ELL409
Assignment 2

Submitted by:
Raghav Gupta
2017EE10544

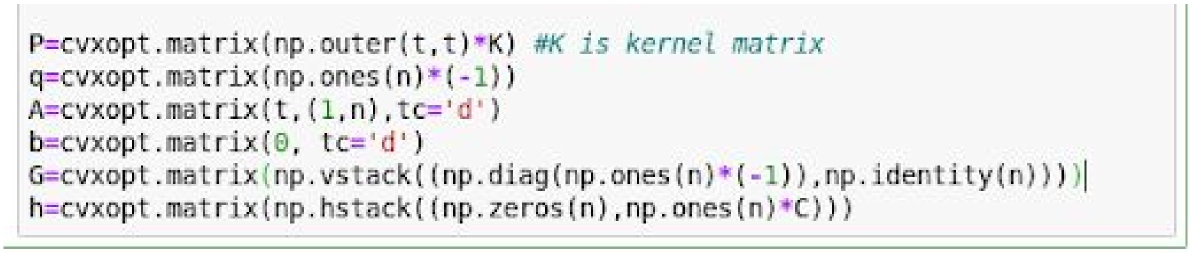
Part 1

Convex Optimization:

- Classes 0 and 1 were chosen and hyperparameters were tuned using 4-fold cross-validation. Classifier was trained using sklearn library.
- Using the tuned hyperparameters, classifier was trained using CVXOPT library.
- The SVM dual problem was converted to the following form to be fed into CVXOPT.

$$\begin{array}{ll} \min\limits_{x} & \frac{1}{2}x^\top P x + q^\top x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

- Matrix x is the matrix which contains the lagrange multipliers and whose optimum value needs to be found to minimize the expression given the constraints.
- Matrix P,q,G,h,A,b are fed to CVXOPT which gives the required lagrange multipliers. The matrices are given below:



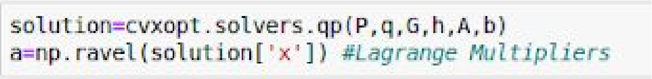
```
P=cvxopt.matrix(np.outer(t,t)*K) #K is kernel matrix
q=cvxopt.matrix(np.ones(n)*(-1))
A=cvxopt.matrix(t,(1,n),tc='d')
b=cvxopt.matrix(0, tc='d')
G=cvxopt.matrix(np.vstack((np.diag(np.ones(n)*(-1)),np.identity(n))))
h=cvxopt.matrix(np.hstack((np.zeros(n),np.ones(n)*C)))
```

- After finding the lagrange multipliers, the predictions were calculated as follows:

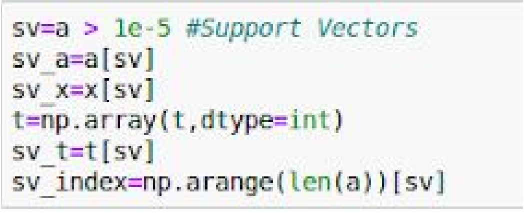
$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b.$$

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right)$$

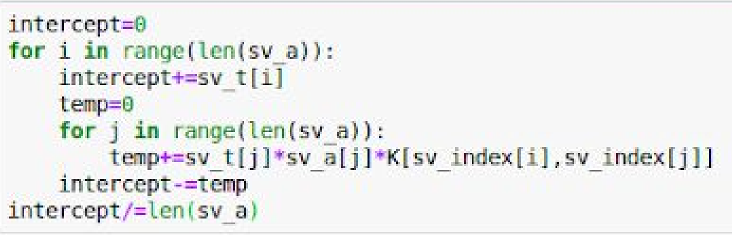
- The code snippets shows how CVX output was used to construct the classifier.



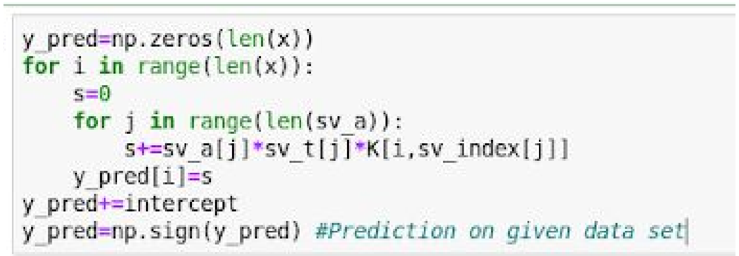
```
solution=cvxopt.solvers.qp(P,q,G,h,A,b)
a=np.ravel(solution['x']) #Lagrange Multipliers
```



```
sv=a > 1e-5 #Support Vectors
sv_a=a[sv]
sv_x=x[sv]
t=np.array(t,dtype=int)
sv_t=t[sv]
sv_index=np.arange(len(a))[sv]
```



```
intercept=0
for i in range(len(sv_a)):
    intercept+=sv_t[i]
    temp=0
    for j in range(len(sv_a)):
        temp+=sv_t[j]*sv_a[j]*K[sv_index[i],sv_index[j]]
    intercept-=temp
intercept/=len(sv_a)
```



```
y_pred=np.zeros(len(x))
for i in range(len(x)):
    s=0
    for j in range(len(sv_a)):
        s+=sv_a[j]*sv_t[j]*K[i,sv_index[j]]
    y_pred[i]=s
y_pred+=intercept
y_pred=np.sign(y_pred) #Prediction on given data set
```

- This process was done for Linear and RBF kernels. The results are given below:

1. Linear Kernel:

The hyperparameter C was found in the next part(Binary Classification)   to be 0.1. For C=0.1, the classifier trained using CVXOPT gave almost similar results.

Library used	Support Vectors	Cross-Validation Accuracy(4 fold)
Sklearn	15	1.0
CVXOPT	16	1.0

Support Vector Indices(Sklearn):

```
[ 35  43  79  89 126 180 223 271 306 440 457 521 526 573 594]
```

Support Vector Indices(CVXOPT):

```
[ 35  43  79  89 126 180 223 271 306 440 457 521 526 529 573 594]
```

2. RBF Kernel:

The hyperparameter C and Gamma were found in the next part(Binary Classification) to be 0.1 and 0.01 respectively. Using these hyperparameters, the classifier trained using CVXOPT gave almost similar results.

Library used	Support Vectors	Cross-Validation Accuracy(4 fold)
Sklearn	132	0.9983
CVXOPT	132	0.9983

Support Vector Indices(Sklearn):

```
[  9  10  19  20  33  35  36  40  43  44  46  52  53  58  62  68  70  72
  75  76  78  79  80  81  87  89  90  95 126 133 135 138 157 160 169 174
 175 178 180 181 182 187 188 199 207 212 218 223 224 236 246 248 250 255
 256 259 262 269 271 273 276 279 286 287 296 298 299 302 305 306 315 316
 329 332 334 335 336 340 346 348 350 352 356 358 360 368 380 381 387 388
 389 397 403 406 423 427 436 439 440 454 457 460 464 468 478 490 492 509
 517 521 526 527 529 531 533 534 535 541 549 553 555 557 558 559 569 573
 574 586 587 588 589 594]
```

Support Vector Indices(CVXOPT):

```
[  9  10  19  20  33  35  36  40  43  44  46  52  53  58  62  68  70  72
  75  76  78  79  80  81  87  89  90  95 126 133 135 138 157 160 169 174
 175 178 180 181 182 187 188 199 207 212 218 223 224 236 246 248 250 255
 256 259 262 269 271 273 276 279 286 287 296 298 299 302 305 306 315 316
 329 332 334 335 336 340 346 348 350 352 356 358 360 368 380 381 387 388
 389 397 403 406 423 427 436 439 440 454 457 460 464 468 478 490 492 509
 517 521 526 527 529 531 533 534 535 541 549 553 555 557 558 559 569 573
 574 586 587 588 589 594]
```

Conclusions:

Convex Optimization gave almost similar results to that of a standard library(Sklearn). For linear kernel, 1 support vector was extra compared to the library one(index 529). For RBF kernel, the support vectors matched exactly. The cross-validation accuracies obtained through both classifiers were same.

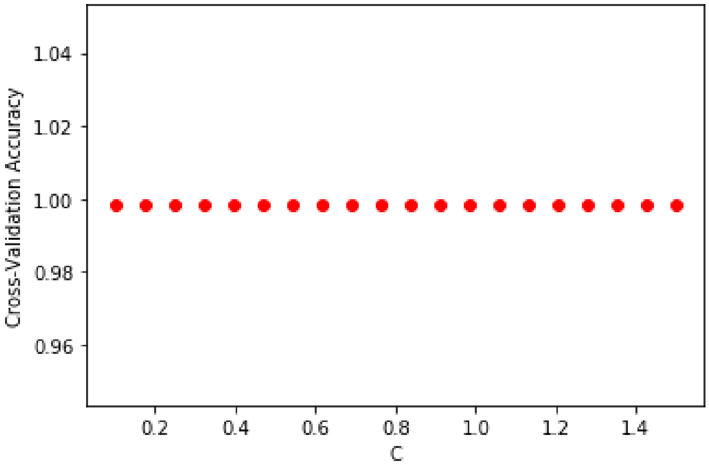
Binary Classification :

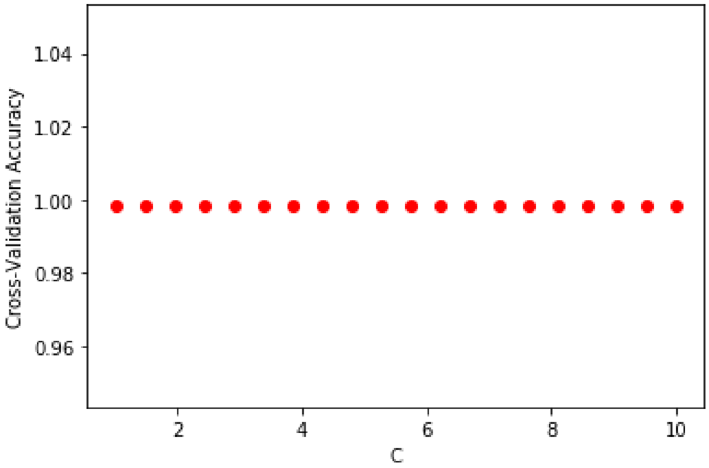
Classes 0 and 1:

- 4-Fold cross-validation was used to tune hyperparameters for different kernels.
- The SVM library used was sklearn.
- Results for different kernels and using all 25 features are given below:

1. Linear Kernel:

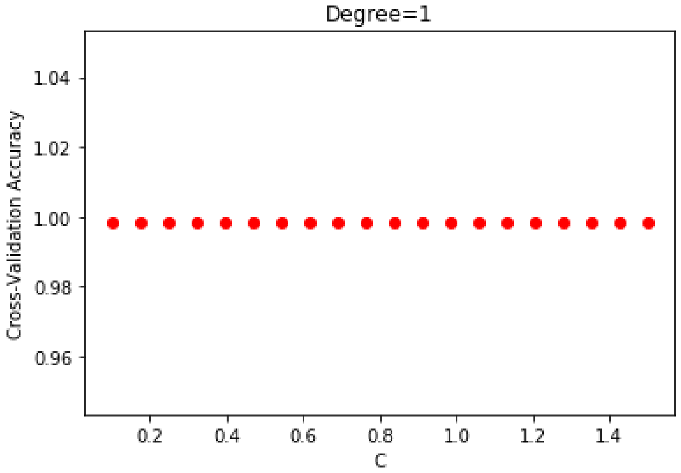
The cross-validation accuracy was almost the same for different values of hyperparameter C.
The average accuracy across the 4 folds was 0.998 for different values if C in between 0.1 and 10. C=0.1 was chosen.
The model trained thus, gave an accuracy of 1.0 on the given data.
The plots are given below for Cross-validation accuracy versus C.

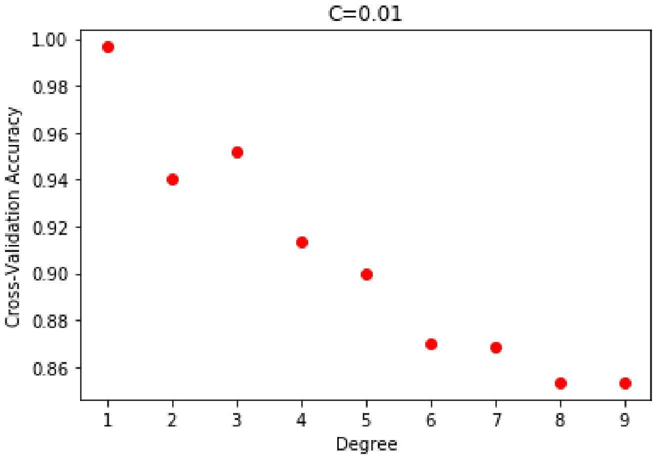


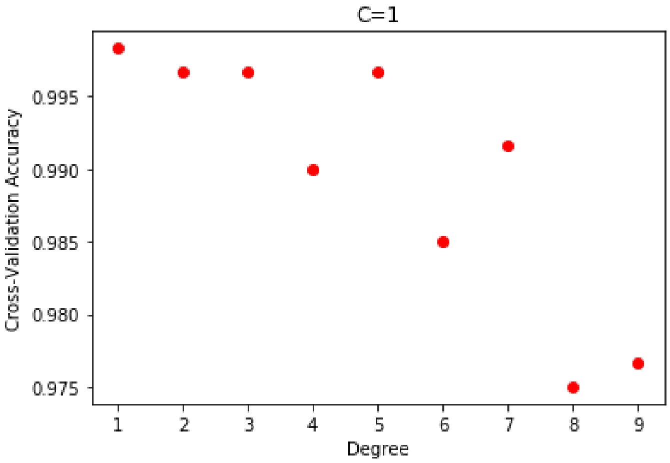


2. Polynomial Kernel:

The values of hyperparameters C and degree were chosen so as to maximize cross-validation accuracy.The plots are given below.





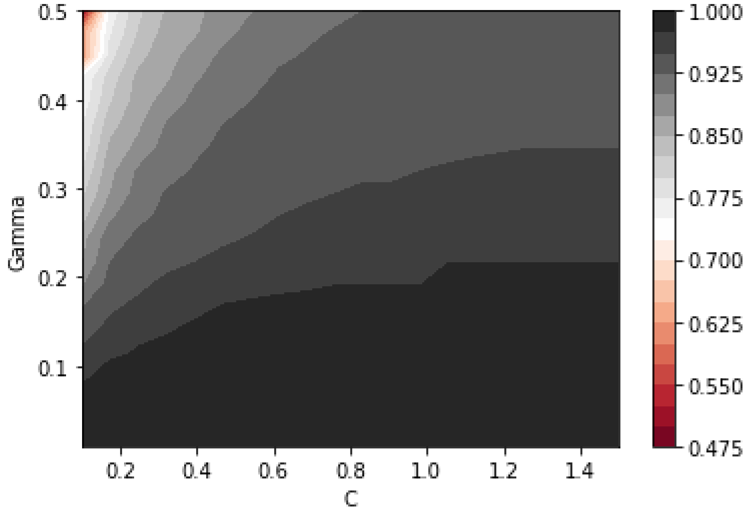


The optimum degree was chosen as 1 as it gave maximum cross-validation of 0.998. The variation with C was not noticeable as clear from the first plot.

Thus C=0.1 and Degree=1 were the hyperparameters chosen.

3. RBF Kernel:

The values of hyperparameters C and Gamma were chosen so as to maximize cross-validation accuracy.The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.

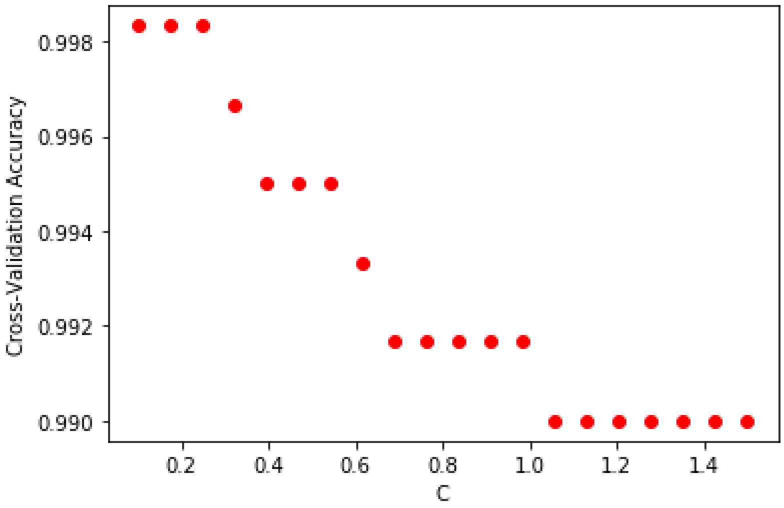


C=0.1 and Gamma=0.01 gave 0.998 Cross-Validation accuracy and thus were the chosen hyperparameters.
Using these hyperparameters the trained model gave an accuracy of 0.99 on the data set.

- Using only first 10 features, the results using different kernels are given below:

1. Linear Kernel:

The cross-validation accuracy remained almost 0.99 by varying C and thus C=0.1 was the chosen parameter.
The model trained using C=0.1 gave accuracy of 0.99
The plot for Cross-Validation Accuracy versus C is as given below.

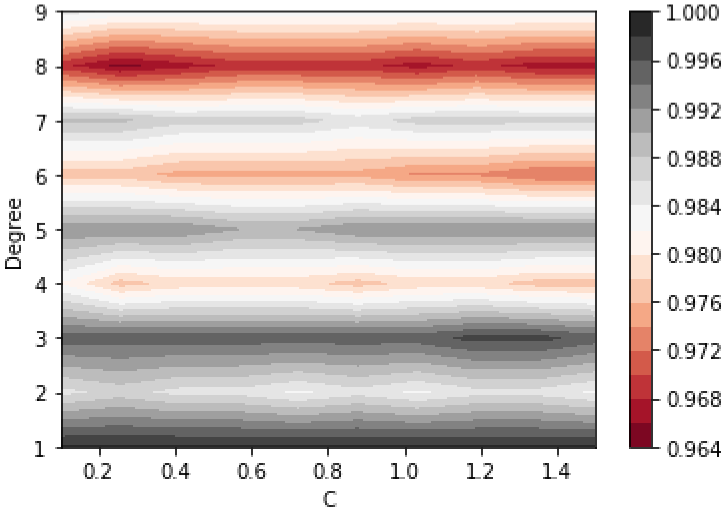


2. Polynomial Kernel:

Changing C had no significant change cross-validation accuracy. So, C=0.1 was chosen

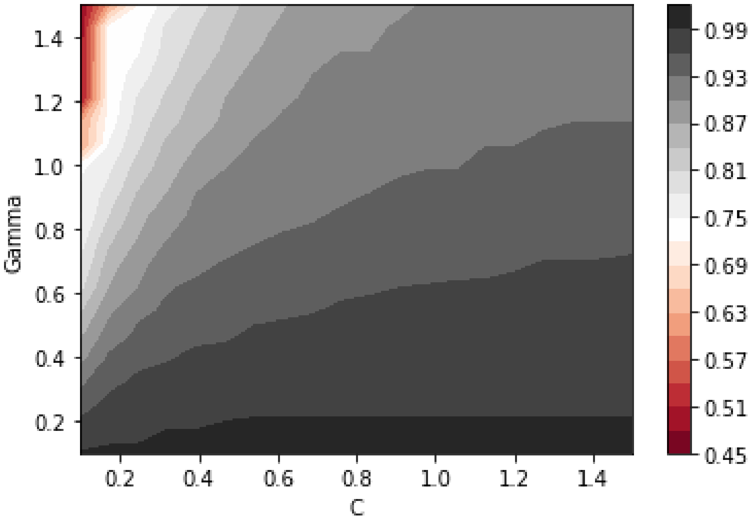
Degree=1 gave highest cross-validation accuracy which was 0.998. Thus C=0.1 and Degree=1 were the chosen hyperparameters. The model trained gave accuracy of 0.996 on the data set.

The plot below shows variation of cross-validation accuracy with C and Degree.



3. RBF Kernel:

The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.



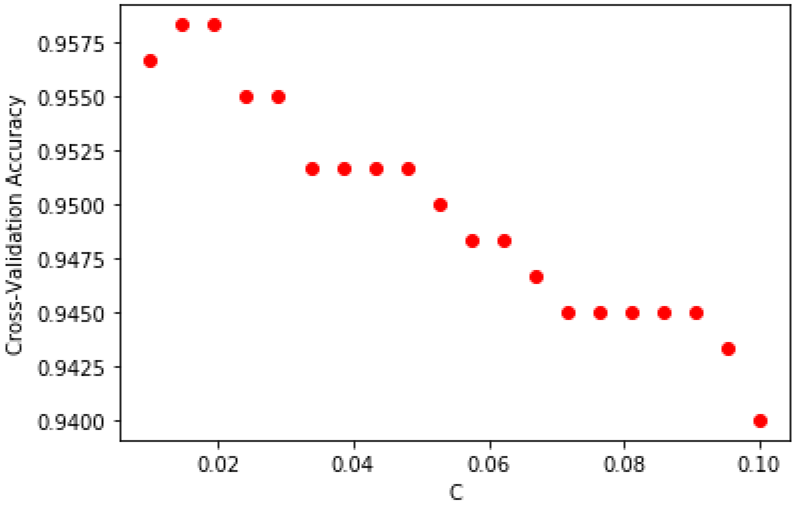
C=0.1 and Gamma=0.01 gave 0.998 Cross-Validation accuracy and thus were the chosen hyperparameters.
Using these hyperparameters the trained model gave an accuracy of 0.99 on the data set.

Classes 2 and 3:

- The same process was repeated with classes 2 and 3 being the chosen ones.
- Using all 25 features the results are given below.

1. Linear Kernel:

C=0.015 gave the best cross-validation accuracy of 0.9583.The model trained using C=0.015 gave accuracy of 0.95.
The plot for Cross-Validation Accuracy versus C is as given below.

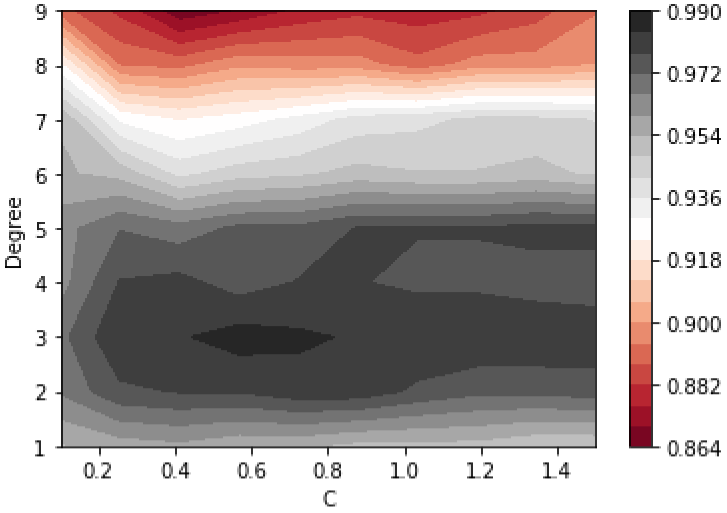


2. Polynomial Kernel:

C=0.56 and Degree=3 gave highest cross-validation accuracy which was 0.986.

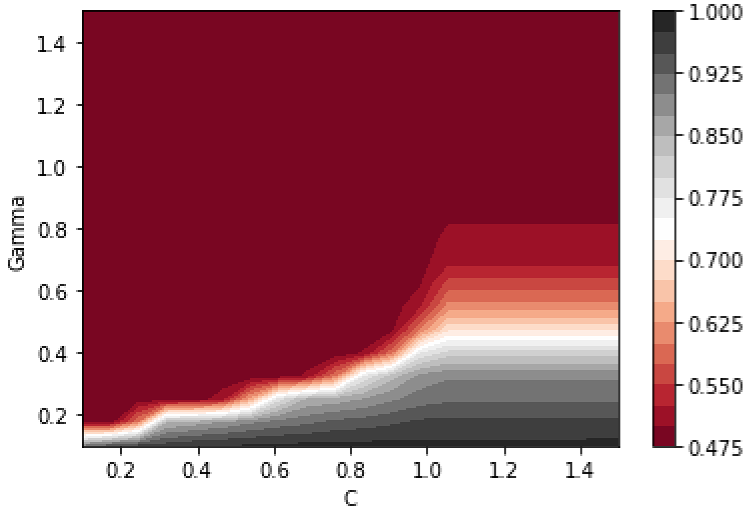
Thus C=0.56 and Degree=3 were the chosen hyperparameters. The model trained gave accuracy of 1.0 on the data set.

The plot below shows variation of cross-validation accuracy with C and Degree.



3. RBF Kernel:

The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.

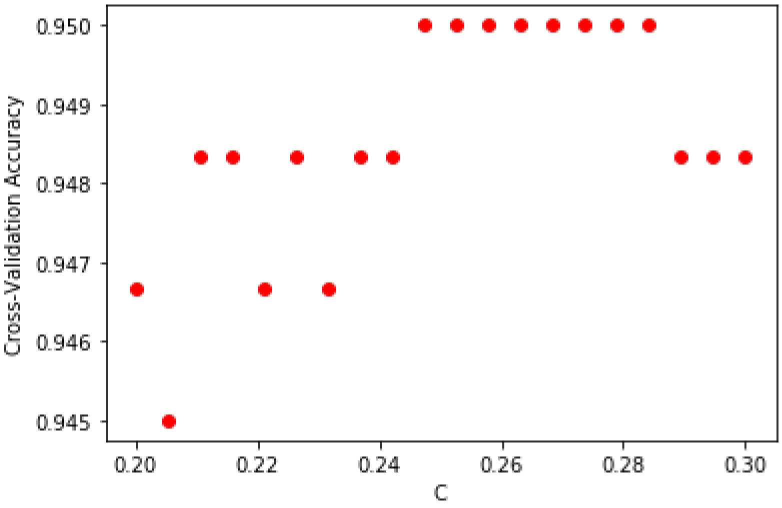


C=0.83 and Gamma=0.1 gave highest Cross-Validation accuracy which was 0.98 and thus were the chosen hyperparameters.
Using these hyperparameters the trained model gave an accuracy of 1.0 on the data set.

- Using only first 10 features, the results using different kernels are given below.

1. Linear Kernel:

C=0.247 gave the best cross-validation accuracy of 0.95.The model trained using C=0.247 gave accuracy of 0.95.
The plot for Cross-Validation Accuracy versus C is as given below.

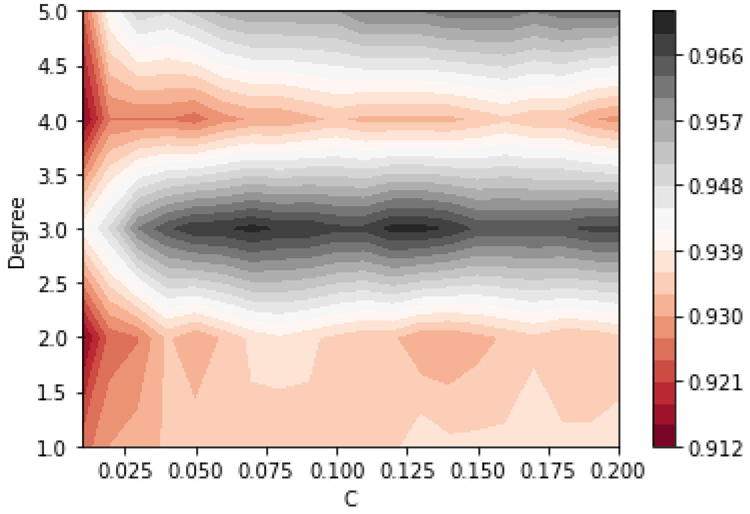


2. Polynomial Kernel:

C=0.07 and Degree=3 gave highest cross-validation accuracy which was 0.97.

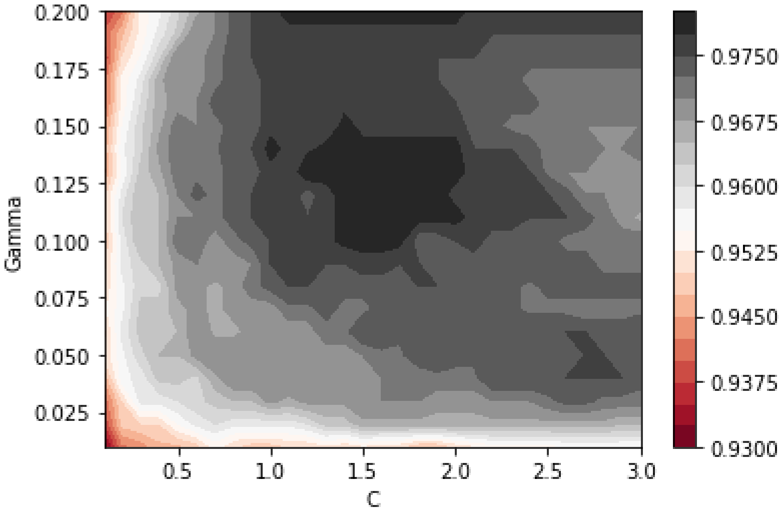
Thus, C=0.07 and Degree=3 were the chosen hyperparameters. The model trained gave accuracy of 0.986 on the data set.

The plot below shows variation of cross-validation accuracy with C and Degree.



3. RBF Kernel:

The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.



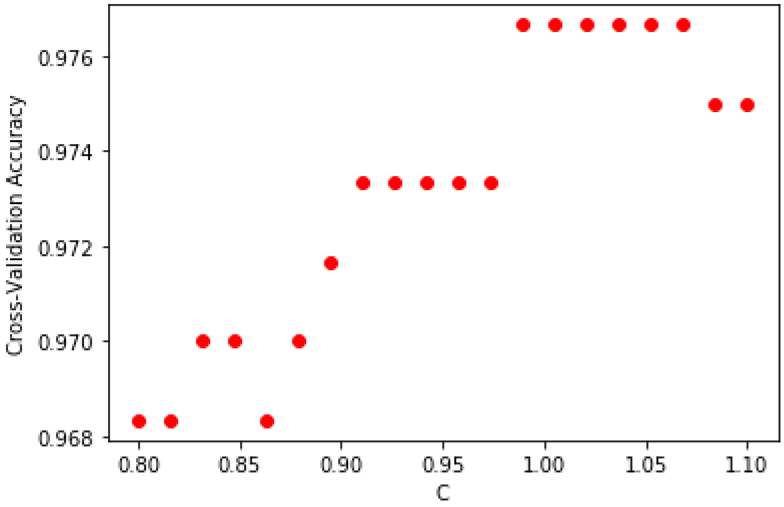
C=1.2 and Gamma=0.13 gave highest Cross-Validation accuracy which was 0.98 and thus were the chosen hyperparameters.
Using these hyperparameters the trained model gave an accuracy of 0.998 on the data set.

Classes 8 and 9:

- The same process was repeated with classes 2 and 3 being the chosen ones.
- Using all 25 features the results are given below.

1. Linear Kernel:

C=1 gave the best cross-validation accuracy of 0.976.The model trained using C=1 gave accuracy of 0.99.
The plot for Cross-Validation Accuracy versus C is as given below.

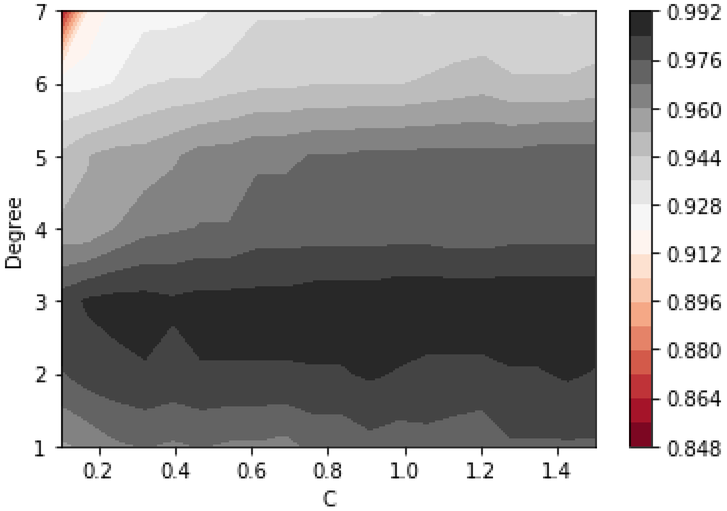


2. Polynomial Kernel:

C=1 and Degree=3 gave highest cross-validation accuracy which was 0.99.

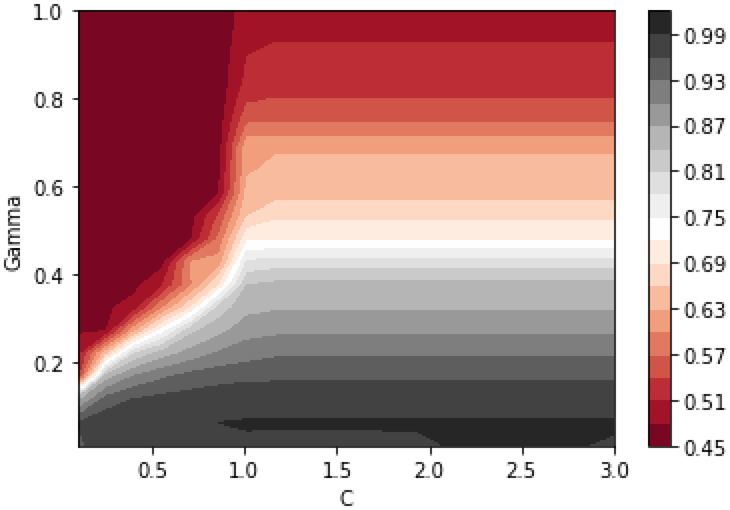
Thus C=1 and Degree=3 were the chosen hyperparameters. The model trained gave accuracy of 1.0 on the data set.

The plot below shows variation of cross-validation accuracy with C and Degree.



3. RBF Kernel:

The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.

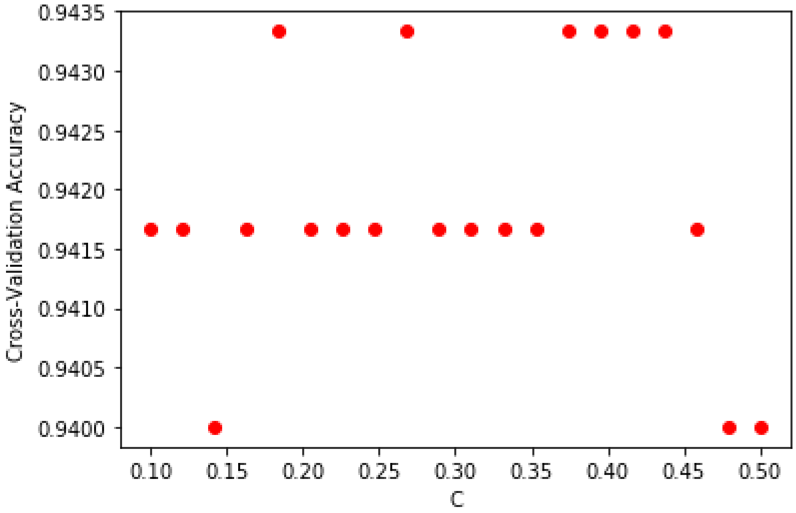


C=0.83 and Gamma=0.04 gave highest Cross-Validation accuracy which was 0.99 and thus were the chosen hyperparameters.
Using these hyperparameters the trained model gave an accuracy of 1.0 on the data set.

- Using only first 10 features, the results using different kernels are given below.

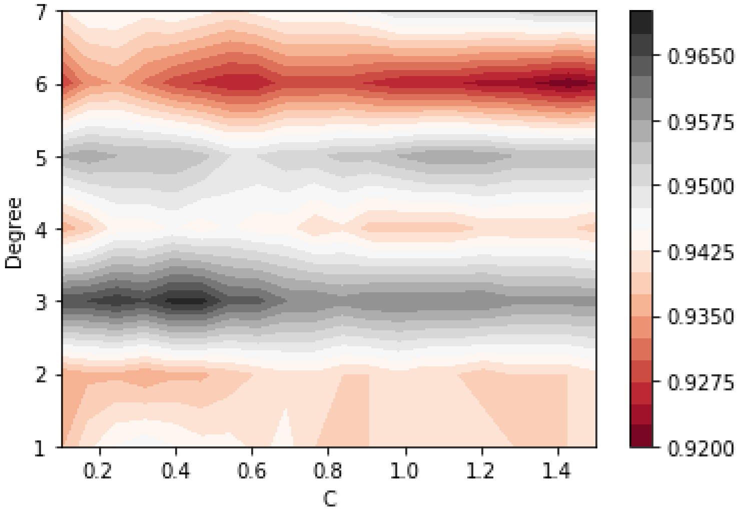
1. Linear Kernel:

C=0.18 gave the best cross-validation accuracy of 0.943.The model trained using C=0.247 gave accuracy of 0.96.
The plot for Cross-Validation Accuracy versus C is as given below.



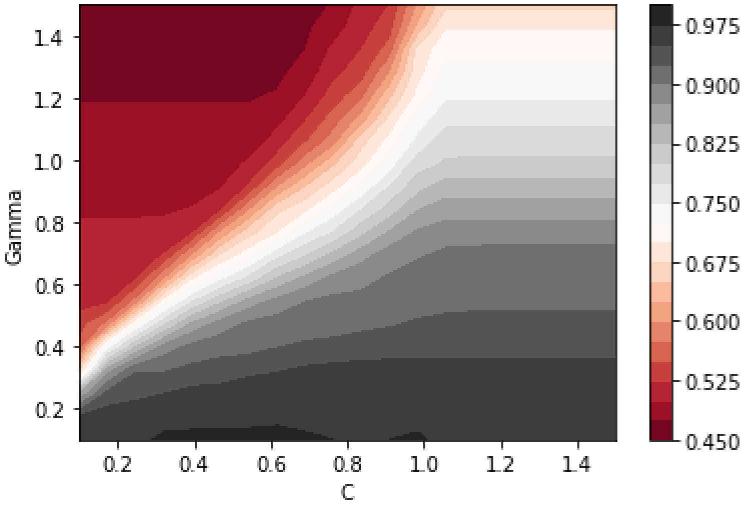
2. Polynomial Kernel:

C=0.4 and Degree=3 gave highest cross-validation accuracy which was 0.97.
Thus, C=0.4 and Degree=3 were the chosen hyperparameters. The model trained gave accuracy of 0.99 on the data set.
The plot below shows variation of cross-validation accuracy with C and Degree.



3. RBF Kernel:

The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.



C=0.4 and Gamma=0.1 gave highest Cross-Validation accuracy which was 0.98 and thus were the chosen hyperparameters. Using these hyperparameters the trained model gave an accuracy of 0.99 on the data set.

Conclusions:

- RBF kernel and linear kernel gave better results than linear kernel.
- In the contour plots the regions in red colour with low values of C and Gamma correspond to underfitting while regions in red colour with high values of C and Gamma correspond to overfitting. The black regions correspond to good fitting.
- Using 10 features, instead of 25 led to almost similar results but on the lower side. The highest cross-validation accuracies were a little less or similar to that obtained with 25 features. This is because a high dimensional data has a better chance of classification. The tuned hyperparameters vary when 10 features and 25 features were used.
- Best degree obtained for Polynomial Kernel was most of the times found to be 3 whereas other hyperparameters varied for other kernels.
- When different pairs of classes were used the hyperparameters for best setting varied. Thus, the hyperparameter settings were dependent on classes chosen.

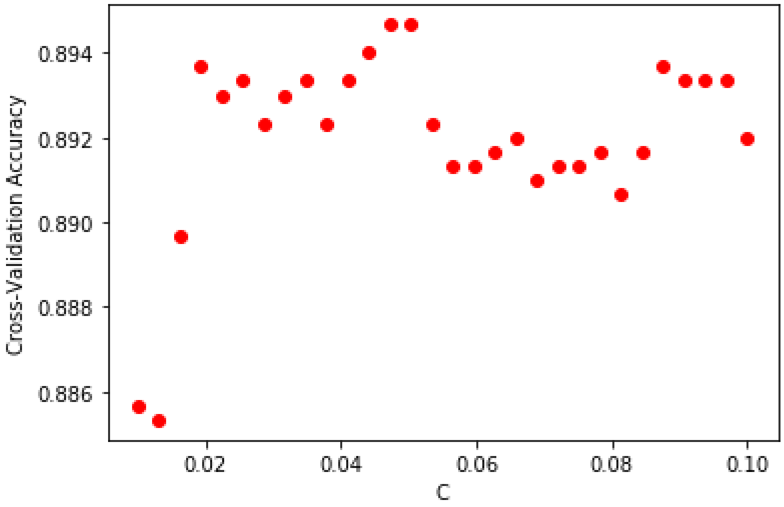
Multiclass Classification:

- Sklearn library was used to train a classifier for all 10 classes.
- The multiclass support is handled according to a one-vs-one scheme.
- Different kernels were tried out and hyperparameters were tuned using 4-fold Cross-validation.
- The results using all 25 features for different kernels are given below:

1. Linear Kernel:

C=0.05 gave the best cross-validation accuracy of 0.895.The model trained using C=0.05 gave accuracy of 0.917.

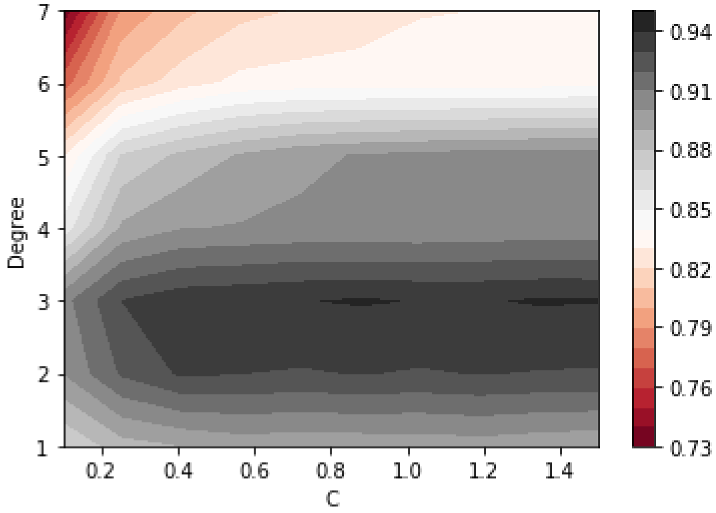
The plot for Cross-Validation Accuracy versus C is as given below.



2. Polynomial Kernel:

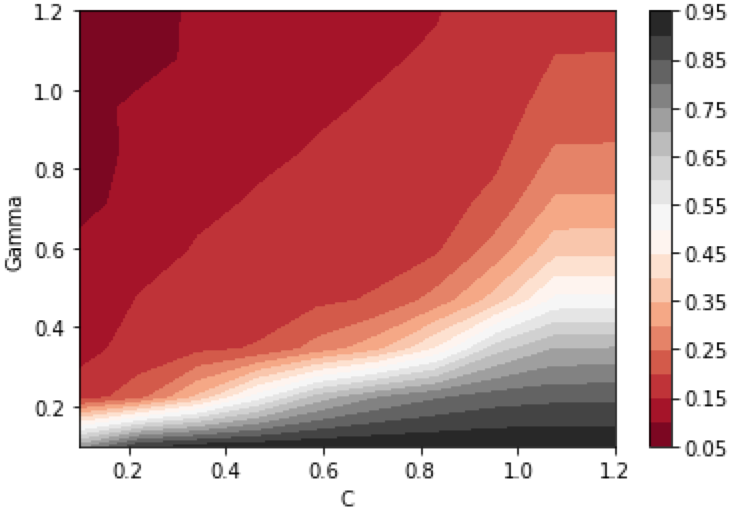
C=1.34 and Degree=3 gave highest cross-validation accuracy which was 0.94.

Thus, C=1.34 and Degree=3 were the chosen hyperparameters. The model trained gave accuracy of 0.923 on the data set.
The plot below shows variation of cross-validation accuracy with C and Degree.



3. RBF Kernel:

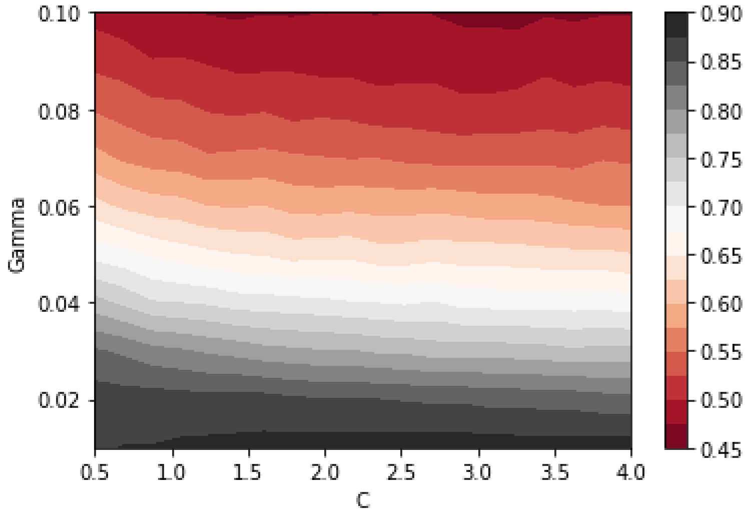
The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.



C=1.07 and Gamma=0.1 gave highest Cross-Validation accuracy which was 0.942 and thus were the chosen hyperparameters.
Using these hyperparameters the trained model gave an accuracy of 0.99 on the data set.

4. <u>Sigmoid Kernel:</u>

The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.



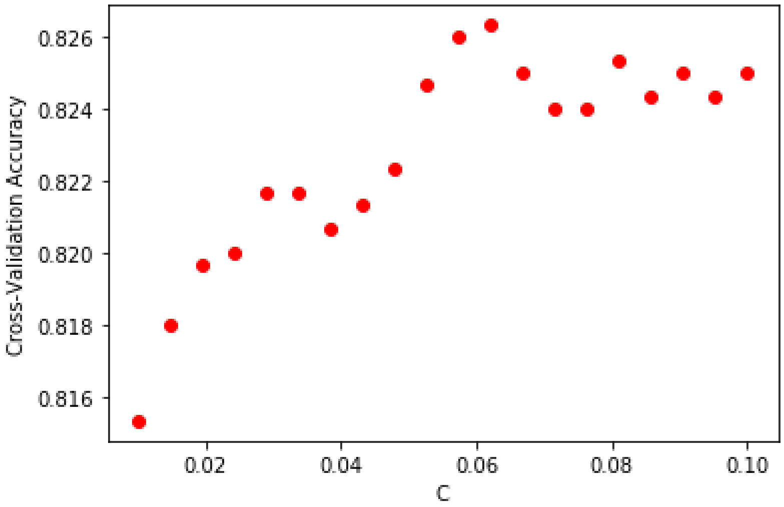
C=2.9 and Gamma=0.01 gave highest Cross-Validation accuracy which was 0.887 and thus were the chosen hyperparameters. Using these hyperparameters the trained model gave an accuracy of 0.903 on the data set.

- Using only first 10 features, the results using different kernels are given below.

1. <u>Linear Kernel:</u>

C=0.06 gave the best cross-validation accuracy of 0.826.The model trained using C=0.06 gave accuracy of 0.838.

The plot for Cross-Validation Accuracy versus C is as given below.

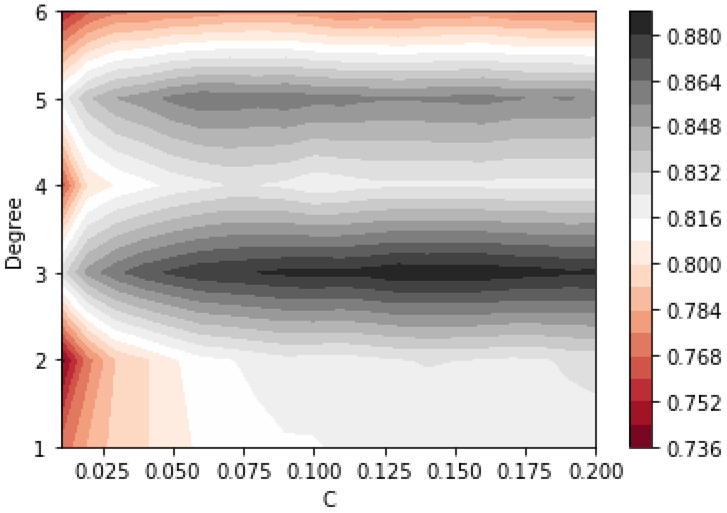


2. Polynomial Kernel:

C=0.13 and Degree=3 gave highest cross-validation accuracy which was 0.88.

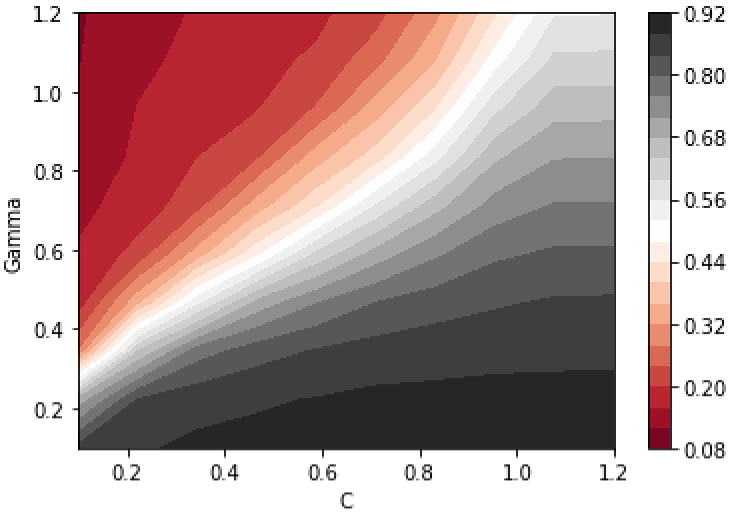
Thus, C=0.13 and Degree=3 were the chosen hyperparameters. The model trained gave accuracy of 0.938 on the data set.

The plot below shows variation of cross-validation accuracy with C and Degree.



3. RBF Kernel:

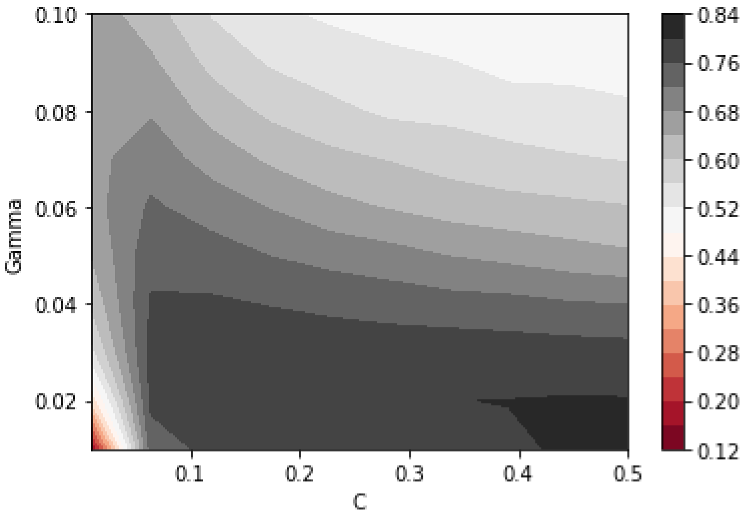
The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.



C=0.83 and Gamma=0.1 gave highest Cross-Validation accuracy which was 0.902 and thus were the chosen hyperparameters. Using these hyperparameters the trained model gave an accuracy of 0.96 on the data set.

4. Sigmoid Kernel:

The contour plot is given below which shows variation of cross-validation accuracy with C and Gamma.



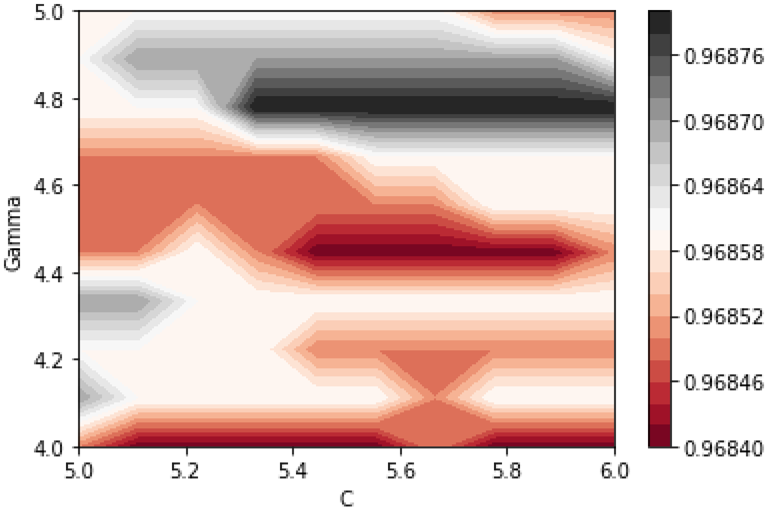
C=0.44 and Gamma=0.02 gave highest Cross-Validation accuracy which was 0.802 and thus were the chosen hyperparameters.
Using these hyperparameters the trained model gave an accuracy of 0.808 on the data set.

Conclusions:

- Same as binary classification RBF and Polynomial kernels gave better results than Linear and Sigmoid kernels. Hyperparameters were tuned using 4-fold cross validation.
- Increasing C and Gamma resulted in overfitting and correspond to the red regions in contour plots. Black regions correspond to good fitting.
- Hyperparameters vary from that obtained in binary classification. This is because for one versus one scheme, the classifier predicts by training each pair of binary classes. During the classification phases each classifier predicts one class and the class which has been predicted most is the answer. Thus, the hyperparameters for multiclass depend on each pair of binary classes rather than a particular pair.
- 25 features gave better cross-validation accuracies compared to using 10 features.

Part 2

- Different kernels were used and cross-validation was used to tune the hyper-parameters. But cross-validation accuracy that was found out using linear, polynomial and RBF kernels could not exceed 0.12.
- Because the range of input data was too large, normalizing the input data was preferred.
- Each feature was scaled by the factor of its maximum minus minimum.
- The results seemed to improve quite a lot after feature scaling for different kernels.
- The highest cross-validation accuracy was found using RBF kernel which was 0.9688 using C=5.33 and Gamma=4.77. The contour plot below shows variation of Cross-Validation accuracy with C and Gamma for RBF kernel.



- The prediction for the test set using the above trained SVM gave accuracy of 0.96 on kaggle.
- Thus, RBF kernel with C=5.33 and Gamma=4.77 was used for predictions.