به نام خداوند جان و خرد

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

یادگیری ماشین

تمرین شماره ۱

نام و نام خانوادگی: علی خرم فر

شماره دانشجویی: ۸۱۰۱۰۲۱۲۹

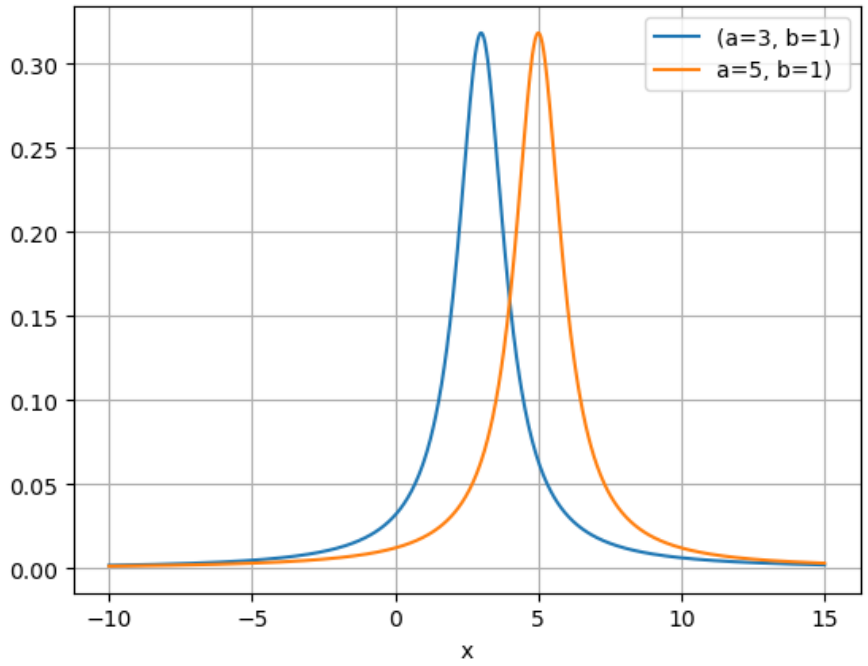
اسفندماه ۱۴۰۲

فهرست مطالب

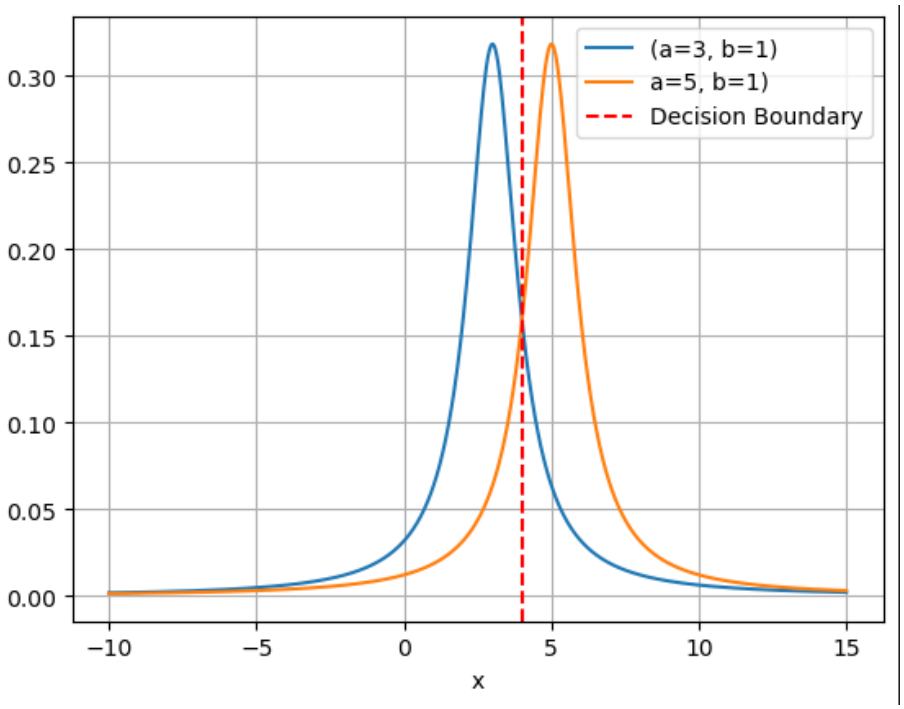
۱_ پاسخ سوال ۱)

۱-۱_ قسمت آ

ابتدا دو توزیع را بر حسب موارد گفته شده با کمک کتابخانه matplotlib رسم می‌کنیم:



مرز تصمیم باتوجه به مسئله برابر x=4



۲-۱_ پاسخ قسمت ث)

در این مسئله داریم:

$$\begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$$

ریسک انتخاب کلاس w1 به شرط مشاهده x

$$R(w_1|x) = \lambda_{11}P(w_1|x) + \lambda_{12}P(w_1|x)$$

ریسک انتخاب کلاس w2 به شرط مشاهده x

$$R(w_2|x) = \lambda_{21}P(w_1|x) + \lambda_{22}P(w_2|x)$$

فرض می‌کنیم که $P(w_2) = P(w_1)$

۲_پاسخ سوال ۲)

پاسخ پیوست شد.

۳_ پاسخ سوال ۳)

پاسخ پیوست شد.

۴_پاسخ سوال ۴)

پاسخ پیوست شد.

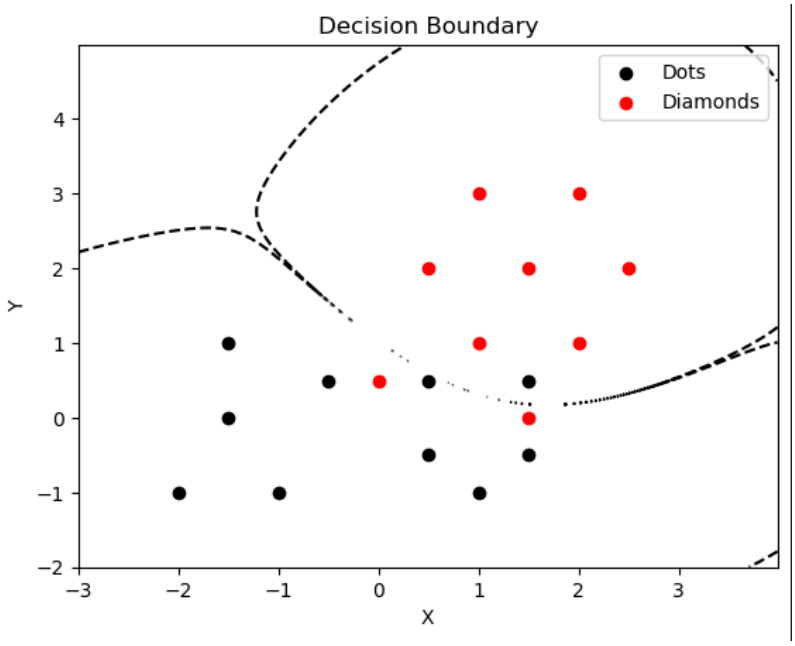
۵_پاسخ سوال ۵)

این سوال به صورت برنامه نویسی حل شد. ابتدا نقاط نقاط در یک آرایه ذخیره شدند. یک آرایه نقاط diamond که همان قرمزها هستند و یک ماتریس برای نقاط مشکی سپس با کمک کتابخانه numpy میانگین و کواریانس محاسبه شدند. خروجی آن‌ها به صورت زیر است:

```
mean_diamonds, mean_dots, cov_diamonds, cov_dots
✓ 0.2s
(array([1.33333333, 1.61111111]),
 array([-0.15, -0.15]),
 array([[0.625     , 0.20833333],
        [0.20833333, 1.11111111]]),
 array([[1.725     , 0.00277778],
        [0.00277778, 0.55833333]]))
```

پاسخ قسمت ب)

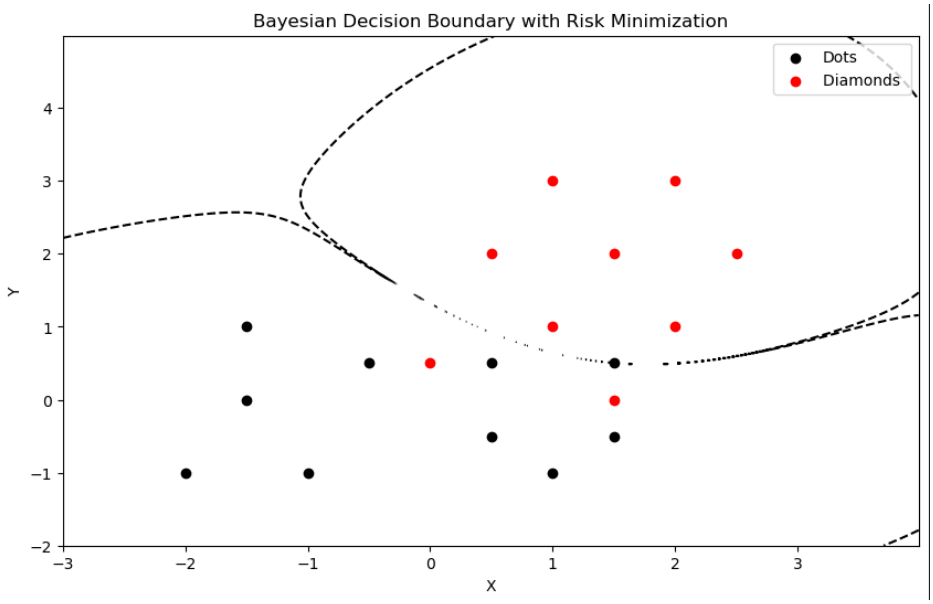
مرز تصمیم نقاطی را می‌یابیم که احتمال تعلق به یک کلاس به اندازه کلاس دیگر باشد. برای توزیع گوسی این مورد حساب شد که خروجی به صورت زیر است. این نمودار با scatter plot از کتابخانه matplotlib رسم شده است. شکل مرز باتوجه به درجه دو بودن معادله ممکن است دایره بیضی سهمی یا هذلولی باشد:



درصد نقاطی که اشتباه طبقه‌بندی شده اند که در بالا مشخص هستند دو قرمز و یک مشکی:

15.789473684210526

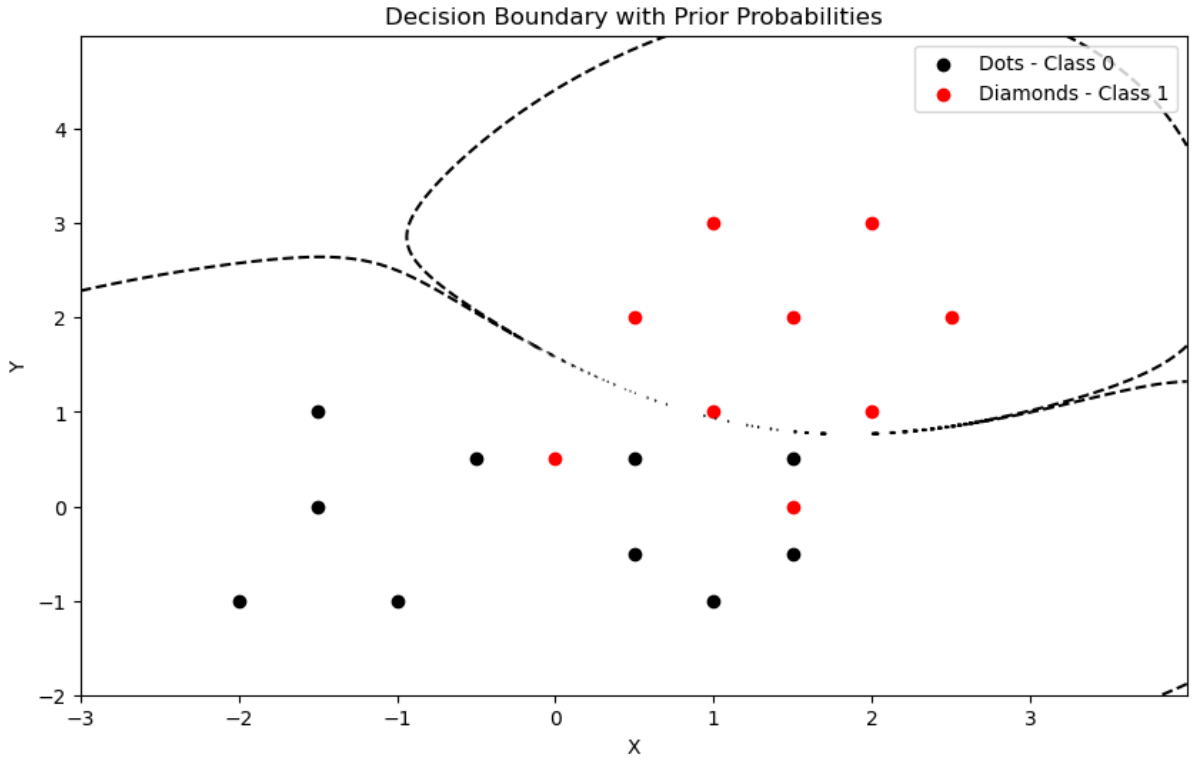
پاسخ قسمت پ)



اگر سختگیرانه عمل کنیم ۳ نقطه اشتباه طبقه‌بندی شده اند. پس میزان خطای تجربی برابر است با همان ۱۵.۷ درصد یا سه نوزدهم

پاسخ قسمت ت)

$$P(\omega_1) = \frac{1}{3}, P(\omega_2) = \frac{2}{3}$$



اینجا کمی خط بالا تر رفته و نقطه سومی که در قبلی ها حتی سختگیرانه خطا داشت اینجا مشکلی ندارد و به درستی طبقه‌بندی شده و فقط ۲ نقطه قرمز مشکل دارند.

۶_پاسخ سوال ۶)

پاسخ پیوست شد.

۷_پاسخ سوال ۷)

۷-۱_ پاسخ قسمت آ

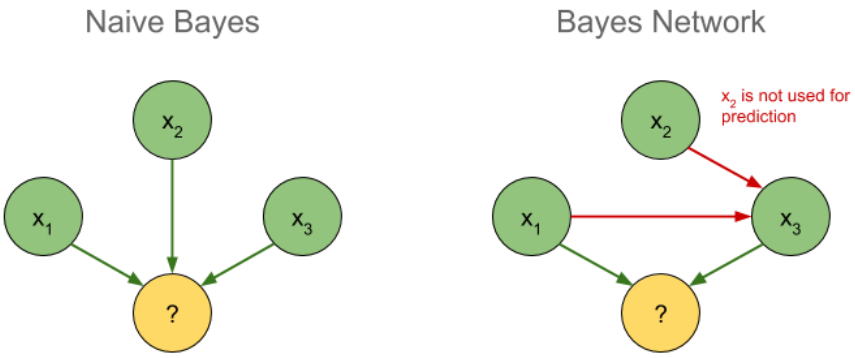
مقایسه دو طبقه‌بند

در این قسمت از سوال قصد داریم که طبقه بند Naïve Bayes را با طبقه‌بند بیزی مقایسه کنیم. در طبقه‌بند بیزی احتمال یک کلاس بر اساس ویژگی‌های آن بر اساس فرمول معروف بیز محاسبه می‌شود.

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

این فرمول از احتمال joint یا توام تمامی ویژگی‌ها برای محاسبه احتمال پسین استفاده می‌کند. در طبقه‌بند Naïve Bayes فرض می‌شود که ویژگی‌ها مستقل هستند و این تفاوت اصلی با طبقه‌بند بیزی است. باتوجه به اینکه در آمار و احتمالات می‌توانیم احتمالات شرطی وقتی که از هم مستقل باشند را در هم ضرب کنیم تا حاصل بدست آید، این مورد باعث کاهش محاسبات در این نوع طبقه‌بند می‌شود. اگرچه در دنیای واقعی ویژگی‌ها مستقل نیستند ولی همین فرض ساده‌سازی در Naïve Bayes باعث می‌شود که محاسبات پیچیدگی کمتری داشته باشند و درعین حال در بسیاری از مسائل به خوبی جواب دهد. به طور کلی وقتی که ویژگی ها مستقل باشند و یا ابعاد زیاد باشد از این نوع طبقه‌بند استفاده می‌کنیم. برای نمونه در درس پردازش زبان طبیعی با کمک این طبقه‌بند تحلیل احساسات تا حد قابل قبولی جواب می‌دهد. هزینه‌ای که می‌دهیم این است که دقت این مدل نسبت به طبقه‌بند بیزی پایین تر است. در اصل طبقه‌بند بیزی مثل یک حالت بهینه است که ما سعی داریم به آن نزدیک شویم ولی باتوجه به هزینه بسیار زیاد برخی ساده‌سازی ها را انجام داده ایم تا بتوانیم در شرایط واقعی از آن استفاده کنیم.

به طور کلی در این نوع از طبقه‌بند، محاسبه احتمال پسین هر کلاس، به شرط ویژگی‌های مشاهده با محاسبه احتمال پیشین هر کلاس بر اساس قضیه بیز انجام می‌شود.



علت پیش‌پردازش داده‌ها

قبل از انجام پردازش روی داده‌ها لازم است تا داده‌ها آماده شوند تا نتیجه پردازش بهتر شود. با توجه به اینکه داده‌ها از چه نوعی هستند پردازش‌های مختلفی انجام می‌شود. علاوه بر این نوع پیش‌پردازش به هدف نیز بستگی دارد. این موضوع در درس استنباط آماری به طور کامل بررسی شد. برای نمونه اگر داده‌ها متنی باشند و هدف تحلیل احساسات با کمک طبقه‌بند بیز باشد لازم است قبل از توکنایز کردن متن، علائم نگارشی حذف و تمام حروف بزرگ به کوچک تبدیل شده و به طور کلی داده‌های خام را نرمال‌سازی کنیم. پیش‌پردازش‌های مختلفی برای داده‌های مختلف انجام می‌شود. برای مثال برای مقادیر null تصمیم‌گیری می‌شود. داده‌ها نرمال می‌شوند و یا متغیرهای Cathegorical به Numerical تبدیل می‌شوند. حال در ادامه به بررسی داده‌های ارائه شده می‌پردازیم و بررسی می‌کنیم که چه پیش‌پردازشی برای این داده‌ها مناسب است و این کار را با ذکر دلیل هر کدام انجام می‌دهیم.

پیش‌پردازش روی دیتاست اول

ابتدا داده‌ها را با کمک کتابخانه Pandas ایمپورت کردیم و ۵ ردیف اول آن را خروجی کردیم:

Preprocessing

```
import pandas as pd
df = pd.read_csv('survey lung cancer.csv')
df.head()
```

0.8s Python

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN	LUNG_CANCER
0	M	69	1	2	2	1	1	2	1	2	2	2	2	2	2	YES
1	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES
2	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO
3	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO
4	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO

این دیتاست مربوط به سرطان ریه است. ویژگی‌های مختلف افراد ارائه شده و در نهایت targetاین دیتاست yes یا no بودن سرطان ریه است.

ابتدا اگر مقدار null وجود دارد آن را حذف می‌کنیم.

```
df.isnull().sum()
```
✓ 0.0s

```
GENDER                  0
AGE                     0
SMOKING                 0
YELLOW_FINGERS          0
ANXIETY                 0
PEER_PRESSURE           0
CHRONIC DISEASE         0
FATIGUE                 0
ALLERGY                 0
WHEEZING                0
ALCOHOL CONSUMING       0
COUGHING                0
SHORTNESS OF BREATH     0
SWALLOWING DIFFICULTY   0
CHEST PAIN              0
LUNG_CANCER             0
dtype: int64
```

از این نظر مشکلی وجود ندارد.

باتوجه به اینکه طبقه‌بند ما باینری است، می‌توانیم مقدار ستون lung cancer را به صورت ۰ و ۱ تبدیل کنیم. این مورد را می‌توانیم برای جنسیت هم انجام داده و جنسیت مرد صفر و جنسیت زن ۱ تعیین شود. همچنین تمامی داده‌ها به صورت ۱-۲ لیبل زده شده که ما میتوانیم آن را به صورت ۰-۱ کنیم. برای اینکار داده‌ها را به نوع Float تبدیل کرده و یک عدد از مقدار آن‌ها کم می‌کنیم.

یک مورد دیگر نیز که در این دیتاست مشاهده شد این بود که برخی مقادیر X هستند. احتمالا مقدار آن‌ها مشخص نیست و باید NaN شوند.

تمامی موارد بالا به عنوان پیش‌پردازش انجام شد.

۲-۷_ پاسخ قسمت ب

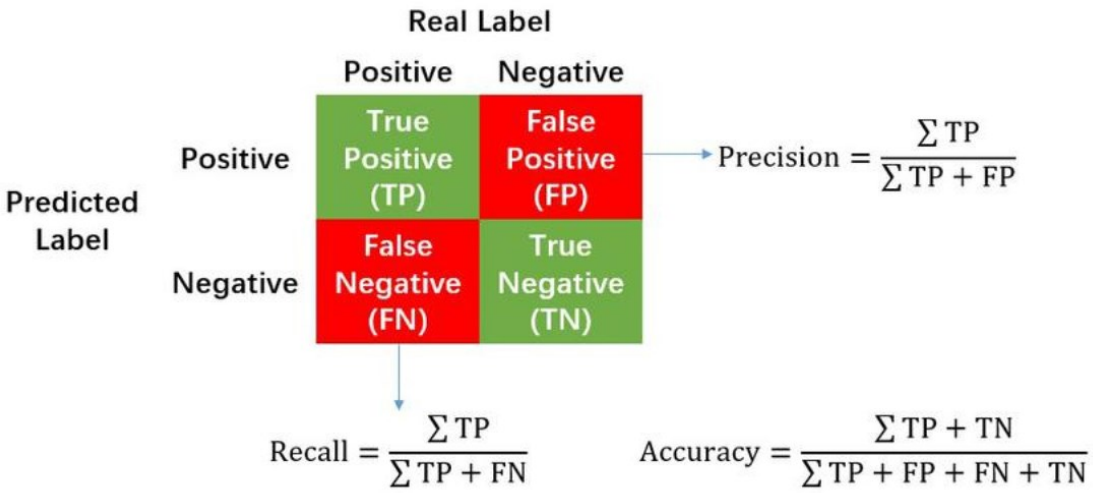
در این قسمت می‌خواهیم بدون استفاده از کتابخانه یک طبقه‌بندی naïve bayes پیاده‌سازی کنیم و به کمک آن وجود یا عدم وجود سرطان ریه در نمونه‌ها را بر اساس ویژگی‌های آن‌ها پیش‌بینی کنیم. قبل از هر عمل طبقه‌بندی لازم است که داده‌های اولیه به دو دسته آموزش و تست دسته‌بندی شوند.

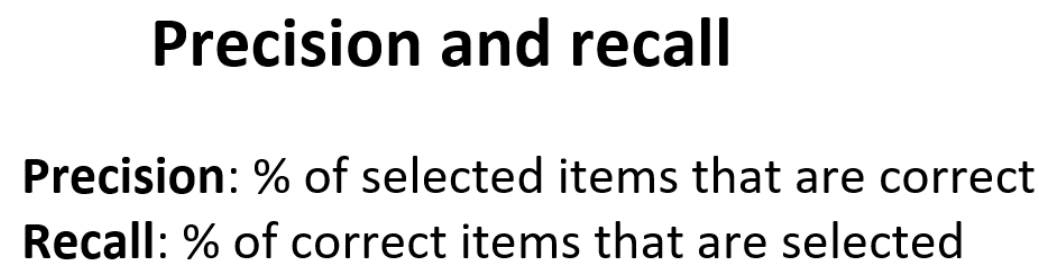
random_state=42 قرار می‌دهیم تا در هربار اجرا دقت را بتوانیم با حالت ثابتی از داده ها افزایش دهیم.

```
from sklearn.model_selection import train_test_split
X = df.drop('LUNG_CANCER', axis=1)
y = df['LUNG_CANCER']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

سپس توابعی برای محاسبه احتمال هر کلاس و احتمال شرطی بر اساس ویژگی‌ها نوشته شد. باتوجه به اینکه تایپ داده‌ها را float کردیم شرط ویژگی را برابر ۰ یا برابر ۱ قرار ندادیم و با ۰.۵مقایسه شد. در اصل اینجا فقط دو کلاس برای هر ویژگی باتوجه به نوع داده فرض شده است. پس از محاسبه احتمالات دقت محاسبه شد که درعین سادگی پیاده‌سازی دقت ۹۱ درصدی حاصل شد.

منظور از دقت نسبت پیش‌بینی‌های درست به کل پیش‌بینی هاست. ولی نکته‌ای که وجود دارد در همه موارد فقط نمیتوان با بررسی پارامتر دقت نتیجه گرفت که مدل خوب است. ممکن است تمامی داده‌ها مقداری باشند که مدل فقط آن را خوب پیش‌بینی میکند. یا در برخی داده‌ها مثل سرطان اینکه بگوییم کسی سرطان ندارد ولی داشته باشد خیلی بدتر از این است که بگوییم سرطان دارد و سرطان نداشته باشد. زیرا که مسئله حیاتی است. پس موارد دیگر را تحلیل می‌کنیم:





Precision همان نسبت تمام موارد مثبتی که درست تشخیص داده شده به موارد مثبت است که درست تشخیص داده شده و یا به اشتباه مثبت تشخیص داده شده‌اند.که در مخرج تمامی مواردی که مدل مثبت تشخیص داده مد نظر است. که این مقدار همان دقت مدل است که در اولین اجرا ۹۶ درصد بود. یعنی ۹۶ درصد مثبت‌هایی که تشخیص داده درست هستند.

Recall یا فراخوانی برابر است با نسبت تعداد موارد مثبتی که درست تشخیص شده به مواردی است که مثبت بوده و درست تشخیص داده و یا منفی بوده و به اشتباه مثبت تشخیص داده است. که در مخرج تمامی موارد مثبت واقعی مد نظر است. که مقدار ۹۵ درصد نشان می‌دهد ۹۵ درصد موارد مثبت واقعی تشخیص داده شده‌اند.

خروجی ماتریس آشفتگی نیز برابر است با [[57, 2], [3, 0]]

یعنی مقدار مثبت‌های پیش‌بینی شده درست ۵۷ ، مثبت‌های پیش‌بینی شده اشتباه ۲، منفی‌های درست ۳ و منفی‌های اشتباه ۰ بود. که برای داده‌های تست بسیار خوب است. ولی احتمالا علت این دقت بالا این است که کلا داده‌های غیر مثبت کم بوده‌اند. یعنی بیشتر داده‌ها سرطان داشته‌اندو تستشان مثبت بوده است.

۳-۷_ پیاده‌سازی با SKLEARN

باتوجه به اینکه مدل متغیر پیوسته‌ای مثل سن دارد از نوع گوسی استفاده کردیم. ابتدا خطای عدم پذیرش NaN توسط این تابع دریافت کردیم. کافی است که مقادیر null را بر اساس اینکه بقیه چه هستند میانگین بگیریم. باتوجه به اینکه طبقه‌بند گوسی است مشکلی پیش نخواهدآمد.

نتایج به صورت زیر هستند. مورد اول accuracy مورد دوم precisionو مورد سوم recall :

```
(0.967741935483871,
 0.9833333333333333,
 0.9833333333333333,
 array([[ 1,  1],[ 1, 59]], dtype=int64))
```

که نزدیک به داده‌های پیش‌بینی شده توسط مدل خودمان بود. اندکی بهبود حاصل شد که احتمالا به دلیل بهینه بودن محاسبات در کتابخانه و همچنین میانگین گیری برای مقادیر nan بود. ترتیب موارد در ماتریس آشفتگی کتابخانه با آنچه در درس خواندیم تفاوت دارد.

۴-۷_ اجرای موارد قبل روی دیتاست دوم

نتایج بدون استفاده از کتابخانه:

```
(0.365607513988809, 1.0, [[7318, 12698], [0, 0]])
```

نتایج در این مرحله بسیار ضعیف است و این مورد قابل پیش‌بینی است. زیرا که توابع بسیار ساده بودند و در اینجا برخی ویژگی‌ها چند کلاس دارند. بنابراین پیاده‌سازی قبلی کل داده‌ها را ۱ پیش‌بینی کرده است. پس این پیاده‌سازی مشکل دارد. مشکل متغیر پیوسته هم هنوز پابرجاست. برای حل این مشکل یک طبقه‌بندی naïve bayes دیگر پیاده سازی کردم تا نتیجه بهتری دریافت شود. این طبقه بند جدید با کمک کدهایی از اینترنت توسعه یافته شده است.

نتایج بدون استفاده از کتابخانه و پیاده‌سازی دوم:

برای پیاده‌سازی این کد دوم از برخی توابع موجود در اینترنت الهام گرفته شده است. نتایج به صورت زیر هستند. مورد اول accuracy مورد دوم precisionو مورد سوم recall :

```
(0.8509692246203038,
 0.7419895054147594,
 0.9081716315933315,
 array([[ 6646,  2311],
        [  672, 10387]]))
```

در این کد جدید به طور دقیق تری naïve bayes پیاده‌سازی شده و حالت عمومی تری دارد و برای داده‌های پیوسته مناسب است. مورد قبلی بیشتر بر روی داده‌هایی مثل دیتاست اول مربوط به سرطان سینه مناسب بود و با ویژگی‌های پیوسته و ویژگی‌ای چندکلاسه سازگاری کامل نداشت.

نتایج با کمک SKLEARN

```
(0.7156274980015987,
 0.8864068441064639,
 0.25485105220005466,
 array([[12459,   239],
        [ 5453,  1865]], dtype=int64))
```

همانطور که مشاهده می‌شود دقت در این دیتاست بسیار بیشتر از پیاده‌سازی خودمان است. دلیل این موضوع می‌تواند این باشد که داده‌های این دیتاست بر خلاف دیتاست قبلی باینری نیستند. همچنین اینکه دیتاست قبلی تعداد مثبت زیادی داشت تاثیری بر روی دقت داشت.

True Positives (TP): 1865

False Positives (FP): 239

True Negatives (TN): 12459

False Negatives (FN): 5453

یعنی مقدار مثبت‌های پیش‌بینی شده درست 1865، مثبت‌های پیش‌بینی شده اشتباه 239، منفی‌های درست 12459و منفی‌های اشتباه 5453بود. که نتیجه بسیار بهتر از پیاده‌سازی خودمان بود.

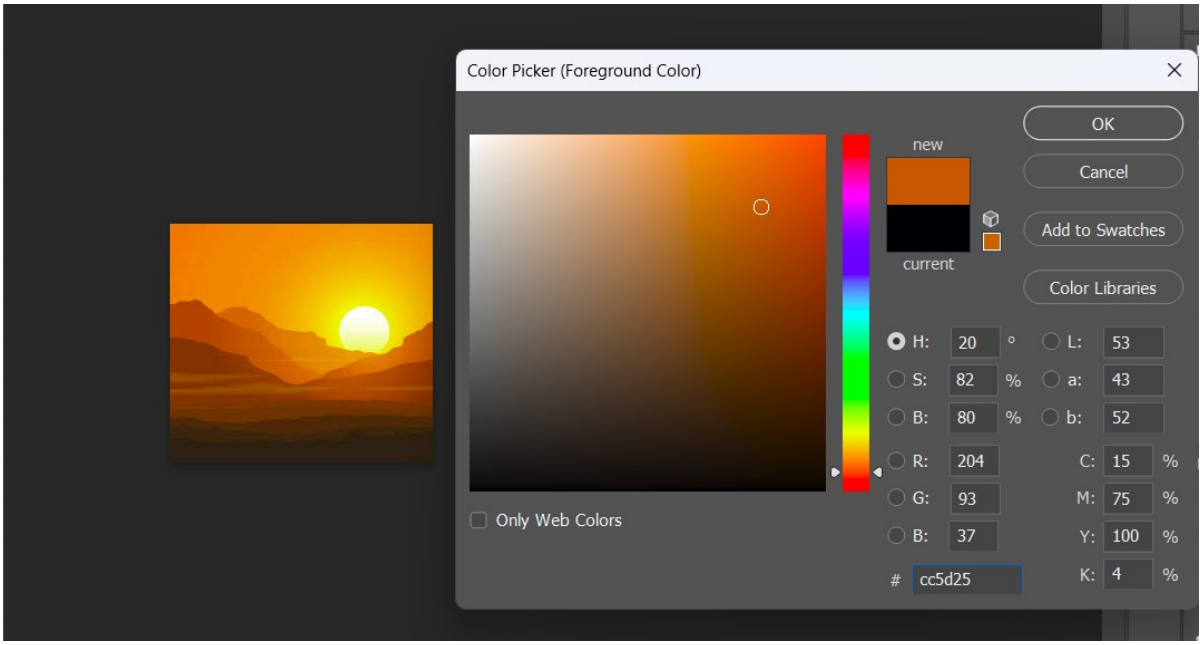
دلیل تفاوت نتایج

دلیل اصلی تفاوت نوع داده‌ها بود که در اینجا کلاس‌ها باینری نبودند و همچنین اینکه یک توازن بین target های ۰ و ۱ وجود داشت و اینطور نبود که تعداد مثبت‌ها بسیار زیاد باشد. زیرا که در پیاده‌سازی اول متوجه شدم کدی که من نوشته‌ام برای مثبت‌ها خیلی خوب جواب می‌دهد. در دیتاست دوم که تعداد منفی ها زیاد بود مدل کارایی نداشت.

۸_پاسخ سوال ۸)

روش‌های مختلفی برای کلاس‌بندی این نوع تصاویر وجود دارد. با نگاه کلی به تصاویر می‌توانیم متوجه شویم که تصویر مربوط به هوای ابری است یا غروب. باتوجه به مطالبی که در درس پردازش تصویر یاد گرفتیم، هر تصویر RGB از ۳ رنگ تشکیل شده است.

تصاویری که از غروب هستند، غالبا رنگ نارنجی دارند. این مورد را در نرم افزار فتوشاپ بررسی کردیم و متوجه شدیم که اگر ۳ رنگ را استخراج کنیم این نتیجه حاصل می‌شود:



می‌توان گفت که مقدار لایه R یا قرمز از دو لایه دیگر بیشتر است. می‌توانیم همین موضوع را مبنا قرار داده و هر تصویر را بر اساس آن بسنجیم.

در این روش رنگ هر پیکسل استخراج میشود. که یک ماتریس ۳ بعدی است. کافی است میانگین ۳ رنگ را محاسبه کنیم. اگر که مقدار لایه R از دو لایه دیگر بیشتر بود، آن را لیبل S را غروب بزنیم.

```
def predict_label(avg_color):
    if avg_color[2] > avg_color[1] and avg_color[2] > avg_color[0]:
        return 's'  # Sunset
    else:
        return 'c'  # Cloudy
```

نتیجه استفاده از این روش برای ارزیابی به صورت زیر است:

```
accuracy,precision,recall
(0.8125, 0.7358490566037735, 0.975)
[[26, 14], [1, 39]]
```

حال به بررسی تصاویری می‌پردازیم که در این روش خطا دارند:

```
result_str = "\n".join(f"{filename}: {'Cloudy' if label == 'c' else 'Sunset'}" for filename, label in zip(image_filenames, predi
print(result_str)
```

✓ 0.0s

```
c1.jpg: Cloudy
c10.jpg: Cloudy
c11.jpg: Cloudy
c12.jpg: Cloudy
c13.jpg: Cloudy
c14.jpg: Cloudy
c15.jpg: Cloudy
c16.jpg: Cloudy
c17.jpg: Sunset
c18.jpg: Cloudy
c19.jpg: Cloudy
c2.jpg: Cloudy
c20.jpg: Sunset
c21.jpg: Sunset
c22.jpg: Cloudy
c23.jpg: Sunset
c24.jpg: Sunset
```

در این روش تصاویری به اشتباه تشخیص داده شده‌اند که رنگ غالب تصویر نزدیک به نارنجی است ولی تصویر در اصل ابری است. باتوجه به ماتریس آشفتگی بیشتر اشتباهات از تصاویر ابری است. زیرا که میانگین لایه R بیشتر از دو لایه دیگر نخواهد بود. نمونه ای از این نوع تصاویر:

پس باید یک روش بهتری پیدا کنیم که مشکل تشخیص تصاویر ابری را نداشته باشد.

برای اینکار یک بازه برای هر کلاس تعریف می‌کنیم که اگر میانگین با آن تفاوت داشت، پیش‌بینی انجام شود. ویا بر اساس فیلترهای مختلف و حاشیه‌ها به بررسی بپردازیم. ولی خب باتوجه به بررسی‌های بیشتر متوجه شدیم که برخی تصاویر هم غروب هستند هم ابری. و مشکل خطا بیشتر مروبوط به لیبل است.