ASSIGNMENT: 2

Intro to Machine Learning (ELL784)

Rohan Kumar Bohara

Entry Number: 2021AMA2095

Part 1:

Firstly, Binary classification has been implemented by utilizing the packages LIBSVM and Sci-kit learn independently. This includes hyperparameter tuning and analysis followed by study of impact of features on classification by taking separate groups of 10 features at a time for each of these packages. This is followed by multi-class classification for the given data using the package LIBSVM followed by hyperparameter tuning and impact of features on validation scores. Throughout the analysis for Part 1A, we have taken 3:1 ratio for proportion of training and testing data for each algorithm, i.e., keeping 75% partition data for training purposes while rest 25% for testing purposes for measurement of metrics. This partition uses standard train_test_split for model selection by sci-kit learn package.

Binary Classification:

Taking First 2 classes out of 10 classes with all features

```
C_range = 10. ** np.arange(-3, 8)
gamma_range = 10. ** np.arange(-5, 4)
degree_range = np.arange(1,10)
coef0_range =np.arange(-15,-5)
```

```
with poly :

SVC(C=0.001, coef0=-15, degree=5, gamma=0.0001, kernel='poly', max_iter=10)
0.9979166666666667 train score 0.9916666666666667 test score
```

```
With RBF:
```

```
SVC(C=128.0, gamma=1.52587890625e-05, max_iter=1000) 0.8754166666666666 train
score 0.855 test score
```

Multi classification :

With RBF:

```
SVC(C=0.00390625, gamma=2.9802322387695312e-08, max_iter=10) 0.60375 train
score 0.585 test score
```

LIBSVM We evaluate our prediction of ‘class_labels’ for testing data (based on training data) w.r.t. actual ‘class_labels’ for testing part. Following plots have been obtained for accuracy vs various hyperparameters settings and multiple randomized instances of the same have been recorded:

1) Test accuracy for ‘polynomial’ kernel VS degree in kernel function

2) Test accuracy for ‘RBF’ kernel VS gamma

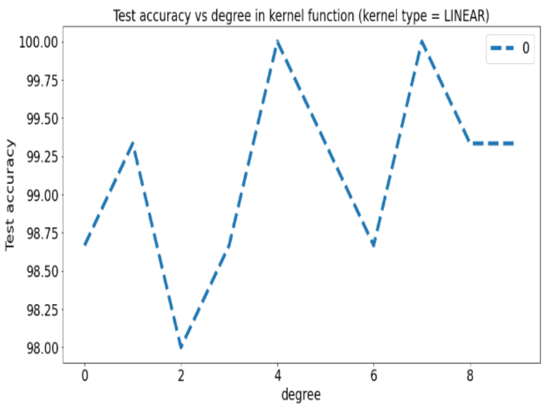
3) Test accuracy VS type of kernel function

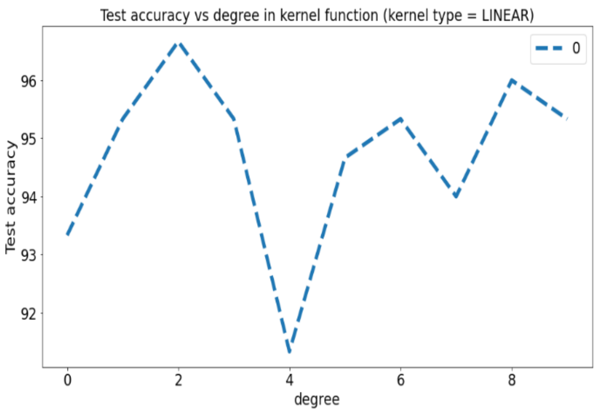
4) Test accuracy VS type of SVM

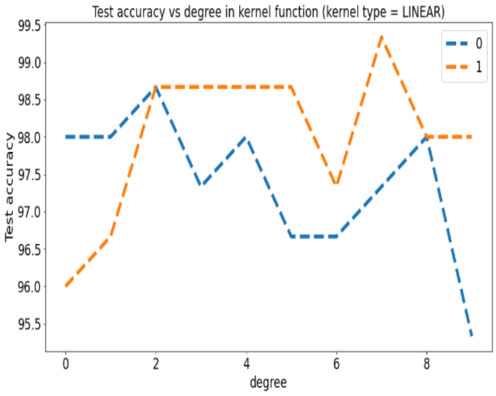
5) Test accuracy VS C-value

The following steps are taken for each of the hyperparameter variations :

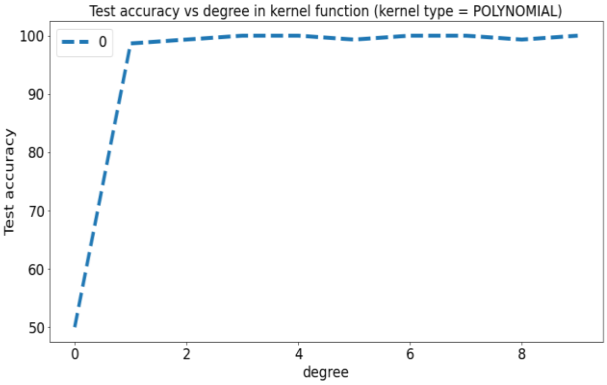
- Degree in kernel function : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- Gamma values : [2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0} ¸2^{1} ¸2^{2} ¸2^{3}]
- Type of kernel function : [0, 1, 2, 3]
- Type of SVM : [0, 1, 2, 3, 4]
- C value : [2^{-2}, 2^{-1}, 2^{0} ¸2^{1} ¸2^{2} ¸2^{3} , 2^{4} , 2^{5}, 2^{6}, 2^{7}]

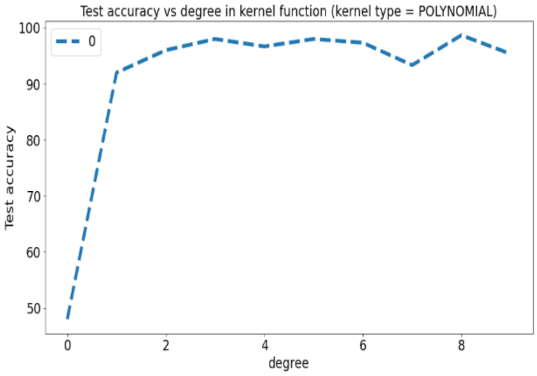


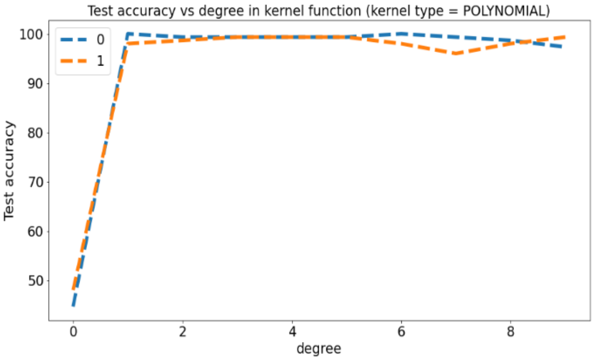


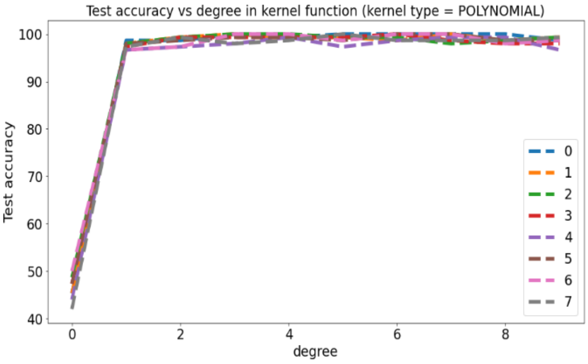


Plot type 1:









Plot type2:

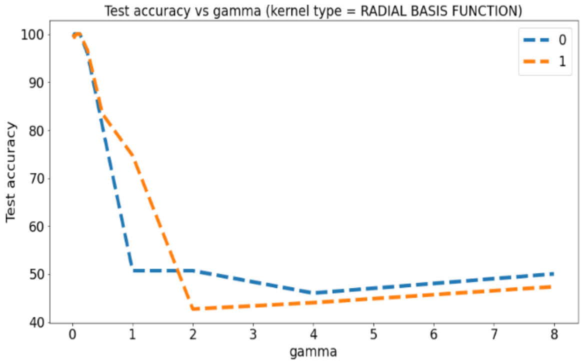


Fig. Binary Classification for Classes : 7, 8 (2 instances)

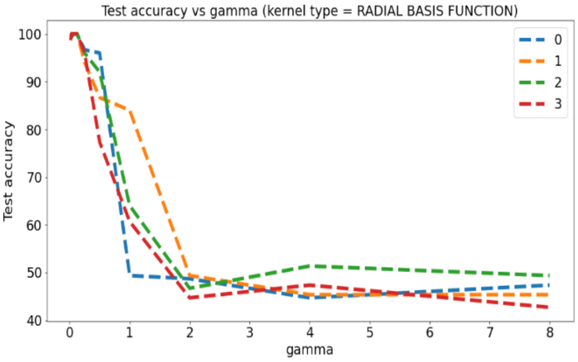


Fig. Binary Classification for Classes : 7, 8 (4 instances)

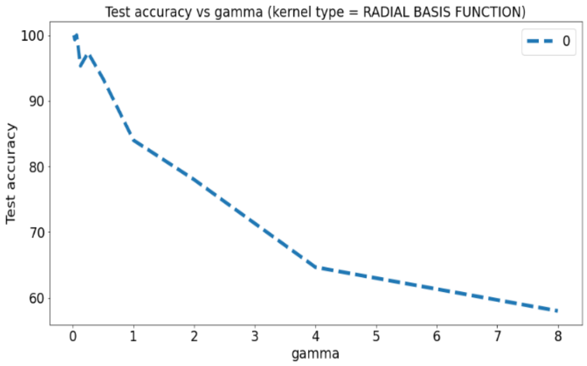


Fig. Binary Classification for Classes : 0, 1

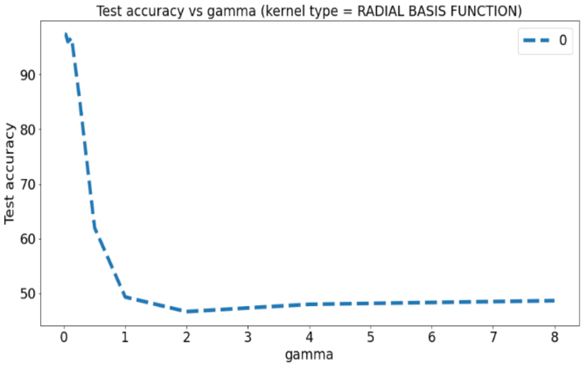
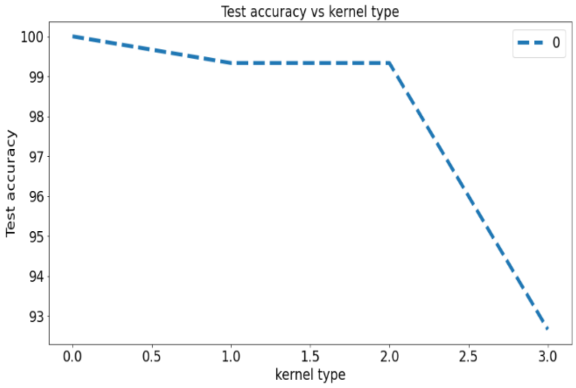


Fig. Binary Classification for Classes : 2, 3



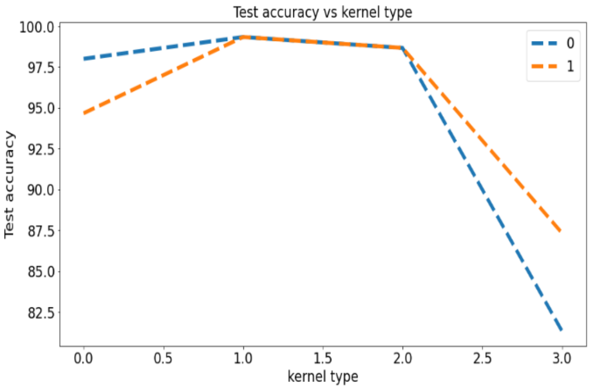
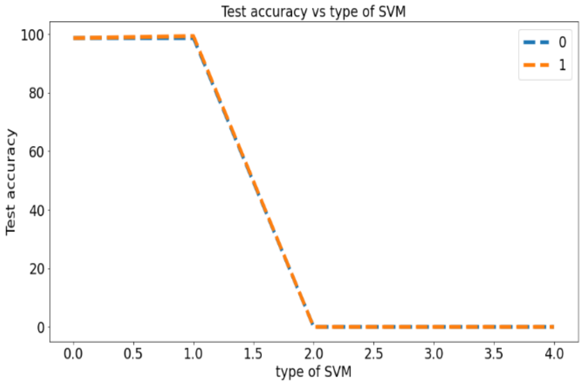


Fig. Binary Classification for Classes : 0, 1

7, 8 (2 instances)



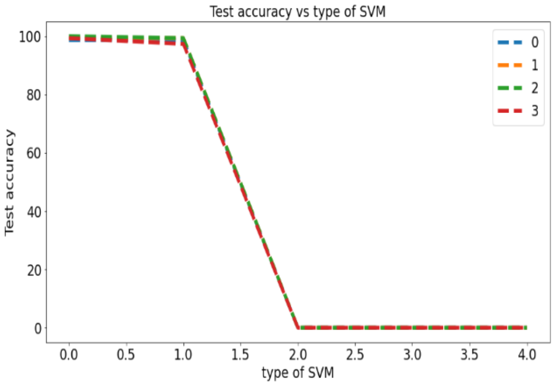


Fig. Binary Classification for Classes : : 0, 9 (2 instances)

0, 9 (4 instances)

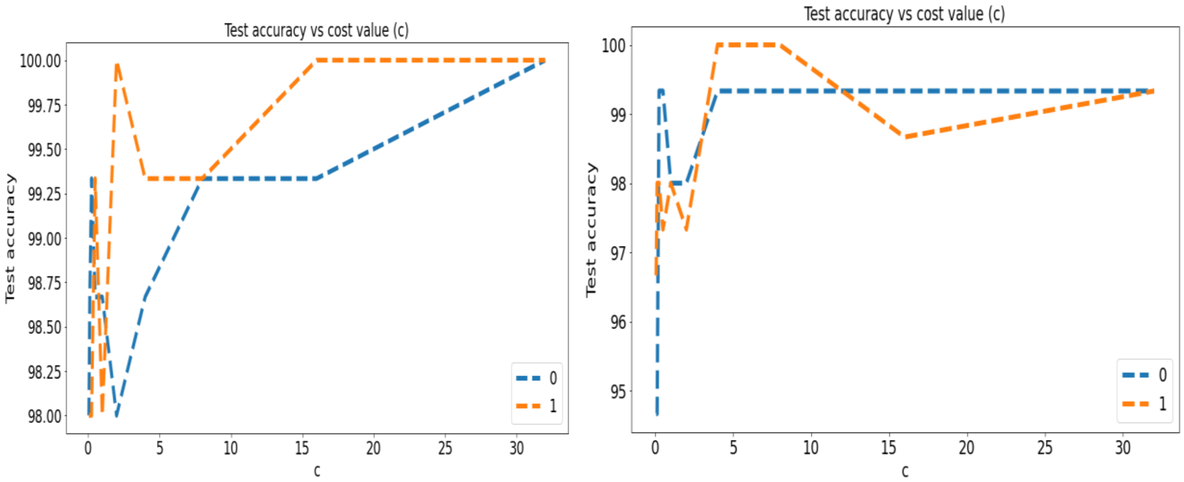


Fig. Binary Classification for Classes : 1, 8 (2 instances) 2, 7 (2 instances)

Conclusions

We find that best results are obtained for the following hyperparameters
:

`-s 0 -c 4 -t 2 -r 1 -d 3 -g 0.032`

1) Kernel Type = Radial Basis Function
2) Degree in kernel function = 3 (for POLYNOMIAL type)
3) Cost value = 4
4) Gamma Value = 0.032 (for RBF)
5) SVM type = 0 (C-SVC)

Results:

We notice that optimal hyperparameter settings change slightly with lesser
number of features under consideration and for different class labels.
However, when all features are considered, there is not much variation in
optimal hyperparameter conditions.

Binary Classification : Sci-kit Learn

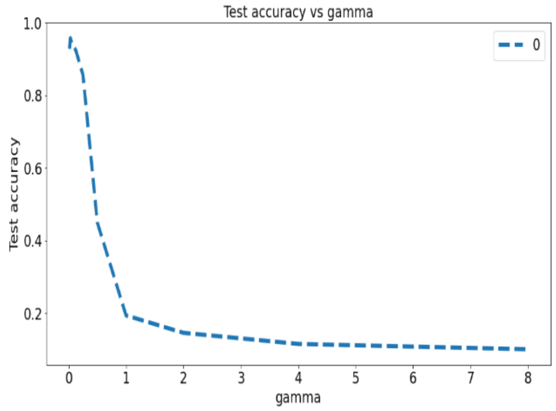
We evaluate our prediction of 'class_labels' for testing data (based on training data)
w.r.t. actual 'class_labels' for testing part. Following plots have been obtained for

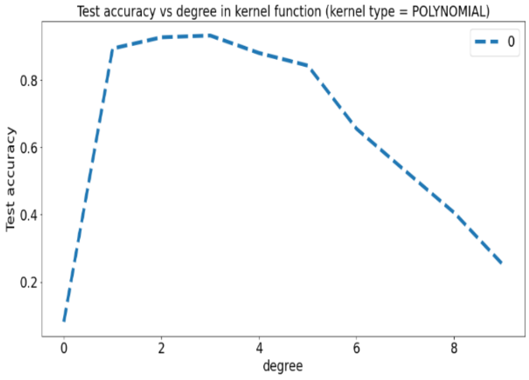
accuracy vs various hyperparameters settings :
1) Test accuracy for ‘polynomial’ kernel VS degree in kernel function
2) Test accuracy for ‘RBF’ kernel VS gamma
3) Test accuracy VS type of kernel function
4) Test accuracy VS C-value

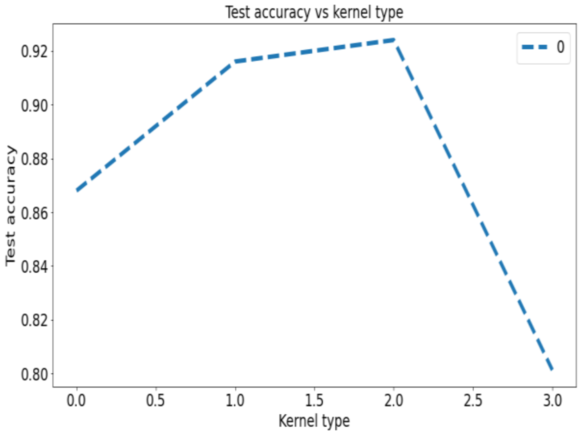
The following steps are taken for each of the hyperparameter variations :

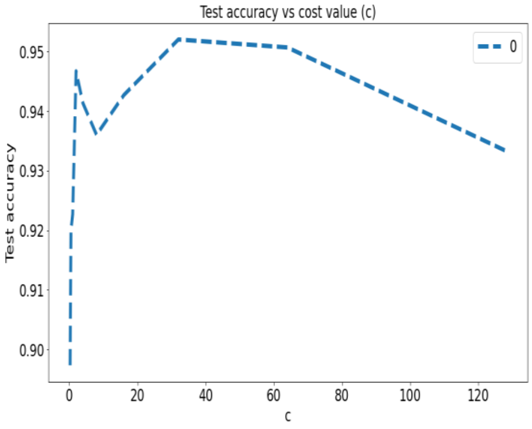
- Degree in kernel function : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- Gamma values : [2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0} ¸2^{1} ¸2^{2} ¸2^{3}]
- Type of kernel function : [0, 1, 2, 3]
- Type of SVM : [0, 1, 2, 3, 4]
- C value : [2^{-2}, 2^{-1}, 2^{0} ¸2^{1} ¸2^{2} ¸2^{3} , 2^{4} , 2^{5}, 2^{6}, 2^{7}]

a) Class labels 0,1

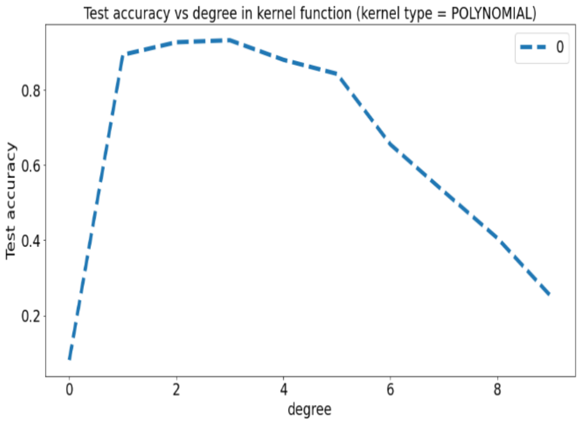


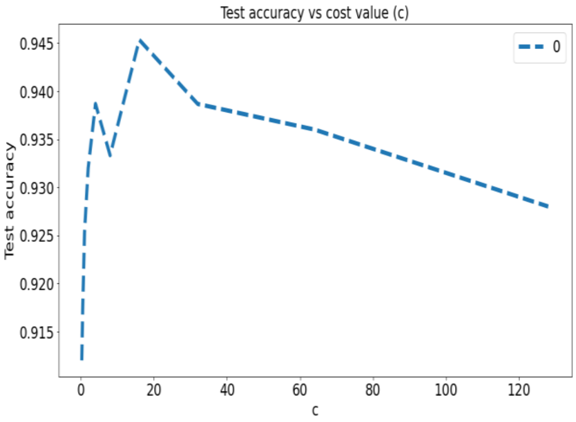


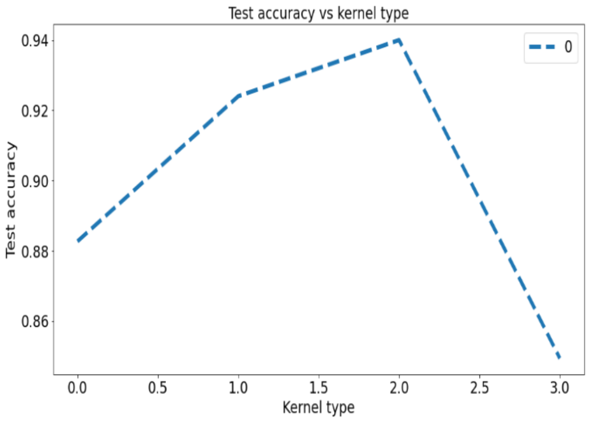


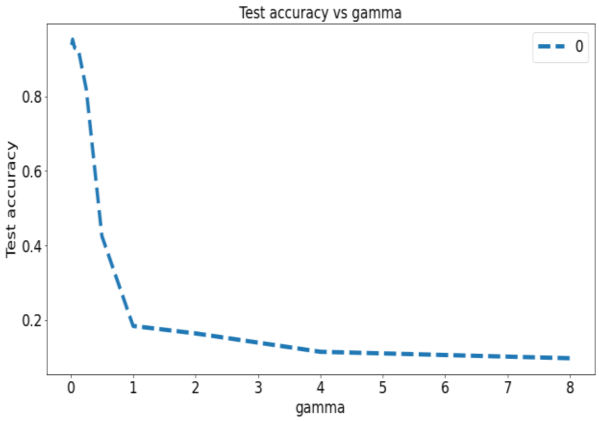


b) Class labels 2,3









Conclusions for hyperparameter Kernel Type = Radial Basis Function Degree in kernel function = 3 (for POLYNOMIAL type) Cost value = 4 Gamma Value = 0.032 (for RBF)

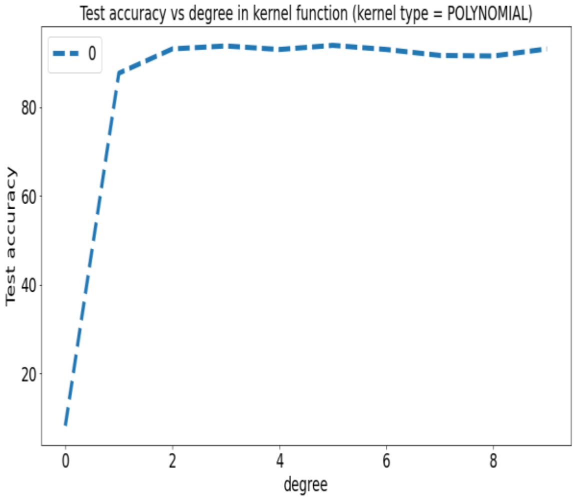
Multi-class Classification: LIBSVM

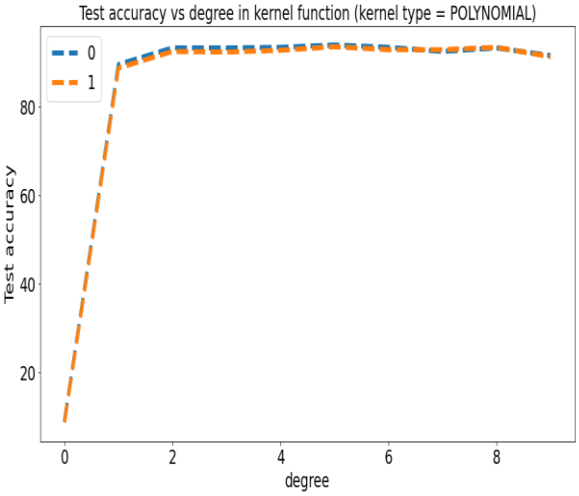
We evaluate our prediction of ‘class_labels’ for testing data (based on training data) w.r.t. actual ‘class_labels’ for testing part. Following plots have been obtained for accuracy vs various hyperparameters settings and multiple randomized instances of the same have been recorded:

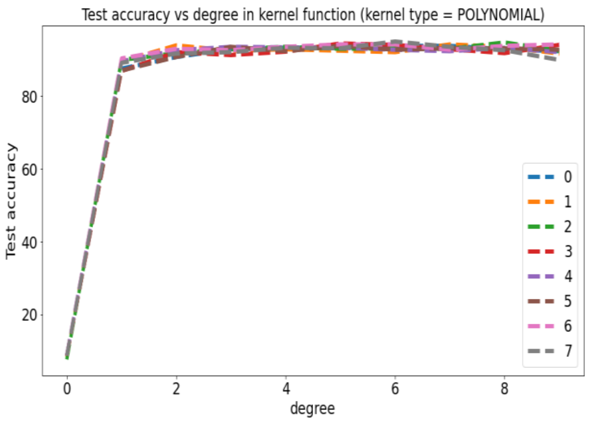
1) Test accuracy for ‘polynomial’ kernel VS degree in kernel function
2) Test accuracy for ‘polynomial’ kernel VS gamma
3) Test accuracy for ‘RBF’ kernel VS gamma
4) Test accuracy VS type of kernel function
5) Test accuracy VS C-value

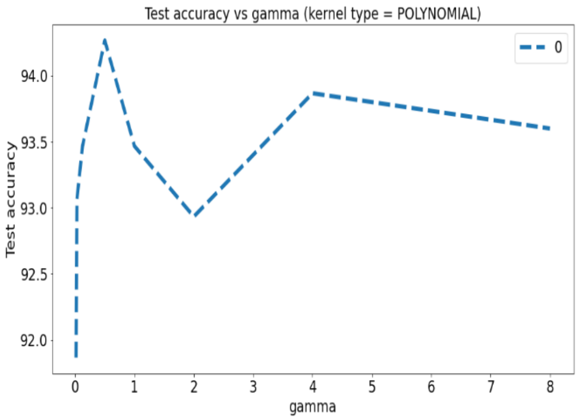
The following steps are taken for each of the hyperparameter variations :

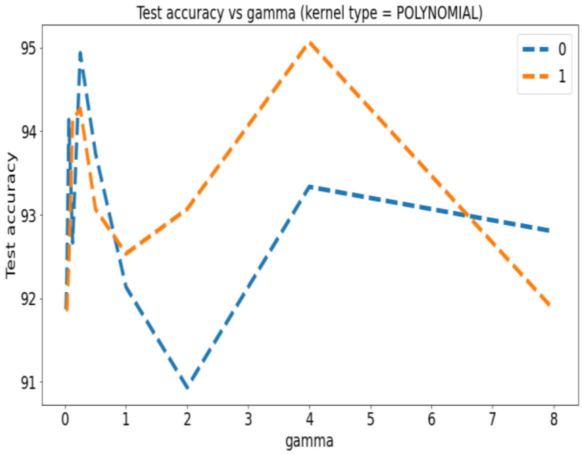
- Degree in kernel function : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- Gamma values : [2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0} ¸2^{1} ¸2^{2} ¸2^{3}]
- Type of kernel function : [0, 1, 2, 3]
- Type of SVM : [0, 1, 2, 3, 4]
- C value : [2^{-2}, 2^{-1}, 2^{0} ¸2^{1} ¸2^{2} ¸2^{3} , 2^{4} , 2^{5}, 2^{6}, 2^{7}]

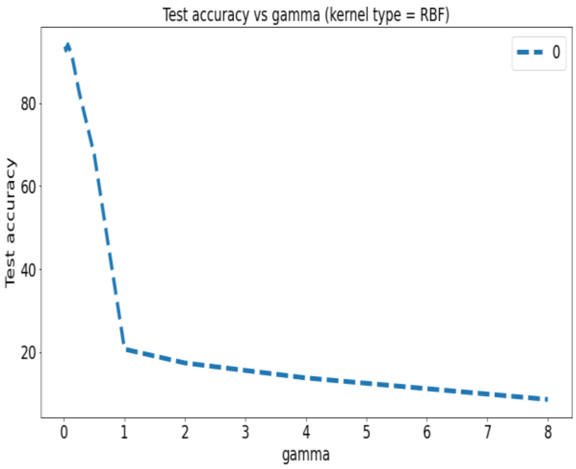


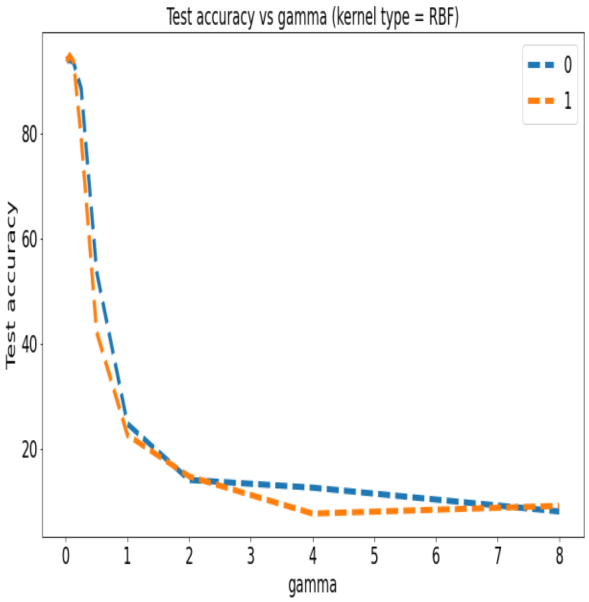


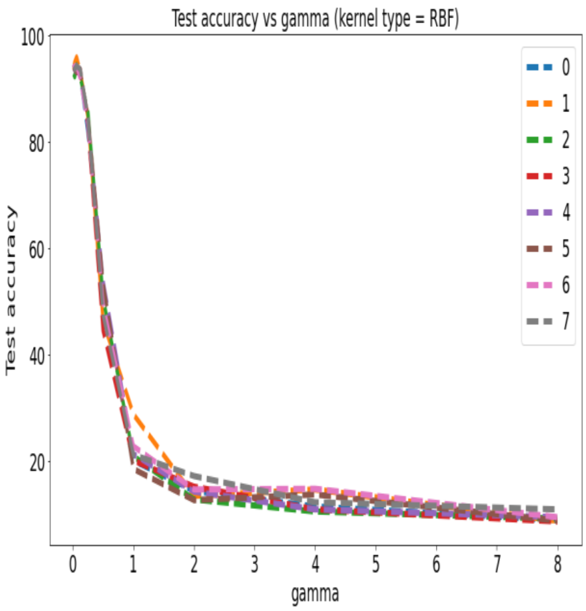

Test accuracy vs degree in kernel function (kernel type = POLYNOMIAL)

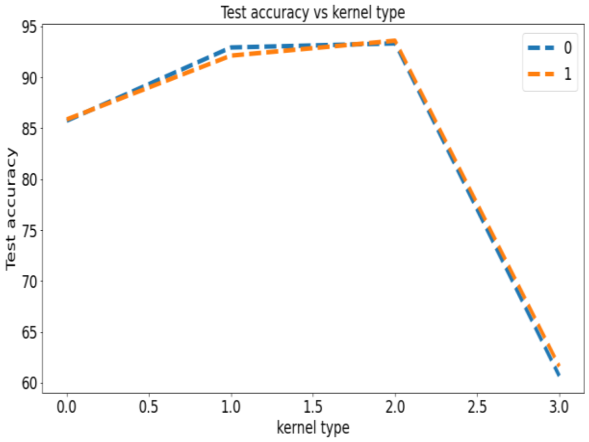

Test accuracy vs gamma (kernel type = POLYNOMIAL)

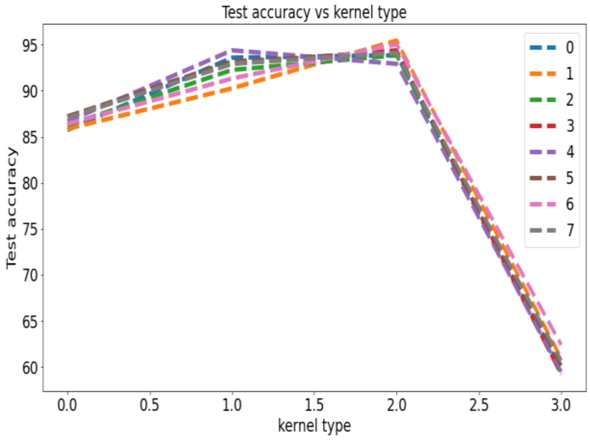

Test accuracy vs gamma (kernel type = POLYNOMIAL)

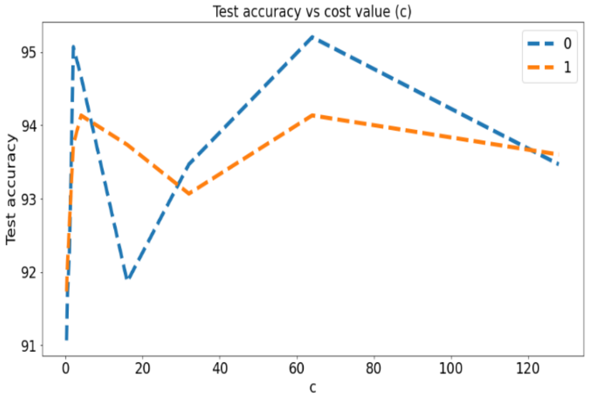

Test accuracy vs gamma (kernel type = RBF)

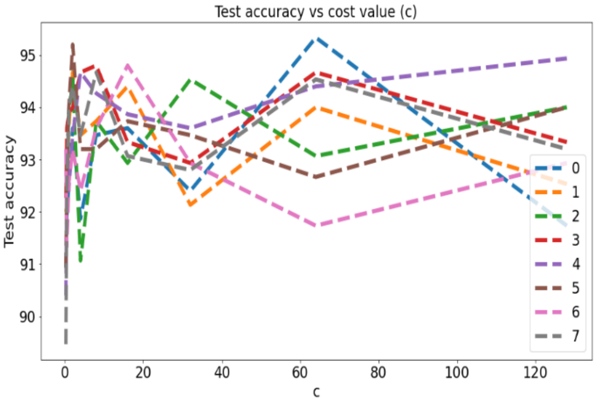

Test accuracy vs gamma (kernel type = RBF)


Test accuracy vs gamma (kernel type = RBF)









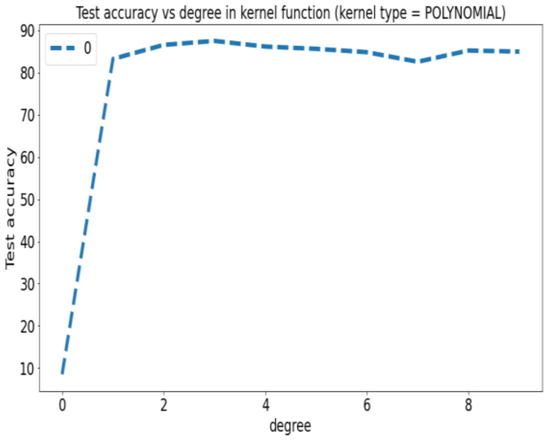
Conclusions for hyperparameter -s 0 -c 4 -t 2 -r 1 -d 4 -g 0.032

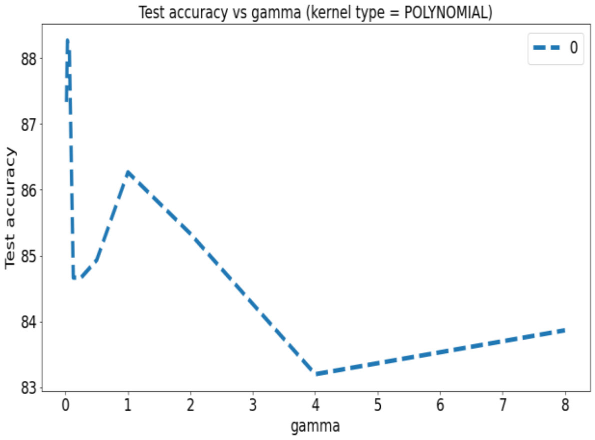
1) Kernel Type = Radial Basis Function
2) Degree in kernel function = 4, Gamma value = 0.5 (for POLYNOMIAL type)
3) Cost value = 4
4) Gamma Value = 0.032 (for RBF)

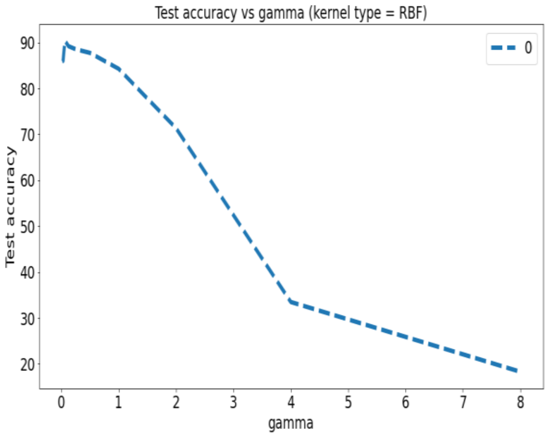
We now consider two pairs of 10 feature sets independently and obtain optimal hyperparameters for the same.

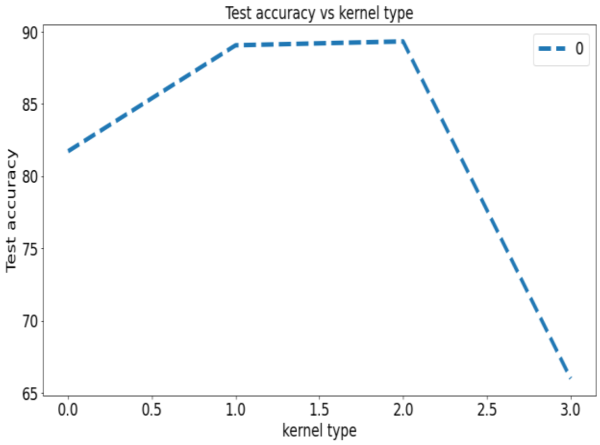
1) Considering features = F1, F2, F3 … F10

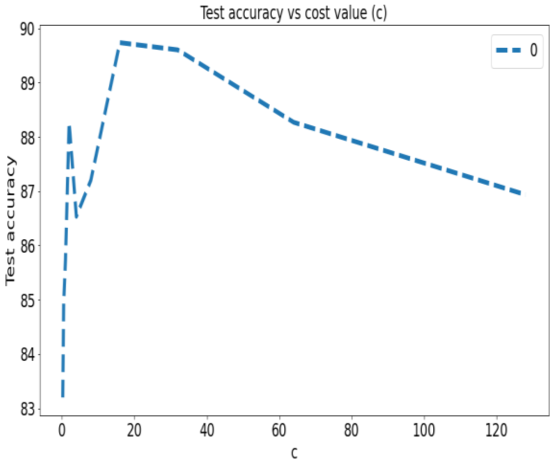
```
-s 0 -c 32 -t 2 -r 1 -d 2 -g 0.032
```



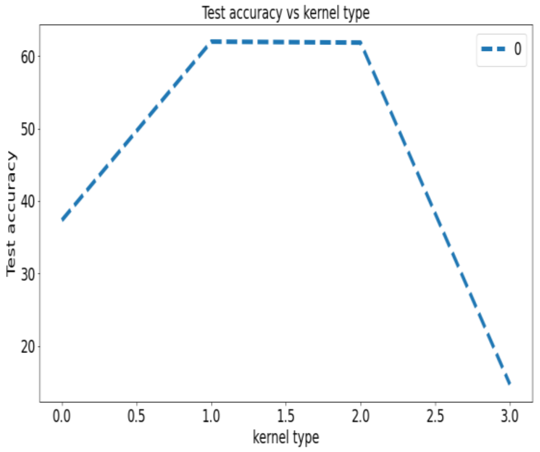


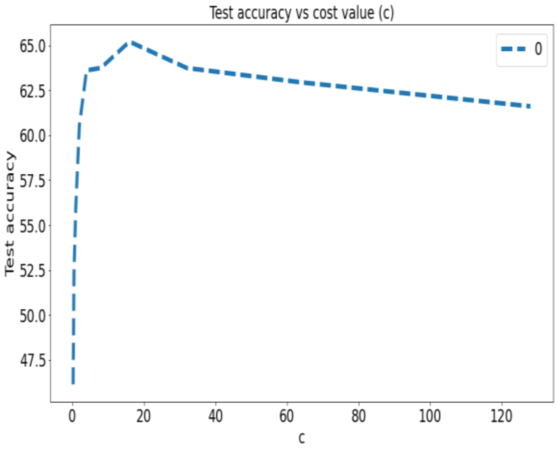


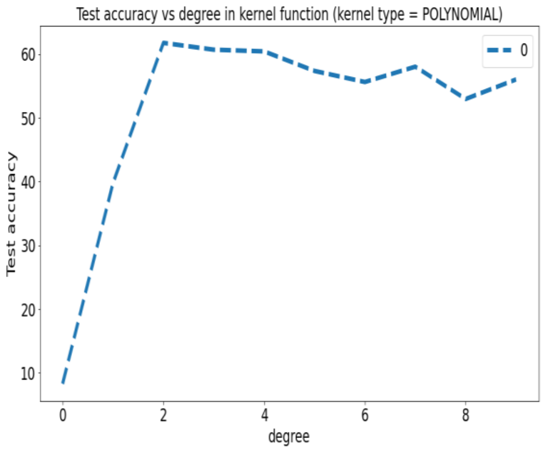


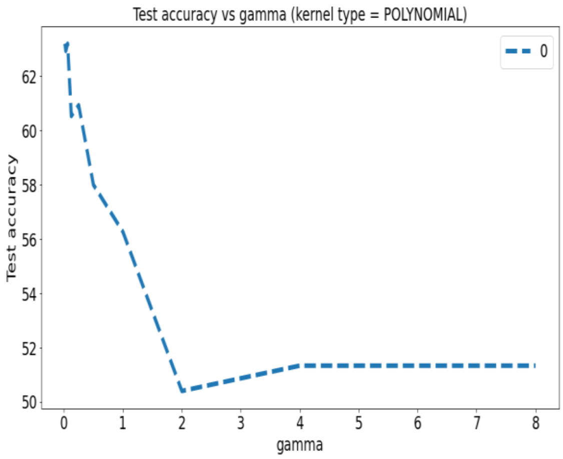


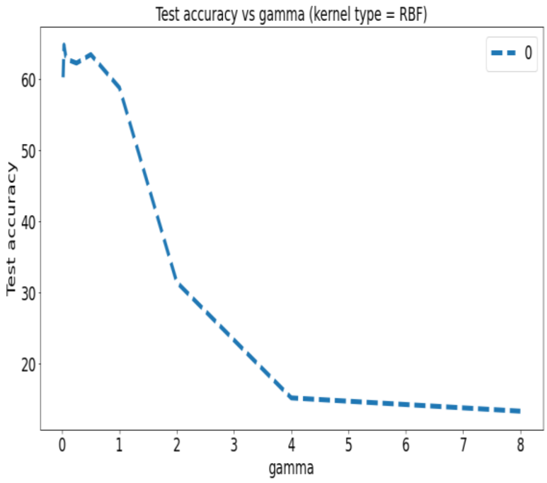
2) Considering features = F11, F12, F13 … F20

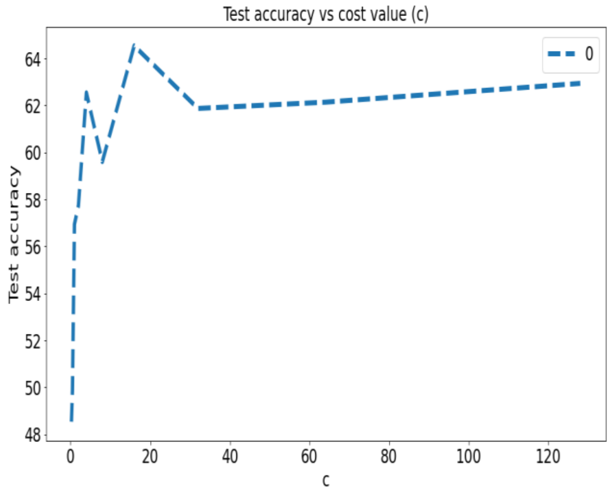












Results:

We notice that optimal hyperparameter settings change considerably with lesser number of features under consideration. Variation is noticed only in a few hyperparameters including C value, whereas most others remain unchanged when different set of features (10 only) are considered. This might be because the features represent similar kind of information, e.g., pixel values which remain similar even if different set of limited pixels are considered.

An attempt to pictorial representation of the 25 features as 5x5 pixel values has been made as well, however did not yield fruitful results for most higher value of labels.

For the given dataset, the following conditions are conclusive -
Conditions for underfitting of data:
a. Linear kernel
b. Very high value of C
c. Very low values of gamma
Conditions for overfitting of data:
a. Polynomial kernel with high degree
b. Very low value of C
c. Very high values of gamma

PART 2:

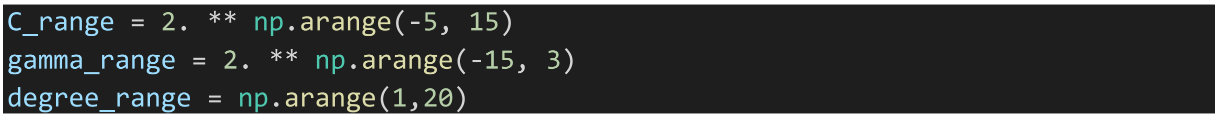
Result with linear kernel:-

Training accuracy : 0.5513333333333333

Testing accuracy: 0.5265

So linear kernel is underfitting or unable to learn the model accurately no matter what are the values of c. This is because of large dimensions of data.

The below are the examined C, gamma and degree (for poly kernel) ranges.



```
C_range = 2. ** np.arange(-5, 15)
gamma_range = 2. ** np.arange(-15, 3)
degree_range = np.arange(1,20)
```

Result with poly kernel:-

C=0.03125, gamma=2.4414062e-05, training accuracy=0.805, testing accuracy = 0.79, degree=8

Result with rbf kernel:-

C=32.0, gamma=0.0625, training accuracy=1, testing accuracy = 0.967