MACHINE LEARNING

An Introduction

Ahmad Ali Abin

Faculty of Computer Science and Engineering
Shahid Beheshti University

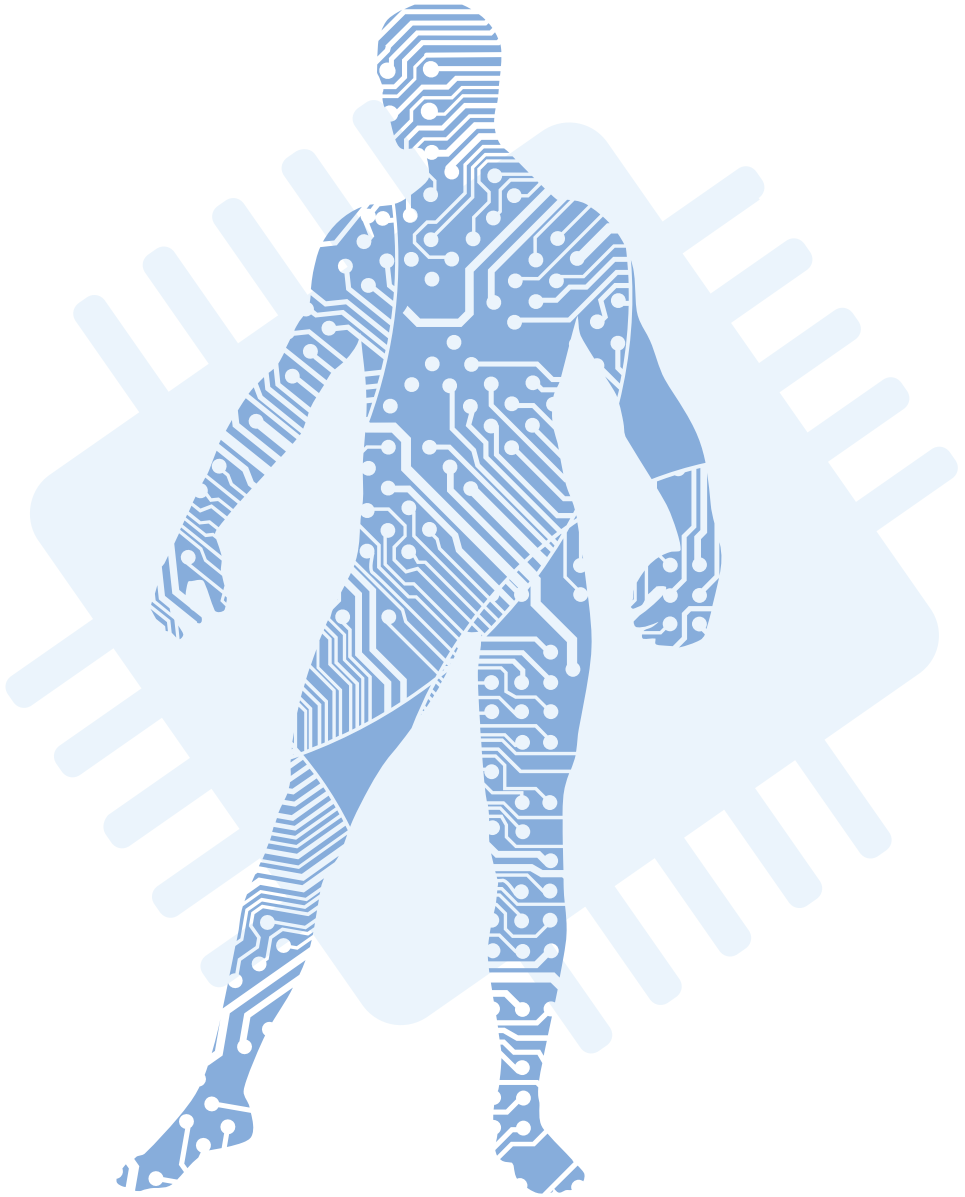
Agenda

01 A brief about machine learning

02 Seven steps of machine learning

03 Tips and tricks

04 Conclusion

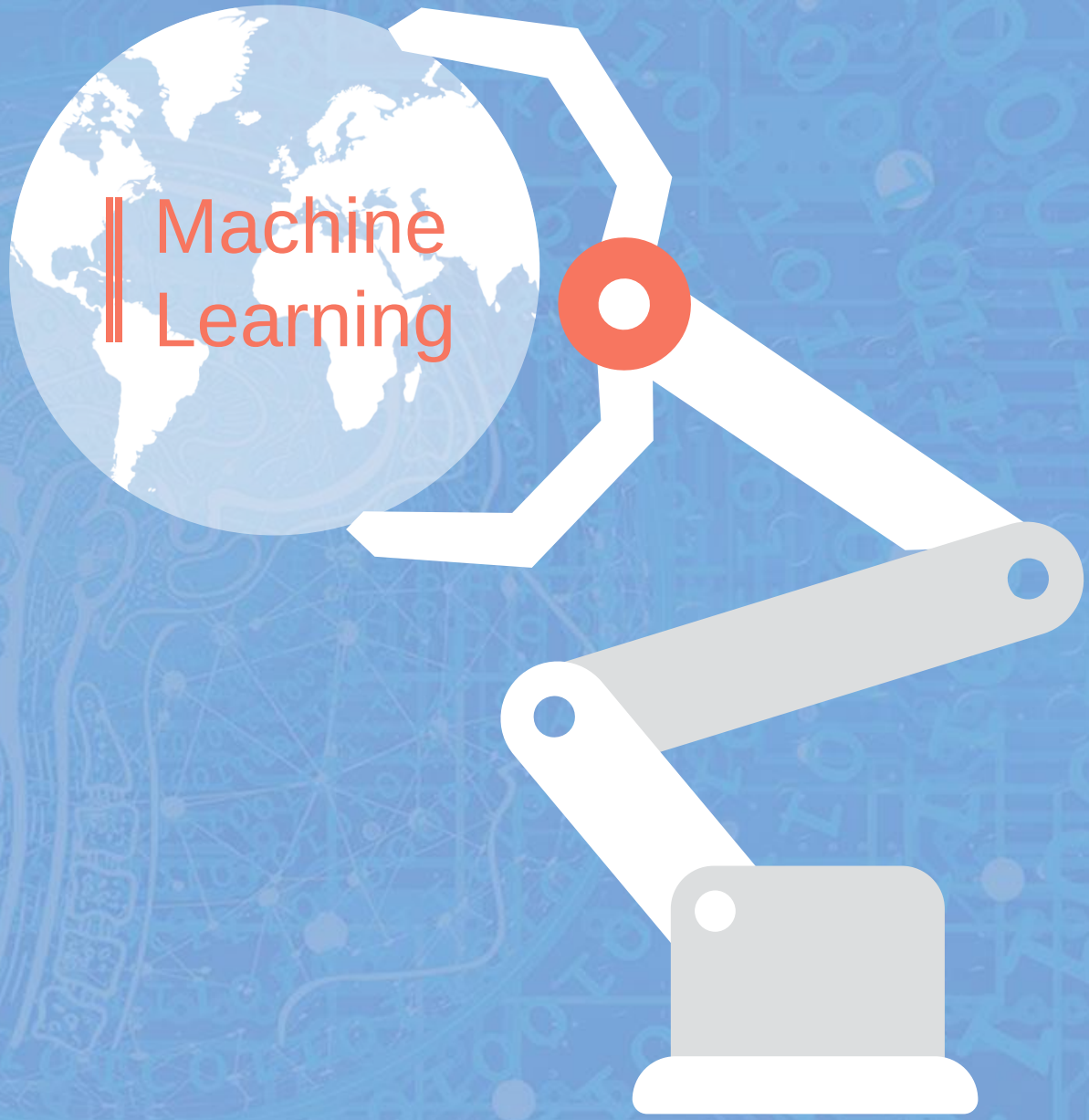


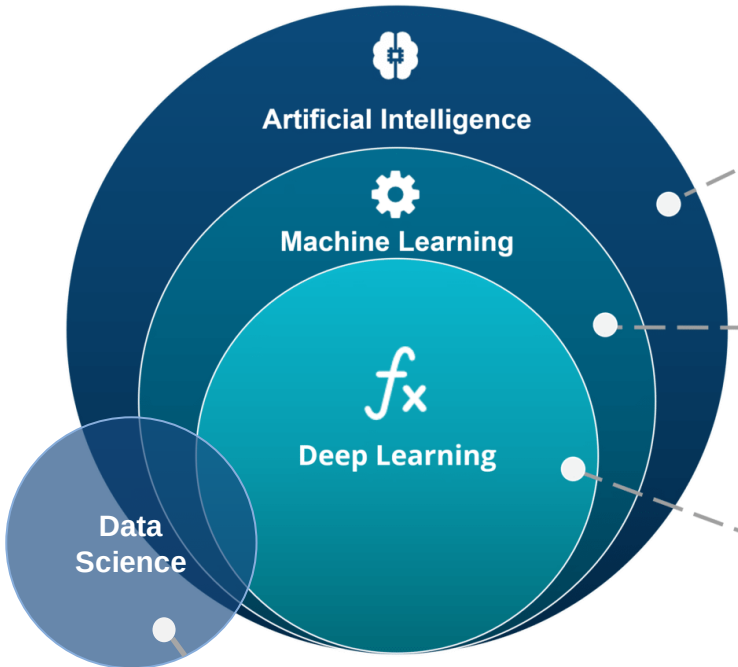
Machine Learning

An introduction to
Machine Learning

What is Machine Learning?

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that which makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.





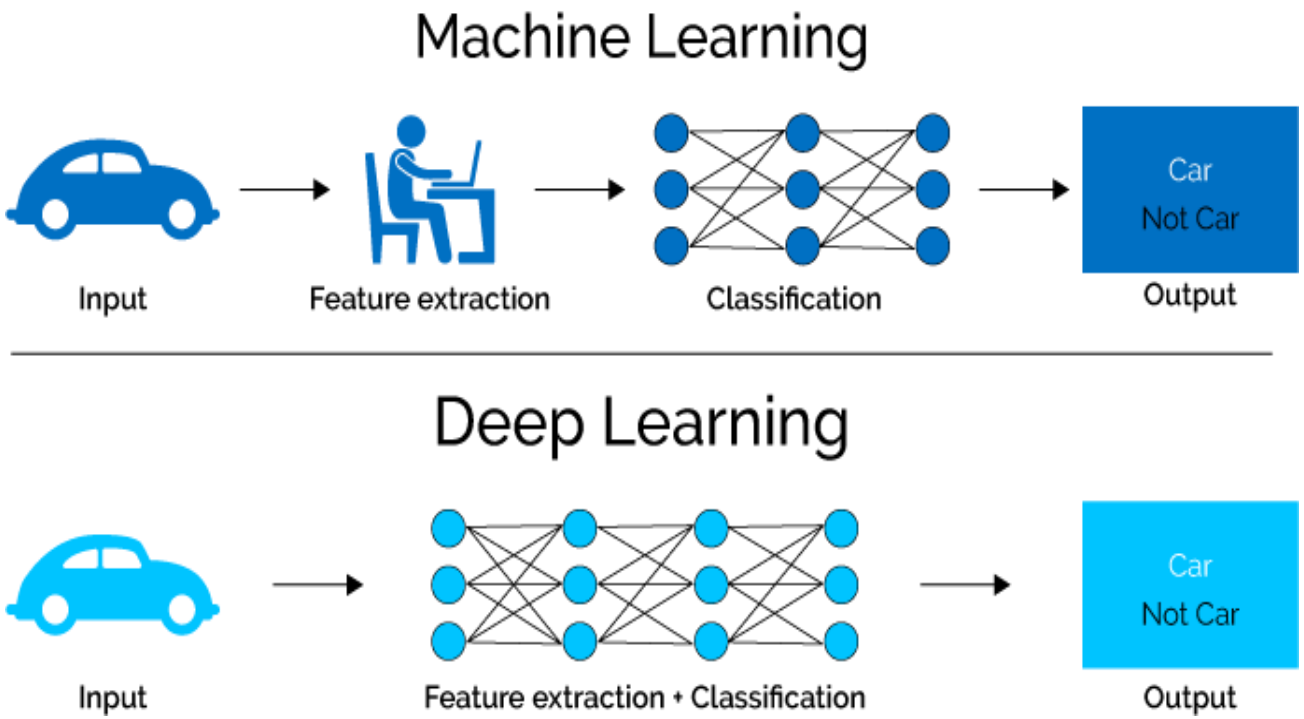
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

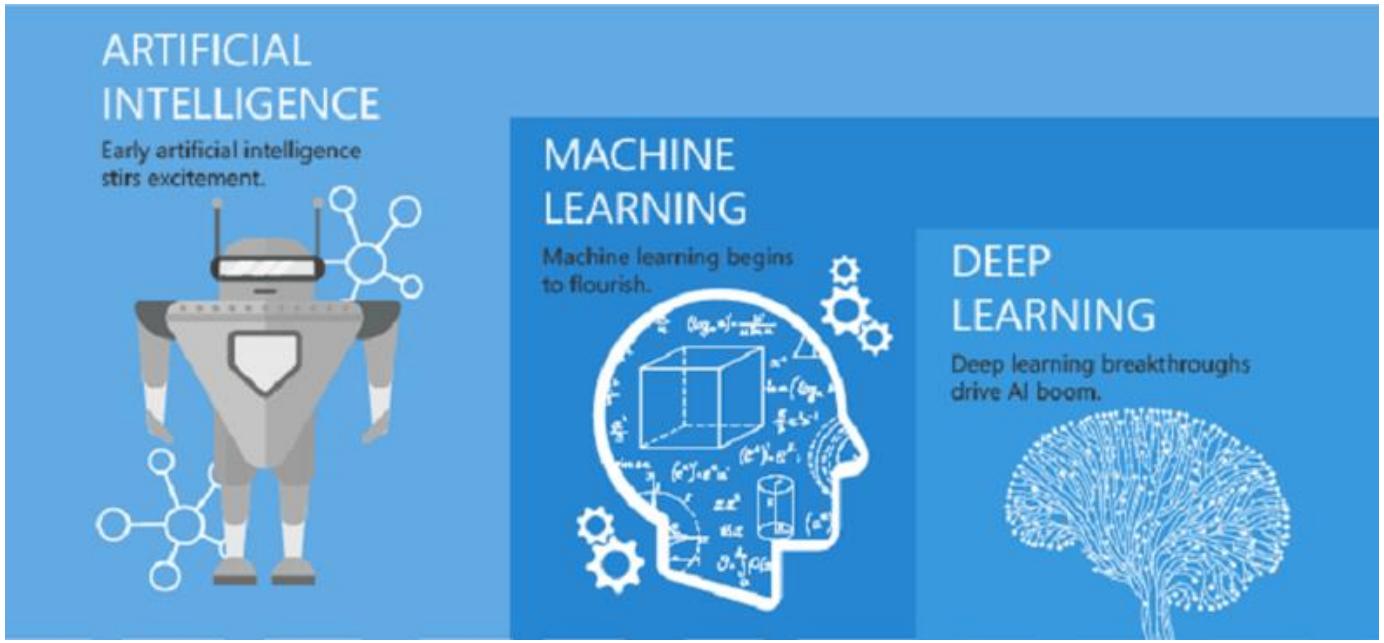
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

DATA SCIENCE
Apply AI, ML and DL techniques to create actual products

ML vs. DL


Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output

Timeline

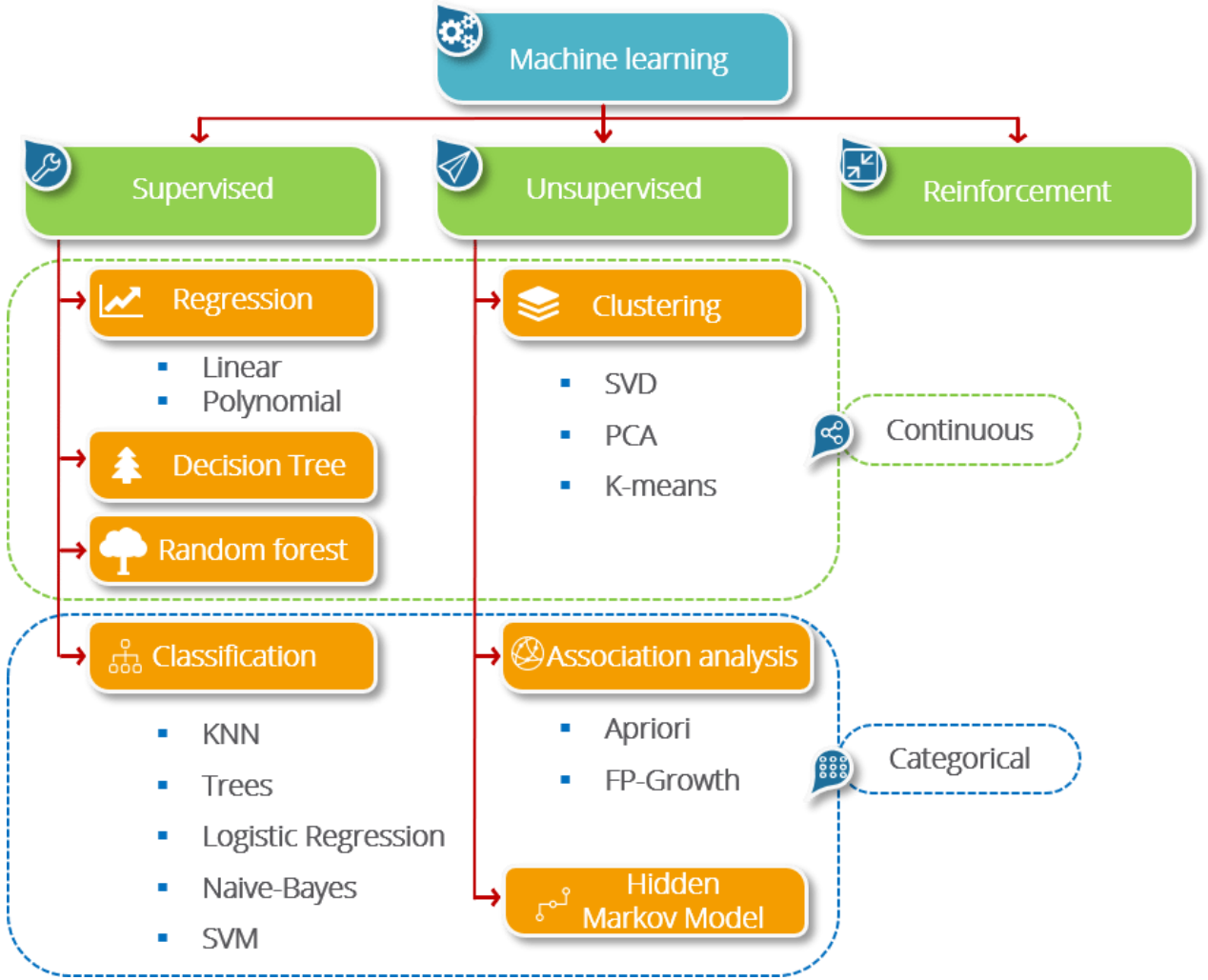


1950's
Artificial Intelligence

1980's
Machine Learning

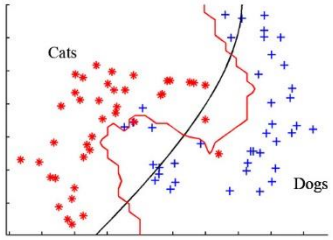
2010's
Deep Learning

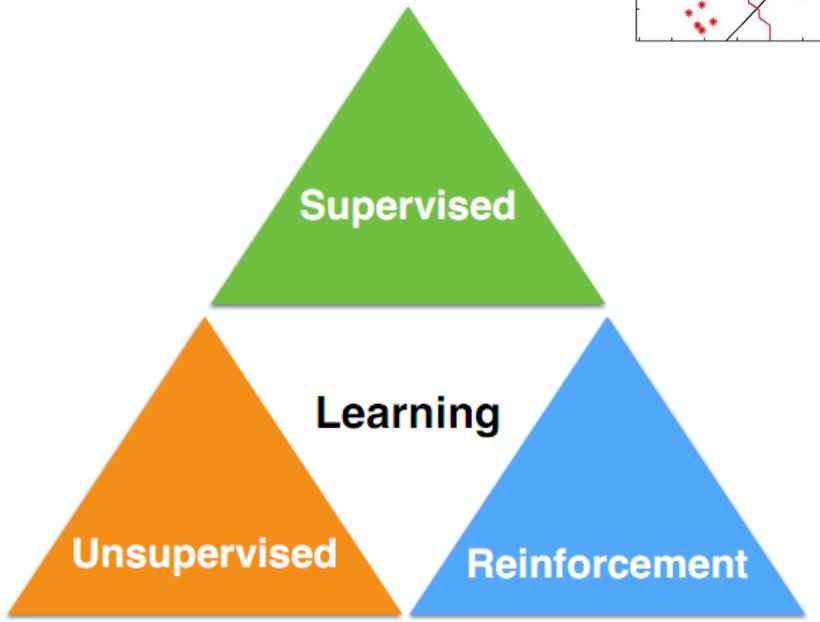
Types of ML algorithms

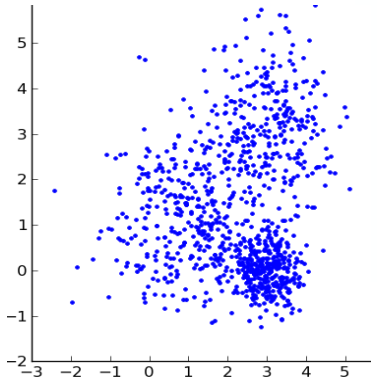


Types of ML algorithms

- Labeled data
- Direct feedback
- Predict outcome/futur

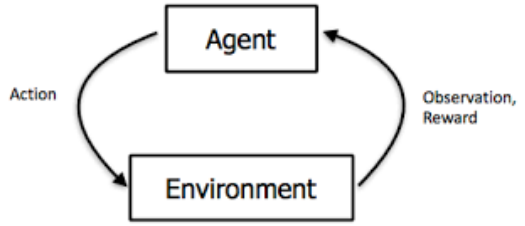






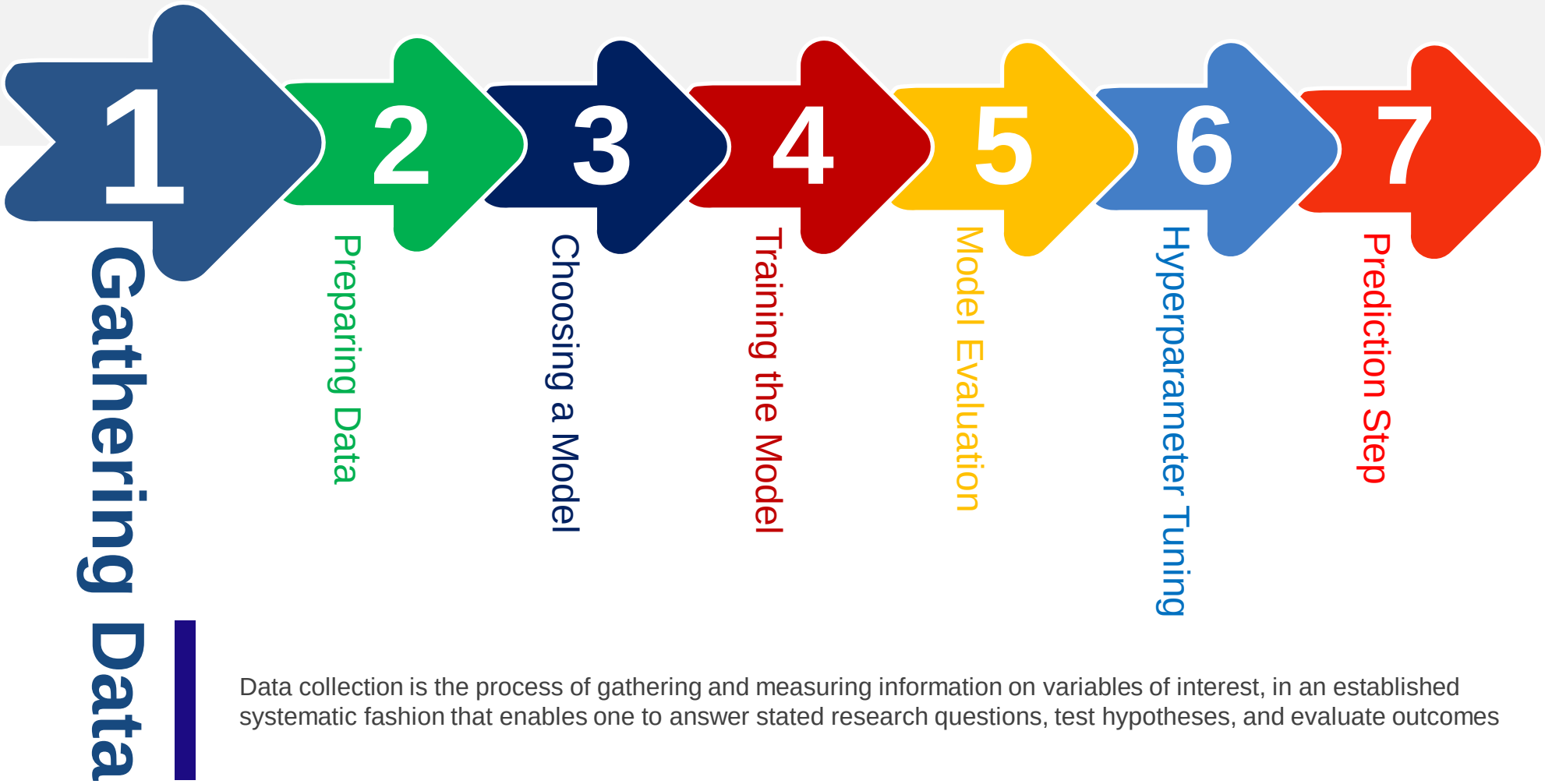
- No labels
- No feedback
- "Find hidden structure"

- Decision process
- Reward system
- Learn series of actions



7 Steps of Machine Learning

7 Steps of Machine Learning



Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes

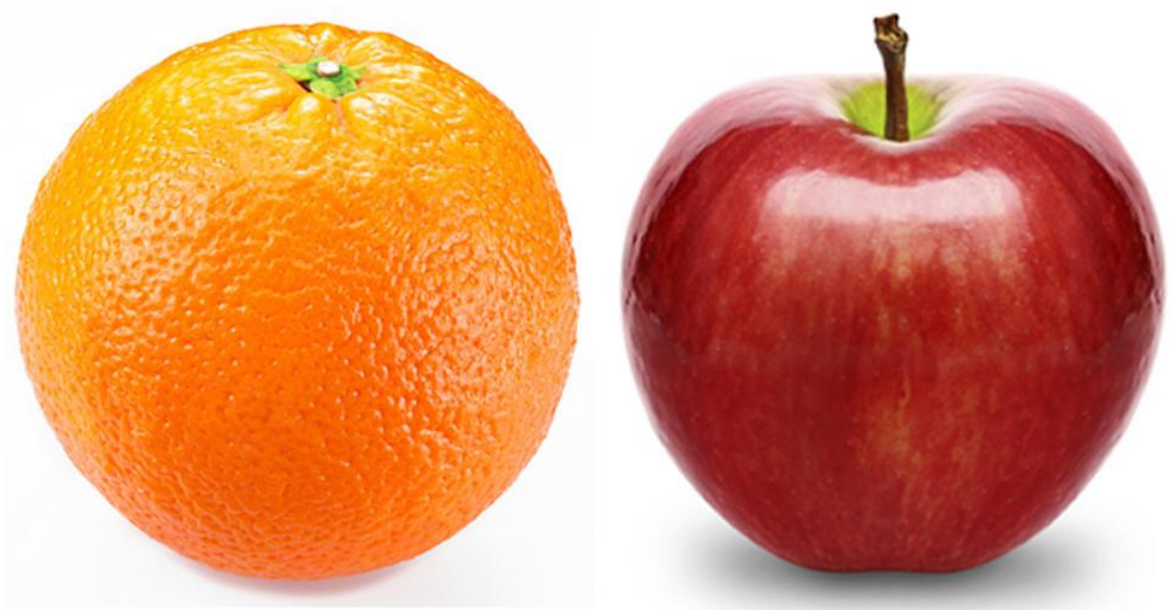
7 Steps of Machine Learning

1

Gathering Data

What is data?

Facts that are collected through observation and used for analysis. Color, size, shape, softness, texture, etc.

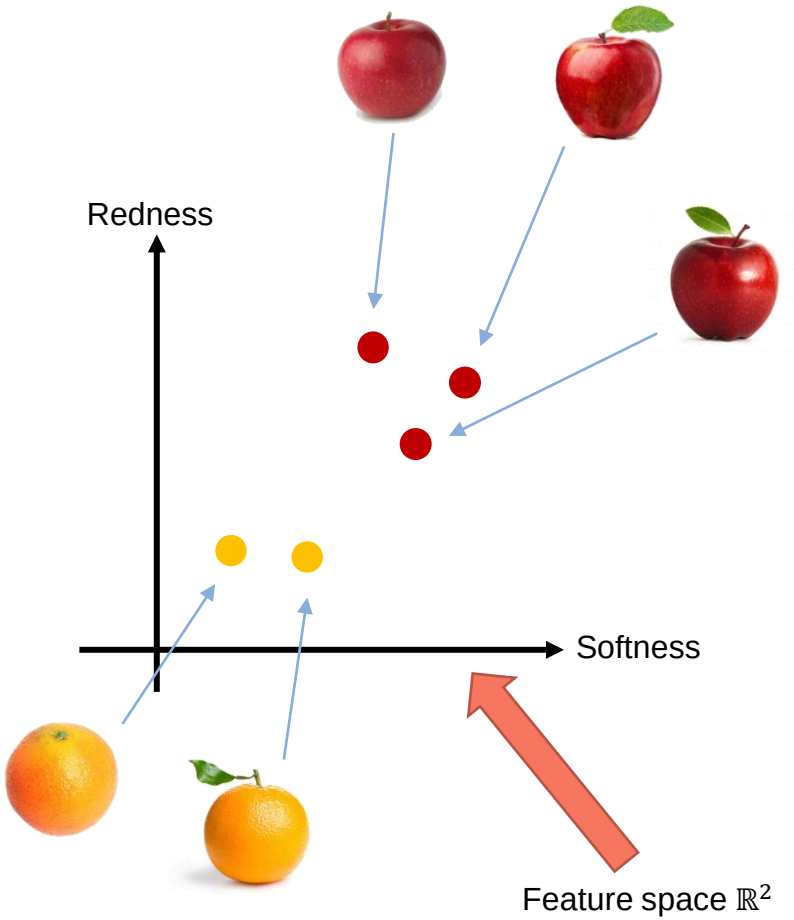


7 Steps of Machine Learning

1

Gathering Data

	Feature		
Softness (0-100)	Redness (0-255)	Class	
80	200	Apple	Instance
25	35	Orange	
75	250	Apple	
90	180	Apple	
.	.	.	
12	53	Orange	

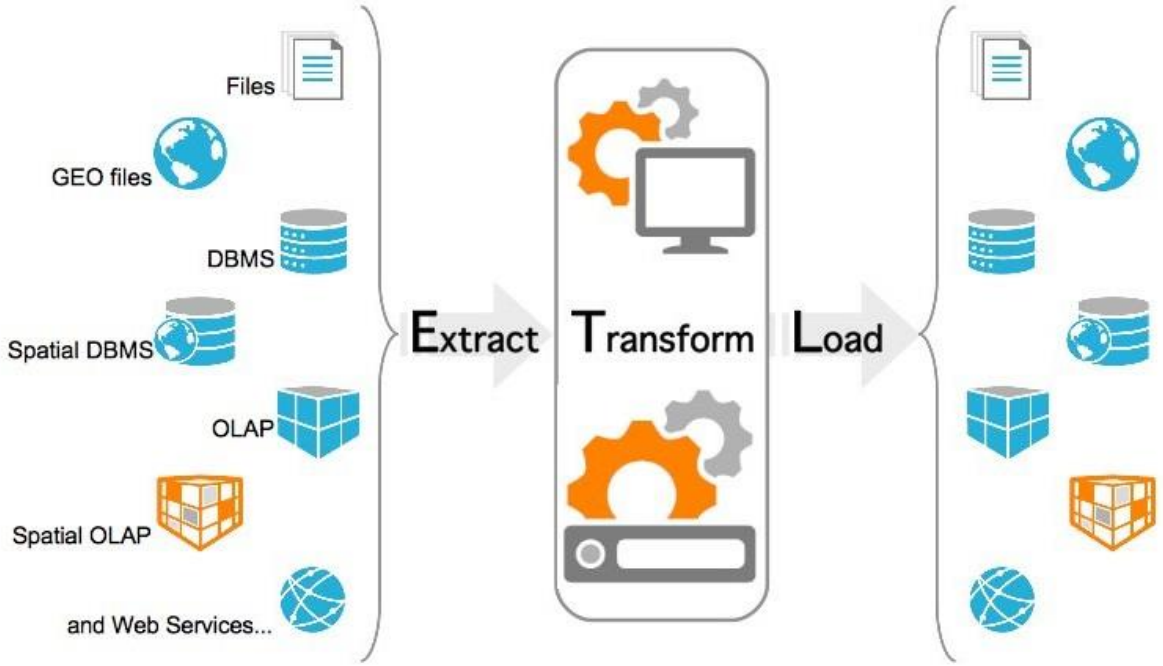


7 Steps of Machine Learning

1

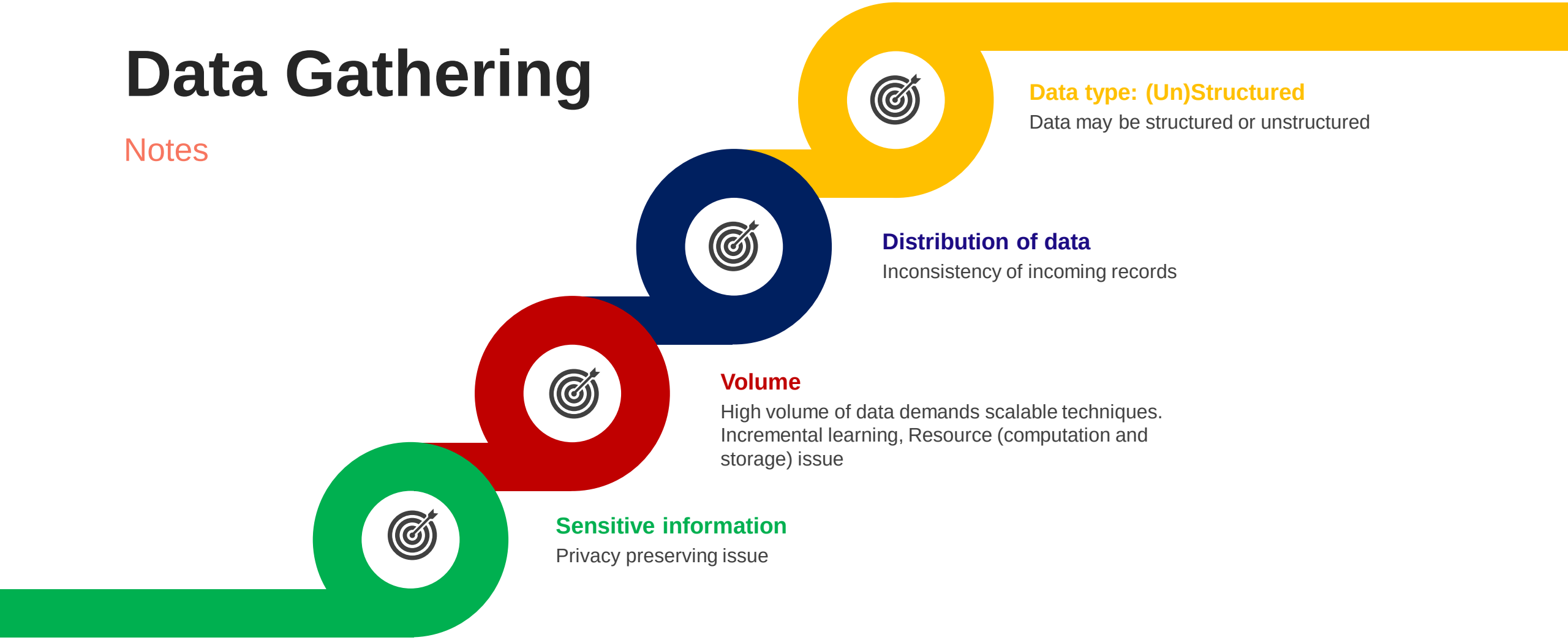
Gathering Data

Data collection is the process of **gathering** and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes



Data Gathering

Notes



Data type: (Un)Structured

Data may be structured or unstructured

Distribution of data

Inconsistency of incoming records

Volume

High volume of data demands scalable techniques. Incremental learning, Resource (computation and storage) issue

Sensitive information

Privacy preserving issue

7 Steps of Machine Learning

Once the data is collected, it's time to assess the condition of it, including looking for trends, outliers, exceptions, incorrect, inconsistent, missing, or skewed information.

Preparing Data

2

Preparing Data

1. Data Exploration and Profiling

Once the data is collected, it's time to assess the condition of it, including looking for trends, outliers, exceptions, incorrect, inconsistent, missing, or skewed information.

2. Formatting data to make it consistent

The next step in great data preparation is to ensure your data is formatted in a way that best fits your machine learning model. If you are aggregating data from different sources, or if your data set has been manually updated by more than one stakeholder, you'll likely discover anomalies in how the data is formatted (e.g. USD5.50 versus $5.50).

3. Improving data quality

Here, start by having a strategy for dealing with erroneous data, missing values, extreme values, and outliers in your data.

4. Feature engineering

This step involves the art and science of transforming raw data into features that better represent a pattern to the learning algorithms.

5. Splitting data into training and test sets

The final step is to split your data into two sets; one for training your algorithm, and another for evaluation purposes.

1. Data Exploration and Profiling

Once the data is collected, it's time to assess the condition of it, including looking for trends, outliers, exceptions, incorrect, inconsistent, missing, or skewed information. This is important because your source data will inform all of your model's findings, so it is critical to be sure it does not contain unseen biases. For example, if you are looking at customer behavior nationally, but only pulling in data from a limited sample, you might miss important geographic regions. This is the time to catch any issues that could incorrectly skew your model's findings, on the entire data set, and not just on partial or sample data sets.

01

Incorrect and inconsistent data discovery

This is the time to catch any issues that could incorrectly skew your model's findings, on the entire data set

02

Discover unseen bias in data

If you are looking at customer behavior nationally, but only pulling in data from a limited sample, you might miss important geographic regions.

03

Data visualization tools and techniques

Use any data visualization tools and techniques or statistical analysis techniques to explore data

2. Formatting data to make it consistent

The next step in great data preparation is to ensure your data is formatted in a way that best fits your machine learning model. If you are aggregating data from different sources, or if your data set has been manually updated by more than one stakeholder, you'll likely discover anomalies in how the data is formatted (e.g. USD5.50 versus $5.50). In the same way, standardizing values in a column, e.g. State names that could be spelled out or abbreviated) will ensure that your data will aggregate correctly. Consistent data formatting takes away these errors so that the entire data set uses the same input formatting protocols.

01

Anomalies in how the data is formatted

$50 vs USD 50

02

Formatting inconsistency in unstructured data

Two different words **length** and **size** of array results in different vectors.

03

Inconsistency checking

Inconsistent values from different sources of data

3. Improving data quality

Start by having a strategy for dealing with erroneous data, missing values, extreme values, and outliers in your data. Self-service data preparation tools can help if they have intelligent facilities built in to help match data attributes from disparate datasets to combine them intelligently. For instance, if you have columns for FIRST NAME and LAST NAME in one dataset and another dataset has a column called CUSTOMER that seem to hold a FIRST and LAST NAME combined, intelligent algorithms should be able to determine a way to match these and join the datasets to get a singular view of the customer.
For continuous variables, make sure to use histograms to review the distribution of your data and reduce the skewness. Be sure to examine records outside an accepted range of value. This “outlier” could be an inputting error, or it could be a real and meaningful result that could inform future events as duplicate or similar values could carry the same information and should be eliminated. Similarly, take care before automatically deleting all records with a missing value, as too many deletions could skew your data set to no longer reflect real-world situations.

01

Handling missing data

This is the time to catch any issues that could incorrectly skew your model’s findings, on the entire data set

02

Outlier removal

If you are looking at customer behavior nationally, but only pulling in data from a limited sample, you might miss important geographic regions.

03

Unbalanced data handling

Use any data visualization tools and techniques or statistical analysis techniques to explore data

Handling missing data: notes

01

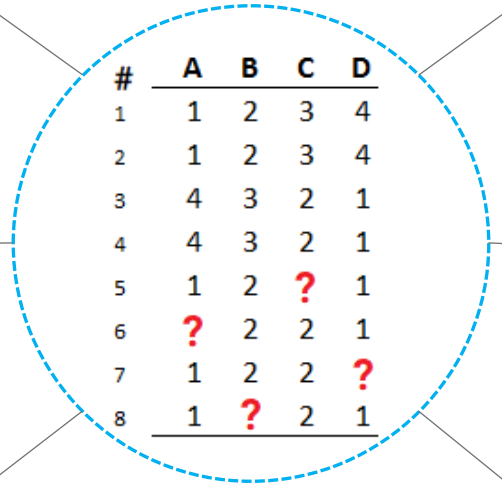
Deleting rows
Delete the rows with at leas one missing value

02

Deleting Columns
Delete the columns with at leas one missing value

03

Replacing with mean/median/mode



#	A	B	C	D
1	1	2	3	4
2	1	2	3	4
3	4	3	2	1
4	4	3	2	1
5	1	2	?	1
6	?	2	2	1
7	1	2	2	?
8	1	?	2	1

04

Assigning a unique category such as ‘unknown’

05

Predict the missing values
With the help of machine learning techniques such as regression on other features

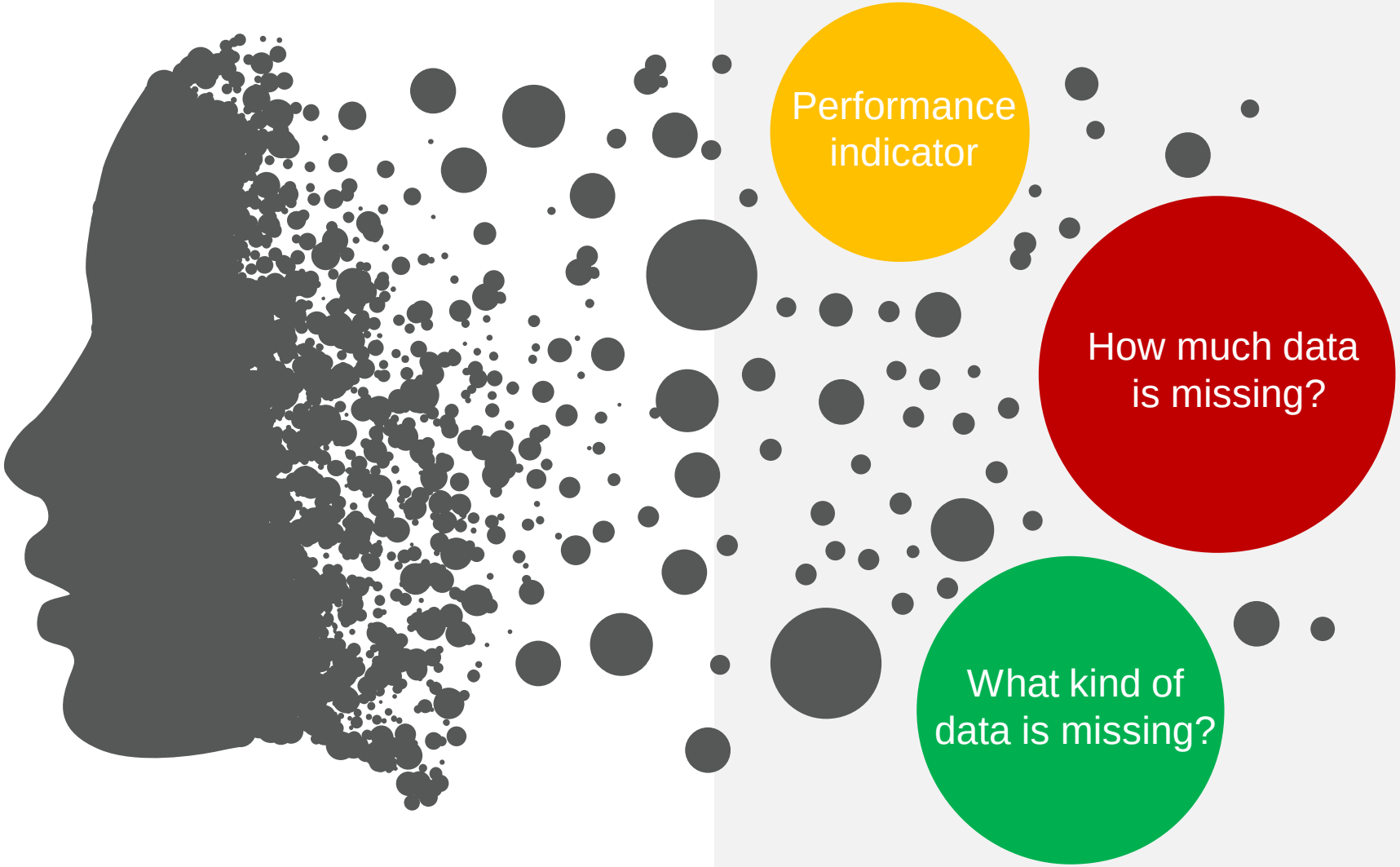
06

Using algorithms which support missing values
Like KNN, Random Forest

TECHNIQUES

Different techniques in handling missing data in machine learning research

Handling missing data

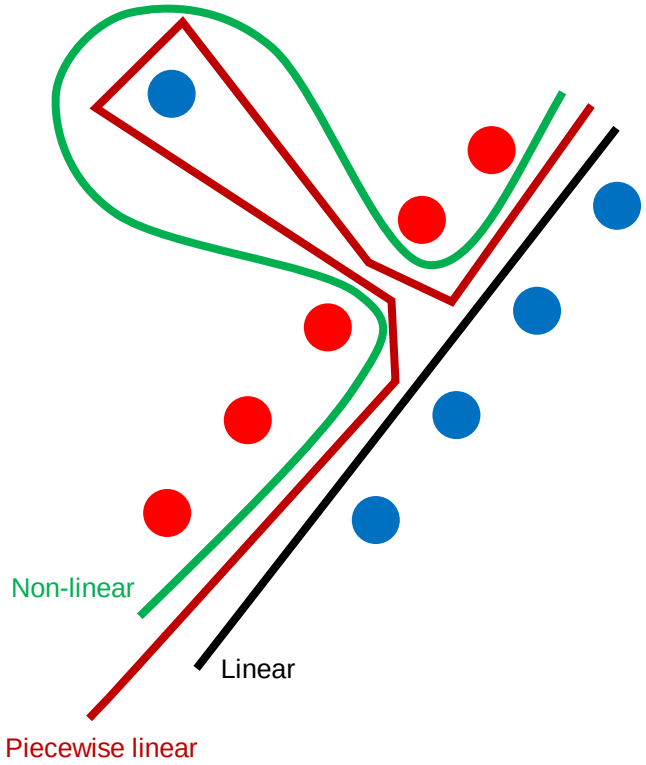


Outlier removal

What are anomalies/outliers?
The set of data points that are considerably different than the remainder of the data

Applications
Credit card fraud detection, telecommunication, fraud detection, network intrusion detection, fault detection

Anomaly Detection Schemes
General Steps: 1) Build a profile of the “normal” behavior. Profile can be patterns or summary statistics for the overall population. 2) Use the “normal” profile to detect anomalies

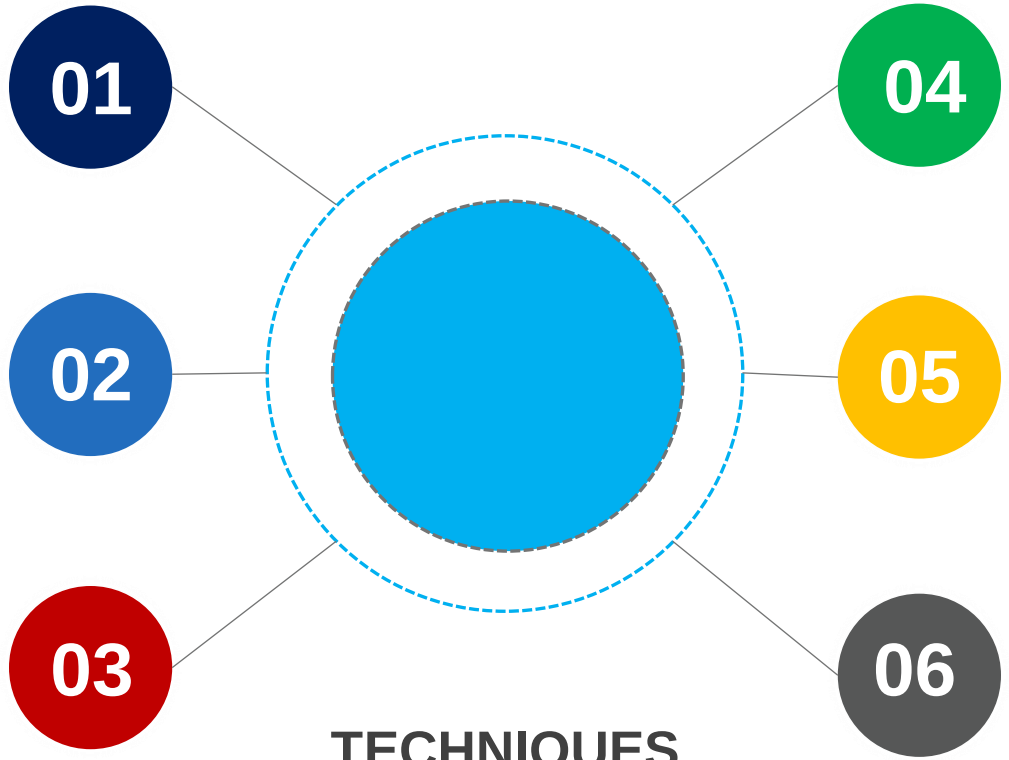


Outlier removal

Graphical Approaches
Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D), Chernoff faces , …
Limitations: Time consuming

Convex Hull Method
Extreme points are assumed to be outliers. Use convex hull method to detect extreme values. What if the outlier occurs in the middle of the data?

Statistical Approaches
Assume a parametric model describing the distribution of the data (e.g., normal distribution). Apply a statistical test that depends on data distribution



TECHNIQUES

Different techniques in outlier removal in machine learning research

Distance-Based Methods

Clustering-Based
Cluster the data into groups of different density. Choose points in small cluster as candidate outliers. Compute the distance between candidate points and non-candidate clusters. If candidate points are far from all other non-candidate points, they are outliers

Nearest-Neighbor Based Approach
Compute the distance between every pair of data points. Data points for which there are fewer than p neighboring points within a distance are outliers.

Density-Based
For each point, compute the density of its local neighborhood. Outliers are points with largest difference in average local density of sample p and the density of its nearest neighbors

Outlier removal : notes

How many outliers
The number of outliers in dataset.

Method is unsupervised
Method is fully unsupervised. Outliers comes from data and may lead algorithm in wrong directions.

Validation can be quite challenging
In conjunction with next steps algorithms

Outlier or rare distribution of data?

Working assumption
There are considerably more “normal” observations than “abnormal” observations

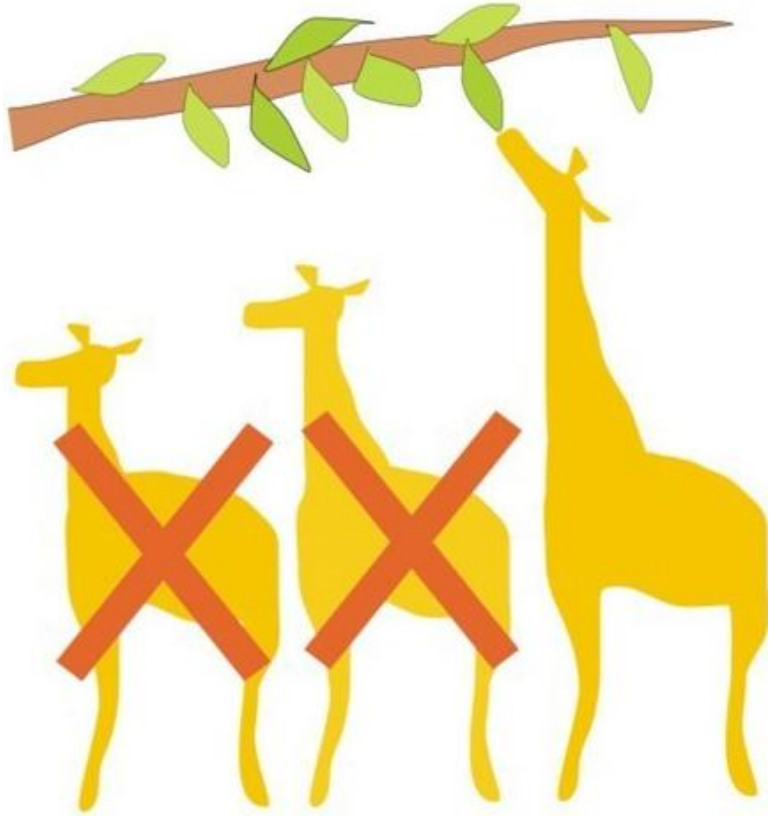
4. Feature Engineering

Feature engineering is the process of using domain knowledge of the data to **select** features or **generate** new features that allow machine learning algorithms to work more accurately.

Coming up with features is difficult, time-consuming, requires expert knowledge.

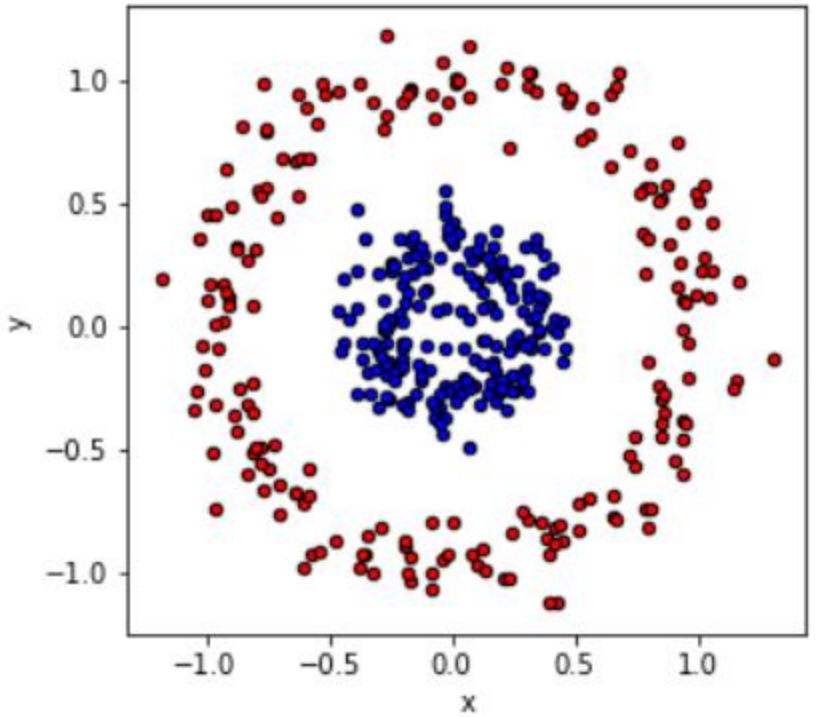
Some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.

Good data preparation and feature engineering is integral to better prediction



4. Feature Engineering

Original space

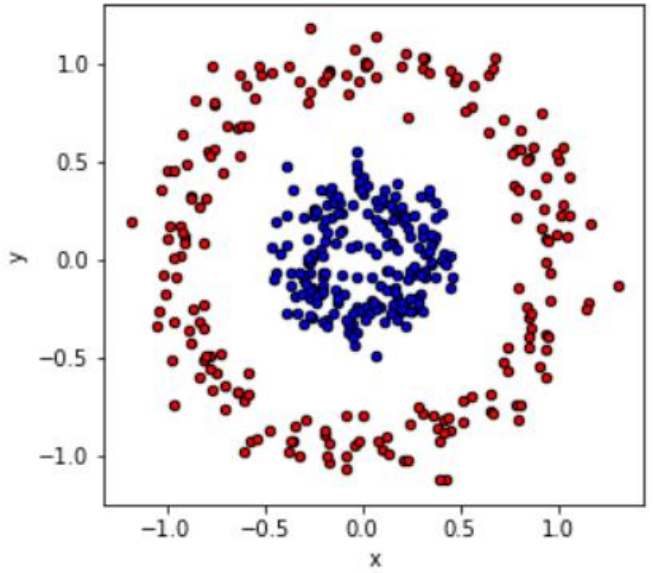


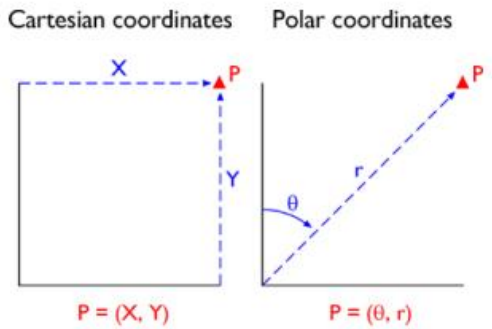
Not possible to separate using linear classifier

4. Feature Engineering

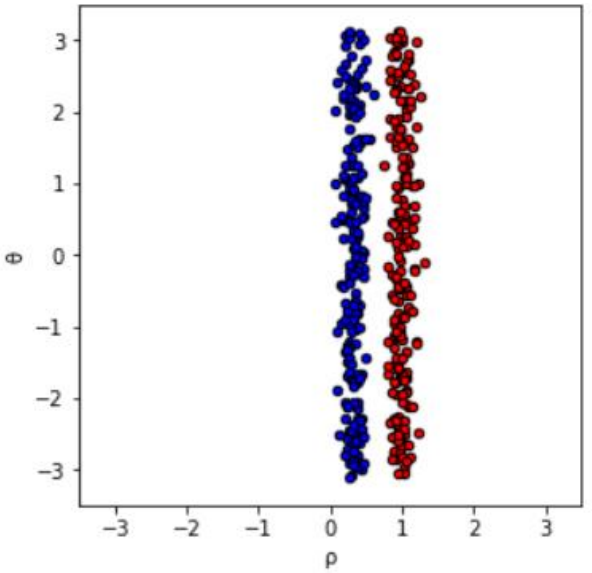
What if we use polar coordinates instead?

Original space





Polar coordinates

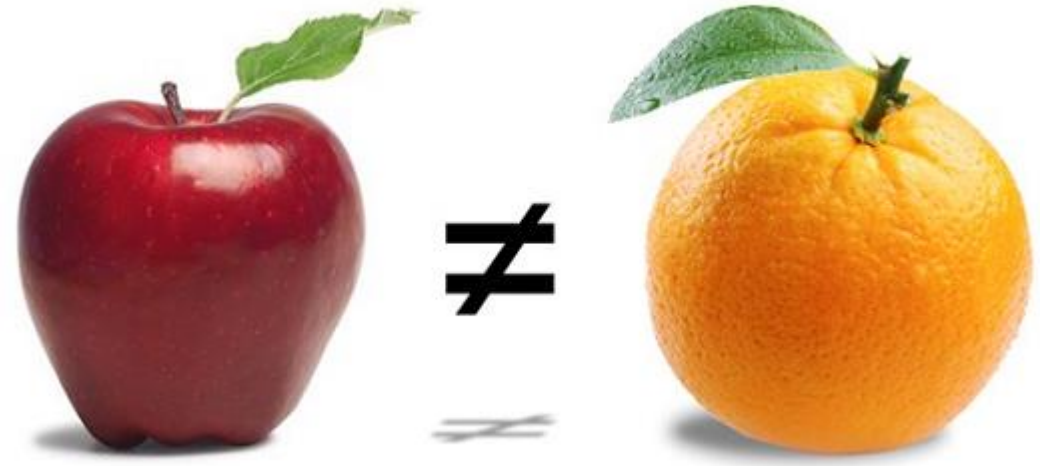


Feature Generation

Feature generation is also known as feature construction or feature extraction. The process of creating new features from raw data or one or multiple features.

Three general methodologies:

1. **Feature extraction**: Domain-specific, Statistical features, predefined class of features
2. **Mapping data to new space:** E.g., Fourier transformation: wavelet transformation, manifold approaches
3. **Feature construction**: Combining features, Data discretization



Feature Selection

Sometimes a lot of features exist. Feature selection is the method of reducing data dimension while doing predictive analysis.

Why:

- Curse of dimensionality
- Easier visualization
- Meaningful distance
- Lower storage need
- Faster computation

Two kinds of feature selection techniques

- Filter Method
- Wrapper Method

Feature Selection

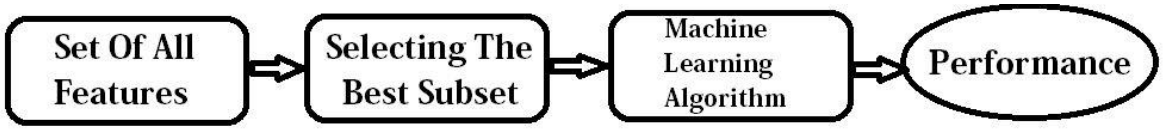
Filter Method

Filter Method
This method uses the variable ranking technique in order to select the variables for ordering and here, the selection of features is independent of the classifiers used. By ranking, it means how much useful and important each feature is expected to be for classification. Removes Irrelevant and redundant features.

Statistical Test: Chi-Square Test to test the independence of two events, Hypothesis testing.

Variance Threshold: This approach of feature selection removes all features whose variance does not meet some threshold. Generally, it removes all the zero-variance features which means all the features that have the same value in all samples.

Information Gain: Information gain measures how much information a feature gives about the class.



$$\mathbf{X} = [\mathbf{x}_1, \cancel{\mathbf{x}_2}, \mathbf{x}_3, \cancel{\mathbf{x}_4}]$$

Feature Selection

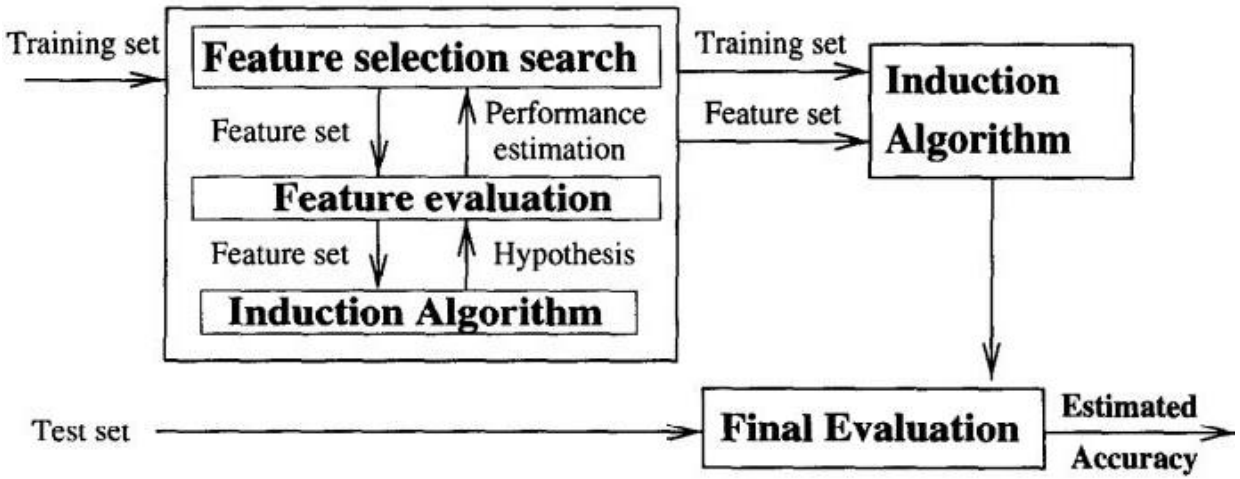
Wrapper Method

In the wrapper approach, the feature subset selection algorithm exists as a wrapper around the induction algorithm. One of the main drawbacks of this technique is the mass of computations required to obtain the feature subset. Some examples of Wrapper Methods are mentioned below:

Genetic Algorithms: This algorithm can be used to find a subset of features.

Recursive Feature Elimination: RFE is a feature selection method which fits a model and removes the weakest feature until the specified number of features is satisfied.

Sequential Feature Selection: This naive algorithm starts with a null set and then add one feature to the first step which depicts the highest value for the objective function and from the second step onwards the remaining features are added individually to the current subset and thus the new subset is evaluated. This process is repeated until the required number of features are added.

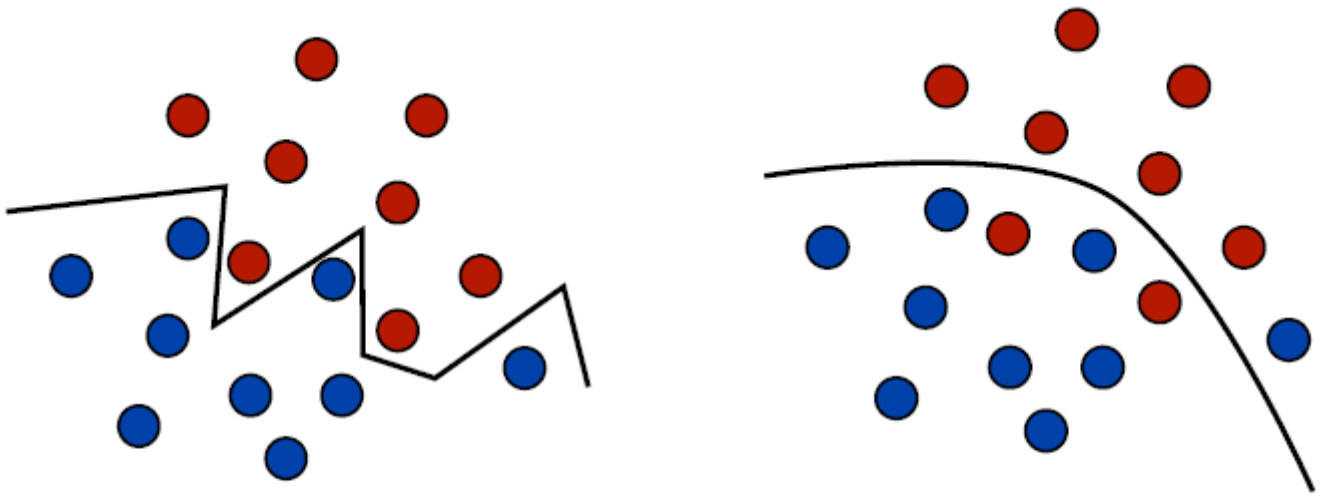


start: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4]$

$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4]$

stop: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_3]$

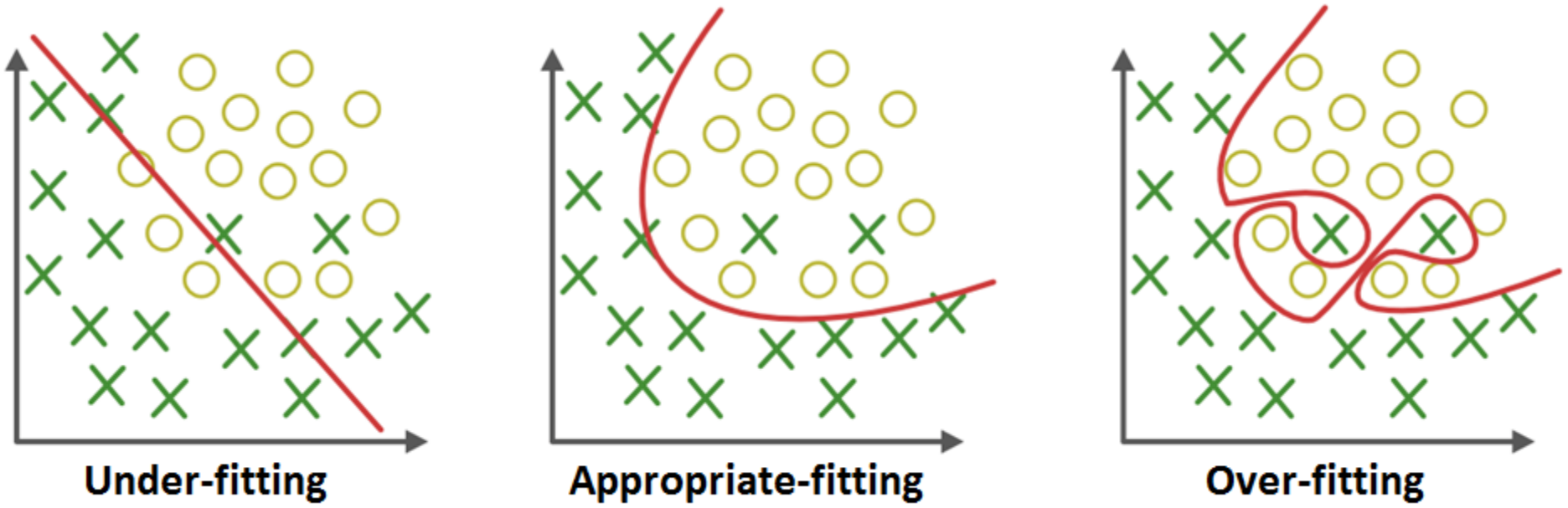
Learning ≠ Fitting



Notion of simplicity/complexity.

→ How do we define complexity?

Overfitting and Underfitting



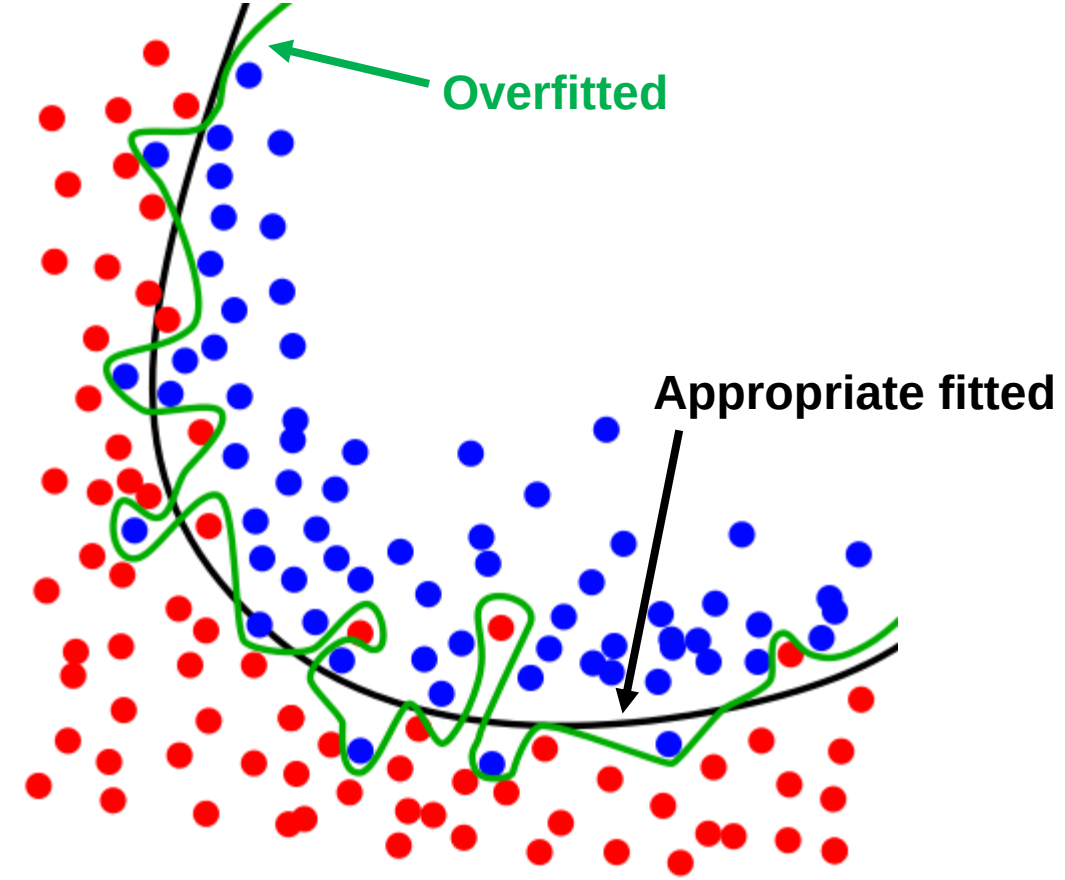
Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.

Underfitting occurs when a model cannot adequately capture the underlying structure of the data.

Overfitting and Underfitting

Overfitting can have many causes and usually is a combination of the following:

- **Too powerful model**: e.g. you allow polynomials to degree 100. With polynomials to degree 5 you would have a much less powerful model which is much less prone to overfitting
- **Not enough data**: Getting more data can sometimes fix overfitting problems
- **Too many features**: Your model can identify single data points by single features and build a special case just for a single data point. For example, think of a classification problem and a decision tree. If you have feature vectors (x_1, x_2, ..., x_n) with binary features and n points, and each feature vector has exactly one 1, then the tree can simply use this as an identifier.

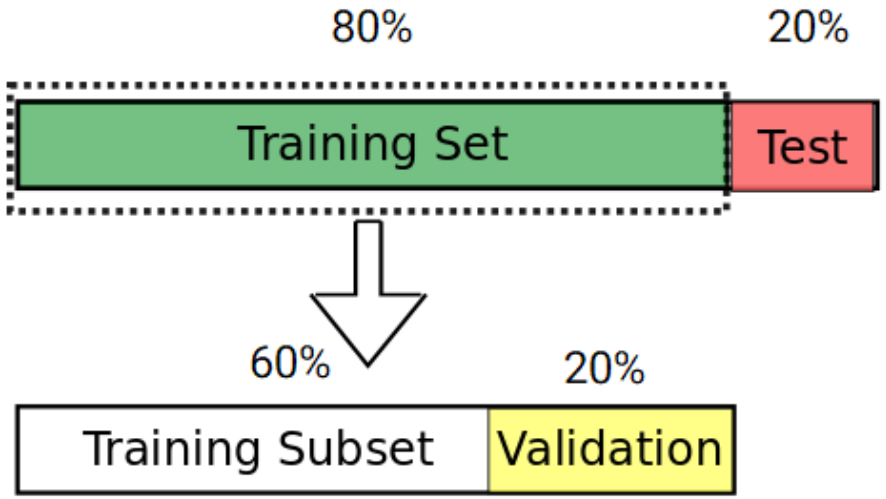


5. Splitting Data

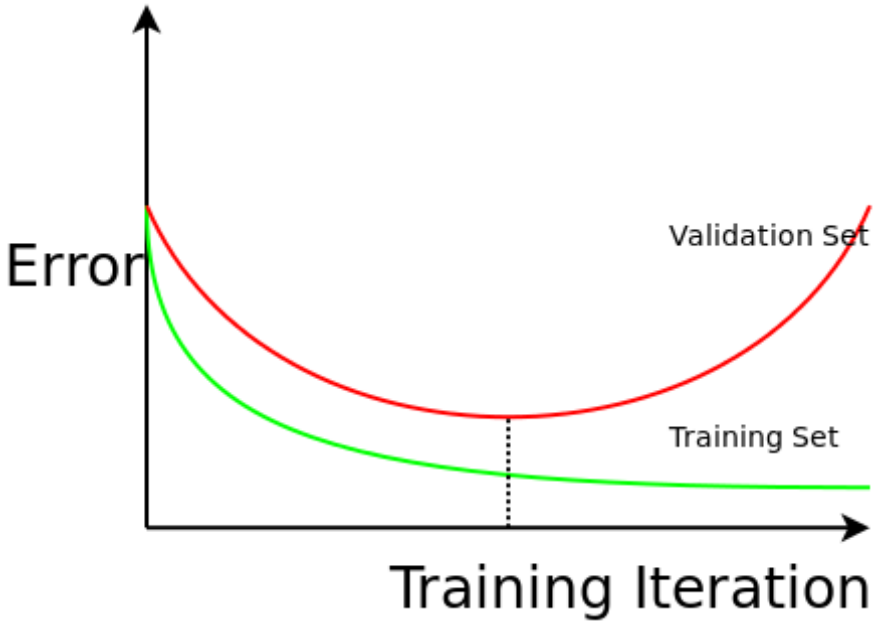
Training set: A set of examples used for learning, that is to fit the parameters [i.e., weights] of the classifier.

Validation set: A set of examples used to avoid overfitting or tune the hyperparameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.

Test set: A set of examples used only to assess the performance [generalization] of a fully specified classifier.



5. Splitting Data

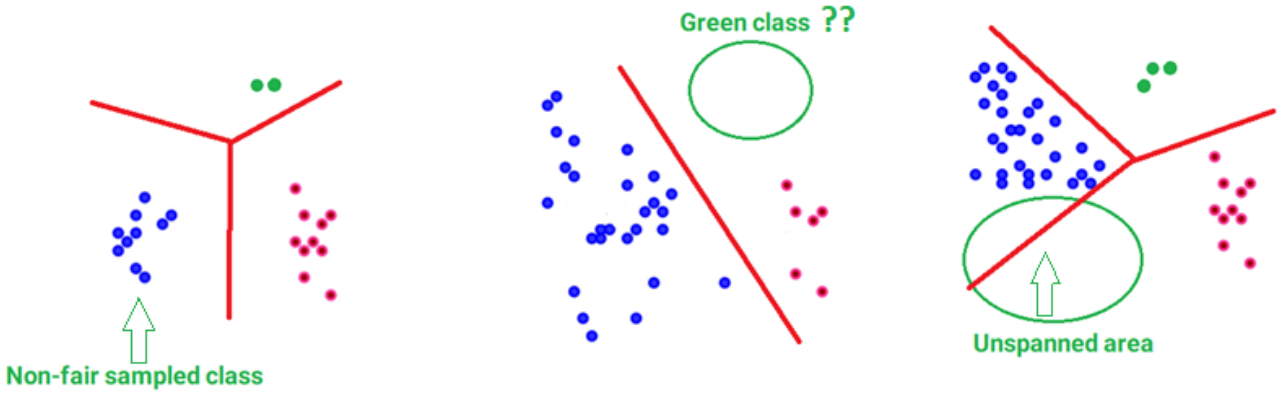


If you plot the training error, and validation error, against the number of training iterations you get the most important graph in machine learning.

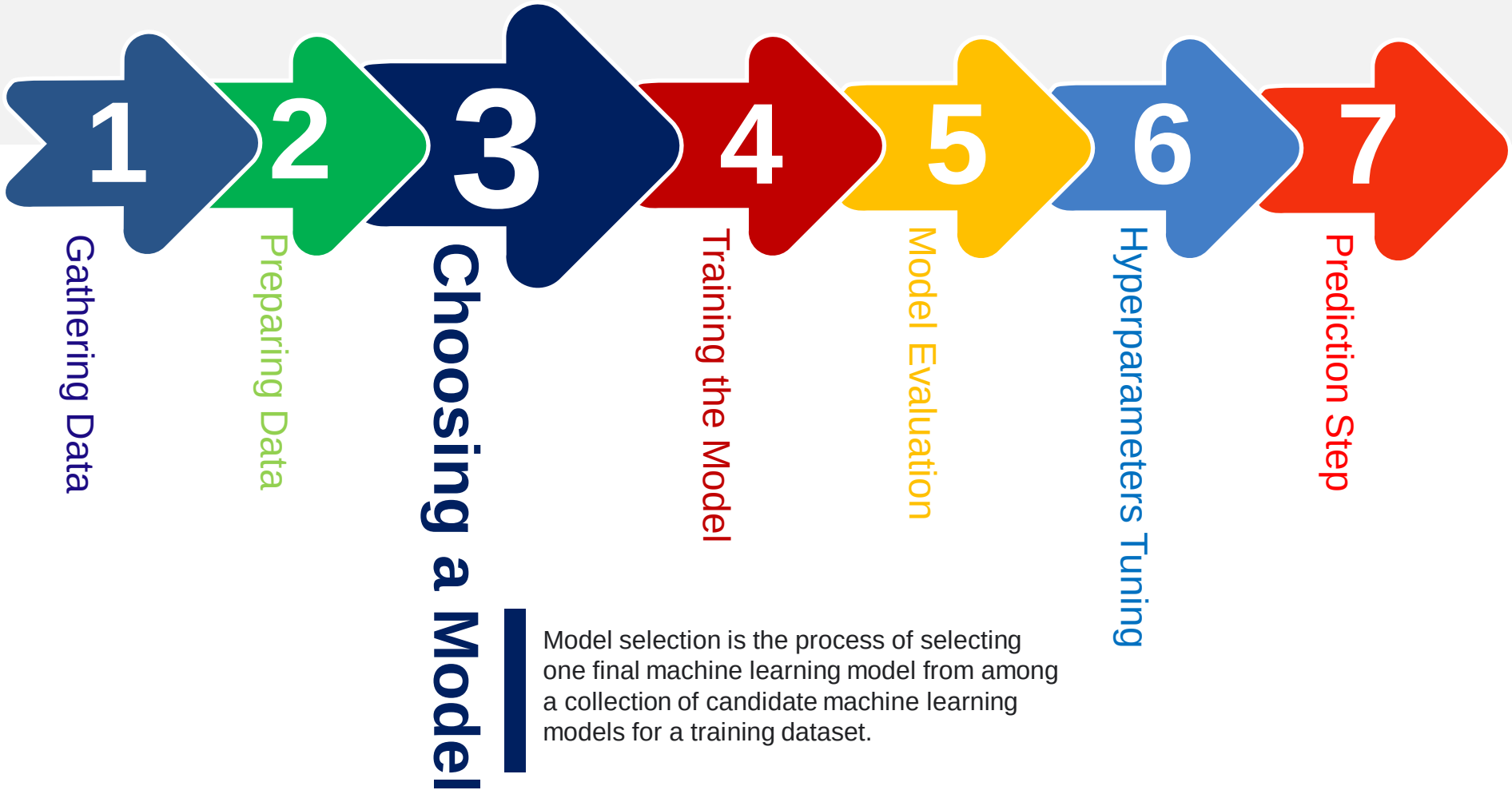
5. Splitting Data: Tips

1. Non-fair sampling
2. Class imbalance problem
3. Unspanned area

Inappropriate sampling may results in a wrong classification of data as the figure shows



7 Steps of Machine Learning



Model selection is the process of selecting one final machine learning model from among a collection of candidate machine learning models for a training dataset.

7 Steps of Machine Learning

3

Choosing a Model

Some Broad ML Tasks:

Classification: assign a category to each item (e.g., document classification).

Regression: predict a real value for each item (prediction of stock values, economic variables).

Clustering: partition data into ‘homogenous’ regions (analysis of very large data sets).

Dimensionality reduction: find lower-dimensional manifold preserving some properties of the data.

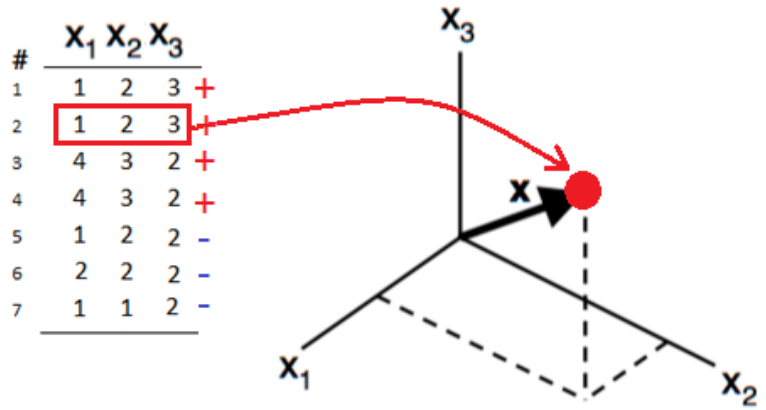
Semi-supervised learning: use existing side information during learning

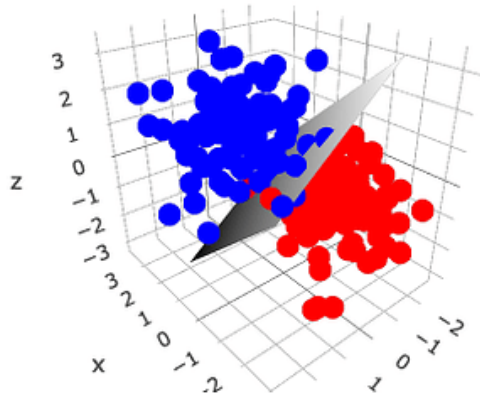
Classification

Different techniques:

- **Decision Tree Induction**
- **Random Forest**
- **Bayesian Classification**
- **K-Nearest Neighbor**
- **Neural Networks**
- **Support Vector Machines**
- **Hidden Markov Models**
- **Rule-based Classifiers**
- **Many More ….**
- **Also Ensemble Methods**

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{matrix} \text{Age} \\ \text{Weight} \\ \text{Pressure} \end{matrix}$$





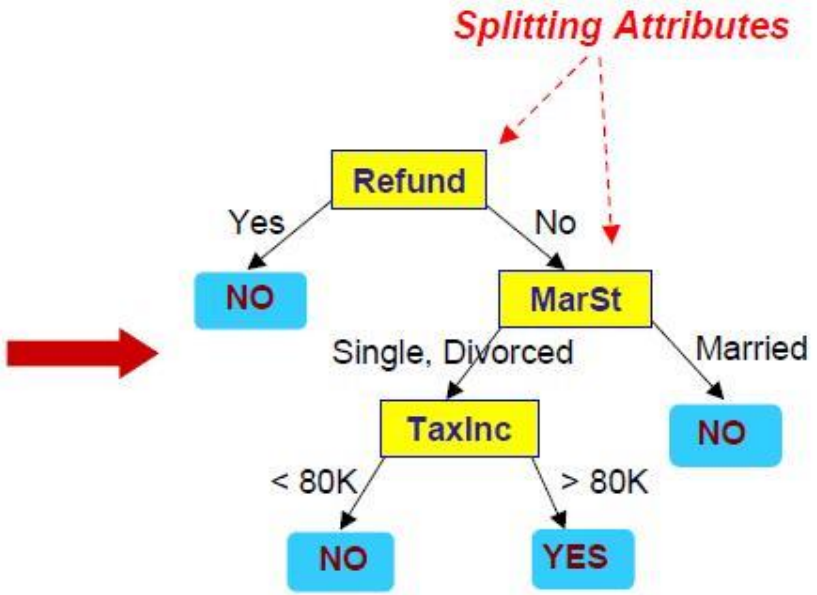
Feature vector **Feature space (3D)**

Decision Tree

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves).

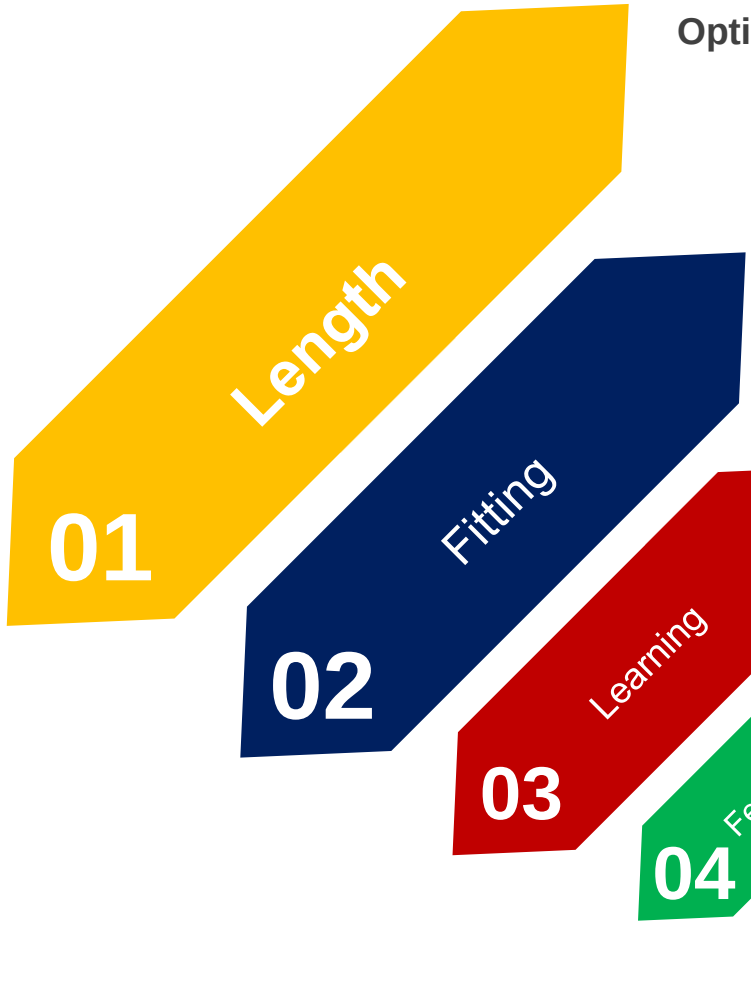
	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

DT: Notes



Optimal tree length

Over-fitting and under-fitting
Pruning and growing as solutions

Incremental learning

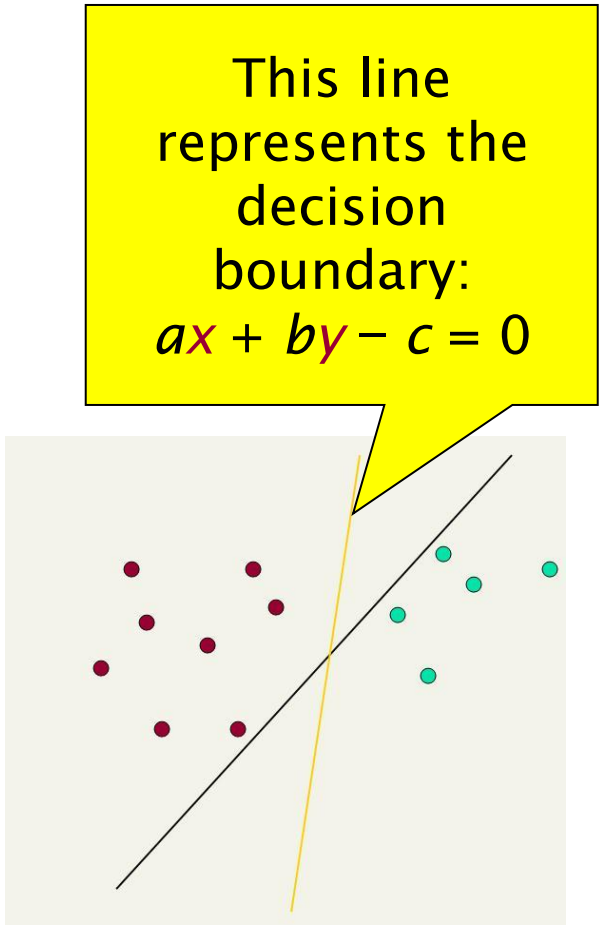
Handling numerical features

Purity measures
Entropy, information gain, Purity

SVM: the story

Lots of possible solutions for a, b, c.

- Some methods find a separating hyper plane, but not the optimal one [according to some criterion of expected goodness].
- SVM (A.k.a. large margin classifier) maximizes the margin around the separating hyperplane.
- Solving SVMs is a quadratic programming problem



SVM: theory

- Constrained optimization:

$$\min_{\mathbf{w},b,\xi} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

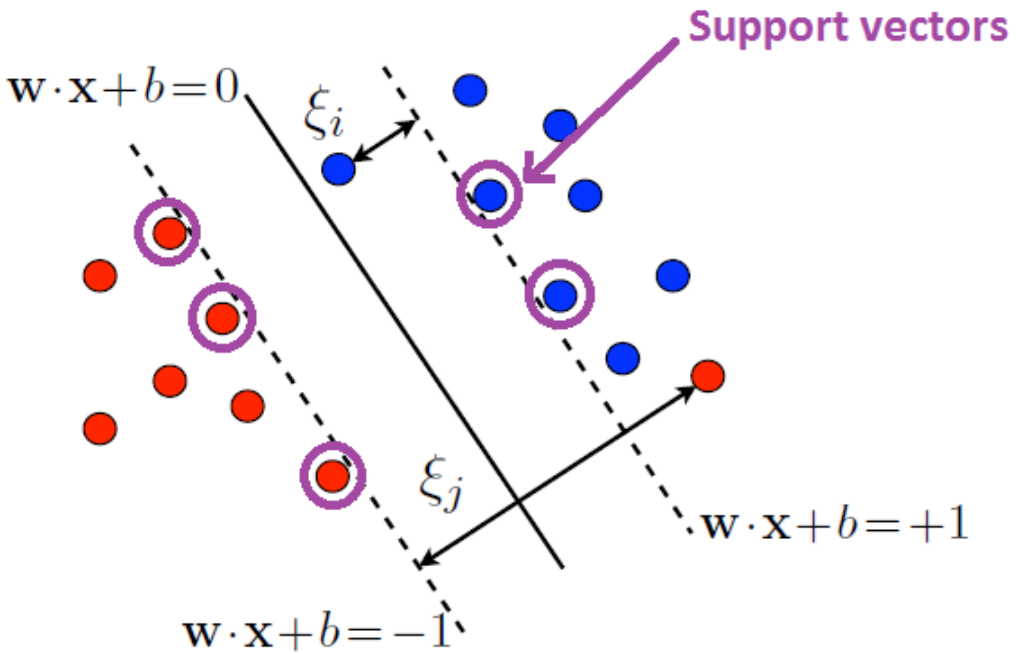
$$\text{subject to} \ \ y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \ \ \wedge \ \ \xi_i \geq 0, i \in [1, m].$$

- Properties:
  - $C \geq 0$ trade-off parameter.
  - Convex optimization.
  - Unique solution.

- Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)$$

$$\text{subject to: } \alpha_i \geq 0 \wedge \sum_{i=1}^{m}\alpha_i y_i = 0, i \in [1, m].$$



The decision function is fully specified by a subset of training samples, the support vectors.

SVM

Notes

Parameter C

Trade-off between maximizing margin and minimizing training error. How do we determine C?

Non-linear classification using Kernels

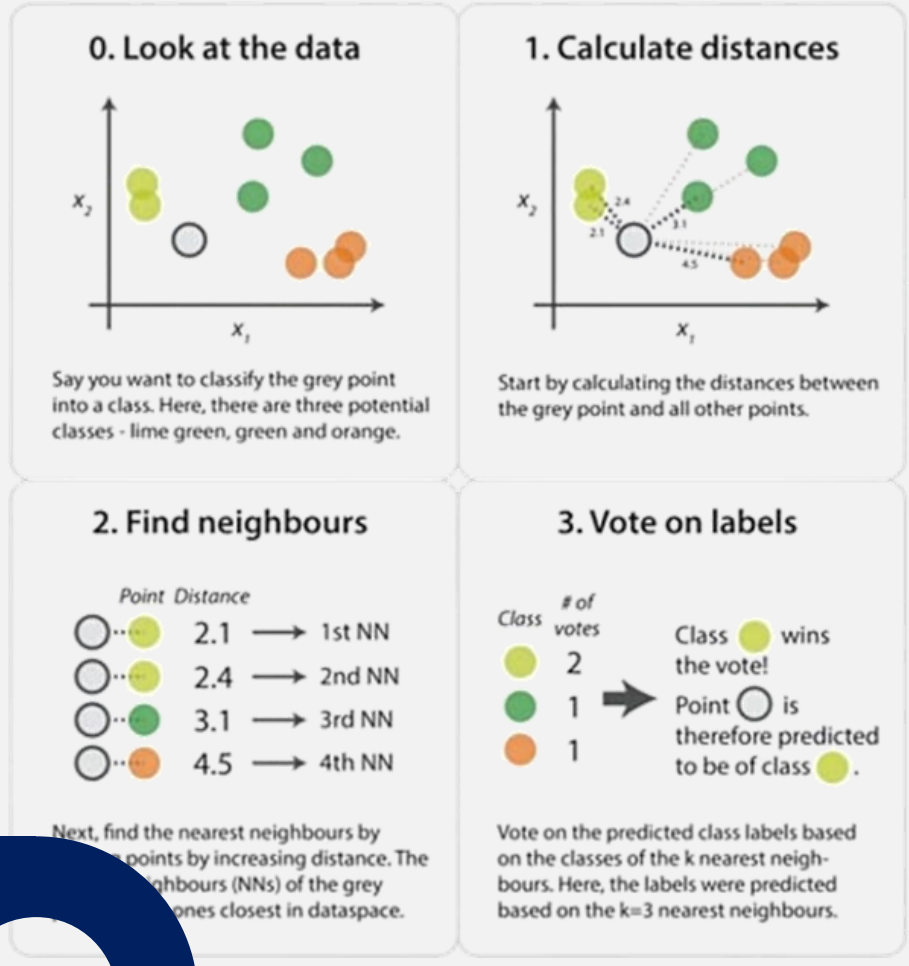
Different kernel types, kernel parameters

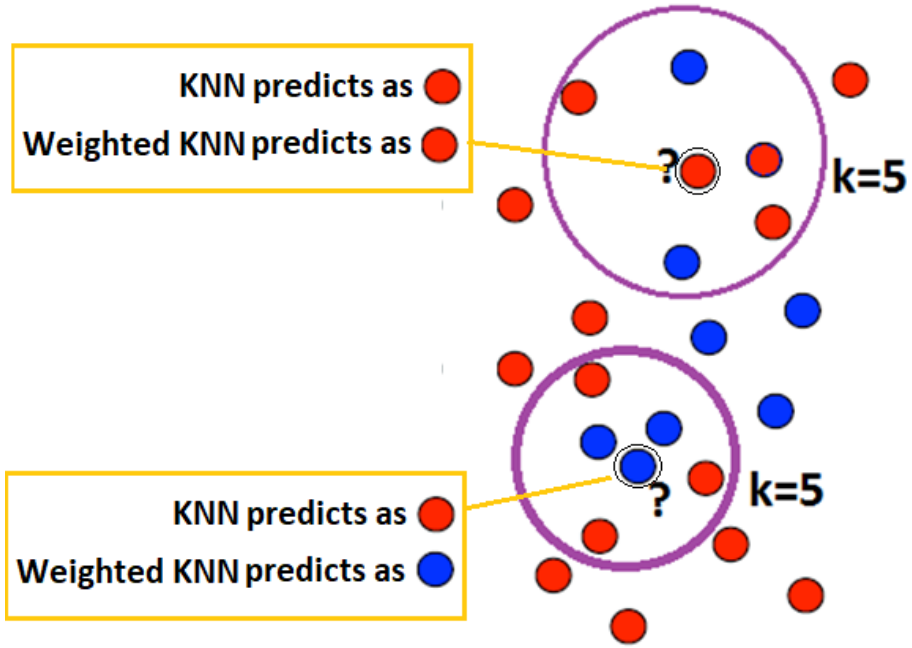
Non-vectorial inputs

The dot product of data is enough to train SVM, dimensionality of data is not an issue.

Difficult to handle missing values, Robust to noise

KNN


0. Look at the data
x_2
x_1
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.
1. Calculate distances
x_2
x_1
Start by calculating the distances between the grey point and all other points.
2. Find neighbours
Point Distance
2.1 → 1st NN
2.4 → 2nd NN
3.1 → 3rd NN
4.5 → 4th NN
Next, find the nearest neighbours by
points by increasing distance. The
hbours (NNs) of the grey
ones closest in dataspace.
3. Vote on labels
Class # of votes
2
1
1
Class wins the vote!
Point is therefore predicted to be of class .
Vote on the predicted class labels based on the classes of the k nearest neigh-bours. Here, the labels were predicted based on the k=3 nearest neighbours.


KNN predicts as
Weighted KNN predicts as
?
k=5
KNN predicts as
Weighted KNN predicts as
?
k=5

KNN: Notes

Weights of neighbors

High computation complexity
Partial distance calculation as a solution

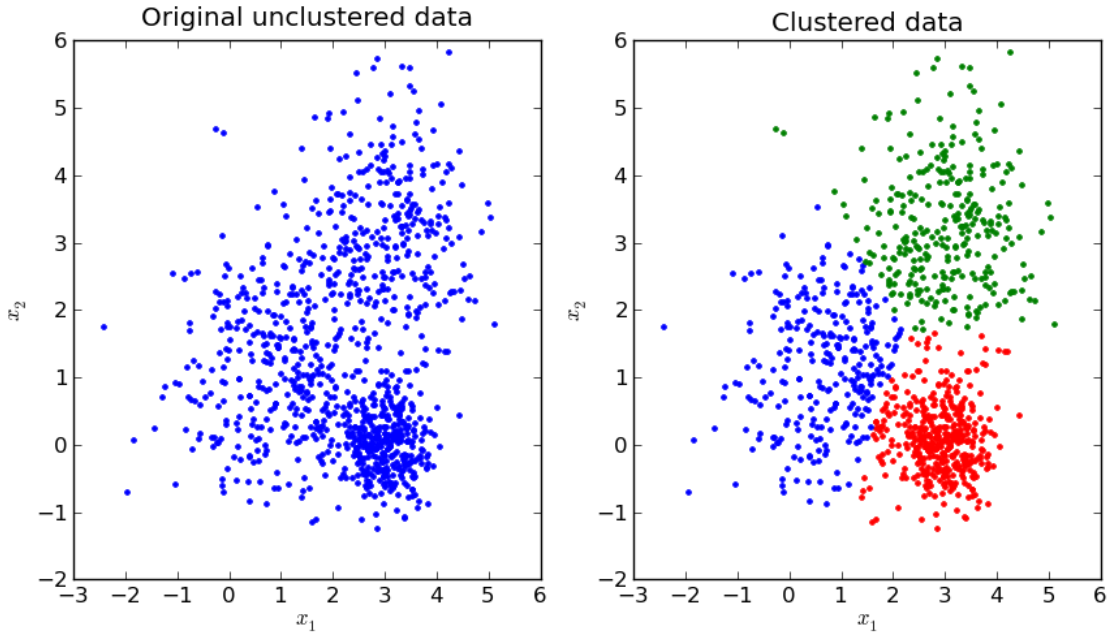
High storage complexity
Delaunay triangulation as a solution

Optimal value of K

The issue of irrelevant features
Irrelevant features easily swamp information from relevant attributes. Sensitive to distance function

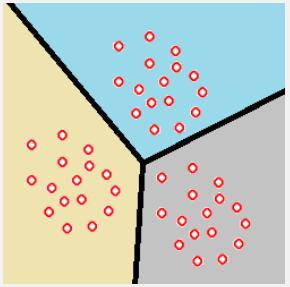
Clustering

Clustering is a method of unsupervised learning and is a common technique for statistical **data** analysis used in many fields.
In **Data** Science, we can use **clustering** analysis to gain some valuable insights from our **data** by seeing what groups the **data** points fall into when we apply a **clustering** algorithm.

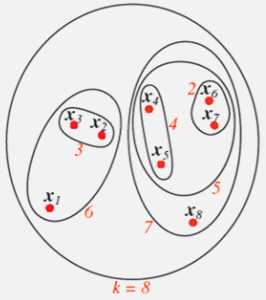


Clustering: Different Techniques (1)

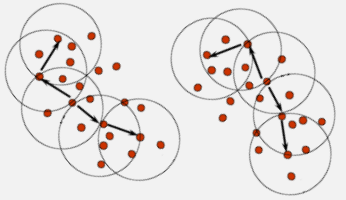
- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids



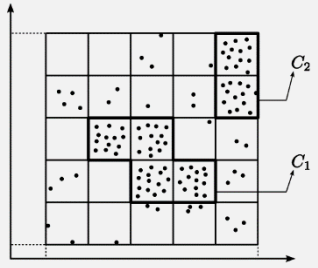
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH



- Density-based approach:
  - Based on connectivity and density functions. Typical methods: DBSACN

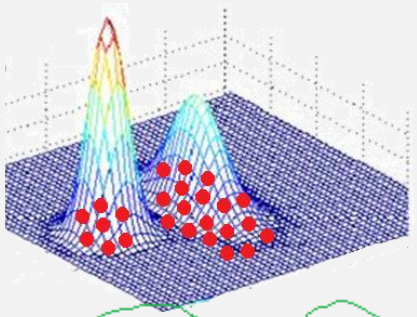


- Grid-based approach:
  - Based on a multiple-level granularity structure. Typical methods: STING

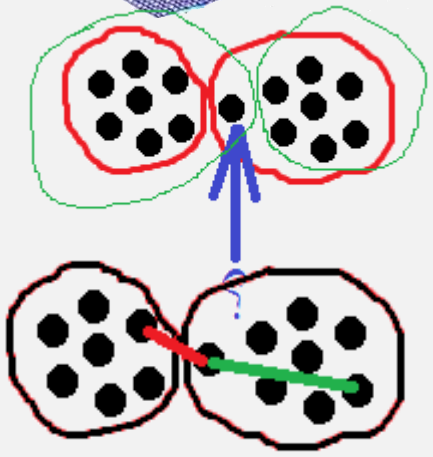


Clustering: Different Techniques (2)

- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other. Typical methods: EM, SOM



- Fuzzy Clustering:
  - Based on the theory of fuzzy membership. Typical methods: C-Means



- Constraint-based:
  - Clustering by considering user-specified or application-specific constraints. Typical methods: constrained clustering

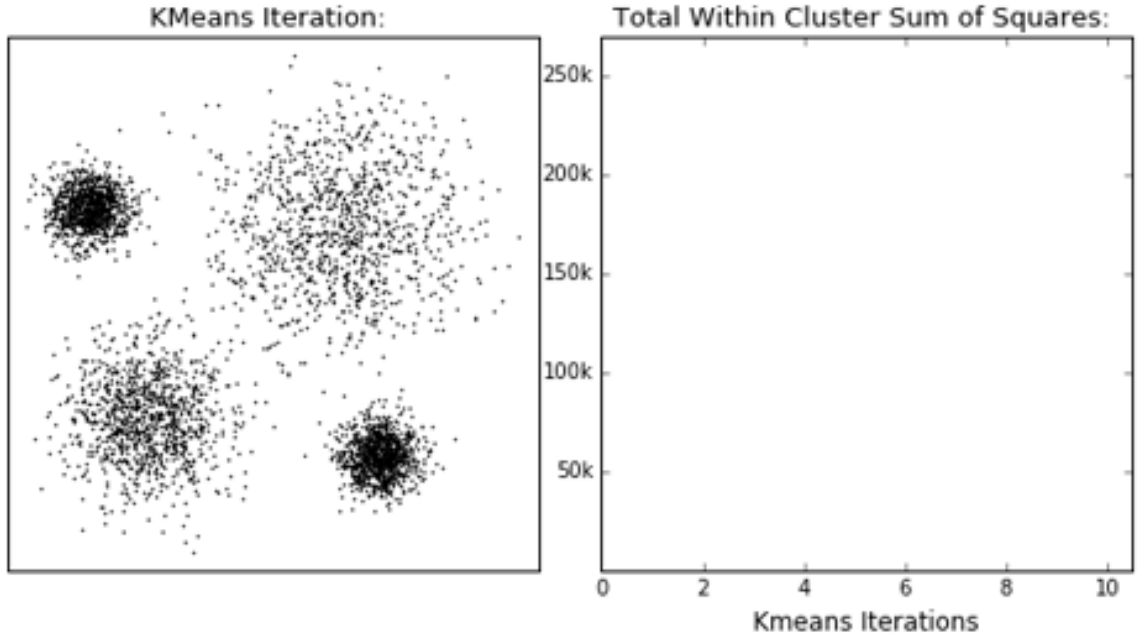
- Graph-based clustering:
  - Use the theory of graph to group data in clusters. Typical methods: Graph-Cut

- Subspace Clustering
- Spectral Clustering
- Consensus Clustering

K-Means

Given k, the **k-means** algorithm works as follows

1. Randomly choose k data points (seeds) to be the initial centroids, cluster centers
2. Assign each data point to the closest centroid
3. Re-compute the centroids using the current cluster memberships.
4. If a convergence criterion is not met, go to 2.



K-Means

Strengths:

- Simple: easy to understand and to implement
- Efficient: Time complexity: O(tkdn),
  where n is the number of data points,
  k is the number of clusters,
  d is dimension of data, and
  t is the number of iterations.
  Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

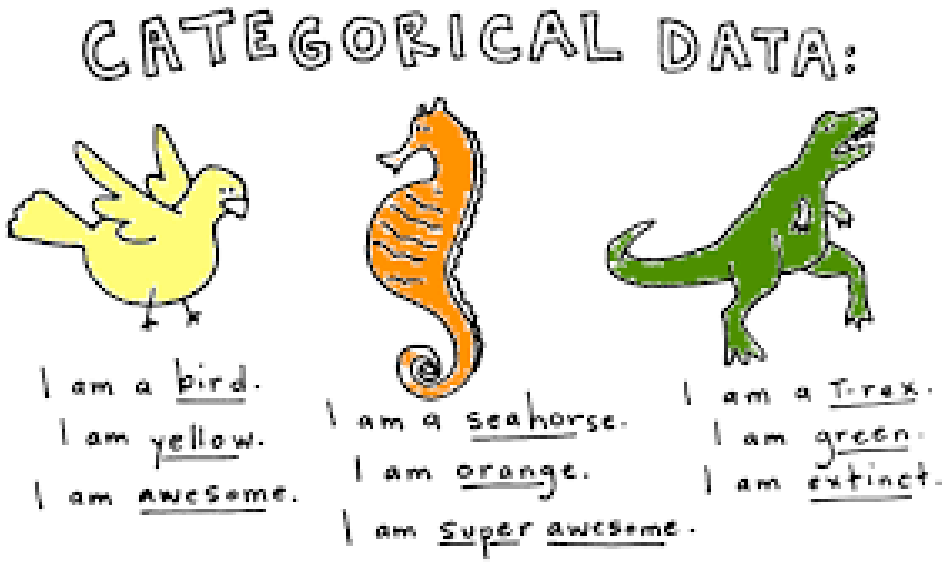
K-Means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, k-modes - the centroid is represented by most frequent values.
  - E.g. if feature vector = (eye color, graduation level)
- The user needs to specify k.
- The algorithm is sensitive to initial seeds.
- The different number of points in each cluster can lead to large clusters being split ‘unnaturally’
- The k-means algorithm is not suitable for discovering clusters with different shapes.
- The algorithm is sensitive to outliers
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

K-Means

The algorithm is only applicable if the mean is defined.

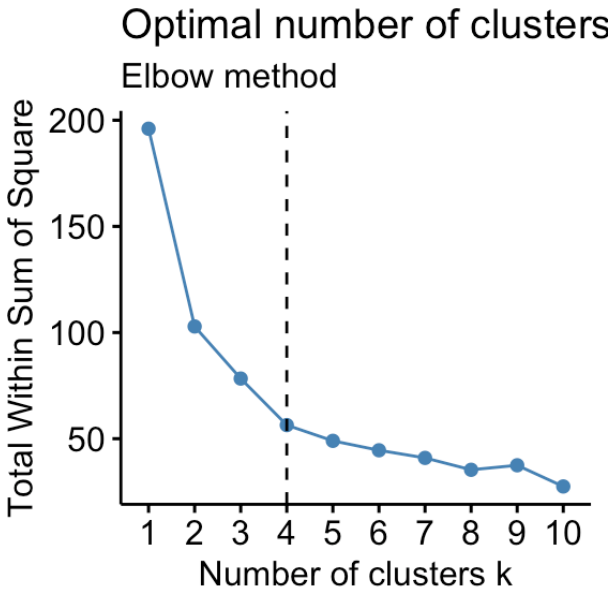
- For categorical data, k-modes - the centroid is represented by most frequent values.
- E.g. if feature vector = (eye color, graduation level)



K-Means

The user needs to specify *k*.

- Use clustering stability techniques to estimate *k*.
- Use elbow method to estimate *k*

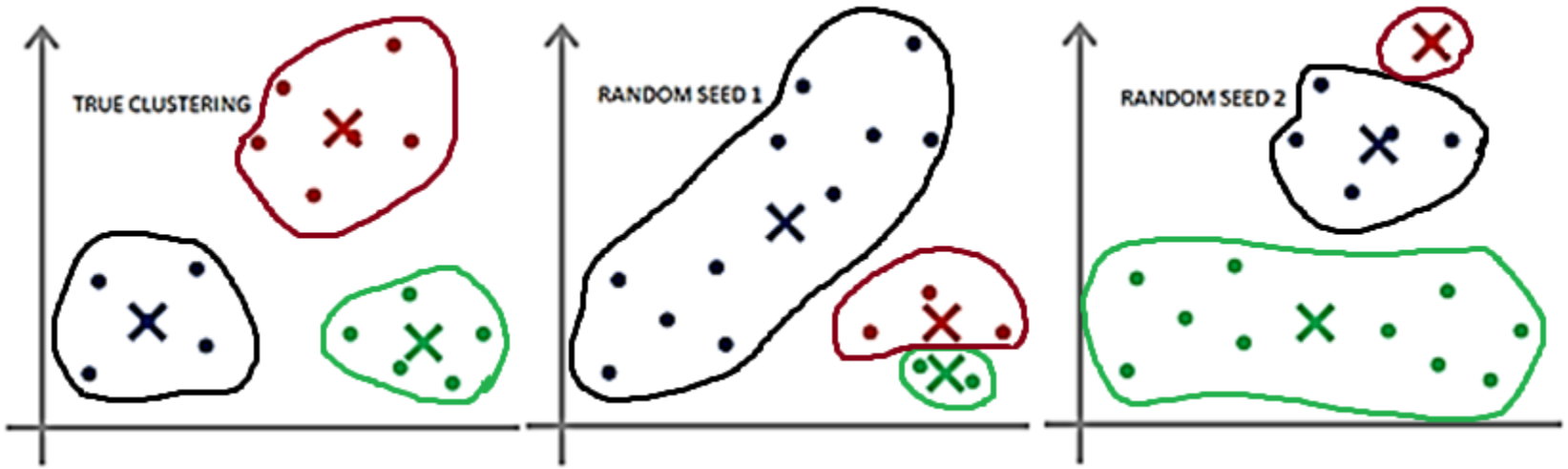


elbow method

K-Means

The algorithm is sensitive to initial seeds.

- Multiple runs, K-means++



K-Means

The Different number of points in each cluster can lead to large clusters being split ‘unnaturally’

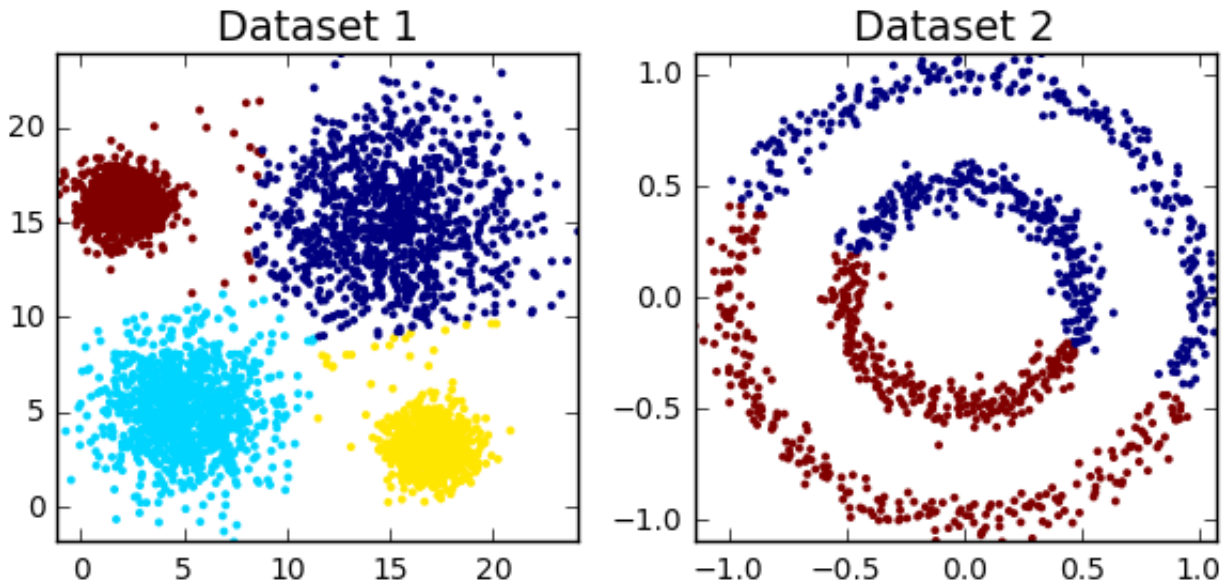
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

$J_e = large$

$J_e = small$

K-Means

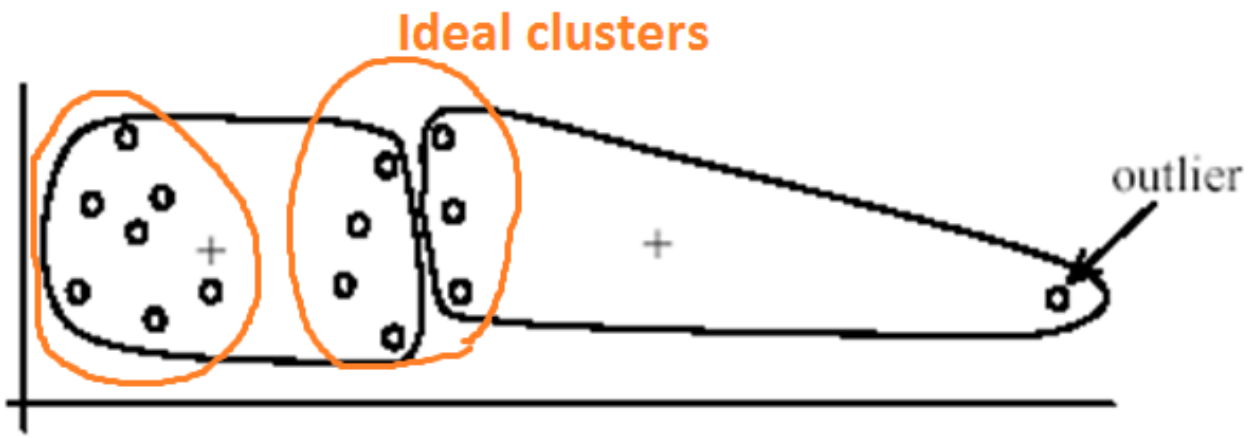
The *k*-means algorithm is not suitable for discovering clusters with different shapes.



K-Means

The algorithm is sensitive to outliers

- Outliers are data points that are very far away from other data points.
- Outliers could be errors in the data recording or some special data points with very different values.



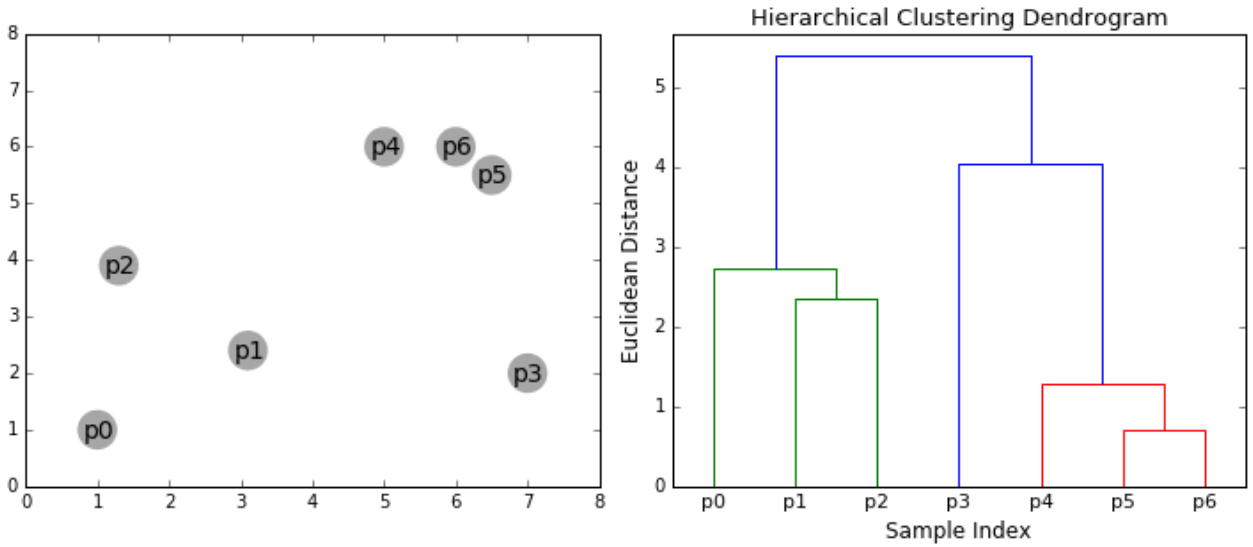
Hierarchical Clustering

Unlike k-means, hierarchical clustering (HC) doesn't require the user to specify the number of clusters beforehand. Instead it returns an output (typically as a dendrogram), from which the user can decide the appropriate number of clusters.

HC typically comes in two flavors (essentially, bottom up or top down):

Agglomerative: The agglomerative method in reverse- individual points are iteratively combined until all points belong to the same cluster.

Divisive: Starts with the entire dataset comprising one cluster that is iteratively split- one point at a time- until each point forms its own cluster.



DBSCAN

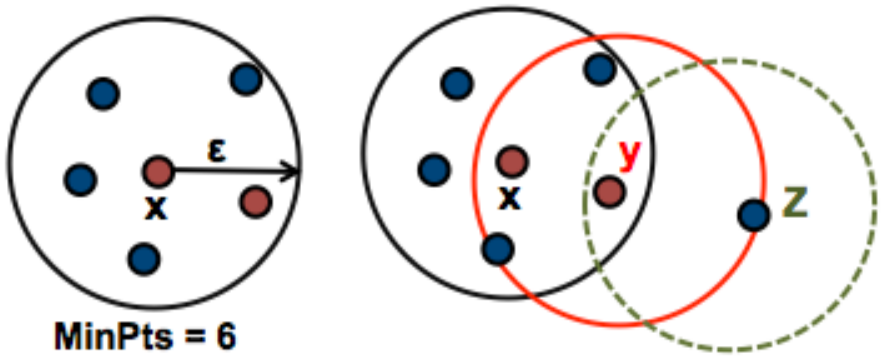
Two important parameters are required for DBSCAN: epsilon (“eps”) and minimum points (“MinPts”). The parameter eps defines the radius of neighborhood around a point x. It’s called the ϵ-neighborhood of x. The parameter MinPts is the minimum number of neighbors within “eps” radius.

- Any point x in the data set, with a neighbor count greater than or equal to MinPts, is marked as a core point.
- We say that x is border point, if the number of its neighbors is less than MinPts, but it belongs to the ϵ-neighborhood of some core point z.
- Finally, if a point is neither a core nor a border point, then it is called a noise point or an outlier.

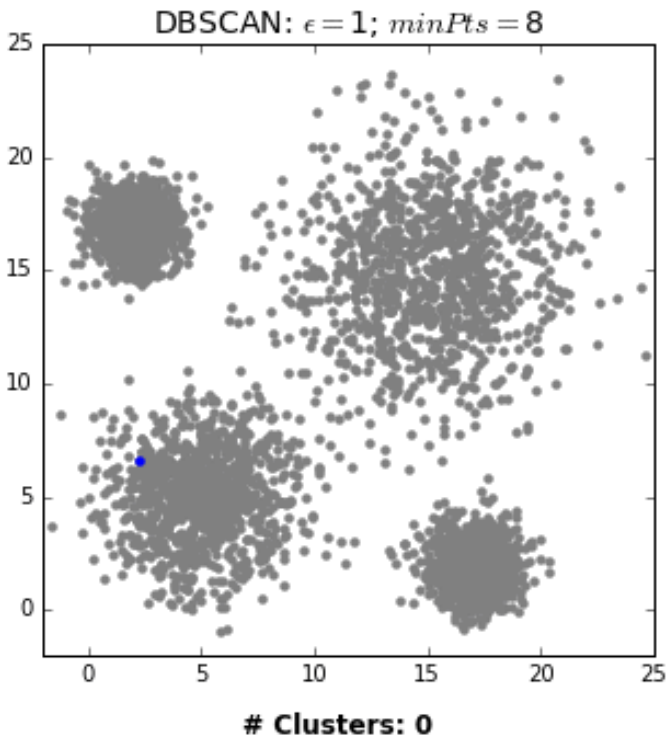
DBSCAN start by defining 3 terms :

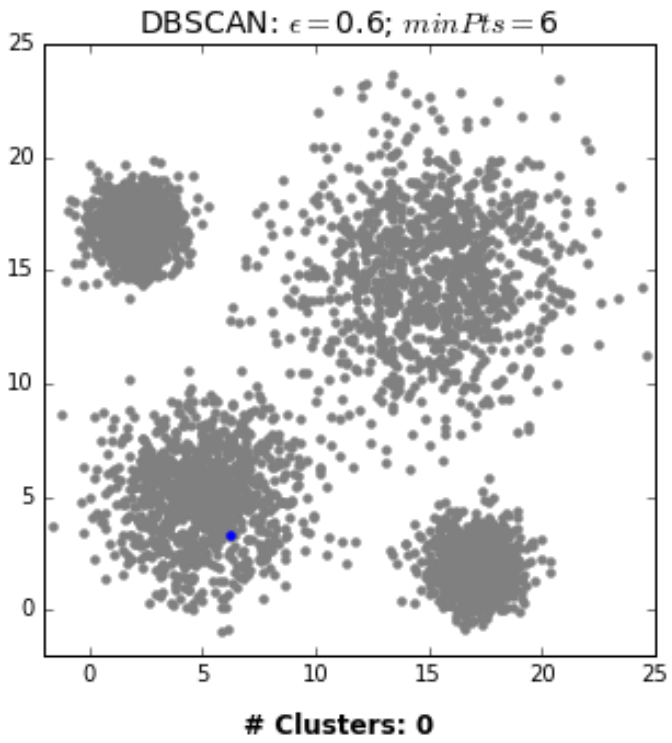
1. *Direct density reachable*,
2. *Density reachable*,
3. *Density connected*.

A density-based cluster is defined as a group of density connected points.



DBSCAN

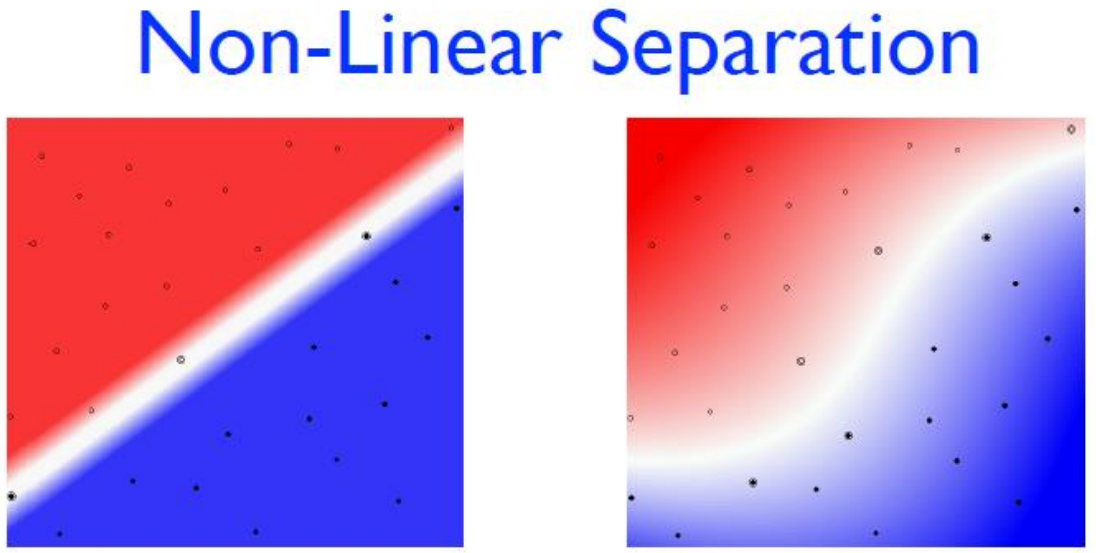

DBSCAN: $\epsilon = 1$; $minPts = 8$
25
20
15
10
5
0
0
5
10
15
20
25
Clusters: 0


DBSCAN: $\epsilon = 0.6$; $minPts = 6$
25
20
15
10
5
0
0
5
10
15
20
25
Clusters: 0

Kernel Method

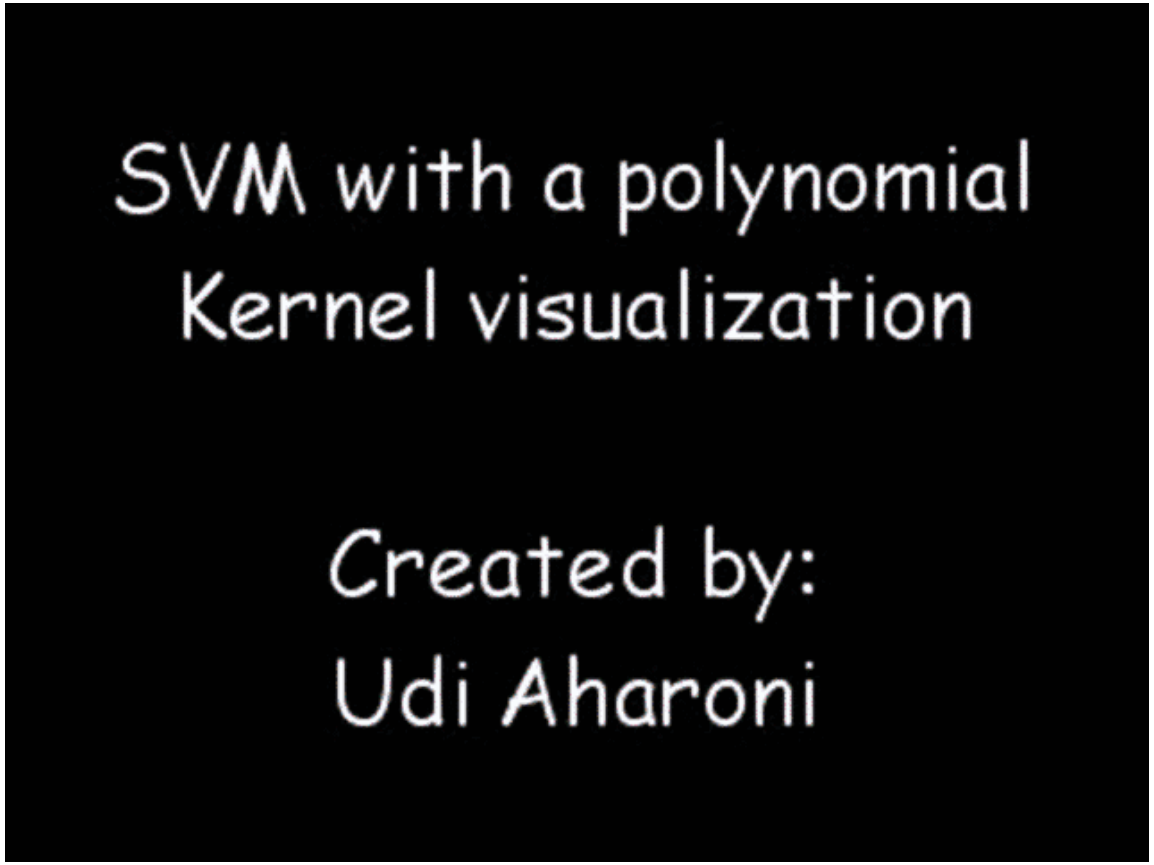
Motivations:

- Efficient computation of inner products in high
- dimension.
- Non-linear decision boundary.
- Non-vectorial inputs.
- Flexible selection of more complex features



Linear separation impossible in most problems. Non-linear mapping from input space to high dimensional feature space $\Phi = X \rightarrow F$ is used in such situations.

Kernel Method



Linear separation impossible in most problems. Non-linear mapping from input space to high dimensional feature space $\Phi = X \rightarrow F$ is used in such situations.

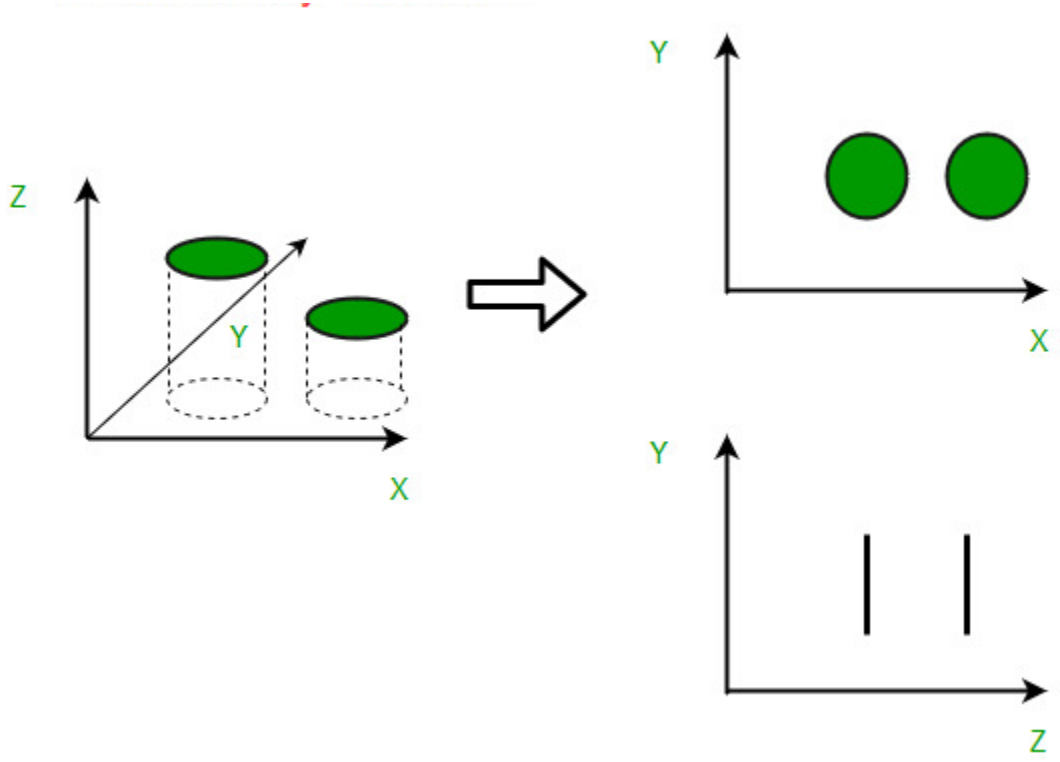
Dimension Reduction

Motivations:

- Some features may be irrelevant
- We want to visualize high dimensional data
- “Intrinsic” dimensionality may be smaller than the number of features
- Cost of computation and storage

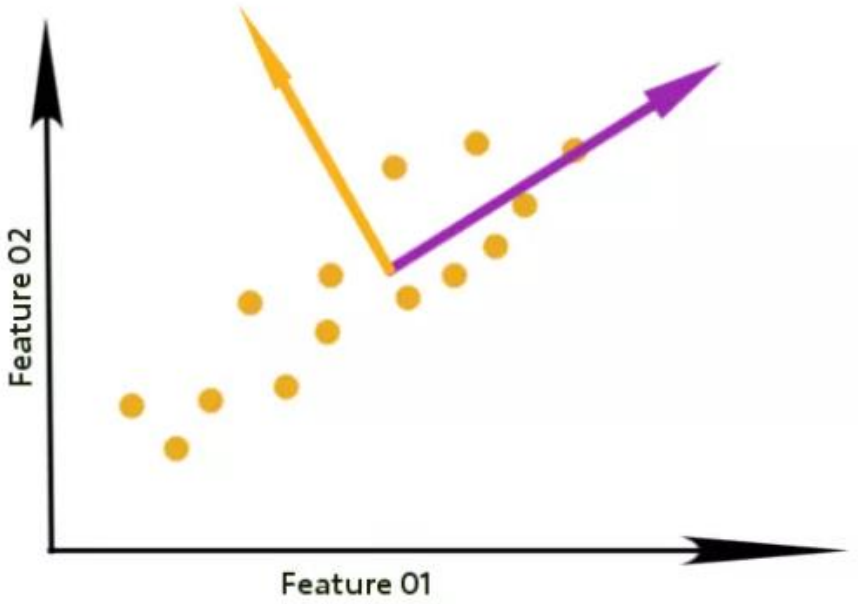
Techniques:

- PCA, SVD, LLE, MDS, LEM, ISOMAP and many others



PCA

- Intuition: find the axis that shows the greatest variation, and project all points into this axis
- Input
  - $X \in \mathbb{R}^d$
- Output:
  - d principle components
  - d eigen vectors $v_1, v_2, \ldots, v_d$ as known as components
  - d eigen values $\lambda_1, \lambda_2, \ldots., \lambda_d$



PCA: notes

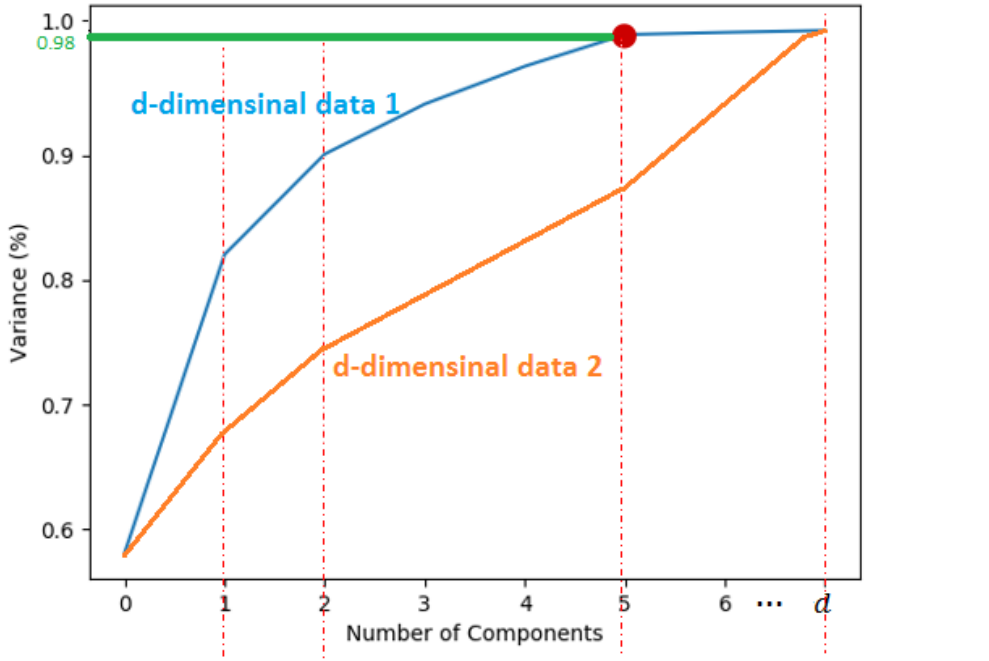
How many PCs should we choose in PCA?

- A Number of components which keep $\alpha\%$ (e.g. 0.98%) of variance of data

In PCA, the total variance is the sum of all eigenvalues

$$\lambda_1 + \lambda_2 + \cdots., +\lambda_d = \sigma_1^2 + \sigma_2^2 + \cdots., +\sigma_d^2$$

$$\lambda_1 > \lambda_2 > \cdots > \lambda_d$$



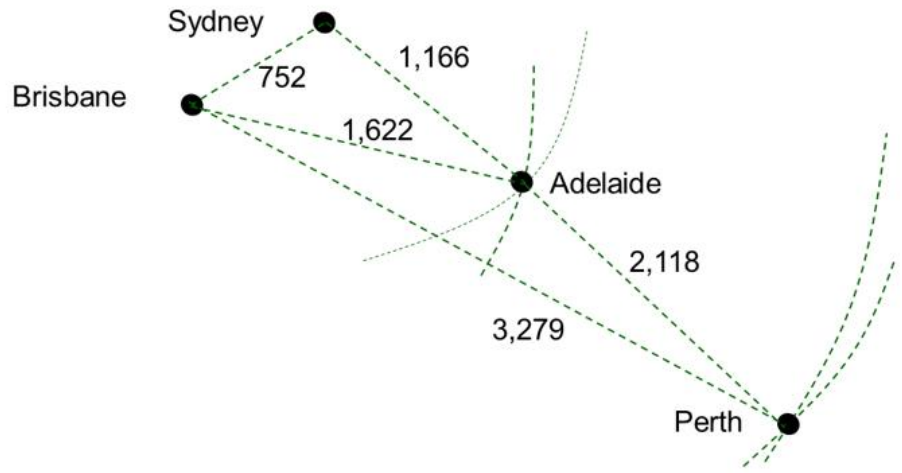
λ_1 $\lambda_1+\lambda_2$ $\lambda_1+\lambda_2+\cdots+\lambda_5$ $\lambda_1+\lambda_2+\cdots+\lambda_d$

$$\lambda_1 > \lambda_2 > \cdots > \lambda_d$$

Multidimensional Scaling (MDS)

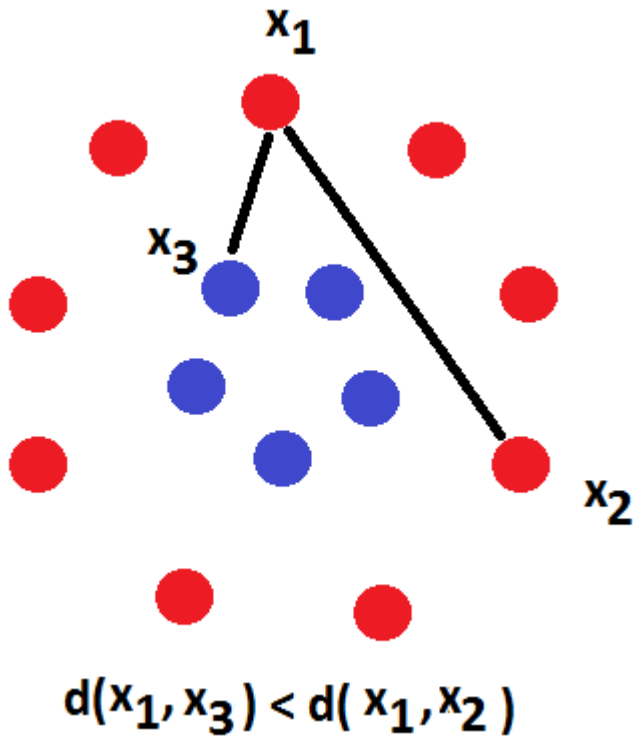
Multidimensional Scaling (MDS) is used to go from a proximity matrix (similarity or dissimilarity) between a series of N objects to the coordinates of these same objects in a p-dimensional space. p is generally fixed at 2 or 3 so that the objects may be visualized easily.

Adelaide	1,166		
Brisbane	752	1,622	
Perth	3,279	2,118	3,606
	Sydney	Adelaide	Brisbane



Metric Learning

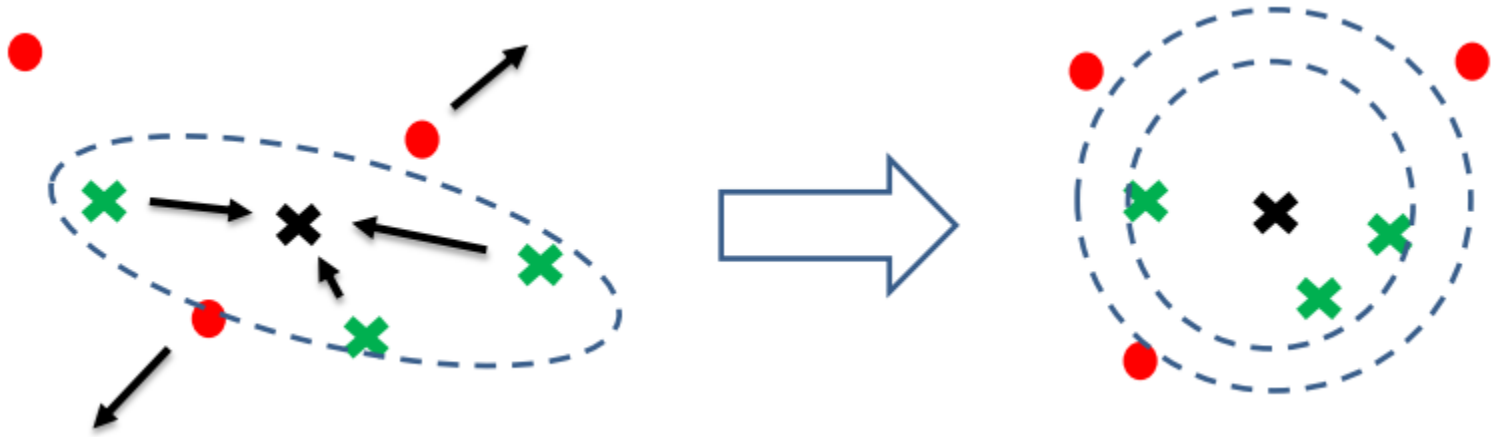
Similarity between objects plays an important role in both human cognitive processes and artificial systems for recognition and categorization. How to appropriately measure such similarities for a given task is crucial to the performance of many machine learning, pattern recognition and data mining methods. This book is devoted to metric learning, a set of techniques to automatically learn similarity and distance functions from data



Metric Learning

- A popular method is metric learning Mahalanobis distance metric learning
- Mahalanobis distance metric is in fact a transformation of data into a new space.

$$Mahalanobis(x, y) = (x - y)^T M^{-1} (x - y)$$

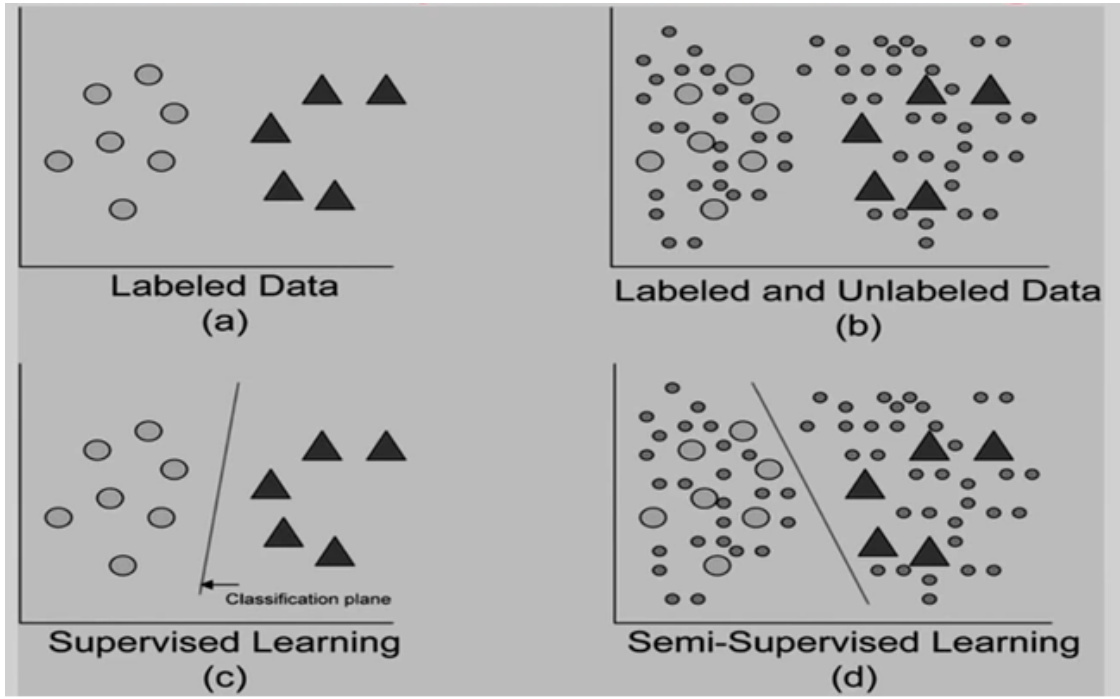


Semi-supervised Learning

Semi-supervised learning is a class of machine learning tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data.

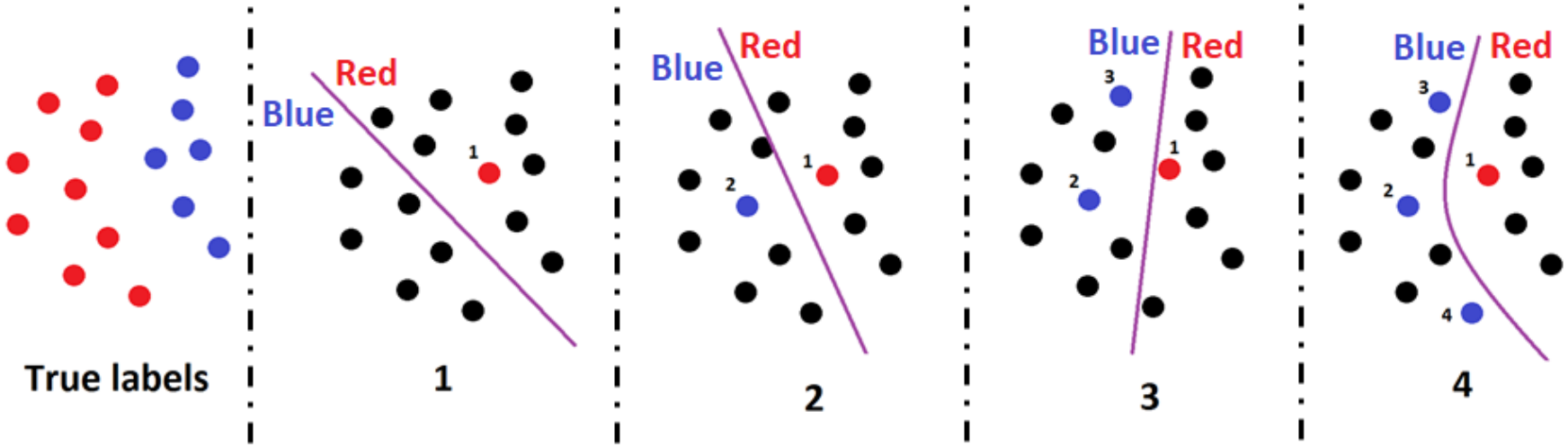
Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data).

Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy.



Active Learning

The concept of active learning is simple—it involves a feedback loop between human and machine that eventually tunes the machine model. The model begins with a set of labeled data that it uses to judge incoming data. Human contributors then label a select sample of the machine's output, and their work is plowed back into the model. Humans continue to label data until the model achieves sufficient accuracy.



7 Steps of Machine Learning

7 Steps of Machine Learning

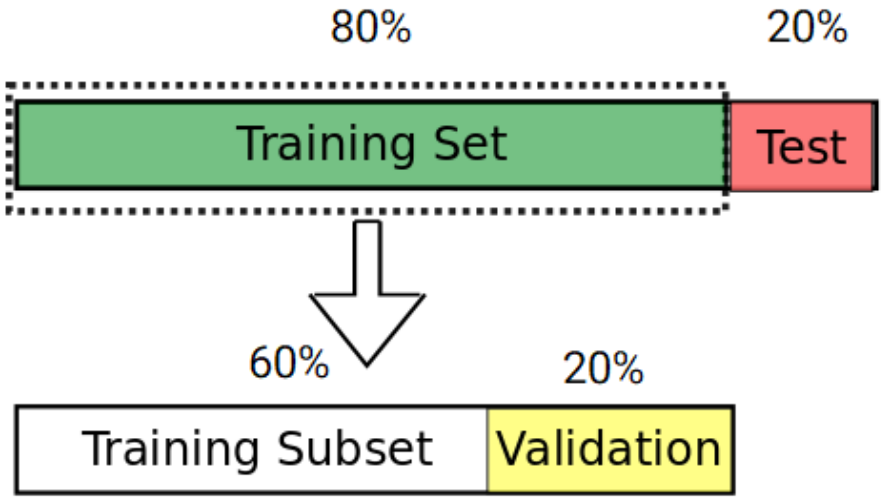
4

Training the Model

Training set: A set of examples used for learning, that is to fit the parameters [i.e., weights] of the classifier.

Validation set: A set of examples used to avoid overfitting or tune the hyperparameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.

Test set: A set of examples used only to assess the performance [generalization] of a fully specified classifier.



7 Steps of Machine Learning

Evaluating machine learning algorithm is an essential part of any project.

7 Steps of Machine Learning

Model Evaluation

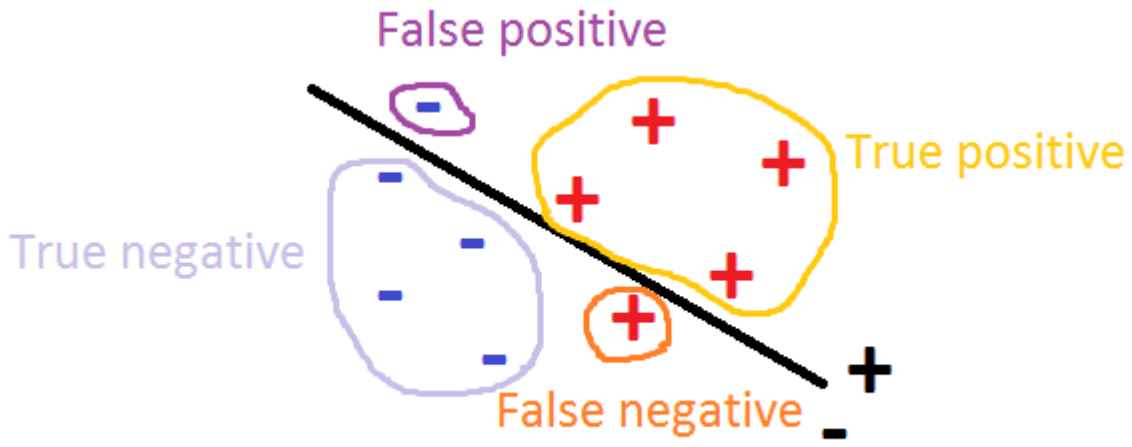
Evaluating machine learning algorithm is an essential part of any project.

Different measures exist to evaluate trained models such as accuracy, TPR, FPR, F-Measure, precision, recall, …

5. Model Evaluation

Confusion Matrix:

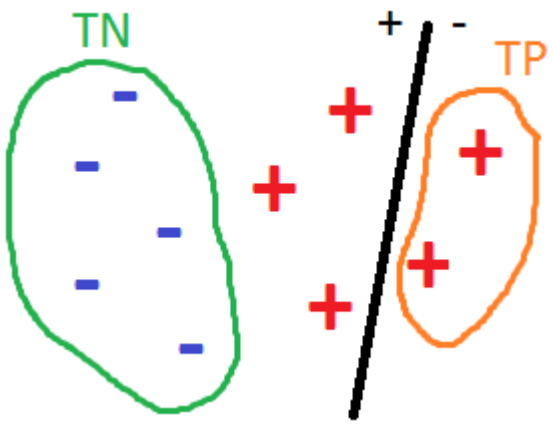
Actual class \ Predicted class	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)



Measures

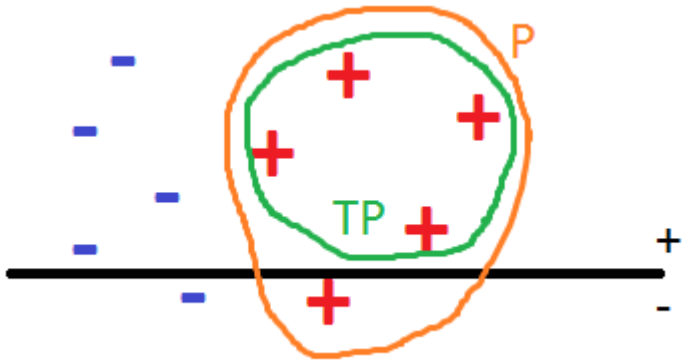
A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

Accuracy= (TP+TN) / ALL



Accuracy= 0.7

True Positive Rate = TP/P



TPR= 0.8

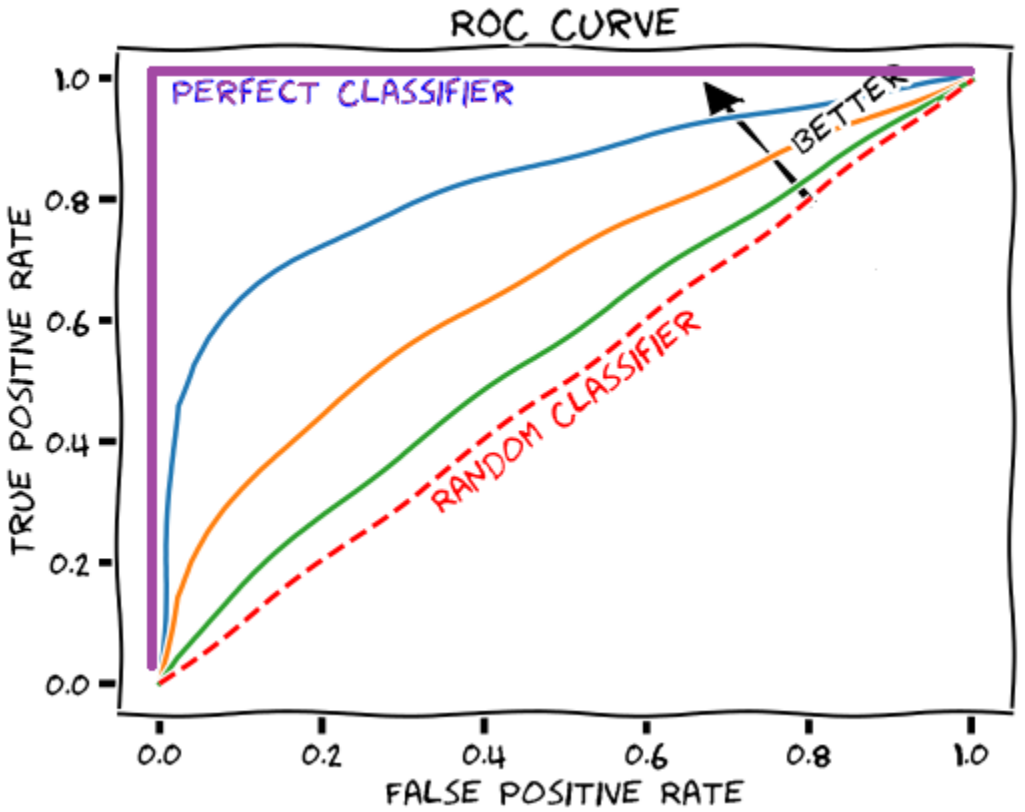
False Positive Rate = FP/N

FPR= 0.2

ROC plot

A ROC plot shows:

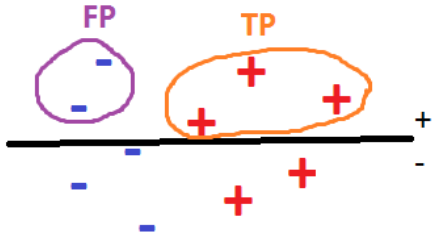
The relationship between TPR and FPR. For example, an increase in TPR results in an increase in FPR.
Test accuracy; the closer the graph is to the top and left-hand borders, the more accurate the test. Likewise, the closer the graph to the diagonal, the less accurate the test. A perfect test would go straight from zero up the top-left corner and then straight across the horizontal.



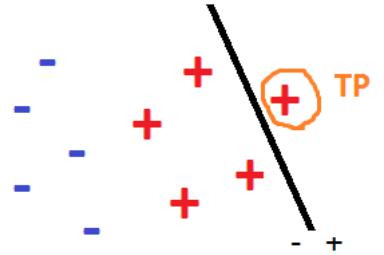
Precision, Recall and F

Precision: exactness – what % of tuples that the classifier labeled as positive are actually positive

Precision = TP/(TP+FP)



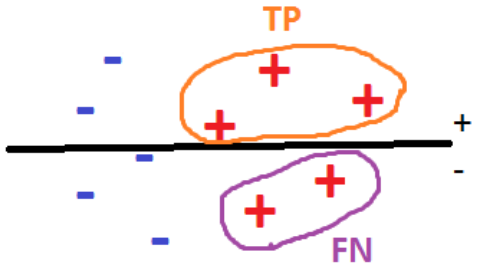
Precision = 3/(3+2)= 60%



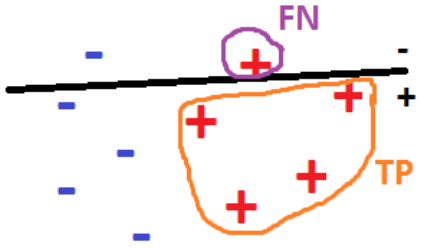
Precision = 1/(1+0)= 100%

Recall: completeness – what % of positive tuples did the classifier label as positive?

Recall = TP/(TP+FN) = TP/P = TPR



Recall = 3/(3+2)= 60%



Recall = 4/(4+1)= 80%

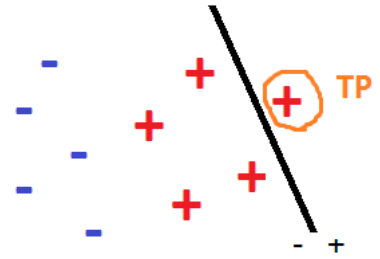
***F* measure (*F*$_1$ or *F*-score)**: harmonic mean of precision and recall,

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

Airport Security Threats

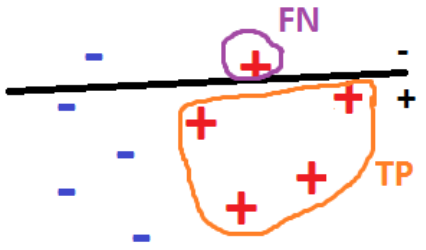
PRECISION OR RECALL?

Airport Security Threats



PRECISION

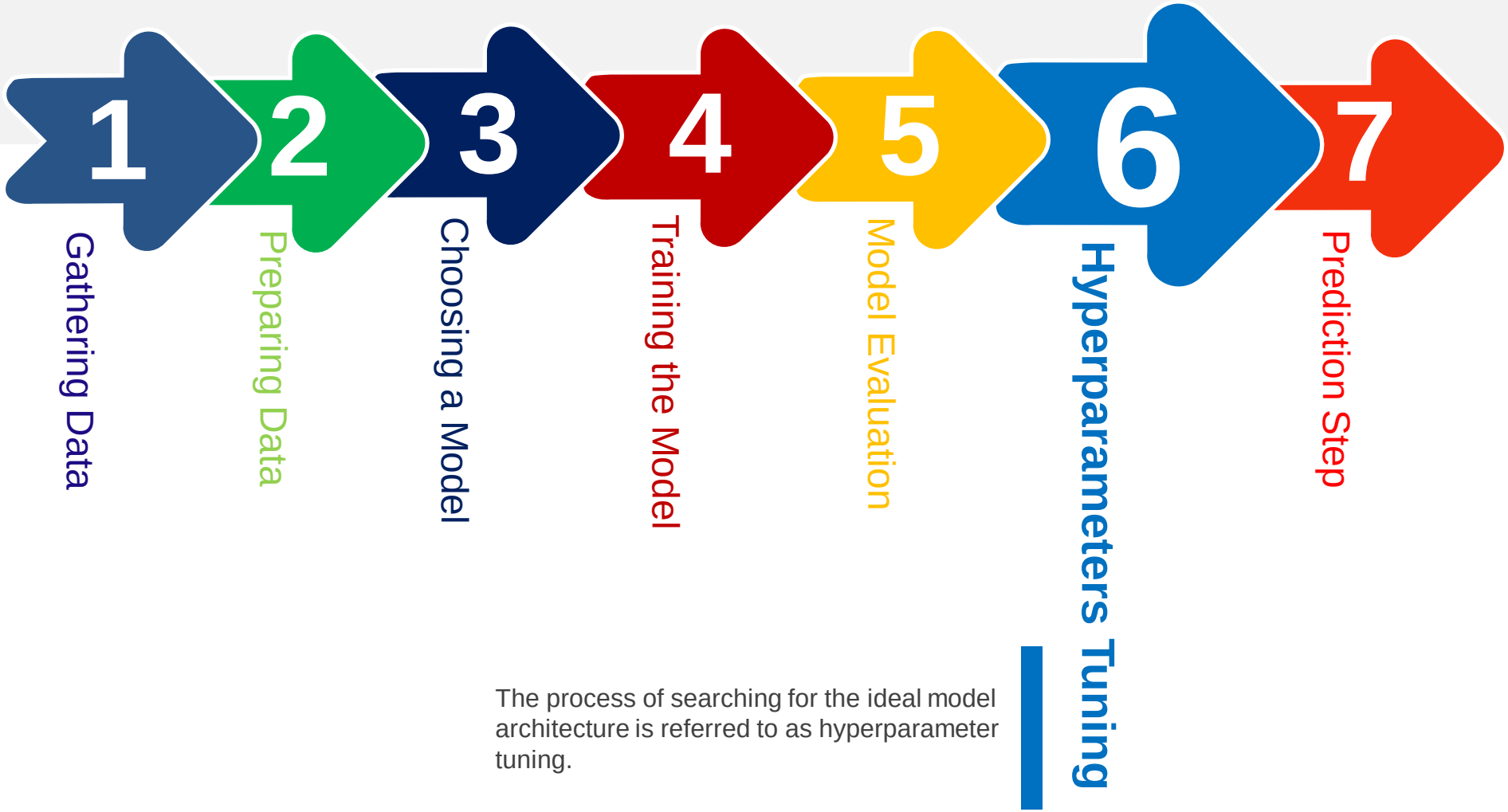
Precision = TP/(TP+FP)



RECALL

Recall = TP/(TP+FN)

7 Steps of Machine Learning



The process of searching for the ideal model architecture is referred to as hyperparameter tuning.

7 Steps of Machine Learning

Hyperparameters Tuning

When creating a machine learning model, you'll be presented with design choices as to how to define your model architecture.

Parameters which define the model architecture are referred to as hyperparameters and thus this process of searching for the ideal model architecture is referred to as hyperparameter tuning.

These hyperparameters might address model design questions such as:

- What degree of polynomial features should I use for my non-linear model?
- What should be the maximum depth allowed for my decision tree?
- What should be the minimum number of samples required at a leaf node in my decision tree?
- How many trees should I include in my random forest?
- How many neurons should I have in my neural network layer?
- How many layers should I have in my neural network?
- What should I set my learning rate to for gradient descent?

Hyperparameters Tuning

I want to be absolutely clear, hyperparameters are not model parameters and they cannot be directly trained from the data. Model parameters are learned during training when we optimize a loss function using something like gradient descent.

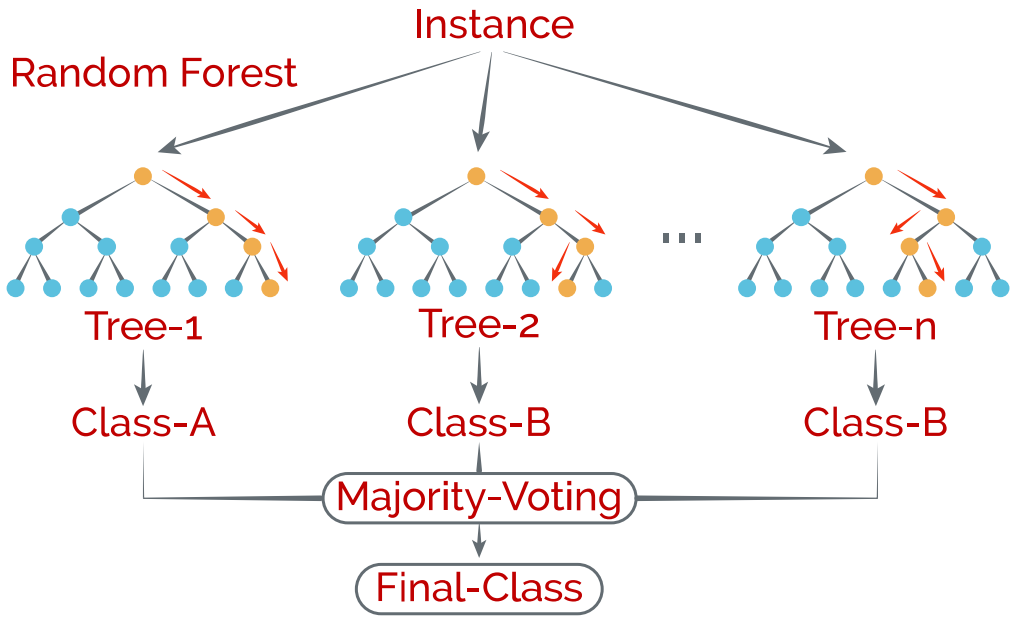
In general, this process includes:

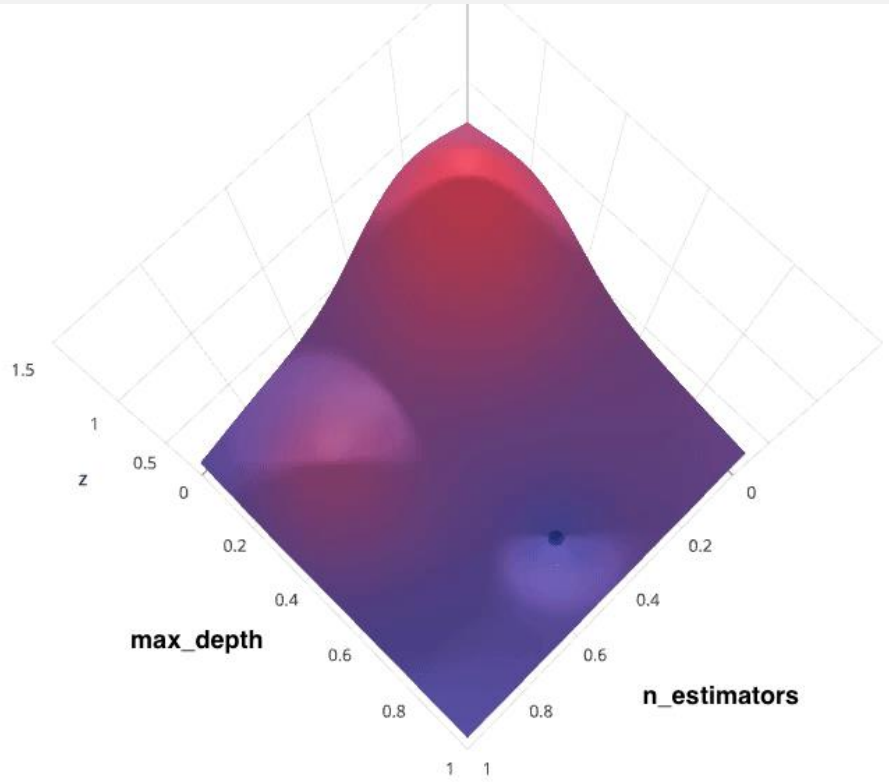
- Define a model
- Define the range of possible values for all hyperparameters
- Define a method for sampling hyperparameter values
- Define an evaluative criteria to judge the model
- Define a cross-validation method

Tuning: An Example

Example: How to search for the optimal structure of a random forest classifier. Random forests are an ensemble model comprised of a collection of decision trees; when building such a model, two important hyperparameters to consider are:

1. How many estimators (i.e.. decision trees) should I use?
2. What should be the maximum allowable depth for each decision tree?



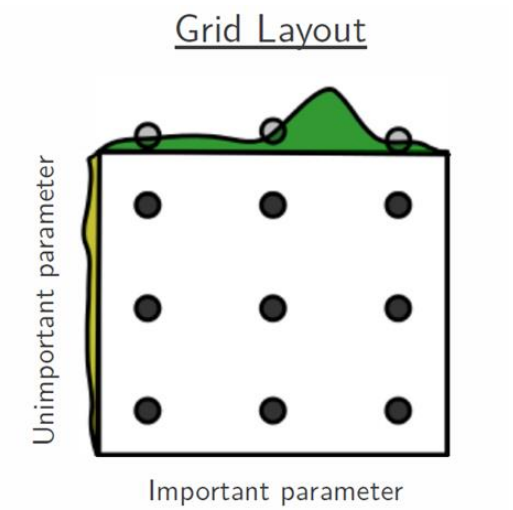


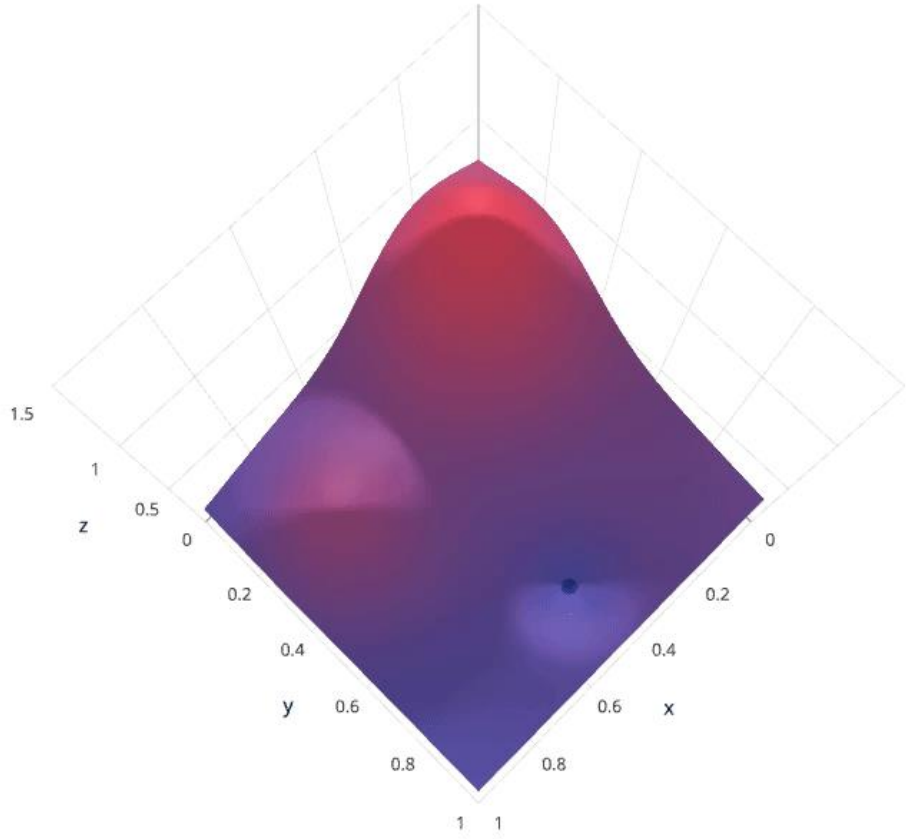
If we had access to such a plot, choosing the ideal hyperparameter combination would be trivial. However, calculating such a plot at the granularity visualized above would be prohibitively expensive.

Tuning Methods

Grid search

Grid search is arguably the most basic hyperparameter tuning method. With this technique, we simply build a model for each possible combination of all of the hyperparameter values provided, evaluating each model, and selecting the architecture which produces the best results.

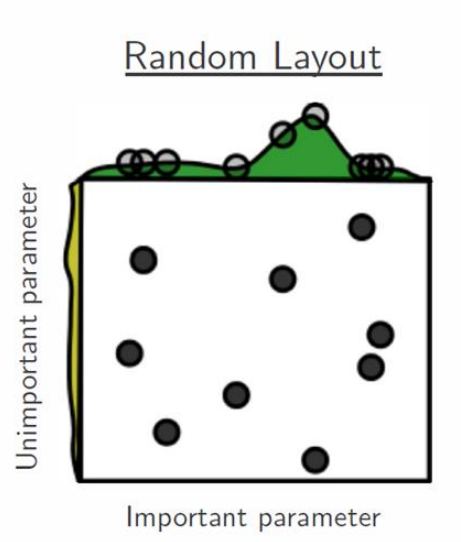


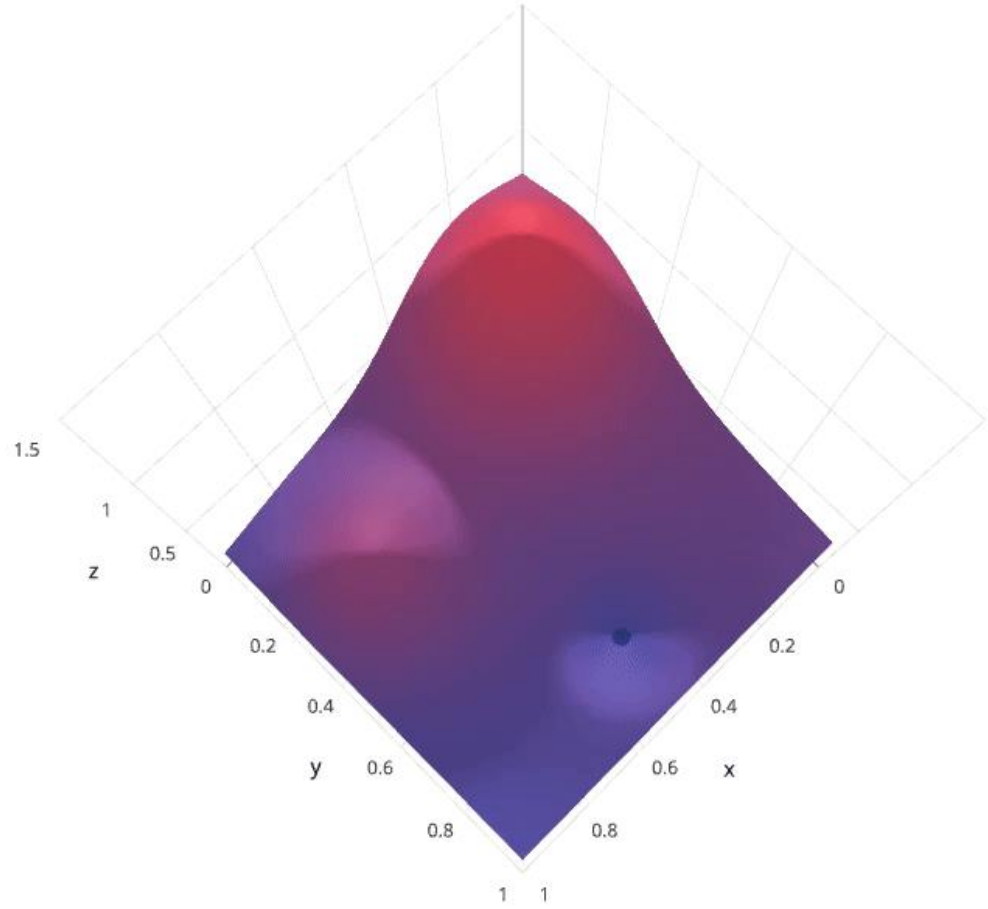


Tuning methods

Random search

Random search differs from grid search in that we longer provide a discrete set of values to explore for each hyperparameter; rather, we provide a statistical distribution for each hyperparameter from which values may be randomly sampled.





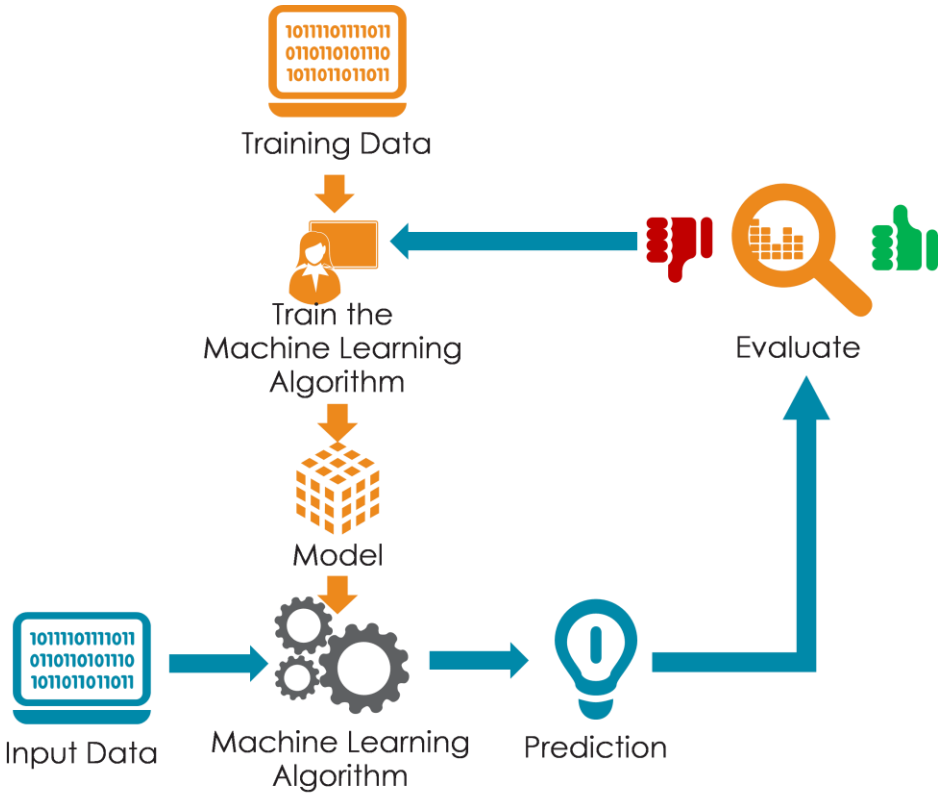
7 Steps of Machine Learning

7 Steps of Machine Learning

7

Prediction Step

Predict unseen data by the trained model



Conclusion

Thank You

Ahmad Ali Abin
Email: a_abin@sbu.ac.ir
Center for Advanced Machine Learning
Web: http://facultymembers.sbu.ac.ir/abin/
Instagram: https://www.instagram.com/cmlsbu/
Call: (021) 29904131- 09127253488