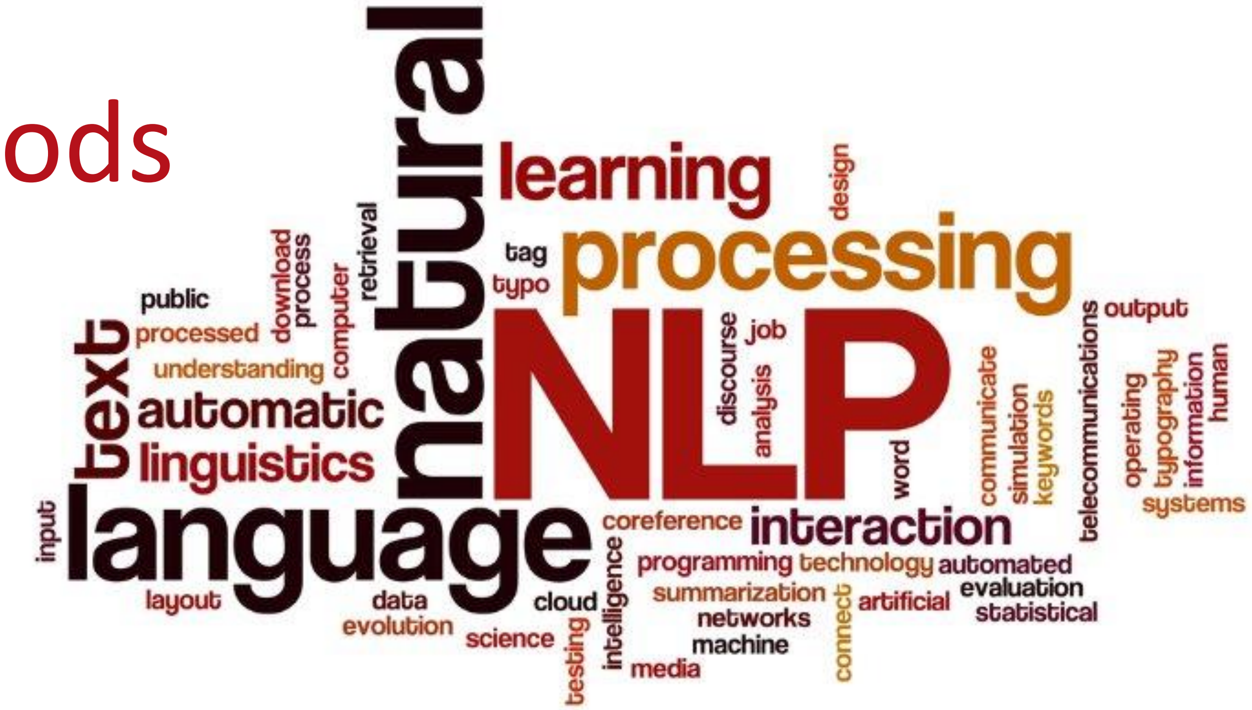
NLP Tasks and Methods

Dr. Ignatius Ezeani

Monday, 01 March 2021



Lecture Outline

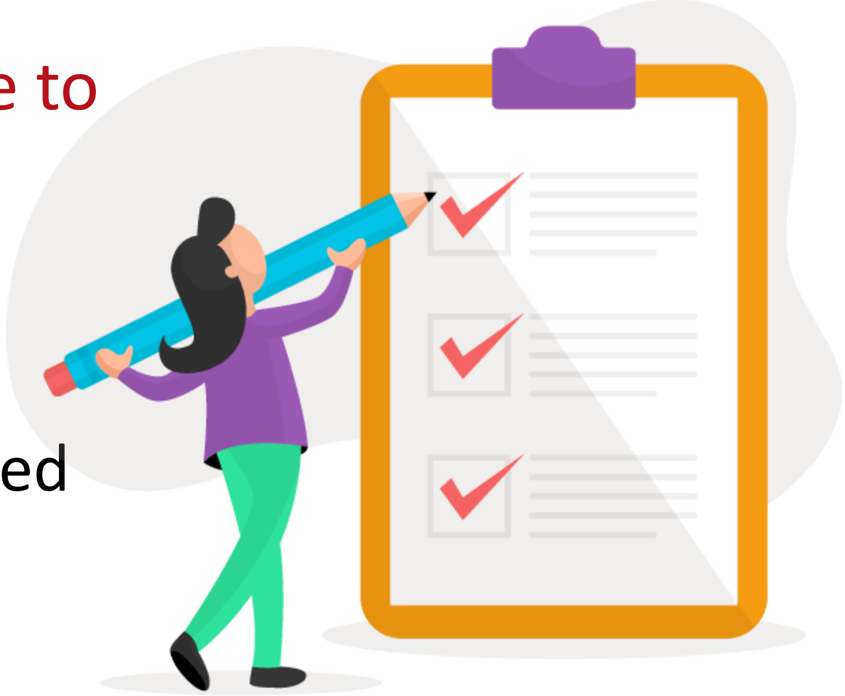
This lecture is structured in three parts:

- Part 1:
  - Why NLP? What is it?
  - Why is it hard?
- Part 2
  - Some common NLP Problems
  - Approaches to solving them
- Part 3
  - Machine Learning for NLP
  - Extracting feature from text

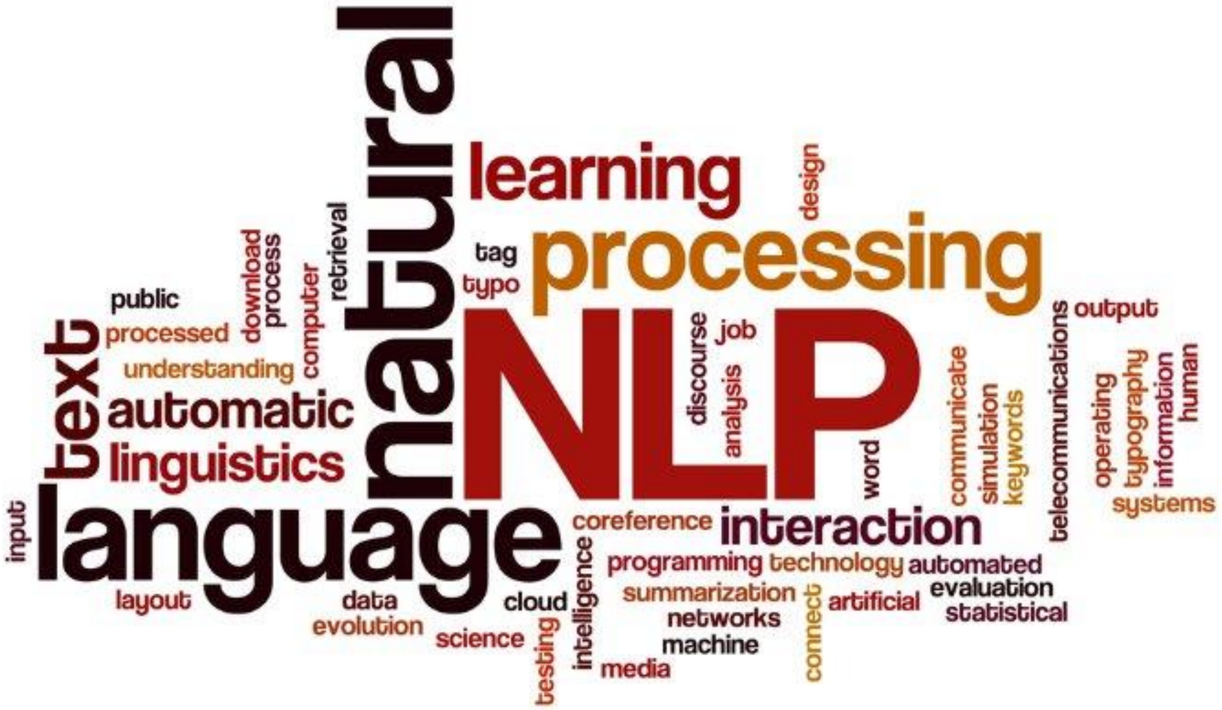
Intended Learning Outcomes

At the end of this lecture, you should be able to

- explain what NLP is and why it is hard
- discuss some common NLP problems and approaches to solving them
- explain how machine learning methods are used for NLP problems
- discuss methods feature extraction from texts



What is NLP and why do we need it?



What is NLP?

In summary NLP is often defined as:

- the study of the computational treatment of natural (human) language

It aims to:

- create systems that *understand* and *generate* (produce) human language

Who is on the internet?

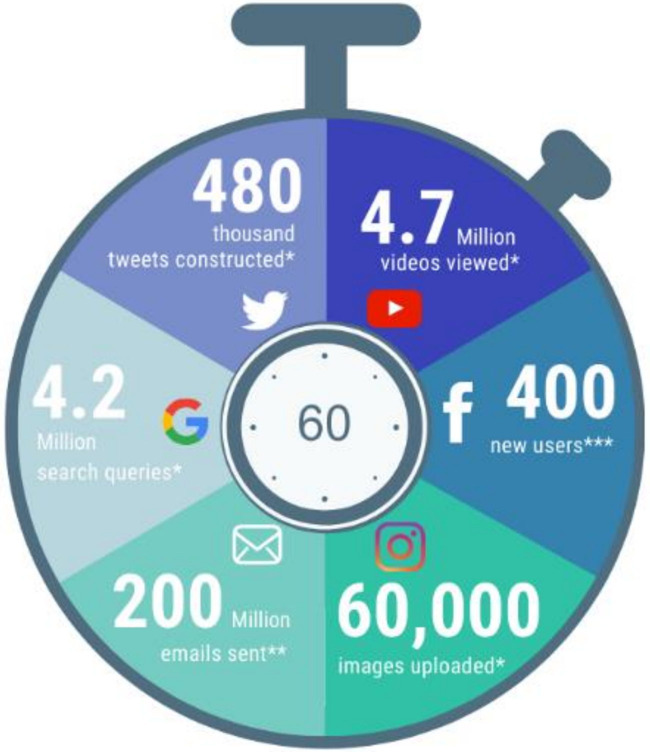
By the end of Jan 2021

- **7.83bn** people on earth
- **5.22bn** mobile phone were active
  - **4.91**bn by end of 2017
- **4.66bn** people were online
  - **3.8bn** by end of 2017
- **4.20bn** social media users

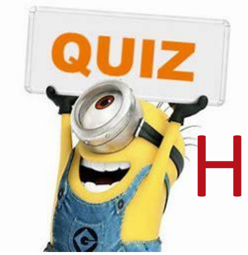
https://www.nodegraph.se/how-much-data-is-on-the-internet/

What happens every minute?

Every 60 seconds... 2020 vs 2017		
	2020	2017
Emails sent	200 million	150 million
Google searches	4.2 million	3.8 million
Tweets sent	480,000	448,800
Instagram images	60,000	66,000
YouTube video	4.7 million	4.4 million
Facebook new users	400	360



https://www.nodegraph.se/how-much-data-is-on-the-internet/

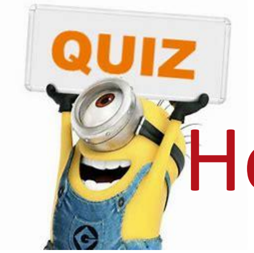


How much data has the internet?

Lancaster University

- **2.7 Zb** in 2017
- **4.4 Zb** by end of 2019
- **44 Zettabytes (Zb)** in 2020
- **175 Zb** by end of 2025
    - A gigabyte is 1,024 megabytes
    - A terabyte is 1,024 gigabytes
    - A petabyte is 1,024 terabytes
    - An exabyte is 1,024 petabytes
    - A zettabyte is 1,024 exabytes

- **Putting that in context**
- **1Zb =** 1,024 exabytes,
- **1Zb =** 1,048,576 pb,
- **1Zb =** 1,073,741,824 tb,
- **1Zb =** 1,099,511,627,776 gb,
- **1Zb =** 1,125,899,910,000,000 mb!



How much data has the internet?

Lancaster University

- **175 Zettabytes!**
- If on DVDs, the stack of DVDs would be long enough to **circle the Earth 222 times!**
- If download at the average current internet connection speed, it would take you **1.8 billion years** to download!

Why Natural Language Processing

- Our devices are now part of our lives and we often cannot function without them.
- We generate tonnes of human language data everyday and we desire to know what the data is telling us, sometimes in real-time.
- We need tools and techniques to process the enormous amount of data on the internet and communicate outputs to us in human language
- That’s why we need natural language processing techniques to create these tools

Why is NLP hard?

Humans Language Ambiguity

lexical, phrase, semantic ambiguities

- Iraqi Head Seeks Arms
  - *Word sense is ambiguous (head, arms)*
- Stolen Painting Found by Tree
  - *Thematic role is ambiguous: tree is agent or location?*
- Ban on Nude Dancing on Governor's Desk
  - *Syntactic structure (attachment) is ambiguous: is the ban or the dancing on the desk?*
- Hospitals Are Sued by 7 Foot Doctors
  - *Semantics is ambiguous : what is 7 foot?*

Why is NLP hard?

Language is subtle.

Similar the contexts may not guarantee

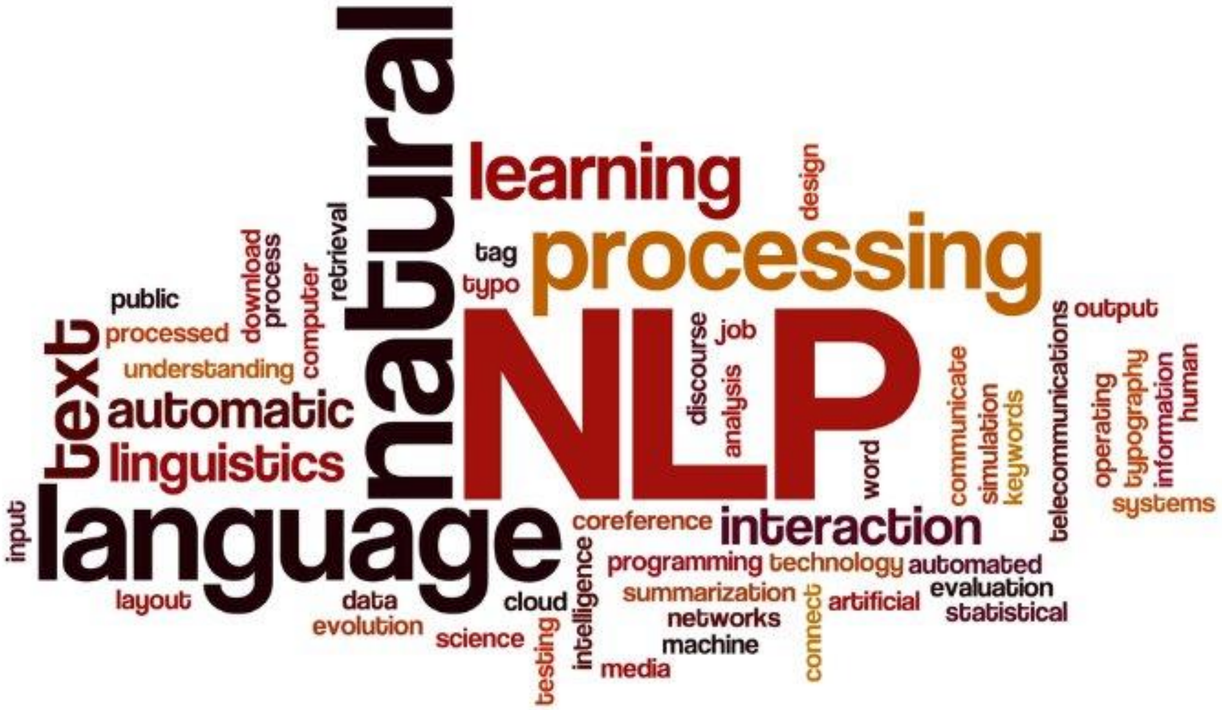
- *He <u>arrived</u> at the lecture*
- *He <u>chuckled</u> at the lecture*
- *He <u>chuckled</u> his way through the lecture*
- ***He <u>arrived</u> his way through the lecture*

Why is NLP hard?

Language is representation is unique

- There is no known universal representation, parsing rules are flexible.
- Language could be domain-specific e.g. *legal, scientific texts*
- Meanings are context-dependent, world knowledge required for interpretation
- There are so many languages, dialects and styles etc.

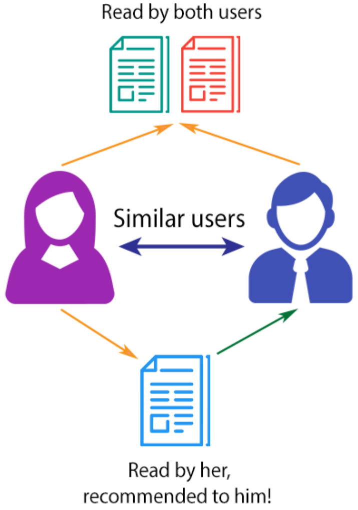
Common NLP Tasks



See an extended list of NLP tasks: https://nlpprogress.com/

Application of NLP

- Language processing
  - Web search engines
  - Text classification: sentiment, topic
  - Spam filtering etc
  - Machine translation
  - Question answering
  - Recommender Systems




"Il est impossible aux journalistes de rentrer dans les régions tibétaines"

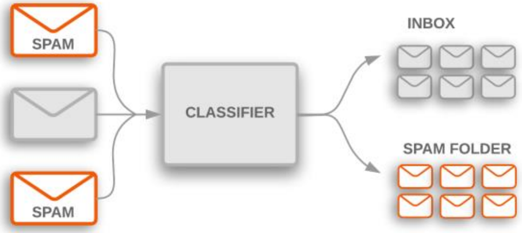
Bruno Philip, correspondant du "Monde" en Chine, estime que les journalistes de l'AFP qui ont été expulsés de la province tibétaine du Qinghai "n'étaient pas dans l'illégalité".

Les faits Le dalaï-lama dénonce l'"enfer" imposé au Tibet depuis sa fuite, en 1959
Vidéo Anniversaire de la rébellion


"It is impossible for journalists to enter Tibetan areas"

Philip Bruno, correspondent for "World" in China, said that journalists of the AFP who have been deported from the Tibetan province of Qinghai "were not illegal."

Facts The Dalai Lama denounces the "hell" imposed since he fled Tibet in 1959
Video Anniversary of the Tibetan rebellion: China on guard



Part of Speech Tagging

- One of the most fundamental tasks in NLP
- A part of speech is a category of words with similar grammatical properties.
- Common English parts of speech are **noun**, **verb**, **adjective**, **adverb**, **pronoun**, **preposition**, **conjunction**, etc.
- It involved applying certain techniques (and statistics) to identify the part of speech of a word in context e.g. in a sentence

Vinken	,	61	years	old
NNP	,	CD	NNS	JJ

https://nlpprogress.com/english/part-of-speech_tagging.html

Word Sense Disambiguation

- WSD associates words in context with the right entry in a sense inventory e.g. WordNet.
- Example, **mouse**:
  - *A <u>mouse</u> consists of an object held in one's hand, with one or more buttons.*
  - Assigns "mouse" with its electronic device sense (the 4th sense in the WordNet).

Noun

- S: (n) **mouse** (any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually hairless tails)
- S: (n) shiner, black eye, **mouse** (a swollen bruise caused by a blow to the eye)
- S: (n) **mouse** (person who is quiet or timid)
- S: (n) **mouse**, computer mouse (a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the device is a ball that rolls on the surface of the pad) *"a mouse takes much more room than a trackball"*

Verb

- S: (v) sneak, **mouse**, creep, pussyfoot (to go stealthily or furtively) *"..stead of sneaking around spying on the neighbor's house"*
- S: (v) **mouse** (manipulate the mouse of a computer)

https://nlpprogress.com/english/word_sense_disambiguation.html

Text Classification

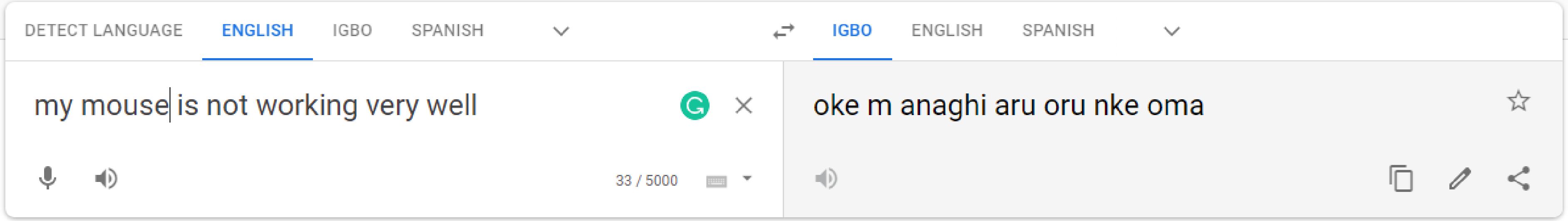
- Text classification is a very common NLP task
- It involves assigning an appropriate category to sentence or document e.g:
  - spam filtering, sentiment analysis, topic modelling, language id

Application	Description	Input	Output
Spam filtering	Detect and filter spam emails	Email	Spam / Not spam
Sentiment analysis	Detect the polarity of text	Tweet, review	Positive / Negative
Topic detection	Detect the topic of text	News article, blog post	Business / Tech / Sports
Language indentification	Detect the language of text	Written text	Igbo / English / Russian

https://nlpprogress.com/english/text_classification.html

Machine Translation

- Machine translation is the task of translating a sentence in a source language to a different target language
- Common approaches:
  - Rule-based
  - Statistical machine translation (e.g. Phrase-Based approach)
  - Neural machine translation



https://en.wikipedia.org/wiki/Machine_translation

Named Entity Recognition

- NER is the task of tagging entities in text with their corresponding type.
- It is a useful subtask for information extraction
- Approaches typically use BIO notation:
  - **B**-beginning, **I**-inside of entities.
  - **O** is used for non-entity tokens.

Mark	Watney	visited	Mars
B-PER	I-PER	O	B-LOC

https://nlpprogress.com/english/named_entity_recognition.html

Information Extraction

- Automatically extracts structured information from unstructured and/or semi-structured data
- Supports the use of logical reasoning to draw inferences from textual data
- Given:
  - "*Yesterday, New York based Foo Inc. announced their acquisition of Bar Corp.*"
- We can extract:
  - $MergerBetween(company_1, company_2, date)$
- Open Information Extraction creates large knowledge bases from the web

Spoken Language Systems

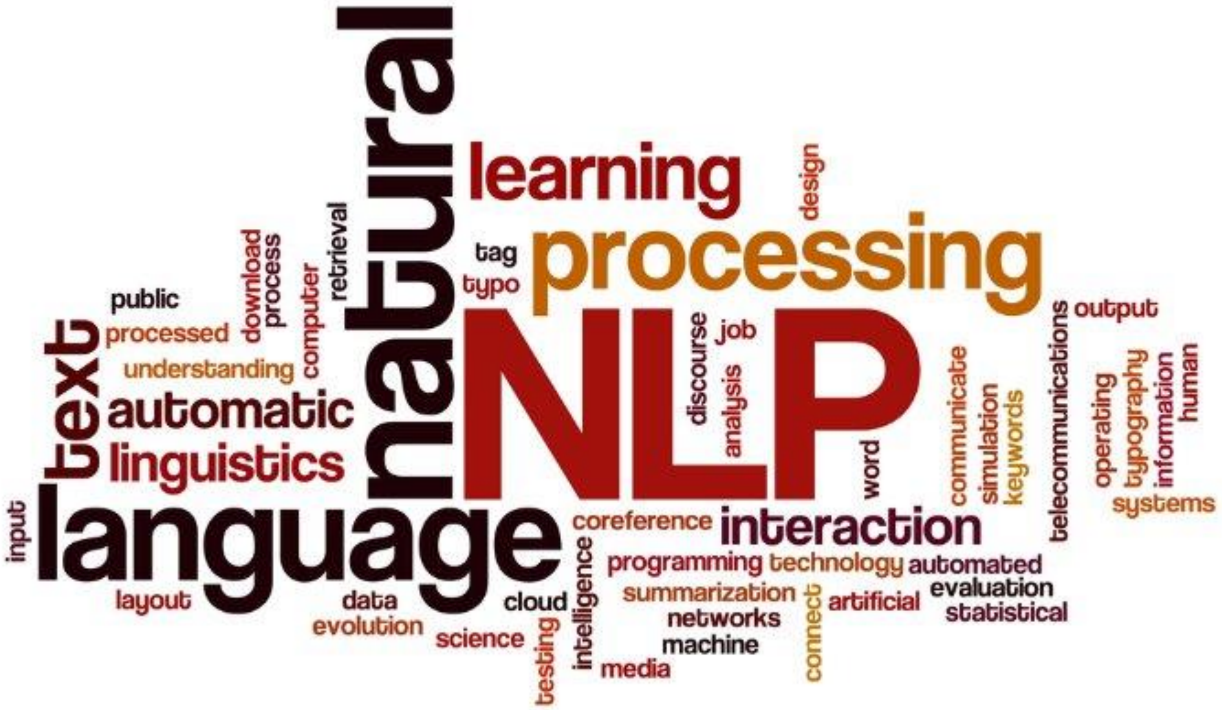
Lancaster University

- Enable the recognition and translation of spoken language into text
- Automatic Speech Recognition
- Text-to-Speech Synthesis
- Dialogue systems
- Examples:
  - Siri, Alexa, Cortana, Google Assistant

Google's supported voices and language: https://cloud.google.com/text-to-speech/docs/voices

Lancaster University

Machine Learning Overview



See an extended list of NLP tasks: https://nlpprogress.com/

Machine Learning

Early definition of machine learning

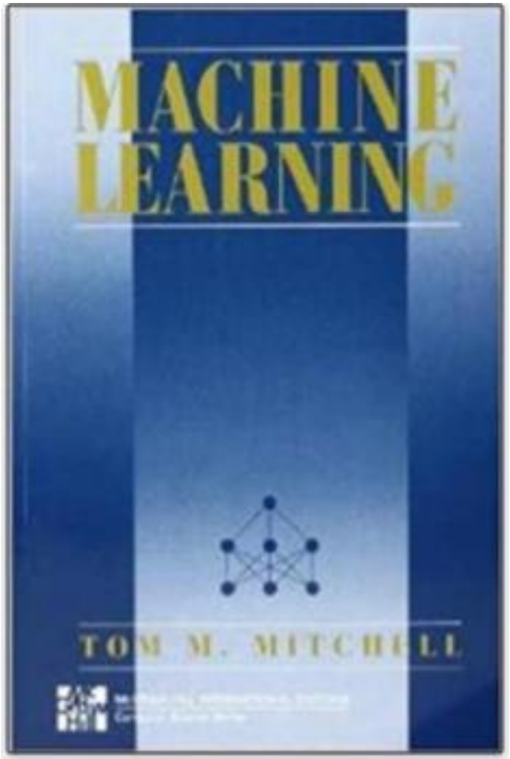
- "*Field of study that gives computers the ability to* ***learn*** *without being* ***explicitly*** *programmed*"
  - – Arthur Samuel (1959)
- ML pioneer that built first "self-learning" program that played checkers by learning from experience
- Inverted alpha-beta pruning widely used in decision tree searching

Machine Learning

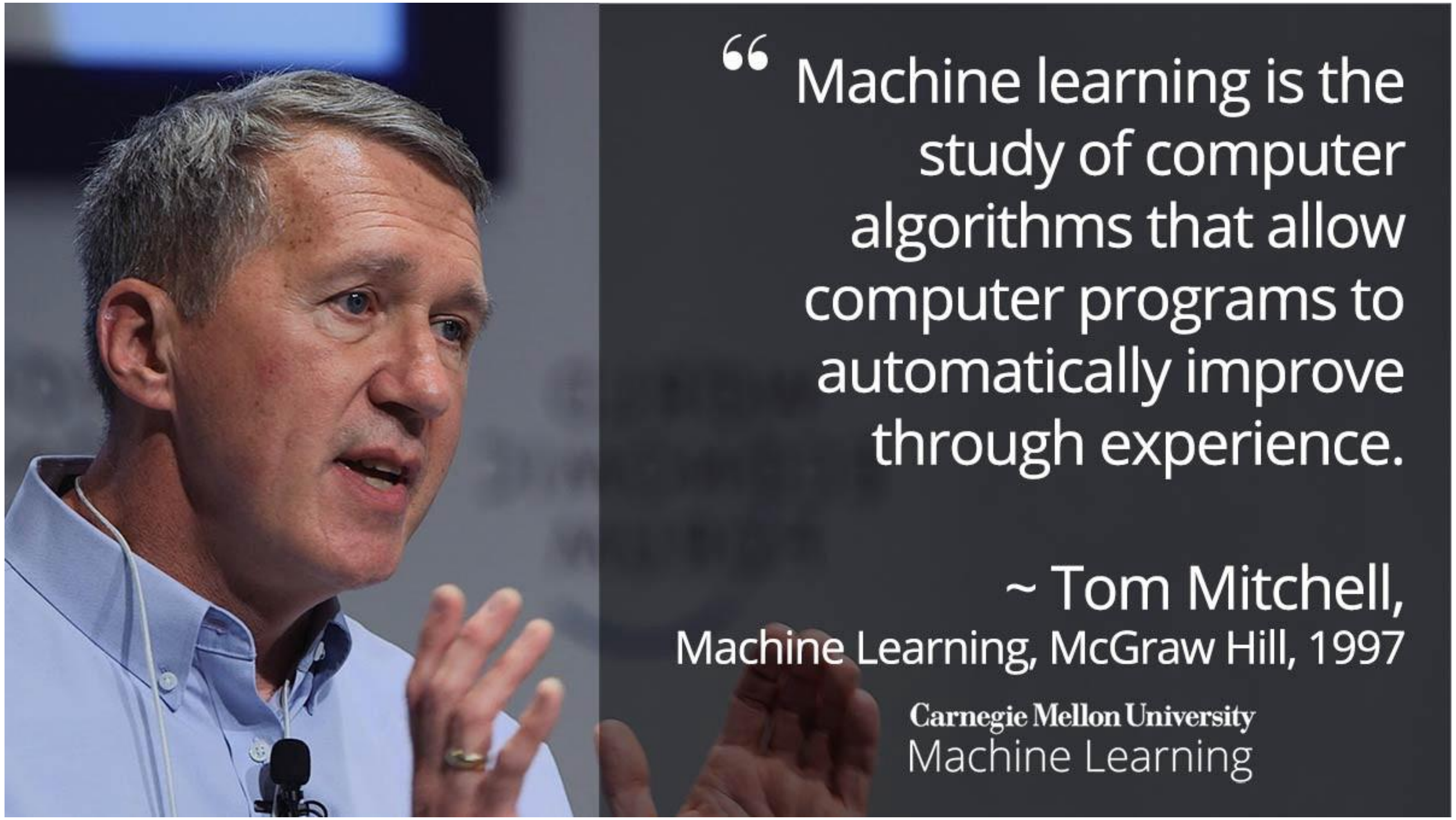
Another popular definition:

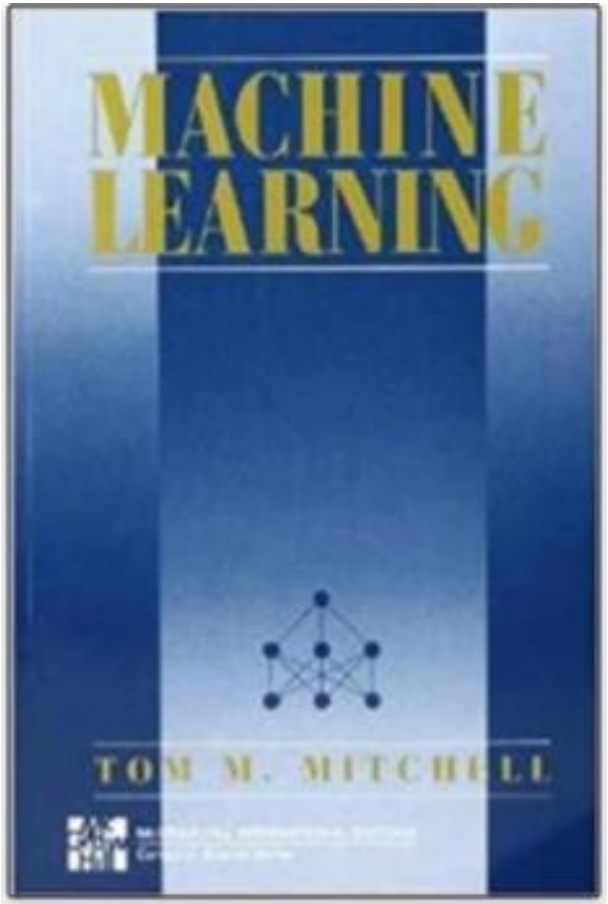
- *"A computer is said to **learn** from experience **E** with respect to task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improved with experience **E**"*
    - – Tom Mitchell (1997)

- Again, the key is learning from experience
- Not explicitly programmed



Machine Learning





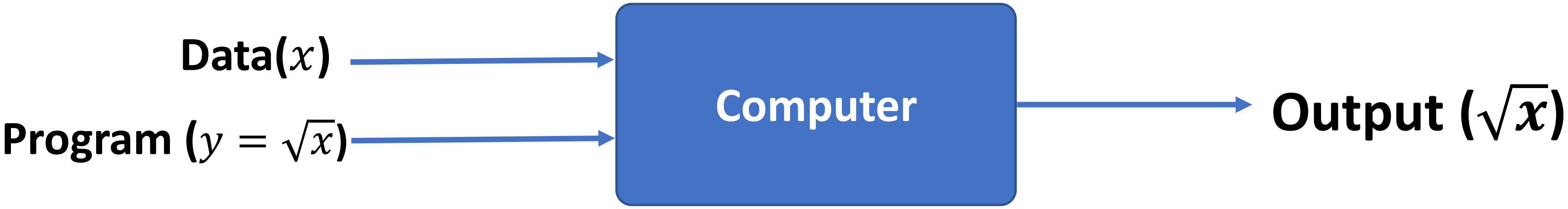
Spam or not SPAM?

Lancaster University

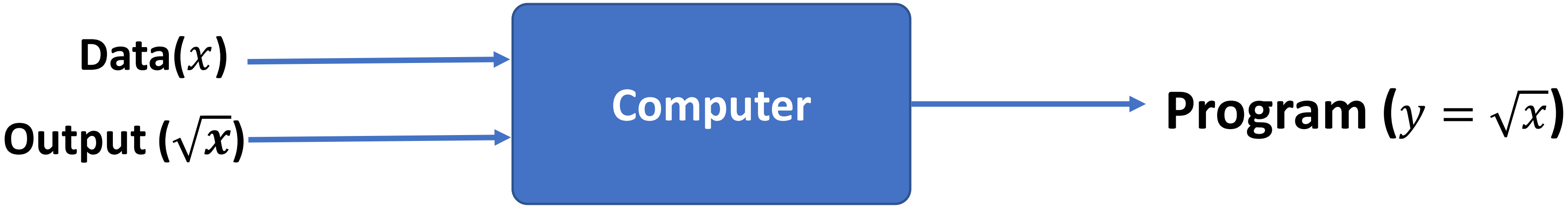
- *Given this definition:*
  - *A computer is said to **learn** from experience **E** with respect to task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improved with experience **E**”*
- My email program watches me mark some emails as spam, and improves on filtering spams. What is the T, E and P in the setting?
  a. Watching me label emails as spam
  b. Classifying emails as spam or not spam
  c. The fraction of emails correctly classified as spam or not
  d. None of the above – this is not a machine learning problem

Software 2.0

- Consider the function $y = f(x)$ (e.g. $y = \sqrt{x}$)
- Traditional Programming (**Software 1.0**)

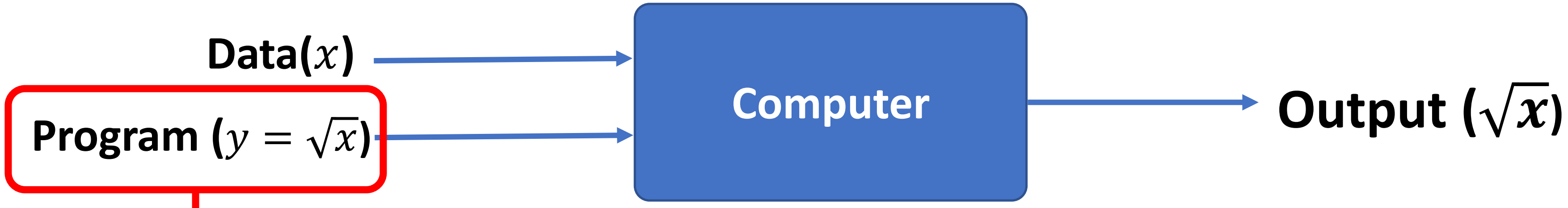


- Machine Learning (**Software 2.0**)

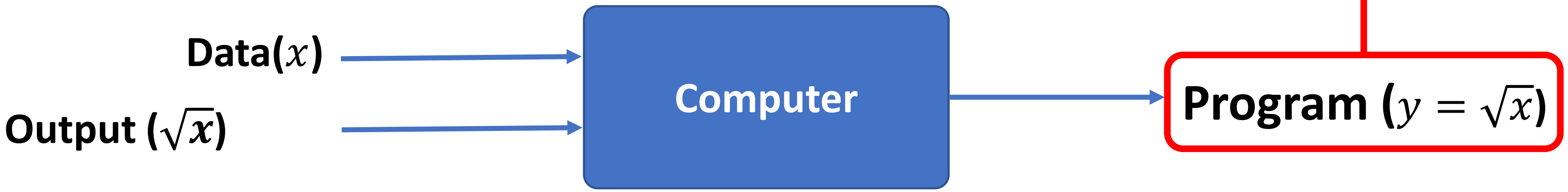


Software 2.0

- Consider the function $y = f(x)$ (e.g. $y = \sqrt{x}$)
- Traditional Programming (**Software 1.0**)



- Machine Learning (Software 2.0)



How things are learned

- Memorization
  - Accumulation of individual facts
  - Limited by
    - Time to observe facts
    - Memory to store facts

Declarative knowledge

How things are learned

- Memorization
  - Accumulation of individual facts
  - Limited by
    - Time to observe facts
    - Memory to store facts
- Generalization
  - Deduce new facts from old facts
  - Limited by accuracy of deduction process
    - Essentially a predictive activity
    - Assumes that the past predicts the future

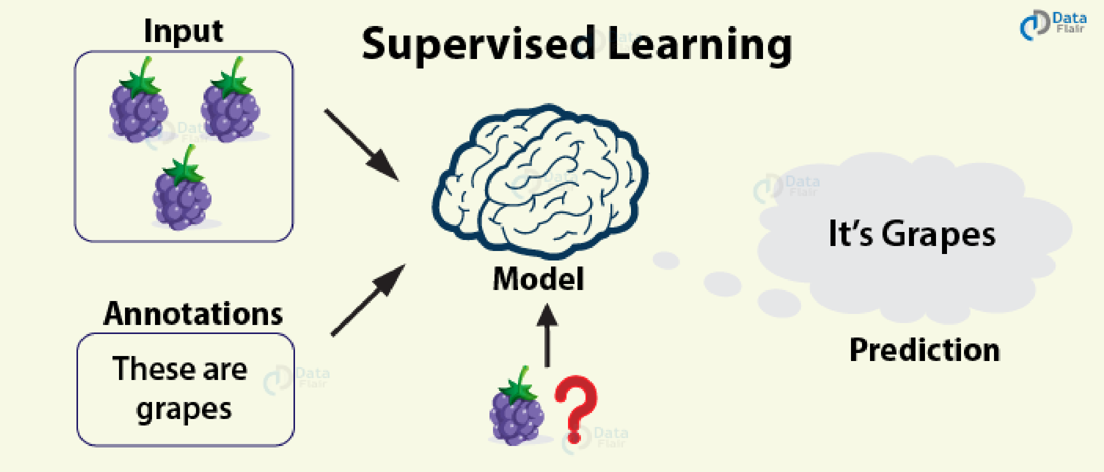
Declarative knowledge

Better

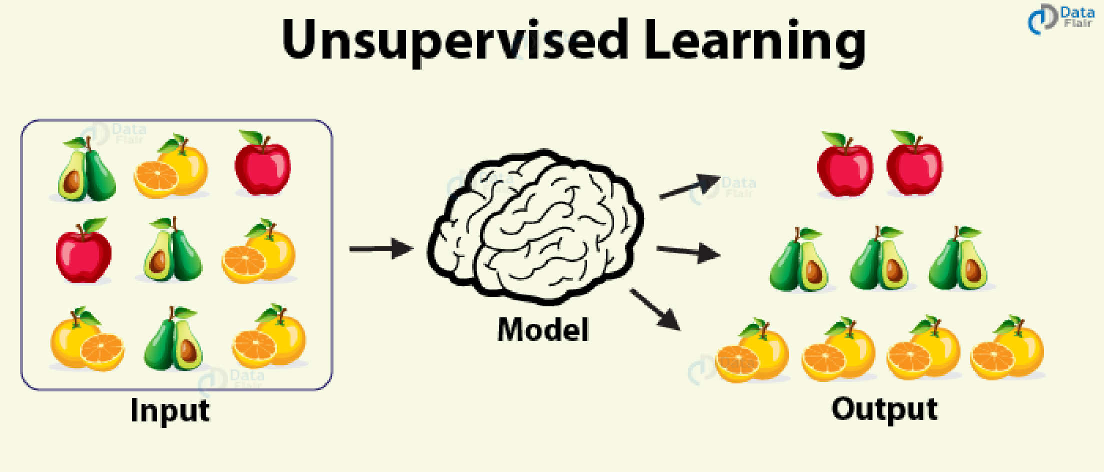
Imperative knowledge

Types Machine Learning

- Supervised
  - Classification
  - Regression

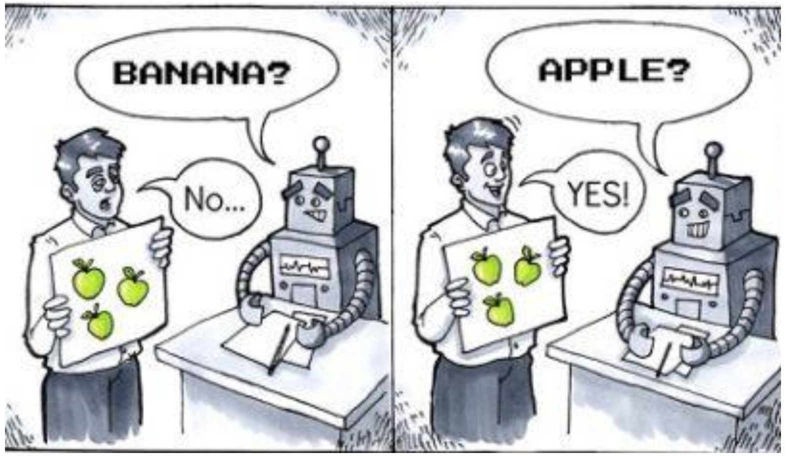


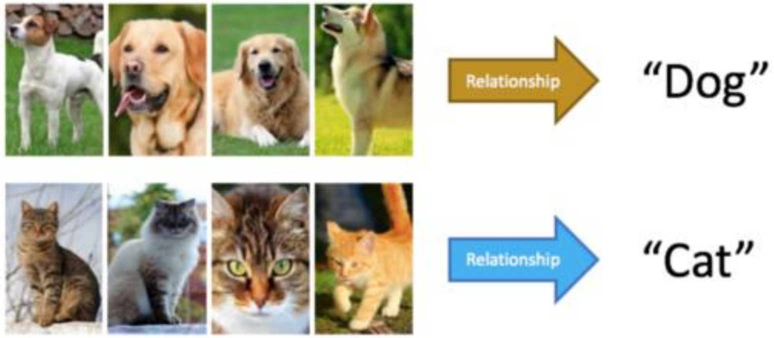
- Unsupervised
  - Clustering
  - Association



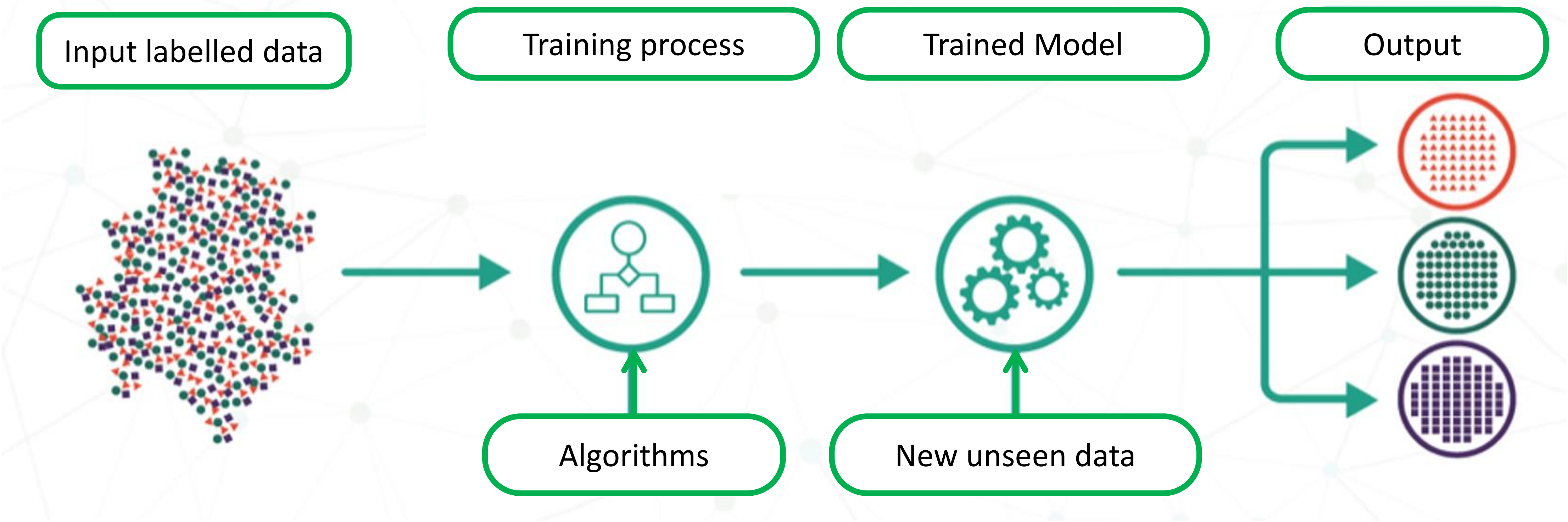
Supervised Learning

- The algorithm learns to map **an input** to a **particular output**.
- Instances of data are presented along with their **correctly labelled** output
- Similar to a **teacher-student** scenario
- The algorithm learns from **experience** to predict new unseen data
- Two broad categories:
  - Regression
  - Classification



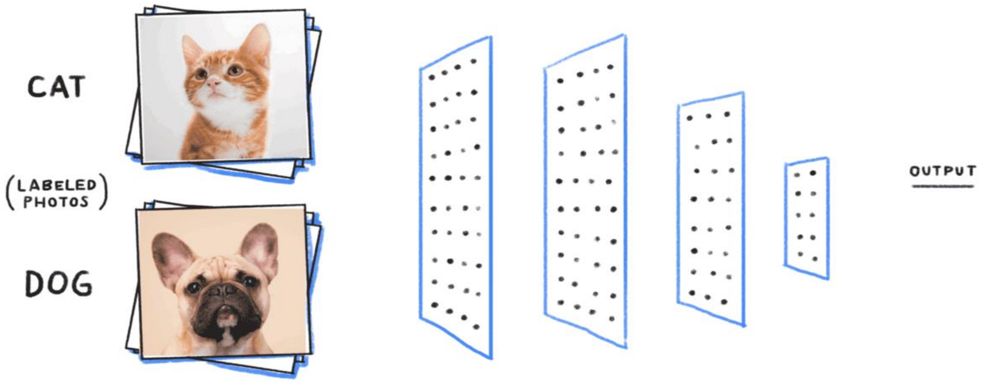


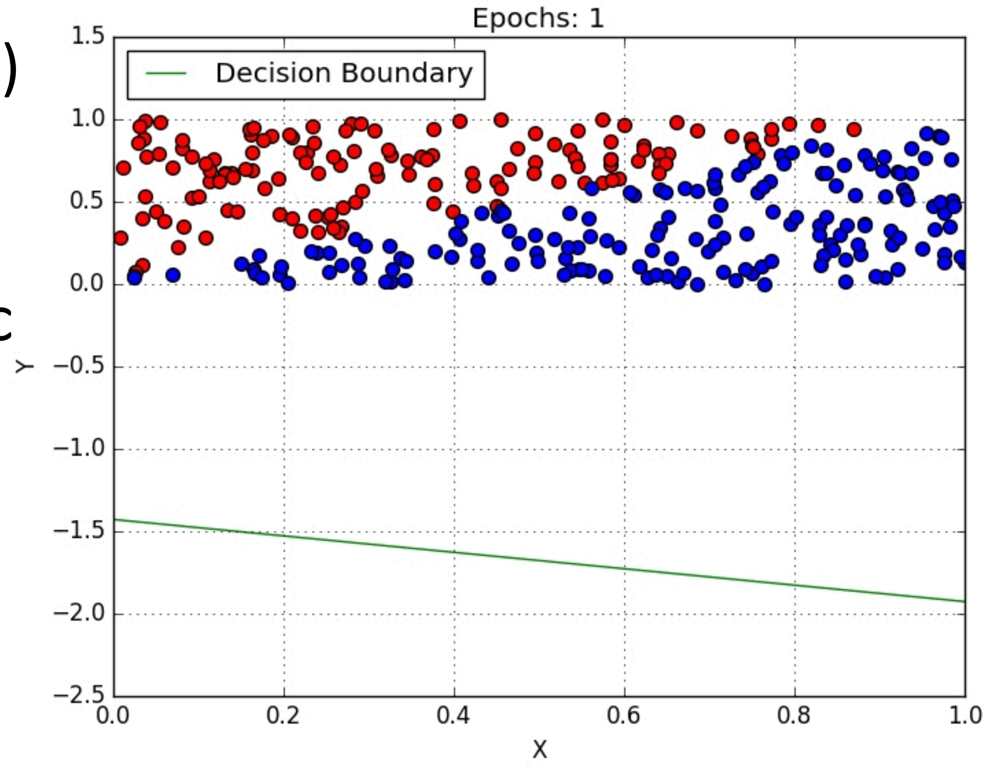
Supervised Learning



Classification

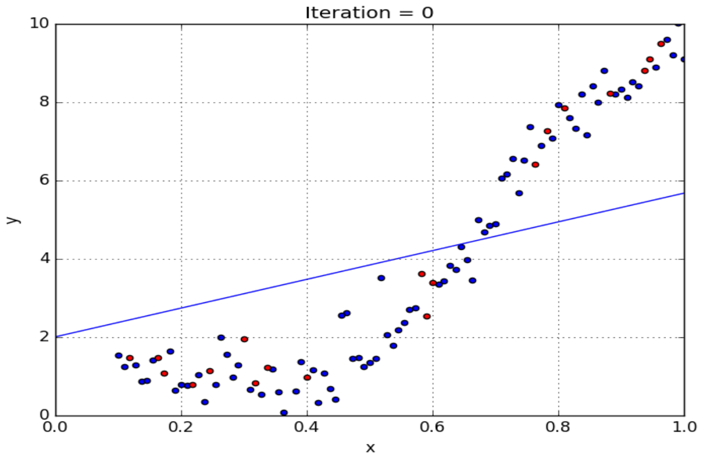
- Classification:
  - Also learns from labelled data (supervised)
  - Predicts a **category** or a **class**
    - Cats|Dogs, Spam|Ham, Cancer|Not Cancer
- Attempts to separate the data into specific categories (or classes or labels)

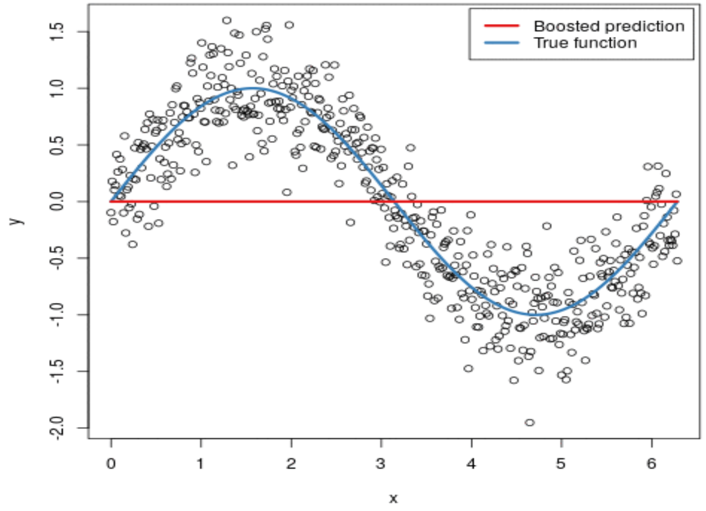




Regression

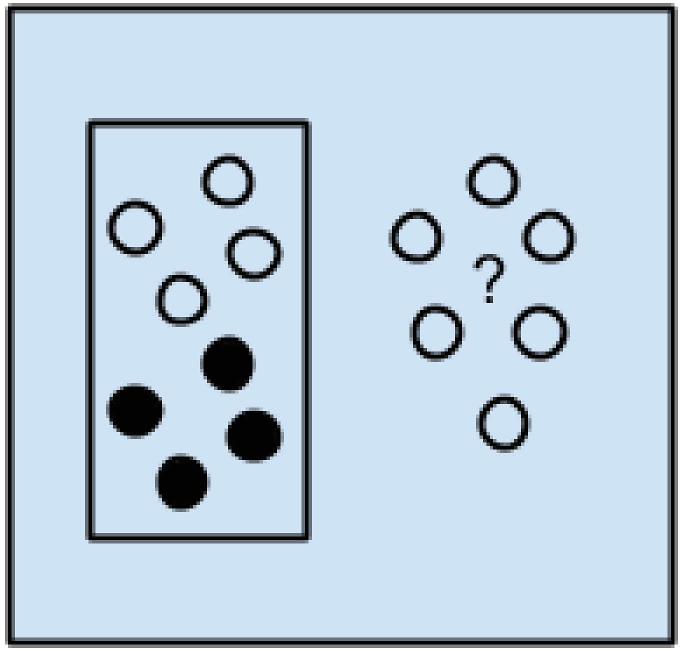
- Learns from labelled data (supervised)
- Predicts a **continuous-valued output**
  - height, price, duration etc.
- Consider a function $y = f(x)$
  - we want our model to predict y_i given x_i
  - x_i not seen during training
- Typically fits some linear or quadratic curve of the data plot
- Linear or logistic regression algorithms are often used



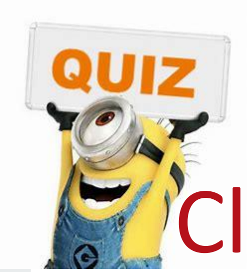


Supervised Learning Algorithms

- Input data = training data
  - with labels e.g. spam/ham or stock price at *t*
- In training
  - the model makes a prediction and is corrected if the prediction is wrong
- Training process continues until a desired accuracy is achieved
- Problem types: Classification and Regression
- **Algorithms**:
  - Logistic Regression
  - Back Propagation Neural Network.



Supervised Learning Algorithms

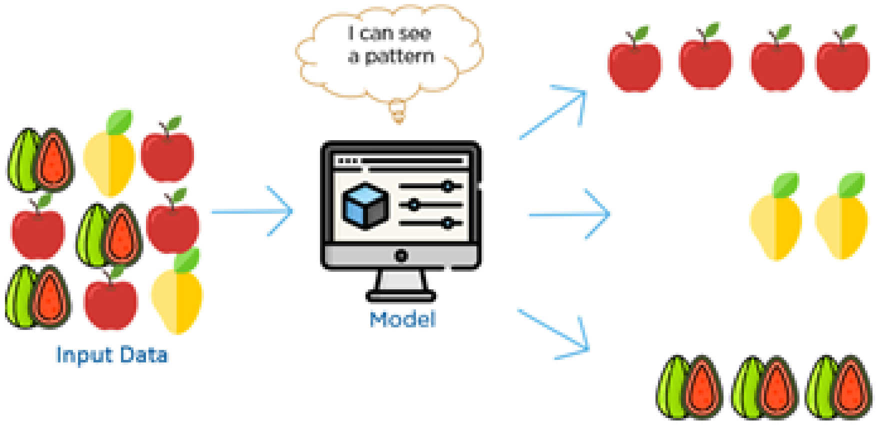


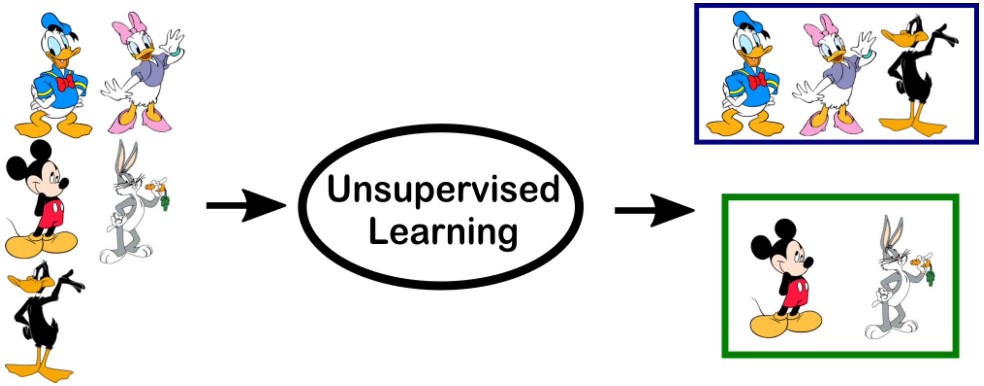
Classification vs Regression

- If we wish to learn models to address the following
  1. Predict how many students will enrol in this module in the next 3 years given the past enrolment data
  2. Predict whether a student will pass the module given previous years records
- How should we proceed
  a. Both are regression problems
  b. Both are classification problems
  c. Problem 1 is regression while Problem 2 is classification
  d. Problem 2 is regression while Problem 1 is classification

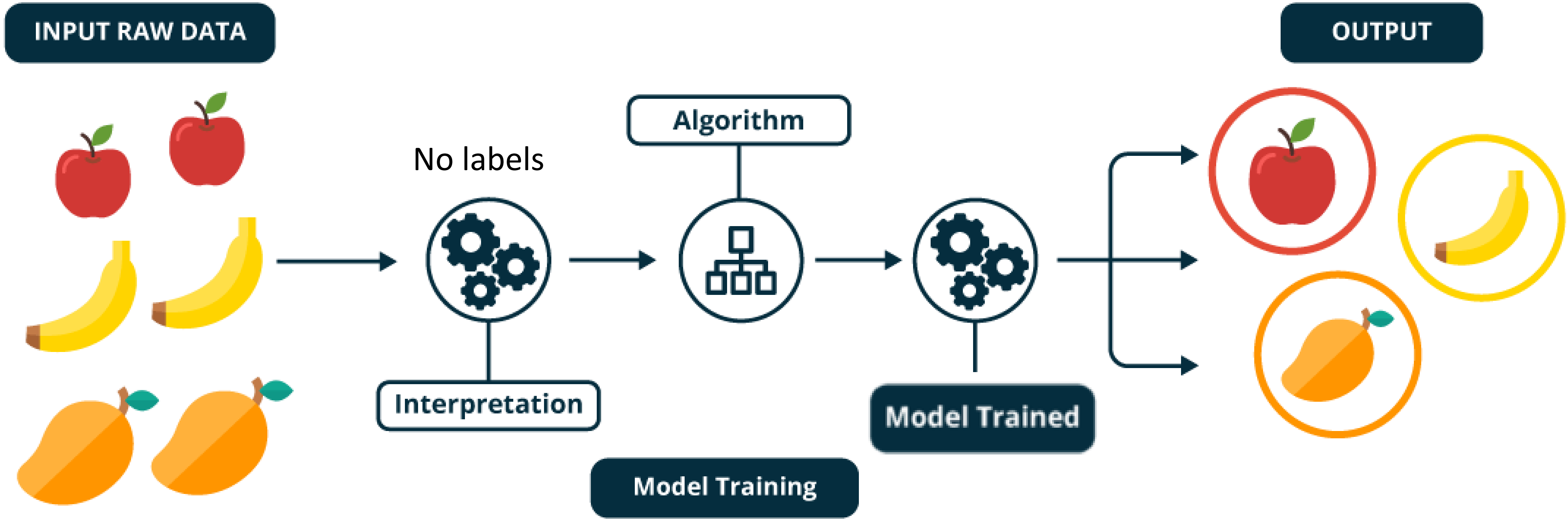
Unsupervised Learning

- Remember the function $y = f(x)$
- With unsupervised learning, only the input data, x, is available
- There are no corresponding labels (classes or categories) i.e. no output variable, y
- Aims at modelling the underlying structure of the data
- Two main categories
  - Clustering
  - Association





Unsupervised Learning

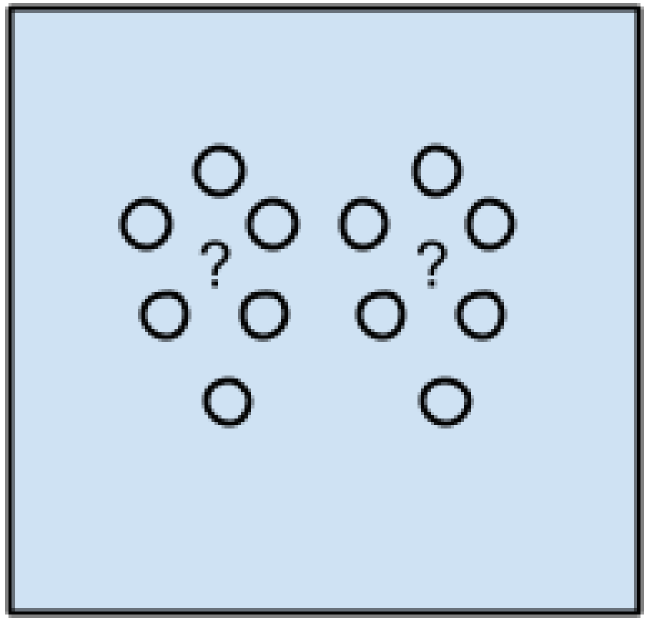


Clustering and Association

- In a clustering problem, we want to discover the inherent groupings in the data:
  - Eg: grouping customers by purchasing behaviour.
- In an association rule learning problem, we want to discover rules that describe large portions of your data
  - e.g. people that buy $\boldsymbol{X}$ also tend to buy $\boldsymbol{Y}$

Unsupervised Learning Algorithms

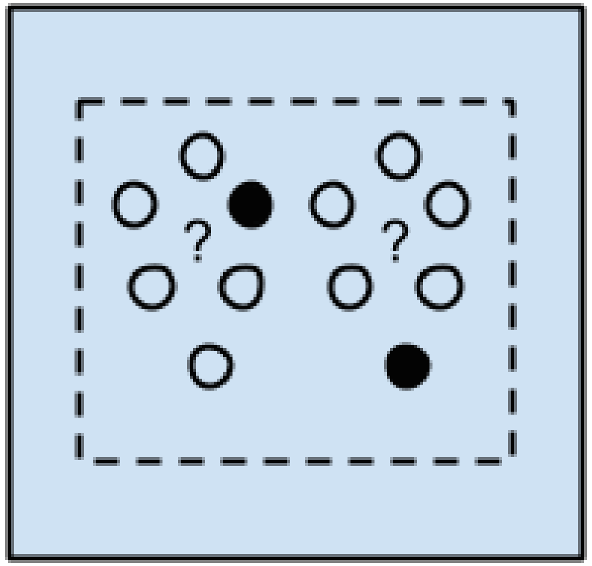
- Input data in not labelled
  - Output not known
- In training
  - Deduces structures present in the input data
  - Extracting general rules, reducing redundancy or organise data by similarity
- Problem types: clustering, dimensionality reduction and association rule learning
- **Algorithms**:
  - K-Means algorithm
  - Apriori algorithm.



Unsupervised Learning Algorithms

Semi-supervised Learning

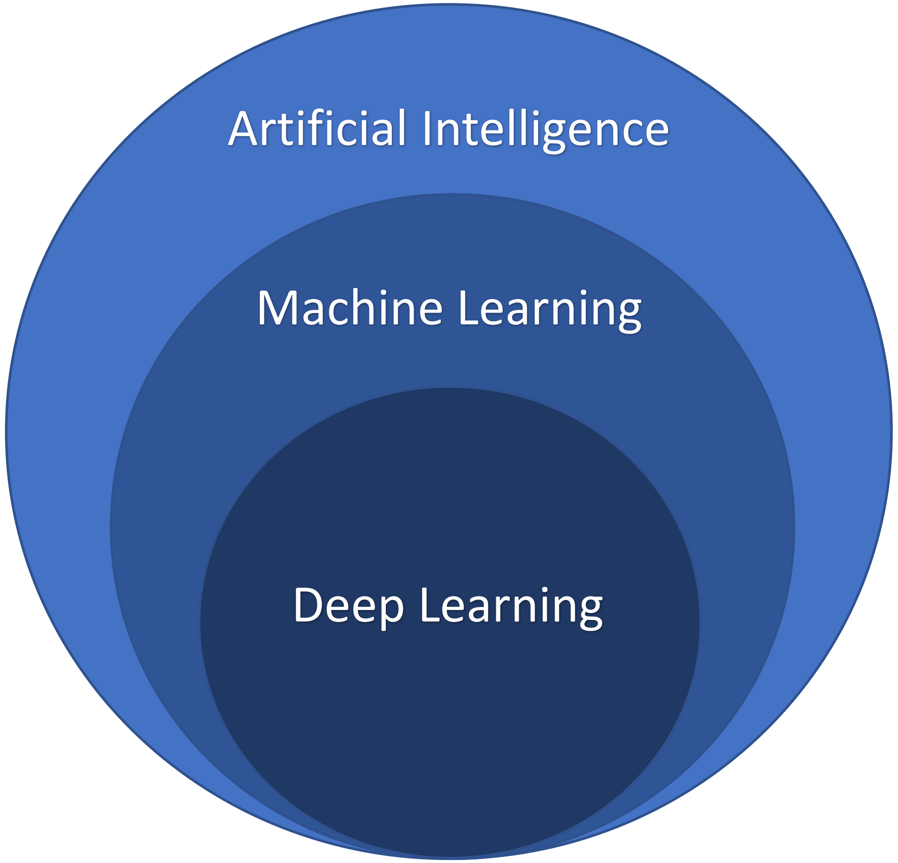
- Semi-supervised learning approach refers to:
  - when we have a large amount of input data (X) but **only some** of the data is labelled (Y)
  - e.g. a photo archive where only some of the images are labelled, (say *dog*, *cat*, *person*) and the majority are unlabelled.
- Many real world problems adopt this method
  - It can be expensive or time-consuming to label data
  - A hybrid design often helps to bridge the gaps
- Algorithms:
  - A flexible combination of supervised and unsupervised algorithms



Semi-supervised Learning Algorithms

AI and Machine Learning

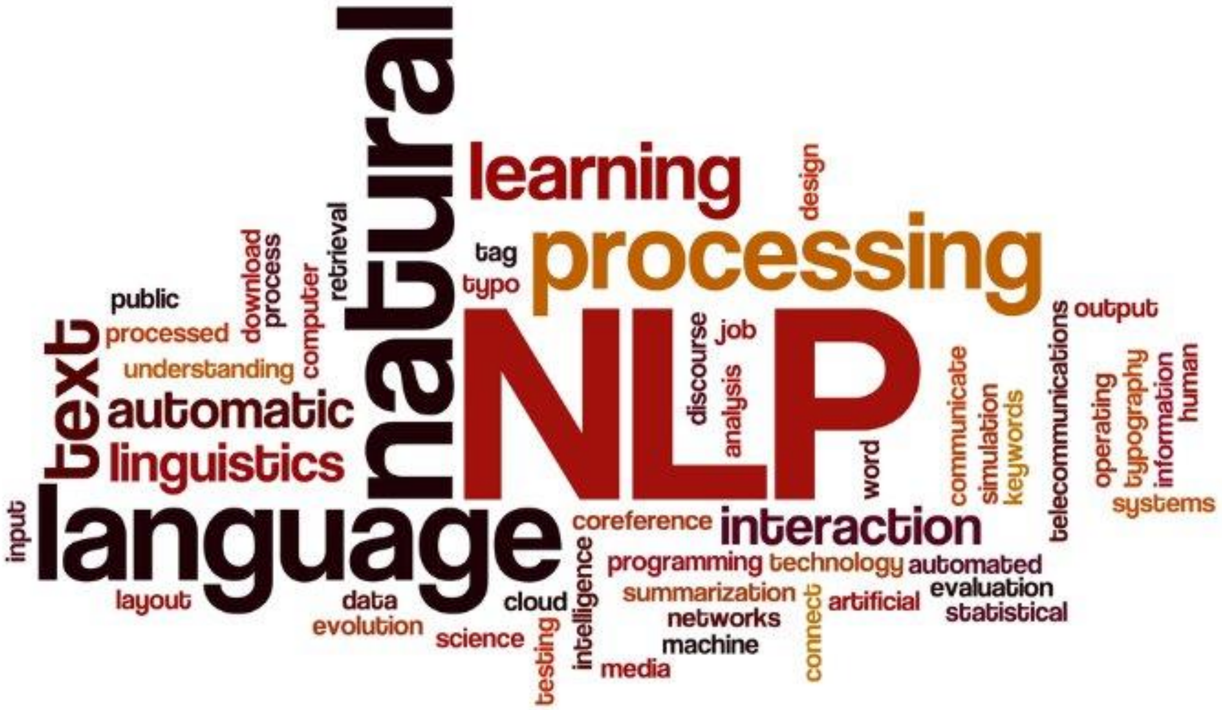
- AI systems were mostly **rule-based**
  - i.e. depended on hand-crafted rules
- Machine learning drives AI
  - Learning algorithms create a logical mapping from data to output
- Deep learning:
  - a subset of ML with additional layers to learn deeper representations data



10 minutes break & Question Time

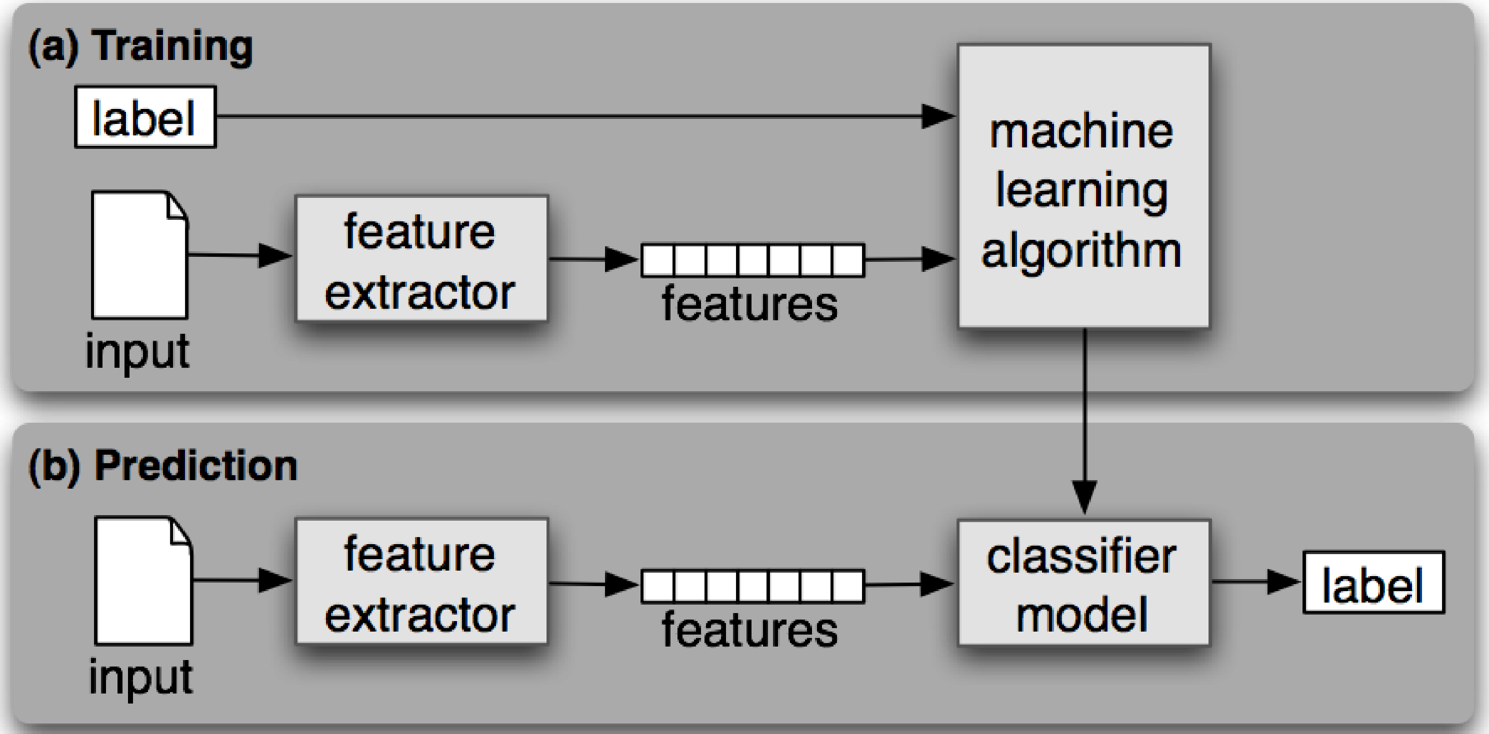
Next: Machine Learning for NLP

Machine Learning for NLP



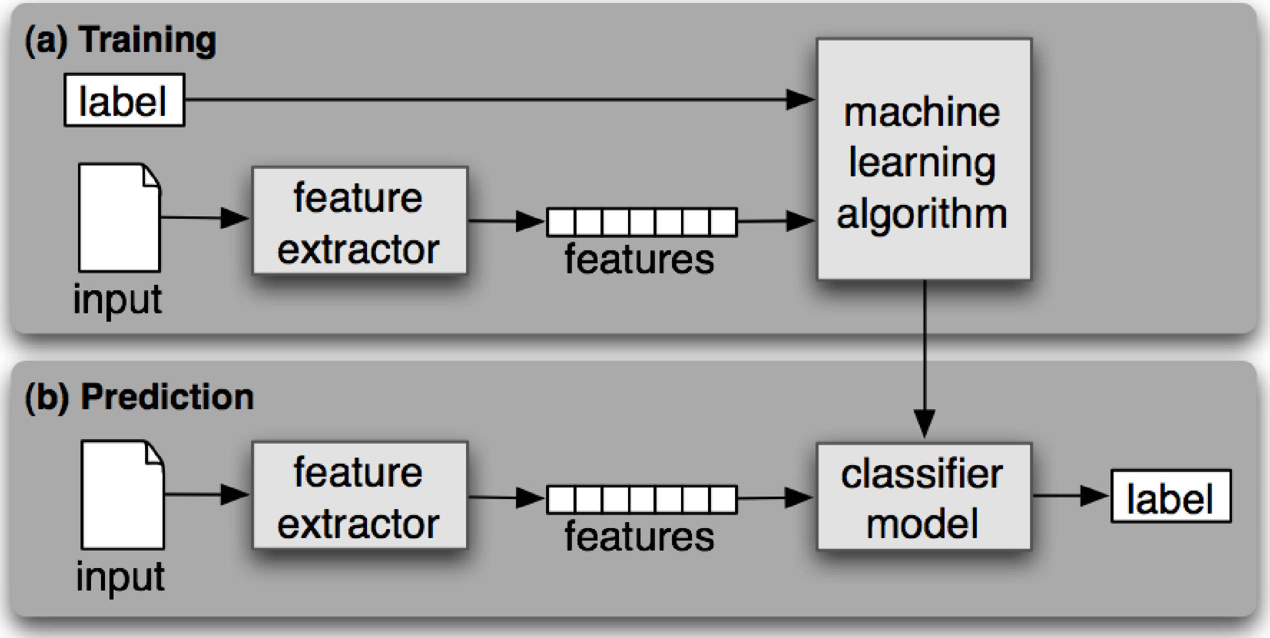
[Machine Learning RoadMap YouTube]

Supervised ML for Text



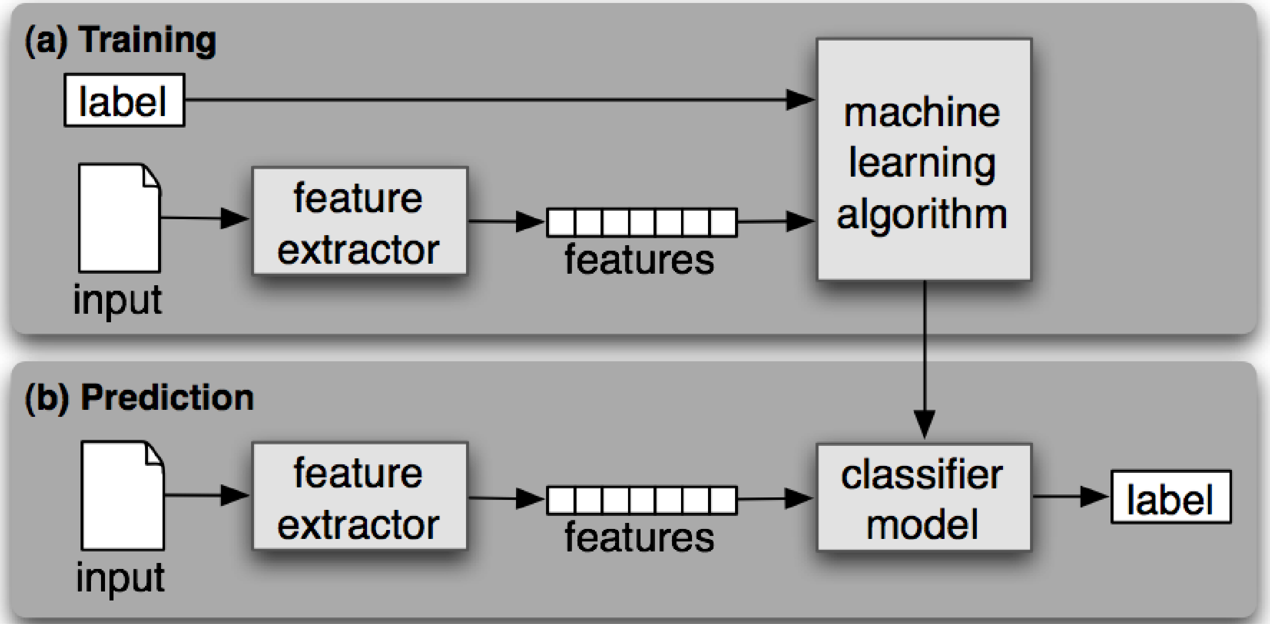
Supervised ML for Text

- Key components of the supervised ML include
- input (**training**) data (instances)
- Correct **labels**
- **Feature extractor**
- Machine **learning algorithm**
- Classifier **model**



Supervised ML for Text

- Key components of the supervised ML include
- input (**training**) data (instances)
- Correct **labels**
- **Feature extractor**
- Machine **learning algorithm**
- Classifier **model**



Feature Extraction

Lancaster University

- **Features** are the key ingredients for creating data instances for training machine learning models
- **Feature extraction** in NLP involves detecting patterns in text that can help in building an accurate model.
  - **POS-tagging:** Words ending in *-ed* tend to be past tense verbs.
  - **Document classification:** Frequent use of *will* is indicative of news text

Feature Extraction

- A **feature extractor** is a function that converts each input value to a *feature set.*
- Choosing simple features often gives good results but careful crafting the features can improve the result significantly
- Wrong features could lead to poor performance.
- Too many features could lead to **overfitting**.

download

```
>>> import nltk >>> nltk.download()
```

What’s in a name?

- The NLTK data contains the **Name** corpus, a collection of about 8k male and female names. Below are 10 of the names

Name	M/F	Name	M/F	Name	M/F	Name	M/F
Abbey		Eddie		Jaime		Nickie	
Barrie		Frank		Kellen		Ollie	
Cary		Gabriel		Lanny		Quentin	
Daniel		Haley		Maddie		Regan	

- Could you tell what gender each of the names is?

download

```
>>> import nltk >>> nltk.download()
```

What’s in a name?

Lancaster University

- The NLTK data contains the **Name** corpus, a collection of about 8k male and female names. Below are 10 of the names

Name	M/F	Name	M/F	Name	M/F	Name	M/F
Abbey		Eddie		Jaime		Nickie	
Barrie		Frank		Kellen		Ollie	
Cary		Gabriel		Lanny		Quentin	
Daniel		Haley		Maddie		Regan	

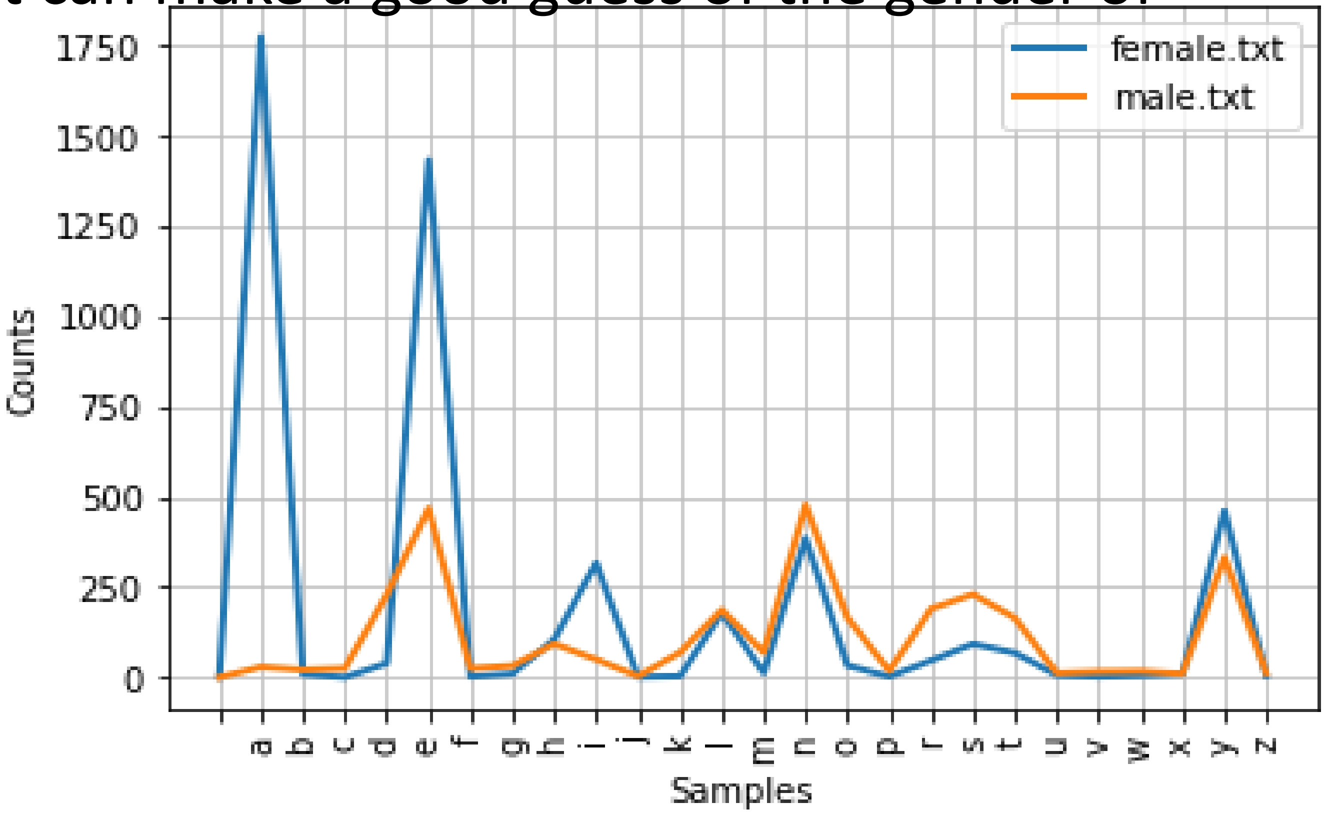
- Could you tell what gender each of the names is?
  - Answer: **Each name can be both male or female?**

Gender Identification

- How do we train a model that can make a good guess of the gender of a name?

Gender Identification

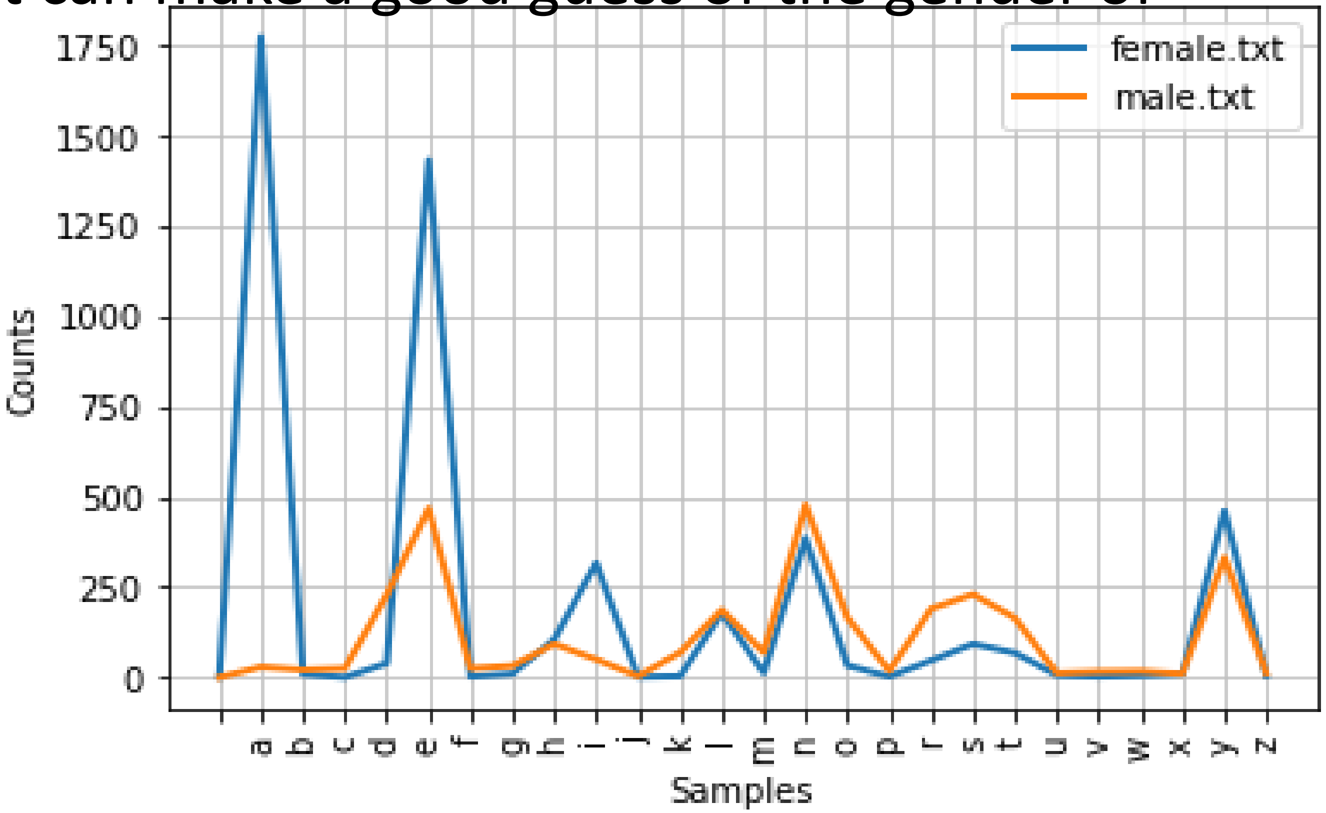
- How do we train a model that can make a good guess of the gender of a name?



Gender Identification

- How do we train a model that can make a good guess of the gender of a name?

- Female names more likely to end in **a**, **e** and **I**
- Male names more likely to end in **k, o, r, s** and **t**
- *Can we extract this feature for training the classifier?*



Demo – Importing the **names** corpus

- We can import `nltk` and download the `names` corpus

```
import nltk
import random
nltk.download('names')
names = nltk.corpus.names
```

Demo – Names in both lists

- We can also look at the names in both lists

```
print(names.fileids())
male_names = names.words('male.txt')
female_names = names.words('female.txt')
male_female = [w for w in male_names if w in female_names]
print(len(male_female))
for name in male_female:
  print(name)
```

Demo – Distribution of last letters

- We can plot the distribution of last names

```
cfd = nltk.ConditionalFreqDist(
        (fileid, name[-1])
        for fileid in names.fileids()
        for name in names.words(fileid))
cfd.plot()
```

Demo – Distribution of last letters

- We can plot the distribution of last names

```
cfd = nltk.ConditionalFreqDist(
    (fileid, name[-1])
    for fileid in names.fileids()
    for name in names.words(fileid))
cfd.plot()
```

Demo – Create Feature Extractors (1)

- We can create the feature extractor function **get_features()**

```
# feature extractor 1
def gender_features(word):
  return {'last_letter': word[-1]}
```

Demo – Create Feature Extractors (2)

- We can create the feature extractor function **get_features2()**

```
# feature extractor 2
def gender_features2(name):
    features = {}
    features["first_letter"] = name[0].lower()
    features["last_letter"] = name[-1].lower()
    for letter in 'abcdefghijklmnopqrstuvwxyz':
        features["count({})".format(letter)] = name.lower().count(letter)
        features["has({})".format(letter)] = (letter in name.lower())
    return features
```

Demo – Create Feature Extractors (3)

- We can create the feature extractor function **get_features3()**

```python
# feature extractor 3
def gender_features3(word):
  return {'suffix1': word[-1:], 'suffix2': word[-2:]}
```

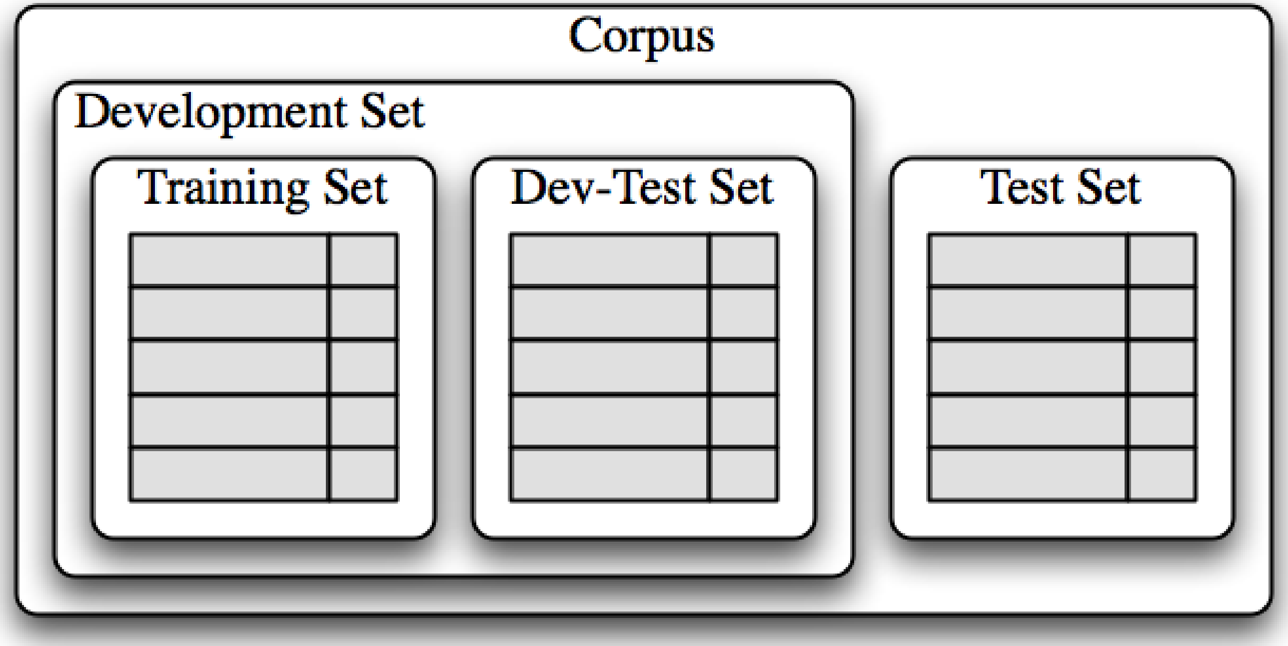
Demo – Building the Training Data

- We can combine and shuffle the list to build the training data

```python
labeled_names = ([(name, 'male') for name in names.words('male.txt')]
                 + [(name, 'female') for name in names.words('female.txt')])
random.shuffle(labeled_names)
# len(labeled_names)
```

Demo – Train:Dev:Test Split

- We need to split our data into the training, development and testing sets



Demo – Train:Dev:Test Split

- We need to split our data into the training, development and testing sets

```
# train-devtest-test split
train_names = labeled_names[1500:]
devtest_names = labeled_names[500:1500]
test_names = labeled_names[:500]
```

Demo – Extracting Features from Data

- We need to apply the feature extractor to each of the data splits
- We also have **gender_features2()** and **gender_features3()**

```
# Extracting the features
train_set = [(gender_features(n), gender) for (n, gender) in train_names]
devtest_set = [(gender_features(n), gender) for (n, gender) in devtest_names]
test_set = [(gender_features(n), gender) for (n, gender) in test_names]
```

Demo – Training and Testing

- We train the classifier with the **train_set** using the **nltk.NaiveBayesClassifier.train()** function
- We also test the classifier on the **devtest_set**

```
# Training the classifier
classifier = nltk.NaiveBayesClassifier.train(train_set)

# apply the classifier to the development test
print(nltk.classify.accuracy(classifier, devtest_set))
```

Online Demo

- You can explore and extend these concepts on the Colab Jupyter Notebook below:
- https://github.com/IgnatiusEzeani/NLP-Lecture/blob/main/Week_18_Lecture_Demo.ipynb

Online Demo

- https://github.com/IgnatiusEzeani/NLP-Lecture/blob/main/Week_18_Lecture_Demo.ipynb

Coming next…

- Flexible Labs (Wednesday and Thursday):
  - Other ML text classification tasks
  - Support for course work
- Week 19 Lectures
  - More complex feature extraction from texts
  - Deep Neural Networks for Text Processing
  - Automatic feature extraction and model training with Neural Networks

Thank you for attending, any questions?