Mohammad Javad Ranjbar

810101173

HW5

Machine Learning, Fall 2022

Contents

سوال اول:

۱. خیر. دو کار جدا هستند:

model Selection به معنی مقایسه عملکرد مدل‌ها مختلف روی داده‌های نمونه‌ی ما هست که از بین این مدل‌ها، مدل با بهترین عملکرد را انتخاب می‌کنیم.

model assessment به معنی ارزیابی مدل انتخاب شده است. به عبارتی دیگر ارزیابی مدل بر پایه معیارهای عملکرد مانند مقدار خطا، دقت و تعمیم‌پذیری مدل می‌باشد.

۲. خطای تعمیم‌پذیری با اعمال مدل بر روی داده‌ای که قبلا توسط مدل دیده نشده به دست می‌آید. حال برای اینکه مدلی را انتخاب کنیم که خطای تعمیم پذیری کمتری داشته باشد. ما باید هم از overfitting و هم از underfitting جلوگیری کنیم. چندین روش برای تعمیم‌پذیر تر کردن مدل وجود دارد:

- استفاده از ترم‌هایی مانند regularization تا از overfitting جلوگیری شود. همچنین از روش‌های Weight Decay نیز می‌توان استفاده کرد.
- استفاده کردن از داده‌های آموزش بیشتر و متونع تر در هنگام آموزش. که با دیدن داده آموزش بیشتر مدل بهتر آموزش می‌بیند و تعمیم‌پذیر تر خواهد بود.
- استفاده از داده validation برای تنظیم کردن پارامترهای مدل. همچنین استفاده از cross validation نیز برای پیدا کردن بهترین مدل کاربردی است.
- متوقف کردن آموزش در لحظه‌ی درست و قبل از overfit شدن.

۳. در صورتی که تعداد داده‌ی کمی داریم بهتر است از مدل‌های هرچه ساده تر استفاده کنیم تا از Overfit شدن جلوگیری کنیم. و البته باید به ویژگی‌هایی که برای آموزش استفاده می‌کنیم دقت کنیم و outlier ها را نیز در نظر بگیریم. همچنین می‌توان از روش‌های ساخت داده‌ی مصنوعی برای آموزش مدل استفاده کرد. علاوه بر این‌ها روش‌هایی همچون ensemble models یعنی تجمیع چند مدل برای کاربرد خود نیز قابل استفاده است.

۴. تعدادی از متریک‌های مدل در جدول زیر مشخص شده است برای مثال متریک Precision را توضیح می‌دهیم:

وقتی مدل به عنوان مثبت دسته می‌کند، چند وقت یکبار درست است؟

$$\text{Precision} = \frac{True\ positive}{All\ the\ positives}$$

هنگامی که ما یک عدم تعادل کلاس داریم، accuracy می تواند به یک معیار غیر قابل اعتماد برای اندازه گیری عملکرد ما تبدیل شود. به عنوان مثال، اگر ما یک تقسیم ۹۹ و ۱ بین دو کلاس A و B داشته باشیم، جایی که رویداد نادر، B، کلاس مثبت ما است، می‌توانیم مدلی بسازیم که ۹۹٪ دقیق باشد فقط با گفتن اینکه همه چیز متعلق به کلاس A است. واضح است که اگر مدلی برای شناسایی کلاس B کاری انجام نمی دهد، نباید به زحمت بسازیم. بنابراین، ما به معیارهای مختلفی نیاز داریم که از این رفتار جلوگیری کند. برای این کار از دقت و فراخوانی به جای دقت استفاده می کنیم.

Metric	Formula
True positive rate (Recall)	$\frac{True\ positive}{True\ positive + \ False\ negative}$
False positive rate	$\frac{False\ positive}{False\ positive + \ True\ negative}$
Precision	$\frac{True\ positive}{True\ positive + \ False\ positive}$
Accuracy	$\frac{True\ positive + \ True\ negative}{True\ positive + \ False\ negative + False\ positive + \ True\ negative}$
F-measure	$\frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

سوال ۲:

۱. Feature selection یا انتخاب ویژگی به معنی جدا کردن زیر مجموعه‌ای از ویژگی‌ها است که مزایای زیر را دارا می‌باشد.

- کاهش overfitting: داده‌های اضافی کمتر یعنی تاثیر داده‌های بی ارزش و نویز بر تصمیم‌گیری کمتر خواهد شد.
- دقت را بهبود می بخشد: داده‌های اضافی و نادرست ممکن است باعث کاهش دقت مدل شود.
- زمان آموزش را کاهش می‌دهد: داده‌های کمتر به این معنی است که الگوریتم‌ها سریع‌تر آموزش می‌بینند.

البته اگر به تعداد بار خیلی زیاد مدل را آموزش دهیم، خود مدل ممکن است بتواند اثر ویژگی‌های نامربوط و اشتباه را حذف کند. اما این کار، باعث زمان بر شدن آموزش و همچنین احتمالا کمتر شدن دقت نهایی خواهد شد.

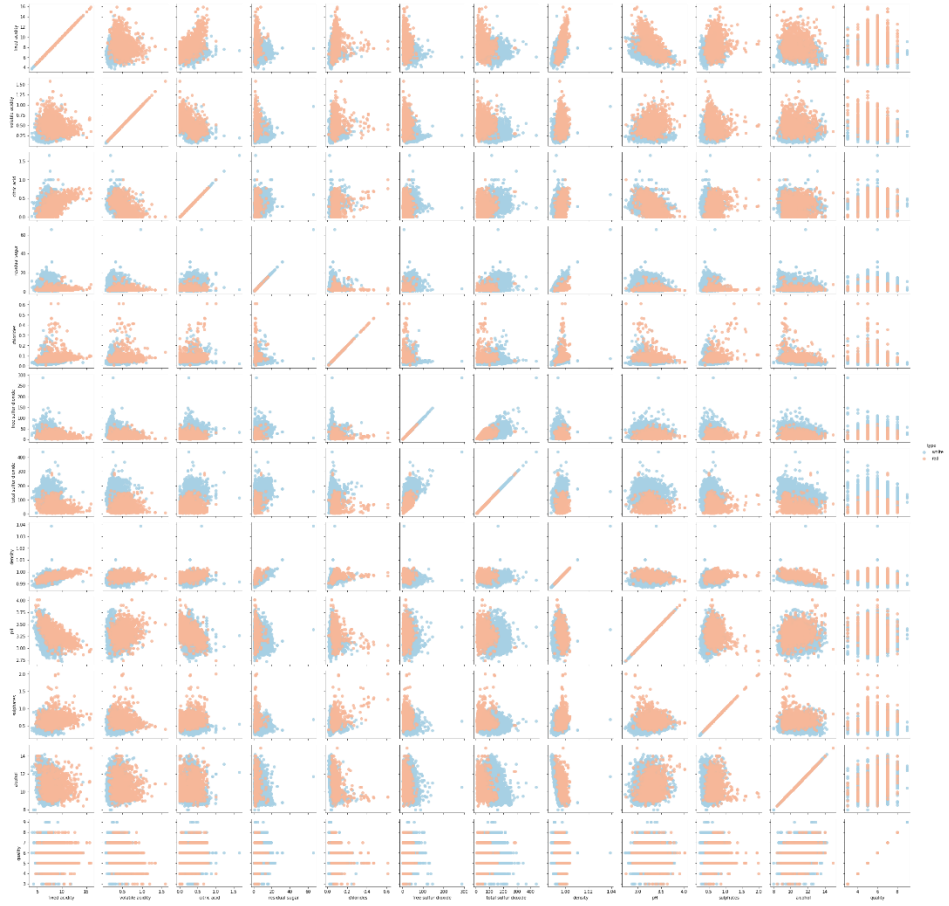
۲. روش fisher's score یک روش انتخاب ویژگی نظارت شده است که برای هر ویژگی fisher score که یکی نسبت از واراینس بین کلاسی به واریانس درون کلاسی است را محاسبه می‌کند. به عبارت دیگر می‌توان گفت ایده کلیدی امتیاز فیشر یافتن زیرمجموعه ای از ویژگی ها است، به طوری که در فضای داده ای که ویژگی های انتخاب شده را در بر می گیرد، فاصله بین نقاط داده در کلاس های مختلف تا حد امکان بزرگ باشد، در حالی که فاصله بین نقاط داده در همان کلاس وجود دارد. تا حد امکان کوچک هستند. این الگوریتم ویژگی با بزرگترین امتیاز را انتخاب می‌کند. این امتیاز به صورت زیر محاسبه می‌شود:

$$S_i = \frac{\Sigma n_j (\mu_{ij} - \mu_i)^2}{\Sigma n_j p_{ij}}$$

که μ_{ij} و p_{ij} به ترتیب میانگین و واریانس ویژگی iام در کلاس jام می‌باشد. n_j تعداد نمونه‌ها در کلاس jام می‌باشد. μ_i نیز میانگین ویژگی iام می‌باشد.

سوال ۳:

نمودار توزیع دو به دو ویژگی‌ها برای این دیتاست به شکل زیر می‌باشد:



با توجه به نمودار بالا به نظر می‌آید که نقاط به صورت خطی از هم جدایی‌پذیر نیستند بنابراین از مدل غیر خطی SVM با کرنل RBF برای این کار استفاده می‌کنیم.

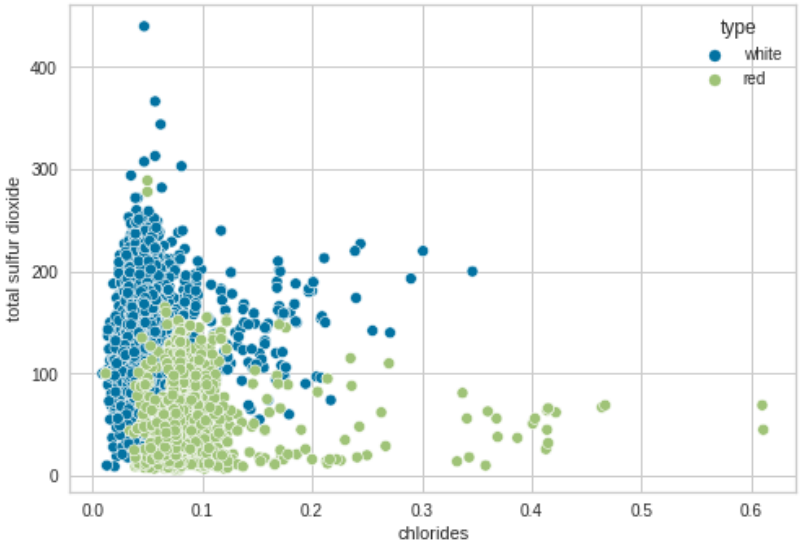
۱. روش Sequential forward selection (SFS) قصد دارد زیرمجموعه‌ای از ویژگی‌ها پیدا کند. این روش به صورت حریصانه در هر مرحله بهترین ویژگی را انتخاب می‌کند. به عبارتی در هر مرحله با استفاده از تخمین زنی (مدل) تک تک همه ویژگی‌ها را امتحان کرده و ویژگی که بهترین عملکرد را می‌دهد انتخاب می‌کنیم و به زیرمجموعه‌ی خود اضافه می‌کنیم و اینکار را تا جای لازم انجام می‌دهیم.

بدون استفاده از کتابخانه

این روش هم بدون استفاده ازSklearn هم با استفاده از sklearn پیاده شده است که نتایج هردو یکسان و به صورت زیر است.

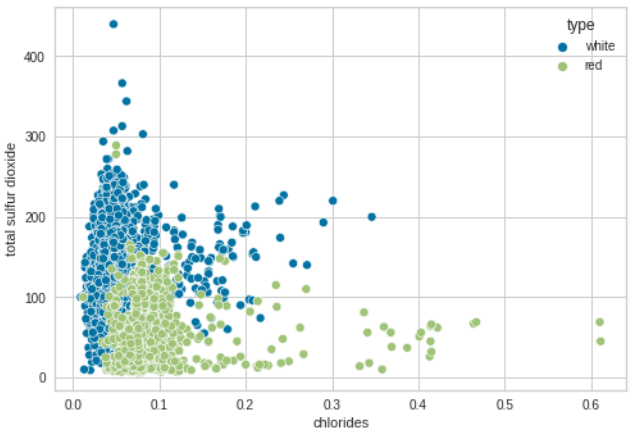
با استفاده از کتابخانه

پس از اعمال عملیات فوق دو ویژگی 'chlorides', 'total sulfur dioxide' به عنوان ویژگی‌های اصلی بازگردانده می‌شوند که نمودار آن‌ها به شکل زیر می‌باشد.



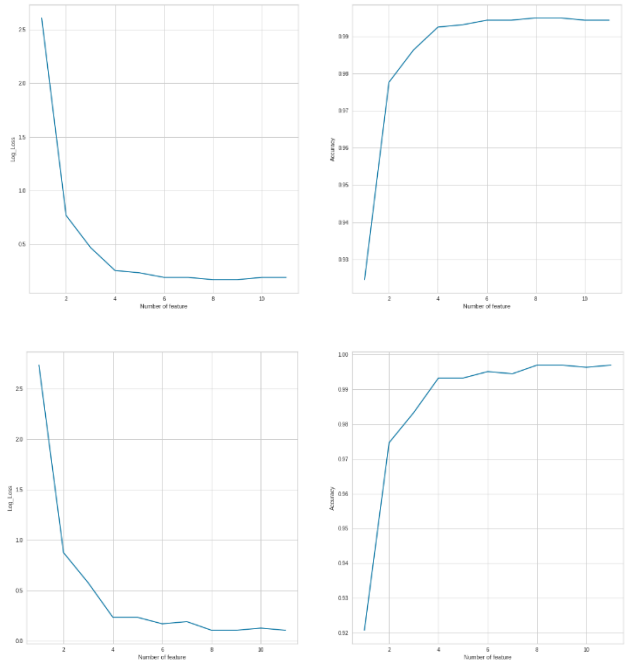
۲. روش Recursive Feature Elimination (RFE) به در ابتدا تخمین زنی (مدلی) بر روی کل ویژگی‌ها آموزش می‌بیند و اهمیت هر ویژگی سنجیده می‌شود، سپس ویژگی با کمترین اهمیت/اثر را حذف می‌کنیم و با زیرمجموعه‌ی جدید این کار را تا جای لازم ادامه می‌دهیم.

پس از اعمال عملیات فوق دو ویژگی 'chlorides', 'total sulfur dioxide' به عنوان ویژگی‌های اصلی بازگردانده می‌شوند که نمودار آن‌ها به شکل زیر می‌باشد.



۳.

الگوریتم RFE سریعتر از الگوریتم SFS است و همچنین از الگوریتم SFS ساده‌تر است اما بسیار وابسته به تعداد ویژگی‌های حذف شده در هر مرحله است که باید توسط ما مشخص شود، اما الگوریتم SFS این وابستگی را ندارد. یکی از مشکلات الگوریتم SFS این است که اگر ویژگی f1 به تنهایی در ابتدا به ما اطلاعات کمی بدهد اما در ترکیب با ویژگی دیگری مانند f2 بهترین ویژگی‌ها باشند. ممکن است ویژگی f1 اما در مرحله اول حذف شود لذا این الگوریتم لزوما زیرمجوعه‌ای بهینه از ویژگی‌ها را به ما نمی‌دهد. نمودار خطا به ترتیب برای SFS و REF به صورت زیر خواهد بود



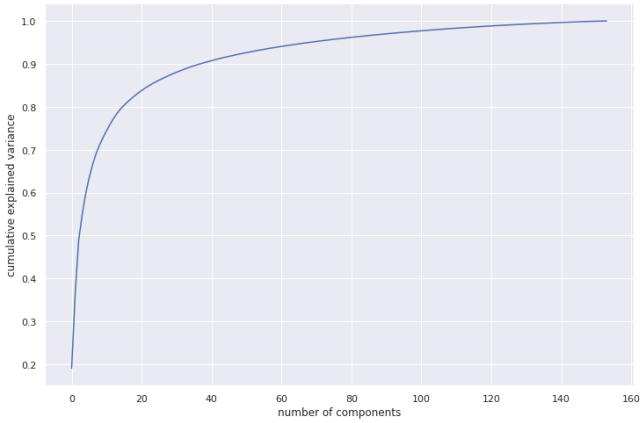
با توجه به نمودار بالا مشخص است که با افزایش ویژگی‌ها دقت و خطا تا جایی با شیب زیاد کاهش می‌یابد، اما پس از اینکه تعداد کافی ویژگی اضافه شد، این شیب به شدت کم شده و حتی به خط صاف نزدیک می‌شود. بنابراین این ویژگی‌ها فقط حجم محاسبات را زیاد کرده و به نظر می‌رسد اطلاعات زیادی به ما نمی‌دهند.

سوال ۴:

PCA با کم کردن بعد داده‌ها باعث می‌شود که ویژگی‌های کمتری برای آموزش داشته باشیم، بنابراین در کار پردازش تصویر که تعداد پیکسل یا ویژگی زیادی دارم با استفاده از PCA، عکس‌ها به ماتریس‌هایی به بعدهای کمتر تصویر می‌شوند که ویژگی‌های زائد یا افزونه حذف می‌شود.

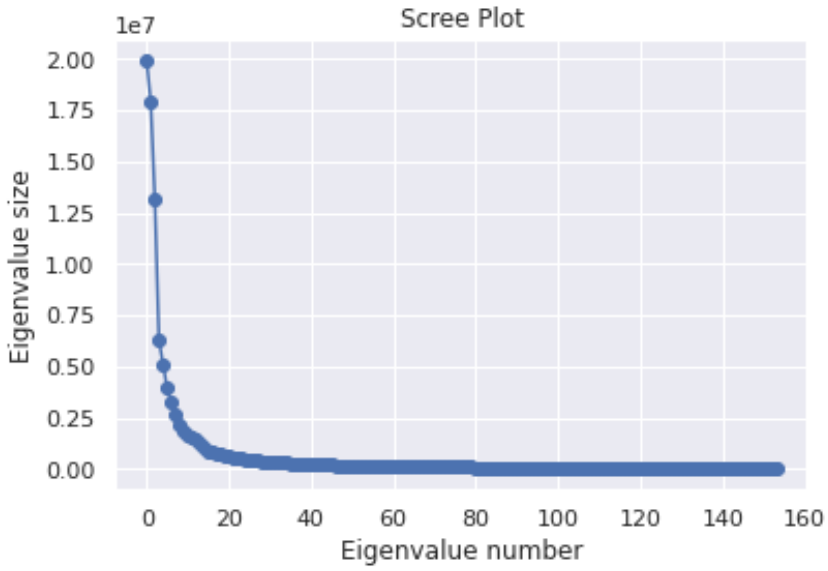
PCA به طور کلی قصد دارد داده‌ها را به فضایی ببرد که واریانس در آن بعد بیشترین حد باشد.

۱. با توجه به نمودار تجمعی زیر می‌توان ایده‌ای از تعداد PCA های مورد نیاز گرفت.



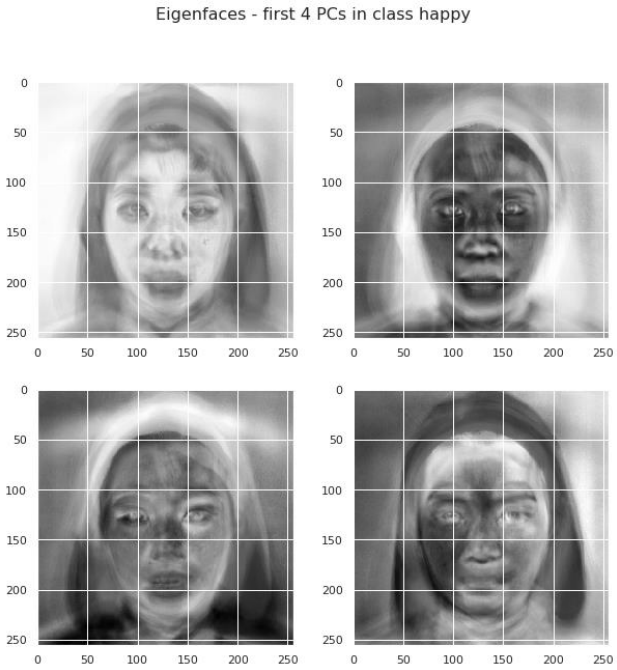
در منحنی نشان داده شده است که به ازای n تا مولفه چند درصد واریانس را شامل می‌شود، برای مثال با ۱۰ مولفه ۷۰ درصد واریانس را داریم. حال مشخص است که برای حفظ ۹۰ درصد واریانس مثلا نیاز به حفظ ۲۵ مولفه داریم.

البته باید توجه داشت که وابسته به کاربرد ما این تعداد فرق می‌کند، برای مثال اگر فقط قصد رسم کردن داده را داریم باید از ۲ یا ۳ مولفه استفاده کنیم. همچنین، روش دیگری نیز وجود دارد که تمام eginvalue ها با مقادیر بزرگ را نگه داریم بنابراین باید نموداری بر حسب آن‌ها بکشیم.

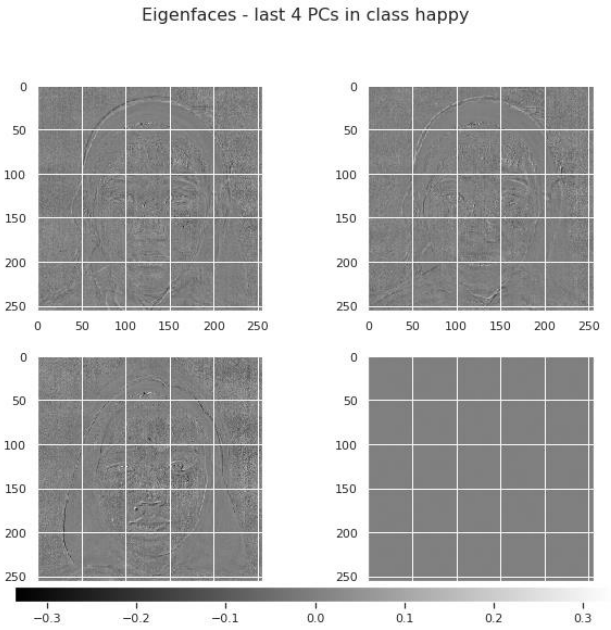


۲. ۴ مقدار ویژه اول برای ویژگی‌های مهمتر و اصلی هستند. در این دیتاست که برای تشخیص احساسات صورت انسان است، این ویژگی‌ها در واقع بخش‌های پر اطلاعات صورت همچون لب، چشم‌ها و ابروها هستند و برای مثال در حالت شاد بودن، چشم‌ها و اطراف دهان مهمترین بخش‌ها هستند. تعدادی از این مقادیر ویژه به صورت زیر هستند:

```
eighen values = [915.2169473962939, 625.1633983950859,
504.6764136095698, 383.604182910301, 369.0867587576559,
348.01660413462304, 290.7657876779423, 251.8774374477482,
226.62934271403503, 217.7734779626308, 196.6070888012572,
189.97287906448324, 153.74531230868774, 137.1301348257419,
131.63045948956864, 129.80638429895203]
```



اما ۴ مقدار ویژه آخر برا اطلاعات کم اهمیت‌تر هستند. برای این دیتاست، بخش‌های خارج صورت یا لبه‌های عکس این اطلاعات را نشان می‌دهد.



۳. نتایج برای داده آموزش و تست بدون استفاده از PCA برای k=1 و k=2 به صورت زیر می‌باشد:

Train data with K=1:

	precision	recall	f1-score	support
angry	1.00	1.00	1.00	30
disgust	1.00	1.00	1.00	18
fear	1.00	1.00	1.00	20
happy	1.00	1.00	1.00	45
sad	1.00	1.00	1.00	21
surprise	1.00	1.00	1.00	20
accuracy			1.00	154
macro avg	1.00	1.00	1.00	154
weighted avg	1.00	1.00	1.00	154

دقت برابر با ۱

****

Test data with K=1:

	precision	recall	f1-score	support
disgust	1.00	0.92	0.96	12
fear	0.83	0.91	0.87	11
happy	0.94	0.83	0.88	18
sad	0.70	0.70	0.70	10
surprise	0.80	1.00	0.89	8
accuracy			0.86	59
macro avg	0.85	0.87	0.86	59
weighted avg	0.87	0.86	0.87	59

دقت برابر با ۸۶ درصد

Train data with K=2:

	precision	recall	f1-score	support
angry	1.00	0.79	0.88	38
disgust	0.72	0.87	0.79	15
fear	0.95	0.95	0.95	20
happy	0.93	0.86	0.89	49
sad	0.71	0.94	0.81	16
surprise	0.80	1.00	0.89	16
accuracy			0.88	154
macro avg	0.85	0.90	0.87	154
weighted avg	0.89	0.88	0.88	154

دقت برابر با ۸۸ درصد

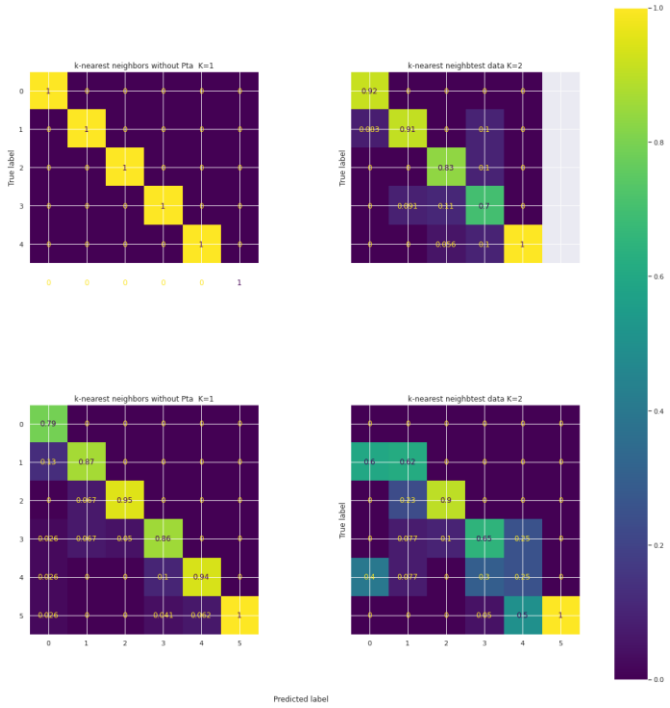
****

Test data with K=2:

	precision	recall	f1-score	support
angry	0.00	0.00	0.00	5
disgust	0.73	0.62	0.67	13
fear	0.75	0.90	0.82	10
happy	0.81	0.65	0.72	20
sad	0.10	0.25	0.14	4
surprise	0.70	1.00	0.82	7
accuracy			0.64	59
macro avg	0.51	0.57	0.53	59
weighted avg	0.65	0.64	0.64	59

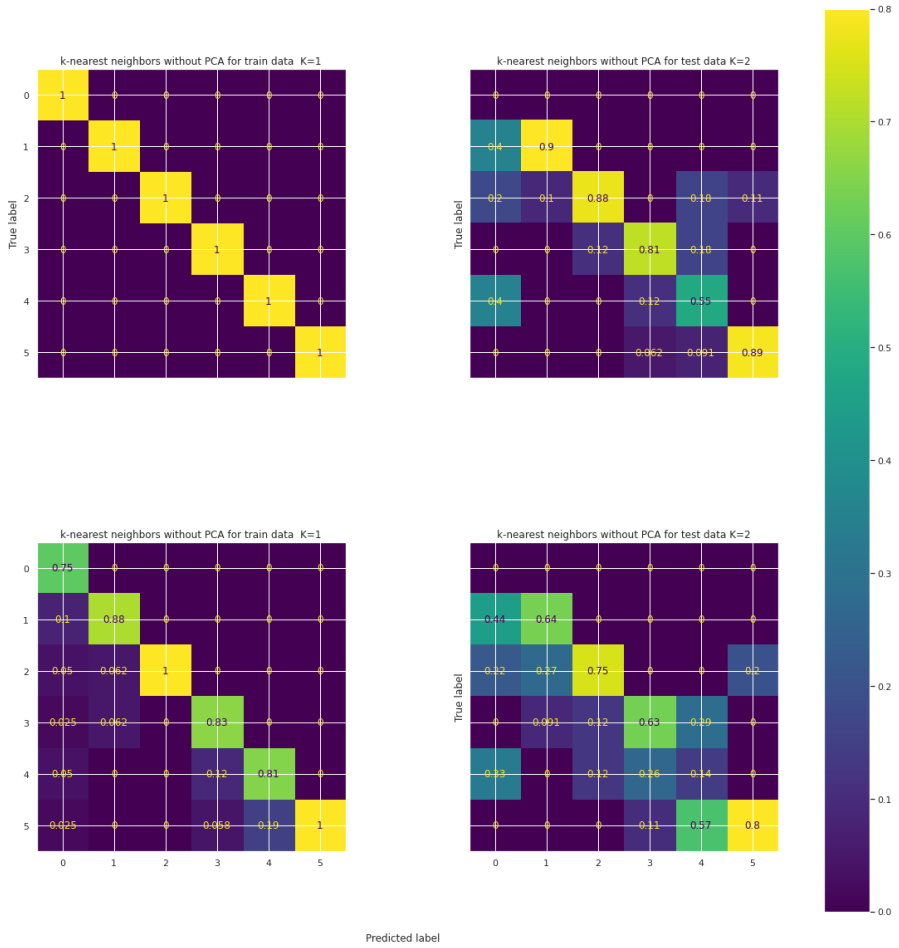
دقت برابر با ٦٤ درصد

همچنین ماتریس درهمریختگی به شکل زیر می‌باشد:



قبل از اعمال PCA داده‌ها را حتما نورمالایز می‌کنیم.

حال نتایج برای داده تست و آموزش با استفاده از PCA به شکل زیر خواهد بود، برای این بخش تعداد ویژگی‌ها را ۱۵ انتخاب کرده‌ایم:



و نتایج به صورت زیر خواهد بود:

```
Train data with K=1:
              precision    recall  f1-score   support

       angry       1.00      1.00      1.00        30
     disgust       1.00      1.00      1.00        18
        fear       1.00      1.00      1.00        20
       happy       1.00      1.00      1.00        45
         sad       1.00      1.00      1.00        21
    surprise       1.00      1.00      1.00        20

    accuracy                           1.00       154
   macro avg       1.00      1.00      1.00       154
weighted avg       1.00      1.00      1.00       154
```

دقت ۱۰۰ درصد

```
************************************************************
Test data with K=1:
```

	precision	recall	f1-score	support
angry	0.00	0.00	0.00	5
disgust	0.82	0.90	0.86	10
fear	0.58	0.88	0.70	8
happy	0.81	0.81	0.81	16
sad	0.60	0.55	0.57	11
surprise	0.80	0.89	0.84	9
accuracy			0.73	59
macro avg	0.60	0.67	0.63	59
weighted avg	0.67	0.73	0.70	59

دقت ۷۳ درصد

Train data with K=2:

	precision	recall	f1-score	support
angry	1.00	0.75	0.86	40
disgust	0.78	0.88	0.82	16
fear	0.85	1.00	0.92	17
happy	0.96	0.83	0.89	52
sad	0.62	0.81	0.70	16
surprise	0.65	1.00	0.79	13
accuracy			0.84	154
macro avg	0.81	0.88	0.83	154
weighted avg	0.88	0.84	0.85	154

دقت ۸۴ درصد

****

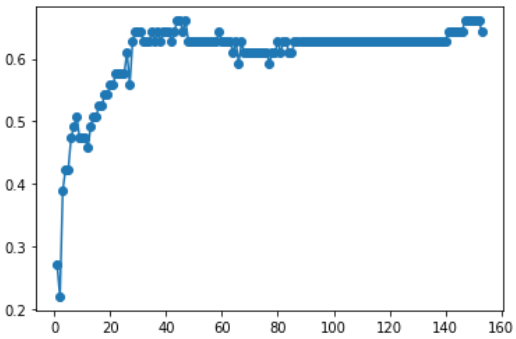
Test data with K=2:

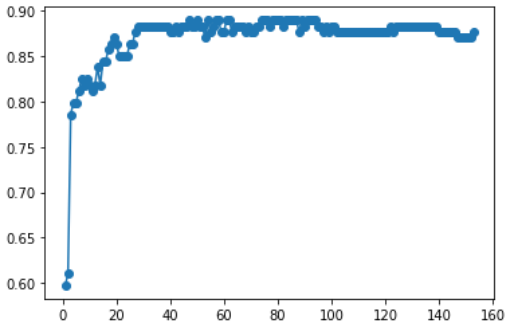
	precision	recall	f1-score	support
angry	0.00	0.00	0.00	9
disgust	0.64	0.64	0.64	11
fear	0.50	0.75	0.60	8
happy	0.75	0.63	0.69	19
sad	0.10	0.14	0.12	7
surprise	0.40	0.80	0.53	5
accuracy			0.51	59
macro avg	0.40	0.49	0.43	59
weighted avg	0.47	0.51	0.48	59

دقت برابر با ۵۱ درصد

با مقدار کمتر از ویژگی‌ها یعنی با استفاده از PCA طبقه‌بند روی داده‌های تست بهتر عمل می‌کند.

۴. مشخص است که با افزایش تعداد PCAها دقت طبقه‌بند ما بهتر می‌شود اما از یک مقداری بزرگتر ویژگی‌های جدید اطلاعات زیادی به طبقه‌بند ما اضافه نمی‌کند و فقط ویژگی‌های زائد و اضافی هستند.





سوال ۵:

۱. ماتریس پراکندگی بین گروهی و درون گروهی به شرح زیر می‌باشد:

ماتریس پراکندگی یک ماتریس برای تخمین کوواریناس می‌باشد. n نمونه‌ی داده با m بعد که با یک ماتریس m در n به صورت $X = [x_1, x_2, x_3, \cdots, x_n]$ نمایش داده می‌شود. میانگین نمونه‌ها به صورت زیر خواهد بود:

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$$

که x_j ستون jام X می‌باشد.

- پراکندگی درون گروهی (S_w):

می‌خواهیم به صورت زیر محاسبه می‌شود:

$$S_w = \sum_{\text{Classes}} \sum_{j \in \text{classes}} (x_j - \bar{x}_c)(x_j - \bar{x}_c)^T$$

بنابراین، ماتریس پراکندگی در هر کلاس را محاسبه می کنیم تا پراکندگی درون هر کلاس را بدست آوریم. که برای هر نمونه x_j یک ماتریس $m * m$ به ما می دهد. سپس همه این ماتریس‌ها را جمع می کنیم تا پراکندگی در هر کلاس را بدست آوریم) و سپس این ماتریس‌های پراکندگی را جمع می کنیم تا اندازه‌ای از پراکندگی در کل مجموعه داده S_w دریافت کنیم.

- پراکندگی بین گروهی (S_B):

$$S_B = \sum_{\text{Classes}} (\bar{x}_c - \bar{x})(x_j - \bar{x})^T$$

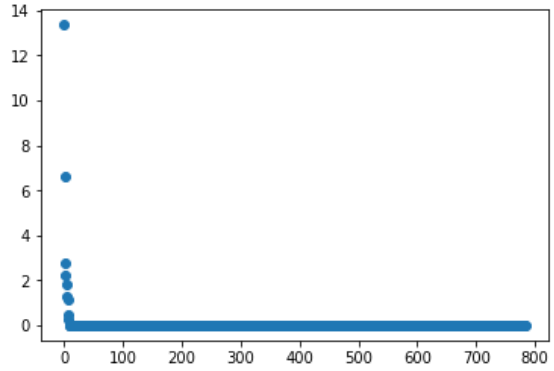
با این فرمول دوم ، ما پراکندگی کل مجموعه داده را اندازه گیری می کنیم، یعنی پراکندگی بین کلاس ها و اینکه کلاس‌های منفرد چقدر دور هستند. اینجا c کلاس های مختلف مجموعه داده ما هستند. $\bar{x}_c$ میانگین هر کلاس است. $\bar{x}$میانگین کل مجموعه داده است.

۲. ماتریس مورد نظر به صورت زیر خواهد بود:

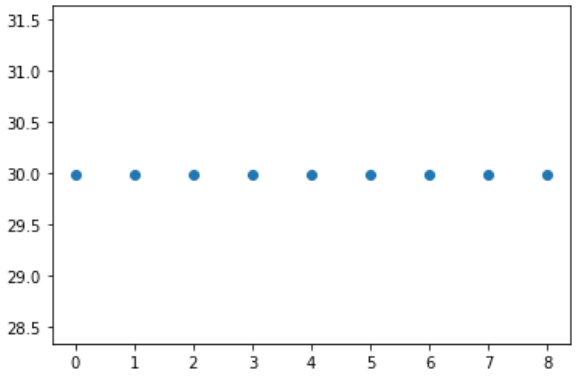
بدون استفاده از کتابخانه:

```
array([[ 2.70078337e-04, -1.63182463e-06, 2.55209184e-04, ..., -
9.75197030e-04, -1.18629642e-03, -3.32073321e-04], [-3.07009171e-04,
1.49940293e-04, -6.92734014e-04, ..., -1.14709588e-04, 2.30790474e-
03, 1.05695674e-03], [ 5.55387864e-04, 4.70118216e-04, 3.19531345e-
03, ..., -1.35692082e-03, -4.62278675e-03, -2.20088480e-03], ..., [
3.88050710e-04, -2.73968375e-04, -2.43984773e-05, ..., 1.85363790e-
03, -6.05166523e-04, -6.35551894e-04], [-2.82521245e-04, -
9.53527329e-04, -1.62591930e-03, ..., -7.36267770e-04, -3.89146406e-
04, -9.58586320e-04], [ 1.53309856e-05, 1.85055246e-04, 1.99895620e-
04, ..., 1.16856326e-03, 9.34558959e-04, 3.65192340e-04]])
```

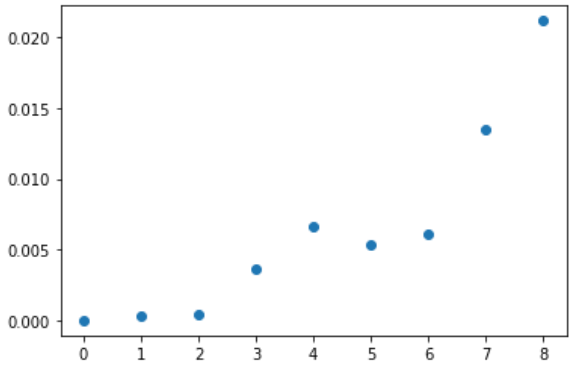
و مقادیر ویژه آن به صورت زیر هستند:



۳. تریس ماتریس مورد نظر واضحا وابسته به تعداد ویژگی نیست و با هر تعداد ویژگی این تریس برابر با یک عدد ثابت خواهد شد. چون مشخصا هربار ماتریس‌های پراکندگی برای کل داده‌ها و کل ویژگی‌ها حساب می‌شود.



اما اگر منظور سوال تعداد المنت تریس ماتریس بر حسب ویژگی‌هاست، با افزایش تعداد ویژگی‌ها مقدار تریس نیز در حال افزایش خواهد بود و نمودار به شکل زیر می‌شود و البته به دلیل اینکه مقدار ویژه‌ها استخراج و مرتب نشده لزوما بزرگترین مقادیر به ترتیب نیستند.



۴.

MNIST fashion، دیتاستی حاوی ۶۰۰۰۰ داده آموزش و ۱۰۰۰۰ داده برای تست می‌باشد. این داده‌ها عکس‌هایی در سایز ۲۸*۲۸ هستند. به عبارتی دیگر ۷۸۴ ویژگی برای هر داده داریم.

حال مراحل را یک بار با استفاده از کتابخانه sklearn و بدون آن انجام می‌دهیم که البته نتایج در هر دو حالت یکسان می‌شود.

بدون استفاده از کتابخانه SKlearn:

کلاس LDA را پیاده‌سازی کرده‌ایم که ماتریس‌های مورد نظر حساب می‌کند.

برای داده آموزش:

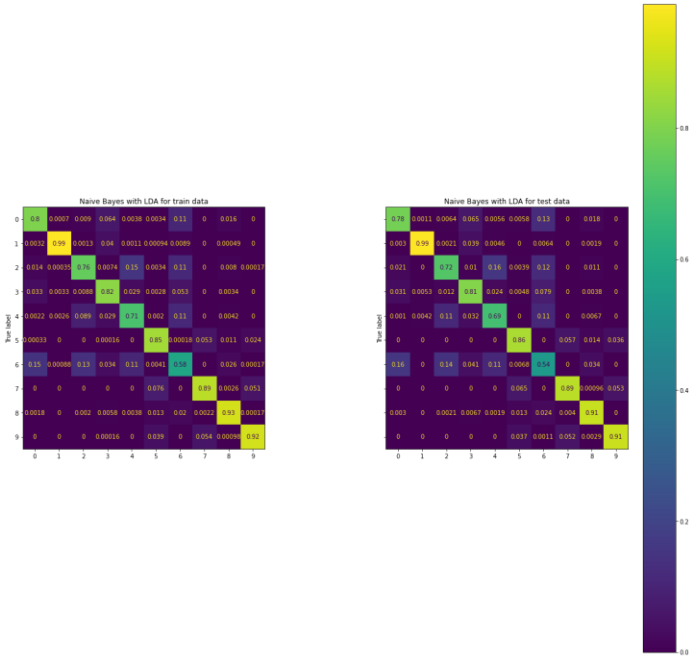
	precision	recall	f1-score	support
0	0.80	0.80	0.80	5980
1	0.94	0.99	0.97	5702
2	0.70	0.76	0.73	5540
3	0.87	0.82	0.84	6354
4	0.77	0.71	0.74	6511
5	0.91	0.85	0.88	6413
6	0.55	0.58	0.57	5648
7	0.87	0.89	0.88	5837
8	0.95	0.93	0.94	6153
9	0.90	0.92	0.91	5862
accuracy			0.83	60000
macro avg	0.83	0.83	0.83	60000
weighted avg	0.83	0.83	0.83	60000

برای داده تست:

	precision	recall	f1-score	support
0	0.77	0.78	0.78	987
1	0.94	0.99	0.96	951
2	0.67	0.72	0.70	931
3	0.84	0.81	0.82	1048
4	0.75	0.69	0.72	1078
5	0.89	0.86	0.88	1033
6	0.51	0.54	0.52	942
7	0.88	0.89	0.88	991
8	0.95	0.91	0.93	1044
9	0.91	0.91	0.91	995
accuracy			0.81	10000
macro avg	0.81	0.81	0.81	10000
weighted avg	0.81	0.81	0.81	10000

که CCR یا دقت نیز برای داده‌ی تست برابر با ۸۱ و برای داده آموزش برابر با ۸۳ درصد می‌باشد.

همچنین ماتریس درهمریختگی به صورت زیر می‌باشد:



۱. حال روند بالا را بدون استفاده از LDA تکرار می‌کنیم:

نتایج برای داده آموزش به صورت زیر است:

	precision	recall	f1-score	support
0	0.60	0.83	0.70	4327
1	0.95	0.57	0.71	10103
2	0.31	0.60	0.41	3130
3	0.44	0.42	0.43	6379
4	0.76	0.37	0.50	12410
5	0.25	0.92	0.40	1657
6	0.04	0.33	0.07	743
7	0.98	0.49	0.66	11901
8	0.72	0.85	0.78	5058
9	0.65	0.91	0.76	4292
accuracy			0.57	60000
macro avg	0.57	0.63	0.54	60000
weighted avg	0.73	0.57	0.60	60000

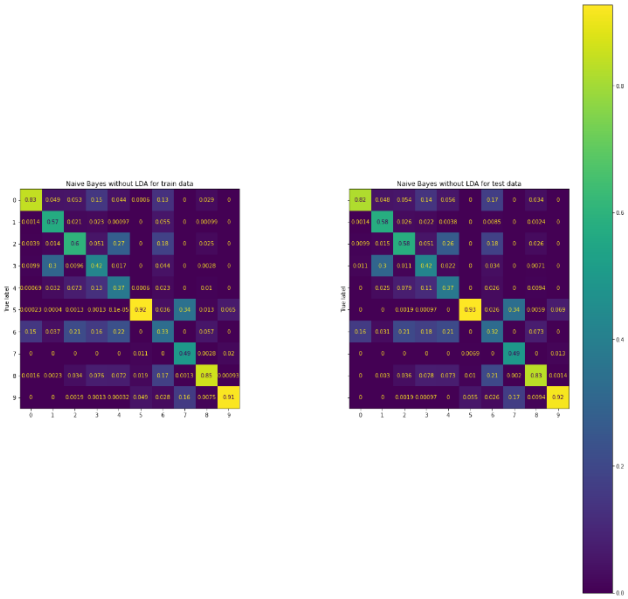
همچنین نتایج برای داده تست به صورت زیر است:

	precision	recall	f1-score	support
0	0.58	0.82	0.68	707
1	0.95	0.58	0.72	1650
2	0.31	0.58	0.41	533
3	0.43	0.42	0.43	1036
4	0.79	0.37	0.50	2122
5	0.27	0.93	0.42	291

6	0.04	0.32	0.07	117
7	0.99	0.49	0.66	1999
8	0.71	0.83	0.77	850
9	0.64	0.92	0.75	695
accuracy			0.57	10000
macro avg	0.57	0.63	0.54	10000
weighted avg	0.74	0.57	0.60	10000

که دقت مدل روی داده آموزش و تست هردو ۵۷ درصد است.

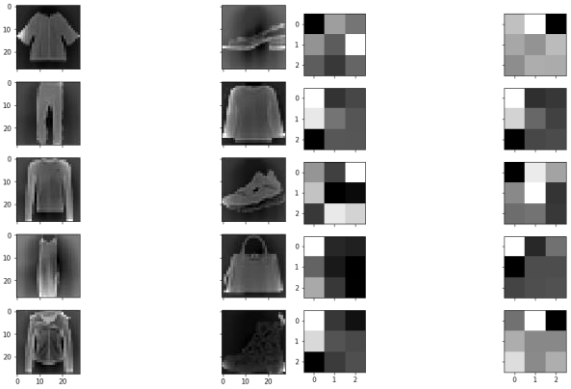
همچنین ماتریس درهمریختگی برای این مدل به صورت زیر است:



با توجه به متریک‌های بدست آمده مشخص است که مدل ما در حالتی که از LDA استفاده کنیم، نتایج بهتری از حالت بدون LDA به دست می‌آورد.

با استفاده از کتابخانه SKlearn:

از آنجا که ۱۰ کلاس داریم تعداد ویژگی‌های ما برابر با $9 = 1 - 10$ خواهد بود. و شکلی تعدادی از عکس‌ها قبل و پس از اعمال LDA به صورت زیر خواهد بود.



حال تخمین‌زن Bayes را برای ویژگی‌هامون آموزش می‌دهیم که نتایج آن برای داده تست و آموزش به ترتیب به

صورت زیر خواهد بود:

برای داده آموزش:

	precision	recall	f1-score	support
0	0.80	0.80	0.80	5980
1	0.94	0.99	0.97	5702
2	0.70	0.76	0.73	5540
3	0.87	0.82	0.84	6354
4	0.77	0.71	0.74	6511
5	0.91	0.85	0.88	6413
6	0.55	0.58	0.57	5648
7	0.87	0.89	0.88	5837
8	0.95	0.93	0.94	6153
9	0.90	0.92	0.91	5862
accuracy			0.83	60000
macro avg	0.83	0.83	0.83	60000
weighted avg	0.83	0.83	0.83	60000

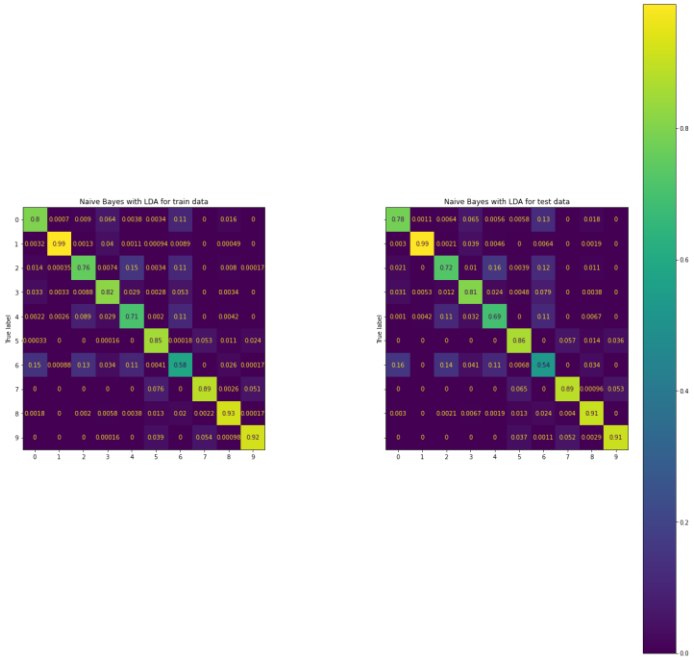
برای داده تست:

	precision	recall	f1-score	support
0	0.77	0.78	0.78	987
1	0.94	0.99	0.96	951
2	0.67	0.72	0.70	931
3	0.84	0.81	0.82	1048
4	0.75	0.69	0.72	1078
5	0.89	0.86	0.88	1033
6	0.51	0.54	0.52	942
7	0.88	0.89	0.88	991
8	0.95	0.91	0.93	1044
9	0.91	0.91	0.91	995
accuracy			0.81	10000

macro avg	0.81	0.81	0.81	10000
weighted avg	0.81	0.81	0.81	10000

که CCR یا دقت نیز برای داده‌ی تست برابر با ۸۱ و برای داده آموزش برابر با ۸۳ درصد می‌باشد.

همچنین ماتریس درهمریختگی به صورت زیر می‌باشد:



۲. حال روند بالا را بدون استفاده از LDA تکرار می‌کنیم:

نتایج برای داده آموزش به صورت زیر است:

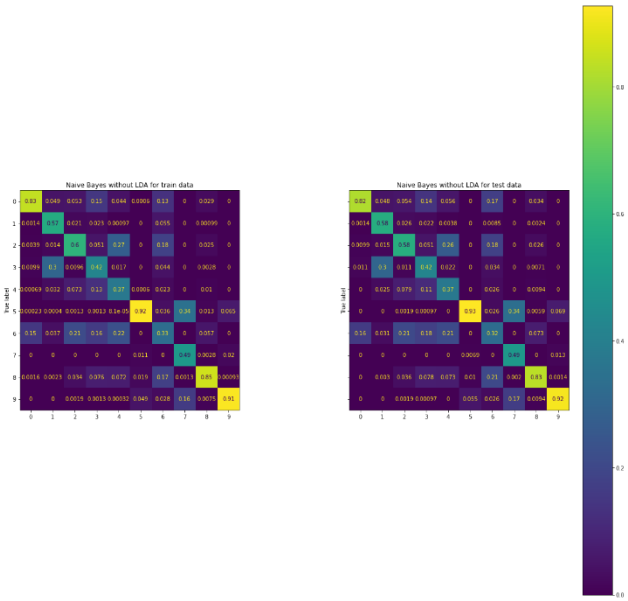
	precision	recall	f1-score	support
0	0.60	0.83	0.70	4327
1	0.95	0.57	0.71	10103
2	0.31	0.60	0.41	3130
3	0.44	0.42	0.43	6379
4	0.76	0.37	0.50	12410
5	0.25	0.92	0.40	1657
6	0.04	0.33	0.07	743
7	0.98	0.49	0.66	11901
8	0.72	0.85	0.78	5058
9	0.65	0.91	0.76	4292
accuracy			0.57	60000
macro avg	0.57	0.63	0.54	60000
weighted avg	0.73	0.57	0.60	60000

همچنین نتایج برای داده تست به صورت زیر است:

	precision	recall	f1-score	support
0	0.58	0.82	0.68	707
1	0.95	0.58	0.72	1650
2	0.31	0.58	0.41	533
3	0.43	0.42	0.43	1036
4	0.79	0.37	0.50	2122
5	0.27	0.93	0.42	291
6	0.04	0.32	0.07	117
7	0.99	0.49	0.66	1999
8	0.71	0.83	0.77	850
9	0.64	0.92	0.75	695
accuracy			0.57	10000
macro avg	0.57	0.63	0.54	10000
weighted avg	0.74	0.57	0.60	10000

که دقت مدل روی داده آموزش و تست هردو ۵۷ درصد است.

همچنین ماتریس درهمریختگی برای این مدل به صورت زیر است:



با توجه به متریک‌های بدست آمده مشخص است که مدل ما در حالتی که از LDA استفاده کنیم، نتایج بهتری از حالت بدون LDA به دست می‌آورد.

سوال ۶:

۱. خیر، لزوما فاصله بهترین معیار برای سنجش شباهت نیست.

برای مثال در شرایطی که داده‌ای با بعدهای بالاتر از سه و در کل بعدهای زیاد داریم، فاصله ممکن است معیار مناسبی نباشد زیرا در این حالت معنی کنار هم بودن همانند حالت دو و سه بعدی نیست. همچنین، معیار فاصله به هر بعد وزن برابری می‌دهد که این نیز می‌تواند مشکل ساز باشد.

۲. ایده اصلی الگوریتم DBSCAN این می‌باشد که نقاطی که فشرده و با چگالی بالا کنار یکدگیر هستند را در یک خوشه بگذارد.

مراحل این الگوریتم به صورت زیر می‌باشد:

مجموعه‌ای از نقاط را در نظر می‌گیریم و دو پارامتر ε و همچنین تعداد نقاط مینمم لازم برای نقطه اصلی (core point) بودن مثلا minPts را داریم، حال روند زیر را دنبال می‌کنیم:

- نقطه‌ی p اگر در فاصله‌ی ε آن k نقطه دیگر باشد آن نقطه اصلی می‌باشد.
- نقطه‌ی q اگر قابل دسترسی مستقیم (directly reachable) از نقطه p است اگر در فاصله‌ی ε از این نقطه باشد.
- نقطه‌ی q قابل دسترس از نقطه‌ی p است به صورتی که مسیری مانند $p_1,\ldots,p_n$ از p به q وجود داشته باشد که نقاط در این مسیر قابل دسترسی مستقیم باشند به جز نقطه q.
- هر نقطه‌ای که قابل دسترس از سایر نقاط نیست یک outlier یا نویز می‌باشد.

حال اگر p نقطه اصلی باشد، تمام نقاطی که از p قابل دسترس هستند با هم در یک خوشه هستند. هر خوشه حداقل یک نقطه اصلی دارد، و نقاط غیر اصلی نیز جزو خوشه هستند اما به عنوان نقطه گوشه‌ای شناخته می‌شوند.

مراحل بالا را انقدر تکرار می‌کنیم تا همه نقاط دارای خوشه شوند.

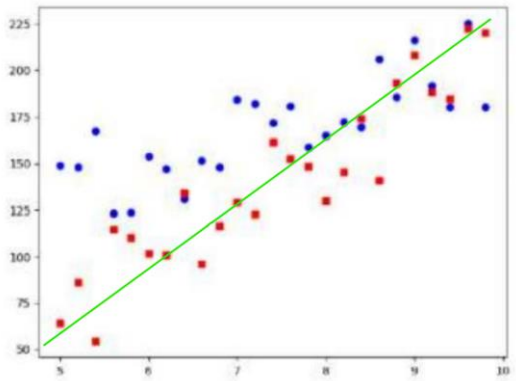
الگوریتم DBSCAN یک الگوریتم density-based clustering می‌باشد.

Optics یا Ordering points to identify the clustering structure نیز یک الگوریتم کلاسترینگ شبیه DBSCAN می‌باشد. اما مشکل تشخیص خوشه‌های معنی دار در داده هایی با چگالی متفاوت را سعی می‌کند که حل کند. برای انجام این کار، نقاط به گونه‌ای مرتب می شوند که نزدیکترین نقاط از نظر مکانی به همسایگی در ترتیب تبدیل می شوند. علاوه بر این، برای هر نقطه یک فاصله ویژه ذخیره می شود که نشان دهنده چگالی است که باید برای یک خوشه پذیرفته شود تا هر دو نقطه متعلق به یک خوشه باشند.

الگوریتم optics علاوه بر پارامترهای تعریف شده در بخش قبل یعنی ε و minPts دو پارامتر دیگر core-dist و reachability-dist را نیز دارا است. تفاوت در optic این است که به جای تثبیت MinPts و ε ، ما فقط MinPts را ثابت می کنیم و شعاع را ترسیم می کنیم که در آن یک نقطه توسط DBSCAN متراکم در نظر گرفته شود. برای مرتب سازی اشیاء در این الگوریتم، آنها را در یک پشته اولویت استفاده می‌شود، به طوری که اشیاء نزدیک در پشته قرار می گیرند. به همین دلیل استفاده از این ساختمان داده این الگوریتم از DBSCAN بهینه‌تر است.

سوال ۷:

PCA تلاش می‌کند که یک ترکیب خطی از ویژگی‌ها بدهد که بیشترین واریانس داده‌ها در آن موجود باشد. بنابراین محور انتخابی آن به صورت زیر خواهد شد.



LDA اما تلاش می‌کند جدایی پذیری کلاس‌ها را بیشترین حد کند. به عبارت دیگر LDA سعی می‌کند تفاوت بین کلاسی را ماکسیمم کند در حالی که تفاوت داخلی کلاسی مینمم می‌شود. پس محور انتخابی آن به صورت زیر خواهد بود: