**Goals**

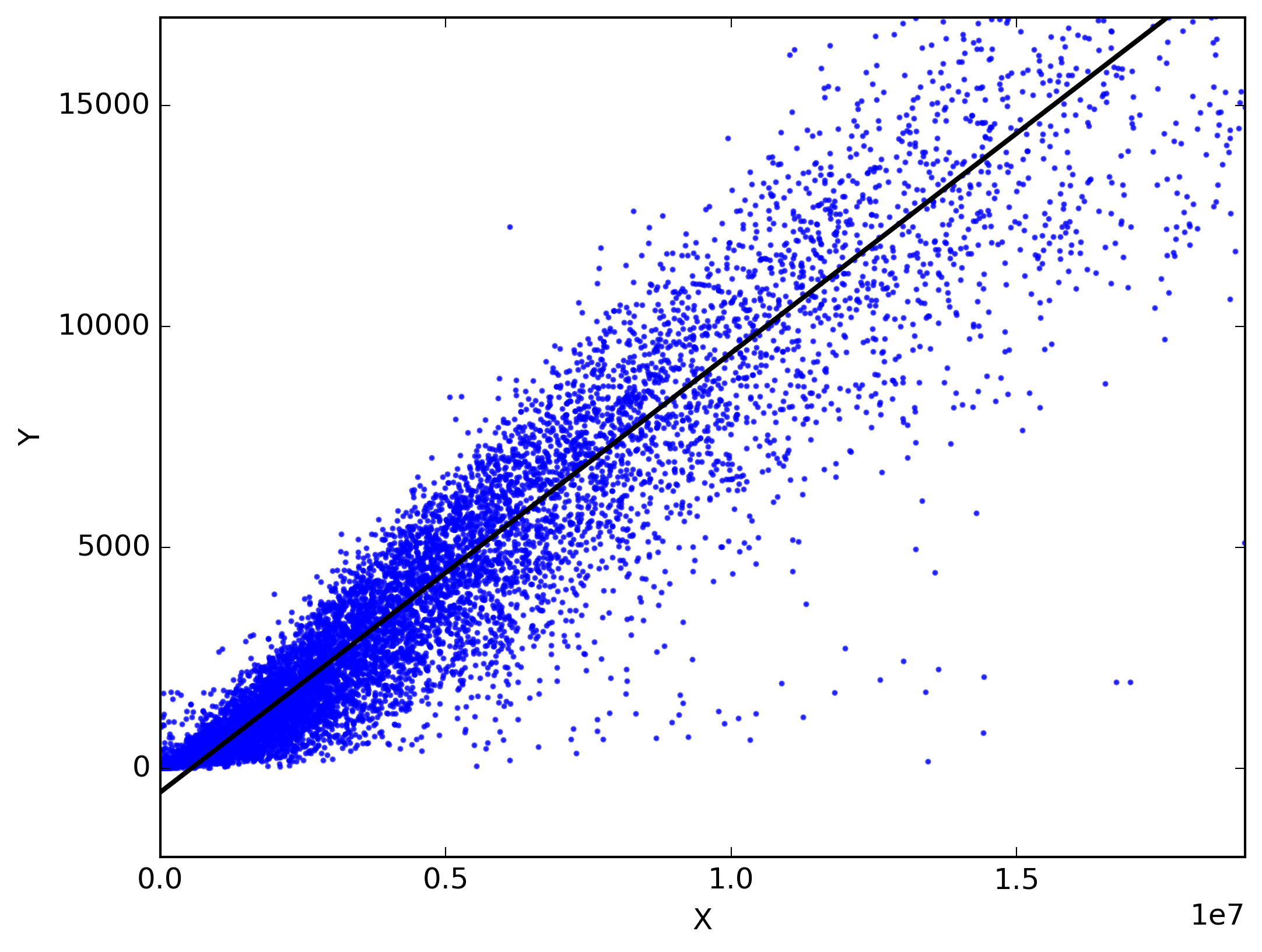
**properties of the following models:**

* **linear models :** explaining the behavior of a **dependent variable** as a function of **explanatory variables (also called independent variables)**. Generally the dependent variable is a continuous variable.

Let’s suppose **X the dependent variable** and **Y1, Y2, Y3** the **explanatory variables.** The linear model will explain the behavior of X using the following formula:

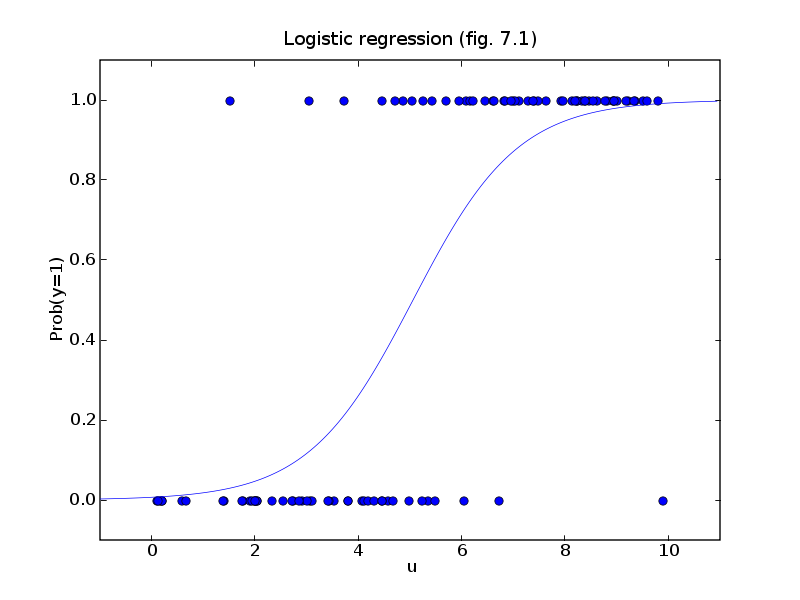
**X = β0 + β1 \* Y1 +β2 \* Y2 + β3 \* Y3**

The purpose of this model is to draw a line generalizing the behavior of the dependent variable as you can see on the graphic below.



* **logistic regression :** model used to predict the value of **a non continuous(discrete) dependent variable** based on the values of **explanatory variables**: it’s a **classification model**. The dependent variable can just have **two possible values**. As an example, with logistic regression we could try to predict someone’s age class(**Young or Old )** based on his **physical characteristics(height, weight, hair color, etc.).**

Unlike the linear regression, the logistic regression draws a S shape line (logistic function) going from 0 to 1 (cf the graph below)

Based on the **independent variables**, this function represents the **propability** of the **dependent variable** of being from one of the **two classes**. The higher this probability is, the more accuracy we have in our classification.

If we take back the **Age Class** example, we could say the function represents the probability of being **Old.** Then for each point of the train set, this probability will be computed and according to its probability we classify the point as **Old** or **Young.**

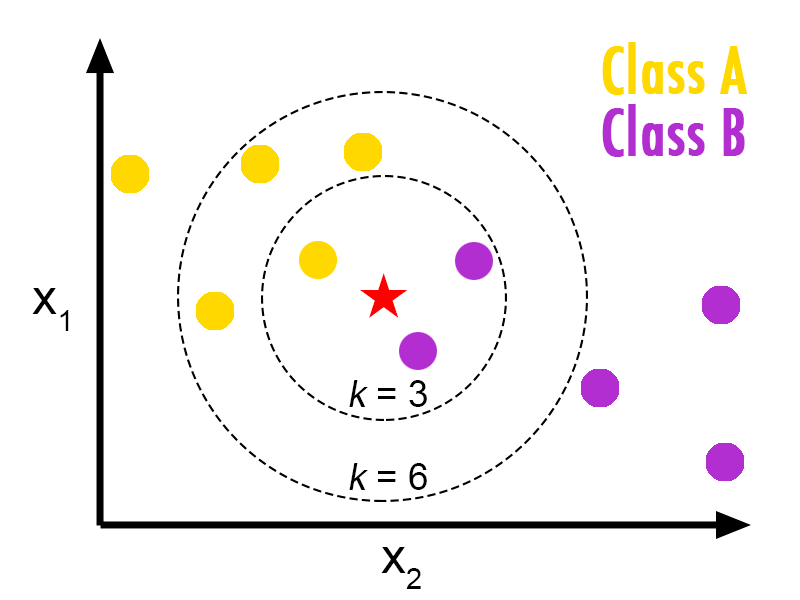
* **K-Nearest Neighbours**

A classification algorithm based on the principle of **majority vote**.

Firstly, the observations from the different classes are ploted in a graph according to their characteristics. Then the classification is made following the steps below:

* + A new observation (whose class is unknown) is plotted in the graph according to its characteristics
  + We search for the K nearest neighbours of that observation
  + The class of the new observation will the class containing the majority of the k nearest neighbours

Here is an illustration of how KNN works

If k=3, the class of the **red star** (the new observation) will be **class B**

If k=6, the class of the new observation will be **class A**

**Advantages of KNN:**

* + Not too affected by the disturbing variations contained in the training data
  + The larger the train set is, the more effective is KNN
  + No training phase
  + Easy to compute complex models

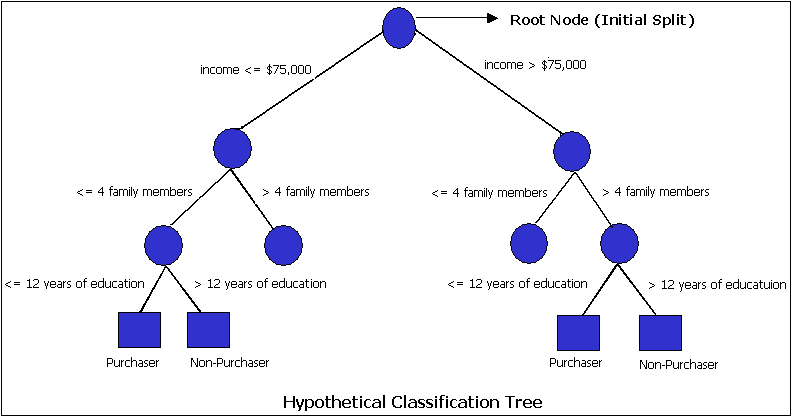
**Disadvantages of KNN:**

* + Difficulty to determine the best value for K
  + The distance metric to use is not clear
  + High computation cost

**N.B:** Toavoid draws(match nul) in the votes:

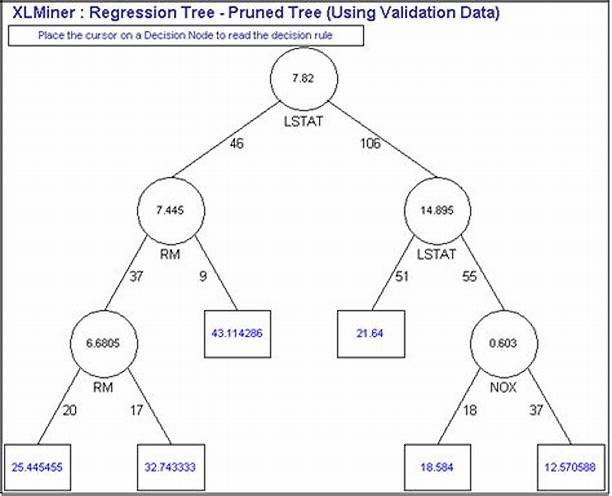
* + **K** should be an odd number
  + K should always be greater than the number of classes
* **classification and regression trees**
  + **Classification trees** are used with categorical (non numeric) dependent variables. They are used to predict the class of an observation. The tree’s building is stopped when the number of instances in a region is too small and/or when the class distribution is pure enough (assez homogène).

Here is an example of classification tree



* + **Regression trees** are used with numerical continuous dependent variables. They predict the mean value of an observation. For this kind of trees, the best splitting is the one with the minimal **Mean Squared Error**

Here is an example of regression tree



**N.B.**

* + When the tree is too complex (too many branches), there is a high risk of overfitting. To reduce that complexity we use the pruning (the act of cutting a tree’s branch).
  + To know how far to prune a tree, we use cross validation and compute many trees with different number of branches. The number of branches with the lowest error is chosen for the final tree

**Advantages**

* + Trees are easy to understand
  + They work well with classification and regression problems
  + They can be plotted graphically and are easily interpreted

**Disadvantages**

* + Not as good prediction accuracy as the other more complex models
  + The larger trees get, the more difficult it is to understand them

**Trees vs linear model**

* + **In regression problems:** When the relation between the predictors and the dependent variable is linear, linear models are better. Otherwise, when that relation is not linear, trees are better
  + **In classification problems:** When the boundary between the regions is linear, linear models are better. Otherwise, when that boundary is non linear, trees are better.
* **stepwise and locally weighted regression**
  + **Cours chapitre 5 page 35**
* **Naive Bayes**

L’objectif est de calculer pour une observation donnée la probabilité d’appartenance à une classe. Cette probabilité est calculée pour chaque classe. La classe ayant la probabilité la plus élevée est celle dans laquelle sera rangée l’observation.

Toutes les variables sont supposées indépendantes.

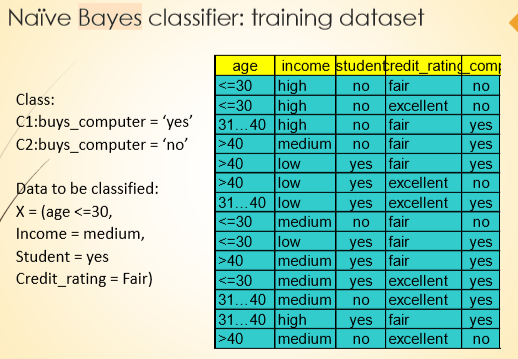
Supposons une observation **X(x1,x2,..,xk)** pouvant appartenir aux classes **Ci avec 0≤i≤n**

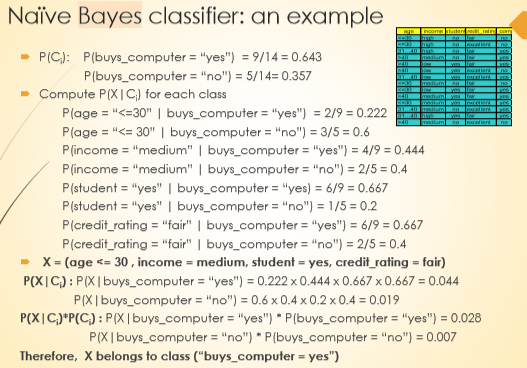
Pour classifier **X** il faut calculer tous les **Pi=P(X/Ci)\*P(Ci)** **avec 0≤i≤n**

Avec **P(X/Ci)= P(x1/Ci)\*P(x2/Ci)\*…\*P(xk/Ci)**

L’observation **X** sera rangée dans la classe avec le Pi le plus élevé.

Voici un exemple plus concret:





**Advantages :**

* + Easy to implement, fast even on very large data sets, parallelizable
  + Good results obtained in many cases
  + Better results with observations containing categorical values(non numeric values)

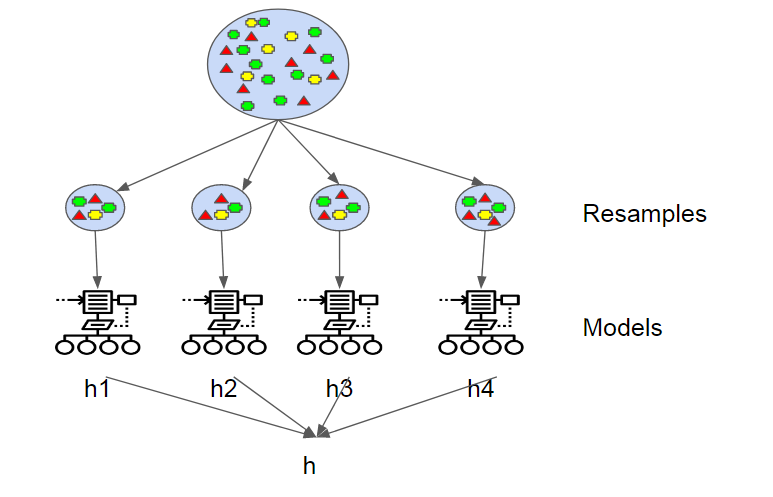
**Disadvantages :**

* + Perte de précision due au fait qu’on suppose toutes les variables indépendentes

**N.B:** Pour que cet algorithme soit utilisé il faut que toutes les probabilités soient différentes de 0. En cas de probabilités nulles, on utilise la correction de Laplace (Laplacian correction) pour dénullifier toutes les probas. (cf PMA Lecture 4 – page 95)

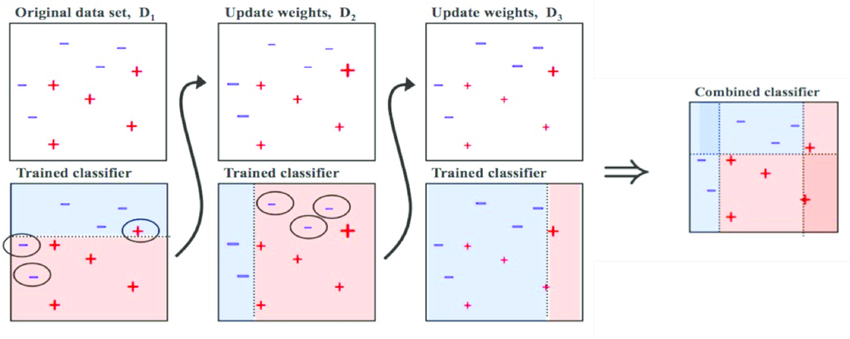
* **Bagging (Parallel Method)**

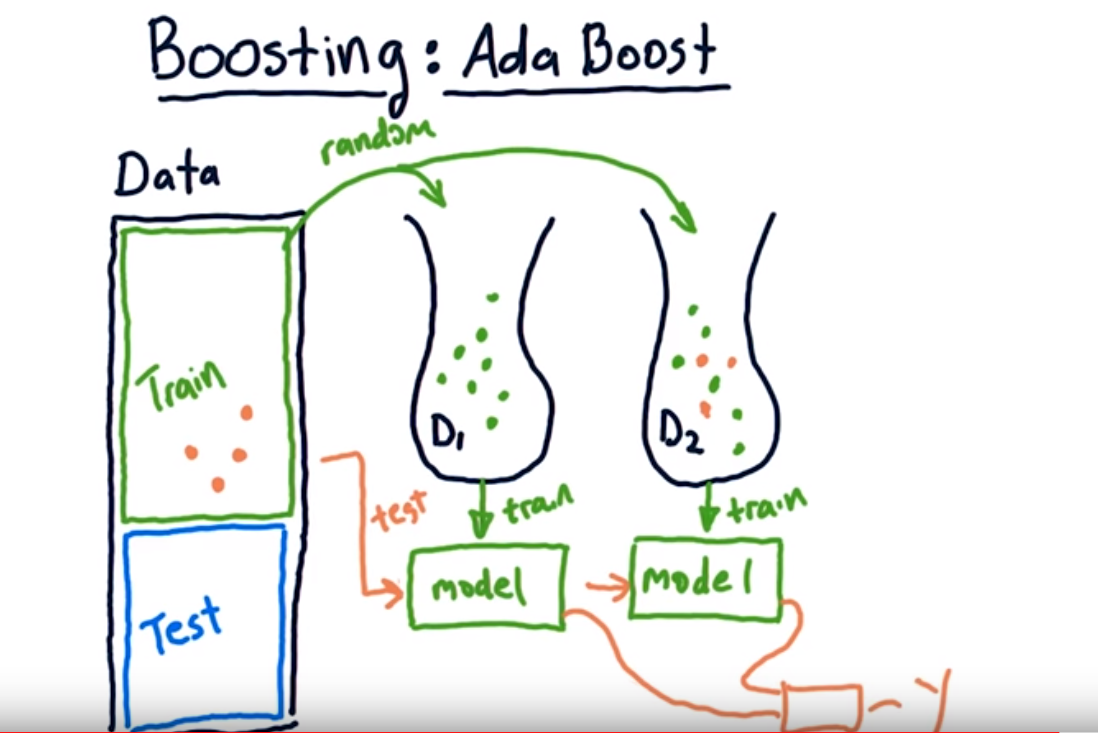
Can apply to Regression and Classification



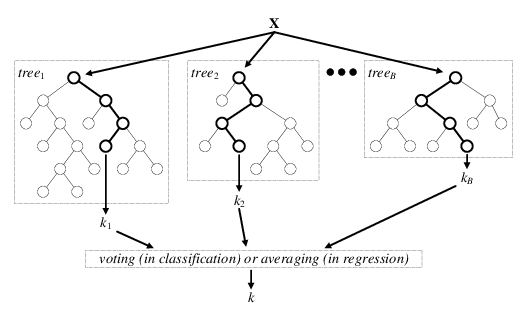
* + Split train data ramdomly n sample (can find same data on the same part)
  + For each part apply the model
  + Take average of all results
* **Boosting :** Ada Boost Algorithm (variation of Bagging, Sequential Method)

Can apply to Classification

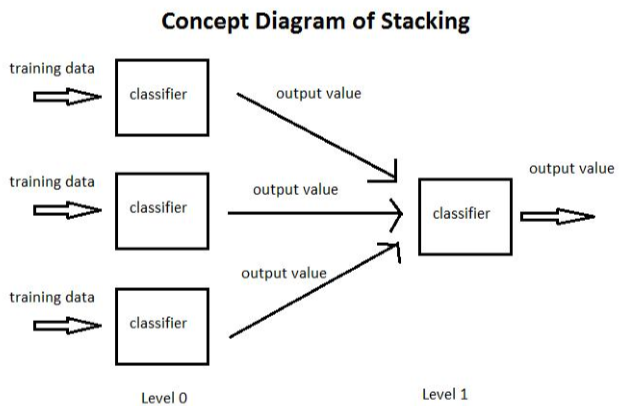




* + Build the first model
    - Take ramdomly a sample from the training dataset
    - Apply the model and test the model with all training dataset
  + Build the second model
    - Take error results from the first model and take randomly a sample from the training dataset
    - Apply the model with previous model and test the model with all training dataset
    - Combine the output on Y
  + Build n models with the same method used for second model
* **random forests ( combine Bagging and random sub-spaces)**



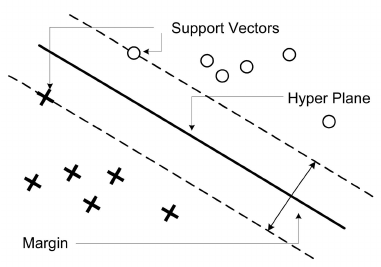
* + Bootstrapping the dataset
  + Make N classification trees, N being the number of columns of the dataset. Each column of the data set becomes the root of one of the trees
  + Pass all the dataset on each trees
  + Voting or averaging for each lines of the dataset
  + The high score are K
  + Properties :
    - low classification (and regression) error
    - no overfitting
    - robust concerning the noise and the number of attributes
    - relatively fast
* **Stacking**



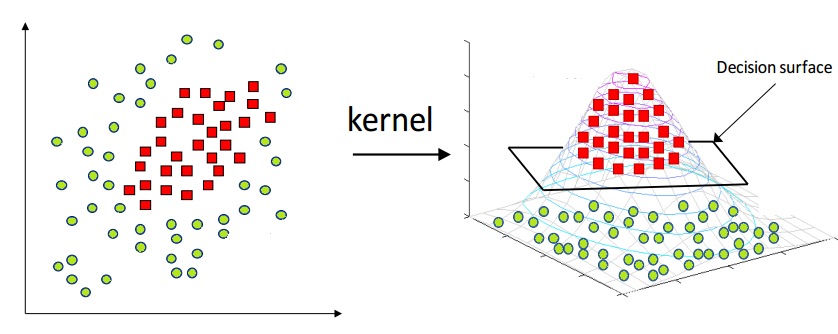
* + On level 0 make different methods of classification as RF, KNN, Booting, etc
  + On level 1 classify output values of each method : STACKING
* **SVM** (Support Vector Machine)

It use to distinct(separate) two class classification

* + Principle



* + Find support vectors for each group data
  + Find margins for each support vectors
  + Kernel



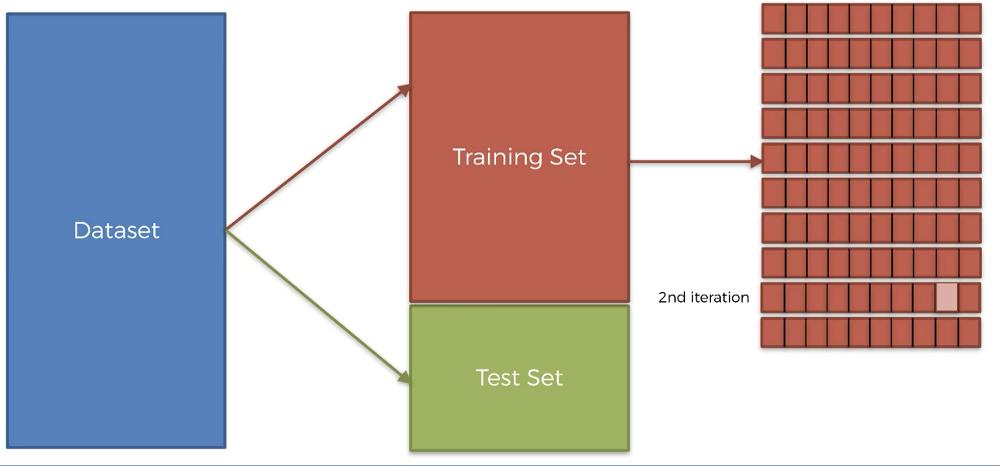
* **Neural networks**
  + **Backpagation**

**properties and purpose of evaluation approaches and metrics:**

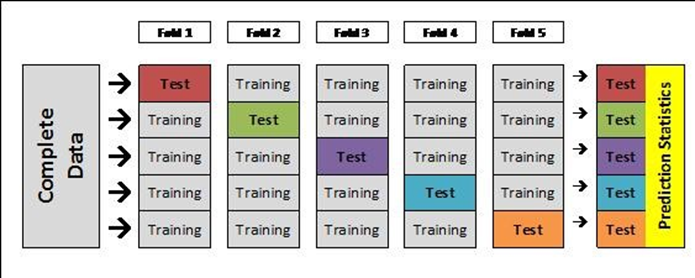
* **Crossvalidation**

**Method to split data for predictive methods**

* + **Split the dataset on 2 parts : Training set and Test set**

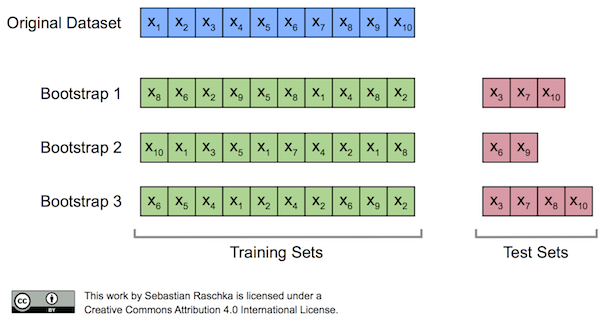


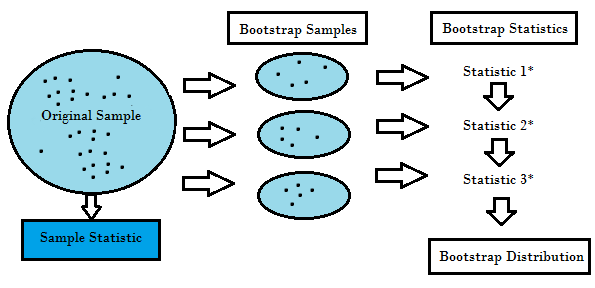
* + Split the training set and apply the model on the data



* + After the training testing the algorithm on the test set
* **Bootstrapping (Resampling)**

Method to split data for predictive methods





* **ROC curves (**Receiver operating characteristic)
* **Sensitivity**
* **specificity**