1. Naive Bayes Əsasları 📊

1.1. Naive Bayes-in Tərifi və Əsas Məntiqi

Naive Bayes — statistik, ehtimal əsaslı bir klassifikasiya alqoritmidir. O, Bayes teoreminə əsaslanır və fərz edir ki, verilənlərin xüsusiyyətləri (features) bir-birindən müstəqildir.

- Adındakı "Naive" (sadəlövh) hissəsi məhz bu müstəqillik fərziyyəsindən gəlir.
- Sadə görünsə də, mətn təsnifatı 🍃, spam filtrləmə 📩, tibbi diaqnostika 🖺 kimi sahələrdə çox uğurlu nəticələr verir.

Əsas məntiq:

- 1. Əvvəlcə hər bir sinifin **prior ehtimalını** tapırıq: $P(C_k)$
- 2. Daha sonra verilən xüsusiyyətlərin həmin sinifdə olma ehtimallarını tapırıq: $P(x_i|C_k)$
- 3. Bayes teoremi ilə verilən nümunənin hansı sinifə aid olduğunu müəyyən edirik.

1.2. Bayes Teoremi və Formulu 📐



Bayes teoremi ehtimal nəzəriyyəsinin əsas bünövrələrindən biridir.

Formul:

$$P(C_k|X) = rac{P(X|C_k) \cdot P(C_k)}{P(X)}$$

Burada:

- C_k k-cı sinif
- X verilən xüsusiyyətlər vektoru (məs: (x_1, x_2, \ldots, x_n))
- ullet $P(C_k|X)$ **posterior ehtimal**: X müşahidə olunduqda C_k sinifinin ehtimalı
- $P(X|C_k)$ **likelihood**: C_k sinifində X-in ehtimalı
- $P(C_k)$ **prior ehtimal**: ümumilikdə C_k sinifinin ehtimalı
- P(X) evidence: bütün siniflərdə X-in ehtimalı

Naive Bayes-da:

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k)$$

çünki fərz edilir ki, xüsusiyyətlər bir-birindən müstəqildir.

1.3. "Naive" (Müstəqillik) Fərziyyəsi 🔍



Naive Bayes-in ən böyük sadələşdirməsi:

"Bütün xüsusiyyətlər sinif daxilində bir-birindən tam müstəqildir"

Məsələn:

- E-mail spam filtri nümunəsi:
 - Xüsusiyyətlər: ["pulsuz" sözünün olması, "klik et" ifadəsinin olması, mesajın uzunluğu]
 - Naive Bayes fərz edir ki, "pulsuz" sözünün olması ilə "klik et" ifadəsinin olması birbirinə təsir etmir.
 - Əslində real həyatda bunlar bağlıdır, amma bu fərziyyə hesablamanı çox sadələşdirir.

Üstünlükləri 🔽 : Hesablama sürətlidir, az miqdarda verilənlərlə belə işləyir.

Çatışmazlıqları A: Xüsusiyyətlər arasında güclü əlaqələr olduqda performans pisləşə bilər.

1.4. Ehtimal Əsaslı Klassifikasiya Yanaşması 🎯



Naive Bayes bir **ehtimal modeli** olduğu üçün qərar qaydası belədir:

Bir müşahidə üçün:

$$\hat{C} = rg \max_{C_k} P(C_k|X)$$

Əməli olaraq:

- 1. Hər sinif üçün **posterior ehtimal** hesabla
- 2. Ən böyük ehtimala malik sinifi seç

Məsələn:

- Verilənlər: Hava Günəşlidir 🐥, Temperatur Yüksəkdir 🍾
- Siniflər: [Futbol oynamaq 🚯 , Oynamamaq 🚫]
- Hesablanır:
 - P(Oynamaq|Günəşli, Yüksək)
 - P(Oynamamaq|Günəşli, Yüksək)
- Hansı böyükdürsə, onu seçirik.

In []:

Oynamamaq: 0.05 Oynamaq: 0.95

Seçilmiş sinif: Oynamaq

1.5. Naive Bayes-in Üstünlükləri və Çatışmazlıqları



- Sadə və sürətli 🕴 (həm təlim, həm proqnoz mərhələsində)
- Kiçik dataset-lərdə belə yaxşı nəticə 📈
- Mətn təsnifatında yüksək performans 🍃
- Səs-küylü verilənlərə qarşı davamlıdır 🌗

Çatışmazlıqlar:

- Xüsusiyyətlər arasındakı əlaqələri nəzərə almır 🔗
- Numerik dəyişənlərdə düzgün paylanma fərziyyəsi (məs: Gaussian) lazım olur 📊
- Sıfır ehtimal problemi (Laplace smoothing lazım ola bilər) 🛠

2. Naive Bayes Növləri 🧠

Naive Bayes ailəsində müxtəlif variantlar var və onların hər biri fərqli tip verilənlər üçün optimallaşdırılıb. Əsas fərqlər **xüsusiyyətlərin paylanma modeli** 🔝 və **ehtimal hesablanma üsulu** 🔈 ilə bağlıdır.

2.1. Gaussian Naive Bayes (Numeric Verilənlər üçün)

• İşləmə prinsipi:

Gaussian Naive Bayes fərz edir ki, hər bir xüsusiyyət **normal (Gaussian) paylanmaya** malikdir ...

Bu o deməkdir ki, hər bir xüsusiyyətin ehtimal sıxlığı belə ifadə olunur:

$$P(x_i|C_k) = rac{1}{\sqrt{2\pi\sigma_k^2}} \mathrm{exp}igg(-rac{(x_i-\mu_k)^2}{2\sigma_k^2}igg)$$

Burada:

- lacksquare μ_k həmin xüsusiyyətin C_k sinfində orta dəyəri
- $lacksquare \sigma_k^2$ həmin xüsusiyyətin C_k sinfində dispersiyası

• Harada istifadə olunur:

- Numerik (real ədədlər) verilənlərdə, məsələn:
 - Tibbi ölçmələr (qan təzyiqi, şəkər səviyyəsi)
 - Sensor məlumatları
 - Ölçü və faiz tipli verilənlər

• Üstünlükləri:

- Sürətli hesablama +
- Kiçik dataset-lərdə də yaxşı işləyir

Çatışmazlıqları:

Verilənlər normal paylanmaya yaxın olmalıdır, əks halda nəticə zəifləyə bilər

Gaussian Naive Bayes Nəticələri:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
1	1.00	0.92	0.96	13
2	0.93	1.00	0.96	13
accuracy			0.98	45
macro avg	0.98	0.97	0.97	45
weighted avg	0.98	0.98	0.98	45

2.2. Multinomial Naive Bayes (Diskret Dəyərlər üçün) 🔢

• İşləmə prinsipi:

Multinomial Naive Bayes **sayma əsaslı verilənlər** üçün nəzərdə tutulub. Ən çox **mətn təsnifatında** istifadə olunur .

Burada ehtimallar xüsusiyyətlərin frekanslarına görə hesablanır:

$$P(X|C_k) = rac{(n_{c_1} + lpha) \cdot (n_{c_2} + lpha) \cdot \cdot \cdot}{(N_c + lpha \cdot m)^n}$$

Burada:

- $lacktriangledown n_{c_i}$ xüsusiyyətin sinifdə neçə dəfə göründüyü
- N_c sinifdəki bütün xüsusiyyətlərin toplam sayı
- *m* unikal xüsusiyyətlərin sayı
- α Laplace smoothing parametri %

• Harada istifadə olunur:

- Mətn təsnifatı (spam filtr 1, sentiment analizi 2)
- Sözlərin və ya tokenlərin tezliyinin ölçüldüyü hallarda

• Üstünlükləri:

- Mətn təsnifatı üçün çox uğurludur
- Yüksək ölçülü (high-dimensional) verilənlərdə yaxşı işləyir

• Çatışmazlıqları:

Yalnız sayma əsaslı verilənlərdə effektivdir

In []:

Multinomial Naive Bayes Nəticələri:

	•			
support	f1-score	recall	precision	
183	0.99	0.99	0.99	0
174	0.99	0.99	0.99	1
357	0.99			accuracy
357	0.99	0.99	0.99	macro avg
357	0.99	0.99	0.99	weighted avg

2.3. Bernoulli Naive Bayes (Binary üçün) 0 1

• İşləmə prinsipi:

Bernoulli Naive Bayes xüsusiyyətlərin **binary (0 və ya 1)** olduğu hallarda istifadə olunur. Yəni hər xüsusiyyət yalnız "var" və ya "yox" kimi qiymət alır. Ehtimallar belə hesablanır:

■ Əgər xüsusiyyət mövcuddursa:

$$P(x_i = 1|C_k)$$

■ Əgər mövcud deyilsə:

$$1 - P(x_i = 1 | C_k)$$

• Harada istifadə olunur:

- Mətn təsnifatı (sözün olub-olmaması əsasındadır, tezlik yox)
- Anket və sorğular (bəli/xeyr tipli cavablar)
- Boolean tipli xüsusiyyətlər ○

• Üstünlükləri:

- Sadə və effektivdir 🔸
- Var/Yox tipli xüsusiyyətlər üçün ideal 6

• Çatışmazlıqları:

Tezlik məlumatlarını istifadə etmir, yalnız mövcudluğa baxır

2.4. Complement Naive Bayes (Disbalanced Verilənlər üçün) 🔱

• İşləmə prinsipi:

Complement Naive Bayes, **sinif balanssızlığı** \Lambda olan dataset-lərdə daha yaxşı nəticə vermək üçün hazırlanıb.

Əsas fərqi odur ki, ehtimallar hesablanarkən **müvafiq sinifdən başqa bütün siniflərin statistikası** istifadə olunur **1**.

Yəni "complement" anlayışı — bir sinifin tamamlayıcısı mənasındadır 🗟 .

• Harada istifadə olunur:

- Mətn təsnifatında (xüsusilə class imbalance olduqda)
- Az sayda nümunəyə malik siniflərin qorunması lazım olanda •

• Üstünlükləri:

- Multinomial NB-yə alternativ kimi daha sabit nəticələr verir

• Çatışmazlıqları:

- Hesablama bir az daha mürəkkəbdir <a>B
- Kiçik dataset-lərdə Multinomial NB qədər stabil olmaya bilər

Complement Naive Bayes Nəticələri:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	172
1	1.00	0.94	0.97	52
accuracy			0.99	224
macro avg	0.99	0.97	0.98	224
weighted avg	0.99	0.99	0.99	224

3. Gaussian Naive Bayes-ə Xas Mövzular



Gaussian Naive Bayes modeli, numerik (real ədədlər) verilənlər üzərində işləyərkən ən çox istifadə olunan Naive Bayes variantıdır. Bu model xüsusiyyətlərin normal (Gaussian) paylanmaya malik olduğunu fərz edir və bütün ehtimal hesablamaları bu fərziyyəyə əsaslanır.

3.1. Gaussian Paylanma Fərziyyəsi 🔔



Gaussian Naive Bayes-də əsas fərziyyə belədir:

Hər bir xüsusiyyət x_i müəyyən bir sinif C_k daxilində **normal paylanır**. Yəni:

$$x_i|C_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$$

Normal paylanmanın sıxlıq funksiyası:

$$P(x_i|C_k) = rac{1}{\sqrt{2\pi\sigma_k^2}} \mathrm{exp}igg(-rac{(x_i-\mu_k)^2}{2\sigma_k^2}igg)$$

Misal:

Əgər tələbələrin imtahan nəticələri (0-100 arası) müəyyən bir sinifdə normal paylanırsa, o zaman Gaussian Naive Bayes bu nəticələrin ehtimal sıxlığını yuxarıdakı formul ilə hesablaya bilər.

3.2. Orta (μ) və Dispersiya (σ^2) Hesablama \searrow



Gaussian Naive Bayes-də hər bir xüsusiyyətin hər bir sinifdə orta və dispersiya dəyərləri ayrıca hesablanır.

Orta:

$$\mu_k = rac{1}{N_k} \sum_{j=1}^{N_k} x_j$$

Dispersiya:

$$\sigma_k^2 = rac{1}{N_k} \sum_{i=1}^{N_k} (x_j - \mu_k)^2$$

Burada:

- $N_k C_k$ sinifindəki nümunələrin sayı
- x_i xüsusiyyətin qiyməti

Misal:

Tibbi dataset-də "qan təzyiqi" xüsusiyyəti üçün:

- C_1 : xəstə
- C_2 : sağlam

Hər sinif üçün ayrı μ_k və σ_k^2 hesablanır.

3.3. Ədədi Xüsusiyyətlər üçün Ehtimal Hesabı 📊

Əgər bizdə ədədi bir xüsusiyyət varsa (məsələn, boy, çəki, temperatur), onun ehtimalını tapmaq üçün normal paylanma sıxlıq funksiyası istifadə olunur.

Hər xüsusiyyətin ehtimalı hesablanır, sonra Naive Bayes fərziyyəsinə görə müstəqillik tətbiq edilərək bütün ehtimallar hasillənir:

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k)$$

3.4. Parametr Təxmin Metodları (MLE)



Gaussian Naive Bayes-də orta (μ) və dispersiya (σ^2) dəyərləri **Maximum Likelihood** Estimation (MLE) üsulu ilə tapılır.

MLE prinsipi — verilənləri ən çox ehtimalla izah edən parametr dəyərlərini tapmaqdır.

Normal paylanma üçün MLE nəticələri:

- ullet μ_k nümunə ortalaması
- σ_k^2 nümunə dispersiyası

Yəni yuxarıda verdiyimiz formullar əslində MLE nəticəsidir.

3.5. Logaritmik Ehtimal İstifadəsi (Rəqəmsal Sabitlik) 👪

Çoxlu xüsusiyyət olduqda ehtimalların hasili çox kiçik ədədlərə gətirib çıxara bilər (floating point underflow problemi).

Bunu həll etmək üçün **log ehtimallar** istifadə olunur:

$$\log P(X|C_k) = \sum_{i=1}^n \log P(x_i|C_k)$$

Logarifm istifadəsi:

Çox kiçik ədədlərin sıfıra yuvarlaglaşmasının garşısını alır

- Hesablama dəqiqliyini artırır
- Hasili toplama əməliyyatına çevirir (daha sürətli)

4. Naive Bayes-in Riyazi Əsasları 📐

Naive Bayes modeli tamamilə **Bayes Teoreminə** və **sadə müstəqillik fərziyyəsinə** əsaslanır. Bu hissədə modelin riyazi təməllərini addım-addım izah edəcəyik.

4.1. Bayes Teoreminin Ətraflı İzahı

Bayes Teoremi iki hadisə arasındakı şərtli ehtimal əlaqəsini ifadə edir:

$$P(C_k|X) = rac{P(X|C_k) \cdot P(C_k)}{P(X)}$$

Burada:

- $P(C_k|X)$ Verilən X xüsusiyyətləri əsasında C_k sinfinin ehtimalı (**posterior ehtimal**)
- $P(X|C_k) C_k$ sinfində X xüsusiyyətlərinin müşahidə edilmə ehtimalı (**likelihood**)
- $P(C_k) C_k$ sinfinin ümumi ehtimalı (**prior ehtimal**)
- P(X) X xüsusiyyətlərinin ümumi ehtimalı (**evidence**)

Naive Bayes-də P(X) bütün siniflər üçün eynidir, ona görə də sinif seçimi zamanı müqayisədə nəzərə alınmır.

4.2. Maksimum A Posteriori (MAP) Təxmini 6

Naive Bayes klassifikatoru sinfi seçərkən MAP prinsipi tətbiq edir:

$$\hat{C} = rg \max_{C_k} P(C_k|X)$$

Bayes Teoremini yerinə qoyduqda:

$$\hat{C} = rg \max_{C_k} rac{P(X|C_k) \cdot P(C_k)}{P(X)}$$

P(X) bütün siniflər üçün eyni olduğundan:

$$\hat{C} = rg \max_{C_k} P(X|C_k) \cdot P(C_k)$$

4.3. Sinif Prior Ehtimallarının ($P(C_k)$) Hesablanması

Prior ehtimal $P(C_k)$, sinifin **ümumi ehtimalı** olub sadəcə olaraq o sinifə aid nümunələrin sayının ümumi nümunə sayına nisbəti kimi tapılır:

$$P(C_k) = rac{ ext{sinif } C_k$$
-ə aid nümunələrin sayı ümumi nümunə sayı

Misal:

100 nümunənin 40-ı "pozitiv", 60-ı "neqativ"dirsə:

- $P(\text{pozitiv}) = \frac{40}{100} = 0.4$ $P(\text{neqativ}) = \frac{60}{100} = 0.6$

4.4. Şərtli Ehtimalların ($P(x_i | C_k)$) Hesablanması



Naive Bayes fərziyyəsinə görə xüsusiyyətlər şərti olaraq müstəqildir:

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k)$$

Bu ehtimalların hesablanması modelin növündən asılıdır:

- Gaussian NB normal paylanma sıxlıq funksiyası ilə
- Multinomial NB tezliklər əsasında
- Bernoulli NB ikili ehtimallar əsasında

4.5. Klassifikasiya Qərar Qaydası 🔽

Yekun gərar qaydası belədir:

- 1. Hər sinif üçün $P(C_k)$ hesabla (prior ehtimal)
- 2. Hər xüsusiyyət üçün $P(x_i|C_k)$ tap
- 3. Bütün xüsusiyyətlərin ehtimallarını vur:

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k)$$

4. MAP qaydası ilə ən yüksək $P(X|C_k) \cdot P(C_k)$ dəyəri olan sinifi seç:

$$\hat{C} = rg \max_{C_k} P(X|C_k) \cdot P(C_k)$$

🥊 **Qısa nəticə:** Naive Bayes-in riyazi əsası sadədir, lakin fərziyyələrinə görə (müstəqillik və paylanma) düzgün tətbiq sahələri seçilməlidir.

5. Preprocessing

Naive Bayes modelləri tətbiq etməzdən əvvəl verilənlərin düzgün hazırlanması çox vacibdir. Bu bölmədə əsas **preprocessing** addımları olacaq.

5.1. Null dəyərlər 💢 🛑

- Əsas problem: Dataset-də bəzi dəyərlər boş ola bilər.
- Həll yolları:
 - 1. Silmək: Null nümunələri dataset-dən çıxarmaq
 - 2. Doldurmaq (Imputation):
 - Ortalama ilə (mean)
 - Median ilə
 - Mode ilə (ən çox təkrarlanan dəyər)

Misal:

Əgər yaş xüsusiyyətində 5 null dəyər varsa, ortalama ilə doldurula bilər:

$$yas_{doldurulmus} = \frac{\sum m\"{o}vcud\ yaslar}{m\"{o}vcud\ n\"{u}munələrin\ sayı}$$

```
In [ ]:
       Əvvəlki DataFrame:
           yaş gəlir
       0 25.0 5000.0
          NaN 6000.0
       2 30.0
                  NaN
       3 22.0 4500.0
       4 NaN 7000.0
       5 28.0
                  NaN
       Null dəyərlər doldurulduqdan sonra:
            yaş gəlir
       0 25.00 5000.0
       1 26.25 6000.0
       2 30.00 5625.0
       3 22.00 4500.0
```

5.2. Kategorik Verilənlərin Kodlaşdırılması 🔤



Naive Bayes yalnız **ədədi verilənlərlə** işləyir. Ona görə kategorik dəyişənləri kodlamaq lazımdır.

1. Label Encoding 🥜

4 26.25 7000.0 5 28.00 5625.0

Hər kateqoriya ədədi bir dəyərə çevrilir:

•	"qırmızı"	\rightarrow	0
_	qirriizi	_	U

- "yaşıl" → 1
- "mavi" → 2

2. One-Hot Encoding

Hər kateqoriya üçün ayrıca sütun yaradılır, dəyər 0 və ya 1 olur.

In []:

Label Encoding:

	rəng	rəng_label
0	qırmızı	1
1	yaşıl	2
2	mavi	0
3	qırmızı	1
4	mavi	0

One-Hot Encoding:

	rəng_mavi	rəng_qırmızı	rəng_yaşıl
0	0.0	1.0	0.0
1	0.0	0.0	1.0
2	1.0	0.0	0.0
3	0.0	1.0	0.0
4	1.0	0.0	0.0

5.3. Ədədi dəyişənlərin normalizasiyası / standarlaşdırılması

Ədədi dəyişənlərin **miqyasları fərqli olduqda** model performansı zəifləyə bilər. İki əsas üsul var:

1. Min-Max Normalization:

$$x' = rac{x - x_{
m min}}{x_{
m max} - x_{
m min}}$$

- Dəyərlər 0–1 aralığına çevrilir
- 2. Standardization (Z-score):

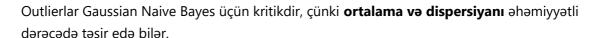
$$x' = rac{x - \mu}{\sigma}$$

• Dəyərlər 0 ortalama və 1 dispersiya ilə normalizasiya olunur

In []:

```
Min-Max Normalization:
   boy çəki
0 0.00 0.00
1 0.25 0.25
2 0.50 0.50
3 0.75 0.75
4 1.00 1.00
Standardization (Z-score):
       boy
              çəki
0 -1.414214 -1.414214
1 -0.707107 -0.707107
2 0.000000 0.000000
3 0.707107 0.707107
4 1.414214 1.414214
```

5.4. Outlier'ların Təsiri 🔔



- Həll yolları:
- Qırmızı xətt (IQR) ilə çıxarmaq
- Logaritmik çevirmə
- Winsorization (outlierləri sərhədlərə yaxınlaşdırmaq)

5.5. Özəllik Seçimi 🔍

Modelin performansını artırmaq üçün ən önəmli xüsusiyyətləri seçmək lazımdır.

- Metodlar:
- 1. Korelasiya analizi
- 2. Chi-square testi (xüsusilə kategorik dəyişənlər üçün)
- 3. Recursive Feature Elimination (RFE)
- 4. Mutual Information

Qeyd: Naive Bayes müstəqillik fərziyyəsinə əsaslandığı üçün çox yüksək korelyasiya olan xüsusiyyətlər performansı azalda bilər.

```
In [ ]:
        Ən yaxşı 2 xüsusiyyət seçildi (X_new):
         [[5 2]
         [4 2]
          [3 3]
          [2 3]
          [1 4]]
```

6. Model Training və Qiymətləndirmə 📊



6.1. GaussianNB, MultinomialNB, BernoulliNB İstifadəsi

- GaussianNB
 - Ədədi (real ədədlər) verilənlər üçün istifadə olunur
 - Normal paylanma fərziyyəsinə əsaslanır
- MultinomialNB 🔢
 - Sayma əsaslı verilənlər üçün optimallaşdırılıb
 - Mətn təsnifatı və söz tezliyi məlumatlarında effektivdir
- BernoulliNB 0 1
 - Binary (var/yox) xüsusiyyətlər üçün istifadə olunur
 - Mətn təsnifatı üçün sözün olub-olmaması əsasında işləyir

Qeyd: Model seçimi dataset-in xüsusiyyət tipinə görə edilir.

In []:	

GaussianNB	Accuracy: 0.9415204678362573				
	preci	.sion	recall	f1-score	support
	0	0.03	0.00	0.00	63
	0	0.93	0.90	0.92	63
	1	0.95	0.96	0.95	108
accurac	:y			0.94	171
macro av	′g	0.94	0.93	0.94	171
weighted av	/g	0.94	0.94	0.94	171

6.2. Train-Test Split və Cross-Validation 🕒

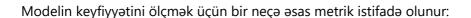
- Train-Test Split
 - Dataset-i **train və test hissələrinə** bölmək vacibdir
 - Adətən 70-80% train, 20-30% test kimi bölünür
 - Train zamanı model parametrləri öyrədilir, test ilə qiymətləndirilir
- Cross-Validation (CV)
 - Dataset-in bir neçə hissəyə bölünərək hər hissənin test üçün istifadə edilməsi
 - K-Fold CV: Dataset K hissəyə bölünür, hər fold bir dəfə test, qalan fold-lar təlim üçün istifadə olunur
 - Daha sabit və ümumiləşdirilmiş performans ölçüsü verir

In []:

GaussianNB 5-Fold CV Accuracy: [0.92105263 0.92105263 0.94736842 0.94736842 0.95575

Average CV Accuracy: 0.9385188635305075

6.3. Performans Metrikləri



1. Accuracy (Dəqiqlik)

$$Accuracy = \frac{D \ddot{u}zg\ddot{u}n \ təxmin \ sayı}{\ddot{u}mumi \ n\ddot{u}munə \ sayı}$$

2. Precision (Dəqiqlik / Məxsusluq) 6

$$Precision = \frac{TP}{TP + FP}$$

- TP: doğru pozitiv
- FP: yalnış pozitiv
- 3. Recall (Həssaslıq)

$$Recall = \frac{TP}{TP + FN}$$

- FN: yalnış negativ
- 4. **F1-Score** 4

$$F1 = 2 \cdot rac{ ext{Precision} \cdot ext{Recall}}{ ext{Precision} + ext{Recall}}$$

Precision və Recall arasında balans yaratmaq üçün istifadə olunur.

In []:

Confusion Matrix:

6.4. Confusion Matrix Analizi 🧩

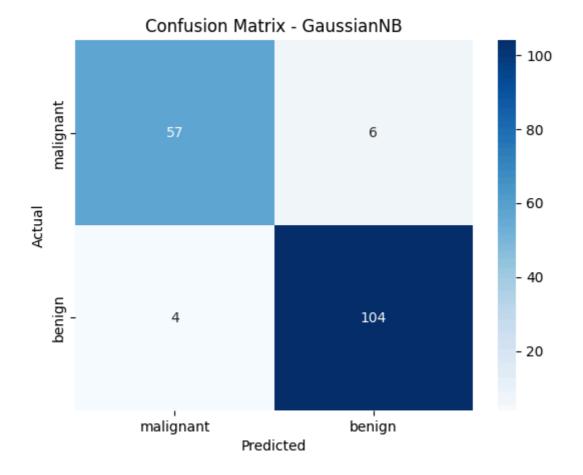


• Confusion Matrix modelin proqnozlarını və əsl sinifləri matris şəklində göstərir:

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

- Bu matris əsasında:
 - Hansı siniflərdə səhvlərin çox olduğu aydın olur
 - Modelin zəif və güclü tərəfləri analiz edilir

n []:	
--------	--



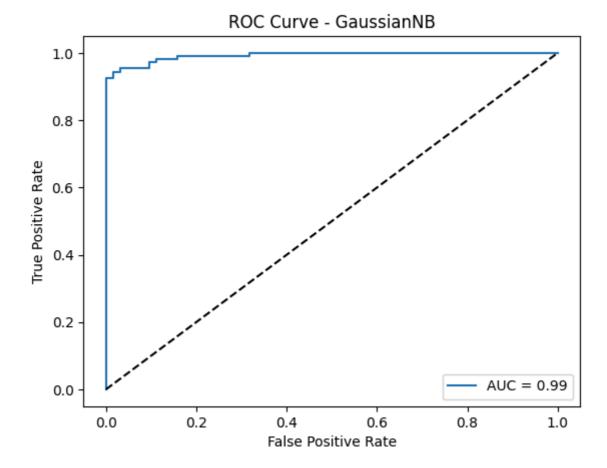
6.5. ROC Əyrisi və AUC Dəyəri 📈

- ROC (Receiver Operating Characteristic) əyrisi:
 - True Positive Rate (TPR) və False Positive Rate (FPR) arasındakı əlaqəni göstərir
 - Yüksək TPR və aşağı FPR, modelin yaxşı performans göstərdiyini göstərir
- AUC (Area Under Curve) dəyəri:
 - ROC əyrisi altındakı sahə
 - 0.5 → təsadüfi performans
 - 1.0 → mükəmməl performans
 - Daha böyük AUC → daha yaxşı model performansı

Qısa nəticə:

Naive Bayes modellərini qiymətləndirmək üçün həm **train-test bölünməsi**, həm **cross-validation**, həm də **accuracy, precision, recall, F1-score, confusion matrix, ROC-AUC** kimi metriklərdən istifadə etmək lazımdır.

In []:	
---------	--



7. Naive Bayes'in Tətbiq Sahələri 🌐

Naive Bayes modelləri sürətli və sadə olmasına görə müxtəlif sahələrdə geniş istifadə olunur.

7.1. Mətn Təsnifatı

- Spam Detection:
 - E-poçt mesajlarının spam və ya normal olduğunu təsnif etmək üçün istifadə olunur
 - Hər bir sözün tekrarlanma ehtimalı əsasında mesajın ehtimalı hesablanır
- Sentiment Analysis:
 - Mətnlərin müsbət, mənfi və ya neytral hisslərini təsnif edir
 - Sosial media postları, məhsul rəyləri və s. üzərində tətbiq olunur
- Nümunə:
 - Mətn: "Bu məhsul çox yaxşıdır!"
 - Naive Bayes ehtimalı hesablayır və nəticəni müsbət kimi təsnif edir

7.2. Sənəd Kateqoriyalaşdırılması (Document Categorization)

- Mətn və sənədləri mövzu və ya kateqoriya üzrə təsnif etmək üçün istifadə olunur
- Hər sənədin içindəki sözlərin yayılma ehtimalına əsaslanır

- Misal:
 - Xəbər məqaləsi: "İdman turniri və qaliblər"
 - Model ehtimal hesablayır və sənədi İdman kategoriyasına yerləşdirir



7.3. Tibbi Diagnostik Sistemlər (Medical Diagnosis Systems)

- Xəstəlikləri simptomlara əsaslanaraq proqnozlaşdırmaq üçün istifadə olunur
- Gaussian Naive Bayes xüsusilə sürekli tibbi ölçümlər üçün uyğundur
- Misal:
 - Simptomlar: yüksək temperatur, baş ağrısı, öskürək
 - Model xəstəliyi (məsələn qrip) ehtimalını hesablayır
- Üstünlük:
 - Kiçik tibbi dataset-lərdə belə sürətli və effektiv nəticə verir

7.4. Real Zamanlı Prognoz Sistemləri (Real-Time Prediction Systems) 🧑 🔸

- Naive Bayes çox sürətli olduğu üçün real zamanlı tətbiqlərdə geniş istifadə olunur
- Ticarət, IoT və sensor məlumatlarında tez cavab vermək vacibdir.
- Misal:
 - Hava sensorları: temperatur və rütubət məlumatlarından yağış ehtimalını prognozlaşdırmag
 - Maliyyə tətbiqləri: valyuta qiymətlərinin trendini real zamanda təxmin etmək

7.5. Fırıldaqçılıq Tapma (Fraud Detection) 💳 👗





- Kredit kartı və bank əməliyyatlarında anomal davranışları aşkarlamaq üçün istifadə olunur
- Hər əməliyyatın normal və ya şübhəli olma ehtimalı hesablanır
- Misal:
 - Birdən-birə böyük məbləğdə alış-veriş
 - Naive Bayes bu əməliyyatın ehtimalını şübhəli kimi müəyyən edir

Qısa nəticə:

Naive Bayes modelləri sürət, sadəlik və statistik etibarlılığı səbəbindən mətn analizi, tibbi diagnostika, real zamanlı prognoz və fırıldagçılıq aşkarlama kimi sahələrdə çox geniş tətbiq olunur.

8. Optimizasiya 🗱 🚀

Naive Bayes modeli sadə olsa da, real dünyada performansı artırmaq üçün bir neçə optimizasiya texnikası mövcuddur.

8.1. Laplace (Additive) Smoothing +

• Problem:

Multinomial və Bernoulli Naive Bayes-də heç vaxt görülməmiş xüsusiyyətlər sıfır ehtimala səbəb olur:

$$P(x_i|C_k)=0$$

Bu isə bütün məhsul ehtimalını sıfırlayır 🗙

• Hall: Laplace smoothing

• Hər xüsusiyyətə kiçik bir dəyər əlavə edilir (adətən $\alpha=1$):

$$P(x_i|C_k) = rac{n_{c_i} + lpha}{N_c + lpha \cdot m}$$

Burada:

- n_{c_i} sinifdə xüsusiyyətin sayı
- lacksquare N_c sinifdəki bütün xüsusiyyətlərin sayı
- m unikal xüsusiyyətlərin sayı
- α smoothing parametri
- Nəticə: Sıfır ehtimallar aradan galxır və model daha stabil olur 🔽

8.2. Class Imbalance 🔱

- Problem: Siniflər balanssız olduqda model majority class-ı üstün tutar
- Həll yolları:
 - 1. Sinif çəkisi (class weight) əlavə etmək
 - 2. Oversampling / Undersampling
 - SMOTE və ya Random Oversampling ilə minority class artırılır
 - Random Undersampling ilə majority class azaldılır
 - 3. Complement Naive Bayes balanssız verilənlər üçün optimallaşdırılmış variant
- Nəticə: Model bütün sinifləri daha ədalətli şəkildə təsnif edir of

8.3.Independence Assumption Violation



- Naive Bayes fərziyyəsi: Xüsusiyyətlər müstəqildir
- Problem: Reallıqda xüsusiyyətlər bir-birinə bağlı ola bilər
 - Məsələn, tibbi dataset-də "yaş" və "təzyiq" yüksək korelyasiyaya malik ola bilər
- Təsiri:
 - Asılılıq artırsa, modelin ehtimalları səhv ola bilər

- Həll yolları:
 - 1. Özəllik seçimi (feature selection) ilə korelyasiya azaldılır
 - 2. PCA və ya Dimensionality Reduction ilə bağımlılıq azaldılır
 - 3. Hybrid modellər (NB + Tree / Logistic Regression) ilə daha doğru nəticə əldə edilir

8.4. Hibrid Naive Bayes Modelləri 🔀

- Hibrid model: Naive Bayes-i digər modellərlə birləşdirir
 - Məsələn: GaussianNB + Logistic Regression, MultinomialNB + Random Forest
- Faydalı halları:
 - Xüsusiyyətlər müstəqillik fərziyyəsini pozduqda
 - Kompleks dataset-lərdə daha yaxşı performans lazım olduqda
- Nəticə: Hibrid model daha yüksək doğruluq və stabil nəticə verir

8.5. Naive Bayes ila Feature Engineering Strategiyalari 🛠

- Feature engineering Naive Bayes üçün çox vacibdir
- Taktikalar:
 - 1. Tokenization və Text Preprocessing (stopword çıxarılması, stemming, lemmatization)
 - 2. **TF-IDF və ya word frequency features** istifadə etmək
 - 3. **Binary features (0/1)** yaratmaq (Bernoulli NB üçün)
 - 4. **Discretization** sürekli xüsusiyyətləri qruplaşdırmaq (Gaussian NB üçün)
 - 5. Interaction features xüsusiyyətlərin kombinasiyalarını yaratmaq, amma diqqətli olmaq lazımdır, müstəqillik fərziyyəsini pozmamaq
- Nəticə: Yaxşı hazırlanmış xüsusiyyətlər modeli daha güclü və dəqiq edir

9. Tətbiq 🪀

9.1. Python ilə GaussianNB Tətbiqi 过 📊





- Python və scikit-learn kitabxanası GaussianNB tətbiqi üçün çox uyğundur
- Addımlar:
 - 1. Dataset-i yükləmək və preprocessing aparmaq
 - 2. Xüsusiyyətləri və etiketi ayırmaq
 - 3. Modeli yaratmaq və train etmək
 - 4. Train nəticələrini qiymətləndirmək

9.2. Model Nəticələrinin Şərh Edilməsi 📈 🧐





Model nəticələrini giymətləndirmək üçün aşağıdakı məqamlar nəzərə alınmalıdır:

- Accuracy, Precision, Recall, F1-score
- Confusion Matrix analizi
- Hər sinif üçün ehtimalların paylanması
- Nümunə:
 - Model "pozitiv" ehtimalı yüksək olan nümunəni düzgün təsnif edirsə, bu nəticə etibarlı sayılır 🔽
 - Səhvlər varsa, hansı xüsusiyyətlər problem yaratdığını araşdırmag lazımdır

9.3. Hiperparametr Requlyasiyası 🗱

- GaussianNB üçün əsas hiperparametr: var_smoothing
 - Kiçik dəyərlər dispersiyanı stabilləşdirir
 - Daha böyük dəyərlər isə modelin həddindən artıq sadələşməsinə səbəb ola bilər
- MultinomialNB və BernoulliNB-də alpha (Laplace smoothing) əhəmiyyətlidir
- Hiperparametr tuning metodları:
 - Grid Search
 - Random Search
 - Cross-validation
- Məqsəd: Modelin həm train, həm də test dəqiqliyini artırmaq 🎯

9.4. Model Deployment 📗

- - 1. Model fayl şəklində saxlanır (pickle, joblib və s.)
 - 2. API və ya servis vasitəsilə tətbiqlərə integrasiya olunur

Train edilmiş model real dünyada istifadə üçün hazırlıq mərhələsi:

- 3. Giriş verilənləri preprocessingdən keçirilir və model proqnoz verir
- Qeyd: Performans və sabitlik təmin olunmalıdır, xüsusilə real zamanlı sistemlər üçün 🔸

9.5. Modelin Vaxtaşırı Yenilənməsi 🔀 🔄

- Real dünya məlumatları zamanla dəyişə bilər (data drift)
- Yeniləmə üsulları:
 - Periodik olaraq modeli yenidən train etmək
 - Yeni nümunələri əlavə edərək incremental learning
 - Hiperparametrləri yenidən optimallaşdırmaq
- Məqsəd: Model uzun müddət etibarlı və dəqiq qalmalıdır 🔽