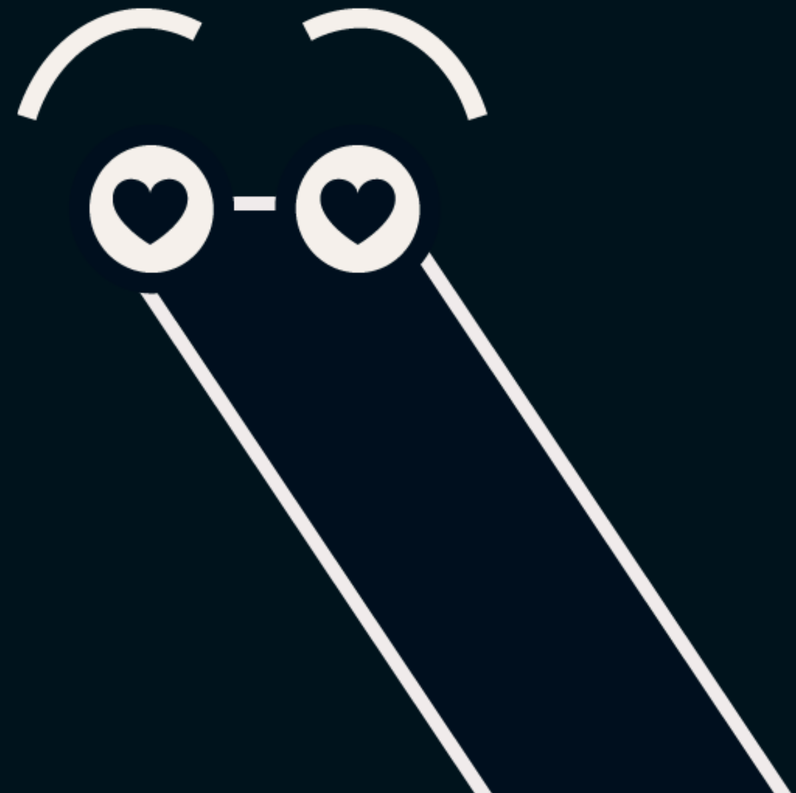
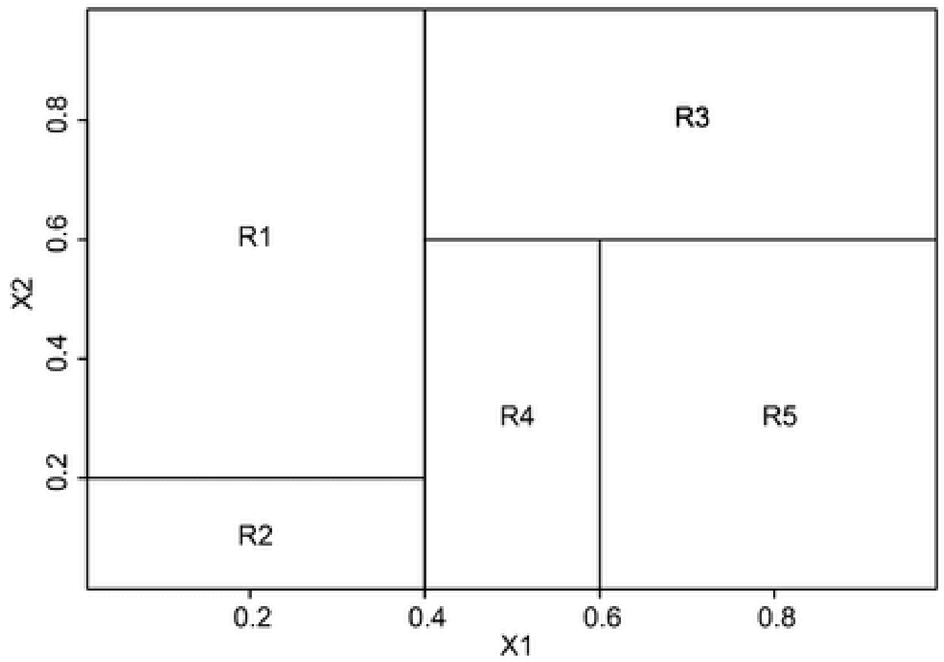
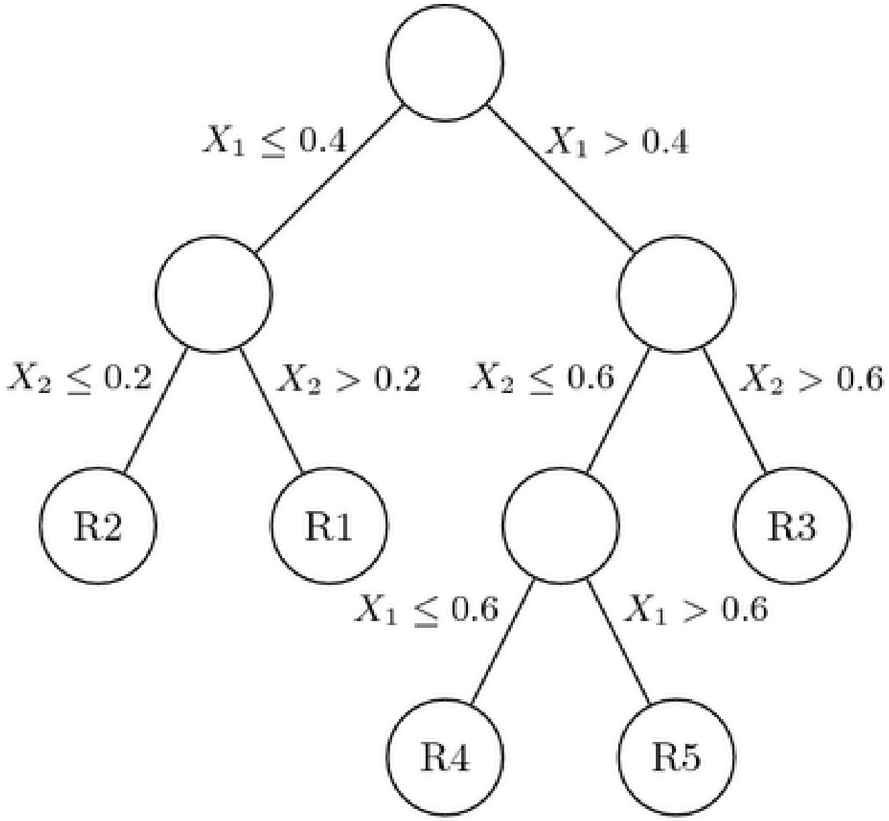
xtream

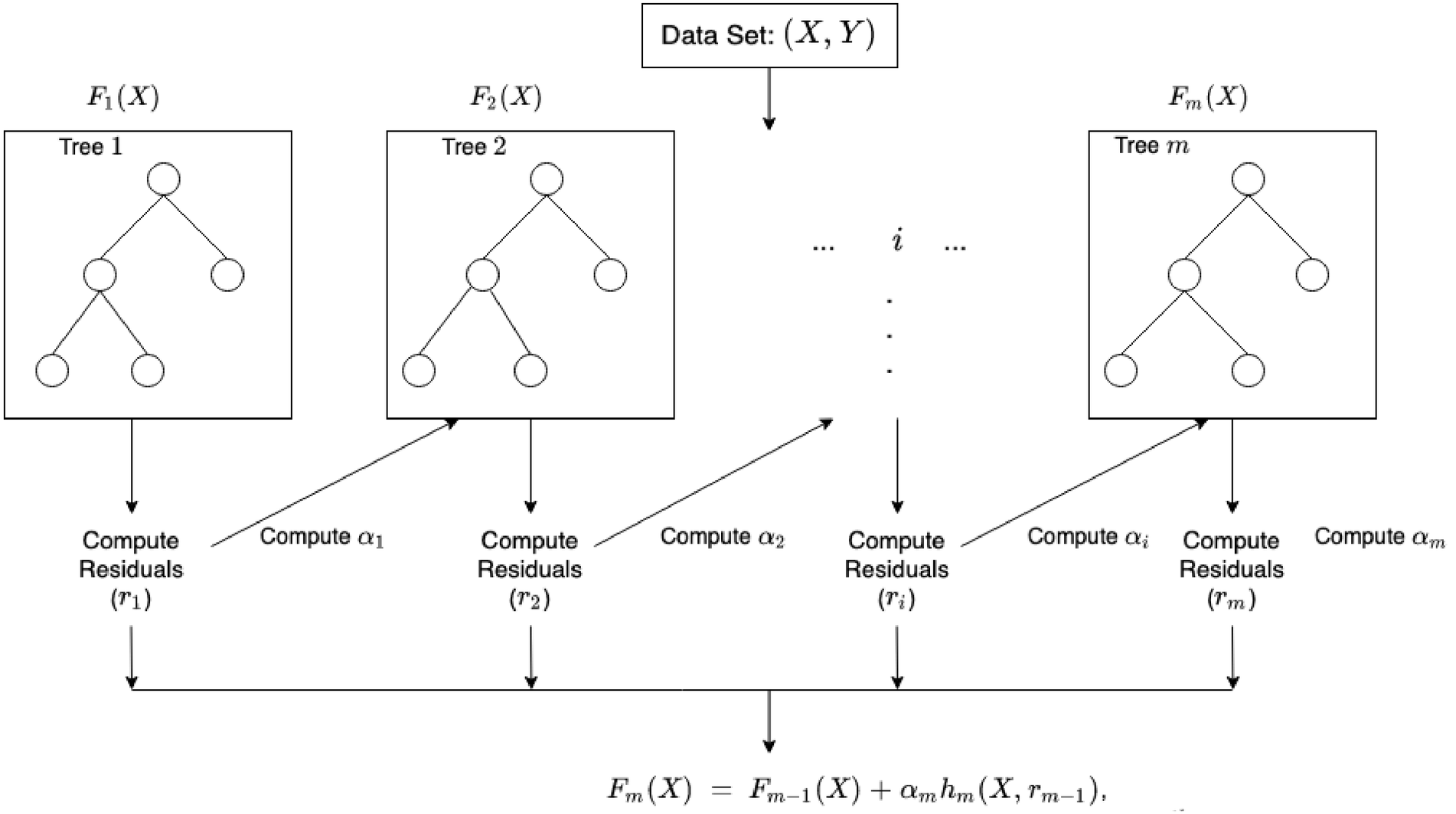
From SHAP to EMB

Emanuele Fabbiani




R1
R2
R3
R4
R5
X1
X2
0.2
0.4
0.6
0.8
0.2
0.4
0.6
0.8


$X_1 \leq 0.4$
$X_1 > 0.4$
$X_2 \leq 0.2$
$X_2 > 0.2$
$X_2 \leq 0.6$
$X_2 > 0.6$
R2
R1
R3
$X_1 \leq 0.6$
$X_1 > 0.6$
R4
R5


Data Set: (X, Y)
$F_1(X)$
$F_2(X)$
$F_m(X)$
Tree 1
Tree 2
Tree m
... i ...
Compute Residuals (r_1)
Compute α_1
Compute Residuals (r_2)
Compute α_2
Compute Residuals (r_i)
Compute α_i
Compute Residuals (r_m)
Compute α_m
$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$

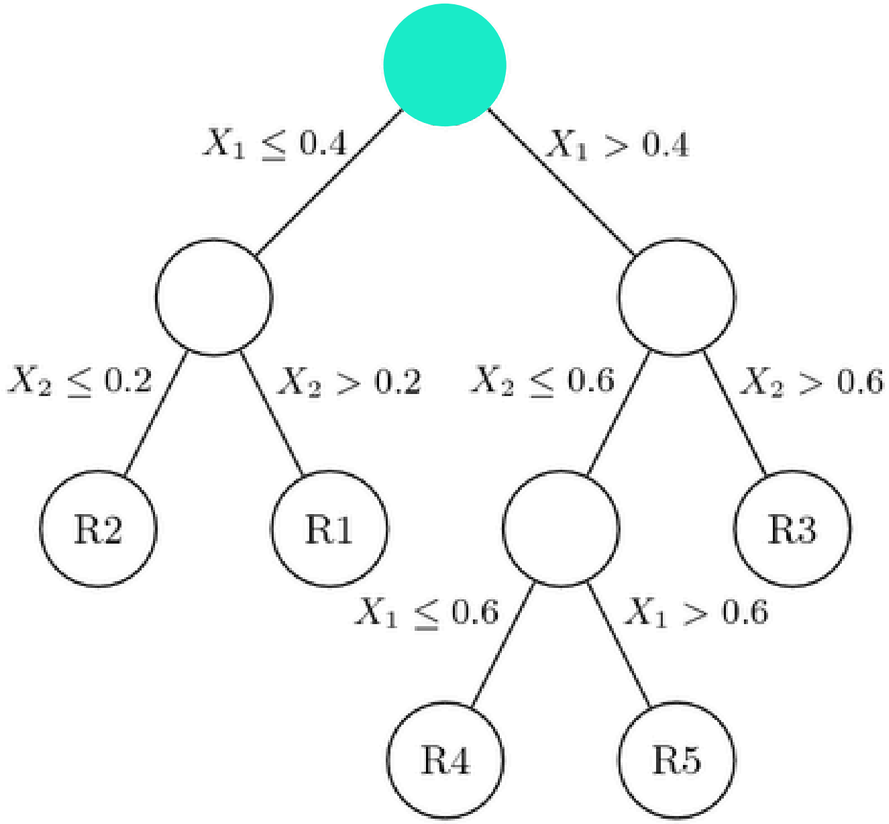
$$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1})$$

$$h_m \sim -\bigtriangledown L$$

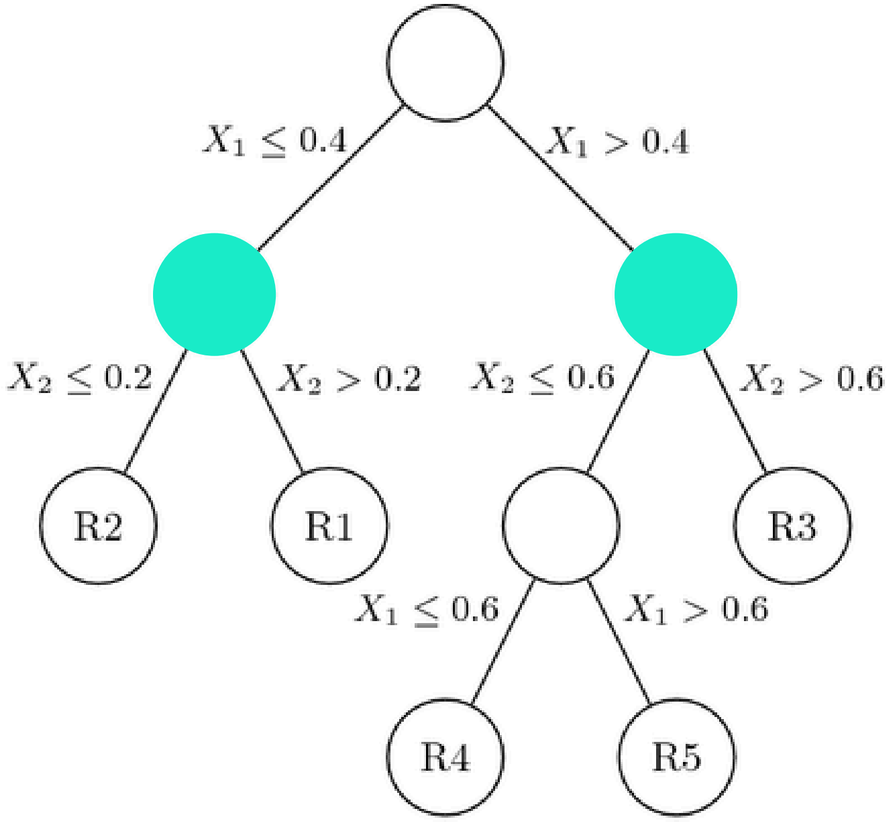
100+

Trees in a Gradient Boosting model

Impurity Metrics



Variance = 100



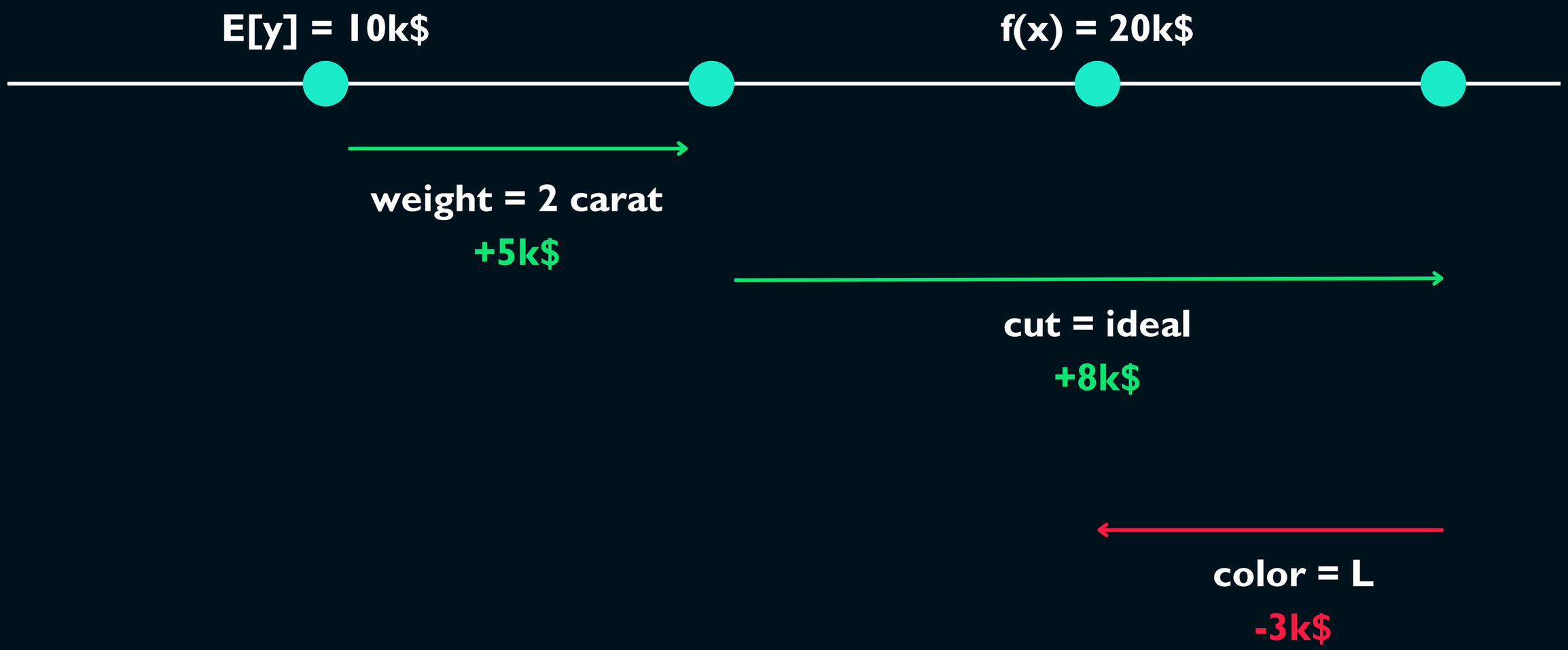
Variance = 100

Variance = 70, 80

MDI for X1 = 25

Favours the variables that produce more splits, that is the ones with higher cardinality.

SHAP


E[y] = 10k$
f(x) = 20k$
weight = 2 carat
+5k$
cut = ideal
+8k$
color = L
-3k$

Local

Can be computed for each sample.

Additive

The effect of each feature sums up with the others.

Shapley Values

How do we fairly distribute money to players who win a game together, based on their individual **contributions**?

Efficiency

All the money must be distributed.

Consistency

If a player contributes more than another, they must get more money.

Theorem

There exists one and only one solution: split the money based on the **average** contribution of each player, over **all possible games** with every subset of players (in any order).

All possible combinations?

How to train the same model **without** some features?

All possible combinations?

Sample!

How to train the same model without some features?

Sample!

All possible combinations?

Sample!

(Efficient algorithms exist for tree and deep learning models.)

How to train the same model without some features?

Sample!

(From some dataset, with all kind of issues.)

SHAP

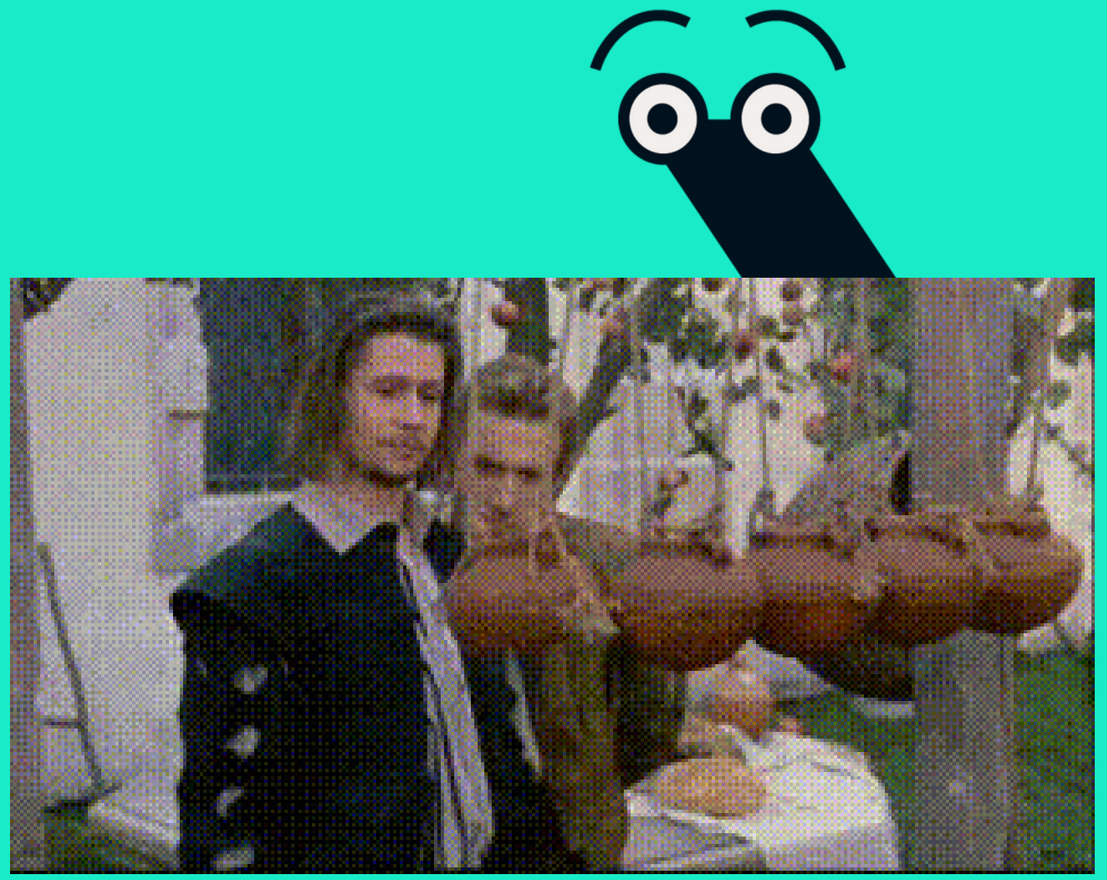
A collection of smart algorithms to approximate Shapley values.

That is, the only fair way of computing feature contribution is black-box models.

EBM


Iteration
Feature 1
Feature 2
Feature 3
1
T(x1)
Residuals
T(x2)
Residuals
T(x3)
2
T(x1)
Residuals
T(x2)
Residuals
T(x3)
...

Live Coding



https://arxiv.org/pdf/1603.02754

https://xgboost.readthedocs.io/en/stable/tutorials/model.html

https://christophm.github.io/interpretable-ml-book/

https://arxiv.org/pdf/1705.07874

https://scikit-learn.org/1.5/modules/tree.html#decision-trees

https://youtu.be/-taOhqkiuIo?si=anI3HVeTi9N46Wbe

https://youtu.be/0yXtdkIL3Xk?si=5AKkI0i7NIU3qGmj

The End

Ego Slide

Emanuele Fabbiani

AI Engineer @ xtream

Professor in AI at Catholic University of Milan

PhD in Applied AI

Speaker at AMLD Lausanne, ODSC London, PyData Berlin, PyCon Florence, PyData Milan, PyData Paris

Lecturer at UniPV, PoliMI, UniCatt, HSLU, Politechnika Wroclawska.