

Network (Entities) Profiling

for classification and anomaly detection

Data Inputs



- Raw data inputs are possible, however it increases the complexity of the machine learning algorithm.
 - Worse results, longer calculation/response times.
- Input data should be the result of raw data processing (complexity reduction).
 - Observation features.
 - Statistical metrics, statistical functions, PCA, scale analysis metrics/descriptors, ...
- Inputs should be normalized.
 - Usually



Variable Reduction



• An event/entity is many times described by multiple descriptors/metrics.

- e.g., mean, variance, maximum, skewness, percentile x%, etc...
- a.k.a. features.

$$e_i = [y_1, y_2, \dots, y_m]$$

• The reduction of variables is mandatory to simplify classification.

• **Principal Components Analysis (PCA)**

- Uses a transformation to convert a set of possibly correlated features into a set of values of uncorrelated variables called principal components.
- The principal components of an event will be a linear combination of the that event features.

$$t_i = e_i W, W = [w_{ij}]_{i,j=1,\dots,m}$$

- The number of principal components is less than or equal to the number of original features.
 - Defined in such a way that the first principal component has the largest possible variance, and the m^{th} (last) component has the smallest variation.
 - The first n components can be chosen to describe the event.
 - W is a $(m \times n)$ matrix.





Data Normalization/Scaling

- Methods:
 - ♦ By maximum absolute value,
 - ♦ By min/max, scaling each feature to a given range,
 - ♦ Standard, removes the mean and scales to unit variance.
 - ♦ ...
- Mandatory when variables/features have different orders of magnitude.
- Removes data bias from quantity, allow to focus on variable and time correlations on data.
 - ♦ e.g., YouTube traffic pattern correlation with video definition must be removed.





Classification vs. Anomaly Detection

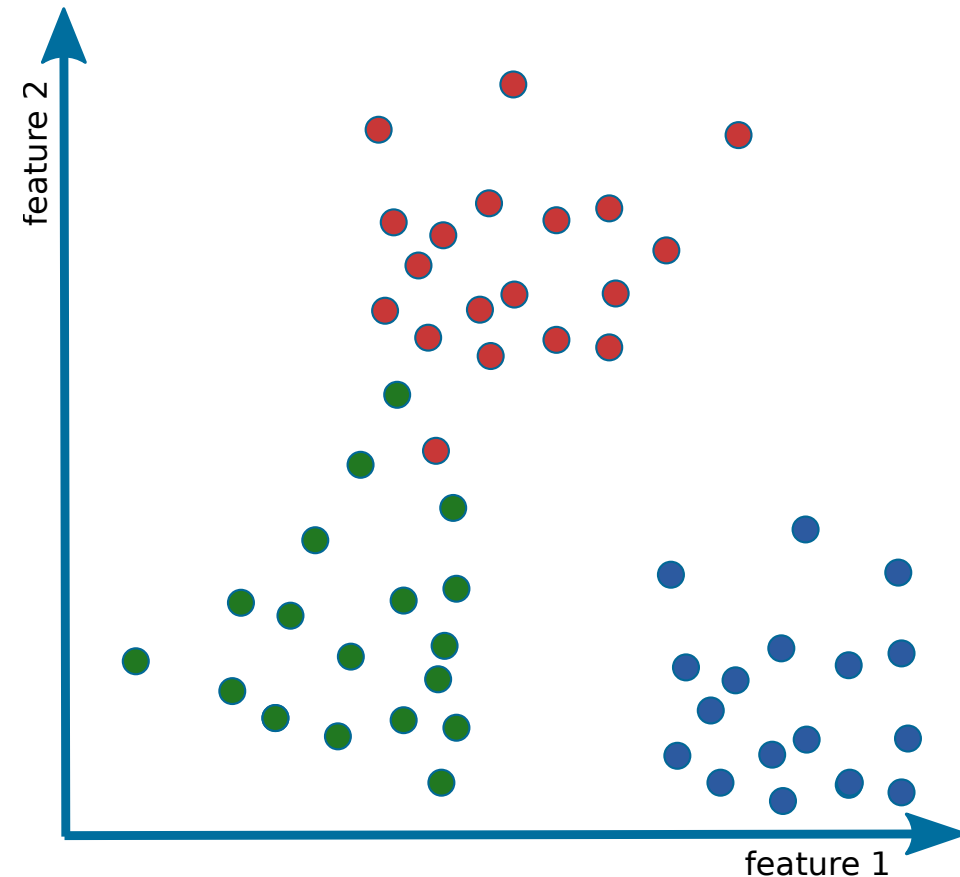
- Classification
 - Requires knowledge (historic data) on all patterns/classes.
 - Does not cope with pattern evolution and appearance of new patterns/classes.
- Anomaly Detection
 - Requires only knowledge (historic data) known of normal patterns/classes.
 - Does not require knowledge (historic data) anomalous patterns/classes.
 - Identify all significantly different patterns as anomalous.
 - Allows to identify never seen anomalies (zero-day detection).
 - May identify as anomalous licit patterns that are evolving



Profile as a N-Dimensional Euclidean Universe



- Each set of N features (reduced or not) in each observation can be seen as a point a N-dimensional Euclidean universe.
- Each point can be:
 - Pre-classified to identify known behaviors/activities.
 - Classified as an belong to a specific group
 - ➔ Short Euclidean distance from the known group points.
 - ➔ Short Euclidean distance from group points previously “grouped” (cluster).
 - Classified as an anomaly.
 - ➔ Large Euclidean distance from the other points.



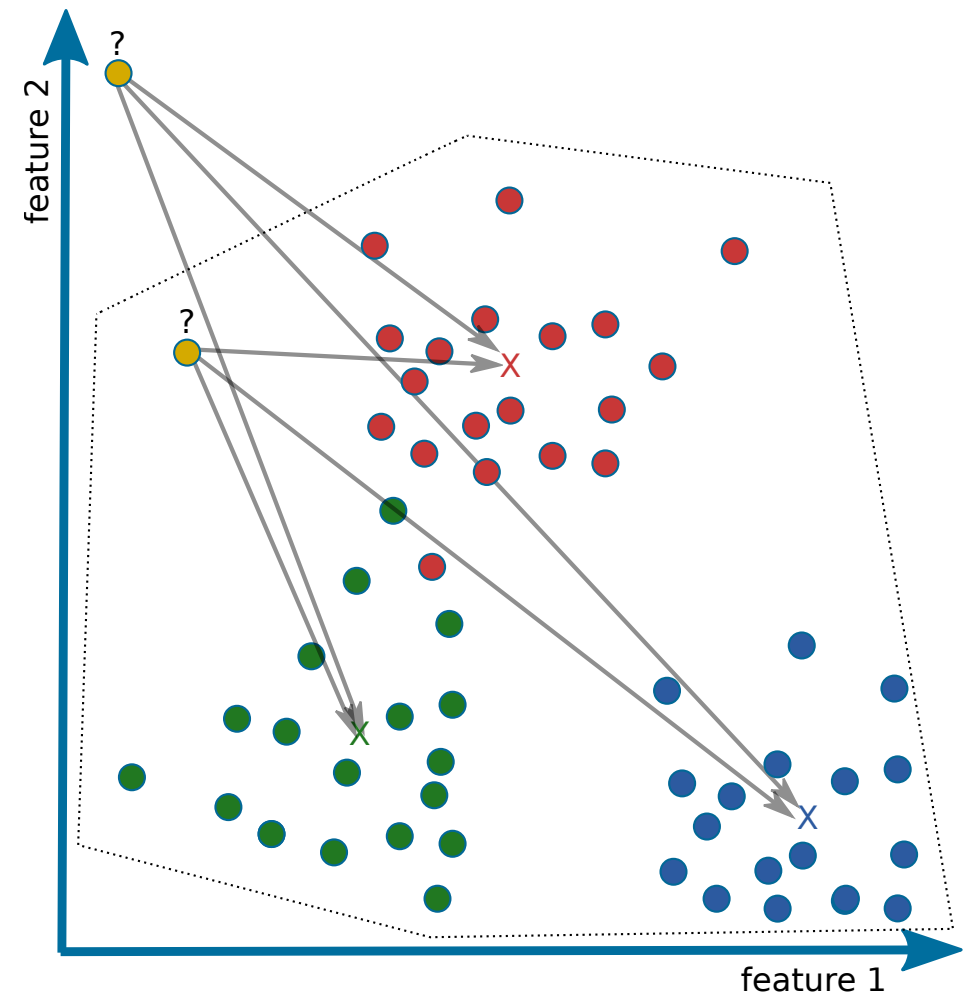
Decision by Statistical Patterns

for differentiation, classification, and anomaly detection



Distances to Central Point(s)

- Group dataset points
 - Use a single group (to detect anomalies),
 - By known classification,
 - By clustering algorithms.
- Find central point of each group.
- For each new dataset point:
 - Calculate Euclidean distances to each group central point,
 - Use distances to classify:
 - ➔ Shortest distance to group,
 - ➔ Probabilistic result based on the relative distances,
 - Ex: $d_1=10, d_2=20, d_3=30 \rightarrow \text{Group1}$
 $\text{prob.} = 10/(10+20+30) = 16.6\%$
 - ➔ Define as anomaly if distance(s) above predefined threshold.

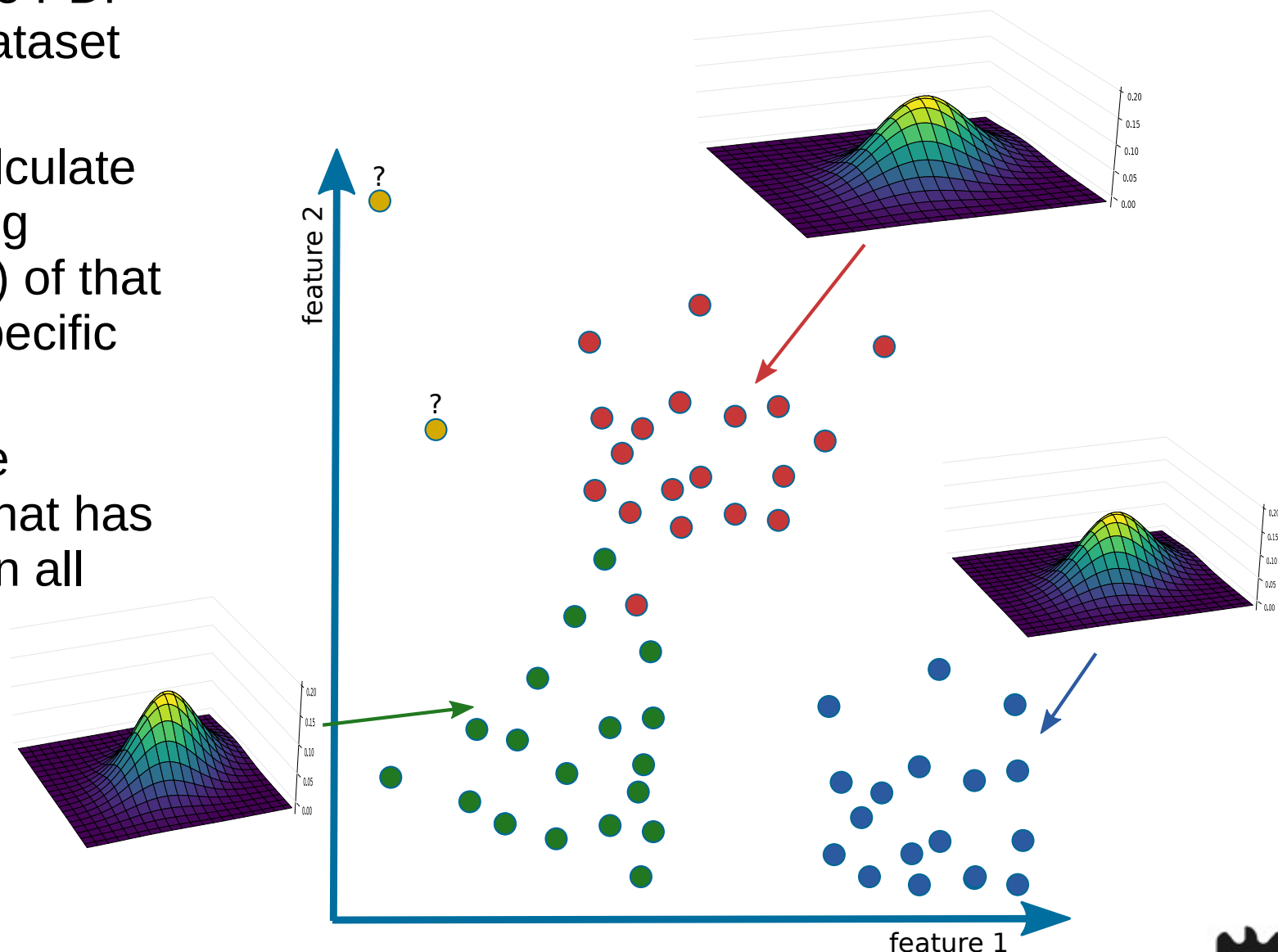


X - Group Central Point
... - Anomaly Boundary



N-Dimensional Distributions

- Infer the multivariate PDF of each group of dataset points.
- For a new point, calculate the probability (using respective the PDF) of that point belong to a specific group.
- An anomaly may be defined as a point that has lower probabilities in all groups.



Decision by Machine Learning

for differentiation, classification, and anomaly detection



Categories

- Supervised learning
 - ♦ Inputs and outputs are given.
 - Outputs may be classification labels or system quantifiers.
 - ♦ Creates a general mapping rule between input and output.
- Unsupervised learning
 - ♦ Only inputs are given.
 - Algorithm must by structure in input data.
 - ♦ Post-classification based on known inputs and found data structure may be done to create a classifier.
- Reinforcement learning
 - ♦ Inputs are given, and “quality” of outputs is defined in terms of reward and penalization (cost functions) relative to the problem goal.





Approaches

- Clustering
- Support vector machines
- Artificial neural networks
 - ◆ Composed of one input and one output layer, and at most one hidden layer in between.
- Deep learning
 - ◆ ANN with more than three layers (including input and output).
 - More than one hidden layer.
- Other
 - ◆ Bayesian networks
 - ◆ Decision tree learning
 - ◆ Genetic algorithms
 - ◆ ...



Classification / Clustering



- Clustering is the process of grouping (classifying) a set of objects in such a way that objects in the same group (cluster) are more “similar” to each other than to those in other clusters.

- Algorithms:

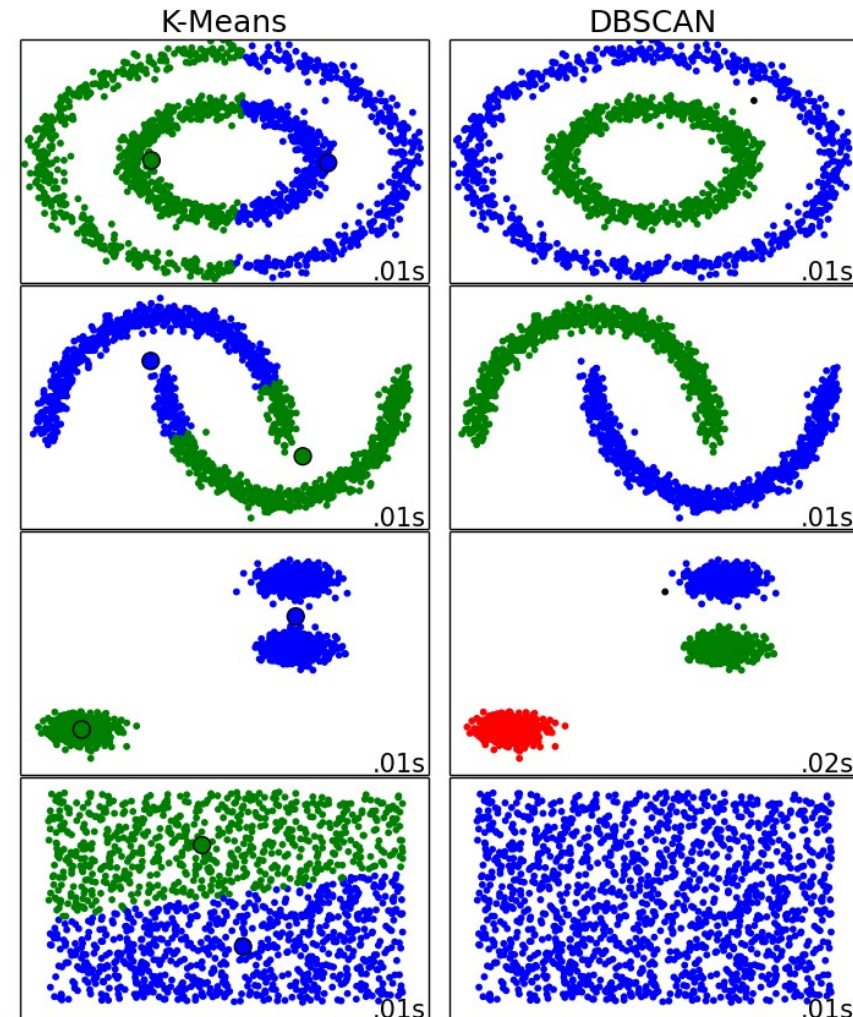
- K-Means

- Requires the a priori knowledge of the number of clusters.
 - Uses the distances between points as metric.

- DBSCAN

- Requires the a priori definition of the neighborhood size.
 - Uses the distances between nearest points as metric.

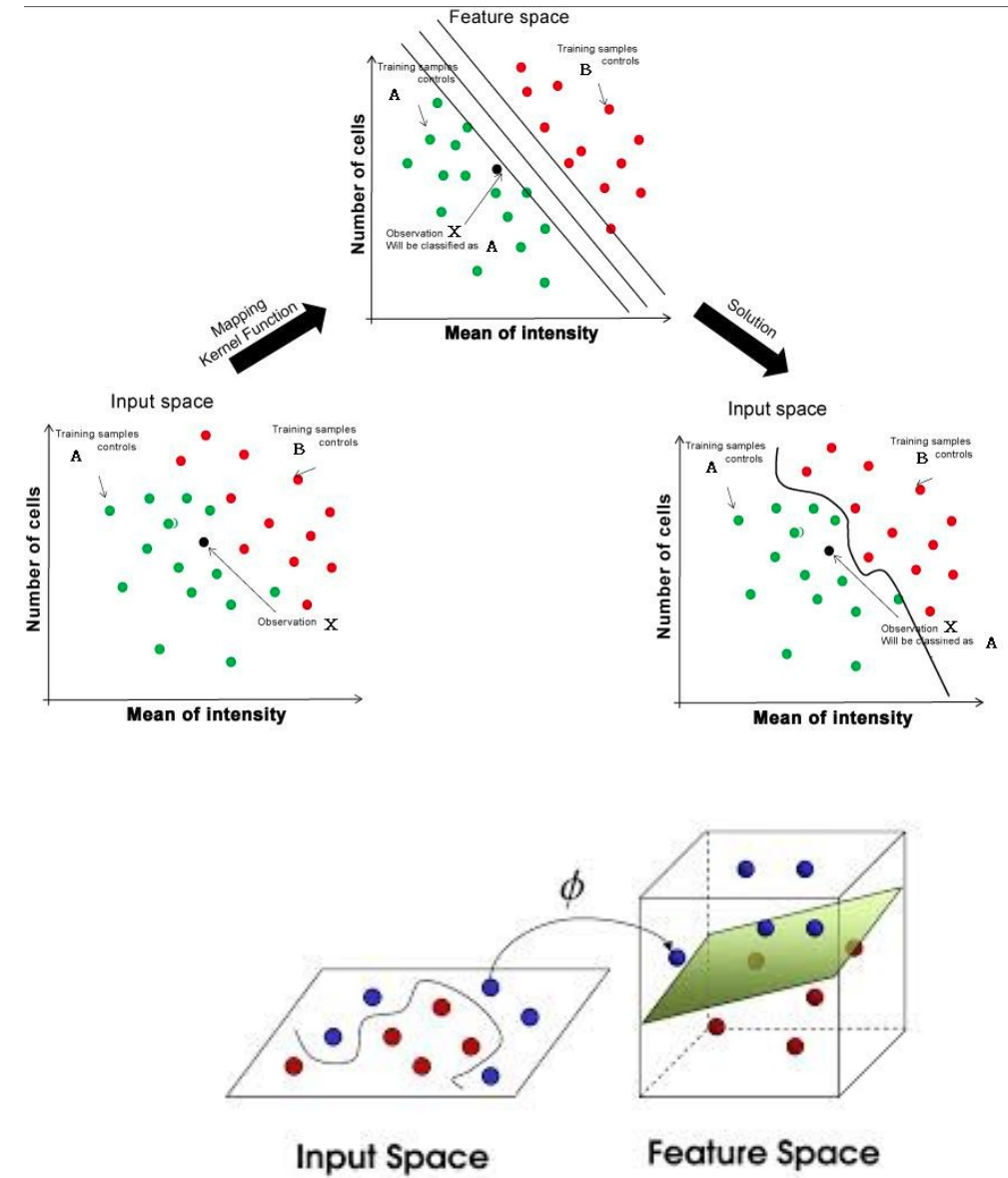
- Others...





Support Vector Machines (SVM)

- Classification defined by a separating hyper-plane-
- Optimal hyper-plane for linearly separable patterns.
- Kernel functions allow the separation of patterns that are not linearly separable by transformations of original data.
- Solutions found using a minimization problem.



One-Class SVM vs. N-Class SVM

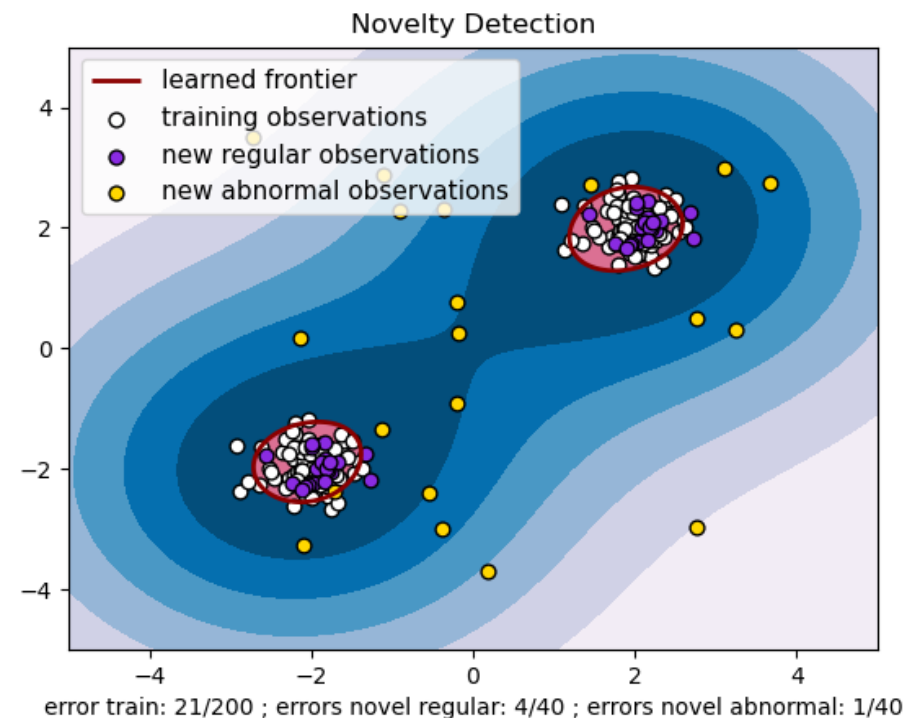
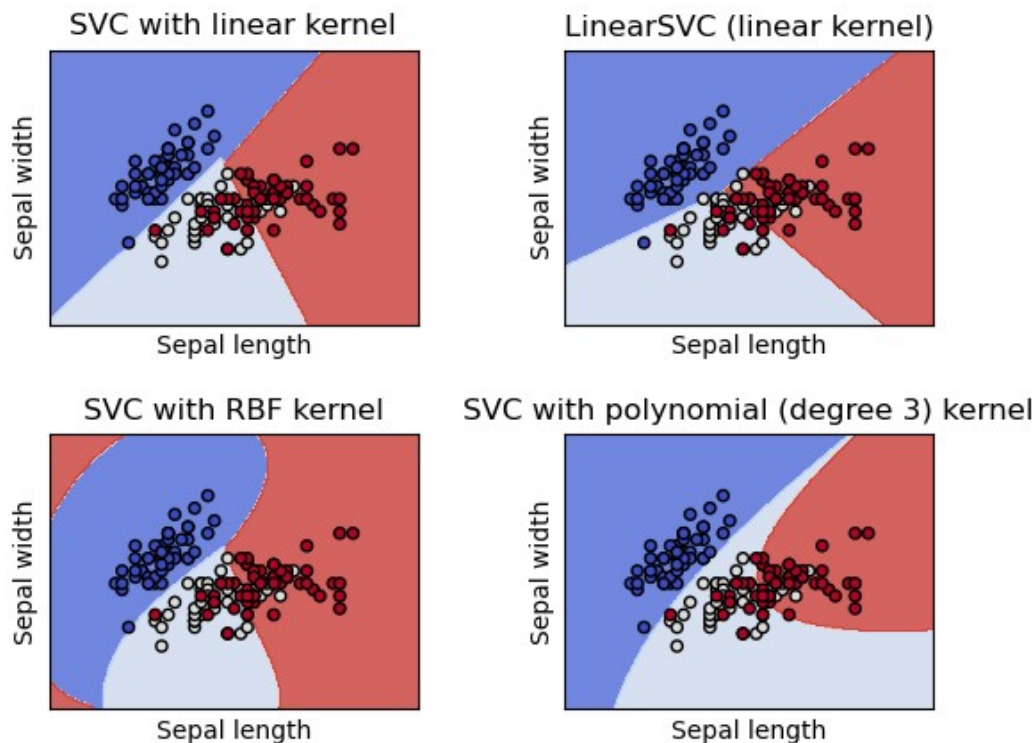


- N-Class SVM

- Infers boundaries between each class.

- One-Class SVM

- Infers “a boundary” that contains all known normal/licit traffic.





Decision Trees

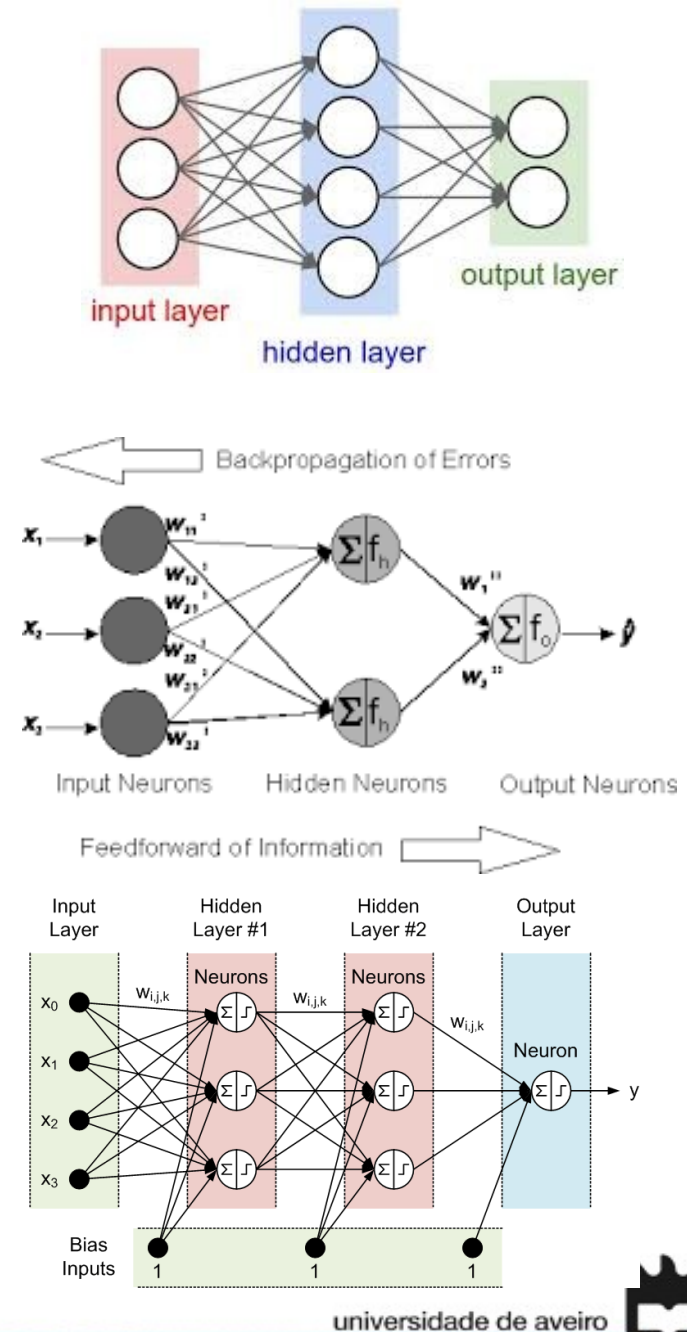
- Data partitions by branching decisions based on features values.
- Decision based on:
 - Location of an observation on the decision tree;
 - Location of an observation on multiple decision trees (forest);
 - Number of partitions/branches required to isolate an observation.
- Variants:
 - Tree Regressor
 - Classification based on data partitions (over branches).
 - Isolation Forests
 - Detects anomalies based on the low number of branches (data partitions) required to isolate an observation.
 - Random Forests
 - Uses multiple tree classifiers on various random sub-samples of the dataset.
 - Averages the results.





Artificial Neural Networks

- Composed by input and output layers, and an optimal hidden layers
 - More than one hidden layer, becomes a deep learning NN.
- Hidden and output layers, perform a weighted sum of the values outputted by the nodes of the previous layer and applies an activation function.
 - Activation functions: linear, tanh, arctan, etc...
 - Weights define the NN, and must be inferred by a training algorithm.
 - ➔ Each node-node connection have a different weight.
- Training algorithms adjust connection weights to minimize the error between inputs and training outputs.
 - Back propagation of error.
 - Levenberg-Marquardt algorithm, Newton and quasi-Newton methods, Gradient descent, and Conjugate gradient.
- Some nodes/layers may have bias inputs to activate/deactivate and/or offset node outputs.



Overview

Nearest Neighbors

Linear SVM

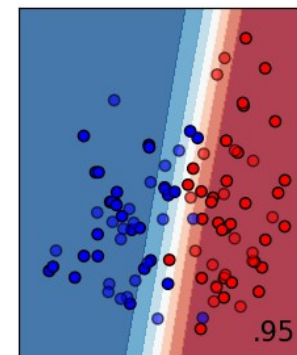
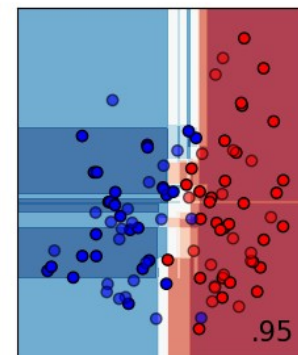
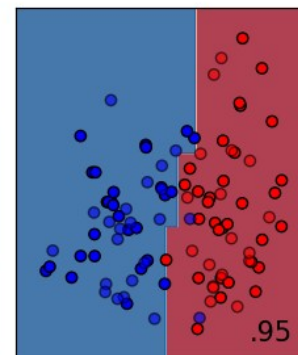
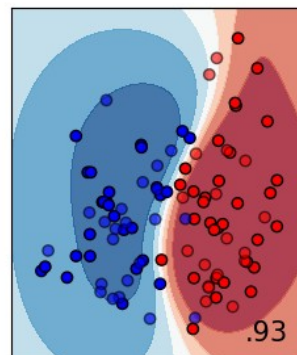
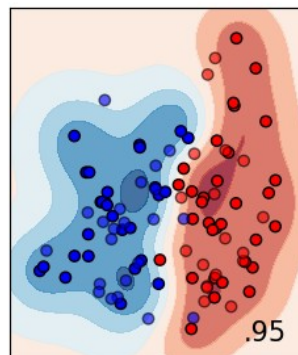
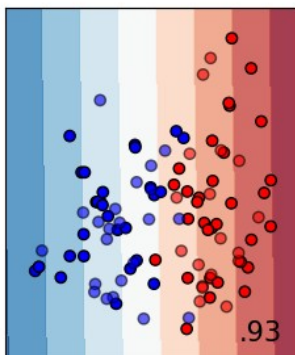
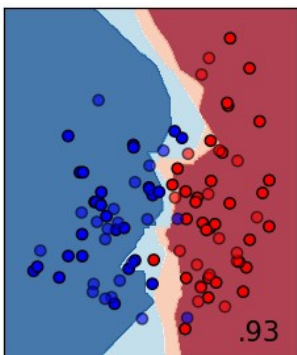
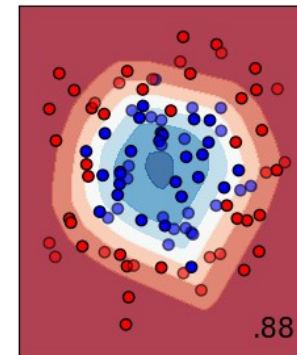
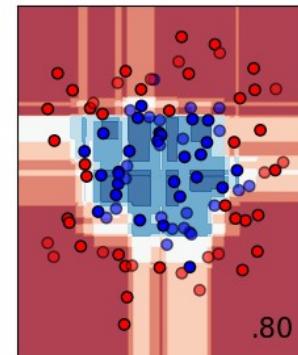
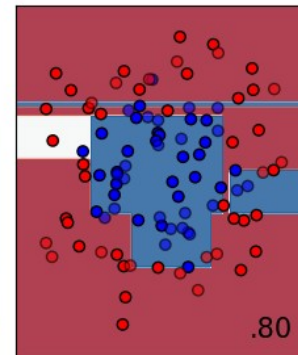
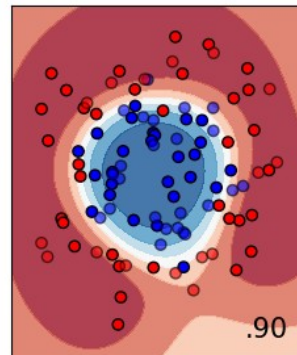
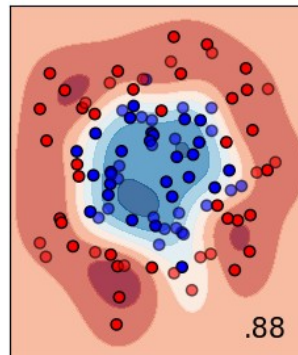
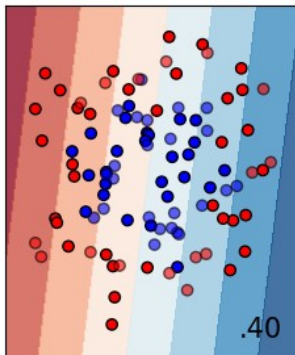
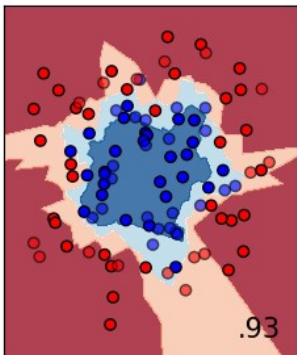
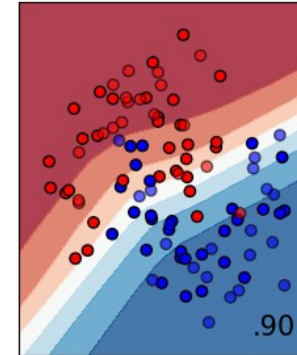
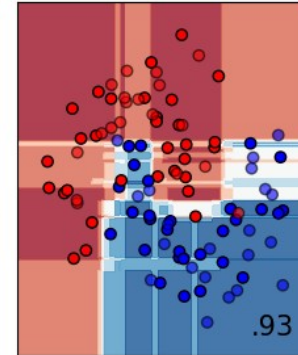
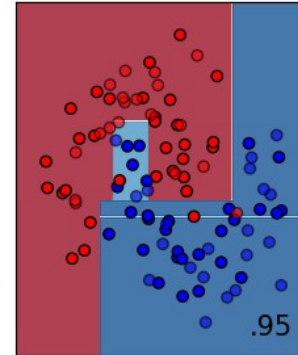
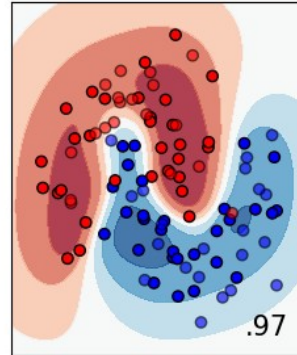
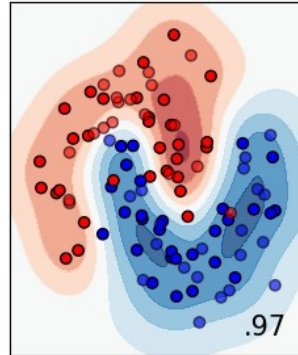
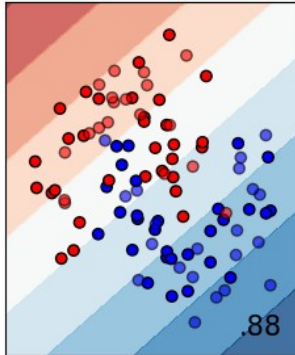
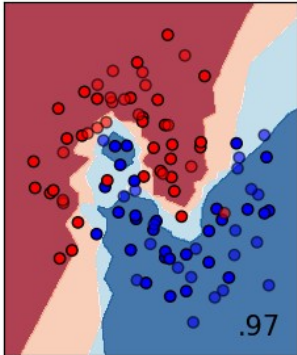
RBF SVM

Gaussian Process

Decision Tree

Random Forest

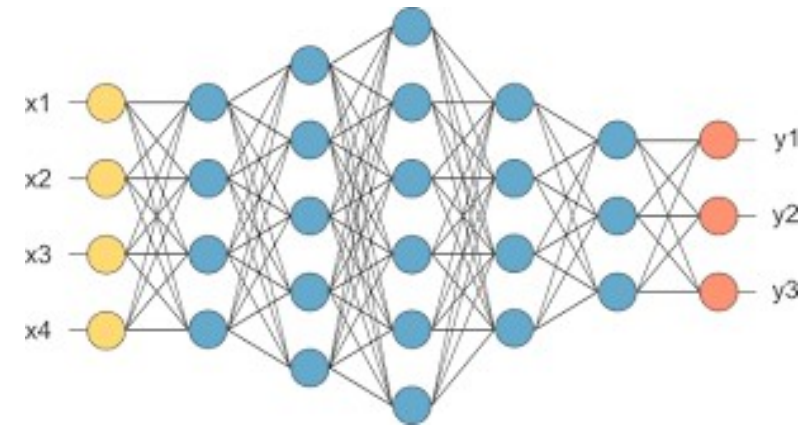
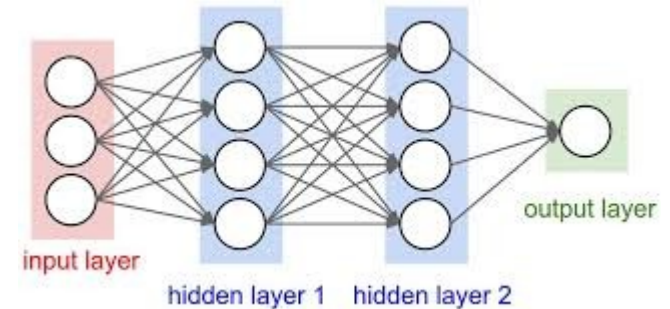
Neural Net





Deep Learning

- Supervised learning algorithms
 - Logistic Regression.
 - Multilayer perceptron.
 - Deep Convolutional Network.
- Unsupervised and semi-supervised learning algorithms
 - Auto Encoders
 - Denoising Autoencoders
 - Stacked Denoising Auto-Encoders
 - Restricted Boltzmann Machines
 - Deep Belief Networks



Ensemble (1)

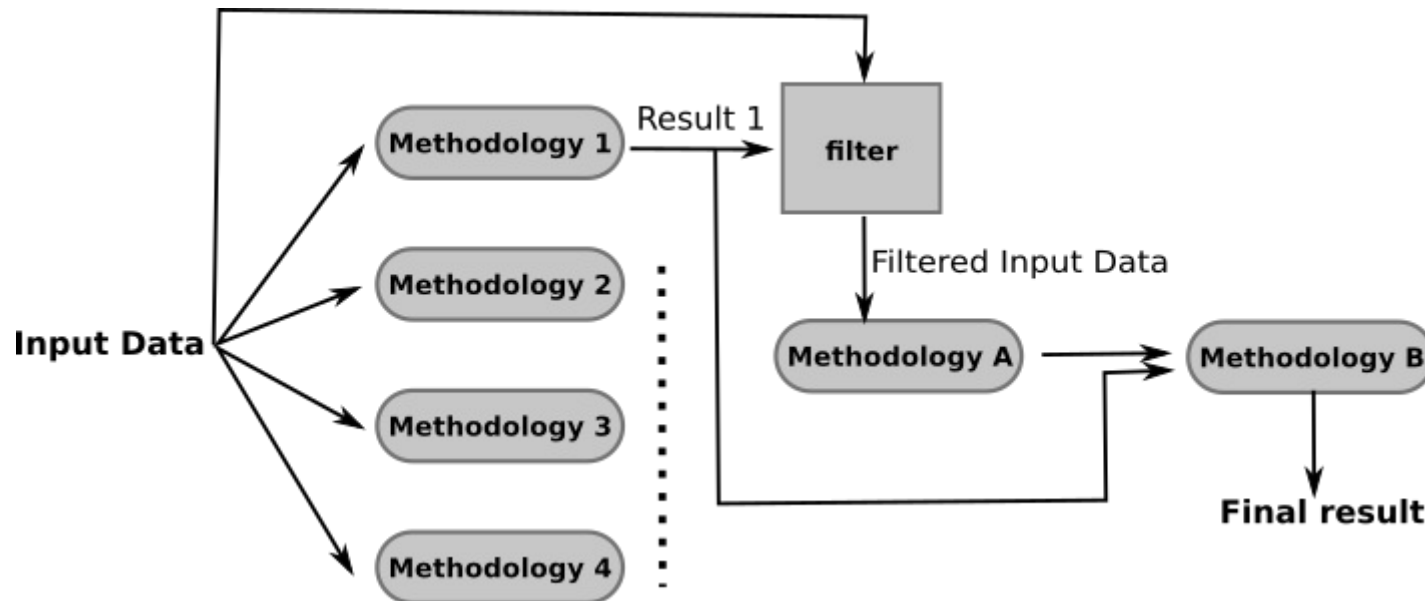
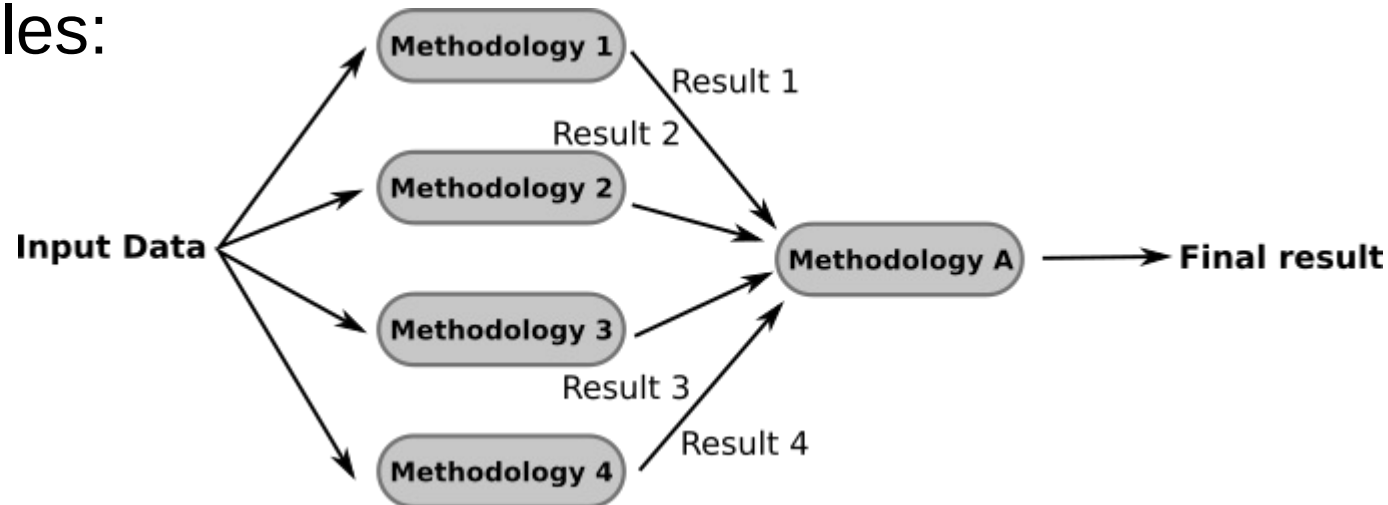


- Ensemble methods use multiple learning methodologies to obtain better than the individual methods.
- Methods:
 - Bayes optimal classifier
 - ➔ Final decision based on the probabilities given by each methodology
 - Bagging
 - ➔ Final decision based on the results given by each methodology with equal weight.
 - ➔ Input data may differ between methodologies
 - Aims to decrease final result variance.
 - Boosting
 - ➔ Final decision based on different methodologies applied in sequence (to correct wrong classifications by the previous methodology).
 - ➔ Previous results may be used to filter input data given to next level classification methodologies.



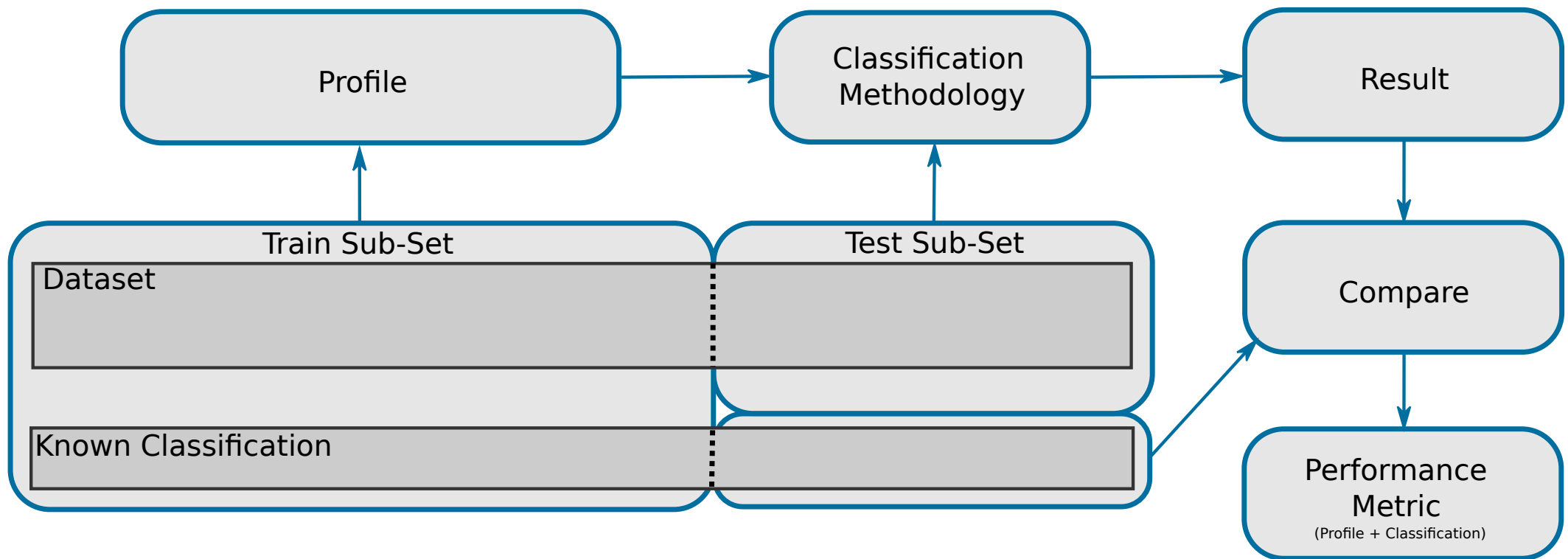
Ensemble (2)

- Examples:



Performance Evaluation

Evaluation Process





Metrics

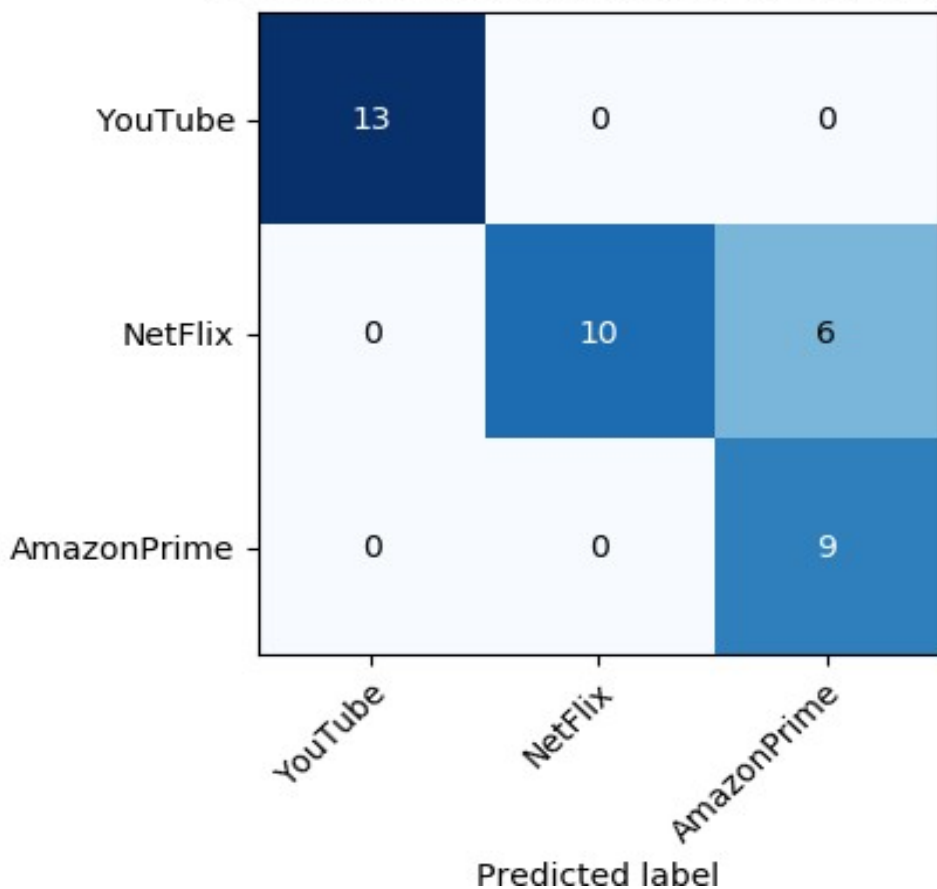
- True Positive (TP) - Correctly predicted positive
 - True Negative (TN) - Correctly predicted negative
- False Positive (FP) - Wrongly predicted as positive
 - False Negative (FN) - Wrongly predicted as negative
- Metrics
 - $\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$
 - $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
 - $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
 - $\text{F1-Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$

Actual Class	Predicted class		
		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative



Confusion Matrix

Confusion matrix, without normalization



Normalized confusion matrix

