به نام خدا

حسین غلامی 97123021

گزارش تمرین عملی سری دوم یادگیری ماشین (بیزین)

در ابتدا به دیتاست را دانلود کرده و برای سادگی نام ستون ها ، و لیبل ها را تغییر دادم

	sepal_length	sepal_width	petal_length	petal_width	y_out
1	5.1	3.5	1.4	0.2	Iris_setosa
2	4.9	3	1.4	0.2	Iris_setosa
3	4.7	3.2	1.3	0.2	Iris_setosa
4	4.6	3.1	1.5	0.2	Iris_setosa

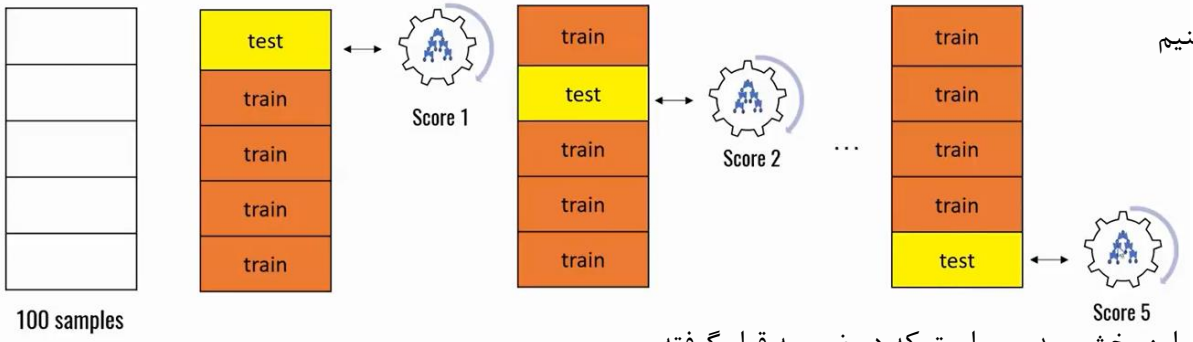
تبدیل شد به :

sl	sw	pl	pw	cl
5.1	3.5	1.4	0.2	st
4.9	3	1.4	0.2	st
4.7	3.2	1.3	0.2	st

هر دوی اطلاعات در پوشه ، data قرار گرفت.

در ادامه ابتدا ساختار kFold را بررسی میکنیم ، که به این صورت است :

به این معنی که ، بخش تست و تمرین را به شکل زیر جدا کرده و برای حالت های مختلف حساب میکنیم. و نتایج را میانگین گیری میکنیم



بهترین توضیح برای این بخش ویدیویی است که در ضمیمه قرار گرفته

برای نمونه در کد خودم :k=2

```
train index of st
 [3, 4, 5, 6, 7, 8, 10, 11, 14, 16, 17, 18, 24, 25, 26, 27, 28, 29, 32, 34, 36, 37, 38, 39, 41,
43, 44, 46]
train index of vs
 [52, 55, 56, 60, 61, 63, 67, 68, 70, 71, 72, 74, 76, 77, 78, 79, 80, 83, 85, 87, 90, 93, 95,
98]
train index of vg
 [103, 105, 110, 111, 116, 118, 119, 121, 123, 126, 127, 128, 131, 132, 133, 134, 136, 137, 142,
144, 146, 148]
test index
 [  0   1   2   9  12  13  15  19  20  21  22  23  30  31  33  35  40  42
  45  47  48  49  51  53  54  57  58  59  62  64  65  66  69  73  75  81
  82  84  86  88  89  91  92  94  96  97  99 100 101 102 104 106 107 108
 109 112 113 114 115 117 120 122 124 125 129 130 135 138 139 140 141 143
 145 147 149]
_________________________________________________________

the number of True is:  74
the number of False is : 1

the accuracy :  0.9866666666666667
```

```
train index of st
 [1, 2, 9, 12, 13, 15, 19, 20, 21, 22, 23, 30, 31, 33, 35, 40, 42, 45, 47, 48, 49]
train index of vs
 [51, 53, 54, 57, 58, 59, 62, 64, 65, 66, 69, 73, 75, 81, 82, 84, 86, 88, 89, 91, 92, 94, 96,
97, 99]
train index of vg
 [101, 102, 104, 106, 107, 108, 109, 112, 113, 114, 115, 117, 120, 122, 124, 125, 129, 130, 135,
138, 139, 140, 141, 143, 145, 147, 149]
test index
 [  3   4   5   6   7   8  10  11  14  16  17  18  24  25  26  27  28  29
  32  34  36  37  38  39  41  43  44  46  50  52  55  56  60  61  63  67
  68  70  71  72  74  76  77  78  79  80  83  85  87  90  93  95  98 103
 105 110 111 116 118 119 121 123 126 127 128 131 132 133 134 136 137 142
 144 146 148]
_________________________________________________________

the number of True is:  69
the number of False is : 6

the accuracy :  0.92

================================================================
total accuracy is : 0.9533333333333334
```

که به صورت:

kf = KFold (n_splits=2,shuffle=True,random_state=np.random)

for train_index,test_index in kf.split(df):

....

نوشته شد ، به این نحو که تابع kfold را با مقادیر رندم صدا کرده ، و در یک for هر بار به یک قسمت از دسته بندی کلی آن دسترسی پیدا میکنیم.

در ادامه ابتدا به بررسی و توضیع بیز میپزدازیم (بخش اول)

با توجه به صورت سوال و توزیع نرمال و تخمین لایکلی هود پرداخته شده ، به دلیل اینکه از چهار ویژگی برای آماده کردن توضیع داریم باید اطلاعات را به فرم ماتریسی نوشته ، و توزیع را بر آن اساس ، تخمین زد.

$$f(x) = \frac{1}{\sqrt{(2\pi)^k \det \Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right),$$

که خوشبختانه توضیع نرمال به صورت ماتریسی در کتابخانه scipy.stats وجود داشت و از آن استفاده شد.

https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.multivariate_normal.html

همچنین برای محاسبه ماتریس میانگین و واریانس(کوواریانس) کتابخانه pandas توابع مربوطه را داشت.

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.cov.html

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.mean.html

که توزیع زیر ، برای سه تابع (سه کلاس st,vs,vg) محاسبه شد.

طبق الگوریتیم ، هر کدام که احتمال بیشتری داشته باشند ، آن اطلاعات متعلق به آن ها است.

در ادامه به بررسی الگوریتم gussian nave Bayse پرداخته مشود(بخش سوم)

مشابه بخش قبل ، تفاوت الگوریتم gussian nave bayse و خود bayse در محاسبه احتمالات توام(joint) است ، که در مفهموم به همان معنی شرط استقلال است

پس یعنی تنها برای پیاده سازی این الگوریتم میبایست ، در ماتریس کواریانس ، اطلاعات بجز شاخه اصلی را صفر کنیم ، که شرط استقلال را فراهم کنیم.

بقیه مشابه قبل است.

در ادامه الگوریتم nave bayse بررسی میشود

در این الگوریتم از روی داده های موجود احتمال را میبایست حساب کرد و شرط استقلال را نیز در نظر گرفته میشود. ولی با توجه به اینکه مقادیر پیوسته هستند ، احتمال عدم وجود دقیقا همان مقدار وجود دارد. پس اگر آن مقدار وجود

نداشته باشد ، نتیجه نهایی صفر میشود ، برای جلوگیری از این امر از یک تخمین m-stimation استفاده میکنند :
(n+mp)/(n+m) که p برابر 1 به تعداد کلاس و m ضریبی است که از validation حاصل میشود.

برای پیاده سازی ابتدا یک دیکشنری از تعداد فیچر هایی در هر کلاس وجود دارد ، تهیه میکنیم :

به شرح زیر (سمت راست صفحه بعد)

سپس با داده های موجود ، سه احتمال برای کلاس هایvg, vs ,st محاسبه میکنیم

و لیبل زده شده ، برابر با بیشترین احتمال است.

```
In [55]: st_dict
Out[55]:
[sl
 4.3    1
 4.4    3
 4.6    1
 4.7    1
 4.8    3
 4.9    4
 5.0    7
 5.1    6
 5.2    2
 5.3    1
 5.4    3
 5.7    2
 5.8    1
dtype: int64, sw
3.9    1
4.0    1
4.4    1
dtype: int64, pl
1.1     1
1.2     2
1.3     4
1.4     8
1.5    12
1.6     4
1.7     3
1.9     1
dtype: int64, pw
0.1     5
0.2    17
0.3     5
0.4     6
```

IPython console | History log

در ادامه به بررسی نتایج ، و شرایطی که در تمرین مورد سوال بود

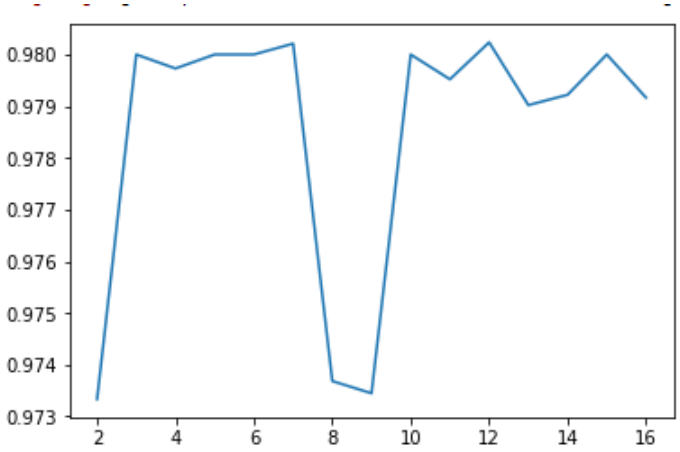
پرداخته میشود.

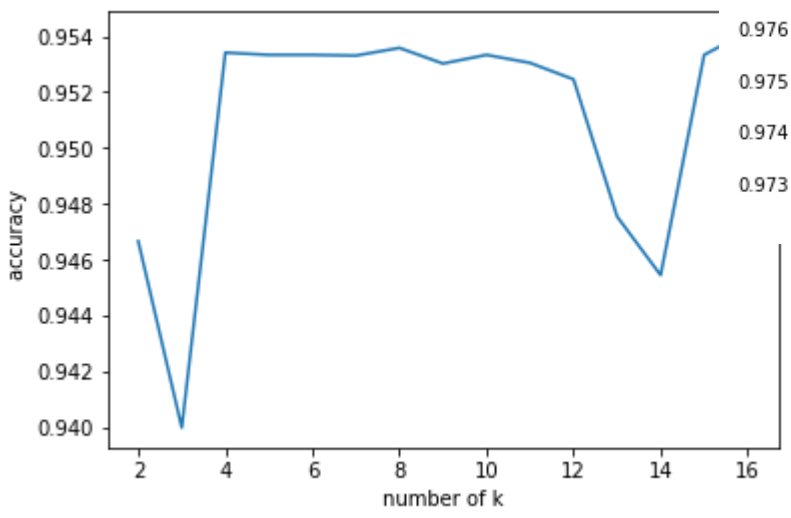
الف) total accuracy is : 0.9733333333333333

برای مقادیر مختلف k در تست های متوالی متفاوت است که نمودار های زیر حاصل شد

که متوان تقریبا گفت به ازای k=4 دقت میتواند 98 درصد باشد تاثیر خاصی ندارد

چرا که همه اطلاعات در حالت های مختلف دیده شده اند



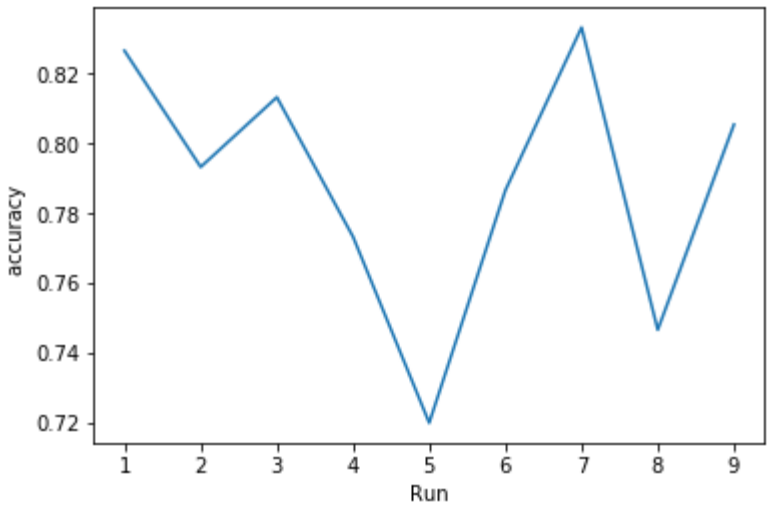


در naïve bayes به شدت به میزان ، نوع تقسیم اطلاعات وابسته است و نتیجه در دفعات متفاوت نوسان های بیشتری میکنند(m، ضریب تخمین m-stimation)

ابتدا به m ر بهترین مقدار m را بررسی میکنیم

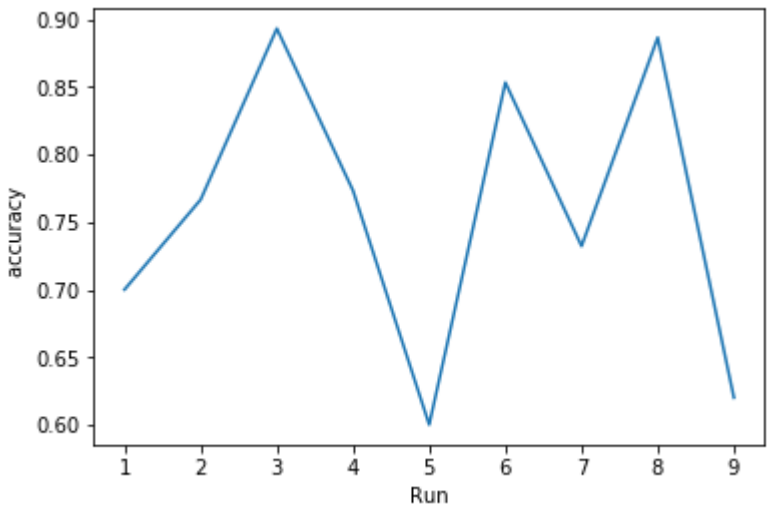
به ازای k=2 ,m=2

دقت به ازای 10 بار اجرا



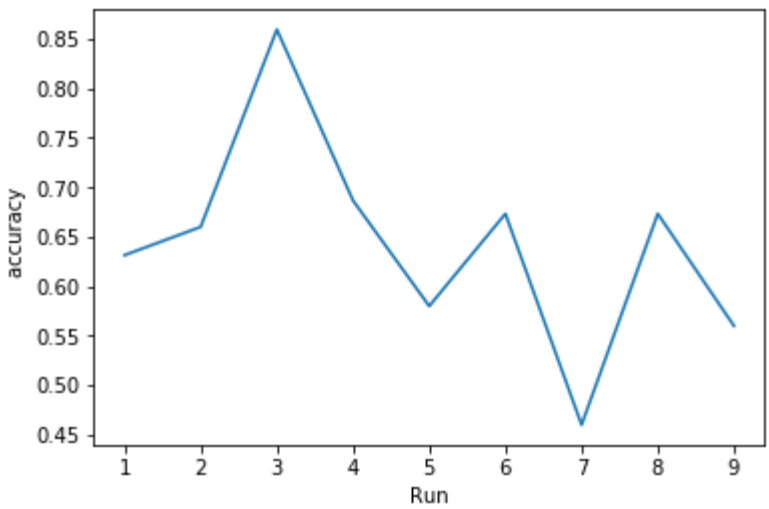
به ازای m=3 و k=2

دامنه نوسان بیشتر شد



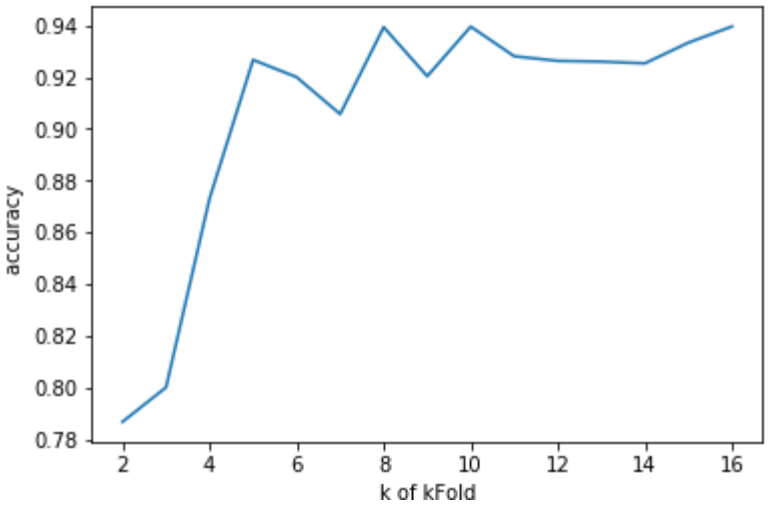
به ازای m=5 و k=2

دقت پایین آمده



برای میزان m عدد 2 یا 3 میتواند گزینه خوبی باشد.

حال میزان k را تغییر میدهیم :



اگر میزان k را افزایش دهیم دقت افزایش مییابد

چرا که باعث بیشتر دیده شدن اطلاعات میشود

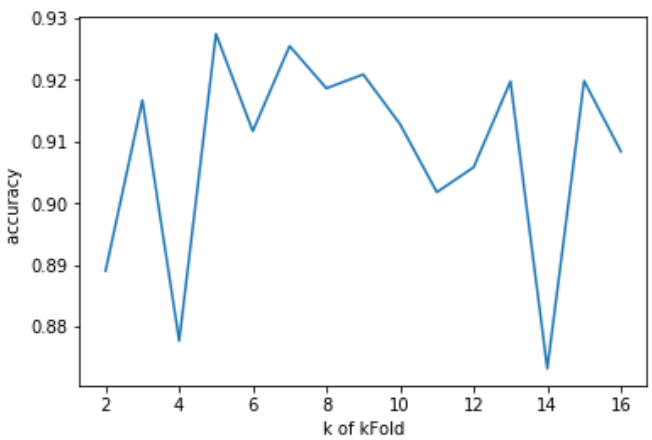
و نتیجه را بهبود میبخشد

نتیجه گیری :

اگر میزان داده مشاهده شده در حالت kfold (یعنی اینکه همه داده ها را ببنیم با تغییر train test) ، بیشتر شود ، نتیجه بهبود میابد ولی همچنان باید توجه داشت که نتیجه حالت بیز با تخمین لایکلی هود ومحاسبه احتمالات توام ، نتیجه بهتری را برای ما دارد. (4 درصد بیشتر)

Naiv bayes بدون تخمین ملایم

به دلیل اینکه در بعضی مواقع احتمال رویداری صفر میشود، و اینکه مقادیر پیوسته هستند ، باعث اشتباه در تصمیم گیری میتواند بشود که دلیل نوسان های نمودار به خاطر همین امیر است اینکه با افزایش مقدار k به یک عدد همگرا نمیشود.



بررسی gussian naïve bayes

در این حالت میزان دقت از bayes کمتر و از naïve base بیشتر است میانگین 95 درصد

که از (حالت با k زیاد در 94 nave bayes درصد بیشتر و از bayes با 98 درصد کمتر است