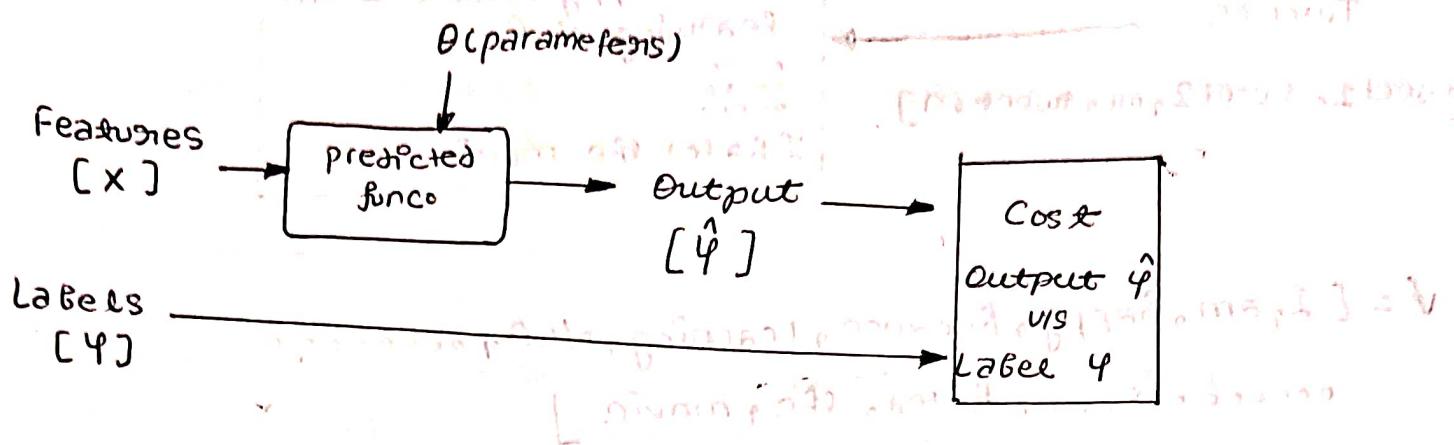


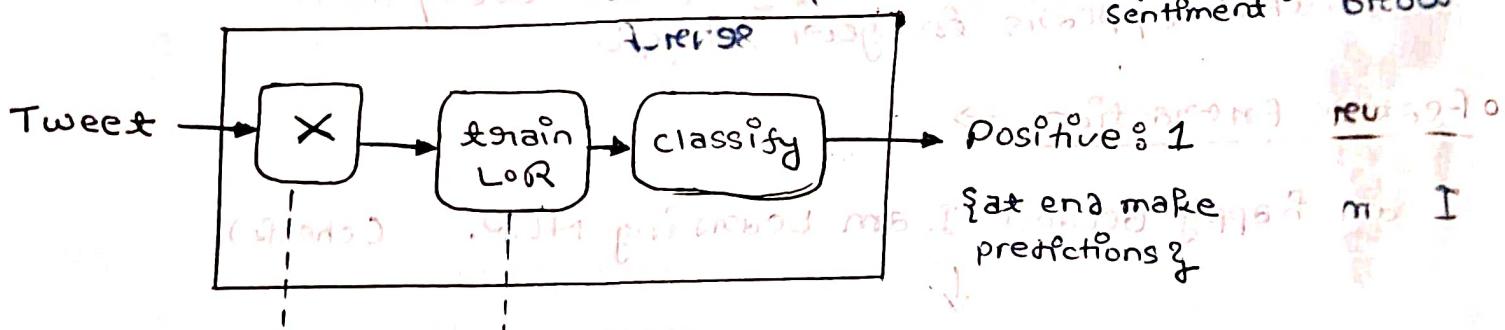
• Logistic Regression for sentiment Analysis of tweets:

In supervised ML,



Log whether the tweet have a positive or negative sentiment?

Tweet: I am Happy Because I am learning NLP



Extract useful features to train your classifier while minimizing the costs

Step 1. Vocabulary & feature extraction:

How to represent a text as a vector → first build a vocabulary, that will allow you to encode any text or any tweet as an array of numbers.

[the, is, the, eaten, an, apple, for, many, people, and, so]

[0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1]

The program is forward to

o Vocabulary

Tweets

[tweet1, tweet2, ..., tweet m]

I am happy because I am learning NLP.

I hated the movie.

- Vocabulary (V) is the list of unique words from your list of tweets.
 - To get that list you'll have to go through all the words from ~~the list~~ all your tweets every new word that appears in your search.
 - Feature Extraction →

I am happy Because I am Learning NLP. (check)

1

$\text{J} = [\text{I, am, happy, Because, learning, NLP, ---, Rated, the, movie}]$

1

- To do so, you will have to check every word from your vocabulary appears in the tweet. You will assign a value of 1 to that feature. If it doesn't appear you assign a value 0.

[I am happy, Because, learning NLP, I hated the movie]

$$1, \downarrow 1, \downarrow 1, \downarrow 1, 2 \quad 1, \quad 1, \quad \dots, \quad 0, \downarrow 0, \downarrow 0, \quad 0] .$$

6 count of 1's & many 0's

Problems with "sparse" Representations —

- A lot of zeros → that's a sparse representation.

e.g. I am happy because I am Learning NLP



[1, 1, 1, 1, 1, 1, ..., 0, 0, 0].

- This representation would have a number of features equal to the size of your entire vocabulary
- This would have a lot of features equal to 0 for every tweet
- With the sparse representation, a logistic regression model would have to learn $(n+1)$ parameters, where $n = \text{size of your vocabulary}$.

$$n = |V| \rightarrow [\theta_0, \theta_1, \theta_2, \dots, \theta_n]$$

Hence, ① Large training time

② Large prediction time.

Positive and Negative frequencies —

Corpus

I am happy Because I am Learning NLP.

I am happy

I am sad, I am not Learning NLP.

I am sad.

↑

↓

Positive

Negative

Vocabulary

I

am

happy

Because

learning

NLP

sad

not

Positive Tweets

I am happy Because I am Learning NLP
 I am happy

Negative tweets

I am sad, I am not learning NLP.
 I am sad.

| Vocabulary | PosFreq(1) | NegFreq(0) |
|------------|------------|------------|
| I | 3 | 3 |
| am | 3 | 3 |
| happy | 2 | 0 |
| because | 1 | 0 |
| learning | 1 | 1 |
| NLP | 1 | 1 |
| sad | 0 | 2 |
| not | 0 | 1 |

- Previously we encoded a ~~text~~ tweet as a vector of dimension CV . Now, we'll learn to encode a tweet or specially represented as a vector of dimension 3.

freqs : dictionary mapping from (word, class) to frequency

$$x_m = \left[1, \sum_w \text{freqs}(w, 1), \sum_w \text{freqs}(w, 0) \right]$$

↓ ↓ ↓
 arbitrary tweet m sum of pos freqs sum of neg freqs

or
 features of
 tweet m

(for ~~some~~ example)

Problems with "Sparse" Representations —

- A lot of zeros → that's a sparse representation.

e.g. I am happy because I am Learning NLP



[1, 1, 1, 1, 1, 1, ..., 0, 0, 0].

- This representation would have a number of features equal to the size of your entire vocabulary
- This would have a lot of features equal to 0 for every tweet
- With the sparse representation, a logistic regression model would have to learn $(n+1)$ parameters, where $n = \text{size of your vocabulary}$

$$n = |V| \rightarrow [e_0, e_1, e_2, \dots, e_n]$$

Hence, ① Large training time

② Large prediction time.

Positive and Negative frequencies

Corpus

I am happy because I am Learning NLP.

I am happy

I am sad, I am not Learning NLP.

I am sad.

Happy

Because

Learning

NLP

Sad

Not

Vocabulary

I

am

happy

Because

learning

NLP

sad

not

Eg1 I am sad, I am not learning NLP.



$$\begin{array}{l} \text{Vocab} = \text{I } \downarrow \text{ am } \downarrow \text{ happy } \downarrow \text{ because } \downarrow \text{ Learning } \downarrow \text{ NLP } \downarrow \text{ sad } \downarrow \text{ not} \\ (\text{Posfreq}) \quad 3 + 3 + \cancel{\text{x}} + \cancel{\text{x}} + 1 + 1 + 0 + 0 = 8 \end{array}$$

$$\begin{array}{l} \text{Vocab} : \text{I } \downarrow \text{ am } \downarrow \text{ happy } \downarrow \text{ because } \downarrow \text{ Learning } \downarrow \text{ NLP } \downarrow \text{ sad } \downarrow \text{ not} \\ \text{Negfreq} : \downarrow \quad \downarrow \\ 3 + 3 + \cancel{\text{x}} + \cancel{\text{x}} + 1 + 1 + 2 + 1 = 11 \end{array}$$

$$X^m = [2, 8, 11].$$

Putting it all together →

I am Happy Because i am Learning NLP @deeplearning

↓ [preprocessing] → report.html = 20218

[happy, learn, nep].

 Feature extraction (Preprocessing)

$bias \leftarrow [1, 4, 2]$ (freq of 1, 0, -1)

I am Happy Because I am Learning NLP @deeplearning

I am sad not learning NLP

o o o o
o o o o

I am sad not learning NLP
I am happy learning NLP

I am sad :c

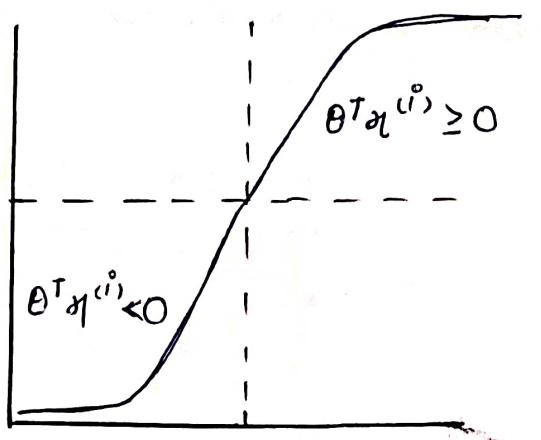
[sad, net, learn, nlp] → [1, 20, 50]

map

→ [1, 5, 35]

$$\left[\begin{array}{ccc} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \\ \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} \end{array} \right] \quad \longleftrightarrow \quad \left[[1, 40, 20], [1, 20, 50], \dots, [1, 5, 35] \right].$$

$$h(x^{(i)}, \theta) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$$

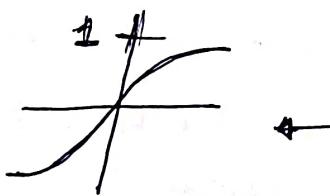


@Yousri and
@Andrew Ng are tuning a
GREAT AI model

[tun, ai, great, model]

$$x^{(i)} = \begin{bmatrix} 1 \\ 3476 \\ 245 \end{bmatrix}$$

Sum of all pos & neg freqs of all the words in your processed tweets



$$\theta = \begin{bmatrix} 0.00003 \\ 0.00150 \\ -0.00120 \end{bmatrix}$$

Outline

- Probabilities
- Baye's Rule (Applied in diff. field, including NLP)
- Build your own Naïve Bayes Tweet Classifier

Corpus of Tweets

| | | Positive | Positive | Positive |
|----------|----------|----------|----------|----------|
| | | Positive | Negative | Negative |
| Positive | Positive | 13 | 7 | 0 |
| | Negative | 0 | 10 | 10 |
| Total | | 13 | 17 | 20 |

Tweets containing the word "happy".

| | | Positive | Positive | Positive |
|----------|----------|----------|----------|----------|
| | | Positive | Negative | Negative |
| Positive | Positive | 4 | 1 | 0 |
| | Negative | 0 | 0 | 1 |
| Total | | 4 | 1 | 1 |

$A \rightarrow$ Positive tweet

$$P(A) = N_{\text{pos}} / N \rightarrow 13/20$$

$$\rightarrow 0.65$$

$P(\text{Negative}) = 1 - P(\text{positive})$

$$\rightarrow 1 - 0.65 \rightarrow 0.35$$

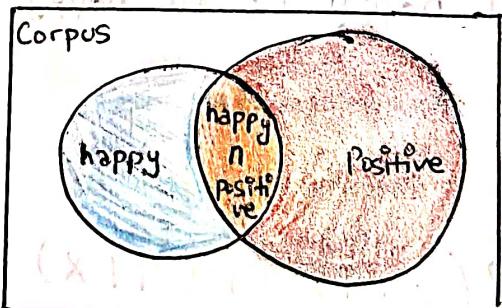
$B \rightarrow$ tweet containing "Happy"

$$P(B) = P(\text{happy}) = N_{\text{happy}} / N$$

$$\rightarrow 4/20 \rightarrow 0.2$$

Probability of the intersection

| | | Positive | Positive | Positive |
|----------|----------|----------|----------|----------|
| | | Positive | Negative | Negative |
| Positive | Positive | 4 | 1 | 0 |
| | Negative | 0 | 0 | 1 |
| Total | | 4 | 1 | 1 |



$$P(A \cap B) = P(A, B) = \frac{3}{20} = 0.15$$

$$P(A|B) = P(\text{positive} | \text{"happy"})$$

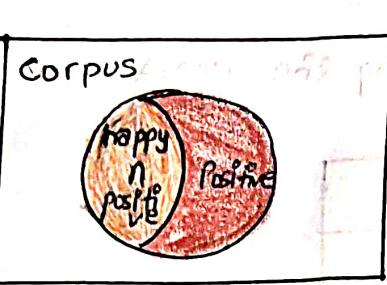
$$= \frac{3}{4} \Rightarrow 0.75$$

prob. of tweet is Positive given that it contains word happy



$$P(B|A) = P(\text{"happy"} | \text{positive})$$

$$= \frac{3}{13} \Rightarrow 0.231$$



Baye's Rule →

$$P(\text{Positive} | \text{"happy"}) = \frac{P(\text{Positive} \cap \text{"happy"})}{P(\text{"happy"})}$$

$$\text{LHS} = (0.75)(0.231) = 0.173$$

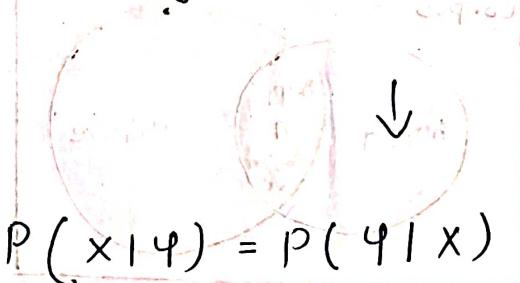
$$P(\text{"happy"}) = 0.231$$

$$P(\text{"happy"} | \text{Positive}) = \frac{P(\text{"happy"} \cap \text{Positive})}{P(\text{Positive})}$$

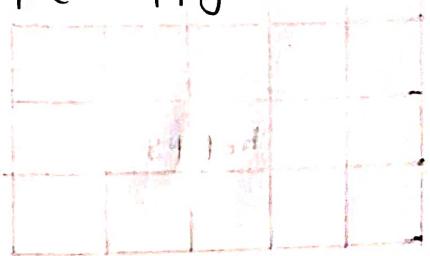
$$P(\text{Positive}) = 0.75$$



$$P(\text{Positive} | \text{"happy"}) = P(\text{"happy"} | \text{Positive}) \times \frac{P(\text{Positive})}{P(\text{"happy"})}$$



$$P(X \cap Y) = P(Y | X) \times \frac{P(X)}{P(Y)}$$



Naive Bayes for Sentiment Analysis

Positive tweets

I am happy Because I am learning NLP.

I am happy, not sad.

Negative tweets

I am sad, I am not learning NLP.

I am sad, not happy

word

I

am

happy

because

learning

NLP

sad

not

freq

(Counting words)

| word | pos | neg |
|----------|-----|-----|
| I | 3 | 3 |
| am | 3 | 3 |
| happy | 2 | 1 |
| because | 1 | 0 |
| learning | 1 | 1 |
| NLP | 1 | 1 |
| sad | 1 | 2 |
| not | 1 | 2 |

| word | pos | neg |
|----------|--|--|
| I | $P\left(\frac{I}{\text{pos}}\right) = \frac{3}{13} = 0.24$ | $P\left(\frac{I}{\text{neg}}\right) = \frac{3}{12} = 0.25$ |
| am | $P\left(\frac{\text{am}}{\text{pos}}\right) = \frac{3}{13} = 0.24$ | $P\left(\frac{\text{am}}{\text{neg}}\right) = \frac{3}{12} = 0.25$ |
| happy | 0.15 | 0.08 |
| because | 0.08 | 0.04 |
| learning | 0.08 | 0.08 |
| NLP | 0.08 | 0.08 |
| sad | 0.08 | 0.17 |
| not | 0.08 | 0.17 |

N class 13 12

↓ helps in computing the conditional probabilities of word in the class

$P(W_i | \text{class})$

e.g. predict sentiment of whole tweet

Tweet : I am happy today ; I am Learning

$$\prod_{i=1}^m \frac{P(w_i | \text{pos})}{P(w_i | \text{neg})}$$

0.20 * 0.20 * 0.14 * 0.20 * 0.20 * 0.10
0.20 * 0.20 * 0.10
= \frac{0.14}{0.10} \Rightarrow 1.4 > 1
(positive sentiment)

a nearly identical conditional probability, words that are equally probable & don't add anything to the sentiment

These are your power words, These carry a lot of weights in determining in your tweet sentiments

conditional probability for the negative class is zero - when this happens no way of comparing b/w positive tweets & negative tweets corpora.

which will become a problem for your calculations

$\prod_{i=1}^m \frac{P(w_i | \text{Pos})}{P(w_i | \text{Neg})}$ Naïve Baye's inference conditional rule for binary classifications

This expression says you are going to take the product across all of the words in your tweets of the probability for each word in positive class divide it by the prob. in neg. class

Naïve Bayes Classifier is a supervised learning algorithm based on Baye's theorem.

It's a probabilistic classifier, which mean it predict on the basis of probability of an object.

Bayes theorem →

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

$$P(A \cap B) = P(A|B) * P(B) = P(B|A) * P(A)$$

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

or

$$\left. \begin{array}{l} P(A \cap B) = P(B|A) * P(A) \\ P(A \cap B) = \frac{P(B|A) * P(A)}{P(B)} \end{array} \right\} \rightarrow \text{Bayes Theorem}$$

$$P(\psi|x) = \frac{P(x|\psi) * P(\psi)}{P(x)}$$

where ψ is class variable and x is dependent feature variable

$$\hookrightarrow (x_1, x_2, x_3, \dots, x_n)$$

$$P(\psi|x_1, x_2, \dots, x_n) = \frac{P(\psi) * P(x_1|x_2, x_3, \dots, x_n|\psi)}{P(x_1, x_2, x_3, \dots, x_n)}$$

$$\rightarrow \frac{P(\psi) * P(x_1|\psi) * P(x_2|\psi) * \dots * P(x_n|\psi)}{(P(x_1) * P(x_2) * P(x_3) * \dots * P(x_n))}$$

ψ = yes or no / 0 or 1 / T or F

$$\frac{P(\psi)}{\exp(-\psi)} * \frac{P(x_1|\psi)}{\exp(-x_1)} * \frac{P(x_2|\psi)}{\exp(-x_2)} * \dots * \frac{P(x_n|\psi)}{\exp(-x_n)} * \frac{P(\psi)}{P(1-\psi)} * \frac{P(x_1)}{P(1-x_1)} * \dots * \frac{P(x_n)}{P(1-x_n)}$$

$$= \frac{P(\psi)}{\exp(-\psi)} * \frac{P(x_1|\psi)}{\exp(-x_1)} * \frac{P(x_2|\psi)}{\exp(-x_2)} * \dots * \frac{P(x_n|\psi)}{\exp(-x_n)} * \frac{P(\psi)}{P(1-\psi)} * \frac{P(x_1)}{P(1-x_1)} * \dots * \frac{P(x_n)}{P(1-x_n)}$$

$$= \frac{P(\psi)}{\exp(-\psi)} * \frac{P(x_1|\psi)}{\exp(-x_1)} * \frac{P(x_2|\psi)}{\exp(-x_2)} * \dots * \frac{P(x_n|\psi)}{\exp(-x_n)} * \frac{P(\psi)}{P(1-\psi)} * \frac{P(x_1)}{P(1-x_1)} * \dots * \frac{P(x_n)}{P(1-x_n)}$$

| Day | Outlook | Temp | Humidity | Wind | Play |
|-----|----------|------|----------|--------|------|
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Strong | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |
| 15 | Sunny | Cool | High | Strong | ? |

Day 15 $P(\text{Yes} | \text{sunny, cool, high, strong})$

\uparrow class variable \downarrow dependent feature vector

$$\Rightarrow P(\text{Yes}) * P(\text{sunny, cool, high, strong} | \text{Yes})$$

$$(ax) P(\text{sunny, cool, high, strong})$$

$$\Rightarrow P(\text{Yes}) * P(\text{sunny} | \text{Yes}) * P(\text{cool} | \text{Yes, sunny}) * P(\text{high} | \text{Yes, sunny, cool}) * P(\text{strong} | \text{Yes, sunny, cool, high})$$

$$P(\text{sunny, cool, high, strong})$$

$$\rightarrow \frac{9}{14} * \frac{2/14}{8/14} * \frac{3}{9} * \frac{3}{9} * \frac{4}{9} \xrightarrow{\substack{\text{not } 1 \text{ & cool} \\ \uparrow \\ p(\text{high} | \text{yes})}} \frac{g+2+3+3+4}{14+9+9+9+9}$$

$$\xrightarrow{\substack{\text{not } 1 \text{ & cool} \\ \uparrow \\ p(\text{strong} | \text{yes})}}$$

$$\frac{p(\text{cool} \cap \text{yes} \cap \text{sunny})}{p(\text{yes} \cap \text{sunny})} \rightarrow \frac{p(\text{cool} | \text{yes})}{p(\text{yes} | \text{sunny})}$$

$$\rightarrow 0.007$$

$$P(\text{No} | \text{sunny, cool, high, strong})$$

↑ class variable ↓ dependent feature vector

$$\Rightarrow P(\text{No}) * P(\text{sunny} | \text{No}) * P(\text{cool} | \text{No, sunny}) * P(\text{high} | \text{No, sunny, cool}) * P(\text{strong} | \text{No, sunny, cool, high})$$

$$P(\text{No}) * P(\text{sunny} | \text{No}) * P(\text{cool} | \text{No}) * P(\text{high} | \text{No}) * P(\text{strong} | \text{No})$$

$$\Rightarrow \frac{5}{14} * \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5} \rightarrow \frac{5 \times 3 \times 4 \times 3}{14 \times 5 \times 5 \times 5} \rightarrow 0.0206$$

wind

| | | | |
|--------|-----|---------------------------------------|--|
| Strong | yes | $P(\text{strong} \text{Yes}) = 4/9$ | $\frac{P(\text{strong} \cap \text{Yes})}{P(\text{Yes})}$ |
| Strong | no | $P(\text{strong} \text{No}) = 3/5$ | $\frac{P(\text{strong} \cap \text{No})}{P(\text{No})}$ |
| weak | yes | $P(\text{weak} \text{Yes}) = 5/9$ | |
| weak | no | $P(\text{weak} \text{No}) = 2/5$ | |

humidity

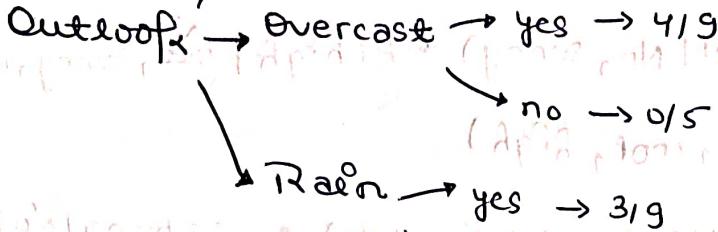
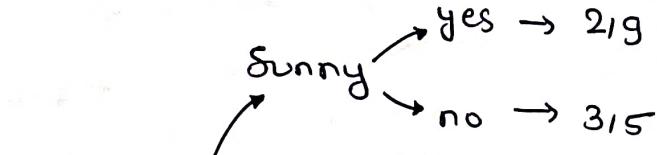
| | | |
|--------|-----|--|
| High | yes | $P(\text{High} \text{Yes}) = \frac{P(\text{High} \cap \text{Yes})}{P(\text{Yes})} \rightarrow 3/9$ |
| High | No | $P(\text{High} \text{No}) = \frac{P(\text{High} \cap \text{No})}{P(\text{No})} \rightarrow 4/5$ |
| Normal | yes | $P(\text{Normal} \text{Yes}) = \frac{P(\text{Normal} \cap \text{Yes})}{P(\text{Yes})} \rightarrow 6/9$ |
| Normal | No | $P(\text{Normal} \text{No}) = \frac{P(\text{Normal} \cap \text{No})}{P(\text{No})} \rightarrow 1/5$ |

Temp

| | | |
|------|-----|-------|
| Hot | yes | $2/9$ |
| Hot | no | $2/5$ |
| Mild | yes | $4/9$ |
| Mild | no | $2/5$ |
| Cold | yes | $3/9$ |
| Cold | no | $1/5$ |

Reliability of a Word

(prob. of a word given full information)



$$P(E) = \frac{2/9 + 4/9}{5/9} = \frac{6/9}{5/9} = \frac{6}{5}$$

$$\begin{aligned} P(F) &= \frac{0/5 + 0/5}{5/9} = \frac{0}{5/9} = 0 \\ P(G) &= \frac{2/5 + 3/9}{5/9} = \frac{18/45 + 15/45}{5/9} = \frac{33/45}{5/9} = \frac{33}{25} \\ P(H) &= \frac{(2/5)(3/9)}{5/9} = \frac{6/45}{5/9} = \frac{6}{25} \end{aligned}$$

$$P(E) = \frac{2/9 + 4/9}{5/9} = \frac{6/9}{5/9} = \frac{6}{5}$$

$$P(F) = \frac{0/5 + 0/5}{5/9} = \frac{0}{5/9} = 0$$

$$P(G) = \frac{2/5 + 3/9}{5/9} = \frac{18/45 + 15/45}{5/9} = \frac{33/45}{5/9} = \frac{33}{25}$$

$$P(H) = \frac{(2/5)(3/9)}{5/9} = \frac{6/45}{5/9} = \frac{6}{25}$$

$$\begin{aligned} P(E) &= \frac{2/9}{5/9} = \frac{2}{5} \\ P(F) &= \frac{0/5}{5/9} = \frac{0}{5} = 0 \\ P(G) &= \frac{33/45}{5/9} = \frac{33}{25} \end{aligned}$$

$$\begin{aligned} P(H) &= \frac{6/45}{5/9} = \frac{6}{25} \\ P(E) &= \frac{2/9}{5/9} = \frac{2}{5} \\ P(F) &= \frac{0/5}{5/9} = \frac{0}{5} = 0 \end{aligned}$$

$$\begin{aligned} P(G) &= \frac{33/45}{5/9} = \frac{33}{25} \\ P(H) &= \frac{6/45}{5/9} = \frac{6}{25} \end{aligned}$$

Laplacian Smoothing

Sometimes you try to calculate the probability of a word happening after a word. To do that you might want to count the nos of times those two words showed up. One after another, divided by the number of times the first word appears. Now what if the ~~two~~ two words never showed up next to each other in the training corpus? You get a probability of zero, and the probability of an entire sequence might go to zero.

Now, how can we fix this?

previously, $p(w_i | \text{class}) = \frac{\text{freq}(w_i, \text{class})}{N_{\text{class}}}$

Class $\in \{\text{positive}, \text{negative}\}$

N_{class} = frequency of all words in class

V_{class} = nos of unique words in class

$$p(w_i | \text{class}) = \frac{\text{freq}(w_i | \text{class}) + 1}{N_{\text{class}} + V_{\text{class}}}$$

[Laplacian Smoothing]

avoids the prob. of being zero,

| word | pos | neg |
|----------|-----|-----|
| I | 3 | 3 |
| am | 3 | 3 |
| happy | 2 | 1 |
| Because | 1 | 0 |
| Learning | 1 | 1 |
| NLP | 1 | 1 |
| sad | 1 | 2 |
| not | 1 | 2 |

$$N_{\text{class}} = 13 \quad V = 8$$

| word | pos | neg |
|----------|--------------------------------|--------------------------------|
| I | $p(I \text{pos})$ | $p(I \text{neg})$ |
| am | $\Rightarrow \frac{3+1}{13+8}$ | $\Rightarrow \frac{3+1}{12+8}$ |
| happy | $\Rightarrow 0.19$ | $\Rightarrow 0.20$ |
| Because | 0.19 | 0.20 |
| Learning | 0.14 | 0.10 |
| NLP | 0.10 | 0.10 |
| sad | 0.10 | 0.05 |
| not | 0.10 | 0.10 |

$$\sum_{i=1}^m p(w_i|\text{pos}) = 1 - 0.12 = 0.88$$

[Laplacian Smoothing]

• Laplacian smoothing to avoid zero probability

$$P(w_i|\text{class})$$

$$= \frac{(0.19 + 0.19) * 0.19}{0.20 + 1} = \frac{0.19 * 0.19}{0.20 + 1} = \frac{0.14 * 0.10}{0.10 + 1} = \frac{0.10 * 0.10}{0.10 + 1} = \frac{0.10}{0.15}$$

$$* \frac{0.10}{0.15}$$

$$\Rightarrow \frac{0.00005054}{0.000045} \Rightarrow 1.12 > 1$$

Log Likelihoods

Words can have many shades of emotional meaning, but for the purpose of sentiment classification, they are simplified into three categories: neutral, positive & negative.

All can be identified by using their conditional probabilities →

$$\text{ratio}(w_i) = \frac{P(w_i | \text{Pos})}{P(w_i | \text{Neg})}$$

| word | Pos | neg | ratio | | |
|----------|------|------|-------|---|----------|
| I | 0.20 | 0.20 | 1 | ∞ | Positive |
| am | 0.20 | 0.20 | 1 | | |
| Happy | 0.14 | 0.10 | 1.4 | 1 | Neutral |
| Because | 0.10 | 0.10 | 1 | | |
| learning | 0.10 | 0.10 | 1 | | |
| NLP | 0.10 | 0.10 | 1 | | |
| Sad | 0.10 | 0.15 | 0.6 | 0 | negative |
| not | 0.10 | 0.15 | 0.6 | | |

- The larger the ratio the more positive the word is going to be.
- Now, on the other hand, negative words have a ratio smaller than 1.

$$\approx \frac{\text{freq}(w_i, 1) + 1}{\text{freq}(w_i, 0) + 1}$$

Naïve Bayes Inference

Class $\in \{\text{pos}, \text{neg}\}$

$w \rightarrow$ set of m words in a tweet

$$\prod_{i=1}^m \frac{P(w_i | \text{Pos})}{P(w_i | \text{Neg})} > 1 \quad \frac{P(\text{Pos})}{P(\text{Neg})}$$

likelihood prior ratio

- A simple, fast and powerful Baseline.
- A probabilistic method used for classification

Log Likelihood → • Sentiments probability calculations requires multiplication of many numbers with values between 0 and 1.

- Carrying out such multiplications on a computer can the risk of numerical underflow when the number returned is so small it can't be stored on your device
- mathematical trick to solve this -

$$\log(a * b) = \log(a) + \log(b)$$

$$\log \left(\frac{P(\text{Pos})}{P(\text{Neg})} \prod_{i=1}^m \frac{P(w_i | \text{Pos})}{P(w_i | \text{Neg})} \right)$$

log (prior multiplied by likelihood)

$$\Rightarrow \log(p_{\text{prior}}) + \log(\text{likelihood})$$

or

$$\log \frac{P(\text{Pos})}{P(\text{neg})} + \sum_{i=1}^n \log \frac{P(w_i | \text{Pos})}{P(w_i | \text{neg})}$$

Calculating Lambda

$$\lambda(I) = \log \frac{0.05}{0.05} = 0$$

| word | pos | neg | λ |
|----------|------|------|-----------|
| I | 0.05 | 0.05 | 0 |
| am | 0.04 | 0.04 | 0 |
| happy | 0.09 | 0.01 | 2.2 |
| Because | 0.01 | 0.01 | 0 |
| learning | 0.03 | 0.01 | 1.1 |
| NLP | 0.02 | 0.02 | 0 |
| sad | 0.01 | 0.09 | -2.2 |
| not | 0.02 | 0.03 | -0.4 |

$$\lambda(w) = \log \frac{P(w | \text{Pos})}{P(w | \text{neg})}$$

$$\text{ratio}(w) = \frac{P(w | \text{Pos})}{P(w | \text{neg})}$$

word
sentiment

$$\lambda(w) = \log \frac{P(w | \text{Pos})}{P(w | \text{neg})}$$

I am happy Because I am learning

↓ ↓ ↓ ↓ ↓ ↓ ↓

$$\log \rightarrow 0 + 0 + 2.2 + 0 + 0 + 0 + 1.1$$

likelihood $\Rightarrow 3.3$

$$\prod_{i=1}^m \frac{p(w_i^i | pos)}{p(w_i^i | neg)} > 1$$

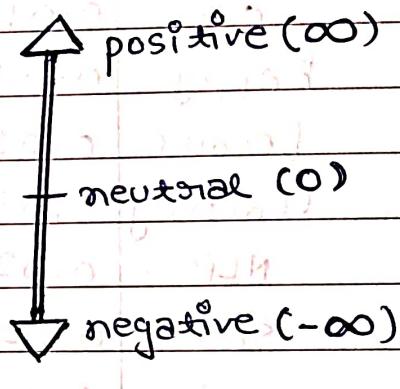
$$\sum_{i=1}^m \log \frac{p(w_i^i | pos)}{p(w_i^i | neg)} > 0$$

or $\sum \log \lambda(w_i^i) > 0$

$$3 \cdot 3 > 0$$

Tweet sentiment,

$$\log \prod_{i=1}^m \text{ratio}(w_i^i) = \sum_{i=1}^m \lambda(w_i^i) > 0$$



Step 5: Get the log ratio

- Training Naïve Bayes

Step 0: Collect and Annotate Corpus

| positive tweets | |
|--------------------------------------|--|
| I am happy Because I am Learning NLP | |
| I am happy, not sad. @ NLP | |
| Negative tweets | |
| I am sad, I am not learning NLP | |
| I am sad, not happy !! | |

Step 1: Data-Preprocessing

- Lowercase
- Remove punctuation, urls, names
- Remove Stopwords
- Stemming
- Tokenize sentences

| positive tweets | |
|------------------------------|-------------------|
| [happy, Because, learn, NLP] | (S) 3.61 (M) 1.72 |
| [happy, not, sad] | (C) 0.01 (W) 0.01 |
| Negative tweets | |
| [sad, not, learn, NLP] | (S) 0.01 (M) 0.01 |
| [sad, not, happy] | (C) 0.01 (W) 0.01 |

Step 2: word-count

Once, you have a clear corpus of processed tweets, you will start by computing the vocabulary for each word in class,

$\text{freq}(w, \text{class})$

| word | Pos | Neg |
|---------|-----|-----|
| happi | 2 | 1 |
| because | 1 | 0 |
| Learn | 1 | 1 |
| NLP | 1 | 1 |
| sad | 1 | 2 |
| not | 1 | 2 |
| Nclass | 7 | 7 |

Step 3: $P(w|\text{class})$

$$\sqrt{\text{class}} = 6$$

$$\frac{\text{freq}(w, \text{class}) + 1}{N_{\text{class}} + \sqrt{\text{class}}}$$

$$N_{\text{class}} + \sqrt{\text{class}}$$

Step 4:
Get Lambda (λ)

$$\lambda(w) = \log \frac{P(w|\text{pos})}{P(w|\text{neg})}$$

| word | Pos | Neg |
|---------|------|------|
| happi | 0.23 | 0.15 |
| because | 0.15 | 0.07 |
| learn | 0.08 | 0.08 |
| NLP | 0.08 | 0.08 |
| sad | 0.08 | 0.17 |
| not | 0.08 | 0.17 |

| word | pos | neg | λ |
|---------|------|------|-----------|
| happi | 0.23 | 0.15 | 0.43 |
| because | 0.15 | 0.07 | 0.6 |
| learn | 0.08 | 0.08 | 0 |
| NLP | 0.08 | 0.08 | 0 |
| sad | 0.08 | 0.17 | -0.75 |
| not | 0.08 | 0.17 | -0.75 |

Step 5%: Get the log prior

D_{pos} = Nos of positive tweets

D_{neg} = Nos of negative tweets

$$\text{log prior} = \log \frac{D_{pos}}{D_{neg}}$$

If dataset is balanced, $D_{pos} = D_{neg}$ and $\text{log prior} = 0$

Testing Naïve Bayes →

- Predict using a Naïve Bayes Model
- Using your validation set to compute model accuracy

| word | λ |
|---------|-----------|
| I | -0.01 |
| the | -0.01 |
| happy | 0.63 |
| because | 0.01 |
| pass | 0.5 |
| NLP | 0 |
| sad | -0.75 |
| not | -0.75 |

• log-likelihood dictionary

$$\lambda(w) = \log \frac{P(w|pos)}{P(w|neg)}$$

$$\text{• log prior} = \log \frac{D_{pos}}{D_{neg}} = 0$$

• Tweet:
(check)

[I, pass, the, NLP, interview]
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

$$\text{Score} = -0.01 + 0.5 - 0.01 + 0 + 0 + \text{log prior}$$

$$\Rightarrow 0.48$$

$$pred = score > 0 \text{ (positive)}$$

$$\circ X_{val} \quad Y_{val} \quad \lambda \quad \text{logprior}$$

$$score = predict(X_{val}, \lambda, \text{logprior})$$

$$pred = score > 0$$

| | | |
|-----------|---------------|----------|
| 0.5 | $0.5 > 0$ | 1 |
| -1 | $-1 > 0$ | 0 |
| 1.3 | $1.3 > 0$ | 1 |
| \vdots | \vdots | \vdots |
| $Score_m$ | $Score_m > 0$ | $Pred_m$ |

produces a vector populated with 0's & 1's indicating if the predicted sentiment is negative or positive.

$$\begin{bmatrix} \varnothing \\ \frac{1}{1} \\ \vdots \\ \vdots \\ \text{pred}_m \end{bmatrix} = = \begin{bmatrix} \varnothing \\ \frac{0}{1} \\ \vdots \\ \vdots \\ \varnothing_{\text{val}_m} \end{bmatrix}$$

0 - 0 → 1
1 - 0 → 0
1 - 1 → 1

$$\frac{1}{m} \sum_{i=1}^m (\text{pred}_i == \varnothing_{\text{val}_i}) \Rightarrow$$

$$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$\text{pred}_m == \varnothing_{\text{val}_m}$

Summary • $X_{\text{val}} \cdot \varnothing_{\text{val}} \rightarrow \text{performance on unseen data}$

- predict using λ and logprior for each new tweet

• Accuracy $\rightarrow \frac{1}{m} \sum_{i=1}^m (\text{pred}_i == \varnothing_{\text{val}_i})$

- what about words that don't appear in $\lambda(w)$?

Applications of Naïve Bayes

$$P(\text{pos} | \text{tweet}) = P(\text{pos}) P(\text{tweet} | \text{pos})$$

$$P(\text{neg} | \text{tweet}) = P(\text{neg}) P(\text{tweet} | \text{neg})$$

① Twitter Sentiment Analysis

$$\frac{P(\text{pos} | \text{tweet})}{P(\text{neg} | \text{tweet})} \approx \frac{P(\text{pos})}{P(\text{neg})} \prod_{i=1}^m \frac{P(w_i | \text{pos})}{P(w_i | \text{neg})}$$

priors likelihood

estimating the probability for each class by using the joint probability of the words in classes

② Identification system

eg Author identification : $P(\text{"shakespeare"} | \text{book})$

$P(\text{"meadley"} | \text{book})$

If you have two large corpora, each written by two different authors, you could train a model to recognize whether a new document was written by one or another

③ Word disambiguation : Consider that you have only two possible interpretations of a given word within a text.

Let, say you don't know if the word Bank in your reading is referring to the bank of a river or to a financial institution

To disambiguate your word calculate the score of documents, given that it refers to each one of the possible meanings.

In this case, if the text refers to the concept of river instead of the concept of money, then the score will be bigger than one.

$P(\text{river} | \text{text})$

$P(\text{money} | \text{text})$

Decision Trees!

| Day | Outlook | Temp | Humidity | Wind | Play |
|-----|----------|------|----------|--------|------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Entropy measures homogeneity of examples,

- Entropy measures the impurity of a collection of examples
It depends from the distribution of the random variable p -

→ S is a collection of training examples

→ p_+ the proportion of positive examples in S

→ p_- the proportion of negative examples in S

▪ Entropy is the measure of disorder or impurity in a node

$$E = - \sum_{i=1}^n p_i \log_2 p_i$$

- Node with a more variable composition considered to be more entropy than node with less variable composition

- Entropy of universal fact = 0;

High Entropy \rightarrow less homogeneous / more impurity
less Entropy \rightarrow more homogeneous / less impurity

↳ Attribute of Outlook

Values (Outlook) = Sunny, Overcast, Rain

$$S = [9+, 5-] \quad \text{entropy}(S) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} \Rightarrow 0.94$$

$$S_{\text{Sunny}} \leftarrow [2+, 3-] \quad \text{entropy}(S_{\text{Sunny}}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \Rightarrow 0.971$$

$$S_{\text{Overcast}} \leftarrow [4+, 0-] \quad \text{Entropy}_{(\text{Overcast})} = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \Rightarrow 0$$

$$S_{\text{Rain}} \leftarrow [3+, 2-] \quad \text{Entropy}_{(\text{Rain})} = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \Rightarrow 0.971$$

▪ Information gain, measure of change in Entropy

$$\text{Gain}(S_{\text{outlook}}) = \text{Entropy}(S) - \sum_{v \in \{\text{sunny}, \text{overcast}, \text{rain}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S_{\text{outlook}}) = \text{Entropy}(S) - \frac{5}{14} \text{Entropy}(S_{\text{sunny}})$$

$$- \frac{4}{14} \text{Entropy}(S_{\text{overcast}}) - \frac{5}{14} \text{Entropy}(S_{\text{rain}})$$

$$\Rightarrow 0.94 - \frac{5}{14} (0.971) - \frac{4}{14} (0) - \frac{5}{14} (0.971) \Rightarrow 0.7464$$

Attribute : Temp

Values(Temp) : hot, mild, cold

$$S = [9+, 5-] \quad \text{Entropy}(S) \rightarrow -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} \Rightarrow 0.94$$

$$\text{Shot} \leftarrow [2+, 2-] \quad \text{Entropy}(S_{\text{shot}}) \rightarrow -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{2} \Rightarrow 1.00$$

$$\text{Smild} \leftarrow [4+, 2-] \quad \text{Entropy}(S_{\text{mild}}) \rightarrow -\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{2} \Rightarrow 0.9183$$

$$\text{Scold} \leftarrow [3+, 1-] \quad \text{Entropy}(S_{\text{cold}}) \rightarrow -\frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{2} \Rightarrow 0.8132$$

$$\text{Gain}(S, \text{Temp}) \Rightarrow \text{Entropy}(S) - \sum_{v \in \{\text{hot}, \text{mild}, \text{cold}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(CS, \text{Temp}) \Rightarrow \text{Entropy}(S) = \frac{4}{15} \text{Entropy}(S_{hot}) - \frac{6}{14} \text{Entropy}(S_{mid}) - \frac{4}{14} \text{Entropy}(S_{cool})$$

$$\Rightarrow 0.94 - \frac{4}{14} (1.0) - \frac{6}{14} (0.9183) - \frac{4}{14} (0.8113) \Rightarrow 0.0289$$

Attribute : Humidity

Values (Humidity) = High, Normal

$$S = [g+, 5-] \quad \text{Entropy}(S) = \frac{-9}{14} \log_2 \frac{9}{2} - \frac{5}{14} \log_2 \frac{5}{2} \Rightarrow 0.94$$

$$S_{high} \leftarrow [3+, 4-] \quad \text{Entropy}_{(S_{high})} = \frac{-3}{7} \log_2 \frac{3}{2} - \frac{4}{7} \log_2 \frac{4}{2} = 0.9052$$

$$S_{normal} \leftarrow [6+, 1-] \quad \text{Entropy}_{(S_{normal})} = \frac{-6}{7} \log_2 \frac{6}{2} - \frac{1}{7} \log_2 \frac{1}{2} \Rightarrow 0.5916$$

$$\text{Gain}(CS, \text{Humidity}) = \text{Entropy}(S) - \sum_{v \in \{\text{High, Normal}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(CS, \text{Humidity}) \Rightarrow 0.94 - \frac{7}{14} (0.9052) - \frac{7}{14} (0.5916) = 0.1516$$

Attribute : Wind

Values (Wind) = Strong, Weak

$$S = [g+, 5-] \quad \text{Entropy}(S) \Rightarrow 0.94$$

$$S_{\text{Strong}} \leftarrow [3+, 3-] \quad \text{Entropy}(S_{\text{Strong}}) = 1.0$$

$$S_{\text{Weak}} \leftarrow [6+, 2-] \quad \text{Entropy}(S_{\text{Weak}}) \Rightarrow -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8} \Rightarrow 0.813$$

$$\text{Gain}(S, \text{Wind}) = \text{Entropy}(S) - \sum_{v \in \{\text{Strong}, \text{Weak}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$\text{Gain}(S, \text{Wind}) = \text{Entropy}(S) - \frac{6}{14} \text{Entropy}(S_{\text{Strong}}) - \frac{8}{14} \text{Entropy}(S_{\text{Weak}})$$

$$\text{Gain}(S, \text{Wind}) = 0.94 - \frac{6}{14} (1.0) - \frac{8}{14} (0.813) \Rightarrow 0.0478$$

$$\text{Gain}(S, \text{Outlook}) \rightarrow 0.2464$$

$$\text{Gain}(S, \text{Temp}) \rightarrow 0.0289$$

$$\text{Gain}(S, \text{Humidity}) \rightarrow 0.1516$$

$$\text{Gain}(S, \text{Wind}) \rightarrow 0.0478$$

$\{D_1, D_2, \dots, D_{14}\}$

$[9+, 5-]$

outlook

Sunny
 $\{D_1, D_2, D_4, D_5, D_9, D_{11}\}$

Overcast
 $\{D_3, D_7, D_{12}, D_{13}\}$

Rain
 $\{D_4, D_5, D_6, D_{10}, D_{14}\}$

$[2+, 3-]$

Yes

$[3+, 2-]$

?

?

| Day | Temp | Humidity | Wind | Play |
|-----|------|----------|--------|------|
| D1 | Hot | High | Weak | No |
| D2 | Hot | High | Strong | No |
| D8 | Mild | High | Weak | No |
| D9 | Cool | Normal | Weak | Yes |
| D11 | Mild | Normal | Strong | Yes |

Attribute : Temp

Values(Temp) = Hot, Mild, Cool

$$S_{\text{Sunny}} = [2+, 3-] \quad \text{Entropy}(S_{\text{Sunny}}) = -\frac{2}{5} \log_2 \frac{2}{2} - \frac{3}{5} \log_2 \frac{3}{2} \Rightarrow 0.97$$

$$S_{\text{Hot}} = [0+, 2-]$$

$$S_{\text{Mild}} = [1+, 1-]$$

$$S_{\text{Cool}} = [1+, 0-]$$

$$\text{Entropy}(S_{\text{Hot}}) \downarrow$$

$$0.0$$

$$\text{Entropy}(S_{\text{Mild}}) \downarrow$$

$$1.0$$

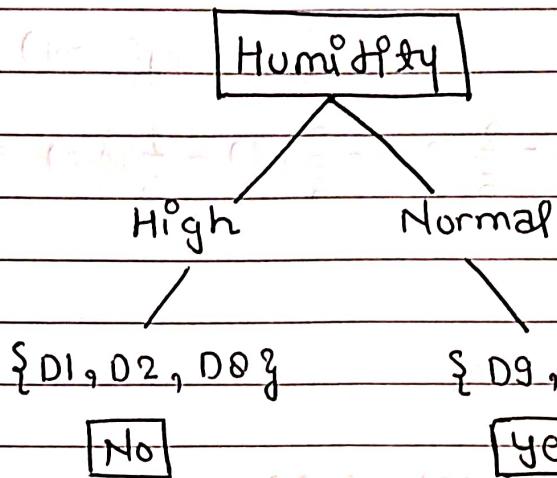
$$\text{Entropy}(S_{\text{Cool}}) \downarrow$$

$$0.0$$

$$Gain(S_{\text{sunny}}, \text{temp}) = 0.570$$

$$Gain(S_{\text{sunny}}, \text{Humidity}) = 0.97 \leftarrow \text{node}$$

$$Gain(S_{\text{sunny}}, \text{Wind}) = 0.0192$$



| Day | Temp | Humidity | Wind | Play |
|-----|------|----------|--------|------|
| D4 | Mild | High | Weak | yes |
| D5 | Cool | Normal | Weak | yes |
| D6 | Cool | Normal | Strong | No |
| D10 | Mild | Normal | Weak | yes |
| D14 | Mild | High | Strong | No |

Attribute: Humidity

Values (Humidity) = High, Normal

Sunny [3+, 2-] Entropy $\Rightarrow 0.97$
 (Ssunny)

Shigh $\leftarrow [1+, 1-]$ Entropy (Shigh) $\Rightarrow 1.0$

Snormal $\leftarrow [2+, 1-]$ Entropy (Snormal) $\Rightarrow 0.9183$

Gain (Srain, humidity) $\Rightarrow 0.0192$

$$\text{Gain}(S_{\text{sunny}}, \text{Temp}) = \text{Entropy}(S) - \sum_{v \in \{\text{hot}, \text{cool}, \text{mild}\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - \frac{2}{5} \text{Entropy}(S_{\text{hot}}) - \frac{2}{5} \text{Entropy}(S_{\text{mild}}) - \frac{1}{5} \text{Entropy}(S_{\text{cool}})$$

$$\text{Gain}(S_{\text{sunny}}, \text{Temp}) \Rightarrow 0.97 - \frac{2}{5}(0.0) - \frac{2}{5}(1) - \frac{1}{5}(0.0) \Rightarrow 0.570$$

Attribute : Humidity

Values(Humidity) : High, Normal

$$S_{\text{sunny}} = [2+, 3-] \quad \text{Entropy}(S) = 0.97$$

$$S_{\text{high}} \leftarrow [0+, 3-] \quad \text{Entropy}(S_{\text{high}}) = 0.0$$

$$S_{\text{normal}} \leftarrow [2+, 0-] \quad \text{Entropy}(S_{\text{normal}}) = 0.0$$

$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) \Rightarrow 0.97$$

Attribute : Wind

Values(Wind) : Strong, Weak

$$S_{\text{sunny}} = [2+, 3-] \quad \text{Entropy}(S) = 0.97$$

$$S_{\text{strong}} \leftarrow [1+, 1-] \quad \text{Entropy}(S_{\text{strong}}) = 1.0$$

$$S_{\text{weak}} \leftarrow [1+, 2-] \quad \text{Entropy}(S_{\text{weak}}) = 0.9103$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) \Rightarrow 0.0192$$

Attribute: Wind

Values (Wind): strong, weak

$$S_{Rain} \leftarrow [3+, 2-] \quad \text{Entropy}(S) = 0.97$$

$$S_{Strong} \leftarrow [0+, 2-] \quad \text{Entropy}(S_{Strong}) = 0.0$$

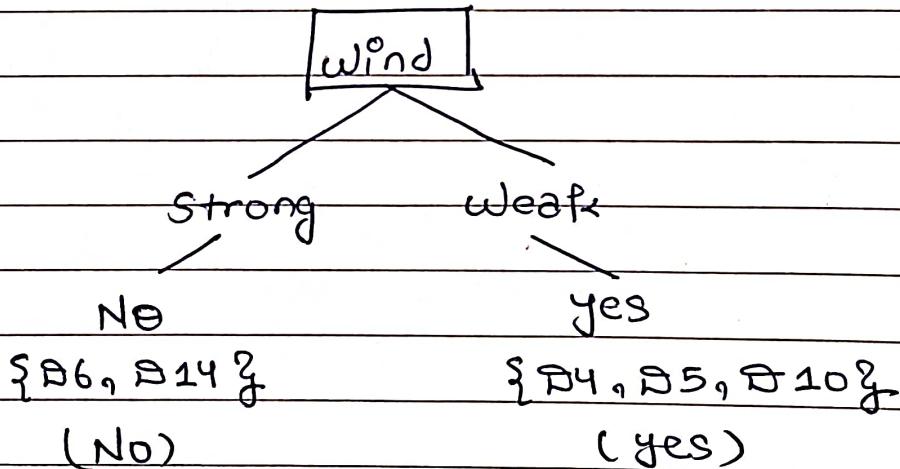
$$S_{Weak} \leftarrow [3+, 0-] \quad \text{Entropy}(S_{Weak}) = 0.0$$

$$\text{Gain}(S_{Rain}, \text{Wind}) = 0.97$$

$$\text{Gain}(S_{Rain}, \text{Temp}) = 0.0192$$

$$\text{Gain}(S_{Rain}, \text{Humidity}) = 0.0192$$

$$\text{Gain}(S_{Rain}, \text{Wind}) = 0.97$$



$\{D_1, D_2, \dots, D_{14}\}$
[9+, 5-]

Outlook

Sunny Overcast Rain

Humidity

$\{D_3, D_7, D_{12}, D_{13}\}$

Wind

High Normal

yes

strong weak

$\{D_1, D_2, D_8\}$

No

$\{D_9, D_{11}\}$

yes

No

$\{D_6, D_{14}\}$

yes

$\{D_4, D_5, D_{10}\}$