CHEAT SHEET

MACHINE LEARNING FOR DUMMIES CHEAT SHEET

From **Machine Learning For Dummies**

By **John Paul Mueller, Luca Massaron**

Machine learning is an incredible technology that you use more often than you think today and with the potential to do even more tomorrow. The interesting thing about machine learning is that both R and Python make the task easier than more people realize because both languages come with a lot of built-in and extended support (through the use of libraries, datasets, and other resources). With that in mind, this cheat sheet helps you access the most commonly needed reminders for making your machine learning experience fast and easy.

CHOOSING THE RIGHT ALGORITHM FOR MACHINE LEARNING

Machine learning involves the use of many different algorithms. This table gives you a quick summary of the strengths and weaknesses of various algorithms.

Algorithm	Best at	Pros	Cons

Random Forest	Apt at almost any machine learning problem Bioinformatics	Can work in parallel Seldom overfits Automatically handles missing values No need to transform any variable No need to tweak parameters Can be used by almost anyone with excellent results	Difficult to interpret Weaker on regression when estimating values at the extremities of the distribution of response values Biased in multiclass problems toward more frequent classes
Gradient Boosting	Apt at almost any machine learning problem Search engines (solving the problem of learning to rank)	It can approximate most nonlinear function Best in class predictor Automatically handles missing values No need to transform any variable	It can overfit if run for too many iterations Sensitive to noisy data and outliers Doesn’t work well without parameter tuning

Linear regression	Baseline predictions Econometric predictions Modelling marketing responses	Simple to understand and explain It seldom overfits Using L1 & L2 regularization is effective in feature selection Fast to train Easy to train on big data thanks to its stochastic version	You have to work hard to make it fit nonlinear functions Can suffer from outliers
Support Vector Machines	Character recognition Image recognition Text classification	Automatic nonlinear feature creation Can approximate complex nonlinear functions	Difficult to interpret when applying nonlinear kernels Suffers from too many examples, after 10,000 examples it starts taking too long to train
K-nearest Neighbors	Computer vision Multilabel tagging Recommender systems Spell checking problems	Fast, lazy training Can naturally handle extreme multiclass problems (like tagging text)	Slow and cumbersome in the predicting phase Can fail to predict correctly due to the curse of dimensionality

Adaboost	Face detection	Automatically handles missing values No need to transform any variable It doesn’t overfit easily Few parameters to tweak It can leverage many different weak-learners	Sensitive to noisy data and outliers Never the best in class predictions
Naive Bayes	Face recognition Sentiment analysis Spam detection Text classification	Easy and fast to implement, doesn’t require too much memory and can be used for online learning Easy to understand Takes into account prior knowledge	Strong and unrealistic feature independence assumptions Fails estimating rare occurrences Suffers from irrelevant features

Neural Networks	Image recognition Language recognition and translation Speech recognition Vision recognition	Can approximate any nonlinear function Robust to outliers Works only with a portion of the examples (the support vectors)	Very difficult to set up Difficult to tune because of too many parameters and you have also to decide the architecture of the network Difficult to interpret Easy to overfit
Logistic regression	Ordering results by probability Modelling marketing responses	Simple to understand and explain It seldom overfits Using L1 & L2 regularization is effective in feature selection The best algorithm for predicting probabilities of an event Fast to train Easy to train on big data thanks to its stochastic version	You have to work hard to make it fit nonlinear functions Can suffer from outliers
SVD	Recommender systems	Can restructure data in a meaningful way	Difficult to understand why data has been restructured in a certain way

PCA	Removing collinearity Reducing dimensions of the dataset	Can reduce data dimensionality	Implies strong linear assumptions (components are a weighted summations of features)
K-means	Segmentation	Fast in finding clusters Can detect outliers in multiple dimensions	Suffers from multicollinearity Clusters are spherical, can’t detect groups of other shape Unstable solutions, depends on initialization

GETTING THE RIGHT LIBRARY FOR MACHINE LEARNING

When working with R and Python for machine learning, you gain the benefit of not having to reinvent the wheel when it comes to algorithms. There is a library available to meet your specific needs — you just need to know which one to use. This table provides you with a listing of the libraries used for machine learning for both R and Python. When you want to perform any algorithm-related task, simply load the library needed for that task into your programming environment.

Algorithm	**Python implementation**	**R implementation**
Adaboost	sklearn.ensemble.AdaBoostClassifier sklearn.ensemble.AdaBoostRegressor	library(ada) : ada

Gradient Boosting	sklearn.ensemble.GradientBoostingClassifier	library(gbm) : gbm
	sklearn.ensemble.GradientBoostingRegressor	
K-means	sklearn.cluster.KMeans	library(stats) : kmeans
	sklearn.cluster.MiniBatchKMeans	
K-nearest Neighbors	sklearn.neighbors.KNeighborsClassifier	library(class): knn
	sklearn.neighbors.KNeighborsRegressor	
Linear regression	sklearn.linear_model.LinearRegression	library(stats) : lm
	sklearn.linear_model.Ridge	library(stats) : glm
	sklearn.linear_model.Lasso	library(MASS) : lm.ridge
	sklearn.linear_model.ElasticNet	library(lars) : lars
	sklearn.linear_model.SGDRegressor	library(glmnet) : glmnet
Logistic regression	sklearn.linear_model.LogisticRegression	library(stats) : glm
	sklearn.linear_model.SGDClassifier	library(glmnet) : glmnet
Naive Bayes	sklearn.naive_bayes.GaussianNB	library(klaR) : NaiveBayes
	sklearn.naive_bayes.MultinomialNB	library(e1071) : naiveBayes
	sklearn.naive_bayes.BernoulliNB	
Neural Networks	sklearn.neural_network.BernoulliRBM	library(neuralnet) : neuralnet
	(in version 0.18 of Scikit-learn, a new implementation of supervised neural network will be introducted)	library(AMORE) : train
		library(nnet) : nnet

PCA	sklearn.decomposition.PCA	library(stats): princomp library(stats) : stats
Random Forest	sklearn.ensemble.RandomForestClassifier sklearn.ensemble.RandomForestRegressor sklearn.ensemble.ExtraTreesClassifier sklearn.ensemble.ExtraTreesRegressor	library(randomForest) : randomForest
Support Vector Machines	sklearn.svm.SVC sklearn.svm.LinearSVC sklearn.svm.NuSVC sklearn.svm.SVR sklearn.svm.LinearSVR sklearn.svm.NuSVR sklearn.svm.OneClassSVM	library(e1071) : svm
SVD	sklearn.decomposition.TruncatedSVD sklearn.decomposition.NMF	library(irlba) : irlba library(svd) : svd

LOCATING THE ALGORITHM YOU NEED FOR MACHINE LEARNING

There are a number of different algorithms you can use for machine learning. However, finding the specific algorithm you want to know about can be difficult. This table provides you with the online location for information about the algorithms used in machine learning.

Algorithm	Type	Python/R URL
Naive Bayes	Supervised classification, online learning	**http://scikit-learn.org/stable/modules/naive_bayes.html** **https://cran.r-project.org/web/packages/bnlearn/index.html**
PCA	Unsupervised	**http://scikit-learn.org/stable/modules/generated/sklearn.decomp** **https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_p**
SVD	Unsupervised	**http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.Trunc** **https://cran.r-project.org/web/packages/svd/index.html**
K-means	Unsupervised	**http://scikit-learn.org/stable/modules/generated/sklearn.cluster.K** **https://cran.r-project.org/web/packages/broom/vignettes/kmeans**
K-nearest Neighbors	Supervised regression and classification	**http://scikit-learn.org/stable/modules/neighbors.html** **https://cran.r-project.org/web/packages/kknn/index.html**
Linear Regression	Supervised regression, online learning	**http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearR** **https://cran.r-project.org/web/packages/phylolm/index.html**
Logistic Regression	Supervised classification, online learning	**http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Logistic** **https://cran.r-project.org/web/packages/HSAUR/vignettes/Ch_logistic_regression**

Neural Networks	Unsupervised Supervised regression and classification	http://scikit-learn.org/dev/modules/neural_networks_supervised.h https://cran.r-project.org/web/packages/neuralnet/index.html
Support Vector Machines	Supervised regression and classification	http://scikit-learn.org/stable/modules/svm.html https://cran.r-project.org/web/packages/e1071/index.html
Adaboost	Supervised classification	http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostC https://cran.r-project.org/web/packages/adabag/index.html
Gradient Boosting	Supervised regression and classification	http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientB https://cran.r-project.org/web/packages/gbm/index.html
Random Forest	Supervised regression and classification	http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomFo https://cran.r-project.org/web/packages/randomForest/index.htm