یادگیری ماشین - فهم داده با آمار و ارقام:

مقدمه :

معمولا در زمان کار با پروژه های یادگیری ماشین، دوتا از مهمترین بخش ها به نام های ریاضیات و داده را نادیده می گیریم. دلیل این امر این است که، ما می دانیم یادگیری ماشین یک رویکرد داده محور است، و مدل یادگیری ماشین ما نتایجی به خوبی یا بدی داده ای که ما برایش فراهم کرده ایم، تولید می کند.

در بخش قبلی، در باره چگونگی بارگیری داده CSV درون پروژه یادگیری ماشین خود صحبت کردیم، اما خوب است قبل از بارگذاری داده، آن را بفهمیم. به دو روش می توان داده ها را فهمید، آمار و ارقام(statistics) و تجسم(visualization).

در این بخش، به کمک دستور العمل های پایتون که در ادامه مطرح می شوند، داده های یادگیری ماشین را به کمک آمار و ارقام درک می کنیم.

نگاهی به داده خام:

اولین دستور العمل این است که به داده خام خود نگاه کنید. نگاه به داده خام مهم است زیرا بینشی که پس از نگاه به داده خام به دست می آوریم، شانس ما را برای پیش پردازش بهتر و همچنین مدیریت داده ها برای پروژه های یادگیری ماشین، تقویت می کند.

در ادامه یک اسکریپت پایتون قرار دارد که با استفاده از تابع head() مربوط به Pandas DataFrame روی مجموعه داده دیابتی های Pima Indians پیاده سازی شده است، تا برای درک بهتر آن به ۵۰ سطر اول آن نگاهی بیاندازیم.

مثال:

```
from pandas import read_csv
path = r"C:\pima-indians-diabetes.csv"
```

```
headernames = ['preg', 'plas', 'pres', 'skin', 'test',
'mass', 'pedi', 'age', 'class']
data = read_csv(path, names=headernames)
print(data.head(50))
```

خروجی:

```
preg plas  pres skin test   mass pedi   age      class
0      6     148   72    35      0     33.6    0.627     50
1
1      1      85   66    29      0     26.6    0.351     31
0
2      8     183   64     0      0     23.3    0.672     32
1
3      1      89   66    23     94     28.1    0.167     21
0
4      0     137   40    35    168     43.1    2.288     33
1
5      5     116   74     0      0     25.6    0.201     30
0
6      3      78   50    32     88     31.0    0.248     26
1
7     10     115    0     0      0     35.3    0.134     29
0
8      2     197   70    45    543     30.5    0.158     53
1
9      8     125   96     0      0      0.0    0.232     54
1
10     4     110   92     0      0     37.6    0.191     30
0
11    10     168   74     0      0     38.0    0.537     34
1
12    10     139   80     0      0     27.1    1.441     57
0
13     1     189   60    23    846     30.1    0.398     59
1
14     5     166   72    19    175     25.8    0.587     51
1
15     7     100    0     0      0     30.0    0.484     32
1
16     0     118   84    47    230     45.8    0.551     31
1
17     7     107   74     0      0     29.6    0.254     31
1
```

18	1	103	30	38	83	43.3	0.183	33
0								
19	1	115	70	30	96	34.6	0.529	32
1								
20	3	126	88	41	235	39.3	0.704	27
0								
21	8	99	84	0	0	35.4	0.388	50
0								
22	7	196	90	0	0	39.8	0.451	41
1								
23	9	119	80	35	0	29.0	0.263	29
1								
24	11	143	94	33	146	36.6	0.254	51
1								
25	10	125	70	26	115	31.1	0.205	41
1								
26	7	147	76	0	0	39.4	0.257	43
1								
27	1	97	66	15	140	23.2	0.487	22
0								
28	13	145	82	19	110	22.2	0.245	57
0								
29	5	117	92	0	0	34.1	0.337	38
0								
30	5	109	75	26	0	36.0	0.546	60
0								
31	3	158	76	36	245	31.6	0.851	28
1								
32	3	88	58	11	54	24.8	0.267	22
0								
33	6	92	92	0	0	19.9	0.188	28
0								
34	10	122	78	31	0	27.6	0.512	45
0								
35	4	103	60	33	192	24.0	0.966	33
0								
36	11	138	76	0	0	33.2	0.420	35
0								
37	9	102	76	37	0	32.9	0.665	46
1								
38	2	90	68	42	0	38.2	0.503	27
1								
39	4	111	72	47	207	37.1	1.390	56
1								

```
40      3      180    64     25      70      34.0     0.271      26
0
41      7      133    84      0       0      40.2     0.696      37
0
42      7      106    92     18       0      22.7     0.235      48
0
43      9      171   110     24     240      45.4     0.721      54
1
44      7      159    64      0       0      27.4     0.294      40
0
45      0      180    66     39       0      42.0     1.893      25
1
46      1      146    56      0       0      29.7     0.564      29
0
47      2       71    70     27       0      28.0     0.586      22
0
48      7      103    66     32       0      39.1     0.344      31
1
49      7      105     0      0       0       0.0     0.305      24
0
```

با مشاهده خروجی بالا متوجه می شویم که ستون اول، شماره سطر را ارائه می کند که برای ارجاع به یک مشاهده خاص می تواند بسیار مفید باشد.

بررسی ابعاد داده:

خوب است بدانیم چه مقدار داده، در قالب سطرها و ستون ها برای پروژه یادگیری ماشین خود داریم. دلایل آن عبارتند از: ۱- فرض کنید سطر ها و ستون های بسیار زیادی داشته باشیم، در این صورت اجرای الگوریتم و آموزش مدل بسیار طول خواهد کشید. ۲- فرض کنید سطر ها و ستون های بسیار کمی داشته باشیم، در این صورت داده کافی برای آموزش مدل وجود ندارد.

کد زیر یک اسکریپت پایتون است که با چاپ خصیصه شکل(shape)، روی Pandas Data Frame پیاده سازی شده است. برای بدست آوردن تعداد کل سطرها و ستون ها، آن را روی مجموعه داده iris پیاده سازی می کنیم.

مثال:

```
from pandas import read_csv
path = r"C:\iris.csv"
data = read_csv(path)
print(data.shape)
```

خروجی:

با توجه به خروجی به سادگی متوجه می شویم که مجموعه داده iris که مورد استفاده است، ۱۵۰ سطر و ۴ ستون دارد.

به دست آوردن نوع داده هر خصیصه: خوب است نوع داده هر یک از خصیصه ها را بدانیم. دلیل این امر این است که، ممکن است بر حسب ضرورت، گاهی نیاز به تبدیل یک نوع داده به دیگری داشته باشیم. برای مثال، ممکن است برای نمایش مقادیر طبقه بندی شده(categorial) یا ترتیبی(ordinal) نیاز به تبدیل رشته(string) به ممیز شناور(floating point) یا اعداد صحیح(int) داشته باشیم. ما می توانیم با نگاه به داده خام، در باره نوع داده خصیصه ها اطلاعاتی به دست آوریم، اما روش دیگر استفاده از خصیصه ***dtypes*** مربوط به Pandas DataFrame است. با کمک خصیصه ***dtypes*** می توانیم نوع داده هر خصیصه را طبقه بندی کنیم. این امر با کمک اسکریپت پایتون که در ادامه آمده است، قابل درک می شود.

مثال:

```
from pandas import read_csv
path = r"C:\iris.csv"
data = read_csv(path)
print(data.dtypes)
```

خروجی:

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
dtype: object
```

با توجه به خروجی بالا، به سادگی می توان نوع داده هر خصیصه را فهمید.

خلاصه آماری داده:

درباره دستور العمل پایتون جهت به دست آوردن شکل داده(تعداد سطر ها و ستون ها) صحبت کردیم، اما در بسیاری از مواقع نیاز داریم تا خلاصه های مربوط به شکل داده را مرور کنیم. این امر با کمک تابع ***describe()*** مربوط به Pandas DataFrame که ۸ خصیصه آماری زیر را برای هر ویژگی داده فراهم می کند، امکان پذیر می شود.

- Count
- Mean
- Standard Deviation
- Minimum Value
- Maximum value
- 25%
- Median i.e. 50%
- 75%

مثال:

```
from pandas import read_csv
from pandas import set_option
path = r"C:\pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test',
'mass', 'pedi', 'age', 'class']
data = read_csv(path, names=names)
set_option('display.width', 100)
set_option('precision', 2)
print(data.shape)
print(data.describe())
```

خروجی:

```
(768, 9)
       preg     plas     pres    skin     test     mass
pedi      age       class
count  768.00   768.00   768.00  768.00   768.00   768.00
768.00    768.00    768.00
mean     3.85   120.89    69.11   20.54    79.80    31.99
0.47     33.24      0.35
std      3.37    31.97    19.36   15.95   115.24     7.88
0.33     11.76      0.48
min      0.00     0.00     0.00    0.00     0.00     0.00
0.08     21.00      0.00
```

```
25%      1.00    99.00    62.00    0.00     0.00    27.30
0.24    24.00     0.00
50%      3.00   117.00    72.00   23.00    30.50    32.00
0.37    29.00     0.00
75%      6.00   140.25    80.00   32.00   127.25    36.60
0.63    41.00     1.00
max     17.00   199.00   122.00   99.00   846.00    67.10
2.42    81.00     1.00
```

با توجه به خروجی بالا، می توان خلاصه‌ی آماری داده‌ی مربوط به مجموعه داده دیابتی های Pima Indian را به همراه شکل داده، مشاهده کرد.

مرور توزیع کلاس:

آمار و ارقام توزیع کلاس (Class distribution statistics) در مسائل طبقه بندی، جایی که نیاز داریم تعادل مقادیر کلاس را بدانیم، مفید است. دانستن توزیع مقدار(value) کلاس مهم است زیرا اگر توزیع بسیار نا متعادل کلاس را داشته باشیم، مانند کلاسی که مشاهدات بسیار بیشتری نسبت به کلاس دیگر داشته باشد، در نتیجه ممکن است در مرحله آماده سازی داده برای پروژه یادگیری ماشین، نیازمند به مدیریت خاصی باشد. به سادگی می توان در پایتون و با کمک Pandas DataFrame، توزیع کلاس را به دست آورد.

مثال:

```
from pandas import read_csv
path = r"C:\pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test',
'mass', 'pedi', 'age', 'class']
data = read_csv(path, names=names)
count_class = data.groupby('class').size()
print(count_class)
```

خروجی:

```
Class
0 500
1 268
dtype: int64
```

با توجه به خروجی بالا، به طور واضح می توان دید که تعداد مشاهدات کلاس 0 تقریبا دو برابر مشاهدات کلاس 1 است.

مرور همبستگی(correlation) بین ویژگی ها:

رابطه بین دو متغیر، همبستگی نام دارد. در آمار ، متداول ترین متد برای محاسبه همبستگی، ضریب همبستگی پیرسون (Pearson's Correlation Coefficient) است. این ضریب، سه مقدار زیر را می تواند داشته باشد.

مقدار ضریب = 1 بیانگر همبستگی کاملا مثبت (full **positive**) بین متغیر ها است.

مقدار ضریب = 1- بیانگر همبستگی کاملا منفی (full **negative**) بین متغیر ها است.

مقدار ضریب = 0 بیانگر این است که مطلقا هیچ همبستگی بین متغیر ها وجود ندارد.

خوب است همیشه پیش از استفاده از مجموعه داده مورد نظر در پروژه یادگیری ماشین خود، همبستگی های دوطرفه بین ویژگی های مجموعه داده را مرور کنیم. دلیل این امر این است که برخی از الگوریتم های یادگیری ماشین مانند رگرسیون خطی (linear regression) و رگرسیون لجستیک (logistic regression) در صورت وجود ویژگی ها با همبستگی بالا، ضعیف عمل خواهند کرد. در پایتون، به سادگی می توانیم ماتریس همبستگی مربوط به ویژگی های مجموعه‌ی داده را با کمک تابع *corr()* روی Pandas DataFrame محاسبه کنیم.

مثال:

```
from pandas import read_csv
from pandas import set_option
path = r"C:\pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test',
'mass', 'pedi', 'age', 'class']
data = read_csv(path, names=names)
set_option('display.width', 100)
set_option('precision', 2)
```

```
correlations = data.corr(method='pearson')
print(correlations)
```

خروجی:

```
preg   plas   pres   skin   test   mass   pedi   age
class
preg   1.00   0.13   0.14  -0.08  -0.07   0.02  -
0.03   0.54   0.22
plas   0.13   1.00   0.15   0.06   0.33   0.22   0.14
0.26   0.47
pres   0.14   0.15   1.00   0.21   0.09   0.28   0.04
0.24   0.07
skin  -0.08   0.06   0.21   1.00   0.44   0.39   0.18
-0.11   0.07
test  -0.07   0.33   0.09   0.44   1.00   0.20   0.19
-0.04   0.13
mass   0.02   0.22   0.28   0.39   0.20   1.00   0.14
0.04   0.29
pedi  -0.03   0.14   0.04   0.18   0.19   0.14   1.00
0.03   0.17
age    0.54   0.26   0.24  -0.11  -0.04   0.04   0.03
1.00   0.24
class  0.22   0.47   0.07   0.07   0.13   0.29   0.17
0.24   1.00
```

ماتریس موجود در خروجی بالا، همبستگی بین همه جفت ویژگی های مجموعه داده را ارائه می کند.

بررسی انحراف(skew) توزیع ویژگی:

انحراف(Skewness) ممکن است به صورت توزیع گوسی(Gaussian) تعریف شود، که به یک سمت یا دیگری یا چپ یا راست منحرف و جابه جا شده باشد. به دلایل زیر، مرور انحراف ویژگی ها از کار های مهم است.

در صورت وجود انحراف در داده، ملزم هستیم در مرحله آماده سازی داده، تصحیح انجام دهیم تا دقت بیشتری از مدل خود به دست بیاوریم.

اکثر الگوریتم های یادگیری ماشین، فرض می کنند که داده دارای توزیع گوسی است، به عنوان مثال توزیع گوسی نرمال یا منحنی زنگی(زنگوله ای).

در پایتون، با استفاده از تابع **skew()** مربوط به Pandas DataFrame ، می توان به سادگی انحراف هر ویژگی را محاسبه کرد.

مثال:

```
from pandas import read_csv
path = r"C:\pima-indians-diabetes.csv"
names = ['preg', 'plas', 'pres', 'skin', 'test',
'mass', 'pedi', 'age', 'class']
data = read_csv(path, names=names)
print(data.skew())
```

خروجی:

```
preg   0.90
plas   0.17
pres  -1.84
skin   0.11
test   2.27
mass  -0.43
pedi   1.92
age    1.13
class  0.64
dtype: float64
```

بر اساس خروجی بالا، انحراف مثبت یا منفی مشاهده می شود. اگر مقدار به صفر نزدیک تر باشد، در نتیجه انحراف کمتری را نشان می دهد.