به نام خدا



دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس شناسایی آماری الگو

استاد رحمتی

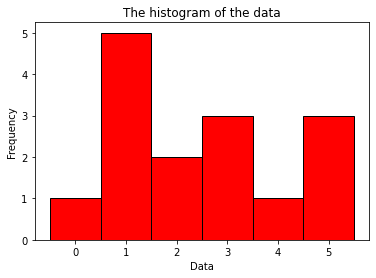
تمرین سوم

علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

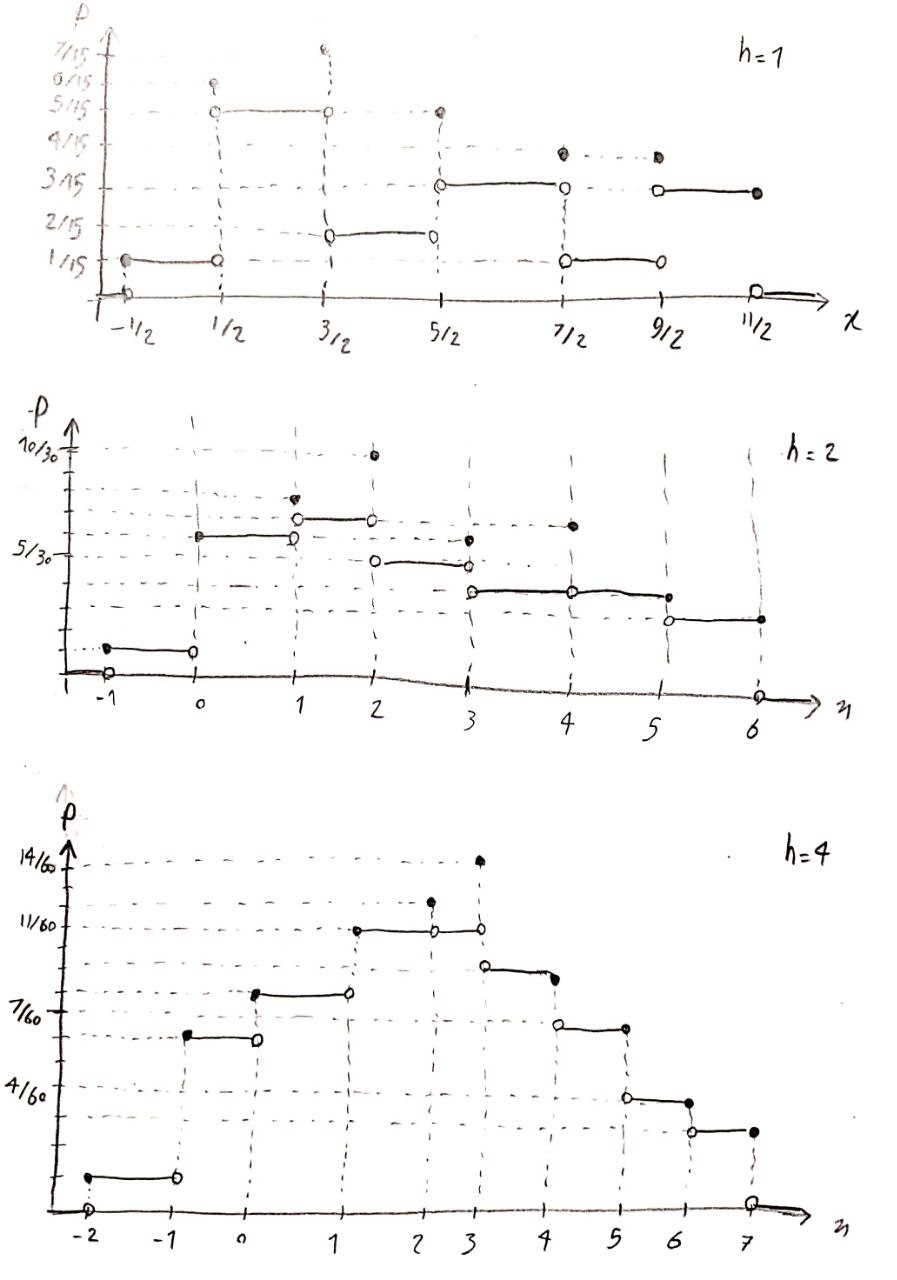
سوال ۱

a)

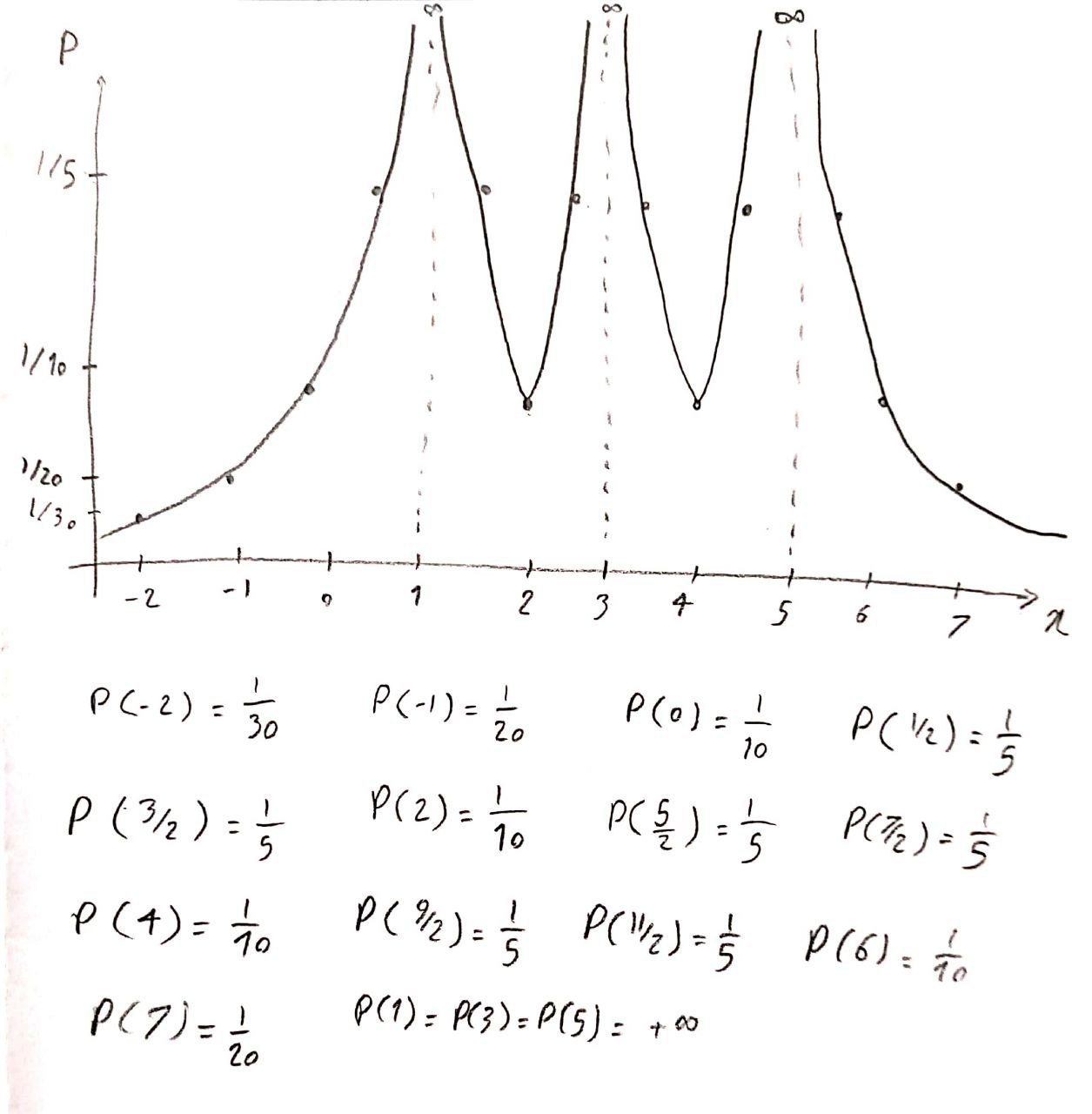


b)

توجه کنید که برای این قسمت از کرنل زیر استفاده کرده‌ام که در اسلاید‌ها آماده است و برخلاف کرنلی که در قسمت ه آمده است، حالت تساوی هم وجود دارد:



c)



d)

e*)*

f)

سوال ۲

a)

b)

c)

باتوجه به ماتریس کواریانس می‌توان گفت که پراکندگی ویژگی دوم داده‌ها نسبت به دو ویژگی دیگر بیشتر است. همچنین همبستگی بیشتری بین دو ویژگی اول و دوم دیده می‌شود درحالی که همبستگی بین ویژگی دوم و سوم نسبتا کم است و نهایتا می‌توان دید که هر سه ویژگی با یکدیگر همبستگی دارند و در یک راستا هستند ولی چون تمامی مقادیر کواریانس از مقادیر واریانس کمتر است این همبستگی در کل زیاد نیست.

d)

e) سه بردار ویژه‌ای که در قسمت قبل پیدا شده است یعنی ، و می‌توانند به عنوان پایه جدید مورد استفاده قرار بگیرند. فقط پیش از آن باید نرمال‌سازی شوند:

f) ماتریس حاصل از سه بردار ویژه برای تصویرکردن داده‌ها می‌تواند استفاده شود. چنانچه قصد داشته باشیم ابعاد را هم کاهش دهیم می‌توان سطر سوم ماتریس (کاهش یک بعد) یا دو سطر دوم (کاهش دو بعد) را حذف کرد.

g)

h) مشابه قسمت ج کواریانس را می‌توان حساب کرد:

i)

j)

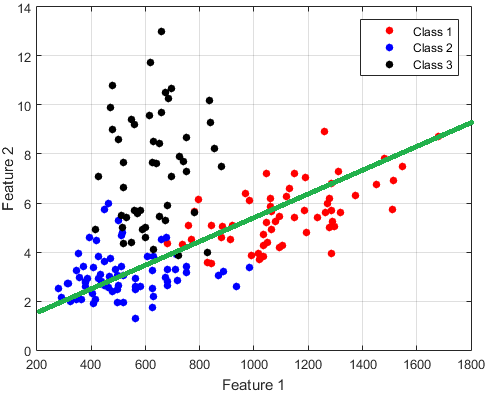
k)

مقادیر ویژه و بردار‌های ویژه ماتریس فوق را می‌یابیم:

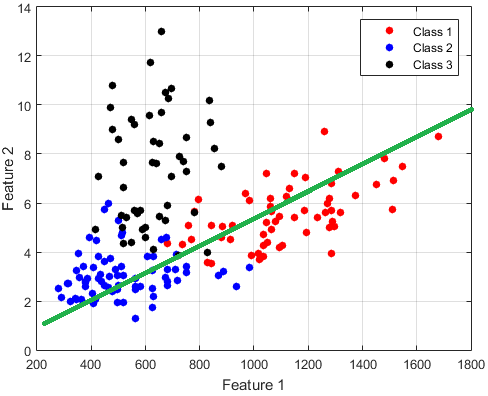
با ضرب کردن W‌ در هر داده‌ای () تصویر‌شده داده جدید حاصل می‌شود. برای کاهش ابعاد می‌توان سطر سوم (کاهش یک بعد) یا سطر دوم و سوم (کاهش دو بعد) را از W‌ حذف کرد.

l)

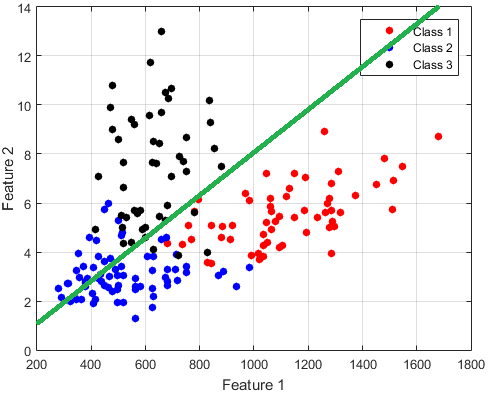
m)



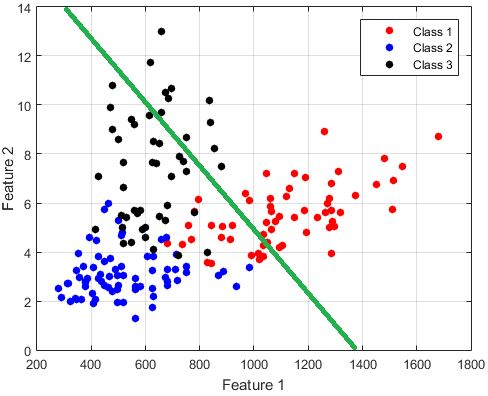
n)



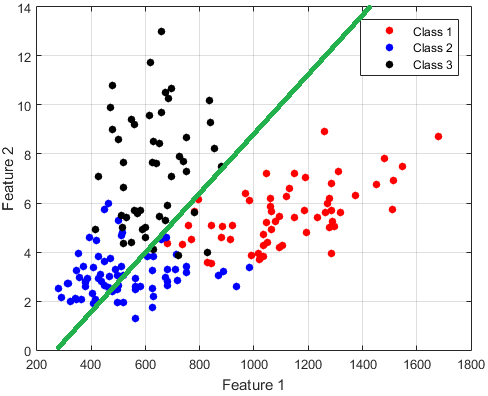
o)



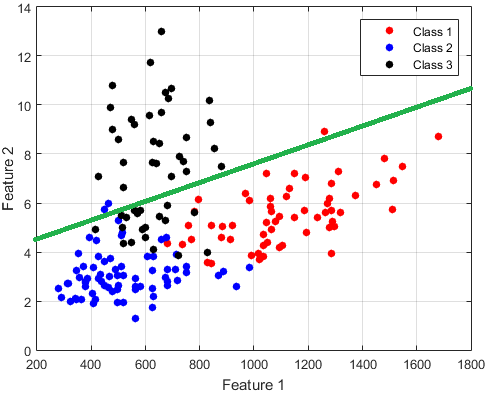
p)



q)



r)



سوال ۳

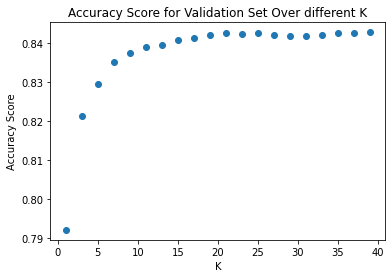
a) قبل از هرچیز ویژگی‌های مربوط به جهت باد و آمدن یا نیامدن باران را عددی کرده‌ام. در کنار حذف این دو، ویژگی‌های Date و Location را به دلیل عدم تبدیل مناسب به ویژگی‌های عددی حذف می‌کنم. ویژگی month‌ را نیز مطابق پیشنهاد سوال اضافه کرده‌ام.

دو جفت ویژگی (Temp3pm, MaxTemp) و (Pressure9am, Pressure3pm) دارای همبستگی بالایی دارند. به دلخواه ویژگی‌های MaxTemp و Pressure9am‌ را حذف می‌کنم.

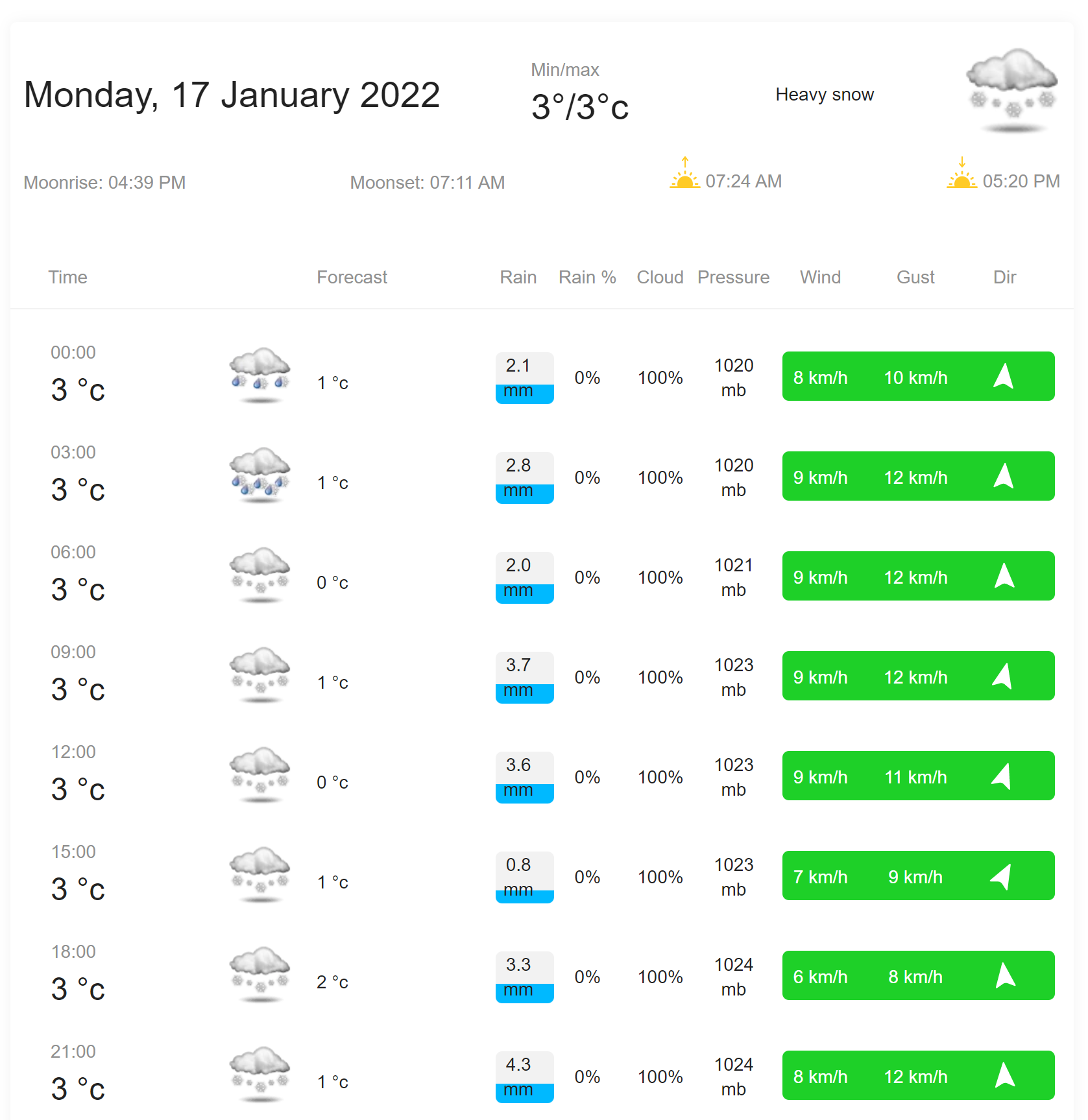
b) برای تابع هدف 11-NN را انتخاب کردم و ویژگی‌های زیر انتخاب شدند:

'Rainfall', 'Sunshine', 'WindGustSpeed', 'WindDir3pm', 'WindSpeed3pm', ' Humidity3pm', 'Pressure3pm', 'Temp9am', 'Temp3pm', 'Month'

c) در تصویر زیر نمودار صحت‌های مختلف روی مجموعه اعتبارسنجی به ازای k های مختلف آورده شده است. به نظر برای kهای بزرگ‌تر از 20 به صحت مناسب می‌توان رسید ولی بیشترین مقدار به 39-NN تعلق داشت و من هم از این مدل استفاده کرده‌ام. صحت برای مجموعه تست برابر با 84.59٪ شد.



d) نتایج آب و هوای شهر رشت برای روز دوشنبه 2۷ دی ۱۴۰۰ از سایت معرفی‌شده به شرح زیر است:



باتوجه به این آمار ویژگی‌های داده به صورت زیر درنظر گرفته شده است. برای بیشتر موارد مقدار ویژگی از آمار قابل حصول بود. برای برخی از ویژگی‌ها به این شکل مقدار حاصل شده است: ( و شاید صحیح نباشد!) برای ویژگی‌ Rainfall‌ مجموع مقدار باران در بازه‌های زمانی درنظر گرفته شده است؛ برای ویژگی WindGustDir و WindGustSpeed به نوعی میانگین‌گیری انجام شده است؛ نهایتا برف و باران مشابه درنظر گرفته شده است. چون به نظر می‌رسد در روز دوشنبه برف باریده است ولی در ویژگی‌ها به عنوان باران درنظر گرفته شده است.

|  |  |  |  |
| --- | --- | --- | --- |
| مقدار | ویژگی | مقدار | ویژگی |
| 1 | Month | 2022-01-17 | Date |
| 3 | MaxTemp | 3 | MinTemp |
| 3 | Temp3pm | 3 | Temp9am |
| 22.6 | Rainfall | Yes | RainToday |
| 10.75 | WindGustSpeed | NNE | WindGustDir |
| NE | WindDir3pm | N | WindDir9am |
| 7 | WindSpeed3pm | 9 | WindSpeed9am |
| None | Sunshine | None | Evaporation |
| None | Humidity3pm | None | Humidity9am |
| 1023 | Pressure3pm | 1023 | Pressure9am |
| None | Cloud3pm | None | Cloud9am |
|  |  | Rasht | Location |

نهایتا با دادن داده روز دوشنبه مدل پیش‌بینی کرد که فردا یعنی سه‌شنبه باران نخواهد آمد.

e) اگر منظور سوال این است که آیا می‌توان در کنار پیش‌بینی بارانی بودن یا نبودن میزان احتمال آن را بیان کرد، باید گفت بله قابل اعمال است. با استراتژی فعلی برای آن که پیشبینی کنیم که یک داده متعلق به چه کلاسی است بررسی می‌کنیم که نزدیک‌ترین همسایه‌ها بیشتر به چه کلاسی تعلق دارند؛ طبیعتا می‌توان تعداد داده‌های نزدیک هر کلاس را به تعداد کل داده‌های نزدیک (پارامتر k) تقسیم کرد و احتمال آنکه داده تست به آن کلاس تعلق داشته باشد را حساب کرد.

اگر منظور از rainfall‌ یکی از ویژگی‌ها مجموعه‌داده به معنای میزان بارش باشد، باز هم امکان‌پذیر است. پیش‌بینی آمدن باران یا نیامدن آن به شکل مسئله‌ای دو کلاسه است. اما پیش‌بینی میزان باران از نوع رگرسیون است. لذا همین استراتژی عینا قابل اعمال نیست ولی با استفاده از روش‌هایی مانند Regression KNN می‌توان مسئله رگرسیون را حل کرد. به این شکل مثلا نزدیک‌ترین داده‌های یک داده تست را درنظر بگیریم و ویژگی هدف آن‌ها را میانگین بگیریم و به عنوان پیش‌بینی اعلام کنیم.

سوال 4

b) ۲۰ مقادیر ویژه بزرگ‌تر و بردارهای ویژه متناسب در این قسمت آورده شده است!

Eigen Value = 174662.95 Eigen Vector =

[-0.12 -0.12 -0.12 -0.12 -0.12 -0.12 -0.12 -0.12 -0.12 -0.12 -0.13 -0.13

-0.13 -0.12 -0.12 -0.12 -0.12 -0.13 -0.13 -0.13 -0.13 -0.13 -0.13 -0.12

-0.12 -0.13 -0.13 -0.13 -0.13 -0.13 -0.13 -0.12 -0.12 -0.13 -0.13 -0.13

-0.13 -0.13 -0.13 -0.12 -0.12 -0.13 -0.13 -0.13 -0.13 -0.13 -0.13 -0.12

-0.12 -0.12 -0.13 -0.13 -0.13 -0.13 -0.13 -0.12 -0.12 -0.12 -0.13 -0.13

-0.13 -0.13 -0.12 -0.12]

Eigen Value = 4224.48 Eigen Vector =

[ 0.1 0.06 0.01 -0.05 -0.1 -0.14 -0.17 -0.2 0.12 0.08 0.03 -0.03

-0.09 -0.14 -0.17 -0.2 0.15 0.11 0.05 -0.01 -0.07 -0.12 -0.17 -0.19

0.18 0.13 0.08 0.01 -0.05 -0.11 -0.15 -0.18 0.2 0.16 0.1 0.04

-0.03 -0.09 -0.13 -0.16 0.21 0.18 0.12 0.06 -0. -0.06 -0.11 -0.15

0.21 0.19 0.14 0.08 0.02 -0.04 -0.09 -0.13 0.21 0.19 0.15 0.1

0.04 -0.01 -0.06 -0.1 ]

Eigen Value = 3226.94 Eigen Vector =

[-0.22 -0.22 -0.21 -0.19 -0.17 -0.14 -0.11 -0.09 -0.19 -0.2 -0.18 -0.16

-0.14 -0.11 -0.08 -0.06 -0.15 -0.15 -0.13 -0.11 -0.08 -0.05 -0.03 -0.01

-0.09 -0.09 -0.07 -0.04 -0.01 0.01 0.03 0.04 -0.04 -0.02 0.01 0.03

0.06 0.07 0.08 0.09 0.01 0.04 0.07 0.09 0.11 0.12 0.13 0.13

0.06 0.09 0.12 0.14 0.16 0.17 0.17 0.17 0.1 0.12 0.15 0.17

0.18 0.19 0.19 0.19]

Eigen Value = 737.01 Eigen Vector =

[-0.01 0.06 0.11 0.09 0.03 -0.07 -0.2 -0.29 -0.03 0.07 0.14 0.13

0.08 -0.04 -0.18 -0.27 -0.06 0.05 0.15 0.16 0.12 0.02 -0.11 -0.22

-0.11 0.01 0.13 0.17 0.14 0.06 -0.05 -0.15 -0.16 -0.06 0.08 0.15

0.16 0.1 0. -0.1 -0.2 -0.11 0.03 0.13 0.15 0.11 0.04 -0.06

-0.23 -0.16 -0.03 0.09 0.14 0.12 0.06 -0.02 -0.24 -0.18 -0.06 0.04

0.1 0.1 0.05 -0.01]

Eigen Value = 579.43 Eigen Vector =

[ 0.26 0.2 0.1 0.01 -0.05 -0.08 -0.09 -0.07 0.24 0.19 0.07 -0.03

-0.1 -0.13 -0.12 -0.08 0.19 0.12 0.01 -0.08 -0.13 -0.13 -0.09 -0.03

0.11 0.04 -0.06 -0.13 -0.16 -0.11 -0.03 0.03 0.04 -0.04 -0.14 -0.17

-0.13 -0.04 0.05 0.1 -0.01 -0.09 -0.15 -0.14 -0.07 0.03 0.13 0.19

-0.04 -0.11 -0.14 -0.1 -0.02 0.1 0.2 0.28 -0.03 -0.06 -0.08 -0.06

0.02 0.13 0.23 0.3 ]

Eigen Value = 464.32 Eigen Vector =

[-0.04 -0.1 -0.18 -0.21 -0.22 -0.21 -0.17 -0.12 0.03 -0.02 -0.08 -0.11

-0.12 -0.1 -0.07 -0.03 0.12 0.09 0.06 0.03 0.03 0.05 0.08 0.08

0.17 0.17 0.14 0.13 0.13 0.15 0.18 0.17 0.15 0.15 0.12 0.11

0.13 0.14 0.17 0.17 0.09 0.08 0.03 0.02 0.03 0.05 0.09 0.11

-0.01 -0.05 -0.1 -0.13 -0.12 -0.08 -0.03 0.03 -0.1 -0.15 -0.2 -0.24

-0.23 -0.18 -0.12 -0.05]

Eigen Value = 276.82 Eigen Vector =

[ 0.04 0. 0.04 0.12 0.13 0.01 -0.18 -0.32 0. -0.04 -0.01 0.12

0.2 0.11 -0.08 -0.25 -0.04 -0.12 -0.1 0.03 0.18 0.18 0.05 -0.12

-0.06 -0.16 -0.18 -0.05 0.13 0.2 0.12 -0.01 0.01 -0.12 -0.2 -0.14

0.04 0.15 0.14 0.05 0.1 -0.02 -0.14 -0.16 -0.04 0.06 0.08 0.04

0.22 0.11 -0.05 -0.15 -0.11 -0.02 0.01 -0.01 0.3 0.2 0.03 -0.1

-0.11 -0.05 -0.02 -0.05]

Eigen Value = 237.25 Eigen Vector =

[ 0.29 0.27 0.12 -0.08 -0.18 -0.16 -0.09 -0. 0.19 0.17 0.04 -0.11

-0.17 -0.11 -0.01 0.05 -0.01 -0.02 -0.05 -0.09 -0.08 -0.01 0.08 0.13

-0.18 -0.15 -0.1 -0.05 -0.02 0.05 0.14 0.19 -0.23 -0.17 -0.05 0.02

0.05 0.08 0.12 0.17 -0.15 -0.07 0.04 0.09 0.07 0.05 0.04 0.03

-0.05 0.05 0.13 0.14 0.07 -0.03 -0.12 -0.15 0.02 0.09 0.15 0.14

0.04 -0.11 -0.23 -0.28]

Eigen Value = 183.86 Eigen Vector =

[ 0.14 0.07 0.03 0.09 0.19 0.22 0.17 0.08 0.08 -0.06 -0.17 -0.14

-0.03 0.04 0.01 -0.06 0.03 -0.13 -0.26 -0.25 -0.16 -0.09 -0.12 -0.17

0.08 -0.04 -0.14 -0.14 -0.06 -0.01 -0.04 -0.14 0.14 0.09 0.03 0.05

0.17 0.21 0.09 -0.03 0.13 0.09 0.05 0.11 0.24 0.27 0.15 -0.05

0.01 -0.03 -0.06 0. 0.12 0.15 0.06 -0.11 -0.1 -0.14 -0.17 -0.14

-0.06 -0.02 -0.07 -0.19]

Eigen Value = 162.61 Eigen Vector =

[-0.07 0.11 0.24 0.22 0.11 0.03 0.02 0.03 -0.18 -0.02 0.11 0.09

-0.01 -0.11 -0.11 -0.07 -0.24 -0.11 0.02 -0.01 -0.12 -0.2 -0.17 -0.07

-0.2 -0.06 0.07 0.04 -0.07 -0.14 -0.07 0.07 -0.07 0.08 0.17 0.12

0. -0.04 0.06 0.21 0.03 0.14 0.19 0.11 -0.03 -0.05 0.09 0.25

0.04 0.11 0.11 -0.01 -0.17 -0.17 0.01 0.22 0. 0.04 0.01 -0.11

-0.25 -0.23 -0.08 0.13]

Eigen Value = 124.13 Eigen Vector =

[ 0.19 0.1 -0.08 -0.18 -0.13 -0. 0.1 0.19 0.13 0.07 -0.08 -0.16

-0.1 0. 0.08 0.11 0. -0.02 -0.02 0.02 0.06 0.03 -0.03 -0.1

-0.14 -0.12 0.03 0.19 0.22 0.06 -0.16 -0.24 -0.2 -0.13 0.05 0.22

0.22 0.04 -0.14 -0.23 -0.1 -0.07 -0.02 0.05 0.07 0.03 -0.02 -0.08

0.09 0.05 -0.05 -0.12 -0.12 -0.03 0.07 0.14 0.24 0.15 -0.04 -0.17

-0.19 -0.07 0.09 0.27]

Eigen Value = 96.52 Eigen Vector =

[-0.19 -0.17 -0.06 0.02 0.02 0.06 0.14 0.18 -0.08 -0.01 0.09 0.08

-0.06 -0.14 -0.1 0.06 0.08 0.15 0.23 0.14 -0.08 -0.23 -0.21 -0.01

0.07 0.08 0.16 0.12 -0.04 -0.12 -0.09 0.01 -0.08 -0.16 -0.14 -0.04

0.07 0.11 0.06 0.07 -0.06 -0.2 -0.24 -0.09 0.09 0.18 0.17 0.1

0.14 -0.04 -0.14 -0.06 0.07 0.11 0.05 -0.04 0.25 0.14 0.04 0.02

0.02 -0.06 -0.2 -0.26]

Eigen Value = 85.85 Eigen Vector =

[-0.02 0.12 0.25 0.16 -0.1 -0.24 -0.13 0.08 -0.13 -0.01 0.16 0.11

-0.12 -0.19 -0.03 0.21 -0.21 -0.16 -0. 0.04 -0.06 -0.08 0.04 0.21

-0.11 -0.11 -0.01 0.06 0.02 -0.01 0.05 0.11 0.14 0.04 -0.01 0.07

0.09 0.03 -0.04 -0.09 0.22 0.04 -0.07 -0.02 0.06 0.01 -0.12 -0.21

0.2 -0.05 -0.2 -0.11 0.08 0.12 -0.04 -0.16 0.14 -0.1 -0.24 -0.13

0.12 0.21 0.14 0.01]

Eigen Value = 79.28 Eigen Vector =

[-0.05 -0.06 -0.1 -0.14 -0.17 -0.18 -0.12 -0.05 0.06 0.11 0.15 0.18

0.16 0.14 0.16 0.14 -0.01 0.03 0.1 0.15 0.12 0.09 0.1 0.08

-0.12 -0.14 -0.15 -0.13 -0.17 -0.22 -0.22 -0.17 0.02 -0.05 -0.09 -0.08

-0.1 -0.13 -0.1 -0.06 0.12 0.13 0.11 0.12 0.14 0.17 0.2 0.12

0.07 0.11 0.08 0.06 0.11 0.15 0.16 0.07 -0.07 -0.09 -0.13 -0.15

-0.13 -0.11 -0.12 -0.12]

Eigen Value = 60.25 Eigen Vector =

[ 0.1 -0.09 -0.04 0.07 0.03 -0.08 -0.05 0.13 0.12 -0.15 -0.1 0.13

0.11 -0.08 -0.13 0.03 0.18 -0.16 -0.14 0.14 0.15 -0.05 -0.15 0.

0.21 -0.17 -0.2 0.09 0.15 -0.02 -0.12 0.08 0.16 -0.16 -0.17 0.11

0.11 -0.11 -0.09 0.14 0.13 -0.11 -0.08 0.18 0.12 -0.15 -0.15 0.16

0.05 -0.11 -0.03 0.22 0.13 -0.16 -0.15 0.14 -0.03 -0.13 -0.06 0.12

0.07 -0.13 -0.12 0.12]

Eigen Value = 56.09 Eigen Vector =

[ 0.29 0.14 -0.05 -0.08 0.05 0.08 -0.07 -0.2 -0.03 -0.16 -0.2 -0.08

0.09 0.16 0.1 -0.08 -0.17 -0.18 -0.07 0.03 0.02 0.04 0.11 0.08

-0.01 0.08 0.22 0.17 -0.09 -0.2 -0.07 0.09 0.04 0.13 0.22 0.13

-0.13 -0.25 -0.14 0.08 -0.08 -0.15 -0.09 0.01 -0.02 -0.04 0.05 0.18

0.02 -0.15 -0.19 -0.01 0.13 0.12 0.06 0.04 0.25 0.06 -0.1 -0.02

0.09 0.02 -0.11 -0.16]

Eigen Value = 53.04 Eigen Vector =

[-0.22 -0.15 0.05 0.17 0.15 0.02 -0.13 -0.29 0.03 0.04 0.06 0.01

0. 0.08 0.12 0.01 0.18 0.11 -0.03 -0.18 -0.19 0. 0.16 0.2

0.15 0.1 -0.03 -0.17 -0.22 -0.09 0.05 0.11 -0.15 -0.15 0.04 0.17

0.12 -0.01 -0.07 -0.06 -0.15 -0.17 0.05 0.25 0.2 0.02 -0.09 -0.12

0.04 -0.06 -0.04 0.06 0.04 -0.07 -0.1 0.01 0.22 0.08 -0.07 -0.11

-0.12 -0.08 0.02 0.2 ]

Eigen Value = 51.8 Eigen Vector =

[-0.01 0.12 0.21 0.13 -0.09 -0.21 -0.14 0.05 -0.18 -0.12 0.02 0.04

-0.05 -0.04 0.07 0.18 -0.1 -0.13 -0.11 -0.07 -0.01 0.1 0.15 0.06

0.26 0.16 -0. -0.05 0.03 0.06 -0.09 -0.26 0.24 0.13 -0.05 -0.05

0.08 0.1 -0.05 -0.19 -0.12 -0.13 -0.13 -0.09 0.04 0.13 0.12 0.06

-0.23 -0.1 0.01 -0.01 -0.05 -0.02 0.05 0.13 -0.04 0.15 0.28 0.15

-0.09 -0.19 -0.13 0.02]

Eigen Value = 43.44 Eigen Vector =

[ 0.03 -0.12 -0.11 0.16 0.14 -0.18 -0.24 0.15 0.18 -0.04 -0.14 0.14

0.23 -0.07 -0.16 0.19 0.14 -0.04 -0.19 0. 0.1 -0.08 -0.13 0.14

0.08 0.02 -0.11 -0.01 0.03 -0.1 -0.13 0.08 -0.18 0.1 0.21 0.07

0.05 0.03 -0.06 0.09 -0.22 0.07 0.12 -0.08 -0.06 0.05 0.01 0.02

-0.16 0.11 0.05 -0.23 -0.14 0.1 0.05 -0.12 -0.04 0.17 0.08 -0.15

-0. 0.21 0.09 -0.16]

Eigen Value = 41.87 Eigen Vector =

[-0.07 -0.05 -0.1 -0.14 -0.11 -0.05 -0.01 -0.01 0.1 0.17 0.16 0.1

0.12 0.18 0.17 0.04 -0.04 -0.01 -0.01 -0.04 -0.04 -0.02 -0.05 -0.14

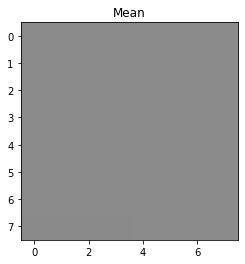
-0.15 -0.1 -0.08 -0.1 -0.12 -0.12 -0.16 -0.16 0.14 0.13 0.16 0.18

0.14 0.16 0.28 0.21 0.08 -0.04 -0.04 0.01 -0.02 -0.04 0.03 0.04

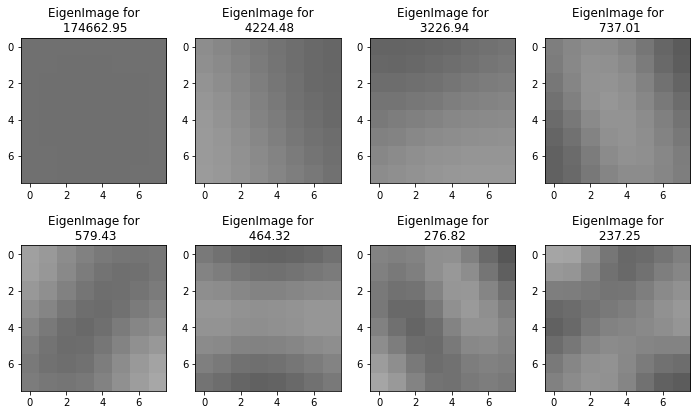
-0.1 -0.25 -0.23 -0.11 -0.15 -0.23 -0.2 -0.1 0.12 0.05 0.08 0.18

0.18 0.08 0.04 0.04]

میانگین:



تصاویر متناسب با ۸ بردار‌های ویژه اول:



e) تصاویر بازسازی‌شده به همراه تصویر اصلی ( که حاصل حفظ تمام ۶۴ مولفه اصلی است در اینجا آورده شده است:



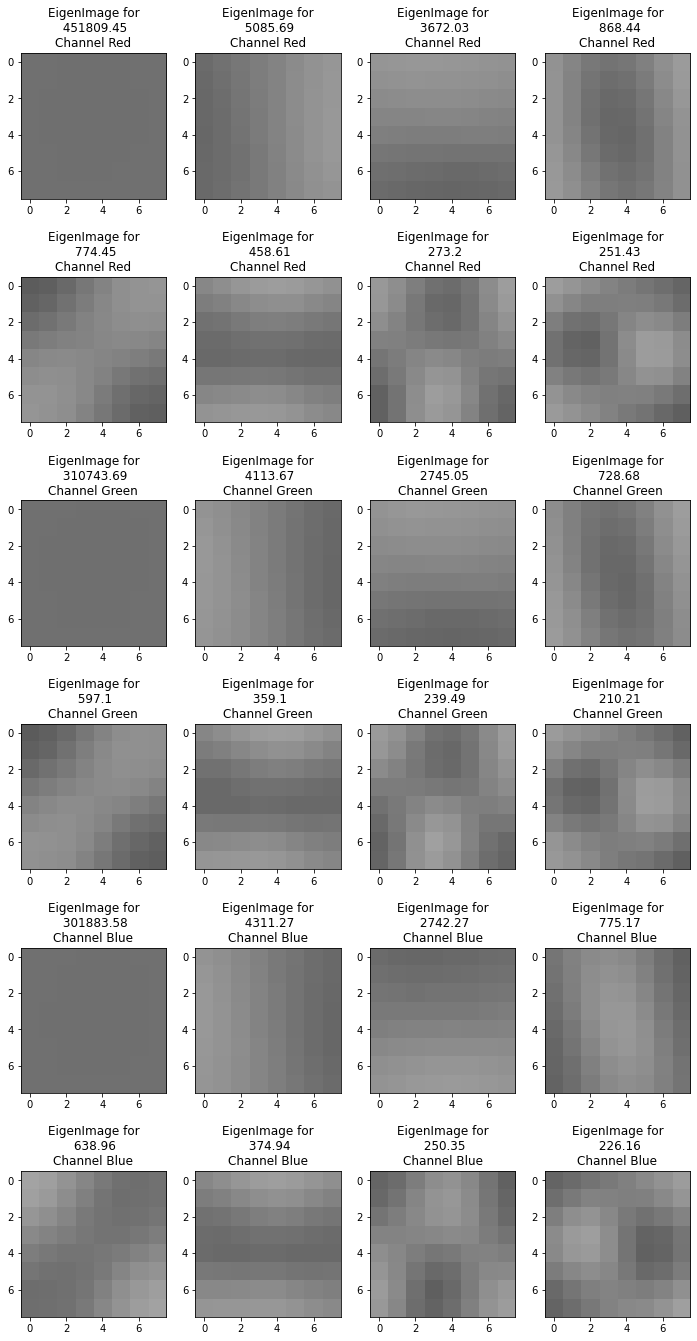
مطابق انتظار با افزایش k کیفیت تصویر بهبود پیدا می‌کند به گونه‌ای که بازسازی تصویر با دو مولفه اصلی یک تصویر کم‌کیفیت ایجاد می‌کند ولی حفظ تمام ۶۴ مولفه، تصویر اصلی را با کیفیت اولیه ایجاد می‌کند. در عین حال می‌توان دید که مولفه‌های اصلی یک تصویر بسیار تاثیرگذارند. به گونه‌ای که تنها با دو مولفه مهم‌تر می‌توان کلیات تصویر اصلی را حفظ کرد و با داشتن ۱۰ مولفه اصلی می‌توان تصویر را با کیفیتی بسیار شبیه به کیفیت اولیه حفظ کرد اما با حافظه‌ای بسیار کمتر!

f) این قسمت مانند قسمت‌های قبل است با دو تفاوت اصلی؛ اول آنکه تصویر این قسمت رنگی است و برای مدیریت آن، تصویر را به سه کانال قرمز، سبز و آبی شکستم و برای هر کانال جداگانه فرآيند قسمت‌های قبل را انجام می‌دهیم و نهایتا سه کانال را برای نمایش نهایی به هم متصل می‌کنیم. تفاوت دوم این بود که ابعاد تصویر بر ۸ بخش‌پذیر نبود؛ برای حل این مشکل تعدادی از سطر‌ها و ستون‌ها را حذف کردیم تا بخش‌پذیری برقرار شود.

میانگین:



بردار‌های ویژه مربوط به ۲۰ مقدار ویژه بزرگ‌تر هر سه کانال در فایل کد موجود است و باتوجه به کم اهمیت بودن و حجیم بودن آن در اینجا آورده نمی‌شود. تصاویر متناسب با ۸ بردارهای ویژه اول سه کانال در این قسمت آورده شده است:



تصاویر بازسازی شده:

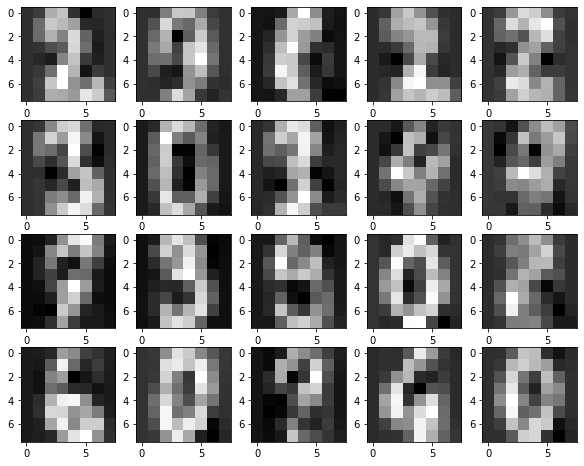
در اینجا هم مانند قبل و مطابق انتظار با افزایش k‌ کیفیت تصویر بهتر می‌شود و با داشتن چند مولفه اول می‌توان تصویر را تا حد زیادی بازسازی کرد. شاید یک تفاوت که نسبت به حالت قبل به چشم بخورد این است که در حاشیه سر، تعدادی از پیکسل‌ها به شکل کم‌رنگ ظاهر شده‌اند که این شاید به دلیل مشکل پیاده‌سازی باشد و شاید هم در اثر ترکیب سه کانال مستقل به وجود آمده باشد.

سوال ۵

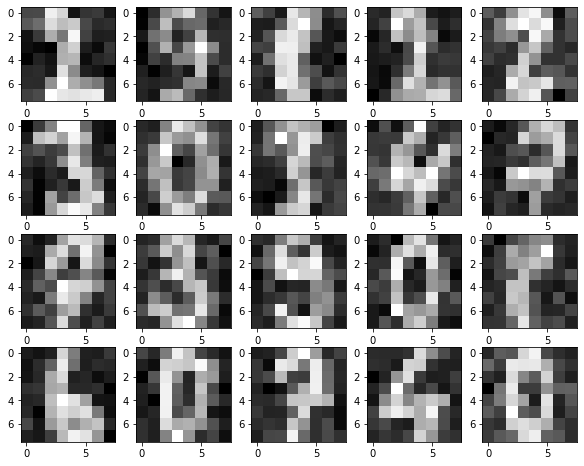
a) مقادیر ویژه عبارت است از ۱۷۹، 163.7، 141.7، 101.1، 69.5، 59.1، 51.8، 44، 40.3، 37، 28.5، 27.3، 21.9، 21.3 و 17.6

b) برای این قسمت از کرنل گاوسین استفاده کرده‌ام. در پیاده‌سازی sklearn این روش، ابرپارامتر bandwidth‌ وجود دارد که باید تنظیم دقیق شود. برای پیدا کردن ابرپارامتر bandwidth در تخمین توزیع کرنلی از روش GridSearchCV پیاده‌سازی شده در کتابخانه sklearn استفاده کردم. بدین ترتیب با جستجویی بر روی مقادیر مختلف bandwidth در بازه ۰ تا ۵ حالتی که به بهترین تخمین منجر شود انتخاب خواهد شد. مقدار 3.52 به عنوان مقدار بهینه برای bandwidth انتخاب شد.

c)



d) مقدار ابرپارامتر bandwidth برای این قسمت برابر با 1.75 شد و نمونه‌های زیر از تولیدات مدل است.

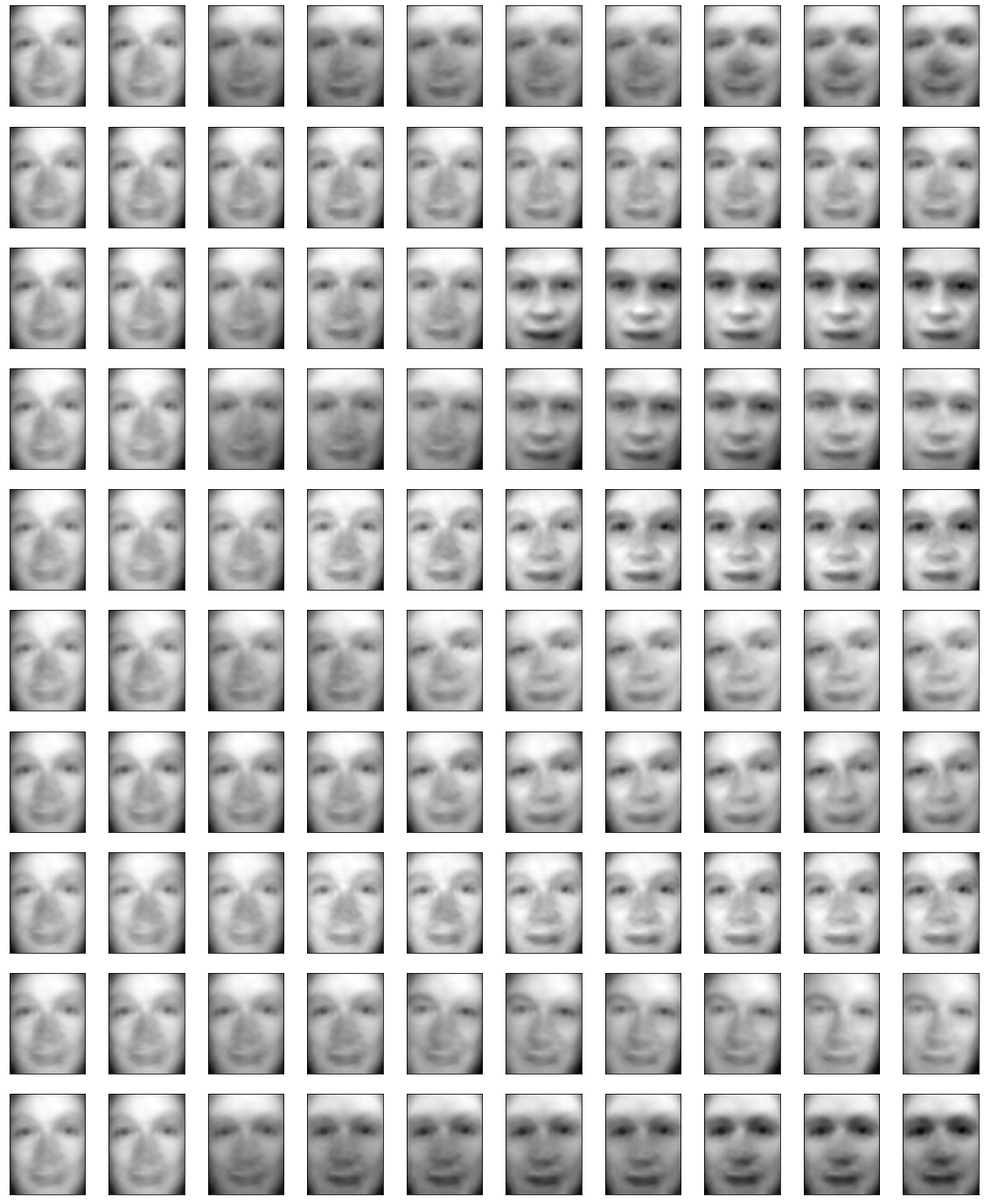


طبیعتا کیفیت تصاویر این قسمت از کیفیت تصاویر قسمت قبل بهتر است. عملا در قسمت قبل از ۱۵ مولفه‌ی با اهمیت بیشتر از 64 مولفه استفاده کرده‌ایم درحالی که در اینجا از تمام ۶۴ مولفه کمک گرفته‌ایم ولی آیا کیفیت در حالت کاهش ابعاد کمتر از یک چهارم شده است؟ واضحا خیر! با مقایسه تصویر می‌بینیم همچنان تصاویر کیفیت قابل قبولی دارند و بیشتر اعداد آن قابل خواندن است پس در مصالحه بین حجم فایل و کیفیت،‌ چنین کاهش جدی حجم در برابر کاهش کم کیفیت ارزنده است.

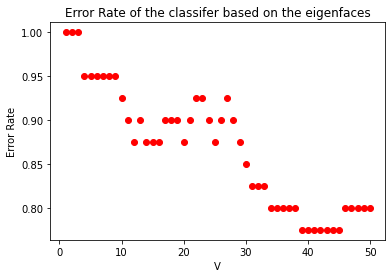
سوال ۶

b)

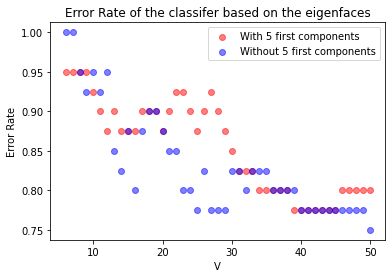
c)



d)



e)



پیش از هرچیز مشاهده می‌شود که افزایش تعداد مولفه‌ها منجر به کاهش خطا در هر دو حالت می‌شود. اما در عین حال خطا کلا بالاست. علت زیاد بودن خطا احتمالا به دلیل کاهش شدید تعداد مولفه‌های داده‌ها باشد. داده‌هایی که ما داریم دارای 4800 مولفه‌اند و در بهترین حالت تنها 50 مولفه اول حفظ شده‌اند که نسبتا کم است و حتی در قسمت c که تصاویر نمایش‌ داده‌شده می‌بینیم تصاویر مبهم هستند.

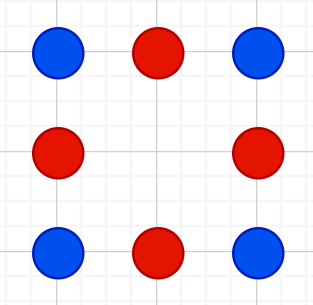
اما در مقایسه خطا بین دو حالتی که ۵ مولفه اول حفظ شده است با حالتی که این اتفاق نیافتاده است، مشاهده می‌کنیم که حذف این ۵ مولفه منجر به نتایج بهتری می‌شود! شاید بتوان به این شکل توجیه کرد؛ مولفه‌های اول برای ساخت یک تصویر چهره بسیار مهم هستند و به واسطه‌ی آن‌ها یک چهره‌ی خام ایجاد می‌شود. عملا مولفه‌های بعدی تمایز بین چهره‌ها را مشخص می‌کند و جزئیات یک چهره را تعیین می‌کند. پس زمانی که بحث دسته‌بندی باشد همین جزئیات اهمیت پیدا می‌کنند. وجود ویژگی‌های بی‌کاربرد (مولفه‌های اول) باعث کم‌اهمیت شدن ویژگی‌های مهم (مولفه‌های بعدی) و در نتیجه خطای بیشتر می‌شود.

سوال ۷

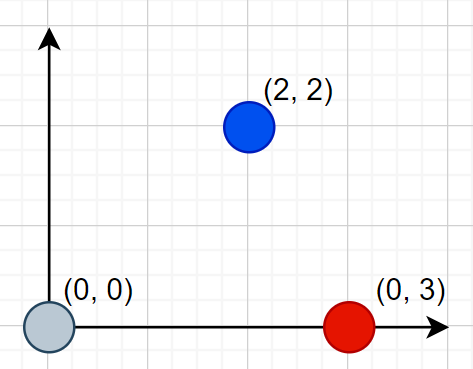
a) روش KNN استعداد بیشتری برای درگیرشدن به داده‌های پرت را دارد؛ در روش Parzen‌ انداره پنجره وابسته به داده نیست و یک مقدار ثابت است ولی در روش KNN‌ اندازه پنجره به داده‌ها وابسته است. بدین ترتیب وجود یک داده پرت در Parzen می‌تواند یک پنجره را خراب کند ولی وجود یک داده پرت در KNN ممکن است منجر به تحت تاثیر قرار گرفتن بخش مهمی از فضای اطراف داده شود به خصوص وقتی که k مقدار کمی را اتخاذ کرده است.

b) زمانی که k=1 است و مجموعه آموزشی به عنوان مجموعه تست به مدل داده شود طبیعتا هر داده به خودش در مجموعه آموزشی نزدیک‌تر است و برچسب خودش را هم می‌گیرد؛ بدین ترتیب خطا صفر خواهد شد.

اگر k=n باشد به ازای هر داده وروی برچسب کلاس اکثریت به کل داده‌ها منصوب خواهد شد. پس به تعداد داده‌های غیر بزرگترین کلاس خطا خواهیم داشت. طبیعتا در این شرایط خطا کمتر یا مساوی است، اگر c تعداد کلاس‌ها در مجموعه آموزشی باشد.

اگر k=3 باشد، نمی‌توان به صورت قطعی چیزی را گفت. به عنوان مثال اگر داده‌های هر کلاس به صورت متمرکز در کنار هم باشند و با داده‌های کلاس دیگر فاصله زیادی داشته باشند خطا صفر خواهد شد ولی مثلا برای شکل روبرو خطا برابر کل داده‌ها می‌شود!

c) به عنوان یک مثال خیلی ساده می‌توانید 1NN و مجموعه آموزش شامل دو داده آبی و قرمز زیر را در نظر بگیرید و مجموعه تست هم تنها شامل داده توسی رنگ باشد. در این مثال اگر فاصله منهتن را درنظر بگیریم داده توسی به کلاس قرمز تعلق پیدا می‌کند چرا که با داده قرمز دارای فاصله ۳ است ولی با داده آبی دارای فاصله ۴ است. حال فاصله اقلیدوسی را درنظر بگیرید. در این حالت داده توسی به کلاس آبی تعلق پیدا می‌کند چراکه با داده آبی دارای فاصله حدودا 2.8 است ولی با داده قرمز دارای فاصله ۳ است.



d) از منظر تئوری اگر تعداد داده‌ها به بی‌نهایت میل پیدا کند، هر دو روش این امکان را دارند که توزیع واقعی را عینا تحمین بزنند و طبیعتا در این حالت خروجی مشابه با یکدیگر خواهند داشت. برای روش Parzen اندازه پنجره اهمیتی ندارد ولی برای روش KNN باید k نیز به بی‌نهایت میل پیدا کند.

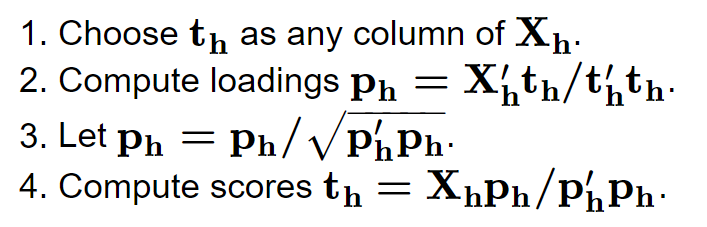
e) چنینی چیزی ممکن نیست. در PCA به تعداد ابعاد، بردار پایه در یک فضای جدید معرفی می‌شود و با داشتن تمام مولفه‌ها که تعدادشان برابر با ابعاد است، هیچ کاهش ابعادی رخ نمی‌دهد و داده‌ها کاملا حفظ می‌شود. از طرفی این مولفه‌ها باید نسبت به هم دارای استقلال خطی داشته باشند و چون ماتریس scatter به تعداد ابعاد ستون دارد اصلا نمی‌توان بیشتر از تعداد ابعاد بردار با استقلال خطی پیشنهاد داد. پس در این شرایط نه وجود مولفه بیشتر مفید است و نه شدنی.

f) مولفه‌های اول PCA بیشتر اطلاعات مربوط به تصویر را حفظ می‌کند و مولفه‌های آخر جزئیات و ریزکاری مربوط به تصویر را در خود دارند. نویزی که در تصویر وجود دارد روی تعداد پیکسل کمی تاثیرگذار است و تاثیر نویز به نوعی در مولفه‌های آخر PCA‌ است و با حذف این مولفه‌ها می‌توان نویز را هم حذف کرد؛ البته طبیعی است که برخی از جزئیات کم اهمیت از تصویر حذف خواهد شد.

g) در روش PCA باید بردار‌های ویژه ماتریس scatter را تشکیل داد که تعداد سطر و ستون آن برابر با تعداد ابعاد مسئله است. در این صورت برای یک عکس یک مگاپیکسلی حدودا یک میلیون بعد وجود خواهد داشت! پیچیدگی زمانی محاسبه بردار‌های ویژه هم نزدیک به است. در این شرایط مشخصا به مشکل خواهیم خورد. در کنار این قسمت از PCA‌ باید پیچیدگی زمانی محاسبه خود ماتریس scatter را نیز در نظر گرفت.

یکی از روش‌هایی که حجم محاسبات و حجم حافظه را کاهش می‌دهد روش NIPALS[[1]](#footnote-1) است. این روش برای وقتی کاربرد دارد که نیاز به تعداد کمی از مولفه‌های اول PCA‌ باشد.[[2]](#footnote-2) هرچند محاسبه تمام آن‌ها همچنان امکان‌پذیر است. طبیعتا هدف اصلی از PCA بدست آوردن مولفه‌های اول است و در شرایطی که اجرای PCA شدنی نیست به نظر مصالحه عقلانی‌ای باشد.

روال کار به این شکل است؛ فرض کنید X ماتریس داده‌ها باشد که توسط میانگین نرمال شده باشد و در این روش به دنبال تجزیه‌ای متناسب با هستیم. برای اولین مولفه تعریف می‌شود و چهار مرحله زیر انجام می‌شود و دو مرحله آخر انقدر انجام می‌شود که h امین مولفه اصلی ( فعلا اولین مولفه) همگرا شود: (در این مراحل پرایم به معنای ترنهاده است)



با درنظر گرفتن و h=h+1 می‌توان مولفه بعدی PCA‌ را محاسبه کرد. بردار ویژه مولفه hام به صورت بدست می‌آید.

اگر تمام مولفه‌های ویژه حساب شود می‌توان با کنار هم قرار دادن به عنوان ستون می‌توان ماتریس T و با کنار هم قرار دادن به عنوان ستون می‌توان ماتریس P را تشکیل داد.

1. https://cran.r-project.org/web/packages/nipals/vignettes/nipals\_algorithm.html [↑](#footnote-ref-1)
2. https://stats.stackexchange.com/a/7116/318893 [↑](#footnote-ref-2)