دانشکده مهندسی برق و کامپیوتر

تمرین پنجم یادگیری ماشین

زهرا ریحانیان

شماره دانشجویی: ۸۱۰۱۰۱۱۷۷

سوال اول

۱- خیر، model selection فرایند انتخاب یک مدل از یک مجموعه مدل های کاندید است که بهترین تناسب را با داده ها دارد. Model assessment فرایند ارزیابی میزان تناسب یک مدل با داده ها و تعمیم آن به داده های دیده نشده است.

۲- یک روش استفاده از cross-validation است. cross-validation تکنیکی است که شامل تقسیم داده ها به زیر مجموعه های متعدد، آموزش مدل بر روی یک زیر مجموعه و سپس آزمایش آن بر روی زیر مجموعه دیگر است. این فرایند تکرار می شود تا زمانی که همه ی زیر مجموعه ها هم برای آموزش و هم برای تست استفاده شوند. سپس مدلی با کمترین میانگین خطا در تمامی زیر مجموعه ها به عنوان بهترین مدل انتخاب می شود. این به کاهش overfitting کمک می کند تا اطمینان حاصل شود که مدل بیش از حد تنظیم نشده است که فقط به یک دیتاست خاص fit شود. در نتیجه خطای generalization حداقل می شود.

۳- وقتی که داده های کمی در دیتاست داریم میتوانیم از روش هایی مثل cross-validation و bootstrapping برای ارزیابی و انتخاب یک مدل استفاده کنیم. cross-validation تکنیکی است که برای ارزیابی دقت یک مدل با تقسیم داده ها به مجموعه های training و testing، سپس استفاده از مجموعه training برای fit کردن مدل و مجموعه testing برای ارزیابی عملکرد آن استفاده می شود. Bootstrapping روش دیگری است که برای ارزیابی مدل ها در صورت وجود داده های محدود استفاده می شود. این روش شامل چندین بار نمونه گیری تصادفی با جایگزینی از دیتاست و fit کردن مدل روی هر sample است. سپس نتایج با هم میانگین گرفته می شود تا یک تخمین کلی از عملکرد مدل ارائه شود.

٤- Precision یکی از متریک های ارزیابی مدل است. Precision این که مدل در شناسایی صحیح کلاس مثبت چقدر خوب است را اندازه می گیرد. به عبارت دیگر از بین تمام پیش بینی های کلاس مثبت، چند مورد واقعا درست بوده است. تنها با استفاده از این معیار برای بهینه سازی یک مدل، موارد مثبت کاذب را می توان به حداقل رساند. این ممکن است برای مثال برای تشخیص تقلب در امتحان مطلوب باشد اما برای تشخیص سرطان کمتر مفید خواهد بود، چون درک کمی از مشاهدات مثبتی که نادیده گرفته می شوند، خواهیم داشت.

سوال دوم

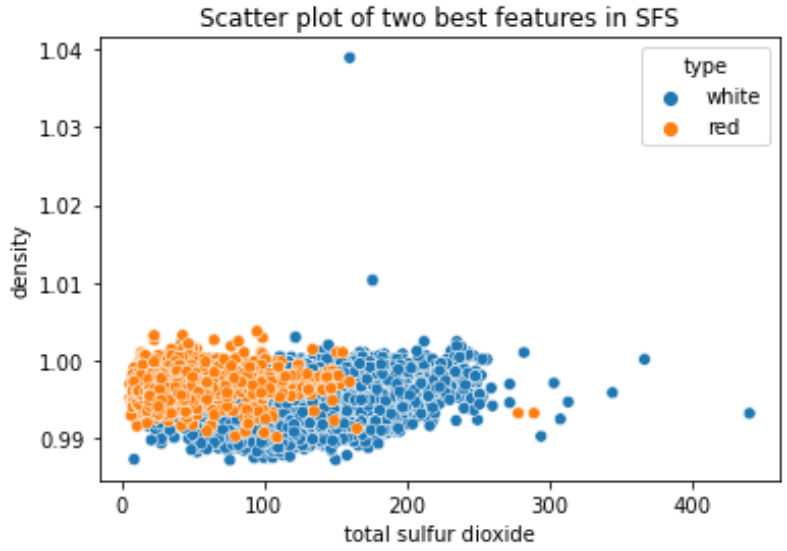
۱- Feature selection فرایند انتخاب زیر مجموعه ای از ویژگی ها از مجموعه بزرگ تر ویژگی ها است که بیشترین ارتباط را با مساله در دست دارند. هدف از آن کاهش پیچیدگی مدل، بهبود دقت آن و کاهش overfitting است. همچنین به کاهش زمان training و بهبود تفسیرپذیری کمک می کند.
دادن همه ی ویژگی ها به یک الگوریتم یادگیری ماشین می تواند منجر به overfitting شود، زیرا ممکن است ویژگی های نامربوطی را انتخاب کند که به دقت مدل کمکی نمی کند. علاوه بر این، داشتن ویژگی های خیلی زیاد میتواند منجر به ناکارآمدی محاسباتی و کاهش سرعت زمان training شود.

۲- Fisher’s score یک روش آماری است که برای انتخاب ویژگی های یک مدل پیش بینی استفاده می شود. با اندازه‌گیری همبستگی بین هر ویژگی و متغیر هدف، و سپس انتخاب ویژگی‌هایی با بالاترین همبستگی، کار می‌کند. امتیاز با در نظر گرفتن نسبت واریانس بین کلاس به واریانس درون کلاس برای هر ویژگی محاسبه می شود. هر چه این نسبت بیشتر باشد، آن ویژگی در پیش بینی متغیر هدف اهمیت بیشتری دارد.

سوال سوم

۱- الگوریتم sequential forward selection به این صورت است که ابتدا بهترین تک ویژگی را انتخاب می کند. یعنی تک تک ویژگی ها را بررسی می کند و آنکه بالاترین score را در مقایسه با بقیه ویژگی ها را بدست آورد را انتخاب می کند. در مرحله بعد، جفت ویژگی های متشکل از بهترین ویژگی و ویژگی های باقی مانده را بررسی میکند. آن جفت ویژگی ای که بهترین score را بدست آورد را انتخاب می کند. مرحله بعد ویژگی های سه تایی متشکل از بهترین جفت ویژگی و ویژگی های باقی مانده را بررسی می کند و سه ویژگی ای که بهترین score را بدست آورد را انتخاب می کند. این فرایند تا جایی ادامه پیدا می کند که یا ویژگی ای باقی نماند یا به تعداد ویژگی از پیش تعیین شده برسیم.
پیاده سازی این الگوریتم در فایل جوپیتر نوت بوک قابل مشاهده می باشد. قبل از استفاده از این الگوریتم لازم است که مقادیر NAN حذف شوند. برای همین از دستور dropna() استفاده شده است. از مدل knn برای قسمت محاسبه score استفاده کردم. به این صورت که ابتدا مدل را روی داده ها fit میکند سپس با استفاده از متد score() که یک معیار ارزیابی پیش فرض هر مدل برای مساله ای که برای حل آن طراحی شده اند ارائه می کند، score محاسبه می شود. به این ترتیب الگوریتم بهترین ویژگی ها را در هر مرحله انتخاب می کند.

در اینجا برای دیتاست داده شده، بهترین جفت ویژگی انتخاب شده، total sulfur dioxide و density هستند. نمودار نقاط این دو ویژگی با لیبل های متناظر برای هر wine به صورت زیر بدست آمد:

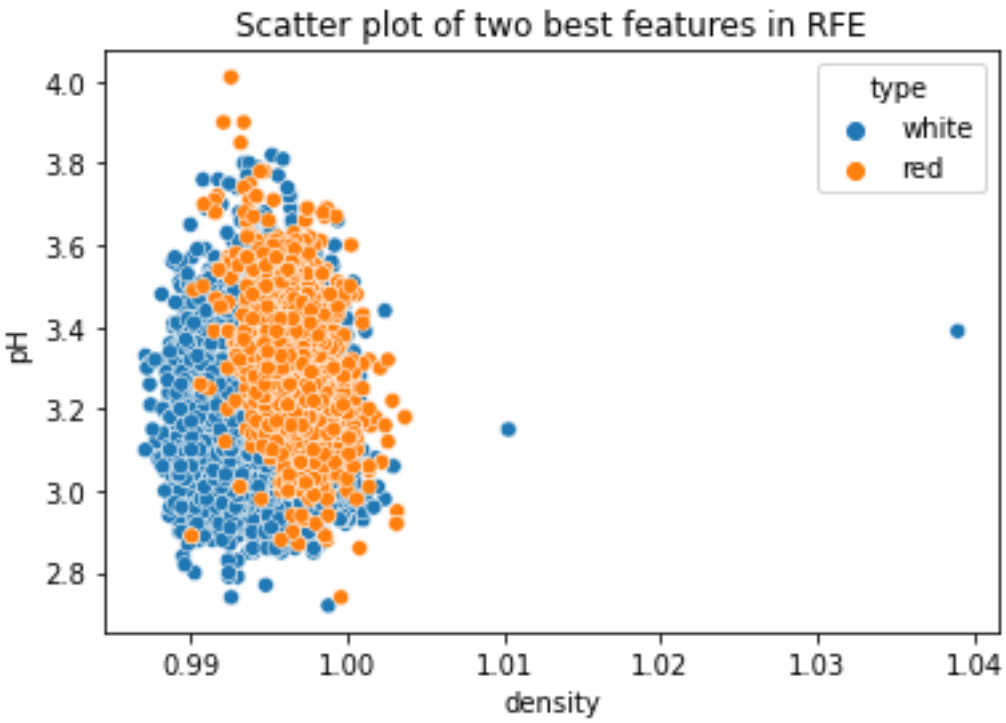


شکل ۱ نمودار نقاط بهترین جفت ویژگی بدست آمده از الگوریتم SFS

همان طور که مشاهده می شود، این دو ویژگی تا حد خوبی دو نوع wine را از هم جدا کرده است.

۲- الگوریتم Recursive Feature Elimination به این صورت عمل می کند که برای مثال d ویژگی داشته باشیم، هر بار یکی از ویژگی ها را حذف می کند و score این $d-1$ ویژگی را حساب می کند. آن ویژگی ای که با حذف آن بیشترین score بدست بیاید را حذف می کند. در مرحله بعد $d-1$ ویژگی داریم. هر بار یکی از ویژگی ها را حذف می کند و score این $d-2$ ویژگی را حساب می کند. مشابه مرحله قبل آن ویژگی ای که با حذف آن بیشترین score بدست بیاید را حذف می کند. به همین ترتیب ادامه می هد تا جایی که به تعداد ویژگی از قبل تعریف شده برسد یا ویژگی ای باقی نماند.

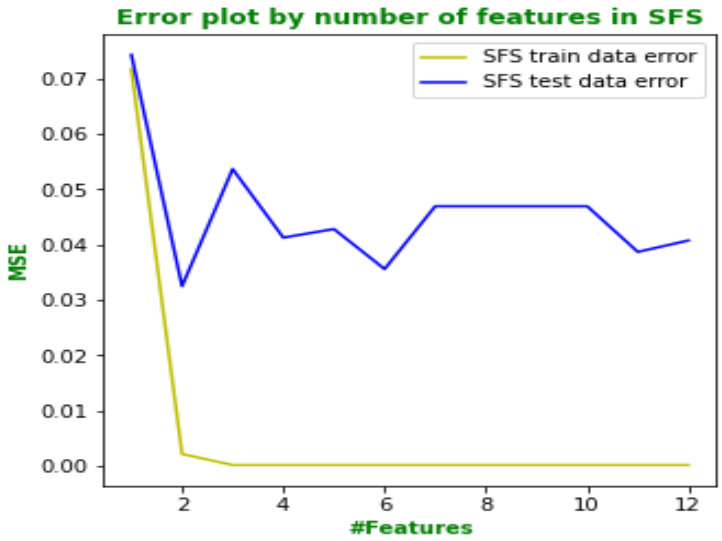
در اینجا برای دیتاست داده شده، بهترین جفت ویژگی انتخاب شده، pH و density هستند. نمودار نقاط این دو ویژگی با لیبل های متناظر برای هر wine به صورت زیر بدست آمد:



همان طور که دیده می شود، این دو ویژگی توانستند تا حد خوبی دو نوع wine را از هم جدا کنند.

۳- برای الگوریتم های گفته شده، در قسمت قبل بهترین ویژگی هایی که توسط هر کدام انتخاب شد، نمایش داده شد.

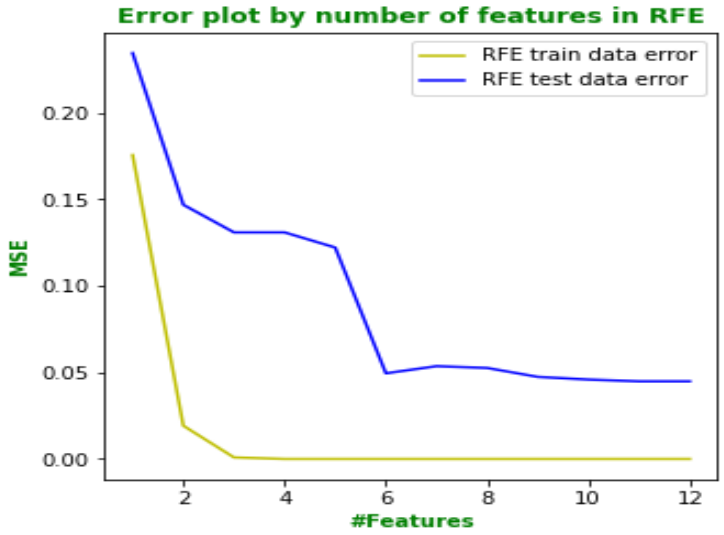
نمودار خطا بر حسب تعداد ویژگی ها برای الگوریتم Sequential Forward Selection :



مقدار خطا برای داده های آموزش در کل از مقدار خطا برای داده های تست کمتر است و یک روند نزولی دارد. یعنی با افزایش تعداد ویژگی ها مقدار خطا کاهش می یابد اما برای داده های تست، یک روند مشخصی ندارد. با

افزایش تعداد ویژگی ها، مقدار خطا گاهی افزایش، گاهی کاهش و گاهی افزایش می یابد. کمترین خطا برای حالت دو ویژگی و بیشترین خطا برای حالت یک ویژگی برای داده های تست بدست آمد.

نمودار خطا بر حسب تعداد ویژگی ها برای الگوریتم Recursive Feature Elimination :



در اینجا همانند نمودار خطا بر حسب تعداد ویژگی ها برای الگوریتم Sequential Forward Selection مقدار خطا برای داده های آموزش در کل از مقدار خطا برای داده های تست کمتر است و یک روند نزولی دارد. اما بر خلاف آن، برای داده های تست، میتوان گفت نمودار در حالت کلی روند نزولی دارد و به خطای کمتری نسبت به آنچه برای الگوریتم Sequential Forward Selection مشاهده شد، می توان رسید.

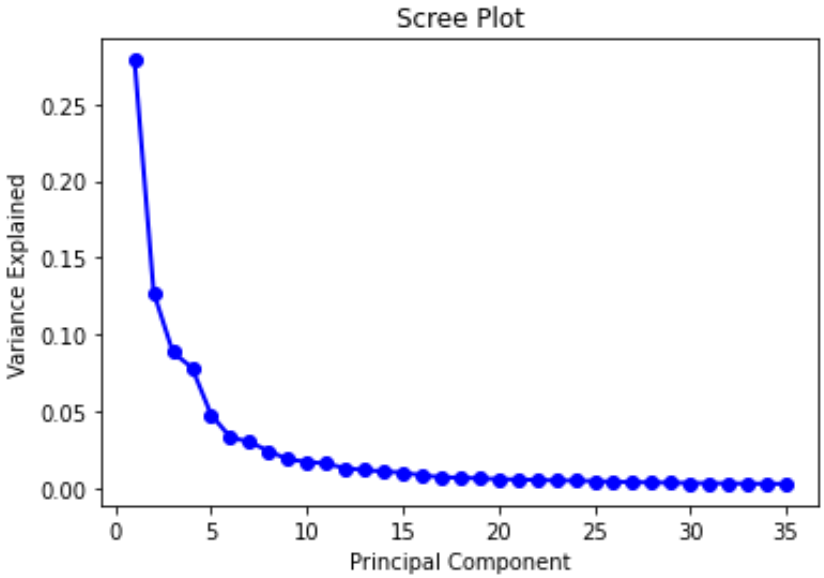
در مورد مدت زمان انجام یا همان سرعت دو الگوریتم، الگوریتم Sequential Forward Selection به طور میانگین 2.85 ثانیه و الگوریتم Recursive Feature Elimination به طور میانگین 7.5 ثانیه برای انتخاب ویژگی زمان می گیرند. پس می توان حدس زد که الگوریتم Sequential Forward Selection سریع تر است.

سوال چهارم

PCA می تواند دقت وظایف پردازش تصویر را با کاهش ابعاد داده ها بهبود بخشد. به عنوان مثال، در کارهای تشخیص چهره، PCA می تواند برای کاهش تعداد ویژگی های یک تصویر چهره از صدها به چند جزء اصلی استفاده شود. این عمل میزان داده‌هایی را که باید پردازش شوند کاهش می‌دهد، که می‌تواند منجر به بهبود دقت و زمان پردازش سریع‌تر شود.

۱- برای پیدا کردن مقادیر ویژه PCA این دیتاست، از کلاس PCA از کتابخانه sklearn استفاده کردم. مقادیر ویژه PCA با نسبت واریانس توضیح داده شده مرتبط هستند، زیرا نشان دهنده مقدار واریانس

در داده ها هستند که توسط هر جزء اصلی توضیح داده می شود. مقادیر ویژه با گرفتن مجموع بارهای مجذور برای هر جزء محاسبه می شود، در حالی که نسبت واریانس توضیح داده شده با تقسیم مقدار ویژه بر مجموع همه مقادیر ویژه محاسبه می شود. بنابراین، یک مقدار ویژه بالاتر منجر به نسبت واریانس توضیح داده شده بالاتر خواهد شد. به همین دلیل از explained_variance_ratio_ که یکی از attribute های کلاس pca است برای نشان دادن مقادیر ویژه است کردم. تعداد کامپوننت ها را ۳۵ تنظیم کردم و روی داده های آموزش، pca زدم. نتیجه به صورت زیر شد:



همان طور که مشاهده می شود با افزایش تعداد کامپوننت ها، مقدار variance explained کاهش می یابد. و یه جایی به بعد تقریبا ثابت می ماند و نزدیک به صفر است.

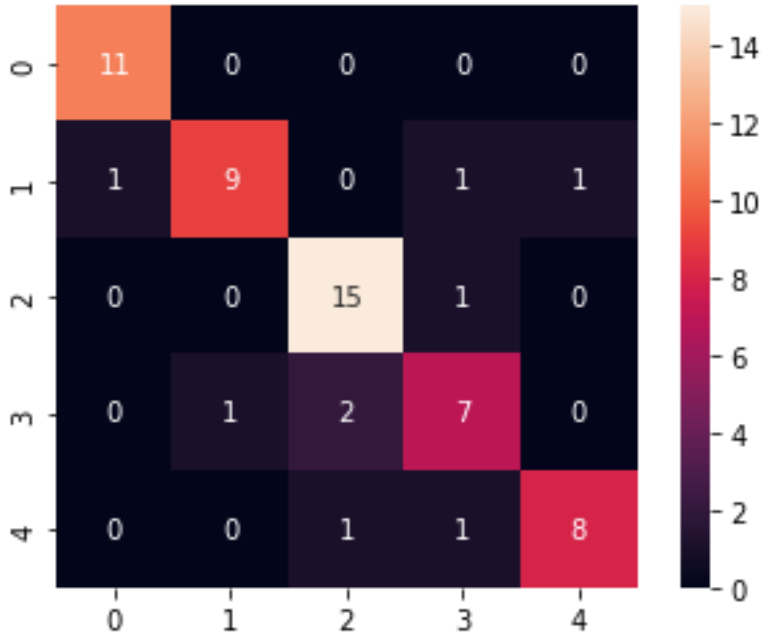
تعداد کامپوننت مناسب در PCA را می توان با بررسی explained variance ratio هر کامپوننت شناسایی کرد. explained variance ratio نشان دهنده میزان واریانس داده هایی است که توسط هر کامپوننت توضیح داده می شود. به طور کلی، کامپوننت هایی با explained variance ratio بیشتر از ۰.۵ مهم در نظر گرفته می شوند و باید در مدل گنجانده شوند. علاوه بر این، می توان از نمودارهای scree یا روش های دیگری مانند Kaiser criterion برای شناسایی تعداد بهینه کامپوننت استفاده کرد.

۲-

۳- طبقه بند KNN بر روی داده های خالص:

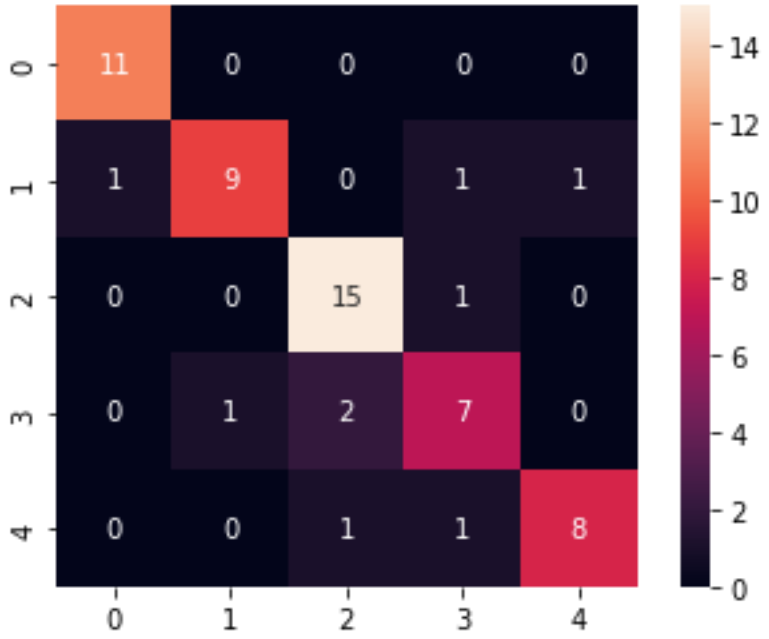
K = 1

ماتریس آشفتگی:



CCR برابر 0.85 شد.

K = 2



CCR برابر 0.85 شد.

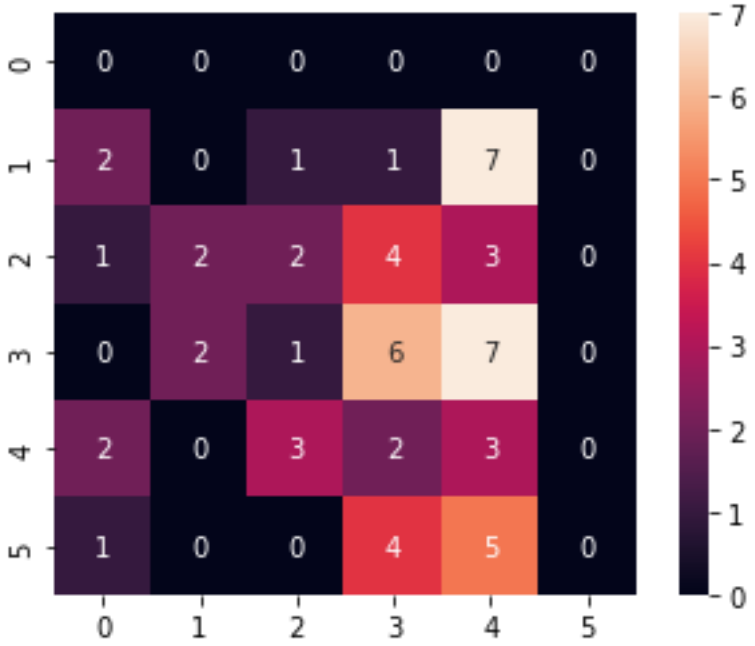
همان طور که مشاهده می شود برای داده های خالص $k = 1$ و $k = 2$ نتایج یکسانی را دادند.

طبقه بند KNN بر روی داده های کاهش بعد یافته:

ابتدا PCA را برای هر دو داده ی آموزش و تست اعمال می کنیم. نتایج به صورت زیر حاصل شد:

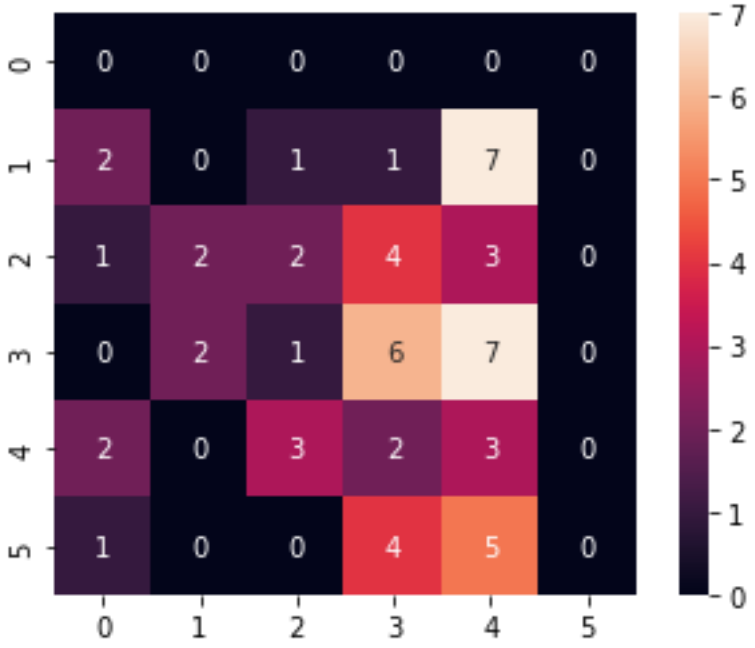
K = 1

ماتریس آشفتگی:



CCR برابر 0.19 شد.

K = 2

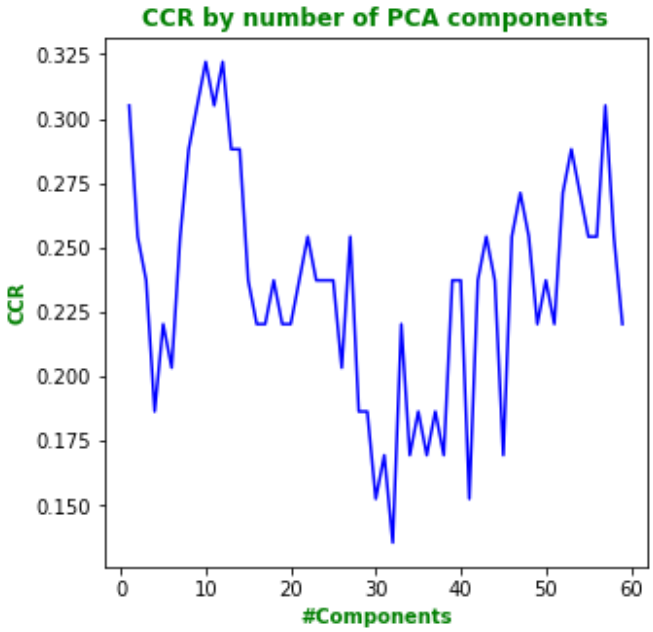


CCR برابر 0.19 شد.

همان طور که مشاهده می شود برای داده های کاهش بعد یافته هم $k = 1$ و $k = 2$ نتایج مشابهی را تولید کرد. دقت در داده های کاهش بعد یافته خیلی کم شد. شاید به این علت باشد که تعداد کامپوننت ها در مقایسه با تعداد ویژگی های اصلی خیلی کمتر است. این بخاطر این است که تعداد کامپوننت pca باید مینیمم تعداد ویژگی ها و تعداد نمونه ها باشد. و این pca را باید روی هر دو داده ی آموزش و تست باید اعمال می شد.

یک تفاوت دیگر با حالتی که با داده های خالص طبقه بند را اعمال کردیم در ماتریس آشفتگی می باشد. ابتدا باید به این اشاره شود که در این دیتاست داده های آموزش یک کلاس بیشتر از داده های تست دارند که آن کلاس angry است در حالی که داده های تست حتی یک نمونه هم از این کلاس ندارد. ماتریس آشفتگی داده های خالص همان طور که مشاهده شد ۵ در ۵ است که ۵ همان تعداد کلاس های متمایز در داده های تست است. یعنی در حالت داده های خالص هیچ داده ای به اشتباه در کلاس angry قرار نگرفت در صورتی که ماتریس آشفتگی داده های کاهش بعد یافته ۶ در ۶ است یعنی در اینجا طبقه بند به اشتباه بعضی از داده های تست را در کلاس angry قرار داده است.

۴- نتیجه ی نمودار خواسته شده به صورت زیر درآمد:



همان طور که مشاهده می شود نمودار یک روند کاملا نزولی یا صعودی ندارد. کمترین CCR مربوط به تعداد کامپوننت ۳۲ است و بالاترین هم مربوط به ۱۰ تا کامپوننت است. بنابراین می توان نتیجه گرفت که CCR رابطه ی مشخصی با تعداد کامپوننت PCA ندارد.

سوال پنجم

۱- اگر داشته باشیم:

$$\overline{x_i} = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{i,j}$$

$$\Sigma_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (x_{i,j} - \overline{x_i})(x_{i,j} - \overline{x_i})^T$$

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{g} N_i \overline{x_i} = \frac{1}{N} \sum_{i=1}^{g} \sum_{j=1}^{N_i} x_{i,j}$$

ماتریس پراکندگی بین کلاسی به صورت زیر تعریف می شود:

$$S_b = \sum_{i=1}^{g} N_i (\overline{x_i} - \overline{x})(\overline{x_i} - \overline{x})^T$$

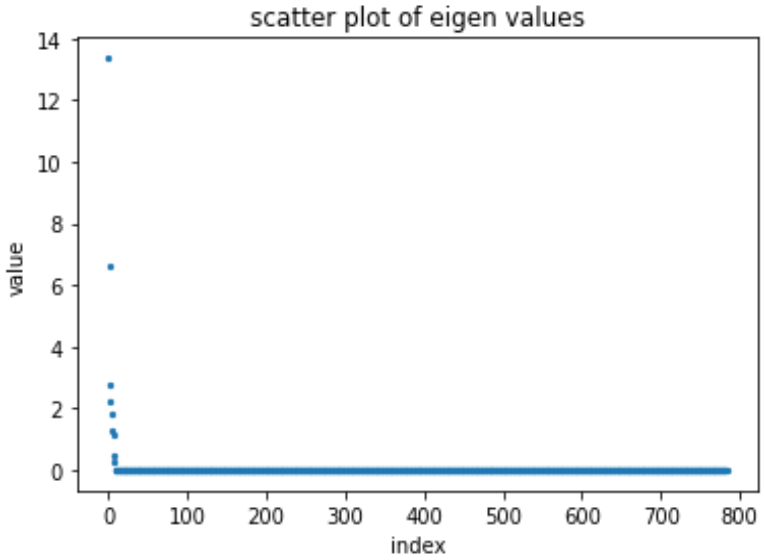
این ماتریس پراکندگی بین دو کلاس یعنی این که چقدر از هم جدا هستند یا دور هستند را نشان میدهد.
ماتریس پراکندگی درون کلاسی به صورت زیر تعریف می شود:

$$S_w = \sum_{i=1}^{g} (N_i - 1)\Sigma_i = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \overline{x_i})(x_{i,j} - \overline{x_i})^T$$

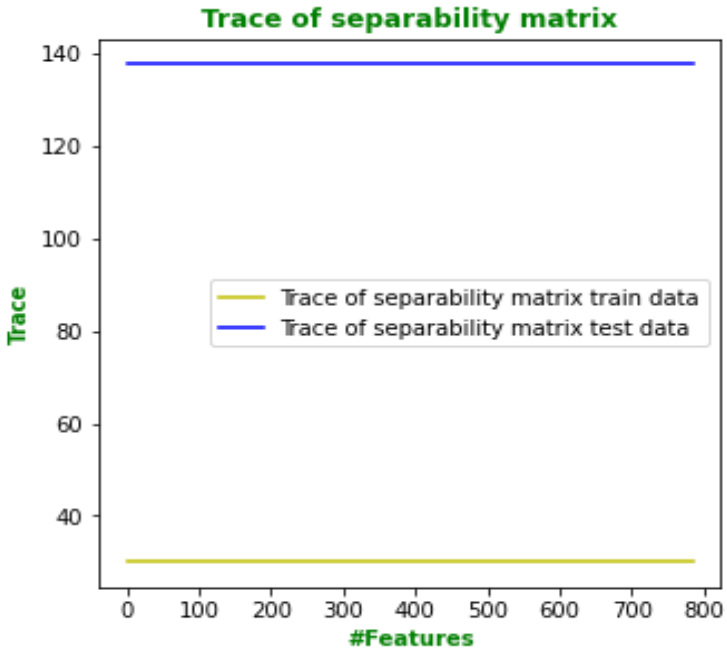
این ماتریس میزان فشردگی یا پراکندگی نمونه های موجود در یک کلاس را نشان می دهد.
پیاده سازی این الگوریتم در فایل جوپیتر نوت بوک موجود می باشد. این الگوریتم به این صورت کار می کند که ابتدا ماتریس پراکندگی درون کلاسی و برون کلاسی را محاسبه می کند. سپس بردار های ویژه

ماتریس جدایی پذیری ($S_W^{-1}S_B$) را بدست می آورد. آن ها را به ترتیب نزولی مقادیر ویژه متناظرشان مرتب می کند و بر حسب تعداد ویژگی مورد نظر حاصل ضرب آن تعداد از بردار های ویژه را با ماتریس ویژگی بدست می آورد و به این ترتیب کاهش بعد را روی داده ها اعمال می کند.

۲- ابتدا با کمک کلاس LDA پیاده سازی شده با تعداد کامپوننت ماکزیمم، ماتریس پراکندگی درون کلاسی و برون کلاسی محاسبه شد و سپس ماتریس جدایی پذیری محاسبه شد. در نهایت مقادیر ویژه به کمک تابع linalg.eig کتابخانه numpy بدست آمد. این مقادیر را در یک نمودار نقاط رسم کردم که حاصل به این صورت شد:



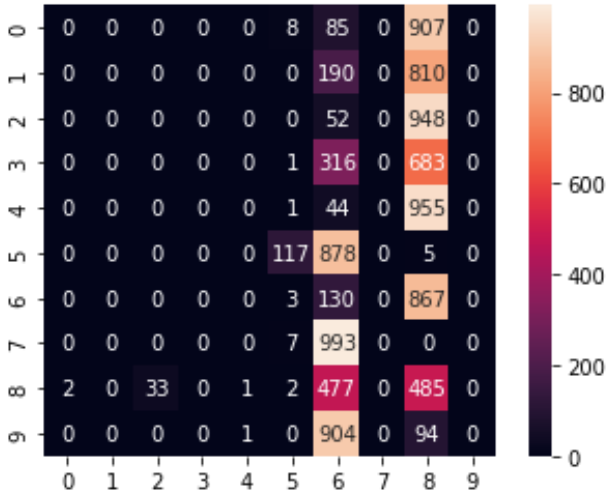
۳- در یک حلقه for تمام حالت های LDA هم برای داده های آموزش و هم برای داده های تست، با تعداد کامپوننت های مختلف را بررسی کردم و برای هر کدام trace ماتریس جدایی پذیری را محاسبه کردم. نتایج را در یک نمودار رسم کردم. نتیجه به صورت زیر حاصل شد:



همان طور که مشاهده می شود trace ماتریس جدایی پذیری با افزایش تعداد ویژگی ها تغییری نمی کند و ثابت می ماند.

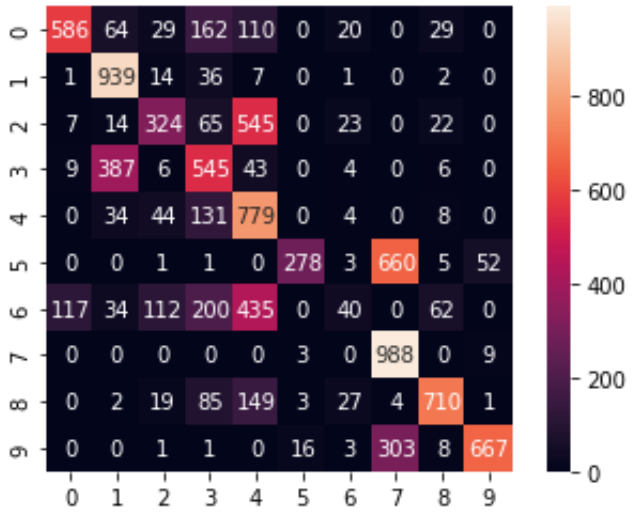
۴- با توجه به دو قسمت قبل، من تعداد کامپوننت را ۱۰ انتخاب کردم. چون با توجه به نمودار قسمت ۲، از مقدار ویژه ی دهم به بعد، مقدار ویژه ثابت باقی می ماند.

موارد خواسته شده در صورت سوال انجام شد و طبقه بند را بر روی داده های کاهش بعد یافته اعمال شد. CCR آن 0.07 شد و ماتریس آشفتگی به صورت زیر در آمد:



همان طور که مشاهده می شود عملکرد اصلا رضایت بخش نیست.

۵- این بار طبقه بند را بدون LDA روی داده اعمال کردم. CCR برابر 0.59 شدو ماتریس آشفتگی به صورت زیر حاصل شد:



همان طور که مشاهده می شود عملکرد طبقه بند بر روی داده ی خالص بهتر از داده ی کاهش بعد یافته شده با LDA شد و نتیجه بسیار رضایت بخش تر از حالت قبل است.

سوال ششم

۱- خیر، استفاده از فاصله بین نمونه ها می تواند معیار خوبی برای عدم تشابه در الگوریتم های خوشه بندی باشد، زیرا امکان مقایسه نمونه های مختلف بر اساس فاصله آنها از یکدیگر را فراهم می کند. با این حال، این روش می تواند در شرایط خاص نادرست یا مخالف باشد. به عنوان مثال، اگر دو نمونه از نظر فاصله بسیار نزدیک به هم باشند اما ویژگی های بسیار متفاوتی داشته باشند، این روش ممکن است به اشتباه آنها را به عنوان مشابه طبقه بندی کند، در حالی که واقعاً کاملاً متفاوت هستند. علاوه بر این، اگر دو نمونه از نظر فاصله از هم دور باشند، اما ویژگی های مشابهی داشته باشند، این روش ممکن است به اشتباه آنها را به عنوان غیرمشابه طبقه بندی کند، در حالی که واقعاً کاملاً مشابه هستند.

۲- DBSCAN مخفف Density-Based Spatial Clustering of Applications with Noise است که یک الگوریتم پایه برای density-based clustering است. این الگوریتم نقاط را در یک مجموعه داده بر اساس چگالی آن ها گروه بندی می کند.

DBSCAN دو پارامتر نیاز دارد: $\varepsilon(esp)$ و حداقل تعداد مورد نیاز برای تشکیل یک ناحیه متراکم (minPts).

مراحل این الگوریتم به صورت زیر می باشد:

۱. نقاطی را در همسایگی $\varepsilon(esp)$ هر نقطه بیابید و نقاط هسته را با تعداد همسایگان بیش از مقدار $minPts$ شناسایی کنید.
۲. کامپوننت های متصل نقاط هسته را در نمودار همسایه بیابید و تمام نقاط غیر هسته را نادیده بگیرید.
۳. هر نقطه غیر هسته را به یک خوشه نزدیک نسبت دهید، اگر آن خوشه یک همسایه $\varepsilon(esp)$ است وگرنه آن را به یک نویز نسبت دهید.

همان طور که گفته شد این الگوریتم در دسته density-based از الگوریتم های clustering قرار می گیرد.

الگوریتم OPTICS(Ordering Points To Identify Clustering Structure) شبیه به DBSCAN است اما به جای استفاده از چگالی، از فاصله دسترسی برای شناسایی خوشه ها استفاده می کند. فاصله قابلیت دسترسی معیاری است که نشان می دهد دو نقطه از نظر فاصله آنها از سایر نقاط مجموعه داده چقدر نزدیک هستند. الگوریتم OPTICS همچنین به انعطاف پذیری بیشتری در هنگام تعریف خوشه ها اجازه می دهد، زیرا می تواند خوشه ها را با اشکال و اندازه های مختلف شناسایی کند. تفاوت اصلی بین DBSCAN و OPTICS این است که DBSCAN برای تشکیل یک خوشه به حداقل تعداد نقاط نیاز دارد، در حالی که OPTICS این نیاز را ندارد.

تفاوت های دیگر OPTICS و DBSCAN :

۱- هزینه حافظه: تکنیک خوشه‌بندی OPTICS به حافظه بیشتری نیاز دارد زیرا یک صف اولویت (Min Heap) را برای تعیین نقطه داده بعدی که نزدیک‌ترین نقطه پردازش شده در حال حاضر از نظر فاصله دسترسی است، حفظ می‌کند. همچنین به قدرت محاسباتی بیشتری نیاز دارد زیرا پرس و جوهای نزدیکترین همسایه پیچیده تر از پرس و جوهای شعاع در DBSCAN هستند.
۲- پارامترهای کمتر: تکنیک خوشه‌بندی OPTICS نیازی به حفظ پارامتر اپسیلون ندارد و فقط در شبه کد بالا برای کاهش زمان صرف شده است. این منجر به کاهش فرآیند تحلیلی تنظیم پارامتر می شود.
۳- OPTICS داده های داده شده را به خوشه ها تفکیک نمی کند. این فقط یک نمودار فاصله قابل دسترسی تولید می کند و بر اساس تفسیر برنامه نویس است که نقاط را بر این اساس خوشه بندی کند.

سوال هفتم

مولفه اول PCA خط قرمز و مولفه اول LDA خط مشکی می توانند باشند. چون PCA جهتی را انتخاب می کند که داده ها بیشترین پراکندگی را دارند و چون unsupervised است به لیبل داده ها کاری ندارد. اما LDA یک الگوریتم supervised است و جهتی را انتخاب میکند که با project داده ها بر روی آن، بیشترین جدا پذیری برای کلاس های آن داده ها رقم بخورد.