Naïve Bayes:

→ Probabilistic ML Algorithm.
→ used for classification.

Use cases:

1. Spam filtering.
2. classifying documents.
   (Tex classification)
3. Sentiment Analysis.

Calculate   Train

Posterior = Prior   0.7

P(spam | dollar)

= P(dollar | spam) × P

Naïve Bayes

It assumes that the features that go into the model is independent to each other.

Based on Bayes theorem

KNN

Adv.

→ Since it is a Prob. model, algo is easy to code.
→ Predictions are real time quick.
→ Scalable.

P( )

Conditional Prob. Revision.

★ Coin Toss / Fair dice Roll

Toss a coin : H T

$P(H) = 0.5$ $P(T) = 0.5$

Roll a dice : 1, 2, 3, 4, 5, 6

$P(1) = 1/6 = 0.16$

Drawing a card from a deck : $P(\text{queen}) = \frac{4}{52}$

$P(Q \mid \text{spade}) = \frac{1}{13}$

$$\boxed{P(A \mid B) = \frac{P(A \cap B)}{P(B)}}$$

Bayes Rule

$P(X|Y)$

$P(Y|X)$?

A B C

$$\boxed{P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)}}$$

$$P(y=k|x_1 \cdots x_n) = \frac{P(x_1|y=k) * P(x_2|y=k) \cdots * P(x_n|y=k) * P(y=k)}{P(x_1) * P(x_2) * P(x_3) \cdots P(x_n)}$$

$P(y = \text{cloudy} | x_1 \cdots x_n) = P(x_1|\text{cloudy}) \; P(x_2|x \; \times P(y=$

y

cloudy

Rainy

overcast

cloudy

Example:

1000 fruits → Banana
orange
others

X_1 = Long	X_2 = Sweet	X_3 = Yellow	Y
0	1	0	Orange
1	0	1	Banana

1000 → 500 (B), 300 (orange), 200 (others)

Type	Long	Not long	Sweet	Not Sweet	yellow	Not yellow	Total
Banana	400	100	350	150	450	50	500
orange	0	300	150	150	300	0	300
others	100	100	150	50	50	150	200
Total	500	500	650	350	800	200	1000

Long	Sweet	Yellow	?

Step 1:

$$P(y = \text{Banana}) = \frac{500}{1000} = 0.5$$

$$P(y = \text{orange}) = \frac{300}{1000} = 0.3$$

$$P(y = \text{others}) = \frac{200}{1000} = 0.2$$

Step 2:

$$P(\text{Long}) = 500/1000 = 0.5$$

$$P(\text{Sweet}) = 650/1000 = 0.65$$

$$P(\text{Yellow}) = 800/1000 = 0.8$$

Step 3:

$$P(\text{long} \mid \text{Banana}) = \frac{400}{500} = 0.8$$

$\dfrac{0.0000001}{[illegible]}$

$$P(\text{sweet} \mid \text{Banana}) = 350/500 = 0.7$$

$$P(\text{yellow} \mid \text{Banana}) = 450/500 = 0.9$$

$$P(\text{Banana} \mid \text{long, sweet \& yellow})$$

$$= \frac{P(\text{long} \mid \text{Banana}) * P(\text{Sweet} \mid \text{Banana}) * P(\text{yellow} \mid \text{Banana}) * P(\text{Banana})}{P(\text{long}) \; P(\text{Sweet}) \; P(\text{yellow})}$$

$$= \frac{0.8 \times 0.7 \times 0.9 \times 0.5}{0.5 \times 0.65 \times 0.8} = \boxed{}$$

$$P(\text{orange} \mid \text{long, sweet \& yellow}) = 0$$

$$P(\text{other fruit} \mid \text{''} \quad) = \frac{0.01875}{0.5 \times 0.65 \times 0.8} =$$

$P(B \mid x_1, x_2, x_3)$ is highest hence the class of fruit for the given row is Banana.

Laplace Correction:

$p(\text{long} \mid \text{orange}) = 0$

& hence overall Prior Prob. become 0.

To avoid this, we increase the count of variable from zero to small value (usually 1) in the numerator. So that overall Prob. doesn't become zero.

$\Downarrow$

Laplace Correction.

Classifiers of NB Algo.

1. Gaussian : It assumes that features follows N.D.
2. Multinomial: useful when your features are discrete.
3. Bernoulli: features are binary in nature then it is useful.

Cons :

→ Require to remove Correlated features because they are voted twice.

→ Zero freq. → Zero Prob.

↳ 1 Laplace Correction.

2 other smoothing Techniques.

EOF>>>