Naive - Bayes

→ 70% probability of Rain

→ Proof so hand for similar condition of a day in past it has rained 70 days out of 100

→ Naive bayes uses the training data to calculate an observed probability of each outcome based on evedence given by features

Ex Text classification, Diagnoise symptoms.

Information from many features may be considered to increase the overall performance of model

	CC	Apply	Cash back	Due date
Spam ← m_1 →	Spam	Spam	Spam	Spam
Bill ← m_2 →	Spam	×	Spam	×
Spam ← m_3 →	Spam	Spam	Spam	Spam

Includes → Weak variables → Impact ↑ if no of Variables ↑

Estimated likelihood of an event is based on the evedence in Hand across Multiple trials of the event occourance

Event	Trial
Head	Coin - toss
Spam	Incoming email
Win lottery	Ticket purchase

→ Probability = $\frac{\text{Success trials}}{\text{Total trials}}$

→ $P(A) = \{0 \text{ to } 1\}$

→ Mutually exclusive (E) Collectively Exhaustive.

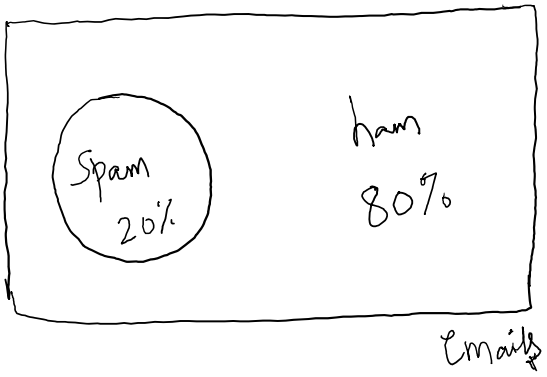
→ Cannot occour at a same time (E) they are the only possible

Outcomes.

$P(\text{Spam}) = 0.2$

$P(\text{ham}) = 0.8$

$P(\text{Spam})' = 0.8$

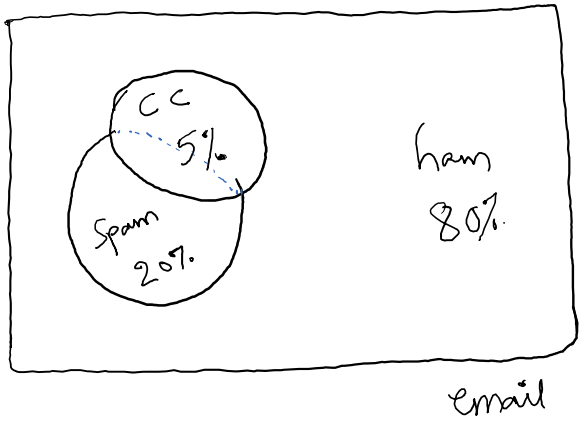


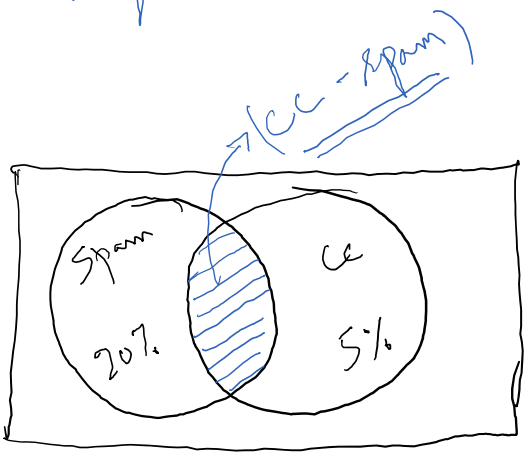
Joint probability

Not all "cc" mails are spam

(&)

Not all spam mails will have "cc"





$P(\text{spam} \cap cc) \rightarrow$ Independent event

$\rightarrow$ Dependent event

Independent event

$$P(A \cap B) = \overset{S}{P(A)} * \overset{CC}{P(B)}$$

$$= 0.2 * 0.05 = \underline{\underline{0.01}}$$

Dependent event $\rightarrow$ Basis of Predictive modeling

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

$$\boxed{P(\text{Spam}|cc) = \frac{P(cc \cap \text{spam})}{P(cc)} = \frac{P(cc|\text{spam}) \cdot P(\text{spam})}{P(cc)}}$$

Freq table to find Components of bayes Theorem

Likelihood	CC: Yes	CC: No	
Spam	4/20	16/20	20
ham	1/80	79/80	80
Total	5/100	95/100	100

24 → 24

$$P(spam|cc) = \frac{P(cc|spam) * P(spam)}{P(cc)}$$

$$= \frac{4/20 \times 20/100}{5/100} = 0.8$$

For Multiple keywords.

$$P(spam | w_1 \cap w_2' \cap w_3' \cap w_4) = \frac{P(w_1 \cap w_2' \cap w_3' \cap w_4 | spam) \cdot P(spam)}{P(w_1 \cap w_2' \cap w_3' \cap w_4)}$$

⇒ Laplace estimation → 20 make it 24

Adding a small no during Calculation to Avoid 0

Steps → Building model.

1 → Data collection

2 → Data Exploration → tm package → (and, but, or) → Removes

→ Split the text in to words → "Tokenization"

↳ DTM → Document Text Matrix

→ Visualise → Word cloud (frequency of word occurance)

→ Indicator creation for frequent words

S_3 → Model → m ← naivebayes(train=....., class=....., laplace=0)

$S_4 \rightarrow$ Evaluate $\rightarrow$ Confusion matrix $\rightarrow$ other ways

$S_5 \rightarrow$ Improvement $\rightarrow$ <u>laplace = 1</u> / change & Run

$S_6 \rightarrow$ Summary $\rightarrow$ o/p Accuracy = ?

laplace estimate = ?