Assignment 3 ELL409

Yashank Singh 2017MT10756

Using Epsilon-SVR, RH-SVR with CVXOPT and SVR using sklearn on Boston Housing Data

NOTE: The data set has been scaled using min-max scaler to bring all feature values in the range of (0,1). k=5 fold cross-validation has been used for all the results in this report and **R^2 coeff.** has been used as a metric to evaluate performance. **For the K folds avg. of R^2 score and MSE has been plotted**.

All the utility functions such as MSE, R^2 coeff, kernels etc. have been implemented from scratch.

1. Epsilon-SVR – epsilon SVR aims to find a function *f(x),* where

f(x) = <w, x> + b with w ∈ X, b ∈ R, x being the input variable, such that the predictions y^i=f(x^i) are at most at a deviation of e from the given target variable y^i and penalizes only the points outside the epsilon region. This is formulated as the following optimization problem.

$$\begin{aligned} L := & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*) - \sum_{i=1}^{\ell}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ & - \sum_{i=1}^{\ell}\alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b) \\ & - \sum_{i=1}^{\ell}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b) \end{aligned}$$

Here L is the Lagrangian and η^i, $\eta*^i$, α^i, $\alpha*^i$ are Lagrange multipliers. Hence the dual variables have to satisfy positivity constraints $\alpha(*)^i$, $\eta(*)^i \geq 0$.
But this is intractable to solve, so we rather solve the dual which reduces to the following after imposing the KKT conditions and kernelizing <x^i,x^j>

$$\begin{aligned} \text{maximize} \quad & \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{\ell}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) \\ -\varepsilon\sum_{i=1}^{\ell}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{\ell}y_i(\alpha_i - \alpha_i^*) \end{cases} \\ \text{subject to} \quad & \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned}$$

The predictions are then done using the following equation for new data point x in the feature space

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)k(x_i, x) + b$$

As sklearn also implements epsilon-SVR, we compare our results with the library function sklearn.svm.SVR while the implanted version uses CVXOPT to find the solution to the dual optimization problem.

*The following results are obtained and plotted for **Linear kernel***

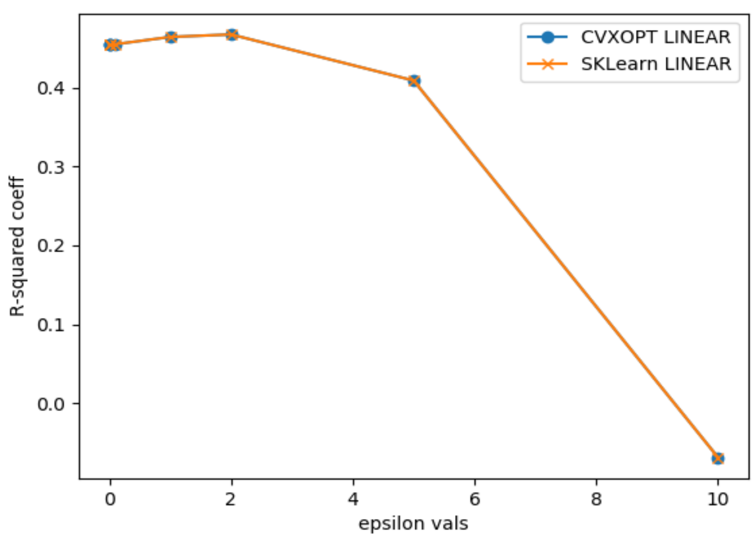
Varying Epsilon

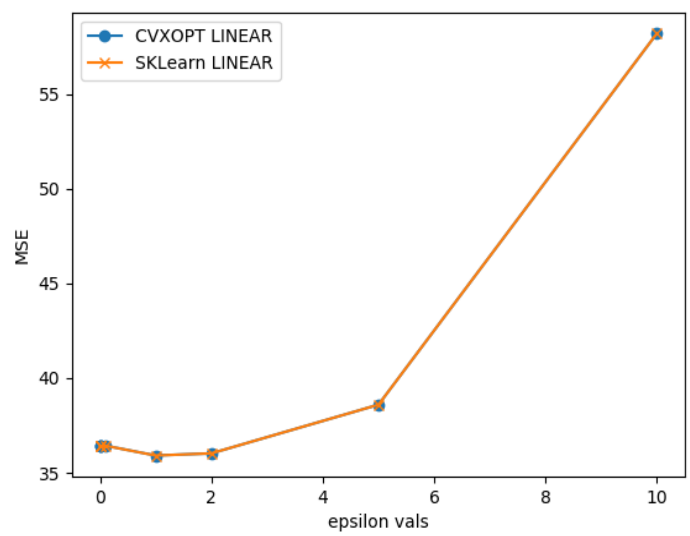
Varying epsilon in range [0.0001, 0.001, 0.01, 0.1, 1, 2, 5, 10] with C=1

The following are the corresponding values rounded to 3 decimal places

R^2: 0.454, 0.454, 0.454, 0.455, 0.464, **0.467**, 0.409, -0.069

MSE: 36.399, 36.399, 36.42 , 36.435, **35.918**, 36.029, 38.597, 58.192



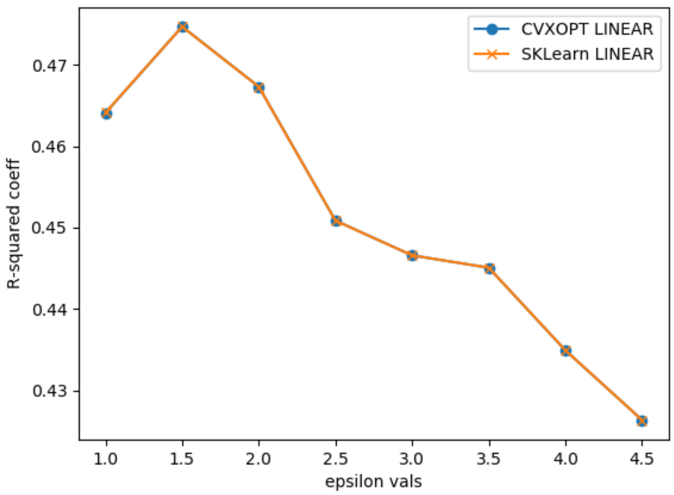


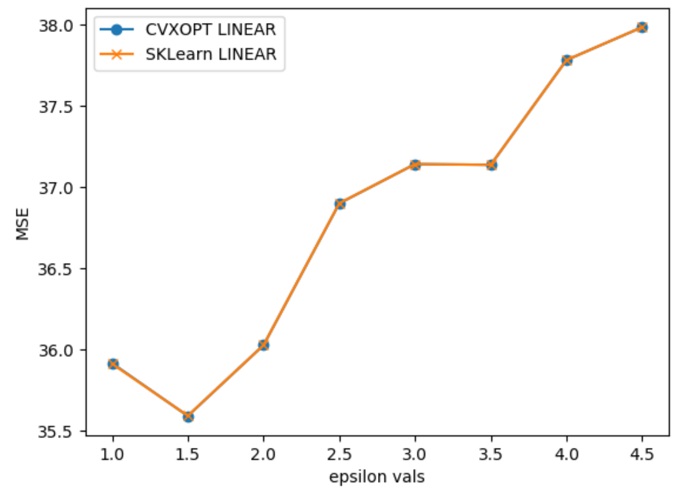
Comments on varying epsilon : Here we observe that the peak of the R^2 coeff occurs in the interval [1,5], as the values of r^2increase uptill 2 and then start decreasing sharply. This is reassured by the plot of the MSE, which takes minima around the value of 2. Hence we further inspect the in the interval [1,5].

*Also observe that as the value of eps goes to 10, R^2 coeff becomes negative, which implies we are in the **underfitting regime**, as we do worse than the mean of Y_train. This happens because we relax the penalty to large variations from y values, as we keep the diameter of the e-tube large. Hence we end up not really learning anything significant.*

Here, we also observe that the model implemented using CVXOPT gives almost the same results as the one using sklearn.svm.SVR. Which, reassures that the implementation is indeed correct.

Varying epsilon in range [1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5] with C=1

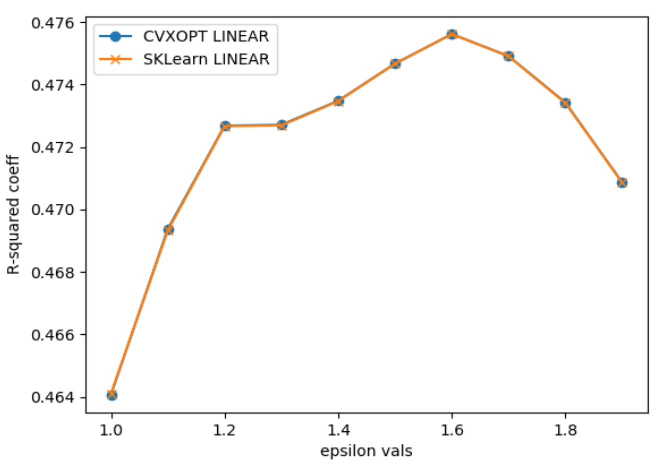


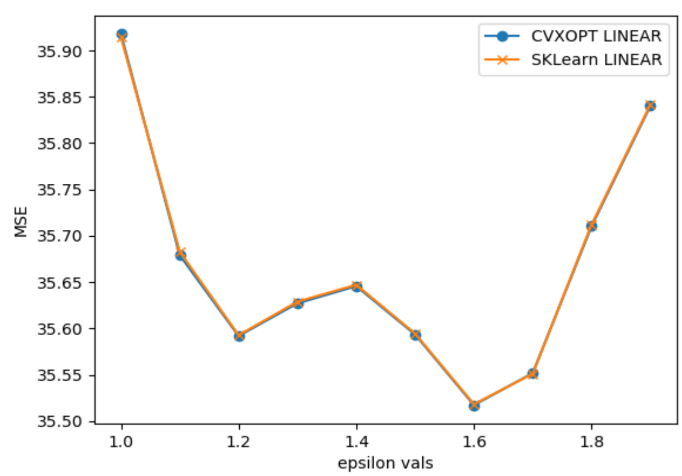


Again, we see that both the models give the same results as.

We observe that the peak of r2 occurs around, eps=1.5, so we fine tune again in the interval [1,2] with step size as 0.1 and C=1.

Varying epsilon in range [1, 2] with step size =0.1 and C=1





We see that the optima occurs at eps=1.6 with R^2 = 0.476 and MSE = 35.51.

The following are the values of r2 and MSE at eps =1.6 for both models for the 5 cuts used to cross-validate the model. The values are almost same.

R2 cvxopt➔ [0.66974, 0.60408, 0.2856 , 0.32084, 0.4978]

R2 sklearn➔ [.66974, 0.60403, 0.28558, 0.32084, 0.49786]

MSE cvxopt➔ [11.40547 36.04344 57.21455 59.57082 13.35256]

MSE sklearn➔ [11.40552 36.04789 57.21609 59.5705 13.35107]

Here we see that, the model isn't performing good in all the cuts. This doesn't generalize well for all the test cuts in the data, hence we happen to be in the overfitting regime or those cuts have outliers that result in higher values of MSE and lower values of R2.

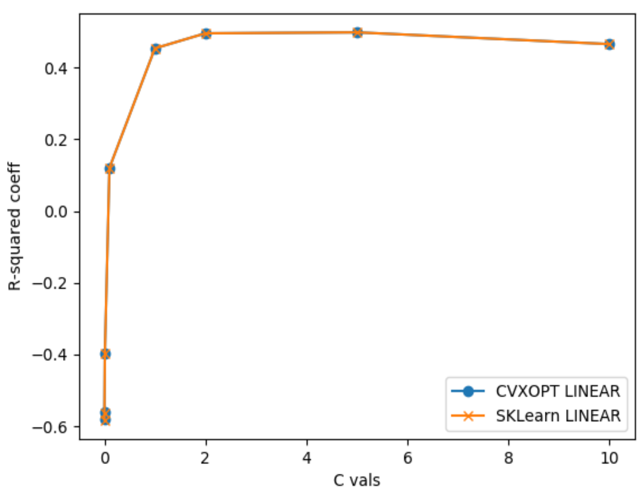
Hence we conclude that for a given C value at lower values of epsilon, we happen to be in the overfitting regime as we give keep the diameter of the e-tube less and for larger values of eps we end up in the underfitting regime.

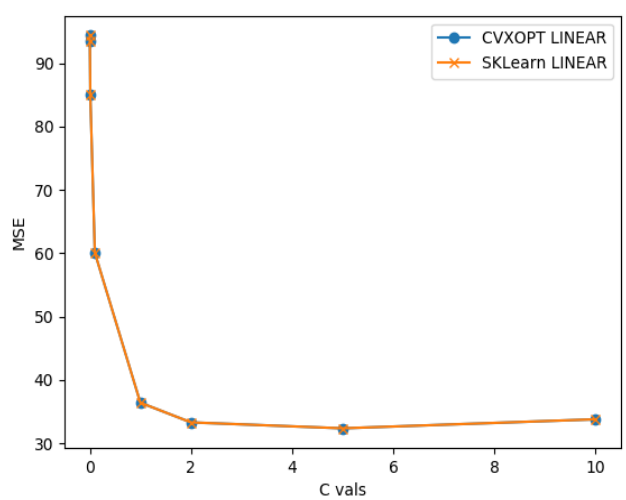
Varying C

Varying C in range [0.0001, 0.001, 0.01, 0.1, 1, 2, 5, 10] with epsilon=1.6

R^2: -0.58105, -0.56153, -0.39543, 0.12066, 0.45398, 0.49596, 0.49866, 0.46589

MSE: 94.3795, 93.4322, 85.0838, 60.0841, 36.4205, 33.3122, 32.3714, 33.7967

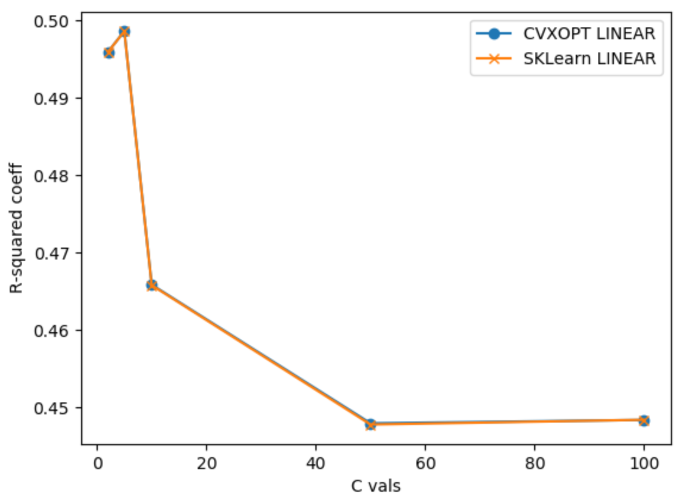


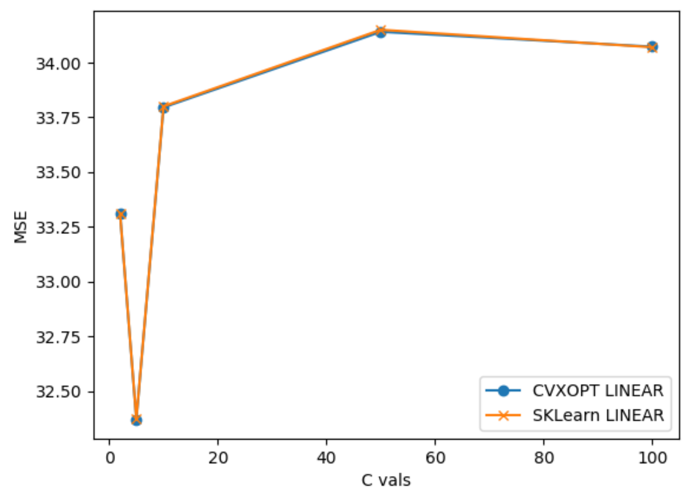


Varying C in range [2, 5, 10, 50, 100] with epsilon=1.6

R^2: 0.49596, 0.49866, 0.46589, 0.44798, 0.44841

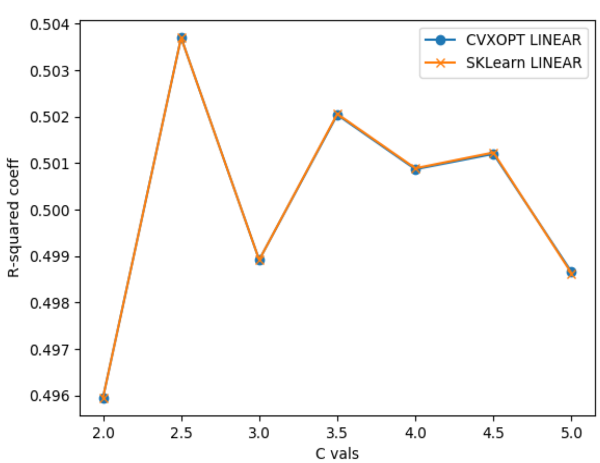
MSE: 33.31231, 32.3709, 33.79604, 34.14139, 34.07327

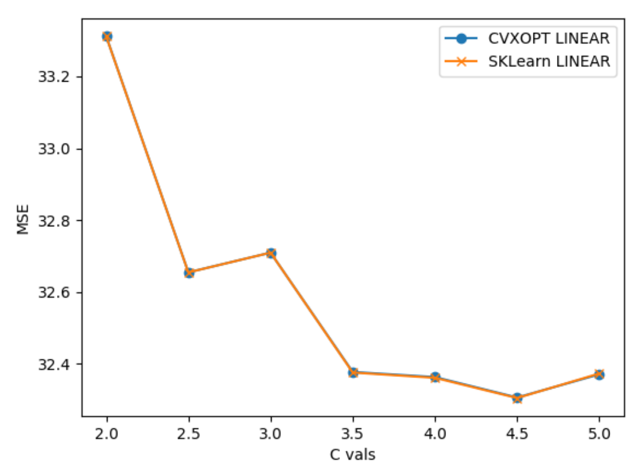




Comments on varying C: From the above graphs, we conclude that for **very low values of C** (reg parameter), the model ends up in the **overfitting regime** as the R2 score turns out to be negative. However as we increase the value of C we move towards the optimal region, and as we increase it further we move towards the **underfitting regime**, and R2 doesn't fall down lot.

Hence we see the values of R2 and MSE for the k folds to find out which value of C in [2,5] results in best generalization.





We see that although the value of MSE is lowest at C=5, but taking a look at the values for k folds, **C=2.5, epsilon=1.6 gives the best generalization.**

MSE : 32.516

cvxopt→[9.52581 29.5633 46.74092 61.63339 15.81046]

sklearn →[9.52559 29.56587 46.73768 61.63363 15.80993]

R2 : 0.5079

cvxopt → [0.72417 0.67526 0.41637 0.29732 0.40536]

sklearn → [0.72418 0.67523 0.41642 0.29732 0.40538]

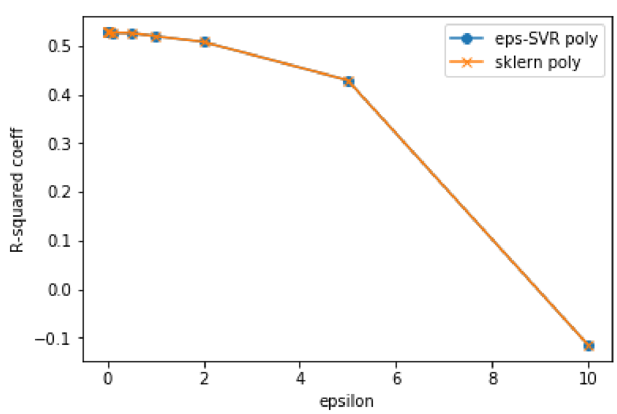
Conclusion : *Hence we conclude that for low values of C, the model underfits the data, and as we move towards higher values, the model tends to overfit, but not by a large amount. This is because as we increase the regularization parameter, the penalty increases on the samples that lie outside the epsilon tube. Hence the model ends up fitting the data with as less miss-classified samples as the epsilon tube allows, as it just now converges to a different optimal value.*

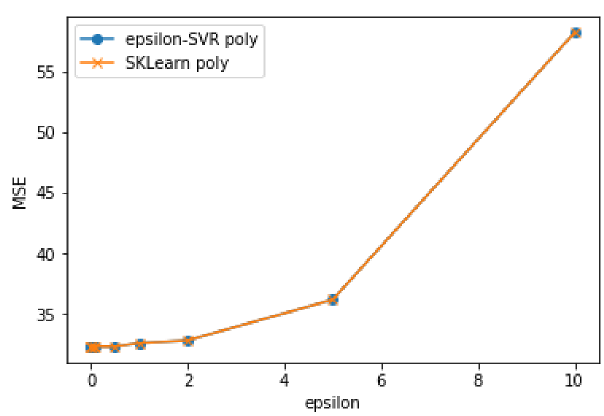
*The following results are obtained and plotted for **Polynomial kernel***

Varying Epsilon

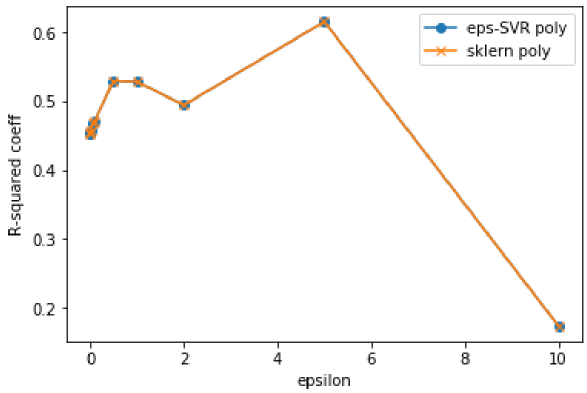
Varying Epsilon in range [0.001,0.005,0.01,0.05,0.1,0.5,1,2,5,10] for , deg=3 (gamma=1 in SKlearn) for different values of C=[0.1,1,10]

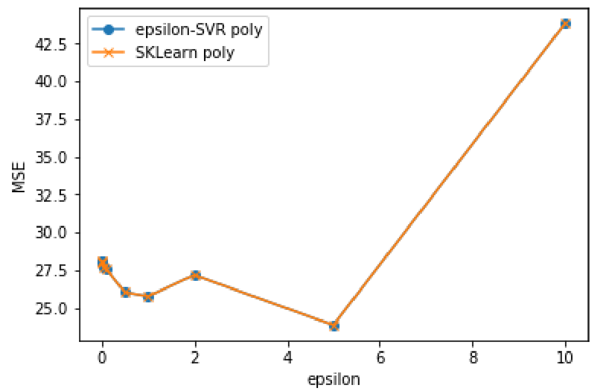
C=0.1



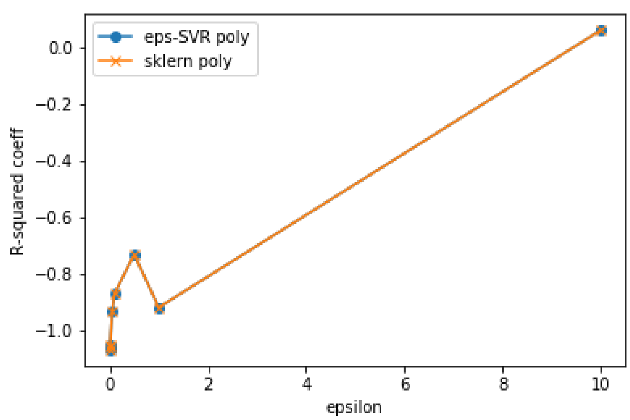


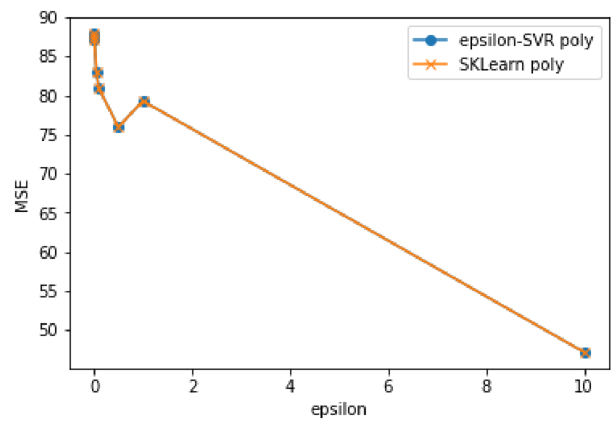
C=1





C=100



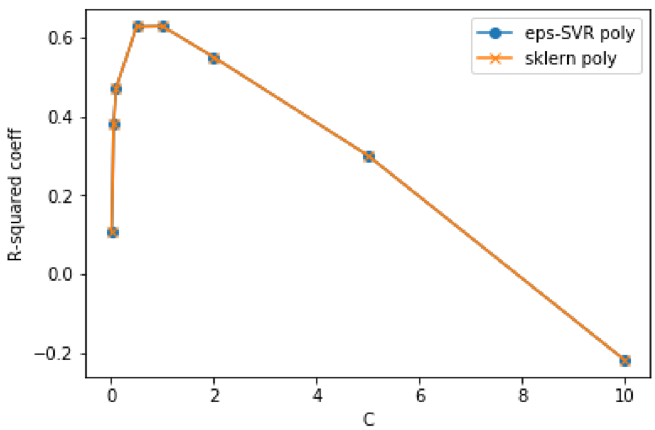


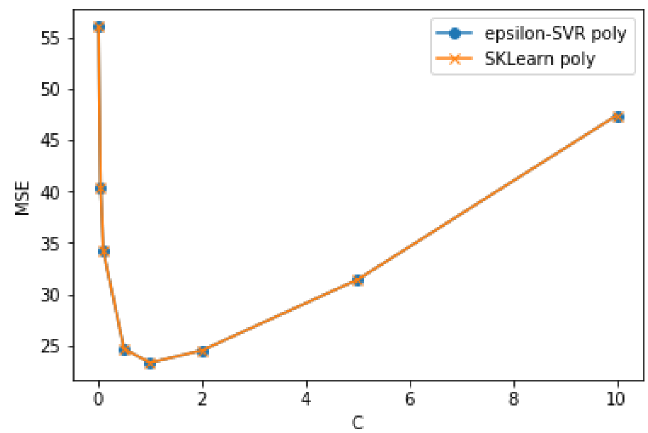
Comments on Varying Epsilon

*We observe that **for a given value of C in the higher region(C>10), the model performance increases as eps increases.** This happens as we start the overfitting regime for low eps , increasing eps relaxes the penalty which increases R2 uptill a certain point, after that, it decreases and for large values epsilon dominates and almost all the points are not penalized hence the model tends to do a little better as we now move to the underfitting regime. **For C in middle range(1<C<10) the R2 coeff increases uptill a certain extent as epsilon increases and then decreases this hppens because we start from the overfitting regime of eps, and then when we increase eps further, the performance goes down a little, this is the local maxima, after that R2 increases with eps again, this happens to be the optimal region between the underfitting and the overfitting regime. As we further increase the value of epsilon, the R2 coeff decreases as we now move to the underfitting regime.** Again for very high values of epsilon, C is ignored and we tend to do a little better. **For higher values of C, as we increase epsilon the R2 value rises marginally as we move from overfitting regime of epsilon to the optimal region, but as we increase epsilon further R2 dips, as we enter the worse of the underfitting regime of both C and epsilon.***

Varying C

Varying C in range [0.01,0.05,0.1,0.5,1,2,5,10] for , deg=3 (gamma=1 in SKlearn) for eps = 4





Comments on Varying C

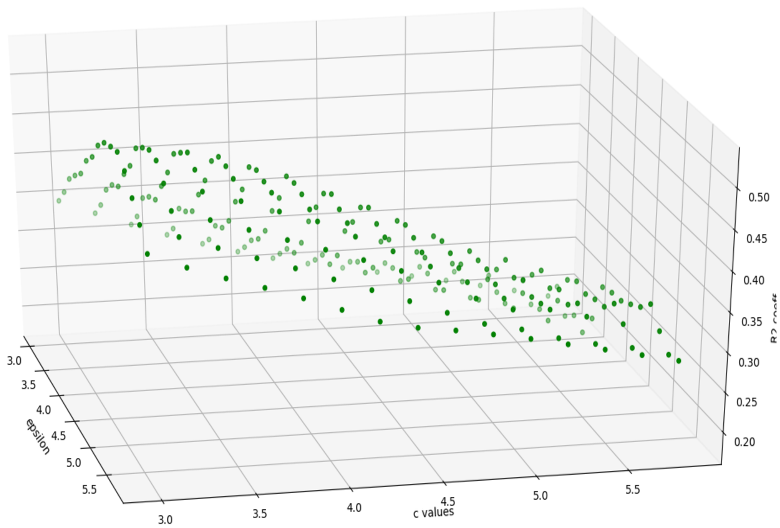
*We observe that for very **low values of C, we tend to be in the overfitting regime** as the R2 values are negative. This happens because the feature space is high dimensional as we are using polynomial kernel so the model complexity is high and as regularization parameter is low we tend to overfit the data. As we increase the value if C, the model performance increases and we tend to be in the **optimal region when C is in [1,4],** as R2 coeff hits a positive high. As we further increase **C >5 we tend to move towards the underfitting regime** as the penalty is too high.*

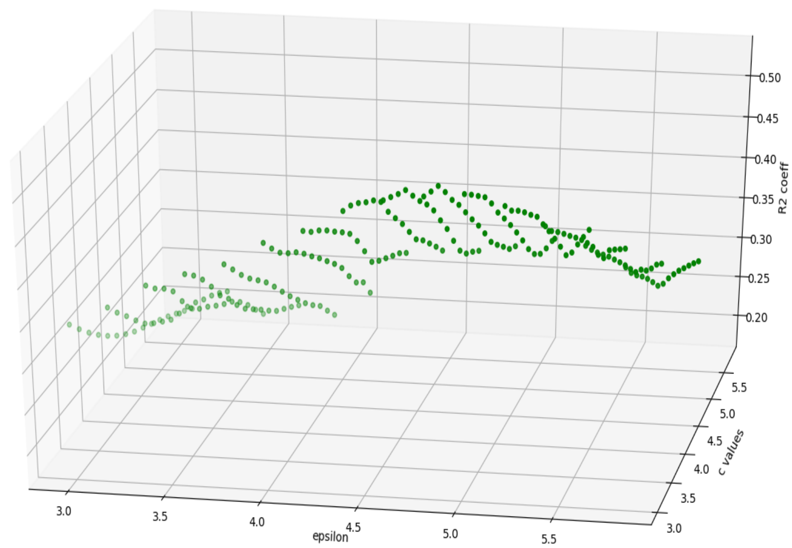
Comparison between CVXOPT implementation and SKlearn

We see that the CVXOPT and SKlearn implementations both give the same results. As can be seen by the R2 and MSE plots the blue line(CVXOPT) and the orange (SKlearn).

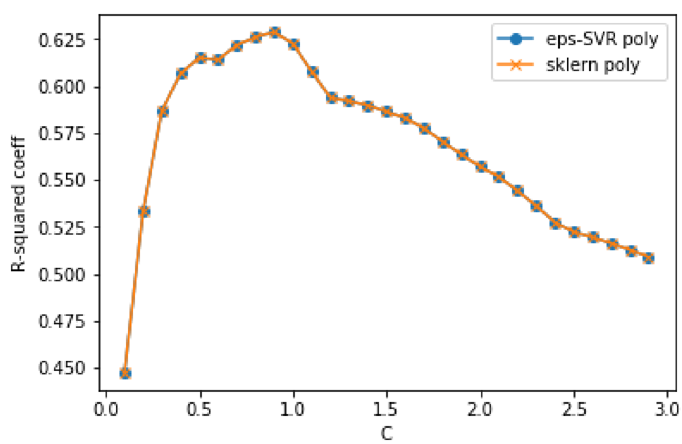
Conclusion : *For the best model performance we need to be in the optimal region of both epsilon and C as R2 is dependent on both C and epsilon.*

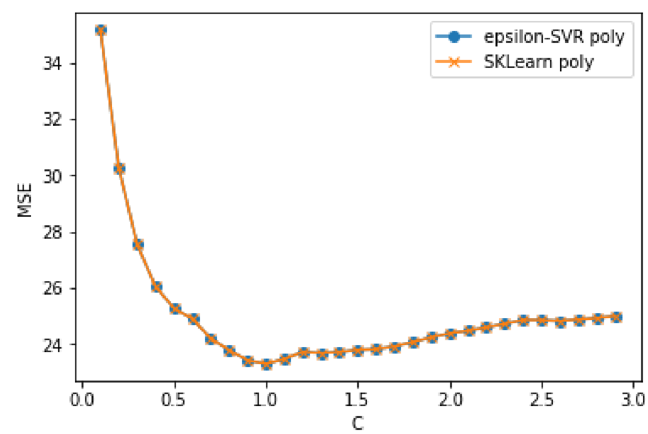
__We now plot the R2 value for epsilon in [3,6] and C in [3,6] for CVXOPT__





__As is it clear from the above 3D plots for a given C value, the R2 coeff takes a maxima at eps=4.5 However for a given epsilon the value of R2 increases as C decreases, so we further decrease our C value below 3 for eps=4.5.__





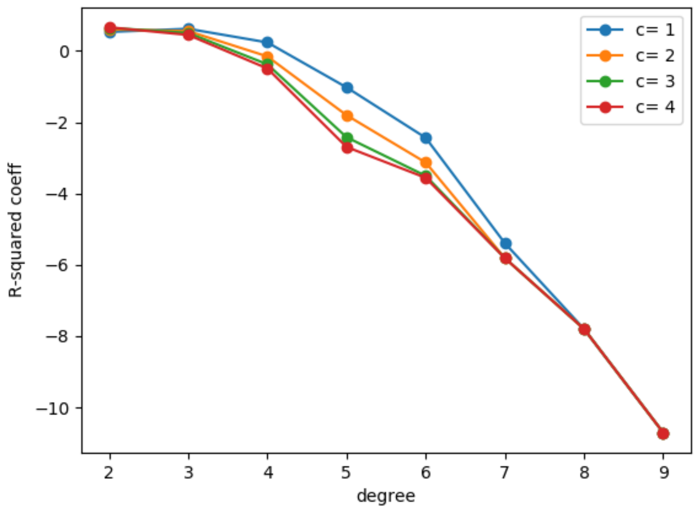
__We get best results at eps =0.5 and C=0.0065 for BOTH CVXOPT and SKlearn__

R2= 0.6225 **kFolds→[0.76702, 0.77837, 0.7747, 0.37505, 0.41734]**

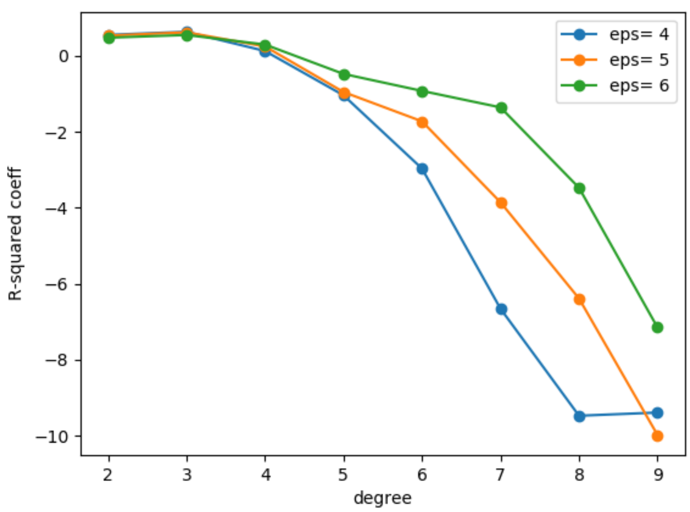
MSE = 23.3147 **kFolds→[8.04588, 20.17663, 18.04391, 54.81515, 15.49191]**

Varying Degree of Poly Kernel

Varying degree from 2 to 10 keeping Eps=4.5 constant plotting for different C



Varying degree from 2 to 9 keeping C=1 constant plotting for different Eps

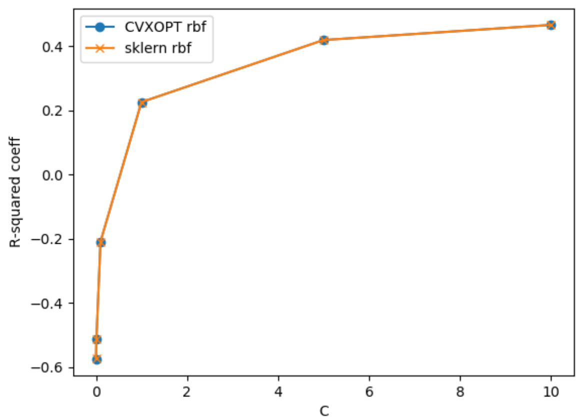


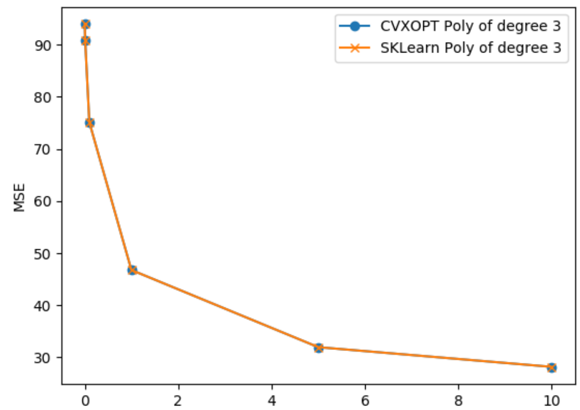
Conclusion : *As we observe that for higher values of degree of the polynomial kernel, for different epsilon and C, as me move towards higher degree the model performance decreases.This happens because of* ***higher degree*** *the model complexity increase and we end up in the* ***overfitting regime.***

The following results are obtained and plotted for ***RBF kernel***

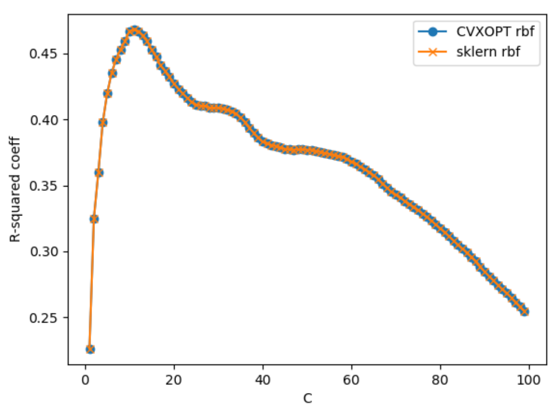
Varying C

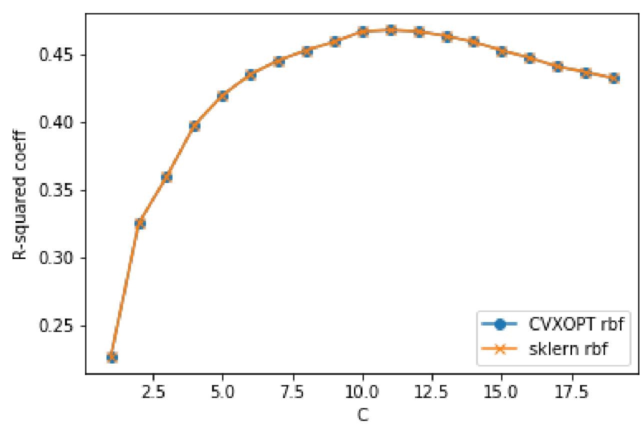
Varying C in [0.001,0.01,0.1,1,5,10] keeping gamma =1 and eps =0.1





We further investigate the region between [10,20] for optimal value of C





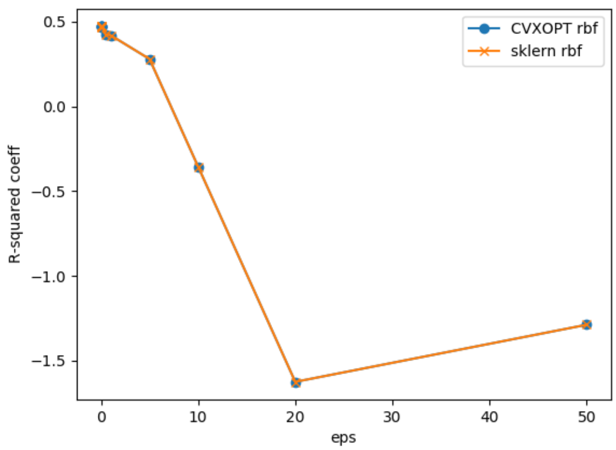
Hence we see that optimal R2 coeff for gamma=1 and eps =0.1 occurs at C=11

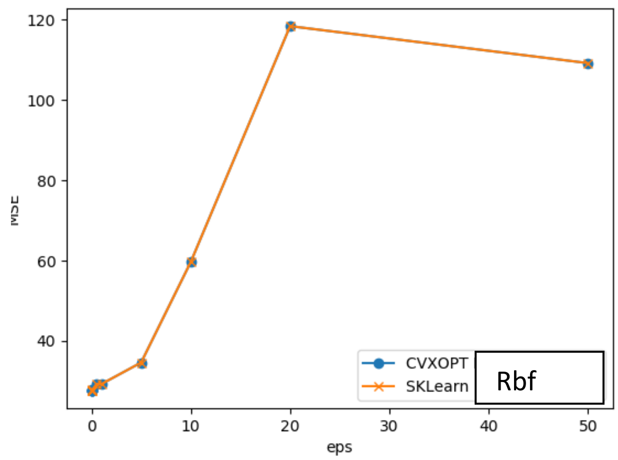
R2= 0.4681 **MSE =27.8304**

Comments on Varying C ***Here we observe that for a given value of eps, increasing the value of C increases the R2 coeff as we start from the overfitting regime due to high model complexity of the rbf kernel and move to the optimal region as C increases. On Further increase C(>40), we move to the overfitting region and R2 coeff becomes negative.***

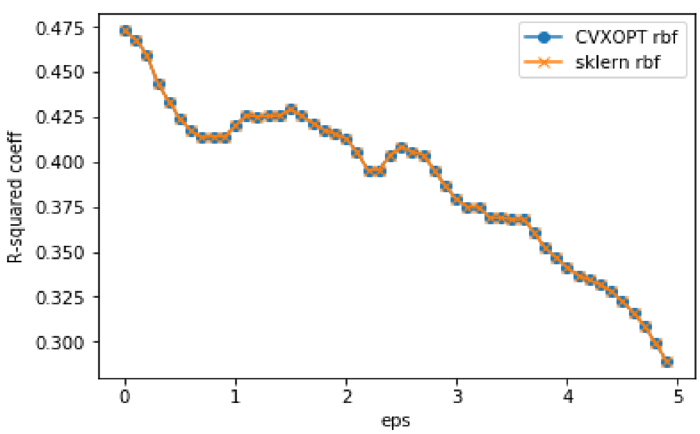
Varying Epsilon

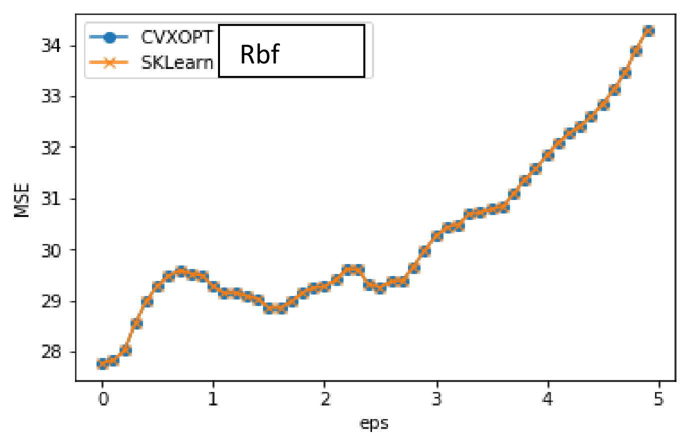
Varying eps in [0.001,0.01,0.1,1,5,10,20,50] with gamma =0.1, C=11





We further investigate the region between [0,5] for optimal value of Epsilon





Hence we see that optimal R2 coeff for gamma=1 and C=11 occurs at eps=0.1

R2= 0.47247 **MSE = 27.78283**

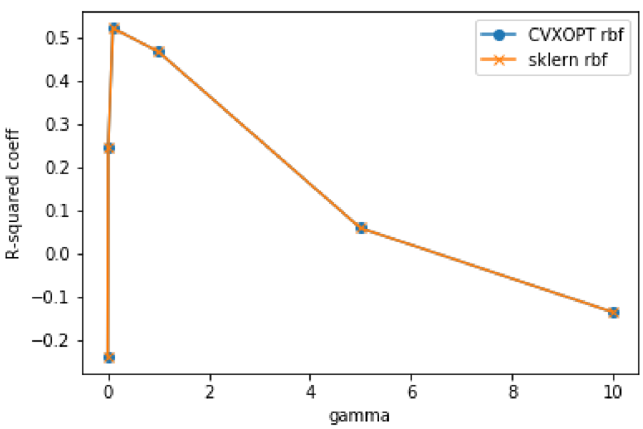
Comments on Varying Epsilon

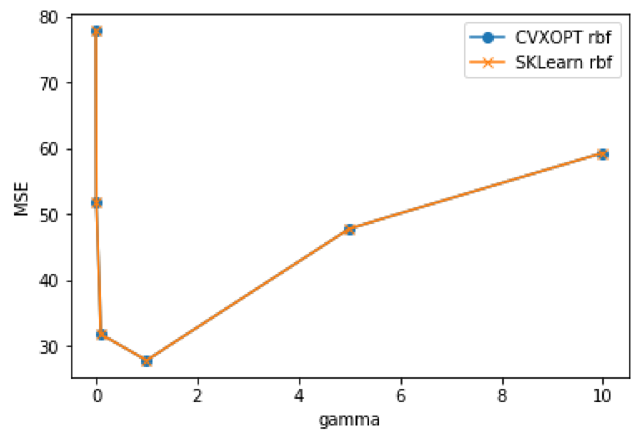
We observe that ***for a given value of C, the value of the R2 coeff increases with increasing eps in the lower region around 0 as the lower value of eps causes over fitting so we move to the optimal region. After a certain point, the value of R2 starts to decrease as epsilon is large***

enough that almost no point is penalized and C value doesn't matter, so we enter the underfitting regime as eps is very large.

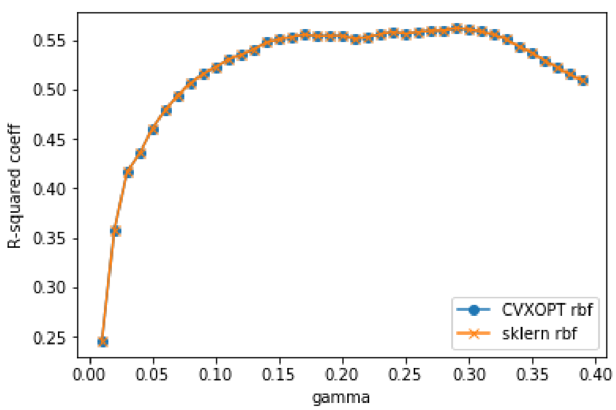
Varying Gamma

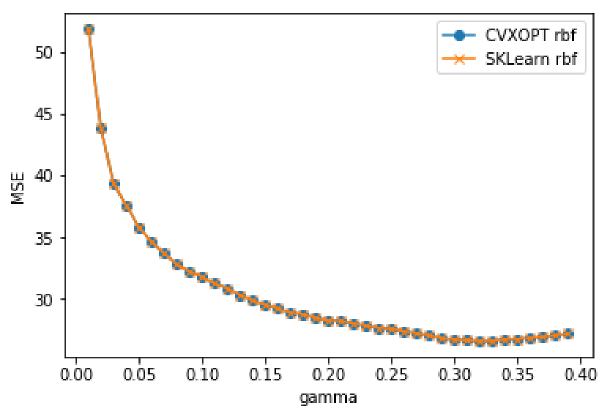
Varying Gamma in [0.001,0.01,0.1,1,5,10] keeping eps =0.1 and C=11





We further investigate the values between [0,0.5] for optimal gamma





Hence we see that the optimal value for both the models occurs at C=11, eps=0.1, gamma=0.32

MSE : 26.5751

cvxopt➔[6.17191 27.29115 24.17537 53.07606 22.30022]

sklearn➔[6.17175 27.29216 24.17515 53.07538 22.30335]

R2 : 0.5550

cvxopt➔[0.82129 0.70022 0.69814 0.39488 0.16128]

sklearn➔[0.82129 0.70021 0.69814 0.39489 0.16116]

Comments on Varying Gamma

We observe that for a given value of C and eps, the value of the R2 coeff increases with increasing gamma in the lower region around 0 as very low values of gamma account for underfitting as the value of variance in the rbf kernel is high, however as we move to larger values of gamma, R2 starts to decrease, as we now move to the overfitting regime as the variance in the rbf kernel density is low.

Comparison of different models and kernels for *Epsilon SVR*

Kernel	CVXOPT		SKlearn	
	R^2	MSE	R^2	MSE
Linear	**0.5079**	**32.5160**	**0.5079**	**32.5160**
Polynomial	**0.62250**	**23.3151**	**0.62250**	**23.3151**
RBF	**0.5550**	**26.5751**	**0.5550**	**26.5751**

Hence we conclude that the CVXOPT and SKlearn implementations both give same results. We get the best performance with polynomial kernel of degree 3 in both models.

2. RH-SVR – RH-SVR aims to find a function *f(x),* where f(x) = <w, x> + b with w ∈ X, b ∈ R, x being the input variable. It does so by considering the equivalent classification problem with the set of data points as (x^i,y+e) with label 1 and (x^i,y-e). This is formulated as the following optimization problem.

$$\min_{\mathbf{w},\delta,\alpha,\beta,\boldsymbol{\xi},\boldsymbol{\eta}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\delta^2 - (\alpha - \beta) + D(\mathbf{e}'\boldsymbol{\xi} + \mathbf{e}'\boldsymbol{\eta})$$
$$\text{s.t.} \quad \mathbf{X}\mathbf{w} + \delta(\mathbf{y} + \varepsilon\mathbf{e}) - \alpha\mathbf{e} + \boldsymbol{\xi} \geqslant 0, \quad \boldsymbol{\xi} \geqslant 0,$$
$$\mathbf{X}\mathbf{w} + \delta(\mathbf{y} - \varepsilon\mathbf{e}) - \beta\mathbf{e} - \boldsymbol{\eta} \leqslant 0, \quad \boldsymbol{\eta} \geqslant 0.$$

However we solve the following dual using CVXOPT as the primal is intractable.
Here we have kernelized the inner product matrix XX' replaced by the kernel matrix K whose ijth entry is k(xi; xj).

$$\min_{\mathbf{u},\mathbf{v}} \quad \frac{1}{2}(\mathbf{u} - \mathbf{v})'(\mathbf{K} + \mathbf{y}\mathbf{y}')(\mathbf{u} - \mathbf{v}) + 2\varepsilon\mathbf{y}'(\mathbf{u} - \mathbf{v})$$
$$\text{s.t.} \quad \mathbf{e}'\mathbf{u} = 1, \quad \mathbf{e}'\mathbf{v} = 1,$$
$$0 \leqslant \mathbf{u} \leqslant D\mathbf{e}, \quad 0 \leqslant \mathbf{v} \leqslant D\mathbf{e},$$

The predictions are made using the following theorem :

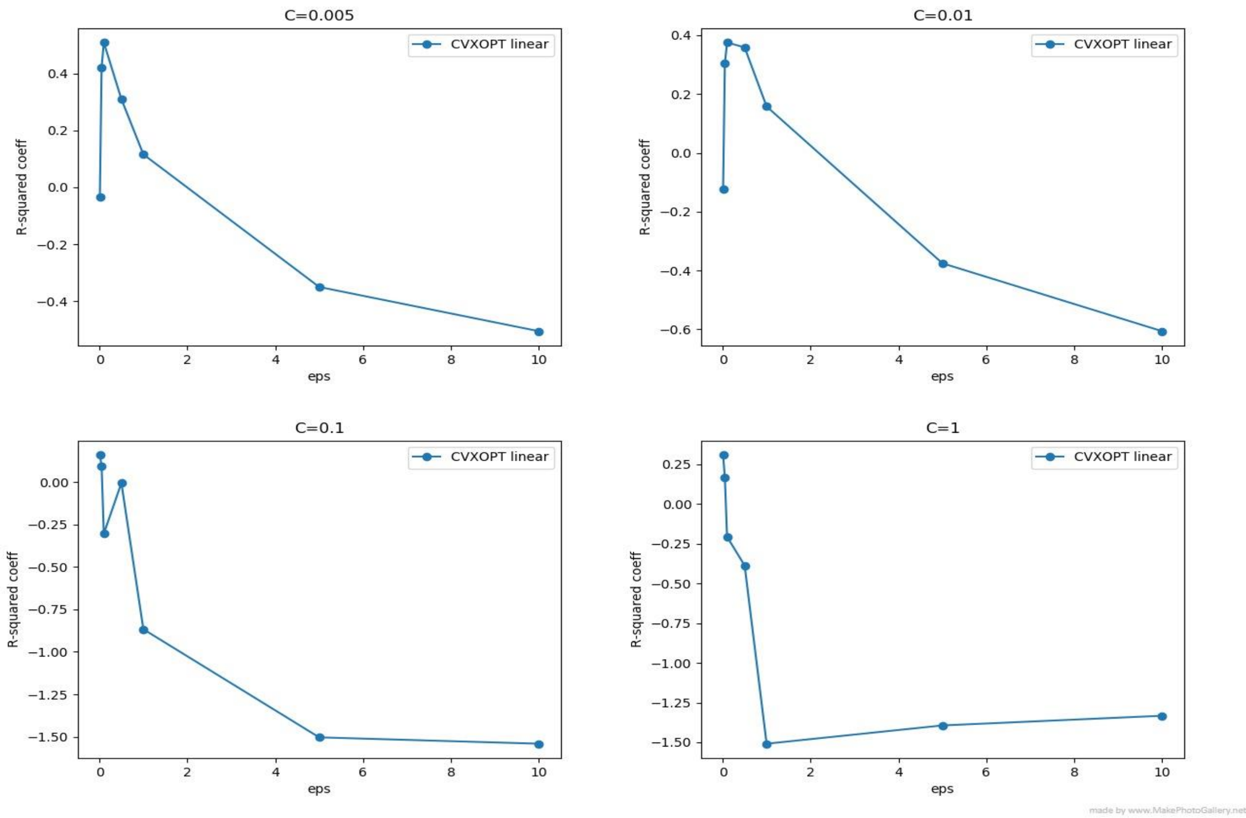
Theorem 6 (Construct $\hat{\varepsilon}$-tube in feature space). *If* $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$ *is the solution to problem* (12), *then the resulting regression model is* $f(\mathbf{x}) = \sum_{i=1}^{\ell}(\bar{v}_i - \bar{u}_i)k(\mathbf{x}_i, \mathbf{x}) + \bar{b}$ *which constructs an* $\hat{\varepsilon}$*-tube in feature space, where* $\bar{u}_i = \hat{u}_i/\hat{\delta}$, $\bar{v}_i = \hat{v}_i/\hat{\delta}$, $\hat{\delta} = (\hat{\mathbf{u}} - \hat{\mathbf{v}})'\mathbf{y} + 2\varepsilon$, *the intercept term* $\bar{b} = (\hat{\mathbf{u}} - \hat{\mathbf{v}})'\mathbf{K}(\hat{\mathbf{u}} + \hat{\mathbf{v}})/(2\hat{\delta}) + (\hat{\mathbf{u}} + \hat{\mathbf{v}})'\mathbf{y}/2$, *and* $\hat{\varepsilon} = -(\hat{\mathbf{u}} - \hat{\mathbf{v}})'\mathbf{K}(\hat{\mathbf{u}} - \hat{\mathbf{v}})/(2\hat{\delta}) + (\hat{\mathbf{v}} - \hat{\mathbf{u}})'\mathbf{y}/2$.

Note : As the RH-SVR model also takes y^i as a feature, we also scale Y_train of our data set to get results. If Y_train is not scaled, predictions are absurd. To calculate R2 and MSE we scale back our predictions.

*The following results are obtained and plotted for **Linear kernel***

Varying Epsilon

Varying epsilon in range [0.01,0.05,0.1,0.5,1,5,10]with different values of C



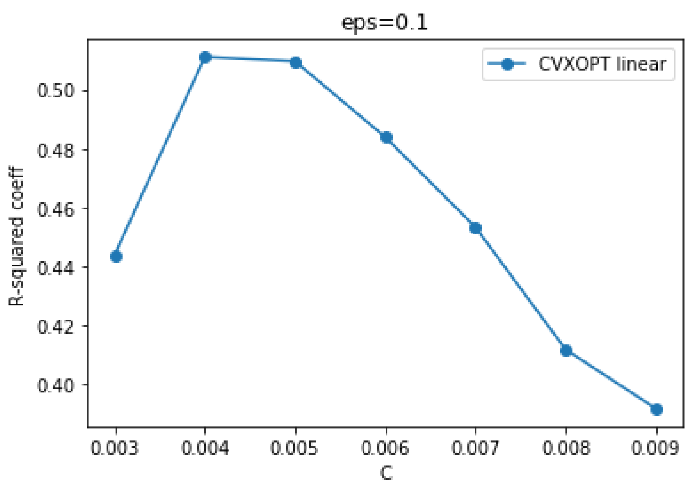
Comments on varying epsilon : here we see that **for lower values of C, as we increase eps from 0.01 to 10, the R2 value increases till eps=0.1 and then decreases.** This happens because **with very low value of eps, we are in the overfitting region,** as the two sets of data points don't have much margin between them. As we increase the value of eps, we move to the underfitting region as large value of eps creates widely spread apart set of data points. **For higher values of C, the value of R2 decreases as eps increases and then fairly becomes constant.** This happens because with fairly high C, we start in the underfitting region of C and overfitting region of eps, so this lands us somewhere in the optimal region. But **as eps is increased we move to the underfitting region and the R2 value becomes negative.**

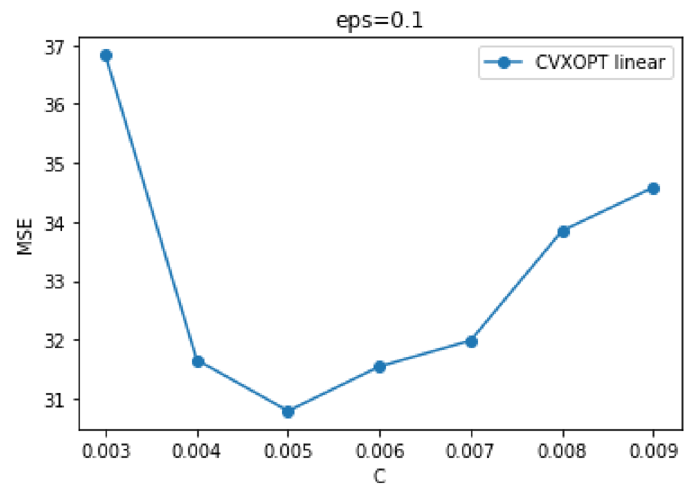
We get best performance at eps =0.1 and C=0.005

R2= 0.50964 **MSE = 32.78634**

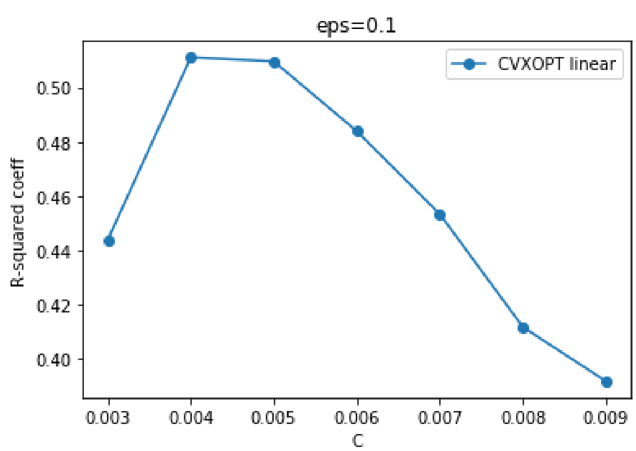
Varying C

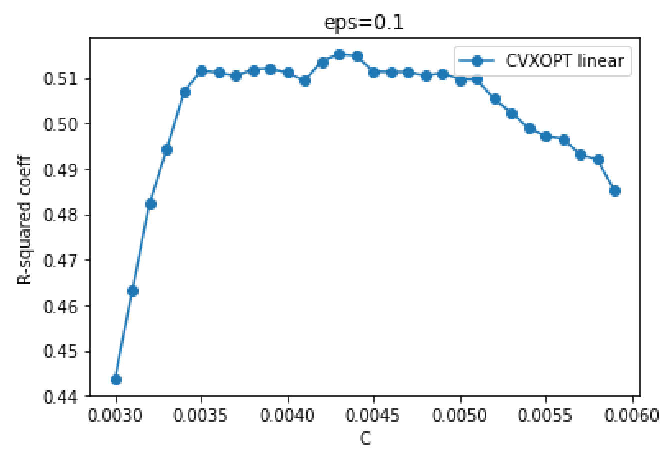
Varying C in range [0.003,0.01] with step size of 0.001 for eps =0.1 to fine tune





We further fine tune between 0.003 and 0.006





Comments on varying C: From the above graphs, we conclude that for **very low values of C** (reg parameter), the model ends up in the **overfitting regime** as the R2 score turns out to be negative. However as we increase the value of C we move towards the optimal region, and as we increase it further we move towards the **underfitting regime**, and R2 falls down.

We get best performance at eps =0.1 and C=0.0043

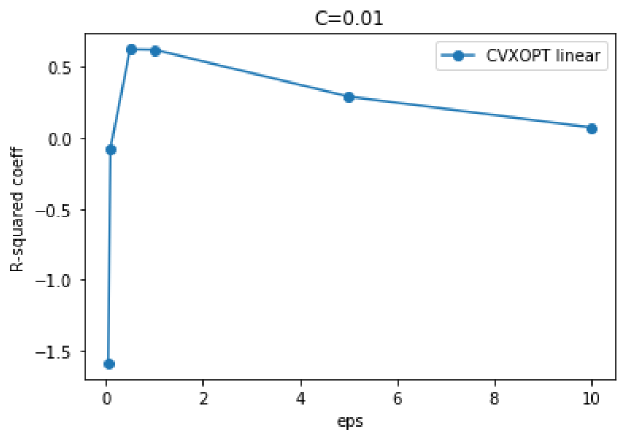
R2= 0.51518 **kFolds→[0.69776, 0.71671, 0.50296, 0.28335, 0.37512]**

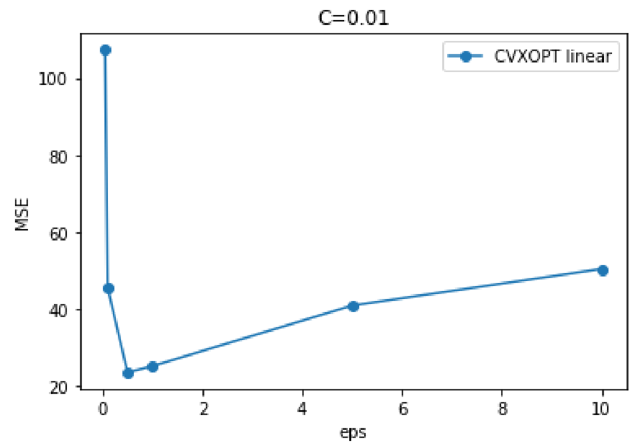
MSE = 31.10163 **kFolds→[10.43793, 25.79013, 39.80682, 62.85866, 16.61461]**

*<u>The following results are obtained and plotted for **Polynomial kernel**</u>*

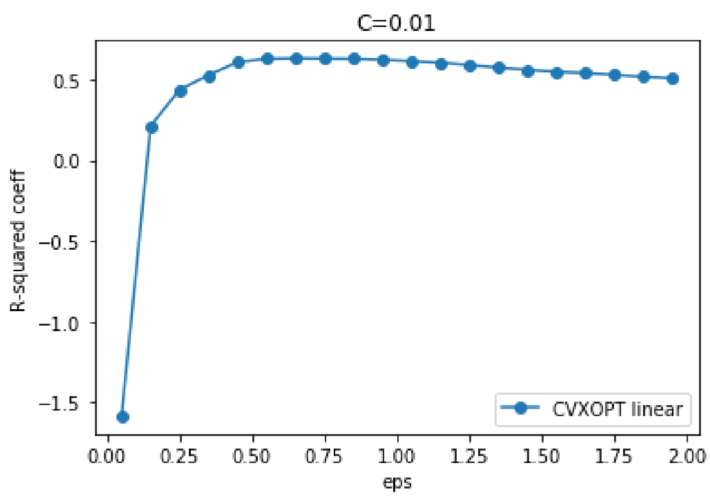
Varying Epsilon

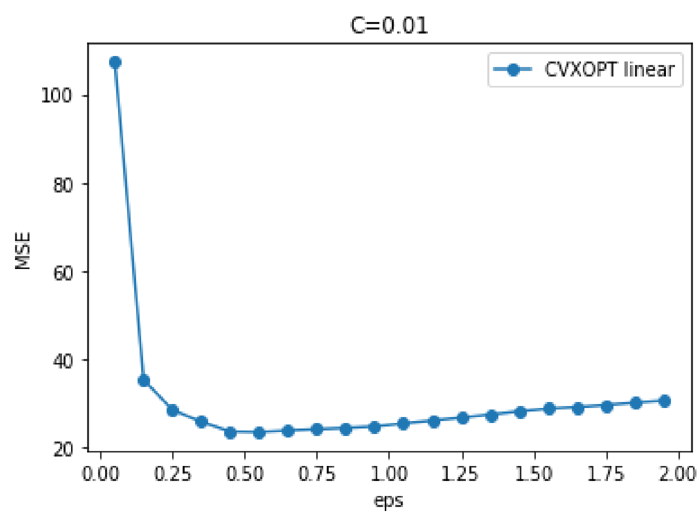
<u>Varying eps in [0.001,0.01,0.1,1,5,10,20,50] with C=0.01, degree=3</u>





We further fine tune eps between 0.05 and 2 with step size of 0.05





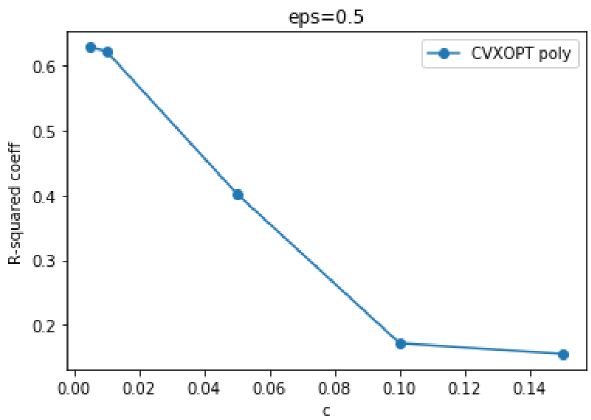
<u>Comments on varying epsilon :</u> here we see that **for very low values of epsilon, the R2 coeff is negative as the model is in the overfitting regime. As we increase the value of eps, we reach the optimal region and increasing epsilon beyond 2 again decreases the value of R2 as we reach the underfitting regime.**

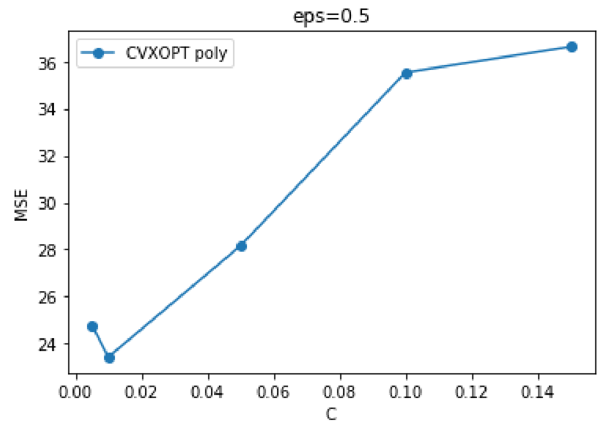
<u>We get best performance at eps =0.5 and C=0.01</u>

R2= 0.62152 **MSE = 23.4245**

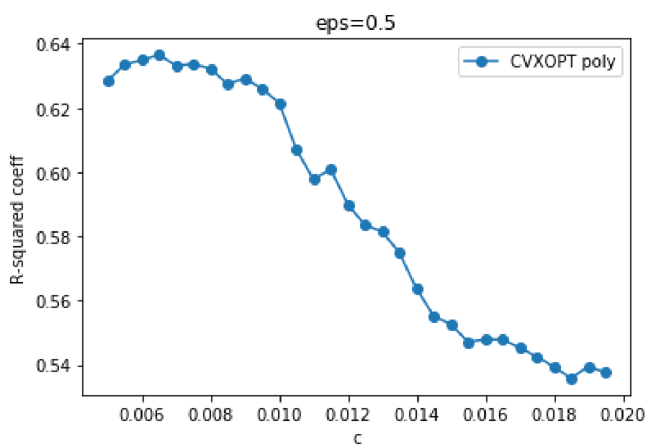
Varying C

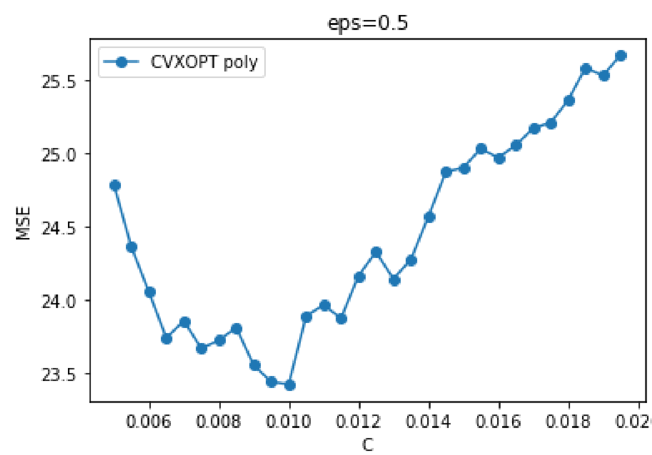
Varying C in range [0.005,0.01,0.05,0.1,0.15] with eps =0.5, degree=3





Varying C in range [0.005,0.02] with step size of 0.005 for eps =0.5 to fine tune





Comments on varying C: From the above graphs, we conclude that for **very low values of C** (C<0.005), the model ends up in the **overfitting regime** as the R2 score turns out to be negative. However as we increase the value of C we move towards the optimal region, and as we increase it further we move towards the **underfitting regime**, and R2 falls down.

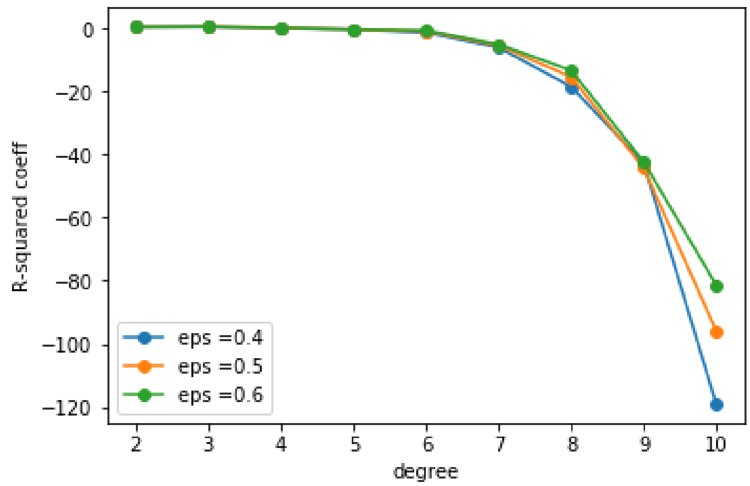
We get best performance at eps =0.5 and C=0.0065

R2= 0.6365 **kFolds→[0.78693, 0.77554, 0.70741, 0.36882, 0.54399]**

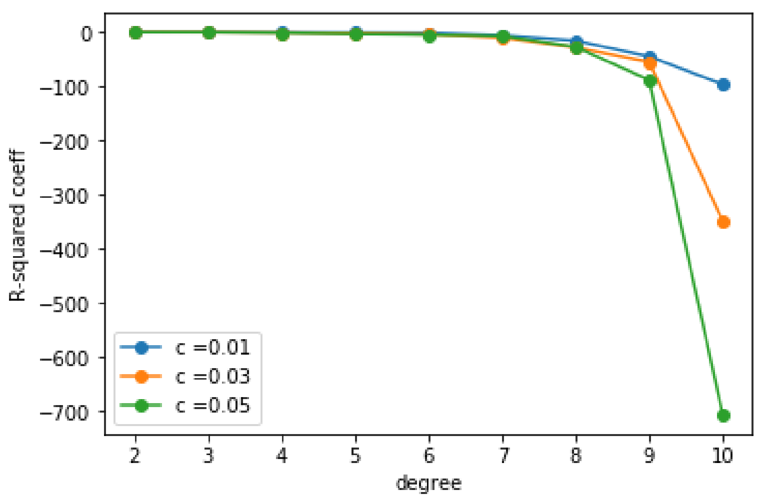
MSE = 23.7424 **kFolds→[7.35855, 20.43394, 23.4326, 55.36232, 12.12464]**

Varying Degree of Poly Kernel

Varying degree in range [2,3,4,5,6,7,8,9,10] for different values of epsilon keeping C constant at C=0.01



Varying degree in range [2,3,4,5,6,7,8,9,10] for different values of C, eps=0.5



Conclusion : *As we observe that for higher values of degree of the polynomial kernel degree >6, for different epsilon and C, as me move towards higher degree the model performance decreases rapidly and becomes large negative. This happens because of **higher degree** the model complexity increase and we end up in the **overfitting regime.***

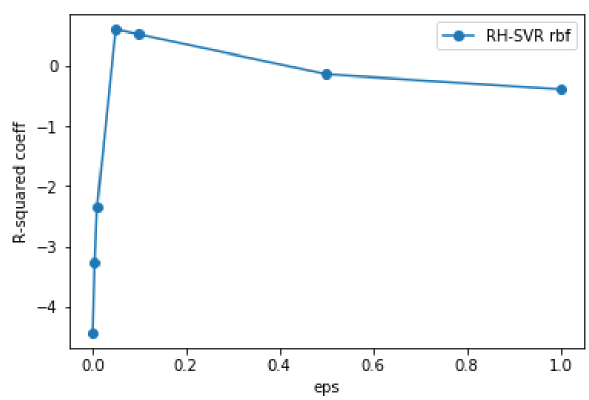
We get best performance for poly ker at eps =0.5 and C=0.0065, degree =3.

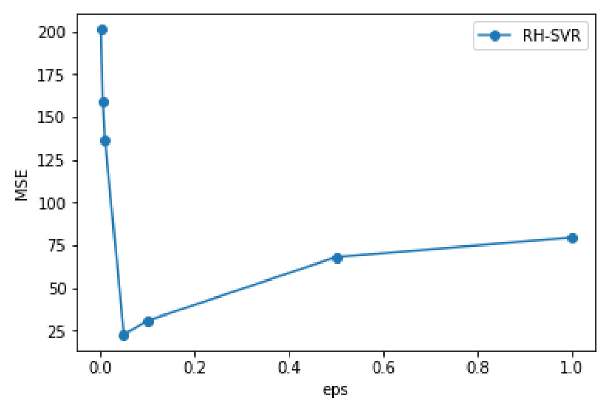
R2= 0.6365   MSE =23.7424

*<u>The following results are obtained and plotted for **Linear kernel**</u>*

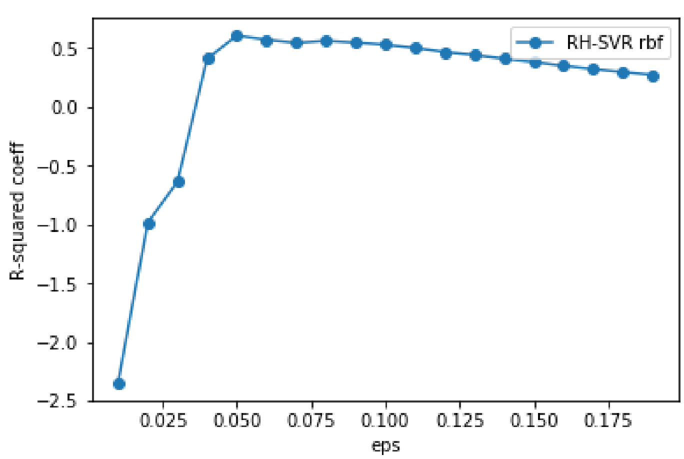
Varying Epsilon

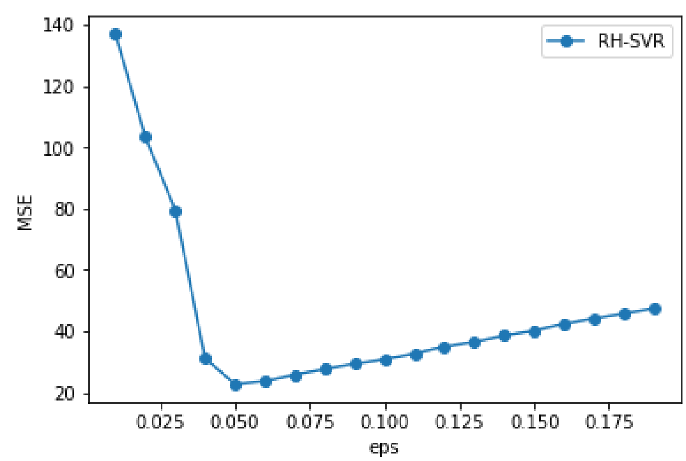
<u>Varying eps in [0.001,0.005,0.01,0.05,0.1,0.5,1] with C=0.005, gamma=0.1</u>





<u>Fine tuning epsilon in the range [0.05,0.2] with step size 0.05</u>





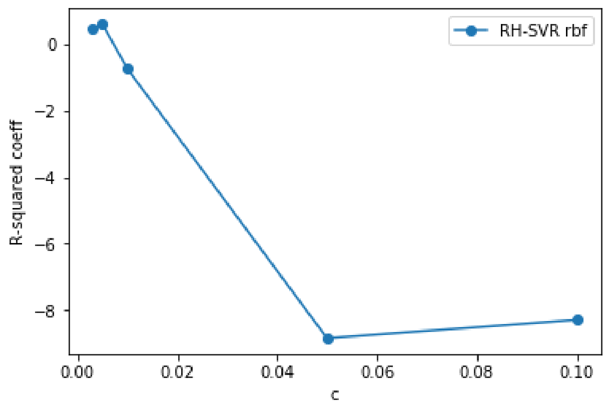
<u>Comments on varying epsilon :</u> here we see that **for very low values of epsilon i.e. eps <0.02, the R2 coeff is negative as the model is in the overfitting regime. As we increase the value of eps, we reach the optimal region i.e. 0.04<eps<0.05 and increasing epsilon beyond 0.2 again causes a significant decrease in the R2 coeff as we now move towards the underfitting regime.**

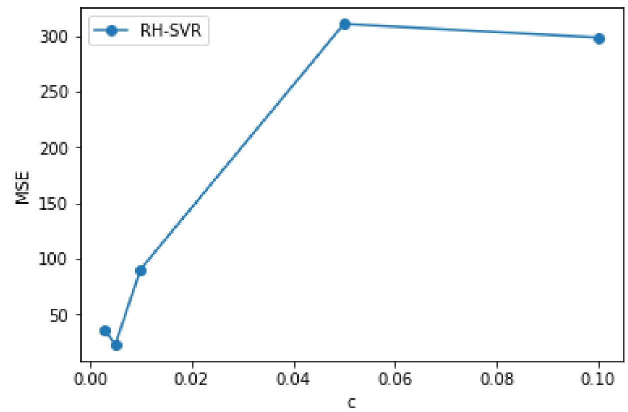
<u>Best R2 value at eps = 0.05 for c=0.005, gamma=0.1</u>

R2 = 0.60765 **MSE = 22.68484**

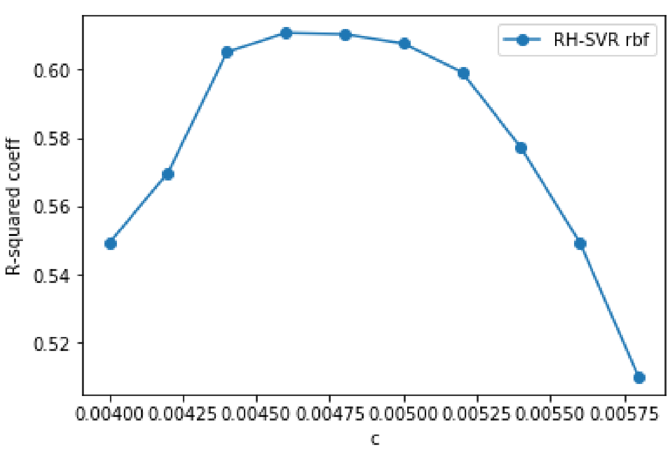
Varying C

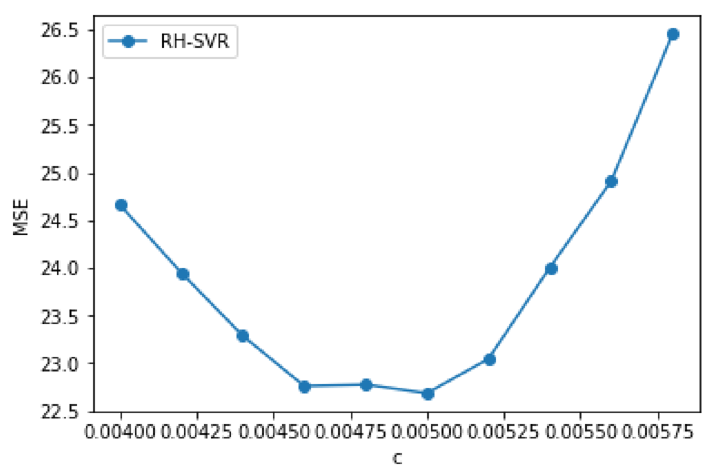
We now vary C in [0.003,0.005,0.01,0.05,0.1] with eps =0.05, gamma=0.1





We fine tune C in [0.004,0.006] with step size of 0.0002





Comments on Varying C

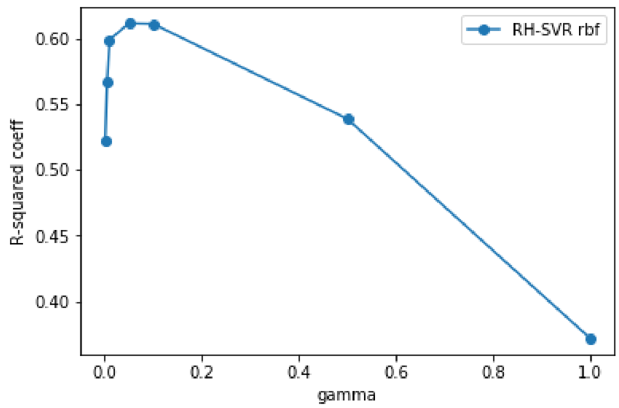
Here we observe that for a given value of eps, increasing the value of C increases the R2 coeff as we start from the overfitting regime(C<0.003) due to high model complexity of the rbf kernel and move to the optimal region(0.004<c<0.006) as C increases. On Further increase C(>0.01), we move to the overfitting region and R2 coeff becomes negative.

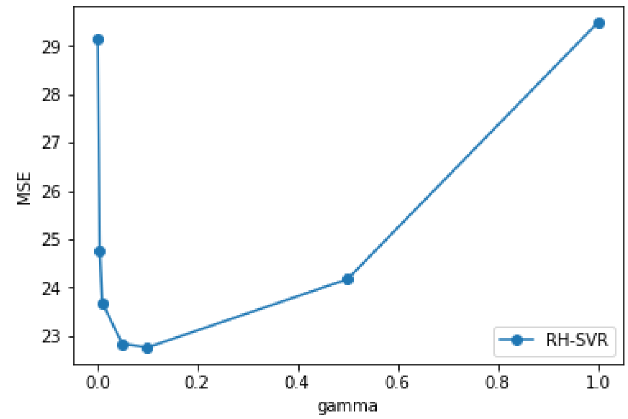
Hence we see that optimal R2 coeff for gamma=0.1 and eps =0.05 occurs at C=0.0046

R2= 0.61079 **MSE = 22.75965**

Varying Gamma

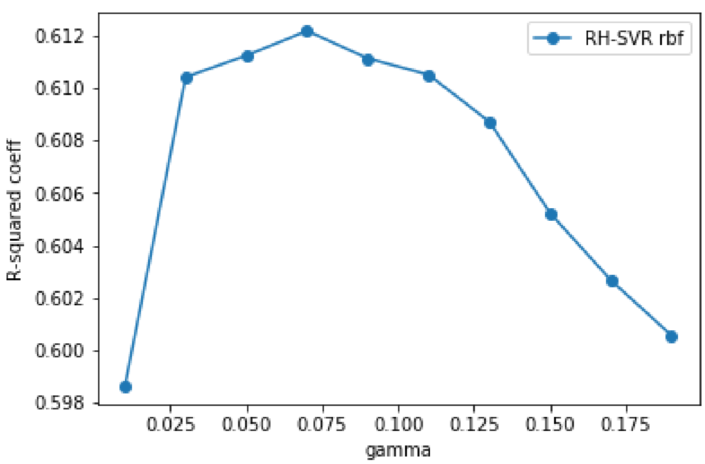
We now vary gamma in [0.001,0.005,0.01,0.05,0.1,0.5,1] with eps =0.05,

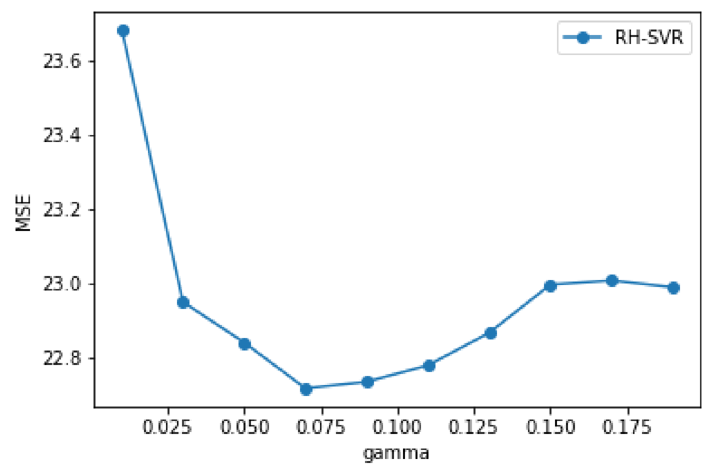




C=0.0046

We fine tune gamma in [0.01,0.2] with step size of 0.002





Comments on Varying Gamma

We observe that for a given value of C and eps, the value of the R2 coeff increases with increasing gamma(0.01<gamma<0.07) in the lower region(<0.001) as very low values of gamma account for underfitting as the value of variance in the rbf kernel is high, however as we move to larger values of gamma(>0.1), R2 starts to decrease, as we now move to the overfitting regime as the variance in the rbf kernel density is low.

We get best performance at eps =0.05 and C=0.0046, gamma = 0.075

R2= 0.6121 **kFolds➔[0.77531, 0.81369, 0.81871, 0.37047, 0.28234]**

MSE = 22.7077 **kFolds➔[7.75958, 16.96076, 14.51938, 55.21743, 19.0815]**

Comparison of different models and kernels

Kernel	Epsilon-SVR		RH-SVR	
	R^2	MSE	R^2	MSE
Linear	0.5079	32.5160	0.50964	32.7863
Polynomial	0.62250	23.3151	0.6365	23.7424
RBF	0.5550	26.5751	0.6121	22.7077

Hence we see that both EPSILON-SVR and RH-SVR give comparable results with poly ker of degree 3 giving best R2 value in both models.

THANK YOU FOR YOUR TIME!

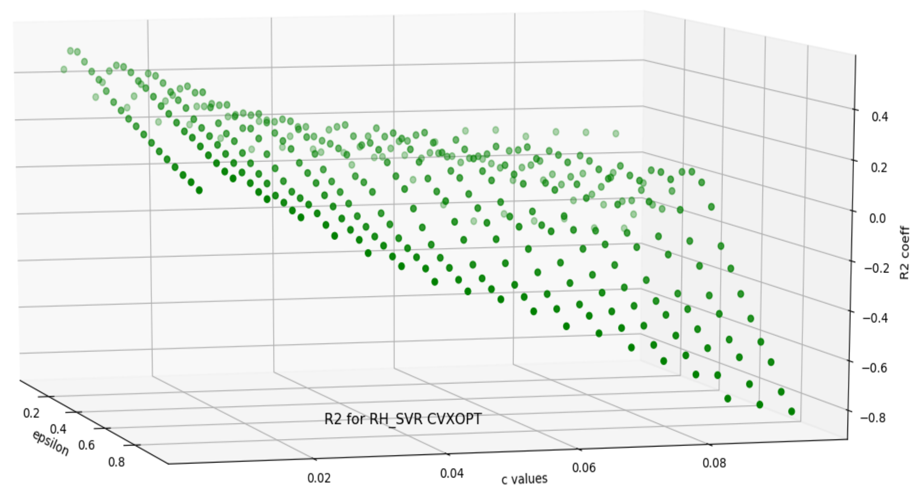
.

.

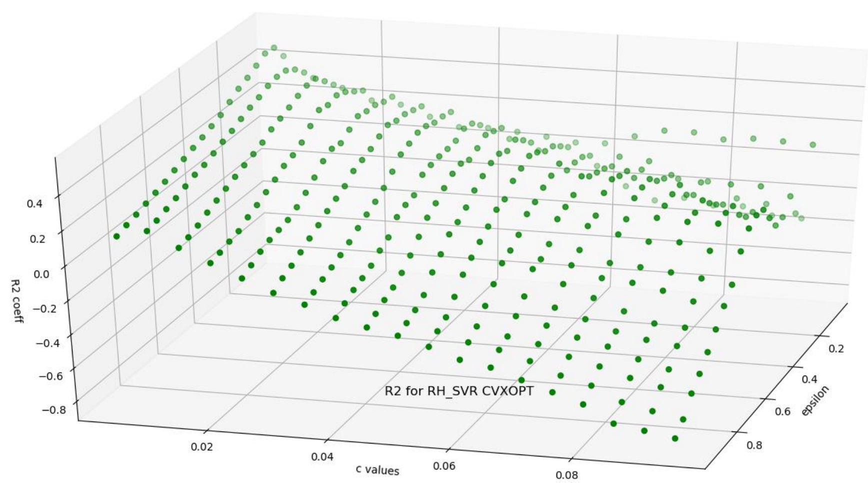
.

TURN OVER TO THE NEXT PAGE FOR SOME BONUS 3-D PLOTS :D

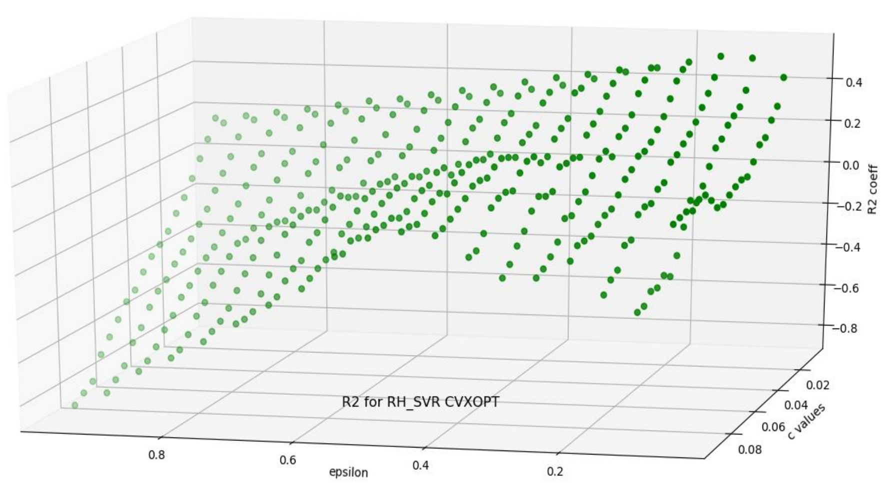
For Linear kernel RH-SVR



R2 for RH_SVR CVXOPT
epsilon
c values
R2 coeff



R2 for RH_SVR CVXOPT
epsilon
c values
R2 coeff



R2 for RH_SVR CVXOPT
epsilon
c values
R2 coeff