# Introduction to Artificial Intelligence for security pros

Cylance

# Contents:

- **Introduction:**
- **Clustering:** K-means and DBSCAN Algorithms
- **Classification:**Logistic Regression and Decision Tree Algorithms
- **Probability:**
- **Deep Learning:**

# Introduction

- **Artificial Superintelligence:** Produce computers which are superior to humans in every way
- **Artificial General Intelligence:** Machine as intelligent as a human and equally capable of solving the broad range of problems that require learning and reasoning. (Turing test)
- **Artificial Narrow Intelligence:** Exploits a computer's superior ability to process vast quantities of data and detect parrerns and relationships that would otherwise be difficult or imposible to a human to detect.

## Machine learning in the security domain

The context in wich we have to focus on protecting a specific service is important. For example, in a website's domains, the context is connection. Fortunately, these activities generate a vast ammount of logs by sensors and software. These logs can provide the contextual data we need to identify and ameliorate threats. These is the kind of precessing ML excels.

# Clustering Using the K-Means and DBSCAN Algorithms

The purpose of cluster analysis is to segregate data into a set of discrete groups or clusters based on similarities among their keey features or attributes.

A variety of statistical, artificial intelligence, and machine learning techniques can be used to create the clusters.

In the network security domain, cluster analysis tipically proceeds through a well-defined series of data preparation and analysis operations.

### Step 1: Data selection and sampling

Ideally, we want to collect all of the data generated by our network, but sometimes, this is neither possible or practical, so we apply statistical sampling techiques that allow us to create more manageable subset of the data for our analysis. The sample should reflect the characteristics of the dataset as closeley as possible, or the accuracy of our results may be compromised.

> For example, if we decide to analyze the internet activity for ten different computers, our sample should include representative log entries from all ten systems.

### Step 2: Feature extraction

In this stage, we decide which data elements within our sample should be extracted and subject to analysis. *In ML, this is known as Features*

> In security, the relevant features might include the percentage of ports that are open, closed, or filtered, the application running on each of these ports, and the application version number.

It is good practice to include as many features as needed, excluding the ones you know are irrelevant, because these increase processor time

### Step 3: Feature Encoding and Vectorization

Most ML algorithms require data to be encoded or represented in some mathematical fashon. One very common way data can be encoded is by mapping each sample and its set of features to a grid of rows and columns. Once structured in this way, each sample is refered to as a *vector* The entire set of rows and columns is refered to as a *matrix* The encoding process we use, depends on wether the data representing each feature is *continuous, categorical*, or od some other type.

Continuous data can ocupy any nuber of infinite value within a range of values. For example, any vlaue between 0 ant 100, for cpu tempertarure.

Categorical data is represented by a small set of permissible values within a much more limited range. For example, software version or name. It is inherently useful in defining groups. For example, OS version or SW name, to group computers with similar characteristics. These categories, must be encoded as numbers before mathematical analysis. One example: OS It is important to normalize data, so the algorithms give the same weight to diferent values and don't overblow gigher values.

### Step 4: Computation and Graphin

Once we finish maping data, we can import it to a suitable statistical analysis or data mining application.

## Clustering with K-Means

Clustering analysis introduces the concept of a "Feature space" that can contain thousands of dimensions, one each for every feature in our sample set. Clustering algorithms assign vectors to particular coordinates in this feature space and then measure the distance between any two vectors to determine weather they are sufficiently similar to be grouped together in the same cluster.

### Caveats

- This version of K-Means works with continuous data only
- The data can be meaningfully grouped into a set of similarly sized clusters

### Clustering session

1. Dataset sampled, vectorized, normalized and imported into scikit-learn
2. Data analyst sets K hyperparameter (Number of clusters to generate)
3. K-means randomly selects three vectors and assigsn coordinate in frature set (Centroids)
4. K-means assigns each vector to it's nearest centroid
5. K-means examnes the average distance from each vector to it's centroid. If the centroid's current location matches this computed average, it remains stationary. Otherwise, the centroid is moved to another coordinate wich matches the computed average
6. K-means repeats step 4 for all of the vectors and reassigns them to clusters based on the new centroid location
7. K-means iterates through step 5-6 until
    1. The centroid stops mooving and it's membership remains fixed: *Convergence*
    2. The algorithm completes the maximum number of iterations specified in advance by the analyst

Once the clustering is completed, the analyst can:

- Evaluate accuracy using a variety of validation techiques
- Convert results into a mathematical model, to assess the cluster membership of new samples
- Analyze the cluster results further using aditional statustical and machine learning techiques

### K-Menas pitfalls and limitations

# Classification

# Probability

# Deep Learning

| Host | Ubuntu | Red Hat Enterprise Linux | SUSE Linux Enterprise Server |
|---|---|---|---|
| A | 1 | 0 | 0 |
| B | 0 | 1 | 0 |
| C | 0 | 0 | 1 |

Figure 1: OS