Statistical Data Analysis

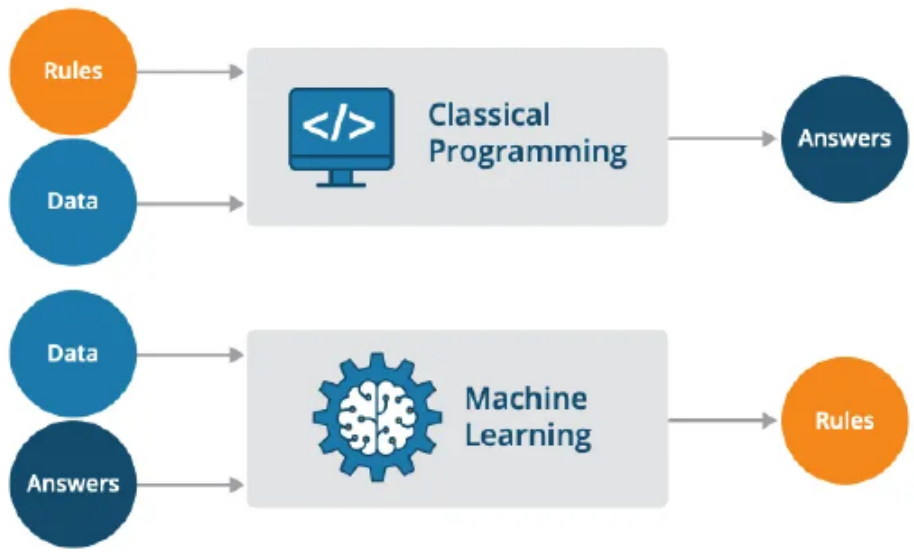
Lecture 6: Machine Learning (Main concepts)

14/October 2025

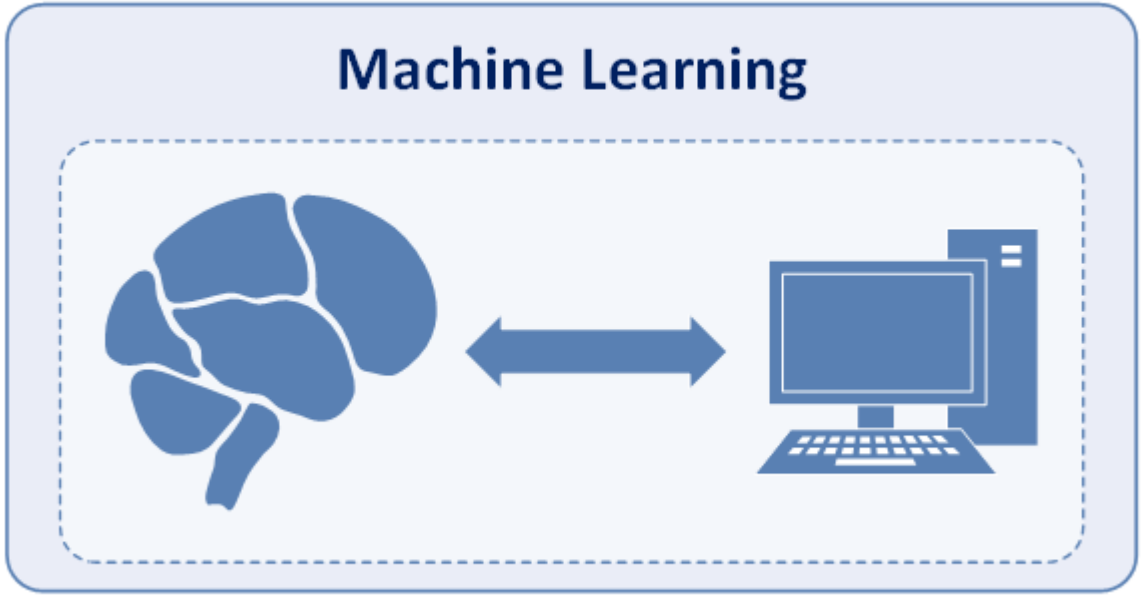
Tania Pereira

tania.pereira@uc.pt

- Artificial intelligence (AI) (1950s) comprises a great variety of sub-fields from science to engineering.
  - Understand and replicate the basis of human intelligence.
- Machine learning (ML) is a sub-field of AI:
  - **Are computers able to perform a specific task by automatically learn the required rules from data?**
- Instead of being explicitly programmed, ML systems are trained.



Machine learning is a subset of AI, which uses algorithms that learn from data to make predictions. These predictions can be generated through supervised learning, where algorithms learn patterns from existing data, or unsupervised learning, where they discover general patterns in data. ML models can predict numerical values based on historical data, categorize events as true or false, and cluster data points based on commonalities.



AI vs. Machine Learning & Deep Learning

AI vs. ML vs. DL

Artificial Intelligence

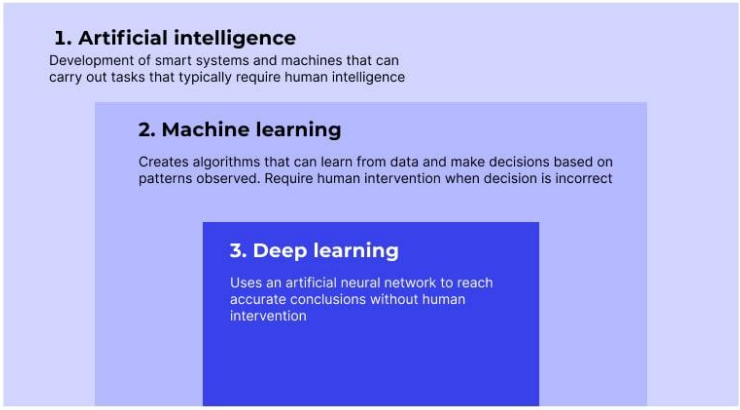
- Artificial Intelligence originated around 1950's
- AI represents simulate intelligence in machines
- AI is a subset of data science
- Aim to build machines which are capable of thinking like humans

Machine Learning

- Machine Learning originated around 1960's
- Machine Learning is the practice of getting machines to make decisions without being programmed
- Machine Learning is a subset of AI & Data Science
- Aim to make machine learn through data so that they can solve problems

Deep Learning

- Deep Learning originated around 1970's
- Deep Learning is the process of using artificial neural networks to solve complex problems
- Deep Learning is a subset of Machine Learning, AI & Data Science
- Aim to build neural networks that authentically discover patterns for feature detection



Supervised *vs* Unsupervised *vs* Reinforcement Learning

Definition

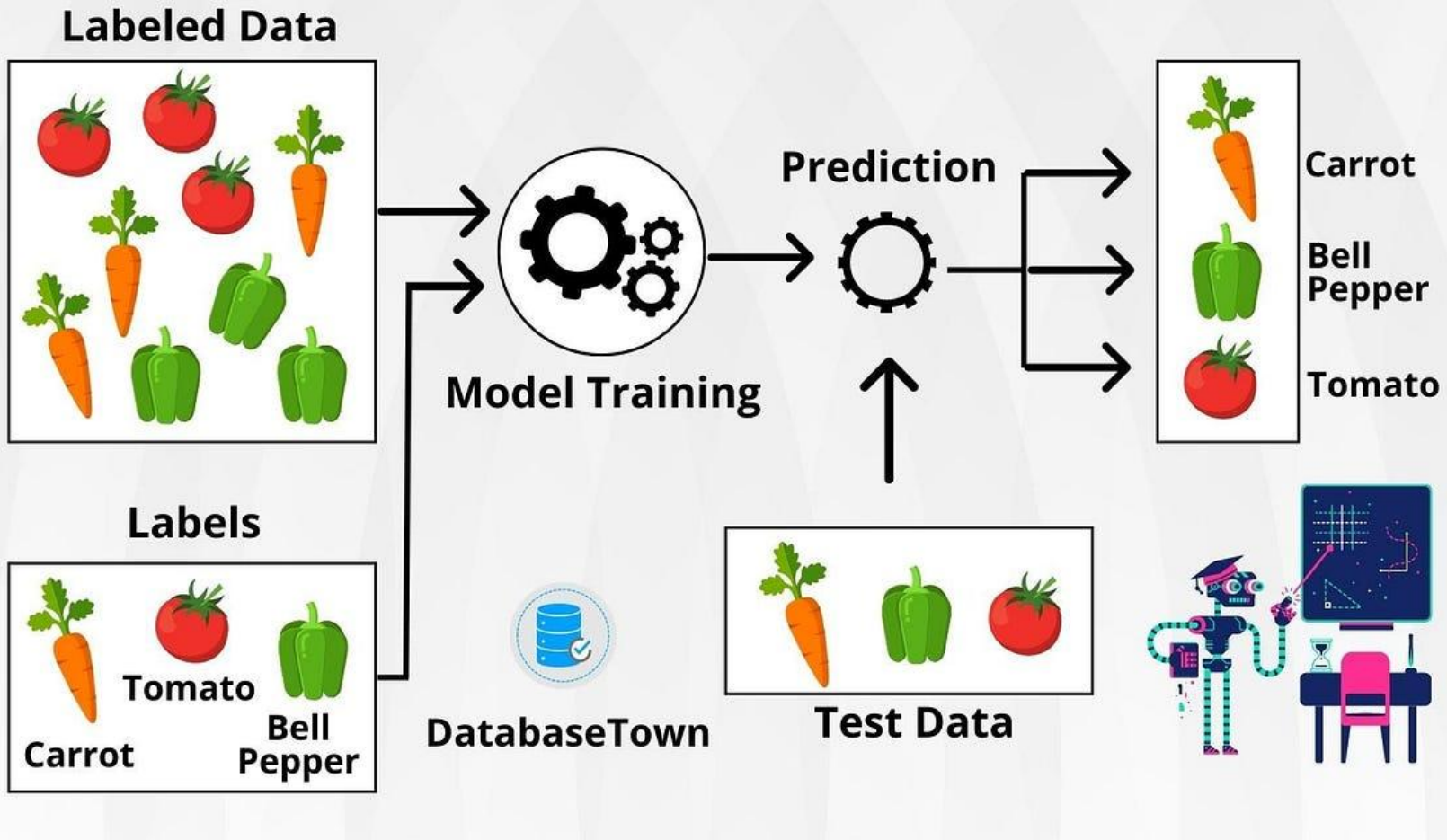
Supervised learning is a method in which we teach the machine using labelled data

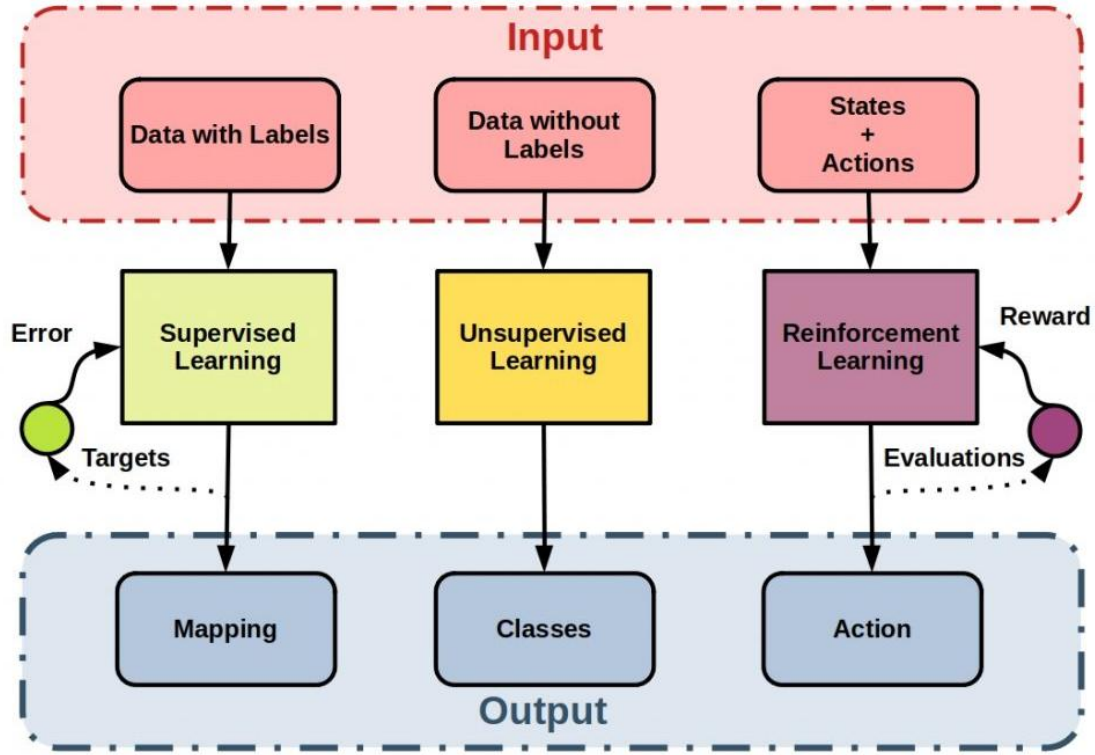
In unsupervised learning the machine is trained on unlabelled data without any guidance

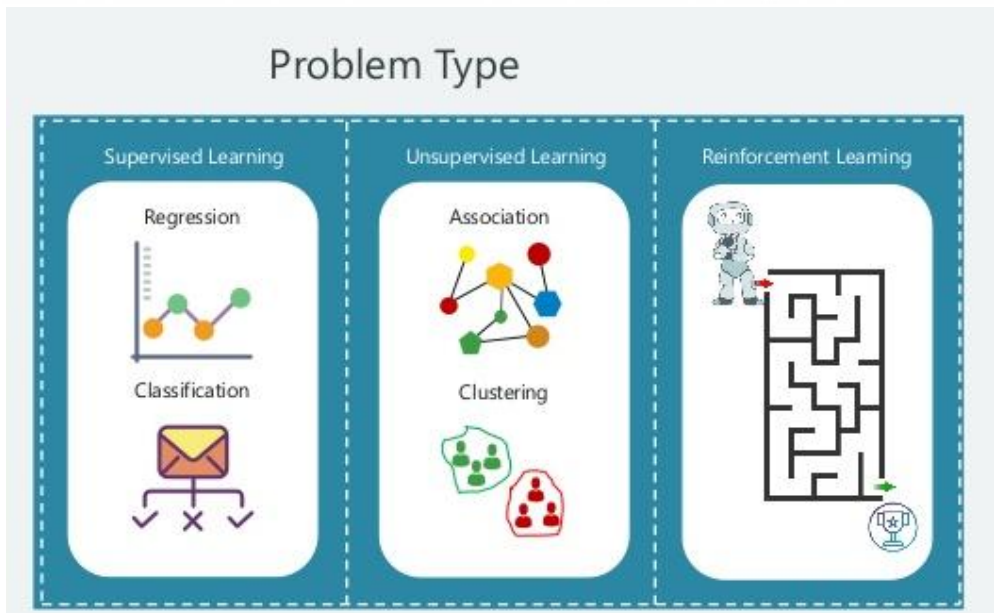
In Reinforcement learning an agent interacts with its environment by producing actions & discovers errors or rewards

SUPERVISED LEARNING

Supervised machine learning is a branch of artificial intelligence that focuses on training models to make predictions or decisions based on labeled training data.

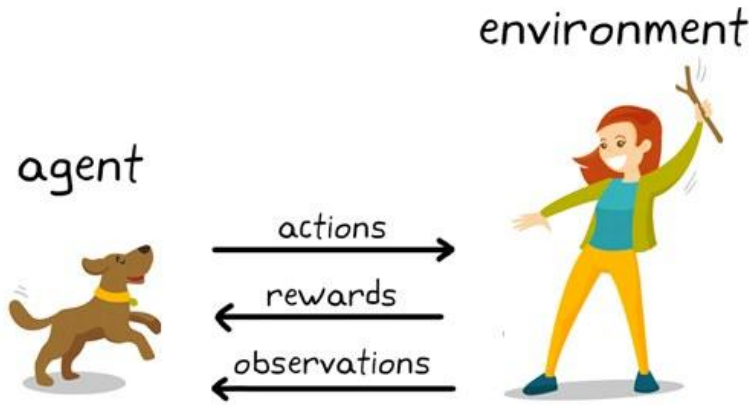


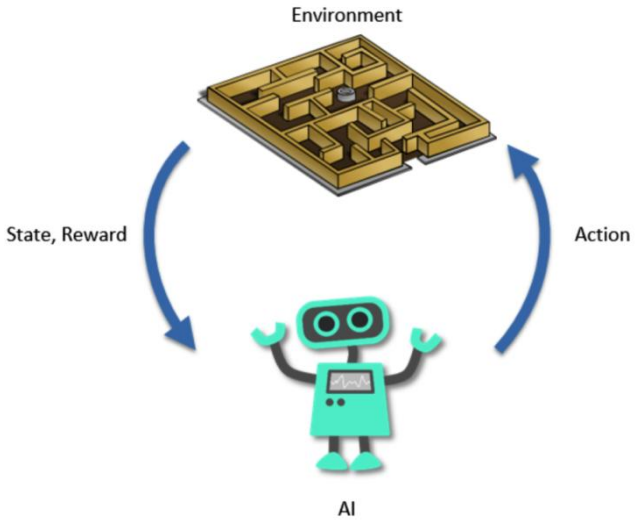

Input
Data with Labels
Data without Labels
States + Actions
Error
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Reward
Targets
Evaluations
Mapping
Classes
Action
Output

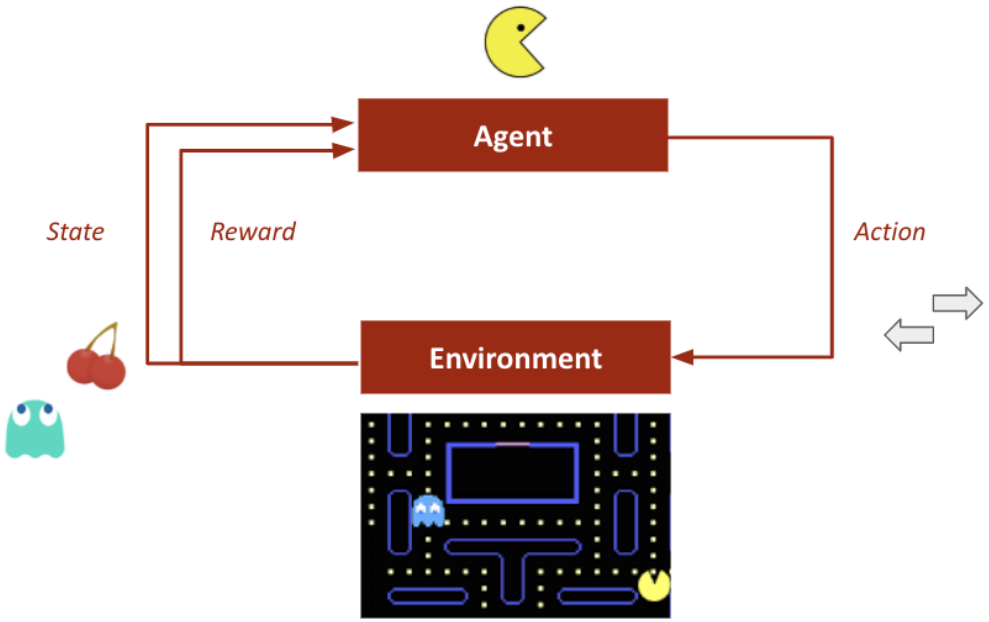

Problem Type
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Regression
Classification
Association
Clustering

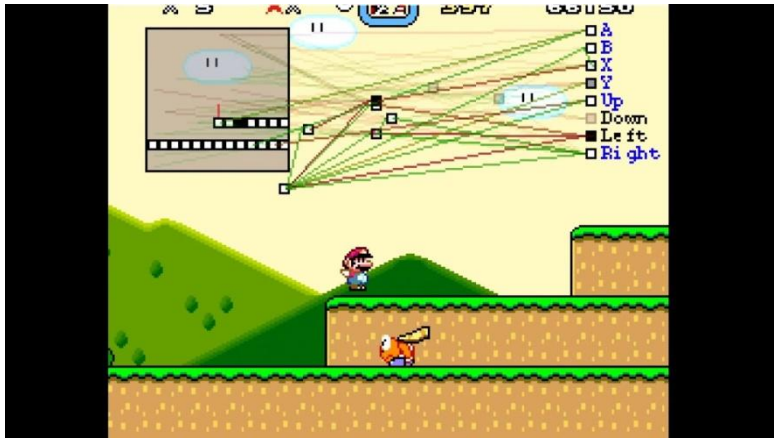

Machine Learning
Supervised Learning
Reinforcement Learning
Unsupervised Learning
Agent
State
Reward
Action
Environment
Classification
• Naïve Bayesian
• Decision tree
• SVM
• k-Nearest Neighbors
• Neural Networks
• ……
Regression
• Logistic Regression
• Stepwise Regression
• Polynomial Regression
• Ridge Regression
• Linear Regression
• ……
Reinforcement learning
• Q-learning
• Policy Gradients
• Deep Q-learning
• Actor-Critic
• DDPG
• ……
Clustering
• WAVE-CLUSTER
• Self Organized Maps
• Gaussian Mixture
• K-means
• DBSCAN
• ……
Density Estimation
• Kernel Density
• Generalized De-noising
• Deep Boltzmann Machine
• Auto-Encoders
• ……
Dimensionality Reduction
• Local Linear Embedding
• Stacked Auto-Encoders
• Principe Component Analysis
• Partial Least
• ……

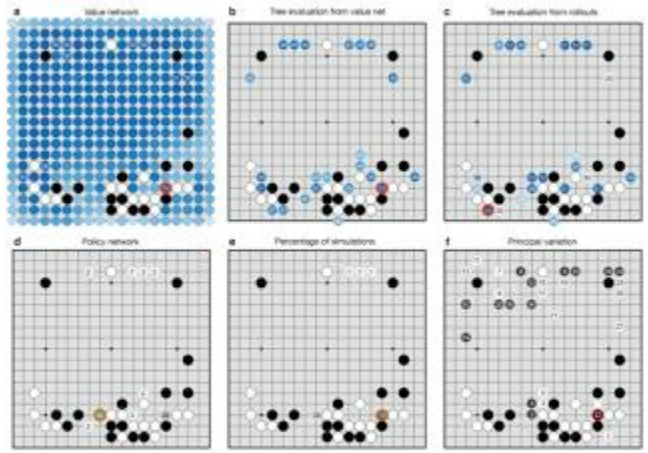
Reinforcement Learning



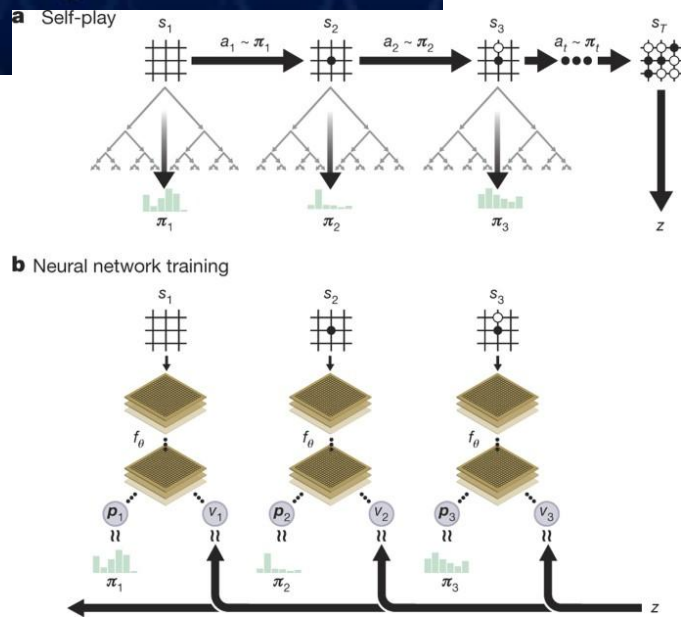


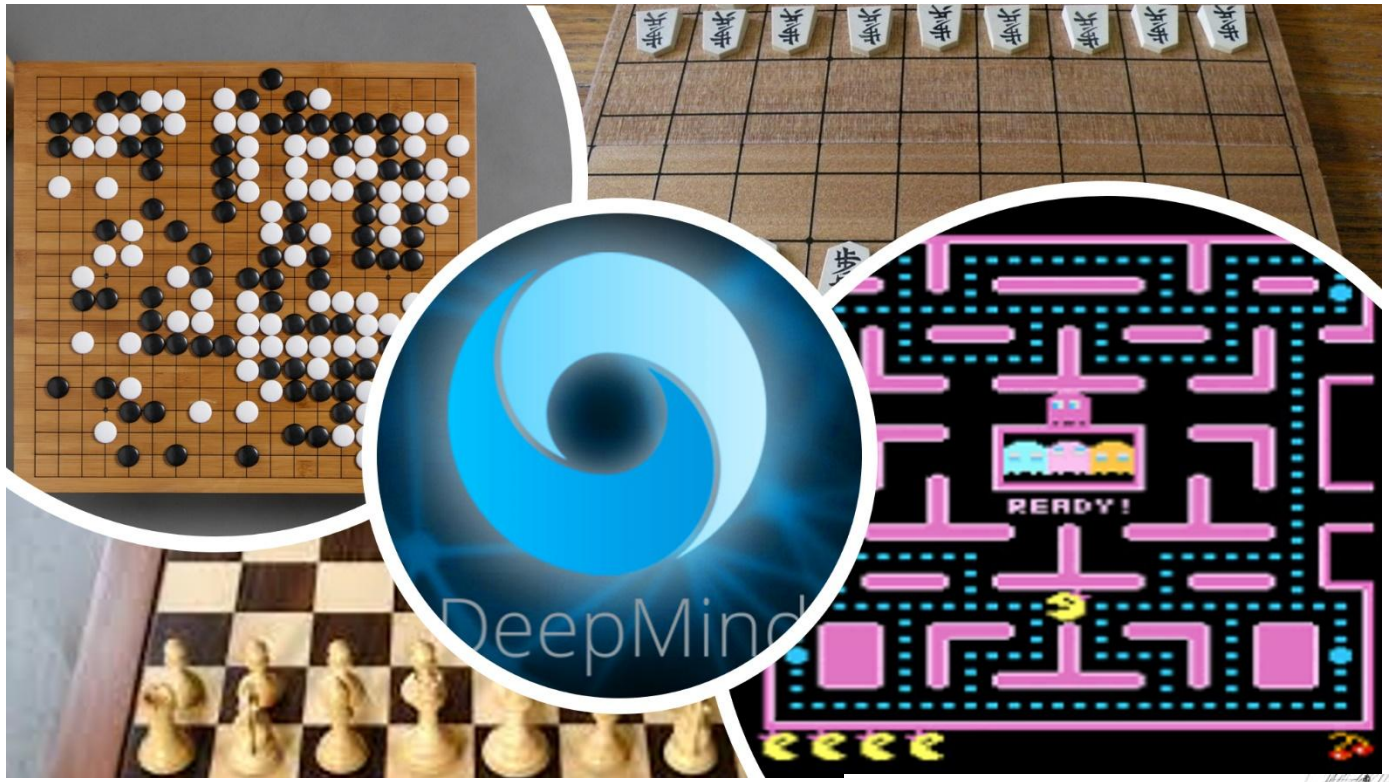


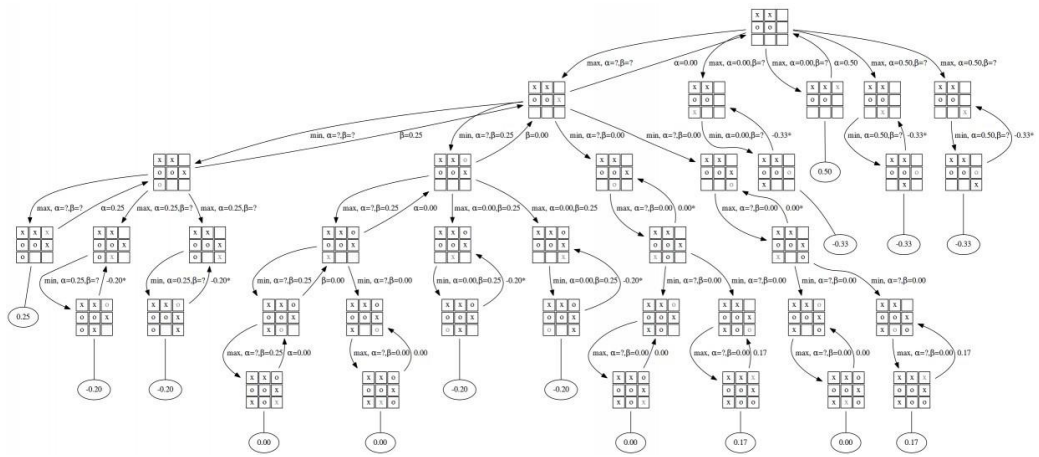


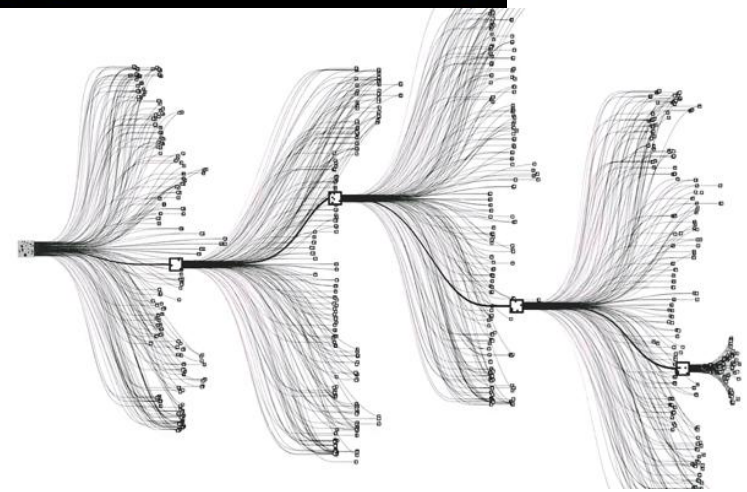


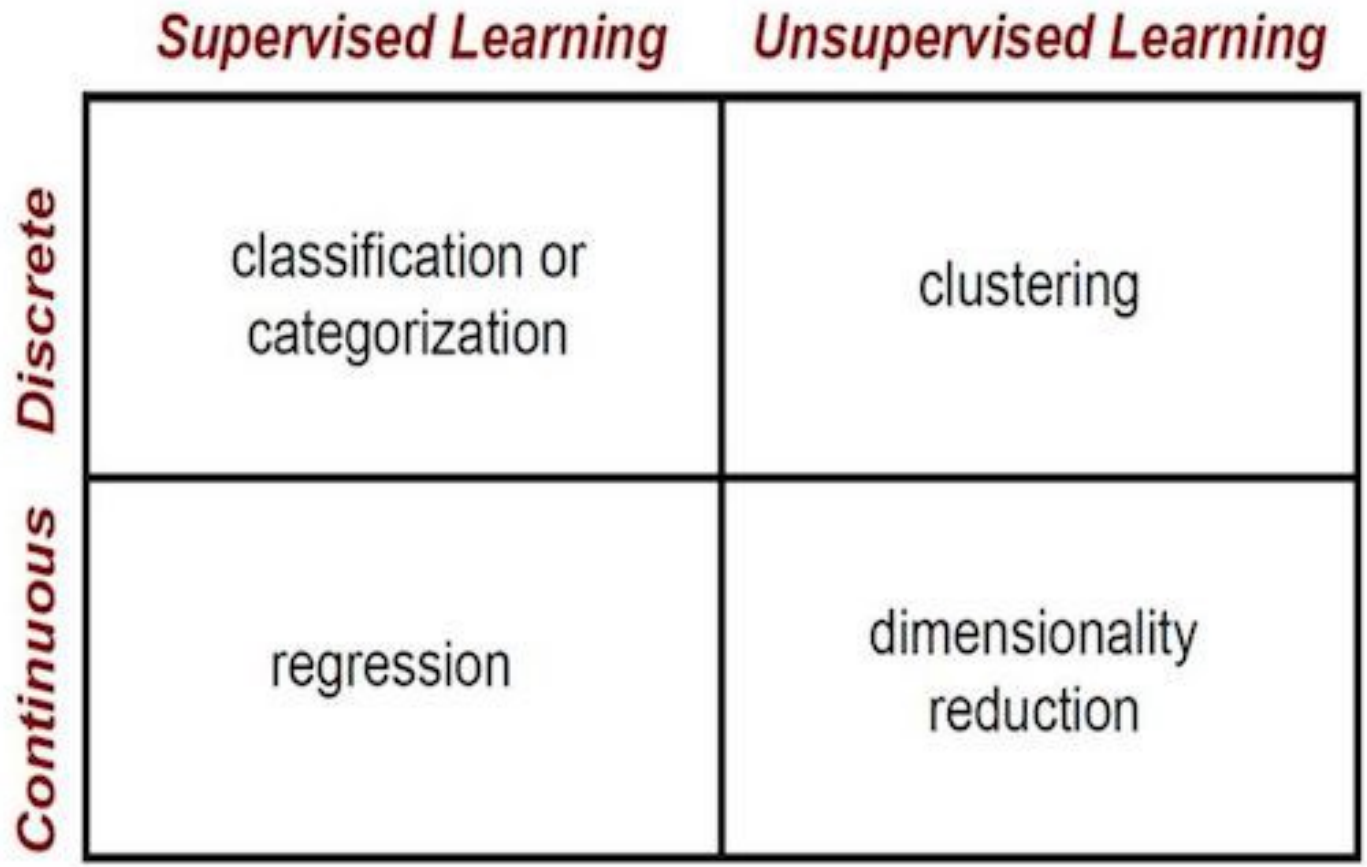
AlphaGo

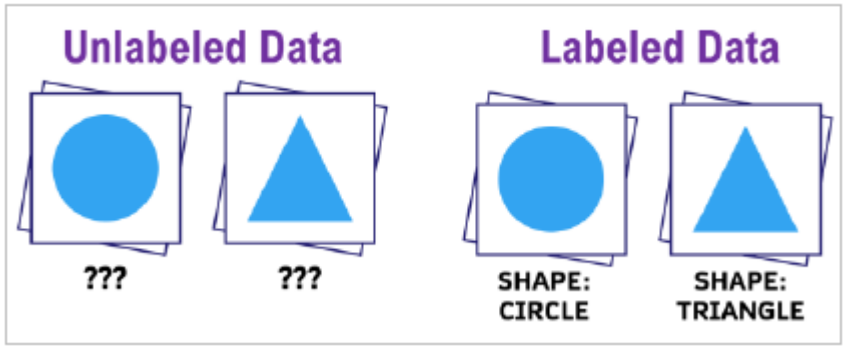



DeepMind
READY!



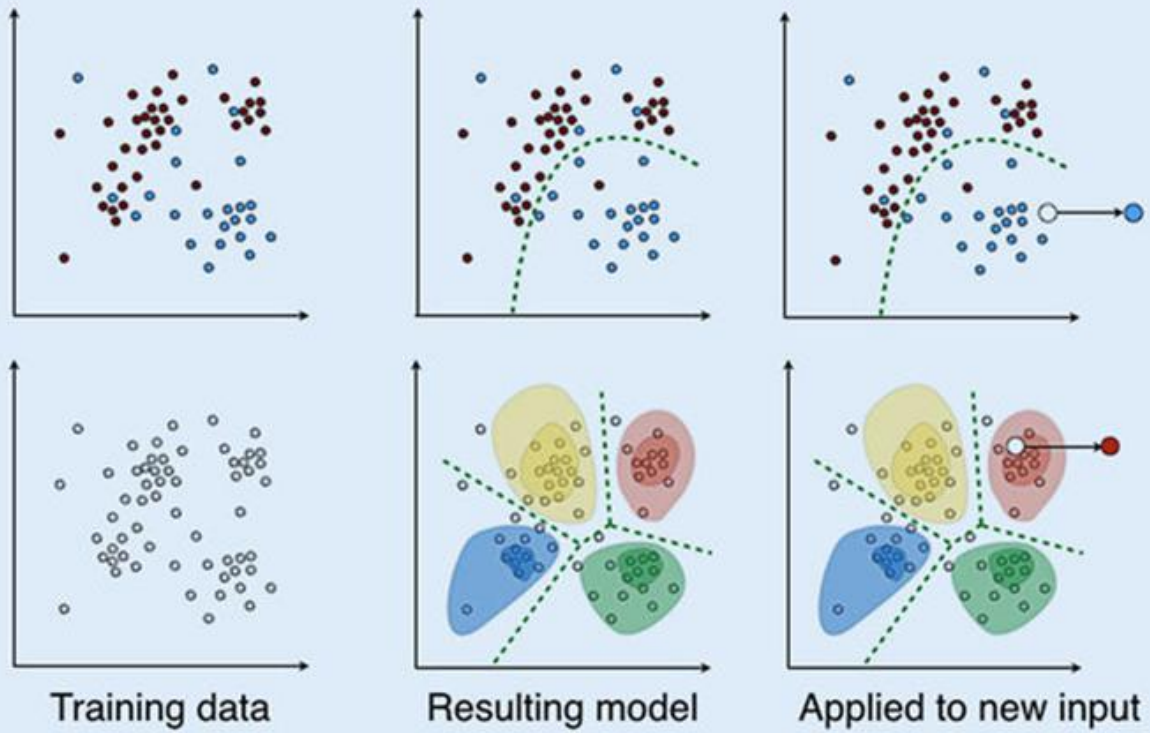


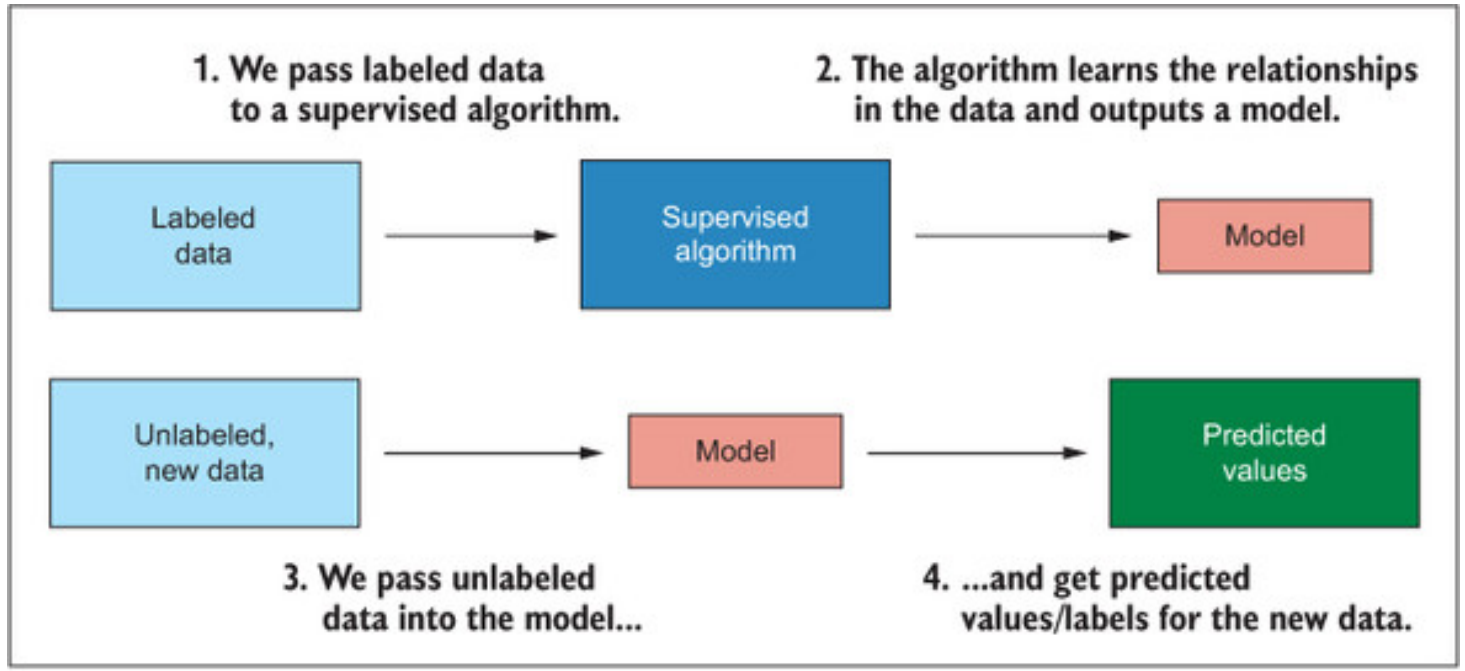

Supervised Learning
Unsupervised Learning
Discrete
Continuous
classification or categorization
clustering
regression
dimensionality reduction

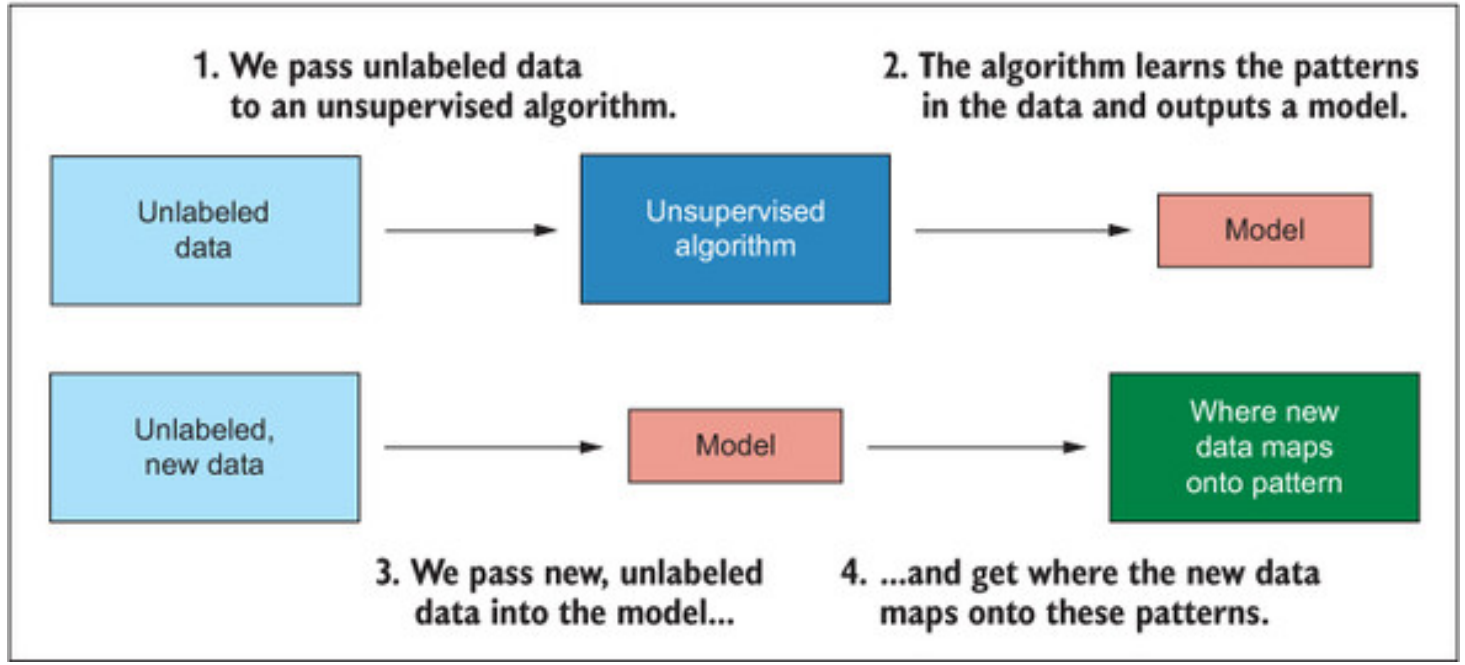


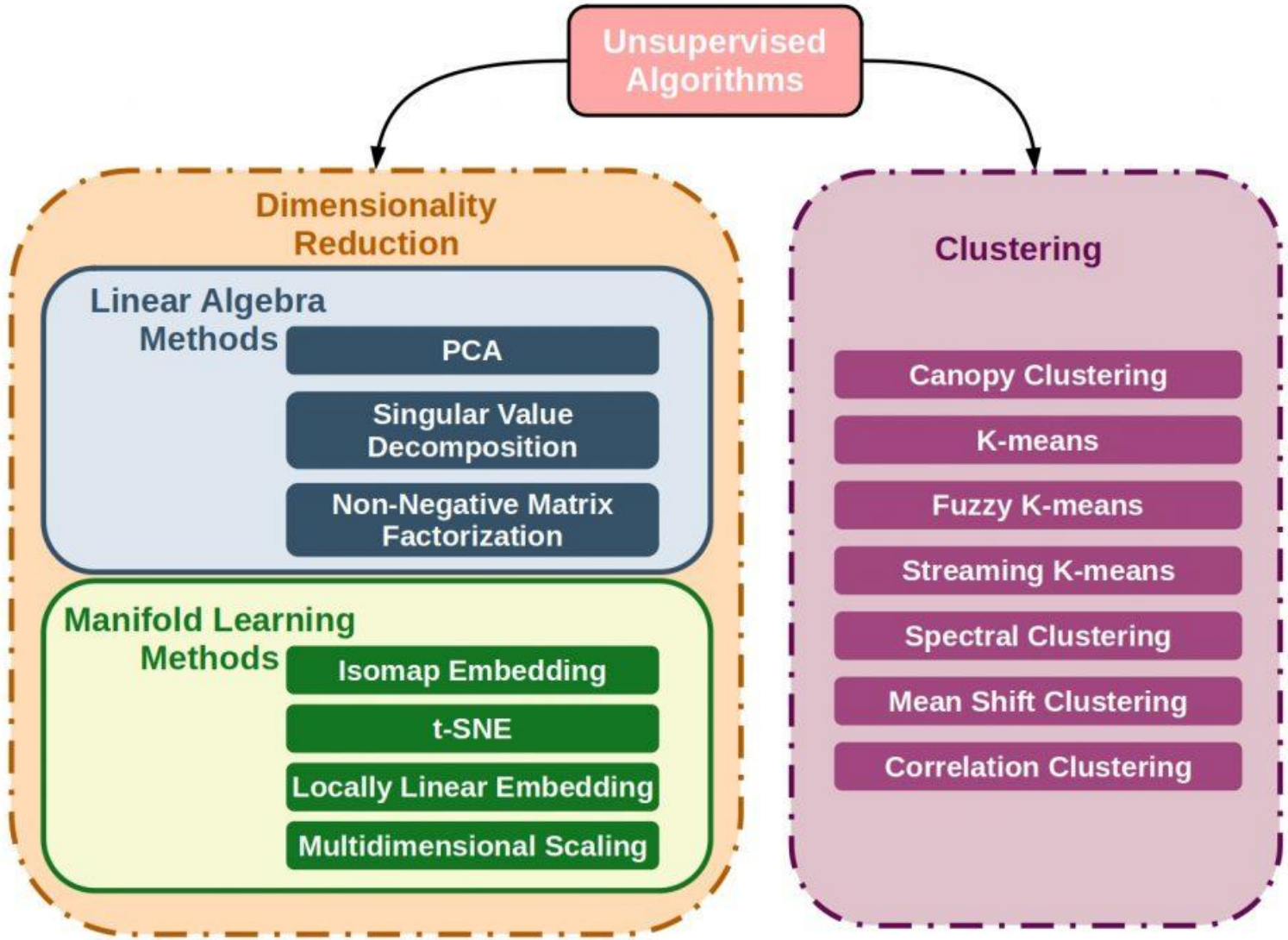
Supervised learning: each training example has a ground truth label. The model learns a decision boundary and replicates the labeling on new data.

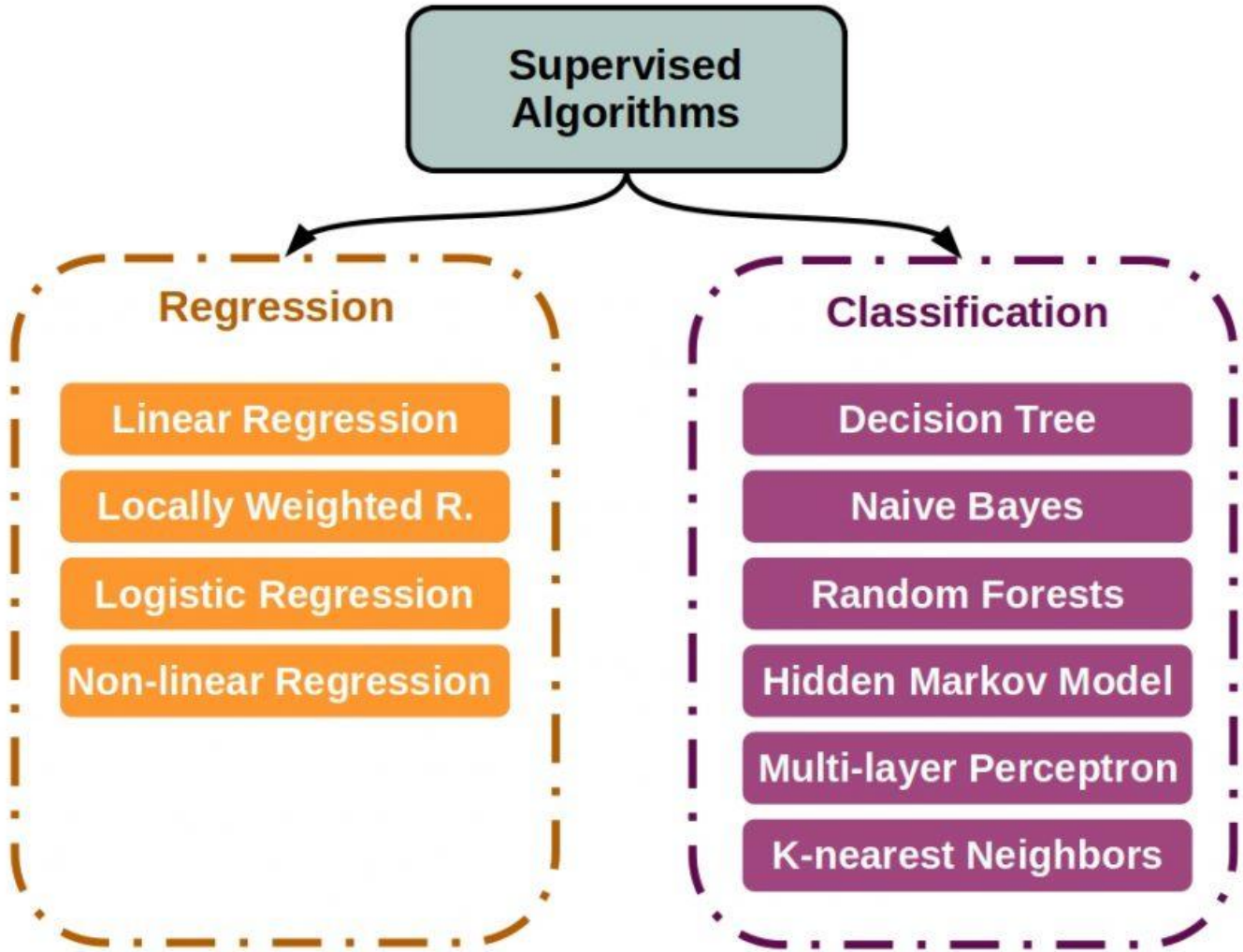
Unsupervised learning: training examples do not have ground truth labels. The model identifies structure such as clusters. New data can be assigned to clusters.




1. We pass labeled data to a supervised algorithm.
2. The algorithm learns the relationships in the data and outputs a model.
Labeled data
Supervised algorithm
Model
Unlabeled, new data
Model
Predicted values
3. We pass unlabeled data into the model...
4. ...and get predicted values/labels for the new data.


1. We pass unlabeled data to an unsupervised algorithm.
2. The algorithm learns the patterns in the data and outputs a model.
Unlabeled data
Unsupervised algorithm
Model
Unlabeled, new data
Model
Where new data maps onto pattern
3. We pass new, unlabeled data into the model...
4. ...and get where the new data maps onto these patterns.


Unsupervised Algorithms
Dimensionality Reduction
Linear Algebra Methods
PCA
Singular Value Decomposition
Non-Negative Matrix Factorization
Manifold Learning Methods
Isomap Embedding
t-SNE
Locally Linear Embedding
Multidimensional Scaling
Clustering
Canopy Clustering
K-means
Fuzzy K-means
Streaming K-means
Spectral Clustering
Mean Shift Clustering
Correlation Clustering


Supervised Algorithms
Regression
Linear Regression
Locally Weighted R.
Logistic Regression
Non-linear Regression
Classification
Decision Tree
Naive Bayes
Random Forests
Hidden Markov Model
Multi-layer Perceptron
K-nearest Neighbors

Machine learning models and their training algorithms

Supervised learning

Data scientists provide input, output and feedback to build model (as the definition).

EXAMPLE ALGORITHMS:

Linear regressions

- Sales forecasting.
- Risk assessment.

Support vector machines

- Image classification.
- Financial performance comparison.

Decision trees

- Predictive analytics.
- Pricing.

Unsupervised learning

Use deep learning to arrive at conclusions and patterns through unlabeled training data.

EXAMPLE ALGORITHMS:

Apriori

- Sales functions.
- Word associations.
- Searcher.

K-means clustering

- Performance monitoring.
- Searcher intent.

Artificial neural networks

- Generate new, synthetic data.
- Data mining and pattern recognition.

Semi-supervised learning

Builds a model through a mix of labeled and unlabeled data, a set of categories, suggestions and exampled labels.

EXAMPLE ALGORITHMS:

Generative adversarial networks

- Audio and video manipulation.
- Data creation.

Self-trained Naïve Bayes classifier

- Natural language processing.

Reinforcement learning

Self-interpreting but based on a system of rewards and punishments learned through trial and error, seeking maximum reward.

EXAMPLE ALGORITHMS:

Q-learning

- Policy creation.
- Consumption reduction.

Model-based value estimation

- Linear tasks.
- Estimating parameters.

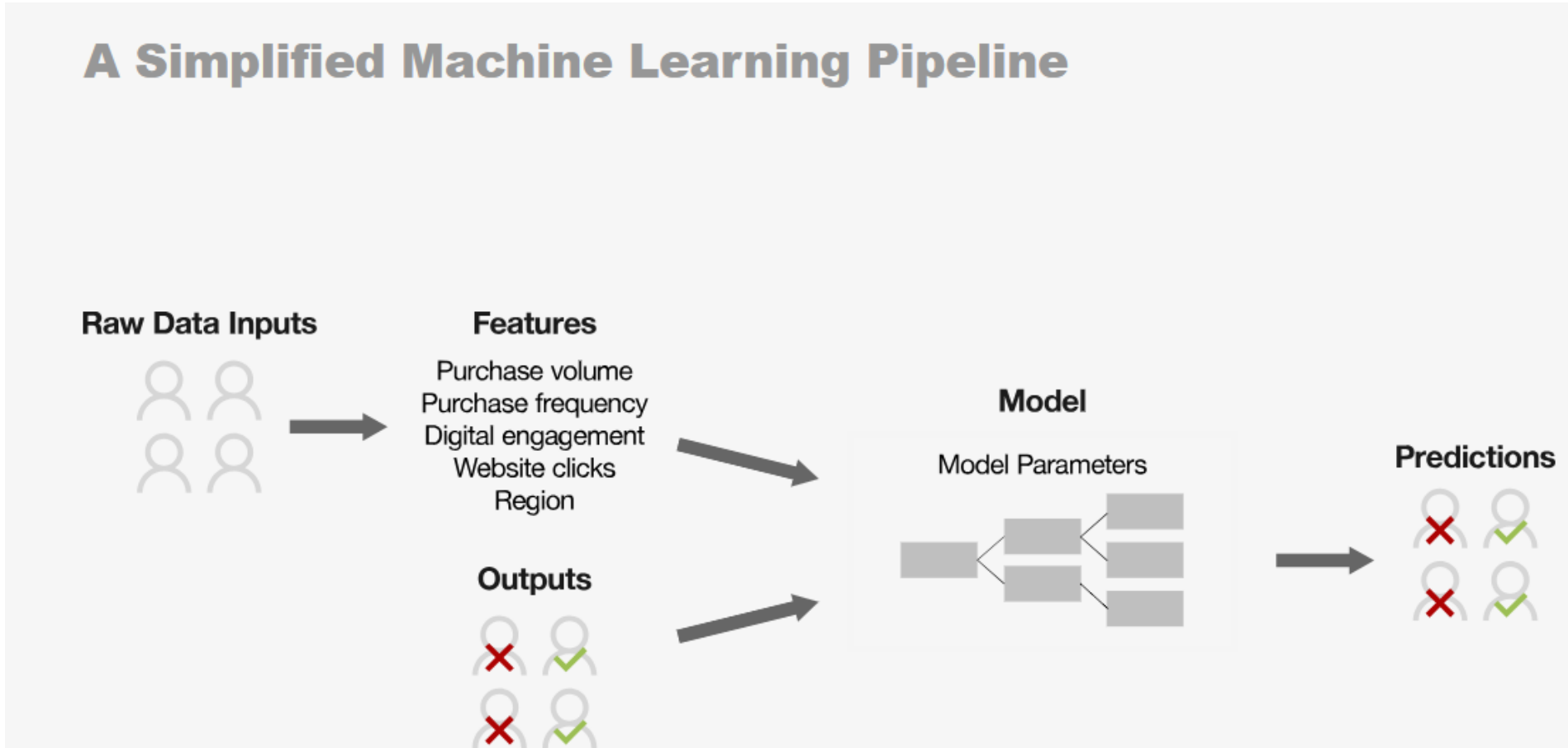
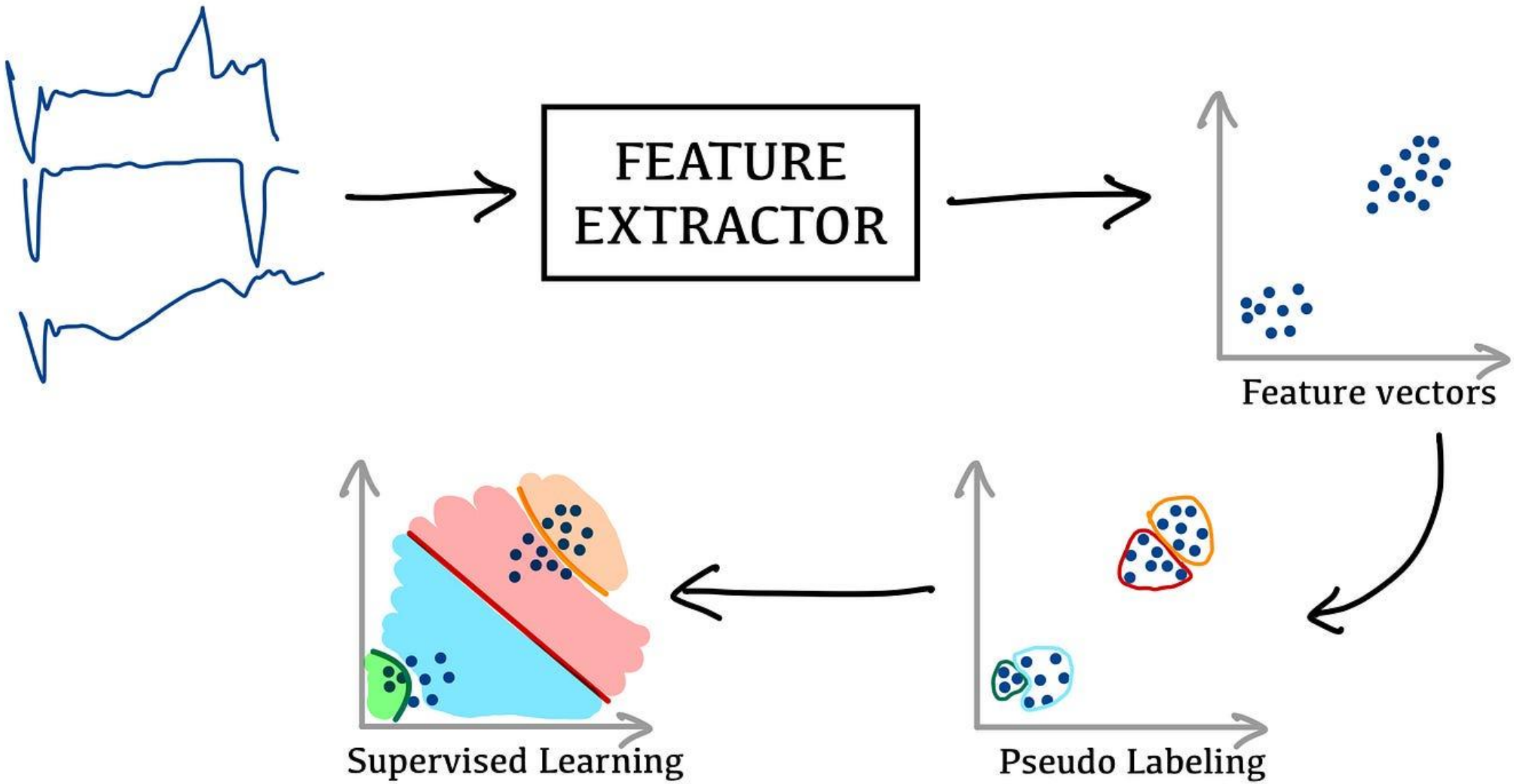


Figure 3 A supervised machine learning pipeline including raw data input, features, outputs, the ML model and model parameters, and prediction outputs. In this example, the machine learning model is trained to classify whether a customer will remain or leave.


FEATURE
EXTRACTOR
Feature vectors
Pseudo Labeling
Supervised Learning

Model Features

- Date
- Customer ID
- Customer Purchases
- Website Traffic

Outputs or Labels

- Historical Customer Attrition (Yes/No Classification)

ML Model

Model Predictions

Customer Attrition (Yes/No Classification)

Model Features, X (m columns)				Outputs, Y
Date	Customer ID	Purchases Past 30 Days ($)	Change in Web Traffic Rolling 30 Days (Visits)	Customer Attrition (Unsubscribed?)
1-Jan	Adams	1,000.00	+1	0
1-Jan	Bhat	-	0	0
1-Jan	Chu	-	-4	1
1-Feb	Adams	200.00	0	0
1-Feb	Bhat	-	-1	1
1-Feb	Chu	400.00	-1	0
1-Mar	Adams	1,000.00	+1	0
1-Mar	Bhat	650.00	+3	0

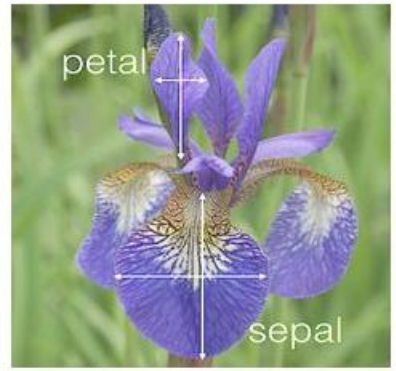
Data points form matrix of *n* rows

The Iris Flower Dataset

Iris setosa

Iris versicolor

Iris virginica



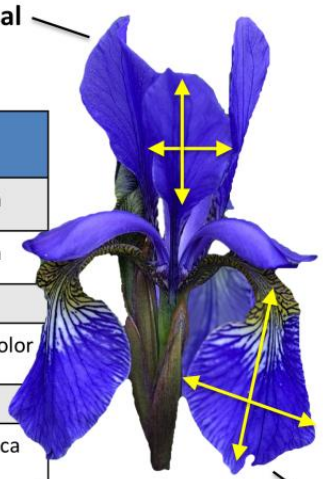
Sepal /Petal

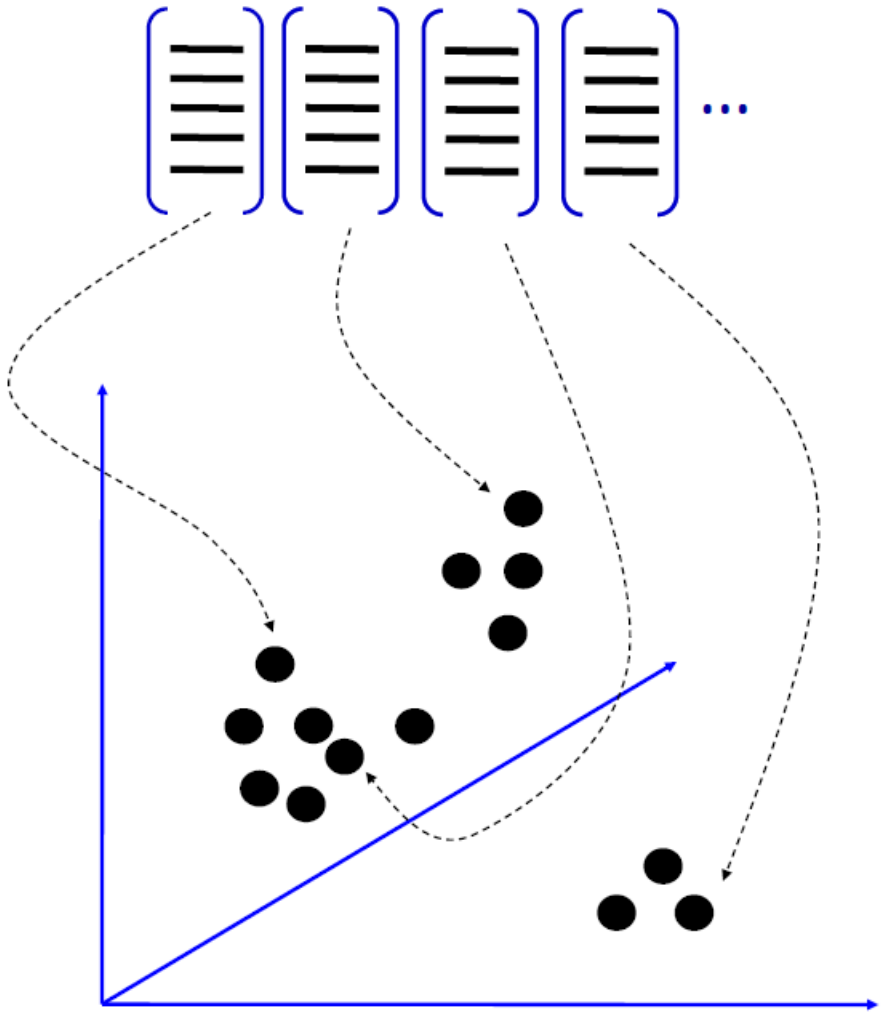
Samples
(instances, observations)

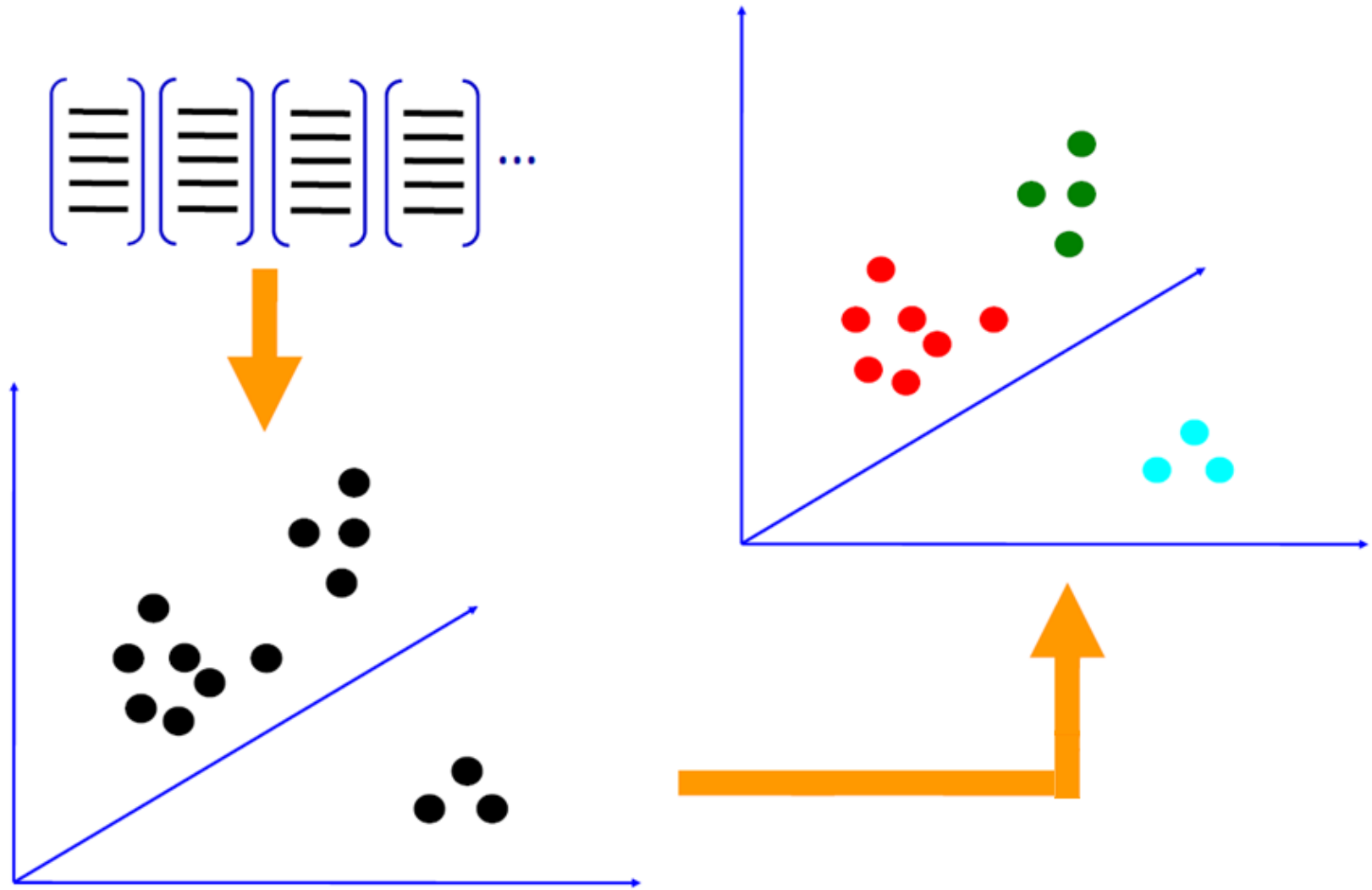
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

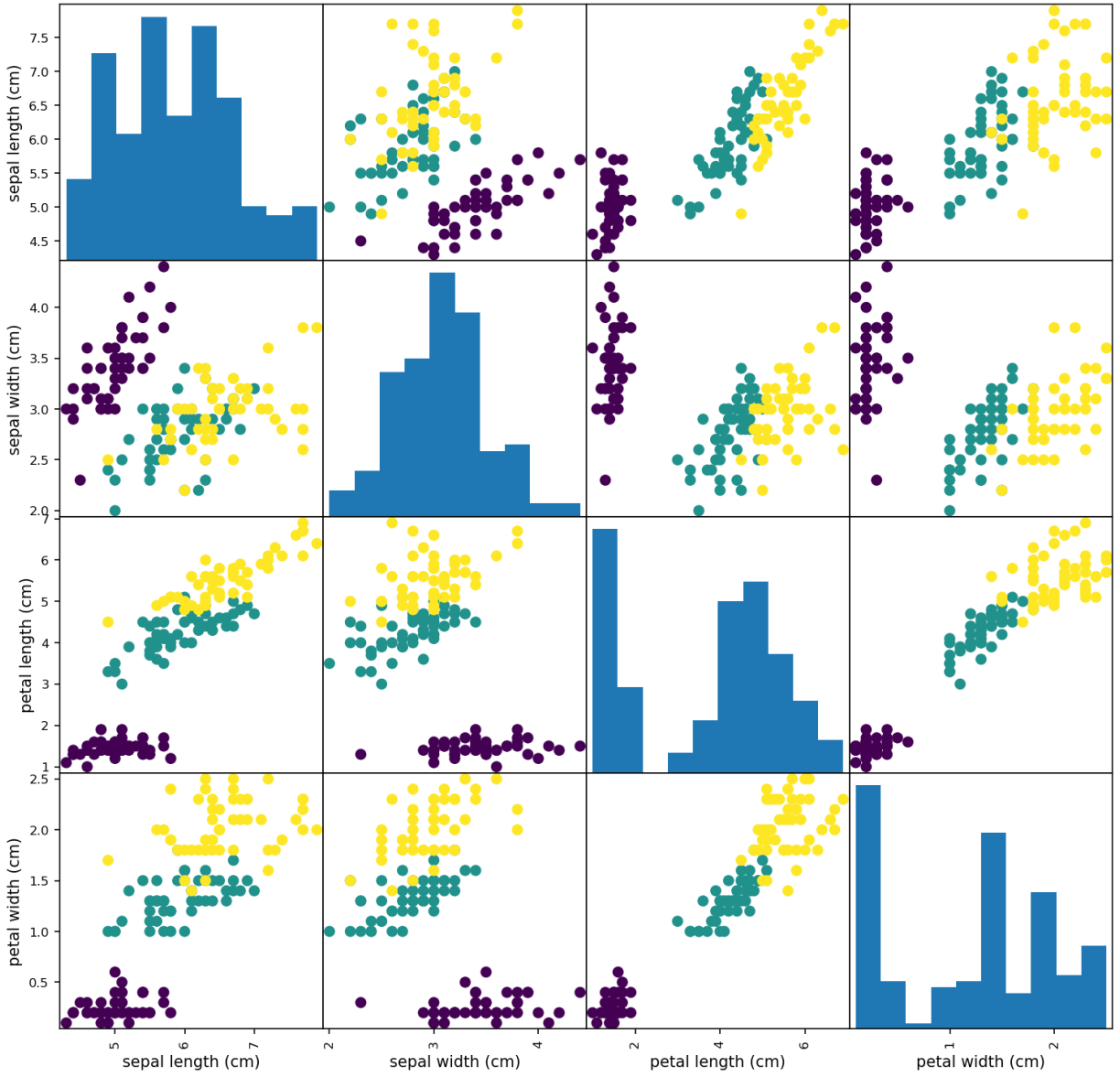
Class labels
(targets)



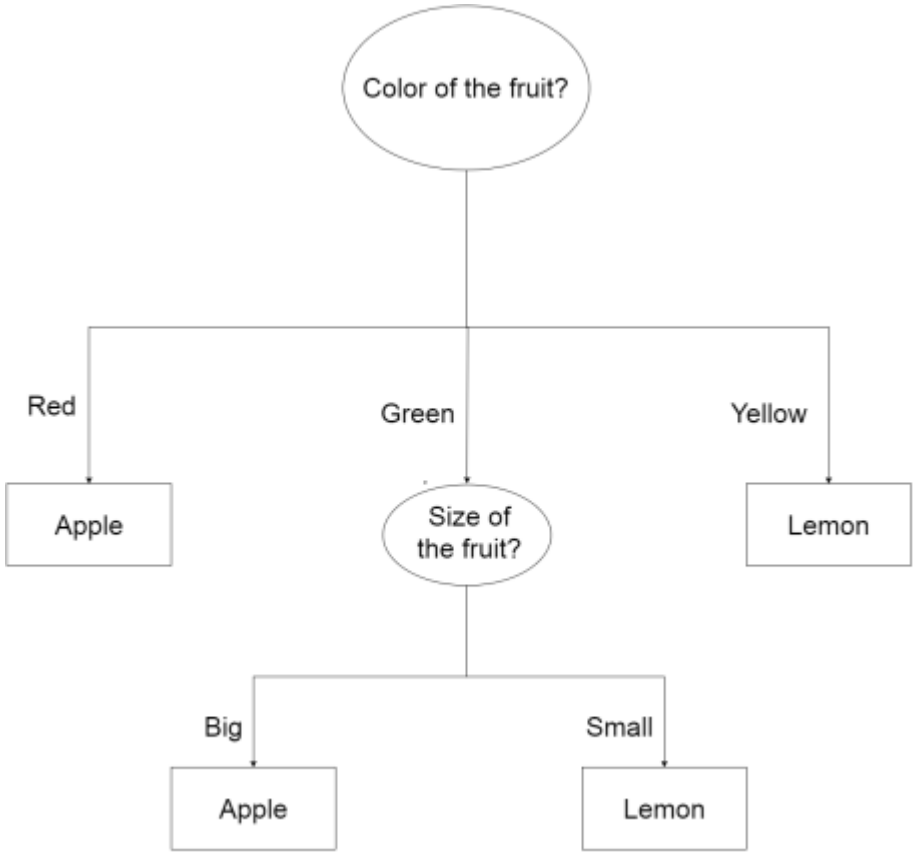


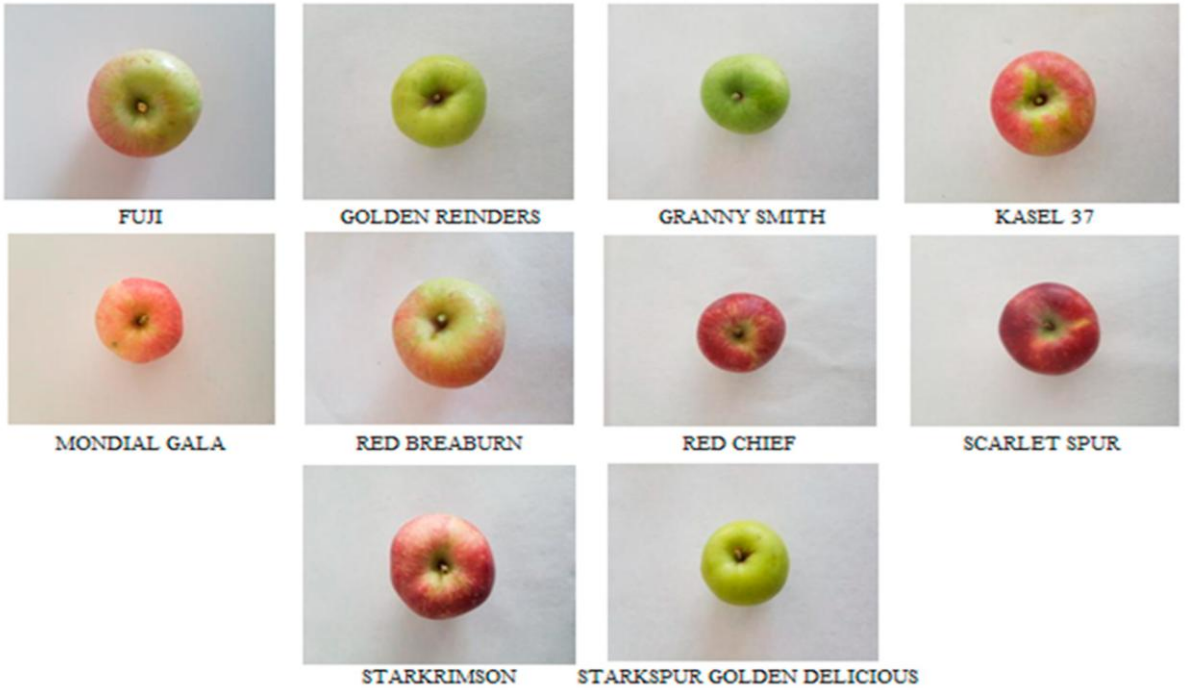


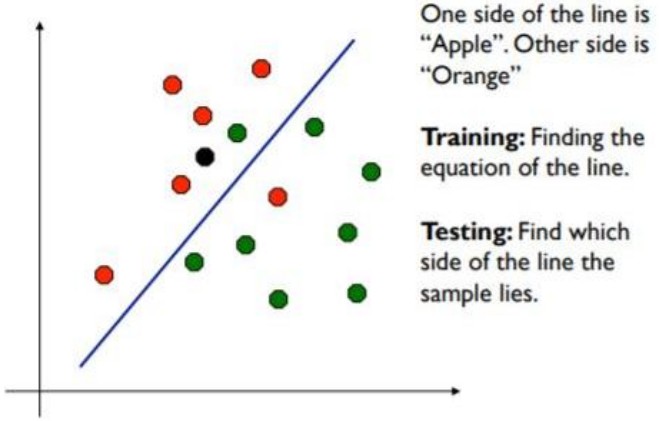
Feature Extraction



Apple vs Lemon





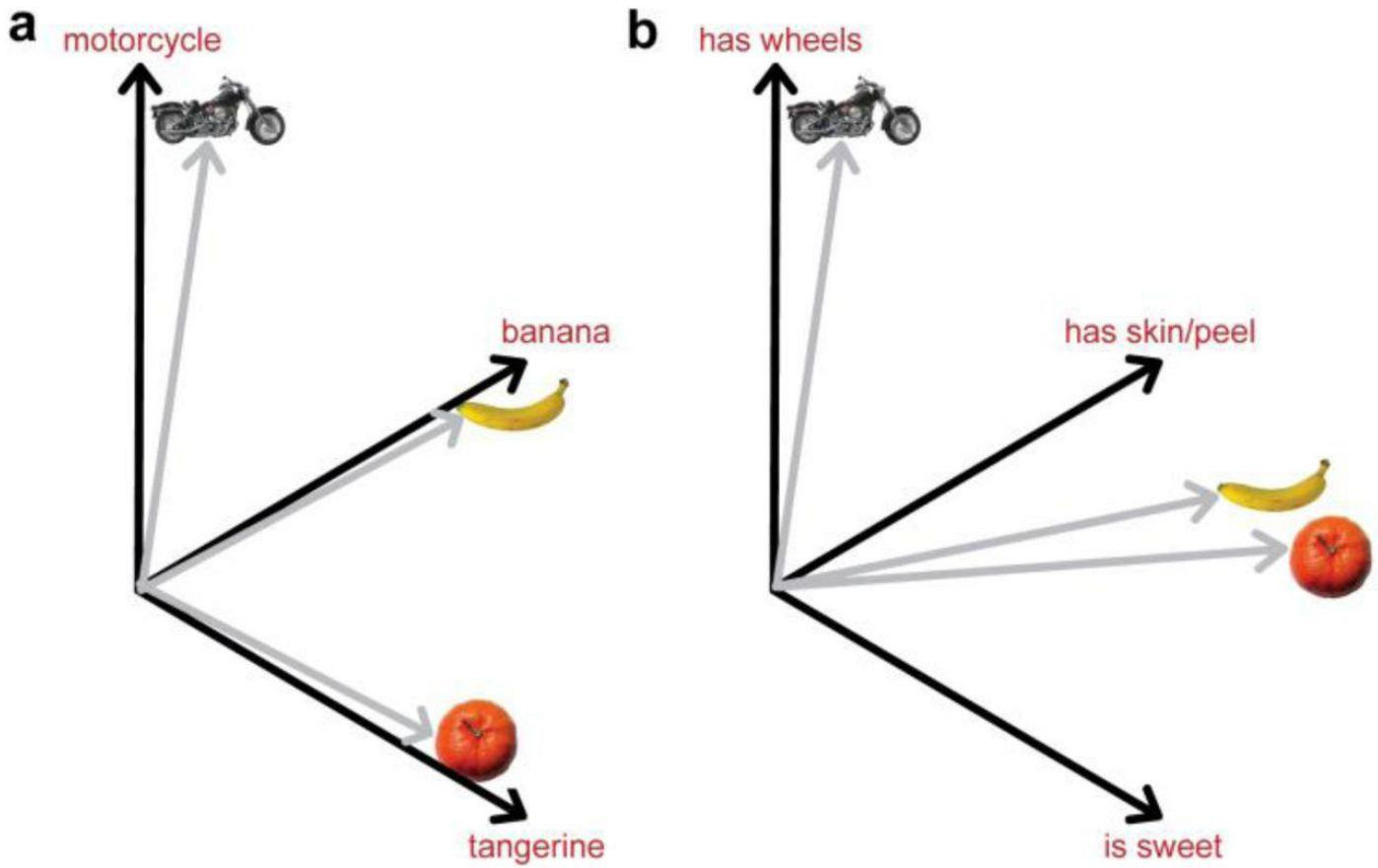


One side of the line is "Apple". Other side is "Orange"

Training: Finding the equation of the line.

Testing: Find which side of the line the sample lies.

Feature Extraction



Feature Extraction

A technique used in machine learning and data analysis to identify and extract relevant information or patterns from raw data to produce a more concise dataset.

Machine Learning:

Sample

Label

dog

cat

horse

Human Learning:

We learn through

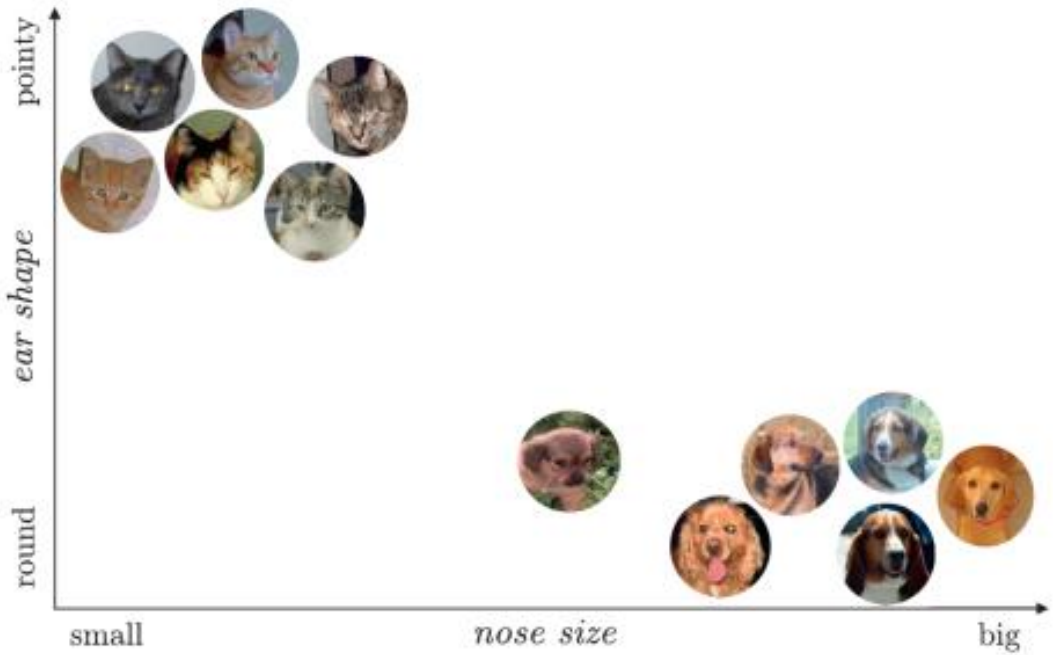
Examples

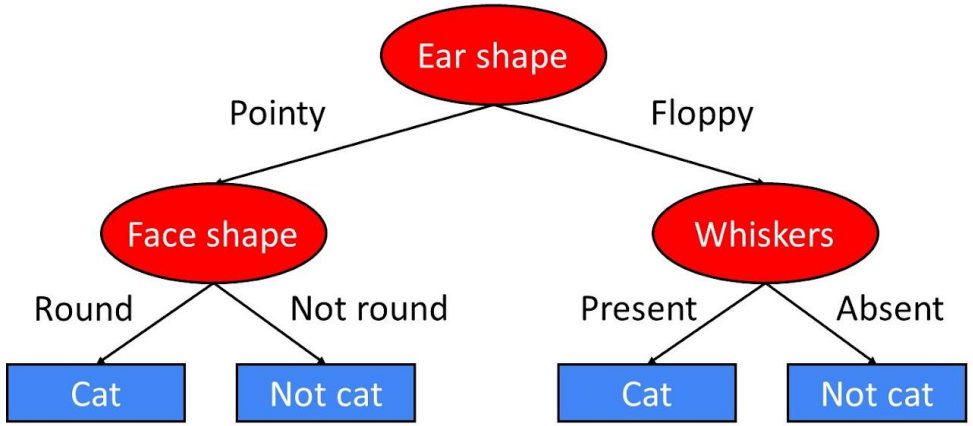
Long Ear Black nose

dog

Diagrams

Comparisons

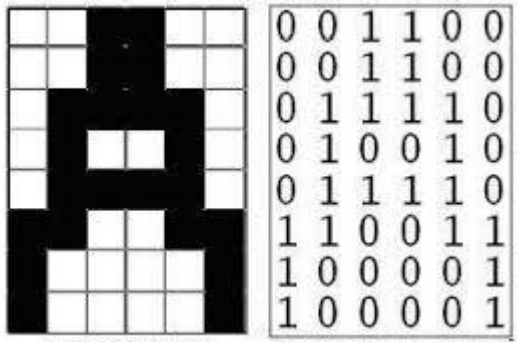




Ear shape: Pointy

Face shape: Round

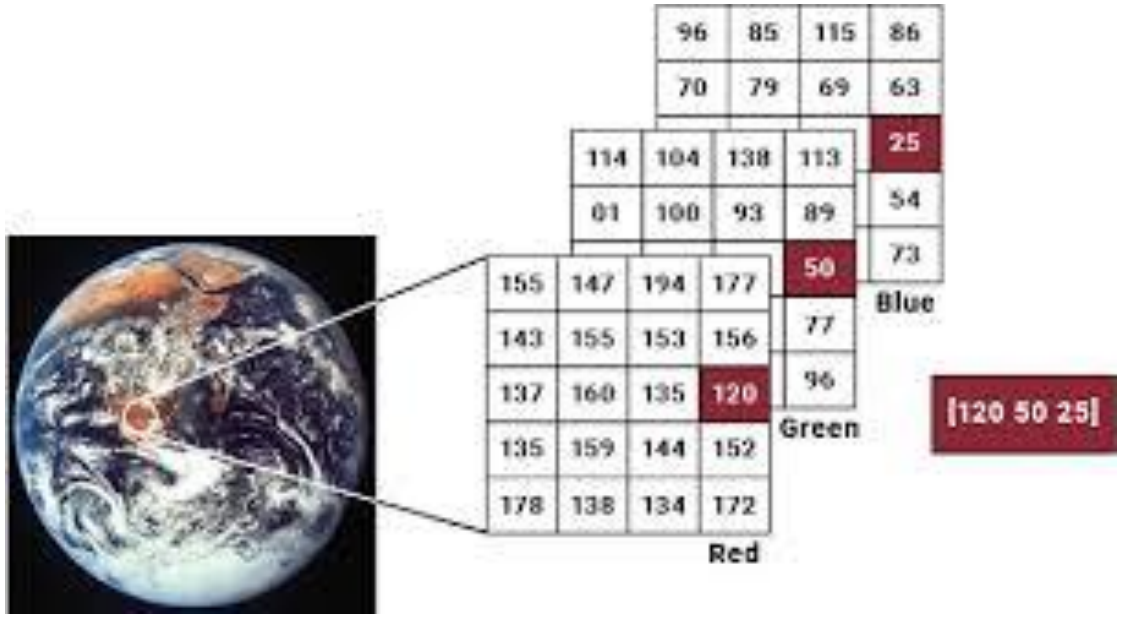
Whiskers: Present

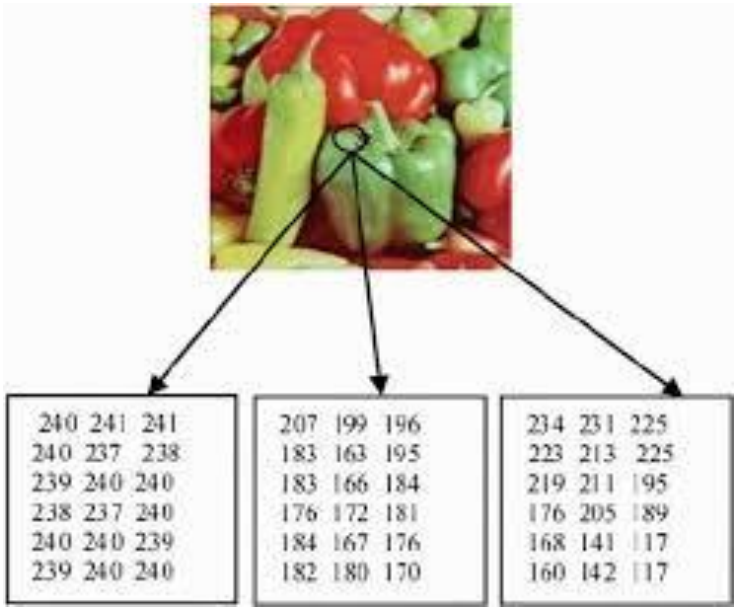


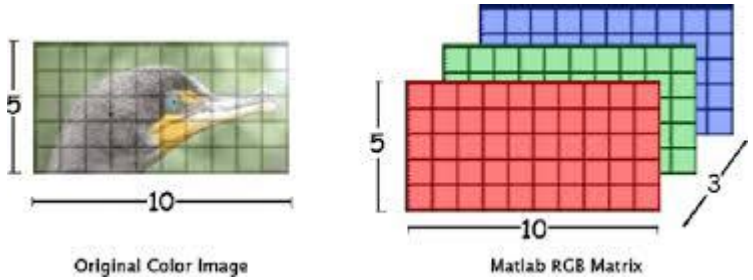
Digitized "A"

0	0	1	1	0	0
0	0	1	1	0	0
0	1	1	1	1	0
0	1	0	0	1	0
0	1	1	1	1	0
1	1	0	0	1	1
1	0	0	0	0	1
1	0	0	0	0	1

Matrix Form of "A"







Original Color Image

Matlab RGB Matrix

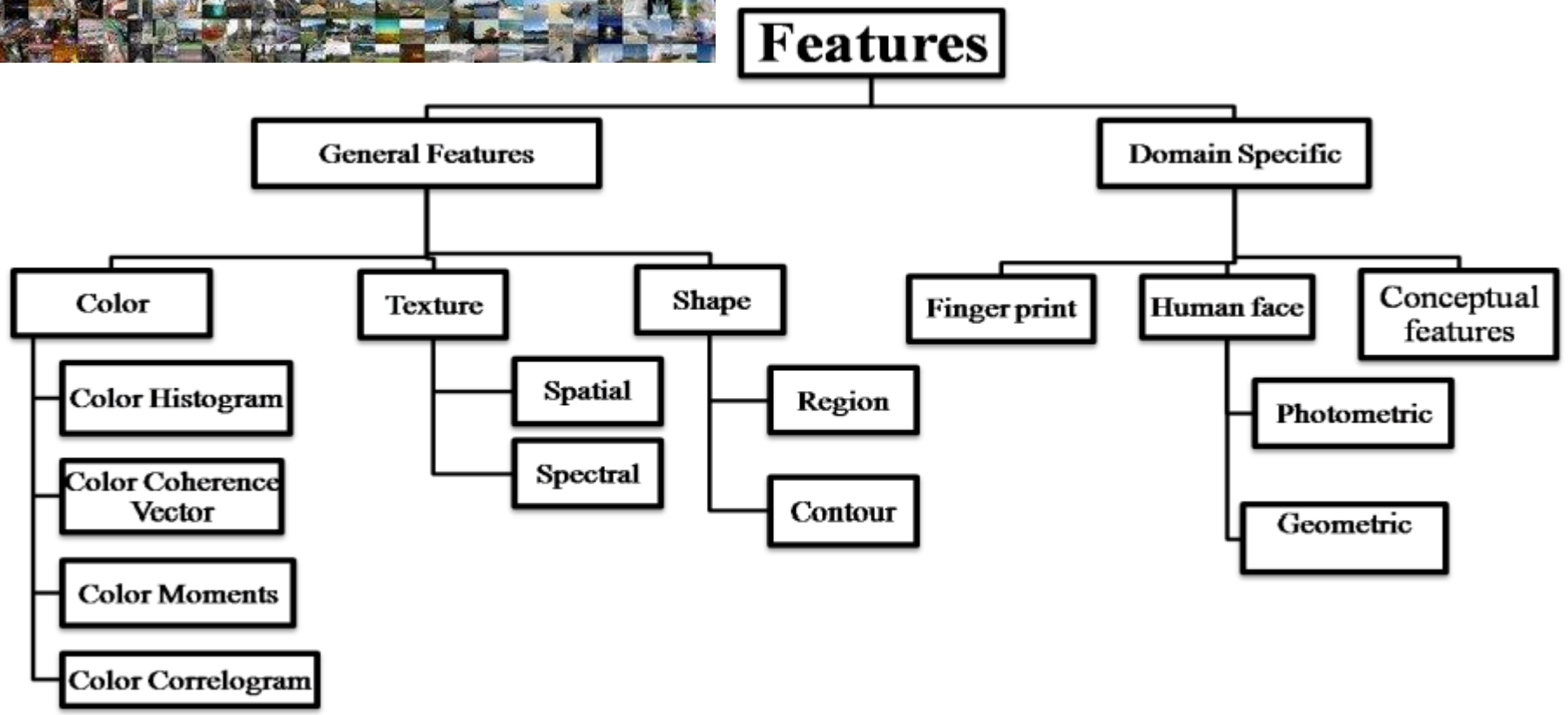
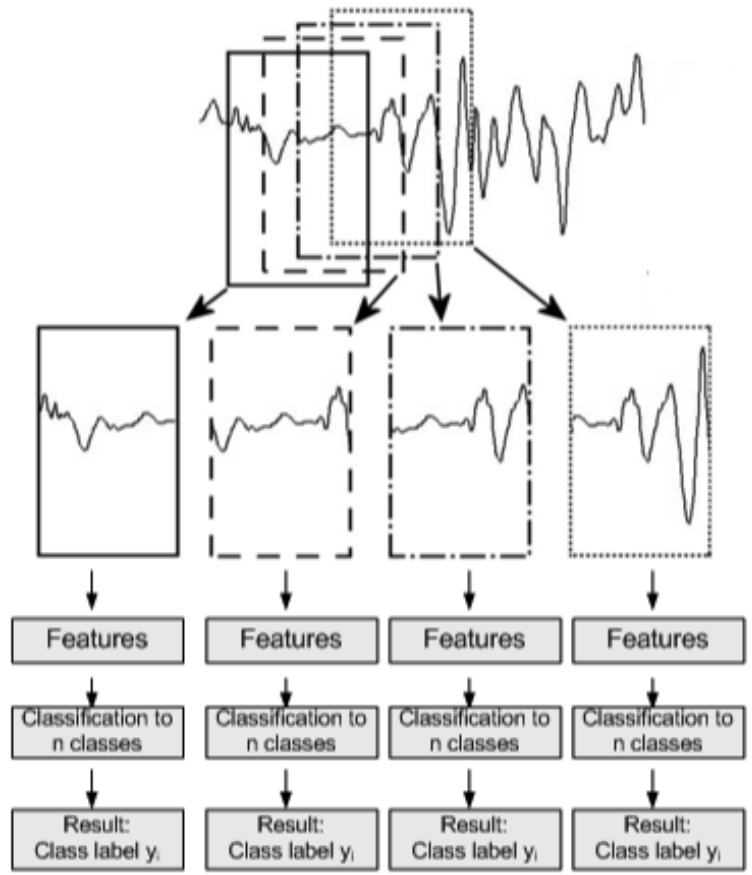
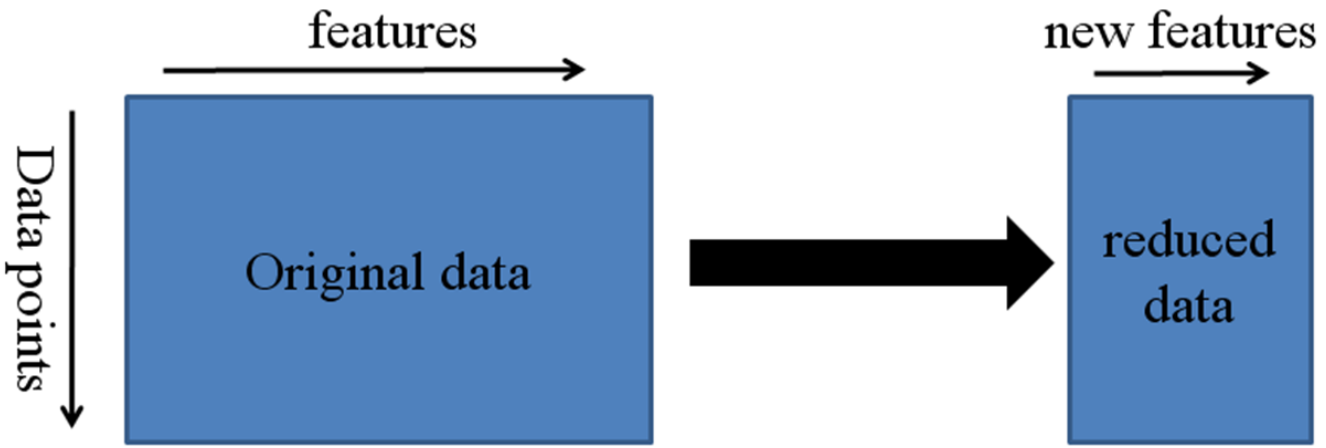


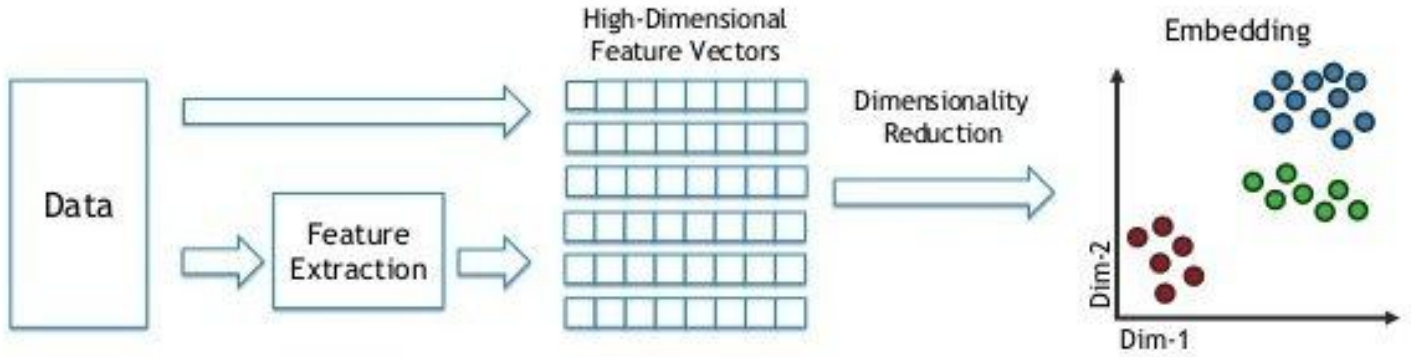
Fig. 1. Classification of feature extraction techniques.


Features
Features
Features
Features
Classification to n classes
Classification to n classes
Classification to n classes
Classification to n classes
Result: Class label y_i
Result: Class label y_i
Result: Class label y_i
Result: Class label y_i

Dimensionality Reduction

Dimensionality Reduction





Dimensionality Reduction

Feature Elimination: we reduce the feature space by elimination feature. The advantages of the feature elimination method include simplicity and maintainability features. We've also eliminated any benefits those dropped variables would bring.

Feature Selection: It combines our input variables in a specific way, then we can drop the "least important" variables while still retaining the most valuable parts of all the variables.

Dimensionality Reduction

Advantages:

- reduces storage space and computation time;
- remove redundant features;
- fastens the time required for performing same computations;
- fewer dimensions then it leads to less computing;
- allow usage of algorithms unfit for a large number of dimensions;
- helps generalize models.

Dimensionality Reduction

- PCA: Principal component analysis
- LDA: Linear discriminant analysis
- t-SNE: t-distributed stochastic neighbor embedding

PCA

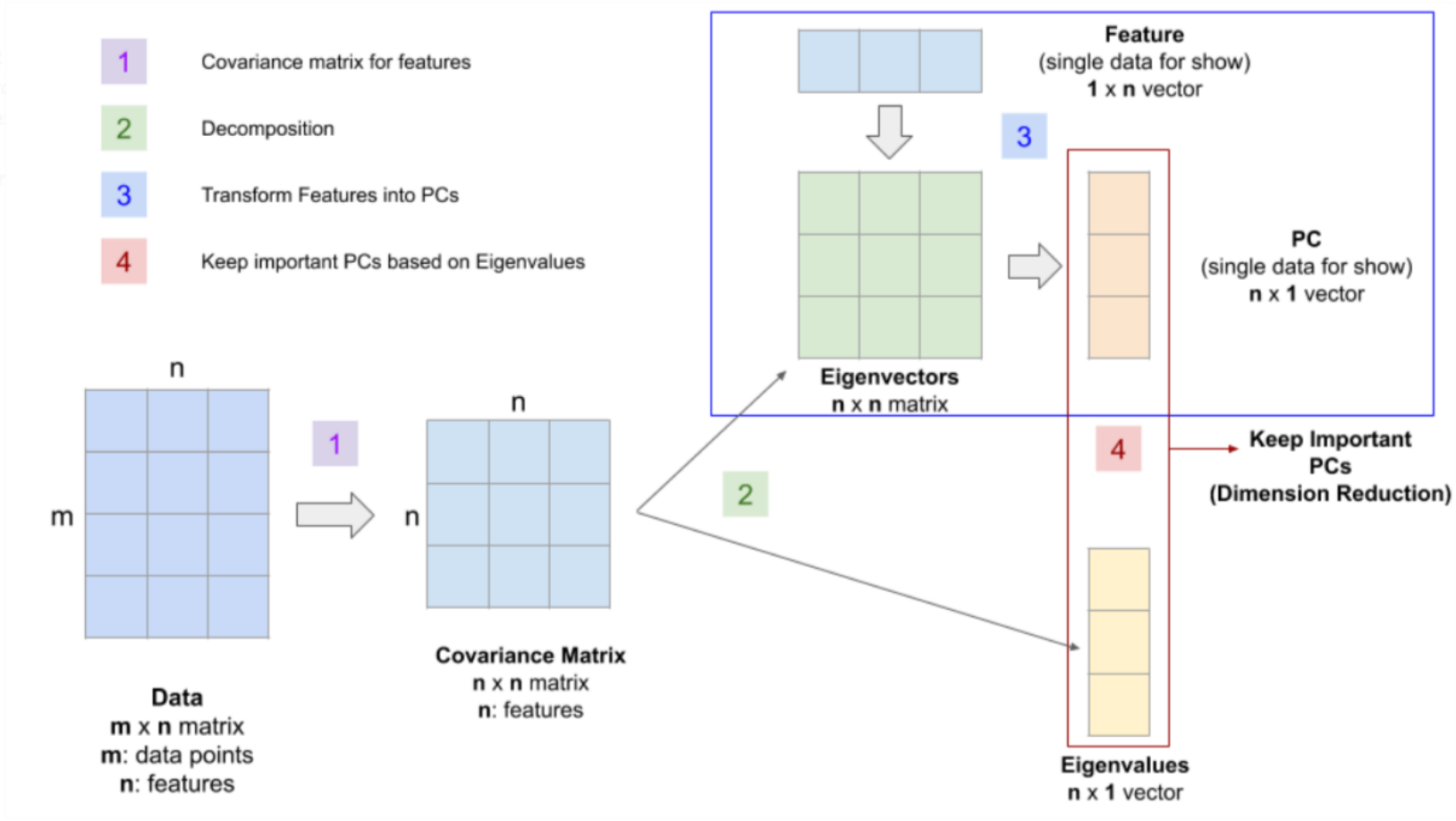
When should I use PCA?

1. Do you want to reduce the no. of variables, but are not able to identify variables to completely remove from consideration?

2. Do you want to ensure your variables are independent of one another?

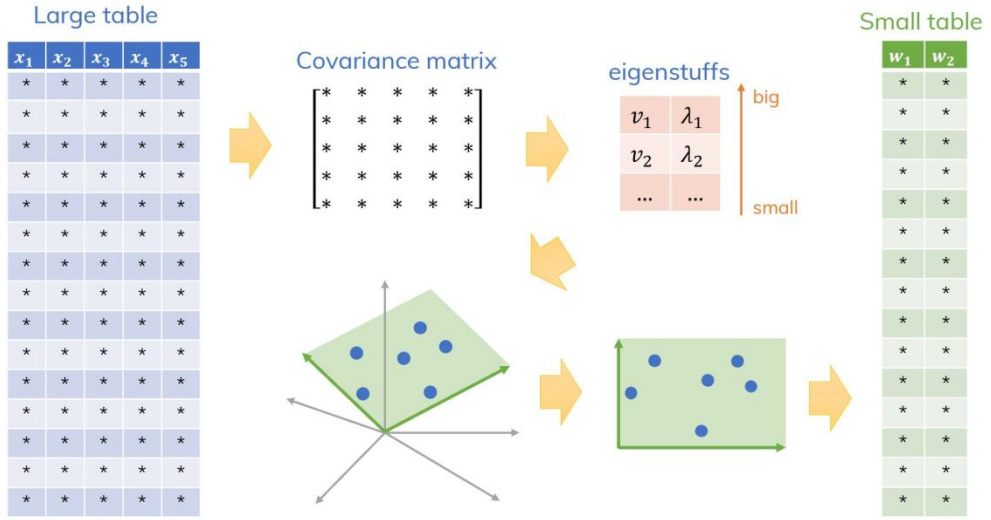
3. Are you comfortable making your independent variable less interpretable?

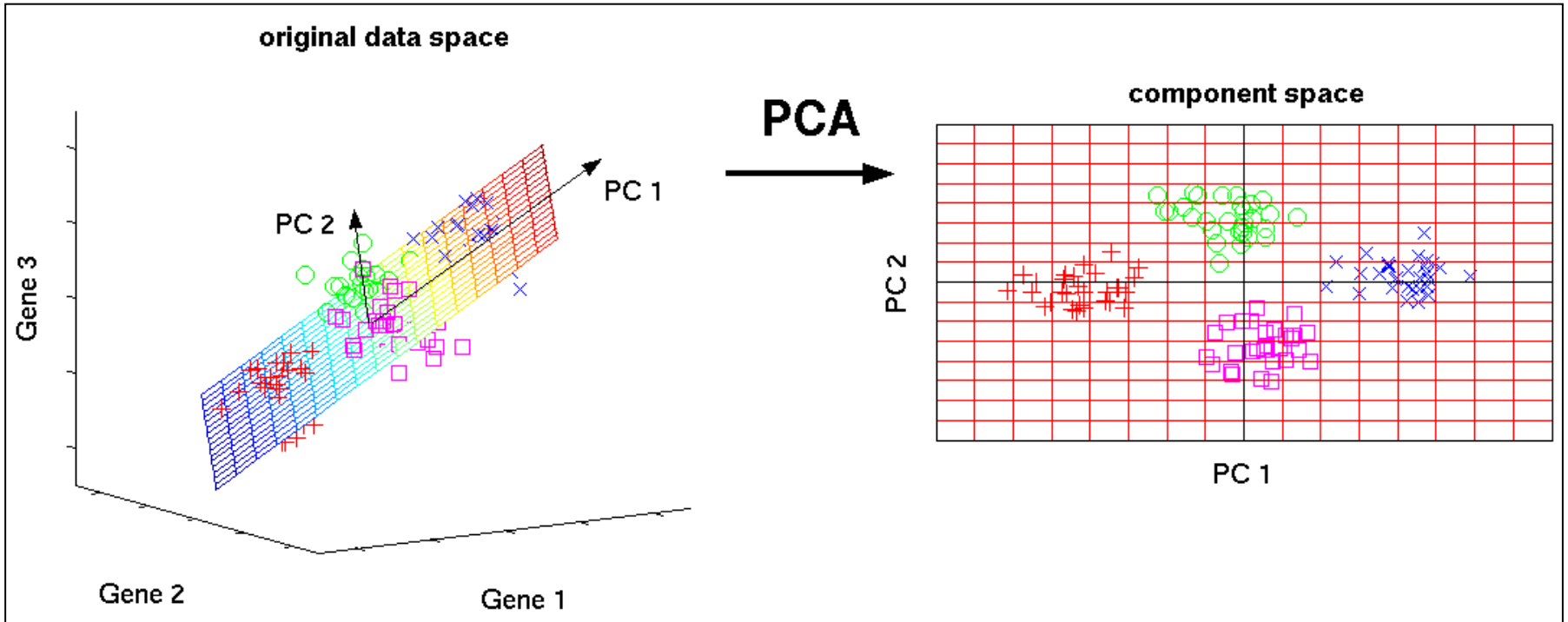
PCA



To remember this definition, we can break it down into four steps:
1 - We identify the relationship among features through a **Covariance Matrix**.
2 - Through the linear transformation or **eigendecomposition** of the Covariance Matrix, we get eigenvectors and eigenvalues.
3 - Then we transform our data using Eigenvectors into **principal components**.
4 - Lastly, we quantify the importance of these relationships using Eigenvalues **and keep the important principal components**.

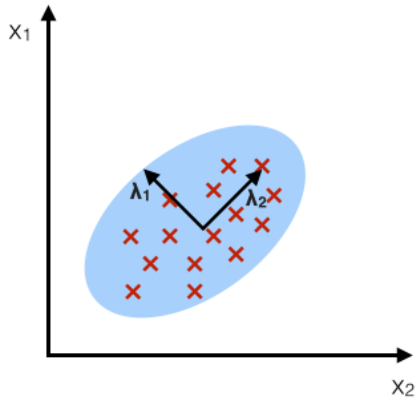
PCA


Large table
x_1 x_2 x_3 x_4 x_5
Covariance matrix
eigenstuffs
v_1 λ_1
v_2 λ_2
big
small
Small table
w_1 w_2

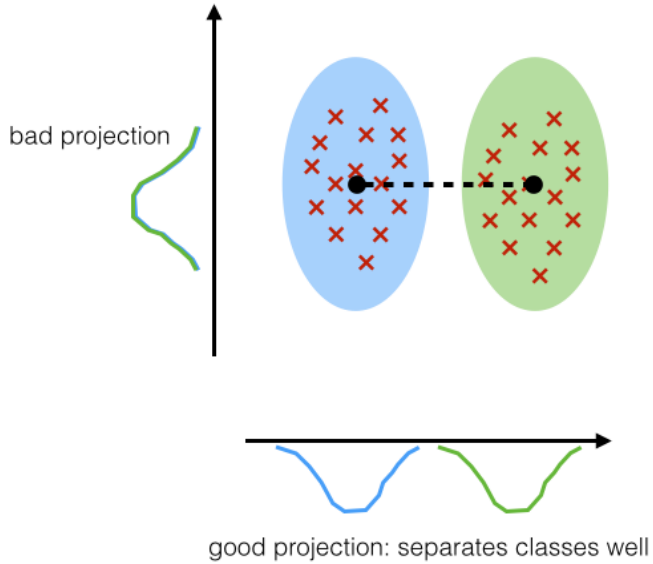

original data space
PC 1
PC 2
Gene 3
Gene 2
Gene 1
PCA
component space
PC 2
PC 1

LDA

PCA:
component axes that maximize the variance

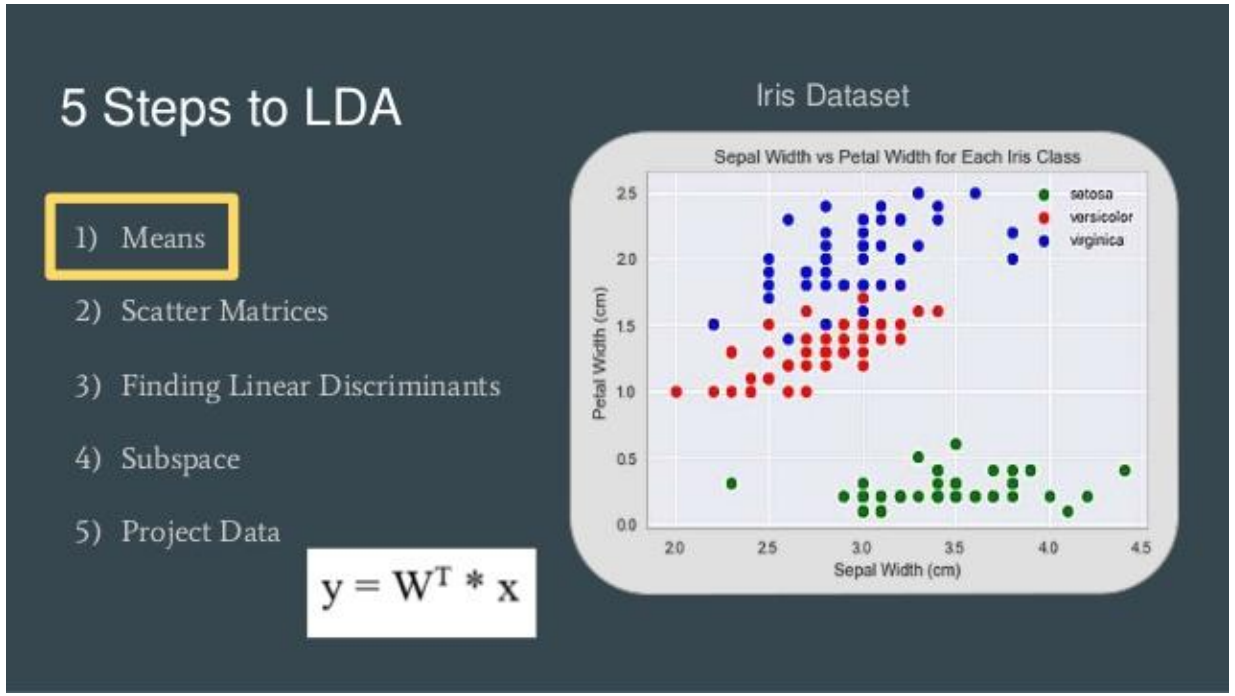


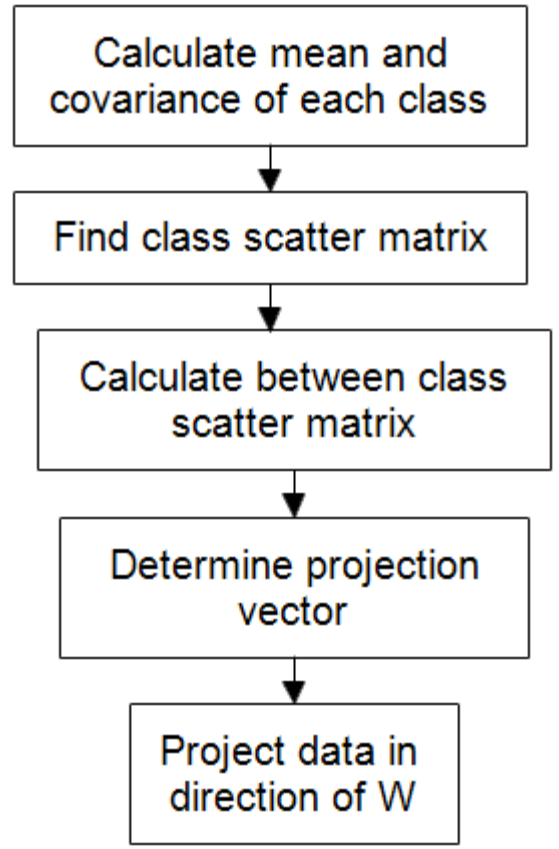
LDA:
maximizing the component axes for class-separation



PCA projects the entire dataset onto a different feature (sub)space, and LDA tries to determine a suitable feature (sub)space in order to distinguish between patterns that belong to different classes.

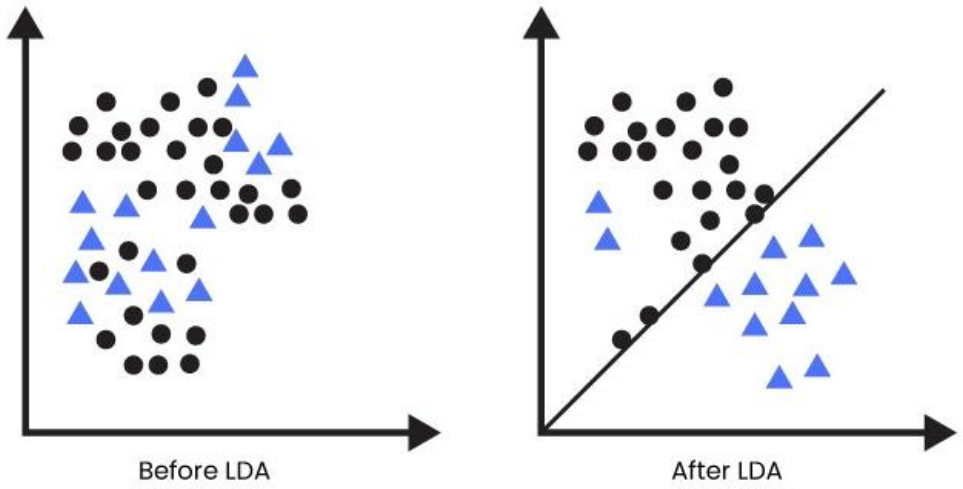
LDA

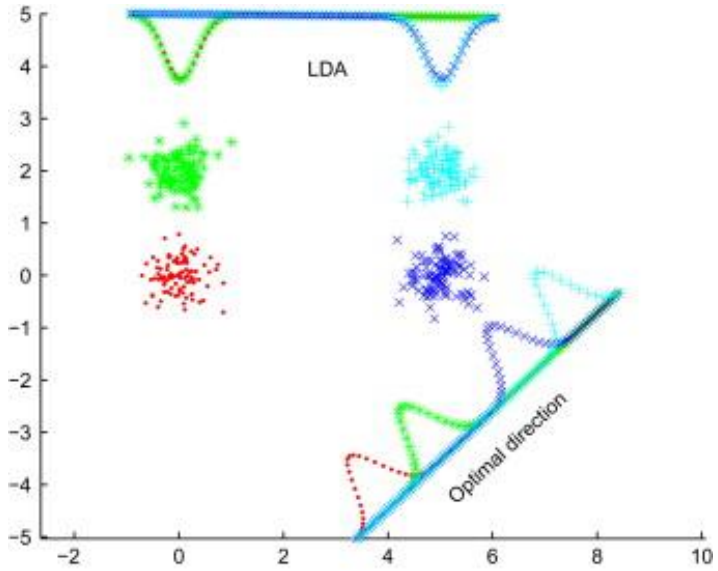


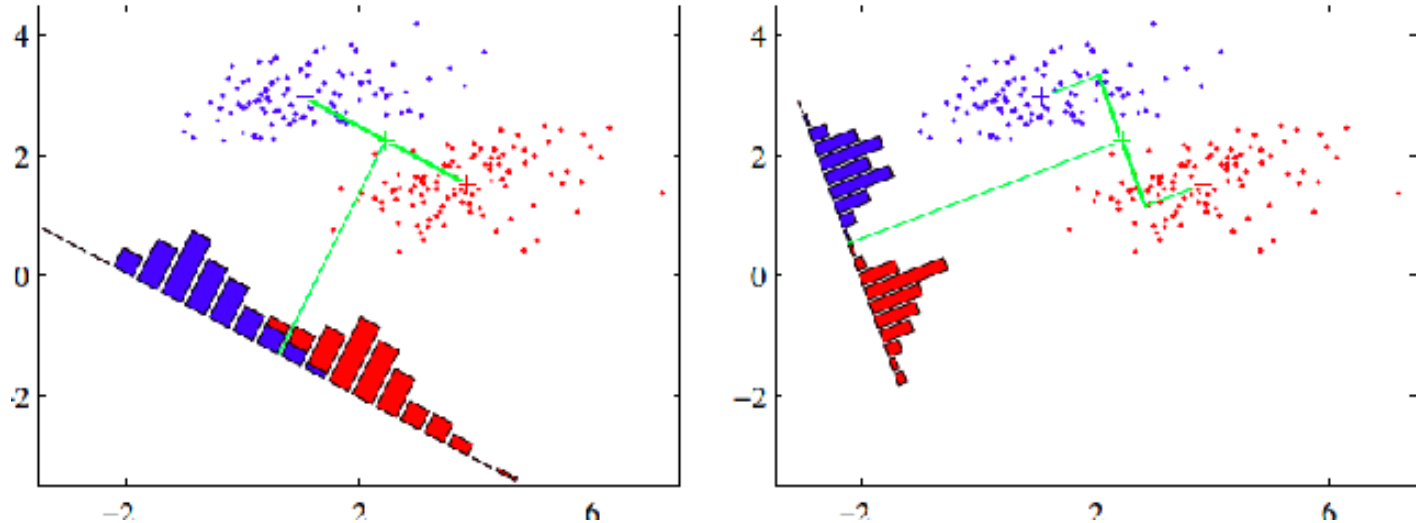


PCA projects the entire dataset onto a different feature (sub)space, and LDA tries to determine a suitable feature (sub)space in order to distinguish between patterns that belong to different classes.

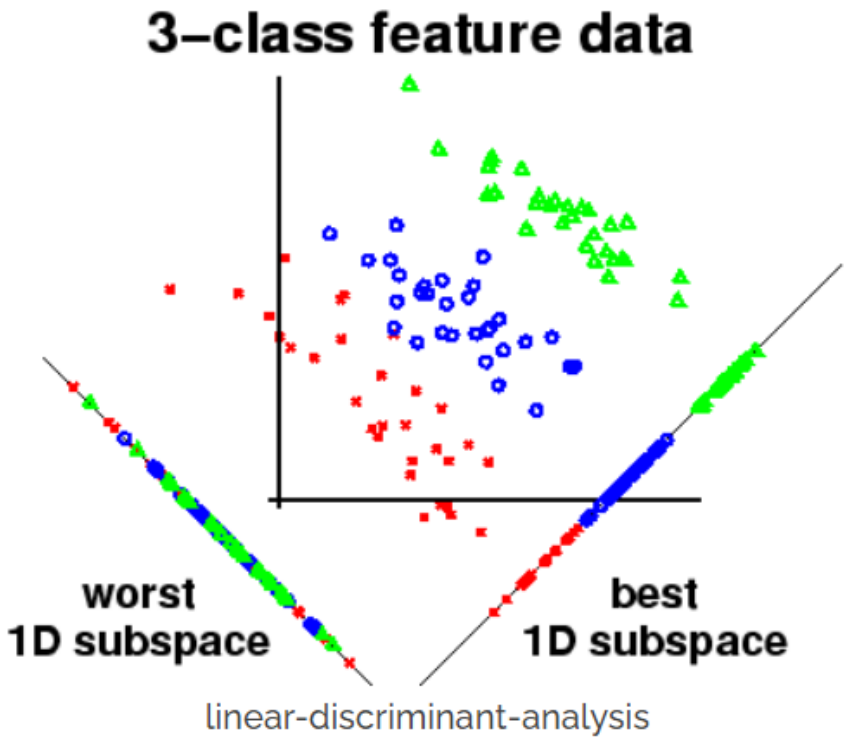
LDA

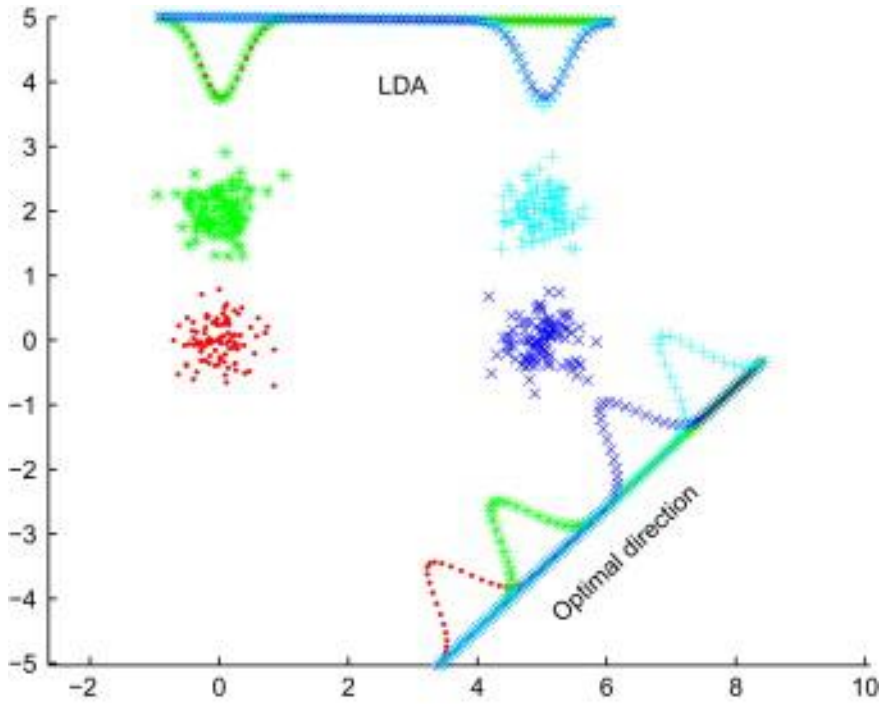




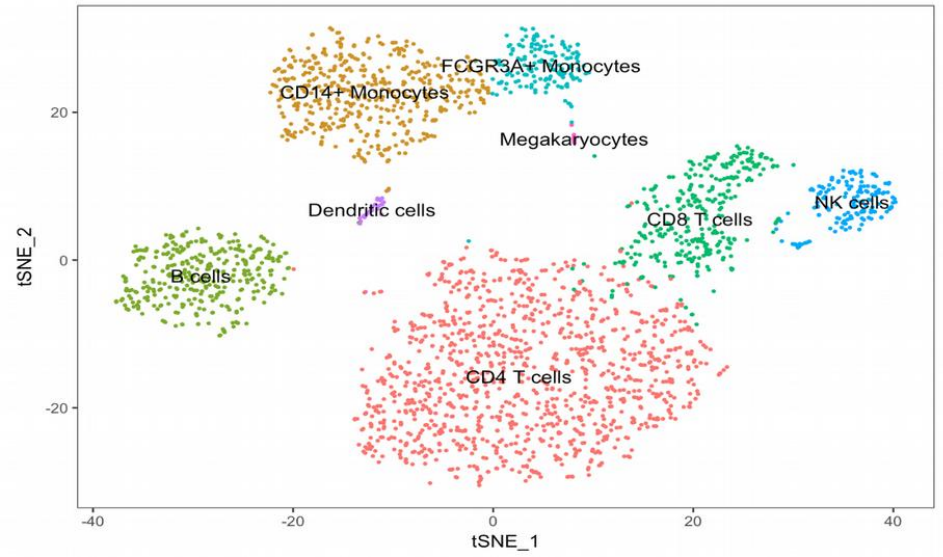


LDA

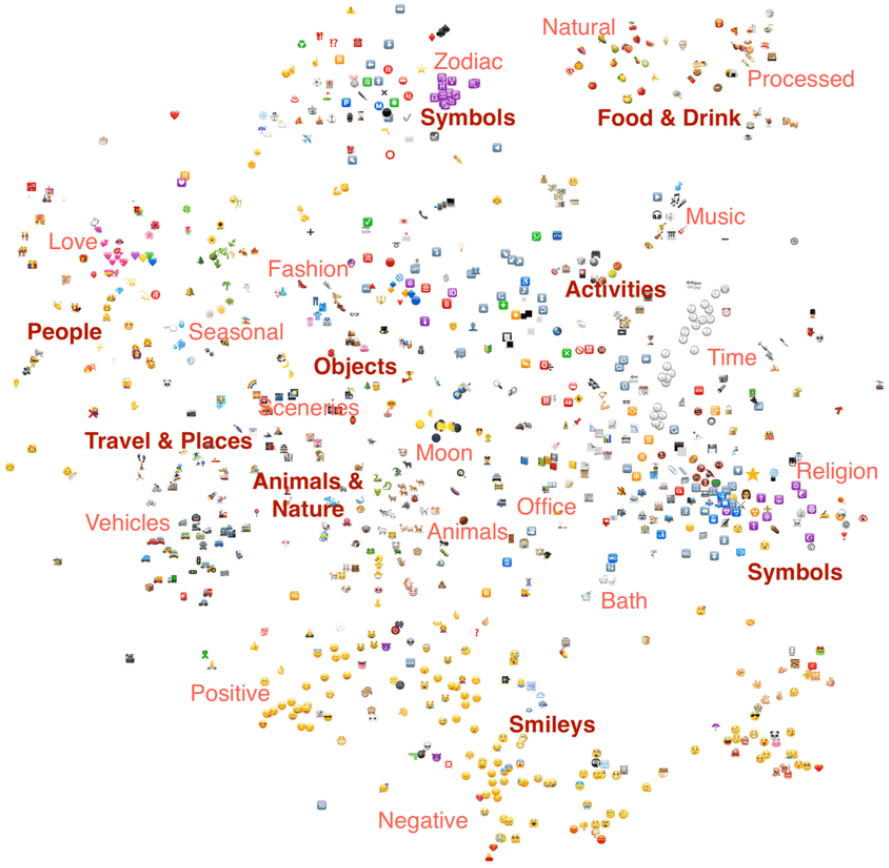

3–class feature data
worst
1D subspace
best
1D subspace
linear-discriminant-analysis


LDA
Optimal direction
5
4
3
2
1
0
−1
−2
−3
−4
−5
−2
0
2
4
6
8
10

t-SNE
t-Distributed Stochastic Neighbor Embedding



t-Distributed Stochastic Neighbor Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data.



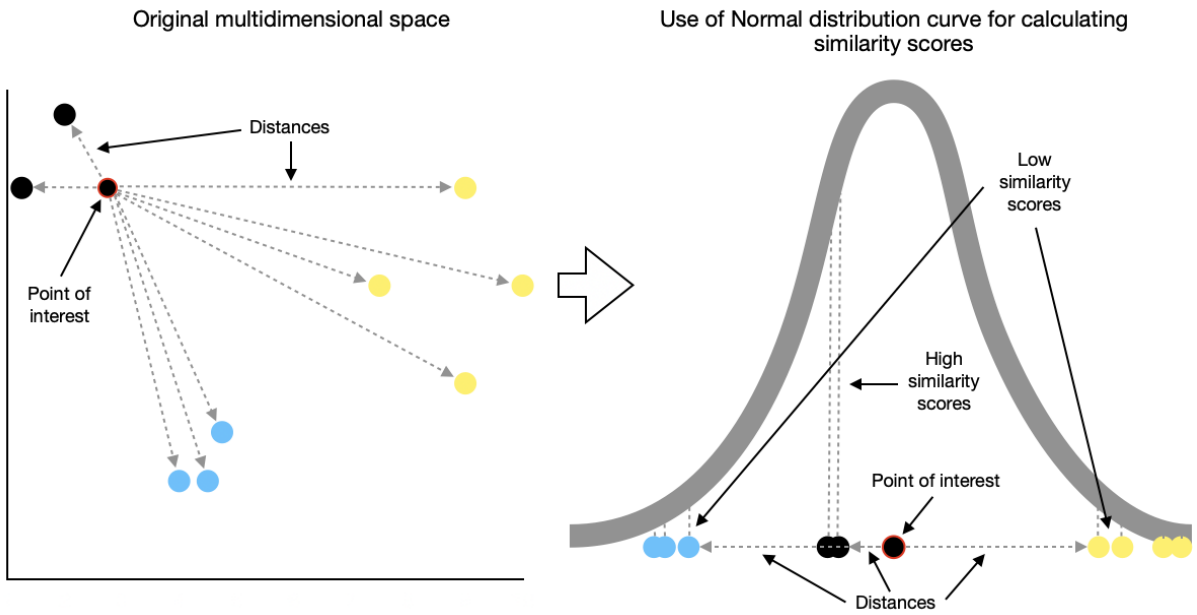
https://www.researchgate.net/publication/331828818_Exploring_Dis-Similarities_in_Emoji-Emotion_Association_on_Twitter_and_Weibo

t-SNE

- t-SNE tends to preserve local struture at the same time preserving the global strututre as much as possible;

- The t-SNE algorithm calculates a **similarity measure between pairs of instances in the high dimensional space and in the low dimensional space**.

- Step 1, **measure similarities between points in the high dimensional space**. Think of a bunch of data points scattered on a 2D space. For each data point (xi) we'll center a Gaussian distribution over that point.



t-SNE

Step 2, It then tries to **minimize the difference between these conditional probabilities** (or similarities) in higher-dimensional and lower-dimensional space for a perfect representation of data points in lower-dimensional space. Instead of using a Gaussian distribution will be used a **Student t-distribution** with one degree of freedom, which is also known as the Cauchy distribution.

Step 3. The last step is that we want these set of probabilities from the low-dimensional space (Qij) to reflect those of the high dimensional space (Pij) as best as possible. **t-SNE creates low-dimensional space with the same number of points as in the original space. Points should be spread randomly on a new space. The goal of this algorithm is to find similar probability distribution in low-dimensional space.**

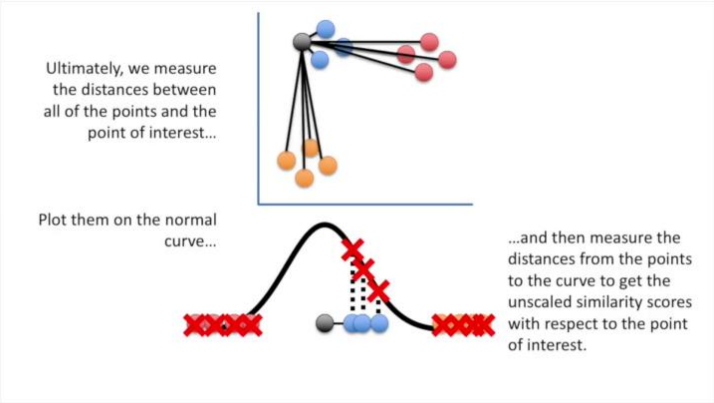
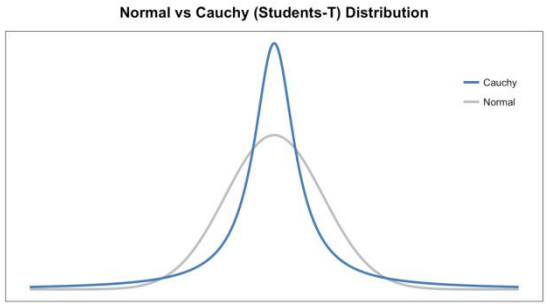
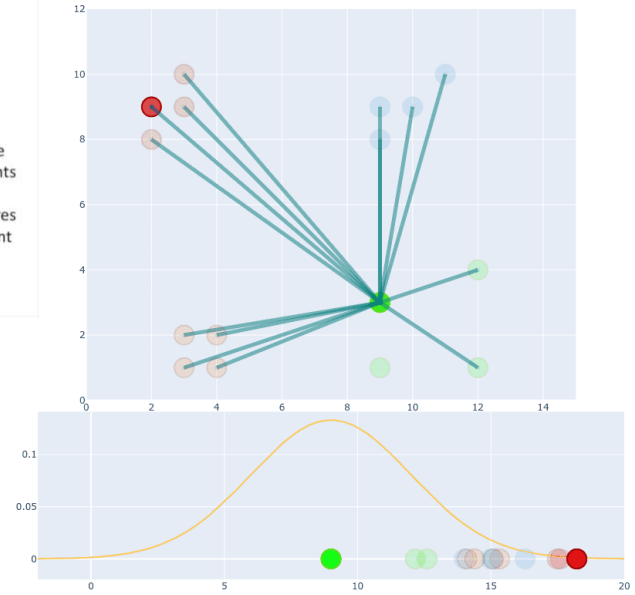
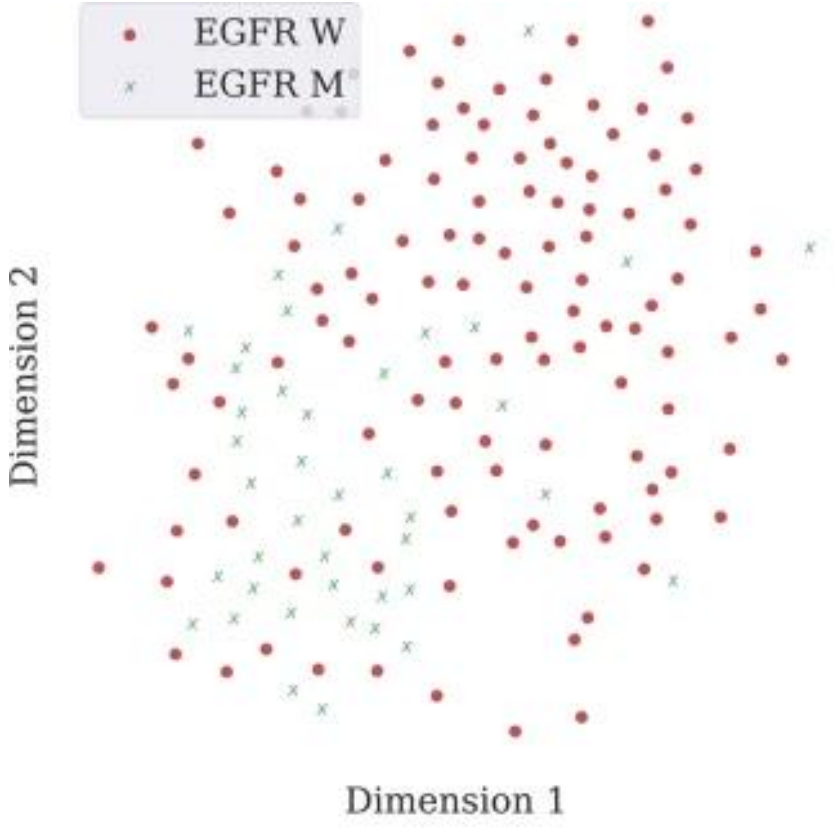


Figure 2 : Mapping of distances on Normal Curve

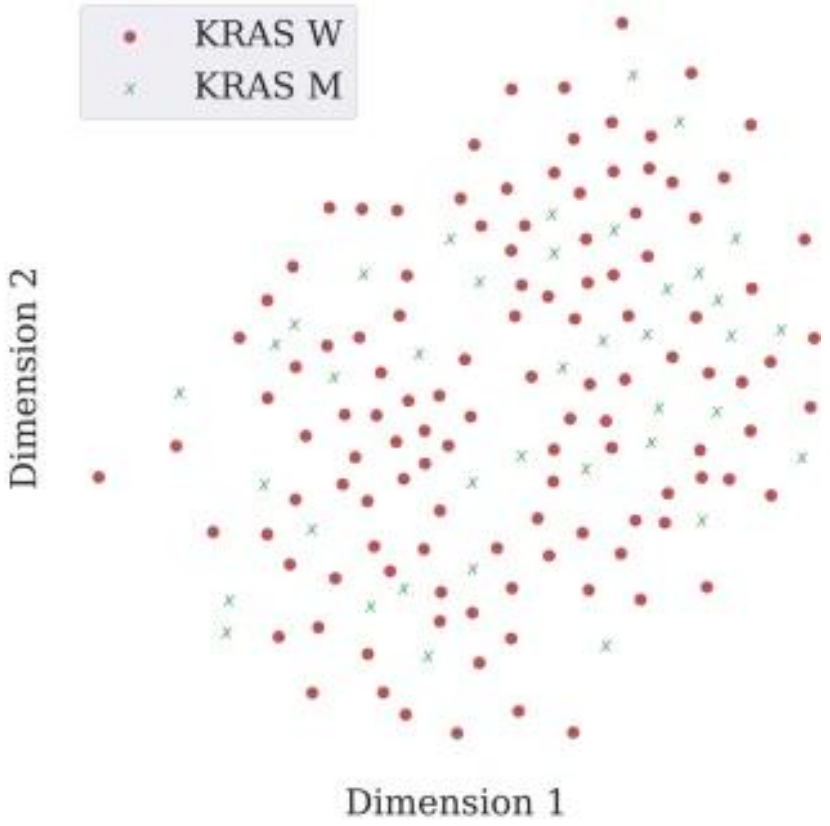




t-SNE Vizualization

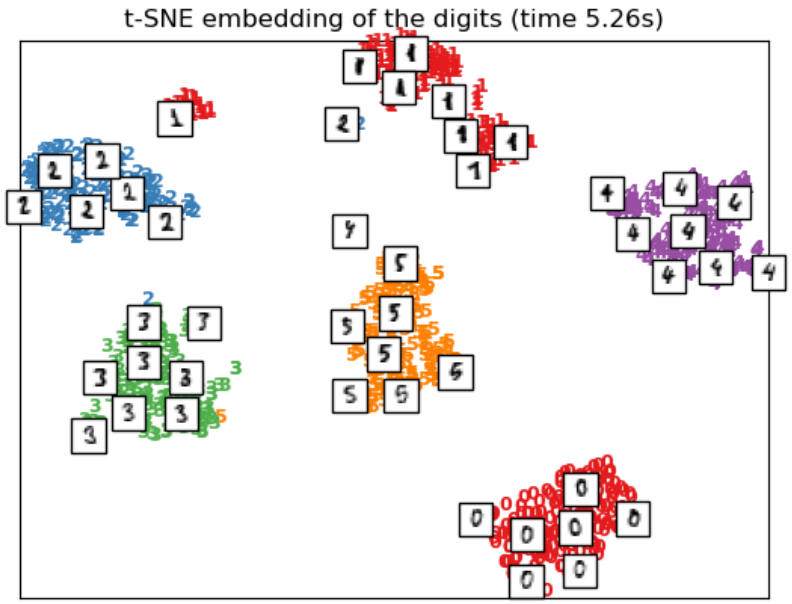


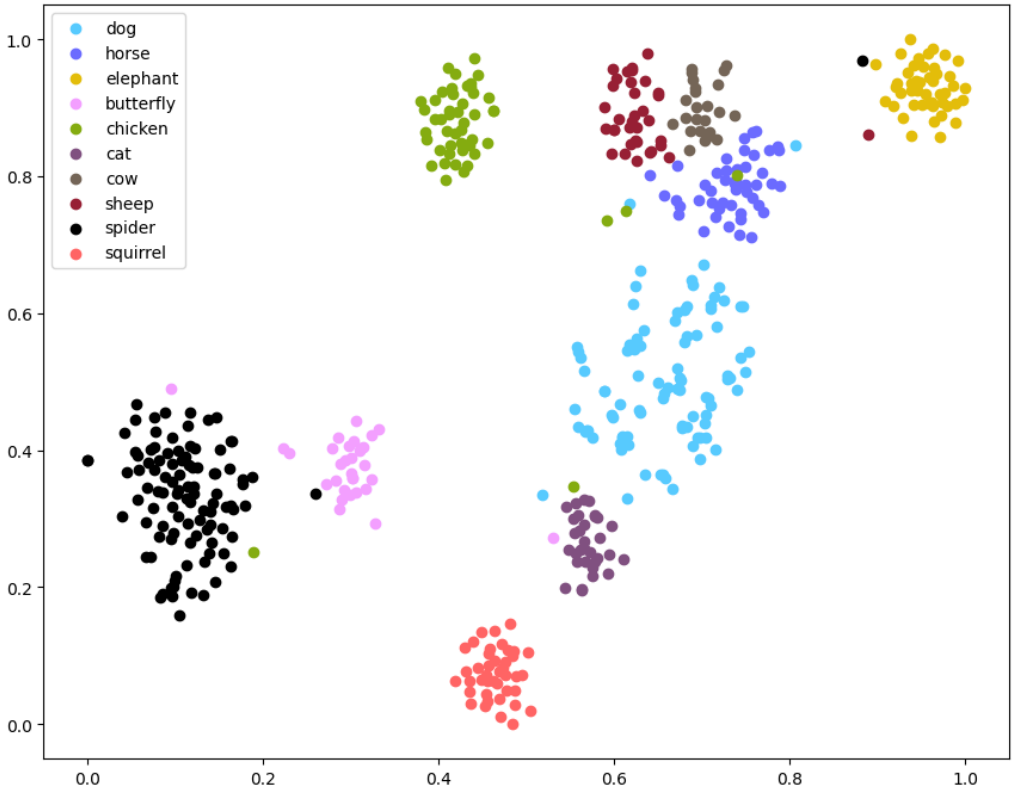
(a) *EGFR* t-SNE using hybrid semantic features.



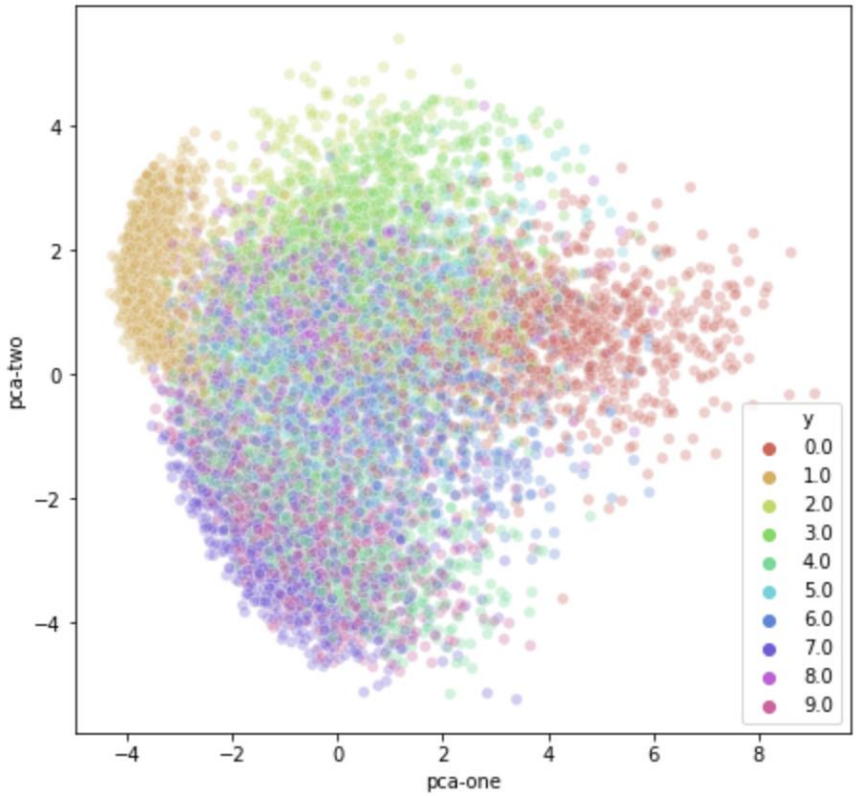
(b) *KRAS* t-SNE using hybrid semantic features.

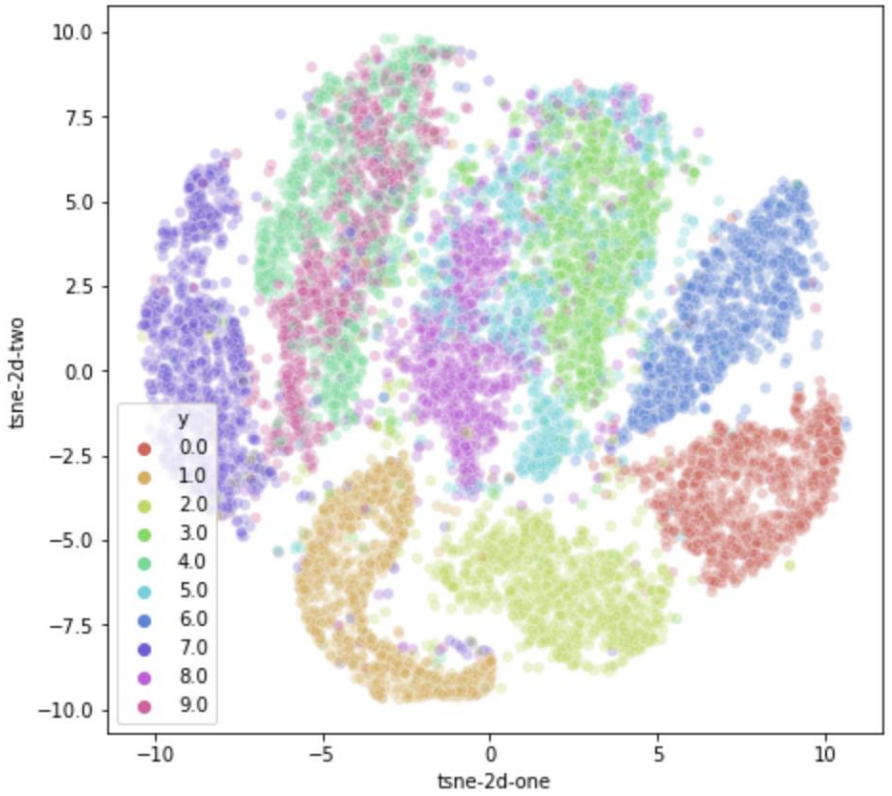
t-SNE Vizualization





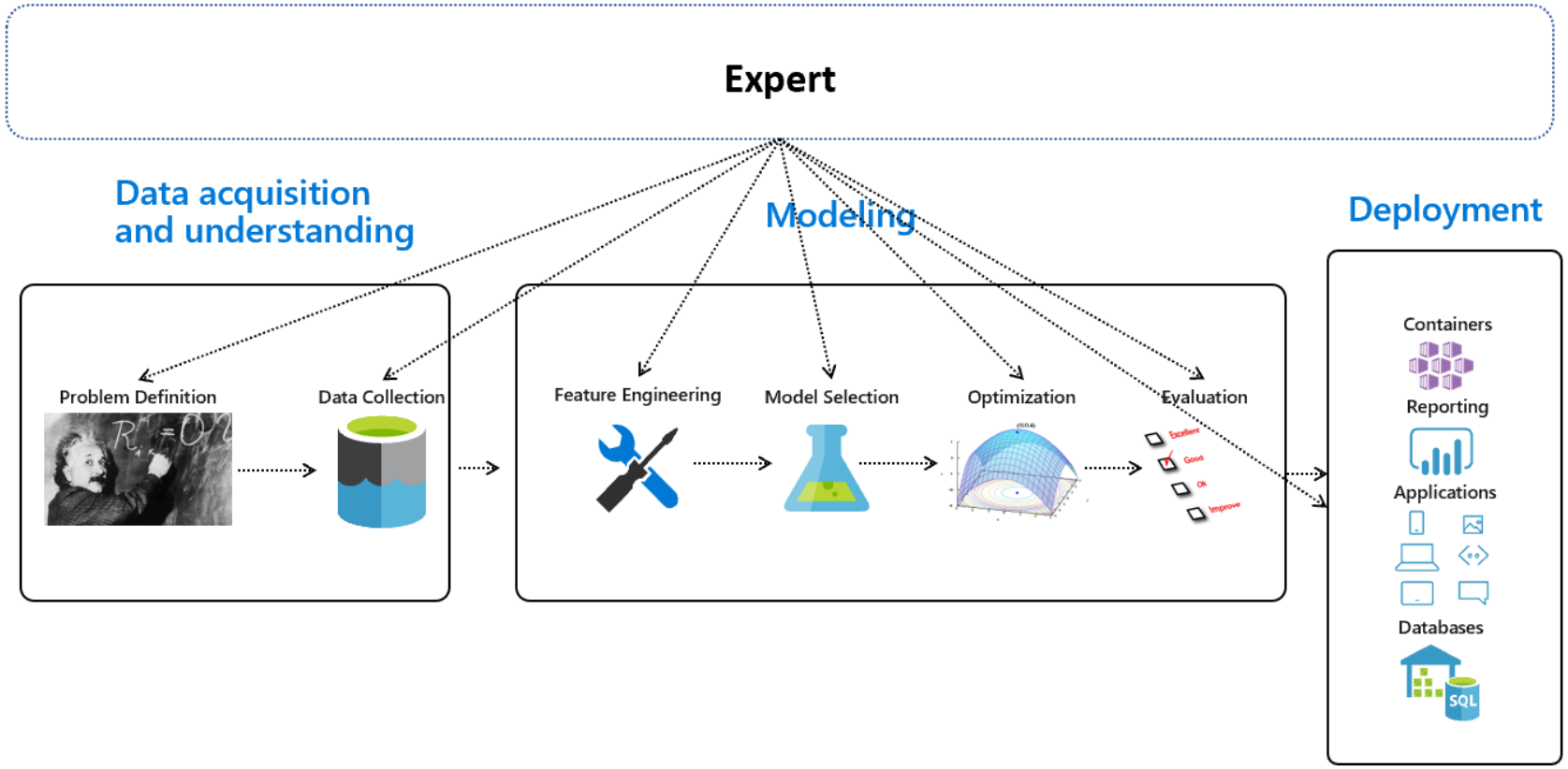
t-SNE vs PCA



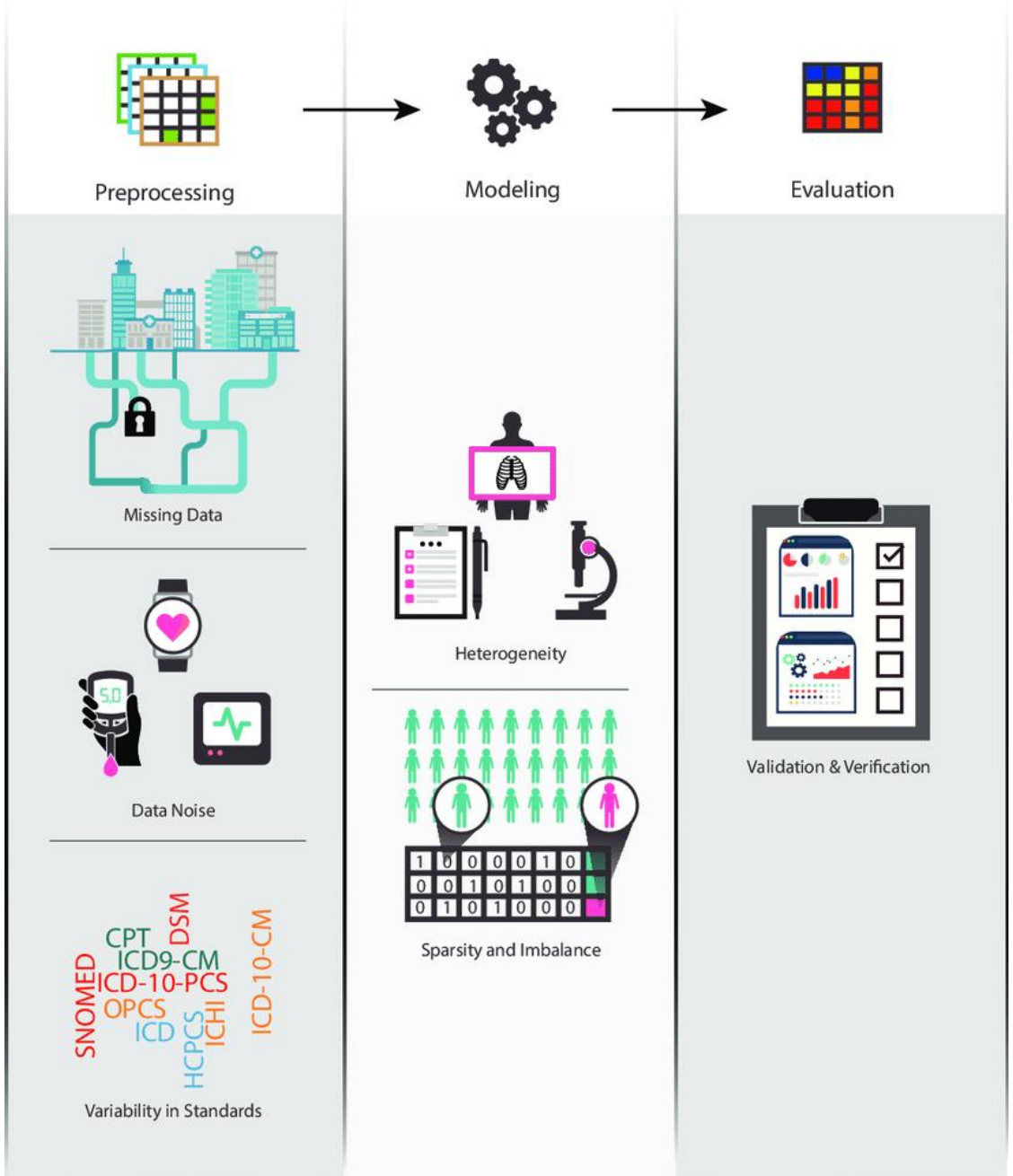


https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b

Supervised Learning




Data
Code
AI
Model


Preprocessing
Modeling
Evaluation
Missing Data
Data Noise
CPT
DSM
ICD9-CM
SNOMED
ICD-10-PCS
OPCS
ICD
ICHI
HCPCS
ICD-10-CM
Variability in Standards
Heterogeneity
Sparsity and Imbalance
Validation & Verification

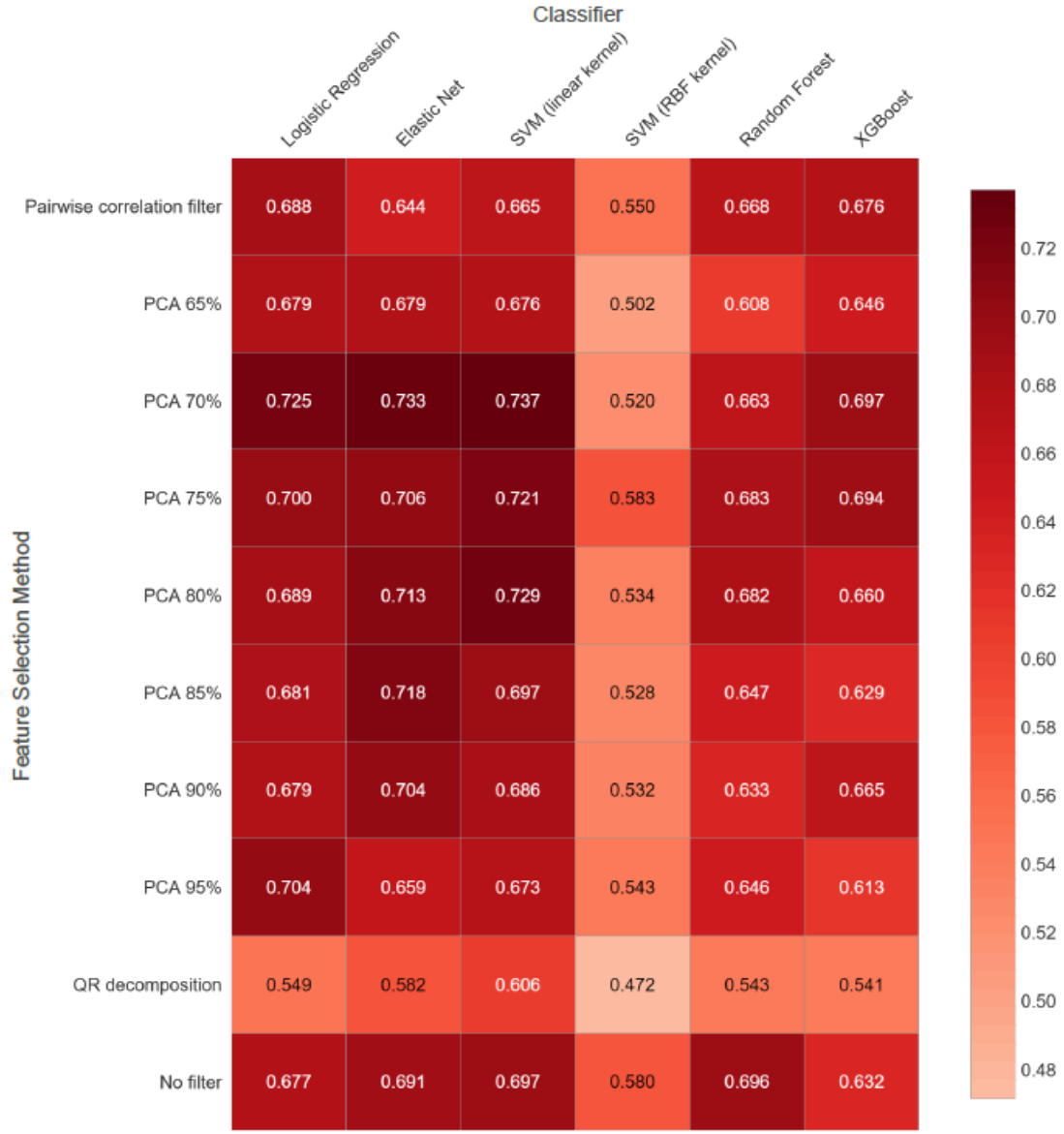


Figure 3. Heatmap with the AUC of each classifier/feature selection combination. Dark colors stand for the best results, while light colors represent the worst outcomes.

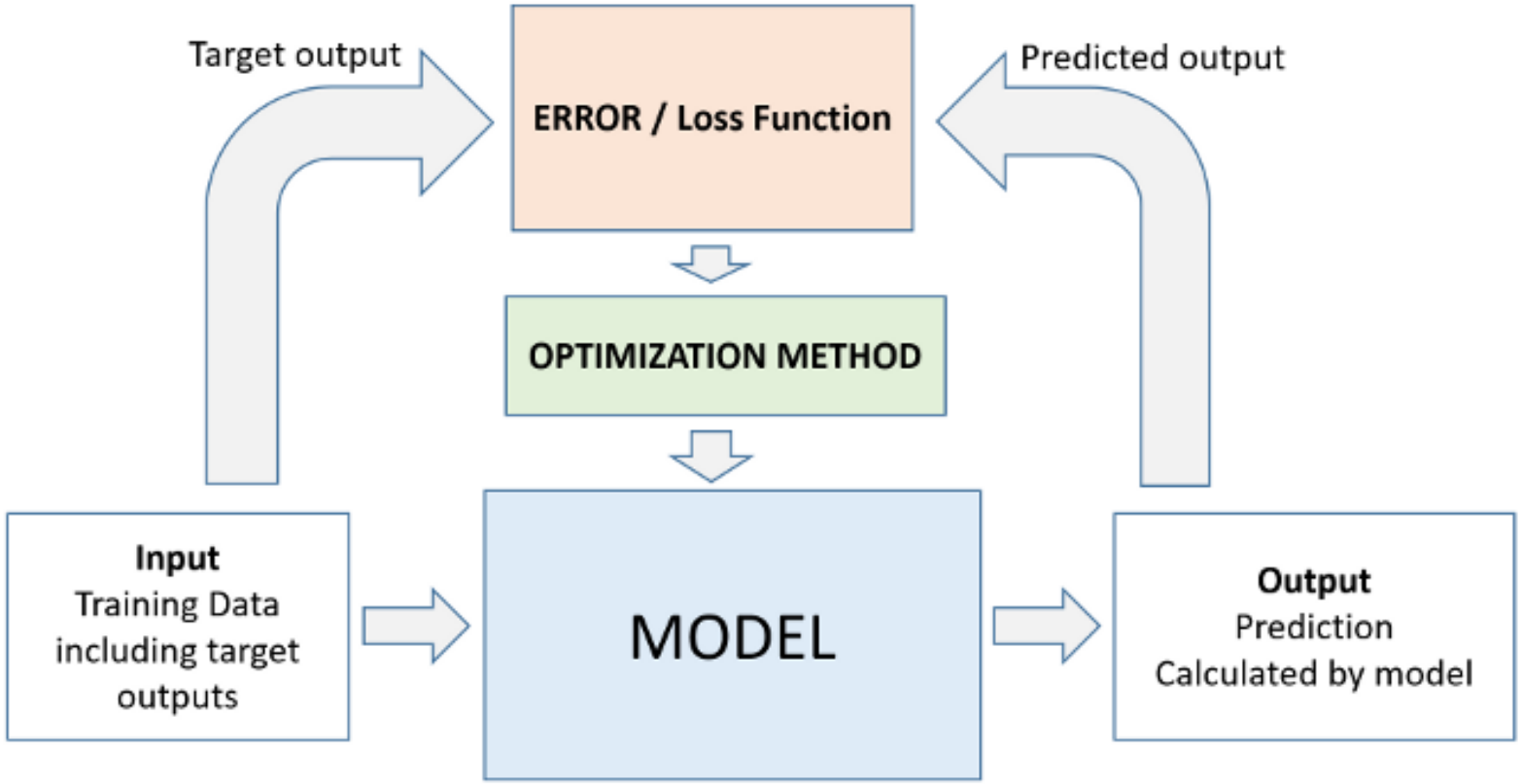
Classification Methods

How does machine learning work?

1.A Decision Process: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labeled or unlabeled, your algorithm will produce an estimate about a pattern in the data.

2.A loss Function: An loss function evaluates the prediction of the model. If there are known examples, an loss function can make a comparison to assess the accuracy of the model.

3.A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this iterative “evaluate and optimize” process, updating weights autonomously until a threshold of accuracy has been met.



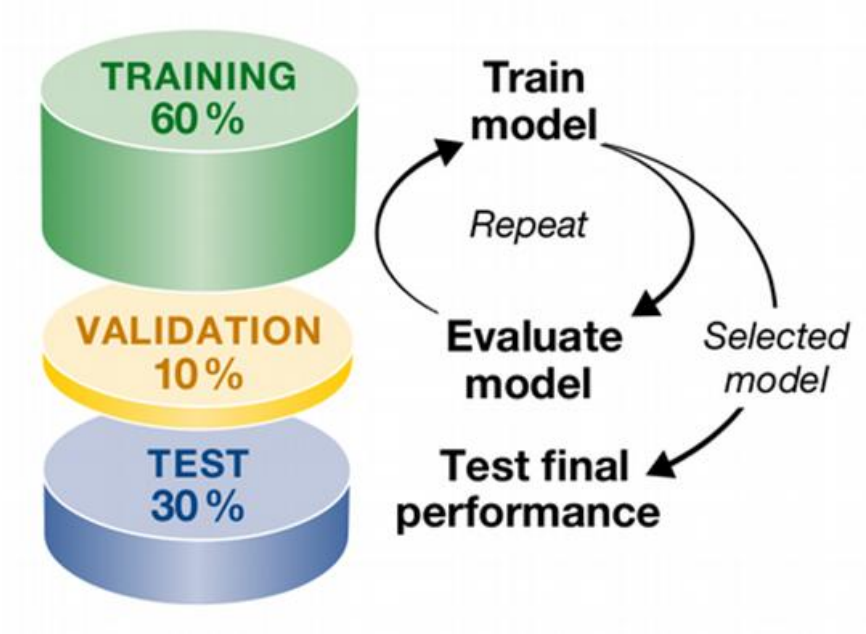
Optimizer

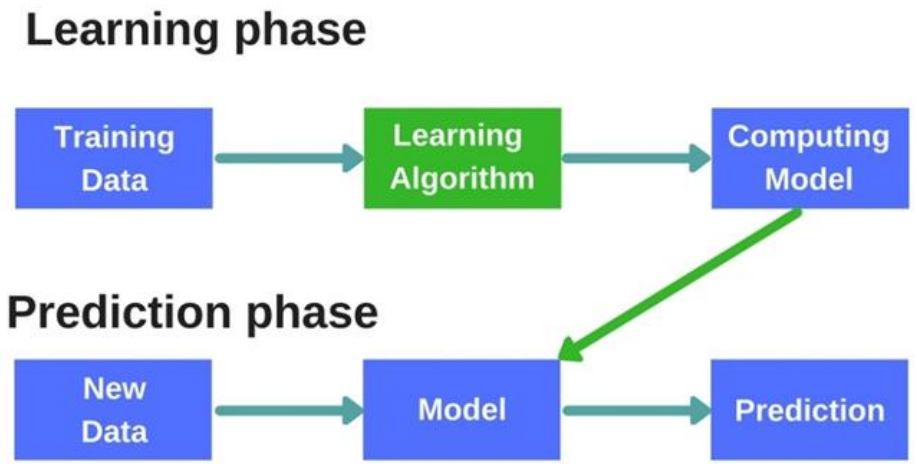
The optimizer function is **responsible for updating the model's parameters in order to minimize the loss function**. The main goal of the optimizer function is **to find the optimal set of parameters that minimize the loss function**, thus improving the model's ability to make accurate predictions.

Loss Functions

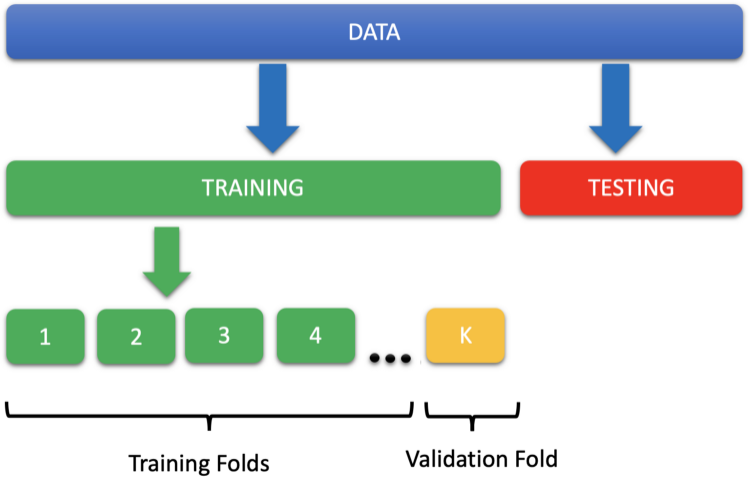
A loss function, is used to measure the accuracy of a model's predictions. It calculates the difference between the predicted output and the actual output for each training sample.
The goal of the model is to minimize the loss function. By minimizing the loss function, we are effectively trying to find the best set of parameters that will produce the most accurate predictions.

Train

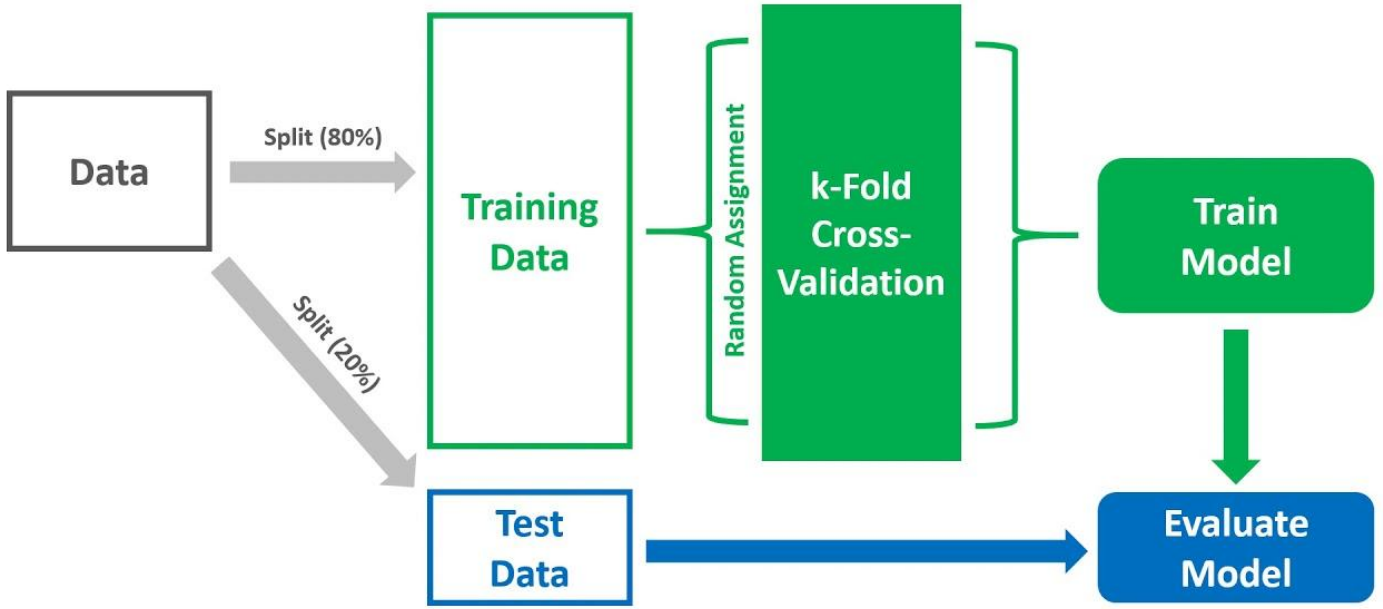

TRAINING
60 %
VALIDATION
10 %
TEST
30 %
Train
model
Repeat
Evaluate
model
Selected
model
Test final
performance


Learning phase
Training
Data
Learning
Algorithm
Computing
Model
Prediction phase
New
Data
Model
Prediction

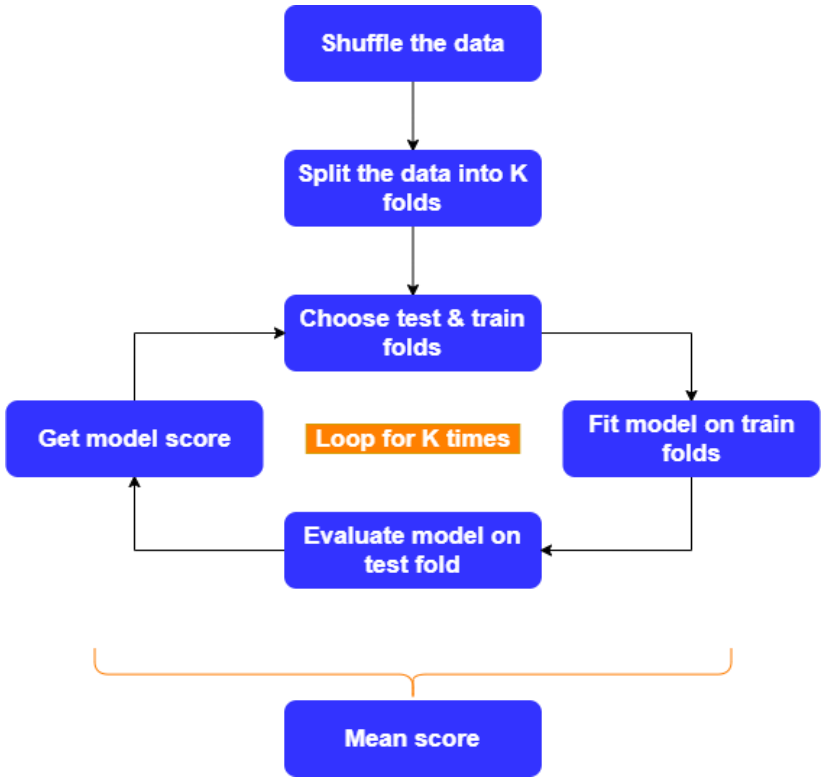
Train

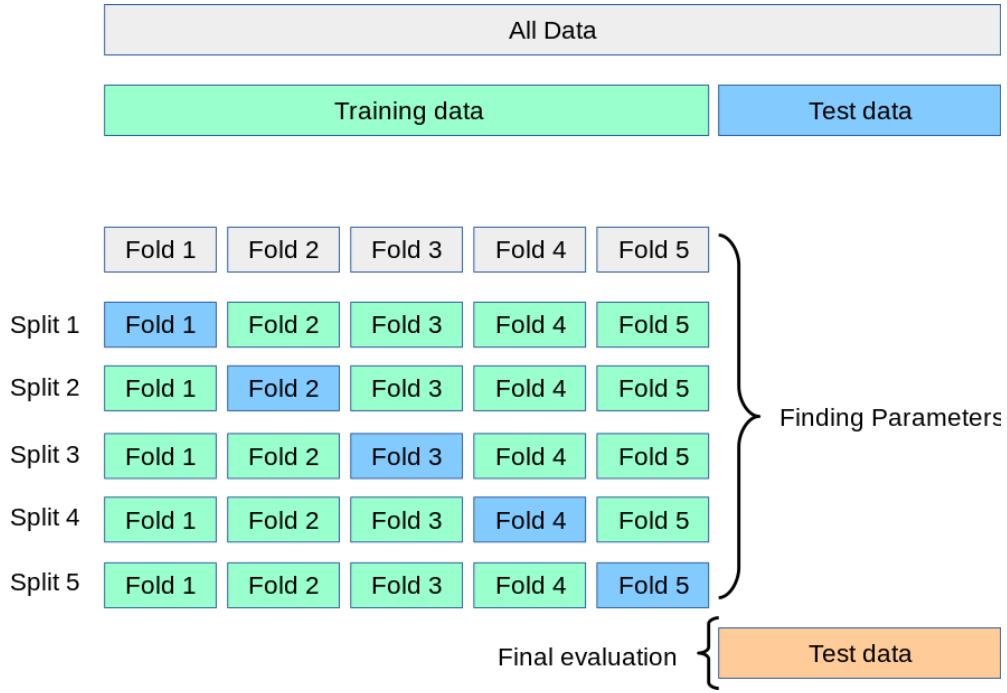


Example: k-Fold Cross-Validation



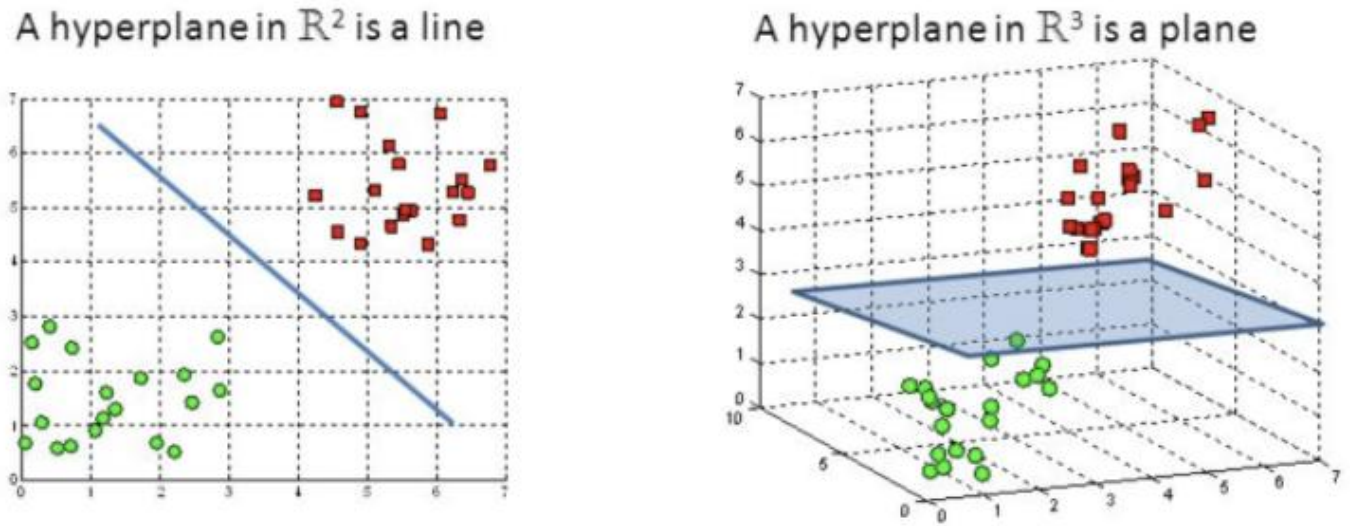
Train


Shuffle the data
Split the data into K folds
Choose test & train folds
Get model score
Loop for K times
Fit model on train folds
Evaluate model on test fold
Mean score


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Parameters
Final evaluation
Test data

SVM

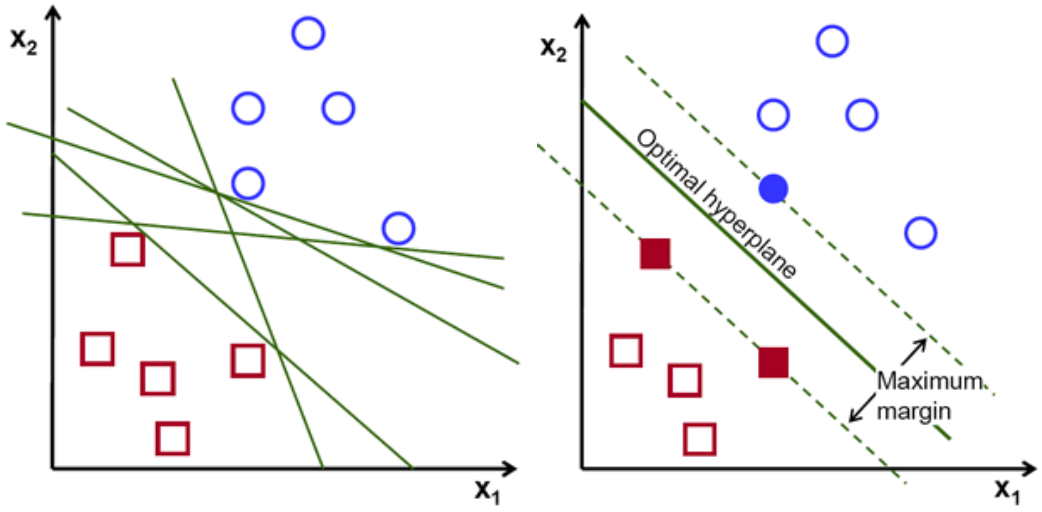
The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. SVM maps training examples to points in space so as to maximise the width of the gap between the two categories.

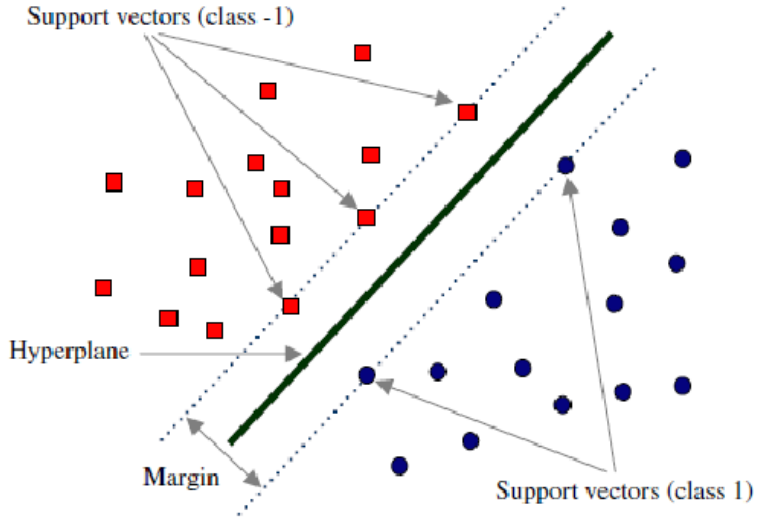


Hyperplanes in 2D and 3D feature space

SVM

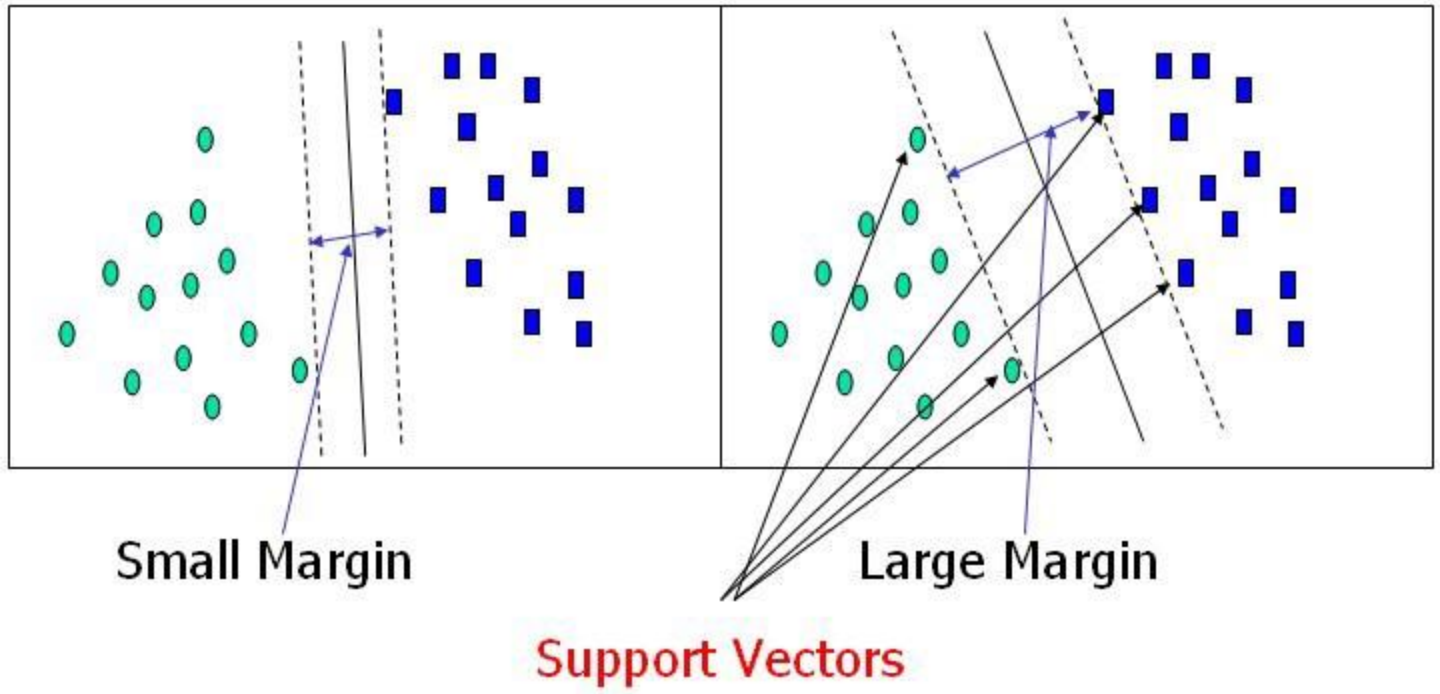
The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.





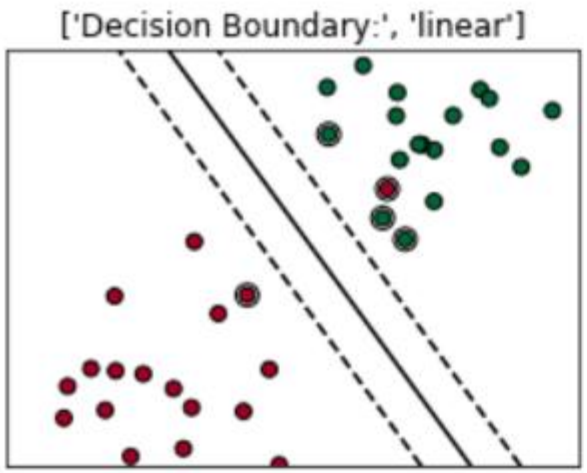
<u>Maximizing</u> the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin.

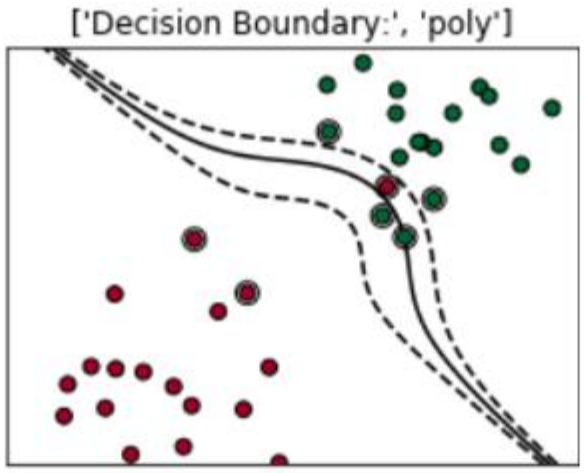
SVM

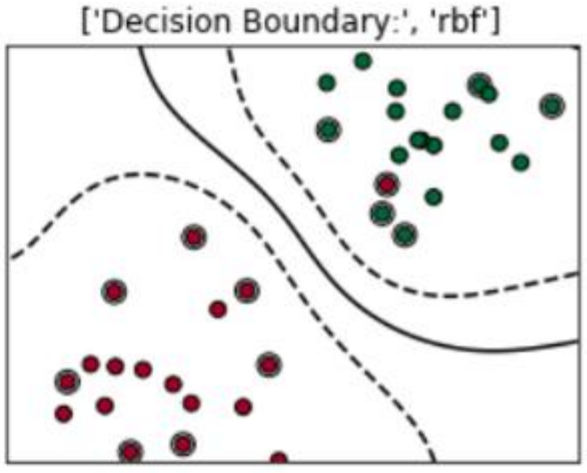


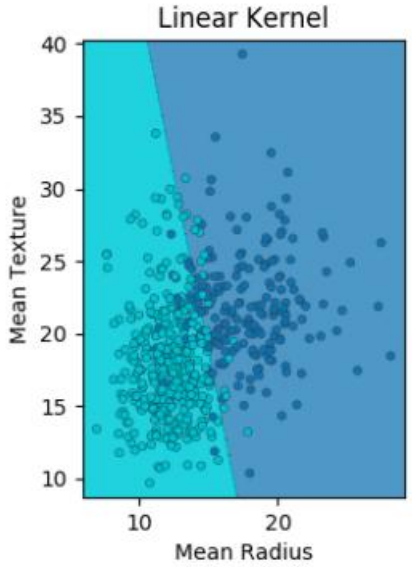
The objective is to find a plane that has the **maximum margin**, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

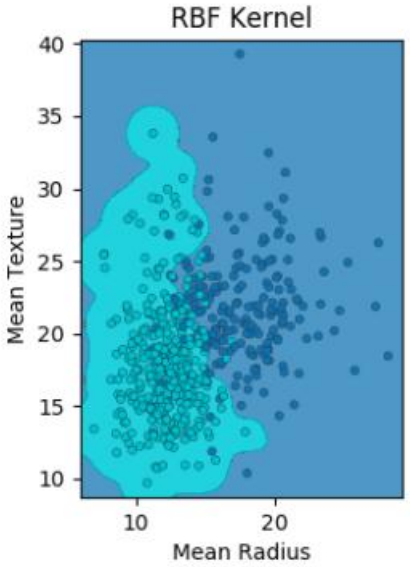
SVM

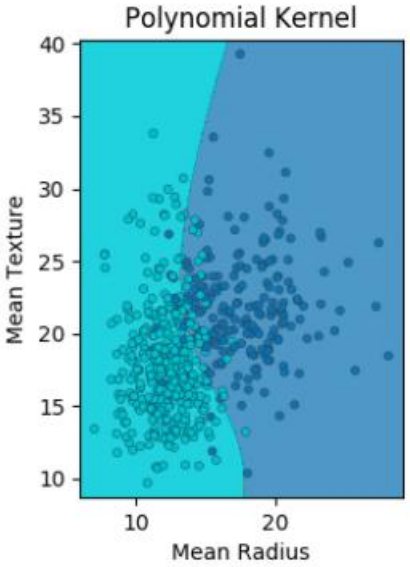

['Decision Boundary:', 'linear']


['Decision Boundary:', 'poly']

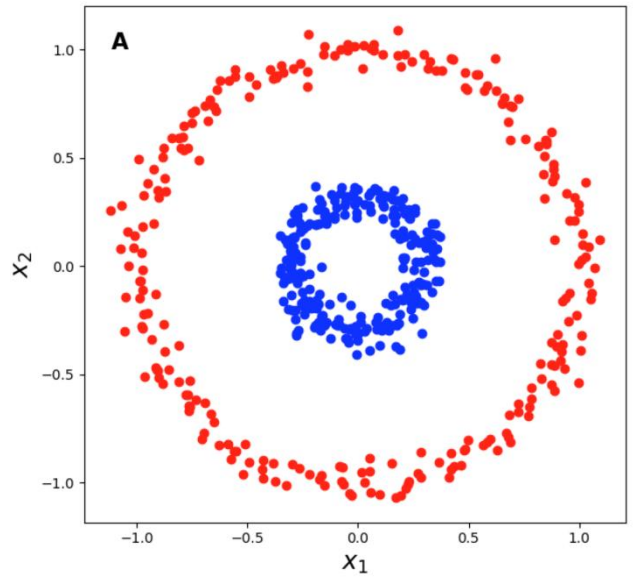

['Decision Boundary:', 'rbf']

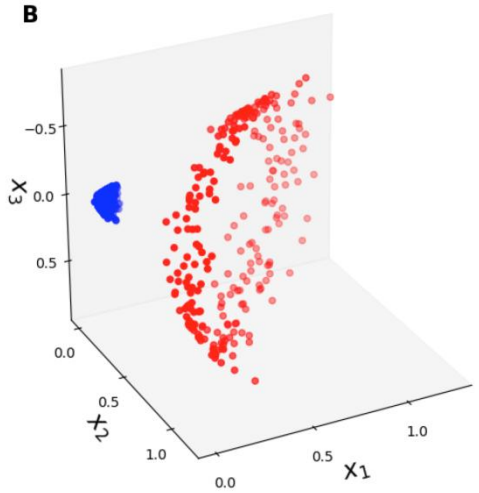

Linear Kernel
Mean Texture
Mean Radius
40
35
30
25
20
15
10
10
20


RBF Kernel
Mean Texture
Mean Radius
40
35
30
25
20
15
10
10
20


Polynomial Kernel
Mean Texture
Mean Radius
40
35
30
25
20
15
10
10
20

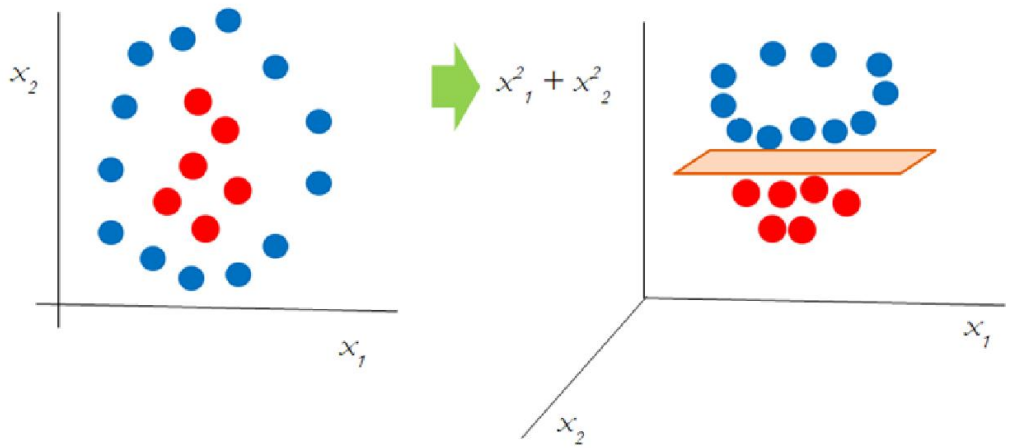
SVM



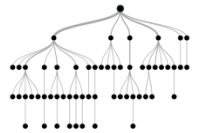


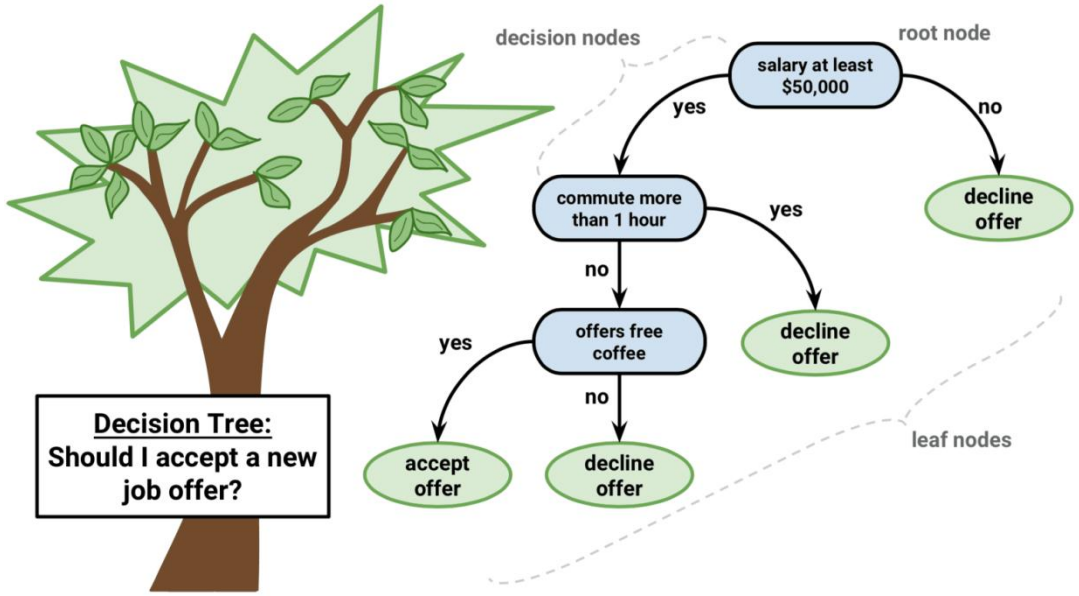
2-Dimensional Linearly Inseparable Classes

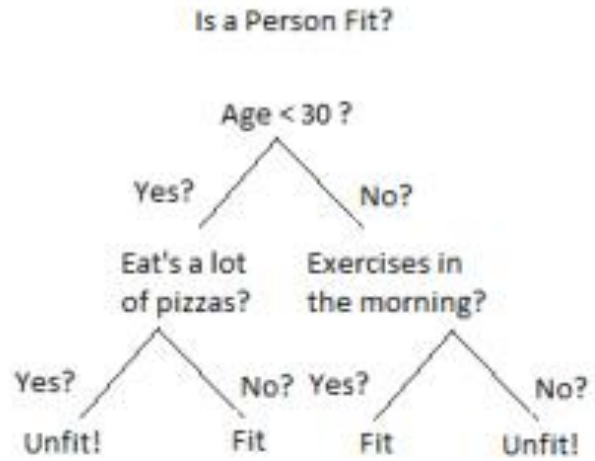
2-Dimensional Linearly Inseparable Classes with Polynomial kernel with Degree 2



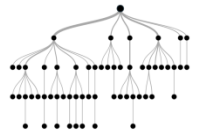
Decision Trees

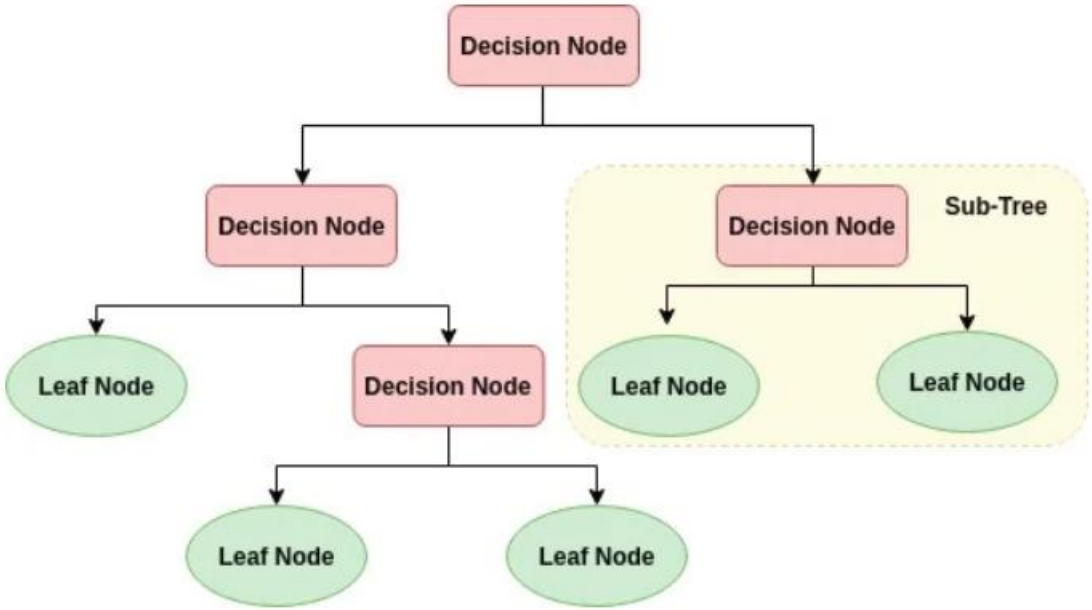






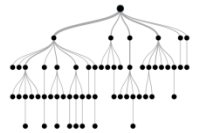
Decision Trees





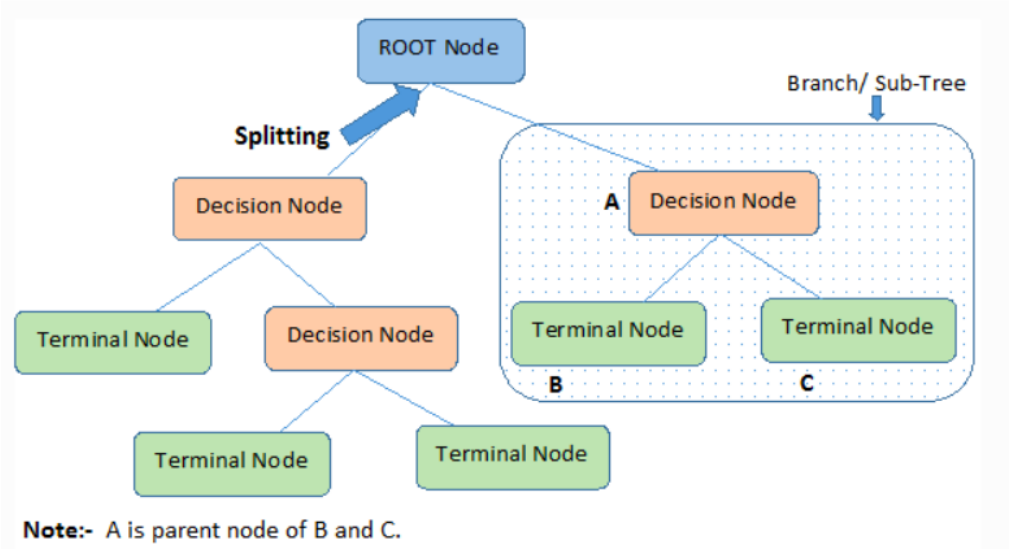
A decision tree is a **decision support tool that uses a tree-like model** of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision Trees

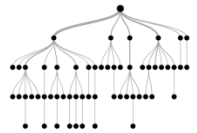


Important Terminology related to Decision Trees

Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
Splitting: It is a process of dividing a node into two or more sub-nodes.
Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.
Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.
Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.



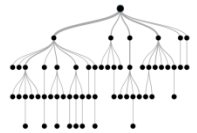
Decision Trees



Some advantages of decision trees are:

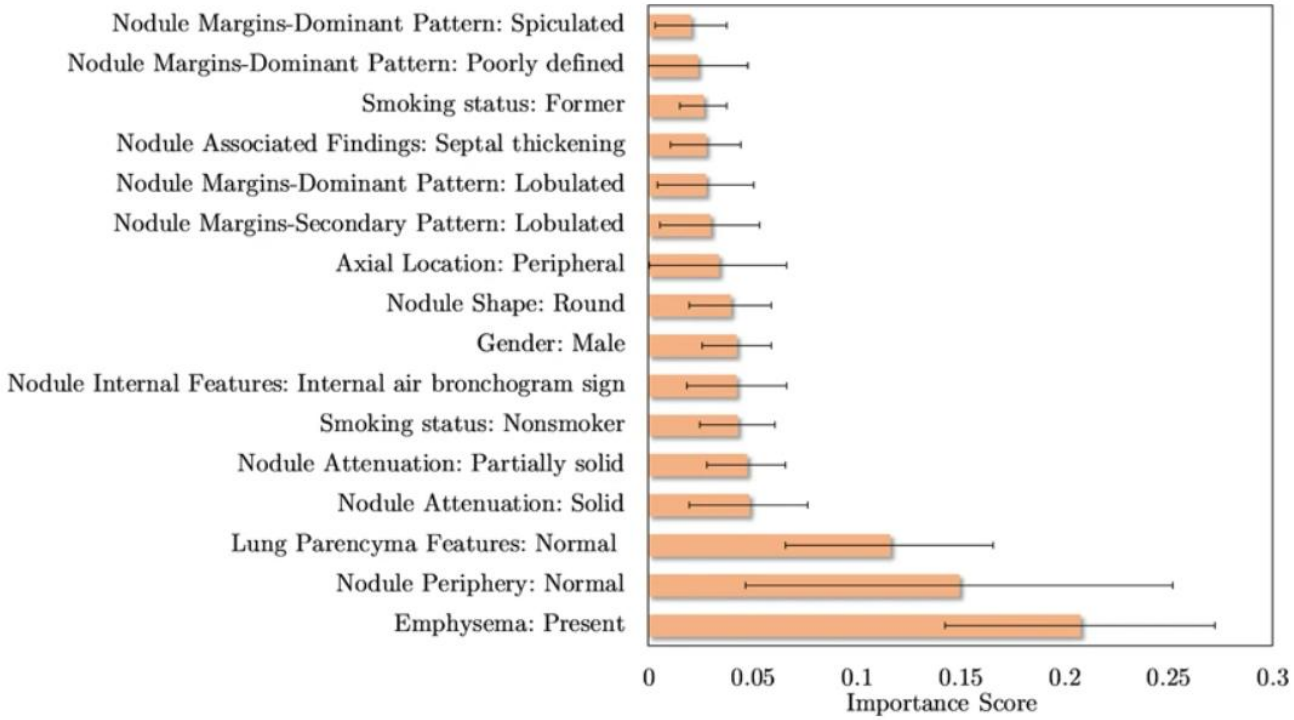
- Simple to **understand and to interpret**. Trees can be visualised.
- Requires **little data preparation**. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
- Able to handle both **numerical and categorical data**. Other techniques are usually specialized in analyzing datasets that have only one type of variable.
- Able to **handle multi-output problems**.

Decision Trees

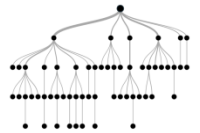


Feature importance

The overall importance of a feature in a decision tree can be computed. The sum of all importances is scaled to 100. This means that each importance can be interpreted as share of the overall model importance.



Decision Trees

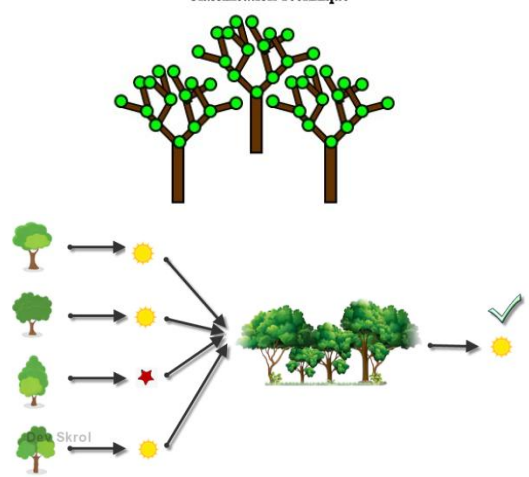


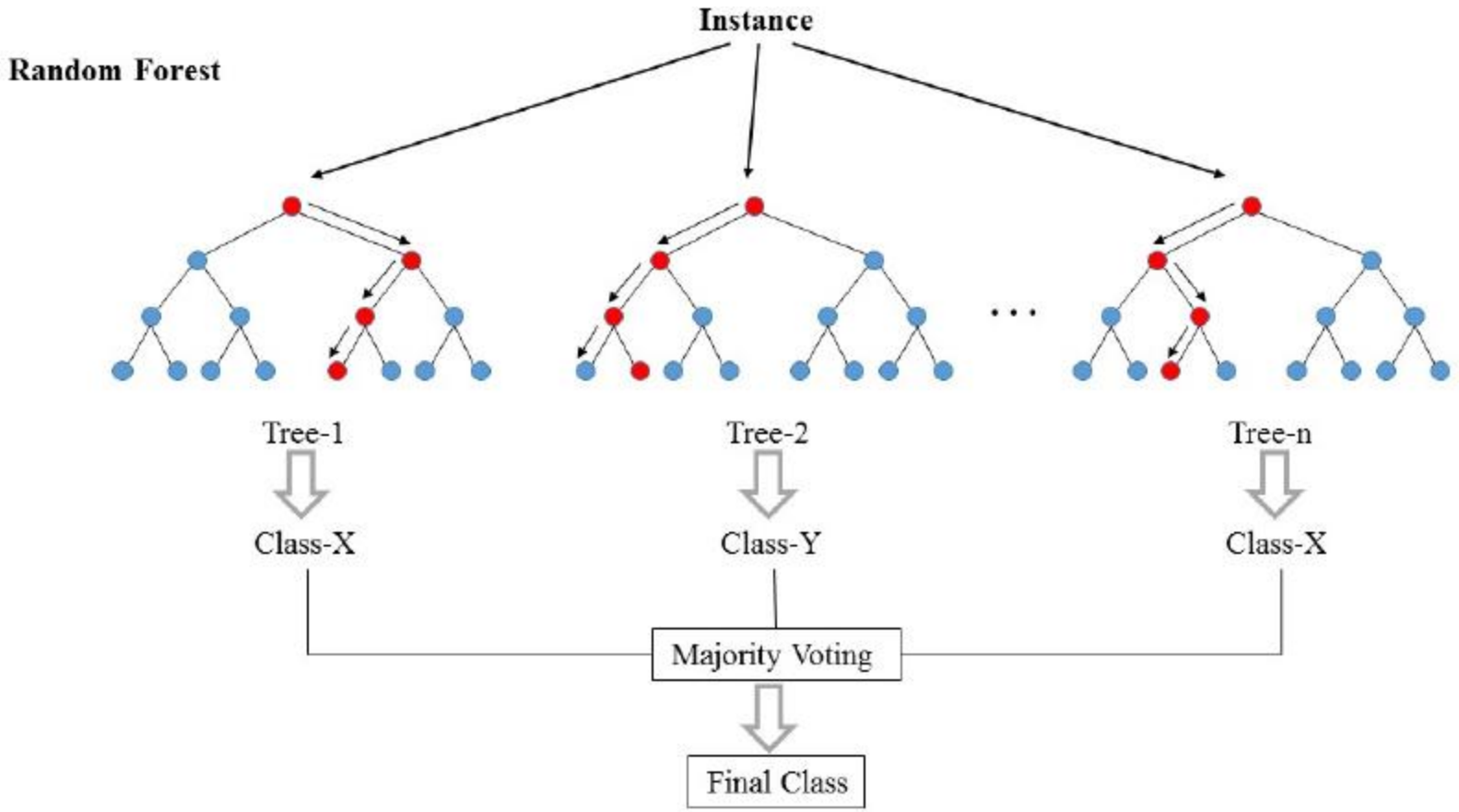
The disadvantages of decision trees include:

- Decision-tree learners can create over-complex trees that **do not generalize** the data well. This is called overfitting.

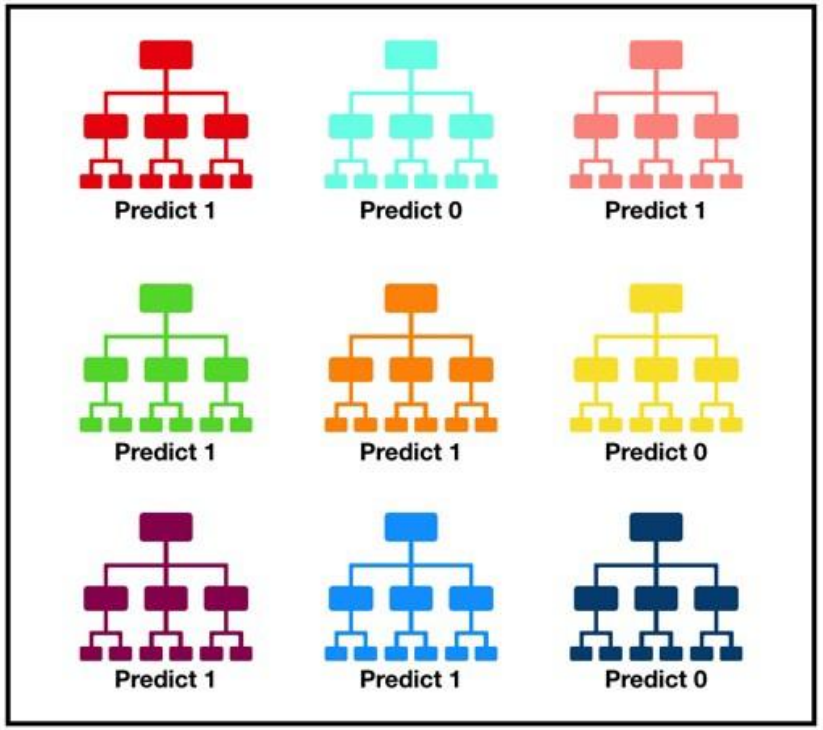
- Decision trees **can be unstable because small variations** in the data might result in a completely different tree being generated.

Random Forest





Random Forest



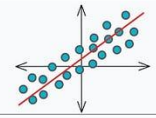
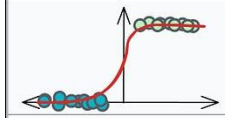
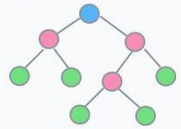
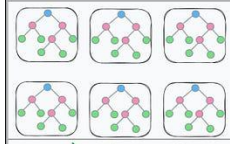
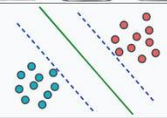
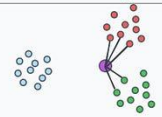
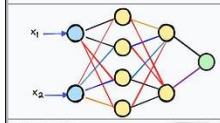
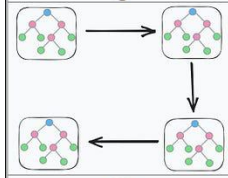
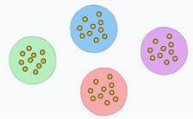
Random forest, like its name implies, consists of a large number of individual decision trees that operate as **an ensemble**. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

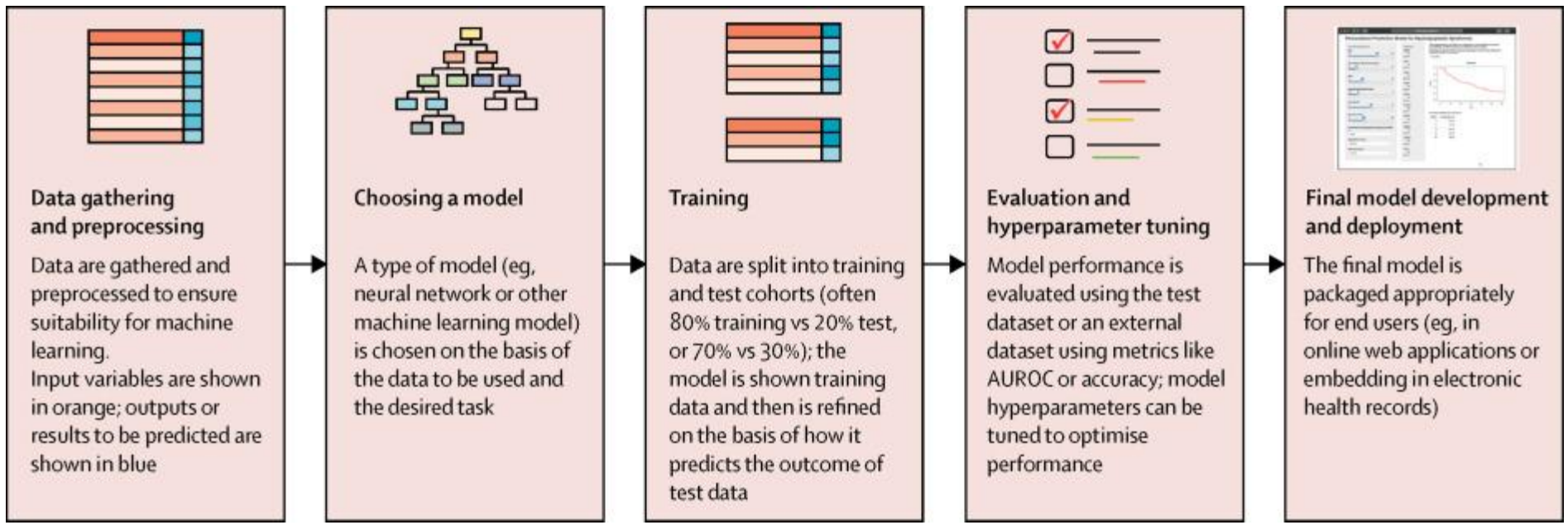
Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Loss functions of 16 Most Popular ML Algorithms

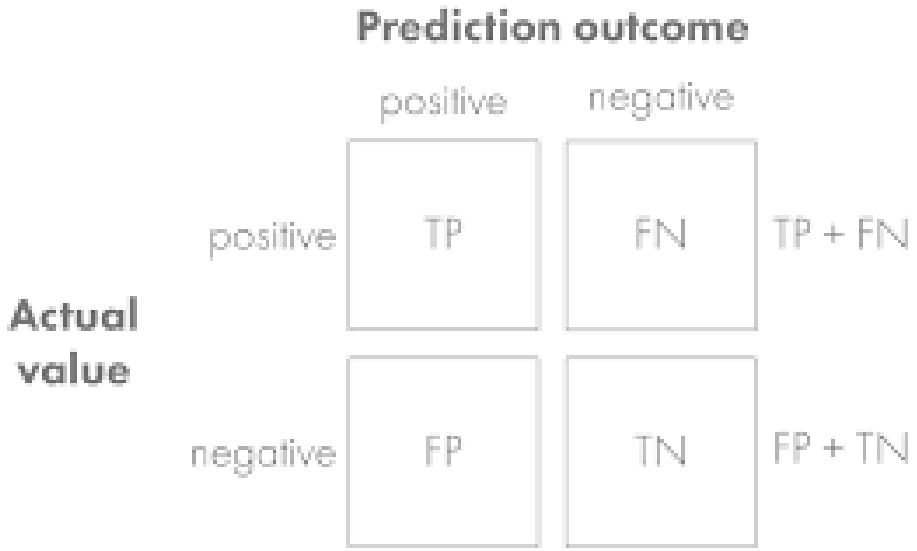
blog.DailyDoseofDS.com

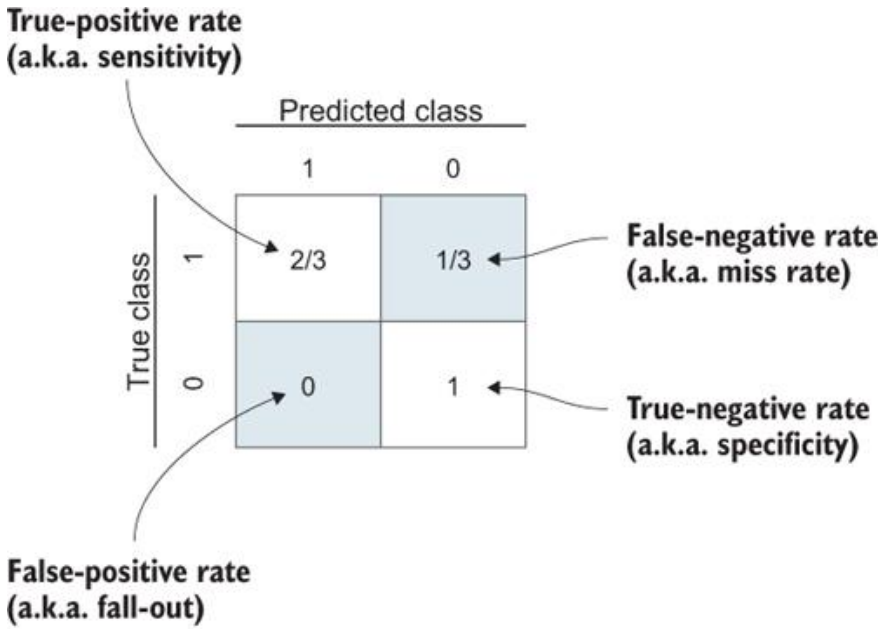
	Linear Regression	Mean Squared Error
	Logistic Regression	Cross-Entropy Loss
	Decision Tree Classifier	Information Gain or Gini impurity
	Decision Tree Regressor	Mean Squared Error
	Random Forest Classifier	Information Gain or Gini impurity
	Random Forest Regressor	Mean Squared Error
	Support Vector Machines (SVMs)	Hinge Loss
	k-Nearest Neighbors	No loss function
$P(B\|A) = \frac{P(\mathrm{B} \cap A)}{P(A)}$	Naive Bayes	No loss function
	Neural Networks	Regression: Mean Squared Error Classification: Cross-Entropy Loss
	AdaBoost	Exponential loss
	Gradient Boosting | LightGBM | CatBoost | XGBoost	Regression: Mean Squared Error Classification: Cross-Entropy Loss
	KMeans Clustering	??

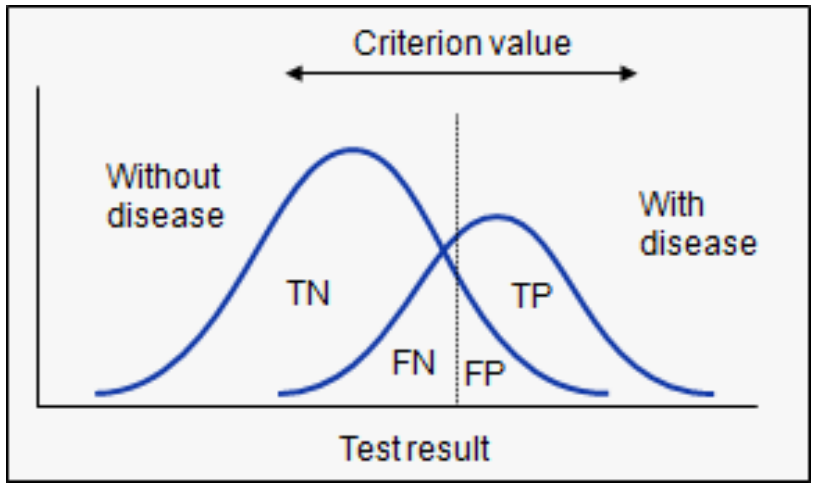
*These are typically used loss functions. It does not mean they are used always in these algorithms.

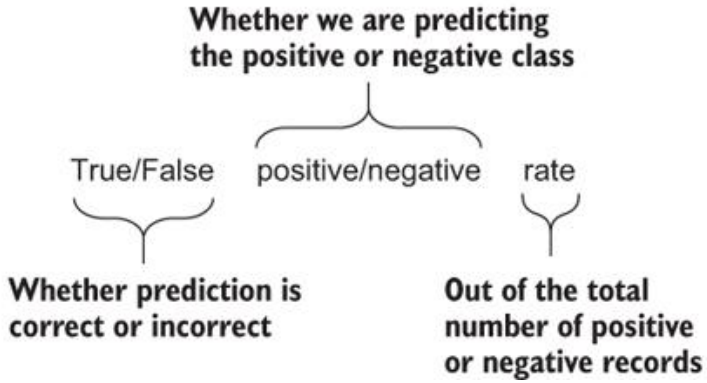

Data gathering and preprocessing
Data are gathered and preprocessed to ensure suitability for machine learning.
Input variables are shown in orange; outputs or results to be predicted are shown in blue
Choosing a model
A type of model (eg, neural network or other machine learning model) is chosen on the basis of the data to be used and the desired task
Training
Data are split into training and test cohorts (often 80% training vs 20% test, or 70% vs 30%); the model is shown training data and then is refined on the basis of how it predicts the outcome of test data
Evaluation and hyperparameter tuning
Model performance is evaluated using the test dataset or an external dataset using metrics like AUROC or accuracy; model hyperparameters can be tuned to optimise performance
Final model development and deployment
The final model is packaged appropriately for end users (eg, in online web applications or embedding in electronic health records)

Metrics

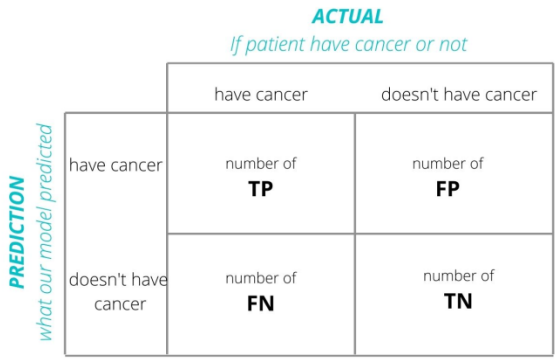

Prediction outcome
positive
negative
Actual value
positive
TP
FN
TP + FN
negative
FP
TN
FP + TN

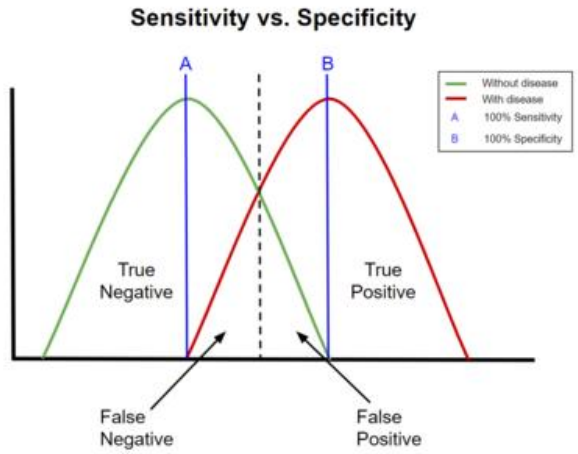

True-positive rate
(a.k.a. sensitivity)
Predicted class
1
0
True class
1
0
2/3
1/3
0
1
False-negative rate
(a.k.a. miss rate)
True-negative rate
(a.k.a. specificity)
False-positive rate
(a.k.a. fall-out)

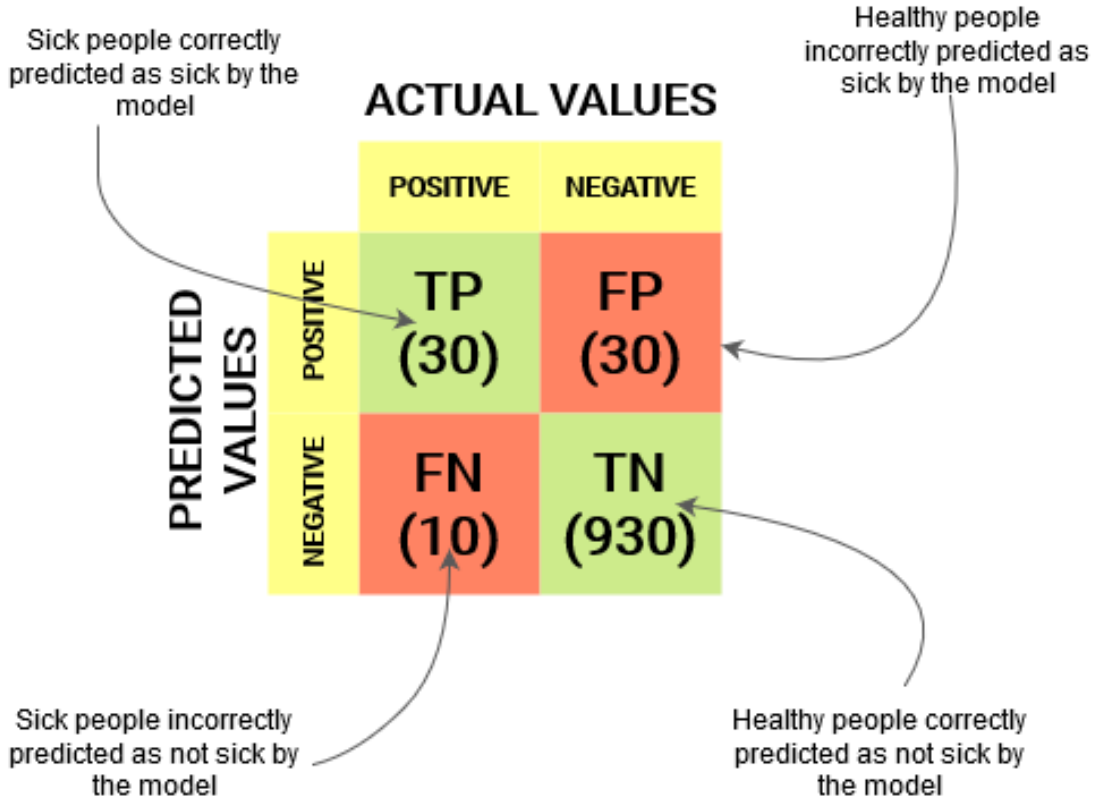

Criterion value
Without disease
With disease
TN
TP
FN
FP
Test result


Whether we are predicting the positive or negative class
True/False
positive/negative
rate
Whether prediction is correct or incorrect
Out of the total number of positive or negative records

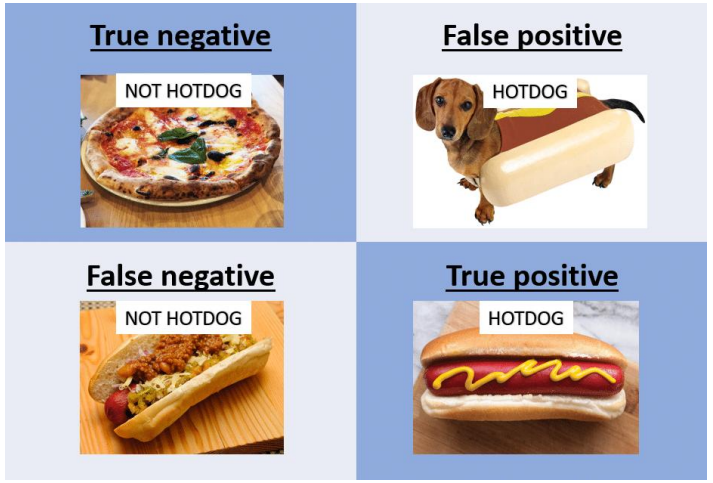
Metrics


ACTUAL
If patient have cancer or not
have cancer
doesn't have cancer
PREDICTION
what our model predicted
have cancer
doesn't have cancer
number of TP
number of FP
number of FN
number of TN


Sensitivity vs. Specificity
A
B
Without disease
With disease
A 100% Sensitivity
B 100% Specificity
True Negative
True Positive
False Negative
False Positive


Sick people correctly predicted as sick by the model
Healthy people incorrectly predicted as sick by the model
ACTUAL VALUES
POSITIVE
NEGATIVE
PREDICTED VALUES
POSITIVE
NEGATIVE
TP (30)
FP (30)
FN (10)
TN (930)
Sick people incorrectly predicted as not sick by the model
Healthy people correctly predicted as not sick by the model

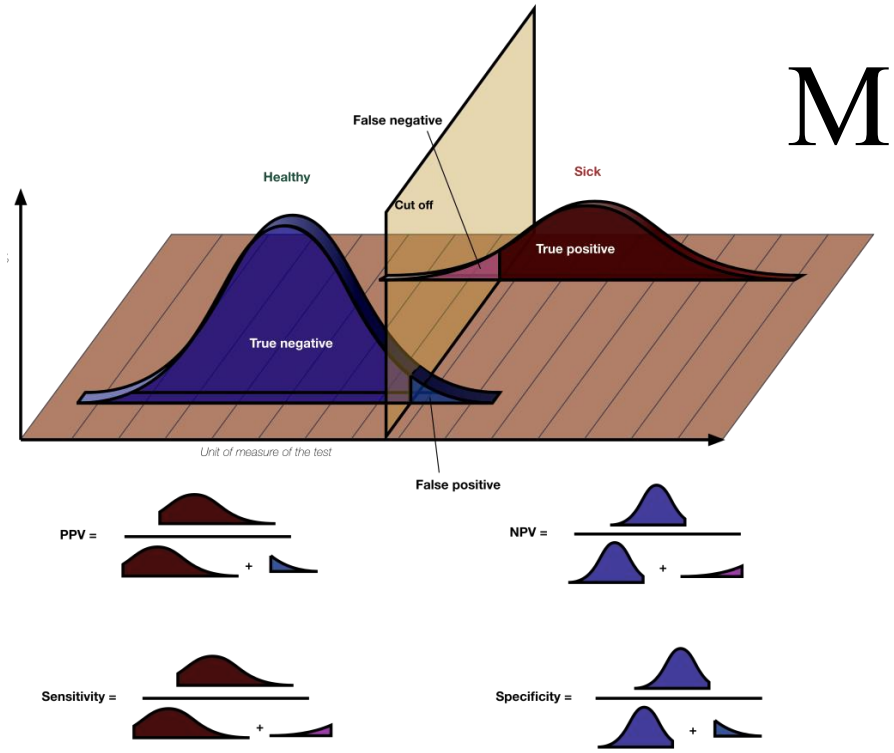
Metrics

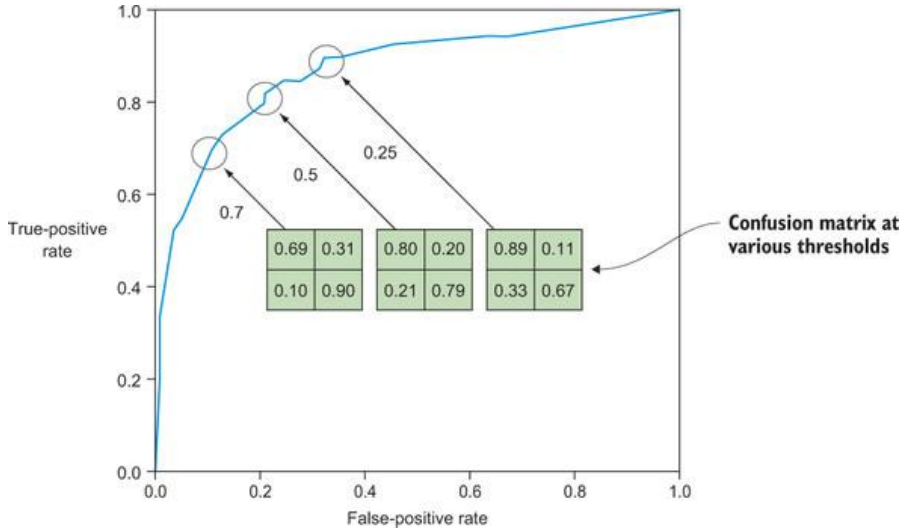


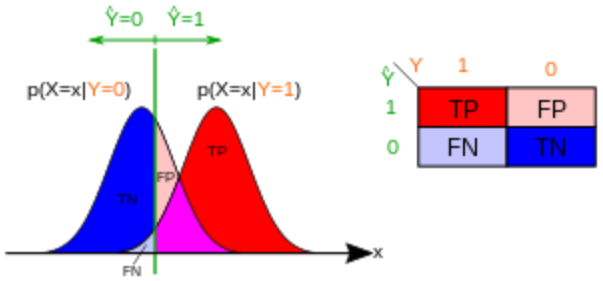
		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$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

Metric	Formula
True positive rate, recall	$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$
False positive rate	$\frac{\mathrm{FP}}{\mathrm{FP+TN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$
Accuracy	$\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$
F-measure	$\frac{2 \cdot \mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$

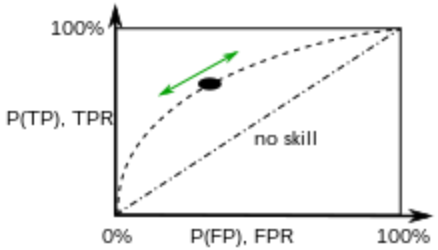
Metrics

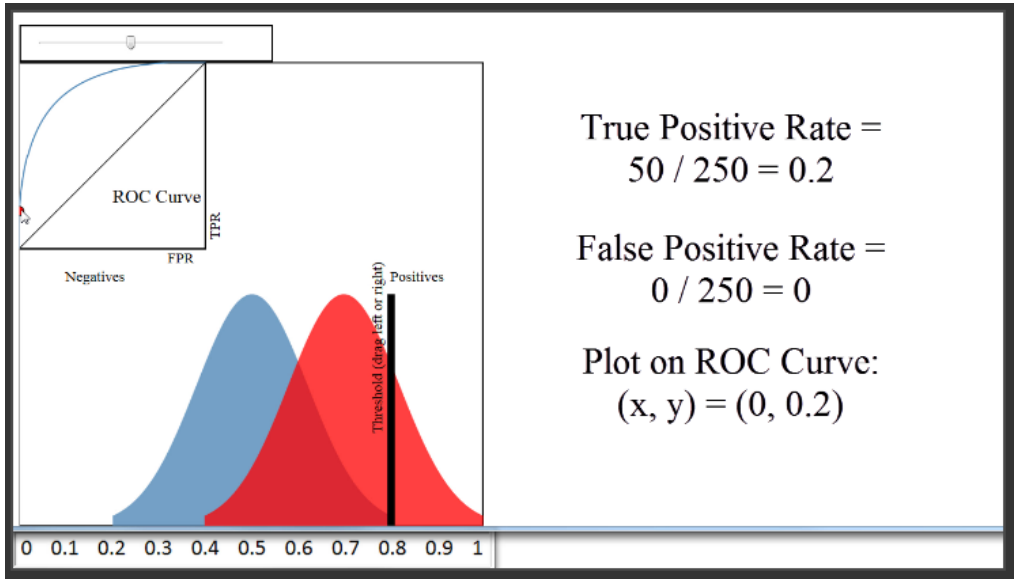

False negative
Healthy
Sick
Cut off
True positive
True negative
Unit of measure of the test
False positive
PPV =
NPV =
Sensitivity =
Specificity =


True-positive rate
False-positive rate
0.7
0.5
0.25
0.69 0.31
0.10 0.90
0.80 0.20
0.21 0.79
0.89 0.11
0.33 0.67
Confusion matrix at various thresholds

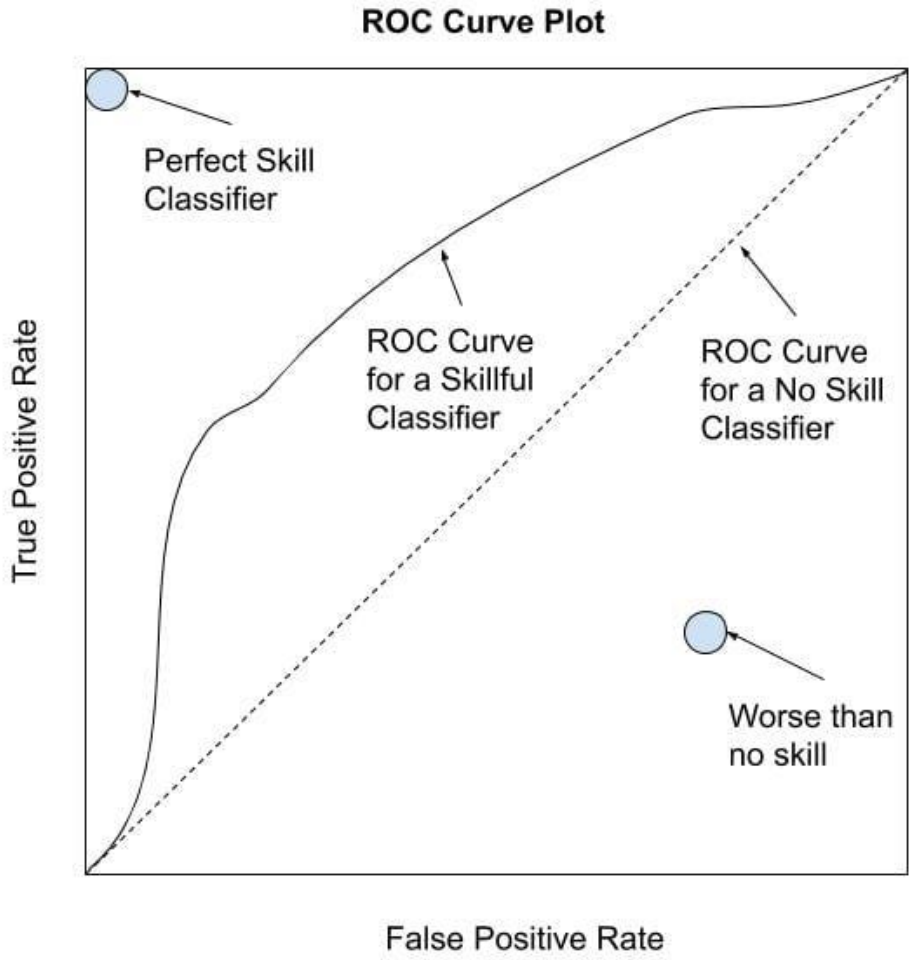

Ŷ=0
Ŷ=1
p(X=x|Y=0)
p(X=x|Y=1)
TN
FP
TP
FN
x

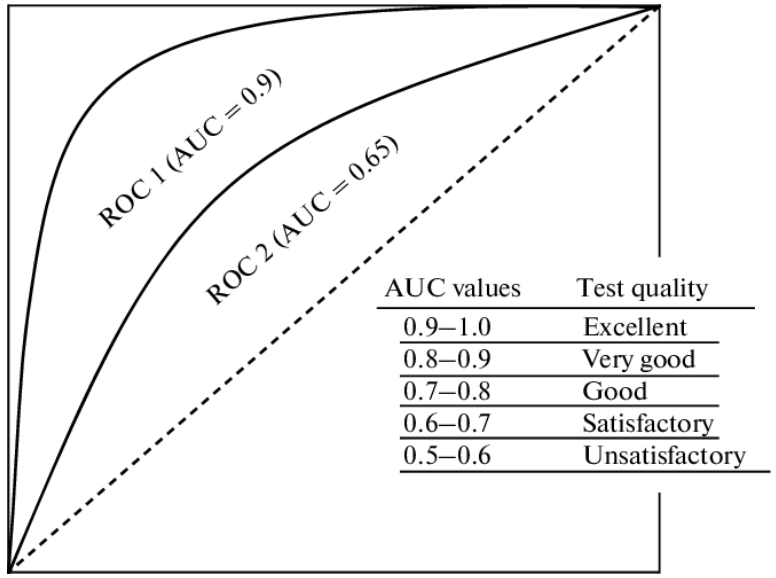
Y
1
0
Ŷ
1
0
TP
FP
FN
TN


100%
P(TP), TPR
no skill
0%
P(FP), FPR
100%

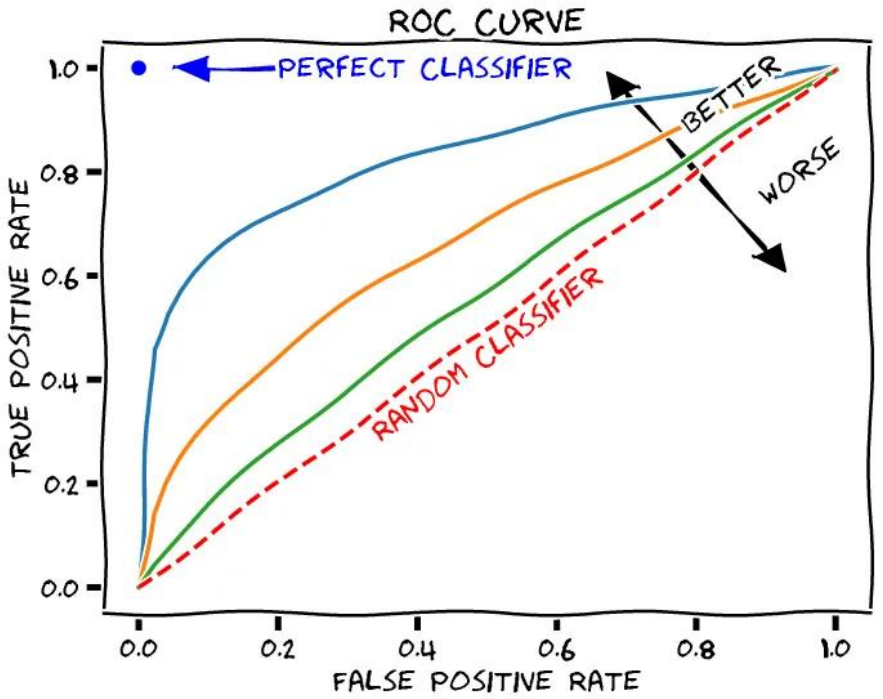

ROC Curve
TPR
FPR
Negatives
Positives
Threshold (drag left or right)
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
True Positive Rate = 50 / 250 = 0.2
False Positive Rate = 0 / 250 = 0
Plot on ROC Curve: (x, y) = (0, 0.2)

Metrics

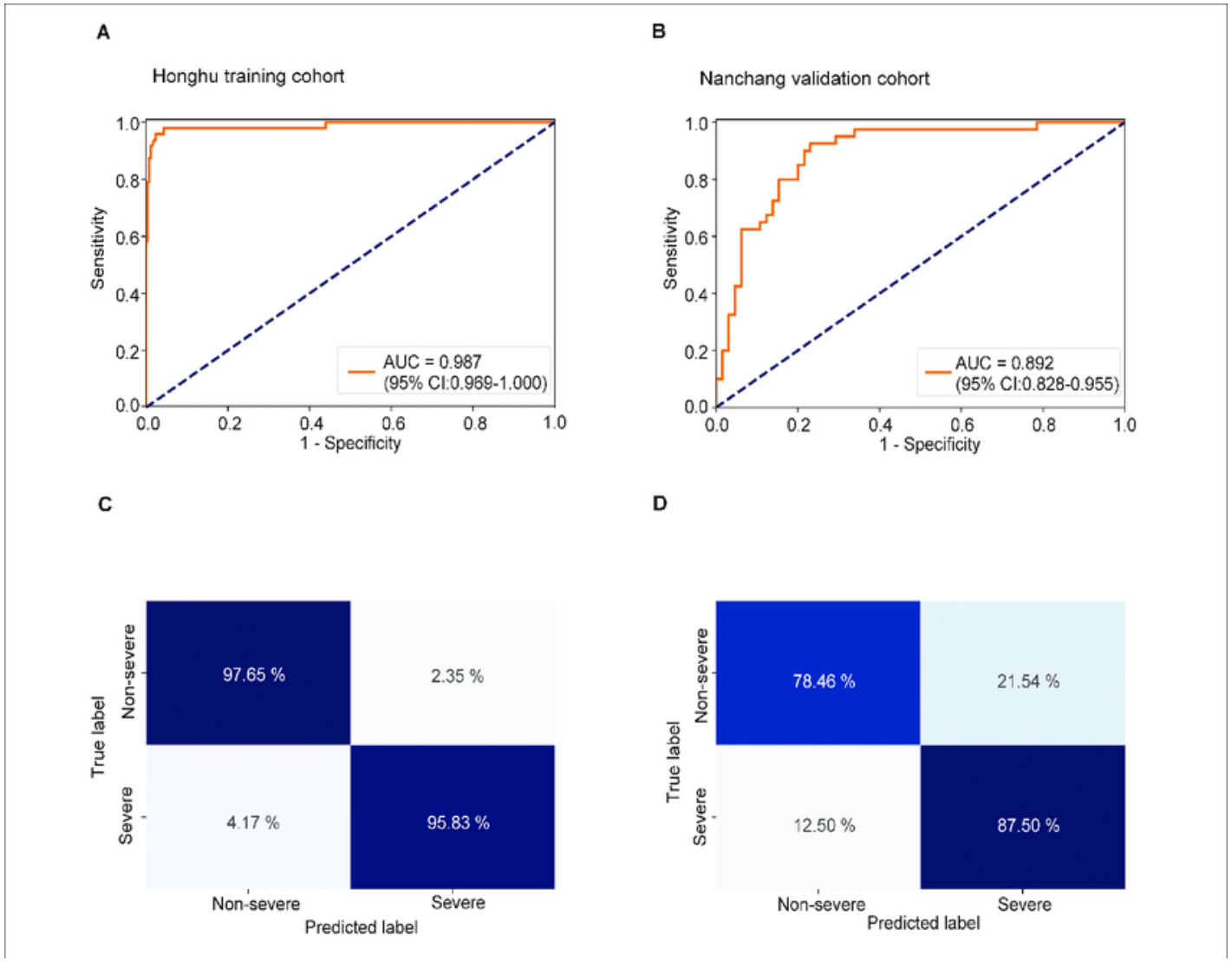




AUC values	Test quality
0.9–1.0	Excellent
0.8–0.9	Very good
0.7–0.8	Good
0.6–0.7	Satisfactory
0.5–0.6	Unsatisfactory

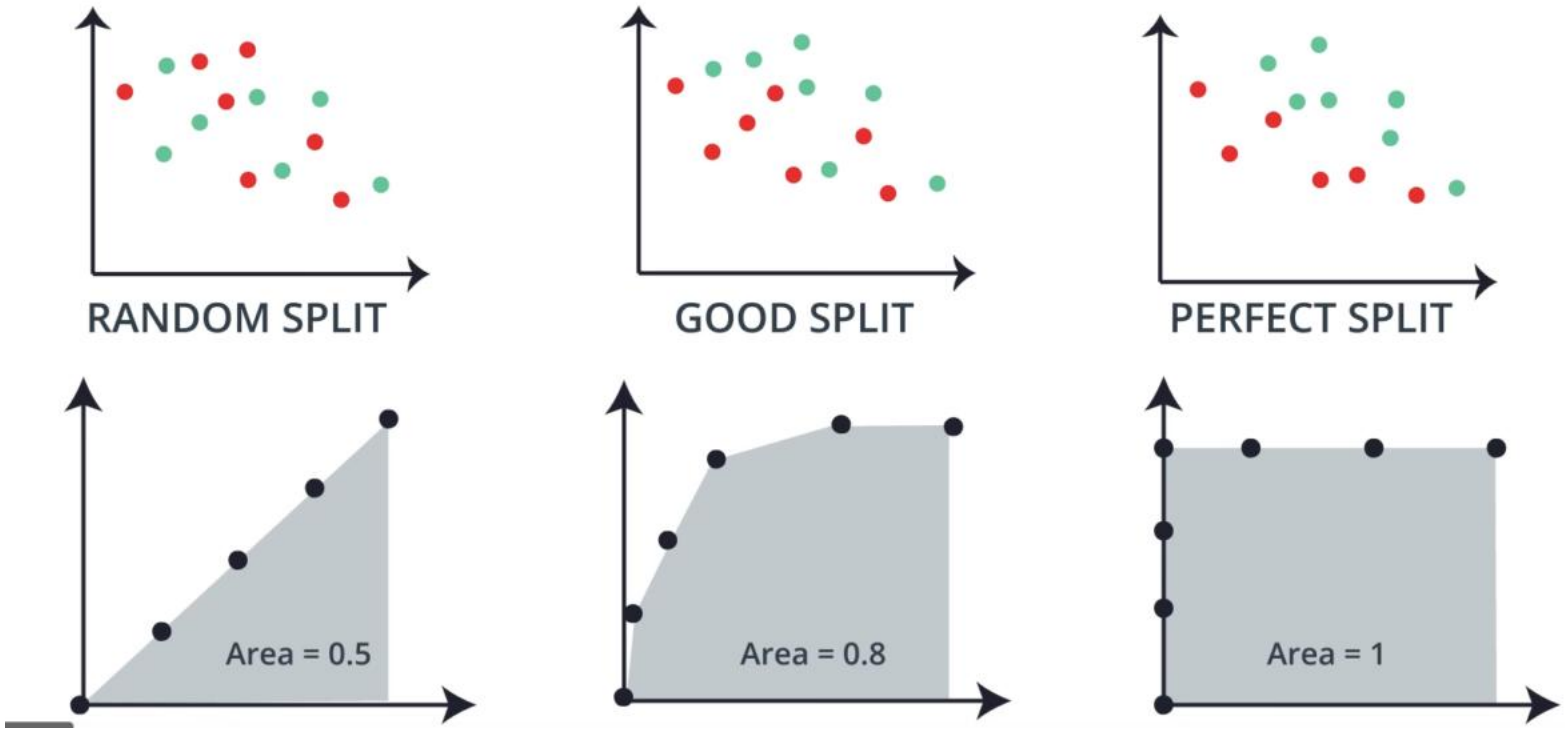


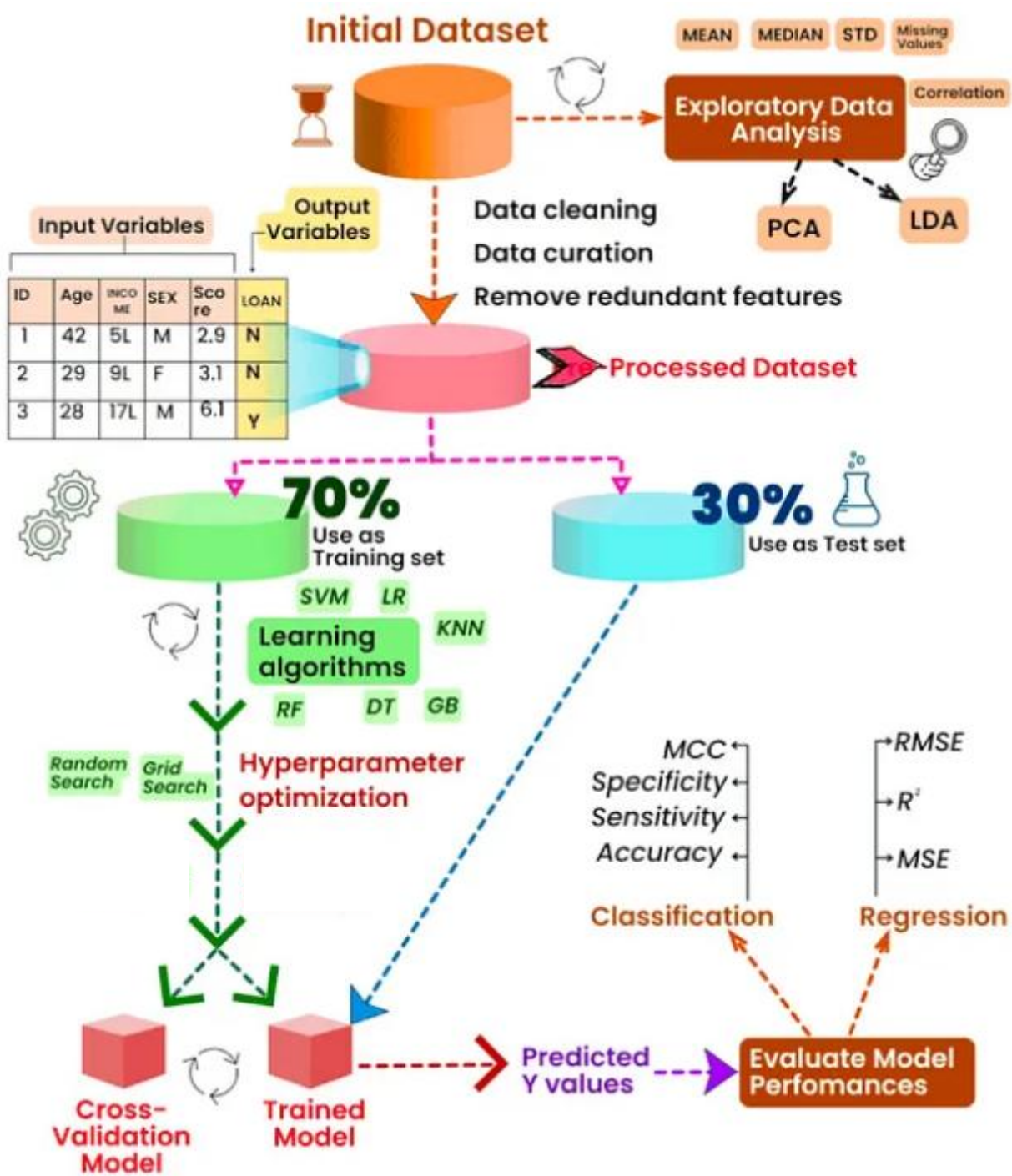
Metrics


A
Honghu training cohort
Sensitivity
1 - Specificity
AUC = 0.987
(95% CI:0.969-1.000)
B
Nanchang validation cohort
Sensitivity
1 - Specificity
AUC = 0.892
(95% CI:0.828-0.955)
C
True label
Non-severe
Severe
97.65 %
2.35 %
4.17 %
95.83 %
Non-severe
Severe
Predicted label
D
True label
Non-severe
Severe
78.46 %
21.54 %
12.50 %
87.50 %
Non-severe
Severe
Predicted label

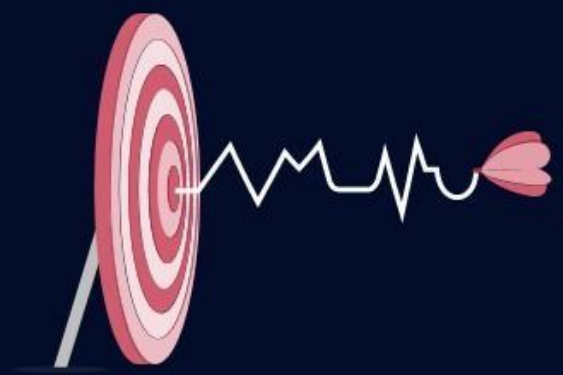
Metrics

- AREA UNDER A ROC CURVE



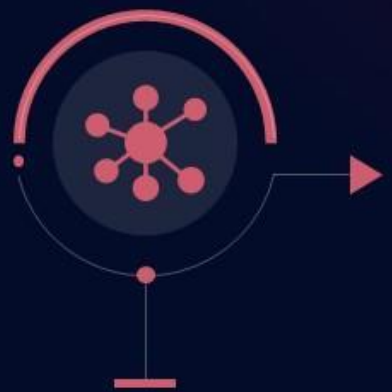

Initial Dataset
MEAN
MEDIAN
STD
Missing Values
Exploratory Data Analysis
Correlation
PCA
LDA
Input Variables
Output Variables
ID
Age
INCOME
SEX
Score
LOAN
1
42
5L
M
2.9
N
2
29
9L
F
3.1
N
3
28
17L
M
6.1
Y
Data cleaning
Data curation
Remove redundant features
Processed Dataset
70%
Use as Training set
30%
Use as Test set
SVM
LR
KNN
Learning algorithms
RF
DT
GB
Random Search
Grid Search
Hyperparameter optimization
MCC
Specificity
Sensitivity
Accuracy
RMSE
R^2
MSE
Classification
Regression
Cross-Validation Model
Trained Model
Predicted Y values
Evaluate Model Perfomances

Objectives of Machine Learning powered Medical Diagnosis



Clinical data analysis allow us to better understand diseases' underlying mechanisms and how risk factors influence their progression. The information includes everything from clinical symptoms to biochemical testing and imaging equipment outputs.

Clinical Data



Machine Learning

Diagnostic Model

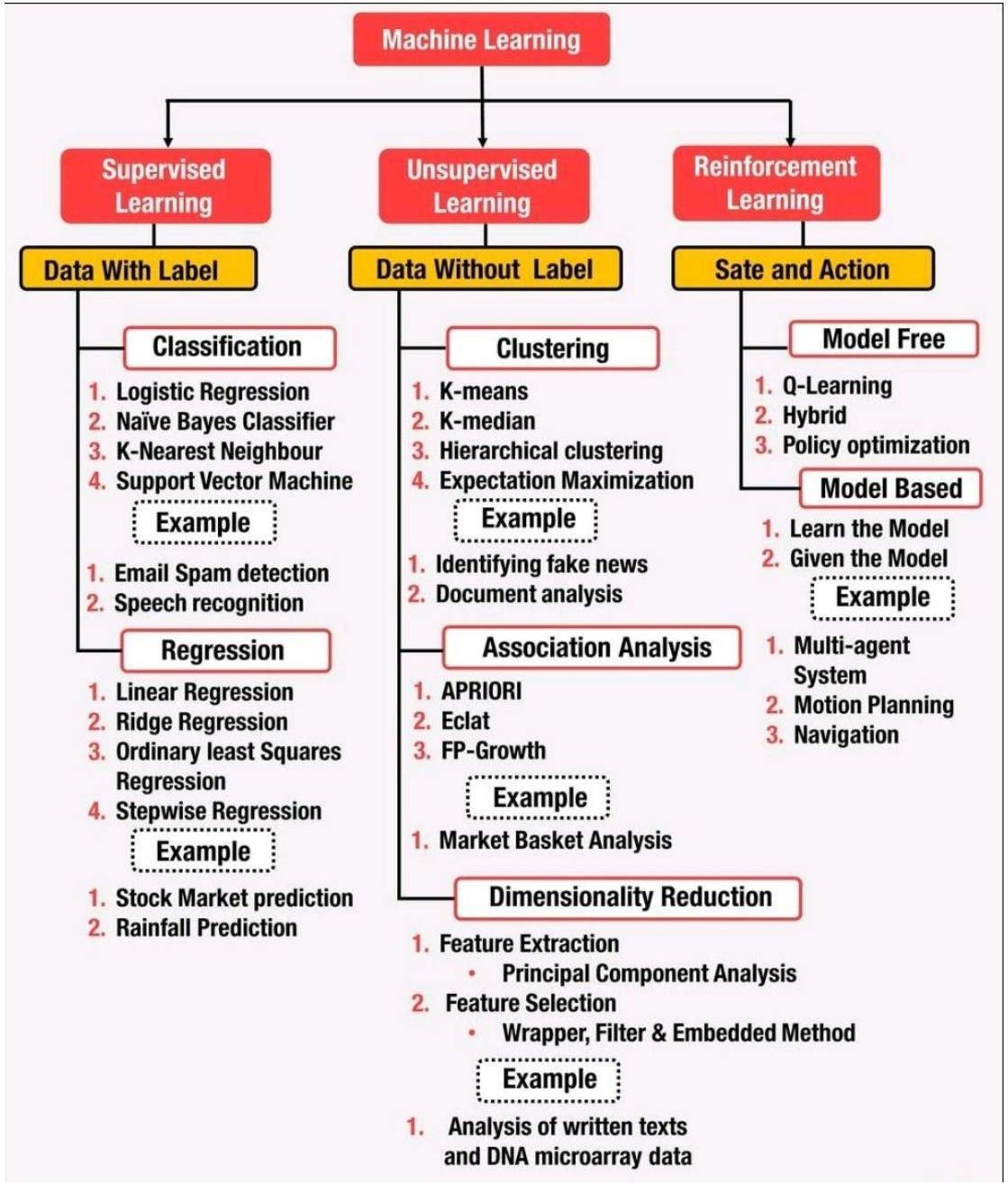
Understand the Type of Problem:

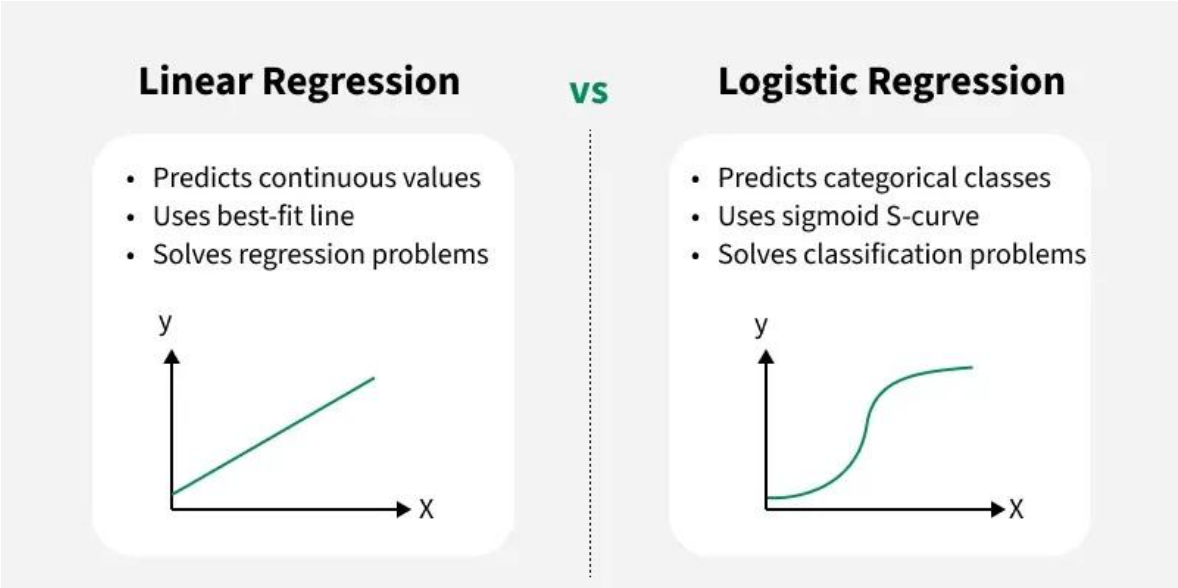
• **Classification:** If your goal is to **categorize data** into predefined labels (e.g., **spam vs. non-spam** emails), you need a **classification algorithm**.

• **Regression:** If you want to **predict a continuous value** (e.g., **house prices**), you'll need a **regression algorithm**.

• **Clustering:** If your goal is to **group similar data points** together without predefined labels (e.g., **customer segmentation**), **clustering algorithms** are suitable.

• **Dimensionality Reduction:** If you need to **simplify your data** by reducing the number of features while preserving its essence (e.g., **reducing noise** in data), **dimensionality reduction techniques** are helpful.





Training a regression algorithm is similar to training a classifier. The feature matrix is comprised of input signals such as sensor data and work orders. The regression model also requires labels, but instead of a binary (1 or 0) indicator of class ("failed" or "not failed"), the label is numeric (time to failure).

Consider the Size and Structure of Your Data:

• **Small Datasets: Simpler algorithms** like **Logistic Regression** or **K-Nearest Neighbors (KNN)** often perform well on smaller datasets.

• **Large Datasets: More complex algorithms** like **Support Vector Machines (SVM)** or **Random Forests** might be better for handling larger datasets.

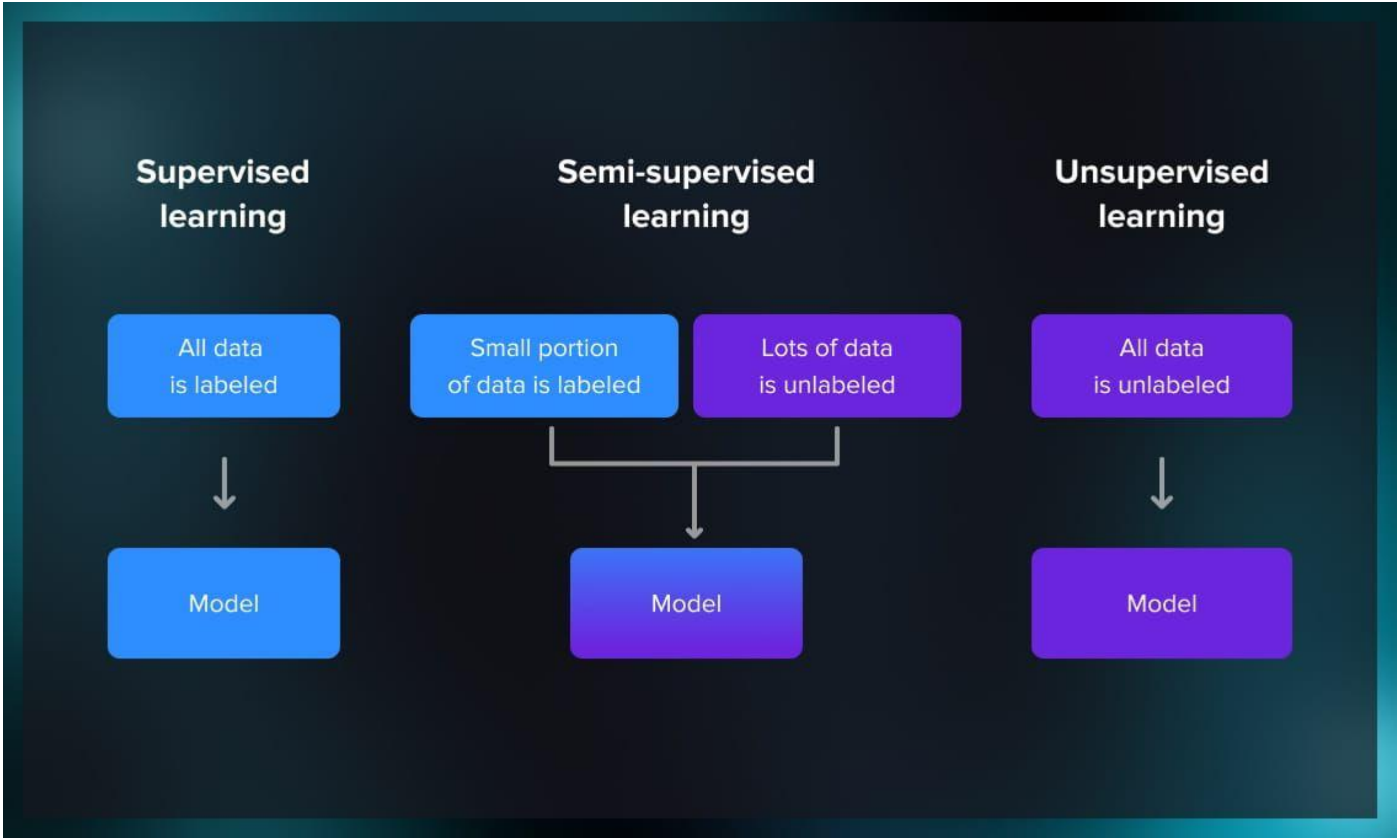
• **High-Dimensional Data:** If your data has many features (e.g., **text data** with thousands of words), consider using algorithms like **Principal Component Analysis (PCA)** for **dimensionality reduction** or **Regularized models** like **Lasso Regression**.

Evaluate the Nature of the Data:

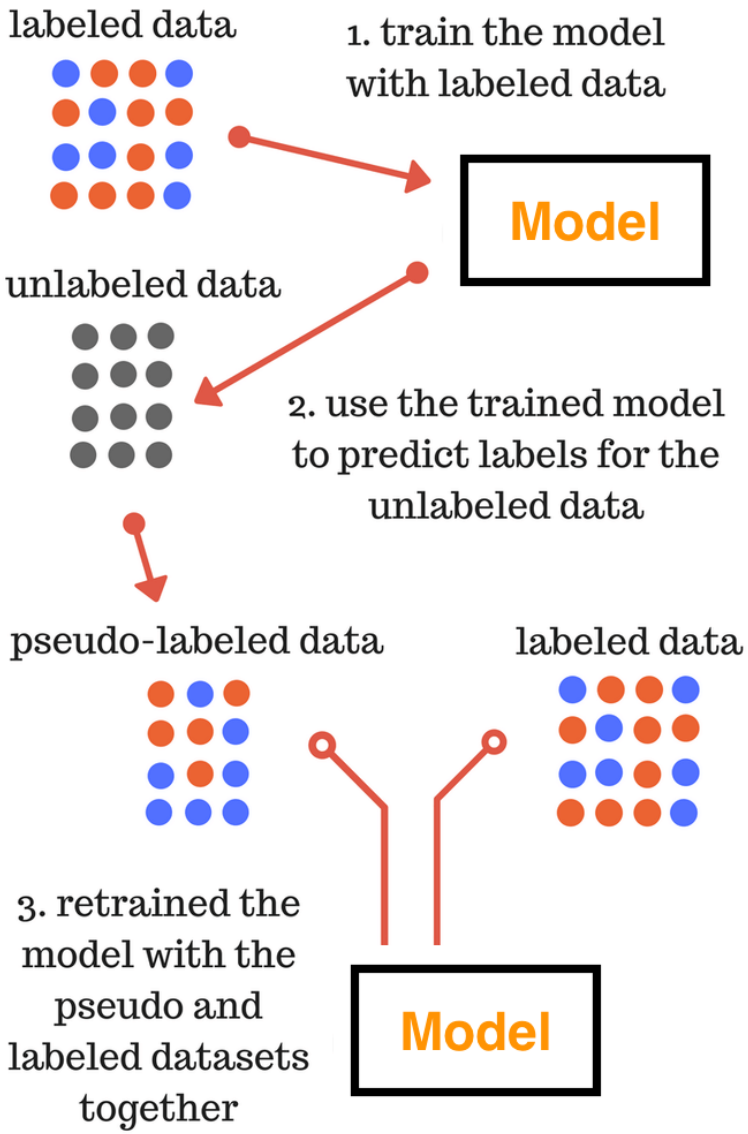
• **Labeled Data:** If your data is **labeled** (i.e., you know the correct output for each input), **supervised learning algorithms** like **Decision Trees**, **SVM**, or **Naïve Bayes** are appropriate.

• **Unlabeled Data:** For **unlabeled data**, you would use **unsupervised learning algorithms** like **K-Means Clustering** or **PCA**.

• **Complexity and Non-Linearity:** If your data shows **complex relationships** that are not **linear**, algorithms like **SVM** with a **non-linear kernel**, or **Neural Networks**, may be more effective.


Supervised learning
All data is labeled
Model
Semi-supervised learning
Small portion of data is labeled
Lots of data is unlabeled
Model
Unsupervised learning
All data is unlabeled
Model

Pseudo Labeling vs Label Label propagation



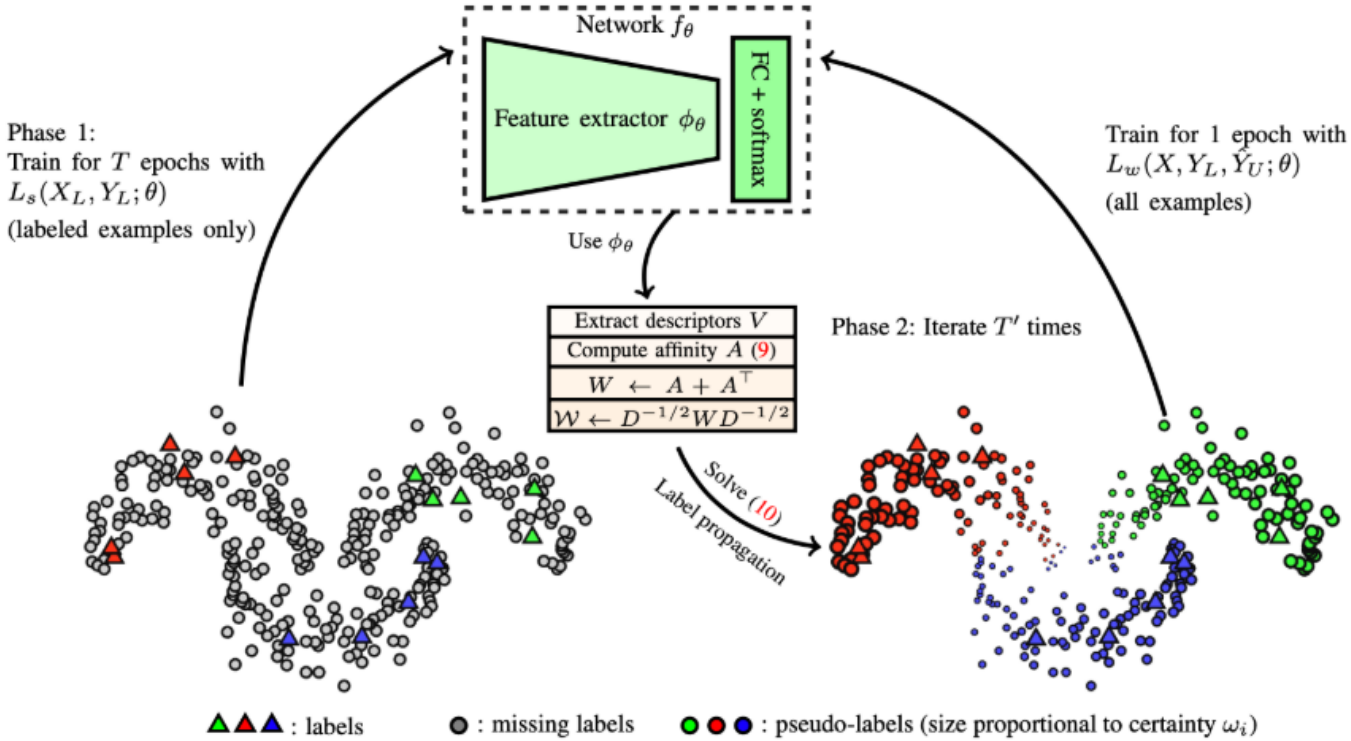


Figure 10: Illustration of how Label Propagation works. (Image source: Iscen et al. 2019)

Use cases of Machine Learning Technology

Machine Learning is broadly used in every industry and has a wide range of applications, especially that involves collecting, analyzing, and responding to large sets of data. The importance of Machine Learning can be understood by these important applications.
Some important applications in which machine learning is widely used are given below:

1.Healthcare: Machine Learning is widely used in the healthcare industry. It helps healthcare researchers to analyze data points and suggest outcomes. Natural language processing helped to give accurate insights for better results of patients. Further, machine learning has improved the treatment methods by analyzing external data on patients' conditions in terms of X-ray, Ultrasound, CT-scan, etc. NLP, medical imaging, and genetic information are key areas of machine learning that improve the diagnosis, detection, and prediction system in the healthcare sector.

2.Automation: This is one of the significant applications of machine learning that helps to make the system automated. It helps machines to perform repetitive tasks without human intervention. As a machine learning engineer and data scientist, you have the responsibilities to solve any given task multiple times with no errors. However, this is not practically possible for humans. Hence machine learning has developed various models to automate the process, having the capability of performing iterative tasks in lesser time.

3.Banking and Finance: Machine Learning is a subset of AI that uses statistical models to make accurate predictions. In the banking and finance sector, machine learning helped in many ways, such as fraud detection, portfolio management, risk management, chatbots, document analysis, high-frequency trading, mortgage underwriting, AML detection, anomaly detection, risk credit score detection, KYC processing, etc. Hence, machine learning is widely applied in the banking and finance sector to reduce error as well as time.

4.Transportation and Traffic Prediction: This is one of the most common applications of Machine Learning that is widely used by all individuals in their daily routine. It helps to ensure highly secured routes, generate accurate ETAs, predict vehicle breakdown, Driving Prescriptive Analytics, etc. Although machine learning has solved transportation problems, it still requires more improvement. Statistical machine learning algorithms helps to build a smart transportation system. Further, deep Learning explored the complex interactions of roads, highways, traffic, environmental elements, crashes, etc. Hence, machine learning technology has improved daily traffic management as well as a collection of traffic data to predict insights of routes and traffic.

5.Image Recognition: It is one of the most common applications of machine learning which is used to detect the image over the internet. Further, various social media sites such as Facebook uses image recognition for tagging the images to your Facebook friends with its feature named auto friend tagging suggestion. Further, now a day's, almost all mobile devices come with exciting face detection features. Using this feature, you can secure your mobile data with face unlocking, so if anyone tries to access your mobile device, they cannot open without face recognition.

1.Speech Recognition: Speech recognition is one of the biggest achievements of machine learning applications. It enables users to search content without writing text or, in other words, 'search by voice'. It can search content/products on YouTube, Google, Amazon, etc. platforms by your voice. This technology is referred to as speech recognition. It is a process of converting voice instructions into the text; hence it is also known as 'Speech to text' or 'Computer speech recognition. Some important examples of speech recognitions are **Google assistant, Siri, Cortana, Alexa**, etc.

2.Product Recommendation: It is one of the biggest achievements made by machine learning which helps various e-commerce and entertainment companies like Flipkart, Amazon, Netflix, etc., to digitally advertise their products over the internet. When anyone searches for any product, they start getting an advertisement for the same product while internet surfing on the same browser. This is possible by machine learning algorithms that work on users' interests or past experience and accordingly recommend them for products. For e.g., when we search for a laptop on the Amazon platform, then it also gets started with so many other laptops having the same categories and criteria. Similarly, when we use Netflix, we find some recommendations for entertainment series, movies, etc. Hence, this is also possible by machine learning algorithms.

3.Virtual Personal Assistance: This feature helps us in many ways, such as searching content using voice instruction, calling a number using voice, searching contact in your mobile, playing music, opening an email, Scheduling an appointment, etc. Now a day, you all have seen advertising like "**Alexa! Play the Music**" this is also done with the help of machine learning. Google Assistant, Alexa, Cortana, Siri, etc., are a few common applications of machine learning. These virtual personal assistants record our voice instructions, send them over to the server on a cloud, decode it using ML algorithms and act accordingly.

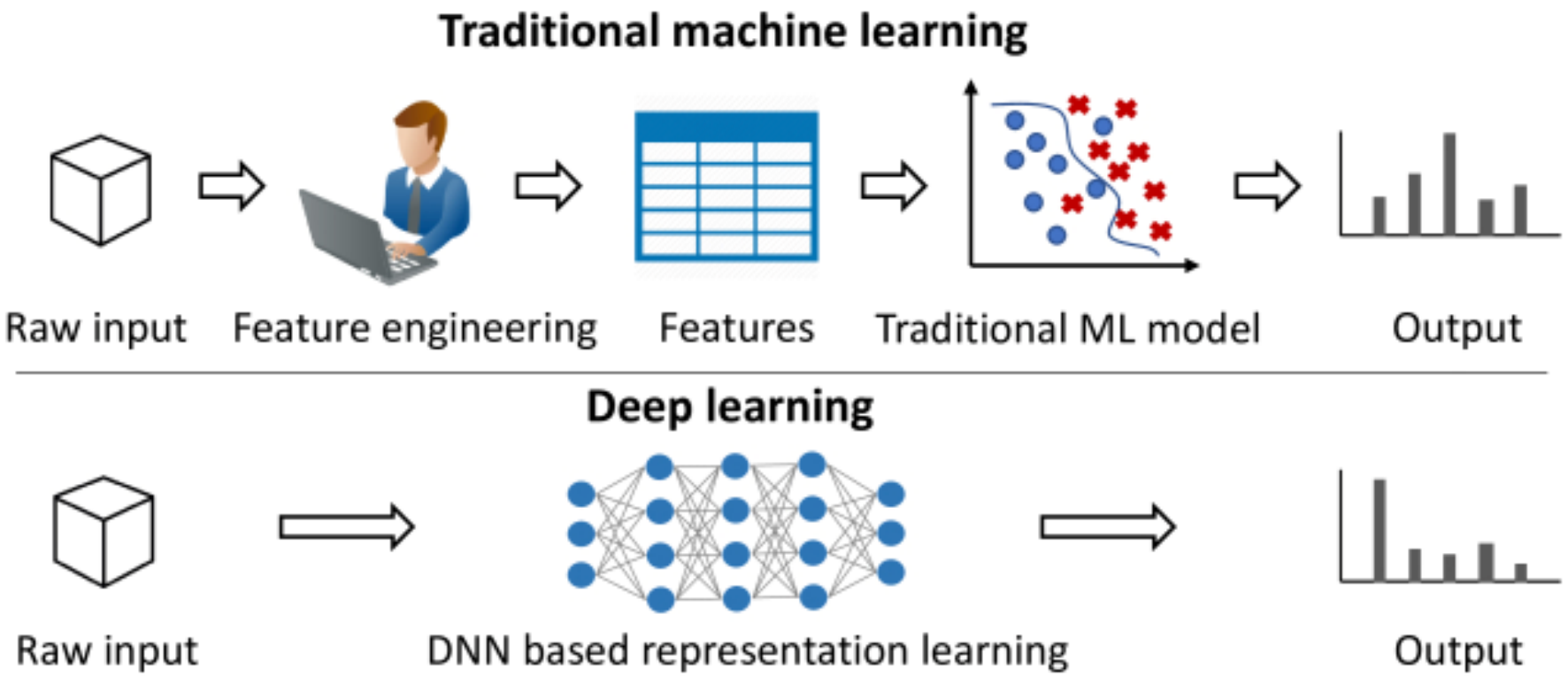
4.Email Spam and Malware detection & Filtering: Machine learning also helps us for filtering emails in different categories such as spam, important, general, etc. In this way, users can easily identify whether the email is useful or spam. This is also possible by machine learning algorithms such as **Multi-Layer Perceptron, Decision tree,** and **Naïve Bayes classifier**. Content filter, header filter, rules-based filter, permission filter, general blacklist filter, etc., are some important spam filters used by Google.

5.Self-driving cars: This is one of the most exciting applications of machine learning. Machine learning plays a vital role in the manufacturing of self-driving cars. It uses an unsupervised learning method to train car models to detect people and objects while driving. Tata and Tesla are the most popular car manufacturing companies working on self-driving cars. Hence, it is a big revolution in a technological era which is also done with the help of machine learning.

6.Credit card fraud detection: Credit card frauds have become very easy targets for online hackers. As the culture of online/digital payments is increasing, the risk of credit/debit cards is parallel increasing. Machine Learning also helps developers to detect and analyze frauds in online transactions. It develops a novel fraud detection method for Streaming Transaction Data, with an objective to analyze the past transaction details of the customers and extract the behavioral patterns. Further, cardholders are clustered into various categories with their transaction amount so that the behavioral pattern of the groups can be extracted respectively. Hence, credit card fraud detection is a novel approach using Aggregation Strategy and Feedback Mechanism of machine learning.

7.Stock Marketing and Trading: Machine learning also helps in the stock marketing and trading sector, where it uses historical trends or past experience for predicting the market risk. As share marketing is another name of marketing risk, machine learning reduces it to some extent and predicts data against marketing risk. Machine learning's **long short-term neural memory network** is used for the prediction of stock market trends.

8.Language Translation: The use of Machine learning can be seen in language translation. It uses the sequence-to-sequence learning algorithms for translating one language into other. Further, it also uses images recognition techniques to identify the text from one language to other. Similarly, Google's GNMT (Google Neural Machine Translation) provides this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it is called automatic translation.


Traditional machine learning
Raw input
Feature engineering
Features
Traditional ML model
Output
Deep learning
Raw input
DNN based representation learning
Output

?

TP5

Finalization Statistical Project

Deliverable for statistical Part:

Code – you do not need to submit a writing report
You can submit a zip file with all the scripts that you want to submit, or just one script that contains all the work.

If you are using a particular dataset, please also submit the data.
All the files submitted should be able to run and replicate your experience.

Make comments on the code that explain: which objective;
why you are using the method and not another one;
try to make a critical analysis of the results;
if possible, take conclusions.

Graphs with axis information.