



## A Holistic Study of Top Data Mining Algorithms

<sup>1</sup>Karan Diware, <sup>2</sup>Aakash Borhade, <sup>3</sup>Swati Ringe<sup>1,2</sup> Undergrad Researcher (BE), <sup>3</sup> Assistant Professor<sup>1,2,3</sup> Computer Engineering Department, Fr Conceicao Rodrigues College of Engineering, Mumbai University, Mumbai, Maharashtra, India

**Abstract**— Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Companies in a wide range of industries - including retail, finance, health care, manufacturing transportation, and aerospace - are already using data mining tools and techniques to take advantage of historical data. The aim of this paper is to present a holistic study of the different Data Mining algorithms - C 4.5, SVM, k-Means, Apriori, Adaboost, NaivesBayes and Random Forests. For each algorithm, we provide a description of the algorithm, discuss the impact of the algorithm, its advantages, disadvantages and applications.

**Keywords**— Data Mining, Support Vector Machines, Random Forests, Adaboost

### I. INTRODUCTION

The amount of raw data stored in corporate databases is exploding. From the trillions of point-of-sale transactions and credit card purchases to pixel-by-pixel images of galaxies, databases are now measured in gigabytes and terabytes. Just to give an idea, a terabyte is equivalent to around 2.2 million books, which is huge. But, raw data, by itself, does not provide much information. In today's competitively savage world, corporates need to turn these terabytes of raw data into useful information to gain significant insights into their customers and markets to guide their business strategies. Data mining is the computer assisted process of digging through and analyzing enormous sets of data. Data mining tools scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. This helps in predicting behaviors and future trends. Also, it allows businesses to make proactive decisions based on the knowledge.

The two "high-level" primary goals of data mining, in practice, are *prediction* and *description*.

**Prediction** involves using some variables or fields in the database to predict unknown or future values of other variables of interest. In pattern recognition and machine learning applications (such as Natural Language Processing, speech recognition) where prediction is often the primary goal of the KDD process.

**Description** focuses on finding human-interpretable patterns describing the data. However, in the context of KDD, description tends to be more important than prediction.

The goals of prediction and description are achieved by using the following primary **data mining tasks**:

- I. **Classification** is learning a function that maps or classifies a data item into one of several predefined classes.
- I. **Regression** is learning a function which maps a data item to a real-valued prediction variable.
- II. **Change and Deviation Detection** focuses on discovering the most significant changes in the data from previously measured or normative values.
- III. **Dependency Modeling** consists of finding a model which describes significant dependencies between variables. It exists at two levels - structural and quantitative.
- IV. **Clustering** is a common descriptive task where one seeks to identify a finite set of categories or clusters to describe the data.
- VII. **Summarization** involves methods for finding a compact description for a subset of data.

In order to systematically conduct data mining analysis, a general process is usually followed. Some of the standard processes include CRISP and SEMMA.

The *Cross-Industry Standard Process for Data Mining* is the dominant data-mining process framework. It's an open standard. It is most widely used by industry members. The following list describes the various phases of the process:

**Business understanding:** The main tasks in this stage include - Identifying business goals, assessing situation, defining data mining goals, developing a project plan

**Data Understanding** This stage can include initial data collection, data description, data exploration, and the verification of data quality. Models such as cluster analysis can also be applied during this phase, with the intent of identifying patterns in the data.

**Data preparation** The main tasks in this stage include selecting data, cleaning data, constructing data, integrating data and formatting data. Data exploration at a greater depth can be applied during this phase.

**Modeling** Mathematical tools and Data mining software tools are used in this phase to identify patterns within the data. The division of data into training and test sets is also needed for modeling. The tasks include selecting techniques, designing tests, building model and assessing models

**Evaluation** In this stage, the discovered patterns are reviewed and their potential for business use is accessed. The tasks include evaluating results, reviewing the processes, determining the next steps.

**Deployment** This is the final stage where the discoveries are put in everyday business. The tasks include Planning deployment, reporting final results, reviewing final results.

The Data mining algorithms have many applications in business, science, and government, such as targeted marketing, web analysis, disease diagnosis and outcome prediction, weather forecasting, credit risk and loan approval, customer relationship modeling, fraud detection, and terrorism threat detection. Data mining is based on methods several fields, but mainly machine learning, statistics, databases, and information visualization. But, in this paper, our main focus will be on some of the most popular algorithms -

- 1) **C4.5** : C4.5 is an algorithm that was developed by Ross Quinlan. This algorithm generates Decision trees which can further be used for problems related to classification.
- 2) **SVM** : A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane.
- 3) **K-Means** : K-means (MacQueen, 1967) is a vector quantization method and one of the simplest unsupervised learning algorithms that solve the well known clustering problem.
- 4) **Apriori Algorithm** : The Apriori Algorithm is a classic and influential algorithm for mining frequent itemsets for boolean association rules.
- 5) **Adaboost Algorithm** : The AdaBoost algorithm, Adaptive Boosting, introduced in 1995 by Freund and Schapire is machine learning meta algorithm for constructing a strong classifier as a linear combination of 'simple' 'weak' classifiers.
- 6) **Naives Bayes** : Naive Bayes classifiers are among the most successful known algorithms for learning to classify text documents and they are based on the Baye's theorem with independent assumption between predictors.
- 7) **Random Forests** : A Random Forest consists of a collection or ensemble of simple tree predictors, each capable of producing a response when presented with a set of predictor values. The Random Forest algorithm was developed by Breiman.

## II. DATA MINING ALGORITHMS

**C 4.5** : C4.5 is an algorithm that was developed by Ross Quinlan. This algorithm generates Decision trees which can further be used for problems related to classification. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason C4.5 is often referred to as a statistical classifier. Similar to ID3, C4.5 uses information gain when generating the decision tree as splitting criteria. C4.5 algorithm further make changes into the previous ID3 algorithm and deals with both discrete and continuous attributes.

C4.5 constructs a classifier in the form of a decision tree. In order to do this, C4.5 is given a set of data representing things that are already classified. It can accept data with categorical or numerical values. To handle continuous values it generates threshold and then divides attributes with values above the threshold and values equal to or below the threshold. C4.5 algorithm can easily handle missing values.

Later on, it was extended by C5.0. C5.0 algorithm is an extension of C4.5 algorithm which is also extension of ID3. It is the classification algorithm which applies in big data set. It is better than C4.5 on the speed, memory and the efficiency. C5.0 model works by splitting the sample based on the field that provides the maximum information gain. The C5.0 model can split samples on basis of the biggest information gain field. C5.0 is easily handles the multi value attribute and missing attribute from data set.

Advantages:

- A. Handling both continuous and discrete attributes
- B. Handling training data with missing attribute values
- C. Pruning trees after creation
- D. Handles noisy data
- E. Quite fast, quite popular and the output is human readable.
- F. The best selling point of decision trees is their ease of interpretation and explanation.

Disadvantages:

1. Small change in data set can cause different decision trees. This happens usually when the data overfits
2. Dataset needs to be big in order for the algorithm to work efficiently

Applications:

- I. Businesses -(for visualization of probabilistic models)
- II. E-commerce-content-based image retrieval.
- III. Medicine - diagnostics of various diseases.
- IV. Industry - Production quality control, non-destructive tests

A popular open-source Java implementation can be found over at OpenTox. Orange, an open-source data visualization and analysis tool for data mining, implements C4.5 in their decision tree classifier.

### Support Vector Machines (SVM):

**Support Vector Machines** are a group of supervised learning methods that can be applied to classification or regression. A Support Vector Machine is actually a discriminative classifier which is formally defined by a separating hyperplane, i.e., when it is given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples.

Support Vector Machines is based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. Support Vector Machine is primarily a classifier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables

Advantages:

- A. Effective in high dimensional spaces.
- B. Still effective in cases where number of dimensions is greater than the number of samples.
- C. Uses a subset of training points in the decision function
- D. (called support vectors), so it is also memory efficient.
- E. Versatile: different *Kernel functions* can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels
- F. Produce very accurate classifiers.

The key features of SVMs are the use of kernels, the absence of local minima, the sparseness of the solution and the capacity control obtained by optimising the margin - Shawe-Taylor and Cristianini (2004).

Disadvantages:

- I. If the number of features is much greater than the number of samples, the method is likely to give poor performances.
- II. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation
- III. A second limitation is speed and size, both in training and testing.
- IV. Although SVMs have good generalization performance, they can be abysmally slow in test phase, a problem addressed in (Burges, 1996; Osuna and Girosi, 1998)
- V. However, from a practical point of view perhaps the most serious problem with SVMs is the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks
- VI. SVM is a binary classifier. To do a multi-class classification, pair-wise classifications can be used (one class against all others, for all classes).

Applications:

- I. Text (and hypertext) categorization : The classification of natural text (or hypertext) documents into a fixed number of predefined categories based on their content like email filtering, web searching, sorting document by topic, etc.
- II. Image classification
- III. It is widely used in bioinformatics for Protein classification, Cancer classification etc.
- IV. Hand-written character recognition

There are many implementations of SVM. A few of the popular ones are scikit-learn, MATLAB and of course libsvm.

### K-means clustering:

K-means is one of the most popular partition clustering method. It is also one of the simplest unsupervised learning algorithms. K-means clustering algorithm was developed by J. MacQueen (1967) and then by J. A. Hartigan and M. A. Wong around 1975

The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume  $k$  clusters) fixed a priori. The main idea is to define  $k$  centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate  $k$  new centroids as barycenters of the clusters resulting from the previous step. After we have these  $k$  new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the  $k$  centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$$

where  $\|x_i^{(j)} - c_j\|^2$  is a chosen distance measure between a data point  $x_i^{(j)}$  and the cluster centre  $c_j$ , is an indicator of the distance of the  $n$  data points from their respective cluster centres.

K-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to work with fuzzy feature vectors.

More advanced algorithms related to k-means are Expectation Maximization algorithm especially Gaussian Mixture, Self-Organizing Maps(SOM) from Kohonen, Learning Vector Quantization(LVQ). Also, to overcome the weakness of k-means, several algorithms had been proposed such as k- methods, fuzzy-c and k-mode.

K-Means clustering generates a specific number of disjoint, flat (non-hierarchical) clusters. It is well suited to generating globular clusters. The K-Means method is numerical, unsupervised, non-deterministic and iterative.

Advantages:

- A. If variables are huge, then K-Means most of the times computationally faster than hierarchical clustering, if we keep k small.
- B. It produce tighter clusters than hierarchical clustering, especially if the clusters are globular.
- C. Fast, robust and easier to understand.
- D. Relatively efficient:  $O(knd)$ ,  
where n is # objects, k is # clusters, d is # dimension of each object, and t is # iteration  
Normally,  $k, t, d \ll n$ .
- E. Gives best result when data set are distinct or well separated from each other.

Disadvantages:

1. Does not work well with clusters (in the original data) of Different size and Different density
2. Algorithm fails for non-linear data set
3. Fixed number of clusters can make it difficult to predict what K should be.
4. Does not work well with non-globular clusters.
5. Different initial partitions can result in different final clusters. It is helpful to rerun the program using the same as well as different K values, to compare the results achieved.
6. Unable to handle noisy data and outliers.
7. Euclidean distance measures can unequally weight underlying factors.
8. The learning algorithm provides the local optima of the squared error function.
9. Applicable only when mean is defined i.e. fails for categorical data.

Applications:

- I. Clustering algorithm can be used in identifying the cancerous data set. It has been found through experiment that cancerous data set gives best results with unsupervised non linear clustering algorithms and hence we can conclude the non linear nature of the cancerous data set.  
\* A Comparison of Fuzzy and Non-Fuzzy clustering Techniques in Cancer Diagnosis ' by X.Y. Wang and J.M. Garibaldi.
- II. Clustering algorithm is the backbone behind the search engines  
\* Clustering Billions of Images with Large Scale Nearest Neighbor Search ' by Ting Liu, Charles Rosenberg and H.A. Rowley.
- III. Clustering algorithm can be used to monitor the students' academic performance by O.J. Oyelade, O.O. Oladipupo and I.C. Obagbuwa
- IV. Clustering Algorithm can be used effectively in Wireless Sensor Network's based application. One application where it can be used is in Landmine detection. Clustering algorithm plays the role of finding the Cluster heads (or cluster center) which collects all the data in its respective cluster.  
\* Wireless Sensor Network based Adaptive Landmine Detection Algorithm by Abhishek Saurabh and Azad Naik.

K-means can be mostly classified as unsupervised but it can also be semi-supervised. K-means can be used to pre-cluster a massive dataset followed by a more expensive cluster analysis on the sub-clusters. A ton of implementations for k-means clustering are available online - Apache Mahout, Julia, R, SciPy, Weka, MATLAB, SAS .

### **Apriori Algorithm:**

In data mining, **Apriori** is a classic algorithm for learning association rules. The Apriori algorithm was proposed by Agarwal and Srikant in 1994. Mining for associations among items in a large database of sales transaction is an important database mining function. Apriori is designed to operate on databases containing transactions (for example, collections of products bought by clients). The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

It is a two-pass algorithm which limits the need for main memory. One of the Key Idea behind Apriori is Monotonicity: If a set of items I appear at least s times, so does every subset J of I. Also an important property of Apriori algorithm is that all nonempty subsets of a frequent itemset must also be frequent

Apriori uses a bottom up approach, where frequent subsets are extended one item at a time and groups of candidates are tested against the data and terminates when no further extension is possible. Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently.

Advantages:

- A. Easy to implement

- B. Easily parallelized
- C. Use large itemset property

Disadvantages:

1. Apriori algo can be slow and bottleneck is candidate generation. For candidate generation process it takes more memory, space and time
2. Assume transaction database is memory resident
3. Requires many database scans
4. It does multiple scan over the database to generate candidate set.
5. The number of database passes are equal to the max length of frequent item set.

Applications:

- I. Web usage mining
- II. Intrusion detection and bioinformatics.
- III. Market Basket Analysis

Apriori is well understood, easy to implement and has many derivatives. Therefore, plenty of implementations of Apriori are available. Some popular ones are the ARtool, Weka, and Orange.

**Adaboost Algorithm:**

The AdaBoost algorithm, Adaptive Boosting, introduced in 1995 by Freund and Schapire is machine learning meta algorithm for constructing a strong classifier as a linear combination of 'simple' 'weak' classifiers, which means if the model failed at some point, we had a sense of which weak learner was to blame. This allowed us to manually tune the models to improve performance.

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

*AdaBoost (Adaptive Boosting) is a powerful classifier that works well on both basic and more complex recognition problems.* AdaBoost is a type of Ensemble Learning where multiple learners are employed to build a stronger learning algorithm. AdaBoost works by choosing a base algorithm (for example, decision trees) and iteratively improving it by accounting for the incorrectly classified examples in the training set.

In Adaboost algorithm, equal weights are assigned to all the training examples and a base algorithm is chosen. At each step of iteration, the base algorithm is applied to the training set and the weights of the incorrectly classified examples are increased. The iteration is done n times, each time applying base learner on the training set with updated weights. The final model is the weighted sum of the n learners.

AdaBoost is extremely successful machine learning meta algorithm formulated by Schapire and Freund which won them the Godel Prize in 2003 for their construction of AdaBoost algorithm.

Advantages:

- A. AdaBoost is a powerful classification algorithm that works well on simple and complex recognition problems.
- B. It has application in wide variety of fields such as biology, computer vision, and speech processing.
- C. It is Fast than the traditional algorithms
- D. Simple and easy to implement
- E. No parameters to tune
- F. Flexible
- E. Versatile – can be used with data that is textual, numeric, discrete, etc. Also has been extended to learning problems well beyond binary classification
- F. Can be used with many different classifiers
- G. Not prone to overfitting
- H. Improves classification accuracy

Disadvantages:

1. AdaBoost can be sensitive to noisy data and outliers. In some problems, however, it can be less susceptible to the overfitting problem than most learning algorithms.
2. Leads to suboptimal solution

Applications:

- I. Game Theory
- II. Boosting for Text Categorization
- III. Human-computer Spoken Dialogue
- IV. Face detection

AdaBoost also has a ton of implementations and variants. A few of them include – scikit-learn, ICSIBOOST, gbm: Generalized Boosted Regression Models

### **Naive Bayes:**

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors.

The Bayesian Classifier is capable of calculating the most probable output depending on the input. It is possible to add new raw data at runtime and have a better probabilistic classifier. A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable.

In the Bayesian (or epistemological) interpretation, probability measures a *degree of belief*. Bayes' theorem then links the degree of belief in a proposition before and after accounting for evidence.

A Bayesian classifier is based on the idea that the role of a (natural) class is to predict the values of features for members of that class.

The idea behind a Bayesian classifier is that, if an agent knows the class, it can predict the values of the other features. If it does not know the class, Bayes' rule can be used to predict the class given (some of) the feature values. In a Bayesian classifier, the learning agent builds a probabilistic model of the features and uses that model to predict the classification of a new example. The simplest case is the naive Bayesian classifier, which makes the independence assumption that the input features are conditionally independent of each other given the classification. In general, the naive Bayesian classifier works well when the independence assumption is appropriate, that is, when the class is a good predictor of the other features and the other features are independent given the class. This may be appropriate for **natural kinds**, where the classes have evolved because they are useful in distinguishing the objects that humans want to distinguish. Natural kinds are often associated with nouns, such as the class of dogs or the class of chairs.

Advantages:

- A. Fast to train (single scan). Fast to classify
- B. Not sensitive to irrelevant features
- C. Handles real and discrete data
- D. Handles streaming data well
- E. It scales linearly with the number of predictors and rows.
- F. The build process for Naive Bayes is parallelized.
- G. The Naive Bayes algorithm affords fast, highly scalable model building and scoring
- H. Naive Bayes can be used for both binary and multiclass classification problems.

Disadvantages:

1. Assumes independence of features
2. Practically, dependence exists among variables
3. Dependencies among these can't be modelled
4. Requires large amount of data for better accuracy
5. Zero problem - If there is no occurrence of the class label and attribute value, then this algorithm will consider the probability of that attribute to be zero.

Applications:

- I. Text classification
- II. Spam filtering
- III. Intrusion detection system
- IV. Hybrid recommender system
- V. Online application - simple emoticon modelling
- VI. Credit Card Fraud detection

Naive Bayes algorithm is based on supervised learning and involves simple arithmetic. It's just tallying up counts, multiplying and dividing. Despite its simplicity, Naive Bayes can be surprisingly accurate. Implementations of Naive Bayes can be found in Orange, scikit-learn, Weka and R.

### **Random Forest:**

The Random Forests algorithm was developed by Leo Breiman and Adele Cutler. The Random Forests algorithm is one of the best among classification algorithms - able to classify large amounts of data with accuracy. Random Forests are an ensemble learning method (also thought of as a form of nearest neighbor predictor) for classification and regression that construct a number of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. For classification problems, this response takes the form of a class membership, which associates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable. Alternatively, for regression problems, the tree response is an estimate of the dependent variable given the predictors.

Random Forests are a combination of tree predictors where each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest. The basic principle is that a group of "weak learners" can come together to form a "strong learner". Random Forests are a wonderful tool for making predictions considering they do not overfit because of the law of large numbers. Introducing the right kind of randomness makes them accurate classifiers and regressors. Single decision trees often have high variance or high bias. Random Forests

attempts to mitigate the problems of high variance and high bias by averaging to find a natural balance between the two extremes. Random Forests are easy to learn and use for both professionals and lay people - with little research and programming required and may be used by folks without a strong statistical background. We can make more accurate predictions without most basic mistakes common to other methods.

The predictions of the Random Forest are taken to be the average of the predictions of the trees:

$$\text{Random Forest Prediction } s = \frac{1}{K} \sum_{k=1}^K K^{\text{th}} \text{ tree response}$$

where the index  $k$  runs over the individual trees in the forest.

Features of Random Forests :

- I. Runs efficiently on large data bases
- II. Most accurate among current algorithms
- III. Handles thousands of input variables without variable deletion
- IV. Gives estimates of what variables are important in the classification
- V. Generates an internal unbiased estimate of the generalization error as the forest building progresses
- VI. Provides effective methods for estimating missing data
- VII. Maintains accuracy when a large proportion of the data are missing
- VIII. Provides methods for balancing error in class population unbalanced data sets
- IX. Generated forests can be saved for future use on other data
- X. Prototypes are computed that give information about the relation between the variables and the classification.
- XI. Computes proximities between pairs of cases that can be used in clustering, locating outliers, or (by scaling) give interesting views of the data
- XII. Capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection
- XIII. Offers an experimental method for detecting variable interactions

Disadvantages:

- I. A large number of trees may make the algorithm slow for real-time prediction.
- II. For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

Additional features:

- Random forests does not overfit. We can run as many trees as you want and it is very fast.
- Random Forests is of the most powerful and successful machine learning techniques
- Both *R* and *Python* have robust packages to implement Random Forests.
- *Fast Random Forest* is an efficient implementation of the Random Forests classifier for the Weka environment
- Random Forests is one of the top 2 methods used by Kaggle competition winners

### III. CONCLUSION

Frankly, there is no single learning algorithm that can perform better than any other when the expected generalization accuracy is the performance measure. It assumes that all possible targets are equally likely. However, this assumption is clearly wrong when in practice because for a given domain, it is often found that not all concept are equally probable. It has been found that the algorithm which performs the best depends on the type of problem that is being considered, the performance matrix used and the dataset characteristics.

No classifier will be the best in all cases due to the 'No Free Lunch Theorem'. So, the list of top 10 algorithms depends on whether the data mining task is supervised or unsupervised (classification or clustering). For supervised machine learning, the top algorithms would be SVM, kNN, naive bayes, regression, HMM, Decision trees. For clustering the algorithms would be k-means, heirarchical, MCFS, etc. The algorithms like adaboost are a combination of multiple weak classifiers to improve performance. Page-ranking, mRmR ranks the feature space or the search space. Essentially the algorithms are good if they fit the task at hand. One potential list of top Data Mining algorithms comes from the Analytics 1305 [2] documentation:

- Kernel Density Estimation and Non-parametric Bayes Classifier
- K-Means
- Kernel Principal Components Analysis
- Linear Regression
- Neighbors (Nearest, Farthest, Range, k, Classification)
- Non-Negative Matrix Factorization
- Support Vector Machines
- Dimensionality Reduction
- Fast Singular Value Decomposition
- Decision Tree
- Bootstapped SVM

The 11 algorithms implemented by 11Ants [1] provide another potential list of top Data Mining algorithms :

- Decision Tree
- Gaussian Processes
- Logistic Regression
- Logit Boost
- Model Tree
- Naïve Bayes
- Nearest Neighbors
- PLS
- Random Forest
- Ridge Regression
- Support Vector Machine

Some of the top algorithms in Oracle Data Mining [3] list includes :

- Classification: logistic regression, naïve bayes, SVM, decision tree
- Regression: multiple regression, SVM
- Attribute importance: MDL
- Anomaly detection: one-class SVM
- Clustering: k-means, orthogonal partitioning
- Association: A Priori
- Feature extraction: NMF

As we can see, there is no absolute list of top 10 algorithms. So, it certainly depends on the task in hand to list the top algorithms. All algorithms are used effectively in various domains to overcome real world difficulties making data mining a boon for people living in an era of technology.

## REFERENCES

- [1] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.686.840&rep=rep1&type=pdf>
- [2] <https://datajobs.com/data-science-repo/Naive-Bayes-%5BKevin-Murphy%5D.pdf>
- [3] [https://www.researchgate.net/profile/Irina\\_Rish/publication/228845263\\_An\\_Empirical\\_Study\\_of\\_the\\_naive\\_Bayes\\_Classifier/links/00b7d52dc3ccd8d692000000.pdf](https://www.researchgate.net/profile/Irina_Rish/publication/228845263_An_Empirical_Study_of_the_naive_Bayes_Classifier/links/00b7d52dc3ccd8d692000000.pdf)
- [4] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.8487&rep=rep1&type=pdf>
- [5] [http://saiconference.com/Downloads/SpecialIssueNo10/Paper\\_3A\\_comparative\\_study\\_of\\_decision\\_tree\\_ID3\\_and\\_C4.5.pdf](http://saiconference.com/Downloads/SpecialIssueNo10/Paper_3A_comparative_study_of_decision_tree_ID3_and_C4.5.pdf)
- [6] [www.cse.unr.edu/~looney/cs773b/SVMmicrocalcification.pdf](http://www.cse.unr.edu/~looney/cs773b/SVMmicrocalcification.pdf)
- [7] <http://research.microsoft.com/en-us/um/people/cburgess/papers/svmtutorial.pdf>
- [8] [www.cs.cornell.edu/people/tj/publications/joachims\\_98a.pdf](http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf)
- [9] [https://air.unipr.it/retrieve/handle/11381/2337353/7766/2011%20Vincenzi%20et%20al.%20Ecological%20Modelling\\_Application%20of%20a%20Random%20Forest%20algorithm.pdf](https://air.unipr.it/retrieve/handle/11381/2337353/7766/2011%20Vincenzi%20et%20al.%20Ecological%20Modelling_Application%20of%20a%20Random%20Forest%20algorithm.pdf)
- [10] [ftp://131.252.97.79/Transfer/Treg/WFRE\\_Articles/Liaw\\_02\\_Classification%20and%20regression%20by%20randomForest.pdf](ftp://131.252.97.79/Transfer/Treg/WFRE_Articles/Liaw_02_Classification%20and%20regression%20by%20randomForest.pdf)
- [11] <http://www.codeproject.com/Articles/70371/Apriori-Algorithm>
- [12] [https://www.it.uu.se/edu/course/homepage/infoutv/ht08/vldb94\\_rj.pdf](https://www.it.uu.se/edu/course/homepage/infoutv/ht08/vldb94_rj.pdf)
- [13] [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5687939&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D5687939](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5687939&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D5687939)
- [14] [www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf](http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf)