Text Classification with ‘Logistic Regression’

(Maximum Entropy Model)

How Make a good classifier

Domain : The data must match the real scenario.

data ต้องอิงสถานการณ์จริง

Data quality and quantity : The annotation must be consistent, and the size must be large .

ข้อมูลต้องมีคุณภาพดีและใหญ่

Feature engineering : What does the model need to pay attention in order to make a good decision ?

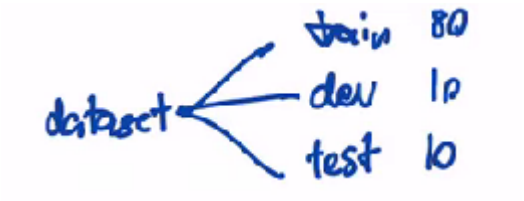
ต้องคิดหาฟีเจอร์ที่เหมาะสมกับงาน

Model : Some models are better than others.

Logistic Regression

Logistic regression (or Maximum Entropy Model) is a statistical model that computes (PY|X) from linear combination of input features. It models a link between each label and each feature (infavor or against).
มันเป็นการใช้สถิติของคำ เพื่อคำนวณ *(PY|X) x = feture vector y = label*
เป็น การหาผลรวม ++ เพื่อให้เรารู้และหาความเชื่อมโยง ระหว่างฟีเจอร์และเรเบล

if Y is multiclass (e.g. positive, neutral, and neagative sentiment), logistic regression should be called ‘multinomial logistic regression. In NLP , we call it gogistic regression or MaxEnt.
ถ้า *Y* เป็นมัลติคลาส เช่น เชิงบวกลบ เฉยๆ เราจะเรียกว่า
multinomial logistic regression ส่วนใหญ่ก็จะเป็นมัลตินออเมียล



Components of ML classifier ส่วนประกอบ ของ ML

1. **Representation** - How do we convert from text to feature vector ?
   หาตัวแทน เพราะข้อความไม่สามารถนำไปคำนวณได้ *ex bagOfWord*
2. Inference/prediction - How do we compute P(Y|X)?
3. Training
   a. Objective function
   b. Optimization algorithm for training

Representation

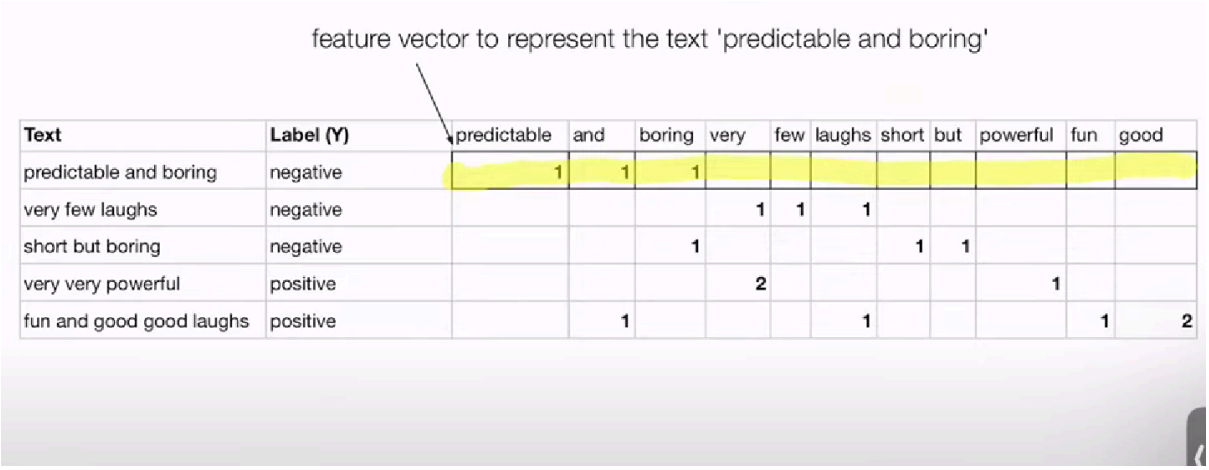
A feature vector represents text. A vector is a list of numbers that can be compared with another vector to measure similarity.

- ➕ Unigram count feature = bag-of-word features
- ➕ Unigram binary feature
- ➕ Unigram TF-IDF feature
- ➕ Bigram count feature

text1 => feature vector

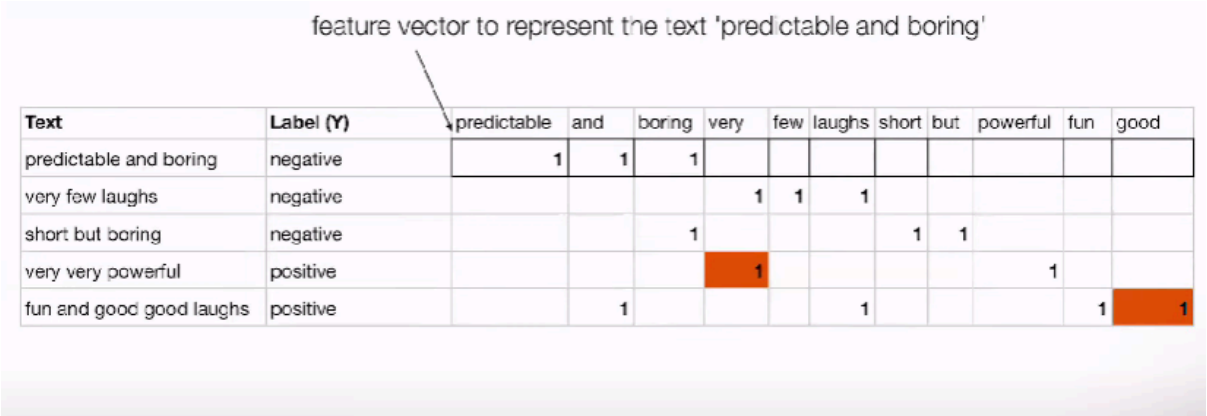
A	B	C	D	E
1	0	1	1	0

text ตัวต่อไปก็ต้องมีแพทเทิลเดียวกัน คือ คอลั่มต้องทั่วถึง ทุกแถว



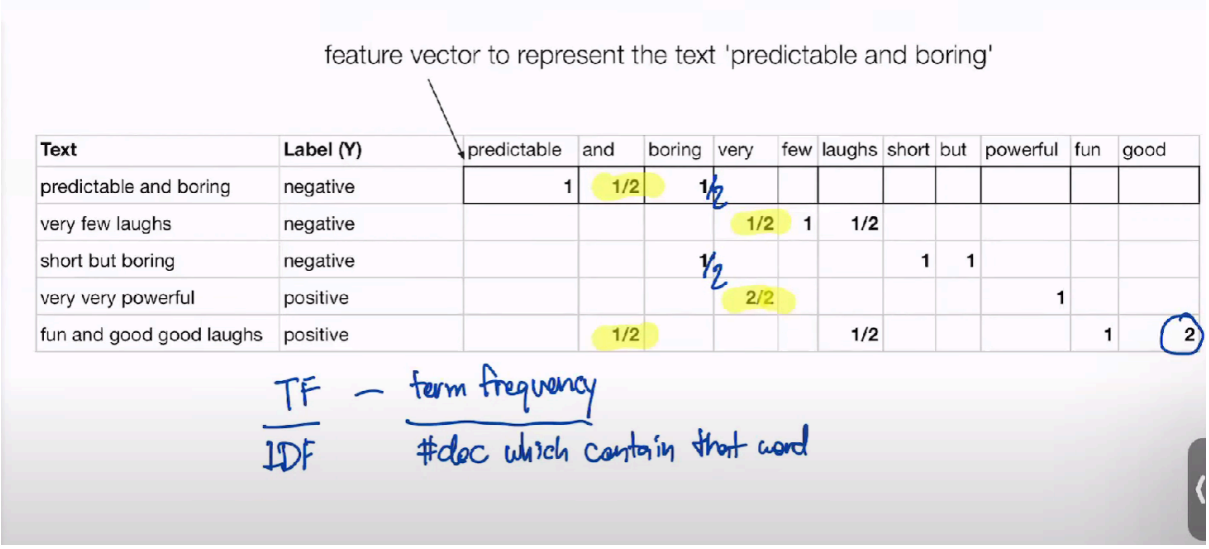
Text	Label (Y)	predictable	and	boring	very	few	laughs	short	but	powerful	fun	good
predictable and boring	negative	1	1	1								
very few laughs	negative				1	1	1					
short but boring	negative			1				1	1			
very very powerful	positive				2					1		
fun and good good laughs	positive		1				1				1	2

Uni = 1 gram = word = นับทีละคำ such as bag-of-word feture



Text	Label (Y)	predictable	and	boring	very	few	laughs	short	but	powerful	fun	good
predictable and boring	negative	1	1	1								
very few laughs	negative				1	1	1					
short but boring	negative			1				1	1			
very very powerful	positive				1					1		
fun and good good laughs	positive		1				1				1	1

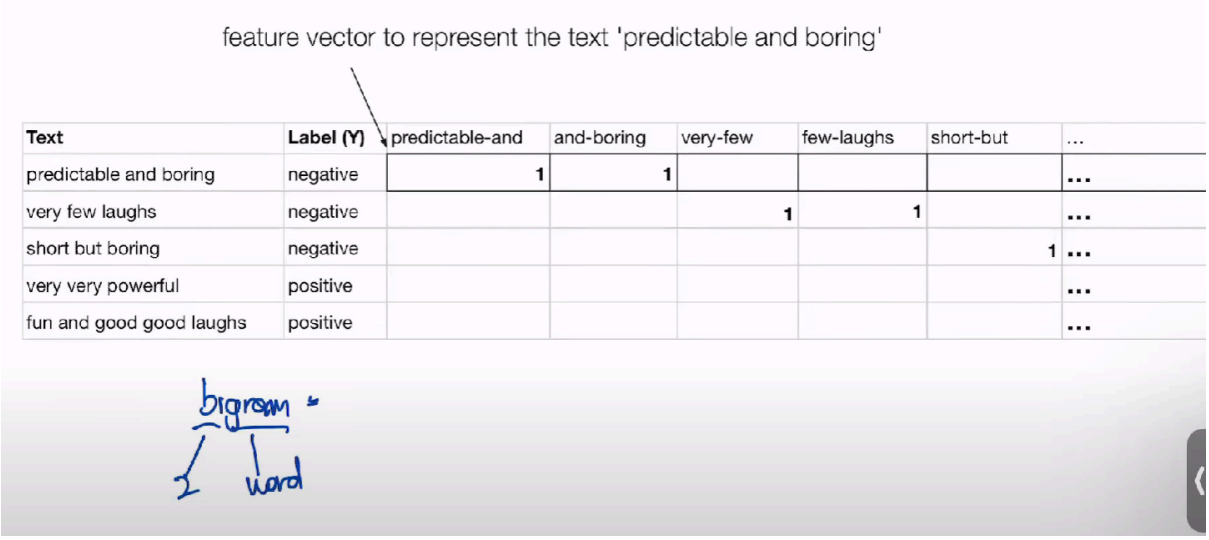
อันนี้แล้วแต่งาน ต้องพิจรณา คืออันนี้จะนับคำที่เจอบ่อยเป็น 1 อยู่ดี **Unigram binary**



Text	Label (Y)	predictable	and	boring	very	few	laughs	short	but	powerful	fun	good
predictable and boring	negative	1	1/2	1/2								
very few laughs	negative				1/2	1	1/2					
short but boring	negative			1/2				1	1			
very very powerful	positive				2/2					1		
fun and good good laughs	positive		1/2				1/2				1	2

TF-IDF = TF =term frquracy / IDF #doc which contain that word

หมายความว่า คำที่เจอบ่อยๆ อาจจะไม่ได้ช่วย หรือ สื่อความหมายถึง label เท่าไร



Biagram feature คือหาคำที่อยู่ติดๆกัน 2 คำ
เช่น very-few few-laughs

มันเป็นการโมเดลของเรา จะได้ฟีเจอร์ที่ยาวมาก เยอะ

Text Representation (Features)

1. Unigram count feature assumes each label has a list of associated keywords. this is very good baseline feature.
2. Unigram binary feature is like ungram count feature but ignores the effects of duplicate words.
3. Unigram TF-IDF feature is like unigram count feature but downweights some of the words that appear in too many documents.
4. Biagram count feature assumes that we must consider at least two adjacent word to be able to predict the label. This feature is always too sparse i.e. too many zeros in toe feature vector.
5. These features do work well if we have a good amont of data, but we will see more advanced representation called ‘word embedings.’ Stay tuned.

   ไม่มีอันไหนดีกว่าอันไหน ต้องลองเอา

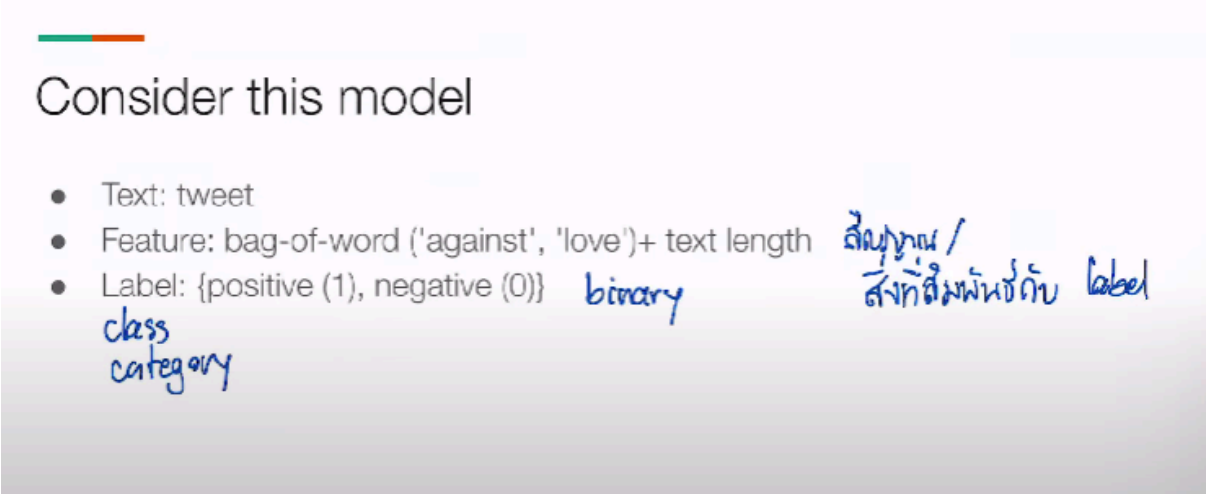
Inference/Prediction - How do we compute P(Y|X)?

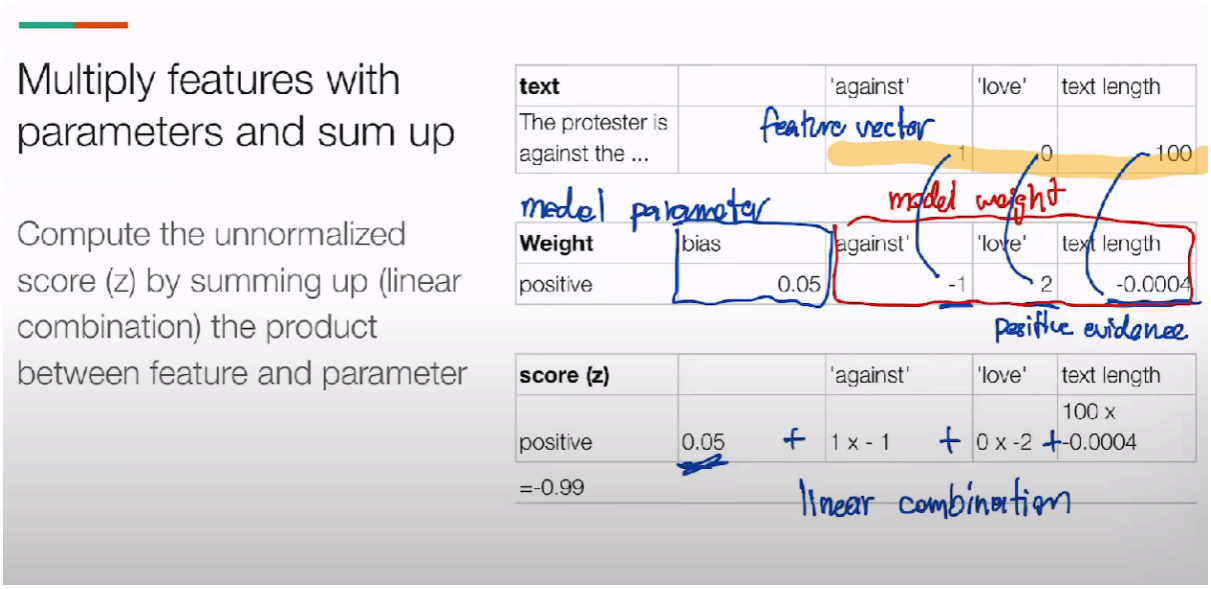
การคำนวณ หาความน่าจะเป็น

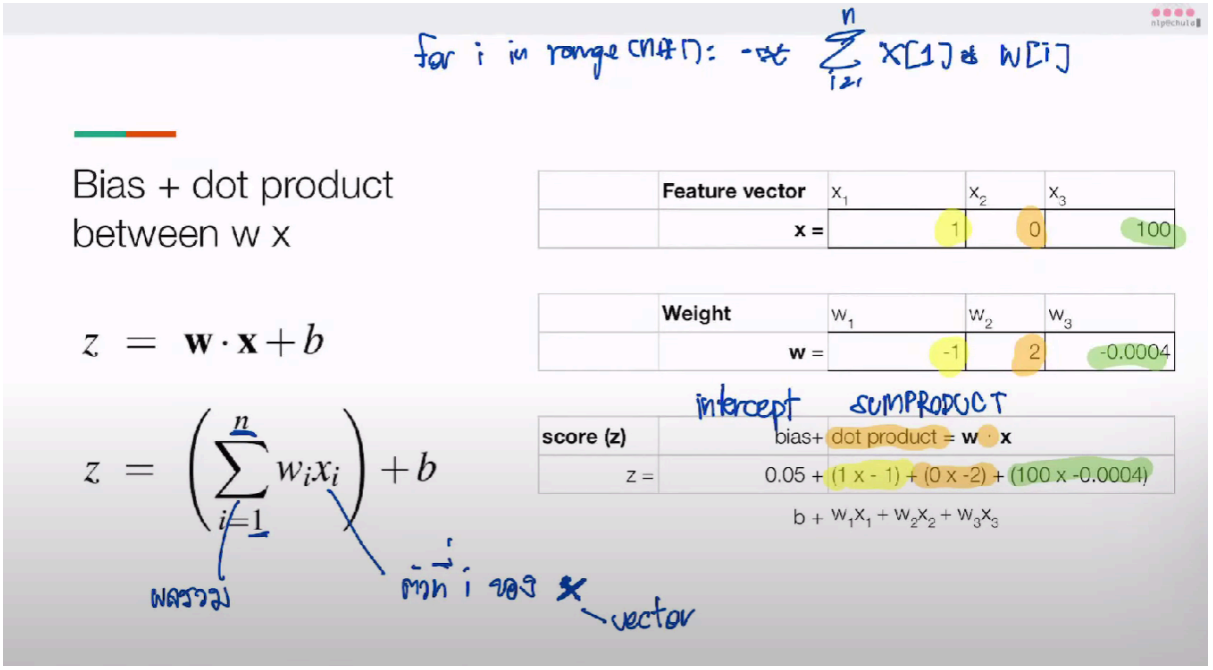
Logistic Regression - inference

Naive Bayes computes P(Y|X) by multiplying up P(x|Y=y) *model parameter form training (product of all features x for each label y)

Logistic regression compute P(Y|X) by using sigmoid function (if 2 classes) or softmax funcion (if > 2 classes)





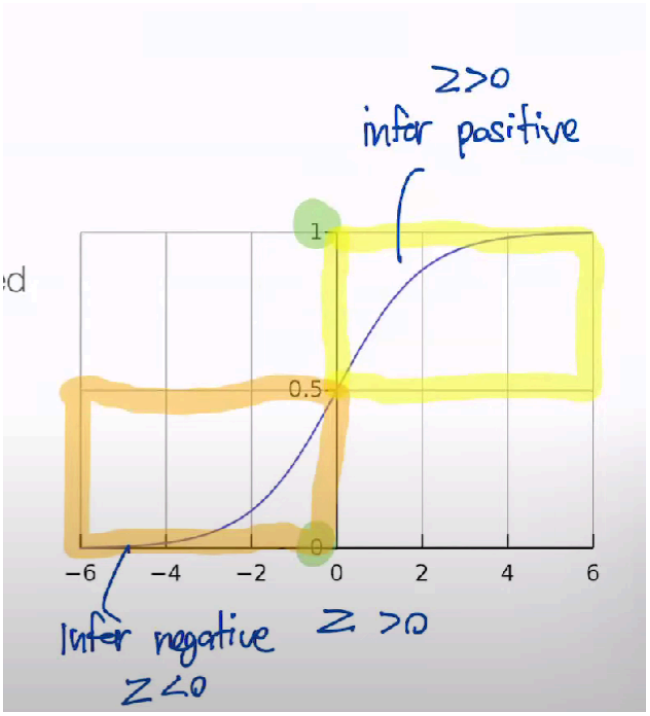


z = สิ่งที่จะนำไปแปลงเป็น probability

Sigmoid !!!

We convert z ito probability P(Y=1|X) by passing Z into a sigmoid function (also called logistic function)

e == 2.71828



Compute P(Y|X)

Using complementation, if Y is not 1, then Y must be 0.

P(Y=0) = 1 - P(Y=1)

$$P(y=1) = \sigma(\mathbf{w}\cdot\mathbf{x}+b) = \frac{1}{1+\exp(-(\mathbf{w}\cdot\mathbf{x}+b))}$$

$$P(y=0) = 1-\sigma(\mathbf{w}\cdot\mathbf{x}+b)$$

z

z

P(y=1)

อยากรู้ของ Nagative class ให้เอา 1 ไป ลบ

(Binary) Logistic Regression

- One instance of the text is represented by a feature vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$
- The logistic regression model has a bias term b (also called intercept term) and weight vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]$. b and $\boldsymbol{w}$ are the model parameters that need to be learned/trained from the training set.
- Bias b represents the score in favor of the positive label regardless of the feature vector.
- Each weight w_i represents the score in favor of the positive label associated with each feature x_i.
- z = unnormalized score computed by bias + the dot product of $\boldsymbol{w}$ and $\boldsymbol{x}$
- The probability of the positive label P(Y=1|X) is computed by passing z into a sigmoid function.

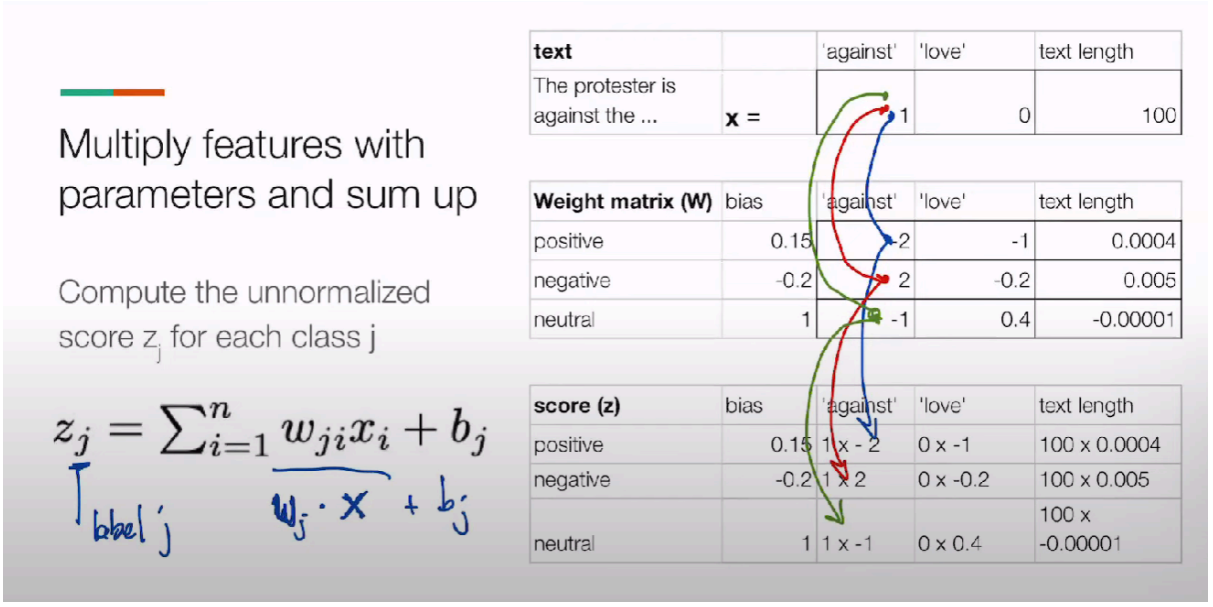
Multiclass Lg1aistic Regression

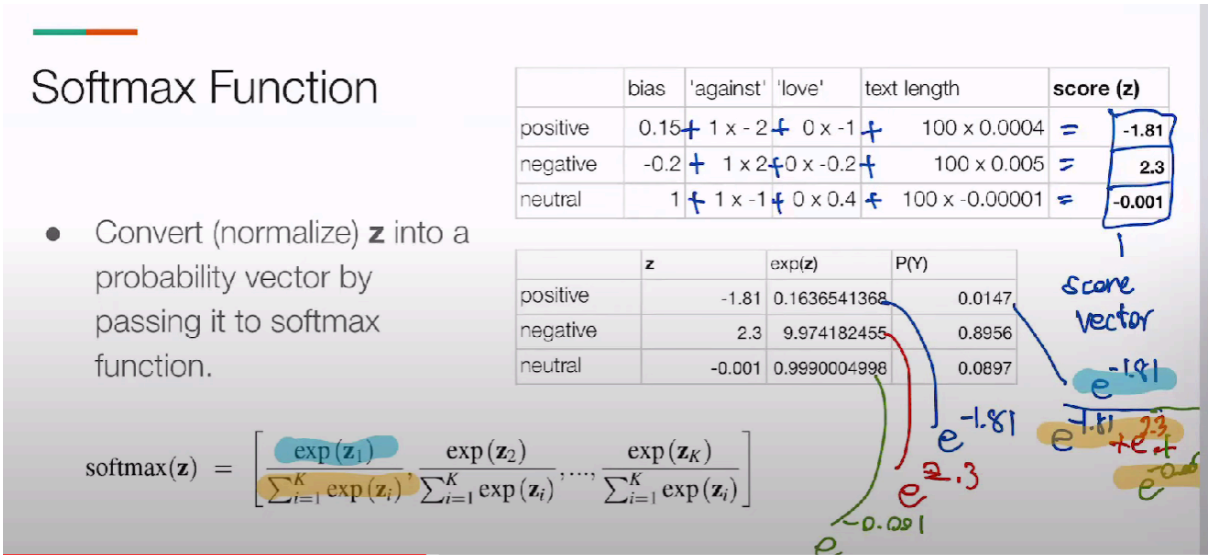
Multiclass logistic regression is just like binary logistic regression, but it supports the scenario where class are > 2.

- ส่วนใหญ่คลาสมากกว่า 2 อยู่แล้ว

The idea is the same but we have weight vector for each class . If there are 3 class , we have three weight vector . We concatenate these vectors and call them weight matrix .

- vector ต่อกันหลายๆกันเรียก matrix





P(Y) = normalize คือ แถวแรก ของ positive = e pow -1.81 / e pow 2.4+ e pow -0.001

Matrix Multiplication