**MACHINE LEARNING & HEALTH CARE ANALYTICS**

**Dr. M. Kamakshaiah**

**kamakshaiah.m@gmail.com**

**MLtRS**

MltRS, stands for Machine Learning through R Shiny, which is open source server level web applicated developed using Shiny platform through R. MLtRS is useful for academics and industry practitioners who are interested in various algorithms of Machine Learning (ML).

Shiny applications are easy to use and doesn’t require to have knowledge of R programming. The applications implements algorithms through very simple approach viz. (1) Upload, (2) Perform and (3) Report. This approach is very useful for beginners in the field of ML and AI.

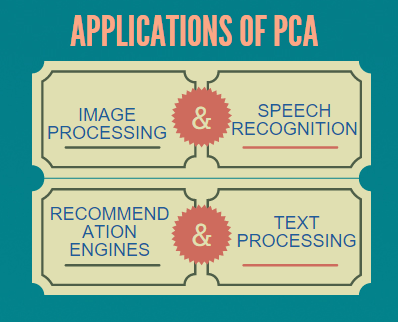
All methodologies or algorithms explained in this document are practiced using MltRS.

Please visit <https://github.com/Kamakshaiah> to know more about MLtRS. This application was tested using various datasets. Datasets were related to different medical specializations such as cardialogy, oncology, diabetes etc. Download datasets from <https://github.com/Kamakshaiah/datasets>.

1. **Principal Component Analysis**

Clinical studies utilizing electronic healthcare records (EHR) usually present a large number of variables. These variables frequently correlate with each other, which will introduce multicollinearity in the regression models. Although the problem of collinearity will not compromise the predictive ability of a regression model, it can interfere in determining the precise effect of each predictor. Additionally, the standard errors of the estimations affected by multicollinearity tend to be large, making the inference over such estimations less precise (wider confidence intervals and bigger P values).

The problem of multicollinearity in clinical studies is ubiquitous, and there are many statistical methods being developed to handle it. One of the most used methods is the principal component analysis (PCA). This statistical approach reduces a set of intercorrelated variables into a few dimensions that gather a big amount of the variability of the original variables. These dimensions are called components and have the properties of collecting highly correlated variables within each component and being uncorrelated with each other.



**References:**

Breast Cancer : <https://towardsdatascience.com/principal-component-analysis-pca-101-using-r-361f4c53a9ff>

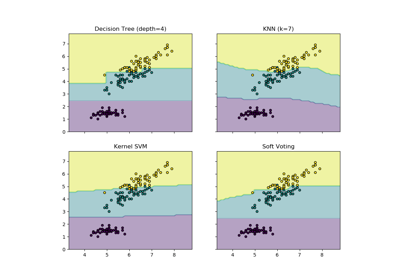
**Practice**: Brest cancer data set (<https://github.com/datasets/breast-cancer/blob/master/data/breast-cancer.csv>)

1. **K Nearest Neighborhood**

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

* In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.



Both for classification and regression, a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

**References**

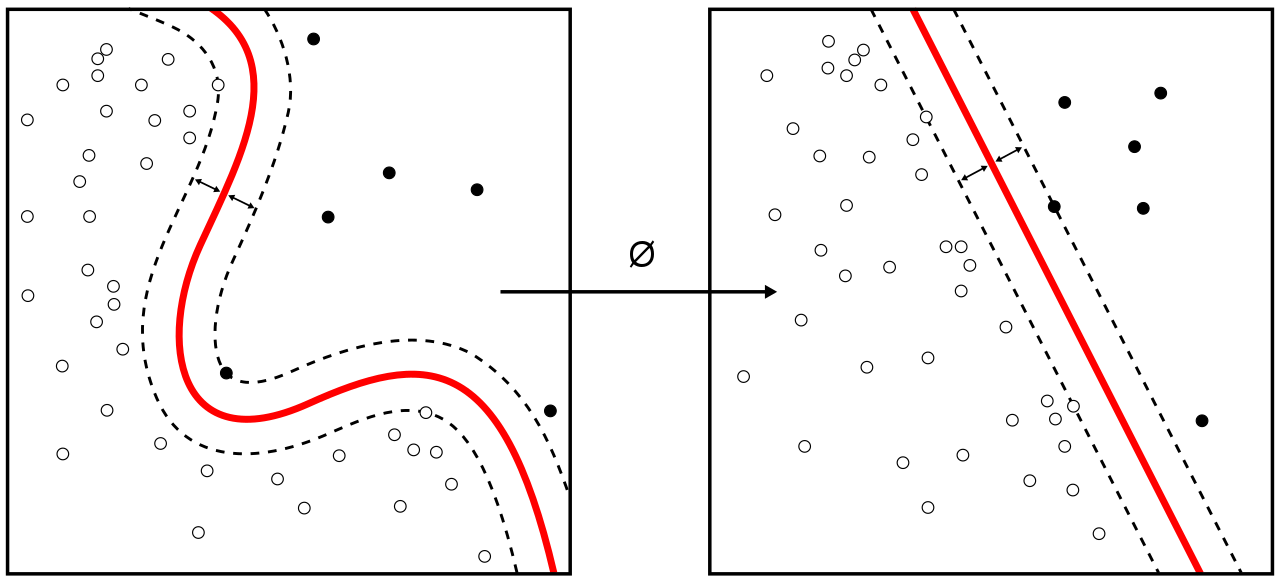
Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm: <https://arxiv.org/pdf/1508.02061.pdf>

**Practice**: Prostrate Cancer Data sets (<https://github.com/datasets/breast-cancer/blob/master/data/breast-cancer.csv>)

**Practice**: Breast Cancer data sets

**3. SVM**

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.



In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data is unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support-vector clustering algorithm, created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

**Reference:**

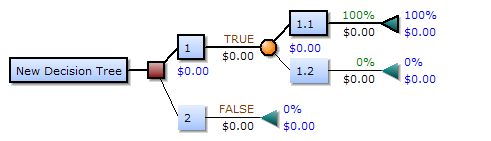
# Application of Support Vector Machine for Prediction of Medication Adherence in Heart Failure Patients

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3092139/>

**Practice**: Heart Failure Dataset

**4.** **Decision tree**

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 are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

**Reference:**

Decision Trees: An Overview and Their Use in Medicine

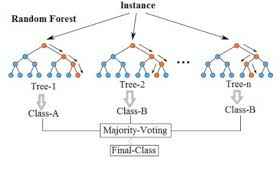
<https://link.springer.com/article/10.1023/A:1016409317640>

**Practice**: Airquality dataset

**5.** **Random forest**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing 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. Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.



**Reference:**

# Random forest prediction of Alzheimer’s disease using pairwise selection from time series data

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0211558>

# Random Forest Algorithm for the Classification of Neuroimaging Data in Alzheimer's Disease: A Systematic Review

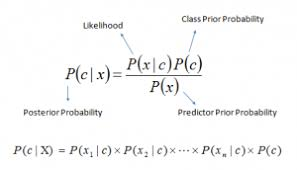
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5635046/>

**Practice:** Heart Failure dataset

**6. Naive Bayes Classifier**

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes has been studied extensively since the 1960s. It was introduced (though not under that name) into the text retrieval community in the early 1960s, and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.



Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

**Reference**

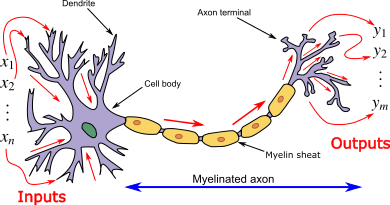
# Applying Naive Bayesian Networks to Disease Prediction: a Systematic Review

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5203736/>

Practice: Sepsis dataset

**7. Neural Networks**

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.



An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

**Reference:**

# Efficient Cancer Detection Using Multiple Neural Networks

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5722487/>

# Breast Cancer Detection by Means of Artificial Neural Networks

<https://www.intechopen.com/books/advanced-applications-for-artificial-neural-networks/breast-cancer-detection-by-means-of-artificial-neural-networks>

**Practice**: Breast Cancer dataset