WHITENING THE BLACKBOX : WHY AND HOW TO EXPLAIN MACHINE LEARNING PREDICTIONS ?

PyData 2015 / Paris

Christophe Bourguignat (@chris_bour)
Marcin Detyniecki
Bora Eang

DISCLAIMER - WHO ARE WE ?

- We are a new Data team

- We are Python & scikit-learn heavy users and hopefully contributors

- We like tricky problems, doubts, questions

- And love to share with data geeks about that !

kaggle · meetup · datascience.net · PyData Paris 2015

Why explaining Machine Learning models ?

Why explaining Machine Learning models ?

Explain why a given loan application did not meet credit underwriting policy

Why explaining Machine Learning models ?

Explain why a given loan application did not meet credit underwriting policy

Explain why a given transaction is suspicious

Why explaining Machine Learning models ?

Explain** why **a given loan application did not meet credit underwriting policy

Explain** why **a given transaction is suspicious

Explain** why **a given job is recommended for an unemployed

Why explaining Machine Learning models ?

French « Conseil d'Etat » recommendation

*Impose to algorithm-based decisions a **transparency** requirement, on personal data used by the algorithm, and **the general reasoning it followed**. Give the person subject to the decision the possibility of submitting its observations*.

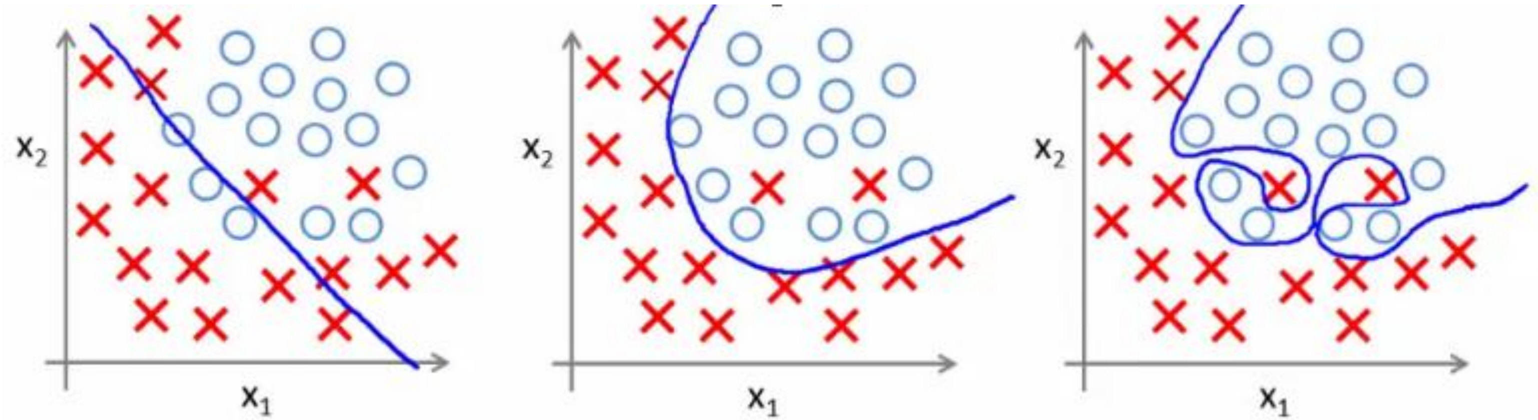
Proposition n° 24 : Imposer aux auteurs de décisions s'appuyant sur la mise en œuvre d'algorithmes une obligation de transparence sur les données personnelles utilisées par l'algorithme et le raisonnement général suivi par celui-ci. Donner à la personne faisant l'objet de la décision la possibilité de faire valoir ses observations.

Vecteur : *loi ou règlement de l'Union européenne.*


ONE DOES NOT SIMPLY
WANT
TO EXPLAIN THE MODEL AT POPULATION
LEVEL
memegenerator.net

What do we *actually* want ?

We don't ask for a "typical profile" of the selected population



We want a reason why an observation got selected by our algorithm

This reason must be simple and understandable (actionable), but can be specific to it

Observation A, next to observation B on our selected population, can be selected for a completely different reason

http://www.holehouse.org/mlclass/07_Regularization.html

Toy Example : Titanic Dataset (1/2)

```
train = pd.read_csv('train.csv')
train.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	NaN	S

```
print train.shape
```

```
(891, 12)
```

Identify most important features. There are 3 of them : Age, Fare, Sex

```
print ['%.2f' % v for v in clf.feature_importances_]
```

```
['0.08', '0.29', '0.04', '0.04', '0.27', '0.25', '0.04']
```

Indeed, surviving rate highly depends on sex

```
print "Male surviving rate : %.2f" % (train[train['Sex'] == 'male']['Survived'].mean())
print "Female surviving rate : %.2f" % (train[train['Sex'] == 'female']['Survived'].mean())
```

```
Male surviving rate : 0.19
Female surviving rate : 0.74
```

Toy Example : Titanic Dataset (2/2)

```
Xpreds[Xpreds['Sex'] == 0].sort(columns='pred', ascending=False).head(5)
```

	Pclass	Age	SibSp	Parch	Fare	Sex	Embarked	truth	pred
8	3	0.75	2	1	19.2583	0	1	1	1
171	1	30.00	0	0	106.4250	0	1	1	1
10	2	32.50	0	0	13.0000	0	3	1	1
81	1	19.00	0	2	26.2833	0	3	1	1
66	2	44.00	1	0	26.0000	0	3	0	1

These women were predicted with a high confidence as « **survived** »

Toy Example : Titanic Dataset (2/2)

```
Xpreds[Xpreds['Sex'] == 0].sort(columns='pred', ascending=False).head(5)
```

	Pclass	Age	SibSp	Parch	Fare	Sex	Embarked	truth	pred
8	3	0.75	2	1	19.2583	0	1	1	1
171	1	30.00	0	0	106.4250	0	1	1	1
10	2	32.50	0	0	13.0000	0	3	1	1
81	1	19.00	0	2	26.2833	0	3	1	1
66	2	44.00	1	0	26.0000	0	3	0	1

These women were predicted with a high confidence as « **survived** »

These women were predicted with as high confidence as « **not survived** »

```
Xpreds[Xpreds['Sex'] == 0].sort(columns='pred', ascending=True).head(5)
```

	Pclass	Age	SibSp	Parch	Fare	Sex	Embarked	truth	pred
111	3	31	0	0	8.6833	0	3	1	0.06
107	3	63	0	0	9.5875	0	3	1	0.06
164	3	15	1	0	14.4542	0	1	1	0.12
162	3	21	2	2	34.3750	0	3	0	0.12
137	3	16	5	2	46.9000	0	3	0	0.14

Toy Example : Titanic Dataset (2/2)

```
Xpreds[Xpreds['Sex'] == 0].sort(columns='pred', ascending=False).head(5)
```

	Pclass	Age	SibSp	Parch	Fare	Sex	Embarked	truth	pred
8	3	0.75	2	1	19.2583	0	1	1	1
171	1	30.00	0	0	106.4250	0	1	1	1
10	2	32.50	0	0	13.0000	0	3	1	1
81	1	19.00	0	2	26.2833	0	3	1	1
66	2	44.00	1	0	26.0000	0	3	0	1

These women were predicted with a high confidence as « **survived** »

These women were predicted with as high confidence as « **not survived** »

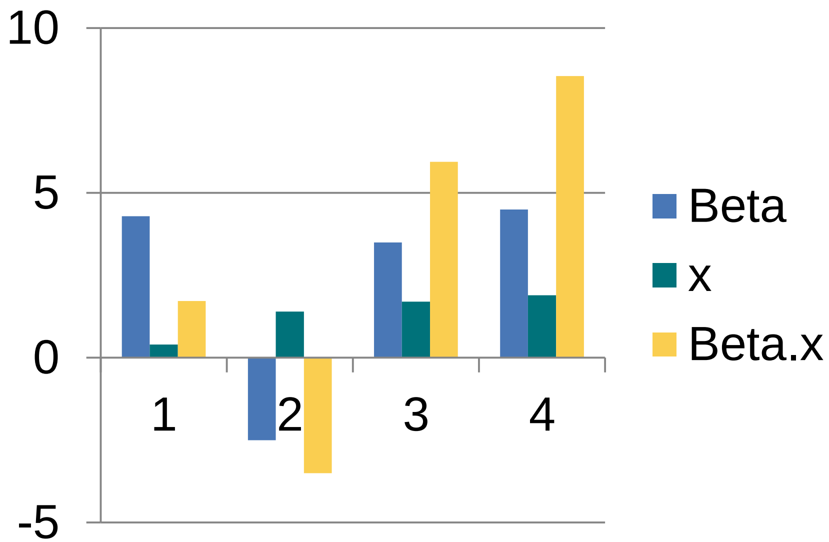
```
Xpreds[Xpreds['Sex'] == 0].sort(columns='pred', ascending=True).head(5)
```

	Pclass	Age	SibSp	Parch	Fare	Sex	Embarked	truth	pred
111	3	31	0	0	8.6833	0	3	1	0.06
107	3	63	0	0	9.5875	0	3	1	0.06
164	3	15	1	0	14.4542	0	1	1	0.12
162	3	21	2	2	34.3750	0	3	0	0.12
137	3	16	5	2	46.9000	0	3	0	0.14

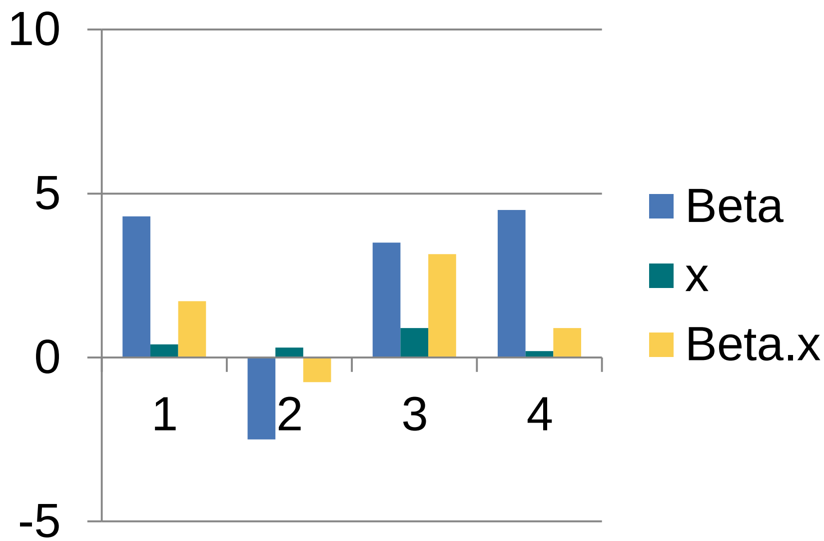
We are looking for a method saying : why ?

A simple case : linear models

$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathrm{T}} \boldsymbol\beta + \varepsilon_i, \qquad i = 1, \ldots, n,$$



Observation 1



Observation 2

"Unfortunately, predictive models that are the most powerful are usually the least interpretable"

http://machinelearningmastery.com/model-prediction-versus-interpretation-in-machine-learning/

A complicated case : Random Forests

- scikit-learn includes the ***.feature_importances_*** attribute …
  - → Implementation from Breiman, Friedman, "*Classification and regression trees*", 1984 (“Gini Importance" or “Mean Decrease Impurity“)
  - → Louppe, 2014 “*Understanding Random Forests*”, PhD dissertation
  - → R package also implements “Mean Decrease Accuracy”

- … but has nothing to show features contribution for a given observation

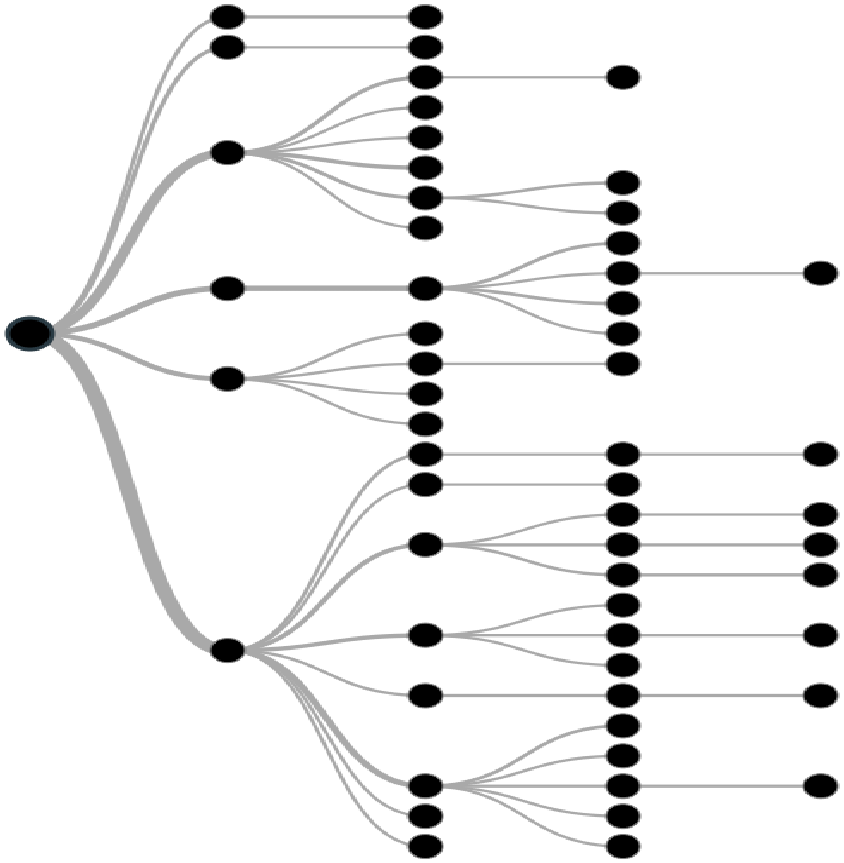
http://ngm.nationalgeographic.com/2012/12/sequoias/quammen-text

How to interpret a forest?

- « What if » explanation
- Sensitivity of the variable (i.e. derivative)
- Feature contribution
  → **« path approach »**

How to interpret a forest?

- « What if » explanation
- Sensitivity of the variable (i.e. derivative)
- Feature contribution
  → **« path approach »**



State Of The Art

http://arxiv.org/abs/1312.1121

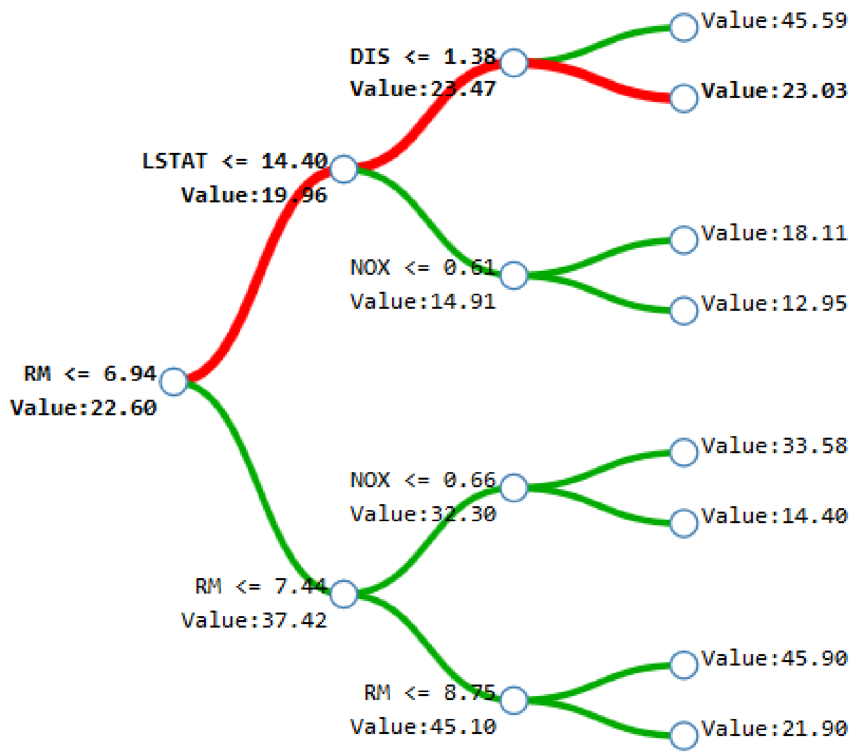
Interpreting random forest classification models using a feature contribution method

Anna Palczewska*[1] and Jan Palczewski†[2] Richard Marchese Robinson‡[3] Daniel Neagu§[1]

[1]Department of Computing, University of Bradford, BD7 1DP Bradford, UK
[2]School of Mathematics, University of Leeds, LS2 9JT Leeds, UK
[3]School of Pharmacy and Biomolecular Sciences, , Liverpool John Moores University, L3 3AF Liverpool, UK

State Of The Art

http://blog.datadive.net/interpreting-random-forests/



Prediction: 23.03 ≈ 22.60 (trainset mean) - 2.64(loss from RM) + 3.52(gain from LSTAT) - 0.44(loss from DIS)

Scikit-Learn IPython demo

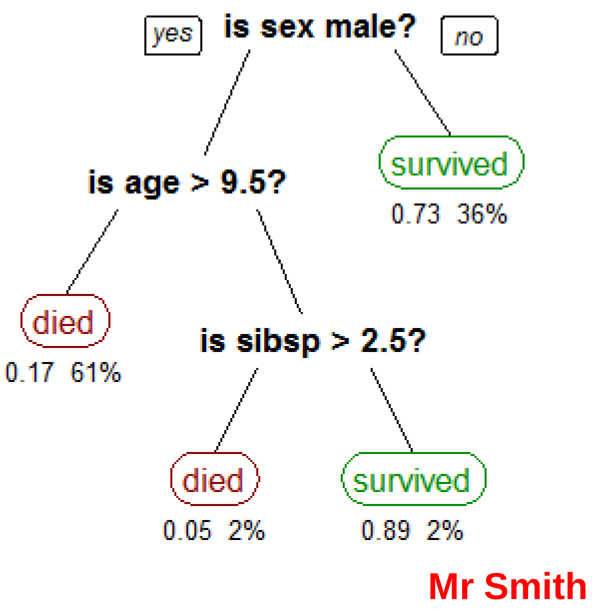
Playing with Titanic Data

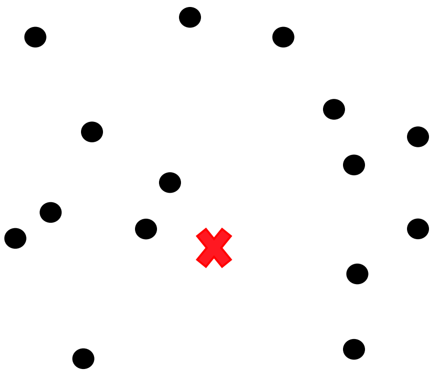
- Traversing the trees and using a trivial metric :
  - +1 when a feature is crossed
  - + impurity when a feature is crossed

- Limitations : Scikit learn stores :
  - **The number of samples for each node (tree_.n_node_samples)**
  - **The breakdown by class (tree_.value), but <u>only for leaves</u>**

What do we need ? (1/2)

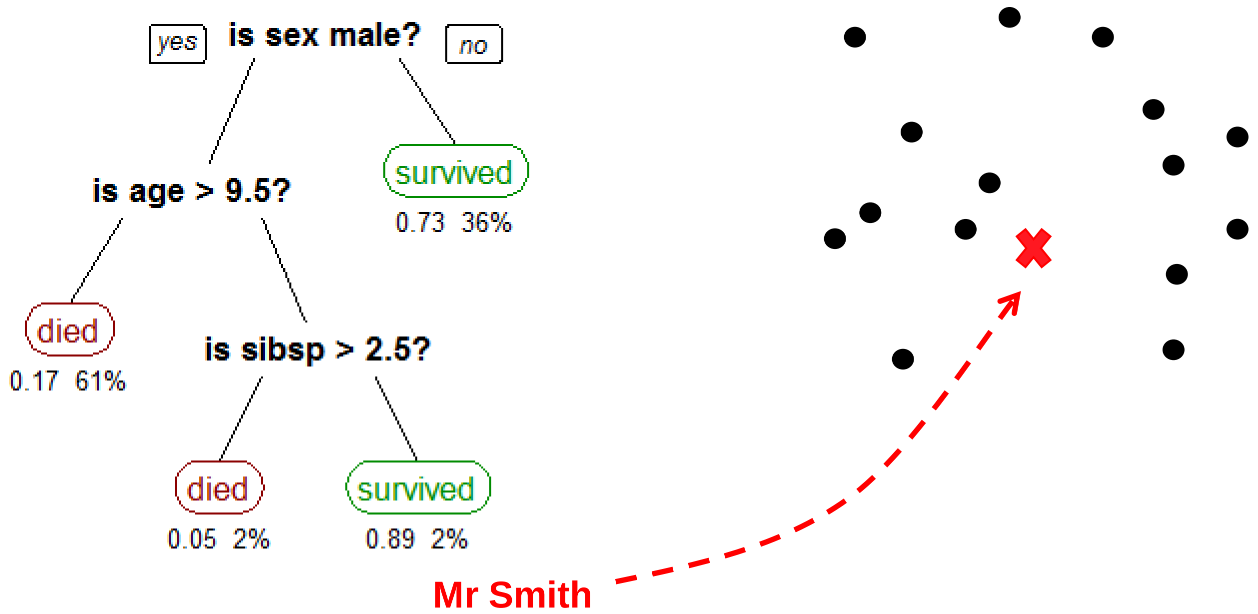
- Be able to compare subsets induced by each tree





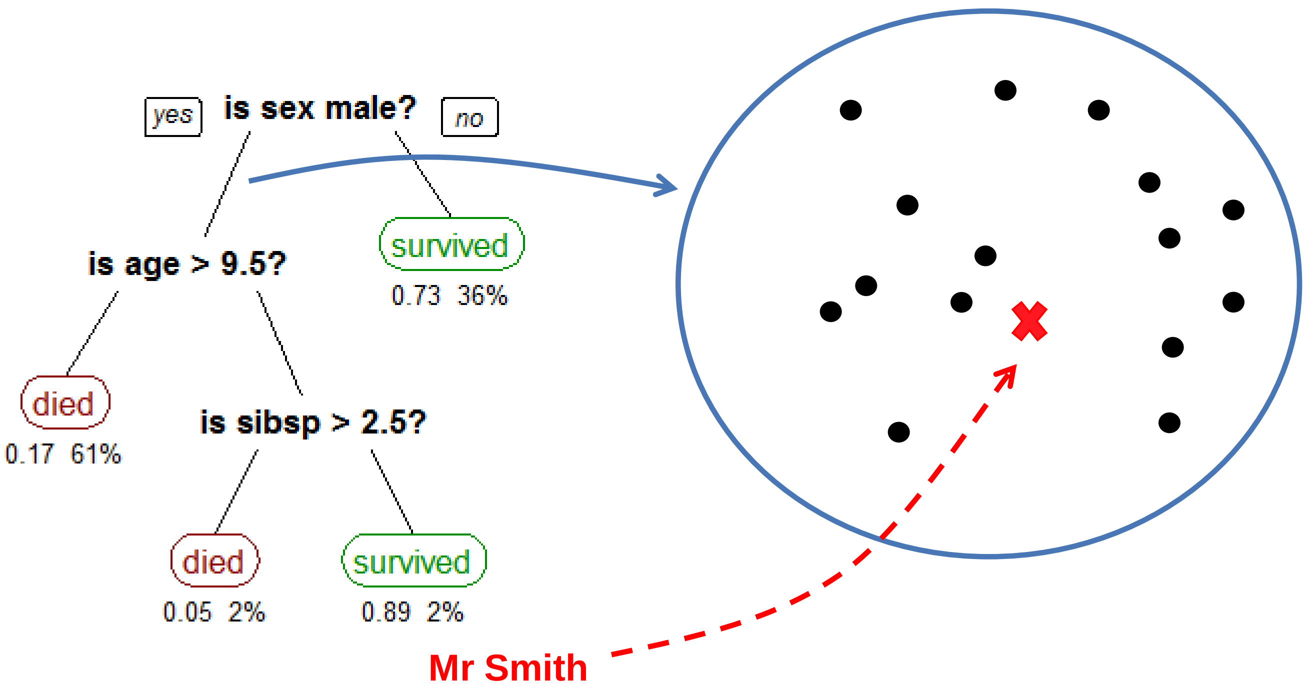
What do we need ? (1/2)

- Be able to compare subsets induced by each tree



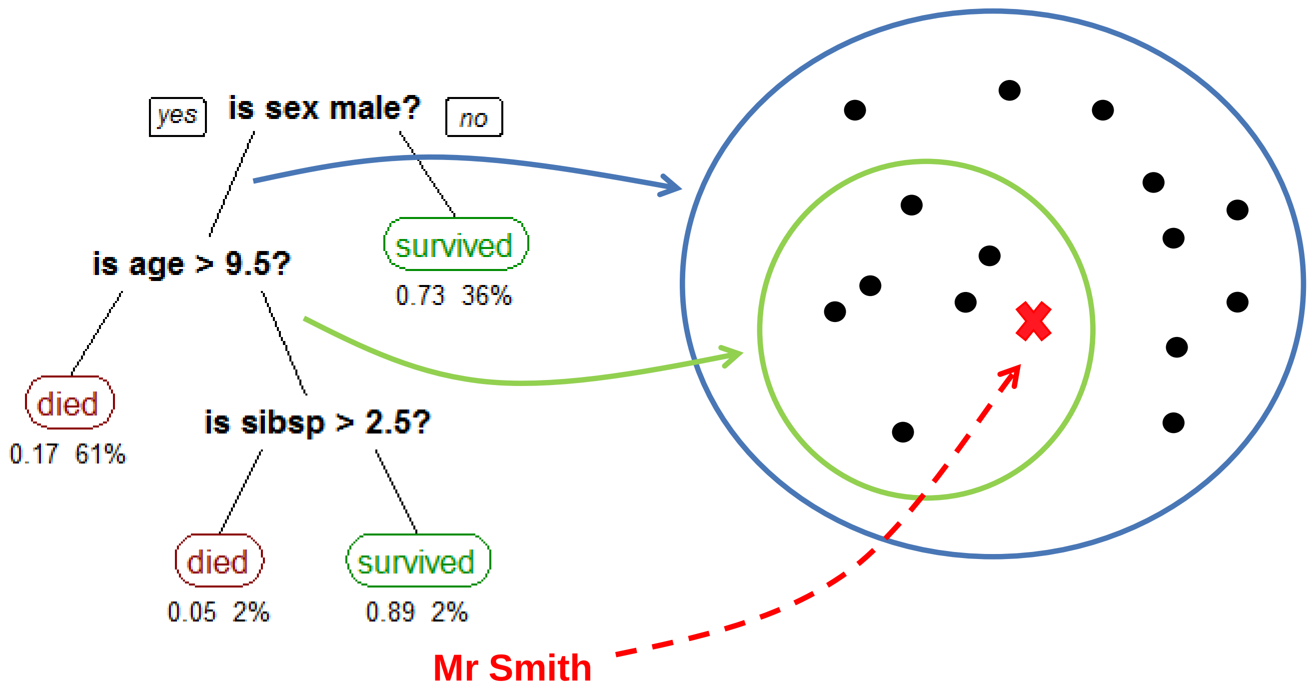
What do we need ? (1/2)

- Be able to compare subsets induced by each tree



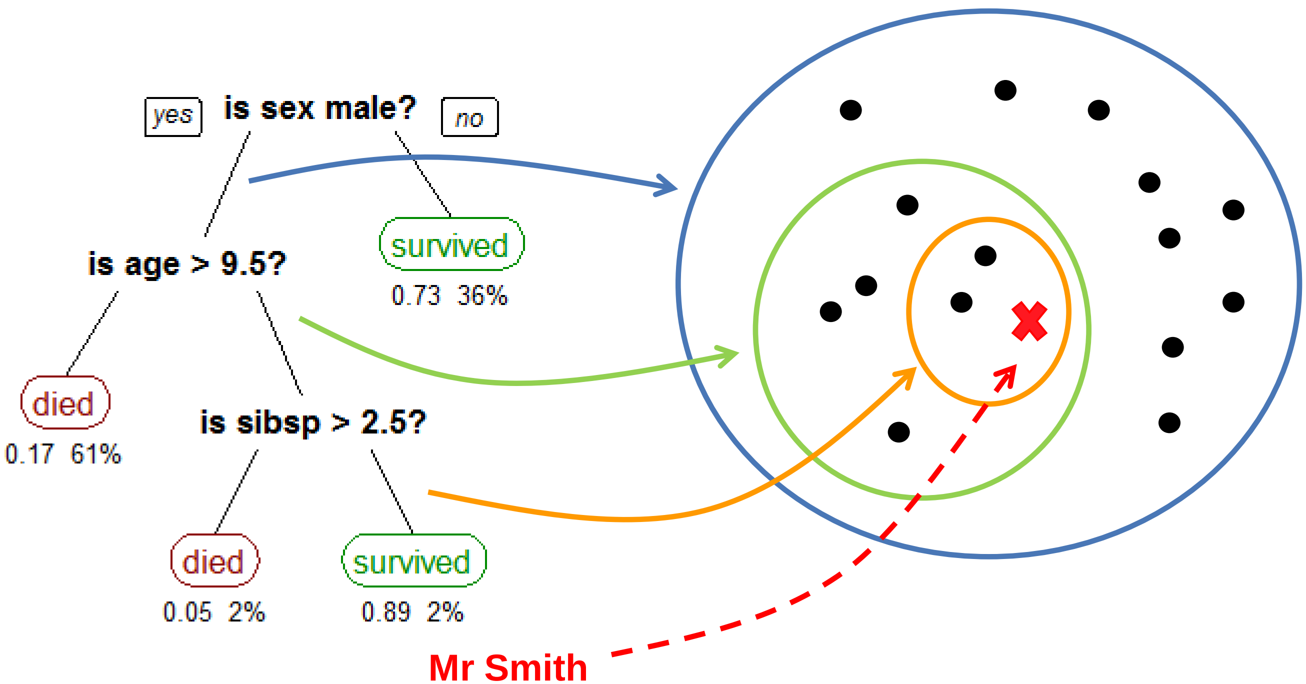
What do we need ? (1/2)

- Be able to compare subsets induced by each tree



What do we need ? (1/2)

- Be able to compare subsets induced by each tree



What do we need (2/2)

- Ideally, we would need easy access to all nodes attributes :
  - **Average_score**
  - **Node_size (absolute or %)**
  - **Number of class 0 samples (absolute or %)**
  - **Number of class 1 samples (absolute or %)**
  - **…**

- For each tree
  - **For each node**
    - *Metric += F(parent_node, node, left_child_node, right_child_node, brother_node)*
    - *E.g : F = parent_node.average_score – node.average_score*

Thank You

and join us, we have many other problems to crack !

Christophe Bourguignat (@chris_bour)
Marcin Detyniecki
Bora Eang