تمرین سوم عملی درس یادگیری ماشین

سید علیرضا طباطبائیان

توجه 1 : جهت مشاهده کد و نتایج ، بر روی لینک گوگل COLAB زیر کلیک کنید . (جهت اجرای دوباره میبایست کد را در کولب خود کپی کرده و فایل دیتا موجود در فایل زیپ را نیز آپلود کنید و کد را ران کنید)

https://colab.research.google.com/drive/1x5w0upadXLLyklTsBG2bGO2ElWRyKkTk#scrollTo=Q6ClcSIocS8W

توجه 2 : دستورات مهم دارای کامنت میباشند و توضیحات تکمیلی در زیر آورده شده است.

قسمت الف) ابتدا کتابخانه های مورد نیاز را به برنامه اضافه کردیم . سپس فایل مورد نظر با عنوان iris.data را خواندیم . سپس داده های هر 3 کلاس را به دو دسته 70 درصد آموزش و 30 درصد تست تقسیم کردیم. سپس داده های آموزش هر دسته را به هم پیوند زدیم. همچنین برای داده های تست نیز همین روند را تکرار کردیم. در مرحله بعد ، میانگین و واریانس هر کلاس را حساب کردیم و سپس سه تابع گوسی را با میانگین و واریانس هر کلاس ساختیم و سپس احتمال هر سمپل تست را برای Maximum Likelihood پیدا کردیم و کلاس آن را با توجه به ماکسیمم شدن احتمال فیچر به شرط کلاس پیدا کردیم و کلاس آن را تشخیص دادیم . سپس صحت و ماتریس سردرگمی را نیز پیدا کردیم. سپس همین فرایند را برای دادگان آموزش نیز تکرار کردیم که صحت و ماتریس سردرگمی آنها در زیر قابل مشاهده است . <u>این مراحل 2 بار انجام شده و میانگین آنها در زیر قابل مشاهده است</u> :

```
Maximum Likelihood Accuracy on Test Data is (percentile) : 95.56

Maximum Likelihood Confusion Matrix on Test Data is (percentile) :

 [[33.33  0.     0.   ]
 [ 0.    30.     3.33]
 [ 0.     1.11 32.22]]
```

```
Maximum Likelihood Accuracy on Train Data is (percentile) : 98.57

Maximum Likelihood Confusion Matrix on Train Data is (percentile) :

 [[33.33  0.     0.   ]
 [ 0.    31.9   1.43]
 [ 0.     0.    33.33]]
```

قسمت ب) در قسمت بعد ، برای پیاده سازی K-Fold ، ابتدا داده های هر کلاس را به 4 دسته تقسیم کردیم و سپس برای هر کلاس ، در هر مرحله ، 3 تا از دسته ها را یکی کرده و دیگری را برای تست قرار دادیم و دوباره بر روی هر بخش ، مراحل بالا را تکرار کردیم و نتایج در زیر قابل مشاهده است :

```
K-Fold Maximum Likelihood Accuracy on Test Data is (percentile) : 97.97

K-Fold Maximum Likelihood Confusion Matrix on Test Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   32.    1.34]
 [ 0.    0.69 32.64]]
```

```
K-Fold Maximum Likelihood Accuracy on Train Data is (percentile) : 98.22

K-Fold Maximum Likelihood Confusion Matrix on Train Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   32.    1.33]
 [ 0.    0.44 32.89]]
```

قسمت پ (الف) : در این قسمت همانند قسمت الف پیش رفتیم و تنها تفاوت آن بود که درایه های غیر قطری ماتریس کوواریانس را برابر صفر قرار دادیم که نتایج آن در زیر قابل مشاهده است :

<u>این مراحل 2 بار انجام شد و میانگین آنها در زیر قابل مشاهده است :</u>

```
Diagonal Maximum Likelihood Accuracy on Test Data is (percentile) : 97.78

Diagonal Maximum Likelihood Confusion Matrix on Test Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   32.22  1.11]
 [ 0.    1.11 32.22]]
```

```
Diagonal Maximum Likelihood Accuracy on Train Data is (percentile) : 95.24

Diagonal Maximum Likelihood Confusion Matrix on Train Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   30.95  2.38]
 [ 0.    2.38 30.95]]
```

قسمت پ(ب) : در این قسمت همانند قسمت الف پیش رفتیم و تنها تفاوت آن بود که درایه های غیر قطری ماتریس کوواریانس را برابر صفر قرار دادیم که نتایج آن در زیر قابل مشاهده است :

```
Diagonal K-Fold Maximum Likelihood Accuracy on Test Data is (percentile) : 95.99

Diagonal K-Fold Maximum Likelihood Confusion Matrix on Test Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   31.36  1.98]
 [ 0.    2.03 31.3 ]]
```

```
Diagonal K-Fold Maximum Likelihood Accuracy on Train Data is (percentile) : 95.78

Diagonal K-Fold Maximum Likelihood Confusion Matrix on Train Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   31.56  1.78]
 [ 0.    2.44 30.89]]
```

قسمت ت) در این قسمت ، به علت استفاده از الگوریتم نایو ، میبایست احتمال ضرب هر فیچر به شرط کلاس در سایر فیچر به شرط کلاس ها را ماکسیمم کنیم پس باید برای هر فیچر از هر کلاس ، یکبار تخمین گوسی بزنیم و آنها را در هم ضرب کنیم و بهترین کلاس که برابر بیشترین احتمال است را بدست اوریم. همچنین در بخش نایو چون احتمال هر کلاس برابر است ، آن را محاسبه نکردیم .
این مراحل 2 بار انجام شد و میانگین آنها در زیر قابل مشاهده است :

```
Naive Accuracy on Test Data is (percentile) : 97.78

Naive Confusion Matrix on Test Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   32.22  1.11]
 [ 0.    1.11 32.22]]
```

```
Naive Accuracy on Train Data is (percentile) : 95.24

Naive Confusion Matrix on Train Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   30.95  2.38]
 [ 0.    2.38 30.95]]
```

قسمت ث) با ترکیب قسمت ب و ت یعنی استفاده همزمان از نایو و K-Fold **به نتایج زیر رسیدیم :**

```
Naive K-Fold Accuracy on Test Data is (percentile) : 95.99

Naive K-Fold Confusion Matrix on Test Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   31.36  1.98]
 [ 0.    2.03 31.3 ]]
```

```
Naive K-Fold Accuracy on Train Data is (percentile) : 95.78

Naive K-Fold Confusion Matrix on Train Data is (percentile) :

 [[33.33  0.    0.  ]
 [ 0.   31.56  1.78]
 [ 0.    2.44 30.89]]
```

مقایسه نتایج :

مقایسه اول) قسمت K-Fold **نتایج بهتری از قسمت الف داشت .**

مقایسه دوم) نتایج قسمت پ(الف) با قسمت ت و نتایج قسمت پ(ب) با نتایج قسمت ث یکسان شد که منطقی است زیرا ماتریس قطری واریانس حکم همان نایو بیز را دارد .

مقایسه سوم) کیفیت نایو کمی بهتر از الف شد اما در تکرار های مختلف ، گاهی برعکس میشود .

مقایسه چهارم) کیفیت نایو با K-Fold **بدتر از** K-Fold **معمولی است اما در تکرار های مختلف ، گاهی برعکس میشود .**

پاسخ سوالات :

سوال : بهترین طبقه بندی کننده کدام است ؟

طبقه بندی کننده K-Fold **بهتر از بقیه ظاهر شد اما اختلاف چشمگیری با سایرین ندارد .**

اثبات ریاضی و دلیل مساوی شدن قسمت الف با پ(الف) : (سوال امتیازی)

ابتدا یک توزیع پیوسته گوس توأم برای ۲ متغیر تعریف می کنیم که کوواریانس آنها قطری باشد:

سپس استقلال آنها را ثابت می کنیم. به همین صورت برای ۴ متغیر نیز اثبات می شود.

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \Rightarrow N(X \mid \mu, \Sigma, \underset{\text{کلاس}}{C}) = \frac{1}{(2\pi)^{D/2}} \times \frac{1}{|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

$$D = 2$$

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$|\Sigma| = \sigma_1^2 \sigma_2^2$$

$$\underbrace{\Sigma^{-1} = \begin{bmatrix} \sigma_1^{-2} & 0 \\ 0 & \sigma_2^{-2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix}}_{\frac{1}{2\pi} \text{ همان}}$$

اثبات استقلال x_1 و x_2:

$$= \frac{1}{\sqrt{2\pi}} \times \frac{1}{\sqrt{2\pi}} \times \frac{1}{\sigma_1 \sigma_2} \times e^{-\frac{1}{2}[x_1-\mu_1 \,,\, x_2-\mu_2] \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}}$$

محاسبه:

$$[x_1-\mu_1 \,,\, x_2-\mu_2] \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} = \left[\frac{x_1-\mu_1}{\sigma_1^2} \,,\, \frac{x_2-\mu_2}{\sigma_2^2}\right]$$

$$\left[\frac{x_1-\mu_1}{\sigma_1^2} \,,\, \frac{x_2-\mu_2}{\sigma_2^2}\right] \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix} = \left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma_1} e^{-\frac{1}{2}\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2} \times \frac{1}{\sqrt{2\pi}\,\sigma_2} e^{-\frac{1}{2}\left(\frac{x_2-\mu_2}{\sigma_2}\right)^2} = \underbrace{N(x_1 \mid \mu_1, \sigma_1^2) \times N(x_2 \mid \mu_2, \sigma_2^2)}$$

پس مستقل اند و استقلال (به شرط کلاس)، معادل همان شرط نایو است.