به نام هستی بخش

تمرین کامپیوتری چهارم

درس مقدمه‌ای بر پردازش سیگنال‌های پزشکی

نویسنده: حمیدرضا ابوئی

شماره دانشجویی: ۹۷۳۳۰۰۲

استاد:

دکتر مرادی

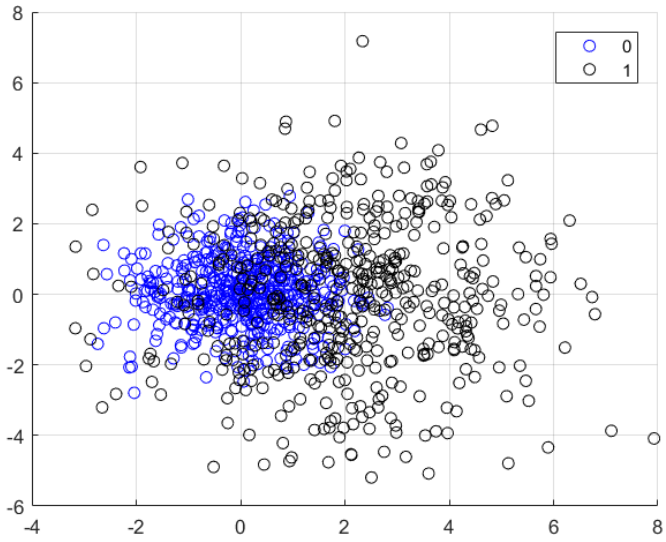
تدریس‌یار:

زهرا دیانی

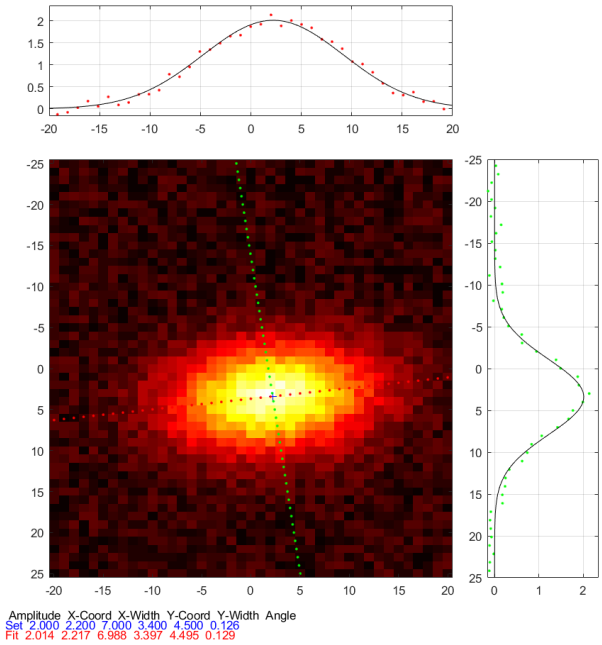
پاییز ۱۴۰۰

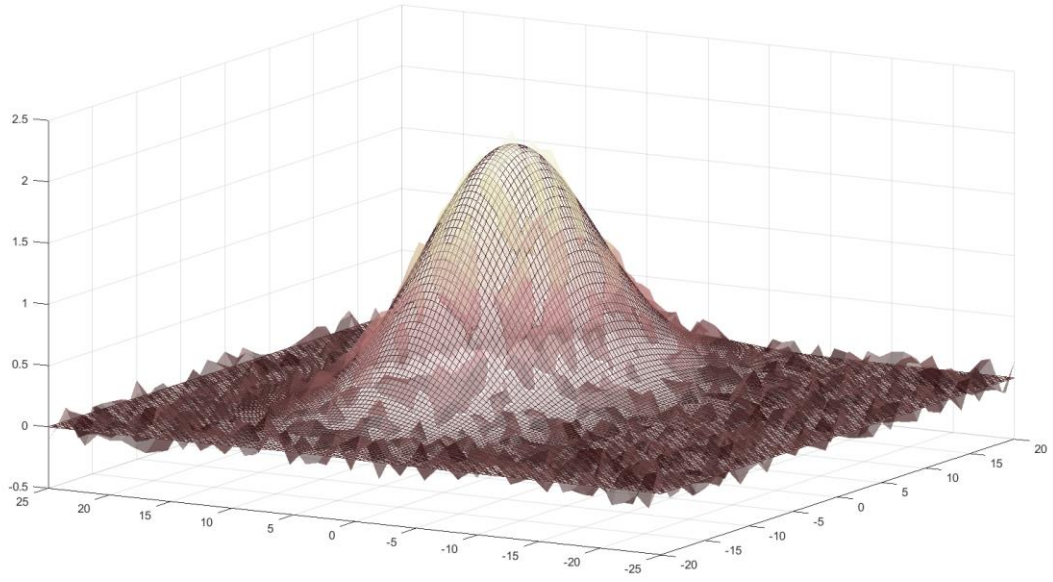
سوال چهارم)

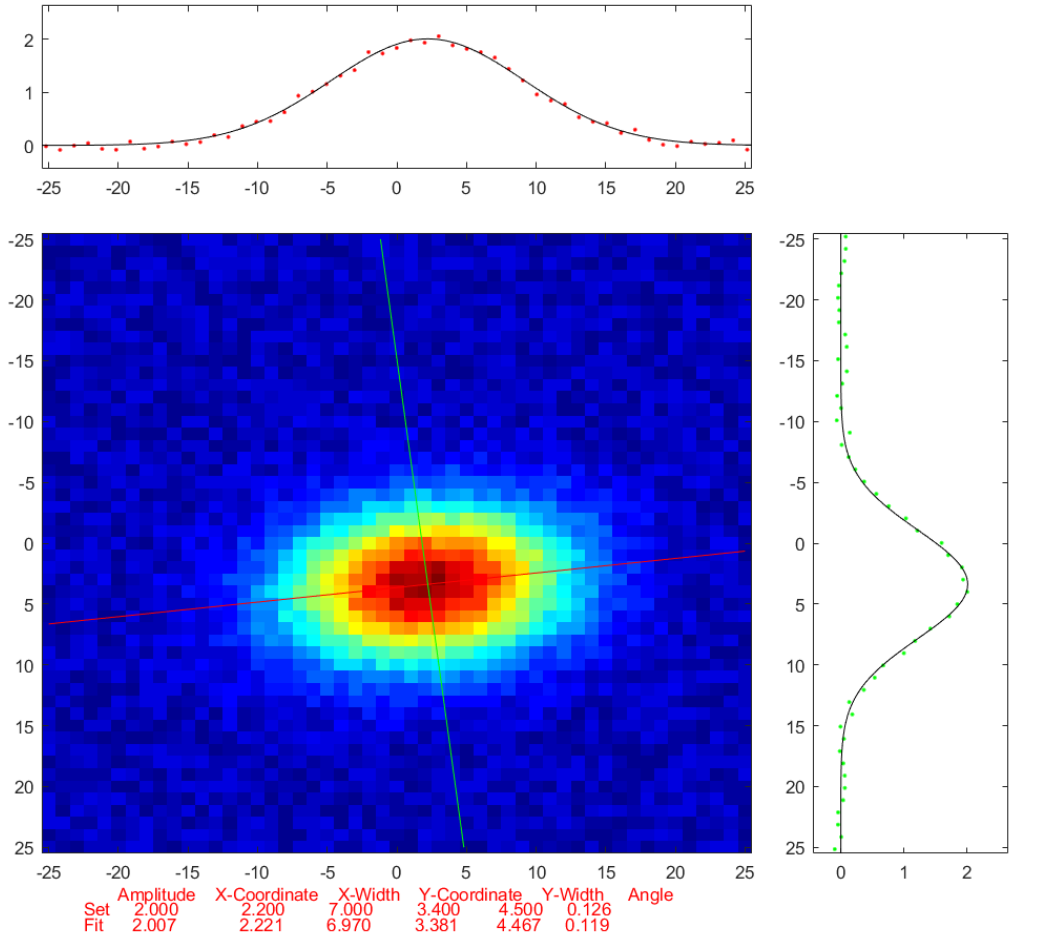
در این تمرین ابتدا تمام داده‌ها به صورت زیر ترسیم شده اند:

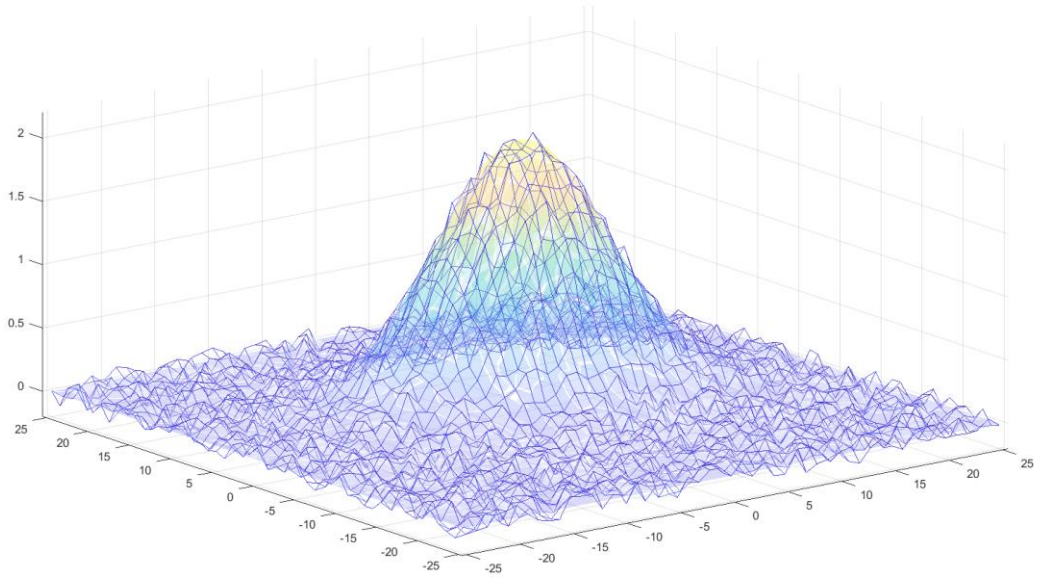


در روش طبقه‌بندی بیزین، باید داده‌ها در فضای دو بعدی، به یک توزیع نرمال fit شوند. مانند شکل‌های زیر:


Amplitude X-Coord X-Width Y-Coord Y-Width Angle
Set 2.000 2.200 7.000 3.400 4.500 0.126
Fit 2.014 2.217 6.988 3.397 4.495 0.129




Amplitude X-Coordinate X-Width Y-Coordinate Y-Width Angle
Set 2.000 2.200 7.000 3.400 4.500 0.126
Fit 2.007 2.221 6.970 3.381 4.467 0.119



همان‌طور که در شکل‌های بالا آمده است با استفاده از فرمول‌های زیر :

Two-dimensional Gaussian function [edit]

In two dimensions, the power to which *e* is raised in the Gaussian function is any negative-definite quadratic form. Consequently, the level sets of the Gaussian will always be ellipses.

A particular example of a two-dimensional Gaussian function is

$$f(x,y) = A\exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_X^2} + \frac{(y-y_0)^2}{2\sigma_Y^2}\right)\right).$$

Here the coefficient A is the amplitude, x_0, y_0 is the center, and σ_x, σ_y are the x and y spreads of the blob. The figure on the right was created using $A = 1$, $x_0 = 0$, $y_0 = 0$, $\sigma_x = \sigma_y = 1$.

The volume under the Gaussian function is given by

$$V = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,dx\,dy = 2\pi A\sigma_X\sigma_Y.$$

In general, a two-dimensional elliptical Gaussian function is expressed as

$$f(x,y) = A\exp\Big(-\big(a(x-x_0)^2 + 2b(x-x_0)(y-y_0) + c(y-y_0)^2\big)\Big),$$

where the matrix

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

is positive-definite.

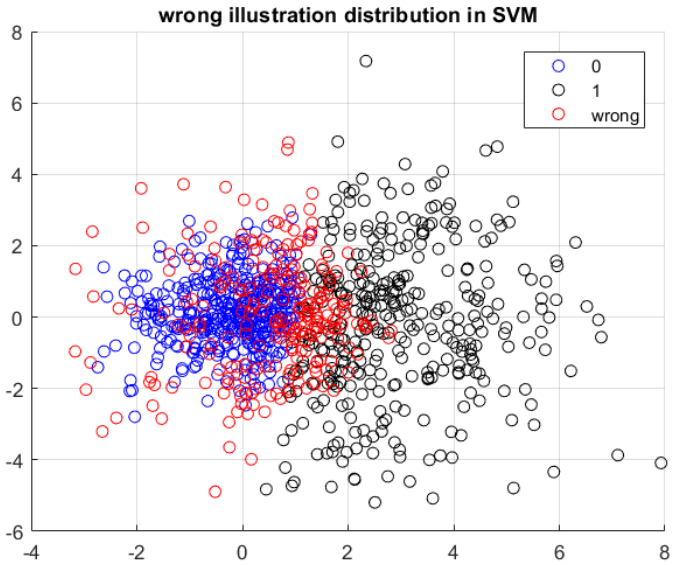
Using this formulation, the figure on the right can be created using $A = 1$, $(x_0, y_0) = (0, 0)$, $a = c = 1/2$, $b = 0$.

برای هر دسته، یک چنین تابعی fit می‌کنیم. سپس برای هر نقطه، مقدار این تابع را در هر دسته می‌یابیم و هر کدام که بیشتر بود، آن نقطه متعلق به آن دسته است.

چند سری تابع از اینترنت پیدا شد که در فایل‌ها موجود است و گاوسین ۲ بعدی به داده‌ها fitمی‌کند ولی برای توزیع دادگان به این صورت کارایی ندارند.

در صورتی که بتوان با بیزین دسته بندی را انجام داد همان‌طور که در اولین سوال اثبات کردیم، بیشترین دقت را خواهیم داشت.

اما در این جا از طبقه بندی کننده svm استفاده کردم:



conf_svm =

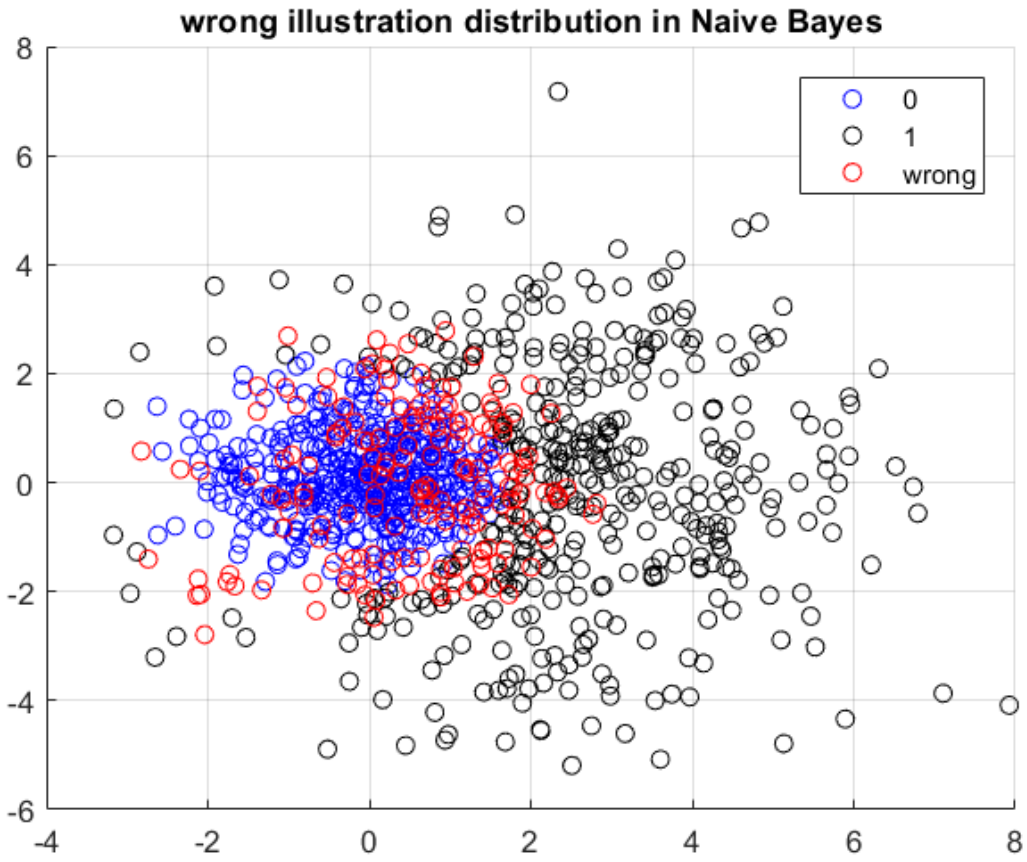
175 37

49 139

svm_percision =

78.5000

با روش نیو بیز نتایج طبقه بندی به صورت زیر است:



conf_Naive =

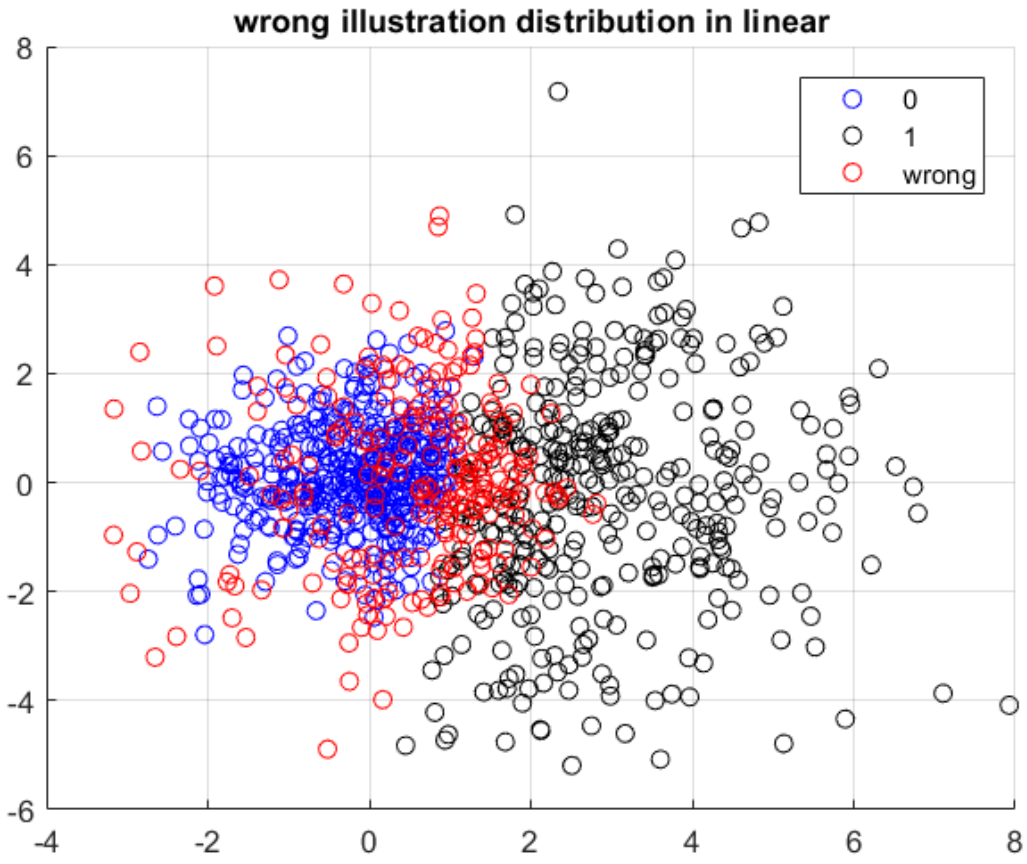
185 27

37 151

Naive_percision =

84

برای طبقه بندی کننده خطی نیز نتایج به صورت زیر است:



conf_Linear =

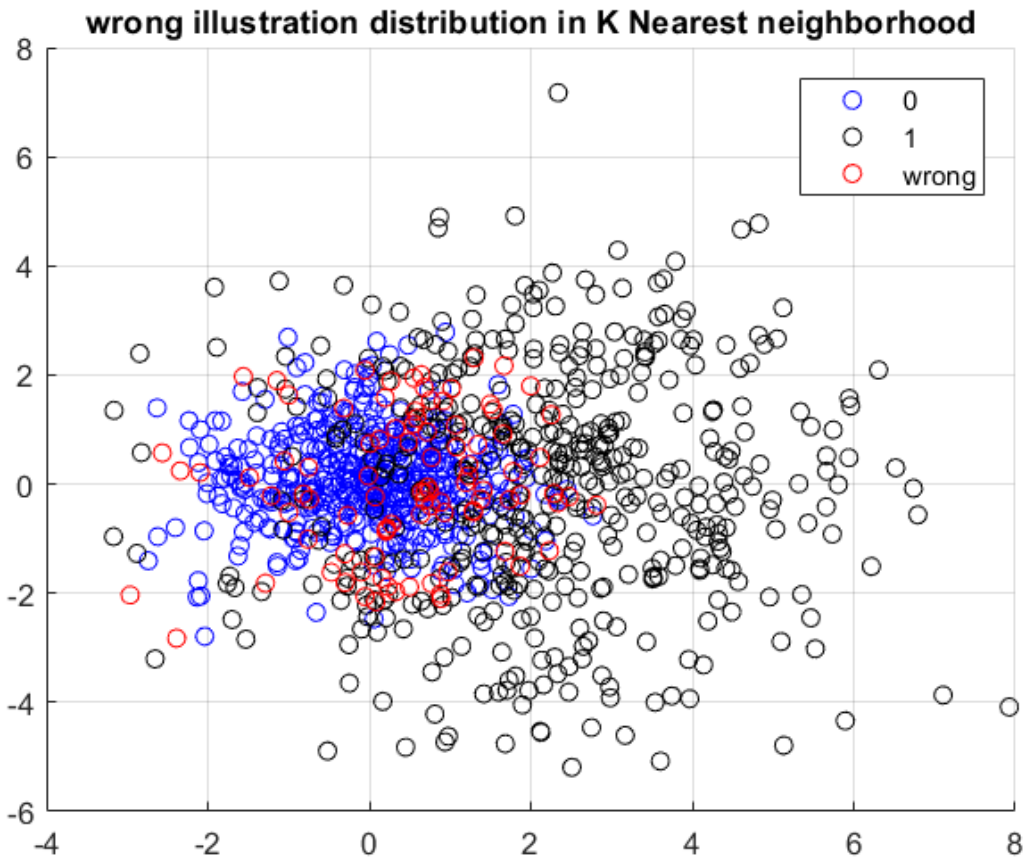
175 37

49 139

Linear_percision =

78.5000

و نتایج برای k نزدیک ترین همسایگی به صورت زیر است :



conf_K_nearest =

154 58

37 151

nearest_percision =

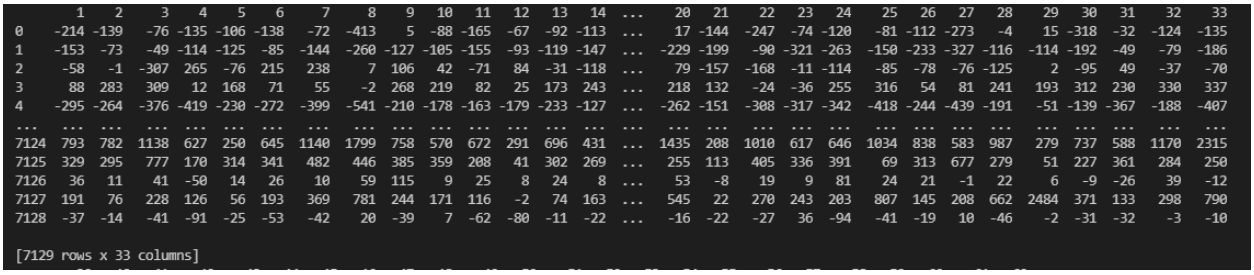
76.2500

تمرین پنجم)

برای این پس از جستجوی اینترنت، فایل نسخه CSV این دیتابیس پیدا شد. (تبدیل این دیتای موجود به دیتای قابل استفاده به سادگی نبود هرچند غیر ممکن به نظر نمیرسد.)

کد این دستور با توجه به این که داده‌های خام از رشته و اعداد تشکیل شده است با استفاده از پایتون زده شده است. با استفاده از کتابخانه pandas می‌توانیم فایل را از ورودی بخوانیم و ستون‌های مورد نیاز را از آن استخراج کنیم که داده‌های زیر به دست می‌آید(ستون‌هایی که عدد داشته اند استخراج شده اند و علت و مفهوم ستون‌های call را متوجه نشدم)

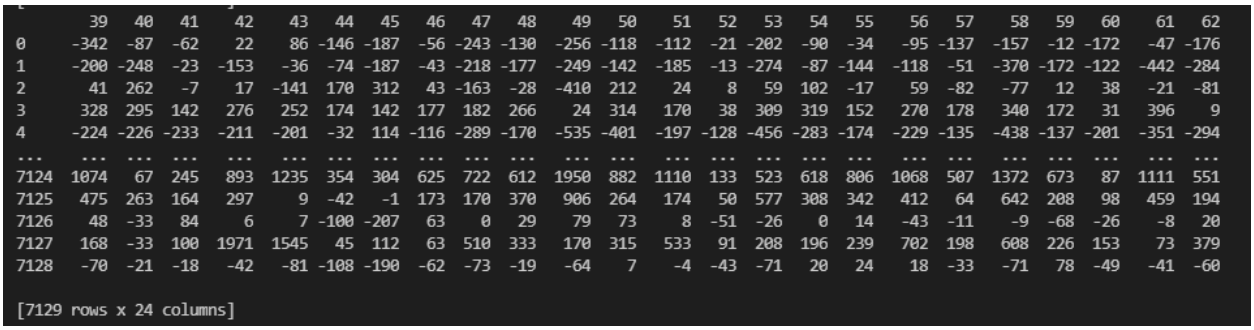
داده‌های استخراج شده از data_set_ALL_AML_train.csv



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	20	21	22	23	24	25	26	27	28	29	30	31	32	33
0	-214	-139	-76	-135	-106	-138	-72	-413	5	-88	-165	-67	-92	-113	...	17	-144	-247	-74	-120	-81	-112	-273	-4	15	-318	-32	-124	-135
1	-153	-73	-49	-114	-125	-85	-144	-260	-127	-105	-155	-93	-119	-147	...	-229	-199	-90	-321	-263	-150	-233	-327	-116	-114	-192	-49	-79	-186
2	-58	-1	-307	265	-76	215	238	7	106	42	-71	84	-31	-118	...	79	-157	-168	-11	-114	-85	-78	-76	-125	2	-95	49	-37	-70
3	88	283	309	12	168	71	55	-2	268	219	82	25	173	243	...	218	132	-24	-36	255	316	54	81	241	193	312	230	330	337
4	-295	-264	-376	-419	-230	-272	-399	-541	-210	-178	-163	-179	-233	-127	...	-262	-151	-308	-317	-342	-418	-244	-439	-191	-51	-139	-367	-188	-407
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7124	793	782	1138	627	250	645	1140	1799	758	570	672	291	696	431	...	1435	208	1010	617	646	1034	838	583	987	279	737	588	1170	2315
7125	329	295	777	170	314	341	482	446	385	359	208	41	302	269	...	255	113	405	336	391	69	313	677	279	51	227	361	284	250
7126	36	11	41	-50	14	26	10	59	115	9	25	8	24	8	...	53	-8	19	9	81	24	21	-1	22	6	-9	-26	39	-12
7127	191	76	228	126	56	193	369	781	244	171	116	-2	74	163	...	545	22	270	243	203	807	145	208	662	2484	371	133	298	790
7128	-37	-14	-41	-91	-25	-53	-42	20	-39	7	-62	-80	-11	-22	...	-16	-22	-27	36	-94	-41	-19	10	-46	-2	-31	-32	-3	-10

[7129 rows x 33 columns]

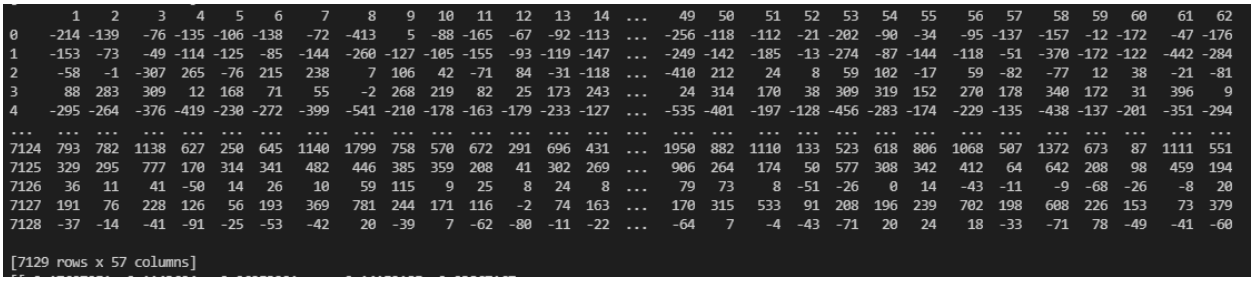
داده‌های استخراج شده از data_set_ALL_AML_independent.csv:



	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
0	-342	-87	-62	22	86	-146	-187	-56	-243	-130	-256	-118	-112	-21	-202	-90	-34	-95	-137	-157	-12	-172	-47	-176
1	-200	-248	-23	-153	-36	-74	-187	-43	-218	-177	-249	-142	-185	-13	-274	-87	-144	-118	-51	-370	-172	-122	-442	-284
2	41	262	-7	17	-141	170	312	43	-163	-28	-410	212	24	8	59	102	-17	59	-82	-77	12	38	-21	-81
3	328	295	142	276	252	174	142	177	182	266	24	314	170	38	309	319	152	270	178	340	172	31	396	9
4	-224	-226	-233	-211	-201	-32	114	-116	-289	-170	-535	-401	-197	-128	-456	-283	-174	-229	-135	-438	-137	-201	-351	-294
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7124	1074	67	245	893	1235	354	304	625	722	612	1950	882	1110	133	523	618	806	1068	507	1372	673	87	1111	551
7125	475	263	164	297	9	-42	-1	173	170	370	906	264	174	50	577	308	342	412	64	642	208	98	459	194
7126	48	-33	84	6	7	-100	-207	63	0	29	79	73	8	-51	-26	0	14	-43	-11	-9	-68	-26	-8	20
7127	168	-33	100	1971	1545	45	112	63	510	333	170	315	533	91	208	196	239	702	198	608	226	153	73	379
7128	-70	-21	-18	-42	-81	-108	-190	-62	-73	-19	-64	7	-4	-43	-71	20	24	18	-33	-71	78	-49	-41	-60

[7129 rows x 24 columns]

سپس تمام داده‌ها را با هم ترکیب می‌کنیم تا داده‌های کلی به دست بیاید:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	49	50	51	52	53	54	55	56	57	58	59	60	61	62
0	-214	-139	-76	-135	-106	-138	-72	-413	5	-88	-165	-67	-92	-113	...	-256	-118	-112	-21	-202	-90	-34	-95	-137	-157	-12	-172	-47	-176
1	-153	-73	-49	-114	-125	-85	-144	-260	-127	-105	-155	-93	-119	-147	...	-249	-142	-185	-13	-274	-87	-144	-118	-51	-370	-172	-122	-442	-284
2	-58	-1	-307	265	-76	215	238	7	106	42	-71	84	-31	-118	...	-410	212	24	8	59	102	-17	59	-82	-77	12	38	-21	-81
3	88	283	309	12	168	71	55	-2	268	219	82	25	173	243	...	24	314	170	38	309	319	152	270	178	340	172	31	396	9
4	-295	-264	-376	-419	-230	-272	-399	-541	-210	-178	-163	-179	-233	-127	...	-535	-401	-197	-128	-456	-283	-174	-229	-135	-438	-137	-201	-351	-294
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7124	793	782	1138	627	250	645	1140	1799	758	570	672	291	696	431	...	1950	882	1110	133	523	618	806	1068	507	1372	673	87	1111	551
7125	329	295	777	170	314	341	482	446	385	359	208	41	302	269	...	906	264	174	50	577	308	342	412	64	642	208	98	459	194
7126	36	11	41	-50	14	26	10	59	115	9	25	8	24	8	...	79	73	8	-51	-26	0	14	-43	-11	-9	-68	-26	-8	20
7127	191	76	228	126	56	193	369	781	244	171	116	-2	74	163	...	170	315	533	91	208	196	239	702	198	608	226	153	73	379
7128	-37	-14	-41	-91	-25	-53	-42	20	-39	7	-62	-80	-11	-22	...	-64	7	-4	-43	-71	20	24	18	-33	-71	78	-49	-41	-60

[7129 rows x 57 columns]

به نظر می‌رسد که این داده ها مربوط به ژن‌های ۵۷ نفر می‌باشد که هر فرد ۷۱۲۹ ژن دارد. با توجه به حجم بالای متغیرها امکان نمایش با Heat map وجود نداشت.

همان‌طور که در توابع دیده می‌شود، در ژن‌های مختلف، واریانس تغییرات می‌تواند با بقیه تفاوت قابل ملاحظه‌ای داشته باشد. این تغییر بسیار زیاد، ممکن است باعث اشتباه در طبقه‌بندی شود و در صورت استفاده از فاصله‌هایی مانند اقلیدسی، تغییرات این ژن‌ها بر سایر ژن‌ها غالب شود. برای رفع این مشکل، داده‌ها را نرمالیزه می‌کنیم:

```
[[-0.17607951 -0.1143694  -0.06253291 ... -0.14152185 -0.03867167
  -0.14481306]
 [-0.1116195  -0.05325636 -0.03574742 ... -0.08900378 -0.32245632
  -0.20718913]
 [-0.05851164 -0.00100882 -0.30970816 ...  0.03833521 -0.02118525
  -0.08171453]
 ...
 [ 0.09660401  0.02951789  0.11002123 ... -0.06976956 -0.02146756
   0.05366889]
 [ 0.04433878  0.01764265  0.05292796 ...  0.03551745  0.01694623
   0.08798113]
 [-0.09225728 -0.03490816 -0.10223104 ... -0.12217856 -0.10223104
  -0.1496064 ]]
```

حال این داده‌ها را به طبقه‌بندی‌کننده سلسله مراتبی مراتبی تراکمی با سه حالت ارتباط بین داده‌های تمام لینک، نزدیک ترین لینک و میانگین با دو دسته بندی و سه دسته بندی، ۵۷ نفر را به صورت‌های زیر طبقه‌بندی می‌کنیم:

برای طبقه‌بندی، می‌توانیم به صورت دستی، نیز عمل کنیم و یک فاصله تعریف کنیم (برای مثال از فاصله اقلیدسی استفاده کنیم یا...) و یا می‌توانیم از توابع آماده پایتون برای آموزش سیستم استفاده کنیم که در این سوال به روش دوم کار را انجام می‌دهیم.

```
Single linkage 2 clusters:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Single linkage 3 clusters:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Complete linkage 2 clusters:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Complete linkage 3 clusters:
[1 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1
 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1]

Average linkage 2 clusters:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

Average linkage 3 clusters:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

همان‌طور که دیده‌می‌شود ۱۷ امین و ۲۰ امین فرد، در بیشتر دسته بندی‌ها از سایر تفاوت بیشتری دارد. در complete link، به نظر می‌رسد کمتر از باقی، دچار نویز و داده پرت شده است.

برای طبقه‌بندی KMean نیز مانند بالا عمل می‌کنیم. در زیر می‌توانیم دسته بندی ۲ تایی و ۳ تایی را همراه با نقاط مرکز به دست آمده را ملاحظه کنیم.

```
KMeans 2 clusters:
[0 1 0 0 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 0 0 1 0 0 0 0 0 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 0 1 1 0 1]
KMeans 2 cluster centers:
[[-0.14495615 -0.1423553  -0.00443004 ...  0.04153506  0.07500147
  -0.08293398]
 [-0.07240653 -0.09711454 -0.01100802 ...  0.01168088  0.08731885
  -0.08624369]]

KMeans 3 clusters:
[0 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 2 1 1 1 1 1 1 0 0 1 0 0 0 0 1 1 1 1
 1 1 1 1 1 1 0 0 0 1 0 1 1 1 1 1 1 1 0 1]
KMeans 3 cluster centers:
[[-0.13602207 -0.1295508  -0.01369873 ...  0.05960072  0.07699723
  -0.08031501]
 [-0.08717248 -0.10669018 -0.00798877 ...  0.00210324  0.08390299
  -0.08848342]
 [ 0.01398762 -0.16706448  0.07969689 ...  0.14222257  0.1265164
  -0.03989504]]
```

برای این که در این روش، به بهینه سراسری برسیم، باید نقاط شروع را مختلف انتخاب کنیم و بهینه‌ترین همه حالات را در نظر می‌گیریم. در این تمرین از ۱۰ سری نقطه اولیه شروع می‌کنیم.

با تشکر.