ELL409 Assignment 2 Report

Aadarsh Gupta, Entry No. – 2019EE10451

Abstract

This report consists of analysis of 2 Parts – Part1 (1A, 1B) and Part2 for respective problems.

1 PART 1

1.1 PART 1A.

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 : 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

Note. 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 : $[2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}]$

Plot Type 1.

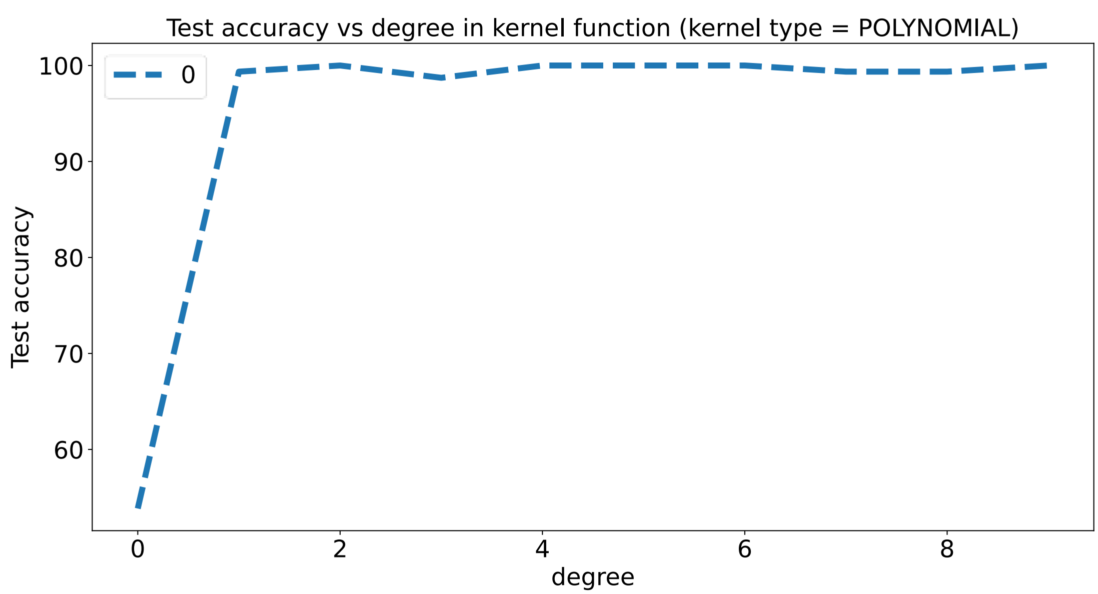


Fig1. Binary Classification for Classes : 0, 1

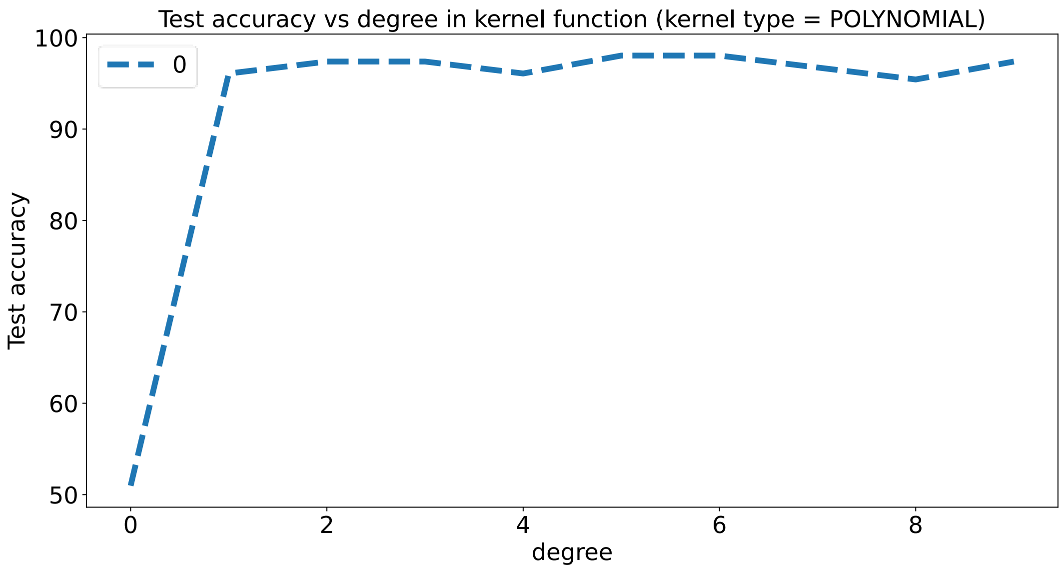


Fig2. Binary Classification for Classes : 2, 3

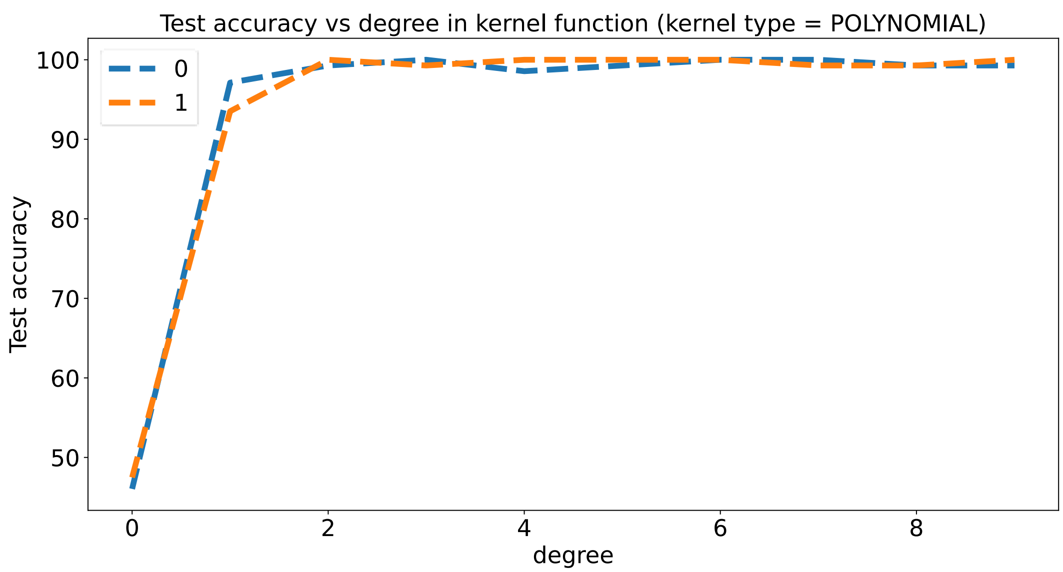


Fig3. Binary Classification for Classes : 4, 5 (2 instances)

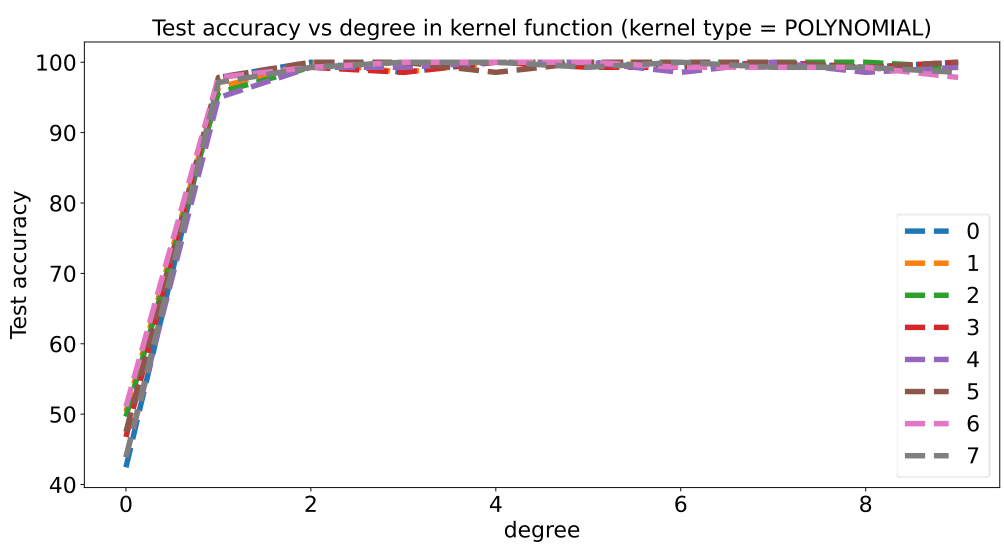


Fig4. Binary Classification for Classes : 4, 5 (8 instances)

Plot Type 2.

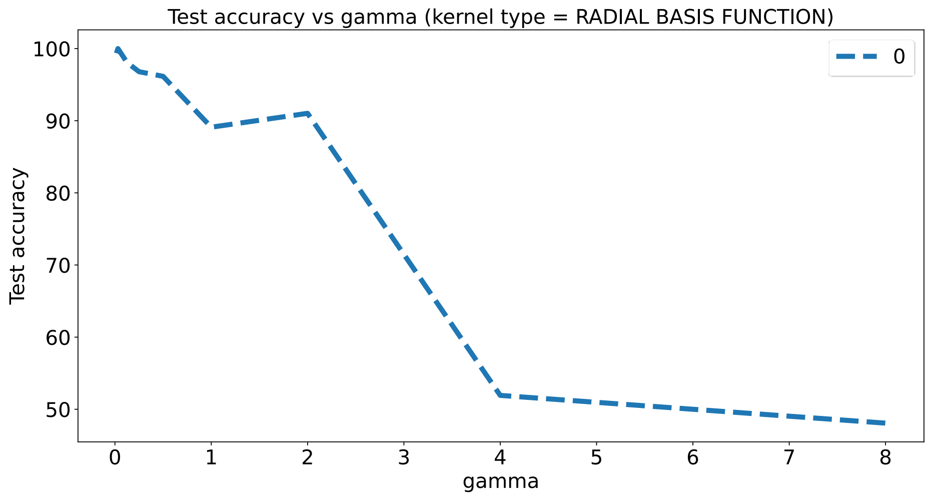


Fig5. Binary Classification for Classes : 0, 1

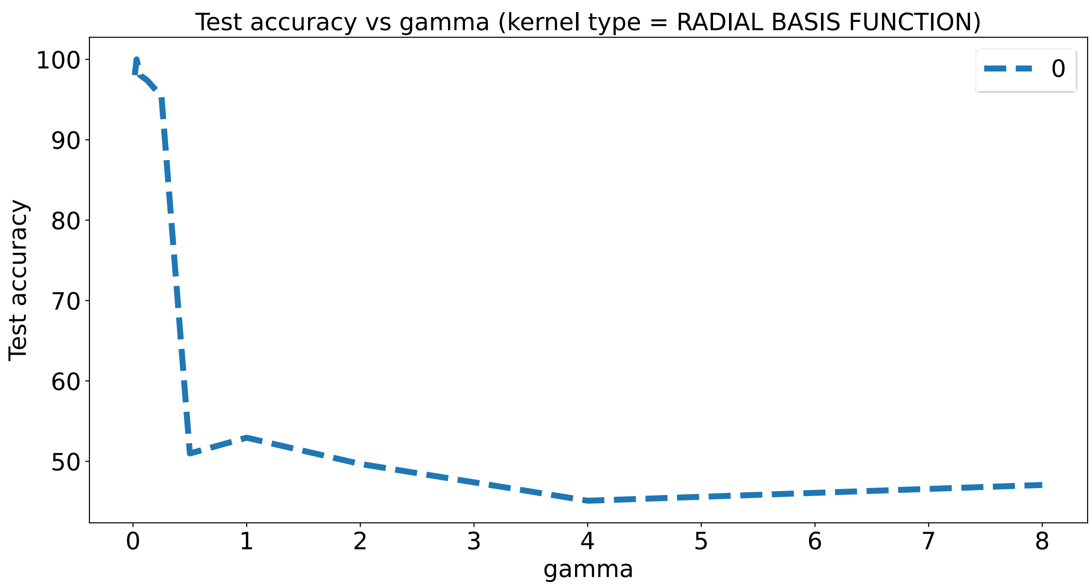


Fig5. Binary Classification for Classes : 2, 3

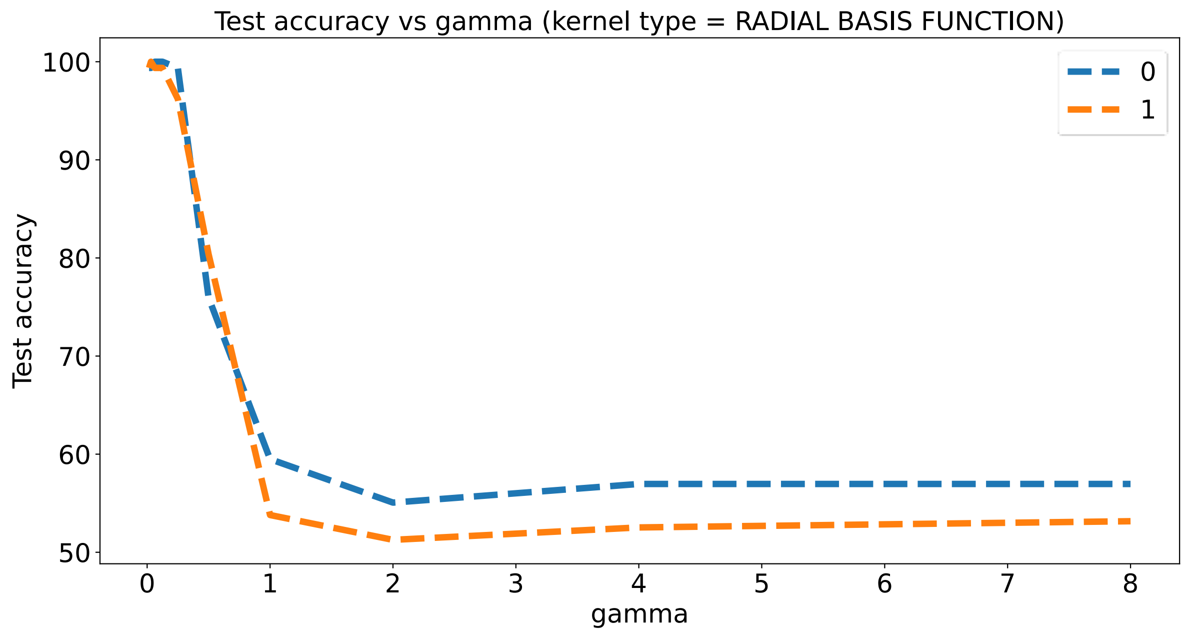


Fig7. Binary Classification for Classes : 6, 7 (2 instances)

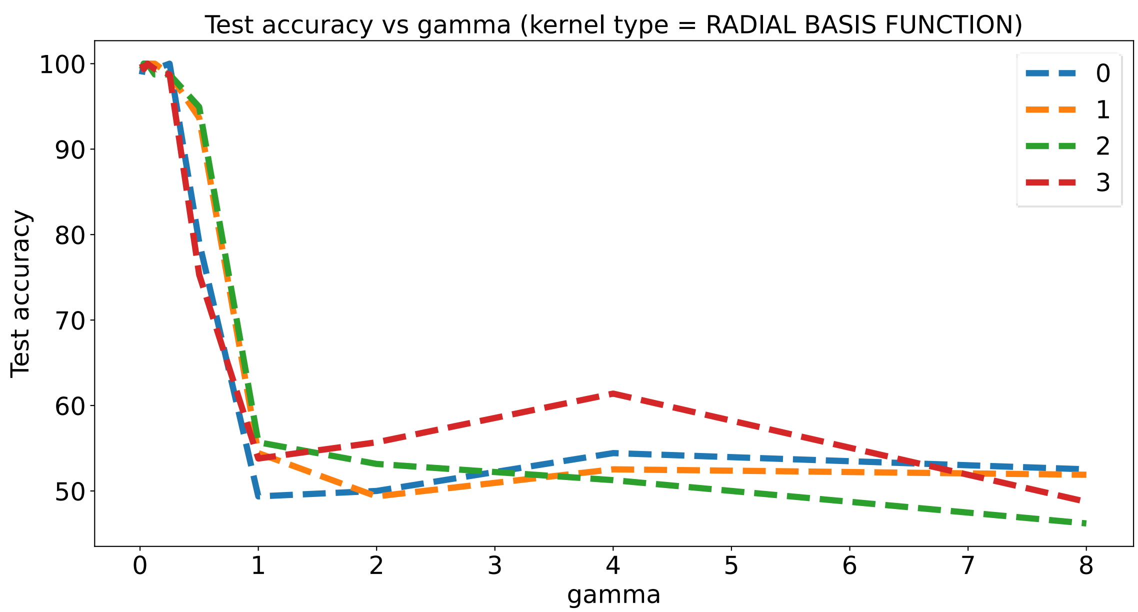


Fig8. Binary Classification for Classes : 6, 7 (4 instances)

Plot Type 3.

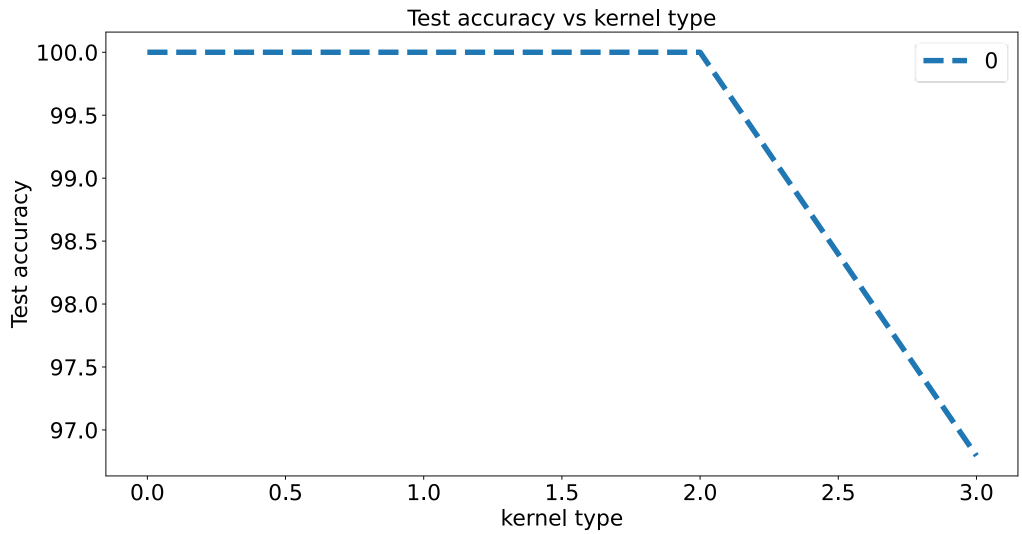


Fig9. Binary Classification for Classes : 0, 1

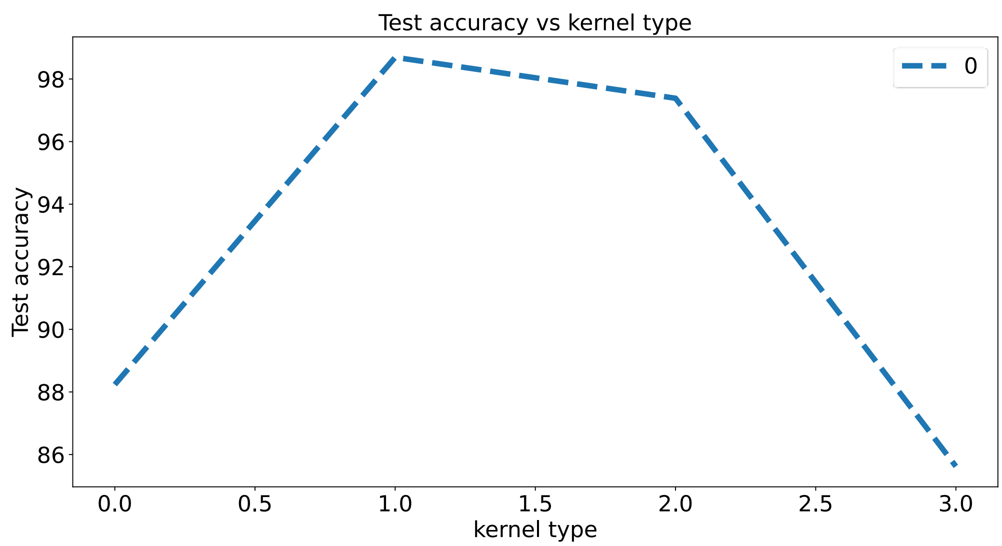


Fig10. Binary Classification for Classes : 2, 3

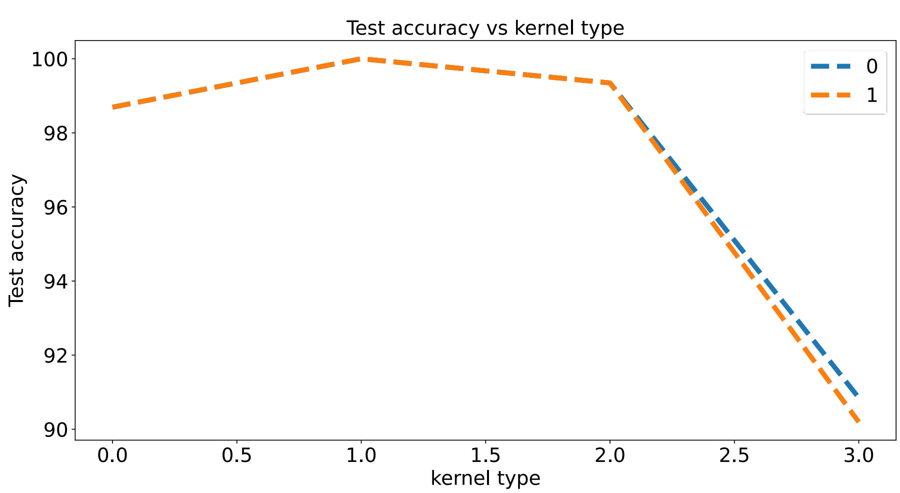


Fig11. Binary Classification for Classes : 7, 8 (2 instances)

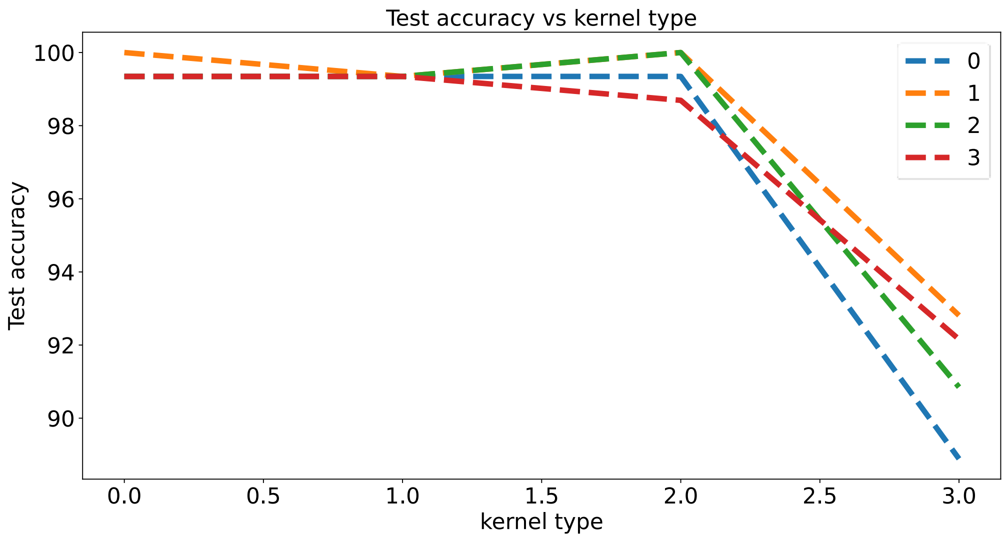


Fig12. Binary Classification for Classes : 7, 8 (4 instances)

Plot Type 4.

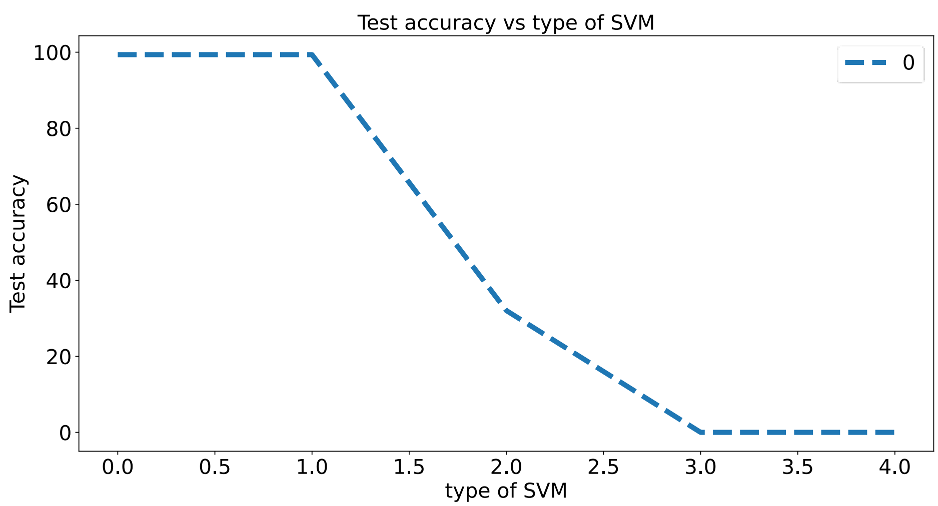


Fig13. Binary Classification for Classes : 0, 1

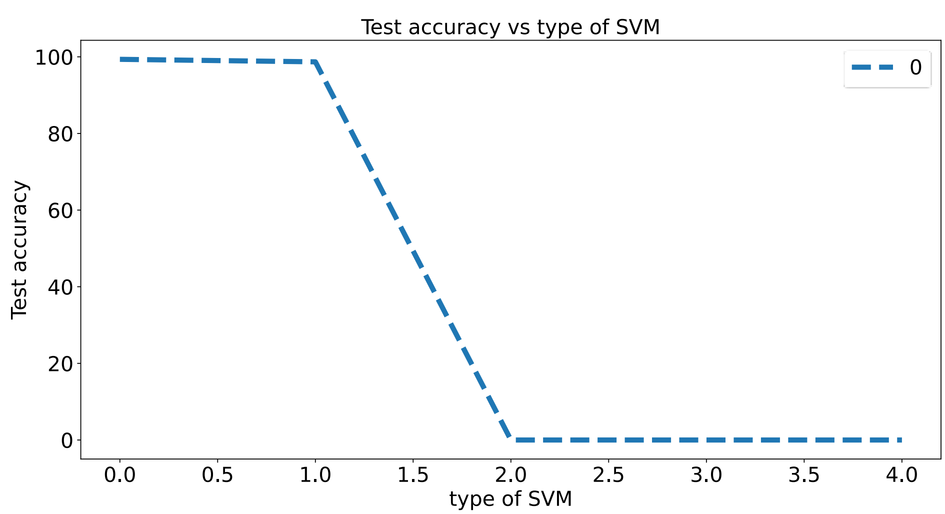


Fig14. Binary Classification for Classes : 2, 3

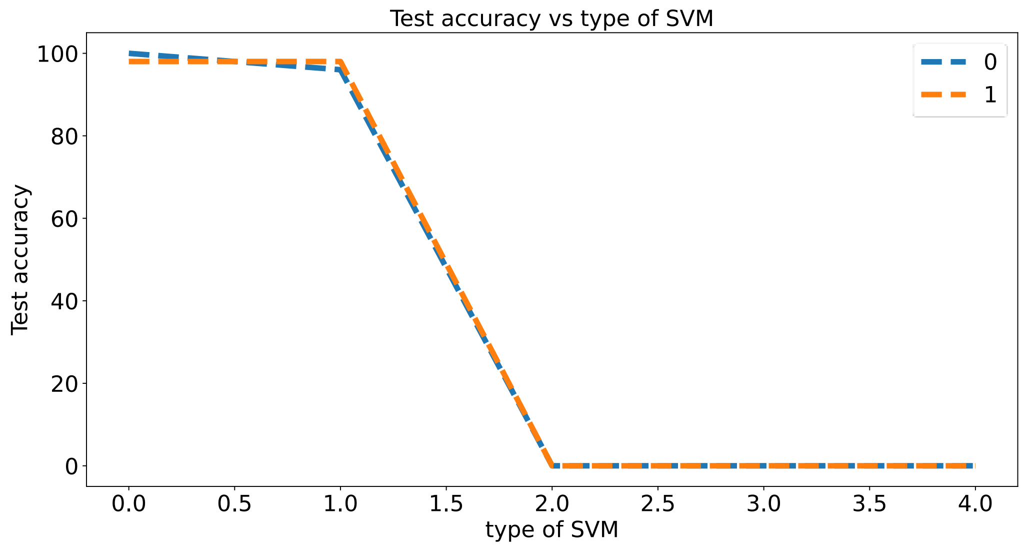


Fig15. Binary Classification for Classes : 0, 9 (2 instances)

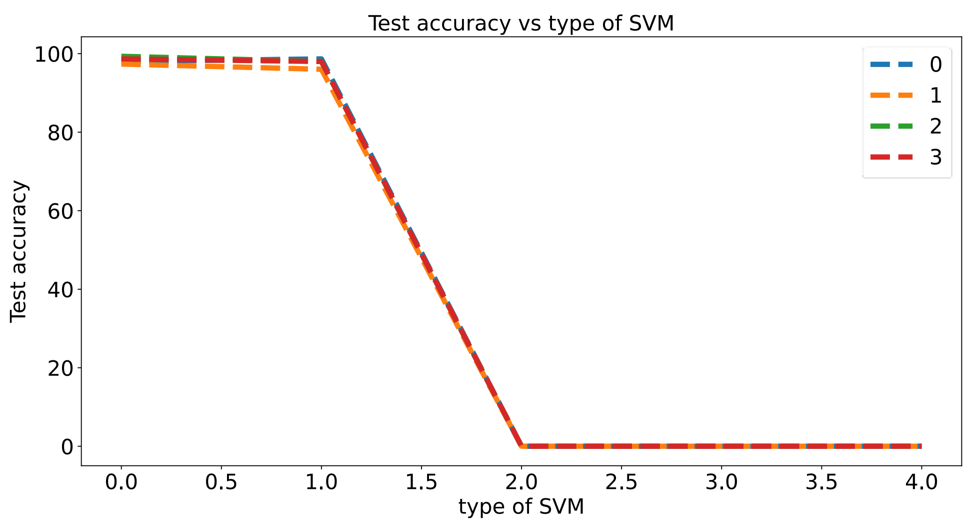


Fig16. Binary Classification for Classes : 0, 9 (4 instances)

Plot Type 5.

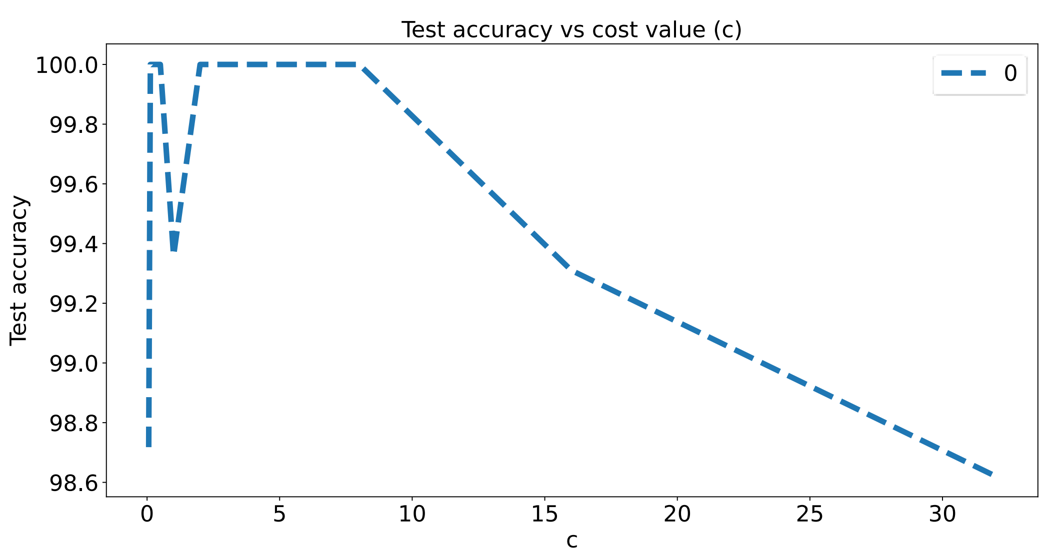


Fig17. Binary Classification for Classes : 0, 1

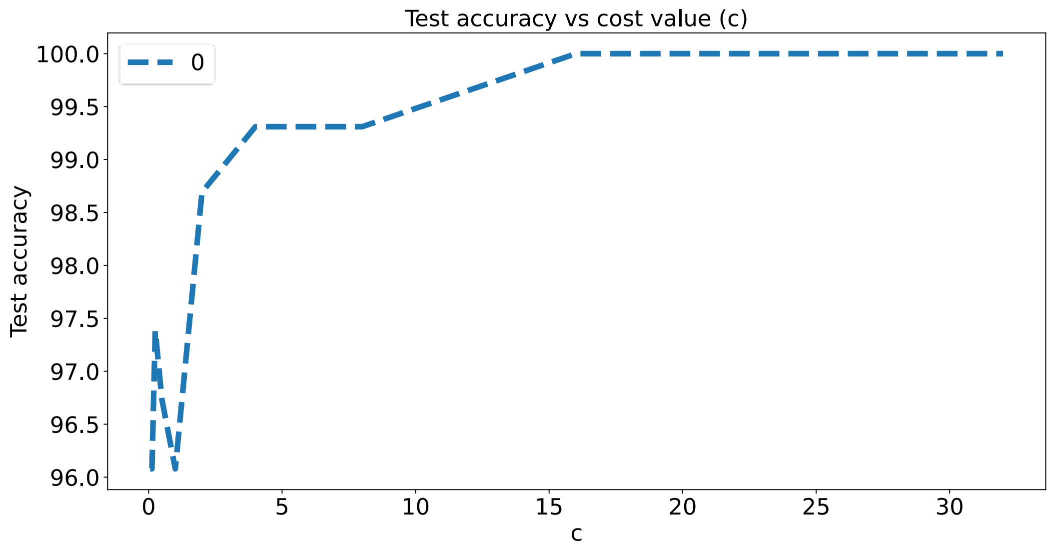


Fig18. Binary Classification for Classes : 2, 3

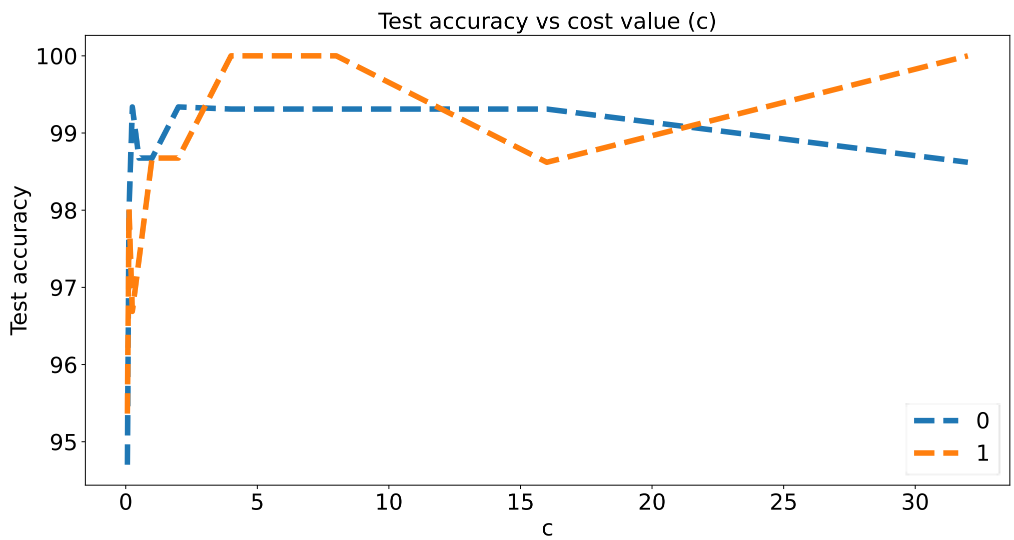


Fig19. Binary Classification for Classes : 1, 8 (2 instances)

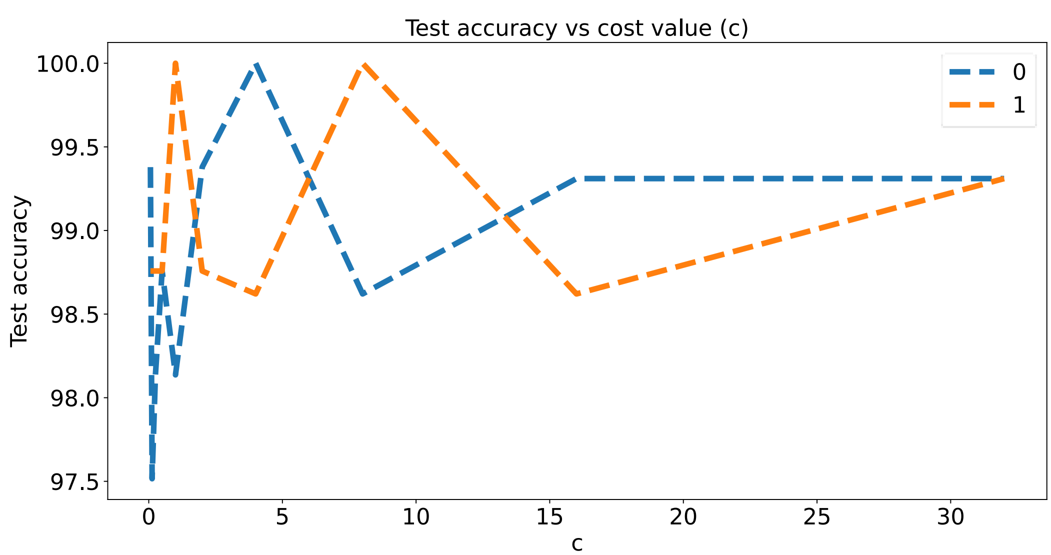


Fig20. Binary Classification for Classes : 2, 7 (2 instances)

Conclusions

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

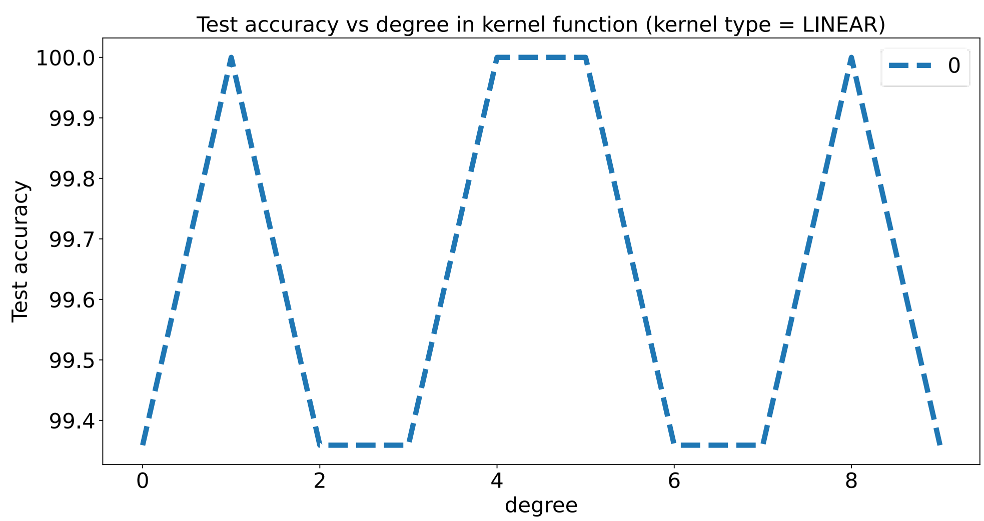
```
-s 0 -c 4 -t 2 -r 1 -d 3 -g 0.032
```

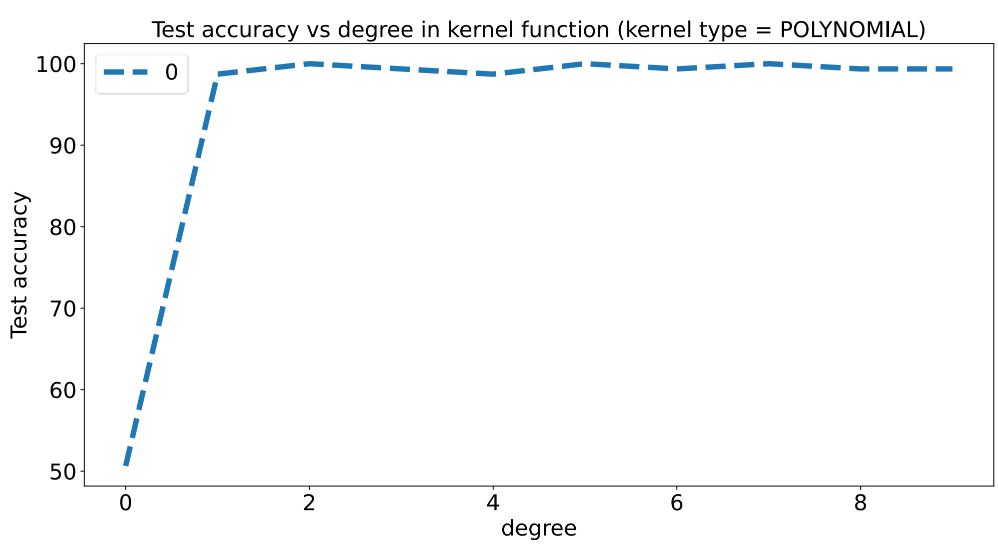
1) Kernel Type = Radial Basis Function
2) Degree in kernel function = 3 (for POLYNOMIAL type)
3) Cost value = 4
4) Gamma Value = 0.032 (for RBF)
5) SVM type = 0 (C-SVC)

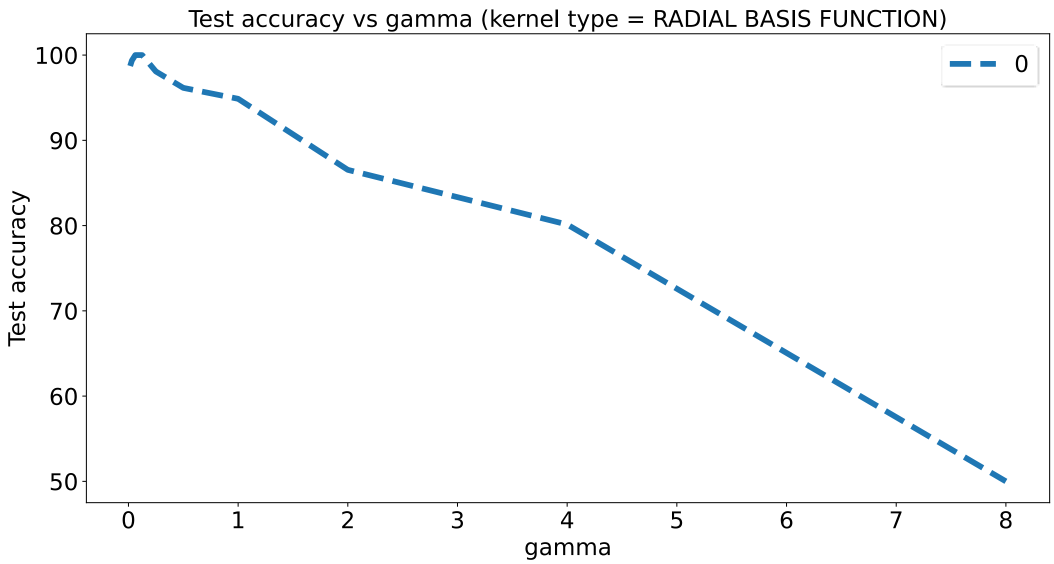
We now consider two pairs of 10 feature sets independently and obtain optimal hyperparameters for the same.

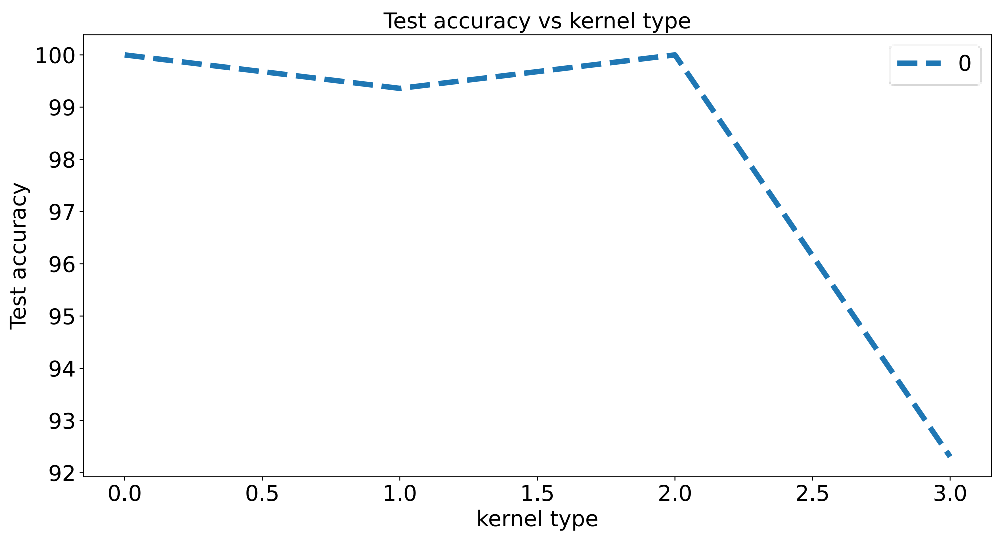
1) Considering features = F1, F2, F3 … F10
   a. class_labels = 0, 1

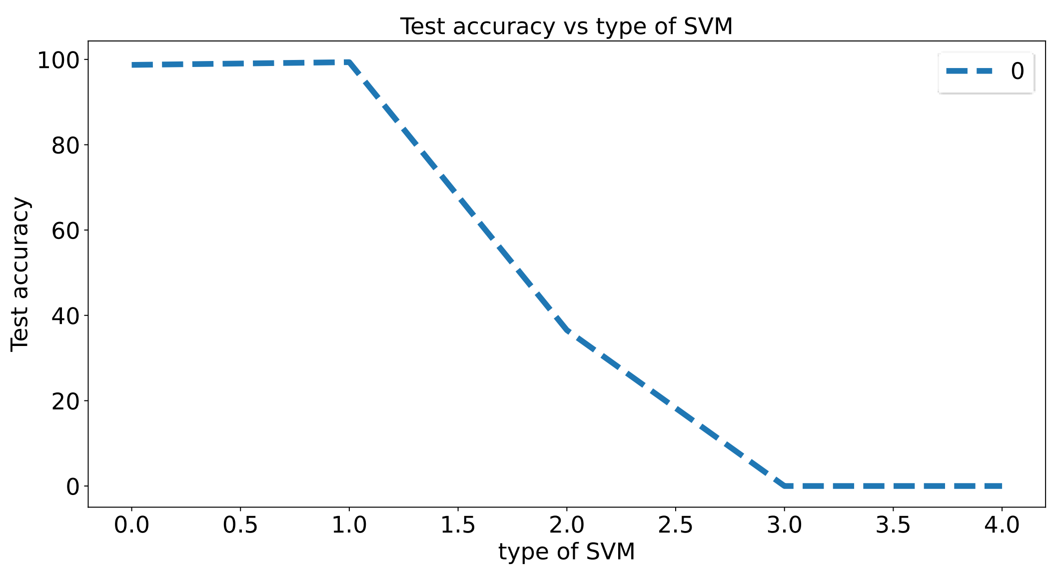
```
-s 0 -c 8 -t 2 -r 1 -d 3 -g 0.032
```

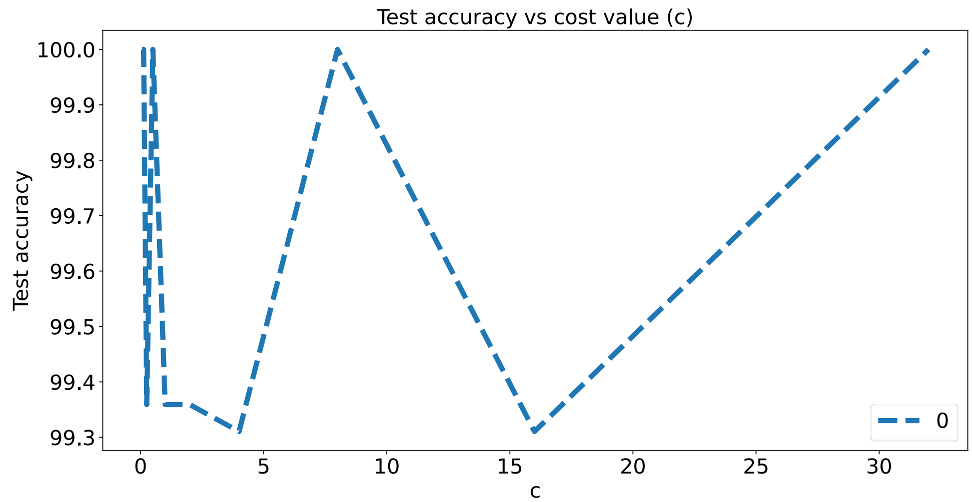





Test accuracy vs gamma (kernel type = RADIAL BASIS FUNCTION)
Test accuracy
gamma
0
100
90
80
70
60
50
0
1
2
3
4
5
6
7
8

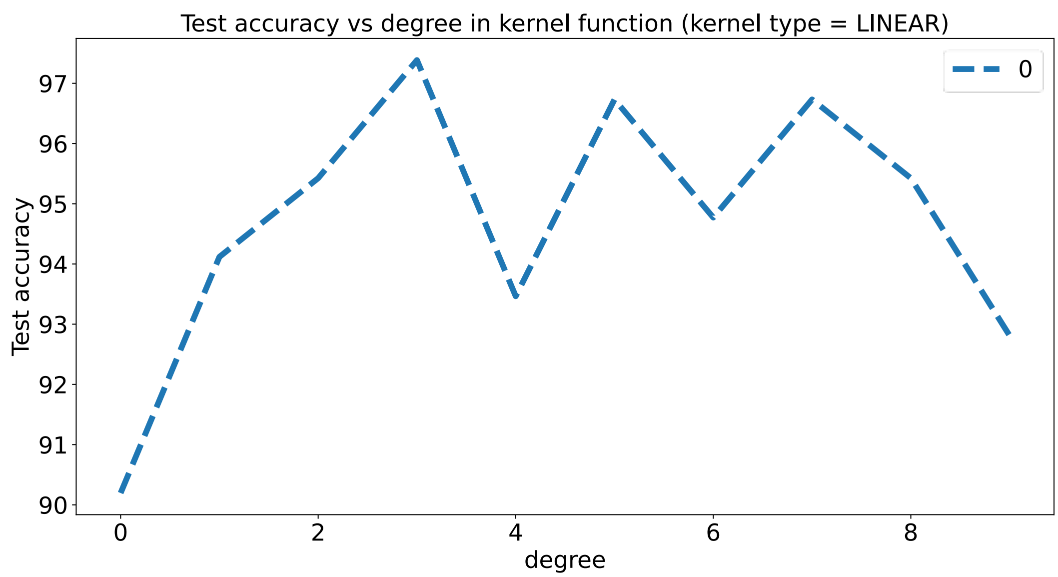

Test accuracy vs kernel type
Test accuracy
kernel type
0
100
99
98
97
96
95
94
93
92
0.0
0.5
1.0
1.5
2.0
2.5
3.0

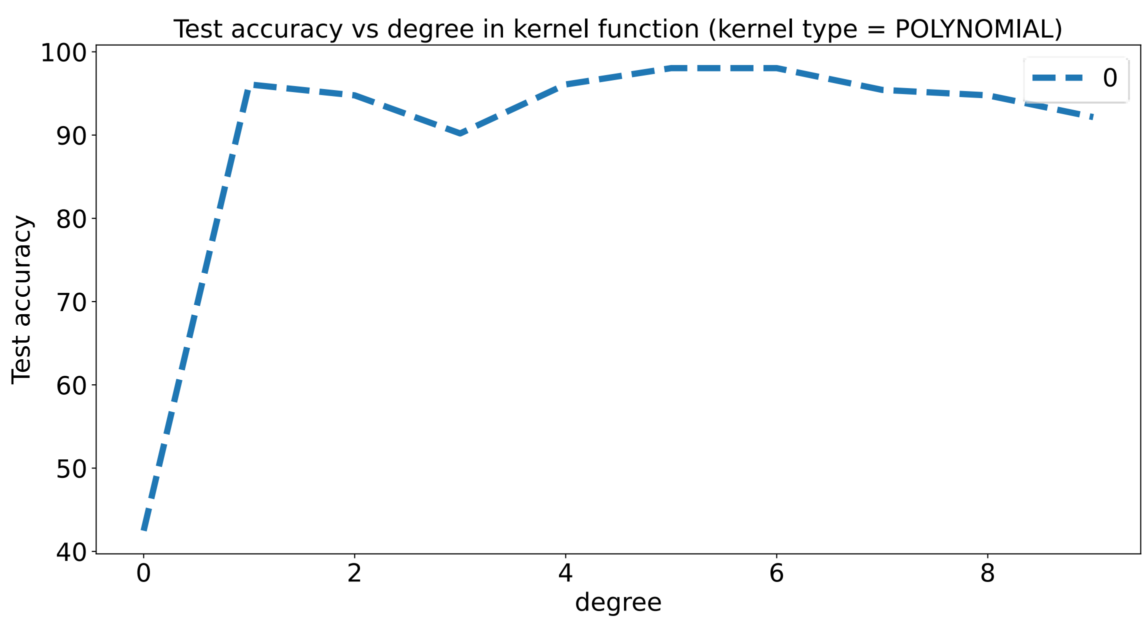

Test accuracy vs type of SVM
Test accuracy
type of SVM
0
100
80
60
40
20
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

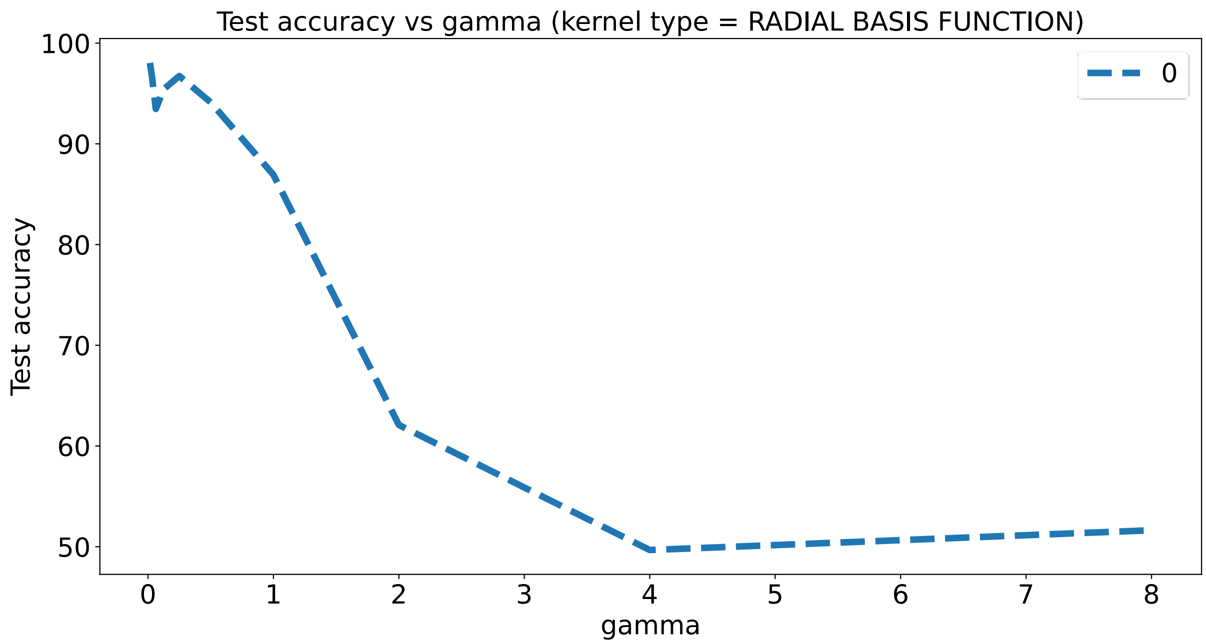


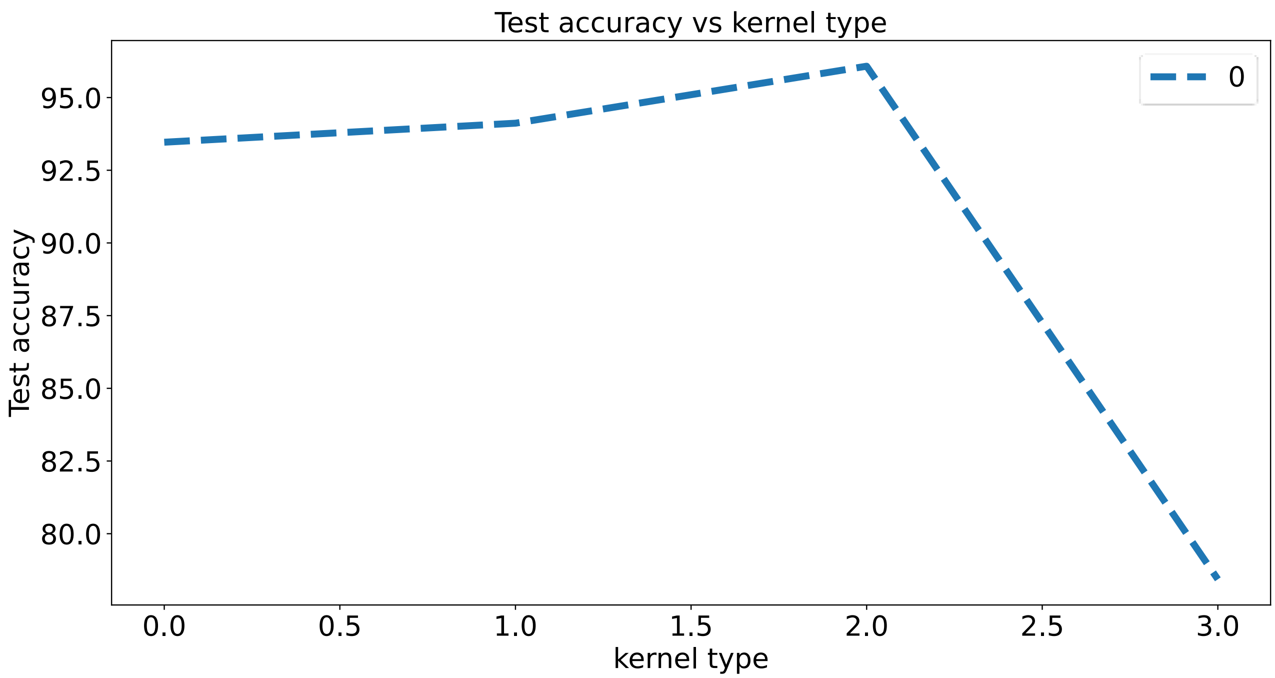
b. class_labels = 2, 3

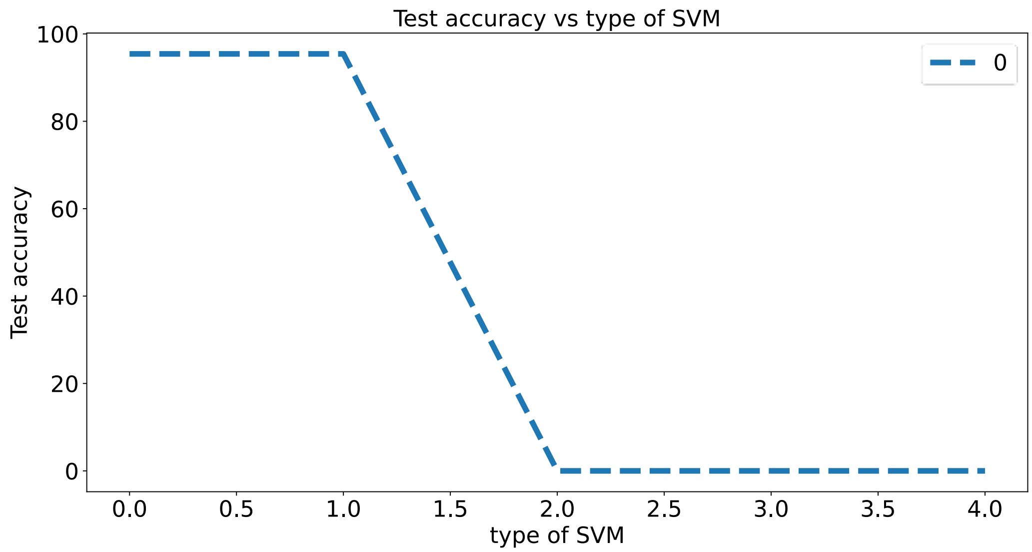
-s 0 -c 4 -t 2 -r 1 -d 3 -g 0.063

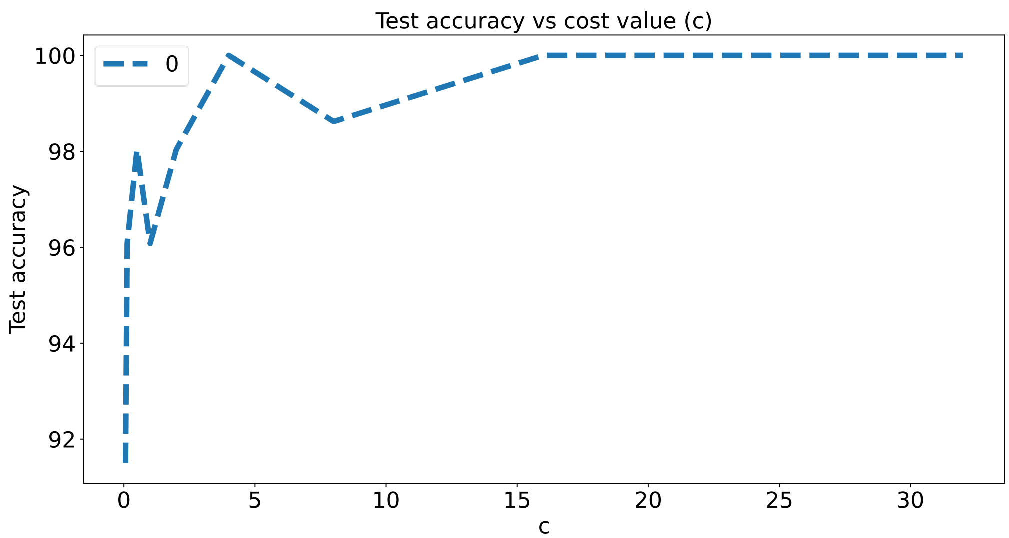





Test accuracy vs gamma (kernel type = RADIAL BASIS FUNCTION)
Test accuracy
gamma
0
100
90
80
70
60
50
0
1
2
3
4
5
6
7
8


Test accuracy vs kernel type
Test accuracy
kernel type
0
95.0
92.5
90.0
87.5
85.0
82.5
80.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

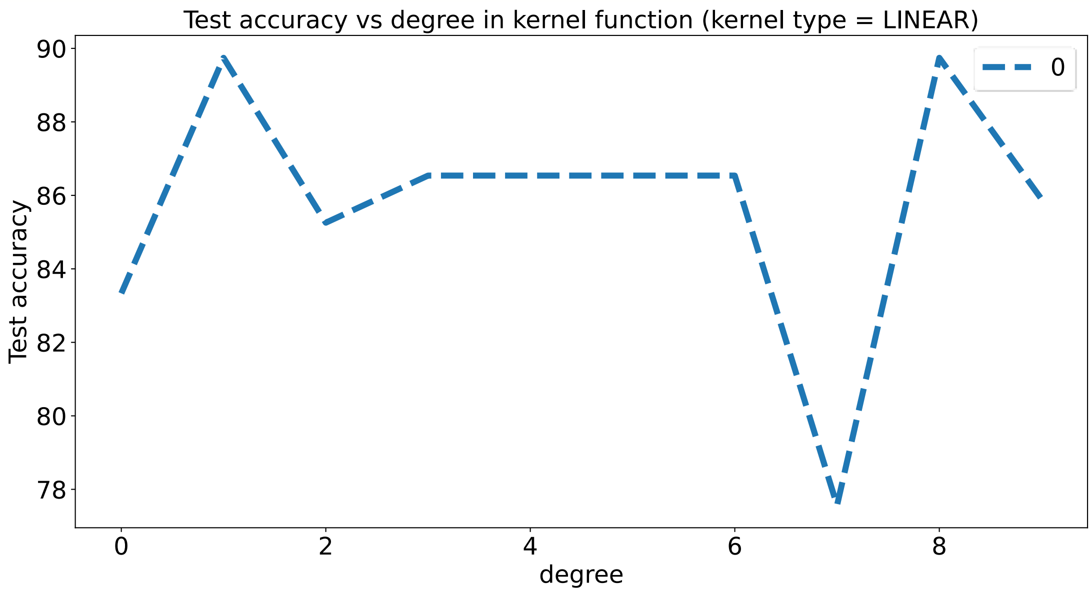


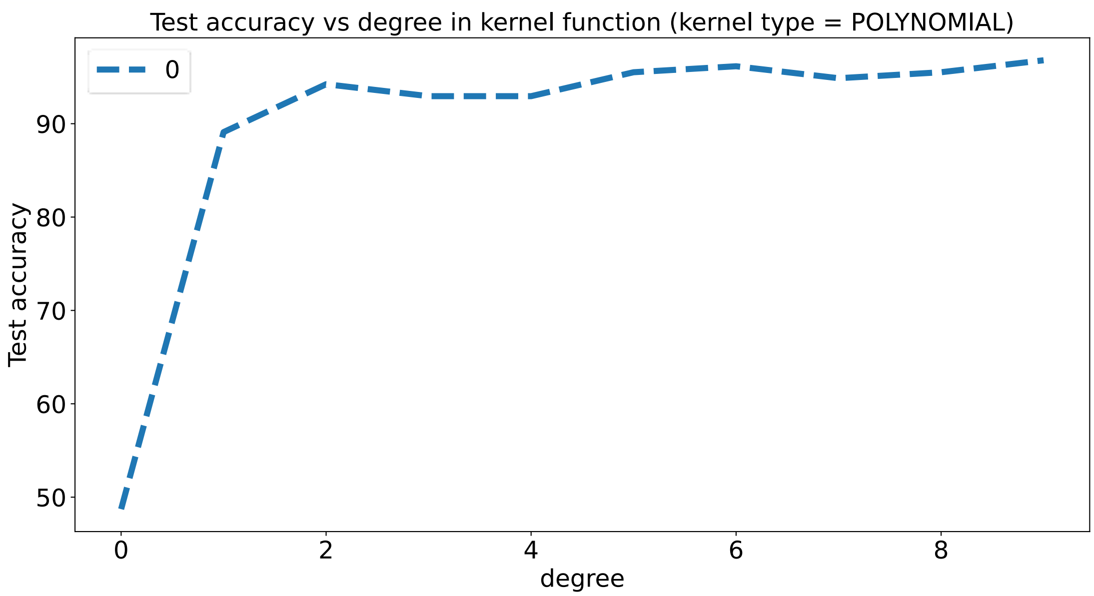


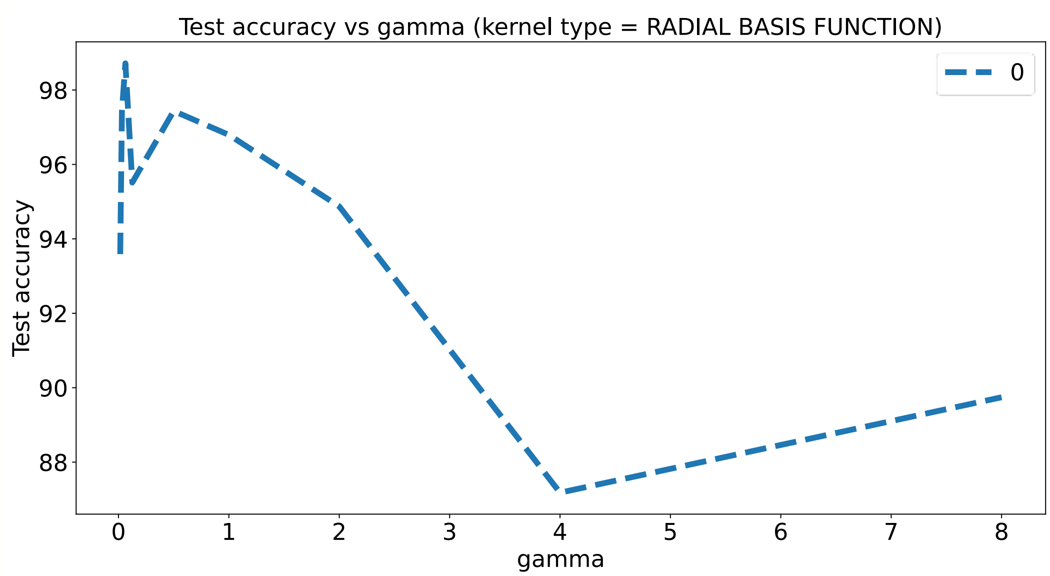
2) Consider features F11, F12, F13 … F20

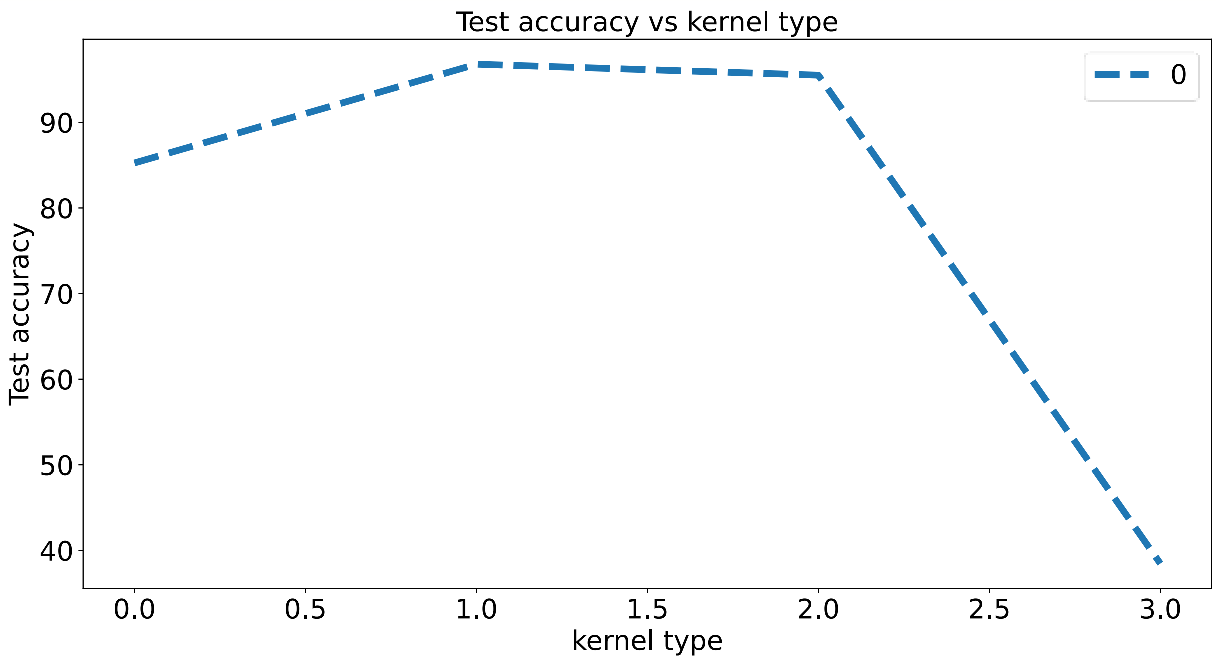
a) class_labels = 0, 1

```
-s 0 -c 0.25 -t 1 -r 1 -d 8 -g 0.032
```

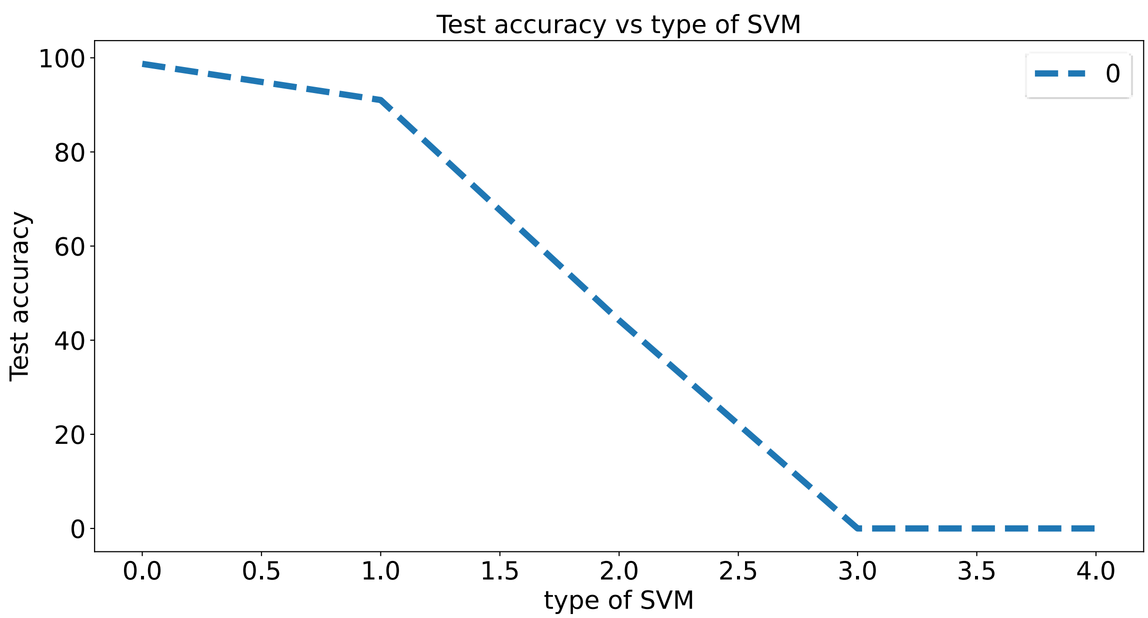

Test accuracy vs degree in kernel function (kernel type = LINEAR)
Test accuracy
degree
0


Test accuracy vs degree in kernel function (kernel type = POLYNOMIAL)
Test accuracy
degree
0

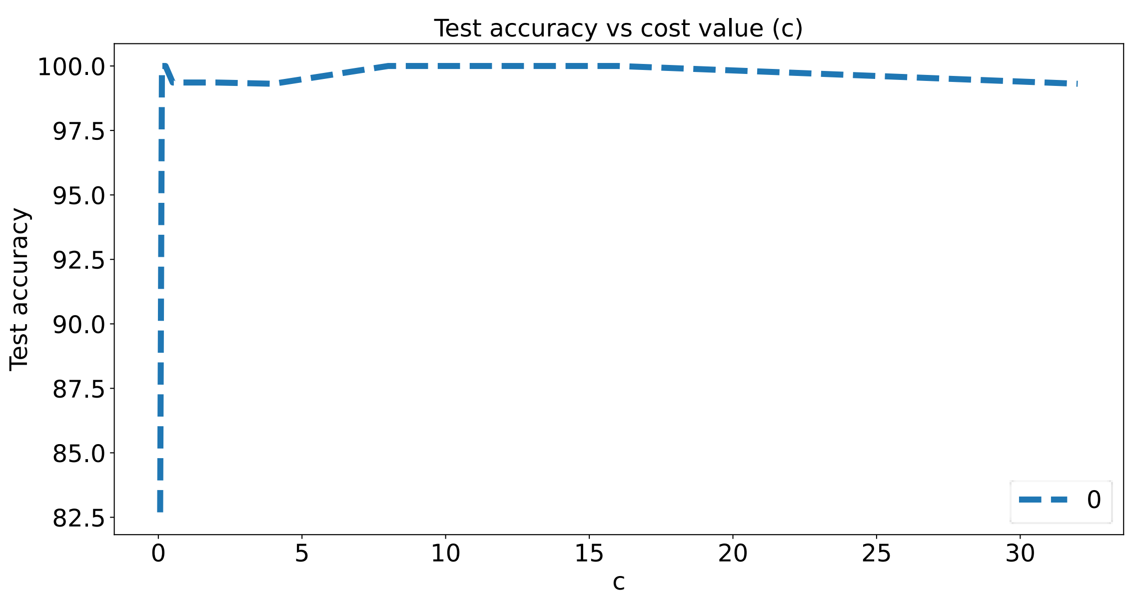

Test accuracy vs gamma (kernel type = RADIAL BASIS FUNCTION)
Test accuracy
gamma
0



Test accuracy vs kernel type
Test accuracy
kernel type
0



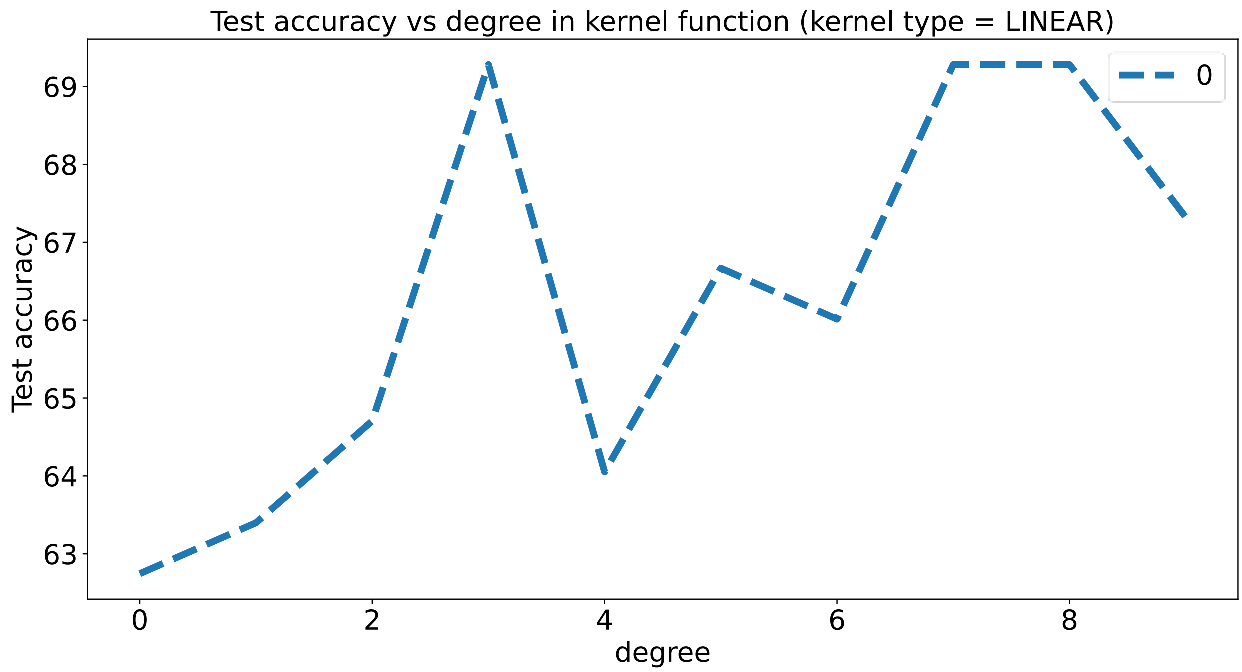
Test accuracy vs type of SVM
Test accuracy
type of SVM
0

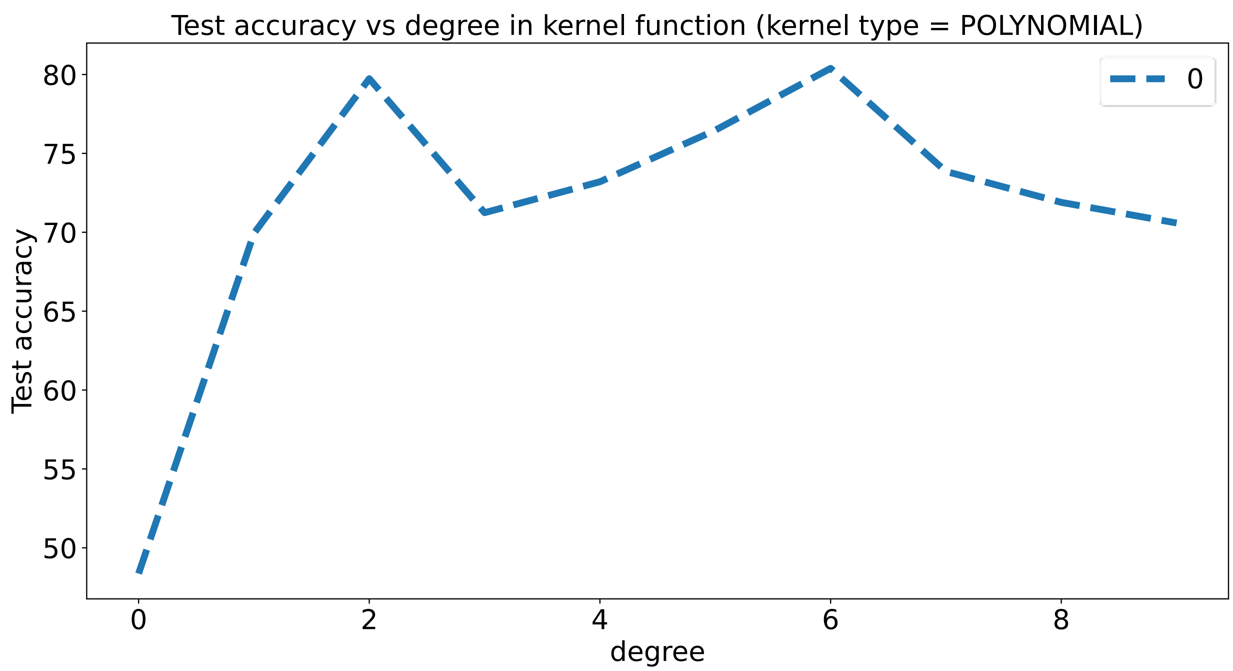


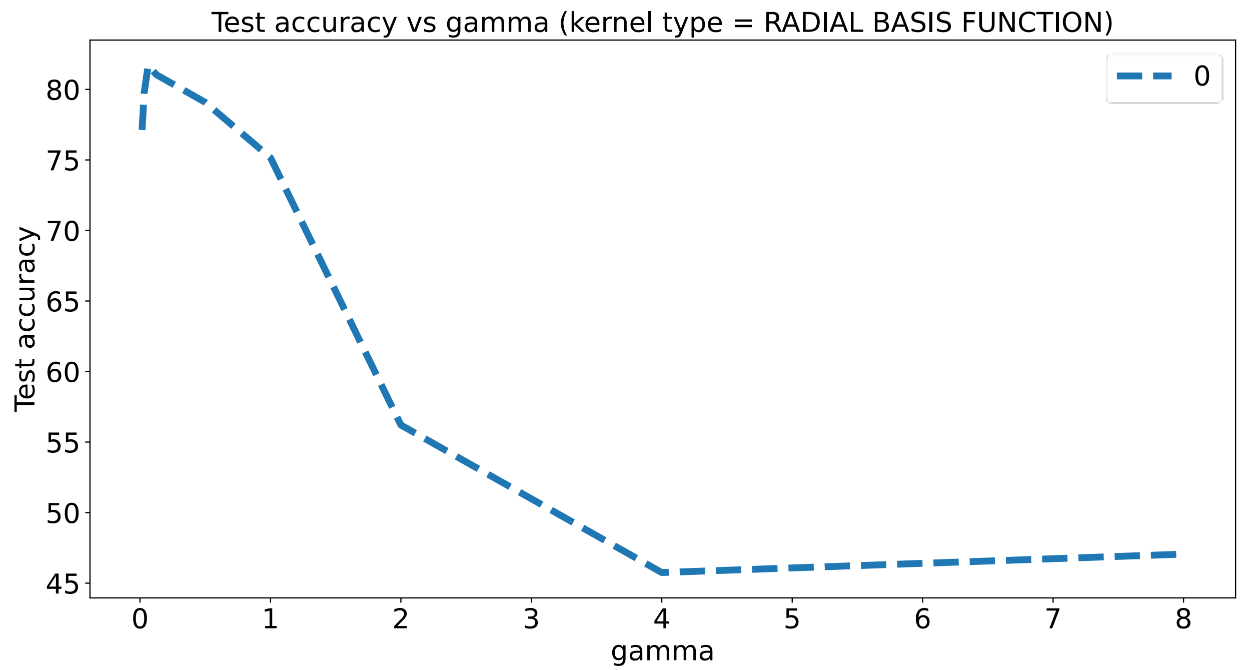
Test accuracy vs cost value (c)
Test accuracy
c
0

b) class_labels = 2, 3

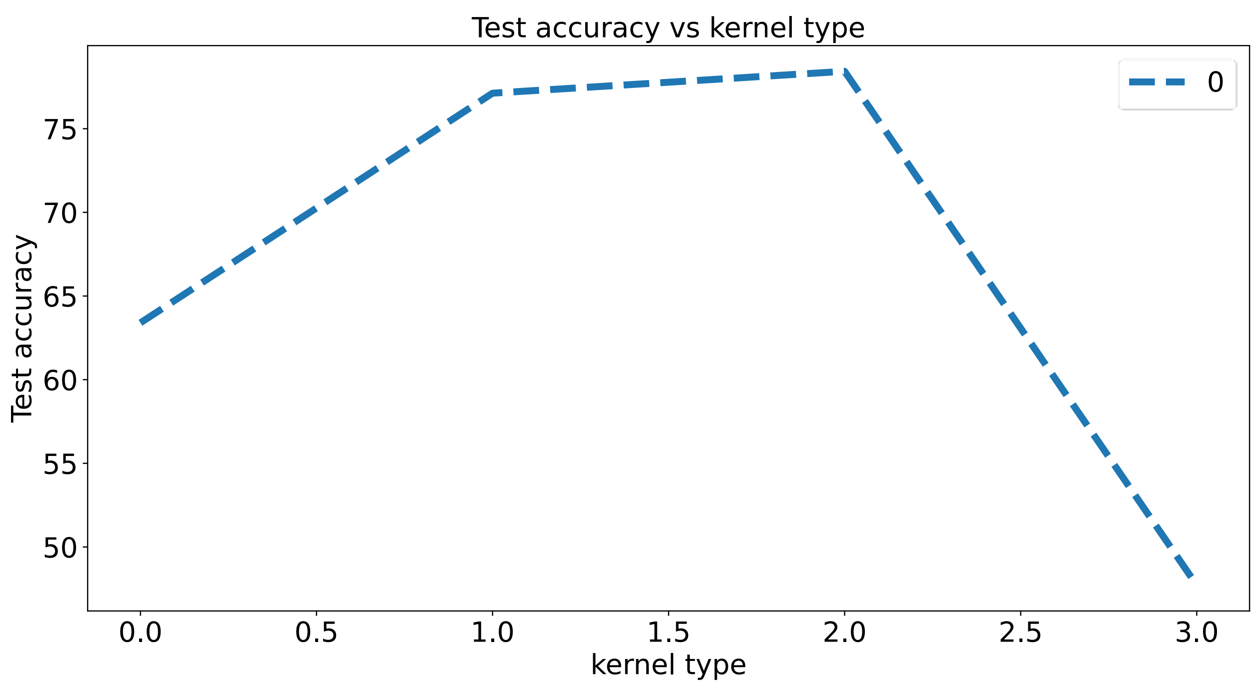
-s 1 -c 8 -t 2 -r 1 -d 3 -g 0.032



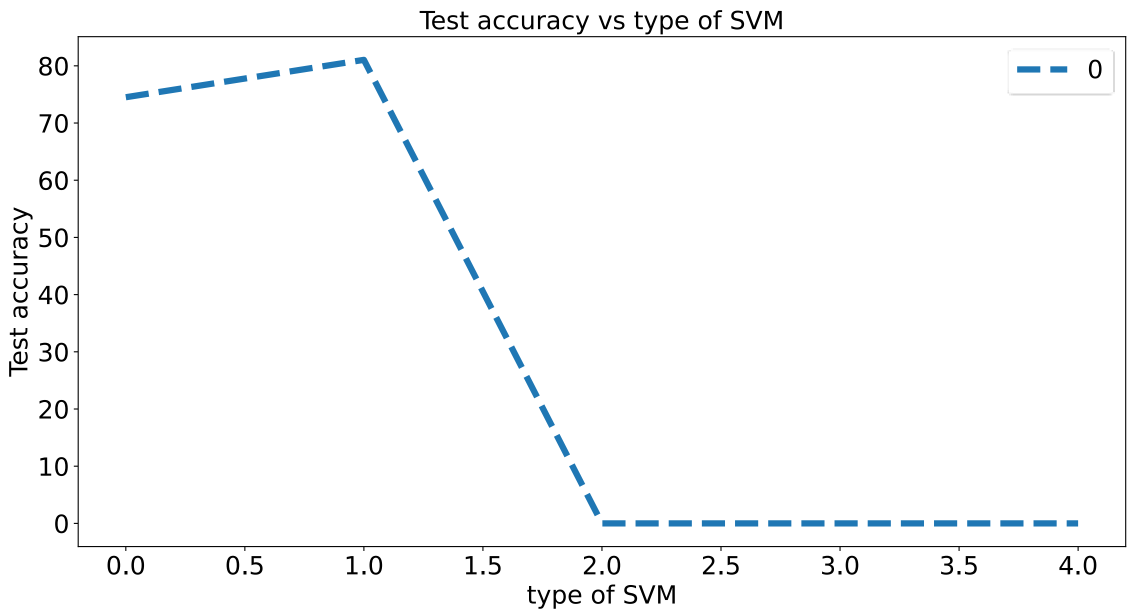


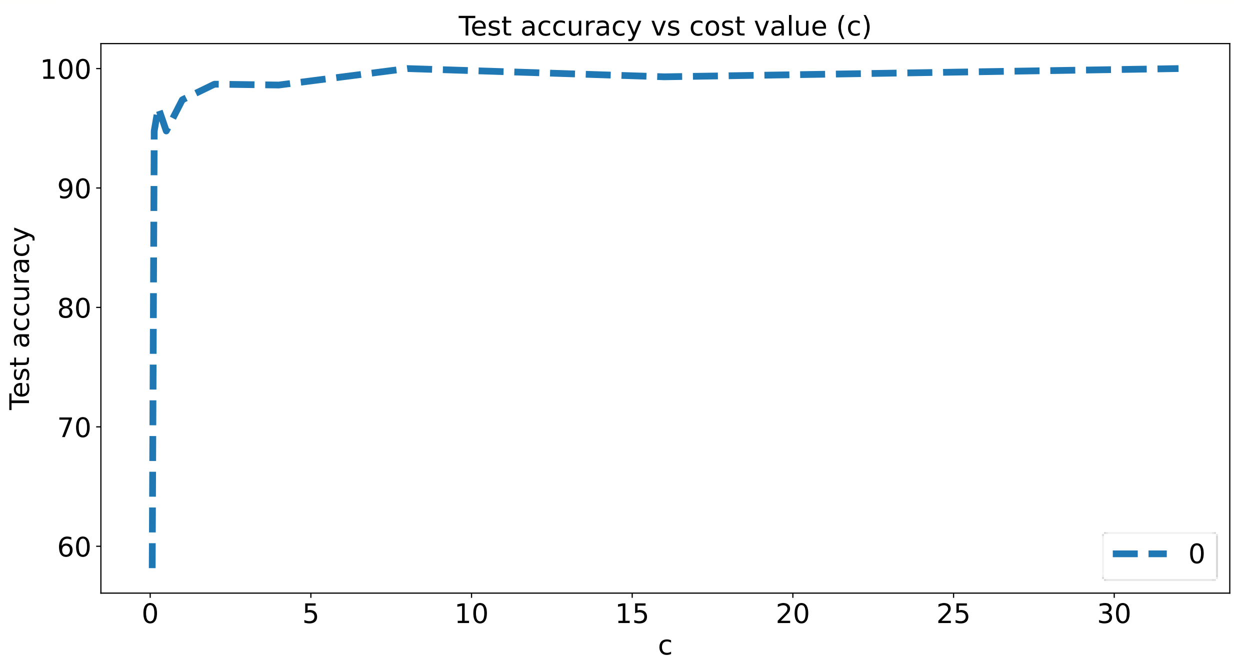


Test accuracy vs gamma (kernel type = RADIAL BASIS FUNCTION)
Test accuracy
gamma
0
45
50
55
60
65
70
75
80
0
1
2
3
4
5
6
7
8



Test accuracy vs kernel type
Test accuracy
kernel type
0
50
55
60
65
70
75
0.0
0.5
1.0
1.5
2.0
2.5
3.0





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 :

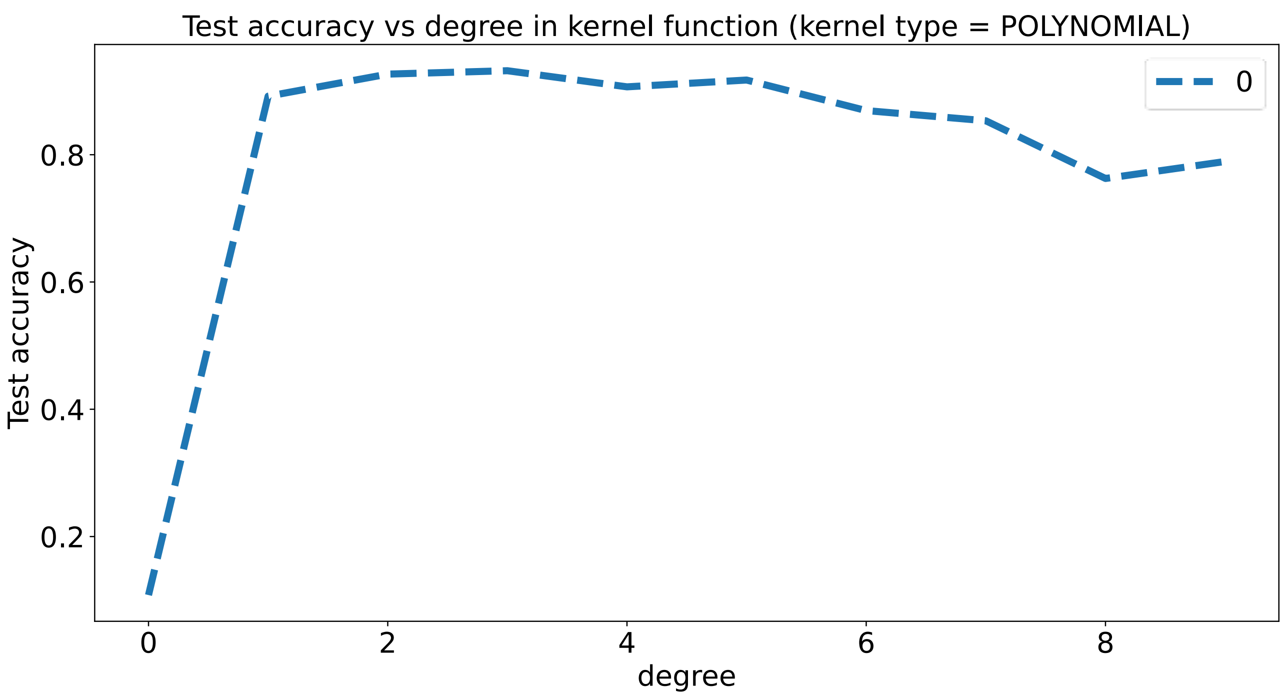
1) Test accuracy for ‘polynomial’ kernel VS degree in kernel function
2) Test accuracy for ‘RBF’ kernel VS gamma
3) Test accuracy VS type of kernel function
4) Test accuracy VS C-value

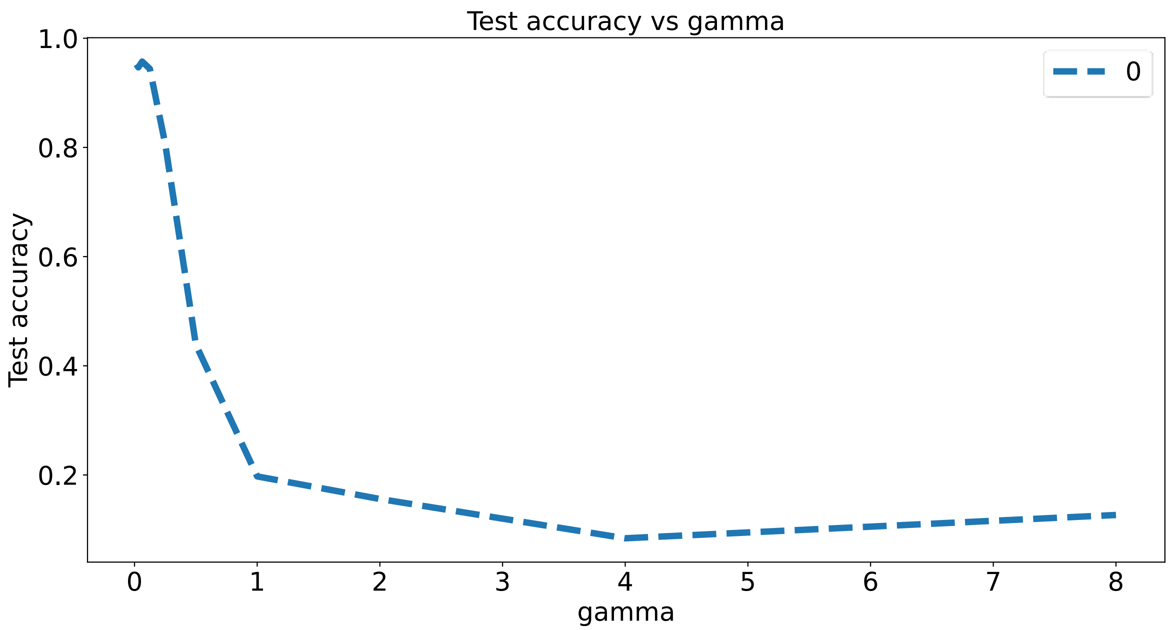
Note. The following steps are taken for each of the hyperparameter variations :

- 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 : [$2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}, 2^{7}$]

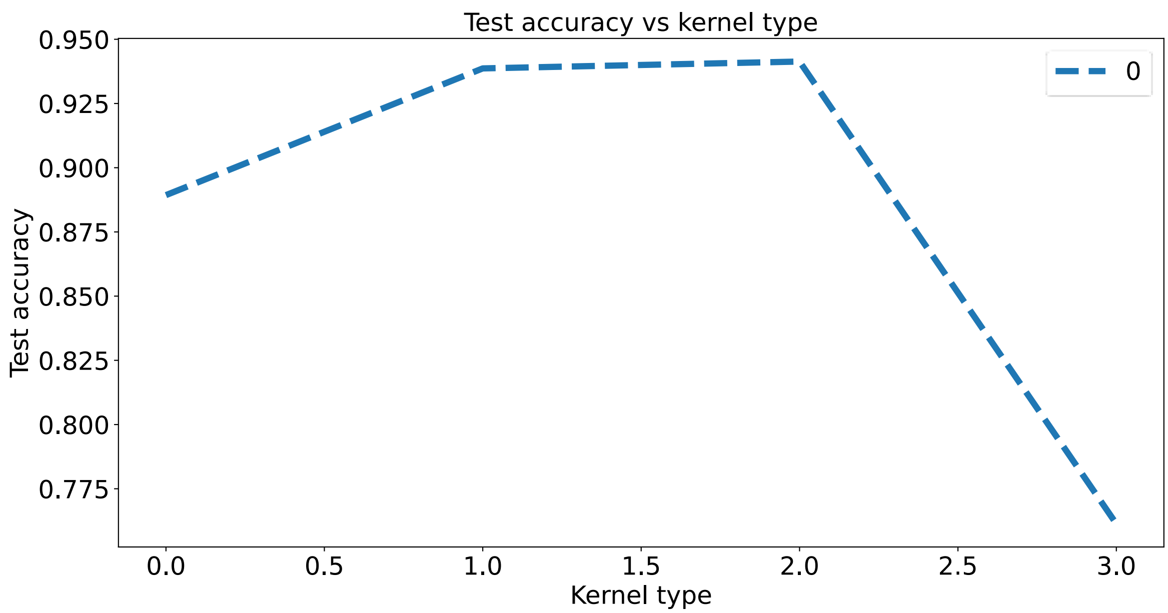
The 4 types of plots are as follows (for the following pair of class_labels) :

a) class_labels = 0, 1

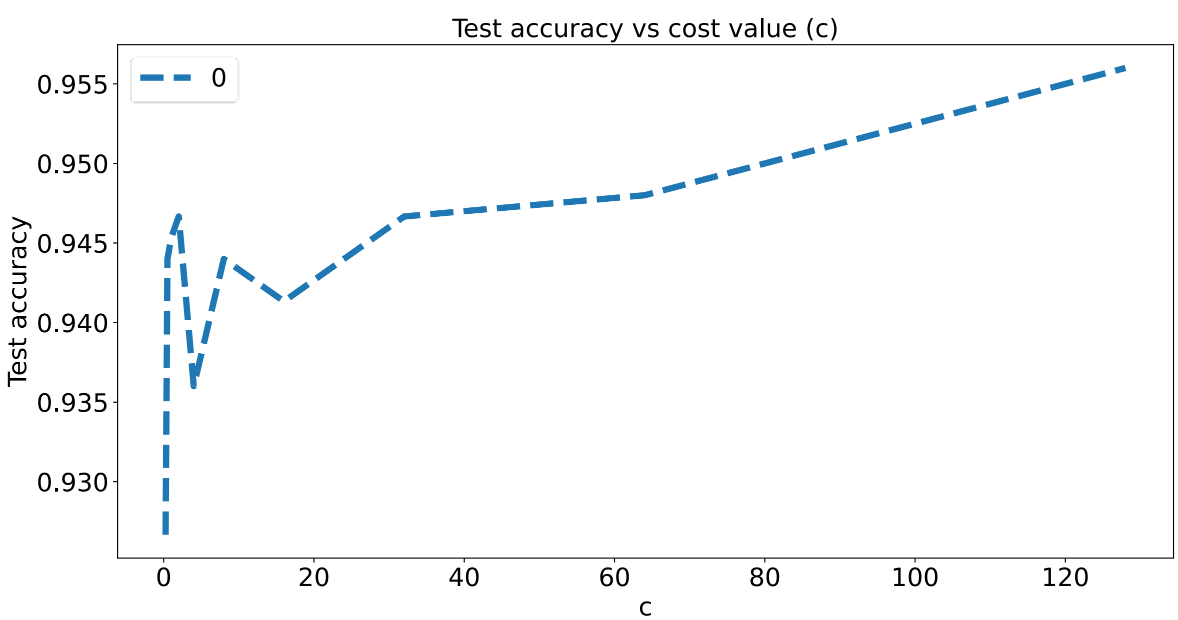




Test accuracy vs gamma
0
Test accuracy
gamma
1.0
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8

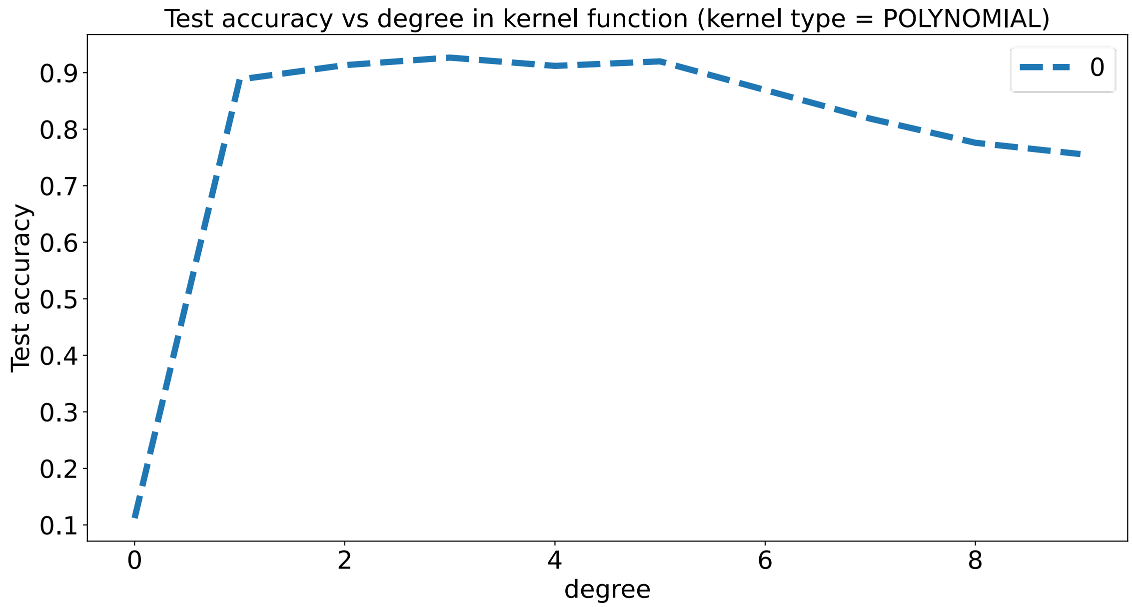


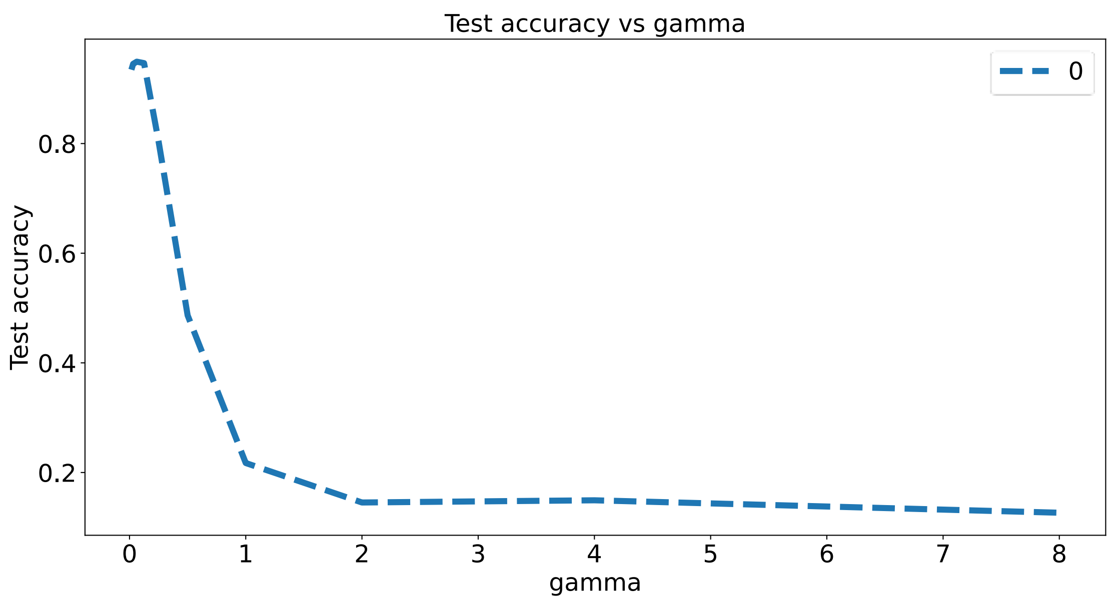
Test accuracy vs kernel type
0
Test accuracy
Kernel type
0.950
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.0
0.5
1.0
1.5
2.0
2.5
3.0

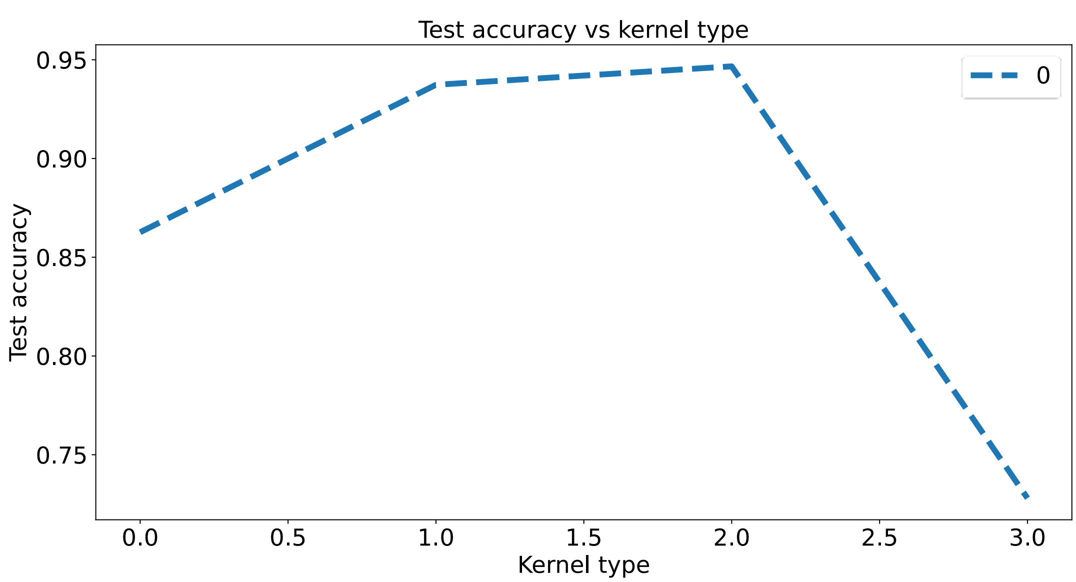


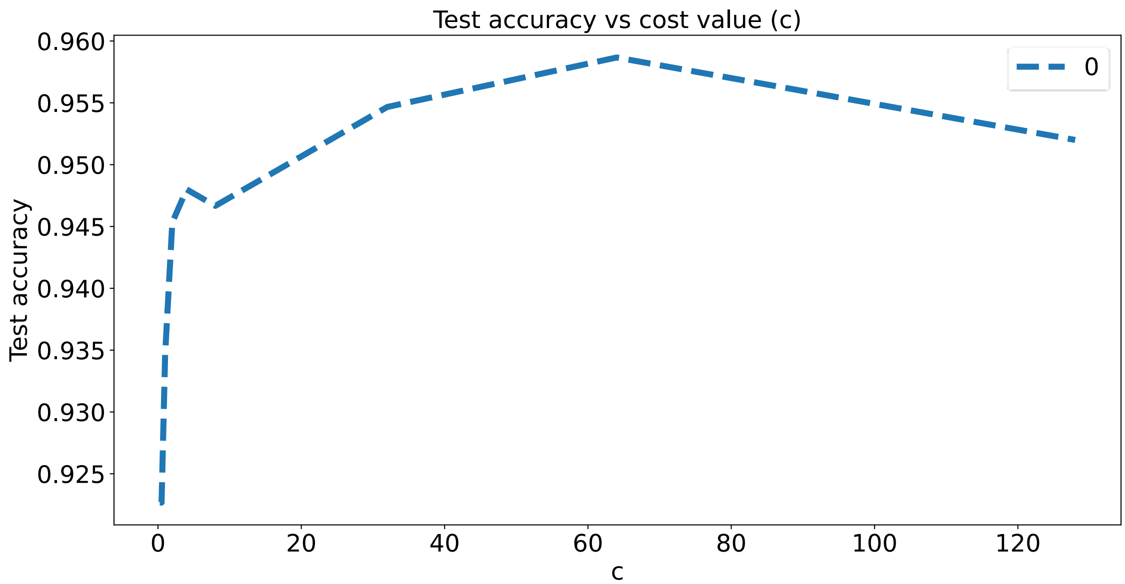
Test accuracy vs cost value (c)
0
Test accuracy
c
0.955
0.950
0.945
0.940
0.935
0.930
0
20
40
60
80
100
120

b) class_labels = 2, 3









Conclusions

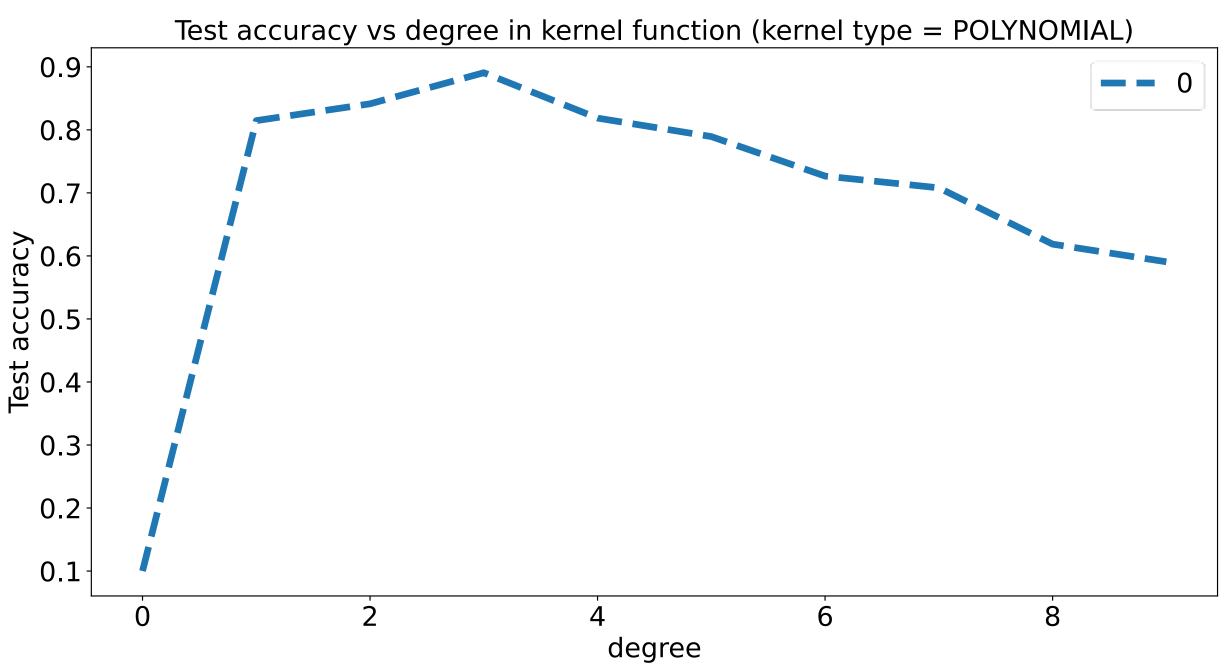
We find that best results are obtained for the following hyperparameters :

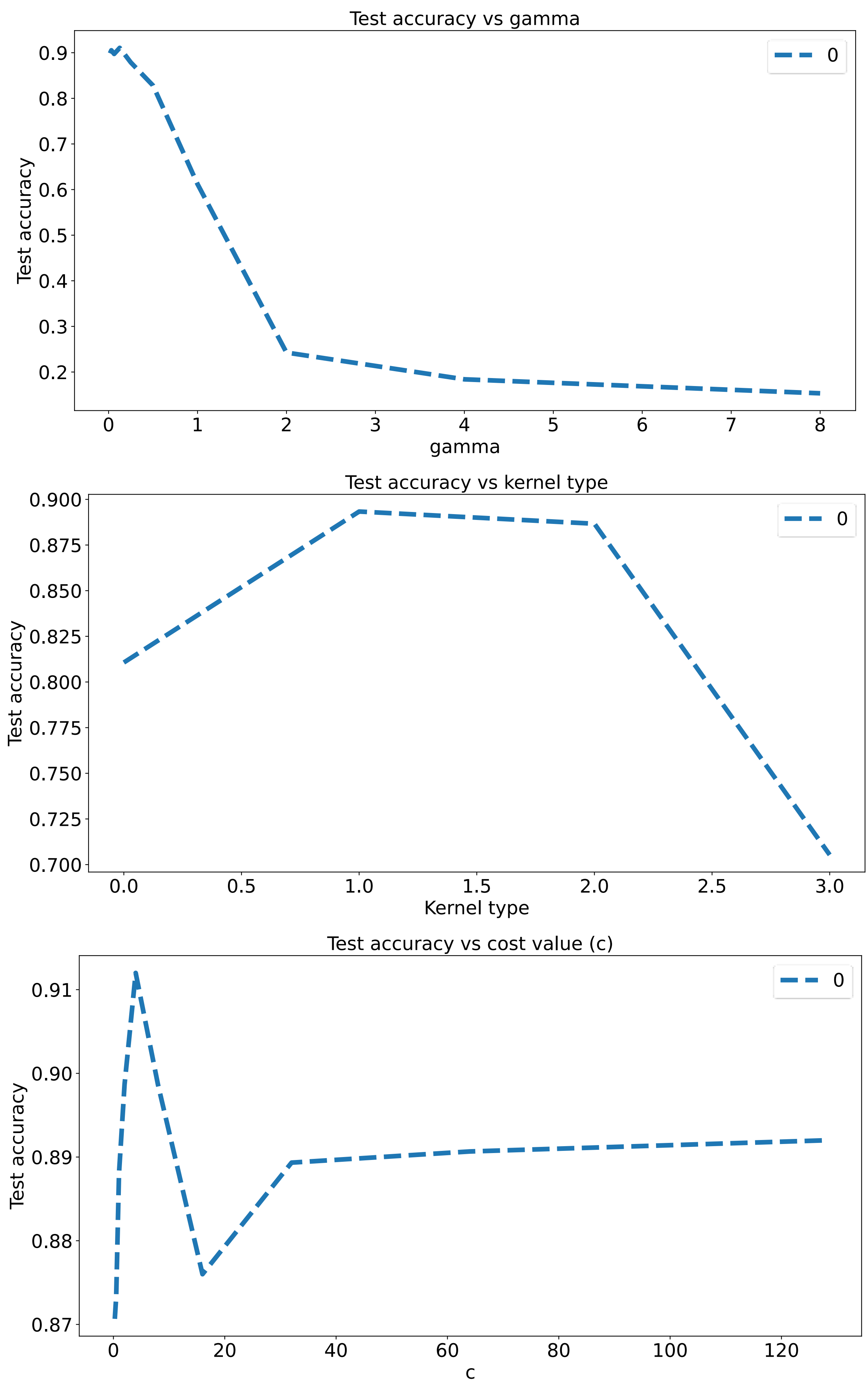
1) Kernel Type = Radial Basis Function
2) Degree in kernel function = 3 (for POLYNOMIAL type)
3) Cost value = 128 (for 0, 1), 64 (for 2, 3)
4) Gamma Value = 0.032 (for RBF)

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

1) Considering features = F1, F2, F3 … F10
    a. class_labels = 0, 1

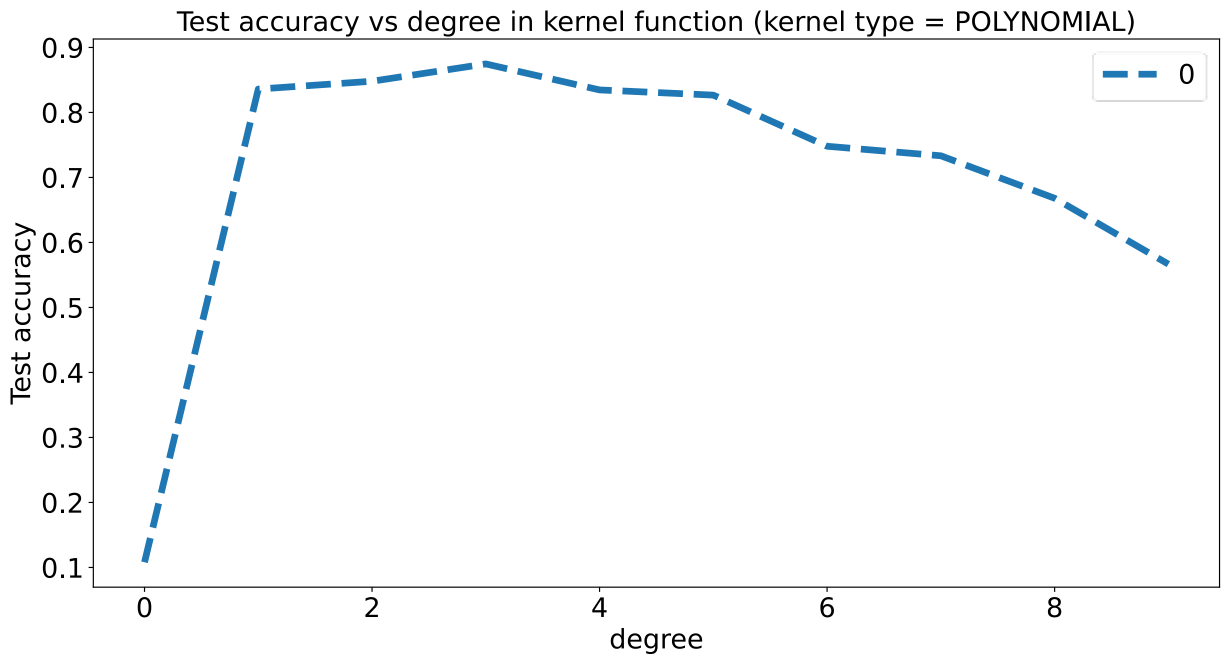
`-c 4 -t 1 -d 3 -g 0.032`

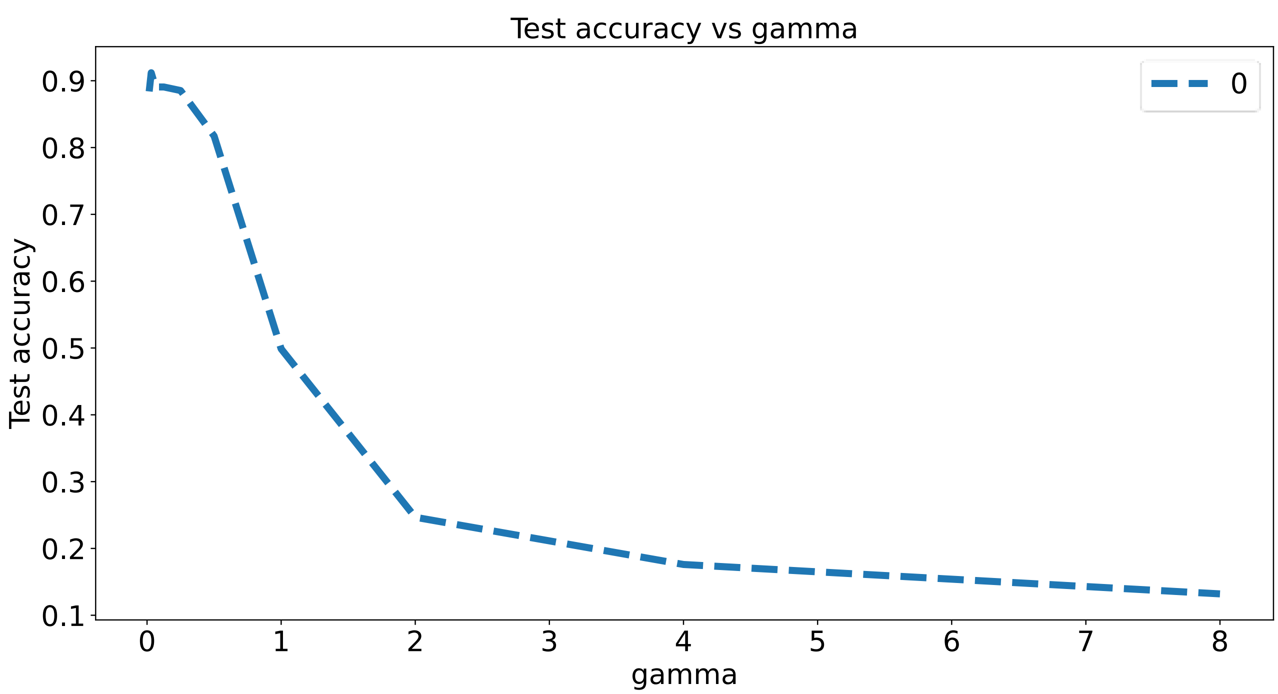


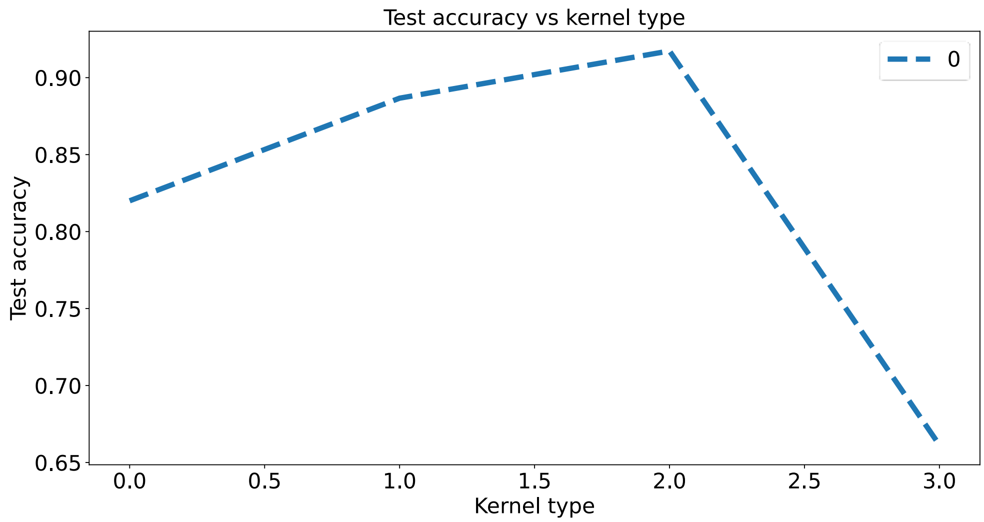

Test accuracy vs gamma
Test accuracy
gamma
0
Test accuracy vs kernel type
Test accuracy
Kernel type
0
Test accuracy vs cost value (c)
Test accuracy
c
0

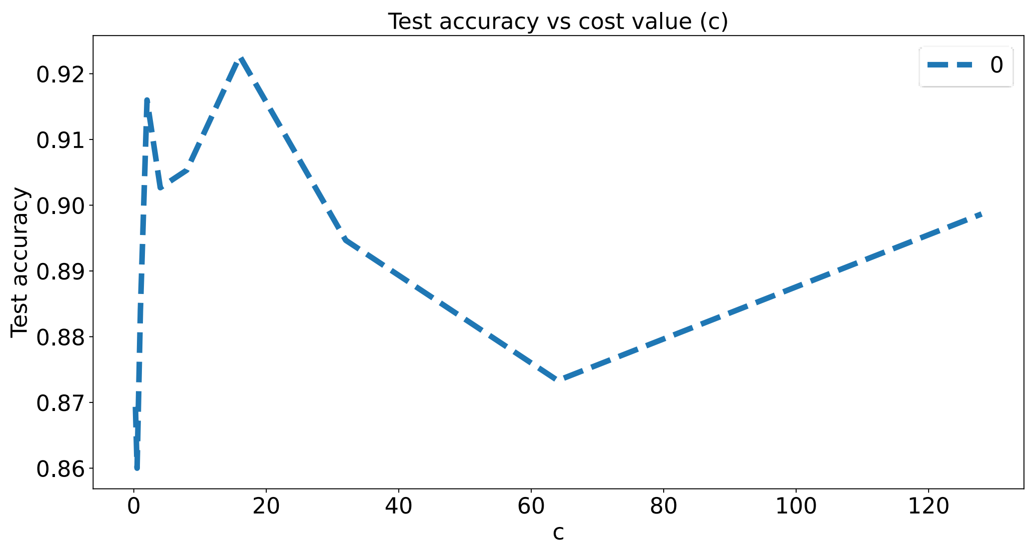
b. class_labels = 2, 3

-c 16 -t 2 -d 3 -g 0.032



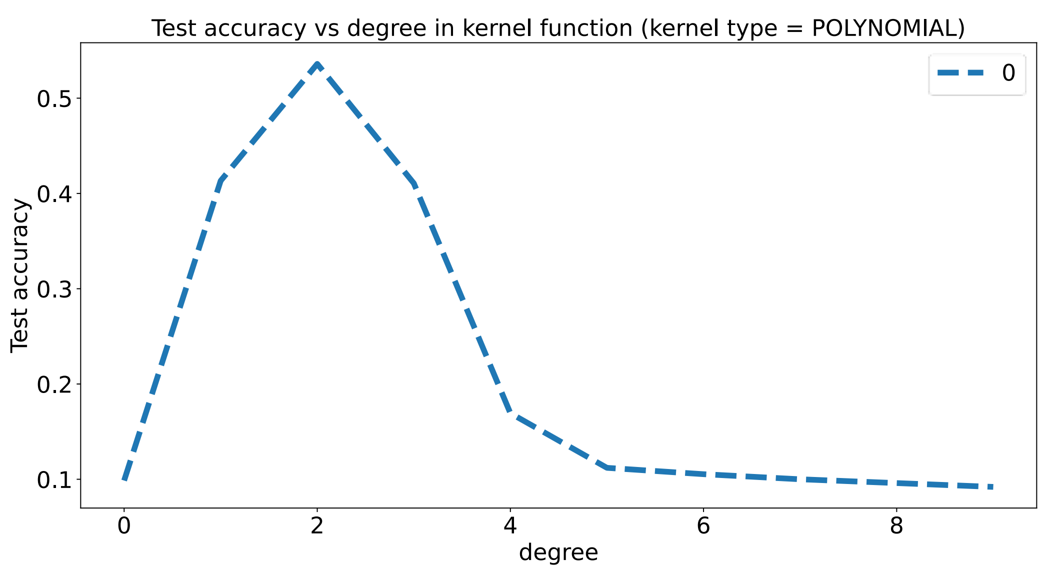


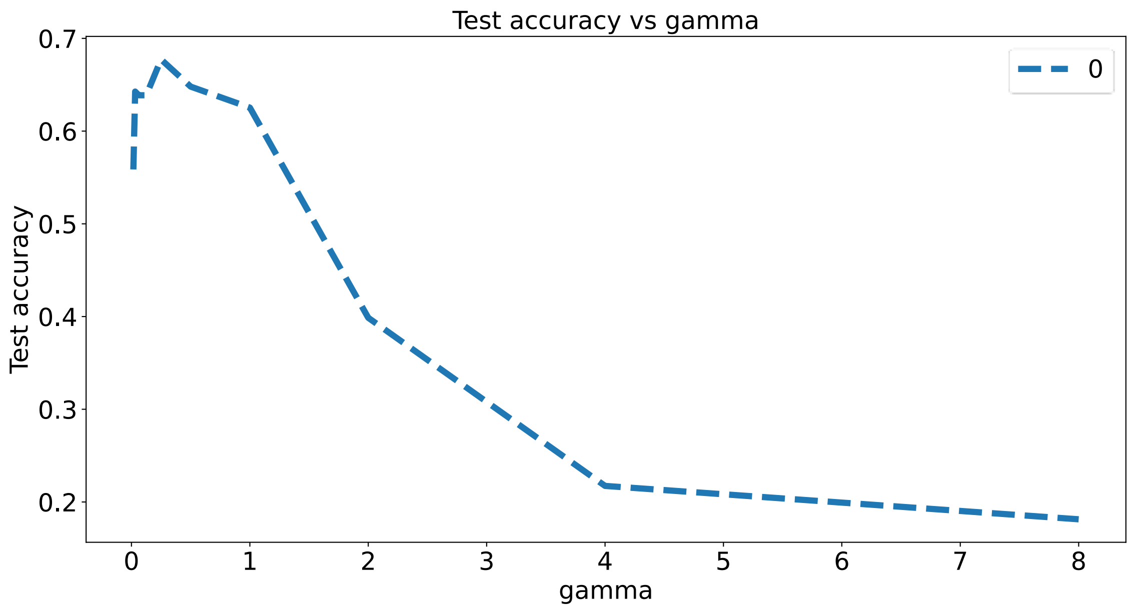




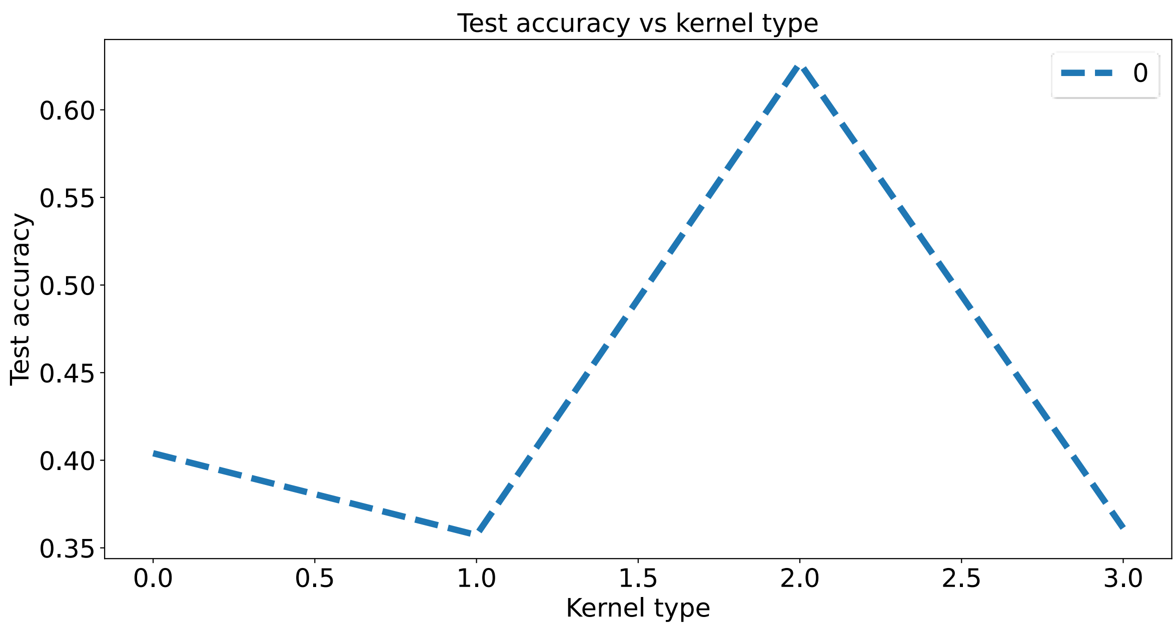
2) Considering features = F11, F12, F13 … F20
   a. class_labels = 0, 1

```
-c 128 -t 2 -d 2 -g 0.25
```

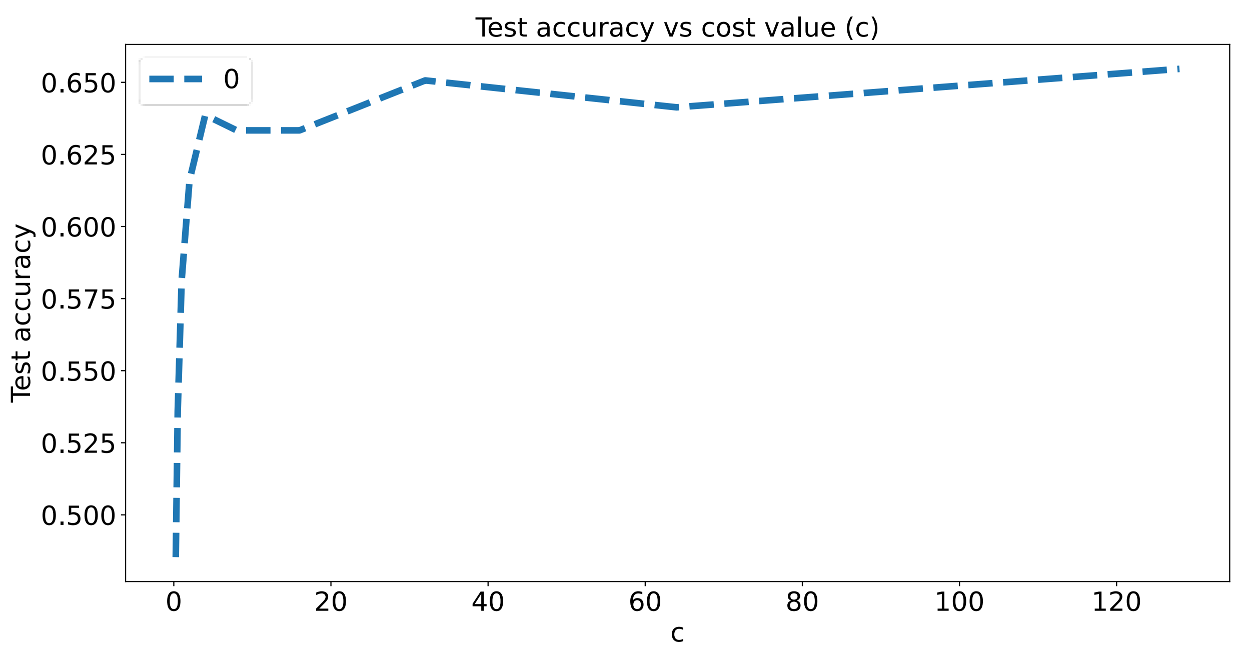




Test accuracy vs gamma
Test accuracy
0.2
0.3
0.4
0.5
0.6
0.7
0
1
2
3
4
5
6
7
8
gamma
0



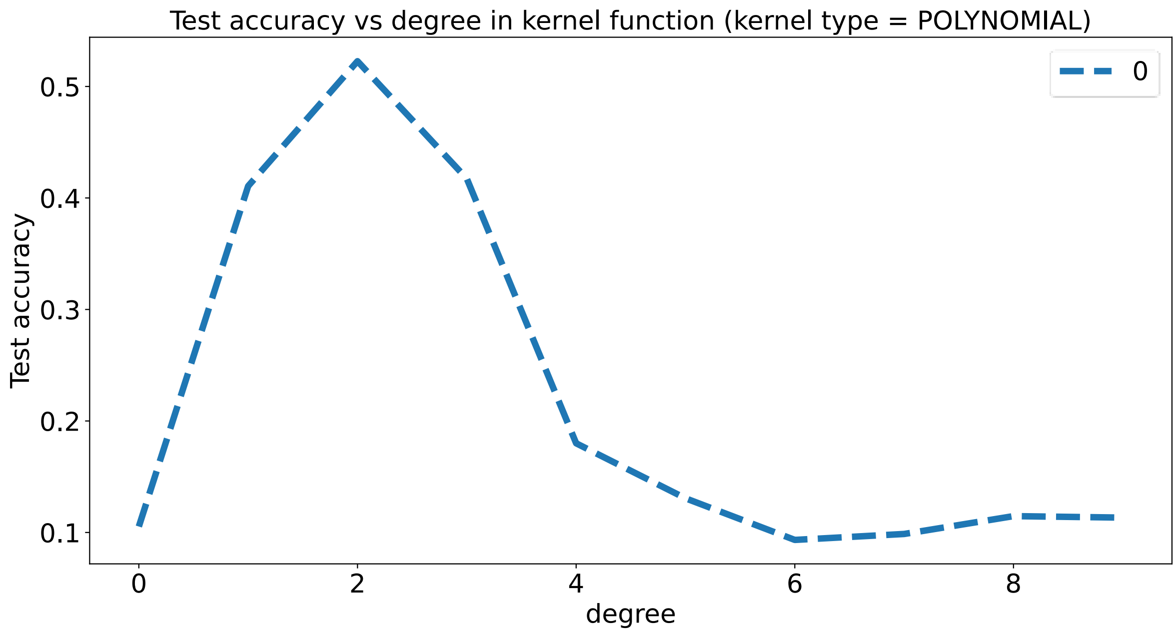
Test accuracy vs kernel type
Test accuracy
0.35
0.40
0.45
0.50
0.55
0.60
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Kernel type
0

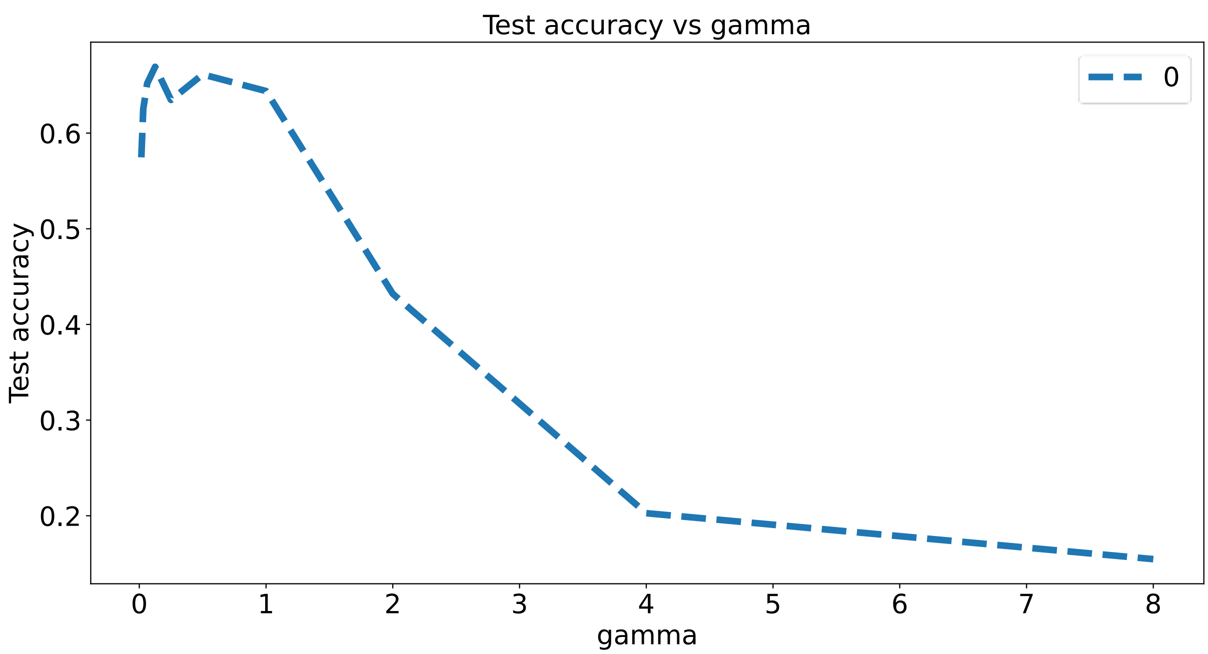


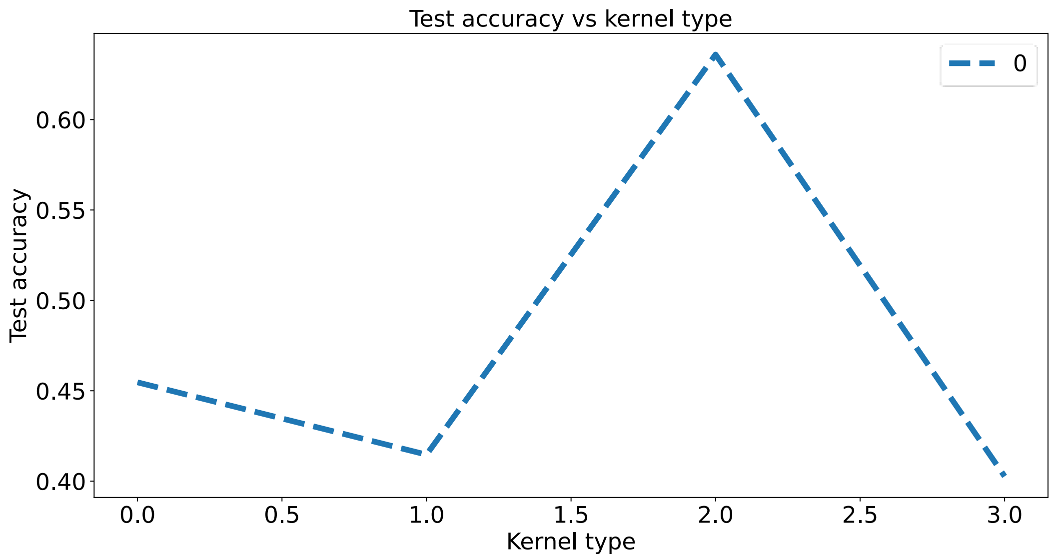
Test accuracy vs cost value (c)
Test accuracy
0.500
0.525
0.550
0.575
0.600
0.625
0.650
0
20
40
60
80
100
120
c
0

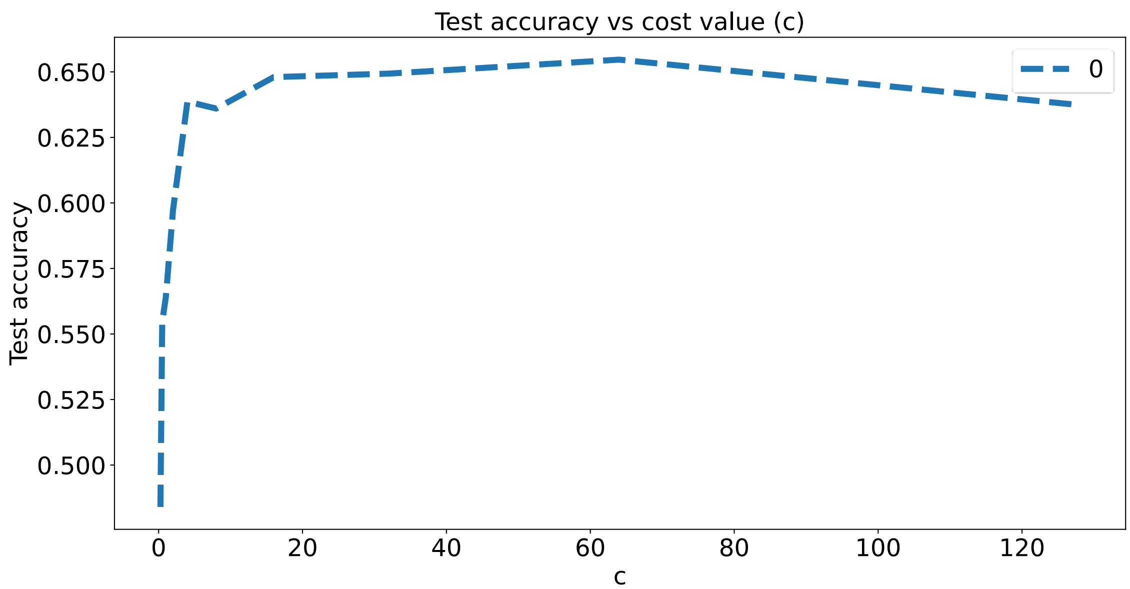
b. class_labels = 2, 3

`-c 64 -t 2 -d 2 -g 0.125`









Results:

We notice that optimal hyperparameter settings change considerably with lesser number of features under consideration and for different class labels. There is marginal variation in optimal hyperparameter conditions when all features are considered.

Multi-class Classification : LIBSVM

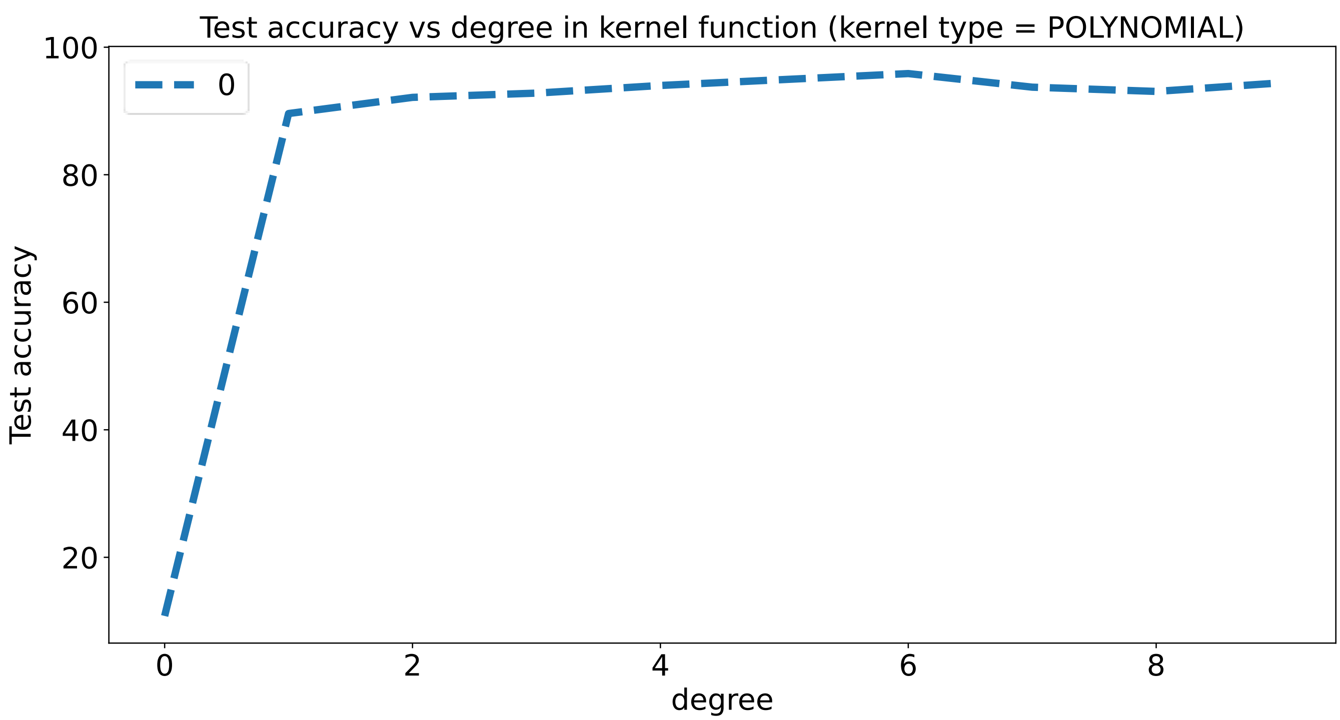
We evaluate our prediction of 'class_labels' for testing data (based on training data) w.r.t. actual 'class_labels' for testing part. Following plots have been obtained for accuracy vs various hyperparameters settings and multiple randomized instances of the same have been recorded:

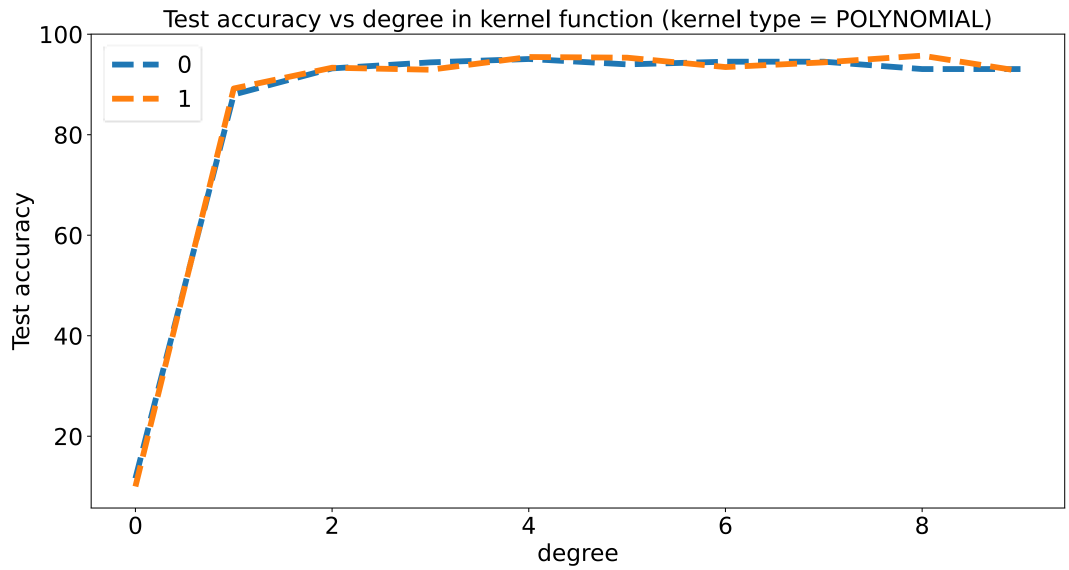
1) Test accuracy for 'polynomial' kernel VS degree in kernel function
2) Test accuracy for 'polynomial' kernel VS gamma
3) Test accuracy for 'RBF' kernel VS gamma
4) Test accuracy VS type of kernel function
5) Test accuracy VS C-value

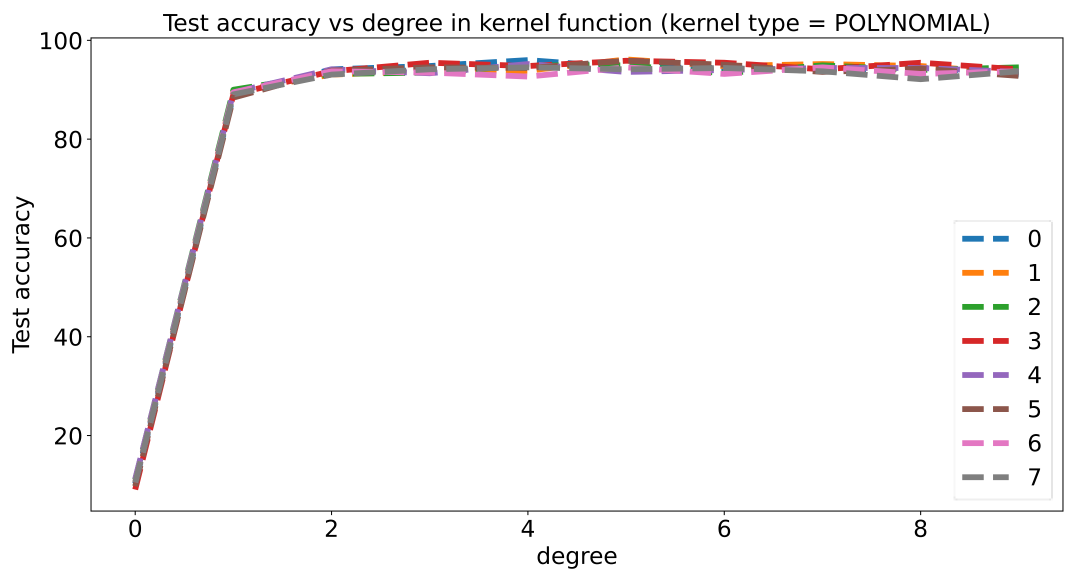
<u>Note.</u> The following steps are taken for each of the hyperparameter variations :

- 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 : [$2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}, 2^{7}$]

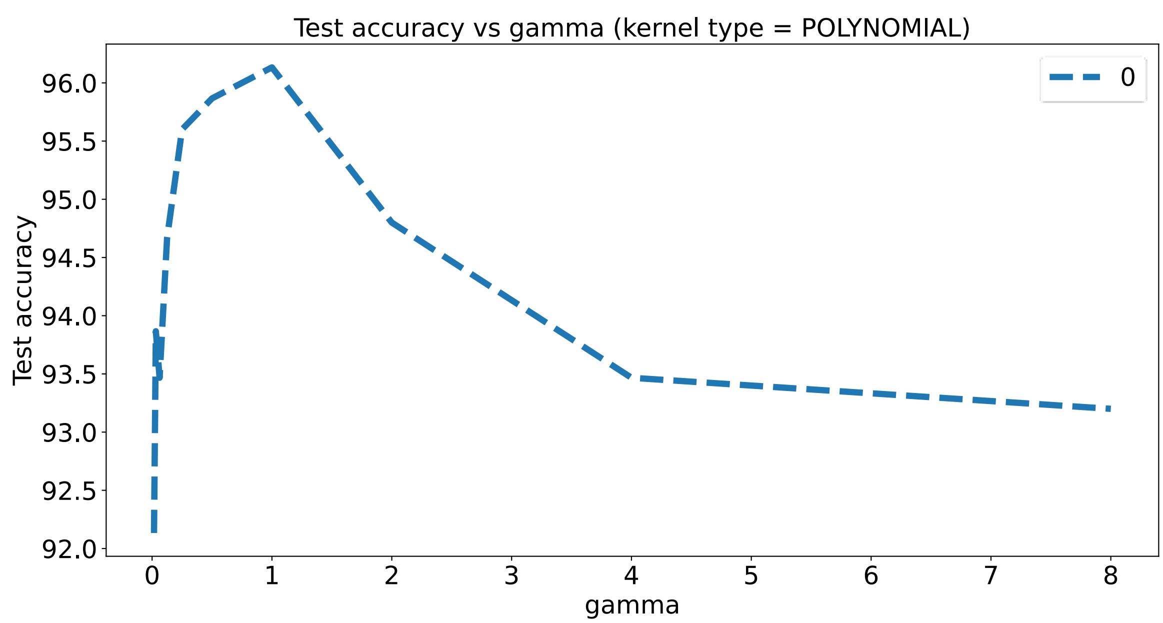
<u>Plot Type 1.</u>

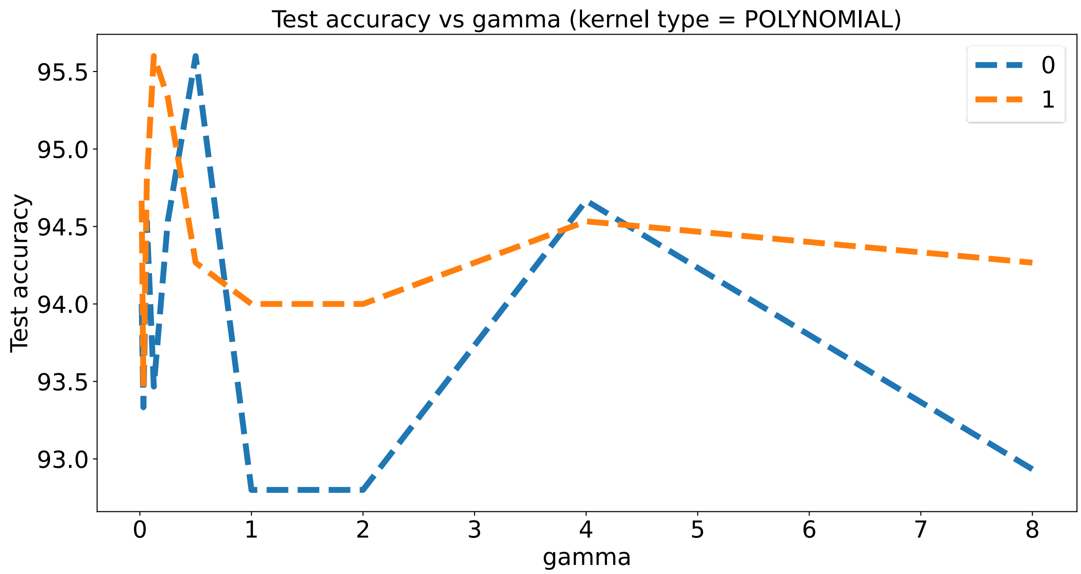



Test accuracy vs degree in kernel function (kernel type = POLYNOMIAL)
Test accuracy
degree
0
1

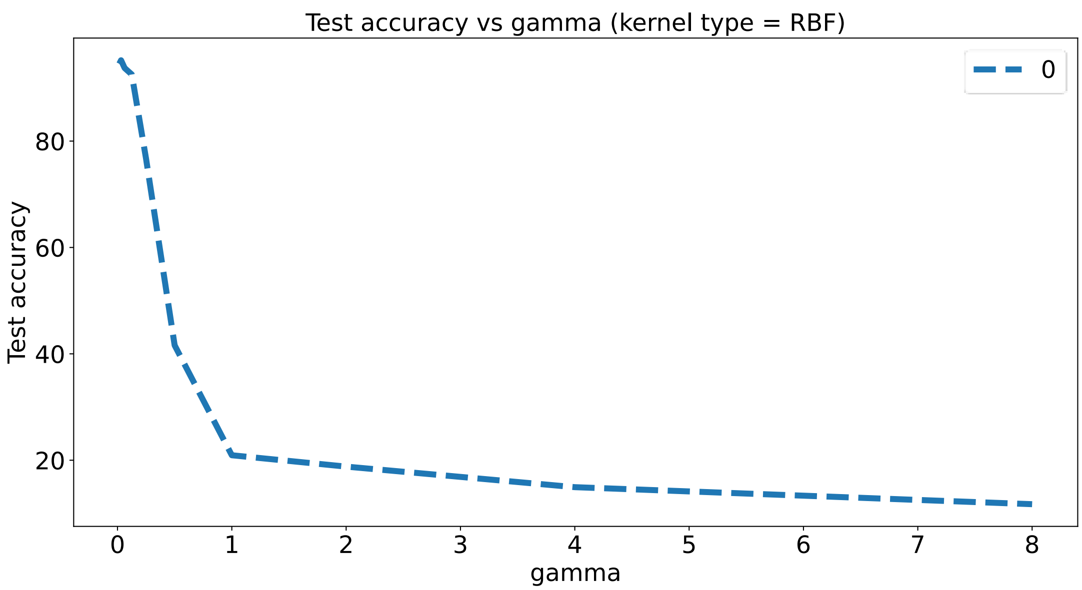

Test accuracy vs degree in kernel function (kernel type = POLYNOMIAL)
Test accuracy
degree
0
1
2
3
4
5
6
7

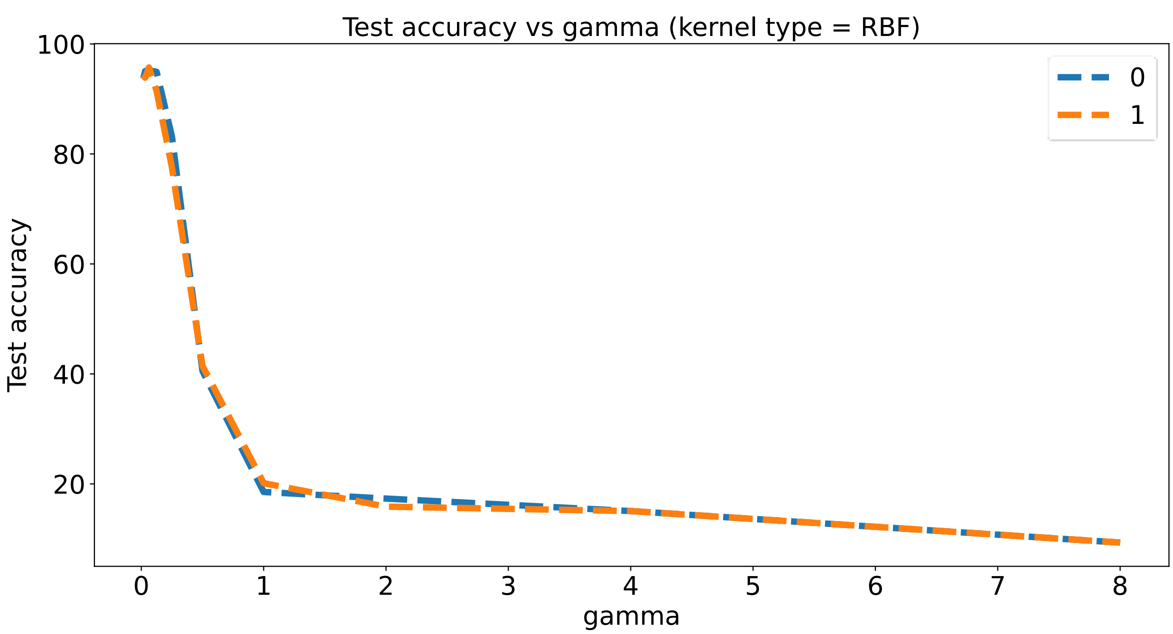
Plot Type 2.

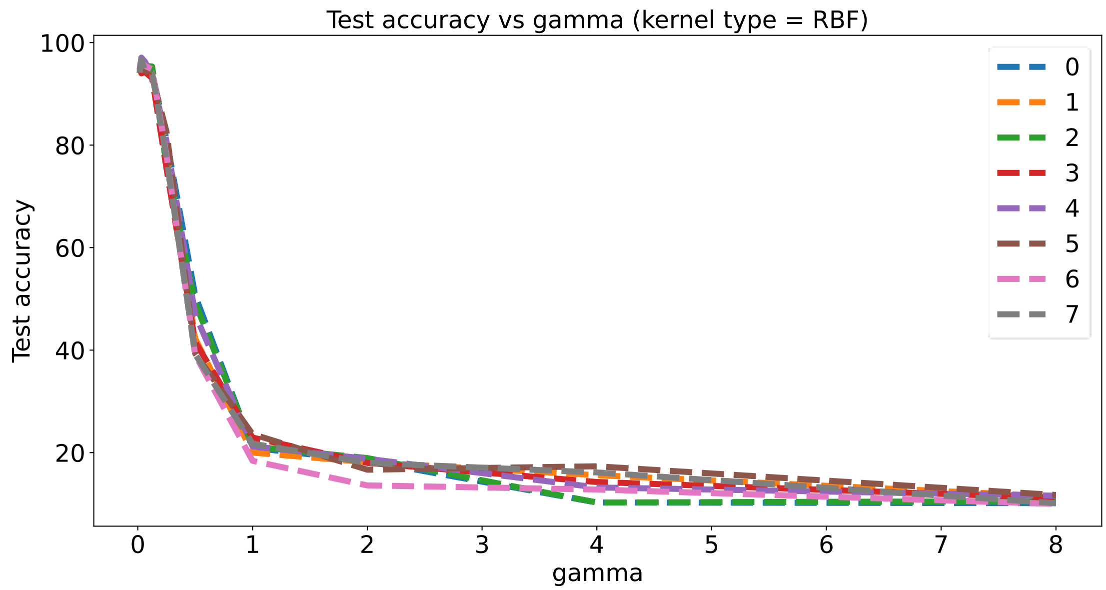

Test accuracy vs gamma (kernel type = POLYNOMIAL)
Test accuracy
gamma
0


Test accuracy vs gamma (kernel type = POLYNOMIAL)
0
1
95.5
95.0
94.5
94.0
93.5
93.0
Test accuracy
0
1
2
3
4
5
6
7
8
gamma

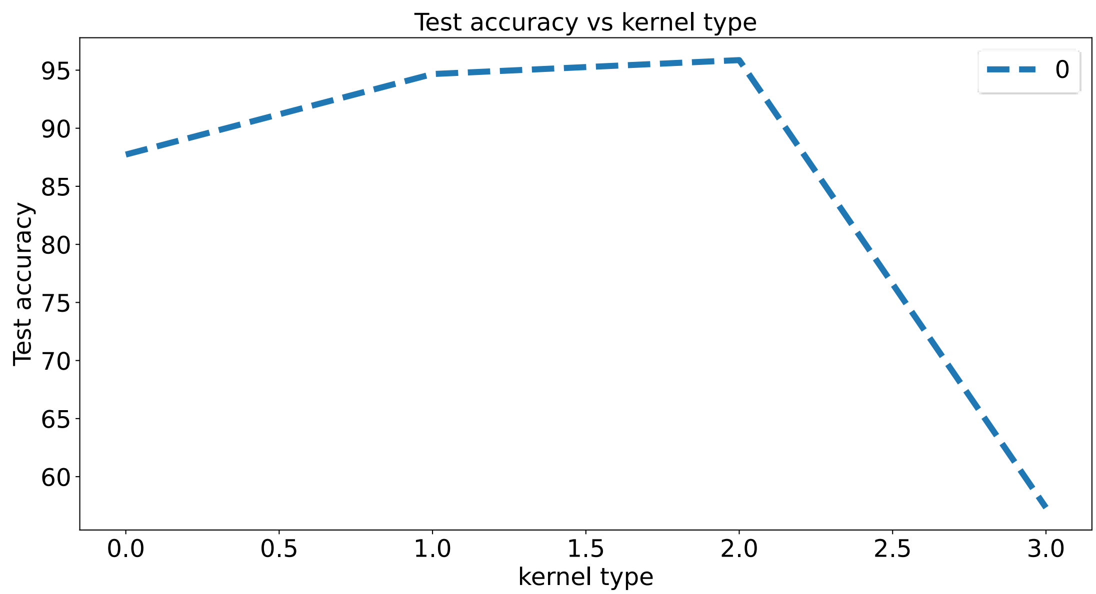
Plot Type 3.

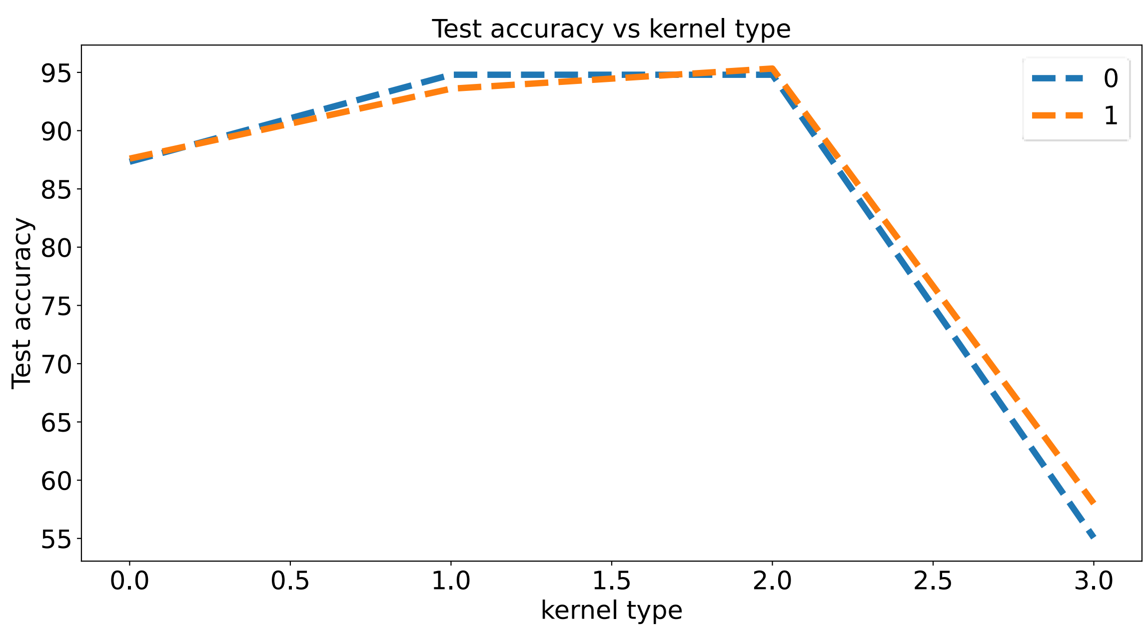

Test accuracy vs gamma (kernel type = RBF)
0
80
60
40
20
Test accuracy
0
1
2
3
4
5
6
7
8
gamma

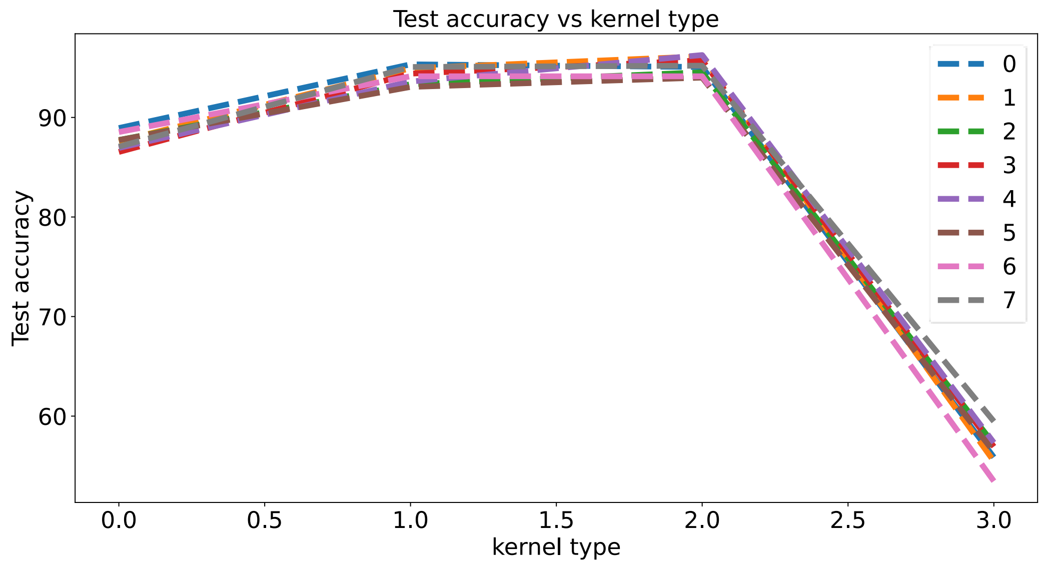

Test accuracy vs gamma (kernel type = RBF)
0
1
100
80
60
40
20
Test accuracy
0
1
2
3
4
5
6
7
8
gamma


Test accuracy vs gamma (kernel type = RBF)
Test accuracy
gamma
0
1
2
3
4
5
6
7

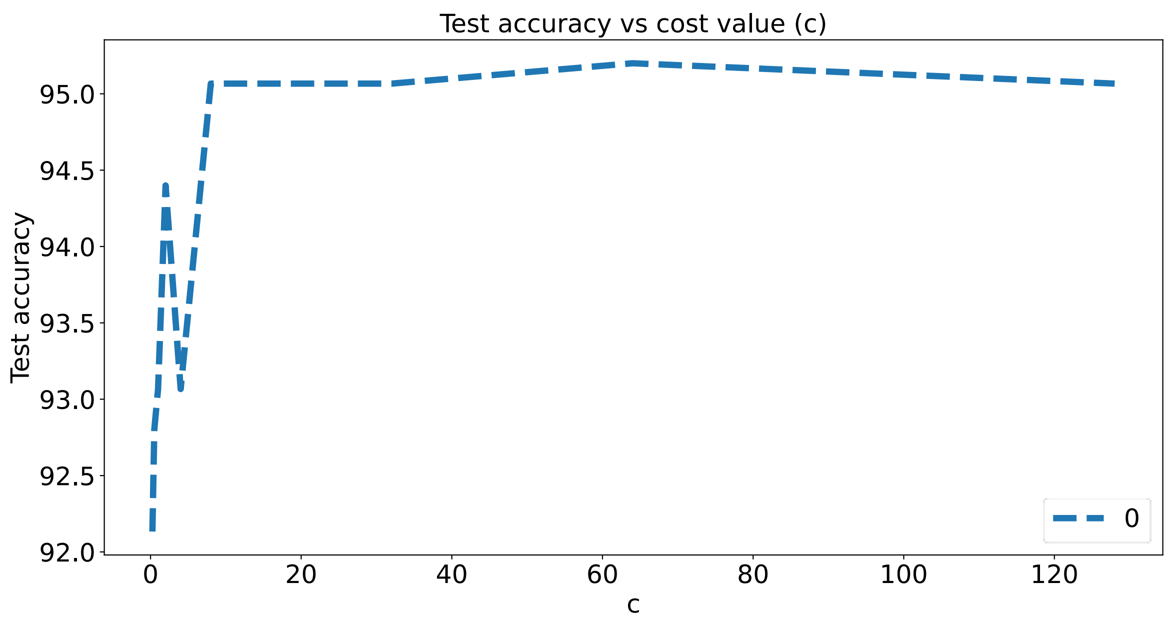
Plot Type 4.

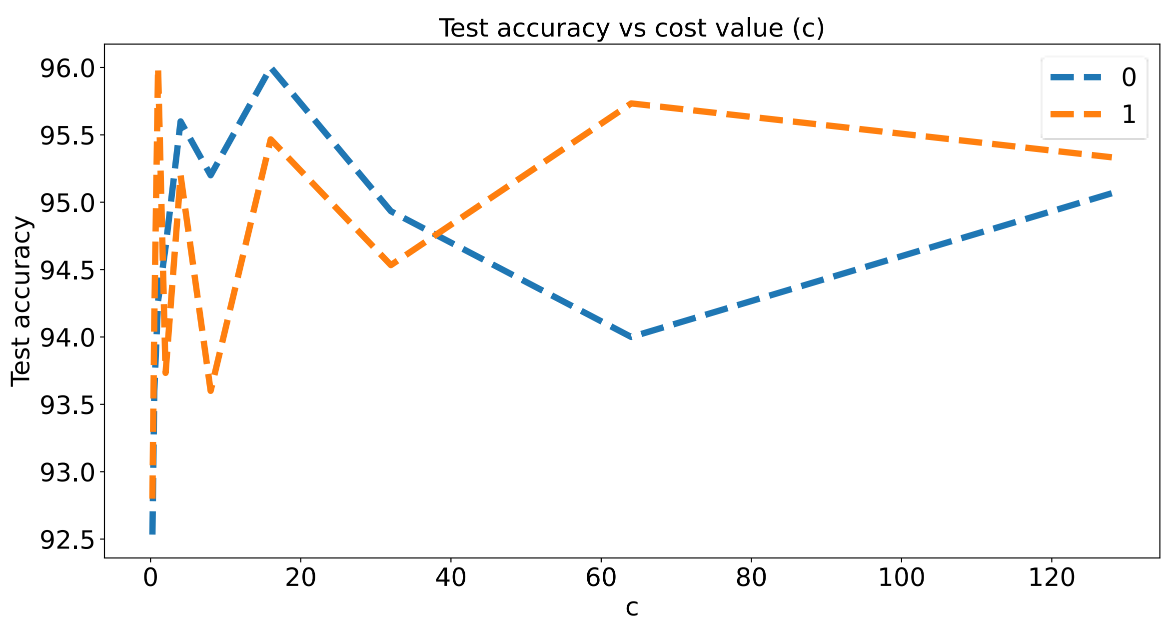

Test accuracy vs kernel type
Test accuracy
kernel type
0

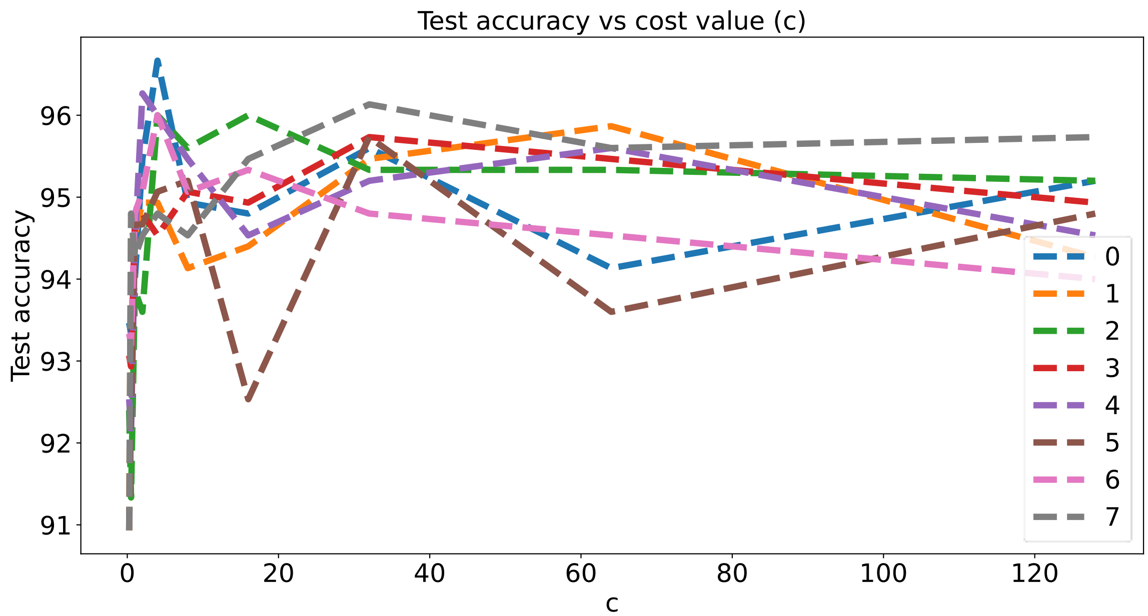

Test accuracy vs kernel type
Test accuracy
kernel type
0
1


Test accuracy vs kernel type
Test accuracy
kernel type
0
1
2
3
4
5
6
7

Plot Type 5.


Test accuracy vs cost value (c)
Test accuracy
c
0


Test accuracy vs cost value (c)
Test accuracy
c
0
1



Conclusions

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

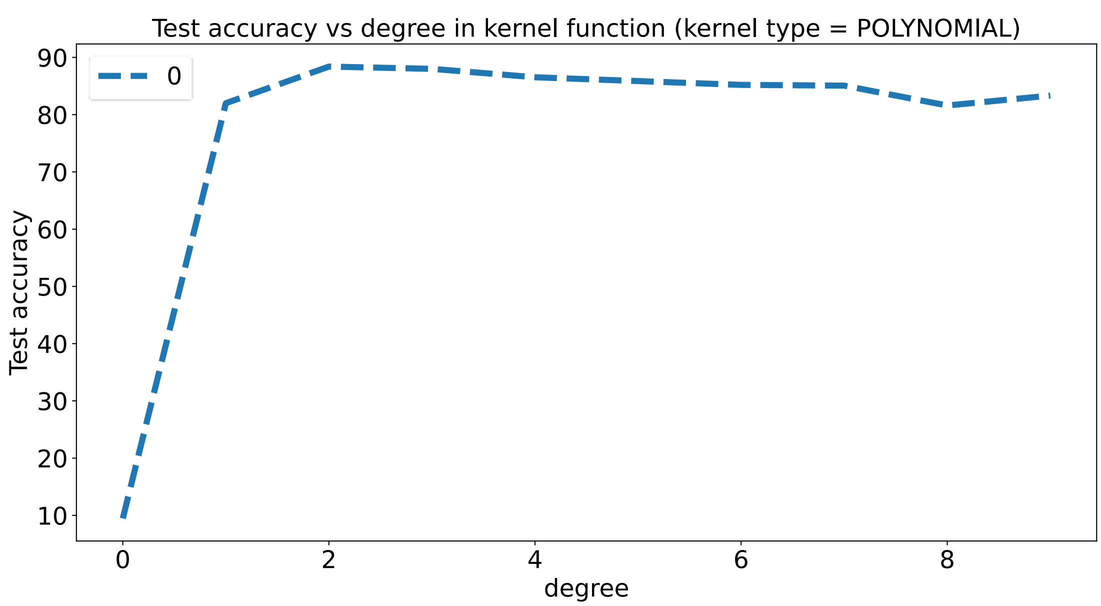
`-s 0 -c 4 -t 2 -r 1 -d 4 -g 0.032`

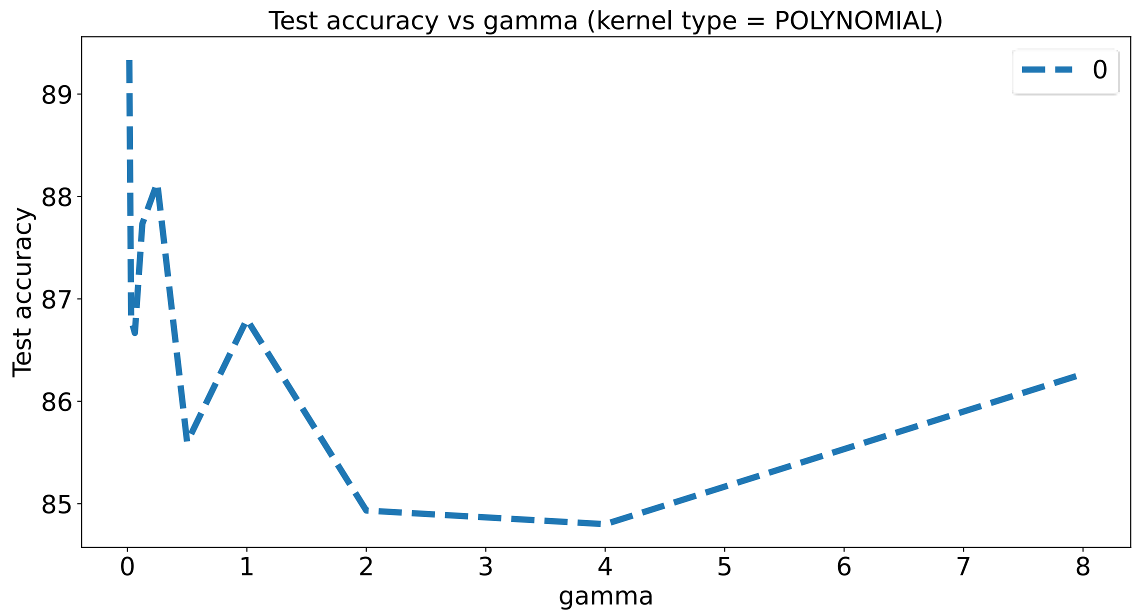
1) Kernel Type = Radial Basis Function
2) Degree in kernel function = 4, Gamma value = 0.5 (for POLYNOMIAL type)
3) Cost value = 4
4) Gamma Value = 0.032 (for RBF)

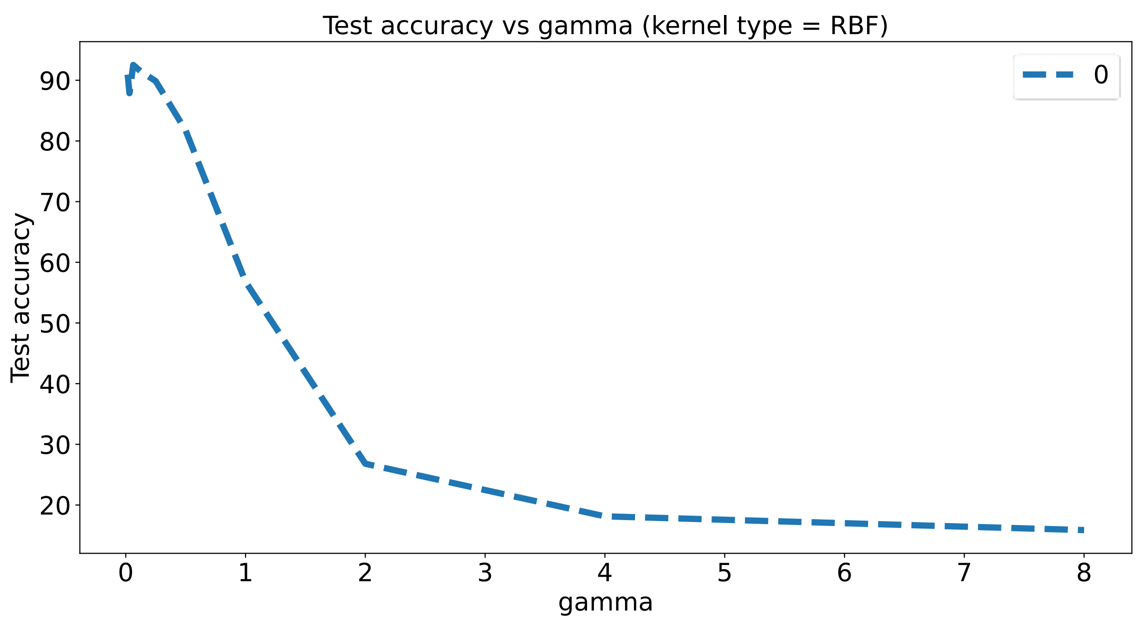
We now consider two pairs of 10 feature sets independently and obtain optimal hyperparameters for the same.

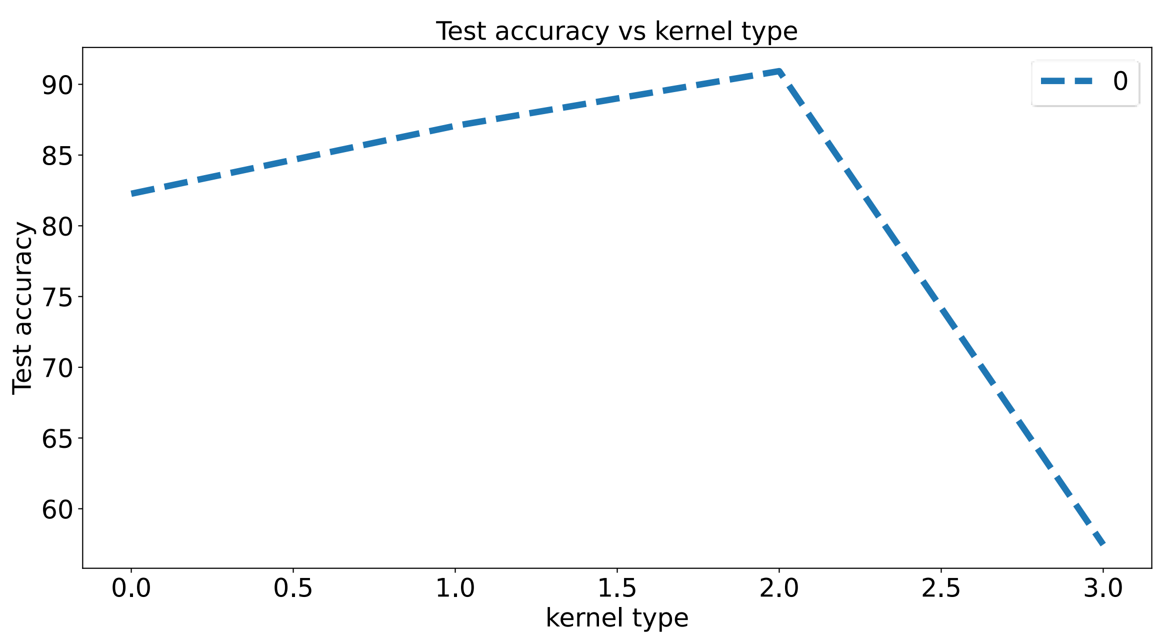
1) Considering features = F1, F2, F3 … F10

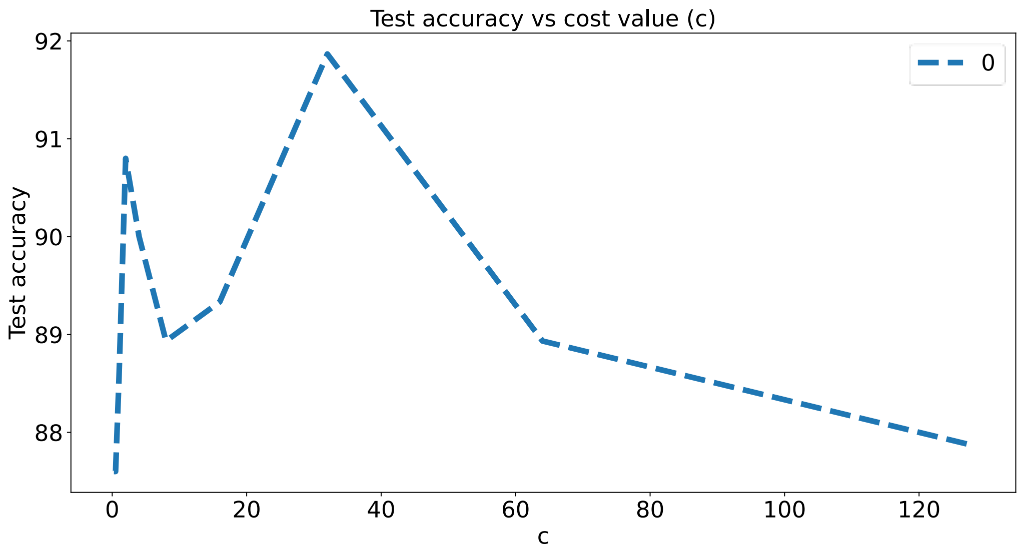
`-s 0 -c 32 -t 2 -r 1 -d 2 -g 0.032`




Test accuracy vs gamma (kernel type = POLYNOMIAL)
0
Test accuracy
85
86
87
88
89
gamma
0
1
2
3
4
5
6
7
8

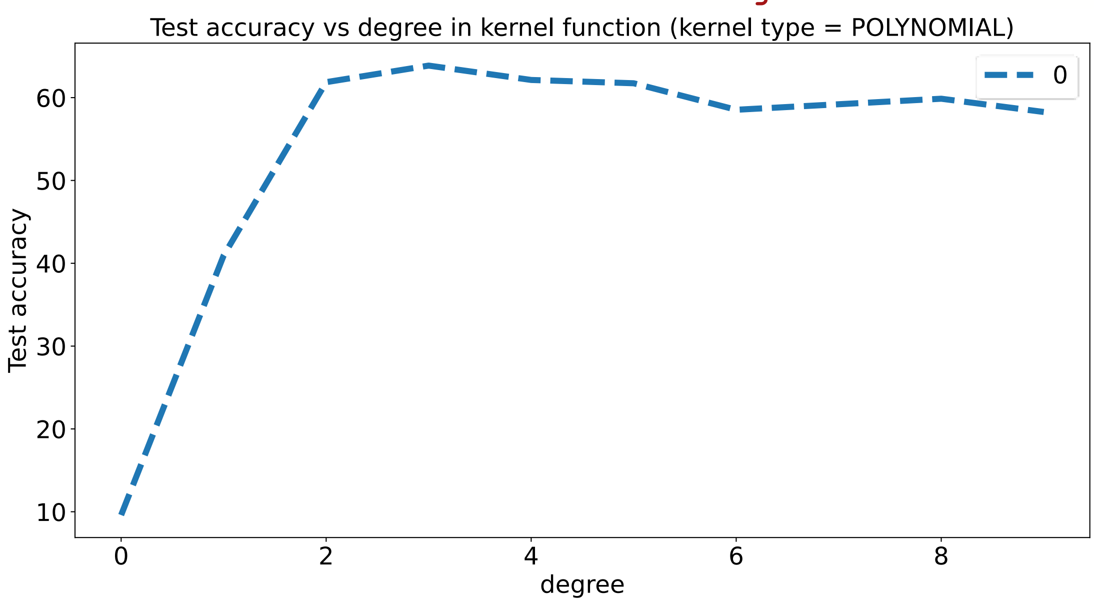

Test accuracy vs gamma (kernel type = RBF)
0
Test accuracy
20
30
40
50
60
70
80
90
gamma
0
1
2
3
4
5
6
7
8

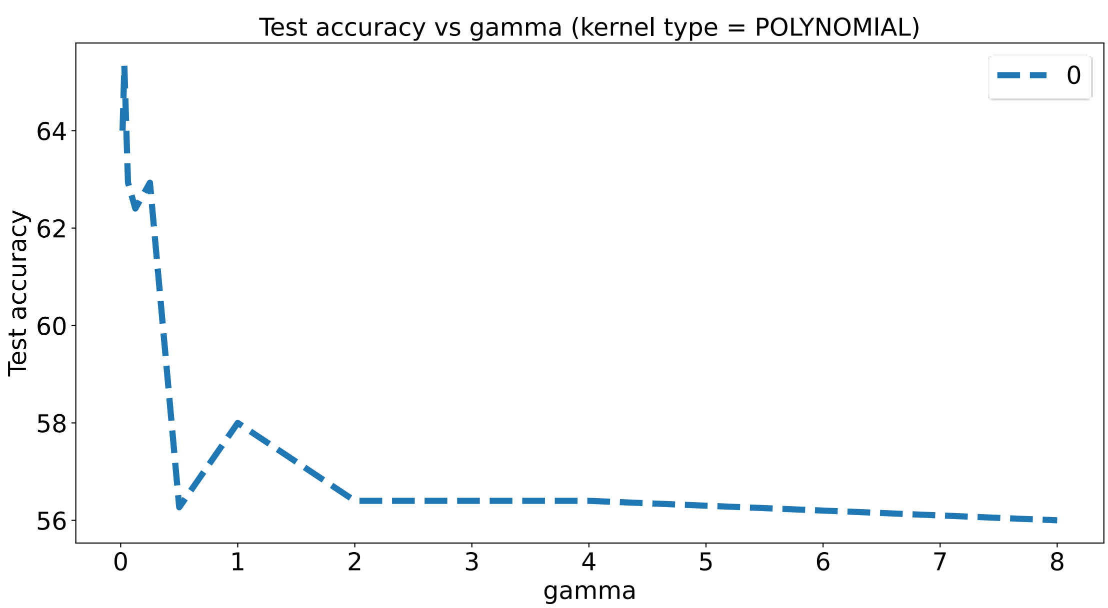

Test accuracy vs kernel type
0
Test accuracy
60
65
70
75
80
85
90
kernel type
0.0
0.5
1.0
1.5
2.0
2.5
3.0

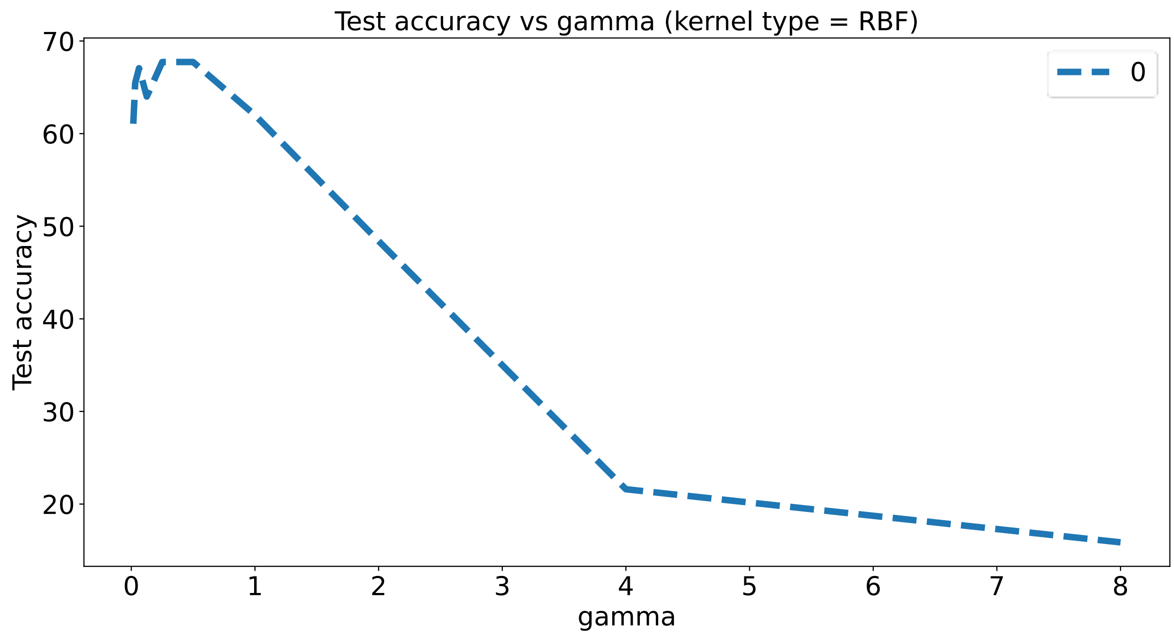


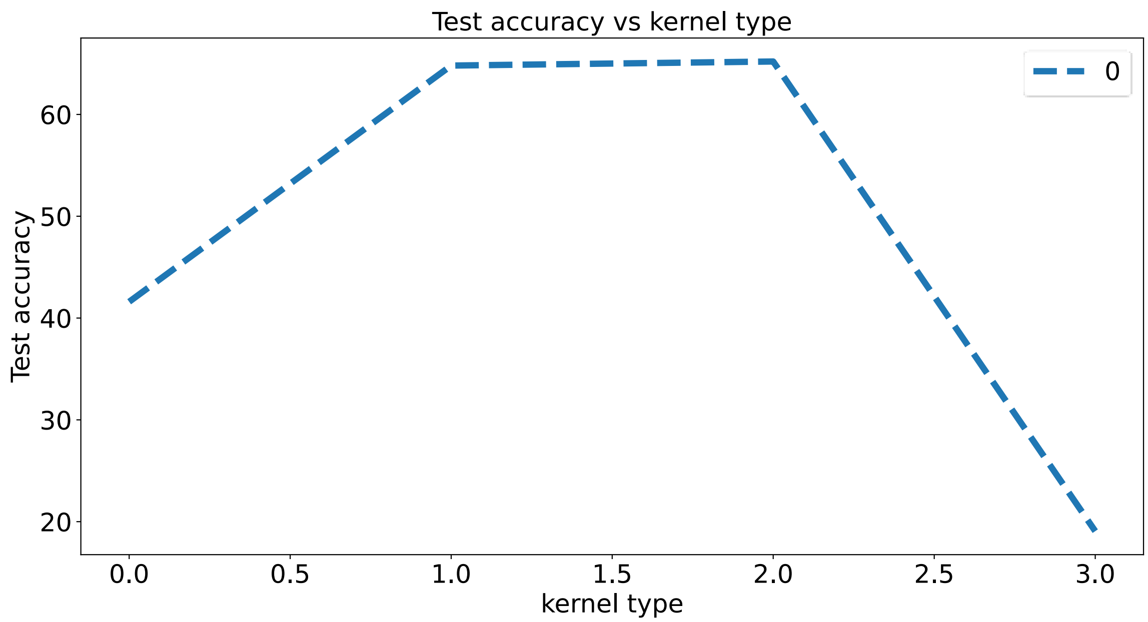
2) Considering features = F11, F12, F13 … F20

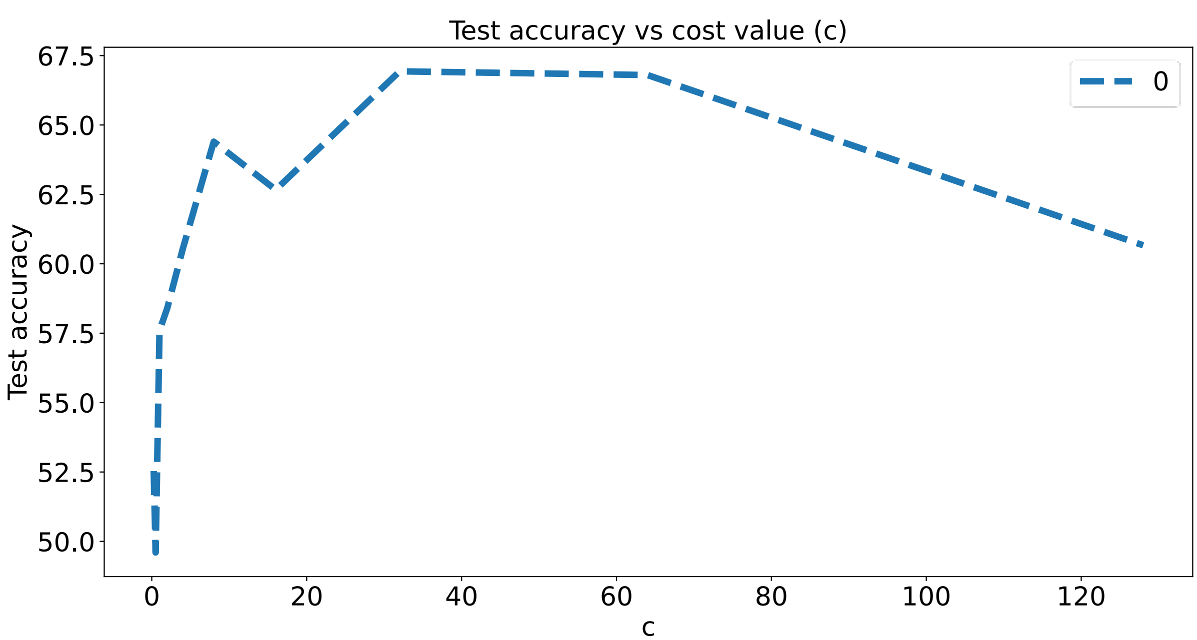
-s 0 -c 32 -t 2 -r 1 -d 3 -g 0.25






Test accuracy vs gamma (kernel type = RBF)
Test accuracy
gamma
0


Test accuracy vs kernel type
Test accuracy
kernel type
0


Test accuracy vs cost value (c)
Test accuracy
c
0

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

2 PART2

We perform multi-class classification using Sci-kit learn package with the training to testing ratio as 3:1, i.e., 75% training data to 25% testing data. We consider the curve patterns found in Part 1A and utilize them to guess the optimal value of hyperparameters with the limited number of score predictions.

We attempt the following variations in kernel types, C value, gamma & degree and find the scores to interpret the below-mentioned conclusions. The function defined takes the following parameters :

```
def save_csv(ktype, cval, gamma_val, deg_val, filename)
```

```
save_csv('linear', 2, 0.03, 2, 'submission.csv')
# SCORE = 0.91750

save_csv('poly', 2, 0.03, 2, 'submission1.csv')
```

```
# SCORE = 0.94625 => 'polynomial' kernel > 'linear' kernel

save_csv('poly', 2, 0.03, 3, 'submission2.csv')
# SCORE = 0.95500 => (DEGREE = 3) > (DEGREE = 2) in 'polynomial' kernel

save_csv('poly', 2, 0.03, 4, 'submission3.csv')
# SCORE = 0.93500 => (DEGREE = 3) > (DEGREE = 4) in 'polynomial' kernel

save_csv('poly', 2, 0.03, 3, 'submission4.csv')
# SCORE = 0.95500

save_csv('poly', 2, 0.05, 3, 'submission5.csv')
# SCORE = 0.94625 => increase in gamma value -> decrease in Score

save_csv('poly', 2, 0.1, 3, 'submission6.csv')
# SCORE = 0.94625

save_csv('poly', 2, 0.08, 3, 'submission7.csv')
# SCORE = 0.94625 => increase or decrease in gamma (beyong 0.5) -
> no change in Score

save_csv('poly', 4, 0.03, 3, 'submission8.csv')
# SCORE = 0.94875

save_csv('poly', 6, 0.03, 3, 'submission9.csv')
# SCORE = 0.94375 => increase or decrease in C -> slight change in Score

save_csv('poly', 2, 0.03, 3, 'submission4.csv')
# SCORE = 0.95500

save_csv('rbf', 2, 0.03, 3, 'submission10.csv')
# SCORE = 0.96625 => 'rbf' kernel > 'polynomial' kernel

save_csv('rbf', 2, 0.05, 3, 'submission11.csv')
# SCORE = 0.96625

save_csv('rbf', 2, 0.1, 3, 'submission12.csv')
# SCORE = 0.96500

save_csv('rbf', 2, 0.2, 3, 'submission13.csv')
# SCORE = 0.92125

save_csv('rbf', 2, 0.15, 3, 'submission14.csv')
# SCORE = 0.95625


save_csv('rbf', 2, 0.07, 3, 'submission15.csv')
# SCORE = 0.96750

save_csv('rbf', 2, 0.08, 3, 'submission16.csv')
```

```
# SCORE = 0.96750 => increase in Score till gamma = 0.8 and then decrease

save_csv('rbf', 2, 0.08, 2, 'submission17.csv')
# SCORE = 0.96750

save_csv('rbf', 2, 0.08, 4, 'submission18.csv')
# SCORE = 0.96750 => no change in Score with variation in degree

save_csv('rbf', 4, 0.08, 3, 'submission19.csv')
# SCORE = 0.96750

save_csv('rbf', 8, 0.08, 3, 'submission20.csv')
# SCORE = 0.96875

save_csv('rbf', 16, 0.08, 3, 'submission21.csv')
# SCORE = 0.96875

save_csv('rbf', 32, 0.08, 3, 'submission22.csv')
# SCORE = 0.96875 => slight increase in Score till C = 8 followed by no ch
ange
```

Conclusions

Scores follow the trend :

- **Linear < Polynomial < RBF**
- **(For Polynomial) C = 2, gamma = 0.03, degree = 3**
- **(For RBF) C = 8, gamma = 0.08, degree = 3 (optimal case, Score =0.96875)**