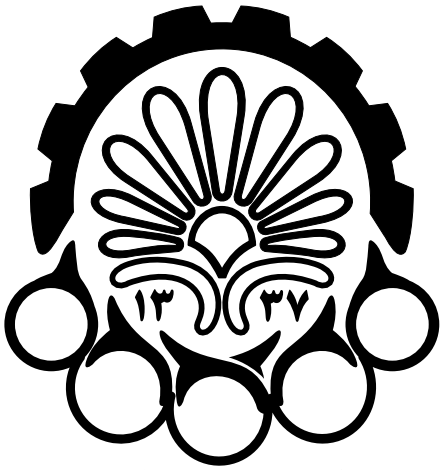


دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

تمر ین سوم درس یادگیری ماشین

دکتر ناظرفرد

غلامرضا دار ۴۰۰۱۳۱۰۱۸

زمستان ۱۴۰۰

فهرست مطالب

بخش اول: پرسشهای تشریحی

سوال ۱)

الف) ماشین های بردار پشتیبان با کرنل خطی پارامتریک هستند زیرا غیروابسته به داده آموزش، فرم کلی decision function آنها یک خط است. اما به عنوان مثال در مورد ماشین های بردار پشتیبان با کرنل RBF این قضیه درست نیست و با تغییر داده آموزش ممکن است فرم decision function تغییر کند.

ب) SVM های Hard Margin به شدت Overfit می‌شوند. پارامتر C در Soft Margin SVM و گاما در RBF برای مقابله با Overfitting استفاده می‌شوند. هرچه C کمتر باشد مدل دچار Regularization بیشتری می‌شود و از Overfitting کاسته می‌شود. پارامتر گاما نیز در Overfitting تاثیر دارد و پیدا کردن مقدار مناسب گاما بسیار مهم است. برای توضیحات بیشتر به قسمت د سوال اول پیاده سازی مراجعه کنید.

ج) در Hard Margin SVM ها Decision boundary فقط به بردار های پشتیبان بستگی دارد و داده های پرت و نویز تاثیر کمتری دارند. اما در Soft Margin SVM ها چه پارامتر C کمتر باشد داده های بیشتری

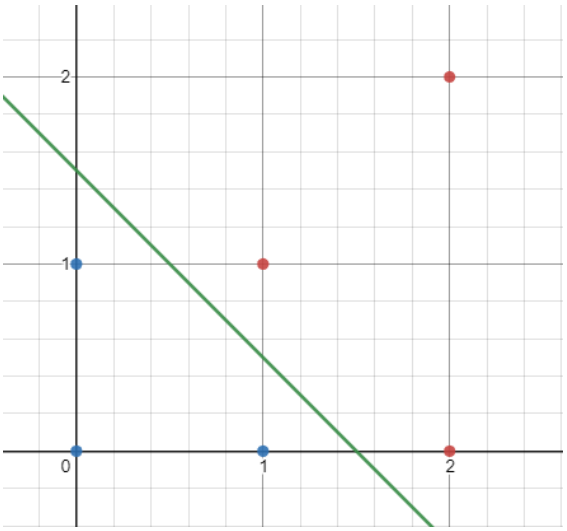
د) خیر. فقط در صورتی این اتفاق می‌افتد که داده ها به کمک یک ترکیب خطی از دسته بند های ضعیف قابل جداپذیری باشند.

هـ) بله. وزن های اختصاص داده شده به داده ها مانند احتمال هستند. مقادیر آنها بین ۰ تا ۱ است و مجموع همه وزن ها ۱ می‌شود.

و) بله. الگوریتم Ada Boost به عنوان loss function از تابع نمایی استفاده میکند که باعث می‌شود داده های misclassify شده را بیشتر جریمه کند.

سوال ۲)

الف) بله جداپذیر خطی هستند.



ب) محاسبه شیب خط و عرض از مبدا

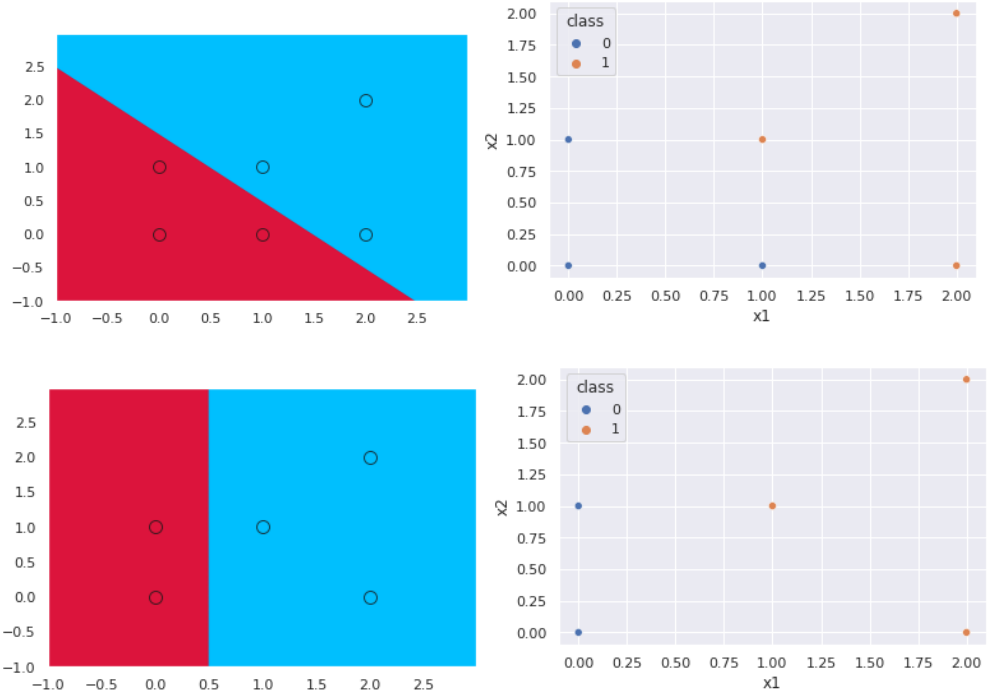
$$m = \frac{\Delta y}{\Delta x} = \frac{0 - 1.5}{1.5 - 0} = -1$$

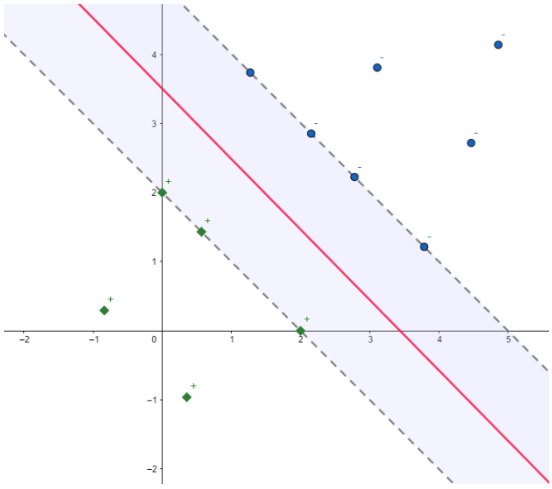
$$b = 1.5$$

معادله خط جداکننده به شکل زیر است $(y = mx + b)$

$$y = -x + 1.5$$

ج) بله تغییر میکند. در مثالی که مشاهده میکنید نقطه $[1,0]$ حذف شد و W از $[2,2]$ به $[2,0]$ تغییر یافت.



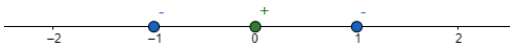


د) به شکل زیر توجه کنید. در SVM(Hard Margin) تضمین می‌شود در ناحیه هاشور زده(جاده) هیچ داده ای وجود نداشته باشد. چون در آن صورت آن نقطه به عنوان بردار پشتیبان انتخاب می‌شد و مارجین و در نتیجه ناحیه هاشور خورده تغییر میکرد. در این مثال اگر تمام بردار های پشتیبان هم حذف شوند تنها نقاطی که از خط جداکننده دور تر هستند باقی خواهند ماند و نقش بردار پشتیبان را به خود میگیرند. در نتیجه این اتفاق **مارجین افزایش** می‌یابد. اگر فقط تعدادی از این بردار ها را حذف کنیم، سایر بردار های پشتیبان مارجین را **ثابت** نگه میدارند.

نکته: تصاویر این سوال بدون استفاده از کد و به صورت دستی اما با نرم افزار رسم شده اند به جز قسمت (ج) که با کمک sklearn برای درک بهتر رسم شده.

سوال ۳)

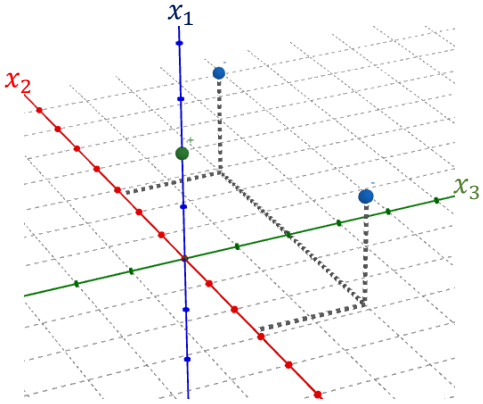
x	Label
0	+
-1	-
1	-



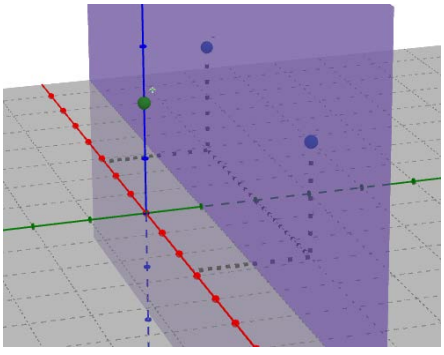
همانطور که مشاهده میکنید داده ها جداپذیر خطی نیستند. با استفاده از کرنل داده شده داده ها را به بعد بالاتر میبریم.

x_1	x_2	x_3	Label
1	0	0	+
1	$-\sqrt{2}$	1	-
1	$\sqrt{2}$	1	-

پس از انتقال به بعد بالاتر داریم(هر واحد grid برابر ۰/۵):



به راحتی میتوان دید که در این حالت داده ها به کمک یک صفحه جدا می‌شوند. **معادله صفحه :** $\boldsymbol{x_3 = \frac{1}{2}}$



حل سوال ۳ با کمک روابط ریاضی :

ابتدا مقادیر متناظر kernel را بدست می‌آوریم.

1	1	1
1	4	0
1	0	4

معادله هایی که داریم را مینویسیم :

$$C1 = \alpha_1 - \alpha_2 - \alpha_3 = 0$$

$$C2.A = \alpha_1(1) - \alpha_2(1) - \alpha_3(1) + b = +1$$

$$C3.B = \alpha_1(1) - \alpha_2(4) - \alpha_3(0) + b = -1$$

$$C3.C = \alpha_1(1) - \alpha_2(0) - \alpha_3(4) + b = -1$$

$$\alpha_1 - 2\alpha_2 = 0 \rightarrow \boldsymbol{\alpha_1 = 2\alpha_2}$$

$$2\alpha_2 - 2\alpha_2 + b = 1 \rightarrow \boldsymbol{b = 1}$$

$$\boldsymbol{\alpha_2 = \alpha_3 = 1}$$

$$\boldsymbol{\alpha_1 = 2}$$

$$h(x) = \sum \alpha_i y_i \, K(x_i, x) + b \geq 0$$

$$w = \sum \alpha_i y_i x_i$$

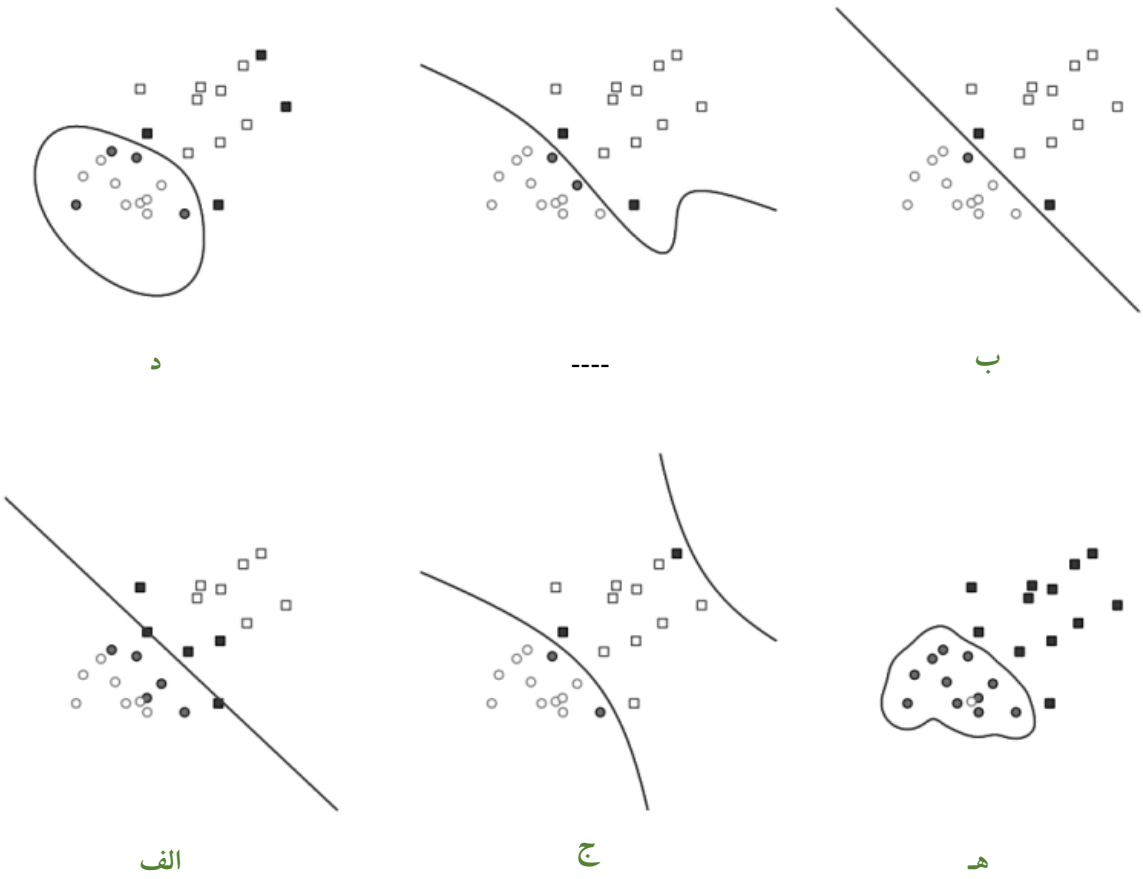
$$w = [0, 0, -2]^T$$

$$h(x) = 2(x_1) - 1\left(x_1 - \sqrt{2}\, x_2 + x_3\right) - 1\left(x_1 + \sqrt{2}\, x_2 + x_3\right) + 1 = 0$$

معادله صفحه جداکننده :

$$\boldsymbol{x_3 = \frac{1}{2}}$$

سوال ٤)



شکل شماره ۳و۴ به طور واضح **خطی** هستند. از بین این دو، شکل ۳ دارای **C بزرگتری** است زیرا به Hard Margin SVM نزدیک تر شده پس ۳ همان **(ب)** است و ۴ همان **(الف)**.

شکل شماره ۲ یک چند جمله ای با درجه بالاتر از ۲ است (۳ یا ۴ یا بیشتر) پس نمیتواند گزینه مناسبی برای **(ج)** باشد.

به این دلیل است که کرنل ۵ را برای شکل **(ج)** برگزیدیم.

شکل شماره ۱و۶ به احتمال زیاد decision boundary کرنل RBF هستند. هر چه مقدار γ بیشتر باشد، بردار های پشتیبان روی ناحیه کوچکتری اثر میگذارند و مدل Complex تر میشود و همچنین Decision boundary تیزتر(!) می‌شود. بنابراین ۶ همان **(هـ)** است و ۱ همان **(د)** است.

سوال ۵)

$$\epsilon^1 = \sum w_i^1 = \frac{1}{8}$$

$$\alpha^1 = \frac{1}{2}\ln\left(\frac{1-\epsilon^1}{\epsilon^1}\right) = \frac{1}{2}\ln\left(\frac{\frac{7}{8}}{\frac{1}{8}}\right) = \mathbf{0.5\ ln7}$$

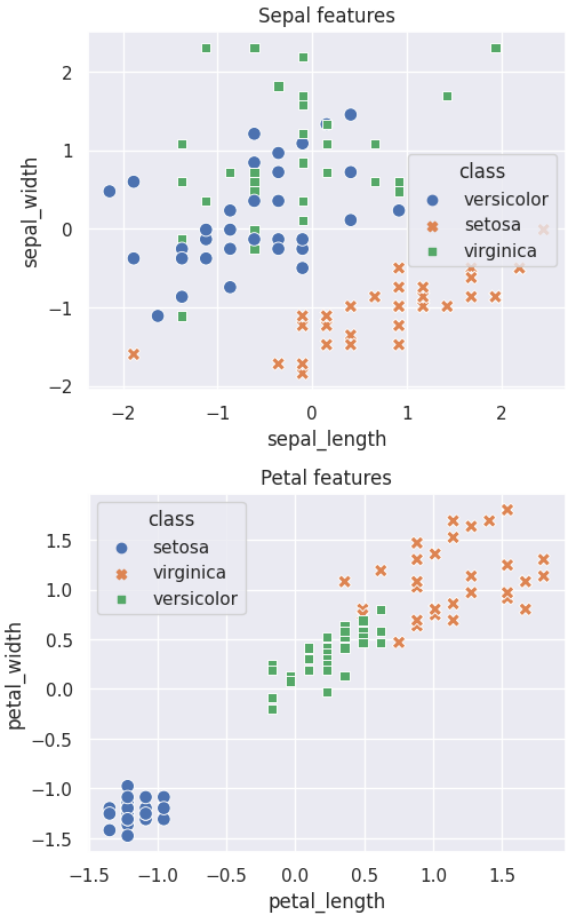
بخش دوم: پیاده سازی

سوال ۱)

نوت بوک مربوط به این تمرین
https://colab.research.google.com/drive/1w78l4MI6mCXaGTqFnSwrd_hlBEdCkzfe?usp=sharing

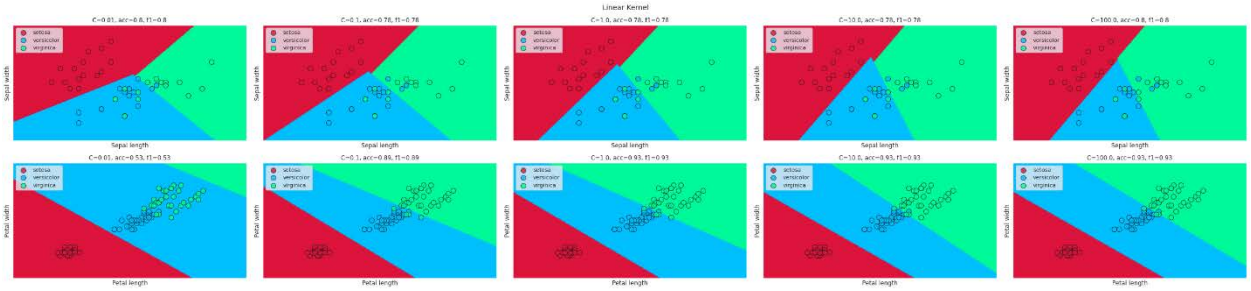
پس از لود کردن دیتاست Iris، آنرا به بخش های Sepal features, Petal features تقسیم میکنیم (زیرا میخواهیم ویژگی ها را دو به دو بررسی کنیم تا قابل رسم باشد). سپس این بخش ها را به نسبت **۳۰ـ۷۰** به داده آموزش و ارزیابی تقسیم میکنیم.

الف) ابتدا به رسم داده ها میپردازیم. در این بخش دو نمودار خواهیم داشت. نمودار اول شامل ویژگی های Sepal Length, Sepal Width است و نمودار دوم شامل ویژگی های Petal Length, Petal Width.

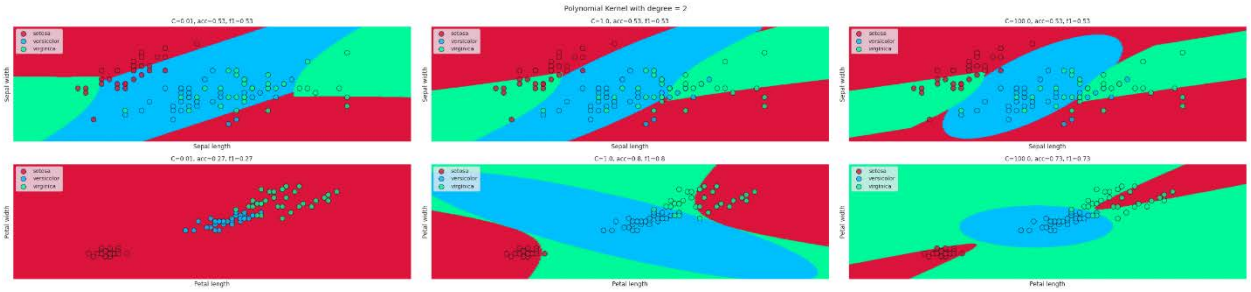


ب) در این بخش میخواهیم به ازای کرنل های مختلف، داده های تست این دیتاست را طبقه بندی کنیم و همچنین Decision boundary را رسم کنیم. این کار را به ازای پارامتر های مختلفی برای هر کرنل انجام میدهیم. در مجموع ۶۴ تصویر خواهیم داشت. مقدار هر پارامتر به همراه Accuracy و F1-score متناظر در بالای هر نمودار قرار داده شده است. **توصیه می‌شود برای دیدن بهتر نمودار ها فایل های تصویری که در پوشه این تمرین وجود دارد را ببینید.**

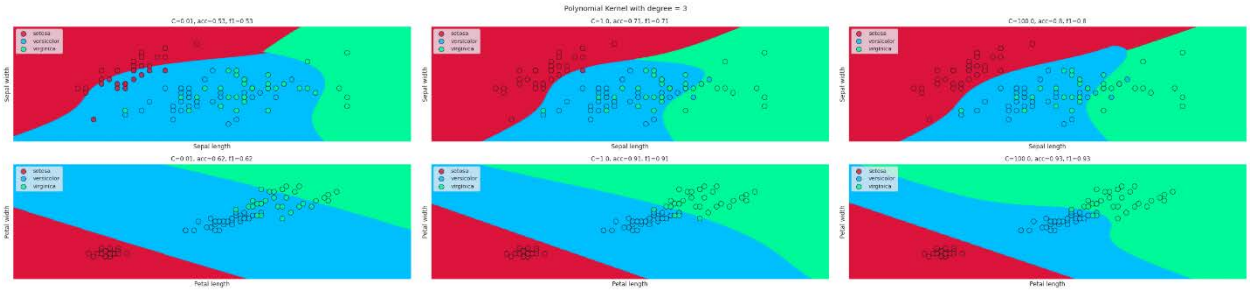
Linear Kernel ; $C \in \{0.01, 0.1, 1, 10, 100\}$



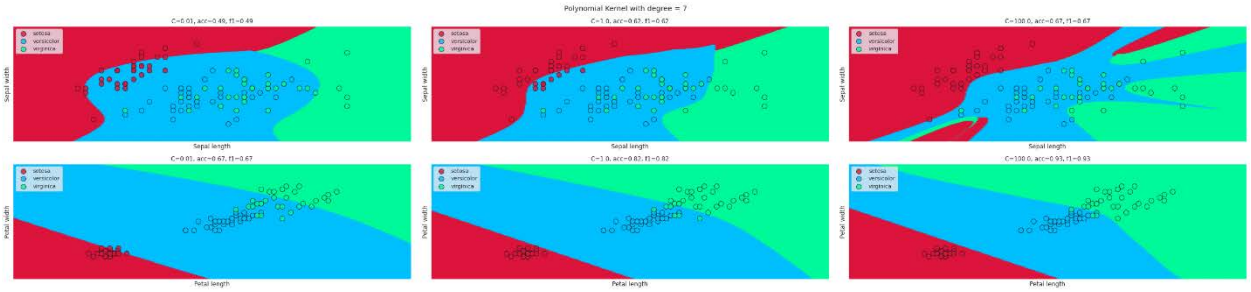
Polynomial Kernel ; $C \in \{0.01, 1, 100\}$ *and* $d = 2$



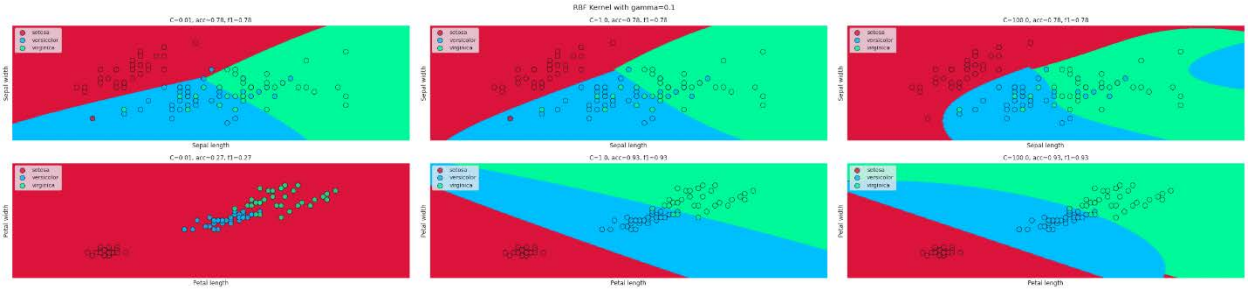
Polynomial Kernel ; $C \in \{0.01, 1, 100\}$ *and* $d = 3$



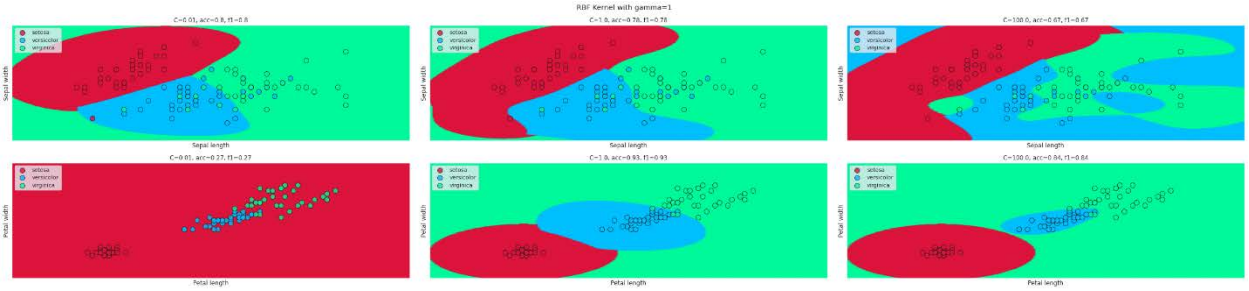
Polynomial Kernel ; $C \in \{0.01, 1, 100\}$ *and* $d = 7$



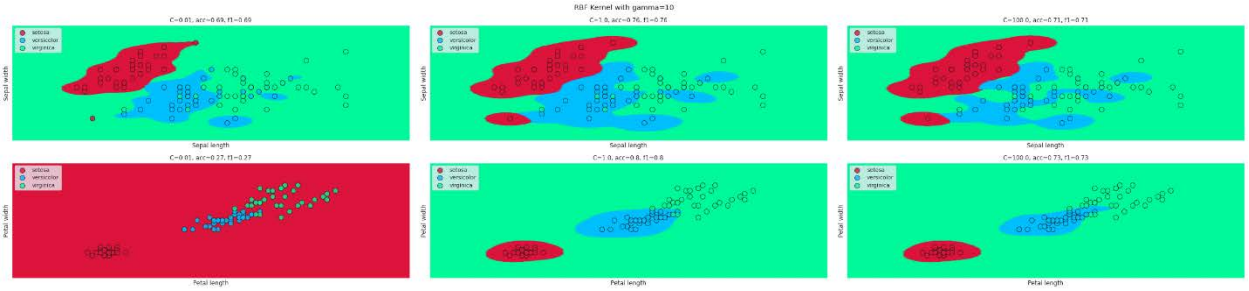
RBF Kernel ; $C \in \{0.01, 1, 100\}$ *and* $\gamma = 0.1$



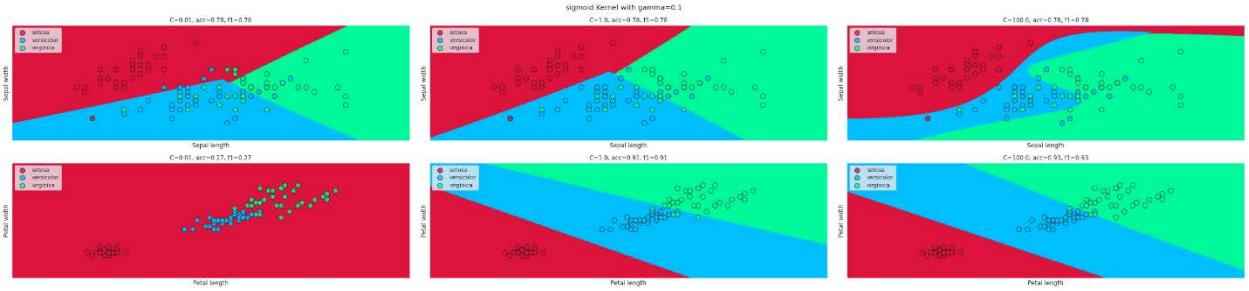
RBF Kernel ; $C \in \{0.01, 1, 100\}$ *and* $\gamma = 1$



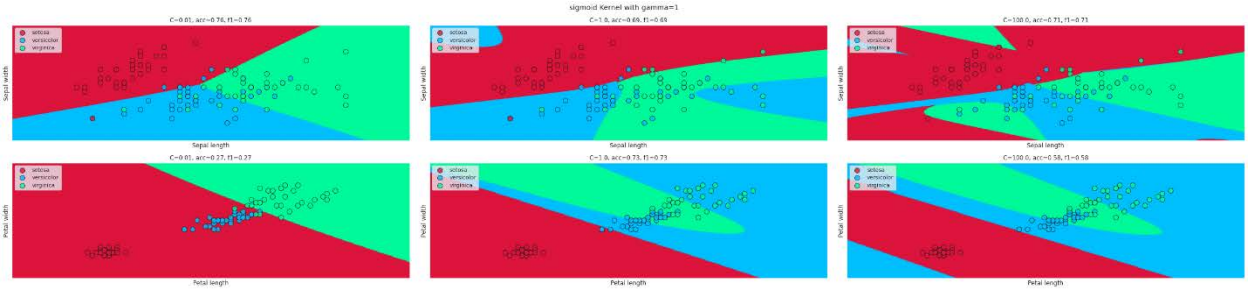
RBF Kernel ; $C \in \{0.01, 1, 100\}$ *and* $\gamma = 10$



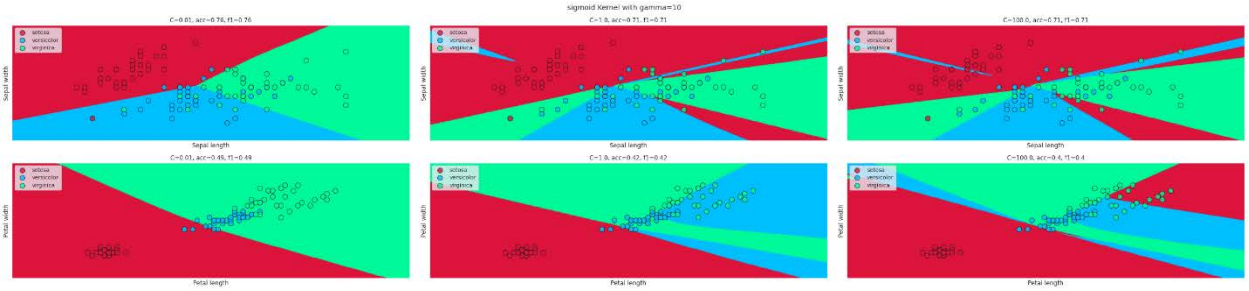
Sigmoid Kernel ; $C \in \{0.01, 1, 100\}$ *and* $\gamma = 0.1$



Sigmoid Kernel ; $C \in \{0.01, 1, 100\}$ *and* $\gamma = 1$



Sigmoid Kernel ; $C \in \{0.01, 1, 100\}$ *and* $\gamma = 10$



ج) معیار Accuracy, F1-measure در تصاویر بالا موجود هستند اما جهت بهتر دیده شدن اعداد به پوشه این تمرین، یا notebook مراجعه کنید.

https://colab.research.google.com/drive/1w78l4MI6mCXaGTqFnSwrd_hlBEdCkzfe?usp=sharing

د) در مورد پارامتر های SVM میتوانیم نکات زیر را بیان کنیم.

C : این پارامتر یک Tradeoff بین Hard Margin بودن SVM و پهنای جاده آن است. هر چه مقدار این پارامتر بیشتر باشد به Hard Margin SVM نزدیک تر می‌شویم و تعداد Misclassification ها کاهش می‌یابد. برای مقابله با Overfitting بهتر است مقدار C را کم کنیم.

Degree : درجه چند جمله ای در کرنل چند جمله ای مشخص میکند منحنی جدا کننده کلاس ها چه مقدار پیچیدگی داشته باشد. اگر درجه ۲ باشد داده ها با یک خط یا یک سهمی از هم جدا میشوند، اگر درجه بالاتر باشد ممکن است منحنی هایی با پیچیدگی بیشتر decision boundary را تعیین کنند. در نتیجه هر چه degree بیشتر باشد احتمال Overfitting بالاتر می‌رود.

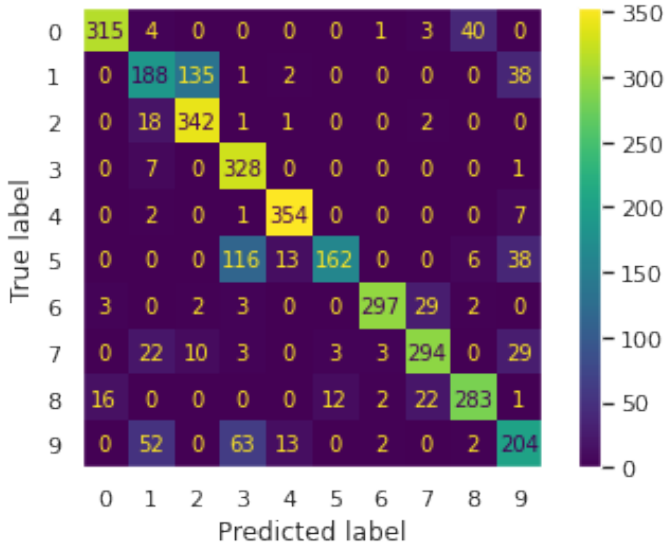
γ : این پارامتر که در کرنل های RBF و Sigmoid مورد استفاده قرار میگیرد، تعیین میکند بردارهای پشتیبان علاوه بر خودشان به چه تعداد داده دیگر تاثیر میگذارند. اگر این پارامتر بسیار زیاد باشد، بردارهای پشتیبان فقط به خودشان اثر میگذارند و Overfitting رخ میدهد. اگر مقدار این پارامتر خیلی کم باشد مدل به Linear SVM تبدیل میشود و در واقع هر بردار پشتیبان روی تمام نقاط دیتاست تاثیر میگذارند. مقادیر میانی این پارامتر بهترین نتیجه را میدهند.

سوال ۲)

الف) Random Forest

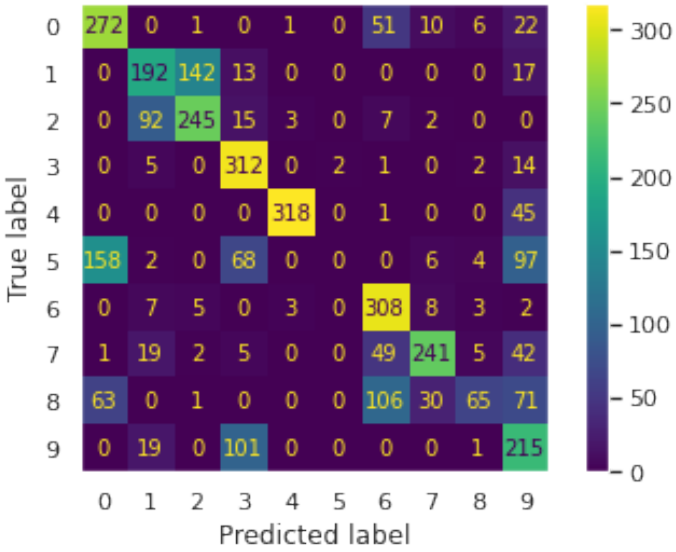
نوت بوک مربوط به این تمرین https://colab.research.google.com/drive/11EkT4FEim1jgfc2BrmBACjDkohlY1an5?usp=sharing

```
Accuracy of the random forest model is 0.7910234419668382
f1-score of the random forest model is 0.7858804633993521
```



ب) Ada Boost

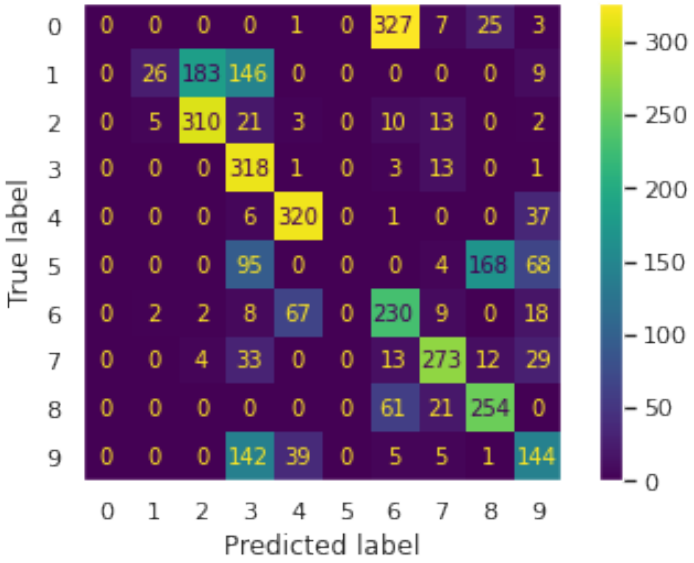
```
Accuracy of the model with 10 classifiers is 0.6197827329902802
f1-score of the model with 10 classifiers is 0.5789423132365334
```



ج) Ada Boost با مقادیر n_estimator متفاوت

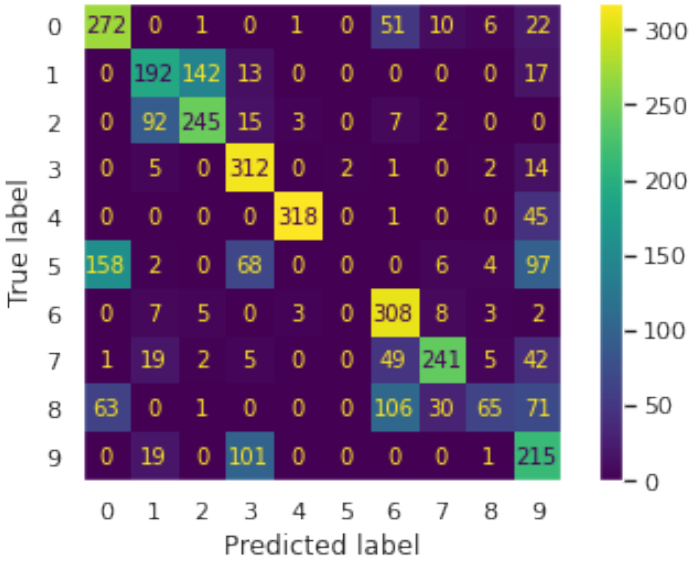
n_estimator = 5

```
Accuracy of the model with 5 classifiers is 0.5360205831903945
f1-score of the model with 5 classifiers is 0.45645290211687045
```



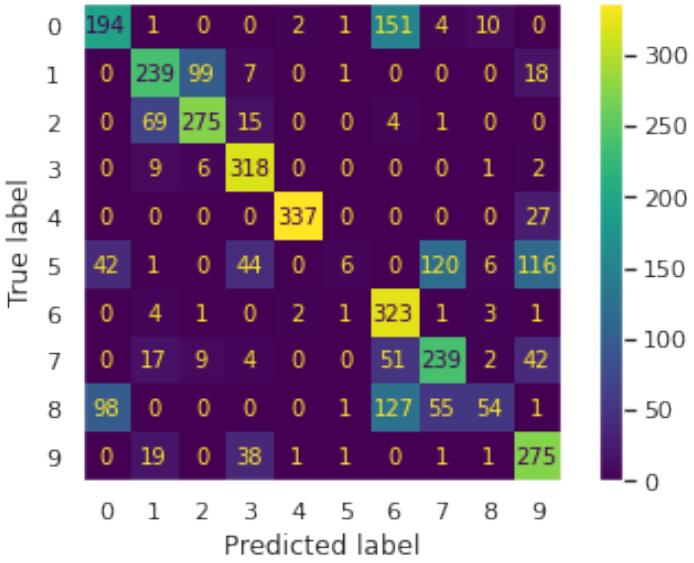
n_estimator = 10

```
Accuracy of the model with 10 classifiers is 0.6197827329902802
f1-score of the model with 10 classifiers is 0.5789423132365334
```



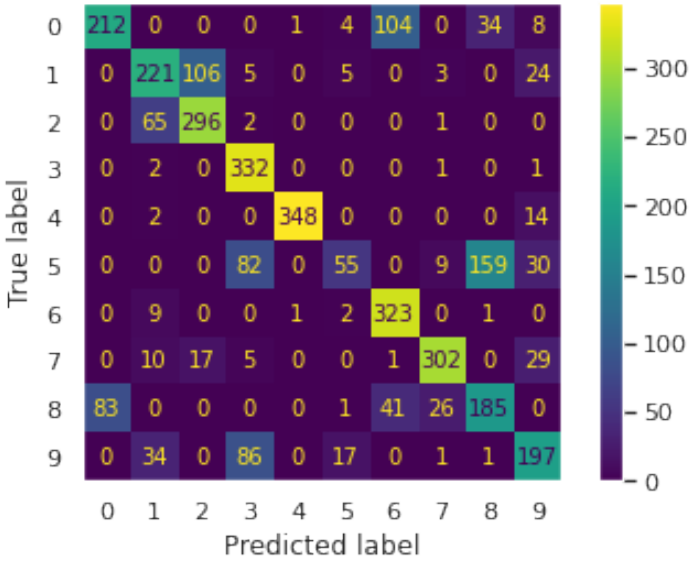
n_estimator = 20

```
Accuracy of the model with 20 classifiers is 0.6460834762721556
f1-score of the model with 20 classifiers is 0.6009251743573313
```



n_estimator = 50

```
Accuracy of the model with 50 classifiers is 0.706403659233848
f1-score of the model with 50 classifiers is 0.6872375017892648
```



د) XGBoost

ابتدا با استفاده از مقادیر پیشفرض XGBoost دسته بند را آموزش میدهیم.

```
accuracy default 0.9614065180102916
f1-score default 0.9667386169827855
```

Test Accuracy : 0.961

F1-Score : 0.966

سپس با در نظر گرفتن مقادیر خاصی برای پارامتر های XGBoost به نتایج زیر دست یافتیم.

$$n\_estimators = 420,$$

$$max\_depth = 11,$$

$$eta = 0.3,$$

$$colsample\_bytree = 0.7,$$

$$subsample = 0.5$$

```
accuracy n_estimators=420, max_depth=11, eta=0.3, colsample_bytree=0.7, subsample=0.5 0.9671240708976558
f1-score n_estimators=420, max_depth=11, eta=0.3, colsample_bytree=0.7, subsample=0.5 0.9671299341223256
```

Test Accuracy : 0.967

F1-Score : 0.967