By: Ritvik Kapila

ENTRY NUMBER: 2017EE10484

ASSIGNMENT - 3

SUPPORT VECTOR REGRESSION

ELL409 | 6th Semester 2019-20 | Electrical Engineering Department

In this assignment, two kinds of Support Vector Regression have been implemented by using the cvxopt Optimization library:

1. Epsilon Support Vector Regression (Epsilon-SVR)
2. Reduced Convex Hull Support Vector Regression (RH-SVR)

Further, the output of these regressions has been compared to the in-built sklearn library to check their correctness. The variables such as C, gamma, epsilon, and the degree of polynomial in the polynomial kernel have been varied in order to find the optimum solution for both the SVR's with multiple kernels such as:

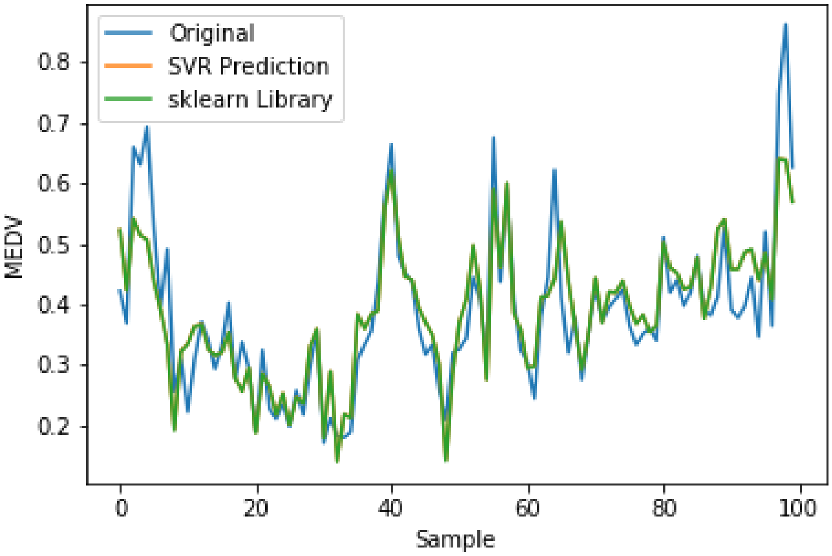
1. Linear Kernel
2. Polynomial Kernel
3. Radial Basis Function Kernel (RBF Kernel)

- The data is first normalized in order to get all the values in the range [0,1].
- The model accuracies are verified by plotting the Mean Square Error (MSE) using k-fold cross validation on the dataset.
- Plots have been made in order to find the optimum values of the hyperparameters where we get the minimum MSE after the k-fold validation.
- Plots for comparisons between inbuilt libraries and developed models have been formed in order to study the MSE.

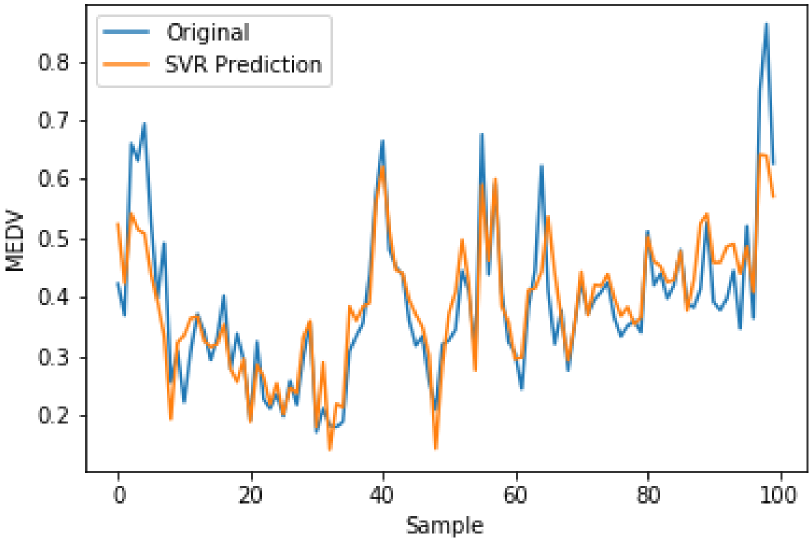
Epsilon Support Vector Regression:

1. Linear Kernel

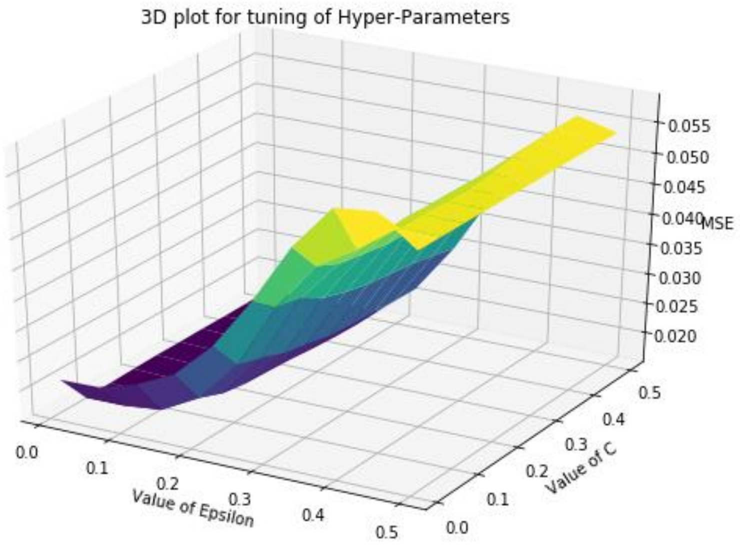
For the linear kernel, the output for the Epsilon SVR using the Convex Optimization library versus the regressor from sklearn gives the following result over the whole dataset for a training set of 406 samples and a test sample of 100 points. This result is also compared with the original output 'MEDV' of the dataset. The output was observed to be in coherence with the original 'MEDV' at C and epsilon approximately equal to 0.1 and 0.05 respectively.

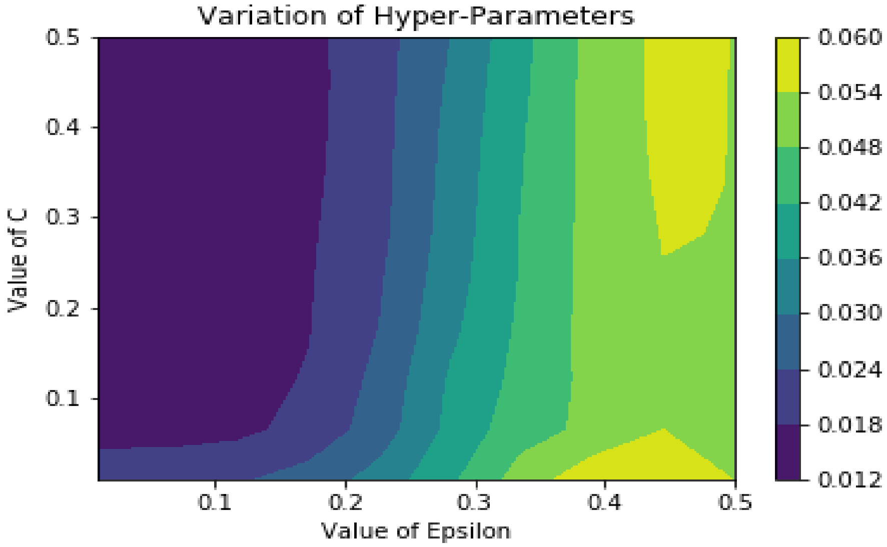


We can fairly see only two curves and that is because SVR from the inbuilt sklearn library gives the exact same result as that of the predicted Epsilon SVR. Here the value of C = 0.1 and epsilon = 0.01. The output from the sklearn library is removed and we can see that the curve of the predicted output exactly traces the curve for the scikit library. This validates the model's accuracy.



Now we find optimal hyperparameters in order to get the best results on the dataset for linear kernel. The following is a plot varying C and epsilon while plotting the MSE for a **5-fold Cross validation**. The blue regions denote lower MSE and yellow regions denote higher MSE.

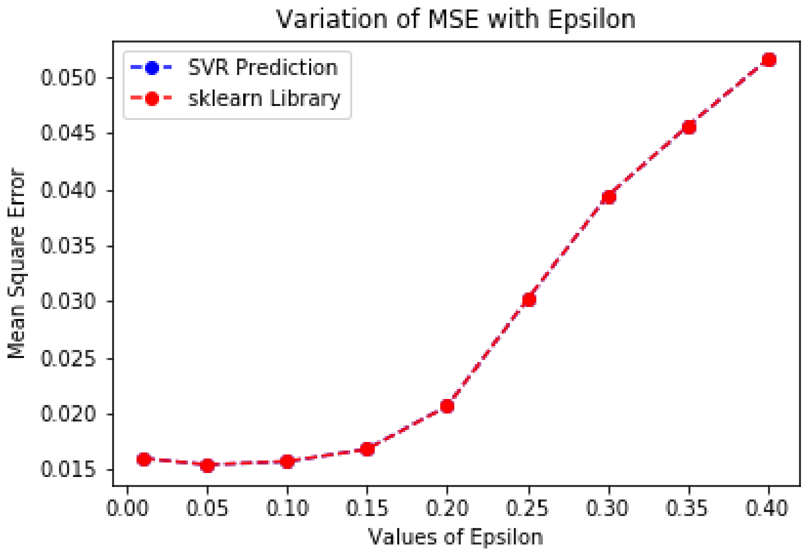


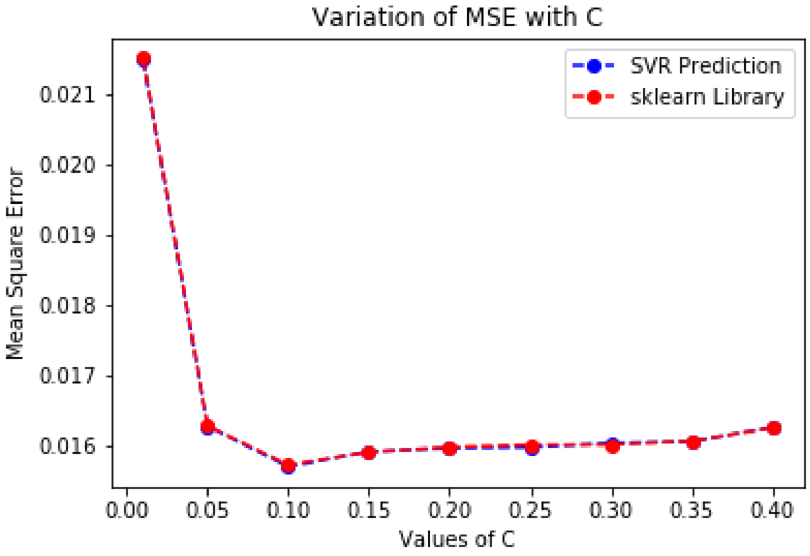


This can be better understood using a 3D plot. Varying these hyperparameters, we get the minimum MSE at:

Optimum C = 0.11888888888888888
Optimum epsilon = 0.06444444444444444

Further comparing the **MSE** for predicted **Epsilon SVR** and the **scikit learn SVR** by varying epsilon and C **near the optimum value** we get the following graphs.



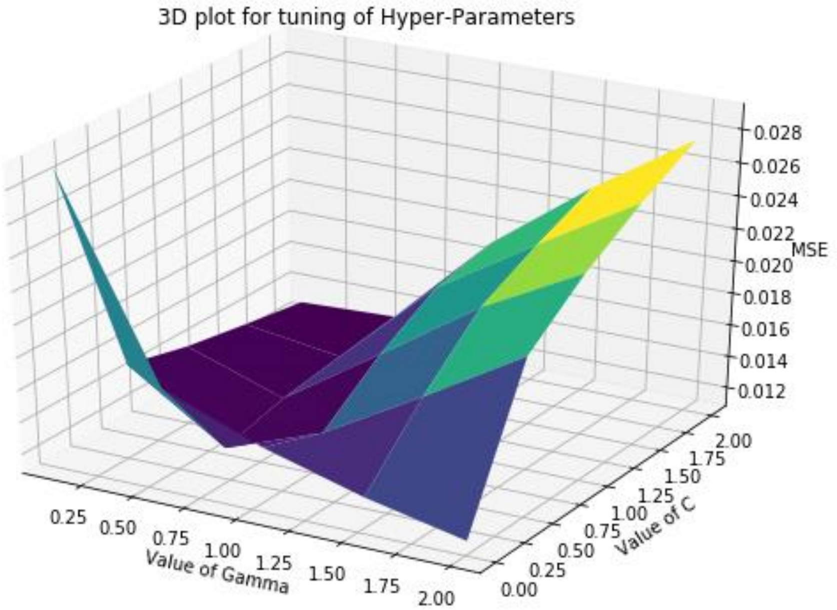


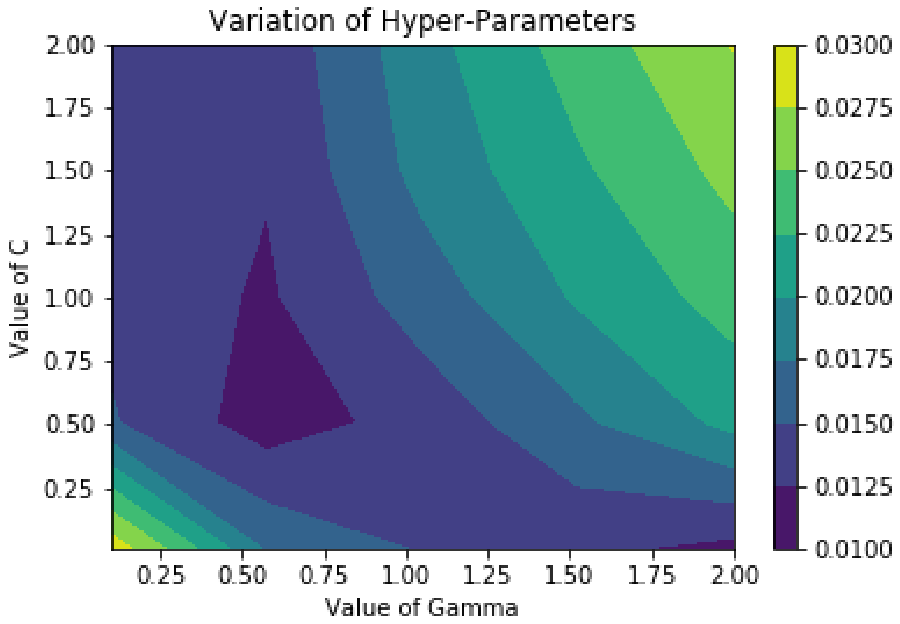
We can clearly see that by varying C and epsilon and comparing the two SVR's, we are getting the same MSE, i.e. overlapping MSE. Hence both the models are coherent with each other.

2. Polynomial Kernel (Degree = 2)

For the polynomial kernel of degree 2, the output was observed coherent with the 'MEDV' values near C = 2, gamma = 1 and epsilon = 0.1.

Further, in the polynomial kernel, we fix the value of epsilon, hence fixing the dimensions of the epsilon tube, and vary C and gamma to find their optimum values using the following graphs.

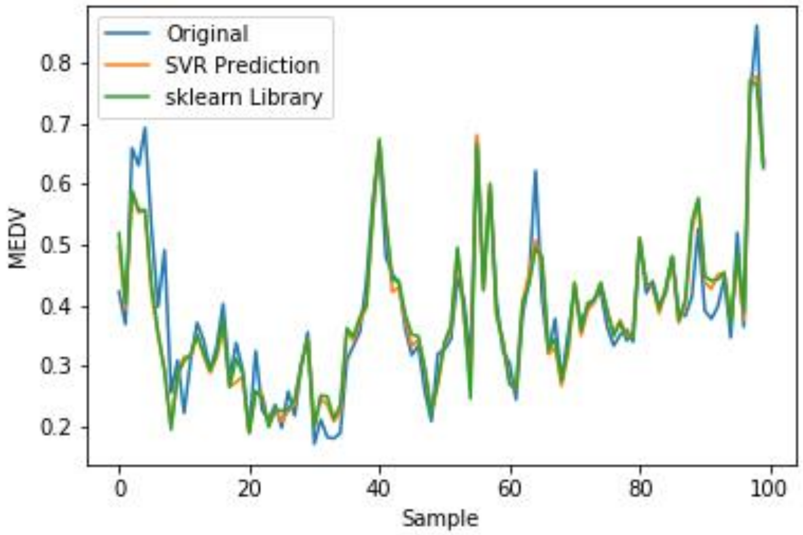




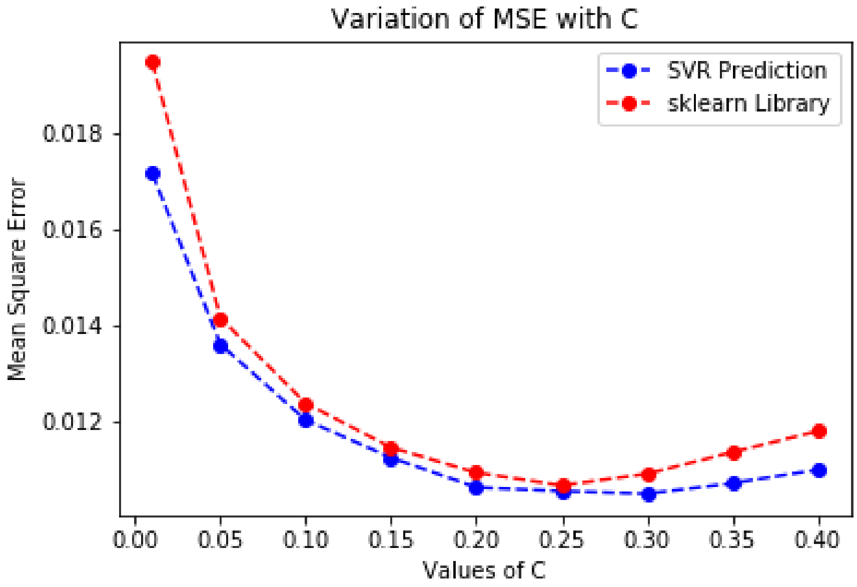
Optimum C = 0.5075

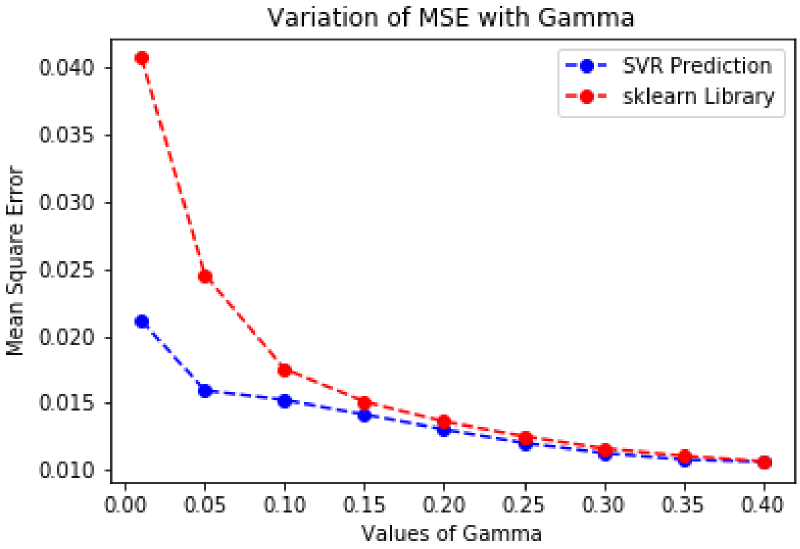
Optimum gamma = 0.575

For these optimum values of C and gamma, we get the following graph for the outputs after taking the training data of 406 samples and test data of 100 points:



Further comparing the **MSE** for predicted **Epsilon SVR** and the **scikit learn SVR** by varying C and gamma**,** we get the following graphs.

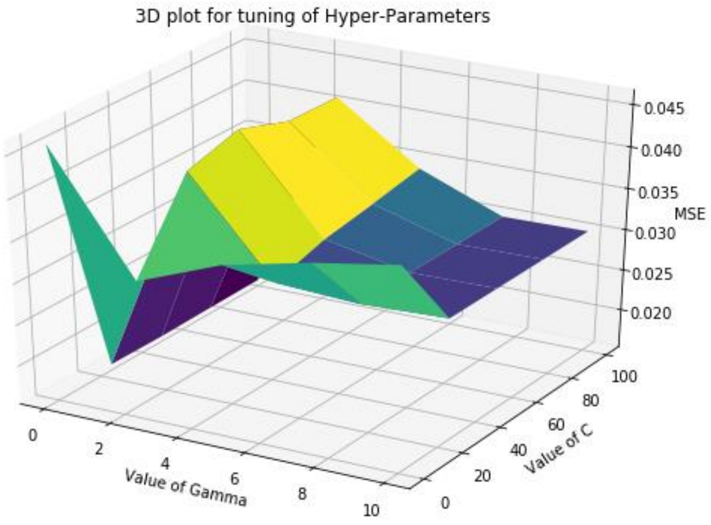


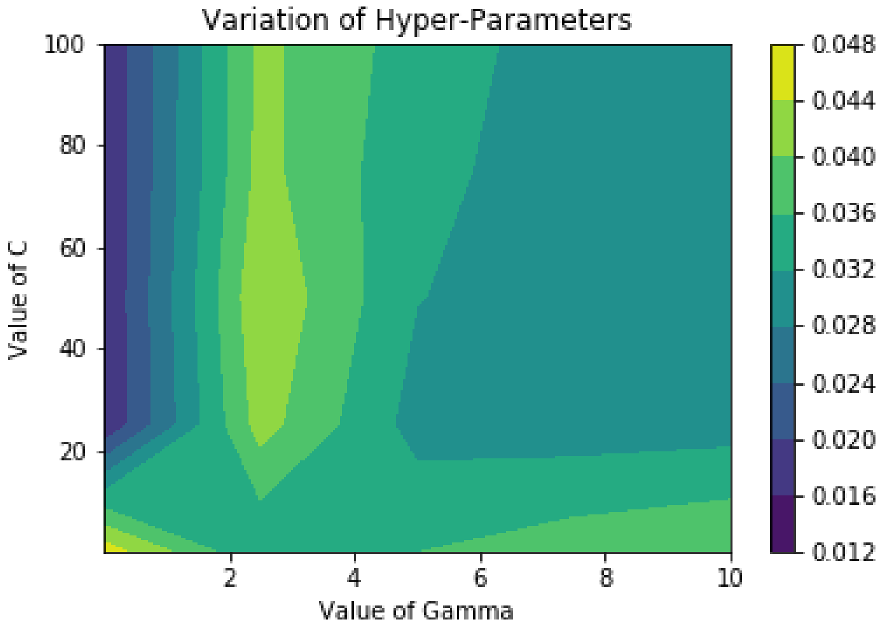


We observe that the MSE for the polynomial kernel is always less for the generated model as we vary gamma and C.

3. RBF Kernel

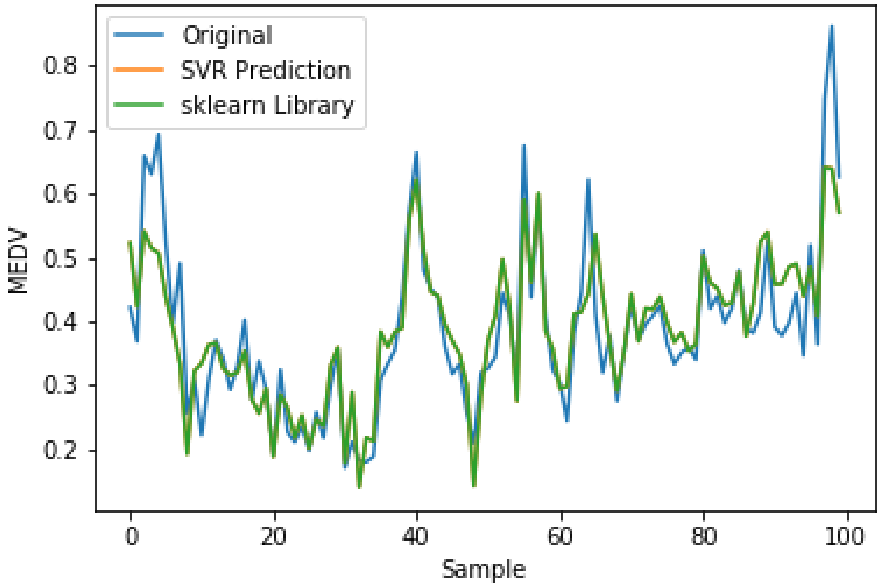
Similar to the polynomial kernel, we fix the width of the epsilon tube, and vary the C and gamma in order to find the optimum solution for the SVR using the following graphs.



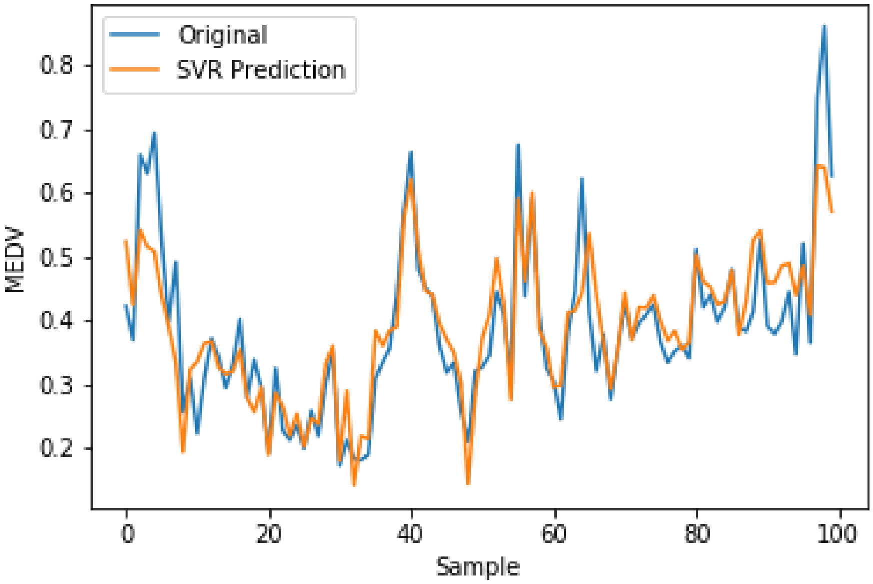


Optimum C = 30.004999999999995
Optimum gamma = 0.001

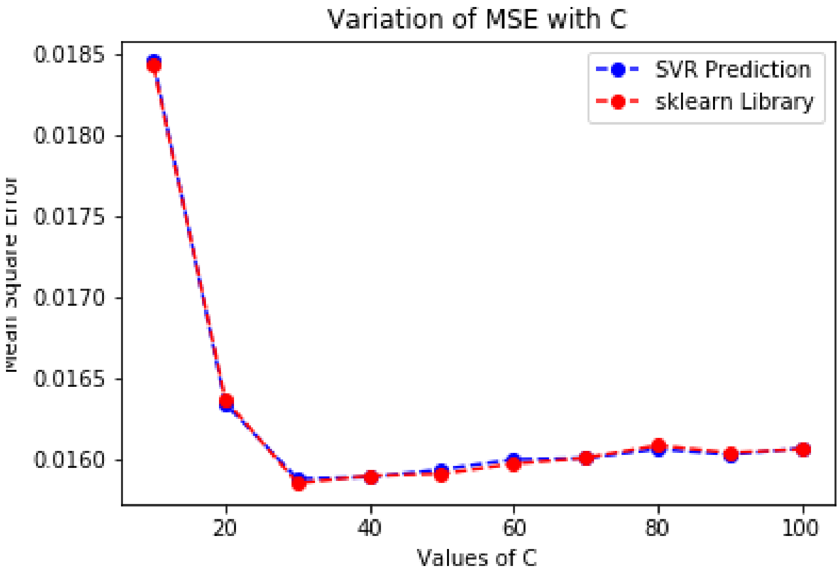
For this optimum value of hyperparameters, we get the following graph for the outputs after taking the training data of 406 samples and test data of 100 points:

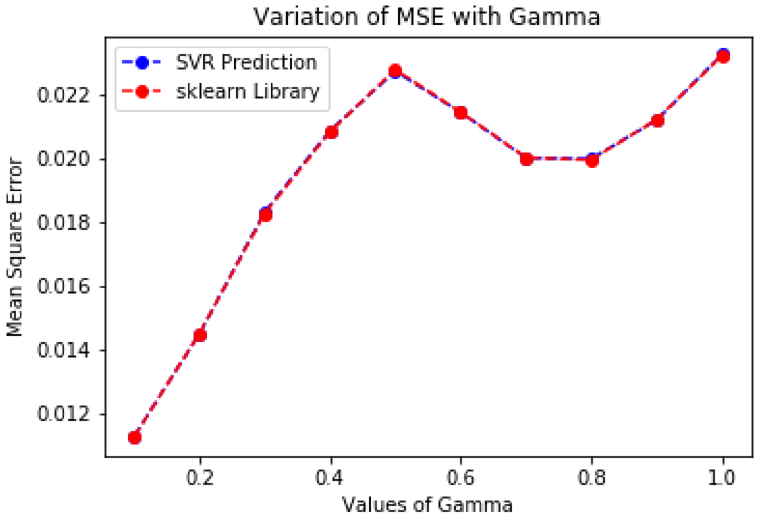


As we can see, the output of the inbuilt SVR coincides with that of the model Epsilon SVR, we can see it clearly in the next plot where the sklearn is removed and both the curves in the two graphs can be seen to trace the same path.



Now we try to find the variation of the MSE for the model SVR and the scikit learn SVR as we vary the hyperparameters in order to compare the efficiency for both.

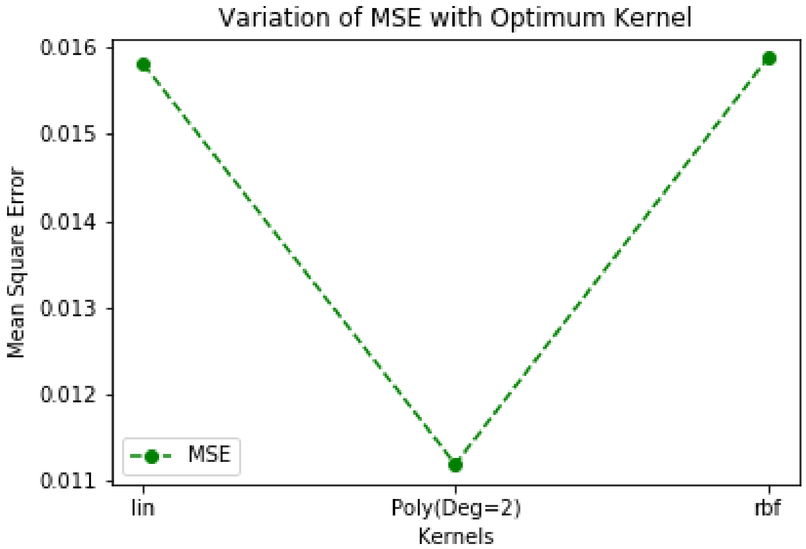




We can clearly see that by varying C and gamma for epsilon = 0.01, and comparing the two SVR's, we are getting the same MSE, i.e. overlapping MSE. Hence both the models are coherent with each other.

Comparing the kernels:

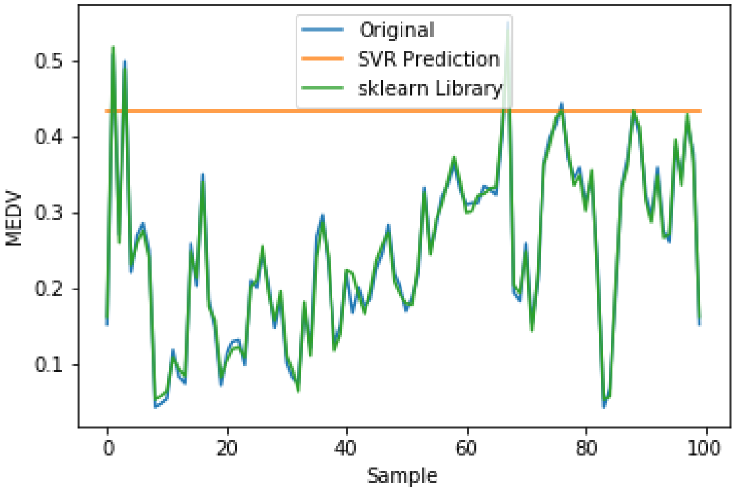
Now we plot the accuracy for different kernels at their optimum hyperparameters in order to find the best fit for the data. This is the result that we obtain:



We can clearly see that polynomial kernel is the best choice for a kernel for this dataset at which we get the minimum MSE as 0.0111 after a 5-fold cross validation.

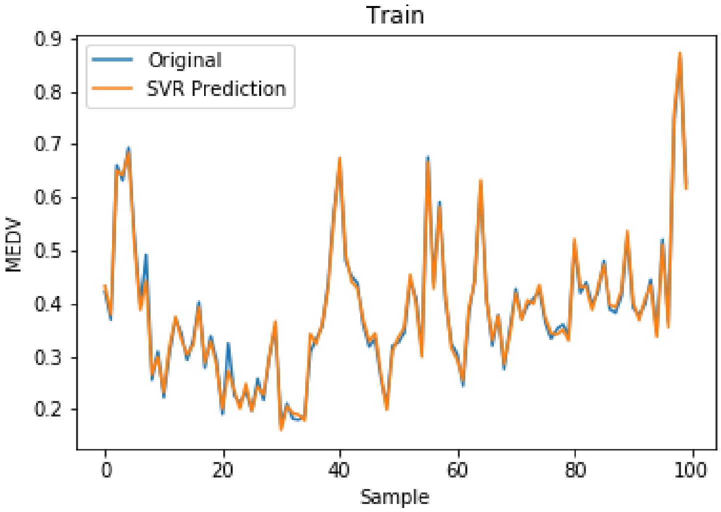
Overfitting and Underfitting:

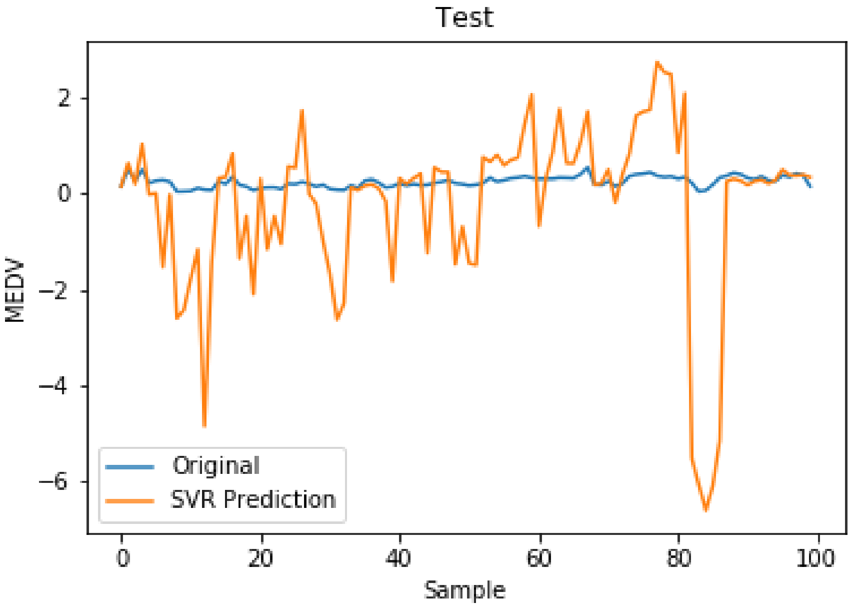
I observed the model to be underfit if I take the value of epsilon to be above a threshold. This happens because the error margin of the epsilon-tube is high and the regressor is not able to learn much. This can be seen in the following graph.



Here the epsilon is taken to be 2, which is higher than the values in the dataset after normalization. The model gives a straight line as all the points anyways lie in the epsilon tube.

Overfitting can be observed if we train on the data, and the model complexity becomes very high. Suppose we increase the complexity of the model by using a polynomial kernel of degree 5, this is the training and testing data results that are observed.





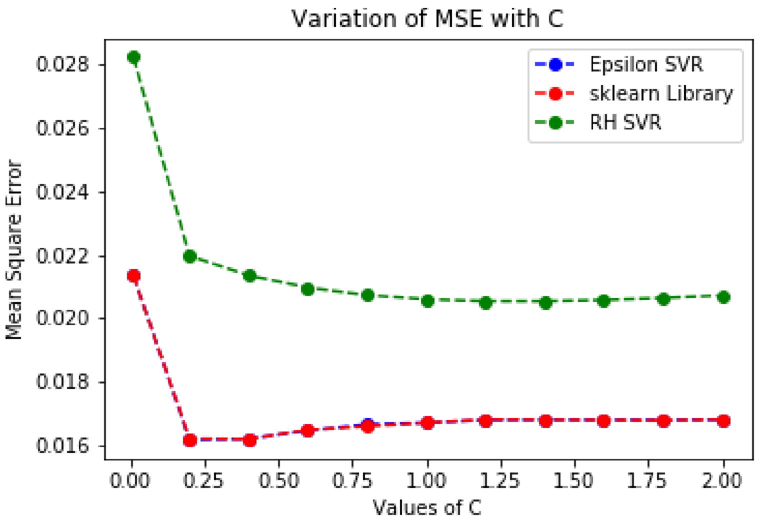
It is evident that the training data is overfit as the dimension of the data is increased more than required. Hence, we observe such results on the test data.

RH-SVR

1. Linear Kernel

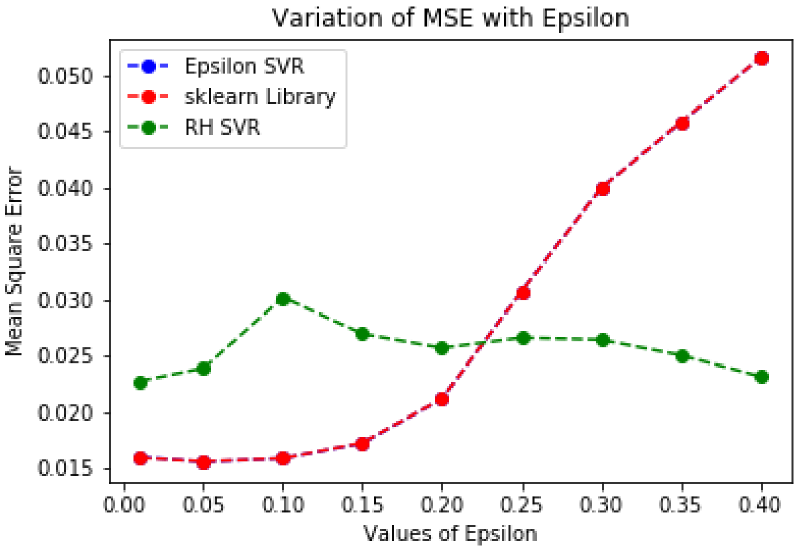
This was the bonus portion of the assignment. I have implemented the Reduced Convex Hull Support Vector Machine and compared it with the Epsilon SVR.

After a 5-fold Cross Validation for a linear kernel, the variation of the hyperparameter C is taken for the Epsilon SVR, the inbuilt SVR and the RH SVR and the MSE is compared in order to compare the results for the models. Here C is varied from [0.01, 2.0], Epsilon = 0.01. The following result is obtained.



Here we can see that the RH SVR prediction has a higher MSE when compared to both the Epsilon SVR and the inbuilt Scikit SVR.

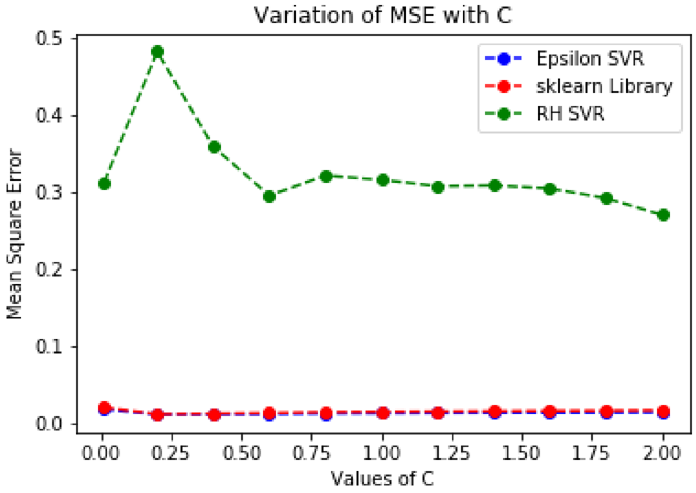
We also study the variation of MSE with changing Epsilon as follows and observe a similar result.

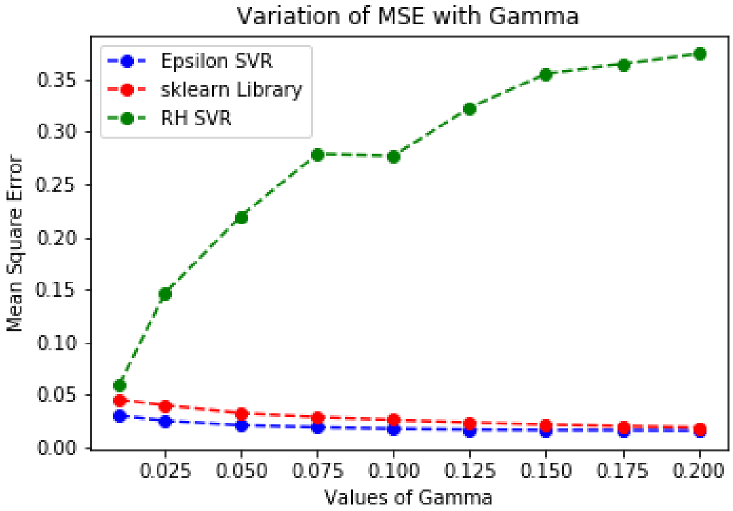


The variation can be understood by the fact that the optimum value of C and epsilon for Epsilon SVR was near 0.1 and 0.01, so around those values of the hyperparameters, Epsilon SVR gives better results, but as we go away from these values i.e. as we increase Epsilon, the RH SVR starts giving a lesser MSE.

2. Polynomial Kernel

Variation of MSE for ploynomial kernel of degree 2 with varying hyperparameter C is also observed, since the parameters have been tuned for the Epsilon SVR, the error for Epsilon SVR comes to be considerably less than that of RH SVR.:

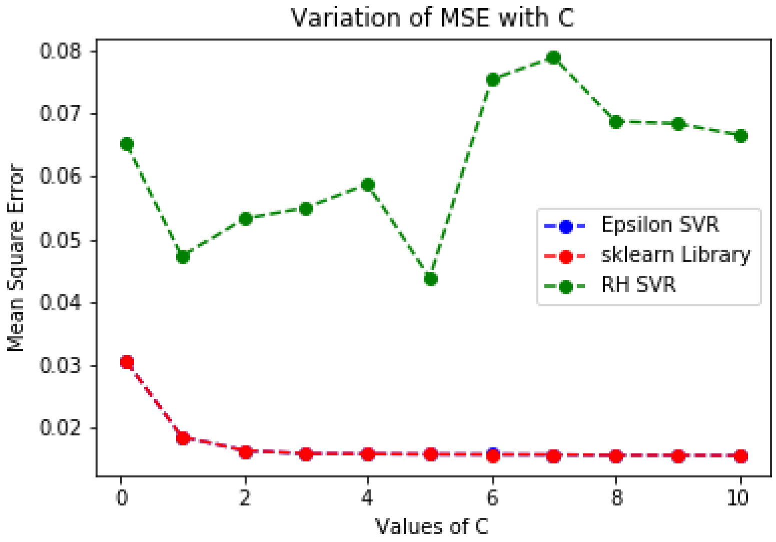


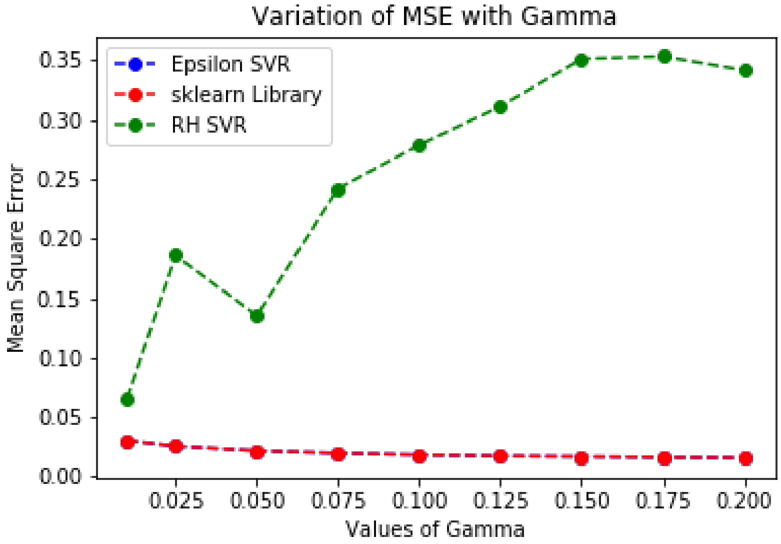


The variation with gamma indicates that the MSE is almost equal for low values of gamma, but the error increases by a large value for RH SVR as we increase gamma.

3. Radial Basis Function Kernel

Variation of MSE with varying hyperparameters C and gamma is as follows:





It is evident that the MSE for Reduced Convex Hull SVR is higher than the Epsilon SVR for these values of the hyperparameters C and gamma.