

به نام خدا

پایان ترم مبانی سیستم های هوشمند

امیر ارشام بهشتی طهرانی

9820293

سوال (1)

سوال (1)

غیرنازی ساز: $a_i x + b_i y + c_i$

نازی ساز: $\mu_{L_i}(x, y)$

مدل:

$$z = \sum_{i=1}^N \mu_{L_i}(x, y) (a_i x + b_i y + c_i)$$

$$\Rightarrow z = \sum_{i=1}^N \mu_{L_i}(x, y) (a_i x + b_i y + c_i)$$

$$\Rightarrow z = \sum_{i=1}^N [a_i \quad b_i \quad c_i] \begin{bmatrix} x \mu_{L_i}(x, y) \\ y \mu_{L_i}(x, y) \\ \mu_{L_i}(x, y) \end{bmatrix}$$

$$\Rightarrow z = \begin{bmatrix} a_1 & b_1 & c_1 & a_2 & b_2 & c_2 & \dots & a_N & b_N & c_N \end{bmatrix}_{1 \times N} \begin{bmatrix} x \mu_{L_1}(x, y) \\ y \mu_{L_1}(x, y) \\ \mu_{L_1}(x, y) \\ \vdots \\ x \mu_{L_N}(x, y) \\ y \mu_{L_N}(x, y) \\ \mu_{L_N}(x, y) \end{bmatrix}_{N \times 1}$$

$$\Rightarrow z = \begin{bmatrix} z^1 \\ z^2 \\ \vdots \\ z^M \end{bmatrix} = \begin{bmatrix} x^1 \mu_{L_1}(x, y) & y^1 \mu_{L_1}(x, y) & \mu_{L_1}(x, y) & \dots & x^M \mu_{L_N}(x, y) & y^M \mu_{L_N}(x, y) & \mu_{L_N}(x, y) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x^M \mu_{L_1}(x, y) & y^M \mu_{L_1}(x, y) & \mu_{L_1}(x, y) & \dots & x^M \mu_{L_N}(x, y) & y^M \mu_{L_N}(x, y) & \mu_{L_N}(x, y) \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ \vdots \\ a_N \\ b_N \\ c_N \end{bmatrix}$$

Φ Θ

$$\Rightarrow Z_{M \times 1} = \Phi_{M \times N} \theta_{N \times 1} \Rightarrow$$
 با توجه به رابطه خطی این رابطه، می‌توانیم با کمک
 pseudo inverse، از رابطه LS کمک بگیریم و
 پارامترهای a_i, b_i, c_i را به بهترین حالت update کنیم.

$$\Rightarrow \Phi^T Z = (\Phi^T \Phi) \theta \rightarrow \theta = (\Phi^T \Phi)^{-1} \Phi^T Z$$

سوال (2)

قسمت 1:

برای انتخاب ویژگی اول به عنوان ریشه درخت باید ویژگی با بیشترین قدرت تفکیک پذیری یا ویژگی با بیشترین بهره اطلاعات (Information Gain) را انتخاب کنیم. با استدلال و طبقه بندی با کمک هر ویژگی می‌توان مشاهده کرد که سطح صاف در صورت 1 بودن جدا سازی بیشتری دارد 3 نمونه سمی و یک نمونه غیر سمی (هر چند در صورت صفر بودن داده‌ها برابر تقسیم می‌شوند دو نمونه سمی و 2 نمونه غیر سمی) اما می‌توان شاخه سمی را ادامه نداد و با اطمینان خوبی سمی در نظر گرفت. به همین دلیل "صاف بودن" در ریشه قرار دارد.

بهره اطلاعات در محاسبات حدود 0.09 است که مقدار پایینی می‌باشد و در نتیجه به خوبی جدا سازی صورت نگرفته است.

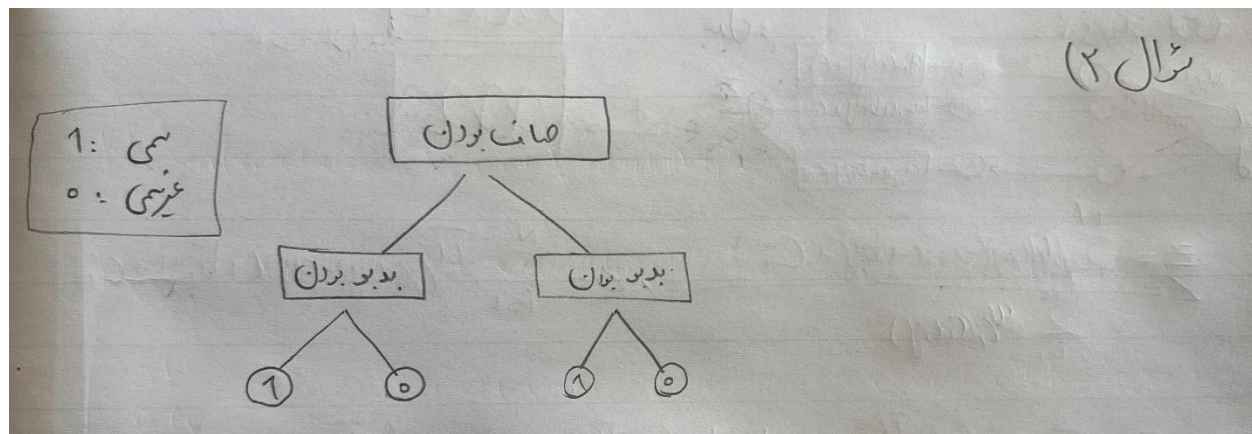
$$\text{Entropy}(x) = -\left(\frac{5}{10} \log_2 \frac{5}{10} + \frac{5}{10} \log_2 \frac{5}{10}\right) = -0.52 + -0.52 = 0.98$$

$$\text{سمی: } -\left(\frac{5}{10} \log_2 \frac{5}{10} + \frac{1}{5} \log_2 \frac{1}{5}\right) = 0.81 + 0.5 = 0.11$$

$$\text{غیر سمی: } -\left(\frac{5}{10} \log_2 \frac{5}{10} + \frac{5}{10} \log_2 \frac{5}{10}\right) = 1$$

$$\text{IG} = 0.98 - \frac{5}{10}(0.11) - \frac{5}{10}(1) = 0.98 - 0.55 - 0.5 = 0.09$$

$$\text{IG} = \text{Entropy}(x) - \sum_{i=1}^N \frac{|X_i|}{|X|} \text{Entropy}(X_i)$$



قسمت 2:

هر سه سمی هستند.

خیر زیرا طبق درخت تصمیم همگی سمی هستند.

سوال 3

قسمت 1:

ابتدا داده ها را در دیتاست، با ویژگی ها در x و $target$ ها در y قرار می دهیم. سپس با کمک تابع `Random Unsampler` موجود در برنامه، `unbalance` بودن نمونه ها در کلاس های مختلف را حل می کنیم. با این کار همچنین تعداد کل نمونه ها از مقدار زیاد حدود 500000 را به حدود 19000 در کل و 2700 در هر کلاس می رسانیم. سپس به کمک `train_test_split` داده ها را با نسبت 85 به 15 درصد به آموزش و تست تقسیم می کنیم تا مدل را با آن ها آموزش داده و ارزیابی کنیم.

به کمک `DecisionTreeClassifier` مدل خود را می سازیم و آن را با داده های `train` آموزش می دهیم به کمک متد `fit()`. در ادامه برای مشاهده درخت و نحوه عملکرد آن از `plot_tree` استفاده می کنیم اما با توجه به تعداد کلاس ها و پیچیدگی احتمالی آن ها و همچنین واضح نبودن شکل خروجی با عمق زیاد از همان ابتدا `hyper param.` ها را تنظیم می کنیم. ابتدا `max_depth` یا عمق درخت را 3 در نظر می گیریم تا محاسبات کمتر شود. ضریب هرس کردن (`pruning`) که با `ccp_alpha` مشخص شده را 0.01 قرار می دهیم.

برای درک بهتر از تقسیم بندی درخت، از `export_text` استفاده کرده تا روند تصمیمات درخت را متوجه شویم.

```
|--- feature_0 <= 2673.50
|   |--- feature_0 <= 2374.50
|   |   |--- feature_6 <= 195.50
|   |   |   |--- class: 3
|   |   |--- feature_6 > 195.50
|   |   |   |--- class: 4
|   |--- feature_0 > 2374.50
|   |   |--- class: 6
|--- feature_0 > 2673.50
|   |--- feature_0 <= 3226.50
|   |   |--- feature_0 <= 2948.50
|   |   |   |--- class: 5
|   |   |--- feature_0 > 2948.50
|   |   |   |--- class: 1
|   |--- feature_0 > 3226.50
|   |   |--- class: 7
```

طبق این درخت ویژگی با بهره اطلاعات بالاتر که به عنوان ریشه درخت انتخاب شده، feature صفرم داده هاست. پس از تقسیم بندی در decision node های بعدی باز هم از همین ویژگی استفاده شده است. اما در عمق سوم درخت مشاهده می شود که در چپ درخت از feature ششم و در سمت راست باز هم از feature صفرم استفاده شده تا کلاس بندی کامل شود. مشکل بزرگ این مدل حتی قبل از بررسی دقت عدم پیشبینی کلاس 2 در داده هاست. در نتیجه باید دوباره تنظیم شود.

قسمت 2:

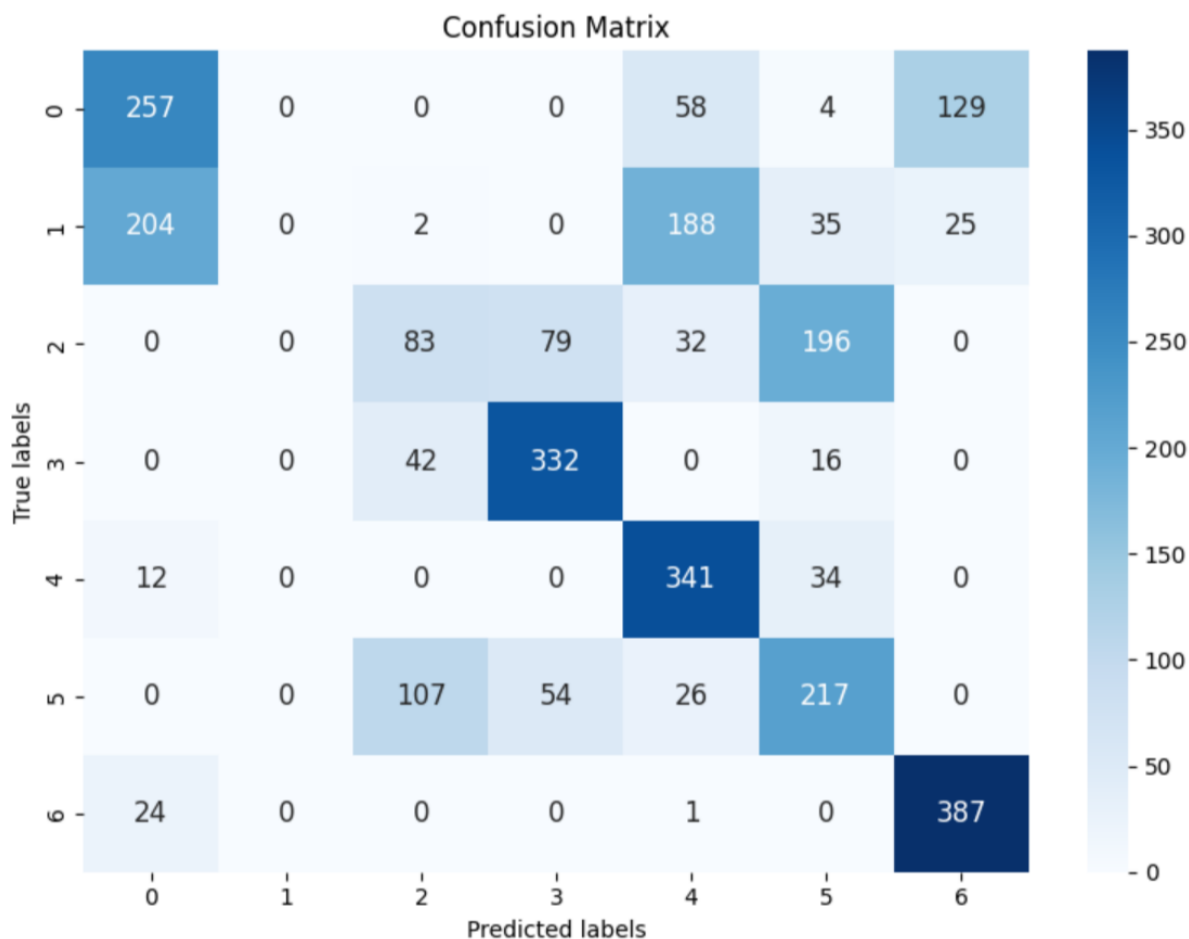
با کمک predict. روی داده های تست y_pred خروجی پیش بینی شده مدل ما خواهد بود. از آن استفاده می کنیم تا ماتریس confusion را تشکیل دهیم و داده ها را تحلیل کنیم.

Train Accuracy: 0.5817425354870289

Test Accuracy: 0.5604852686308492

مشاهده می شود دقت روی داده های آموزش و تست پایین است.

ماتریس درهم ریختگی:



تحلیل:

با توجه به ماتریس کلاس 0، 3 و 4 تقریباً توسط مدل به خوبی کلاس بندی شده اند. کلاس 1 بدترین حالت ممکن داشته و هیچ کدام از نمونه هایش به درستی پیش بینی نشده اند.

شاخصه ها:


```
c = model.n_classes_  
n_c = model.tree_.node_count  
n_l = model.tree_.n_leaves  
n_s = model.tree_.n_node_samples
```

```
print(c)  
print(n_c)  
print(n_l)  
print(n_s)
```

```
7  
11  
6  
[16344  7172  4130  1372  2758  3042  9172  6240  3690  2550  2932]
```

شاخصه اول تعداد کلاس ها که 7 تا بوده است. شاخه دوم تعداد گره ها که 11 گره داریم.

برای تعداد برگ ها که کلاس ها را تخمین می زنند، 6 برگ داریم. (1 عدد کمتر که مشکل بزرگ مدل است) و در نهایت تعداد نمونه ها در 11 گره موجود را می بینیم.

بهبود مدل:

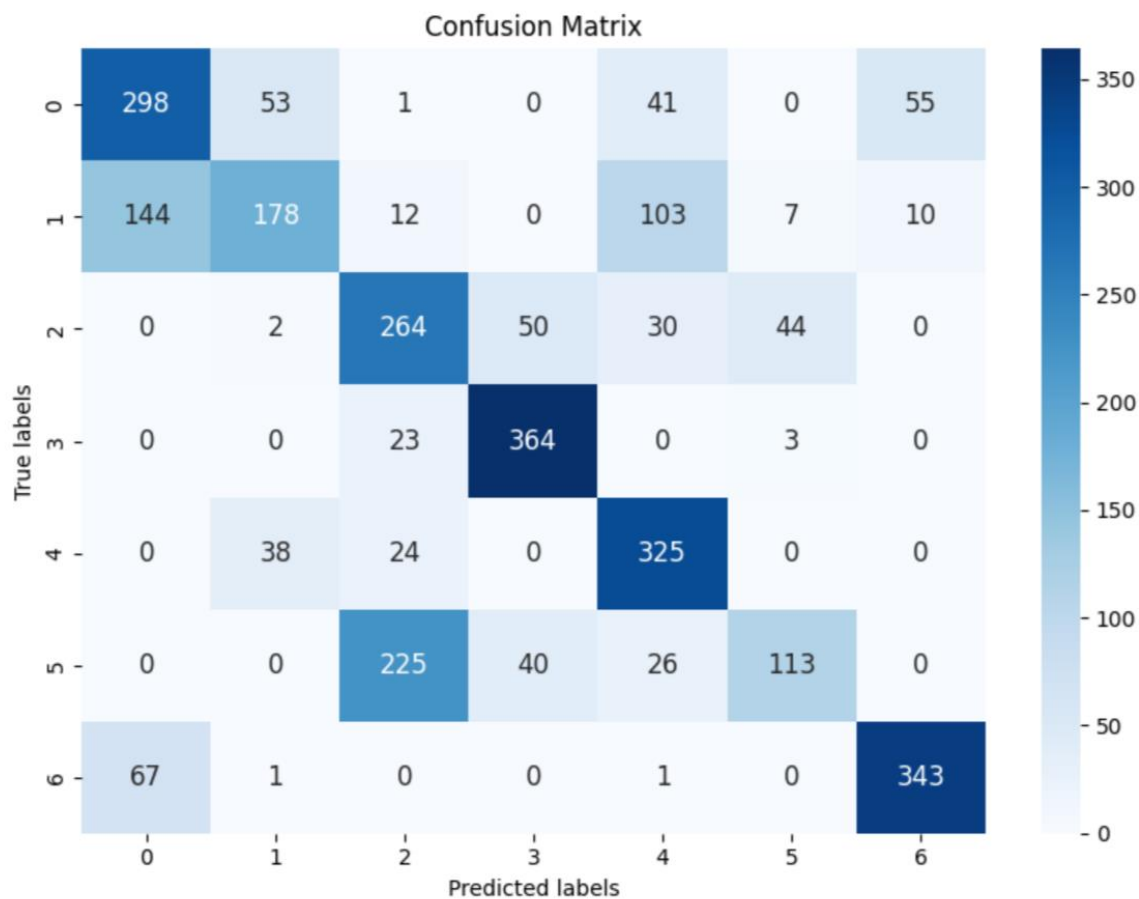
در این حالت ابتدا مدل را با "آنتروپی" به جای gini در نظر می گیریم هر چند احتمالا تفاوت زیادی در دقت نخواهد داشت. برای رسیدن به دقت بهتر نیاز داریم تا عمق درخت را افزایش دهیم برای این کار آن را روی 6 در نظر می گیریم. ضریب هرس کردن درخت را هم برابر 0.005 قرار داده زیرا در حالات دیگر و بیشتر از این مقدار دقت کاهش می یافت.

در ادامه model2 را تعریف می کنیم.

Train Accuracy: 0.6666054821341165

Test Accuracy: 0.6533795493934142

دقت مدل کمی افزایش یافته است.



با توجه به ماتریس هم نمونه کمی بهتر عملکردده اند.

```
c = model2.n_classes_
n_c = model2.tree_.node_count
n_l = model2.tree_.n_leaves
n_s = model2.tree_.n_node_samples
```

```
print(c)
print(n_c)
print(n_l)
print(n_s)
```

7

57

29

```
[16344  7185  4377  1458   144  1314   461   195   266   853  2919   839
    23    816  2080  1200   611   589   880  2808  2579  1771   788   217
    571   983   603   380   808   229  9159  4268  3085  1854   721   416
    305  1133  1024   109  1231   664   567  1183   844   575   269   339
    4891  1959  1070   889  2932   883   703   180  2049]
```


می بینیم که شاخصه ها هم متناسب با مدل تغییر کرده اند.

هرس کردن علاوه بر جل.گیری از overfit باعث می شود تا برخی نقاط درخت که نمونه های یک کلاس از بقیه بیشتر است ادامه پیدا نکنند که هم از overfit جلوگیری کرده و هم پارامترها را کاهش می دهد.

سوال 4 قسمت 2:

سوال (4) «قیمت دوم»

$$f(x_1, x_2) = \sin(x_1 \pi) + \cos(x_2 \pi) + \sin(x_1 \pi) \cos(x_2 \pi)$$

نشان بده: $\|f - g\|_\infty < \left\| \frac{\partial f}{\partial x_1} \right\|_\infty h_1 + \left\| \frac{\partial f}{\partial x_2} \right\|_\infty h_2 < \epsilon$

با قرار دادن رابطه کوید ترازع (تت مورد نظر)، مای دانم مقما دت $\epsilon = 0.1$ براده
خواه شد چون در بدین انتخاب $\{h\}$ ، ϵ را براده خواهیم کرد.

$$f = \sin(x_1 \pi) + \cos(x_2 \pi) + \sin(x_1 \pi) \cos(x_2 \pi) \Rightarrow$$

$$\Rightarrow \begin{cases} \frac{\partial f}{\partial x_1} = \pi \cos(x_1 \pi) + \pi \cos(x_1 \pi) \cos(x_2 \pi) = \pi \cos(x_1 \pi) (1 + \cos(x_2 \pi)) \\ \frac{\partial f}{\partial x_2} = -\pi \sin(x_2 \pi) - \pi \sin(x_2 \pi) \sin(x_1 \pi) = -\pi \sin(x_2 \pi) (1 + \sin(x_1 \pi)) \end{cases}$$

$$\begin{cases} \left\| \frac{\partial f}{\partial x_1} \right\|_\infty = 2\pi & \begin{matrix} x_2=0 \\ x_1=0 \end{matrix} \\ \left\| \frac{\partial f}{\partial x_2} \right\|_\infty = 2\pi & \begin{matrix} x_1=\frac{1}{2} \\ x_2=-\frac{1}{2} \end{matrix} \end{cases} \Rightarrow 2\pi h_1 + 2\pi h_2 \leq 0.1 \Rightarrow$$

نام بران $\Rightarrow 4\pi h \leq 0.1 \Rightarrow h \leq \frac{0.1}{4\pi} \Rightarrow h \leq 0.0079 \Rightarrow h = 0.005$

$\Rightarrow n = \frac{1 - (-1)}{0.005} + 1 = \frac{2000}{5} + 1 = 401$ تایزن برای
نقدار تریع نازی نر x

\Rightarrow نقدار تریان: $401 \times 401 = 160801$

تعداد توابع تعلق عدد بسیار بزرگی شده است. برای حل این روش می توان از

کران مرتبه ۲ استفاده کرد که
$$\|f-g\| = \frac{1}{8} \left[\left\| \frac{\partial f}{\partial x_1} \right\| h_1 + \left\| \frac{\partial f}{\partial x_2} \right\| h_2 \right]$$
 است می باشد.

اما در شرایط دیگر با توجه به افتار سیستم می توان مدل های برای شناسایی سیستم طراحی

کرد که با کمک زوج های ورودی و خروجی در رفتار سیستم تعداد توابع تعلق حتی کمتری از

کران مرتبه ۲ به است آمد.