The background of the slide is a blurred photograph of a desk. In the center, a pair of black-rimmed glasses rests on an open notebook. To the left of the glasses, a red ribbon is tied in a loop. The overall lighting is soft and focused on the desk items.

Data Science

Class 3: NLP project

US – Embassy



Table of content

- NLP and its application
- Basic concepts
- EDA and ETL techniques in NLP
- Machine Learning models
- Deep Learning models

NLP & applications

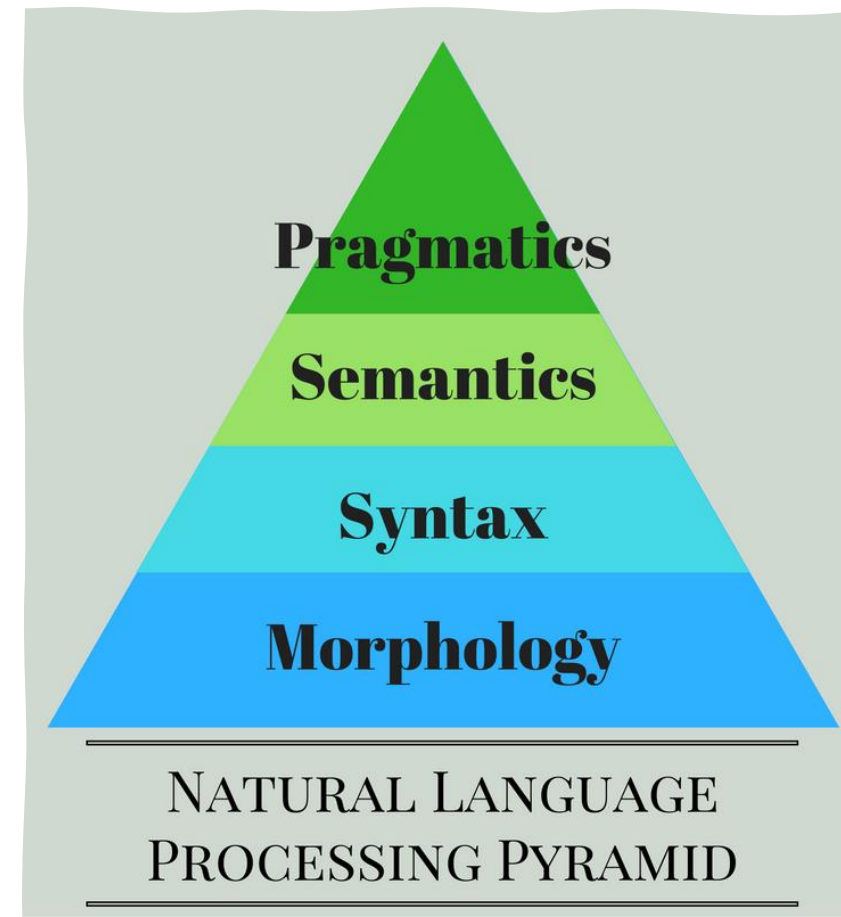
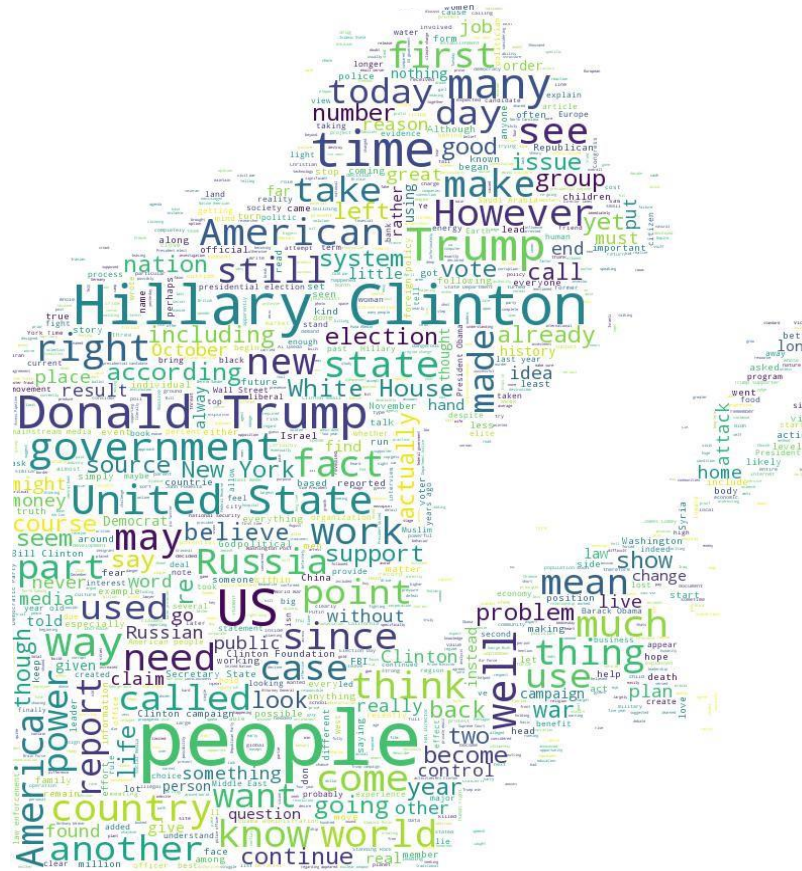
Natural language processing (NLP) is a subfield of

- Linguistics,
- Computer science, and
- AI (artificial intelligence)

concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. [[Wikipedia](#)]

- Linguistics,
- Computer science, and
- AI(artificial intelligence)

concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. [[Wikipedia](#)]



Concept 1.

CountVectorizer

- **CountVectorizer** is a great tool provided by the [scikit-learn library](#) in Python.
- It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.



CountVectorize (cts.)

Advantage

- Easy to compute and understand

Disadvantage

- It also doesn't identify the relationships between words such as linguistic similarity between words.
- When text-dataset has a huge-unique vocabularies, the shape of data will be increased.

```
corpus = ['this is the first document',  
          'this document is the second document',  
          'and this is the third one',  
          'is this the first document?',  
          'this Document is not yours..']
```

```
dvnt.view_word_freq(corpus)
```

```
*=====
|There are 5 sentences in this corpus.
|=====
|The number of the different words is 11, and ... they are:
|=====
*      1: and,
*      2: document,
*      3: first,
*      4: is,
*      5: not,
*      6: one,
*      7: second,
*      8: the,
*      9: third,
*     10: this,
*     11: yours,
```

	and	document	first	is	not	one	second	the	third	this	yours
this is the first document	0	1	1	1	0	0	0	1	0	1	0
this document is the second document	0	2	0	1	0	0	1	1	0	1	0
and this is the third one	1	0	0	1	0	1	0	1	1	1	0
is this the first document?	0	1	1	1	0	0	0	1	0	1	0
this Document is not yours..	0	1	0	1	1	0	0	0	0	1	1

Concept 2. Tfidf Vectorizer

- TF-IDF means Term Frequency - Inverse Document Frequency.
- This is a statistic that is based on the frequency of a word in the corpus but it also provides a numerical representation of how important a word is for statistical analysis.

TFIDF

For a term i in document j :

$$w_{i,j} = tf_{i,j} \times \log \left(\frac{N}{df_i} \right)$$

$tf_{i,j}$ = number of occurrences of i in j
 df_i = number of documents containing i
 N = total number of documents



Tfidf-Vectorizer

Advantage

- TF-IDF is better than Count Vectorizers because it not only focuses on the frequency of words present in the corpus but also provides the importance of the words.

Disadvantage

- It only focuses on the frequency of occurrence of a word, leading to it having almost no contextual meaning.

```
dvnt.textdata_to_countvect(df, 'my_text')
```

	my_text	and	document	first	is	not	one	second	the	third	this	yours
0	this is the first document	0	1	1	1	0	0	0	1	0	1	0
1	this document is the second document	0	2	0	1	0	0	1	1	0	1	0
2	and this is the third one	1	0	0	1	0	1	0	1	1	1	0
3	is this the first document?	0	1	1	1	0	0	0	1	0	1	0
4	this Document is not yours..	0	1	0	1	1	0	0	0	0	1	1

```
dvnt.textdata_to_TFIDF(df, 'my_text')
```

	my_text	and	document	first	is	not	one	second	the	third	this	yours
0	this is the first document	0.000000	0.427120	0.611659	0.361255	0.000000	0.000000	0.000000	0.427120	0.000000	0.361255	0.000000
1	this document is the second document	0.000000	0.646126	0.000000	0.273244	0.000000	0.000000	0.573434	0.323063	0.000000	0.273244	0.000000
2	and this is the third one	0.514923	0.000000	0.000000	0.245363	0.000000	0.514923	0.000000	0.290099	0.514923	0.245363	0.000000
3	is this the first document?	0.000000	0.427120	0.611659	0.361255	0.000000	0.000000	0.000000	0.427120	0.000000	0.361255	0.000000
4	this Document is not yours..	0.000000	0.338411	0.000000	0.286226	0.600678	0.000000	0.000000	0.000000	0.000000	0.286226	0.600678



Using **co-occurrence matrix** can solves that problem partially.

- Co - occurrence matrix, is a symmetric square matrix, each row or column will be the vector representing the corresponding word.
- The context can be defined as a document or a window within a collection of documents, with an optional vector of weights applied to the co-occurrence counts.

```
corpus = ["I love you and football",
          "Dont worried! :D just kidding www.google.com.vn",
          "I love statistics :v",
          "I love Machine Learning and Mathematics"]
dvnt.get_co_occurrence_matrix(corpus)
```

[illegible]

Concept 3. Co-occurrence matrix (cts)

Advantage.



- It preserves the semantic relationship between words.
- It uses SVD (singular value decomposition) at its core to reduce the size of vector, which produces more accurate word vector representations than existing methods.
- It uses factorization which is a well-defined problem and can be efficiently solved.
- It has to be computed once and can be used anytime once computed. In this sense, it is faster in comparison to others.

Disadvantage

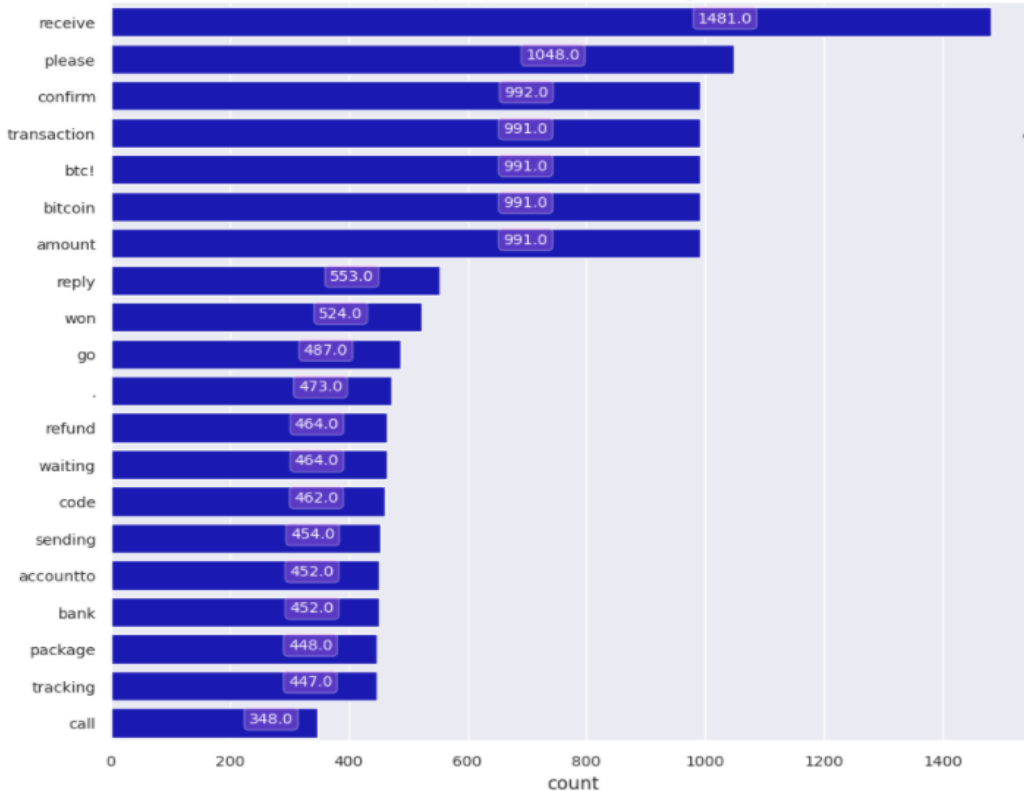


- The disadvantage of **Co-occurrence Matrix (CM)** is when the text_data contains a large numbers of vocabularies; hence it requires huge memory to store the co-occurrence matrix.
- To make the representation of words clearer and save memory, choose or remove some unnecessary words (such as **stopword**

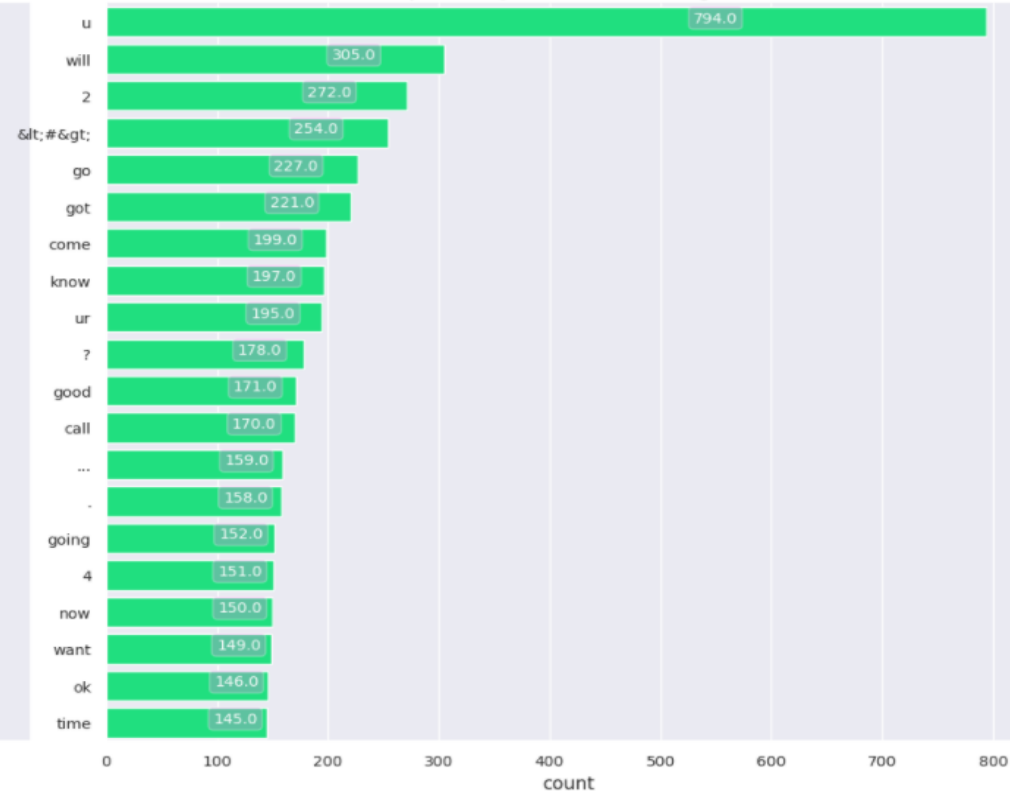
Concept 4. N-grams analytics

- N-grams is a contiguous sequence of **n items** from a given sample of text or speech. The *items* can be *phonemes, syllables, letters, words* or *base pairs* according to the application.

Top 20 spam most common.unigram



Top 20 non-spam most common.unigram



Unigram


- please,
- receive,
- etc.

Bi-gram

- please confirm
- take care
- etc

Tri-gram

- please confirm transaction,
- you won 600\$,
- Etc.



Concept 4. N-gram analytics (cts)

Advantage.

- Contains all the information that the first N-1 words can provide. These words have a strong binding force on the appearance of the current word,
- The higher N-values, the more meaningful in the contextual.
- If the keyword weight is known to be very large, it may be appropriate to use the N-gram model.
- Easy to view the top-Ngrams to give the analytic-decision when doing ETL.

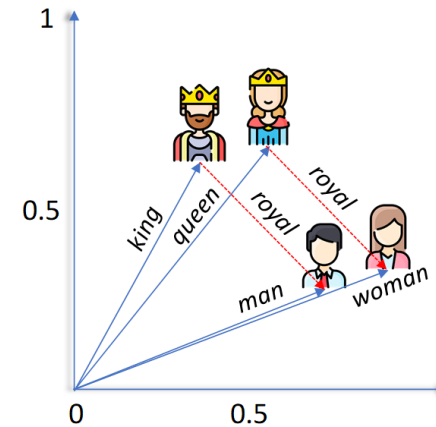
Disadvantage

- The disadvantage of **N-grams** is it requires a considerable amount of training text to determine the parameters of the model, So the common N value is generally 1,2.
- It is constructed based on discrete unit words that do not have any genetic attributes between each other, and thus does not have the semantic advantages satisfied by word vectors in a continuous space: **words of similar meaning have similar word vectors**, which is used as a system.

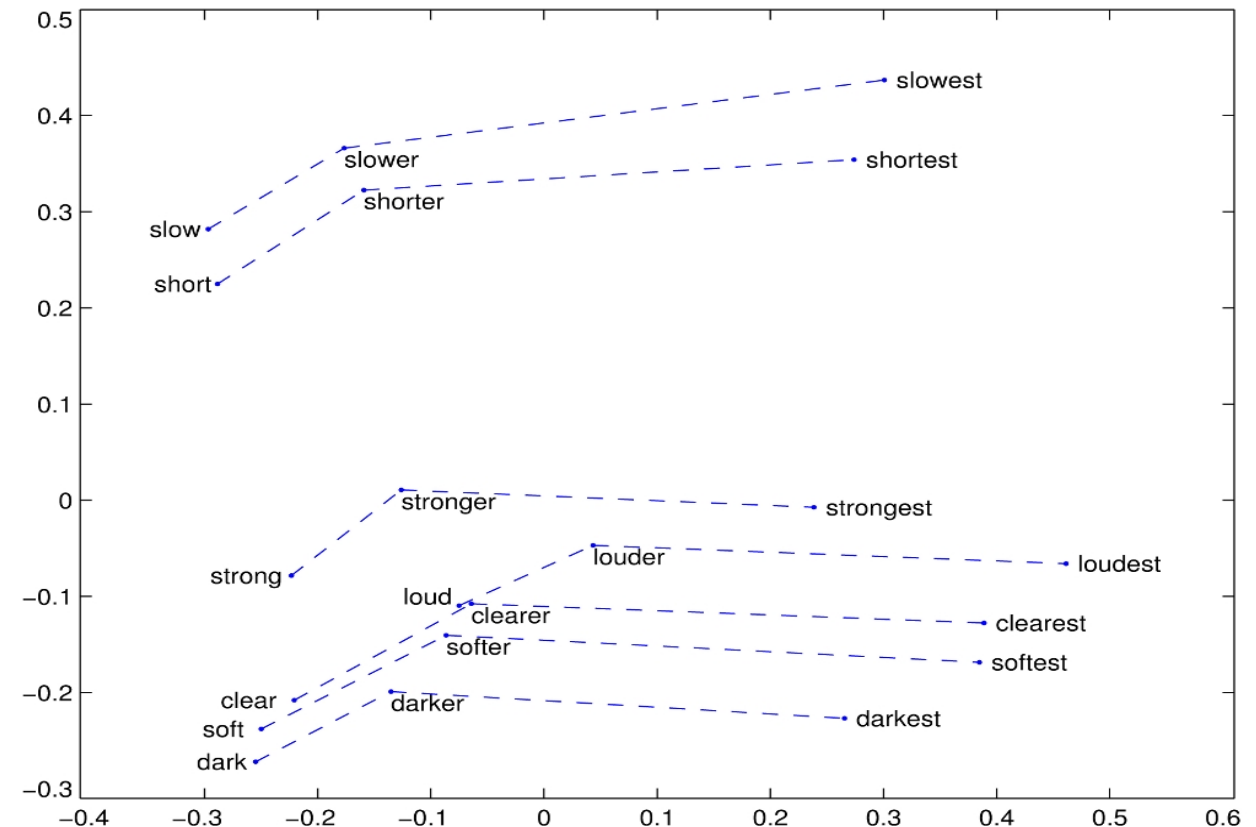
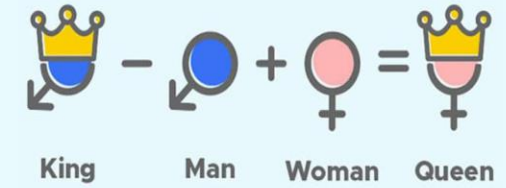
Concept 5.

Word2Vec & GloVe

- **Word2vec** is an unsupervised method based on the idea that similar words have similar neighbors.
- After Tomas Mikolov et al. released the [word2vec](#) tool, there was a boom of articles about word vector representations **GloVe: Global Vectors for Word Representation**



2vec



EDA and ETL techniques in NLP

Defintion

- EDA (see Class 1)
- ETL (extract transform & loading)

Why ETL?

- Help DA, DS work more efficiently
- Help an organization become more data driven.

```
text = ["click https://www.kaggle.com/ and https://seaborn.pydata.org. 123go to receive free-gift!! \U0001F600 Saigon 2022/12-01",
"@David Life is suffering, please came back! 12people in our class has been failed! \U0001F64F, please contact Prof.Elena",
"your update Verison has been rejected, we can't finish downloading your update! Please try again later",
"#memory, #Thanksgiving at #NewYork City!?!% 25 Nov 2021, Happy with @Alex @Ashley @David and @Charles! NY; 25 Nov, 2021",
"Good morning, how are you",
"123jerky =)) @LOL is feeling happy, #EA-Sport submitted to surname@hotmail.com",
"The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages"]
df = pd.DataFrame({'text': text})
%time dvnt.view_url_emoij_etc(df, 'text')
```

CPU times: user 48.5 ms, sys: 8.69 ms, total: 57.2 ms
Wall time: 55.8 ms

	text	wcount	unq_wcount	stop_word_count	url_count	avg_wlen	char_count	punct_count	hastg_count	mentn_count	max_wlen	emoji_cnt	capt_cnt	email_cnt	time_cnt	char&num_cnt
0	click https://www.kaggle.com/ and https://seab...	11	11	2	2	9.090909	110	17	0	0	27	1	2	0	1	1
1	@David Life is suffering, please came back! 12...	18	17	5	0	5.222222	111	6	0	1	10	0	4	0	0	1
2	your update Verison has been rejected, we can'...	16	15	7	0	5.437500	102	3	0	0	11	0	2	0	0	0
3	#memory, #Thanksgiving at #NewYork City!?!% 25 ...	19	18	3	0	5.263158	118	15	3	4	13	0	13	0	2	0
4	Good morning, how are you	5	5	3	0	4.200000	25	1	0	0	8	0	1	0	0	0
5	123jerky =)) @LOL is feeling happy, #EA-Sport ...	10	10	2	0	6.900000	78	9	1	2	19	1	6	1	0	1
6	The SMS Spam Collection is a set of SMS tagged...	38	29	14	0	4.710526	216	5	0	0	12	0	18	0	1	0

ETL pipeline in NLP

Extract

- **Source:**
 - csv,
 - json,
 - api, etc.
- **Elements:**
 - hashtag,
 - link,
 - emojiicon,
 - key-words, etc.

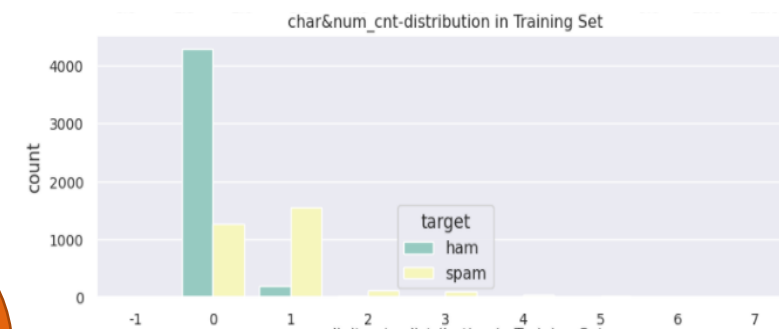
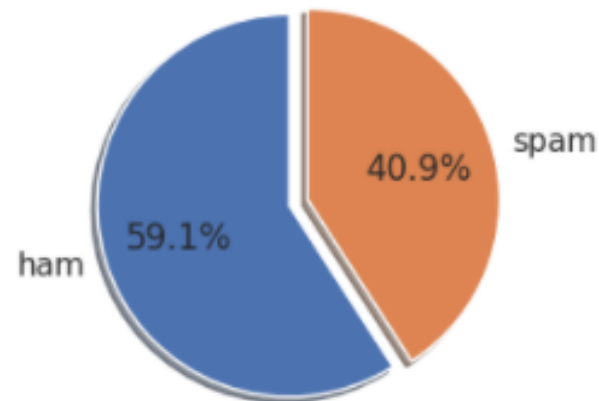
Transform

- Combining data from different-sources
- Cleaning, parsing date
- Encoding
- Filling miss-value
- Scaling
- Removing outliers
- Drop duplicates
- Feature engineering

Loading

- Send and save the transformed data to a database.

EDA results




target	1	0.014	0.052	-0.15	0.62	0.61	0.24	0.22	-0.069	0.099	0.76	-0.076	0.12	0.028	0.4	0.49	0.82
wcount	0.014	1	0.98	0.84	-0.1	-0.21	0.95	0.48	0.2	0.043	0.024	-0.015	0.33	0.0029	0.28	0.22	0.12
uniq_wcount	0.052	0.98	1	0.81	-0.097	-0.2	0.94	0.45	0.16	0.051	0.043	-0.016	0.35	0.0057	0.32	0.26	0.15
stop_word_count	-0.15	0.84	0.81	1	-0.25	-0.35	0.72	0.21	0.1	-0.024	-0.17	0.0014	0.12	-0.0029	0.039	0.07	-0.1
url_count	0.62	-0.1	-0.097	-0.25	1	0.57	0.093	0.39	-0.047	0.0047	0.76	-0.044	0.0079	0.0056	-0.013	0.054	0.51
avg_wlen	0.61	-0.21	-0.2	-0.35	0.57	1	0.059	0.25	-0.0049	0.07	0.77	-0.062	0.022	0.035	0.15	0.12	0.47
char_count	0.24	0.95	0.94	0.72	0.093	0.059	1	0.58	0.22	0.073	0.28	-0.037	0.37	0.019	0.34	0.31	0.29
punct_count	0.22	0.48	0.45	0.21	0.39	0.25	0.58	1	0.47	0.048	0.43	-0.017	0.21	0.038	0.17	0.13	0.25
hastg_count	-0.069	0.2	0.16	0.1	-0.047	-0.0049	0.22	0.47	1	0.0021	0.01	-0.0024	0.055	-0.0039	0.08	-0.0024	-0.031
mentn_count	0.099	0.043	0.051	-0.024	0.0047	0.07	0.073	0.048	0.0021	1	0.085	-0.009	0.067	0.26	0.069	0.084	0.08
max_wlen	0.76	0.024	0.043	-0.17	0.76	0.77	0.28	0.43	0.01	0.085	1	-0.066	0.13	0.051	0.26	0.32	0.64
emoji_cnt	-0.076	-0.015	-0.016	0.0014	-0.044	-0.062	-0.037	-0.017	-0.0024	-0.009	-0.066	1	-0.03	-0.0028	-0.045	-0.045	-0.083
capt_cnt	0.12	0.33	0.35	0.12	0.0079	0.022	0.37	0.21	0.055	0.067	0.13	-0.03	1	0.02	0.25	0.29	0.18
email_cnt	0.028	0.0029	0.0057	-0.0029	0.0056	0.035	0.019	0.038	-0.0039	0.26	0.051	-0.0028	0.02	1	-0.011	-0.0028	0.013
time_cnt	0.4	0.28	0.32	0.039	-0.013	0.15	0.34	0.17	0.08	0.069	0.26	-0.045	0.25	-0.011	1	0.44	0.58
char&num_cnt	0.49	0.22	0.26	0.07	0.054	0.12	0.31	0.13	-0.0024	0.084	0.32	-0.045	0.29	-0.0028	0.44	1	0.5
max_digit_rate	0.82	0.12	0.15	-0.1	0.51	0.47	0.29	0.25	-0.031	0.08	0.64	-0.083	0.18	0.013	0.58	0.5	1

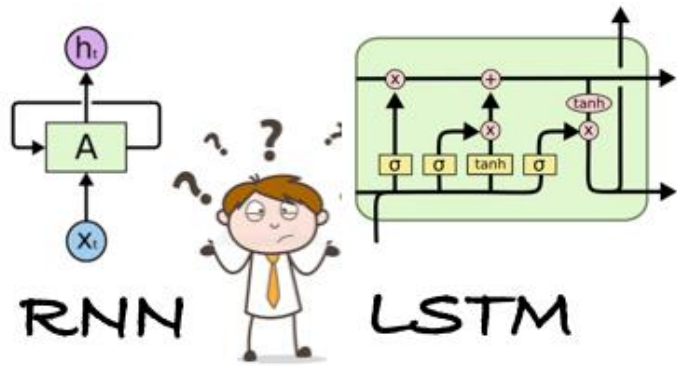
In the spam messages, we usually met:

- Word contained both characters & number
- Mentioned datetime
- Max word-length,
- etc

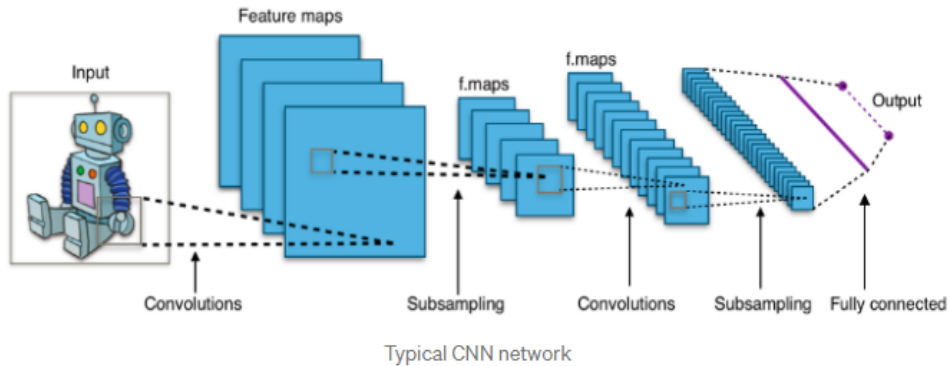
Feature to extract in spam-detection



Cases	Examples (discriptions)
number of hastags	#memories
number of url_link and unique url in email, messages / tweets / etc	https//:google.com
mention someone else	@David
hour of day / day of week / or any mentioned-timestamp when the email or messages / tweets was send / posted	2020-12-12 , 21 Jun 2020 , etc
number of emojiicon	:) , :v , =)) , etc
number of capitalized words	AbBa MoHameD
sum of all the character-lengths of word	len (word_splited)
number of words containing letters and numbers	128abc9*, 29Jun, etc
number of words containing only numbers or letters	123St, 92.No
max ratio of digit characters to all characters of each word	max ([len (digit(word)) / len (word) for word in words])
max the charecter-lengths of all words.	max ([len (word) for word in words])
number of words in email, messages or tweets / etc	len (word.split())
max length of word	max ([len (w) for w in words])
average length of word	mean ([len (w) for w in words])
number of punctuation	



All used models



Simpliest algorithms

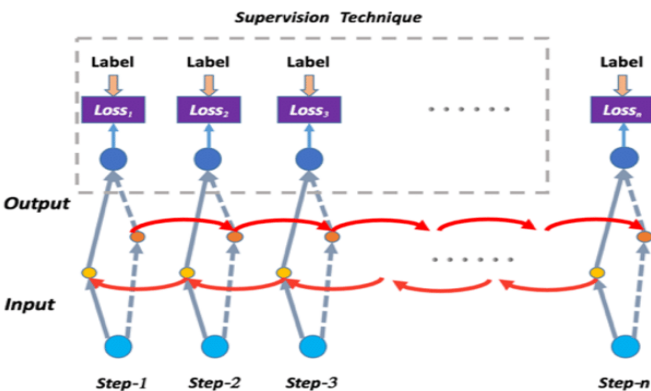
- KNN
- SVM
- Logistic Regression

Ensemble learning

- RandomForest
- ExtraTrees
- AdaBoost
- XGBoost,

Benchmark algorithms

- CNN
- RNN / LSTM
- GRU
- Bidirectional LSTM



Suffix : Transfer Learning

Since the limited of time, we only focus on the definition, advantage and disadvantage of 3 transfer learning models:

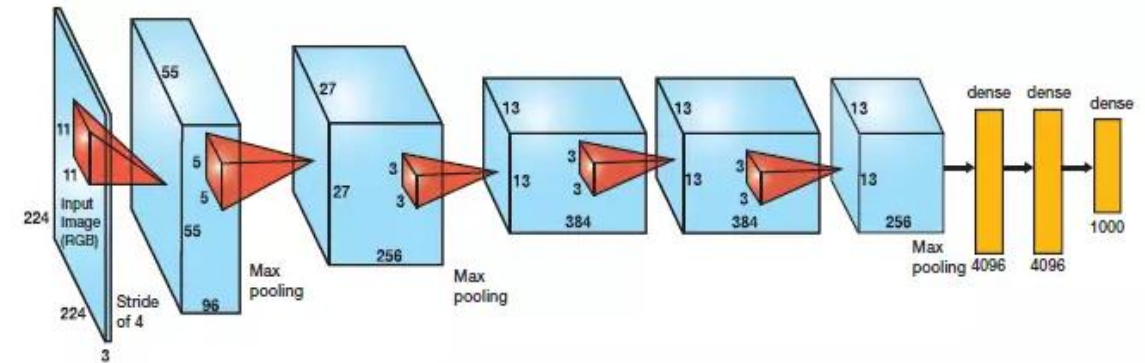
- VGG16
- AlexNet
- ResNet

Architecture	Top-1 Accuracy	Top-5 Accuracy	Year
Alexnet	57.1	80.2	2012
Inception-V1	69.8	89.3	2013
VGG	70.5	91.2	2013
Resnet-50	75.2	93	2015
InceptionV3	78.8	94.4	2016
Resnext-101	80.9	95.6	2017
Polynet	81.3	95.8	2017
DPN-131	81.5	95.8	2017
SE-Net-1	82.7	96.2	2017
PNasNet-5(N=4,F=216) *	82.9	96.2	2017
SE-Net-2	83.1	96.4	2018
AmoebaNet-C (N=6, F=228) *	83.1	96.3	2018

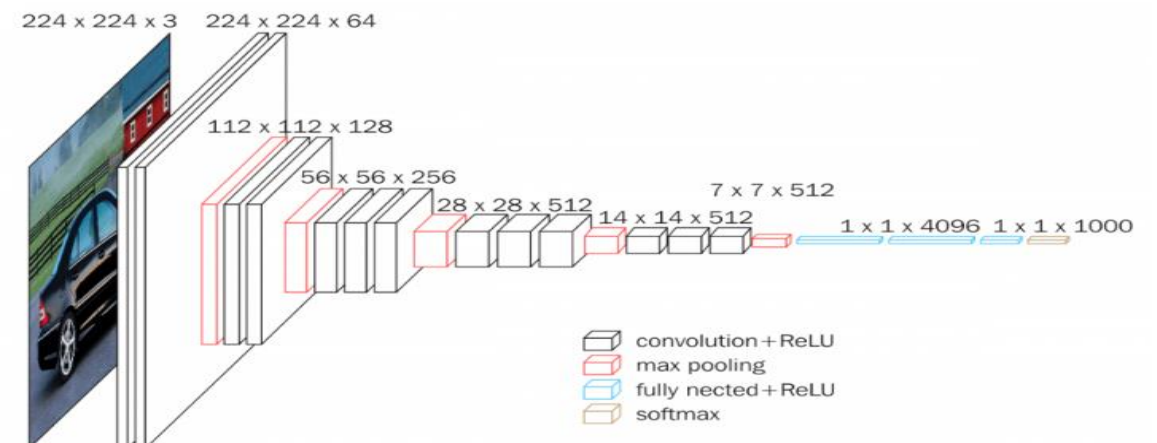
* network discovered using NAS

Made with ❤ at [CV-Tricks.com](https://www.cv-tricks.com)

Performance of various Neural Network architectures on Imagenet dataset



Architecture of Alexnet which won 2012 Imagenet challenge.

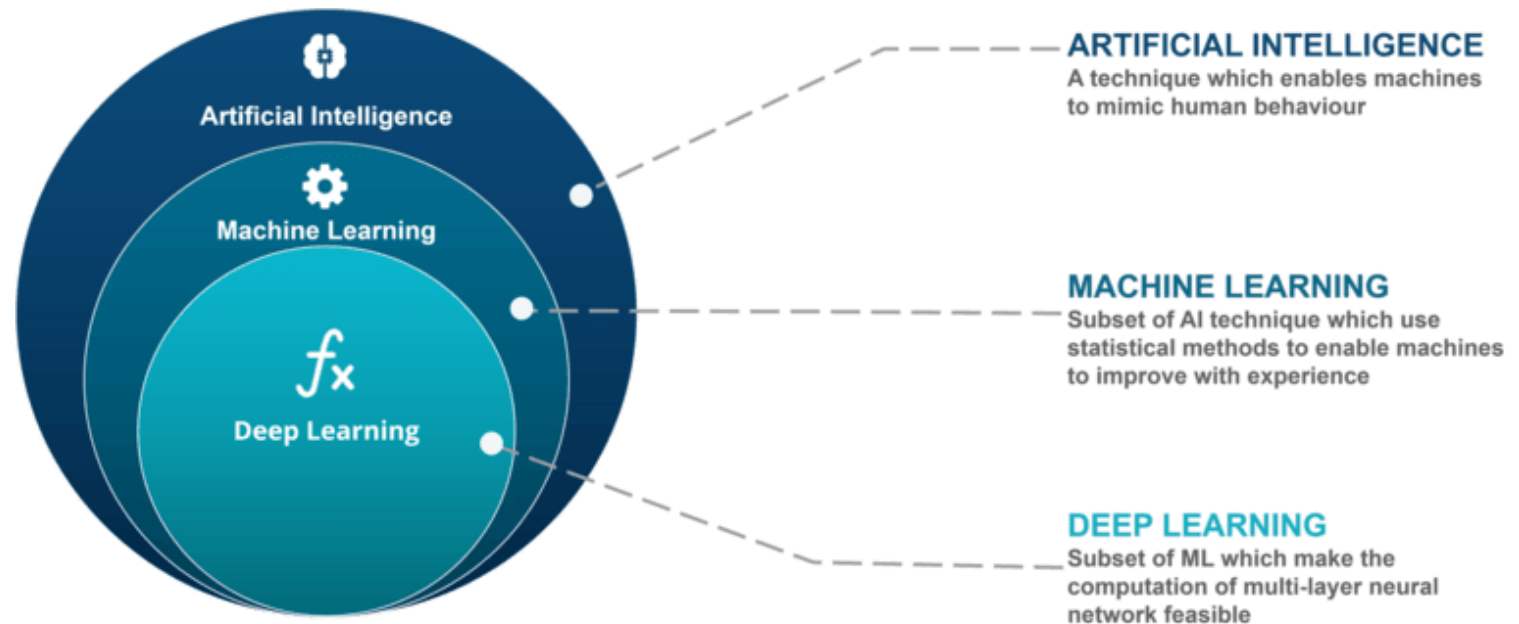


Comparison	VGG	ResNet	AlexNet
1st - release	2013	2015	2012 Alex Krizhevsky , Ilya Sutskever Geoffrey Hinton
Numbers of version	3 (VGG, VGG16, VGG19)	To many (ResNet18, ResNet50, ResNet34, ResNet101, ResNet152, etc)	1
Properties	<ul style="list-style-type: none"> - 3-4 ReLu layers - Grouping multiple convolution layers with smaller kernel sizes - Input shape = (224, 224, 3) - Max-depth from 16 to 19 layers 		<ul style="list-style-type: none"> - 8 layers and, - ReLu activation function which was a major discovery in deep learning
Advantage	<ul style="list-style-type: none"> - Massive improvement in accuracy and an improvement in speed - Brought with it various architectures built on the similar concept 	<ul style="list-style-type: none"> - Reduces the training time and improves accuracy. 	<ul style="list-style-type: none"> - Deeper architecture with 8 layers which means that is better able to extract feature. - Worked well for the time with color images. - Not limit the output unlike other activation function
Disadvantage	<ul style="list-style-type: none"> - This model experiences the vanishing gradient problem. - VGG is slower than the newer ResNet architecture 	<ul style="list-style-type: none"> - The complexity of an identical VGG network caused the degradation problem which was solved by residual learning. 	<ul style="list-style-type: none"> - Struggles to learn features from image sets - Take more time to achive high accuracy

Reminders

1. Machine Learning vs Deep Learning

Comparison Metrics	Machine Learning	Deep Learning
Algorithm structure	Simple	Complex or multi-layered
Data requirement	Low	High
Hardware requirement	Low	High
Feature learning	Manual input need	It learns by itself
Solving problems	Simple decisions	Complex decisions
Execution time	High	Low
Interpretation	Easy	Difficult





2. Neurons and Neural Network

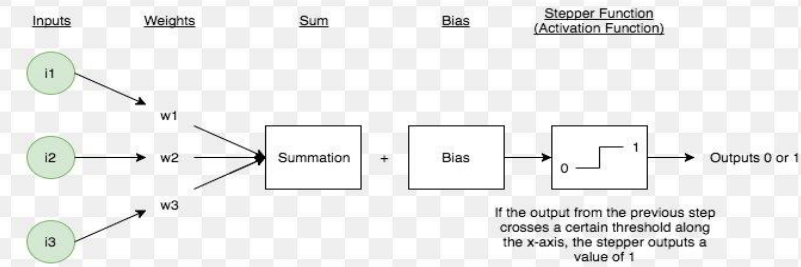
- **Neural networks**, also known as ANNs, are a subset of [machine learning](#) and are at the heart of [deep learning](#) algorithms.
- Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.
- There are two building blocks of a Neural Network
 1. Each layer consists of small individual units called neurons.
 2. A **neuron** in a neural network can be better understood with the help of biological neurons.
 3. An artificial neuron is similar to a biological neuron. It receives input from the other neurons, performs some processing, and produces an output.

3. Layers

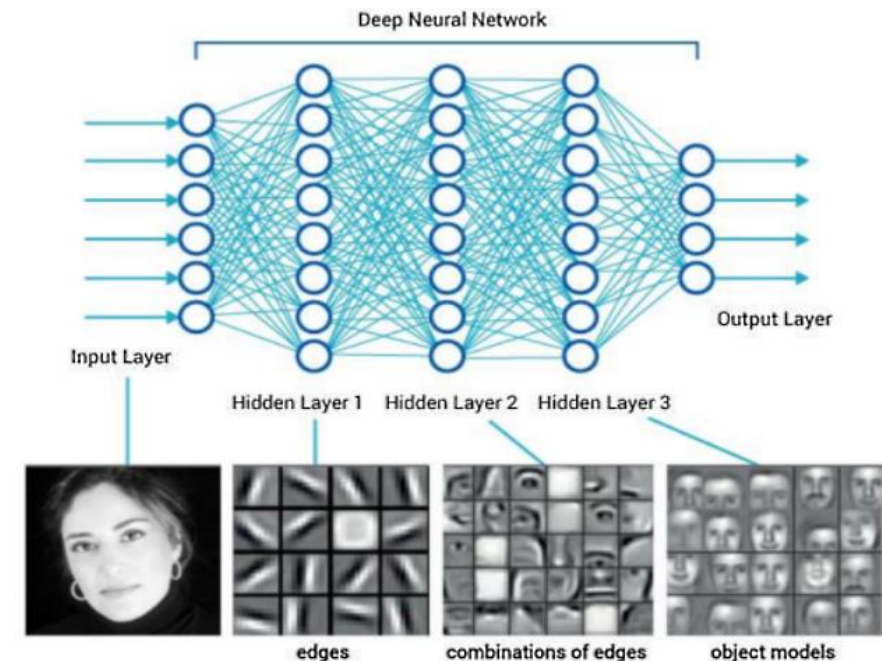
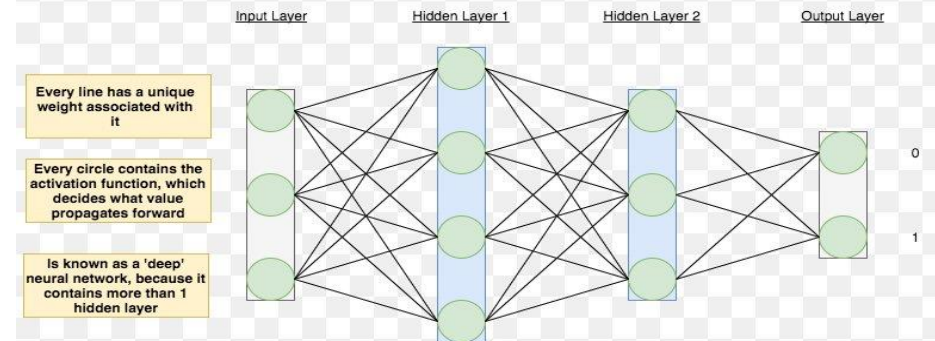
A neural network is made up of vertically stacked components called **Layers**.

- **Input Layer**– This layer will accept the data and pass it to the rest of the network.
- **Hidden Layer**– Hidden layers are either one or more in number for a neural network. Hidden layers are the ones that are actually responsible for the excellent performance and complexity of neural networks. They perform multiple functions at the same time such as data transformation, automatic feature creation, etc.
- **Output layer**– The output layer holds the result or the output of the problem. Raw images get passed to the input layer and we receive output in the output layer

Simple Neuron Diagram

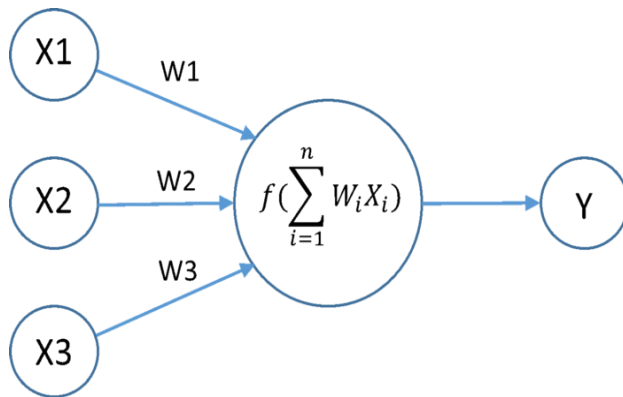


Deep Neural Network Diagram



4. Activation function

In artificial neural networks, the **activation function** of a node defines the output of that node given an input or set of inputs



Name	Plot	Function, $f(x)$	Range	Order of continuity
Identity		x	$(-\infty, \infty)$	C^∞
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$\{0, 1\}$	C^{-1}
Logistic, sigmoid, or soft step		$\sigma(x) = \frac{1}{1 + e^{-x}}$	$(0, 1)$	C^∞
Hyperbolic tangent (tanh)		$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$	C^∞
Rectified linear unit (ReLU) ^[12]		$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max\{0, x\} = x \mathbf{1}_{x>0}$	$[0, \infty)$	C^0

References

- [All of ImageNet: ResNet, UNet, AlexNet, etc.](#)
- [NLP and applications](#)
- [Spam-detection problem](#)
- [Fake-news detection](#)
- [Pre-processing in Data Mining](#)
- [GloVe Word Embedding](#)
- [Word2Vec](#)

Final exercises

- Will be shown later in this Friday

Summary-THE END
