



Data Mining Course - Project #3

Professors:
Dr. Farahani, Dr. Kheradpishe

Ali Nikkhah - 99422197

May 2021

1.

الگوریتم های SVM از مجموعه ای از توابع ریاضی که به عنوان کرنل تعریف می شوند، استفاده می کنند. وظیفه کرنل این است که داده ها را به شکل مورد نیاز تبدیل کند. الگوریتم های مختلف SVM، از انواع مختلف توابع کرنل استفاده می کنند. این توابع می توانند انواع متفاوت داشته باشند. به عنوان مثال خطی، غیرخطی، چند جمله ای، تابع پایه شعاعی (RBF) و سیگموئید.

توابع کرنل، برای داده های تربیتی، نمودار ها، متن ها، تصاویر و همچنین بردار ها معرفی می شوند. پرکاربردترین نوع تابع کرنل، RBF است. زیرا دارای پاسخ محلی و متناهی در کل بازه محور X است. اما از لحاظ محاسباتی، محاسبه همه فیچر های اضافه مخصوصا در ترینینگ سنت

های بزرگ، هزینه زیادی دارد.

توابع کرنل، ضرب داخلی بین دو نقطه در یک فضای ویژگی مناسب را برمی گردانند. بنابراین، با هزینه محاسباتی کم، حتی در فضاهای با ابعاد بالا، مفهومی از شباهت را تعریف می کنند.

کرنل چند جمله ای

این کرنل در پردازش تصویر پرکاربرد است. معادله آن به صورت زیر است :

$$k(x_i, x_j) = (x_i \cdot x_j + 1)^d$$

که در آن d درجه چند جمله ای است.

کرنل گاوسی

این کرنل برای اهداف عمومی است. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود ندارد استفاده می شود. معادله آن به صورت زیر است :

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

تابع پایه شعاعی گاوسی (RBF)

این کرنل برای اهداف عمومی کاربرد دارد. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود نداشته باشد، مورد استفاده قرار می گیرد. معادله آن به صورت زیر است :

$$k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

و برای

$$\gamma > 0$$

گاهی اوقات با استفاده از پارامتر زیر استفاده می شود :

$$\gamma = \frac{1}{2\sigma^2}$$

کرنل RBF لاپلاس

این هم یک کرنل برای اهداف عمومی است. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود ندارد استفاده می شود. معادله آن به صورت زیر است :

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma}\right)$$

کرنل سیگموئید

می توان این کرنل را در شبکه های عصبی مورد استفاده قرار داد. معادله مریبوط به آن عبارت است از :

$$k(x, y) = \tanh(\alpha x^T y + c)$$

کرنل spline خطی بصورت یک بعدی

این هم یک کرنل برای اهداف عمومی است. کاربرد زیادی دارد. این کرنل اغلب در دسته بندی متن مورد استفاده قرار می گیرد. کرنل spline همچنین در مسائل رگرسیون عملکرد خوبی دارد. معادله آن عبارت است از :

$$k(x, y) = 1 + xy + xy \min(x, y) - \frac{x+y}{2} \min(x, y)^2 + \frac{1}{3} \min(x, y)^3$$

2.

In [113]:	import numpy as np import pandas as pd import matplotlib.pyplot as plt import seaborn as sns from scipy import stats																																																																																																																																																																															
In [114]:	mobile_test = pd.read_csv("test.csv") mobile_train = pd.read_csv("train.csv") display(mobile_train.shape) mobile_train.head(10)																																																																																																																																																																															
Out[114]:	(2000, 21) <table border="1"><thead><tr><th></th><th>battery_power</th><th>blue</th><th>clock_speed</th><th>dual_sim</th><th>fc</th><th>four_g</th><th>int_memory</th><th>m_dep</th><th>mobile_wt</th><th>n_cores</th><th>px_height</th><th>px_width</th><th>ram</th><th>sc_h</th><th>sc_w</th><th>talk_time</th><th>three_g</th><th>touch_screen</th><th>wifi</th><th>price_range</th></tr></thead><tbody><tr><td>0</td><td>842</td><td>0</td><td>2.2</td><td>0</td><td>1</td><td>0</td><td>7</td><td>0.6</td><td>188</td><td>2</td><td>...</td><td>20</td><td>756</td><td>2549</td><td>9</td><td>7</td><td>19</td><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1021</td><td>1</td><td>0.5</td><td>1</td><td>0</td><td>1</td><td>53</td><td>0.7</td><td>136</td><td>3</td><td>...</td><td>905</td><td>1988</td><td>2631</td><td>17</td><td>3</td><td>7</td><td>1</td><td>1</td><td>0</td><td>2</td></tr><tr><td>2</td><td>563</td><td>1</td><td>0.5</td><td>1</td><td>2</td><td>1</td><td>41</td><td>0.9</td><td>145</td><td>5</td><td>...</td><td>1263</td><td>1718</td><td>2603</td><td>11</td><td>2</td><td>9</td><td>1</td><td>1</td><td>0</td><td>2</td></tr><tr><td>3</td><td>615</td><td>1</td><td>2.5</td><td>0</td><td>0</td><td>0</td><td>10</td><td>0.8</td><td>131</td><td>6</td><td>...</td><td>1216</td><td>1788</td><td>2769</td><td>16</td><td>8</td><td>11</td><td>1</td><td>0</td><td>0</td><td>2</td></tr><tr><td>4</td><td>1821</td><td>1</td><td>1.2</td><td>0</td><td>13</td><td>1</td><td>44</td><td>0.6</td><td>141</td><td>2</td><td>...</td><td>1208</td><td>1212</td><td>1411</td><td>8</td><td>2</td><td>15</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>5</td><td>1859</td><td>0</td><td>0.5</td><td>1</td><td>3</td><td>0</td><td>22</td><td>0.7</td><td>164</td><td>1</td><td>...</td><td>1004</td><td>1654</td><td>1067</td><td>17</td><td>1</td><td>10</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>6</td><td>1821</td><td>0</td><td>1.7</td><td>0</td><td>4</td><td>1</td><td>10</td><td>0.8</td><td>139</td><td>8</td><td>...</td><td>381</td><td>1018</td><td>3220</td><td>13</td><td>8</td><td>18</td><td>1</td><td>0</td><td>1</td><td>3</td></tr></tbody></table>		battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range	0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	0	1	1	1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	1	0	2	2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1718	2603	11	2	9	1	1	0	2	3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1788	2769	16	8	11	1	0	0	2	4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	1	0	1	5	1859	0	0.5	1	3	0	22	0.7	164	1	...	1004	1654	1067	17	1	10	1	0	0	1	6	1821	0	1.7	0	4	1	10	0.8	139	8	...	381	1018	3220	13	8	18	1	0	1	3
	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range																																																																																																																																																												
0	842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	0	0	1	1																																																																																																																																																											
1	1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	1	1	0	2																																																																																																																																																											
2	563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1718	2603	11	2	9	1	1	0	2																																																																																																																																																											
3	615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1788	2769	16	8	11	1	0	0	2																																																																																																																																																											
4	1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	1	1	0	1																																																																																																																																																											
5	1859	0	0.5	1	3	0	22	0.7	164	1	...	1004	1654	1067	17	1	10	1	0	0	1																																																																																																																																																											
6	1821	0	1.7	0	4	1	10	0.8	139	8	...	381	1018	3220	13	8	18	1	0	1	3																																																																																																																																																											

```

7   1954    0     0.5    1    0     0     24   0.8    18/    4 ...    512    1149   /00    16    3     5     1     1     1     1     0
8   1445    1     0.5    0    0     0     53   0.7    174    7 ...    386    836   1099   17    1     20    1     0     0     0
9    509    1     0.6    1    2     1     9    0.1    93    5 ...   1137   1224   513    19    10    12    1     0     0     0

```

10 rows × 21 columns

```
In [113]: mobile_train.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2800 entries, 0 to 1999
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   battery_power    2800 non-null   int64  
 1   blue           2800 non-null   int64  
 2   clock_speed    2800 non-null   float64 
 3   dual_sim       2800 non-null   int64  
 4   fc             2800 non-null   int64  
 5   four_g         2800 non-null   int64  
 6   int_memory     2800 non-null   int64  
 7   m_dep          2800 non-null   float64 
 8   mobile_wt      2800 non-null   int64  
 9   n_cores        2800 non-null   int64  
 10  pc             2800 non-null   int64  
 11  px_height      2800 non-null   int64  
 12  px_width       2800 non-null   int64  
 13  ram            2800 non-null   int64  
 14  sc_h           2800 non-null   int64  
 15  sc_w           2800 non-null   int64  
 16  talk_time      2800 non-null   int64  
 17  three_g        2800 non-null   int64  
 18  touch_screen    2800 non-null   int64  
 19  wifi            2800 non-null   int64  
 20  price_range    2800 non-null   int64  
dtypes: float64(2), int64(19)
memory usage: 328.2 KB
```

```
In [144]: from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

y = mobile_train["price_range"].values
x_data=mobile_train.drop(["price_range"],axis=1)
x=(x_data-np.min(x_data))/(np.max(x_data)-np.min(x_data))
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=1)

svc=SVC(random_state=1)
svc.fit(x_train,y_train)
svc_pred = svc.predict(x_test)
svc_acc_score = accuracy_score(y_test, svc_pred)
accuracy = accuracy_score(y_test, svc_pred)

print("Accuracy : ", accuracy * 100)
print(classification_report(y_test,svc_pred))

Accuracy :  83.75
          precision    recall  f1-score   support
0       0.91   0.93   0.92    92
1       0.77   0.78   0.77    96
2       0.74   0.81   0.77    106
3       0.97   0.83   0.89    106

accuracy
macro avg       0.84   0.84   0.84    400
weighted avg    0.84   0.84   0.84    400
```

3 & 4.

C > 0 : soft margin
more larger C => more harder margin

gamma: Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

```
In [5]: for g in [1.,5.,10.,25.,50.]:
    for c in [1, 3, 5, 10, 40, 60, 80, 100]:
        svc = SVC(kernels="rbf", C=c, gamma=g)
        svc.fit(x_train, y_train)
        svc_pred = svc.predict(x_test)
        svc_acc_score = accuracy_score(y_test, svc_pred)

        accuracy = accuracy_score(y_test, svc_pred)
        print("C=", c)
        print("Gamma=", g)
        print("Accuracy : ", accuracy * 100)

        print(classification_report(y_test,svc_pred))

C= 1
Gamma= 0.1
Accuracy :  87.5
          precision    recall  f1-score   support
0       0.95   0.96   0.95    92
1       0.82   0.89   0.85    96
2       0.79   0.84   0.81    106
3       0.98   0.83   0.90    106

accuracy
macro avg       0.88   0.88   0.88    400
weighted avg    0.88   0.88   0.88    400

C= 3
Gamma= 0.1
Accuracy :  90.75
          precision    recall  f1-score   support
0       0.95   0.96   0.95    92
1       0.86   0.91   0.88    96
2       0.85   0.89   0.87    106
3       0.98   0.89   0.93    106

accuracy
macro avg       0.91   0.91   0.91    400
weighted avg    0.91   0.91   0.91    400

C= 5
Gamma= 0.1
Accuracy :  91.25
          precision    recall  f1-score   support
0       0.93   0.95   0.94    92
1       0.87   0.90   0.88    96
2       0.89   0.90   0.89    106
3       0.97   0.92   0.94    106

accuracy
macro avg       0.91   0.91   0.91    400
weighted avg    0.91   0.91   0.91    400

C= 10
Gamma= 0.1
Accuracy :  92.0
          precision    recall  f1-score   support
0       0.91   0.96   0.93    92
1       0.89   0.88   0.88    96
2       0.90   0.92   0.91    106
3       0.98   0.92   0.95    106

accuracy
macro avg       0.92   0.92   0.92    400
weighted avg    0.92   0.92   0.92    400

C= 40
Gamma= 0.1
Accuracy :  90.5
          precision    recall  f1-score   support
0       0.95   0.95   0.95    92
1       0.88   0.89   0.88    96
2       0.85   0.89   0.87    106
3       0.95   0.91   0.93    106

accuracy
macro avg       0.91   0.91   0.91    400
weighted avg    0.91   0.91   0.91    400

C= 60
Gamma= 0.1
Accuracy :  90.25
          precision    recall  f1-score   support
0       0.96   0.95   0.95    92
1       0.88   0.89   0.88    96
2       0.83   0.90   0.86    106
3       0.96   0.89   0.92    106

accuracy
macro avg       0.91   0.90   0.90    400
weighted avg    0.91   0.90   0.90    400

C= 80
Gamma= 0.1
Accuracy :  89.5
```

precision recall f1-score support
 0 0.94 0.92 0.93 92
 1 0.86 0.86 0.86 96
 2 0.83 0.91 0.86 106
 3 0.96 0.89 0.92 106

 accuracy macro avg weighted avg
 0.90 0.90 0.90 400

 C= 100
 Gamma= 0.1
 Accuracy : 89.25
 precision recall f1-score support
 0 0.94 0.92 0.93 92
 1 0.86 0.85 0.86 96
 2 0.82 0.91 0.86 106
 3 0.96 0.89 0.92 106

 accuracy macro avg weighted avg
 0.90 0.89 0.89 400

 C= 1
 Gamma= 0.5
 Accuracy : 79.5
 precision recall f1-score support
 0 0.89 0.89 0.89 92
 1 0.69 0.77 0.73 96
 2 0.70 0.73 0.71 106
 3 0.93 0.80 0.86 106

 accuracy macro avg weighted avg
 0.80 0.80 0.80 400

 C= 3
 Gamma= 0.5
 Accuracy : 83.75
 precision recall f1-score support
 0 0.91 0.90 0.91 92
 1 0.76 0.81 0.79 96
 2 0.76 0.80 0.78 106
 3 0.94 0.84 0.89 106

 accuracy macro avg weighted avg
 0.84 0.84 0.84 400

 C= 5
 Gamma= 0.5
 Accuracy : 83.25
 precision recall f1-score support
 0 0.92 0.89 0.91 92
 1 0.75 0.80 0.77 96
 2 0.75 0.79 0.77 106
 3 0.94 0.85 0.89 106

 accuracy macro avg weighted avg
 0.84 0.83 0.84 400

 C= 10
 Gamma= 0.5
 Accuracy : 83.5
 precision recall f1-score support
 0 0.92 0.89 0.91 92
 1 0.75 0.81 0.78 96
 2 0.76 0.78 0.77 106
 3 0.93 0.86 0.89 106

 accuracy macro avg weighted avg
 0.84 0.84 0.84 400

 C= 40
 Gamma= 0.5
 Accuracy : 83.25
 precision recall f1-score support
 0 0.92 0.89 0.91 92
 1 0.75 0.81 0.78 96
 2 0.75 0.78 0.77 106
 3 0.93 0.85 0.89 106

 accuracy macro avg weighted avg
 0.84 0.83 0.84 400

 C= 60
 Gamma= 0.5
 Accuracy : 83.25
 precision recall f1-score support
 0 0.92 0.89 0.91 92
 1 0.75 0.81 0.78 96
 2 0.75 0.78 0.77 106
 3 0.93 0.85 0.89 106

 accuracy macro avg weighted avg
 0.84 0.83 0.84 400

 C= 80
 Gamma= 0.5
 Accuracy : 83.25
 precision recall f1-score support
 0 0.92 0.89 0.91 92
 1 0.75 0.81 0.78 96
 2 0.75 0.78 0.77 106
 3 0.93 0.85 0.89 106

 accuracy macro avg weighted avg
 0.84 0.83 0.84 400

 C= 100
 Gamma= 0.5
 Accuracy : 83.25
 precision recall f1-score support
 0 0.92 0.89 0.91 92
 1 0.75 0.81 0.78 96
 2 0.75 0.78 0.77 106
 3 0.93 0.85 0.89 106

 accuracy macro avg weighted avg
 0.84 0.83 0.84 400

 C= 1
 Gamma= 0.1
 Accuracy : 87.5
 precision recall f1-score support
 0 0.95 0.96 0.95 92
 1 0.82 0.89 0.85 96
 2 0.79 0.84 0.81 106
 3 0.98 0.83 0.90 106

 accuracy macro avg weighted avg
 0.88 0.88 0.88 400

 C= 3
 Gamma= 0.1
 Accuracy : 90.75
 precision recall f1-score support
 0 0.95 0.96 0.95 92
 1 0.86 0.91 0.88 96
 2 0.85 0.89 0.87 106
 3 0.98 0.89 0.93 106

 accuracy macro avg weighted avg
 0.91 0.91 0.91 400

 C= 5
 Gamma= 0.1
 Accuracy : 91.25
 precision recall f1-score support
 0 0.93 0.95 0.94 92
 1 0.87 0.90 0.88 96
 2 0.89 0.90 0.89 106
 3 0.97 0.92 0.94 106

 accuracy macro avg weighted avg
 0.91 0.91 0.91 400

 C= 10
 Gamma= 0.1
 Accuracy : 92.0
 precision recall f1-score support
 0 0.91 0.96 0.93 92

```

1 0.89 0.88 0.88 96
2 0.90 0.92 0.91 100
3 0.98 0.92 0.95 100

accuracy 0.92 400
macro avg 0.92 0.92 0.92 400
weighted avg 0.92 0.92 0.92 400

C= 40
Gamma= 0.1
Accuracy : 90.5
precision recall f1-score support
0 0.96 0.95 0.95 92
1 0.88 0.89 0.88 96
2 0.85 0.89 0.87 100
3 0.95 0.91 0.93 100

accuracy 0.91 400
macro avg 0.91 0.91 0.91 400
weighted avg 0.91 0.91 0.91 400

C= 60
Gamma= 0.1
Accuracy : 90.25
precision recall f1-score support
0 0.96 0.95 0.95 92
1 0.88 0.89 0.88 96
2 0.83 0.90 0.86 100
3 0.96 0.89 0.92 100

accuracy 0.90 400
macro avg 0.91 0.90 0.90 400
weighted avg 0.91 0.90 0.90 400

C= 80
Gamma= 0.1
Accuracy : 89.5
precision recall f1-score support
0 0.94 0.92 0.93 92
1 0.86 0.86 0.86 96
2 0.83 0.91 0.86 100
3 0.96 0.89 0.92 100

accuracy 0.90 400
macro avg 0.90 0.90 0.90 400
weighted avg 0.90 0.90 0.90 400

C= 100
Gamma= 0.1
Accuracy : 89.25
precision recall f1-score support
0 0.94 0.92 0.93 92
1 0.86 0.85 0.86 96
2 0.82 0.91 0.86 100
3 0.96 0.89 0.92 100

accuracy 0.89 400
macro avg 0.90 0.89 0.89 400
weighted avg 0.90 0.89 0.89 400

C= 1
Gamma= 0.25
Accuracy : 86.0
precision recall f1-score support
0 0.92 0.93 0.93 92
1 0.78 0.82 0.80 96
2 0.77 0.84 0.81 100
3 0.99 0.85 0.91 100

accuracy 0.86 400
macro avg 0.87 0.86 0.86 400
weighted avg 0.87 0.86 0.86 400

C= 3
Gamma= 0.25
Accuracy : 88.0
precision recall f1-score support
0 0.90 0.93 0.91 92
1 0.83 0.85 0.84 96
2 0.84 0.87 0.85 100
3 0.97 0.87 0.92 100

accuracy 0.88 400
macro avg 0.88 0.88 0.88 400
weighted avg 0.88 0.88 0.88 400

C= 5
Gamma= 0.25
Accuracy : 88.5
precision recall f1-score support
0 0.92 0.93 0.93 92
1 0.81 0.86 0.84 96
2 0.83 0.86 0.85 100
3 0.98 0.89 0.93 100

accuracy 0.89 400
macro avg 0.89 0.89 0.89 400
weighted avg 0.89 0.89 0.89 400

C= 10
Gamma= 0.25
Accuracy : 88.25
precision recall f1-score support
0 0.93 0.92 0.92 92
1 0.82 0.84 0.83 96
2 0.82 0.87 0.84 100
3 0.97 0.90 0.93 100

accuracy 0.88 400
macro avg 0.89 0.88 0.88 400
weighted avg 0.89 0.88 0.88 400

C= 40
Gamma= 0.25
Accuracy : 87.5
precision recall f1-score support
0 0.94 0.91 0.93 92
1 0.81 0.82 0.82 96
2 0.81 0.86 0.83 100
3 0.95 0.91 0.93 100

accuracy 0.88 400
macro avg 0.88 0.88 0.88 400
weighted avg 0.88 0.88 0.88 400

C= 60
Gamma= 0.25
Accuracy : 87.5
precision recall f1-score support
0 0.94 0.91 0.93 92
1 0.81 0.82 0.82 96
2 0.81 0.86 0.83 100
3 0.95 0.91 0.93 100

accuracy 0.88 400
macro avg 0.88 0.88 0.88 400
weighted avg 0.88 0.88 0.88 400

C= 80
Gamma= 0.25
Accuracy : 87.5
precision recall f1-score support
0 0.94 0.91 0.93 92
1 0.81 0.82 0.82 96
2 0.81 0.86 0.83 100
3 0.95 0.91 0.93 100

accuracy 0.88 400
macro avg 0.88 0.88 0.88 400
weighted avg 0.88 0.88 0.88 400

C= 100
Gamma= 0.25
Accuracy : 87.5
precision recall f1-score support
0 0.94 0.91 0.93 92
1 0.81 0.82 0.82 96
2 0.81 0.86 0.83 100
3 0.95 0.91 0.93 100

accuracy 0.88 400
macro avg 0.88 0.88 0.88 400
weighted avg 0.88 0.88 0.88 400

C= 1
Gamma= 0.5
Accuracy : 79.5
precision recall f1-score support
0 0.89 0.89 0.89 92
1 0.69 0.77 0.73 96
2 0.70 0.73 0.71 100
3 0.89 0.89 0.89 100

```

	0.25	0.50	0.75	1.00
accuracy				
macro avg	0.80	0.80	0.80	0.80
weighted avg	0.80	0.80	0.80	0.80
C= 3 Gamma= 0.5 Accuracy : 83.75	precision	recall	f1-score	support
0	0.91	0.90	0.91	92
1	0.76	0.81	0.79	96
2	0.76	0.80	0.78	106
3	0.94	0.84	0.89	106
accuracy				
macro avg	0.84	0.84	0.84	400
weighted avg	0.84	0.84	0.84	400
C= 5 Gamma= 0.5 Accuracy : 83.25	precision	recall	f1-score	support
0	0.92	0.89	0.91	92
1	0.75	0.88	0.77	96
2	0.75	0.79	0.77	106
3	0.94	0.85	0.89	106
accuracy				
macro avg	0.84	0.83	0.84	400
weighted avg	0.84	0.83	0.83	400
C= 10 Gamma= 0.5 Accuracy : 83.5	precision	recall	f1-score	support
0	0.92	0.89	0.91	92
1	0.75	0.81	0.78	96
2	0.76	0.78	0.77	106
3	0.93	0.86	0.89	106
accuracy				
macro avg	0.84	0.84	0.84	400
weighted avg	0.84	0.83	0.84	400
C= 40 Gamma= 0.5 Accuracy : 83.25	precision	recall	f1-score	support
0	0.92	0.89	0.91	92
1	0.75	0.81	0.78	96
2	0.75	0.78	0.77	106
3	0.93	0.85	0.89	106
accuracy				
macro avg	0.84	0.83	0.84	400
weighted avg	0.84	0.83	0.83	400
C= 60 Gamma= 0.5 Accuracy : 83.25	precision	recall	f1-score	support
0	0.92	0.89	0.91	92
1	0.75	0.81	0.78	96
2	0.75	0.78	0.77	106
3	0.93	0.85	0.89	106
accuracy				
macro avg	0.84	0.83	0.84	400
weighted avg	0.84	0.83	0.83	400
C= 80 Gamma= 0.5 Accuracy : 83.25	precision	recall	f1-score	support
0	0.92	0.89	0.91	92
1	0.75	0.81	0.78	96
2	0.75	0.78	0.77	106
3	0.93	0.85	0.89	106
accuracy				
macro avg	0.84	0.83	0.84	400
weighted avg	0.84	0.83	0.83	400
C= 100 Gamma= 0.5 Accuracy : 83.25	precision	recall	f1-score	support
0	0.92	0.89	0.91	92
1	0.75	0.81	0.78	96
2	0.75	0.78	0.77	106
3	0.93	0.85	0.89	106
accuracy				
macro avg	0.84	0.83	0.84	400
weighted avg	0.84	0.83	0.83	400
C= 1 Gamma= 1 Accuracy : 68.75	precision	recall	f1-score	support
0	0.87	0.78	0.82	92
1	0.55	0.66	0.60	96
2	0.56	0.60	0.58	106
3	0.86	0.72	0.78	106
accuracy				
macro avg	0.71	0.69	0.70	400
weighted avg	0.71	0.69	0.69	400
C= 3 Gamma= 1 Accuracy : 72.25	precision	recall	f1-score	support
0	0.86	0.80	0.83	92
1	0.61	0.69	0.64	96
2	0.61	0.66	0.64	106
3	0.87	0.75	0.80	106
accuracy				
macro avg	0.74	0.72	0.73	400
weighted avg	0.74	0.72	0.73	400
C= 5 Gamma= 1 Accuracy : 72.25	precision	recall	f1-score	support
0	0.86	0.80	0.83	92
1	0.61	0.69	0.64	96
2	0.61	0.66	0.64	106
3	0.87	0.75	0.80	106
accuracy				
macro avg	0.74	0.72	0.73	400
weighted avg	0.74	0.72	0.73	400
C= 10 Gamma= 1 Accuracy : 72.25	precision	recall	f1-score	support
0	0.86	0.80	0.83	92
1	0.61	0.69	0.64	96
2	0.61	0.66	0.64	106
3	0.87	0.75	0.80	106
accuracy				
macro avg	0.74	0.72	0.73	400
weighted avg	0.74	0.72	0.73	400
C= 40 Gamma= 1 Accuracy : 72.25	precision	recall	f1-score	support
0	0.86	0.80	0.83	92
1	0.61	0.69	0.64	96
2	0.61	0.66	0.64	106
3	0.87	0.75	0.80	106
accuracy				
macro avg	0.74	0.72	0.73	400
weighted avg	0.74	0.72	0.73	400
C= 60 Gamma= 1 Accuracy : 72.25	precision	recall	f1-score	support
0	0.86	0.80	0.83	92
1	0.61	0.69	0.64	96
2	0.61	0.66	0.64	106
3	0.87	0.75	0.80	106
accuracy				
macro avg	0.74	0.72	0.73	400
weighted avg	0.74	0.72	0.73	400
accuracy	- - -	- - -	- - -	- - -

```

macro avg    0.74    0.72    0.73    400
weighted avg  0.74    0.72    0.73    400

C= 80
Gamma= 1
Accuracy : 72.25
precision    recall   f1-score   support
0            0.86    0.80    0.83    92
1            0.61    0.69    0.64    96
2            0.61    0.66    0.64    106
3            0.87    0.75    0.80    106

accuracy
macro avg    0.74    0.72    0.73    400
weighted avg  0.74    0.72    0.73    400

C= 100
Gamma= 1
Accuracy : 72.25
precision    recall   f1-score   support
0            0.86    0.80    0.83    92
1            0.61    0.69    0.64    96
2            0.61    0.66    0.64    106
3            0.87    0.75    0.80    106

accuracy
macro avg    0.74    0.72    0.73    400
weighted avg  0.74    0.72    0.73    400

```

```

In [6]: for c in [1, 3, 5, 10, 40, 60, 80, 100, 1e10]:
    svc = SVC(kernel="linear", C=c)
    svc.fit(x_train, y_train)
    svc_pred = svc.predict(x_test)
    svc_acc_score = accuracy_score(y_test, svc_pred)

    accuracy = accuracy_score(y_test, svc_pred)
    print("C=", c)
    print("Accuracy : ", accuracy * 100)

    print(classification_report(y_test,svc_pred))

C= 1
Accuracy : 93.0
precision    recall   f1-score   support
0            0.97    0.98    0.97    92
1            0.98    0.92    0.91    96
2            0.88    0.91    0.89    106
3            0.98    0.92    0.95    106

accuracy
macro avg    0.93    0.93    0.93    400
weighted avg  0.93    0.93    0.93    400

C= 3
Accuracy : 94.5
precision    recall   f1-score   support
0            0.98    0.99    0.98    92
1            0.92    0.94    0.93    96
2            0.91    0.92    0.91    106
3            0.98    0.94    0.96    106

accuracy
macro avg    0.95    0.95    0.95    400
weighted avg  0.95    0.94    0.95    400

C= 5
Accuracy : 95.25
precision    recall   f1-score   support
0            0.97    0.99    0.98    92
1            0.95    0.93    0.94    96
2            0.92    0.94    0.93    106
3            0.98    0.95    0.97    106

accuracy
macro avg    0.95    0.95    0.95    400
weighted avg  0.95    0.95    0.95    400

C= 10
Accuracy : 95.75
precision    recall   f1-score   support
0            0.98    1.00    0.99    92
1            0.94    0.94    0.94    96
2            0.93    0.93    0.93    106
3            0.99    0.96    0.98    106

accuracy
macro avg    0.96    0.96    0.96    400
weighted avg  0.96    0.96    0.96    400

C= 40
Accuracy : 95.25
precision    recall   f1-score   support
0            0.98    0.97    0.97    92
1            0.93    0.94    0.93    96
2            0.93    0.94    0.93    106
3            0.98    0.96    0.97    106

accuracy
macro avg    0.95    0.95    0.95    400
weighted avg  0.95    0.95    0.95    400

C= 60
Accuracy : 95.5
precision    recall   f1-score   support
0            0.98    0.97    0.97    92
1            0.94    0.94    0.94    96
2            0.93    0.95    0.94    106
3            0.98    0.96    0.97    106

accuracy
macro avg    0.96    0.95    0.96    400
weighted avg  0.96    0.95    0.96    400

C= 80
Accuracy : 96.25
precision    recall   f1-score   support
0            0.97    0.98    0.97    92
1            0.97    0.94    0.95    96
2            0.94    0.97    0.95    106
3            0.98    0.96    0.97    106

accuracy
macro avg    0.96    0.96    0.96    400
weighted avg  0.96    0.96    0.96    400

C= 100
Accuracy : 96.5
precision    recall   f1-score   support
0            0.97    0.98    0.97    92
1            0.96    0.95    0.95    96
2            0.95    0.96    0.96    106
3            0.98    0.97    0.98    106

accuracy
macro avg    0.96    0.97    0.96    400
weighted avg  0.97    0.96    0.96    400

```

```

In [7]: for g in [.1,.5,.10,.25,.50,1]:
    for c in [1, 3, 5, 10, 40, 60, 80, 100]:
        svc = SVC(kernel="poly", C=c, gamma=g)
        svc.fit(x_train, y_train)
        svc_pred = svc.predict(x_test)
        svc_acc_score = accuracy_score(y_test, svc_pred)

        accuracy = accuracy_score(y_test, svc_pred)
        print("C=", c)
        print("Gamma= ", g)
        print("Accuracy : ", accuracy * 100)

        print(classification_report(y_test,svc_pred))

C= 1
Gamma= 0.1
Accuracy : 80.0
precision    recall   f1-score   support
0            0.85    0.96    0.90    92
1            0.74    0.72    0.73    96
2            0.69    0.78    0.73    106

```

3 0.98 0.75 0.85 106

	accuracy	precision	recall	f1-score	support
accuracy	0.81	0.80	0.80	0.80	400
macro avg	0.81	0.80	0.80	0.80	400
weighted avg	0.81	0.80	0.80	0.80	400

C= 3
Gamma= 0.1
Accuracy : 85.25

	precision	recall	f1-score	support
0	0.98	0.95	0.92	92
1	0.81	0.82	0.82	96
2	0.76	0.85	0.80	106
3	0.98	0.80	0.88	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.85	0.86	0.85	400
macro avg	0.86	0.85	0.86	0.85	400
weighted avg	0.86	0.85	0.86	0.85	400

C= 5
Gamma= 0.1
Accuracy : 86.0

	precision	recall	f1-score	support
0	0.91	0.92	0.92	92
1	0.81	0.84	0.83	96
2	0.77	0.87	0.82	106
3	0.98	0.81	0.89	106

	accuracy	precision	recall	f1-score	support
accuracy	0.87	0.86	0.86	0.86	400
macro avg	0.87	0.86	0.86	0.86	400
weighted avg	0.87	0.86	0.86	0.86	400

C= 10
Gamma= 0.1
Accuracy : 87.5

	precision	recall	f1-score	support
0	0.89	0.92	0.91	92
1	0.81	0.85	0.83	96
2	0.83	0.87	0.85	106
3	0.98	0.86	0.91	106

	accuracy	precision	recall	f1-score	support
accuracy	0.88	0.88	0.88	0.88	400
macro avg	0.88	0.88	0.88	0.88	400
weighted avg	0.88	0.88	0.88	0.88	400

C= 40
Gamma= 0.1
Accuracy : 87.75

	precision	recall	f1-score	support
0	0.90	0.91	0.91	92
1	0.82	0.83	0.82	96
2	0.83	0.88	0.85	106
3	0.97	0.89	0.93	106

	accuracy	precision	recall	f1-score	support
accuracy	0.88	0.88	0.88	0.88	400
macro avg	0.88	0.88	0.88	0.88	400
weighted avg	0.88	0.88	0.88	0.88	400

C= 60
Gamma= 0.1
Accuracy : 87.0

	precision	recall	f1-score	support
0	0.92	0.90	0.91	92
1	0.80	0.81	0.80	96
2	0.81	0.86	0.83	106
3	0.96	0.91	0.93	106

	accuracy	precision	recall	f1-score	support
accuracy	0.87	0.87	0.87	0.87	400
macro avg	0.87	0.87	0.87	0.87	400
weighted avg	0.87	0.87	0.87	0.87	400

C= 80
Gamma= 0.1
Accuracy : 86.0

	precision	recall	f1-score	support
0	0.93	0.89	0.91	92
1	0.78	0.81	0.80	96
2	0.78	0.86	0.82	106
3	0.97	0.88	0.92	106

	accuracy	precision	recall	f1-score	support
accuracy	0.87	0.86	0.86	0.86	400
macro avg	0.87	0.86	0.86	0.86	400
weighted avg	0.87	0.86	0.86	0.86	400

C= 100
Gamma= 0.1
Accuracy : 86.0

	precision	recall	f1-score	support
0	0.93	0.88	0.91	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.98	0.88	0.93	106

	accuracy	precision	recall	f1-score	support
accuracy	0.87	0.86	0.86	0.86	400
macro avg	0.87	0.86	0.86	0.86	400
weighted avg	0.87	0.86	0.86	0.86	400

C= 1
Gamma= 0.5
Accuracy : 85.75

	precision	recall	f1-score	support
0	0.93	0.87	0.90	92
1	0.76	0.81	0.79	96
2	0.79	0.86	0.82	106
3	0.97	0.89	0.93	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.86	0.86	0.86	400
macro avg	0.86	0.86	0.86	0.86	400
weighted avg	0.86	0.86	0.86	0.86	400

C= 1
Gamma= 0.5
Accuracy : 85.75

	precision	recall	f1-score	support
0	0.93	0.87	0.90	92
1	0.76	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.97	0.89	0.93	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.86	0.86	0.86	400
macro avg	0.86	0.86	0.86	0.86	400
weighted avg	0.86	0.86	0.86	0.86	400

C= 3
Gamma= 0.5
Accuracy : 85.5

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.85	0.86	0.86	400
macro avg	0.86	0.85	0.86	0.86	400
weighted avg	0.86	0.85	0.86	0.86	400

C= 5
Gamma= 0.5
Accuracy : 85.5

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.85	0.86	0.86	400
macro avg	0.86	0.85	0.86	0.86	400
weighted avg	0.86	0.85	0.86	0.86	400

C= 10
Gamma= 0.5
Accuracy : 85.5

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.85	0.86	0.86	400
macro avg	0.86	0.85	0.86	0.86	400
weighted avg	0.86	0.85	0.86	0.86	400

C= 40
Gamma= 0.5
Accuracy : 85.5

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.85	0.86	0.86	400
macro avg	0.86	0.85	0.86	0.86	400
weighted avg	0.86	0.85	0.86	0.86	400

C= 60
Gamma= 0.5
Accuracy : 85.5

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106

	accuracy	precision	recall	f1-score	support
accuracy	0.86	0.85	0.86	0.86	400
macro avg	0.86	0.85	0.86	0.86	400
weighted avg	0.86	0.85	0.86	0.86	400

```

macro avg    0.86    0.85    0.86    400
weighted avg  0.86    0.85    0.86    400

C= 80
Gamma= 0.5
Accuracy : 85.5
precision    recall   f1-score support
0           0.93    0.86    0.89    92
1           0.77    0.81    0.79    96
2           0.79    0.87    0.83    106
3           0.96    0.88    0.92    106

accuracy
macro avg    0.86    0.85    0.86    400
weighted avg  0.86    0.85    0.86    400

C= 100
Gamma= 0.5
Accuracy : 85.5
precision    recall   f1-score support
0           0.93    0.86    0.89    92
1           0.77    0.81    0.79    96
2           0.79    0.87    0.83    106
3           0.96    0.88    0.92    106

accuracy
macro avg    0.86    0.85    0.86    400
weighted avg  0.86    0.85    0.86    400

C= 1
Gamma= 0.1
Accuracy : 80.0
precision    recall   f1-score support
0           0.85    0.96    0.90    92
1           0.74    0.72    0.73    96
2           0.69    0.78    0.73    106
3           0.98    0.75    0.85    106

accuracy
macro avg    0.81    0.80    0.80    400
weighted avg  0.81    0.80    0.80    400

C= 3
Gamma= 0.1
Accuracy : 85.25
precision    recall   f1-score support
0           0.90    0.95    0.92    92
1           0.81    0.82    0.82    96
2           0.76    0.85    0.80    106
3           0.98    0.88    0.88    106

accuracy
macro avg    0.86    0.85    0.86    400
weighted avg  0.86    0.85    0.85    400

C= 5
Gamma= 0.1
Accuracy : 86.0
precision    recall   f1-score support
0           0.91    0.92    0.92    92
1           0.81    0.84    0.83    96
2           0.77    0.87    0.82    106
3           0.98    0.81    0.89    106

accuracy
macro avg    0.87    0.86    0.86    400
weighted avg  0.87    0.86    0.85    400

C= 10
Gamma= 0.1
Accuracy : 87.5
precision    recall   f1-score support
0           0.89    0.92    0.91    92
1           0.81    0.85    0.83    96
2           0.83    0.87    0.85    106
3           0.98    0.86    0.91    106

accuracy
macro avg    0.88    0.88    0.88    400
weighted avg  0.88    0.88    0.88    400

C= 40
Gamma= 0.1
Accuracy : 87.75
precision    recall   f1-score support
0           0.90    0.91    0.91    92
1           0.82    0.83    0.82    96
2           0.83    0.88    0.85    106
3           0.97    0.89    0.93    106

accuracy
macro avg    0.88    0.88    0.88    400
weighted avg  0.88    0.88    0.88    400

C= 60
Gamma= 0.1
Accuracy : 87.0
precision    recall   f1-score support
0           0.92    0.90    0.91    92
1           0.80    0.81    0.80    96
2           0.81    0.86    0.83    106
3           0.96    0.91    0.93    106

accuracy
macro avg    0.87    0.87    0.87    400
weighted avg  0.87    0.87    0.87    400

C= 80
Gamma= 0.1
Accuracy : 86.0
precision    recall   f1-score support
0           0.93    0.89    0.91    92
1           0.78    0.81    0.80    96
2           0.78    0.86    0.82    106
3           0.97    0.88    0.92    106

accuracy
macro avg    0.87    0.86    0.86    400
weighted avg  0.87    0.86    0.86    400

C= 100
Gamma= 0.1
Accuracy : 86.0
precision    recall   f1-score support
0           0.93    0.88    0.91    92
1           0.77    0.81    0.79    96
2           0.79    0.87    0.83    106
3           0.98    0.88    0.93    106

accuracy
macro avg    0.87    0.86    0.86    400
weighted avg  0.87    0.86    0.86    400

C= 1
Gamma= 0.25
Accuracy : 88.25
precision    recall   f1-score support
0           0.89    0.92    0.91    92
1           0.80    0.85    0.83    96
2           0.86    0.86    0.86    106
3           0.98    0.90    0.94    106

accuracy
macro avg    0.88    0.88    0.88    400
weighted avg  0.89    0.88    0.88    400

C= 3
Gamma= 0.25
Accuracy : 86.75
precision    recall   f1-score support
0           0.90    0.91    0.91    92
1           0.81    0.79    0.80    96
2           0.80    0.88    0.84    106
3           0.97    0.89    0.93    106

accuracy
macro avg    0.87    0.87    0.87    400
weighted avg  0.87    0.87    0.87    400

C= 5
Gamma= 0.25
Accuracy : 86.0
precision    recall   f1-score support
0           0.93    0.89    0.91    92
1           0.78    0.81    0.80    96
2           0.78    0.86    0.82    106
3           0.97    0.88    0.92    106

accuracy
macro avg    0.87    0.86    0.86    400
weighted avg  0.87    0.86    0.86    400

```

C= 10	Gamma= 0.25	Accuracy : 85.75	precision	recall	f1-score	support
0	0.93	0.87	0.90	92		
1	0.77	0.80	0.79	96		
2	0.79	0.87	0.83	106		

	3	0.97	0.89	0.93	100
accuracy				0.86	400
macro avg		0.86	0.86	0.86	400
weighted avg		0.86	0.86	0.86	400

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106

3	0.96	0.88	0.92	106
accuracy			0.85	400
macro avg	0.86	0.85	0.86	400
weighted avg	0.86	0.85	0.86	400

```

          0
Gamma= 0.25
Accuracy : 85.5
      precision    recall   f1-score  support
          0       0.93     0.86     0.89      92
          1       0.77     0.81     0.79      96
          2       0.79     0.87     0.83      106

```

	2	0.79	0.87	0.83	100
	3	0.96	0.88	0.92	100
accuracy				0.85	400
macro avg		0.86	0.85	0.86	400
weighted avg		0.86	0.85	0.86	400

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96

2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106
accuracy			0.85	400
macro avg	0.86	0.85	0.86	400
weighted avg	0.86	0.85	0.86	400

```
C= 100
Gamma= 0.25
Accuracy : 85.5
precision    recall   f1-score   support

          0       0.93      0.86      0.89      92
          1       0.77      0.81      0.79      06
```

1	0.77	0.81	0.79	90
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106
accuracy			0.85	400
macro avg	0.86	0.85	0.86	400
weighted avg	0.86	0.85	0.86	400

```
C= 1
Gamma= 0.5
Accuracy : 85.75
precision    recall   f1-score   support
          0       0.93      0.87      0.90      92
```

1	0.76	0.81	0.79	96
2	0.79	0.86	0.82	106
3	0.97	0.89	0.93	106
accuracy			0.86	400
macro avg	0.86	0.86	0.86	400
weighted avg	0.86	0.86	0.86	400

```
C= 3
Gamma= 0.5
Accuracy : 85.5
precision    recall   f1-score   support
          0      0.93     0.86     0.89     92
```

1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106
accuracy			0.85	400
macro avg	0.86	0.85	0.86	400
weighted avg	0.96	0.85	0.96	400

```

weighted avg      0.86      0.85      0.86      460

C= 5
Gamma= 0.5
Accuracy : 85.5
precision      recall      f1-score      support
          1       1.00      1.00      1.00      460

```

0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106
accuracy			0.85	400
macro avg	0.86	0.85	0.86	400

```

weighted avg      0.86      0.85      0.86      400

C= 10
Gamma= 0.5
Accuracy : 85.5
          precision    recall   f1-score  support

```

0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106
accuracy			0.85	400
macro avg	0.86	0.85	0.86	400

```

macro avg      0.86      0.85      0.86      400
weighted avg   0.86      0.85      0.86      400

C= 40
Gamma= 0.5
Accuracy : 85.5

precision      recall      f1-score      support

```

0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106
accuracy			0.85	400

```

macro avg      0.86      0.85      0.86      400
weighted avg   0.86      0.85      0.86      400

C= 60
Gamma= 0.5
Accuracy : 85.5
      precision    recall   f1-score  support

```

0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106
accuracy			0.85	406

	precision	recall	f1-score	support
macro avg	0.86	0.85	0.86	400
weighted avg	0.86	0.85	0.86	400

C= 80
 Gamma= 0.5
 Accuracy : 85.5

	precision	recall	f1-score	support
0	0.93	0.86	0.89	926
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106

```

accuracy      0.85      0.85      0.86      0.86      0.86
macro avg     0.86      0.85      0.86      0.86      0.86
weighted avg  0.86      0.85      0.86      0.86      0.86

C= 100
Gamma= 0.5
Accuracy : 85.5

```

	precision	recall	f1-score	support
0	0.93	0.86	0.89	92
1	0.77	0.81	0.79	96
2	0.79	0.87	0.83	106
3	0.96	0.88	0.92	106

accuracy		0.85	0.85	400
macro avg	0.86	0.85	0.86	400
weighted avg	0.86	0.85	0.86	400

C= 1
Gamma= 1

```

Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

C= 3
Gamma= 1
Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

C= 5
Gamma= 1
Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

C= 10
Gamma= 1
Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

C= 40
Gamma= 1
Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

C= 60
Gamma= 1
Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

C= 80
Gamma= 1
Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

C= 100
Gamma= 1
Accuracy : 85.5
precision    recall   f1-score   support
0           0.93      0.86      0.89      92
1           0.77      0.81      0.79      96
2           0.79      0.87      0.83     106
3           0.96      0.88      0.92     106

accuracy
macro avg       0.86      0.85      0.86     400
weighted avg    0.86      0.85      0.86     400

```

5.

A.

```
In [116]: minimum = mobile_train['battery_power'].min()
maximum = mobile_train['battery_power'].max()

mobile_train_binned = mobile_train.copy()
bins = np.linspace(minimum, maximum, 4)
labels = [0, 1, 2]
mobile_train_binned['battery_power'] = pd.cut(mobile_train['battery_power'], bins=bins, labels=labels, include_lowest=True)
mobile_train_binned.head(10)
```

```
Out[116]:   battery_power  blue  clock_speed  dual_sim  fc  four_g  int_memory  m_dep  mobile_wt  n_cores ... px_height  px_width  ram  sc_h  sc_w  talk_time  three_g  touch_screen  wifi  price_range
0          0         0      2.2        0   1       0      7   0.6     188     2 ...     20    756  2549     9    7   19      0      0     1      1
1          1         1      0.5        1   0       1     53   0.7     136     3 ...    905   1988  2631    17    3    7   1     1     0     2
2          0         1      0.5        1   2       1     41   0.9     145     5 ...   1263   1716  2603    11    2    9   1     1     0     2
3          0         1      2.5        0   0       0     10   0.8     131     6 ...   1216   1786  2769    16    8   11   1     0     0     2
4          2         1      1.2        0   13      1     44   0.6     141     2 ...   1208   1212  1411     8   2    15   1     1     0     1
5          2         0      0.5        1   3       0     22   0.7     164     1 ...   1000   1654  1067    17    1   10   1     0     0     1
6          2         0      1.7        0   4       1     10   0.8     139     8 ...   381   1018  3220    13    8   18   1     0     1     3
7          2         0      0.5        1   0       0     24   0.8     187     4 ...   512   1149  700     16    3    5   1     1     1     0
8          1         1      0.5        0   0       0     53   0.7     174     7 ...   386   836  1099    17    1   20   1     0     0     0
9          0         1      0.6        1   2       1      9   0.1     93     5 ...  1137   1224  513    19    10   12   1     0     0     0
```

10 rows × 21 columns

B.

One-hot encoding transforms categorical features to a format that works better with classification and regression algorithms. It's very useful in methods where multiple types of data representation is necessary.

For example, some vectors may be optimal for regression (approximating functions based on former return values), and some may be optimal for classification (categorization into fixed sets/classes, typically binary):

```
In [65]: from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import StandardScaler

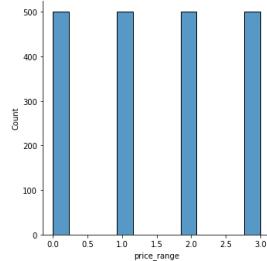
#Normalizing the data
sc = StandardScaler()
X=mobile_train.iloc[:, :-1].values
mobile_train.iloc[:, 20:21].values
X = sc.fit_transform(X)
# One Hot Encoder
ohe=OneHotEncoder()
y=ohe.fit_transform(y).toarray()
y
```

```
Out[65]: array([[0., 1., 0., 0.],
 [0., 0., 1., 0.],
 [0., 0., 1., 0.],
 ...,
 [0., 0., 0., 1.],
 [1., 0., 0., 0.],
 [0., 0., 0., 1.]])
```

C.

```
In [93]: # Check Distribution of price range
sns.displot(mobile_train['price_range'])
```

```
Out[93]: <seaborn.axisgrid.FacetGrid at 0x7f0b8e407c88>
```



price_range is normally distributed! So no need to transformation.

The log transform is a specific example of a family of transformations known as power transforms. In statistical terms, these are variance-stabilizing transformations. We can apply a power transform directly by calculating the log or square root of the variable, although this may or may not be the best power transform for a given variable.

There are two popular approaches for such automatic power transforms; they are:

```
Box-Cox Transform  
Yeo-Johnson Transform
```

We can use this methods when data is skewed.

but there is no skewed feature:

```
In [94]: skew_features = mobile_train[num_feets].skew().sort_values(ascending=False)  
skewness = pd.DataFrame({'Skew': skew_features})  
skewness
```

```
Out[94]: Skew  
fc    1.019811  
px_height    0.666271  
sc_w    0.633787  
clock_speed    0.178084  
m_dep    0.089082  
int_memory    0.057889  
battery_power    0.031898  
blue    0.020016  
pc    0.017306  
px_width    0.014787  
talk_time    0.009512  
ram    0.006628  
mobile_wt    0.006558  
n_cores    0.003628  
price_range    0.000000  
touch_screen    -0.012009  
wifl    -0.028024  
dual_sim    -0.038035  
four_g    -0.086144  
sc_h    -0.098884  
three_g    -1.228142
```

C.

sc_h: Screen Height of mobile in cm

sc_w: Screen Width of mobile in cm

m_dep: Mobile Depth in cm

```
In [148]: mobile_train_extended = mobile_train.copy()  
mobile_train_extended['aria'] = mobile_train['sc_h'] * mobile_train['sc_w']  
mobile_train_extended['aria']
```

```
Out[148]: 0      63  
1      51  
2      22  
3     128  
4      16  
..  
1995     92  
1996    110  
1997      9  
1998    180  
1999     76  
Name: aria, Length: 2000, dtype: int64
```

```
In [149]: mobile_train_extended['volume'] = mobile_train['sc_h'] * mobile_train['sc_w'] * mobile_train['m_dep']  
mobile_train_extended['volume']
```

```
Out[149]: 0      37.8  
1      35.7  
2      19.8  
3     102.4  
4      9.6  
..  
1995     41.6  
1996    22.0  
1997     6.3  
1998    18.0  
1999     68.4  
Name: volume, Length: 2000, dtype: float64
```

```
In [150]: mobile_train_extended
```

```
Out[150]:   battery_power  blue  clock_speed  dual_sim  fc  four_g  int_memory  m_dep  mobile_wt  n_cores ... ram  sc_h  sc_w  talk_time  three_g  touch_screen  wifl  price_range  aria  volume  
0          842     0       2.2      0   1     0      7   0.6    188     2 ... 2549    9   7    19     0     0   1     1   63   37.8  
1         1021     1       0.5      1   0     1     53   0.7    136     3 ... 2631   17   3    7    1     1   0     2   51   35.7  
2          563     1       0.5      1   2     1     41   0.9    145     5 ... 2603   11   2    9    1     1   0     2   22   19.8  
3          615     1       2.5      0   0     0     10   0.8    131     6 ... 2769   16   8   11    1     0   0     2   128  102.4  
4         1821     1       1.2      0  13     1     44   0.6    141     2 ... 1411    8   2    15    1     1   0     1   16    9.6  
..  
1995        794     1       0.5      1   0     1     2   0.8    106     6 ... 668   13   4    19    1     1   0     0   52   41.6  
1996        1965     1       2.6      1   0     0     39   0.2    187     4 ... 2032   11   10   16    1     1   1     2   110  22.0  
1997        1911     0       0.9      1   1     1     36   0.7    108     8 ... 3057    9   1    5    1     1   0     3   9     6.3  
1998        1512     0       0.9      0   4     1     46   0.1    145     5 ... 869   18   10   19    1     1   1     0   180  18.0  
1999        510      1       2.0      1   5     1     45   0.9    168     6 ... 3919   19   4    2    1     1   1     3   76   68.4
```

2000 rows x 23 columns

6.

```
In [151]: y = mobile_train.binned['price_range'].values  
x_data=mobile_train_binned.drop(['price_range'],axis=1)  
x_train, x_test, y_train, y_test = train_test_split(x_data,y,test_size = 0.2,random_state=1)  
  
svc=SVC(random_state=1)  
svc.fit(x_train,y_train)  
svc_pred = svc.predict(x_test)  
svc_acc_score = accuracy_score(y_test, svc_pred)  
accuracy = accuracy_score(y_test, svc_pred)  
  
print("binning method on 'battery power'\n")  
print("Accuracy : ", accuracy * 100)  
print(classification_report(y_test,svc_pred))  
  
binning method on 'battery power'  
Accuracy : 76.25  
precision    recall   f1-score   support  
          0.88      0.84      0.86      0.9
```

```

    1     0.67    0.75    0.71    96
    2     0.68    0.67    0.67    106
    3     0.86    0.88    0.83    106

accuracy          0.76    400
macro avg       0.77    0.76    0.77    400
weighted avg    0.77    0.76    0.76    400

```

```
In [152]:-
y = mobile_train_extended["price_range"].values
x_data=mobile_train_extended.drop(["price_range"],axis=1)
x_train, x_test, y_train, y_test = train_test_split(x_data,y,test_size = 0.2,random_state=1)

svc=SVC(random_state=1)
svc.fit(x_train,y_train)
svc_pred = svc.predict(x_test)
svc_acc_score = accuracy_score(y_test, svc_pred)
accuracy = accuracy_score(y_test, svc_pred)

print("SVC on extended dataset\n")
print("Accuracy : ", accuracy * 100)
print(classification_report(y_test,svc_pred))

SVC on extended dataset
```

```

Accuracy : 94.5
precision    recall   f1-score   support
0            0.97    0.99    0.98    92
1            0.92    0.95    0.93    96
2            0.92    0.91    0.91    106
3            0.97    0.94    0.96    106

accuracy          0.94    400
macro avg       0.95    0.95    0.95    400
weighted avg    0.95    0.94    0.94    400

```

```
In [153]:-
mobile_train_combined = mobile_train_extended
mobile_train_combined['battery_power'] = mobile_train_binned['battery_power']
mobile_train_combined
```

```
Out[153]:-
battery_power bluo clock_speed dual_sim fc four_g int_memory m_dep mobile_wt n_cores ... ram sc_h sc_w talk_time three_g touch_screen wifi price_range aria volume
0             0     0      2.2     0   1     0      7   0.6    188    2 ... 2549   9   7    19   0      0   1     1   63   37.8
1             1     1      0.5     1   0     1     53   0.7    136    3 ... 2631  17   3    7   1      1   0     2   51   35.7
2             0     1      0.5     1   2     1     41   0.9    145    5 ... 2603  11   2    9   1      1   0     2   22   19.8
3             0     1      2.5     0   0     0     10   0.8    131    6 ... 2769  16   8   11   1      0   0     2   128  102.4
4             2     1      1.2     0  13     1     44   0.6    141    2 ... 1411   8   2    15   1      1   0     1   16   9.6
...
1095           0     1      0.5     1   0     1     2   0.8    106    6 ... 668   13   4    19   1      1   0     0   52   41.6
1096           2     1      2.6     1   0     0     39   0.2    187    4 ... 2032  11   10   16   1      1   1     2   110  22.0
1097           2     0      0.9     1   1     1     36   0.7    108    8 ... 3057   9   1    5   1      1   0     3   9    6.3
1098           2     0      0.9     0   4     1     46   0.1    145    5 ... 869   18   10   19   1      1   1     0   180  18.0
1099           0     1      2.0     1   5     1     45   0.9    168    6 ... 3919  19   4    2   1      1   1     3   76   68.4
```

2000 rows x 23 columns

```
In [154]:-
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

y = mobile_train_combined["price_range"].values
x_data=mobile_train_combined.drop(["price_range"],axis=1)
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=1)

svc=SVC(random_state=1)
svc.fit(x_train,y_train)
svc_pred = svc.predict(x_test)
svc_acc_score = accuracy_score(y_test, svc_pred)
accuracy = accuracy_score(y_test, svc_pred)

print("SVC on combined dataset\n")
print("Accuracy : ", accuracy * 100)
print(classification_report(y_test,svc_pred))

SVC on combined dataset
```

```

Accuracy : 83.75
precision    recall   f1-score   support
0            0.91    0.93    0.92    92
1            0.77    0.78    0.77    96
2            0.74    0.81    0.77    106
3            0.97    0.83    0.89    106

accuracy          0.84    400
macro avg       0.84    0.84    0.84    400
weighted avg    0.84    0.84    0.84    400

```

So, Best Accuracy is %94.5 for extended dataset.

Best Regards.