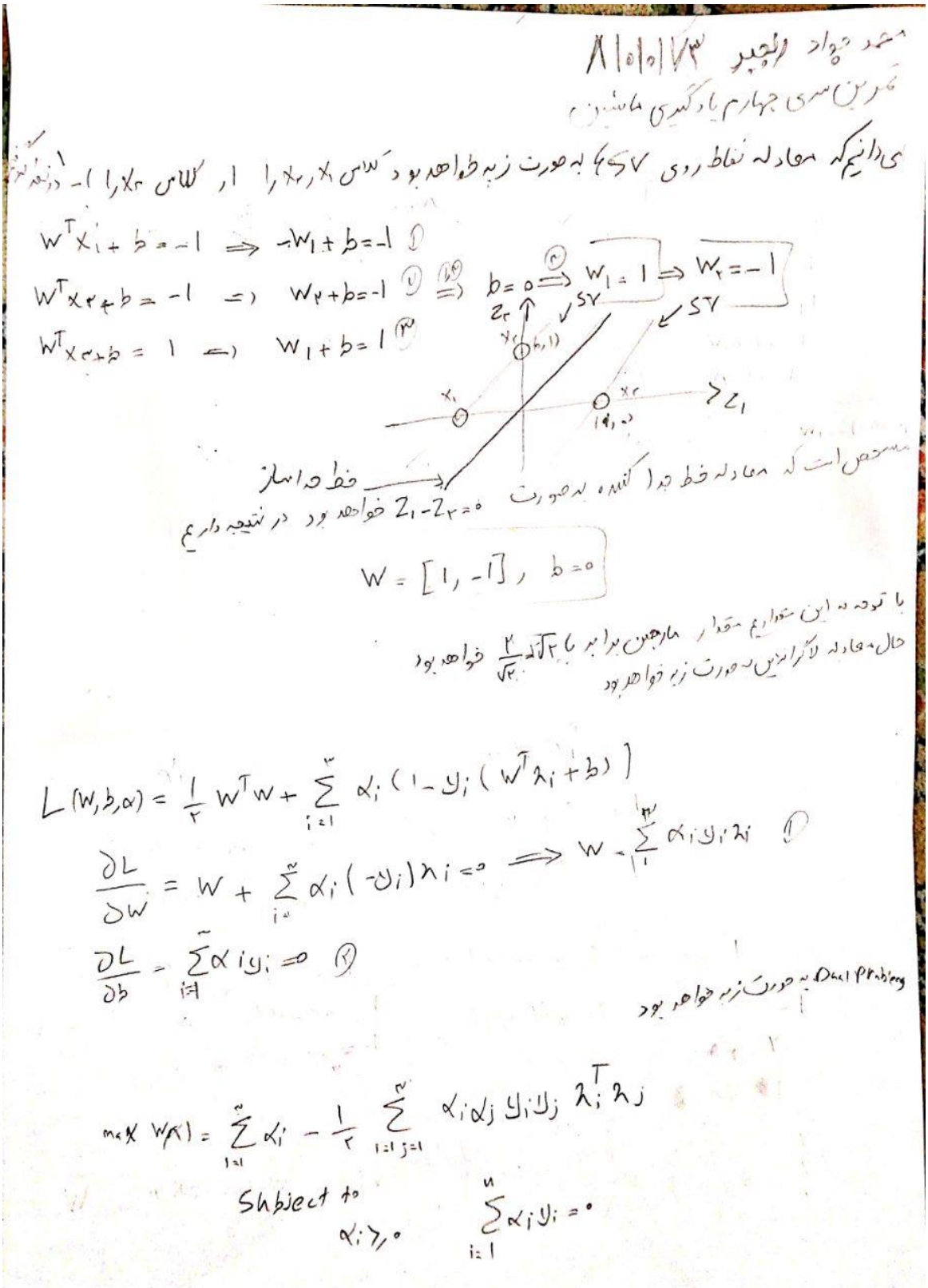
Mohammad Javad Ranjbar

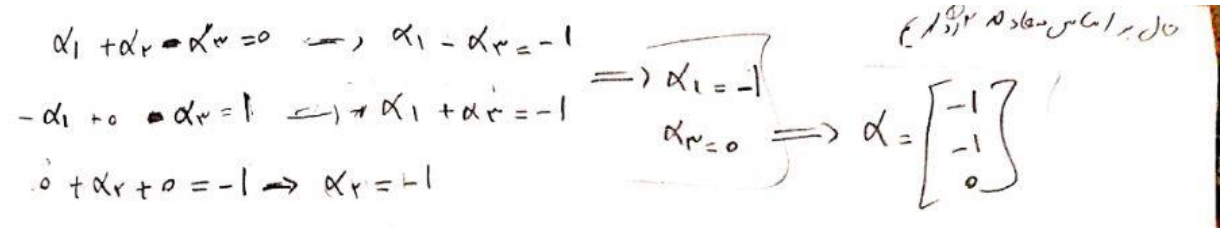
810101173

HW4

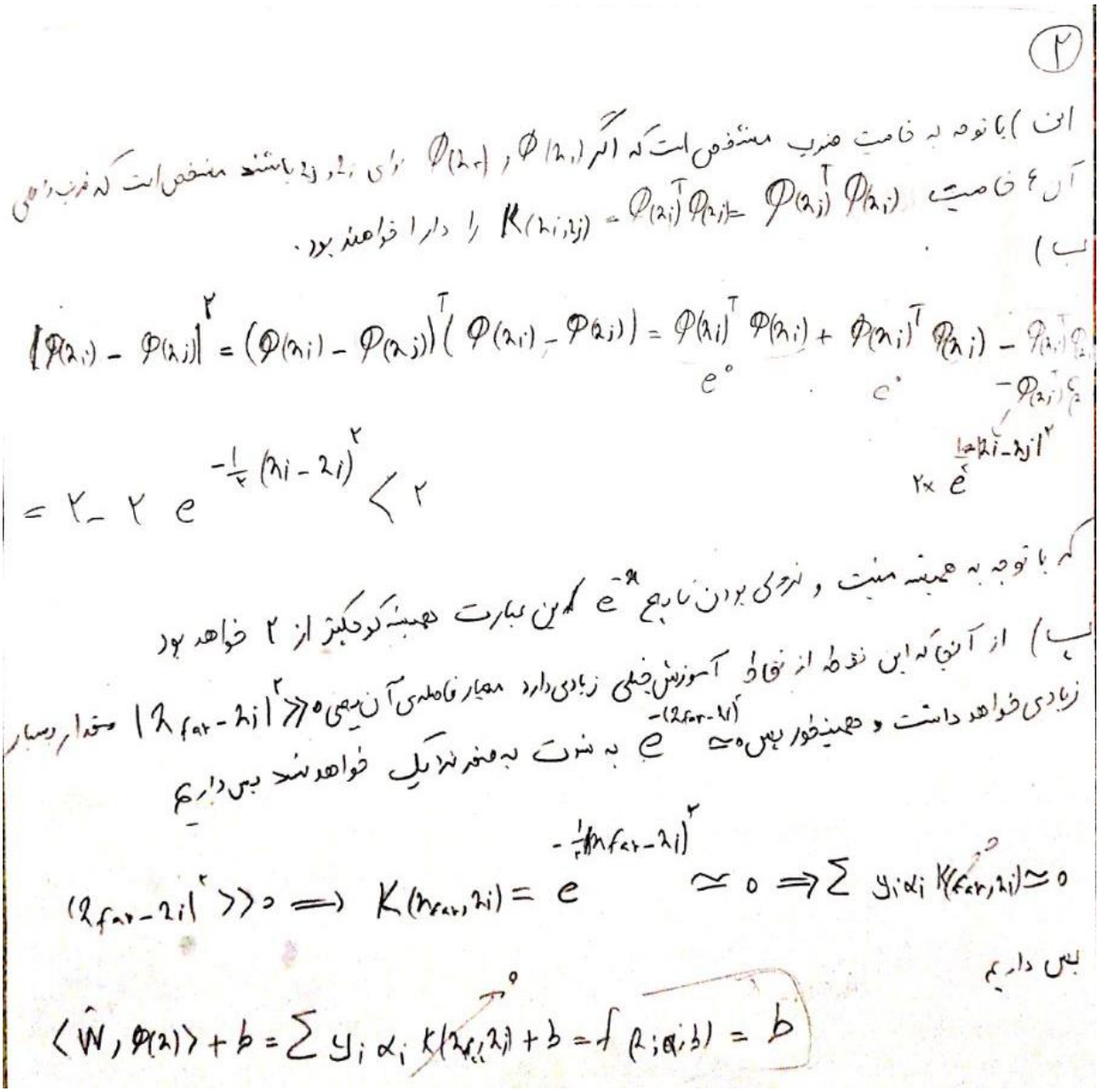
Machine Learning, Fall 2022

سوال ۱:





سوال ۲:



سوال ۳:

(۳)

Soft margine خطی با $C=1$: برای شکل ۴ می باشد

این SVM باید خطی باشد پس یا شکل ۲ می تواند باشد یا شکل ۴. حال با توجه به اینکه مقدار C نسبتا کوچکتری است یعنی جا برای خطای بیشتری داریم پس باید شکلی را انتخاب کنیم که دو کلاس را آسانگیرانه تر جدا می کند که شکل ۴ است

Soft margine خطی با $C=10$: برای شکل ۲ می باشد

این SVM خطی است پس یا شکل ۲ می تواند باشد یا شکل ۴. حال با توجه به مقدار بزرگ C باید اجازه ی خطای کمی داشته باشیم در نتیجه شکلی را انتخاب می کنیم که دو کلاس را سختگیرانه تر جدا می کند و پیش شکل ۲ انتخاب می شود

Hard margine با کرنل $K(x_i,x_j)=x_i x_j+(x_i.x_j)^2$: برای شکل های ۱ و ۳ باشد

از آنجاکه تابع کرنل ها توان دوبی می باشد باید به شکل سهمی دور باشد که تنها شکل ۱ این را فراهم می کند شکل ۱ است

Hard margin با کرنل $K(x_i,x_j)=\exp(-\ln(x_i-x_j)^2)$: برای شکل ۶ است

با توجه به خمیر خطی بودن این جدایی ساز و زیاد بودن مقدار ln جدای نقاطی که فاصله کمی از هم دارند مقدار کرنل بیشتر از حالت با $\lambda=\frac{1}{10}$ خواهد بود پس برای طبقه بندی نیاز به تعداد زیادی SV نداریم پس من شکل ۶ و ۱ باید شکل ۱ را انتخاب کنم

Hard margin با کرنل $K(x_i,x_j)=\exp(-\frac{1}{1}(x_i-x_j)^2)$: برای شکل ۱ است

این جدایی ساز غیر خطی بوده و باید یا برای شکل ۱ باشد یا ۶ حال با توجه به کم بودن مقدار $\lambda=\frac{1}{10}$ بعضی برای نزدیک به هم مقدار کرنل به شدت کم خواهد شد در نتیجه به SV های زیادی برای طبقه بندی نیاز داریم که بین شکل ۶ و ۱، باید شکل ۱ را انتخاب کنم.

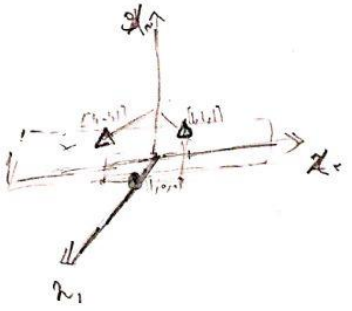
سوال ۴:

(۴)

الف) نقاط ما با اعمال کرنل به شکل زیر درخواهند آمد:

$$\phi(x_1) = [1, 0, 0], \quad \phi(x_2) = [1, -1, 1], \quad \phi(x_3) = [1, 1, 1]$$

حال اگر دستگاه دو بعدی رسم کنیم



مشخص است که در شکل بالا نقاط را می‌توان با یک صفحه از یکدیگر جدا کرد پس خطی جدایی پذیر هستند

ب) معادله لاگرانژ را می‌نویسیم

$$L(w,\alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{3} \alpha_i \left(y_i \left(w^T \phi(x_i) + b\right) - 1\right)$$

حال برای بدست آوردن شرط بهینه به آن‌ها مشتق گرفته و برابر با صفر می‌گذاریم:

$$\frac{\partial L(w,\alpha)}{\partial w} = w + \sum_{i=1}^{3} \alpha_i y_i \phi(x_i) = 0 \rightarrow$$

$$\frac{\partial L(w,\alpha)}{\partial b} = \sum_{i=1}^{3} \alpha_i y_i = 0$$

حال جاگذاری می‌کنیم:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + \alpha_1 x_1 \times 1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 x_2 \times 1 \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + \alpha_3 x_3 \times 1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 0$$

$$w_1 = \alpha_1 + \alpha_2 + \alpha_3 = 0 \quad (1)$$

$$w_2 - 0 = \alpha_2 + \alpha_3 = 0 \quad (2)$$

$$w_3 - 0 = \alpha_2 + \alpha_3 = 0 \quad (3)$$

$$-\alpha_1 + \alpha_2 + \alpha_3 = 0 \quad (4)$$

بر اساس ۴ داریم $w_1 = 0$ که اثر در معادله قید نگذاری کنیم معادلات زیر را بدست می آوریم

$$y_1 \times (w_0^T \phi(x_1) + b) = 1 \Rightarrow$$

$$-1 \times ([w_1, w_2, w_3] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b) = 1 \Rightarrow b = -1$$

$$1 \times ([0, w_1, w_2] \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + b = 1 \Rightarrow -w_2 + w_3 + b = 1$$

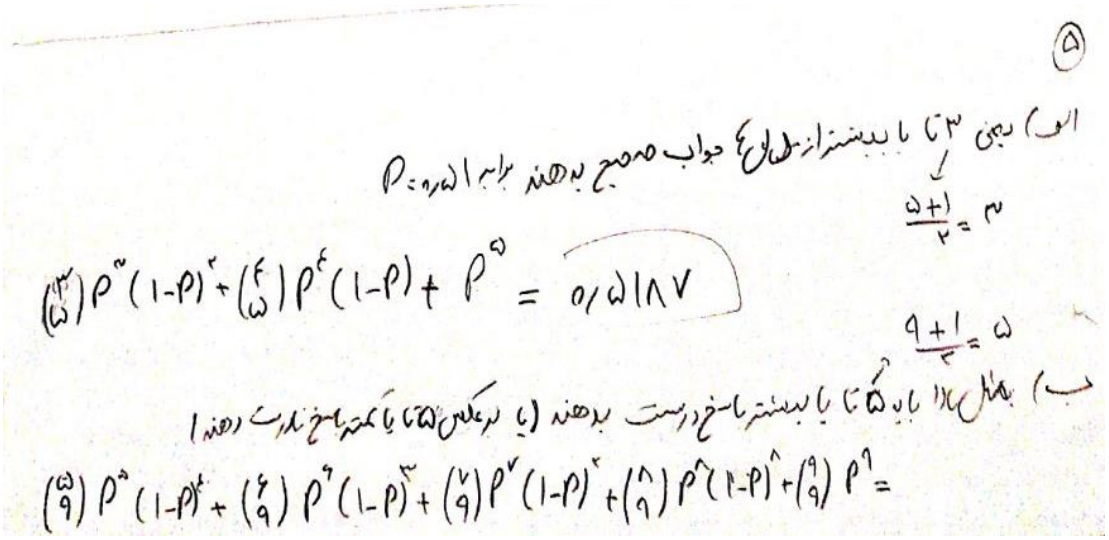
$$1 \times ([0, w_1, w_2] \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + b = 1 \Rightarrow w_2 + w_3 + b = 1$$

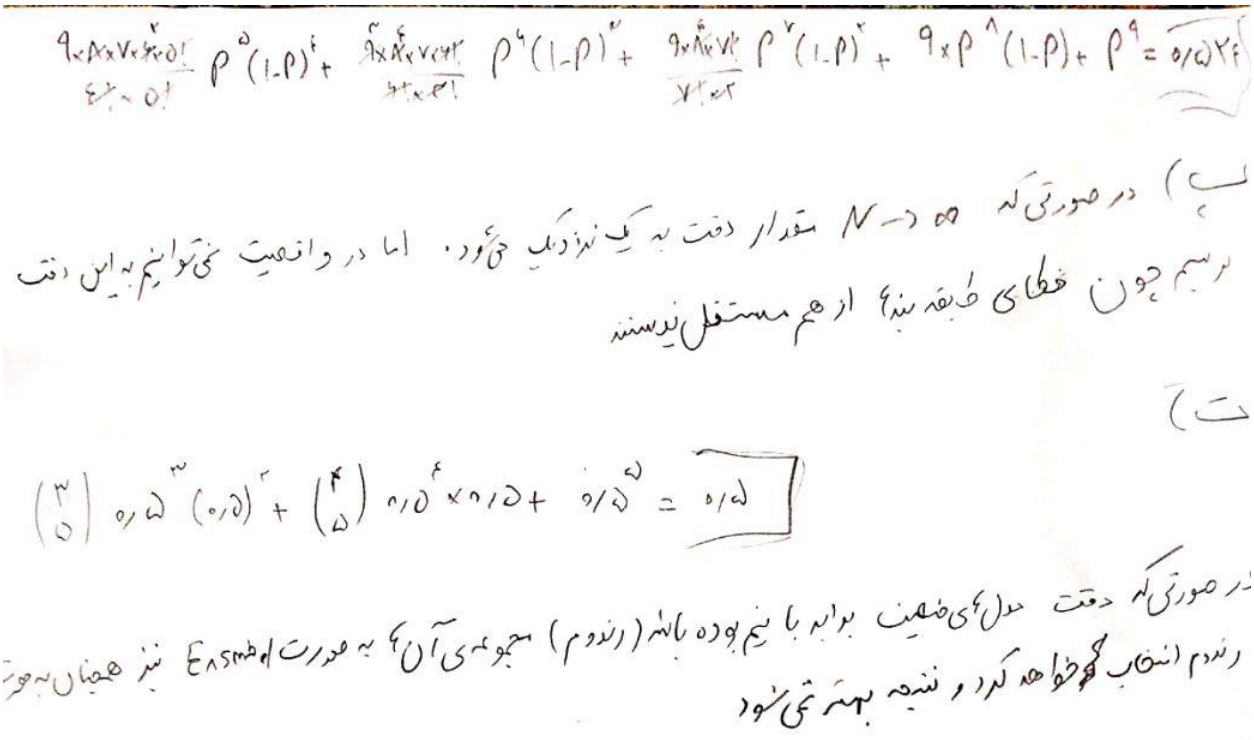
$$2w_3 = 2 \Rightarrow w_3 = 1$$

$$w_2 = 0$$

پس $\hat{w} = (0, 0, 1)$, $b = -1$ و اندازه فاصله بین بردارها $\frac{2}{|w|} = 1$ بوده و حاشیه بین $\frac{1}{2}$ می باشد

سوال ۵:





سوال ۶:

الف)

کرنل‌های linear، poly، rbf، sigmoid و precomputed در این کتابخانه موجود هستند که استفاده‌ی هرکدام به شرح زیر است:

Linear kernel:

کرنل خطی زمانی استفاده می شود که داده ها به صورت خطی قابل تفکیک باشند، یعنی با استفاده از یک خط می توان آنها را جدا کرد. این یکی از رایج ترین کرنل هایی است که مورد استفاده قرار می گیرد. کرنل خطی معمولاً در مجموعه داده‌هایی با مقادیر زیادی ویژگی استفاده می‌شود، زیرا افزایش ابعاد در این مجموعه داده‌ها لزوماً تفکیک‌پذیری را بهبود نمی‌بخشد.

Poly kernel:

در یک کرنل poly، داده ها با استفاده از یک تابع چند جمله ای در فضایی با ابعاد بالاتر نگاشت می شوند. حاصل ضرب نقطه ای نقاط داده در فضای اصلی و تابع چند جمله ای در فضای جدید گرفته می شود. کرنل چند جمله ای اغلب در مسائل طبقه بندی SVM استفاده می شود که در آن داده ها به صورت خطی قابل تفکیک نیستند. با نگاشت داده‌ها در فضایی با ابعاد بالاتر، کرنل چند جمله‌ای گاهی اوقات می‌تواند ابرصفحه‌ای را پیدا کند که کلاس‌ها را از هم جدا می‌کند. این کرنل به شکل زیر می‌باشد:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

Rbf kernel:

زمانی که مجموعه داده به صورت خطی غیرقابل تفکیک باشد یا به عبارت دیگر مجموعه داده غیرخطی باشد، توصیه می شود از کرنلی مانند RBF استفاده شود. این کمک می کند تا زمانی که هیچ دانش قبلی از داده ها وجود ندارد، جداسازی مناسب انجام شود. RBF از منحنی های نرمال در اطراف نقاط داده استفاده می کند و آنها را جمع می کند تا مرز تصمیم را بتوان با یک نوع شرایط توپولوژی تعریف کرد، مانند منحنی هایی که مجموع آنها بالاتر از مقدار ۰/۵ است. این کرنل معمولا به صورت دیفالت انتخاب میشه و خاصیت‌هایی مثل نرم (smooth)، ایستا و.. دارد که بهتر از سایر کرنل‌ها عمل می‌کند.

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

Sigmoid kernel:

کرنل Sigmoid از نظر تئوری برای یک ماشین بردار پشتیبان (SVM) پیشنهاد شده است، زیرا از یک شبکه عصبی سرچشمه می گیرد، اما تاکنون به طور گسترده در عمل مورد استفاده قرار نگرفته است. کرنل Sigmoid معمولاً مشکل ساز یا نامعتبر است زیرا شرط Mercer را در همه κ و θ برآورده نمی کند.

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

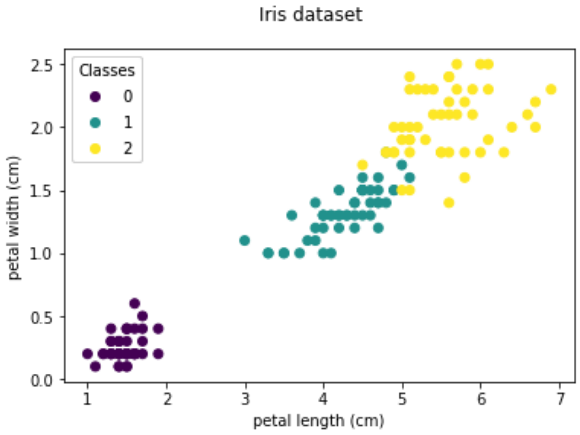
Precomputed kernel:

کرنل‌های svm در این کتابخانه تا حد زیادی کند هستند. با استفاده از این کرنل می‌توانیم کرنل‌هایی که قبلا حساب شده‌اند را برای

اموزش و ازمون بدهیم تا هم از زمان و هم از حافظه برای اموزش مدل خود بهتر استفاده کنیم.

ب)

داده‌های ما به صورت زیر پخش شده اند:



با توجه به شکل بالا می‌توانیم حدس بزنیم که کرنل خطی تا حد خوبی داده‌ها را جداسازی می‌کند.

حال داده‌ها را به دو دسته اموزش و آزمون تقسیم می‌کنیم (با نسبت ۰/۳۳) و برای هریک از کرنل‌های خواسته شده در صورت سوال اینکار را انجام می‌دهیم:

- SVM with Linear Kernel, one-vs-rest

```
The train accuracy for SVM with Linear Kernel, one-vs-rest=  0.91
The test accuracy for SVM with Linear Kernel, one-vs-rest=  0.92
```

- SVM with Linear Kernel, one-vs-one

```
The train accuracy for SVM with Linear Kernel, one-vs-one=  0.95
The test accuracy for SVM with Linear Kernel, one-vs-one=  1.0
```

- SVM with RBF Kernel, one-vs-rest

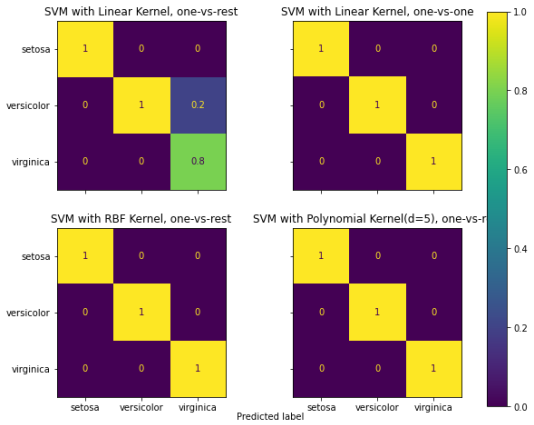
```
The train accuracy for SVM with RBF Kernel, one-vs-rest=  0.95
The test accuracy for SVM with RBF Kernel, one-vs-rest=  1.0
```

- SVM with Polynomial Kernel (d=5), one-vs-rest

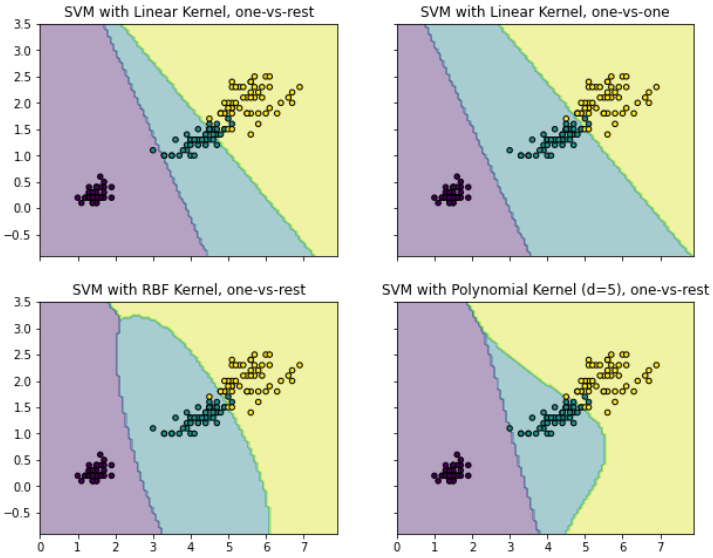
```
The train accuracy for SVM with Polynomial Kernel (d=5), one-vs-
rest=  0.96
The test accuracy for SVM with Polynomial Kernel (d=5), one-vs-rest=

1.0
```

دلیل اینکه دقت در داده‌های اموزش کمتر از داده‌های تست می‌باشد فقط به خاطر تصادفی بودن انتخاب این داده‌هاست و این دقت ممکن است تغییر کند.

ماتریس درهمریختگی به شکل زیر خواهد بود:



و همچنین مرزهای تصمیم گیری به شکل‌های زیر خواهد بود:



مشخص است که مدل هر سه مدل خطی و poly و rbf نسبتا خوب مرزهای تصمیم گیری را جداسازی کرده‌اند و با توجه به نتایج مدل خطی، نیاز به مدل‌های پیچیده تر همچون rbf نداشتیم. مدل poly نیز توانسته با یک شکل سهمی وار این نقاط را از هم جدا کند. ناحیه تصمیم گیری مدل one vs all نسبت به مدل one vs one کمی فشرده تر شده است. در این نوع تصمیم گیری چند کلاسه، هربار یک لیبل مثبت را دربرابر همه‌ی لیبل‌های منفی مقایسه می‌کنیم برای همین تعداد لیبل‌های منفی هربار بیشتر از حالت عادی است که باعث فشرده تر شدن ناحیه شده است.

در اینجا نمونه‌ی آموزش برای یه نوع شافل دیگر از داده‌ها هم قرار می‌دهم:

- SVM with Linear Kernel, one-vs-rest

```
The train accuracy for SVM with Linear Kernel, one-vs-rest=  0.94

The test accuracy for SVM with Linear Kernel, one-vs-rest=  0.86
```

- SVM with Linear Kernel, one-vs-one

```
The train accuracy for SVM with Linear Kernel, one-vs-one=  0.98
The test accuracy for SVM with Linear Kernel, one-vs-one=  0.96
```

- SVM with RBF Kernel, one-vs-rest

```
The train accuracy for SVM with RBF Kernel, one-vs-rest=  0.97
The test accuracy for SVM with RBF Kernel, one-vs-rest=  0.96
```

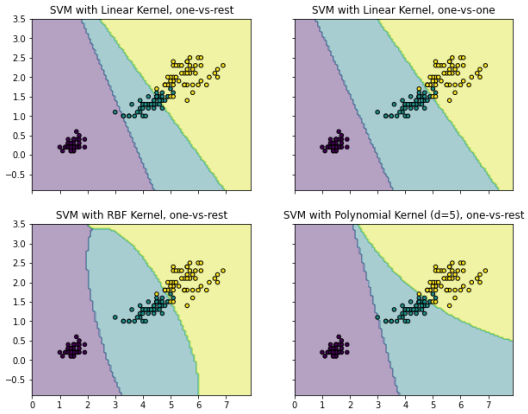
- SVM with Polynomial Kernel (d=5), one-vs-rest

```
The train accuracy for SVM with Polynomial Kernel (d=5), one-vs-
rest=  0.95
The test accuracy for SVM with Polynomial Kernel (d=5), one-vs-rest=

0.96
```

ماتریس درهم ریختگی:

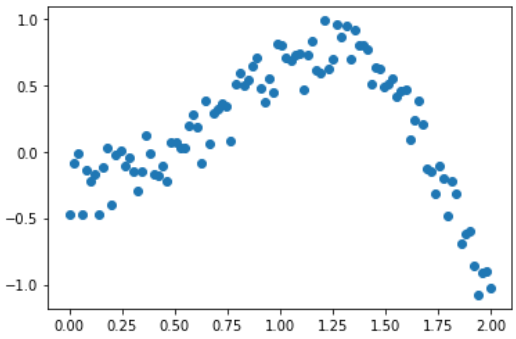
نتایج تقریبا همانند بالاست، فقط با توجه به داده‌ی تست جدید مدل در کلاس ۱ و ۲ (زرد و ابی) که داده‌ها نزدیک بهم هستند تعدادی داده را اشتباه طبقه بندی کرده است.

ناحیه تصمیم گیری:



سوال ۷:

نمودار داده‌های ما به شکل زیر است:



حال برای کرنل خطی مقادیر C برابر با [1, 10, 100, 1000] برای مدل در نظر می‌گیریم.

همینطور برای کرنل چند جمله ای مقادیر زیر را در نظر می‌گیریم:

```
'C': [1, 10, 100, 1000], 'gamma': [0.0001,0.001,0.1,1,10],
'coef0':[0.1,1,5,10]
```

Coef0 برای نامتقارن کردن کرنل استفاده می‌شود.

برای کرنل rbf مقادیر زیر را در نظر می‌گیریم:

```
'C': [1, 10, 100, 1000], 'gamma': [0.0001,0.001,0.1,1,10]
```

و مدل‌های خود را با کراس ولیدیشن ۵ آموزش می‌دهیم، که نتایج به صورت زیر خواهد شد:

```
The best parameters for linear are: {'C': 1, 'kernel': 'linear'}
The best parameters for poly are: {'C': 1, 'coef0': 0.1, 'gamma': 10,
'kernel': 'poly'}
   The best parameters for rbf are: {'C': 10, 'gamma': 1, 'kernel': 'rbf'}
```

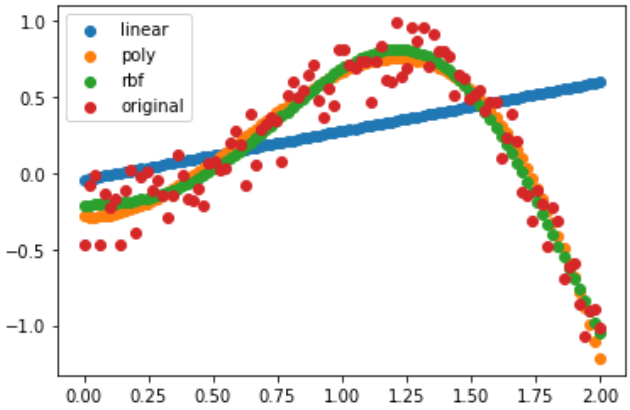
که نتایج مدل‌ها به صورت زیر خواهد بود:

```
The coefficient of determination for the linear classifer is: -
0.1765743645776896
The coefficient of determination for the poly classifer is:
0.9113147506100493
The coefficient of determination for the rbf classifer is:
0.9203043104715829
```

و همچنین خطا به صورت زیر است:

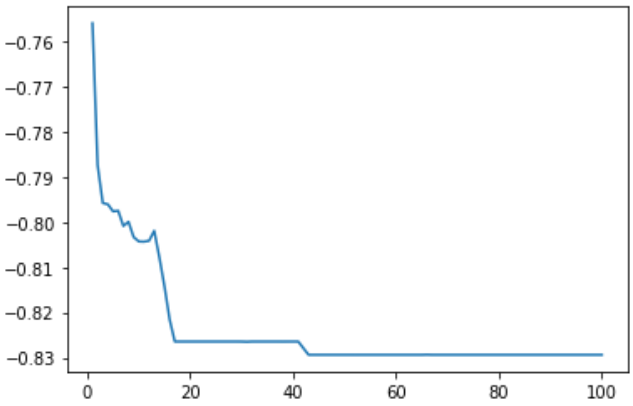
```
The mean_squared_error for the linear classifer is: 0.2817008417424394
The mean_squared_error for the poly classifer is: 0.021233429994248064
The mean_squared_error for the rbf classifer is: 0.019081108257409087
```

و نتایج در نمودار به صورت زیر خواهد بود:



با توجه به نمودار بالا و مقدار خطا وcoefficient of determination مشخص است که کرنل rbf از دو کرنل دیگه بهتر عمل کرده است، که با توجه به شکل خاص sin این پیش بینی میشد. همینطور کرنل poly از کرنل خطی بهتر عمل کرده و تقریبا به شکل sin فیت شده است که این به خاطر ضریب coef10 است که شکل منحنی را نامتقارن کرده است و توان سه قابلیت به فیت شدن به شکل بالا را داشت اما اگه بخش بزرگتر از sin به عنوان داده‌ها انتخاب میشد، poly نیز بهتر نمی‌توانست جواب بدهد. کرنل خطی نیز بدترین کرنل بوده و برای شکل sin کافی نیست.

البته می‌توان کارهای بالا را بدون استفاده از کتابخانه نیز انجام داد و نحوه کم شدن ارور را دید، برای مثال برای کرنل خطی بهترین c را با توجه به شکل زیر می‌توان پیدا کرد:



که با توجه به نمودار بالا با افزایش c مقدار خطا در حال افزایش است پس c=1 بهترین انتخاب است.