Shahid Beheshti

University

گزارش تمرین سری سوم
واحد درسی داده کاوی دوره الکترونیکی

جناب آقای دکتر خردپیشه

رحیم اکبری 99422028

00/03/14

•الگوریتم های SVM از مجموعه ای از توابع ریاضی که به عنوان کرنل تعریف می شوند، استفاده می کنند. وظیفه کرنل این است که داده ها را به عنوان ورودی گرفته و آن ها را به شکل مورد نیاز تبدیل کند. الگوریتم های مختلفSVM ، از انواع مختلف توابع کرنل استفاده می کنند. این توابع می توانند انواع متفاوتی داشته باشند.

توابع کرنل ، برای داده های ترتیبی ، نمودار ها ، متن ها ، تصاویر و همچنین بردار ها معرفی می شوند. پرکاربردترین نوع تابع کرنل، RBF است. زیرا دارای پاسخ محلی و متناهی در کل بازه محور x است.

توابع کرنل ، ضرب داخلی بین دو نقطه در یک فضای ویژگی مناسب را برمی گردانند. بنابراین ، با هزینه محاسباتی کم، حتی در فضاهای با ابعاد بالا، مفهومی از شباهت را تعریف می کنند.

کرنل چندجملهای: این کرنل در پردازش تصویر پرکاربرد است. معادله آن به صورت زیر است :

$$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$$

که در آن d درجه چند جمله ای است.

کرنل rbf: این کرنلی برای اهداف عمومی کلربرد دارد. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود نداشته باشد، مورد استفاده قرار می گیرد. معادله آن به صورت زیر است :

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

کرنل سیگموید: می توان این کرنل را در شبکه های عصبی مورد استفاده قرار داد. معادله مربوط به آن عبارت است از :

$$k(x, y) = \tanh(\alpha x^T y + c)$$

کرنل گاوسی: این یک کرنل برای اهداف عمومی است. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود ندارد استفاده می شود. معادله آن به صورت زیر است :

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

3. ابتدا دیتاست را به دو قسمت دادههای آموزش و تست تقسیم میکنیم.

سپس مدل svm را با کرنل linear بر روی دادههای آموزش اجرا کرده و سپس خروجی آن را برای دادههای تست ارزیابی میکنیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[145   0   0   0]
 [  3 145   7   0]
 [  0   3 140   3]
 [  0   0   1 153]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.9716666666666667
```

دقت مدل 97 درصد است که دقت بسیار خوبی میباشد.

پس از آن مدل svm را با کرنل rbf بر روی داده¬های آموزش اجرا کرده و سپس خروجی آن را برای داده-های تست ارزیابی میکنیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[144   1   0   0]
 [ 10 141   4   0]
 [  0   6 134   6]
 [  0   0   2 152]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.9516666666666667
```

دقت مدل ۹۵ درصد است که دقت بسیار خوبی است، اما پایینتر از حالت قبلی است.

پس از آن مدلsvm را با کرنل sigmoid بر روی داده¬های آموزش اجرا کرده و سپس خروجی آن را برای داده-های تست ارزیابی می¬کنیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[ 87   0   0  58]
 [ 12  10   0 133]
 [  2  25   0 119]
 [ 25 123   0   6]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.17166666666666666
```

دقت مدل ۱۷ درصد است که دقت بسیار ضعیفی است.

پس از آن مدلsvm را با کرنل poly و پارامتر درجه ۳ بر روی داده¬های آموزش اجرا کرده و سپس خروجی آن را برای داده-های تست ارزیابی می¬کنیم.

ماتریس پریشانی بصورت زیر است

```
confusion matrix:
 [[145   0   0   0]
 [ 10 140   5   0]
 [  0   3 139   4]
 [  0   0   2 152]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.96
```

دقت مدل ۹۶ درصد است که دقت بسیار خوبی است.

سپس آن مدل svm را با کرنل poly و پارامتر درجه ۵ بر روی داده¬های آموزش اجرا کرده و سپس خروجی آن را برای داده-های تست ارزیابی می¬کنیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[145   0   0   0]
 [ 13 135   7   0]
 [  0   2 142   2]
 [  0   0   4 150]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.9533333333333334
```

دقت مدل 95 درصد است که دقت بسیار خوبی است و تقریبا مشابه حالت قبل است.

4. در دیتاستهایی که بصورت خطی جداییپذیر هستند، soft-margin مناسبتر است، به این خاطر که اگر از hard-margin استفاده کنیم، وجود یک داده پرت میتواند در عملکرد الگوریتم طبقهبندی ما تاثیر منفی داشته باشد.

برای بررسی soft-margin و hard-margin باید پارامتر C را تغییر میدهیم. هرچه پارامتر C را افزایش دهیم از هارد مارجین به سمت سافت مارجین حرکت میکنیم.

ابتدا مقدار C را برابر 0.1 قرار میدهیم. ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[144   1   0   0]
 [ 15 129  11   0]
 [  0  12 126   8]
 [  0   0  13 141]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.9
```

سپس مقدارC را برابر ۰.۵ قرار می¬دهیم. ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[144   1   0   0]
 [  8 142   5   0]
 [  0   7 130   9]
 [  0   0   3 151]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.945
```

سپس مقدارC را برابر ۵ قرار میدهیم. ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[144   1   0   0]
 [  7 142   6   0]
 [  0   2 140   4]
 [  0   0   4 150]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.96
```

پس از آن مقدارC را برابر ۱۰ قرار میدهیم. ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[143   2   0   0]
 [  4 144   7   0]
 [  0   2 140   4]
 [  0   0   4 150]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.9616666666666667
```

نتایج بالا نشان میدهد که برای دیتاست ما روش **soft-margin** مناسب است.

.5

(آ) در ابتدا از روش binning استفاده میکنیم و سه دسته برای قدرت باتری در نظر میگیریم. دیتاست به شکل زیر میشود.

dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_width	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range	battery
0	1	0	7	0.6	188	2	...	756	2549	9	7	19	0	0	1	1	0
1	0	1	53	0.7	136	3	...	1988	2631	17	3	7	1	1	0	2	0
1	2	1	41	0.9	145	5	...	1716	2603	11	2	9	1	1	0	2	0
0	0	0	10	0.8	131	6	...	1786	2769	16	8	11	1	0	0	2	0
0	13	1	44	0.6	141	2	...	1212	1411	8	2	15	1	1	0	1	2

حال فیچر قبلی را حذف کرده و مدلسازی را دوباره انجام میدهیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[133  21   0   0]
 [ 13 112  25   0]
 [  0  20  99  23]
 [  0   0  19 135]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.7983333333333333
```

نتیجه بالا نشان میدهد که استفاده روش **binning** باعث کاهش عملکرد مدل میشود.

(ب) در دیتاست گاهی با ستونهایی مواجه میشویم که در آنها اعداد اشاره به گونه یا نوع خاصی دارد، این ستونها کتگوریکال هستند. اگر در دیتاست ستونهای عددی نیز وجود داشته باشد، در این صورت مدل یادگیری ماشین گیج میشود که کدامیک عددی است و کدامیک کتگوریکال. برای حل این مشکل و بالا بردن کارایی مدل، از روش one hot encoding استفاده میکنیم، به این صورت که به ستونهای کتگوریکال یک ماتریس اختصاص میدهد. تعداد سطرهای هر ماتریس برابر با تعداد گونههای آن ستون است. در هر سلول مقدار 1 یعنی وجود داشتن آن گونه و مقدار صفر یعنی عدم وجود آن گونه خاص.

دیتاست پس از استفاده از روش one hot encoding به شکل زیر در میاید.

er	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	touch_screen	wifi	price_range	battery	0	1	2	3	4	5
12	0	2.2	0	1	0	7	0.6	188	2	...	0	1	1	0	1.0	0.0	0.0	0.0	0.0	1.0
21	1	0.5	1	0	1	53	0.7	136	3	...	1	0	2	0	0.0	1.0	1.0	1.0	1.0	0.0
53	1	0.5	1	2	1	41	0.9	145	5	...	1	0	2	0	0.0	1.0	1.0	1.0	1.0	0.0
15	1	2.5	0	0	0	10	0.8	131	6	...	0	0	2	0	0.0	1.0	0.0	0.0	1.0	0.0
21	1	1.2	0	13	1	44	0.6	141	2	...	1	0	1	2	0.0	1.0	0.0	1.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
94	1	0.5	1	0	1	2	0.8	106	6	...	1	0	0	0	0.0	1.0	1.0	1.0	1.0	0.0
65	1	2.6	1	0	0	39	0.2	187	4	...	1	1	2	2	0.0	1.0	1.0	0.0	1.0	1.0
11	0	0.9	1	1	1	36	0.7	108	8	...	1	0	3	2	1.0	0.0	1.0	1.0	1.0	0.0
12	0	0.9	0	4	1	46	0.1	145	5	...	1	1	0	1	1.0	0.0	0.0	1.0	1.0	1.0
10	1	2.0	1	5	1	45	0.9	168	6	...	1	1	3	0	0.0	1.0	1.0	1.0	1.0	1.0

حال فیچرهای قبلی را حذف کرده و مدلسازی را دوباره انجام میدهیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[142   1   0   0]
 [  8 147   0   0]
 [  0   8 136  12]
 [  0   0   4 142]]
```

دقت مدل بصورت زیر است.

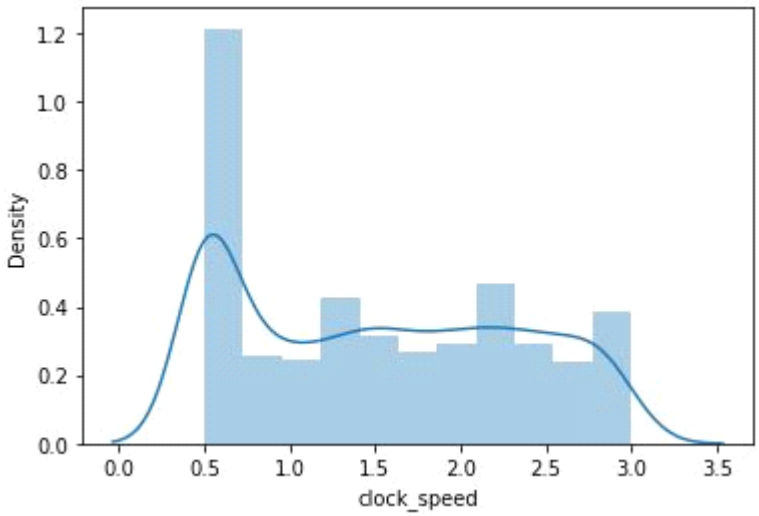
```
accuracy score =  0.945
```

برای دیتاست ما این روش تاثیر چندانی در دقت مدل ندارد و تقریبا مشابه حالتی است که مقدار **C** را **0.5** در نظر گرفتیم.

(ج) در دیتاستهایی که مقادیر ستونها از توزیع نرمان پیروی نمیکند، میتوانیم با استفاده از تبدیل لگاریتمی، نمایی و یا سایر تبدیلها آنها را نرمال کنیم.

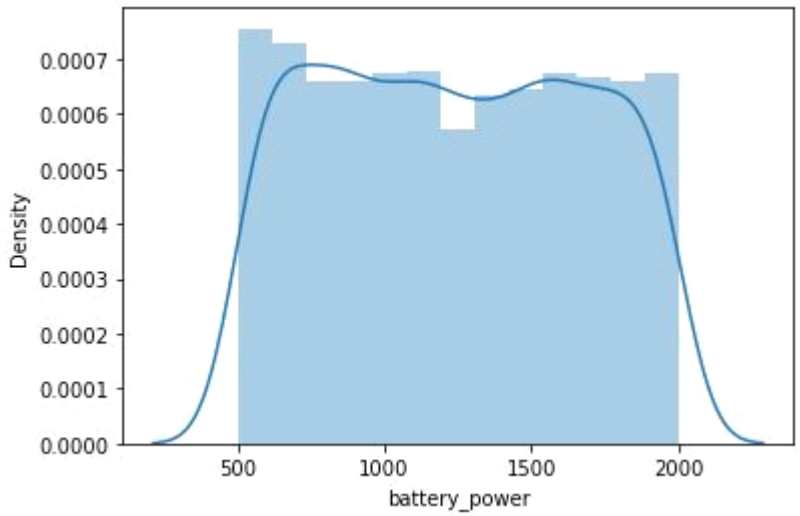
در اینجا چند ستونی که دارای مقدار پیوسته هستند را بررسی میکنیم.

نمودار clock_speed بصورت زیر است.



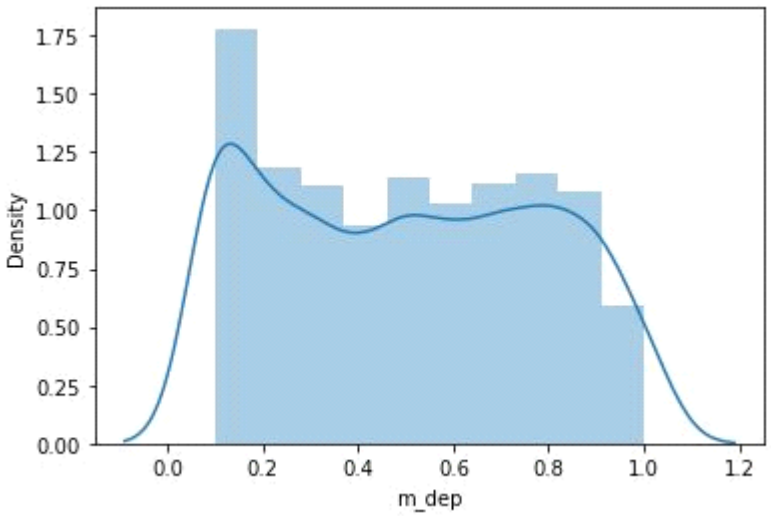
نمودار نشان میدهد که توزیع دادهها در این ستون نرمال نیست. تابع لگاریتمی را بر روی آن پیاده میکنیم.

نمودار battery_power بصورت زیر است.



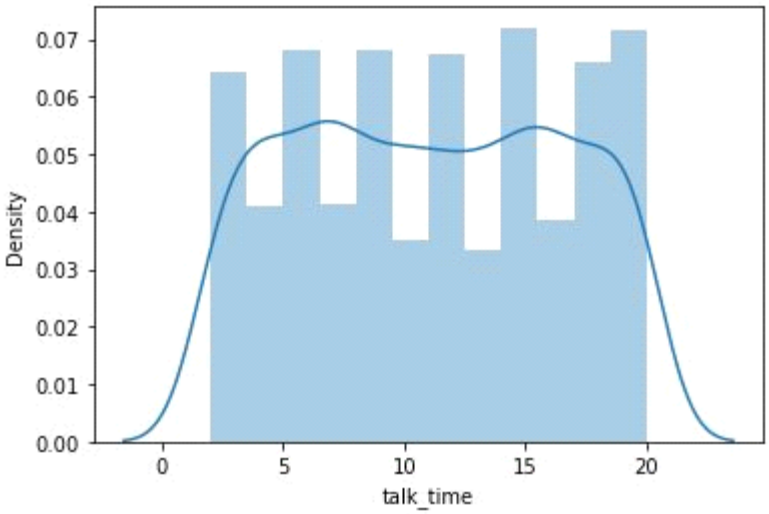
نمودار نشان می¬دهد که توزیع داده¬ها در این ستون نرمال نیست. تابع لگاریتمی را بر روی آن پیاده می¬کنیم.

نمودار m_dep بصورت زیر است.



نمودار نشان می¬دهد که توزیع داده¬ها در این ستون نرمال نیست. تابع لگاریتمی را بر روی آن پیاده می¬کنیم.

نمودار talk_time بصورت زیر است.



نمودار نشان میدهد که توزیع داده ها در این ستون نرمال نیست. تابع لگاریتمی را بر روی آن پیاده می¬کنیم.

دیتاست به شکل زیر تغییر میکند.

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_width	ram	sc_h	sc_w	talk_time	three_g
0	9.717676	0	1.137504	0	1	0	7	-0.736966	188	2	...	756	2549	9	7	4.247928	0
1	9.995767	1	-1.000000	1	0	1	53	-0.514573	136	3	...	1988	2631	17	3	2.807355	1
2	9.136991	1	-1.000000	1	2	1	41	-0.152003	145	5	...	1716	2603	11	2	3.169925	1
3	9.264443	1	1.321928	0	0	0	10	-0.321928	131	6	...	1786	2769	16	8	3.459432	1
4	10.830515	1	0.263034	0	13	1	44	-0.736966	141	2	...	1212	1411	8	2	3.906891	1

پس از تبدیل ستونها دادهها را به دو قسمت آموزش و تست تقسیم کرده و مدلسازی را انجام میدهیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[124  21   0   0]
 [ 17 110  21   0]
 [  0  24 108  17]
 [  0   0  28 130]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.7866666666666666
```

استفاده از این روش در دیتاست ما باعث کاهش دقت مدل میشود.

(د) فیچر مساحت را به دیتاست اضافه میکنیم. دیتاست به شکل زیر میشود.

dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	ram	sc_h	sc_w	talk_time	three_g	touch_screen	wifi	price_range	battery	area
0	1	0	7	-0.736966	188	2	...	2549	9	7	4.247928	0	0	1	1	0	15120
1	0	1	53	-0.514573	136	3	...	2631	17	3	2.807355	1	1	0	2	0	1799140
1	2	1	41	-0.152003	145	5	...	2603	11	2	3.169925	1	1	0	2	0	2167308
0	0	0	10	-0.321928	131	6	...	2769	16	8	3.459432	1	0	0	2	0	2171776
0	13	1	44	-0.736966	141	2	...	1411	8	2	3.906891	1	1	0	1	2	1464096

سپس مدلسازی را انجام داده و نتایج را بررسی میکنیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[110   0   0  46]
 [106   0   0  49]
 [ 94   0   0  50]
 [ 80   0   0  65]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.2916666666666667
```

نتیجه بالا نشان میدهد که اضافه کردن فیچر مساحت، به شدت دقت مدل را کاهش میدهد.

حال همه روشهای قسمت ۵ را با هم اعمال کرده و نتایج را بررسی میکنیم.

دیتاست به شکل زیر میشود.

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	wifi	price_range	battery	0	1	2	3	4
0	842	0	1.137504	0	1	0	7	-0.736966	188	2	...	1	1	0	1.0	0.0	0.0	0.0	0.0
1	1021	1	-1.000000	1	0	1	53	-0.514573	136	3	...	0	2	0	0.0	1.0	1.0	1.0	1.0
2	563	1	-1.000000	1	2	1	41	-0.152003	145	5	...	0	2	0	0.0	1.0	1.0	1.0	1.0
3	615	1	1.321928	0	0	0	10	-0.321928	131	6	...	0	2	0	0.0	1.0	0.0	0.0	1.0
4	1821	1	0.263034	0	13	1	44	-0.736966	141	2	...	0	1	2	0.0	1.0	0.0	1.0	1.0

سپس مدلسازی را انجام داده و نتایج را بررسی میکنیم.

ماتریس پریشانی بصورت زیر است.

```
confusion matrix:
 [[119   0   0  28]
 [107   0   0  51]
 [ 98   0   0  43]
 [ 88   0   0  66]]
```

دقت مدل بصورت زیر است.

```
accuracy score =  0.30833333333333335
```

نتیجه بالا نشان میدهد که اعمال همه روشهای قسمت **5** باهم باعث کاهش شدید کارایی مدل میشود.