**AdaBoost:**

Boosting in an approach in order to enhance the performance metrics of machine learning algorithms [39]. The AdaBoost algorithm was developed by Freund and Schapire [38] in 1995. Generally a training set (a1,b1)*, …,* (an, bn) is given as input to this algorithm where ai belongs to a domain or instance space A. For bi, it is assumed that each label is under some label B and B is supposed to have values of { -1, +1 } initially. The AdaBoost algorithm runs on weak classifiers or base learning classifiers iteratively for a series of epochs x=1, …, X. A set of weights (which are initially all equal values) over the training set is maintained which is updated in each epoch. After each epoch, all the incorrectly classified instances are given increased weights while the correctly classified instances get a lower value. This leverages the algorithm to pay more attention to the misclassified instances. The main task of this learner is to perceive a weak hypothesis hx: A 🡪 {-1, +1} for the training set. How well a weak learning algorithm performed in AdaBoost is measured by evaluating its error using error margin formulas such as LSE, LAD and the likes.

**Gradient Boosting:**

A greedy approach towards boosting is taken by [40] to optimize the AdaBoost algorithm. The main theory behind the algorithm is similar to that of AdaBoost in the sense that it works in an iterative manner and updates the weight in each round for the instances present in the sample. The basis of this boosting is based on gradient descent optimization of weight and this leads to the birth of gradient boosting. Gradient boosting can be used where the weak learners like regression trees can be implemented for both regression and classification, making it a powerful algorithm which is competitive with its predecessors and highly robust.

**XGBoost:**

The latest approach in boosting is a method that further improves the Gradient boosting algorithm, namely, XGBoost. This originated from the solution of one of the top three teams for optimizing gradient boosted trees. XGBoosting is basically a supervised learning method which implements the optimization of loss functions (similar to gradient boost) and Taylor’s theorem-based approximation.

Is the loss function that is optimized where yi is the label of the instance, ft is the fitting of the tree for the feature xi of the instance, Ω is the regularization parameter for the fitting tree for the feature xi of the instance and t denotes the present time. The difference here is that the loss function cannot be optimized using traditional optimization methods in Euclidean space [41] and needs to utilize Taylor based Theorem to do so.

**Decision Tree:**

Decision Tree [42,43] sorts the instances from the root to the leaf nodes in order to classify them. In the tree, each node defines a test of some features of the instance and each branch from the node relates the possible values of this feature. The tree begins by classifying the root node and moves down the tree and is repeated for each subtree. Which attribute to select for splitting and in which order is determined by a) Information Gain and b) Gini impurity.

**Random Forest:**

A multitude of decision trees are randomly generated which are then bootstrapped and bagged – this is the basic random forest algorithm which is ultimately an ensemble algorithm [44,45,46]. The bootstrap method is mainly used to evaluate statistics on a domain by re-sampling or replacing the instances in the dataset [49]. Bagging or Bootstrap aggregating[47,48] is a technique where the sample data are taken from the original dataset and then from each sample it develops a predictor. Finally it takes the decision from averaging. For the improvement of the estimation for unstable approximations this method is highly effective. Especially in large datasets with high dimensions where it is not possible to perceive a good model within one round.

**Linear Regression:**

Linear Regression is a statistical linear approach used for detecting and modeling the relations between a dependent variable and one or more independent variables [50]. This type of model presumes that the regression function is linear in the inputs[51]. For an input vector = , we want to predict a factual output . The linear regression model has the form [51]. Here the s are unknown coefficients, are variables which can come from different sources and are basic expansions that lead to a polynomial representation. The model is linear in the parameters, the source of does not matter. [51] This linear relationship can be used in order to predict the value of *B* for a given value of *A* using a straight line.

**Logistic Regression:**

The necessity of a logistic regression model appears in order to model the K number of classes’ posterior probabilities using linear functions in . In this method, it is a must to ensure that the probabilities sum to 1 and stay between 0 and 1. [52]

Formally, the model logistic regression model is [53]:

Solving for , this gives

=

The transformed probability is more comprehensible than the untransformed probability. It can be predicted that is 1 when is greater than or equal to 0.5 and is 0 when is less than 0.5 in order to reduce the incorrect classification rate. Hence, considering it as 1 when is non-negative, and considering it as 0 otherwise. Thus we can get a linear classifier by using logistic regression. The two predicted classes are separated by a decision boundary. The decision boundary comes from the solution of . If is one dimensional then it is a point, it is a line when is two dimensional and thus it goes on for higher dimensions of . [53]

**Support Vector Regression (SVR)**

A Support Vector Machine (SVM) is a classifier where it classifies the distinguishable classes by hyperplanes. A hyperplane maximizes the margin between two classes. SVM is one type of supervised learning model where the training dataset is labeled. In output, the model gives an optimized hyperplane to classify the new instances for the testing dataset. The hyperplane becomes a line for two dimensional space. However, this model is robust enough to classify instances in higher dimensional spaces. It can also be implemented for regression tasks. [54]

Support Vector Regression (SVR) is one kind of application form of SVM. It is also a supervised learning model. This model tends to reduce the generalized error rather than reducing the observed errors in training. Thus it tries to obtain optimum generalized performance. The synthesis between a regularization term and training error refers to the generalization error bound. The complexity of this specific hypothesis is controlled by the regularization term. A subset of the training dataset only works in this model. It disregards the other training data which are within the reach of model prediction (within a certain threshold).[55]

# Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is a class of deep, artificial neural network (ANN).[56] It is composed of more than one perceptron. An MLP consists of at least three layers of nodes. They are: i) an input layer- in order to receive signal, ii) an output layer - in order to make a decision on the input, iii) a random number of hidden layers in between the input and output layers. These hidden layers are the main engine of the MLP. Even an MLP with one hidden layer can calculate any continuous function. [60] Without the input nodes, each and every node uses a nonlinear activation function. MLP uses backpropagation to train the model.[57][58] MLP is different from the other linear perceptrons because of its non-linear characteristics and multiple layers. Due to this characteristics, MLP can differentiate data which is not linearly distinguishable .[59] Mathematically this can be written as

where denotes the vector of weights, is the vector of inputs, is the bias and is the activation function.[61]

**References:**

1. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 11155(1):119–139, August 1997.
2. Freund, Y., Robert, S., Abe, N.: A short introduction to boosting. J. Jpn. Soc. Artif. Intell. 14, 771–780 (1999)
3. Friedman, Jerome H. “Greedy Function Approximation: A Gradient Boosting Machine.” The Annals of Statistics, vol. 29, no. 5, 2001, pp. 1189–1232.
4. Chen, Tianqi & Guestrin, Carlos. (2016). XGBoost: A Scalable Tree Boosting System. 785-794. 10.1145/2939672.2939785.
5. Mitchell, T. M. (1997). Machine Learning. New York City: McGraw Hill.
6. Rokach, Lior & Maimon, Oded. (2005). Decision Trees. 10.1007/0-387-25465-X\_9.
7. Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.
8. Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844.
9. Breiman, L. Random forests. Mach. Learn. 2001
10. Biau, Gérard & Scornet, Erwan. (2015). A Random Forest Guided Tour. TEST. 25. 10.1007/s11749-016-0481-7.
11. G. Biau. Analysis of a random forests model. Journal of Machine Learning Research, 13:1063–1095, 2012.
12. <https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/>
13. Introduction to Linear Regression Analysis, 5th Edition by G. Geoffrey Vining, Elizabeth A. Peck, Douglas C. Montgomery Released April 2012- chapter 1
14. The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition

Authors: Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome - chapter 3

1. The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition

Authors: Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome - chapter 4

1. <https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf>
2. http://www.saedsayad.com/support\_vector\_machine.htm
3. <https://pdfs.semanticscholar.org/c5a9/67eaded74a9fc414de4ad5120b0b66acd2c3.pdf>
4. The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition

Authors: Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome -

1. Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961
2. Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "[Learning Internal Representations by Error Propagation](https://apps.dtic.mil/dtic/tr/fulltext/u2/a164453.pdf)". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986.
3. Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function [*Mathematics of Control, Signals, and Systems*](https://en.wikipedia.org/wiki/Mathematics_of_Control,_Signals,_and_Systems), 2(4), 303–314.
4. <https://pathmind.com/wiki/multilayer-perceptron>
5. <http://users.ics.aalto.fi/ahonkela/dippa/node41.html>