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.

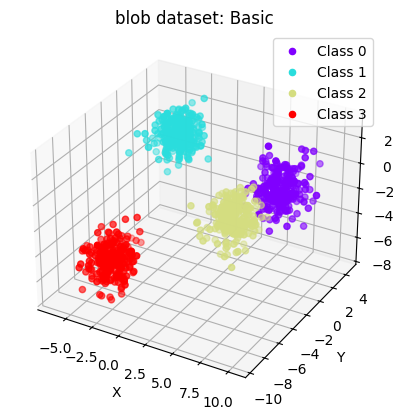
 Loss Function: این بخش از ساختار یادگیری ماشین شامل انواع گوناگونی از توابع هزینه مانند MSE، Cross-Entropy و یا انواع دیگر توابع هزینه می‌شود. تابع هزینه خصوصاً برای الگوریتم‌هایی که شامل بهینه‌سازی از طریق مشتق هستند دارای اهمیت ویژه‌ای است؛ زیرا توابع هزینه‌ای که مشتق‌پذیر نیستند یا مشتق آنها در نواحی 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.

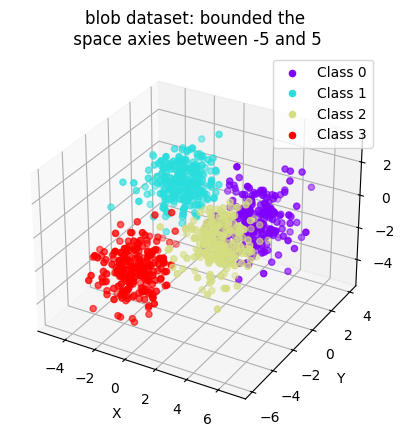
Optimizer: این قسمت از ساختار یادگیری ماشین شامل راه‌حل ریاضی برای بهینه‌سازی وزن‌های داخلی مدل یادگیری ماشین است. با درنظرگرفتن ویژگی‌های مجموعه‌داده و انتخاب صحیح الگوریتم بهینه‌سازی می‌توانیم زمان آموزش و عملکرد ساختار یادگیری ماشین را بهبود ببخشیم. یک الگوریتم بهینه‌سازی مناسب باعث می‌شود تا مدل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

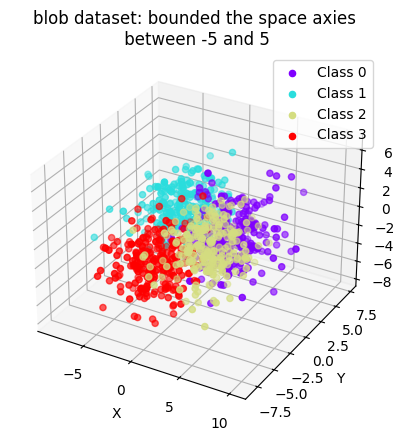
 Decision Policy Module: این قسمت از ساختار یادگیری ماشین باتوجه‌به وظیفه ساختار، یکی از انواع تبدیل‌های موجود را روی خروجی پیاده می‌کند. این تبدیل برای وظیفه طبقه‌بندی (Classification) می‌تواند تبدیل خروجی احتمالاتی به خروجی دو وضعیتی باشد که از آن با عنوان One-Hot Transform یاد می‌شود. همچنین در این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

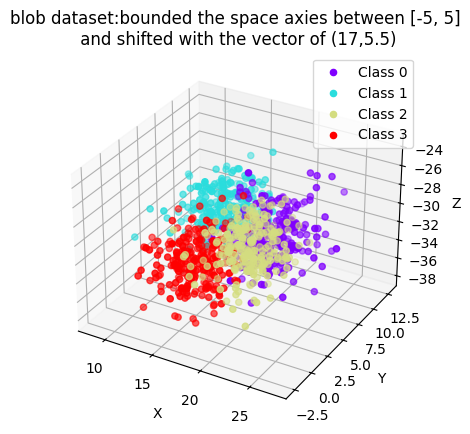
 به‌منظور تغییر طبقه‌بندی از ۲ کلاس به چند کلاس، می‌تواند در تمام قسمت‌های ساختار تأثیر بگذارد بااین‌حال مهم‌ترین قسمت شامل Loss Function و Optimizer است که باید برای استفاده به‌عنوان یک طبقه‌بند چند کلاسه با شرایط خروجی مدل سازگاری داشته باشند. به‌عنوان نمونه BCELoss در کتابخانه PyTorch به‌منظور استفاده 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.

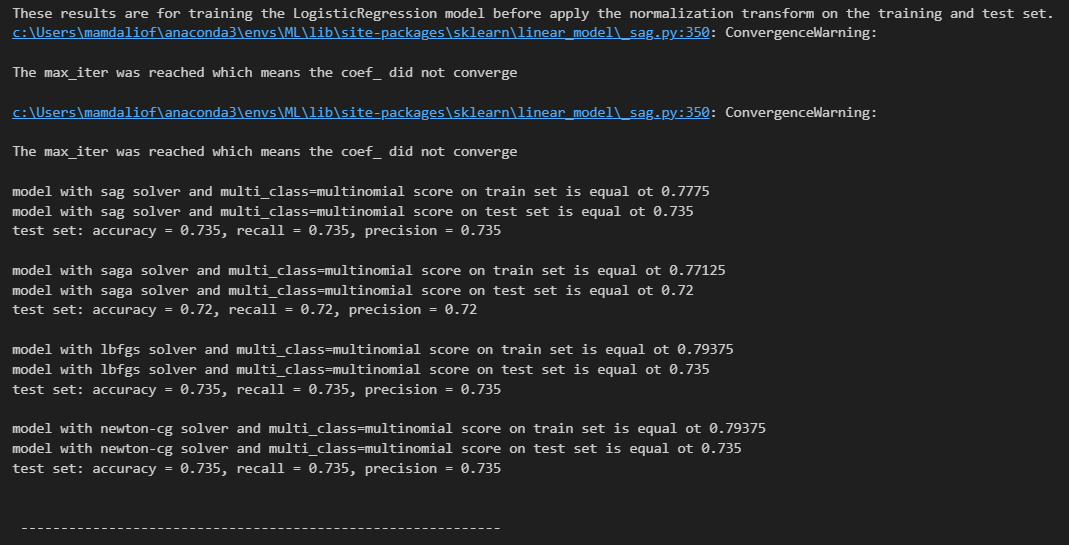
همان‌طور که از این تصویر مشخص است تفکیک‌پذیری داده‌ها در این مجموعه‌داده پالش برانگیز نمی‌باشد و تنها چالش این مجموعه‌داده جداسازی نقاط بنفش (class0) از نقاط زرد (class2) است. در ادامه به‌منظور چالش‌برانگیز کردن این مجموعه‌داده از میان پارامترهای قابل‌تنظیم در متد make\_blobs، center\_box= [-5, 5] قرار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

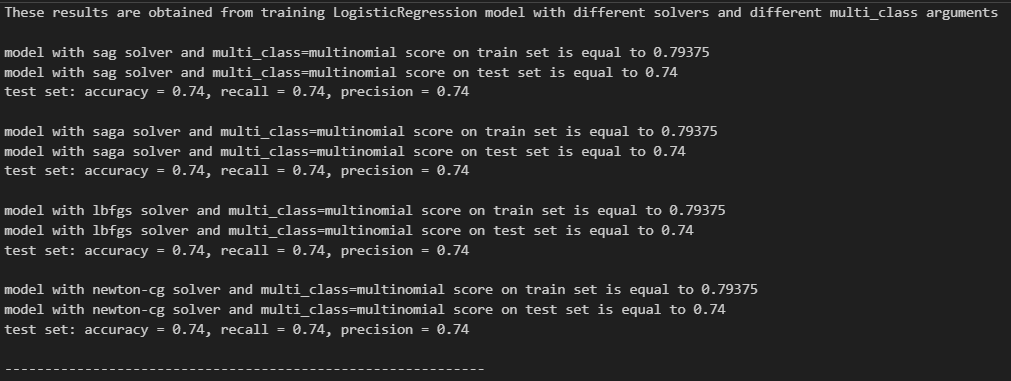
## با استفاده از حداقل دوطبقه بند خطی آمادۀ پایتون (در model\_linear.sklearn (و در نظر گرفتن فراپارامترهای مناسب، چهار کلاس موجود در دیتاست قسمت قبلی را از هم تفکیک کنید. ضمن توضیح روند انتخاب فراپارامترها (مانند تعداد دورۀ آموزش و نرخ یادگیری)، نتیجۀ دقت آموزش و ارزیابی را نمایش دهید. برای بهبود نتیجه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

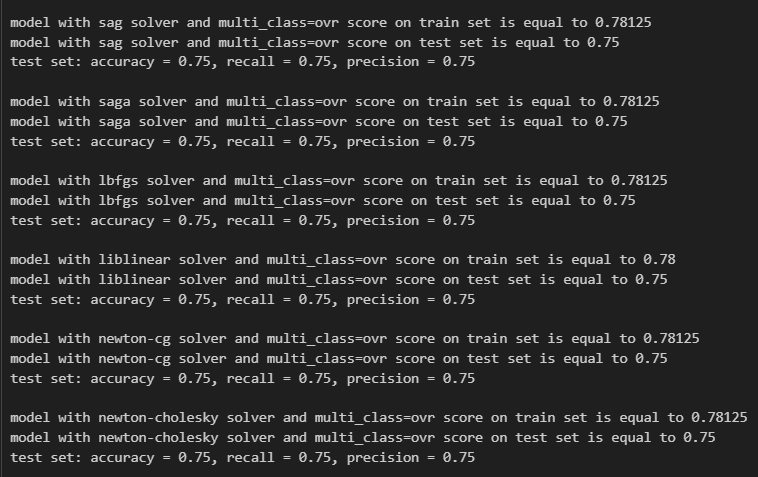
در ادامه از دو طبقه‌بند خطی LogisticRegression و SGDClassifier استفاده شده است و با استفاده از کتابخانه sklearn پیاده‌سازی آن انجام شده است. این نکته حائز اهمیت است که 20 درصد از مجموعه داده مذکور برای انجام عمل ارزیابی و 80 درصد برای عمل آموزش جدا شده است.

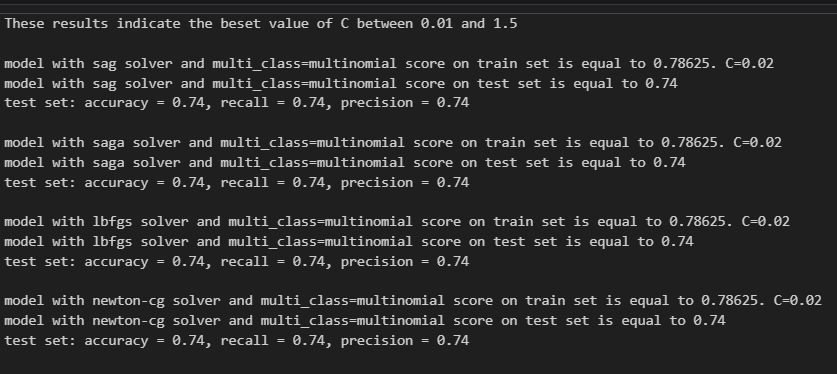
در ابتدا این مدل روی مجموعه‌داده خام آموزش و به‌ازای دو رویکرد متفاوت multinomial و ovr و تمام solverهای موجود آموزش‌دیده است که نتایج آن به شرح زیر است.

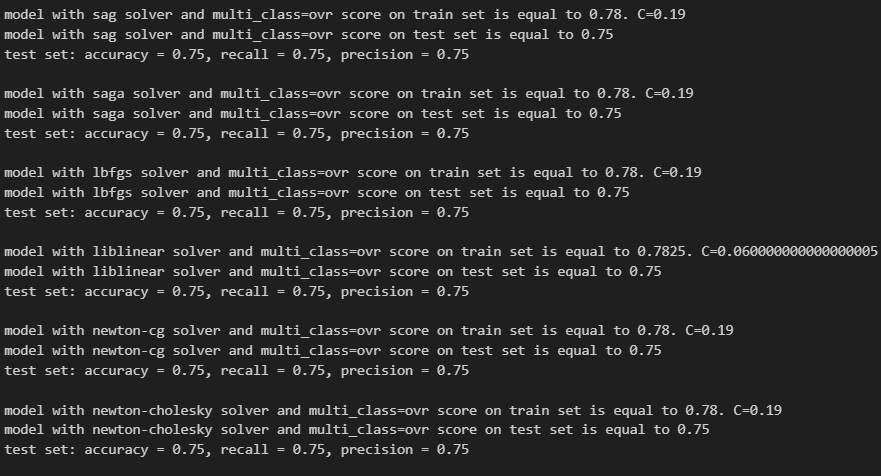
همان‌طور که مشخص است تعدادی از الگوریتم‌ها همگرا نشده‌اند و نتیجه این آموزش از لحاظ بار محاسباتی و دقت مناسب نمی‌باشد.

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

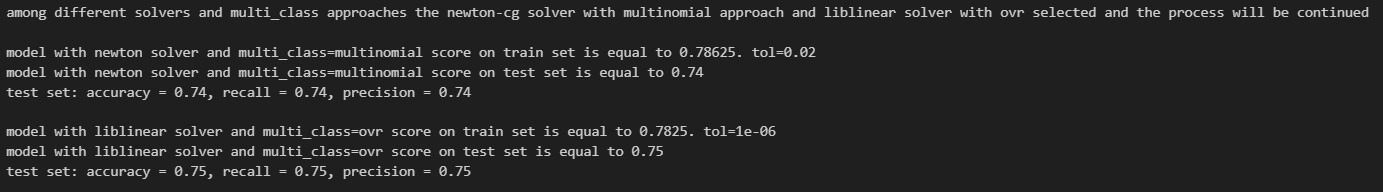




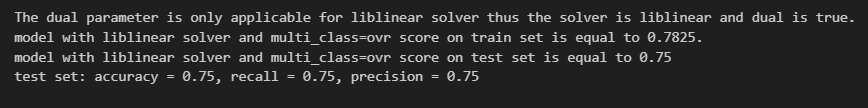
در ادامه بین بازه 0.01 تا 1.5 با گام 0.01 مقادیر مختلفی برای پارامتر C که میزان Regularization را در فرایند آموزش تعیین می‌کند بررسی شده است تا بهترین مقدار محاسبه شود. نتایج این مجموعه آموزش به شرح زیر است:  




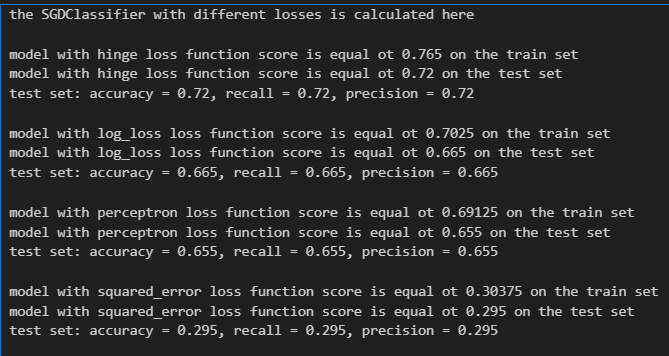
در ادامه دو مدل از میان تمام مدل‌های موجود انتخاب شده و باقی آموزش‌ها بر روی این مدل‌ها صورت خواهد گرفت. فراپارامتر بعدی که مقدار بهینه آن باید محاسبه شود tol است. مقدار این فراپارامتر در بازه تا عوض شده است تا حداقل مقدار خطای قابل قبول برای مجموعه‌داده آموزش محاسبه شود. این آموزش‌ها در شرایطی انجام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

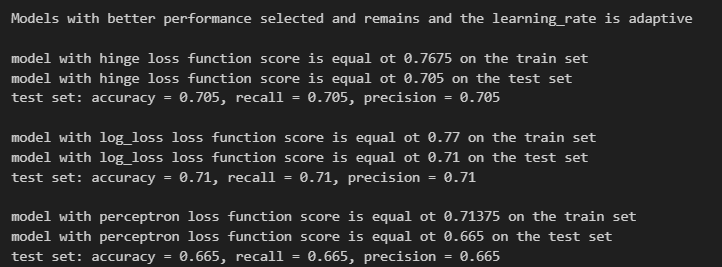


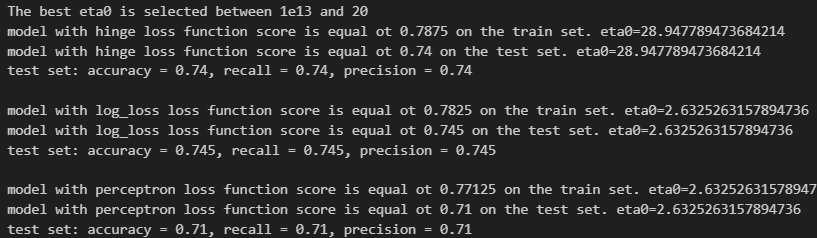
در انتها فراپارامتر dual برای solver liblinear فعال شده است تا تغییراتی که این فراپارامتر در آموزش ایجاد می‌کند مشاهده شود. نتایج حاصل شده در ادامه قابل‌مشاهده است.

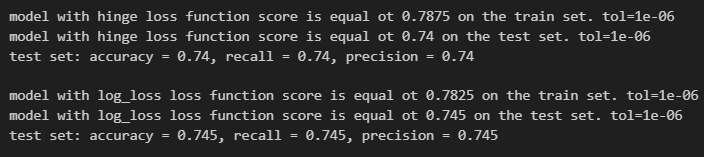


حال مدل SGDClassifier به درنظرگرفتن lossهای موجود آموزش‌دیده شده است که نتایج آن در ادامه قابل‌مشاهده است. در ابتدا مدل روی داده‌های نرمال آموزش‌دیده است.

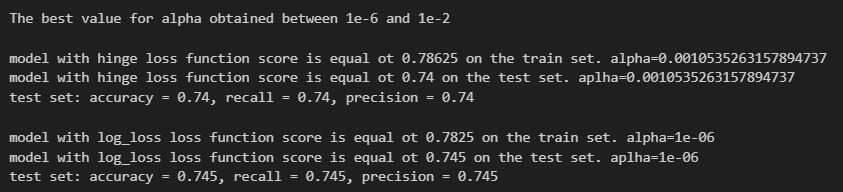
همان‌طور که مشخص است بعضی از مدل‌ها عملکرد مناسبی از خود نشان نداده‌اند که در نتیجه از ادامه فرایند آموزش حذف خواهند شد. در ادامه فراپارامتر  learning\_rate='adaptive' قرار داده شده است و نتایج آموزش آن قابل نسبت به نتایج گذشته بهتر است.

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 بدست آوردن 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 مجموعه داده valdidation به ازای مقادیر 0.1و 0.15و 0.2 محاسبه شده است.*

**

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

LogisticRegression(penalty="l2",

                            dual=True,

                            C=0.06,

                            solver="liblinear",

                            max\_iter=1500,

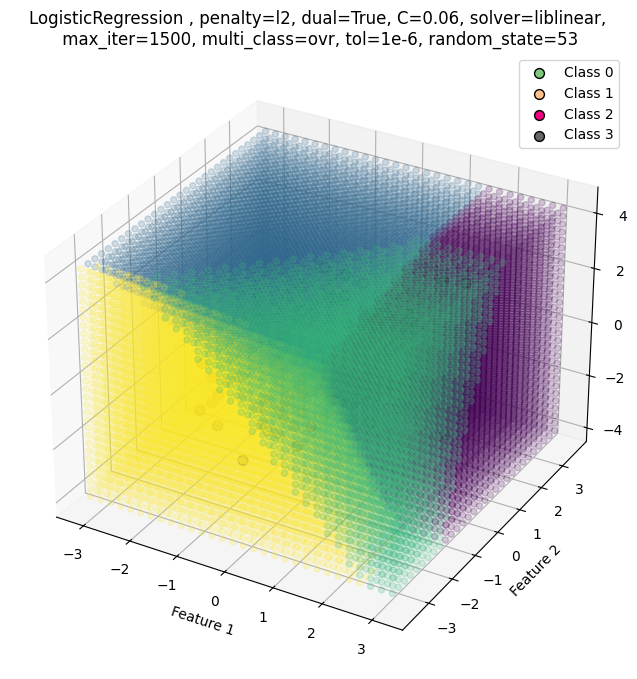
                            multi\_class="ovr",

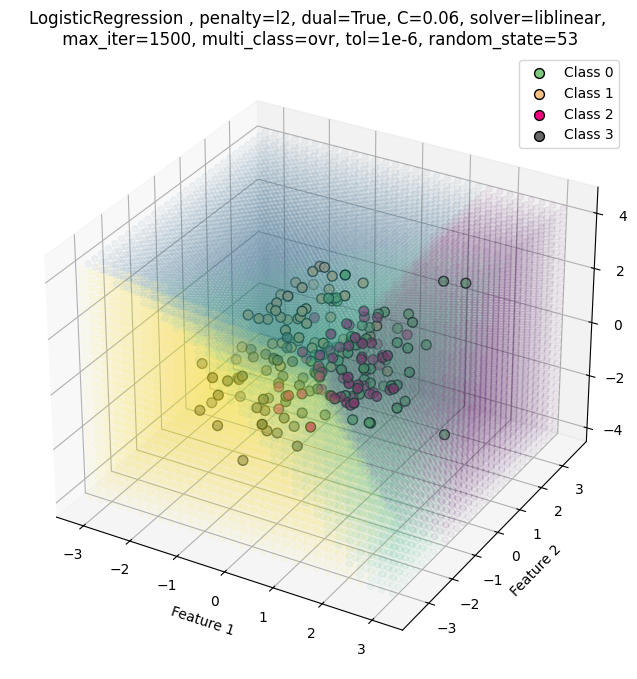
                            tol=1e-6,

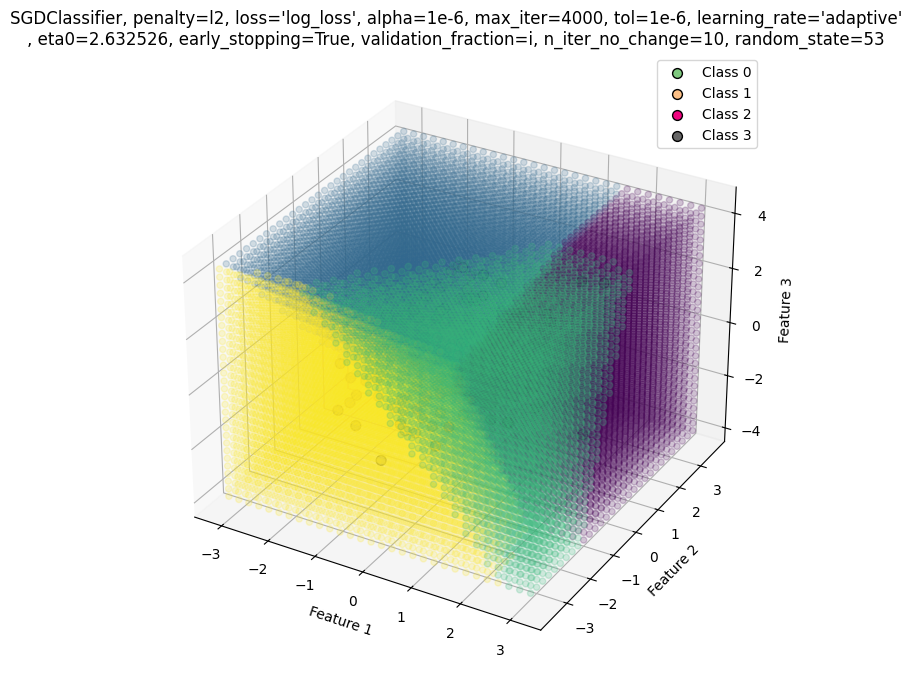
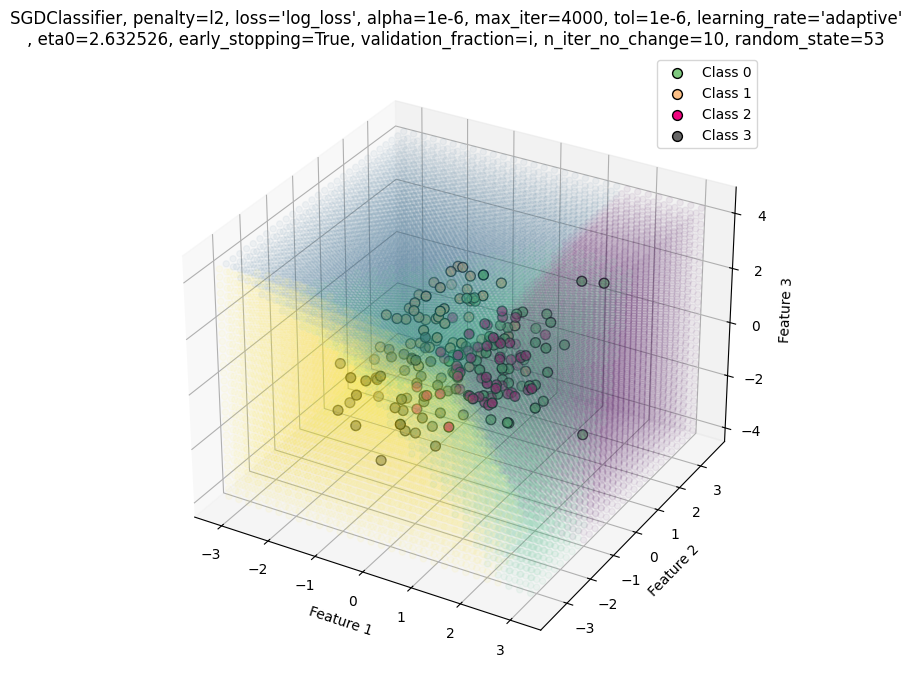
                            random\_state=53)

## 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.

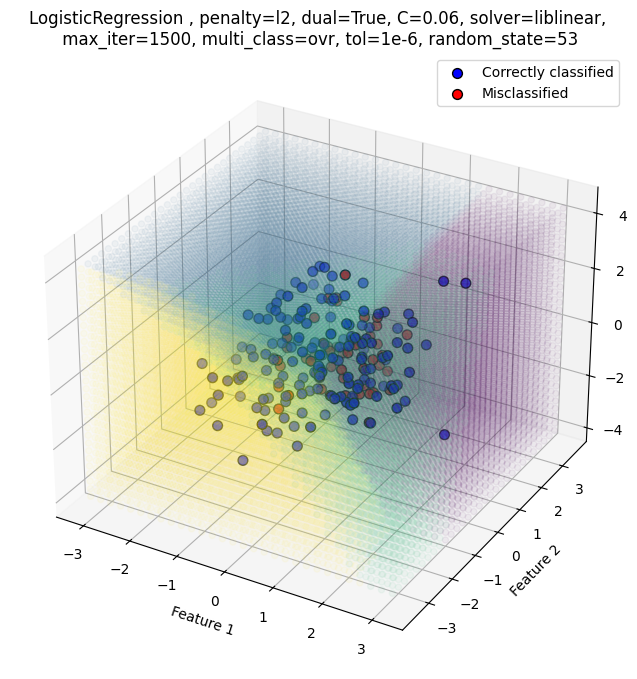


نمایش این نمودار به صورت interactive در داخل فایل Q1.1.ipynb قابل رویت است.



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

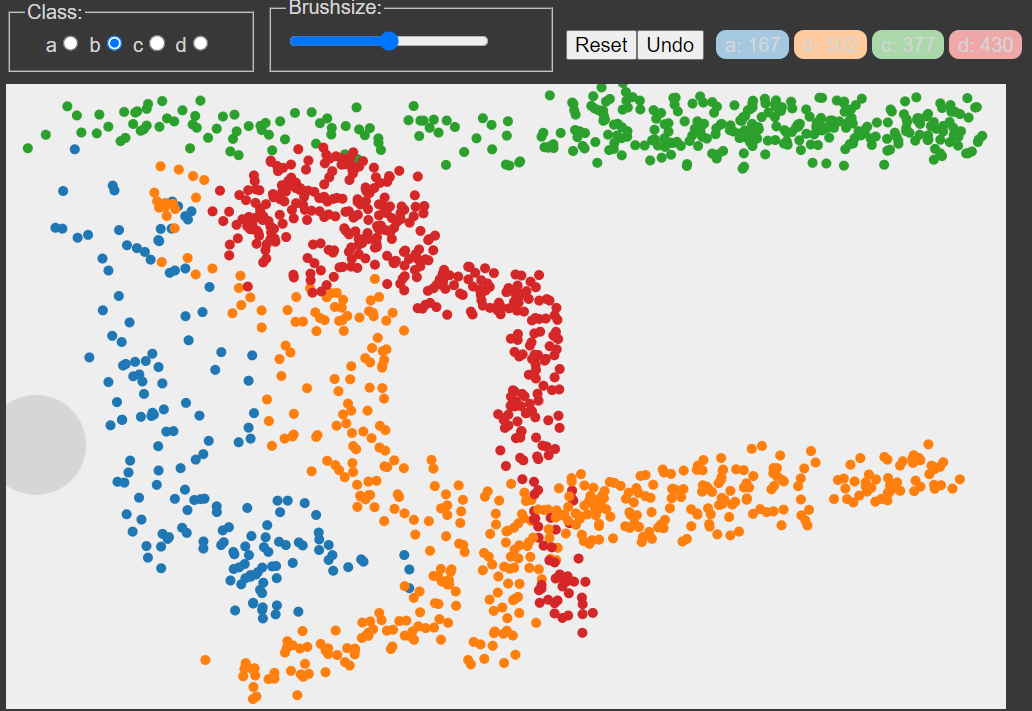
در ادامه نقاطی که درست پیش‌بینی شده اند را نسبت به نقاطی که اشتباه پیش‌بینی شده‌اند نمایش داده شده است.





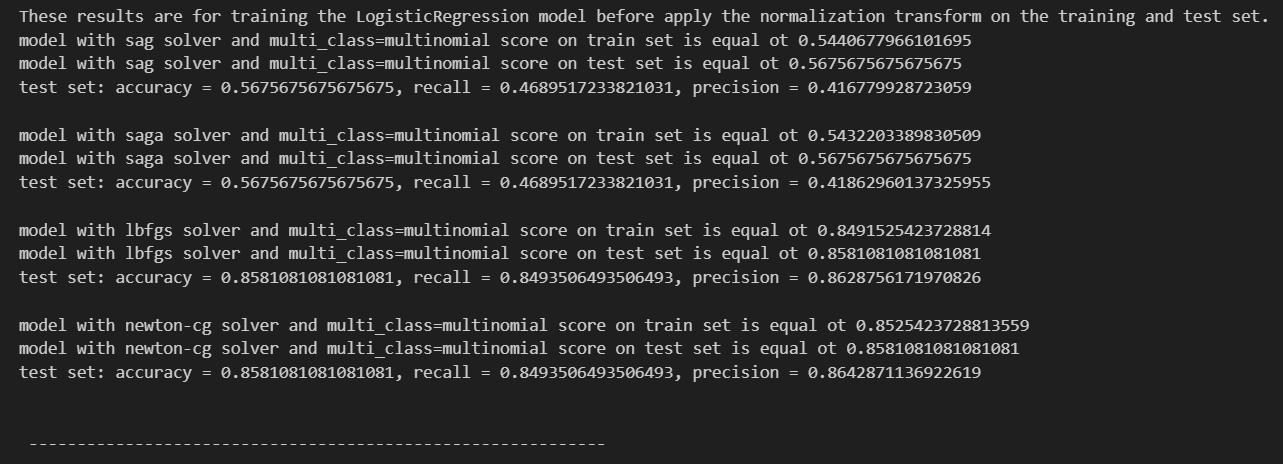
## فرآیندی مشابه قسمت «۲» را با تعداد کلاس و ویژگی دلخواه؛ اما با استفاده از ابزار drawdata تکرار کنید. قسمت های «۳» و «۴» را برای این داده های جدید تکرار و نتایج را به صورتی مناسب نشان دهید.

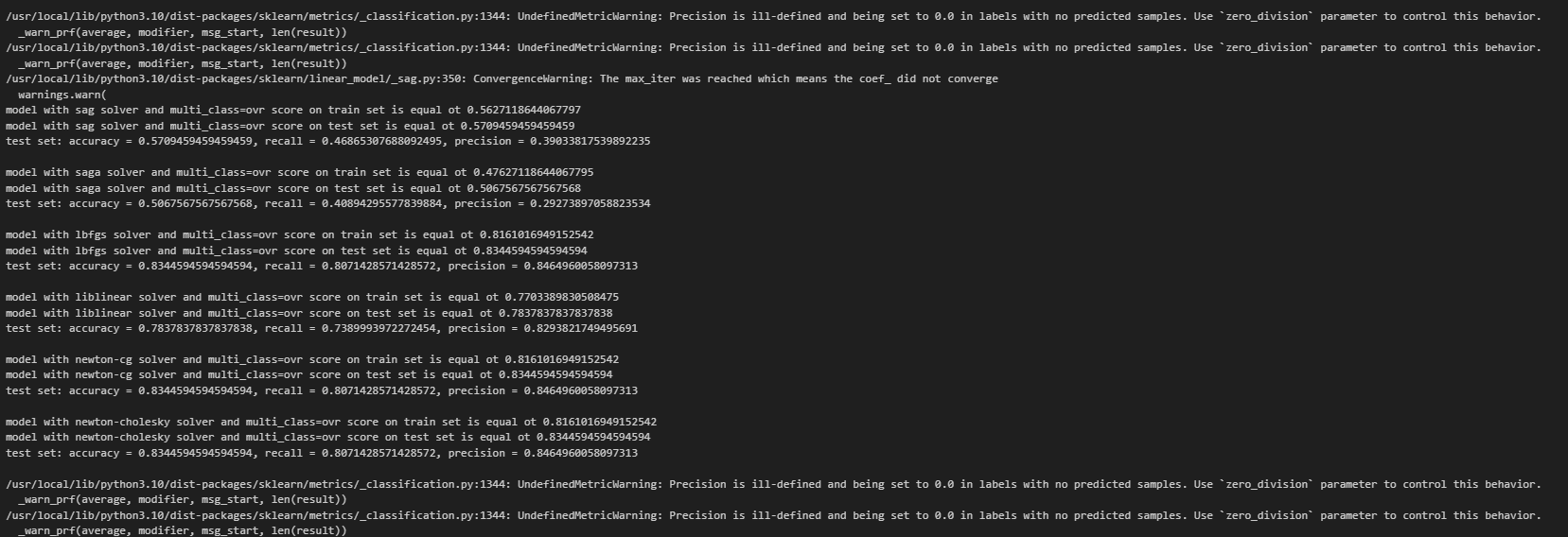
در ابتدا مجموعه داده را با استفاده از کتابخانه drawdata ایجاد می‌کنیم.



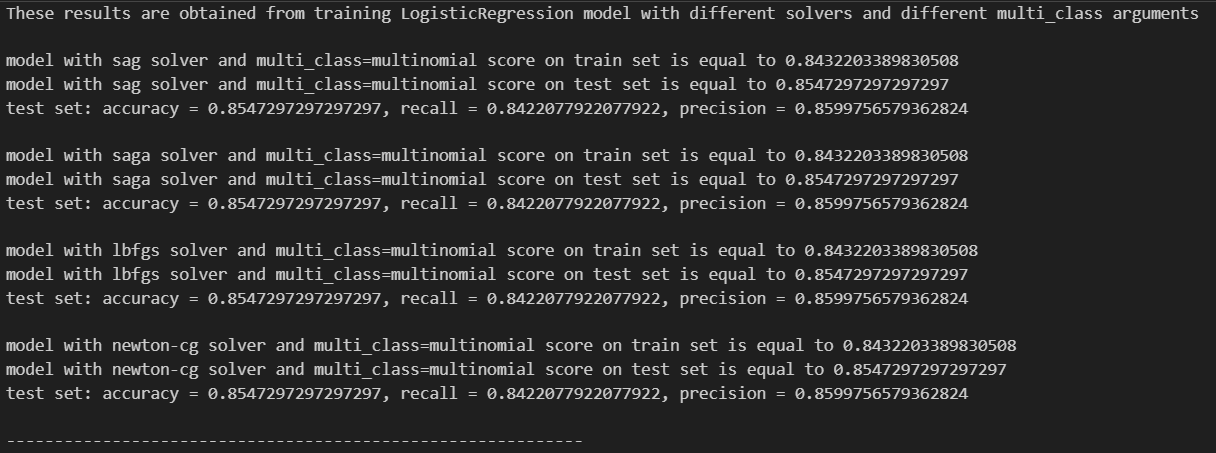
مجموعه داده فوق به منظور آموزش و ارزیابی ساختار یادگیری ماشین استفاده شده است. 20 درصد از این مجموعه داده برای ارزیابی و 80 درصد آن برای آموزش استفاده شده است.

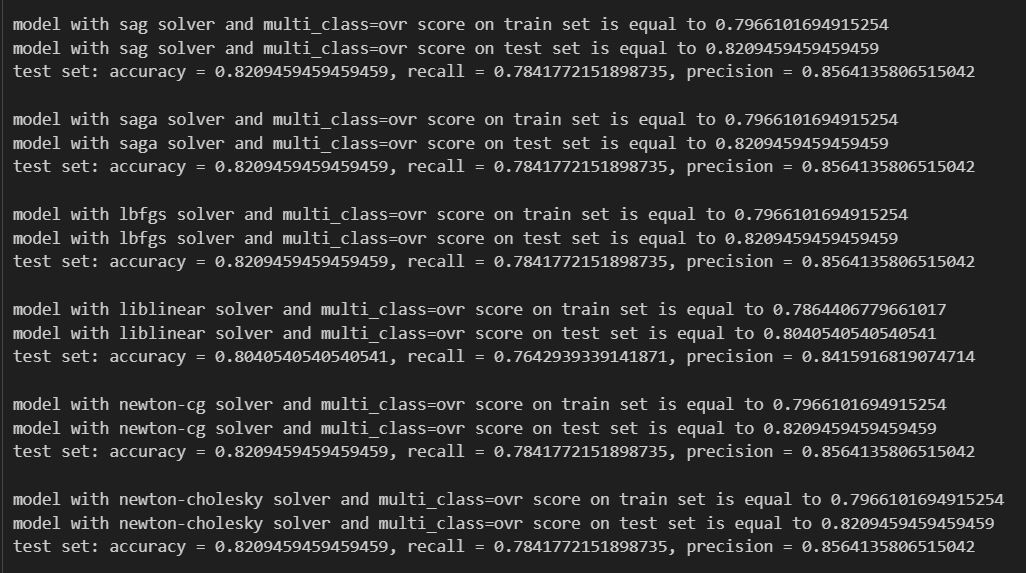
تمام مراحلی که در قسمت قبل توضیح داده شد را روی این مجموعه داده پیاده می‌کنیم که تصاویر نتایج بخش‌های گوناگون به ترتیب زیر می‌باشد.



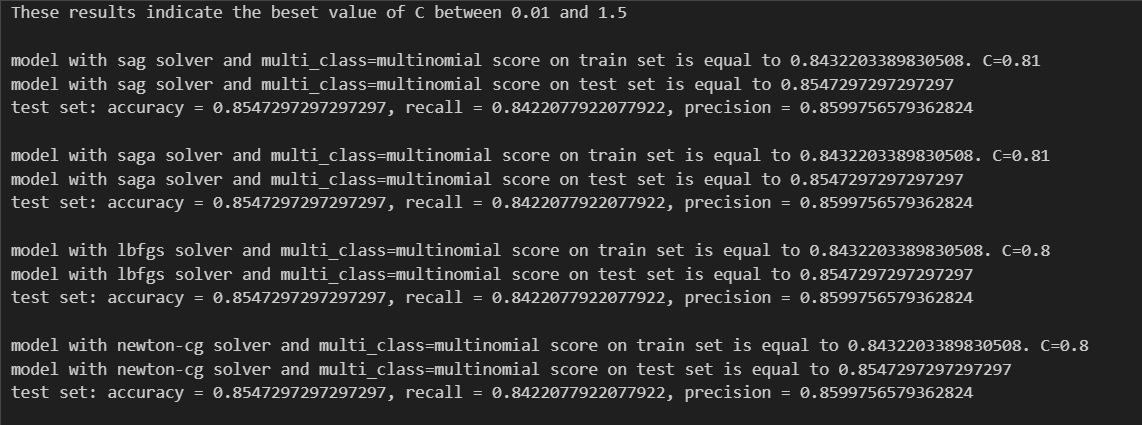


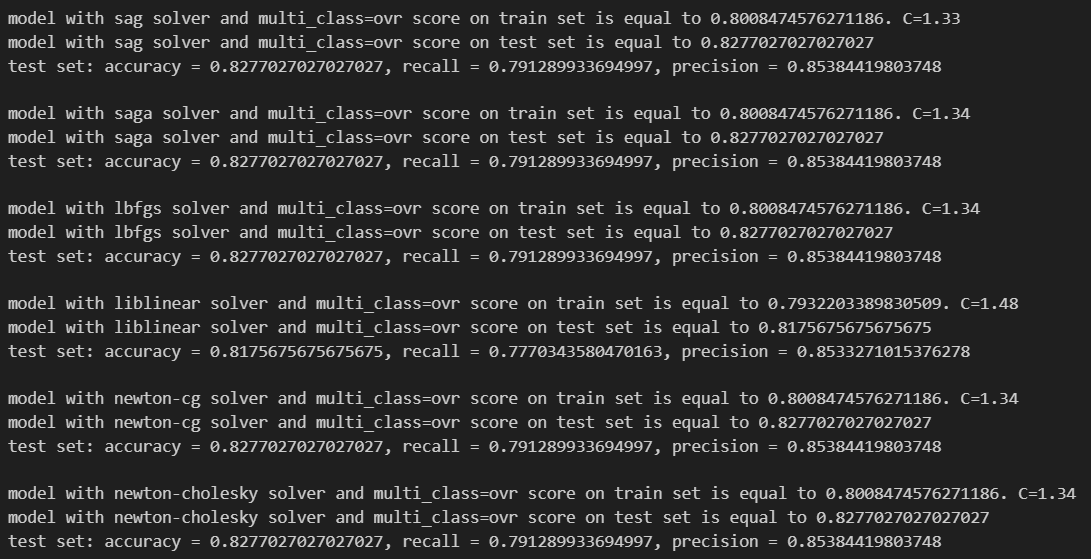
همانطور که مشخص است پیش از اعمال normalization مدل‌ها همگرا نشده‌اند که این به معنی بهینه نبودن فرایند آموزش و عملکرد مدل‌ها است. در ادامه مدل روی داده‌های نرمال شده آموزش داده شده است.





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

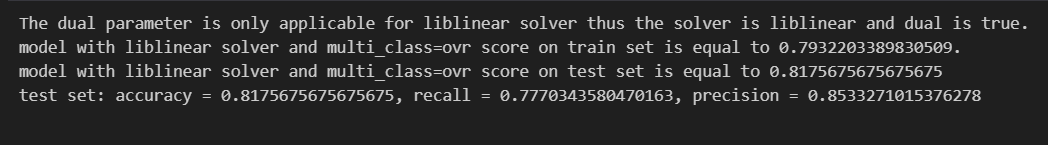


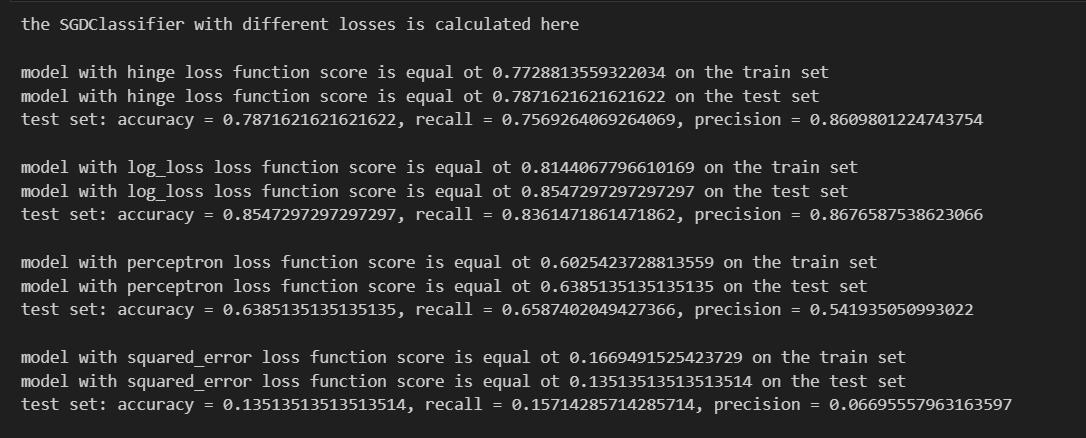


در گام بعدی دو مدل که عملکرد بهتری دارند را نگه می‌داریم و بهترین میزان tol را برایشان محاسبه می‌کنیم.

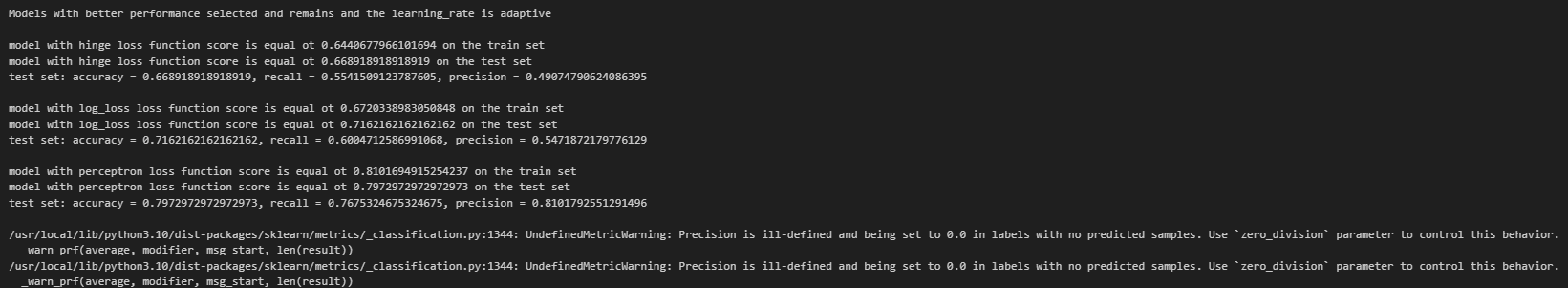


در انتها فراپارامتر dual را برای liblinear فعال می‌کنیم و مدل را آموزش می‌دهیم.

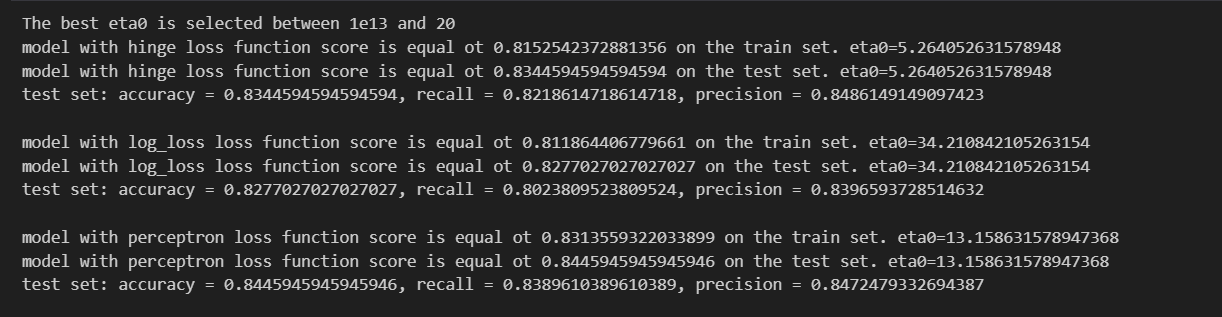


بهترین مدل LogisticRegresion در آخرین آموزش حاصل شده است. حال به سراغ مدل SGDClassifier می‌رویم.

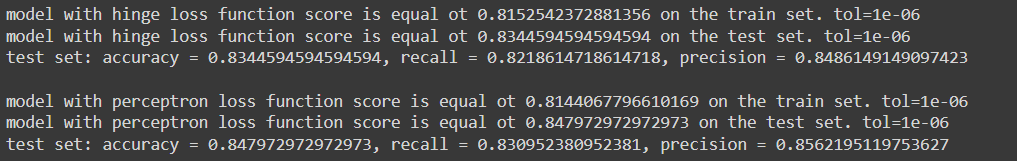
تصویر فوق نشان می‌دهد که این مدل به روی دیتای نرمال شده چگونه عمل می‌کند.



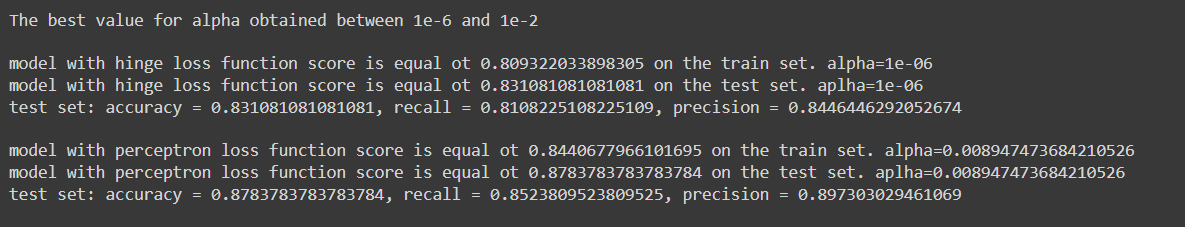
همانطور که مشخص است عملکرد بعضی مدل‌ها کاهش پیدا کرده اما این به دلیل مقدار eta0 اولیه می‌باشد و با اصلاح آن نتیجه بهتر خواهد شد بنابراین به سراغ بهینه‌سازی مقدار eta0 می‌رویم.



با بهینه سازی 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.

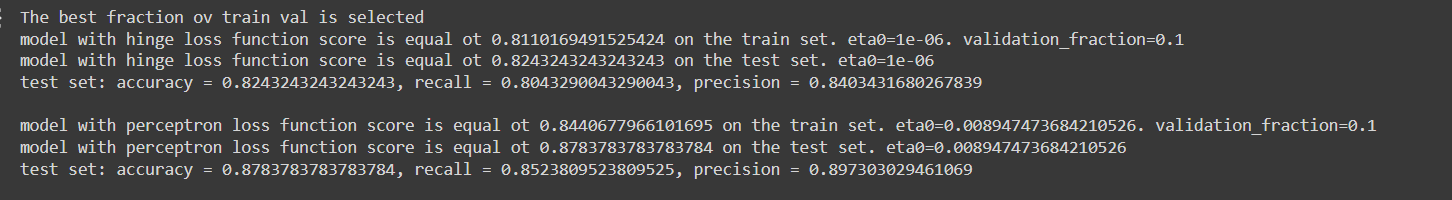


در ادامه بهترین مقدار فراپارامتر alpha را حساب می‌کنیم.



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

در انتها بهترین نسبت مجموعه داده صحت‌سنجی نسبت به آموزش را بدست می‌آوریم.



همانطور که از مقایسه قابل تشخیص 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,

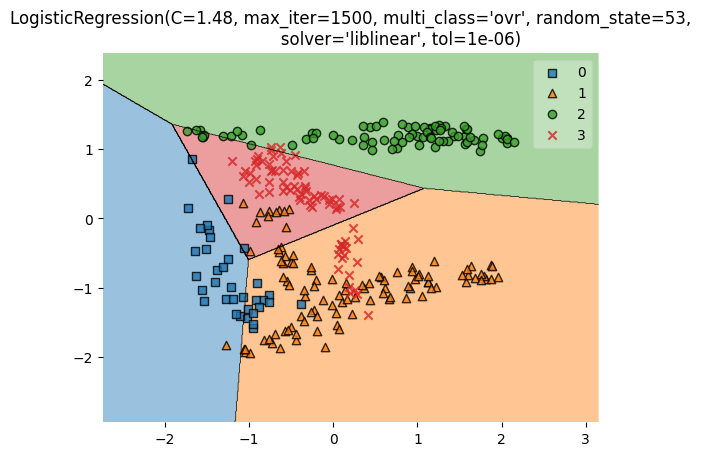
                      early\_stopping=True,

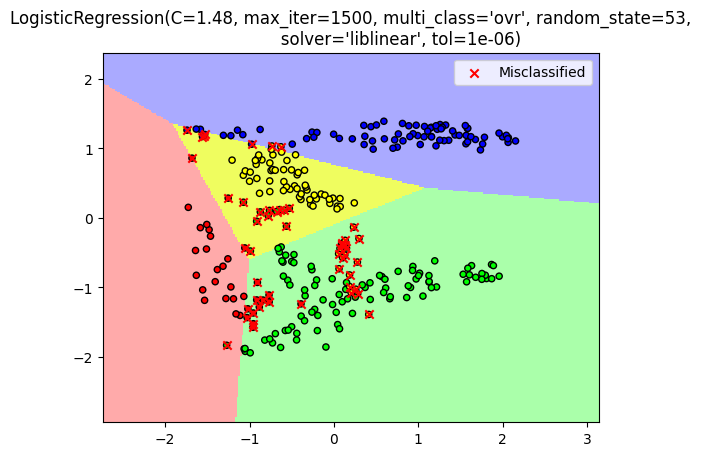
                      validation\_fraction=0.1,

                      n\_iter\_no\_change=10,

                      random\_state=53)

حال به رسم نمودارهای خواسته شده در صورت سوال، برای مدل LogisticRegression می‌پردازیم.





حال نتایج را برای مدل SGDClassifier بررسی می‌کنیم.

