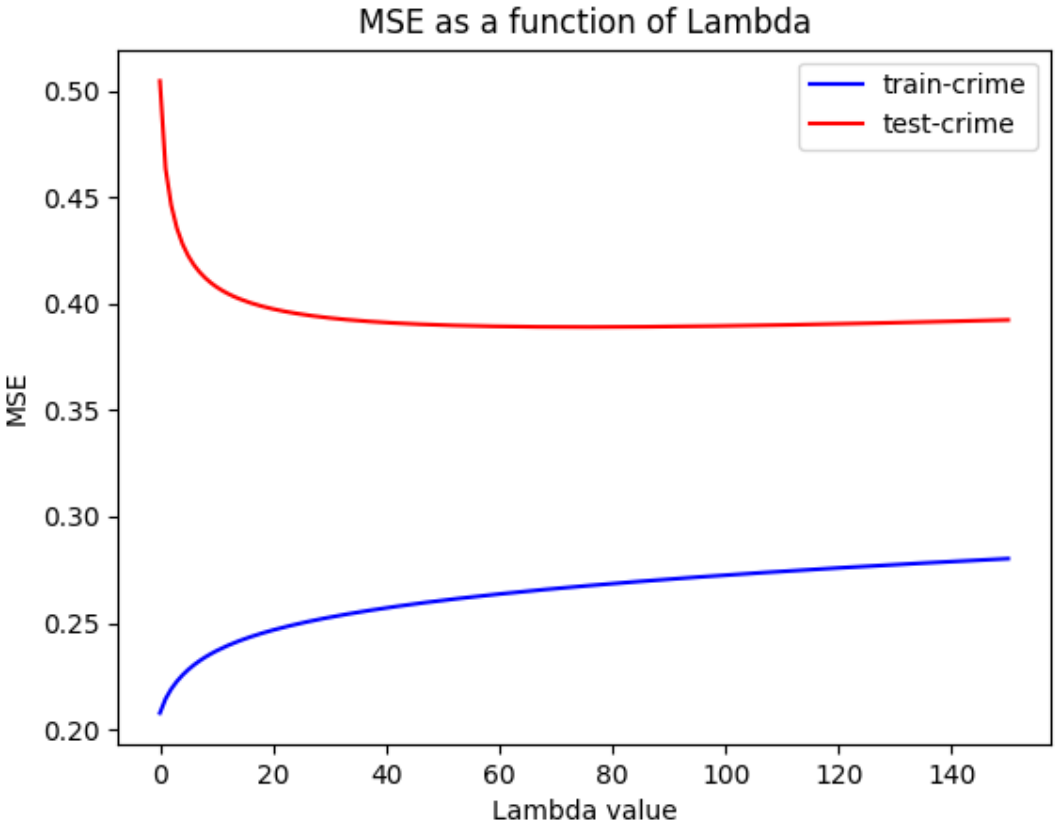
PROGRAMMING PROJECT 2

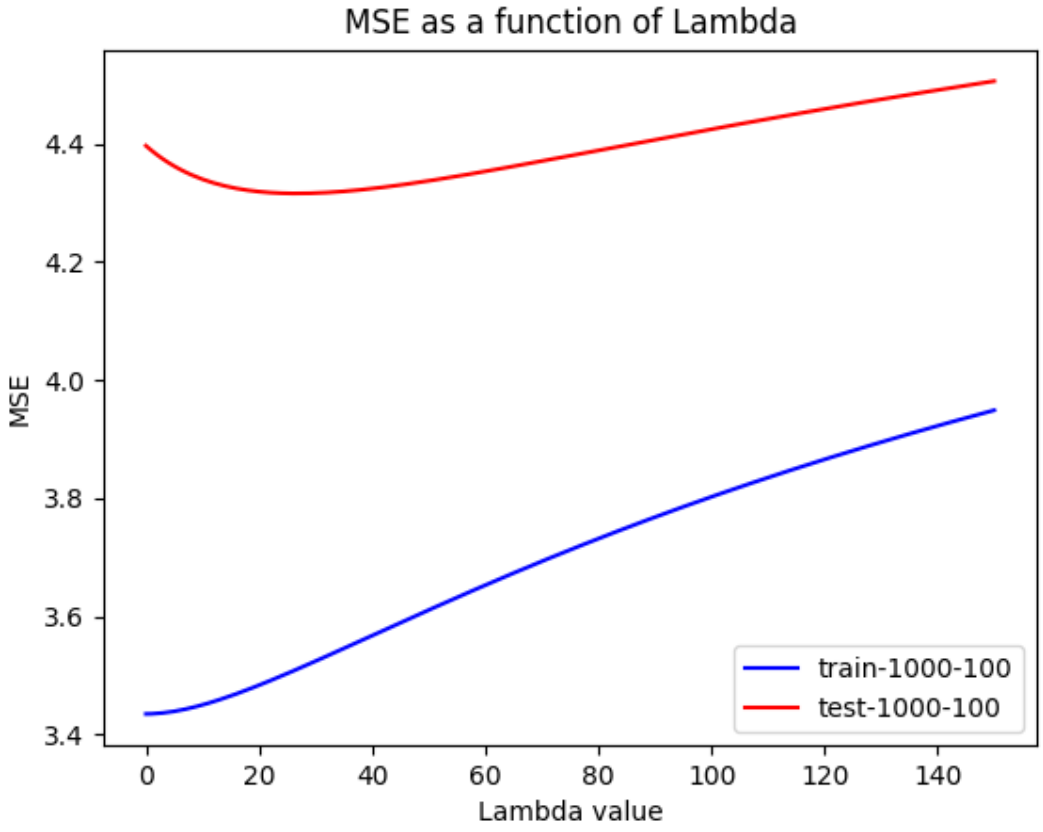
TASK 1:

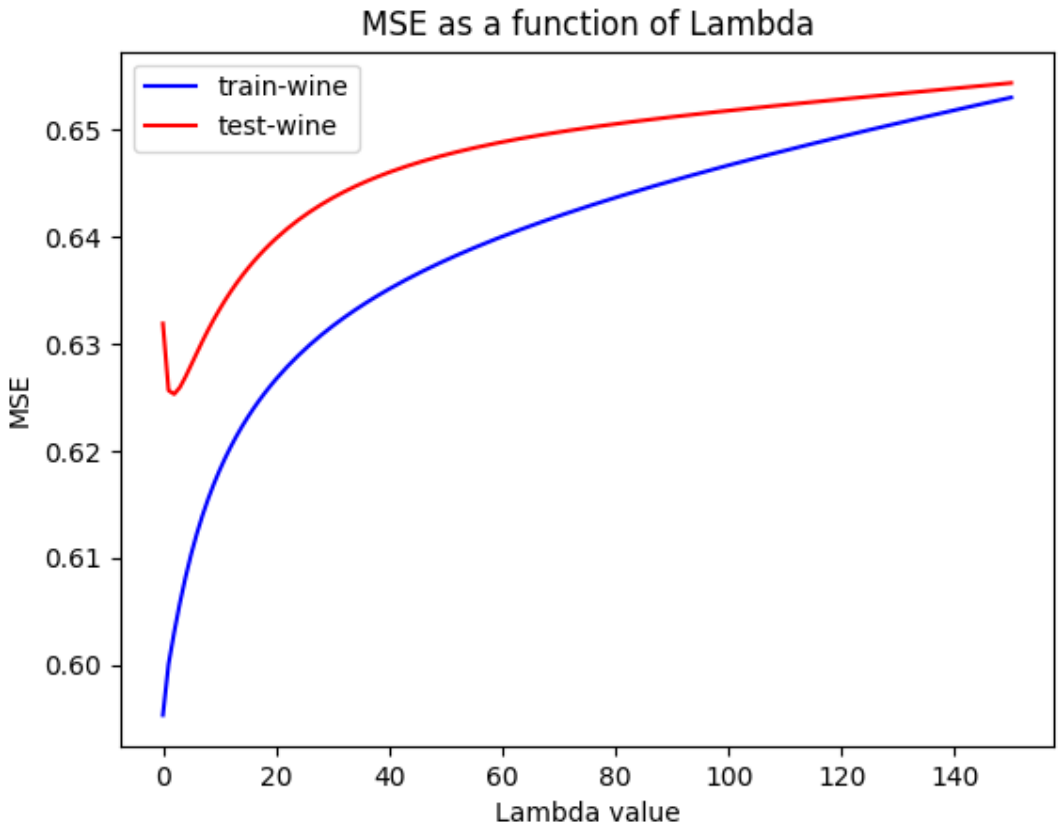
From the following plots we can obtain the results of doing linear regression on the 5 datasets for task 1. It was also mentioned that the MSE for the true function for 100-10 was 3.78, 100-100 was 3.78 and 1000-100 was 4.015. This can serve as a reference while verifying the results.






MSE as a function of Lambda
train-crime
test-crime
MSE
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
20
40
60
80
100
120
140
Lambda value


MSE as a function of Lambda
MSE
4.4
4.2
4.0
3.8
3.6
3.4
train-1000-100
test-1000-100
0
20
40
60
80
100
120
140
Lambda value



Q. Why can't the training set MSE be used to select λ?

From the plots, we can clearly see that the MSE increases with the lambda, the regularization parameter. This is however observed only after λ=0. At λ=0 our algorithm gets the best possible answer with the lowest MSE. But then there is no regularization as for λ=0, we have the value of regularization parameter as 0 and thus we are in fact doing unregularized linear regression. And as we can see the test set MSE is the highest for λ=0. This is due to overfitting the model to the train set. This means that we can't solely use the training set performance to choose the λ as even though it gives the lowest MSE for λ=0, this model would fare badly in terms of test case dataset. Regularization addresses the problem of overfitting on the training data.

Q. How does λ affect the error on the test set?

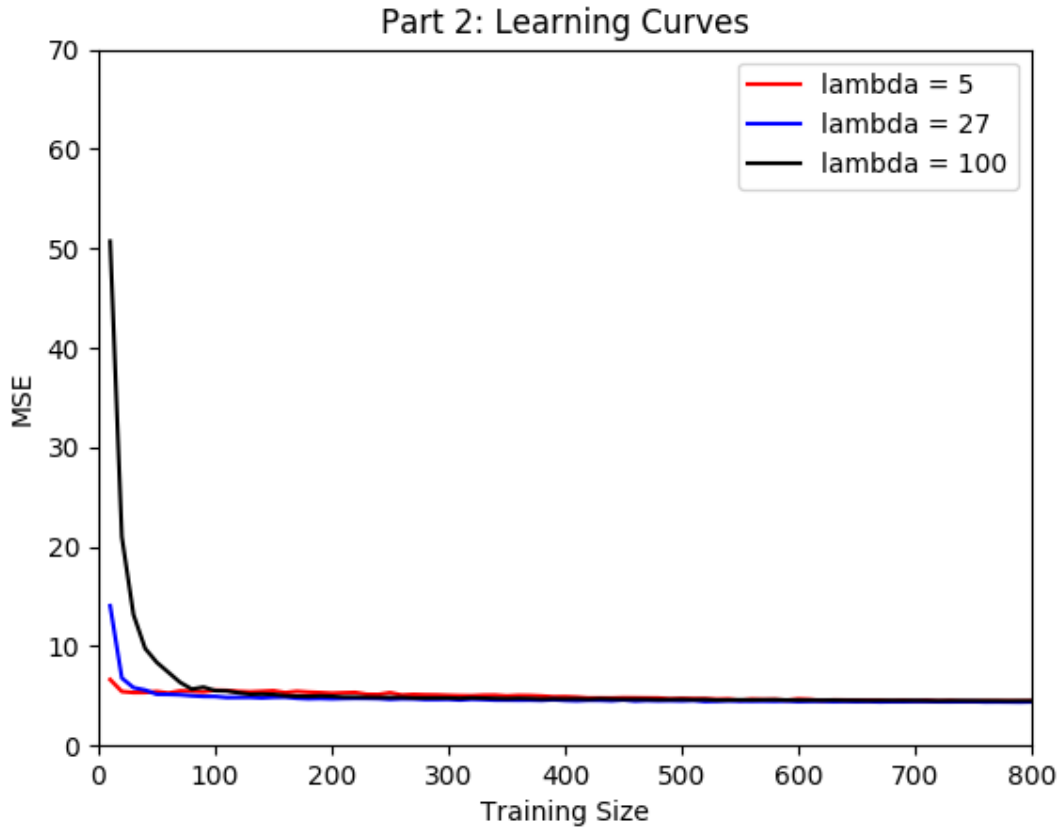
As we can see from the plots, the test set MSE is an inverse parabola. The MSE of the test set first decreases with the increase in λ and then after a point keeps increasing. This implies that there is a perfect regularization parameter for each of the datasets. Very small values of λ are almost like unregularized linear regression and thus don't contribute towards solving the problem of overfitting. Whereas too big values of λ tend to overshadow the patterns hiding in the dataset. However, there will exist a perfect lambda at the point where MSE for test is minimum.

Q. How does the choice of optimal λ vary with the number of features and number of examples? How do you explain these variations?

As we can see from the first 3 datasets, 100-10, 100-100, 1000-100, if we increase the number of features, we get an optimal λ which is higher than the dataset with less features. We can also see that the unregularized version of linear regression with λ=0 performs very badly on the test set. We can infer from the plot of 100-10 and 100-100 that having more features might not necessarily mean a better model as many of the features might end up being extraneous ones as they might not contribute towards solving the problem and might serve as noise. Thus, we get a higher value of lambda to counter the overfitting done due to the high number of features. Also, from the plots of 1000-100 and 100-100 we can see that with the same number of features and increasing the data we get a better performance of the perfect lambda. Although having a lot of data increases the chance of overfitting, λ takes care of that and the model ends up performing better as it now has more training examples to learn from.

TASK 2

The plot of 3 different values of lambda: too small, just right and too big (5, 27, 100) is shown below:



From the figure we can infer that as the training size increases, different values of lambda give almost similar performance on the test set data. The key difference between these three lambdas are in between around 10-100. That is the training size very low. At the very small level, there is not a lot of overfitting happening and thus big values of lambda fare very badly and too small values

perform best. However, after a point these too small values also start performing badly and the perfect value of lambda ends up giving the best estimate.

TASK 3

The output of task 3 is:

Dataset test-100-10: 4.18010155781

Found at (lambda, alpha, beta) : 4.68753172878 1.20197823347 0.256420287481

Dataset test-100-100: 7.35246238289

Found at (lambda, alpha, beta) : 2.31997709886 1.12347258704 0.484260205668

Dataset test-1000-100: 4.33835151521

Found at (lambda, alpha, beta) : 10.3538132571 2.73595738124 0.264246351881

Dataset crime: 0.391102301805

Found at (lambda, alpha, beta) : 130.950301647 425.645089372 3.25043229391

Dataset wine: 0.626746102226

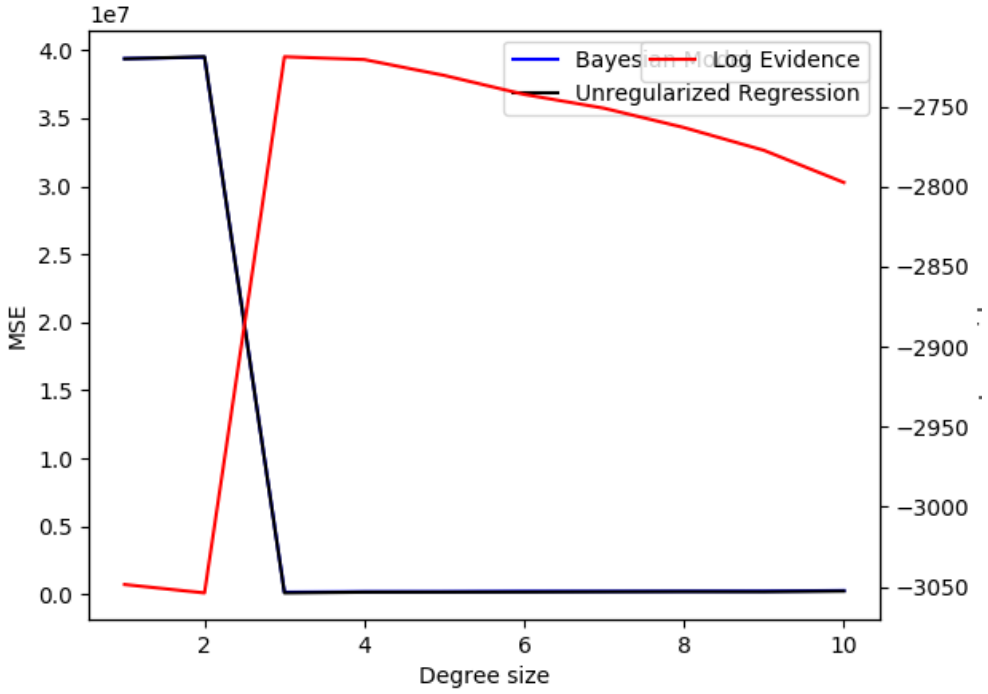
Found at (lambda, alpha, beta) : 3.82888436422 6.16377595148 1.60980989896

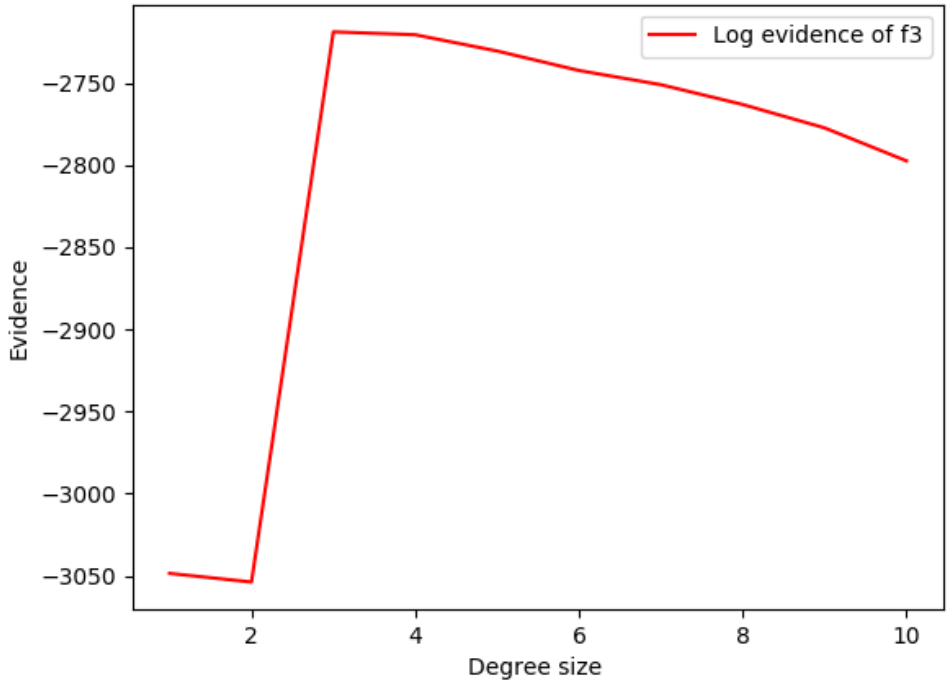
We get the output of task 3 after calculating the parameters of the prior alpha and beta such that the evidence function is maximized. After comparing these outputs with the output of task 1 we can say that Bayesian model selection performs comparable to the perfect lambda in task 1. This is in spite the fact we have chosen the optimal lambda based just on the training set instead of cross-validation as done in task 1.

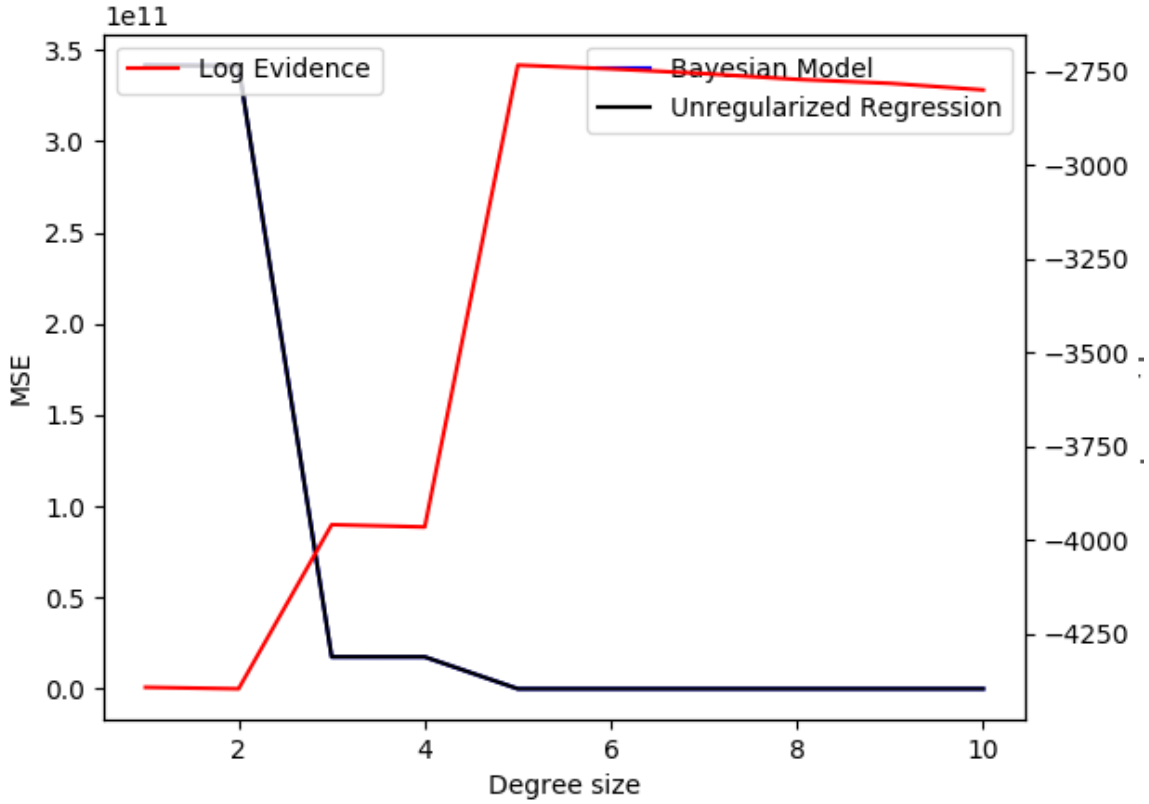
The correlation between increasing the number of features and number of examples remains the same though. When we go from 100-10 to 100-100, the performance worsens due to many features being added that might not be contributing a lot and thus over-fitting happens. Whereas from 100-100 to 1000-100 the performance becomes better as now the algorithm has more examples and a better sense of the data distribution and the patterns in the dataset become obvious.

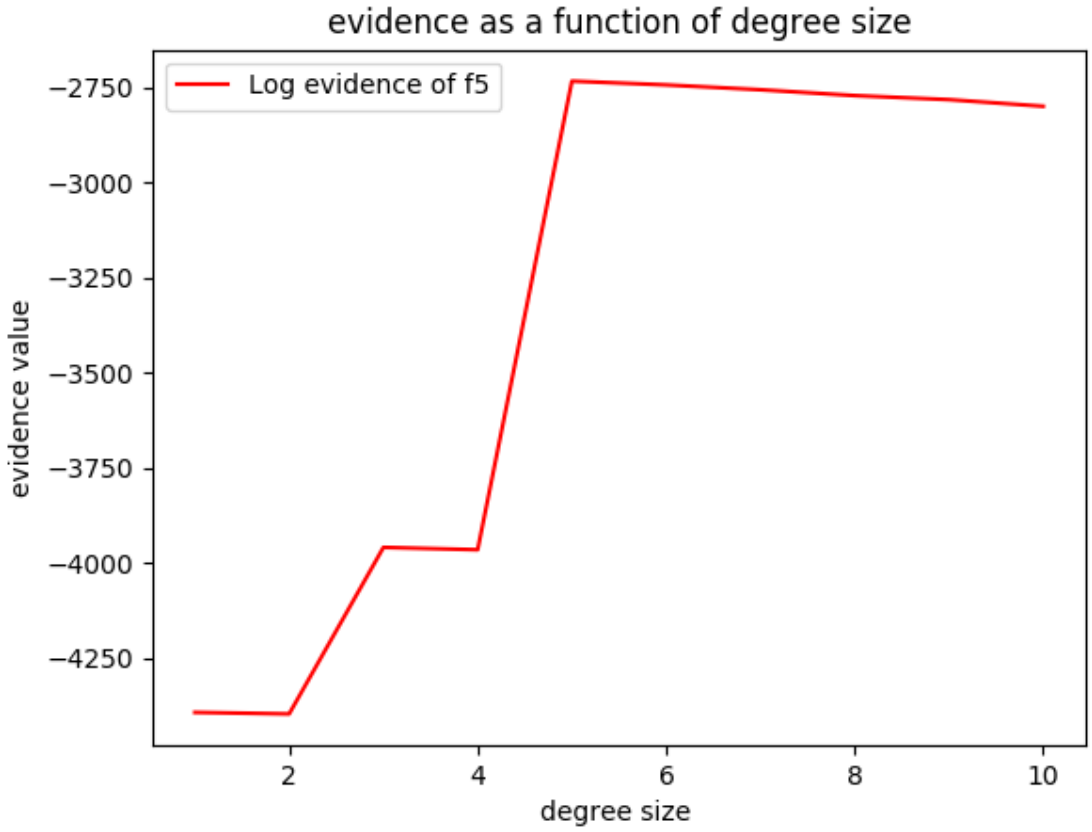
TASK 4

The plots for task 4 are as follows:

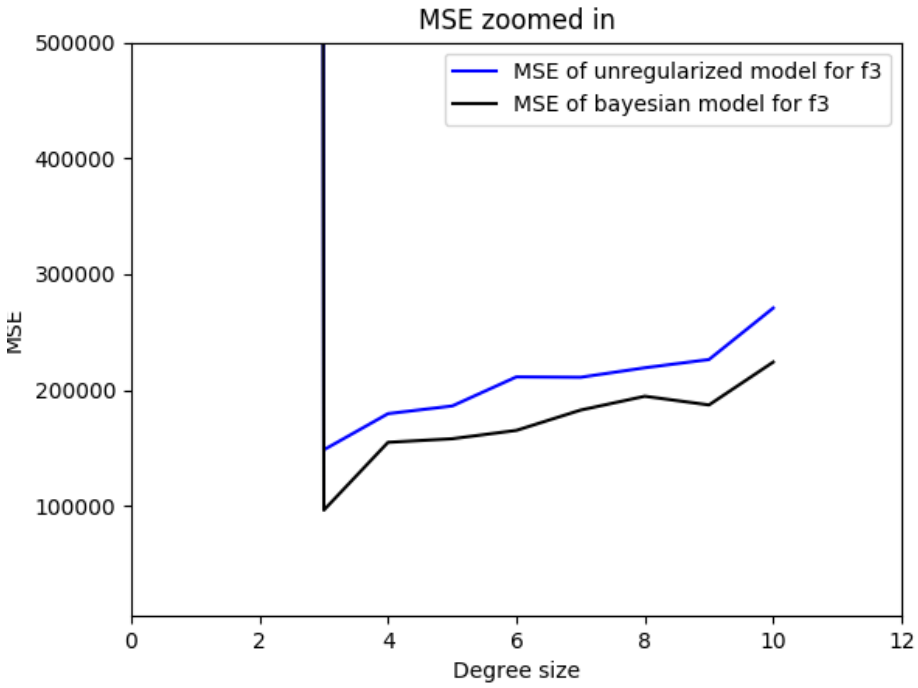


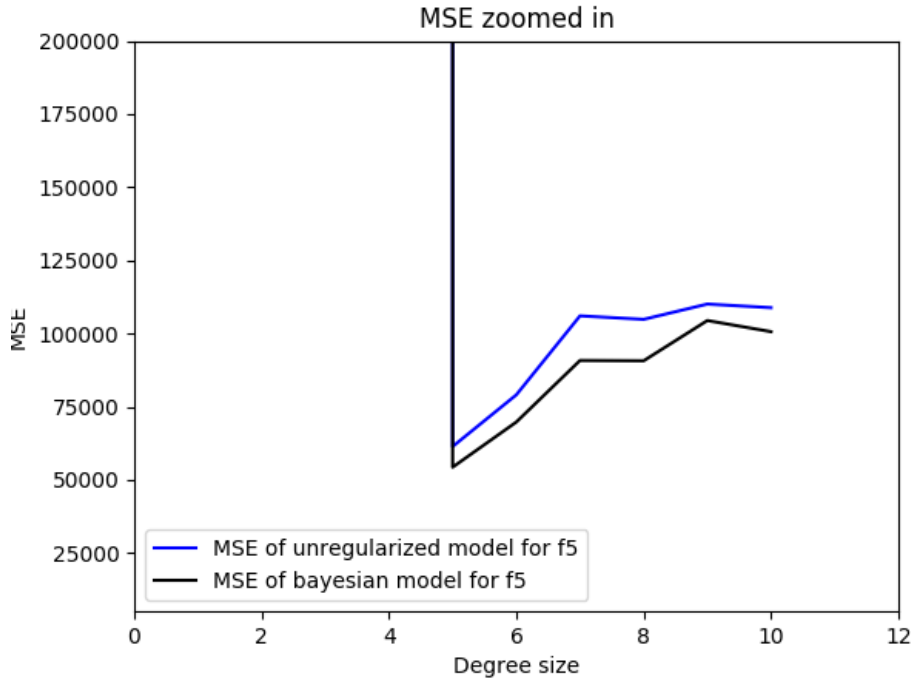



1e11
Log Evidence
Bayesian Model
Unregularized Regression
MSE
Degree size
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−2750
−3000
−3250
−3500
−3750
−4000
−4250
2
4
6
8
10


evidence as a function of degree size
Log evidence of f5
evidence value
degree size
−2750
−3000
−3250
−3500
−3750
−4000
−4250
2
4
6
8
10

In these graphs we can see how the dimensions affects the Bayesian model selection algorithm and the unregularized linear regression model. The graphs were coming out badly scaled due to the large value and it seemed as if there is no difference between Bayesian and Unregularized. However after dimension 3 that is d > 3, we can notice a significant difference. For this I have plotted a zoomed in graph as follows:





From the graphs we can see that for the case of Bayesian model selection, log evidence serves as a good indicator to select the optimum number of dimensions. When log evidence is high, test set MSE is low and when log evidence is low, test set MSE is high (log evidence has negative values). At d

= 3 we can see that the graphs intersect and thus d = 3 seems like a good choice for the number of dimensions. This dimension =3 gives us an optimal trade-off as MSE is low and log evidence is high. If d gets lower, there will be underfitting and if d gets higher there will be overfitting. As we keep increasing dimensions the log evidence reaches near negative infinity. Also from the zoomed in graphs we can see that Bayesian model performs better than the non-regularized model.