به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس سیستم‌های هوشمند

تمرین شماره ۲

نام و نام خانوادگی: سیاوش شمس

شماره دانشجویی: ۸۱۰۱۹۷۶۴۴

آبان ۱۴۰۰

فهرست سوالات

سوال ۱ ..... ۳

الف: ۴

ب: ۶

ج: ۷

سوال ۲ ..... ۸

الف: ۸

ب: ۱۲

ج: ۱۳

سوال ۳ ..... ۱۴

الف: ۱۴

ب: ۱۸

توضیح روش LMNN: ۱۸

توضیح روش NCA ۱۹

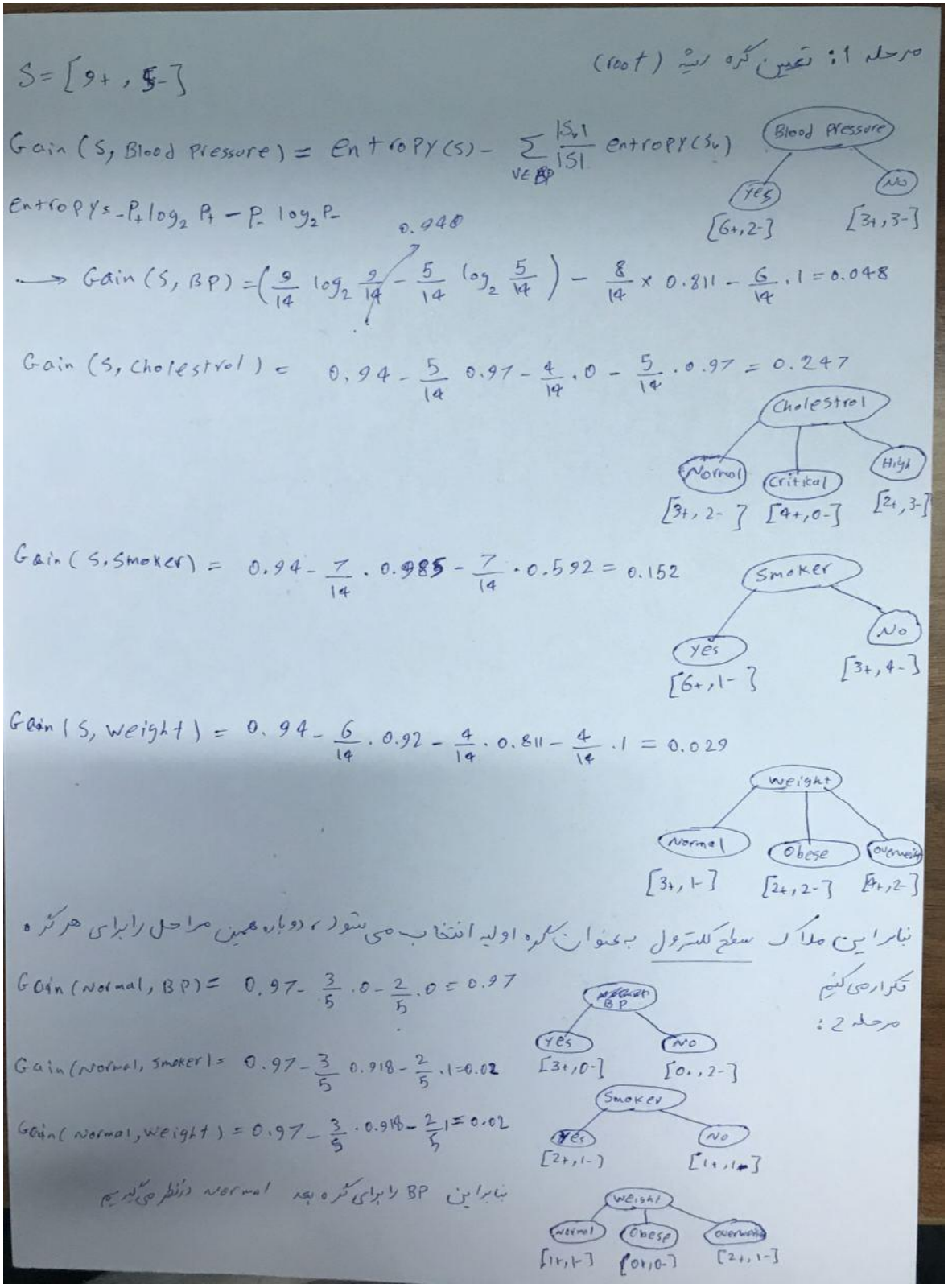
سوال ۱

ابتدا در قسمت الف به کمک الگوریتم ID3 درخت تصمیم [1]را بر حسب جدول داده شده آموزش می دهیم، سپس در قسمت ب به کمک درخت ساخته شده در قسمت الف داده های جدول تست را پیش بینی میکنیم و با نتایج اصلی مقایسه میکنیم تا دقت و ماتریس آشفتگی را محاسبه کنیم. سپس در قسمت ج دلیل مشکل بیش‌برازش برای درخت های تصمیم و دو روش برای جلوگیری از این مشکل ارائه می دهیم.

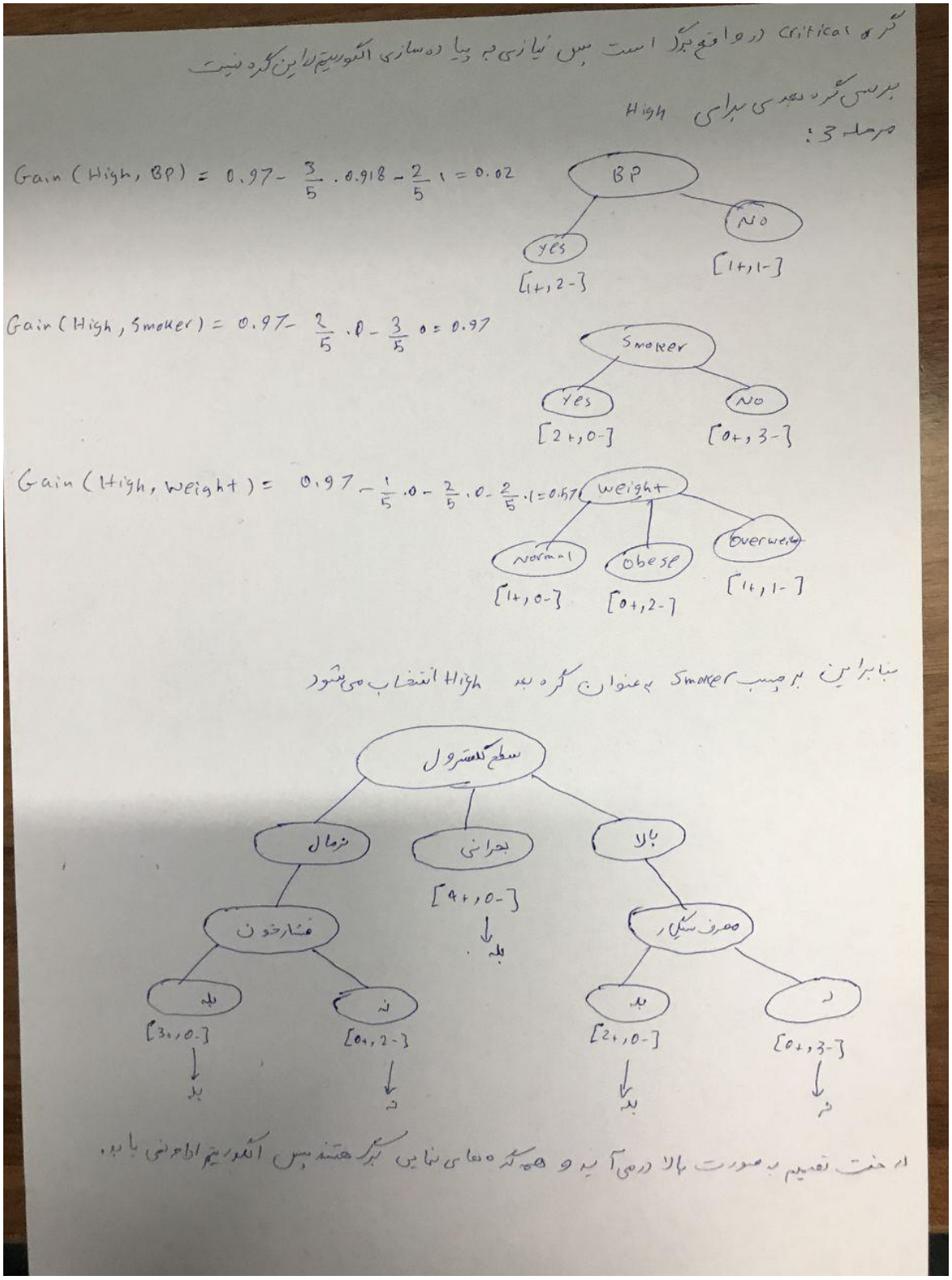
❋ عکس های قرار داده شده در گزارش مربوط به سوال اول به صورت جداگانه در فایل ارسالی قرار گرفته اند تا در صورت افت کیفیت در درسترس باشند.

[1] Decision Tree

الف:

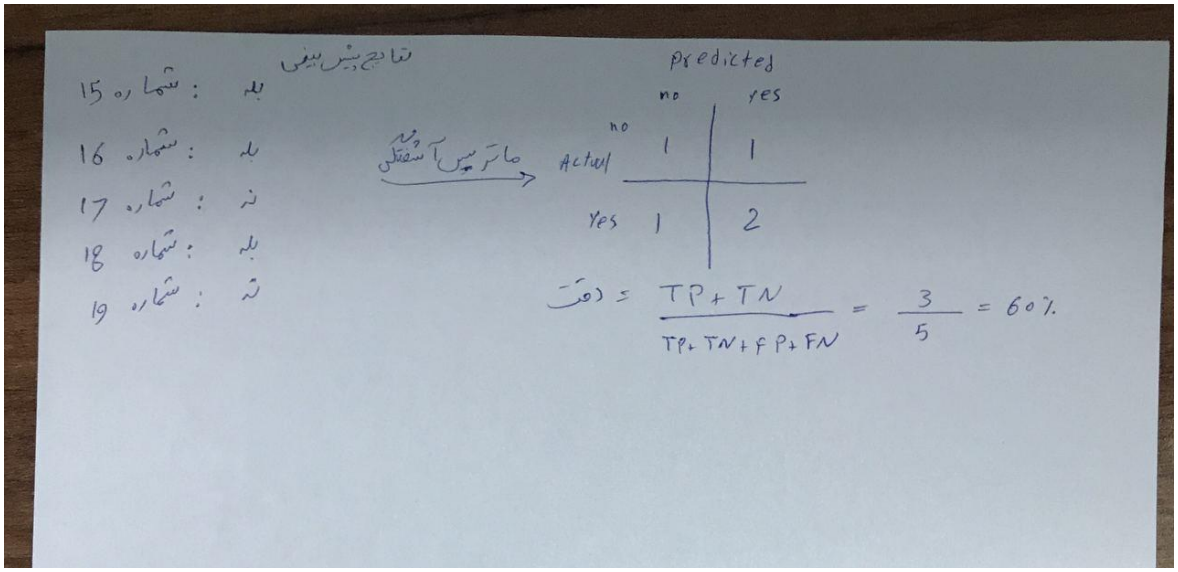


شکل ۱-۱– خروجی الگوریتم آموزش درخت تصمیم

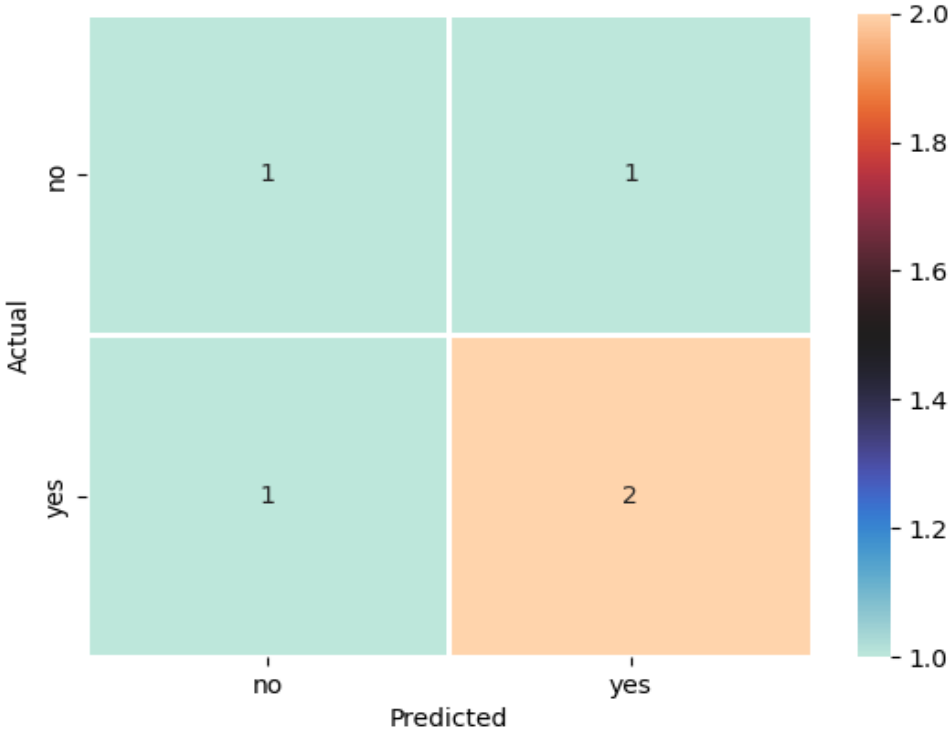


شکل۱-۲ – خروجی الگوریتم آموزش درخت تصمیم

ب:



شکل ۱-۳– خروجی الگوریتم آموزش درخت تصمیم



شکل ۱-۴ : ماتریس آشفتگی[1] مربوط به پیشبینی داده های تست

[1] Confusion Matrix

همانطور که میبینیم مدل آموزش داده شده ما سه داده اول جدول را درست پیش بینی کرد اما دو داده دیگر را اشتباه پیش بینی کرد، دلیل این پیش بینی اشتباه این است که در داده های آموزش این کیس از داده وجود نداشته و این دو داده به نوعی داده نویزی محسوب می شوند چون نیاز به برچسب[1] های جدیدی در درخت تصمیم دارند. این مشکل با روش هایی مانند جنگلهای تصادفی تا حدودی بهبود میابد.

ج:

درخت تصمیم در برابر بیش برازش[2] مقاوم نیست، الگوریتم درخت تصمیم(ID3) تا وقتی که تمام داده های آموزش به گره برگ [3] برسند اجرا می شود. با توجه به این الگوریتم، وجود داده های نویزی ممکن است باعث ایجاد برچسب های جدید و بزرگ شدن درخت و در نتیجه افزایش خطا شود.

دو روش برای جلوگیری از بیش برازش:

۱) از رشد درخت با یک معیار ریاضی جلوگیری کنیم: مثلا معیار انتخابی می تواند $minimize\left[size(tree) + \alpha * size\left(missclassification(tree)\right)\right]$ انتخاب شود که باعث می شود اگر اضافه کردن برچسبی باعث کاهش خطا تا حد قابل قبولی نشود آن برچسب را به درخت اضافه نکند.

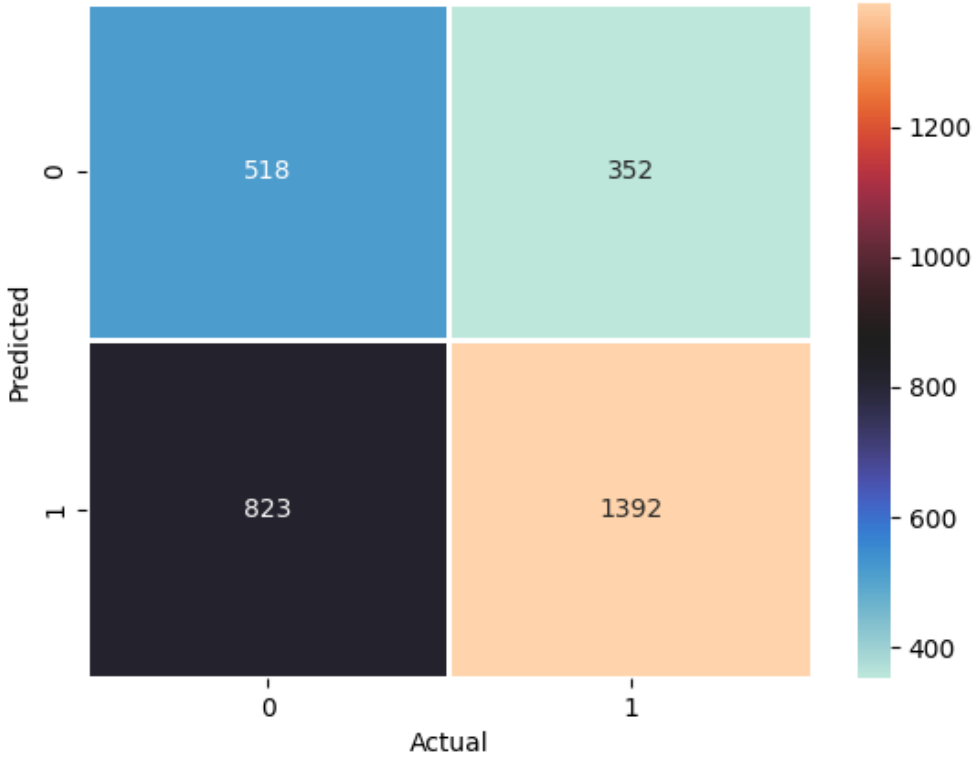
۲) ساخت کل درخت و سپس هرس کردن[4] آن: وقتی تعداد داده های ما به اندازه کافی زیاد باشد می توان درصدی از داده ها را برای اعتبارسنجی[5] استفاده کرد و برسی کرد که اگر هرس کردن درخت باعث افزایش یا ثابت ماندن دقت می شود، درخت را هرس کرد.

[1] Label
[2] Overfitting
[3] Leaf node
[4] Post-pruning
[5] Validation

سوال ۲

در این سوال، به کمک الگوریتم ID3 درخت تصمیم را با عمق های مختلف با دادگان داده شده آموزش می دهیم، و دقت آن ها را با استفاده از ماتریس آشفتگی برسی میکنیم. سپس الگوریتم جنگل های تصادفی را روی دادگان پیاده سازی میکنیم و با قسمت قبل مقایسه میکنیم، و در قسمت آخر هم با استفاده از کتابخانه *Scikit-Learn* الگوریتم جنگل های تصادفی را پیاده سازی میکنیم. برای دقیق تر شدن مدل، ابتدا دادگان را بر میزنیم[1] (در همه قسمت ها از random_state = 0 استفاده شده است)

الف:

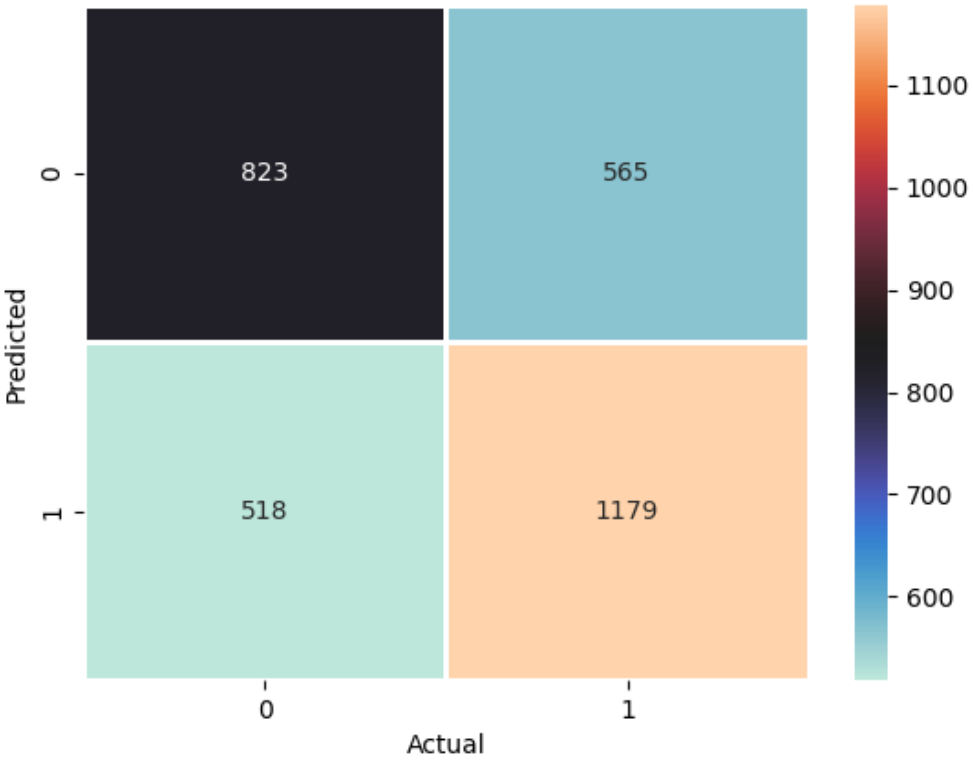


شکل۲-۱ – ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۲

```
 Actual       0     1
Predicted
0           518   352
1           823  1392
Accuracy of prediction is equal to:  61.912479740680716 %
```

شکل ۲-۲– دقت مدل و ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۲

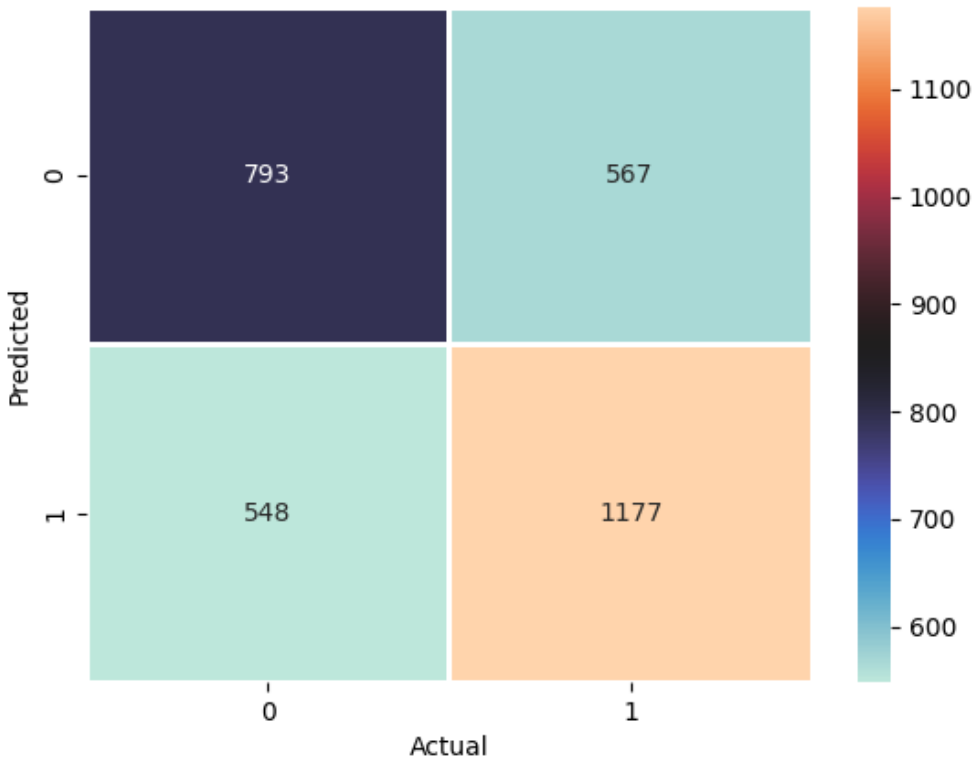
[1] Shuffle



شکل ۲-۳– ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۳

```
 Actual        0     1
Predicted
0            823   565
1            518  1179
Accuracy of prediction is equal to:  64.89465153970826 %
```

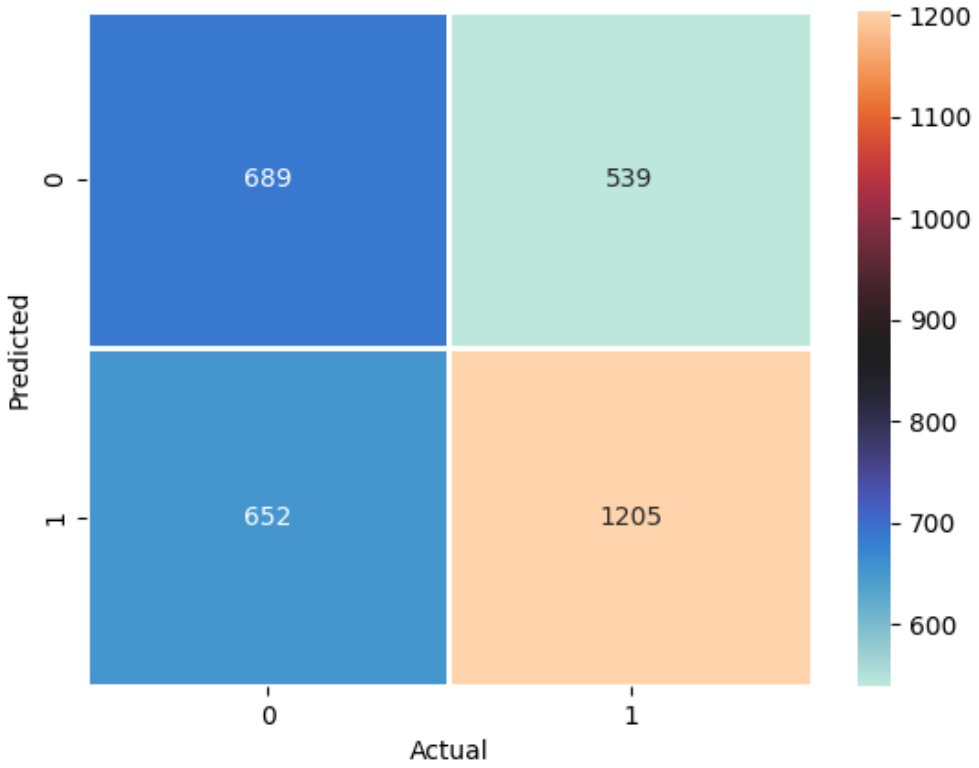
شکل ۲-۴– دقت مدل و ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۳



شکل ۲-۵- ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۴

```
 Actual       0     1
Predicted
0           793   567
1           548  1177
Accuracy of prediction is equal to:  63.85737439222042 %
```

شکل ۲-۶- دقت مدل ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۴



شکل ۲-۷– ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۵

```
 Actual       0     1
Predicted
0           689   539
1           652  1205
Accuracy of prediction is equal to:  61.39384116693679 %
```

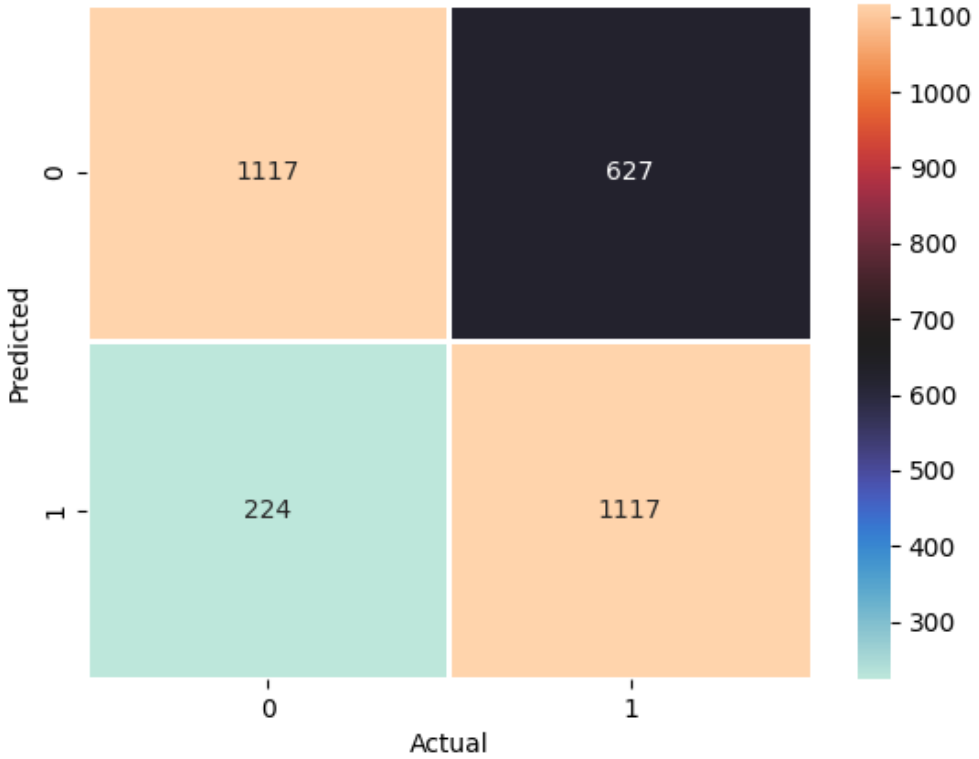
شکل۲-۸ –دقت مدل و ماتریس آشفتگی حاصل از پیش بینی داده های تست به ازای عمق ۵

با مشاهده ماتریس های آشفتگی و دقت های مدل به ازای عمق های مختلف درخت تصمیم، متوجه میشویم که در افزایش عمق درخت از ۲ به ۳ دقت مدل در پیش بینی داده های تست اندکی افزایش میابد و بعد از آن با افزایش عمق درخت دقت مدل کاهش میابد، نتیجه میگیریم بهترین عمق درخت برای این مسئله برابر ۳ است، با عمق ۲ و ۱ این مدل دچار مشکل کم‌برازش[1] می شود و با عمق بیشتر از ۳ دچار مشکل بیش‌برازش[2] می شود.

[1] Underfitting
[2] Overfitting

ب:

در این قسمت الگوریتم جنگل های تصادفی[1] را پیاده سازی کرده و ۱۰۰ درخت تصادفی آموزش می دهیم و سپس برچسب داده های تست را به کمک این درختها پیش بینی میکنیم و در هر مرحله پاسخی که در اکثریت درختها یکسان بود را به عنوان جواب انتخاب میکنیم.



شکل۲-۹ –دقت مدل و ماتریس آشفتگی حاصل از پیش بینی داده های تست به کمک جنگل های تصادفی به ازای عمق ۳

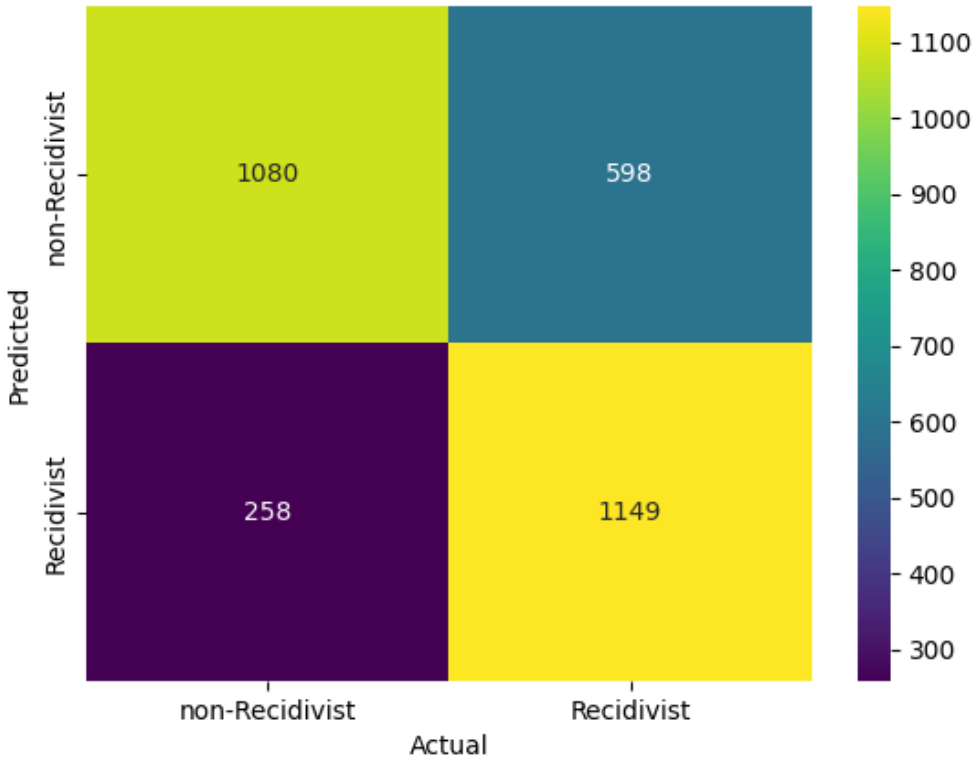
```
 Actual        0     1
Predicted
0           1117   627
1            224  1117
Accuracy of prediction is equal to:  72.41491085899514 %
```

شکل۲-۱۰ –دقت مدل و ماتریس آشفتگی حاصل از پیش بینی داده های تست به کمک جنگل های تصادفی به ازای عمق ۳

[1] Random forest

با مقایسه با قسمت قبل متوجه می شویم دقت حدود 6% افزایش پیدا کرده است به دلیل اینکه در الگوریتم جنگل های تصادفی با میانگین گیری پیش بینی بر اساس درخت های غیر هم بسته[1] خطای مدل کاهش میابد، به عبارتی مشکل بیش‌برازش کمتر می شود. البته در این دادگان بین برخی از ویژگی ها و برچسب آن داده همبستگی[2] خیلی کمی وجود دارد که این باعث می شود اگر تعداد درخت ها در جنگل تصادفی ما کم باشد دقت ما ثابت بماند یا حتی کاهش یابد.

ج:



شکل ۲-۱۱ - ماتریس آشفتگی مربوط به پیشبینی داده های تست

Predicted	0	1
Actual		
0	1080	598
1	258	1149

Accuracy: 72.25283630470017

شکل ۲-۱۲ - ماتریس آشفتگی مربوط به پیشبینی داده های تست

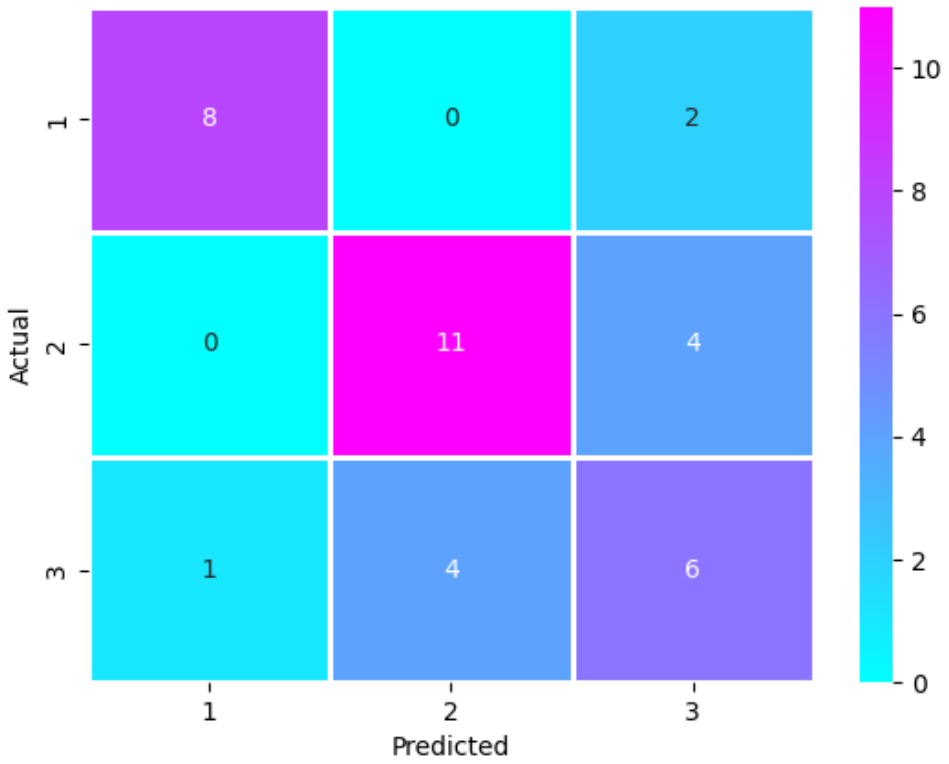
[1] Uncorrelated
[2] Correlation

دقت روش این قسمت تقریبا با دقت روش قسمت ب برابر است. زیرا الگوریتم پیاده سازی شده یکسان است.

سوال ۳

در قسمت اول طبق خواسته سوال داده ها را به دو دسته تست و آموزش تقسیم میکنیم، سپس به ازای K=3,5,7,9 ماتریس آشفتگی و دقت روش را محاسبه می کنیم، در این سوالات برای محاسبه فاصله از معیار فاصله اقلیدسی[1] استفاده کردیم. در قسمت ب به کمک یادگیری بر اساس معیار [2] به دو روش LMNN و NCA دادگان را آماده کرده و به ازای تعداد همسایه مختلف برچسب داده های تست را پیش بینی میکنیم و مقایسه میکنیم.

الف:

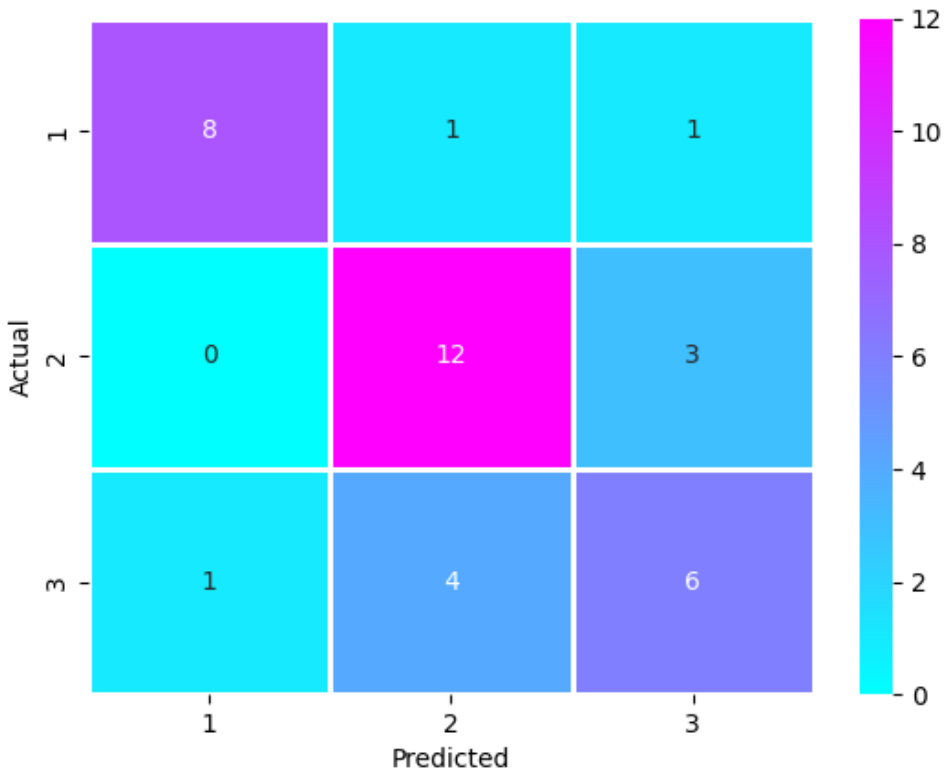


شکل۳-۱ – ماتریس آشفتگی به ازای k=5

[1] Euclidian Distance
[2] Metric Learning

```
Predicted  1   2  3
Actual
1          8   0  2
2          0  11  4
3          1   4  6
Accuracy of prediction is equal to:  69.44444444444444 %
```

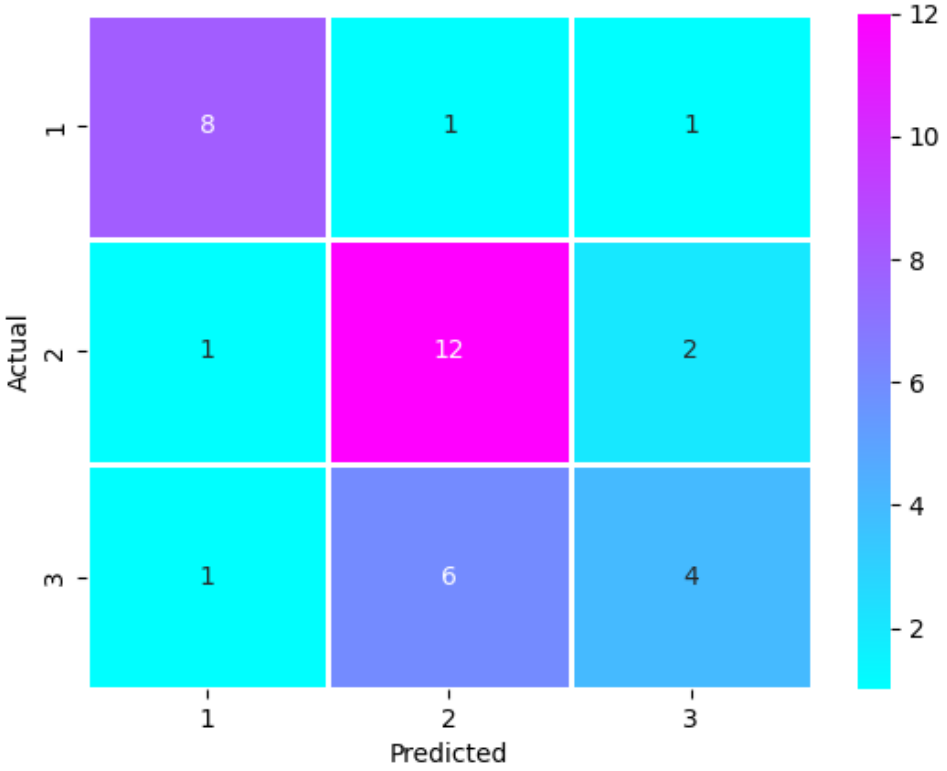
شکل ۳-۲–دقت و ماتریس آشفتگی به ازای k=5



شکل ۳-۳–دقت و ماتریس آشفتگی به ازای k=3

```
Predicted  1   2  3
Actual
1          8   1  1
2          0  12  3
3          1   4  6
Accuracy of prediction is equal to:  72.22222222222221 %
```

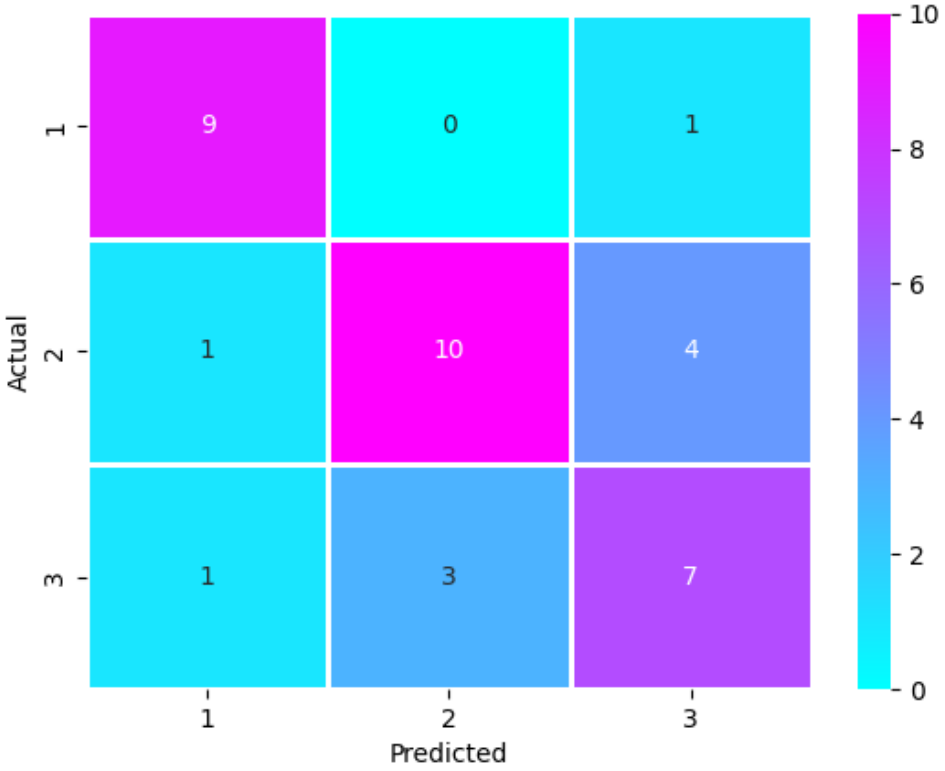
شکل ۳-۴–دقت و ماتریس آشفتگی به ازای k=3



شکل ۳-۵ –دقت و ماتریس آشفتگی به ازای k=7

```
Predicted  1   2  3
Actual
1          8   1  1
2          1  12  2
3          1   6  4
Accuracy of prediction is equal to:  66.66666666666666 %
```

شکل۳-۶ –دقت و ماتریس آشفتگی به ازای k=7



شکل۳-۷ – ماتریس آشفتگی به ازای k=9

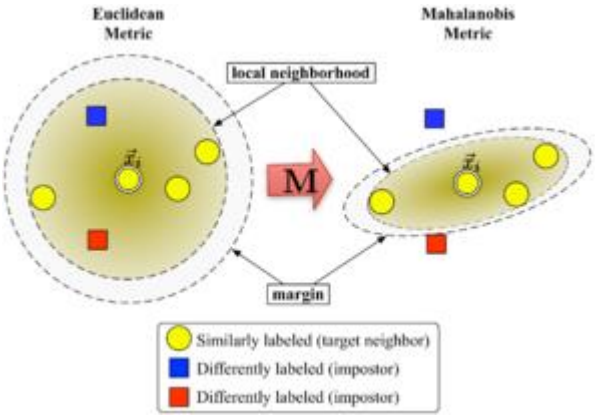
```
Predicted  1   2  3
Actual
1          9   0  1
2          1  10  4
3          1   3  7
Accuracy of prediction is equal to:  72.22222222222221 %
```

شکل۳-۸ –دقت و ماتریس آشفتگی به ازای k=9

ب:

توضیح روش LMNN:

حاشیه بزرگ همسایه های نزدیک یا LMNN روشی است که معیار سراسری[1] را به صورت نظارت شده[2] می آموزد تا دقت استفاده از الگوریتم KNN را افزایش دهد. شهود اصلی پشت این روش یادگیری یک معیار است که تحت آن همه ی داده های نمونه در مجموعه آموزش با حداقل k نمونه که برچسب آنها یکسان است محاصره شده باشد. شکل ۳-۹ شهودی از نتیجه این روش به ما می دهد.



شکل ۳-۹– تصویر شماتیک LMNN

همسایه های هدف[3] داده هایی هستند که باید تحت معیار یادگیری شده نزدیک ترین همسایه ها به هم شوند. در حالی که **فریب دهنده**[4]ها داده هایی هستند که در بر چسب آن ها با برچسب همسایه های هدف متفاوت است ولی تحت فاصله اقلیدسی در یک محدوده قرار می گیرند، هدف اجرای الگوریتم LMNN کمینه کردن تعداد نمونه های فریب دهنده در مجموعه آزمایشی است.

هدف اول در الگوریتم، مینیمم کردن فاصله داده نمونه و همسایه های هدف آن میباشد:

$$\sum_{i,j \in N_i} d(\vec{x}_i, \vec{x}_j).$$

شکل ۳-۱۰– فاصله داده نمونه از همسایه های هدف

هدف دوم، جریمه کردن فاصله به فریب دهنده هایی می باشد که فاصله آنها کمتر از همسایه های هدف $\vec{x}_j$ کمتر از یک واحد است.(نتیجه این کار دور کردن داده ها با برچسب متفاوت از نمونه داده $\vec{x}_i$ است)

[1] Global Metric
[2] Supervised
[3] Target Neighbors
[4] Imposters

$$\sum_{i,j\in N_i,l,y_l\neq y_i} [d(\vec{x}_i,\vec{x}_j)+1-d(\vec{x}_i,\vec{x}_l)]_+$$

شکل ۳-۱۱– تابعی که باید طبق هدف دوم مینیمم شود

با ترکیب دو هدف بالا مسئله بهینه سازی ما به صورت زیر در می آید:

$$\min_{\mathbf{M}} \sum_{i,j\in N_i} d(\vec{x}_i,\vec{x}_j)+\lambda\sum_{i,j,l}\xi_{ijl}$$
$$\forall_{i,j\in N_i,l,y_l\neq y_i}$$
$$d(\vec{x}_i,\vec{x}_j)+1-d(\vec{x}_i,\vec{x}_l)\leq \xi_{ijl}$$
$$\xi_{ijl}\geq 0$$
$$\mathbf{M}\succeq 0$$

شکل ۳-۱۲– مسئله بهینه سازی در الگوریتم LMNN

که فراپارامتر λ[1] به وسیله اعتبارسنجی متقابل[۲] تعیین می شود

منبع: https://en.wikipedia.org/wiki/Large_margin_nearest_neighbor

توضیح روش NCA

تحلیل مؤلفه های همسایگی یا NCA در واقع یک روش یادگیری نظارت شده برای طبقه بندی داده های چند متغیره به کلاس های مجزا با توجه به معیار فاصله داده شده بر روی داده ها است.

هدف این روش، یادگیری معیار فاصله با یافتن یک تبدیل خطی از داده‌های ورودی است، به طوری که میانگین اعتبارسنجی متقابل در فضای تبدیل‌شده به حداکثر برسد.

برای پیدا کردن ماتریس تبدیل A، یک تابع هدف که دقت طبقه بندی در فضای تبدیل شده را توصیف کند، تعریف میکنیم و سعی در پیدا کردن A^* میکنیم به طوری که تابع هدف بیشینه شود

$$A^* = \mathrm{argmax}_A f(A)$$

شکل ۳-۱۳– تابع هدف در الگوریتم NCA

در این روش کل داده های موجود را که توسط تبدیل نگاشت شده اند را به عنوان نزدیک ترین همسایه تصادفی[۳] در نظر می گیریم، سپس یک تابع بیشینه هموار[۴] از مربع فاصله اقلیدسی نقطه مورد نظر از تمام نقاط دیگر در فضای تبدیل به صورت زیر تعریف میکنیم:

[1] hyperparameter
[2] Cross-validation
[3] stochastic nearest neighbors
[4] SoftMax

$$p_{ij} = \begin{cases} \frac{e^{-\|Ax_i - Ax_j\|^2}}{\sum_k e^{-\|Ax_i - Ax_k\|^2}}, & \text{if} j \neq i \\ 0, & \text{if} j = i \end{cases}$$

شکل ۳-۱۴

که p_{ij} احتمال طبقه بندی همسایه j ام نقطه i است، حال احتمال درست طبقه بندی کردن نقطه i برابر احتمال طبقه بندی همه همسایه های آن نقطه با یک برچسب است:

$$p_i = \sum_{j \in C_i} p_{ij}$$

شکل ۳-۱۵

بنابراین تابع هدف ما به صورت زیر تعریف می شود:

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i$$

شکل ۳-۱۶

در نتیجه برای بهینه سازی تابع هدف گرادیان آن را محاسبه میکنیم:

$$\frac{\partial f}{\partial A} = -2A \sum_i \sum_{j \in C_i} p_{ij} \left(x_{ij} x_{ij}^T - \sum_k p_{ik} x_{ik} x_{ik}^T \right)$$

$$= 2A \sum_i \left(p_i \sum_k p_{ik} x_{ik} x_{ik}^T - \sum_{j \in C_i} p_{ij} x_{ij} x_{ij}^T \right)$$

شکل۳-۱۷ – مسئله بهینه سازی در الگوریتم NCA

که در آن $x_{ij} = x_i - x_j$

با حل معادله فوق ماتریس بهینه A به دست می آید.

منبع: https://en.wikipedia.org/wiki/Neighbourhood_components_analysis

در زیر نمونه ای از داده های آموزش پس از اعمال دو روش معیار LMNN و NCA آمده است.

جدول ۳-۱– بر چسب و ویژگی ها قبل از اعمال معیار یادگیری

	Class	1	2	3	4	5	6	7	8	9	10	11	12
0	1	13.74000	1.67000	2.25000	16.40000	118	2.60000	2.90000	0.21000	1.62000	5.85000	0.92000	3.20000
1	3	12.79000	2.67000	2.48000	22.00000	112	1.48000	1.36000	0.24000	1.26000	10.80000	0.48000	1.47000
2	2	12.37000	1.13000	2.16000	19.00000	87	3.50000	3.10000	0.19000	1.87000	4.45000	1.22000	2.87000
3	1	13.56000	1.73000	2.46000	20.50000	116	2.96000	2.78000	0.20000	2.45000	6.25000	0.98000	3.03000
4	2	13.05000	5.80000	2.13000	21.50000	86	2.62000	2.65000	0.30000	2.01000	2.60000	0.73000	3.10000
5	2	11.56000	2.05000	3.23000	28.50000	119	3.18000	5.08000	0.47000	1.87000	6.00000	0.93000	3.69000
6	1	14.06000	2.15000	2.61000	17.60000	121	2.60000	2.51000	0.31000	1.25000	5.05000	1.06000	3.58000
7	3	12.36000	3.83000	2.38000	21.00000	88	2.30000	0.92000	0.50000	1.04000	7.65000	0.56000	1.58000
8	2	12.25000	1.73000	2.12000	19.00000	80	1.65000	2.03000	0.37000	1.63000	3.40000	1.00000	3.17000
9	2	12.08000	1.83000	2.32000	18.50000	81	1.60000	1.50000	0.52000	1.64000	2.40000	1.08000	2.27000
10	3	13.36000	2.56000	2.35000	20.00000	89	1.40000	0.50000	0.37000	0.64000	5.60000	0.70000	2.47000
11	3	13.88000	5.04000	2.23000	20.00000	80	0.98000	0.34000	0.40000	0.68000	4.90000	0.58000	1.33000
12	1	14.20000	1.76000	2.45000	15.20000	112	3.27000	3.39000	0.34000	1.97000	6.75000	1.05000	2.85000
13	2	12.37000	1.07000	2.10000	18.50000	88	3.52000	3.75000	0.24000	1.95000	4.50000	1.04000	2.77000

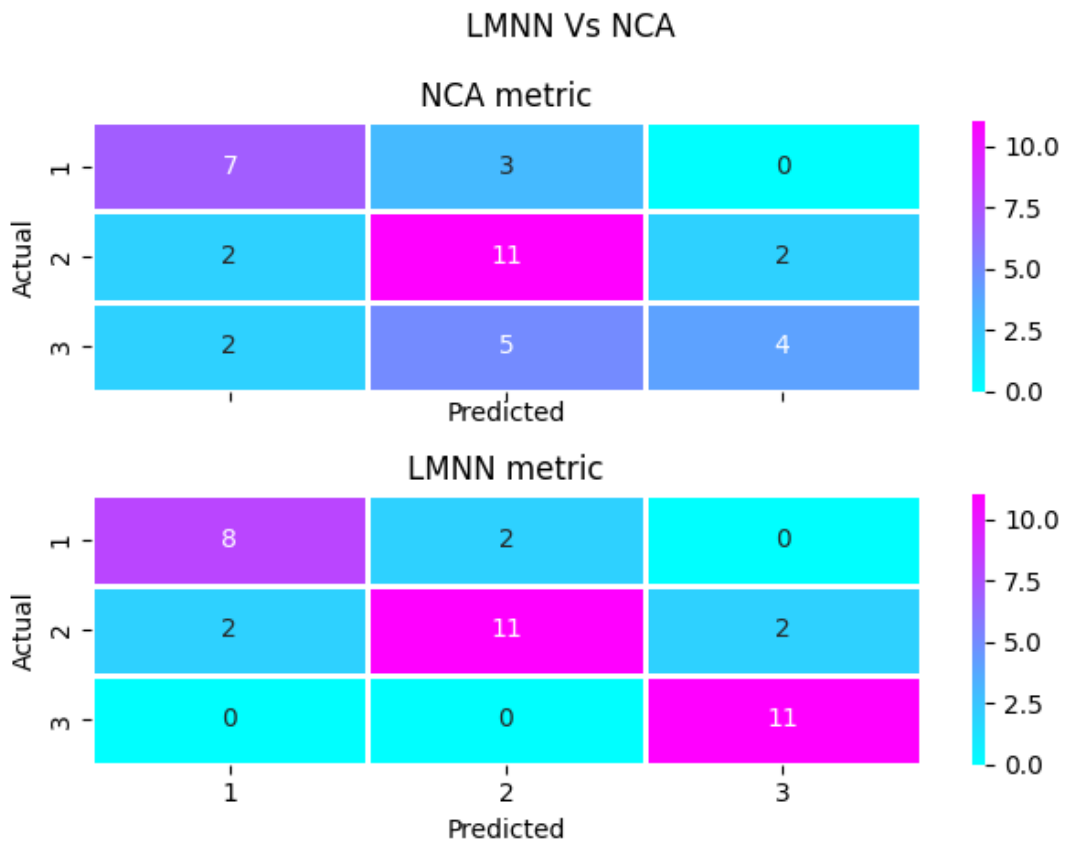
جدول ۳-۲– ویژگی ها بعد از اعمال معیار LMNN

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	9.04182	2.58709	11.04084	-3.33683	-0.00197	-0.14823	-1.69312	0.11626	1.55481	-5.00754	-0.81305	1.33624	0.20745
1	9.15455	2.02516	5.25609	-1.34619	0.08816	0.03648	-2.17106	-0.99751	-1.43793	-1.63672	-0.72989	0.78680	0.05555
2	9.40741	0.74857	4.50343	-1.30134	0.03158	2.02999	0.31893	-0.42005	-0.15138	-2.90322	0.54052	1.94252	0.03119
3	8.73316	2.49237	11.43382	-3.41403	-0.01261	-0.11039	-2.30281	0.13007	2.25105	-5.24098	-0.79655	0.82297	0.22396
4	10.11224	2.67033	3.96589	-1.29747	0.05352	1.49471	0.18947	-0.51855	-0.11047	-2.69270	0.10651	2.32379	0.03575
5	8.83269	0.94590	5.04903	-1.26501	0.04913	1.57200	1.23691	-0.46550	-1.42536	-3.01071	0.30400	2.41138	0.02296
6	8.76398	3.18024	13.55930	-4.03005	-0.00948	-0.59562	-2.63868	0.46448	2.18653	-5.90287	-0.99214	1.26302	0.26765
7	8.75588	2.42957	5.64851	-1.54303	0.07684	0.72484	-2.25250	-0.41304	-0.63989	-2.08869	-0.57485	0.72926	0.07473
8	9.03089	1.16028	5.37316	-1.53328	0.02303	0.32725	-0.61382	-0.12081	0.47250	-3.03470	0.14765	2.12092	0.06269
9	8.94076	1.09214	5.31316	-1.41513	0.02426	0.53843	-0.87873	-0.01767	0.61959	-3.01412	0.24586	1.43481	0.06163
10	9.03319	2.26539	8.20293	-2.28205	0.04749	-0.44877	-2.94090	-0.11615	0.47649	-3.37210	-0.77632	1.15542	0.13560
11	10.24657	2.67595	4.64167	-1.27084	0.07961	0.12945	-1.85732	-0.60270	-0.58030	-1.95993	-0.44644	0.94677	0.05201
12	8.46348	3.46767	15.13261	-4.64801	-0.04990	-0.73780	-2.39408	0.78072	3.33557	-6.36903	-1.23095	0.20329	0.32250
13	8.88659	1.19902	6.79891	-2.06822	0.00056	1.39033	0.24454	-0.04282	0.78741	-3.81868	0.05308	1.36771	0.09955

جدول ۳-۳ – ویژگی ها بعد از اعمال معیار NCA

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	-39.29081	-66.82158	-12.08224	42.34010	-253.44156	-6.44981	27.86824	-8.97715	-13.60456	114.64903	-7.79125	10.51816	15.82018
1	-2.45188	-16.28439	-3.02185	49.55585	-24.95414	-1.95721	11.77050	-4.87063	-2.69457	92.94408	-6.15185	3.97530	-13.4215
2	-2.87134	-17.80121	-2.89484	39.75160	-39.63347	0.39215	12.60528	-4.12068	-2.15196	69.12074	-4.05765	5.45346	-8.32439
3	-43.46836	-71.84141	-12.86763	45.67460	-278.91671	-6.80627	28.89860	-9.35066	-14.33603	117.09522	-7.94003	10.73042	19.87329
4	0.45328	-9.29365	-2.34040	42.45257	-24.45187	-0.01068	11.43423	-3.78107	-1.10714	65.38991	-4.47569	5.62637	-10.8326
5	-2.16231	-14.57199	-2.00346	57.27053	-10.23077	0.00429	15.35679	-4.65363	-1.47308	89.21481	-5.86344	6.33032	-16.8332
6	-53.94682	-85.90184	-15.29436	43.27186	-343.59515	-8.79429	33.08186	-10.52999	-18.64976	125.72480	-8.60054	12.69734	27.50366
7	-8.76903	-22.72639	-4.17756	42.14381	-75.82291	-1.85598	12.46817	-4.51645	-5.06181	79.00647	-5.26564	4.72069	-3.19608
8	-9.80733	-26.16767	-4.43498	37.63474	-85.86901	-2.48614	13.66472	-4.43995	-4.64320	68.49639	-4.30991	6.48232	-1.03338
9	-8.01388	-23.60899	-3.73906	37.51390	-73.11134	-2.22674	12.46338	-4.09616	-3.92434	66.37667	-4.11316	5.34146	-3.14677
10	-24.88808	-46.55467	-8.18400	40.08316	-180.19089	-5.38339	18.34555	-6.42364	-10.71780	88.24780	-5.98737	7.62394	10.58745
11	-1.60005	-14.13749	-2.83414	39.61925	-46.07796	-2.19662	9.59756	-3.79223	-3.58679	67.17742	-4.52584	3.84696	-6.63663
12	-64.96015	-100.82867	-18.01725	37.85857	-417.90625	-9.79525	37.61457	-11.41484	-21.67202	130.22524	-8.85776	13.20353	39.31579
13	-18.41909	-38.17708	-6.64403	38.37927	-134.63400	-1.98309	19.00716	-5.72459	-6.95656	80.28667	-5.12628	7.21501	4.51040

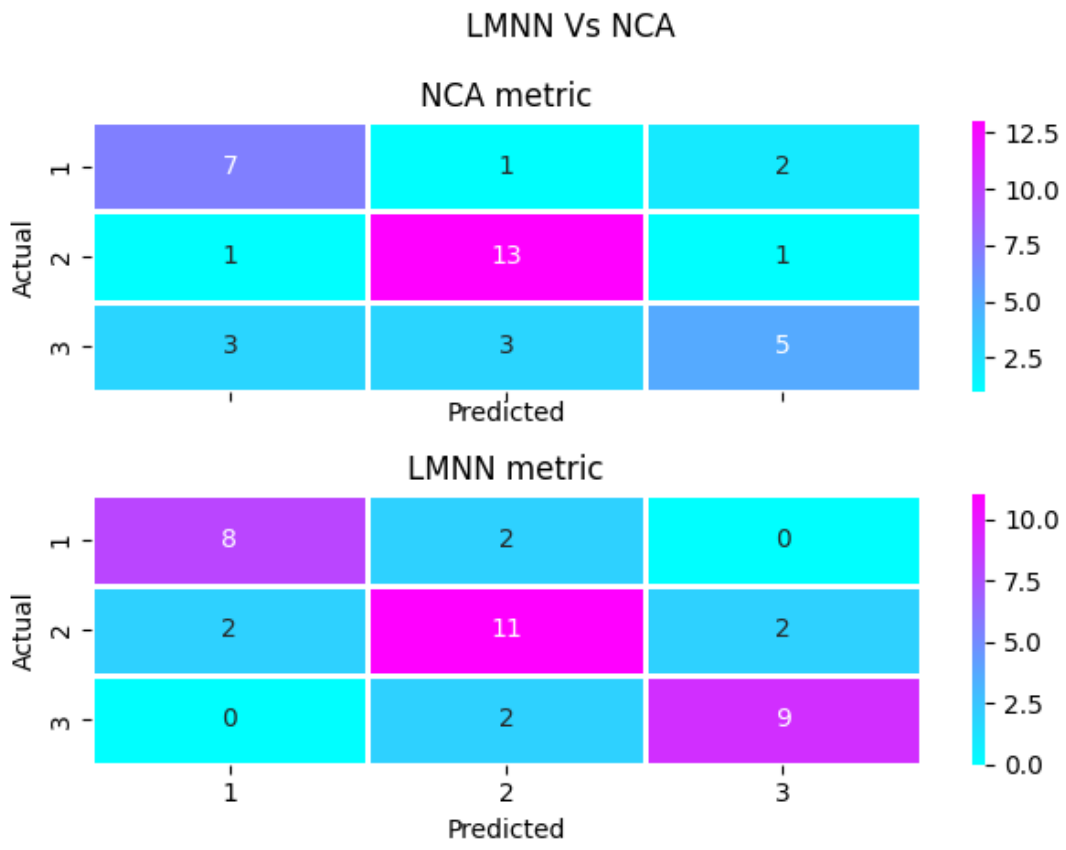
با مشاهده ویژگی ها و برچسب آنها میبینیم که اندازه داده های ویژگی متعلق به هر برچسب با بقیه بر چسب ها بیشتر فاصله گرفته و در نتیجه با پیاده سازی الگوریتم KNN فاصله بین بر چسب های غیر مشابه بیشتر میشود.



شکل۳-۱۸ – ماتریس آشفتگی بعد از اعمال دو معیار LMNN و NCA به ازای k=3

```
Accuracy of prediction with LMNN metric is equal to:  83.33333333333334 %
Accuracy of prediction with NCA metric is equal to:  61.111111111111114 %
```

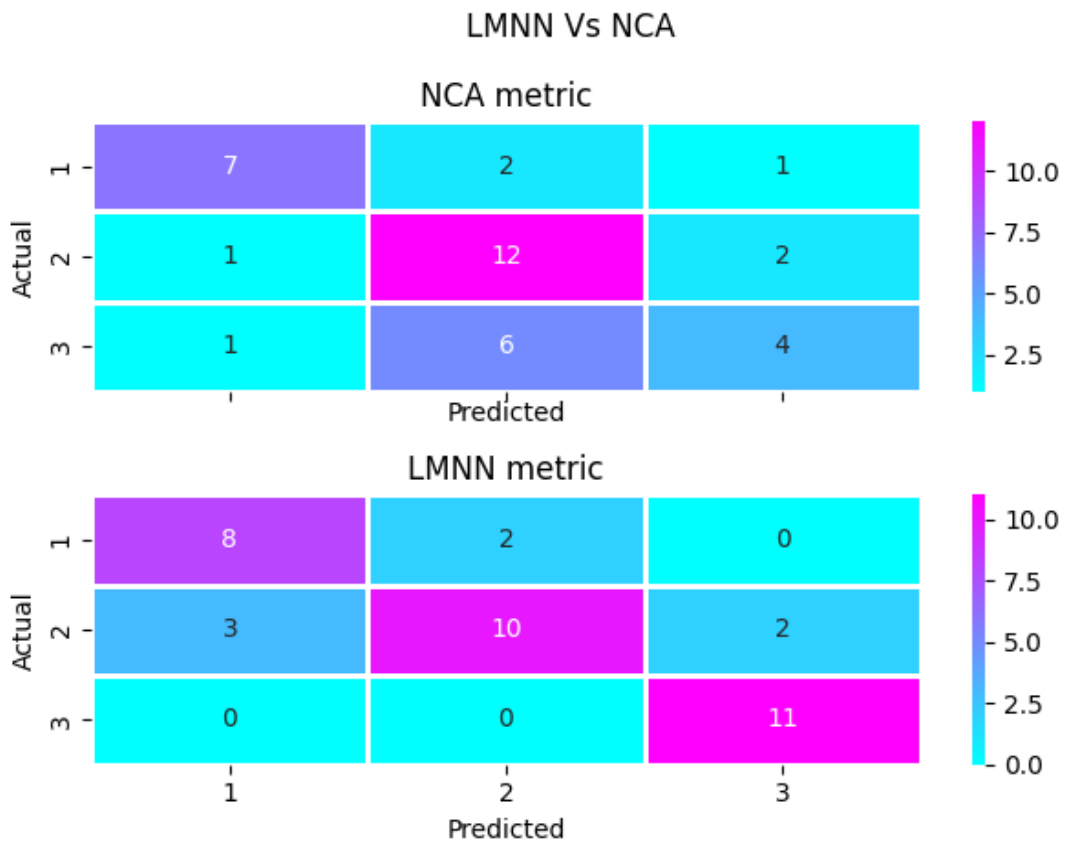
شکل۳-۱۹ – دقت پیش بینی پس از اعمال دو معیار LMNN و NCA به ازای k=3



شکل۳-۲۰ – ماتریس آشفتگی بعد از اعمال دو معیار LMNN و NCA به ازای k=5

```
Accuracy of prediction with LMNN metric is equal to:  77.77777777777779 %
Accuracy of prediction with NCA metric is equal to:  69.44444444444444 %
```

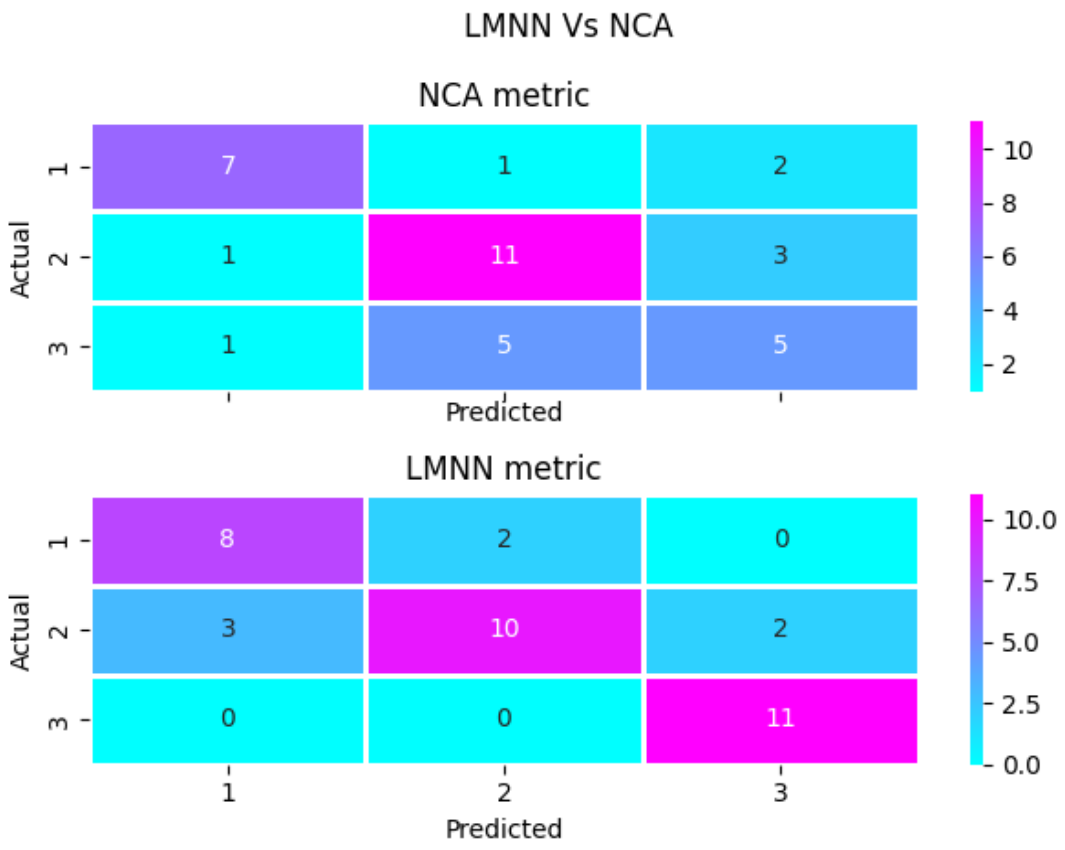
شکل ۳-۲۱– دقت پیش بینی پس از اعمال دو معیار LMNN و NCA به ازای k=5



شکل ۳-۲۲– ماتریس آشفتگی بعد از اعمال دو معیار LMNN و NCA به ازای k=7

```
Accuracy of prediction with LMNN metric is equal to:  80.55555555555556 %
Accuracy of prediction with NCA metric is equal to:  63.888888888888886 %
```

شکل۳-۲۳ – دقت پیش بینی پس از اعمال دو معیار LMNN و NCA به ازای k=7



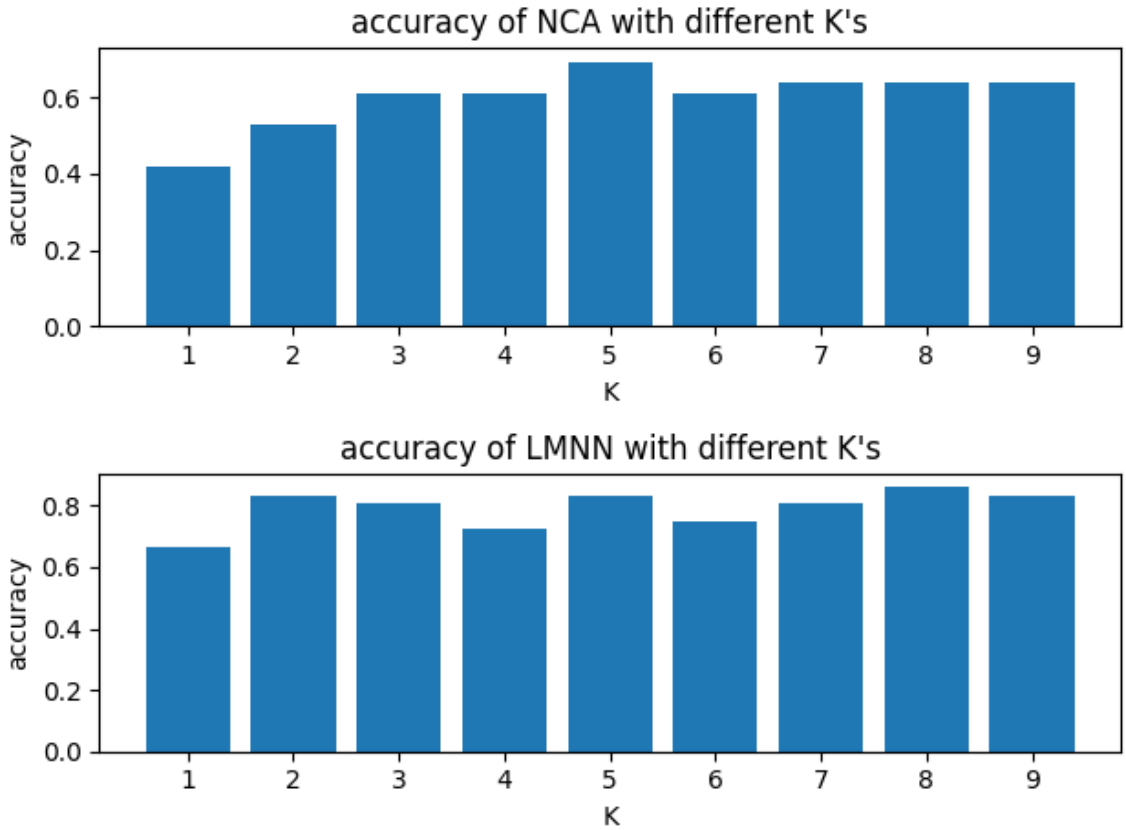
شکل۳-۲۴ – ماتریس آشفتگی بعد از اعمال دو معیار LMNN و NCA به ازای k=9

```
Accuracy of prediction with LMNN metric is equal to:  80.55555555555556 %
Accuracy of prediction with NCA metric is equal to:  63.888888888888886 %
```

شکل۳-۲۵ – دقت پیش بینی پس از اعمال دو معیار LMNN و NCA به ازای k=9

با مقایسه نتایج قسمت ب با الف میبینیم که پس از اعمال معیار LMNN دقت ما به ازای K های مختلف افزایش قابل توجهی داشته است، اما برای روش NCA دقت تغییر خاصی نداشته و در بعضی موارد اندکی کاهش یافته است.

به منظور برسی مناسب ترین K نمودار خطا بر حسب K های مختلف را رسم میکنیم



شکل ۳-۲۶– دقت پیش بینی پس از اعمال دو معیار LMNN و NCA به ازای K های مختلف

با توجه به نمودار بالا نتیجه می گیریم بهترین تعداد همسایه (K) برای مدل ما برابر K = 5 است البته در روش LMNN تعداد همسایه های K=2,8 هم دقت خوبی می دهند.