**به نام خدا**

# گزارش تمرین NaiveBayes

**ریحانه آهنی ۹۸۲۳۰۰۹**

**پاییز ۱۴۰۱**

در ابتدا باید کتابخانه های مورد نیاز را import کنیم.

```
import pandas as pd

from sklearn.naive_bayes import GaussianNB, BernoulliNB
from sklearn.model_selection import train_test_split, KFold
from sklearn.metrics import accuracy_score, confusion_matrix

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

سپس دیتاست را با استفاده از کتابخانه pandas بارگزاری می کنیم

```
dataset = pd.read_csv('BreastCancer.csv')
dataset.head()
```

و چند سطر اول آن را برای بررسی صحت در باز کردن دیتاست مشاهده می کنیم.

| Sample code number | Clump Thickness | Uniformity of Cell Size | Uniformity of Cell Shape | Marginal Adhesion | Single Epithelial Cell Size | Bare Nuclei | Bland Chromatin | Normal Nucleoli | Mitoses | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| 1000025 | 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 0 |
| 1002945 | 5 | 4 | 4 | 5 | 7 | 10 | 3 | 2 | 1 | 0 |
| 1015425 | 3 | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 1 | 0 |
| 1016277 | 6 | 8 | 8 | 1 | 3 | 4 | 3 | 7 | 1 | 0 |
| 1017023 | 4 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 1 | 0 |

همانطور که مشاهده میشود، سطر اول صرفا ID هر Sample است و اطلاعاتی مهم برای استفاده مدل ها ندارد. سطر آخر نیز Class می باشد که ۰ به معنای benign و ۱ به معنای malignant می باشد و ما باید این مقدار را توسط مدل محاسبه کنیم.

پس از بررسی باید مقادیر X و Y را جدا کنیم.

و برای train و test داده جداسازی کنیم.

```
features = ['Clump Thickness', 'Uniformity of Cell Size',
       'Uniformity of Cell Shape', 'Marginal Adhesion',
       'Single Epithelial Cell Size', 'Bare Nuclei', 'Bland Chromatin',
       'Normal Nucleoli', 'Mitoses', 'Class']

Y = dataset['Class'].to_numpy()
X = dataset[features].to_numpy()
```

**بخش الف) نتایج مدل Bernoulli**

این مدل با استفاده از روش Naïve Bayes میزان دقت حدود ۹۱ درصد در داده تست دریافت می کند.

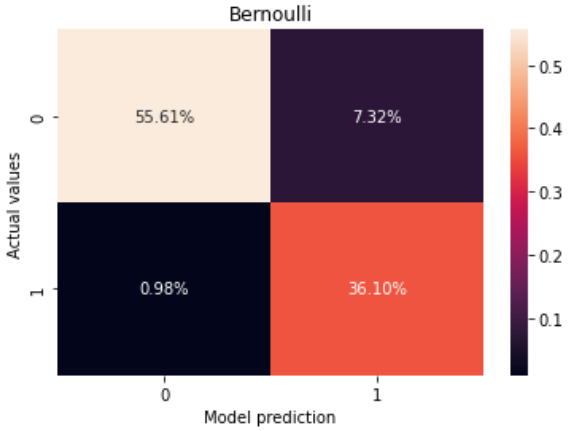
در برخی از مسائل، تشخیص صحیح نمونه‌های مربوط به یکی از دسته‌ها برای ما اهمیت بیشتری دارد. به عنوان مثال، دیتاست سرطان سینه را در نظر بگیرید که در آن، هدف شناسایی افراد مبتلا به بیماری سرطان سینه است. فرض کنید برای افرادی که مبتلا به این بیماری هستند، خطر مرگ وجود دارد و جهت رفع این خطر، نیاز به دریافت نوعی داروی خاص دارند. در این شرایط، تشخیص درست بیماران دارای اهمیت بسیار زیادی است.

به این معنا که خطا در تشخیص افراد سالم قابل چشم پوشی است اما برای شناسایی افراد بیمار نمی‌توان این احتمال را به جان خرید. به عبارت دیگر، انتظار ما تشخیص تمام افراد بیمار است، بدون جا انداختن، حتی اگر فرد سالمی به اشتباه جز افراد بیمار دسته‌بندی شود. در چنین مواقعی، که دقت و صحت تشخیص یک دسته در مقایسه با دقت و صحت تشخیص کلی، اهمیت بیشتری دارد، مفهوم ماتریس درهم‌ریختگی به کمک ما می‌آید.

بر اساس مثالی که پیش‌تر بیان شد، فرض کنید تعلق به دسته افراد بیمار را مثبت بودن و عدم تعلق به این دسته را منفی بود در نظر بگیریم. هر نمونه یا فردی در واقعیت، متعلق به یکی از کلاسهای مثبت یا منفی است و از سوی دیگر، از هر الگوریتمی که برای دسته‌بندی داده‌ها استفاده شود، در نهایت هر نمونه عضو یکی از این دو دسته دسته‌بندی خواهد شد. بنابراین برای هر نمونه داده، یکی از چهار حالتی که در ادامه بیان شده، ممکن است اتفاق بیفتد.

- نمونه عضو دسته مثبت باشد و عضو همین کلاس تشخیص داده شود ( مثبت صحیح )
- نمونه عضو کلاس مثبت باشد و عضو کلاس منفی تشخیص داده شود ( منفی کاذب )
- نمونه عضو کلاس منفی باشد و عضو همین کلاس تشخیص داده شود ( منفی صحیح )
- و در نهایت، نمونه عضو کلاس منفی باشد و عضو کلاس مثبت تشخیص داده شود منفی کاذب )

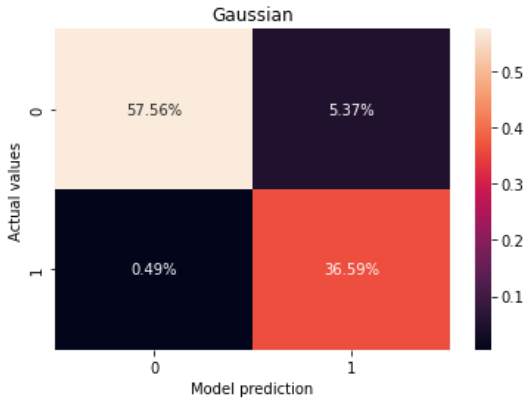
پس از اجرای الگوریتم دسته‌بندی، با توجه به توضیحات و تعاریف ذکر شده، می‌توان عملکرد یک طبقه‌بند را به کمک جدولی بررسی کرد.




تفسیر نمودار مقابل که خروجی مربوط به این بخش است موارد به شکل زیر است:

- منفی صحیح: ۵۵
- منفی کاذب: ۰.۹
- مثبت صحیح: ۳۶
- مثبت کاذب: ۷.۳۲

## بخش ب) نتایج مدل Gaussian




این مدل میزان دقت ۹۴ بر روی داده تست بدست می آورد.

تفسیر نمودار مقابل که خروجی مربوط به این بخش است موارد به شکل زیر است:

- منفی صحیح: ۵۷
- منفی کاذب: ۰.۴
- مثبت صحیح: ۳۶
- مثبت کاذب: ۵

مشاهده میشود که مدل Gaussian بهتر از Bernoulli عمل می کند و درصد خطای کمتری دارد.

## بخش پ) بهبود دقت مدل

برای بهبود دقت و کاهش منفی کاذب، می توان میزان اهمیت کلاس مثبت را افزایش داد، در این صورت مثبت کاذب و مثبت صحیح افزایش  و منفی صحیح و منفی کاذب کاهش پیدا می کنند.

برای پیاده سازی آن از پارامتر sample_weight در تابع fit استفاده می کنیم. این به اختیار تغییر اهمیت هر نمونه را می دهد. با اینکار ما اهمیت نمونه های منفی را پایین برده ( با وزن 1e-3 ) و نمونه های مثبت را تغییر نمی دهیم. با اینکار اهمیت کلاس مثبت بالاتر رفته و تعداد مثبت کاذب کاهش می یابد.