
بسم الله الرحمن الرحيم

عنوان

تمرین پنجم درس یادگیری ماشین (یادگیری گروهی)

دانشجو

امیرحسین جراره – ۴۰۰۶۱۶۰۰۴

استاد درس

دکتر عبدی هجراندوست

الگوریتم XGBoost

XGBoost مخفف Extreme Gradient Boosting است، که در آن اصطلاح Gradient Boosting از مقاله Greedy Function Approximation: A Gradient Boosting Machine اثر فریدمن نشات می گیرد. درختان تقویت‌شده با گرادیان مدتی است که وجود داشته است و مطالب زیادی در مورد این موضوع وجود دارد. این آموزش درختان تقویت شده را به روشی مستقل و اصولی با استفاده از عناصر یادگیری نظارت شده توضیح می دهد. ما فکر می کنیم این توضیح تمیزتر، رسمی تر است و به فرمول مدل مورد استفاده در XGBoost انگیزه می دهد.

مدل در یادگیری نظارت شده معمولا به ساختار ریاضی اشاره دارد که توسط آن پیش بینی می شود

از ورودی ساخته شده است. یک مثال رایج یک مدل خطی است که در آن پیش‌بینی به صورت داده ارائه شده است، ترکیبی خطی از ویژگی های ورودی وزنی. مقدار پیش‌بینی بسته به کار، یعنی رگرسیون یا طبقه‌بندی، می‌تواند تفسیرهای متفاوتی داشته باشد. برای مثال، می‌توان آن را تبدیل به لجستیک کرد تا احتمال کلاس مثبت در رگرسیون لجستیک را بدست آورد و همچنین می‌توان از آن به عنوان امتیاز رتبه‌بندی زمانی که می‌خواهیم خروجی‌ها را رتبه‌بندی کنیم، استفاده کرد.

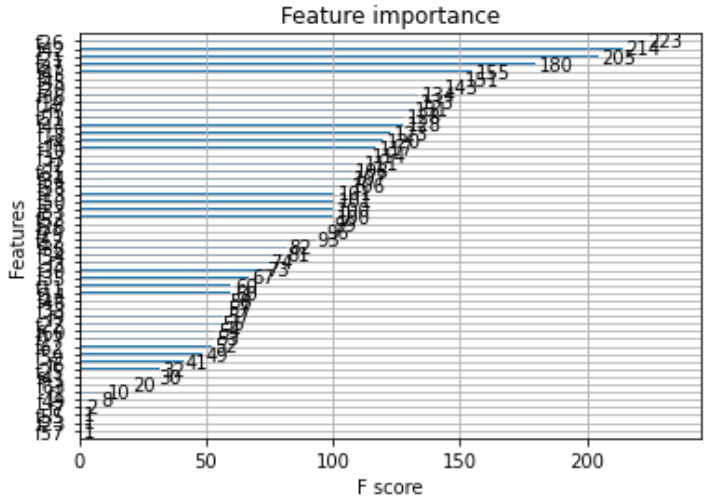
پارامترها بخش نامشخصی هستند که باید از داده ها یاد بگیریم. در مسائل رگرسیون خطی، پارامترها ضرایب هستند.

مسئله Mnist در XGBoost :

برای شروع و ارزیابی از مجموعه دادگان اعداد دست نویس شروع کردیم که به خروجی زیر رسیدیم.

```
Accuracy: 0.972222
```

همچنین نرخ معیار اهمیت ویژگی بر حسب معیار f score نیز به صورت زیر می باشد.

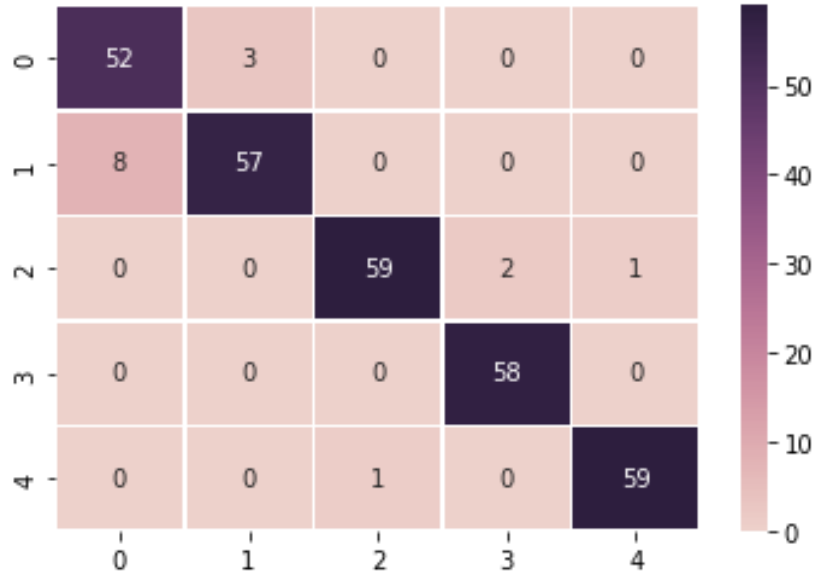


دقت در مقایسه با درخت تصمیم با دقت ۸۸ و شبکه عصبی ANN با دقت ۹۱ درصد بهتر بوده و دقت ۹۸ درصد را کسب کرده است ولی از دقت CNN بادقت ۹۹ درصد کمتر است.

مسئله Persian LPR **در** XGBoost **:**

مسئله اعداد پلاک و پیاده سازی آن مانند دیتاست مجموعه دادگان اعداد دست نویس می باشد فقط دو حرف S و W را به اعداد ۰ و ۱ نگاشت داده ایم. سپس اعداد را به صورت numpy array مانند اعداد دست نویس مرتب نموده و برچسب ها را تنظیم می کنیم(با استفاده از کتابخانه glob) خروجی های مطلوب به همراه ماتریس در همریختگی برابر است با :

	precision	recall	f1-score	support
0.0	0.87	0.95	0.90	55
1.0	0.95	0.88	0.91	65
2.0	0.98	0.95	0.97	62
3.0	0.97	1.00	0.98	58
7.0	0.98	0.98	0.98	60
accuracy			0.95	300
macro avg	0.95	0.95	0.95	300
weighted avg	0.95	0.95	0.95	300

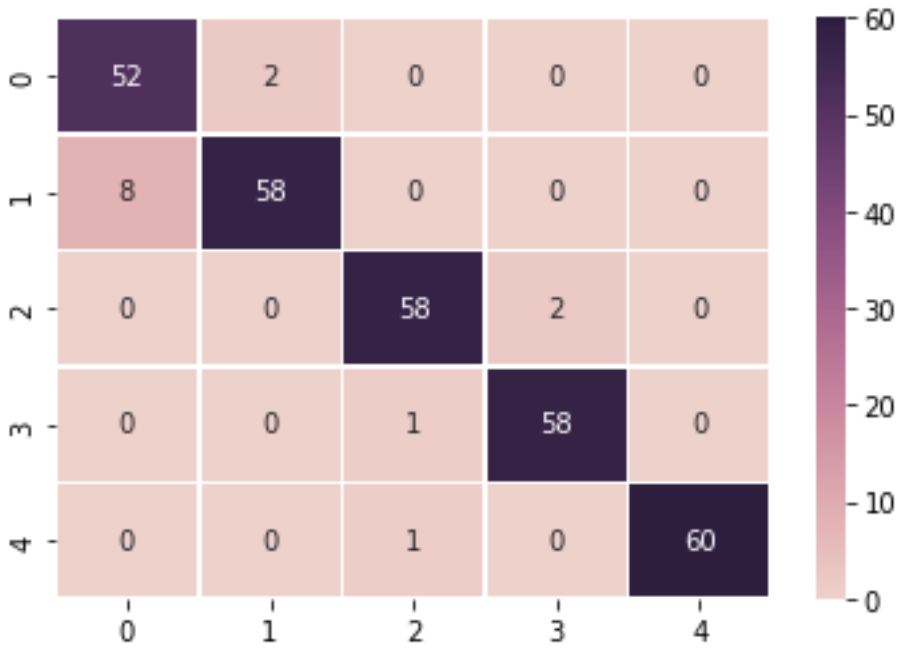


دقت با توجه به داده کم ۹۵ درصد می باشد.

مسئله Persian LPR **در** ADABoost **:**

مانند حالت قبل خروجی را به یک الگوریتم ADABoost نیز می دهیم. خروجی های مطلوب به همراه ماتریس درهمریختگی برابر است با:

	precision	recall	f1-score	support
0	0.87	0.96	0.91	54
1	0.97	0.88	0.92	66
2	0.97	0.97	0.97	60
3	0.97	0.98	0.97	59
7	1.00	0.98	0.99	61
accuracy			0.95	300
macro avg	0.95	0.96	0.95	300
weighted avg	0.96	0.95	0.95	300



دقت تقریبا برابر XGBoost می باشد.

مسئله Persian LPR **در** Random Forest **:**

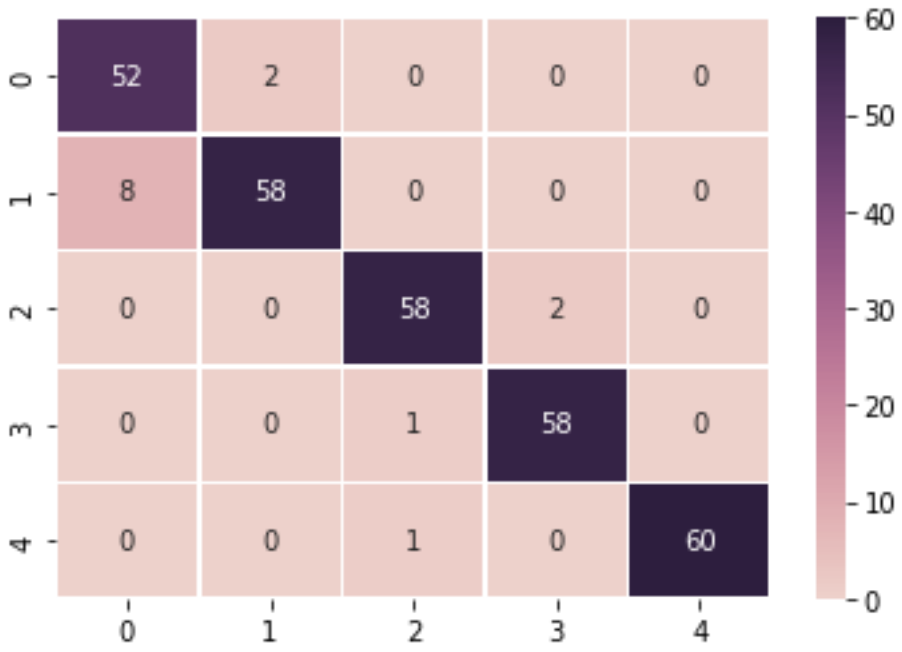
مانند حالت قبل خروجی را به یک الگوریتم Random Forest نیز می دهیم. خروجی های مطلوب به همراه ماتریس درهمریختگی برابر است با:

```
              precision    recall  f1-score   support

           0       0.87      0.96      0.91        54
           1       0.97      0.88      0.92        66
           2       0.97      0.97      0.97        60
           3       0.97      0.98      0.97        59
           7       1.00      0.98      0.99        61

    accuracy                           0.95       300
   macro avg       0.95      0.96      0.95       300
weighted avg       0.96      0.95      0.95       300

AxesSubplot(0.125,0.125;0.62x0.755)
```



دقت تقریبا برابر XGBoost می باشد.

مسئله Persian LPR **در درخت تصمیم :**

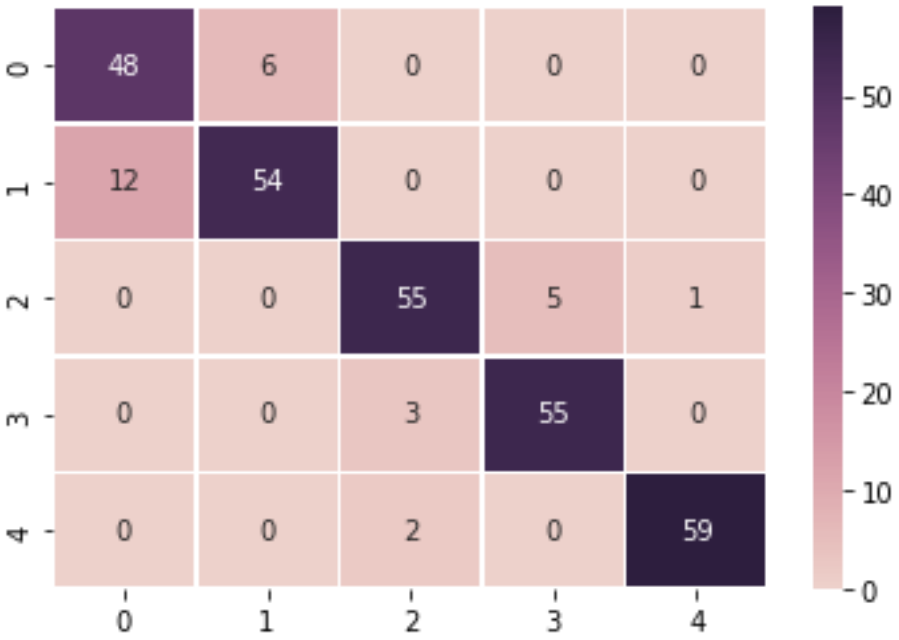
مانند حالت قبل خروجی را به یک الگوریتم درخت تصمیم معمولی نیز می دهیم. خروجی های مطلـوب به همراه ماتریس درهمریختگی برابر است با:

```
              precision    recall  f1-score   support

           0       0.80      0.89      0.84        54
           1       0.90      0.82      0.86        66
           2       0.92      0.90      0.91        61
           3       0.92      0.95      0.93        58
           7       0.98      0.97      0.98        61

    accuracy                           0.90       300
   macro avg       0.90      0.90      0.90       300
weighted avg       0.91      0.90      0.90       300

AxesSubplot(0.125,0.125;0.62x0.755)
```



همانطور که مشاهده می شود دقت از سایر روش ها کمتر و برابر ۹۰ درصد می باشد.

مسئله Titanic **در** XGBoost**:**

این مسئله نیز مانند حالت قبل می باشد. هدف تعیین زنده بودن یا فوت افراد بر حسب ویژگی هـای آن ها در کشتی تایتانیک می باشد . مشکل از آنجایست که برخی از داده های دیتاست ناقص می باشند که به صورت زیر می باشد.

```
pclass 0
name 0
sex 0
age 263
sibsp 0
parch 0
ticket 0
fare 1
cabin 1014
embarked 2
```

```
boat 823
body 1188
home.dest 564
```

سپس داده هایی که در تعدا بالا وجود ندارند یا تفاوتی ایجاد نمی کننـد ماننـد نـام افـراد را drop مـی کنیم. این داده ها عبارتند از:

```
'age', 'cabin','body','boat', 'home.dest','name','ticket'
```

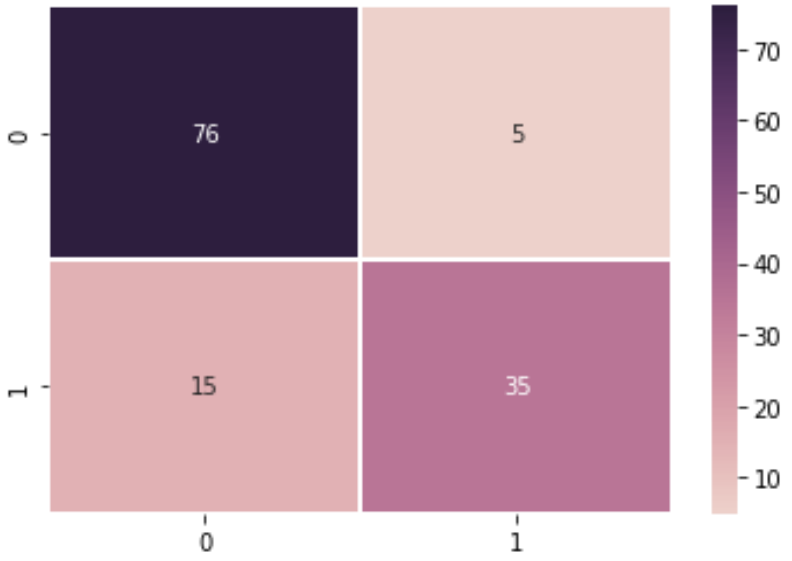
همچنین سایر داده ها که تفاوت ایجاد می کنند یا قابل بازیابی هستند را با میانگین و یا با سایر داده ها مانند کابین افراد بازیابی می کنیم. این داده عبارتند از :

```
pclass 0
sex 0
sibsp 0
parch 0
fare 1 #pred 1 samples with mean
embarked 2 #pred 2 samples with mean
nulls 0
cabin_mapped 0
```

سپس تایپ داده ها را به تایپ های قابل درک تبدیل و به شبکه می دهیم . خروجی های مطلوب به همراه ماتریس درهمریختگی برابر است با :

```
              0       0.84      0.94      0.88        81
              1       0.88      0.70      0.78        50

    accuracy                            0.85       131
   macro avg       0.86      0.82      0.83       131
weighted avg       0.85      0.85      0.84       131
```


76
5
15
35
0
1
0
1
70
60
50
40
30
20
10