


## تمامی مراحل و اجراهای این بخش در فایل Q3 KNN.ipynb

(آ)

طبق مشخصات این دیتاست ۴۵۲ ردیف داده و ۲۸۰ ستون دارد. این داده ها در ۱۶ گروه طبقه بندی شده اند که البته فراوانی ۳ گروهشان صفر است پس عملاً با ۱۳ گروه تشخیصی سر و کار داریم که البته بالانس هم نیستند.

طبق اطلاعات دیتاست، مقدار های گم شده با '?' وارد شده اند. برای شناسایی این ها اول مقدار آنها را به NAN جایگزین میکنیم تا ابزار بیشتری برای دستکاری آنها در اختیار باشد. مشخص میشود که تنها ستون هایی که مقدار گم شده دارند شماره های ۱۱ تا ۱۵ هستند و تعداد مقادیرهای گم شده آنها هم به صورت زیر است:

```
val 11      8
val 12     22
val 13       1
val 14    376
val 15       1
```



چون داده ها بالانس نیستند قبل از حذف سطرهایی که مقدار گم شده دارند میخواهم مطمئن شوم که اطلاعاتی که از دست می دهم تعداد کلاسها را دستکاری نمکند. (مثلاً اگر بیشترشان از کلاس ۱ باشند که بیشترین فراوانی را دارد مشکل زیادی نیست ولی اگر مثلاً همه شان از یک کلاس باشند و فراوانی آن را به طرز قابل توجهی کاهش دهند حذف قابل قبول نیست.) می توان بررسی کرد اما به دلیل اینکه حذف داده در حالت کلی روش خوبی نیست، یک حالت دیگر مثلاً جاگذاری با میانگین را انتخاب میکنم و این ۵ ستون را هرجا مقدار خالی داشت با میانگین ستون پر میکنم.

(ب)

چون KNN به مقدار متغیرها وابسته و حساس است برای اینکه تاثیر فاکتور های پیشبینی را یکسان کنم باید داده ها را تبدیل کنم. از بین تبدیل ها از تبدیل مینیمم ماکسیمم استفاده میکنم. با توجه به آنچه در کتاب گفته شد و با توجه به اینکه داخل این ۲۷۹ ستون هم متغیر کیفی (تبدیل شده به عدد) داریم و هم مقادیر اندازه گیری شده ی پیوسته داریم، بهترین تبدیلی که در کلاس معرفی شده و میتوان روی همه ستون ها اعمال کرد استفاده از MinMaxScaler() است.

$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

اول دو تا دیتا ست یکی شامل فیچرها و یکی شامل تک ستون متغیر هدف از دیتاست اصلی جدا میکنیم. بعد با نسبت ۳۰ به ۷۰ این داده ها را به دو مجموعه تست و آموزش تقسیم می کنم.

حالا طبق خواسته ی سوال یک مدل کلاس KNN میسازم که k را برابر ۱ بگیرد و داده های آموزش را بهش میدهم. بعد از آموزش مدل، داده های تست را برای ارزیابی استفاده میکنم. (۱۲۶ تا داده تست دارم)

نتایج بدست آمده از این مدل برای داده های تست:

[ 59, 4, 0, 0, 3, 5, 1, 0, 0, 4, 0, 1]
[ 9, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0]
[ 1, 0, 5, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 4, 0, 0, 0, 1, 0, 1, 0, 0]
[ 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
[ 3, 1, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1]
[ 8, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 2]
[ 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

ماتریس confusion را که بررسی کنیم اول عناصر روی قطر را که نشان دهنده ی درستی ها بودند بررسی میکنم. عنصر روی درایه ii نشان میدهد که چند مورد از بین داده های تست توی کلاس i بوده و درست تشخیص داده شده. به طور شهودی اگر راجب این ماتریس نظر دهیم به نظر نمی رسد که مدل خوبی داشته باشیم. مثلا عملکرد این مدل برای کلاس نظیر سطر ۲ بد بوده. از بین ۱۰ موردی که متعلق به این کلاس بودند، ۹ تا را غلط به کلاس اولی نسبت داده. ۴ مورد از داده هایی که توی کلاس یک بودند را هم به اشتباه به این کلاس برده و تنها ۱ مورد درست دارد. یا مثلا توی کلاس نظیر سطر ۱۰ فقط ۵ تشخیص را درست داده و بقیه را اشتباه به کلاس یک برده و علاوه بر آن تعداد مجموعا ۸ مورد از کلاسهای دیگر را هم به غلط به این کلاس آورده. اگر کلی تر نگاه کنم، بیشتر آمار اشتباه ها توی سطر و ستون کلاس یک ثبت شده و مخصوصا توی ستون اول. یعنی تعداد زیادی از اشتباهات به دلیل این بوده که چیزی که توی یک کلاس دیگر هست را به کلاس یک برده. این ناشی از بالانس نبودن داده ها و فراوانی قابل توجه کلاس یک هست ولی از طرف دیگر عدد ۱ هم برای k با توجه به این خصوصیت داده ها انتخاب مناسبی نیست. برای بررسی بهتر معیارهایی که بدست آمده را ببینیم :

	precision	recall	f1-score	support
1	0.70	0.77	0.73	77
2	0.12	0.08	0.10	13
3	1.00	0.71	0.83	7
4	1.00	0.67	0.80	6
5	0.00	0.00	0.00	4
6	0.22	0.29	0.25	7
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	0
9	1.00	0.50	0.67	4
10	0.38	0.33	0.36	15
14	0.00	0.00	0.00	1
16	0.00	0.00	0.00	2
accuracy			0.57	136
macro avg	0.37	0.28	0.31	136
weighted avg	0.59	0.57	0.57	136

خب هنوز هم مدل خوبی نداریم! اول از accuracy شروع کنم. معیار چندان خوبی نیست اما نشان میدهد از کل داده هایی که قرار بوده کلاس بندی کند تنها ۴۵ درصد را درست انجام داده. اگر داده ها اینجا بالانس بود احتمالا همین عدد هم بدست نمیامد و مقدار خیلی کمتر بود و بزرگی این عدد به خاطر بزرگ بودن عدد فراوانی داده های کلاس اول است. معیار بعدی precision به طرز عجیبی برای چند تا از کلاسها عدد ۱ یعنی ماکزیمم مقدار گرفته در حالی که چند جای دیگر صفر شده. این عدم تعادل اول به نحوه ی محاسبه ی این معیار برمیگردد و دوم اینکه تعداد ۱۶ کلاس (۱۳ تا موجود) بر ای ۱۲۶ داده خیلی مقدار زیادی هست و مثلا اگر از یک کلاس یک فراوانی داشته باشیم و همان هم غلط تشخیص داده شود precision برابر صفر میشود. از طرف دیگر زیاد بودن این عدد در سطرهای دیگر به خاطر این است که ارزش بندی تشخیص های غلط به این کلاس را لحاظ میکند. مثلا اگر تمام ۱۲۶ مورد را به یک کلاس ببریم این معیار نتیجه کاملا نامطلوب نشان میدهد. در حالی که accuracy شاید خیلی تغییر نکند (در داده های فعلی). اگر مهم باشد که موردی به اشتباه به کلاس ۳ نرود در حالی که متعلق به این کلاس نیست این مدل نتیجه خوبی دارد. اما اگر همین اهمیت را برای کلاس ۶ داشته باشیم مدل کاملا نامطلوب است. در مقابل معیار recall هست که تعداد درستی های تشخیص را از بین تمام مقدارهای واقعی میسنجد. اگر مهم باشد تمام موارد کلاس ۱ را درست تشخیص دهیم، این معیار مناسب است (مثلا اینجا اگر معادل کلاس ۱ وضعیت اورژانسی بیمار باشد) اگر همین اهمیت برای کلاسهای ۱۶ و ۶ یا ... که مقدار کمی دارند داشته باشیم، مدل اصلا خوب نیست. (چون دقیقا نمیدانم کدام کلاس وضعیت چقدر خوب یا بد دارد پس فقط در حالت کلی و بدون ملاحظات پزشکی معیارها را تفسیر میکنم) حالا اگر بخواهم نتیجه دو ستون قبل را ادغام کنم با معیار f1-score نتیجه میشود که باز هم مدل برای کلاسهای مختلف فراز و نشیب زیادی دارد. و به طور متوسط عملکردش ضعیف است. حالا نتایج داده های آموزش را ببینیم :

همان طور که از  $k=1$  انتظار میرفت یک مدل با تمام معیارها در سطح عالی برای داده های آموزشی. روی ماتریس اگر نگاه کنیم حتی یک مورد اشتباه کلاس بندی وجود ندارد و البته به همین دلیل تمام اعداد برای تمام معیارها هم برابر ۱ شده.

```

[168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 9, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 18, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 35, 0, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0]
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20]

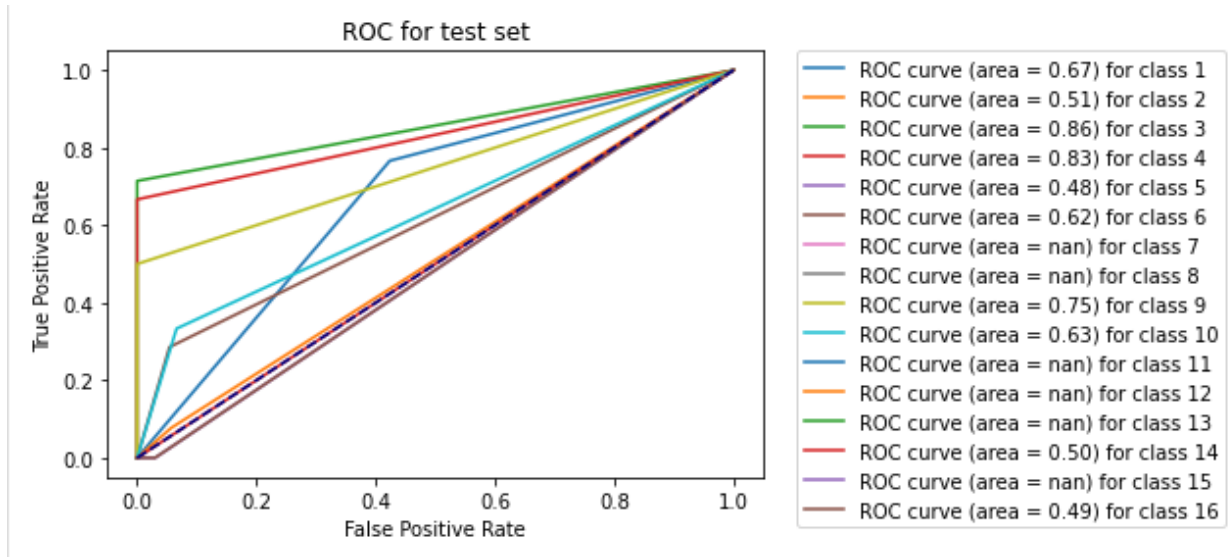
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	168
2	1.00	1.00	1.00	31
3	1.00	1.00	1.00	8
4	1.00	1.00	1.00	9
5	1.00	1.00	1.00	9
6	1.00	1.00	1.00	18
7	1.00	1.00	1.00	3
8	1.00	1.00	1.00	2
9	1.00	1.00	1.00	5
10	1.00	1.00	1.00	35
14	1.00	1.00	1.00	3
15	1.00	1.00	1.00	5
16	1.00	1.00	1.00	20
accuracy			1.00	316
macro avg	1.00	1.00	1.00	316
weighted avg	1.00	1.00	1.00	316

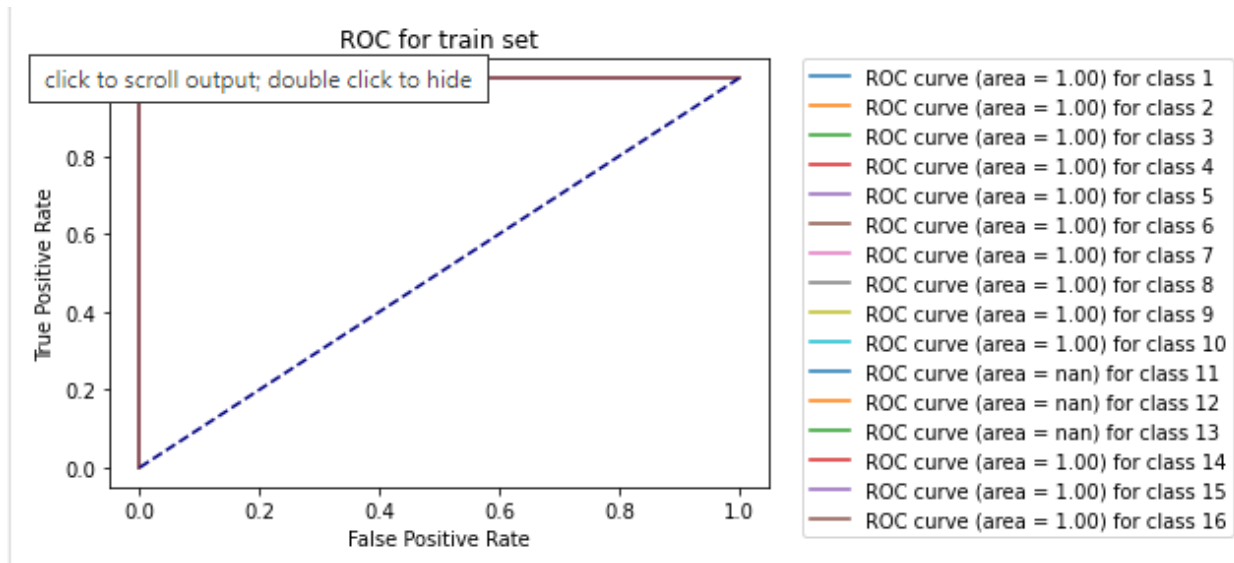
اگر این را با نتیجه داده های تست مقایسه کنیم واضح است که مدل overfit شده. هرچقدر  $k$  را افزایش دهیم مدل پیچیدگی اش کمتر میشود پس به ازای افزایش  $k$  میتوان مشکل بیش برازش را حل کرد تا با قبول یک مقدار خطا روی قسمت آموزش، بتواند روی داده های تست نتیجه خوبی داشته باشد. (bias variance tradeoff)

در آخر بررسی نمودار ROC که من برای راحتی مقایسه همه ی ۱۶ کلاس را روی یک نمودار رسم کردم.

هرچه مساحت زیر منحنی نظیر یک کلاس عدد بزرگتری باشد (نزدیک به یک) یعنی مدل بهتری داریم برای اینکه تعیین کنیم داده ای متعلق به آن کلاس هست یا نه. اینجا مثلاً برای داده های تست برای کلاس ۳ مدل بهتری بوده.



در مورد نمودار تست تمام منحنی ها روی هم افتاده است و یک شکل دارد. (به جز کلاسهای که فراوانی شان در دیتاست صفر هست) و همگی اینها مقدار برابر یک (مساحت زیر نمودار) دارند که خوب در تقابل داده های تست به وضوح بیش برآزش را نشان می دهد.



حالا همین گزارش های بدست آمده برای مدل KNN بازای ۳۰ تا همسایه :

ماتریس داده های تست :

از ماتریس ها معلوم میشود که مدل اینطوری کار میکند که کل تشخیص ها را به کلاس ۱ میبرد! و به دلیل بالانس نبودن داده های تست (نباید هم بالانس باشد!) به سمت کلاس ۱ accuracy بدتری بدست نیامده و فرقی با حالت قبل از این نظر ندارد!!

[76,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[17,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 7,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 4,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 2,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 7,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 1,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 1,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[16,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 5,	0,	0,	0,	0,	0,	0,	0,	0,	0]

ماتریس داده های آموزش:

[169,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 27,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 8,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 11,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 11,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 18,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 2,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 2,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 8,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 34,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 4,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 5,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]
[ 17,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0]

از انجایی که مدل کلا فقط یک کلاس طبقه بندی گرفته و همه را به آن ارجاع داده فقط نتایج برای کلاس ۱ داریم چون فقط عدد تشخیص این کلاس مقدار غیر صفر برای صورت کسر معیار دارد (عددهای روس قطر) . به اضافه این واقعیت که داده های این کلاس داخل داده های تست فراوان اند مدل به ظاهر نتیجه خوبی برای کلاس اول گرفته. اما در حقیقت مدل چنان بد است که اصلا کار طبقه بندی انجام نمیدهد! درست است که نسبت به حالت قبلی اختلاف خطاها کمتر شد اما اینجا پیچیدگی مدل انقدر کم است که حتی روی داده های آموزش هم خطای زیادی دارد و به طور واضحی مدل underfit است. به یک مقدار k خیلی کمتر نیاز داریم.

داده های تست :

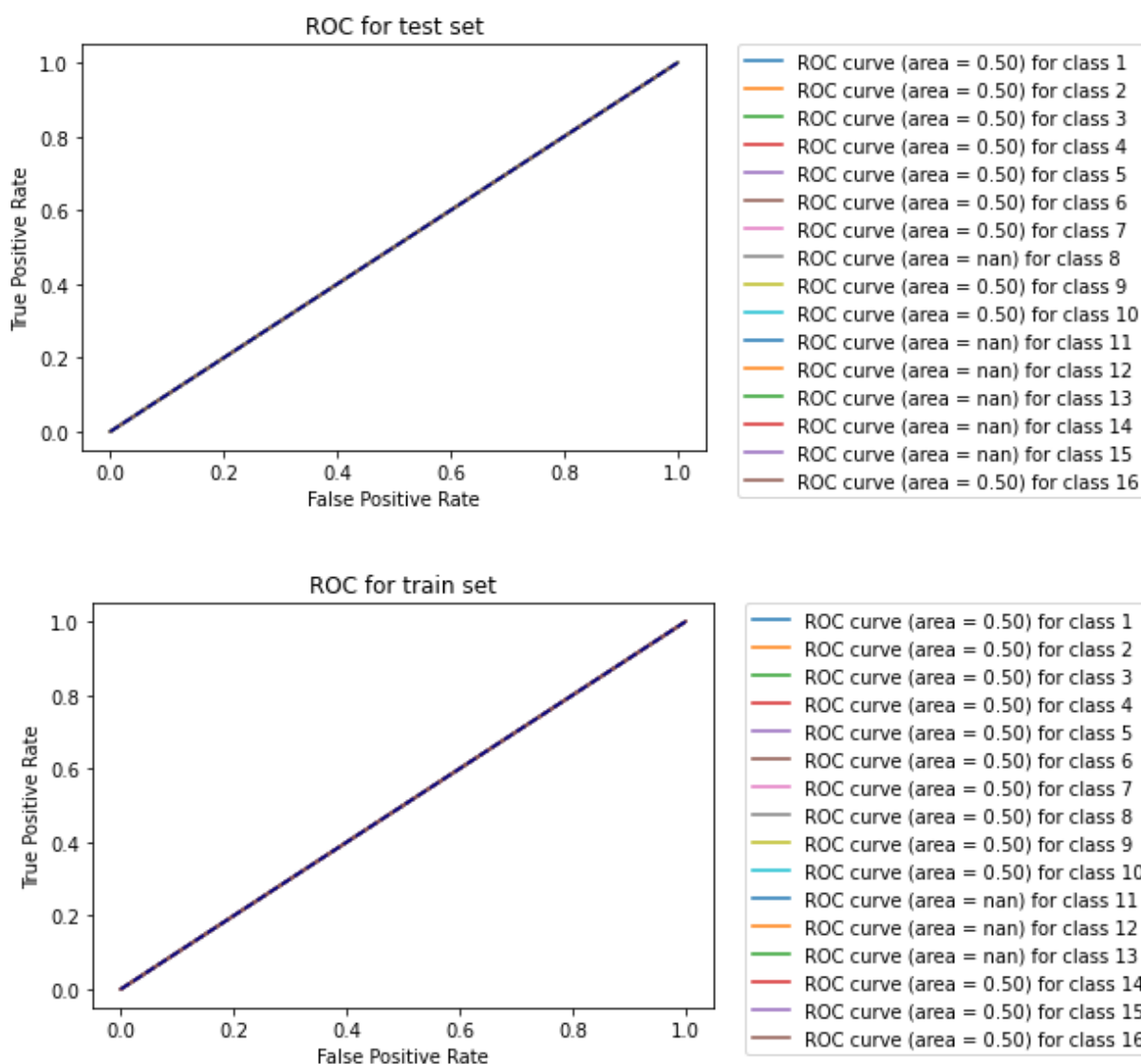
	precision	recall	f1-score	support
1	0.56	1.00	0.72	76
2	0.00	0.00	0.00	17
3	0.00	0.00	0.00	7
4	0.00	0.00	0.00	4
5	0.00	0.00	0.00	2
6	0.00	0.00	0.00	7
7	0.00	0.00	0.00	1
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	16
16	0.00	0.00	0.00	5
accuracy			0.56	136
macro avg	0.06	0.10	0.07	136
weighted avg	0.31	0.56	0.40	136

داده های آموزشی :

	precision	recall	f1-score	support
1	0.53	1.00	0.70	169
2	0.00	0.00	0.00	27
3	0.00	0.00	0.00	8
4	0.00	0.00	0.00	11
5	0.00	0.00	0.00	11
6	0.00	0.00	0.00	18
7	0.00	0.00	0.00	2
8	0.00	0.00	0.00	2
9	0.00	0.00	0.00	8
10	0.00	0.00	0.00	34
14	0.00	0.00	0.00	4
15	0.00	0.00	0.00	5
16	0.00	0.00	0.00	17
accuracy			0.53	316
macro avg	0.04	0.08	0.05	316
weighted avg	0.29	0.53	0.37	316

آخرین کار هم مقایسه نمودارهای ROC و مساحت آنها که برای هر کلاس در کنار نمودار نوشته شده :

در هر دو مورد داده های تست و آموزش عملکرد به یک مقدار بد است! بعضی کلاسها که داده کافی نبوده همچنان صفر و بدون مقدار و بقیه هم برابر ۰٫۵۰ شده که مانند این هست که به طور شانسی تعیین کنیم این داده عضو کلاس یک هست یا نه. (همه رنگها و همه نمودارها روی یک خط افتاده)



ج)

اول پارامترها را تعیین کنیم بعد با استفاده از GridSearchCV مقادیر مناسب را با الگوریتم ارزیابی متقابل پیدا می‌کنیم. در قسمت قبلی نتیجه ۱ و ۳۰ را داشتیم و یکی خیلی زیاد و یکی خیلی کم بود. پس عدد مناسب برای  $k$  را باید داخل این بازه پیدا کنیم. معیار فاصله هم همان که در سوال گفته شده. اگر معیار ارزیابی خاصی برایمان اولویت دارد ان را هم تعیین کنیم ولی اینجا من استفاده نکردم و خالی گذاشتم.

بعد از جستجو بهترین حالتی که برای KNN روی این داده ها وجود دارد :

```
{'metric': 'cosine', 'n_neighbors': 5}
```



یعنی فاصله کسینوسی و مقدار  $k$  برابر ۵ و با برآوردگیری از داده های تست و آموزش با مدل جدید داریم :

تست :

```
[75, 1, 0, 0, 1, 0, 0, 0, 0, 0]
[12, 1, 0, 0, 0, 0, 0, 0, 0, 0]
[ 0, 0, 6, 0, 0, 1, 0, 0, 0, 0]
[ 3, 0, 0, 3, 0, 0, 0, 0, 0, 0]
[ 2, 1, 0, 0, 0, 0, 0, 1, 0, 0]
[ 7, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 3, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[14, 1, 0, 0, 0, 0, 0, 0, 0, 0]
[ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[ 1, 0, 0, 0, 0, 1, 0, 0, 0, 0]
```

نتیجه ای که از داده های تست بدست آمده نزدیکی زیادی به نتیجه تست به ازای  $k=30$  دارد و همینطور که از اختلاف اعداد مشخص است در واقع بهبود یافته همان است. یعنی پیچیدگی مدل بیشتر شده تا تعادل بهتری بین تست و آموزش به وجود بیاید.

	precision	recall	f1-score	support
1	0.64	0.97	0.77	77
2	0.25	0.08	0.12	13
3	1.00	0.86	0.92	7
4	1.00	0.50	0.67	6
5	0.00	0.00	0.00	4
6	0.00	0.00	0.00	7
9	1.00	0.25	0.40	4
10	0.00	0.00	0.00	15
14	0.00	0.00	0.00	1
16	0.00	0.00	0.00	2
accuracy			0.63	136
macro avg	0.39	0.27	0.29	136
weighted avg	0.51	0.63	0.54	136

نوسان معیارهای  $f1$ , precision, recall روی گروه های مختلف زیاد است و علت همان مسائل مطرح شده روی قسمت قبلی هست. بستگی به اینکه مثلا گروه ۹ و تشخیص درست این کلاس چه اهمیت و چه هزینه ای برای ما دارد میتوان معیار مناسب برای ارزیابی مدل را انتخاب کرد. مثلا مدل برای همین کلاس ۹ در precision عملکرد خوبی داشته یعنی احتمال اینکه مدل موردی از کلاسهای دیگر را به کلاس ۹ نسبت دهد کم است اما در عوض معیار recall ضعیفی دارد که یعنی تعداد عناصری از این کلاس که به اشتباه در کلاسهای دیگر میافتد زیاد است. اگر وضعیت کسی که در واقعیت کلاس ۹ هست معادل نیازمند اقدامات و مراقبتهای فوری باشد این مدل از انجامی که نمی تواند این بیماران را شناسایی کند (مثلا ممکن است

وضعیت آنها را در کلاس سالم هیا غیر اورژانسی ها طبقه بندی کند) مدل قابل استفاده ای نیست و ریسک زیادی دارد اگر در این جهت استفاده شود. ولی اگر برعکس این باشد یعنی مثلا شرایط بیمار ۹ طوری باشد که مراقبتهای لازم برای ان برای سایر رده ها ضرر داشته باشد یا اینکه هزینه سنگینی داشته باشد و نخواهم بیماران اشتباهی را به این بخش بفرستیم این عدد precision خوب است! ولی در کل این اختلاف زیاد در معیارهای ارزیابی مطلوب نیست. هرچه تمام اعداد مقدار بالاتر داشته باشد و به یکدیگر نزدیک تر باشند یعنی مدل به ازای اعضای تمامی کلاسها نتیجه ی یکنواخت تر و مطلوب تری می دهد. اگر کلاس ۳ را در نظر بگیریم در همه معیارهای بالا عملکرد مطلوبی دارد.

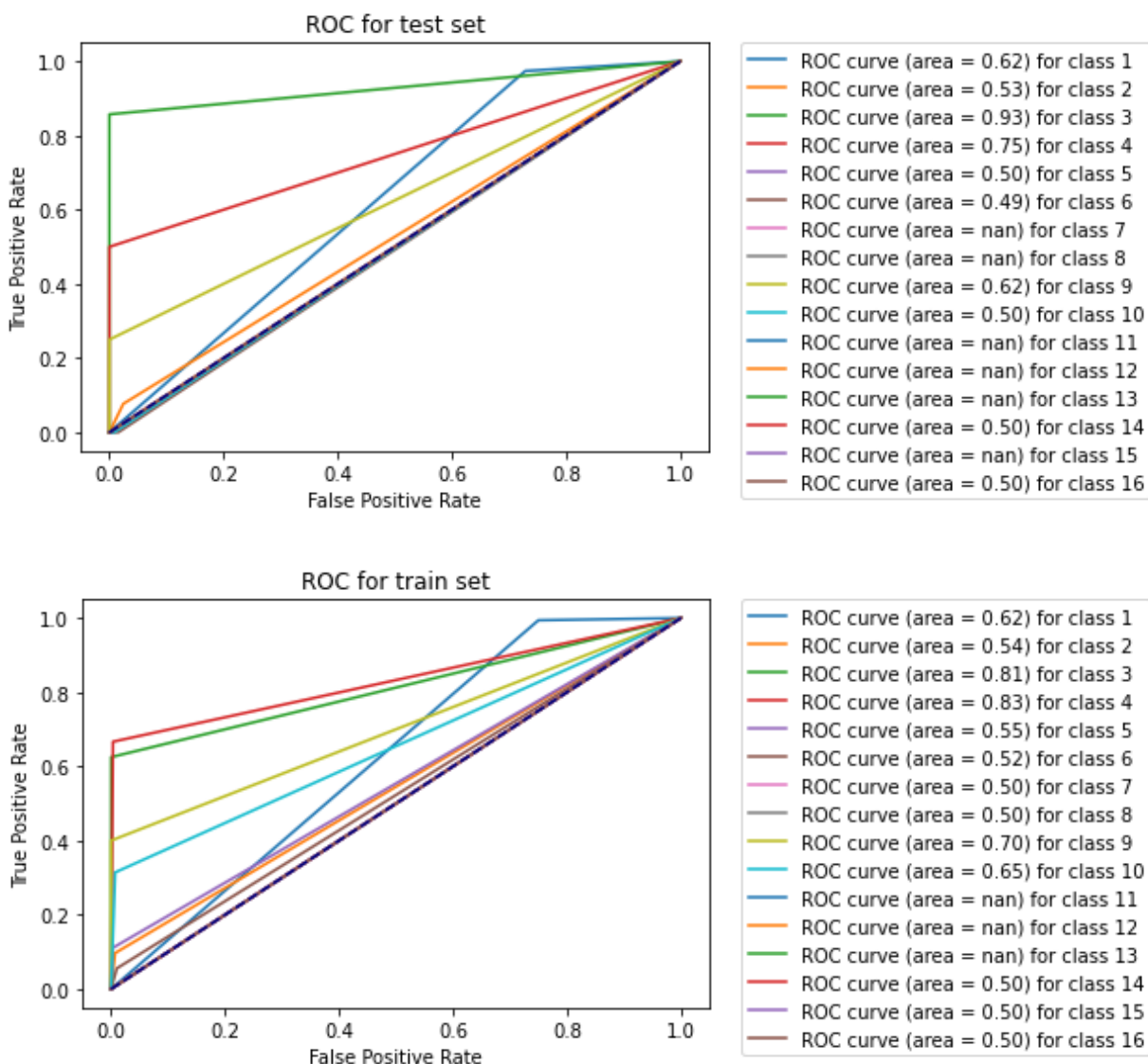
در داده های آموزشی :

```
[167, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
[ 27, 3, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0].
[  3, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
[  3, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0].
[  8, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0].
[ 17, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0].
[  2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
[  2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
[  2, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0].
[ 24, 0, 0, 0, 0, 0, 0, 0, 0, 11, 0, 0, 0].
[  3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
[  3, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0].
[ 17, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0].
```

	precision	recall	f1-score	support
1	0.60	0.99	0.75	168
2	0.60	0.10	0.17	31
3	1.00	0.62	0.77	8
4	0.86	0.67	0.75	9
5	0.50	0.11	0.18	9
6	0.25	0.06	0.09	18
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	2
9	1.00	0.40	0.57	5
10	0.85	0.31	0.46	35
14	0.00	0.00	0.00	3
15	0.00	0.00	0.00	5
16	0.00	0.00	0.00	20
accuracy			0.62	316
macro avg	0.43	0.25	0.29	316
weighted avg	0.57	0.62	0.53	316

در نهایت مقایسه ی مقدار ROC مساحت نمودارها به ازای این مدل بهینه ای که پیدا شده :

برای حدود نیمی از کلاس ها عملکرد بالای ۰٫۵ داریم و این خوب است. همگونی نتایج تست و آموزش اینجا کاملاً محسوس است. بهترین عملکرد را برای تشخیص کلاس ۳ داریم. بقیه کلاس هایی که در ۰٫۵ ماندند نشان میدهند که اگر از مدل برای تشخیص این کلاس ها استفاده کنیم تشخیص چندان معتبر نخواهد بود.



اختلاف داده های تست و آموزش نسبت به  $k=1$  خیلی بهتر شده. تفاوت نتایج بدست آمده برای مجموعه های آموزش و تست خیلی کمتر شده و به نظرم دقت موجود در حدی هست که قابل قبول باشد و مشکل underfit نداشته باشیم. اما به نظرم ناهمگونی عملکرد مدل برای کلاس های مختلف باعث می شود اعتبار مدل خیلی زیاد نباشد و به نظرم نمی شود ان را مدل خوبی دانست. (مخصوصاً که موضوع استفاده از پزشکی هست) یکی از اصلی ترین مشکلات KNN این است که تا کاملاً به داده ها بستگی دارد و مثلاً شاید اگر داده ها مجموعه ی دقیق تر و مناسب تری به لحاظ توزیع کلاس ها بود، میشد که مدل بهتری ساخت. البته روی همین داده ها هم با انجام پیش پردازش ها و روش های بالانس کردن داده ها یا داده افزایی و ... احتمالاً بشود مدلی با کارایی بهتر هم ساخت.

پ.ن: مثلاً من طی ۳ بار تقسیم مختلف داده های تست و آموزش با ۳ مجموعه آموزش مختلف با استفاده GridSearchCV به ۳ نتیجه مختلف برای مدل مطلوب رسیدم که  $k$  برابر ۳ و ۵ و ۴ بود و متریک های متفاوتی هم داشتند. و نتایج و اعداد بدست آمده هم طبیعتاً باهم تفاوت داشت. نتایج را ضبط نکردم اما همین تفاوت بستگی زیاد مدل KNN به داده ها را نشان می دهد.