تمرین شماره 2 - دیتاست دوم بیماری قلبی

محمد زیاری - 97222047

قضیه بیز

قضیه بیز یک فرمول ریاضیاتی است که در پیدا کردن احتمال شرطی به کار میرود. احتمال شرطی، احتمال وقوع یک نتیجه بر اساس رخداد نتیجه قبلی در شرایط مشابه است. قضیه بیز راهی برای تجدید نظر در پیش بینی ها یا نظریه های موجود با ارائه شواهد جدید یا اضافی ارائه می دهد. درواقع از احتمال پیشین استفاده میکند تا بتواند احتمال پسین را پیشبینی کند.

احتمال پیشین، در استنتاج آماری بیزی، احتمال وقوع یک رویداد قبل از جمع‌آوری داده‌های جدید است. به عبارت دیگر، نشان دهنده بهترین ارزیابی منطقی از احتمال یک نتیجه خاص بر اساس دانش فعلی قبل از انجام تست است.

احتمال پسین، احتمال تجدید نظر شده یک رویداد پس از در نظر گرفتن داده های جدید است. احتمال پسین با به روز رسانی احتمال قبلی با استفاده از قضیه بیز محاسبه می شود. از نظر آماری احتمال پسین، احتمال وقوع رویداد A با توجه به وقوع رویداد B است.

فرمول ساده این قضیه به شرح زیر است:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) \, . \, P(B|A)}{P(B)}$$

به بیان دیگر این قضیه بیان می کند که احتمال شرطی یک رویداد، بر اساس وقوع یک رویداد دیگر، برابر است با احتمال رخداد دوم با توجه به اولین رویداد ضرب در احتمال رویداد اول.

به طور کل Naive Bayes گروهی از الگوریتم‌های طبقه‌بندی یادگیری ماشین نظارت شده بر اساس قضیه بیز هستند. این یک تکنیک طبقه بندی ساده است، اما عملکرد بالایی دارد. این طبقه بندها زمانی استفاده می شوند که ابعاد داده ما زیاد باشد. مسائل طبقه بندی پیچیده را نیز می توان با

استفاده از طبقه بند های Naive Bayes پیاده سازی کرد. این طبقه بندها فرض می کنند که ارزش یک feature خاص مستقل از ارزش هر feature دیگر است.

حال به طور خاص سه دسته بند را بررسی میکنیم.

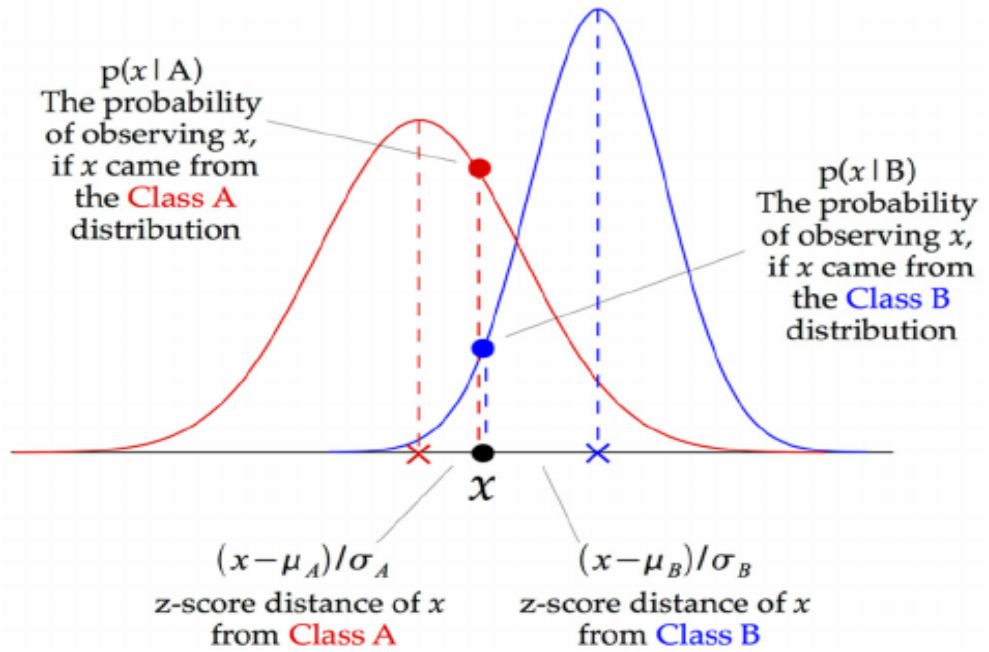
دسته بند Gaussian Naive Bayes

هنگام کار با داده‌های پیوسته، فرضیه اغلب این است که مقادیر پیوسته مرتبط با هر کلاس بر اساس توزیع نرمال (یا گوسی) توزیع می‌شوند. Likelihood ویژگی ها به صورت زیر فرض می شود:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Gaussian Naive Bayes از ویژگی‌ها و مدل‌های ارزش پیوسته پشتیبانی می‌کند که هر کدام با توزیع گاوسی (نرمال) مطابقت دارند.

یک رویکرد برای ایجاد یک مدل ساده این است که فرض کنیم داده ها با یک توزیع گاوسی بدون کوواریانس (ابعاد مستقل) توصیف می شوند. این مدل را می توان به سادگی با یافتن میانگین و انحراف معیار نقاط در هر برچسب، متناسب کرد.



تصویر بالا نشان می دهد که چگونه یک دسته بند کننده (GNB) کار می کند. در هر نقطه داده، فاصله z-score بین آن نقطه و میانگین هر کلاس محاسبه می‌شود، یعنی فاصله از میانگین کلاس بر انحراف معیار آن کلاس تقسیم می‌شود.

دسته بند Multinomial Naive Bayes

الگوریتم چند جمله ای ساده بیز یک روش یادگیری احتمالی است که بیشتر در پردازش زبان طبیعی (NLP) استفاده می شود. این الگوریتم بر اساس قضیه بیز است و برچسب یک متن مانند یک ایمیل یا مقاله روزنامه را پیش بینی می کند. احتمال هر تگ را برای یک نمونه معین محاسبه می کند و سپس تگ با بالاترین احتمال را به عنوان خروجی می دهد.
در این حالت برحسب مدل احتمالی یا توزیع چند جمله‌ای ، برداری از n ویژگی برای یک مشاهده به صورت $X = (x_1, \dots x_n)$ با احتمالات $p = (p_1, \dots p_n)$ در نظر گرفته میشود.

در این دسته بند با استفاده از قضیه بیز، احتمال وقوع کلمات و تعداد آنها را میشماریم و به پردازش متن میپردازیم.

دسته بند Bernoulli Naive Bayes

این دسته بند برای داده های گسسته استفاده می شود و بر اساس توزیع برنولی کار می کند. ویژگی اصلی Bernoulli Naive Bayes این است که ویژگی ها را فقط به عنوان مقادیر باینری مانند true یا false، بله یا خیر، موفقیت یا شکست، 0 یا 1 و غیره می پذیرد. بنابراین وقتی مقادیر ویژگی باینری هستند می دانیم که باید از طبقه بندی کننده Bernoulli Naive Bayes استفاده کنیم.

توزیع برنولی: همانطور که با مقادیر باینری سروکار داریم، اجازه دهید "p" را به عنوان احتمال موفقیت و "q" را به عنوان احتمال شکست و q=1-p را در نظر بگیریم.
برای متغیر تصادفی 'X' در توزیع برنولی داریم:

$$p(x) = P[X = x] = \begin{cases} q = 1 - p & x = 0 \\ p & x = 1 \end{cases}$$

که در آن x تنها میتواند مقادیر 0 یا 1 باشد

همچنین این دسته بند از قانون احتمالاتی زیر پیروی میکند:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

مقایسه باهم : چند جمله ای ساده بیز یک بردار ویژگی را در نظر می گیرد که در آن یک عبارت معین تعداد دفعات ظاهر شدن یا فرکانس را نشان می دهد. از سوی دیگر، برنولی یک الگوریتم باینری است که در صورت وجود یا عدم وجود ویژگی استفاده می شود و گاوسی بر اساس توزیع پیوسته است.

- **پیاده سازی**

با توجه به بررسی های صورت گرفته در سوال اول کاملا با مفهوم gaussian naive bayes اشنا شدیم. برای پیاده سازی بدون پکیج این روش از towards datasciense کمک گرفتم که مفاهیم را پیاده سازی کرده بود.

سپس 3 فیچری که گفته شد را برای لیبل x انتخاب می کنیم و y یا همان تارگت مان را هم جدا می کنیم و پس از تبدیل به numpy array و جدا کردن داده های تست و آموزشی آنها را به مدلمان می دهیم تا روی داده های آموزشی یادگیری را آغاز کند.

پس از فیت کردن نوبت به آن می رسد تا روی داده های تست تابع predict را صدا بزنیم تا پیشبینی را روی آن انجام دهیم. پس از آن دقت مدلمان تقریبا 50 درصد می باشد که یعنی هیچ یادگیری درستی صورت نگرفته است و اصلا مدل خوبی نداریم.

میزان معیار هایی که لازم به گزارش بود در جدول زیر آمده است.

	precision	recall	f1-score	support
0	0.50	0.03	0.06	30
1	0.51	0.97	0.67	31
accuracy			0.51	61
macro avg	0.50	0.50	0.36	61
weighted avg	0.50	0.51	0.37	61

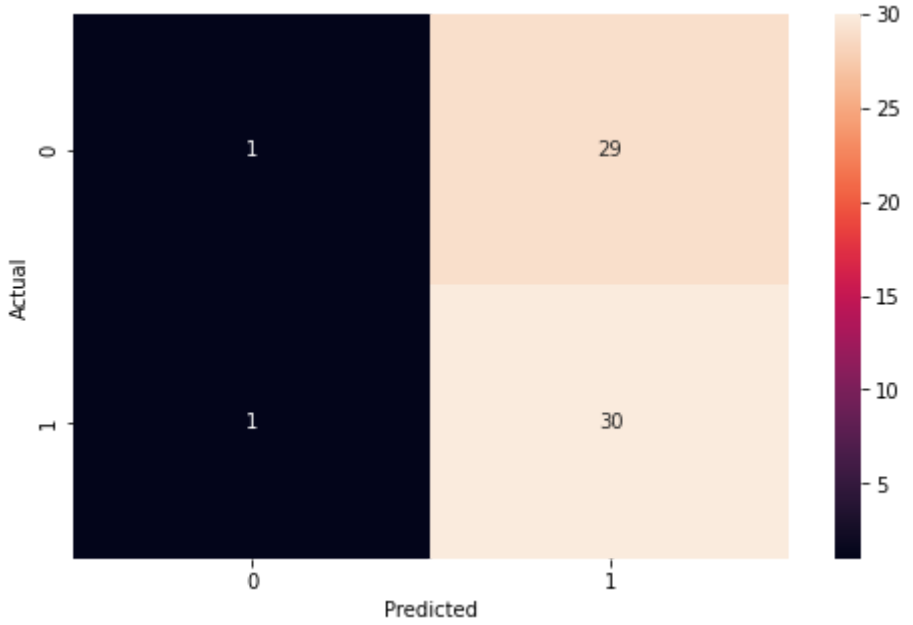
همانطور که مشخص است میزان precision یا معیار صحت که برابر با فرمول زیر است در هر دو کلاس 50 درصد است. در واقع نصف این درستها را به درستی تشخیص دادیم .

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

اما تفاوت معنادار در recall یا معیار پوشش مشخص می شود که فرمولش در پایین امده است و در کلاس 1 نزدیک به 97 درصد و در کلاس 0 تنها 3 درصد است. واضح است که مدل به سمت حفظ کردن رفته است و بیشتر داده ها را به لیبل 1 وصل کرده است.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

همانطور که پیشبینی کردیم و در confusion matrix زیر هم مشاهده می کنیم بیش از 95 درصد داده ها برای کلاس 1 پیشبینی شده اند که واضح است مدلمان در حال memorize کردن می باشد و چیزی که از معیارهای بالا فهمیدیم درست بود.

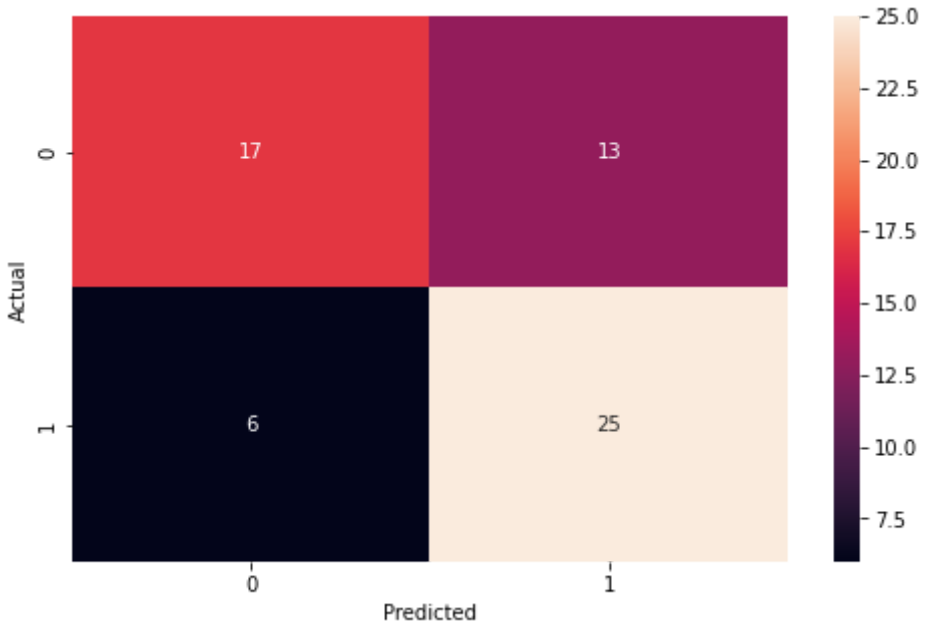


اما زمانی که از پکیج sklearn استفاده میکنیم میزان دقت مان تقریبا به 69 درصد می رسد که پیشرفت چشمگیری نسبت به حالت بدون پکیج دارد. هرچند این مدل هم خیلی خوب نیست. میزان معیارها نیز به صورت زیر است.

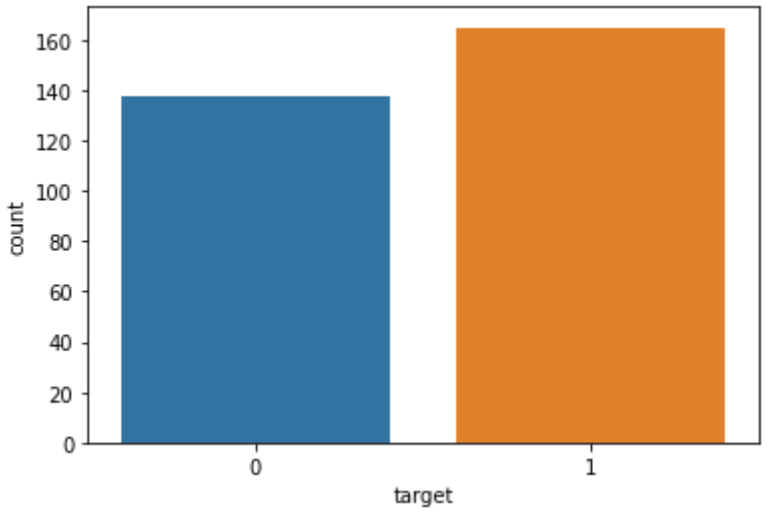
	precision	recall	f1-score	support
0	0.74	0.57	0.64	30
1	0.66	0.81	0.72	31
accuracy			0.69	61
macro avg	0.70	0.69	0.68	61
weighted avg	0.70	0.69	0.68	61

طبق f1-score دو کلاس مشخص است که دیگر به ان شکل به سمت کلاس 1 باياس نداریم و کمی متعادلتر پیشبینی صورت پذیرفته است. هرچند طبق معیار recall میزان پوشش روی داده های درست کلاس 1 نزدیک به 24 درصد بیشتر از کلاس 0 است که هرچند اندکی پیشرفت نسبت به

حالت بدون پکیج محسوب می شود اما هنوز مدل خوبی نیست. نمودار confusion matrix را نیز در زیر با هم مشاهده می کنیم .



مشخص است بیشترین اشکال در آن است که داده های زیادی که در کلاس 0 جای می گرفتند در کلاس 1 پیشبینی شده اند که نشان دهنده اندکی بالانس نبودن مدلمان می باشد. از این رو نمودار target را در شکل زیر مشاهده می کنیم تا ببینیم داده های اولیه بالانس بوده اند یا خیر.



مشخص است که داده ها خیلی هم imbalanced نمی باشند از این رو مشکل اصلی همان کم بودن داده هاست. برای بیشتر کردن داده ها از upsampling می توانیم استفاده کنیم که من در این دیتاست این کار را انجام ندادم.