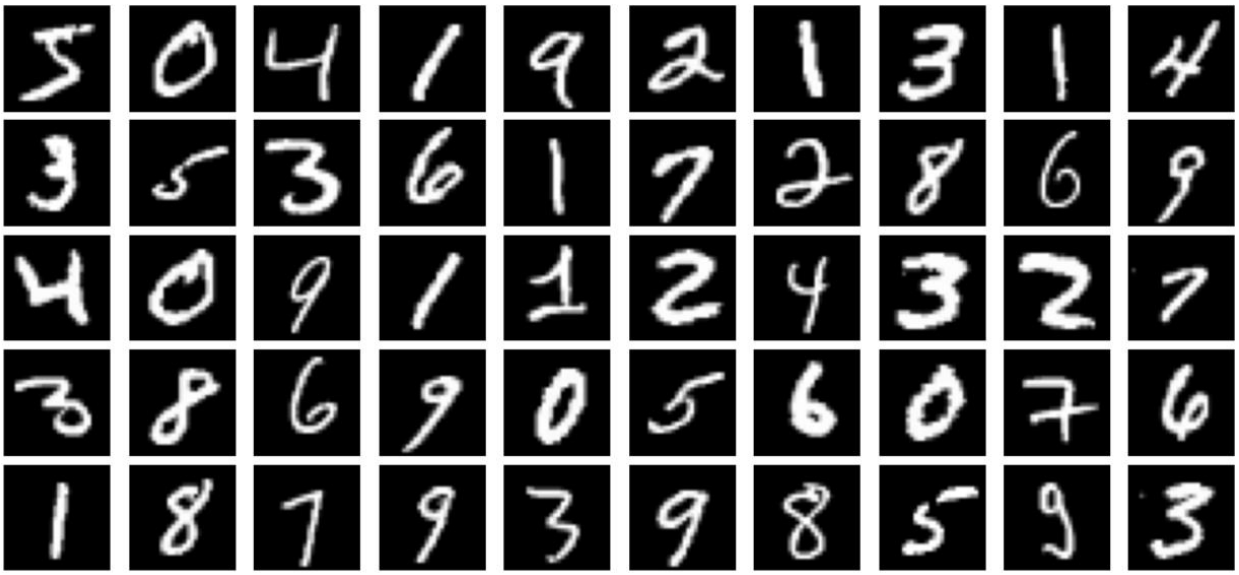
تمرین اول)

نمایش 50 تصویر اول دیتاست mnist :



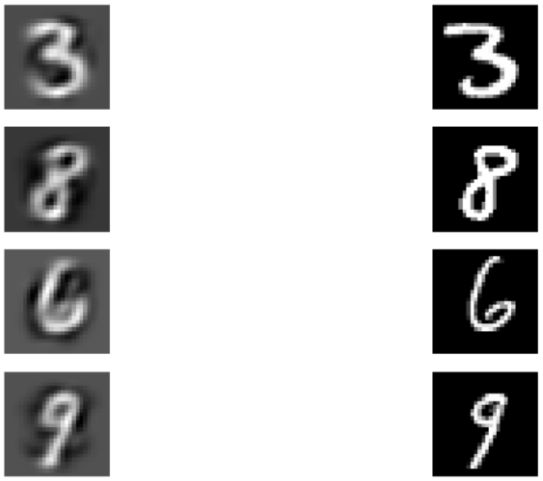
ابتدا دیتا را به داده های رتین و تست تقسیم می کنیم و بعد از reshape و normalize کردن دیتا ، ماتریس کوواریانس را تشکیل می دهیم و با دستور numpy.linalg.eig مقادیر و بردار های ویژه ی آن را می یابیم و مقادیر ویژه و بردار های ویژه ی متناظر آن را را به ترتیب نزولی سورت می کنیم .

سپس تعدادی از بزگترین مقدار های ویژه را انتخواب می کنیم تا مجموع آنان بزرگتر مساوی 0.7 ضربدر واریانس کل (total variance) شود.

که انتخواب 26 مولفه (component) برای آن کافی می باشد.

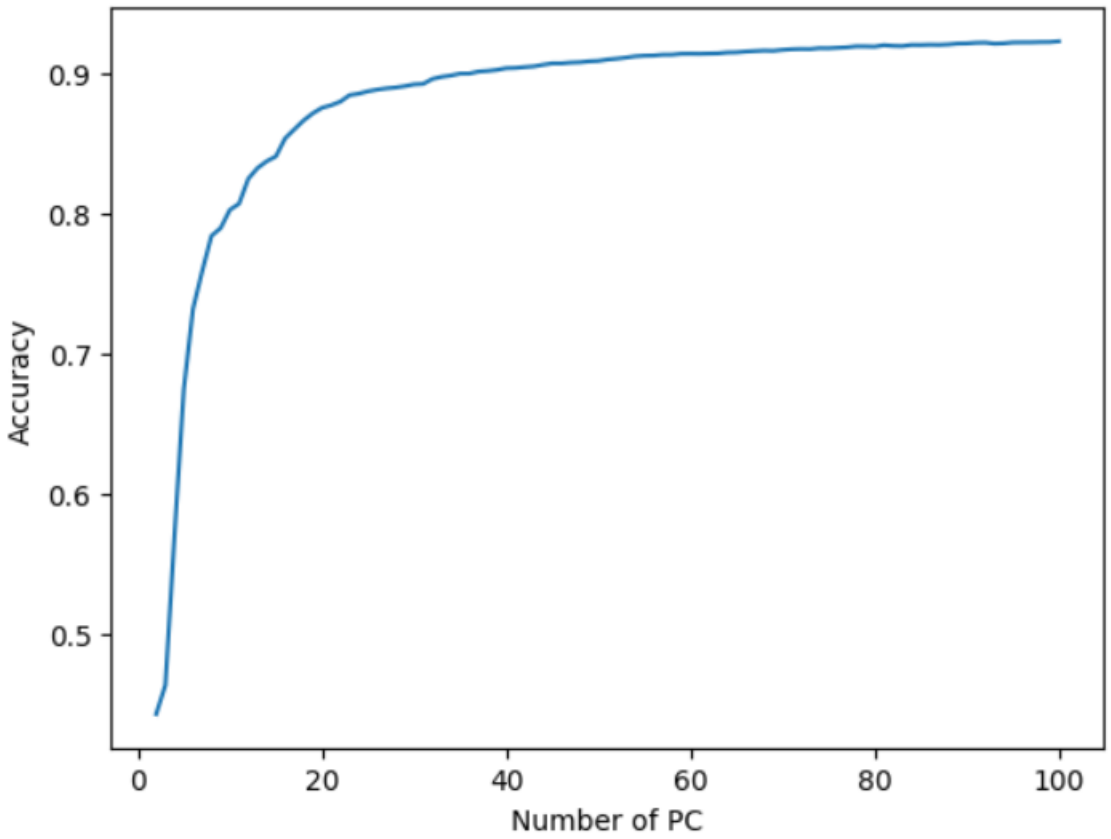
حال همانطور که در صورت سوال توضیح داده شده است، دیتا ی اصلی را بر روی ماتریس 26 بردار ویژه ی ماتریس کوواریانس نگاشت می دهیم و دوباره به پایه ی قبلی تصویر برمی گردانیم و تصویر را بازسازی می کنیم .

چند نمونه از مقایسه ی تصاویر اصلی و بازسازی شده با pc 26 :



در بخش بعد PCA را با استفاده از کتابخانه ی sklearn انجام دادیم و به ازای تعداد PC های مختلف ، Accuracy کلسیفیکیشن تصاویر مختلف به دست آورده شد. (از Logistic Regression برای کلسیفیکیشن استفاده شده است)

نمودار train accuracy بر حسب تعداد PC ها :



number of components that maximize the accuracy is 100 with accuracy 0.9225 .

تمرین دوم) Decision tree

الگوریتم ID3 یک الگوریتم یادگیری ماشین است که برای ساخت درخت تصمیم استفاده می‌شود. این الگوریتم بر اساس اطلاعات موجود در داده‌ها، بهترین ویژگی را برای تقسیم داده‌ها به دست می‌آورد. در این الگوریتم، ابتدا برای هر ویژگی معیار اطلاعات (Information Gain) محاسبه می‌شود. سپس ویژگی با بیشترین معیار اطلاعات انتخاب می‌شود و داده‌ها بر اساس آن ویژگی تقسیم می‌شوند. این فرآیند تا زمانی ادامه می‌یابد که همه داده‌ها در یک گروه قرار بگیرند یا دیگر ویژگی‌های قابل استفاده وجود نداشته باشند. در نهایت، درخت تصمیم حاصل از این الگوریتم می‌تواند برای پیش‌بینی و تصمیم‌گیری در موارد جدید استفاده شود.

زمانی که ویژگی ها از نوع کتگوریکال باشند، یک node به تعداد کتگوری های مانده در فیچر انتخواب شده expand می شود و به همین صورت درخت تشکیل می شود.

در این سوال چون با فیچر های numerical (نه کتگوریکال) سرو کار داریم ، هنگامی که می خواهیم یک نود را expand کنیم ، در ستون فیچر ،میانه ی آن را به عنوان threshold ست می کنیم و داده ها را بر اساس این ترشولد به دو قسمت تقسیم می کنیم (یعنی هر نود دارای دو child می باشد) و اکیورسی بدست آمده بر روی داده ی تست دیتاست mnist ، 81.71 می باشد.

توجه : همچنین می توانیم به جای انتخاب میانه به جای threshold ،، آن را learn کنیم ، یعنی با ترشولد های مختلف Entropy gain را محاسبه کنبم و ماکزیمم آن را انتخاب کنیم. این کار اکیورسی را حدود 2 تا 3 درصد افزایش می دهید ولی time complexity الگوریتم بسار بالا خواهد رفت ، به نحوی که برای این دیتا با انتخاب میانه به عنوان threshold ، اجرای الگریتم حدود 2 دقیقه طول می کشد و با learn کردن threshold حدود 3 ساعت ، فیت شدن آن طول خواهد کشید.

تمرین سوم) Classification with SVM

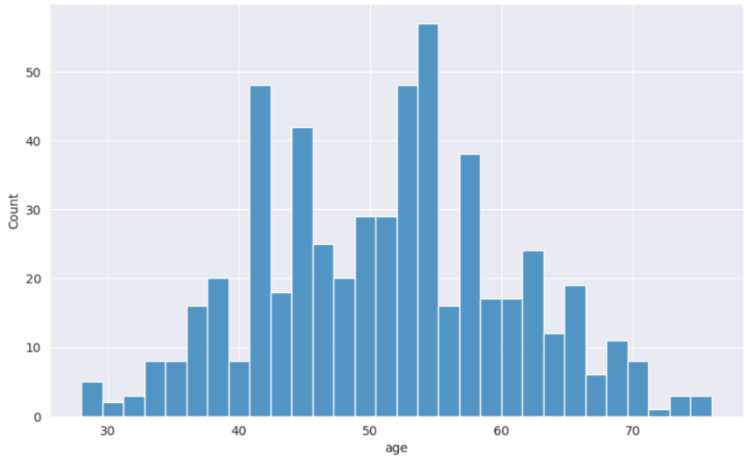
ابتدا دیتاست را لود می کنیم.

- این دیتا ست دارای 11 فیچر (بدون در نظر گرفتن target) و 1190 نمونه می باشد.
- سپس وجود missing value هارا بررسی می کنیم که مشاهده شد که هیچ missing value ای در دیتاست وجود ندارد.
- حال توضیع داده های در کلاس 0 و 1 را بررسی می کنیم که مشاهده می شود دیتاست تا حد قابل قبولی بالانس می باشد و درصد توضیع داده ها در کلاس ها به صورت زیر می باشد :

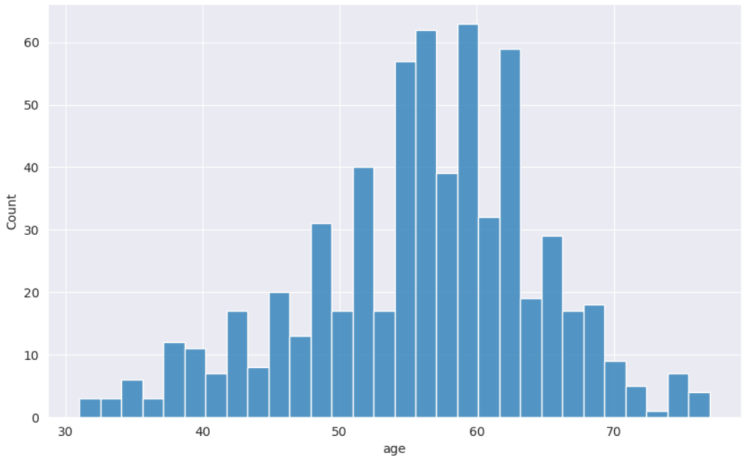
$$class\ 1 : 629\ samples\ (52.8\ \%)$$
$$class\ 0 : 561\ samples\ (47.1\ \%)$$

- نمودار توضیع سن و جنسیت کلاس های مختلف :

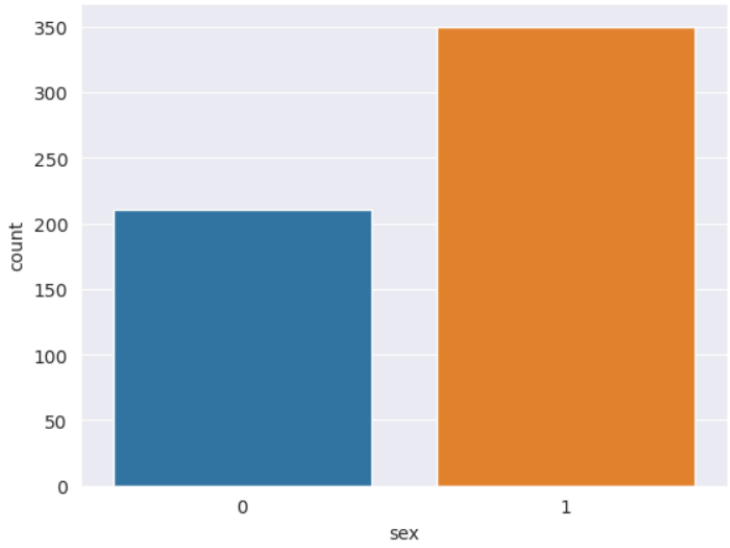
age histogram of class 0:



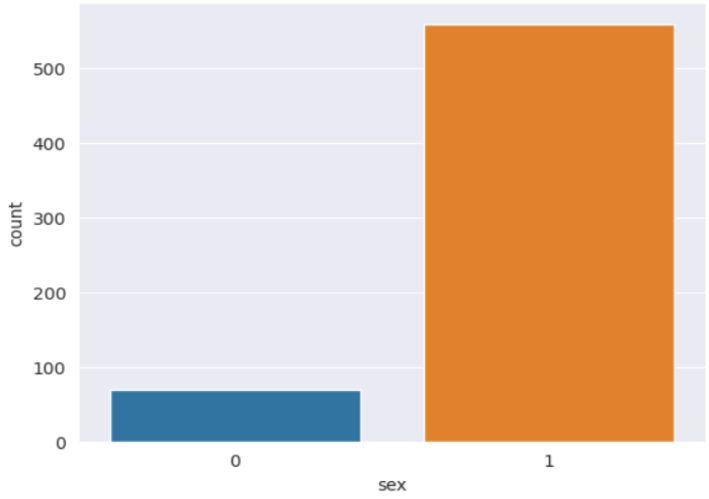
age histogram of class 1:



barplot of sex feature for class 0:



barplot of sex feature for class 1:



- حال در فیچر های numerical ، سمپل هایی که outlier هستند را با استفاده از Z-Test و با ترشولد std*3 حذف می کنیم.
- و در نهایت بعد از نرمالایز کردن فیجر های numerical ، SVM را بر روی داده ی ترین فیت می کنیم و با استفاده از gridsearch هایپرپارامتر های بهینه را می یابیم .

for Linear kernel :
Accuracy: 0.833810888252149
Precision: 0.8059701492537313
Recall: 0.8950276243093923
F1 Score: 0.8481675392670157

for RBF kernel :
Accuracy: 0.8510028653295129
Precision: 0.8208955223880597
Recall: 0.9116022099447514
F1 Score: 0.8638743455497382

for Polynomial kernel :
Accuracy: 0.8653295128939829
Precision: 0.8722222222222222
Recall: 0.8674033149171271
F1 Score: 0.8698060941828255