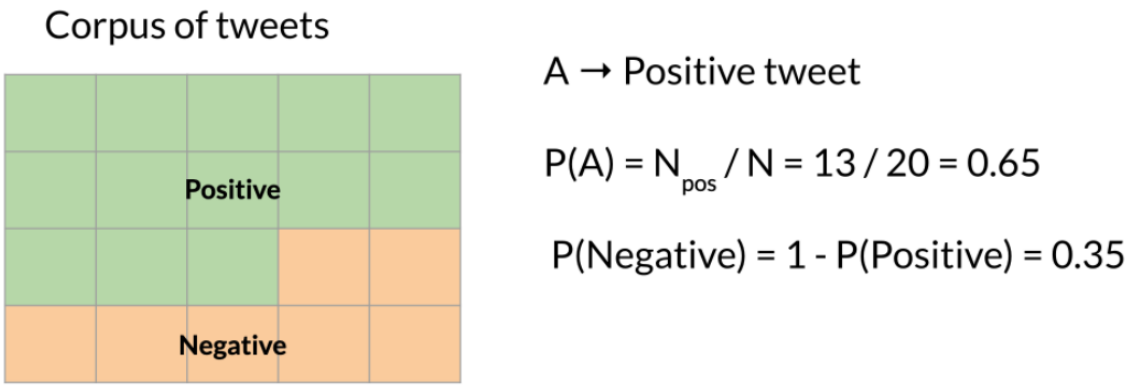
WEEK 2, NAÏVE BAYES

Probability and Bayes Rule

Probability : Pernah kan belajar probability ?



Inti Digambar diatas,ilustrasiin probabilitas :

P(A) → Probabilitas Tweet Positif

N = 20

Pos = 13

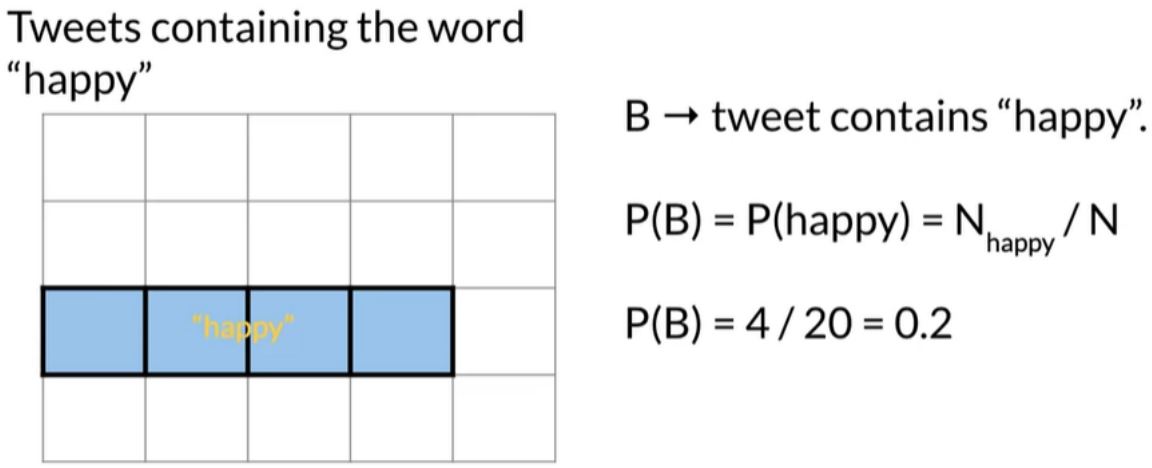
Neg = 7

P(A) = Total Positif / Total data =

13 / 20 = 0.65

P(~A) = 1 – P(A) = 1 – 0.65 = 0.35

Gimana kalau misalnya sekaarang, B → Tweet yang memiliki kata ‘happy’



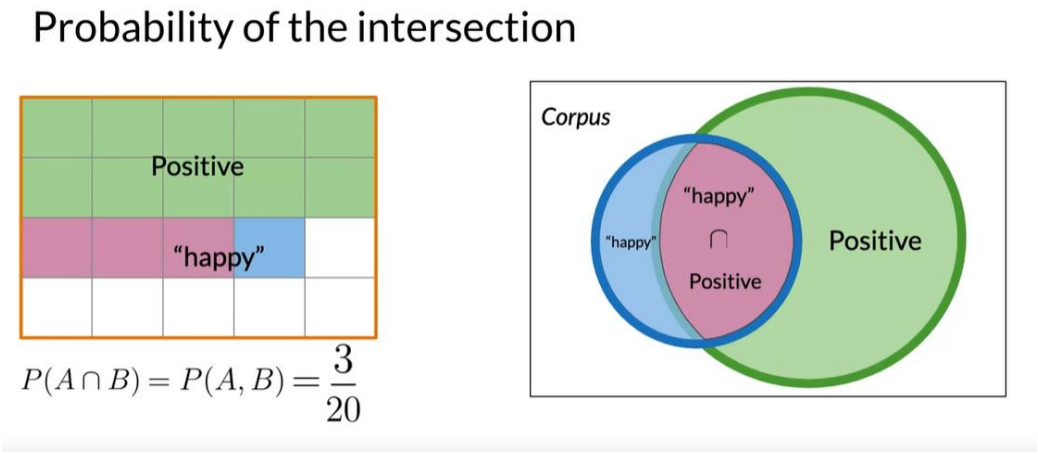
N_{Happy} = Total Tweets yang memiliki kata Happy didalamnya

Misal dari contoh tadi kita Cuma punya 4 Tweets yang memiliki kata happy

P (happy) = N_{Happy} / TotalTweets

P(happy) = 4 / 20 = 0.2

Jadi probability kata happy dalam dataset tweet Cuma 0.2/ 20% saja



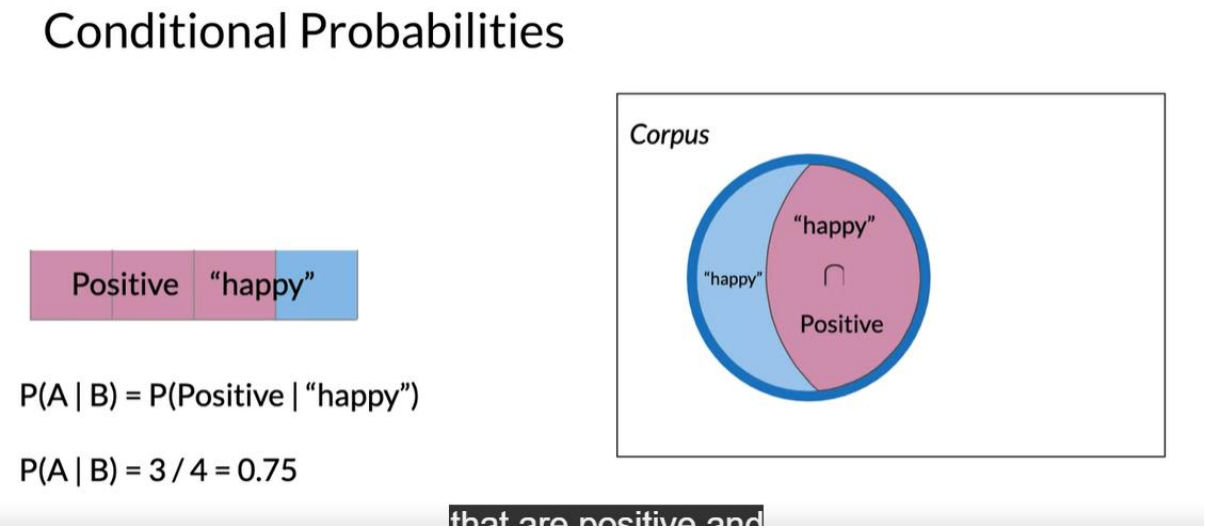
Positif , Happy= 3

Negatif,happy = 1

So Ini apa ? gini, di dataset ini kan dimisalkan ada positif dan negatif, nah missal kita ingin ngitung, Probabilitas Tweet positif yang mengandung kata Happy, P (A Iris B) = 3 / 20.

Itu doang, atleast paham lah probabilitas apaan

BAYES RULE



Aku kasih tau cara bacanya aja ya :

A = Positif

B = Happy

P(A|B) = Nilai Probabilitas (A) apabila diberikan nilai(B)

P(positif | Happy) = Nilai Probabilitas (positif) apabila diberikan kata (Happy) didalam inputan

Dari rumus diatas kita bisa mendapatkan bahwasanya Berapa probabilitas tweet positif yang mengandung kata happy)

Maka dari itu cukup dengan = 3 / 4 = 0.75 , Jadi probabilitas sebuah tweet diklasifikasikan positif apabila dia mengandung kata happy sebesar 75%. PAHAM Ndak ?

Kalau misalnya gini, gimans ?

P(Happy | positif) = Jadi ini bacanya probabilitas tweet tersebut mengandung kata happy apabila inputan memiliki kelas positif. Nah kita hitung P(Happy | positive) Kita tahu bahwasanya terdapat 3 tweet dengan isian kata happy didalamnya dan kelasnya positif, sehingga kita bisa nulis,

3 / 13 = 0.231, Jadi 23% kemungkinan terdapat kata happy, apabila tweet tersebut memiliki kelas positif 😊, oh ya 13 itu total positif ya.

Dan Conditional Probabilities ini disebut sebagai → Baye's RULE

So rumus Bayes Rule itu kek mans ?

P(X | Y) = P (Y| X) * (P(X) / P(Y)) , Sebenernya sama aja kek conditional probability, so pake aja itu kalau mau .

Misal nih kita pengen tau

P("happy" | Positive) = P ("happy" **iris** Positive) / P (positive)

= 3 / 13

=0.23

P(positive | "happy") = P (positive **iris** "happy") / P ('happy')

= 3 / 4

= 0.75

P(Positive | "happy") = P ("happy" **|** Positive) * P(Positive) / P("happy")

= (3/13) * 13 / 4

= 0.23 * 13 / 4

= 0.75

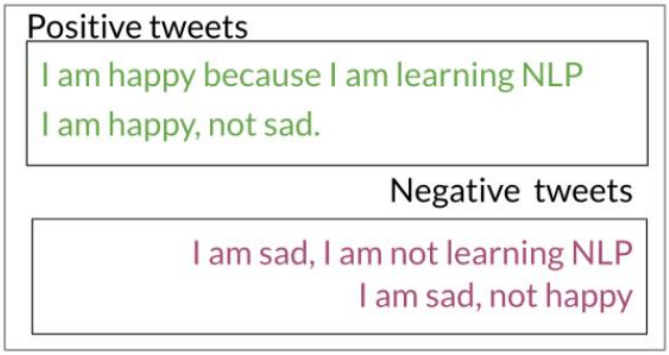
P("happy" | Positive) = P (Positive **|** "happy") * P("happy") / P (Positive)

= 0.75 * 4 / 13

= 0.231 < NJir sama >

So Naïve bayes ini gunain Conditional probability , atau Bayes rule

Kalau di week satu ini sebenernya yang diminta buat model yang sama tapi gunain naïve bayes



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_{class}	13	12

Itu si N_{class} Negatif totalnya harusnya 13 ya, tapi biar sesuai video pakenya yang 12 aja ya

Oke, dari sini sebenernya kita dah tau mau ngapain, inget Naïve Bayes itu menggunakan conditional probability, Jadi kita sekarang ngitung masing masing kata itu probabilitas di positif maupun negatifnya berapa.

Ngitungnya kek gini ,contohne gini aja ya

P (“I” | Pos) = P (I **iris** Pos) / P (POS) = 3 / 13 = 0.24, DST

Cara bacanya , Probabilitas Kata “I” apabila dia berasal dari tweet positif. Kita mengetahui dari contoh ini 24% kemungkinan tweet tersebut positif apabila terdapat kata “I” 😊.

word	Pos	Neg
I	0.24	0.25
am	0.24	0.25
happy	0.15	0.08
because	0.08	0
learning	0.08	0.08
NLP	0.08	0.08
sad	0.08	0.17
not	0.08	0.17

Oh ya, total dari pada probabilitas positif, maupun negatif harus 1 ya.

Mungkin ada yang bertanya, itu kalau misalnya probabilitas antar kata di kelas positif maupun negatif sama, kayak gimana tuh ? Artinya untuk saat ini **kata kata tersebut tidak memberikan kontribusi terhadap sentiment yang dihasilkan**. Bisa coba lihat kalimat **Sad** dan **not** , keliatan kan kontribusinya bakalan lebih besar ke arah negatif,dibandingkan positif. Sama halnya dengan kata **Happy**, yang lebih mencerminkan tweet menjadi positif, Benar ?

Nah ada hal yang ganjil gak ? Ada dong, Keliatan kan setiap kata itu punya kontribusinya masing masing, So **Hubungan antar kata** dipertimbangkan gak ? **NO !** Dengan demikian ini Naïve bayes Mengasumsikan **Bahwa setiap kata dalam tweets, independent. Then, Ya gada hubungan ga bisa mencerminkan real world dong !.**

But, ada permasalahan lainnya 😊 itu klean liat 0 Gak ? nah 0 ini bakalan problem buat kalkulasi nanti, so kita harus pecahin masalah ini.

Nah ini Naïve Bayes formula untuk **Binary classification, Sebenernya ini turunan dari bayes rule, yowes gw noob di mathematics,intinya ini ngitung kemungkinan sebuah tweet positif atau negatif.**

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

Itu yang symbol Π maksudnya perkalian , Kan kalau ∑ Sum. So Misal nih kita punya kalimat

“I Am Happy, because Learning”, maka dengan rumus diatas dan tabel yang udah kita buat, bisa kita kalkulasikan sebagai berikut.

(0.24 / 0.25) * (0.24 / 0.25) * (0.15 / 0.08) * **(0.08 / 0)** = **LOL**

Nah kan dah tau masalane di mana kalau apapun dibagi 0. Ya Ndak bisa , Reason ? Teangan we lah

So, Cara ngatasinya adalah dengan **Laplacian Smoothing**.

Kalau ngeliat yang week 1 , kita ngitung probabilitas setiap kata pake rumus ini

$$P\left(\mathrm{w_i} \mid \text{ class }\right) = \frac{\text{freq }(\mathrm{w_i}, \text{class })}{\mathrm{N_{class}}} \quad \text{class } \in \{ \text{ Positive, Negative } \}$$

Dengan Laplacian smoothing,rumusnya jadi kek gini

$$P\left(\mathrm{w_i} \mid \text{class}\right) = \frac{\text{freq}(\mathrm{w_i}, \text{class })+1}{(\mathrm{N_{class}} +\mathrm{V})}$$

Oke, gw jelasin kenapa ini bisa ngatasin masalah yang 0 tadi.

Liat yang bagian atas freq(wi,class) + 1, artinya berapun nilai yang dihasilkan dari kalkulasi freq(wi,class), akan ditambahkan 1, lah kalau yang bawah gimana ?Gw kasih contoh

Misal V = 8 yak

Kata **Because** itu tidak ada di negatif tweets ,dengan laplacian jadinya ya kegini

P (“Prim” | Positive) = 1 / 12 + TotalUniqueWords

Loh kok harus ada **V ?** missal nih missal,

Aku buat tabel kek dibawah ini, Pake + 1 , tapi ngga ditambahin **V dibagian pembaginya, pakenya yang Positif aja yak**

Word	Pos
I	0.30
Am	0.30
Happy	0.23
Because	0.15
Learning	0.15
NLP	0.15
Sad	0.15
Not	0.15
TOTAL	➢ 1.xx

Nah lho, keliatan kan ? Hasil sum probabilitasnya ga 1.0 , maka dari itu perlu dibenerin, Tau akar masalahnya dimana ? **Setiap kali ada unique word ,itu ditambah 1 kan ? Harusnya dia ikut di pembagiannya, kenapa? COBA liat kita ngebagiin hanya dengan total frekuensi kata kan? DAN ITU TIDAK MEREPRSENTASIKAN si +1 . maka dari itu tot frekuensi kata setelah kita tambah 1 disetiap unique word pasti hasilnya akhir total probabilitasnya akan 1.0, makanya perlu penggunaan V itu. NIh gambarne**

word	Pos	Neg
I	0.19	0.20
am	0.19	0.20
happy	0.14	0.10
because	0.10	0.05
learning	0.10	0.10
NLP	0.10	0.10
sad	0.10	0.15
not	0.10	0.15
Sum	1	1

Ratio of probabilities

word	Pos	Neg	ratio
I	0.20	0.20	1
am	0.20	0.20	1
happy	0.14	0.10	1.4
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
not	0.10	0.15	0.6

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

So disini ada ratio of probabilities, Rasio adalah definisi seberapa besar sesuatu dibandingkan dengan sesuatu yang lain (dalam kuantitas, angka, nilai). Jadi ini rasio kata “I” pada kelas positif maupun negatif besarnya adalah 1, dalam artian , pada dataset ini frekuensi kata “I” totalnya sama pada setiap kelas . Full Rumus Naïve Bayes Dari Deeplearning.ai , ini mah Turunan.

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

Jyang kotak biru itu yang hasil seluruh probabilitas kata ,baik di kelas positif ataupun dikeslas negatif pada **satu** observasi **Tweet,disebutnya Likelihood.** Nah itu yang Kotak warna Ungu, fungsinya untuk ngambil rasio probabilitas antara kelas positif dengan negatif,disebutnya **prior ratio**.

Misal nih, kita punya P(Pos) = 0.50 , dan p(Neg) =0.50 RAsionya jadi 1:1 kan ? Nah ini kasus nya balanced, kalau ndak balanced ya ga akan jad 1

Missal kita punya P(pos) = 0.75 dan p(Neg) = 0.25 Rasionya ? 3 : 1, jadi disetiap 4 tweet ada 3 positif dan 1 negatif, Tentu harus diperhatikan masalah imbalanced ini, karena kalau misalnya besar tentu nanti hasilnya malah condong ke salah satu kelas,

Oke sekatang kita masuk kedalam – dalamnya lagi, Liat Rumus ini

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

Apa yang ada dipikiran ? Yep Perkalian, dan ini bisa > 1, Selama hasilnya dapat >1 , kita ga tau range nya dong , ye gak ? Nah bisa aja hasil prductnya > 100, dengan begitu kita harus transformasi dia, disini gunain Log transformation . Nah rumusne jadinya gini

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

$$> \quad log\frac{P(pos)}{P(neg)}+\sum_{i=1}^{n}log\frac{P(w_i|pos)}{P(w_i|neg)}$$

Kok yang perkalian jadi pertambahan ? MANGGA LIAT RUMUS RUMUS LOG

Nah sekarang kita pengen ngitung tuh log ratio setiap kata, tabel ini ngga sesuai ya sama yang asli, contoh doang

word	Pos	Neg
I	0.05	0.05
am	0.04	0.04
happy	0.09	0.01
because	0.01	0.01
learning	0.03	0.01
NLP	0.02	0.02
sad	0.01	0.09
not	0.02	0.03

Sebelumnya kita tidak menggunakan logaritma kan pada saat mengkalkulasi ratio, nah dengan menggunakan log,ya kita bisa buat symbol baru lah, Lambda

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

Log Likelihood, Part 1

Summing the Lambdas

doc: I am happy because I am learning.

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

$$\lambda(\text{happy}) = log\frac{0.09}{0.01} \approx 2.2$$

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

Nah sekarang kita bisa kalulasi total dari pada lambda, i.e log(ratio) nya :

Log Likelihood

doc: I am happy because I am learning.

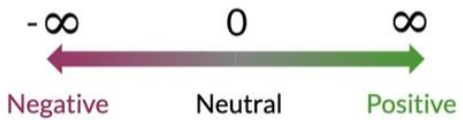
$$\sum_{i=1}^{m} log\frac{P(w_i|pos)}{P(w_i|neg)} = \sum_{i=1}^{m} \lambda(w_i)$$

log likelihood = 0 + 0 + 2.2 + 0 + 0 + 0 + 1.1 = **3.3**

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

Nah, kalu di sum ternyata totalnya 3.3, mengindikasikan bahwa kalimat tersebut masuk kedalam kelas positif, dibawah ini adalh range value untuk kelasnya ya. Kok ada Netral ? Well ini Cuma indikasi bahwasanya kalimat yang akan dilakukan prediksi memiliki kata kata yang tidak mengindikasikan dia ke kelas positif ataupun negatif, solusi ?well, kalau dapet 0 bisa dibuat kelas baru ataupun ignore saja tapi tetp liat awal mula kalimat nya seperti apa, mungkin ada informasi yang dihilangkan, Nanti kita point masalh tersebut di error analysis

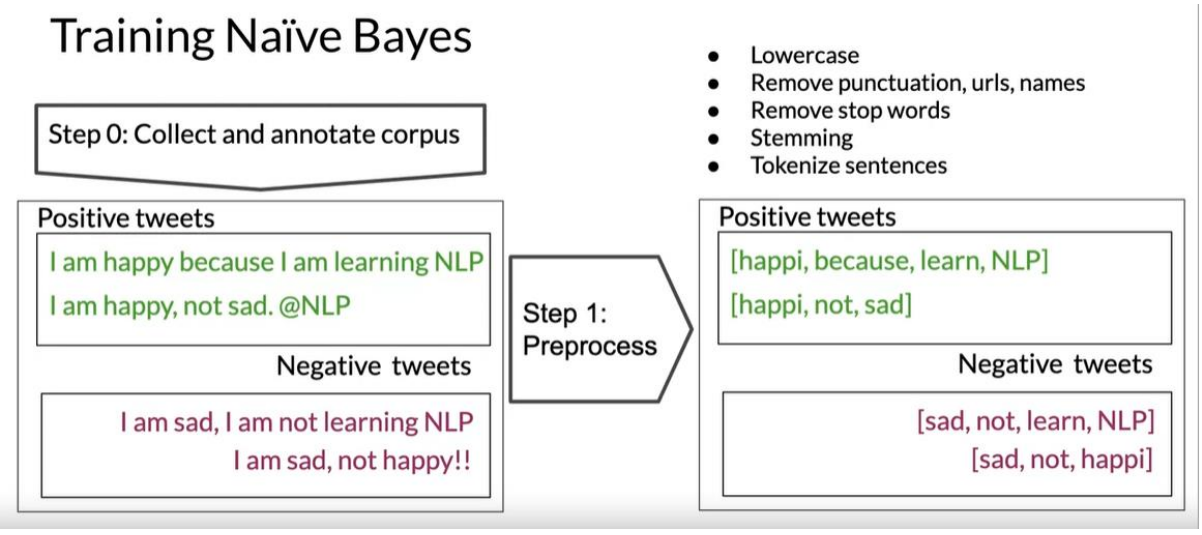
$$\sum_{i=1}^{m} log \frac{P(w_i|pos)}{P(w_i|neg)} > 0$$



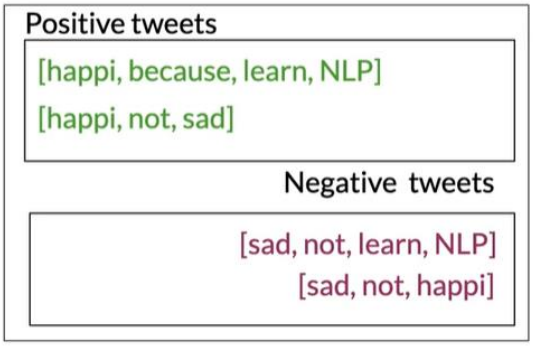
Pipeline Naïve Bayes yang dibentuk kek gini :

0. Get or annotate a dataset with positive and negative tweets
1. Preprocess the tweets: process_tweet(tweet) → [w_1, w_2, w_3, ...]
2. Compute freq(w, class)
3. Get P(w | pos), P(w | neg)
4. Get λ(w)
5. Compute logprior = log(P(pos) / P(neg))

Ndak ada Gradient descent beneran menggunakan **Frekuensi tabel, Ini secara detailnya**



Training Naïve Bayes



Step 2: Word count

freq(w, class)

word	Pos	Neg
happi	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

Training Naïve Bayes

freq(w, class)

word	Pos	Neg
happi	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

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

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

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

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

Step 4: Get lambda

word	Pos	Neg	λ
happy	0.23	0.15	0.43
because	0.15	0.07	0.6
learning	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

Training Naïve Bayes

Step 5: Get the log prior

D_{pos} = Number of positive tweets
D_{neg} = Number of negative tweets

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

If dataset is balanced, $D_{pos} = D_{neg}$ and logprior = 0.

Bingung gak tuh? Logprior apaan ? y aitu namalain dari pada priorratio aja, oh ya dan kalau misalnya datanya **balanced** otomatis logpriornya 0 ya bukan 1 :v, log(1) berapa ?

Nah ssekarang ngelakuin prediksi gunain model yang udah dibuat pake rumus ini

$$> \quad log\frac{P(pos)}{P(neg)} + \sum_{i=1}^{n} log\frac{P(w_i|pos)}{P(w_i|neg)}$$

Predict using Naïve Bayes

- log-likelihood dictionary $\lambda(w) = log\frac{P(w|pos)}{P(w|neg)}$
- $logprior = log\frac{D_{pos}}{D_{neg}} = 0$
- Tweet: [I, pass, the, NLP, interview]

$$score = -0.01 + 0.5 - 0.01 + 0 + logprior = 0.48$$

$$pred = score > 0$$

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

Itu kan kata **Interview** ngga ada di table frequencies, so treated as **neutral word, 0.**

Nah itu aja sih yang dibuat,selanjutnya ya cuman informasi aja terkait implementasi Naïve bayes

Naïve Bayes Applications

- Sentiment analysis
- Author identification
- Information retrieval
- Word disambiguation
- Simple, fast and robust!

Naïve Bayes bisa banyak diterapin, tapi y itu tadi banyak limitasi dan permasalahan kalau mau dipake, yaitu

Independence: Not true in NLP

Relative frequency of classes affect the model

Inget Imbalance dataset itu mostly affect ke setiap model machine learning . And Independence, contohnya gini

Misal ada kalimat

Sore ini agak panas, dan kering musim ___ beda (Panas, Gugur, Dingin, Semi) Si naïve bayes karena mengasumsikan independent ya, probabilitas jawabannya bisa aja hampir sama

ERROR ANALYSIS di Naïve Bayes Classfier dan bisa berlaku ke algoritma ML lain

- Removing punctuation and stop words
- Word order
- Adversarial attacks

Ini contoh contohnya

Processing as a Source of Errors: Punctuation

Tweet: My beloved grandmother ☓

processed_tweet: [belov, grandmoth]

Hasil preprocessing malah jadi Positf dia, nah ini kenapa punctuation begitu penting untuk dipertimbangkan terhadap penghapusan.

Processing as a Source of Errors: Removing Words

Tweet: This is not good, because your attitude is not even close to being nice.

processed_tweet: [good, attitude, close, nice]

Processing as a Source of Errors: Word Order

Yowes dah lah ya

Adversarial attacks

Sarcasm, Irony and Euphemisms

Tweet: This is a ridiculously powerful movie. The plot was gripping and I cried right through until the ending!

processed_tweet: [ridicul, power, movi, plot, grip, cry, end]