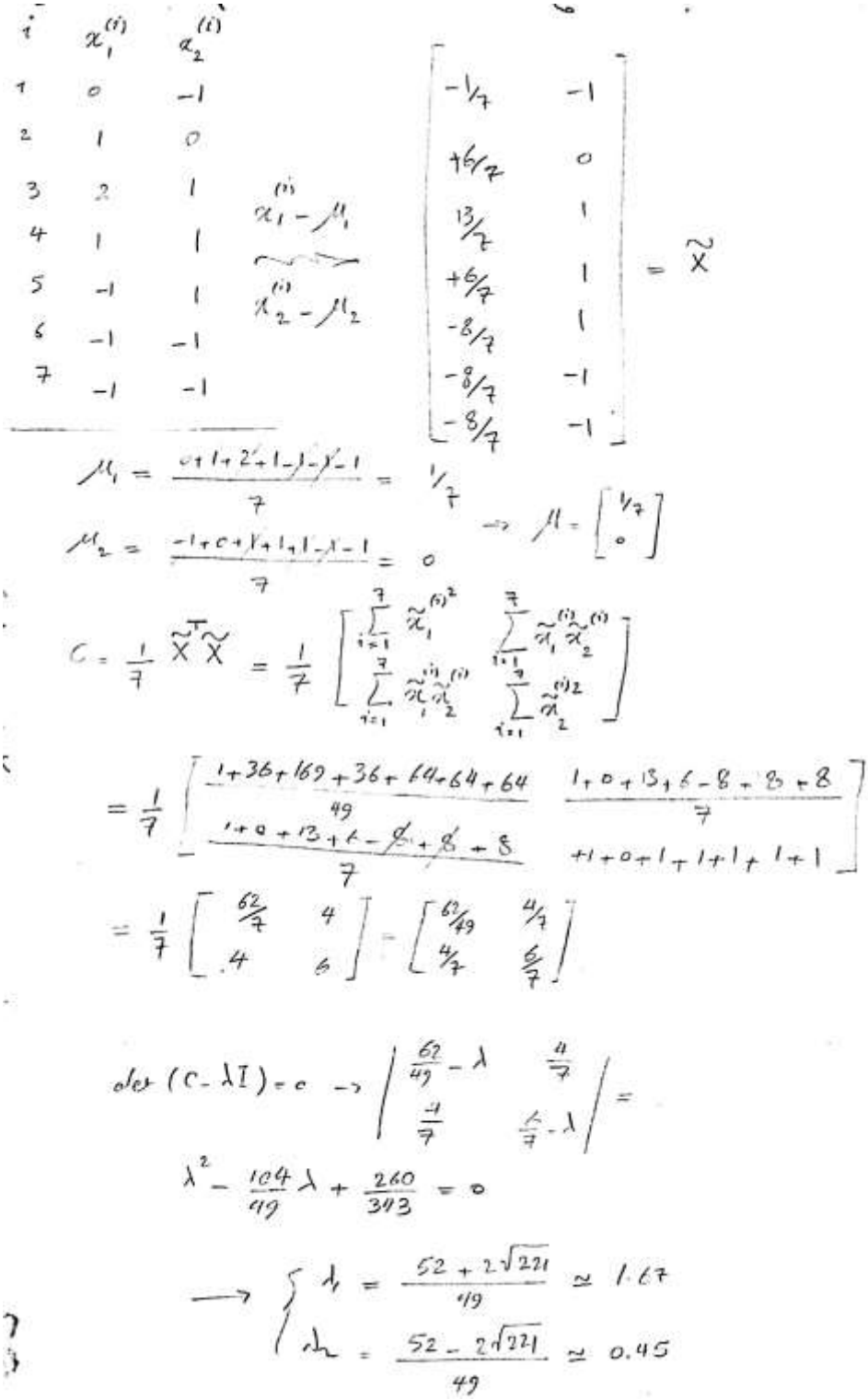
فایل ضمیمه : **hw4.ipynb**

سوال ۱ :



$$CV_1 = \lambda_1 V_1$$

$$\begin{bmatrix} \frac{12}{49}v_1 + \frac{4}{7}v_2 \\ \frac{4}{7}v_1 + \frac{6}{7}v_2 \end{bmatrix} = \begin{bmatrix} 1.67\, v_1 \\ 1.67\, v_2 \end{bmatrix}$$

$$\downarrow$$

$$\simeq \begin{bmatrix} 1.26v_1 + 0.57v_2 \\ 0.57v_1 + 0.86v_2 \end{bmatrix} \Rightarrow \begin{array}{l} -0.41v_1 = -0.57v_2 \\ 0.57v_1 = +0.81v_2 \end{array}$$

$$\Rightarrow v_2 = 0.91\, v_1 \Rightarrow \begin{bmatrix} 1.4 \\ 1 \end{bmatrix} = V_1 \rightsquigarrow \text{1st PC}$$

$$CV_2 = \lambda_2 V_2$$

$$\begin{bmatrix} 1.26v_1 + 0.57v_2 \\ 0.57v_1 + 0.86v_2 \end{bmatrix} = \begin{pmatrix} 0.45v_1 \\ 0.45v_2 \end{pmatrix} \Rightarrow \begin{array}{l} 1.81\, v_1 = -057 v_2 \\ 0.57\, v_1 = -0.41 v_2 \end{array}$$

$$\Rightarrow v_2 = -1.9\, v_1 \Rightarrow \begin{pmatrix} -0.7 \\ 1 \end{pmatrix} = V \rightsquigarrow \text{2nd PC}$$

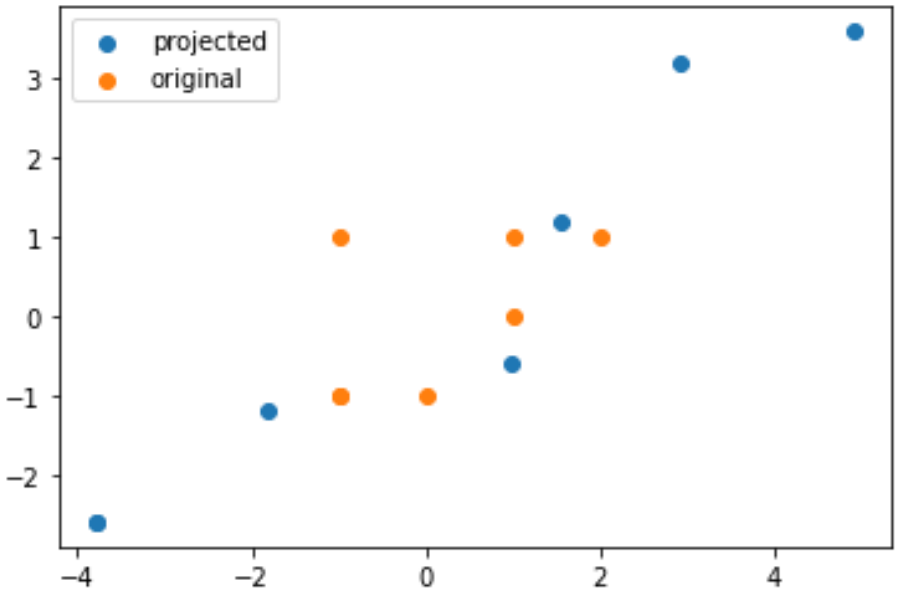
$$A = V_1$$

$$X' = \tilde{X}A = \begin{bmatrix} -\frac{1}{7} \times \frac{14}{10} - 1 = -1.2 \\ \frac{6}{7} \times \frac{14}{10} - 0 = 1.2 \\ \frac{13}{7} \times \frac{14}{10} + 1 = 3.6 \\ \frac{6}{7} \times \frac{14}{10} + 1 = 3.2 \\ -\frac{8}{7} \times \frac{14}{10} + 1 = -0.6 \\ -\frac{8}{7} \times \frac{14}{10} - 1 = -2.6 \\ -\frac{8}{7} \times \frac{14}{10} - 1 = -2.6 \end{bmatrix}$$

$$A^T = V_1^T = \begin{bmatrix} 1.4 & 1 \end{bmatrix}$$

$$X_{re} = X' A^T - \begin{bmatrix} \mu_1 & \mu_2 \\ \mu_1 & \mu_2 \\ \mu_1 & \mu_2 \end{bmatrix}_{7\times 2}$$

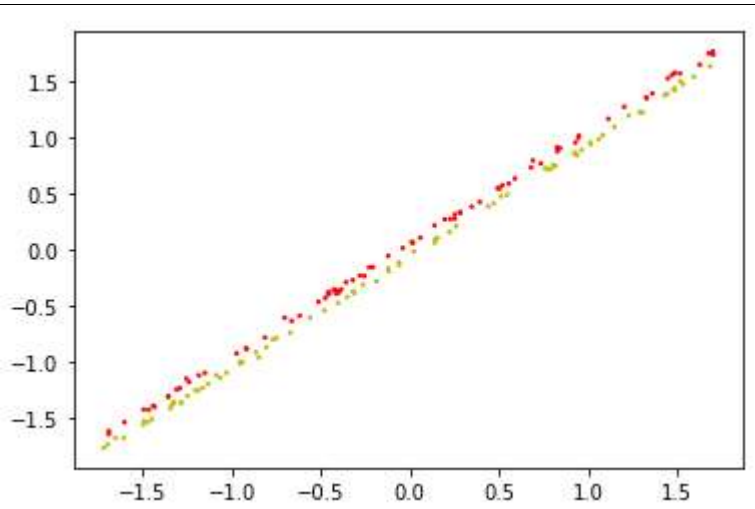
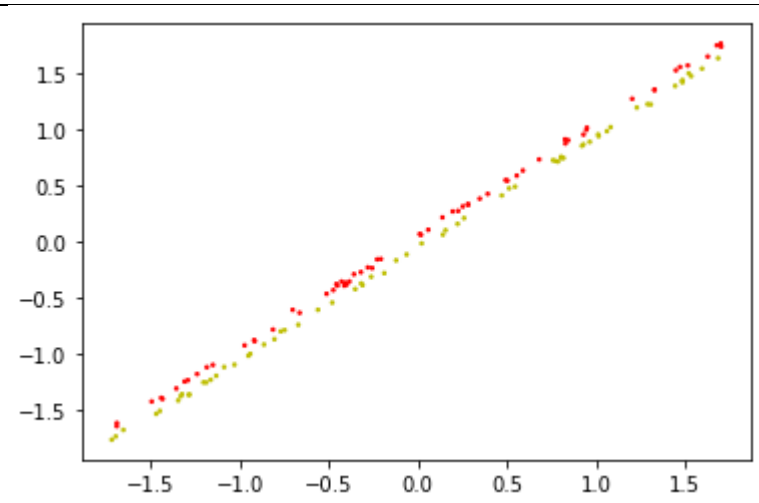
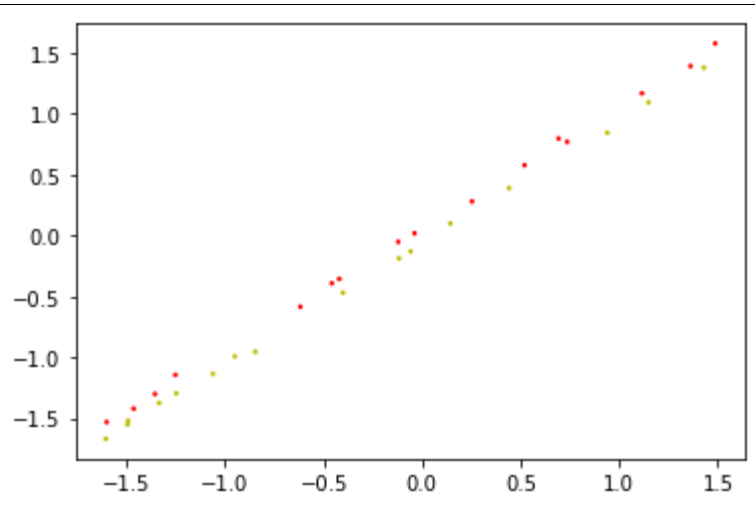
$$= \begin{bmatrix} -1.68-\mu_1 & -1.2-\mu_2 \\ 1.68-\mu_1 & 1.2-\mu_2 \\ 5.04-\mu_1 & 3.6-\mu_2 \\ 3.08-\mu_1 & 3.2-\mu_2 \\ -0.84-\mu_1 & -0.6-\mu_2 \\ -3.64-\mu_1 & -2.6-\mu_2 \\ -3.64-\mu_1 & -2.6-\mu_2 \end{bmatrix} = \begin{bmatrix} -1.82 & -1.2 \\ 1.54 & 1.2 \\ 4.9 & 3.6 \\ 2.93 & 3.2 \\ 0.98 & -0.6 \\ -3.78 & -2.6 \\ -3.78 & -2.6 \end{bmatrix}$$



سوال ۲ :

ویژگی ها و برچسب های هردو دسته دادگان تست و آموزش را به صورت ماتریس های داده ی numpy خواندم. برچسب های ۰ و ۱ را به -۱ و ۱ تبدیل کردم (فقط برای راحتی خطاب داده ها) و کلاس -۱ کلاس قرمز و کلاس +۱ را کلاس زرد می‌نامم. قبل از انجام هرکار داده ها را با StandardScaler از sklearn استاندارد می‌کنم. این استانداردسازی با براورد میانگین و واریانس داده های آموزش به عنوان میانگین و واریانس مجهول توزیع انجام شده. و روی داده های تست هم همین مدل استفاده می‌شود.

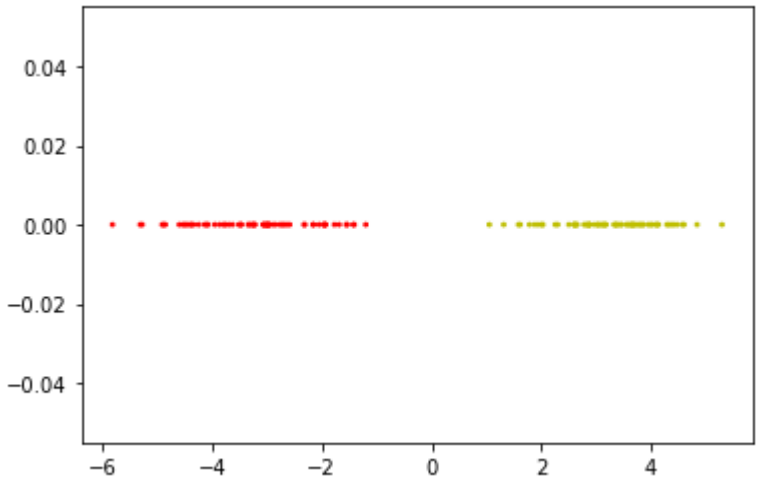
در قدم اول نمایش داده های تست و آموزش و هر دو باهم :

All data	
Train data	
Test data	

با استفاده از تابع LinearDiscriminantAnalysis از مجکوعه کتابخانه ی sklearn یک مدل LDA روی داده های آموزش برازش می‌دهم. بردار راستای LDA به دست می‌آید :

```
[ 386.49722, -386.96594]
```

که داده های دوبعدی را به فضای تک بعد می‌برد. داده ها ی آموزش روی این فضای تک بعدی:



واضح است که داد ها را می‌توان کاملا با تعین یک عدد آستانه از هم تفکیک کرد. برای تعین این مقدار آستانه از یک پرسپترون باز هم از کتابخانه ی sklearn استفاده می‌کنم. این مدل perc1 بعد از برازش روی داده های آموزش، برابر با بردار زیر می‌شود:

```
[2.49924668        1.]
```

عدد ۱ بایاس مدل است. حالا با برازش این بردار مدل روی داده های آموزش دقت آموزش برابر می‌شود با :

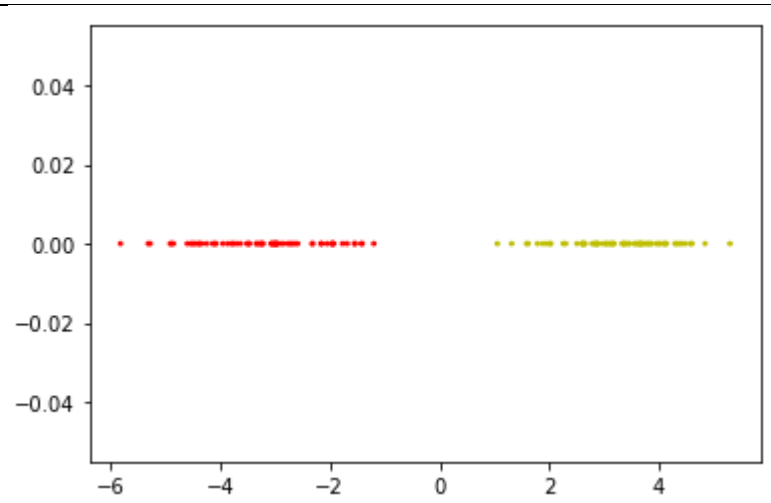
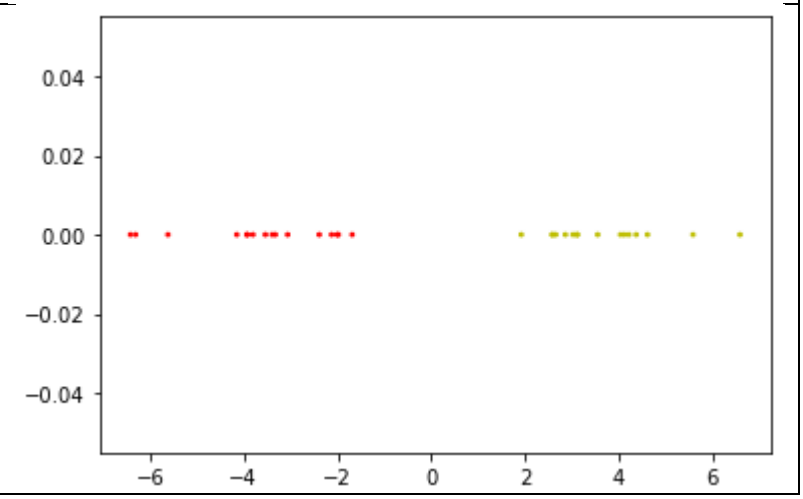
	precision	recall	f1-score	support
-1	1.00	1.00	1.00	64
1	1.00	1.00	1.00	64
accuracy			1.00	128
macro avg	1.00	1.00	1.00	128
weighted avg	1.00	1.00	1.00	128

و بعد از تبدیل داد های تست با بردار راستای LDA و دسته بندی نتایج با perc1 روی داده های تست دارم :

	precision	recall	f1-score	support
-1	1.00	1.00	1.00	16
1	1.00	1.00	1.00	16
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

دقت دسته بند دو کلاسه روی هر دو دادگان تست و آموزش برابر ۱۰۰ درصد هست و خطای مدل روی این داده ها صفر است.

اگر داده های کلاس بندی شده ی یک بعدی را روی شکل نشان دهم ، هیچ داده ای که خطا کلاس بندی شده باشد (رنگ سیاه) ندارم. دو کلاس فاصله ی زیاد از هم دارند.

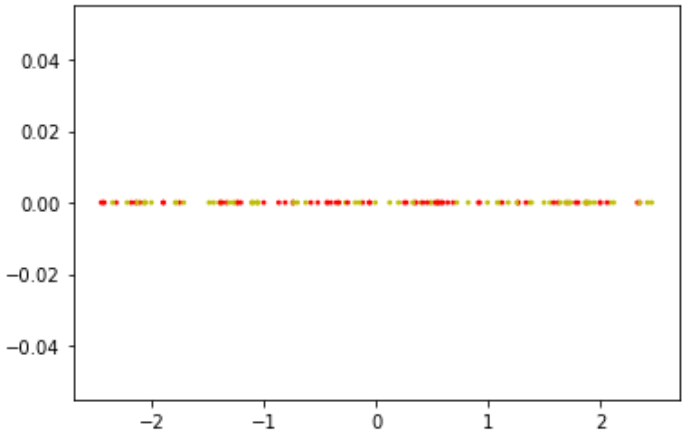
Train data	
Test data	

با استفاده از مدل PCA از کتابخانه ی sklearn داده های آموزش را با معیار PCA به یک فضای تک بعدی منتقل می‌کنم.

ماتریس کوواریانس این تبدیل :

```
[[1.0078739  1.0062541],
 [1.0062541  1.0078735]]
```

این داده های به دست آممده را روی محور بر اساس رنگ کلاس هستند :



این تبدیل داده های کلاس را از هم جدا نمی کند بلکه بیشتر از قبل آن ها را توی هم ادغام می‌کند. بازهم با یک مدل پرسپترون داده های آموزش را مدل می‌کنم. مدل perc2 به دست آمده :

```
[-0.5916291          0.]
```

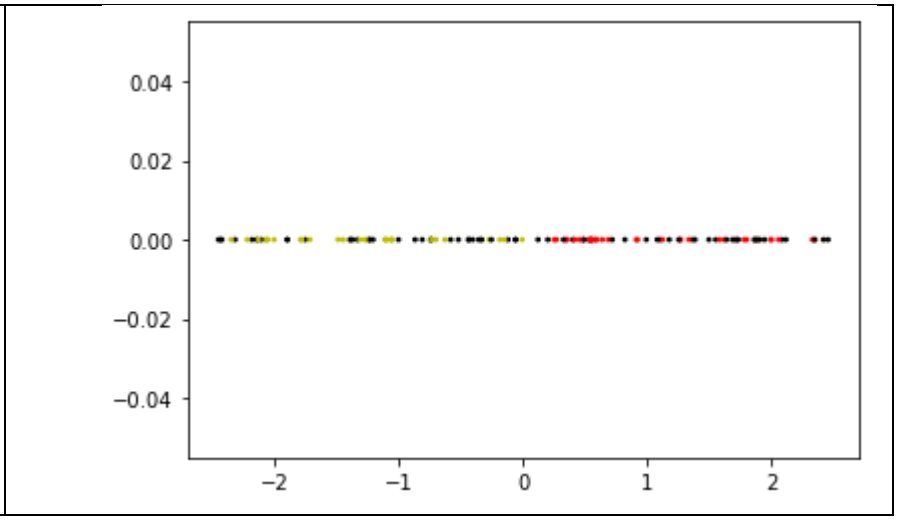
داده های آموزش تک بعدی شده با مولفه ی PCA را با این مدل کلاس بندی میکنم. نتایج زیر به دست می‌آید.

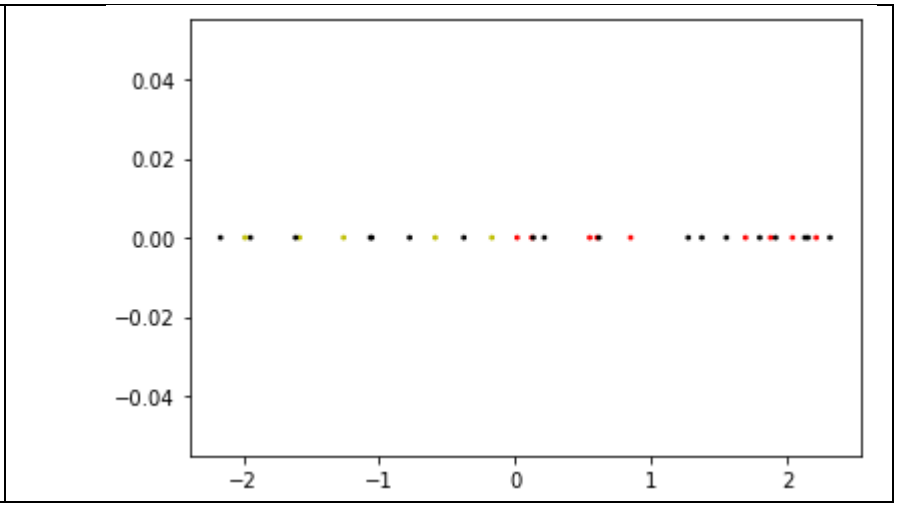
	precision	recall	f1-score	support
-1	0.50	0.50	0.50	64
1	0.50	0.50	0.50	64
accuracy			0.50	128
macro avg	0.50	0.50	0.50	128
weighted avg	0.50	0.50	0.50	128

دقت مدل روی مجموعه ی آموزش پایین است و تنها نیمی از داده ها در کلاس درست طبقه بندی می‌شوند. همچنین اگر معیار PCA را روی داده های تست اعمال و سپس با همین مدل دسته بندی کنم :

	precision	recall	f1-score	support
-1	0.45	0.56	0.50	16
1	0.42	0.31	0.36	16
accuracy			0.44	32
macro avg	0.43	0.44	0.43	32
weighted avg	0.43	0.44	0.43	32

دقت مدل از همان ۵۰ درصد هم پایین تر می‌آید. همچنین روی شکل :

Train data	

Test data	

اگر اولین نمایش داده ها در دو بعد را دوباره نگاه کنیم، روی همان نمودار جداسازی بهتری بین داده های می‌شد متصور بود تا روی داده های انتقال یافته با PCA. این مدل فاقد کارایی است.

تبدیل PCA برای انتقال داده ها و کاهش بعد با هدف بیشینه سازی واریانس داده های باقیمانده پیش می‌رود و تلاش می‌کند داده ها را در جهتی نگاشت دهد که حداکثر واریانس در آن وجود دارد و از طرفی ابعادی که تقریبا نمودار یکنواخت و مسطحی دارند را حذف کند. PCA به برچسب داده ها کاملا بی توجه است (روش بدون نظارت) در حالی که معیار LDA به طور خاص روی داده های برچسب دار با هدف تشدید تمایز بین دو کلای در بعد پایین تر انتقال را انجام می‌دهد. از روی همین می توان انتظار داشت که اگر هدف دسته بندی داده ها باشد، معیار LDA نتیجه ی مطلوب تری خواهد داشت. این نتیجه روی مدل هایی که بالا ساخته شد به خوبی مشخص شد. اگر به اولین نمایش داده ها (دو بعدی) برگردیم، این دو کلاس روی همان دو بعد از هم تفکیک پذیرند اما فاصله ی دو کلاس از خط جداکننده ی فرضی کوچک است. کاری که LDA انجام داد تشدید همین فاصله بود طوری که روی محور اعدادی که LDA ساخت، فاصله ی آخرین داده های دو کلاس حدود ۲ است در حالی که تمرکز داده های هر کلاس را هم بالاتر برد که این تمرکز روی نمایش دوبعدی تقریبا اصلا دیده نمی‌شد. معیار LDA واریانس داده ها را به طور انتخابی و با توجه به برچسب کلاسشان کم یا زیاد می کند در حالی که PCA در جهت حفظ واریانس کل داده ها تلاش می‌کند که روی این داده های نزدیک به هم (ولی جدایی پذیر) باعث ادغام کامل دسته ها شد چون بیشترین واریانس داده های اولیه نه در جهتی که آن دو کلاس از هم متمایز می‌شد (خط فرضی بین دو کلاس) که در همان جهتی تقریبا عمود بر آن بود. یعنی واریانس داده ها در جهت اشتباهی برای هدف جداسازی حفظ شد چون PCA بردار مولفه اول را انتخاب می‌کند در حالی که آنچه در این دسته بندی نیاز است، حفظ واریانس در جهت بردار مولفه ی دوم است و به همین دلیل است که در PCA کلاس ها جداناپذیر شد.

سوال ۳:

فایل را با scipy خواندم. برای اینکه راخت تر با داد ها کار کنم بخش داده هایی که تیاز دارم یعنی faces را که مجموعا ۴۰۰ عکس و هرعکس دارای ۶۴*۶۴ مولفه بود را به صورت یک ماتریس داده ی numpy از ۴۰۰ ردیف و ۶۴*۶۴ ستون ذخیره کردم. این عکس ها به ترتیب ۱۰ تا ۱۰ عکس های ۴۰ نفر اند پی یک بردار ۴۰۰ تایی هم برای برچسب ها درست کردم که البته چون کار دسته بندی و شناسایی چهره خواسته نشده بود به کار نیامد! این ۴۰۰ ردیف را بدون shuffle کردن با train_test_split از sklearn به دو دسته ۲۸۰ تایی و ۱۲۰ به ترتیب به عنوان آموزش و تست تقسیم کردم.

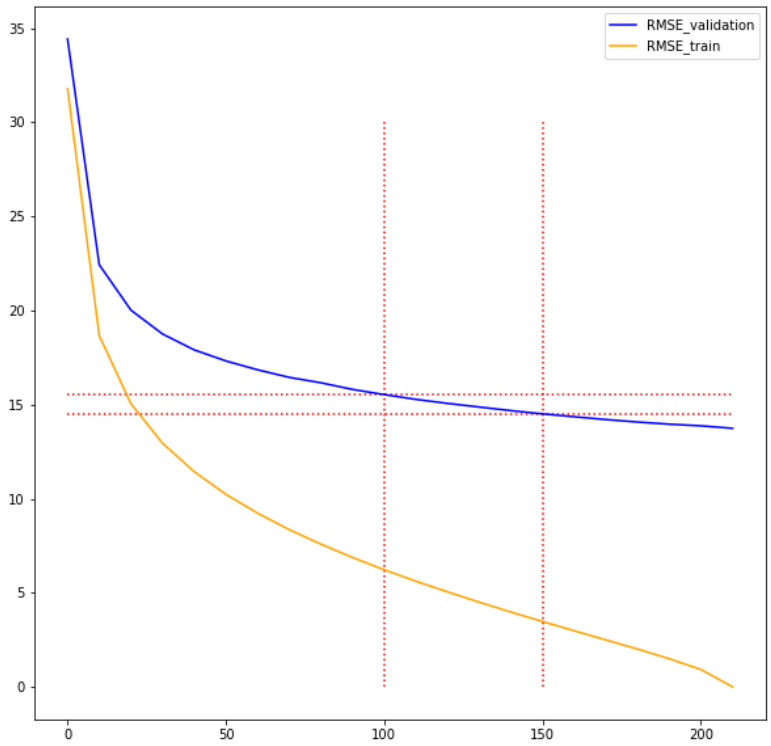
با یک مدل PCA از sklearn ۵ این داده ها را از فضای ۶۴*۶۴ به ۴۵ بعد می‌برم. مدل فقط با ۲۸۰ داده ی اول آموزش دیده. سپس داده های آموزش و تست را با مولفه های به دست آممده به فضای ۴۵ بعد می‌برم و بعد دوباره به همان فضای اولیه بازسازی می‌کنم. مقدار RMSEبه دست آمده بین داده های اولیه ی هر مجموعه با فضای بازسازی شده برابر است با :

```
RMSE on train data = 11.525403972573551
RMSE on test data = 18.023206734576966
```

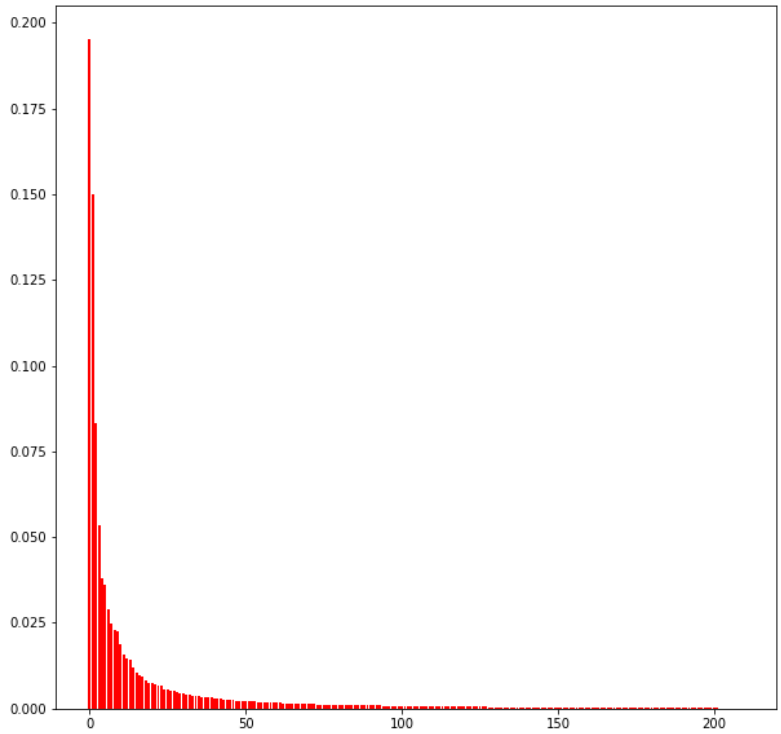
در زیر یک تصویر دلخواه از مجموعه ی تست را (تصویر نظیر ردیف 1 تست) در دو حالت اولیه و بازسازی شده با ۴۵ مولفه ی اصلی داریم.

original	projected
	

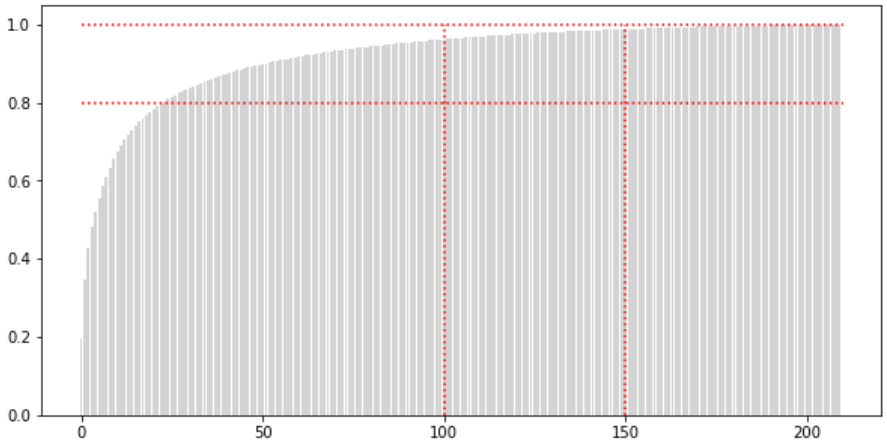
برای انتخاب بعد بهینه مجموعه ی آموزش را به اندازه ی ۷۵ به ۲۵ تقسیم میکنم. و بخش آخر را برای validation نگه می‌دارم. برای فضای ثانویه از کل مجموعه ی ۲۱۰ تایی تست یک مجموعه ی بزرگ از ۱ تا ۲۱۰ می توانم به عنوان عدد بعد انتخاب کنم (اگر داده ها بیشتر بود، می توانستم تا ۶۴*۶۴ را به عنوان بعد ثانویه بگیرم ولی اینجا محدودیت تعداد برای ماتریس داده ها دارم) با شروع از صفر و ۱۰ تا ۱۰ تا جلو رفتن هر بار یک تبدیل PCA روی مجموعه ی تست می‌سازم. به ازای هر کدام از این تبدیل های ساخته شده روی مجموعه های ۲۱۰ داده ی آموزش و ۷۰ داده ی اعتبارسنجی مقدار تبدیل را محاسبه و سپس قضای اولیه را با همان بردارها بازسازی می‌کنم. RMSE را محاسبه می‌کنم. نمودار این اعداد به دست آمده :



اگر روی بخش مشخص شده ی نمودار با خط چین های قرمز دقت کنم اولا از ۱۰۰ به بعد شیب خط آبی اعتبارسنجی ملایم تر می‌شود و بهبود بیشتر روی داده های تست است. پس عدد بعد را می‌توان بعد از ۱۰۰ انتخاب کرد. از طرفی بعد از ۱۵۰ هم شیب زیادی ملایم می‌شود. و ۵۰ بعد باقیمانده بیشتر فقط حدود ۱ واحد روی RMSE تاثیر دارد. از این مقدار می‌شود چشم پوشید.
پس عددم را جایی بین ۱۰۰ تا ۱۵۰ انتخاب می‌کنم.
اگر روی نمودار میله ای نسبت واریانس های توزیع یافته به ازای هر بردار مولفه ی ۱ تا ۲۱۰ دقت کنیم کاهش نسبت از ۱۰۰ به بعد به سختی قابل مشاهده است.



برای دید بهتر نمودار همین نسبت ها را به صورت تجمعی رسم می‌کنم. از بردار ۱۵۰ به بعد تاثیر بسیار ناچیز است. و آهرین بردارهایی که تاثیر ثابل توجه دارند همان ۱۰۰ تا ۱۵۰ هستند. روی همان بازه ای که در نظر داشتم نقطه ی وسط را به عنوان بعد ثانویه انتخاب می‌کنم. البته خود ۱۰۰ یا ۱۵۰ هم برای انتخاب مناسب است چون به لحاظ دقت بازهم تفاوت چندانی روی این بازه برای این تعداد داده نداریم. من ۱۲۵ را انتخاب می‌کنم.



انتظار دارم نتیجه ی تبدیل PCA نهایی که با بعد ۱۲۵ می‌سازم روی داده های تست مقداری حدود ۱۵ از روی نمودار اول مثل داده های اعتبارسنجی داشته باشد. آخرین مدل را با ۱۲۵ مولفه ی اصلی می‌سازم. نتایج :

```
RMSE on train data = 4.775306976555008
RMSE on validation data = 14.968495895510301
RMSE on test data = 16.10800194377165
```

دقت بازسازی روی تست هم همان نزدیک مقدار انتظار است. خطا ی بین داده های اموزش اولیه و بازسازی شده روی داده های آموزش خیلی کم است.

سوال ۴ :

$X = \begin{bmatrix} & \\ & \end{bmatrix}_{n \times d}$

فرض : فضای d بعدی ، n ردیف داده

هدف : نگاشت داده‌ها بر فضای m بعدی با هدف بیشینه‌سازی واریانس که $m < d$

اگر فرض کنیم : $m = 1$ و بردار V_1 یک بردار $d \times 1$ باشد جهت فضای تازه را بگیریم :

$$V_1^T V_1 = 1$$

$x'^{(i)} = V_1^T x^{(i)} \rightarrow \mathbb{R}^{m=1} \leftarrow \mathbb{R}^d$

واگر μ' برابر میانگین داده‌ها روی فضای d بعدی اول و μ' برابر میانگین داده‌ها روی فضای $m = 1$ بعدی دوم باشد :

$$\mu' = \frac{1}{n} \sum_{i=1}^{n} x^{(i)'} = \frac{1}{n} \sum V_1^T x^{(i)} = V_1^T \left(\frac{1}{n} \sum_{i=1}^{n} x^{(i)} \right)$$

$$= V_1^T \mu$$

همچنین برای واریانس داده‌های فضای تازه داریم :

$$Var(X') = \frac{1}{n} \sum_{i=1}^{n} (x^{(i)'} - \mu')^2 = \frac{1}{n} \sum_{i=1}^{n} (V_1^T x^{(i)} - V_1^T \mu)^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} (V_1^T x^{(i)} - V_1^T \mu)(V_1^T x^{(i)} - V_1^T \mu)^T$$

$$= \frac{1}{n} \sum_{i=1}^{n} V_1^T (x^{(i)} - \mu)(x^{(i)T} V_1 - \mu^T V_1)$$

$$= \frac{1}{n} \sum_{i=1}^{n} V_1^T (x^{(i)} - \mu)(x^{(i)} - \mu)^T V_1$$

$$= V_1^T \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - \mu)(x^{(i)} - \mu)^T \right)}_{\text{ماتریس کوواریانس فضای } d \text{ بعدی}} V_1 = V_1^T S V_1 = J(V_1)$$

حالا طبق هدف ، می‌خواهیم واریانس X' را بیشینه کنیم :

$$\max \quad J(V_1) = V_1^T S V_1$$

$$\text{s.t.} \quad \|V_1\| = 1$$

قید یکه بودن به این دلیل اضافه شد که هدف بهینه‌سازی جهت بزرگ کردن $J(V_1)$ از راه بزرگ کردن اندازه‌ی V_1 نبود . می‌توان این قید را

ترکیب ضریب در رابطه J تبدیل کرد:

$$\max \quad J(V_1) = V_1^T S V_1 + \lambda_1 (1 - V_1^T V_1)$$

$$\text{حل} \Rightarrow \quad \frac{\partial J}{\partial V_1} = 2SV_1 - 2\lambda_1 V_1 = 0$$

دنبال V_1 ای می‌گردیم که در این رابطه $SV_1 = \lambda_1 V_1$ صدق کند. این همان رابطه‌ی بردار ویژه‌ی ماتریس S است و λ_1 مقدار ویژه‌ی متناظر آن

$$\xrightarrow{\times V_1^T} \quad SV_1 = \lambda_1 V_1 \qquad \text{گرفتیم } \|V_1\| = 1$$

$$V_1^T S V_1 = V^T(\lambda_1 V_1) = \lambda_1 V_1^T V_1$$

$$V_1^T S V_1 = \lambda_1 \xrightarrow{Var(X') = V_1^T S V_1} Var(X') = \lambda_1$$

پس به ازای V_1 که $\frac{\partial J}{\partial V_1} = 0$، مقدار $J(V_1) = Var(X')$ برابر می‌شود با مقدار ویژه‌ای از S که بیشترین مقداری که می‌تواند بگیرد برابر می‌شود با $\max\{\lambda_1, \lambda_2, \dots\}$ که در آن λ_i ها مقادیر ویژه‌ی S هستند و برای V_2 و V_3 و ... هم می‌توان به ترتیب مثل که V_1 را انتخاب کردیم بزرگترینشان را انتخاب کرد (معیار انتخاب PCA)