Google Developer Student Clubs

GDSC Sunway University presents

Getting Started with Natural Language Processing

Welcome everybody!

Sit back and relax, the session will begin shortly ☺️

While waiting...

Create a Kaggle Account:

https://www.kaggle.com/

Download/Open slide:

https://github.com/Grg0rry/NLP-Workshop

What is Google Developer Student Clubs (GDSC)?

Fancy name but what is it about?

University based community groups for students interested in **Google developer technologies.**

Students grow their knowledge in a **peer-to-peer learning** environment and **build solutions** that solve local problems.

Google Developer Student Clubs
Sunway University

Sunway Tech Club x GDSC Sunway University

A partnership with Google Developers!

Sunway Tech Club (STC) is partnered with the Google Developer Student Clubs (GDSC) program in Sunway University!

SUNWAY TECH CLUB X Google Developer Student Clubs
Sunway University

Google Developer Student Clubs

{

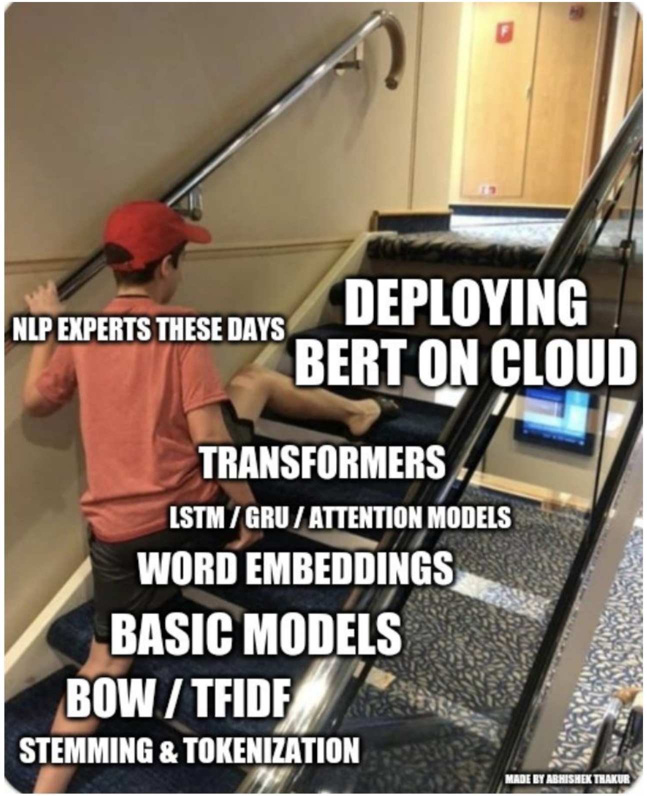
Workshop:
"Getting Started with Natural Language Processing"

}

By Gregory :)

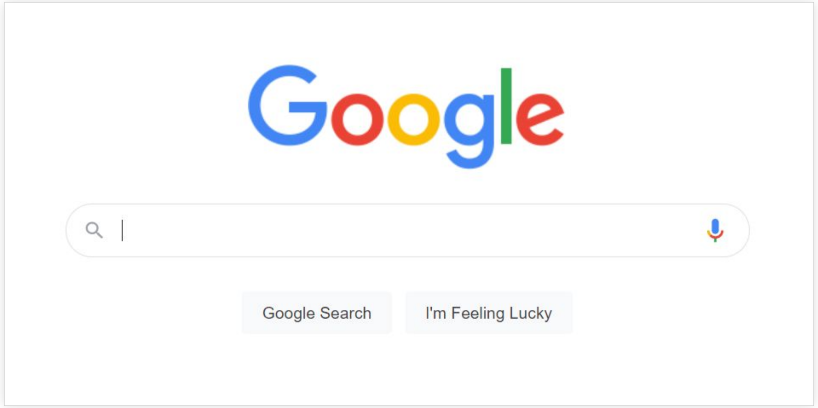
Overview

- Introduction to Natural Language Processing
- Machine Learning Workflow
    - Data Preprocessing
    - Feature Extraction - BoW & TFIDF *only :'(*
    - Model Fitting
    - Model Evaluation
- Hands On Session
  (Prediction on tweets dataset)



Real World Examples

Google Translate



What is Natural Language Processing?

“

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the **branch of artificial intelligence** or AI—concerned with giving computers the **ability to understand text and spoken words** in much the same way human beings can.

”

– stolen from the internet (IBM)

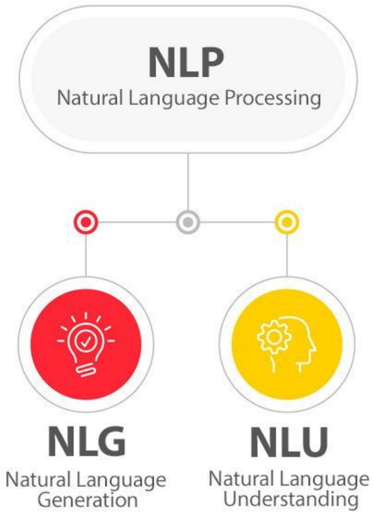
Challenges to Natural Language Processing

- **Mostly Unstructured Data**
  Not standardized and Lacks a set of rules/format that the data follows
  (eg. Age ⟶ Integer 0 to 100++, Marital status ⟶ single, married, widowed, divorced)

- **Ambiguous & Uniqueness of Language**
  - Words with multiple meaning
  - Local slang
  - Short forms/acronym
  - Misspellings
  - Different languages

- **Require Translation**
  Text to numerical form for computation needs

How does Natural Language Processing work?

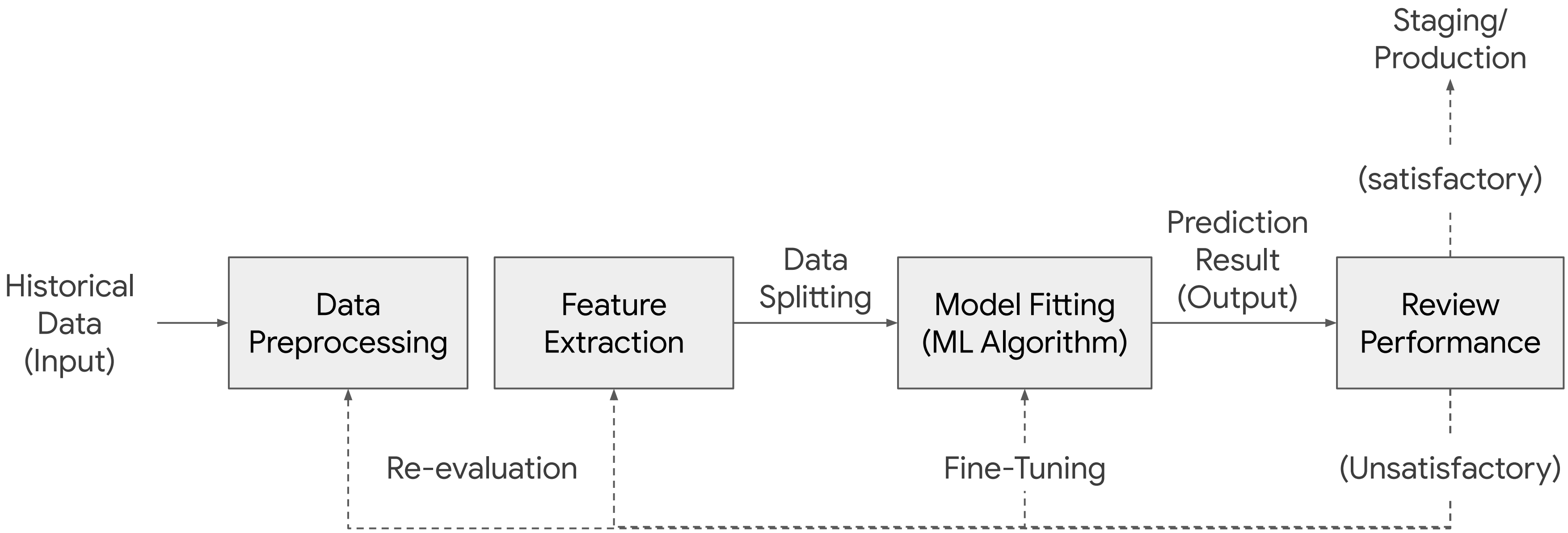


Natural Language Processing	
Natural Language Understanding	Natural Language Generation
Involve with understanding the • context, • semantics, • sentiment, • intent, and • syntax *(aka. meaning of a given text)*	Involve with finding out how to • communicate, • form its sentences, and • use appropriate wording so that it can be well understood by the reader/listener

Use Cases of Natural Language Processing

- Natural Language Processing (NLP)
  - Natural Language Understanding (NLU)
    - Text Classification/Clustering
      Eg. whether an email is spam or non spam
    - Sentiment Analysis
      Eg. whether the review given is +ve or -ve
    - Topic Modelling
      Eg. categorising a group of articles/documents
    - Entity/Information Extraction
      Eg. extracting email address from a body of text
  - Natural Language Generation (NLG)
    - Language Translation
    - Generating Codes
    - Text Summarization
    - Conversational AI/Chatbots

Machine Learning Workflow



Hands On Sesh :o

https://github.com/Grg0rry/NLP-Workshop

Dataset and Notebook URLs

https://www.kaggle.com/competitions/nlp-getting-started/data?select=train.csv

colab

https://colab.research.google.com/drive/1WGOwNNVwVMLQO6qSbFOkt0A7m5c6pFDo?usp=sharing

Frequently Used Terminology

No	ChatGPT Quotes
1.	"Love is a powerful emotion that can inspire kindness and compassion."
2.	"Technology has transformed the way we connect and access information."
3.	"Exercise benefits physical health, mental well-being, and emotional resilience."
4.	"Education is essential for social and economic progress."
5.	"The arts enrich human culture through creative expression and exploration."

Corpus

Document

{
‘Education’ : 1,
‘is’ : 1,
‘essential’ : 1,
‘for’ : 1,
... More
}

Vocabulary

Word

Data Preprocessing

https://github.com/Grg0rry/NLP-Workshop

01. Data Preprocessing

Purpose of Data Preprocessing:

1. Reduce words/complexity
   - Quicker processing (Reduced training and prediction time)
   - Reduces noise that could disrupt the model's performance

2. Normalizes text data
   - Reduce ambiguity in text
   - Standardizes the corpus of text

01. Data Preprocessing (Cont.)

01 → 02 → 03 → 04

01 — <u>Tokenization</u>

Breaks down text into words/sentences
(String to List of tokens)

02 — <u>Filter Noises</u>

Filter out unwanted characters
(eg. symbols, punctuations, etc.)

03 — <u>Remove Stopwords</u>

Remove words that don't carry much meaning
(eg. the, I, you, etc.)

04 — <u>Stemming/ Lemmatization</u>

Transform words to root form
(eg. running to run, cooked to cook)

Additional Preprocessing Methods:

- Removal of PII data (Entity Extraction),
- Spelling Correction,

01. Data Preprocessing (Cont.)

Difference between Stemming and Lemmatization:

Stemming	Lemmatization

● Faster preprocessing
● Possibility of incorrect context
    ○ *Eg. "universal", "university", "universe" → "univers"*
	● Slower preprocessing
	● Carries more accurate text context
	    ○ Part-of-Speech (POS) Tagging
	    ○ Returns root word based of POS

Part-of-Speech (POS) Tagging Example:

Why	not	tell	someone	?
adverb	adverb	verb	noun	punctuation mark, sentence closer

Feature Extraction

https://github.com/Grg0rry/NLP-Workshop

02. Feature Extraction

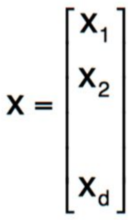
Purpose of Feature Extraction:

- Represent text in numerical form
- Maintains the context and meaning

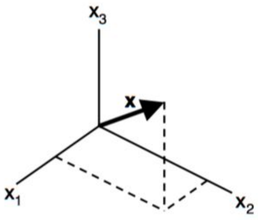
How it works?

Text vectorization

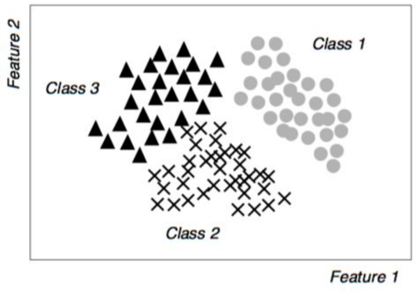
- Represented as vectors in vector space
- Capture the semantics and relationship between words



Feature vector



Feature space (3D)



Scatter plot (2D)

02. Feature Extraction (Cont.)

- 01 Bag-of-Words (BoW)
- 02 Term Frequency-Inverse Document Frequency (TF-IDF)
- 03 N-grams
- 04 Word Embeddings

Sample Corpus:

Document 1: The movie was bad!

Document 2: Great movie!

Document 3: Bad writing, so boring

02. Feature Extraction (Cont.)

Bag-of-Words (BoW)

	the	movie	was	bad	great	writing	so	boring
The movie was bad!	1	1	1	1	0	0	0	0
Great movie!	0	1	0	0	1	0	0	0
Bad writing, so boring movie	0	1	0	1	0	1	1	1

- **Count frequency** of tokens repeating in the corpus (**Vocabulary**)
- More frequent = More important

02. Feature Extraction (Cont.)

Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency:

- Count frequency of tokens same as BoW

Inverse Document Frequency:

- log(N / df)

N: number of documents
df: number of documents with the word

Output = TF * IDF

	the	movie	was	bad	great	writing	so	boring
The movie was bad!	1	1	1	1	0	0	0	0
Great movie!	0	1	0	0	1	0	0	0
Bad writing, so boring movie	0	1	0	1	0	1	1	1

02. Feature Extraction (Cont.)

Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency:
- Count frequency of tokens same as BoW

Inverse Document Frequency:
- log(N / df)

N: number of documents
df: number of documents with the word

Output = TF * IDF

	the	ng
The movie was bad!	1	
Great movie!	0	
Bad writing, so boring movie	0	

"the":

document 1:
TF = 1,

IDF = log(3/1)
= 0.4771

TF*IDF = 1 * 0.4771
= 0.4771

document 2:
TF = 0,

IDF = log(3/1)
= 0.4771

TF*IDF = 0 * 0.4771
= 0

02. Feature Extraction (Cont.)

Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency:
- Count frequency of tokens same as BoW

Inverse Document Frequency:
- log(N / df)

N: number of documents
df: number of documents with the word

Output = TF * IDF

	the
The movie was bad!	0.48
Great movie!	0
Bad writing, so boring movie	0

"the":

document 1:
TF = 1,

IDF = log(3/1)
= 0.4771

TF*IDF = 1 * 0.4771
= 0.4771

document 2:
TF = 0,

IDF = log(3/1)
= 0.4771

TF*IDF = 0 * 0.4771
= 0

02. Feature Extraction (Cont.)

Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency:
- Count frequency of tokens same as BoW

Inverse Document Frequency:
- log(N / df)

N: number of documents
df: number of documents with the word

Output = TF * IDF

	the	movie
The movie was bad!	0.48	1
Great movie!	0	1
Bad writing, so boring movie	0	1

"movie":

document 1:
TF = 1,

IDF = log(3/3)
= 0

TF*IDF = 1 * 0
= 0

document 2:
TF = 1,

IDF = log(3/3)
= 0

TF*IDF = 1 * 0
= 0

02. Feature Extraction (Cont.)

Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency:

- Count frequency of tokens same as BoW

Inverse Document Frequency:

- log(N / df)

N: number of documents
df: number of documents with the word

Output = TF * IDF

	the	movie
The movie was bad!	0.48	0
Great movie!	0	0
Bad writing, so boring movie	0	0

“movie”:

document 1:
TF = 1,

IDF = log(3/3)
= 0

TF*IDF = 1 * 0
= 0

document 2:
TF = 1,

IDF = log(3/3)
= 0

TF*IDF = 1 * 0
= 0

02. Feature Extraction (Cont.)

Term Frequency - Inverse Document Frequency (TF-IDF)

Term Frequency:

- Count frequency of tokens same as BoW

Inverse Document Frequency:

- log(N / df)

N: number of documents
df: number of documents with the word

Output = TF * IDF

	the	movie	was	bad	great	writing	so	boring
The movie was bad!	0.48	0	0.48	0.18	0	0	0	0
Great movie!	0	0	0	0	0.48	0	0	0
Bad writing, so boring movie	0	0	0	0.18	0	0.48	0.48	0.48

- The more frequent the word across the corpus, the less important it is
- Reduces impact of noise words

02. Feature Extraction (Cont.)

N-grams

"Bad writing, so boring movie"

Bigrams (Two Words)

Bad writing so boring movie
Bad **writing so** boring movie
Bad writing **so boring** movie
Bad writing so **boring movie**

Trigrams (Three Words)

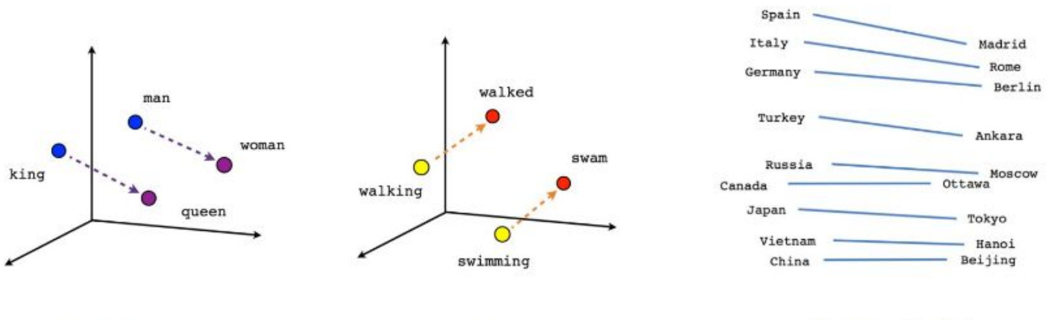
Bad writing so boring movie
Bad **writing so boring** movie
Bad writing **so boring movie**

- Useful for predicting next word in sentence
- Identify words that are commonly together

02. Feature Extraction (Cont.)

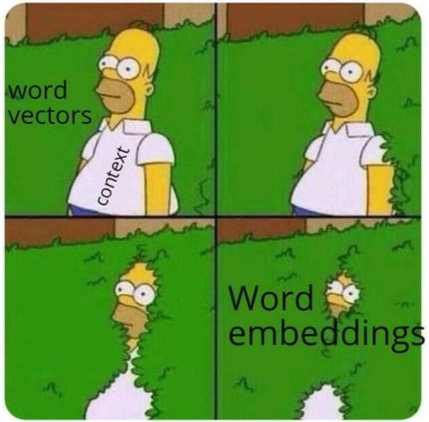
Word Embeddings

Example Algorithm: word2vec, GloVe, fasttext



Male-Female   Verb tense   Country-Capital

Word embeddings
in a nutshell



https://www.shanelynn.ie/get-busy-with-word-embeddings-introduction/

Word Embeddings Vector Space:
https://projector.tensorflow.org/

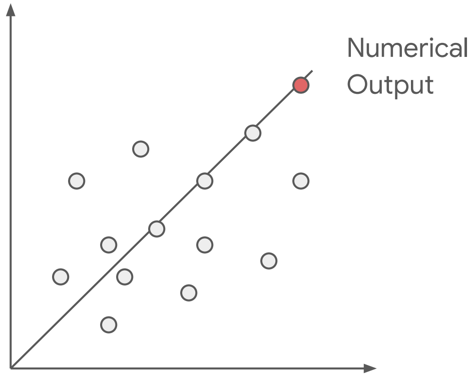
Model Fitting

https://github.com/Grg0rry/NLP-Workshop

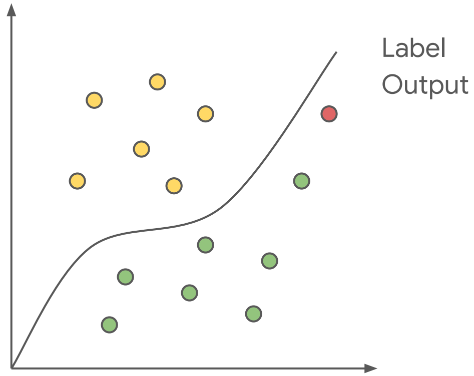
03. Model Fitting

Choosing the Right Machine Learning Model

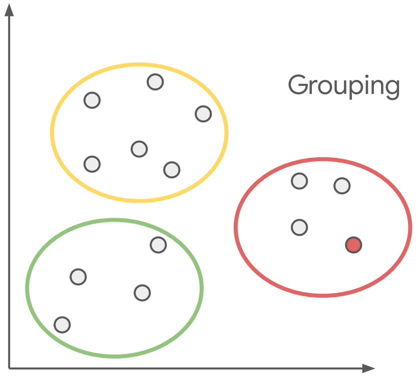
1. Identify the problem to solve



Estimation (Regression)



Classification



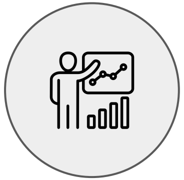
Clustering

https://vitalflux.com/most-common-types-machine-learning-problems/

03. Model Fitting

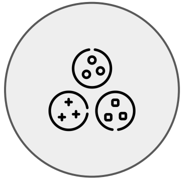
<u>Choosing the Right Machine Learning Model</u>

2. Identify the approach to take



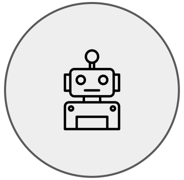
Supervised Learning

- Learns off labelled dataset



Unsupervised Learning

- Discover patterns with unlabelled dataset

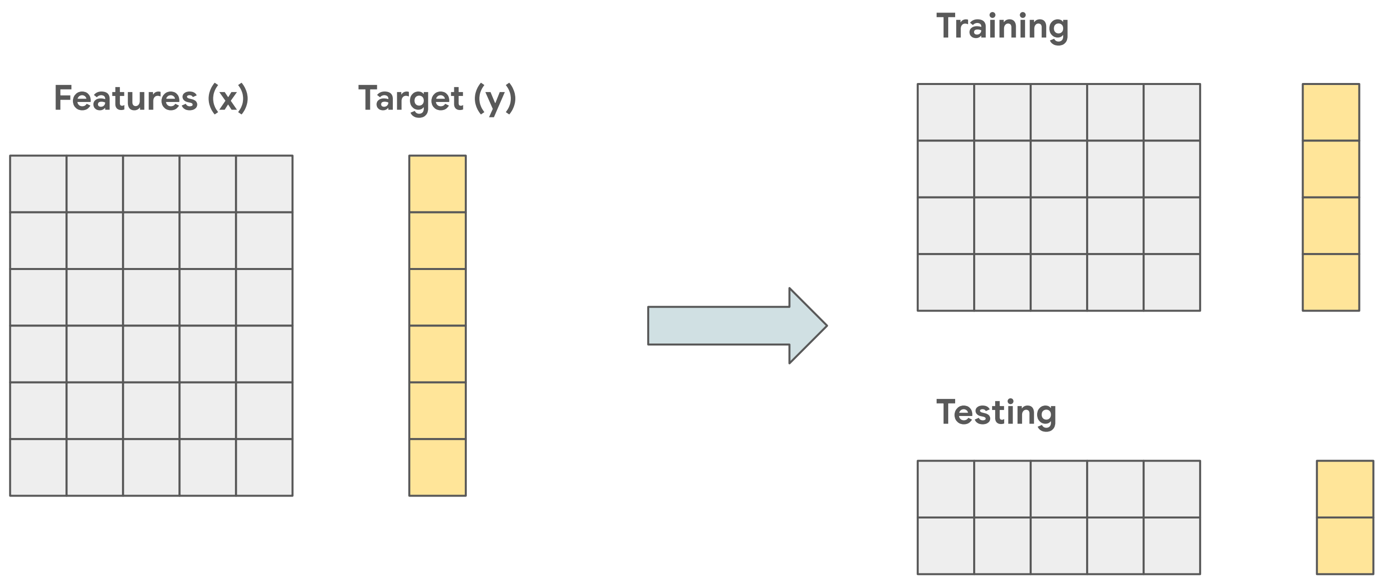


Reinforcement Learning

- Learns off simulations (feedback/reward)

03. Model Fitting

Data Splitting (Supervised Learning)

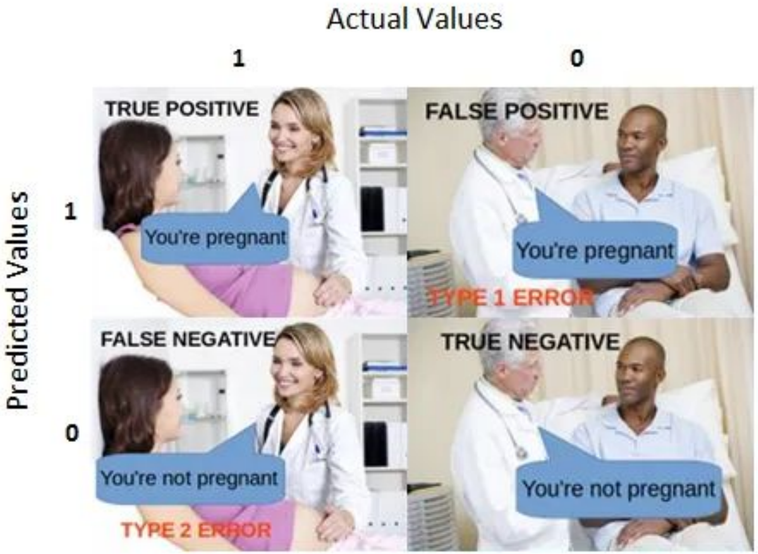


03. Model Fitting

Model Evaluation (Classification)

Confusion Matrix:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

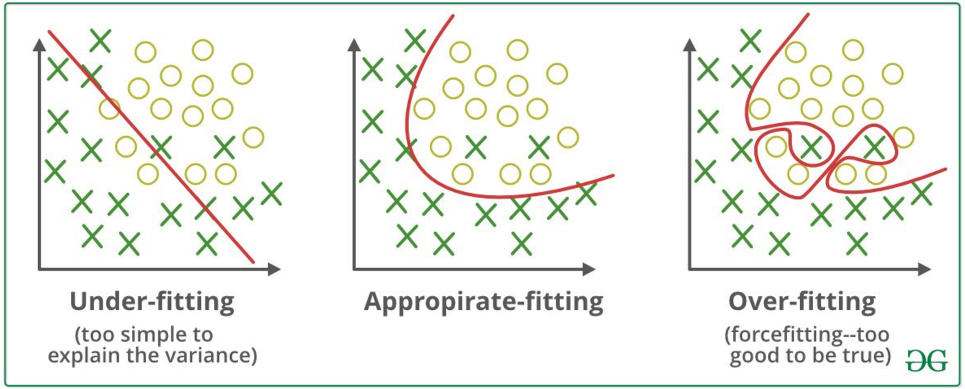


https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

03. Model Fitting

Model Evaluation (Classification)

Overfitting and Underfitting



Overfitting:

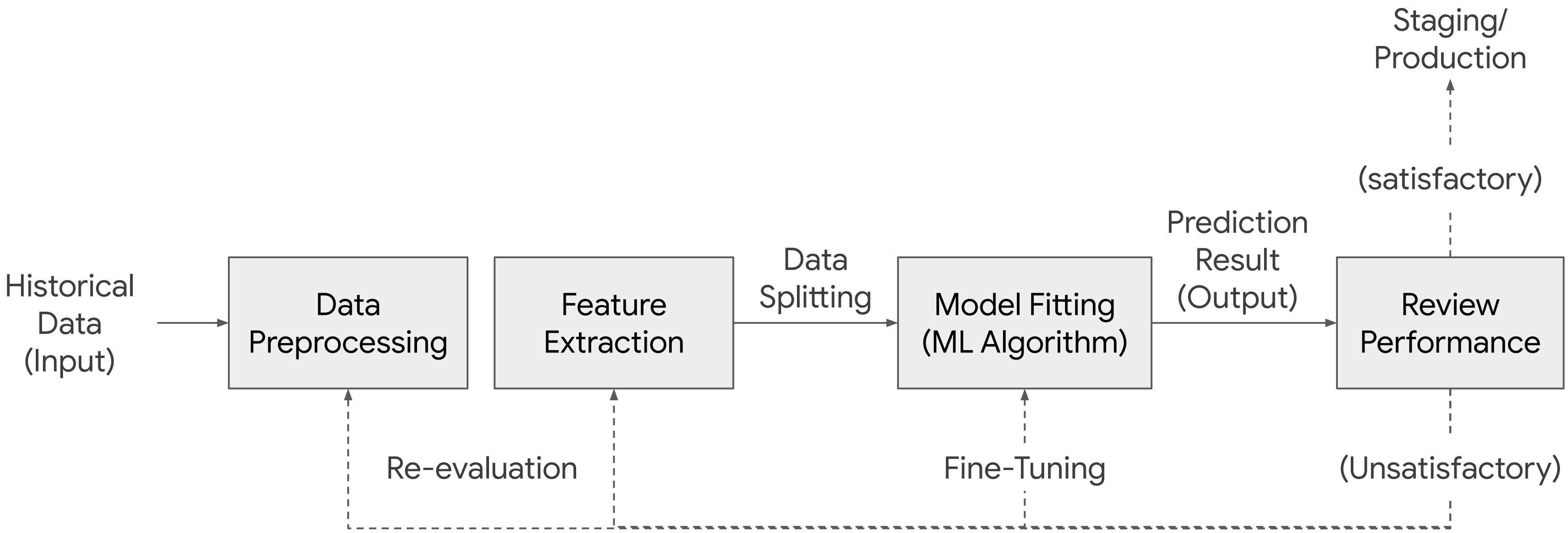
Training accuracy >>> Testing accuracy

Underfitting:

Training accuracy < Testing accuracy

https://machinelearningmastery.com/overfitting-machine-learning-models/

Machine Learning Workflow



Getting Started with Natural Language Processing

Q&A

SCAN ME

Google Developer Student Clubs

Group Photo

Follow Us on Social Media

Keep up with the cool kids

https://gdsc.community.dev/sunway-university/

fb.com/gdsc.sunway

instagram.com/gdsc.sunway

GDSC Sunway University Youtube Channel

Thank you!

@grgrrry

Tan Yong Jern (Gregory)