

# Machine Learning Overview

## MACHINE LEARNING IN EMOJI

### Becoming Human.AI

#### SUPERVISED

human builds model based on input / output

#### UNSUPERVISED

human input, machine output  
human utilizes if satisfactory

#### REINFORCEMENT

human input, machine output  
human reward/punish, cycle continues

#### BASIC REGRESSION

##### LINEAR

`linear_model.LinearRegression()`

Lots of numerical data



##### LOGISTIC

`linear_model.LogisticRegression()`

Target variable is categorical



#### CLUSTER ANALYSIS

##### K-MEANS

`cluster.KMeans()`

Similar datum into groups based on centroids



##### ANOMALY DETECTION

`covariance.EllipticalEnvelope()`

Finding outliers through grouping



#### CLASSIFICATION

##### NEURAL NET

`neural_network.MLPClassifier()`

Complex relationships. Prone to overfitting  
Basically magic.



##### K-NN

`neighbors.KNeighborsClassifier()`

Group membership based on proximity



##### DECISION TREE

`tree.DecisionTreeClassifier()`

If/then/else. Non-contiguous data.  
Can also be regression.



##### RANDOM FOREST

`ensemble.RandomForestClassifier()`

Find best split randomly  
Can also be regression



##### SVM

`svm.SVC()` `svm.LinearSVC()`

Maximum margin classifier. Fundamental Data Science algorithm



##### NAIVE BAYES

`GaussianNB()` `MultinomialNB()` `BernoulliNB()`

Updating knowledge step by step with new info



#### FEATURE REDUCTION

##### T-DISTRI STOCHASTIC NEIB EMBEDDING

`manifold.TSNE()`

Visual high dimensional data. Convert similarity to joint probabilities



##### PRINCIPLE COMPONENT ANALYSIS

`decomposition.PCA()`

Distill feature space into components that describe greatest variance



##### CANONICAL CORRELATION ANALYSIS

`decomposition.CCA()`

Making sense of cross-correlation matrices



##### LINEAR DISCRIMINANT ANALYSIS

`lda.LDA()`

Linear combination of features that separates classes



#### OTHER IMPORTANT CONCEPTS

##### BIAS VARIANCE TRADEOFF

##### UNDERFITTING / OVERFITTING

##### INERTIA

##### ACCURACY FUNCTION

$(TP+TN) / (P+N)$

##### PRECISION FUNCTION

`manifold.TSNE()`

##### SPECIFICITY FUNCTION

$TN / (FP+TN)$

##### SENSITIVITY FUNCTION

$TP / (TP+FN)$