Modul 3

Text Mining

Data Science Program

Purwadhika
Startup and Coding School

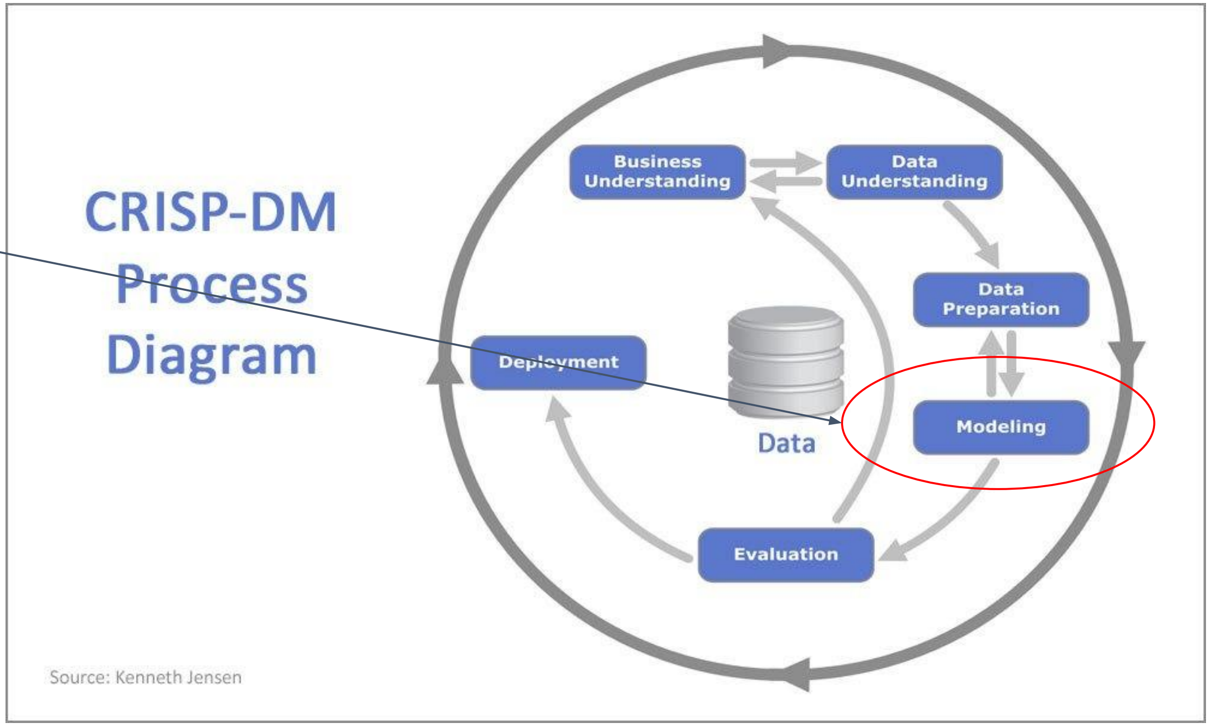
Outline

What Is Text Data and Text Mining?

Text Preprocessing

Text Exploration

Text Classification



Purwadhika
Startup and Coding School

What Is Text Data and Text Mining ?

Purwadhika
Startup and Coding School

What Is Text Data and Text Mining?

Text Data:

- i.g. if we want to classify an email message as either a legitimate email or spam,
- the content of the email will certainly contain important information for this classification task
- email is the text data

Text Mining :

- Text mining is the process of distilling actionable insight from text

label	message
ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
ham	U dun say so early hor... U c already then say...
ham	Nah I don't think he goes to usf, he lives around here though
...	...
spam	This is the 2nd time we have tried 2 contact u. U have won the £750 Pound prize. 2 claim is easy, call 087187272008 NOW1! Only 10p per minute. BT-national-rate.
ham	Will ü b going to esplanade fr home?
ham	Pity, * was in mood for that. So...any other suggestions?
ham	The guy did some bitching but I acted like i'd be interested in buying something else next week and he gave it to us for free
ham	Rofl. Its true to its name

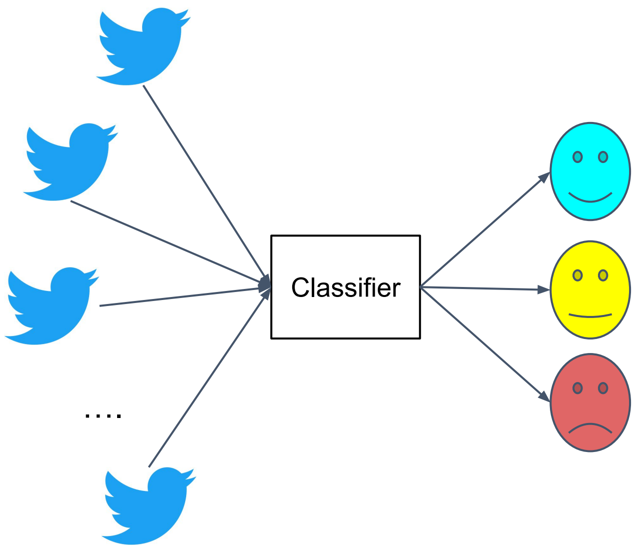
Corpus

Documents

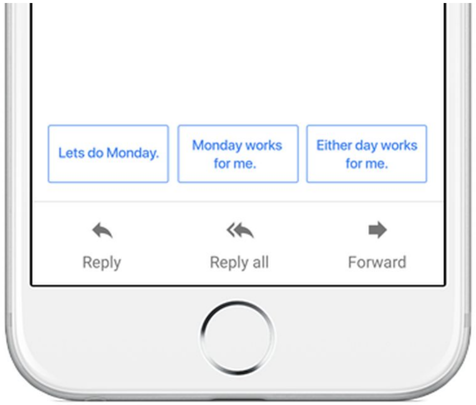
Text Data Source

Purwadhika
Startup and Coding School

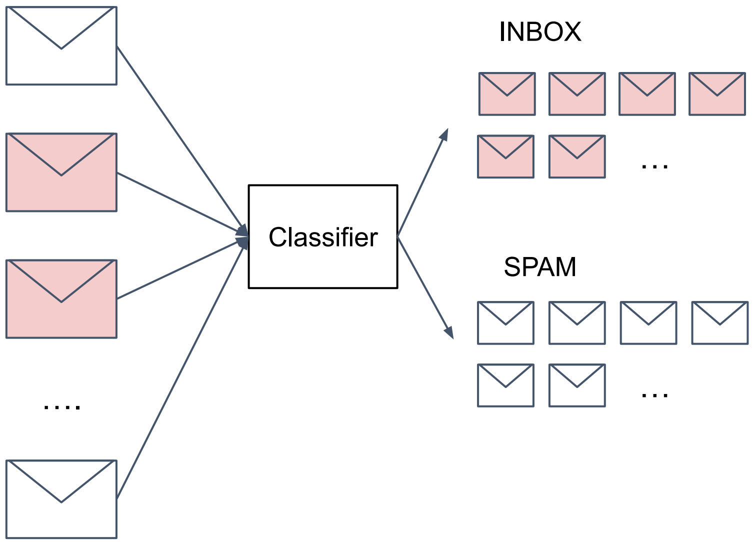
Application Of Text Data



Sentiment Analysis



Smart Reply



Document Classification

Example : Sentiment Analysis

A technique to extract information that contain perspective about certain issue

Text	Information Extracted
I love this food	Positive
This food is really awesome	Positive
This is one of the food that i could never bring myself to eat	Negative
I hardly know about this food	Neutral
There is no better food than this one	Positive
Everyone should try this food	Positive

Purwadhika
Startup and Coding School

Why Do We Care About Text Data ?

We already show you some use cases of text data utilization, spam detection, smart reply and sentiment analysis

- Social media growing, evolving, and affect an organization's public efforts
- Understand your customer better and faster
- The digitization of formerly paper records, such as feedback
- Online content from an organization, its competitors and outside sources, such as blogs, continues to grow
- etc

Text Preprocessing

Purwadhika
Startup and Coding School

Preprocessing

Text Preprocessing Part 1:

- converting to lowercase
- contraction
- remove or convert number into text
- remove punctuation, marks
- remove white spaces
- remove stop words and particular words

Text Preprocessing Part 2:

- Stemming
- Lemmatization
- Part of Speech tagging
- entity recognition
- Bag of words
- N-Grams

Document Term Matrix:

- Term Frequency (TF)
- Term Frequency - Inverse Document Frequency (TF-IDF)

Contraction

Contractions are shortened version of words or syllables, for example

- I’ve done it → I have done it
- I’m here → I am here
- You’re smart → You are smart

Stopwords

Stopwords are words that occur too often and do not provide any additional insight

Stopwords example:

I, me, myself, we, our, ourselves, you, yourself, …..

Stemming

The process of transforming to the root word

For example you have caring, cares, cared, caringly carefully then you want to consider them as the same words “care”

We need stemming because treating them as the same words will reduce overfitting

caring - cares - cared - caringly - carefully → care - care - care - care - care

Lemmatization

The process of transforming to the dictionary base form

caring, cares, cared, caringly carefully will be transformed into care, care, care caringly carefully

caringly and carefully are listed in the dictionary

purpose of lemmatization is also to reduce overfitting

caring - cares - cared - caringly - carefully → care - care - care - caringly - carefully

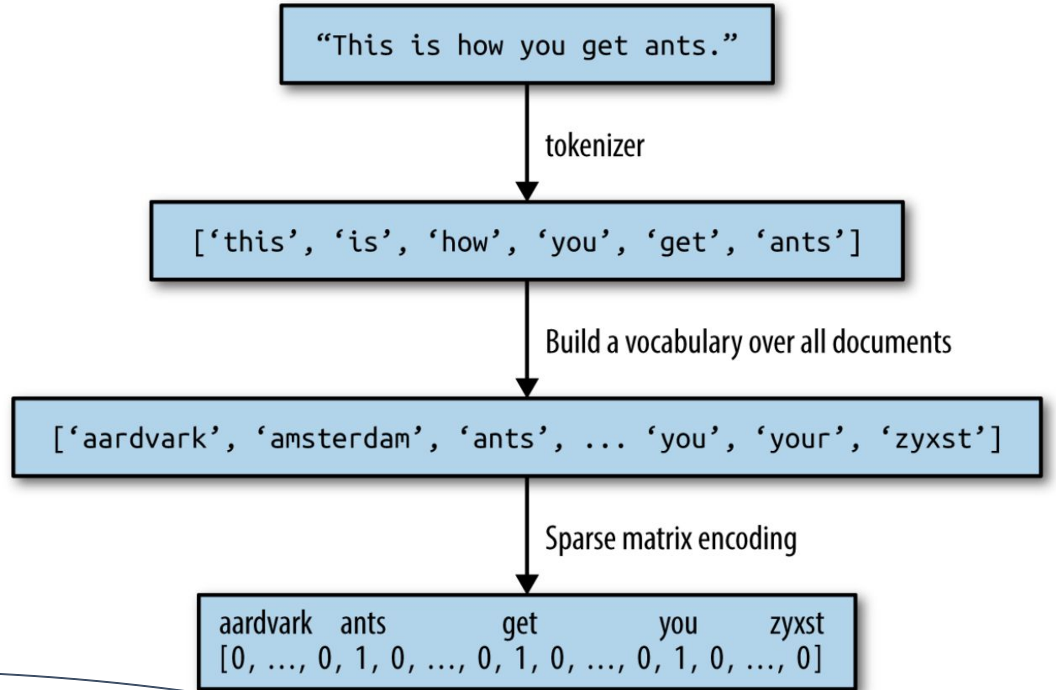
Bag of Words

- represent text for machine learning
- count how often each words appear in each text
- discarding many structure such as chapters, paragraphs, sentences and formatting

Step by step to represent text for machine learning

- tokenization, splitting each words
- vocabulary building, vocabulary over all documents
- encoding, count how often words show up

Bag of Words



	aardvark	ants	...	get	...	you	zyxst
D1	0	1	...	0	...	1	0
D2	0	0	...	1	...	1	0
D3	0	0	...	2	...	0	0
...	...	...	...	...	...	...	...

Document Term Matrix

Purwadhika Startup and Coding School

N-Grams

Bag of words ignore word order completely

These two sentences will have the same tokenization.

- It’s bad, not good at all
- It’s good, not bad at all

Method	Text
	It’s bad, not good at all → bad not good all
Tokenization	“bad” “not” “good” “all”
	It’s good, not bad at all → good not bad all
Tokenization	“bad” “not” “good” “all”

Machine learning will treat these sentences the same while the meaning is actually different

Purwadhika
Startup and Coding School

N-Grams

Method	Text
	It’s bad, not good at all → bad not good all
Tokenization	“bad” “not” “good” “all”
2-grams	“bad” “not” “good” “all” “bad not” “not good” “good all”
3-grams	“bad” “not” “good” “all” “bad not” “not good” “good all” “bad not good” “not good all”
	It’s good, not bad at all → good not bad all
Tokenization	“bad” “not” “good” “all”
2-grams	“bad” “not” “good” “all” “good not” “not bad” “bad all”
3-grams	“bad” “not” “good” “all” “good not” “not bad” “bad all” “good not bad” “not bad all”

Purwadhika
Startup and Coding School

Text Preprocessing Part 1

Method	Text
	I’m still here in November 2020, enjoying my happy life.
Lowercase	i’m still here in november 2020, enjoying my happy life.
Remove contraction	i am still here in november 2020, enjoying my happy life.
Remove or convert number into text	i am still here in november , enjoying my happy life.
Remove punctuation	i am still here in november enjoying my happy life
Remove white spaces	i am still here in november enjoying my happy life
Remove stop words and particular words	i here november enjoying happy life

Purwadhika
Startup and Coding School

Text Preprocessing Part 2

Method	Text
	i am here november enjoying happy life
Bag of Words	“i” “am” “here” “november” “enjoying” “happy” “life”
N-Grams (2-grams)	“i am” “here november” “november enjoying” … “happy life”
Stemming	“i” “am” “here” “november” “enjoy” “happy” “life”
Lemmatization	“i” “be” “here” “november” “enjoy” “happy” “life”

Optional

Optional

In term of building machine, we can choose the combination of method using cross validation

Document Term Matrix (DTM)

Original Statement

- (D1) fun learning is fun
- (D2) I can do this all day
- (D3) I hate this feeling

Processed Statement

- (D1) fun learn fun
- (D2) I can do all day
- (D3) I hate feel

	learning	fun	I	can	do	all	day	hate	feel
D1	1	2	0	0	0	0	0	0	0
D2	0	0	1	1	1	1	1	0	0
D3	0	0	1	0	0	0	0	1	1

Corpus

Purwadhika
Startup and Coding School

Term Frequency (TF)

- Frequency term in the document
- i.e. if the word appears twice, the frequency in the vector will be 2
- (D1) fun learning is fun

	learning	fun	I	can	do	all	day	hate	feel
D1	1	2	0	0	0	0	0	0	0
D2	0	0	1	1	1	1	1	0	0
D3	0	0	1	0	0	0	0	1	1

Purwadhika
Startup and Coding School

Term Frequency - Inverse Document Frequency (TF-IDF)

- rescale features by how informative we expect them to be
- give high weight to any term appear often in particular document, not in many documents
- tfidf(word, doc) = tf(word) log((N+1)/(Nw+1)) + 1, with
  - tf(word, doc) : term freq of certain word of document
  - Nw : number of doc where the words appear
  - N : number of doc in training set

	learning	fun	I	can	do	all	day	hate	feel
D1	1.693	2.386	0	0	0	0	0	0	0
D2	0	0	1.287	1.693	1.693	1.693	1.693	0	0
D3	0	0	1.287	0	0	0	0	1.693	1.693

TF-IDF Calculation

- tfidf(word, doc) = tf(word,doc) log((N+1)/(Nw+1)) + 1, with
- tfidf for word learning and document D1
- N = 3, Nw = 1, tf(learning,D1) = 1
- tfidf(learning, D1) = 1 log(4/2) + 1 = 1.693

	learning	fun	I	can	do	all	day	hate	feel
D1	1.693	2.386	0	0	0	0	0	0	0
D2	0	0	1.287	1.693	1.693	1.693	1.693	0	0
D3	0	0	1.287	0	0	0	0	1.693	1.693

Text Exploration

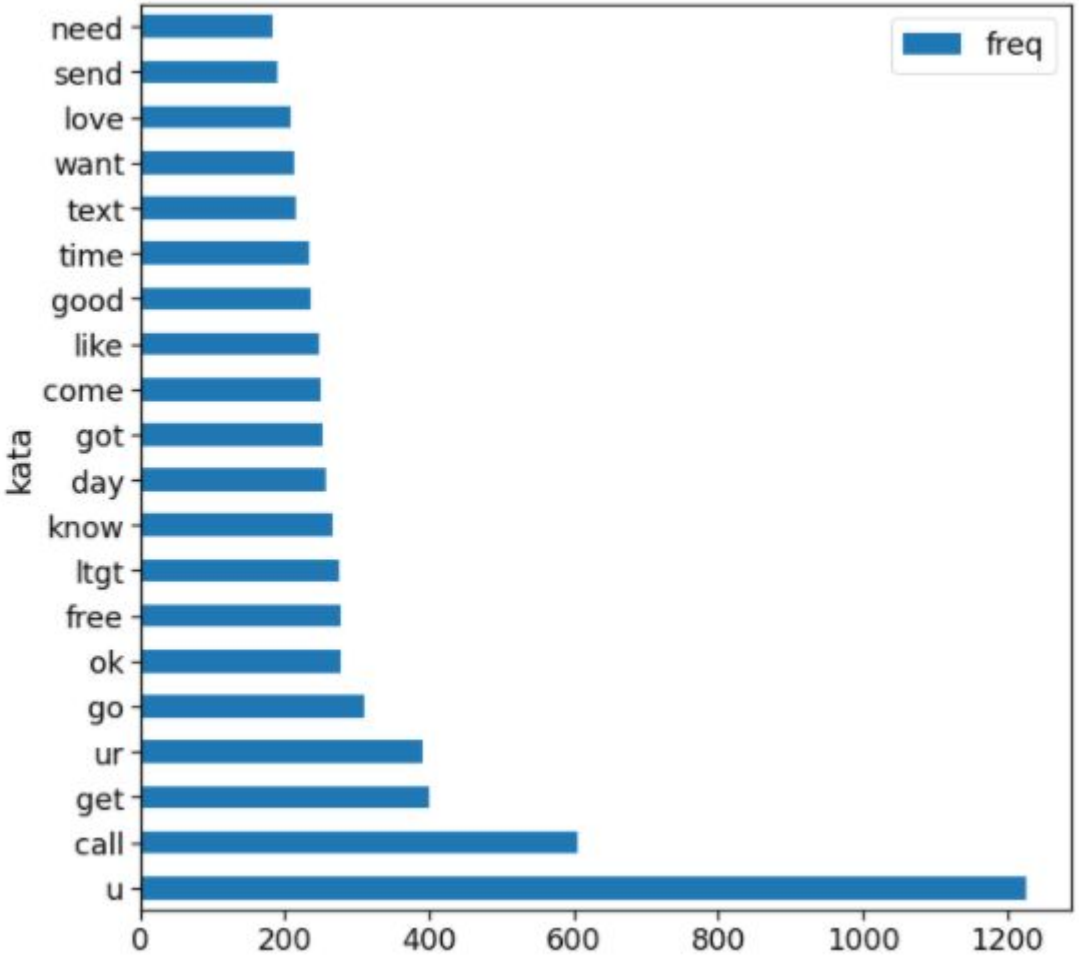
Purwadhika
Startup and Coding School

Text Exploration Method

- Word Frequency
- Word Cloud
- Length of Sentence

Purwadhika
Startup and Coding School

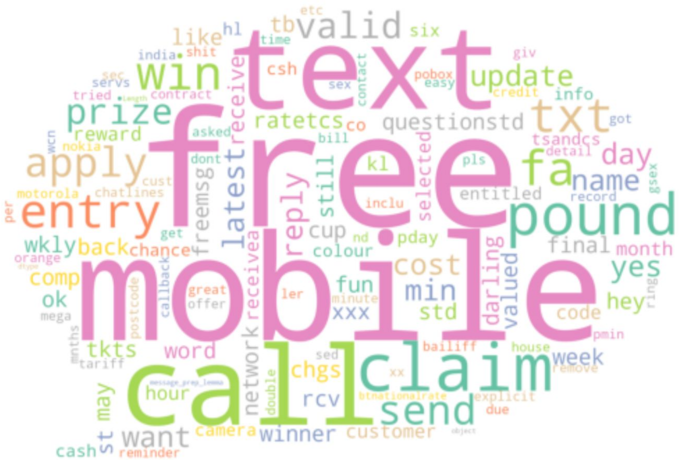
Word Frequency



Can be used to identify whether there are still words frequently occur but not meaningful

Purwadhika
Startup and Coding School

Word Cloud



SPAM

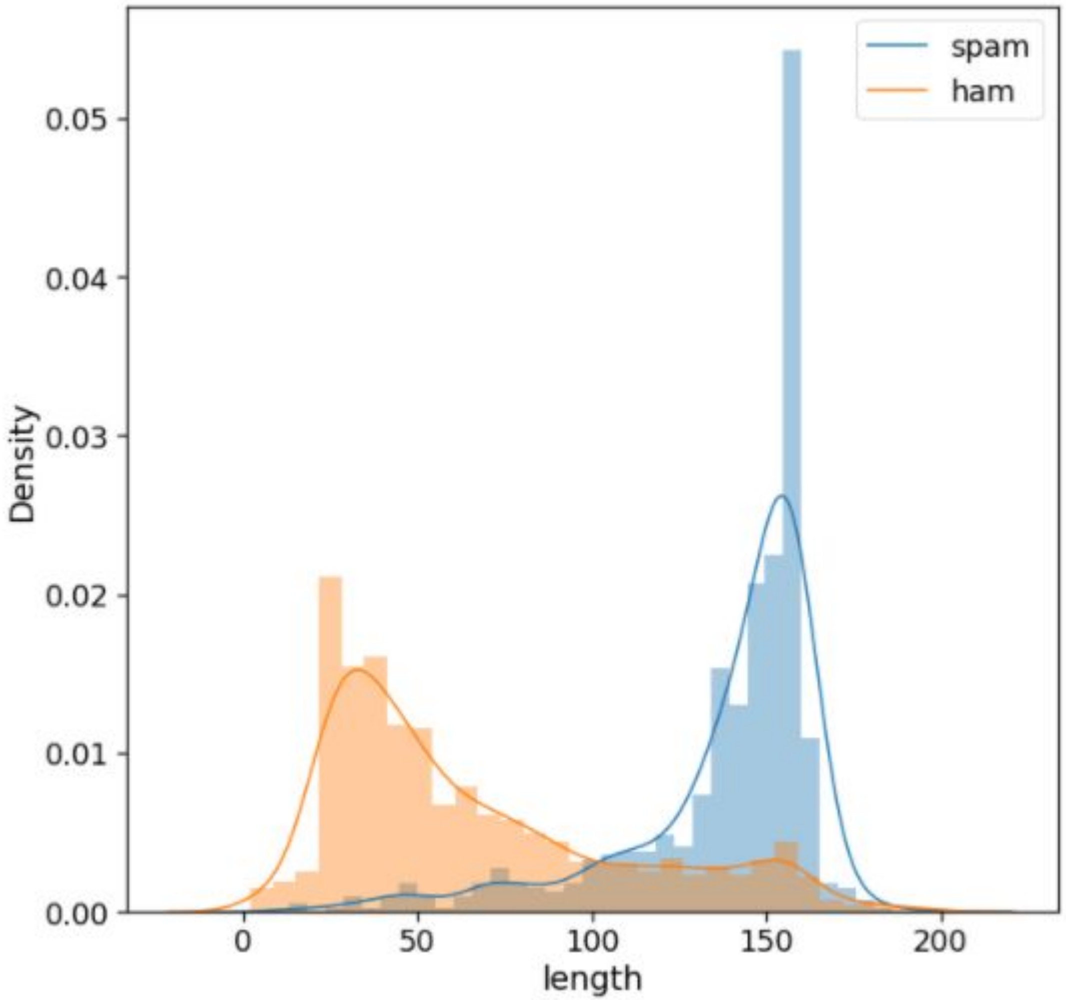
NON-SPAM

Can be used to identify whether there are still words frequently occur but not meaningful

Can be used as comparison

Purwadhika
Startup and Coding School

Length of Sentence



In identify what can differ between spam and non-spam we can utilize another content such as length of the character

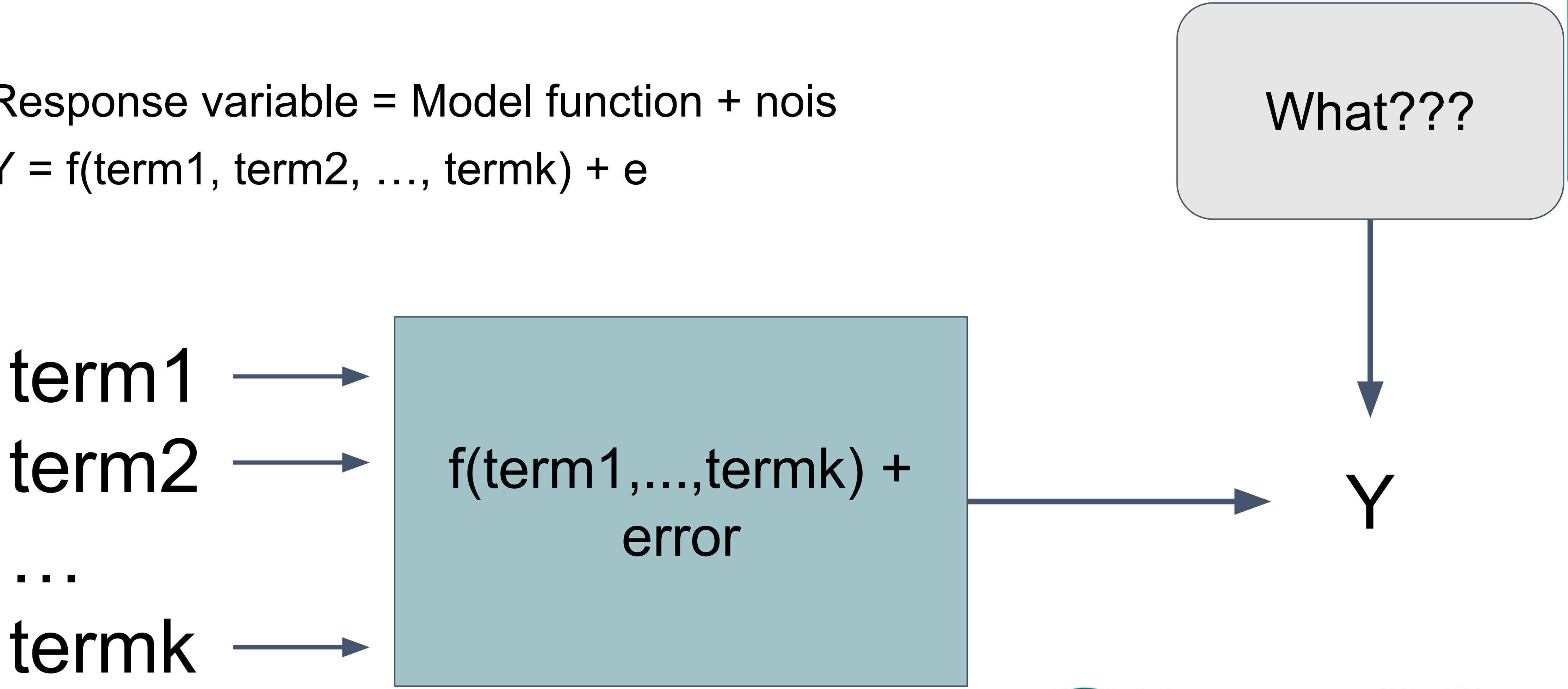
Purwadhika
Startup and Coding School

Text Classification

Text Classification

Response variable = Model function + nois

Y = f(term1, term2, …, termk) + e



Purwadhika
Startup and Coding School

Features

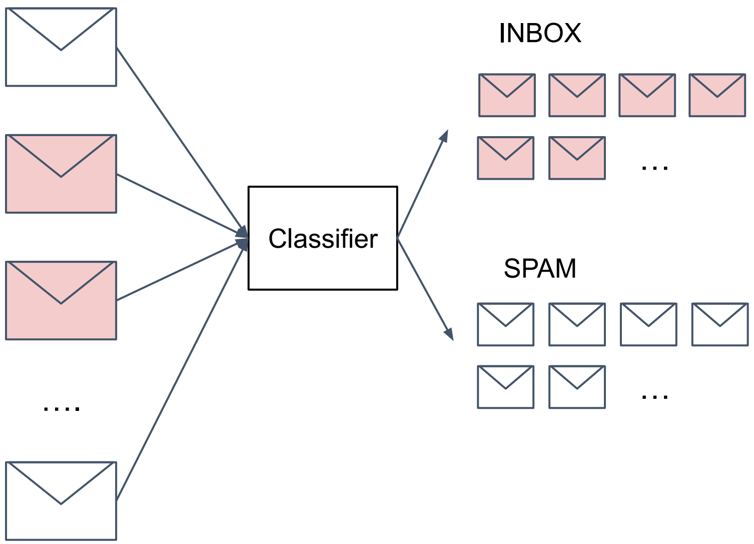
In text classification we use each words as features, the method we use can be either Term Frequency or Term Frequency - Inverse Document Frequency

TF	learning	fun	I	can	do	all	day	hate	feel
D1	1	2	0	0	0	0	0	0	0
D2	0	0	1	1	1	1	1	0	0
D3	0	0	1	0	0	0	0	1	1

TF-IDF	learning	fun	I	can	do	all	day	hate	feel
D1	1.693	2.386	0	0	0	0	0	0	0
D2	0	0	1.287	1.693	1.693	1.693	1.693	0	0
D3	0	0	1.287	0	0	0	0	1.693	1.693

Purwadhika
Startup and Coding School

Spam Detection



Spam Detection

Problem	How might we differentiate spam or legitimate messages so we can save our time by avoiding spam messages ?
Data	5572 messages with 4825 legitimate massages and 747 spam
ML Objective	Minimize false rate of prediction
Action	Place non-spam messages on inbox and suspected spam on spam
Value	Saving time by avoiding spam messages

Purwadhika
Startup and Coding School

Machine Learning Algorithm

You can use some algo you already learned

- Logistic Regression
- RF
- Boosting
- etc

There are another method such as

- Naive Bayes
- Support Vector Classifier (SVC)
- Deep Learning

Purwadhika
Startup and Coding School

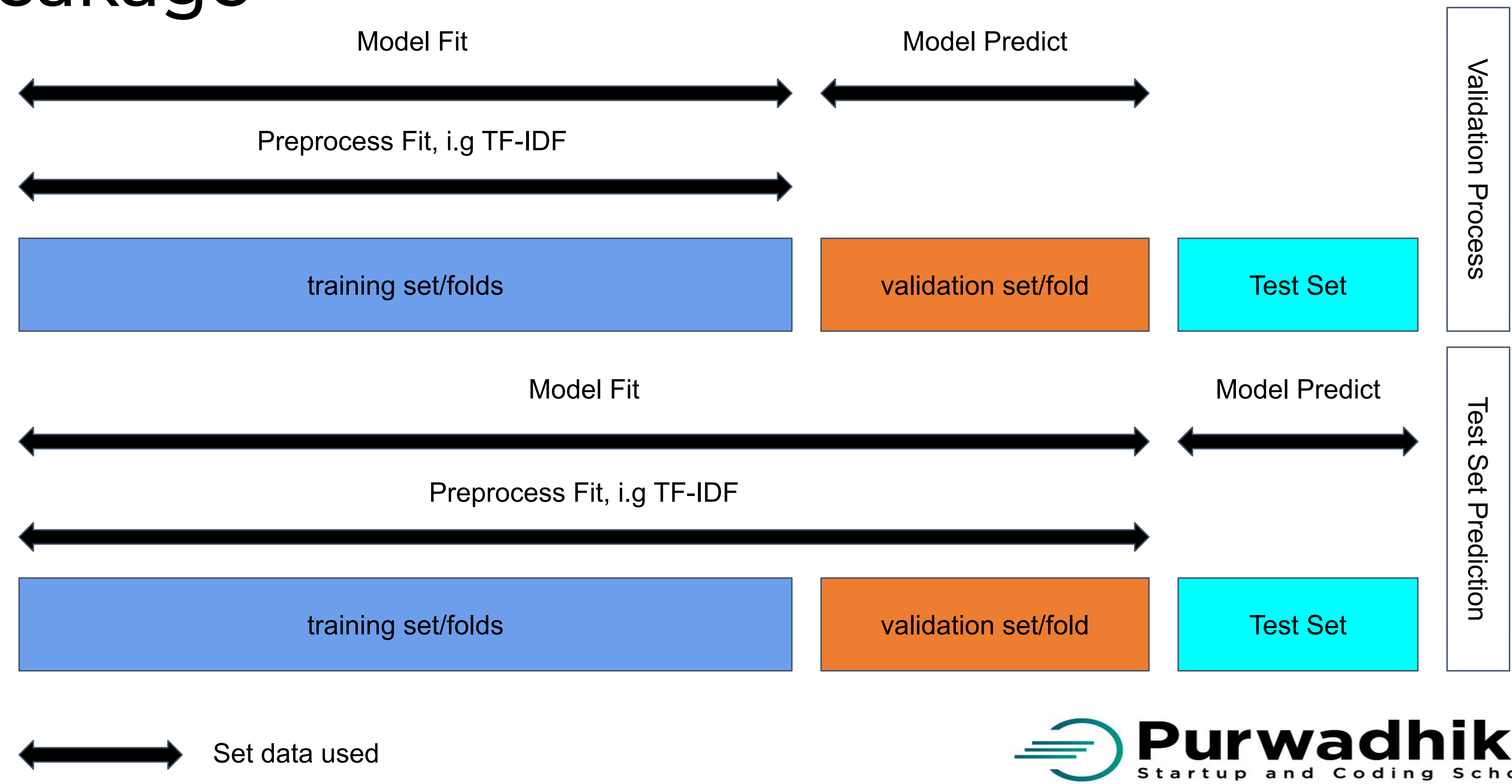
Preprocessing

- TF-IDF makes use of statistical properties of the training set (data size)
- we should use pipeline to ensure the model selection/hyperparameter tuning result are valid

```
# cross validation
cv = StratifiedKFold(n_splits = 5, random_state = 12)

# model spec
param_grid = {'logisticregression__C': [0.001, 0.01, 0.1, 1],
              'tfidfvectorizer__min_df':[5,20,50,100],
              'tfidfvectorizer__max_df':[0.7,0.8,0.9],
              'tfidfvectorizer__ngram_range':[(1,1),(1,2),(1,3)]}

# model
model = LogisticRegression(solver='newton-cg',random_state = 11)
tf_idf = TfidfVectorizer()
pipe = make_pipeline(tf_idf,model)

# model search
grid = GridSearchCV(pipe,
                    param_grid,
                    cv=cv,
                    refit = 'f1_score',
                    n_jobs = -1)
grid.fit(text_train, y_train)

# result
print("Best cross-validation score: {:.2f}".format(grid.best_score_))
print("Best parameters: ", grid.best_params_)
```

Purwadhika
Startup and Coding School

Proper Preprocessing : No Information Leakage



Purwadhika
Startup and Coding School

References

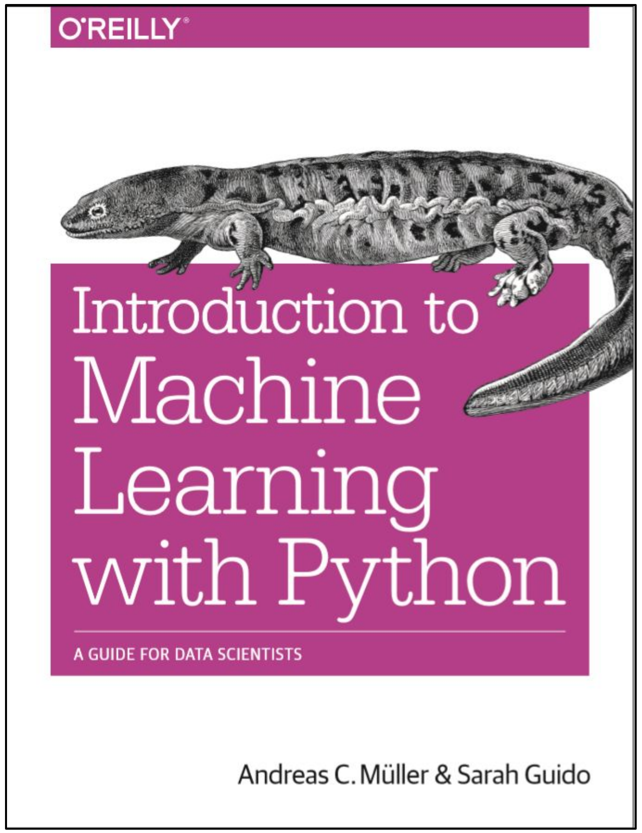
https://medium.com/@irvanseptiar/introduction-sentiment-analysis-mudah-5785f88e435d

https://medium.com/swlh/text-normalization-7ecc8e084e31

https://towardsdatascience.com/simple-wordcloud-in-python-2ae54a9f58e5

Purwadhika
Startup and Coding School

References



Text Mining in Practice with R

Ted Kwartler

WILEY

Mastering Machine Learning with Python in six Steps

A Practical Implementation Guide to Predictive Data Analytics using Python

Manohar Swamynathan

Apress