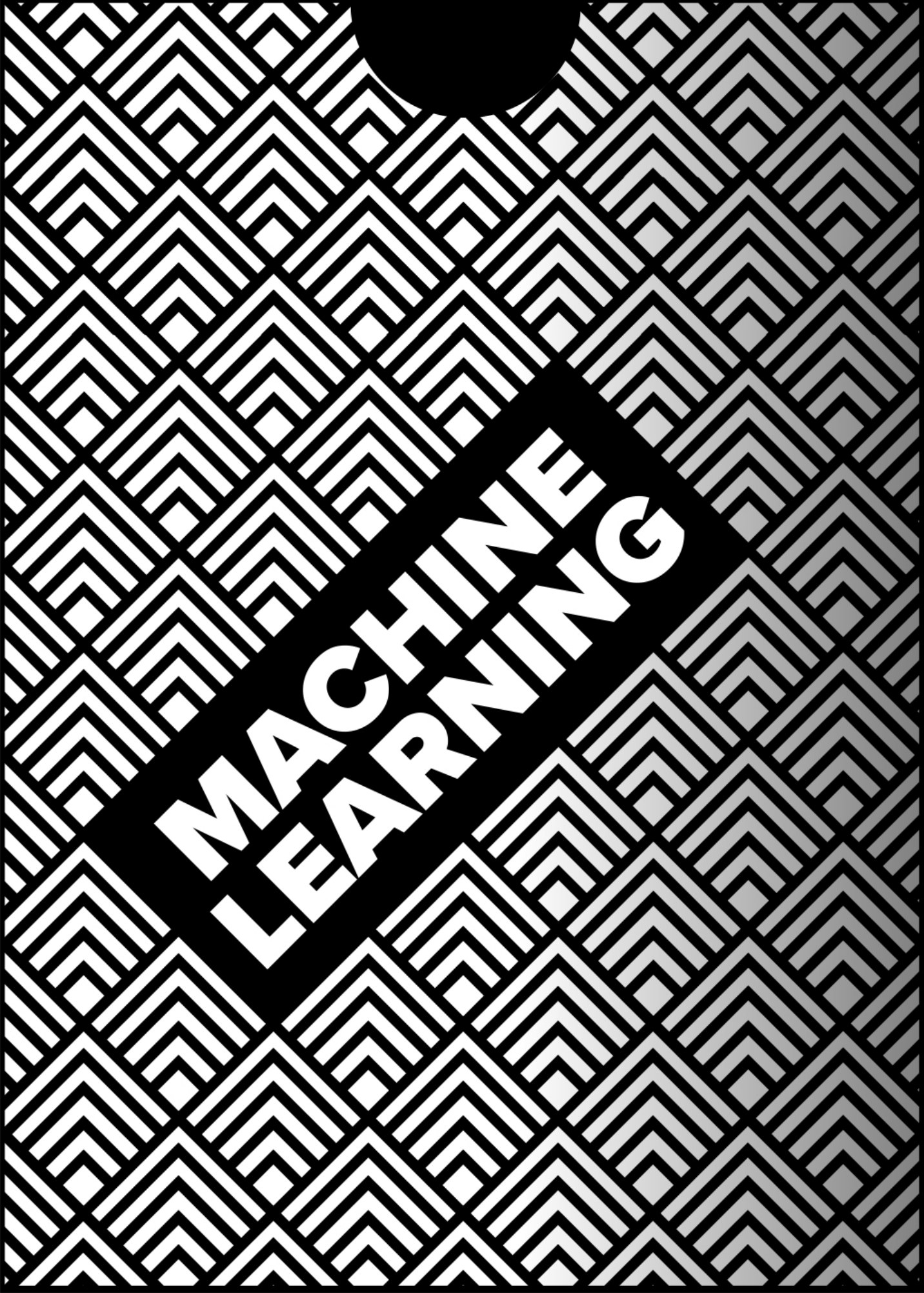

MACHINE
LEARNING

MACHINE LEARNING

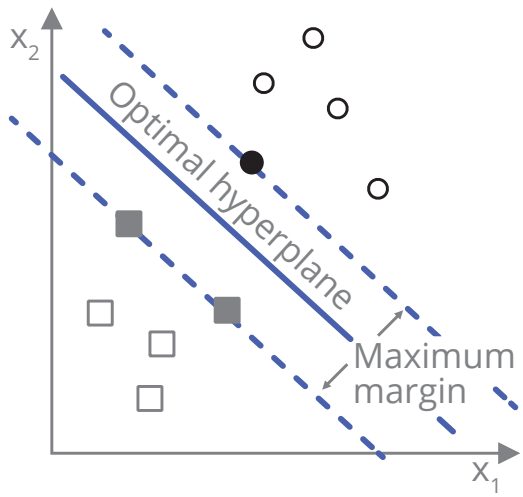
softserve

Support Vector Machines

R AD C

ALGORITHM FAMILY

SVMs are supervised learning algorithms which can be used for both classification or regression problems (except OneClassSVM which is unsupervised). Given labeled training data, the algorithms output an optimal hyperplane which categorizes new examples.



CHARACTERISTICS:

- ★ Big Data — computation and memory-intensive while training, data sampling can be used as a workaround
- ★★ Small Data — good accuracy even for small # of observations
- ★★★ Imbalanced Data — compensates imbalance through class weights, works fine out of the box for moderately imbalanced data
- ★★ Results Interpretation — applicable for regulated fields (i.e. credit scoring)
- ★ Online Learning — most implementations only support batch setting
- ★★ Ease of Use — moderate number of parameters.

IMPLEMENTATIONS:

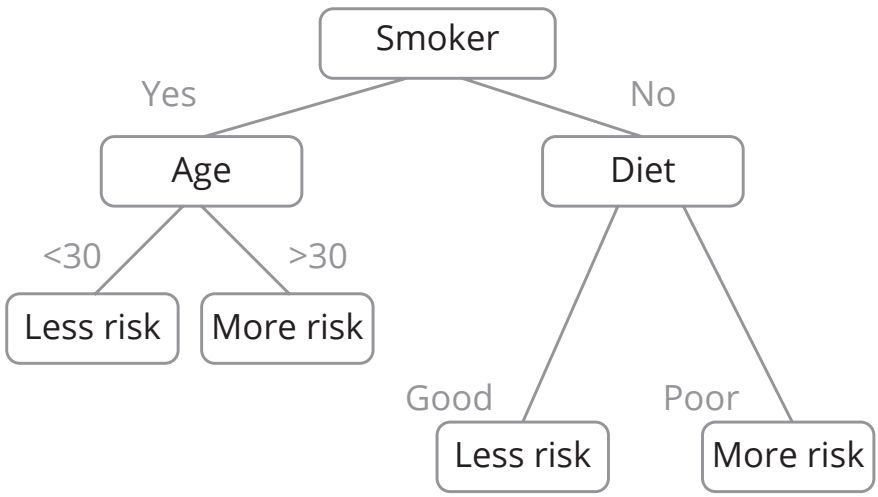
Linear SVM, Non-Linear SVM, One-Class SVM.

Decision Tree Learning

R AD C

ALGORITHM FAMILY

Decision Tree Learning uses a decision tree structure to go from observations about an item to conclusions about the item's target value. It is one of the most interpretable families of machine learning algorithms. This approach can be used for both classification or regression problems.



CHARACTERISTICS:

★★ Big Data — interpretability is getting worse on large datasets

★★ Small Data — sufficient generalization even for very small dataset, but can lead to overfitting

★★ Imbalanced Data — can be handled by stratified bootstrap technique

★★★ Results Interpretation — represented by a set of decision rules

★★★ Online Learning — can be trained sequentially

★★★ Ease of Use — models tuning is user-friendly

ALGORITHMS:

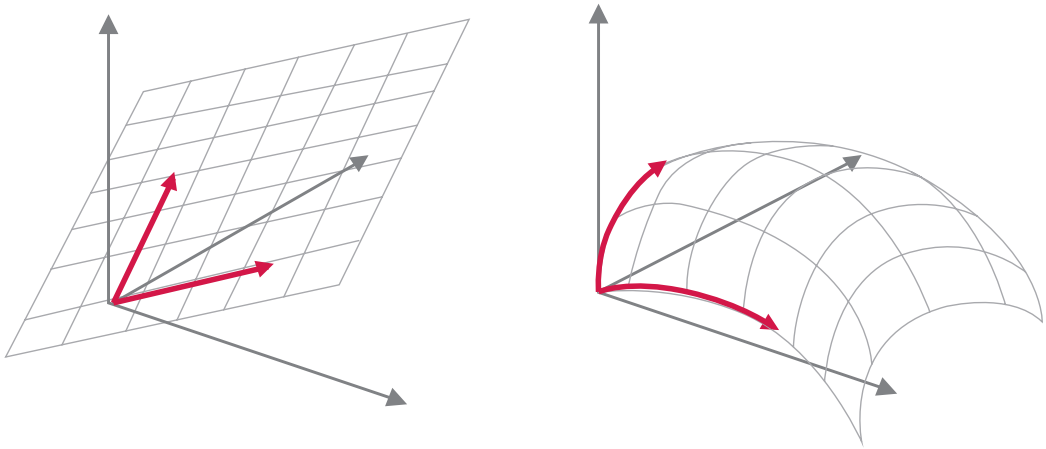
Classification/Regression Decision Tree, Random Forest, Isolation Forest.

Generalized Linear Models

R C

ALGORITHM FAMILY

A class of linear models with a common property — the dependent variable is linearly related to the factors and covariates via a specified link function. Generalized Linear Models are proposed as a way of unifying various statistical models.



CHARACTERISTICS:

★★★ Big Data — there are a lot of implementations for large datasets, that include iterative approaches for solving linear equation systems

★★★ Small Data — overfitting resistance due to representing statistically significant dependencies only

★ Imbalanced Data — poor, can be handled by applying imbalance compensations techniques, e.g. complex sampling

★★★ Results Interpretation — results drivers can be easily extracted

★★ Online Learning — can be trained with gradient methods and mini batching

★★★ Ease of Use — models tuning is user-friendly

ALGORITHMS:

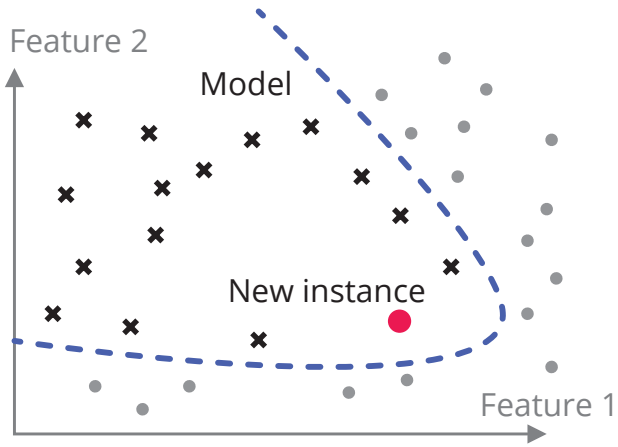
Logistic Regression, Bayesian Naive Classifier, Linear Regression, Bayesian Linear Regression.

Instance-Based Learning

R C

ALGORITHM FAMILY

Instance-Based Learning (sometimes called memory-based learning) is a family of learning algorithms that, instead of performing explicit generalization, compares new problem instances with instances seen in training and stored in memory. This approach can be used for both classification or regression problems.



CHARACTERISTICS:

- ★ Big Data — generally, the application of this algorithm family for big data is not feasible due to memory and prediction speed restrictions
- ★★★ Small Data — good accuracy, even for a small number of observations
- ★★ Imbalanced Data — more robust for data with unbalanced classes and is efficient for multiclass classification with a small number of features
- ★★ Results Interpretation — transparent inference; process, but without the possibility of getting explanation rules
- ★★★ Online Learning — can be trained sequentially
- ★★★ Ease of Use — models tuning is user-friendly

ALGORITHMS:

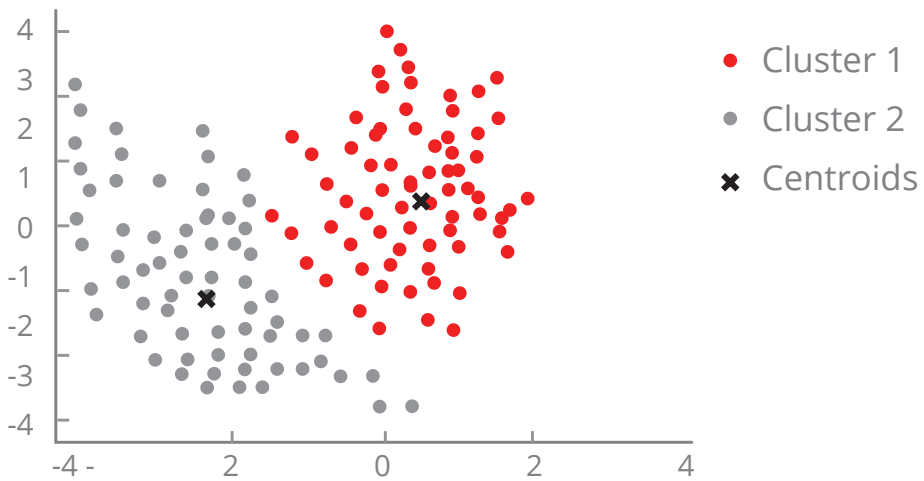
K-Nearest Neighbors, Radius Neighbors.

Centroid-Based Clustering

CL

ALGORITHM FAMILY

Centroid based models represent clusters by a central vector, which does not need to be an actual object. The analyst needs to define the number of clusters in advance and the clustering algorithm then targets the optimization problem of finding k centers and assigning objects to the nearest center, in a way that minimizes the squared distances.



CHARACTERISTICS:

- ★★★ Big Data — can handle big data well, but it's only fast for low dimensional data
- ★ Noise Resistance — sensitive to noisy data
- ★★★ Interpretability — clustering results easy to interpret, except unimodal distribution cases
- ★★★ Online Learning — can be trained sequentially
- ★★ Data Flexibility — extremely difficult due to limited set of distance metrics and their properties

ALGORITHMS:

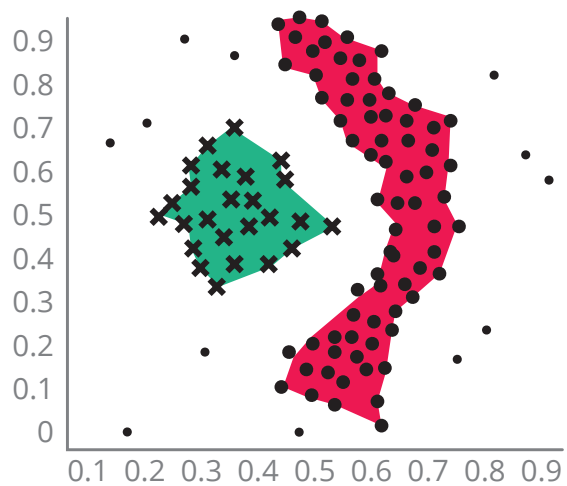
K-Means Clustering.

Density-Based Clustering

CL

ALGORITHM FAMILY

In density-based clustering, clusters are defined as areas of higher density in comparison with the rest of the dataset. Objects, which do not belong to the high-density clusters, are usually considered to be noise and border points.



CHARACTERISTICS:

- ★★ Big Data — depends on implementation, worse case scenarios are characterized by sub-quadratic time or memory complexity
- ★★★ Noise Resistance — has a notion of noise, and is robust to outliers
- ★★★ Interpretability — clusters are characterized by a higher density than the remainder of the dataset
- ★★★ Online Learning — can be trained sequentially
- ★★ Data Flexibility — possible, but in the case when a symmetric distance function can be defined

ALGORITHMS:

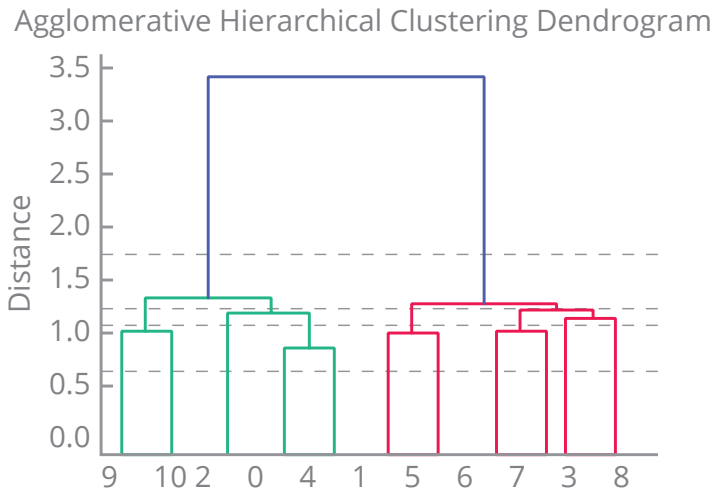
DBSCAN.

Hierarchical Clustering

CL

ALGORITHM FAMILY

Also called connectivity based clustering, this family is based on the idea that objects are more related to nearby objects than those further away. The formation of clusters usually is represented by a dendrogram. There are two approaches to this: agglomerative ("bottom up") and divisive ("top down").



CHARACTERISTICS:

- ★ Big Data — distance matrix is computationally and memory intense
- ★★ Noise Resistance — noisy observations can be separated in small clusters
- ★★★ Interpretability — high, as clusters extraction is intuitive and clear
- ★ Online Learning — requires recalculating distance matrix and performing agglomeration from scratch
- ★★ Data Flexibility — single/complete approaches can detect data structures of different complexity

ALGORITHMS:

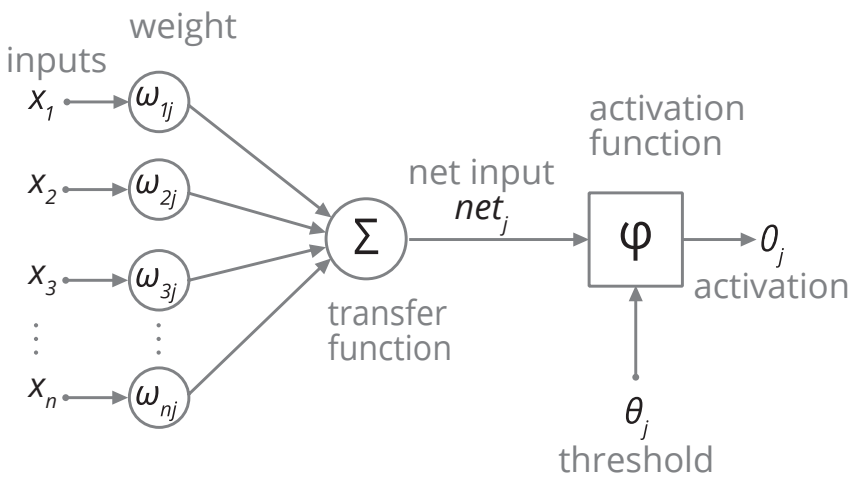
Single-Linkage Clustering, Complete-Linkage Clustering, Average-Linkage Clustering.

Artificial Neural Network

R AD C CL

ALGORITHM FAMILY

ANN is a computational model based on the structure and functions of biological neural networks. It can be used for both classification and regression as well as unsupervised learning. The model works well for unstructured data and complex non-linear relationships but usually is computationally expensive.



CHARACTERISTICS:

★★★ Big Data — performance positively correlates with data volume, but at the expense of computational resources

★ Small Data — the less data available, the bigger chance other algorithms will outperform neural networks

★★ Imbalanced Data — class imbalance increases the possibility of overfitting, but it can be mitigated with class weights

★ Results Interpretation — usually difficult to interpret;

★★★ Online Learning — can be trained sequentially

★ Ease of Use — requires substantial understanding of the ANN field

ALGORITHMS:

Feedforward ANN (Multilayer Perceptron).

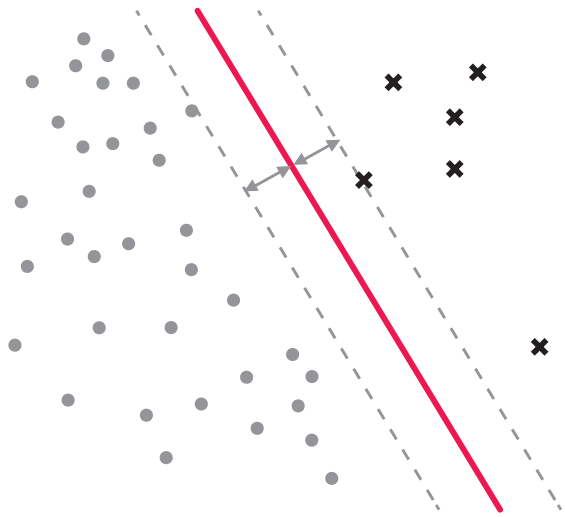
MACHINE LEARNING

softserve

Linear SVM

SUPPORT VECTOR MACHINES | ALGORITHM

An SVM algorithm with a linear kernel.



CHARACTERISTICS:

- ★½ Accuracy — depends on data nature (linear separability is required)
- ★★ Training Speed — linear dependency on a training dataset size
- ★★★ Prediction Speed — depends on # of support vectors and features
- ★★ Overfitting Resistance — be careful with model hyperparameters
- ★ Probabilistic Interpretation — can be calculated using an expensive cross-validation

TIPS:

- Great performance in high-dimensional space
- Works well in case when # of features is larger than # of observations
- Margins maximization provides good robustness.

IMPLEMENTATIONS:

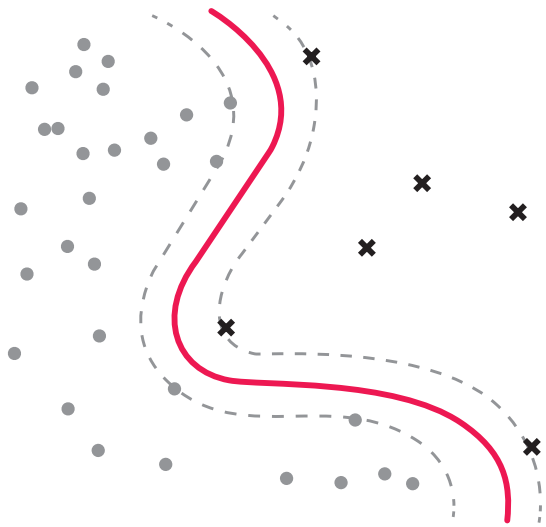
R, Python (scikit-learn), Spark (mllib), Tensorflow.

Non-Linear SVM

R C

SUPPORT VECTOR MACHINES | ALGORITHM

An SVM algorithm with a non-linear kernel like Gaussian (e.g. RBF).



CHARACTERISTICS:

★★★ Accuracy — high accuracy for most tasks when tuning is done well

★ Training Speed — training time is high on large datasets;

★★ Prediction Speed — depends on # of support vectors and features

★★ Overfitting Resistance — theoretically, SVMs should be highly resistant to overfitting, but in practice it depends on kernel parameters

★ Probabilistic Interpretation — can be calculated using an expensive cross-validation

TIPS:

- Great performance in high-dimensional space
- Expert knowledge about the problem can be built into a model by engineering the kernel
- Deep expertise in machine learning is required

IMPLEMENTATIONS:

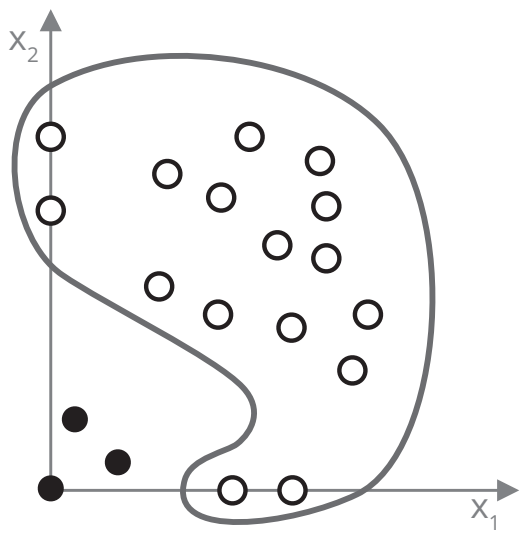
R, Python (scikit-learn).

One-Class SVM

AD

SUPPORT VECTOR MACHINES | ALGORITHM

One-class SVM is an unsupervised algorithm that learns a decision function for novelty detection: classifying new data as similar or different to the training set.



CHARACTERISTICS:

- ★ Training Speed — quadratic dependence on the dataset size
- ★★ Prediction Speed — linear dependence on # of processed observations, can vary on # of support vectors and features
- ★★ Noise Sensitivity — less robust against noise in the training dataset in comparison with other algorithms
- ★★★ Distribution Invariance — good with any kind of data distribution

TIPS:

- Great performance in high-dimensional space
- Can be used to create an anomaly detection model

IMPLEMENTATIONS:

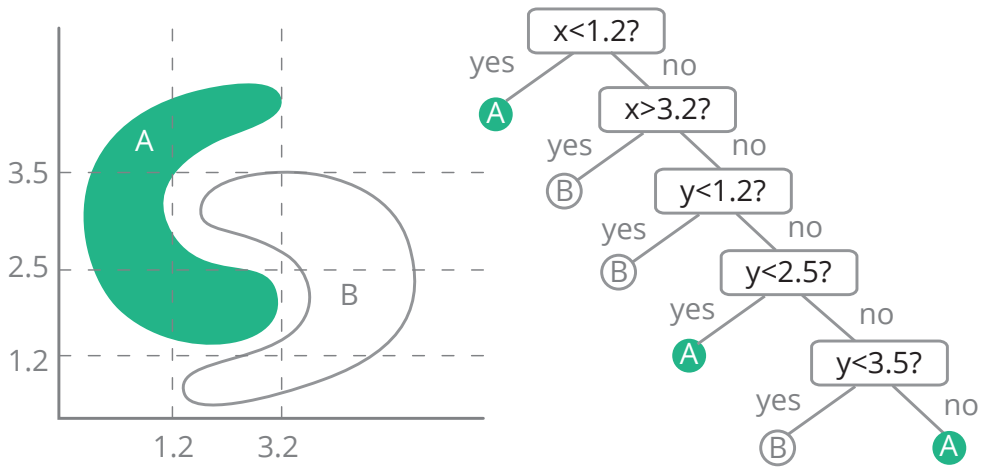
R, Python (scikit-learn).

Classification/Regression Trees

R C

DECISION TREE LEARNING | ALGORITHM

A non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.



CHARACTERISTICS:

- ★★ Accuracy — depends heavily on # of features; a tree with few samples in high-dimensional space is very likely to overfit
- ★★★ Training Speed — heuristic algorithms provide the linear dependency on a training dataset size
- ★★★ Prediction Speed — depends on the tree size
- ★ Overfitting Resistance — tends to create over-complex trees that don't generalize well from the training data
- ★★ Probabilistic Interpretation — can be extracted

TIPS:

- Naturally works with both categorical and numerical data
- The most interpretable results

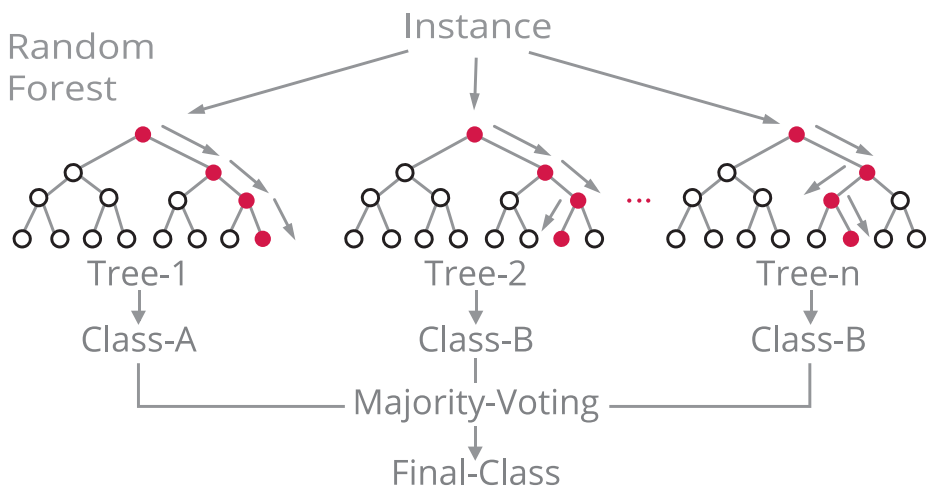
IMPLEMENTATIONS:

R, Python (scikit-learn), Spark (mllib).

Random Forest

DECISION TREE LEARNING | ALGORITHM

An ensemble learning method that constructs a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It has improved predictive accuracy and controls over-fitting.



CHARACTERISTICS:

★★★ Accuracy — due to bootstrap aggregation of independent trees

★★½ Training Speed — depends on model complexity, but it can be parallelized

★★½ Prediction Speed — depends on model complexity but it can be improved due to independent trees

★★★ Overfitting Resistance — complexity doesn't lead to overfitting

★★ Probabilistic Interpretation — can be extracted

TIPS:

- Naturally works with both categorical and numerical data
- High accuracy without extensive tuning

IMPLEMENTATIONS:

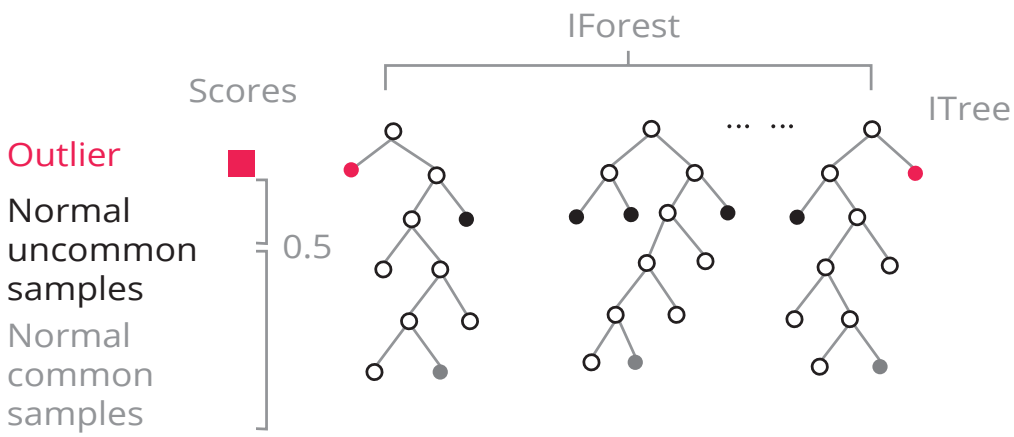
R, Python (scikit-learn), Spark (mllib), Tensorflow.

Isolation Forest

AD

DECISION TREE LEARNING | ALGORITHM

Isolation Forest (IForest) is an unsupervised algorithm that learns set of trees, which give an ability to classify new observation as an anomaly or not. The algorithm expects that outliers are easier to isolate from rest, so “tree path” length is used as a criterion of separation.



CHARACTERISTICS:

- ★★ Training Speed — linear dependence on # of observations, can vary on data/model complexity, easily parallelized
- ★★★ Prediction Speed — almost linear, depends on # of observations
- ★★ Noise Sensitivity — IForest is built using trees, but bootstrapping improves its resistance to noise
- ★★★ Distribution Invariance — good with any kind of data distribution

TIPS:

- Can achieve high detection performance with high-dimensional problems that have a large number of irrelevant attributes
- Works well even with anomalies in the training set

IMPLEMENTATIONS:

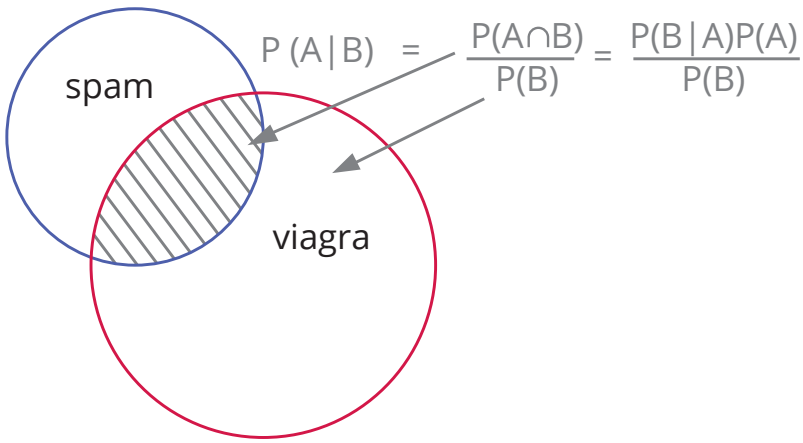
R, Python (scikit-learn).

Bayesian Naive Classifier

C

GENERALIZED LINEAR MODELS | ALGORITHM

A simple probabilistic classifier based on Bayes' Theorem with an assumption of strong independence (naive) among features.



CHARACTERISTICS:

★★½ Accuracy — a big dataset is required to make reliable estimations of the probability of each class

★★★ Training Speed — converges toward its asymptotic faster than other methods, like logistic regression

★★★ Prediction Speed — depends on # of classes

★★★ Overfitting Resistance — great generalization to unseen data based on its training set, due to a very simple hypothesis function

★★ Probabilistic Interpretation — classes probabilities are naturally estimated for binary classification

TIPS:

- Can handle missing data
- Converges quicker than discriminative models like logistic regression, so you need less training data

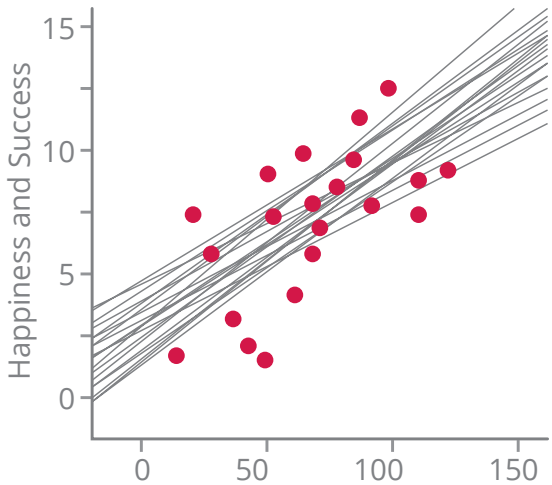
IMPLEMENTATIONS:

R, Python (scikit-learn), Spark (mllib).

Bayesian Linear Regression

GENERALIZED LINEAR MODELS | ALGORITHM

Bayesian linear regression models treat regression coefficients and the disturbance variance as random variables, rather than fixed but unknown quantities. This assumption leads to a more flexible model and intuitive inferences.



CHARACTERISTICS:

★★½ Accuracy — determined by the confidence level of estimated distribution parameters

★★ Training Speed — depends on the model complexity

★★★ Prediction Speed — similar to a linear regression model

★★ Overfitting Resistance — due to prior probabilities, it's possible to constrain model parameters by some more probable value, inferred from the overall data

TIPS:

- More accurate in small samples
- Can incorporate prior information

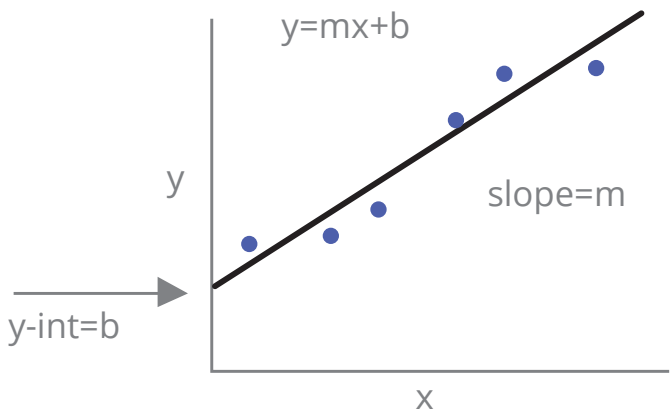
IMPLEMENTATIONS:

R, Python (scikit-learn), Tensorflow.

Linear Regression

GENERALIZED LINEAR MODELS | ALGORITHM

A linear approach for modelling relationship between variables. Used for predicting continuous output, based on one or multiple inputs. The algorithm is simple and fast, often used at the early exploration stage, but could be overly simplistic.



CHARACTERISTICS:

★★½ Accuracy — limited due to only linear relations in data

★★★ Training Speed — linear dependence on the training dataset size and # of features

★★★ Prediction Speed — prediction can be performed as multiplication of two matrixes, so it is fast and easy to parallelize

★ Overfitting Resistance — very sensitive to outliers and # of features; recommended outliers treatment and regularization techniques

TIPS:

- In some realizations categorical variables should be transformed to a set of binary ones to be used in the algorithm
- Results are interpretable as features influences are represented by learned coefficients

IMPLEMENTATIONS:

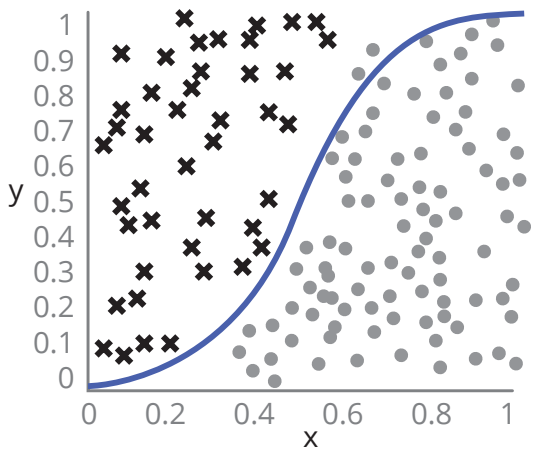
R, Python (scikit-learn), Spark (mllib), Tensorflow.

Logistic Regression

C

GENERALIZED LINEAR MODELS | ALGORITHM

A powerful tool for two-class and multiclass classification. As a linear regression, it is fast and simple.



CHARACTERISTICS:

- ★★ Accuracy — separates observations into two groups. If observations are not linearly separable, it's impossible to achieve high accuracy
- ★★★ Training Speed — fast for binary classification, it can also be trained with a large dataset using online training approach
- ★★★ Prediction Speed — predictions can be extracted relatively fast and computationally easy
- ★★ Overfitting Resistance — considered robust to outliers. Be careful with # of features. Recommended regularization techniques
- ★★ Probabilistic Interpretation — extracted by weighting predictions

TIPS:

- Multiclass-classification is done using one-vs-all or one-vs-one
- Results are interpretable as the influence of the features are; represented by learned coefficients

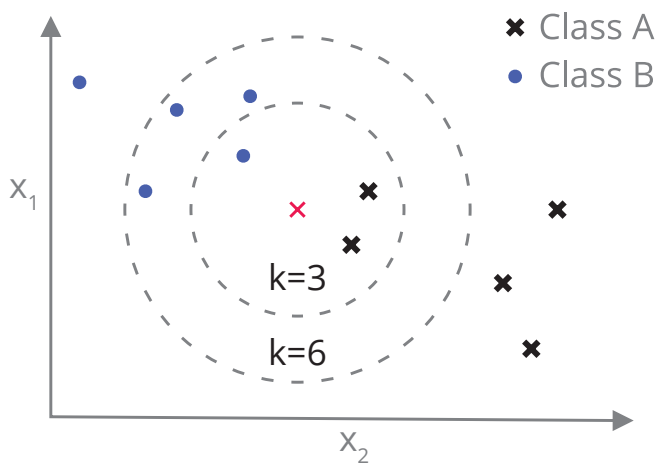
IMPLEMENTATIONS:

R, Python (scikit-learn), Spark (mllib), Tensorflow.

K-Nearest Neighbors (KNN)

INSTANCE-BASED LEARNING | ALGORITHM

A non-parametric supervised learning method used for classification and regression; a type of lazy learning, where the function is only approximated locally and all computation is deferred until classification.



CHARACTERISTICS:

- ★★ Accuracy — sufficient accuracy for most tasks, but there is a tradeoff between accuracy vs avoiding overfitting
- ★★ Training Speed — training time is high on large datasets
- ★ Prediction Speed — full training set processing is required
- ★★ Overfitting Resistance — with an increase of k nearest training objects, the probability of overfitting decreases
- ★★★ Probabilistic Interpretation — naturally determined by the inference process

TIPS:

- One of the simplest machine learning algorithms
- Good choice for low dimensional space

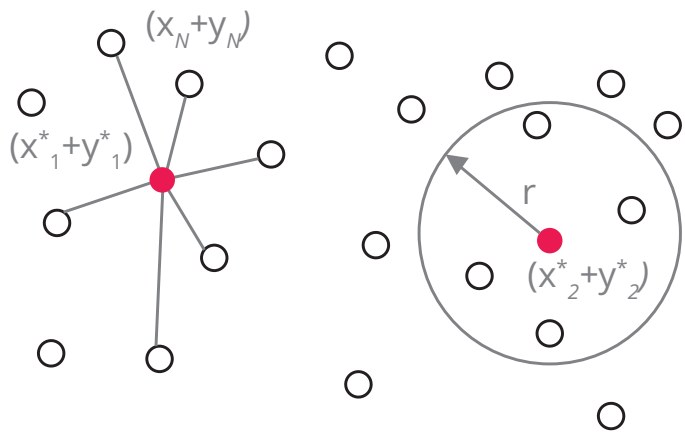
IMPLEMENTATIONS:

R, Python (scikit-learn), Tensorflow.

Radius Neighbors

INSTANCE-BASED LEARNING | ALGORITHM

An alternative to the KNN algorithm, wherein the nearest neighbor is determined by a radius hyper-parameter. Also is used for classification and regression tasks.



CHARACTERISTICS:

- ★★ Accuracy — sufficient accuracy for most tasks, but there is a tradeoff between accuracy vs avoiding overfitting
- ★★ Training Speed — training time is high on large datasets
- ★ Prediction Speed — full training set processing is required
- ★★ Overfitting Resistance — with an increase of radius the probability of overfitting decreases
- ★★★ Probabilistic Interpretation — naturally determined by the inference process

TIPS:

- One of the simplest machine learning algorithms
- Good choice for data that isn't sampled uniformly
- For high-dimensional spaces, this method is less effective due to so-called "curse of dimensionality"

IMPLEMENTATIONS:

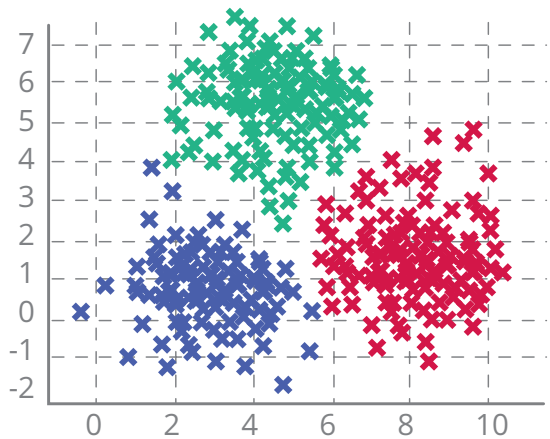
R, Python (scikit-learn).

K-Means Clustering

CL

CENTROID-BASED CLUSTERING | ALGORITHM

A method of vector quantization, popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.



CHARACTERISTICS:

- ★★ Time Complexity — very fast but falls in local minima, needs restarting
- ★★ Memory Complexity — sub-quadratic complexity
- ★ Deterministic Results — depends on the initial choice of cluster centers
- ★★ Tuning Complexity — user friendly, with small number of hyper parameters

TIPS:

- The result of the algorithm is the cluster centers
- Forms of clusters can only be hyperspheres

IMPLEMENTATIONS:

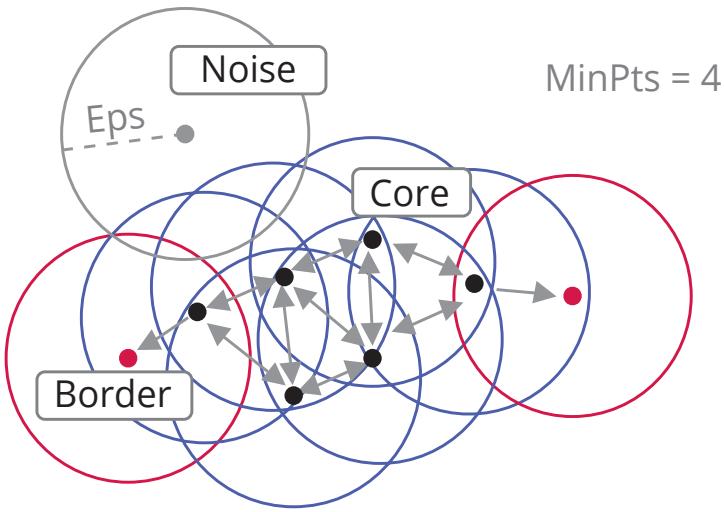
R, Python (scikit-learn), Spark (mllib), Tensorflow.

DBSCAN

CL

DENSITY-BASED CLUSTERING | ALGORITHM

The most popular density-based clustering method. DBSCAN is based on the so-called “density-reachability” cluster model. It connects points that satisfy a density criterion like a minimum number of other objects within a defined radius.



CHARACTERISTICS:

- ★ Time Complexity — requires a linear number of range queries on the database
- ★★ Memory Complexity — sub-quadratic complexity
- ★★ Deterministic Results — border points that are reachable from more than one cluster can be part of either cluster, depending on the order in which the data is processed
- ★★★ Tuning Complexity — algorithm parameters can be set by a domain expert, if the data is well understood

TIPS:

- Does not require # of clusters as a parameter
- Good results for non-sphere clusters with equal density

IMPLEMENTATIONS:

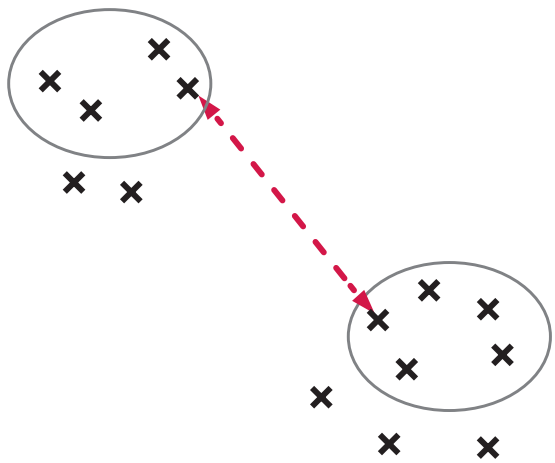
R, Python (scikit-learn).

Single-Linkage Clustering

CL

HIERARCHICAL CLUSTERING | ALGORITHM

An agglomerative clustering approach, when at each step two clusters are combined if the distance between the closest observations is the lowest among all clusters combinations.



CHARACTERISTICS:

- ★★½ Time complexity — slow when building a distance matrix, especially with high-dimensional data and many observations
- ★★ Memory complexity — quadratic complexity
- ★★★ Deterministic results — agglomerate approach is clearly defined
- ★★ Tuning complexity — dendrogram can be split by # of groups or height

TIPS:

- Catches complicated data structures
- Tends to produce long thin clusters, which may lead to difficulties using the algorithm
- Useful in anomaly detection

IMPLEMENTATIONS:

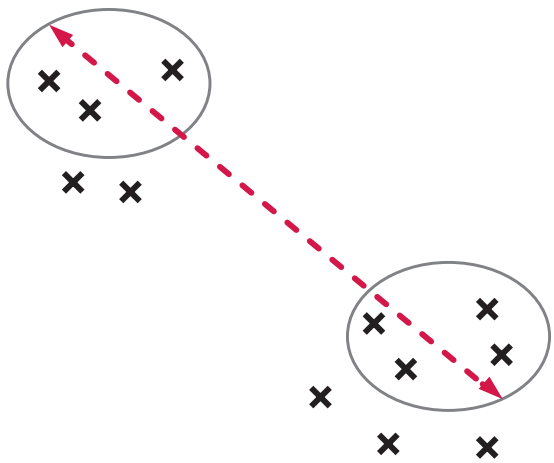
R, Python (scikit-learn).

Complete-Linkage Clustering

CL

HIERARCHICAL CLUSTERING | ALGORITHM

An agglomerative clustering approach, when at each step two clusters are combined if the distance between the farthest observations is the lowest among all clusters combinations.



CHARACTERISTICS:

- ★★½ Time complexity — slow when building a distance matrix, especially with high-dimensional data and many observations
- ★★ Memory complexity — quadratic complexity;
- ★★★ Deterministic results — agglomerate approach is clearly defined
- ★★ Tuning complexity — dendrogram can be split by # of groups or height

TIPS:

- Catches complicated data structures
- Avoids the drawback of singe-linkage clustering (chaining phenomenon)
- Tends to find clusters with approximately equal diameters

IMPLEMENTATIONS:

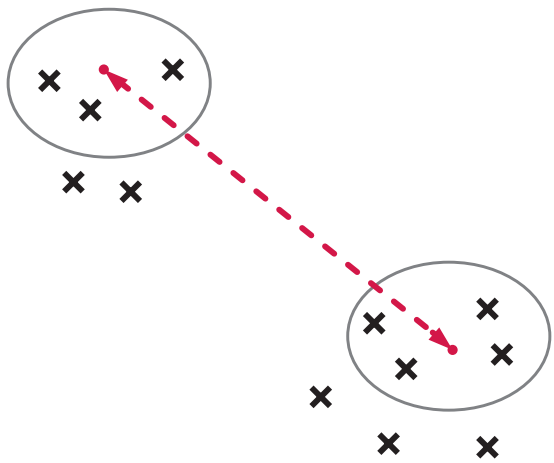
R, Python (scikit-learn).

Average-Linkage Clustering

CL

HIERARCHICAL CLUSTERING | ALGORITHM

An agglomerative clustering approach, when at each step two clusters are combined if the average distance between all data points of these clusters is the lowest among all clusters combinations.



CHARACTERISTICS:

- ★ Time complexity — slower than single or complete linkage-algorithms as more operations are needed to compute distances
- ★★ Memory complexity — quadratic complexity
- ★★★ Deterministic results — agglomerate approach is clearly defined
- ★★ Tuning complexity — dendrogram could be split by # of groups or height

TIPS:

- All points from clusters are contributing equally, compared with single/complete linkages

IMPLEMENTATIONS:

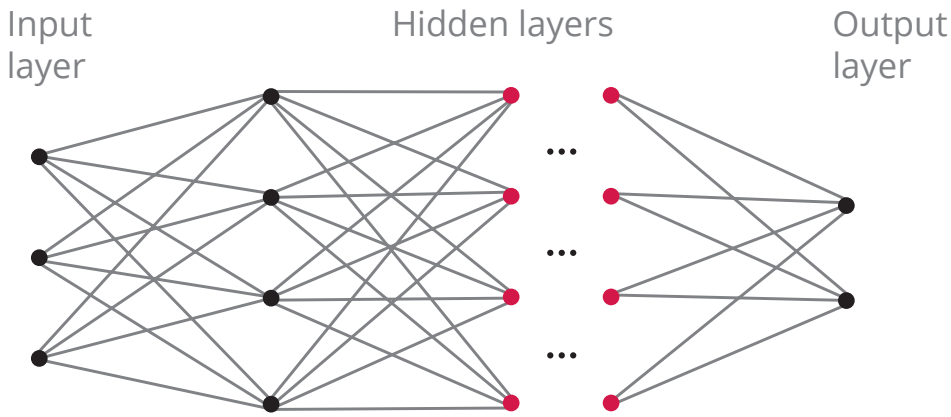
R, Python (scikit-learn).

Multilayer Perceptron

R C

ARTIFICIAL NEURAL NETWORK | ALGORITHM

A class of Feedforward ANNs that consists of at least one hidden layer with the nonlinear activation function. Popular activation functions include rectified linear unit (ReLU), sigmoid function and hyperbolic tangent.



CHARACTERISTICS:

★★★ Accuracy — works well for both linear and nonlinear dependencies

★★ Training Speed — depends heavily on model complexity and on training dataset size

★★★ Prediction Speed — depends on # of model features, scales well

★ Overfitting Resistance — requires big training set and regularization

★★★ Probabilistic Interpretation — thanks to the softmax activation

TIPS:

- Good performance for high dimensional space
- Works well with numerical and categorical data

IMPLEMENTATIONS:

R, Python (scikit-learn), Spark (mllib, classificatory only), Tensorflow.

MACHINE LEARNING

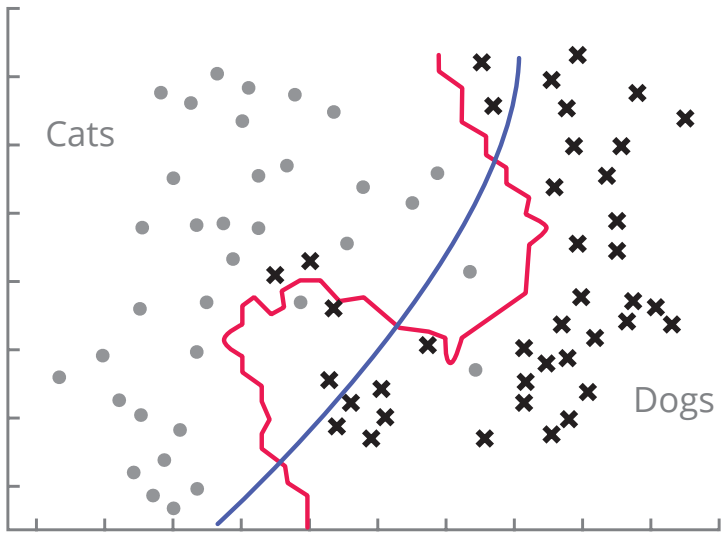
softserve

Classification

C

PROBLEM

Classification is the problem of identifying to which category an object belongs to. Considered an instance of supervised learning and predictive modeling. Classification is distinguished to one-class, binary (two-class) and multiclass tasks.



SAMPLE USE-CASES:

Credit scoring, Spam filtering, Image and speech recognition, OCR, Document classification.

ALGORITHM FAMILIES:

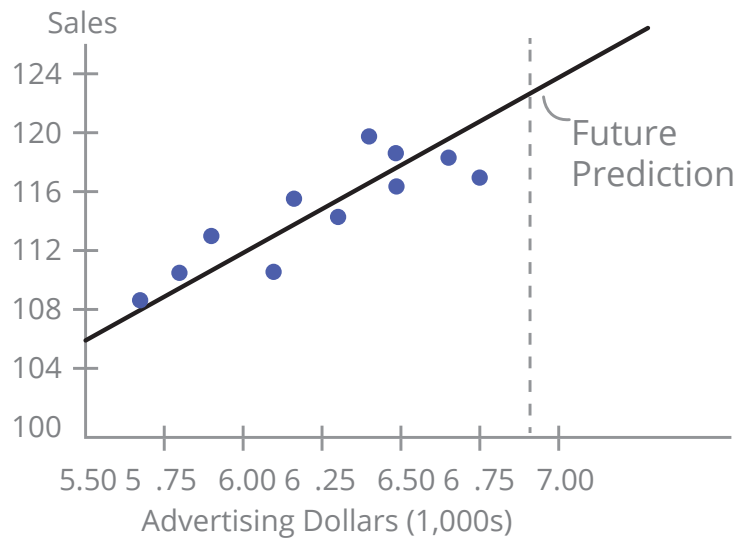
Support Vector Machines, Decision Tree Learning, Instance Based Learning, Generalized Linear Models, Neural Networks.

Regression

PROBLEM

Regression is the problem of predicting continuous-valued attribute associated with an object. Considered an instance of supervised learning and predictive modeling.

Regression is related to classification, but the two are different. Informally, classification predicts whether something will happen, whereas regression predicts how much something will happen.



SAMPLE USE-CASES:

Dynamic pricing, Stock price prediction, Risk estimation, Drug response.

ALGORITHM FAMILIES:

Support Vector Machines, Decision Tree Learning, Instance Based Learning, Generalized Linear Models, Neural Networks.

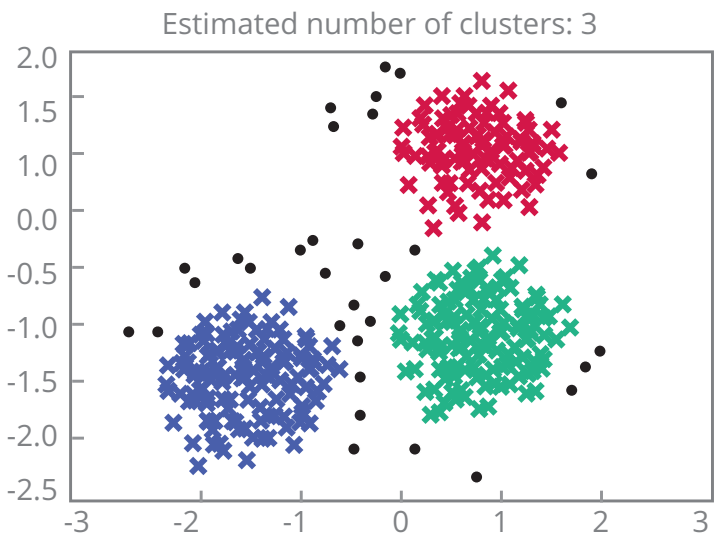
Clustering

CL

PROBLEM

Clustering is the problem of automatic grouping of similar objects into sets. Considered an instance of unsupervised learning and descriptive modeling.

Unlike classification, the output categories (labels) are not known beforehand in clustering.



SAMPLE USE-CASES:

Customer segmentation, Grouping shopping items, Crime hot spots identification, Outlier detection.

ALGORITHM FAMILIES:

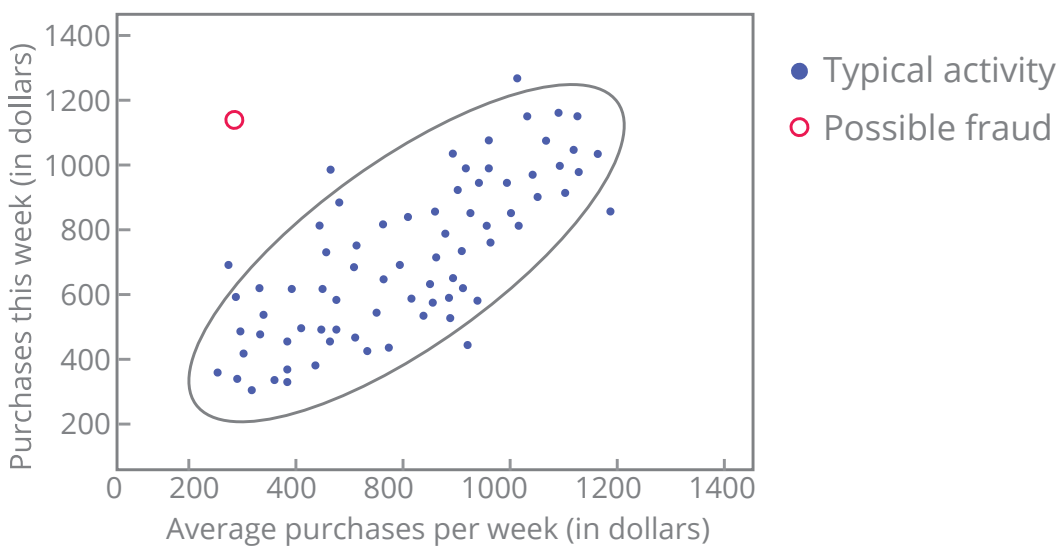
Centroid Based Clustering, Connectivity Based Clustering, Density Based Clustering.

Anomaly Detection

AD

PROBLEM

Anomaly Detection is the problem of identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. It relies on a number of machine learning techniques which include unsupervised, supervised and semi-supervised.



SAMPLE USE-CASES:

Detection of Bank fraud, Security intrusion, System faults, Medical problems, Textual errors.

ALGORITHM FAMILIES:

Support Vector Machines, Decision Trees Learning.