

به نام خدا

راضیه مسعودی 9935598

شیرین محبی 9935537

تمرین چهارم یادگیری ماشین

## پارت اول :

ابتدا کتابخانه های مورد نیاز را اضافه می کنیم سپس دیتاست ها را لود کرده آن ها را به float تبدیل و با کم کردن داده های هر ستون از میانگین آن ستون، تقسیم بر انحراف معیار آن ستون، داده ها را نرمال کرده و 30% از داده ها را به عنوان داده ی تست در نظر می گیریم و بقیه را داده ی ترین در نظر می گیریم، همچنین داده های دیتاست های Glass، Diabetes، BreastTissue، Sonar، Wine و Ionosphere را به ترتیب با شماره های از یک تا شش نشان می دهیم (هم داده تست هم ترین).

تابع one\_NN: این تابع فاصله ی اقلیدسی هر داده ی تست را با همه ی داده های ترین به دست آورده و لیبل داده ی ترینی که کمترین فاصله را با آن داده ی تست دارد به عنوان لیبل داده ی تست در نظر می گیرد و در نهایت با لیبل اصلی داده ی ترین مقایسه کرده و دقت و زمان اجرای الگوریتم را بر می گرداند.

تابع linear\_1NN: این تابع در فرمول یک به جای K فرمول linear جدول یک را می گذارد و فرمول نهایی را پیاده سازی می کند و لیبل داده ی ترینی که کمترین فاصله (dist) را با آن داده ی تست دارد به عنوان لیبل داده ی تست در نظر می گیرد و در نهایت با لیبل اصلی داده ی ترین مقایسه کرده و دقت و زمان اجرای الگوریتم را بر می گرداند؛ لازم به ذکر است  $c=1$  گرفتیم و  $X.T@X$  می شود نرم دوم X به توان 2.

فرمول یک:

$$K(x_i, x_i) + K(x_{i'}, x_{i'}) - 2K(x_i, x_{i'})$$

جدول یک:

<b>Linear</b>	$k(x, x') = x^T x' + c$
<b>polynomial</b>	$k(x, x') = (\alpha x^T x' + c)^d$
<b>Exponential</b>	$k(x, x') = \exp(\frac{\ x - x'\ }{2\sigma^2})$
<b>Gaussian</b>	$k(x, x') = \exp(\frac{\ x - x'\ ^2}{2\sigma^2})$
<b>power</b>	$k(x, x') = -\ x - x'\ ^d$

تابع polynomial\_1NN: این تابع در فرمول یک به جای K فرمول polynomial جدول یک را می گذارد و فرمول نهایی را با توجه به مقادیر d داده شده در فایل هوم ورک پیاده سازی می کند و لیبل داده ی ترینی که کمترین فاصله (dist) را با آن داده ی تست دارد به عنوان لیبل داده ی تست در نظر می گیرد و در نهایت با لیبل اصلی داده ی ترین مقایسه کرده و دقت و زمان اجرای الگوریتم را بر می گرداند؛ لازم به ذکر است  $c=1$  گرفتیم و [X.T@X](#) می شود نرم دوم X به توان 2.

تابع RBF\_1NN: این تابع در فرمول یک به جای K فرمول زیر را می گذارد و فرمول نهایی را با توجه به مقدار بهترین سیگما به دست آمده بین صفر و یک مبتنی بر

دقت پیاده سازی می کند و لیبل داده ی ترینی که کمترین فاصله (dist) را با آن داده ی تست دارد به عنوان لیبل داده ی تست در نظر می گیرد و در نهایت با لیبل اصلی داده ی ترین مقایسه کرده و دقت و زمان اجرای الگوریتم را بر می گرداند.

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

سپس توابع بالا را برای دیتاست های مختلف فراخوانی کرده و دقت و زمان اجرای هر تابع را به دست می آوریم.

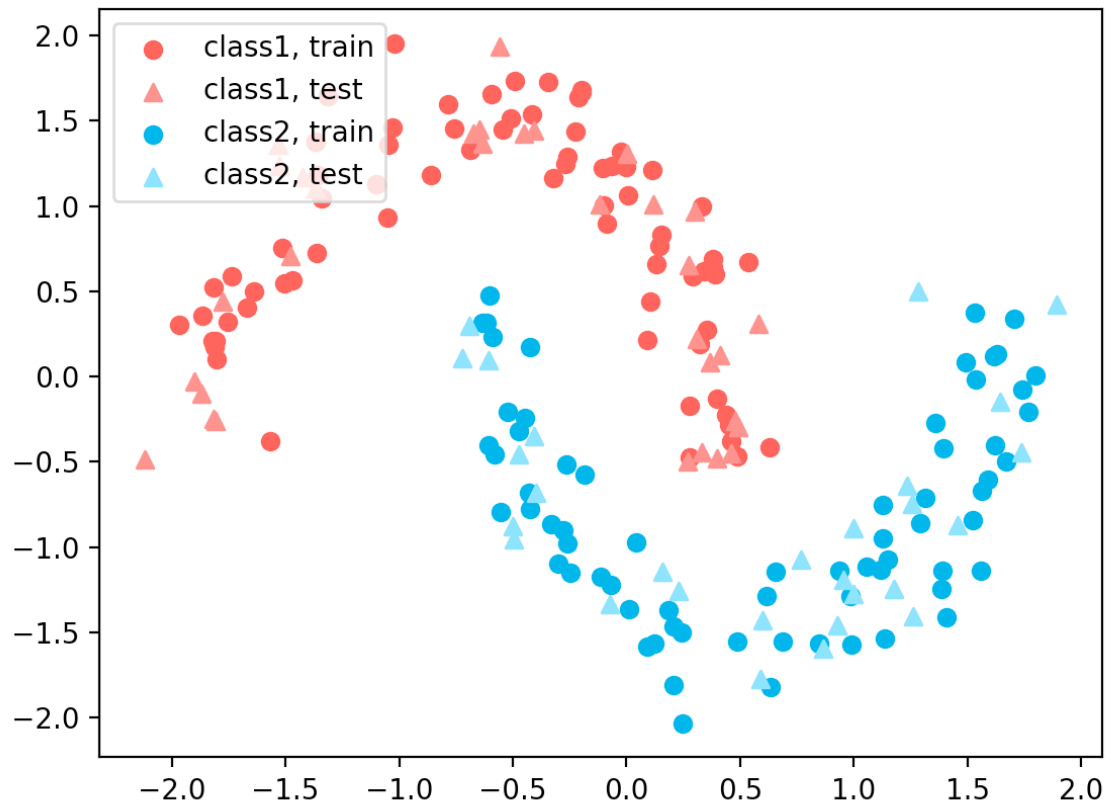
Dataset	1NN	1NN+Linearkernel	1NN+RBFkernel	1NN+Polynomialkernel (d = 1)	1NN+Polynomialkernel (d = 2)	1NN+Polynomialkernel (d = 3)
BreastTissue	71.875000	71.875000	68.750000	71.875000	71.875000	71.875000
Diabetes	69.696970	69.696970	69.696970	69.696970	68.398268	69.264069
Glass	75.384615	75.384615	75.384615	75.384615	73.846154	69.230769
Ionosphere	83.018868	83.018868	79.245283	83.018868	76.415094	76.415094
Sonar	92.063492	92.063492	90.476190	92.063492	80.952381	74.603175
Wine	92.592593	92.592593	92.592593	92.592593	96.296296	94.444444

	Dataset	1NN	1NN+Linearkernel	1NN+RBFkernel	1NN+Polynomialkernel ( $d = 1$ )	1NN+Polynomialkernel ( $d = 2$ )	1NN+Polynomialkerr ( $d =$
0	BreastTissue	0.005001	0.003994	0.005996	0.003997	0.003999	0.0039
1	Diabetes	0.020987	0.031981	0.061965	0.033979	0.035978	0.0709
2	Glass	0.003996	0.009994	0.016987	0.008997	0.012990	0.0129
3	Ionosphere	0.010993	0.014993	0.026987	0.014989	0.014991	0.0199
4	Sonar	0.005996	0.008995	0.015992	0.008995	0.008995	0.0109
5	Wine	0.002998	0.005996	0.009999	0.005996	0.007994	0.0079

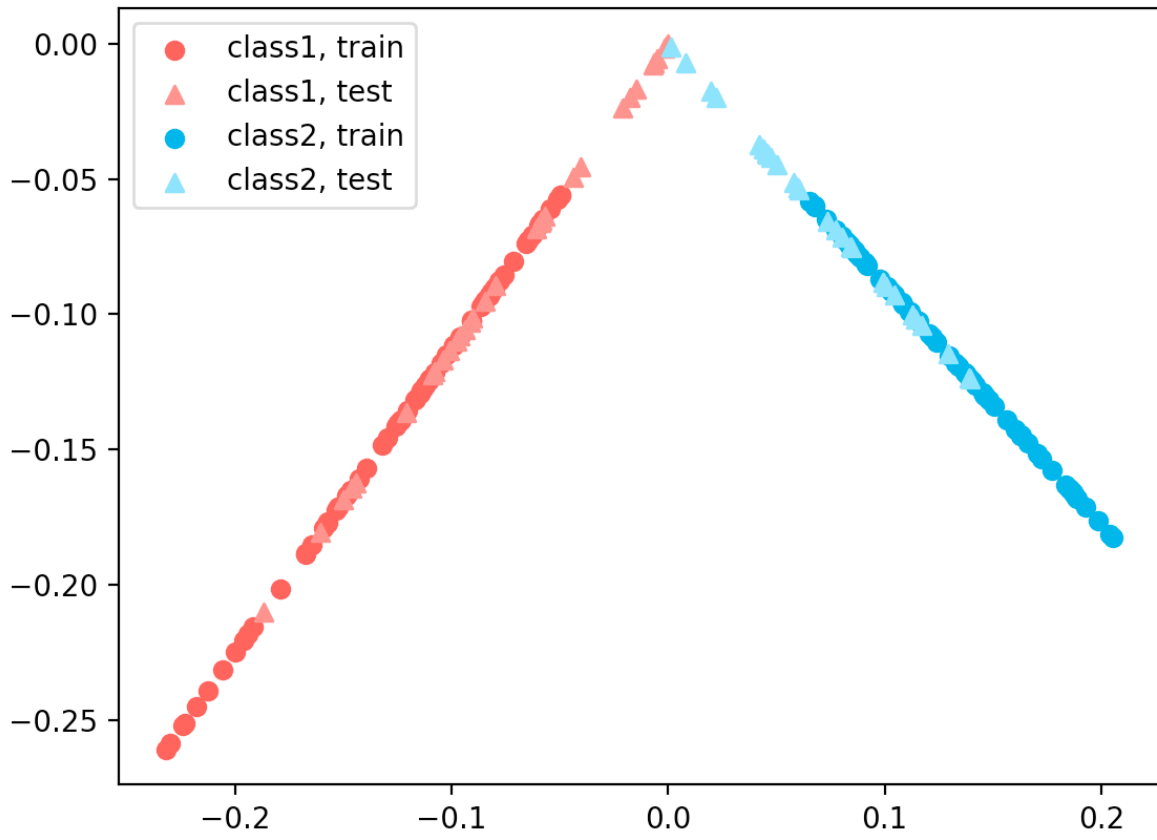
## پارت دوم:

ابتدا دیتا را از فایل خوانده و فیچرها و لیبل ها را از هم جدا میکنیم. سپس فیچرها را زیرو مین کرده و تقسیم بر انحراف معیار میکنیم تا نرمال شوند. بعد طبق سودوکد، ماتریس ها را می سازیم و در هم ضرب میکنیم. با استفاده از لایبرری اسکیکپای، و تابع ایگن، ایگن وکتور های ماتریس داده شده را پیدا کرده و آن دو ایگن وکتوری که بیشترین مقدار وزن را دارند را انتخاب کرده و دیتا را با استفاده از این ایگن وکتورها به دو بعد می بریم، و سپس این دو بعد را نمایش می دهیم. سیگما های مختلف را باید امتحان کنیم تا ببینیم در کدام حالت دو کلاس بیشترین جداسازی را از یکدیگر دارند. که پس از آزمون و خطا به این نتیجه می رسیم که بهترین سیگما برای تمام دیتاست ها یک دهم می باشد.

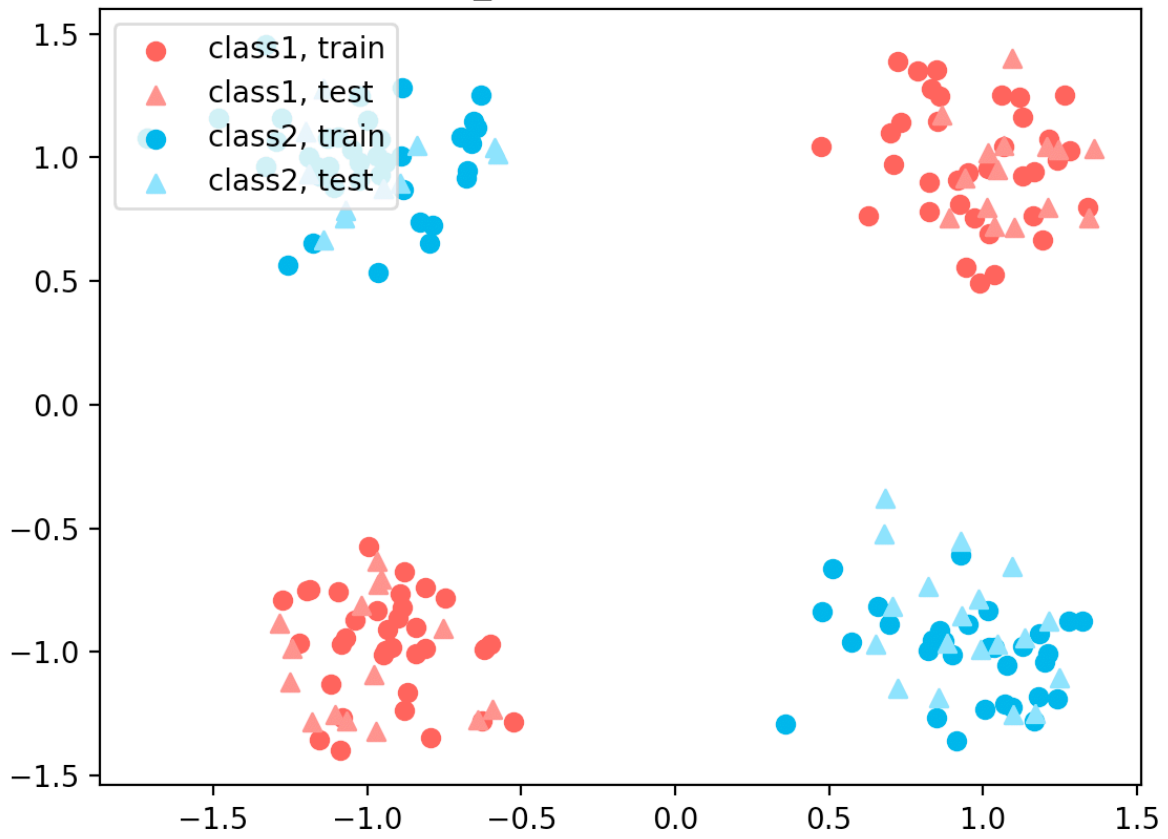
Twomoons.txt original space



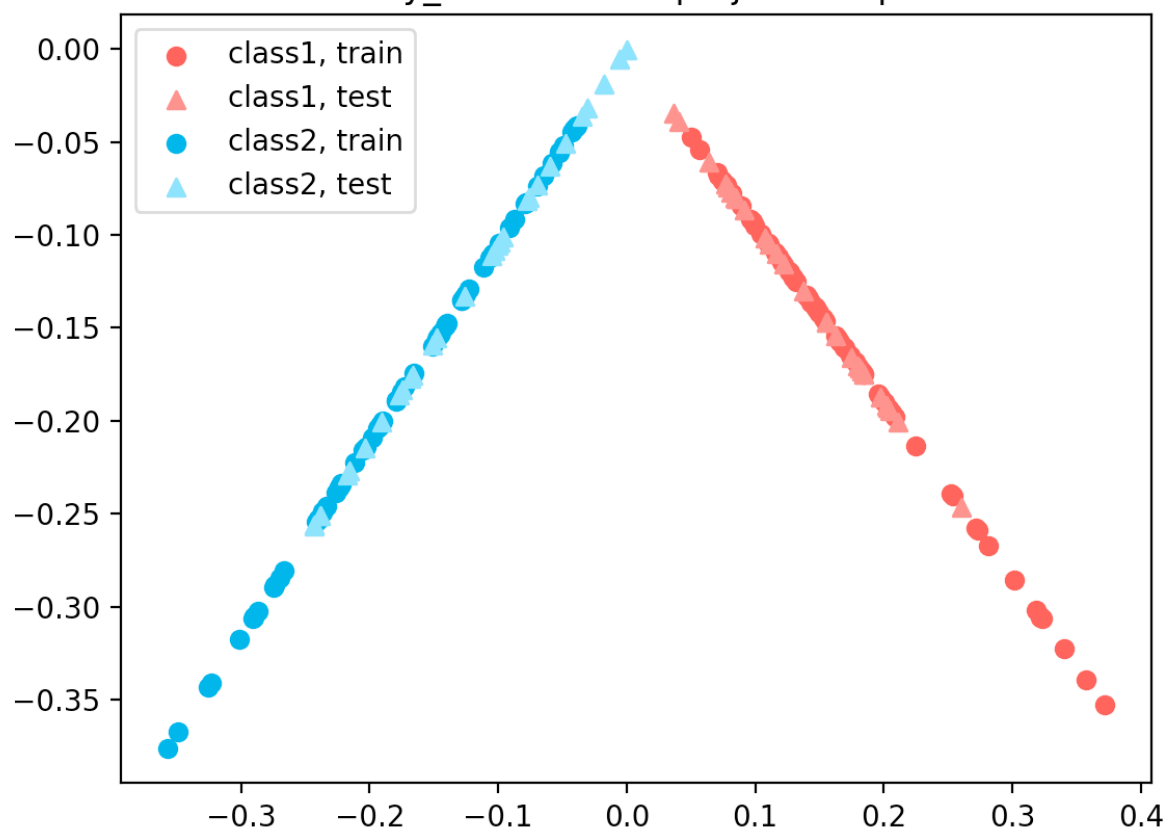
Twomoons.txt after projection space



Binary\_XOR.txt original space

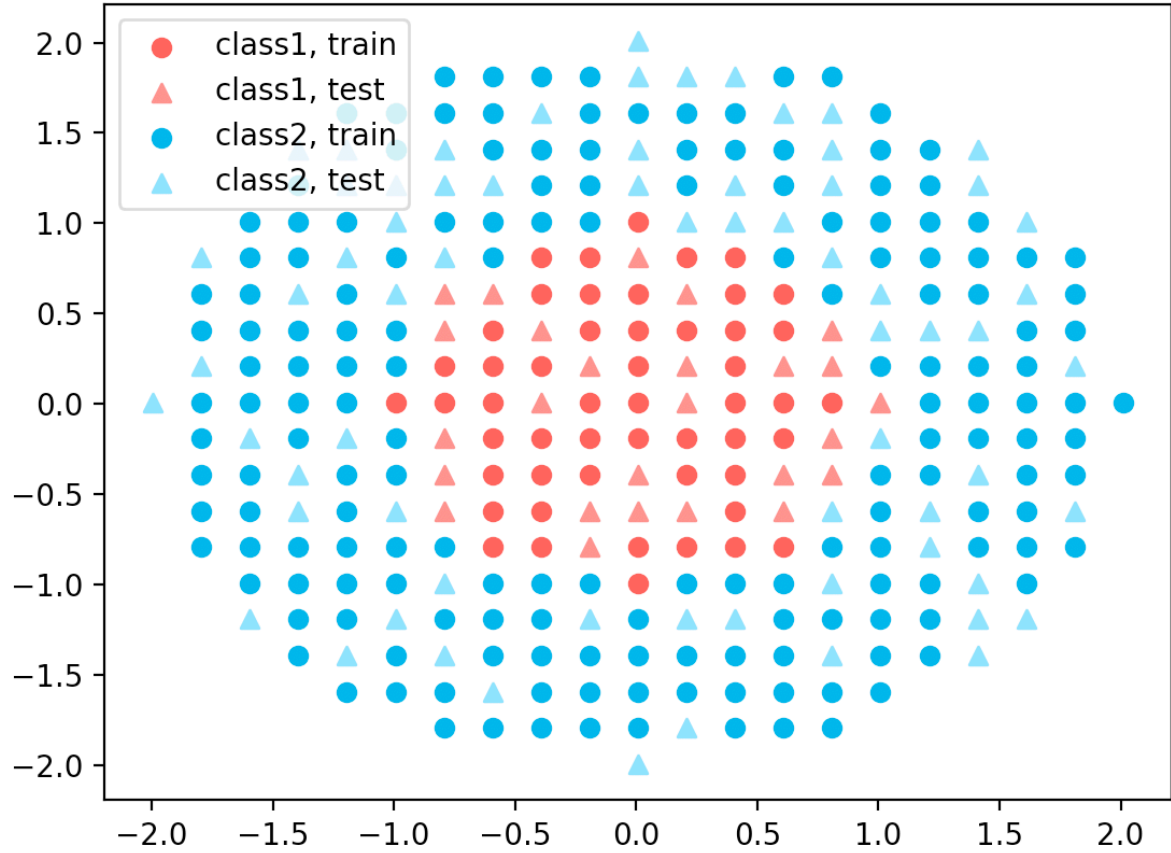


Binary\_XOR.txt after projection space

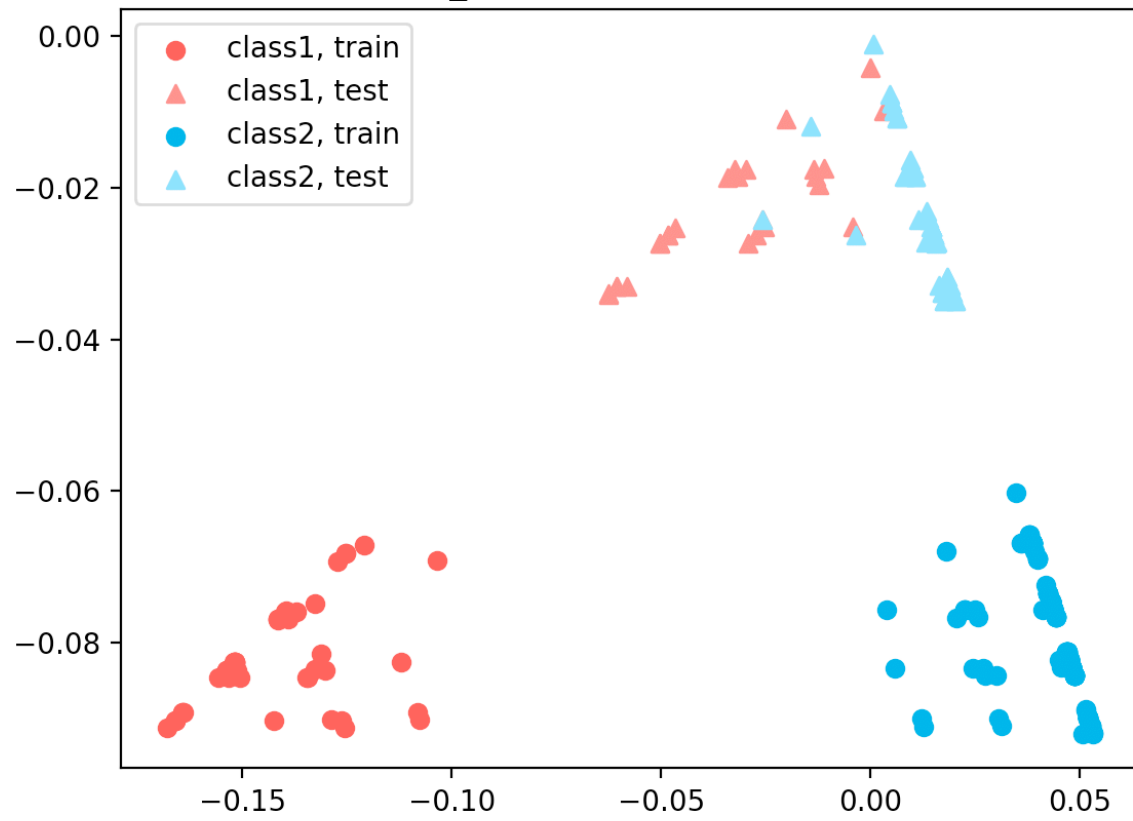




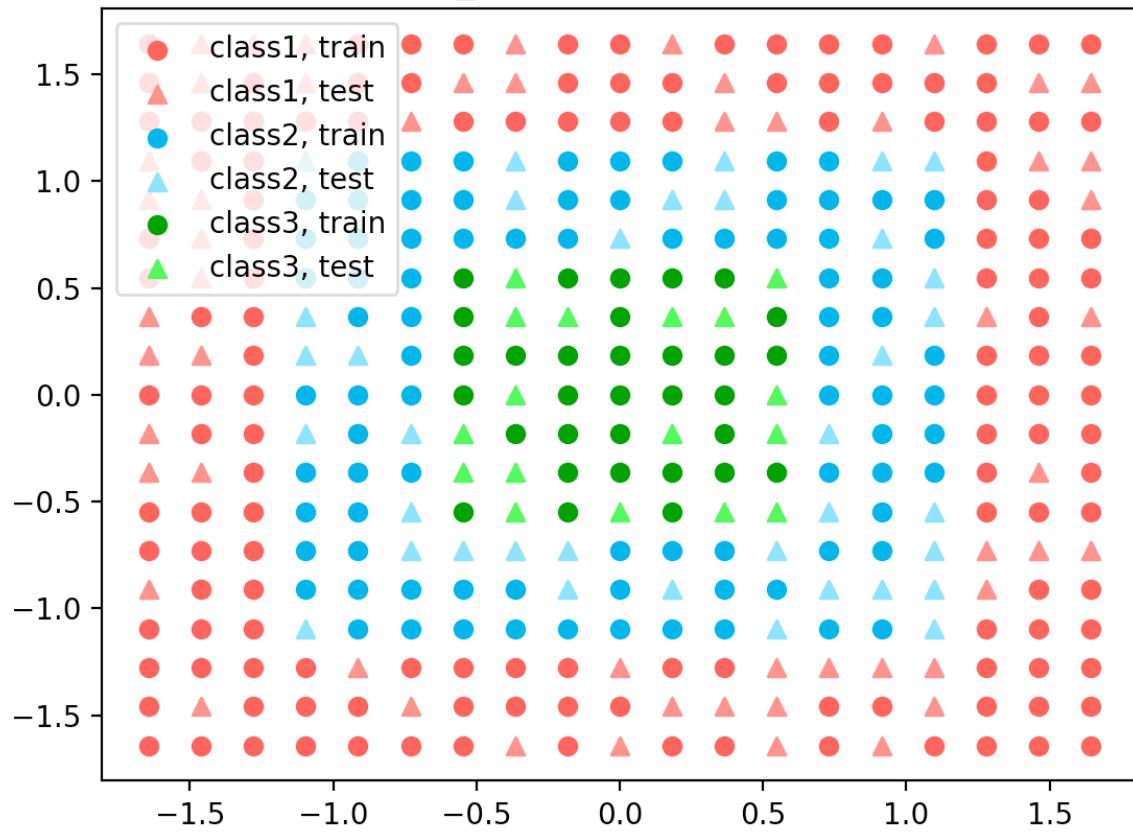
Concentric\_rings.txt original space

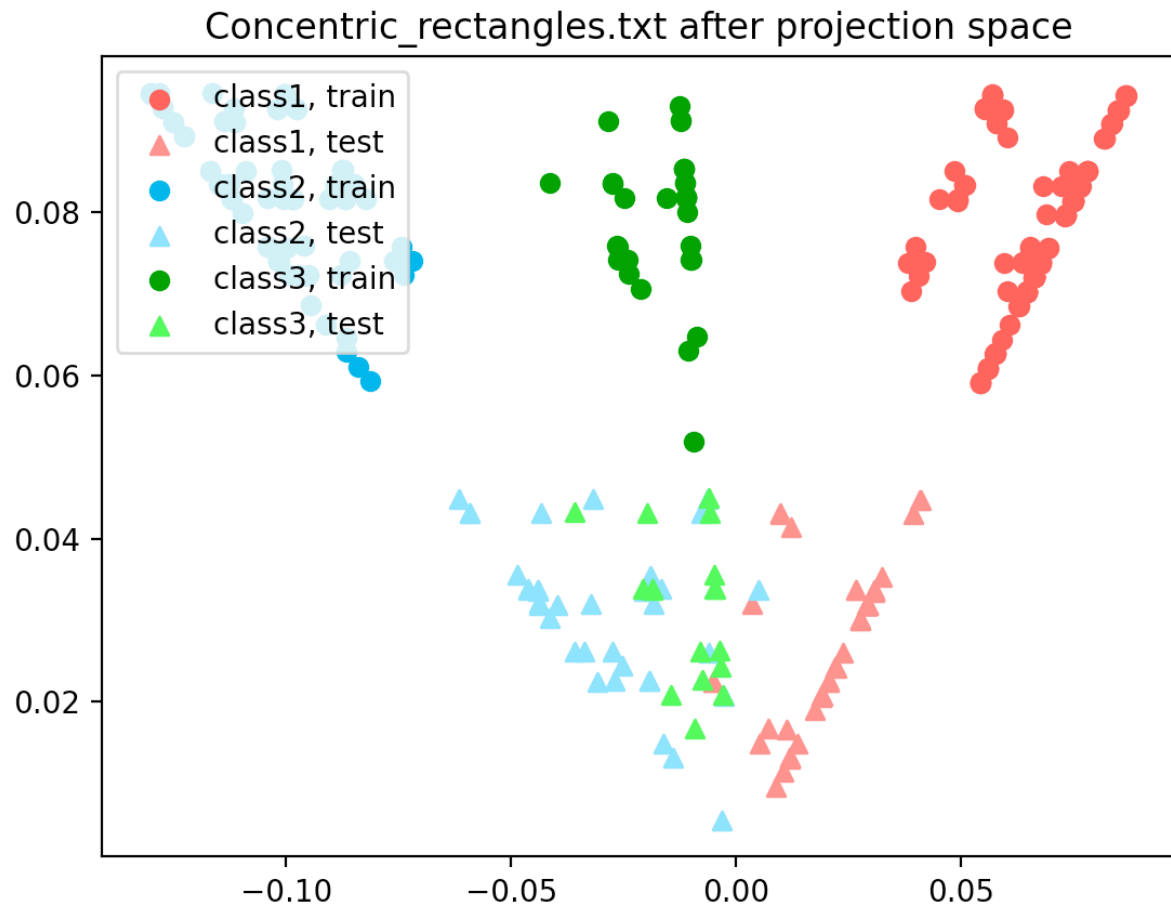


Concentric\_rings.txt after projection space



Concentric\_rectangles.txt original space





Why do we use kernels in different algorithms?

Using kernels, we do not need to embed the data into the Feature space explicitly, because a number of algorithms only require the inner products between the mapped vectors

Which kernel had the best results? Why?

به طور متوسط کرنل های خطی،  $\text{polynomial}(d=1)$  و اقلیدسی عملکرد بهتری داشتند اما در نهایت چون داده ها پیوسته و از توزیع نرمال بودند (داده ها را نرمال کرده ایم) پس انتظار می رود کرنل RBF بهترین عملکرد را داشته باشد زیرا هم متداول تر است و هم دارای پاسخ محلی و متناهی در کل محور  $x$  است.

Which kernels had the same performance? Why?

کرنل خطی، اقلیدسی و  $\text{polynomial}$  با  $d=1$  عملکرد مشابه به هم دارند زیرا کرنل  $\text{polynomial}$  با  $d=1$  عملاً همان کرنل خطی است.

What is kernel trick? Why do we use kernels in learning algorithm?

کرنل تریک این امکان را به ما می دهد که بدون محاسبه مختصات داده ها در فضایی با ابعاد بالاتر، در فضای اصلی ویژگی کار کنیم. توابع کرنل ضرب داخلی بین دو نقطه را با تعریف مفهومی از شباهت، با هزینه محاسباتی اندک حتی در فضاهایی با ابعاد بالا در یک  $\text{feature space}$  مناسب برمی گردانند.

What is the difference between Delta and RBF kernels? Explain about the properties of these kernels.

delta kernel usually uses over labels. and show the similarity of 2 labels. if they are the same, their respective delta kernel is 1 otherwise it's 0. so there is no real value for this kind of similarity. but RBF uses to show similarity of features. and it's value is real. and it uses gaussian kernel function and it shows distance between  $x_i$  and  $x_j$  based on  $\sigma$ .

