

# Natural Language Processing with Python

Eddy

[eddyhu71@gmail.com](mailto:eddyhu71@gmail.com)

<https://github.com/EddyHu71>

# Bag of Words

Sentence 1 : I like learn programming

Sentence 2 : I hate learn programming

Sentence 3 : I like learn programming and design

	I	like	hate	learn	programming	and	design
Sen 1	1	1		1	1		
Sen 2	1		1	1	1		
Sen 3	1	1		1		1	1

# Tahukah Anda?



**Mendoakan orang mati maka  
Kita akan mendapat dosa**

Right place.

Right time.

Right mind.

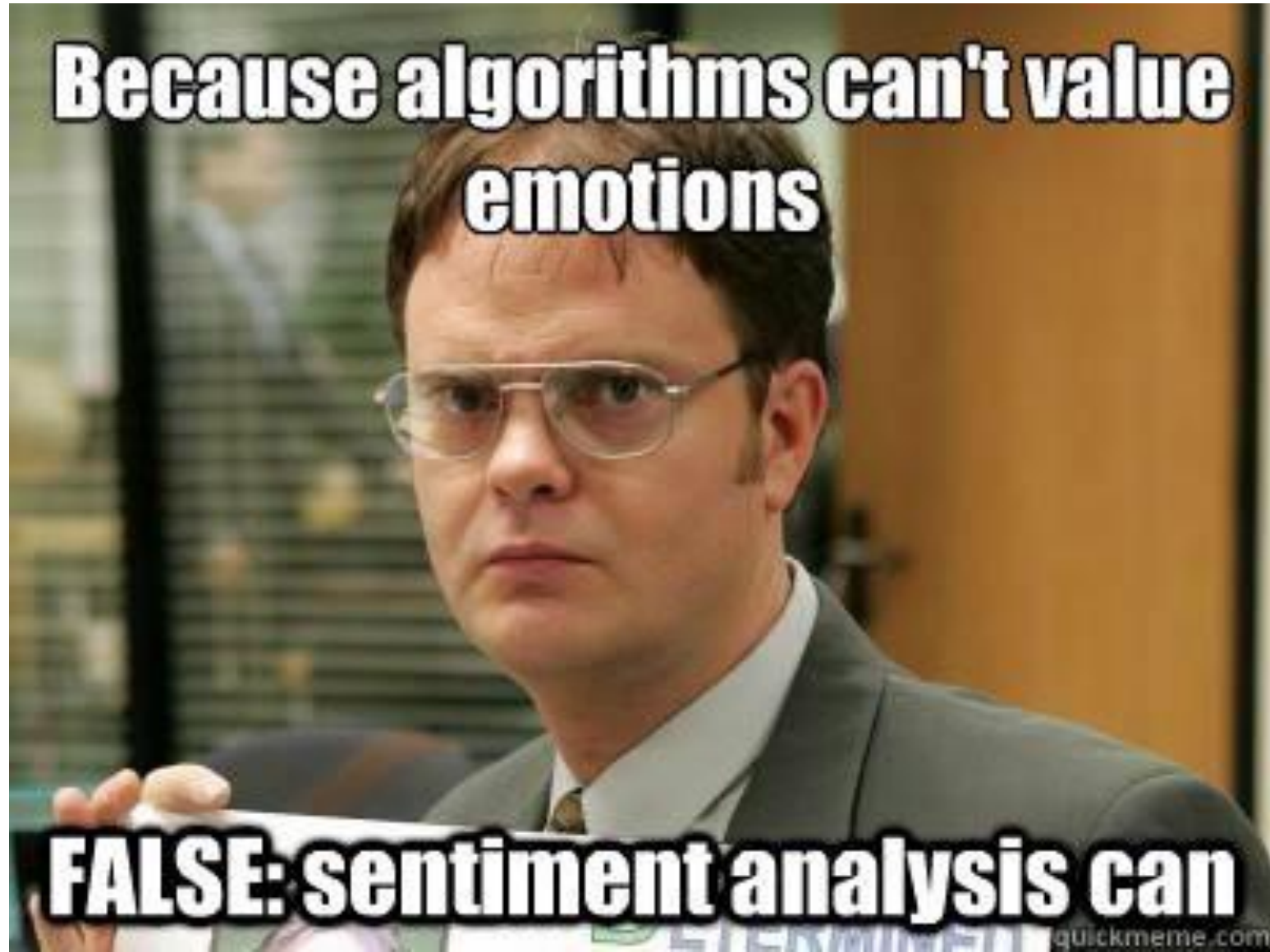
Right now.

Right direction...

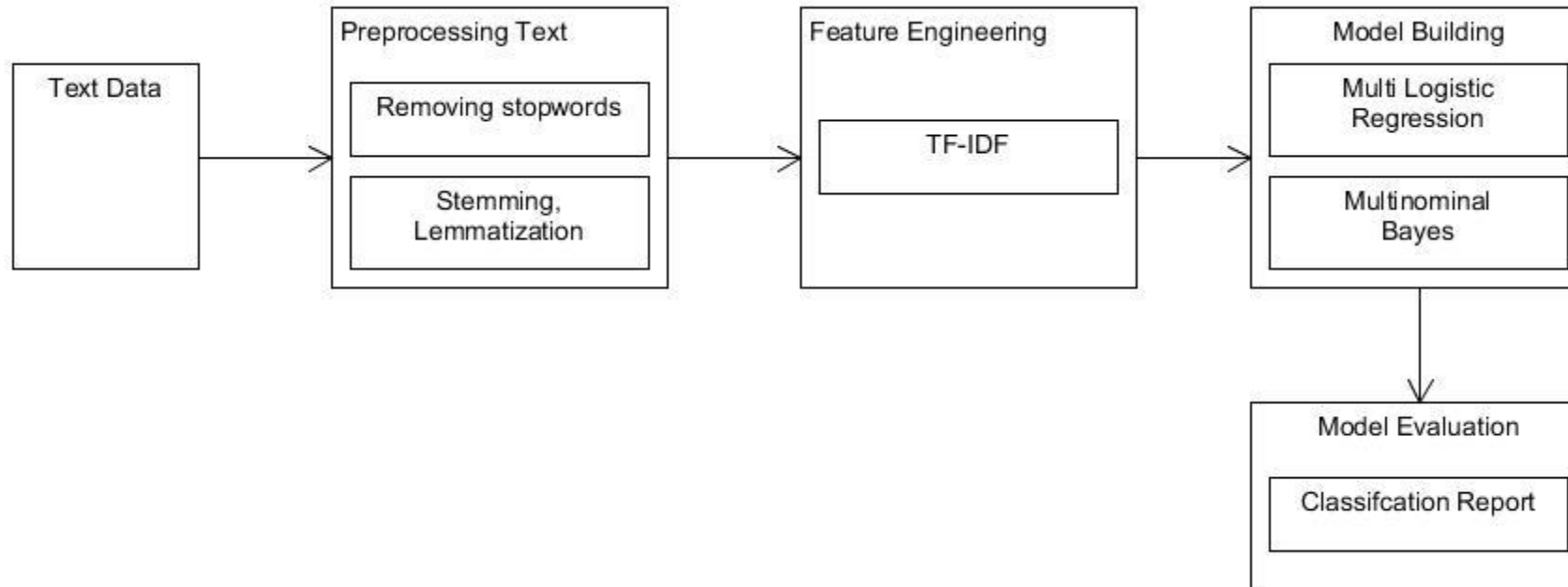


<https://t.co/UAX656VMRY>

# Sentiment Analysis



# Sentiment Analysis WorkFlow



# **Multinomial Logistic Regression**



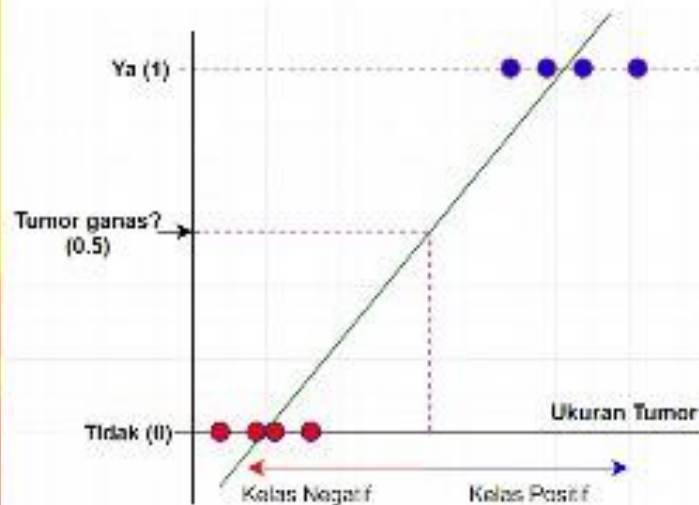
# Linear Regression

- The output is prediction.
- The output of data type usually continuous.

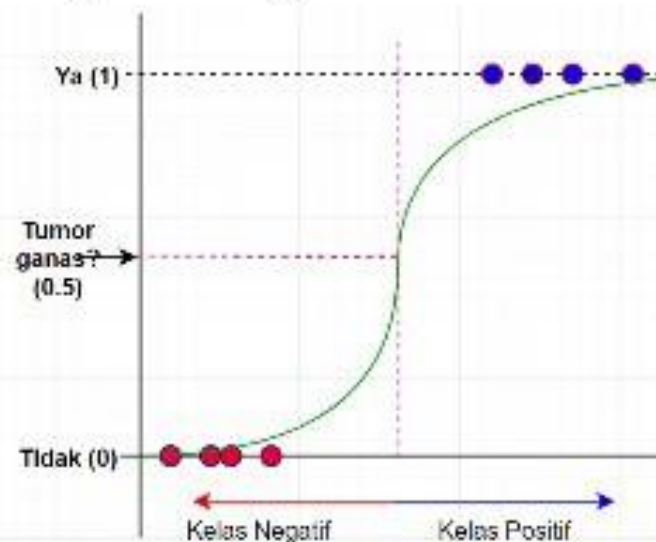
$$y = b_0x + b_1$$



## Linear Regression



## Logistic Regression



# Logistic Regression

- The output is classification.
- The output of data type usually discrete variable.
- For multinominal logistic regression, it's used for multi class classification. Logistic regression used for binary classification.

$$f(x) = \frac{e^{y_1 = b_0x + b_1}}{1 + e^{-y_1}} = \frac{1}{1 + e^{-(b_0x + b_1)}}$$

$$Y = b_0 + b_1 * X$$

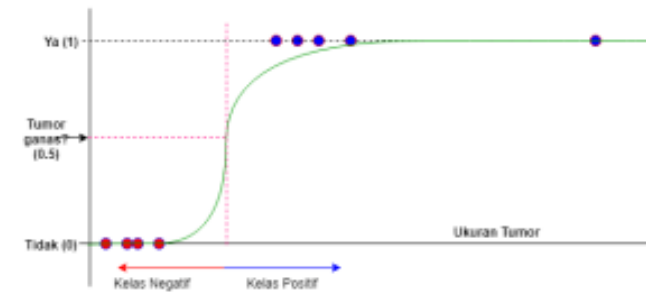
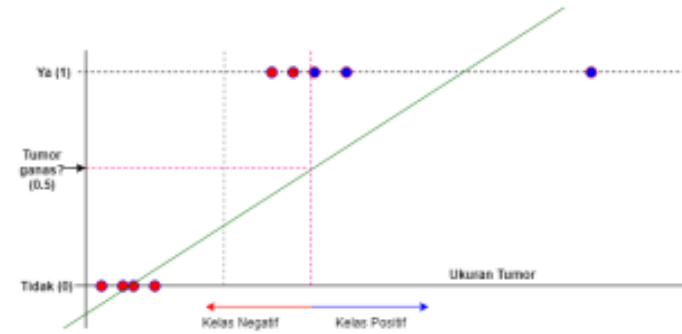
LINEAR  
FUNCTION

$$P = \frac{1}{1 + e^{-Y}}$$

SIGMOID  
FUNCTION

$$P = \frac{1}{1 + e^{-(b_0 + b_1 * X)}}$$

LOGISTIC  
FUNCTION



# **Multinomial Bayes**

# Naive Bayes

- The output is classification.
- The output of data type usually discrete.
- For multinominal, it's for multi-class classification.

$$P(L|features) = \frac{P(features|L)P(L)}{P(features)}$$

If you want to predict more than 2 labels:

$$\frac{P(L_1|features)}{P(L_2|features)} = \frac{P(L_1|features)P(L_1)}{P(L_2|features)P(L_2)}$$

# **Feature Engineering**

**TF-IDF**



# Term Frequency (TF)

A method to count the weight of word that used in a text / sentence.

$$tf = \begin{cases} 1 + \log_{10}(f_{t_1d}), & f_{t_1d} > 0 \\ 0, & f_{t_1d} = 0 \end{cases}$$

# Inverse Document Frequency (IDF)

How term distributed widely in document.

$$IDF_j = \log \frac{D}{df_1}$$

**So, what is TF-IDF?**

# TF-IDF

A method to count the weight of word that used in a text / sentence and term distributed widely in document.

$$w_{ij} = tf_{ij} \times idf_j$$

$$w_{ij} = tf_{ij} \times \log \frac{D}{df_j}$$

If  $D = df_j$  which can causes 0, so add 1 in IDF equation.

$$w_{ij} = tf_{ij} \times \log \frac{D}{df_j} + 1$$

**Let's code**