Deloitte.

Deloitte Data Science Academy

Lesson 4:

Natural Language Processing, Computer Vision, Auto ML

Introduction to advanced ML topics

19/11/2019

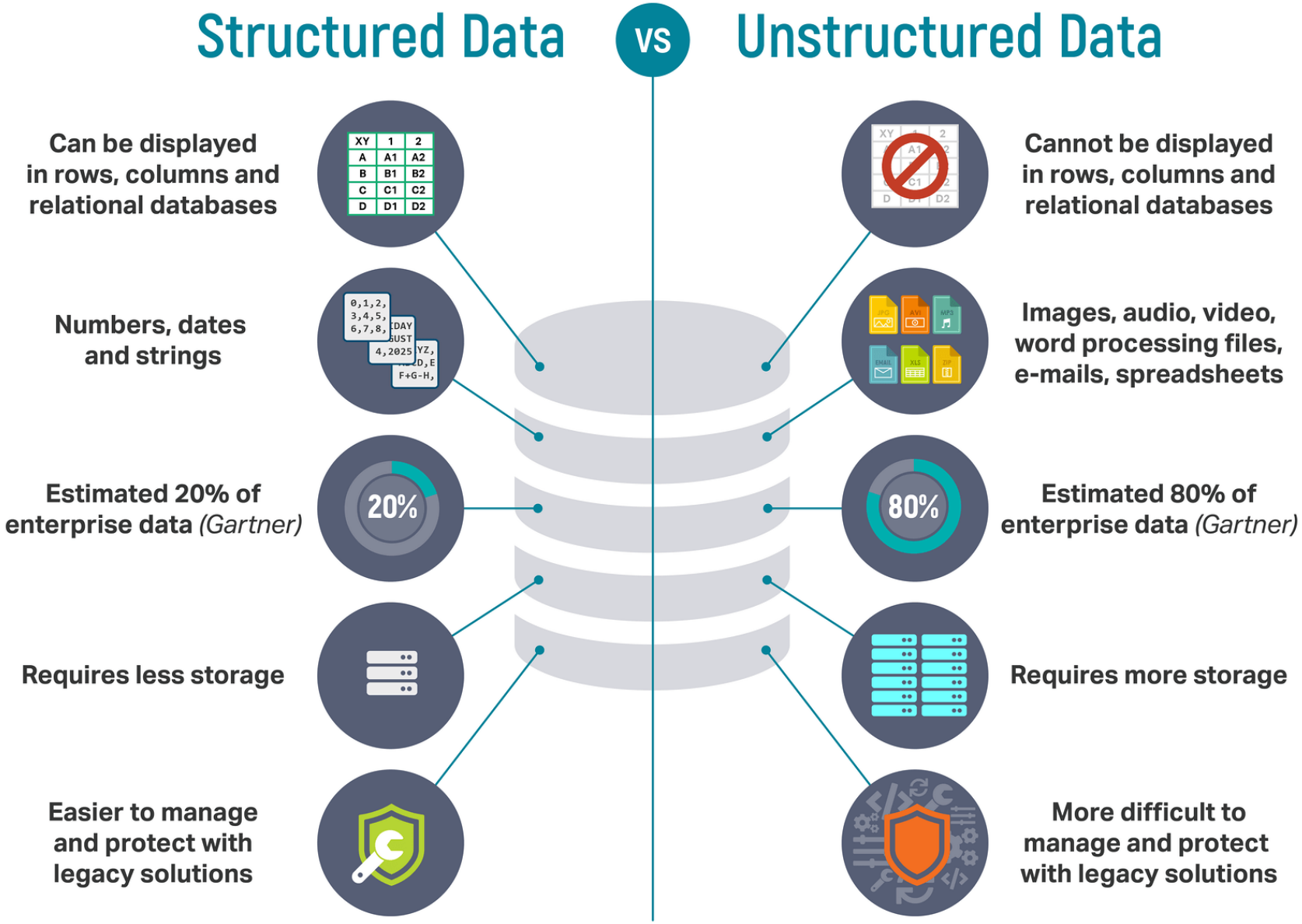
Today's program

1	Introduction to Data Science, Data Science Tools, Keboola introduction, Python basics, Git – Code collaboration, Pure Python data exploration	HA1
2	Features and dataset preparation, Clustering, Behavioral segmentation, Hands on training for data exploration and clustering with Python, Anomaly Detection	HA2
3	Regression and Classification, Hands on training for classification and regression	HA3
4	**Introduction to NLP and Computer Vision, AutoML**	**HA4**

Jupyter Lab, Jupyter Notebook, Python, Anaconda, Miniconda

Introduction to unstructured and big data

Structured Data vs Unstructured Data



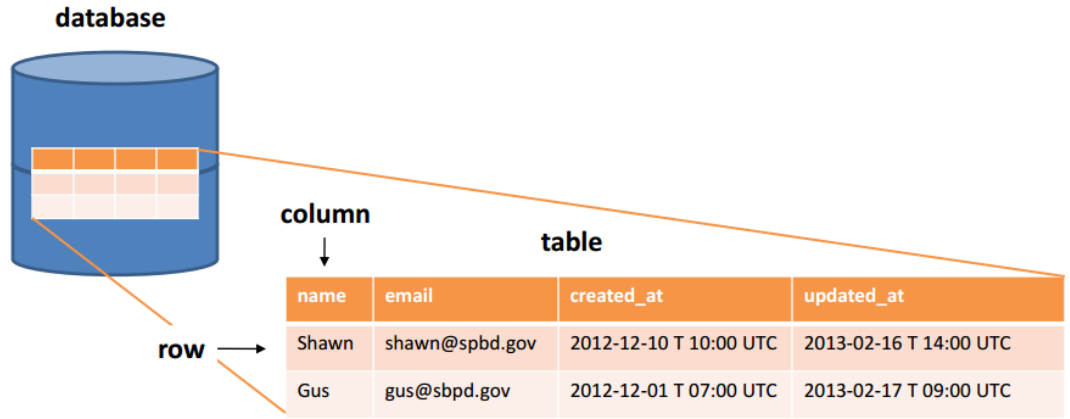
https://lawtomated.com/structured-data-vs-unstructured-data-what-are-they-and-why-care/

Structured data

Properties of structured data

Structured data is data that is arranged in a **fixed pre-defined structure and format**

- Resides in **relational databases**, i.e. a table of rows and columns containing related information
- **Organised** and labelled to identify meaningful relationships between data
- **Easily accessible (SQL)** and usable in analyzes

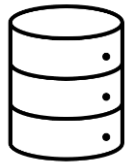


https://code.tutsplus.com/tutorials/relational-databases-for-dummies--net-30244

Unstructured data

Properties of unstructured data

Unstructured data is data that is not **structured via pre-defined data format or schema**



- resides as objects in **non-relational database like NoSQL**

- **not organised** and labelled to identify meaningful relationships between data

- **often not directly applicable in analyzes**

https://www.nicepng.com/maxp/u2w7r5e6t4w7u2q8/

Human and Machine-generated unstructured data

Origin of unstructured data

Human-generated unstructured data

- **Text files**: word processing files, spreadsheets, presentations, emails
- **Social Media**: contributions to social networks (tweets,...)
- **Media**: MP3, digital photos, audio recordings and video files

https://theconversation.com/to-serve-a-free-society-social-media-must-evolve-beyond-data-mining-94704

Machine-generated unstructured data

- **Satellite imagery** (www.spaceknow.com)
- **Scientific data**

https://www.planet.com/products/planet-imagery/

Big data phenomenon

Exponencialy explosion of produced and stored data

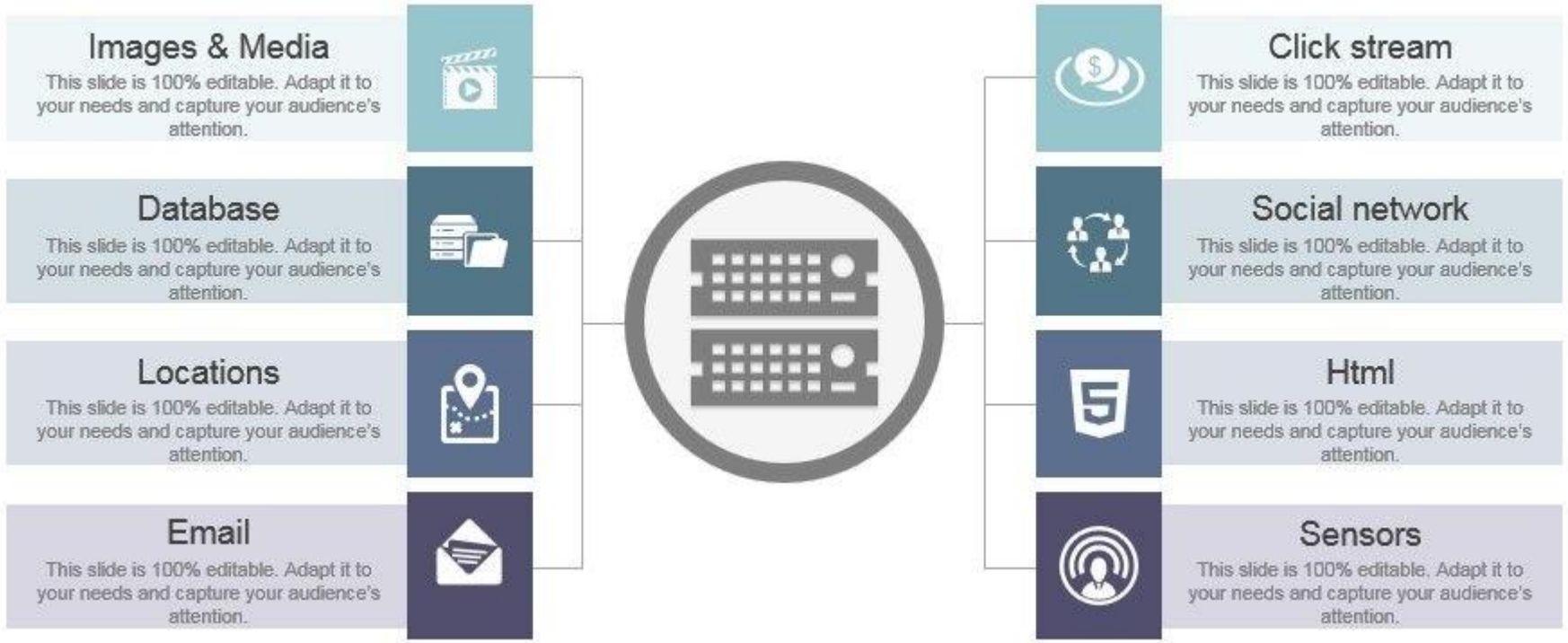
90% of the world's data has been produced in just the **last two years**.

By **2020** the total amount of data stored was **50x** larger than **2015**.

https://www.btelligent.com/en/portfolio/big-data/

Big data phenomenon

Different data sources

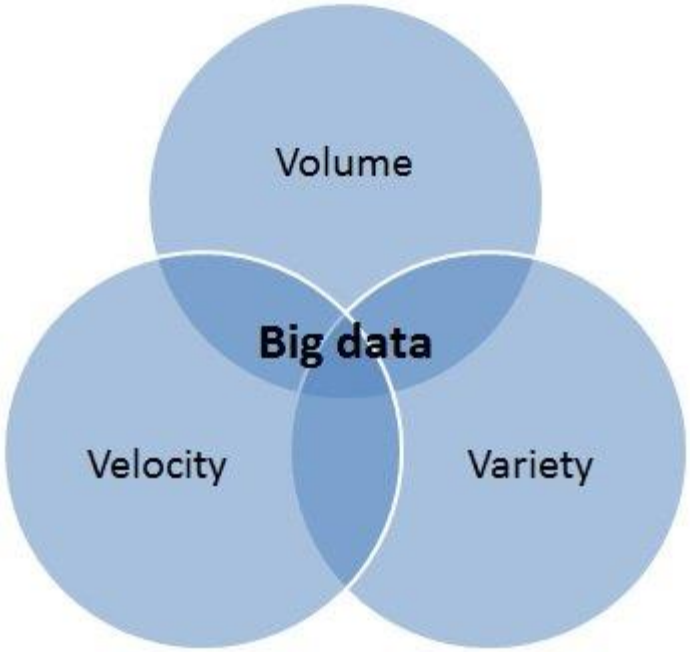


https://www.slideteam.net/big-data-sources-and-concept-ppt-sample-file.html

Big data phenomenon

Big data 3 V's properties

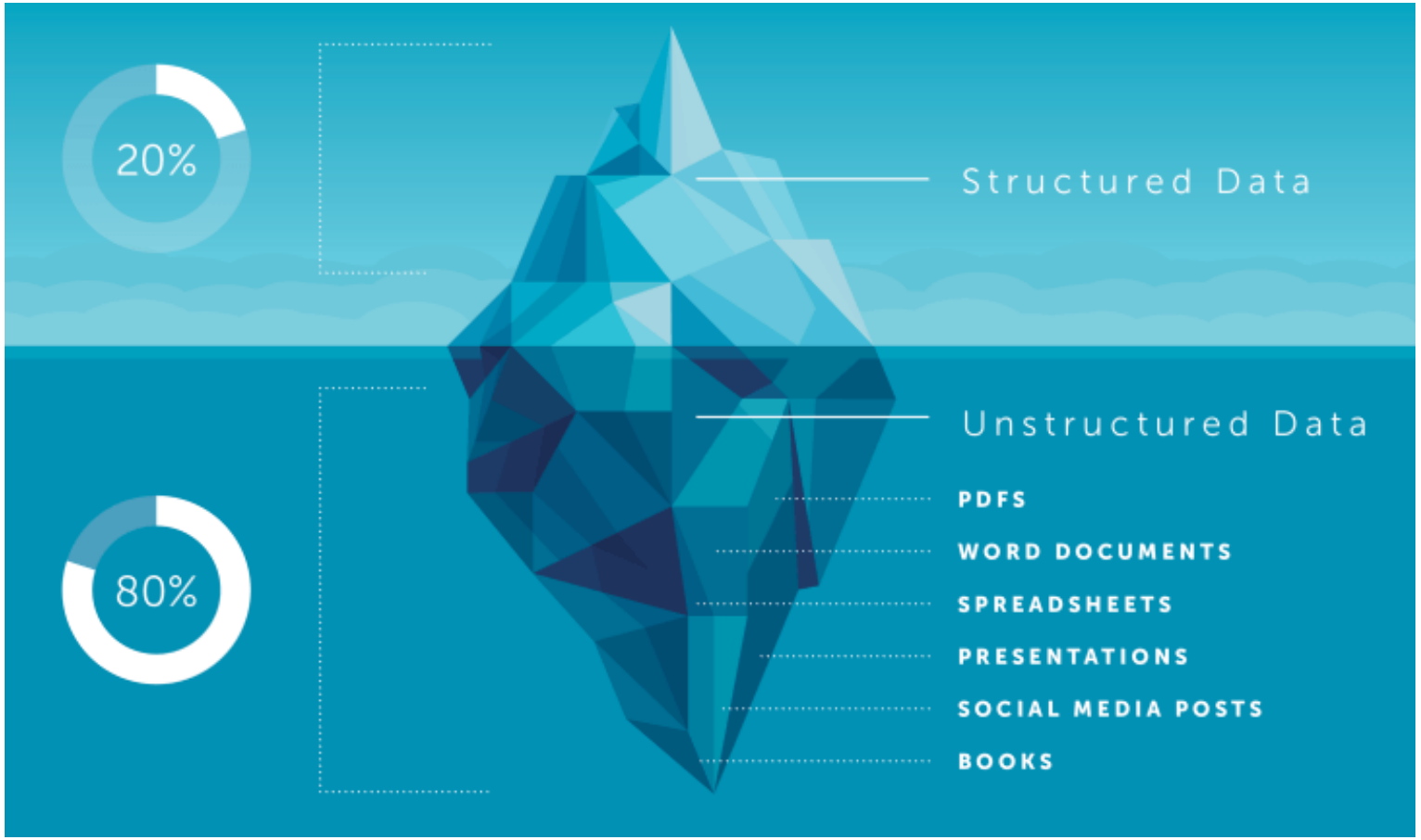
- Volume: increasingly faster-growing masses of data
- Velocity: created and processed at high speed and in real time
- Variety: diverse sources and formats
- (Veracity)
- (Value)



https://bigdataldn.com/intelligence/big-data-the-3-vs-explained/

Big data phenomenon

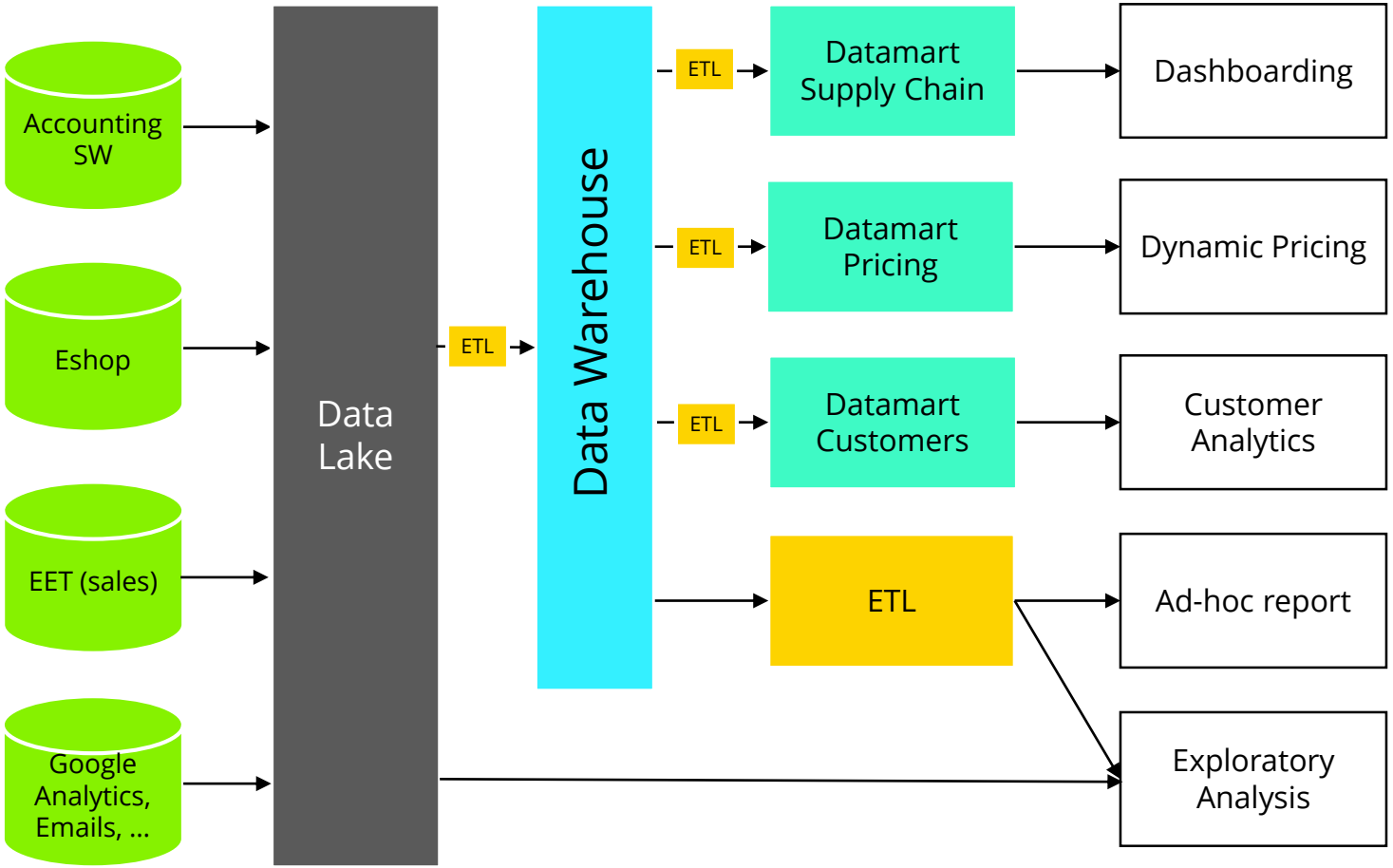
Uneven growth of structured and unstructured data → Uneven distribution



https://lawtomated.com/structured-data-vs-unstructured-data-what-are-they-and-why-care/

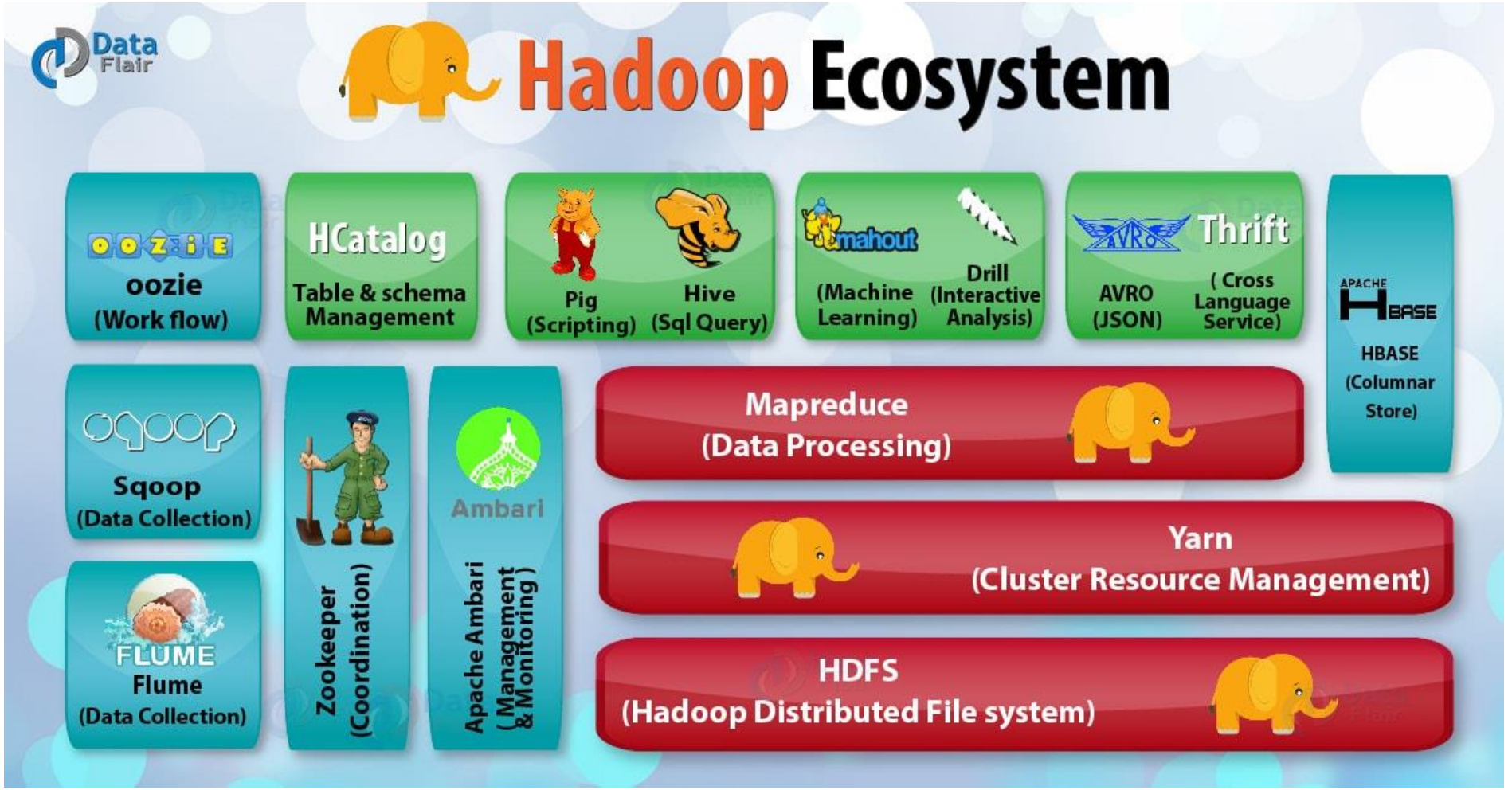
Data Lake

Data storage to store "everything"



- Data collected in Data Lake **does not neccessarily have purpose** and not much effort is made into cleansing and transforming the data.
- Data Lakes are more likely to also contain semi-structured or unstructured data.
- Data Lakes are more likely to employ some of the Big data technologies like distributed file systems, cloud computing or data streaming.

Big Data – Hadoop and its ecosystem



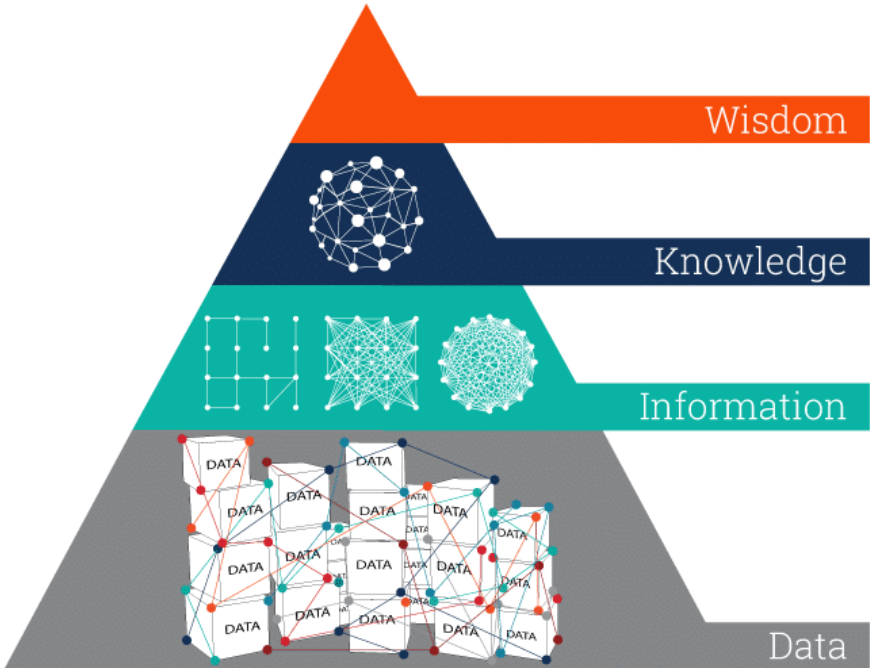
https://towardsdatascience.com/big-data-analysis-spark-and-hadoop-a11ba591c057

Big Data and its need for AI

AI comes to the fore

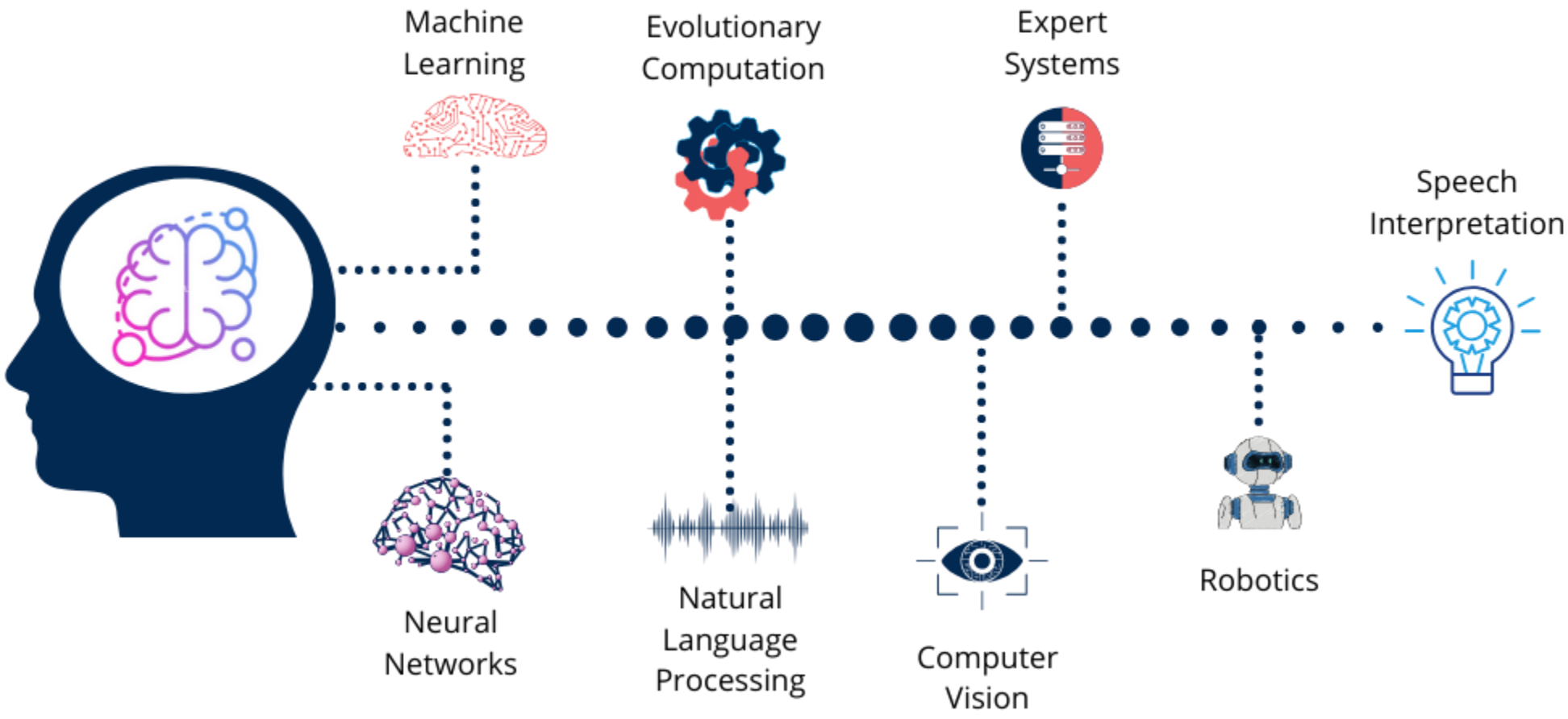
- Ever-increasing volumes of (un)structured data produced naturally create a demand for their **processing and analysis**
- Processing and analysis of ever-increasing volumes of (un)structured data cannot be performed efficiently by humans.

Artificial intelligence comes to the fore.

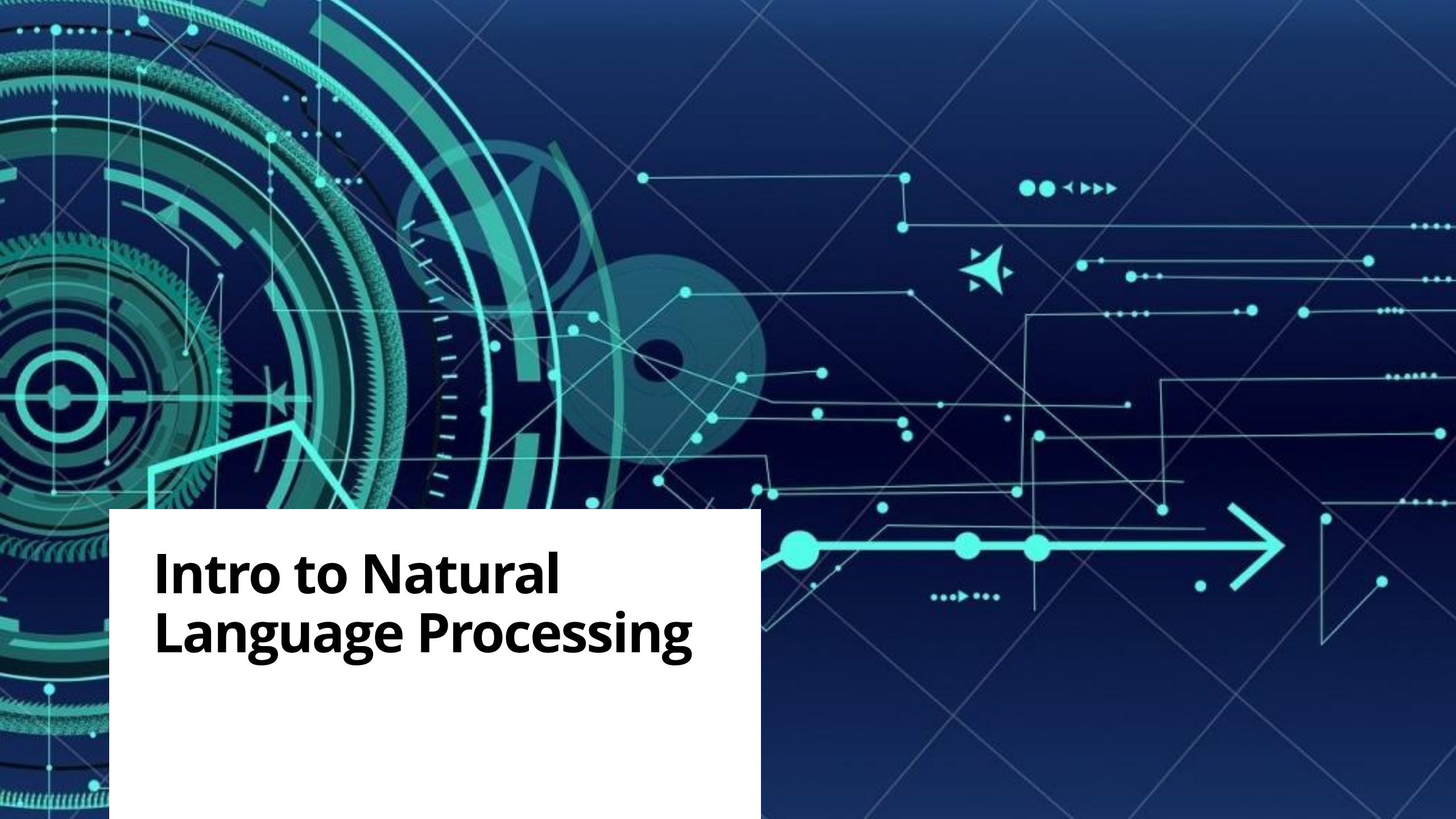


https://www.ontotext.com/knowledgehub/fundamentals/dikw-pyramid/

Artificial intelligence



https://www.cybiant.com/resources/what-is-artificial-intelligence/?v=928568b84963



Intro to Natural Language Processing

Natural Language Processing

Natural Language Processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages.

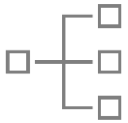
NLP is a less business term for and/or tools used in **Text Analytics/Text Mining** and has more scientific vibe.

Natural Language Processing

Use cases

Information retrieval

Named entity recognition



Text classification

Clustering

Summarization

Use cases

Information retrieval

Navigate end users in the large amounts of text to help them find the right document that is matching their search criteria

Three basic methods:

- **Full text scanning**

  → Full scan across all documents for every search query

- **Indexing**

  → Build an index of words first and then just search within that index

- **Vectorization of documents and users queries**

  → create a vector representation of documents and query and then just compare them

Use cases

Information retrieval → Indexing

Suppose we have two documents:

Doc. 1	This is my sample sentence.
Doc. 2	This car is beautiful.

Both documents are represented by a list of words:

Doc. 1	this; is; my; sample; sentence
Doc. 2	this; car; is; beautiful;

Index →

this	Doc. 1, Doc. 2
is	Doc. 1, Doc. 2
my	Doc. 1
sample	Doc. 1
sentence	Doc. 1
car	Doc. 2
beautiful	Doc. 2

Use cases

Information retrieval → Vectorization of documents and users queries

Suppose we have two documents and one user query:

Doc. 1	This is my sample sentence.
Doc. 2	This car is beautiful.
Query	Beautiful car.

Both documents are represented by a list of words:

Doc. 1	this; is; my; sample; sentence
Doc. 2	this; car; is; beautiful;

Vocabulary
[this; is; my; sample; sentence, car, beautiful]

Vectors representing documents and query:

Doc. 1	[1, 1, 1, 1, 1, 0, 0]
Doc 2.	[1, 1, 0, 0, 0, 1, 1]
Query	[0, 0, 0, 0, 0, 1, 1]

Use cases

Information retrieval → Vectorization of documents and users queries

Suppose we have two documents and one user query:

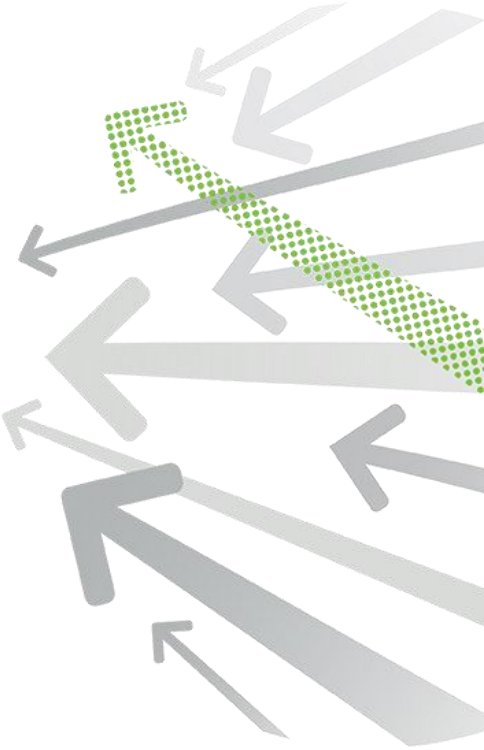
Doc. 1	This is my sample sentence.
Doc. 2	This car is beautiful.
Query	Beautiful car.

We obtained their simplest vector representations:

Doc. 1 [1, 1, 1, 1, 1, 0, 0]

Doc 2. [1, 1, 0, 0, 0, 1, 1]

Query [0, 0, 0, 0, 0, 1, 1]



Search based on vector distance/similarity of vectors

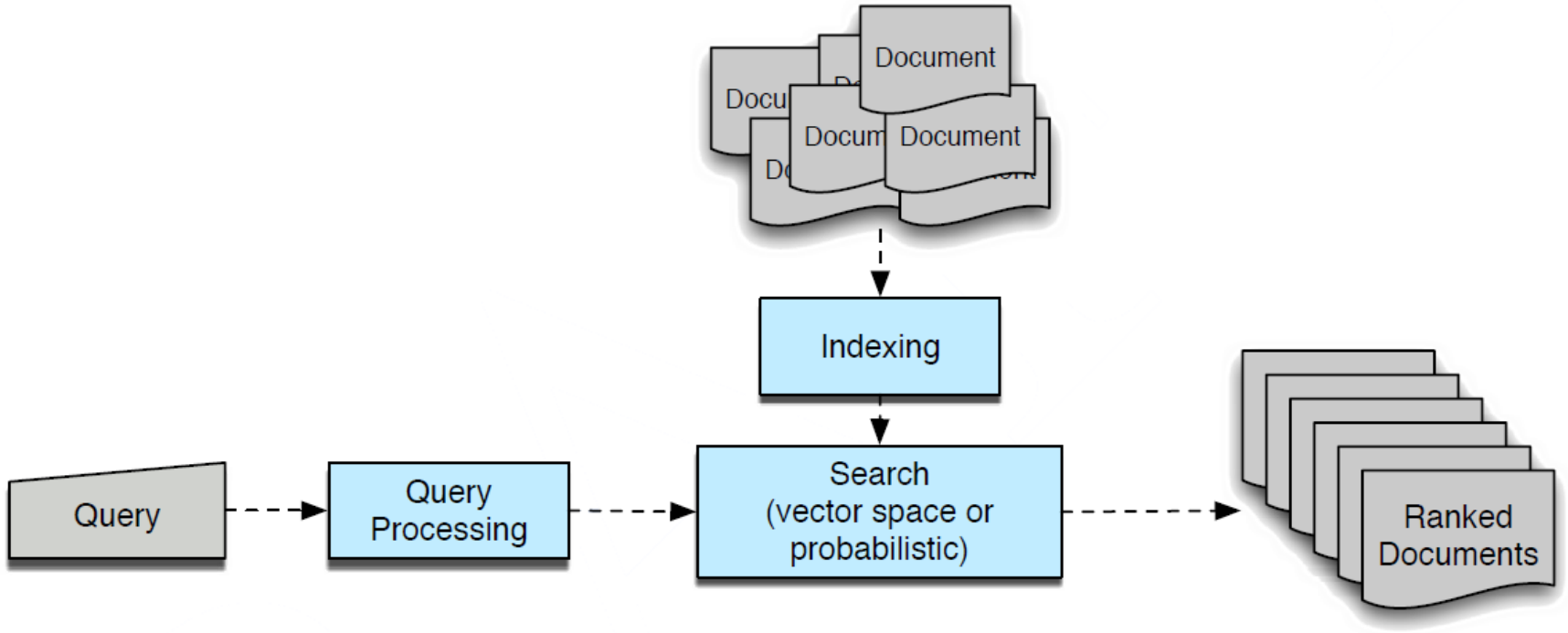
Cosine similarity: $similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$

Similarity(Query, Doc. 1) = **0**

Similarity(Query, Doc. 2) = **0.7**

Use cases

Information retrieval



https://devopedia.org/information-retrieval

Use cases

Named entity recognition

Identification of entities mentioned in the text

- Both known and unknown entities
- People names, company names, locations etc.
- Named entities (NEs) can be one **word** or **multi word**

Example:

"**Apple** is looking at buying **U.K.** startup for **$1 billion**"

Apple	ORG
U. K.	GPE
$1 billion	MONEY

Use cases

Named entity recognition

Three basic methods:

- **Gazetteer methods (list of NEs)**

  → search all occurences of NEs from lists in a target strings

- **Rule–based extraction**

  → linguistic grammar-based techniques

- **Semi-supervised/supervised machine learning**

  → ER is solved as **classification task** for each token in a sequence

Use cases

Text classification

Grouping texts into **predefined** categories based on their content

Applications:

- **Sentiment analysis**

  → categories: negative X neutral X positive sentiment

- **Topic Labeling**

  → categories: politics X sport X …

- **Language Detection**

  → categories: English X Czech X …

- **Intent Detection**

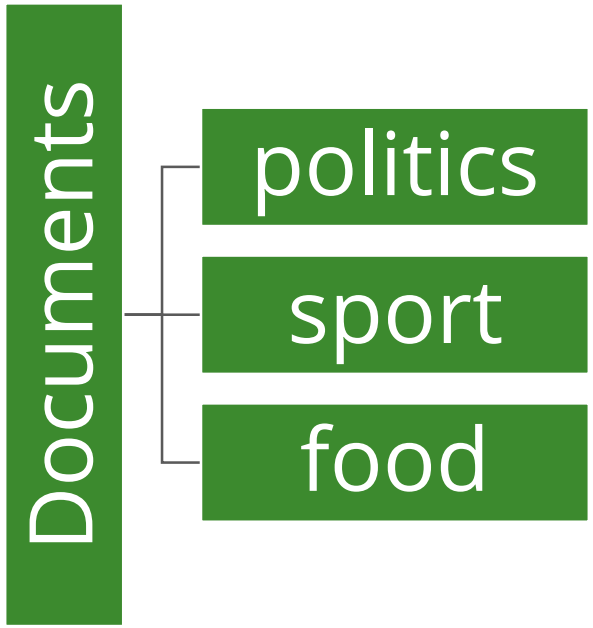
  → categories: Interested X Not Interested

Use cases

Text classification → methods

Two basic methods:

- **Rule-based/lexicon-based approach**
- **Machine learning based approach (supervised) → Traditional**
  → Neural-based



Use cases

Text classification → Rule-based/lexicon-based approach

classifies texts into organized groups by using a set of **handcrafted linguistic rules and lexicons**

These rules:

- instructs the system to use semantically relevant elements of a text to identify relevant categories based on its content
- consists of an antecedent or pattern and **a predicted category**

Two types of approach:

- **Simple Rule-based/lexicon-based approach**

  → uses primarily lexicons

- **Modified Rule-based/lexicon-based approach**

  → work with lexicons, but also with context and grammatical/linguistic rules

Use cases

Text classification → Rule-based/lexicon-based approach

Suppose we have a lexicon of terms related to sports and politics:

Sports = [football, basketball, LeBron James]

Politics = [Donald Trump, Hillary Clinton, Putin]

Lexicons are now used for text classification:

Text = „When is LeBron James' first game with the Lakers? “

count the number of words in the text related to:

- Sports → 1
- Politics → 0

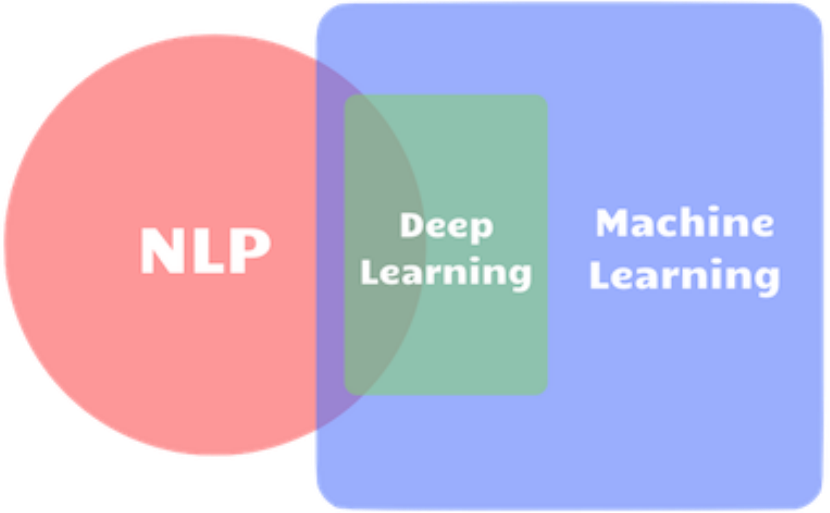
} text is classified as **sports**

Use cases

Text classification → Machine learning based approach (supervised)

classifies texts into organized groups by using **learned model of:**

- **traditional ML methods**
  - designed for structured data
- **neural-based ML methods (Deep Learning)**
  - designed for (un)structured data



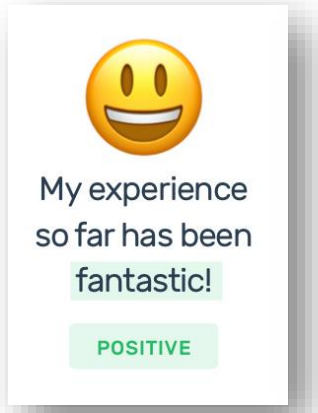
https://sigmoidal.io/natural-language-processing-vs-machine-learning-vs-deep-learning/

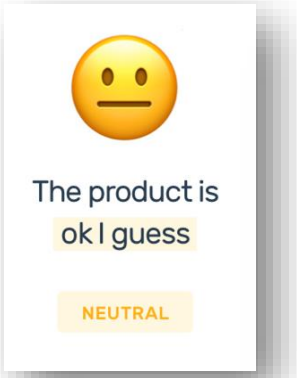
Use cases

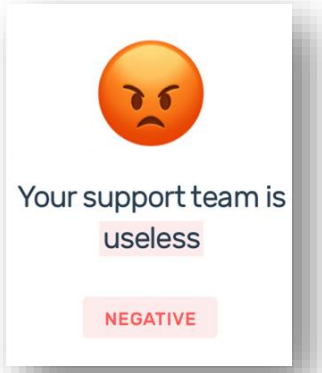
Text classification → Application → Sentiment Analysis

recognition of opinions, subjective views or attitudes in free unstructured texts

- positive, neutral or negative sentiment







Use cases

Text classification → Application → Topic Labeling

recognition of what a given text is talking about

- often used for structuring and organizing text data

Example: classify customer feedback

Text

"Customer service is terrible. I was on hold for hours."

Customer feedback categories

Customer Support
Ease of Use
Features
Pricing

Label: **Customer Support**
Confidence: **98 %**

Use cases

Text classification → Application → Language Detection

process of classifying incoming text according to its language

- often for routing purposes (e.g., route support tickets according to their language to the appropriate team).

Text

"Text mining je vědecká disciplína na pomezí dolování z dat, strojového učení a počítačové lingvistiky."

Language categories

Czech
English
German
Dutch

Label: **Czech**
Confidence: **92 %**

Use cases

Text classification → Application → Intent Detection

analyzes text to understand the reason behind feedback and

- allow identify customer's **intent to purchase a product**
- often used for customer service, marketing email responses,...

Text

"The software looks pretty cool.
I'd love to schedule a time to talk more."

Intent categories

Interested
Not Interested
Unsubscribe
Wrong Person

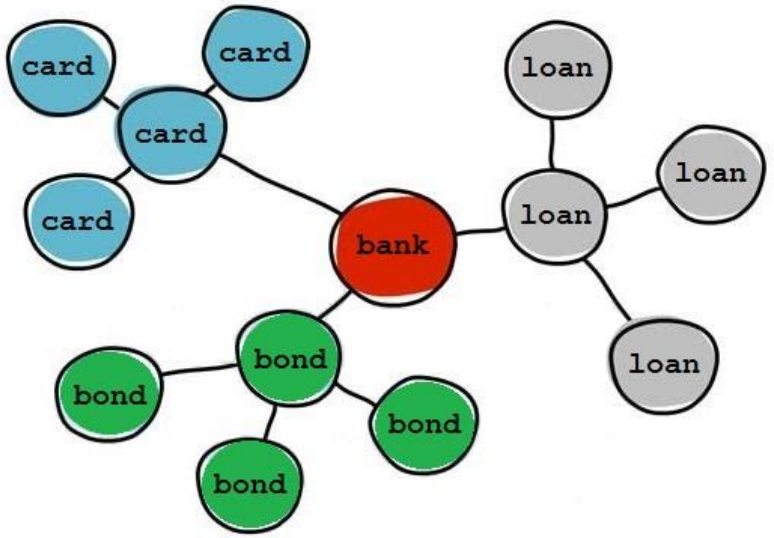
Label: **Interested**
Confidence: **99 %**

Use cases

Cluster analysis

allow grouping of texts into groups based on their similarity

- without training data (clusters are recognized automatically)
- used for **Content analysis, clustering**



Calculation of similarity

- based on comparing vector representations of texts
- **similarity of texts == similarity of numeric vectors of texts**
- metrics: Cosine similarity, Euclidean distance,...

Methods

- unsupervised ML methods for clustering (k-means,...)

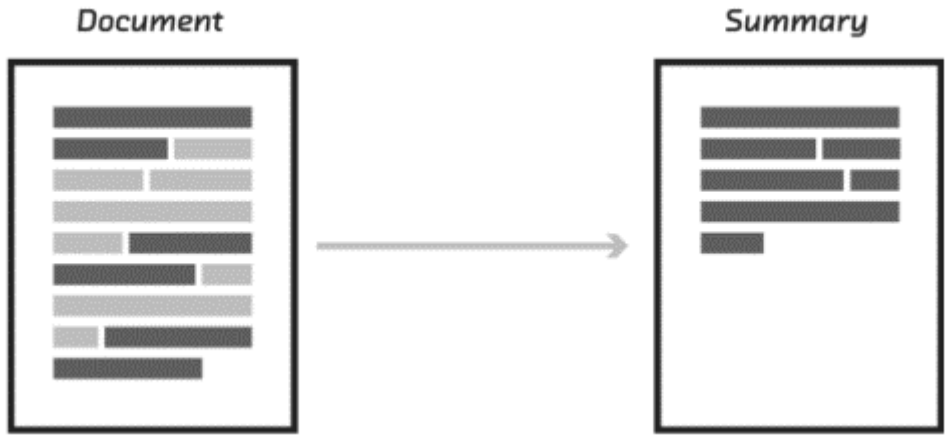
  → need a structured representation of texts (discussed in detail later)

Use cases

Summarization

allow extracting summaries from the original huge text without losing vital information

- Requirements: fluent, continuous and depict the significant



https://medium.com/@ondenyi.eric/extractive-text-summarization-techniques-with-sumy-3d3b127a0a32

Use cases

Summarization

Two basic approaches to text summarization:

- **Extractive Summarization**

  → **extracts** several parts, such as phrases and sentences, from a piece of text and stack them together to

  create a summary.

  → the summary obtained contains **exact sentences from the original text**

- **Abstractive Summarization**

  → use advanced NLP techniques to **generate** an entirely new summary.

  → the summary obtained **does not** contain exact sentences from the original text

Processing text data

Text data

Focus on sources of text data

Transcription notes

Client's emails

Call centre calls

Logs

Internal documents

User reviews

Online discussions

Social networks

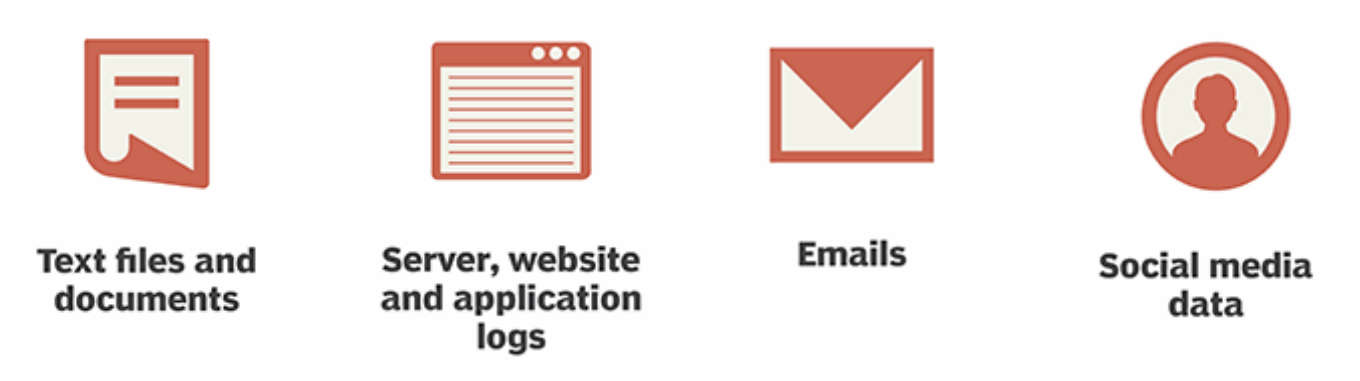
Text data

Focus on text as data

Text as data usually consists of documents which can represent:

- words,
- sentences,
- paragraphs of free flowing text

} not structured data via pre-defined format
unstructured data



ML over text data

Use of ML methods over text data

In previous tasks, we often used text data in the context of **ML methods**

ML methods on textual data can be divided into two basic groups:

- **Traditional**
- **Neural-based**

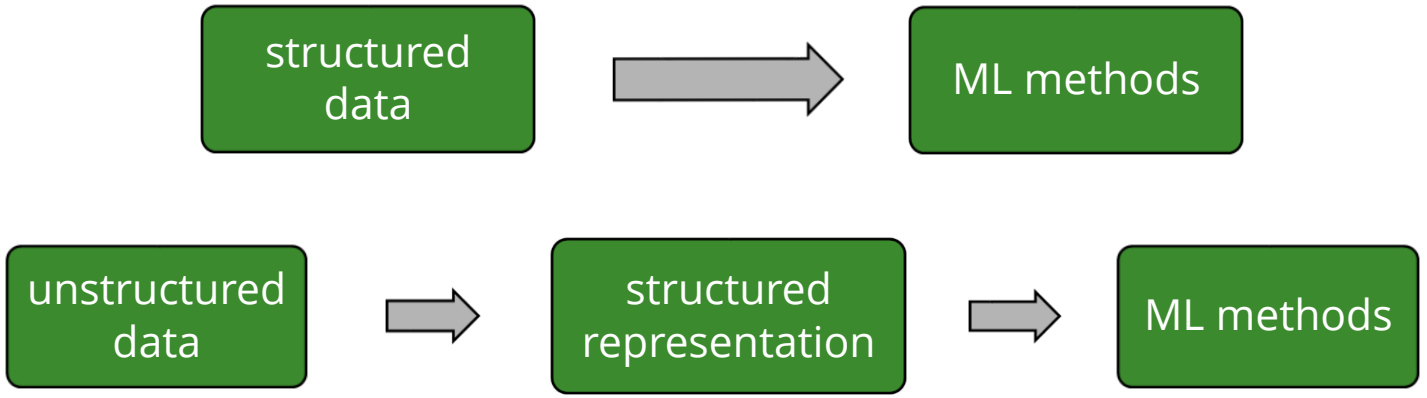
Traditional ML methods over text data

Use of traditional ML methods over text data

In previous tasks, we often used text data in the context of **traditional ML methods**, but:

- ML methods are not able to work directly with unstructured data,
- can only work with **structured data**

structured representation of unstructured data

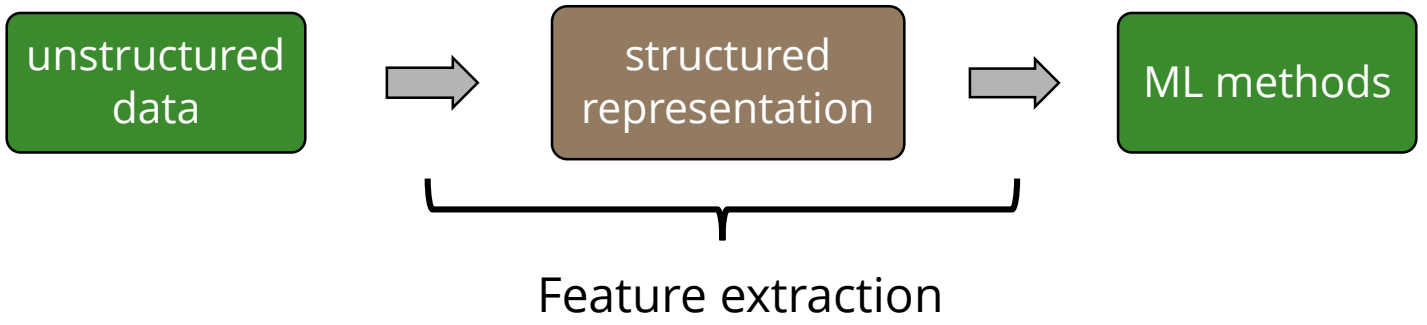


Traditional ML methods over text data

Structured representation

Structured representation of text document obtained using **feature extraction** must:

- appropriately **represent** the content of the original unstructured text document,
- be **suitable** for analysis or to drive machine learning (ML) algorithms.



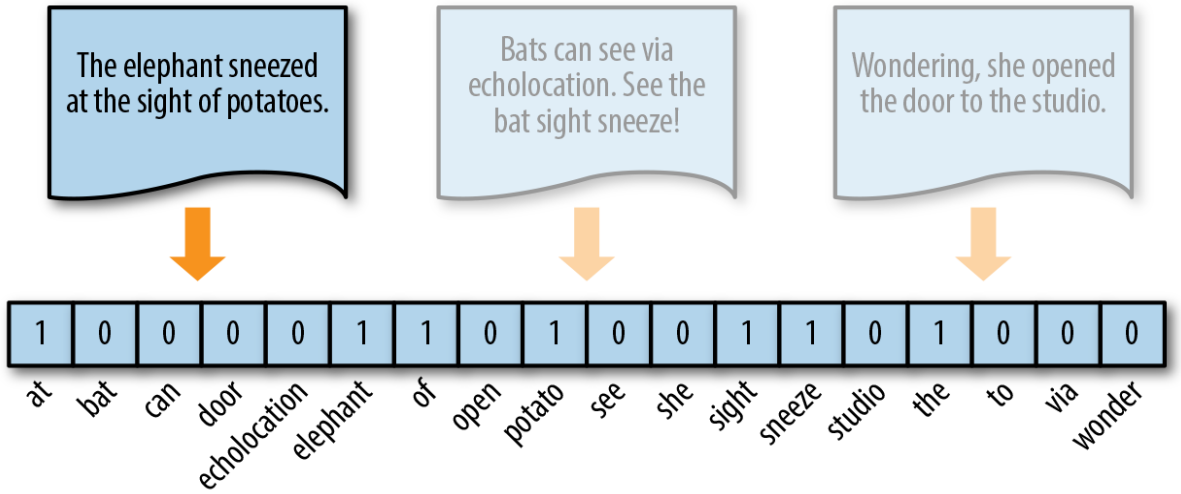
- One way to create a **structured representation** of documents is to **vectorize** them.

Traditional ML methods over text data

Vectorization of documents/Feature extraction

Vectorization of documents

- represent each text document with a **structured numeric vector**

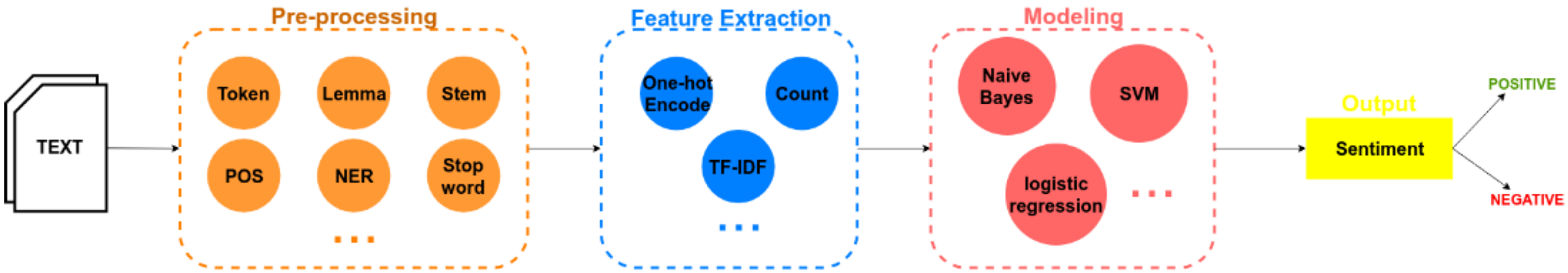


https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html

- vectorization of entire text documents brings the problem of **excessive dimensionality**
  → each form of a word or phrase represents a dimension
  → problem eliminated by using NLP techniques for **text preprocessing**

Traditional ML methods over text data

Workflow



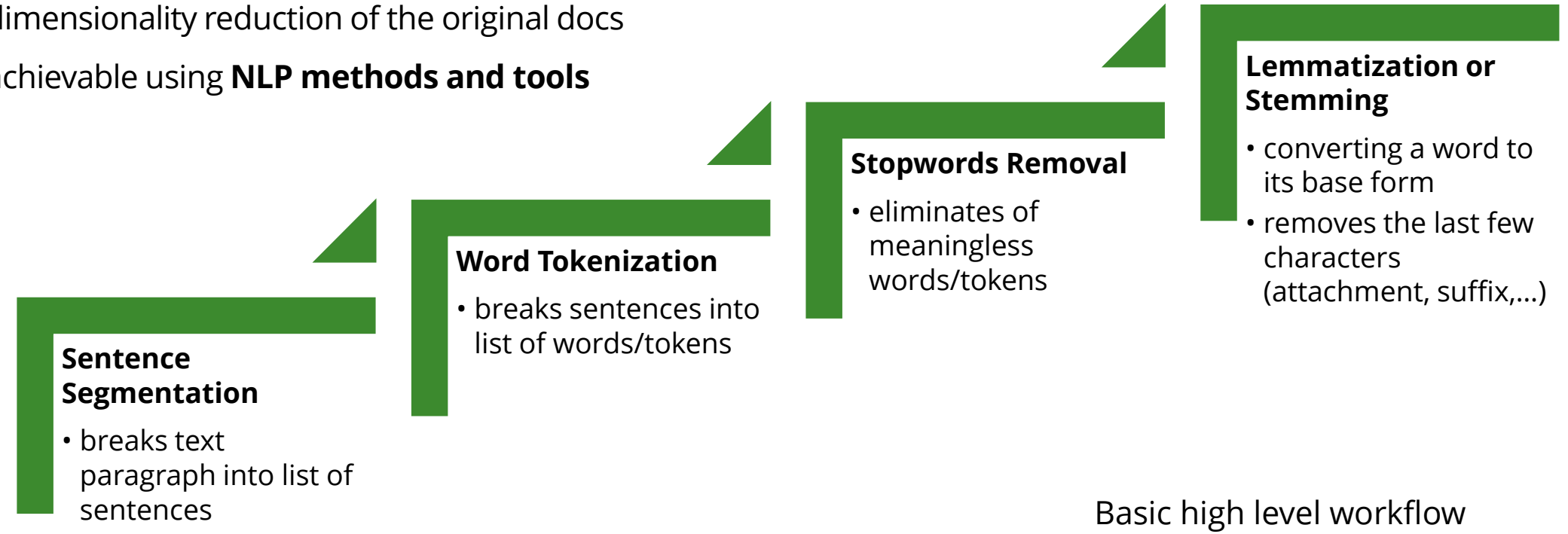
https://www.researchgate.net/publication/341998176_Sentiment_Analysis_Based_on_Deep_Learning_A_Comparative_Study

Text preprocessing

NLP methods for Text preprocessing

Aim of text preprocessing:

- represent each text document with **list of relevant terms/tokens**
- dimensionality reduction of the original docs
- achievable using **NLP methods and tools**



Basic high level workflow

Text preprocessing

NLP method → Word tokenization

Word tokenization is the process of splitting a large sample of text into words

Text

'This movie is very scary and long'

Tokens

['this', 'movie', 'is', 'very', 'scary', 'and', 'long']

Text preprocessing

NLP method → Stopwords Removal

Stopwords Removal is the process of removing words which does not add much meaning to a sentence

- Stopwords can safely be ignored without sacrificing the meaning of the sentence
- words like the, he, have etc.

Tokens

['this', 'movie', 'is', 'very', 'scary', 'and', 'long']

Stopwords removed

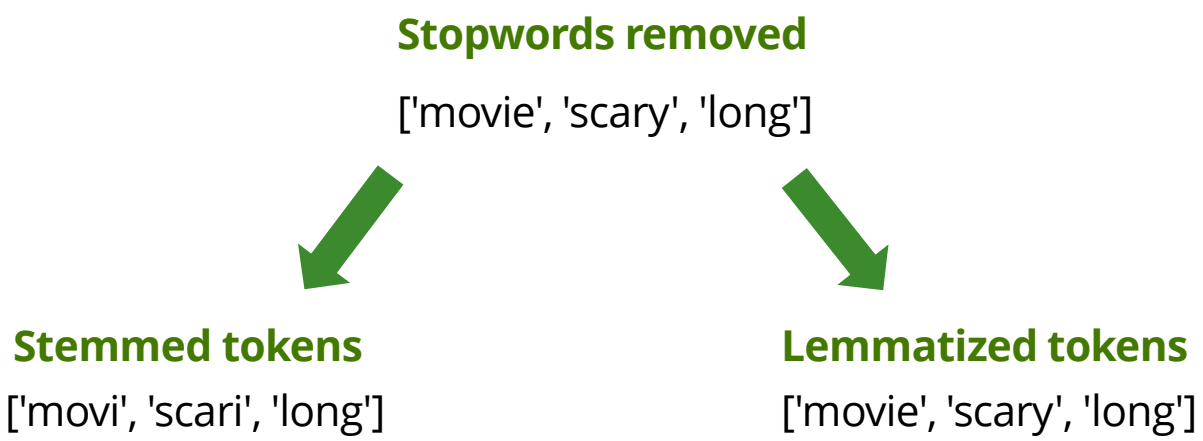
['movie', 'scary', 'long']

Text preprocessing

NLP method → Stemming and Lemmatization

Both generate the root form of the inflected words

- Stemming → reduces word to their word stem
  → stem – often **non-dictionary** base form of a word
- Lemmatization → reduces word to their word lemma
  → lemma – **dictionary** base form of a word

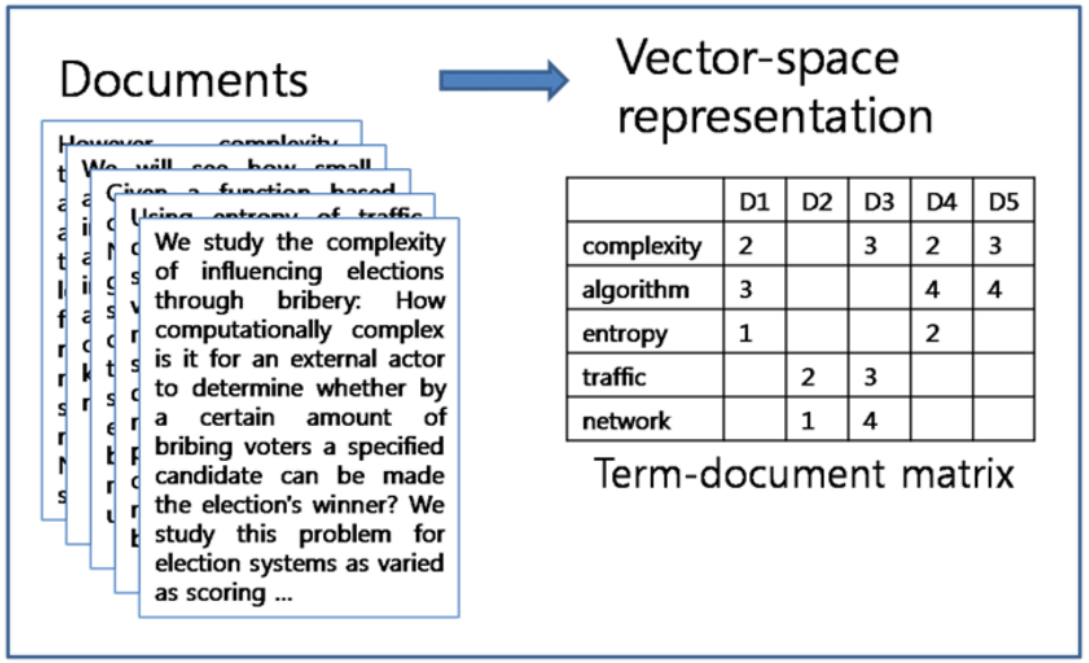


Feature extraction

Methods for vector representation of list of features of documents

Aim of vector representation:

- represent each text document with a **fixed structured numeric vector** instead of a individual list of relevant tokens



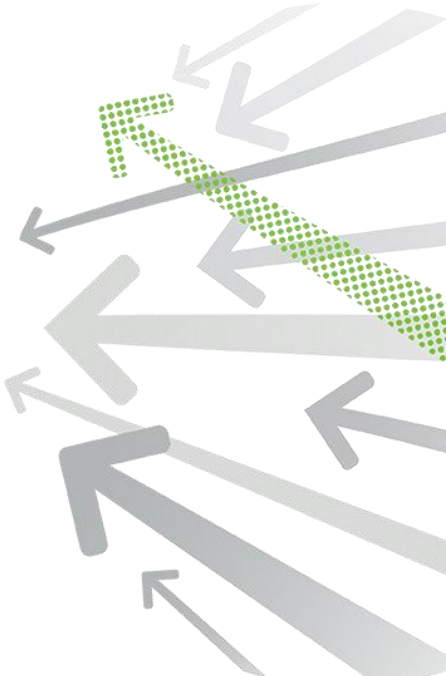
	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Feature extraction

Methods for vector representation of list of relevant tokens of documents

Vectorization process depends on the selected model:

- Traditional feature extraction models
  - → **Binary vectorizer**
  - → **Bag of Words (BoW) Model**
  - → **Term Frequency-Inverse Document Frequency (TF-IDF)**
- Neural-based feature extraction:
  - → **own trained model of neural network**
  - → **pretrained model (BERT)**



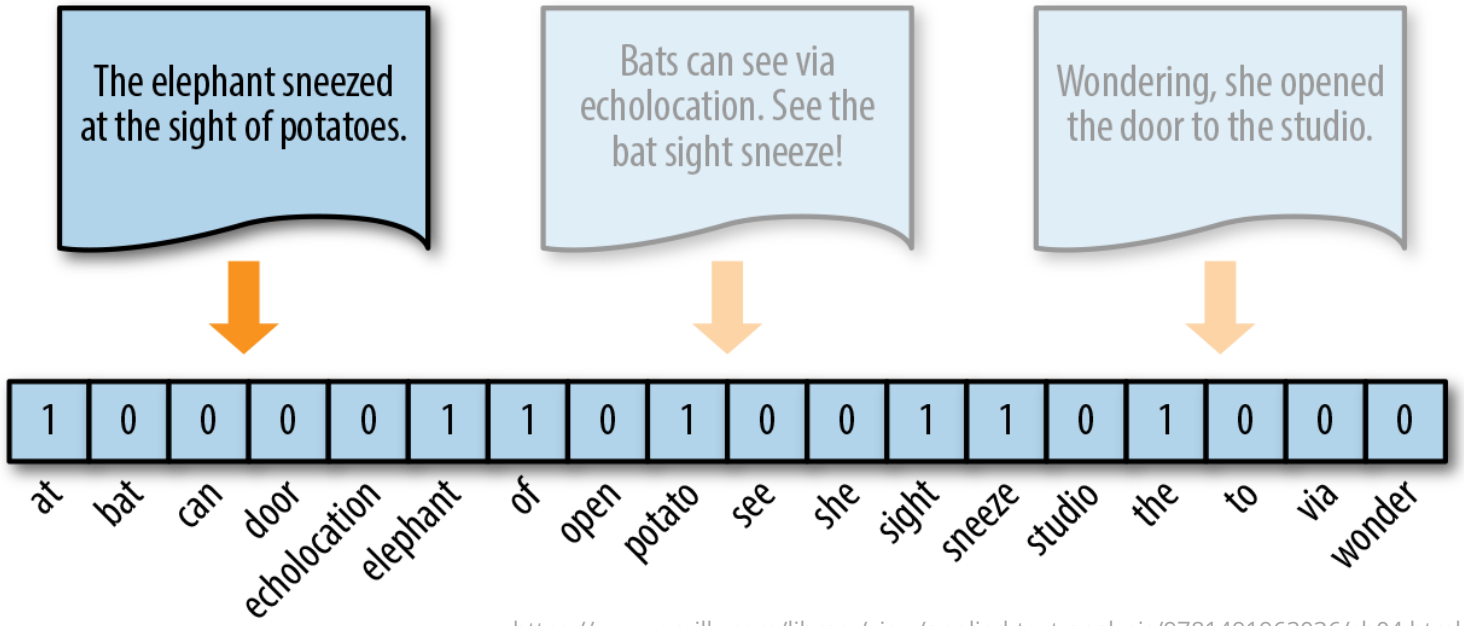
Traditional feature extraction models

Binary vectorizer

- The weight in the vector of the given document expresses the fact whether the given feature from the vocabulary appears in the list of features of the given document or not.

  → If the feature **is** in the features list of the document = **1**

  → If the feature **is not** in the features list of the document = **0**



https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html

Traditional feature extraction models

Binary vectorizer - example

List of features of 3 documents

['movi scari long', 'movi scari slow', 'movi spooki good']

Vocalbulary

['good', 'long', 'movi', 'scari', 'slow', 'spooki']

Calculation of weights of vector for each document

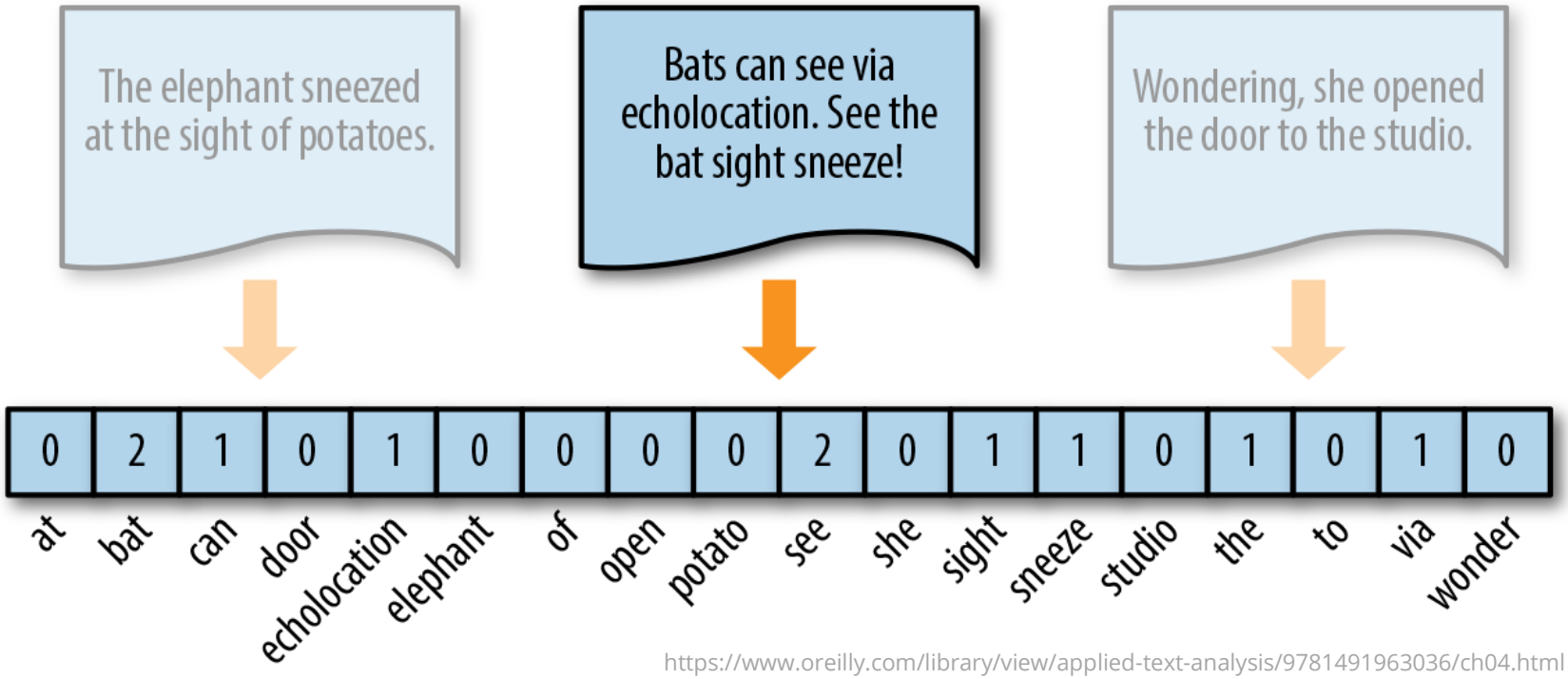
	Feature 1 'good'	Feature 2 'long'	Feature 3 'movi'	Feature 4 'scari'	Feature 5 'slow'	Feature 6 'spooki'
Doc. 1	0	1	1	1	0	0
Doc. 2	0	0	1	1	1	0
Doc. 3	1	0	1	0	0	1

- Example: **Structured vector representation of Doc. 1 = (0, 1, 1, 1, 0, 0)**

Traditional feature extraction models

Bag of Words (BoW) Model

- The weight in the vector of a given document expresses the number of occurrences of the given feature/term from vocabulary in the list of features/terms of the given document.



https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html

Traditional feature extraction models

Bag of Words (BoW) Model - example

List of features of 3 documents

['movi scari long', 'movi scari slow', 'movi spooki good']

Vocalbulary

['good', 'long', 'movi', 'scari', 'slow', 'spooki']

Calculation of weights of vector for each document

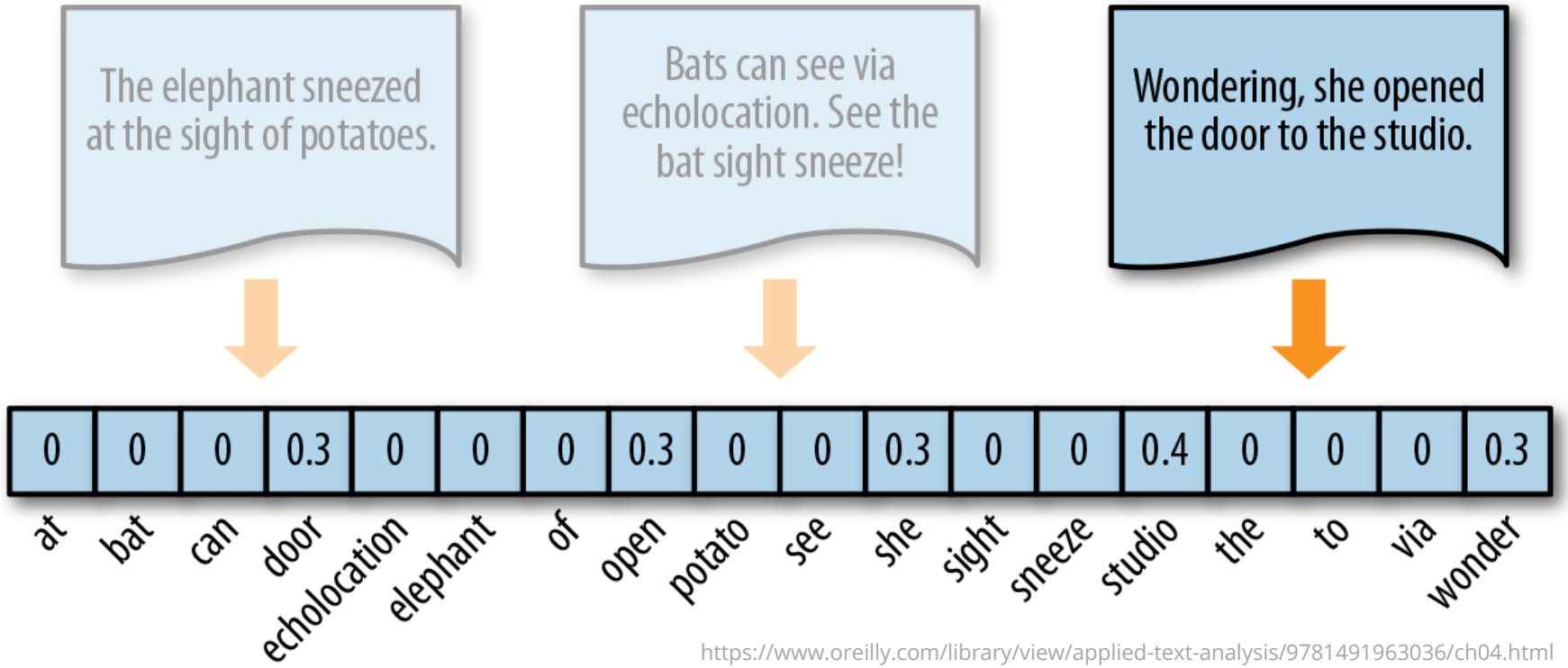
	Feature 1 'good'	Feature 2 'long'	Feature 3 'movi'	Feature 4 'scari'	Feature 5 'slow'	Feature 6 'spooki'
Doc. 1	0	1	1	1	0	0
Doc. 2	0	0	1	1	1	0
Doc. 3	1	0	1	0	0	1

- Example: **Structured vector representation of Doc. 1 = (0, 1, 1, 1, 0, 0)**

Traditional feature extraction models

Term Frequency-Inverse Document Frequency (TF-IDF)

- The weight in the vector of a given document expresses the weight of individual feature/term from vocabulary in the document, **in the context of all documents**.



https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html

Traditional feature extraction models

Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF)

- Unlike the BoW model, it represents **a more sophisticated approach** to creating vector representations of lists of features of the original documents.
- During the calculation of the vector weights of a given document, this approach does not take into account only the given document (individual list of features), but takes into account **the entire document base** (all lists of features).

- Approach intuition:

→ if the given feature occurs in the given document, but also **in all others**, then the weight of the given feature will be **negligible** in the given document

→ if the given feature occurs in the given document and **in no other**, then the weight of the given feature will be **significant** in the given document

Traditional feature extraction models

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

Calculation of weights of vector for each document

	Doc. 1	Doc. 2	Doc. 3	TF 1	TF 2	TF 3	IDF	TF-IDF 1	TF-IDF 2	TF-IDF 3
Feature 1 'good'	0	0	1	0	0	1/3	0.48	0	0	0.16
Feature 2 'long'	1	0	0	1/3	0	0	0.48	0.16	0	0
Feature 3 'movi,	1	1	1	1/3	1/3	1/3	0	0	0	0
Feature 4 'scari	1	1	0	1/3	1/3	0	0.18	0.06	0.06	0
Feature 5 'slow'	0	1	0	0	1/3	0	0.48	0	0.16	0
Feature 6 'spooki'	0	0	1	0	0	1/3	0.48	0	0	0.16

$$TF_{i,j} = \frac{num.\,of\ occur.\,of\ j\ feature\ in\ i\ document}{number\ of\ features\ in\ i\ document}$$

$$IDF_j = \log\left(\frac{num.\,of\ documents}{num\ of\ documents\ with\ feature\ j}\right)$$

$$TF - IDF_{i,j} = TF_{i,j} * IDF_j$$

- Example: **Structured vector representation of Doc. 1 = (0, 1, 1, 1, 0, 0)**

- The procedure for calculating weights using the BoW model in Python is available in section „**Term Frequency-Inverse Document Frequency (TF-IDF)**“ of hands-on „**Text Analytics**“

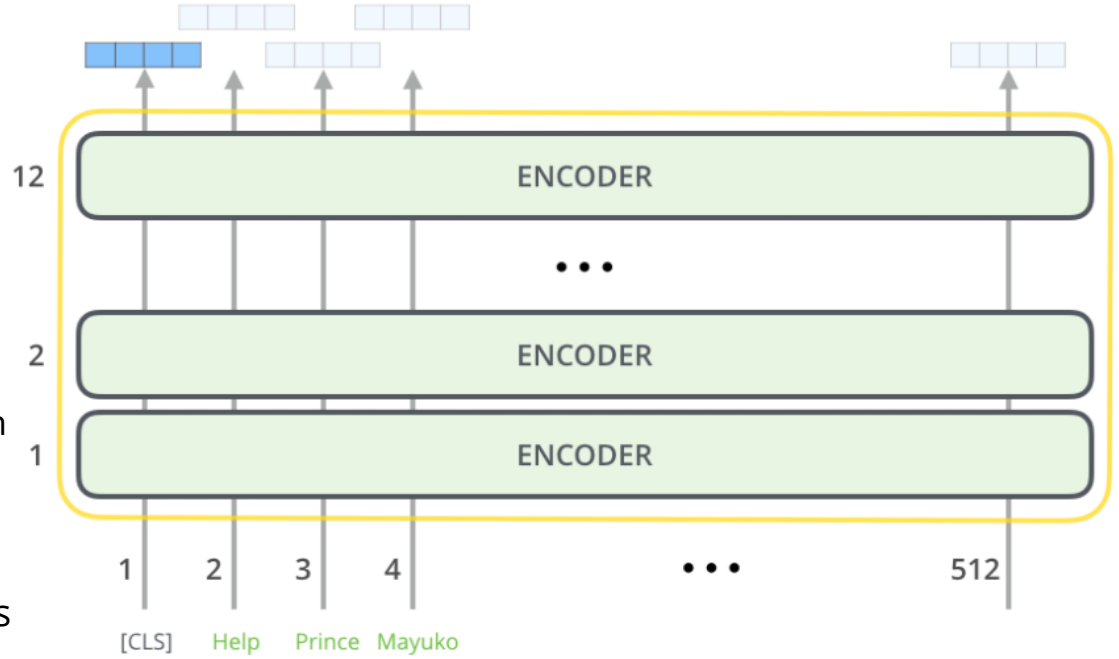
Neural-based feature extraction

Use of a pre-trained neural network model (BERT) for feature extraction purposes

Bidirectional Encoder Representations from Transformers (BERT)

- a pre-trained NLP model developed by Google in 2018
- the model has a wide application across NLP tasks

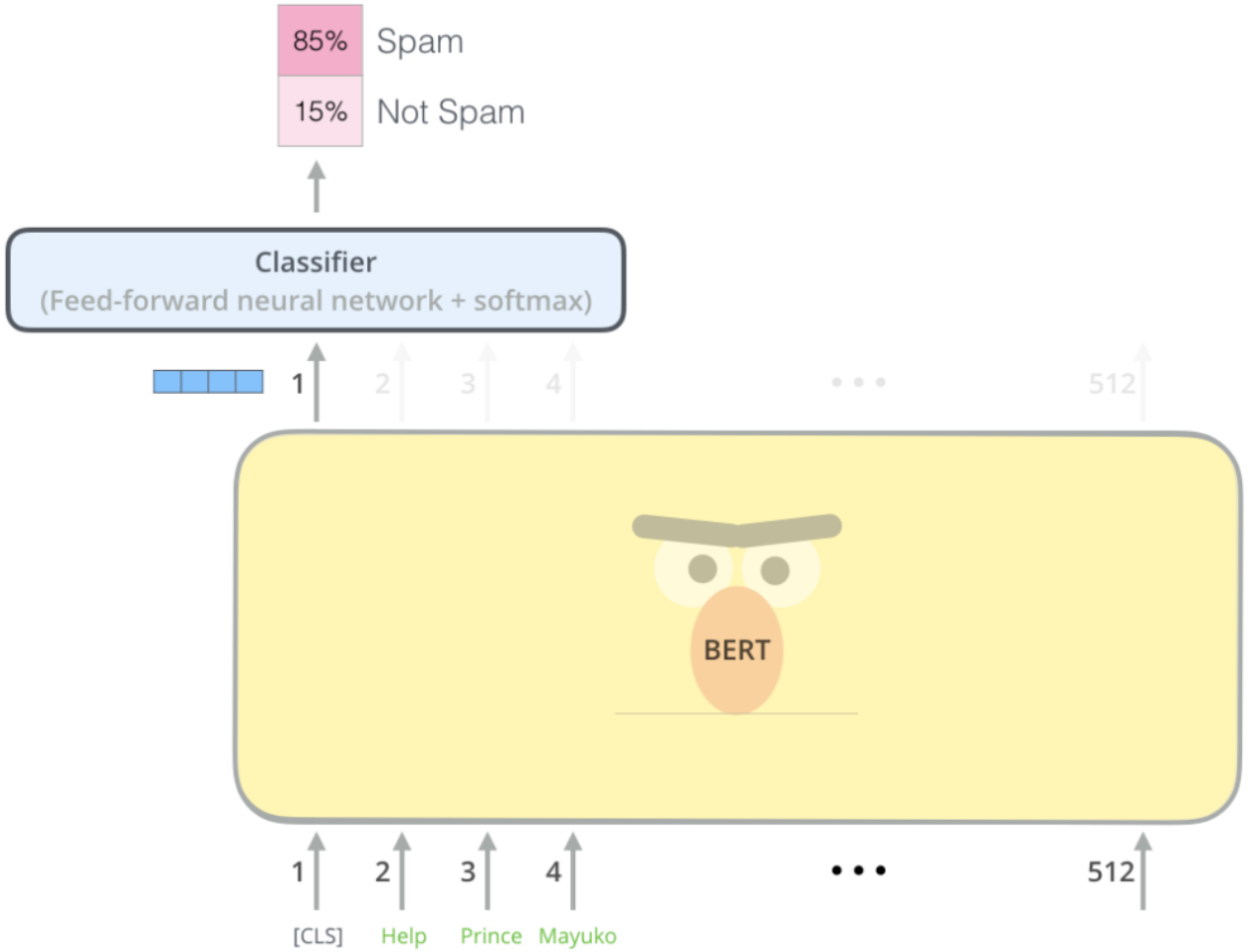
- works as **Transformer Encoder:**
  - **Word embedding**
    - transforms tokens into its vector representation
    - each token == vector of 768 weight
  - **Sentence/document embedding**
    - transforms sentence/dokument into its vector representation
    - each doc == vector of 768 weight
- [CLS] vector
  - represents a **vector representation** of the original documents



http://jalammar.github.io/illustrated-bert/

Neural-based feature extraction

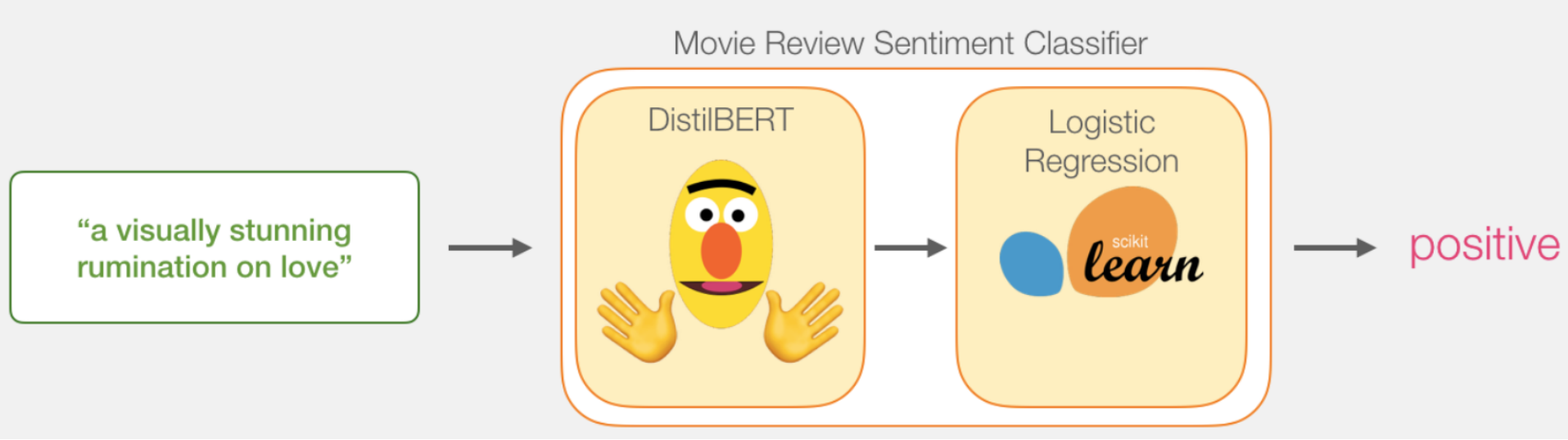
Use of a pre-trained neural network model (BERT) for feature extraction purposes



http://jalammar.github.io/illustrated-bert/

Sentiment Analysis

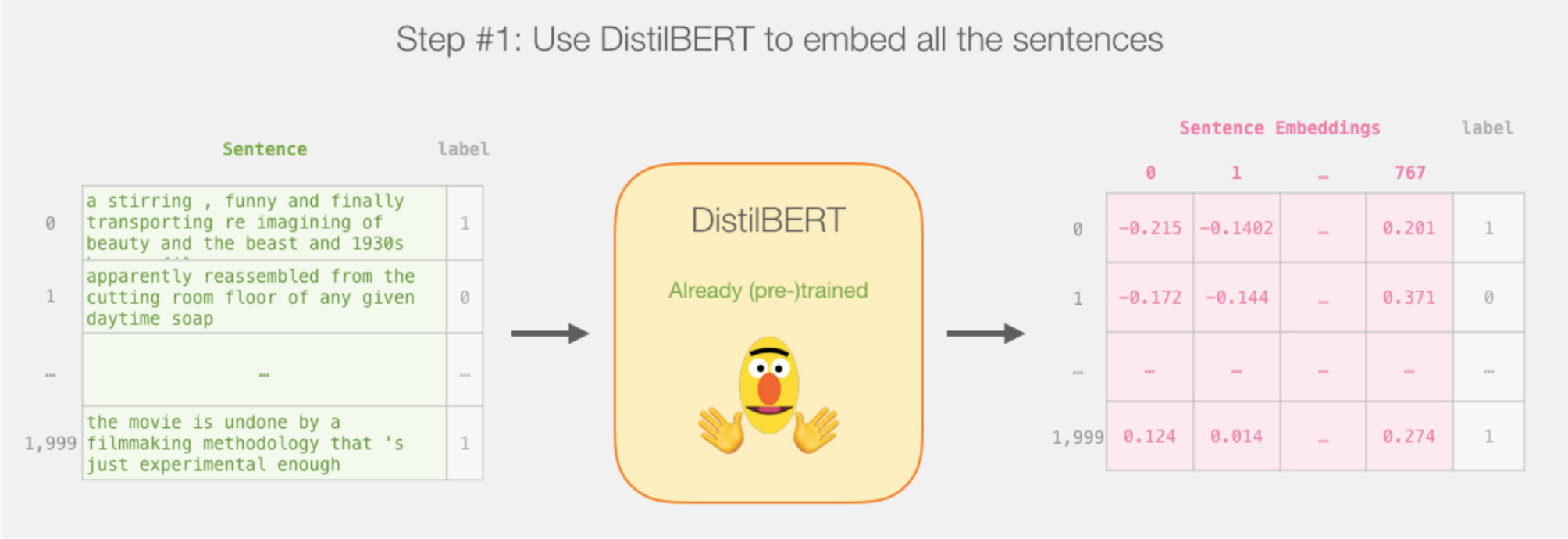
Use traditional ML method over extracted features with a pre-trained neural network model (DistilBERT)



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Sentiment Analysis

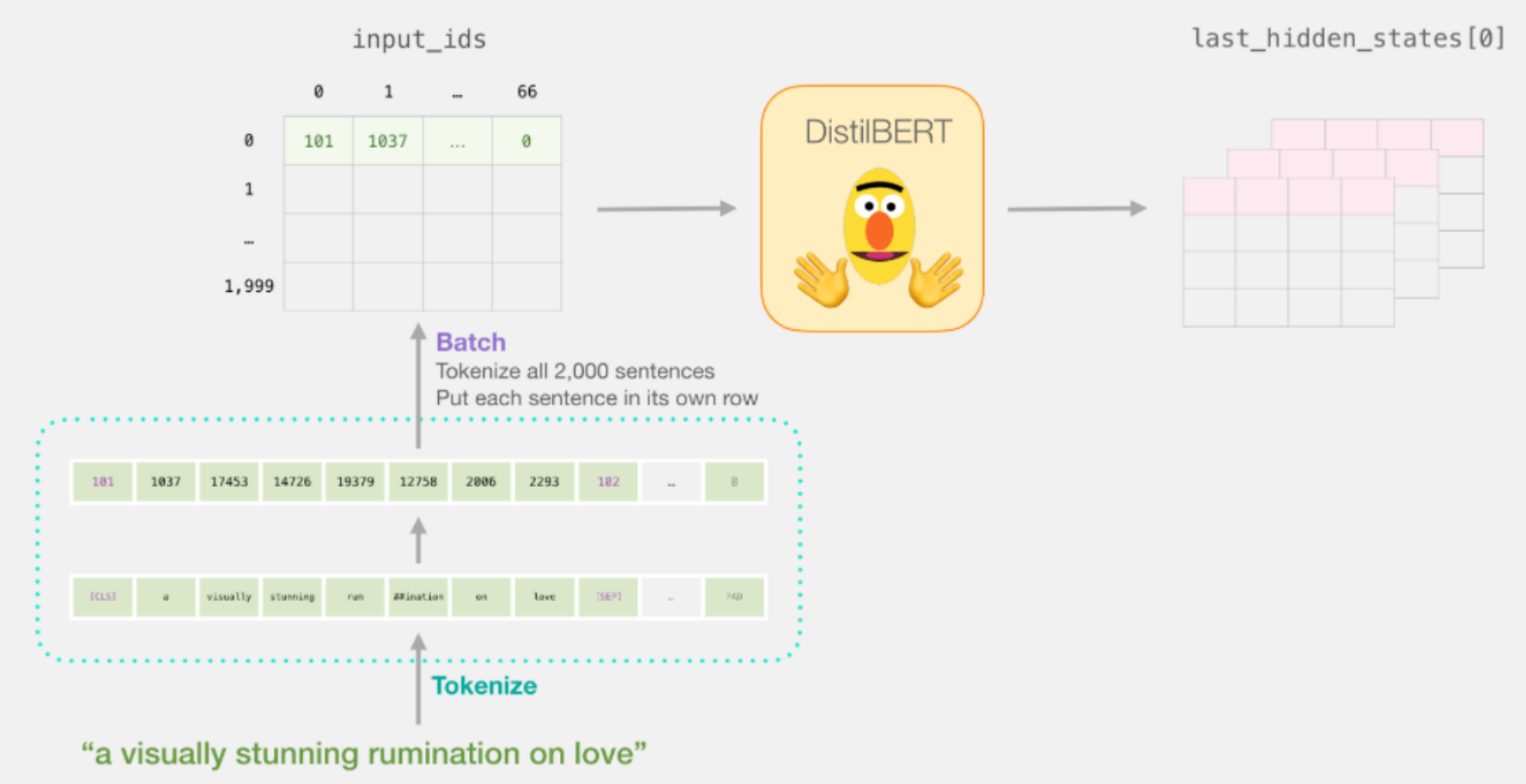
1, Use DistilBERT for feature extraction



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Sentiment Analysis

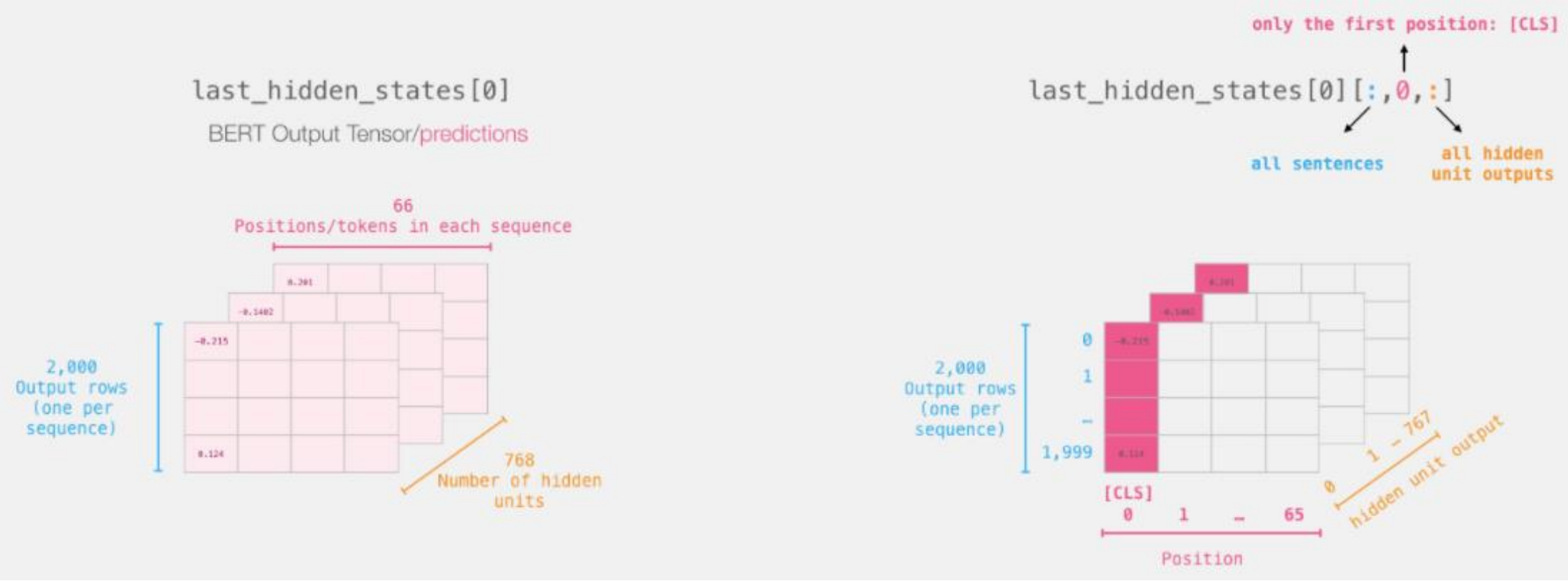
1, Use DistilBERT for feature extraction



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Sentiment Analysis

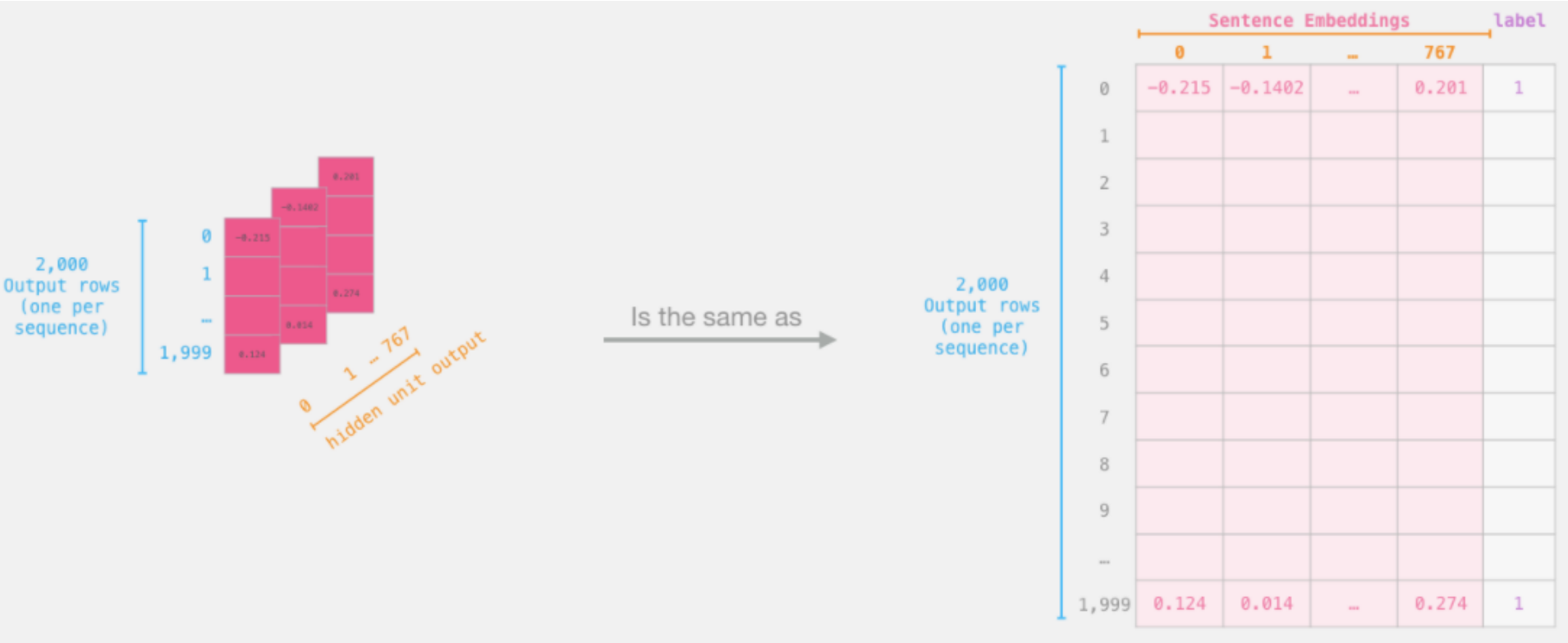
1, Use DistilBERT for feature extraction



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Sentiment Analysis

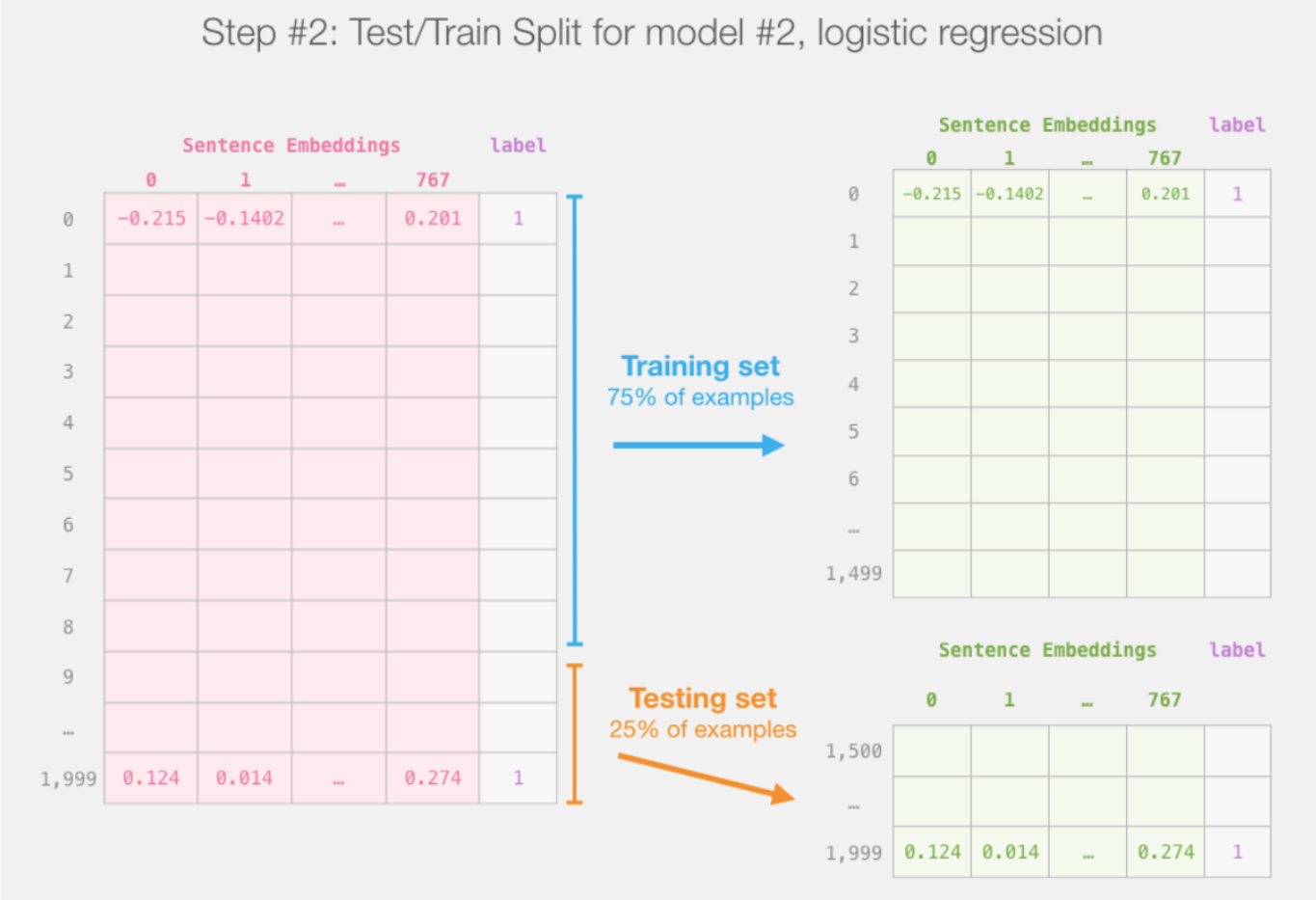
1, Use DistilBERT for feature extraction



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Sentiment Analysis

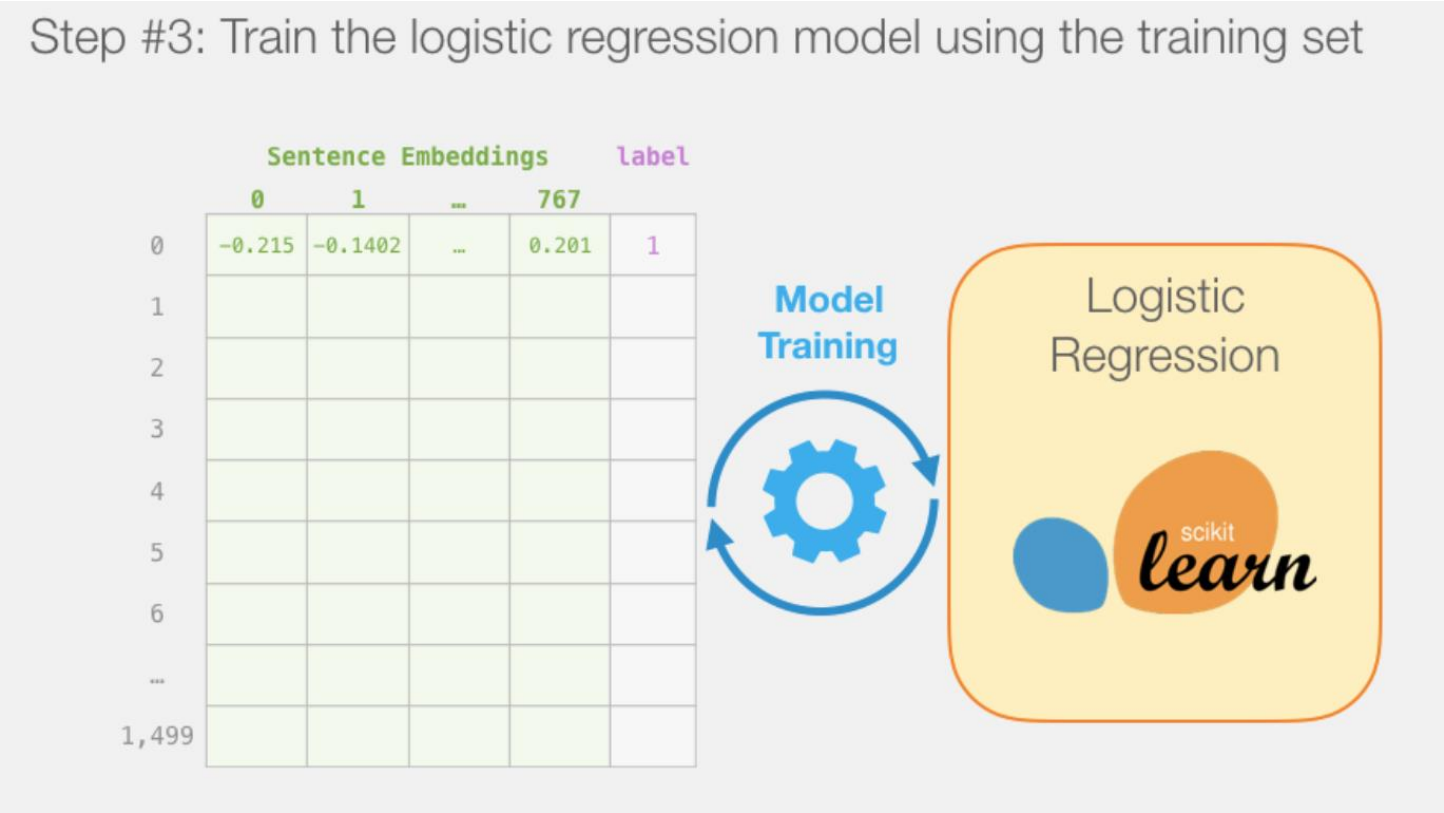
2, Splitting of the dataset into training and testing part



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Sentiment Analysis

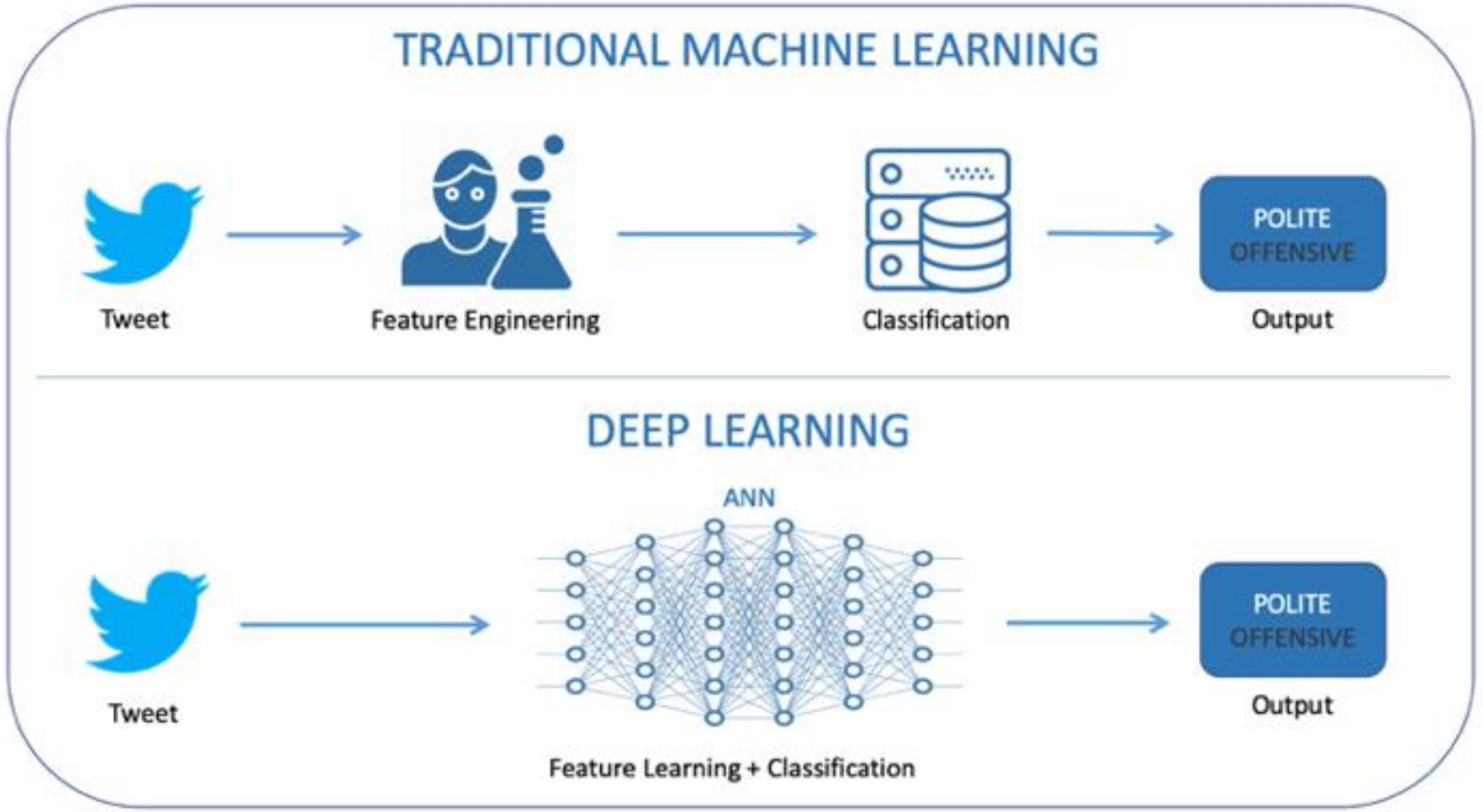
3, Logistic regression model training



https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Neural-based ML methods over text data

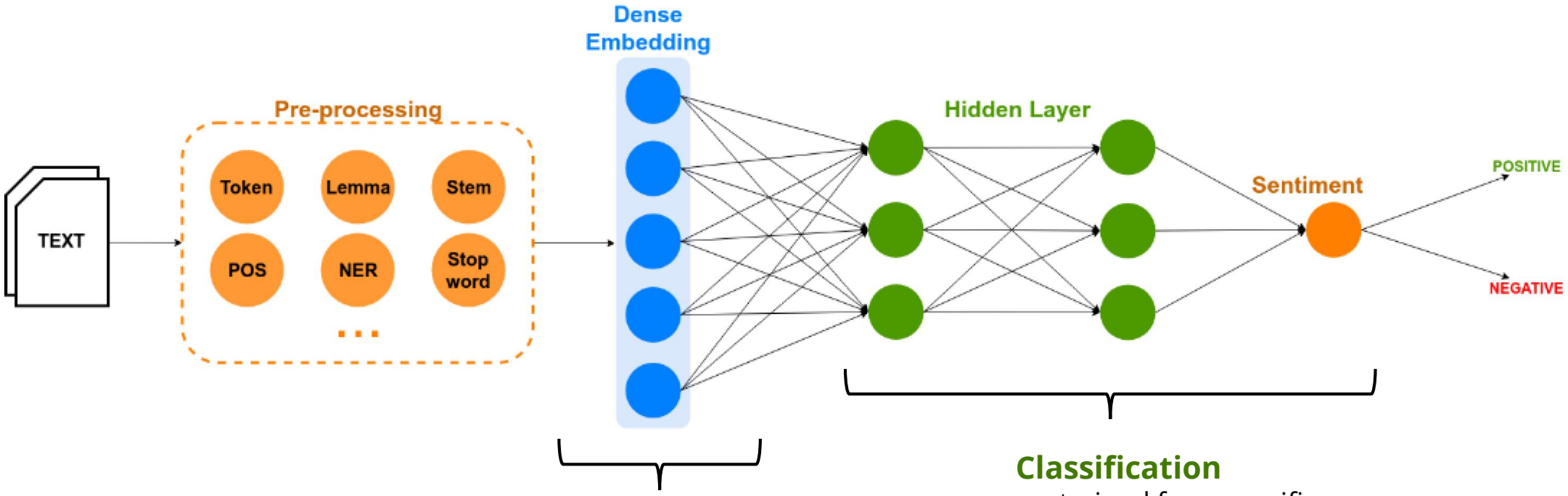
Comparison of traditional ML with Deep Learning



https://eastbanctech.com/technology-insights/what-the-tech/demystifying-deep-learning-and-artificial-intelligence/

Neural-based ML methods over text data

Workflow



Feature extraction

- pre-trained layers (BERT,...)

Classification

- trained for a specific purpose

https://www.researchgate.net/publication/341998176_Sentiment_Analysis_Based_on_Deep_Learning_A_Comparative_Study

Image Analytics

Computer vision (CV)

Computer vision

Definition of CV

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos.

Aim

- Computer vision system ≈ Human (brain) vision system

Computer vision

Use cases

Self-Driving Cars

Healthcare

Facial Recognition

Augmented Reality & Mixed Reality

Use cases

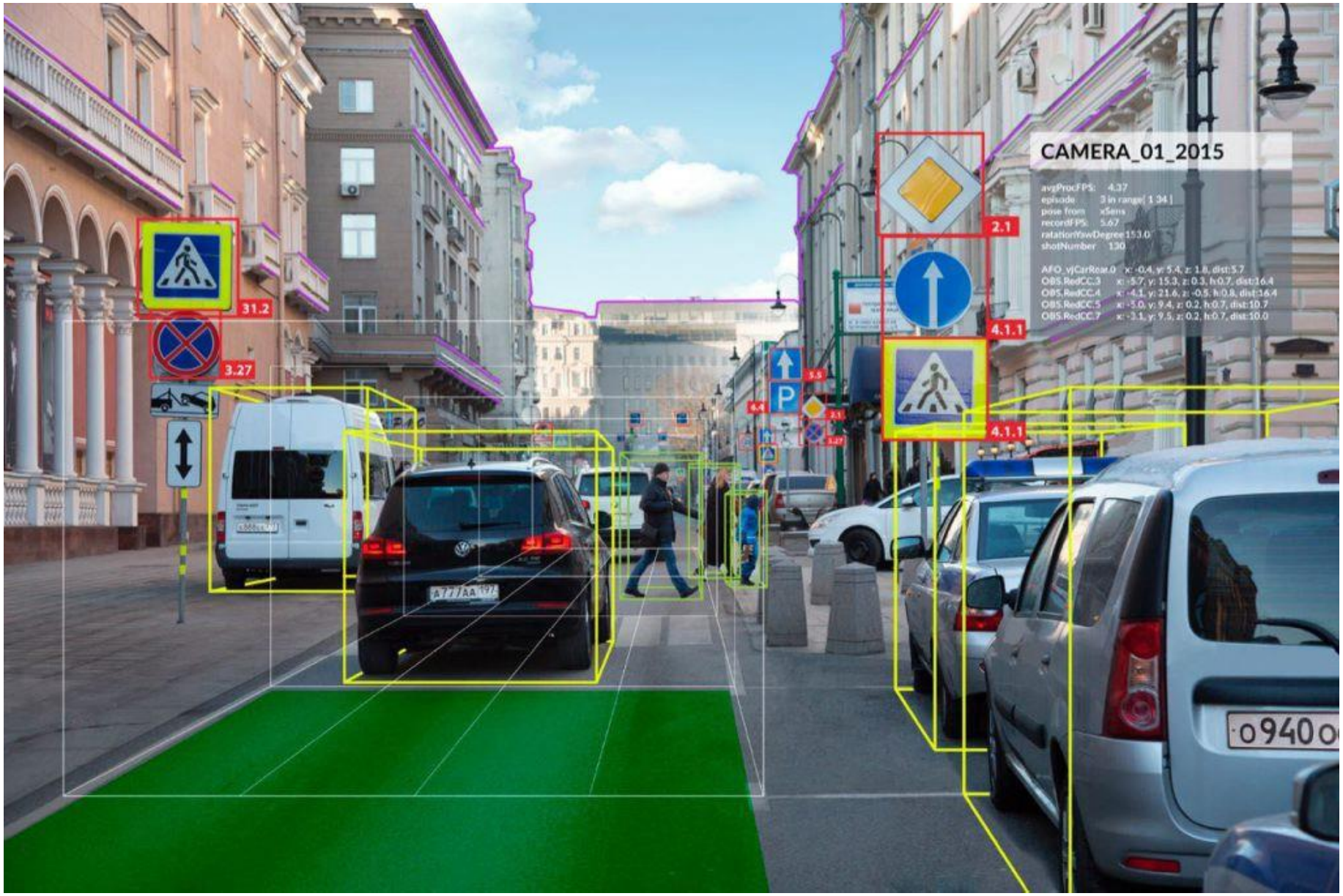
Self-Driving Cars

CV enables self-driving cars to make sense of their surroundings

Process

1. cameras capture video from different angles around the car
2. feed it to computer vision software
3. processes the images in real-time to find the extremities of roads, read traffic signs, detect other cars, objects and pedestrians.

- self-driving car can then steer its way on streets and highways, avoid hitting obstacles, and (hopefully) safely drive its passengers to their destination



https://miro.medium.com/max/1024/1*URl8Xawv6biXul2XoIUwyQ.jpeg

Use cases

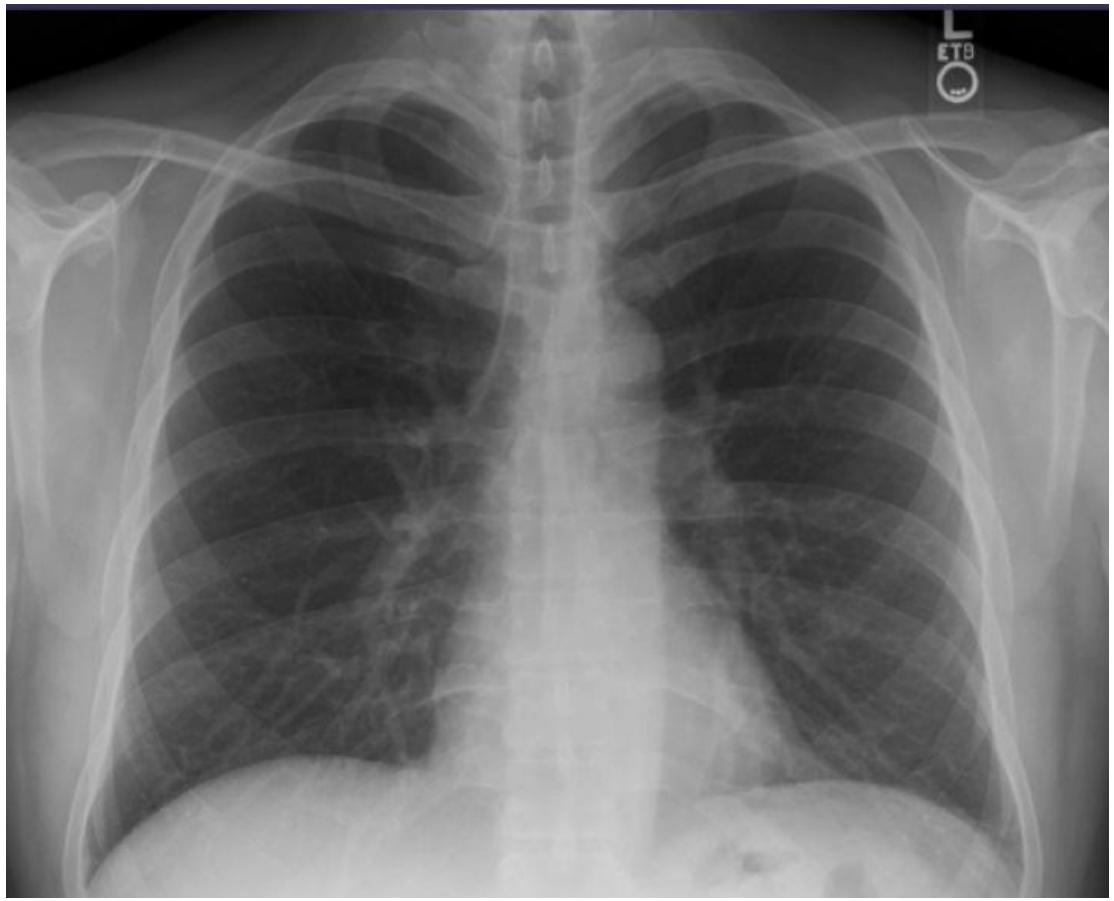
Healthcare

CV enables the automation of the process of disease detection and diagnosis

- **Computer-aided diagnosis systems**
  - CADs assist doctors in the interpretation of medical images and disease detection
  - detecting:
    - cancerous moles in skin images
    - symptoms in x-ray and MRI scans

https://thehealthcaretechnologyreport.com/wp-content/uploads/2020/01/healthcare-ai2.jpg

Pneumonia detection

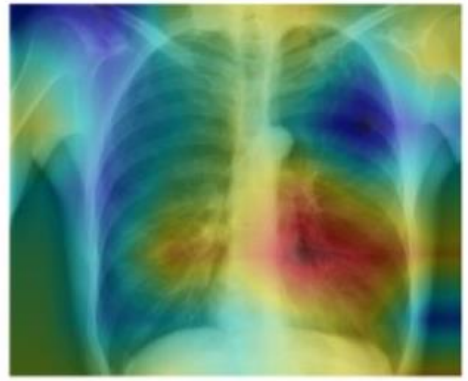


CADs

Input
Chest X-Ray Image

CheXNet
121-layer CNN

Output
Pneumonia Positive (85%)



https://paperswithcode.com/method/chexnet

Use cases

Facial Recognition

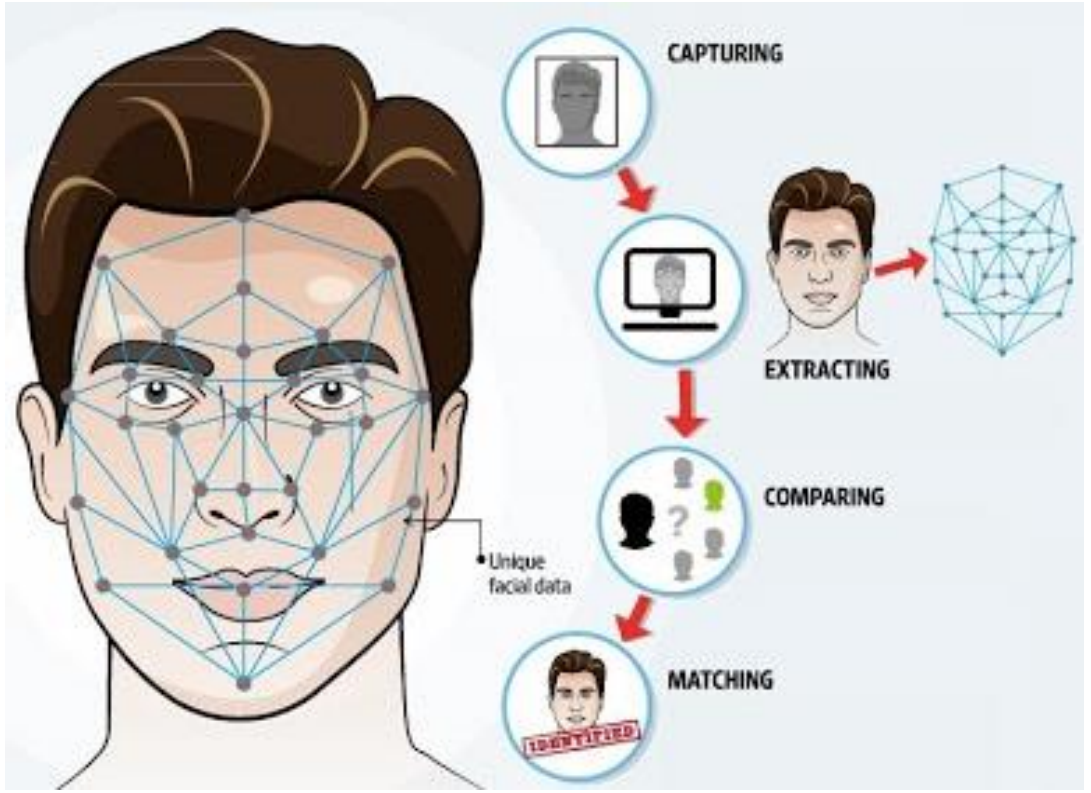
CV enables computers to match images of people's faces to their identities

Principle

- comparing the vector representation of the input face with the vector representations of the faces in the DB

Process

1. Face detection in the image
2. Visual Feature Extraction
3. Vector representation of extracted features
4. Face recognition
   - matching with vector representations of visual features in DB



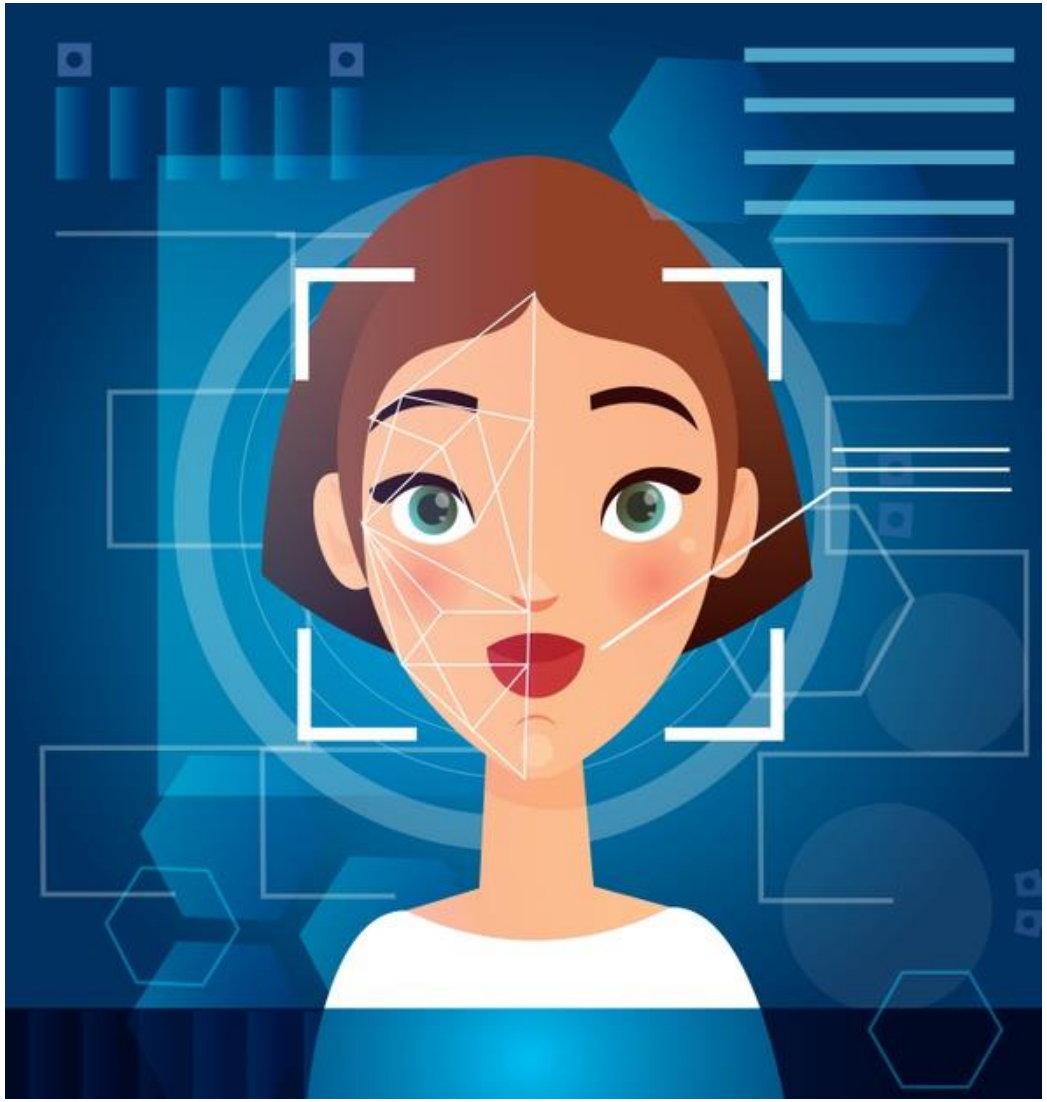
https://medium.com/@fenjiro/face-id-deep-learning-for-face-recognition-324b50d916d1

Use cases

Facial Recognition

Applications

- crime prevention in retail
- unlock mobile devices
- ad targeting
- search for missing persons
- withdrawals from ATMs



Use cases

Augmented Reality & Mixed Reality

CV enables computing devices to overlay and embed virtual objects on real world imagery

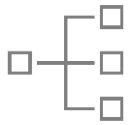
Process

1. detect objects and it's location in real world
2. placing virtual objects in physical world

https://miro.medium.com/max/1200/1*VvJNJxDWlLB4ZIn-L94viA.png

Computer vision

Types of CV tasks



Classification

Classification + Localization

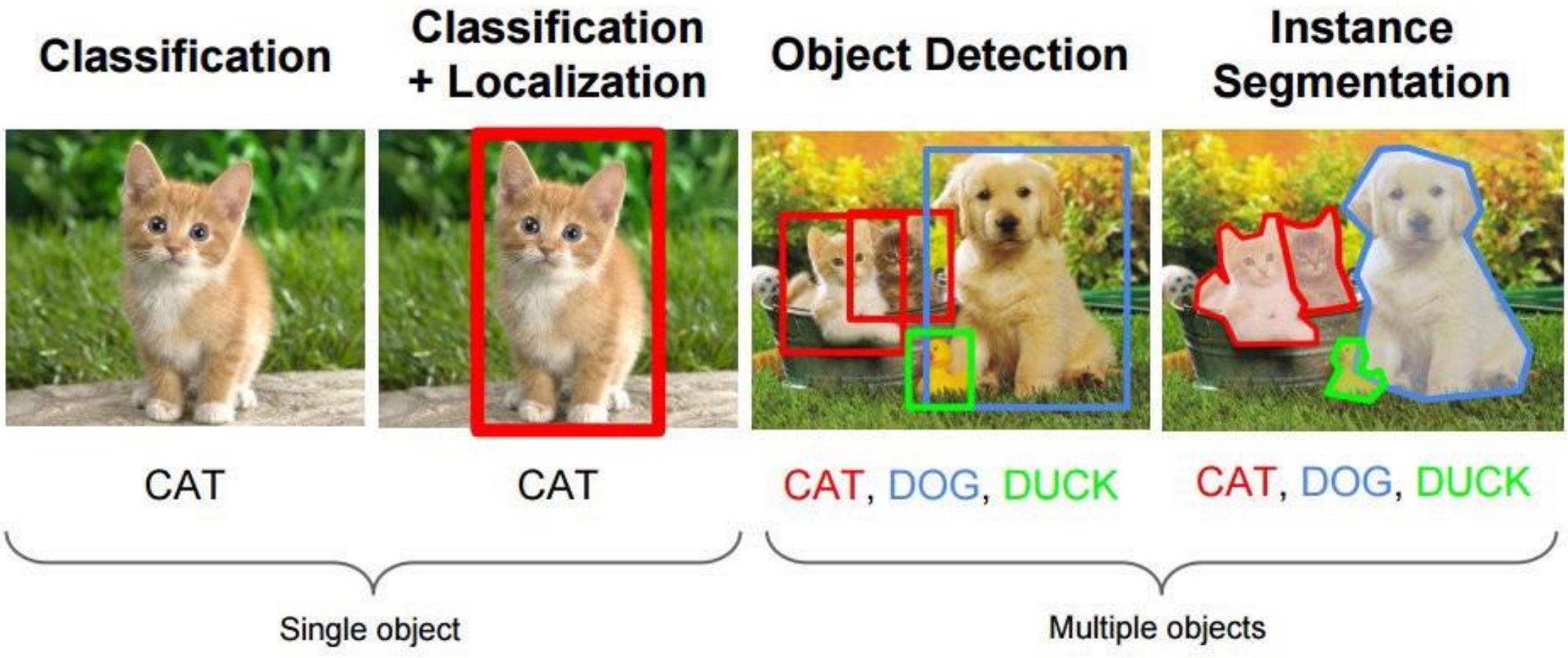
Object detection

Image segmentation

Single object

Multiple objects

Computer vision tasks



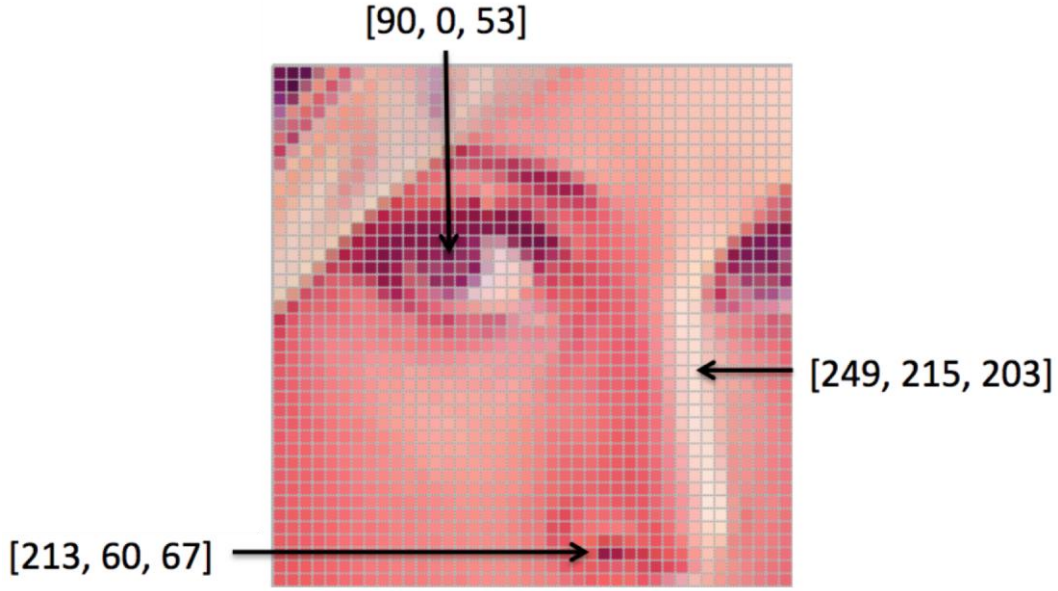
https://miro.medium.com/max/945/1*z89KwWbF59XXrsXXQCECPA.jpeg

Image data

Focus on images as data

Images are numbers

- image is a array of pixels
  - each pixel is represented by one/more numbers



https://ai.stanford.edu/~syyeung/cvweb/Pictures1/colorpixels.png

Grayscale image

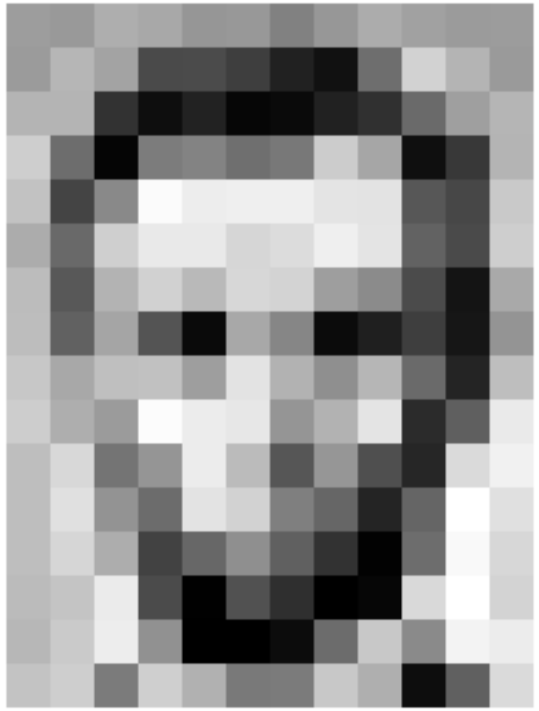
Colored image

Image data

Focus on images as data

Grayscale image

- Each pixel is represented by single number <0;255>



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

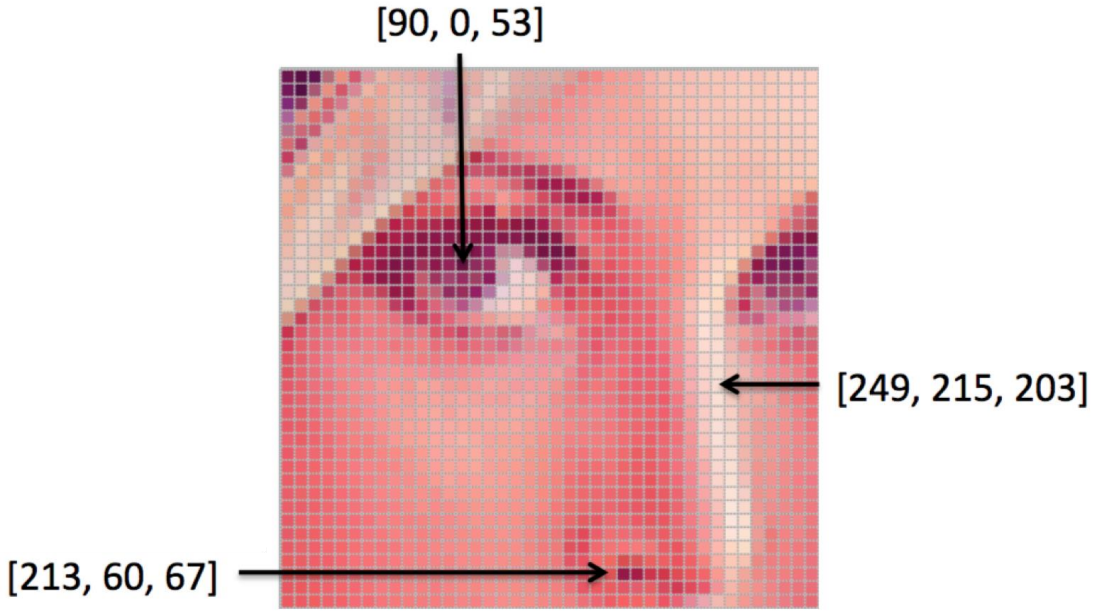
https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e

Image data

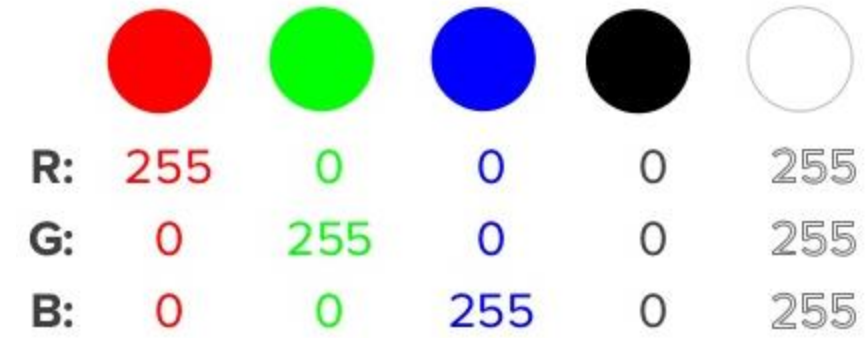
Focus on images as data

Colored image

- Each pixel is represented by three numbers <0;255>
  - three values represent three color channels (RGB)



https://ai.stanford.edu/~syyeung/cvweb/Pictures1/colorpixels.png



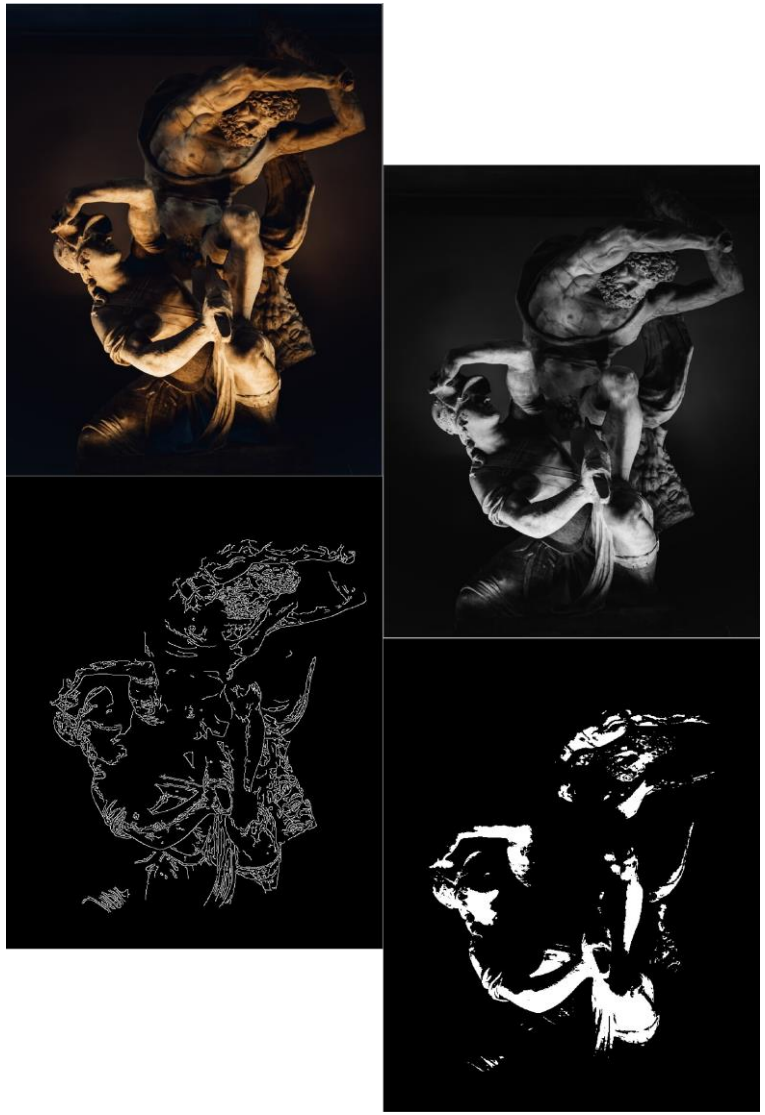
https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e

Image preprocessing

Step before analysis

Two main reasons:

- **input standardization/complexity reduction**
  - Resize image
  - Color conversion to grayscale
  - Geometric Transformations
- **suppressing undesired distortions and enhancing desirable features**
  - reduce the noise
  - Segmentation
  - Pixel brightness transformations



The application of individual image preprocessing techniques **is not automatic** and always depends on the specific data and task!

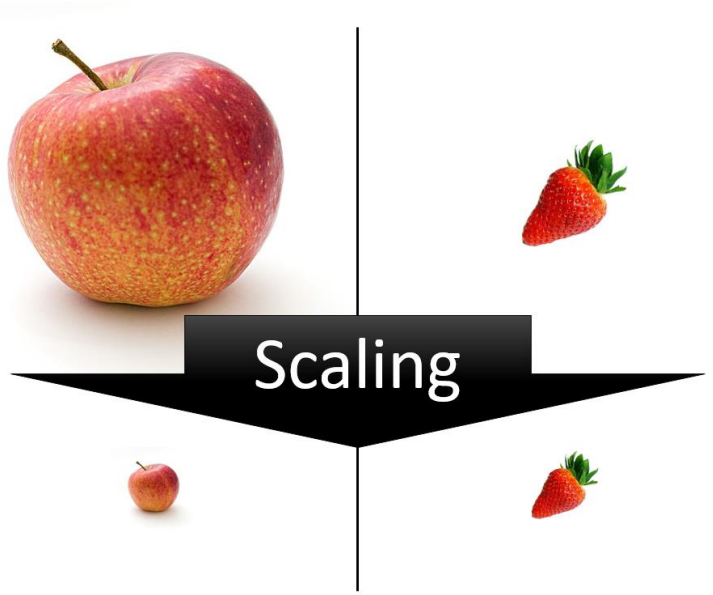
https://miro.medium.com/max/5764/1*Et39W12Lc57sOMxK2feYHQ.png

Image preprocessing

Input standardization/complexity reduction

Input standardization/complexity reduction

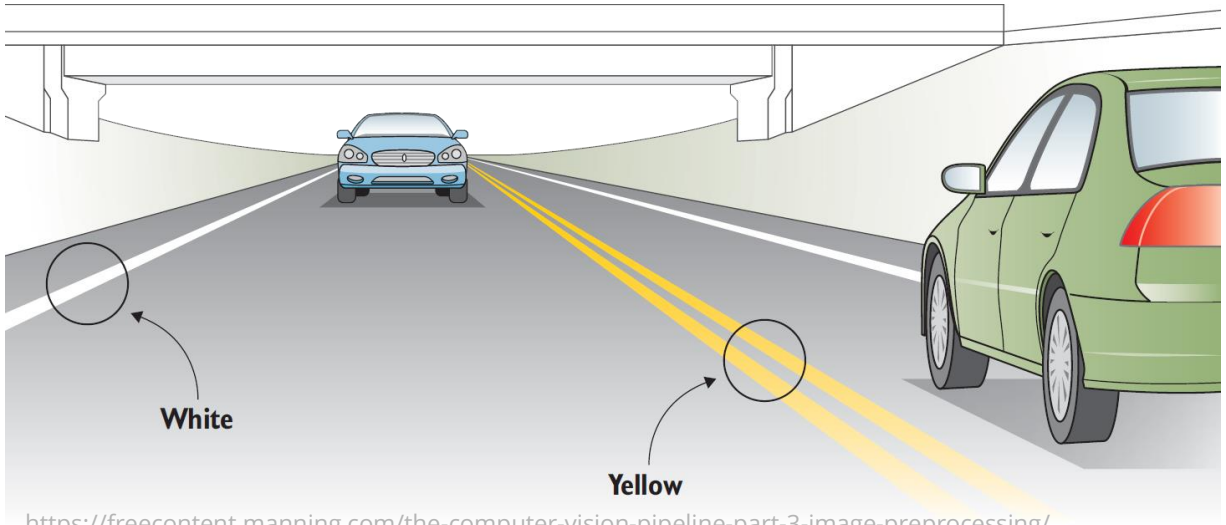
- resize the images to a unified dimension
  - scaling images to identical widths and heights



https://miro.medium.com/max/2625/1*yR54MSI1jjnf2QeGtt57PA.png

- convert color images to grayscale to reduce computation complexity
  - only in use cases where it **isn't necessary** to recognize and interpret color in images

- Geometric Transformations



https://freecontent.manning.com/the-computer-vision-pipeline-part-3-image-preprocessing/

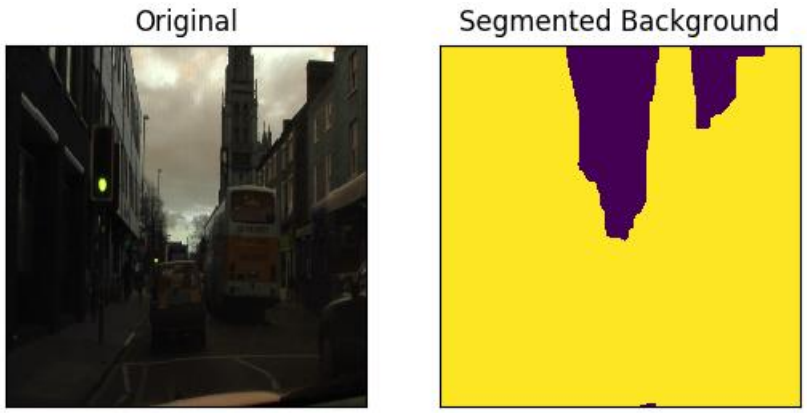
Image preprocessing

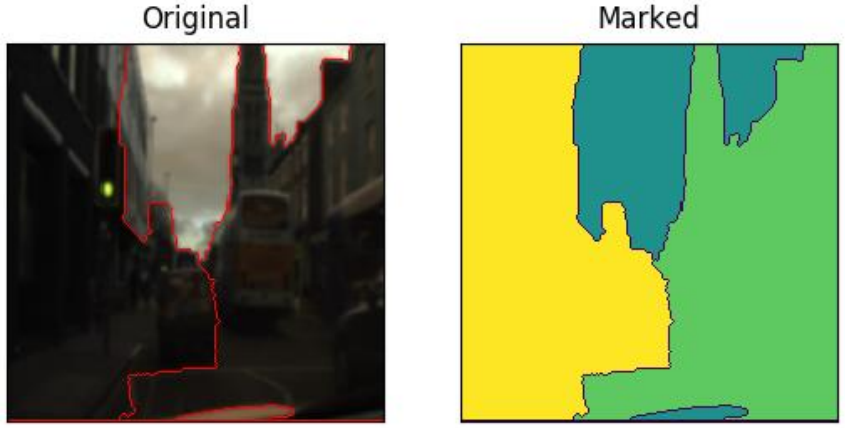
Suppressing undesired distortions and enhancing desirable features

Suppressing undesired distortions and enhancing desirable features

- reduce the noise
  - **Gaussian blur technique** to smooth images to remove unwanted noise

- Segmentation
  - separating the background from foreground objects





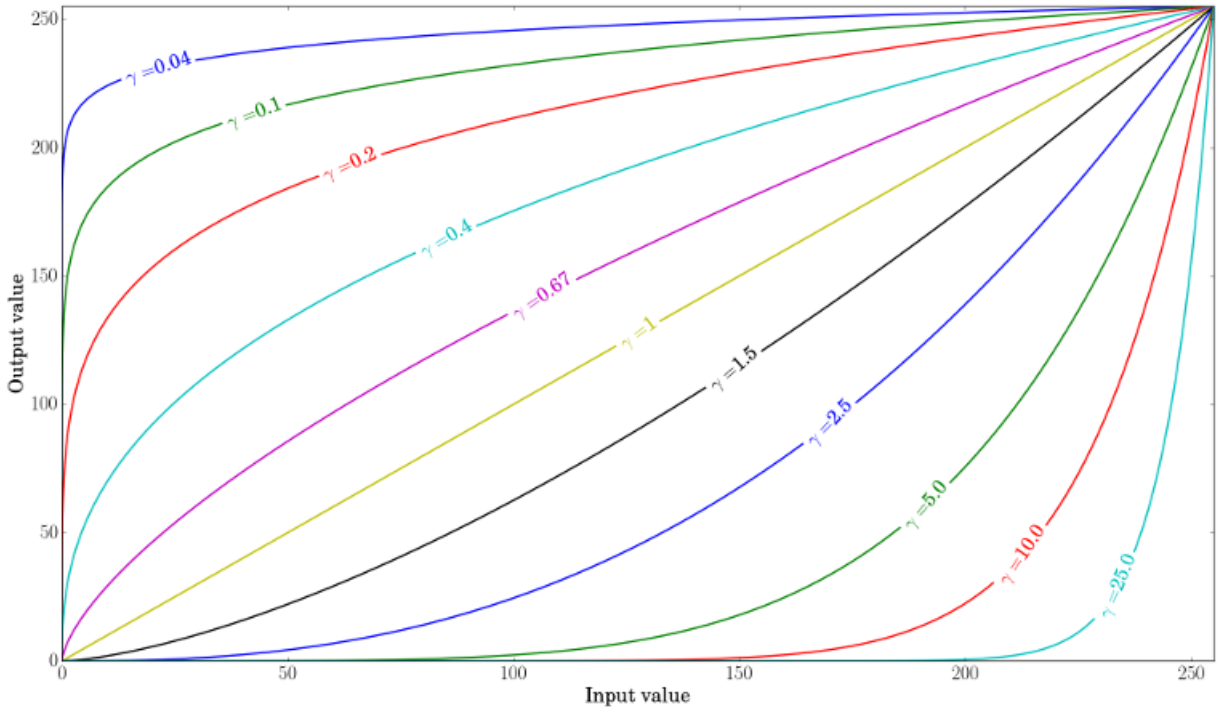
https://towardsdatascience.com/image-pre-processing-c1aec0be3edf

Image preprocessing

Suppressing undesired distortions and enhancing desirable features

Suppressing undesired distortions and enhancing desirable features

- Pixel brightness transformations
  - **Gamma Correction** – is a brightness and contrast adjustment non-linear technique



https://lh6.googleusercontent.com/lkx5IHS4_JEGvP1CXoajLvNOMrskibiqVpEA5hfRQULiPOfVt24fRy6y0WrN-tVKR91y_ITNHAPAjxmRl9vGM2GLlGyUln9UOwBN2O9-pz5CLA8OiYyJYM0dOZYPL2DQoGTxdTcZjLugtDnFmA

Image preprocessing

Gamma Correction

Pixel brightness transformations y = 0.2

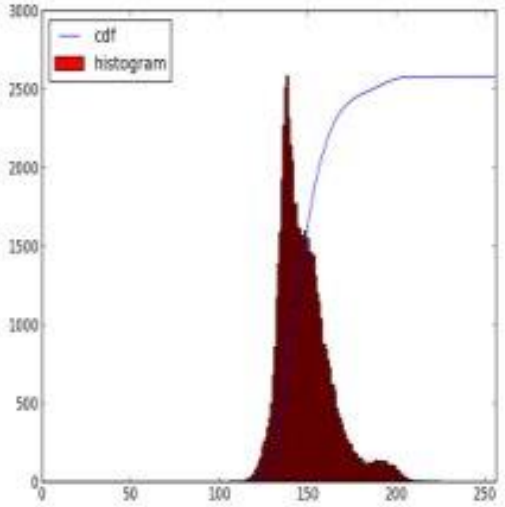
https://lh3.googleusercontent.com/bYa3YeZluYhsiW0QvlzW8NqdFa-XFNhkv3ux8UXIKU7kawkSZGoGbnMmjJqIkkAiAyMdy4F4ETCANMkVIcwWyHQR5GJzGhACg5IuhzrFekf-exlIFrDgoC6ONDvnGepSP_XPwQGfd36th7P4ww

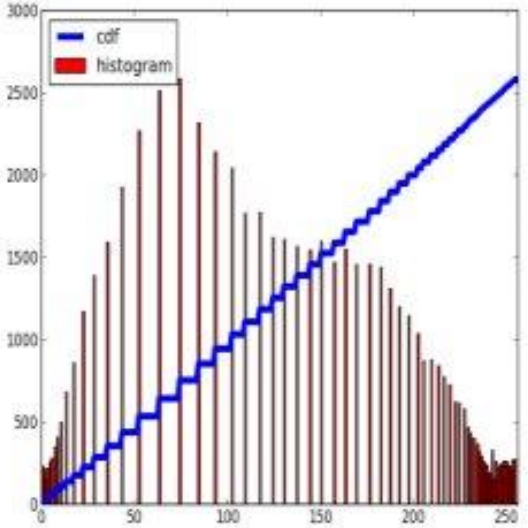
Image preprocessing

Suppressing undesired distortions and enhancing desirable features

Suppressing undesired distortions and enhancing desirable features

- Pixel brightness transformations
  - **Histogram equalization** – contrast enhancement technique
  - sophisticated method for modifying contrast of an image such that its intensity histogram has the desired shape

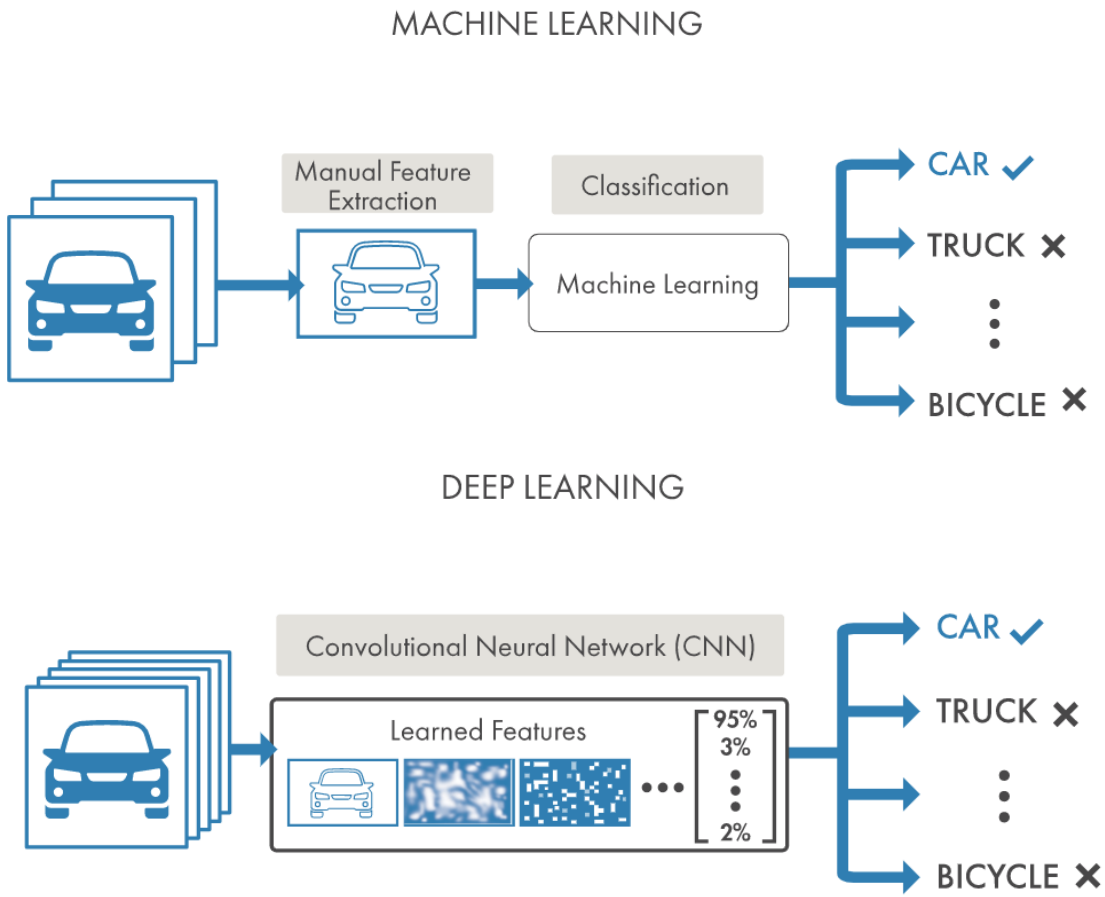




https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_equalization/py_histogram_equalization.html

ML in images classification

Use of ML methods over images data

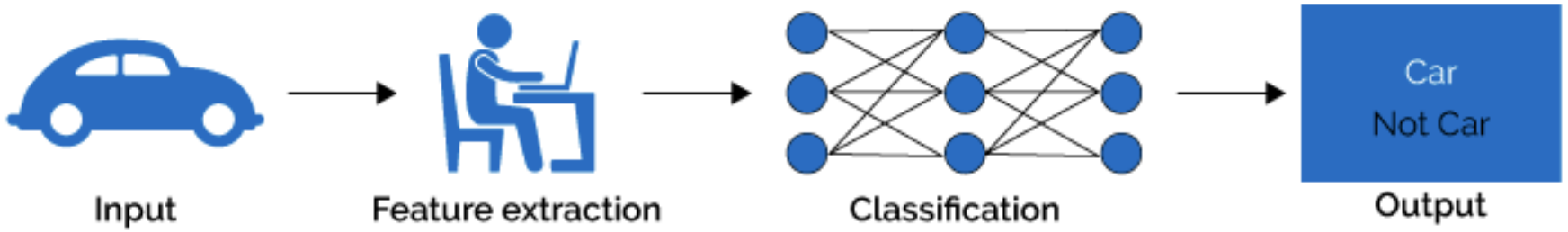


https://www.mathworks.com/discovery/deep-learning.html?fbclid=IwAR3nhHwsNjBlgVWDo5fl5opdWdZ-MUKPJbmwC1uD3aCWsnUGwshqyjWa97E

Traditional ML in images classification

Traditional ML over images

- Separated **extraction features** and **modeling**



- **Feature extraction**
  - Global feature descriptors
  - Local features descriptors
- **Classification**
  - ML classification methods (SVM, Neural net, Logistic regression,...)

Traditional ML in images classification

Feature extraction

- **Aim of feature extraction**
  - transform images into their vector representations
- **How?**
  - extract relevant information from images in terms of numerical values (features vectors)
  - vector representations of an image is derived from the **features vectors**

- **Global features**
  - represent the image as a whole
    - contour representations,
    - shape descriptors,
    - texture features

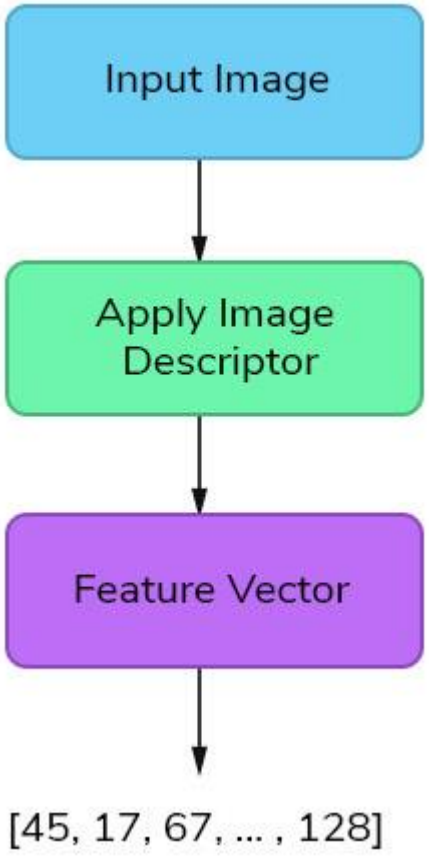
- **Local features**
  - represent key points in the image

Traditional ML in images classification

Global features descriptors

- **CV techniques for Global features extraction**
  - Global feature descriptors
    - **Color** - *Color Channel Statistics* (Mean, Standard Deviation) and *Color Histogram*
    - **Shape** - *Hu Moments, Zernike Moments*
    - **Texture** - *Haralick Texture, Local Binary Patterns (LBP)*

- As part of the feature extraction process, multiple descriptors are typically applied at once
  - each descriptor returns a feature vector
  - concatenate each feature vector to form a **single global feature vector**
  - **Vector representation of an image == single global feature vector**



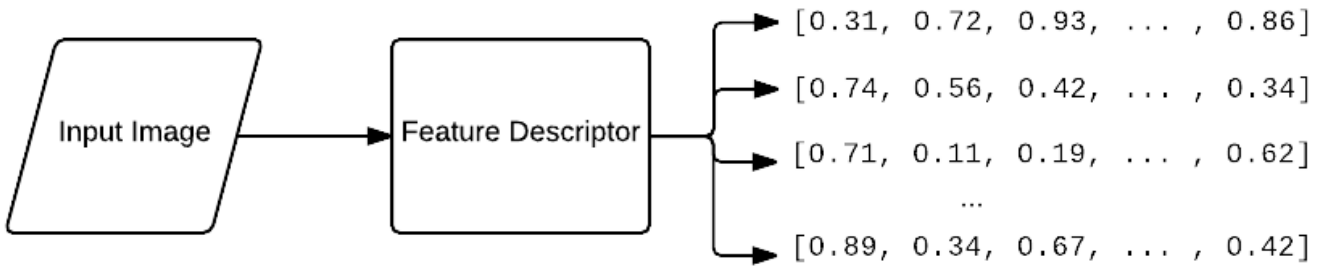
https://gogul.dev/images/software/plants-species/feature_extraction.jpg

Traditional ML in images classification

Local feature descriptors

- **CV techniques for Local features extraction**
  - Local feature descriptors
    - *SIFT* (Scale Invariant Feature Transform)
    - *SURF* (Speeded Up Robust Features)
    - *ORB* (Oriented Fast and Rotated BRIEF)
    - *BRIEF* (Binary Robust Independed Elementary Features)

- Local feature descriptors extracts key points in the image
  - Each extracted point is described by a vector
  - **Image == list of extracted features vectors**



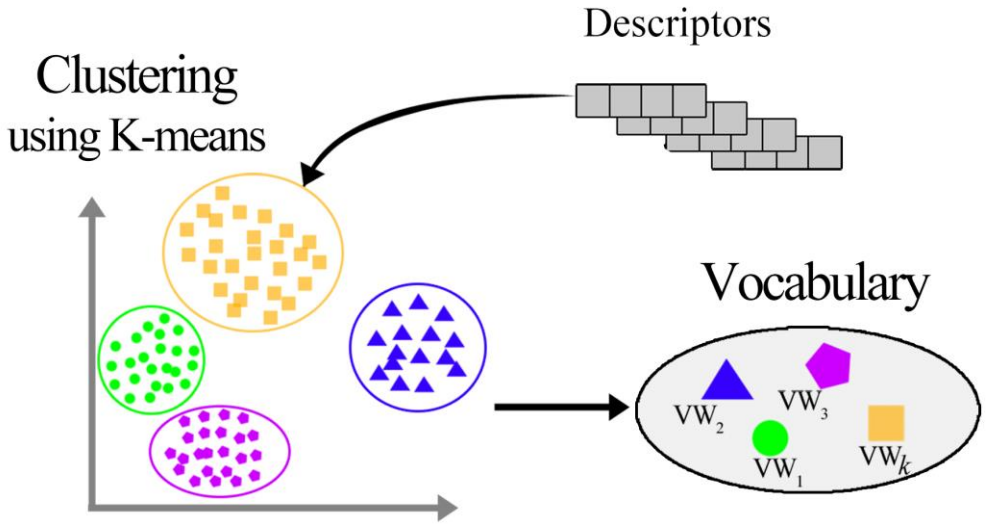
Traditional ML in images classification

Local feature descriptors – from features vectors to vector representation of an image (Bag of visual words)

Bag of visual words

- Similar images have similar features. If we group features on the basis of similarity into groups, we also achieve partial separation of images

1. Dictionary/Vocabulary construction
   - Clustering of all feature vectors from all images in dataset
   - resulting cluster centers (i.e., centroids) are treated as **dictionary of visual words**

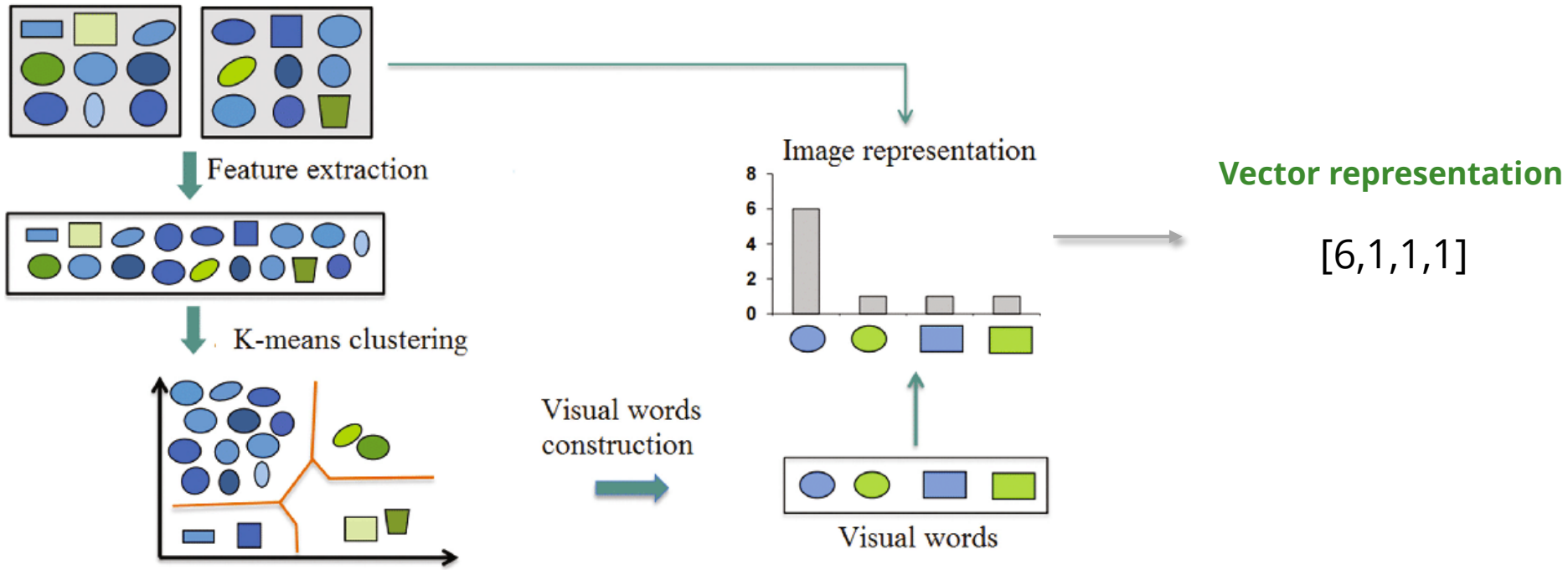


https://ww2.mathworks.cn/matlabcentral/fileexchange/62232-bag-of-visual-words?s_tid=prof_contriblnk

Traditional ML in images classification

Local feature descriptors – from features vectors to vector representation of an image

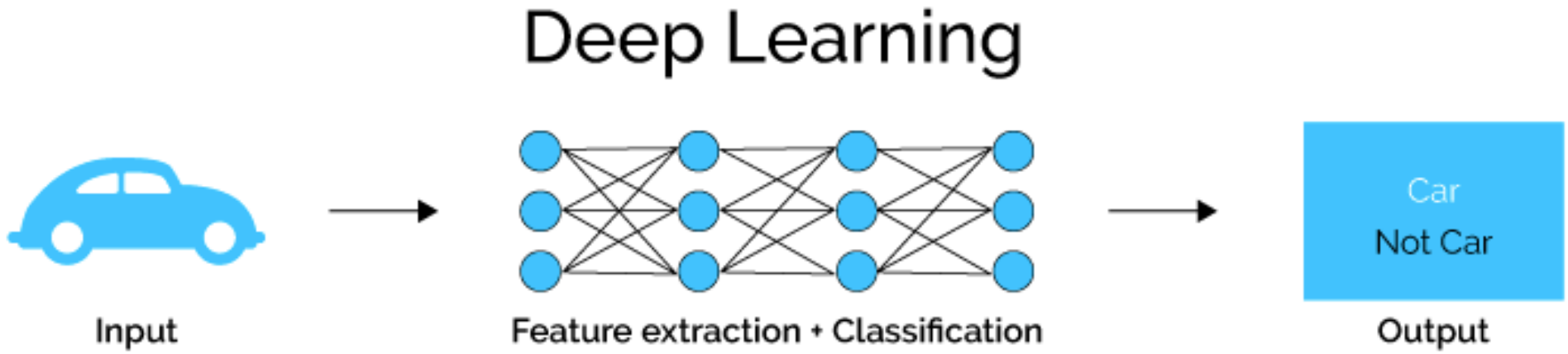
2. Vector quantization
    - for each picture we make:
        - a histogram that expresses the frequency of individual visual words from the dictionaries in the given picture



https://www.researchgate.net/profile/Julio-Leon/publication/318734959/figure/fig3/AS:613946444767316@1523387647886/Bag-of-Visual-Words-scheme.png

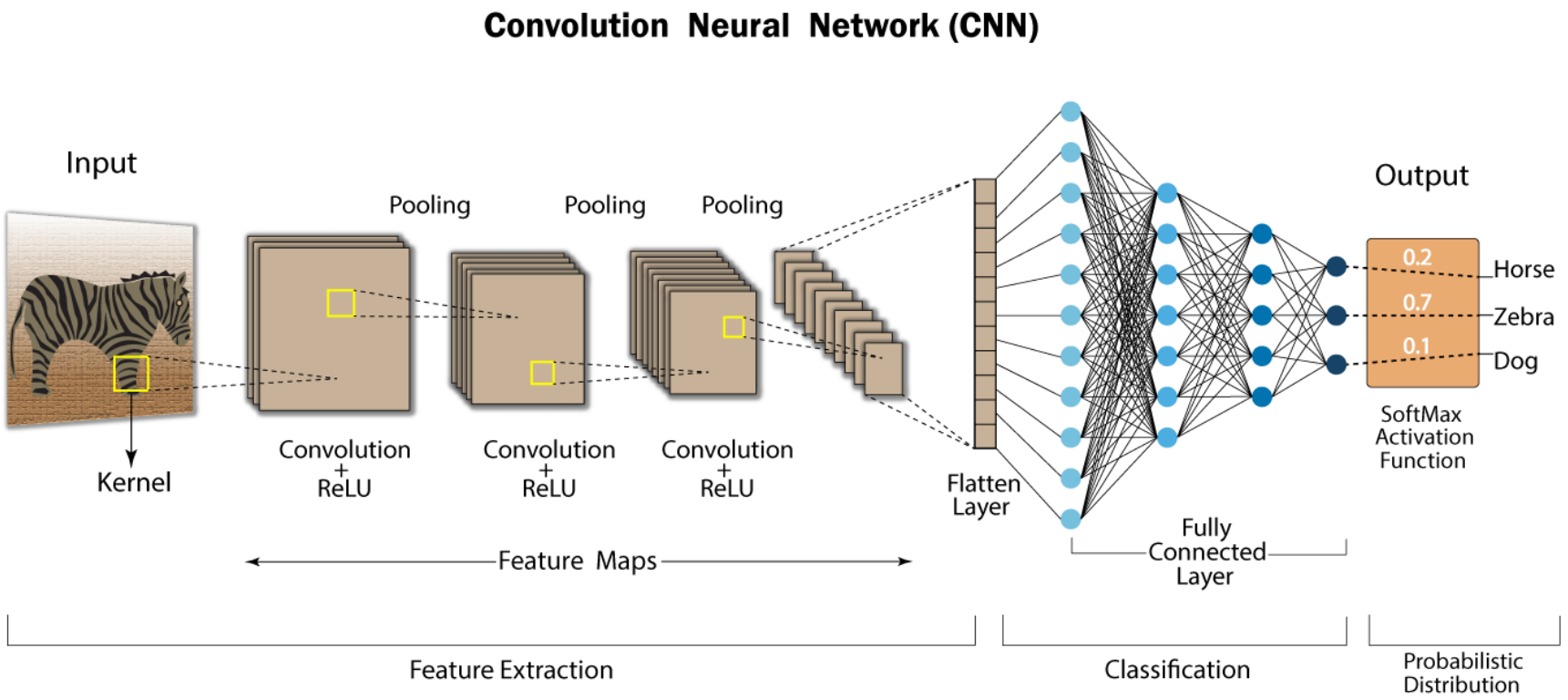
Neural-based ML methods over text data

Deep learning



Neural-based ML methods over images

Convolution NN – principles



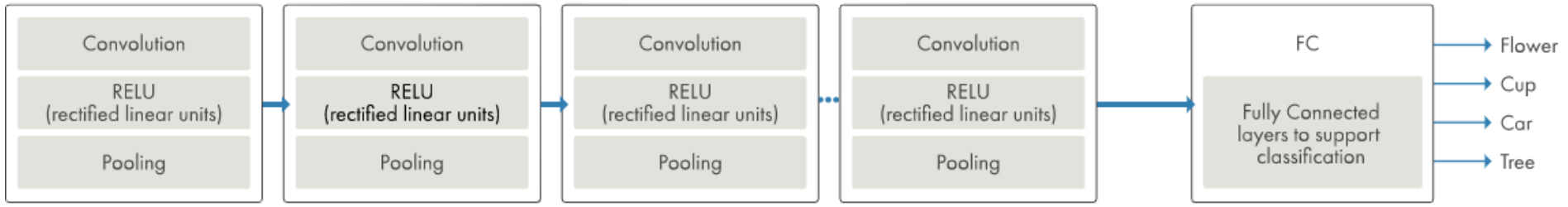
- own trained layers (Conv2D, MaxPooling2D)
- pre-trained layers (MobileNet V2,...)

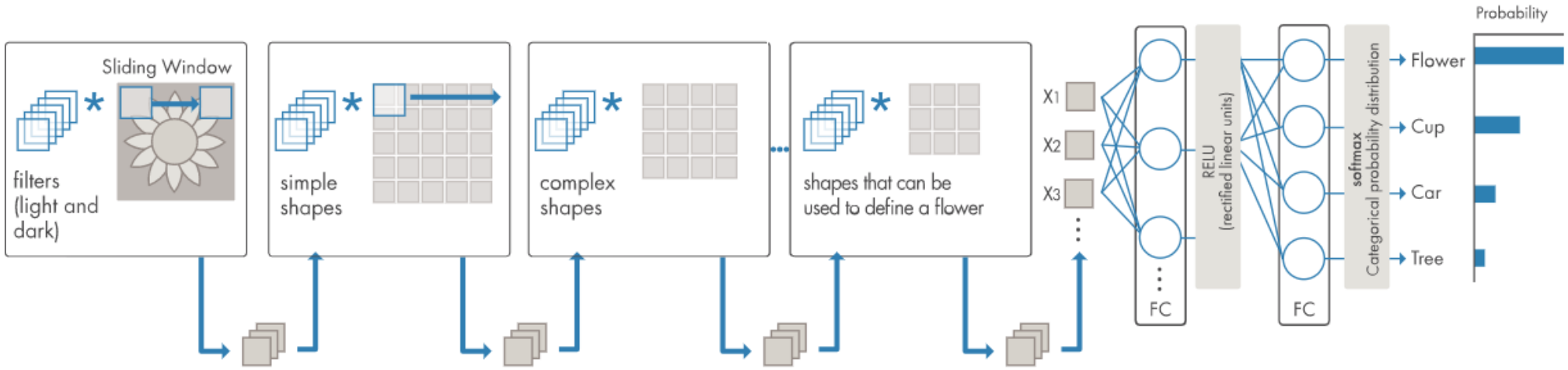
- trained for a specific purpose

https://developersbreach.com/convolution-neural-network-deep-learning/

Neural-based ML methods over images

Convolution NN – principles





https://www.mathworks.com/discovery/deep-learning.html?fbclid=IwAR3nhHwsNjBlgVWDo5fl5opdWdZ-MUKPJbmwC1uD3aCWsnUGwshqyjWa97E

Final Remarks

Conclusion and next steps

- Thank you for attention
- In case of any questions feel free to ask or email me at pmilicka@deloittece.com
- The course was heavy, but hopefully gave you an overview of tools and understanding of basic algorithms and dataset preparation.
- There are still many things I personally do not know as this field is dynamic and new algorithms and packages are being made. So be curious and explore the data science and ML yourself.
- You will learn the most by trying something yourself, even if it would be on dummy datasets.

References:

- https://kaggle.com
- www.datacamp.com
- www.hackerrank.com
- www.towardsdatascience.com
- ...

Deloitte.