The Naive Bayesian Classifier

Numerical Attributes

Presented by:
Dr Javed Iqbal

Material belongs to Prof Saed Sayad
http://www.saedsayad.com

You're trying to figure out whether an email is **spam** or **not spam**. You notice certain words like:

- "win", "free", "cash" → often appear in spam
- "meeting", "project", "schedule" → usually not spam

So, when you see a new email, you **look at the words**, and ask:

"Given these words, what is the probability this email is spam?"

That's exactly what **Naive Bayes** does using **Bayes' Theorem**!

"What is the probability of a class (e.g., Spam), given the observed features (e.g., words)?"

It calculates:

1. **Prior Probability** of each class (e.g., how often spam occurs)
2. **Likelihood** of features (e.g., how often each word appears in spam emails)
3. Combines them to pick the **most probable class**

💡 Why "Naive"?

Because it **naively assumes** all features (words) are **independent**. So it treats:

- P("win", "cash", "now" | spam)
  as
- P("win" | spam) × P("cash" | spam) × P("now" | spam)

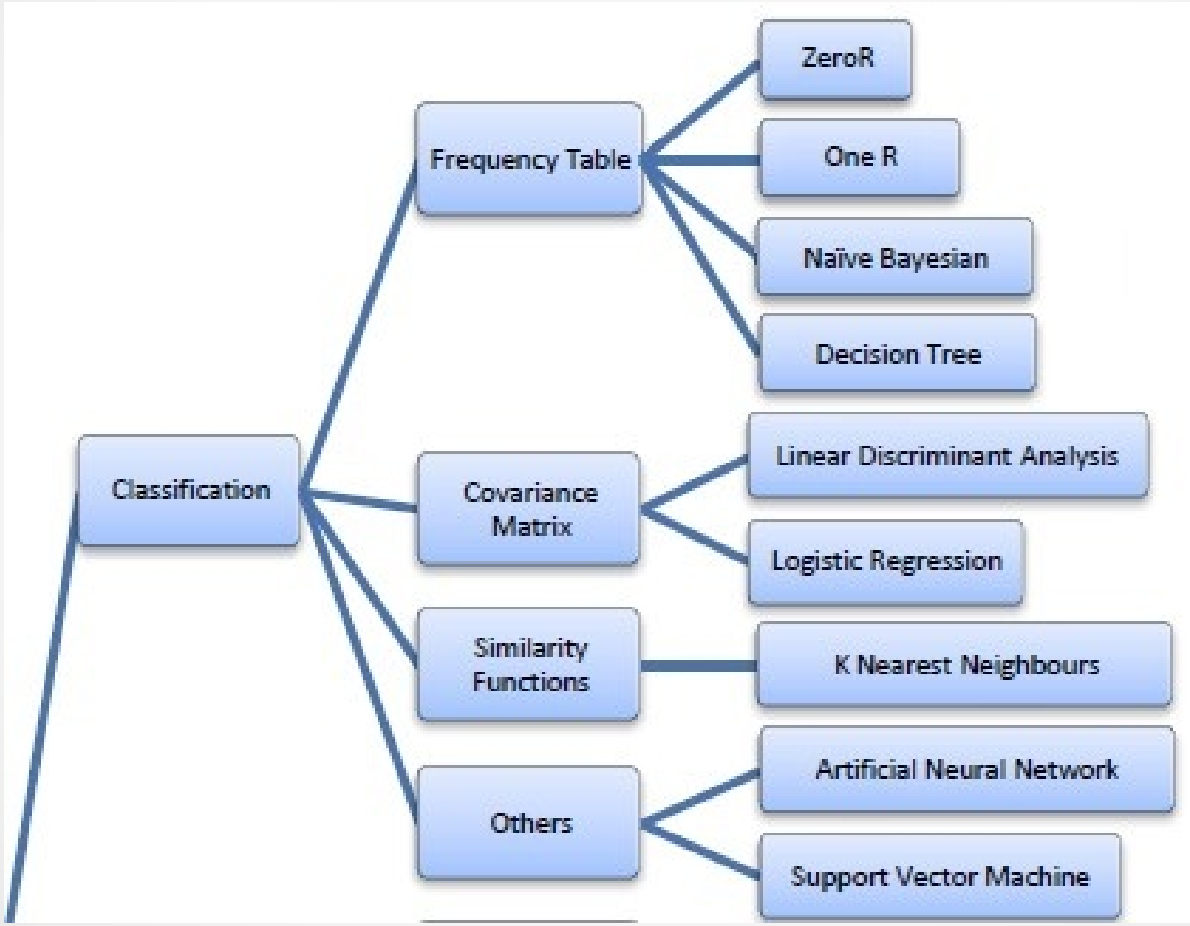
Even though words in real sentences depend on each other, this simplification makes the model:

- Very fast
- Surprisingly accurate in many tasks (especially text classification)

🎁 In Practice:

When classifying something, Naive Bayes just:

1. Checks which class each feature favors
2. Multiplies all the evidence
3. Chooses the class with the **highest overall probability**

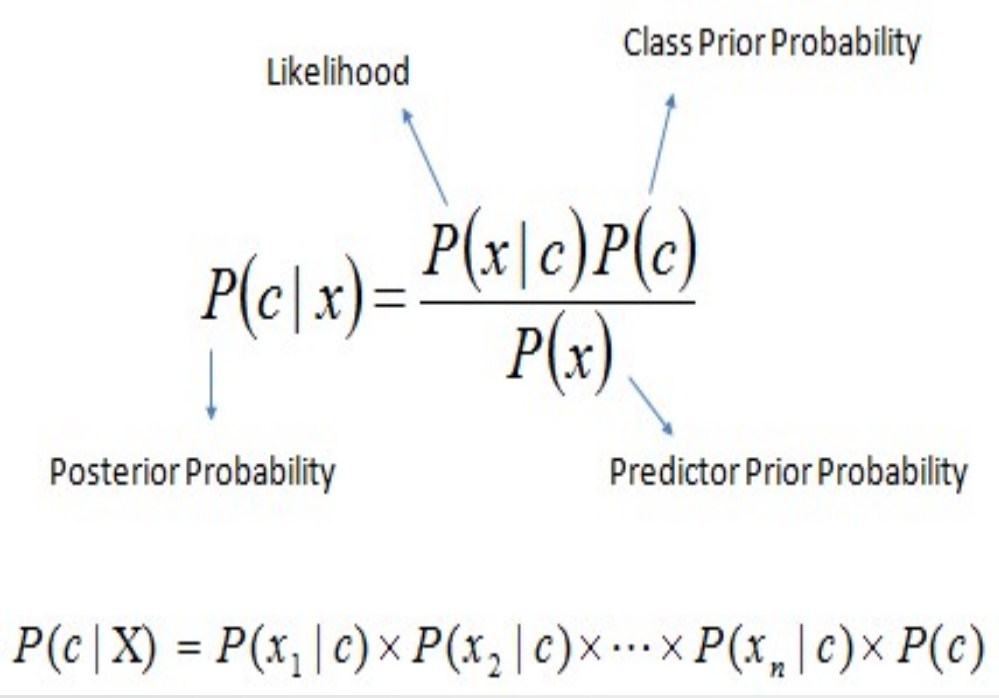

Classification
Frequency Table
ZeroR
One R
Naïve Bayesian
Decision Tree
Covariance Matrix
Linear Discriminant Analysis
Logistic Regression
Similarity Functions
K Nearest Neighbours
Others
Artificial Neural Network
Support Vector Machine

The Naive Bayesian Classifier

- The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors
- A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets
- Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods

How it works

- Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c)
- NB classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence



- P(c|x) is the posterior probability of class (target) given predictor (attribute)
- P(c) is the prior probability of class
- P(x|c) is the likelihood which is the probability of predictor given class
- P(x) is the prior probability of predictor

Example

- The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target
- Then, transforming the freq. tables to likelihood tables and finally using the Naive Bayesian equation to calculate the posterior probability for each class
- The class with the highest posterior probability is the outcome of prediction

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$P(x) = P(Sunny) = 5/14 = 0.36$

$P(c) = P(Yes) = 9/14 = 0.64$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$

Example

Which one is the best predictor ?

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

P(Yes) = 9 / 14
P(No) = 5 / 14

Example

Frequency Tables

		Play Golf	
		Yes	No
Outlook	Sunny	3 3/9	2 2/5
	Overcast	4 4/9	0 0/5
	Rainy	2 2/9	3 3/5

		Play Golf	
		Yes	No
Temp.	Hot	2 2/9	2 2/5
	Mild	4 4/9	2 2/5
	Cool	3 3/9	1 1/5

		Play Golf	
		Yes	No
Humidity	High	3 3/9	4 4/9
	Normal	6 6/9	1 1/5

		Play Golf	
		Yes	No
Windy	False	6 6/9	2 2/5
	True	3 3/9	3 3/5

Example

- Let's assume we have a day with:

  Outlook = Rainy

  Temp = Mild

  Humidity = Normal

  Windy = True

  Likelihood of Yes = P(Outlook=Rainy|Yes)*P(Temp=Mild|Yes)*P(Humidity=Normal|Yes)*P(Windy=True|Yes)*P(Yes) =

  2/9 * 4/9 * 6/9 * 3/9 * 9/14 = 0.014109347

  Likelihood of No = P(Outlook=Rainy|No)* P(Temp=Mild|No)*P(Humidity=Normal|No)*P(Windy=True|No)*P(No) =

  3/5 * 2/5 * 1/5 * 3/5 * 5/14 = 0.010285714

- Now we normalize:

  P(Yes) = 0.014109347/(0.014109347+0.010285714) = 0.578368999

  P(No) = 0.010285714/(0.014109347+0.010285714) = 0.421631001

Example

- Let's assume we have a day with:

  Outlook = Rainy

  Temp = Mild

  Humidity = Normal

  Windy = True

  Likelihood of Yes = P(Outlook=Rainy|Yes)*P(Temp=Mild|Yes)*P(Humidity=Normal|Yes)*P(Windy=True|Yes)*P(Yes) =

  2/9 * 4/9 * 6/9 * 3/9 * 9/14 = 0.014109347

  Likelihood of No = P(Outlook=Rainy|No)* P(Temp=Mild|No)*P(Humidity=Normal|No)*P(Windy=True|No)*P(No) =

  3/5 * 2/5 * 1/5 * 3/5 * 5/14 = 0.010285714

- Now we normalize:

  P(Yes) = 0.014109347/(0.014109347+0.010285714) = 0.578368999

  P( ) = 0.010285714/(0.014109347+0.010285714) = 0.421631001

The zero-frequency problem

- When an attribute value (Outlook=Overcast) doesn’t occur with every class value (Play Golf=no)
- Add 1 to all the counts

Numerical Predictors

- Numerical variables need to be transformed to their categorical counterparts (binning) before constructing their frequency tables
- The other option we have is using the distribution of the numerical variable to have a good guess of the frequency
- For example, one common practice is to assume normal distributions for numerical variables

Normal Distribution

- The probability density function for the normal distribution is defined by two parameters (mean and standard deviation)

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Mean

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \mu\right)^2\right]^{0.5}$$

Standard deviation

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

Normal distribution

Example of Numerical Predictors

		Humidity	*Mean*	*StDev*
Play Golf	yes	86 96 80 65 70 80 70 90 75	79.1	10.2
	no	85 90 70 95 91	86.2	9.7

$$P(\text{humidity} = 74 \mid \text{play} = \text{yes}) = \frac{1}{\sqrt{2\pi}(10.2)} e^{-\frac{(74-79.1)^2}{2(10.2)^2}} = 0.0344$$

$$P(\text{humidity} = 74 \mid \text{play} = \text{no}) = \frac{1}{\sqrt{2\pi}(9.7)} e^{-\frac{(74-86.2)^2}{2(9.7)^2}} = 0.0187$$

Predictors Contribution

- Kononenko's information gain as a sum of information contributed by each attribute can offer an explanation on how values of the predictors influence the class probability

$$log_2 P(c|x) - log_2 P(c)$$

Nomograms

- The contribution of predictors can also be visualized by plotting nomograms
- Nomogram plots log odds ratios for each value of each predictor
- Lengths of the lines correspond to spans of odds ratios, suggesting importance of the related predictor
- It also shows impacts of individual values of the predictor