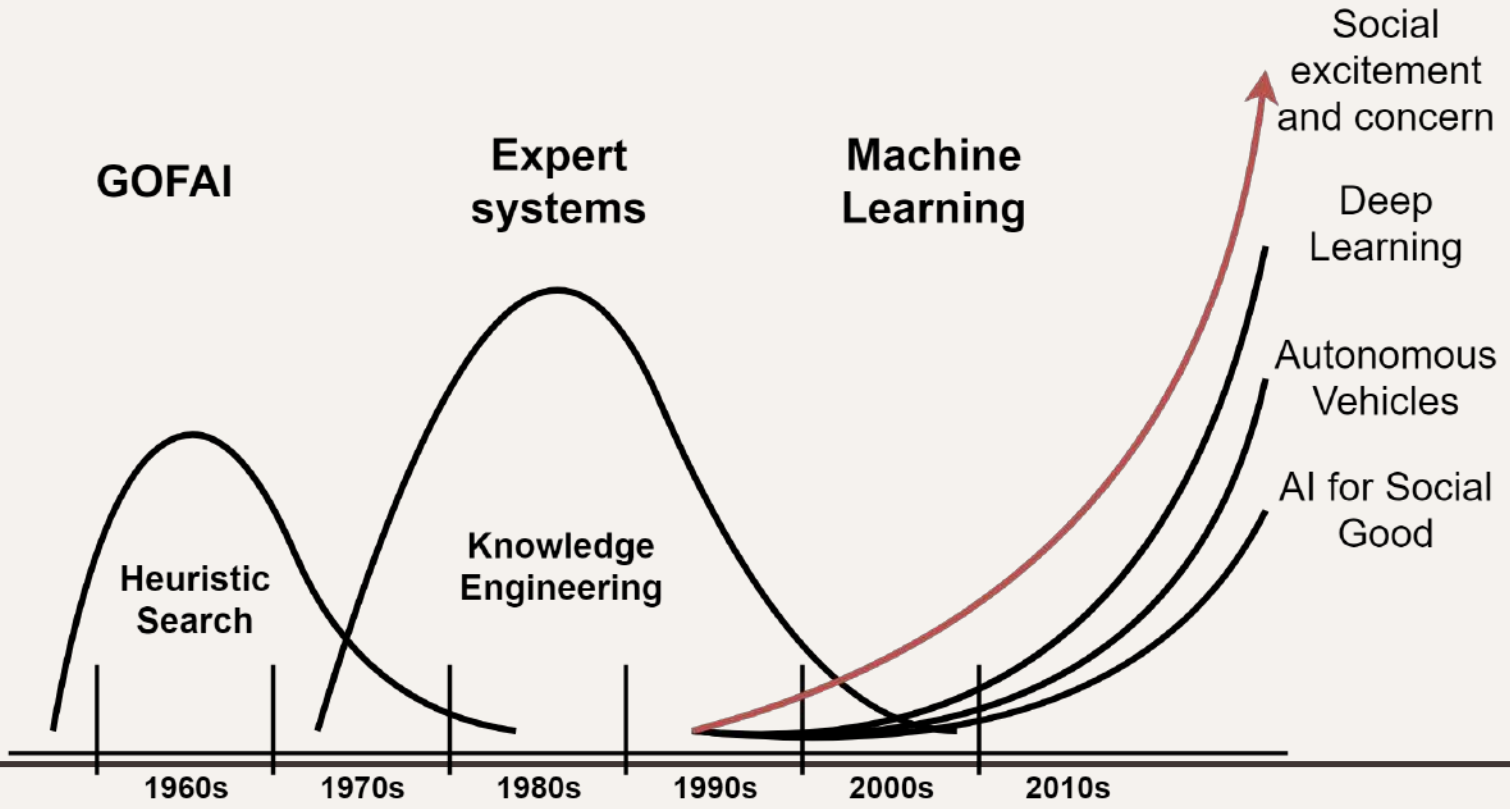
Machine Learning

Refresher

About Me

- Masters and PhD on Artificial Intelligence and Machine Learning
- Researcher at IT Aveiro
- Areas of interest: Artificial Intelligence, Machine Learning, text mining, stream mining, IoT, M2M

AI & ML

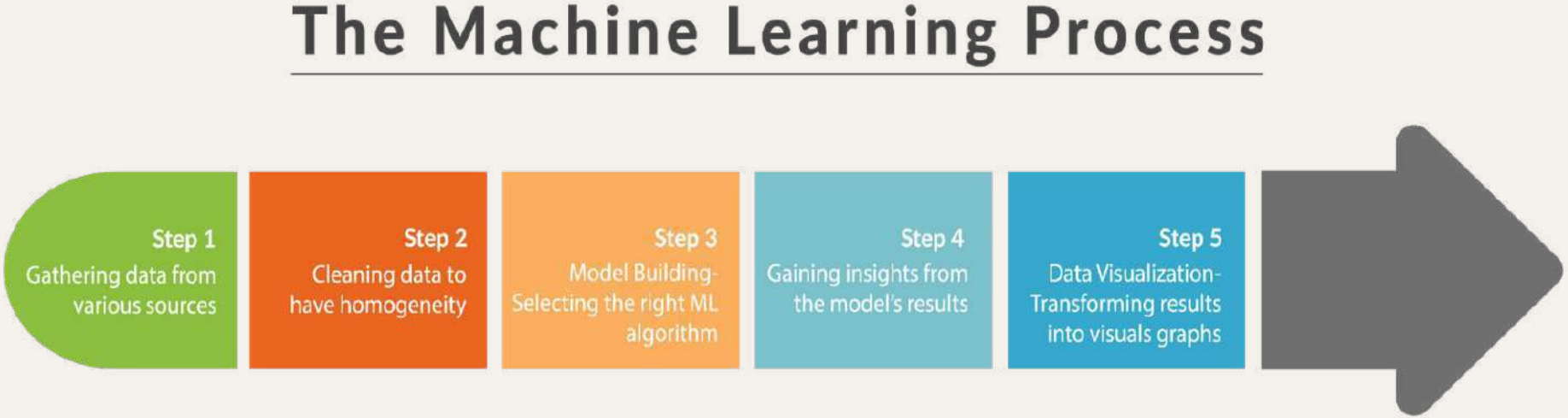

Social excitement and concern
GOFAI
Expert systems
Machine Learning
Deep Learning
Autonomous Vehicles
AI for Social Good
Heuristic Search
Knowledge Engineering
1960s
1970s
1980s
1990s
2000s
2010s

What is ML (Why should i Care)?

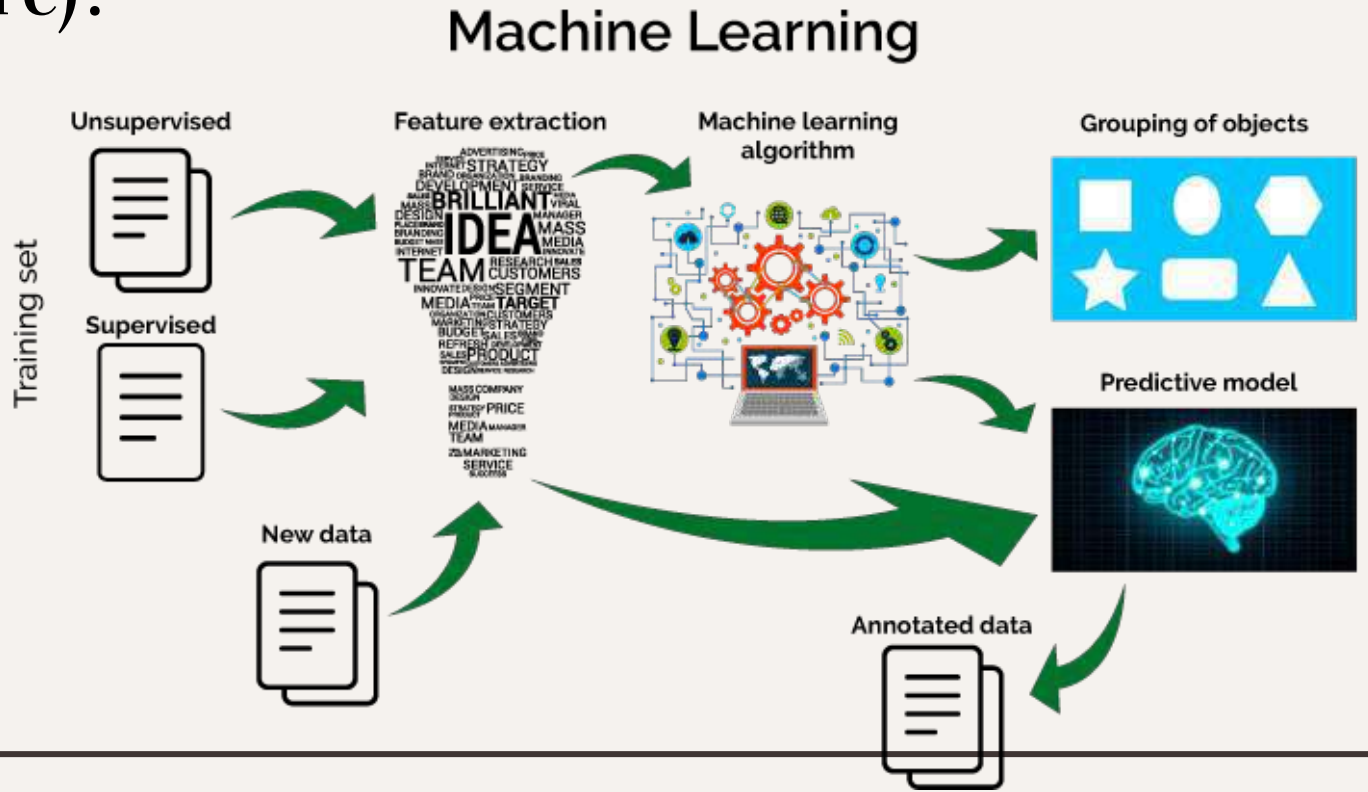
What does machine learning mean?

The term machine learning (abbreviated ML) refers to the capability of a machine to improve its own performance. It does so by using a statistical model to make decisions and incorporating the result of each new trial into that model. In essence, the machine is programmed to learn through trial and error.

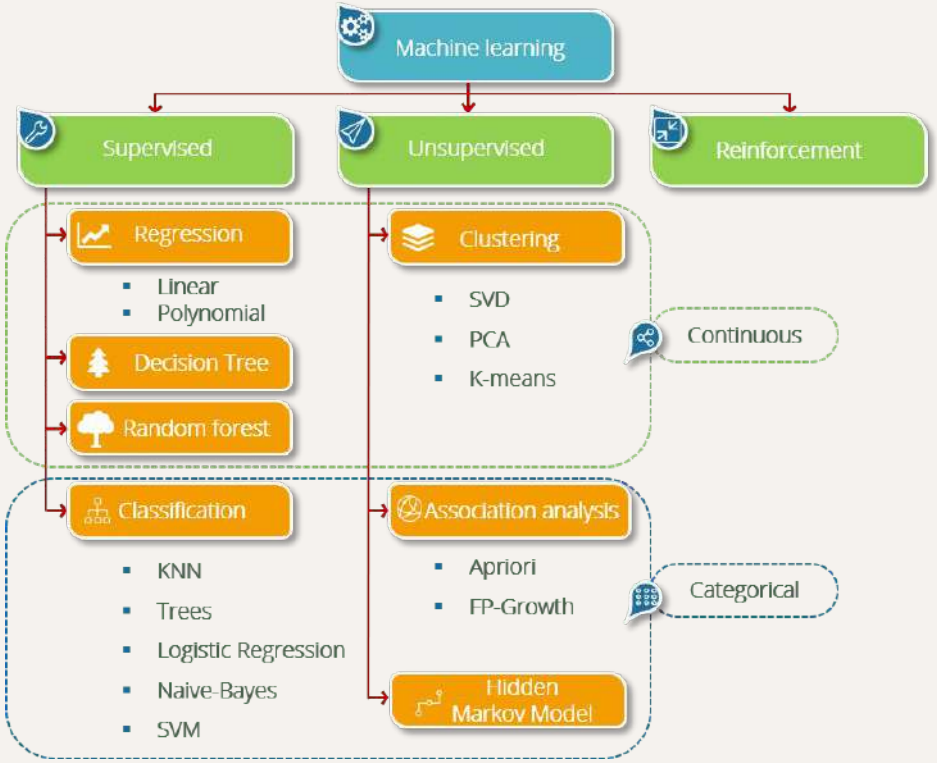
What is ML (Why should i Care)?



What is ML (Why should i Care)?

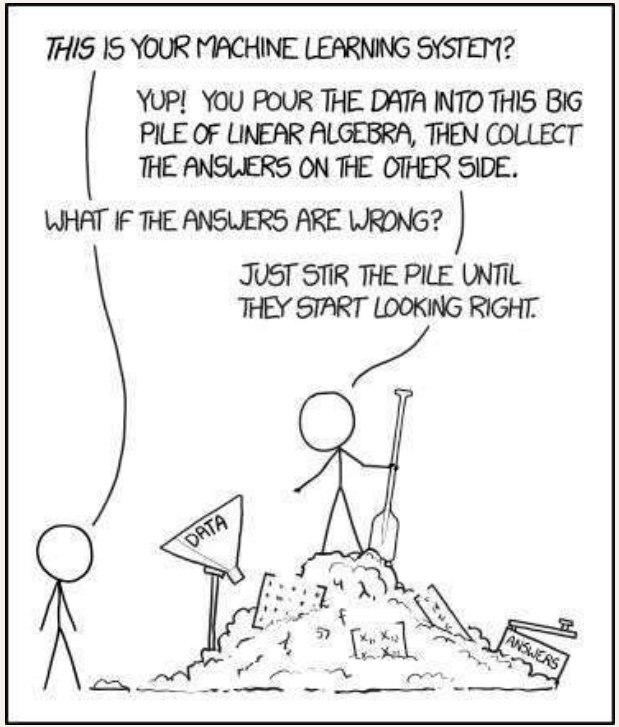


What is ML (Why should i Care)?

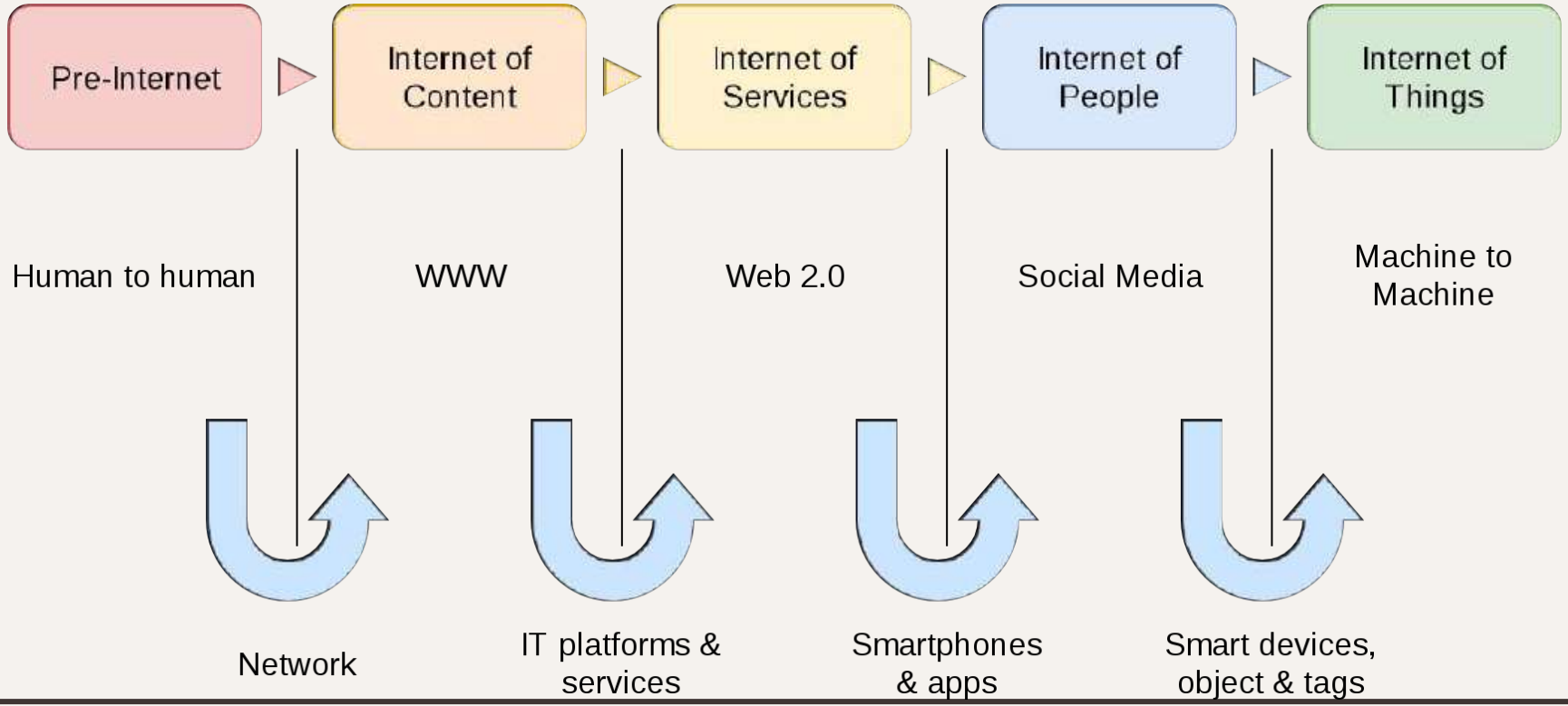


What is ML?

- A body of knowledge related with learning methods for machines (computers)
- Research area
- Opportunities for something useful

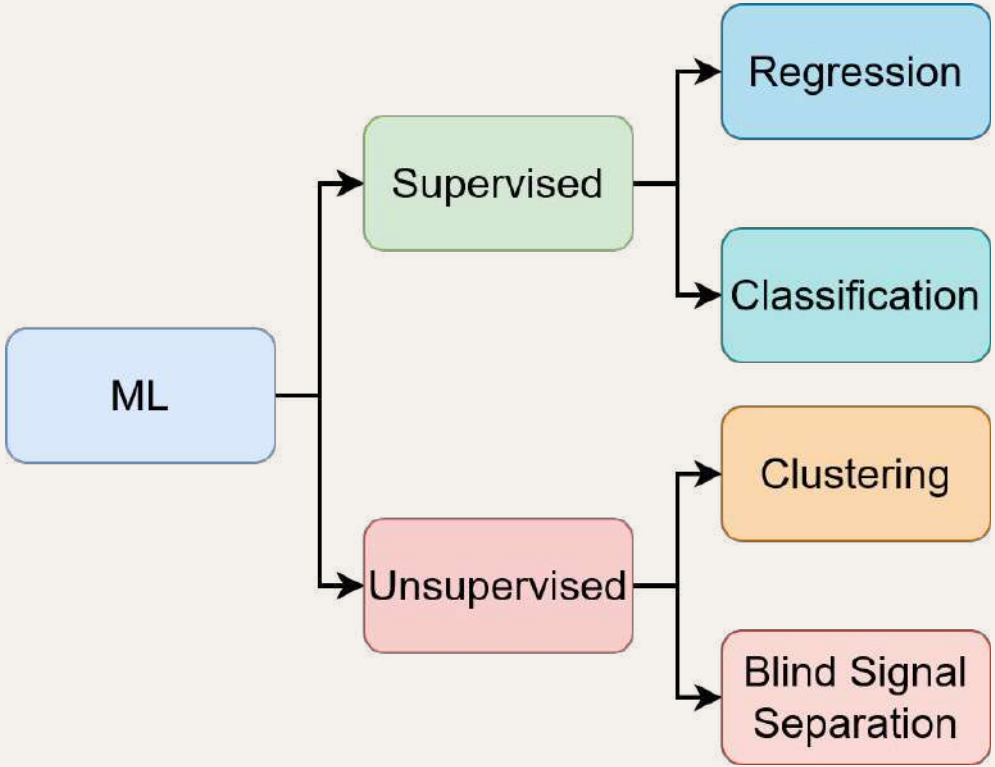


Why Should You Care?

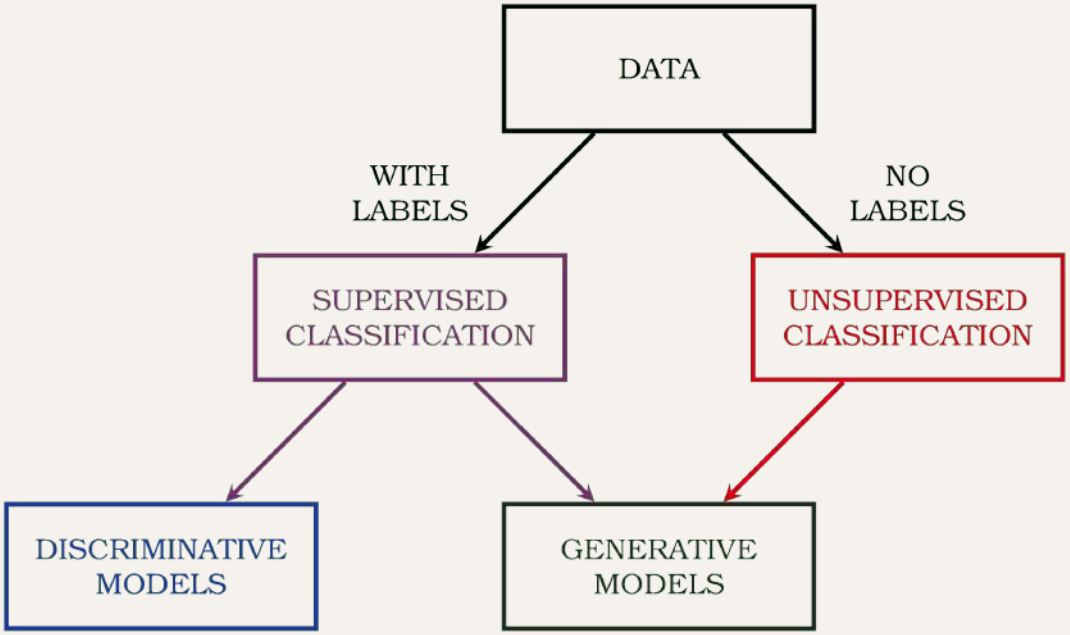


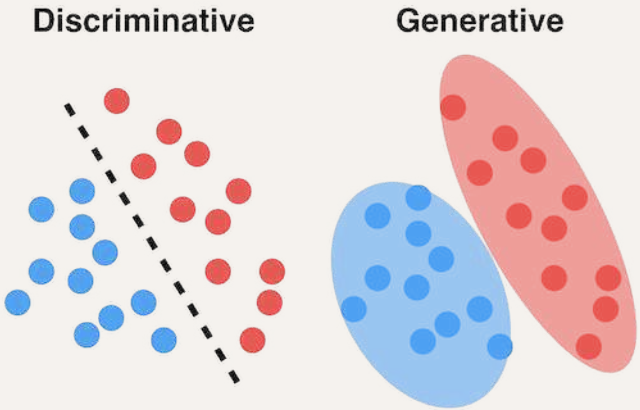
Taxonomy

Taxonomies...


ML
Supervised
Unsupervised
Regression
Classification
Clustering
Blind Signal Separation

Taxonomies...


DATA
WITH LABELS
NO LABELS
SUPERVISED CLASSIFICATION
UNSUPERVISED CLASSIFICATION
DISCRIMINATIVE MODELS
GENERATIVE MODELS


Discriminative
Generative

Taxonomies...

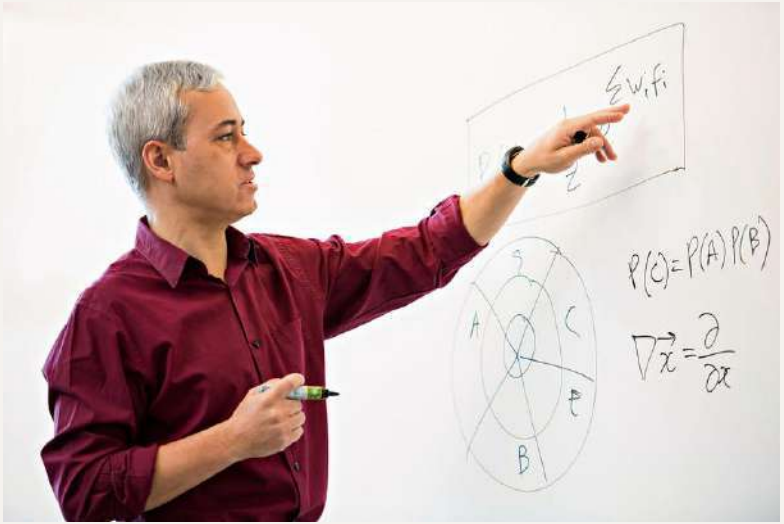
Induction symbolic reasoning

Neural Networks connections modelled on brain's neurons

Evolutionary algorithms learn from random generations (genetic algorithm)

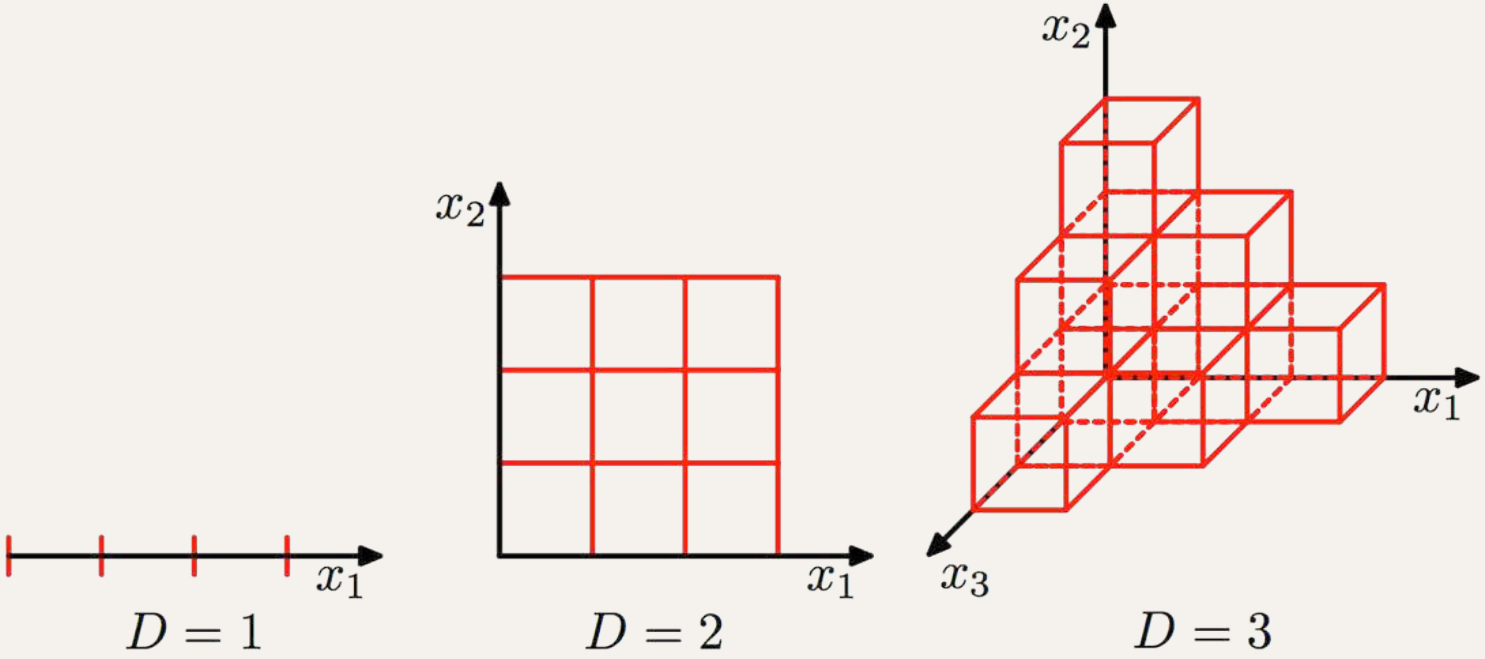
Bayesian inference probabilistic models based on bayes' theorem

Analogy learns by finding similar examples



Limitations

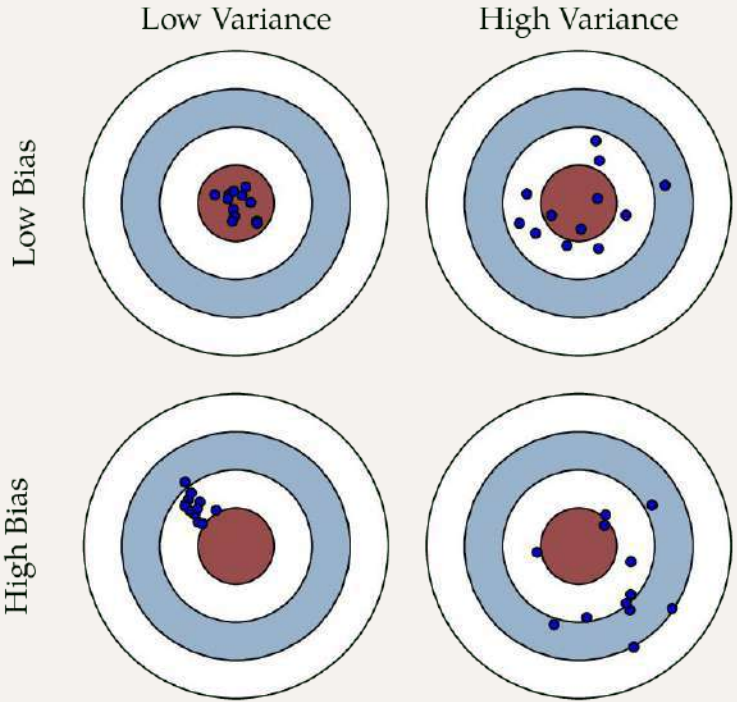
Limitations...



Limitations...

- Our model is a simplification of reality
- Simplification is based on assumptions (model bias)
- Assumptions fail in certain situations

Bias and Variance



Terminology

Terminology

Dataset: organized set of examples, typically composed of features and labels

Feature: single property of an example (input variable)

Label: classification category of an example (output variable)

Example: single instance of a dataset

Aprendizagem Aplicada à Segurança

Mário Antunes

September 22, 2023

University of Aveiro

Table of Contents

SPAM

- The term “spam” is internet slang that refers to unsolicited commercial email (UCE).
- The first reported case of spam occurred in 1898, when the New York Times reported unsolicited messages circulating in association with an old swindle.
- The term “spam” was coined in 1994, based on a now-legendary Monty Python’s Flying Circus sketch, where a crowd of Vikings sings progressively louder choruses of “SPAM! SPAM! SPAM!”

SPAM


1
THE SPANISH PRISONER
The New York Times reports of unsolicited messages circulating in association with an old swindle.
1898
2
POST MAIL
Advertisement based on unsolicited content has been mailed to our doors by Post Mail services for over a century!
Early 1900s
3
ARPANET
The first reported case of email spam is attributed to Digital Equipment Corporation and circulated to 400 ARPANET users.
1978
4
THE EMAIL EPIDEMIC
A growing fraction of emails is spam. Platforms and ISPs start investing in spam filtering techniques.
Mid 1990s
5
SEARCH ENGINES
Web content spam and link farms are common forms of spamdexing, the manipulation of Web search result ranking.
1995
6
SOCIAL NETWORKS
The rise of Facebook, Twitter, and Reddit leads to new opportunities for spammers to reach billions of Social Web users.
2000s
7
FAKE REVIEWS
Giants of e-commerce like Amazon and Alibaba fight the manipulation of product popularity by opinion spam.
Early 2000s
8
PHISHING
Social engineering and disguise may allow attackers to trick victims into revealing sensitive information. Ransomware are used to extort funds from the victims.
Mid 2000s
9
SOCIAL BOTS
Millions of accounts operated by software populate social media to carry out nefarious spam campaigns.
Early 2010s
10
FALSE NEWS
Spam Websites are created to deliberately propagate false news related to politics, public health, and social issues.
Mid 2010s
AI SPAM
Systems based on AI can manipulate reality, producing indistinguishable alternatives. AIs can also be target of manipulation and spam to elicit behaviors of the AI system or of its users.
2018+

SPAM

Dear Sir,
I am prince [illegible] from Nigeria. Your help would be very appreciated.
I want to transfer all of my fortune outside if Nigeria due to a frozen account,
If you could be so kind and transfer small sum of 3 500 USD to my account,
I would be able to unfreeze my account and transfer my money outside of
Nigeria. To repay your kindness, I will send 1 000 000 USD to your account.

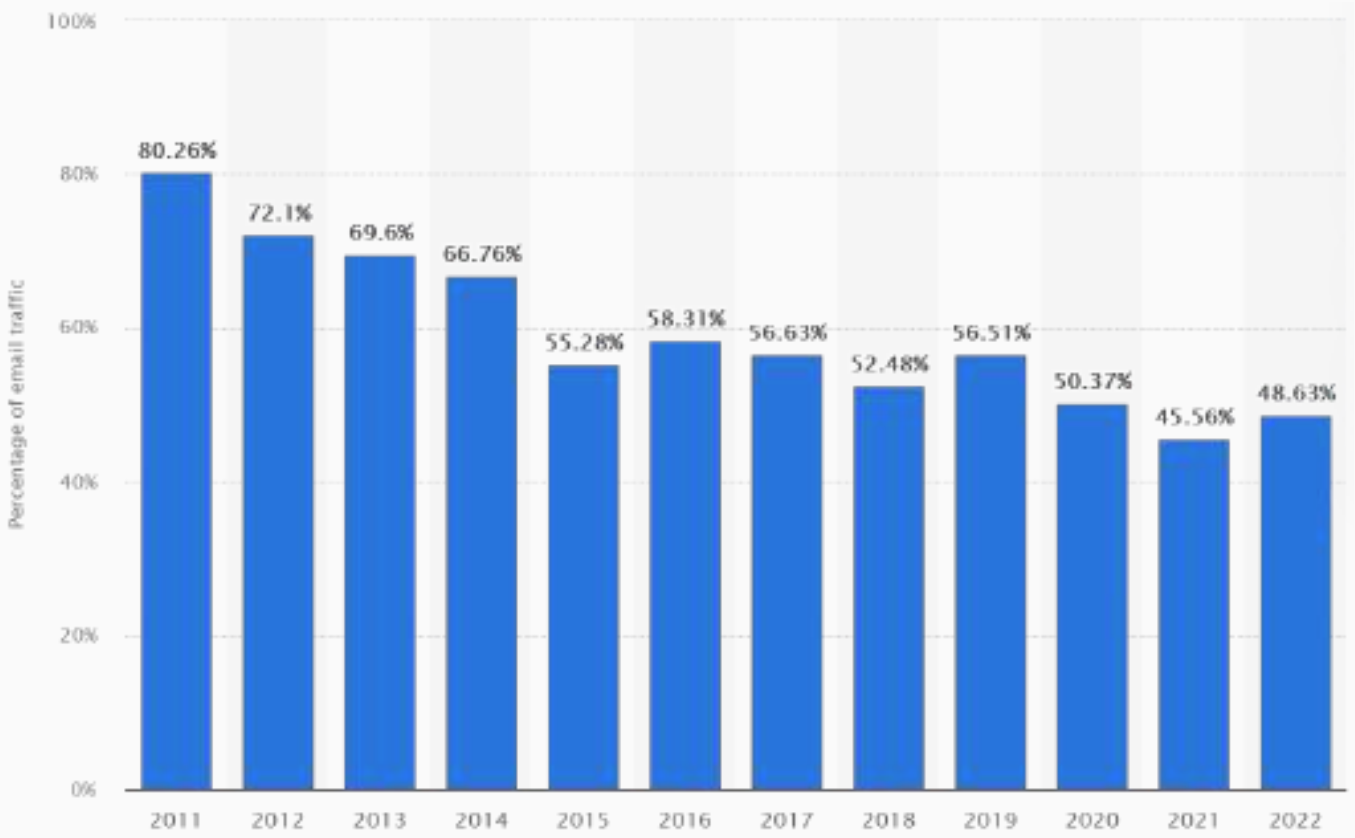
Please contact me to proceed

Prince [illegible]

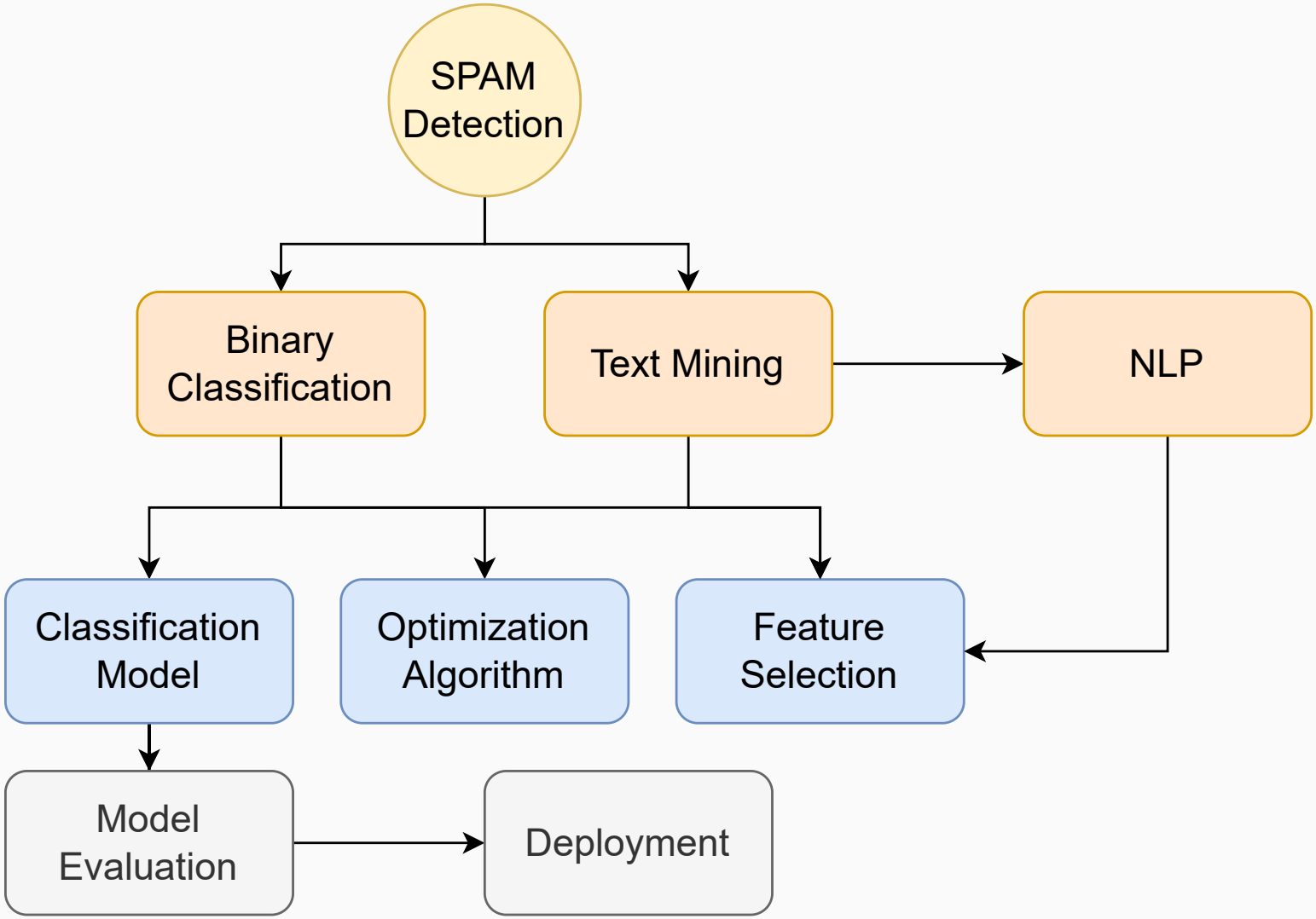
Fight against SPAM

- *Huge* list of `https://en.wikipedia.org/wiki/Anti-spam_techniques`
- From common sense to *Bayesian spam filtering*
- Unfortunately it is a costly battle

Fight against SPAM

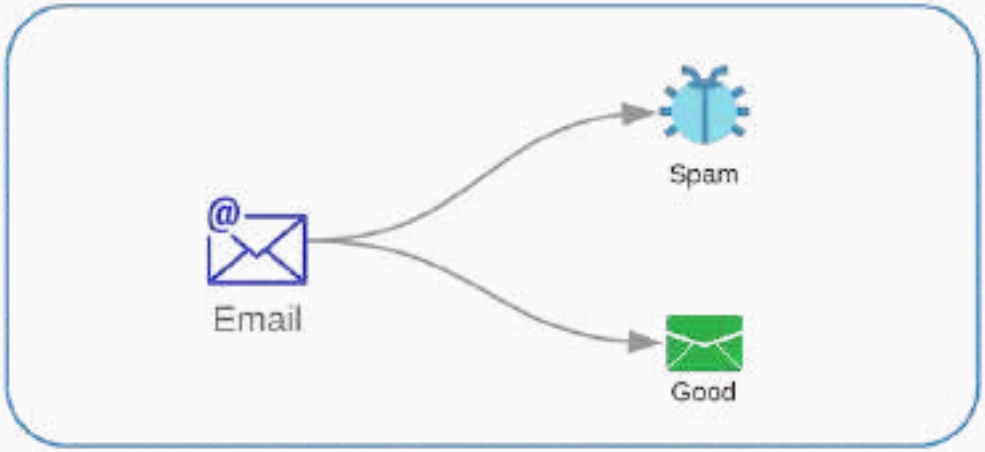


SPAM Detection



Binary Classification

- Binary classification is the task of classifying the elements of a set into two groups (each called class) on the basis of a classification rule.
- For this application one message can either be spam or ham.



Binary classification

Text Mining

- Text mining is the process of deriving high-quality information from text.
- Combines concepts from Machine Learning, Linguistic and statistical analysis.
- In this area we will explore the methods used to rank words/tokens and the BoW model.

Bag of Words (Bow) model

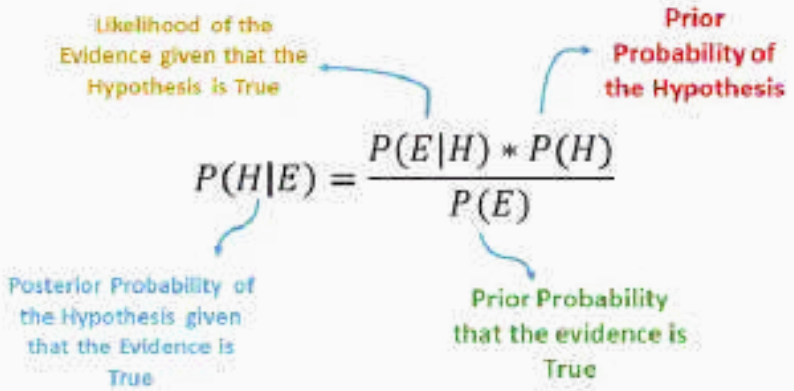
	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0

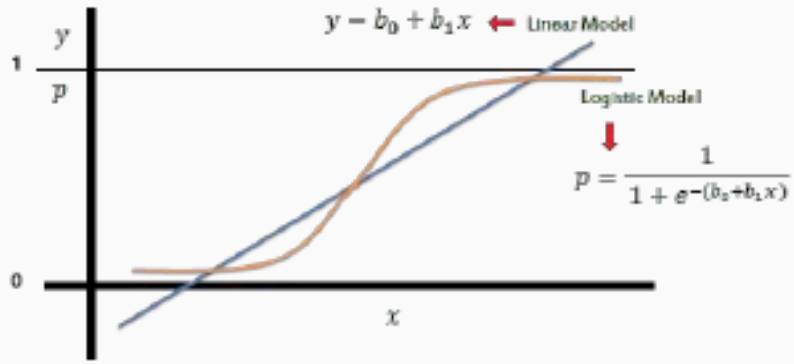
Natural Language Processing (NLP)

- NLP gives the computers the ability to understand text.
- Combines *Sintax* and *Semantic* into the analysis.
- One famous exemples are the Large Language Models (LLMs) that power OpenAI Chat GPT.

Classification Model

- SPAM detection is “considered” a toy example.
- As such, we will explore two of the simples learning models: Naive Bayes and Logistic Regression.



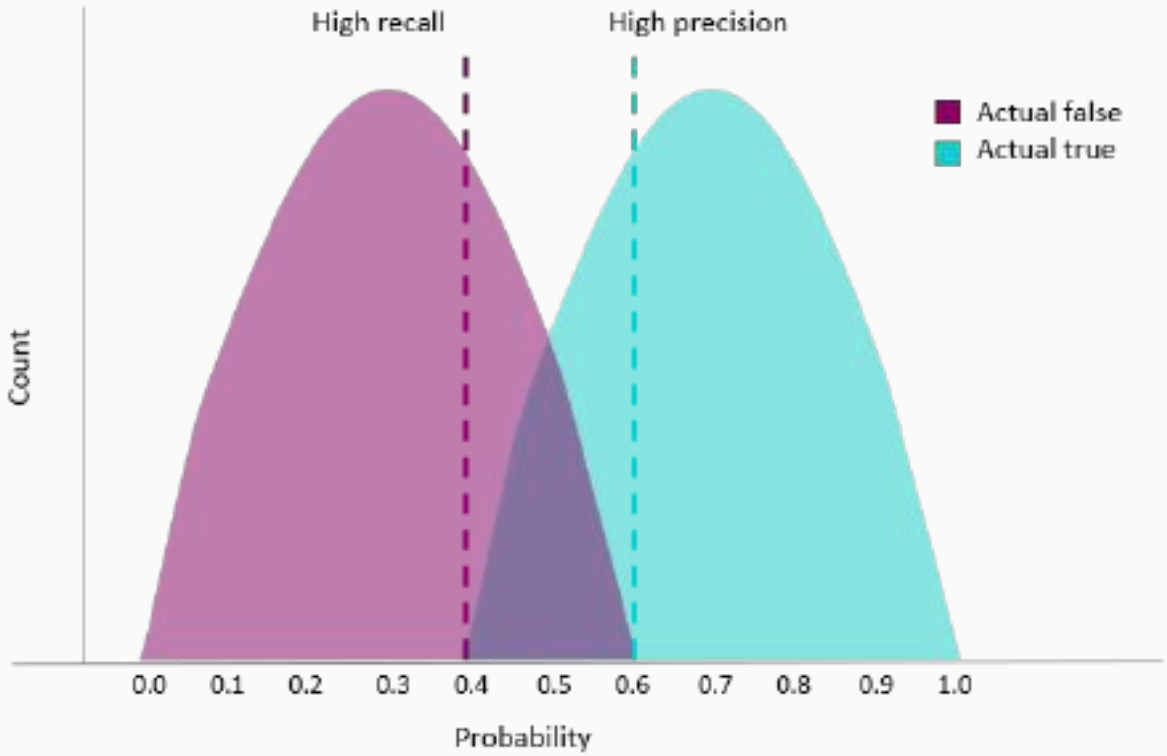


Model Evaluation

- Classification model can be evaluated using a confusing matrix
- The simplest methods to evaluate a model is throuhgh accuracy: $acc = \frac{TP+TN}{TN+TN+FP+FN}$

	Predicted Positive	Predicted Negative	
Actual Positive	**TP** *True Positive*	**FN** *False Negative*	Sensitivity $\frac{TP}{(TP+FN)}$
Actual Negative	**FP** *False Positive*	**TN** *True Negative*	Specificity $\frac{TN}{(TN+FP)}$
	Precision $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Model Evaluation



Aprendizagem Aplicada à Segurança

Mário Antunes

October 14, 2023

Universidade de Aveiro

Context

It is becoming difficult to identify Cybersecurity attacks. These attacks can originate internally due to malicious intent or negligent actions or externally by malware, target attacks, and APT (Advanced Persistent Threats).

But insider threats are more challenging and can cause more damage than external threats because they have already entered the network.

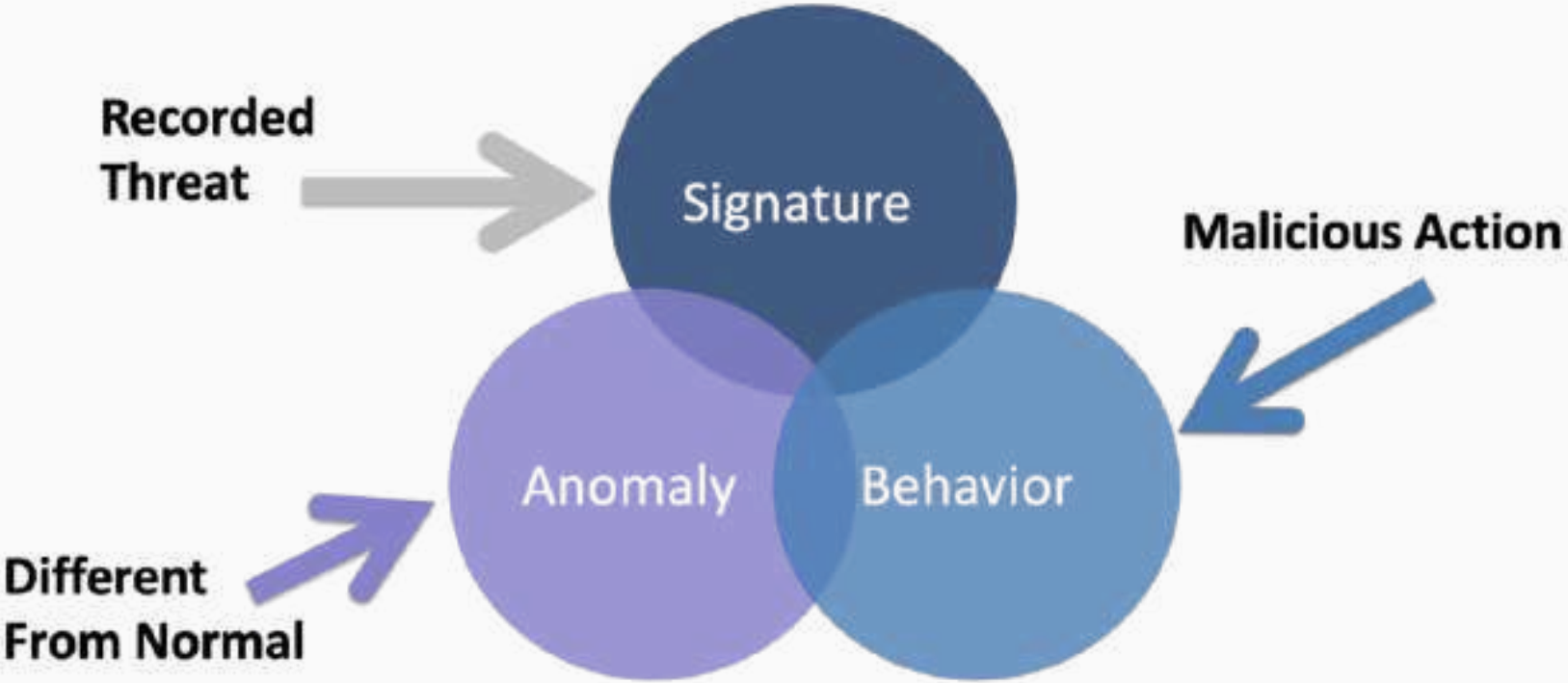
These activities present unknown threats and can steal, destroy or alter the assets.

Context

Earlier firewalls, web gateways, and some other intrusion prevention tools are enough to be secure, but now hackers and cyber attackers can bypass approximately all these defense systems.

Therefore with making these prevention systems strong, it is also equally essential to use detection. So that if hackers get into the network, the system should be able to detect their presence.

Context


Recorded
Threat
Signature
Malicious Action
Anomaly
Behavior
Different
From Normal

Context

Signature detection requires knowing what to look for and comparing hashes or other strings to identify a match. Signature detection is a common feature found within antivirus and IPS/IDS products.

Behavior detection looks for malicious or other known behavior characteristics and alarms the SOC when a match is made. An example is identifying port scanning or a file attempting to encrypt your hard drive, which is an indication of ransomware behavior. Antimalware and sandboxes are examples of tools that heavily leverage behavior detection capabilities.

Anomay detection it takes into consideration hot topics including big data, threat intelligence, and “zero-day” detection.

Anomaly Detection

Anomaly detection, also called outlier detection, is the identification of unexpected events, observations, or items that differ significantly from the norm:

- Anomalies in data occur only very rarely
- The features of data anomalies are significantly different from those of normal instances

What is an anomaly?

Generally speaking, an **anomaly** is something that differs from a norm: a deviation, an exception. In software engineering, by anomaly we understand a rare occurrence or event that doesn't fit into the pattern, and, therefore, seems suspicious. Some examples are:

- sudden burst or decrease in activity;
- error in the text logs;
- sudden rapid drop or increase in temperature.

What is an anomaly?

Common reasons for outliers are:

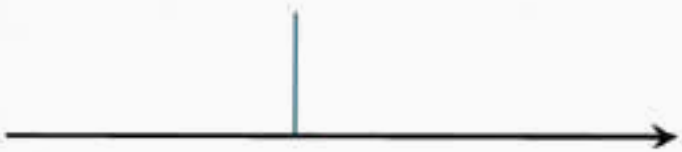
- data preprocessing errors;
- noise;
- fraud;
- attacks.

Types of Anomalies

Anomalies can be broadly categorized as:

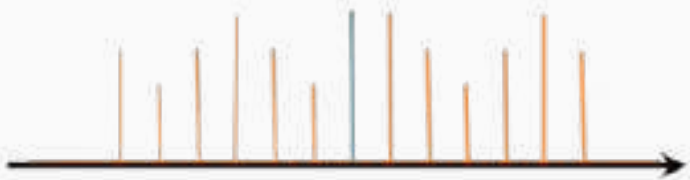
- Point anomalies: A single instance of data is anomalous if it's too far off from the rest.
- Contextual anomalies: The abnormality is context specific. This type of anomaly is common in time-series data.
- Collective anomalies: A set of data instances collectively helps in detecting anomalies.

Types of Anomalies



(a) Point Anomaly

(b) Collective Anomaly



(c) Contextual Anomaly

Anomaly Detection - Example #1

Network anomalies: Anomalies in network behavior deviate from what is normal, standard, or expected. To detect network anomalies, network owners must have a concept of expected or normal behavior. Detection of anomalies in network behavior demands the continuous monitoring of a network for unexpected trends or events.

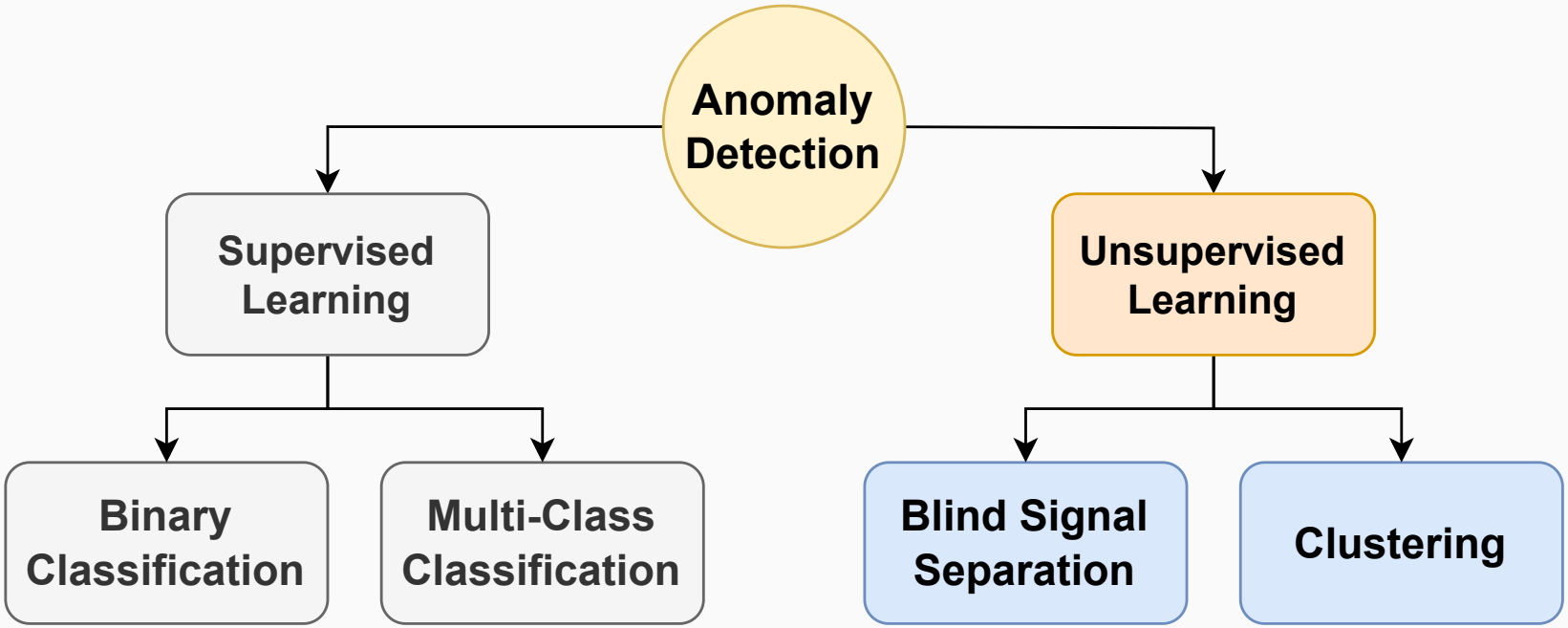
Anomaly Detection - Example #2

Application performance anomalies: These are simply anomalies detected by end-to-end application performance monitoring. These systems observe application function, collecting data on all problems, including supporting infrastructure and app dependencies. When anomalies are detected, rate limiting is triggered and admins are notified about the source of the issue with the problematic data.

Anomaly Detection - Example #3

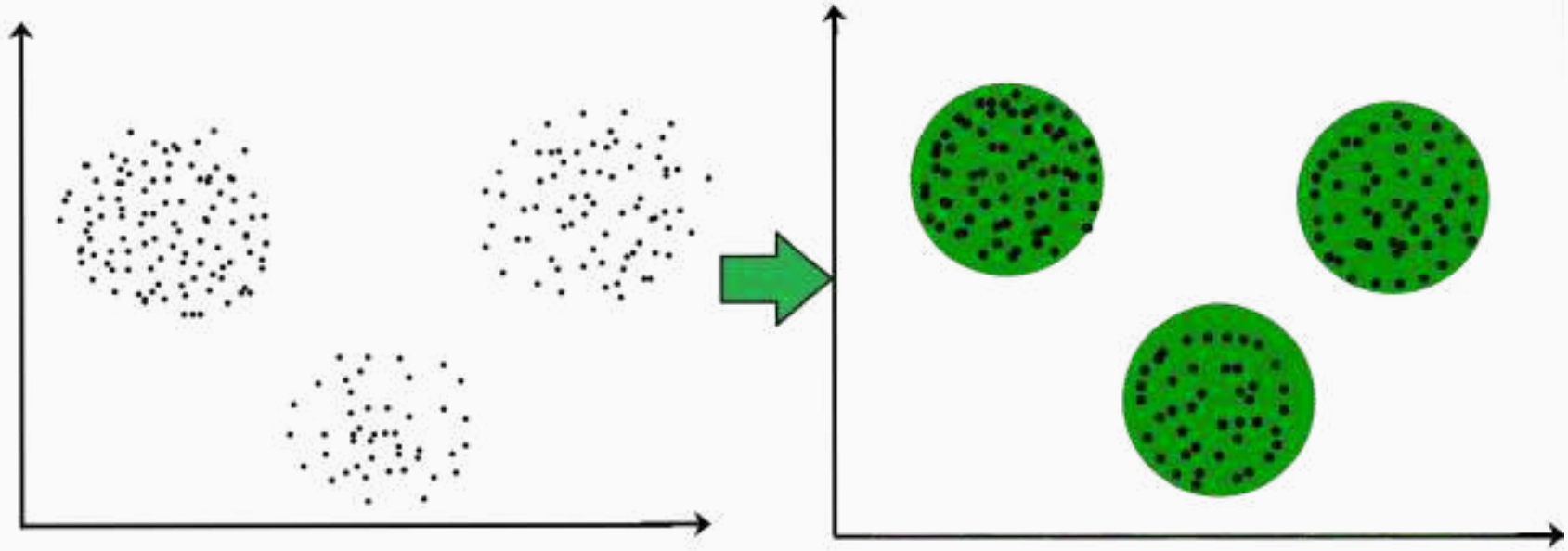
Web application security anomalies: These include any other anomalous or suspicious web application behavior that might impact security such as CSS attacks or DDOS attacks.

Anomaly Detection



Clustering

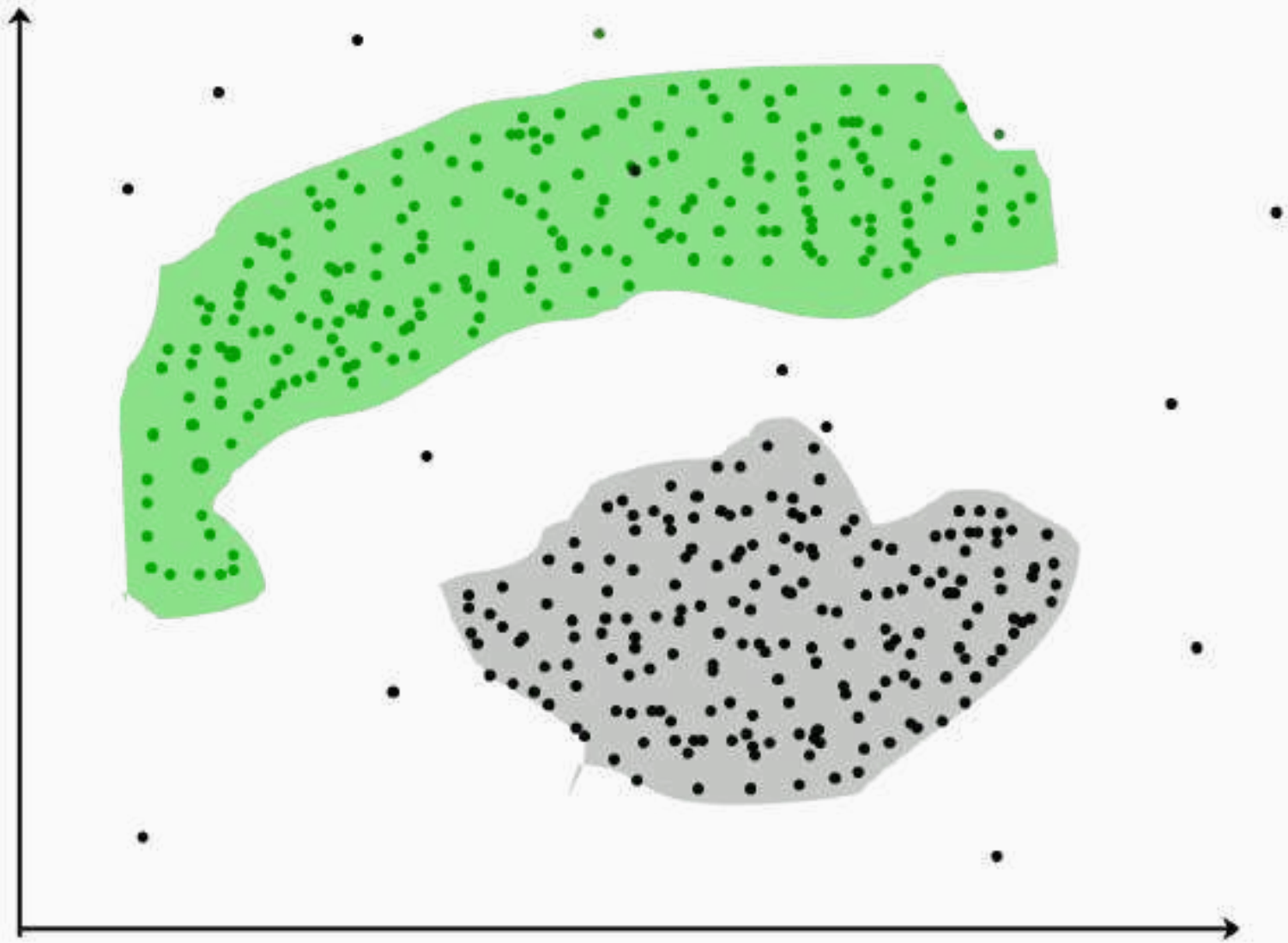
Type of **unsupervised learning method**. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.



Clustering Methods

- **Density-Based Methods:** These methods consider the clusters as the dense region having some similarities and differences from the lower dense region of the space. These methods have good accuracy and the ability to merge two clusters.
- **Hierarchical Based Methods:** The clusters formed in this method form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one.
- **Partitioning Methods:** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter.

Clustering: Anomaly Detection



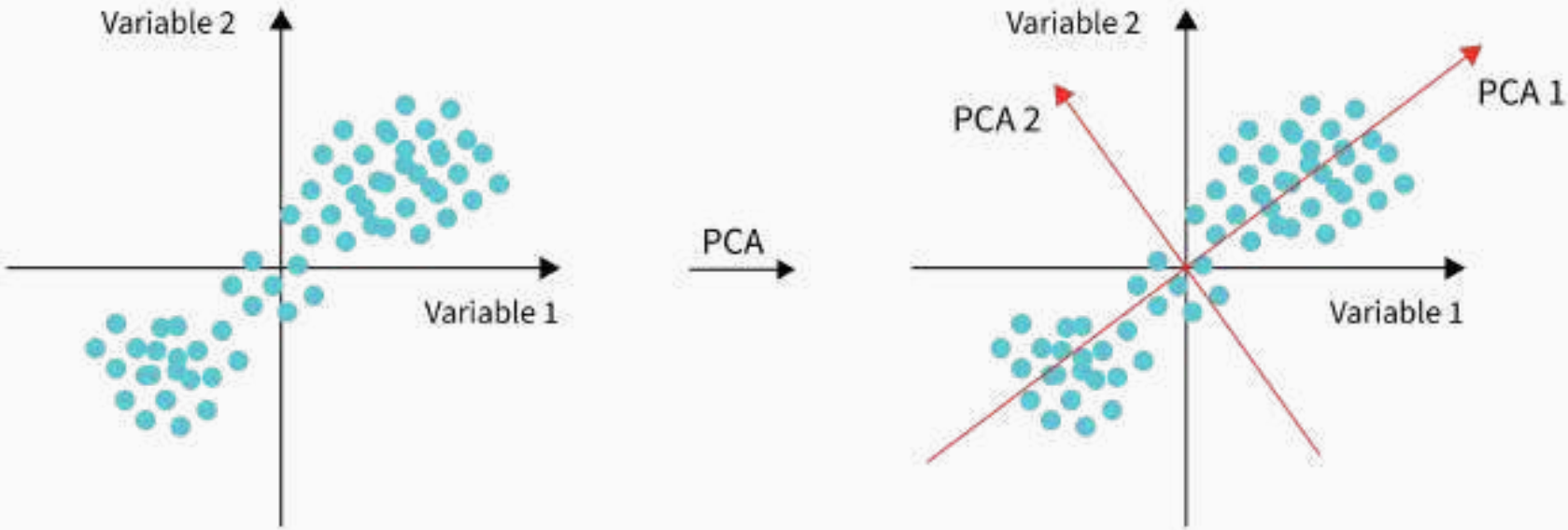
Blind Source Separation

Blind Source Separation (BSS) refers to a problem where both the sources and the mixing methodology are unknown, only mixture signals are available for further separation process.

In several situations it is desirable to recover all individual sources from the mixed signal, or at least to segregate a particular source.

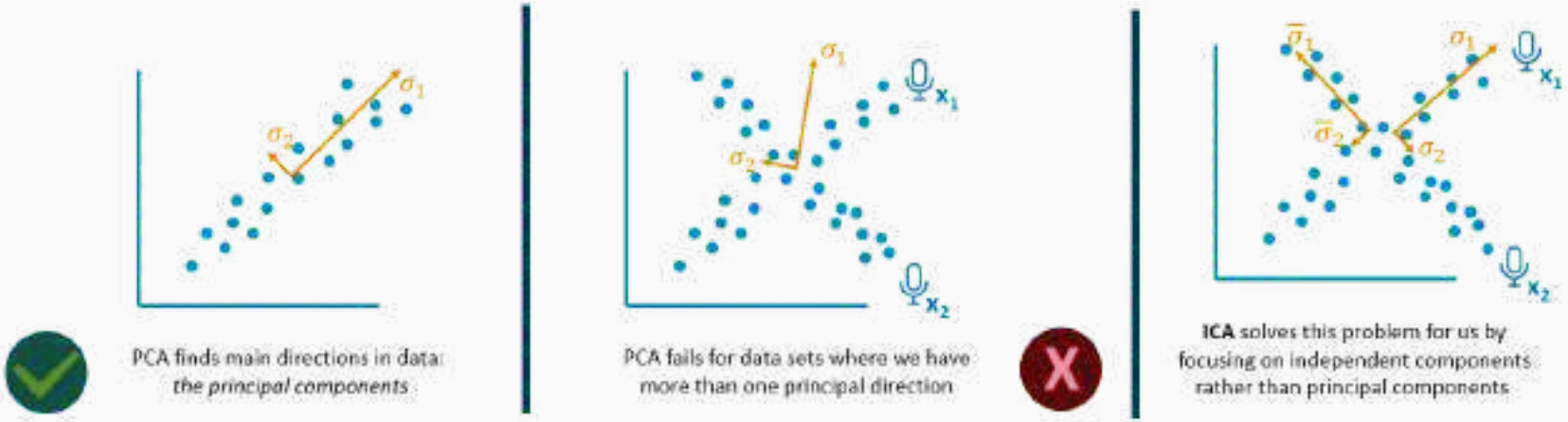
Blind Source Separation: PCA

** Principal component analysis**, or PCA, is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of “summary indices” that can be more easily visualized and analyzed.



Blind Source Separation: ICA

Independent Component Analysis (ICA) is a powerful technique in the field of data analysis that allows you to separate and identify the underlying independent sources in a multivariate data set.

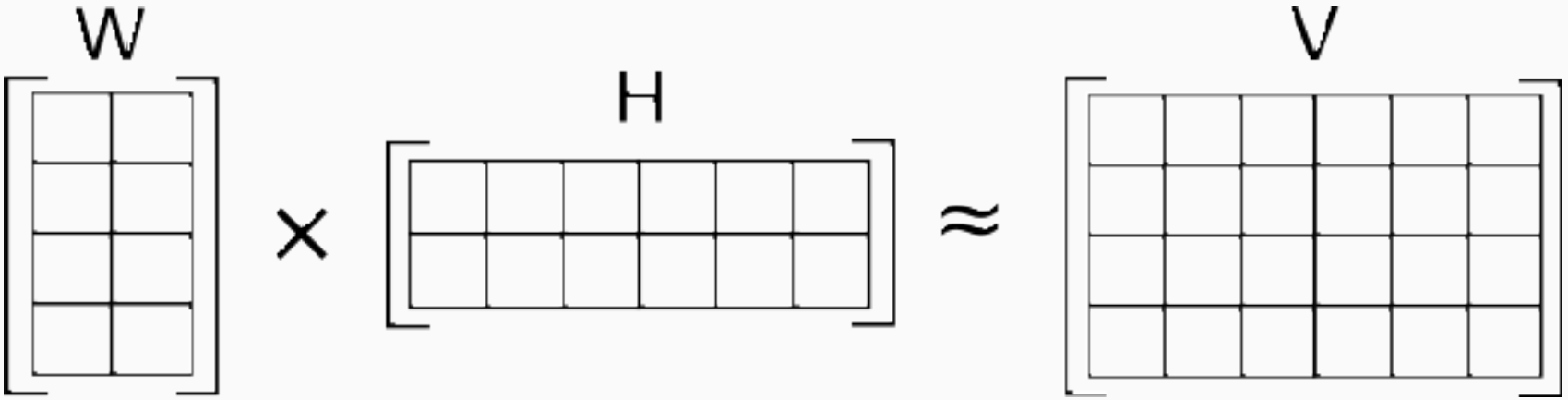


Blind Source Separation: NNMF

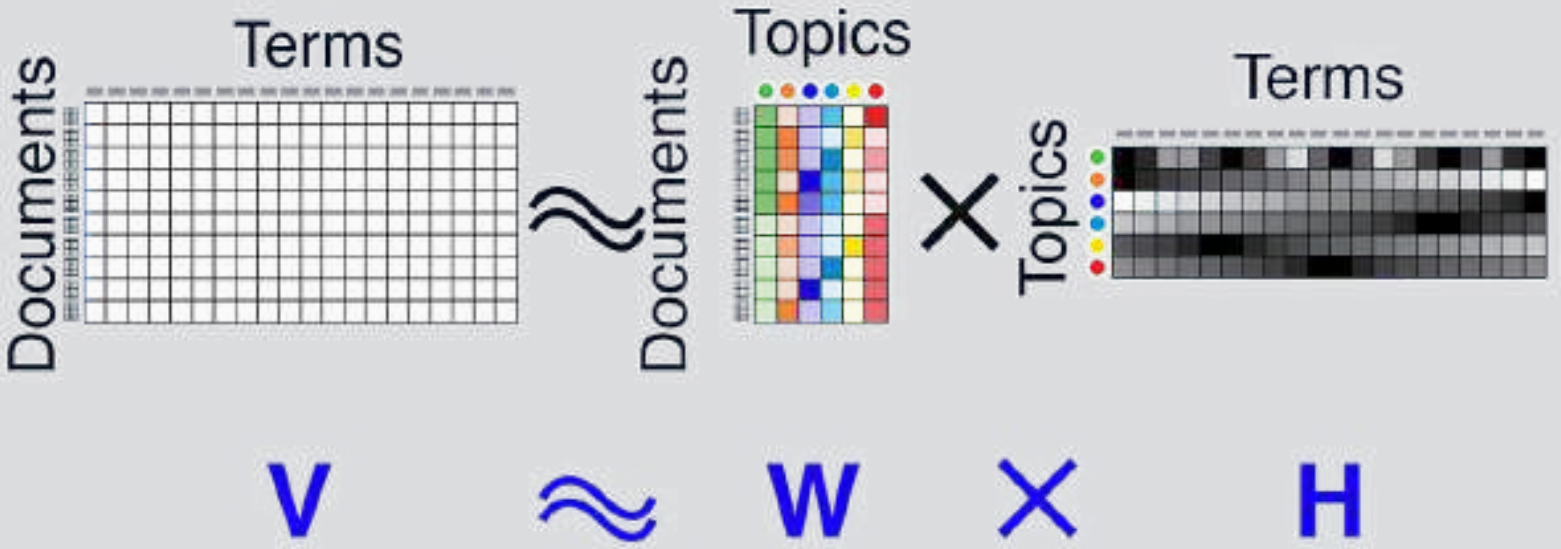
Non-negative matrix factorization (NNMF) is a group of algorithms in multivariate analysis and linear algebra where a matrix V is factorized into two matrices W and H, with the property that all three matrices have no negative elements.

This non-negativity makes the resulting matrices easier to inspect. Also, in applications such as processing of audio spectrograms or muscular activity, non-negativity is inherent to the data being considered.

Since the problem is not exactly solvable in general, it is commonly approximated numerically.



Non-Negative Matrix Factorization Diagram - Example



Visible Variables
Input
Document x Term Matrix
n x m
10 x 20

Weights
Feature Set
Document x Topic Matrix
n x p
10 x 6

Hidden Variables
Coefficients
Topic x Term Matrix
p x m
6 x 20

Aprendizagem Aplicada à Segurança

Mário Antunes

November 10, 2023

Universidade de Aveiro

Context

Context

Malware, or malicious software, is any program or file that is intentionally harmful to a computer, network or server.

Malware can infect networks and devices and is designed to harm those devices, networks and/or their users in some way.

Depending on the type of malware and its goal, this harm may present itself differently to the user or endpoint. In some cases, the effect malware has is relatively mild and benign, and in others, it can be disastrous.

Creeper virus (1971)

Computer pioneer John von Neumann's posthumous work Theory of Self-Reproducing Automata, was published in 1966. Five years later, the first known computer virus, called Creeper, was a written by Bob Thomas. Written in PDP-10 assembly language, Creeper could reproduce itself and move from computer to computer across the nascent ARPANET.

Creeper did no harm to the systems it infected - Thomas developed it as a proof of concept, and its only effect was that it caused connected teletype machines to print a message that said "I'M THE CREEPER: CATCH ME IF YOU CAN."

ILOVEYOU worm (2000)

Onel de Guzman crafted his creation with straightforward criminal intent: he couldn't afford dialup service, so he built a worm that would steal other people's passwords so he could piggyback off of their accounts.

But the malware so cleverly took advantage of a number of flaws in Windows 95 that it spread like wildfire, and soon millions of infected computers were sending out copies of the worm and beaming passwords back to a Filipino email address.

de Guzman was never charged with a crime, because nothing he did was illegal in the Philippines at the time, but he expressed regret in an interview 20 years later, saying he never intended the malware to spread as far as it did.

CryptoLocker ransomware (2013)

CryptoLocker became famous for its rapid spread and its powerful asymmetric encryption that was (at the time) uniquely difficult to break.

It also became famous due to something unusual in the malware world: a happy ending. In 2014, the U.S. DoJ and peer agencies overseas managed to take control of the Gameover Zeus botnet, and restore the files of CryptoLocker victims free of charge.
Unfortunately, CryptoLocker spread via good old-fashioned phishing as well, and variants are still around.

Mirai botnet (2016)

Internet of Things (IoT) devices are omnipresent, ignored, and often go unpatched for years. The Mirai botnet was actually similar to some of the early malware we discussed because it exploited a previously unknown vulnerability and wreaked far more havoc than its creator intended.

In this case, the malware found and took over IoT gadgets (mostly CCTV cameras) that hadn't had their default passwords changed. Paras Jha, the college student who created the Mirai malware, intended to use the botnets he created for DoS attacks that would help settle scores in the obscure world of Minecraft server hosting, but instead he unleashed an attack that focused on a major DNS provider and cut off much of the U.S. east coast from the internet for the better part of a day.

Clop ransomware (2019-Present)

Clop (sometimes written Cl0p) is another ransomware variant that emerged on the scene in 2019 and has grown increasingly prevalent since. In addition to preventing victims from accessing their data, Clop allows the attacker to withdraw that data as well.

What makes Clop so interesting and dangerous, however, is not how it's deployed, but by whom. It's at the forefront of a trend called Ransomware-as-a-Service, in which a professionalized group of hackers does all the work for whoever will pay them enough (or share in a percentage of the ransomware riches they extract from victims).

How To Recognize Malware

How To Recognize Malware

1. **Signature-based detection:** Signature-based detection uses known digital indicators of malware to identify suspicious behavior. Lists of indicators of compromise (IOCs) can be used to identify a breach. While IOCs can be effective in identifying malicious activity, they are reactive in nature.
2. **Static file analysis:** Examining a file's code, without running it, to identify signs of malicious intent. File names, hashes, strings such as IP addresses, and file header data can all be evaluated to determine whether a file is malicious.

How To Recognize Malware #2

3. **Dynamic malware analysis:** Dynamic malware analysis executes suspected malicious code in a safe environment called a sandbox. This closed system enables security professionals to watch and study the malware in action without the risk of letting it infect their system or escape into the enterprise network.
4. **Dynamic monitoring of mass file operations:** Observing mass file operations such as rename or delete commands to identify signs of tampering or corruption. Dynamic monitoring often uses a file integrity monitoring tool to track and analyze the integrity of file systems through both reactive forensic auditing and proactive rules-based monitoring.

How To Recognize Malware #3

5. **File extensions blocklist:** File extensions are letters occurring after a period in a file name, indicating the format of the file. This classification can be used by criminals to package malware for delivery. As a result, a common security method is to list known malicious file extension types in a "blocklist" to prevent unsuspecting users from downloading or using the dangerous file.

6. **Application allowlist:** The opposite of a blocklist/blocklisting, where an organization authorizes a system to use applications on an approved list. Allowlisting can be very effective in preventing nefarious applications through rigid parameters. However, it can be difficult to manage and reduce an organization's operational speed and flexibility.

How To Recognize Malware #4

7. **Malware honeypot:** A malware honeypot mimics a software application or an application programming interface (API) to draw out malware attacks in a controlled, non-threatening environment. Similarly, a honeypot file is a decoy file to draw and detect attackers. In doing so, security teams can analyze the attack techniques and develop or enhance antimalware solutions to address these specific vulnerabilities, threats or actors.

8. **Cyclic redundancy check (CRC):** A calculation on a collection of data, such as a file, to confirm its integrity. One of the most common checksums used is a CRC, which involves analysis of both value and position of a group of data. Checksumming can be effective for identifying corruption in data but is not foolproof for determining tampering.

How To Recognize Malware #5

9. **File entropy:** As threat intelligence and cybersecurity evolves, adversaries increasingly create dynamic malware executables to avoid detection. This results in modified files that have high entropy levels. As a result, a file's data change measured through entropy can identify potential malware.
10. **Machine learning analysis:** Machine learning (ML) is a subset of artificial intelligence (AI), and refers to the process of teaching algorithms to learn patterns from existing data to predict answers on new data. This technology can analyze file behavior, identify patterns and use these insights to improve detection of novel and unidentified malware.

Malware Detection

Malware Detection

An efficient, robust and scalable malware recognition module is the key component of every cybersecurity product. Malware recognition modules decide if an object is a threat, based on the data they have collected on it. This data may be collected at different phases:

- Pre-execution phase data is anything you can tell about a file without executing it. This may include executable file format descriptions, code descriptions, binary data statistics, text strings and information extracted via code emulation and other similar data.
- Post-execution phase data conveys information about behavior or events caused by process activity in a system.

Malware Detection #2

In the early part of the cyber era, the number of malware threats was relatively low, and simple manually created pre-execution rules were often enough to detect threats.

The rapid rise of the Internet and the ensuing growth in malware meant that manually created detection rules were no longer practical - and new, advanced protection technologies were needed.

Today, machine learning boosts malware detection using various kinds of data on host, network and cloud-based anti-malware components.

Machine learning Approaches

Unsupervised learning

The goal is to discover the structure of the data or the law of data generation.

Large unlabeled datasets are available to cybersecurity vendors and the cost of their manual labeling by experts is high – this makes unsupervised learning valuable for threat detection.

Machine learning Approaches #2

Supervised learning

Supervised learning is a setting that is used when both the data and the right answers for each object are available. The goal is to fit the model that will produce the right answers for new objects.

Supervised learning consists of two stages:

- **Training** a model and fitting a model to available training data.
- **Applying** the trained model to new samples and obtaining predictions.

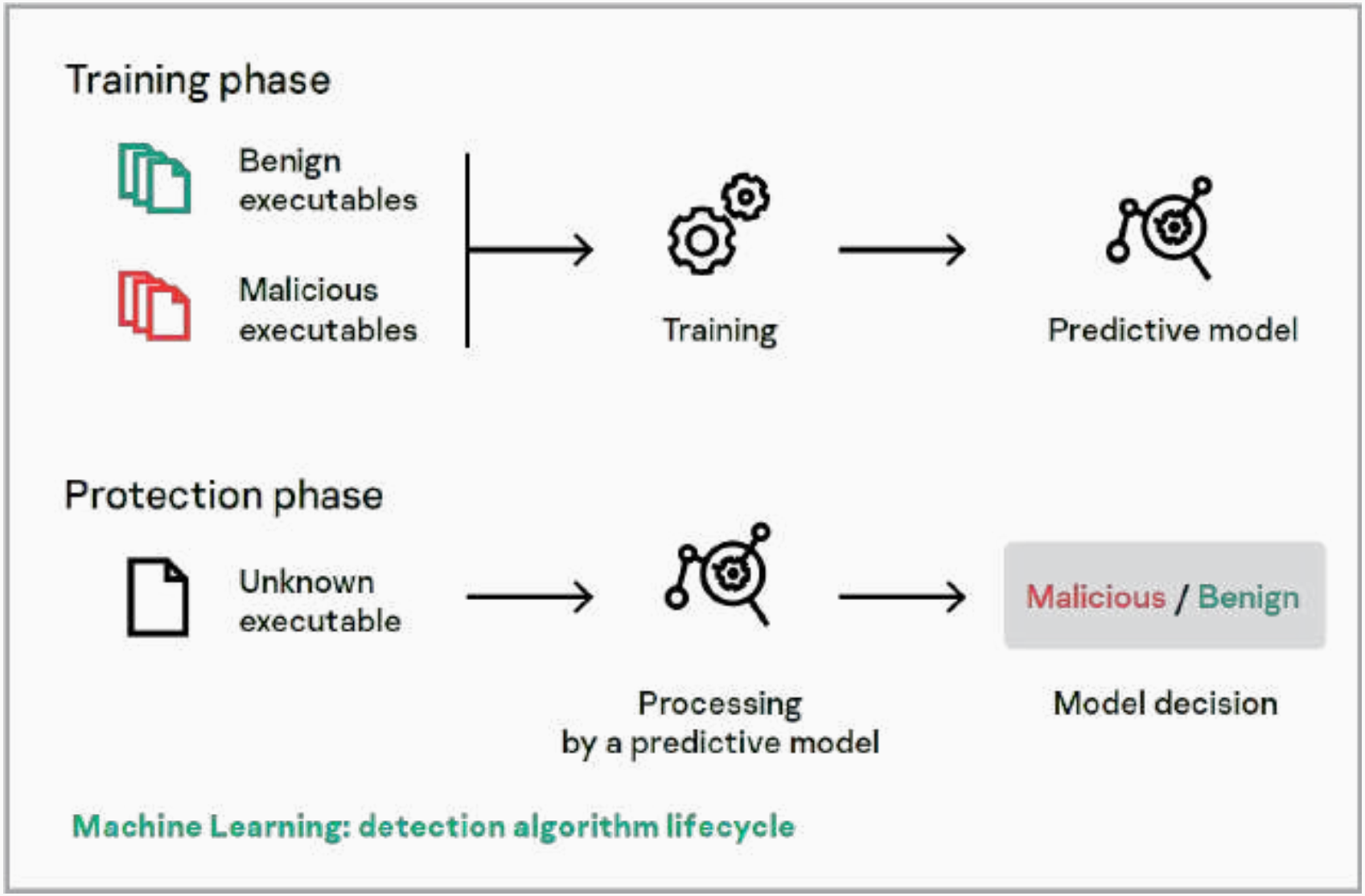
Machine learning Approaches #3

This training information is utilized during the training phase, when we search for the best model that will produce the correct label Y for previously unseen objects given the feature set X.

In the case of malware detection, X could be some features of file content or behavior, for instance, file statistics and a list of used API functions. Labels Y could be malware or benign, or even a more precise classification, such as a virus, Trojan-Downloader or adware.

After we have trained a model and verified its quality, we are ready for the next phase – applying the model to new objects. In this phase, the type of the model and its parameters do not change. The model only produces predictions.

Machine learning Approaches #4



Requirements

Deep learning

Deep learning is a special machine learning approach that facilitates the extraction of features of a high level of abstraction from low-level data. Deep learning has proven successful in computer vision, speech recognition, natural language processing and other tasks.

A deep learning model can learn complex feature hierarchies and incorporate diverse steps of malware detection pipeline into one solid model that can be trained end-to-end, so that all of the components of the model are learned simultaneously.

Requirements #2

Large representative datasets are required

It is important to emphasize the **data-driven** nature of this approach. A created model depends heavily on the data it has seen during the training phase to determine which features are statistically relevant for predicting the correct label.

We must train our models on a data set that correctly represents the conditions where the model will be working in the real world. This makes the task of collecting a **representative dataset** crucial for machine learning to be successful.

Requirements #3

The trained model has to be interpretable (XAI)

Most of the model families used currently, like deep neural networks, are called **black box models.** Black box models are given the input X, and they will produce Y through a complex sequence of operations that can hardly be interpreted by a human.

For example, when a false alarm occurs, and we want to understand why it happened, we ask whether it was a problem with a training set or the model itself. The **interpretability** of a model determines how easy it will be for us to manage it, assess its quality and correct its operation.

Requirements #4

False positive rates must be extremely low

False positives happen when an algorithm mistakes a malicious label for a benign file. Our aim is to make the false positive rate as low as possible. This is complicated by the fact that there are lots of clean files in the world, and they keep appearing.

To address this problem, it is important to impose high requirements for both machine learning models and metrics that will be optimized during training, with the clear focus on low false positive rate (FPR) models.

Requirements #5

Models adaptability

Outside the malware detection domain, machine learning algorithms regularly work under the assumption of **fixed data distribution**, which means that it doesn't change with time. When we have a training set that is large enough, we can train the model so that it will effectively reason any new sample in a test set. As time goes on, the model will continue working as expected.

Requirements #6

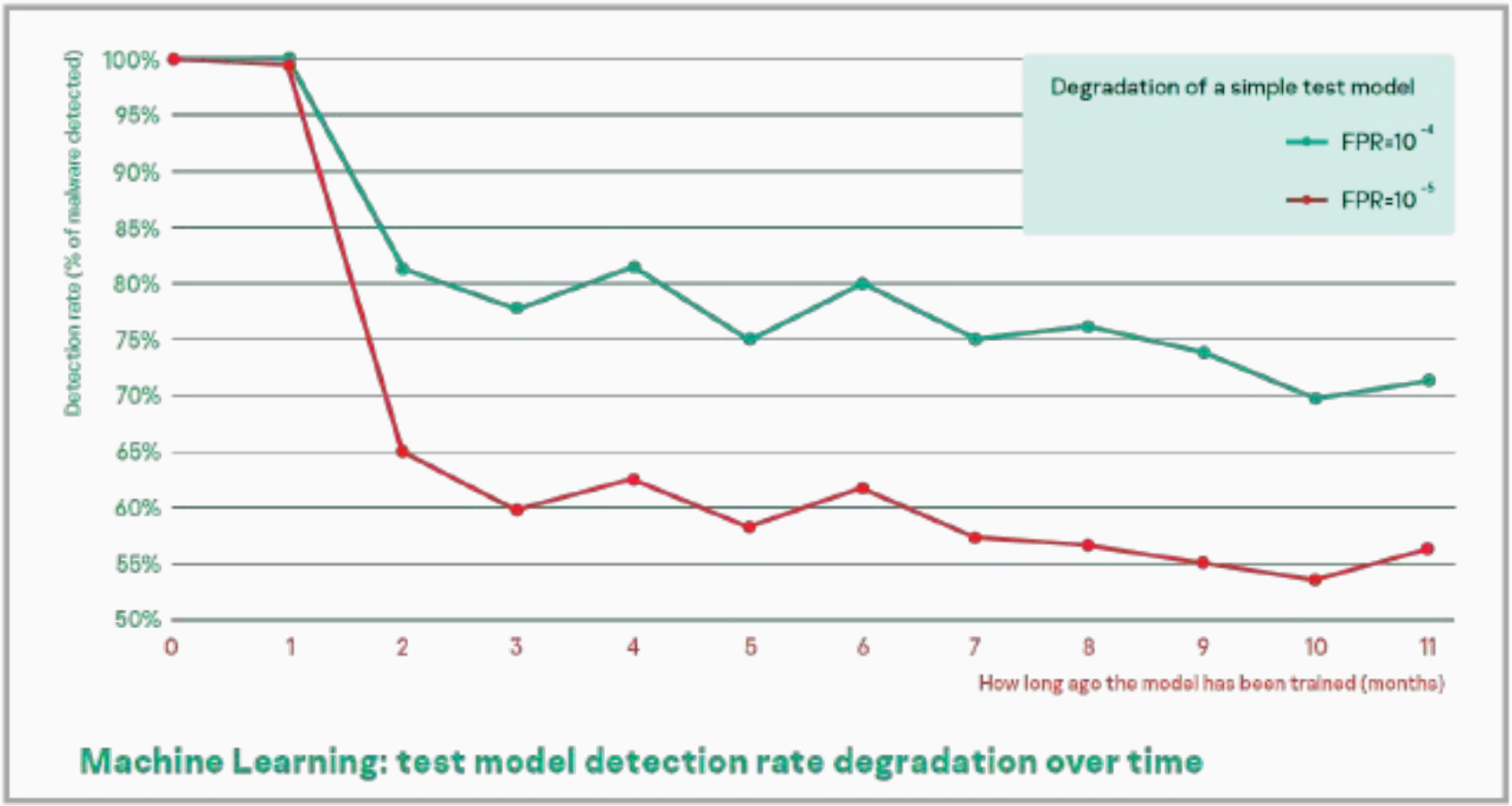
Models adaptability

After applying machine learning to malware detection, we have to face the fact that our data distribution isn't fixed:

- Active adversaries (malware writers) constantly work on avoiding detections and releasing new versions of malware files that differ significantly from those that have been seen during the training phase.
- Thousands of software companies produce new types of benign executables that are significantly different from previously known types. The data on these types was lacking in the training set, but the model, nevertheless, needs to recognize them as benign.

Requirements #6

Models adaptability

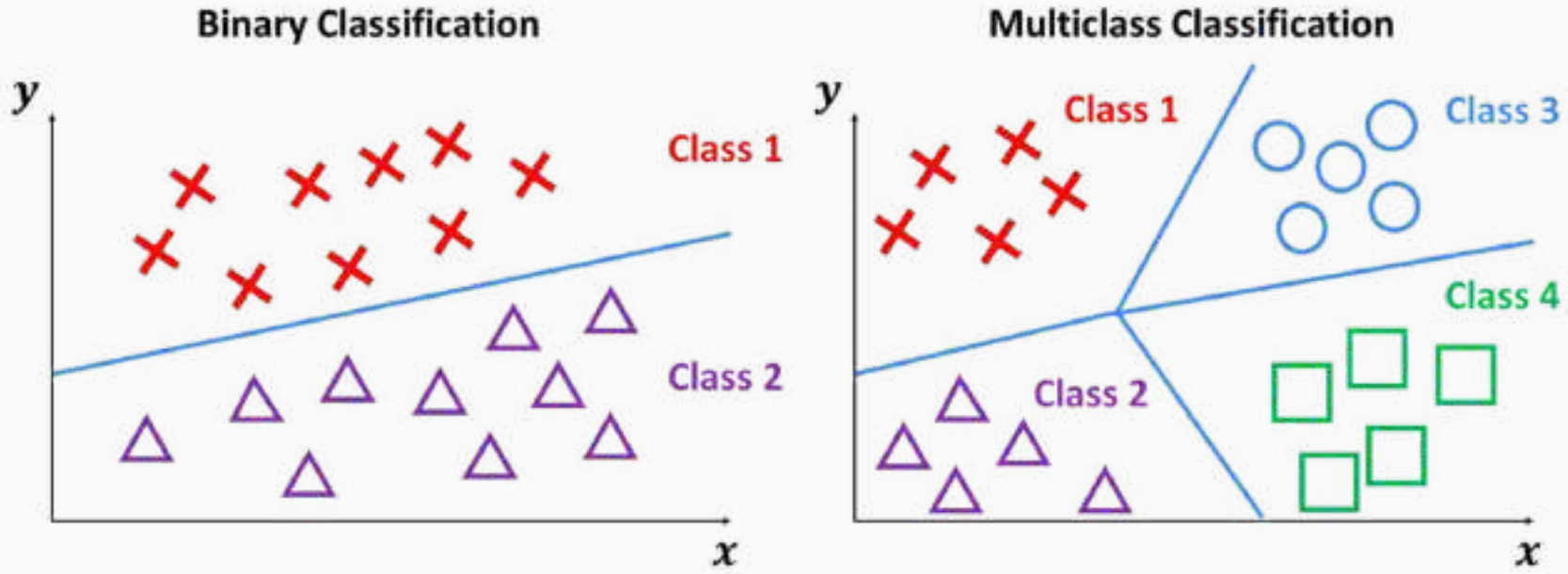


Machine Learning: test model detection rate degradation over time

Multi-Class Classification

Multi-Class Classification

In machine learning and statistical classification, multiclass classification or multinomial classification is the problem of classifying instances into one of three or more classes (classifying instances into one of two classes is called binary classification).



Multi-Class Classification #2

One-Vs-Rest (One-Vs-All)

It involves splitting the multi-class dataset into multiple binary classification problems. A binary classifier is then trained on each binary classification problem and predictions are made using the model that is the most confident.

- Binary Classification Problem 1: red vs [blue, green]
- Binary Classification Problem 2: blue vs [red, green]
- Binary Classification Problem 3: green vs [red, blue]

Multi-Class Classification #3

One-Vs-One

The formula for calculating the number of binary datasets, and in turn, models, is as follows: $(NumClasses * (NumClasses–1))/2$ Each binary classification model may predict one class label and the model with the most predictions or votes is predicted by the one-vs-one strategy.

- Binary Classification Problem 1: red vs. blue
- Binary Classification Problem 2: red vs. green
- Binary Classification Problem 3: red vs. yellow
- Binary Classification Problem 4: blue vs. green
- Binary Classification Problem 5: blue vs. yellow
- Binary Classification Problem 6: green vs. yellow