Some classical ML models

Roadmap

- We shall present some classical parametric ML models
- As in Lecture 2, we shall denote X the $n \times p$ data matrix or design matrix
- White box models vs black box models
  - Some models are white-box models, meaning that the knowledge of their coefficients brings clear information about the impact of each input variables. Examples : Linear Models, Decision Tress
  - The opposite are black box models. Examples : Ensemble Methods, Neural Networks

Linear Regression

- $\mathcal{D}_n = \{(x_i, y_i) \in \mathcal{X} \times \mathbb{R},\ i = 1, \cdots, n\}$
- $f_\beta(x) = \beta_j x_j + \beta_0$?
- Least-squares fit (equivalent to MLE under the assumption of Gaussian noise):

$$\widehat{\beta} := \text{Argmin}||Y - X \cdot \beta||^2 = (X^T X)^{-1} XY$$

Solution uniquely defined when $X^T X$ invertible

Ridge regression (Hoerl & Kennard 1970)

- To ensure uniqueness, one may add a penalty and solve

$$\widehat{\beta} := \text{Argmin}||Y - X \cdot \beta||^2 + \lambda||\beta||^2$$

- Solution unique and always exists

$$\widehat{\beta} := (X^T X + \lambda I)^{-1} XY$$

- λ is an hyperparameter. Has to be tuned!

Ridge regression (Hoerl & Kennard 1970)

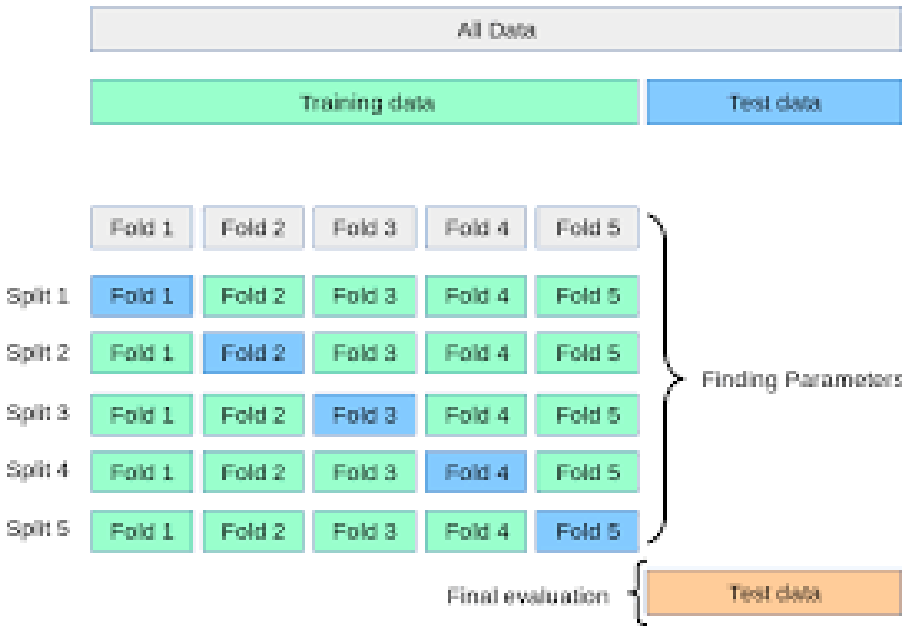
Tuning of λ?

Data splitting strategy: cross-validation:

- Cut the training set in k equally-sized chunks.
- K folds: one chunk to test, the $K-1$ others for training
- Cross-validation score: perf averaged over the K folds.

Ridge regression (Hoerl & Kennard 1970)

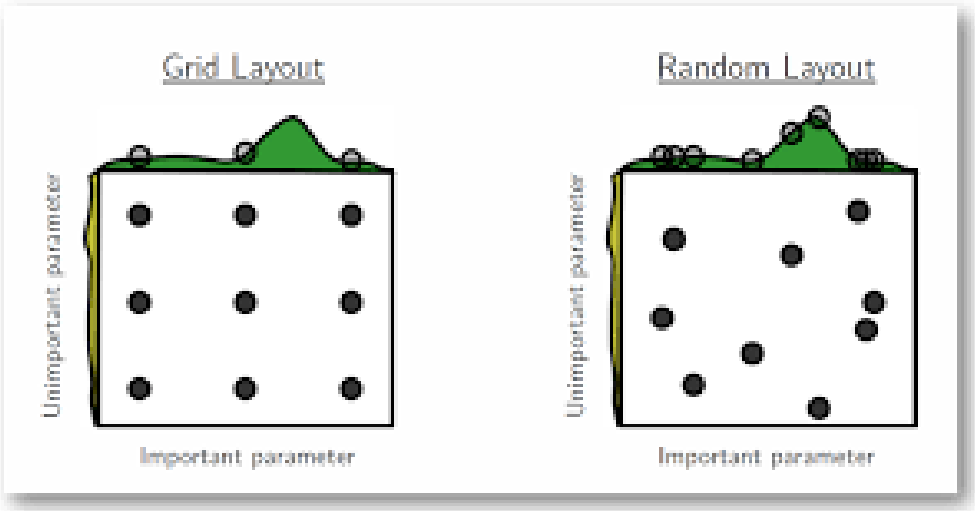
Tuning of λ?



Ridge regression (Hoerl & Kennard 1970)

Tuning of λ?

On each fold we use a grid search on λ



Ridge regression (Hoerl & Kennard 1970)

A practical example

- We study data from D. Card (2001) about the impact of education on labor market earnings[1]
- The dataset can be downloaded on `http://fmwww.bc.edu/ec-p/data/wooldridge/card.dta`
- We try to explain the variable `educ` in function of the other ones
- We compare OLS and Ridge without tuning of λ

[1] https://davidcard.berkeley.edu/papers/return-to-schooling.pdf

Ridge regression (Hoerl & Kennard 1970)

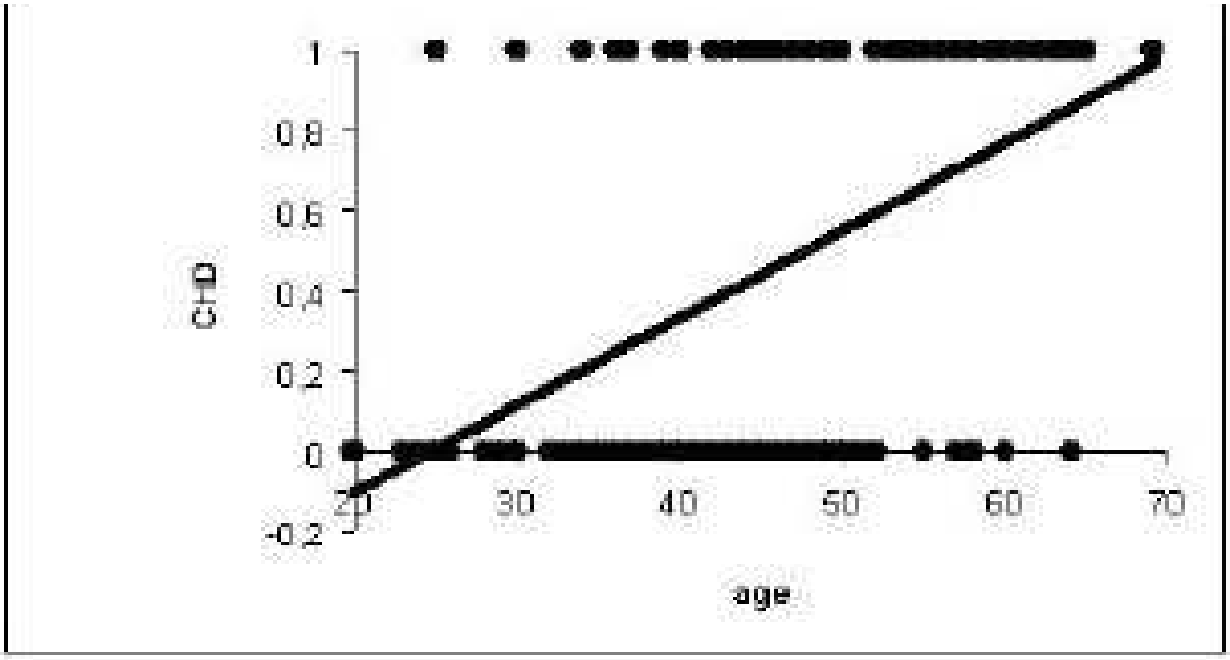
A practical example

	OLS	RIDGE	RIDGE2
Train	0.536856	0.536855	0.536854
Test	0.524978	0.525048	0.525118

A small improvement with ridge

Logistic Regression

If $y \in \{-1, 1\}$, linear regression makes no sense!



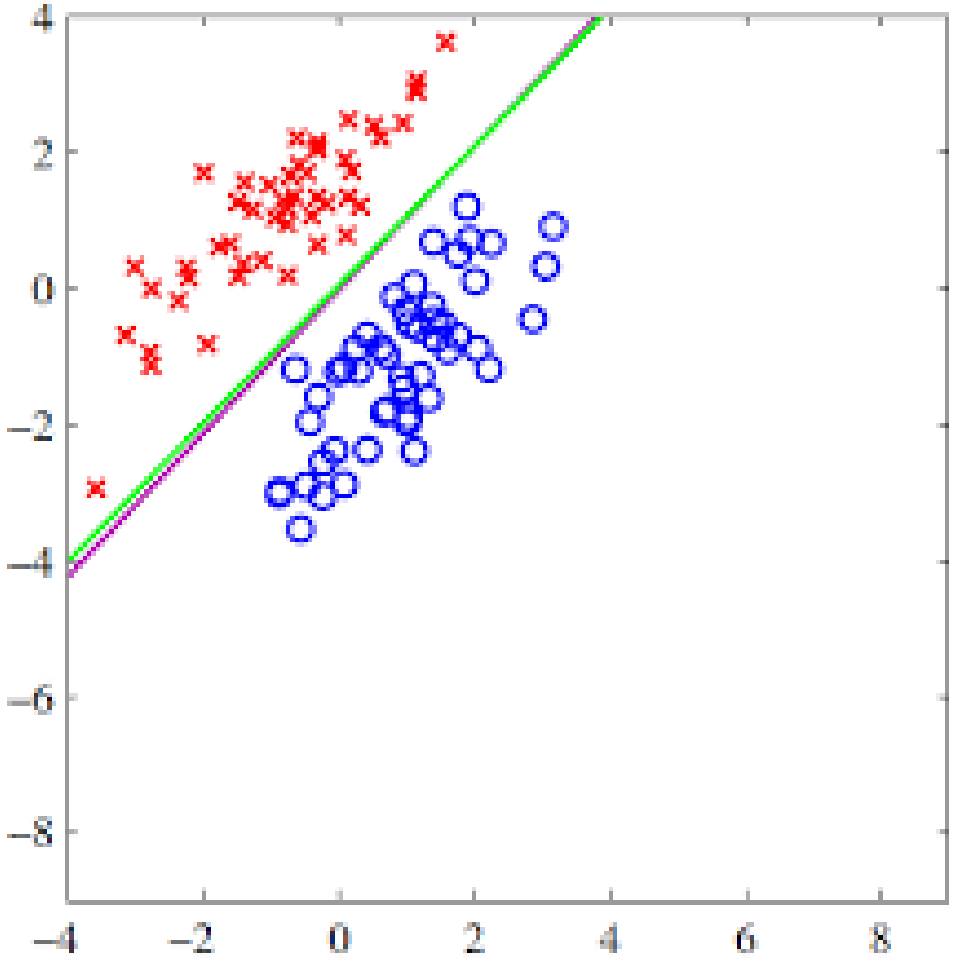
Logistic Regression

- Model the log-odds ratio as a linear function of x?

$$\log\left(\frac{P[Y=1|x]}{1-P[Y=1|x]}\right) = \sum_j \beta_j x_j + \beta_0 = f_\beta(x)$$

- One can also add a regularization.

Support Vector Machine (SVM)



How can we find a separating hyperplane between two classes?

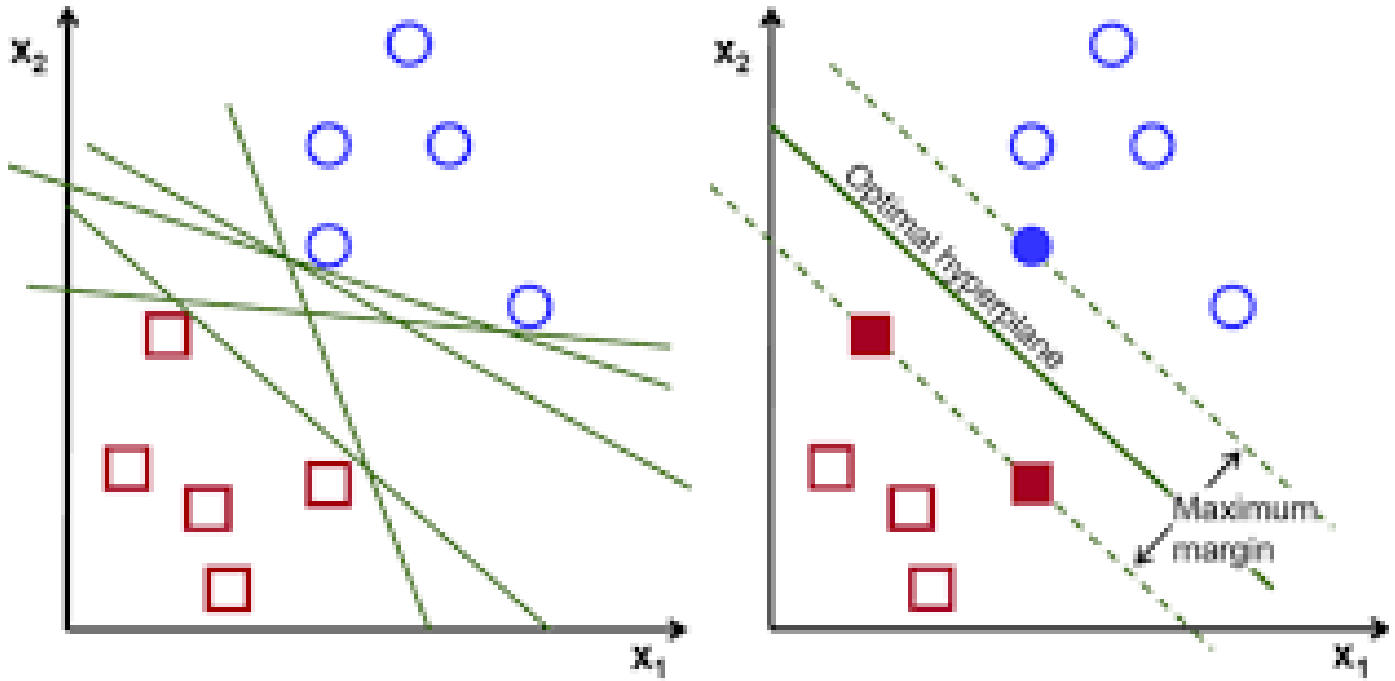
Support Vector Machine (SVM)

Linear SVMs for classification problems

- We want to find a decision function of SVM of the form $f_w(x) = \text{sgn}(w^T x + b)$
- That is
  - $w^T x + b = 0$: decision boundary
  - if $w^T x + b > 0$ assign label $y_i = 1$
  - if $w^T x + b < 0$ assign label $y_i = 1$

Support Vector Machine (SVM)

Linear SVMs for classification problems



How can we find a separating hyperplane between two classes?
Solution : we maximize the margin

Support Vector Machine (SVM)

Linear SVMs for classification problems

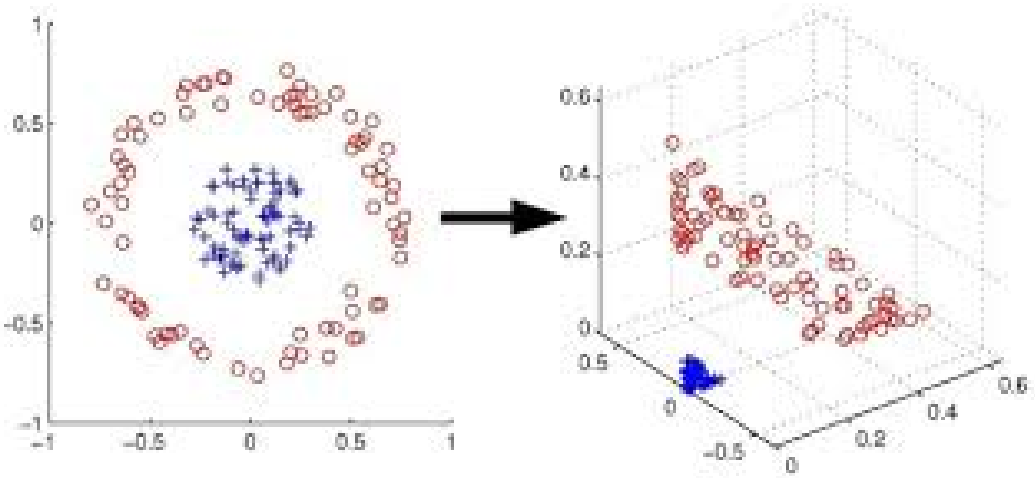
- Margin maximization can be reformulated as the non-smooth, penalized convex optimization problem

$$\text{Argmin}_w \sum_i \min(1 - y_i(w^T x_i + b), 0) + \|w\|^2$$

- Solved using quadratic programming in the dual domain.
- Observe that predicting the label of a new observation involves only a scalar product $(f_w(x) = \text{sgn}(w^T x + b))$

Support Vector Machine (SVM)

Non linear SVMs for classification problems



What can we do in the non linear case?

Support Vector Machine (SVM)

Non linear SVMs for classification problems

- Transform the non linear problem into a linear one, using a non linear function ϕ and adapt linear SVM
- In the non linear SVM algorithm, the quantity $K(x_{new}, \cdot) =< \phi(x_{new}), \phi(\cdot)$ is involved for the prediction of the label of any new observation. It is a kernel
- In non linear SVM we have only to choose this kernel K. It is the kernel trick

Several possible kernels available in sklearn

`https://scikit-learn.org/stable/modules/svm.htmlsvm-kernels`

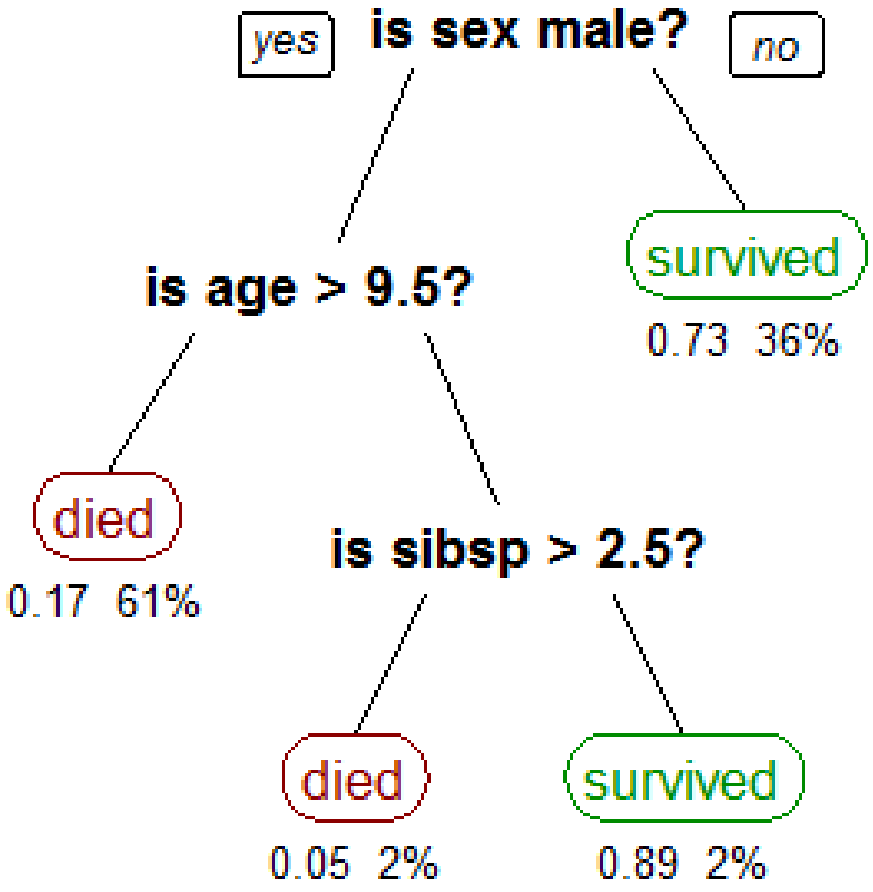
Decision trees

Decision trees

- An exploratory analysis tool
- Representation of the data in a hierarchical manner by a sequence of test allowing to predict an output variable

Decision trees

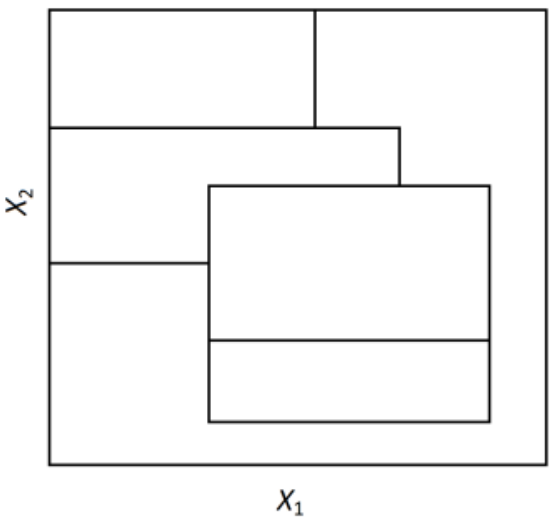
Who should be saved in Titanic?

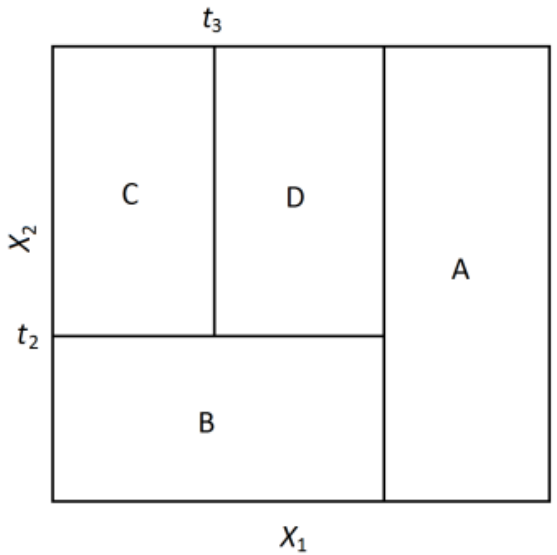


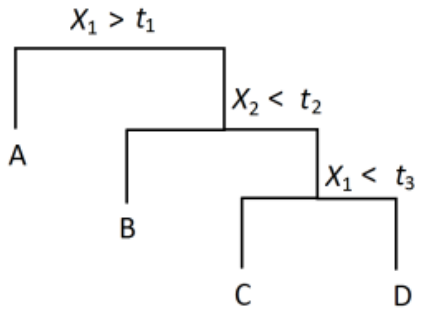
Decision trees

- Each node is corresponding to a variable and test it
- One generates several leafs at each iteration corresponding to a partition in the space of input variables

Decision trees



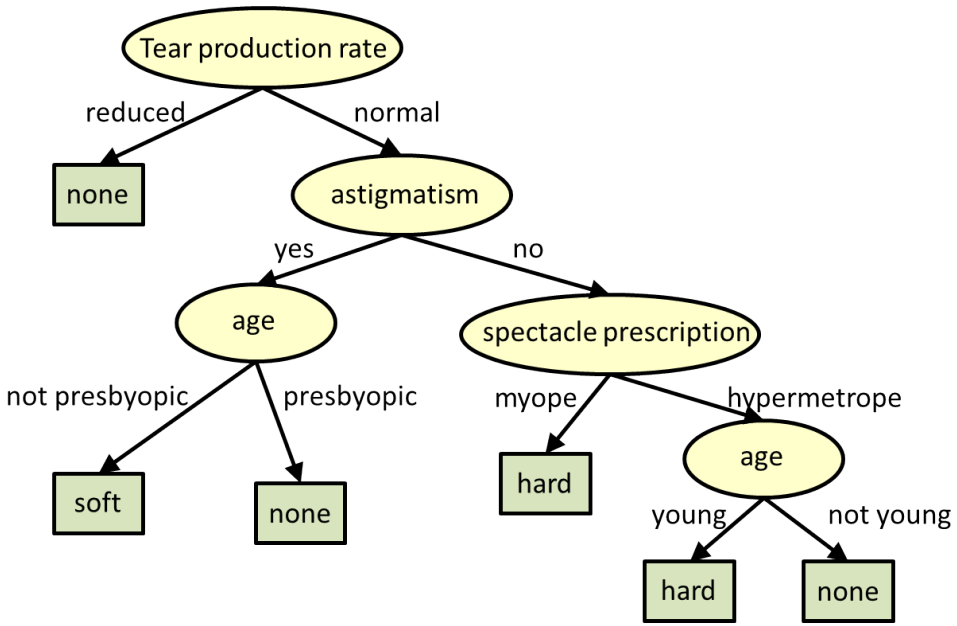




Decision trees

- Inputs : points of the `features` spaces that are characterized by numerical or categorical variables
- Target : classes (classification) or value (regression)

Decision trees



Decision trees

A two step procedure to define decision trees

- Construction of a maximal tree
- Pruning : construction of a sub-sequence of optimal decision trees to avoid overfitting

Decision trees

Construction of the maximal tree

- Creating a binary decision tree is actually a process of dividing up the input space.
- A greedy approach is used to divide the space called recursive binary splitting

Decision trees

Construction of the maximal tree

- Splitting rule?
- To simplify, let us assume that the input variables are continuous
- A split is of the form

$$\{X^{(j)} \leq d\} \cup \{X^{(j)} > d\}$$

- Several possible splits? We choose (j, d), minimizing a given cost function

Decision trees

Splitting rule in the regression case

- The aim is to minimize the intra-group variance after split of a node t into two nodes t_L and t_R
- The variance of a node t is defined as

$$V(t) = \frac{1}{\#t} \sum_{i,\, x_i \in t} (y_i - \bar{y}_t)^2$$

where y_t is the mean of the values y_i associated to the observations of node t

- We have then to minimize

$$V(t) = \frac{\#t_L}{\#t} \sum_{i,\, x_i \in t_L} (y_i - \bar{y}_{t_L})^2 + \frac{\#t_R}{\#t} \sum_{i,\, x_i \in t_R} (y_i - \bar{y}_{t_R})^2 = \frac{\#t_L}{\#t} V(t_L) + \frac{\#t_R}{\#t} V(t_R)$$

Decision trees

Splitting rule in the classification case

- In the classification case, where the classes belong to $\{1, \cdots, L\}$, the impurity of each node could be defined by the means of Gini index
- The Gini index of a node t is defined as

$$\Phi(t) = \sum_{c=1}^{L} \widehat{p_t^c}(1 - \widehat{p_t^c})$$

where $\widehat{p_t^c}$ is the proportion of observations belonging to class c in node t.

- The aim is then to maximize for each node t and each possible split

$$\Phi(t) - \frac{\#t_L}{\#t}\Phi(t_L) - \frac{\#t_R}{\#t}\Phi(t_R)$$

Decision trees

- Second step of the algorithm : pruning
- We search for the best pruned sub-tree (best : lowest generalisation error).
- Maximal tree : low bias and high variance
- We want to decrease the variance

Decision trees

- Pruning : model selection procedure, where the models are all possible subtrees
- This procedure minimises a penalized criterion where the penalty is proportional to the number of leafs of the tree

$$crit_\alpha(T) = \overline{err}(T) + \alpha|T|$$

where

$$\overline{err}(T) = \frac{1}{n} \sum_{\text{t feuille de } T} \sum_{(x_i,y_i)\in t} (y_i - \bar{y}_t)^2$$

Decision trees

Pro and cons

Pro

- Interpretable model
- Few preprocessing
- Numerical cost low
- Inputs can be both qualitative and quantitative

Cons : instability

- Few change in the data can lead to very different DT
- High variance estimators

Ensemble methods

Principle

- Ensemble methods are techniques that create multiple models and then combine them to produce improved results
- Assume that we are given $G_1(\cdot), \cdots, G_q(\Theta_q)$ a collection of ML models $\theta_1, \cdots, \Theta_q$ q i.i.d. random variables aléatoires
- We can define a predictor G in several ways
  - Averaging : $G = \frac{1}{q} \sum G_s(\cdot)$
  - Majoritary vote : $G = argmax_k \sum_s 1_{G_s(\cdot)=k}$
- In what follows, our building blocks will be decision trees

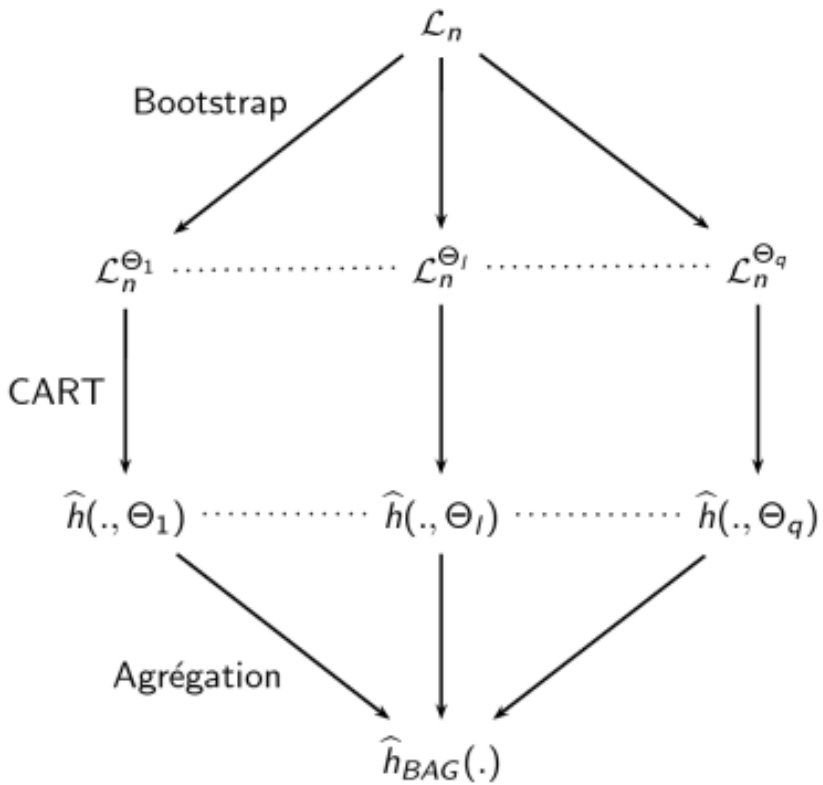
Bagging

Bagging consists in the following procedure

- We sample by bootstrap from the initial training set q datasets $\mathcal{D}_n^{\theta_1}, \cdots, \mathcal{D}_n^{\Theta_q}$,
- Then q decisions trees are fitted $G(\cdot, \mathcal{D}_n^{\theta_1}), \cdots, G(\cdot, \mathcal{D}_n^{\Theta_q})$
- One aggregates these base predictors

Bagging

General principle



Random Forests-Random Inputs

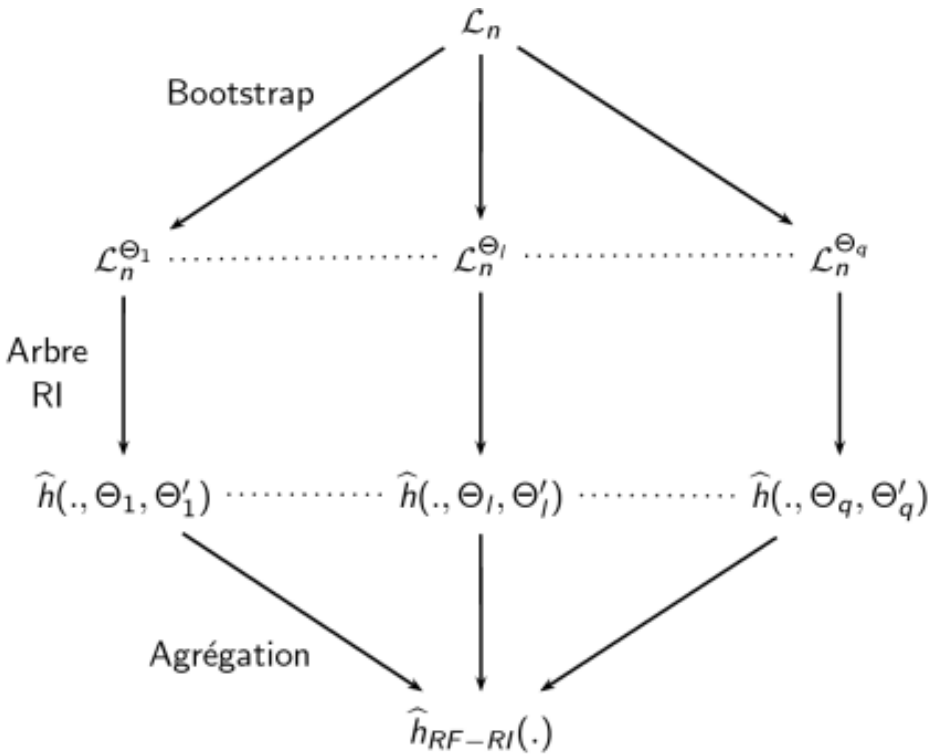
- Useful when the number of observations and the number of variables are both large
- A first step consists in generating several samples as in bagging
- Thereafter on each sample, we apply a variant of CART
  - To split a node we first draw at random, a given number on tire m of variables
  - One searches at the best split among the m selected variables
- One aggregates this family of decision trees

Random Forests-Random Inputs

- If $m = p$, we recover Bagging
- If $m = 1$ we have a procedure very different from Bagging. The choice of split variable is completely random

Random Forests-Random Inputs

General principle



Random Forests-Random Inputs

- In practice, performance of RF are better than Bagging
- Heuristic explanation : added randomness helps to make as different as possible the decision trees

Random Forests-Random Inputs

OOB error

- The RF algorithm also allows to estimate the generalisation error of the model
- This error is the Out-Of-Bag (OOB) error

Random Forests-Random Inputs

OOB error

- Let us fix an (x_i, y_i) of the training set
- Let us consider the set of all trees defined on bootstrap samples not co,taining this observation
- One agregates only the prediction of these trees to define our prediction $\widehat{y}$ of y_i
- After computing this quantity for all observations, one calculate the global error
- This quantity is called OOB error of the Forests-RI predictor

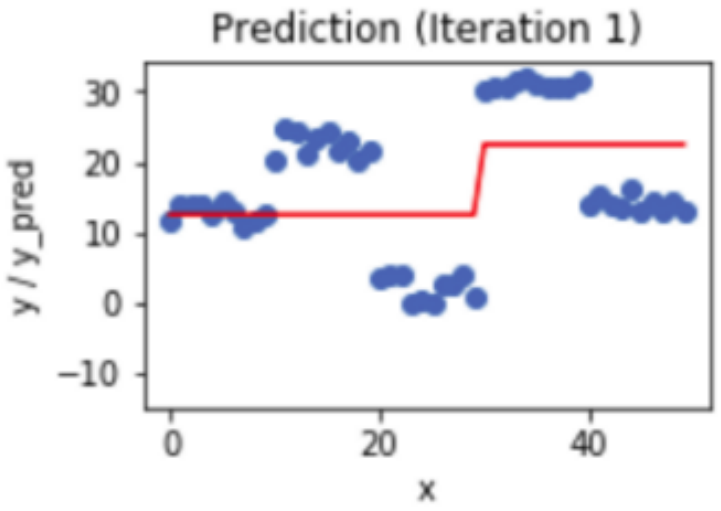
Gradient Boosting Trees

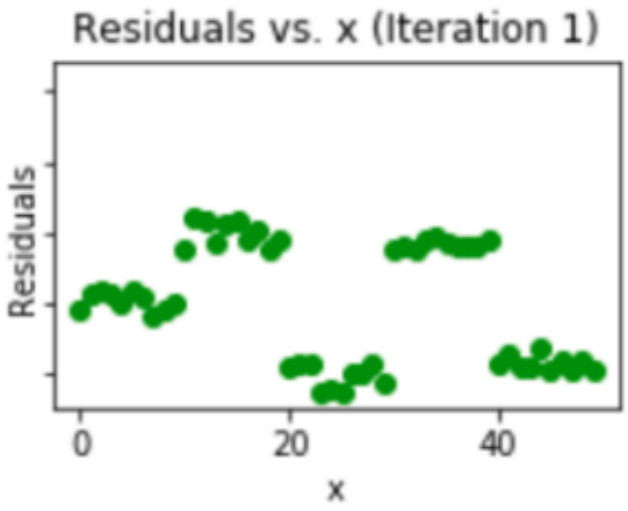
The algorithm of Gradient boosting se fait en différentes étapes

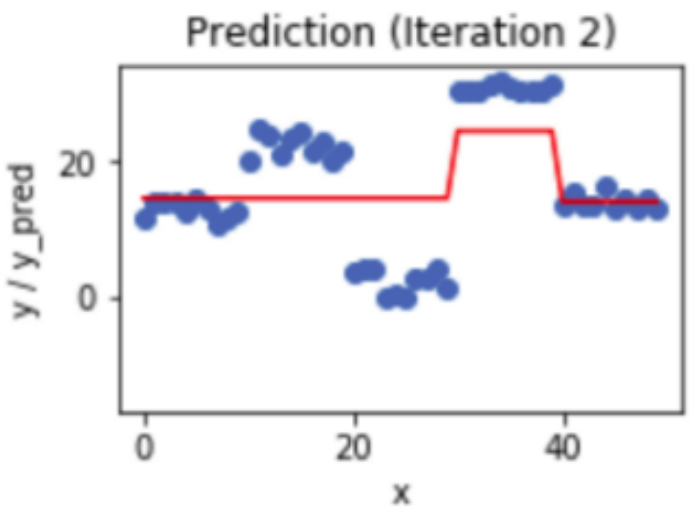
- One fits a decision tree
- One calculates the residuals
- One fits a decision tree on residuals and one adds it to the previous one

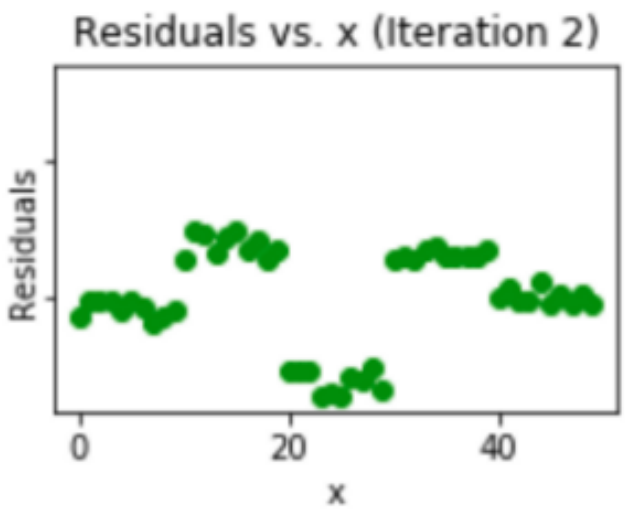
One iterates the procedure.

Gradient Boosting Trees









Why quantile regression?

- Classical regression estimate conditional expectation
- How could we estimate the median or the percentiles of the output random variable?

Why quantile regression?

- Quantile regression models the whole distribution of the output variable whereas classical regression only allows to estimate the conditional expectation
- To perform Quantile Regression, we do not need any assuption on the output random variable. More robust to misspecification and outliers
- Quantile Regression is invariant with respect to monotonous transformation

Why quantile regression?

When should we use quantile regression?

- To have prediction intervals
- No assumption on the output variable

An example

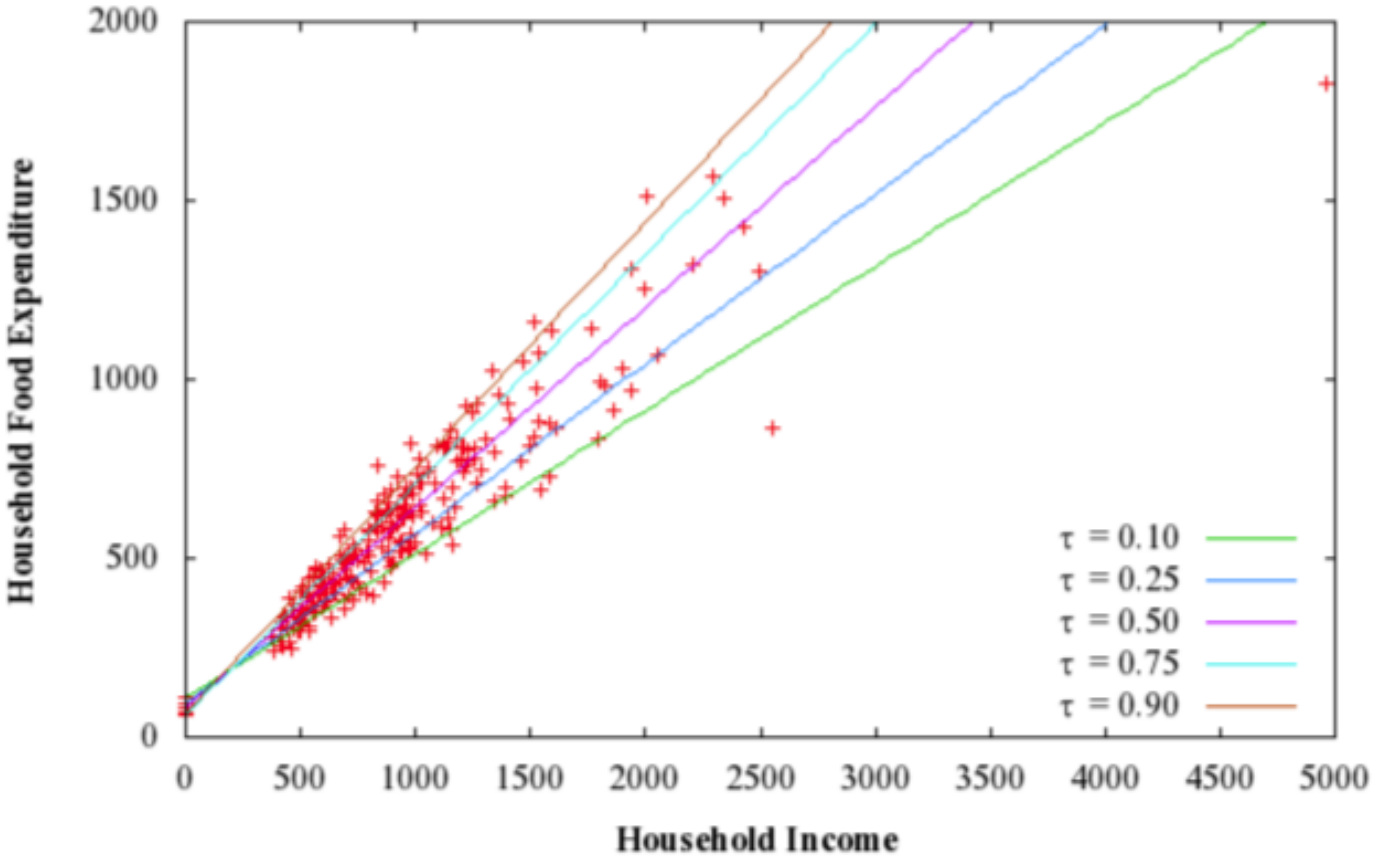


Figure (A): Household Income vs. Food Expenditure

An example

Two situations

- Graphique (A) : variance of Y is constant whatever X may be. Classical regression may be adapted
- Graphique (B) : variance of Y increases when X increases. Quantile regression may be more adapted than classical one

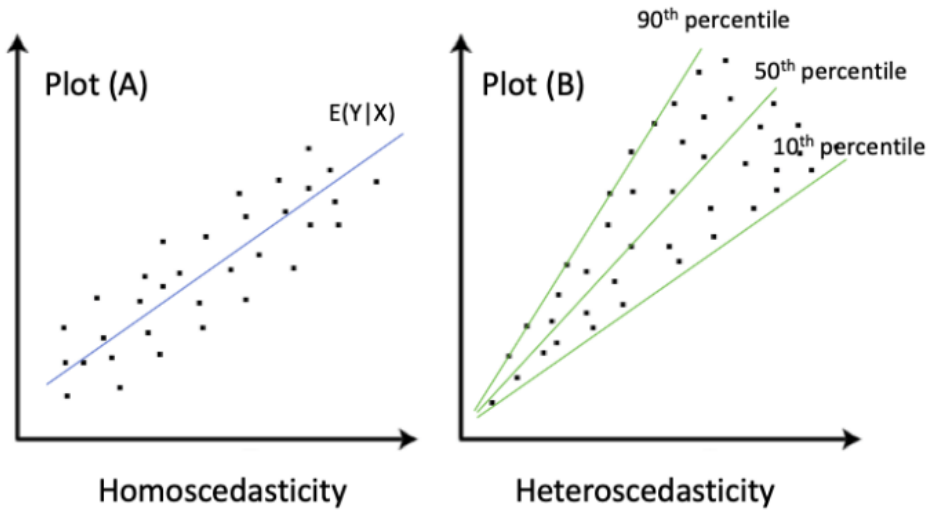


Figure (B): Homoskedasticity vs. Heteroscedasticity

An example

The Boston Housing Dataset

- Boston Housing dataset : 506 observations and 14 variables.
- Aim : predict the median price of a property in function of some variables

An example

The Boston Housing Dataset

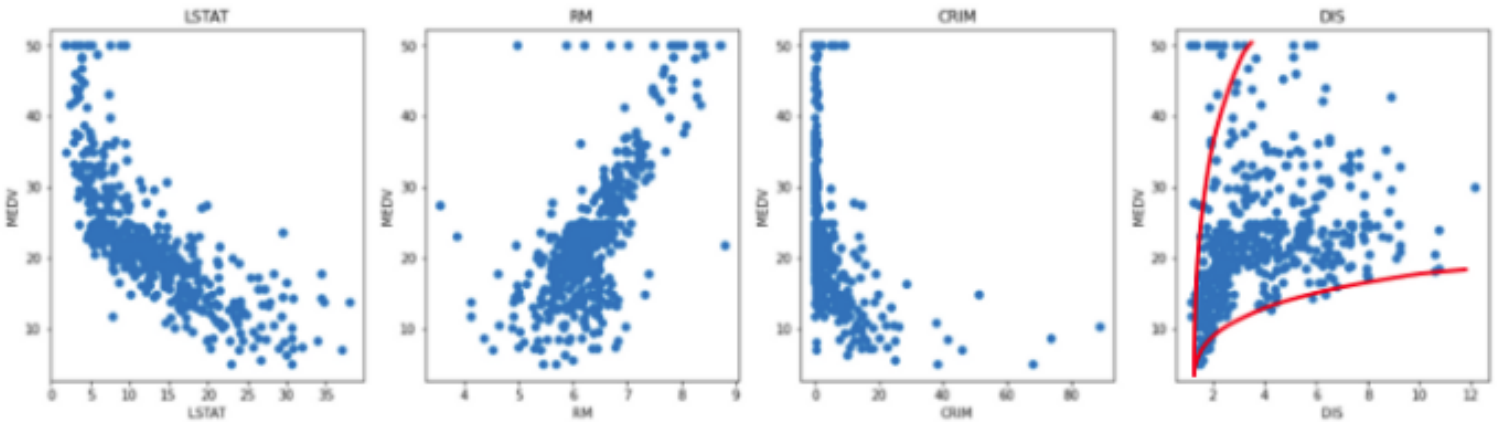
Explicative variables

- CRIM: per capita crime rate by town.
- ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS: proportion of non-retail business acres per town.
- CHAS: Charles River dummy variable (= 1 if tract bounds river).
- NOX: nitrogen oxides concentration (parts per 10 million).
- RM: average number of rooms per dwelling.
- AGE: proportion of owner-occupied units built prior to 1940.
- DIS: weighted mean of distances to five Boston employment centres.
- RAD: index of accessibility to radial highways.
- TAX: full-value property-tax rate per $10,000.
- PTRATIO: pupil-teacher ratio by town.
- LSTAT: lower status of the population (percent).

An example

The Boston Housing Dataset

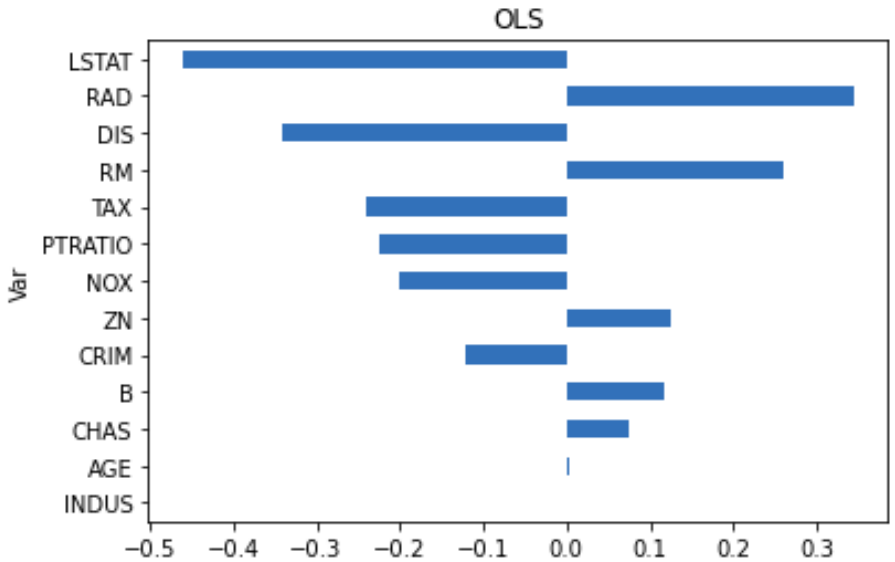
Heteroscedasticity?



An example

The Boston Housing Dataset

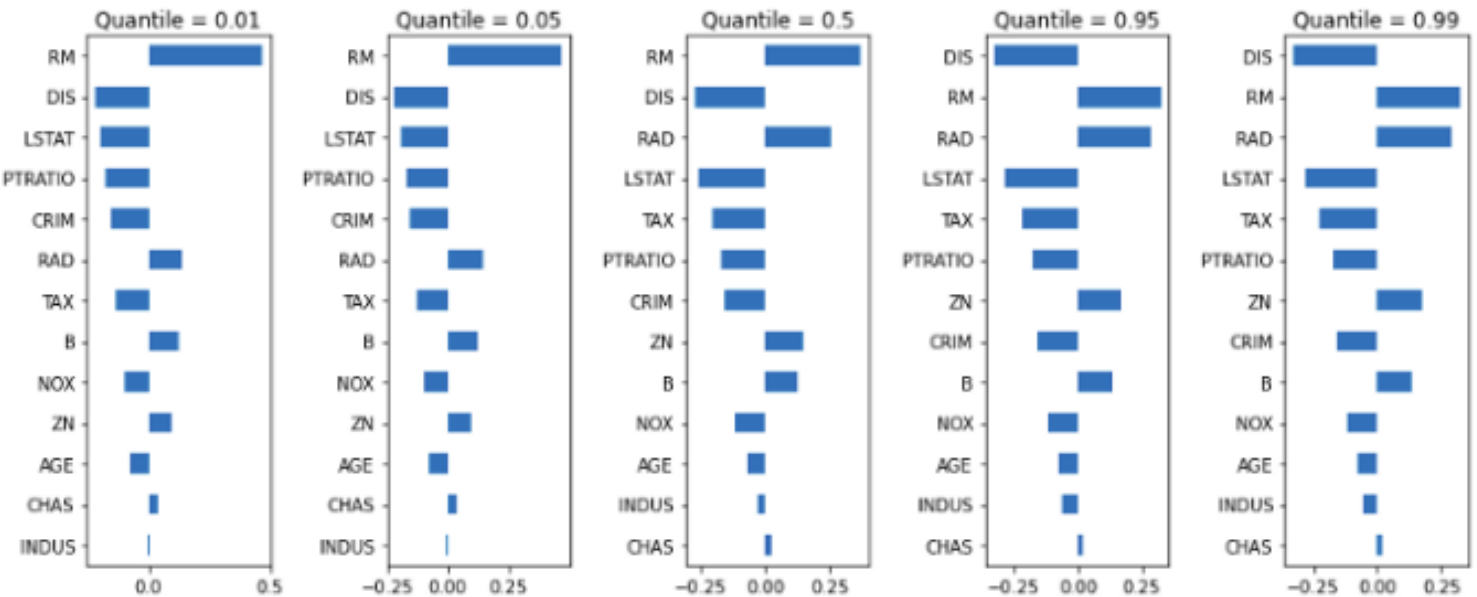
Classical linear regression



An example

The Boston Housing Dataset

Quantile regression for five percentiles



Un exemple

How it works?

Example on $y_i = x_i\beta + \varepsilon_i$

- Classical regression : one minimises the sum of the squares of the errors

$$L(\beta) = \sum_i (y_i - x_i\beta)^2$$

- Median regression : one minimises the sum of absolute values of the erros

$$L(\beta) = \sum_i |y_i - x_i\beta|$$

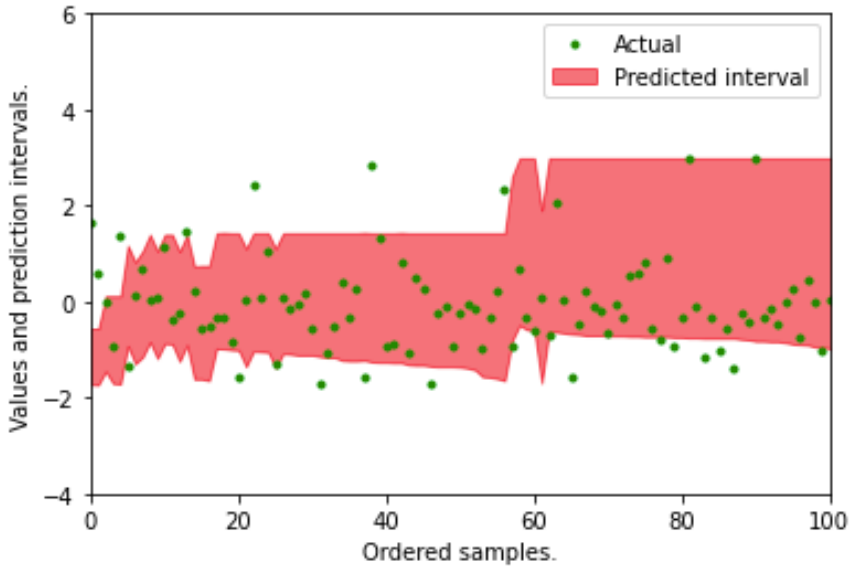
- Quantile Regression for an α quantile

$$L_\alpha(\beta) = \sum_i \rho_\alpha(y_i - x_i\beta) \text{ avec } \rho_\alpha(z) = \begin{cases} z(\alpha - 1) \text{ si } z < 0 \\ \alpha z \text{ si } z > 0 \end{cases}$$

An example

The Boston Housing Dataset

Confidence intervals for prediction



Quantile Random forest

1. As in RF, one generates k trees. One keeps in memory all values of the observations associated to this node
2. For $X = x$ given at each tree t one calculates the quantile minimising the quantile loss

Quantile Random forest

The main differences between random forest and quantile random forest are the following

- For each node of each tree, random forest keep only the mean of the observations associated to this node
- Quantile Random Forest keep in memory all values of the observations associated to this node
- We can deduce conditional distributions of the output variable and not only its conditional mean

An example

The Boston Housing Dataset

Confidence intervals for prediction