http://web.cs.hacettepe.edu.tr/~aykut/classes/spring2013/bil682/tomgauld.jpg

DS501: Machine learning, Part 2

Prof. Randy Paffenroth
rcpaffenroth@wpi.edu

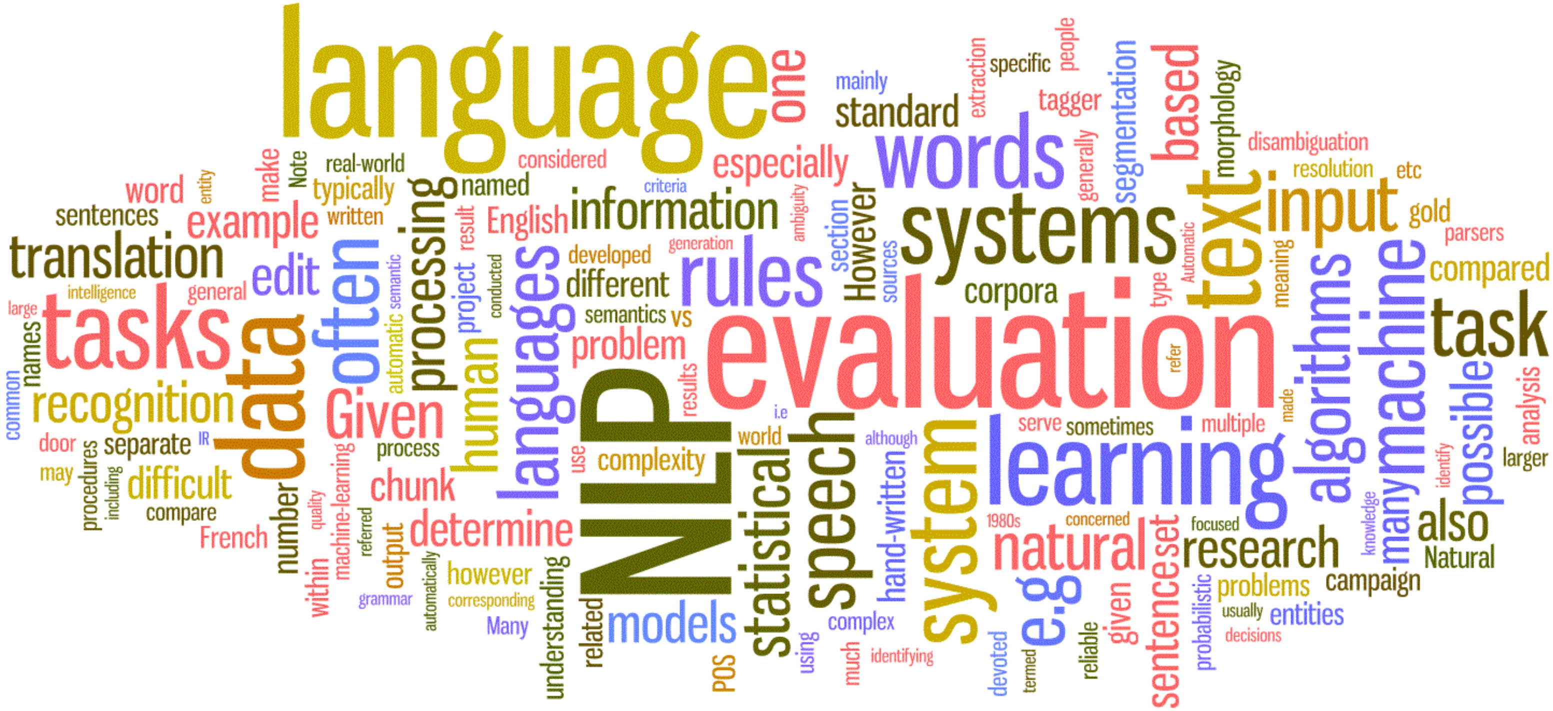
Worcester Polytechnic Institute

Announcements

- Midterms are being graded as we speak...

Announcements

- Case Study 3 is ready and lets have a conversation...



Course plan

- Original
  - Case study 3 out ~~3/23~~ 3/16
  - Case study 3 due 4/6
  - Case study 4 out 4/13
  - Case study 4 due 4/27
  - Final exam 4/27

- ~~Possible~~
  - ~~Case study 3 out 3/16~~
  - ~~Case study 3 due 3/30~~
  - ~~Case study 4 out 4/6~~
  - ~~Case study 4 due 4/20~~
  - ~~Final exam 4/27~~

Learning **objectives** for this machine learning class.

- Supervised Regression
  - Linear Regression
  - High dimensional and non-linear
  - Model selection
  - Ridge Regression
  - **Lasso Regression**

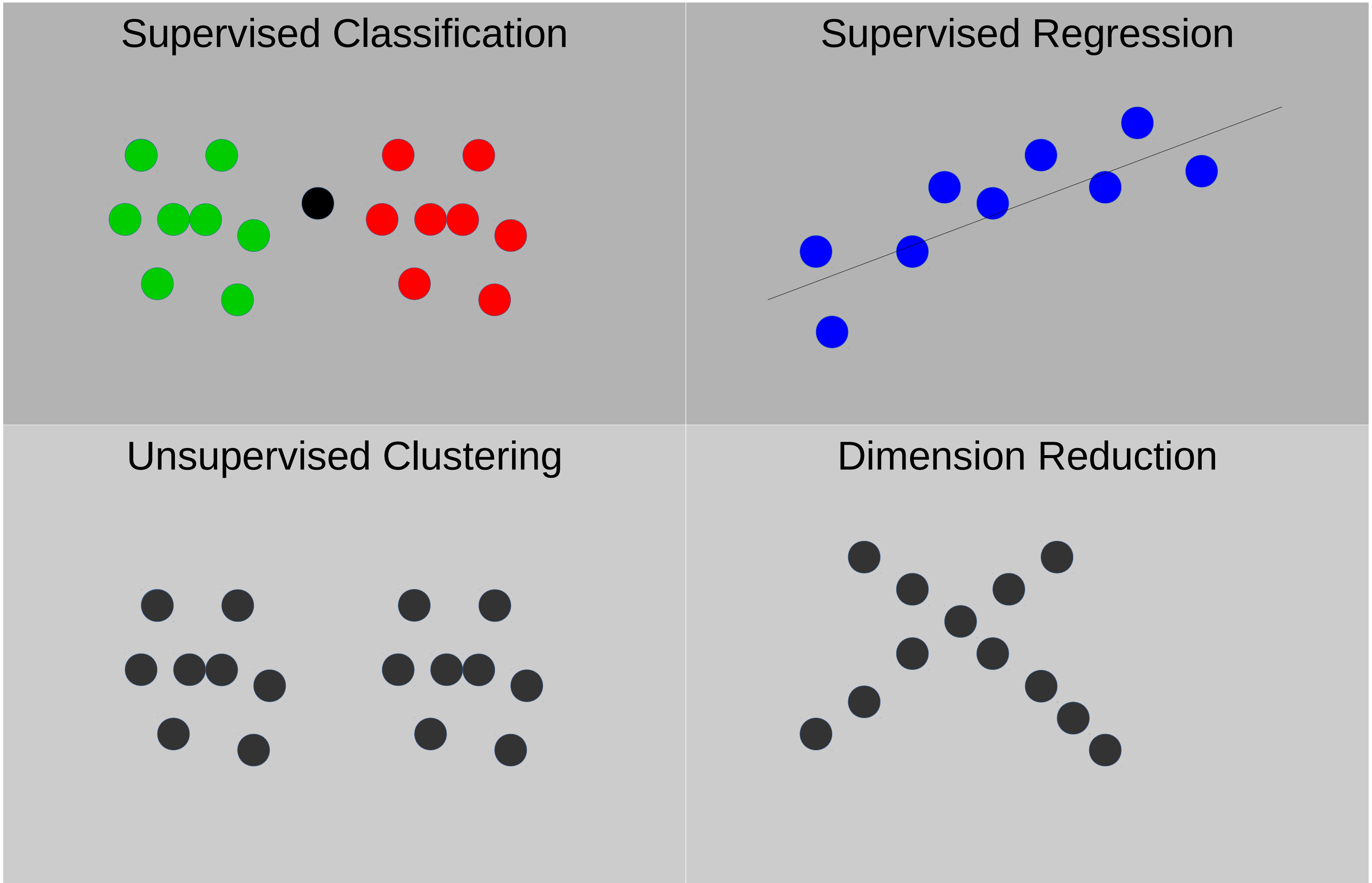
- Advanced techniques and unsupervised learning.
  - Trees
  - Ensemble learning
  - K-means
  - **Manifold learning**

- Learn some Python packages, including:
  - scikit-learn
  - mayavi

Inspired by: http://scikit-learn.org/stable/tutorial/statishttp://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.htmltical_inference/supervised_learning.html

Review!

The kinds of machine learning



Supervised Classification

Supervised Regression

Unsupervised Clustering

Dimension Reduction

Iris data set

Features:

sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

"Iris virginica" by Frank Mayfield - originally posted to Flickr as Iris virginica shrevei BLUE FLAG. Licensed under Creative Commons Attribution-Share Alike 2.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Iris_virginica.jpg#mediaviewer/File:Iris_virginica.jpg

Catagories:

setosa
versicolor
virginica

"Kosaciec szczecinkowaty Iris setosa". Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Kosaciec_szczecinkowaty_Iris_setosa.jpg#mediaviewer/File:Kosaciec_szczecinkowaty_Iris_setosa.jpg

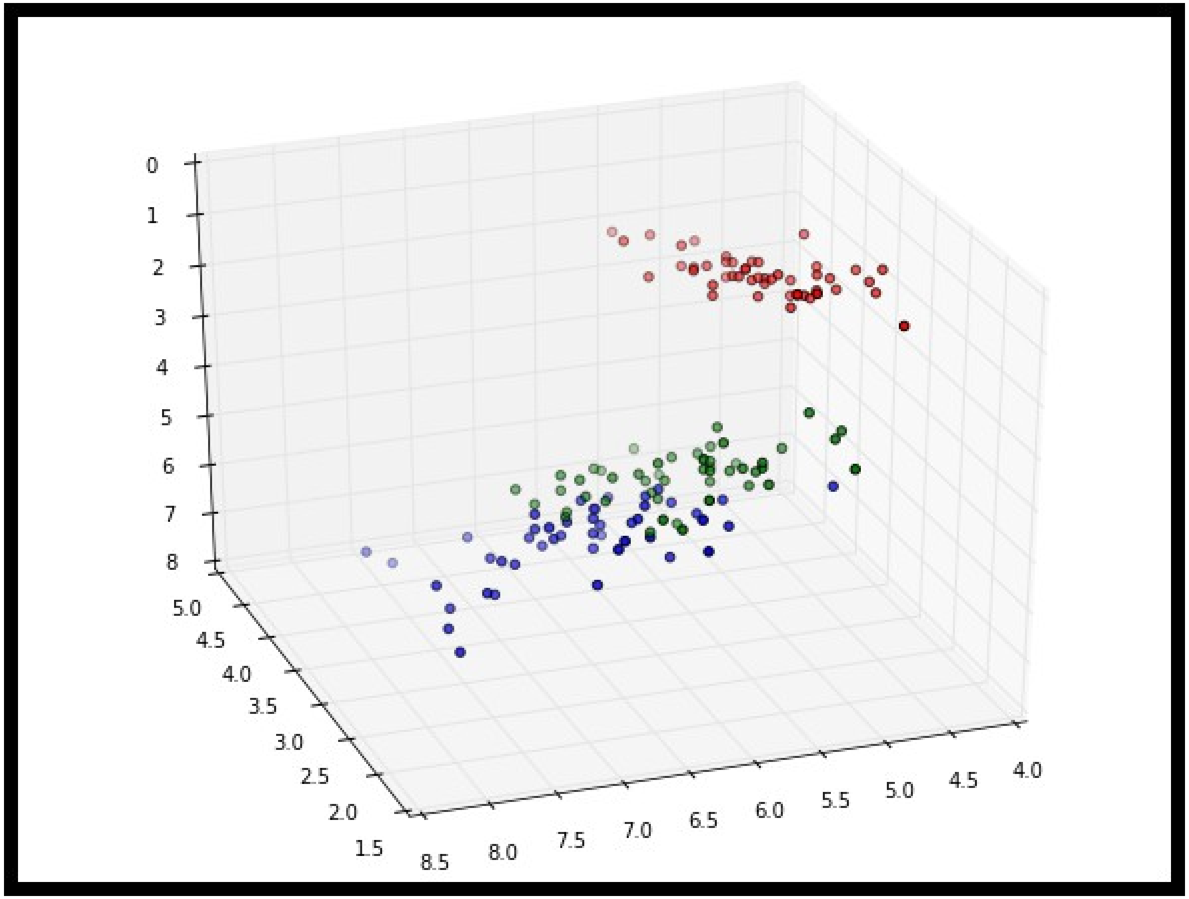
"Iris versicolor 3". Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Iris_versicolor_3.jpg#mediaviewer/File:Iris_versicolor_3.jpg

Iris data set

$f_1(x, y, z, w)$

$f_2(x, y, z, w)$

$f_3(x, y, z, w)$



What is PCA?

- Principle Component Analysis
  - Commonly used tool for visualization and data pre-processing.

Linear fn

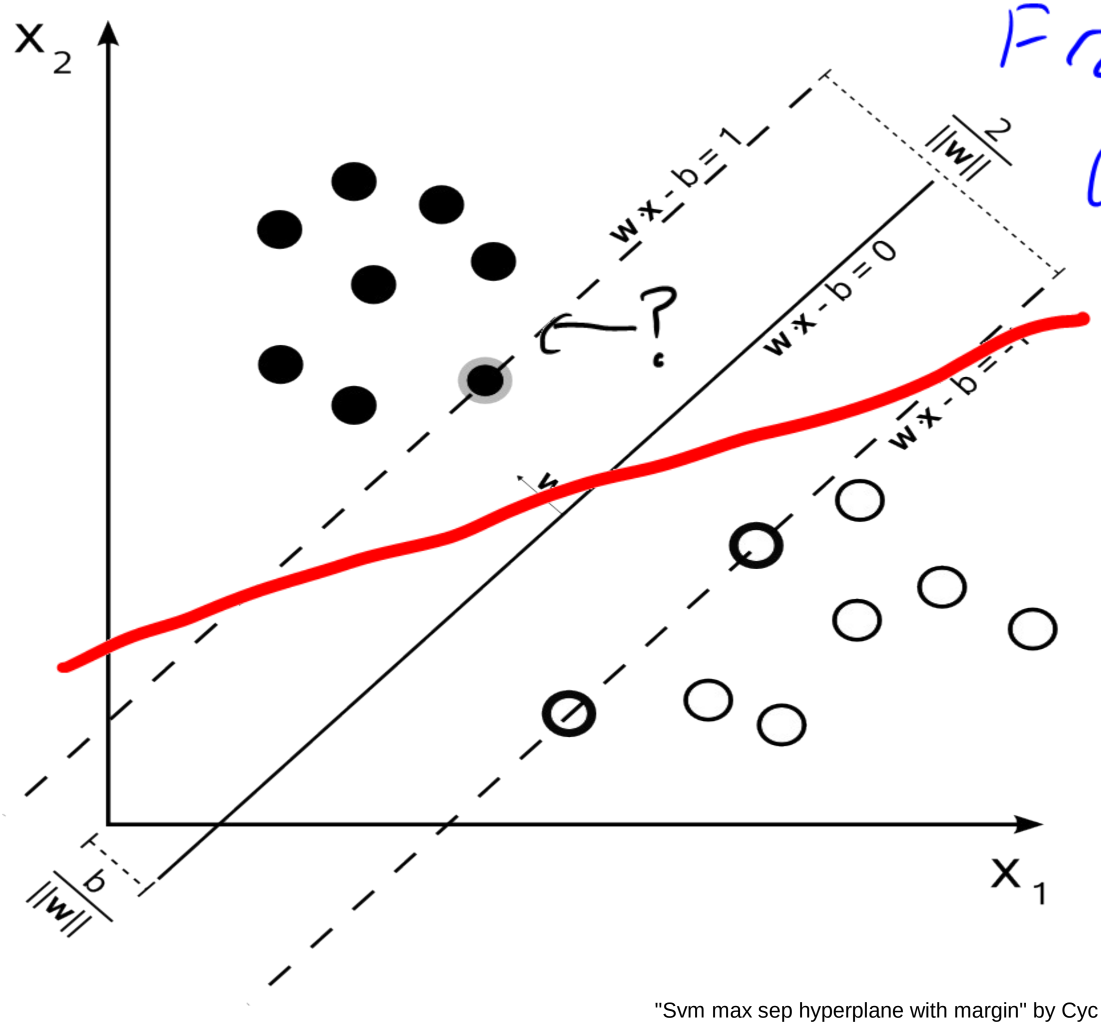
$z_1 = a_1 x_1 + b_1 y_1 + c_1 z_1 + d_1 w_1$

Pick $a, b, c, d,$ to maximize variance

What is Linear Support Vector Machine (SVM)?

- Maximum margin classifier
  - Computes a linear “decision boundary” that splits the data into two regions.
  - Allows one to predict a classification of a point based upon which side of the decision boundary it lay on.

SVM



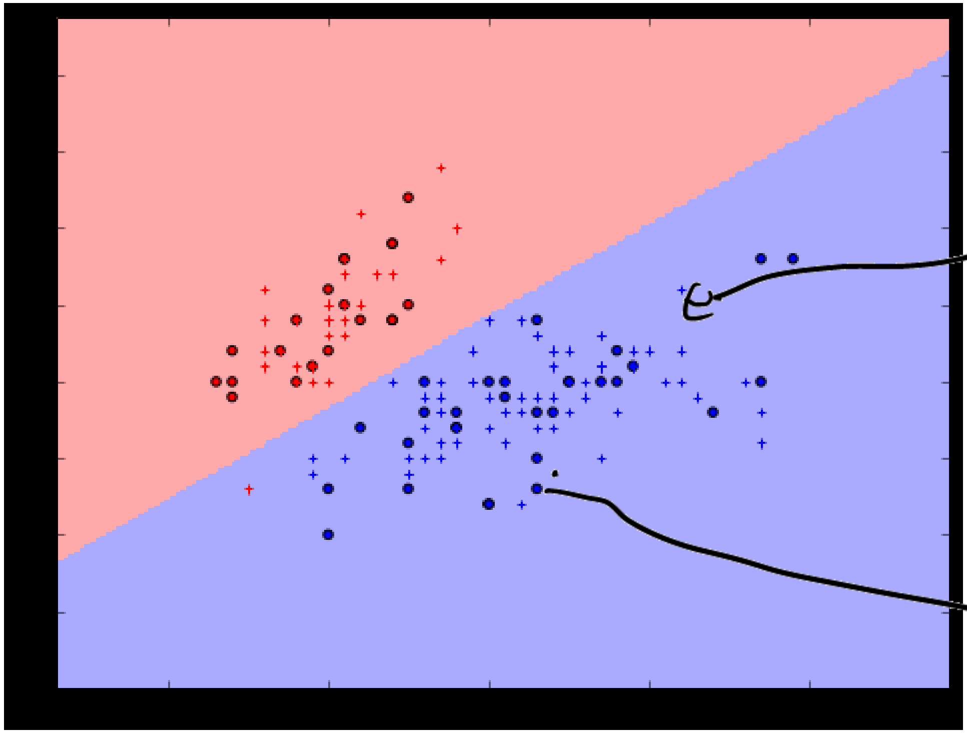
x_2
x_1
$\mathbf{w}\cdot\mathbf{x} - b = 1$
$\mathbf{w}\cdot\mathbf{x} - b = 0$
$\mathbf{w}\cdot\mathbf{x} - b = -1$
$\frac{2}{\|\mathbf{w}\|}$
$\frac{b}{\|\mathbf{w}\|}$
$\mathbf{w}$
?

as Far as Possible
From the
closest
Points

maximum
margin

"Svm max sep hyperplane with margin" by Cyc - Own work. Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png#mediaviewer/File:Svm_max_sep_hyperplane_with_margin.png

SVM



training data "+"

training data "o"

But wait! Training vs. testing!

Testing data

computing errors

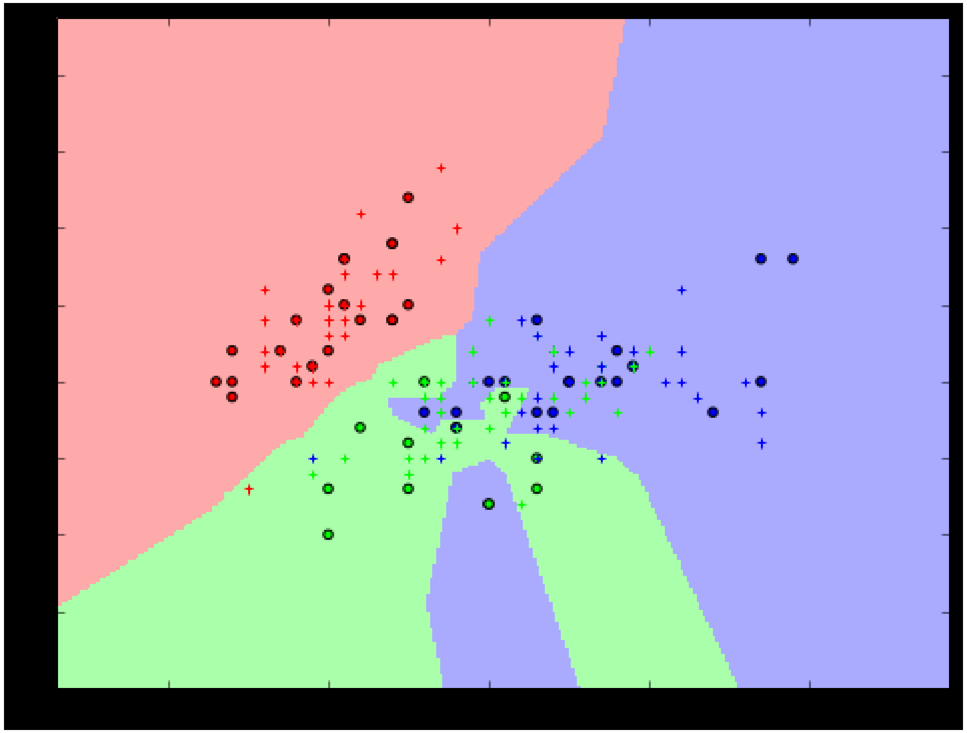
Train data

computing your model

What is K-NN?

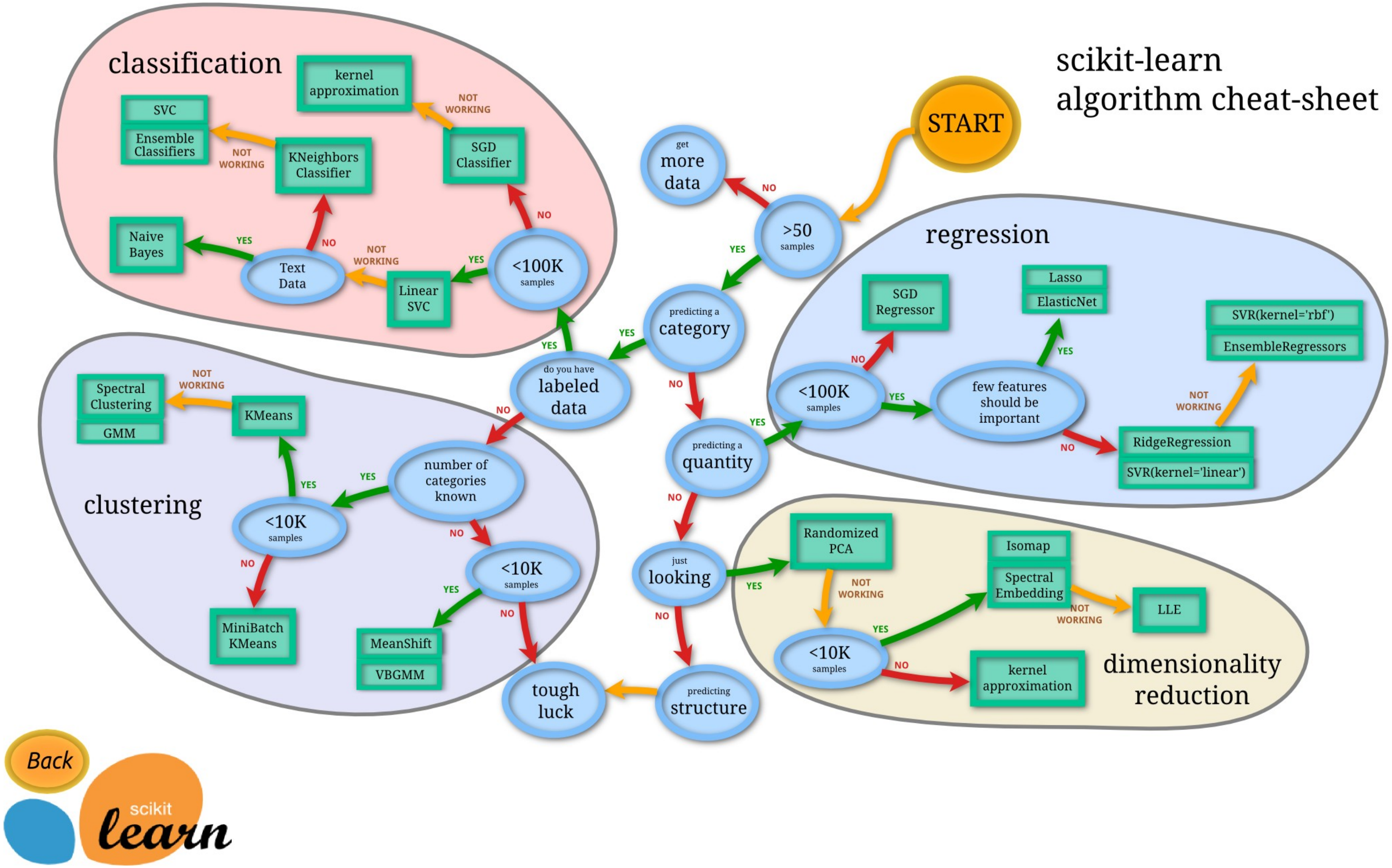
- K-nearest neighbors
- Another common classification algorithm
  - Perhaps the most common

K-NN



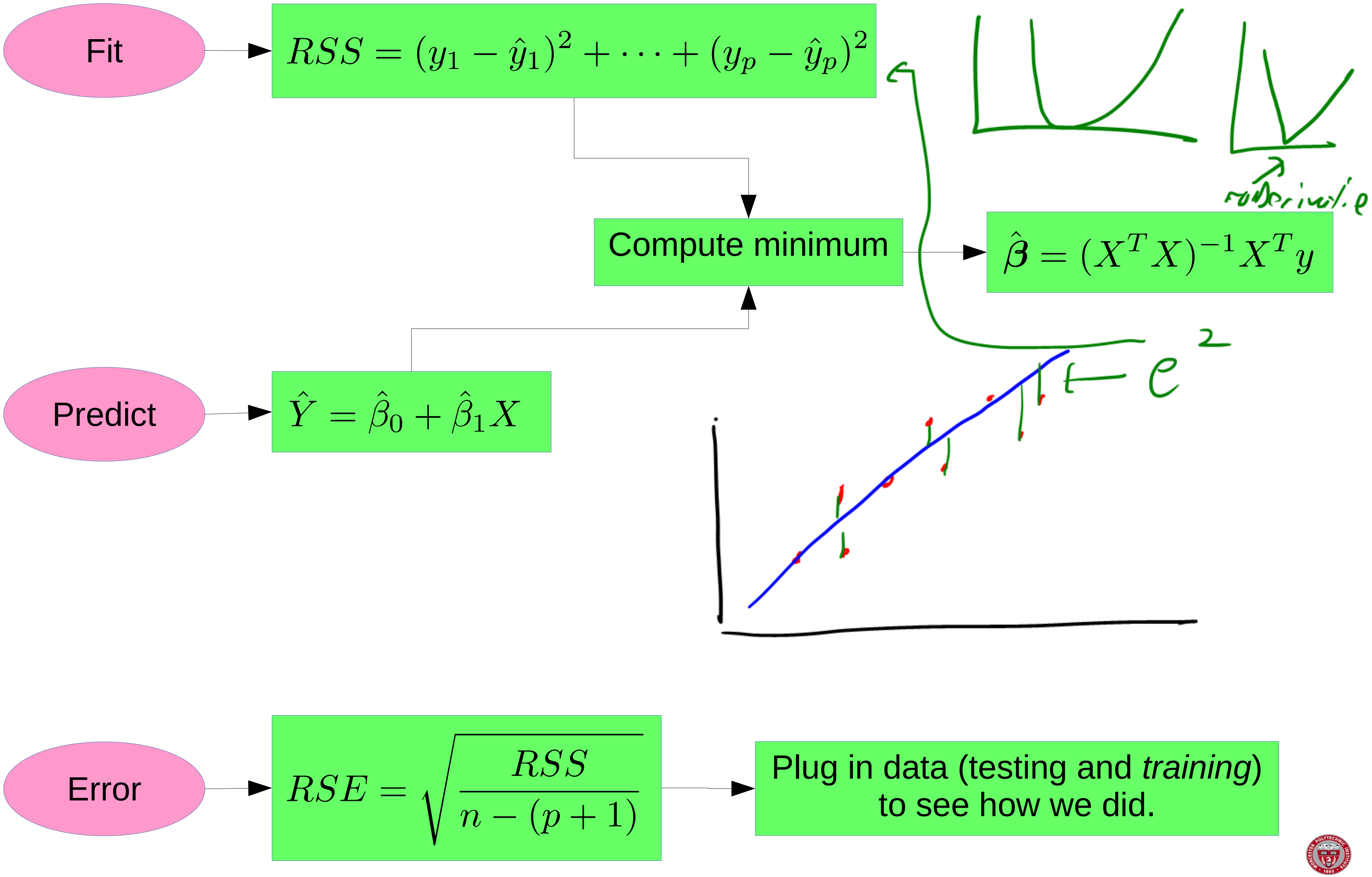
New Material!

scikit-learn



http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Linear Regression flow chart



Fit

$$RSS = (y_1 - \hat{y}_1)^2 + \cdots + (y_p - \hat{y}_p)^2$$

Compute minimum

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T y$$

Predict

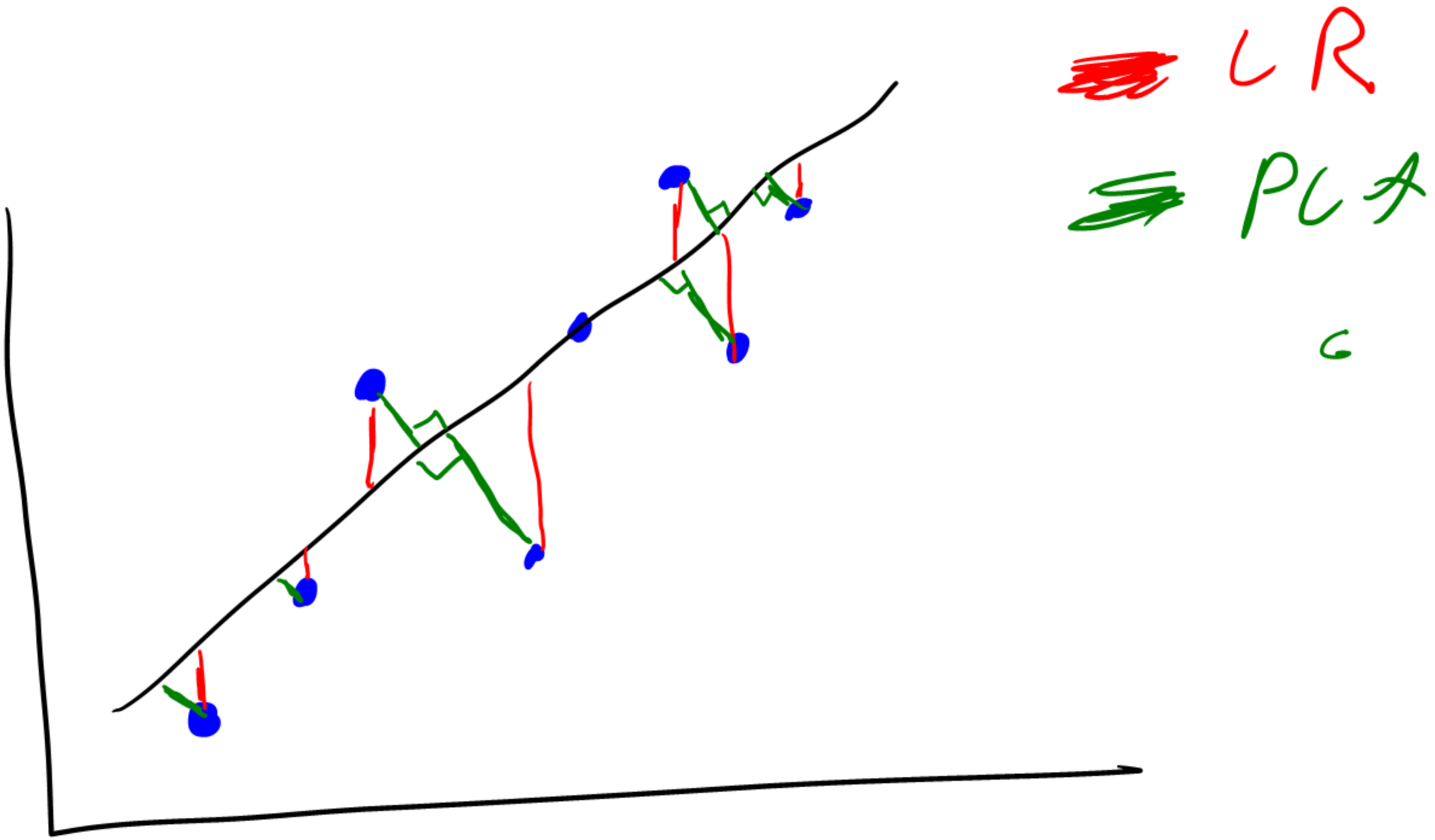
$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

Error

$$RSE = \sqrt{\frac{RSS}{n - (p+1)}}$$

Plug in data (testing and *training*) to see how we did.

Relationship between Linear Regression and PCA...



Multiple-linear regression

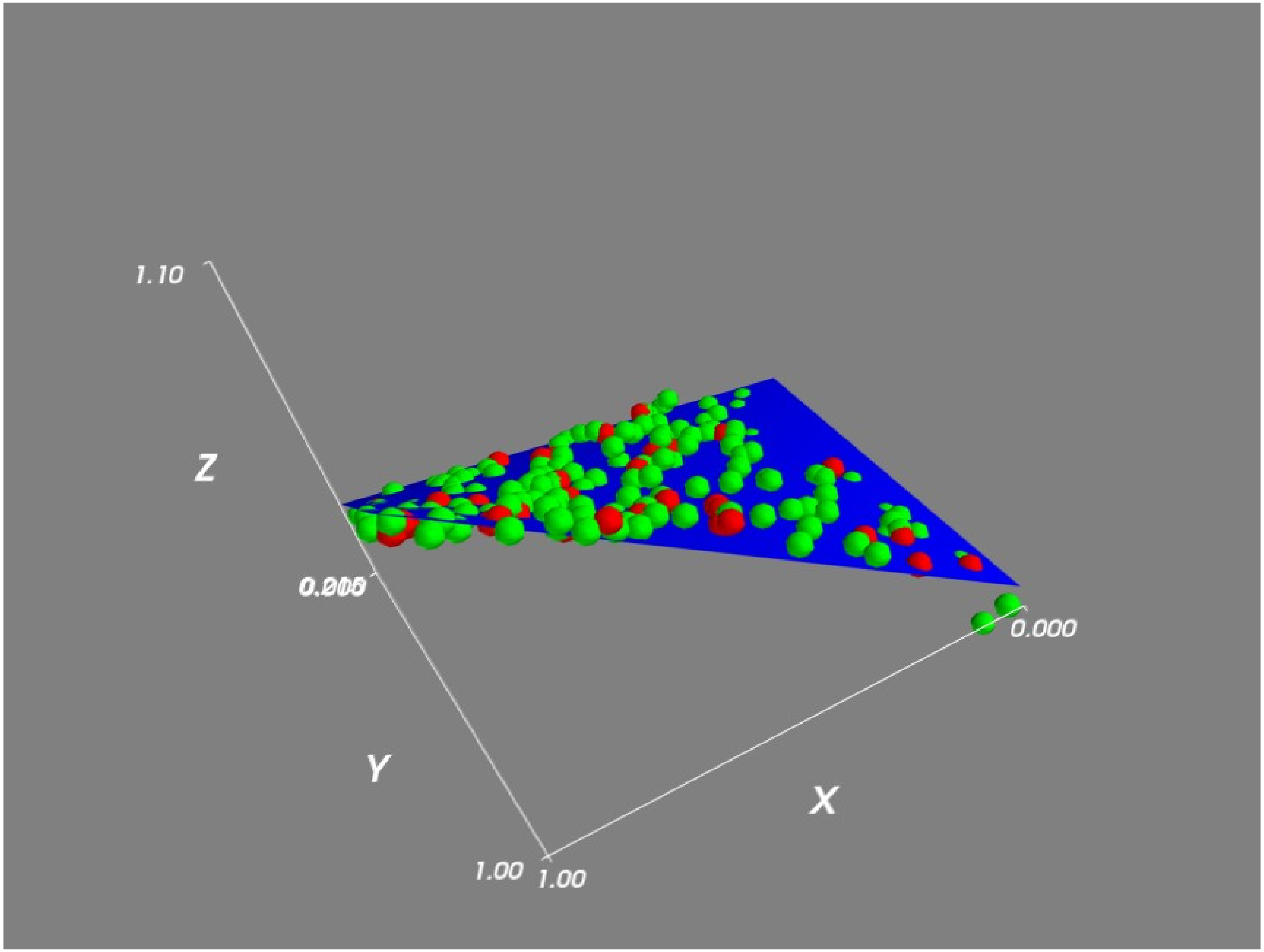
$$\overset{\text{Gas mileage}}{\overset{\downarrow}{Y}} = \beta_0 + \beta_1 \overset{\text{weight}}{\overset{\downarrow}{X_1}} + \cdots + \beta_k \overset{\text{engine si, etc}}{\overset{\downarrow}{X_k}}$$

$$y_i = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_1 x_{k,i} \quad \hat{y}_0 = \beta_0 + \beta_1 x_{1,0} + \cdots + \beta_1 x_{k,0}$$

“Non-linear” regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_{\overset{2}{1}} + \beta_3 X_1 X_2$$

See it in Python

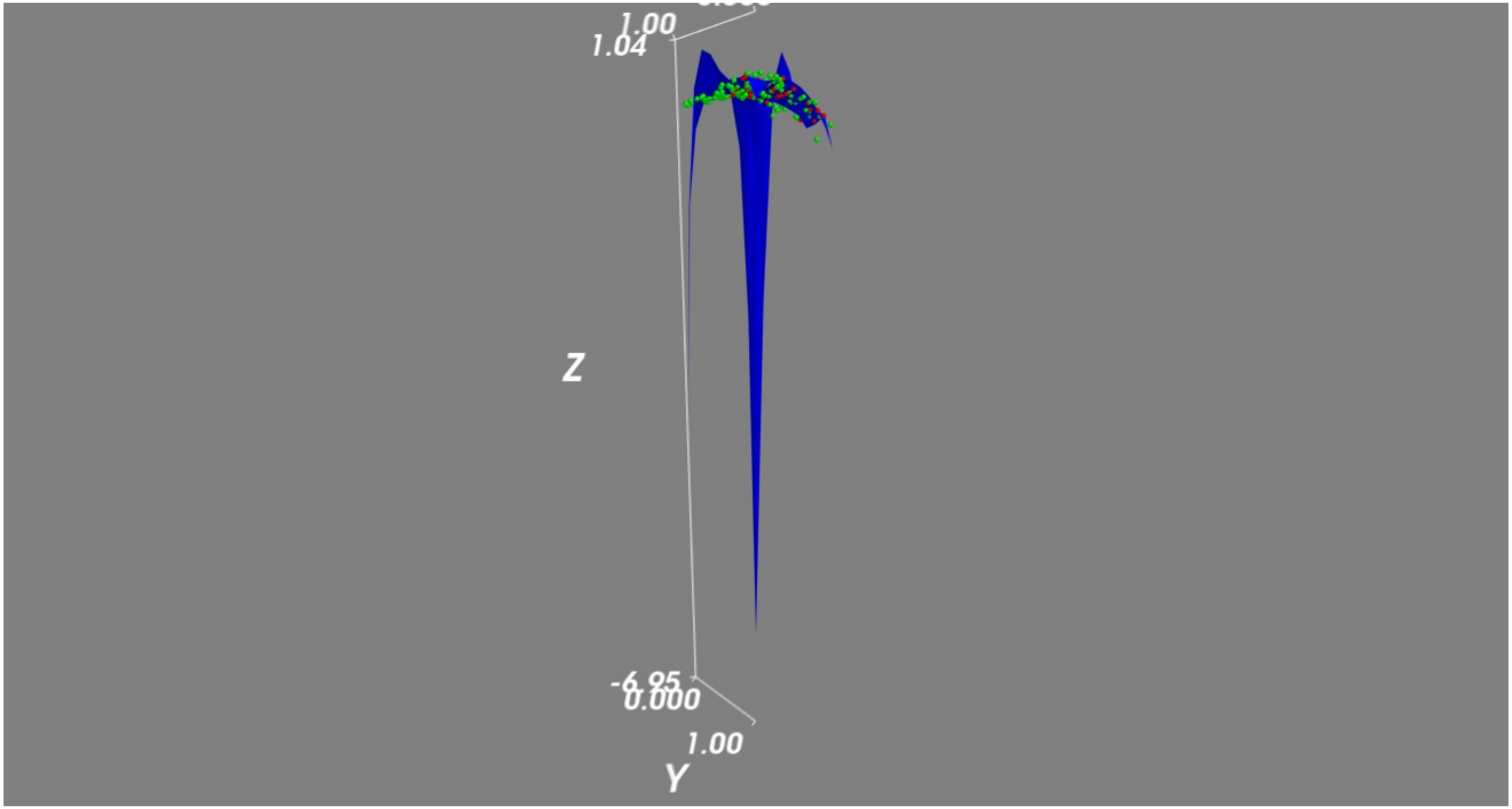


Too much of a good thing...

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1 + \beta_3 X_1 X_2$$

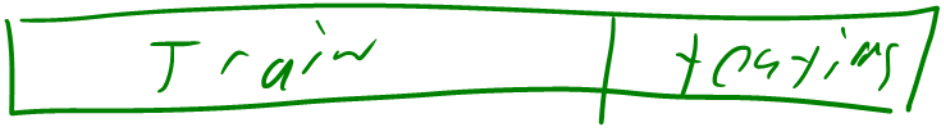
$$+ \beta_4 x_1^2 x_2^2 \quad + \beta_5 x_1^3 x_2^7 \quad + \beta_6 \cos(x_1) e^{x_2}$$

See it in Python



Cross Validation

- Validation set



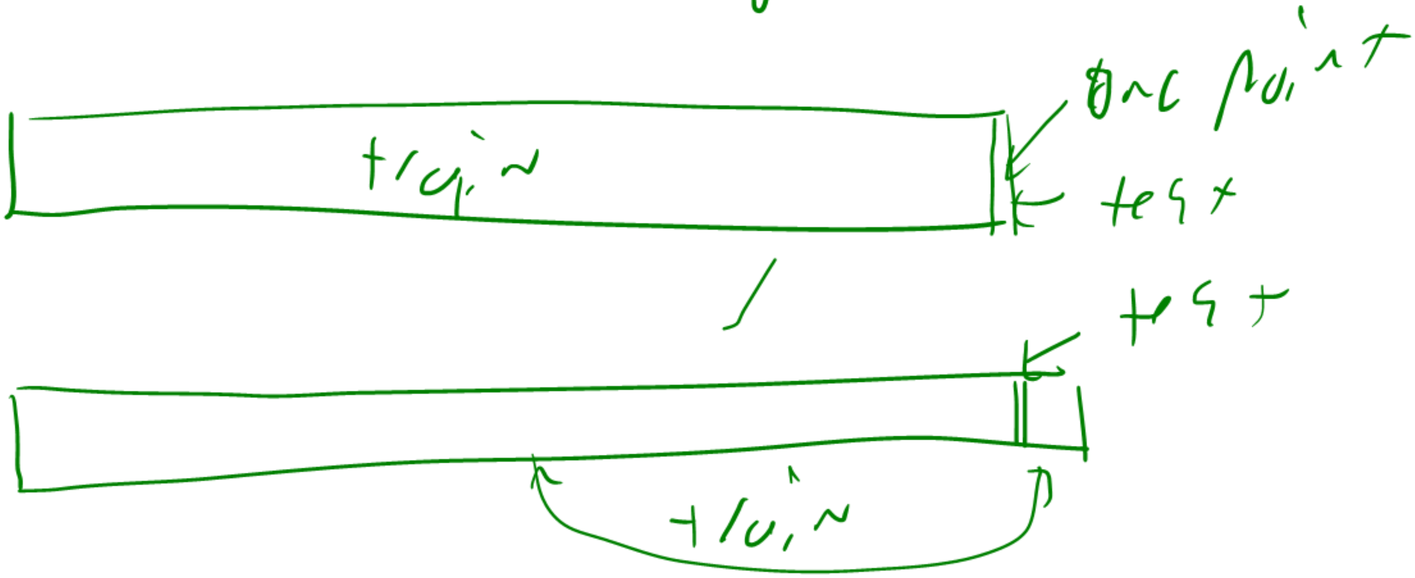
- K-fold
- Leave-one-out cross validation (LOOCV)

4-Fold crossvalidation

train	train	train	Test
train	train	test	train
train	test	train	train
Test	train	train	train

L O O C V

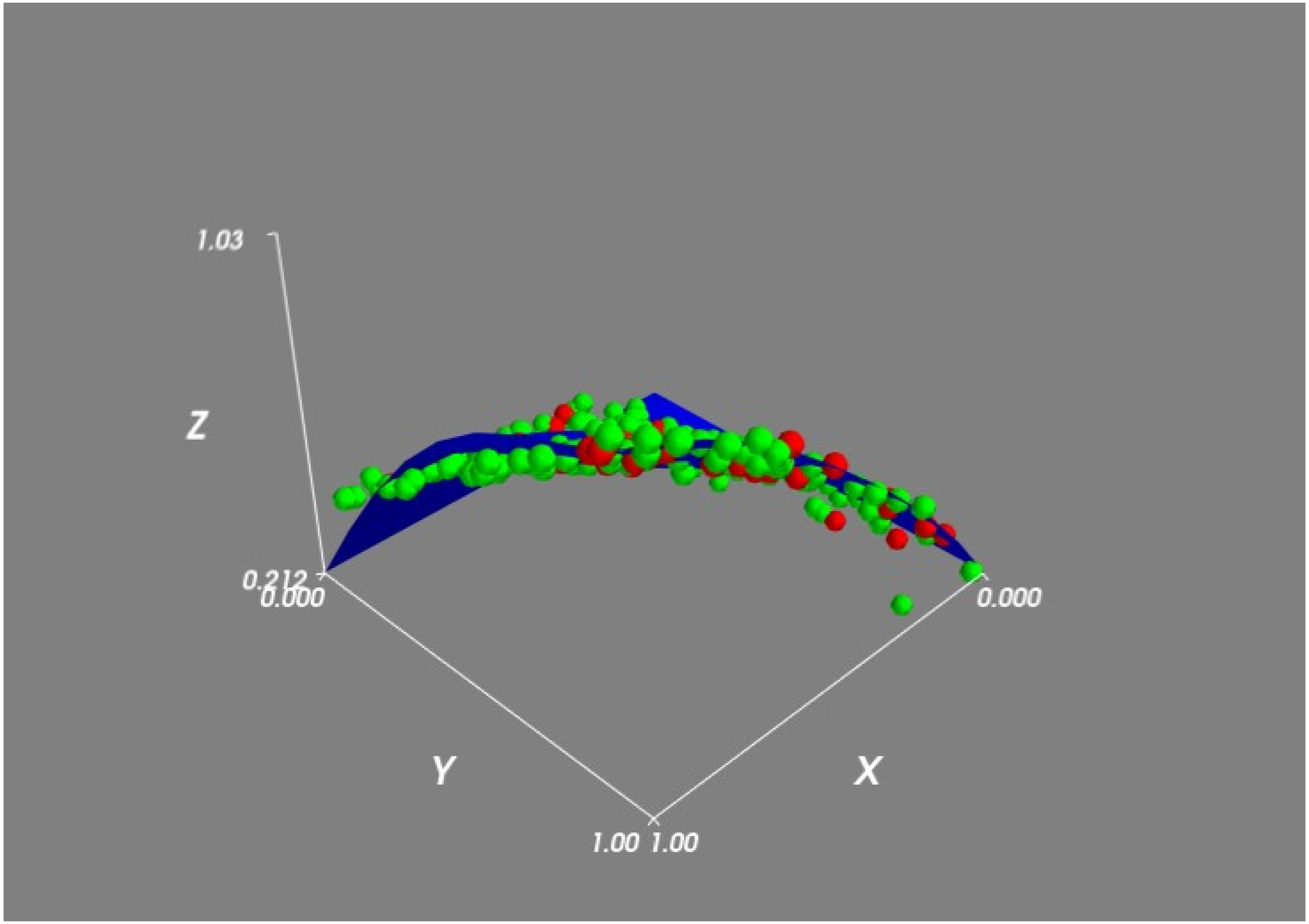
Leave One Out Cross Validation



Feature selection

- Can someone describe:
  - Best-subset selection
  - Forward stepwise selection
  - Backward stepwise selection
- Recursive Feature Elimination (RFE) is what we will use today: http://axon.cs.byu.edu/Dan/778/papers/Feature%20Selection/guyon*.pdf
  - It would take us too far astray to talk about the details of this algorithm, but it is a close cousin of backward selection.
  - Steps
    - 1. Train the classifier.
    - 2. Compute the ranking criterion for all features.
    - 3. Remove the feature with smallest ranking criterion.

See it in Python

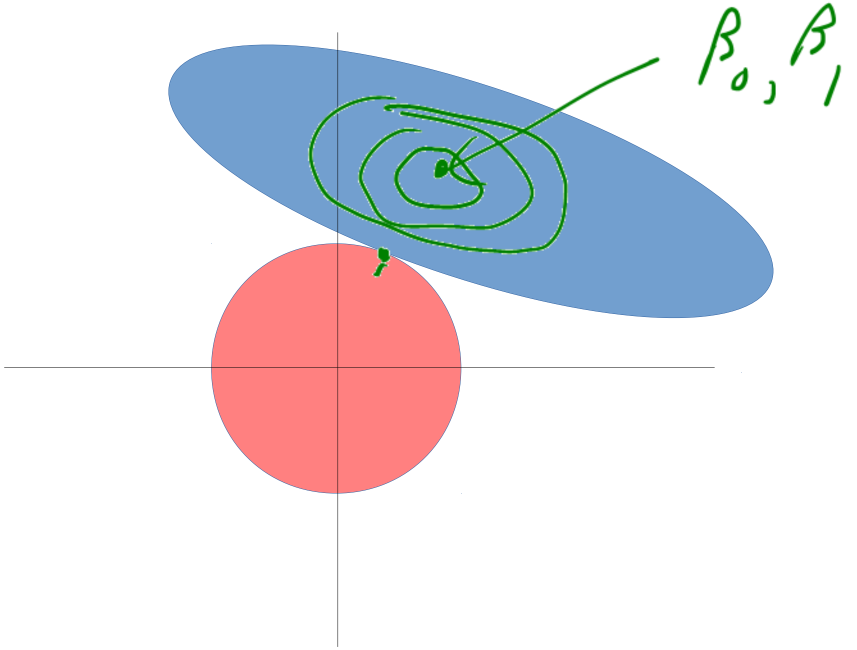


Ridge Regression

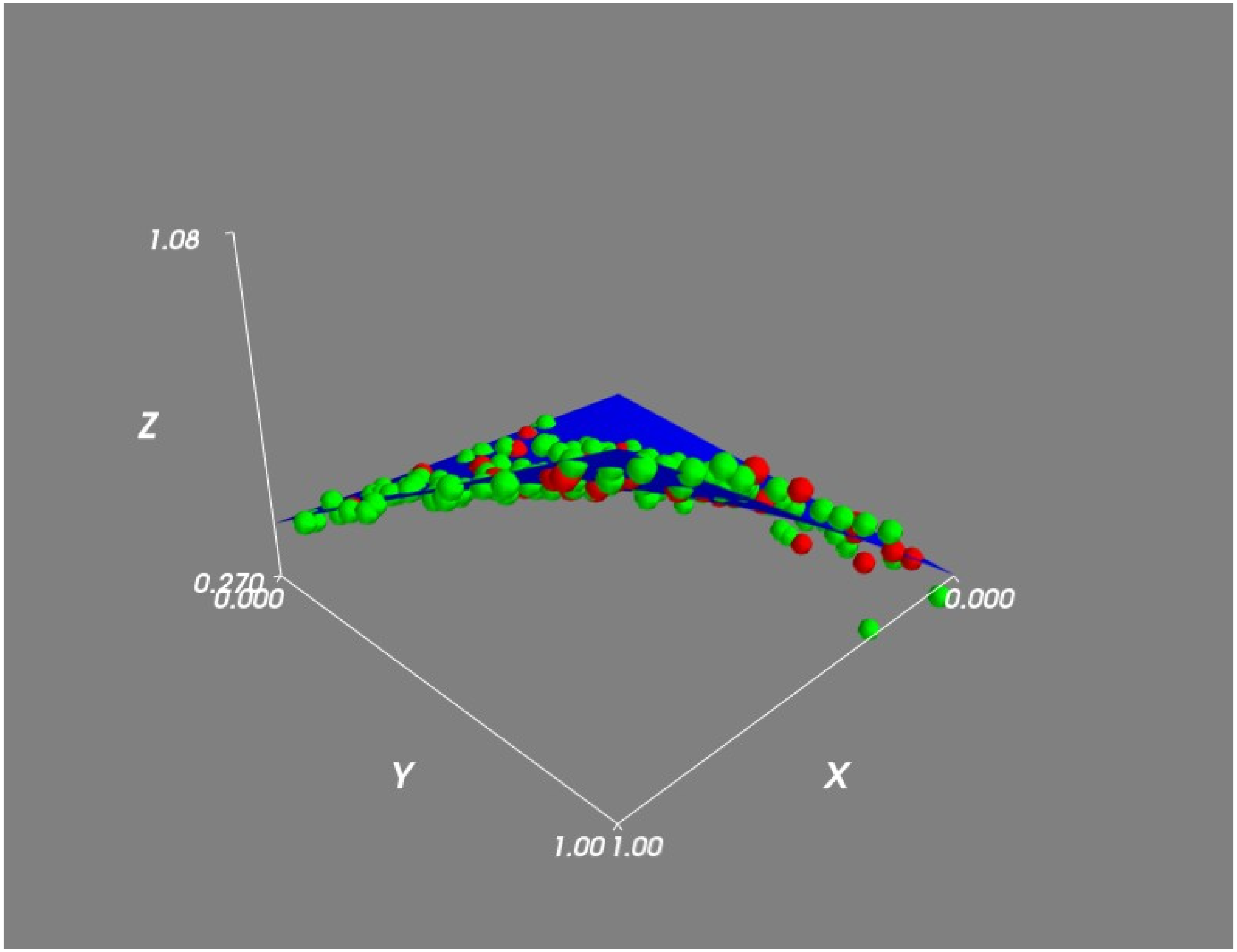
$$\sum_{i=1}^{n}(y_i-\beta_0-\beta_1x_1)^2+\lambda(\beta_0^2+\beta_1^2)=RSS+\lambda(\beta_0^2+\beta_1^2)$$

$$\min_{\beta}\left[\sum_{i=1}^{n}(y_i-\beta_0-\beta_1x_1)^2\right]\text{s.t.}\beta_0^2+\beta_1^2\leq s$$

Budget



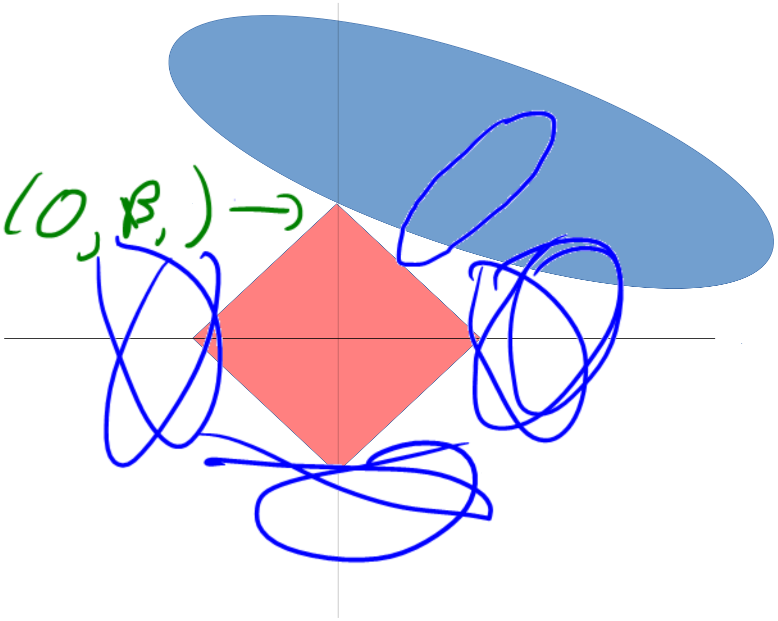
See it in Python



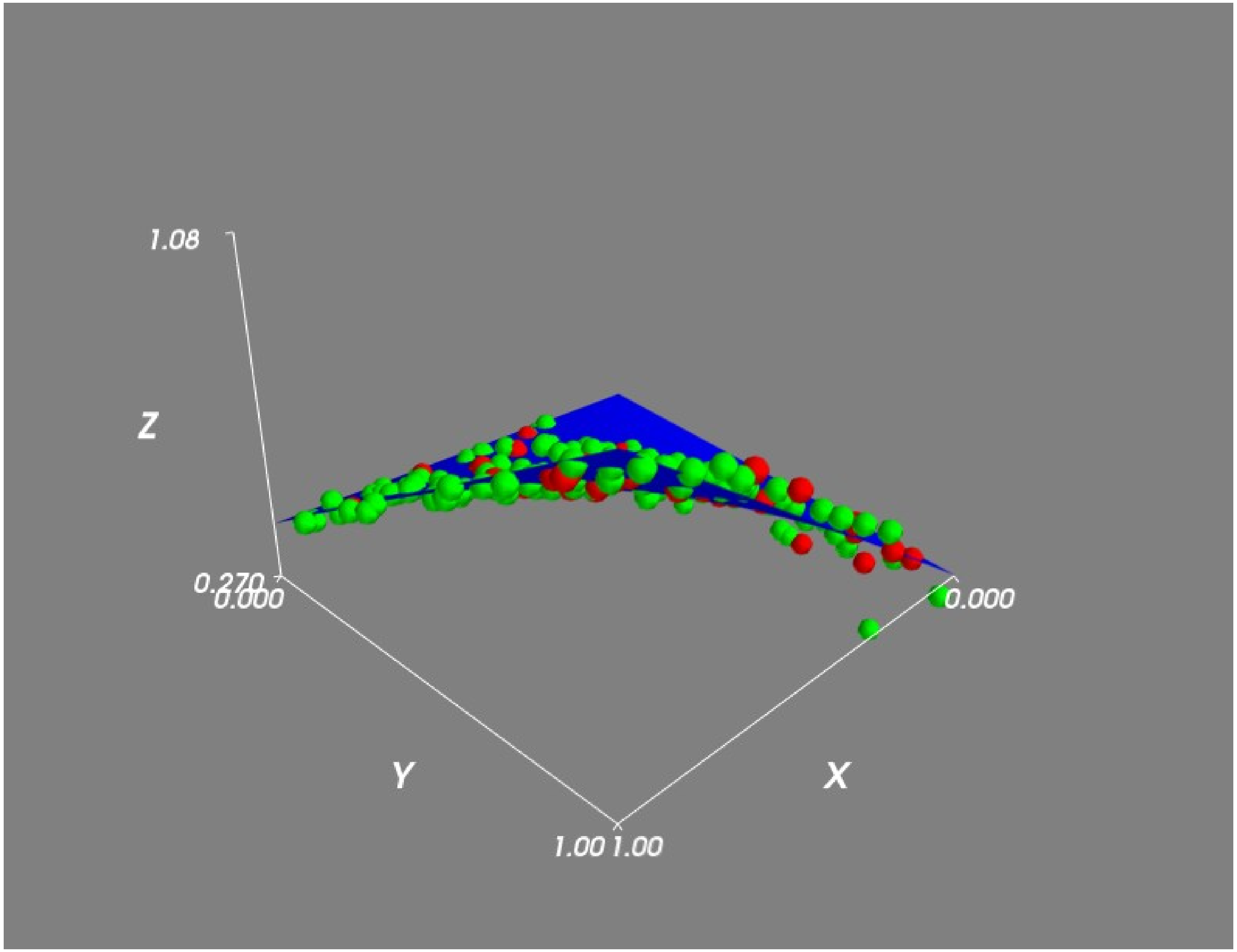
Lasso Regression

$$\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1)^2 + \lambda(|\beta_0| + |\beta_1|) = RSS + \lambda(|\beta_0| + |\beta_1|)$$

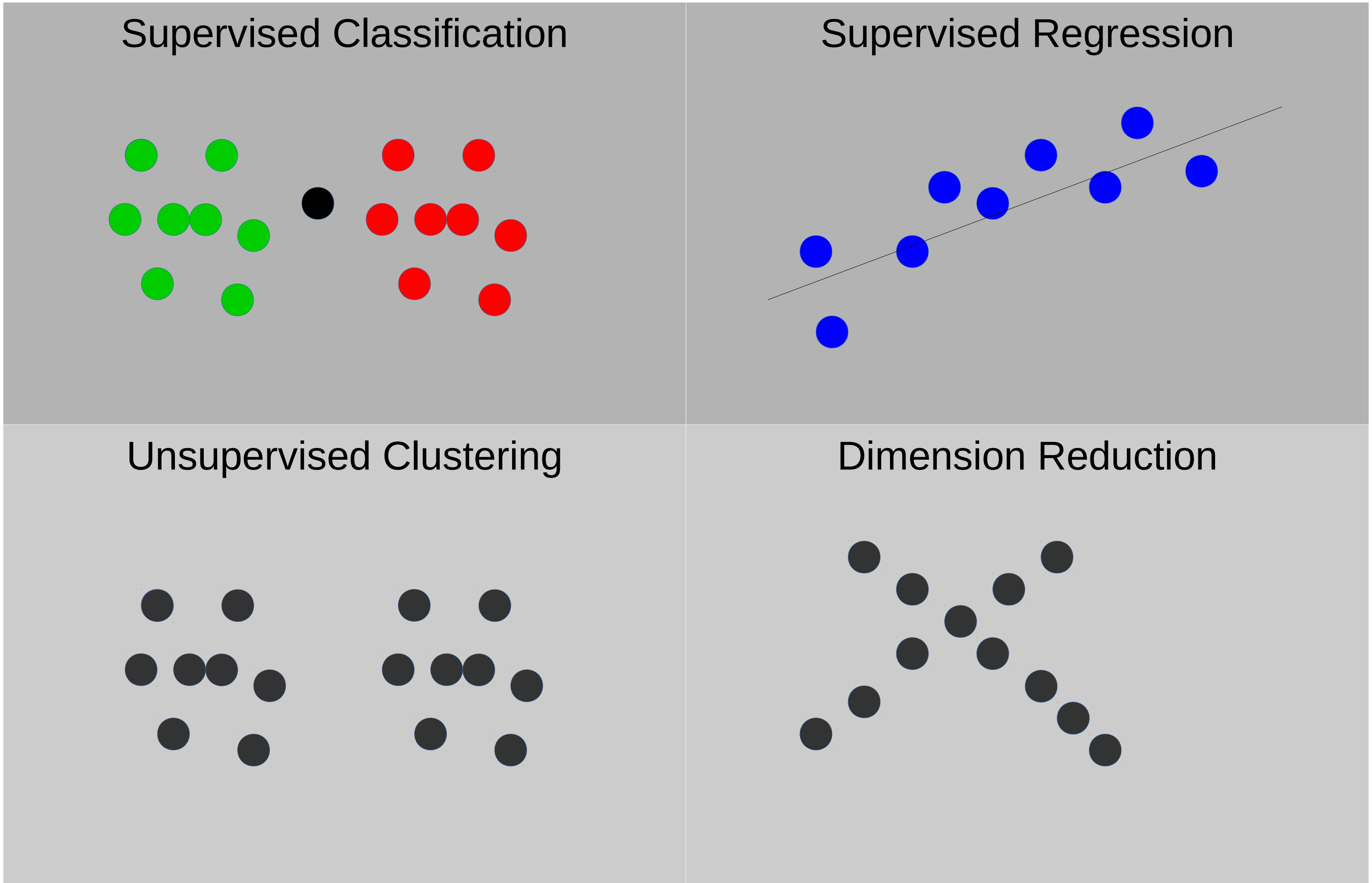
$$\min_{\beta}\left[\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_1)^2\right] \text{s.t.} |\beta_0| + |\beta_1| \leq s$$



See it in Python



Let's move on...



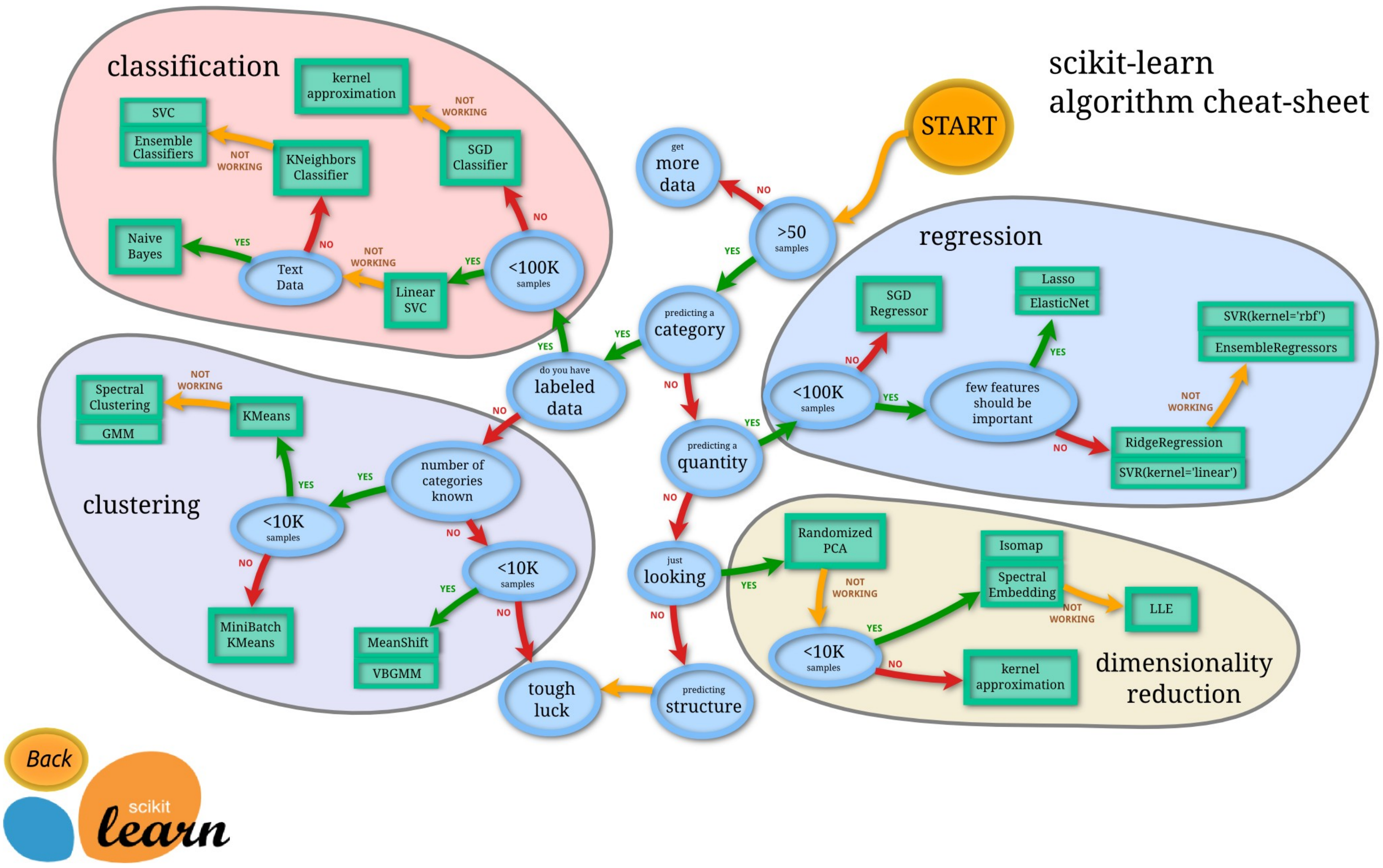
Supervised Classification

Supervised Regression

Unsupervised Clustering

Dimension Reduction

scikit-learn



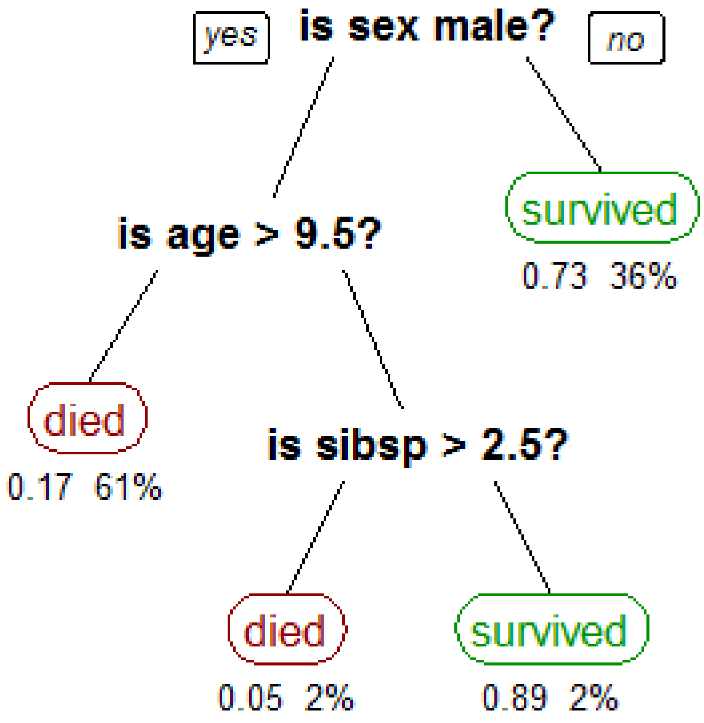
http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Decision tree

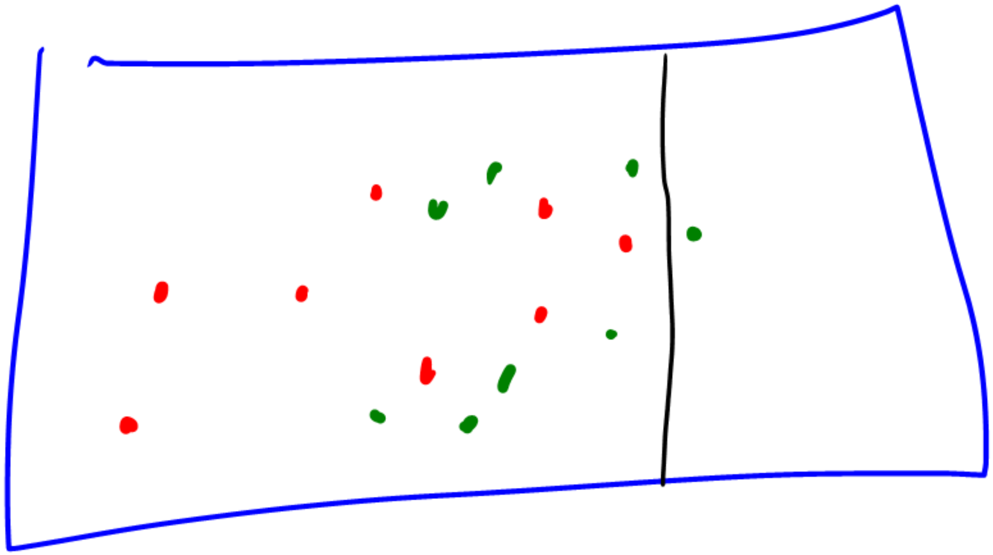
- Quite commonly used in data mining.
- Each node in the tree splits the data into (classically) two groups.
  - To make things easy (and fast) you classically perform each split on a single variable.
- Each leaf node then represents a value (or perhaps range of values) for the response based upon the input variables.

Titanic data

Probability of survival
Percentage in that leaf



"CART tree titanic survivors" by Stephen Milborrow - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:CART_tree_titanic_survivors.png#mediaviewer/File:CART_tree_titanic_survivors.png



Decision tree: Making the splits

$$\hat{p}_{mk} = Pr(Y = k | X \text{ is in region } k)$$

$$E = \max_k \hat{p}_{mk}$$

Entropy

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

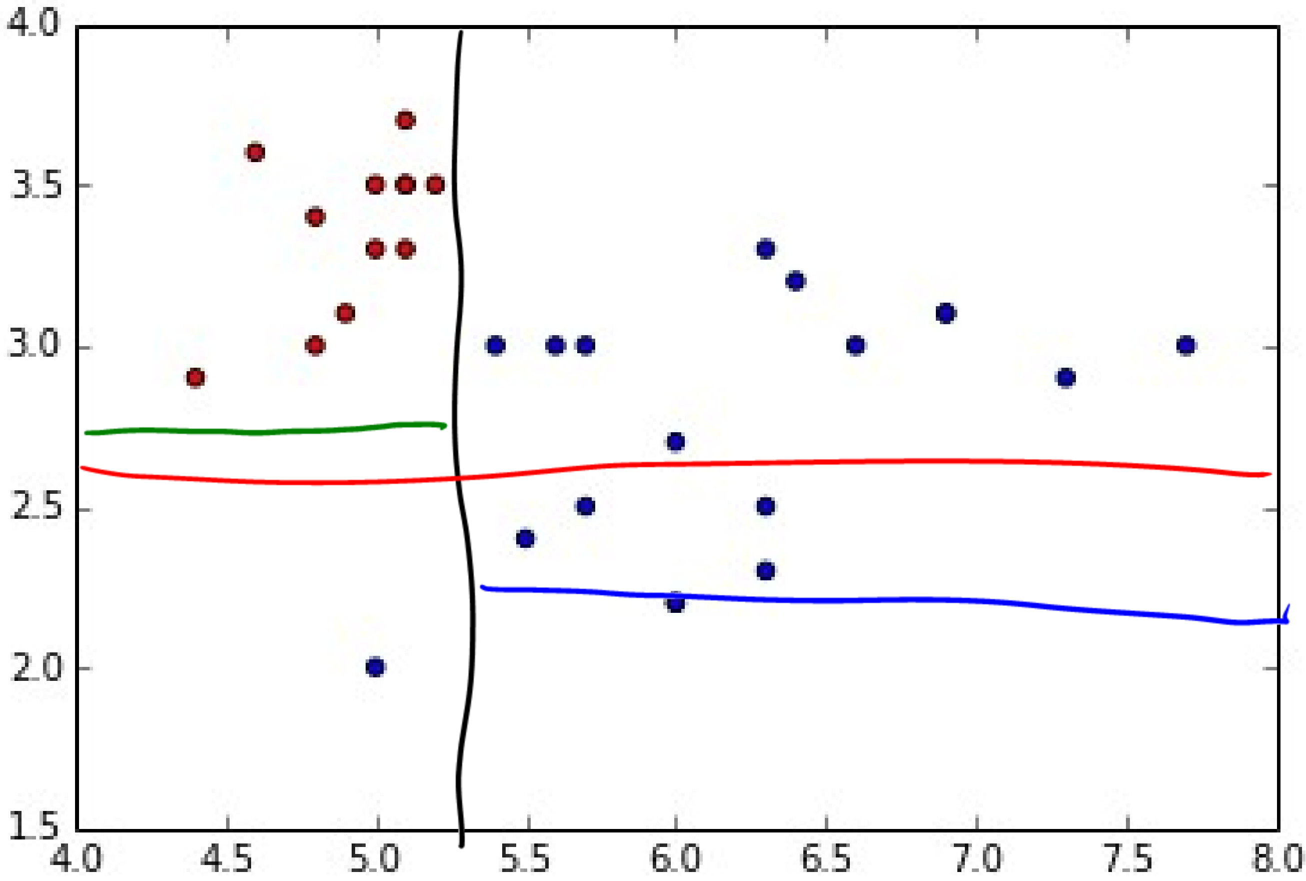
GINI

Decision Trees

1) which subset

2) which prediction

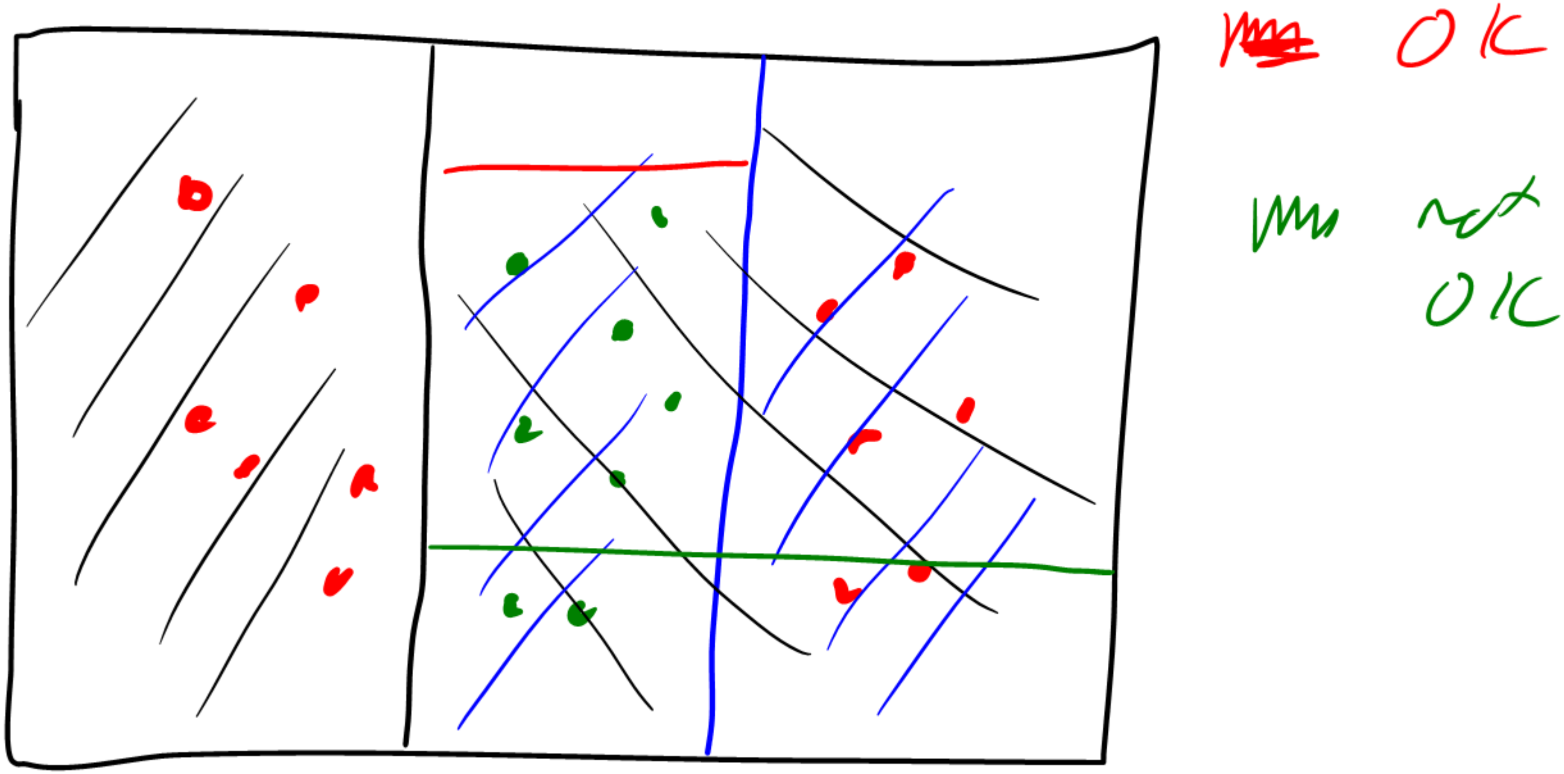
3) where in the prediction



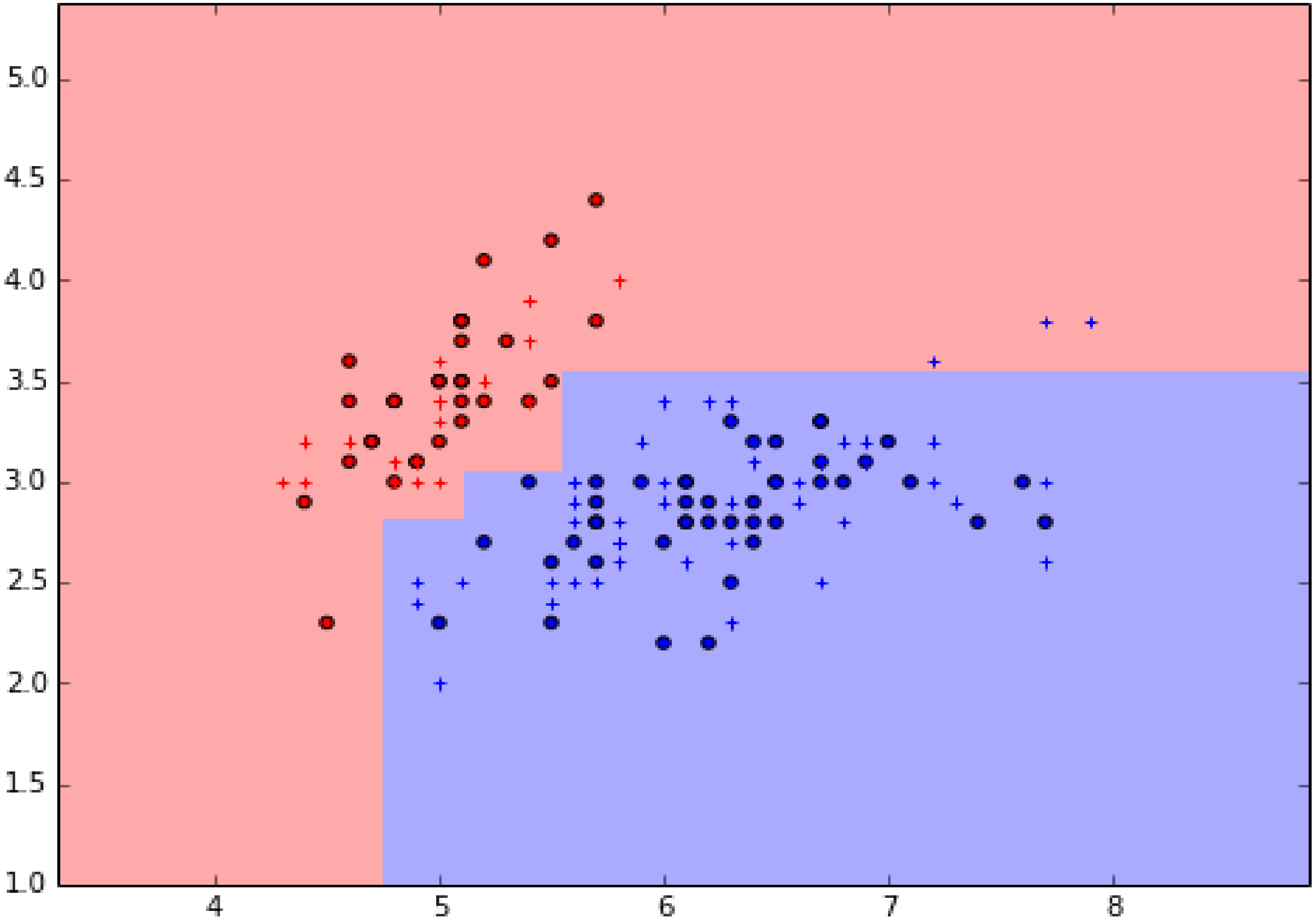
valid

valid

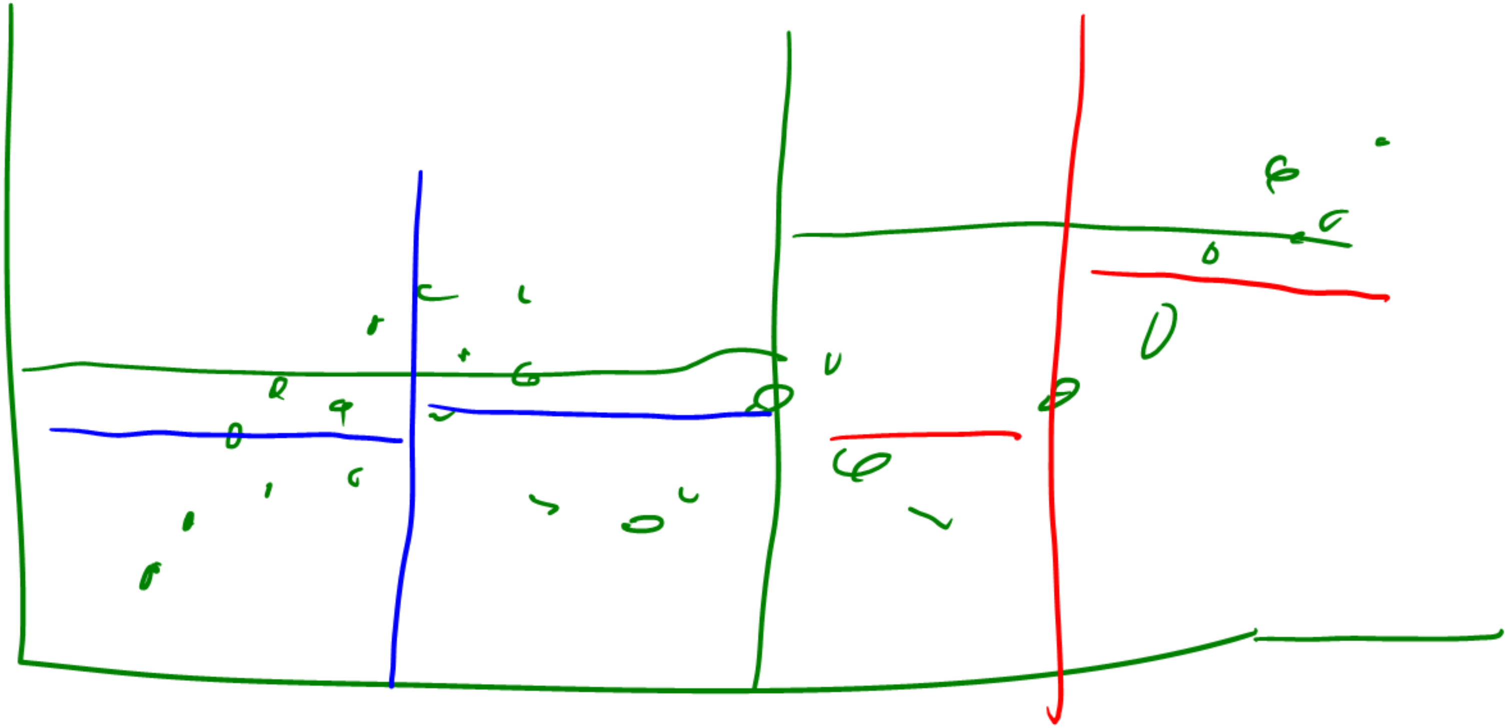
invalid


OK
not OK

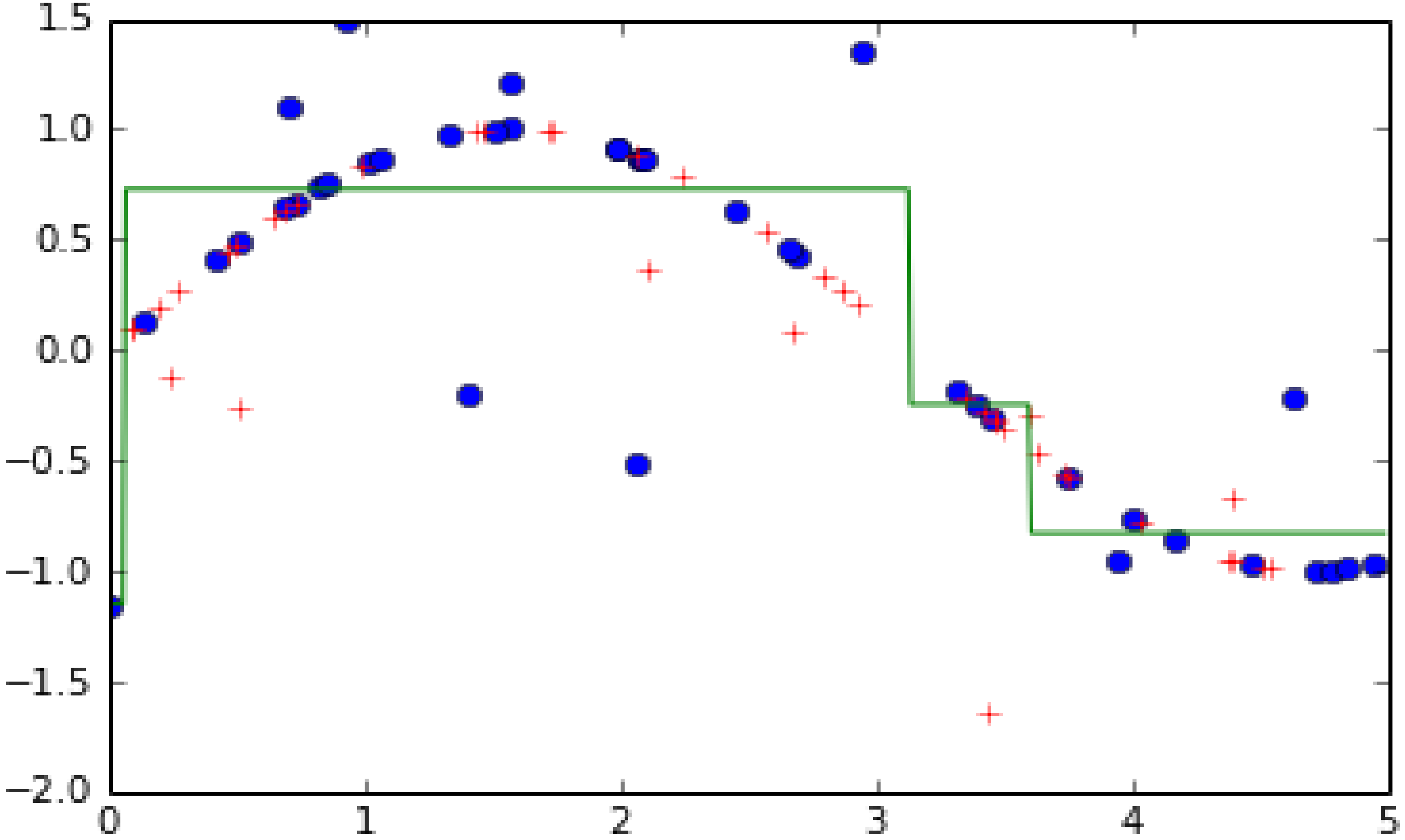
See it in Python



Regression trees



See it in Python



Ensemble Learning: Random Forest

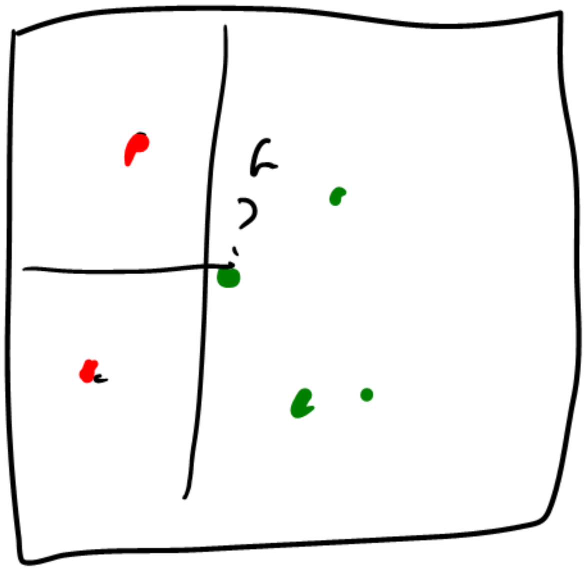
- As you can tell by the name, this idea revolves around having many trees.

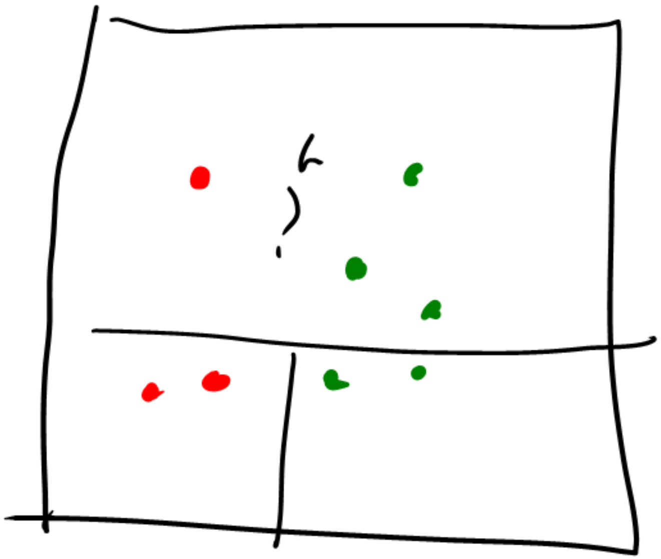
"Kellerwald 004" by Willow - Own work. Licensed under Creative Commons Attribution-Share Alike 2.5 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Kellerwald_004.jpg#mediaviewer/File:Kellerwald_004.jpg

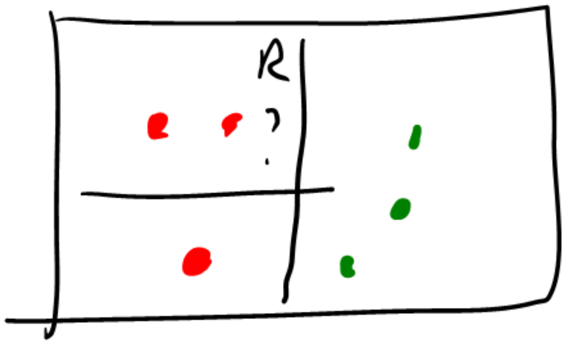
Tree bagging: Bootstrap aggregation

- Bootstrapping is one of my favorite algorithms in statistical learning.
- An extremely powerful idea for doing statistical learning with limited data.
- Generate many random samples of your data, with replacement, and train a tree on each...

"Kellerwald 004" by Willow - Own work. Licensed under Creative Commons Attribution-Share Alike 2.5 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Kellerwald_004.jpg#mediaviewer/File:Kellerwald_004.jpg







G, G, R,

answer

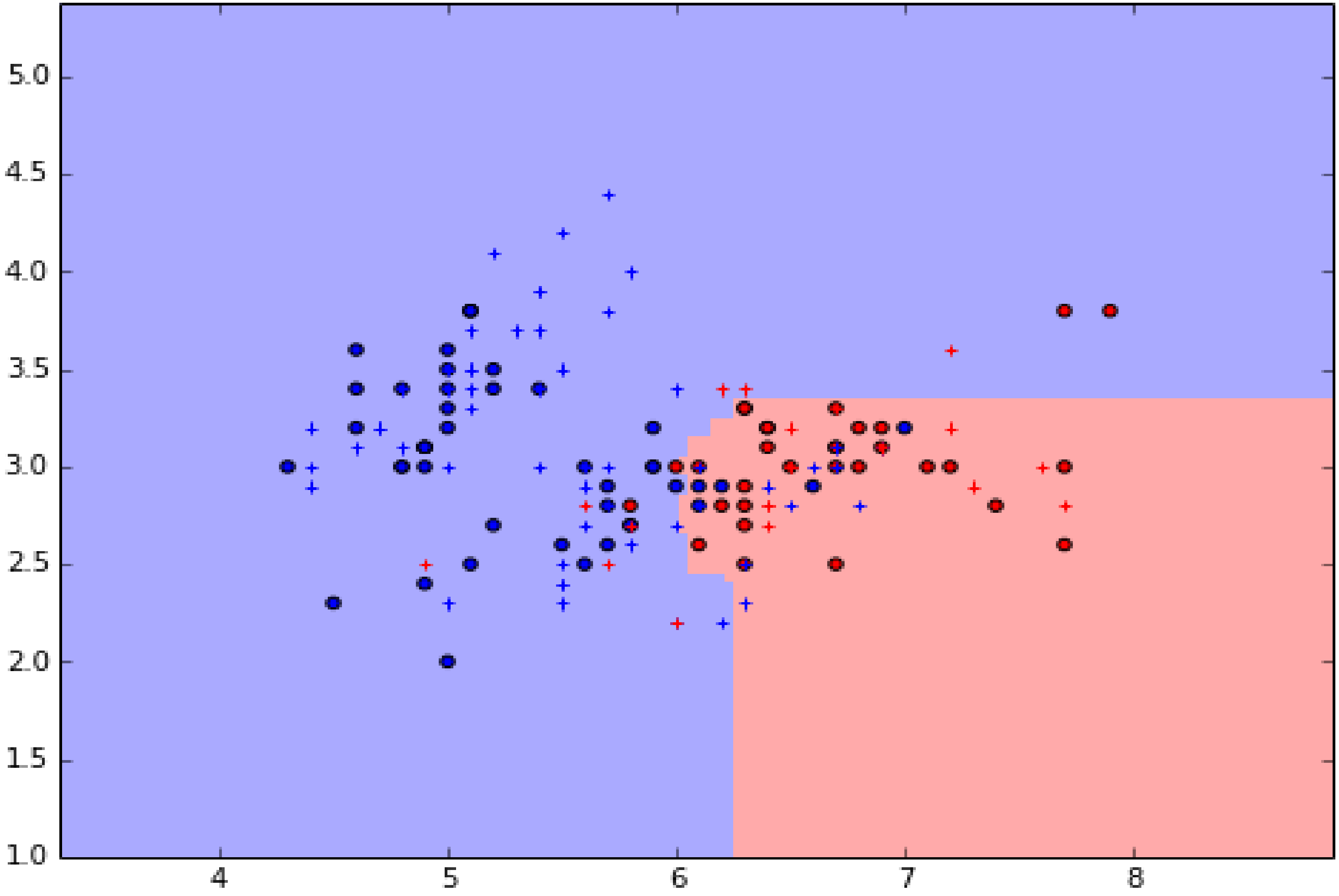
G

Random forests

- Add even more randomness by randomly selecting for each tree a subset of the features it is allowed to split on.
    - Reduces correlation between the trees!
    - Not all trees can pick the "obvious" best predictor to split on first.

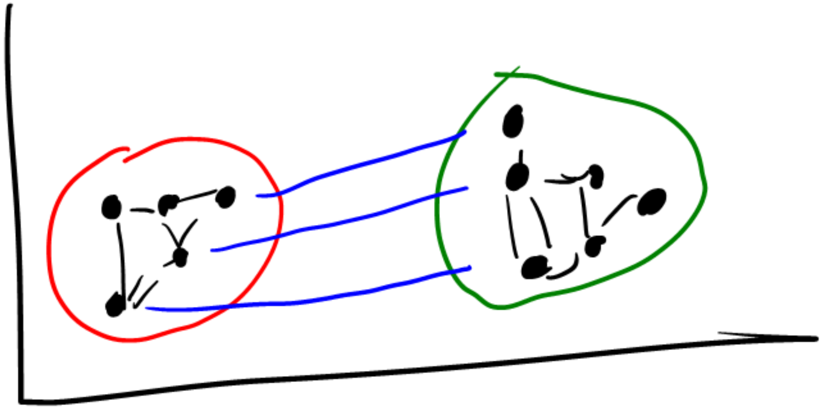
"Kellerwald 004" by Willow - Own work. Licensed under Creative Commons Attribution-Share Alike 2.5 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Kellerwald_004.jpg#mediaviewer/File:Kellerwald_004.jpg

See it in Python



K-Means

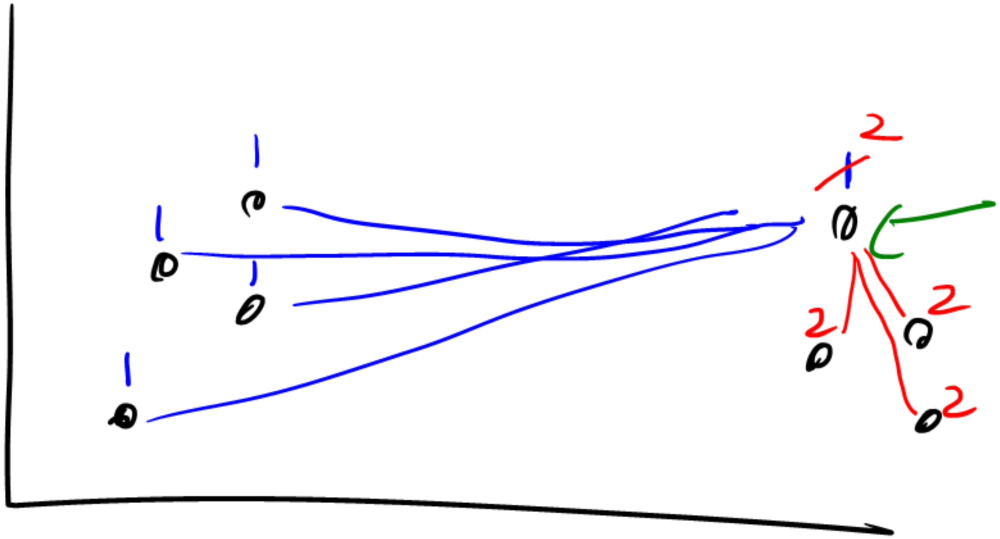
- Perhaps the single most used unsupervised classification algorithm.
- Given a number of classes k, divide the data into groups so that the distance within a group is “small” compared to the distances between groups.



K-Means

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

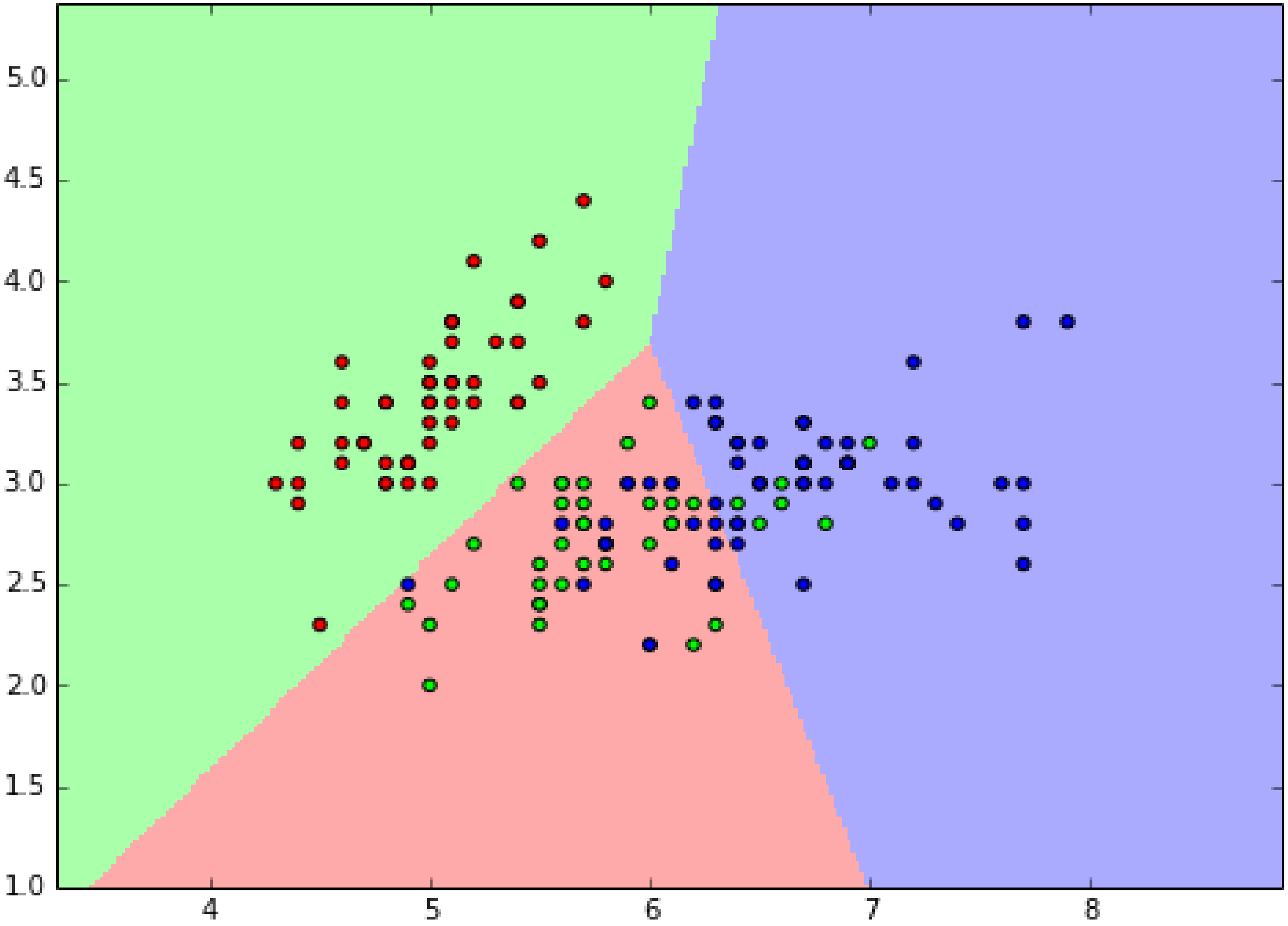
Centroid of the cluster



K-Means

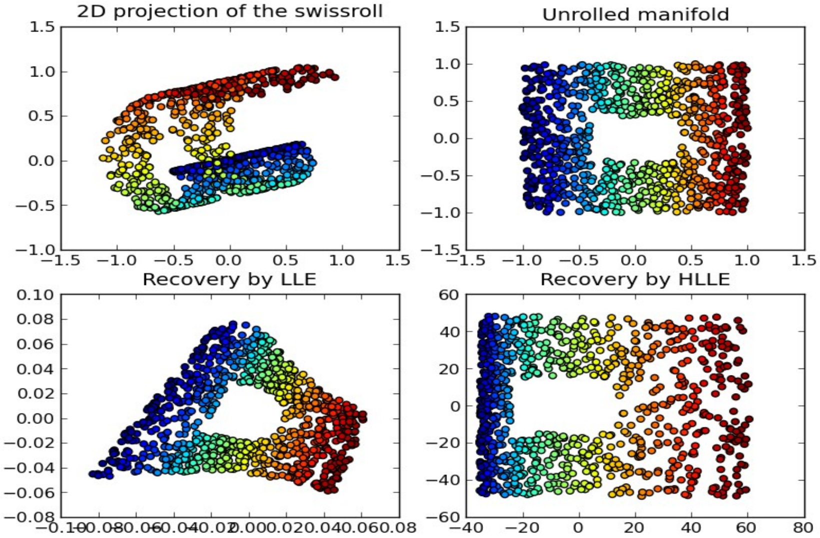
- Lloyd's iterative refinement algorithm
  - Assign each measurement to the cluster whose mean gives the least sum of distance squared (i.e. the nearest)
  - Calculate new means to be the centroids (i.e. average) of the observations in each cluster.

See it in Python



Manifold learning

- As the last item in our foray into machine learning we will dip our toes into manifold learning.
  - There are many algorithms, see Wikipedia.

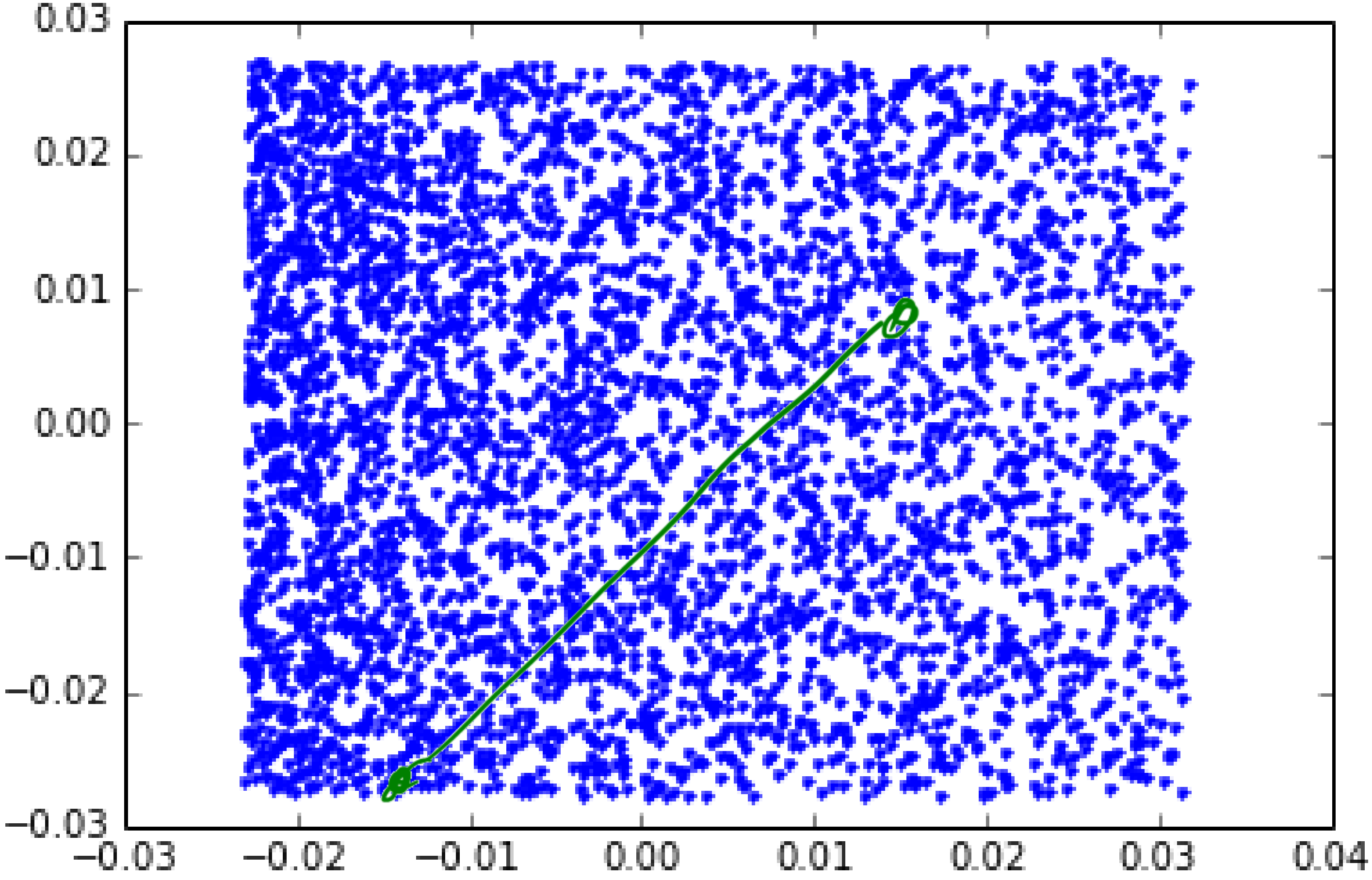


"Lle hlle swissroll" by Olivier Grisel - Generated using the Modular Data Processing toolkit and matplotlib.. Licensed under Creative Commons Attribution 3.0 via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Lle_hlle_swissroll.png#mediaviewer/File:Lle_hlle_swissroll.png

Local Tangent Space Alignment

- A very rough outline of the algorithm
  - Compute the collection of points nearest each point.
  - Compute the tangent space at each point (e.g. using PCA!)
  - Solve an optimization problem to align the tangent spaces.

See it in Python


0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.03
−0.02
−0.01
0.00
0.01
0.02
0.03
0.04