آتوسا مالمیر چگینی

97106251

گزارش پروژه ی بردار ماشین پشتیبان

*** در مورد هر بخش ابتدا هدف آن بخش ، سپس توضیح خلاصه ای از کد و در نهایت آزمایشاتی برای بررسی تاثیر مقادیر مختلف برای پارامتر ها آورده شده است .

بخش اول :

هدف :

در این بخش باید ابتدا نقاطی دلخواه را بر روی صفحه ی مختصات ایجاد کرده و سپس از آن جایی که مسئله ، دسته بندی دو کلاسه است ؛ هر یک از نقاط ایجاد شده را به صورت رندم در یکی از دو کلاس 0 یا 1 قرار میدهیم . حال داده ها را به دو دسته ی آموزشی و آزمایشی تقسیم میکنیم . در آخر با استفاده از SVM Classifier در کتابخانه ی Sklearn اقدام به یادگیری چگونگی دسته بندی شدن داده های آموزشی میکنیم و وقتی آموزش SVM تمام شد ، مدل ایجاد شده را بر روی داده های آزمایشی تست میکنیم و دقت به دست آمده را گزارش میکنیم .

توضیح کد :

- انتخاب dataSet

به طور کلی در این بخش 3 dataSet انتخاب کردم که با استفاده از توابع آماده ی make_moons و make_circles و تابعی دیگر که نقاط دو کلاس را به گونه ای در صفحه قرار میدهد که به صورت خطی جدا پذیر باشد ؛ ایجاد شده اند .

- Classifiers

هر کدام از dataset های بالا را با 4 نوع SVM کلاس بندی میکنیم که تفاوت این 4 نوع در هسته ی آن ها است . این 4 تا SVM به ترتیب دارای هسته های linear و RBFو polyو sigmoid هستند .

در ابتدای کد یک متغیر h تعریف کردیم که دقت محور های مختصات را نشان میدهد .(step size)

سپس یک آرایه از نام Classifier ها ساختیم که بعدا در نشان دادن نتایج با plot بتوانیم از آن استفاده کنیم .

بعد آرایه ی Classifer هایمان را ایجاد کردیم که شامل همان 4 Classifier ای است که بالا تر آن را توضیح دادیم .

در چهار خط بعد یعنی :

```
X, y = make_classification(n_features=2, n_redundant=0, n_informative=2,
                           random_state=1, n_clusters_per_class=1,
                           n_samples=400)
rng = np.random.RandomState(2)
X += rng.uniform(size=X.shape)
linearly_separable = (X, y)
```

نقاطی (x, y) ایجاد میکنیم که به صورت خطی جدا پذیر باشند و این نقاط یکی از سه dataset ای هستند که بالاتر آن را توضیح دادم .

سپس آرایه ی dataset هارا ایجاد میکنیم که شامل 3 dataSet متفاوت است .

در خط بعدی یک پنجره ایجاد میکنیم که بعدا نتایج کلاس بندی های متفاوت را روی آن نشان دهیم .

```
figure = plt.figure(figsize=(27, 9))
```

حالا برای هر dataset تمام 4 نوع Classifier را بر روی آن امتحان میکنیم و دقت هر کدام را هم خروجی میدهیم تا بتوانیم بررسی کنیم که کدام هسته بهتر عمل کرده است .

پس به ازای هر dataset ابتدا نقاط آن را به دو دسته ی آموزشی و آزمایشی تقسیم میکنیم .(توسط تابع train_test_split)

در سه خط بعدی تنها مقیاس بندی برای محور های مختصات را انجام داده ایم :

```
x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))
```

در کد بالا مثلا xx نشان دهنده ی این است که محور x ها از x_min تا x_max را دارد و هر گام به اندازه ی h است .

از خط 44 تا 54 صفحه را به 15 قسمت تقسیم کردیم که 3 سطر و 5 ستون دارد و در هر سطر اولی مربوط به داده ها است . و 4 تای بعدی مربوط به 4 جدا کننده هستند .

نکته ی دیگر این است که داده های آموزشی به پررنگ و داده های آزمایشی کم رنگ هستند .

(داده های کلاس اول هم به رنگ سبز و داده های کلاس دوم به رنگ بنفش نشان داده شده اند .)

حالا حلقه ی دیگری داریم که SVM ها را بر روی dataset اجرا میکند .

در خط زیر یک مدل میسازیم که داده های آموزشی را یاد میگیرد :

```
clf.fit(X_train, y_train)
```

در خط بعد دقت مدل ایجاد شده بر روی داده های آزمایشی بررسی میشود .

```
score = clf.score(X_test, y_test)
```

در خطوط زیر هم خط (یا منحنی) جدا کننده که توسط SVM ایجاد شده است را رسم میکنیم .

```
if hasattr(clf, "decision_function"):
    Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
else:
    Z = clf.predict_proba(np.c_[xx.ravel(), yy.ravel()])[:, 1]
Z = Z.reshape(xx.shape)
ax.contourf(xx, yy, Z, cmap=cm, alpha=.8)
```

در خطوط زیر نقاط را هم اضافه میکنیم که بتوان دید که مدل با چه دقتی توانسته است داده های دو کلاس را جدا کند .

```
# Plot also the training points
ax.scatter(X_train[:, 0], X_train[:, 1], c=y_train, cmap=cm_bright)
# and testing points
ax.scatter(X_test[:, 0], X_test[:, 1], c=y_test, cmap=cm_bright,
           alpha=0.6)
```

در خط زیر هم دقت مدل در داده های آزمایشی را به ازای هر Classifier در زیر نمودار مربوط به آن مینویسیم :

```
ax.text(xx.max() - .3, yy.min() + .3, ('%.2f' % score).lstrip('0'),
        size=15, horizontalalignment='right')
```

در خطوط بعدی هم تعدادی عدد چاپ میشود که گزارشی از نحوه ی عملکرد Classifier است .

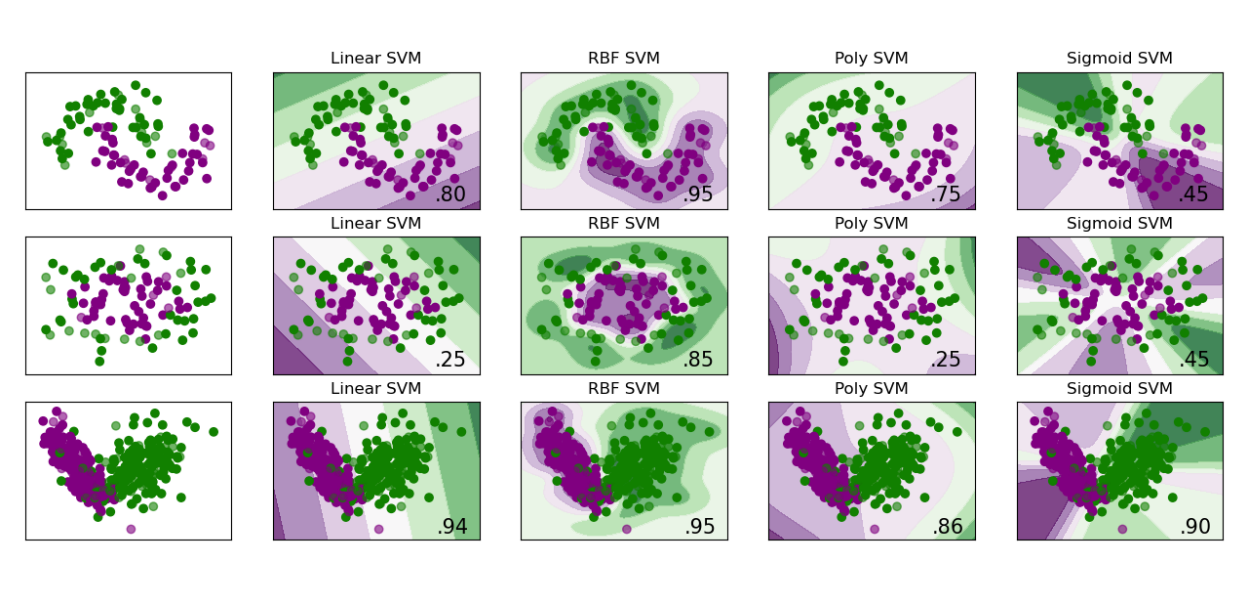
بررسی تاثیر پارامتر های مختلف :

۱) درصد داده های آموزشی و آزمایشی :

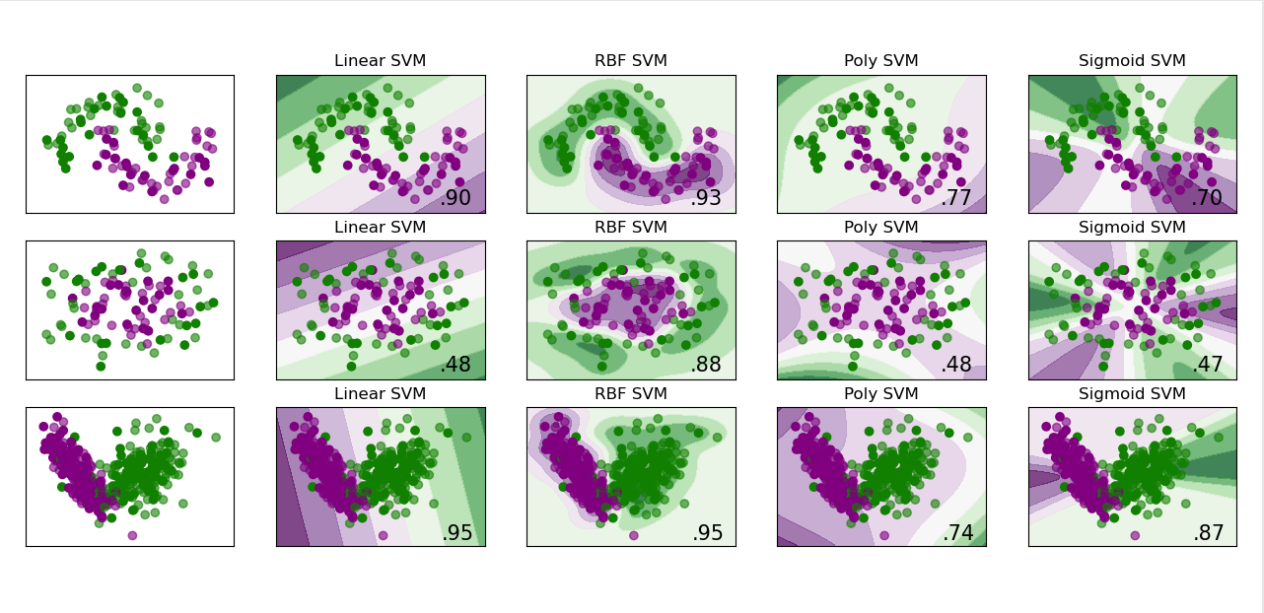
این پارامتر را در بخش زیر تنظیم میکنیم :

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3)
```

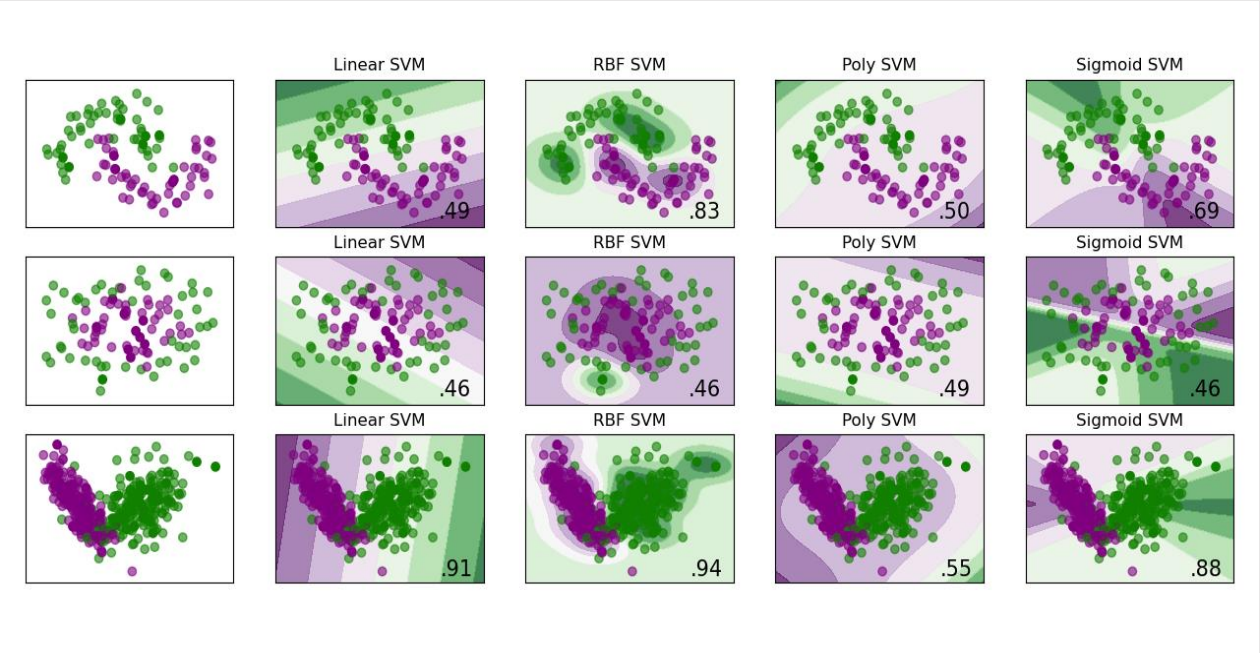
Test = 0.1 / train = 0.9 :



Test = 0.6 / train = 0.4 :



Test = 0.9 / train = 0.1 :

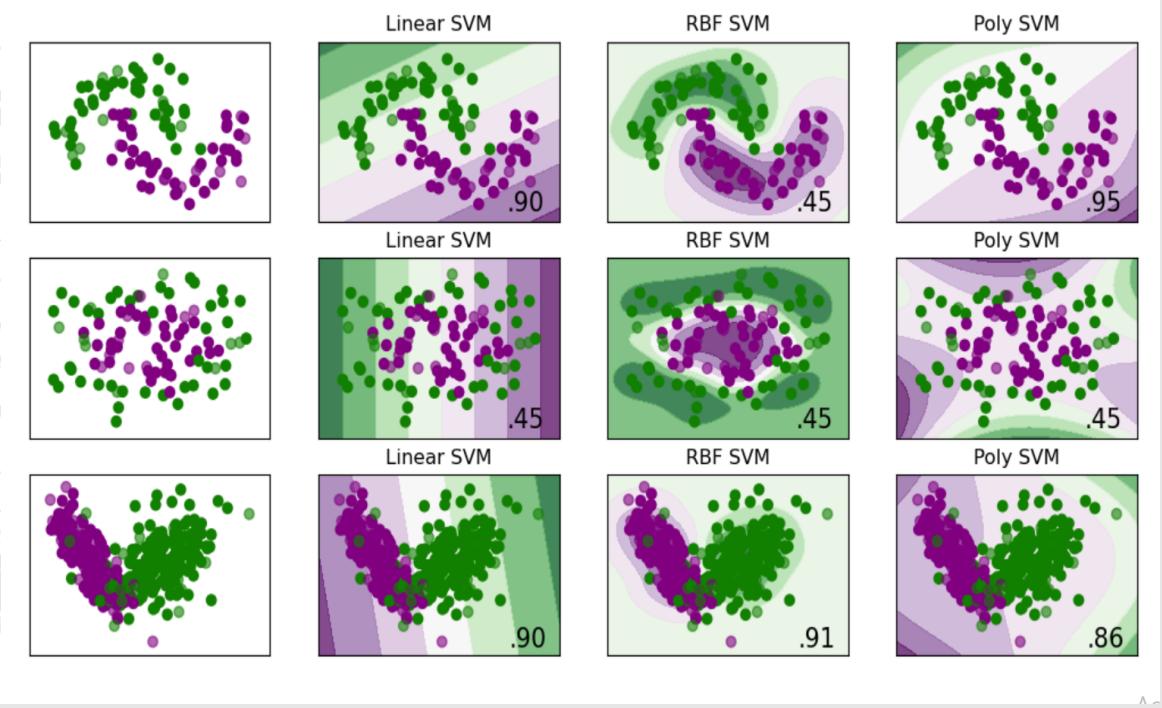


*** در سه آزمایش بالا درصد داده های تست را به تدریج بیشتر کردیم و نتیجه ی جالبی که از این آزمایش ها به دست آمد این است که به طور کلی با بیشتر شدن داده های آموزشی دقت باید بیشتر شود اما میبینیم اگر داده های تست خیلی کم باشند (مثل آزمایش اول) ممکن است دقت خیلی کم شود و دلیل آن هم این است که اگر تعداد داده های تست خیلی کم شود و حتی یکی از آن ها اشتباه کلاس بندی شوند در درصد دقت تاثیر زیادی میگذارد .(در این جا هم تعداد داده ها چون کم است پس این اتفاق افتاده است .)

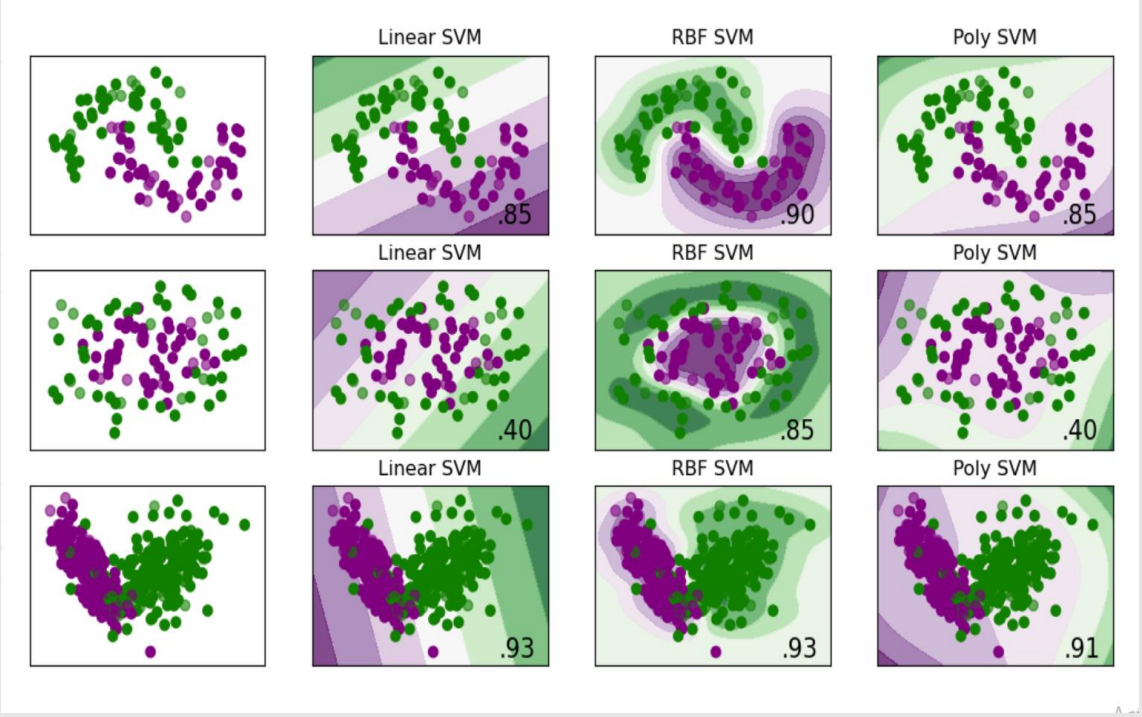
۲) تغییر پارامتر C در Classifier ها :

*** به نظر میرسد تغییر پارامتر C فقط برای Classifier های linear و RBF و poly است .

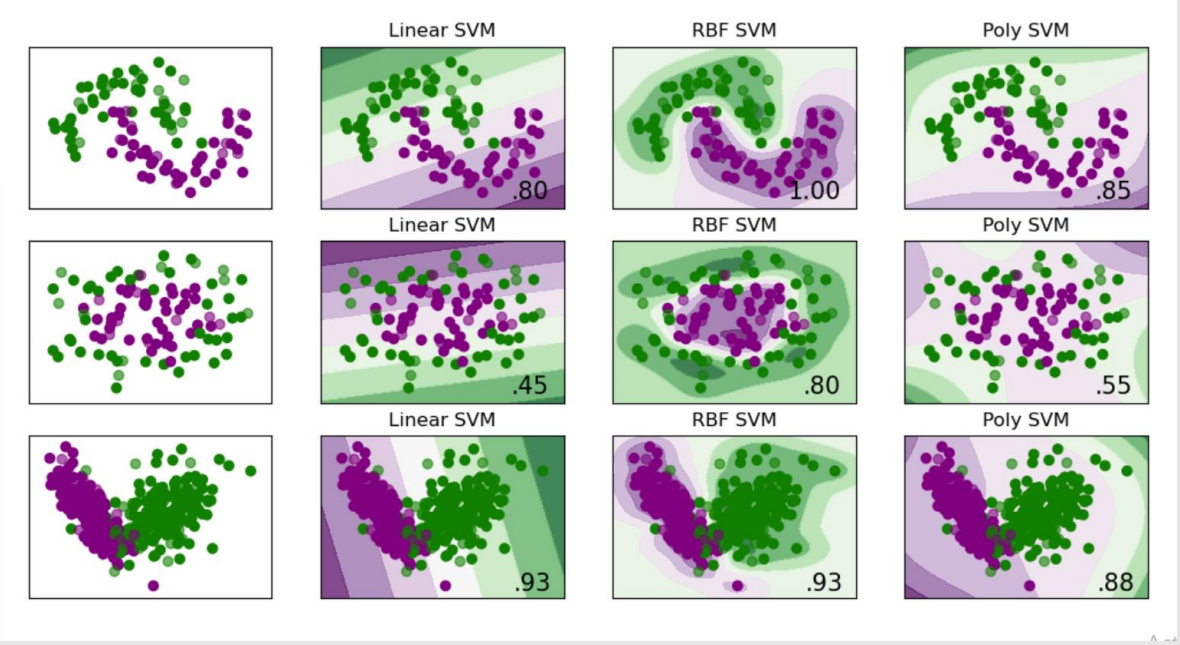
C = 0.03 :


Linear SVM
RBF SVM
Poly SVM
.90
.45
.95
Linear SVM
RBF SVM
Poly SVM
.45
.45
.45
Linear SVM
RBF SVM
Poly SVM
.90
.91
.86

C = 0.3 :

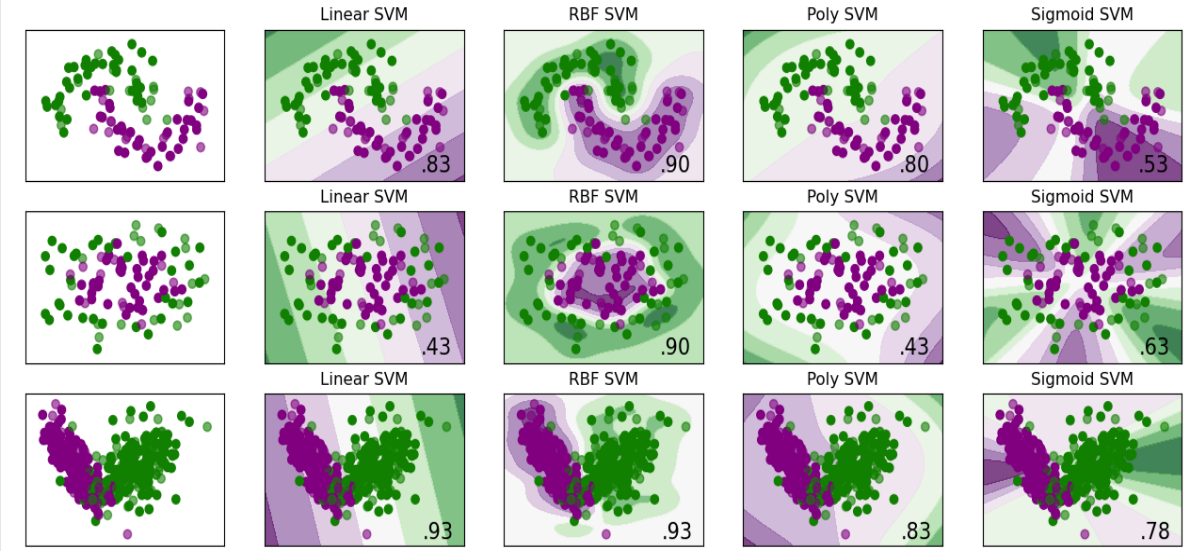

Linear SVM
RBF SVM
Poly SVM
.85
.90
.85
Linear SVM
RBF SVM
Poly SVM
.40
.85
.40
Linear SVM
RBF SVM
Poly SVM
.93
.93
.91

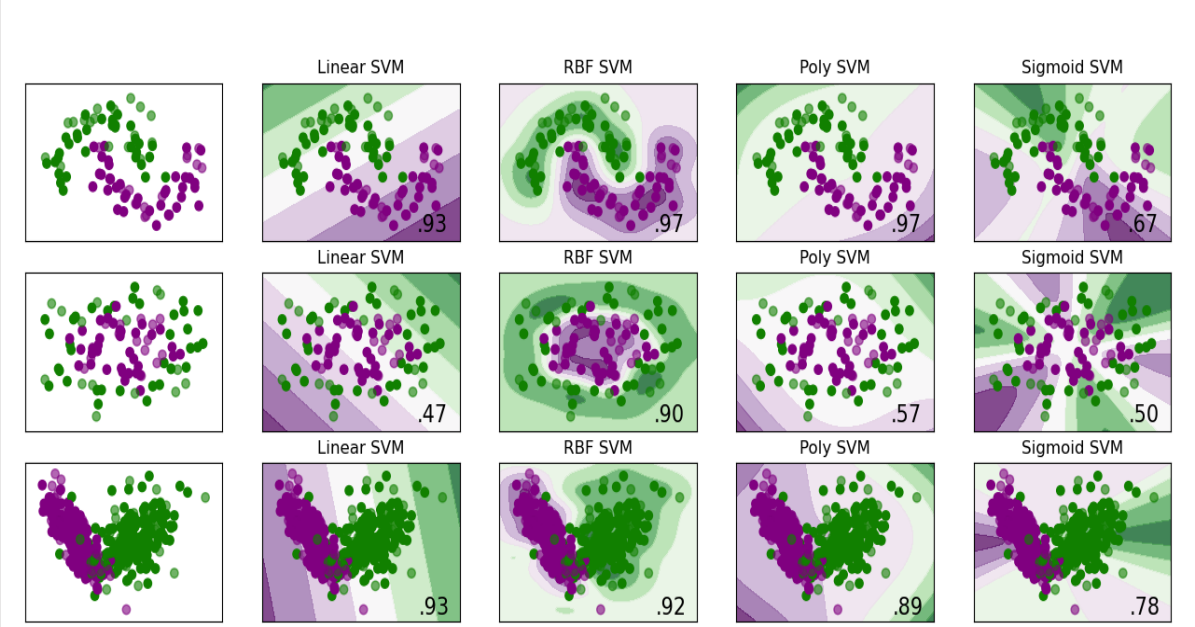
C = 0.7 :



❊❊❊ به نظر میرسد به طور کلی با زیاد کردن C معمولا دقت بیشتر میشود البته از یک جایی به بعد زیاد کردن C تاثیری در بیشتر شدن دقت SVM در داده های آزمایشی ندارد .

۳) بررسی دقت SVM در dataset 3 :





*** با توجه به دو آزمایش بالا و آزمایش های دیگر که قبلا انجام داده بودیم میتوان متوجه شد که به طور کلی دقت SVM های مختلف در داده های dataset سوم (یعنی داده هایی که به تقریبا به صورت خطی جدا پذیر هستند.) بیشتر است .پس از آن معمولا دقت در داده های moon یعنی dataset اول بیشتر است و dataset سوم کم ترین دقت را دارد .

۴) بررسی دقت چهار Classifier (هسته های مختلف) و مقایسه ی آن ها با هم :

تا این جای کار 8 آزمایش انجام دادیم و متوجه میشویم موارد زیر از آزمایشات باللا نتیجه میشوند :

1) در تمامی dataset ها معمولا SVM با هسته ی RBF خیلی دقیق تر و بهتر عمل کرده است .
2) SVM با هسته ی linear معمولا در dataset دوم یعنی داده هایی که به صورت دایره هایی در هم تنیده شده اند خوب عمل نمیکند و دقت خیلی پایینی دارد اما در داده های سومی چون تقریبا به صورت خطی جدا پذیر هستند عملکرد خیلی خوبی دارد .
3) به طور کلی به نظر میرسد کلاس بندی داده های دومی برای SVM نسبت به بقیه ی dataset ها سخت تر است و دقت داده های آموزشی برای این dataset از بقیه به طور قابل ملاحظه ای کم تر است .
4) SVM با هسته ی poly برای هر دو dataset اول و سوم معمولا دقت خوبی به دست می دهد .

بخش دوم :

هدف :

در این جا نیز مانند بخش اول ، هدف کلاس بندی کردن داده ها با استفاده از SVM Classifier است . با این تفاوت که در این بخش تعداد کلاس ها بیشتر است و مسئله از دسته بندی دو کلاسه به multi classifier تبدیل میشود .

توضیح کد :

در ابتدا عکس های اعداد را در متغیری به نام image_dataset ذخیره میکنیم . برای این کار از یک تابع استفاده میکنیم که تعدادی خروجی دارد که مهم ترین آن ها data و target است که data همان X ها هستند(عکس ها)و target ها نشان دهنده ی این است که هر عکس در واقع چه عددی را نشان میدهد . (برای مشخص کردن این موضوع هم تمام عکس های مربوط به هر عدد را در پوشه ای با نام همان عدد قرار دادم).

*** این تابع دو واقع همان تابع استفاده شده در پروژه ی شبکه ی عصبی است .

سپس با استفاده از train_test_split دیتاست load شده را به دو دسته ی train و test تقسیم میکنیم . در این تابع یک پارامتر test_size داریم که با استفاده از آن میتوانیم درصد داده های train و test را تغییر دهیم . سپس مانند بخش اول یک آرایه به نام Classifiers می سازیم که در آن 4 تا SVM داریم که 4 نوع هسته ی متفاوت دارند .(هسته ها به ترتیب مانند بخش اول هستند .)

سپس یک حلقه داریم که به ازای هر classifier ابتدا مدل را با داده های آموزشی ، آموزش میدهد :

```
clf.fit(X_train, y_train)
```

و بعد دقت مدل به دست آمده را داده های آزمایشی بررسی میکند :

```
score = clf.score(X_test, y_test)
```

در نهایت هم تعدادی عدد چاپ میکنیم که گزارشی از نحوه ی عملکرد classifier های مختلف روی داده های تصویری را نشان میدهند .

بررسی تاثیر پارامتر های مختلف :

1) بررسی دقت چهار Classifier (هسته های مختلف) و مقایسه ی آن ها با هم :

```
Linear SVM
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       367
           1       0.99      1.00      0.99       289
           2       0.90      0.97      0.93       201
           3       0.96      0.91      0.94       223
           4       0.95      0.93      0.94       194
           5       0.91      0.90      0.90       162
           6       0.99      0.98      0.98       206
           7       0.94      0.97      0.95       194
           8       0.94      0.87      0.90       156
           9       0.98      0.93      0.96       196

    accuracy                           0.95      2188
   macro avg       0.95      0.95      0.95      2188
weighted avg       0.95      0.95      0.95      2188

Accuracy: 0.953382084095064
```

```
RBF SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python3
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.17      1.00      0.29       367
           1       1.00      0.00      0.01       289
           2       0.00      0.00      0.00       201
           3       0.00      0.00      0.00       223
           4       0.00      0.00      0.00       194
           5       0.00      0.00      0.00       162
           6       0.00      0.00      0.00       206
           7       0.00      0.00      0.00       194
           8       0.00      0.00      0.00       156
           9       0.00      0.00      0.00       196

    accuracy                           0.17      2188
   macro avg       0.12      0.10      0.03      2188
weighted avg       0.16      0.17      0.05      2188

Accuracy: 0.16819012797074953
```

```
Poly SVM
              precision    recall  f1-score   support

           0       0.98      0.95      0.97       367
           1       0.95      1.00      0.97       289
           2       0.93      0.85      0.89       201
           3       0.95      0.86      0.90       223
           4       0.55      0.94      0.69       194
           5       0.93      0.81      0.87       162
           6       0.96      0.89      0.93       206
           7       0.94      0.85      0.89       194
           8       0.92      0.85      0.88       156
           9       0.96      0.79      0.87       196

    accuracy                           0.89      2188
   macro avg       0.91      0.88      0.89      2188
weighted avg       0.92      0.89      0.90      2188

Accuracy: 0.8907678244972578
```

```
Sigmoid SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.17      1.00      0.29       367
           1       0.00      0.00      0.00       289
           2       0.00      0.00      0.00       201
           3       0.00      0.00      0.00       223
           4       0.00      0.00      0.00       194
           5       0.00      0.00      0.00       162
           6       0.00      0.00      0.00       206
           7       0.00      0.00      0.00       194
           8       0.00      0.00      0.00       156
           9       0.00      0.00      0.00       196

    accuracy                           0.17      2188
   macro avg       0.02      0.10      0.03      2188
weighted avg       0.03      0.17      0.05      2188

Accuracy: 0.16773308957952468
```

*** با توجه به آزمایش بالا به نظر میرود با شرایط یکسان (مثلا تعداد داده هاس آموزشی و آزمایشی) هسته های Linear و Poly خیلی بهتر از هسته های Sigmoid و RBF عمل میکنند .

2) درصد داده های آموزشی و آزمایشی :

Test = 0.1 / train = 0.9 :

```
Linear SVM
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       122
           1       0.99      1.00      0.99        94
           2       0.93      0.95      0.94        65
           3       0.99      0.93      0.96        81
           4       0.94      0.92      0.93        66
           5       0.94      0.94      0.94        48
           6       0.97      0.97      0.97        62
           7       0.99      0.96      0.97        71
           8       0.95      0.95      0.95        59
           9       0.95      0.97      0.96        62

    accuracy                           0.96       730
   macro avg       0.96      0.96      0.96       730
weighted avg       0.96      0.96      0.96       730

Accuracy: 0.963013698630137
```

```
RBF SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python38
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.17      1.00      0.29       122
           1       1.00      0.15      0.26        94
           2       0.00      0.00      0.00        65
           3       0.00      0.00      0.00        81
           4       0.00      0.00      0.00        66
           5       0.00      0.00      0.00        48
           6       0.00      0.00      0.00        62
           7       0.00      0.00      0.00        71
           8       0.00      0.00      0.00        59
           9       0.00      0.00      0.00        62

    accuracy                           0.19       730
   macro avg       0.12      0.11      0.06       730
weighted avg       0.16      0.19      0.08       730

Accuracy: 0.1863013698630137
```

```
Poly SVM
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       122
           1       0.97      1.00      0.98        94
           2       0.94      0.98      0.96        65
           3       1.00      0.96      0.98        81
           4       0.88      0.91      0.90        66
           5       0.98      0.96      0.97        48
           6       0.95      0.95      0.95        62
           7       1.00      0.94      0.97        71
           8       0.98      0.98      0.98        59
           9       0.97      0.95      0.96        62

    accuracy                           0.97       730
   macro avg       0.97      0.96      0.97       730
weighted avg       0.97      0.97      0.97       730

Accuracy: 0.9684931506849315
```

```
Sigmoid SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python38\l
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.17      1.00      0.29       122
           1       0.00      0.00      0.00        94
           2       0.00      0.00      0.00        65
           3       0.00      0.00      0.00        81
           4       0.00      0.00      0.00        66
           5       0.00      0.00      0.00        48
           6       0.00      0.00      0.00        62
           7       0.00      0.00      0.00        71
           8       0.00      0.00      0.00        59
           9       0.00      0.00      0.00        62

    accuracy                           0.17       730
   macro avg       0.02      0.10      0.03       730
weighted avg       0.03      0.17      0.05       730

Accuracy: 0.16712328767123288

Process finished with exit code 0
```

Test = 0.35 / train = 0.65 :

```
Linear SVM
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       433
           1       0.99      1.00      0.99       348
           2       0.87      0.94      0.90       238
           3       0.96      0.91      0.94       270
           4       0.94      0.91      0.93       231
           5       0.88      0.90      0.89       185
           6       0.97      0.97      0.97       227
           7       0.95      0.96      0.96       221
           8       0.94      0.87      0.90       182
           9       0.97      0.95      0.96       217

    accuracy                           0.95      2552
   macro avg       0.94      0.94      0.94      2552
weighted avg       0.95      0.95      0.95      2552

Accuracy: 0.9486677115987461
```

```
RBF SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python38\l
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.17      1.00      0.29       433
           1       1.00      0.11      0.20       348
           2       0.00      0.00      0.00       238
           3       0.00      0.00      0.00       270
           4       0.00      0.00      0.00       231
           5       0.00      0.00      0.00       185
           6       0.00      0.00      0.00       227
           7       0.00      0.00      0.00       221
           8       0.00      0.00      0.00       182
           9       0.00      0.00      0.00       217

    accuracy                           0.18      2552
   macro avg       0.12      0.11      0.05      2552
weighted avg       0.17      0.18      0.08      2552

Accuracy: 0.18456112852664577
```

```
Poly SVM
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       433
           1       0.98      0.99      0.99       348
           2       0.93      0.94      0.93       238
           3       0.96      0.92      0.94       270
           4       0.81      0.94      0.87       231
           5       0.93      0.91      0.92       185
           6       0.98      0.95      0.96       227
           7       0.96      0.95      0.95       221
           8       0.94      0.89      0.91       182
           9       0.99      0.92      0.95       217

    accuracy                           0.95      2552
   macro avg       0.94      0.94      0.94      2552
weighted avg       0.95      0.95      0.95      2552

Accuracy: 0.9463166144200627
```

```
Sigmoid SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python38
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.17      1.00      0.29       433
           1       0.00      0.00      0.00       348
           2       0.00      0.00      0.00       238
           3       0.00      0.00      0.00       270
           4       0.00      0.00      0.00       231
           5       0.00      0.00      0.00       185
           6       0.00      0.00      0.00       227
           7       0.00      0.00      0.00       221
           8       0.00      0.00      0.00       182
           9       0.00      0.00      0.00       217

    accuracy                           0.17      2552
   macro avg       0.02      0.10      0.03      2552
weighted avg       0.03      0.17      0.05      2552

Accuracy: 0.16967084639498434
```

*** همان طور که از آزمایشات بالا مشخص است اگر درصد داده های تست بیشتر شود (از 0.1 به 0.35) دقت SVM در کلاس بندی داده ها کم تر میشود . برای مثال در Linear SVM دقت از 0.963 به 0.948 و در RBF SVM دقت از 0.186 به 0.184 و در Poly SVM دقت از 0.968 به 0.946 کاهش پیدا کرده است . البته چون تعداد داده ها در این dataset بسیار زیاد است تاثیر بیشتر شدن داده های آزمایشی نسبت به داده های آموزشی در دقت ها چنداد مشخص نیست .(در Sigmoid SVM دقت 0.002 بیشتر هم شده است که میتوانیم از آن صرف نظر کنیم .)

- در بخش های بعدی آزمایشات تغییر پارامتر ها را فقط بر روی Linear و Poly انجام میدهیم ؛ چون همان طور که در بخش اول نشان دادیم دقت Sigmoid و RBF بسیار کم است و بنابراین آن ها را کنار میگذاریم .

3) تاثیر پارامتر C بر Linear SVM :

C = 0.03 :

Linear SVM

	precision	recall	f1-score	support
0	0.97	0.99	0.98	433
1	0.99	1.00	0.99	348
2	0.88	0.95	0.91	238
3	0.97	0.92	0.94	270
4	0.95	0.93	0.94	231
5	0.89	0.91	0.90	185
6	0.97	0.98	0.98	227
7	0.96	0.97	0.96	221
8	0.93	0.87	0.90	182
9	0.98	0.95	0.97	217
accuracy			0.95	2552
macro avg	0.95	0.95	0.95	2552
weighted avg	0.95	0.95	0.95	2552

Accuracy: 0.9533699059561128

C = 0.2 :

```
Linear SVM
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       433
           1       0.99      1.00      0.99       348
           2       0.87      0.94      0.90       238
           3       0.96      0.91      0.94       270
           4       0.94      0.92      0.93       231
           5       0.88      0.90      0.89       185
           6       0.97      0.97      0.97       227
           7       0.96      0.97      0.96       221
           8       0.94      0.87      0.90       182
           9       0.97      0.95      0.96       217

    accuracy                           0.95      2552
   macro avg       0.94      0.94      0.94      2552
weighted avg       0.95      0.95      0.95      2552

Accuracy: 0.9494514106583072
```

C = 0.7 :

```
Linear SVM
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       433
           1       0.99      1.00      0.99       348
           2       0.87      0.94      0.90       238
           3       0.96      0.91      0.94       270
           4       0.93      0.91      0.92       231
           5       0.88      0.90      0.89       185
           6       0.97      0.97      0.97       227
           7       0.95      0.96      0.96       221
           8       0.94      0.87      0.90       182
           9       0.97      0.95      0.96       217

    accuracy                           0.95      2552
   macro avg       0.94      0.94      0.94      2552
weighted avg       0.95      0.95      0.95      2552

Accuracy: 0.9486677115987461
```

*** به نظر میرسد با بیشتر شدن پارامتر C دقت Linear SVM کم تر میشود .

4) تاثیر پارامتر C بر Poly SVM :

C = 0.03 :

```
Poly SVM
              precision    recall  f1-score   support

           0       0.98      0.96      0.97       433
           1       0.95      1.00      0.97       348
           2       0.92      0.85      0.88       238
           3       0.96      0.86      0.91       270
           4       0.58      0.93      0.72       231
           5       0.93      0.85      0.89       185
           6       0.97      0.91      0.94       227
           7       0.95      0.86      0.90       221
           8       0.93      0.85      0.89       182
           9       0.96      0.82      0.88       217

    accuracy                           0.90      2552
   macro avg       0.91      0.89      0.89      2552
weighted avg       0.92      0.90      0.90      2552

Accuracy: 0.8989028213166145
```

C = 0.2 :

```
Poly SVM
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       433
           1       0.97      1.00      0.98       348
           2       0.93      0.94      0.94       238
           3       0.96      0.91      0.94       270
           4       0.79      0.93      0.86       231
           5       0.92      0.90      0.91       185
           6       0.98      0.95      0.96       227
           7       0.95      0.94      0.95       221
           8       0.94      0.88      0.91       182
           9       0.98      0.91      0.94       217

    accuracy                           0.94      2552
   macro avg       0.94      0.93      0.94      2552
weighted avg       0.95      0.94      0.94      2552

Accuracy: 0.942398119122257
```

C = 0.7 :

```
Poly SVM
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       433
           1       0.98      0.99      0.99       348
           2       0.94      0.95      0.94       238
           3       0.96      0.93      0.95       270
           4       0.87      0.94      0.90       231
           5       0.94      0.92      0.93       185
           6       0.98      0.96      0.97       227
           7       0.96      0.96      0.96       221
           8       0.93      0.92      0.93       182
           9       0.99      0.95      0.97       217

    accuracy                           0.96      2552
   macro avg       0.95      0.95      0.95      2552
weighted avg       0.96      0.96      0.96      2552

Accuracy: 0.9565047021943573
```

*** به نظر میرسد با بیشتر شدن C دقت Poly SVM در دسته بندی داده ها بیشتر میشود که کاملا مخالف نتیجه ای است که در مورد Linear SVM گرفتیم .

بخش سوم :

هدف :

این بخش دقیقا مانند بخش دوم است با این تفاوت که در این جا به 3 تا SVM نیاز داریم که اولی کلاس بندی 2و 7 ، دومی کلاس بندی 2و 3 و سومی کلاس بندی س و ص را بر عهده دارد . در این جا هم مانند بخش های قبل SVM با هسته های مختلف را روی dataset ها امتحان میکنیم .

توضیح کد :

کد این بخش بسیار شبیه کد بخش دو است . تنها تفاوت این است که کد بخش دو سه بار به ازای سه dataset مختلفی که در صورت سوال ذکر شده است ؛ تکرار میشود .

بنابراین هیچ نکته ی جدیدی برای توضیح دادن ندارد .

بررسی تاثیر پارامتر های مختلف :

1) بررسی دقت چهار Classifier (هسته های مختلف) و مقایسه ی آن ها با هم :

```
2,7 Detection
Linear SVM
              precision    recall  f1-score   support

           0       1.00      0.98      0.99        86
           1       0.98      1.00      0.99        94

    accuracy                           0.99       180
   macro avg       0.99      0.99      0.99       180
weighted avg       0.99      0.99      0.99       180

Accuracy: 0.9888888888888889
2,7 Detection
RBF SVM
              precision    recall  f1-score   support

           0       0.49      1.00      0.65        86
           1       1.00      0.03      0.06        94

    accuracy                           0.49       180
   macro avg       0.74      0.52      0.36       180
weighted avg       0.75      0.49      0.34       180

Accuracy: 0.4944444444444446
```



```
2,7 Detection
Poly SVM
              precision    recall  f1-score   support

           0       1.00      0.99      0.99        86
           1       0.99      1.00      0.99        94

    accuracy                           0.99       180
   macro avg       0.99      0.99      0.99       180
weighted avg       0.99      0.99      0.99       180

Accuracy: 0.9944444444444445
2,7 Detection
Sigmoid SVM
              precision    recall  f1-score   support

           0       0.48      1.00      0.65        86
           1       0.00      0.00      0.00        94

    accuracy                           0.48       180
   macro avg       0.24      0.50      0.32       180
weighted avg       0.23      0.48      0.31       180

Accuracy: 0.4777777777777778
```

```
2,3 Detection
Linear SVM
              precision    recall  f1-score   support

           0       0.96      0.94      0.95        86
           1       0.95      0.97      0.96        94

    accuracy                           0.96       180
   macro avg       0.96      0.95      0.96       180
weighted avg       0.96      0.96      0.96       180

Accuracy: 0.9555555555555556
```

```
2,3 Detection
RBF SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python3
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.48      1.00      0.65        86
           1       0.00      0.00      0.00        94

    accuracy                           0.48       180
   macro avg       0.24      0.50      0.32       180
weighted avg       0.23      0.48      0.31       180

Accuracy: 0.4777777777777778
```

```
2,3 Detection
Poly SVM
              precision    recall  f1-score   support

           0       0.98      0.94      0.96        86
           1       0.95      0.98      0.96        94

    accuracy                           0.96       180
   macro avg       0.96      0.96      0.96       180
weighted avg       0.96      0.96      0.96       180

Accuracy: 0.9611111111111111
```

```
2,3 Detection
Sigmoid SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python38\
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.48      1.00      0.65        86
           1       0.00      0.00      0.00        94

    accuracy                           0.48       180
   macro avg       0.24      0.50      0.32       180
weighted avg       0.23      0.48      0.31       180

Accuracy: 0.4777777777777778
```

```
W,S Detection
Linear SVM
              precision    recall  f1-score   support

           0       0.97      0.97      0.97        86
           1       0.97      0.97      0.97        94

    accuracy                           0.97       180
   macro avg       0.97      0.97      0.97       180
weighted avg       0.97      0.97      0.97       180

Accuracy: 0.9666666666666667
W,S Detection
RBF SVM
              precision    recall  f1-score   support

           0       1.00      0.12      0.21        86
           1       0.55      1.00      0.71        94

    accuracy                           0.58       180
   macro avg       0.78      0.56      0.46       180
weighted avg       0.77      0.58      0.47       180

Accuracy: 0.5777777777777777
```

```
W,S Detection
Poly SVM
              precision    recall  f1-score   support

           0       0.95      0.97      0.96        86
           1       0.97      0.96      0.96        94

    accuracy                           0.96       180
   macro avg       0.96      0.96      0.96       180
weighted avg       0.96      0.96      0.96       180

Accuracy: 0.9611111111111111
```

```
W,S Detection
Sigmoid SVM
C:\Users\ASUS ZENBOOK\AppData\Local\Programs\Python\Python38\
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.48      1.00      0.65        86
           1       0.00      0.00      0.00        94

    accuracy                           0.48       180
   macro avg       0.24      0.50      0.32       180
weighted avg       0.23      0.48      0.31       180

Accuracy: 0.4777777777777778
```

*** در مورد هر سه dataset میبینیم که در این جا هم مانند بخش دوم SVM با هسته ی Linear و Poly بسیار بهتر از SVM با هسته ی RBF و Sigmoid عمل کرده است و دقت آن ها به طور قابل ملاحظه ای بیشتر است .

- چون دقت Linear SVM و Poly SVM خیلی بهتر شده است بقیه ی آزمایش ها را تنها بر روی این دو Classifier انجام میدهیم .

2) درصد داده های آموزشی و آزمایشی :

Test = 0.1 / train = 0.9 :

```
2,7 Detection
Linear SVM
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       1.00      1.00      1.00        33

    accuracy                           1.00        60
   macro avg       1.00      1.00      1.00        60
weighted avg       1.00      1.00      1.00        60

Accuracy: 1.0
```

```
2,7 Detection
Poly SVM
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       1.00      1.00      1.00        33

    accuracy                           1.00        60
   macro avg       1.00      1.00      1.00        60
weighted avg       1.00      1.00      1.00        60

Accuracy: 1.0
```

```
W,S Detection
Linear SVM
              precision    recall  f1-score   support

           0       0.93      0.96      0.95        27
           1       0.97      0.94      0.95        33

    accuracy                           0.95        60
   macro avg       0.95      0.95      0.95        60
weighted avg       0.95      0.95      0.95        60

Accuracy: 0.95
```

```
W,S Detection
Poly SVM
              precision    recall  f1-score   support

           0       0.93      0.96      0.95        27
           1       0.97      0.94      0.95        33

    accuracy                           0.95        60
   macro avg       0.95      0.95      0.95        60
weighted avg       0.95      0.95      0.95        60

Accuracy: 0.95
```

Test = 0.35 / train = 0.65 :

```
2,7 Detection
Linear SVM
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       102
           1       0.99      1.00      1.00       108

    accuracy                           1.00       210
   macro avg       1.00      1.00      1.00       210
weighted avg       1.00      1.00      1.00       210

Accuracy: 0.9952380952380953
```

```
2,7 Detection
Poly SVM
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       102
           1       0.99      1.00      1.00       108

    accuracy                           1.00       210
   macro avg       1.00      1.00      1.00       210
weighted avg       1.00      1.00      1.00       210

Accuracy: 0.9952380952380953
```

```
W,S Detection
Linear SVM
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       102
           1       0.95      0.97      0.96       108

    accuracy                           0.96       210
   macro avg       0.96      0.96      0.96       210
weighted avg       0.96      0.96      0.96       210

Accuracy: 0.9619047619047619
```

```
W,S Detection
Poly SVM
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       102
           1       0.96      0.96      0.96       108

    accuracy                           0.96       210
   macro avg       0.96      0.96      0.96       210
weighted avg       0.96      0.96      0.96       210

Accuracy: 0.9619047619047619
```

*** همانطور که از آزمایش ها مشخص است با کم تر شدن تعداد داده های آموزشی دقت Classifier ها هم کم تر میشود .

3) تاثیر پارامتر C بر Linear SVM :

C = 0.03

```
2,3 Detection
Linear SVM
              precision    recall  f1-score   support

           0       0.98      0.94      0.96       102
           1       0.95      0.98      0.96       108

    accuracy                           0.96       210
   macro avg       0.96      0.96      0.96       210
weighted avg       0.96      0.96      0.96       210

Accuracy: 0.9619047619047619
```

C = 0.2 :

```
2,3 Detection
Linear SVM
              precision    recall  f1-score   support

           0       0.97      0.94      0.96       102
           1       0.95      0.97      0.96       108

    accuracy                           0.96       210
   macro avg       0.96      0.96      0.96       210
weighted avg       0.96      0.96      0.96       210

Accuracy: 0.9571428571428572
```

C = 1 :

```
2,3 Detection
Linear SVM
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       102
           1       0.95      0.97      0.96       108

    accuracy                           0.96       210
   macro avg       0.96      0.96      0.96       210
weighted avg       0.96      0.96      0.96       210

Accuracy: 0.9619047619047619
```

*** به نظر میرسد رابطه ی خطی بین C و بهتر شدن دقت Linear SVM وجود ندارد .

4) تاثیر پارامتر C بر Poly SVM :

C = 0.03 :

```
2,3 Detection
Poly SVM
              precision    recall  f1-score   support

           0       0.92      0.95      0.93       102
           1       0.95      0.92      0.93       108

    accuracy                           0.93       210
   macro avg       0.93      0.93      0.93       210
weighted avg       0.93      0.93      0.93       210

Accuracy: 0.9333333333333333
```

C = 0.2 :

```
2,3 Detection
Poly SVM
              precision    recall  f1-score   support

           0       0.98      0.94      0.96       102
           1       0.95      0.98      0.96       108

    accuracy                           0.96       210
   macro avg       0.96      0.96      0.96       210
weighted avg       0.96      0.96      0.96       210

Accuracy: 0.9619047619047619
```

C = 1 :

```
2,3 Detection
Poly SVM
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       102
           1       0.95      0.97      0.96       108

    accuracy                           0.96       210
   macro avg       0.96      0.96      0.96       210
weighted avg       0.96      0.96      0.96       210

Accuracy: 0.9619047619047619
```

*** به نظر میرسد زیاد کردن C تا یک جایی باعث دقیق تر شدن Poly SVM در کلاس بندی میشود ؛ اما از یک جایی به بعد چندان تاثیری در دقیق تر شدن آن ندارد .

بررسی و مقایسه ی کلاس بندی داده های USPS به کمک SVM و شبکه ی عصبی :

بعد از انجام آزمایش های بسیار زیاد ☺ به نظر میرسد اگر از شبکه ی عصبی برای کلاس بندی داده ها استفاده کنیم بعد از طی شدن تعداد خاصی چرخه نتیجه دیگر عملا بیشتر بهبود پیدا نمیکند و به همین دلیل میتوان این تعداد چرخه را (تا جایی که نتیجه در حال بهتر شدن است .) به عنوان Max_iteration به Classifier داد و به این ترتیب زمان اجرای برنامه را تا حد خیلی خوبی کم کرد. اما در SVM در این مورد نمتوان تصمیم گیری کرد به همین دلیل برای من کلاس بندی کردن داده ها به کمک شبکه ی عصبی خیلی کم تر از SVM زمان لازم داشت .

اما به جهت دقت کلاس بندی به نظر میرسد در صورت انتخاب هسته ی مناسب SVM بهتر عمل میکند و دقت بالا تری در داده های آموزشی به دست می دهد .