

Naive Bayes

&

Support Vector Machines

[ML - 1]

Naive Bayes Recap

Conditional Probab [Bayes theorem]

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

Therefore

$$P(y=1/x) = \frac{P(x|y=1) * P(y=1)}{P(x)}$$

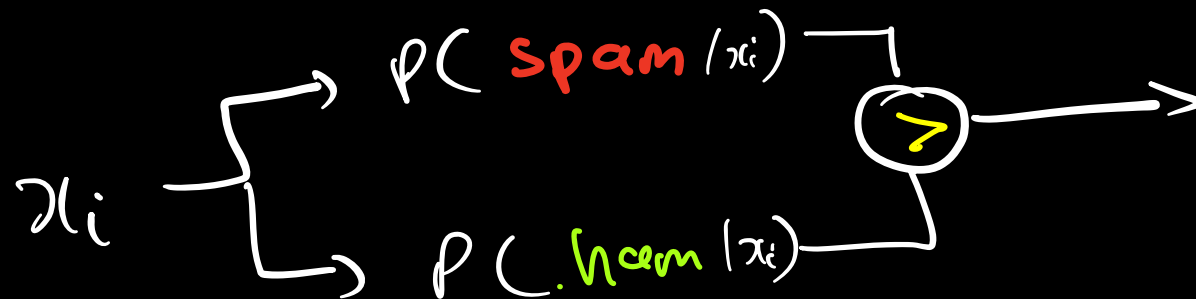
$$P(y=0/x) = \frac{P(x|y=0) * P(y=1)}{P(x)}$$

In Simple terms:

$$P(\text{spam} / \text{words}) = \frac{P(\text{words} / \text{spam}) * P(\text{spam})}{P(\text{words})}$$

$$P(\text{ham} / \text{words}) = \frac{P(\text{words} / \text{ham}) * P(\text{ham})}{P(\text{words})}$$

NB Classifier



Terms

1. $P(\text{words} / \text{span}) = \text{likelihood}$

2. $P(\text{words}) = \text{evidence}$

3. $P(\text{span}) = \text{prior}$

A diagram showing the components of Bayes' theorem. The equation $P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$ is centered. Above the equation, the word 'posterior' has an arrow pointing to $P(A|B)$. Above the numerator, 'likelihood' has an arrow pointing to $P(B|A)$. To the right of the numerator, 'prior' has an arrow pointing to $P(A)$. Below the denominator, 'evidence' has an arrow pointing to $P(B)$.

$$\begin{array}{ccc} \text{posterior} & \text{likelihood} & \\ \downarrow & \downarrow & \\ P(A|B) = \frac{P(B|A) * P(A)}{P(B)} & & \text{prior} \\ & \uparrow & \\ & \text{evidence} & \end{array}$$

1. Evidence:

→ No need to calculate (common den.)

2. Prior:

→ $P(y = i)$

easy → df. classes. value_counts (norm = True)

3. Likelihood

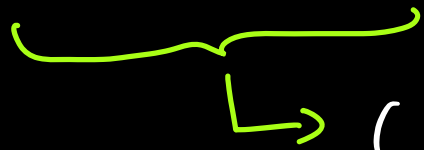
$P(x_i / y_i = j)$

$x_i = [w_1, w_2, \dots, w_n]$

$$\therefore P(w_1, w_2 \dots w_m / y = 1)$$

Naive assumption:

$$P(w_1 / y = 1) \cdot P(w_2 / y = 1) \dots P(w_m / y = 1)$$



```
(
    df.loc[df.classes == 1]["text"].
    contains(w1).sum() /
    (df.classes == 1).sum()
)
```

Hence,

$$P(\text{spam} / \text{words}) = \frac{P(\text{spam}) \times \prod P(w_i / \text{spam})}{K}$$

Log Trick

$$P(w_1 / y=1) \cdot P(w_2 / y=1) \cdot \dots \cdot P(w_{50} / y=1)$$

→ These are lots of mult of values ≤ 1

Eg: $0.5^{50} \sim \underline{\underline{10^{-16}}}$ → Possible compute issues.

hence, we use

$$\text{antilog}(\log(p(w_1/y=1) \cdot p(w_2/y=1) \cdot \dots \cdot (p(w_{50}/y=1)))$$

$$= e^{\log(p_1 \cdot p_2 \cdot p \dots p_{50})}$$

$$= e^{\log(p_1) + \log(p_2) \dots + \log(p_{50})}$$

→ More stable computation

Laplace Smoothing

What if we get a new word in test data?

$$p(w_i / y=1) \cdot \underbrace{p(w_{new} / y=1)}_{?} \dots (p w_{so} / y=1)$$

a) $\rightarrow 0$ [Overall o/p = 0]

b) $\rightarrow 1$ [Overall o/p = unchanged]

if w_{new} is
avl for $y=0$
but not $y=1$

c) \rightarrow Laplace smoothing

$$p(w_i / y=1) = \frac{\# w_i \text{ occurs when } y=1}{\# y=1}$$

$$= \frac{n_i}{n}$$

$$p_{lap(k)}(w_i / y=1) = \frac{n_i + k}{n + CK} \rightarrow \text{smoothing factor}$$

\nwarrow
 # possible values of x_i

Example

H

H

T

No smoothing $k=0$
 $k=1$
 $k=100$

$P(H)$

$2/3$

$3/5$

$\frac{102}{203}$

$P(T)$

$1/3$

$2/5$

$\frac{101}{203}$

66% 33%

60% 40%

51% 49%

1

So if we decide a

suitable k , then the prob is not affected much.

↓
Over smooth

But, when $n_i = 0$

$$= \frac{n_i + k}{n_i + ck}$$

$$= \frac{0 + k}{0 + ck} \neq 0$$

↓
small number

$$\approx 0$$