

## ۹.

می‌خواهیم یک سیستم مبتنی بر قانون ایجاد کنیم که در آن بتوان با ایجاد یک مجموعه از قوانین عمومی (General) با حداقل تعداد قوانین، خروجی مورد انتظار برای یک سری داده را پیدا کنیم.

برای این منظور فرض کنید داده‌هایی به شکل زیر داریم که هر داده دارای یک بردار ورودی با خصوصیت‌های باینری ( $x_1, x_2, x_3$ ) و یک خروجی کاراکتری ( $y$ ) است. در این مرحله، قوانین را خود ایجاد و وارد سیستم می‌کنیم اما هدف نهایی چنین سامانه‌هایی این است که سیستم توانایی یادگیری قوانین را داشته باشد.

$x_1$	$x_2$	$x_3$	$y$
True	False	False	a
True	True	False	a
False	False	False	b
False	False	True	b
False	True	True	b
False	True	False	c
True	False	True	d

قوانین سامانه همانند نمونه زیر دارای یک بخش شرط (condition) و یک بخش عمل (action) است. هدف این است که بردار ورودی نمونه‌ها با بخش شرط قوانین جور (match) شود و هر قانونی که با نمونه ورودی جور شد، بخش عمل آن به عنوان خروجی مرتبط با نمونه ورودی بازگردانده شود. انتظار این است که خروجی سامانه همانند خروجی مورد انتظار برای نمونه باشد.

بخش شرط قوانین از یک ادبیات سه‌تایی  $\{0, 1, \#\}$  استفاده می‌کند که در آن 0 با مقدار false از نمونه‌ی ورودی، 1 با مقدار true از نمونه‌ی ورودی و # (don't care) با هر دوی مقادیر true و false جور می‌شود.

به خاطر وجود #، ممکن است بیش از یک قانون با یک نمونه ورودی جور شود و هر یک از این قوانین، خروجی متفاوتی را پیشنهاد کند. برای انتخاب خروجی مناسب، نیاز است میزان دقت قوانین را بررسی کنیم. دقت هر قانون مشخص می‌کند قانون مربوطه در مقابل نمونه‌های ورودی تا چه میزان خروجی را درست پیش بینی کرده است. لذا در صورتی که بیش از یک قانون با نمونه ورودی جور شد، خروجی‌ای به عنوان خروجی سیستم انتخاب می‌شود که قوانین دارای آن عمل دارای میانگین وزنی بیشتری باشند.

برای پیش‌بینی صحیح نمونه‌های بالا می‌توان از ۴ قانون زیر استفاده کرد:

	Cond/ act	accuracy
1.	1## / a	0.66
2.	0## / b	0.75
3.	010 / c	1
4.	101 / d	1

اکنون فرض کنید ورودی (true, false, true) به سیستم وارد شود. قوانین ۱ و ۴ با این ورودی جور می‌شود. قانون یک خروجی را a و قانون ۴ خروجی d را پیشنهاد می‌کند. از آنجا که دقت قانون ۴ از قانون ۱ بیشتر است، لذا خروجی قانون ۴ به عنوان خروجی سیستم انتخاب می‌شود.

توجه داشته باشید که در حالت کلی ممکن است قوانین بیشتری با ورودی جور شوند. بنابراین همانطور به پیش‌تر گفته شد لازم است بین قوانینی که خروجی مشترکی را پیشنهاد می‌کنند میانگین وزنی گرفته شود و خروجی با بیشترین میانگین انتخاب شود. به عنوان مثال فرض کنید قوانینی با شرایط زیر با یک ورودی جور شوند.

y	accuracy
a	0.66
b	0.75
b	1
a	1
b	0.5
c	0.25

میانگین وزنی برای هر خروجی به شکل زیر خواهد بود:

y	mean
a	$(0.66 + 1) / 2$
b	$(0.75 + 1 + 0.5) / 3$
c	$0.25/1$

بنابراین a به عنوان خروجی انتخاب می‌شود.

هدف این کارگاه پیاده‌سازی برنامه‌ای است که بتوان مجموعه قوانین را در آن تعریف کرده، با دریافت ورودی خروجی سیستم را نمایش دهد. این سامانه را می‌توان با ۳ کلاس زیر پیاده‌سازی کرد:

- **کلاس Constances:** شامل مقادیر static که برخی ویژگی‌های سیستم از جمله طول بردار نمونه ورودی را مشخص می‌کند.
- **کلاس Rule:** شامل یک بخش condition هم اندازه‌ی بردار ویژگی نمونه‌ها که می‌تواند مقادیر {0, 1, #} بگیرند، یک عمل که پیش‌بینی خروجی مورد انتظار برای نمونه جور شده است و یک مقدار که میزان دقت قانون را مشخص می‌کند. همچنین عملیاتی که جور بودن یک نمونه با قانون را بررسی کند و...
- **کلاس RuleBase:** شامل یک مجموعه از قوانین و عملیاتی برای انتخاب بهترین خروجی برای نمونه ورودی با توجه به دقت قوانین جور شده و...

در پیاده‌سازی خود از موارد تمرین شده در کارگاه ۷ و ۸ برای اجرا و دریافت ورودی‌ها از کاربر کمک بگیرید.