.

1. Supervised Learning

How it works: This algorithm consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

2. Unsupervised Learning

How it works: In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means.

3. Reinforcement Learning:

How it works: Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process

List of Common Machine Learning Algorithms

Here is the list of commonly used machine learning algorithms. These algorithms can be applied to almost any data problem:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. SVM
5. Naive Bayes
6. KNN
7. K-Means
8. Random Forest
9. Dimensionality Reduction Algorithms
10. Gradient Boosting algorithms
    1. GBM
    2. XGBoost
    3. LightGBM
    4. CatBoost

.

.

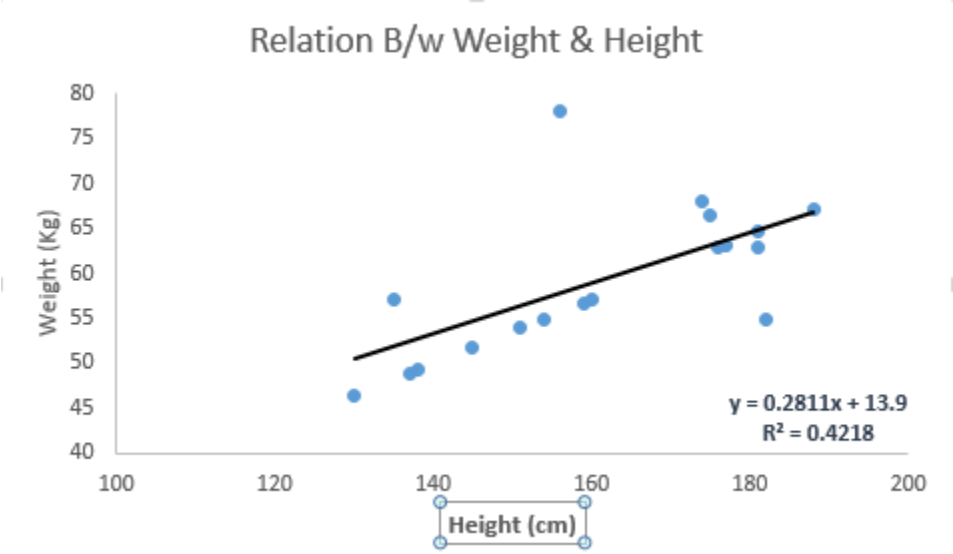
1. Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a *X + b.

The best way to understand linear regression is to relive this experience of childhood. Let us say, you ask a child in fifth grade to arrange people in his class by increasing order of weight, without asking them their weights! What do you think the child will do? He / she would likely look (visually analyze) at the height and build of people and arrange them using a combination of these visible parameters. This is linear regression in real life! The child has actually figured out that height and build would be correlated to the weight by a relationship, which looks like the equation above.

In this equation:

- Y – Dependent Variable
- a – Slope
- X – Independent variable
- b – Intercept



These coefficients a and b are derived based on minimizing the sum of squared difference of distance between data points and regression line.

Look at the below example. Here we have identified the best fit line having linear equation **y=0.2811x+13.9**. Now using this equation, we can find the weight, knowing the height of a person.

.

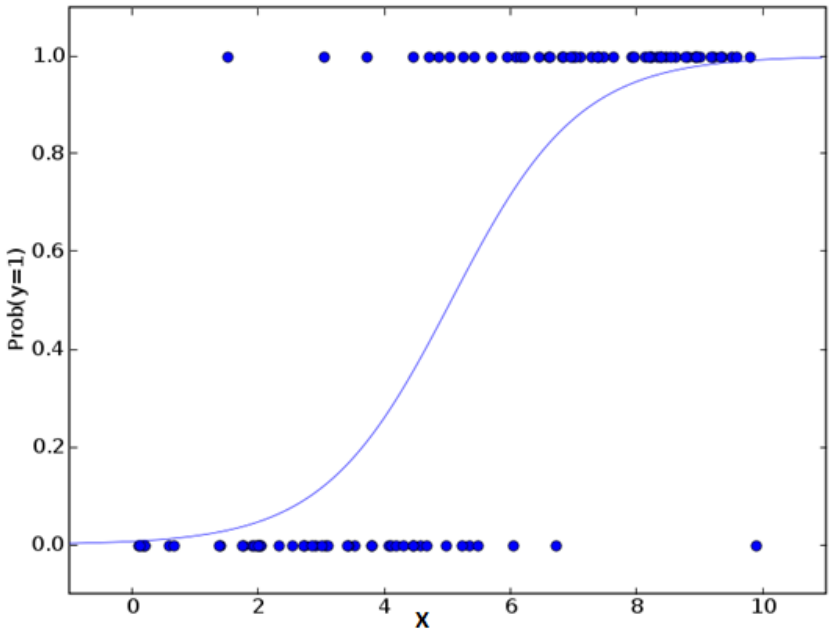
.

Linear Regression is of mainly two types: Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is characterized by one independent variable. And, Multiple Linear Regression(as the name suggests) is characterized by multiple (more than 1) independent variables. While finding best fit line, you can fit a polynomial or curvilinear regression. And these are known as polynomial or curvilinear regression.

2. Logistic Regression

Don't get confused by its name! It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as **logit regression**. Since, it predicts the probability, its output values lies between 0 and 1 (as expected).

Again, let us try and understand this through a simple example.



Let's say your friend gives you a puzzle to solve. There are only 2 outcome scenarios – either you solve it or you don't. Now imagine, that you are being given wide range of puzzles / quizzes in an attempt to understand which subjects you are good at. The outcome to this study would be something like this – if you are given a trignometry based tenth grade problem, you are 70% likely to solve it. On the other hand, if it is grade fifth history question, the probability of getting an answer is only 30%. This is what Logistic Regression provides you.

Coming to the math, the log odds of the outcome is modeled as a linear combination of the predictor variables.

.

.

Above, p is the probability of presence of the characteristic of interest. It chooses parameters that maximize the likelihood of observing the sample values rather than that minimize the sum of squared errors (like in ordinary regression).

Now, you may ask, why take a log? For the sake of simplicity, let's just say that this is one of the best mathematical way to replicate a step function. I can go in more details, but that will beat the purpose of this article.

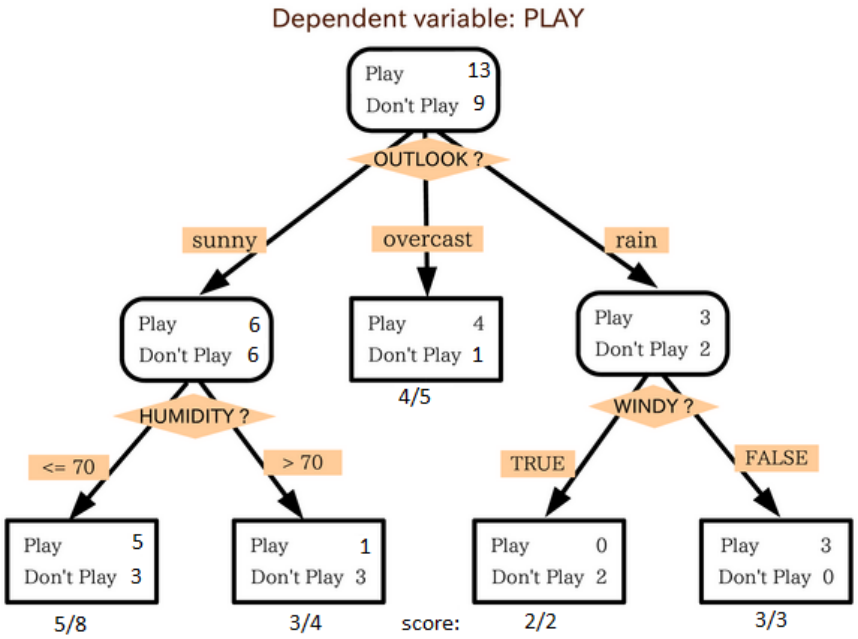
Furthermore..

There are many different steps that could be tried in order to improve the model:

- including interaction terms
- removing features
- regularization techniques
- using a non-linear model

3. Decision Tree

This is one of my favorite algorithm and I use it quite frequently. It is a type of supervised learning algorithm that is mostly used for classification problems. Surprisingly, it works for both categorical and continuous dependent variables. In this algorithm, we split the population into two or more homogeneous sets. This is done based on most significant attributes/ independent variables to make as distinct groups as possible. For more details, you can read: Decision Tree Simplified.



.

.

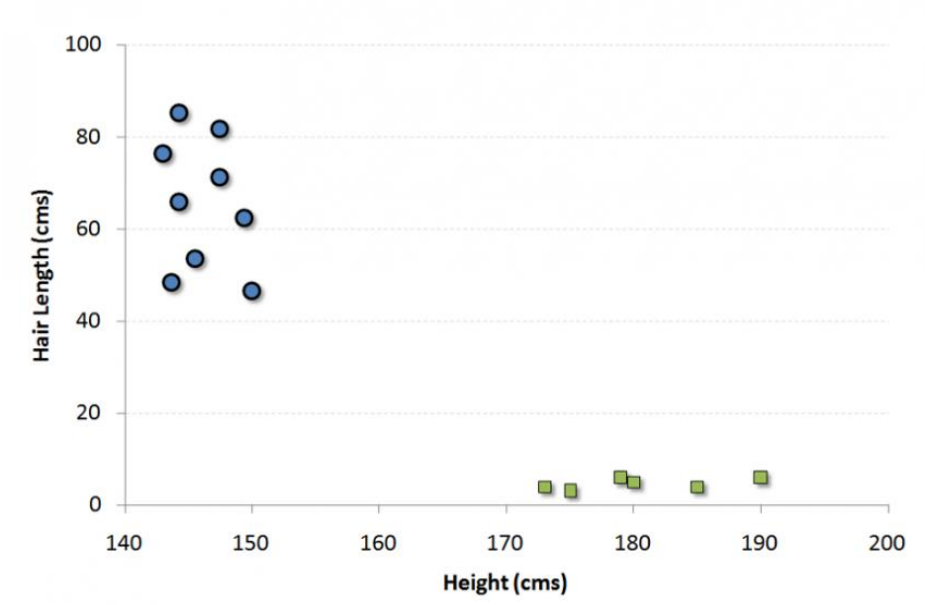
4. SVM (Support Vector Machine)

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we'd first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as **Support Vectors**)

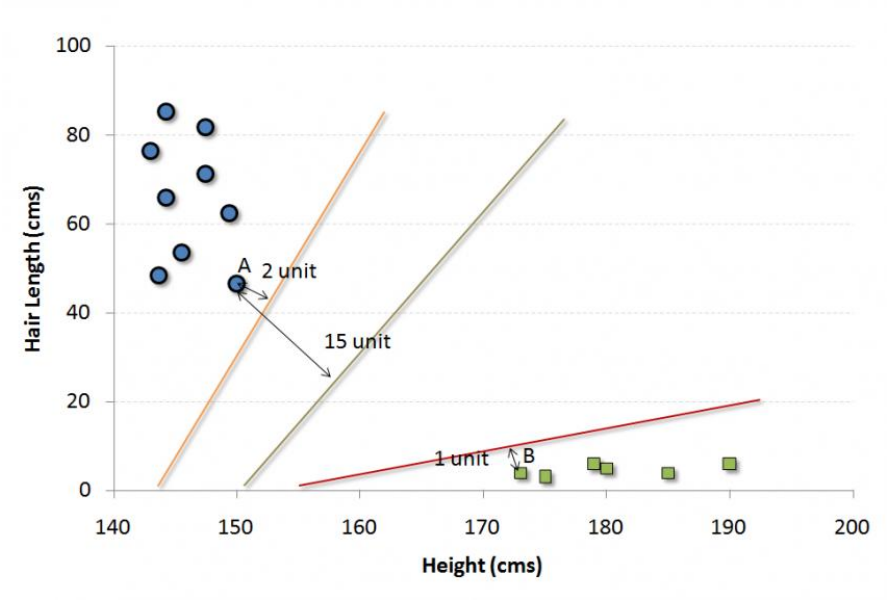
In the example shown above, the line which splits the data into two differently classified groups is the *black* line, since the two closest points are the farthest apart from the line. This line is our classifier. Then, depending on where the testing data lands on either side of the line, that's what class we can classify the new data as.

More: [Simplified Version of Support Vector Machine]



.

.



Think of this algorithm as playing JezzBall in n-dimensional space. The tweaks in the game are:

- You can draw lines / planes at any angles (rather than just horizontal or vertical as in classic game)
- The objective of the game is to segregate balls of different colors in different rooms.
- And the balls are not moving.

5. Naive Bayes

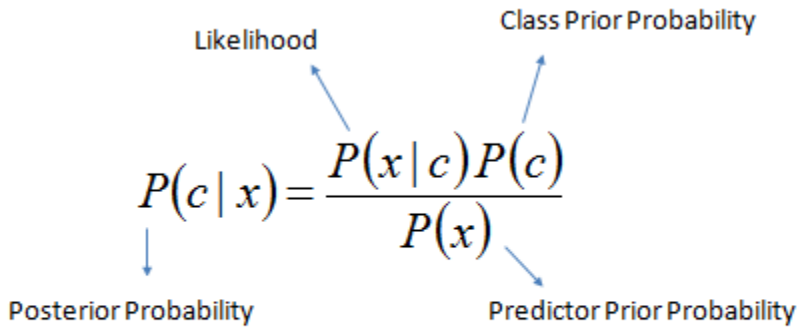
It is a classification technique based on Bayes' theorem with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.

Naive Bayesian model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

.

.



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Here,

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Example: Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play'. Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set to frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will pay if weather is sunny, is this statement is correct?

.

.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

We can solve it using above discussed method, so P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

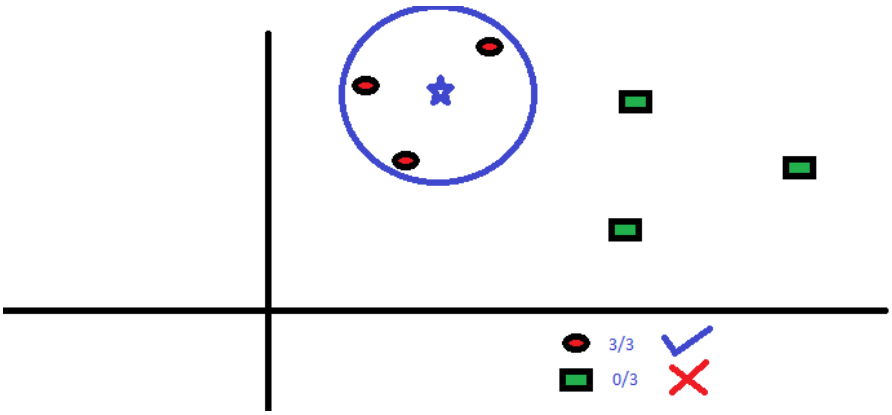
6. KNN (K- Nearest Neighbors)

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function.

These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If K = 1, then the case is simply assigned to the class of its nearest neighbor. At times, choosing K turns out to be a challenge while performing KNN modeling.

.

.



More: [Introduction to k-nearest neighbors : Simplified].

NN can easily be mapped to our real lives. If you want to learn about a person, of whom you have no information, you might like to find out about his close friends and the circles he moves in and gain access to his/her information!

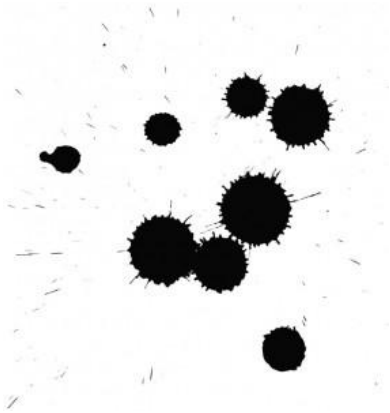
Things to consider before selecting KNN:

- KNN is computationally expensive
- Variables should be normalized else higher range variables can bias it
- Works on pre-processing stage more before going for KNN like outlier, noise removal

7. K-Means

It is a type of unsupervised algorithm which solves the clustering problem. Its procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters). Data points inside a cluster are homogeneous and heterogeneous to peer groups.

Remember figuring out shapes from ink blots? k means is somewhat similar this activity. You look at the shape and spread to decipher how many different clusters / population are present!



.

.

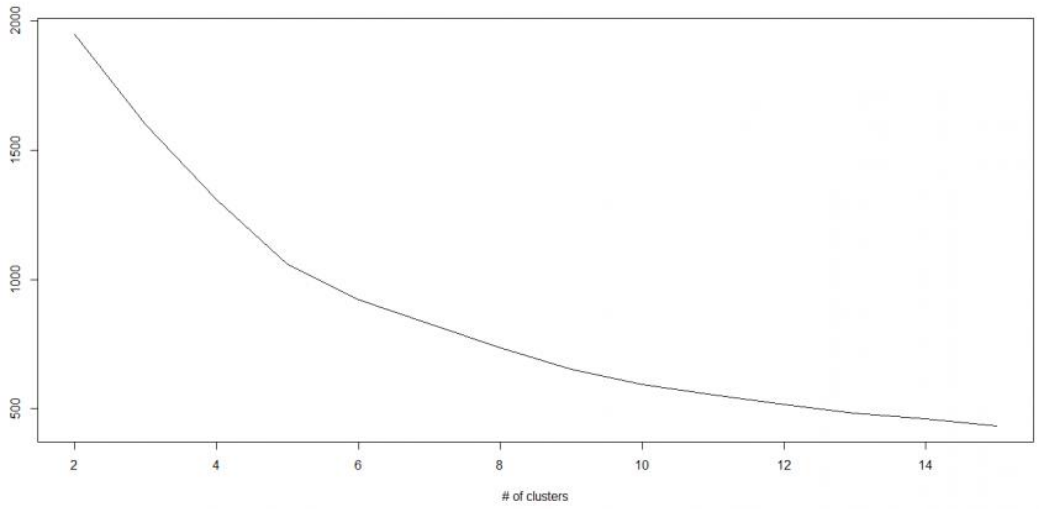
How K-means forms cluster:

1. K-means picks k number of points for each cluster known as centroids.
2. Each data point forms a cluster with the closest centroids i.e. k clusters.
3. Finds the centroid of each cluster based on existing cluster members. Here we have new centroids.
4. As we have new centroids, repeat step 2 and 3. Find the closest distance for each data point from new centroids and get associated with new k-clusters. Repeat this process until convergence occurs i.e. centroids does not change.

How to determine value of K:

In K-means, we have clusters and each cluster has its own centroid. Sum of square of difference between centroid and the data points within a cluster constitutes within sum of square value for that cluster. Also, when the sum of square values for all the clusters are added, it becomes total within sum of square value for the cluster solution.

We know that as the number of cluster increases, this value keeps on decreasing but if you plot the result you may see that the sum of squared distance decreases sharply up to some value of k, and then much more slowly after that. Here, we can find the optimum number of cluster.



8. Random Forest

Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we've collection of decision trees (so known as "Forest"). To classify a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is planted & grown as follows:

.

.

1. If the number of cases in the training set is N, then sample of N cases is taken at random but *with replacement*. This sample will be the training set for growing the tree.
2. If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
3. Each tree is grown to the largest extent possible. There is no pruning.

For more details on this algorithm, comparing with decision tree and tuning model parameters, I would suggest you to read these articles:

1. Introduction to Random forest – Simplified
2. Comparing a CART model to Random Forest (Part 1)
3. Comparing a Random Forest to a CART model (Part 2)
4. Tuning the parameters of your Random Forest model

9. Dimensionality Reduction Algorithms

In the last 4-5 years, there has been an exponential increase in data capturing at every possible stages. Corporates/ Government Agencies/ Research organisations are not only coming with new sources but also they are capturing data in great detail.

For example: E-commerce companies are capturing more details about customer like their demographics, web crawling history, what they like or dislike, purchase history, feedback and many others to give them personalized attention more than your nearest grocery shopkeeper.

As a data scientist, the data we are offered also consist of many features, this sounds good for building good robust model but there is a challenge. How'd you identify highly significant variable(s) out 1000 or 2000? In such cases, dimensionality reduction algorithm helps us along with various other algorithms like Decision Tree, Random Forest, PCA, Factor Analysis, Identify based on correlation matrix, missing value ratio and others.

To know more about this algorithms, you can read "Beginners Guide To Learn Dimension Reduction Techniques".

10. Gradient Boosting Algorithms

10.1. GBM

GBM is a boosting algorithm used when we deal with plenty of data to make a prediction with high prediction power. Boosting is actually an ensemble of learning algorithms which combines the prediction of several base estimators in order to improve robustness over a single estimator. It combines multiple weak or average predictors to a build strong predictor. These boosting algorithms always work well in data science competitions like Kaggle, AV Hackathon, CrowdAnalytix.

.

.

GradientBoostingClassifier and Random Forest are two different boosting tree classifier and often people ask about the difference between these two algorithms.

10.2. XGBoost

Another classic gradient boosting algorithm that's known to be the decisive choice between winning and losing in some Kaggle competitions.

The XGBoost has an immensely high predictive power which makes it the best choice for accuracy in events as it possesses both linear model and the tree learning algorithm, making the algorithm almost 10x faster than existing gradient booster techniques.

The support includes various objective functions, including regression, classification and ranking.

One of the most interesting things about the XGBoost is that it is also called a regularized boosting technique. This helps to reduce overfit modelling and has a massive support for a range of languages such as Scala, Java, R, Python, Julia and C++.

Supports distributed and widespread training on many machines that encompass GCE, AWS, Azure and Yarn clusters. XGBoost can also be integrated with Spark, Flink and other cloud dataflow systems with a built in cross validation at each iteration of the boosting process.

10.3. LightGBM

LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency
- Lower memory usage
- Better accuracy
- Parallel and GPU learning supported
- Capable of handling large-scale data

The framework is a fast and high-performance gradient boosting one based on decision tree algorithms, used for ranking, classification and many other machine learning tasks. It was developed under the Distributed Machine Learning Toolkit Project of Microsoft.

Since the LightGBM is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise. So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms.

.

.

Also, it is surprisingly very fast, hence the word ‘Light’.

10.4. Catboost

CatBoost is a recently open-sourced machine learning algorithm from Yandex. It can easily integrate with deep learning frameworks like Google’s TensorFlow and Apple’s Core ML.

The best part about CatBoost is that it does not require extensive data training like other ML models, and can work on a variety of data formats; not undermining how robust it can be.

Make sure you handle missing data well before you proceed with the implementation.

Catboost can automatically deal with categorical variables without showing the type conversion error, which helps you to focus on tuning your model better rather than sorting out trivial errors.

Reference:- https://www.analyticsvidhya.com/

Simple Linear Regression Tutorial for Machine Learning

- caret - Classification and Regression Training: Unified interface to **~150 ML algorithms in R.**

Linear Models are one of the oldest and most well known statistical prediction algorithms which nowdays is often categorized as a "machine learning algorithm." Generalized Linear Models (GLMs) are are a framework for modeling a response variable y that is bounded or discrete.

- Poisson regression for count data.

.

.

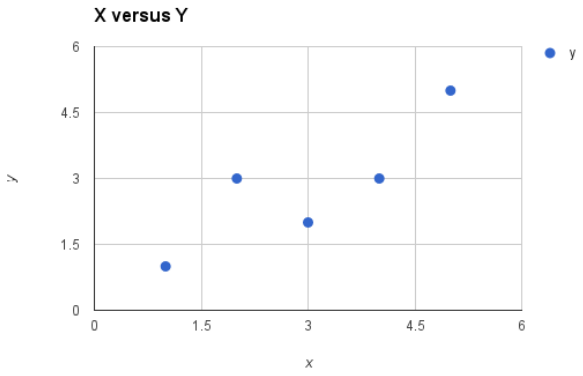
- Logistic regression and probit regression for binary data.
- Multinomial logistic regression and multinomial probit regression for categorical data.
- Ordered probit regression for ordinal data

Below is the raw data.

```
1 x     y
2 1     1
3 2     3
4 4     3
5 3     2
6 5     5
```

The attribute x is the input variable and y is the output variable that we are trying to predict. If we got more data, we would only have x values and we would be interested in predicting y values.

Below is a simple scatter plot of x versus y.



We can see the relationship between x and y looks kind of linear. As in, we could probably draw a line somewhere diagonally from the bottom left of the plot to the top right to generally describe the relationship between the data.

This is a good indication that using linear regression might be appropriate for this little dataset.

.

.

Simple Linear Regression

When we have a single input attribute (x) and we want to use linear regression, this is called simple linear regression.

If we had multiple input attributes (e.g. x1, x2, x3, etc.) This would be called multiple linear regression. The procedure for linear regression is different and simpler than that for multiple linear regression, so it is a good place to start.

In this section we are going to create a simple linear regression model from our training data, then make predictions for our training data to get an idea of how well the model learned the relationship in the data.

With simple linear regression we want to model our data as follows:
$y = B0 + B1 * x$

This is a line where y is the output variable we want to predict, x is the input variable we know and B0 and B1 are coefficients that we need to estimate that move the line around.

Technically, B0 is called the intercept because it determines where the line intercepts the y-axis. In machine learning we can call this the bias, because it is added to offset all predictions that we make. The B1 term is called the slope because it defines the slope of the line or how x translates into a y value before we add our bias.

The goal is to find the best estimates for the coefficients to minimize the errors in predicting y from x.

Simple regression is great, because rather than having to search for values by trial and error or calculate them analytically using more advanced linear algebra, we can estimate them directly from our data.

We can start off by estimating the value for B1 as:

$B1 = sum((xi-mean(x)) * (yi-mean(y))) / sum((xi – mean(x))^2)$

.

.

Where mean() is the average value for the variable in our dataset. The xi and yi refer to the fact that we need to repeat these calculations across all values in our dataset and i refers to the i'th value of x or y.

We can calculate B0 using B1 and some statistics from our dataset, as follows:

B0 = mean(y) – B1 * mean(x)

Not that bad right? We can calculate these right in our spreadsheet.

Estimating The Slope (B1)

Let's start with the top part of the equation, the numerator.

First we need to calculate the mean value of x and y. The mean is calculated as:

1/n * sum(x)

Where n is the number of values (5 in this case). You can use the AVERAGE() function in your spreadsheet. Let's calculate the mean value of our x and y variables:

mean(x) = 3

mean(y) = 2.8

Now we need to calculate the error of each variable from the mean. Let's do this with x first:

x	mean(x)	x - mean(x)
1	**3**	**-2**
2	**3**	**-1**
4	**3**	**1**
3	**3**	**0**
5	**3**	**2**

.

.

Now let’s do that for the y variable

y	mean(y)	y - mean(y)
1	2.8	-1.8
3	2.8	0.2
3	2.8	0.2
2	2.8	-0.8
5	2.8	2.2

We now have the parts for calculating the numerator. All we need to do is multiple the error for each x with the error for each y and calculate the sum of these multiplications.

x - mean(x)	y - mean(y)	Multiplication
-2	-1.8	3.6
-1	0.2	-0.2
1	0.2	0.2
0	-0.8	0
2	2.2	4.4

Summing the final column we have calculated our numerator as 8.

Now we need to calculate the bottom part of the equation for calculating B1, or the denominator. This is calculated as the sum of the squared differences of each x value from the mean.

We have already calculated the difference of each x value from the mean, all we need to do is square each value and calculate the sum.

x - mean(x)	squared
-2	4
-1	1
1	1

.

.

0 0

2 4

Calculating the sum of these squared values gives us up denominator of 10

Now we can calculate the value of our slope.

B1 = 8 / 10

B1 = 0.8

Estimating The Intercept (B0)

This is much easier as we already know the values of all of the terms involved.

B0 = mean(y) – B1 * mean(x)

or

B0 = 2.8 – 0.8 * 3

or

B0 = 0.4

Easy.

Making Predictions

We now have the coefficients for our simple linear regression equation.

y = B0 + B1 * x

or

y = 0.4 + 0.8 * x

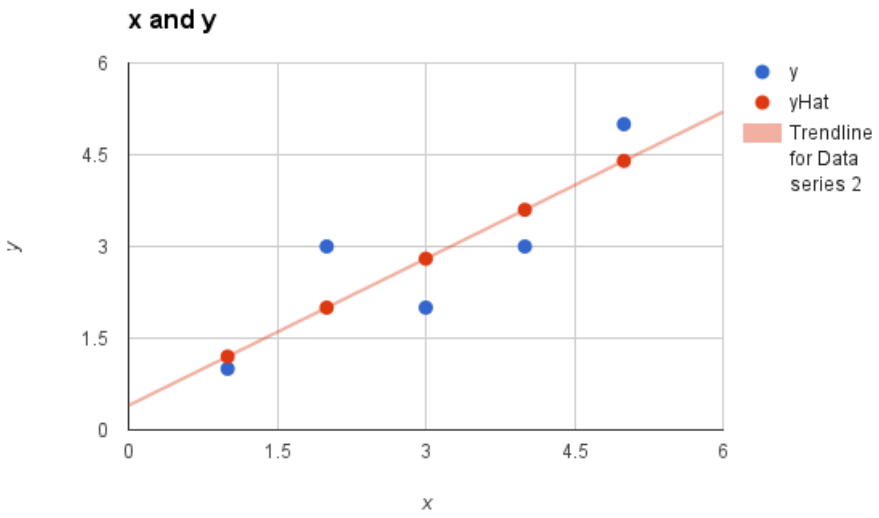
Let’s try out the model by making predictions for our training data.

.

.

x	y	predicted y
1	1	1.2
2	3	2
4	3	3.6
3	2	2.8
5	5	4.4

We can plot these predictions as a line with our data. This gives us a visual idea of how well the line models our data.



Estimating Error

We can calculate a error for our predictions called the Root Mean Squared Error or RMSE.
RMSE = sqrt(sum((pi – yi)^2)/n)

Where sqrt() is the square root function, p is the predicted value and y is the actual value, i is the index for a specific instance, n is the number of predictions, because we must calculate the error across all predicted values.

First we must calculate the difference between each model prediction and the actual y values.

pred-y	y	error

.

.

1.2	**1**	**0.2**
2	**3**	**-1**
3.6	**3**	**0.6**
2.8	**2**	**0.8**
4.4	**5**	**-0.6**

We can easily calculate the square of each of these error values (error*error or error^2).

error	**squared error**
0.2	**0.04**
-1	**1**
0.6	**0.36**
0.8	**0.64**
-0.6	**0.36**

The sum of these errors is 2.4 units, dividing by n and taking the square root gives us:

RMSE = 0.692

Or, each prediction is on average wrong by about 0.692 units.

Shortcut

Before we wrap up I want to show you a quick shortcut for calculating the coefficients.

Simple linear regression is the simplest form of regression and the most studied. There is a shortcut that you can use to quickly estimate the values for B0 and B1.

Really it is a shortcut for calculating B1. The calculation of B1 can be re-written as:

B1 = corr(x, y) * stdev(y) / stdev(x)

Where corr(x) is the correlation between x and y an stdev() is the calculation of the standard deviation for a variable.

.

.

Correlation (also known as Pearson's correlation coefficient) is a measure of how related two variables are in the range of -1 to 1. A value of 1 indicates that the two variables are perfectly positively correlated, they both move in the same direction and a value of -1 indicates that they are perfectly negatively correlated, when one moves the other moves in the other direction.

Standard deviation is a measure of how much on average the data is spread out from the mean.

You can use the function PEARSON() in your spreadsheet to calculate the correlation of x and y as 0.852 (highly correlated) and the function STDEV() to calculate the standard deviation of x as 1.5811 and y as 1.4832.

Plugging these values in we have:

B1 = 0.852 * 1.4832 / 1.5811

B1 = 0.799

Close enough to the above value of 0.8. Note that we get 0.8 if we use the fuller precision in our spreadsheet for the correlation and standard deviation equations.

Summary

In this post you discovered how to implement linear regression step-by-step in a spreadsheet. You learned:

- How to make predictions using your learned model.

Categorize the algorithms you may come across in the field.

- The first is a grouping of algorithms by the **learning style**.
- The second is a grouping of algorithms by **similarity** in form or function (like grouping similar animals together).

Both approaches are useful, but we will focus in on the grouping of algorithms by similarity and go on a tour of a variety of different algorithm types.

After reading this post, you will have a much better understanding of the most popular machine learning algorithms for supervised learning and how they are related.

Algorithms Grouped by Learning Style

.

.

There are different ways an algorithm can model a problem based on its interaction with the experience or environment or whatever we want to call the input data.

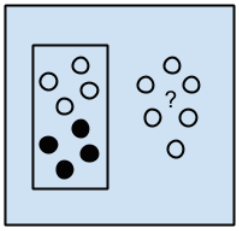
It is popular in machine learning and artificial intelligence textbooks to first consider the learning styles that an algorithm can adopt.

There are only a few main learning styles or learning models that an algorithm can have and we'll go through them here with a few examples of algorithms and problem types that they suit.

This taxonomy or way of organizing machine learning algorithms is useful because it forces you to think about the roles of the input data and the model preparation process and select one that is the most appropriate for your problem in order to get the best result.

Let's take a look at three different learning styles in machine learning algorithms:

1. Supervised Learning



Supervised Learning Algorithms

Input data is called training data and has a known label or result such as spam/not-spam or a stock price at a time.

A model is prepared through a training process in which it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data.

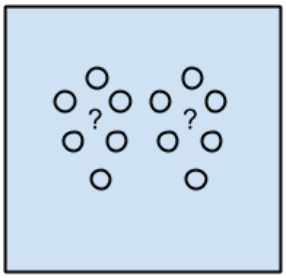
Example problems are classification and regression.

Example algorithms include Logistic Regression and the Back Propagation Neural Network.

2. Unsupervised Learning

.

.



Unsupervised Learning Algorithms

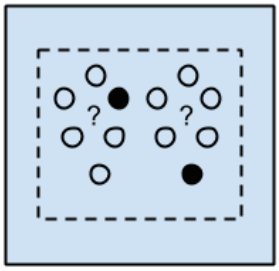
Input data is not labeled and does not have a known result.

A model is prepared by deducing structures present in the input data. This may be to extract general rules. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize data by similarity.

Example problems are clustering, dimensionality reduction and association rule learning.

Example algorithms include: the Apriori algorithm and k-Means.

3. Semi-Supervised Learning



Semi-supervised Learning Algorithms

Input data is a mixture of labeled and unlabelled examples.
There is a desired prediction problem but the model must learn the structures to organize the data as well as make predictions.

Example problems are classification and regression.

.

.

Example algorithms are extensions to other flexible methods that make assumptions about how to model the unlabeled data.

Overview

When crunching data to model business decisions, you are most typically using supervised and unsupervised learning methods.

A hot topic at the moment is semi-supervised learning methods in areas such as image classification where there are large datasets with very few labeled examples.

Algorithms Grouped By Similarity

Algorithms are often grouped by similarity in terms of their function (how they work). For example, tree-based methods, and neural network inspired methods.

I think this is the most useful way to group algorithms and it is the approach we will use here.

This is a useful grouping method, but it is not perfect. There are still algorithms that could just as easily fit into multiple categories like Learning Vector Quantization that is both a neural network inspired method and an instance-based method. There are also categories that have the same name that describe the problem and the class of algorithm such as Regression and Clustering.

We could handle these cases by listing algorithms twice or by selecting the group that subjectively is the "best" fit. I like this latter approach of not duplicating algorithms to keep things simple.

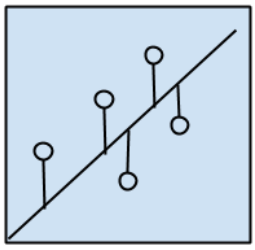
In this section, I list many of the popular machine learning algorithms grouped the way I think is the most intuitive. The list is not exhaustive in either the groups or the algorithms, but I think it is representative and will be useful to you to get an idea of the lay of the land.

Please Note: There is a strong bias towards algorithms used for classification and regression, the two most prevalent supervised machine learning problems you will encounter.
If you know of an algorithm or a group of algorithms not listed, put it in the comments and share it with us. Let's dive in.

Regression Algorithms

.

.



Regression Algorithms

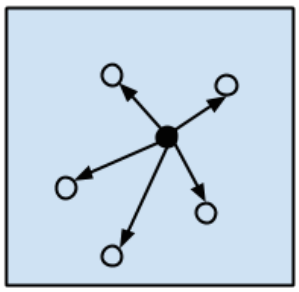
Regression is concerned with modeling the relationship between variables that is iteratively refined using a measure of error in the predictions made by the model.

Regression methods are a workhorse of statistics and have been co-opted into statistical machine learning. This may be confusing because we can use regression to refer to the class of problem and the class of algorithm. Really, regression is a process.

The most popular regression algorithms are:

- Ordinary Least Squares Regression (OLSR)
- Linear Regression
- Logistic Regression
- Stepwise Regression
- Multivariate Adaptive Regression Splines (MARS)
- Locally Estimated Scatterplot Smoothing (LOESS)

Instance-based Algorithms



Instance-based Algorithms

Instance-based learning model is a decision problem with instances or examples of training data that are deemed important or required to the model.

.

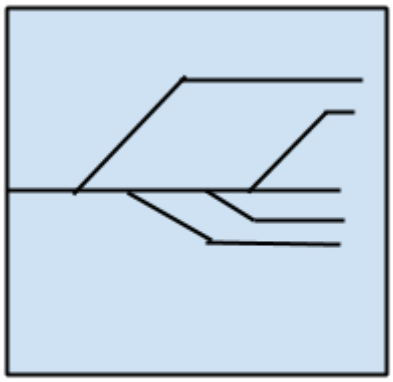
.

Such methods typically build up a database of example data and compare new data to the database using a similarity measure in order to find the best match and make a prediction. For this reason, instance-based methods are also called winner-take-all methods and memory-based learning. Focus is put on the representation of the stored instances and similarity measures used between instances.

The most popular instance-based algorithms are:

- k-Nearest Neighbor (kNN)
- Learning Vector Quantization (LVQ)
- Self-Organizing Map (SOM)
- Locally Weighted Learning (LWL)

Regularization Algorithms



Regularization Algorithms

An extension made to another method (typically regression methods) that penalizes models based on their complexity, favoring simpler models that are also better at generalizing.

I have listed regularization algorithms separately here because they are popular, powerful and generally simple modifications made to other methods.

The most popular regularization algorithms are:

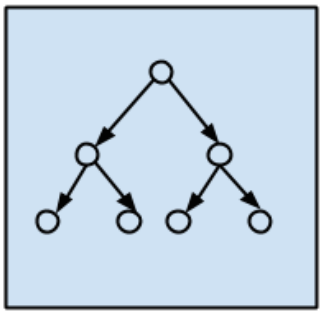
- Ridge Regression
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net

.

.

- Least-Angle Regression (LARS)

Decision Tree Algorithms



Decision Tree Algorithms

Decision tree methods construct a model of decisions made based on actual values of attributes in the data.

Decisions fork in tree structures until a prediction decision is made for a given record. Decision trees are trained on data for classification and regression problems. Decision trees are often fast and accurate and a big favorite in machine learning.

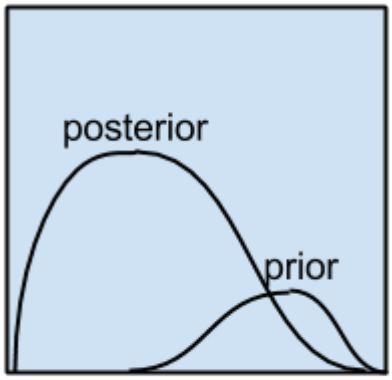
The most popular decision tree algorithms are:

- Classification and Regression Tree (CART)
- Iterative Dichotomiser 3 (ID3)
- C4.5 and C5.0 (different versions of a powerful approach)
- Chi-squared Automatic Interaction Detection (CHAID)
- Decision Stump
- M5
- Conditional Decision Trees

Bayesian Algorithms

.

.



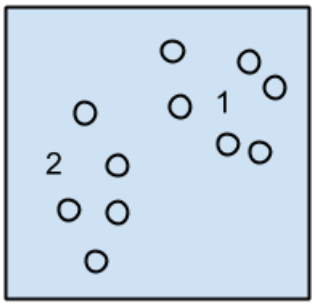
Bayesian Algorithms

Bayesian methods are those that explicitly apply Bayes' Theorem for problems such as classification and regression.

The most popular Bayesian algorithms are:

- Naive Bayes
- Gaussian Naive Bayes
- Multinomial Naive Bayes
- Averaged One-Dependence Estimators (AODE)
- Bayesian Belief Network (BBN)
- Bayesian Network (BN)

Clustering Algorithms



Clustering Algorithms

Clustering, like regression, describes the class of problem and the class of methods.

.

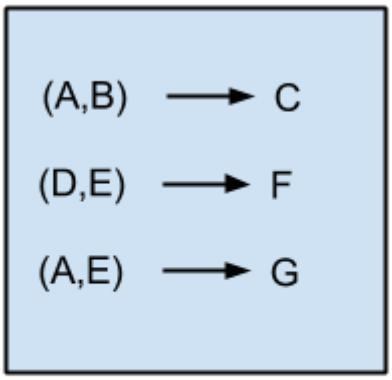
.

Clustering methods are typically organized by the modeling approaches such as centroid-based and hierarchal. All methods are concerned with using the inherent structures in the data to best organize the data into groups of maximum commonality.

The most popular clustering algorithms are:

- k-Means
- k-Medians
- Expectation Maximisation (EM)
- Hierarchical Clustering

Association Rule Learning Algorithms



Association Rule Learning Algorithms

Association rule learning methods extract rules that best explain observed relationships between variables in data.

These rules can discover important and commercially useful associations in large multidimensional datasets that can be exploited by an organization.

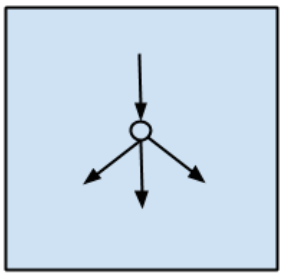
The most popular association rule learning algorithms are:

- Apriori algorithm
- Eclat algorithm

Artificial Neural Network Algorithms

.

.



Artificial Neural Network Algorithms

Artificial Neural Networks are models that are inspired by the structure and/or function of biological neural networks.

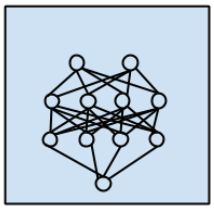
They are a class of pattern matching that are commonly used for regression and classification problems but are really an enormous subfield comprised of hundreds of algorithms and variations for all manner of problem types.

Note that I have separated out Deep Learning from neural networks because of the massive growth and popularity in the field. Here we are concerned with the more classical methods.

The most popular artificial neural network algorithms are:

- Perceptron
- Back-Propagation
- Hopfield Network
- Radial Basis Function Network (RBFN)

Deep Learning Algorithms



Deep Learning Algorithms

Deep Learning methods are a modern update to Artificial Neural Networks that exploit abundant cheap computation.

.

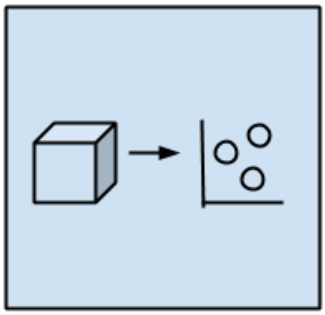
.

They are concerned with building much larger and more complex neural networks and, as commented on above, many methods are concerned with semi-supervised learning problems where large datasets contain very little labeled data.

The most popular deep learning algorithms are:

- Deep Boltzmann Machine (DBM)
- Deep Belief Networks (DBN)
- Convolutional Neural Network (CNN)
- Stacked Auto-Encoders

Dimensionality Reduction Algorithms



Dimensional Reduction Algorithms

Like clustering methods, dimensionality reduction seek and exploit the inherent structure in the data, but in this case in an unsupervised manner or order to summarize or describe data using less information.

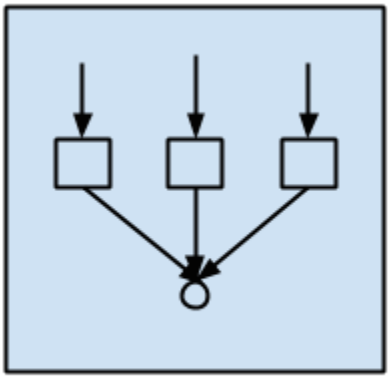
This can be useful to visualize dimensional data or to simplify data which can then be used in a supervised learning method. Many of these methods can be adapted for use in classification and regression.

- Principal Component Analysis (PCA)
- Principal Component Regression (PCR)
- Partial Least Squares Regression (PLSR)
- Sammon Mapping
- Multidimensional Scaling (MDS)
- Projection Pursuit
- Linear Discriminant Analysis (LDA)
- Mixture Discriminant Analysis (MDA)
- Quadratic Discriminant Analysis (QDA)
- Flexible Discriminant Analysis (FDA)

.

.

Ensemble Algorithms



Ensemble Algorithms

Ensemble methods are models composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction.

Much effort is put into what types of weak learners to combine and the ways in which to combine them. This is a very powerful class of techniques and as such is very popular.

- Boosting
- Bootstrapped Aggregation (Bagging)
- AdaBoost
- Stacked Generalization (blending)
- Gradient Boosting Machines (GBM)
- Gradient Boosted Regression Trees (GBRT)
- Random Forest

Other Algorithms

Many algorithms were not covered.

For example, what group would Support Vector Machines go into? Its own?

I did not cover algorithms from specialty tasks in the process of machine learning, such as:

- Feature selection algorithms
- Algorithm accuracy evaluation
- Performance measures

I also did not cover algorithms from specialty subfields of machine learning, such as:

- Computational intelligence (evolutionary algorithms, etc.)

.

.

- Computer Vision (CV)
- Natural Language Processing (NLP)
- Recommender Systems
- Reinforcement Learning
- GraphicalModels

---

https://github.com/josephmisiti/awesome-machine-learning/blob/master/README.md#r-general-purpose

Awesome Machine Learning

A curated list of awesome machine learning frameworks, libraries and software (by language). Inspired by awesome-php.
If you want to contribute to this list (please do), send me a pull request or contact me @josephmisiti Also, a listed repository should be deprecated if:

- Repository's owner explicitly say that "this library is not maintained".
- Not committed for long time (2~3 years).

Further resources:

- For a list of free machine learning books available for download, go here.
- For a list of (mostly) free machine learning courses available online, go here.
- For a list of blogs on data science and machine learning, go here.
- For a list of free-to-attend meetups and local events, go here.

Table of Contents

.

.

    - General-Purpose Machine Learning
    - Natural Language Processing
    - Sequence Analysis
    - Gesture Recognition
- Common Lisp
    - General-Purpose Machine Learning
- Clojure
    - Natural Language Processing
    - General-Purpose Machine Learning
    - Data Analysis / Data Visualization
- Crystal
    - General-Purpose Machine Learning
- Elixir
    - General-Purpose Machine Learning
    - Natural Language Processing
- Erlang
    - General-Purpose Machine Learning
- Go
    - Natural Language Processing
    - General-Purpose Machine Learning
    - Data Analysis / Data Visualization
- Haskell
    - General-Purpose Machine Learning
- Java
    - Natural Language Processing
    - General-Purpose Machine Learning
    - Data Analysis / Data Visualization
    - Deep Learning
- Javascript
    - Natural Language Processing
    - Data Analysis / Data Visualization
    - General-Purpose Machine Learning
    - Misc
- Julia
    - General-Purpose Machine Learning
    - Natural Language Processing
    - Data Analysis / Data Visualization
    - Misc Stuff / Presentations

.

.

- Lua
  - General-Purpose Machine Learning
  - Demos and Scripts
- Matlab
  - Computer Vision
  - Natural Language Processing
  - General-Purpose Machine Learning
  - Data Analysis / Data Visualization
- .NET
  - Computer Vision
  - Natural Language Processing
  - General-Purpose Machine Learning
  - Data Analysis / Data Visualization
- Objective C
  - General-Purpose Machine Learning
- OCaml
  - General-Purpose Machine Learning
- Perl
  - Data Analysis / Data Visualization
  - General-Purpose Machine Learning
- Perl 6
- PHP
  - Natural Language Processing
  - General-Purpose Machine Learning
- Python
  - Computer Vision
  - Natural Language Processing
  - General-Purpose Machine Learning
  - Data Analysis / Data Visualization
  - Misc Scripts / iPython Notebooks / Codebases
  - Kaggle Competition Source Code
  - Neural Networks
  - Reinforcement Learning
- Ruby
  - Natural Language Processing
  - General-Purpose Machine Learning
  - Data Analysis / Data Visualization
  - Misc

.

.

APL

General-Purpose Machine Learning

- naive-apl - Naive Bayesian Classifier implementation in APL

C

General-Purpose Machine Learning

- Darknet - Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation.
- Recommender - A C library for product recommendations/suggestions using collaborative filtering (CF).
- Hybrid Recommender System - A hybrid recommender system based upon scikit-learn algorithms.

.

.

Computer Vision

- CCV - C-based/Cached/Core Computer Vision Library, A Modern Computer Vision Library
- VLFeat - VLFeat is an open and portable library of computer vision algorithms, which has Matlab toolbox

Speech Recognition

- HTK -The Hidden Markov Model Toolkit (HTK) is a portable toolkit for building and manipulating hidden Markov models.

C++

Computer Vision

- DLib - DLib has C++ and Python interfaces for face detection and training general object detectors.
- EBLearn - Eblearn is an object-oriented C++ library that implements various machine learning models
- OpenCV - OpenCV has C++, C, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.
- VIGRA - VIGRA is a generic cross-platform C++ computer vision and machine learning library for volumes of arbitrary dimensionality with Python bindings.

General-Purpose Machine Learning

- BanditLib - A simple Multi-armed Bandit library.
- Caffe - A deep learning framework developed with cleanliness, readability, and speed in mind. [DEEP LEARNING]
- CNTK - The Computational Network Toolkit (CNTK) by Microsoft Research, is a unified deep-learning toolkit that describes neural networks as a series of computational steps via a directed graph.
- CUDA - This is a fast C++/CUDA implementation of convolutional [DEEP LEARNING]
- CXXNET - Yet another deep learning framework with less than 1000 lines core code [DEEP LEARNING]
- DeepDetect - A machine learning API and server written in C++11. It makes state of the art machine learning easy to work with and integrate into existing applications.

.

.

- Distributed Machine learning Tool Kit (DMTK) - A distributed machine learning (parameter server) framework by Microsoft. Enables training models on large data sets across multiple machines. Current tools bundled with it include: LightLDA and Distributed (Multisense) Word Embedding.
- DLib - A suite of ML tools designed to be easy to imbed in other applications
- DSSTNE - A software library created by Amazon for training and deploying deep neural networks using GPUs which emphasizes speed and scale over experimental flexibility.
- DyNet - A dynamic neural network library working well with networks that have dynamic structures that change for every training instance. Written in C++ with bindings in Python.
- encog-cpp
- Fido - A highly-modular C++ machine learning library for embedded electronics and robotics.
- igraph - General purpose graph library
- Intel(R) DAAL - A high performance software library developed by Intel and optimized for Intel's architectures. Library provides algorithmic building blocks for all stages of data analytics and allows to process data in batch, online and distributed modes.
- LightGBM - Microsoft's fast, distributed, high performance gradient boosting (GBDT, GBRT, GBM or MART) framework based on decision tree algorithms, used for ranking, classification and many other machine learning tasks.
- libfm - A generic approach that allows to mimic most factorization models by feature engineering.
- MLDB - The Machine Learning Database is a database designed for machine learning. Send it commands over a RESTful API to store data, explore it using SQL, then train machine learning models and expose them as APIs.
- mlpack - A scalable C++ machine learning library
- proNet-core - A general-purpose network embedding framework: pair-wise representations optimization Network Edit
- ROOT - A modular scientific software framework. It provides all the functionalities needed to deal with big data processing, statistical analysis, visualization and storage.
- shark - A fast, modular, feature-rich open-source C++ machine learning library.
- Shogun - The Shogun Machine Learning Toolbox
- sofia-ml - Suite of fast incremental algorithms.
- Stan - A probabilistic programming language implementing full Bayesian statistical inference with Hamiltonian Monte Carlo sampling
- Timbl - A software package/C++ library implementing several memory-based learning algorithms, among which IB1-IG, an implementation of k-nearest neighbor classification, and IGTree, a decision-tree approximation of IB1-IG. Commonly used for NLP.
- Vowpal Wabbit (VW) - A fast out-of-core learning system.

.

.

- Warp-CTC - A fast parallel implementation of Connectionist Temporal Classification (CTC), on both CPU and GPU.
- XGBoost - A parallelized optimized general purpose gradient boosting library.
- LKYDeepNN - A header-only C++11 Neural Network library. Low dependency, native traditional chinese document.

Natural Language Processing

- BLLIP Parser - BLLIP Natural Language Parser (also known as the Charniak-Johnson parser)
- colibri-core - C++ library, command line tools, and Python binding for extracting and working with basic linguistic constructions such as n-grams and skipgrams in a quick and memory-efficient way.
- CRF++ - Open source implementation of Conditional Random Fields (CRFs) for segmenting/labeling sequential data & other Natural Language Processing tasks.
- CRFsuite - CRFsuite is an implementation of Conditional Random Fields (CRFs) for labeling sequential data.
- frog - Memory-based NLP suite developed for Dutch: PoS tagger, lemmatiser, dependency parser, NER, shallow parser, morphological analyzer.
- libfolia - C++ library for the FoLiA format
- MeTA - MeTA : ModErn Text Analysis is a C++ Data Sciences Toolkit that facilitates mining big text data.
- MIT Information Extraction Toolkit - C, C++, and Python tools for named entity recognition and relation extraction
- ucto - Unicode-aware regular-expression based tokenizer for various languages. Tool and C++ library. Supports FoLiA format.

Speech Recognition

- Kaldi - Kaldi is a toolkit for speech recognition written in C++ and licensed under the Apache License v2.0. Kaldi is intended for use by speech recognition researchers.

Sequence Analysis

- ToPS - This is an objected-oriented framework that facilitates the integration of probabilistic models for sequences over a user defined alphabet.

.

.

Gesture Detection

- grt - The Gesture Recognition Toolkit (GRT) is a cross-platform, open-source, C++ machine learning library designed for real-time gesture recognition.

Common Lisp

General-Purpose Machine Learning

- mgl - Neural networks (boltzmann machines, feed-forward and recurrent nets), Gaussian Processes
- mgl-gpr - Evolutionary algorithms
- cl-libsvm - Wrapper for the libsvm support vector machine library

Clojure

Natural Language Processing

- Clojure-openNLP - Natural Language Processing in Clojure (opennlp)
- Infections-clj - Rails-like inflection library for Clojure and ClojureScript

General-Purpose Machine Learning

- Touchstone - Clojure A/B testing library
- Clojush - The Push programming language and the PushGP genetic programming system implemented in Clojure
- Infer - Inference and machine learning in Clojure
- Clj-ML - A machine learning library for Clojure built on top of Weka and friends
- DL4CLJ - Clojure wrapper for Deeplearning4j
- Encog - Clojure wrapper for Encog (v3) (Machine-Learning framework that specializes in neural-nets)
- Fungp - A genetic programming library for Clojure
- Statistiker - Basic Machine Learning algorithms in Clojure.
- clortex - General Machine Learning library using Numenta's Cortical Learning Algorithm
- comportex - Functionally composable Machine Learning library using Numenta's Cortical Learning Algorithm

.

.

- cortex - Neural networks, regression and feature learning in Clojure.
- lambda-ml - Simple, concise implementations of machine learning techniques and utilities in Clojure.

Data Analysis / Data Visualization

- Incanter - Incanter is a Clojure-based, R-like platform for statistical computing and graphics.
- PigPen - Map-Reduce for Clojure.
- Envision - Clojure Data Visualisation library, based on Statistiker and D3

Crystal

General-Purpose Machine Learning

- machine - Simple machine learning algorithm.
- crystal-fann - FANN (Fast Artifical Neural Network) binding.

Elixir

General-Purpose Machine Learning

- Simple Bayes - A Simple Bayes / Naive Bayes implementation in Elixir.

Natural Language Processing

- Stemmer - An English (Porter2) stemming implementation in Elixir.

Erlang

General-Purpose Machine Learning

- Disco - Map Reduce in Erlang

Go

.

.

Natural Language Processing

- go-porterstemmer - A native Go clean room implementation of the Porter Stemming algorithm.
- paicehusk - Golang implementation of the Paice/Husk Stemming Algorithm.
- snowball - Snowball Stemmer for Go.
- go-ngram - In-memory n-gram index with compression.
- word-embedding - Word Embeddings: the full implementation of word2vec, GloVe in Go.

General-Purpose Machine Learning

- gago - Multi-population, flexible, parallel genetic algorithm.
- Go Learn - Machine Learning for Go
- go-pr - Pattern recognition package in Go lang.
- go-ml - Linear / Logistic regression, Neural Networks, Collaborative Filtering and Gaussian Multivariate Distribution
- bayesian - Naive Bayesian Classification for Golang.
- go-galib - Genetic Algorithms library written in Go / Golang
- Cloudforest - Ensembles of decision trees in go/Golang.
- gobrain - Neural Networks written in go
- GoNN - GoNN is an implementation of Neural Network in Go Language, which includes BPNN, RBF, PCN
- MXNet - Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Dep Scheduler; for Python, R, Julia, Go, Javascript and more.
- go-mxnet-predictor - Go binding for MXNet c_predict_api to do inference with pre-trained model
- neat - Plug-and-play, parallel Go framework for NeuroEvolution of Augmenting Topologies (NEAT)

Data Analysis / Data Visualization

- go-graph - Graph library for Go/Golang language.
- SVGo - The Go Language library for SVG generation
- RF - Random forests implementation in Go

Haskell

.

.

General-Purpose Machine Learning

- haskell-ml - Haskell implementations of various ML algorithms.
- HLearn - a suite of libraries for interpreting machine learning models according to their algebraic structure.
- hnn - Haskell Neural Network library.
- hopfield-networks - Hopfield Networks for unsupervised learning in Haskell.
- caffegraph - A DSL for deep neural networks
- LambdaNet - Configurable Neural Networks in Haskell

Java

Natural Language Processing

- Cortical.io - Retina: an API performing complex NLP operations (disambiguation, classification, streaming text filtering, etc...) as quickly and intuitively as the brain.
- IRIS - Cortical.io's FREE NLP, Retina API Analysis Tool (written in JavaFX!) - See the Tutorial Video
- CoreNLP - Stanford CoreNLP provides a set of natural language analysis tools which can take raw English language text input and give the base forms of words
- Stanford Parser - A natural language parser is a program that works out the grammatical structure of sentences
- Stanford POS Tagger - A Part-Of-Speech Tagger (POS Tagger
- Stanford Name Entity Recognizer - Stanford NER is a Java implementation of a Named Entity Recognizer.
- Stanford Word Segmenter - Tokenization of raw text is a standard pre-processing step for many NLP tasks.
- Tregex, Tsurgeon and Semgrex - Tregex is a utility for matching patterns in trees, based on tree relationships and regular expression matches on nodes (the name is short for "tree regular expressions").
- Stanford Phrasal: A Phrase-Based Translation System
- Stanford English Tokenizer - Stanford Phrasal is a state-of-the-art statistical phrase-based machine translation system, written in Java.
- Stanford Tokens Regex - A tokenizer divides text into a sequence of tokens, which roughly correspond to "words"
- Stanford Temporal Tagger - SUTime is a library for recognizing and normalizing time expressions.
- Stanford SPIED - Learning entities from unlabeled text starting with seed sets using patterns in an iterative fashion

.

.

- Stanford Topic Modeling Toolbox - Topic modeling tools to social scientists and others who wish to perform analysis on datasets
- Twitter Text Java - A Java implementation of Twitter's text processing library
- MALLET - A Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text.
- OpenNLP - a machine learning based toolkit for the processing of natural language text.
- LingPipe - A tool kit for processing text using computational linguistics.
- ClearTK - ClearTK provides a framework for developing statistical natural language processing (NLP) components in Java and is built on top of Apache UIMA.
- Apache cTAKES - Apache clinical Text Analysis and Knowledge Extraction System (cTAKES) is an open-source natural language processing system for information extraction from electronic medical record clinical free-text.
- ClearNLP - The ClearNLP project provides software and resources for natural language processing. The project started at the Center for Computational Language and EducAtion Research, and is currently developed by the Center for Language and Information Research at Emory University. This project is under the Apache 2 license.
- CogcompNLP - This project collects a number of core libraries for Natural Language Processing (NLP) developed in the University of Illinois' Cognitive Computation Group, for example illinois-core-utilities which provides a set of NLP-friendly data structures and a number of NLP-related utilities that support writing NLP applications, running experiments, etc, illinois-edison a library for feature extraction from illinois-core-utilities data structures and many other packages.

General-Purpose Machine Learning

- aerosolve - A machine learning library by Airbnb designed from the ground up to be human friendly.
- AMIDST Toolbox - A Java Toolbox for Scalable Probabilistic Machine Learning.
- Datumbox - Machine Learning framework for rapid development of Machine Learning and Statistical applications
- ELKI - Java toolkit for data mining. (unsupervised: clustering, outlier detection etc.)
- Encog - An advanced neural network and machine learning framework. Encog contains classes to create a wide variety of networks, as well as support classes to normalize and process data for these neural networks. Encog trains using multithreaded resilient propagation. Encog can also make use of a GPU to further speed processing time. A GUI based workbench is also provided to help model and train neural networks.
- FlinkML in Apache Flink - Distributed machine learning library in Flink
- H2O - ML engine that supports distributed learning on Hadoop, Spark or your laptop via APIs in R, Python, Scala, REST/JSON.

.

.

- htm.java - General Machine Learning library using Numenta's Cortical Learning Algorithm
- java-deeplearning - Distributed Deep Learning Platform for Java, Clojure, Scala
- Mahout - Distributed machine learning
- Meka - An open source implementation of methods for multi-label classification and evaluation (extension to Weka).
- MLlib in Apache Spark - Distributed machine learning library in Spark
- Hydrosphere Mist - a service for deployment Apache Spark MLLib machine learning models as realtime, batch or reactive web services.
- Neuroph - Neuroph is lightweight Java neural network framework
- ORYX - Lambda Architecture Framework using Apache Spark and Apache Kafka with a specialization for real-time large-scale machine learning.
- Samoa SAMOA is a framework that includes distributed machine learning for data streams with an interface to plug-in different stream processing platforms.
- RankLib - RankLib is a library of learning to rank algorithms
- rapaio - statistics, data mining and machine learning toolbox in Java
- RapidMiner - RapidMiner integration into Java code
- Stanford Classifier - A classifier is a machine learning tool that will take data items and place them into one of k classes.
- SmileMiner - Statistical Machine Intelligence & Learning Engine
- SystemML - flexible, scalable machine learning (ML) language.
- WalnutiQ - object oriented model of the human brain
- Weka - Weka is a collection of machine learning algorithms for data mining tasks
- LBJava - Learning Based Java is a modeling language for the rapid development of software systems, offers a convenient, declarative syntax for classifier and constraint definition directly in terms of the objects in the programmer's application.

Speech Recognition

- CMU Sphinx - Open Source Toolkit For Speech Recognition purely based on Java speech recognition library.

Data Analysis / Data Visualization

- Flink - Open source platform for distributed stream and batch data processing.
- Hadoop - Hadoop/HDFS
- Onyx - Distributed, masterless, high performance, fault tolerant data processing. Written entirely in Clojure.
- Spark - Spark is a fast and general engine for large-scale data processing.

.

.

- Storm - Storm is a distributed realtime computation system.
- Impala - Real-time Query for Hadoop
- DataMelt - Mathematics software for numeric computation, statistics, symbolic calculations, data analysis and data visualization.
- Dr. Michael Thomas Flanagan's Java Scientific Library

Deep Learning

- Deeplearning4j - Scalable deep learning for industry with parallel GPUs

Javascript

Natural Language Processing

- Twitter-text - A JavaScript implementation of Twitter's text processing library
- natural - General natural language facilities for node
- Knwl.js - A Natural Language Processor in JS
- Retext - Extensible system for analyzing and manipulating natural language
- TextProcessing - Sentiment analysis, stemming and lemmatization, part-of-speech tagging and chunking, phrase extraction and named entity recognition.
- NLP Compromise - Natural Language processing in the browser

Data Analysis / Data Visualization

- D3.js
- High Charts
- NVD3.js
- dc.js
- chartjs
- dimple
- amCharts
- D3xter - Straight forward plotting built on D3
- statkit - Statistics kit for JavaScript
- datakit - A lightweight framework for data analysis in JavaScript
- science.js - Scientific and statistical computing in JavaScript.
- Z3d - Easily make interactive 3d plots built on Three.js
- Sigma.js - JavaScript library dedicated to graph drawing.

.

.

- C3.js- customizable library based on D3.js for easy chart drawing.
- Datamaps- Customizable SVG map/geo visualizations using D3.js.
- ZingChart- library written on Vanilla JS for big data visualization.
- cheminfo - Platform for data visualization and analysis, using the visualizer project.
- Learn JS Data
- AnyChart
- FusionCharts

General-Purpose Machine Learning

- Convnet.js - ConvNetJS is a Javascript library for training Deep Learning models[DEEP LEARNING]
- Clusterfck - Agglomerative hierarchical clustering implemented in Javascript for Node.js and the browser
- Clustering.js - Clustering algorithms implemented in Javascript for Node.js and the browser
- Decision Trees - NodeJS Implementation of Decision Tree using ID3 Algorithm
- DN2A - Digital Neural Networks Architecture
- figue - K-means, fuzzy c-means and agglomerative clustering
- Gaussian Mixture Model - Unsupervised machine learning with multivariate Gaussian mixture model
- Node-fann - FANN (Fast Artificial Neural Network Library) bindings for Node.js
- Kmeans.js - Simple Javascript implementation of the k-means algorithm, for node.js and the browser
- LDA.js - LDA topic modeling for Node.js
- Learning.js - Javascript implementation of logistic regression/c4.5 decision tree
- Machine Learning - Machine learning library for Node.js
- machineJS - Automated machine learning, data formatting, ensembling, and hyperparameter optimization for competitions and exploration- just give it a .csv file!
- mil-tokyo - List of several machine learning libraries
- Node-SVM - Support Vector Machine for Node.js
- Brain - Neural networks in JavaScript **[Deprecated]**
- Brain.js - Neural networks in JavaScript - continued community fork of Brain
- Bayesian-Bandit - Bayesian bandit implementation for Node and the browser.
- Synaptic - Architecture-free neural network library for Node.js and the browser
- kNear - JavaScript implementation of the k nearest neighbors algorithm for supervised learning

.

.

- NeuralN - C++ Neural Network library for Node.js. It has advantage on large dataset and multi-threaded training.
- kalman - Kalman filter for Javascript.
- shaman - Node.js library with support for both simple and multiple linear regression.
- ml.js - Machine learning and numerical analysis tools for Node.js and the Browser!
- Pavlov.js - Reinforcement learning using Markov Decision Processes
- MXNet - Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Dep Scheduler; for Python, R, Julia, Go, Javascript and more.
- deeplearnjs - A hardware-accelerated machine intelligence library for the web

Misc

- sylvester - Vector and Matrix math for JavaScript.
- simple-statistics - A JavaScript implementation of descriptive, regression, and inference statistics. Implemented in literate JavaScript with no dependencies, designed to work in all modern browsers (including IE) as well as in Node.js.
- regression-js - A javascript library containing a collection of least squares fitting methods for finding a trend in a set of data.
- Lyric - Linear Regression library.
- GreatCircle - Library for calculating great circle distance.

Julia

General-Purpose Machine Learning

- MachineLearning - Julia Machine Learning library
- MLBase - A set of functions to support the development of machine learning algorithms
- PGM - A Julia framework for probabilistic graphical models.
- DA - Julia package for Regularized Discriminant Analysis
- Regression - Algorithms for regression analysis (e.g. linear regression and logistic regression)
- Local Regression - Local regression, so smooooth!
- Naive Bayes - Simple Naive Bayes implementation in Julia
- Mixed Models - A Julia package for fitting (statistical) mixed-effects models
- Simple MCMC - basic mcmc sampler implemented in Julia
- Distance - Julia module for Distance evaluation
- Decision Tree - Decision Tree Classifier and Regressor

.

.

- Neural - A neural network in Julia
- MCMC - MCMC tools for Julia
- Mamba - Markov chain Monte Carlo (MCMC) for Bayesian analysis in Julia
- GLM - Generalized linear models in Julia
- Gaussian Processes - Julia package for Gaussian processes
- Online Learning
- GLMNet - Julia wrapper for fitting Lasso/ElasticNet GLM models using glmnet
- Clustering - Basic functions for clustering data: k-means, dp-means, etc.
- SVM - SVM's for Julia
- Kernel Density - Kernel density estimators for julia
- Dimensionality Reduction - Methods for dimensionality reduction
- NMF - A Julia package for non-negative matrix factorization
- ANN - Julia artificial neural networks
- Mocha - Deep Learning framework for Julia inspired by Caffe
- XGBoost - eXtreme Gradient Boosting Package in Julia
- ManifoldLearning - A Julia package for manifold learning and nonlinear dimensionality reduction
- MXNet - Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Dep Scheduler; for Python, R, Julia, Go, Javascript and more.
- Merlin - Flexible Deep Learning Framework in Julia
- ROCAnalysis - Receiver Operating Characteristics and functions for evaluation probabilistic binary classifiers
- GaussianMixtures - Large scale Gaussian Mixture Models
- ScikitLearn - Julia implementation of the scikit-learn API
- Knet - Koç University Deep Learning Framework

Natural Language Processing

- Topic Models - TopicModels for Julia
- Text Analysis - Julia package for text analysis

Data Analysis / Data Visualization

- Graph Layout - Graph layout algorithms in pure Julia
- LightGraphs - Graph modeling and analysis
- Data Frames Meta - Metaprogramming tools for DataFrames

.

.

- Julia Data - library for working with tabular data in Julia
- Data Read - Read files from Stata, SAS, and SPSS
- Hypothesis Tests - Hypothesis tests for Julia
- Gadfly - Crafty statistical graphics for Julia.
- Stats - Statistical tests for Julia
- RDataSets - Julia package for loading many of the data sets available in R
- DataFrames - library for working with tabular data in Julia
- Distributions - A Julia package for probability distributions and associated functions.
- Data Arrays - Data structures that allow missing values
- Time Series - Time series toolkit for Julia
- Sampling - Basic sampling algorithms for Julia

Misc Stuff / Presentations

- DSP - Digital Signal Processing (filtering, periodograms, spectrograms, window functions).
- JuliaCon Presentations - Presentations for JuliaCon
- SignalProcessing - Signal Processing tools for Julia
- Images - An image library for Julia

Lua

General-Purpose Machine Learning

- Torch7
  - cephes - Cephes mathematical functions library, wrapped for Torch. Provides and wraps the 180+ special mathematical functions from the Cephes mathematical library, developed by Stephen L. Moshier. It is used, among many other places, at the heart of SciPy.
  - autograd - Autograd automatically differentiates native Torch code. Inspired by the original Python version.
  - graph - Graph package for Torch
  - randomkit - Numpy's randomkit, wrapped for Torch
  - signal - A signal processing toolbox for Torch-7. FFT, DCT, Hilbert, cepstrums, stft
  - nn - Neural Network package for Torch
  - torchnet - framework for torch which provides a set of abstractions aiming at encouraging code re-use as well as encouraging modular programming

.

.

- nngraph - This package provides graphical computation for nn library in Torch7.
- nnx - A completely unstable and experimental package that extends Torch's builtin nn library
- rnn - A Recurrent Neural Network library that extends Torch's nn. RNNs, LSTMs, GRUs, BRNNs, BLSTMs, etc.
- dpnn - Many useful features that aren't part of the main nn package.
- dp - A deep learning library designed for streamlining research and development using the Torch7 distribution. It emphasizes flexibility through the elegant use of object-oriented design patterns.
- optim - An optimization library for Torch. SGD, Adagrad, Conjugate-Gradient, LBFGS, RProp and more.
- unsup - A package for unsupervised learning in Torch. Provides modules that are compatible with nn (LinearPsd, ConvPsd, AutoEncoder, ...), and self-contained algorithms (k-means, PCA).
- manifold - A package to manipulate manifolds
- svm - Torch-SVM library
- lbfgs - FFI Wrapper for liblbfgs
- vowpalwabbit - An old vowpalwabbit interface to torch.
- OpenGM - OpenGM is a C++ library for graphical modeling, and inference. The Lua bindings provide a simple way of describing graphs, from Lua, and then optimizing them with OpenGM.
- sphagetti - Spaghetti (sparse linear) module for torch7 by @MichaelMathieu
- LuaSHKit - A lua wrapper around the Locality sensitive hashing library SHKit
- kernel smoothing - KNN, kernel-weighted average, local linear regression smoothers
- cutorch - Torch CUDA Implementation
- cunn - Torch CUDA Neural Network Implementation
- imgraph - An image/graph library for Torch. This package provides routines to construct graphs on images, segment them, build trees out of them, and convert them back to images.
- videograph - A video/graph library for Torch. This package provides routines to construct graphs on videos, segment them, build trees out of them, and convert them back to videos.
- saliency - code and tools around integral images. A library for finding interest points based on fast integral histograms.
- stitch - allows us to use hugin to stitch images and apply same stitching to a video sequence
- sfm - A bundle adjustment/structure from motion package
- fex - A package for feature extraction in Torch. Provides SIFT and dSIFT modules.

.

.

    - OverFeat - A state-of-the-art generic dense feature extractor
- Numeric Lua
- Lunatic Python
- SciLua
- Lua - Numerical Algorithms
- Lunum

Demos and Scripts

- Core torch7 demos repository.
    - linear-regression, logistic-regression
    - face detector (training and detection as separate demos)
    - mst-based-segmenter
    - train-a-digit-classifier
    - train-autoencoder
    - optical flow demo
    - train-on-housenumbers
    - train-on-cifar
    - tracking with deep nets
    - kinect demo
    - filter-bank visualization
    - saliency-networks
- Training a Convnet for the Galaxy-Zoo Kaggle challenge(CUDA demo)
- Music Tagging - Music Tagging scripts for torch7
- torch-datasets - Scripts to load several popular datasets including:
    - BSR 500
    - CIFAR-10
    - COIL
    - Street View House Numbers
    - MNIST
    - NORB
- Atari2600 - Scripts to generate a dataset with static frames from the Arcade Learning Environment

.

.

Matlab

Computer Vision

- Contourlets - MATLAB source code that implements the contourlet transform and its utility functions.
- Shearlets - MATLAB code for shearlet transform
- Curvelets - The Curvelet transform is a higher dimensional generalization of the Wavelet transform designed to represent images at different scales and different angles.
- Bandlets - MATLAB code for bandlet transform
- mexopencv - Collection and a development kit of MATLAB mex functions for OpenCV library

Natural Language Processing

- NLP - An NLP library for Matlab

General-Purpose Machine Learning

- Training a deep autoencoder or a classifier on MNIST digits - Training a deep autoencoder or a classifier on MNIST digits[DEEP LEARNING]
- Convolutional-Recursive Deep Learning for 3D Object Classification - Convolutional-Recursive Deep Learning for 3D Object Classification[DEEP LEARNING]
- t-Distributed Stochastic Neighbor Embedding - t-Distributed Stochastic Neighbor Embedding (t-SNE) is a (prize-winning) technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.
- Spider - The spider is intended to be a complete object orientated environment for machine learning in Matlab.
- LibSVM - A Library for Support Vector Machines
- LibLinear - A Library for Large Linear Classification
- Machine Learning Module - Class on machine w/ PDF, lectures, code
- Caffe - A deep learning framework developed with cleanliness, readability, and speed in mind.
- Pattern Recognition Toolbox - A complete object-oriented environment for machine learning in Matlab.
- Pattern Recognition and Machine Learning - This package contains the matlab implementation of the algorithms described in the book Pattern Recognition and Machine Learning by C. Bishop.

.

.

- Optunity - A library dedicated to automated hyperparameter optimization with a simple, lightweight API to facilitate drop-in replacement of grid search. Optunity is written in Python but interfaces seamlessly with MATLAB.

Data Analysis / Data Visualization

- matlab_gbl - MatlabBGL is a Matlab package for working with graphs.
- gamic - Efficient pure-Matlab implementations of graph algorithms to complement MatlabBGL's mex functions.

.NET

Computer Vision

- OpenCVDotNet - A wrapper for the OpenCV project to be used with .NET applications.
- Emgu CV - Cross platform wrapper of OpenCV which can be compiled in Mono to e run on Windows, Linus, Mac OS X, iOS, and Android.
- AForge.NET - Open source C# framework for developers and researchers in the fields of Computer Vision and Artificial Intelligence. Development has now shifted to GitHub.
- Accord.NET - Together with AForge.NET, this library can provide image processing and computer vision algorithms to Windows, Windows RT and Windows Phone. Some components are also available for Java and Android.

Natural Language Processing

- Stanford.NLP for .NET - A full port of Stanford NLP packages to .NET and also available precompiled as a NuGet package.

General-Purpose Machine Learning

- Accord-Framework -The Accord.NET Framework is a complete framework for building machine learning, computer vision, computer audition, signal processing and statistical applications.
- Accord.MachineLearning - Support Vector Machines, Decision Trees, Naive Bayesian models, K-means, Gaussian Mixture models and general algorithms such as Ransac, Cross-validation and Grid-Search for machine-learning applications. This package is part of the Accord.NET Framework.
- DiffSharp - An automatic differentiation (AD) library providing exact and efficient derivatives (gradients, Hessians, Jacobians, directional derivatives, and matrix-free

.

.

Hessian- and Jacobian-vector products) for machine learning and optimization applications. Operations can be nested to any level, meaning that you can compute exact higher-order derivatives and differentiate functions that are internally making use of differentiation, for applications such as hyperparameter optimization.

- Vulpes - Deep belief and deep learning implementation written in F# and leverages CUDA GPU execution with Alea.cuBase.
- Encog - An advanced neural network and machine learning framework. Encog contains classes to create a wide variety of networks, as well as support classes to normalize and process data for these neural networks. Encog trains using multithreaded resilient propagation. Encog can also make use of a GPU to further speed processing time. A GUI based workbench is also provided to help model and train neural networks.
- Neural Network Designer - DBMS management system and designer for neural networks. The designer application is developed using WPF, and is a user interface which allows you to design your neural network, query the network, create and configure chat bots that are capable of asking questions and learning from your feed back. The chat bots can even scrape the internet for information to return in their output as well as to use for learning.
- Infer.NET - Infer.NET is a framework for running Bayesian inference in graphical models. One can use Infer.NET to solve many different kinds of machine learning problems, from standard problems like classification, recommendation or clustering through to customised solutions to domain-specific problems. Infer.NET has been used in a wide variety of domains including information retrieval, bioinformatics, epidemiology, vision, and many others.

Data Analysis / Data Visualization

- numl - numl is a machine learning library intended to ease the use of using standard modeling techniques for both prediction and clustering.
- Math.NET Numerics - Numerical foundation of the Math.NET project, aiming to provide methods and algorithms for numerical computations in science, engineering and every day use. Supports .Net 4.0, .Net 3.5 and Mono on Windows, Linux and Mac; Silverlight 5, WindowsPhone/SL 8, WindowsPhone 8.1 and Windows 8 with PCL Portable Profiles 47 and 344; Android/iOS with Xamarin.
- Sho - Sho is an interactive environment for data analysis and scientific computing that lets you seamlessly connect scripts (in IronPython) with compiled code (in .NET) to enable fast and flexible prototyping. The environment includes powerful and efficient libraries for linear algebra as well as data visualization that can be used from any .NET language, as well as a feature-rich interactive shell for rapid development.

Objective C

.

.

General-Purpose Machine Learning

- YCML - A Machine Learning framework for Objective-C and Swift (OS X / iOS).
- MLPNeuralNet - Fast multilayer perceptron neural network library for iOS and Mac OS X. MLPNeuralNet predicts new examples by trained neural network. It is built on top of the Apple's Accelerate Framework, using vectorized operations and hardware acceleration if available.
- MAChineLearning - An Objective-C multilayer perceptron library, with full support for training through backpropagation. Implemented using vDSP and vecLib, it's 20 times faster than its Java equivalent. Includes sample code for use from Swift.
- BPN-NeuralNetwork - It implemented 3 layers neural network (Input Layer, Hidden Layer and Output Layer) and it named Back Propagation Neural Network (BPN). This network can be used in products recommendation, user behavior analysis, data mining and data analysis.
- Multi-Perceptron-NeuralNetwork - it implemented multi-perceptrons neural network (ニューラルネットワーク) based on Back Propagation Neural Network (BPN) and designed unlimited-hidden-layers.
- KRHebbian-Algorithm - It is a non-supervisor and self-learning algorithm (adjust the weights) in neural network of Machine Learning.
- KRKmeans-Algorithm - It implemented K-Means the clustering and classification algorithm. It could be used in data mining and image compression.
- KRFuzzyCMeans-Algorithm - It implemented Fuzzy C-Means (FCM) the fuzzy clustering / classification algorithm on Machine Learning. It could be used in data mining and image compression.

OCaml

General-Purpose Machine Learning

- Oml - A general statistics and machine learning library.
- GPR - Efficient Gaussian Process Regression in OCaml.
- Libra-Tk - Algorithms for learning and inference with discrete probabilistic models.
- TensorFlow - OCaml bindings for TensorFlow.

Perl

Data Analysis / Data Visualization

.

.

- Perl Data Language, a pluggable architecture for data and image processing, which can be used for machine learning.

General-Purpose Machine Learning

- MXnet for Deep Learning, in Perl, also released in CPAN.
- Paws::MachineLearning, using AWS machine learning platform from Perl.
- Algorithm::SVMLight, implementation of Support Vector Machines with SVMLight under it.
- Several machine learning and artificial intelligence models are included in the AI namespace. For instance, you can find Naïve Bayes.

Perl 6

- Support Vector Machines
- Naïve Bayes

Data Analysis / Data Visualization

- Perl Data Language, a pluggable architecture for data and image processing, which can be used for machine learning.

General-Purpose Machine Learning

PHP

Natural Language Processing

- jieba-php - Chinese Words Segmentation Utilities.

General-Purpose Machine Learning

- PHP-ML - Machine Learning library for PHP. Algorithms, Cross Validation, Neural Network, Preprocessing, Feature Extraction and much more in one library.
- PredictionBuilder - A library for machine learning that builds predictions using a linear regression.

.

.

Python

Computer Vision

- Scikit-Image - A collection of algorithms for image processing in Python.
- SimpleCV - An open source computer vision framework that gives access to several high-powered computer vision libraries, such as OpenCV. Written on Python and runs on Mac, Windows, and Ubuntu Linux.
- Vigranumpy - Python bindings for the VIGRA C++ computer vision library.
- OpenFace - Free and open source face recognition with deep neural networks.
- PCV - Open source Python module for computer vision
- face_recognition - Face recognition library that recognize and manipulate faces from Python or from the command line
- dockerface - Easy to install and use deep learning Faster R-CNN face detection for images and video in a docker container.

Natural Language Processing

- NLTK - A leading platform for building Python programs to work with human language data.
- Pattern - A web mining module for the Python programming language. It has tools for natural language processing, machine learning, among others.
- Quepy - A python framework to transform natural language questions to queries in a database query language
- TextBlob - Providing a consistent API for diving into common natural language processing (NLP) tasks. Stands on the giant shoulders of NLTK and Pattern, and plays nicely with both.
- YAlign - A sentence aligner, a friendly tool for extracting parallel sentences from comparable corpora.
- jieba - Chinese Words Segmentation Utilities.
- SnowNLP - A library for processing Chinese text.
- spammy - A library for email Spam filtering built on top of nltk
- loso - Another Chinese segmentation library.
- genius - A Chinese segment base on Conditional Random Field.
- KoNLPy - A Python package for Korean natural language processing.
- nut - Natural language Understanding Toolkit
- Rosetta - Text processing tools and wrappers (e.g. Vowpal Wabbit)

.

.

- BLLIP Parser - Python bindings for the BLLIP Natural Language Parser (also known as the Charniak-Johnson parser)
- PyNLPl - Python Natural Language Processing Library. General purpose NLP library for Python. Also contains some specific modules for parsing common NLP formats, most notably for FoLiA, but also ARPA language models, Moses phrasetables, GIZA++ alignments.
- python-ucto - Python binding to ucto (a unicode-aware rule-based tokenizer for various languages)
- python-frog - Python binding to Frog, an NLP suite for Dutch. (pos tagging, lemmatisation, dependency parsing, NER)
- python-zpar - Python bindings for ZPar, a statistical part-of-speech-tagger, constiuency parser, and dependency parser for English.
- colibri-core - Python binding to C++ library for extracting and working with with basic linguistic constructions such as n-grams and skipgrams in a quick and memory-efficient way.
- spaCy - Industrial strength NLP with Python and Cython.
- PyStanfordDependencies - Python interface for converting Penn Treebank trees to Stanford Dependencies.
- Distance - Levenshtein and Hamming distance computation
- Fuzzy Wuzzy - Fuzzy String Matching in Python
- jellyfish - a python library for doing approximate and phonetic matching of strings.
- editdistance - fast implementation of edit distance
- textacy - higher-level NLP built on Spacy
- stanford-corenlp-python - Python wrapper for Stanford CoreNLP
- CLTK - The Classical Language Toolkit
- rasa_nlu - turn natural language into structured data
- yase - Transcode sentence (or other sequence) to list of word vector
- Polyglot - Multilingual text (NLP) processing toolkit
- DrQA - Reading Wikipedia to answer open-domain questions

General-Purpose Machine Learning

- auto_ml - Automated machine learning for production and analytics. Lets you focus on the fun parts of ML, while outputting production-ready code, and detailed analytics of your dataset and results. Includes support for NLP, XGBoost, LightGBM, and soon, deep learning.
- machine_learning - automated build consisting of a web-interface, and set of programmatic-interface API, for support vector machines. Corresponding dataset(s)

.

.

are stored into a SQL database, then generated model(s) used for prediction(s), are stored into a NoSQL datastore.

- XGBoost - Python bindings for eXtreme Gradient Boosting (Tree) Library
- Bayesian Methods for Hackers - Book/iPython notebooks on Probabilistic Programming in Python
- Featureforge A set of tools for creating and testing machine learning features, with a scikit-learn compatible API
- MLlib in Apache Spark - Distributed machine learning library in Spark
- Hydrosphere Mist - a service for deployment Apache Spark MLLib machine learning models as realtime, batch or reactive web services.
- scikit-learn - A Python module for machine learning built on top of SciPy.
- metric-learn - A Python module for metric learning.
- SimpleAI Python implementation of many of the artificial intelligence algorithms described on the book "Artificial Intelligence, a Modern Approach". It focuses on providing an easy to use, well documented and tested library.
- astroML - Machine Learning and Data Mining for Astronomy.
- graphlab-create - A library with various machine learning models (regression, clustering, recommender systems, graph analytics, etc.) implemented on top of a disk-backed DataFrame.
- BigML - A library that contacts external servers.
- pattern - Web mining module for Python.
- NuPIC - Numenta Platform for Intelligent Computing.
- Pylearn2 - A Machine Learning library based on Theano.
- keras - Modular neural network library based on Theano.
- Lasagne - Lightweight library to build and train neural networks in Theano.
- hebel - GPU-Accelerated Deep Learning Library in Python.
- Chainer - Flexible neural network framework
- prophet - Fast and automated time series forecasting framework by Facebook.
- gensim - Topic Modelling for Humans.
- topik - Topic modelling toolkit
- PyBrain - Another Python Machine Learning Library.
- Brainstorm - Fast, flexible and fun neural networks. This is the successor of PyBrain.
- Crab - A flexible, fast recommender engine.
- python-recsys - A Python library for implementing a Recommender System.
- thinking bayes - Book on Bayesian Analysis
- Image-to-Image Translation with Conditional Adversarial Networks - Implementation of image to image (pix2pix) translation from the paper by isola et al.[DEEP LEARNING]

.

.

- Restricted Boltzmann Machines -Restricted Boltzmann Machines in Python. [DEEP LEARNING]
- Bolt - Bolt Online Learning Toolbox
- CoverTree - Python implementation of cover trees, near-drop-in replacement for scipy.spatial.kdtree
- nilearn - Machine learning for NeuroImaging in Python
- neuropredict - Aimed at novice machine learners and non-expert programmers, this package offers easy (no coding needed) and comprehensive machine learning (evaluation and full report of predictive performance WITHOUT requiring you to code) in Python for NeuroImaging and any other type of features. This is aimed at absorbing the much of the ML workflow, unlike other packages like nilearn and pymvpa, which require you to learn their API and code to produce anything useful.
- imbalanced-learn - Python module to perform under sampling and over sampling with various techniques.
- Shogun - The Shogun Machine Learning Toolbox
- Pyevolve - Genetic algorithm framework.
- Caffe - A deep learning framework developed with cleanliness, readability, and speed in mind.
- breze - Theano based library for deep and recurrent neural networks
- pyhsmm - library for approximate unsupervised inference in Bayesian Hidden Markov Models (HMMs) and explicit-duration Hidden semi-Markov Models (HSMMs), focusing on the Bayesian Nonparametric extensions, the HDP-HMM and HDP-HSMM, mostly with weak-limit approximations.
- mrjob - A library to let Python program run on Hadoop.
- SKLL - A wrapper around scikit-learn that makes it simpler to conduct experiments.
- neurolab - https://github.com/zueve/neurolab
- Spearmint - Spearmint is a package to perform Bayesian optimization according to the algorithms outlined in the paper: Practical Bayesian Optimization of Machine Learning Algorithms. Jasper Snoek, Hugo Larochelle and Ryan P. Adams. Advances in Neural Information Processing Systems, 2012.
- Pebl - Python Environment for Bayesian Learning
- Theano - Optimizing GPU-meta-programming code generating array oriented optimizing math compiler in Python
- TensorFlow - Open source software library for numerical computation using data flow graphs
- yahmm - Hidden Markov Models for Python, implemented in Cython for speed and efficiency.
- python-timbl - A Python extension module wrapping the full TiMBL C++ programming interface. Timbl is an elaborate k-Nearest Neighbours machine learning toolkit.
- deap - Evolutionary algorithm framework.

.

.

- pydeep - Deep Learning In Python
- mlxtend - A library consisting of useful tools for data science and machine learning tasks.
- neon - Nervana's high-performance Python-based Deep Learning framework [DEEP LEARNING]
- Optunity - A library dedicated to automated hyperparameter optimization with a simple, lightweight API to facilitate drop-in replacement of grid search.
- Neural Networks and Deep Learning - Code samples for my book "Neural Networks and Deep Learning" [DEEP LEARNING]
- Annoy - Approximate nearest neighbours implementation
- skflow - Simplified interface for TensorFlow, mimicking Scikit Learn.
- TPOT - Tool that automatically creates and optimizes machine learning pipelines using genetic programming. Consider it your personal data science assistant, automating a tedious part of machine learning.
- pgmpy A python library for working with Probabilistic Graphical Models.
- DIGITS - The Deep Learning GPU Training System (DIGITS) is a web application for training deep learning models.
- Orange - Open source data visualization and data analysis for novices and experts.
- MXNet - Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Dep Scheduler; for Python, R, Julia, Go, Javascript and more.
- milk - Machine learning toolkit focused on supervised classification.
- TFLearn - Deep learning library featuring a higher-level API for TensorFlow.
- REP - an IPython-based environment for conducting data-driven research in a consistent and reproducible way. REP is not trying to substitute scikit-learn, but extends it and provides better user experience.
- rgf_python - Python bindings for Regularized Greedy Forest (Tree) Library.
- skbayes - Python package for Bayesian Machine Learning with scikit-learn API
- fuku-ml - Simple machine learning library, including Perceptron, Regression, Support Vector Machine, Decision Tree and more, it's easy to use and easy to learn for beginners.
- Xcessiv - A web-based application for quick, scalable, and automated hyperparameter tuning and stacked ensembling
- PyTorch - Tensors and Dynamic neural networks in Python with strong GPU acceleration
- ML-From-Scratch - Implementations of Machine Learning models from scratch in Python with a focus on transparency. Aims to showcase the nuts and bolts of ML in an accessible way.

.

.

Data Analysis / Data Visualization

- SciPy - A Python-based ecosystem of open-source software for mathematics, science, and engineering.
- NumPy - A fundamental package for scientific computing with Python.
- Numba - Python JIT (just in time) complier to LLVM aimed at scientific Python by the developers of Cython and NumPy.
- NetworkX - A high-productivity software for complex networks.
- igraph - binding to igraph library - General purpose graph library
- Pandas - A library providing high-performance, easy-to-use data structures and data analysis tools.
- Open Mining - Business Intelligence (BI) in Python (Pandas web interface)
- PyMC - Markov Chain Monte Carlo sampling toolkit.
- zipline - A Pythonic algorithmic trading library.
- PyDy - Short for Python Dynamics, used to assist with workflow in the modeling of dynamic motion based around NumPy, SciPy, IPython, and matplotlib.
- SymPy - A Python library for symbolic mathematics.
- statsmodels - Statistical modeling and econometrics in Python.
- astropy - A community Python library for Astronomy.
- matplotlib - A Python 2D plotting library.
- bokeh - Interactive Web Plotting for Python.
- plotly - Collaborative web plotting for Python and matplotlib.
- vincent - A Python to Vega translator.
- d3py - A plotting library for Python, based on D3.js.
- PyDexter - Simple plotting for Python. Wrapper for D3xterjs; easily render charts in-browser.
- ggplot - Same API as ggplot2 for R.
- ggfortify - Unified interface to ggplot2 popular R packages.
- Kartograph.py - Rendering beautiful SVG maps in Python.
- pygal - A Python SVG Charts Creator.
- PyQtGraph - A pure-python graphics and GUI library built on PyQt4 / PySide and NumPy.
- pycascading
- Petrel - Tools for writing, submitting, debugging, and monitoring Storm topologies in pure Python.
- Blaze - NumPy and Pandas interface to Big Data.
- emcee - The Python ensemble sampling toolkit for affine-invariant MCMC.
- windML - A Python Framework for Wind Energy Analysis and Prediction

.

.

- vispy - GPU-based high-performance interactive OpenGL 2D/3D data visualization library
- cerebro2 A web-based visualization and debugging platform for NuPIC.
- NuPIC Studio An all-in-one NuPIC Hierarchical Temporal Memory visualization and debugging super-tool!
- SparklingPandas Pandas on PySpark (POPS)
- Seaborn - A python visualization library based on matplotlib
- bqplot - An API for plotting in Jupyter (IPython)
- pastalog - Simple, realtime visualization of neural network training performance.
- caravel - A data exploration platform designed to be visual, intuitive, and interactive.
- Dora - Tools for exploratory data analysis in Python.
- Ruffus - Computation Pipeline library for python.
- SOMPY - Self Organizing Map written in Python (Uses neural networks for data analysis).
- somoclu Massively parallel self-organizing maps: accelerate training on multicore CPUs, GPUs, and clusters, has python API.
- HDBScan - implementation of the hdbscan algorithm in Python - used for clustering
- visualize_ML - A python package for data exploration and data analysis.
- scikit-plot - A visualization library for quick and easy generation of common plots in data analysis and machine learning.
- Bowtie - A dashboard library for interactive visualizations using flask socketio and react.

Misc Scripts / iPython Notebooks / Codebases

- BioPy - Biologically-Inspired and Machine Learning Algorithms in Python.
- pattern_classification
- thinking stats 2
- hyperopt
- numpic
- 2012-paper-diginorm
- A gallery of interesting IPython notebooks
- ipython-notebooks
- data-science-ipython-notebooks - Continually updated Data Science Python Notebooks: Spark, Hadoop MapReduce, HDFS, AWS, Kaggle, scikit-learn, matplotlib, pandas, NumPy, SciPy, and various command lines.
- decision-weights
- Sarah Palin LDA - Topic Modeling the Sarah Palin emails.

.

.

- Diffusion Segmentation - A collection of image segmentation algorithms based on diffusion methods
- Scipy Tutorials - SciPy tutorials. This is outdated, check out scipy-lecture-notes
- Crab - A recommendation engine library for Python
- BayesPy - Bayesian Inference Tools in Python
- scikit-learn tutorials - Series of notebooks for learning scikit-learn
- sentiment-analyzer - Tweets Sentiment Analyzer
- sentiment_classifier - Sentiment classifier using word sense disambiguation.
- group-lasso - Some experiments with the coordinate descent algorithm used in the (Sparse) Group Lasso model
- jProcessing - Kanji / Hiragana / Katakana to Romaji Converter. Edict Dictionary & parallel sentences Search. Sentence Similarity between two JP Sentences. Sentiment Analysis of Japanese Text. Run Cabocha(ISO--8859-1 configured) in Python.
- mne-python-notebooks - IPython notebooks for EEG/MEG data processing using mne-python
- Neon Course - IPython notebooks for a complete course around understanding Nervana's Neon
- pandas cookbook - Recipes for using Python's pandas library
- climin - Optimization library focused on machine learning, pythonic implementations of gradient descent, LBFGS, rmsprop, adadelta and others
- Allen Downey's Data Science Course - Code for Data Science at Olin College, Spring 2014.
- Allen Downey's Think Bayes Code - Code repository for Think Bayes.
- Allen Downey's Think Complexity Code - Code for Allen Downey's book Think Complexity.
- Allen Downey's Think OS Code - Text and supporting code for Think OS: A Brief Introduction to Operating Systems.
- Python Programming for the Humanities - Course for Python programming for the Humanities, assuming no prior knowledge. Heavy focus on text processing / NLP.
- GreatCircle - Library for calculating great circle distance.
- Optunity examples - Examples demonstrating how to use Optunity in synergy with machine learning libraries.
- Dive into Machine Learning with Python Jupyter notebook and scikit-learn - "I learned Python by hacking first, and getting serious *later*. I wanted to do this with Machine Learning. If this is your style, join me in getting a bit ahead of yourself."
- TDB - TensorDebugger (TDB) is a visual debugger for deep learning. It features interactive, node-by-node debugging and visualization for TensorFlow.
- Suiron - Machine Learning for RC Cars.

.

.

- Introduction to machine learning with scikit-learn - IPython notebooks from Data School's video tutorials on scikit-learn.
- Practical XGBoost in Python - comprehensive online course about using XGBoost in Python

Neural Networks

- NeuralTalk - NeuralTalk is a Python+numpy project for learning Multimodal Recurrent Neural Networks that describe images with sentences.
- Neuron - Neuron is simple class for time series predictions. It's utilize LNU (Linear Neural Unit), QNU (Quadratic Neural Unit), RBF (Radial Basis Function), MLP (Multi Layer Perceptron), MLP-ELM (Multi Layer Perceptron - Extreme Learning Machine) neural networks learned with Gradient descent or LeLevenberg–Marquardt algorithm.
- Data Driven Code - Very simple implementation of neural networks for dummies in python without using any libraries, with detailed comments.

Kaggle Competition Source Code

- wiki challenge - An implementation of Dell Zhang's solution to Wikipedia's Participation Challenge on Kaggle
- kaggle insults - Kaggle Submission for "Detecting Insults in Social Commentary"
- kaggle_acquire-valued-shoppers-challenge - Code for the Kaggle acquire valued shoppers challenge
- kaggle-cifar - Code for the CIFAR-10 competition at Kaggle, uses cuda-convnet
- kaggle-blackbox - Deep learning made easy
- kaggle-accelerometer - Code for Accelerometer Biometric Competition at Kaggle
- kaggle-advertised-salaries - Predicting job salaries from ads - a Kaggle competition
- kaggle amazon - Amazon access control challenge
- kaggle-bestbuy_big - Code for the Best Buy competition at Kaggle
- kaggle-bestbuy_small
- Kaggle Dogs vs. Cats - Code for Kaggle Dogs vs. Cats competition
- Kaggle Galaxy Challenge - Winning solution for the Galaxy Challenge on Kaggle
- Kaggle Gender - A Kaggle competition: discriminate gender based on handwriting
- Kaggle Merck - Merck challenge at Kaggle
- Kaggle Stackoverflow - Predicting closed questions on Stack Overflow
- kaggle_acquire-valued-shoppers-challenge - Code for the Kaggle acquire valued shoppers challenge
- wine-quality - Predicting wine quality

.

.

Reinforcement Learning

- DeepMind Lab - DeepMind Lab is a 3D learning environment based on id Software's Quake III Arena via ioquake3 and other open source software. Its primary purpose is to act as a testbed for research in artificial intelligence, especially deep reinforcement learning.
- Gym - OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms.
- Universe - Universe is a software platform for measuring and training an AI's general intelligence across the world's supply of games, websites and other applications.
- ViZDoom - ViZDoom allows developing AI bots that play Doom using only the visual information (the screen buffer). It is primarily intended for research in machine visual learning, and deep reinforcement learning, in particular.

Ruby

Natural Language Processing

- Awesome NLP with Ruby - Curated link list for practical natural language processing in Ruby.
- Treat - Text REtrieval and Annotation Toolkit, definitely the most comprehensive toolkit I've encountered so far for Ruby
- Ruby Linguistics - Linguistics is a framework for building linguistic utilities for Ruby objects in any language. It includes a generic language-independent front end, a module for mapping language codes into language names, and a module which contains various English-language utilities.
- Stemmer - Expose libstemmer_c to Ruby
- Ruby Wordnet - This library is a Ruby interface to WordNet
- Raspel - raspell is an interface binding for ruby
- UEA Stemmer - Ruby port of UEALite Stemmer - a conservative stemmer for search and indexing
- Twitter-text-rb - A library that does auto linking and extraction of usernames, lists and hashtags in tweets

General-Purpose Machine Learning

- Awesome Machine Learning with Ruby - Curated list of ML related resources for Ruby
- Ruby Machine Learning - Some Machine Learning algorithms, implemented in Ruby
- Machine Learning Ruby

.

.

- jRuby Mahout - JRuby Mahout is a gem that unleashes the power of Apache Mahout in the world of JRuby.
- CardMagic-Classifier - A general classifier module to allow Bayesian and other types of classifications.
- rb-libsvm - Ruby language bindings for LIBSVM which is a Library for Support Vector Machines
- Random Forester - Creates Random Forest classifiers from PMML files

Data Analysis / Data Visualization

- rsruby - Ruby - R bridge
- data-visualization-ruby - Source code and supporting content for my Ruby Manor presentation on Data Visualisation with Ruby
- ruby-plot - gnuplot wrapper for ruby, especially for plotting roc curves into svg files
- plot-rb - A plotting library in Ruby built on top of Vega and D3.
- scruffy - A beautiful graphing toolkit for Ruby
- SciRuby
- Glean - A data management tool for humans
- Bioruby
- Arel

Misc

- Big Data For Chimps
- Listof - Community based data collection, packed in gem. Get list of pretty much anything (stop words, countries, non words) in txt, json or hash. Demo/Search for a list

Rust

General-Purpose Machine Learning

- deeplearn-rs - deeplearn-rs provides simple networks that use matrix multiplication, addition, and ReLU under the MIT license.
- rustlearn - a machine learning framework featuring logistic regression, support vector machines, decision trees and random forests.
- rusty-machine - a pure-rust machine learning library.
- leaf - open source framework for machine intelligence, sharing concepts from TensorFlow and Caffe. Available under the MIT license. **[Deprecated]**

.

.

- RustNN - RustNN is a feedforward neural network library.

R

General-Purpose Machine Learning

- ahaz - ahaz: Regularization for semiparametric additive hazards regression
- arules - arules: Mining Association Rules and Frequent Itemsets
- biglasso - biglasso: Extending Lasso Model Fitting to Big Data in R
- bigrf - bigrf: Big Random Forests: Classification and Regression Forests for Large Data Sets
- bigRR - bigRR: Generalized Ridge Regression (with special advantage for p >> n cases)
- bmrm - bmrm: Bundle Methods for Regularized Risk Minimization Package
- Boruta - Boruta: A wrapper algorithm for all-relevant feature selection
- bst - bst: Gradient Boosting
- C50 - C50: C5.0 Decision Trees and Rule-Based Models
- caret - Classification and Regression Training: Unified interface to ~150 ML algorithms in R.
- caretEnsemble - caretEnsemble: Framework for fitting multiple caret models as well as creating ensembles of such models.
- Clever Algorithms For Machine Learning
- CORElearn - CORElearn: Classification, regression, feature evaluation and ordinal evaluation
- CoxBoost - CoxBoost: Cox models by likelihood based boosting for a single survival endpoint or competing risks
- Cubist - Cubist: Rule- and Instance-Based Regression Modeling
- e1071 - e1071: Misc Functions of the Department of Statistics (e1071), TU Wien
- earth - earth: Multivariate Adaptive Regression Spline Models
- elasticnet - elasticnet: Elastic-Net for Sparse Estimation and Sparse PCA
- ElemStatLearn - ElemStatLearn: Data sets, functions and examples from the book: "The Elements of Statistical Learning, Data Mining, Inference, and Prediction" by Trevor Hastie, Robert Tibshirani and Jerome Friedman Prediction" by Trevor Hastie, Robert Tibshirani and Jerome Friedman
- evtree - evtree: Evolutionary Learning of Globally Optimal Trees
- forecast - forecast: Timeseries forecasting using ARIMA, ETS, STLM, TBATS, and neural network models
- forecastHybrid - forecastHybrid: Automatic ensemble and cross validation of ARIMA, ETS, STLM, TBATS, and neural network models from the "forecast" package

.

.

- fpc - fpc: Flexible procedures for clustering
- frbs - frbs: Fuzzy Rule-based Systems for Classification and Regression Tasks
- GAMBoost - GAMBoost: Generalized linear and additive models by likelihood based boosting
- gamboostLSS - gamboostLSS: Boosting Methods for GAMLSS
- gbm - gbm: Generalized Boosted Regression Models
- glmnet - glmnet: Lasso and elastic-net regularized generalized linear models
- glmpath - glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model
- GMMBoost - GMMBoost: Likelihood-based Boosting for Generalized mixed models
- grplasso - grplasso: Fitting user specified models with Group Lasso penalty
- grpreg - grpreg: Regularization paths for regression models with grouped covariates
- h2o - A framework for fast, parallel, and distributed machine learning algorithms at scale -- Deeplearning, Random forests, GBM, KMeans, PCA, GLM
- hda - hda: Heteroscedastic Discriminant Analysis
- Introduction to Statistical Learning
- ipred - ipred: Improved Predictors
- kernlab - kernlab: Kernel-based Machine Learning Lab
- klaR - klaR: Classification and visualization
- lars - lars: Least Angle Regression, Lasso and Forward Stagewise
- lasso2 - lasso2: L1 constrained estimation aka ‘lasso’
- LiblineaR - LiblineaR: Linear Predictive Models Based On The Liblinear C/C++ Library
- LogicReg - LogicReg: Logic Regression
- Machine Learning For Hackers
- maptree - maptree: Mapping, pruning, and graphing tree models
- mboost - mboost: Model-Based Boosting
- medley - medley: Blending regression models, using a greedy stepwise approach
- mlr - mlr: Machine Learning in R
- mvpart - mvpart: Multivariate partitioning
- ncvreg - ncvreg: Regularization paths for SCAD- and MCP-penalized regression models
- nnet - nnet: Feed-forward Neural Networks and Multinomial Log-Linear Models
- oblique.tree - oblique.tree: Oblique Trees for Classification Data
- pamr - pamr: Pam: prediction analysis for microarrays
- party - party: A Laboratory for Recursive Partytioning
- partykit - partykit: A Toolkit for Recursive Partytioning
- penalized - penalized: L1 (lasso and fused lasso) and L2 (ridge) penalized estimation in GLMs and in the Cox model

.

.

- penalizedLDA - penalizedLDA: Penalized classification using Fisher's linear discriminant
- penalizedSVM - penalizedSVM: Feature Selection SVM using penalty functions
- quantregForest - quantregForest: Quantile Regression Forests
- randomForest - randomForest: Breiman and Cutler's random forests for classification and regression
- randomForestSRC - randomForestSRC: Random Forests for Survival, Regression and Classification (RF-SRC)
- rattle - rattle: Graphical user interface for data mining in R
- rda - rda: Shrunken Centroids Regularized Discriminant Analysis
- rdetools - rdetools: Relevant Dimension Estimation (RDE) in Feature Spaces
- REEMtree - REEMtree: Regression Trees with Random Effects for Longitudinal (Panel) Data
- relaxo - relaxo: Relaxed Lasso
- rgenoud - rgenoud: R version of GENetic Optimization Using Derivatives
- rgp - rgp: R genetic programming framework
- Rmalschains - Rmalschains: Continuous Optimization using Memetic Algorithms with Local Search Chains (MA-LS-Chains) in R
- rminer - rminer: Simpler use of data mining methods (e.g. NN and SVM) in classification and regression
- ROCR - ROCR: Visualizing the performance of scoring classifiers
- RoughSets - RoughSets: Data Analysis Using Rough Set and Fuzzy Rough Set Theories
- rpart - rpart: Recursive Partitioning and Regression Trees
- RPMM - RPMM: Recursively Partitioned Mixture Model
- RSNNS - RSNNS: Neural Networks in R using the Stuttgart Neural Network Simulator (SNNS)
- RWeka - RWeka: R/Weka interface
- RXshrink - RXshrink: Maximum Likelihood Shrinkage via Generalized Ridge or Least Angle Regression
- sda - sda: Shrinkage Discriminant Analysis and CAT Score Variable Selection
- SDDA - SDDA: Stepwise Diagonal Discriminant Analysis
- SuperLearner and subsemble - Multi-algorithm ensemble learning packages.
- svmpath - svmpath: svmpath: the SVM Path algorithm
- tgp - tgp: Bayesian treed Gaussian process models
- tree - tree: Classification and regression trees
- varSelRF - varSelRF: Variable selection using random forests
- XGBoost.R - R binding for eXtreme Gradient Boosting (Tree) Library

.

.

- Optunity - A library dedicated to automated hyperparameter optimization with a simple, lightweight API to facilitate drop-in replacement of grid search. Optunity is written in Python but interfaces seamlessly to R.
- igraph - binding to igraph library - General purpose graph library
- MXNet - Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Dep Scheduler; for Python, R, Julia, Go, Javascript and more.
- TDSP-Utilities - Two data science utilities in R from Microsoft: 1) Interactive Data Exploration, Analysis, and Reporting (IDEAR) ; 2) Automated Modeling and Reporting (AMR).

Data Analysis / Data Visualization

- ggplot2 - A data visualization package based on the grammar of graphics.

SAS

General-Purpose Machine Learning

- Enterprise Miner - Data mining and machine learning that creates deployable models using a GUI or code.
- Factory Miner - Automatically creates deployable machine learning models across numerous market or customer segments using a GUI.

Data Analysis / Data Visualization

- SAS/STAT - For conducting advanced statistical analysis.
- University Edition - FREE! Includes all SAS packages necessary for data analysis and visualization, and includes online SAS courses.

High Performance Machine Learning

- High Performance Data Mining - Data mining and machine learning that creates deployable models using a GUI or code in an MPP environment, including Hadoop.
- High Performance Text Mining - Text mining using a GUI or code in an MPP environment, including Hadoop.

.

.

Natural Language Processing

- Contextual Analysis - Add structure to unstructured text using a GUI.
- Sentiment Analysis - Extract sentiment from text using a GUI.
- Text Miner - Text mining using a GUI or code.

Demos and Scripts

- ML_Tables - Concise cheat sheets containing machine learning best practices.
- enlighten-apply - Example code and materials that illustrate applications of SAS machine learning techniques.
- enlighten-integration - Example code and materials that illustrate techniques for integrating SAS with other analytics technologies in Java, PMML, Python and R.
- enlighten-deep - Example code and materials that illustrate using neural networks with several hidden layers in SAS.
- dm-flow - Library of SAS Enterprise Miner process flow diagrams to help you learn by example about specific data mining topics.

Scala

Natural Language Processing

- ScalaNLP - ScalaNLP is a suite of machine learning and numerical computing libraries.
- Breeze - Breeze is a numerical processing library for Scala.
- Chalk - Chalk is a natural language processing library.
- FACTORIE - FACTORIE is a toolkit for deployable probabilistic modeling, implemented as a software library in Scala. It provides its users with a succinct language for creating relational factor graphs, estimating parameters and performing inference.
- Montague - Montague is a semantic parsing library for Scala with an easy-to-use DSL.

Data Analysis / Data Visualization

- MLlib in Apache Spark - Distributed machine learning library in Spark
- Hydrosphere Mist - a service for deployment Apache Spark MLLib machine learning models as realtime, batch or reactive web services.
- Scalding - A Scala API for Cascading
- Summing Bird - Streaming MapReduce with Scalding and Storm
- Algebird - Abstract Algebra for Scala

.

.

- xerial - Data management utilities for Scala
- PredictionIO - PredictionIO, a machine learning server for software developers and data engineers.
- BIDMat - CPU and GPU-accelerated matrix library intended to support large-scale exploratory data analysis.
- Flink - Open source platform for distributed stream and batch data processing.
- Spark Notebook - Interactive and Reactive Data Science using Scala and Spark.

General-Purpose Machine Learning

- DeepLearning.scala - Creating statically typed dynamic neural networks from object-oriented & functional programming constructs.
- Conjecture - Scalable Machine Learning in Scalding
- brushfire - Distributed decision tree ensemble learning in Scala
- ganitha - scalding powered machine learning
- adam - A genomics processing engine and specialized file format built using Apache Avro, Apache Spark and Parquet. Apache 2 licensed.
- bioscala - Bioinformatics for the Scala programming language
- BIDMach - CPU and GPU-accelerated Machine Learning Library.
- Figaro - a Scala library for constructing probabilistic models.
- H2O Sparkling Water - H2O and Spark interoperability.
- FlinkML in Apache Flink - Distributed machine learning library in Flink
- DynaML - Scala Library/REPL for Machine Learning Research
- Saul - Flexible Declarative Learning-Based Programming.
- SwiftLearner - Simply written algorithms to help study ML or write your own implementations.
- Smile - Statistical Machine Intelligence and Learning Engine

Swift

General-Purpose Machine Learning

- Bender - Fast Neural Networks framework built on top of Metal. Supports TensorFlow models.
- Swift AI - Highly optimized artificial intelligence and machine learning library written in Swift.
- BrainCore - The iOS and OS X neural network framework

.

.

- swix - A bare bones library that includes a general matrix language and wraps some OpenCV for iOS development.
- DeepLearningKit an Open Source Deep Learning Framework for Apple's iOS, OS X and tvOS. It currently allows using deep convolutional neural network models trained in Caffe on Apple operating systems.
- AIToolbox - A toolbox framework of AI modules written in Swift: Graphs/Trees, Linear Regression, Support Vector Machines, Neural Networks, PCA, KMeans, Genetic Algorithms, MDP, Mixture of Gaussians.
- MLKit - A simple Machine Learning Framework written in Swift. Currently features Simple Linear Regression, Polynomial Regression, and Ridge Regression.
- Swift Brain - The first neural network / machine learning library written in Swift. This is a project for AI algorithms in Swift for iOS and OS X development. This project includes algorithms focused on Bayes theorem, neural networks, SVMs, Matrices, etc..
- Perfect TensorFlow - Swift Language Bindings of TensorFlow. Using native TensorFlow models on both macOS / Linux.
- Awesome CoreML - A curated list of pretrained CoreML models
- Awesome Core ML Models - A curated list of machine learning models in CoreML format.

TensorFlow

General-Purpose Machine Learning

- Awesome TensorFlow - A list of all things related to TensorFlow

********** THE END**********

.