بسم الله الرحمن الرحیم

تمرین سری 6 یادگیری ماشین

دکتر جمال الدین گلستانی

امیرحسین رستمی 96101635

دانشکده مهندسی برق

توجه: نمودار های ضمیمه شده برای یک ران دلخواه از کد های داخل ژوپیتر است و اگر مجدد ران کنید به علت shuffling ای که در دیتاست ها انجام می شود و نیز انتخاب رندوم خود فرآیند های یادگیری،نمودار ها یحتمل تغییرات کمی با نمودارهای فعلی داشته باشند و نیز اعداد ارقام خطا نیز اندکی متفاوت باشد ولی الگوی نمودار ها و اعداد یکسان خواهد بود.

سوال اول:

الف - عمق و تعداد گره در هر لایه را برابر $T = 8$ و $n = 8$ قرار دهید. آنگاه الگوریتم یادگیری را پنج بار، هر بار با تعداد گام $i = 10, 100, 200, 300, 400$ اجرا نمایید. پس از هر بار اجرا، خطای حاصله برای داده‌های آموزشی، همچنین خطای حاصله برای داده‌های تست را به دست آورید. این دو نوع خطا را به ترتیب L_T, L_S می نامیم. L_T, L_S را بر حسب i ترسیم نمایید. هر یک از دو خطا در چه مقدار i مینیمم می‌گردد. برداشت خود را از منحنی تغییرات L_T, L_S و مقایسه آنها را بیان کنید.

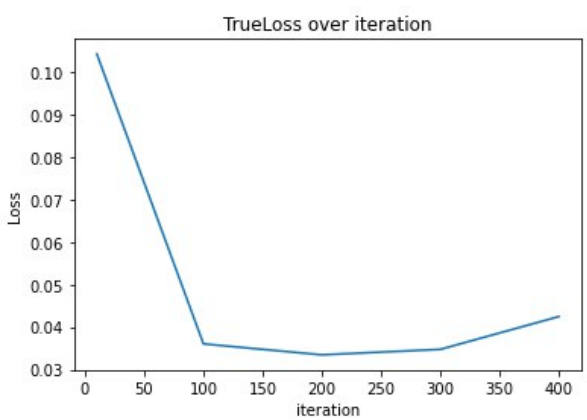
پاسخ الف

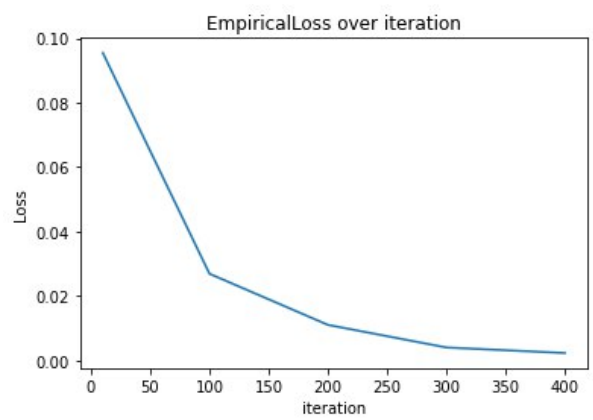
بررسی نمودار خطایِ داده آموزشی و خطایِ واقعی برحسب iteration:

تاثیر iteration از لحاظ تعوری به دو شکل است:

1- اولا تا یه حدی زیاد شدن iteration باعث می شود که مدل خوب برفضای مساله adapt شود و تا آنجا خطای true loss افت می کند.(توجه کنید که خطای empirical همواره در حال کاهش است).
2- دوما از یه حدی بیشتر بودن iteration باعث می شود که مدل روی داده های آموزشی over fit کند و لذا خواهیم داشت که نمودار خطای حقیقی بعد از یه حدی از افزایش iteration داریم که به علت over fit شدن داریم که خطای حقیقی شروع به افزایش می کند.

حال برویم سراغ نمودار های این بخش و داریم که نمودار خطا ها به صورت زیر می گردد:





نمودار های فوق انتظارات تعوری ذکر شده را تایید می کند.

داریم که خطای empirical با افزایش iteration در حال کاهش است و کمترین خطایش در این حین در "۴۰۰ = تعداد گام" است ولی خطای حقیقی داریم که در "۲۰۰ = تعداد گام" به حداقل می رسد.

نکته 1 : بنده پس از صحبت با تی ای مربوطه(ایمیل ذکر شده در صورت تمرین جهت پرسش سوالات) متوجه شدم که منظور از شبکه با n لایه در اصل شبکه با n-1 لایه نهان است.

نکته 2:می دانیم که خطای حقیقی قابل اندازه گیری نیست(جز موارد بسیار نادری و منظور بنده از خطای حقیقی در توضیحات ارایه شده همان خطا روی داده test است که به آن با نام true loss اشاره کردم.

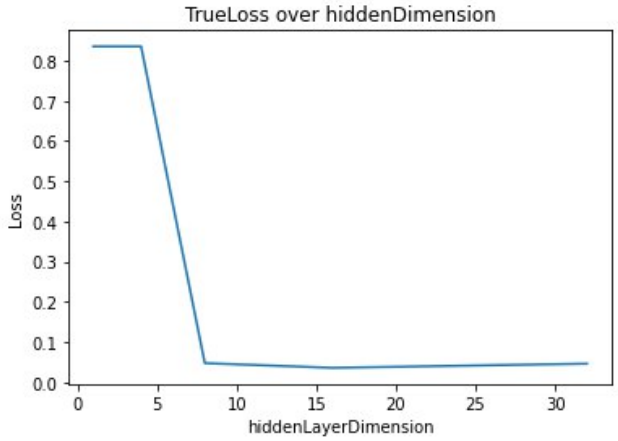
ب- تعداد گام های الگوریتم را برابر $i = 100$ و تعداد لایه‌ها را برابر $n = 8$ قرار دهید و این بار الگوریتم را هفت بار به ازای $n = 1, 2, 4, 8, 16, 32$ اجرا نمایید و بررسی‌های خواسته شده در بند الف را برای این حالت تکرار کنید .

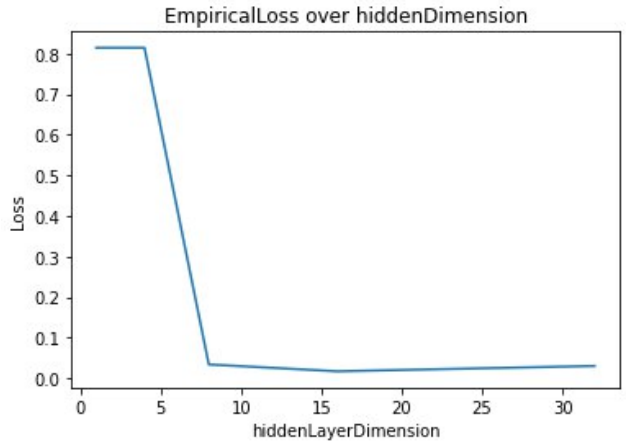
همانطور که می دانید لایه های نهان به نوعی پیچیدگی مدل مساله را در درون خود با طی کردن پروسه training حفظ می کنند و هر اندازه ابعاد لایه های نهان(تعداد گره ها در یک لایه) افزایش یابد داریم که پیچیدگی بیشتری را مدل یادگیری مان می تواند حفظ کند و لذا داریم که با افزایش ابعاد لایه های نهان،خطایِ ما افت می کند(چه خطای حقیقی و چه خطای empirical).البته این گفته ها در شرایطی است که تعداد گام های iteration به گونه ای نباشد که منتها به قرار گرفتن در حالت overfitting شود.

توجه کنید که توضیحاتی که دادیم قرار نیست دقیقا بی خطا به همین شکلی که گفتیم برقرار باشد چون محدودیت های موجود ممکن است باعث شود نمودار ها دقیقا به شکلی که گفتیم نباشد اما روند همین است.

حال بریم سراغ نمودار ها:

نمودارها خطا بر حسب hidden Dimension به شرح زیر است:





نمودار های فوق انتظارات تعوری ذکر شده را تایید می کند.

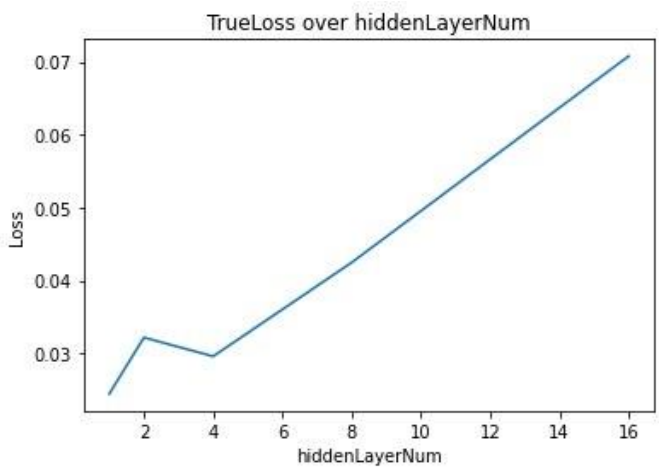
ج- تعداد گام های الگوریتم را برابر $i = 100$ و تعداد گره در هر لایه را برابر $n = 8$ قرار دهید. الگوریتم را شش بار به ازای $T = 1, 2, 4, 8, 16$ اجرا نمایید و بررسی های خواسته شده در بند الف را برای این حالت تکرار کنید.

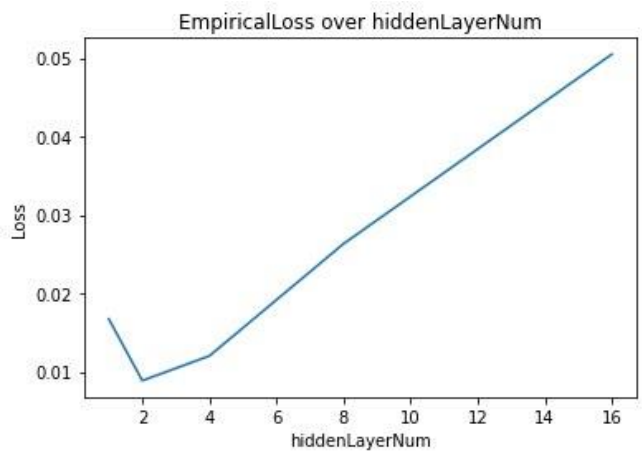
همانطور که می دانید تعداد لایه های نهان همانند ابعاد لایه های نهان به نوعی پیچیدگی مدل مساله را در درون خود با طی کردن پروسه training حفظ می کنند.اما افزایش تعداد لایه ها باعث می شود با سرعت بالایی اندازه فضای H ما زیاد شود و این باعث می شودکه زود در معرض overfitting قرار بگیریم،البته توجه کنید تا یه حدی افزایش hidden layer به مدل سازی کردن پیچیدگی مساله ما کمک می کند اما به علت رشد شدید |H| این افزایش تعداد لایه های نهان زود منتها به over fit شدن می گردد.

توجه کنید که توضیحاتی که دادیم قرار نیست دقیقا بی خطا به همین شکلی که گفتیم برقرار باشد چون محدودیت های موجود ممکن است باعث شود نمودار ها دقیقا به شکلی که گفتیم نباشد اما روند همین است.

حال بریم سراغ نمودار ها:

نمودارها خطا بر حسب تعداد لایه های نهان به شرح زیر است:





نمودار های فوق انتظارات تعوری ذکر شده را تایید می کند.

سوال دوم:

ابتدا داده های دیتاست را shuffle می کنیم و سپس نیمه اول از داده های دیتاست را به عنوان دیتای train و نیمه باقی را به عنوان دیتای test انتخاب می کنیم و در training و testing ِ مدل یادگیری،از آن ها استفاده می کنیم.

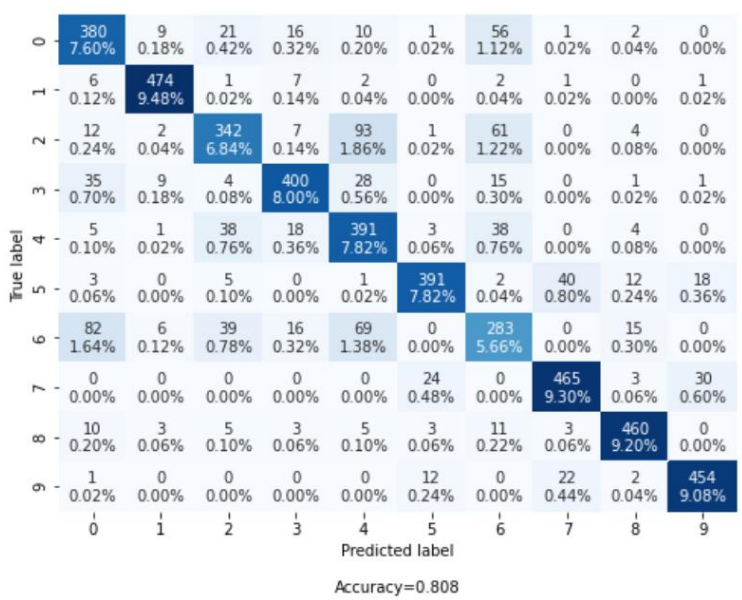
- روش اول(استفاده از SVM با کرنل خطی):

ابتدا به کمک ویژوالایز کردن خروجی classification_report و confusion_matrix به بررسی وضعیت عملکردی این روش می پردازیم.

خروجی classification report:

	precision	recall	f1-score	support
0	0.71	0.77	0.74	496
1	0.94	0.96	0.95	494
2	0.75	0.66	0.70	522
3	0.86	0.81	0.83	493
4	0.65	0.79	0.71	498
5	0.90	0.83	0.86	472
6	0.60	0.55	0.58	510
7	0.87	0.89	0.88	522
8	0.91	0.91	0.91	503
9	0.90	0.92	0.91	491
accuracy			0.81	5001
macro avg	0.81	0.81	0.81	5001
weighted avg	0.81	0.81	0.81	5001

خروجی confusion matrix:



همانطور که مشاهده می کنید این روش با دقت تقریبی ۸۱ درصد،دسته بندی داده های test را به درستی انجام می دهد.

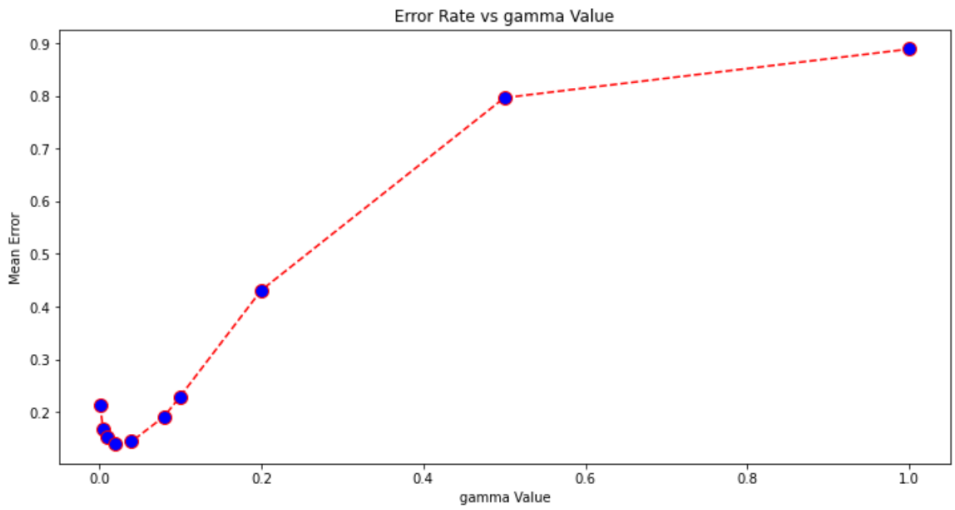
- روش دوم(استفاده از SVM با کرنل گاوسی):

همانطور که می دانید در این روش یک پارامتر آزادی داریم و آن پارامتر گاما است که به نوعی تعیین گر حاشیه امن ابرصفحه یادگرفته شده است.

برای تعیین اینکه کدام گاما برای این دیتاست بهترین مدل SVM گاوسی را مدل می کند لازم است تا روی حالات مختلف گاما جاروب بزنیم.

پس از جاروب روی مقادیر مختلف گاما به نمودار زیر که بیانگر خطا بر حسب مقدار گاما است رسیدیم:

بهترین گاما برای این دیتاست به صورت تقریبا برابر ۰/۲ است.



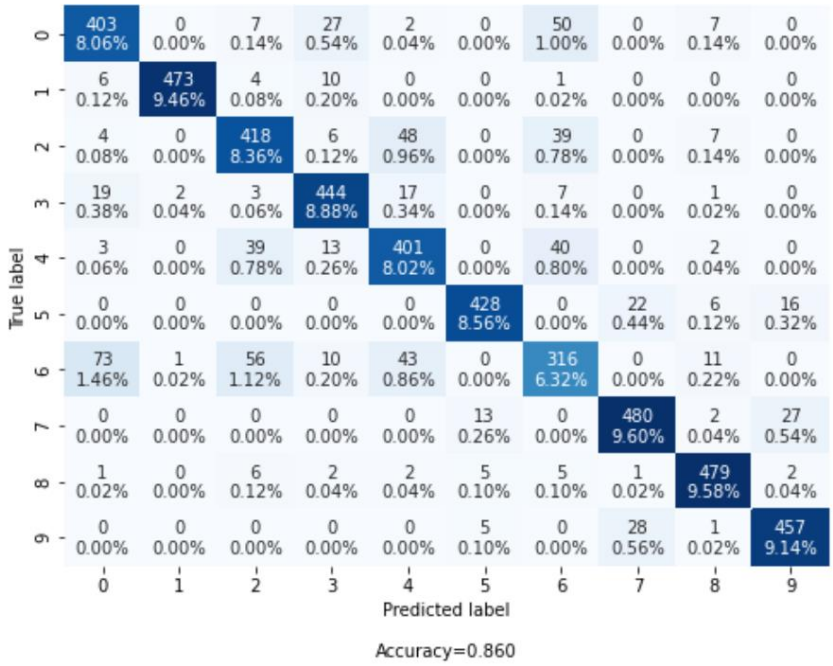
حال برای گاما برابر ۰/۲ نتایج classification report و confusion matrix را رسم می کنیم:

همانطور که مشاهده می کنید از لحاظ عملکردی SVM گاوسی بهتر از SVM خطی عمل می کند.

	precision	recall	f1-score	support
0	0.79	0.81	0.80	496
1	0.99	0.96	0.98	494
2	0.78	0.80	0.79	522
3	0.87	0.90	0.88	493
4	0.78	0.81	0.79	498
5	0.95	0.91	0.93	472
6	0.69	0.62	0.65	510
7	0.90	0.92	0.91	522
8	0.93	0.95	0.94	503
9	0.91	0.93	0.92	491
accuracy			0.86	5001
macro avg	0.86	0.86	0.86	5001
weighted avg	0.86	0.86	0.86	5001

در ادامه به بررسی ماتریس confusion در این حالت می پردازیم.

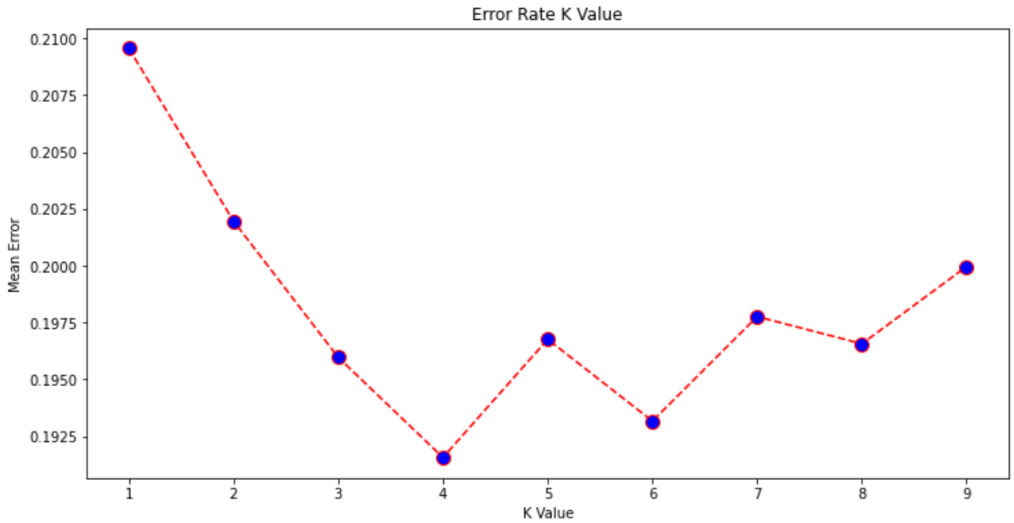
ویژوالایز شده ماتریس confusion:



همانطور که مشاهده می کنید این روش با دقت تقریبی ۸۶ درصد،دسته بندی داده های test را به درستی انجام می دهد.(بهتر شدن کیفیت مدل یادگیری نسبت به حالت قبل).

- روش سوم(استفاده از KNN):

در این روش "مترِ" استفاده شده همان "فاصله اقلدیسی" است.همانطور که میدانید مقدار k تعیین نشده است و لذا لازم است که با جاروب روی مقادیر مختلف k مقدار بهینه برای مدل سازی با این دیتاست را بیابیم.
نمودار جاروب به شرح زیر گردید:

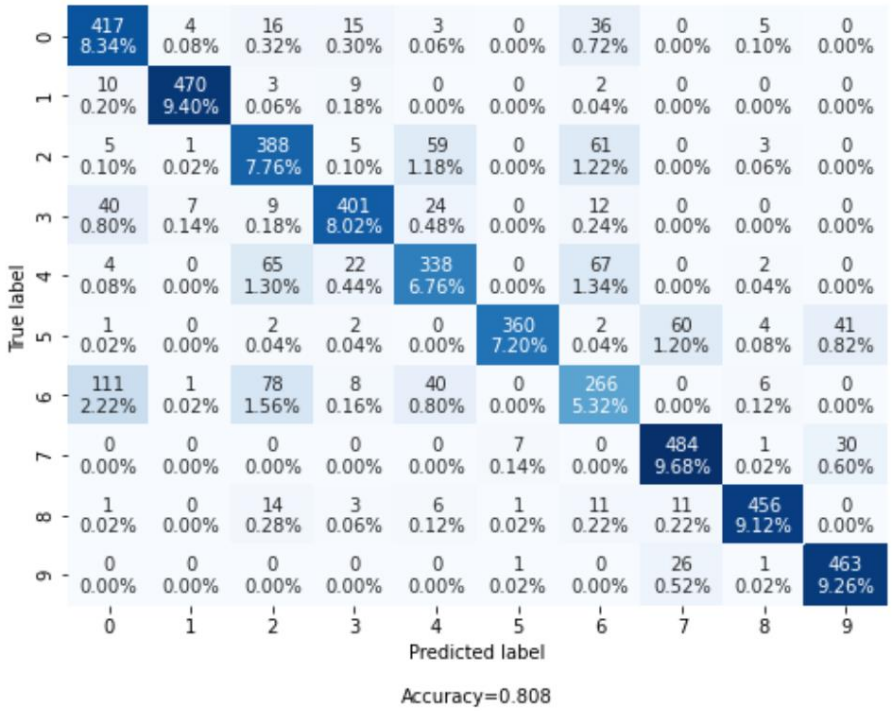


همانطور که مشاهده می کنید به ازای k برابر 4 مقدار loss مینیمم گردید و حال مقدار k را برابر 4 تنظیم کرده و ماتریس confusion و classification report را به دست می آوریم.

مقادیر مرتبط با classification report:

	precision	recall	f1-score	support
0	0.71	0.84	0.77	496
1	0.97	0.95	0.96	494
2	0.67	0.74	0.71	522
3	0.86	0.81	0.84	493
4	0.72	0.68	0.70	498
5	0.98	0.76	0.86	472
6	0.58	0.52	0.55	510
7	0.83	0.93	0.88	522
8	0.95	0.91	0.93	503
9	0.87	0.94	0.90	491
accuracy			0.81	5001
macro avg	0.81	0.81	0.81	5001
weighted avg	0.81	0.81	0.81	5001

ویژولایز شده ماتریس confusion:



همانطور که مشاهده می کنید این روش با دقت تقریبی ۸۱ درصد،دسته بندی داده های test را به درستی انجام می دهد.(از لحاظ عملکردی در برآیند توانمندی اش در حد SVM خطی است).

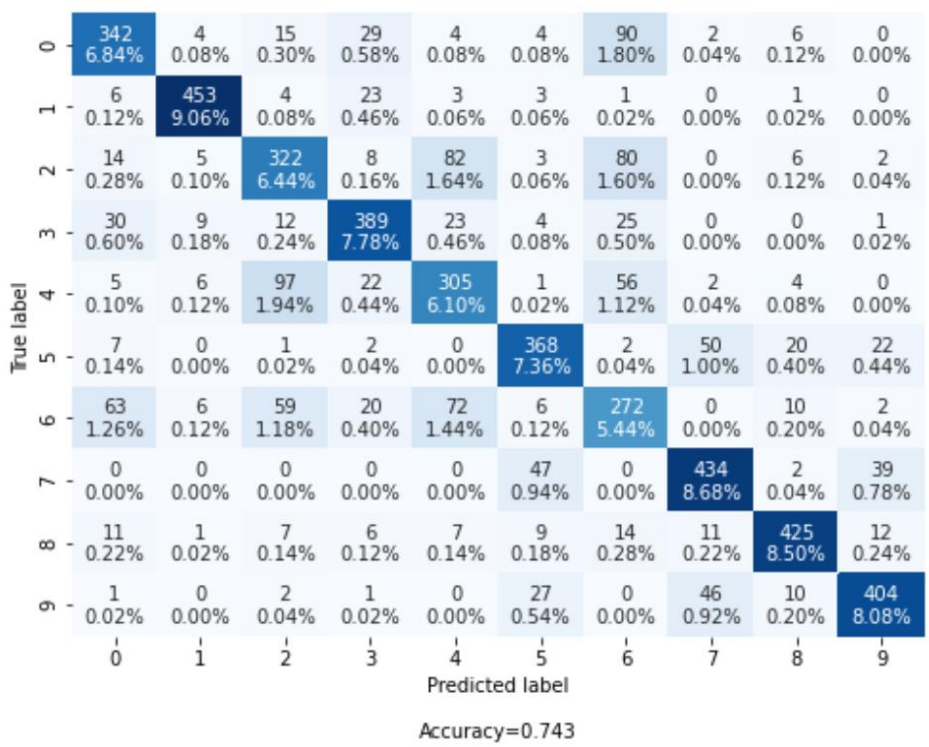
- روش چهارم(استفاده از Decision Tree):

طبق توضیحات صورت سوال محدودیتی روی عمق درخت و تعداد نود های درخت نمی گذاریم و از پارامتر های پیشفرض جهت مدل سازی استفاده می کنیم.

خروجی classification report:

	precision	recall	f1-score	support
0	0.71	0.69	0.70	496
1	0.94	0.92	0.93	494
2	0.62	0.62	0.62	522
3	0.78	0.79	0.78	493
4	0.61	0.61	0.61	498
5	0.78	0.78	0.78	472
6	0.50	0.53	0.52	510
7	0.80	0.83	0.81	522
8	0.88	0.84	0.86	503
9	0.84	0.82	0.83	491
accuracy			0.74	5001
macro avg	0.75	0.74	0.74	5001
weighted avg	0.74	0.74	0.74	5001

خروجی ماتریس confusion:



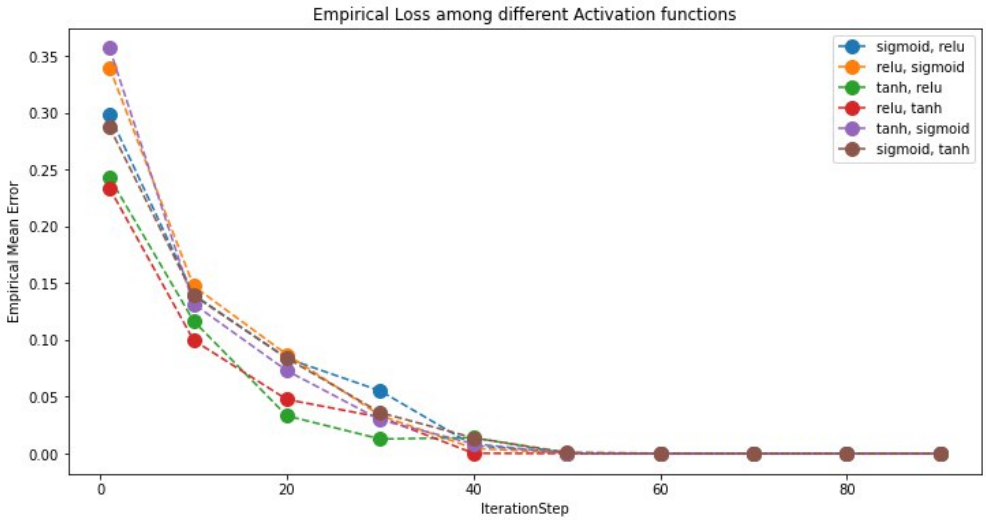
همانطور که مشاهده می کنید این روش با دقت تقریبی ۷۴/۳ درصد،دسته بندی داده های test را به درستی انجام می دهد.(از لحاظ عملکردی در برآیند توانمندی اش از باقی مدل ها ضعیف تر است).

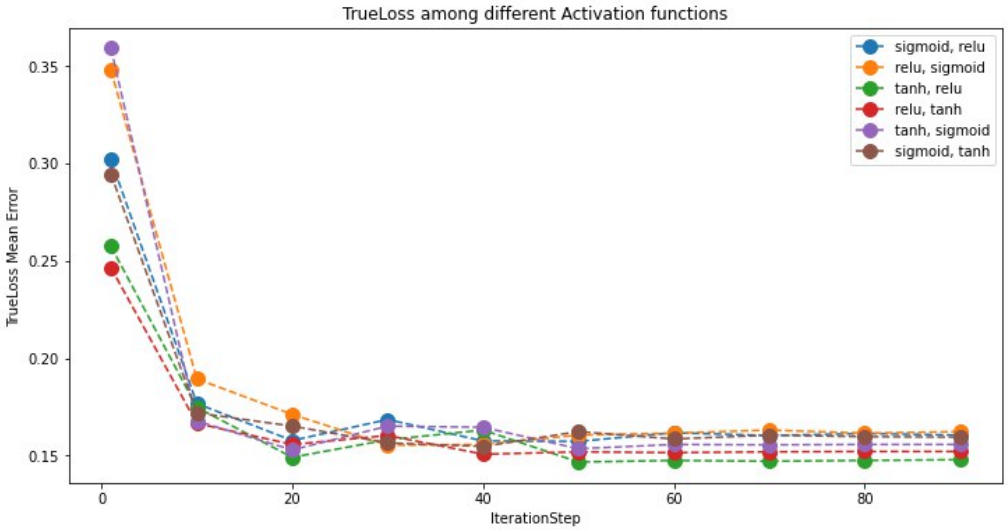
- روش پنجم(استفاده از شبکه عصبی):

تنها پارامتری که در انتخاب آن آزادی داریم تابع فعال سازی لایه های میانی است.

همانطور که می دانید تعداد توابع activation زیاد بود و در این بررسی ما نظر خود را معطوف سه تابع معروف sigmoid,tanh,relu و حالت خطی می کنیم و انواع حالت های ممکن با این 4 تابع را بررسی می کنیم.

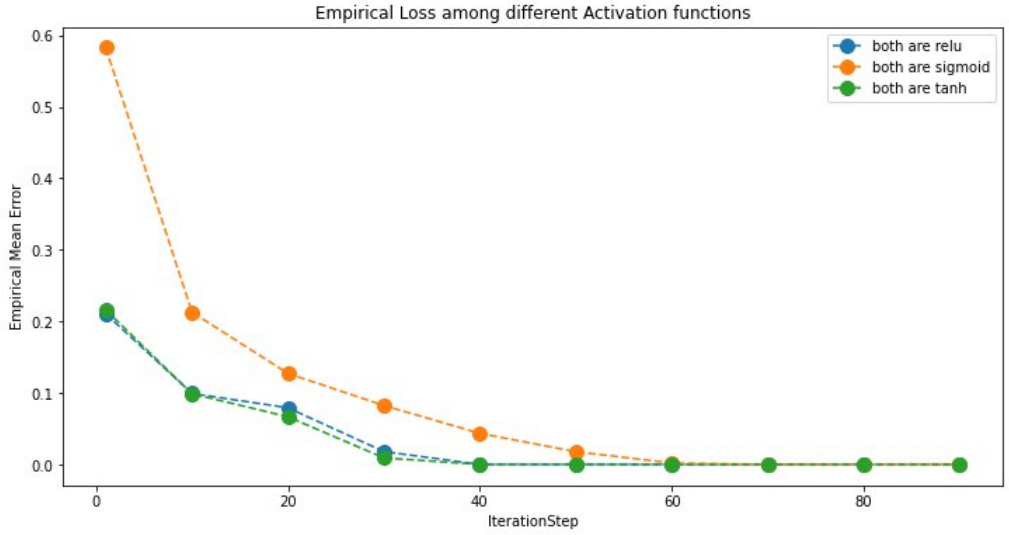
حالت اول(توابع فعال سازی لایه اول و دوم متفاوت باشند و هیچ کدام خطی نباشند):

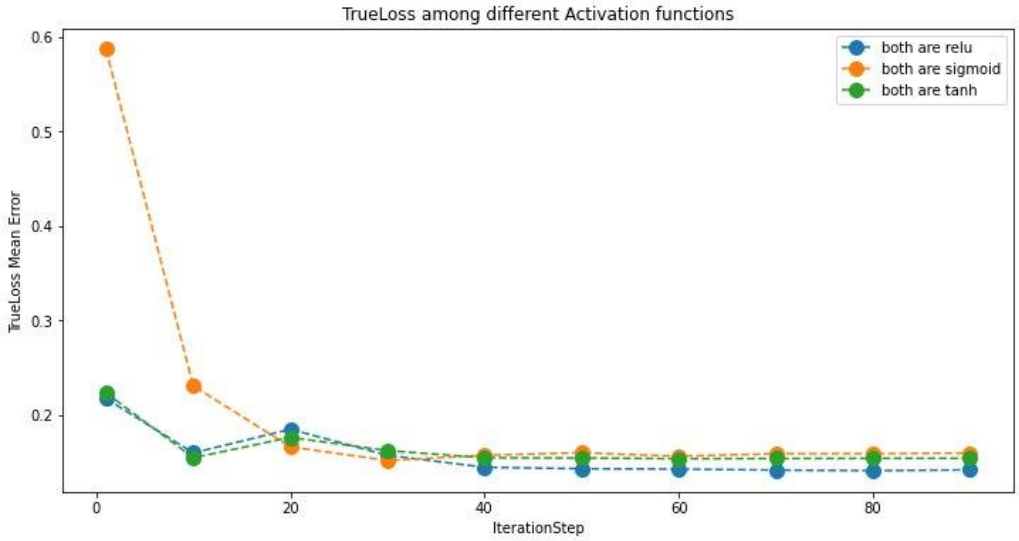




جهت مقایسه روش ها بهتر است موردی را انتخاب کنیم که True Loss کمتری داشته باشد(و خطای داده های آموزشی آن هم معقول باشد) و با بررسی نمودار های فوق داریم که حالتی که یکی از توابع relu باشد و دیگری tanh باشد از لحاظ خطای True Loss و هم چنین Empirical loss در وضعیت بهتری قرار دارد.

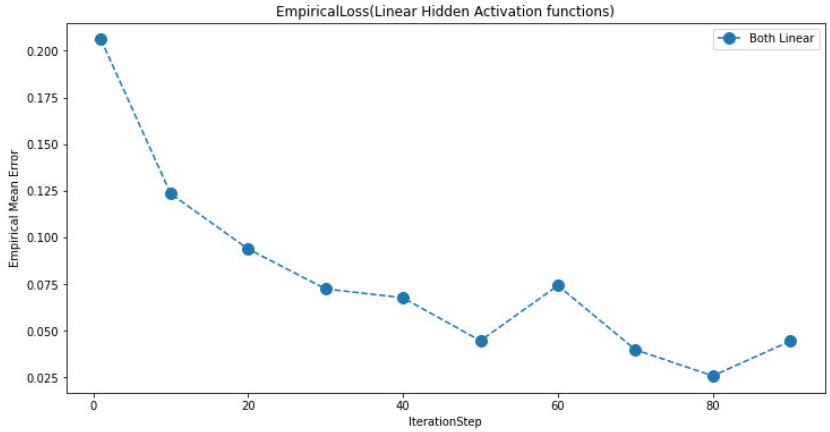
حالت دوم:(توابع فعال سازی لایه اول و دوم یکسان باشند و هیچ کدام خطی نباشند):

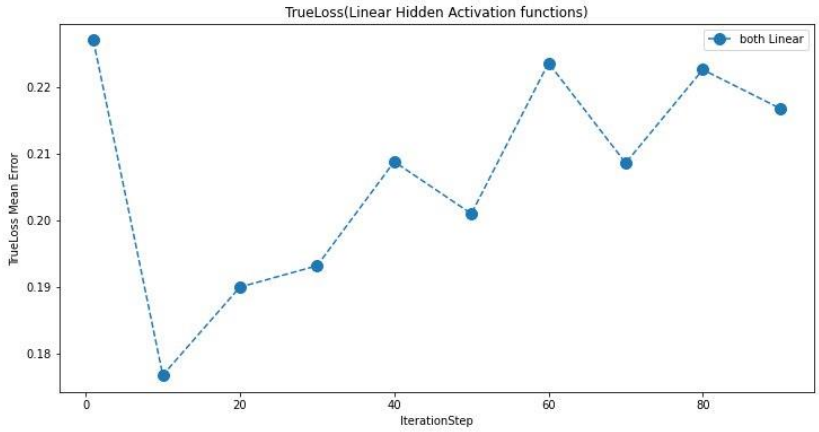




جهت مقایسه روش ها بهتر است موردی را انتخاب کنیم که True Loss کمتری داشته باشد(و خطای داده های آموزشی آن هم معقول باشد) و با بررسی نمودار های فوق داریم که حالتی که جفت توابع relu یا tanh باشند از لحاظ خطای True Loss و هم چنین Empirical loss در وضعیت بهتری قرار دارد.

حالت سوم:(جفت توابع فعال سازی linear باشد):





همانطور که مستحضر هستید هنگامی که همه activation function ها از نوع linear باشد داریم که انگار داریم مساله را با تابع خطی شبیه سازی می کنیم یعنی در اصل label ها به صورت ترکیب خطی ای از فیچر ها خواهند بود و اگر مساله چنین باشد و بتواند مدل سازی خطی برای مساله درنظرگرفت داریم که خطا بسیار کم خواهد بود ولی اگر مساله خطی نباشد(تقریبا هم خطی نباشد) رفتار مدل خطی برایش بسیار نامناسب بوده و همانطور که نمودار های فوق را مشاهده می کنید با توجه به نمودار True loss می تواند این برداشت رو کرد که مساله خطی نیست زیرا خطای True loss به صورت ناهمگون بالا پایین می شود و این یعنی مدل سازی خطی نمی تواند پیچیدگی های موجود در مساله را درخود هضم کند.

جمع بندی:(از لحاظ عملکردی)

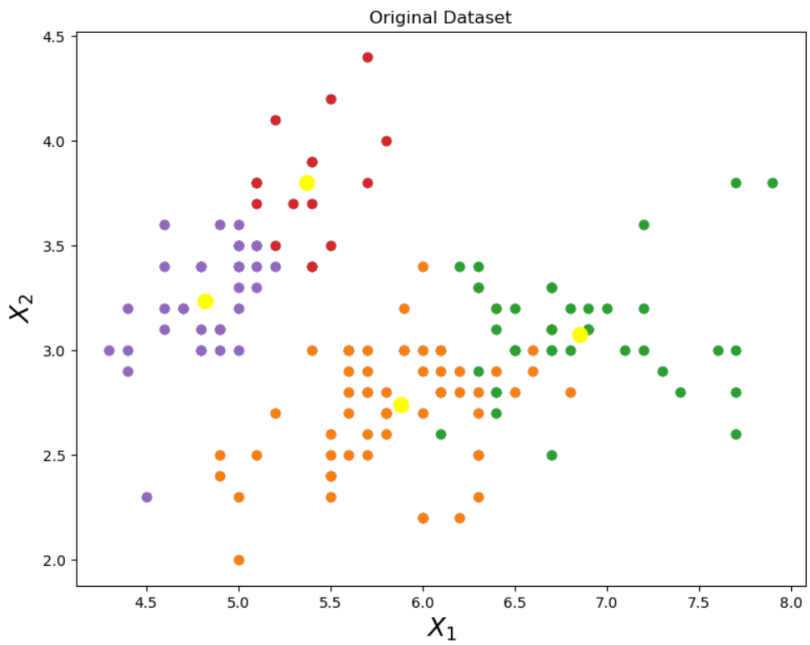
اگر بخواهیم خیلی ریز خطاهارو زیرذره بین ببریم داریم که خطای حالتی که جفت activation ها یکی باشند و جفت relu یا جفت tanh باشند داریم که کمترین True loss را خواهد داشت و حالتی که جفت خطی باشند بیشترین خطارو خواهد داشت.

سوال سوم:

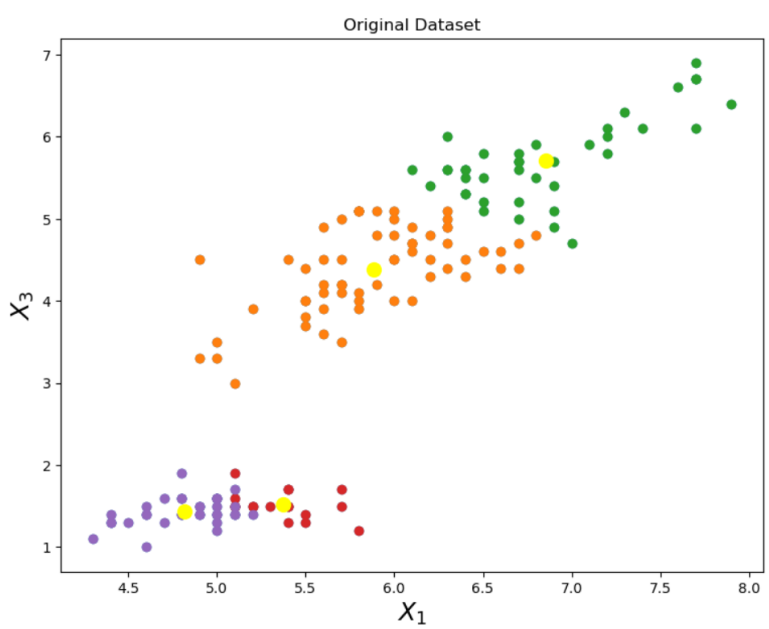
الف)پیاده سازی خواسته شده با جزییات تمام در کد ژوپیتر ضمیمه شده انجام شده است.

ب)ابتدا خوشه بندی را به کمک همه چهار فیچر انجام می دهیم و سپس نمودار های خوشه هارا در هر 6 حالت رسم می کنیم:(مراکز خوشه ها با توپ های زرد مشخص شده است و اعضای هر دسته رنگ مخصوص خود را دارند).

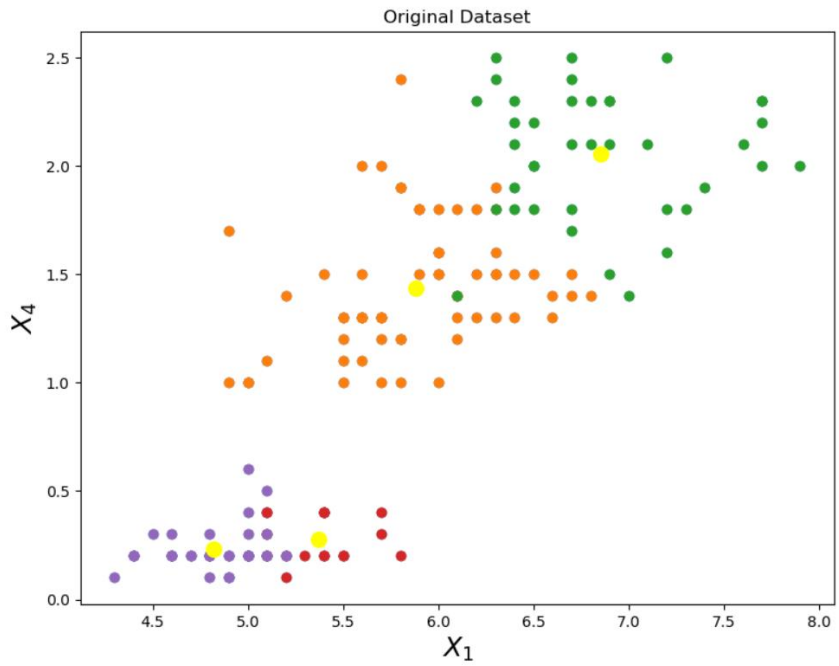
1. نمودار $X_2 - X_1$:



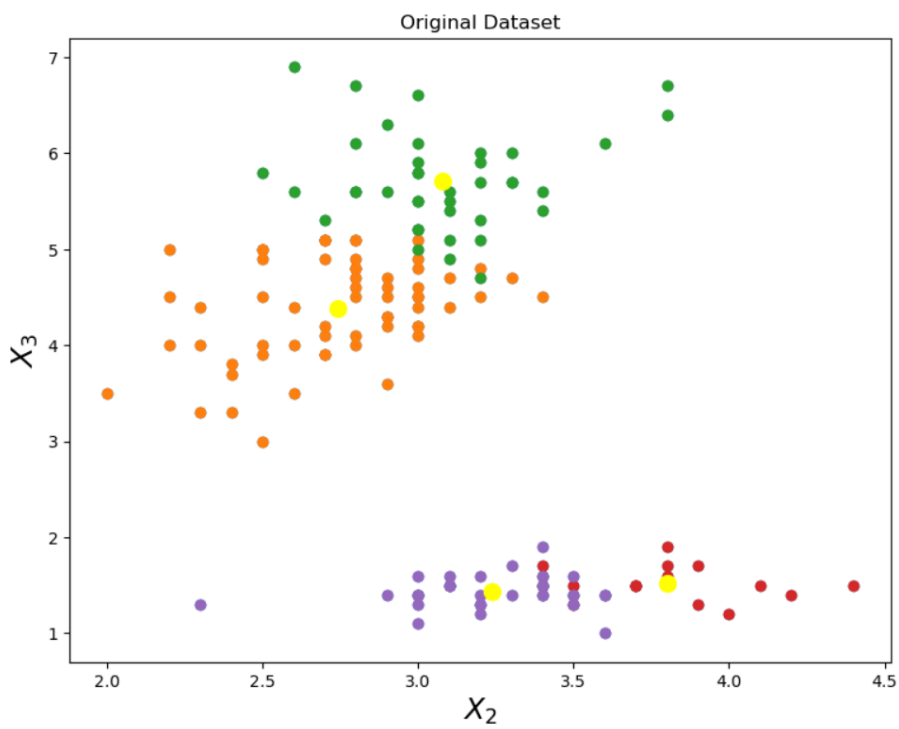
2. نمودار $X_3 - X_1$:



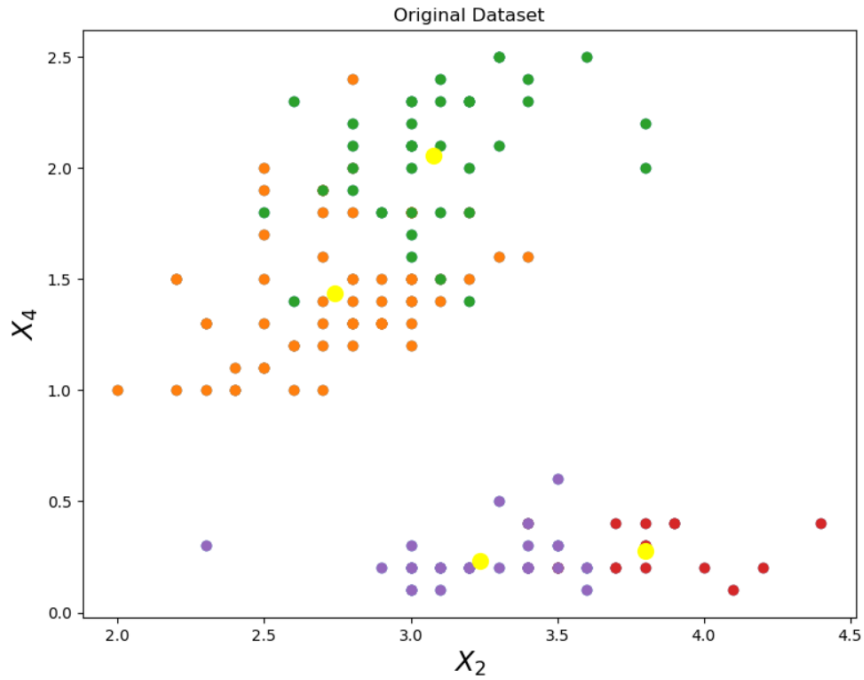
3. نمودار $X_4 - X_1$



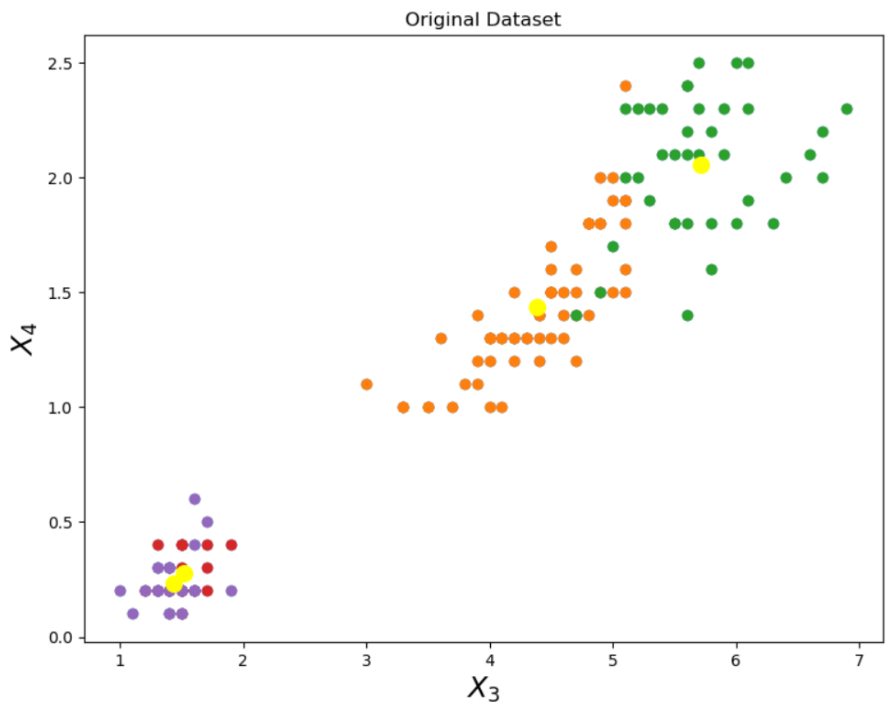
4. نمودار $X_3 - X_2$



5. نمودار $X_4 - X_2$



6. نمودار $X_4 - X_3$

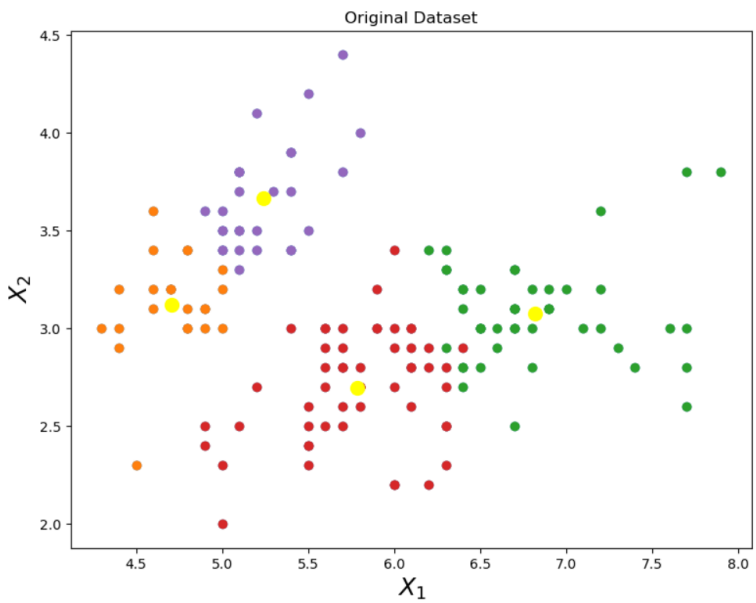


ج)بله می تواند از یکی این ویژگی هارا در تعیین خوشه بندی صرف نظر کرد،برای نشان دادن این برقراری،نمودار های خوشه بندی را در فقدان ویژگی صرفنظر شده رسم می کنیم و مشاهده می کنیم که در تعیین خوشه بندی اثرکمی داشته و کماکان خوشه بندی ها به شکل قسمت قبل(با تغییرناچیز) برقرار اند.

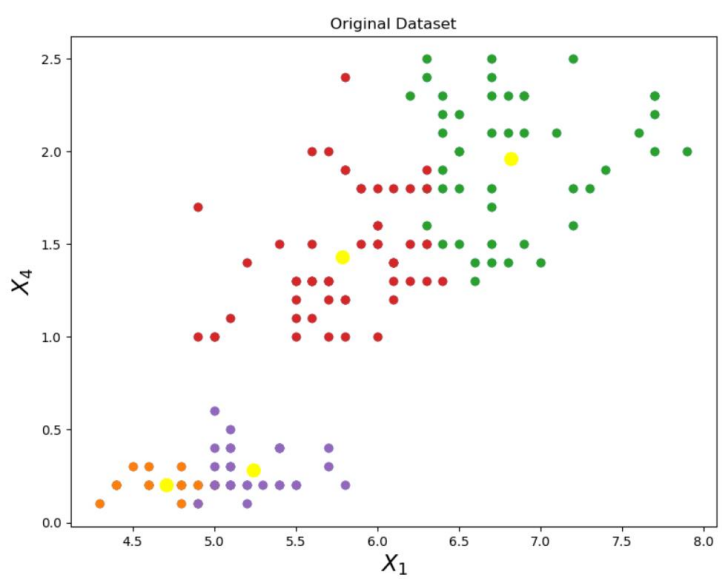
ویژگی درنظرگرفته نشده در فرآیند K means : X_3

حال خوشه بندی را به کمک سه فیچر باقی مانده انجام می دهیم و سپس نمودار های خوشه هارا در هر 3 حالت رسم می کنیم:(مراکز خوشه ها با توپ های زرد مشخص شده است و اعضای هر دسته رنگ مخصوص خود را دارند).

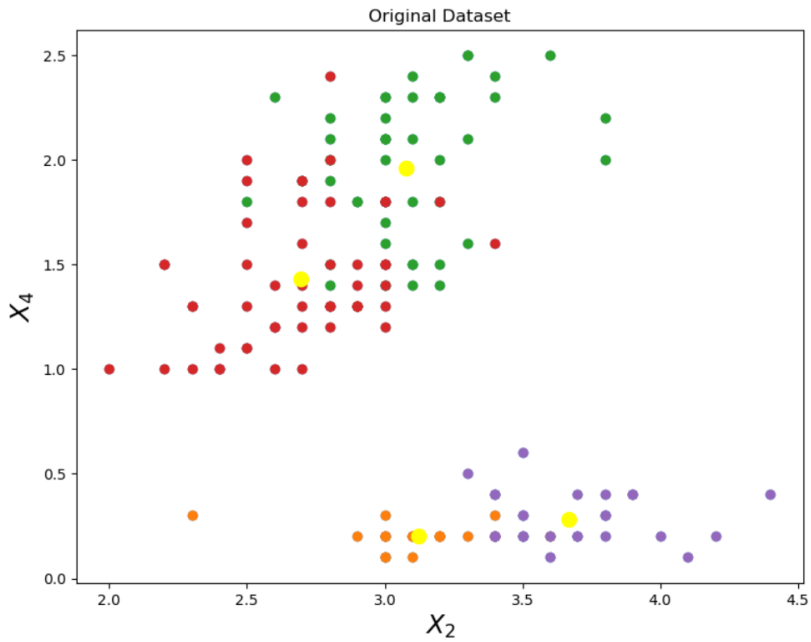
1. نمودار $X_2 - X_1$:



2. نمودار $X_4 - X_1$:



3. نمودار $X_4 - X_2$:



دلایل این رخداد:

همانطور که می دانید این چهار ویژگی عبارت اند از:

1- **طولِ کاسبرگ**
2- **عرضِ کاسبرگ**
3- **طولِ گلبرگ**
4- **عرضِ گلبرگ**

4 پارامتر فوق دارای استقلال کامل نیستند و یک گیاه نمی تواند هر طول و عرض دلخواهی را داشته باشد و تقریبا با مشخص شدن 3 پارامتر از 4 پارامتر فوق به نوعی حدودی برای پارامتر آخر تعیین می شود(مگر برخی گونه های نادر) و به عبارت بهتر فضای گل های با این 4 فیچر دارای بعدی کمتر از 4 است و به همین دلیل اگر یکی از این ویژگی هارا در فرآیند K means درنظر نگیریم آنقدر خطا در clustering رخ نمی دهد(جز برخی گونه های نادر) و نتایج به دست آمده هم صحه ای بر گفته بالا است.