NAÏVE BAYES

MACHINE LEARNING – CLASSIFICATION ALGORITHM

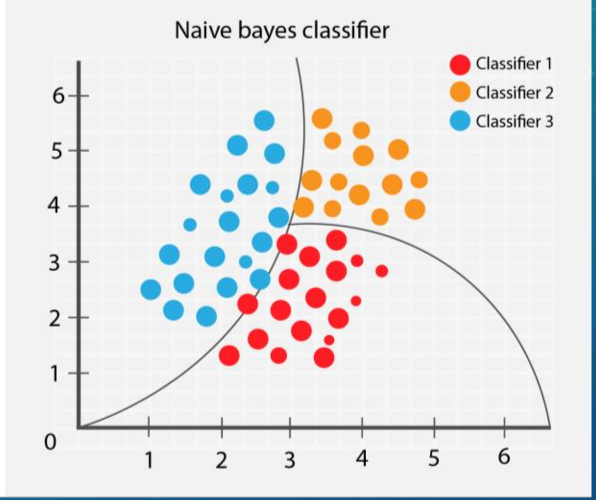
NAÏVE BAYES

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

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

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$



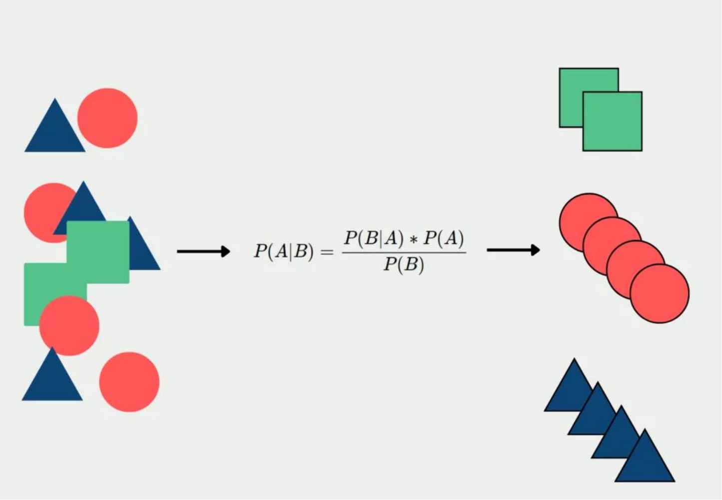
Advantages:

- naive Bayes classifiers have worked quite **well in many real-world situations**, famously document classification and spam filtering.
- They require **a small amount of training data** to estimate the necessary parameters
- Naive Bayes learners and classifiers can be **extremely fast** compared to more sophisticated methods.

Disadvantages:

- Naive Bayes assumes independence among features, which is often unrealistic. Its performance may degrade with highly correlated features and imbalanced datasets.

NAÏVE BAYES WORK FLOW



1) Find the Prior Probability
2) Find the conditional probability and Build probability table
3) Predict the output

GAUSSIAN NAIVE BAYES

`GaussianNB` implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

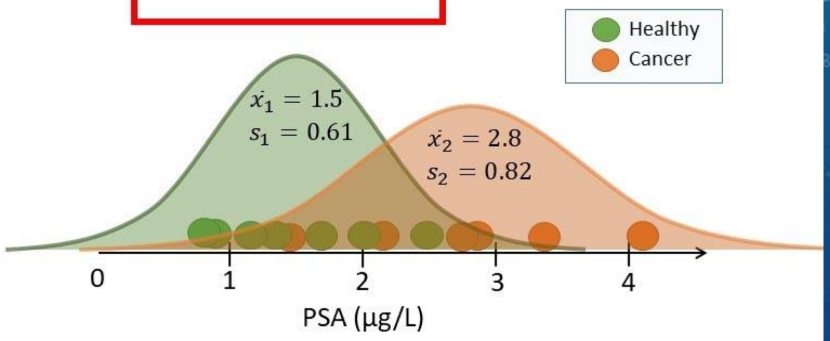
$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The parameters σ_y and μ_y are estimated using maximum likelihood.

Gaussian Naive Bayes

Status	PSA
Cancer	4.1
Cancer	3.4
Cancer	2.9
Cancer	2.8
Cancer	2.7
Cancer	2.1
Cancer	1.6
Healthy	2.5
Healthy	2.0
Healthy	1.7
Healthy	1.4
Healthy	1.2
Healthy	0.9
Healthy	0.8

$$\boxed{f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}}$$



Code: https://github.com/krthiksha/Machine-Learning-Classification_module/blob/main/4.NB_classification.ipynb

MULTINOMIAL NAIVE BAYES

- The multinomial Naive Bayes classifier is suitable for classification with **discrete features** (e.g., word counts for **text classification**).
- The multinomial distribution normally requires integer feature counts.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Message	offer	free	win	hello	meeting	Class
"Free offer just for you"	1	1	0	0	0	Spam
"Win a free lottery now"	0	1	1	0	0	Spam
"Hello, are we meeting today?"	0	0	0	1	1	Ham
"Let's schedule the meeting"	0	0	0	0	1	Ham
"Special offer! Win prizes"	1	0	1	0	0	Spam

Code: https://github.com/krthiksha/Machine-Learning-Classification_module/blob/main/4.NB_classification.ipynb

COMPLEMENT NAIVE BAYES

- The Complement Naive Bayes classifier (CNB) was **designed to correct the "severe assumptions" made by the standard Multinomial Naive Bayes classifier** (MNB)
- It is particularly **suited for imbalanced data sets**.
- CNB uses statistics from the *complement* of each class **to compute the model's weights**.
- CNB is more stable than MNB

$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}}$$

$$w_{ci} = \log \hat{\theta}_{ci}$$

$$w_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|}$$

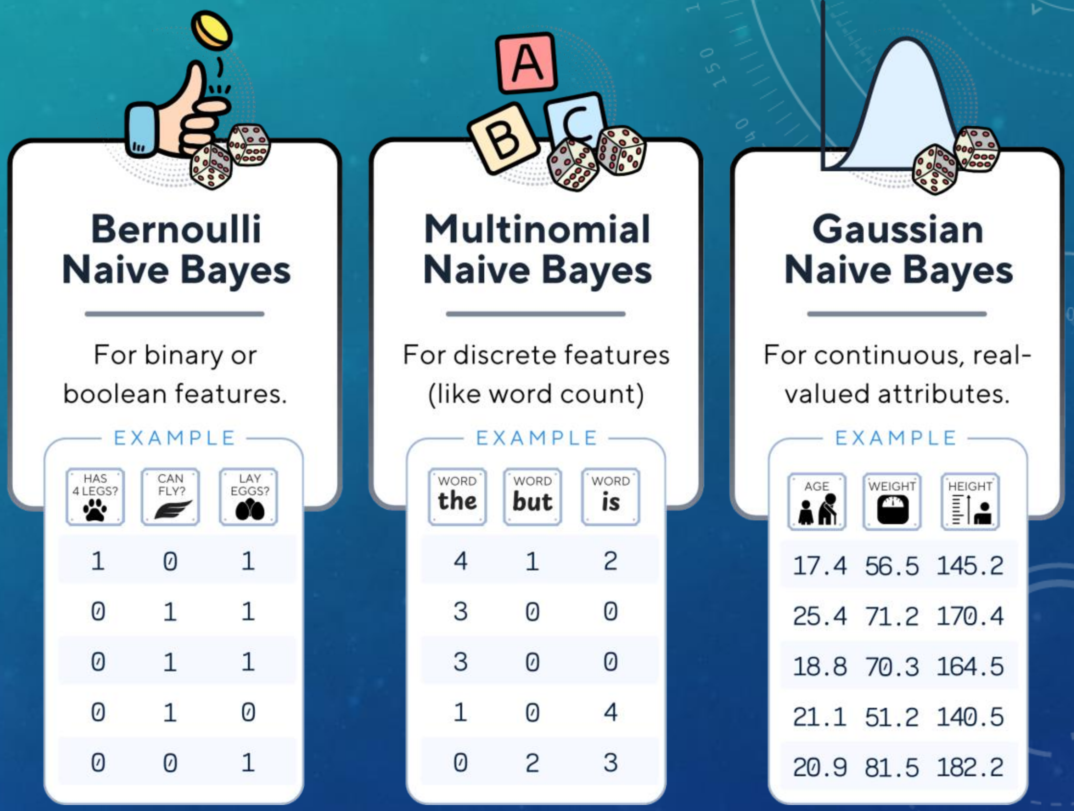
$$\hat{c} = \arg\min_c \sum_i t_i w_{ci}$$

Scenario	CNB	MNB
Balanced dataset	✅	✅
Imbalanced dataset	✅✅	❌
Short text with little context	✅	✅
Multi-class classification	✅	✅
Need for fast, lightweight model	✅	✅

Code: https://github.com/krthiksha/Machine-Learning-Classification_module/blob/main/4.NB_classification.ipynb

BERNOULLI NAIVE BAYES

- **multivariate Bernoulli models**.
- **<u>BernoulliNB is designed for binary/boolean features.</u>**
- there may be **multiple features** but **each one is assumed to be a binary-valued** (Bernoulli, boolean) variable.
- In case of text classification, **Bernoulli calculate word occurrence vectors** (rather than word count vectors)



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

Code: https://github.com/krthiksha/Machine-Learning-Classification_module/blob/main/4.NB_classification.ipynb

CATEGORICAL NAIVE BAYES CLASSIFIER

- The categorical Naive Bayes classifier is suitable for **classification with discrete features that are categorically distributed.**
- The categories of each feature are drawn from a categorical distribution.

$$P(x_i = t \mid y = c \,;\, \alpha) = \frac{N_{tic} + \alpha}{N_c + \alpha n_i},$$

Sample	Age Group	Gender	Class
1	Young	Male	Not Buy
2	Young	Female	Buy
3	Middle	Male	Not Buy
4	Young	Male	Not Buy
5	Middle	Female	Buy
6	Senior	Female	Not Buy
7	Senior	Male	Buy
8	Middle	Male	Buy
9	Young	Female	Not Buy
10	Middle	Female	Buy

Code: https://github.com/krthiksha/Machine-Learning-Classification_module/blob/main/4.NB_classification.ipynb

THANK YOU !!