A Gentle Introduction to Data Science

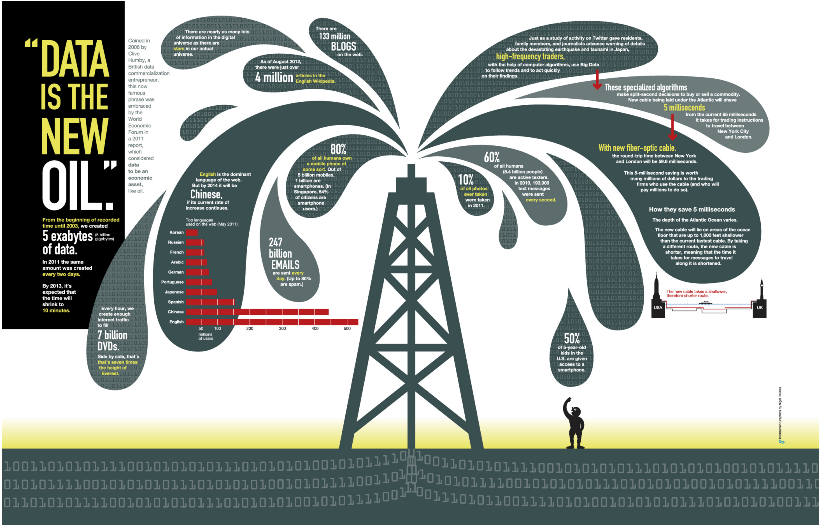
E. Le Pennec

DSE2017 - Brest - 04/07/2017

Outline

Outline

Data Is The New Oil?



A New Context

Data everywhere

- Huge volume,
- Huge variety...

Affordable computation units

- Cloud computing
- Graphical Processor Units (GPU)...

- Growing academic and industrial interest!

Data Science

Major Influences

Four major influences act today:

- The formal theories of statistics
- Accelerating developments in computers and display devices
- The challenge, in many fields, of more and ever larger bodies of data
- The emphasis on quantification in an ever wider variety of disciplines

Data Science

Major Influences - Tukey (1962)

Four major influences act today:

- The formal theories of statistics
- Accelerating developments in computers and display devices
- The challenge, in many fields, of more and ever larger bodies of data
- The emphasis on quantification in an ever wider variety of disciplines

- He was talking of Data Analysis.
- Data mining, Machine learning, Big Data...

Example of *off the shelves* solution

Spark Lightning-fast cluster computing

amazon web services

```scala
def run(params: Params) {
  val conf = new SparkConf()
      .setAppName(s"BinaryClassification with $params")
  val sc = new SparkContext(conf)

  Logger.getRootLogger.setLevel(Level.WARN)

  val examples = MLUtils.loadLibSVMFile(sc, params.input).cache()

  val splits = examples.randomSplit(Array(0.8, 0.2))
  val training = splits(0).cache()
  val test = splits(1).cache()
  val numTraining = training.count()
  val numTest = test.count()
  println(s"Training: $numTraining, test: $numTest.")
  examples.unpersist(blocking = false)

  val updater = params.regType match {
    case L1 => new L1Updater()
    case L2 => new SquaredL2Updater()
  }

  val algorithm = new LogisticRegressionWithSGD()
      algorithm.optimizer
        .setNumIterations(params.numIterations)
        .setStepSize(params.stepSize)
        .setUpdater(updater)
        .setRegParam(params.regParam)
  val model = algorithm.run(training).clearThreshold()

  val prediction = model.predict(test.map(_.features))
  val predictionAndLabel = prediction.zip(test.map(_.label))

  val metrics = new BinaryClassificationMetrics(predictionAndLabel)
  val myMetrics = new MyBinaryClassificationMetrics(predictionAndLabel)

  println(s"Empirical CrossEntropy = ${myMetrics.crossEntropy()}.")
  println(s"Test areaUnderPR = ${metrics.areaUnderPR()}.")
  println(s"Test areaUnderROC = ${metrics.areaUnderROC()}.")

  sc.stop()
}
```

Big Data Is (Quite) Easy

Example of *off the shelves* solution

```
export AWS_ACCESS_KEY_ID=<your-access-keyid>
export AWS_SECRET_ACCESS_KEY=<your-access-key-secret>
cellule/spark/ec2/sparl-ec2 -i cellule.pem -k cellule -s <number of machines> launch <cluster-name>
ssh -i cellule.pem root@<your-cluster-master-dns>
spark-ec2/copy-dir ephemeral-hdfs/conf
ephemeral-hdfs/bin/hadoop distcp s3n://celluledecalcul/dataset/raw/train.csv /data/train.csv
scp -i cellule.pem cellule/challenge/target/scala-2.10/target/scala-2.10/challenges_2.10-0.0.jar

cellule/spark/bin/spark-submit \
        --class fr.cc.challenge.Preprocess \
        challenges_2.10-0.0.jar \
        /data/train.csv \
        /data/train2.csv

cellule/spark/bin/spark-submit \
       --class fr.cc.sparktest.LogisticRegression \
       challenges_2.10-0.0.jar \
       /data/train2.csv
```

$\Rightarrow$ Logistic regression for arbitrary large dataset!

Doing Data Science

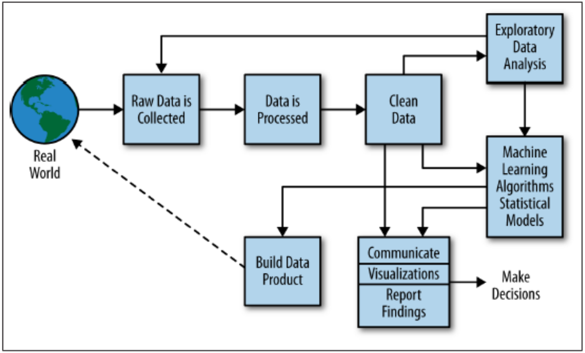
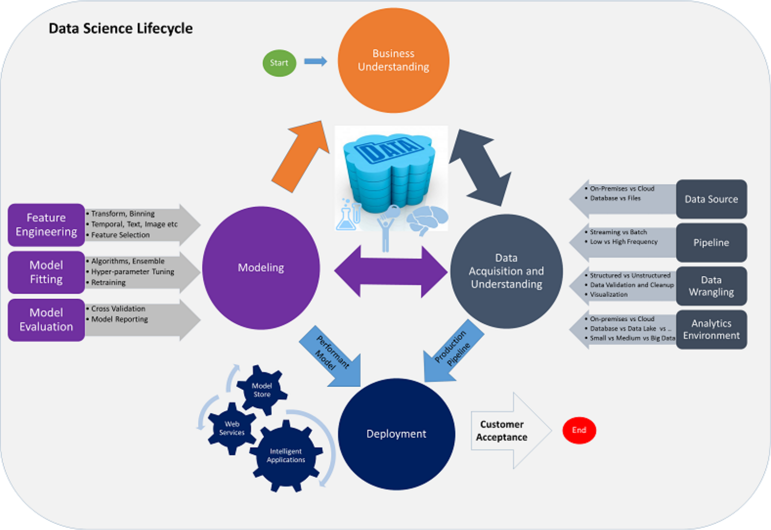


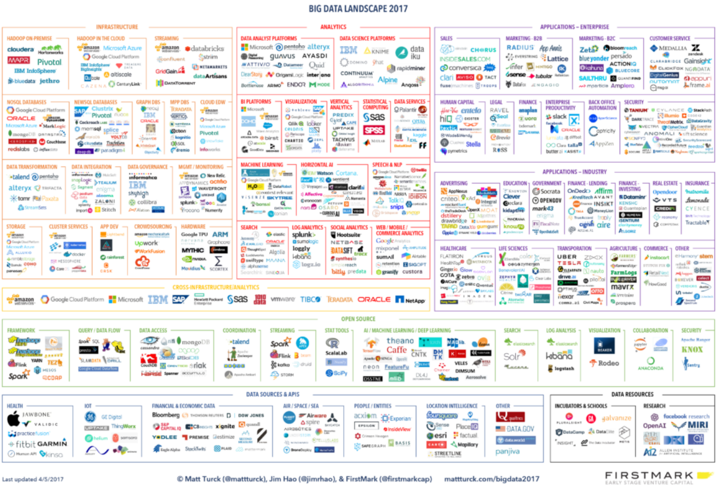
Figure 2-2. The data science process

- Doing Data Science: Straight talk from the frontline.
  - Rachel Schutt, Cathy O'Neil
  - O'Reilly

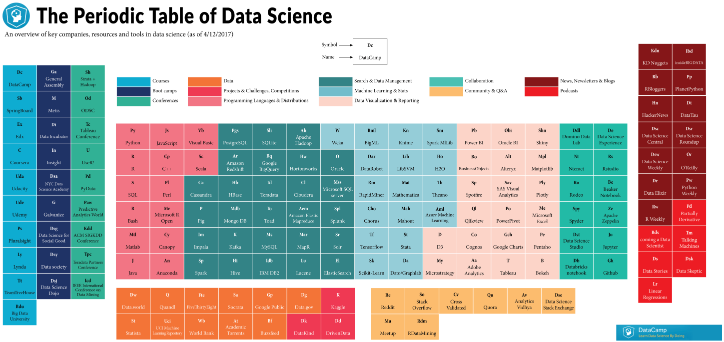
Data Science Is (Quite) Complex!



Data Science Is (Quite) Complex!



Data Science Is (Quite) Complex!



Outline

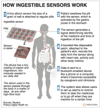
New Interdisciplinary Challenges

- Applied math **AND** Computer science
- Huge importance of domain specific knowledge: physics, signal processing, biology, health, marketing, environmental science...

Some joint math/CS/domain challenges

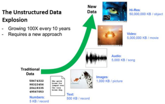
- Data acquisition
- Unstructured data and their representation
- Huge dataset and computation
- Visualization
- Software(s)
- Domain specific issue!

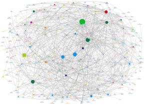
Some Challenges



Data Acquisition

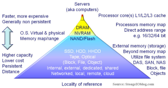
- How to measure new things?
- How to choose what to measure?
- How to deal with distributed sensors?
- How to look for new sources of informations?

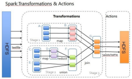


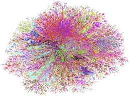


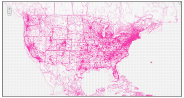
Unstructured Data

- How to store efficiently the data?
- How to describe (model) them to be able to process them?
- How to combine data of different nature?







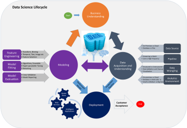


Huge Dataset

- How to take into account the locality of the data?
- How to construct distributed architectures?
- How to design adapted algorithms?

Visualization

- How to look at the data?
- How to present results?
- How to help taking better informed decision?



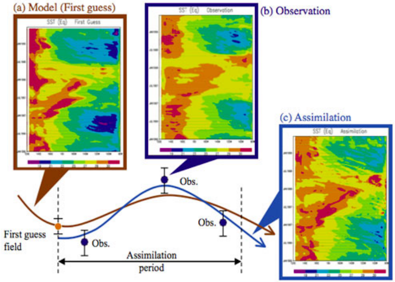
Software(s)

- How to construct a consistent ecosystem?
- How to construct interoperable systems?

Domain Specific Knowledge

- How to find the real problem at hand?
- How to incorporate human expertise?
- How to measure the performance?

Some Challenges

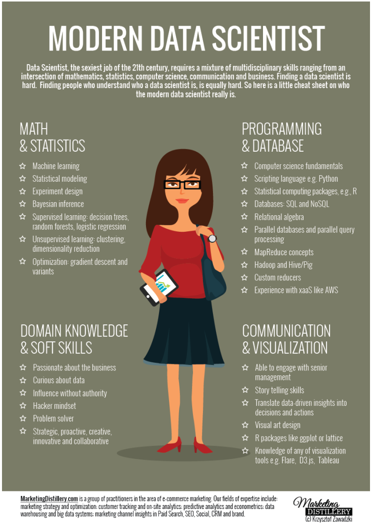


Environment Science

- Data/Model coupling.
- Multiscale modeling / Multimodal modeling.
- Long term/short term prediction.
- Prediction vs understanding.

Outline

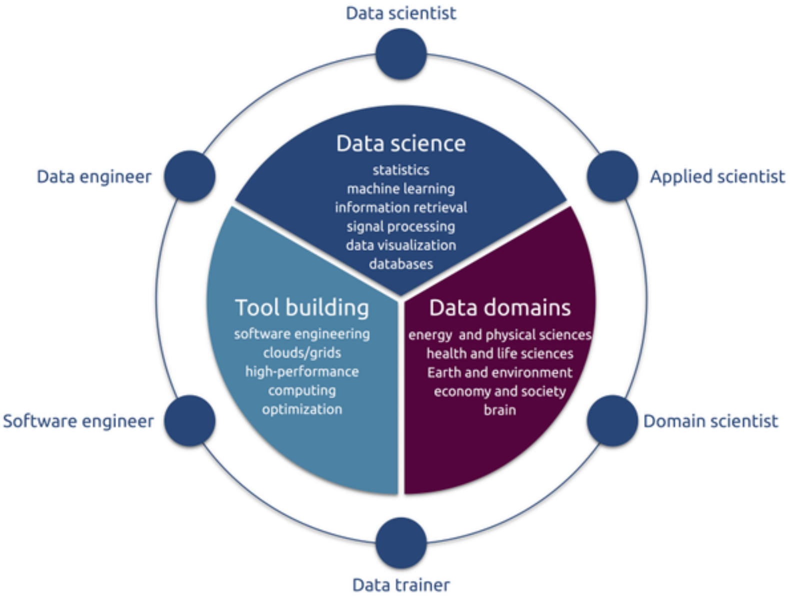
Skills



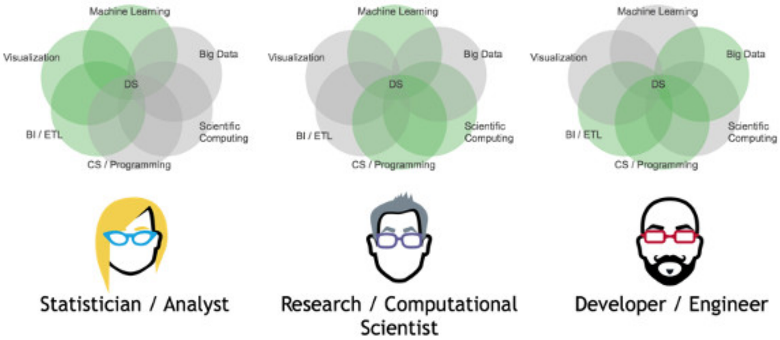
Data Scientist

- **Mix** of various skills.
- **Hard to be an expert of everything!**

More Than One Type Of Data Scientist!



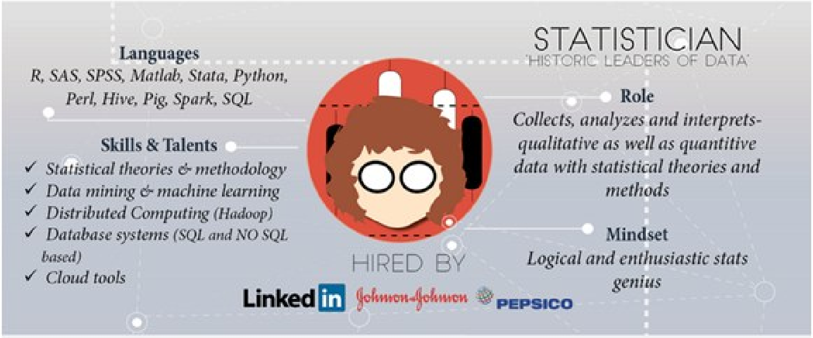
Data Science Team



- Importance of balanced **teams**.

Outline

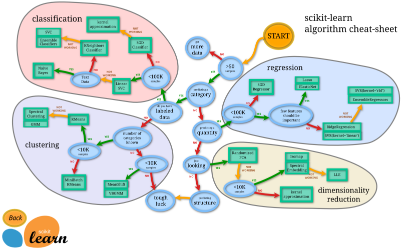
Statistician Point of View

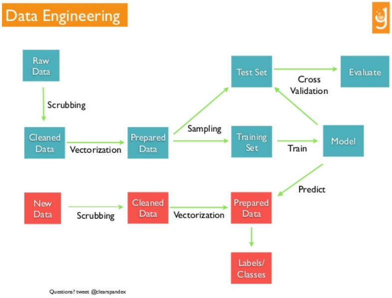


Disclaimer

- I'm a statistician with a signal processing background... posing as a data scientist.
- Not that different in the end...

ML in Practice



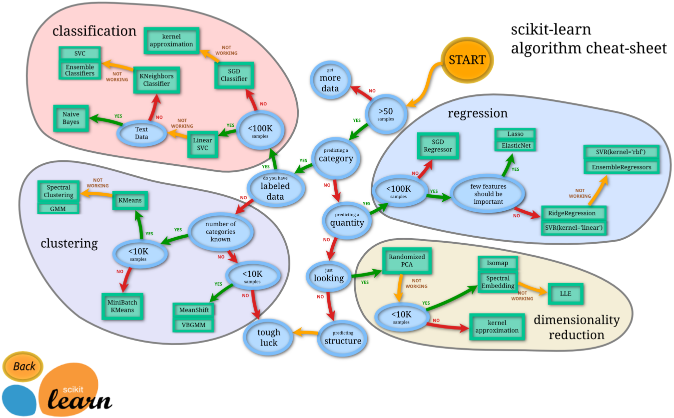


Practical ML

- Build models.
- Test and compare them.
- Use the *best* one...

- No uniformly better methods!
- Mathematical justification...

Machine Learning



Several methodologies

- Lots of methods...
- Only two main principles...

Supervised Learning

Experience, Task and Performance measure

- **Training data** : $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- **Predictor**: $f : \mathcal{X} \to \mathcal{Y}$ measurable
- **Cost/Loss function** : $\ell(f(\mathbf{X}), Y)$ measure how well $f(\mathbf{X})$ "predicts" Y
- **Risk**:

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{X}))\right]\right]$$

- Often $\ell(f(\mathbf{X}), Y) = \mathbf{1}_{Y \neq f(\mathbf{X})}$ or $\ell(f(\mathbf{X}), Y) = |f(\mathbf{X}) - Y|^2$

Goal

- Learn a rule to construct a **classifier** $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. **the risk** $\mathcal{R}(\widehat{f})$ is **small on average** or with high probability with respect to $\mathcal{D}_n$.

Best Solution

- The best solution f^* (which is independent of $\mathcal{D}_n$) is

$$f^* = \arg\min_{f \in \mathcal{F}} R(f) = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{x}))\right]\right]$$

Bayes Classifier (explicit solution)

- In binary classification with $0-1$ loss:

$$f^*(\mathbf{X}) = \begin{cases} +1 & \text{if} \quad \mathbb{P}\left(Y = +1|\mathbf{X}\right) \geq \mathbb{P}\left(Y = -1|\mathbf{X}\right) \\ & \quad \Leftrightarrow \mathbb{P}\left(Y = +1|\mathbf{X}\right) \geq 1/2 \\ -1 & \text{otherwise} \end{cases}$$

- In regression with the quadratic loss

$$f^*(\mathbf{X}) = \mathbb{E}\left[Y|\mathbf{X}\right]$$

Issue: Explicit solution requires to **know** $\mathbb{E}\left[Y|\mathbf{X}\right]$ for all values of $\mathbf{X}$!

Goal

Machine Learning

- Learn a rule to construct a **classifier** $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. **the risk** $\mathcal{R}(\widehat{f})$ is **small on average** or with high probability with respect to $\mathcal{D}_n$.

Canonical example: Empirical Risk Minimizer

- Restrict f to a subset of functions $\mathcal{S} = \{f_\theta, \theta \in \Theta\}$
- Replace the minimization of the average loss by the minimization of the empirical loss

$$\widehat{f} = f_{\widehat{\theta}} = \underset{f_\theta, \theta \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(\mathbf{X}_i))$$

- Examples:
  - Linear regression
  - Linear discrimination with
    $$\mathcal{S} = \{\mathbf{x} \mapsto \mathtt{sign}\{\beta^T \mathbf{x} + \beta_0\} \,/ \beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}\}$$

Probality vs Optimization?

How to find a good function f with a *small* risk

$$R(f) = \mathbb{E}\left[\ell(Y, f(X))\right] \quad ?$$

Canonical approach: $\widehat{f}_{\mathcal{S}} = \operatorname{argmin}_{f \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(\mathbf{X}_i))$

Problems

- How to choose $\mathcal{S}$?
- How to compute the minimization?

A Probabilistic Point of View

Solution: For $\mathbf{X}$, estimate $Y|\mathbf{X}$ plug this estimate in the Bayes classifier: **(Generalized) Linear Models, Kernel methods, k-nn, Naive Bayes, Tree, Bagging...**

An Optimization Point of View

Solution: If necessary replace the loss ℓ by an upper bound ℓ' and minimize the empirical loss: **SVR, SVM, Neural Network,Tree, Boosting**

Probabilistic Approach

- If $Y|\mathbf{X}$ is known, one can compute the best solution f^*

$$\arg\min_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{X}} \left[\mathbb{E}_{Y|\mathbf{X}} \left[\ell(Y, f(\mathbf{x})) \right] \right]$$

Bayes Plugin

- **Learning:** Estimation of $Y|x$ and pluging of this estimate in the Bayes classifier
- **Plugin**: a classifier $\widehat{f} : \mathcal{X} \to \mathcal{Y}$
  - $\ell^{0/1}$ loss:
  $$\hat{f}(\mathbf{x}) = \begin{cases} +1 & \text{if } \widehat{p}_{+1}(\mathbf{x}) \geq \widehat{p}_{-1}(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$
  - Quadratic loss:
  $$\hat{f}(\mathbf{x}) = \mathbb{E}\left[Y|\mathbf{x}\right]$$

- **Instantiations**:
  - Generative Modeling and Bayesian Methods
  - Parametric Conditional Models
  - Kernel Conditional Density Methods
- Importance of a corresponding efficient **numerical scheme**!

Optimization Approach

- The best solution f^* is the one minimizing
$$f^* = \arg\min R(f) = \arg\min \mathbb{E}\left[\ell(Y, f(X))\right]$$

Empirical Risk Minimization

- Restrict f to a subset of functions $\mathcal{S} = \{f_\theta, \theta \in \Theta\}$
- Replace the minimization of the average loss by the minimization of the empirical loss
$$\widehat{f} = f_{\widehat{\theta}} = \underset{f_\theta, \theta \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

- **Issue:** Minimization may be impossible in practice.
- **Solution:** Replace ℓ by ℓ' a simpler (convex) majorant and **minimize** this upper-bound.
- **Instantiation:** Regression, SVM, Neural Networks...
- Importance of a corresponding efficient **numerical scheme**!

Probabilistic vs Optimization

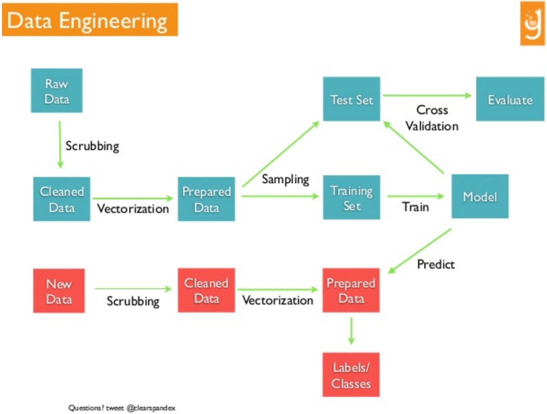
Probabilistic Approach

- **Principle:** estimate the **conditional law** $Y|X$ and use it to take an **informed** decision.
- **Motto:** If you know the world, everything is easy!
- Emphasis on **Interpretation**
- **Pro:**
  - Interpretable models.
  - Lots of flexibility in the generative model.
  - Simultaneous decision optimization.
- **Cons:**
  - Computational issue.
  - No need to know the law to take a decision.

Optimization Approach

- **Principle:** construct a **surrogate decision** criterion and use it to take an **optimized** decision.
- **Motto:** You should focus on your goal!
- Emphasis on **Prediction**
- **Pro:**
  - Focus on the true goal!
  - Can use very clever optimization algorithm.
  - No need to obtain the best solution.
- **Cons:**
  - Black box model.
  - Not robust to a change of decision zone.

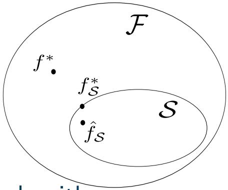
Model Validation



Competition between methods

- Compare methods by their performance...
- on data not used to choose parameters! (Cross Validation)
- Use the best one in the end.

Bias-Variance Dilemna



- General setting:
  - $\mathcal{F} = \{\text{measurable fonctions } \mathcal{X} \to \mathcal{Y}\}$
  - Best solution: $f^* = \text{argmin}_{f \in \mathcal{F}} \mathcal{R}(f)$
  - Class $\mathcal{S} \subset \mathcal{F}$ of functions
  - Ideal target in $\mathcal{S}$: $f^*_{\mathcal{S}} = \text{argmin}_{f \in \mathcal{S}} \mathcal{R}(f)$
  - Estimate in $\mathcal{S}$: $\widehat{f}_{\mathcal{S}}$ obtained with a numerical algorithm

Approximation error and estimation error (Bias/Variance)

$$\mathcal{R}(\widehat{f}_{\mathcal{S}}) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f^*_{\mathcal{S}}) - \mathcal{R}(f^*)}_{\text{Approximation error}} + \underbrace{\mathcal{R}(\widehat{f}_{\mathcal{S}}) - \mathcal{R}(f^*_{\mathcal{S}})}_{\text{Estimation error}}$$

- Different behavior for different model complexity
- **Low complexity model** are easily learned but the approximation error ("bias") may be large (**Under-fit**).
- **High complexity model** may contains a good ideal target but the estimation error ("variance") can be large (**Over-fit**)

Conclusion

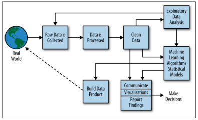
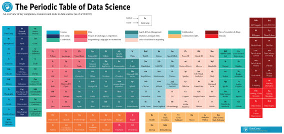
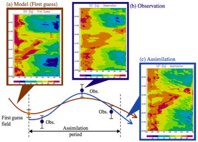
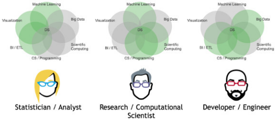
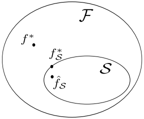


Figure 2-2. The data science process









- Data Science is **not** a new thing.
- Big Data: easier and easier ability to deal with large dataset.
- Environment science: coupling complex modeling and data is the key!
- Importance of collaboration (and team) in Data Science.
- Practical insights can be learned from theory.