Machine Learning

Assignment 5

Fatemeh Nadi 810101285

February 11, 2023

سوال اول

بخش اول

$Model\ selection$ به فرآیند تخمین عملکرد مدل‌های مختلف به منظور انتخاب بهترین مدل بر اساس عواملی مانند دقت، سرعت و غیره اطلاق می‌شود. از سوی دیگر $model\ assessment$ به فرآیندی اشاره دارد که در آن یک مدل انتخاب می‌شود و عملکرد آن بر روی داده‌های جدید آزمایش می‌شود.

بخش دوم

برای انتخاب مدلی با generalization error کمتر، می توان از قسمت کردن داده به مجموعه‌های $Train, Test, Validation$ بهره برد. قسمتی از داده ها را با عنوان Train برای آموزش استفاده می‌کنیم ر قسمت دیگر یعنی Test را را به مدل نمی‌دهیم. سپس یک مجموعه را می توان در فرآیند آموزش به عنوان اعتبارسنجی برای تنظیم hyperparameter ها برای کمک به یادگیری بهتر و همچنین برای تخمین generalization error استفاده کرد. مجموعه تست همچنین می تواند برای نشان دادن generalization error استفاده شود. به این روش cross-validation می‌گویند و برای پیاده سازی آن روش‌های زیادی وجود دارد مانند k-fold و نمونه گیری تصادفی. پس از پیاده‌سازی این روش، مدلی انتخاب می‌شود که validation کمتر دارد یا test error آن در دورهای مختلف کمتر از میانگین است. به این ترتیب می توان مطمئن شد که مدل با generalization error کمتر انتخاب شده است.

بخش سوم

هنگامی که که تعداد داده‌های کمی در دیتاست داریم، چندین رویکرد وجود دارد که می توان در نظر گرفت.

- استفاده مکرر از k-fold
- استفاده از روش leave one out

در این روش، تعداد بخش‌ها برابر با تعداد مشاهدات است. به این ترتیب در هر بار، یک مشاهده از داده‌های آموزش کنار گذاشته شده و مدل براساس بقیه ساخته می‌شود. سپس خطای مقداری که کنار گذاشته شده اندازه‌گیری می‌شود. با تکرار این عمل برای همه مشاهدات میانگین خطای حاصل، برآوردی برای خطای مدل خواهد بود.

بخش چهارم

معیارهایی مانند $precision, recall, accuracy$ راه‌های خوبی برای ارزیابی مدل‌های classification هستند که برای مجموعه داده‌های متوازن به‌کار می‌روند، اما اگر داده‌ها نامتوازن باشند، روش‌های دیگر مانند ROC/AUC در ارزیابی عملکرد مدل بهتر عمل می‌کنند.

- Precision در حالت ایده آل برای یک طبقه بندی خوب باید 1 باشد. Precision فقط زمانی 1 می شود که صورت و مخرج برابر باشند، یعنی:

$$TP = TP + FP$$

این همچنین به این معنی است که FP صفر است. با افزایش FP مقدار مخرج بزرگتر از صورتگر می شود و مقدار Precision کاهش می یابد.
به بیان دیگر به معنای این است که بین مواردی که به عنوان positive اعلام شده‌اند، چه تعداد از آنها به درستی positive هستند. یعنی:

$$Precision = \frac{TP + FP}{TP}$$

- Recall در حالت ایده آل برای یک طبقه بندی خوب باید 1 باشد. تنها زمانی که صورت و مخرج برابر باشند، Recall برابر 1 می شود، یعنی:

$$TP = TP + FN$$

همچنین به این معنی است که FN صفر است. با افزایش FN مقدار مخرج بزرگتر از صورتگر می شود و مقدار Recall کاهش می یابد.
به بیان دیگر به معنای این است که چند درصد از مواردی که positive هستند، اعلام شده‌اند. یعنی:

$$Recall = \frac{TP}{TP + FN}$$

بنابراین در حالت ایده آل در یک classifier خوب، ما می خواهیم هم Precision و هم Recall برابر یک باشد که همچنین به معنای FN و FP صفر است. بنابراین ما به معیاری نیاز داریم که هم Precision و هم Recall را در نظر بگیرد. امتیاز F1 معیاری است که هم Precision و هم Recall را در نظر می گیرد و به صورت زیر تعریف می شود:

$$F1 - SCORE = 2 \times \frac{recall \times precision}{recall + precision}$$

امتیاز F1 تنها زمانی 1 می شود که Precision و Recall هر دو 1 باشد. پس امتیاز F1 تنها زمانی بالا می شود که Precision و Recall هر دو بالا باشند. امتیاز F1 میانگین هارمونیک Precision و Recall است و معیاری بهتر از accuracy است.

سوال دوم

بخش اول

تکنیک‌های feature selection برای کاهش تعداد متغیرهای ورودی با حذف ویژگی‌های اضافی یا نامربوط و محدود کردن مجموعه ویژگی‌ها به موارد مرتبط با مدل یادگیری ماشین استفاده می‌شوند.درک چگونگی انتخاب ویژگی‌های مهم در یادگیری ماشین از منظر کارایی الگوریتم یادگیری ماشین بسیار مهم است. ویژگی‌های نامربوط، اضافی و نویزی می‌توانند الگوریتم را آلوده کنند و بر عملکرد یادگیری، دقت و هزینه محاسباتی تأثیر منفی بگذارند.همچنین، حذف پیش‌بینی‌کننده‌های همبسته در feature selection، چند خطی بودن را کاهش می‌دهد و بنابراین به مدل‌هایی مانند رگرسیون خطی یا لجستیک را که در برابر پیش‌بینی‌کننده‌های همبسته آسیب‌پذیر هستند، امکان می‌دهد تا به خوبی fit شوند. با توجه به اینکه اندازه و پیچیدگی مجموعه های داده به طور تصاعدی در حال رشد است، feature selection اهمیت فزاینده ای دارد. سه مزیت کلیدی feature selection عبارتند از:

- Over-fitting را کاهش می دهد: داده های اضافی کمتر به معنای شانس کمتری برای تصمیم گیری بر اساس نویز است.
- دقت را بهبود می بخشد: داده های گمراه کننده کمتر به معنای دقت مدل سازی بهتر است.
- زمان تمرین را کاهش می دهد: داده های کمتر به معنای الگوریتم های سریعتر است.

مزایای اصلی انجام feature selection به جای اینکه مدل یادگیری ماشین بفهمد کدام ویژگی مهمتر است، عبارتند از:

- تولید مدل‌های ساده‌تر: توضیح مدل‌های ساده آسانتر است و مدلی که بیش از حد پیچیده و غیرقابل توضیح باشد، ارزشمند نیست.
- زمان آموزش کوتاه تر: انتخاب زیرمجموعه دقیق تر از ویژگی ها، مدت زمان مورد نیاز برای آموزش یک مدل را کاهش می دهد.
- کاهش واریانس: دقت برآوردهایی را که می توان برای یک شبیه سازی معین به دست آورد افزایش می دهد.
- اجتناب از مزاحمت ابعاد بالای داده[1]: Curse of Dimensionality ماهیت انفجاری افزایش ابعاد داده و افزایش تصاعدی حاصل از آن در تلاش های محاسباتی مورد نیاز برای پردازش و یا تجزیه و تحلیل آن را توصیف می کند. در داده های با ابعاد بالا، تعداد ویژگی ها (یعنی ابعاد یا متغیرها) بسیار بیشتر از تعداد مشاهدات است. این منجر به چندین مشکل می شود:
  - پراکندگی: با تعداد ابعاد زیاد، احتمال زیادی وجود دارد که بسیاری از ابعاد داده اندک یا بدون داده باشند. این منجر به ماتریس های داده پراکنده می شود و یافتن روابط معنی دار بین ویژگی ها را دشوار می کند.
  - اندازه‌گیری‌های فاصله: فاصله اقلیدسی که معمولاً در بسیاری از الگوریتم‌ها استفاده می‌شود، در فضاهای با ابعاد بالا کم‌معنا می‌شود. این به این دلیل است که اکثر ابعاد احتمالاً متعامد هستند و تأثیر کمی بر فاصله کلی بین مشاهدات دارند.
  - بیش‌برازش: فضای با ابعاد بالا فرصت‌های زیادی را برای بیش‌برازش ایجاد می‌کند، جایی که یک مدل به خوبی با داده‌های آموزشی مطابقت دارد و به خوبی به داده‌های جدید تعمیم نمی‌یابد.

[1] curse of high dimensionality

بخش دوم

ایده اصلی fisher's score 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 یک کلاس تا حد امکان کوچک است. به طور خاص، با توجه به m ویژگی داده شده، ماتریس داده ورودی $X \in R^{d\times n}$ به $Z\in R^{m\times n}$ کاهش می‌یابد. سپس fisher's score به صورت زیر محاسبه می شود:

$$F(\mathbf{Z}) = \mathrm{tr}\left\{(\tilde{\mathbf{S}}_b)(\tilde{\mathbf{S}}_t+\gamma\mathbf{I})^{-1}\right\}$$

که γ یک پارامتر منظم‌سازی مثبت است، S_b~ ماتریس پراکندگی بین کلاسی نامیده می‌شود و S_t~ ماتریس پراکندگی کل نامیده می‌شود و به صورت زیر تعریف می‌شوند:

$$\tilde{\mathbf{S}}_b = \sum_{k=1}^{c} n_k(\tilde{\boldsymbol{\mu}}_k-\tilde{\boldsymbol{\mu}})(\tilde{\boldsymbol{\mu}}_k-\tilde{\boldsymbol{\mu}})^T$$

$$\tilde{\mathbf{S}}_t = \sum_{i=1}^{n}(\mathbf{z}_i-\tilde{\boldsymbol{\mu}})(\mathbf{z}_i-\tilde{\boldsymbol{\mu}})^T,$$

که در آن μ_k~ و n_k به ترتیب میانگین بردار و اندازه کلاس k ام در فضای داده کاهش‌یافته هستند. یعنی Z~μ، $\sum_{k=1}^{c} n_k\mu_k$~= بردار میانگین کلی داده‌های کاهش‌یافته است. از آنجایی که S_t~ معمولاً singular است، یک عبارت آشفتگی γI اضافه می کنیم تا آن را semi-definite کنیم.

از آنجایی که $(d\ m)$ نامزد Z از X وجود دارد، مسئله انتخاب ویژگی یک مسئله بهینه سازی ترکیبی و بسیار چالش برانگیز است. برای کاهش دشواری، استراتژی اکتشافی پرکاربرد، محاسبه امتیاز برای هر ویژگی به طور مستقل طبق معیار F است. به عبارت دیگر، فقط $X_j \in R^{1\times n}$ را در نظر می گیرد. در این حالت، فقط $(d\ 1) = d$ نامزدها وجود دارد. به طور خاص، فرض کنید μ^j_k و σ^j_k میانگین و انحراف معیار کلاس k و مربوط به ویژگی j باشد و همچنین μ^j و σ^j میانگین و انحراف معیار کل مجموعه داده مربوط به ویژگی j را نشان دهند. در این صورت fisher's score ویژگی j ام مطابق فرمول زیر محاسبه می شود:

$$F(\mathbf{x}^j) = \frac{\sum_{k=1}^{c} n_k(\mu_k^j-\mu^j)^2}{(\sigma^j)^2},$$

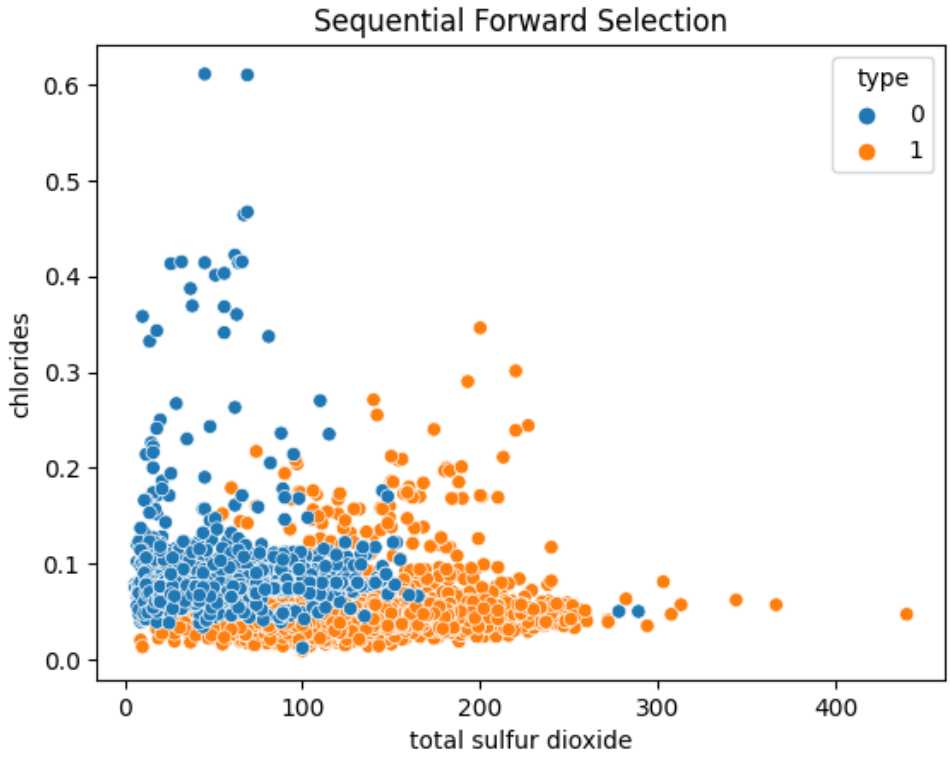
$$(\sigma^j)^2 = \sum_{k=1}^{c} n_k(\sigma_k^j)^2$$

پس از محاسبه fisher's score برای هر ویژگی، ویژگی هارا برحسب امتیاز آنها مرتب و m ویژگی دارای رتبه برتر را با امتیازهای بزرگتر انتخاب می‌کند.

سوال سوم

بخش اول : Sequential Forward Selection

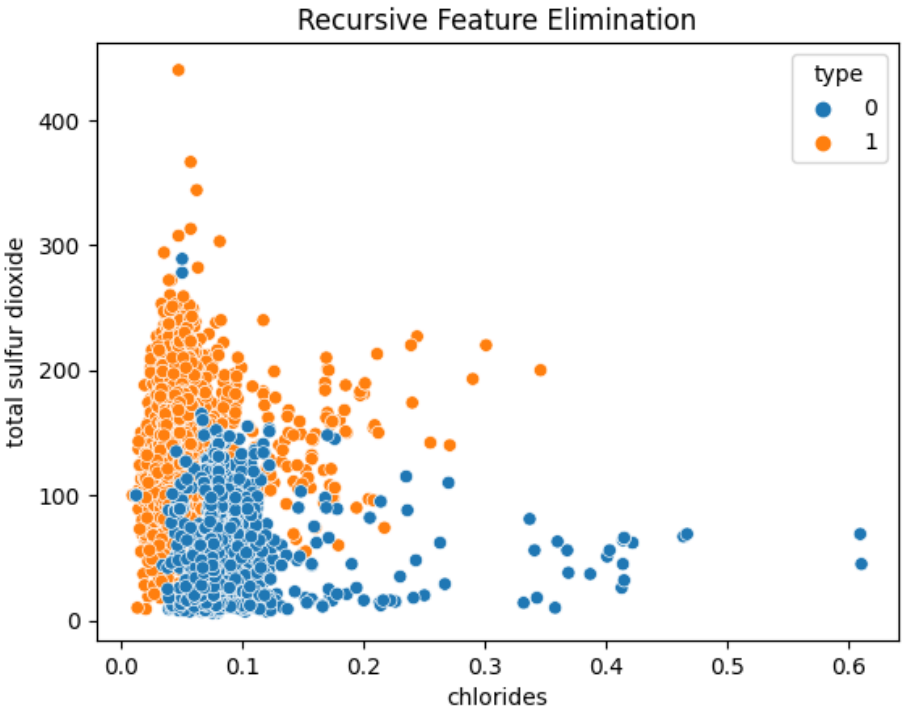
این الگوریتم اینگونه عمل می‌کند که در ابتدا از یک آرایه خالی از ویژگی‌ها(بهترین ویژگی‌ها) شروع کرده و سپس از میان ویژگی‌های باقی مانده، ویژگی را انتخاب می‌کند که بیشترین دقت را در طبقه بند به ما می‌دهد.
در مراحل بعد از میان ویژگی‌های باقی مانده باید ویژگی انتخاب شود که بیشترین دقت درصورت انجام طبقه‌بندی با این دو ویژگی را داشته باشیم و به همین صورت ادامه می‌دهیم تا به حداکثر تعداد دلخواه ویژگی و یا به یک آستانه‌ای از طبقه بندی برسیم.



طبقه‌بند مورد استفاده در این الگوریتم برای انتخاب بهترین ویژگی KNN است.
که در نهایت دو ویژگی total sulfur dioxide و chlorides انتخاب شده است در مدت زمان 1303.045 میلی ثانیه.

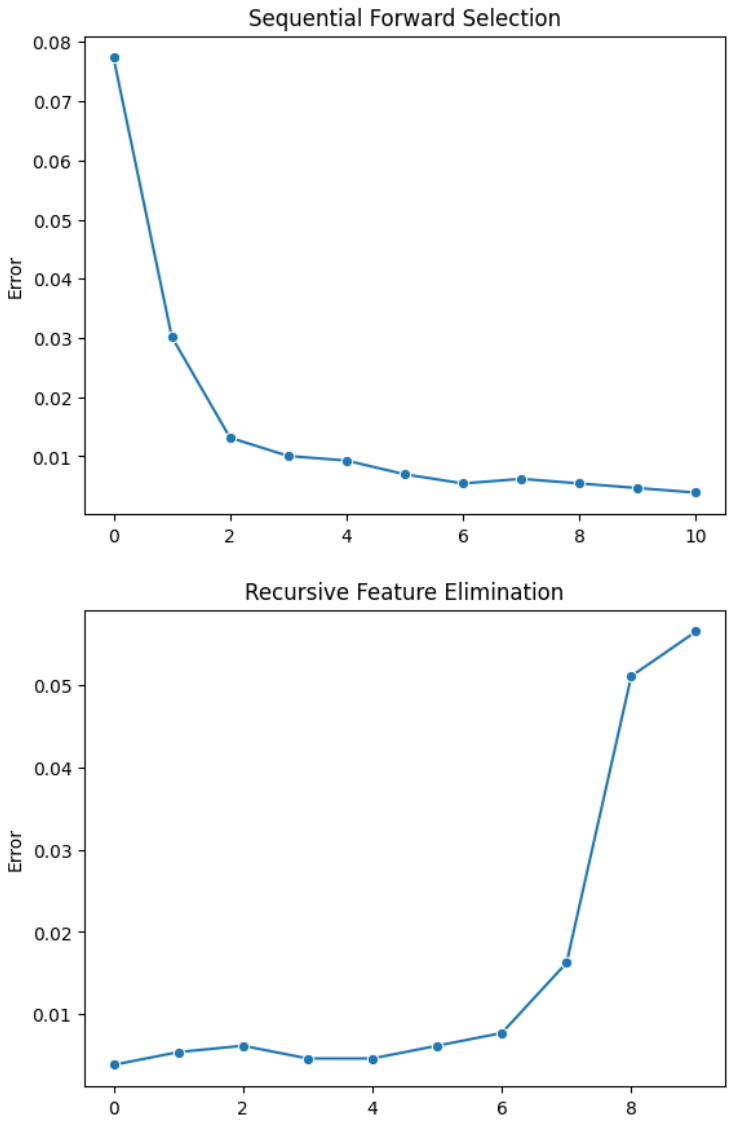
بخش دوم Recursive Feature Elimination

این الگوریتم از یک دیتاست کامل شروع می‌کند که شامل همه ویژگی‌هاست و درنهایت آن ویژگی را حذف می‌کند که با حذف آن ویژگی ما کمترین کاهش دقت را دارا باشیم و در نهایت تا جایی حذف می‌کنیم که به یک تعداد معین ویژگی یا به یک حداقل دقت برسیم به طوریکه با حذف یک ویژگی بیشتر دقت نهایی ما از یک آستانه‌ای کمتر شود.



طبقه‌بند مورد استفاده در این الگوریتم برای انتخاب بهترین ویژگی KNN است.
که در نهایت دو ویژگی total sulfur dioxide و chlorides انتخاب شده است در مدت زمان 1911.556 میلی ثانیه.
توجه شود که به طور اتفاقی این دو ویژگی در هر دو الگوریتم یکسان شده و الزاما این دو روش یکسان عمل نمی‌کنند.
و مدت زمان الگوریتم دوم همان‌طور که مشاهده می‌کنید خیلی بیشتر از الگوریتم اول است چرا که با تعداد ویژگی بیشتری هر بار طبقه‌بندی صورت می‌گیرد.

بخش سوم



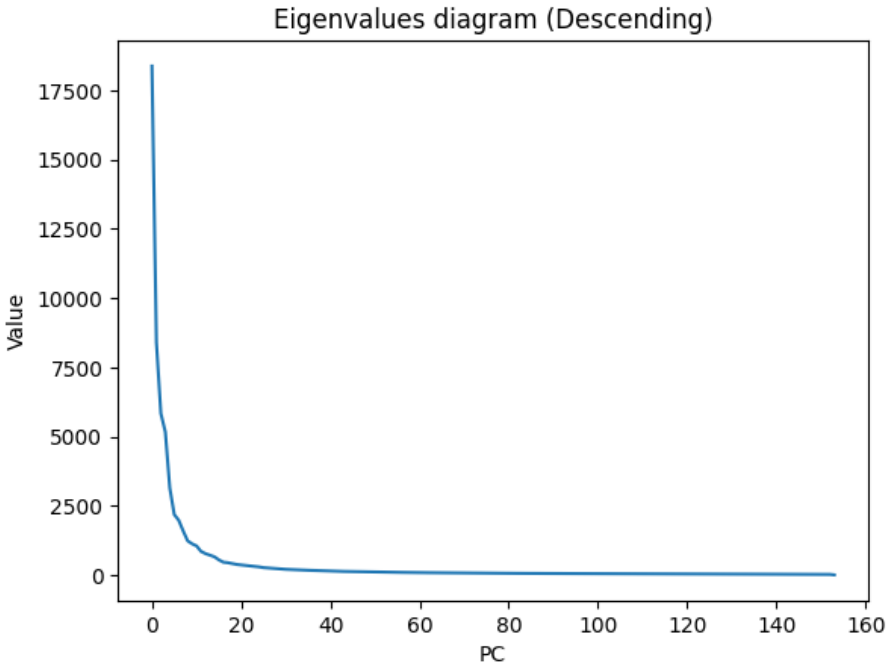
دقت الگوریتم هر دو الگوریتم به علت اینکه در این مسئله خاص یکسان شده است در نهایت مانند هم است اما در حالت کلی معمولا روش backward elimination بهتر عمل می‌کند و forward selection خیلی حریصانه‌تر عمل می‌کند.
همانطور که ذکر شده سرعت الگوریتم forward selection تقریبا دو برابر روش دوم است و از این جهت بهینه‌تر عمل می‌کند.

سوال چهارم

به علت حجیم بودن داده‌هایی از جنس تصویر می‌توان گفت با زیاد شدن دیتا ما دچار پدیده‌ی نحسی ابعاد می‌شویم به طوری که فرض کنید در یک دیتاست کوچک شامل ۱۰۰ عکس ۲۵۰ * ۲۵۰ ابعاد بردار ویژگی نزدیک به ۶۴ برابر ۱۰۰ خواهد شد در نتیجه اگر ما بتوانیم با روش‌های موثر کاهش بعد، بتوانیم داده‌ها را به فضایی با ابعاد کمتر برسانیم به طوری که بیشترین اطاعات حفظ شود، هم در انجام محاسبات سرعت ما افزایش میابد و هم به دقت بالاتری خواهیم رسید.
توجه به این نکته نیز ضروری است که در تصویر نقاط معمولا ویژگی‌های نزدیک بهم را دارند درنتجه با کاهش بعد خیلی زیاد ما اطلاعات زیادی که تکراری است را از دست خواهیم داد و در ازا تنها اطلاعات مفید در جهت حل مسئله را دارا خواهیم بود.

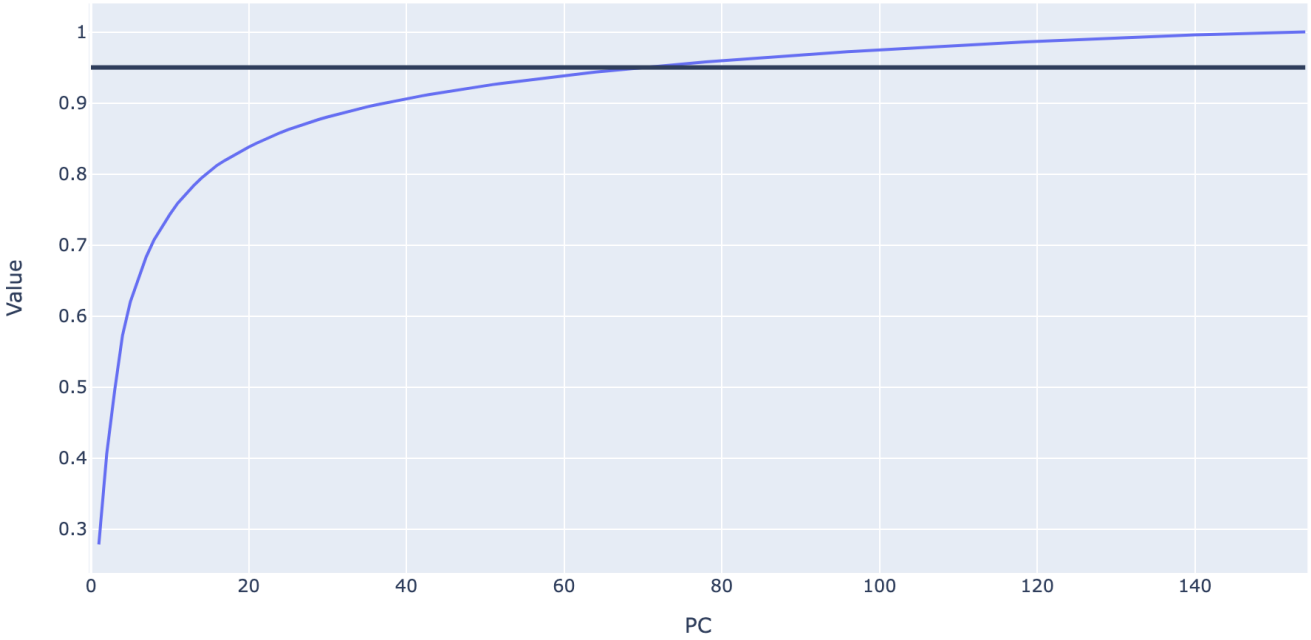
بخش اول

در ابتدا تمام داده‌های آموزش را استاندارد می‌کنیم تا در همه جهت‌ها به یک میزان باشند و دقت در نهایت امر به اسکیل داده‌ها وابسته نباشد.



همان‌طور که در شکل می‌بینیم حدود ۹۰ درصد اطلاعات در ۷۲ مولفه اول داده‌ها موجود است و در واقع مابقی جهت‌ها اطلاعات اضافی به مدل ما اضافه نخواهد کرد.

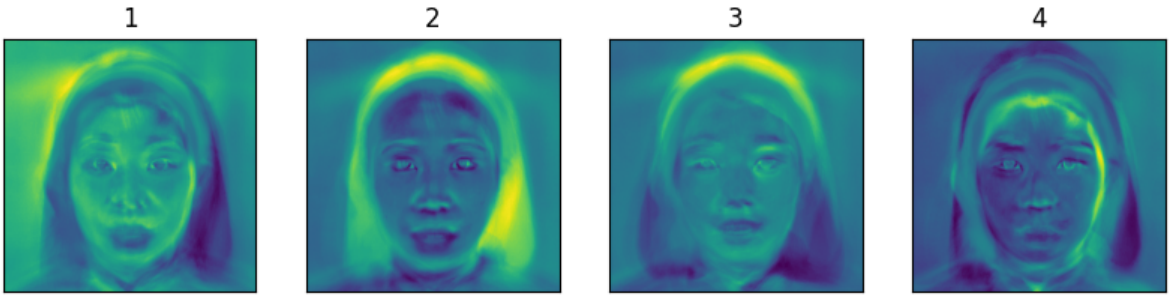
در شکل زیر می‌توانید مقدار دقیق آن‌را مشاهده نمایید.



یکی از راه‌های انتخاب تعداد مولفه‌ها آن است که ما می‌خواهیم چند درصد اطلاعات حفظ شود.
یا آنکه چه مولفه‌هایی از یک آستانه‌ای کمتر اطلاعات دارند و نزدیک به صفر است مقادیر پراکندگی آن مولفه‌ها.

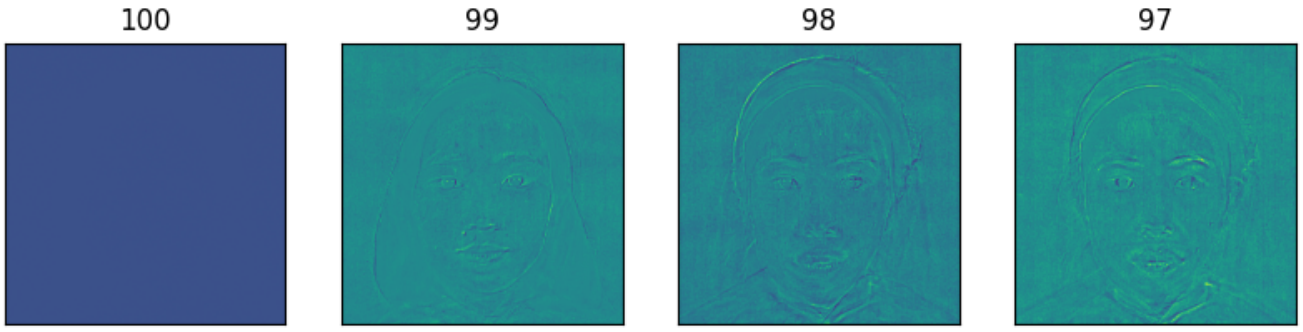
بخش دوم

چهار مولفه اول - مولفه‌هایی که در آن فضا ما بیشترین اطلاعات تصاویر را داریم.



تا حد زیادی با تنها حفظ یک جهت و یک مولفه از داده همانطور که مشاهده میشود بخش زیادی از اطلاعات حفظ شده و قابل تشخیص هستند تصاویر تا حد زیادی.

چهار مولفه آخر



مولفه‌هایی که کمترین میزان اطلاعات از داده‌ی ابتدایی را شامل می‌شوند و همانطور که مشاهده می‌شود شامل اطلاعات زیادی نمی‌باشد و با حذف این مولفه ما در مجموع اطلاعات زیادی را از دست نخواهیم داد.

سوال ششم

بخش اول

خیر. الگوریتم‌های مبتنی بر فاصله‌ای مانند k-means بیشتر مناسب زمانی‌اند که خوشه‌ها شکل ساده‌ای داشته باشند و تا حدی هم‌سایز باشند و در شرایطی که خوشه‌ها اشکال پیچیده‌تر مانند مارپیچ داشته باشند یا دارای اندازه‌های متفاوتی باشند، به اندازه‌ی کافی کارآمد نیستند و خوشه‌بندی را به درستی انجام نمی‌دهد. علاوه بر این موضوع، الگوریتم‌های مبتنی بر فاصله به نقاط پرت حساسند و یک نقطه‌ی پرت می‌تواند تأثیر زیادی بر خوشه‌های حاصل داشته باشد و در صورت حضور تعداد قابل توجهی نقطه‌ی پرت در داده‌ها، احتمالا این روش از خوشه‌بندی مناسب نخواهد بود. در این شرایط بهتر است به سراغ الگوریتم‌های مبتنی بر چگالی برویم.

بخش دوم

الگوریتم DBSCAN که مخفف عبارت Density Based Spatial Clustering of Applications with Noise است یکی از معروف‌ترین الگوریتم‌های خوشه‌بندی مبتنی بر چگالی به شمار می‌رود که نیازی هم به تعیین تعداد خوشه‌ها ندارد و خود الگوریتم می‌تواند با توجه به تراکم، خوشه‌ها را شناسایی کند. پیش از تشریح روش کار این الگوریتم، ابتدا به چند مقدمه می‌پردازیم؛ الگوریتم DBSCAN به دو پارامتر حساس است که می‌توانند توسط کاربر تعیین شوند: شعاع (Epsilon) و حداقل تعداد نقاط موجود در شعاع (MinPts) در صورتی که تعداد نقاط هم‌جوار نقطه‌ی مورد بررسی که در شعاع تعیین‌شده قرار دارند، به تعداد MinPts برسد، نقطه‌ی فعلی یک نقطه‌ی Core یا هسته محسوب می‌شود. همچنین نقاط دیگری که در همسایگی یک نقطه‌ی Core قرار دارند اما تعداد همسایه‌های آن‌ها به MinPts نمی‌رسد، نقاط Border یا مرزی خوانده می‌شوند. در نهایت به نقاطی که نه Core و نه مرزی باشند، Outlier یا پرت گفته می‌شود.

حال به روش خوشه‌بندی در این الگوریتم می‌پردازیم. در این الگوریتم، ابتدا نقطه‌ای به صورت تصادفی انتخاب شده و به اولین خوشه نظیر می‌شود. در صورتی که این نقطه یک نقطه‌ی Core باشد، تمام نقاط موجود در شعاع آن نیز به این خوشه نظیر می‌شوند. پس از آن نقطه‌ی دیگری در همسایگی نقطه‌ی فعلی انتخاب شده و این پروسه تکرار می‌شود و خوشه گسترش می‌یابد. لازم به ذکر است نقاط مرزی، تنها می‌توانند به خوشه ملحق شوند و نمی‌توانند خوشه را گسترش دهند؛ بعبارتی اگر نقطه‌ی مورد بررسی یک نقطه‌ی مرزی باشد، نقاط هم‌جوار آن به خوشه ملحق نمی‌شوند بلکه تنها خود این نقطه‌ی مرزی به خوشه نظیر می‌شود و برای ادامه‌ی کار نقطه‌ی دیگری انتخاب می‌شود. این روند تا جایی ادامه پیدا می‌کند که نقطه‌ی واجد شرایط دیگری وجود نداشته باشد. در این صورت خوشه تکمیل شده و در ادامه، نقطه‌ی تصادفی دیگری که متعلق به خوشه‌ی خاصی نباشد، تعیین و روند مشابهی برای آن طی می‌شود تا سرانجام تمام نقاط بررسی شوند. همچنین باید گفت این الگوریتم نقاط پرت را به هیچ خوشه‌ای نظیر نمی‌کند.
واضح است که هر چه شعاع کوچک‌تر در نظر گرفته شود، خوشه‌های بیشتر و کوچک‌تری تشکیل می‌شوند و هر چه تعداد MinPts بزرگ‌تر لحاظ شود، احتمال ایجاد خوشه‌ها کاهش می‌یابد.

تفاوت با الگوریتم OPTICS

همان‌طور که پیش‌تر گفتیم، DBSCAN به دو پارامتر شعاع و MinPts حساس است. الگوریتم OPTICS که کوتاه‌شده‌ی عبارت Ordering Points To Identify the Clustering Structure است نیز یکی دیگر از روش‌های خوشه‌بندی مبتنی بر چگالی است که تا حد خوبی مشابه DBSCAN عمل می‌کند. تفاوت این الگوریتم با DBSCAN در اینست که در این الگوریتم سعی شده حساسیت به پارامترها کم شود و ساختاری برای ترتیب خوشه‌بندی نقاط ارائه گردد.

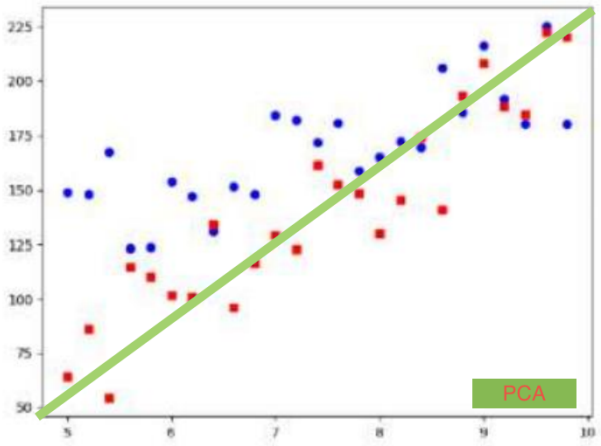
برای پیاده‌سازی این تکنیک به محاسبه‌ی دو مورد زیر نیاز است:

- Core distance: برای نقطه‌ی p به کم‌ترین فاصله‌ای که در شعاع آن، حداقل تعداد MinPts نقطه حضور داشته باشند، Core distance گوییم.

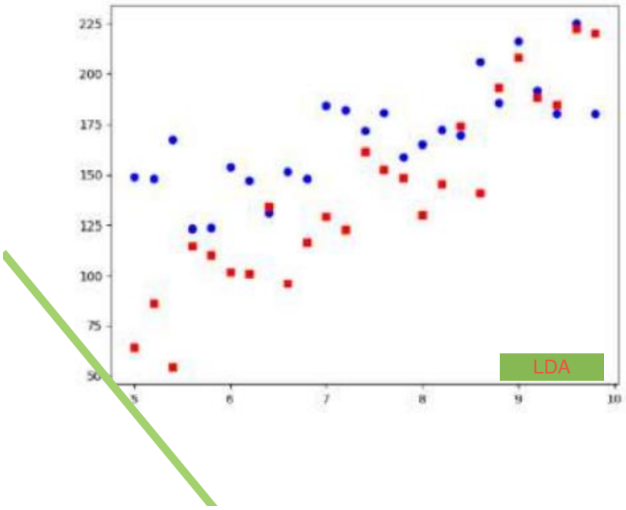
- Reachability distance: به کم‌ترین فاصله‌ی بین نقطه‌ی p و نقطه‌ی q، که باعث می‌شود نقطه‌ی p از q قابل دسترسی باشد، reachability distance گوییم. در صورتی که q یک نقطه‌ی core نباشد، این فاصله «تعریف‌نشده» لحاظ می‌شود. و در غیر اینصورت ماکزیمم مقدار بین Core distance مربوط به نقطه‌ی q و فاصله‌ی بین q و p بعنوان reachability distance انتخاب و ذخیره می‌شود.

این الگوریتم به دلیل لزوم محاسبه‌ی فواصل ذکرشده، به حافظه و قدرت محاسباتی بیشتری نسبت به DBSCAN نیاز دارد و نهایتاً برخلاف DBSCAN ترتیبی از پردازش داده‌ها برای خوشه‌بندی ارائه می‌دهد که بر مبنای آن می‌توان برای خوشه‌بندی تصمیم بهتری گرفت. همچنین این الگوریتم نسبت به DBSCAN به پارامترها مقاوم‌تر است و چندان نیازی به حفظ پارامتر شعاع ندارد.

سوال هفتم



PCA جهتی را انتخاب می‌کند که داده‌ها در آن بیشترین پراکندگی را دارد بنابراین همان‌طور که در شکل نشان داده شده است خط سبز رنگ جهتی است که در آن دیتا بیشترین پراکندگی دارد به عنوان مولفه نخست و مابقی جهت‌ها درصورت نیاز عمود بر این جهت انتخاب می‌شود.
توجه شود باتوجه به اینکه PCA بدون در نظر گرفتن لیبل داده‌ها کاهش بعد را اعمال می‌کند و صرفا براساس پراکندگی آن‌ها عمل می‌کند در نتیجه همانند شکل بالا ممکن است اشتباه عمل کرده چرا که در این دیتا جهتی که بیشترین پراکندگی را دارد، جهت درست برای جداکردن دیتا در دو دسته نمی‌باشد.



LDA یک الگوریتم نظارت شده است بدین صورت که با در نظر گرفتن لیبل داده‌ها جهتی را انتخاب می‌کند که در آن کلاس‌ها در آن بیشترین جدایی پذیری را از هم دارند.
همان‌طور که مشاهده می‌شود خط سبز رنگ محوری است که در آن دو کلاس تا حد خوبی از هم تفکیک شده‌اند هر چند خطایی وجود دارد اما آن به ذات دیتا برمی‌گردد. بنابراین LDA به وضوح در این مسئله به خوبی عمل کرده است.