ELL-409 ASSIGNMENT

REPORT

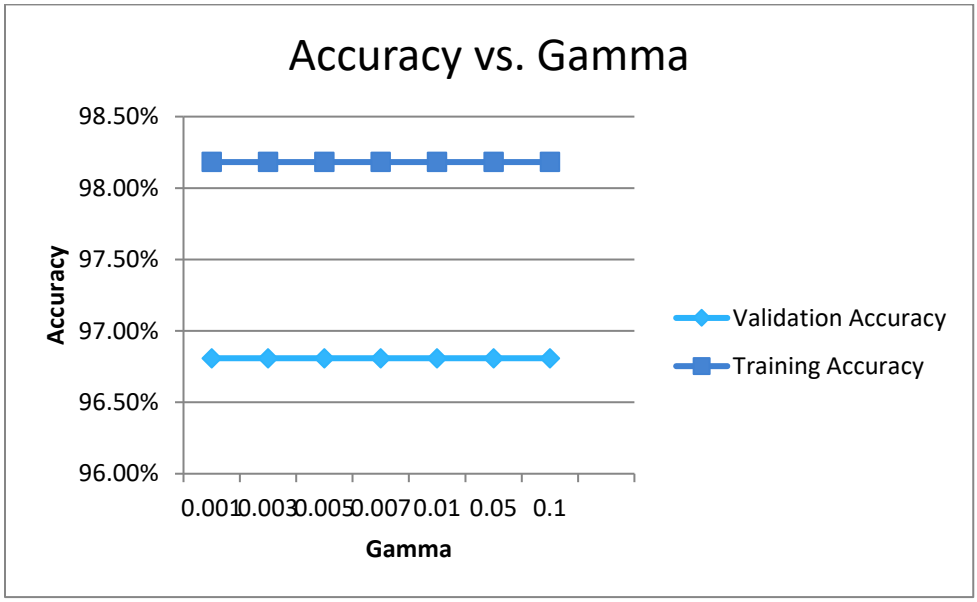
SOURAV

2019CS10404

- **Part 1**
  - **Part 1A**
    - ✓ **Binary class**
    - ✓ **Multi class**
  - **Part 1B**
  - **Part 1C**
- **Part 2**

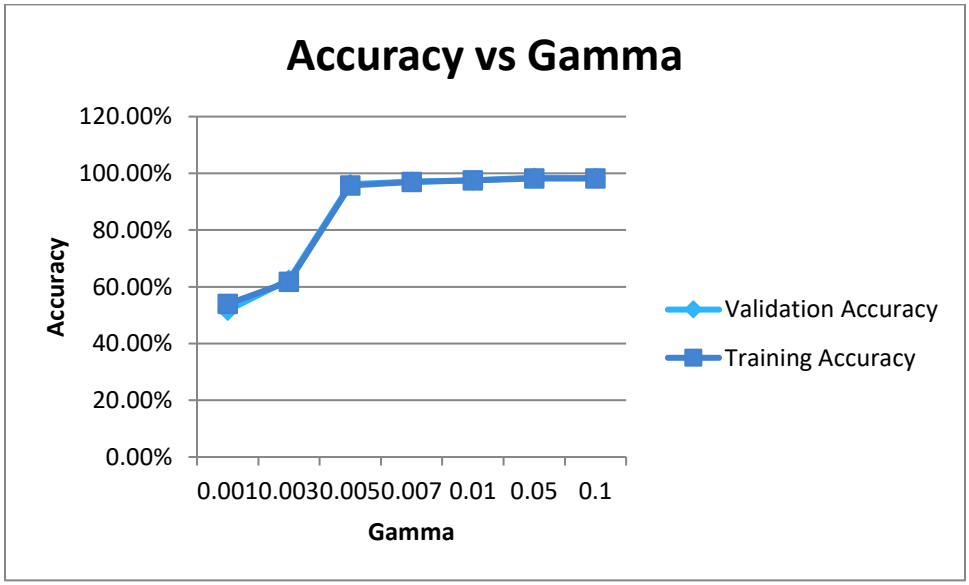
A.1) BINARYCLASS :

Kernel_type = 0 ,Class_Used = (1,2)



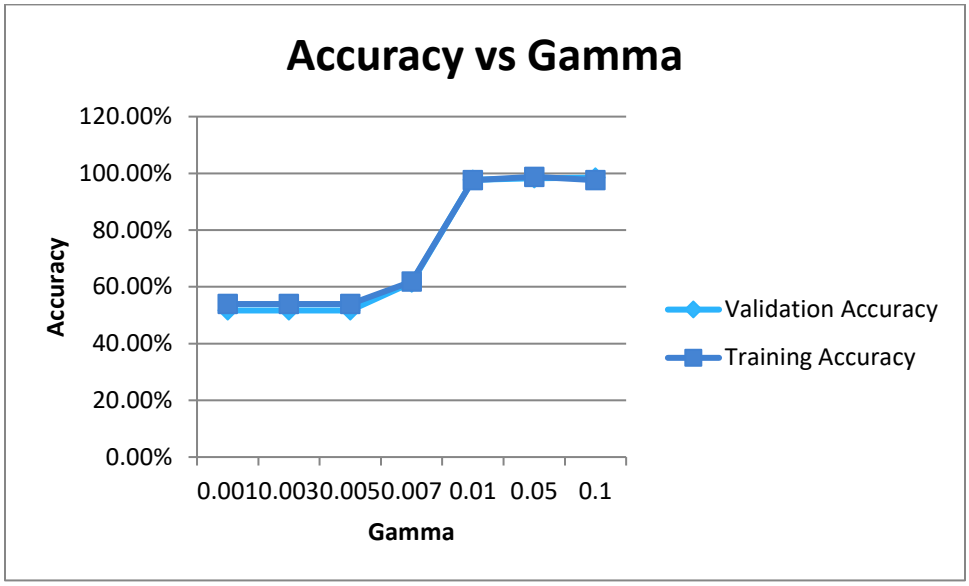
When we use Kernel_type = 0 i.e Linear function then with changing gamma doesn't change because we know that the Linear Kernel function does not depend on the Gamma and we can easily able to see that In Linear Kernel the error training is less than the validation error is also low so this is an example of Good fitting.

Kernel_type = 1 ,Class_Used = (1,2),Degree = 2



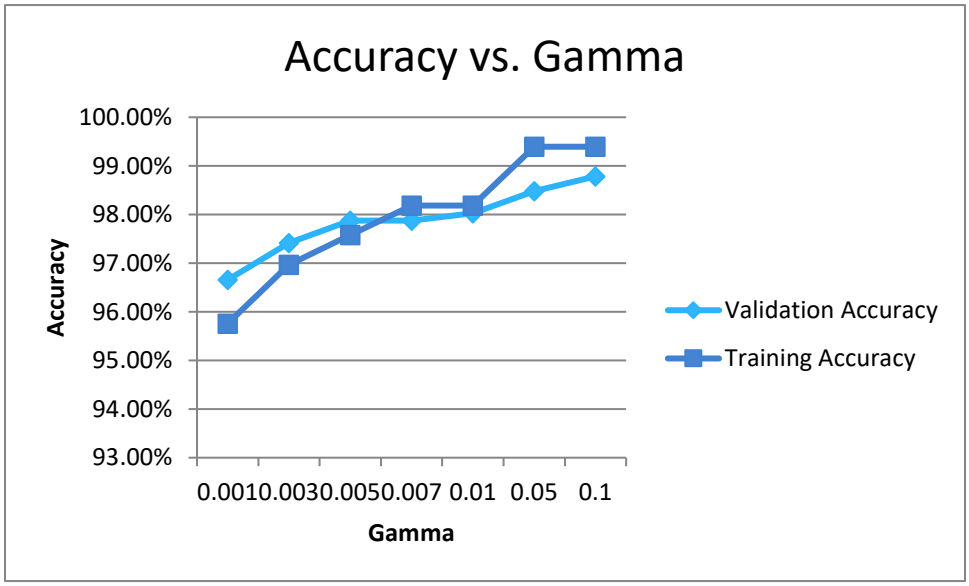
When we use Kernel_type = 1 i.e Polynomial function with degree = 2 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.005 so out curve is at underfitting of the curve and the error is less for Gamma>=0.005 so our curve is at Good Fitting.

Kernel_type = 1 ,Class_Used = (1,2),Degree = 3



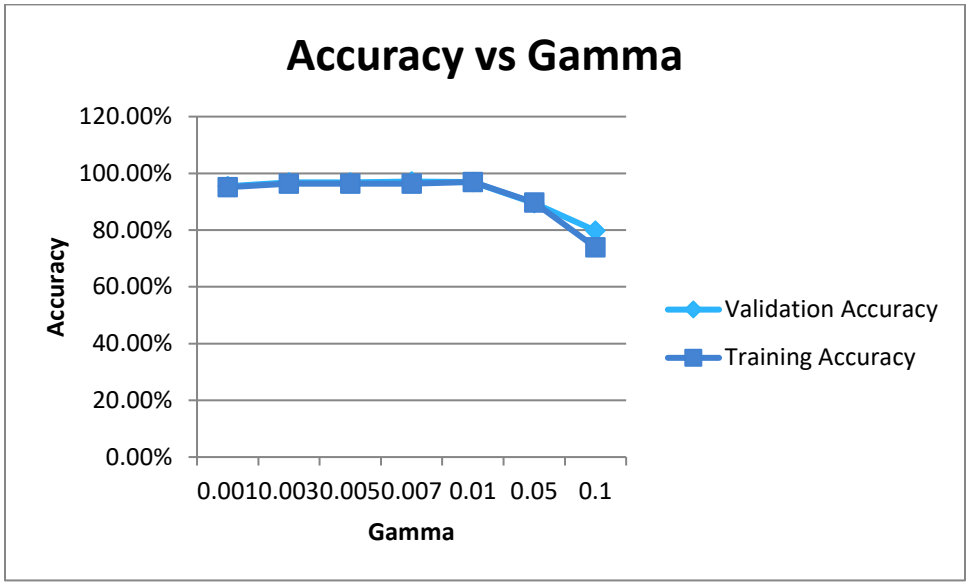
When we use Kernel_type = 1 i.e Polynomial function with degree = 3 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.01 so out curve is at underfitting of the curve and the error is less for Gamma>=0.01 so our curve is at Good Fitting.

Kernel_type = 2 ,Class_Used = (1,2)



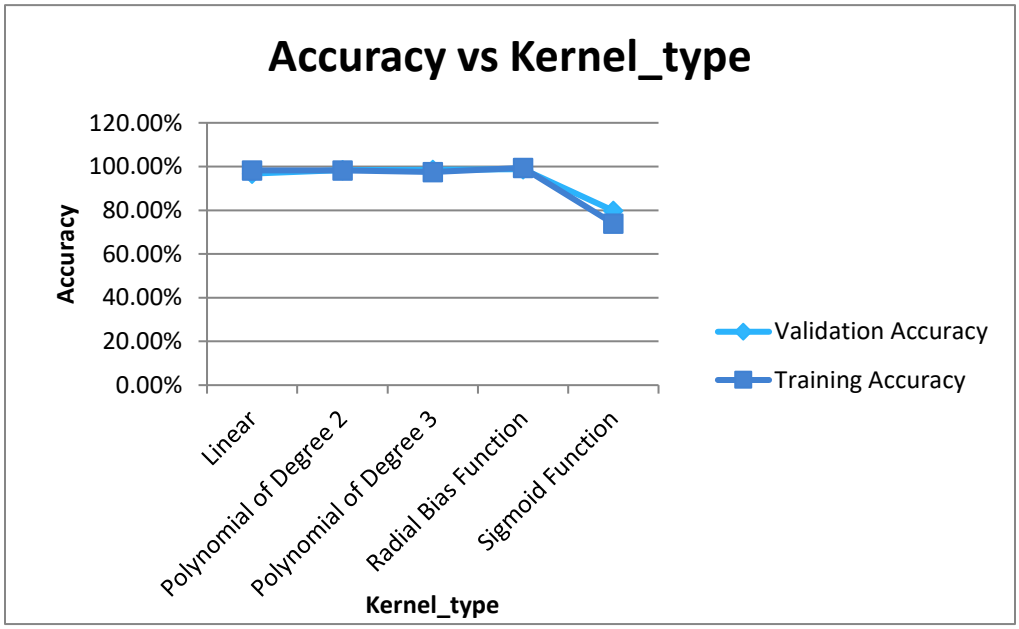
When we use Kernel_type = 2 i.e Radial Base Function then it changes because we know that the Radial Base Kernel function depends on the Gamma and we can easily able to see that In Radial Base Kernel the error training and validation is approx similar for every value of gamma and the error is decreasing for increasing the value of gamma But thus the error is not so much bad i.e accuracy is > 95 => so our curve is at Good fitting.

Kernel_type = 3 ,Class_Used = (1,2)



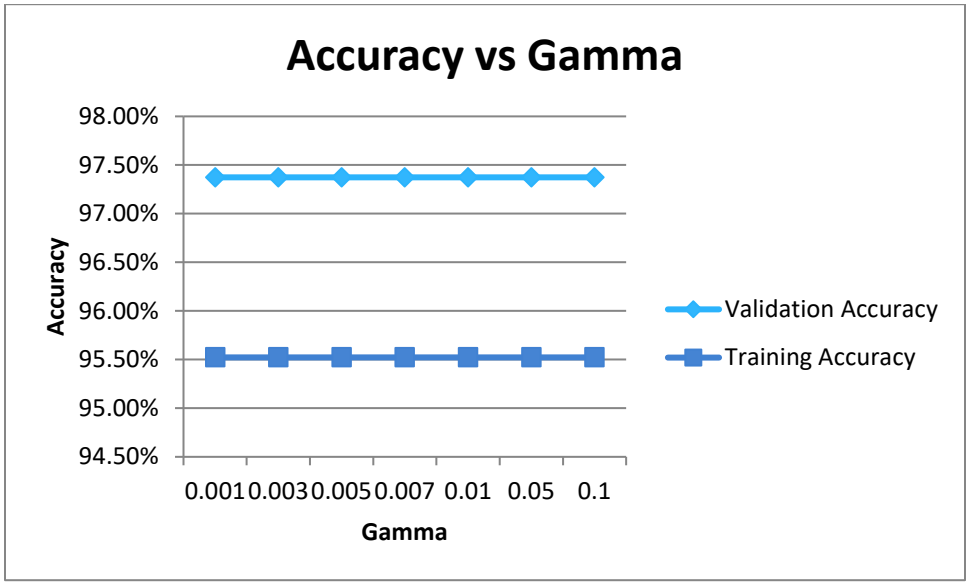
When we use Kernel_type = 3 i.e Sigmoid Function then it changes because we know that the Sigmoid Kernel function depends on the Gamma and we can easily able to see that In Sigmoid Kernel the error training and validation is approx similar for every value of gamma and the error is increasing for increasing the value of gamma => the error is low for gamma<=0.05 so our curve is at Good fitting. The error is high for gamma>0.05 so our curve is underfitting.

Gamma = 0.1, Class_Used = (1,2)



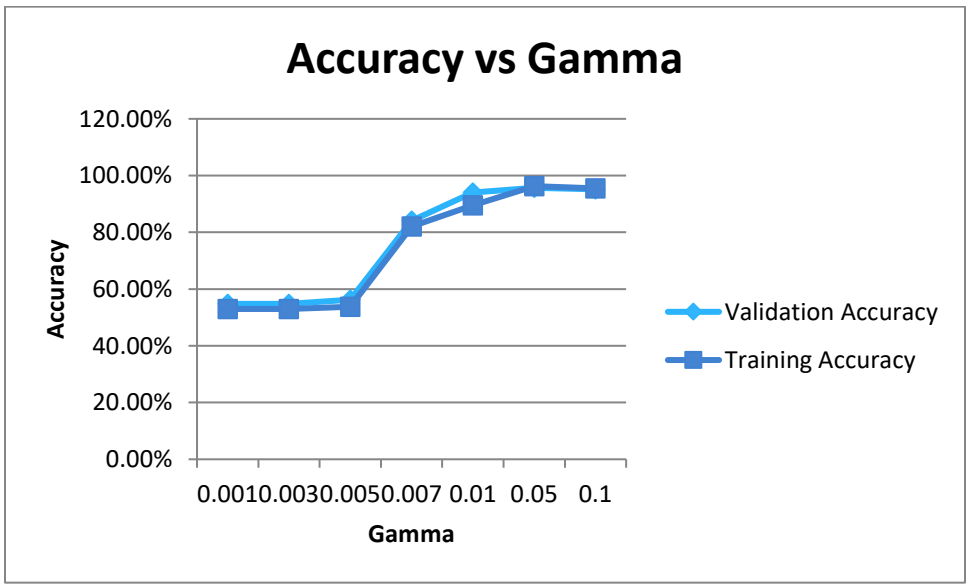
From the above curve, the gamma is fixed for all the kernels i.e 0.01 for this gamma value. Both validation and training error is approximately same except the sigmoid kernel all have the low error => all at the Good fitting while in the sigmoid kernel the error is high => our model is at Underfitting.

Kernel_type = 0 ,Class_Used = (4,5)



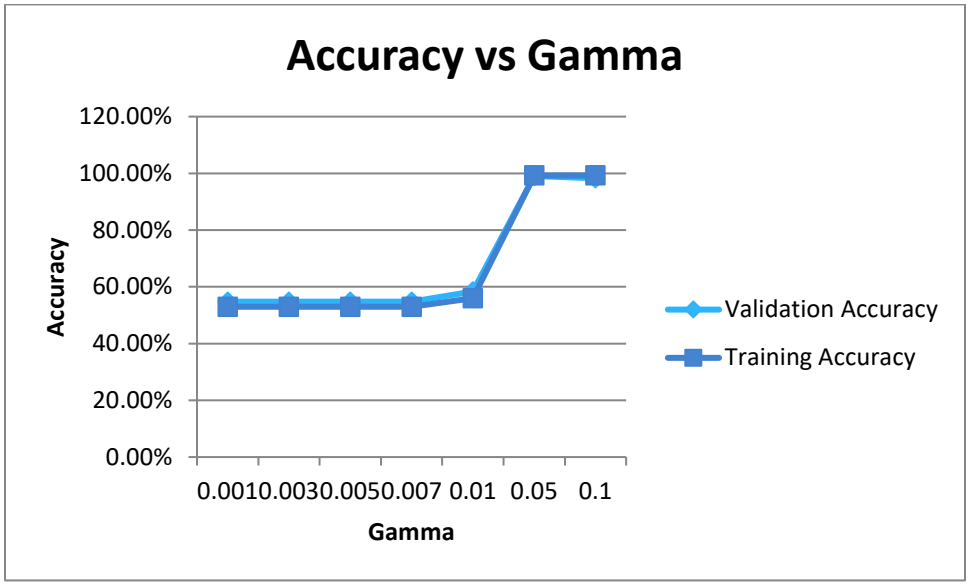
When we use Kernel_type = 0 i.e Linear function then with changing gamma doesn't change because we know that the Linear Kernel function does not depend on the Gamma and we can easily able to see that In Linear Kernel the error training is less than the validation error is also low so this is an example of Good fitting.

Kernel_type = 1 ,Class_Used = (4,5),Degree =2



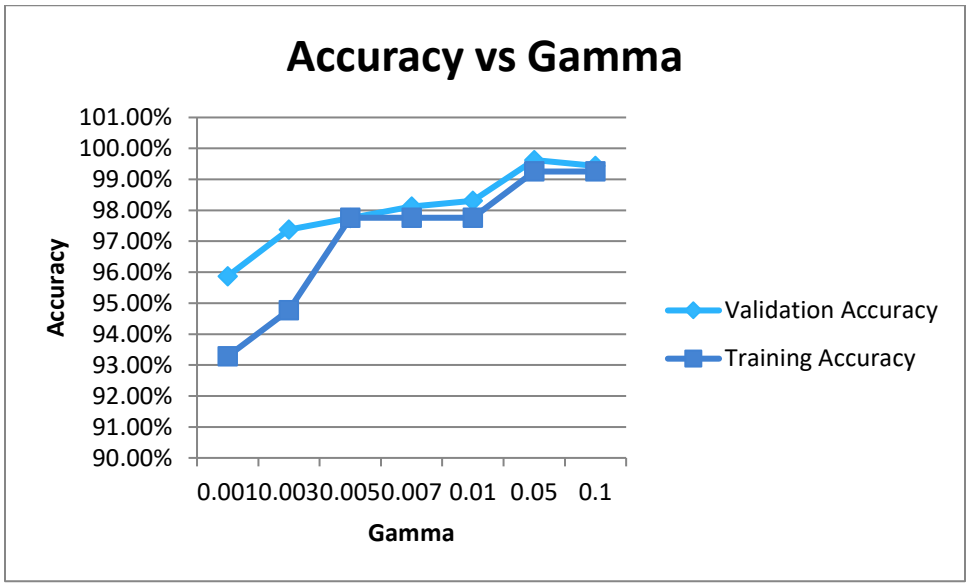
When we use Kernel_type = 1 i.e Polynomial function with degree = 2 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.007 so out curve is at underfitting of the curve and the error is less for Gamma>=0.007 so our curve is at Good Fitting.

Kernel_type = 1 ,Class_Used = (4,5),Degree = 3



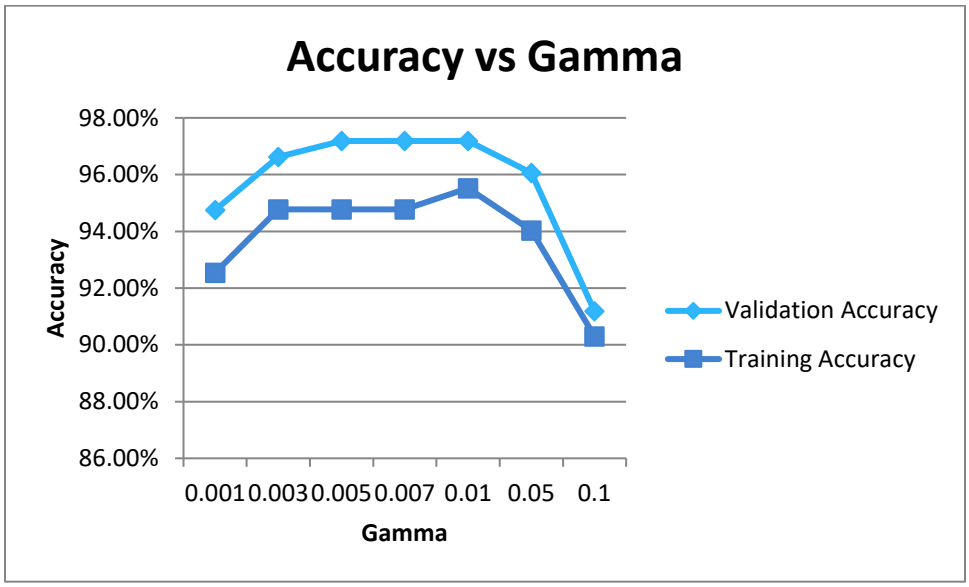
When we use Kernel_type = 1 i.e Polynomial function with degree = 3 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.05 so out curve is at underfitting of the curve and the error is less for Gamma>=0.05 so our curve is at Good Fitting.

Kernel_type = 2 ,Class_Used = (4,5)



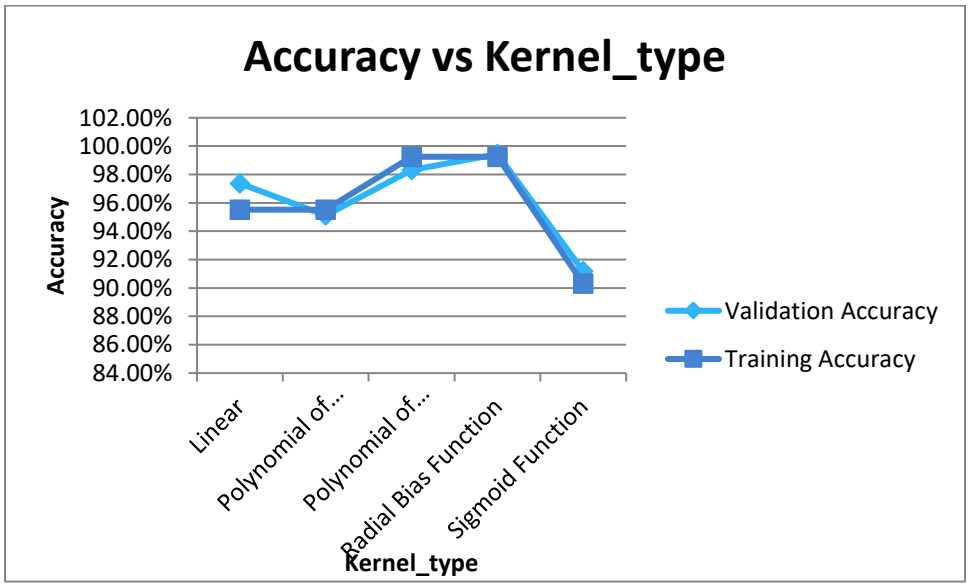
When we use Kernel_type = 2 i.e Radial Base Function then it changes because we know that the Radial Base Kernel function depends on the Gamma and we can easily able to see that In Radial Base Kernel the error training and validation is approx similar for every value of gamma and the error is decreasing for increasing the value of gamma But thus the error is not so much bad i.e accuracy is > 93 => so our curve is at Good fitting.

Kernel_type = 3 ,Class_Used = (4,5)



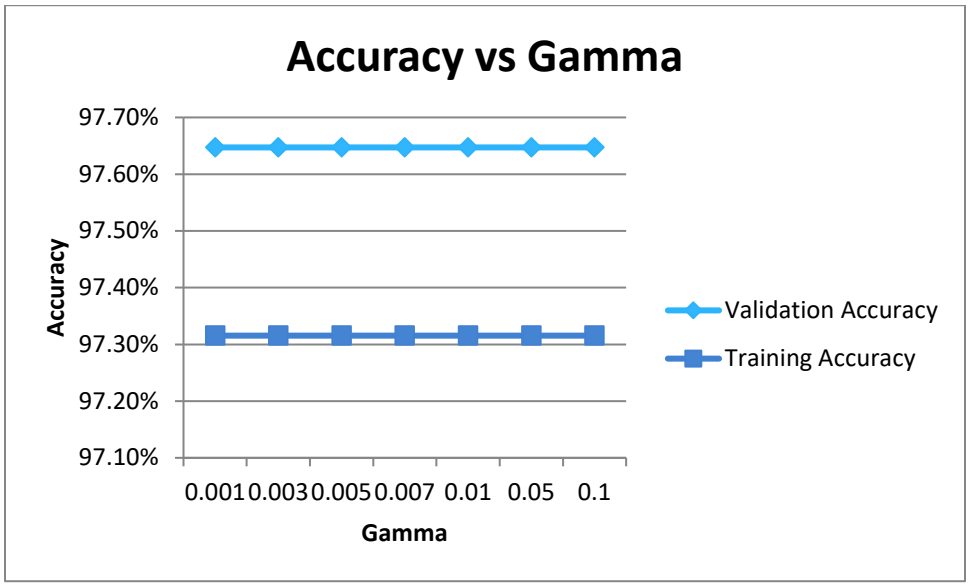
When we use Kernel_type = 3 i.e Sigmoid Function then it changes because we know that the Sigmoid Kernel function depends on the Gamma and we can easily able to see that In Sigmoid Kernel the error training and validation is approx similar for every value of gamma and the error is increasing for increasing the value of gamma => the error is low for gamma<=0.1 so our curve is at Good fitting. The error is high for gamma>0.01 so our curve is underfitting.

Gamma = 0.1, Class_Used = (4,5)



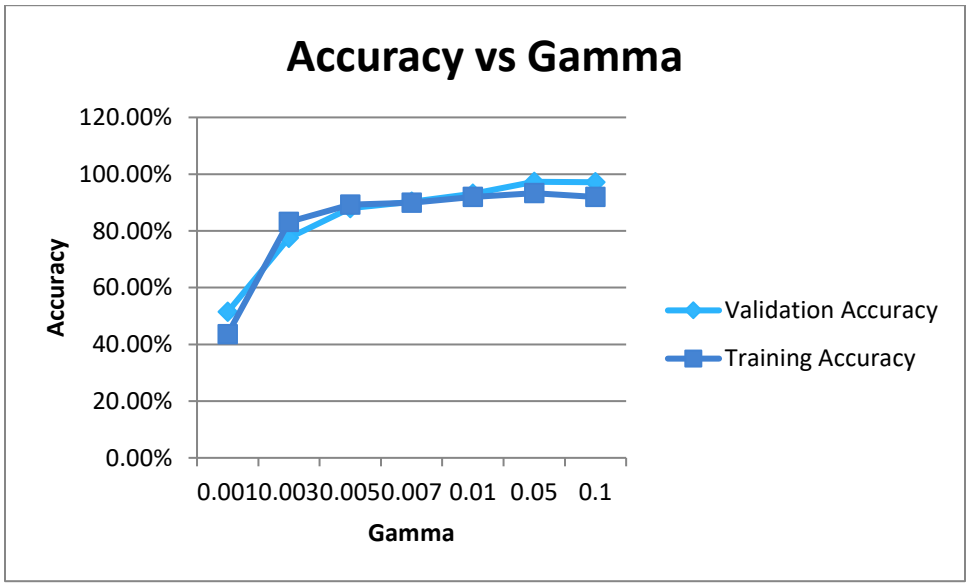
From the above curve, the gamma is fixed for all the kernels i.e 0.01 for this gamma value. Both validation and training error is approximately same except the sigmoid kernel all have the low error => all at the Good fitting while in the sigmoid kernel the error is high => our model is at Underfitting.

Kernel_type = 0 ,Class_Used = (0,9)



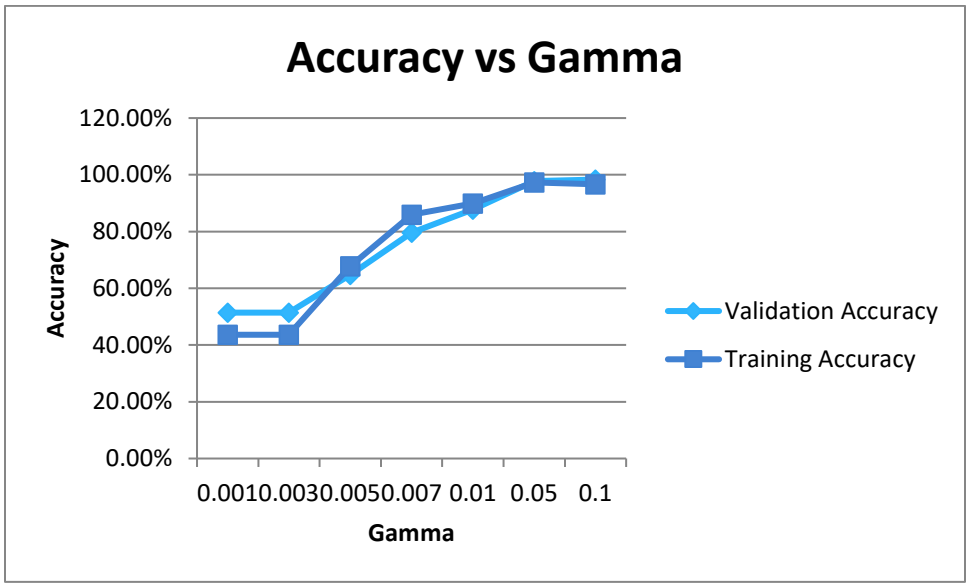
When we use Kernel_type = 0 i.e Linear function then with changing gamma doesn't change because we know that the Linear Kernel function does not depend on the Gamma and we can easily able to see that In Linear Kernel the error training is less than the validation error is also low so this is an example of Good fitting.

Kernel_type = 1 ,Class_Used = (0,9),Degree = 2



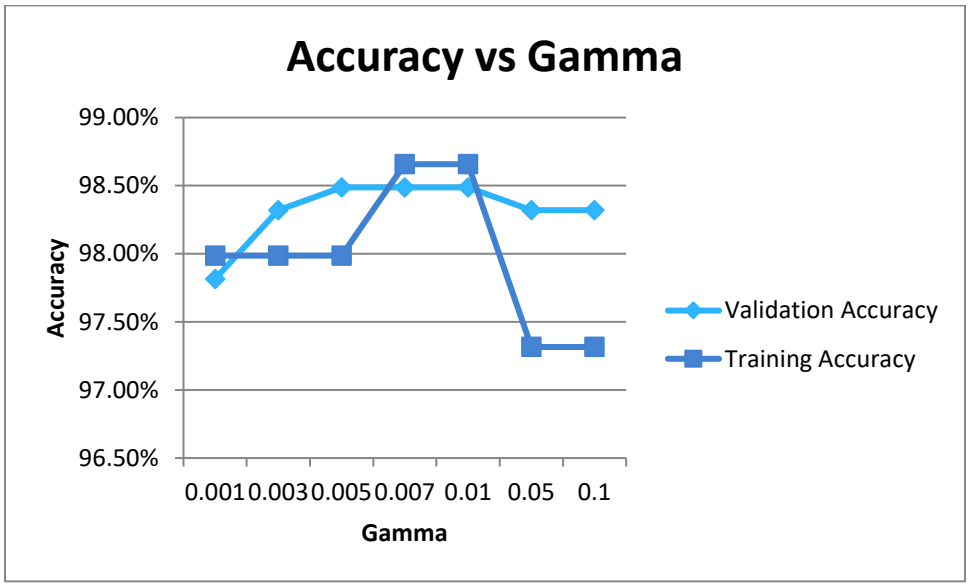
When we use Kernel_type = 1 i.e Polynomial function with degree = 2 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.001 so out curve is at underfitting of the curve and the error is less for Gamma>=0.001 so our curve is at Good Fitting.

Kernel_type = 1 ,Class_Used = (0,9),Degree = 3



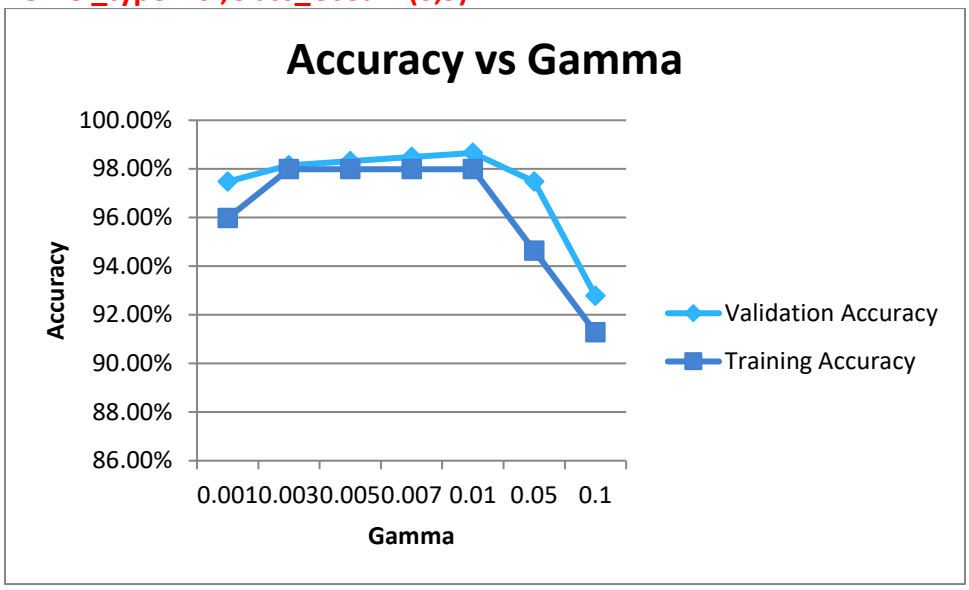
When we use Kernel_type = 1 i.e Polynomial function with degree = 3 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.007 so out curve is at underfitting of the curve and the error is less for Gamma>=0.007 so our curve is at Good Fitting.

Kernel_type = 2 ,Class_Used = (0,9)



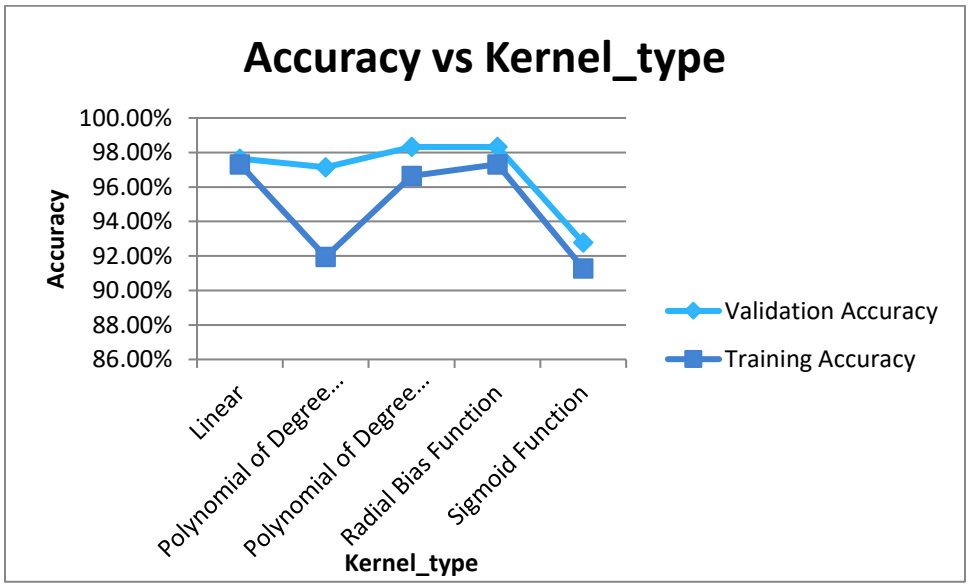
When we use Kernel_type = 2 i.e Radial Base Function then it changes because we know that the Radial Base Kernel function depends on the Gamma and we can easily able to see that In Radial Base Kernel the error training is less in compare validation error but still we can say that the curve is at the Good fitting stage for gamma<=0.1 but after that, we can easily see that the validation error is more wherein training it is low =>so our curve is at Overfitting.

Kernel_type = 3 ,Class_Used = (0,9)



When we use Kernel_type = 3 i.e Sigmoid Function then it changes because we know that the Sigmoid Kernel function depends on the Gamma and we can easily able to see that In Sigmoid Kernel the error training and validation is approx similar for every value of gamma and the error is increasing for increasing the value of gamma => the error is low for gamma<=0.1 so our curve is at Good fitting. The error is high for gamma>0.1 so our curve is underfitting.

Gamma = 0.1, Class_Used = (0,9)

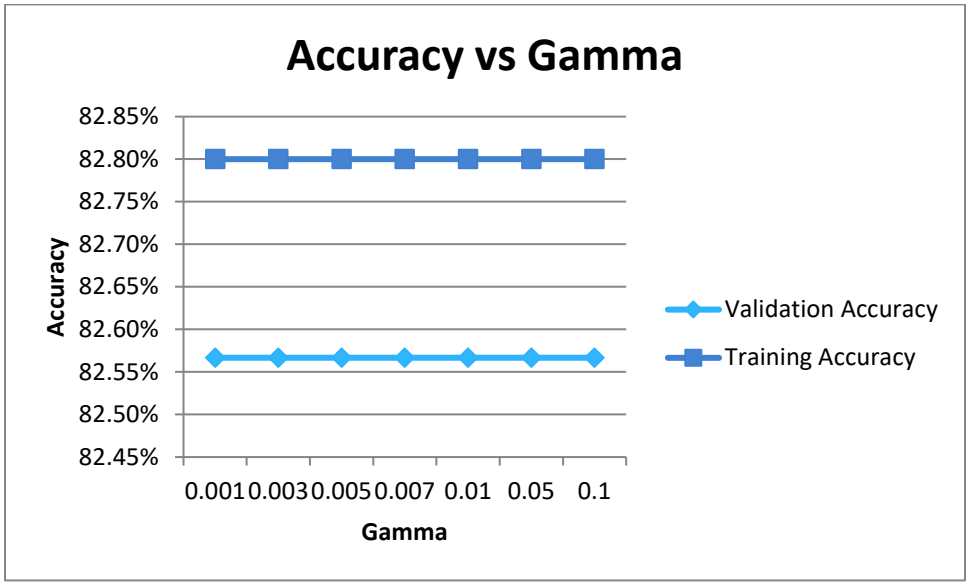


From the above curve, the gamma is fixed for all the kernels i.e 0.01 for this gamma value. Both validation and training error is approximately same except the sigmoid kernel all have the low error => all at the Good fitting while in the sigmoid kernel the error is high => our model is at Underfitting.

Yes, The data is a little bit depending on the class type which we chose and also gives the best result for the same parameter in this case but maybe this is not giving the best result in every case because let's say you choose two classes whose classification is very easy then other it doesn't depend a lot on classes.

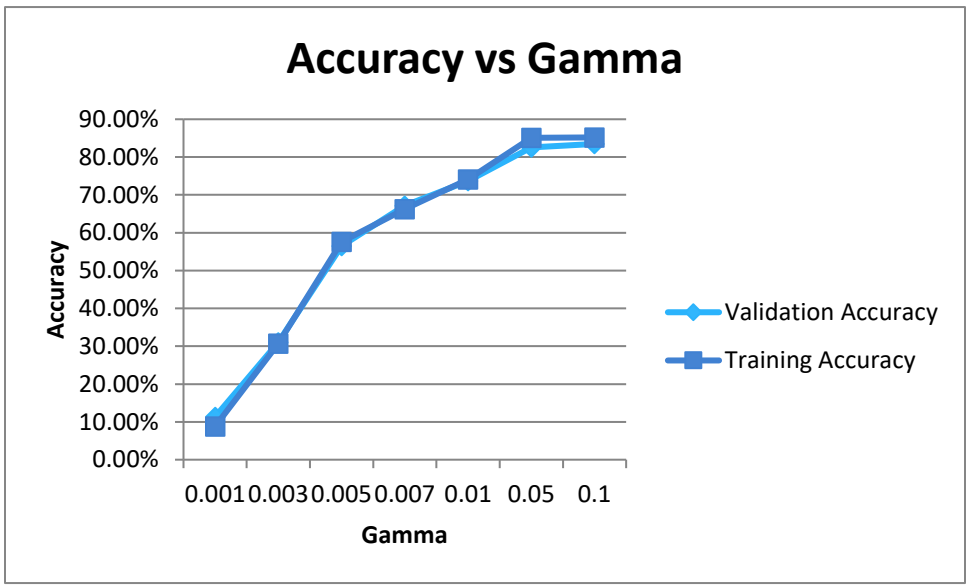
A.2) MULTICLASS :

Kernel_type = 0, 10 FEATURES



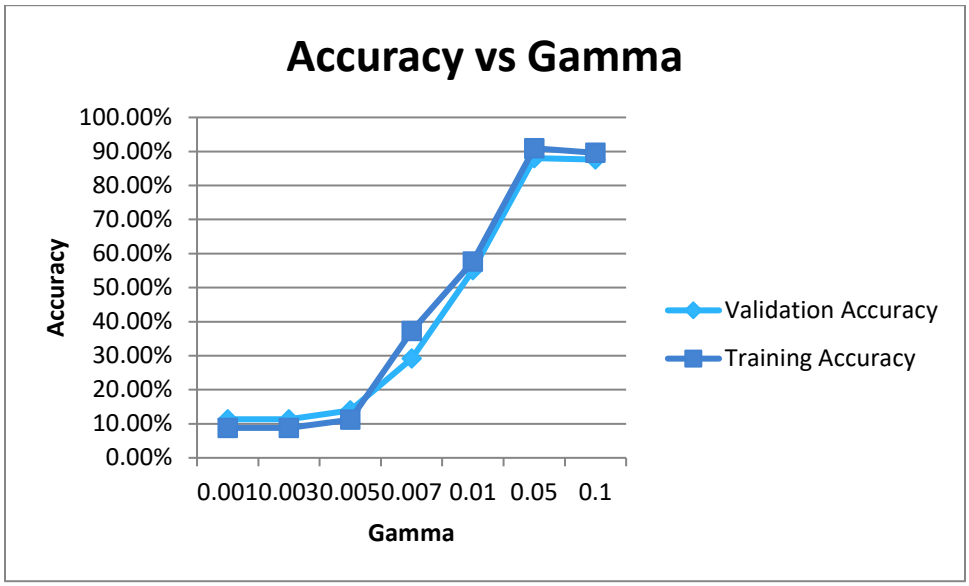
When we use Kernel_type = 0 i.e Linear function than with changing gamma doesn't change because we know that the Linear Kernel function does not depend on the Gamma and we can easily able to see that In Linear Kernel the error training is less than the validation error is also low so this is an example of Good fitting (as we take all the 10 classes).

Kernel_type = 1,Degree = 2, 10 FEATURES



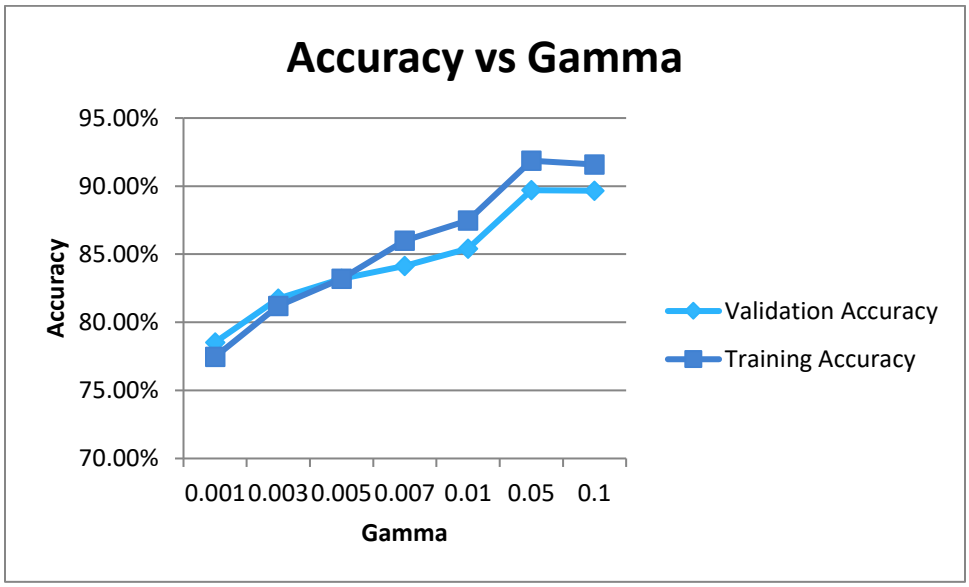
When we use Kernel_type = 1 i.e Polynomial function with degree = 2 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.01 so out curve is at underfitting of the curve and the error is less for Gamma>=0.01 so our curve is at Good Fitting.

Kernel_type = 1,Degree = 3, 10 FEATURES



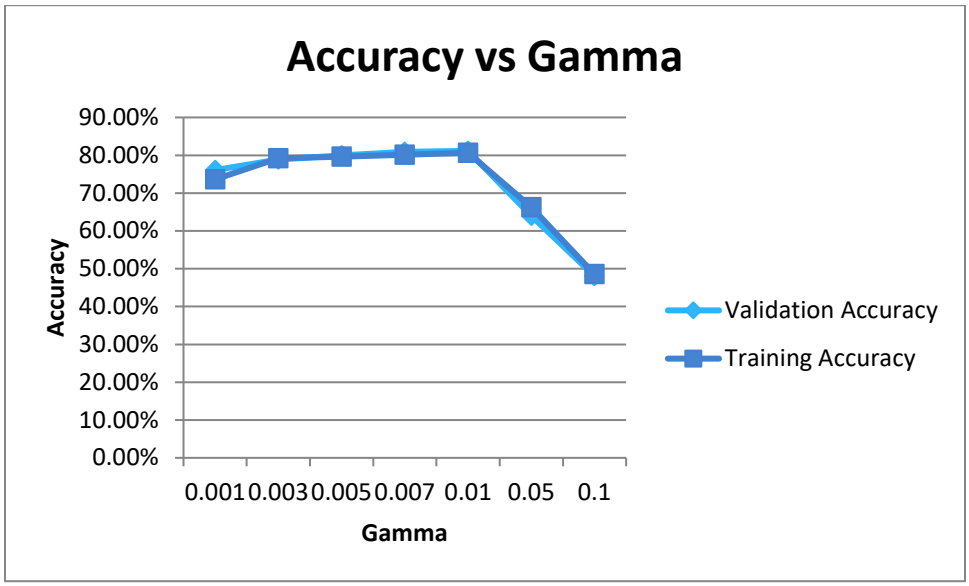
When we use Kernel_type = 1 i.e Polynomial function with degree = 3 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.05 so out curve is at underfitting of the curve and the error is less for Gamma>=0.05 so our curve is at Good Fitting.

Kernel_type = 2, 10 FEATURES



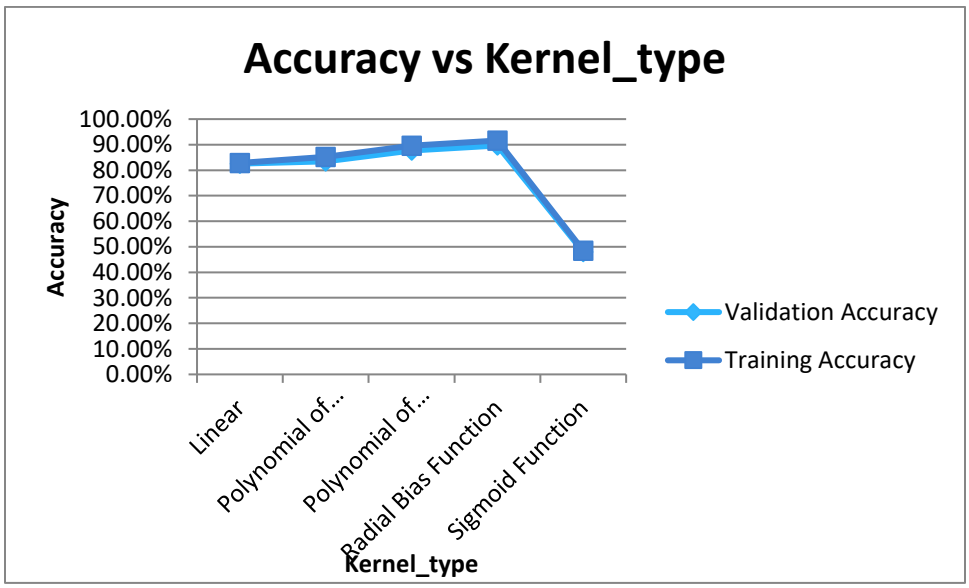
When we use Kernel_type = 2 i.e Radial Base Function then it changes because we know that the Radial Base Kernel function depends on the Gamma and we can easily able to see that In Radial Base Kernel the error training is less in compare validation error but still we can say that the curve is at the Good fitting stage for gamma>=0.001 but we can easily see for gamma <0.001 error is increasing for both train and validation =>so our curve is at Underfitting.

Kernel_type = 3, 10 FEATURES



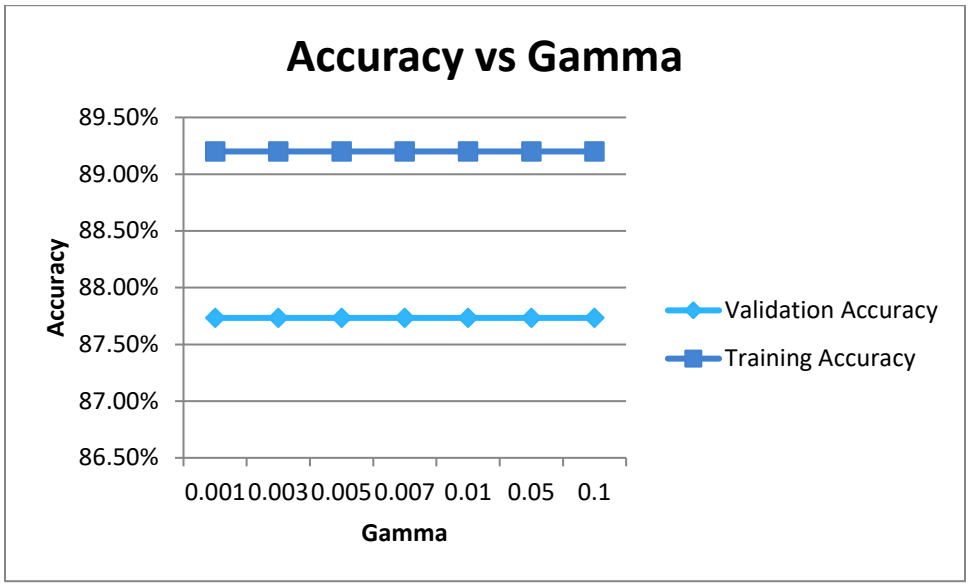
When we use Kernel_type = 3 i.e Sigmoid Function then it changes because we know that the Sigmoid Kernel function depends on the Gamma and we can easily able to see that In Sigmoid Kernel the error training and validation is approx similar for every value of gamma and the error is increasing for increasing the value of gamma => the error is low for 0.001<= gamma<=0.01 so our curve is at Good fitting. The error is high for gamma>0.01 so our curve is underfitting. The error is high for gamma<0.001 so our curve is underfitting.

Gamma = 0.1, 10 FEATURES



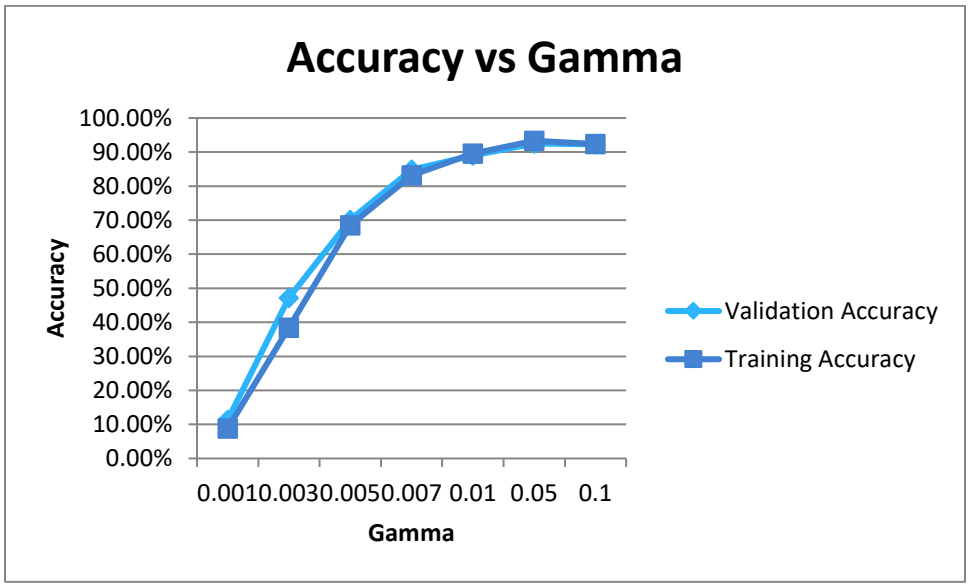
From the above curve, the gamma is fixed for all the kernels i.e 0.01 for this gamma value. Both validation and training error is approximately same except the sigmoid kernel all have the low error => all at the Good fitting while in the sigmoid kernel the error is high => our model is at Underfitting.

Kernel_type = 0, 25 FEATURES



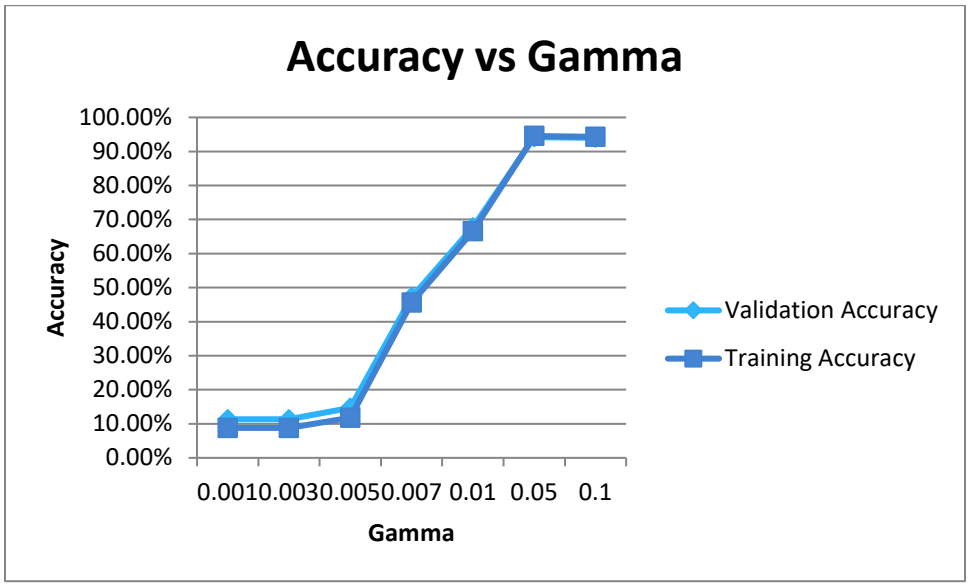
When we use Kernel_type = 0 i.e Linear function than with changing gamma doesn't change because we know that the Linear Kernel function does not depend on the Gamma and we can easily able to see that In Linear Kernel the error training is less than the validation error is also low so this is an example of Good fitting (as we take all the 10 classes).

Kernel_type = 1,Degree = 2, 25 FEATURES



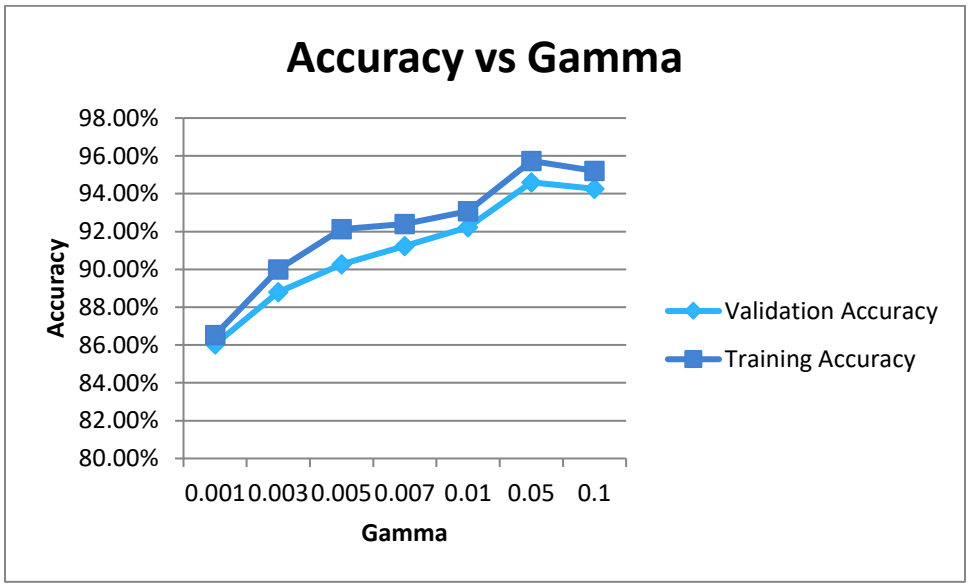
When we use Kernel_type = 1 i.e Polynomial function with degree = 2 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.007 so out curve is at underfitting of the curve and the error is less for Gamma>=0.007 so our curve is at Good Fitting.

Kernel_type = 1,Degree = 3, 25 FEATURES



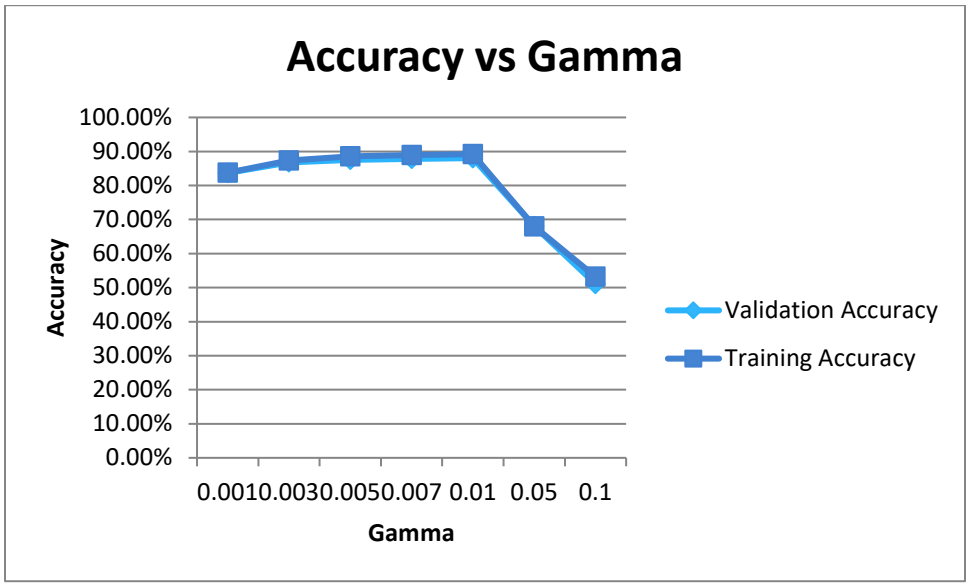
When we use Kernel_type = 1 i.e Polynomial function with degree = 3 then it changes because we know that the Polynomial Kernel function depends on the Gamma and we can easily able to see that In Polynomial Kernel the error training and validation is approx similar for every value of gamma and the error is high for Gamma <0.05 so out curve is at underfitting of the curve and the error is less for Gamma>=0.05 so our curve is at Good Fitting.

Kernel_type = 2, 25 FEATURES



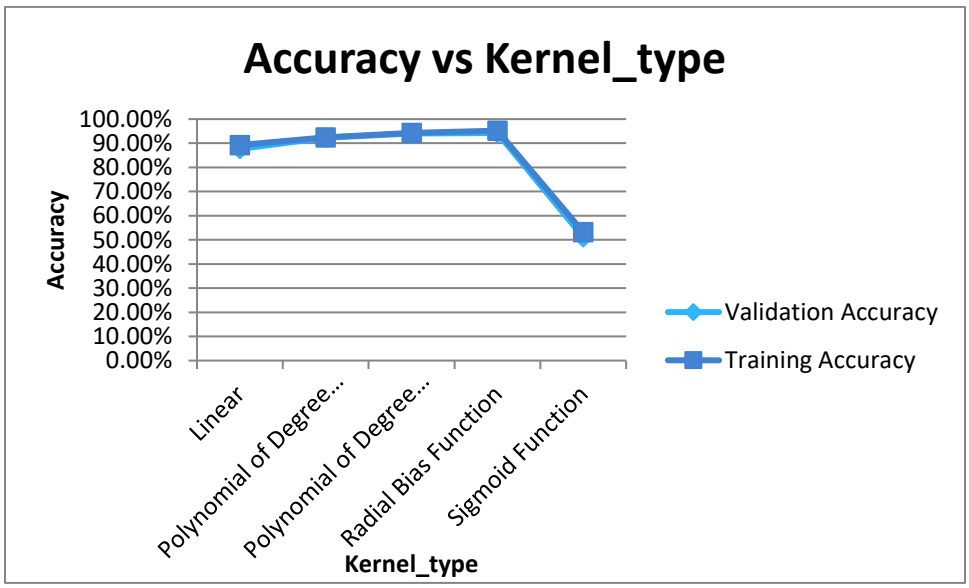
When we use Kernel_type = 2 i.e Radial Base Function then it changes because we know that the Radial Base Kernel function depends on the Gamma and we can easily able to see that In Radial Base Kernel the error training is less in compare validation error but still we can say that the curve is at the Good fitting stage for gamma>=0.001 but we can easily see for gamma <0.001 error is increasing for both train and validation =>so our curve is at Underfitting.

Kernel_type = 3, 25 FEATURES



When we use Kernel_type = 3 i.e Sigmoid Function then it changes because we know that the Sigmoid Kernel function depends on the Gamma and we can easily able to see that In Sigmoid Kernel the error training and validation is approx similar for every value of gamma and the error is increasing for increasing the value of gamma => the error is low for gamma<=0.01 so our curve is at Good fitting. The error is high for gamma>0.01 so our curve is underfitting.

Gamma = 0.1, 25 FEATURES



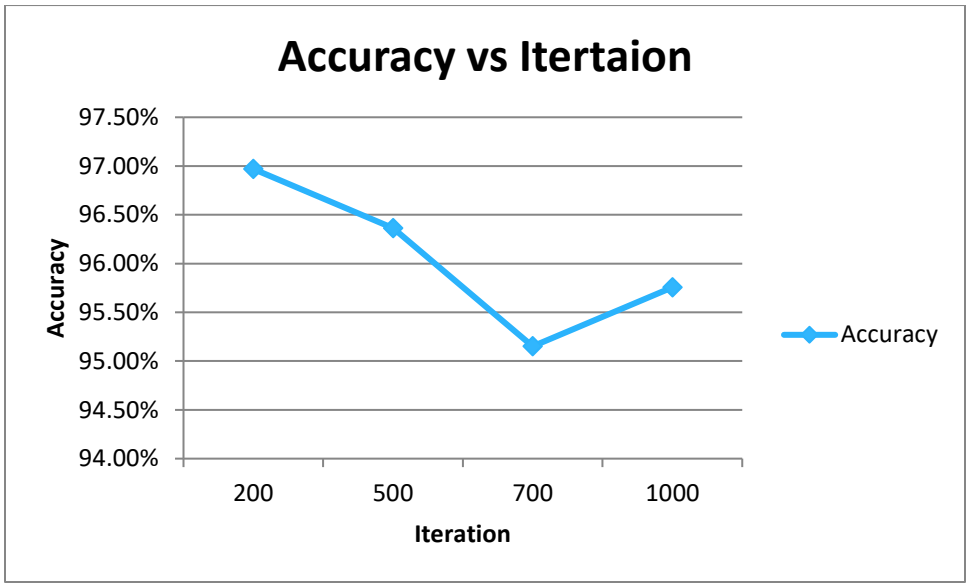
From the above curve, the gamma is fixed for all the kernels i.e 0.01 for this gamma value. Both validation and training error is approximately same except the sigmoid kernel all have the low error => all at the Good fitting while in the sigmoid kernel the error is high => our model is at Underfitting.

Here we see that when classification using 25 features results are better than the 10 features the usefulness of more features is that we classify our data more precisely by using more features. When we compare our data to

the Binary data then we saw a lot of change in the Data and a lot of Differences between the accuracy for training data and validation data also the reason this is maybe in Binary we use 2 classes so the classification of taken two problems may easy and the exceptional cases are less while if we add some more cases then the probability of classification error increases that the reason behind them that the accuracy is so much changing I multiclass in compare to binary class.

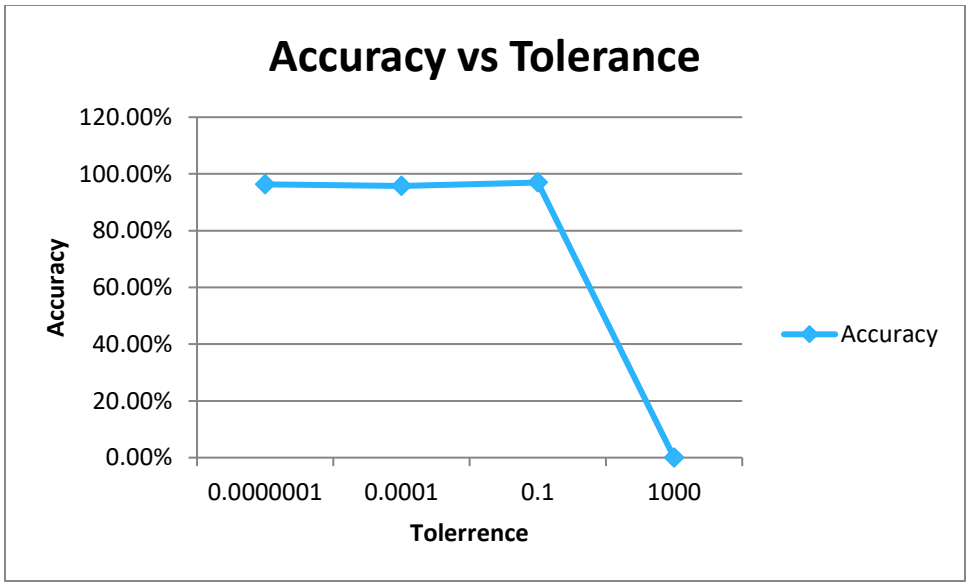
B.) SIMPLIFIED SMO :

Regularization(C) = 1 ,Class_Used = (1,2), Tolerance = 0.1



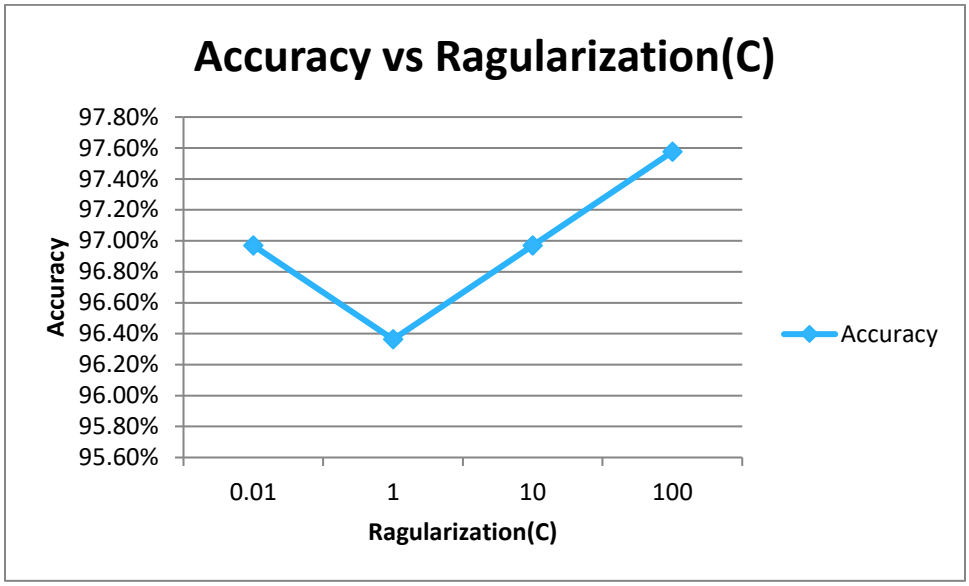
In this case we calculate the graph between the number of iterations performed on a Simplified SMO and we see that we we increase iteration initially then the acuuracy is decrases initially but after a certain times it also increase and the accuracy is quite good than what we see in the case of SVM vector but while calculating this I takes much time then SMO.

Regularization(C) = 1 ,Class_Used = (1,2), Iterations = 200



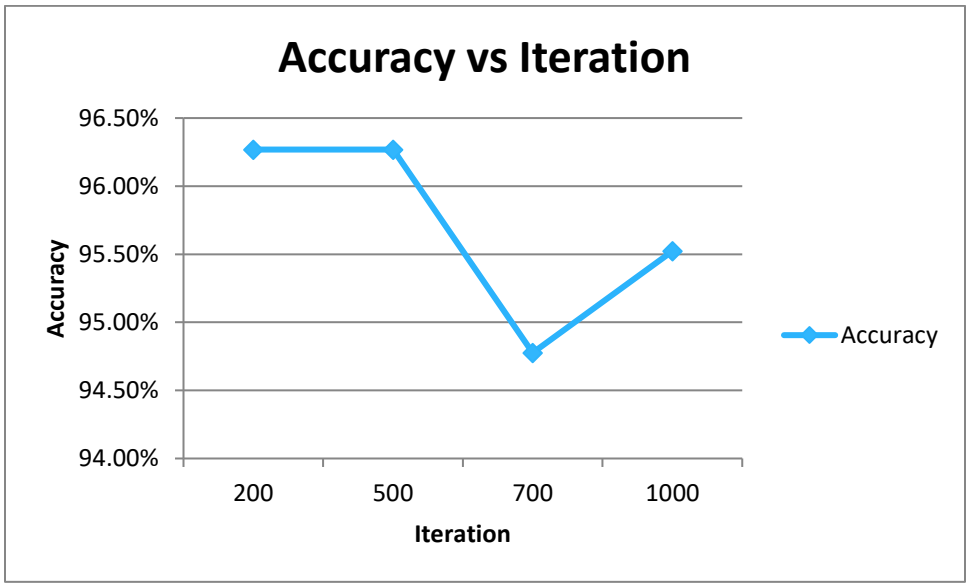
In this case we calculate the graph between the Tolerance change on a Simplified SMO and we see that when the tolerance is greater than 0 then the accuracy is 0 and when tolerance is less than 0 then we can easily see from the curve that the accuracy is somehow constant and also we can see ghat the accuracy is good for all the caes other than the tolerance is greater than 0 that means the SMO gives better accuracy than the SVM but it will takes some time more than SVM for evaluating the data.

Iterations = 200 ,Class_Used = (1,2), Tolerance = 0.1



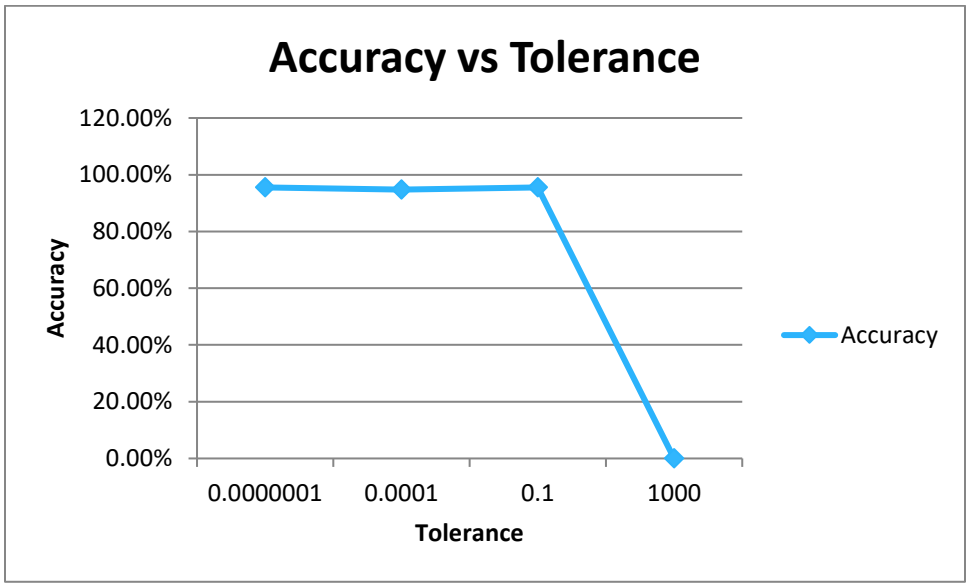
In this case we calculate the graph between the Regularization change on a Simplified SMO and we see that when the to Regularization is less than 0.01 the accuracy is increasing and the regularization is >1 also the regulasrization is increasing but between then them it is increasing but also we can see from the curve the change in accuracy is not much it is <1% that means that the accuracy is quite good enough when we compare this to SVM model but also the same problem that it will take more time.

Regularization(C) = 1 ,Class_Used = (4,5), Tolerance = 0.1



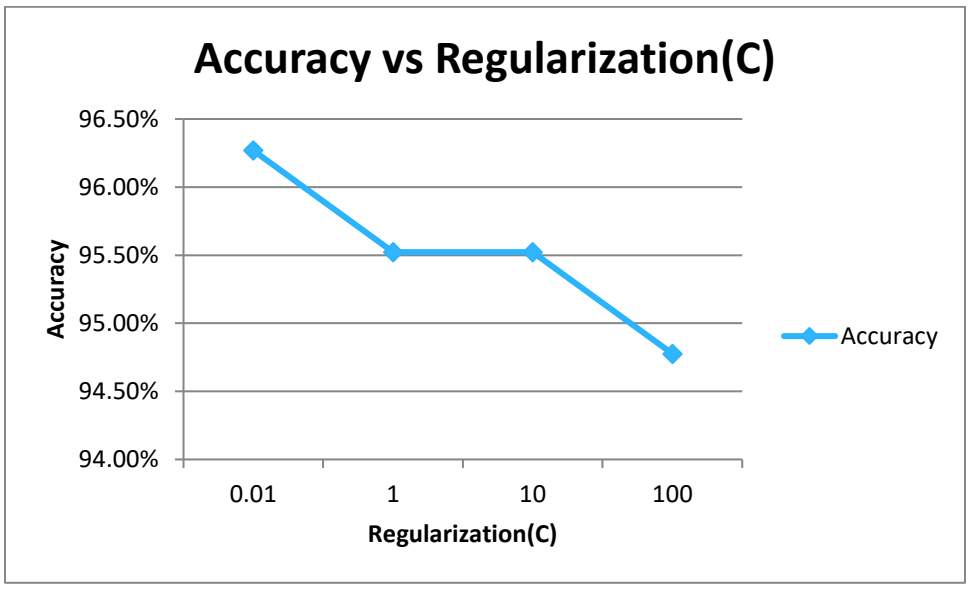
Now see the cae for the class (4,5) In this case we calculate the graph between the number of iterations performed on a Simplified SMO and we see that we we increase iteration initially then the acuuracy is decrases initially but after a certain times it also increase and the accuracy is quite good than what we see in the case of SVM vector but while calculating this I takes much time then SMO.

Regularization(C) = 1 ,Class_Used = (4,5), Iterations = 200



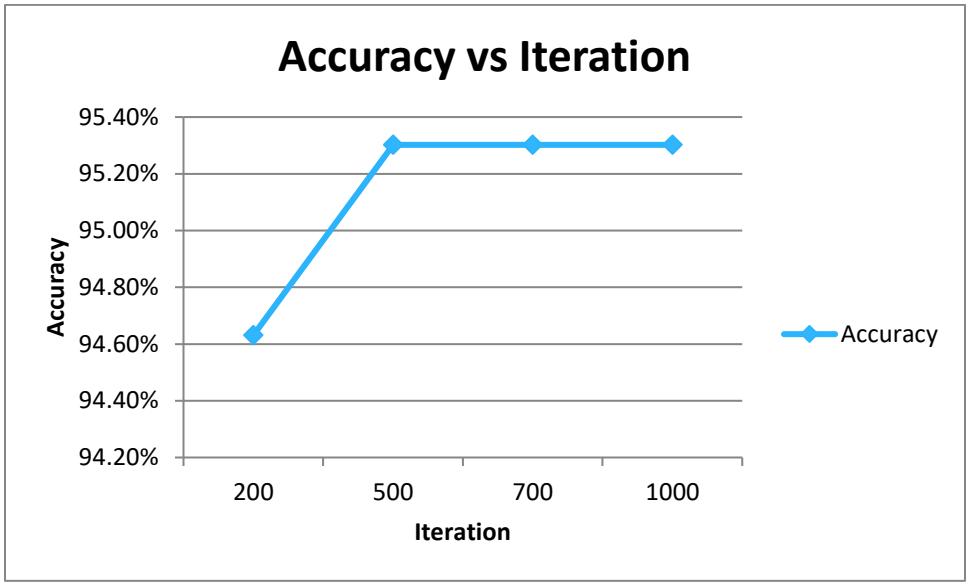
Now see the cae for the class (4,5) In this case we calculate the graph between the Tolerance change on a Simplified SMO and we see that when the tolerance is greater than 0 then the accuracy is 0 and when tolerance is less than 0 then we can easily see from the curve that the accuracy is somehow constant and also we can see ghat the accuracy is good for all the caes other than the tolerance is greater than 0 that means the SMO gives better accuracy than the SVM but it will takes some time more than SVM for evaluating the data.

Iterations = 200 ,Class_Used = (4,5), Tolerance = 0.1



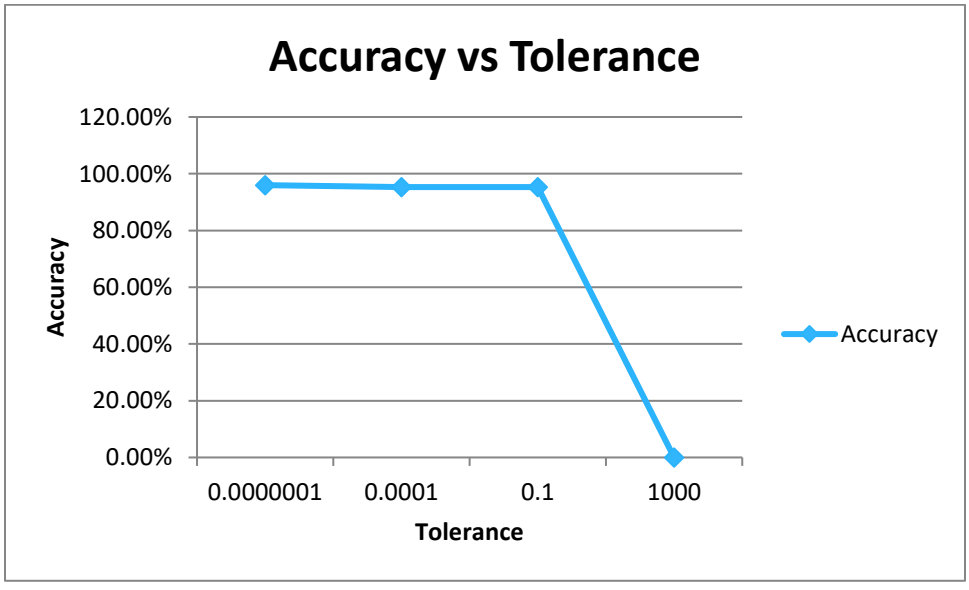
Now see the cae for the class (4,5) In this case we calculate the graph between the Regularization change on a Simplified SMO and we see that when the to Regularization is less than 0.01 the accuracy is increasing and the regularization is >1 also the regulasrization is increasing but between then them it is increasing but also we can see from the curve the change in accuracy is not much it is <1% that means that the accuracy is quite good enough when we compare this to SVM model but also the same problem that it will take more time.

Regularization(C) = 1 ,Class_Used = (0,9), Tolerance = 0.1



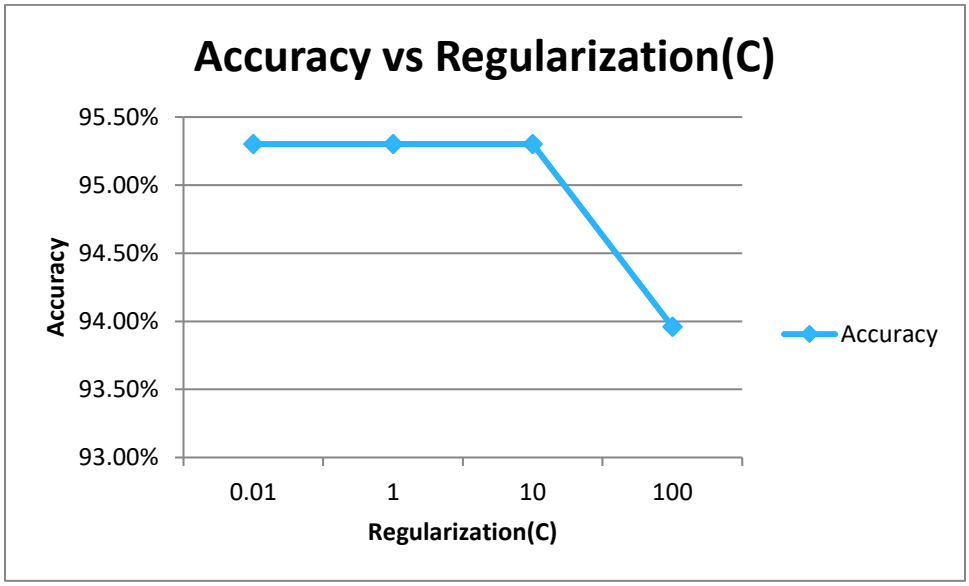
Now see the cae for the class (0,9)) In this case we calculate the graph between the number of iterations performed on a Simplified SMO and we see that we we increase iteration initially then the acuuracy is decrases initially but after a certain times it also increase and the accuracy is quite good than what we see in the case of SVM vector but while calculating this I takes much time then SMO.

Regularization(C) = 1 ,Class_Used = (0,9), Iterations = 200



Now see the cae for the class (0,91) In this case we calculate the graph between the Tolerance change on a Simplified SMO and we see that when the tolerance is greater than 0 then the accuracy is 0 and when tolerance is less than 0 then we can easily see from the curve that the accuracy is somehow constant and also we can see ghat the accuracy is good for all the caes other than the tolerance is greater than 0 that means the SMO gives better accuracy than the SVM but it will takes some time more than SVM for evaluating the data.

Iterations = 200 ,Class_Used = (0,9), Tolerance = 0.1



Now see the cae for the class (0,9) In this case we calculate the graph between the Regularization change on a Simplified SMO and we see that when the to Regularization is less than 0.01 the accuracy is increasing and the regularization is >1 also the regulasrization is increasing but between then them it is increasing but also we can see from the curve the change in accuracy is not much it is <1% that means that the accuracy is quite good enough when we compare this to SVM model but also the same problem that it will take more time.

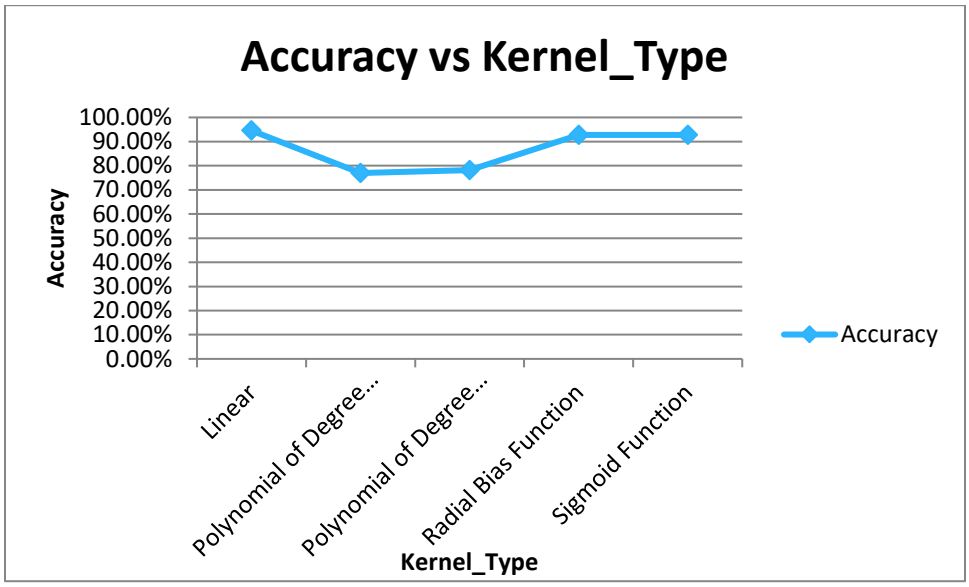
Here we can easily see that in each class case compare to the binary class the Acuuracy we obtain is better than the Accuracy we calculated using the SVM model this means that the SMO predict the data in a better way then the SVM Model but the main problem in this is that the SMO take more time than the SVM model so if we try our mdel for very large data then may be SMO take years to predict the data where as in case of SVM we can see that the Acuuracy difference is not much less than SMO but SVM is fast so that we predict our data in less time.

C.) FULL SMO :

Here we design a FULL SMO so the difference between the FULL and Simplified SMO is the time difference i.e both work on the same concept but in FULL SMO we optimized our Simplified SMO so that it can run fast so that we can process on big datas here we analysis on three classes that we use of Simplified SMO.

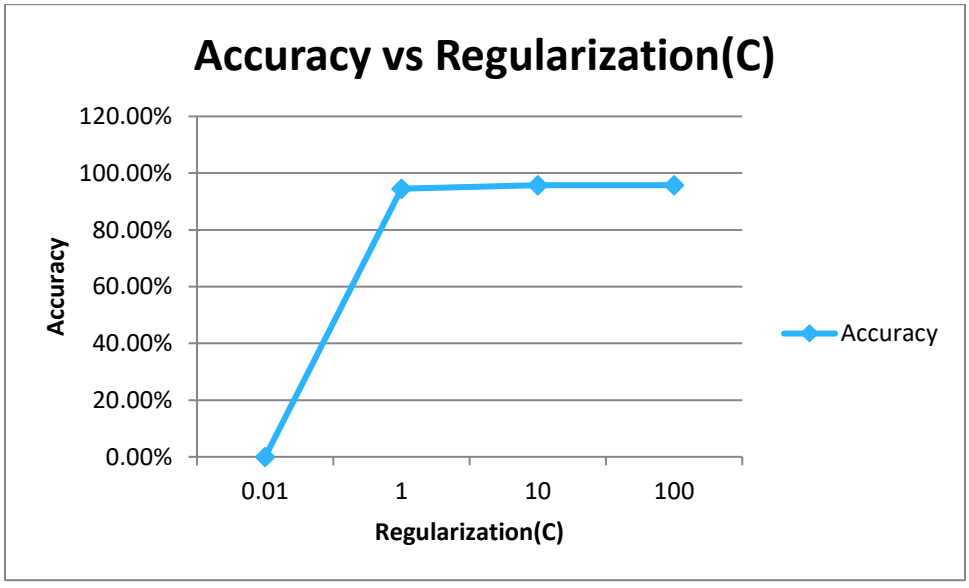
Refrences used for making this code is : Microsoft Word - smoTR.doc (iitd.ac.in)

Regularization(C) = 1 ,Class_Used = (1,2), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



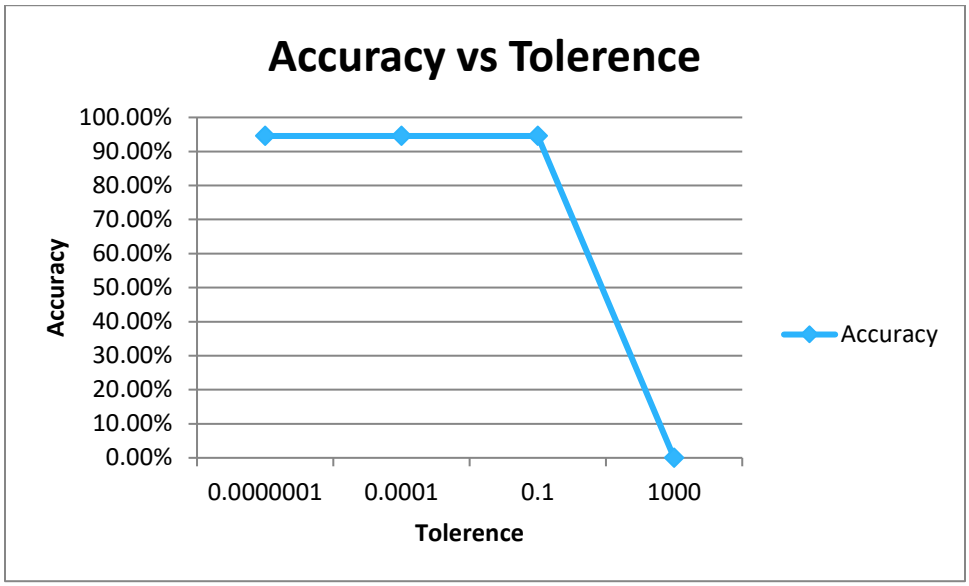
As we see from the curve that the accuracy for Linear Radial Bais Function and Sigmoid Function is quite well where as in case of the polynomial kernel we can easily able to see that it is increasing accuracy while we increase the degree of polynomial but the Overall accuracy is wuite very well when we compare this to the Simplified SMO the basic difference between them is the timing and speed the speed foe Full SMO is very fast in compare to the Simplified SMO.

Kernel_Type = 0 ,Class_Used = (1,2), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



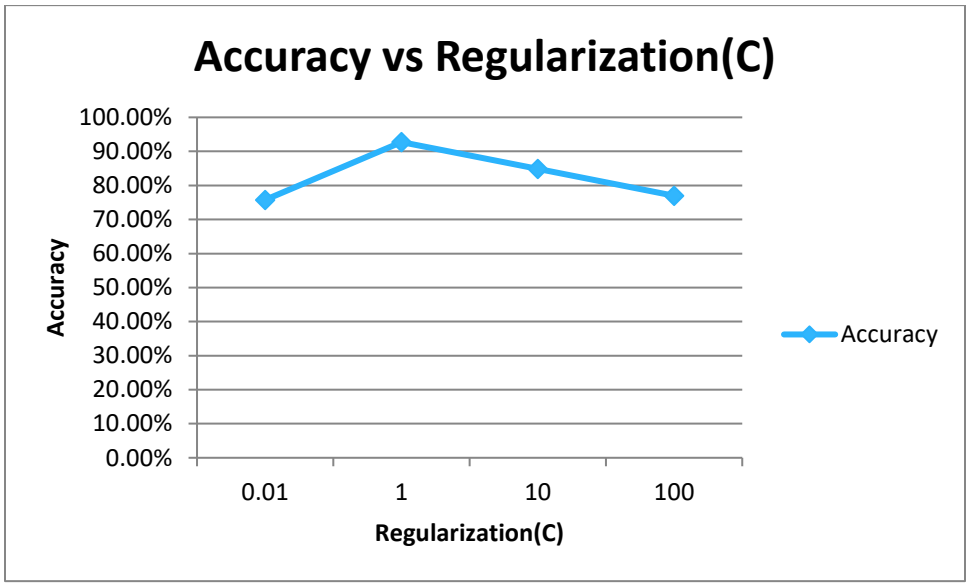
In this case we calculate the graph between the Regularization change on a FULL SMO and we see that when the to Regularization is less than 1 then Accuracy will be zero but when regularization>1 the accuracy is increasing a little bit.Similary the difference between the Simplified and full is the execution time it takes less execution time than simplified.

Kernel_Type = 0 ,Class_Used = (1,2), Regularization(C) = 1, Epsilon = 0.1, Gamma = 0.1



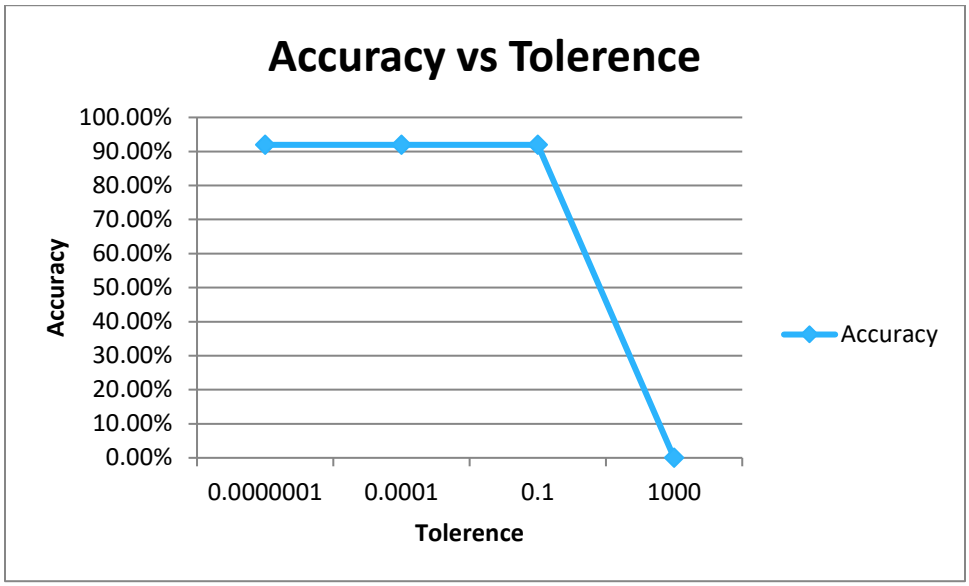
In this case we calculate the graph between the Tolerence change and the accuracy here we see that when tolerance is <1 then the Accuracy is similar in every case but when we increase tolerance to the 1 or greater than one then the accuracy falls to zero because due to KKT conditions.Again the difference between the Simplified and the full SMO is the execution time it takes less execution time.

Kernel_Type = 2 ,Class_Used = (1,2), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



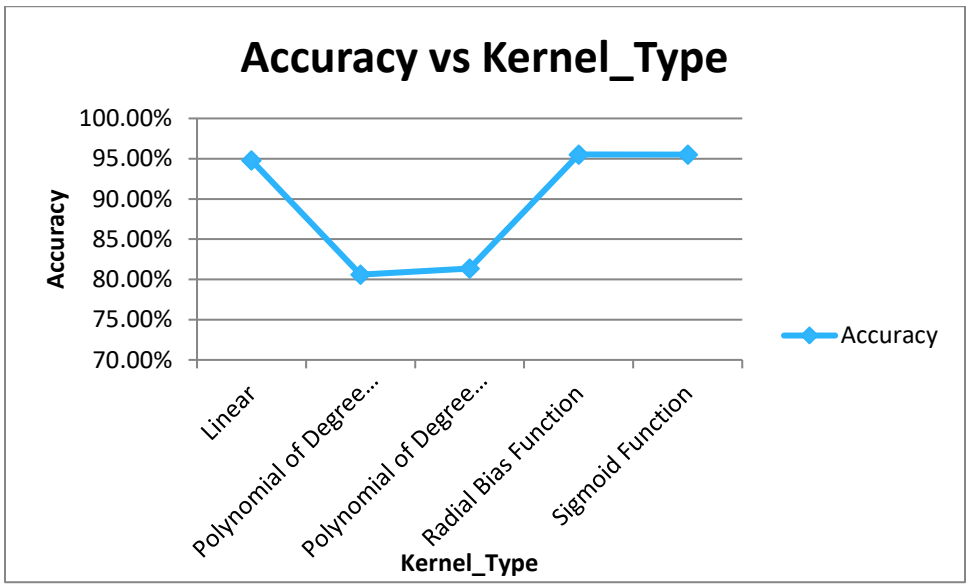
In this case we calculate the graph between the regularization and the Accuracy the kerel and the class and other para,ter mention above the curve thatw e use so here we can see that the Accuracy is increasing upto C = 1 after that the Accuracy is falling down.

Kernel_Type = 2 ,Class_Used = (1,2), Regularization(C) = 1, Epsilon = 0.1, Gamma = 0.1



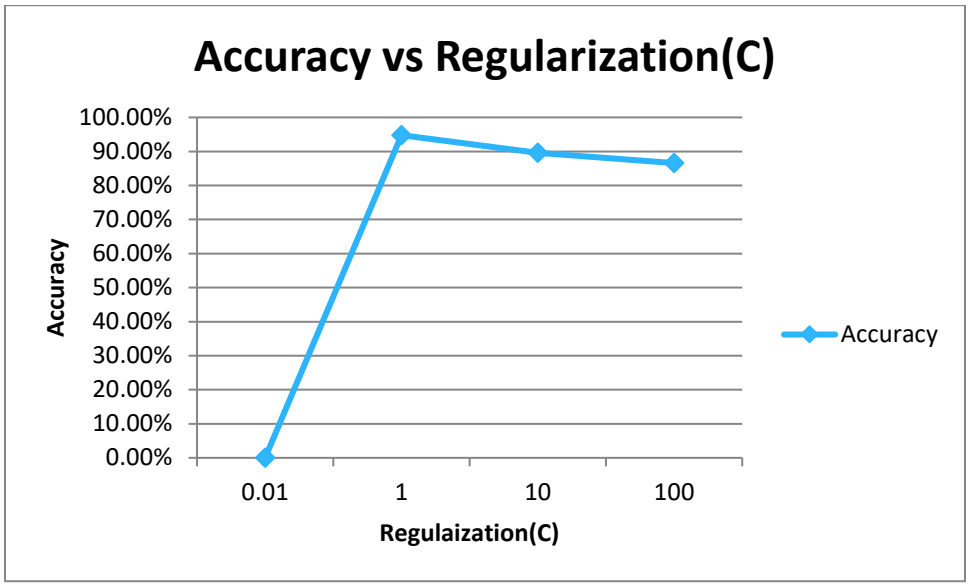
As we see in the above cases in Simplified as well as FULL SMO the Tolernce doedn't effect the accuracy until it is less than 1 when it is graeter than one than in that case the accuracy falls to zero due to KKT conditions.

Regularization(C) = 1 ,Class_Used = (4,5), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



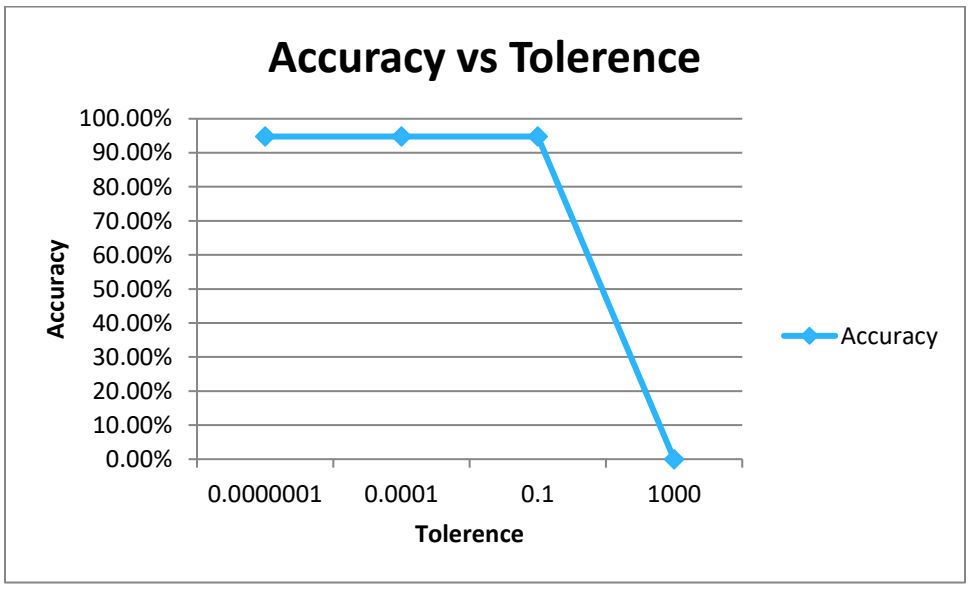
As we see from the curve that the accuracy for Linear Radial Bais Function and Sigmoid Function is quite well where as in case of the polynomial kernel we can easily able to see that it is increasing accuracy while we increase the degree of polynomial but the Overall accuracy is wuite very well when we compare this to the Simplified SMO the basic difference between them is the timing and speed the speed foe Full SMO is very fast in compare to the Simplified SMO.

Kernel_Type = 0 ,Class_Used = (4,5), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



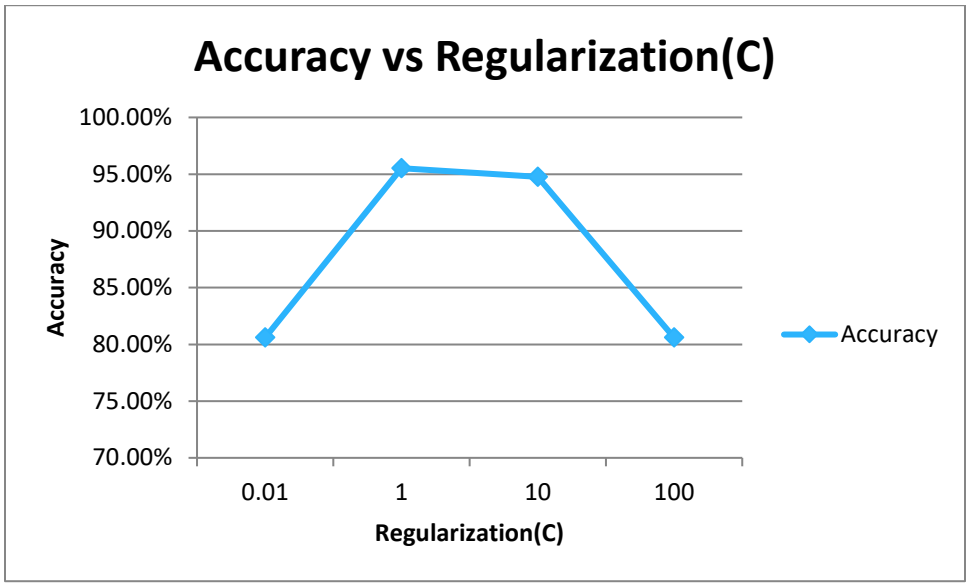
In this case we calculate the graph between the Regularization change on a FULL SMO and we see that when the to Regularization is less than 1 then Accuracy will be zero but when regularization>1 the accuracy is increasing a little bit.Similary the difference between the Simplified and full is the execution time it takes less execution time than simplified.

Kernel_Type = 0 ,Class_Used = (4,5), Regularization(C) = 1, Epsilon = 0.1, Gamma = 0.1



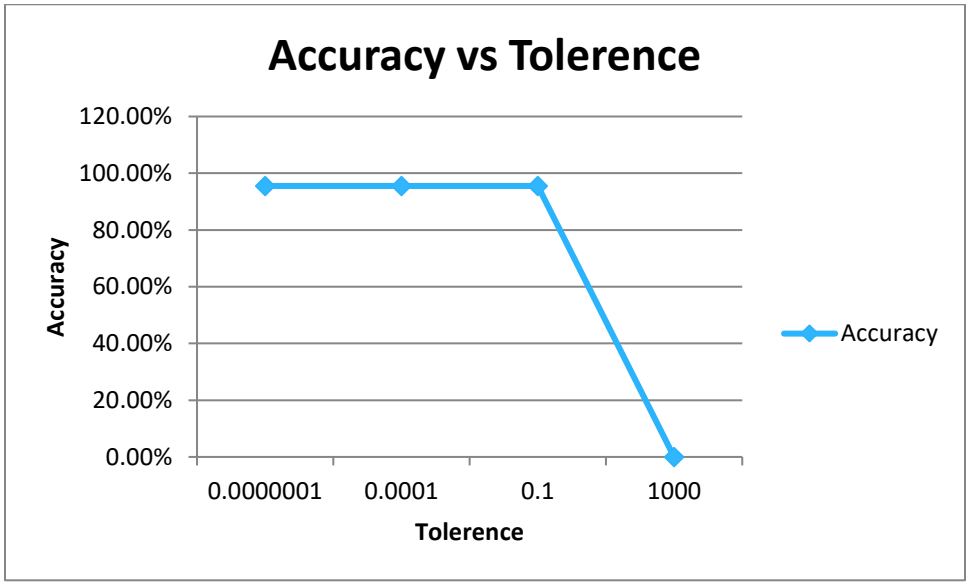
As we see in the above cases in Simplified as well as FULL SMO the Tolernce doedn't effect the accuracy until it is less than 1 when it is graeter than one than in that case the accuracy falls to zero due to KKT conditions.

Kernel_Type = 2 ,Class_Used = (4,5), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



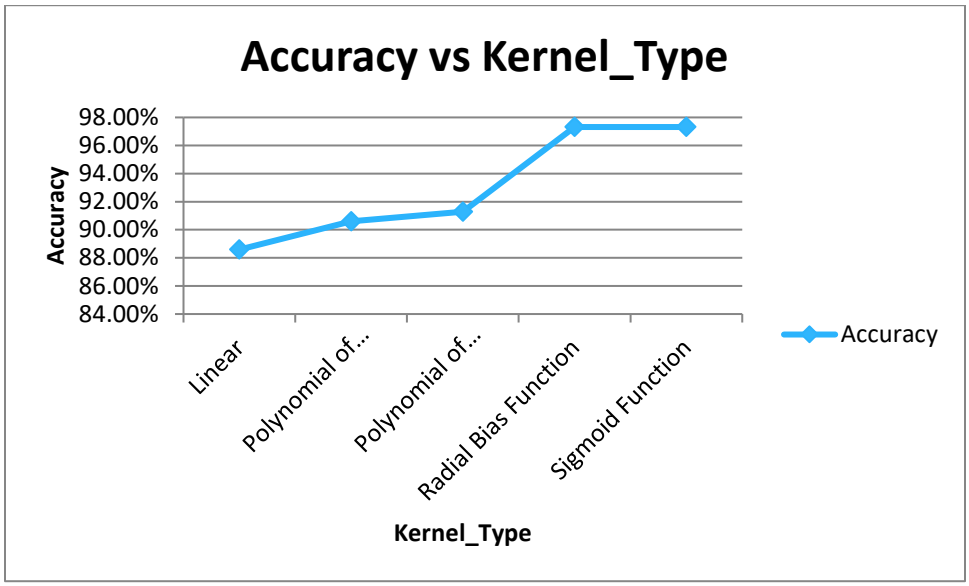
In this case we calculate the graph between the regularization and the Accuracy the kerel and the class and other para,ter mention above the curve thatw e use so here we can see that the Accuracy is increasing upto C = 1 after that the Accuracy is falling down.

Kernel_Type = 2 ,Class_Used = (4,5), Regularization(C) = 1, Epsilon = 0.1, Gamma = 0.1



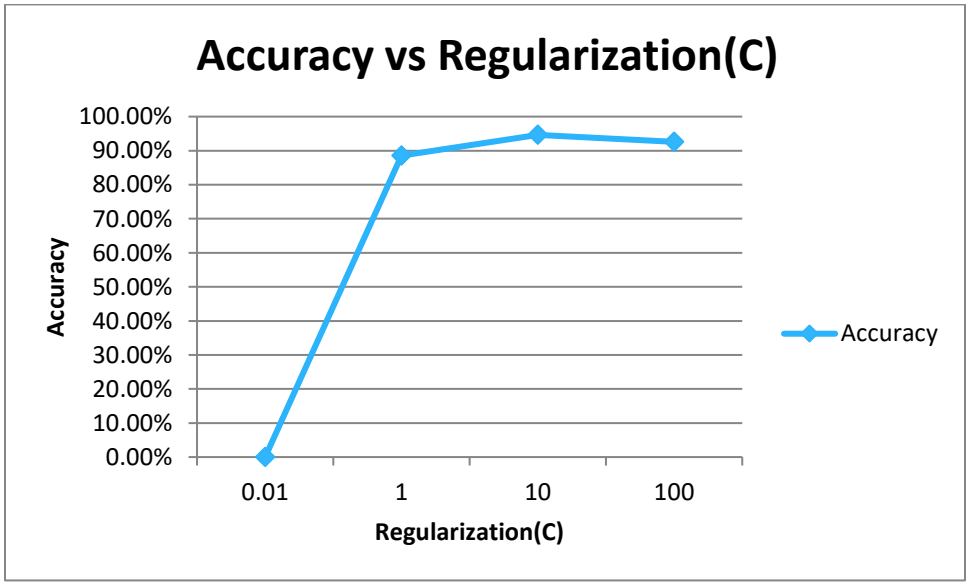
As we see in the above cases in Simplified as well as FULL SMO the Tolernce doedn't effect the accuracy until it is less than 1 when it is graeter than one than in that case the accuracy falls to zero due to KKT conditions.

Regularization(C) = 1 ,Class_Used = (0,9), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



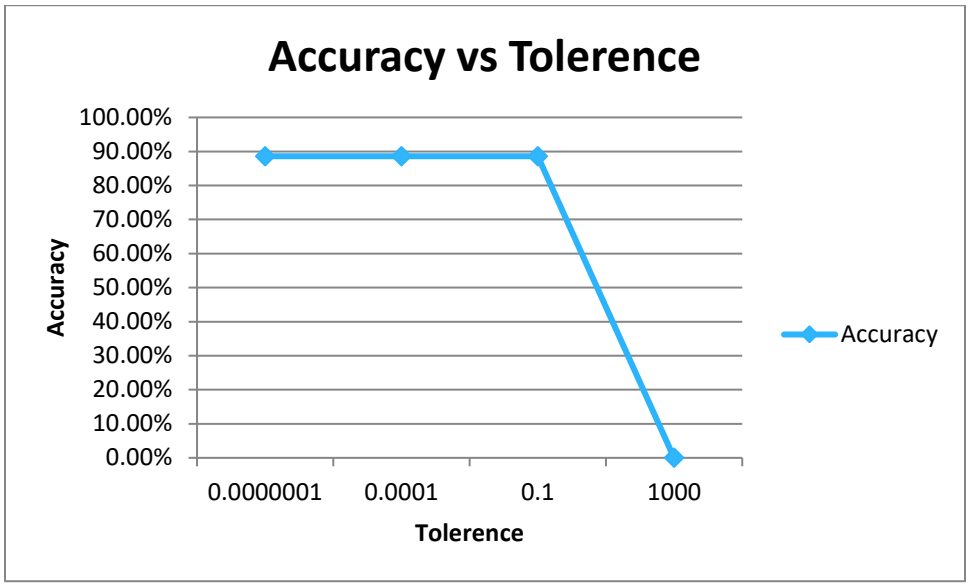
As we see from the curve that the accuracy for Radial Bais Function and Sigmoid Function is quite well where as in case of the polynomial kernel we can easily able to see that it is increasing accuracy while we increase the degree of polynomial but the Overall accuracy is wuite very well when we compare this to the Simplified SMO the basic difference between them is the timing and speed the speed foe Full SMO is very fast in compare to the Simplified SMO.

Kernel_Type = 0 ,Class_Used = (0,9), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



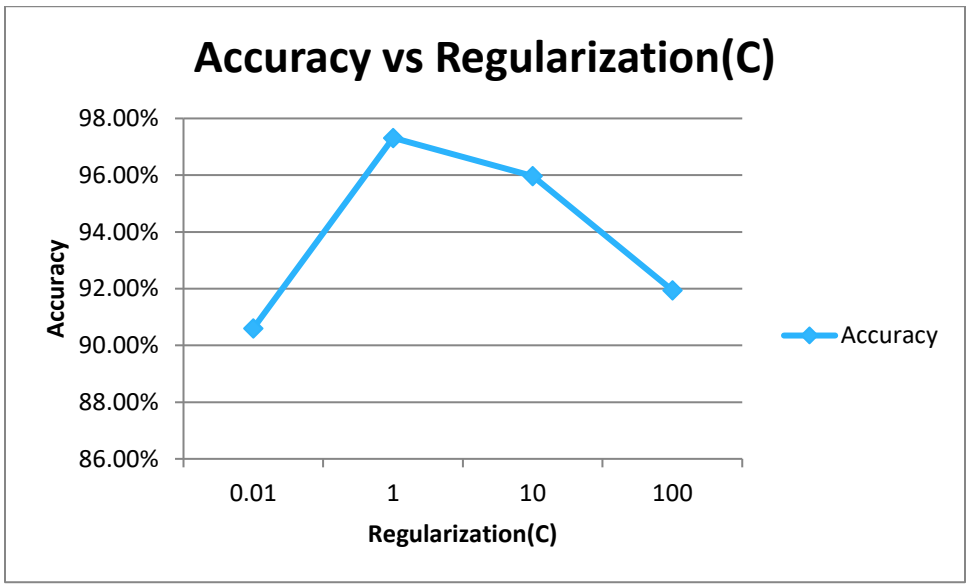
In this case we calculate the graph between the regularization and the Accuracy the kerel and the class and other para,ter mention above the curve thatw e use so here we can see that the Accuracy is increasing upto C = 10 after that the Accuracy is falling down.

Kernel_Type = 0 ,Class_Used = (4,5), Regularization(C) = 1, Epsilon = 0.1, Gamma = 0.1



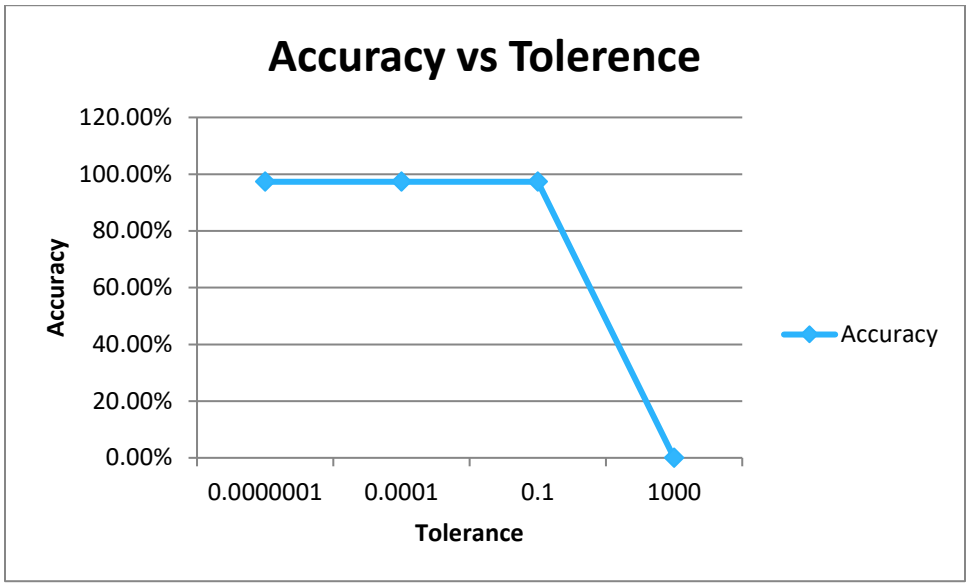
As we see in the above cases in Simplified as well as FULL SMO the Tolernce doedn't effect the accuracy until it is less than 1 when it is graeter than one than in that case the accuracy falls to zero due to KKT conditions.

Kernel_Type = 2 ,Class_Used = (0,9), Tolerance = 0.1, Epsilon = 0.1, Gamma = 0.1



In this case we calculate the graph between the regularization and the Accuracy the kerel and the class and other para,ter mention above the curve thatw e use so here we can see that the Accuracy is increasing upto C = 1 after that the Accuracy is falling down.

Kernel_Type = 2 ,Class_Used = (4,5), Regularization(C) = 1, Epsilon = 0.1, Gamma = 0.1



As we see in the above cases in Simplified as well as FULL SMO the Tolernce doedn't effect the accuracy until it is less than 1 when it is graeter than one than in that case the accuracy falls to zero due to KKT conditions.

Here we see the data for the three classes that we used for our BINRAYCLASS and Also for the Simplified SMO and here we used some parameters like Gamma,Regurization(C),Tolerence, Epsilon,Kernerl_type basically we here analysis for only two kernerl_types i.e Linera and Radial bais Function and the Accuracy is look like preety well when we compare our data with the Simplified SMO case data predicted for the same parameters then we can see thaht the accuracy is quite similar for each and every case. But the main differnec we strike here is that we opotimized our Simplified SMO to full SMO so that as we see in the last section form the Comparison of Simplified SMO and BINARYCLASS case we see that the SMO is very Slow in compare to the SVM models so then we optimzized our SMO so that it works faster then the work earlier andat the time of execution we can easily see the difference for execution time it takes is very very less in compare to hat it takes earlier case.All the above cases are for 25 features but we can excute the programme and saw that when we use less features the accuracy we get at that is less in compare to the more features case.so here we can easily say that more number of features is good for better classification of the data.Here we use the approach for train the lagrainge multipliers is previous Simplied SMO approach along with use some objective function for tarin the Langrange Multipliera and also use Hueriestics for trining that the Hueristic is that we take the If E1 is positive then SMO chooses an example with minimum error E2. If E1 is negative then SMO chooses maximum error E2 the Error calculated by the SMO output minus the original output.

PART 2.) SVM Kaggle Submission :

So for this part of Assignment I use the code similar to the PART1A in this code by measuring the Cross validation accuracy I set the different paramters at different values for finding the cross validation accuracy so my accuracy that gives me best result so for that I try different vaues by changing the values of Gamma, Kernel type , number of features included and also in some cases changes the regularization parameter i.e C and while trying for the polynomial kernel I also chnges the degree for the kernel so form all the data I find a band values i.e at gamma = 0.06 - 0.08 and C = 1 and Kernel = Radial Base kernel that gives me the best validation accuracy so I Tried that submission on the Kaggle and Finally I got my best submission on the value Gamma = 0.07 and the value of accuracy on the sample test case is 0.96625 that is pretty good accuracy.