به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

ماشین لرنینگ

تمرین 5

نام و نام خانوادگی	محمد مشرقی
شماره دانشجویی	810199492
تاریخ ارسال گزارش	

فهرست

۱- ۳

الف ۳

ب ۴

2 ..... ۵

3 ..... ۶

الف ۶

ب ۸

4 ..... ۹

ب ۱۰

5 ..... ۱۱

الف ۱۱

ب ۱۵

6 ..... ۱۸

ب ۲۰

۱-

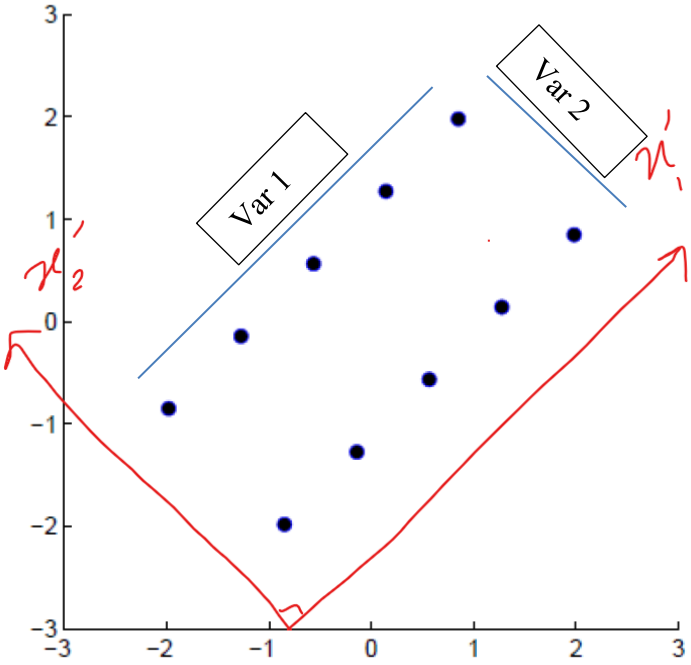
الف

الف) نمایش محور اول و دوم PCA برای داده هایی که در شکل نمایش داده شده اند. آیا نوع برچسب داده ها در انتخاب محورهای PCA موثر است؟ چرا؟ (۵ نمره)

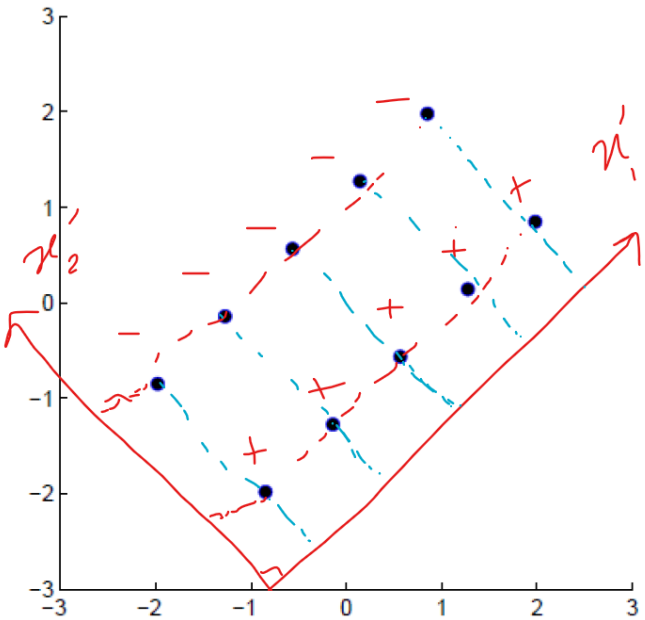
خیر

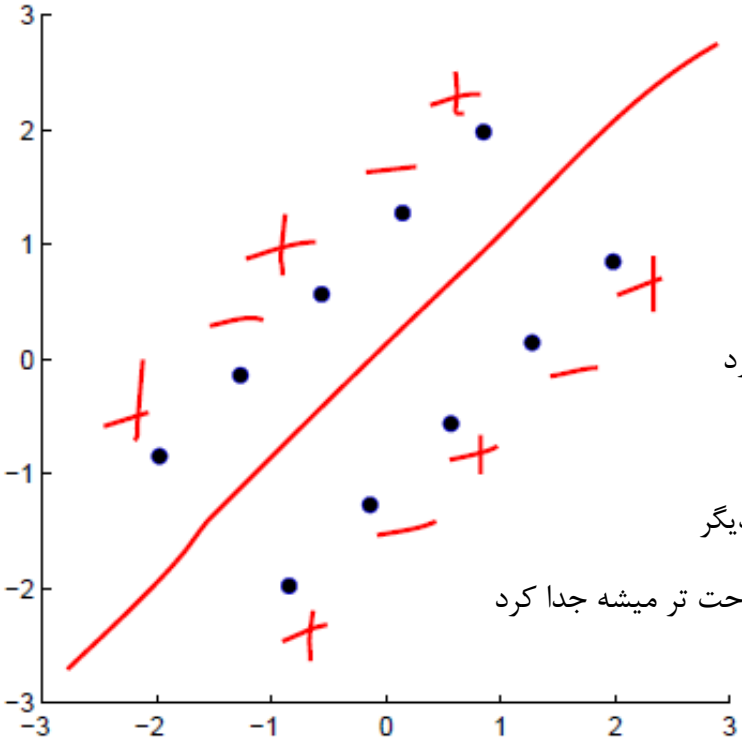
در یادگیری ماشین، PCA(تجزیه مؤلفه‌های اصلی) یک تکنیک بدون ناظر (unsupervised) برای کاهش ابعاد معمولاً استفاده می‌شود که برای استخراج ویژگی و تجسم داده به کار می‌رود. در فرآیند محاسباتPCA ، برچسب‌ها یا اطلاعات کلاس مستقیماً در نظر گرفته نمی‌شوند.

در PCA داده به نحوی کاهش بعد پیدا می کنند که **بیشترین واریانس** ممکن رو داشته باشند.



ژ





ب

2D data: 100% error
1D data from PCA: 0% error

در اینجا

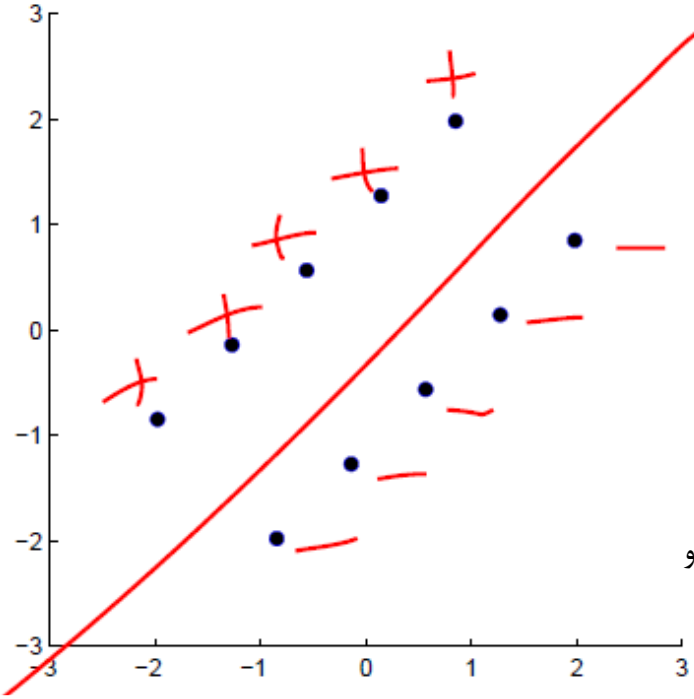
دو بعدی نمی توان مسیر بهینه ای پیدا کرد

و اگر خطی ترسیم کنه هر دو خطا دارند

در یک بعدی چون نقاط مثبت روی هم دیگر

و نقاط منفی هم روی یکدیگر می افتند راحت تر میشه جدا کرد

و جدایی دقیق میشود



2D data 0% error
1D data from PCA: 100% error

در اینجا

در دو بعد هر کدوم رو می توان با یک خطی بدست

آورد و دقت صد درصد گرفت

اما در یک بعدی چون داده های مثبت و منفی

رو هم قرار می گیرند اصلا قابل تشخیص نیست و

بدرد نمی خورد

Q_2 $\quad S_1 = \sum_{i=1}^{n} (y_{1i} - m_1) \Rightarrow S_1^2 = \sum_{i=1}^{n} (y_{1i} - m_1)^2 = \sum_{i=1}^{n} y_{1i}^2 + m_1^2 - 2m_1 y_{1i}$

$$\Longrightarrow S_1^2 = \sum_{i=1}^{n} y_{1j}^2 + m_1^2 n - 2m_1^2 n_1 = \sum_{i=1}^{n} y_{1j}^2 - m_1^2 n_1$$

$$\boxed{\sum_{i=1}^{n} y_{1j}^2 = S_1^2 + m_1^2 n_1} \quad , \quad \boxed{\sum_{i=1}^{n} y_{2j}^2 = S_2^2 + m_2^2 n_2}$$

به طور مشابه می‌گیریم

$$J = \frac{1}{n_1 n_2} \sum \sum (y_i - y_j)^2 = \frac{1}{n_1 n_2} \sum \sum \left(y_{1i}^2 + y_{2j}^2 - 2 y_{1i} y_{2j} \right)$$

$$= \frac{1}{n_1 n_2} \sum \left[n_2 y_{1i}^2 + S_2^2 + n_2 m_2^2 - 2 y_{1i} n_2 m_2 \right]$$

$$= \frac{1}{n_1 n_2} \sum \left[n_2 S_1^2 + n_2 n_1 m_1^2 + n_1 S_2^2 + n_1 n_2 m_2^2 - 2 n_2 m_2 n_1 m_1 \right]$$

$$= \frac{1}{n_1} S_1^2 + \underline{m_1^2} + \frac{1}{n_2} S_2^2 + \underline{m_2^2} - \underline{2 m_1 m_2}$$

$$= \frac{1}{n_1} S_1^2 + \frac{1}{n_2} S_2^2 + \left[m_1 - m_2 \right]^2$$

الف

a) $P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$

$$\hat{\lambda} = \arg\max_{\lambda} E_z\left[\sum_{i=1}^{n} \log p(x_i, z_i \mid \lambda)\right]$$

$$P(x, \lambda) = \sum_{k=1}^{k} \alpha_k c(x \mid \lambda_k) \; ; \; \sum_{k=1}^{k} \alpha_k = 1$$

$$\left.\begin{array}{l} D = \{x_1, \dots, x_n\} \\ H = \{z_1, \dots, z_n\} \end{array}\right\} \xrightarrow[\text{encoder}]{\text{one hot}} z_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{k \times 1} \Rightarrow P(x_i, z_i \mid \lambda) = \prod_{k=1}^{k} (\alpha_k c(x \mid \lambda_k))^{z_{ik}}$$

$$L(\lambda) = \log P(D, H \mid \lambda) = \sum_{i=1}^{n} \log p(x_i, z_i \mid \lambda) = \sum$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{k} z_{ik} (\log(\alpha_k) + \log c(x \mid \lambda_k))$$

$$\Rightarrow E_z[L(\lambda)] = \sum_{i=1}^{n} \sum_{k=1}^{k} E[z_{ik}] (\log(\alpha_k) + \log c(x \mid \lambda))$$

$$E_{z_i \mid x_i}[z_{ik}] = E[z_{ik} \mid x_{ik}] = Pr(z_{ik} = 1 \mid x_i, \lambda^t) = \frac{P(x_i \mid z_{ik} = 1, \lambda^t) P(z_{ik} = 1)}{P(x_i \mid \lambda^t)}$$

$$= \frac{c(x_i \mid \lambda_k^t) \alpha_k^t}{\sum_{k=1}^{k} \alpha_k c(x_i \mid \lambda_k^t)} = \gamma_{ik}^t$$

$$\arg\max_{\lambda} \sum_{i=1}^{n} \sum_{k=1}^{k} \gamma_{ik}^{t} \left(\log \alpha_k + \log c(x \mid \lambda_k) \right)$$

$$s.t \sum_{k=1}^{k} \alpha_k = 1$$

$$\Rightarrow Q(\lambda) = \sum_{i=1}^{n} \sum_{k=1}^{k} \gamma_{ik}^{t} \left(\log \alpha_k + \log c(x \mid \lambda_k) \right) - \mu \left(\sum_{k=1}^{k} \alpha_k - 1 \right)$$

مشتق می‌گیریم

$$\Rightarrow \frac{\partial Q(\lambda)}{\partial \alpha_j} = \sum_{i=1}^{n} \frac{\gamma_{ij}^{t}}{\alpha_j} - \mu = 0 \Rightarrow \alpha_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^{t}}{\mu}$$

مالا کل α را جمع می‌کنیم

$$\Rightarrow \sum_{j=1}^{k} \alpha_j = \frac{\sum_{j=1}^{k} \sum_{i=1}^{n} \gamma_{ij}^{t}}{\mu} = 1 \Rightarrow \boxed{\mu = \sum_{j=1}^{k} \sum_{i=1}^{n} \gamma_{ij}^{t}}$$

$$\Rightarrow \alpha_j = \frac{\sum_{i=1}^{n} \gamma_{ij}^{t}}{\sum_{j=1}^{k} \sum_{i=1}^{n} \gamma_{ij}^{t}} = \frac{\sum_{i=1}^{n} \gamma_{ij}^{t}}{n} \Rightarrow \boxed{\hat{\alpha}_j = \sum_{i=1}^{n} \frac{\gamma_{ij}^{t}}{n}}$$

$$\frac{\partial Q(\lambda)}{\partial \lambda_j} = \sum_{i=1}^{n} \gamma_{ij}^{t} \left(\frac{x_i}{\lambda_j} - 1 \right) = 0 \Rightarrow \boxed{\hat{\lambda}_j = \frac{\sum \gamma_{ij}^{t} x_i}{\sum_{i=1}^{n} \gamma_{ij}^{t}}}$$

ب

3) b) $C = X^T X = \sum_{i=1}^{n} x_i x_i^T$

Sample covariance $= \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$

PCA: $\max_p p^T C p \Rightarrow L(p) = p^T C p - \lambda (\|p\|_2^2 - 1)$

s.t. $\|p\|_2^2 = 1$

$\Rightarrow \nabla_p L(p) = 0 \Rightarrow 2Cp - 2\lambda p = 0 \Rightarrow \boxed{Cp = \lambda p}$

بردار ویژه

مقدار ویژه

$Var(z) = p^T C p \xrightarrow{p_1} Var(z) = p_1^T (\lambda) p_1 = \lambda_1 p_1^T p_1 = \lambda_1 \underbrace{\|p_1\|_2^2}_{=1} = \lambda$

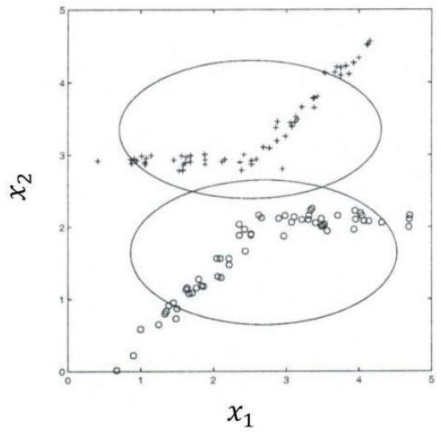
حال واریانس می‌گیریم

$$\boxed{Var(z) = \lambda_1}$$

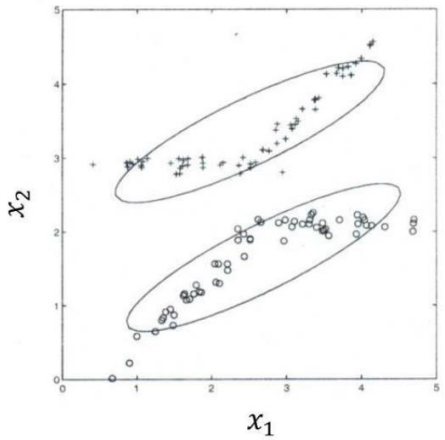
4

الف)فرض کنید دو مدل زیر نتیجه اعمال الگوریتم *EM* و استفاده از Gaussian Mixture Modell باشد،

با ذکر دلیل توضیح دهید کدام یک از مدلهای زیر مناسب تر است (۵ نمره)

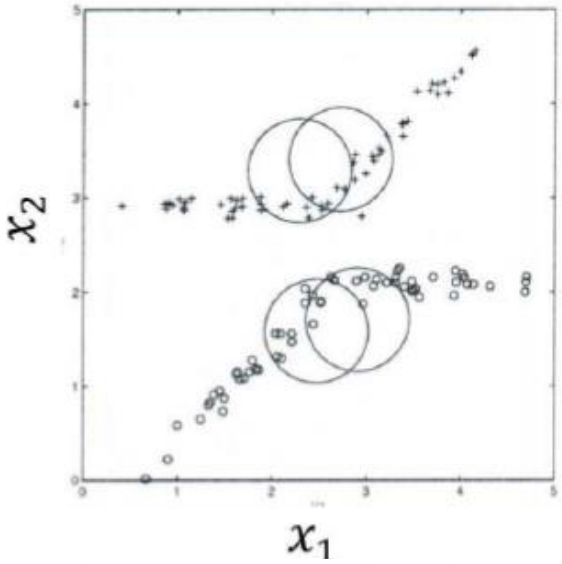


ب



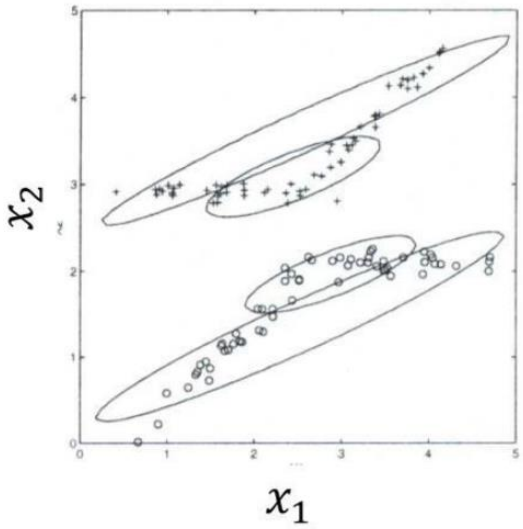
الف

گزینه الف بهتر است چون در ب کلی ناحیه در بر دارد که در آن داده نیست و هایپر پارامتر آن درست ست نشده منظور این است که در ب کلی ناحیه هست که احتمال میره در آن داده ای باشد اما نیست در حالی که در الف این جور نواحی خیلی خیلی کمتر است.

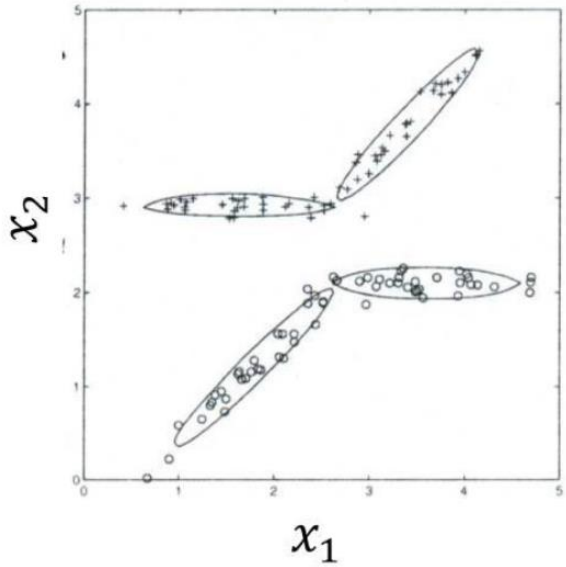


ب

با ذکر دلیل توضیح دهید کدام یک از موارد زیر خروجی اولین گام الگوریتم EM و استفاده از Gaussian Mixture Modelخواهد بود .(۵ نمره)



ب



الف

خروجی بعدی الف خواهد بود چون در روش gaussian mixture model دنبال اضافه کردن نزدیک ترین داده به داده های خود میل می کند که به فرم خود نزدیک تر است که این مثال نقضی برای ب هستش چون داده های رو جمع اوری کرده که خیلی خیلی از مرکز شروع ابتدایی فاصله داشته

5

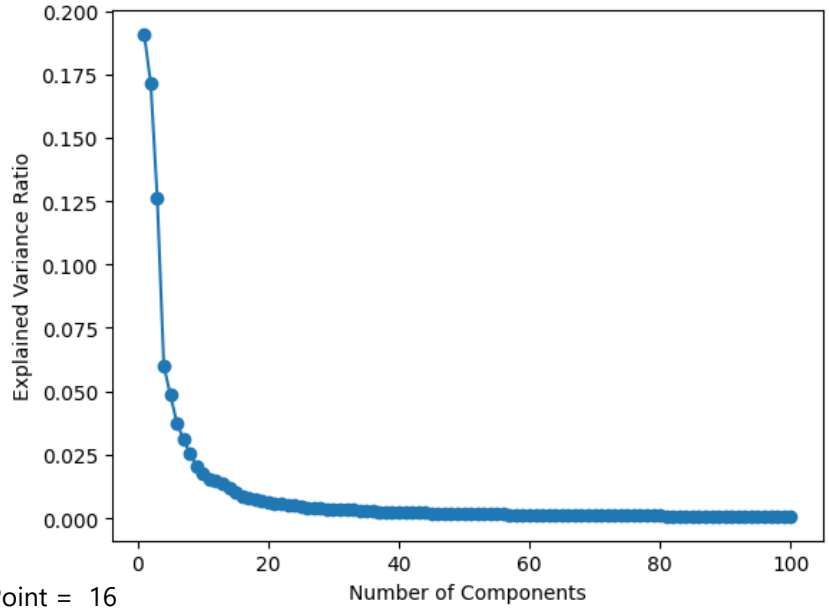
الف

ابتدا داده ها رو می گیریم و تعداد و فیچر آن را در میاریم

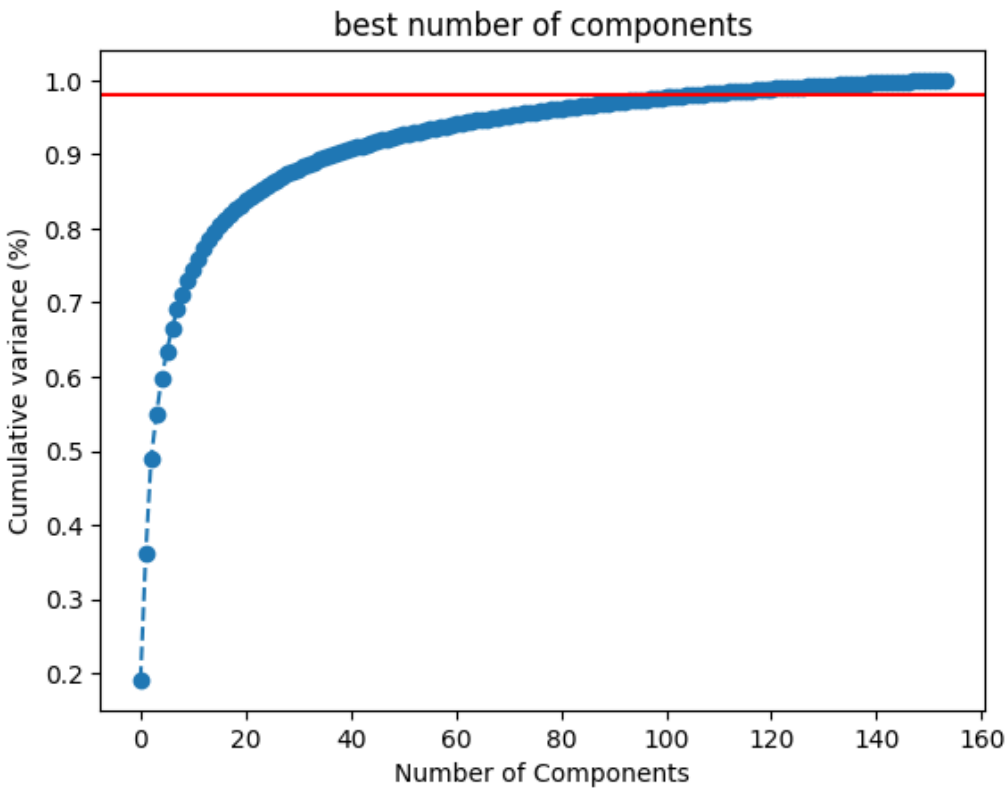
```
samples =  154
feature =  65536
```

سپس با PCA از ۱۵۴ فضا می بریم و مقایسه می کنیم

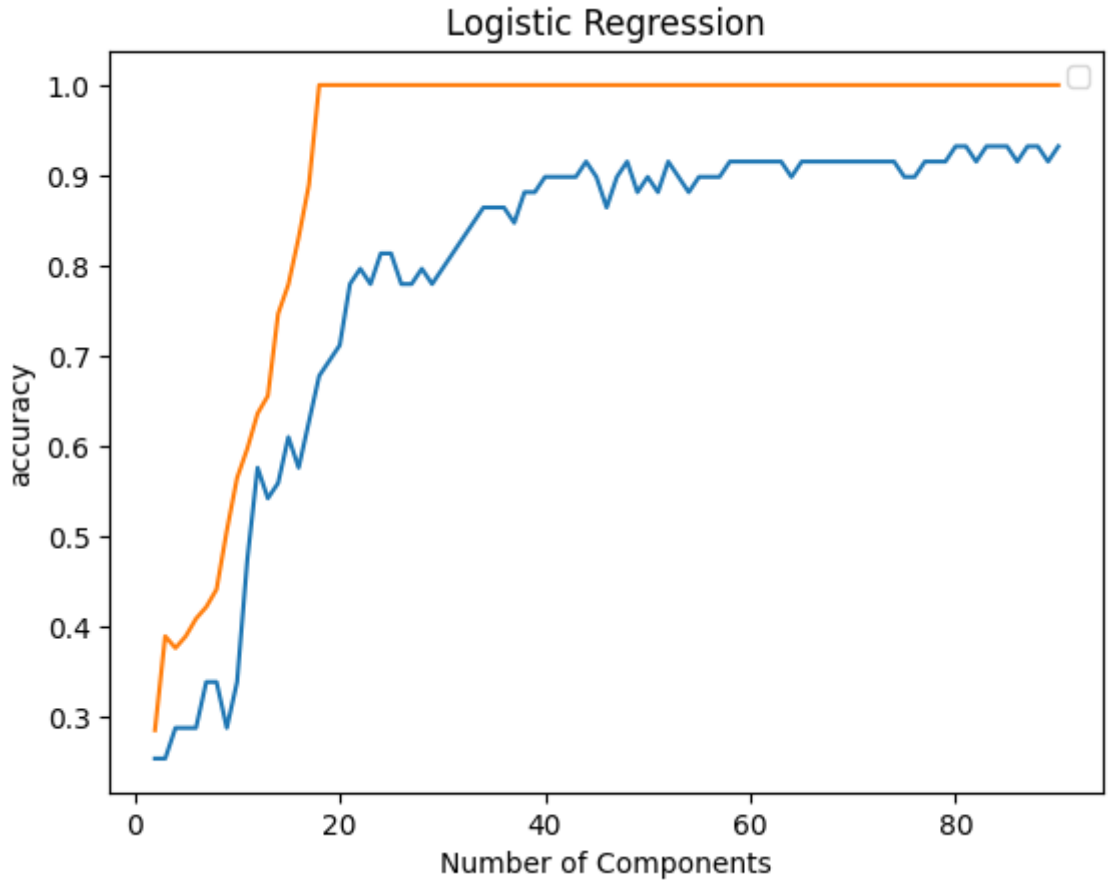
حال با توجه به عکس نقطه زانویی را در می یابیم :



Elbow Point = 16



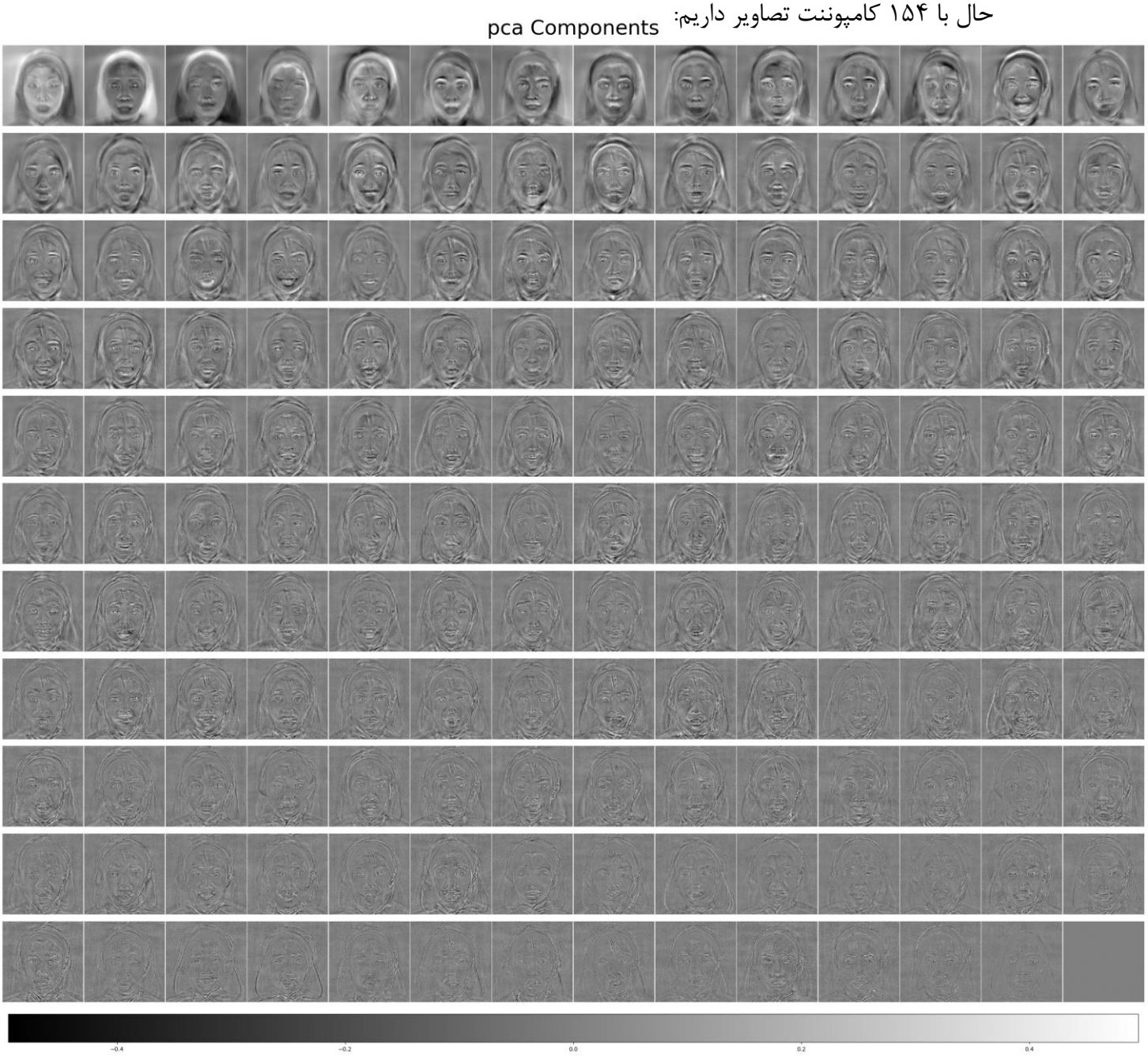
برای اینکه بفهمیم بهترین تعداد کامپوننت چند تاست می توان با جمع کردن آن و گذاشتن یک حد بهترین آن را بیابیم با توجه به شکل بالا حد ما 0.98 بود که در نتیجه برای آن ۱۰۰ تا کامپوننت نیاز است.



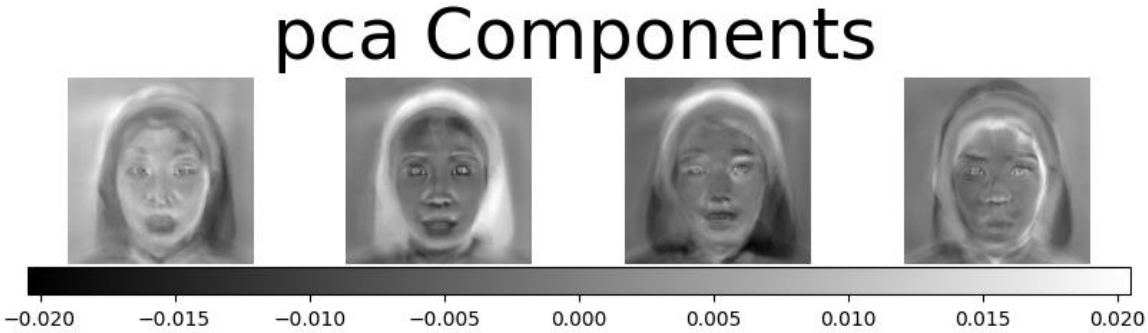
می بینیم که حتی با لاجستیک رگرشن با بیشتر شدن تعداد کامپوننت دقت بالا می رود

ب

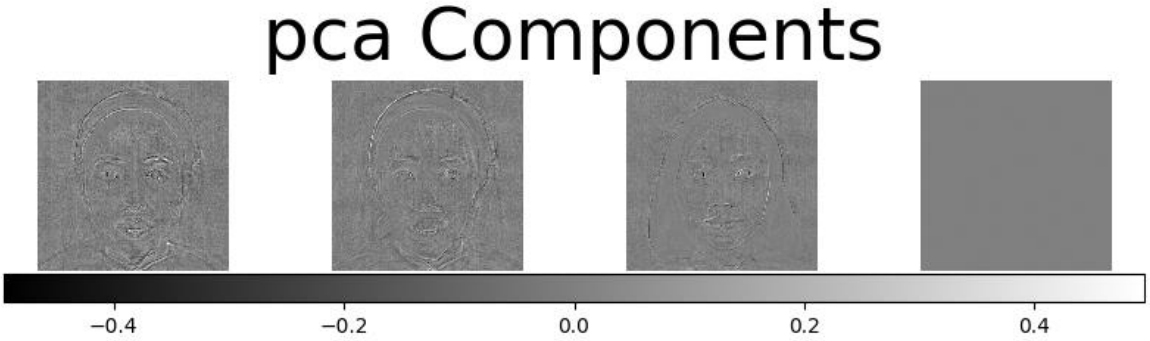
حال با ۱۵۴ کامپوننت تصاویر داریم:



۴تا اول



۴تای آخر



با توجه به دو عکس می بینیم که مقادیر ویژه اول بزرگتر از آخری ها هستن و جزییات بیشتری معلوم است چون واریانس بزرگتری دارند و جزییات بیشتر واضح اند چون پراکندگی بیشتری دارد

اما ۴ تا اخر نمیتوان به خوبی تفاوت را دید و جزییات خیلی کمی را نشر می دهند. و پراکندگی کمتر دارد

ب

ماتریس پراکندگی درون کلاسی :

```
[[     514      448      448 ...      439      372      379]
 [     448     3723     6467 ...    10278     4314      664]
 [     448     6467    35431 ...    46530    13256     2456]
 ...
 [     439    10278    46530 ... 18310882  6260804   530248]
 [     372     4314    13256 ...  6260804  5045140   573847]
 [     379      664     2456 ...   530248   573847   258841]]
```

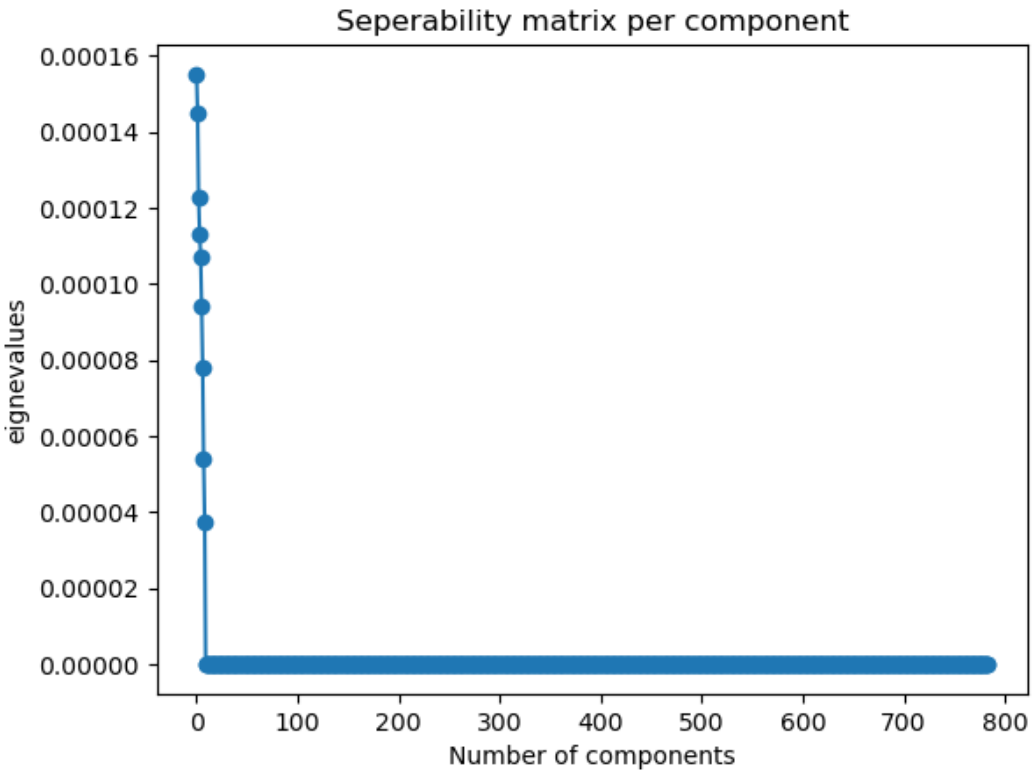
بین کلاسی:

```
 [[ 1.29888889e-05  2.22888889e-05  2.02861111e-04 ... -9.18427778e-03
  -3.35802778e-03 -1.51066667e-04]
 [ 2.22888889e-05  3.37947222e-04  2.31965278e-03 ...  1.08641944e-02
   6.06411111e-03  1.30763611e-03]
 [ 2.02861111e-04  2.31965278e-03  1.98100139e-02 ... -1.20862250e-01
  -8.42765556e-02 -3.69195833e-03]
 ...
 [-9.18427778e-03  1.08641944e-02 -1.20862250e-01 ...  5.93001829e+01
   1.61146536e+01  1.30130458e+00]
 [-3.35802778e-03  6.06411111e-03 -8.42765556e-02 ...  1.61146536e+01
   6.90668900e+00  5.64335194e-01]
 [-1.51066667e-04  1.30763611e-03 -3.69195833e-03 ...  1.30130458e+00
   5.64335194e-01  5.72470583e-02]]
```

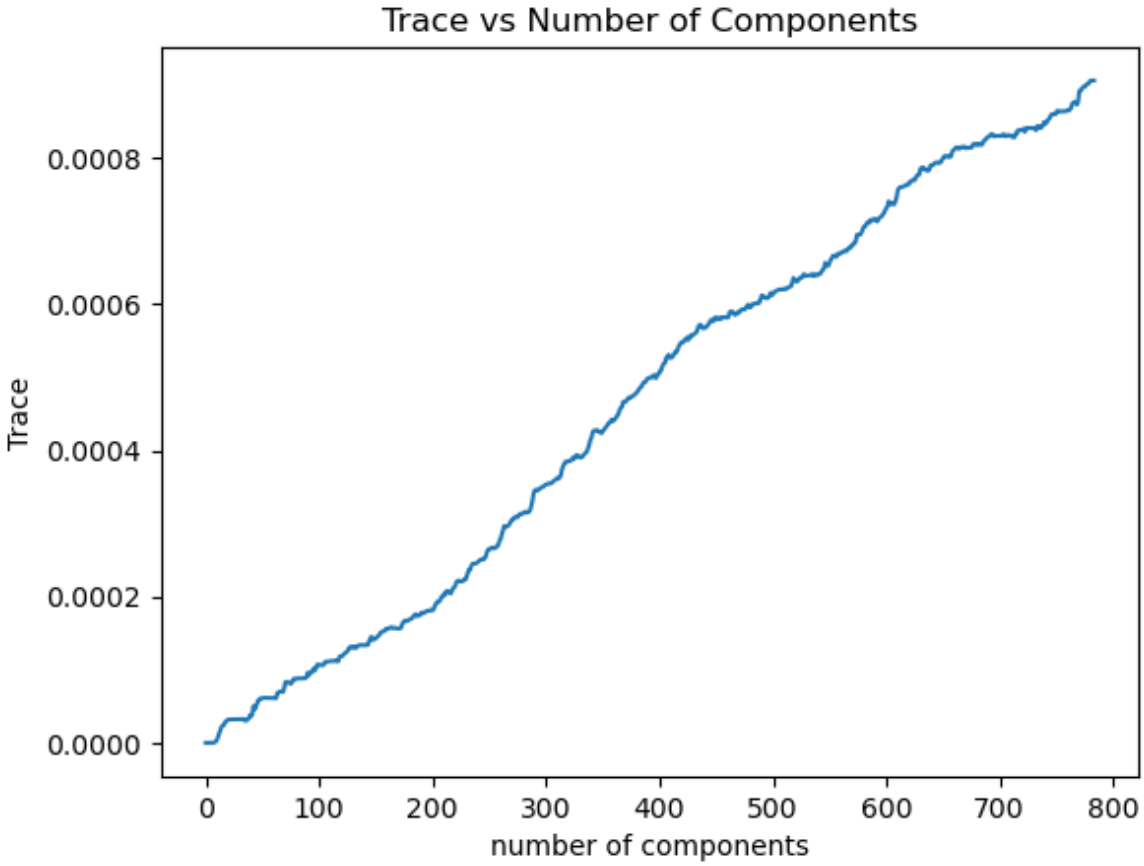
ماتریس جدا پذیر:

```
[[ 2.13934914e-08 -1.02190875e-08 -3.13146643e-08 ... -8.05114621e-06
  -2.39457391e-06 -8.20089262e-08]
 [-5.75212111e-09  1.88008271e-08  4.35162725e-08 ... -3.93126223e-06
   7.27613772e-07  1.76273832e-07]
 [ 2.40659769e-09  5.04017094e-09  1.08547239e-07 ... -3.05113466e-07
  -1.10193495e-06 -1.80593923e-07]
 ...
 [ 1.63087843e-10 -4.05208988e-10 -1.87719999e-09 ...  1.51242469e-07
  -3.13233567e-09 -2.58033051e-09]
 [-1.13436496e-10 -1.44236991e-09 -6.63855026e-09 ... -6.23108281e-08
  -2.46009182e-08 -1.36722401e-08]
 [ 1.03122389e-10  1.37615825e-09  6.48750320e-09 ...  5.61157138e-07
   1.56562554e-07  1.53267886e-08]]
```

ماتریس جدایی پذیر بر تعداد کامپوننت



ب

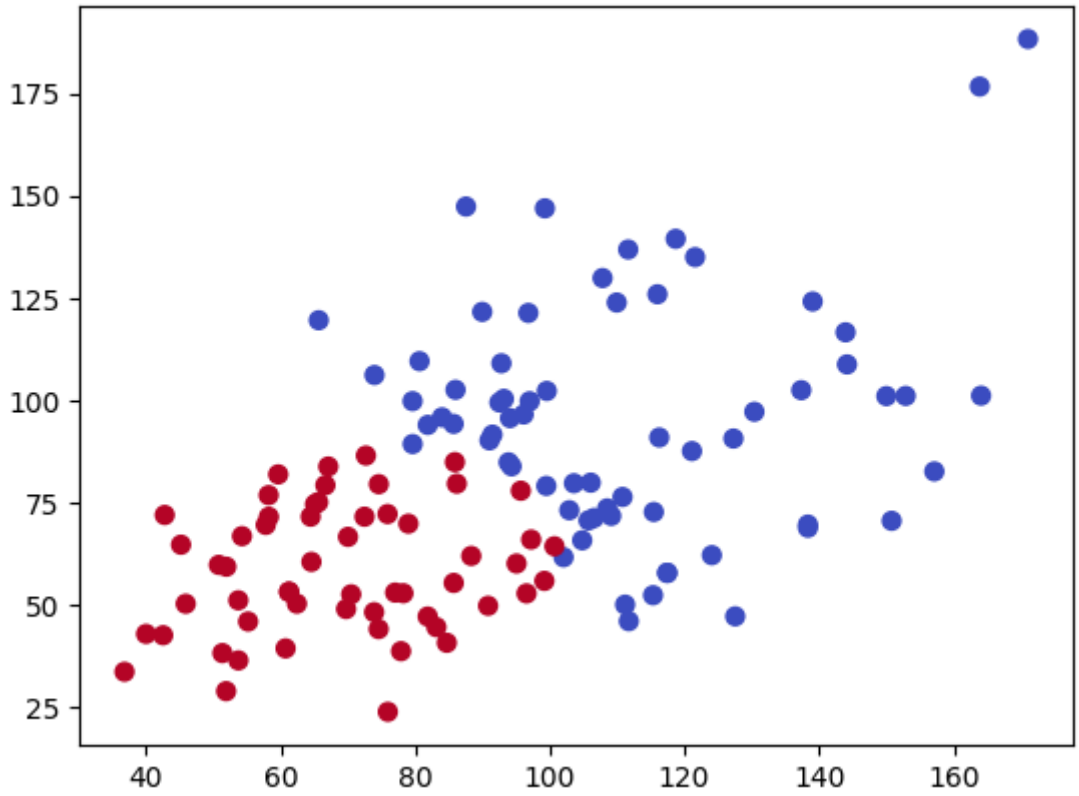


با توجه به عکس هرچی تعداد کامپوننت بیشتر می شود تریس ما بزرگتر می شود و این یعنی پراکندگی بین کلاسی بیشتر و نتایج بهتری بدست میاد از طریق LDA

6

ابتدا فایل ها را مرتب می کنیم و لیبل بندی می کنیم.

بعد مطابق عکس با دو کامپوننت استفاده کنیم داریم:



مقادیرمون :

```
covariance matrix :  [[[750.50925833 268.02257105]
                       [268.02257105 973.96356225]]

 [[399.9097764  132.02703445]
  [132.02703445 308.16754808]]]
```

ماتریس اول برا داده های آبی و دومی برا قرمز

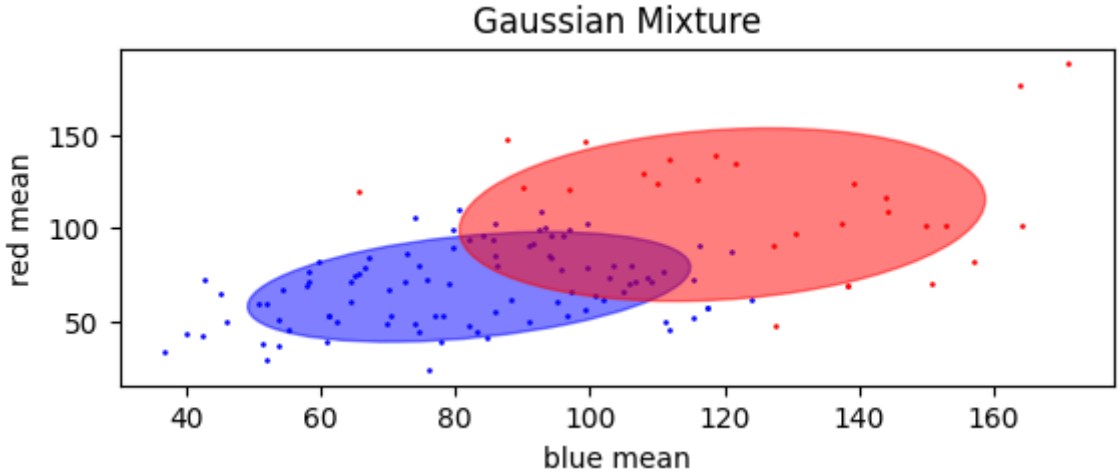
نسبت دو داده:

```
Weights  :  [0.55838044 0.44161956]
```

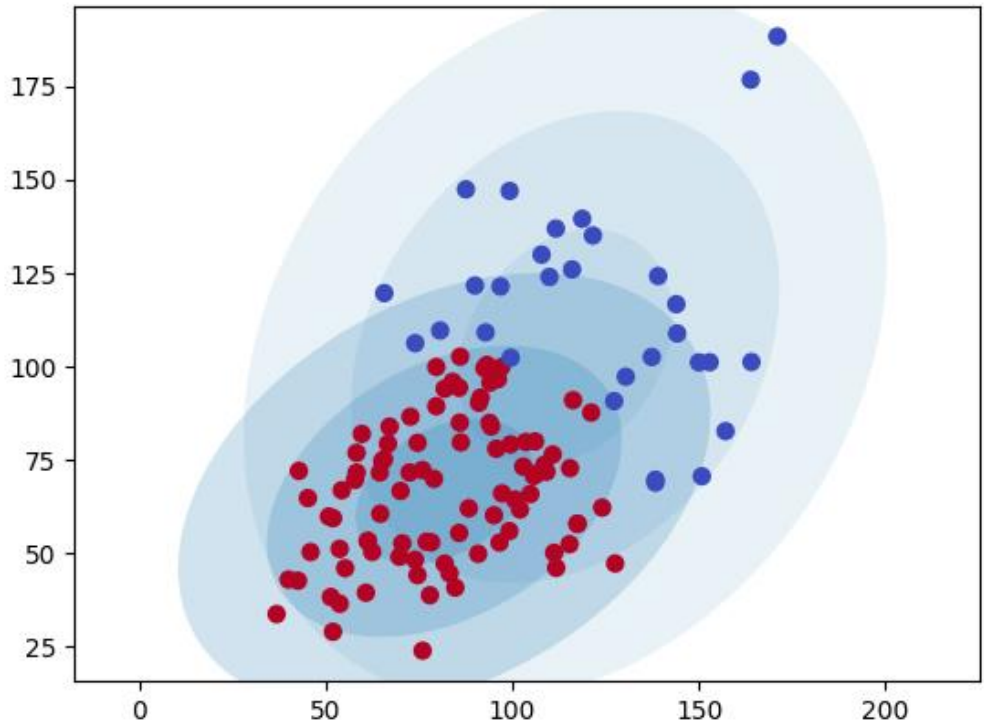
میانگین رنگ آبی و رنگ قرمز برا داده های آبی و قرمز

```
mean :   [[106.81416325  91.77359052]
 [ 73.21415131  62.11229303]]
```

که خروجی نهایی :



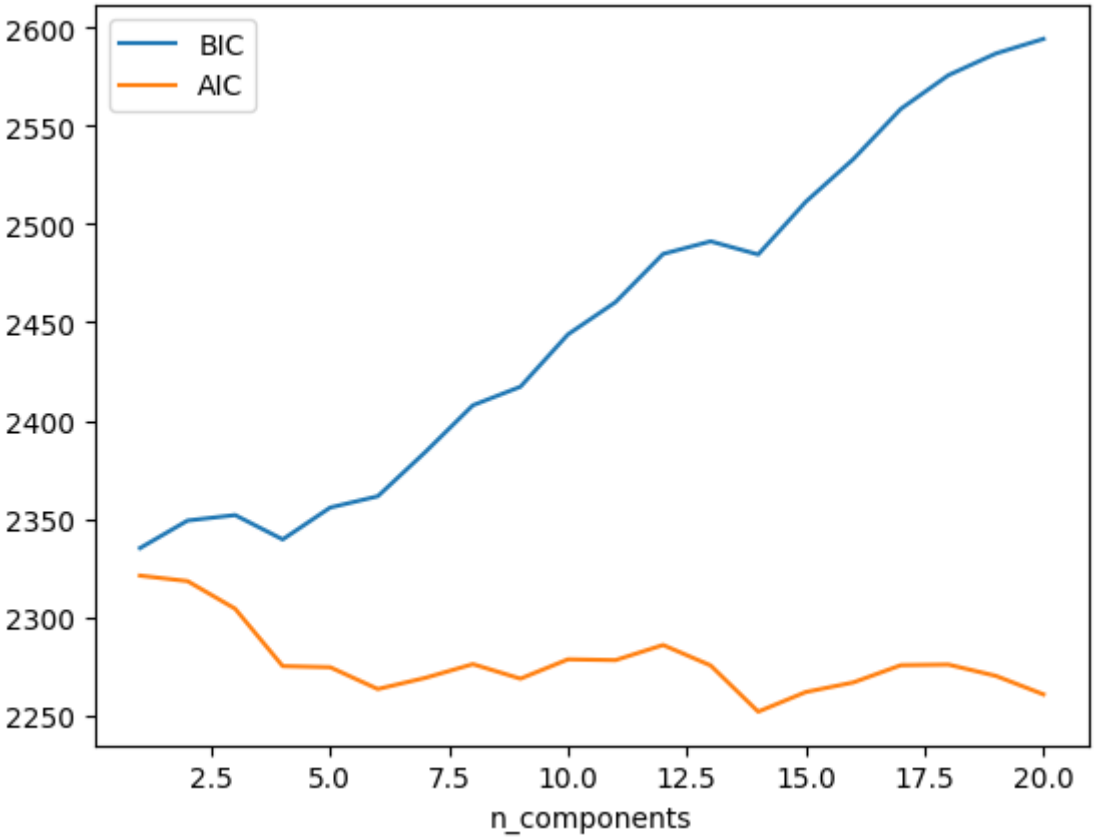
حال برای کانکتور داریم



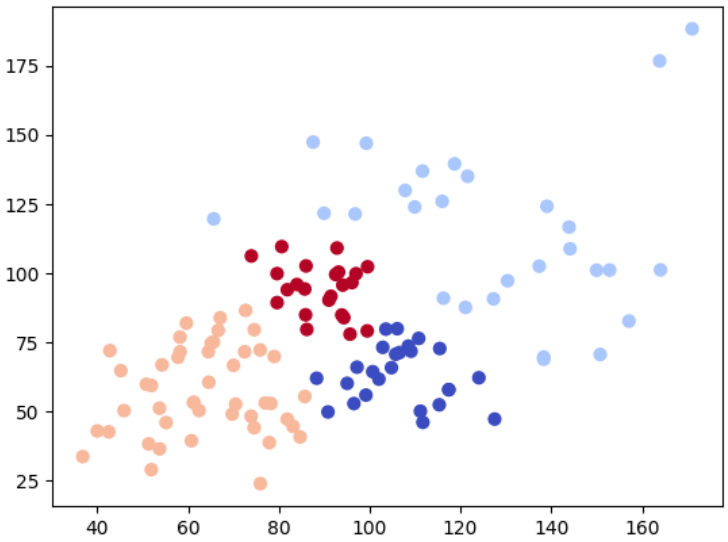
با توجه به عکس می بینیم که هر کدام از نقاط به دو مرکز میانگین آبی یا قرمز بوده اند آن رنگی شدند.

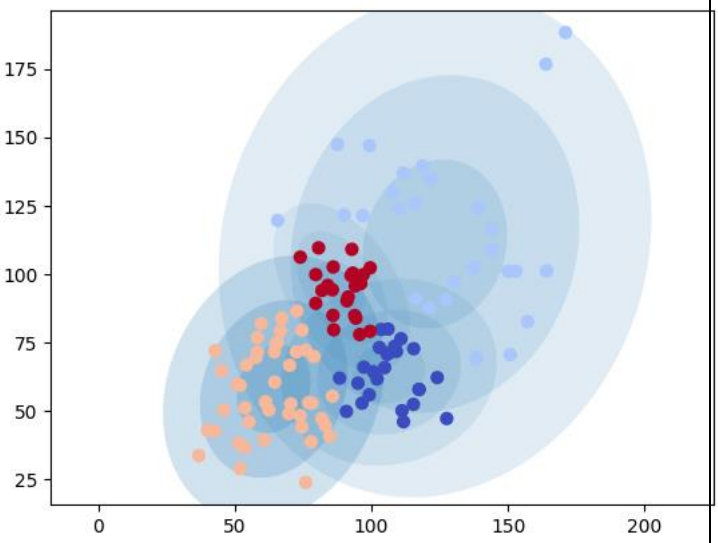
ب

اگر بخواهیم با نمودار BiC و AIC بسنجیم بهترین نقطه برابر با ۴ هست.



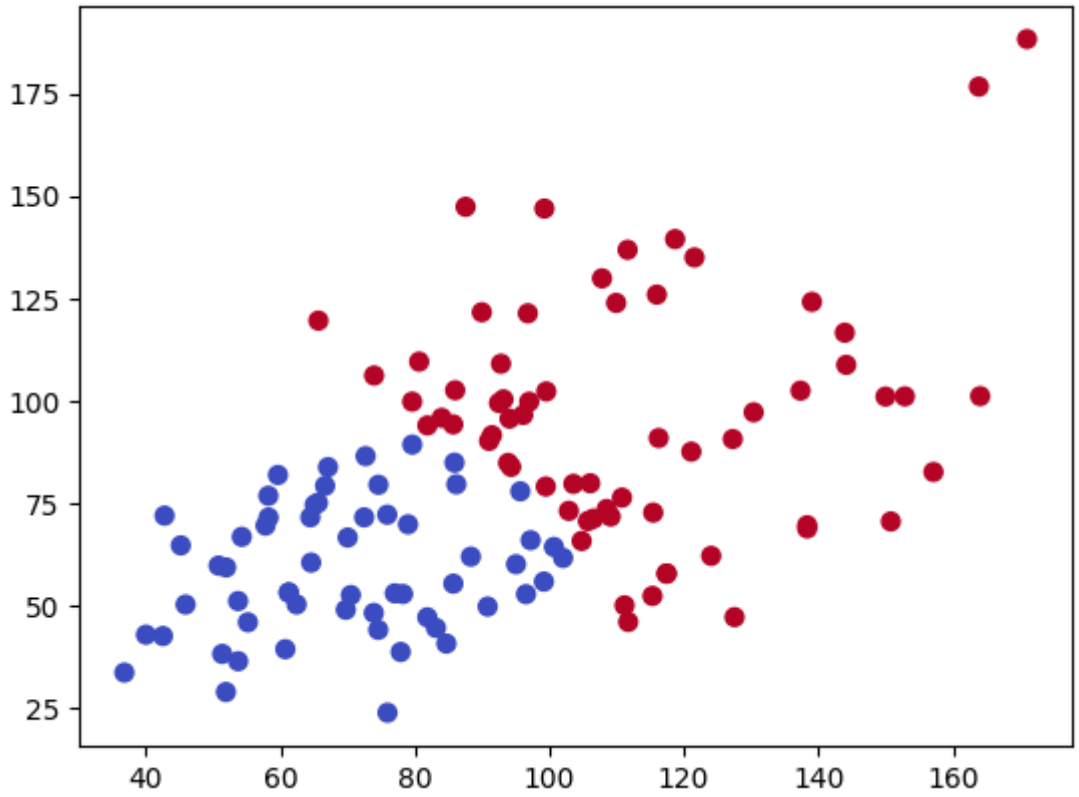
که شکل آن





اگر از روش K fold استفاده کنیم و کراس ولیدیشن بزنیم بهترین کامپوننت را ۲ به ما می دهد

```
best_score =  -9.571189380771305
```



و بقیه نتایج در بالا الف اورده شده است.