به نام خدا

حسین غلامی 97123021

گزارش تمرین عملی سری دوم یادگیری ماشین (بیزین)

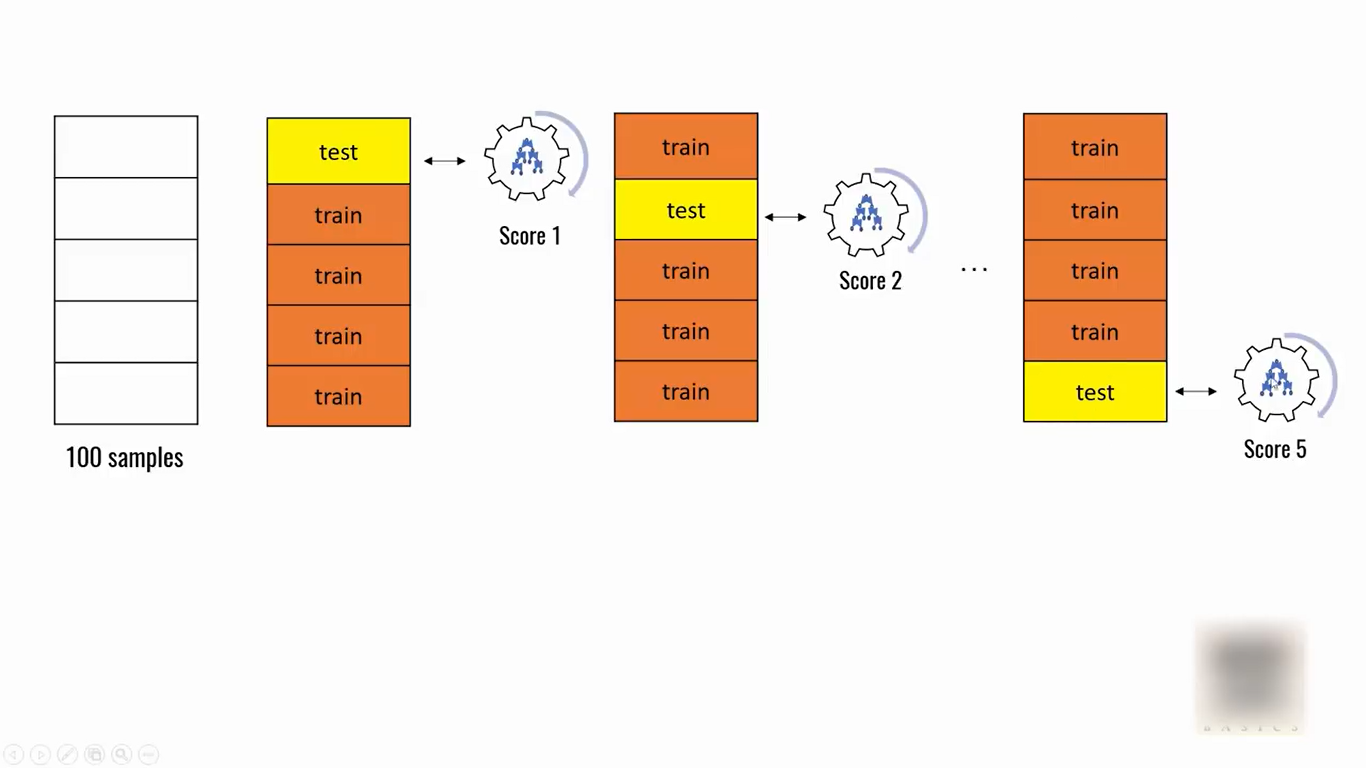
در ابتدا به دیتاست را دانلود کرده و برای سادگی نام ستون ها ، و لیبل ها را تغییر دادم

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

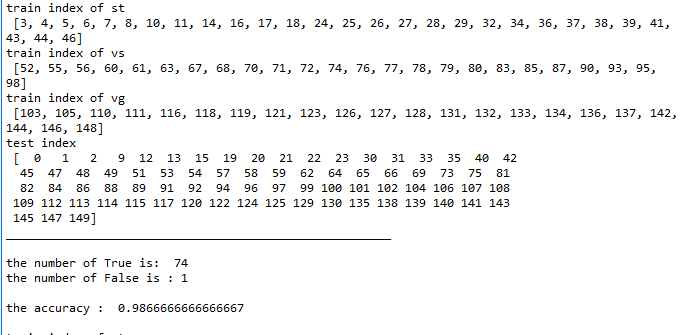
هر دوی اطلاعات در پوشه ، data قرار گرفت.

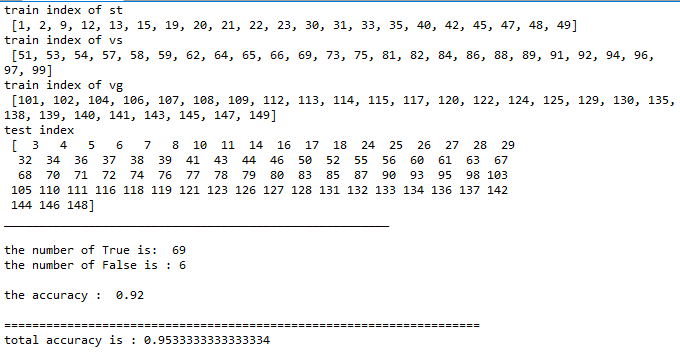
در ادامه ابتدا ساختار kFold را بررسی میکنیم ، که به این صورت است :

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





که به صورت:

kf = KFold (n\_splits=2,shuffle=True,random\_state=np.random)

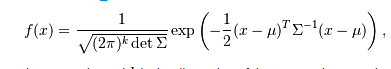
for train\_index,test\_index in kf.split(df):

....

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.



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.

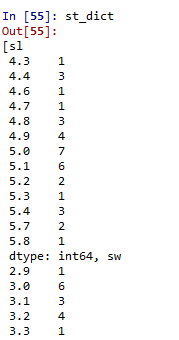
بقیه مشابه قبل است.

در ادامه الگوریتم nave bayse بررسی میشود

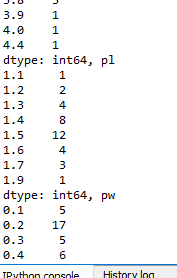
در این الگوریتم از روی داده های موجود احتمال را میبایست حساب کرد و شرط استقلال را نیز در نظر گرفته میشود. ولی با توجه به اینکه مقادیر پیوسته هستند ، احتمال عدم وجود دقیقا همان مقدار وجود دارد. پس اگر آن مقدار وجود نداشته باشد ، نتیجه نهایی صفر میشود ، برای جلوگیری از این امر از یک تخمین m-stimation استفاده 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 های st ,vs ,vg محاسبه میکنیم

و لیبل زده شده ، برابر با بیشترین احتمال است.

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

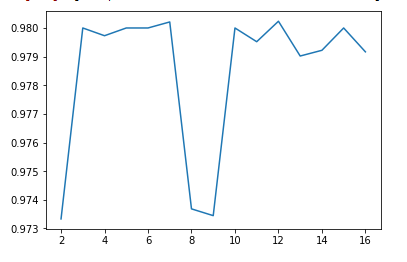
پرداخته میشود.

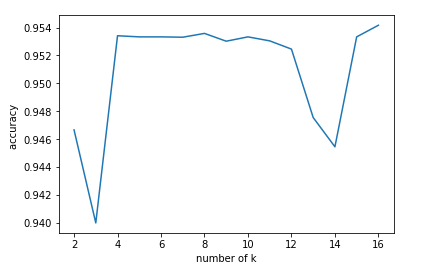
الف) total accuracy is : 0.9733333333333333

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

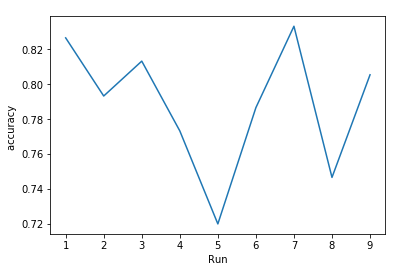
که متوان تقریبا گفت به ازای k=4 دقت میتواند 98 درصد باشد تاثیر خاصی ندارد

چرا که همه اطلاعات در حالت های مختلف دیده شده اند



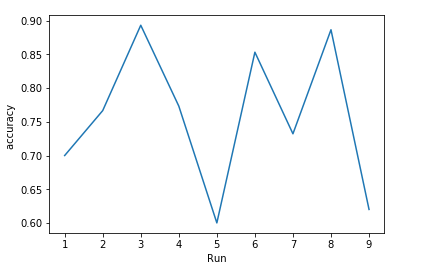


در naïve bayes به شدت به میزان ، نوع تقسیم اطلاعات وابسته است و نتیجه در دفعات متفاوت نوسان های بیشتری میکنند) m، ضریب تخمین m-stimation)

ابتدا به m ر بهترین مقدار m را بررسی میکنیم

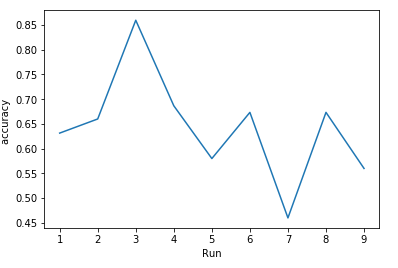
به ازای k=2 ,m=2

دقت به ازای 10 بار اجرا



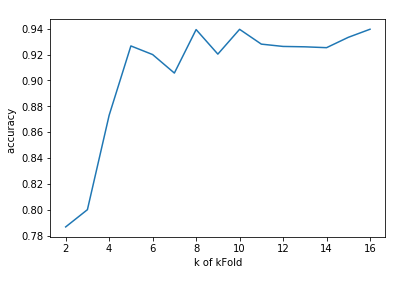
به ازای m=3 و k=2

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

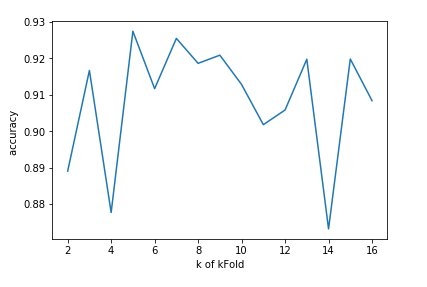
و نتیجه را بهبود میبخشد

نتیجه گیری :

اگر میزان داده مشاهده شده در حالت kfold (یعنی اینکه همه داده ها را ببنیم با تغییر test train) ، 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 nave bayes 94 درصد بیشتر و از bayes با 98 درصد کمتر است

