سؤال 1

## فرآیند آموزش و ارزیابی یک مدل طبقه بند خطی را به صورت دیاگرامی بلوکی نمایش دهید و در مورد اجزای مختلف این دیاگرام بلوکی توضیحاتی بنویسید. تغییر نوع طبقه بندی از حالت دوکلاسه به چندکلاسه در کدام قسمت از این دیاگرام بلوکی تغییراتی ایجاد می کند؟ توضیح دهید.

Dataset:این قسمت شامل داده‌های خام و دسته‌بندی متناظر با آن است. کیفیت و کمیت مجموعه‌داده به‌صورت مؤثری روی عملکرد مدل یادگیری ماشین تأثیر خواهد گذاشت.

Pre-Processing Module: این بخش از ساختار یادگیری ماشین داده‌های خام را دریافت می‌کند و متناسب با وظیفه (Task) ساختار و نوع داده ورودی، با انجام انواع گوناگون تبدیل‌ها (Transforms) مانند حذف داده‌های پرت، استخراج ویژگی و فیلترگذاری مناسب داده‌ها را برای پردازش و آموزش الگوریتم یادگیری ماشین بهبود می‌بخشد.  بخش پیش‌پردازش یک ساختار یادگیری ماشین می‌تواند روی کیفیت و کمیت یک مجموعه‌داده تأثیرگذار باشد، به‌عنوان نمونه با انجام تبدیل‌های ساخت داده مصنوعی (Augmentation) می‌توان تا حدی کمیت مجموعه‌داده را جبران کرد و با استفاده از این تبدیل‌ها به‌صورت وزن‌دار می‌توان برخی از مشکلات کیفی مجموعه‌داده مانند عدم تعادل (imbalance) را برطرف کرد.

Model: این بخش از ساختار یادگیری ماشین شامل الگوریتم یادگیری ماشین است که باتوجه‌به وظیفه (Regression, Classification, Segmentation) و ویژگی‌های متفاوت مجموعه‌داده مانند کیفیت و بار محاسباتی مطلوب، می‌تواند از میان انواع گوناگون انتخاب شود.

Post-Processing Module: این بخش از ساختار یادگیری ماشین شامل روش‌هایی است که باتوجه‌به خروجی الگوریتم یادگیری ماشین می‌تواند انتخاب شود تا پیش‌بینی‌های اشتباهی که به‌صورت آماری از یک الگوی خاص پیروی می‌کنند اصلاح شوند.

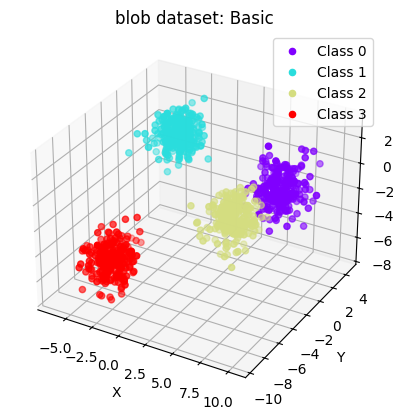
 Loss Function: این بخش از ساختار یادگیری ماشین شامل انواع گوناگونی از توابع هزینه مانند MSE، Cross-Entropy و یا انواع دیگر توابع هزینه می‌شود. تابع هزینه خصوصاً برای الگوریتم‌هایی که شامل بهینه‌سازی از طریق مشتق هستند دارای اهمیت ویژه‌ای است؛ زیرا توابع هزینه‌ای که مشتق‌پذیر نیستند یا مشتق آنها در نواحی خاصی خیلی زیاد یا خیلی کم است برای فرایند آموزش مشکل‌ساز خواهند بود. تابع هزینه می‌تواند بعضی از مشکلات کیفی مجموعه‌داده مانند عدم تعادل را برطرف کند.

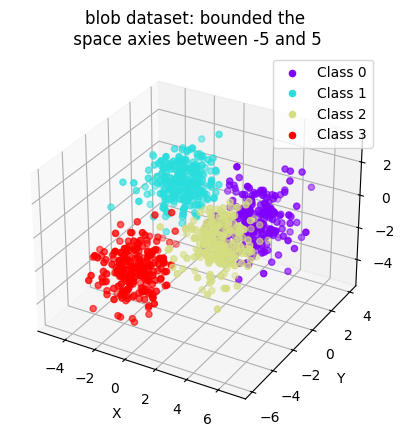
Optimizer: این قسمت از ساختار یادگیری ماشین شامل راه‌حل ریاضی برای بهینه‌سازی وزن‌های داخلی مدل یادگیری ماشین است. با درنظرگرفتن ویژگی‌های مجموعه‌داده و انتخاب صحیح الگوریتم بهینه‌سازی می‌توانیم زمان آموزش و عملکرد ساختار یادگیری ماشین را بهبود ببخشیم. یک الگوریتم بهینه‌سازی مناسب باعث می‌شود تا مدل یادگیری ماشین از نقاط بهینه محلی عبور کند و به نقاطی برود که تابع هزینه برای آنها مقدار کمتری را نشان می‌دهد.

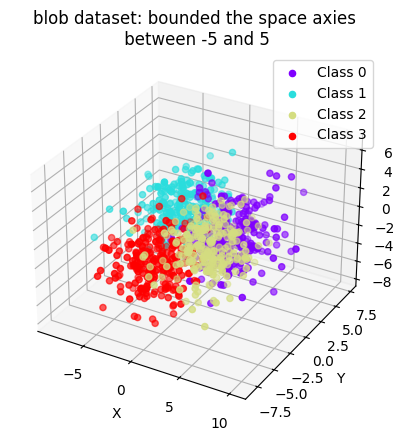
 Decision Policy Module: این قسمت از ساختار یادگیری ماشین باتوجه‌به وظیفه ساختار، یکی از انواع تبدیل‌های موجود را روی خروجی پیاده می‌کند. این تبدیل برای وظیفه طبقه‌بندی (Classification) می‌تواند تبدیل خروجی احتمالاتی به خروجی دو وضعیتی باشد که از آن با عنوان One-Hot Transform یاد می‌شود. همچنین در این بخش معیارهای ارزیابی ساختار مانند Recall، Precision و یا Accuracy محاسبه می‌شود.

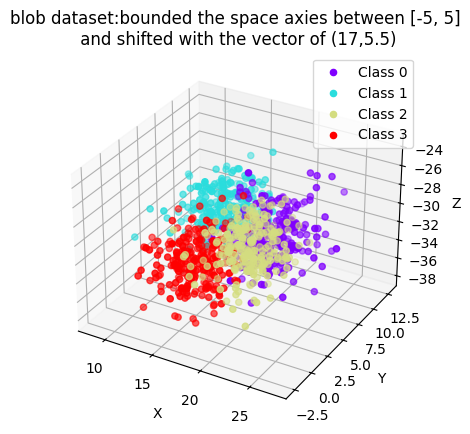
 به‌منظور تغییر طبقه‌بندی از ۲ کلاس به چند کلاس، می‌تواند در تمام قسمت‌های ساختار تأثیر بگذارد بااین‌حال مهم‌ترین قسمت شامل Loss Function و Optimizer است که باید برای استفاده به‌عنوان یک طبقه‌بند چند کلاسه با شرایط خروجی مدل سازگاری داشته باشند. به‌عنوان نمونه BCELoss در کتابخانه PyTorch به‌منظور استفاده در یک مسئله طبقه‌بندی چند کلاسه مطلوب نیست و همچنین در مدل LogisticRegression از کتابخانه sklearn اگر برای وظیفه طبقه‌بندی چند کلاسه استفاده شود، نمی‌تواند از liblinear به‌عنوان solver استفاده کند. در انتها به‌منظور تبدیل مدل از طبقه‌بند دو کلاسه به چند کلاسه می‌توان دو رویکرد multinomial یا one-vs-rest استفاده کرد که هنگام استفاده از رویکرد one-vs-rest، وظیفه یک طبقه‌بندی چند کلاسه به چندوظیفه طبقه‌بندی دو کلاسه تبدیل می‌شود و محدودیت‌های فوق‌الذکر برای این رویکرد وجود ندارد.

## با استفاده از datasets.sklearn، یک دیتاست با ۱۰۰۰ نمونه، ۴ کلاس و ۳ ویژگی تولید کنید و آن را به صورتی مناسب نمایش دهید. آیا دیتاستی که تولید کردید چالش برانگیز است؟ چرا؟ به چه طریقی می توانید دیتاست تولیدشده خود را چالش برانگیزتر و سخت تر کنید؟

به‌منظور ساخت یک مجموعه‌داده طبقه‌بندی از کتابخانه sklearn و متد make\_blobs استفاده شده است. نتیجه ساخت یک مجموعه‌داده متعادل با 4 کلاس و 3 ویژگی با تنظیمات پیش‌فرض متد make\_blobs در تصویر زیر نمایش داده شده است.

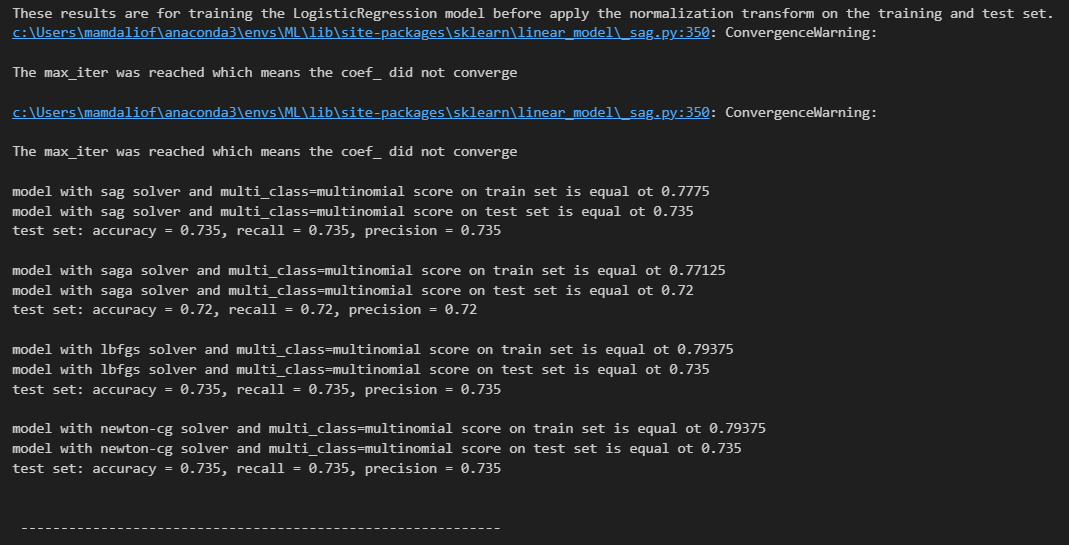
همان‌طور که از این تصویر مشخص است تفکیک‌پذیری داده‌ها در این مجموعه‌داده پالش برانگیز نمی‌باشد و تنها چالش این مجموعه‌داده جداسازی نقاط بنفش (class0) از نقاط زرد (class2) است. در ادامه به‌منظور چالش‌برانگیز کردن این مجموعه‌داده از میان پارامترهای قابل‌تنظیم در متد make\_blobs، center\_box= [-5, 5] قرار داده می‌شود. نتیجه این تنظیمات مجموعه‌داده زیر است.

در این مجموعه‌داده با کم‌شدن محدوده مجاز حضور نقاط در فضا، باعث نزدیک‌شدن مرکز دسته‌های هر کلاسه به یکدیگر شده است و در نتیجه آن چالش جداسازی نقاط از هم و تشخیص درجه تعلق آنها به هر دسته چالش‌برانگیزتر شده است؛ اما همچنان جداسازی این داده‌ها از یکدیگر می‌تواند چالش‌برانگیزتر باشد. با تغییر انحراف از معیار مربوط به دستور ساخت مجموعه‌داده می‌توان چالش این مجموعه‌داده را افزایش داد. در ادامه انحراف از معیار مجموعه‌داده برابر 2 قرار داده شده است و نتیجه آن در شکل زیر قابل مشاهده است.

در مجموعه‌داده فوق تفکیک داده‌ها از یکدیگر بسیار مشکل شده است و می‌تواند وظیفه‌ای چالش‌برانگیز برای هر ساختار یادگیری ماشین باشد. به‌منظور اضافه‌کردن نامعینی می‌توان نویز را به نقاط مجموعه‌داده اضافه کرد که در ادامه این آزمایش از انجام آن صرف‌نظر شده است. در ادامه مجموعه‌داده فوق با مقادیر ثابت [17, 5.5, -30.33] جمع شده است تا تاثیر normalization مجموعه داده قابل تشخیص باشد.

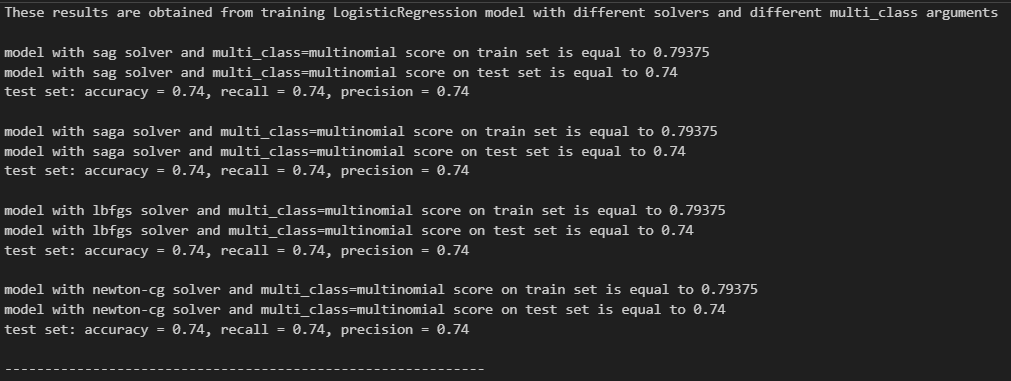
## با استفاده از حداقل دوطبقه بند خطی آمادۀ پایتون (در model\_linear.sklearn (و در نظر گرفتن فراپارامترهای مناسب، چهار کلاس موجود در دیتاست قسمت قبلی را از هم تفکیک کنید. ضمن توضیح روند انتخاب فراپارامترها (مانند تعداد دورۀ آموزش و نرخ یادگیری)، نتیجۀ دقت آموزش و ارزیابی را نمایش دهید. برای بهبود نتیجه از چه تکنیک هایی استفاده کردید؟

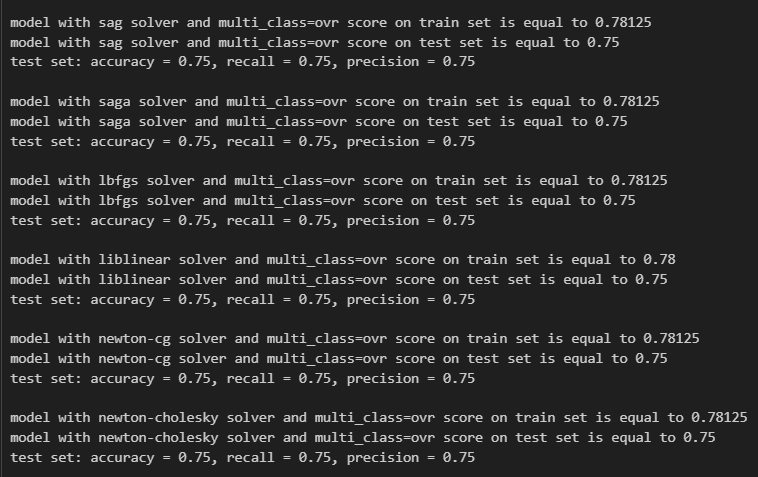
در ادامه از دو طبقه‌بند خطی LogisticRegression و SGDClassifier استفاده شده است و با استفاده از کتابخانه sklearn پیاده‌سازی آن انجام شده است. این نکته حائز اهمیت است که 20 درصد از مجموعه داده مذکور برای انجام عمل ارزیابی و 80 درصد برای عمل آموزش جدا شده است.

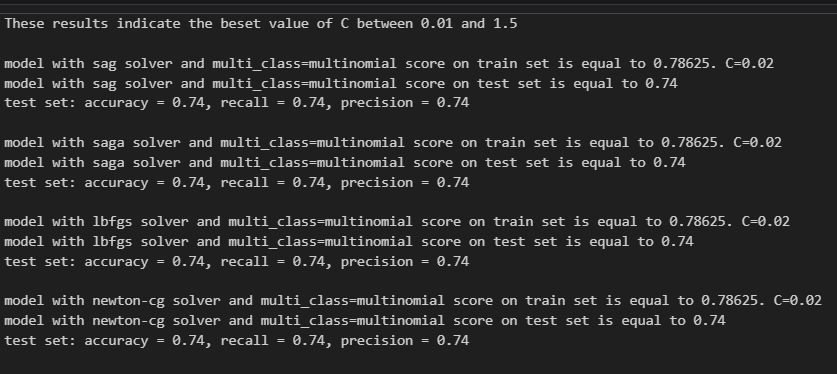
در ابتدا این مدل روی مجموعه‌داده خام آموزش و به‌ازای دو رویکرد متفاوت multinomial و ovr و تمام solverهای موجود آموزش‌دیده است که نتایج آن به شرح زیر است.

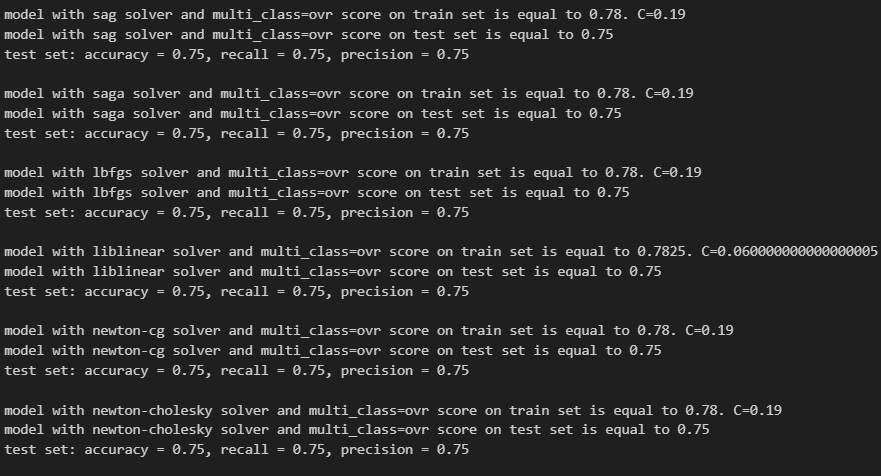
همان‌طور که مشخص است تعدادی از الگوریتم‌ها همگرا نشده‌اند و نتیجه این آموزش از لحاظ بار محاسباتی و دقت مناسب نمی‌باشد.

در گام بعد تبدیل Normalization روی مجموعه‌داده انجام می‌شود و نتایج آموزش به شرح زیر است.

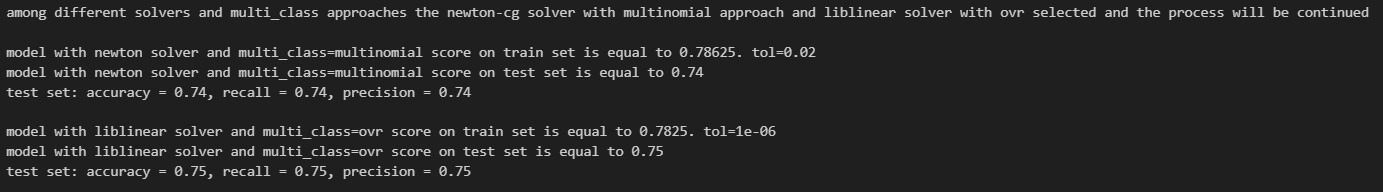




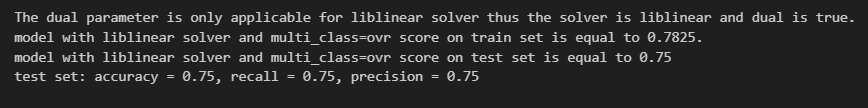
در ادامه بین بازه 0.01 تا 1.5 با گام 0.01 مقادیر مختلفی برای پارامتر C که میزان Regularization را در فرایند آموزش تعیین می‌کند بررسی شده است تا بهترین مقدار محاسبه شود. نتایج این مجموعه آموزش به شرح زیر است:  




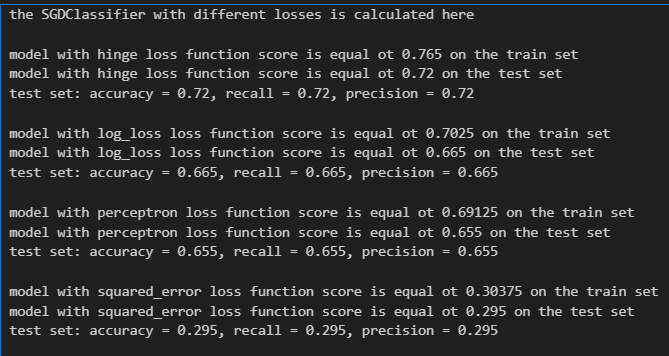
در ادامه دو مدل از میان تمام مدل‌های موجود انتخاب شده و باقی آموزش‌ها بر روی این مدل‌ها صورت خواهد گرفت. فراپارامتر بعدی که مقدار بهینه آن باید محاسبه شود tol است. مقدار این فراپارامتر در بازه تا عوض شده است تا حداقل مقدار خطای قابل قبول برای مجموعه‌داده آموزش محاسبه شود. این آموزش‌ها در شرایطی انجام شده است که مقدار C از قسمت قبل بدست آمده است.

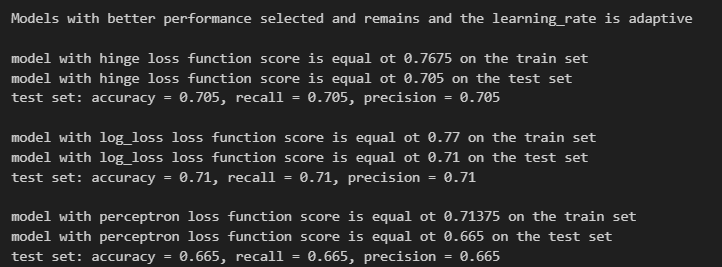


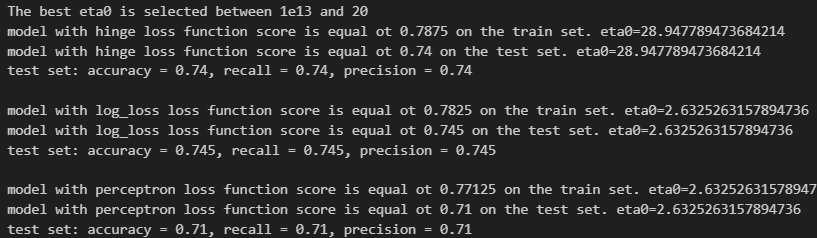
در انتها فراپارامتر dual برای solver liblinear فعال شده است تا تغییراتی که این فراپارامتر در آموزش ایجاد می‌کند مشاهده شود. نتایج حاصل شده در ادامه قابل‌مشاهده است.

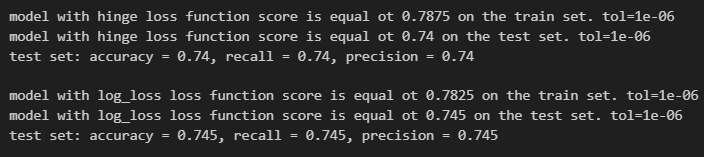


حال مدل SGDClassifier به درنظرگرفتن lossهای موجود آموزش‌دیده شده است که نتایج آن در ادامه قابل‌مشاهده است. در ابتدا مدل روی داده‌های نرمال آموزش‌دیده است.

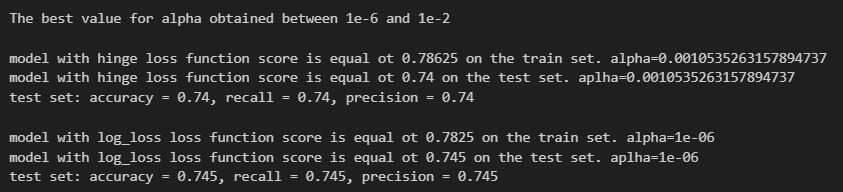
همان‌طور که مشخص است بعضی از مدل‌ها عملکرد مناسبی از خود نشان نداده‌اند که در نتیجه از ادامه فرایند آموزش حذف خواهند شد. در ادامه فراپارامتر  learning\_rate='adaptive' قرار داده شده است و نتایج آموزش آن قابل نسبت به نتایج گذشته بهتر است.

در ادامه مقدار اولیه eta0 در بازه 0.0001 تا 50 انتخاب خواهد شد و نتایج بهترین آموزش در ادامه قابل رویت است.

نتایج به‌دست‌آمده نشان‌دهنده بهبود فرایند آموزش است. در ادامه دو مدل که بهترین عملکرد را دارند انتخاب شده و باقی آموزش‌ها بروی آنها انجام خواهد گرفت. آموزش بعدی به‌منظور به‌دست‌آوردن بهترین مقدار tol است که حداقل میزان قابل‌قبول برای تابع هزینه را روی مجموعه‌داده آموزش مشخص می‌کند.



برای به‌دست‌آوردن بهترین مقدار alpha که میزان regularization را مشخص می‌کند مقادیر آلفا را از تا جابجا می‌کنیم که نتیجه بهترین مقدار برای alpha در شکل زیر مشخص است.

**

*در انتها بهترین نسبت مجموعه‌داده valdidation به‌ازای مقادیر 0.1و 0.15و 0.2 محاسبه شده است.*

**

*نکته مهم درمورد نتایج خروجی این است که در تصاویر فوق، مقادیر precision و recall به‌صورت micro محاسبه شده است که با مقدار score برابر است؛ اما در فایل notebook این مقادیر به‌صورت macro محاسبه شده است. در ادامه بهترین مدل بر اساس اطلاعات و matricهای موجود روی مجموعه‌داده ارزیابی، مدلی است که به فراپارامتر‌های زیر آموزش ببیند.*

LogisticRegression(penalty="l2",

                            dual=True,

                            C=0.06,

                            solver="liblinear",

                            max\_iter=1500,

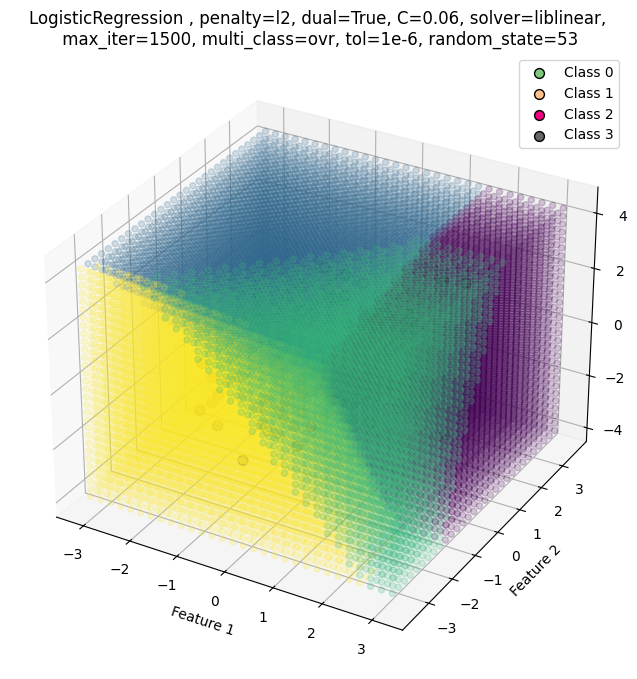
                            multi\_class="ovr",

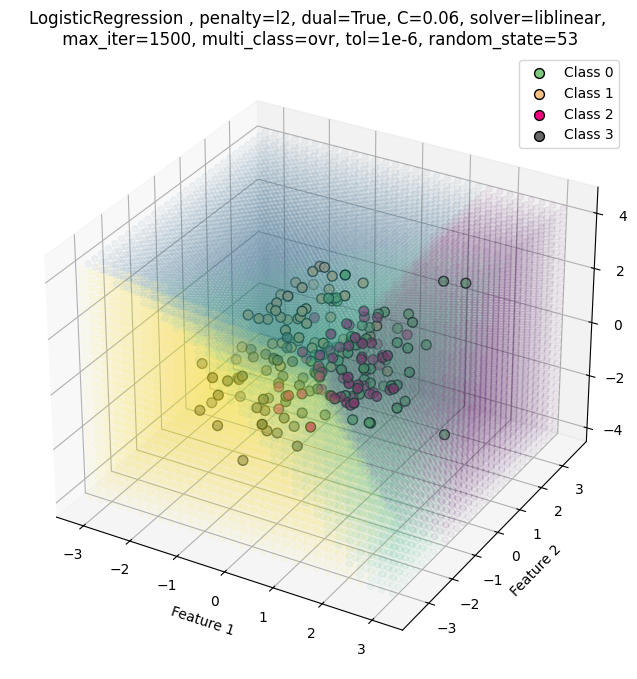
                            tol=1e-6,

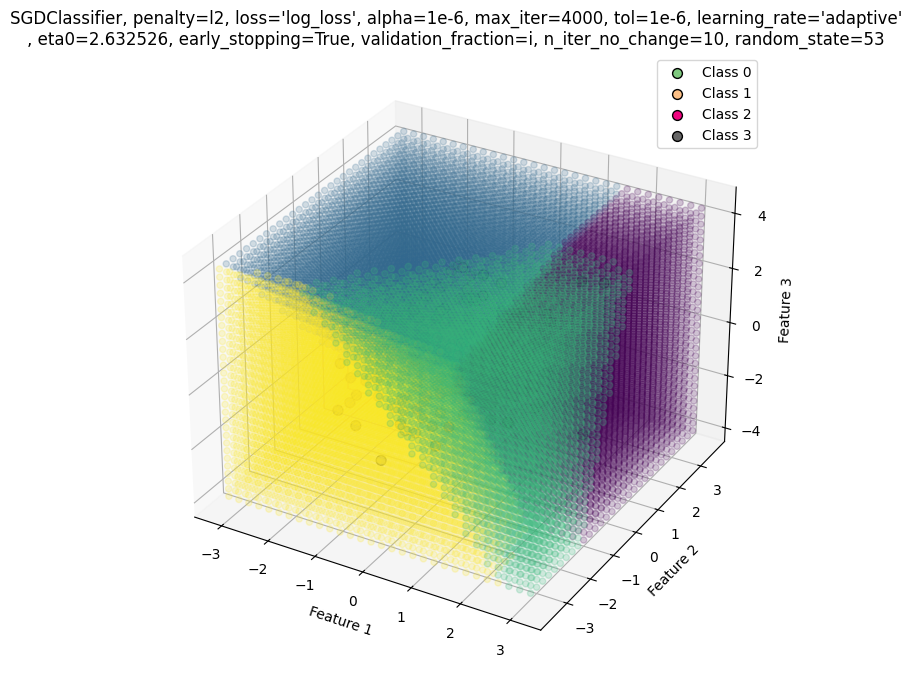
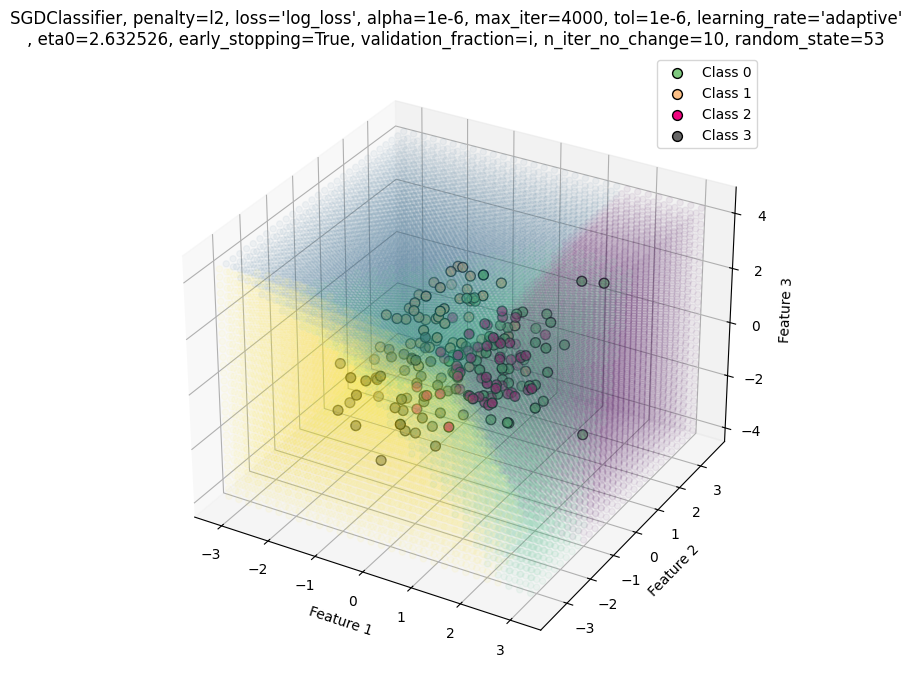
                            random\_state=53)

## مرز و نواحی تصمیم گیری برآمده از مدل آموزش دیدۀ خود را به همراه نمونه ها در یک نمودار نشان دهید. اگر می توانید نمونه هایی که اشتباه طبقه بندی شده اند را با شکل و رنگ متفاوت نمایش دهید.

بدین منظور از ChatGPT کمک گرفته شده است. مرز تصمیم و نتایج پیش‌بینی‌های بهترین مدل روی مجموعه داده ارزیابی به شرح زیر است.

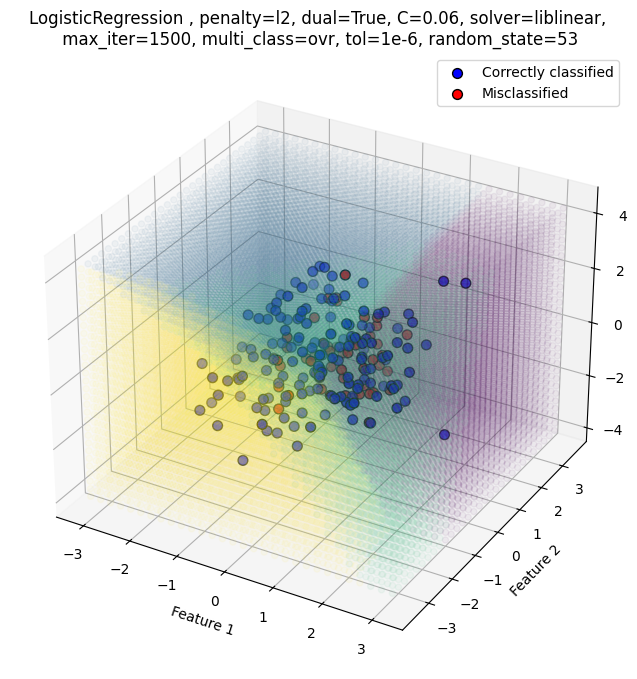


نمایش این نمودار به‌صورت interactive در داخل فایل Q1.1.ipynb قابل رویت است.



نمایش این نمودار به صورت interactive در داخل فایل Q1.1.ipynb قابل رویت است.

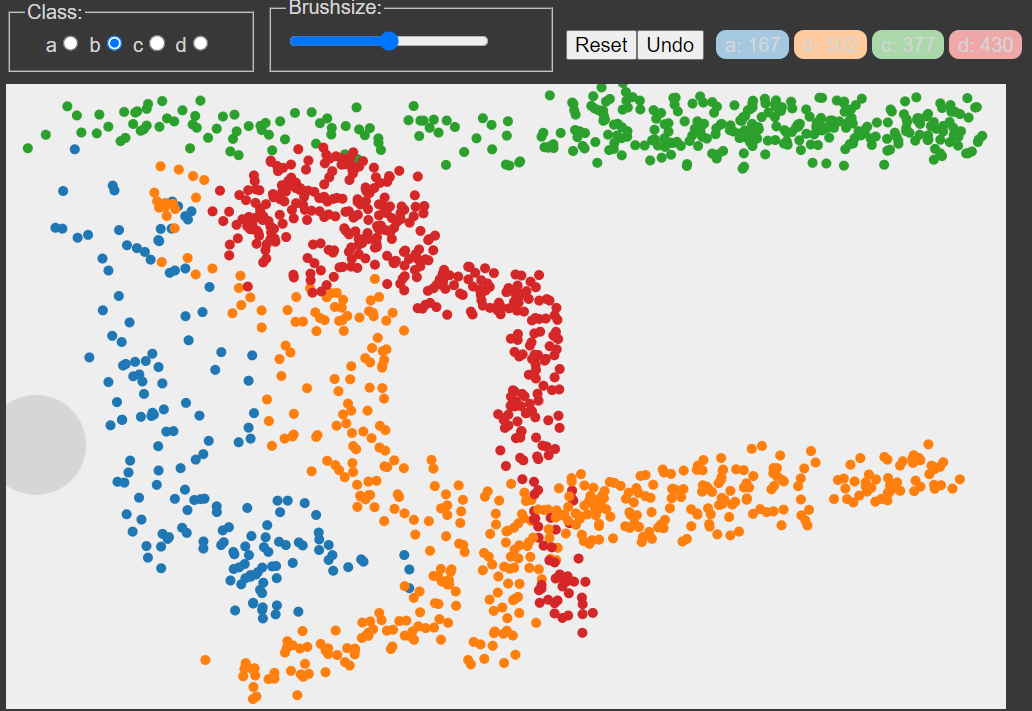
در ادامه نقاطی که درست پیش‌بینی شده اند را نسبت به نقاطی که اشتباه پیش‌بینی شده‌اند نمایش داده شده است.





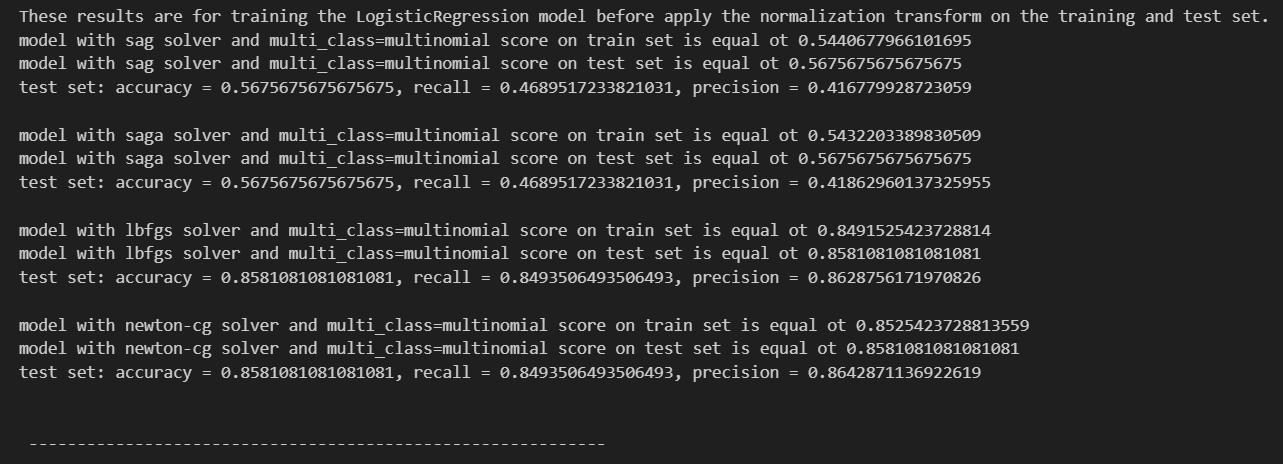
## فرآیندی مشابه قسمت «۲» را با تعداد کلاس و ویژگی دلخواه؛ اما با استفاده از ابزار drawdata تکرار کنید. قسمت های «۳» و «۴» را برای این داده های جدید تکرار و نتایج را به صورتی مناسب نشان دهید.

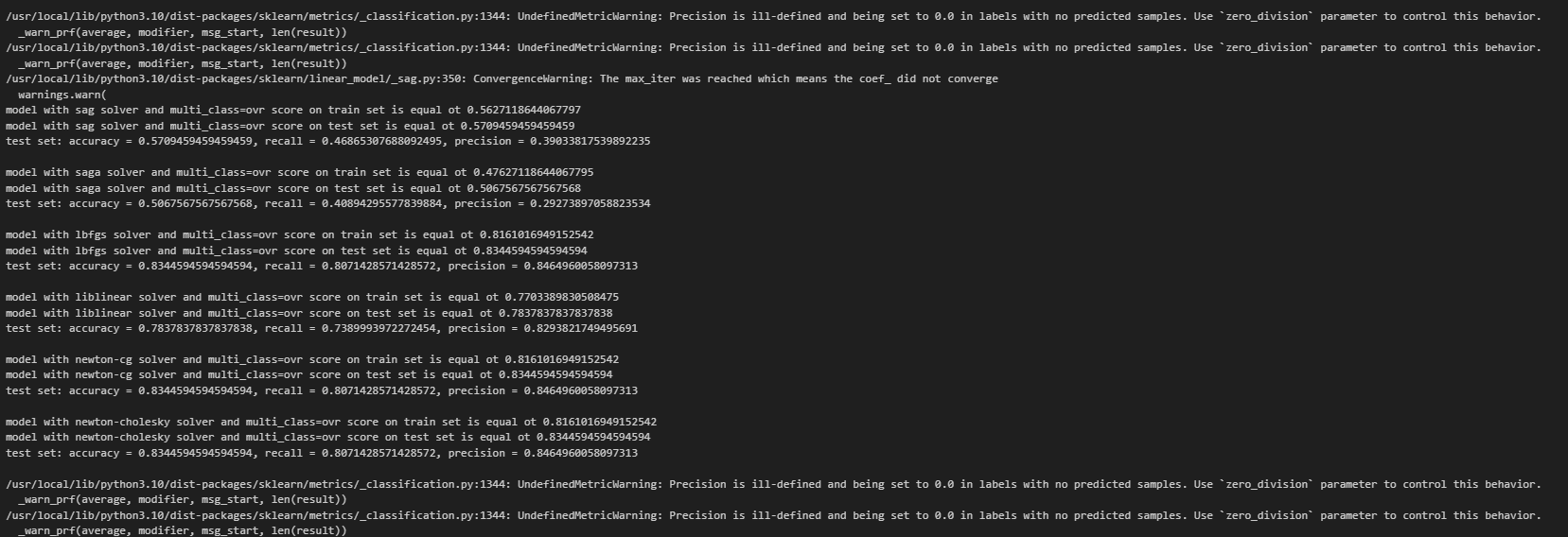
در ابتدا مجموعه داده را با استفاده از کتابخانه drawdata ایجاد می‌کنیم.



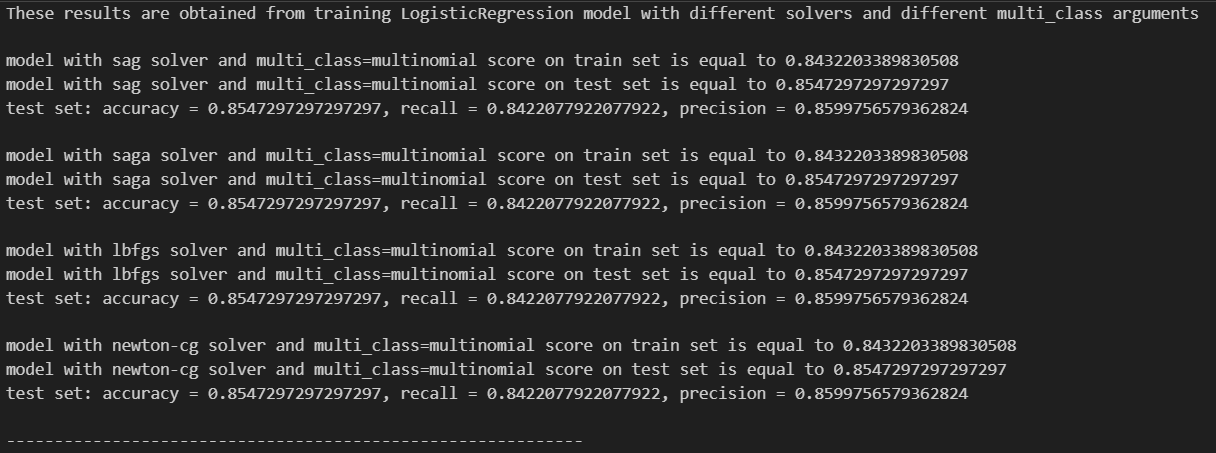
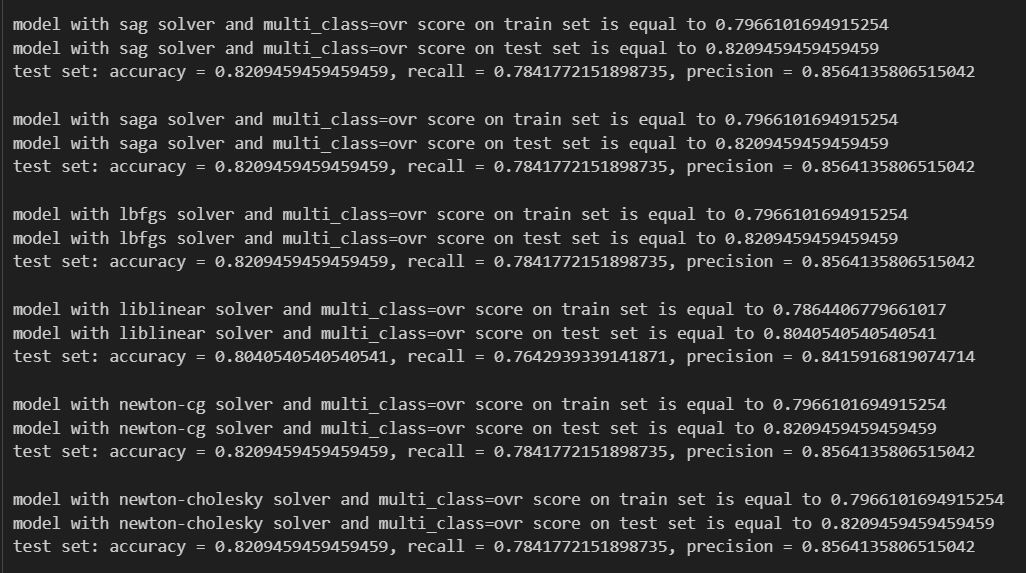
مجموعه داده فوق به منظور آموزش و ارزیابی ساختار یادگیری ماشین استفاده شده است. 20 درصد از این مجموعه داده برای ارزیابی و 80 درصد آن برای آموزش استفاده شده است.

تمام مراحلی که در قسمت قبل توضیح داده شد را روی این مجموعه داده پیاده می‌کنیم که تصاویر نتایج بخش‌های گوناگون به ترتیب زیر می‌باشد.

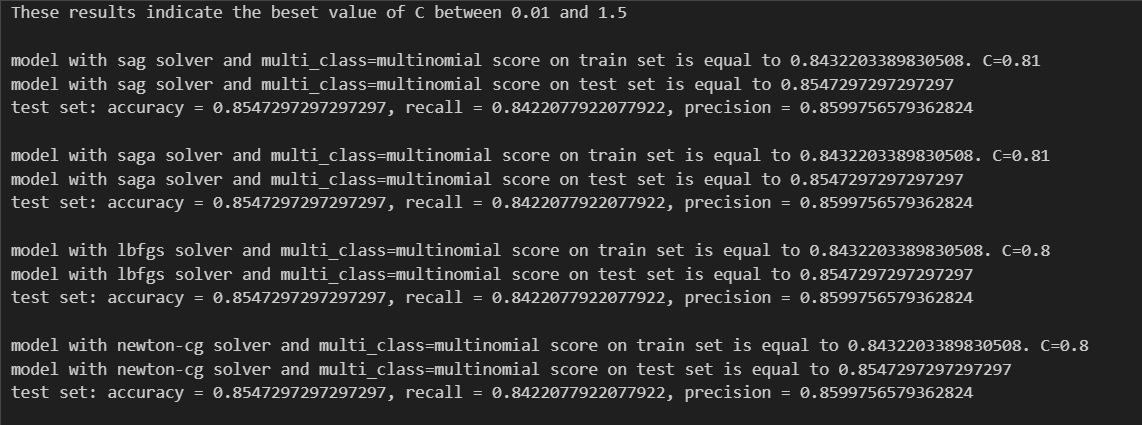


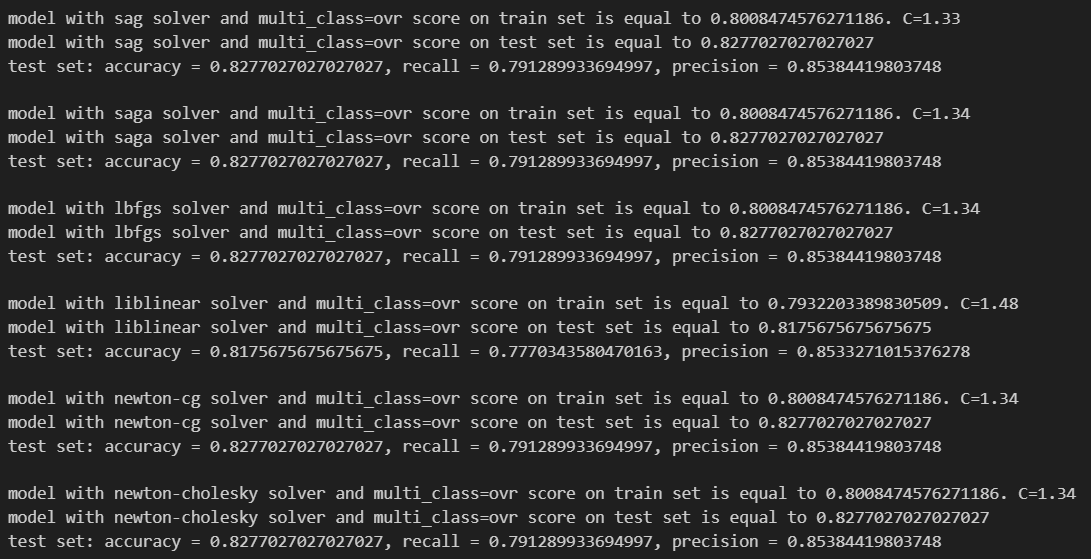


همانطور که مشخص است پیش از اعمال normalization مدل‌ها همگرا نشده‌اند که این به معنی بهینه نبودن فرایند آموزش و عملکرد مدل‌ها است. در ادامه مدل روی داده‌های نرمال شده آموزش‌داده‌شده است.



همانطور که مشخص است نتایج روی داده‌های نرمال بهتر است. حال به محاسبه فراپارامتر C می‌پردازیم.

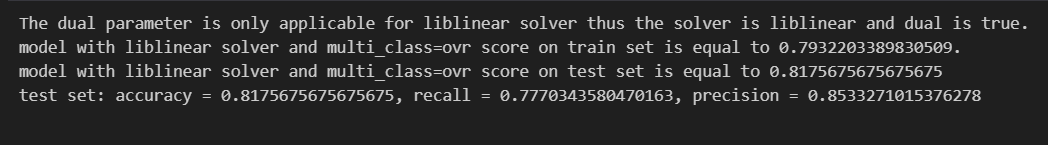


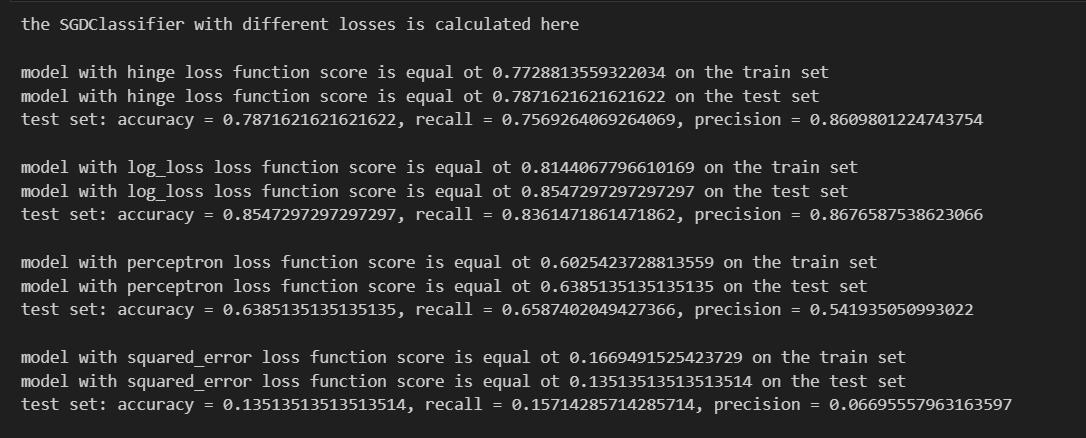


در گام بعدی دو مدل که عملکرد بهتری دارند را نگه می‌داریم و بهترین میزان tol را برایشان محاسبه می‌کنیم.

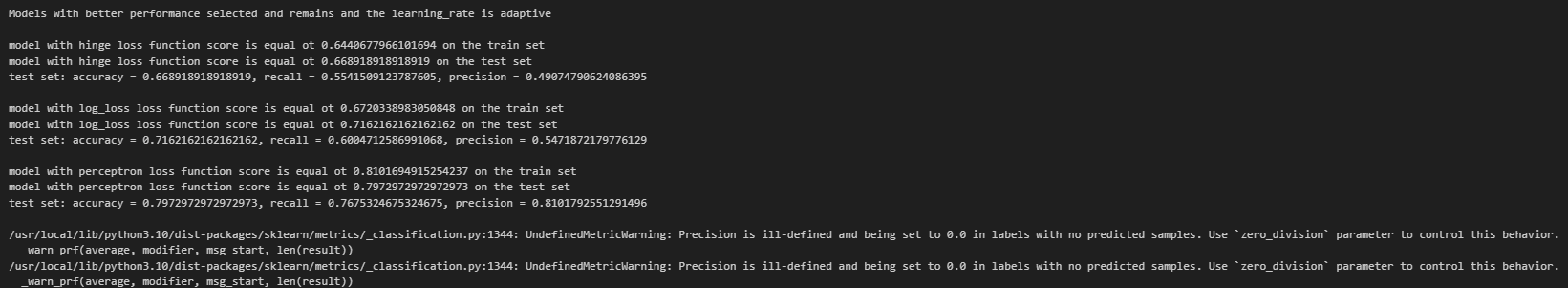


در انتها فراپارامتر dual را برای liblinear فعال می‌کنیم و مدل را آموزش می‌دهیم.

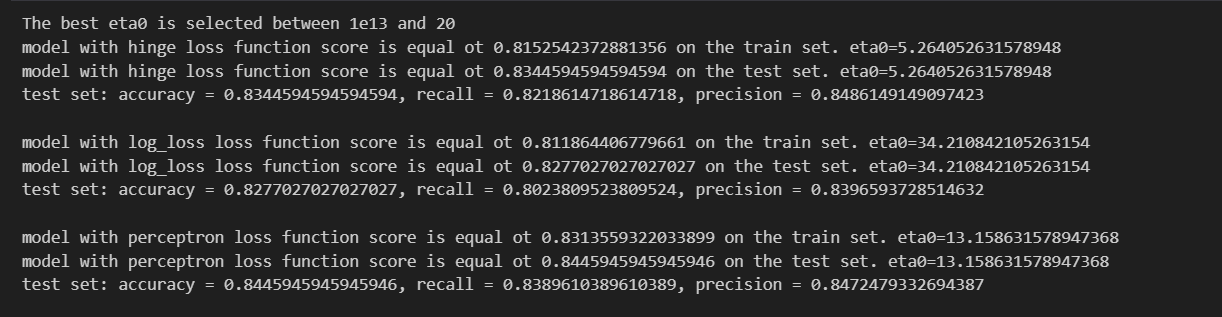


بهترین مدل LogisticRegresion در آخرین آموزش حاصل شده است. حال به سراغ مدل SGDClassifier می‌رویم.

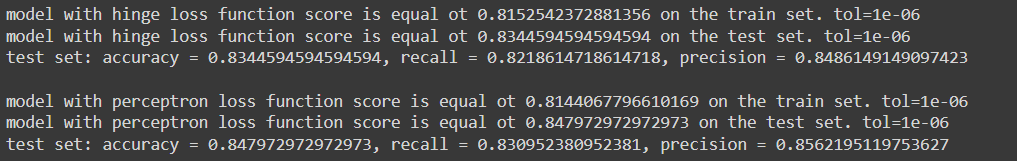
تصویر فوق نشان می‌دهد که این مدل به روی دیتای نرمال شده چگونه عمل می‌کند.



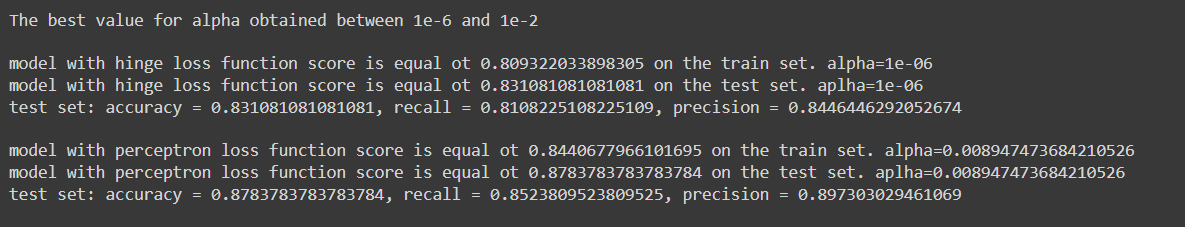
همان‌طور که مشخص است عملکرد بعضی مدل‌ها کاهش پیدا کرده؛ اما این به دلیل مقدار eta0 اولیه است و با اصلاح آن نتیجه بهتر خواهد شد؛ بنابراین به سراغ بهینه‌سازی مقدار eta0 می‌رویم.



با بهینه‌سازی مقدار eta0 و بررسی عملکرد مدل‌ها به این نتیجه می‌رسیم که adaptive learning rate می‌تواند به بهبود نتایج کمک کند. حال دو مدلی که عملکرد بهتری دارند را انتخاب کرده و باقی آموزش‌ها را به روی آنها انجام می‌دهیم.

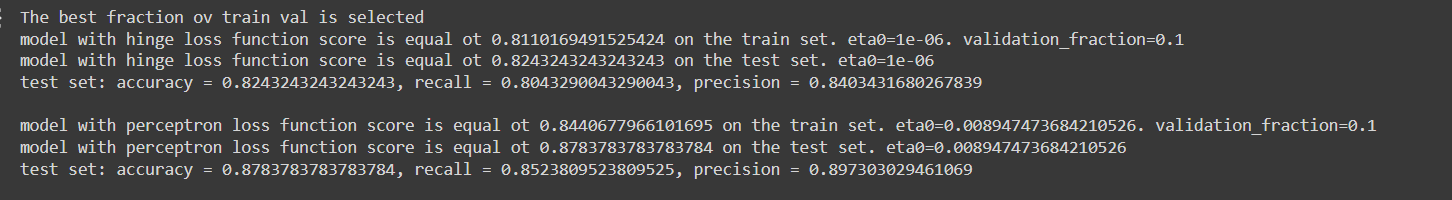


در ادامه بهترین مقدار فراپارامتر alpha را حساب می‌کنیم.



همان‌طور که مشخص است عملکرد مدل perceptreon اختلاف زیادی با مدل‌ها دیگر دارد.

در انتها بهترین نسبت مجموعه‌داده صحت‌سنجی نسبت به آموزش را به دست می‌آوریم.



همان‌طور که از مقایسه قابل‌تشخیص است بهترین مدل SGDClassifier نسبت به بهترین مدل LogisticRegression عملکرد مطلوب‌تری دارد؛ بنابراین اگر یک مدل با مشخصات زیر آموزش ببیند می‌تواند بهترین نتیجه را در پیش‌بینی کلاس‌های این دیتاست داشته باشد.

SGDClassifier(penalty="l2",

                      loss='perceptron',

                      alpha=0.008947473684210526,

                      max\_iter=4000,

                      tol=1e-6,

                      learning\_rate='adaptive',

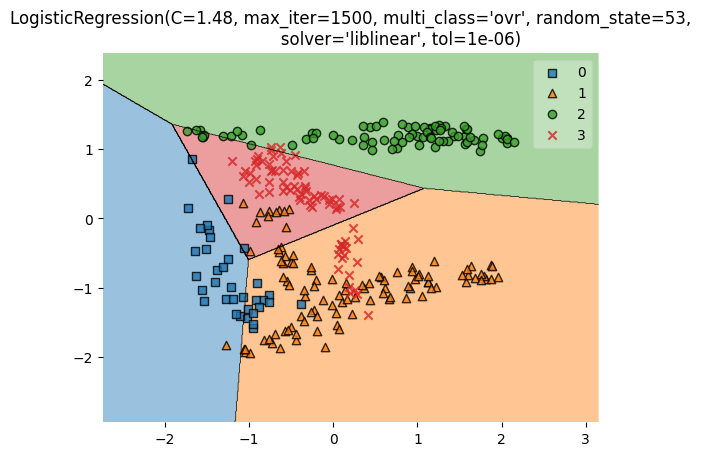
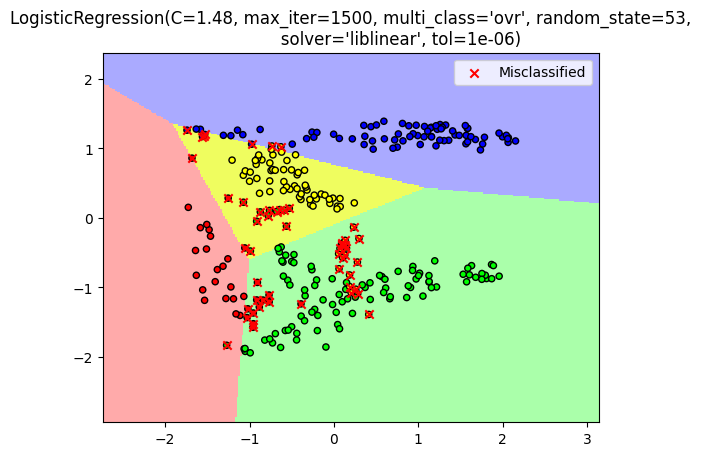
                      eta0=13.15863157,

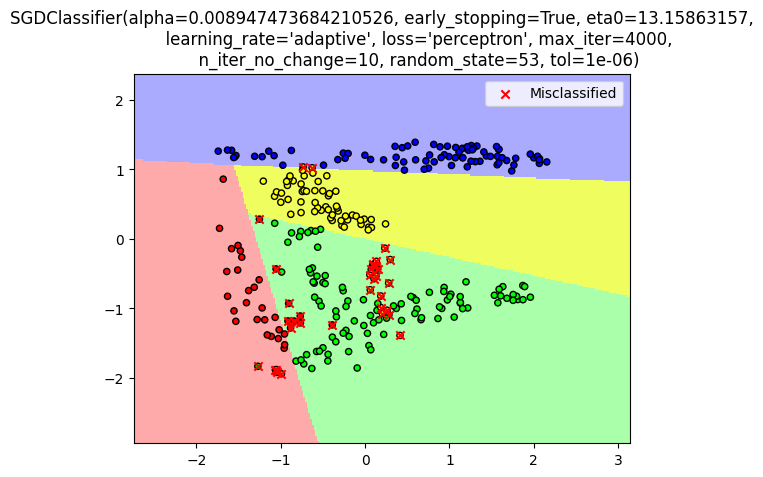
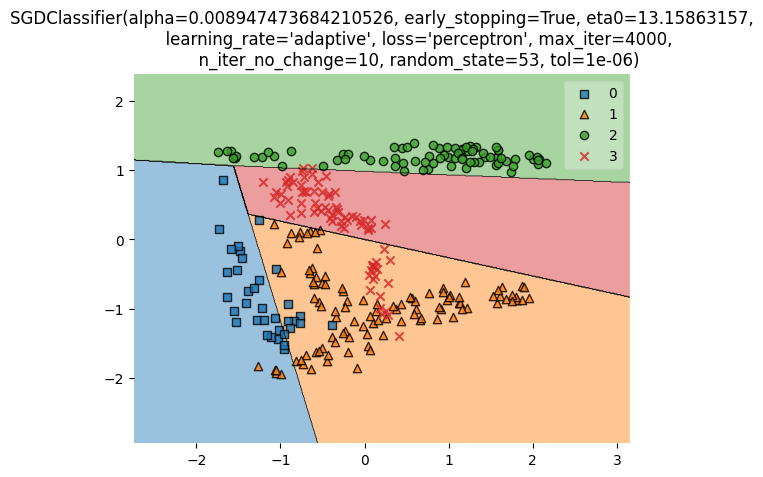
                      early\_stopping=True,

                      validation\_fraction=0.1,

                      n\_iter\_no\_change=10,

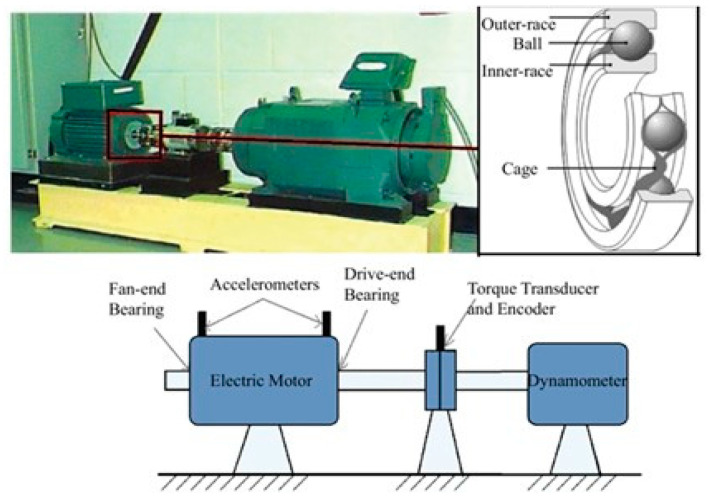
                      random\_state=53)

حال به‌رسم نمودارهای خواسته شده در صورت سؤال، برای مدل LogisticRegression می‌پردازیم.

حال نتایج را برای مدل SGDClassifier بررسی می‌کنیم.

سؤال 2

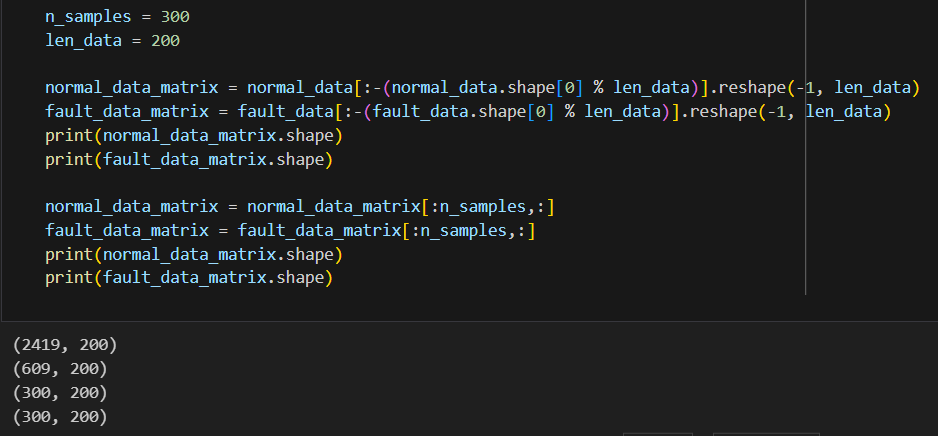
## با مراجعه به صفحۀ دیتاست Bearing CWRU با یک دیتاست مربوط به حوزۀ «تشخیص عیب» آشنا شوید. با جستجوی آن در اینترنت و مقالات، توضیحاتی از اهداف، ویژگی ها و حالت های مختلف این دیتاست ارائه کنید. در ادامه، ابتدا به صفحۀ داده های سالم مراجعه کنید و داده های کلاس سالم (X\_Normal۱ (را دریافت کنید. سپس، به صفحۀ داده های عیب در حالت k12 مراجعه کرده و داده های کلاس عیب (X\_007IR (را دریافت کنید.

CWRU Bearing به عنوان یک مجموعه داده جامع و معیار (Benchmark) برای ارزیابی الگوریتم‌های یادگیری ماشین با وظیفه (Task) طبقه‌بندی (Classification) در زمینه تشخیص عیب شناخته می‌شود. این مجموعه داده به منظور تشخیص خرابی در Ball Bearing توسعه یافته است. این مجموعه داده به منظور توسعه و آزمایش الگوریتم‌های تشخیص خرابی مورد استفاده قرار می‌گیرد.[[1]](#endnote-1) [[2]](#endnote-2)  
با توجه به شکل زیر ابزار موجود در جمع‌آوری دیتای CWRU، شامل یک موتور دو اسب بخار،token converter, encoder, dynamometer, و دستگاه کنترل است که به منظور جمع‌آوری داده به یکدیگر متصل شده اند. یک بلبرینگ معمولی شامل یک مسیر داخلی، یک مسیر خارجی، توپ ها و یک قفس است که توپ ها را در جای خود نگه می دارد. بلبرینگ شفت موتور گشتاور را از طریق dynamometer و سیستم کنترل الکترونیکی به شافت منتقل می شود[[3]](#endnote-3). 

این مجموعه شامل سیگنال‌های لرزش ball bearingها است. سنسورهای لرزش در فرایند جمع‌آوری دیتا در انتهای drive، fan و Basement قرار داده شده است. در ادامه به منظور ایجاد شرایط نزدیک به حالت واقع، عیوب به صورت مصنوعی در دستگاه ایجاد می‌شوند تا توسط حسگرها مقدار لرزش دستگاه معیوب حین کار اندازه‌گیری شود. به صورت کلی با استفاده از داده‌های حسگرها می‌توان سه وضعیت عادی، عیب در drive و عیب در fan را مشخص کرد.iii   
حسگرها با نرخ 12000 نمونه در ثانیه از سه محل متفاوت میزان لرزش را ثبت کرده‌اند این درحالی است که برای حسگری که در محل drive قرار دارد با نرخ 48000 نمونه در ثانیه نمونه‌برداری انجام شده است.i

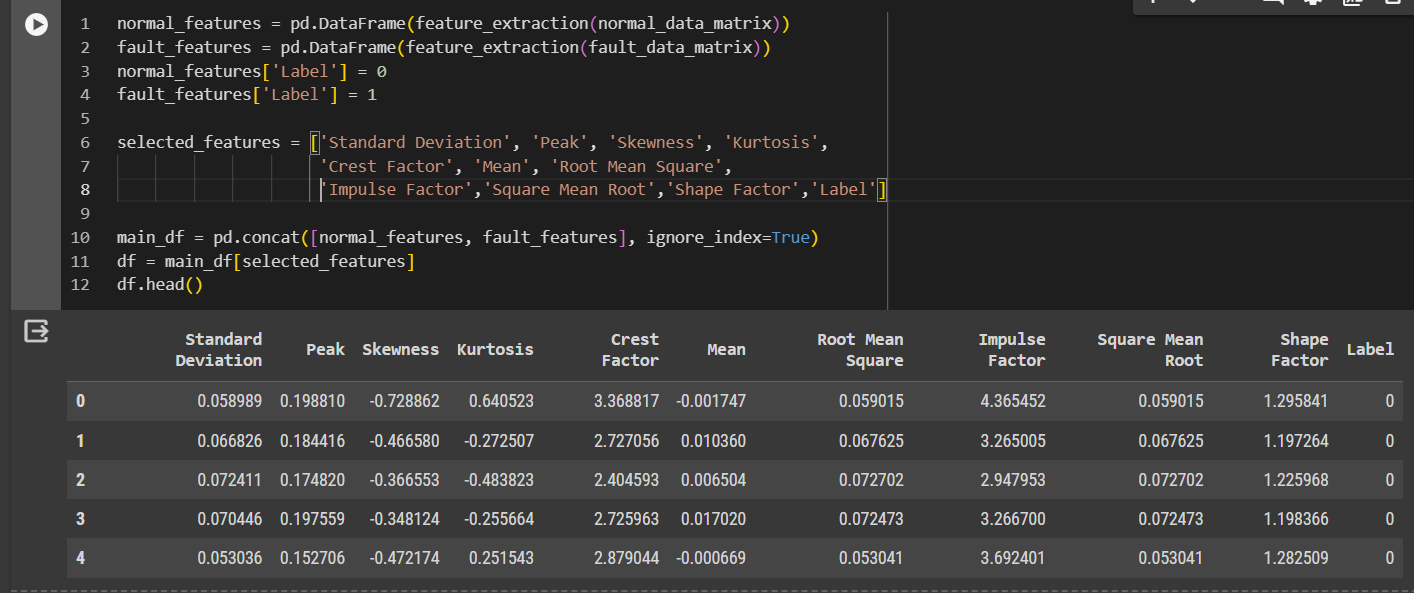
## برای تشکیل دیتاست مراحل زیر را انجام دهید

## ز هر کلاس Ⅿ نمونه با طول N جدا کنید (Ⅿ حداقل ۱۰۰ و N حداقل ۲۰۰ باشد). یک ماتریس از داده های هر دو کلاس به همراه برچسب مربوطه تشکیل دهید. می توانید پنجره ای به طول N در نظر بگیرید و در نهایت یک ماتریس N × Ⅿ از داده های هر کلاس استخراج کنید.

کد فوق به منظور حرکت یک پنجره با طول 200 روی مجموعه داده می‌باشد. در نتیجه آن ماتریس داده‌های سالم شامل 2419 نمونه و ماتریس داده‌های ناسالم شامل 609 نمونه است. در ادامه تعداد 300 نمونه از هرکدام جدا شده تا در فرایند آموزش و ارزیابی استفاده شود.

## در مورد اهمیت استخراج ویژگی در یادگیری ماشین توضیحاتی بنویسید. سپس، با استفاده از حداقل ۸ عدد از روش های ذکرشده در جدول ،۱ ویژگی های دیتاست قسمت «۲‐آ» را استخراج کنید و یک دیتاست جدید تشکیل دهید.

استخراج ویژگی در الگوریتم‌های یادگیری ماشین خصوصاً الگوریتم‌های قدیمی‌تر از مهم‌ترین قسمت‌ها است. روش‌های استخراج ویژگی مبتنی بر روش‌های گوناگون ریاضی هستند و متناسب با وظیفه ساختار یادگیری ماشین و نوع داده‌های ورودی باید از بین آنها بهترین‌ها را انتخاب کرد. به‌عنوان‌مثال تبدیل فوریه یکی از انواع ویژگی است که می‌توان آن را از داده‌هایی که خاصیت فرکانسی دارند استخراج کرد و تنها هارمونیک‌هایی که شامل اطلاعات موردنیاز مسئله هست را بررسی کرد. گروهی دیگر از روش‌های استخراج ویژگی مقادیر نسبی را به‌عنوان خروجی خود نشان می‌دهند که با استفاده از این گروه از روش‌ها می‌توانیم حساسیت ساختار یادگیری ماشین را نسبت به تغییرات عملکرد حسگرها و یا تغییر دقت حسگرها کاهش دهیم که این مسئله باعث همگانی شدن ساختار می‌شود. از سوی دیگر گاهی اوقات ویژگی‌های مربوط به یک‌طبقه خاص در مسئله وابستگی به دامنه سیگنال‌های ورودی دارد که این مسئله با گروه دیگر از روش‌های استخراج ویژگی مانند RMS یا روش قله به دست می‌آیند. کاربرد دیگر روش‌های استخراج ویژگی در شرایطی است که ابعاد داده‌های ورودی زیاد است و امکان بررسی آنها به‌صورت مستقل وجود ندارد؛ بنابراین با روش‌های کاهش ابعاد مانند PCA مسئله را ساده‌تر می‌کنیم.

در ادامه برچسب‌های هرکدام از کلاس‌ها را در ستونی جدید به اسم label ذخیره می‌کنیم و ویژگی‌های موجود در متغیر selected\_features را استخراج کرده یک مجموعه‌داده جدید را ایجاد می‌کنیم.

## اهمیت فرایند برزدن را مطرح کنید و مجموعه‌داده را به زیربخش آموزش و ارزیابی تقسیم کنید.

بر زدن در فرایند آموزش الگوریتم یادگیری ماشین باعث می‌شود که در گام نخست روش‌های مبتنی بر مشتق به‌صورت پایدارتر همگرا بشوند و احتمال وقوع یک batch با شرایطی خاص را کاهش می‌دهد. از سوی دیگر بر زدن باعث می‌شود که الگوریتم با احتمال بیشتری از نقاط بهینه محلی عبور کند و به سمت نقطه بهینه جهانی حرکت کند.   
با استفاده از دستور زیر عملیات بر زدن مجموعه‌داده رخ خواهد داد.

df\_shuffled = df.sample(frac=1).reset\_index(drop=True)

در ادامه با استفاده از دستور زیر مجموعه‌داده با نسبت 0.8 به 0.2 به زیرمجموعه‌های آموزش و ارزیابی تقسیم می‌شود این درحالی است که اب تنظیم پارامتر stratify از هر کلاس به تعداد مساوی انتخاب می‌کنیم تا با مشکل عدم تعادل روبرو نشویم.

X\_train\_raw, X\_test\_raw, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20, stratify=y, random\_state=53)

## حداقل دو روش برای نرمال سازی داده ها را با ذکر اهمیت این فرآیند توضیح دهید و با استفاده از یکی از این روش ها، داده ها را نرمال کنید. آیا از اطلاعات بخش «ارزیابی» در فرآیند نرمال سازی استفاده کردید؟ چرا؟

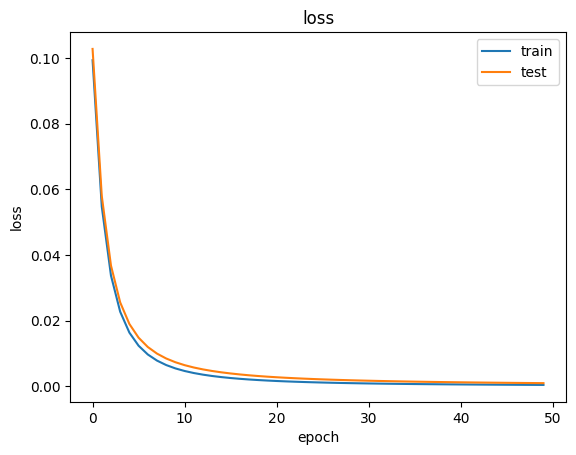
در ابتدا باید توضیح دهیم که فرایند نرمالیزه‌کردن در کدام ابعاد ماتریس داده انجام می‌شود. اگر تبدیل ریاضی در جهت بردار نمونه انجام شود به آن batch normalization می‌گویند و اگر در جهت بردار ویژگی استفاده شود به آن layer normalization می‌گویند.

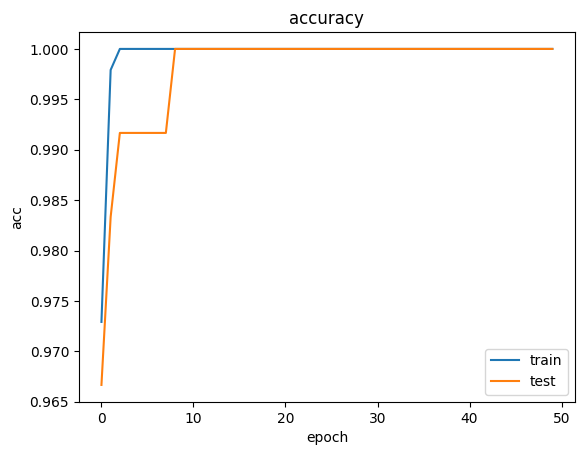
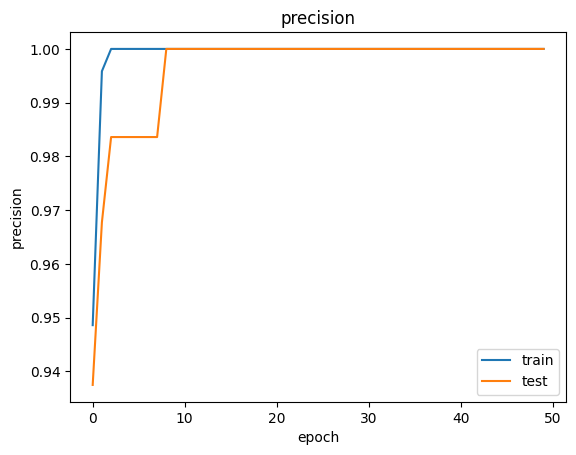
روش اول نرمال کردن داده با استفاده از نرم دوم بردار داده است. در این روش با استفاده از فرمول زیر تمام مقادیر مربوط به یک ویژگی خاص که در بردار آن ویژگی قرار دارد نرمال می‌شود. این روش از نرمال کردن باعث می‌شود که علاوه بر حفظ جهت بردار داده، نسبت اهمیت هر درایه نسبت به باقی درایه‌ها ثابت بماند. این روش معمولا به صورت batch normalization استفاده می‌شود.

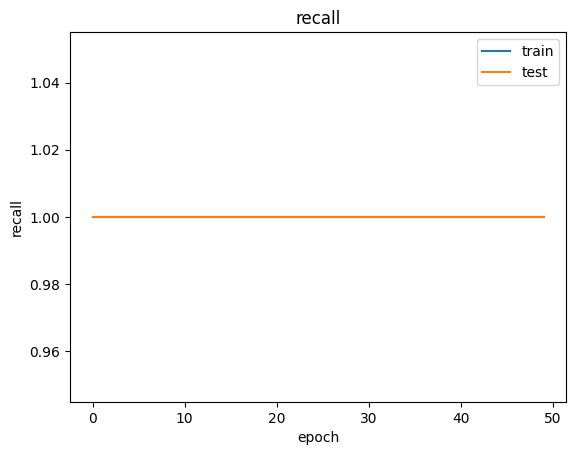
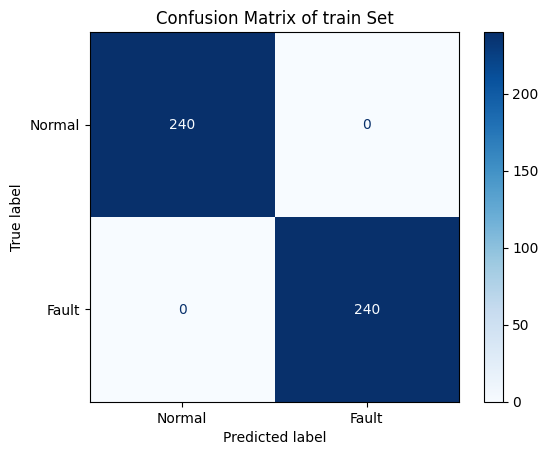
روش دوم نرمال کردن بر اساس حداکثر و حداقل مقدار است که این روش مقادیر را بین 0 و 1 تصویر می‌کند و در بسیاری از ساختارهای یادگیری ماشین خصوصا در پردازش تصاویر استفاده می‌شود. فرمول این روش به شکل زیر است.

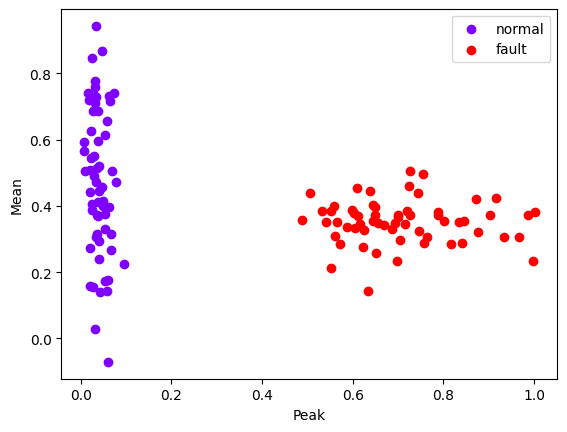
در این بخش از روش نرمال کردن حداکثر و حداقل استفاده شده است و محدوده داده‌ها به عددی بین 0 و 1 تصویر شده. نکته حائز اهمیت این است که از اطلاعات مجموعه داده ارزیابی در بدست آوردن مقادیر حداقل و حداکثر استفاده نشده است چون در شرایطی عملی داده‌های ارزیابی توسط ساختار یادگیری ماشین دیده نشده و ما از مقادیر آن اطلاعی نداریم بنابراین باید عملیات نرمال سازی را بر اساس داده‌های آموزش انجام دهیم و در ادامه هنگام ارزیابی مدل با استفاده از مقادیر بدست آمده از مجموعه داده آموزش، مجموعه داده ارزیابی را نرمال کنیم.

## بدون استفاده از کتابخانه های آمادۀ پایتون، مدل طبقه بند، تابع اتلاف و الگوریتم یادگیری و ارزیابی را کدنویسی کنید تا دو کلاس موجود در دیتاست به خوبی از یکدیگر تفکیک شوند. نمودار تابع اتلاف را رسم کنید و نتیجۀ ارزیابی روی داده های تست را با حداقل ۲ شاخصه محاسبه کنید. نمودار تابع اتلاف را تحلیل کنید. آیا می توان از روی نمودار تابع اتلاف و قبل از مرحلۀ ارزیابی با قطعیت در مورد عمل کرد مدل نظر داد؟ چرا و اگر نمی توان، راه حل چیست؟

مدل یادگیری خطی Perceptron به‌منظور پیاده‌سازی انتخاب شد و با کمک ChatGPT کد آن نوشته شده است. این مدل با نرخ یادگیری 0.01، تابع هزینه MSE و کاهش نرخ یادگیری به ازای هر 10 epoch بدون بهبود در هزینه تابع ارزیابی آموزش دیده است.  
نمودارهای مربوط به این آموزش و ارزیابی‌های آن به شکل زیر است.





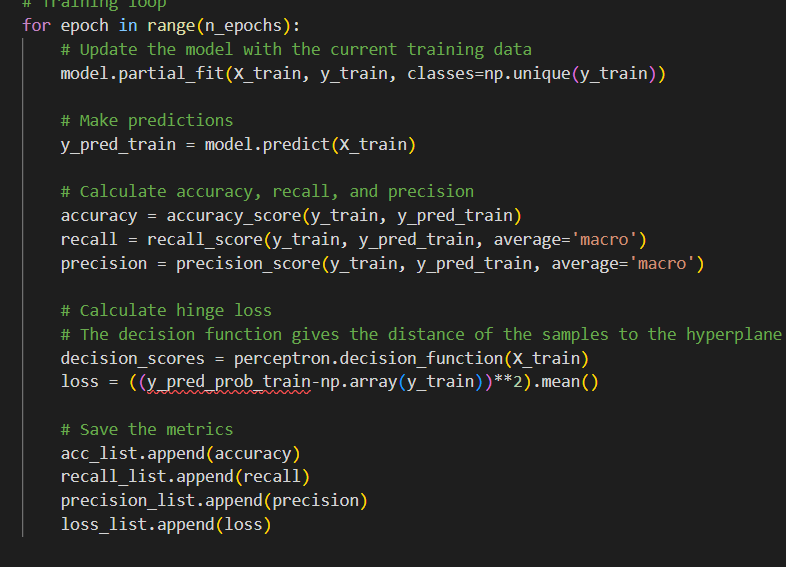
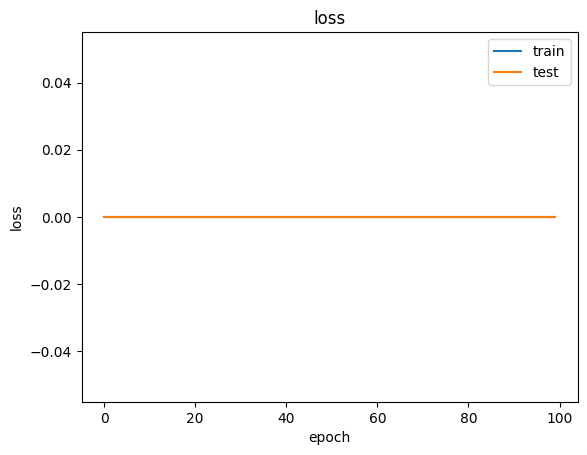
همان‌طور که مشخص است نمودار تابع هزینه هم برای مجموعه‌داده آموزش و هم ارزیابی به‌صورت نزولی عمل کرده و تا نزدیکی مقدار صفر رفته است نکته حائز اهمیت در این نمودار، کم‌بودن مقدار تابع هزینه در همان epoch اول است که نشان می‌دهد این مجموعه‌داده از تفکیک‌پذیری خوبی برخوردار است.

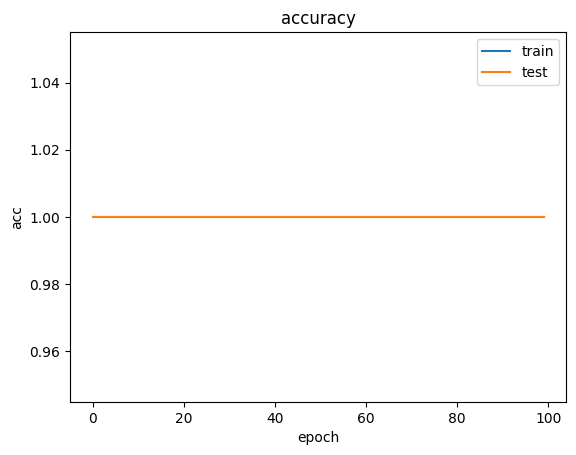
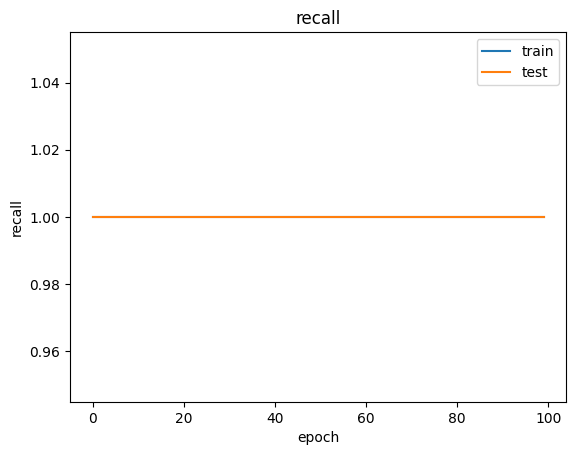
نمودار فوق مربوط به ترسیم ویژگی میانگین به ویژگی حداکثر مقدار داخل پنجره برای مجموعه‌داده ارزیابی است و همان‌طور که مشخص است این مجموعه‌داده به‌صورت ذاتی از تفکیک‌پذیری مناسبی برخوردار است و نتایج به‌دست‌آمده از آموزش کاملاً درست هستند. تنها ابهام موجود مربوط به نمودار recall است که از ابتدا 1 می‌باشد. دلیل رخ دادن این مسئله این است که مقدار این metric در انتهای epoch اول و بعد از بروزرسانی وزن‌های مدل محاسبه شده است بنابراین چون تفکیک پذیری مجموعه داده بالا است و مدل به راحتی در epoch اول به نتیجه مطلوب رسیده این معیار از همان ابتدا 1 است.

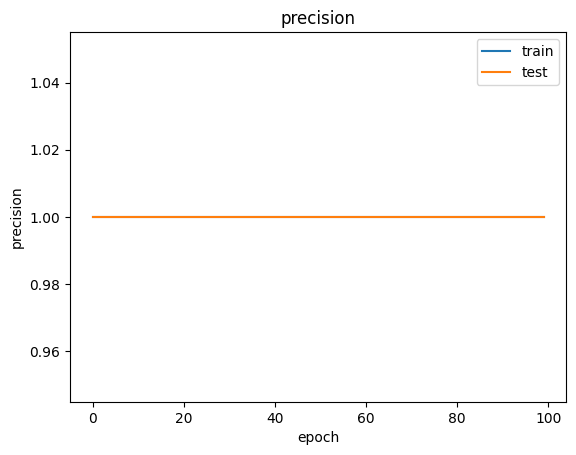
برای اظهار نظر درمورد عملکرد مدل به روی مجموعه‌داده ارزیابی با استفاده از نمودار تابع هزینه مجموعه‌داده آموزش، می‌توان گفت که با کاهش سریع تابع هزینه در ابتدا، بهبود عملکرد مدل روی مجموعه‌داده ارزیابی که مبدائی یکسان با مجموعه‌داده آموزش دارد، انتظار می‌رود؛ اما به‌صورت دقیق نمی‌توان این مسئله را مشخص کرد و حتی ممکن است در شرایط خاص برای مجموعه‌داده ارزیابی نتیجه‌گیری اشتباه باشد. برای رفع این مشکل می‌توان یک قسمت از مجموعه‌داده آموزش را به‌عنوان مجموعه‌داده صحت‌سنجی انتخاب کرد و با بررسی آن روی عملکرد مدل روی مجموعه‌داده ارزیابی نظر داد.

## فرآیند آموزش و ارزیابی را با استفاده از یک طبقه بند خطی آمادۀ پایتون (در model\_linear.sklearn (انجام داده و نتایج را مقایسه کنید. در حالت استفاده از دستورات آمادۀ سایکیت لرن، آیا راهی برای نمایش نمودار تابع اتلاف وجود دارد؟ پیاده سازی کنید

با استفاده از کد فوق و کتابخانه sklearn، مدل perceptron آموزش دیده است که همانند نتیایج قبلی به دقت 1 رسیده است. برای ثبت نتایج هر epoch نیاز است تا از متد partial\_fit استفاده کنیم. این متد با هربار فراخوانی یک epoch مدل را آموزش می‌دهد و می‌توان بعد از آموزش مقادیر هزینه را ثبت نمود.

 کد فوق نمایش روش استفاده از partial\_fit را نشان می‌دهد که در ادامه نتایج ذخیره شده در این روش، نشان داده شده است.  
در ادامه نمودارهای مربوط به این آموزش نمایش داده شده است.

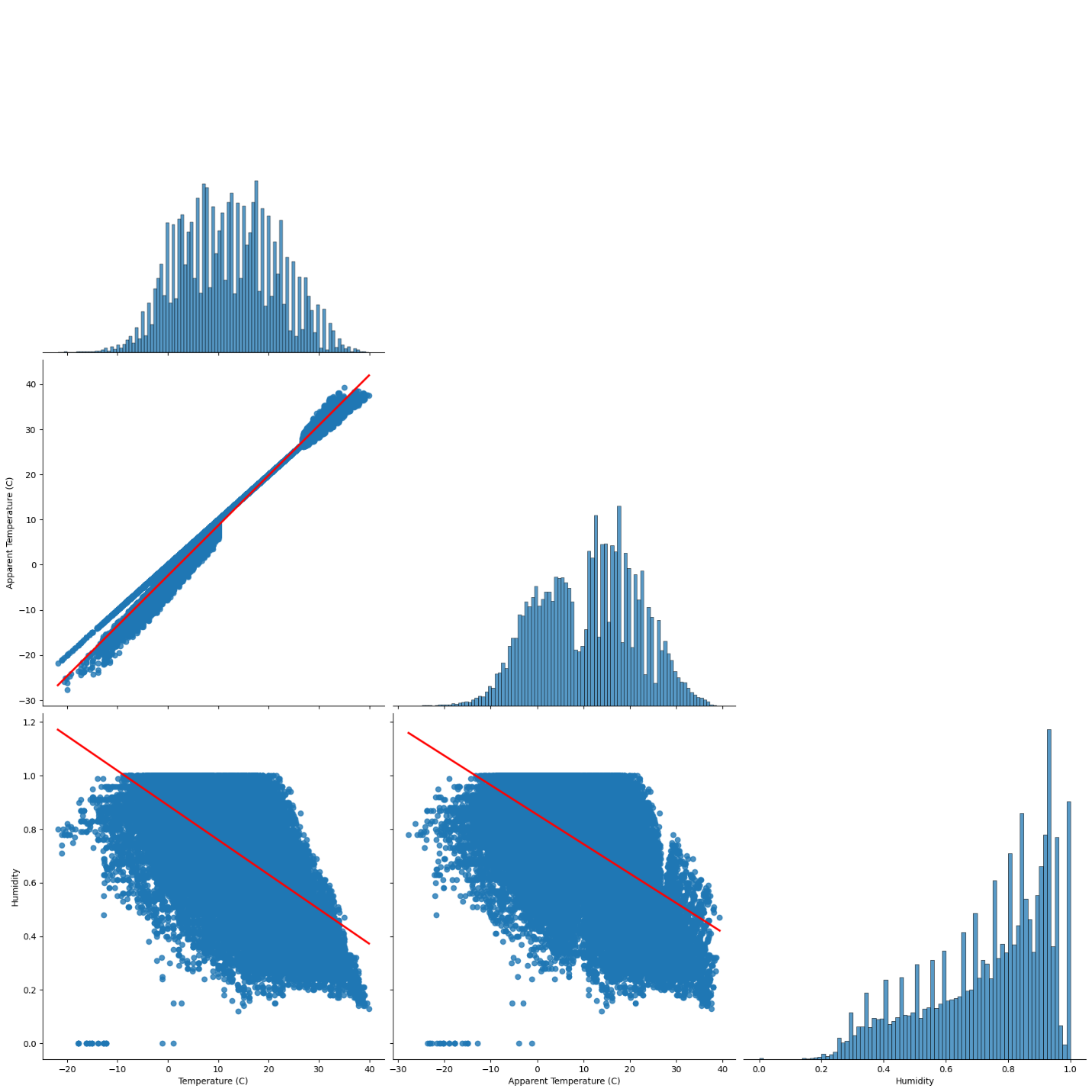


همانطور که مشخص است از اولین epoch نتیجه تابع هزینه این آموزش هم برای مجموعه‌داده آموزش و هم برای ارزیابی به به صفر رسیده و تمام معیارها به مقدار 1 رسیده‌اند علت تفاوت این نمودار با آموزش قبلی این است که الگوریتم‌های کتابخانه sklearn بهینه‌هستند و در همان epoch اول به بهترین میزان خود رسیده است.

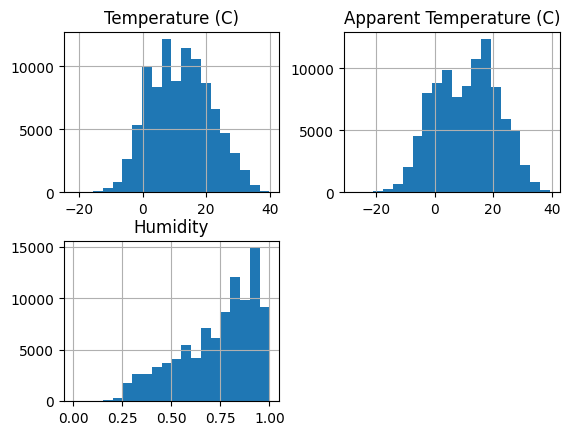
سوال3

## یک دیتاست در زمینۀ آب و هوا با نام 2006-2016 Szeged in Weather را در نظر بگیرید. در این دیتاست هدف آن است که ارتباط بین Humidity با Temperature و هم چنین ارتباط بین Humidity و Temperature Apparent پیدا شده و با کمک داده های Humidity و Temperature تخمین انجام شود.

## ابتدا هیت مپ ماتریس همبستگی و هیستوگرام پراکندگی ویژگی ها را رسم و تحلیل کنید

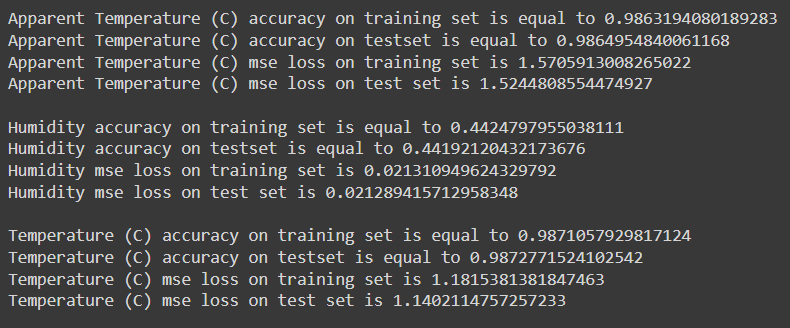
نمودار فوق مربوط به heat map داده‌های موجود است و همان‌طور که مشخص است واریانس رطوبت بسیار کم است که این به معنی تغییرات کم رطوبت هوا در زمان‌های متفاوت در محل اندازه‌گیری است. از سوی دیگر واریانس دما و دمای ظاهری در طول ساعات متفاوت بسیار زیاد است که این مسئله باتوجه‌به تفاوت دما در طول روز و شب به علت حضور خورشید یا شرایط متفاوت جوی مانند وزیدن باد منطقی است. از سوی دیگر با توجه‌به منفی بودن ضریب covariance مشخص است که افزایش دمای هوا و دمای قابل احساس، تاثیر منفی روی رطوبت هوا دارد و عکس این رابطه نیز صحیح می‌باشد. ضریب covariance مربوط به دما و دمای حسی مقداری مثبت و بزرگ می‌باشد که نشان دهنده این است که افزایش هریک از این متغییرها تاثیر مستقیم و زیادی روی دیگری خواهد گذاشت.

نمودار فوق مربوط به رسم نقاط دو مولفه از میان دما، دمای قابل احساس و رطوبت نسبت به همدیگر می‌باشد و نمودارهای روی قطر هیستوگرام پراکندگی داده می‌باشد باید توجه کرد که محور‌ها نرمال شده‌اند.

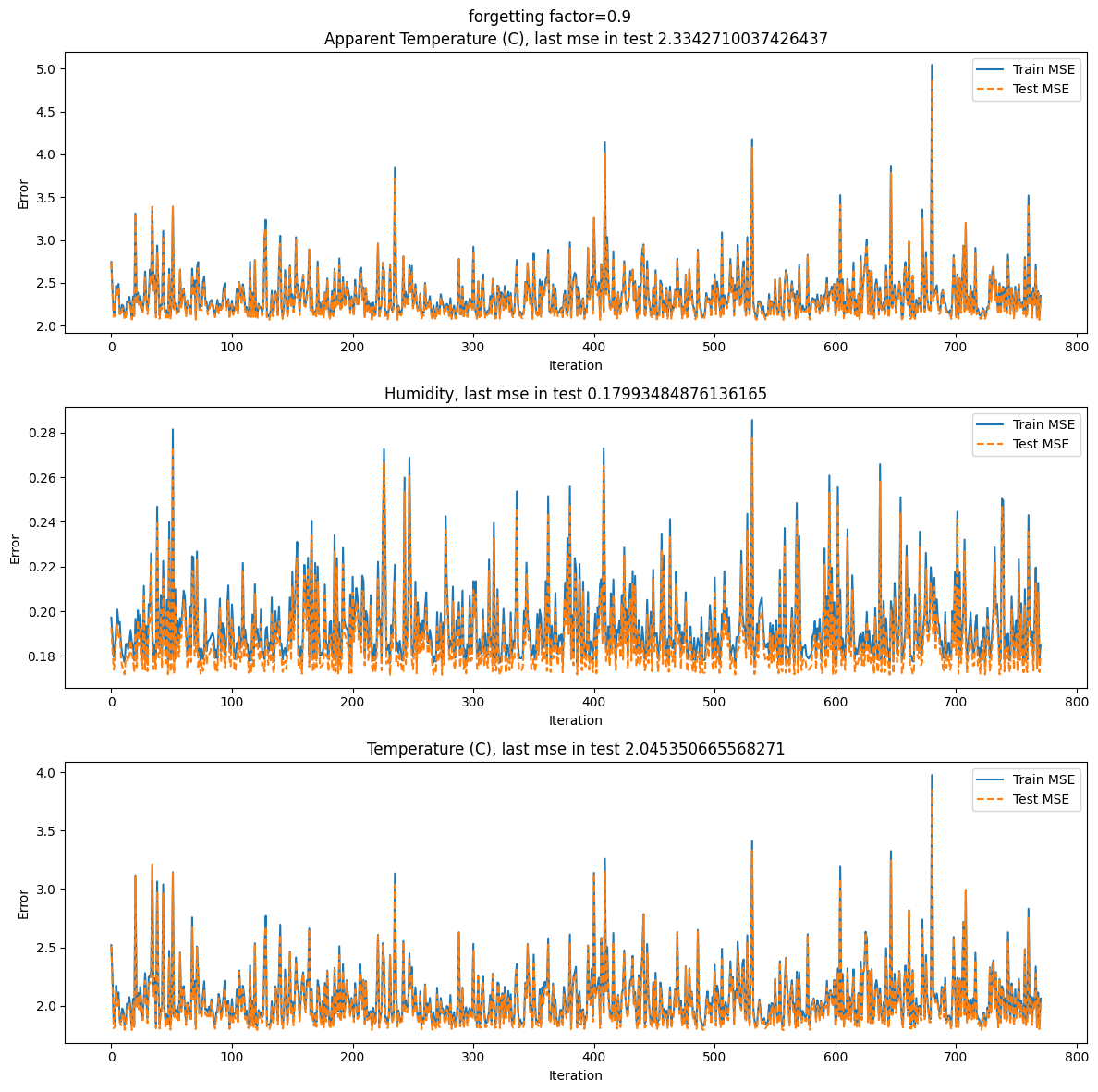
نمودار فوق هیستوگرام حسگرها به صورت نرمال نشده است و همانطور که مشخص است چون رطوبت مقداری بین 0 و 1 دارد میزان واریانس آن به نسبت کم شده است. از طرف دیگر نمودارهای دما نشان‌دهنده توضیحی نرمال است.

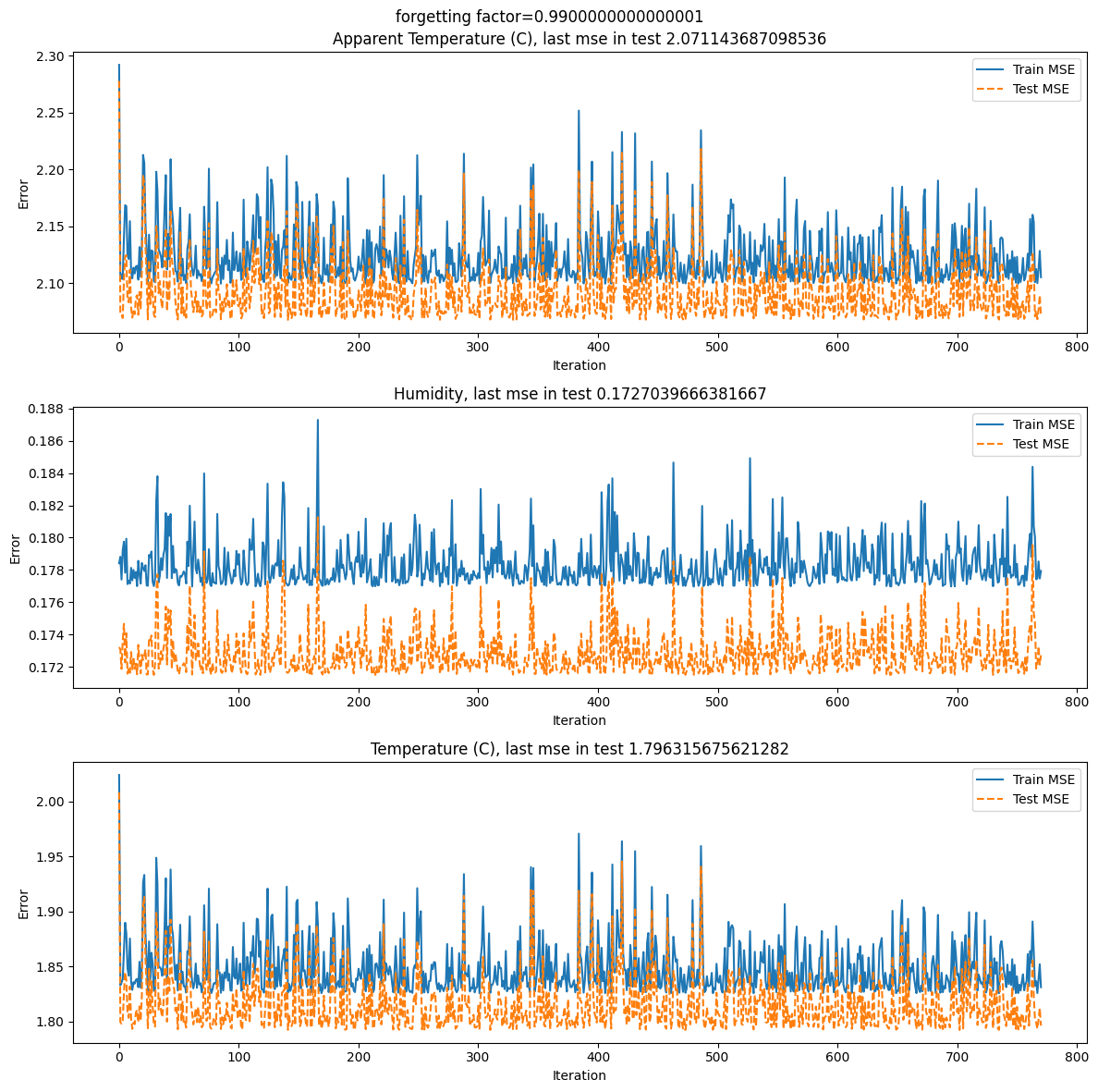
## روی این دیتاست، تخمین LS و RLS را با تنظیم پارامترهای مناسب اعمال کنید. نتایج به دست آمده را با محاسبۀ خطاها و رسم نمودارهای مناسب برای هر دو مدل با هم مقایسه و تحلیل کنید.

با استفاده از کتابخانه sklearn مدل LinearRegression را پیاده سازی شده است و مجموعه‌داده با نسبت 0.2 برای مجموعه داده ارزیابی و 0.8 برای مجموعه آموزش جدا شده است و نتایج آن به شرح زیر است.



در ادامه الگوریتم RLS پیاده سازی شده و بهترین میزان برای forgetting factor ابتدا از بین 0.1 تا 0.9 با گام 0.1 بدست آمد و سپس از بین 0.9 تا 0.99 با گام 0.01 انتخاب شد که بهترین نتیجه به ازای 0.99 می‌باشد.



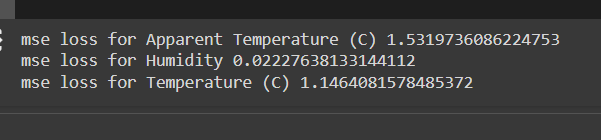


نمودارهای فوق مربوط به forgetting factorهای 0.9 و 0.99 است و بقیه نمودارها در فایل Q3.ipynb می‌باشد. همانطور که مشخص است میزان mse برای LS کمتر از RLS می‌باشد بنابراین در این مقایسه الگوریتم LS عملکرد بهتری دارد.

## در مورد Square Least Weighted توضیح دهید و آن را روی دیتاست داده شده اعمال کنید.

Weighted least square بر پایه least square بنا شده و با اضافه کردن یک ماتریس مربعی به هر نمونه از داده‌های جمع‌آوری شده وزنی مشخص را نسبت می‌دهد که این مسئله می‌تواند بعضی ایرادات کمی مجموعه داده را برطرف کنند به عنوان نمونه در یک مجموعه داده نامتعادل با افزایش وزن داده‌هایی که کمتر هستند یا افزایش وزن داده‌هایی که از اهمیت بیشتری برخوردار هستند می‌توان به نتیجه مطلوب‌تری رسید. رابطه مربوط به WLS به شکل زیر است.

در حل این سوال به منظور لحاظ کردن عنصر زمان در محاسبه بهترین وزن‌ها، به هر نمونه از مجموعه داده وزنی معکوس شماره مکانی آن داده شده است. نتایج آموزش به شرح زیر می‌باشد.



همانطور که مشخص است این مدل عملکرد بهتری نسبت به RLS دارد اما با اختلاف کمی از LS عملکرد ضعیف‌تری دارد.

1. <https://engineering.case.edu/bearingdatacenter> [↑](#endnote-ref-1)
2. [XiongMeijing/CWRU-1: Multiclass classification of vibration signals of faulty bearings (github.com)](https://github.com/XiongMeijing/CWRU-1) [↑](#endnote-ref-2)
3. Yoo, Y., Jo, H. and Ban, S.W., 2023. Lite and efficient deep learning model for bearing fault diagnosis using the CWRU dataset. *Sensors*, *23*(6), p.3157. [↑](#endnote-ref-3)