

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



درس سیستم‌های هوشمند

تمرین شماره 4

نام و نام خانوادگی : شایان واصف احمدزاده

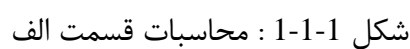
شماره دانشجویی : 810197603

مهر 1400

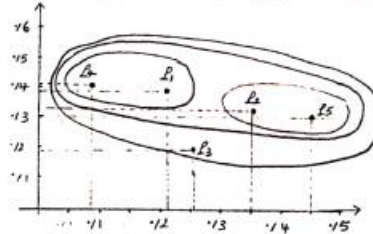
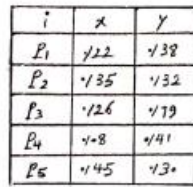
فهرست سوالات

- سوال 1: تحلیلی 3
- الف: خوشه بندی با روش کا-میانگین 3
- ب: خوشه بندی سلسله مراتبی 4
- سوال 2 : پیاده سازی الگوریتم خوشه بندی 6
- الف: تاثیر تعداد خوشه ها 6
- ب: تاثیر تکرار آزمایش 7
- سوال 3 : یادگیری نیمه نظارت شده (امتیازی) 11
- الف: رگرسیون لجستیک 11
- ب: ارزیابی طبقه بند 12
- ج/د : یادگیری نیمه نظارت شده / شرایط استفاده 16
- سوال 4 : مقدمات احتمال 22
- الف: سوالات تحلیلی 22
- ب: سوال شبیه سازی 24

(الف)



(پ)



محال توابع دونه / نقاط راجع به محاوره و فاصله (Distance matrix) را تشکیل می‌دهیم :

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	0.15	0			
P_3	0.20	0.75	0		
P_4	0.74	0.28	0.29	0	
P_5	0.25	0.77	0.22	0.39	0

سید، مثل P_2 و P_5 را، دید، است، مرا می‌دیم، و در کلاس اول به صورت
"بزرگداشت"



نقاط p_2 و p_5

سبحان ہذا زبانِ مانتہ فاضلہ - نیاز داریم بحاسبہ زیر را انجام دسیم :

$$\min [\text{dist}(P_2, P_5), P_1] = \min [\text{dist}(P_2, P_1), \text{dist}(P_5, P_1)] = \min [1.15, 1.25] = 1.15$$

$$\min[\text{dist}(P_2, P_5), P_3] = \min[\text{dist}(P_2, P_3), \text{dist}(P_5, P_3)] = \min[.175, .122] = .122$$

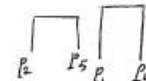
$$\min[\text{dist}(P_2, P_5), P_4] = \min[\text{dist}(P_2, P_4), \text{dist}(P_5, P_4)] = \min[1.28, 1.39] = 1.28$$

	P_1	$P_{2,5}$	P_3	P_4
P_1	0			
$P_{2,5}$	175	0		
P_3	120	175	0	
P_4	174	128	129	0

نقاط P_1, P_2

حل بائیس را برزسانی کنیم:

+ پس در شکل اولیه، $P_1 P_2$ را یک سیم قرار دهیم و دو دایره را برزسانی کنیم.



۳ بدای سرور، رسانی عاتقین، ماحول، نیاز داریم دو محاسبه زیر را انجام دهیم.

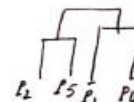
$$\min [\text{dist}(P_1, P_4) + (P_2, P_5)] = \min [\text{dist}(P_1, (P_2, P_5)), \text{dist}(P_4, (P_2, P_5))] = \min [1.75, 1.28] = 1.25$$

$$\min[\text{dist}(P_1, P_4), P_3] = \min[\text{dist}(P_1, P_3), \text{dist}(P_4, P_3)] = \min[(1, 2), (1, 2)] = 1, 2$$

	P_1, P_4	P_2, P_5	P_3
P_1, P_4	•		
P_2, P_5	(-175)	•	
P_3	-12	-175	•

استغفر الله من ذنوبي
و من ذنوب اخوتي

حاله فائزین را بنور ربانی و کسب و صفا ندرده مشفق است و در مامله عظیم دایم که اوست انعام عظیم است
و در کسب هم سعادت نیز سوزش را بشود و در نگار اولیه به روش ایجاد شده را یک بسته قرار می‌دهیم



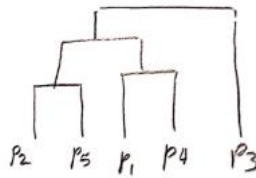
شکل 1-2-1: محاسبات قسمت ب

در این مرحله، نیاز داریم محاسبه کنیم که آیا باید این دو گروه را با هم ادغام کنیم یا نه.

$$\min[\text{dist}((P_1, P_4), (P_2, P_5)), P_3] = \min[\text{dist}((P_1, P_4), P_3), \text{dist}((P_2, P_5), P_3)] = \min(1.12, 1.75) = 1.12$$

	P_1, P_4	P_3
P_1, P_4	0	
P_3	1.12	0

در این مرحله، نیاز داریم محاسبه کنیم که آیا باید این دو گروه را با هم ادغام کنیم یا نه.



در این مرحله، نیاز داریم محاسبه کنیم که آیا باید این دو گروه را با هم ادغام کنیم یا نه.

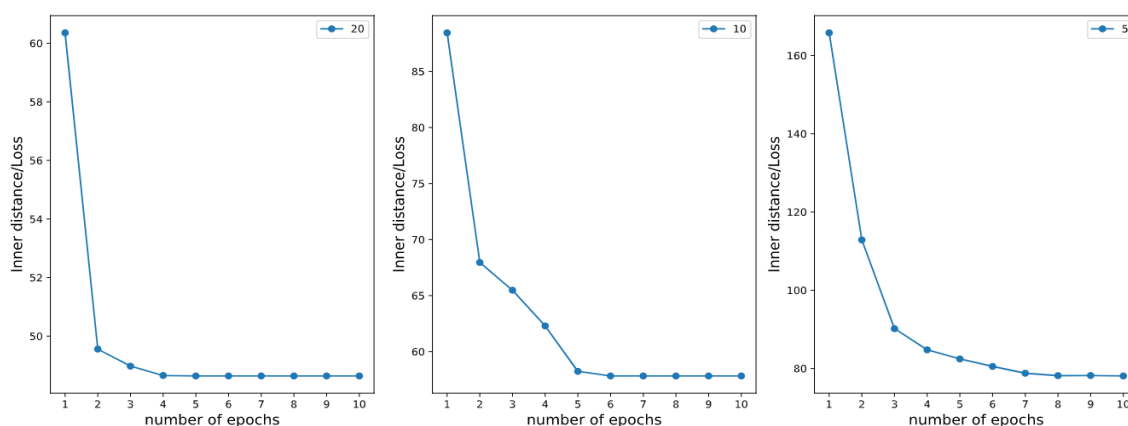
شکل 1-2-2: ادامه محاسبات قسمت ب

سوال 2 :

الف)

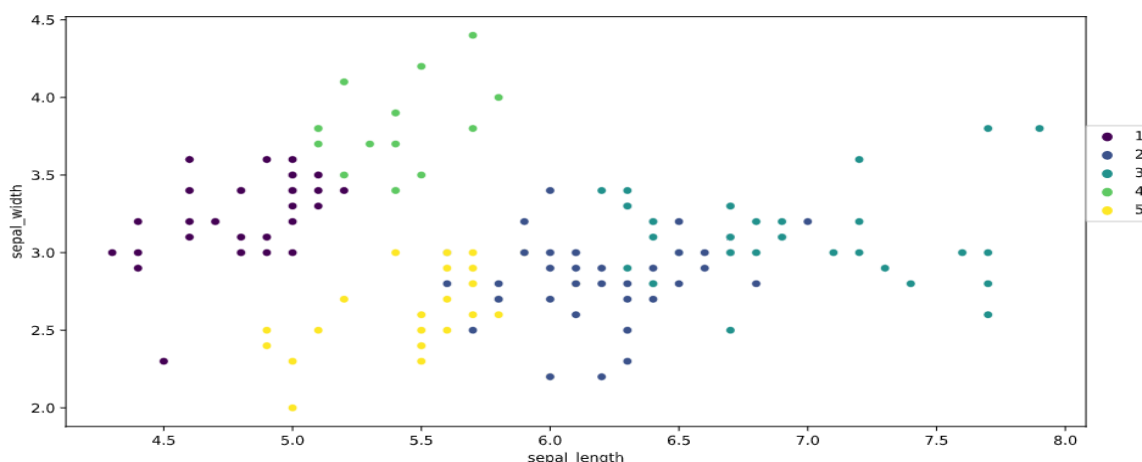
ابتدا طبق خواسته سوال ، الگوریتم k-means را برای تعداد 5,10,20 خوشه تکرار می کنیم . همچنین اگر به جای اینکه نقاط ابتدایی که مراکز اولیه خوشه های (Clusters) ما هستند را کاملاً رندوم انتخاب کنیم از بین sample های موجود انتخاب کنیم (برای مثال از بین 150 داده موجود ، k داده را به صورت تصادفی انتخاب کنیم و مراکز خوشه اولیه قرار دهیم) ، الگوریتم مورد نظر در تعداد Iteration بسیار کمتری همگرا می شود و از نظر زمانی بهینه تر می باشد .

همچنین فاصله درون کلاسی (Inner distance) که فاصله دادگان درون یک خوشه از میانگین آن خوشه می باشد را به عنوان خطا (Loss) در نظر می گیریم که قصد داریم آنرا کمینه کنیم . در زیر مقادیر خطا را بر حسب تعداد epoch ها تا همگرایی الگوریتم برای سه مقدار متفاوت cluster آورده ایم :



شکل 1-1-2 : مقدار Loss بدست آمده بر حسب Iteration

در ادامه نحوه partitioning برای k=5 برای دو تا از ویژگی ها آورده شده است :



شکل 2-1-2 : نحوه partitioning برای تعداد 5 cluster

طبق شکل 1-1-2 ، طبیعتاً انتظار داریم با افزایش تعداد cluster ، فاصله درونی (Loss) کاهش پیدا کند که این اتفاق افتاد ولی برای بدست آوردن تعداد بهینه cluster نیاز به معیار بهتری داریم که در بخش بعد بررسی می‌کنیم .

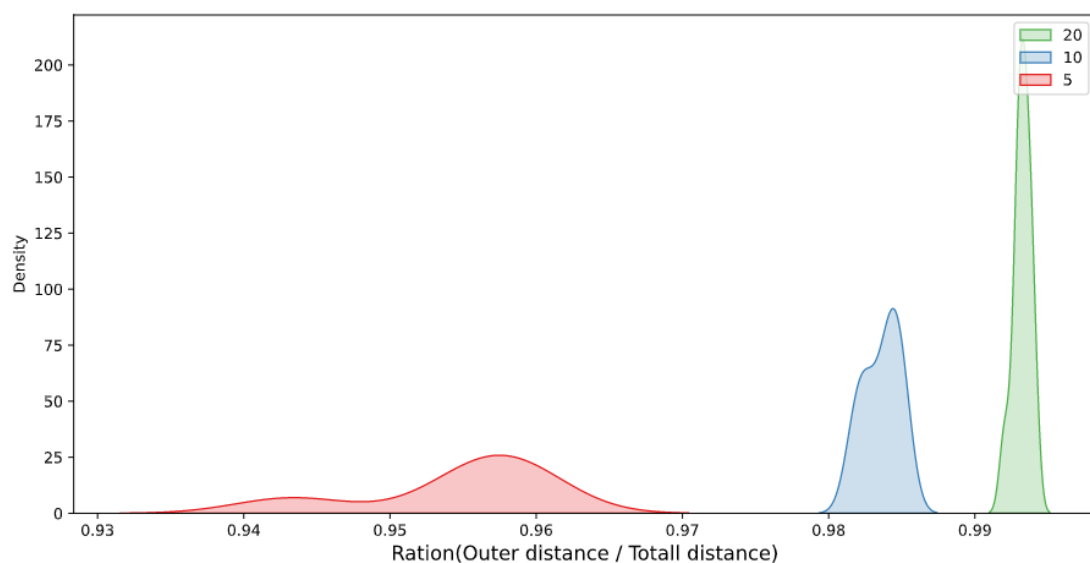
(ب)

در این قسمت برای هر کدام از تعداد خوشه مطرح شده در سوال ، الگوریتم را به تعداد 10 بار اجرا کرده و 4 نمودار Outer distance , Ratio, Inner distance و kde را رسم می‌کنیم :

Ratio را به صورت زیر تعریف می‌کنیم:

$$Ratio = \frac{Outer_{distance}}{Outer_{distance} + Inner_{distance}}$$

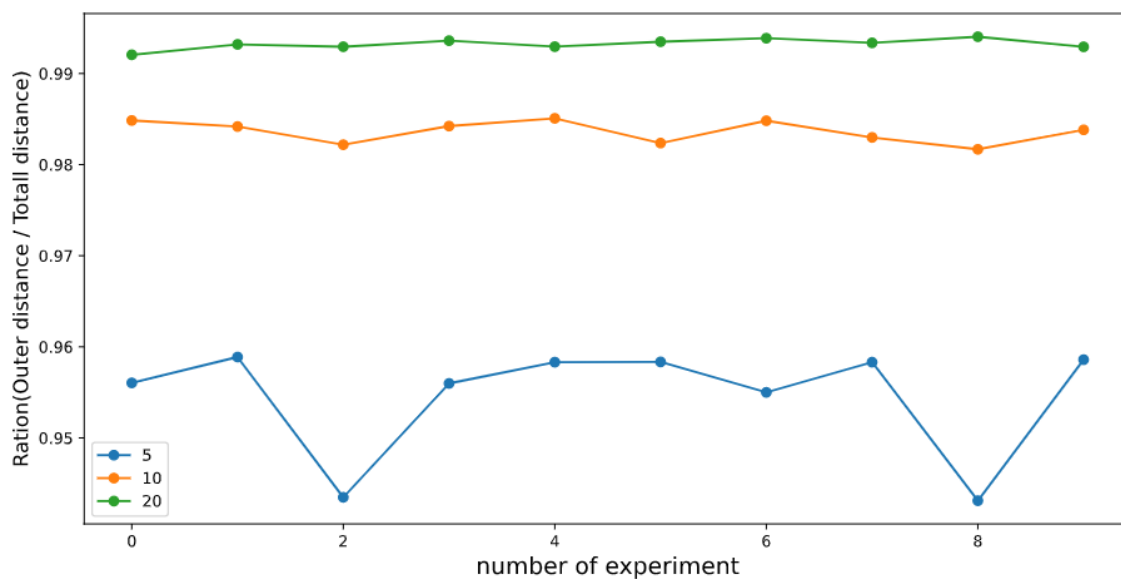
طبق رابطه بالا ، هر چه قدر مقدار Inner distance کمتر باشد و مقدار Outer distance بیشتر باشد ، خوشه بندی ما بهتر خواهد بود و بنابراین هر چه قدر Ratio عدد نزدیکتری به 1 پیدا کند ، می‌توان گفت که خوشه بندی بدست آمده به طور کلی وضعیت بهتری دارد .



شکل 2-1-2 : kde بدست آمده از Ratio های بدست آمده برای 10 بار تکرار آزمایش

طبق نمودار بالا ، kde های مربوط به خوشه های مختلف از هم مجزا هستند و این به این معنی است که با افزایش تعداد خوشه ها ، معیار Ratio بهبود پیدا کرده است که تا حدی قابل پیش بینی نیز می‌باشد . زیرا با افزایش خوشه ها مقدار Inner distance (Loss) کاهش پیدا می‌کند و بنابراین مخرج کسر کاهش پیدا کرده و Ratio افزایش پیدا می‌کند.

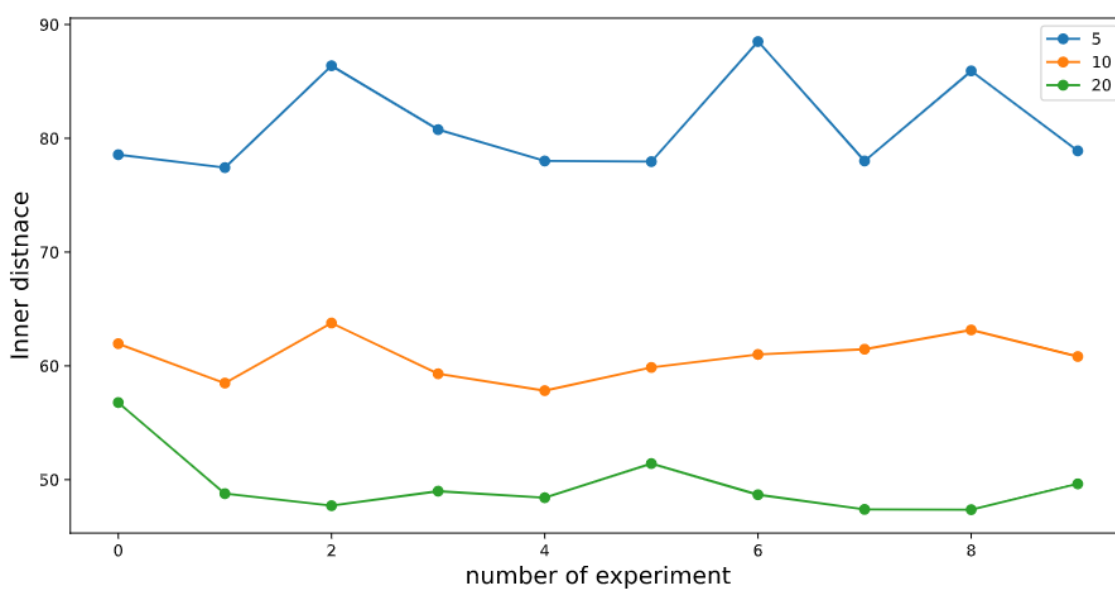
همچنین نمودار بالا را می‌توان به صورت دیگری نیز نمایش داد . در واقع این بار مقادیر Ratio را بر حسب تعداد آزمایش رسم می‌کنیم :



شکل 2-1-3: Ratio بر حسب تعداد آزمایش

همانطور که مشاهده می‌شود، در تمامی آزمایش‌ها مقادیر Ratio برای تعداد خوشه‌های متفاوت از هم مجزا می‌باشد.

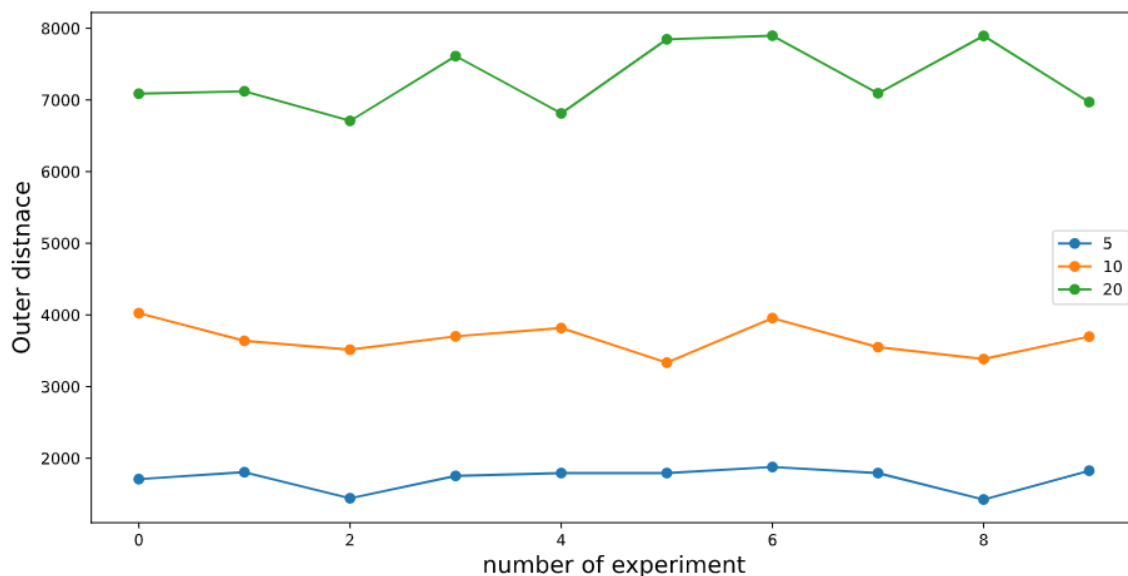
همچنین نمودار Inner distance (Loss) را بر حسب تعداد دفعات آزمایش رسم می‌کنیم:



شکل 2-1-4: مقدار Loss بر حسب تعداد آزمایش

طبق مشاهدات بالا، برای تمامی آزمایش‌های انجام شده، مقدار Loss بدست آمده برای هر خوشه از یکدیگر مجزا می‌باشند.

در نهایت مقدار فاصله بین cluster (Outer distance) را برای هر خوشه رسم می کنیم . توجه کنید که کمینه کردن Inner distance می تواند معادل بیشینه کردن Outer distance باشد ، از آنجایی که جمع این دو برابر مقدار ثابتی می باشد .



شکل 5-1-2: مقدار Outer distance بر حسب تعداد آزمایش

تا الان طبق مشاهداتی که داشتیم ، با افزایش تعداد خوشه ها وضع به صورت کلی بهتر می شد و با بررسی معیار های بالا نیز این امر را بررسی کردیم . ولی سوال اصلی اینجاست که این افزایش تعداد خوشه تا چه حد تاثیر گذار خواهد بود ؟

در واقع باید بررسی کنیم که همچنان با افزایش تعداد خوشه ها ، مقدار Loss/Ratio همچنان به همان حد افزایش/کاهش پیدا می کند یا خیر . به این منظور از روش Elbow استفاده می کنیم .

در ابتدا برای کل تعداد آزمایش های انجام شده ، مقادیر میانگین و واریانس Ratio را حساب می کنیم :

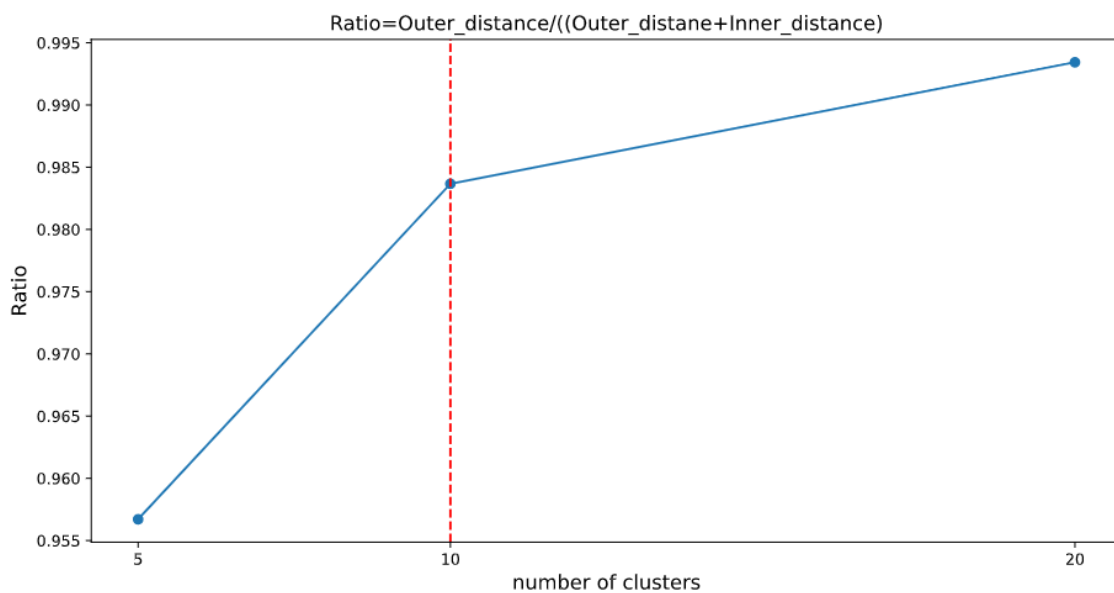
```
print(f"Mean of all experiments for 5,10,20 clustes are :{np.mean(Coeff,axis=0)}\n")
print(f"Variance of all experiments for 5,10,20 clustes are :{np.var(Coeff,axis=0)}")
```

Mean of all experiments for 5,10,20 clustes are : [0.95670486 0.98366938 0.9934309]

Variance of all experiments for 5,10,20 clustes are : [2.41284530e-05 3.44643520e-06 2.56675915e-07]

شکل 6-1-2: مقدار میانگین و واریانس بدست آمده از Ratio برای تعداد خوشه مختلف

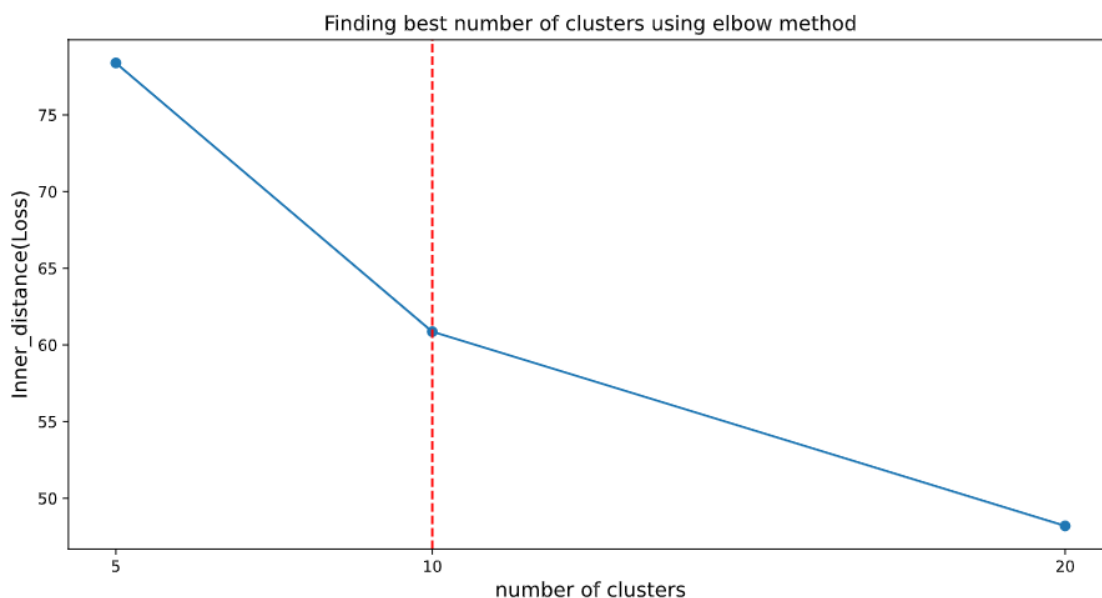
حال مقادیر میانگین بدست آمده را در یک نمودار رسم می کنیم :



شکل 7-1-2: نمودار میانگین Ratio بدست آمده بر حسب تعداد خوشه

طبق نمودار بالا، در $k=10$ ، نمودار حالت Elbow به خود گرفته است و نشان می‌دهد که با افزایش تعداد 10 خوشه به 20 خوشه، performance نسبت به حالت 5 خوشه به 10 خوشه افزایش چندانی پیدا نمی‌کند.

همینطور همین نمودار را می‌توان برای میانگین بدست آمده از Loss نیز انجام داد:



شکل 8-1-2: نمودار میانگین Loss بدست آمده بر حسب تعداد خوشه

در این نمودار هم، نقطه $k=10$ ، نقطه زانوئی بوده و بنابراین می‌توان تعداد 10 خوشه را به عنوان تعداد بهینه بدست آمده اعلام کرد.

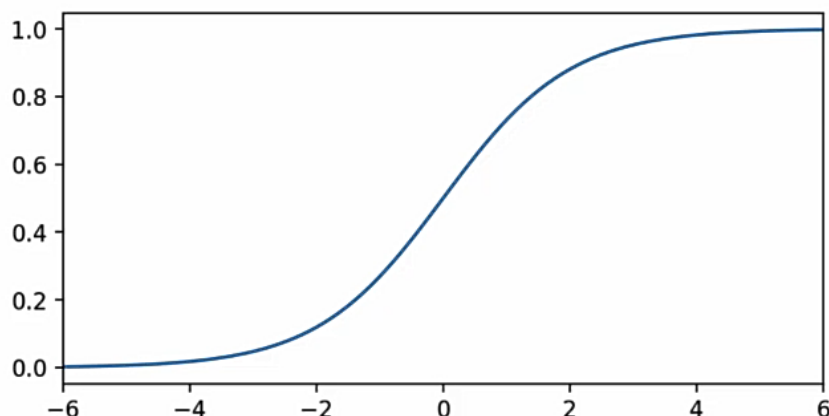
سوال 3:

(الف)

Logistic Regression با انتقال Linear regression به یک مدل قابل طبقه بندی عمل می کند.

فرم تابع کلی Logistic regression یک sigmoid می باشد :

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

شکل 1-1-3 : تابع Sigmoid

طبق شکل بالا ، محور y ها ، احتمال تعلق به یک کلاس می باشد به طوریکه اگر ورودی x ، بزرگتر از صفر باشد ، مقدار خروجی بزرگتر از 0.5 بوده و متعلق به کلاس A است. در غیر این صورت ، کوچکتر از 0.5 بوده و متعلق به کلاس B خواهد بود .

حال می دانیم که فرم تابع Linear regression به صورت زیر می باشد :

$$\hat{y} = \sum_{i=0}^n \beta_i * X_i$$

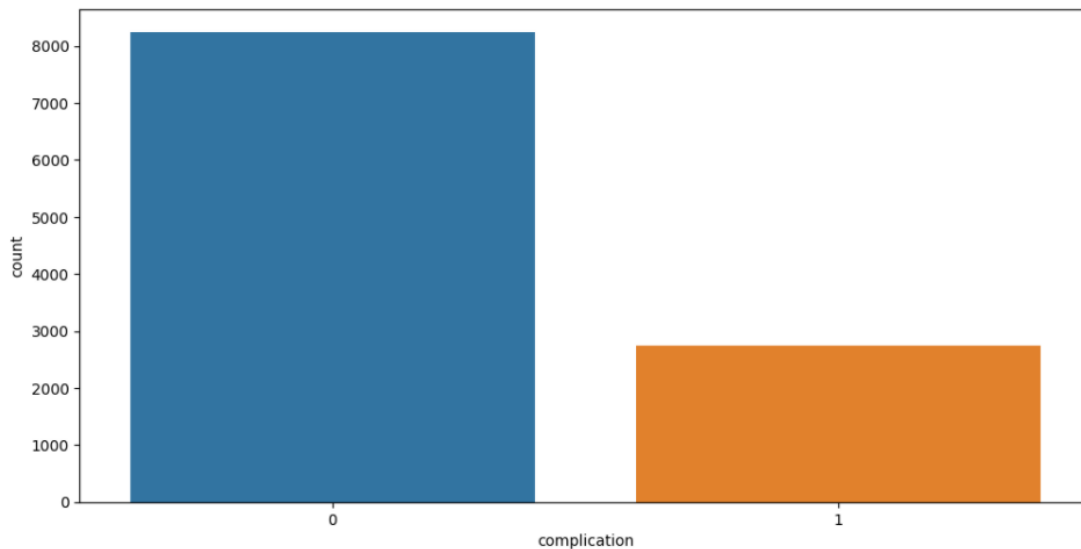
پس کافی است تا \hat{y} بالا را به عنوان ورودی تابع sigmoid بالا بدهیم :

$$\hat{y} = \frac{1}{1 + e^{-\sum_{i=0}^n \beta_i * X_i}}$$

در ادامه طبق توضیحات سوال ، دادگان را به دو دسته آموزش (با لیبل و بدون لیبل) و آزمون تقسیم می کنیم .

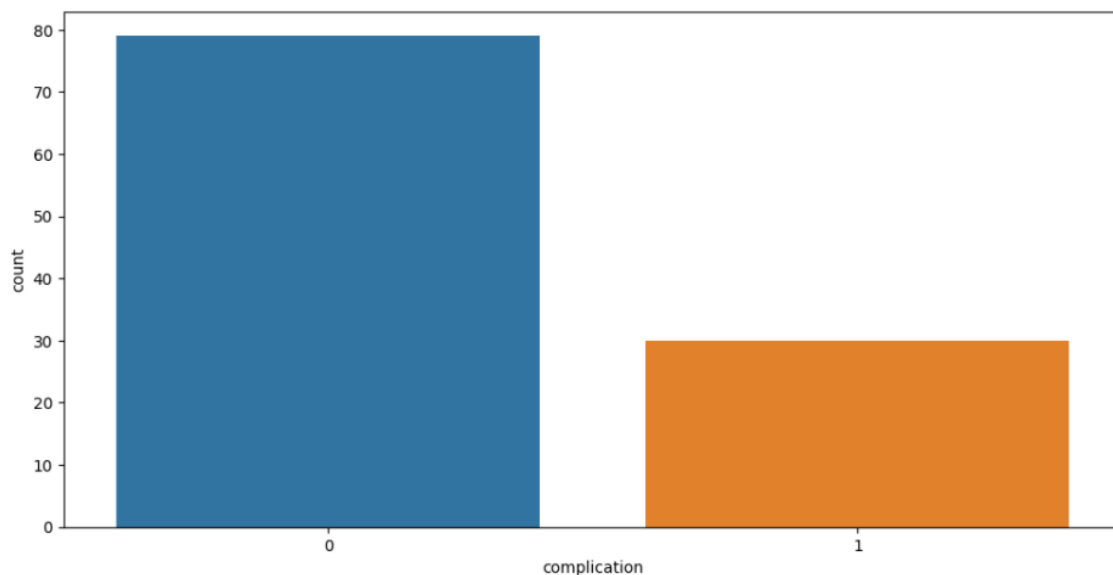
در ادامه count plot مربوط به دادگان آموزش و همچنین دادگان دارای لیبل در مجموعه آموزش را رسم می کنیم :

```
plt.figure(figsize=(12,6),dpi=100)
sns.countplot(data=train,x='complication');
```



شکل 2-1-3 : *count plot* مربوط به 75 درصد دادگان آموزش

```
plt.figure(figsize=(12,6),dpi=100)
sns.countplot(data=train_labeled,x='complication');
```



شکل 2-1-3 : *count plot* مربوط به 1 درصد دادگان آموزش با لیبل

طبق هر دو شکل آورده شده در بالا ، دیتاست داده شده ، *unbalanced* بوده و تعداد دادگان با لیبل 0 تا 3 برابر دادگان با لیبل 1 می باشد .

(ب)

طبق کلاس logistic regression در کتابخانه sklearn ، دادگان لیبل دار را آموزش می دهیم :

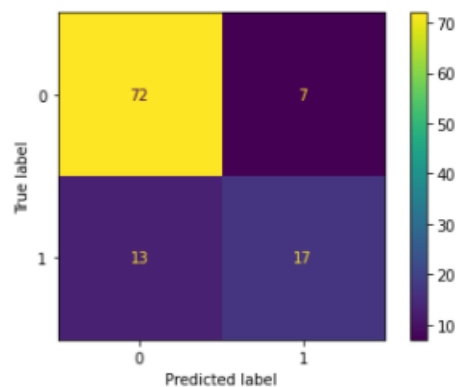
دقت در بین دادگان لیبل دار (1%) :

```
y_pred = log_model.predict(X)
```

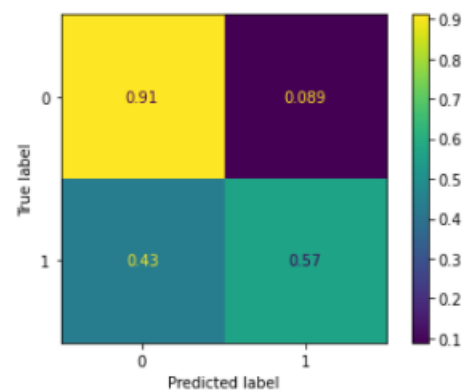
```
accuracy_score(y,y_pred)
```

```
0.8165137614678899
```

شکل 1-2-3 : دقت طبقه بند بر روی دادگان آموزش



شکل 2-2-3 : ماتریس آشفتگی طبقه بند بدست آمده

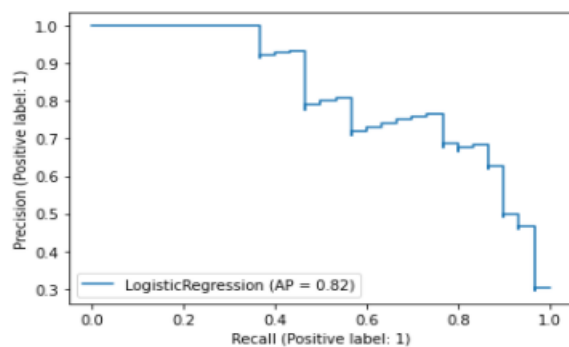


شکل 3-2-3 : ماتریس آشفتگی طبقه بند بدست آمده (نرمال شده)

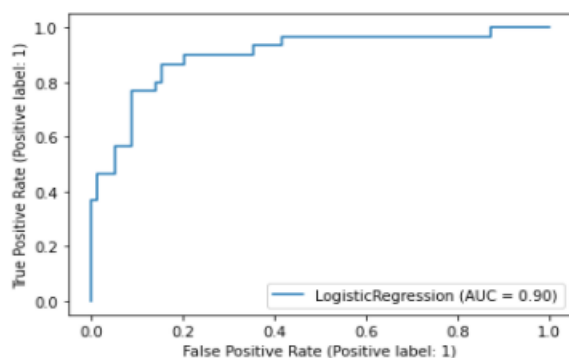
```
print(classification_report(y,y_pred))
```

	precision	recall	f1-score	support
0	0.85	0.91	0.88	79
1	0.71	0.57	0.63	30
accuracy			0.82	109
macro avg	0.78	0.74	0.75	109
weighted avg	0.81	0.82	0.81	109

شکل 4-2-3 : گزارش طبقه بند بر روی دادگان آموزش



شکل 4-2-3 : نمودار precision-recall طبقه بند



شکل 5-2-3 : نمودار ROC طبقه بند

🕒 دقت در بین دادگان آزمون (25%) :

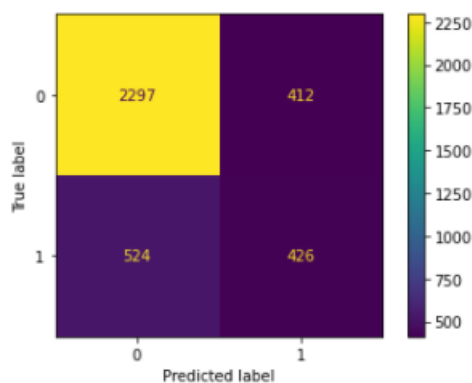
```
accuracy_score(y_test,y_pred_test)
```

```
0.7441924022957093
```

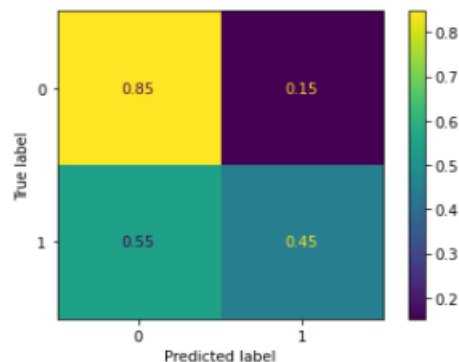
```
f1_score(y_test,y_pred_test,average=None)
```

```
array([0.83074141, 0.47651007])
```

شکل 6-2-3 : Accuracy , F1-score طبقه بند



شکل 7-2-3 : ماتریس آشفتگی طبقه بند



شکل 8-2-3: ماتریس آشفتگی طبقه بند (نرمال شده)

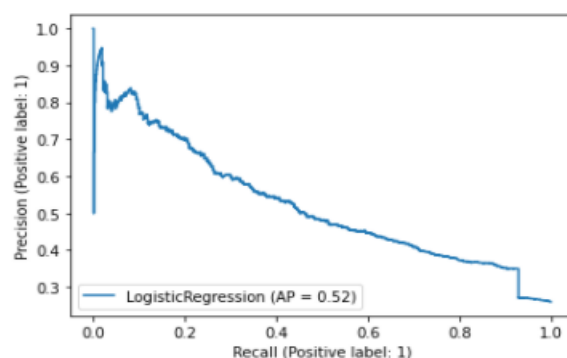
*همانطور که مشاهده می‌شود ، مدل بر روی دادگانی که از قبل ندیده است ضعیف تر عمل می‌کند. همچنین به دلیل نوع خاص دیتاست که مربوط به آمار پزشکی می‌باشد ، مقدار TP برای ما از اهمیت بیشتری برخوردار است . بنابراین به عنوان یک معیار ، مقدار $\frac{TP}{FN}$ را در نظر می‌گیریم که همان نسبت مستقیمی با recall دارد .

از طرفی precision نیز از مقدار TP استفاده می‌کند پس می‌توان گفت معیار F1-score هم که در صورت سوال ذکر شده معیار مناسبی می‌باشد .

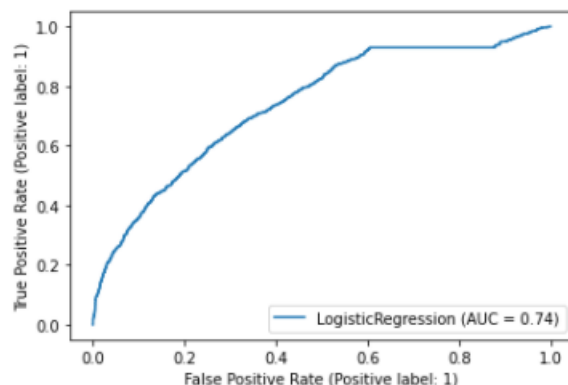
در اینجا مقدار recall=0.45 می‌باشد . حال سعی می‌کنیم در بخش های بعد این مقدار را بهبود دهیم .

	precision	recall	f1-score	support
0	0.81	0.85	0.83	2709
1	0.51	0.45	0.48	950
accuracy			0.74	3659
macro avg	0.66	0.65	0.65	3659
weighted avg	0.73	0.74	0.74	3659

شکل 9-2-3: گزارش طبقه بند بر روی دادگان آزمون



شکل 9-2-3: نمودار precision-Recall

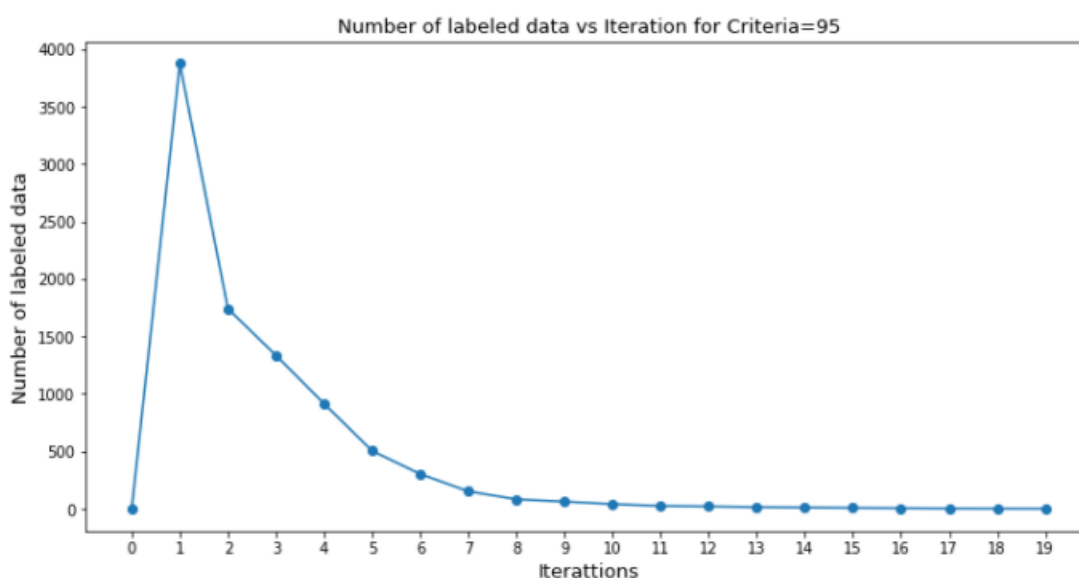


شکل 10-2-3 : نمودار ROC

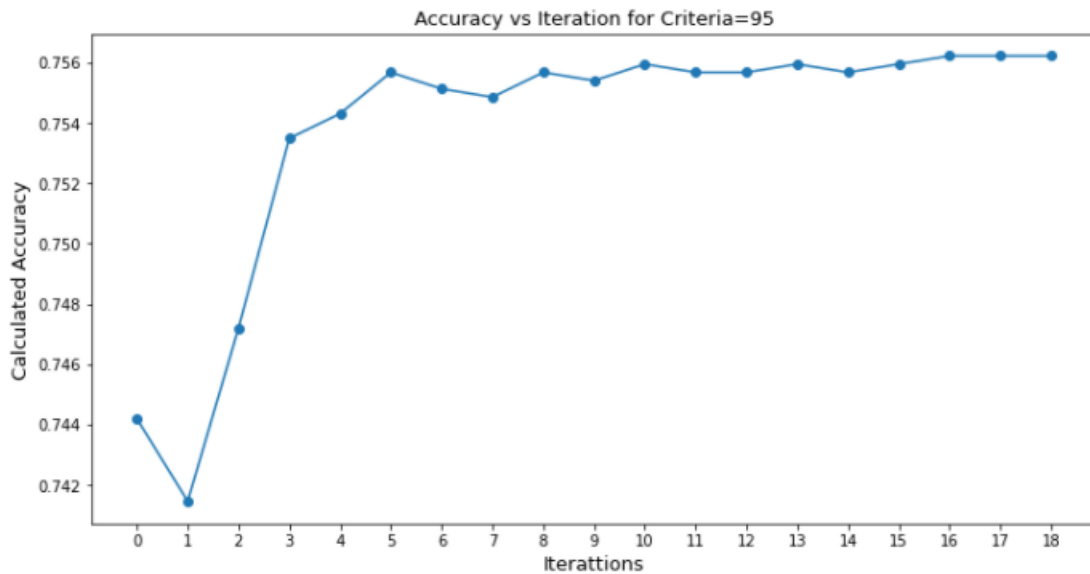
ج/د)

در این قسمت قصد داریم به کمک الگوریتم self-training ، دادگان را به صورت semi-supervised آموزش دهیم. طبق این الگوریتم ، ابتدا دادگان خام با لیبل را در نظر می گیریم و یک بار شبکه را آموزش می دهیم. سپس بر روی 74% دادگان آموزش بدون لیبل صرفا پیشبینی (predict) انجام می دهیم به صورتی که متناظر با هر Sample ، احتمال تعلق آن به کلاس صفر یا یک را بدست می آوریم. حال با کمک تعیین یک Criteria ، تنها Sample هایی که بیشترین مقدار احتمالی تعلق به یک کلاس در آنها از Criteria داده شده بیشتر باشد را انتخاب می کنیم و دادگان آموزش لیبل دار را بروز رسانی می کنیم و دوباره احتمالات جدید را بدست می آوریم. این الگوریتم را تا جایی ادامه می دهیم که داده ای برای اضافه شدن به مجموعه لیبل دار وجود نداشته باشد.

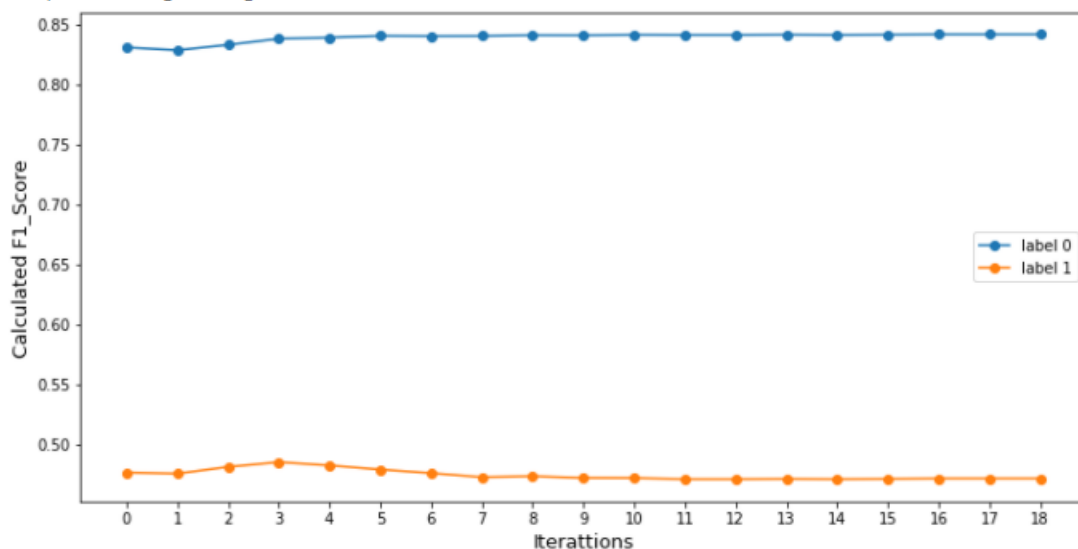
ابتدا برای Criteria=95 ، سه نمودار مختلف تعداد داده های اضافه شده ، Accuracy و معیار F1 را رسم می کنیم :



شکل 1-3-3 : تعداد دادگان لیبل دار اضافه شده در هر epoch



شکل 2-3-3: تعداد دادگان لیبل دار اضافه شده در هر epoch



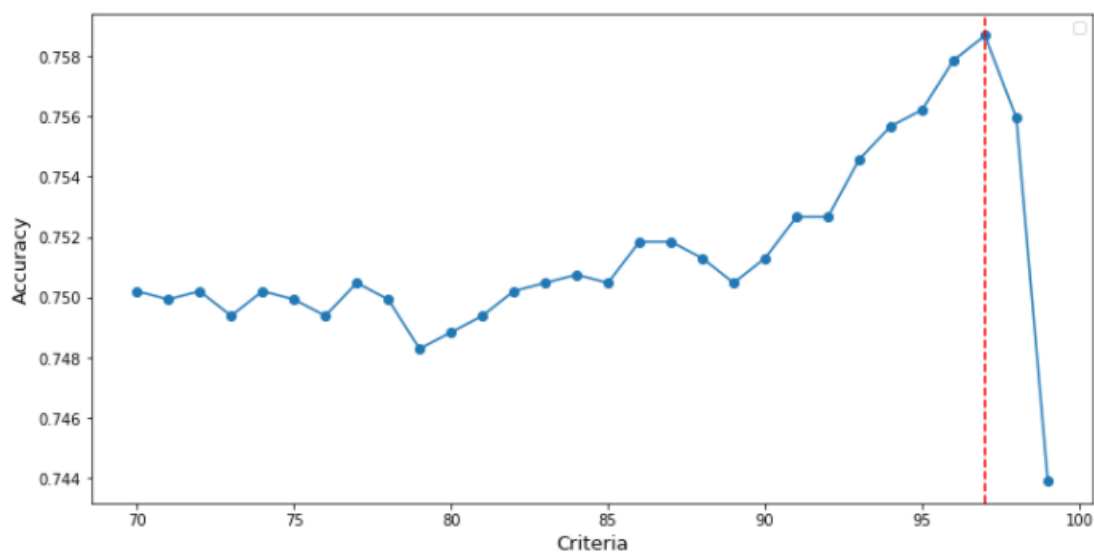
شکل 3-3-3: F1-score مربوط به هر لیبل (0 و 1) بر حسب epoch

*طبق نمودار اول ، مشاهده می شود که در epoch های اولیه (دو epoch اول) ، تعداد دادگان لیبل خورده افزایش می یابد و پس از آن با ادامه الگوریتم ، تعداد آنها کمتر شده تا هنگامی که به صفر می رسد و شرط توقف الگوریتم می باشد.

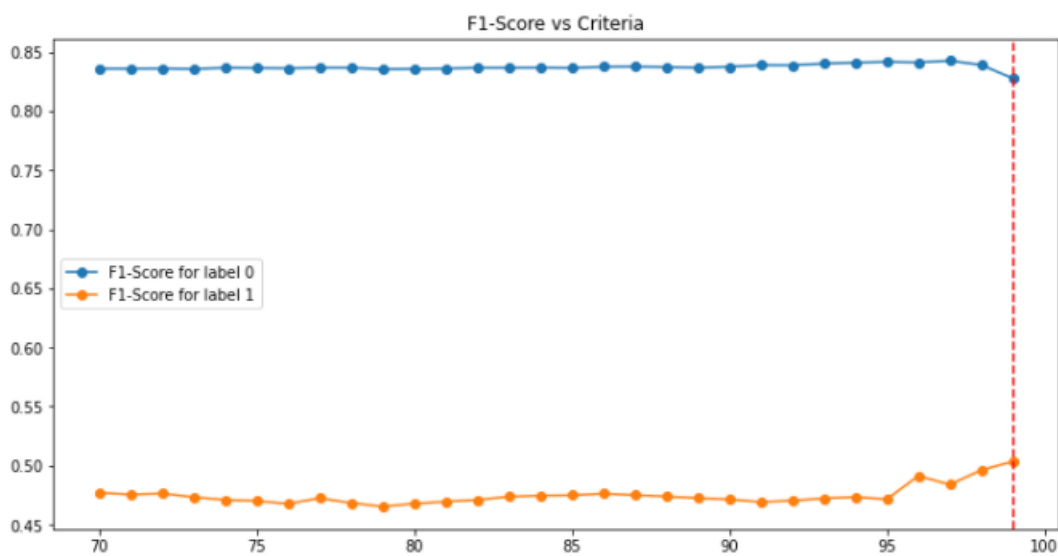
*در نمودار دوم ، مشاهده می شود که با افزایش epoch ها ، دقت تا حدودی افزایش پیدا کرده و بعد ثابت می ماند . در اینجا تا 1 درصد دقت در مجموعه آزمون افزایش یافته است .

*در نمودار سوم ، می توان مشاهده کرد که برای Criteria=95 ، معیار F1 برای دو لیبل تقریباً تغییر خاصی نکرده و ثابت مانده است .

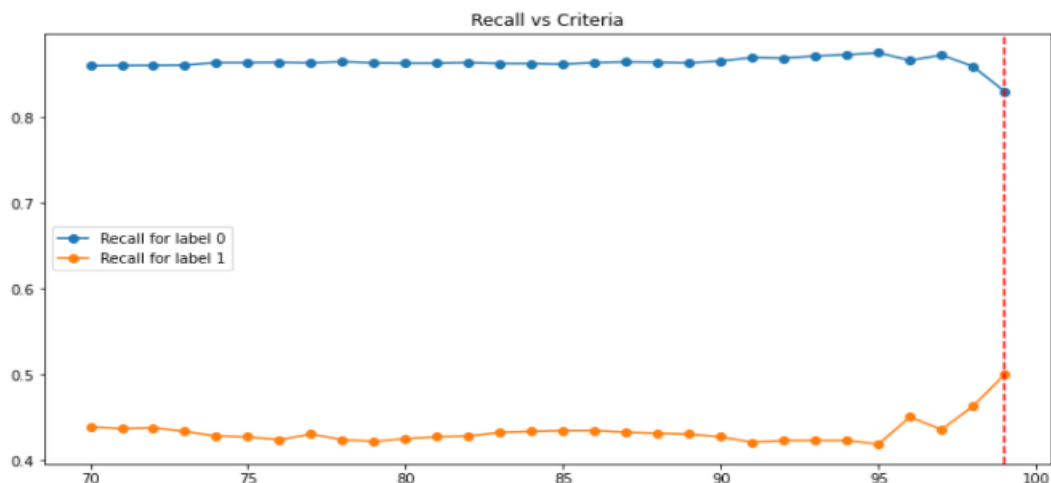
حال قصد داریم به ارزیابی Criteria های مختلف، بهترین معیار دقت (Accuracy)، F1-score و Recall که در طول Iteration های مختلف بدست آمده است را رسم کنیم:



شکل 3-4: بهترین دقت Accuracy بدست آمده در هر Criteria



شکل 3-5: بهترین F1-score بدست آمده در هر Criteria



شکل 3-3-6: بهترین بدست آمده در هر Criteria

طبق نمودار های بدست آمده ، تحلیل های زیر را ارائه می دهیم :

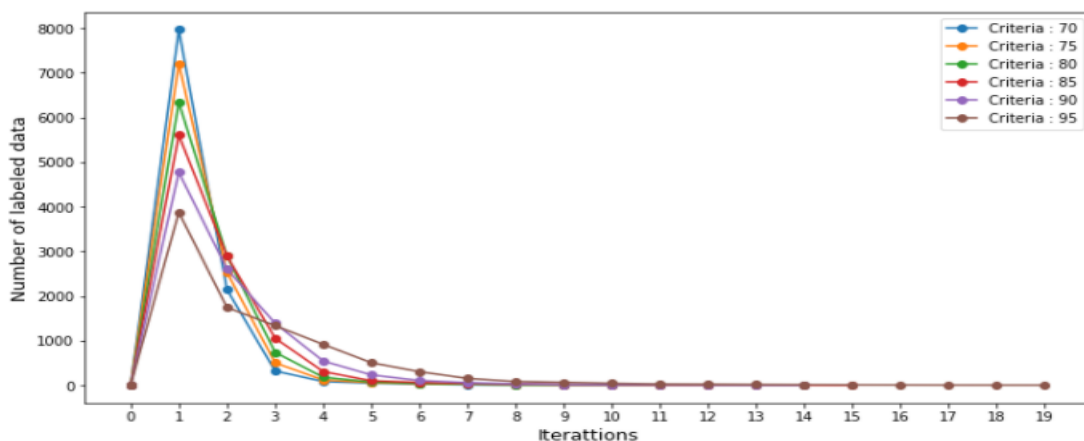
*در نمودار اول (Accuracy) ، بهترین دقت مربوط به Criteria=97 می باشد و بعد از آن دقت کاهش پیدا کرده است .

*در نمودار دوم (F1) ، بهترین F1-score مربوط به Criteria=99 می باشد و تا Criteria=95 ، این معیار جهش خاصی نداشته است .

*در نمودار سوم (Recall) ، بهترین F1-score مربوط به Criteria=99 می باشد و تا Criteria=95 ، این معیار جهش خاصی نداشته است .

همانطور که مشاهده شد ، روند تغییرات نمودار دوم و سوم مشابه هم می باشد و استفاده از هر دوی این معیارها درک خوبی از روند تغییرات TP می دهد.

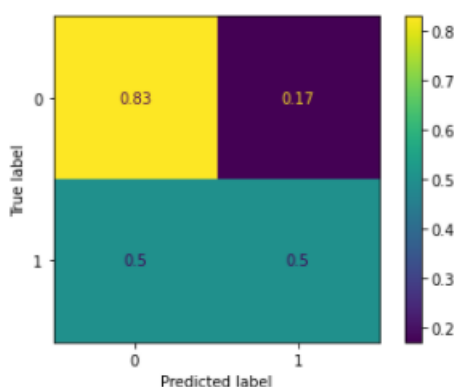
در ادامه ، روند تغییرات دادگان لیبل خورده در طول epoch های مختلف را بر حسب 6 مقدار مختلف Criteria رسم می کنیم :



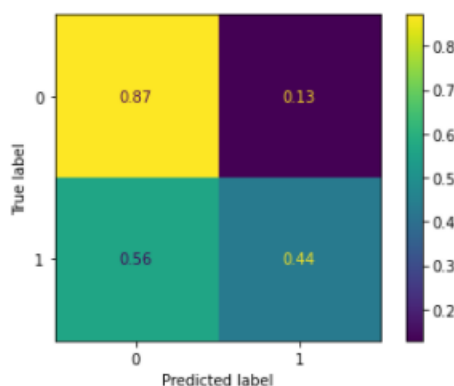
شکل 3-3-7: تعداد دادگان لیبل خورده در هر epoch بر حسب Criteria

طبق نمودار بالا ، با افزایش criteria ، نقطه پیک نمودار (در اولین Epoch) کاهش پیدا می کند که منطقی می باشد زیرا با افزایش Criteria سخت گیرانه تر عمل می کنیم . در عوض شیب نمودار در ادامه کاهش پیدا کرده و تعداد Iteration ها افزایش پیدا می کند. دلیل را می توان اینگونه توجیه کرد که با اینکه با شرایط سخت گیرانه تر تعداد داده لیبل خورده کمتری را در ابتدا اضافه می کنیم ولی داده های اضافه شده قابل اطمینان تر هستند و بنابراین در epoch های بعد ، مدل با اطمینان بیشتری تصمیم می گیرد و بهتر آموزش می یابد پس با دقت بیشتری به دادگان لیبل خورده اضافه می کند و شیب نمودار به سرعت صفر نمی شود.

در ادامه ، ماتریس آشفتگی را برای بهترین Criteria بدست آمده (برای Accuracy برابر 97 و F1 و Recall برابر 99) رسم می کنیم :



شکل 8-3-3 : ماتریس آشفتگی بدست آمده برای بهترین Criteria بدست آمده از معیار F1/Recall



شکل 8-3-3 : ماتریس آشفتگی بدست آمده برای بهترین Criteria بدست آمده از معیار Accuracy

همانطور که در شکل بالا مشاهده می کنید ، با معیار F1/Recall ، مقدار TP برابر 0.5 بدست آمده است در حالیکه با معیار Accuracy ، معیار TP برابر 0.44 بدست آمده است .

طبق قسمت های قبل ، مقدار TP در حالتیکه تنها از 1% دادگان لیبل خورده استفاده می کردیم برابر 0.45 بود . با مقایسه مقادیر بدست آمده می توان گفت ، وقتی از معیار Accuracy برای بهترین Criteria استفاده می کنیم ، مدل سعی می کند تعداد TN ها را افزایش دهد که در اینجا می بینیم که TN بدست

آمده برای معیار Accuracy ، 0.87 بوده که نسبت به حالت اولیه (0.85) افزایش یافته است در حالیکه مقدار TP آن 1 درصد کاهش داشته است .

ولی هنگامی که از معیار F1/Recall در انتخاب بهترین Criteria استفاده می کنیم ، مقدار TP به 0.5 رسیده که نشان می دهد 6 درصد در انتخاب لیبل مثبت دقیق تر شده ایم که برای این دیتاست ، اتفاق خوبی می باشد و از طرف دیگر مقدار TN کاهش یافته است که در اینجا برای ما زیاد اهمیت ندارد.

سوال 4:

(الف)

شیان و مصنف - سوال چهارم (مقدمت احتمال) بنام خدا ۸۱۰۱۹۷۲۰۳

الف) سوالات تفسیری:

الف - اگر در انتعاش در این سوال این است که یک یک به یک unbiased (یعنی شیب خط برابری نیست) یک یک در حالت بیased داشته باشیم. راجع به انتعاش از تقابل می باشد. چون یک یک یک را دوبار تیراج می کنیم. اگر بار اول بیشتر بار دوم خط آمدن آن حالت را منفی و اگر بار اول خط و بار دوم بیشتر بود آن حالت را یک بنامیم. اگر غیر این دو حالت بیش (HT) (TH)

آمد (TH, HH). آنگاه این کار را تکرار کنیم تا یک حالت unbiased برسیم.

اگر یک بار را (Round)، تعداد ۲ بار آزمایش یک یک که بنامیم، واضح است که احتمال تولید حالت منفی یا حالت یک یک در حالت است. پس یک یک به یک unbiased ایجاد کنیم.

محیط طبق آنگونه که می توان تعداد متوسط تیراج یک یک (ps) را برای یک حالت به احتمال شیب بودن p می باشد و حساب کرد: چون یک یک تعداد تیراج یک یک برابر t باشد. اگر در اول موفق شویم، در دوم یعنی از دو حالت دوم که دقیقاً ۲ تیراج داریم. حال در صورتی که در اول موفق نشویم، در دوم از اول باید شروع کنیم $(2+t)$. بنابراین امید ریاضی t در رابطه زیر صدق می کند:

$$E(t) = \frac{2p(1-p) \times 2 + (1-p(1-p)) \times (2+E(t))}{HT \text{ or } TH} \quad t = E(t) \quad t = \frac{2}{2p(1-p)}$$

برای مثال، برای $p = \frac{2}{3}$ داریم:

$$t = \frac{2}{2 \times \frac{2}{3} \times \frac{1}{3}} = \frac{2}{\frac{4}{9}} = \frac{9}{2} = 4.5$$

پس برای موفقیت در ایجاد یک از دو حالت منفی یا یک باید به طور متوسط ۴.۵ بار یک تیراج کنیم.

الف - ۱-۲: ابتدا متغیر Z را برابر $Z = \min\{X, Y\}$ در نظر بگیریم و Z را می بسوزیم. برای این منظور احتمال زیر را می بسوزیم:

$$P(Z > a) = P(\min\{X, Y\} > a) = P(X > a, Y > a) = P(X > a)P(Y > a) = e^{-a\lambda_1 - a\lambda_2} = e^{-a(\lambda_1 + \lambda_2)}$$

$$F_Z(a) = 1 - P(Z > a) = 1 - e^{-a(\lambda_1 + \lambda_2)} \quad f_Z(z) = (\lambda_1 + \lambda_2)e^{-(\lambda_1 + \lambda_2)z}$$

بنابراین متغیر Z از جنس تابع نمایی با پارامتر $(\lambda_1 + \lambda_2)$ می باشد. در ادامه داریم:

$$Z = \min(X, Y) \rightarrow E(Z) + E(k) = E(Z + k) = E(X + Y) = E(X) + E(Y) = \frac{1}{\lambda_1} + \frac{1}{\lambda_2}$$

$$k = \max(X, Y) \rightarrow E(k) = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - \frac{1}{\lambda_1 + \lambda_2} = \frac{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2}{\lambda_1\lambda_2(\lambda_1 + \lambda_2)}$$

عبارت کرنی می داریم: $E(Z) = \frac{1}{\lambda_1 + \lambda_2}$ پس داریم:

در نهایت برای جوابش بین Z و k داریم:

$$Cov(Z, k) = E(Zk) - E(Z)E(k) = E(XY) - E(Z)E(k)$$

پس بازنویس متغیر بودن X و Y :

$$E(X)E(Y) = \frac{1}{\lambda_1} \times \frac{1}{\lambda_2}$$

$$\rightarrow Cov(Z, k) = \frac{1}{\lambda_1\lambda_2} - \frac{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2}{\lambda_1\lambda_2(\lambda_1 + \lambda_2)} \times \frac{1}{\lambda_1\lambda_2} = \frac{(\lambda_1 + \lambda_2)^2 - \lambda_1^2 - \lambda_2^2 - \lambda_1\lambda_2}{\lambda_1\lambda_2(\lambda_1 + \lambda_2)^2} = \frac{1}{(\lambda_1 + \lambda_2)^2}$$

شکل 4-1-1: محاسبات بخش الف

الف - 2-2:

$$V, U \sim N(\mu, \sigma^2) \quad * \max\{U, V\} + \min\{U, V\} = U + V$$

$$\rightarrow \text{Cov}(V, \max(V, U)) + \text{Cov}(V, \min(V, U)) = \text{Cov}(V, \max(V, U) + \min(V, U)) = \text{Cov}(V, U + V)$$

$$= \text{Cov}(V, V) = \text{var}(V)$$

* از طرفی می توانیم \min, \max را صورت زیر رسم می دهیم:

$$\max(V, U) = -\min(-V, -U)$$

$$\rightarrow \text{Cov}(V, \max(V, U)) = \text{Cov}(V, -\min(-V, -U)) = \text{Cov}(-V, \min(-V, -U)) \xrightarrow[V \rightarrow -V]{U \rightarrow -U} = \text{Cov}(V, \min(V, U))$$

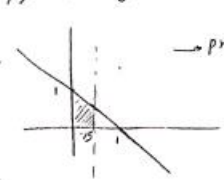
* $\rightarrow \text{Cov}(V, \max(V, U)) = \text{Cov}(V, \min(V, U)) \xrightarrow{(2)} 2 \text{Cov}(V, \max(V, U)) = \text{var}(V)$

$$\rightarrow \text{Cov}(V, \max(V, U)) = \text{Cov}(V, \min(V, U)) = \frac{1}{2} \text{var}(V)$$

الف - 2-3:

$$f(x, y) = C(1-x-y), \quad x > 0, y > 0, x+y < 1$$

* $\text{Pr}(X < 1/5)$:



$$\rightarrow \text{Pr}(X < 1/5) = \int_{-\infty}^{\infty} \int_{-\infty}^{1/5} f(x, y) dx dy = \int_{y=0}^{1/5} \int_{x=0}^{1/5} C(1-x-y) dx dy$$

$$= \int_{y=0}^{1/5} \int_{x=0}^{1/5} (C - Cx - Cy) dx dy = \int_{x=0}^{1/5} (C - Cx)(1-x) dx = \int_{x=0}^{1/5} (Cx - Cx^2) dx = \left[\frac{Cx^2}{2} - \frac{Cx^3}{3} \right]_{x=0}^{1/5} = \frac{7C}{48}$$

→ برای یابی C از رابطه نرمالیزاسیون می کنیم:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = 1 \rightarrow \int_{x=0}^1 \int_{y=0}^{1-x} C(1-x-y) dy dx = 1$$

$$\rightarrow \left(\frac{Cx}{2} - \frac{Cx^2}{2} + \frac{Cx^3}{6} \right) \Big|_{y=0}^{1-x} = -\frac{C}{6} = 1 \rightarrow C = 6 \rightarrow \text{Pr}(X < 1/5) = \frac{7}{48} \times 6 = \frac{7}{8}$$

* $E(X+Y)$:

$$= \int_{x=0}^1 \int_{y=0}^{1-x} (x+y) 6(1-x-y) dy dx = 6 \int_{x=0}^1 \int_{y=0}^{1-x} (x+y - x^2 - xy - y^2 - xy) dy dx$$

$$= 6 \int_{x=0}^1 \left((x-x^2)(1-x) - 2x \frac{(1-x)^2}{2} + \frac{(1-x)^3}{3} - \frac{(1-x)^3}{3} \right) dx = \int_{x=0}^1 (6x - 12x^2 + 6x^3 - 6x^3 + 12x^2 - 3 + 3x^4 - 6x^4 + 2x^4 - 6x^4) dx$$

$$= \int_{x=0}^1 (1 - 3x^2 + 2x^4) dx = \left[x - x^3 + \frac{2x^5}{5} \right]_{x=0}^1 = 1 - 1 + \frac{2}{5} = \frac{2}{5}$$

شکل 4-1-2: ادامه محاسبات بخش الف

اگر معیار زمان را امت میزنیم، $\text{Pr}(X < 1/5) = \frac{7}{8}$ نشان می دهد که احتمال اینکه کارمند اول کمتر از 15 ساعت بر روی پروژه وقت بگذارد برابر $\frac{7}{8}$ می باشد. این به این معنی است که احتمالاً کارمند دوم بخش بیشتری از کار را جلو می برد. همچنین $E(X+Y) = 1/5$ نشان می دهد که بطور متوسط این دو کارمند بر روی پروژه چه مقدار زمان صرف می کنند که در اینجا برابر 1/5 ساعت می باشد.

شکل 4-1-3: ادامه محاسبات بخش الف

(ب

ب-1:

برای شبیه سازی به کمک کتابخانه datetime و timedelata اقدام به ایجاد یک روز مشخص در یک سال می کنیم . در ادامه برای تعداد رندوم 23 ، به تعداد 23 تاریخ تولد در سال 2022 ایجاد می کنیم :

```
random_birthdays(23)
```

```
[datetime.datetime(2022, 5, 22, 0, 0),  
datetime.datetime(2022, 7, 16, 0, 0),  
datetime.datetime(2022, 8, 28, 0, 0),  
datetime.datetime(2022, 6, 3, 0, 0),  
datetime.datetime(2022, 5, 1, 0, 0),  
datetime.datetime(2022, 12, 27, 0, 0),  
datetime.datetime(2022, 6, 16, 0, 0),  
datetime.datetime(2022, 6, 1, 0, 0),  
datetime.datetime(2022, 1, 15, 0, 0),  
datetime.datetime(2022, 12, 31, 0, 0),  
datetime.datetime(2022, 3, 2, 0, 0),  
datetime.datetime(2022, 9, 11, 0, 0),  
datetime.datetime(2022, 1, 18, 0, 0),  
datetime.datetime(2022, 5, 25, 0, 0),  
datetime.datetime(2022, 10, 18, 0, 0),  
datetime.datetime(2022, 12, 20, 0, 0),  
datetime.datetime(2022, 4, 28, 0, 0),  
datetime.datetime(2022, 10, 5, 0, 0),  
datetime.datetime(2022, 2, 27, 0, 0),  
datetime.datetime(2022, 4, 20, 0, 0),  
datetime.datetime(2022, 9, 19, 0, 0),  
datetime.datetime(2022, 2, 23, 0, 0),  
datetime.datetime(2022, 2, 14, 0, 0)]
```

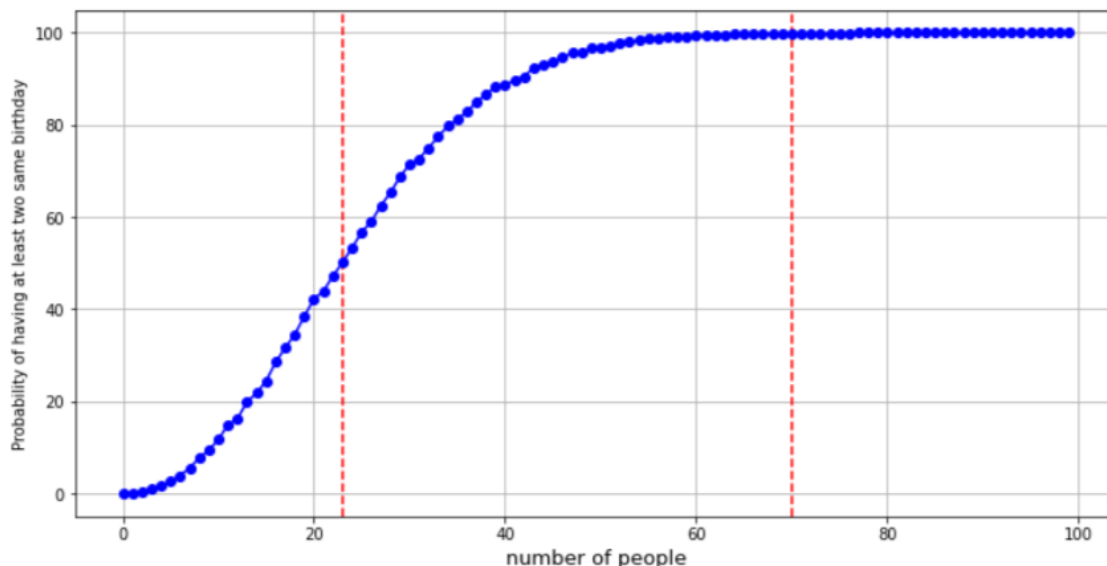
شکل 4-2-1-1: تولید 23 تاریخ تولد در سال 2022 به صورت رندوم

در ادامه تابعی به نام 'determine_probability' می نویسیم که با گرفتن دو پارامتر تعداد افراد و تعداد آزمایش ، به طور میانگین ، احتمال اینکه حداقل دو نفر دارای تاریخ تولد یکسانی باشند را بر میگرداند :

```
def determine_probability(number_of_people, run_amount=10000):  
    dups_found = 0  
    for i in range(run_amount):  
        birthdays = random_birthdays(number_of_people)  
        duplicates = set(x for x in birthdays if birthdays.count(x) > 1)  
        #print(duplicates)  
        if len(duplicates) >= 1:  
            dups_found += 1  
    return dups_found/run_amount * 100
```

شکل 4-2-1-2: تابع 'determine_probability'

در نهایت به ازای 0 نفر تا 100 نفر ، مقادیر احتمالی بدست آمده را در یک نمودار رسم می‌کنیم و حداقل تعداد افرادی را که نیاز است تا به ترتیب به احتمال 50% و حدود 100% ، حداقل دو نفر دارای تولد یکسان باشند را اعلام می‌کنیم :



شکل 3-1-2-4 : تابع 'determine_probability'

طبق نمودار بالا ، در یک جمع 23 نفره ، احتمال اینکه حداقل دو نفر دارای تاریخ تولد یکسان باشند برابر 50% می‌باشد و همچنین وجود حداقل 70 نفر در یک جمع ، تا اطمینان بسیار خوبی (100%) ، تضمین می‌کند که حداقل دو نفر در جمع دارای تاریخ تولد یکسان هستند.

این مساله ، گویی دچار تناقض می‌باشد . زیرا اصل لانه کبوتری به ما می‌گوید که اگر تعداد روز های سال برابر 365 می‌باشد ، حداقل 366 نفر باید در جمع حضور داشته باشند تا حداقل دو تای آنها دارای تاریخ تولد یکسان باشند.

ب-2:

در این قسمت ، قضیه حد مرکزی را بررسی می‌کنیم . طبق این قضیه اگر یک توزیع آماری دلخواه داشته باشیم که لزوماً توزیع آن نرمال نیست ، اگر از *population* کلی ، در هر بار تعدادی *sample* انتخاب کنیم و میانگین آن *sample* ها را به عنوان نمونه آماری جدید در نظر بگیریم ، می‌توان نشان داد که توزیع میانگین *Sample* های بدست آمده ، توزیعی نرمال با میانگین و واریانس زیر می‌باشد :

$$\mu_{new} = \mu_{population}, \sigma_{new}^2 = \frac{\sigma_{population}^2}{n}$$

که n تعداد *sample* انتخابی از کل *population* می‌باشد .

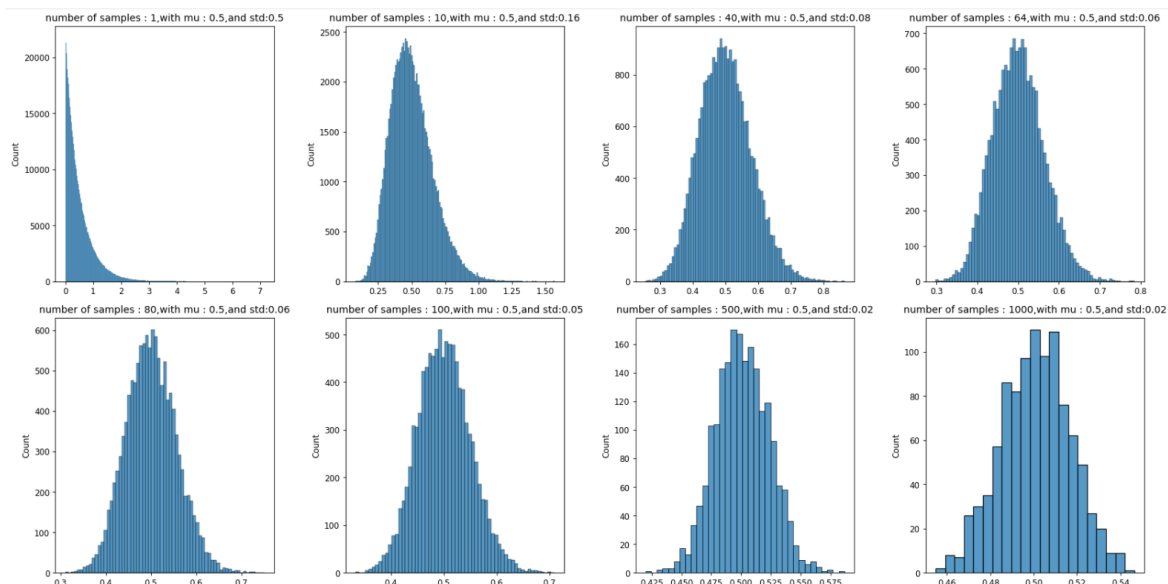
برای بررسی صحت این موضوع ، یکبار برای *population* با توزیع نمایی با $\beta = 2$ or $\lambda = 0.5$ و بار دیگر برای توزیع *binomial* با پارامتر های (20,0.8) آزمایش زیر را انجام می‌دهیم :

توزیع نمایی :

در این آزمایش ابتدا یک لیست از تعداد *Sample* که در هر بار از *population* می‌خواهیم برداشت کنیم ، درست می‌کنیم :

```
list= [1,10,40,64,80,100,500,1000]
```

در ادامه تابعی می‌نویسیم که به ازای مقادیر موجود در لیست بالا ، هیستوگرام میانگین *sample* های برداشت شده را به صورت *subplot* نمایش دهد :

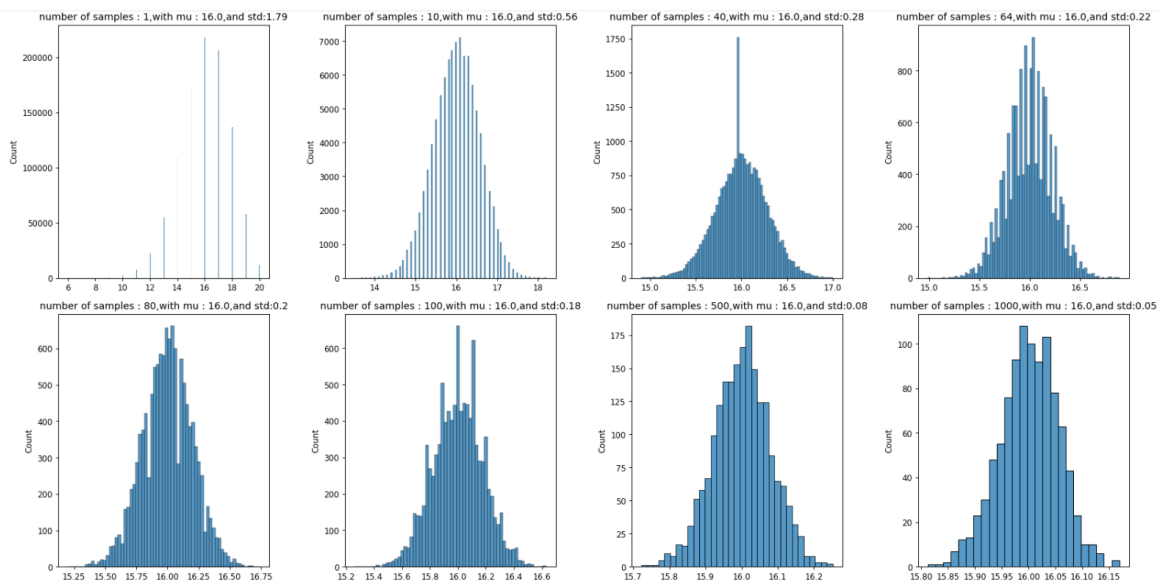


شکل 2-4-2-1: هیستوگرام میانگین بدست آمده به ازای انتخاب تعداد *sample* مختلف برای تابع احتمال نمایی

طبق نمودار بالا مشاهده می‌کنیم که با افزایش تعداد *Sample* برداشتی از *population* ، میانگین توزیع جدید همچنان ثابت می‌ماند ولی σ (Standard deviation) با نسبت $1/\sqrt{n}$ کاهش می‌یابد . این به این معنی است که با افزایش تعداد *sample* برداشتی ، نمودار بدست آمده *sharp* تر می‌شود ولی همچنان میانگین آن تغییر نمی‌کند.

توزیع binomial :

تمام مراحل بخش قبل را این بار برای توزیع binomial تکرار می‌کنیم :



شکل 2-2-2-4: هیستوگرام میانگین بدست آمده به ازای انتخاب تعداد sample مختلف برای تابع احتمال binomial

طبق مشاهدات بالا ، نتایج مانند قسمت قبل می‌باشد و این نشان می‌دهد که نتیجه گیری ما قابل تعمیم است .