**سوال سوم:**

بخش اول :

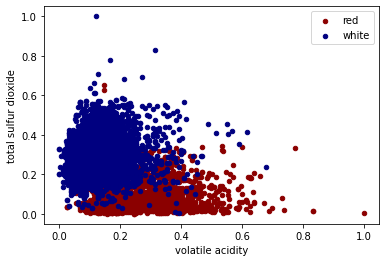
با توجه به صورت سوال دو الگوریتم یادشده را پیاده سازی کردیم. روش Sequential Feature Selection که یکی از روش های متداول feature selection است از دو بخش کلی تشکیل شده است:

1- یک تابع هدف-criterion- که به دنبال به حداقل رساندن آن در تمام زیرمجموعه های ممکن است. معیار رایج برای مدل های رگرسیون MSE و برای مدل های طبقه بند نرخ طبقه بندی نادرست است.

2-یک الگوریتم جستجوی ترتیبی که ویژگی ها را در حال ارزیابی تابع هدف به زیرمجموعه اضافه یا کم میکند. از آنجایی که جستجو/ مقایسه (search) exhaustive تابع هدف تمام زوج ویژگی های ممکن از یک مجموعه n-عضوی عملا غیرممکن است. پس از یک الگوریتم جستجوی ترتیبی که فقط در یک جهت کوچک یا بزرگ میشود استفاده میکنیم.

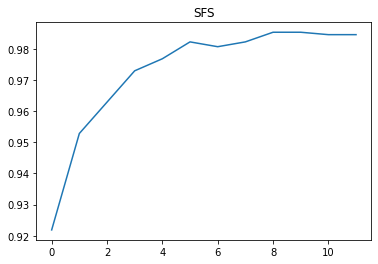
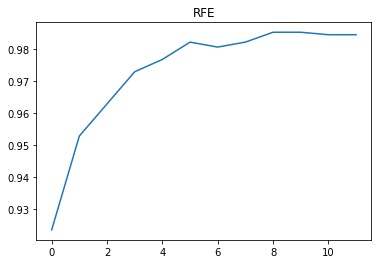
در sfs\_sequnetial forward selection ویژگی هاـfeature- به صورت ترتیبی به یک زیرمجموعه تهی(در آغاز) افزوده میشود تا اینکه افزوده شدن ویژگی جدید باعث کاهش تابع هدف نشود.(در اینجا ما در عمل شروع به train دادگان موجود میکنیم تا در هر بار افزوده شدن ان ویژگی که باعث افزایش دقت میشود را به زیر مجموعه اضافه میکنیم.)

و در sfs\_sequnetial backward selection ویژگی ها در ابتدا همگی در یک مجموعه قرار دارند و ما به صورت ترتیبی شروع به حذف ویژگی ها میکنیم تا جایی که حذف جدید باعث کاهش تابع هدف نشود.(در این بخش ما در ابتدا ویژگیها را شروع به حذف میکنیم و ان feature را انتخاب میکنیم که حذفش باعث افزایش دقت شود.) در ادامه شکل خروجی طبق بند اورده شده است:

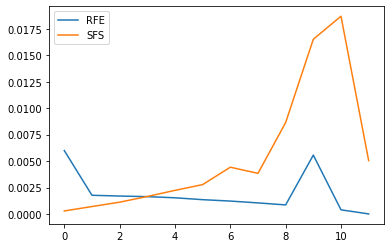


بخش سوم:

در اینجا برای بررسی دقت از روش logistic regression مبتنی بر تعداد ویژگی استفاده میکنیم تعداد ویژگی برای سوال از 1-12 تغییر میکند که بر اساس مقدار دقت رسم شده است به طور کلی نتایج هر دو الگوریتم از نظر دقت یکسان شد.



اما از نظر پیچیدگی زمانی الگوریتم sfs پیچیده تر است که در ادامه شکل های مربوط به دقت هر دو الگوریتم و سرعت(زمان عملکرد) آورده شده است.



ــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ

‌**سوال 4:**

**0-یک از روشهای معمول برای کاهش بعدـPCA/principle component analysis میباشد. اگر ما با تعداد ویژگی های زیاد سروکار داشته باشیم معمولا این حتمی است که اگر همه ی ویژگی ها را مورد بررسی قرار دهیم 1- اولا حافظه و زمان را تلف کردیم. 2- ثانیا به علت اینکه به دادگان train وابستگی شدید پیدا کردیم مدل overfit خواهد شد . برای جلوگیری از این موضوع میتوان از PCA استفاده کرد. در اینجا هدف ما پیدا کردن جهتی است که بیشترین تغییرات داده(بیشترین میزان اطلاعات) در آنجا قرار میگیرد. سپس با تصویر کردن داده ها روی جهت یاد شده ابعاد کاهش می‌یابد. این الگوریتم را میتوان به صورت تکراری-iterative- ادامه داد. از کاربردهای این روش میتوان به استخراج ویژگی/حالت تصاویر اشاره کرد. با کاهش ابعاد دیتاست تصاویر‍‍‍‍‌ تعداد ویژگی های کاربردی کمتر میشود. مثلا اگر PCA-0.9 اعمال کنیم به این معناست که کاهش بعد تا جایی صورت میگیرد که 90٪ واریانس داده ها را شامل شود. بعلاوه اگر این پارامتر را به صورت عددی بزرگتر از یک بیان کنیم به این معناست که ابعاد تا آن مقدار کاهش می‌یابد.**

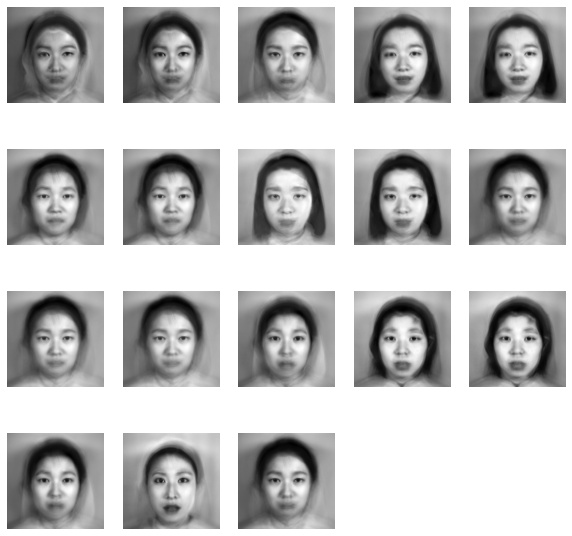
1. **برای یافتن بعد های حذف شده روشهای گوناگونی وجود دارد. در یکی از آن ها ابتدا پراکندگی داده ها را در ابعاد جدید محاسبه و براساس مقدار پراکندگی به دست آمده برای هر بعد ابعاد جدید مرتب میشود. پراکندگی تجمعی با شروع از بعد نخست به صورت ترتیبی برای بقیه ابعاد حساب میشود. وقتی به 80-90 درصد رسیدیم الگوریتم را متوقف میکنیم و ابعادی که پراکندگی آن ها لحاظ نشده را حذف میکنیم.**
2. **در این بخش در ابتدا دادگان اصلی سپس4 مقدار ویژه اول و نهایی داده بعد از اعمال PCA 90 درصد را نشان میدهیم.**

**تصویر دادگان اصلی به صورت زیر است:**

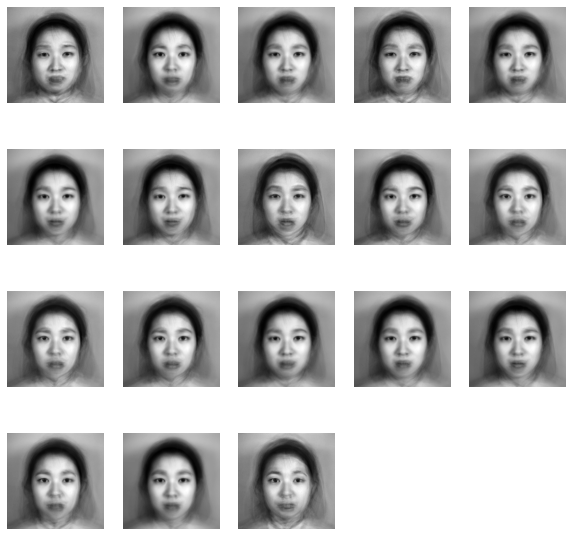
****

سپس 4 مقدار ویژه اول ترسیم شده است:

لازم به ذکر است برای حل این سوال از حالت disgust استفاده کردیم.



همانطور که دیده میشود تصاویر اندکی تارتر شده‌اند هرچند ارتباط با تصویر اصلی هنوز مشهود است.

در آخر نیز 4 مقدار نهایی رسم شده است در اینجا نیز به طور مشابه با حالت قبل هرچند ارتباط با داده اصلی مشهود است تصاویر اندکی تار گشته‌اند.

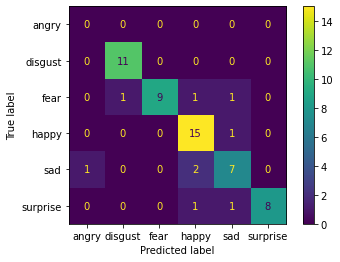
1. در الگوریتم knn فاصله داده(point) که میخواهیم پیشبینی کنیم را نسبت به دادگان train حساب کرده با توجه به k نقطه نزدیک به آن کلاس نسبت میدهیم. برای محاسبه فاصله معیارهای متفاوتی وجود دارد: 1- اقلیدسی 2- منهتن و...

میزان CCR و ماتریس آشفتگی به شرح زیر است:

"original data, nearest neighbors = 1"

CCR : 0.847457627118644

confusion matrix:

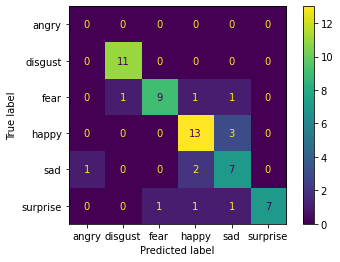


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

"PCA data, nearest neighbors = 1"

CCR : 0.7966101694915254

confusion matrix:

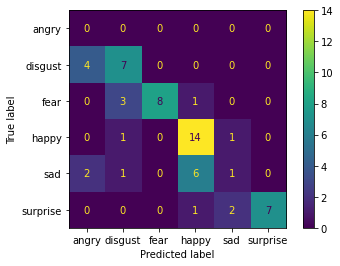


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

"original data, nearest neighbors = 2"

CCR : 0.6271186440677966

confusion matrix:

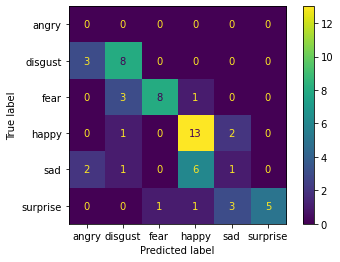


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

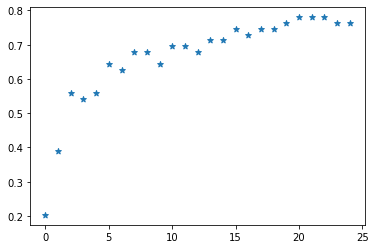
"PCA data, nearest neighbors = 2"

CCR : 0.5932203389830508

confusion matrix:



4-نمودار را برای تعداد 0-25 ویژگی رسم کرده ایم . همان طور که دیده میشود بعدا از تقریبا 20 ویژگی CCR تغییر گسترده ای نداشته است.

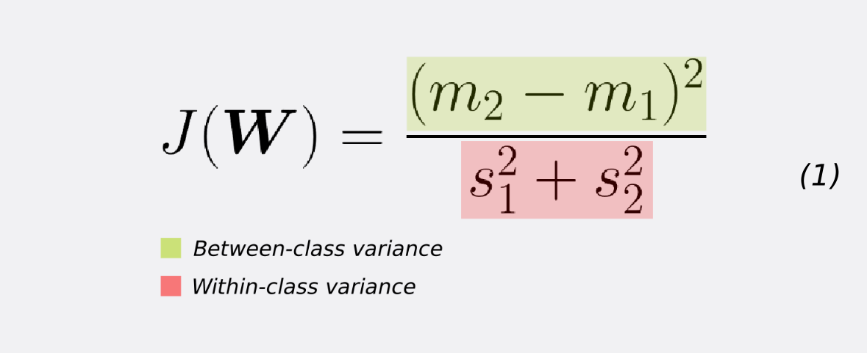


**سوال پنجم:**

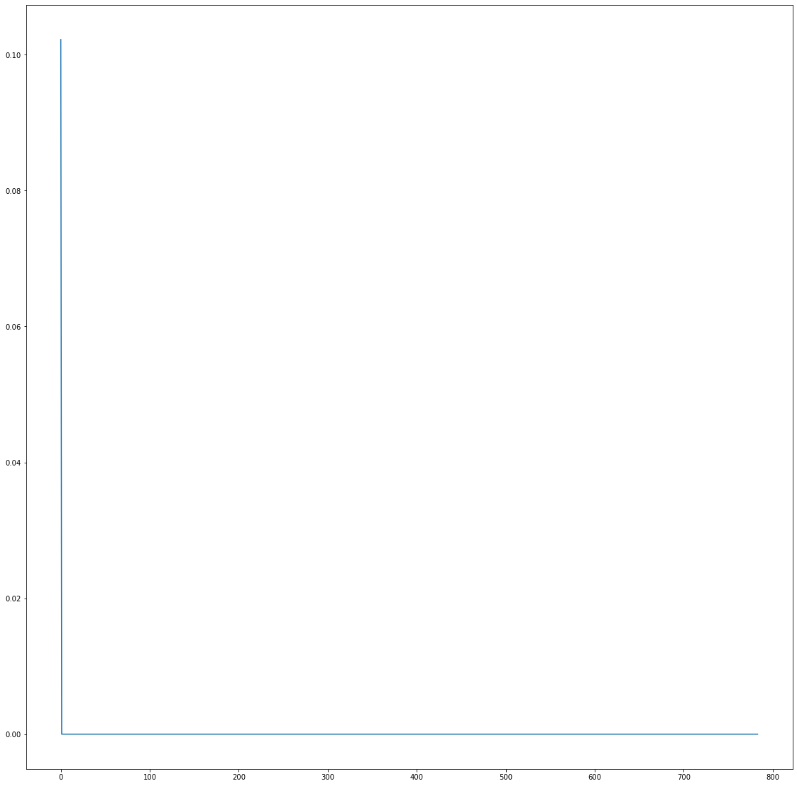
1. ماتریس پراکندگی بین کلاسی میزان اختلاف و پراکندگی بین کلاس های گوناگون را نشان میدهد(به عنوان معیار مرکز /هسته کلاس- میانگین در نظر گرفته میشود)اما ماتریس درون کلاسی انحراف داده ها از core کلاس است.

Sw = S1 + S2 + S3 +…+ Sn

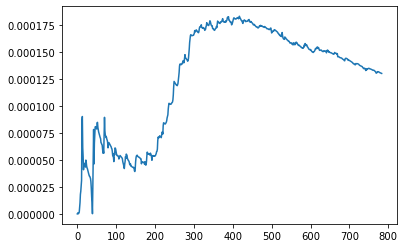
n تعداد کلاس ها و s ماتریس پراکندگی داده ها حول core کلاس است. در ماتریس بین کلاسی Sb همانطور که گفته شد فاصله بین coreها نشان داده میشود. در LDA معیار مورد نظر J(w) میباشد که باید ماکزیمم شود. wها عمان بردار وزن است رابطه آن به صورت زیر است:



1. بعد از محاسبه LDA در بخش اول ماتریس خواسته شده محاسبه شد و براساس تعداد ویژگی/مقدار ویژه نشان داده شده است.

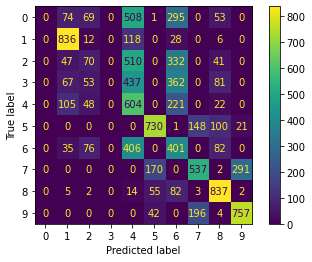


1. به طور کلی نمودار صعودی است و دارای برخی مینیمم‌های محلی است هرچه میزان مقادیر ویژه روند صعودی داشته باشد یعنی روند LDA مناسبتر انجام شده است و در جهت ماکزیمم کردن J عمل کرده است.

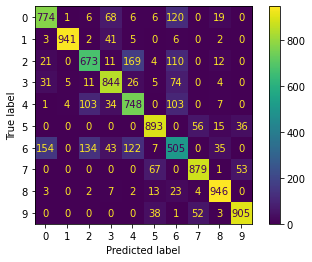


4و5- در ادامه ماتریس آشفتگی و CCR برای مدل naïve bayes در دو حالت عادی و LDA آورده شده است. میزان دقت(CCR) زمانی که از LDA استفاده میشود به مراتب بهتر شده است. با وجود اینکه این روش بیشتر برای این است که میزان دقت روی داده دیده نشده قابل قبول باشد و همچنین در حافظه مصرفی و زمان اجرا صرفه جویی شود.

original data :



LDA applied data :



0.47724772477247723

0.8108810881088109

7- توضیحات سوال در بخش نوشتاری اورده شده است در اینجا فقط به ترسیم جهت های PCA و LDA میپردازیم.

