به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس یادگیری ماشین

تمرین اول

نام و نام خانوادگی	امیرحسین پورداود
شماره دانشجویی	۸۱۰۱۰۱۱۲۰
تاریخ ارسال گزارش	۱۴۰۲،۰۸،۰۶

فهرست

پاسخ ۱. طبقه بندی چند کلاسه

۱-۱. بخش الف

A) حالت دو کلاسه را در نظر می‌گیریم:

Bayes classifier

$$y = \arg\max_{y} P(y|x)$$

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Likelihood ratio

$$\frac{P(x|y=0)}{P(x|y=1)} \gtrless \frac{P(y=1)}{P(y=0)}$$

Error:

$$P(error) = P(y \neq \hat{y}) = P(y=1, \hat{y}=0) + P(y=0, \hat{y}=1)$$

$$= P(y=1, x \in R_2) + P(y=0, x \in R_1)$$

$$+ P(y=1, x \in R_1) - P(y=1, x \in R_1)$$

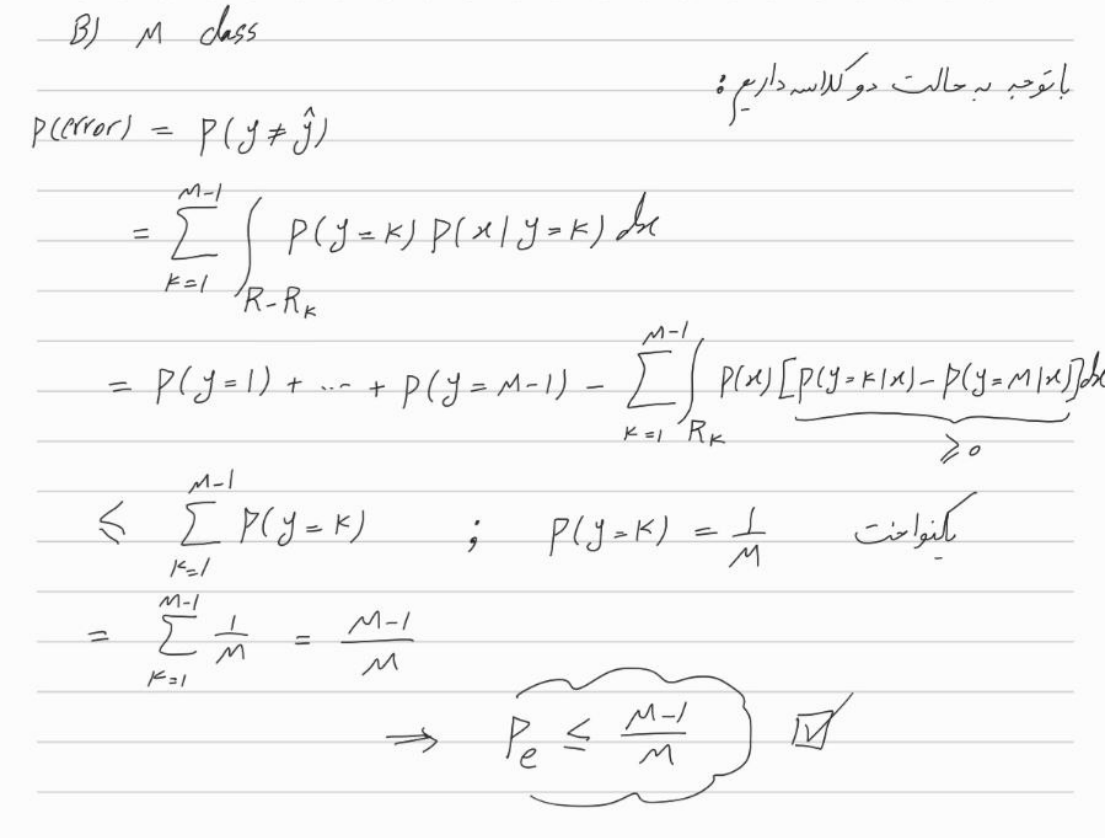
$$= P(y=1) + P(y=0, x \in R_1) - P(y=1, x \in R_1)$$

$$= P(y=1) + \int_{x \in R_1} P(y=0|x)P(x) - P(y=1|x)P(x)\, dx$$

$$P(error) = P(y=1) - \int_{x \in R_1} P(x)\underbrace{[P(y=1|x) - P(y=0|x)]}_{\geq 0} dx$$

در Bayese classifier، انتخاب کلاس بهینه براساس بیشینه بودن مقدار aposteriori انجام می‌شود، بنابراین مقدار $P(y=1|x) - P(y=0|x)$ در عبارت بالا همیشه مثبت است، پس انتگرال نیز مثبت شده و خطا را کمینه می‌کند. در حالت کلاس‌های بیشتر نیز مقدار بالا مثبت می‌شود که در قسمت بعد مشخص خواهد بود.

۱-۲. بخش ب



۱-۳. بخش ج

برای رسم نمودار ROC (Receiver Operating Characteristic) در حالت چند کلاسه، می‌توان از روش‌های زیر استفاده کرد:

۱. **روش یک در مقابل همه (One-vs-All):**

برای هر کلاس، یک classifier دوکلاسه را آموزش می‌دهیم که بین آن کلاس و سایر کلاس‌ها تمایز قائل شود. سپس از هر classifier ، نرخ درست‌نمایی (TP) و نرخ نادرست‌نمایی (FP) را محاسبه کنید و آنها را در نمودار ROC رسم می‌کنیم.

۲. **روش یک در مقابل دیگری (One-vs-One):**

برای هر جفت کلاس (M*(M-1)/2 جفت)، یک classifier دوکلاسه را آموزش میدهیم که بین دو کلاس در نظر گرفته شده تمایز قائل شود. سپس از هر classifier، نمودار ROC را رسم می‌کنیم.

۳. **روش چند کلاسه مستقیم:**

می‌توان یک classifier چندکلاسه را آموزش داد که بتواند بین تمام کلاس‌ها تمایز قائل شود. سپس با استفاده از تابع تصمیم classifier ، احتمال تعلق هر نمونه به هر کلاس را محاسبه کرد. سپس با استفاده از مقادیر احتمال تعلق به هر کلاس، می‌توانیم نمودار ROC را برای هر کلاس رسم کنیم.

در هر سه روش فوق، با محاسبه نرخ درست‌نمایی (True Positive Rate) و نرخ نادرست‌نمایی (False Positive Rate) برای هر کلاس، می‌توانید نقاط مختلف روی نمودار ROC را بدست آورید و آن را رسم کنید. سپس می‌توانید با استفاده از روش‌های تجمیعی مانند روش میانگین وزن‌دار و یا روش میکرو و ماکرو، نتیجه‌ای کلی از عملکرد دسته‌بند را در نمودار ROC چندکلاسه بدست آورید

۱-۴. بخش د

عملکرد بهینه Naive Bayes به ویژگی‌های مشخصی در مجموعه داده‌ها بستگی دارد. در زیر توضیح می‌دهیم که در چه شرایطی و به چه دلایلی Naive Bayes عملکرد بهینه دارد:

۱. **استقلال شرطی متغیرها:**

Naive Bayes با استفاده از فرض استقلال شرطی بین ویژگی‌ها کار می‌کند. به این معنی که فرض می‌کند وجود یک ویژگی در یک کلاس، به وجود دیگر ویژگی‌ها در همان کلاس وابسته نیست. در صورتی که این فرض درست باشد یا به طور نزدیکی برآورده شود، Naive Bayes می‌تواند عملکرد بهینه داشته باشد.

۲. **توزیع نمونه‌ها:**

Naive Bayes بر اساس توزیع نمونه‌ها و احتمالات شرطی کلاس‌ها کار می‌کند. اگر توزیع توأمان یا شرطی ویژگی‌ها در هر کلاس به خوبی توصیف شود و تفاوت قابل توجهی بین توزیع کلاس‌ها وجود داشته باشد، Naive Bayes می‌تواند عملکرد بهینه داشته باشد.

۳. **تعداد ویژگی‌ها:**

تعداد ویژگی‌ها در Naive Bayes تأثیر زیادی بر عملکرد دارد. در مجموعه داده‌هایی با تعداد ویژگی‌های کم و متناسب با اندازه مجموعه داده، Naive Bayes می‌تواند به صورت بهینه عمل کند. با افزایش تعداد ویژگی‌ها، احتمال وقوع همبستگی و وابستگی بین ویژگی‌ها افزایش می‌یابد که ممکن است باعث کاهش دقت Naive Bayes شود.

۴. **تعادل در توازن کلاس‌ها:**

در مجموعه داده‌هایی که توازن خوبی در تعداد نمونه‌های هر کلاس وجود دارد، Naive Bayes به خوبی عمل می‌کند. در صورتی که تعداد نمونه‌های یک کلاس نسبت به سایر کلاس‌ها بسیار کم یا بسیار زیاد باشد، ممکن است Naive Bayes در تشخیص و دسته‌بندی کلاس کمتر دقت کند.

۵. **مقاومت در برابر داده‌های ناقص و نویزی:**

Naive Bayes به دلیل سادگی خود مقاومت خوبی درنسبت به داده‌های ناقص و نویزی دارد. این به این معنی است که اگر مجموعه داده‌ها دارای مقادیر ناقص یا نادرست در برخی از ویژگی‌ها باشد، Naive Bayes می‌تواند به خوبی با این داده‌ها کار کند و دسته‌بندی دقیقی ارائه دهد.

در کل، Naive Bayes می‌تواند در مجموعه داده‌هایی که ویژگی‌ها مستقل از یکدیگر هستند، توزیع نمونه‌ها مشخص است و تعداد ویژگی‌ها متناسب با اندازه مجموعه داده است، عملکرد بهینه داشته باشد. همچنین، مقاومت Naive Bayes در برابر داده‌های ناقص و نویزی نیز یکی از مزایای آن است. با این حال، در مواردی که ویژگی‌ها بین خود وابستگی زیادی داشته باشند یا تعداد ویژگی‌ها بسیار زیاد باشد، دقت Naive Bayes کاهش خواهد یافت.

پاسخ ۲ – ناحیه تصمیم در طبقه بندی بیز

Q2/

$$p(x|y=1) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right) \quad x \geq 0$$

parameters: $\theta \geq 0$, $\sigma > 0$

$$p(x|y=2) = \theta x \cdot \exp(-\theta x) \quad x \geq 0$$

$$p(y=1) = p(y=2) = \frac{1}{2}$$

Bayesian classifier:

$$\hat{y} = \arg\max_i \; p(x|y_i)\,p(y_i)$$

$$p(x|y=1)\,p(y=1) \underset{2}{\overset{1}{\gtrless}} p(x|y=2)\,p(y=2)$$

$$x \geq 0 \longrightarrow \frac{x}{\sigma^2}\exp\left(\frac{-x^2}{2\sigma^2}\right) \underset{2}{\overset{1}{\lessgtr}} \theta x \exp(-\theta x)$$

$$\exp\left(-\frac{x^2}{2\sigma^2} + \theta x\right) \underset{2}{\overset{1}{\lessgtr}} \theta\sigma^2 \xrightarrow{\ln} -\frac{x^2}{2\sigma^2} + \theta x \underset{2}{\overset{1}{\lessgtr}} \ln\theta\sigma^2$$

$$\Rightarrow x^2 - 2\theta\sigma^2 x \underset{1}{\overset{2}{\gtrless}} -2\sigma^2\ln\theta\sigma^2$$

$$\Rightarrow (x-\theta\sigma^2)^2 - \theta^2\sigma^4 \underset{1}{\overset{2}{\lessgtr}} -2\sigma^2\ln\theta\sigma^2$$

$$\Rightarrow (x-\theta\sigma^2)^2 \underset{1}{\overset{2}{\gtrless}} \underbrace{\theta^2\sigma^4 - 2\sigma^2\ln\theta\sigma^2}_{\alpha}$$

class ②: $x-\theta\sigma^2 > \sqrt{\alpha} \Rightarrow x > \theta\sigma^2 + \sqrt{\alpha}$

$x-\theta\sigma^2 < -\sqrt{\alpha} \Rightarrow 0 \leq x < \theta\sigma^2 - \sqrt{\alpha}$ ☑

class ①: $-\sqrt{\alpha} < x-\theta\sigma^2 < \sqrt{\alpha} \Rightarrow 0 \leq x < \theta\sigma^2 + \sqrt{\alpha}$ ✓

ساده‌سازی انجام دهید:

$$\theta_6^2 \pm \sqrt{\alpha} = \theta_6^2 \pm \sqrt{\theta_6^4\left(1 - \frac{2}{\theta_6^2}\ln\theta_6^2\right)}$$

$$= \theta_6^2\left[1 \pm \underbrace{\sqrt{1 - \frac{2}{\theta_6^2}\ln\theta_6^2}}_{\simeq 1}\right] \simeq \begin{cases} \oplus & 2\theta_6^2 \\ \ominus & 0 \end{cases}$$

پس ساده‌سازی نواحی به صورت زیر درمی‌آید:

class ②: $x \geq 2\theta_6^2$

class ①: $0 \leq x \leq 2\theta_6^2$

IV

پاسخ ۳ – ماتریس ریسک

۳-۱. بخش الف

Q3/

$$\begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \begin{cases} w_1 \rightarrow y = 1 \\ w_2 \rightarrow y = 2 \end{cases}$$

A)

$$P(y=2) = 1 - P(y=1)$$

$$R(i|x) = \lambda_{i1} P(y=1|x) + \lambda_{i2} P(y=2|x) \Rightarrow \begin{cases} R(y=1|x) = P(y=2|x) \\ R(y=2|x) = P(y=1|x) \end{cases}$$

$$R = \int_{R_2} R(y=1|x) P(x) dx + \int_{R_1} R(y=2|x) P(x)$$

$$R = \int_{R_2} P(x|y=1) P(y=1) dx + \int_{R_1} P(x|y=2) P(y=2) dx$$

$$= P(y=1) \int_{R_2} P(x|y=1) dx + (1 - P(y=1)) \int_{R_1} P(x|y=2) dx$$

برای بدست آوردن prior بهینه داریم:

$$\frac{\partial R(P(y=1))}{\partial P(y=1)} = 0$$

$$= \int_{R_2} P(x|y=1) dx - \int_{R_1} P(x|y=2) dx = 0$$

$$\rightarrow \int_{R_2} P(x|w_1) dx = \int_{R_1} P(x|y=2) dx \quad \checkmark$$

۳-۲. بخش ب

B)

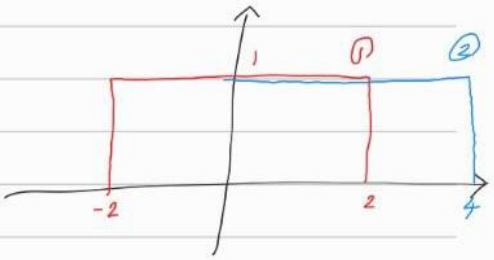
همیشه یکتا نیست.

بعنوان مثال نقض فرض کنیم prior ها برابر باشند $P(y=1) = P(y=2) = \frac{1}{2}$

فرض کنیم:

$$P(x|y=1) = \begin{cases} 1 & -2 \le x \le 2 \\ 0 & ow \end{cases}$$

$$P(x|y=2) = \begin{cases} 1 & 0 \le x \le 4 \\ 0 & ow \end{cases}$$



$$R_1 = [-2, 1] \quad R_1 = [0, 2]$$

پس با توجه به اینکه معادله بالا را ارضای کنند، جواب یکتا نیست.

پاسخ ۴ – تخمین MAP

Q4/

$x \sim N(\mu, \sigma^2)$ $\quad p(\mu) = \frac{\mu}{\sigma_\mu^2} \exp\left(\frac{-\mu^2}{2\sigma_\mu^2}\right)$

$$\hat{\mu} = \underset{\mu}{\operatorname{argmax}} \prod_{k=1}^{N} p(x_k|\mu)\, p(\mu)$$

$$= \underset{\mu}{\operatorname{argmax}} \underbrace{\prod_{k=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x_k-\mu)^2}{2\sigma^2}\right) \times \frac{\mu}{\sigma_\mu^2} \exp\left(\frac{-\mu^2}{2\sigma_\mu^2}\right)}_{L(\mu)}$$

$$l(\mu) = \log L(\mu) = -\frac{N}{2}\log 2\pi\sigma^2 + \log\frac{\mu}{\sigma_\mu^2} + \sum_{k=1}^{N}\left[\frac{-(x_k-\mu)^2}{2\sigma^2}\right] + \frac{-\mu^2}{2\sigma_\mu^2}$$

$$\frac{\partial l(\mu)}{\partial \mu} = \frac{1}{\mu} + \sum_{k=1}^{N}\left(\frac{x_k-\mu}{\sigma^2} - \frac{\mu}{\sigma_\mu^2}\right) = \frac{1}{\mu} - \frac{\mu}{\sigma_\mu^2} - \frac{N\mu}{\sigma^2} + \frac{1}{\sigma^2}\sum_{k=1}^{N} x_k$$

$$= \frac{1}{\mu} - \underbrace{\left(\frac{N}{\sigma^2} + \frac{1}{\sigma_\mu^2}\right)}_{R}\mu + \underbrace{\frac{1}{\sigma^2}\sum_{k=1}^{N} x_k}_{Z}$$

$$\Rightarrow \frac{\partial l(\mu)}{\partial \mu} = 0$$

$$\Rightarrow \frac{1}{\mu} - R\mu + Z = 0 \Rightarrow R\mu^2 - Z\mu - 1 = 0$$

$$\hat{\mu} = \frac{Z \pm \sqrt{Z^2+4R}}{2R} = \frac{Z + \sqrt{Z^2\left(1+\frac{4R}{Z^2}\right)}}{2R}$$

$$\Rightarrow \hat{\mu} = \frac{Z}{2R}\left(1 + \sqrt{1+\frac{4R}{Z^2}}\right) \quad \checkmark$$

پاسخ ۵ – تخمین ML

۵-۱. بخش الف

Q5/

$$f_Y(y|\theta) = \begin{cases} \frac{1}{\theta} r y^{r-1} e^{-\frac{y^r}{\theta}} & \theta>0,\ y>0 \\ 0 & ow \end{cases}, \quad \forall r>0$$

A)

$$L(\theta) = \prod_{i=1}^{N} f_Y(y_i|\theta) \rightarrow l(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log f_Y(y_i|\theta)$$

$$\log f_Y(y|\theta) = \log \frac{1}{\theta} r y^{r-1} e^{-\frac{y^r}{\theta}} = \log \frac{r}{\theta} + (r-1)\log y - \frac{y^r}{\theta}$$

$$l(\theta) = N\log r - N\log\theta + (r-1)\sum_{i=1}^{N} \log y_i - \frac{1}{\theta}\sum_{i=1}^{N} y_i^r \quad \checkmark$$

$$\frac{\partial l(\theta)}{\partial \theta} = -\frac{N}{\theta} + \frac{1}{\theta^2}\sum_{i=1}^{N} y_i^r = 0 \Rightarrow N\theta = \sum_{i=1}^{N} y_i^r \Rightarrow \hat{\theta} = \frac{1}{N}\sum_{i=1}^{N} y_i^r$$

۵-۲. بخش ب

B)

$$\theta^{MAP} = \arg\max \prod_{i=1}^{N} P(y_i|\theta)\, P(\theta)$$

$$\theta^{ML} = \arg\max \prod_{i=1}^{N} P(y_i|\theta)$$

درصورتی که توزیع prior پارامتر θ ثابت و برای همه داده‌ها یکسان باشد، تخمین MAP به ML میل می‌کند.

به بیان دیگر می‌توان گفت که وقتی prior بصورت یکنواخت باشد، داده‌ها بطور قابل توجهی تأثیر بیشتری در تخمین گذارند و اطلاعات اولیه کمتری در تخمین MAP در نظر گرفته میشود و به ML نزدیک می‌شود.

پاسخ ۶ – شبیه سازی Breast Cancer

۶-۱. بخش الف

Naive Bayes Classifier یک الگوریتم طبقه‌بندی احتمالاتی است که بر مبنای قاعده بیز مبتنی بر احتمالات کار می‌کند. این الگوریتم بر اصل سادگی و سرعت محاسباتی بالا متکی است و در بسیاری از مسائل طبقه‌بندی عملکرد خوبی دارد.

در Naive Bayes classifier، فرضیه استقلال شرطی بین ویژگی‌ها در نظر گرفته می‌شود. به عبارت دیگر، فرض می‌شود که وجود یک ویژگی در یک کلاس نسبت به وجود سایر ویژگی‌ها در همان کلاس وابستگی کمی دارد. این فرض "ساده‌ترین" شکل از استفاده از احتمالات بیز است، زیرا برای محاسبه احتمال نهایی یک کلاس بر اساس ویژگی‌ها، تنها باید احتمال هر ویژگی به شرط کلاس مورد نظر را محاسبه کنیم و سپس این احتمالات را با یکدیگر ضرب کنیم.

تفاوت ساختاری اصلی بین Naive Bayes classifier و یک classifier بیزی معمولی در فرض استقلال شرطی است. در classifier بیزی معمولی، هیچ فرضیه خاصی در مورد استقلال شرطی ویژگی‌ها ارائه نمی‌شود و محاسبات احتمالاتی بر اساس تمام ترکیب‌های ممکن از ویژگی‌ها صورت می‌گیرد. این به معنای این است که classifier بیزی معمولی برای مسائل با تعداد زیادی ویژگی، محاسبات پیچیده‌تری نسبت به Naive Bayes classifier دارد.

از Naive Bayes classifier به دلایل زیر استفاده می‌شود:

۱. **سرعت محاسباتی بالا:** به دلیل سادگی محاسبات و فرضیه استقلال شرطی، Naive Bayes classifier سریعتر از classifier بیزی معمولی است.
۲. **عملکرد خوب در مسائل با تعداد زیادی ویژگی:** در مسائلی که تعداد ویژگی‌ها زیاد است، محاسبات classifier Naive Bayes سریع‌تر از classifier بیزی معمولی است.
۳. **عملکرد قابل قبول در مسائل واقعی:** اگرچه فرض استقلال شرطی بین ویژگی‌ها در بسیاری از موارد واقعی برقرار نیست، اماز آنجا که Naive Bayes classifier فرضیه استقلال شرطی را ارائه می‌دهد و محاسبات سریعی را انجام می‌دهد.

در موارد زیر می‌تواند استفاده از این classifier منطقی باشد:

I. **مجموعه داده‌های بزرگ:** زمانی که مجموعه داده‌ها بزرگ و تعداد ویژگی‌ها زیاد است، استفاده از طبقه‌بند Naive Bayes می‌تواند مناسب باشد. به دلیل سرعت محاسباتی بالا، قابلیت پردازش سریع‌تر از طبقه‌بند بیزی معمولی را دارد.
II. **مجموعه داده‌های با تنوع بالا:** اگر مجموعه داده‌ها ویژگی‌های متنوعی داشته باشد و وجود یک ویژگی با وجود سایر ویژگی‌ها مستقل به نظر برسد، طبقه‌بند Naive Bayes قادر به دسته‌بندی مناسب خواهد بود.

III. **مسائل ساده:** Naive Bayes classifier در مسائل ساده و کوچک، که داده‌ها کم و ویژگی‌ها کمتر هستند، می‌تواند عملکرد قابل قبولی داشته باشد.

به طور کلی، اگر فرض استقلال شرطی بین ویژگی‌ها در مسئله موردنظر برقرار یا به طور نزدیکی برآورده شود و سرعت پردازش مهم باشد، استفاده از Naive Bayes classifier می‌تواند منطقی باشد. با این حال، در مواردی که وابستگی شدید بین ویژگی‌ها وجود دارد یا فرض استقلال شرطی برقرار نیست، روش‌های دیگری مانند classifier بیزی معمولی یا شبکه‌های عصبی ژرف ممکن است بهتر عمل کنند.

۶-۲. بخش ب

خروجی های خواسته شده با استفاده از روش Naïve Bayes و بدون استفاده از کتابخانه های رایج در پیاده سازی شده و کد قسمت ها ی مختلف در شکل های زیر موجود می باشد.

۱) ابتدا کتابخانه های مورد نیاز را اضافه و دیتا را لود میکنیم، سپس ۵ ردیف اول داده را برای درک و پردازش بهتر مشاهده میکنیم:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
```

0.0s Python

Read Data

```python
data = pd.read_csv('data.csv')
data.head()
```

0.0s Python

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_wo
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	17.33	184
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	23.41	158
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	25.53	152
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	26.50	98
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	16.67	152

5 rows × 33 columns

۲) سپس تعداد ستون ها و ویژگی های موجود را بررسی کرده و موارد غیر قابل استفاده را از مجموعه داده هایمان حذف میکنیم و تعداد لیبل ها را نیز مشخص میکنیم.

```python
# feature names as a list
col = data.columns
print(col)
```

0.0s

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

```python
list = ['Unnamed: 32','id','diagnosis']

# y includes our labels and x includes our features
y = data.diagnosis # M or B
x_ = data.drop(list,axis = 1 )

B, M = y.value_counts()
print('Number of Benign: ',B)
print('Number of Malignant : ',M)
```

0.0s

```
Number of Benign:  357
Number of Malignant :  212
```

۳) مشخصات هر ویژگی اعم از میانگین و واریانس و بیشینه و کمینه و ... را مشاهده کرده و در فرایند feature selection را آغاز میکنیم.

```
1  x_.describe()
```
✓ 0.1s

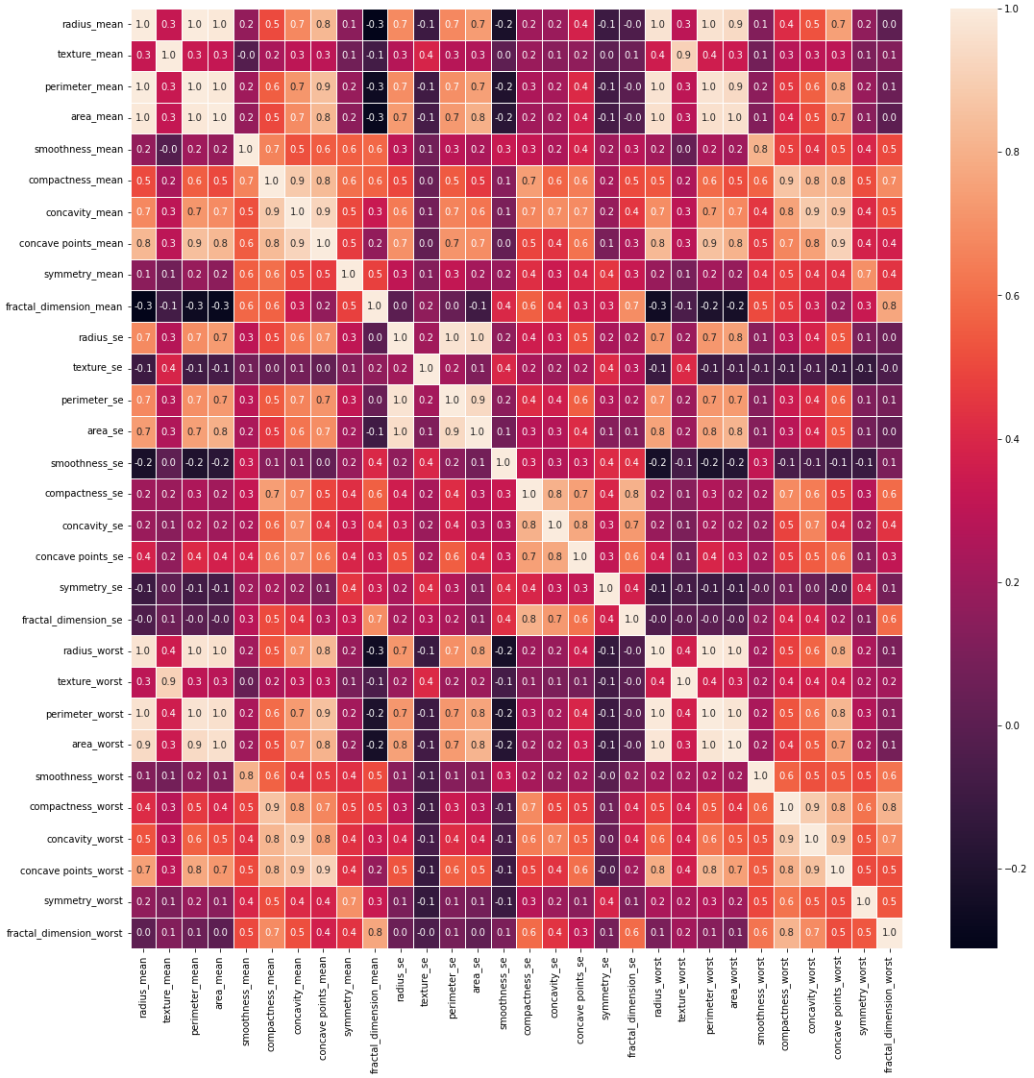
	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean	...
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	...
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798	...
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060	...
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960	...
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700	...
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540	...
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120	...
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440	...

8 rows × 30 columns

۴) میزان correlation هر بردار ویژگی با بردار های ویژگی دیگر را برای انتخاب بهینه ویژگی ها بررسی میکنیم که در اینجا نمودار heaatmap رسم شده است:

```
1  # correlation map
2  f,ax = plt.subplots(figsize=(18, 18))
3  sns.heatmap(x_.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
```
✓ 4.1s

```
<AxesSubplot:>
```



همانطور که در شکل مشاهده می شود radius_mean ، perimeter_mean و area_mean با یکدیگر همبستگی دارند، بنابراین ما فقط area_mean را استفاده خواهیم کرد. نحوه انتخاب یک ویژگی مورد استفاده به این صورت است که فقط به نمودار ها نگاه می‌کنیم و بطور مثال در area_mean واضح به نظر می‌رسد که بهتر است. اما ما نمی‌توانیم بدون تلاش، جداسازی دقیقی را بین سایر ویژگی‌های مرتبط انجام دهیم.

- Compactness_mean، concavity_mean و concave points_mean با یکدیگر همبستگی دارند. بنابراین ما فقط concavity_mean را انتخاب می کنیم.
- جدای از اینها radius_se، perimeter_se و area_se همبستگی دارند و ما فقط از area_se استفاده می کنیم.
- radius_worst، perimeter_worst و area_worst همبستگی دارند، بنابراین ما از area_worst استفاده می کنیم.
- Compactness_Worst، Concavity_Worst و Concav points_Worst بنابراین ما از concavity_worst استفاده می کنیم.
- Compactness_se، concavity_se و concave points_se بنابراین ما از concavity_se استفاده می کنیم.
- texture_mean و texture_worst همبستگی دارند و ما از texture_mean استفاده می کنیم. area_worst و area_mean همبستگی دارند، ما از area_mean استفاده می کنیم.

۵) ویژگی های گفته شده را حذف کرده و دوباره نمودار correlation را مشاهده کرده تا از انتخاب خود مطمعن شویم:

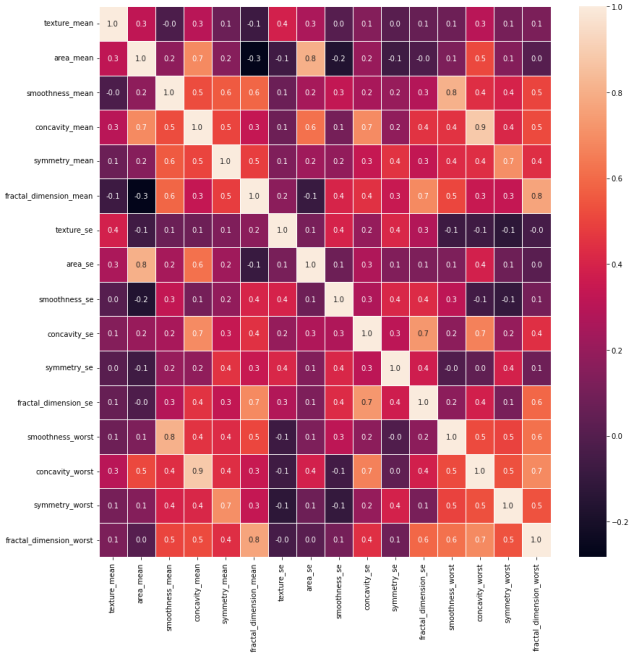
```
drop_list1 = ['perimeter_mean','radius_mean','compactness_mean','concave points_mean',\
              'radius_se','perimeter_se','radius_worst','perimeter_worst',\
              'compactness_worst','concave points_worst','compactness_se',\
              'concave points_se','texture_worst','area_worst']
X = x_.drop(drop_list1,axis = 1 ) # do not modify x, we will use it later
X.head()
```

✓ 0.0s

	texture_mean	area_mean	smoothness_mean	concavity_mean	symmetry_mean	fractal_dimension_mean	texture_se	area_se	smoothness_se	concavity_se	symmetry_se	fractal_di
0	10.38	1001.0	0.11840	0.3001	0.2419	0.07871	0.9053	153.40	0.006399	0.05373	0.03003	
1	17.77	1326.0	0.08474	0.0869	0.1812	0.05667	0.7339	74.08	0.005225	0.01860	0.01389	
2	21.25	1203.0	0.10960	0.1974	0.2069	0.05999	0.7869	94.03	0.006150	0.03832	0.02250	
3	20.38	386.1	0.14250	0.2414	0.2597	0.09744	1.1560	27.23	0.009110	0.05661	0.05963	
4	14.34	1297.0	0.10030	0.1980	0.1809	0.05883	0.7813	94.44	0.011490	0.05688	0.01756	

```
#correlation map
f,ax = plt.subplots(figsize=(14, 14))
sns.heatmap(X.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
```

✓ 1.2s



۶) در قسمت اخر میانگین و واریانس هر کلاس را بدست آورده و بوسیله قاعده بیز کلاس بندی میکنیم:

```python
# Calculating mean and std for 2 class.
X0_mean = np.mean(X_train[y_train == 0])
print('_____ \n', 'X0 Mean : \n _____ \n', X0_mean)
X0_std = np.std(X_train[y_train == 0])
print('_____ \n', 'X0 STD : \n _____ \n', X0_std)
X1_mean = np.mean(X_train[y_train == 1])
print('_____ \n', 'X1 Mean : \n _____ \n', X1_mean)
X1_std = np.std(X_train[y_train == 1])
print('_____ \n', 'X1 STD : \n _____ \n', X1_std)
```

✓ 0.0s

```
_____
 X0 Mean :
 _____
texture_mean              17.836948
area_mean                468.995582
smoothness_mean            0.092163
concavity_mean             0.046494
symmetry_mean              0.173019
fractal_dimension_mean     0.062844
texture_se                 1.195043
area_se                   21.258100
smoothness_se              0.007172
concavity_se               0.026462
symmetry_se                0.020166
fractal_dimension_se       0.003668
smoothness_worst           0.124551
concavity_worst            0.166273
symmetry_worst             0.267414
fractal_dimension_worst    0.079606
dtype: float64
_____
 X0 STD :
 _____
texture_mean               3.951366
```

```python
test_predictions = []
confusion_matrix_ = np.array([[0, 0], [0, 0]])

for i in range(len(y_test)):
    if NB_prob(normalize1(X_test.iloc[i])) * 1/len(X_train[y_train == 1]) > NB_prob(normalize0(X_test.iloc[i])) * 1/len(X_train[y_train == 0]):
        test_predictions.append(1)
    elif NB_prob(normalize1(X_test.iloc[i])) * 1/len(X_train[y_train == 1]) < NB_prob(normalize0(X_test.iloc[i])) * 1/len(X_train[y_train == 0]):
        test_predictions.append(0)
    confusion_matrix_[y_test[i]][test_predictions[i]] += 1

accurracy = (confusion_matrix_[0][0] + confusion_matrix_[1][1]) / len(y_test)
precision = confusion_matrix_[0][0] / (confusion_matrix_[0][0] +  confusion_matrix_[0][1])
recall = confusion_matrix_[0][0] / (confusion_matrix_[0][0] +  confusion_matrix_[1][0])


print("The confusion matrix is:")
print(confusion_matrix_)
print("The accuracy is:  %f " %(accurracy*100) )
print("The precision is: %f "  %(precision*100))
print("The recall is: %f" %(recall*100))
```

✓ 0.6s

```
The confusion matrix is:
[[89 19]
 [ 2 61]]
The accuracy is:  87.719298
The precision is: 82.407407
The recall is: 97.802198
```

۶-۳. بخش پ

این بار با استفاده از کتابخانه همان کار قسمت ب را انجام داده ایم و خروجی های خواسته شده را در شکل زیر بدست آورده ایم:

Part C

```
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)

confusion_matrix_ = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print('Confusion matrix is:')
print(confusion_matrix_)
print('My Acurracy matrix is: %f' %(accuracy*100) )
print("The precision is: %f "  %(precision*100))
print("The recall is: %f" %(recall*100))
```

0.0s

```
Confusion matrix is:
[[103   5]
 [  7  56]]
My Acurracy matrix is: 92.982456
The precision is: 91.803279
The recall is: 88.888889
```

همانطور که دیده می شود دقت کتابخانه Sklearn از الگوریتم دستی پیاده شده بهتر می باشد.

پاسخ ۷ – شبیه سازی طبقه بندی دو کلاسه

در این سوال قصد داریم با استفاده از میانگین رنگ عکس های داده شده و مقایسه با آبی و سبز تشخیص بدهیم که عکس داده شده مربوط به دریا (آبی) است یا متعلق به جنگلی(سبز) می باشد.

ابتدا معیار برای رنگ های آبی و سبز را مشخص می کنیم و سپس با استفاده از روش Maximum Likelihood کلاسی که کمترین فاصله را از میانگین سبز و آبی یک تصویر دارد انتخاب می شود.

بردار های معیار برای سبز بصورت (۰، ۲۵۵، ۰) و برای قرمز بصورت (۰، ۰، ۲۵۵) میباشد.

2 Class Classifier

```
1  jungle = np.array([0, 255, 0])
2  sea = np.array([0, 0, 255])
```

✓ 0.0s

نتایج نهایی بصورت زیر خواهد بود:

Results

+ Code + Markdown

```
1  accuracy = (TP + TN) / (TP + FP + TN + FN) * 100
2  precision = TP / (TP + FP) * 100
3  recall = TP / (TP + FN) * 100
4
5  print("confusion matrix is:")
6  print(confusion_mat)
7  print('--------------------------')
8  print("accuracy is: " + str(accuracy)+ "%")
9  print("precision is: " + str(precision))
10 print("recall is: " + str(recall))
```

✓ 0.0s

```
confusion matrix is:
[[40  0]
 [ 1 41]]
--------------------------
accuracy is: 98.78048780487805%
precision is: 100.0
recall is: 97.5609756097561
```

درصد دقت به ما نشان می دهد که تصاویر با دقت خوبی طبقه بندی شده اند.ولی در مواردی که رنگ ها مشابه کلاس دیگر باشد یا در طیف آن رنگ نباشد، در آنصورت این روش ممکن است نتواند به خوبی عمل کند چون ممکن است فاصله میانگین عکس از میانگین رنگ عکس دیگر کمتر باشد و عکس به اشتباه تشخیص داده شود. همچنین دقت روش به ما حس نمیدهد که تشخیص دهیم این روش خوب عملکرده است و یا نه در واقع precisionو recallتوصیف کننده بهتری برای مقایسه این روش با سایر روش ها می باشد.

در این قسمت میخواهیم، نتایج را برای داده هایی که اشتباه بررسی شده است را بررسی کنیم.

در مجموع ۳ داده کلاس آن اشتباه محاسبه شده که بصورت زیر مشاهده میشود:

The true label is *Jungle* and predicted as a *Sea*

The true label is *Jungle* and predicted as a *Sea*

The true label is *Sea* and predicted as a *Jungle*

در داده اول و دوم به دلیل سفید بودن و نزدیک بودن این رنگ و همچنین آسمان موجود در آن، باعث شده که این اشتباه رخ دهد و مدل نتواند به خوبی تشخیص دهد.

در داده سوم نیز به دلیل سبز بودن رنگ آب دریا، مدل آن را جنگل تشخیص داده است.