

هوش مصنوعی

«تمرین شماره ۱»

ایده کلی الگوریتم One-R این است که قوانینی بسازیم که ویژگی‌های یک مجموعه داده را بررسی کند، بهترین ویژگی موجود را برگزیده و بر اساس آن به تعدادی قانون برسد که بر اساس آن‌ها بتواند نمونه‌های جدید را کلاس‌بندی کند. برای انجام این کار، برای هر ویژگی، ابتدا مقادیری که آن ویژگی می‌تواند بگیرد را در شاخه‌های جدا قرار می‌دهیم. تعیین کلاس برای هر شاخه هر ویژگی ساده است: آن کلاسی که فراوانی بیشتری دارد، انتخاب می‌شود و میزان خطا (همان نمونه‌هایی که در کلاس اکثریت قرار نمی‌گیرند) محاسبه می‌شود. هر بار با در نظر گرفتن هر ویژگی، به یک مجموعه قوانین (که شامل یک قانون برای هر مقدار آن ویژگی می‌شود) می‌رسیم. در انتها، میزان خطای کل را برای قوانین هر ویژگی محاسبه می‌کنیم و بهترین مجموعه قوانین (آن قوانینی که خطای کلشان کمتر از همه باشد) را برمی‌گزینیم. (Ian. H. Witten, Eibe Frank, 2005)

ابتدا با استفاده از پکیج pandas فایل داده ستون‌ستون خوانده شد و در متغیرهای feature1, feature2, feature3, feature4 ذخیره شد و کلاس نمونه‌ها نیز به همین ترتیب در متغیر res قرار داده شد.

سپس، با استفاده از یک دیکشنری برای هر ویژگی، مقادیر آن ویژگی به صورت اتوماتیک به دست آمد. کلیدهای این دیکشنری بیانگر مقادیر هر فیچر و مقادیر کلیدها بیانگر فراوانی هر مقدار در آن فیچر است؛ برای مثال در فیچر اول این دیکشنری {'A': 5, 'B': 4, 'C': 5} است؛ که یعنی در فیچر اول پنج مقدار A، چهار مقدار B و پنج مقدار C وجود دارد.

در مرحله بعد، برای به دست آوردن فراوانی هر کلاس در هر مقدار، دیکشنری دوم ساخته شد. کلیدهای این دیکشنری از تاپل‌های دوتایی تشکیل شده است که عضو اول تاپل مقدار فیچر و عضو دوم کلاس آن مقدار در فیچر است. مقادیر این دیکشنری فراوانی هر تاپل (مقدار، کلاس) است. برای مثال در فیچر اول این دیکشنری به صورت زیر است:

{('A', 0): 3, ('A', 1): 2, ('B', 1): 4, ('C', 1): 3, ('C', 0): 2}

که یعنی تعداد مقدار A با کلاس صفر برابر با سه، مقدار A با کلاس ۱ برابر با یک عدد، مقدار B با کلاس ۱ برابر با چهار عدد است و به همین ترتیب فراوانی هر (مقدار، کلاس) ای محاسبه می‌شود. با این

حال، همان‌طور که در این دیکشنری نیز مشاهده می‌شود، در صورتی که یک مقدار فقط در یک کلاس ظاهر شود و فراوانی‌اش در کلاس دیگر صفر باشد (مثل B که چهار بار در دیکشنری تکرار شده و هر چهار بار با کلاس ۱ مشاهده شده است و هرگز با کلاس صفر مشاهده نشده است) در این دیکشنری فعلی ثبت نمی‌شود.

برای جلوگیری از این مساله، ابتدا اعضای دیکشنری دوم را به یک لیست تغییر دادیم (چرا که در این قسمت و هم‌چنین در قسمت‌های پیش رو، نیاز بود که به اعضا بر حسب شاخصشان دسترسی پیدا کنیم). و در صورتی که طول لیست (همان دیکشنری دوم) فرد بود، چک کردیم که کدام کلید تنها یک بار در دیکشنری وجود دارد (این کلید، همان کلیدی است که فراوانی‌اش در کلاس غایب صفر است.) و کلید غایب را با فراوانی صفر به لیست اضافه کردیم. سپس لیست را مرتب کردیم که اعضا به ترتیب قرار بگیرند. برای مثال، در پایین لیست مرتب‌شده فیچر اول را مشاهده می‌کنید:

```
C:\Users\Sara\AppData\Local\Programs\Python\Python36\python.exe "E:/CL/Semester 2/AI/HW/HW1/1.py"
[ (('A', 0), 3), (('A', 1), 2), (('B', 0), 0), (('B', 1), 4), (('C', 0), 2), (('C', 1), 3) ]
```

در مرحله بعد میزان خطا باید محاسبه شود. خطای مربوط به هر مقدار در متغیر `error_rates` ذخیره شده و به یک لیست (`f1_error, f2_error, f3_error, f4_error`) الصاق می‌شد. خطای کل فیچر نیز در متغیرهای `t_e1, t_e2, t_e3` و `t_e4` محاسبه می‌شد و به لیست `total_errors` اضافه می‌شد. (لازم به ذکر است که با این که خطای کل است که برای ما اهمیت دارد، تنها به دلیل این که چیزی را از قلم نینداخته باشیم، خطای هر مقدار هر فیچر را نیز در لیست `error_rates` نگاه داشتیم. هرچند عملاً جای دیگری از آن‌ها استفاده نکردیم.)

حال باید کمترین میزان خطای کل را انتخاب کرده و قوانین مربوط به آن فیچر را به عنوان قاعده خود برگزینیم. خطای کل هر فیچر، که در لیست `total_errors` ذخیره شده، به صورت زیر است:

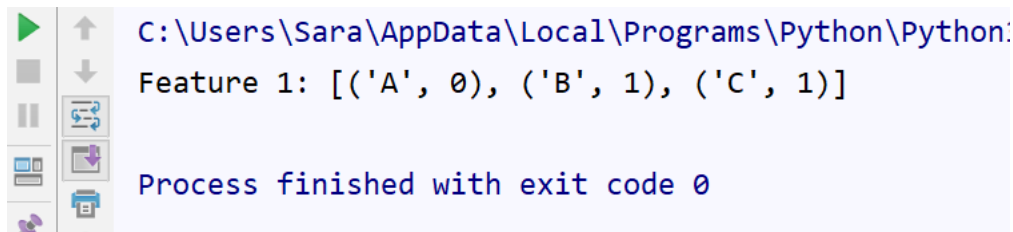
```
C:\Users\Sara\AppData\Local\Programs\Python\Python36\python.exe "E:/CL/Semester 2/AI/HW/HW1/1.py"
[0.2857142857142857, 0.35714285714285715, 0.2857142857142857, 0.35714285714285715]

Process finished with exit code 0
```

همان‌طور که مشاهده می‌شود خطای فیچر اول و فیچر سوم یکسان هستند، بنابراین می‌توان هر یک از این دو فیچر را انتخاب کرد و قاعده نهایی را با آن نوشت. ما در این جا با استفاده از متد `index()` شاخص مینیموم خطا را در لیست به دست آوردیم و به این طریق فیچر منتخب -فیچر اول- برگزیده شد.

همان طور که مشاهده می‌شود، مقدار خطای کل حدود ۰/۲۹ یا $\frac{4}{14} \left(\frac{e1}{\text{تعداد کل}} \right)$ است.

در ادامه، قوانین (که در error_rates قرار داشت که حاوی چهار مجموعه قانون برای چهار فیچر بود) بر اساس فیچر منتخب (فیچر ۱؛ که بر اساس شاخصش به آن دسترسی پیدا کردیم) استخراج شد. در نهایت، خروجی برنامه (قوانین استخراج شده) به شرح زیر است:



```
C:\Users\Sara\AppData\Local\Programs\Python\Python:
Feature 1: [('A', 0), ('B', 1), ('C', 1)]
Process finished with exit code 0
```

برای استفاده راحت‌تر، این قوانین در یک تابع (OneR) قرار داده شدند.

در ادامه نیز سه روش برای گرفتن نمونه تست در نظر گرفته شد: (۱) فرستادن مستقیم یک نمونه به تابع OneR، (۲) تابعی که ورودی را از کاربر بگیرد و (۳) تابعی که نمونه‌های تست را از روی یک فایل اکسل بخواند.

لازم به ذکر است که در صورتی که در نمونه تست مقدار فیچر اول ناموجود^۱ باشد، کلاس پیش‌بینی شده ۰ و ۱ نخواهد بود و کلاس ۱- (مختص داده‌های ناموجود) خواهد بود. همان‌طور که در تابع زیر نیز مشاهده می‌شود:

```
def OneR(x):
    ''' Returns the class of the given instance based on the trained
    model. Returns -1 for test cases whose first value is missing.
    Input: an instance (a tuple with four str elements)
    Output: the predicted class (int)
    '''
    for i in range(len(rules)):
        if x[ind] == rules[i][0]:
            return rules[i][1]
    return -1
```

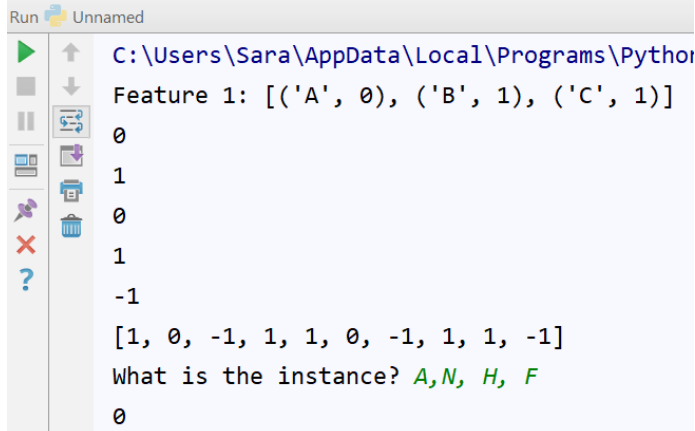
البته، راه‌های بهتری نیز برای مدیریت داده‌های ناموجود وجود دارد؛ برای مثال یک راه این بود که به چنین نمونه‌هایی کلاس آن مقداری نسبت داده شود که در فیچر منتخب در اکثریت است. با این حال، ما فعلاً و برای این تمرین همان شیوه ساده اول را انتخاب کردیم؛ زیرا گمان می‌کردیم شیوه‌های دیگر

^۱. missing

مدیریت داده‌های ناموجود هر یک پیچیدگی‌های خاص خود را داشته باشد و با توجه به این که ما هنوز به آن‌ها احاطه کامل نداریم، ممکن است دچار اشتباه شویم.

در انتها، تصویری از شیوه اجرای برنامه قرار می‌دهیم. همان‌طور که مشاهده می‌شود نمونه‌های تست به هر سه شیوه یادشده گرفته شده‌اند.

```
260 print(OneR(("A", "H", "N", "F")))
261 print(OneR(("B", "C", "H", "T")))
262 print(OneR(("A", "M", "H", "T")))
263 print(OneR(("C", "M", "N", "F")))
264 print(OneR(("?", "C", "N", "F")))
265 print(xlsx_test("test"))
266 print(users_input_test())
267 |
```



Run Unnamed

C:\Users\Sara\AppData\Local\Programs\Python\Python38-64\Scripts\python.exe

Feature 1: [('A', 0), ('B', 1), ('C', 1)]

0

1

0

1

-1

[1, 0, -1, 1, 1, 0, -1, 1, 1, -1]

What is the instance? A,N, H, F

0

منابع:

Ian. H. Witten, Eibe Frank. (2005). *Data Mining : practical machine learning tools and techniques*. Diane Cerra.