OCTO Technology

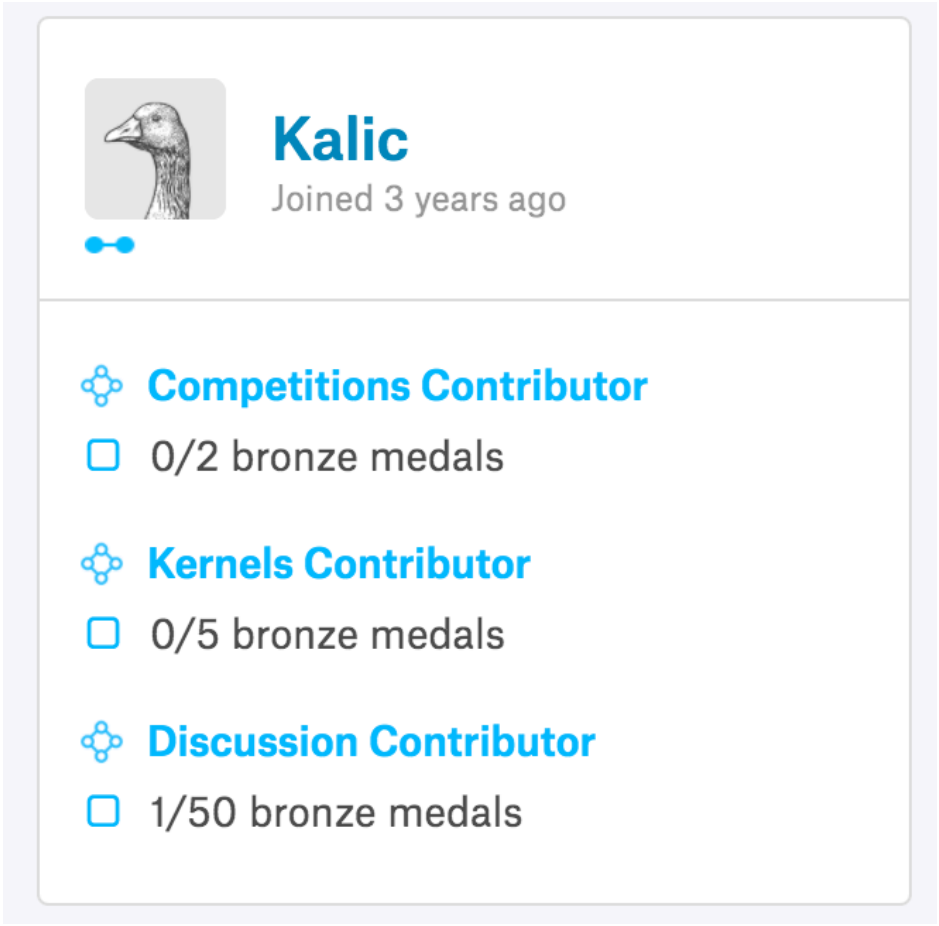
FEATURE IMPORTANCE AND ENSEMBLE METHODS

A new perspective on feature utilization

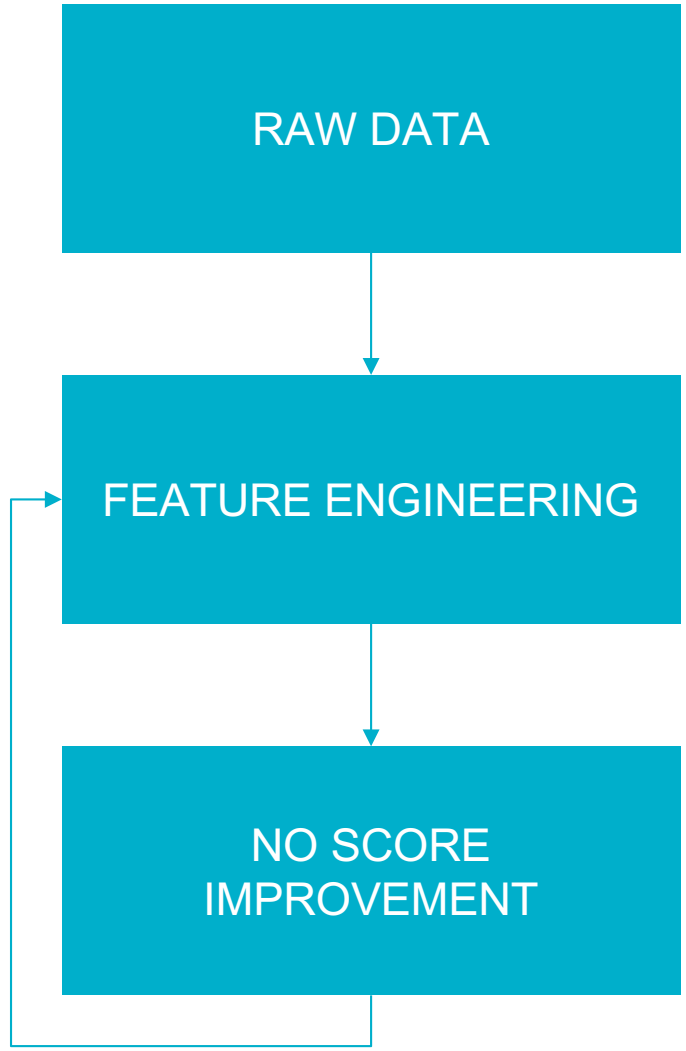
BIG DATA VILNIUS– NOVEMBER 2017 CONSTANT BRIDON

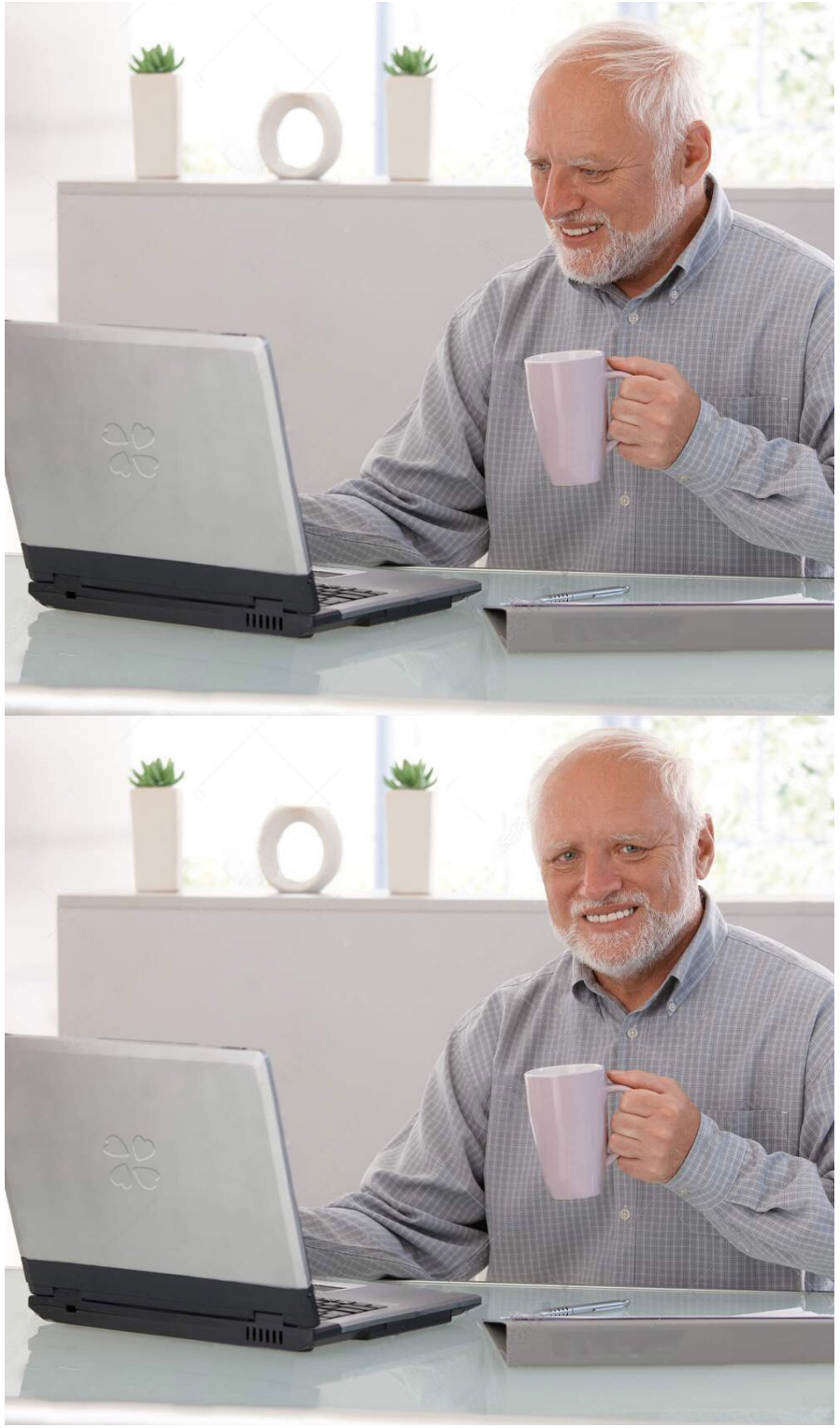
KAGGLE

Home of frustration and despair



FEATURE ENGINEERING IS CRUEL





AGENDA

01 > RECALLS & DEFINITIONS

02 > FEATURE IMPORTANCE & ENSEMBLE METHODS

03 > GRADIENT BOOSTING : TREECEPTION

04 > POINT FEATURE IMPORTANCE ?

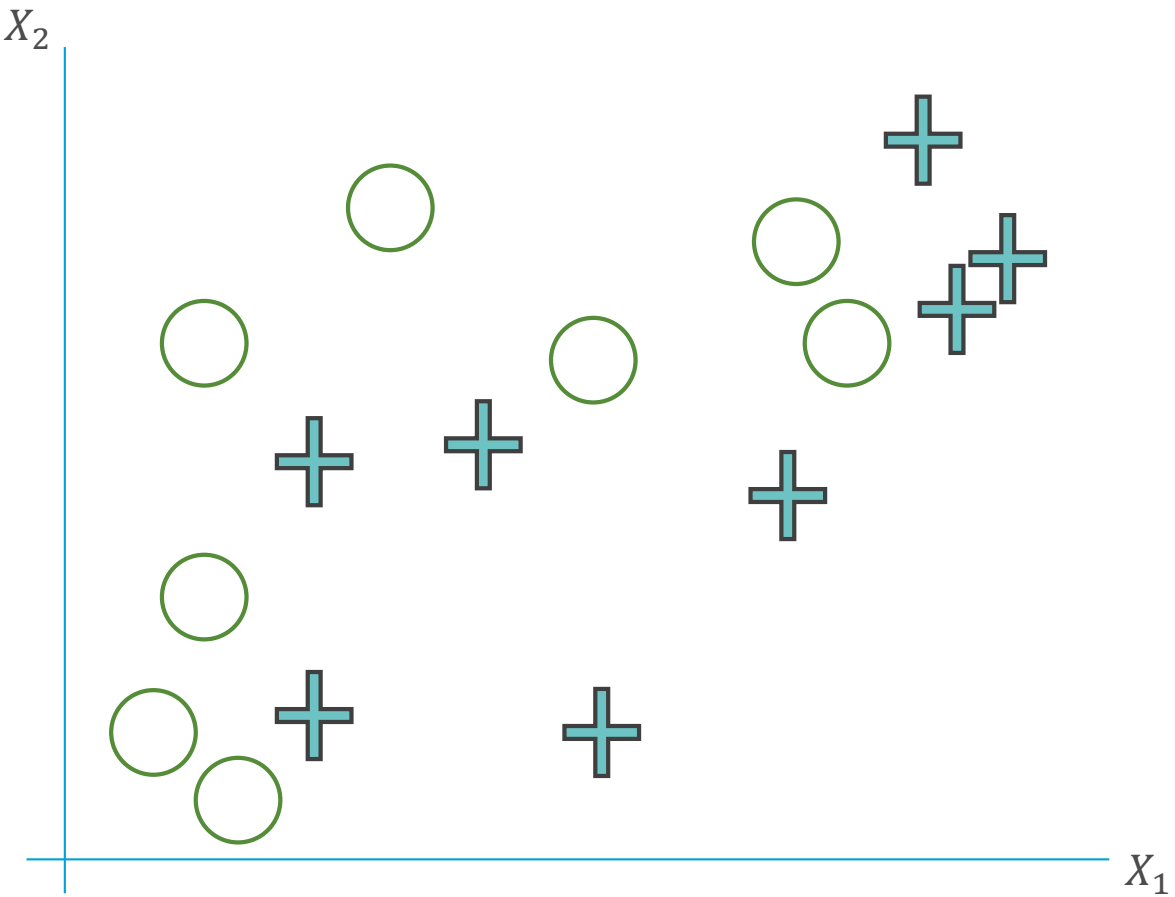
05 > LIMITS & WAYS FORWARD

06 > ANNEXES

01 > RECALLS & DEFINITIONS

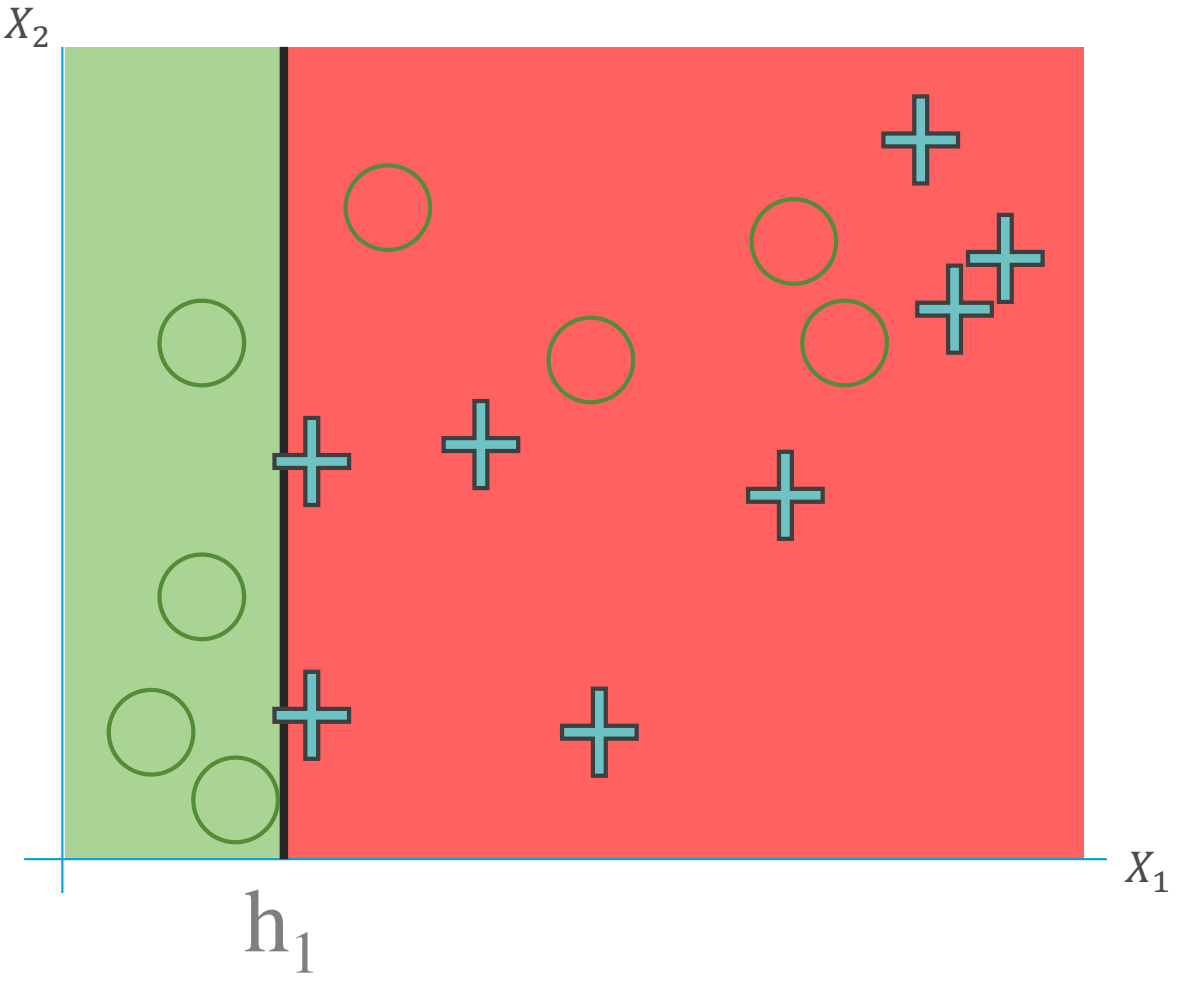
BOOSTING : LEARNING FROM YOUR MISTAKES

Simple binary classification problem using two variables



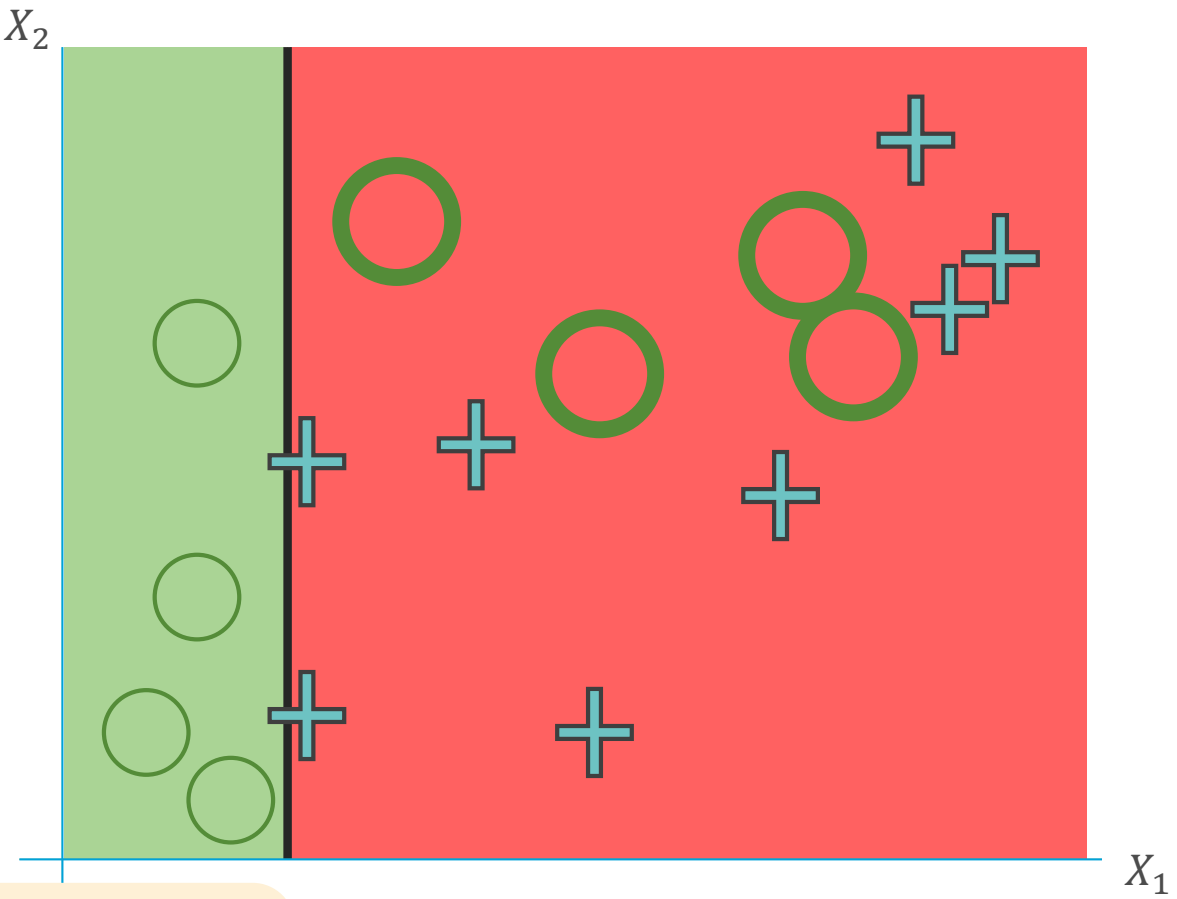
BOOSTING– WEAK LEARNERS – ROUND 1

First weak learner on one variable splits the dataset



BOOSTING– WEAK LEARNERS – ROUND 2

Since first weak learner is too weak, it asks help from second weak learner



I got all the O on the left hand side right. However, I got 4 O wrong on the right hand side, and I highlighted them !

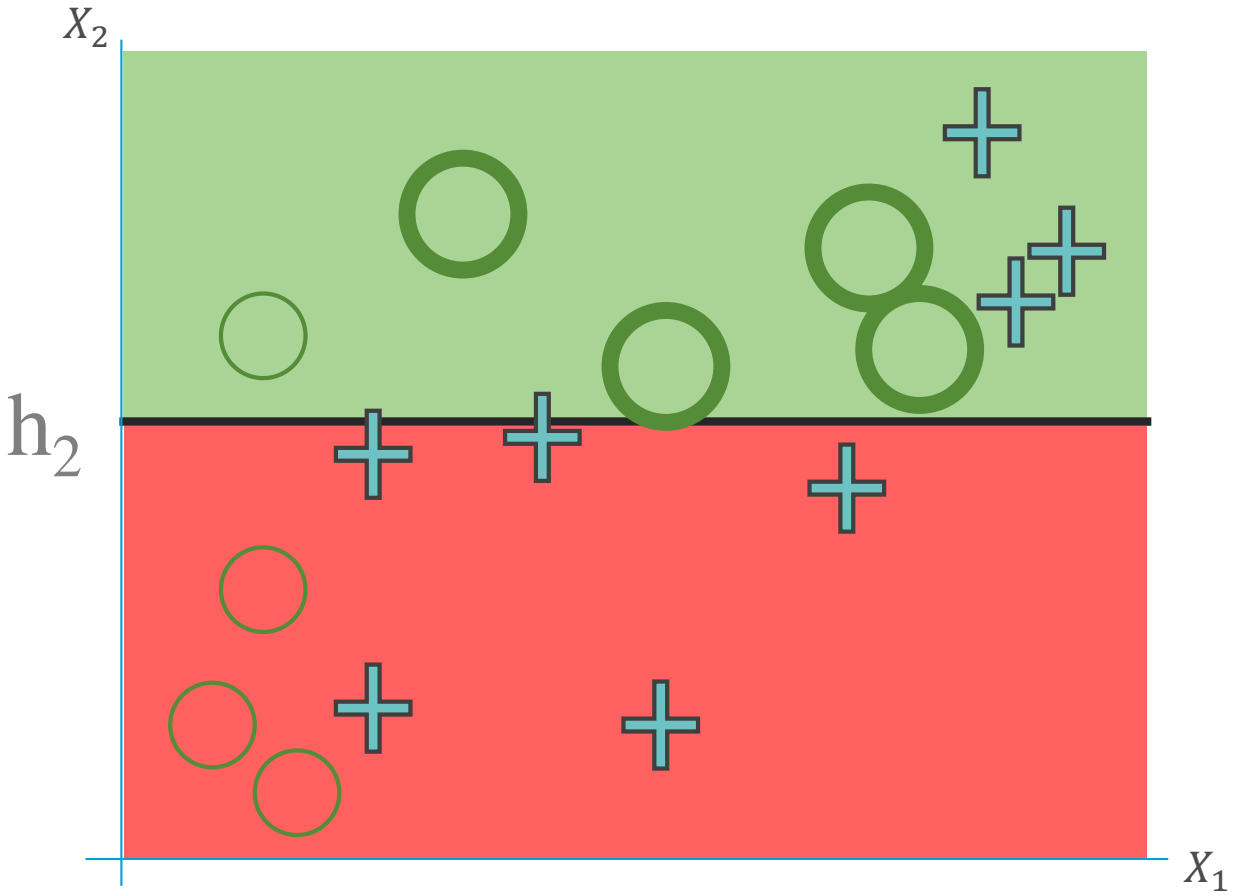
h_1

h_2

Don't worry, I'll handle them !

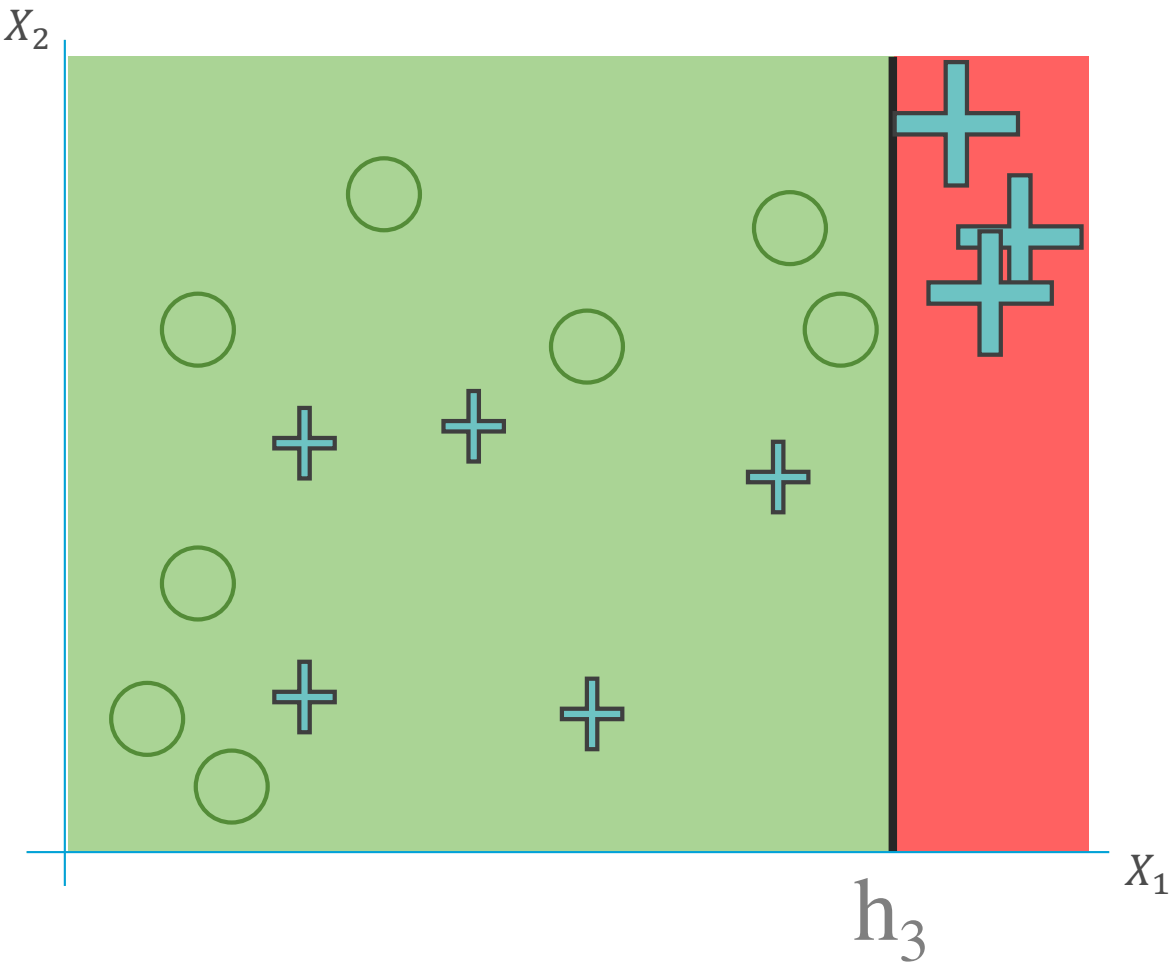
BOOSTING– WEAK LEARNERS – ROUND 2

Second weak learner correctly predicts the mistakes of first weak learner



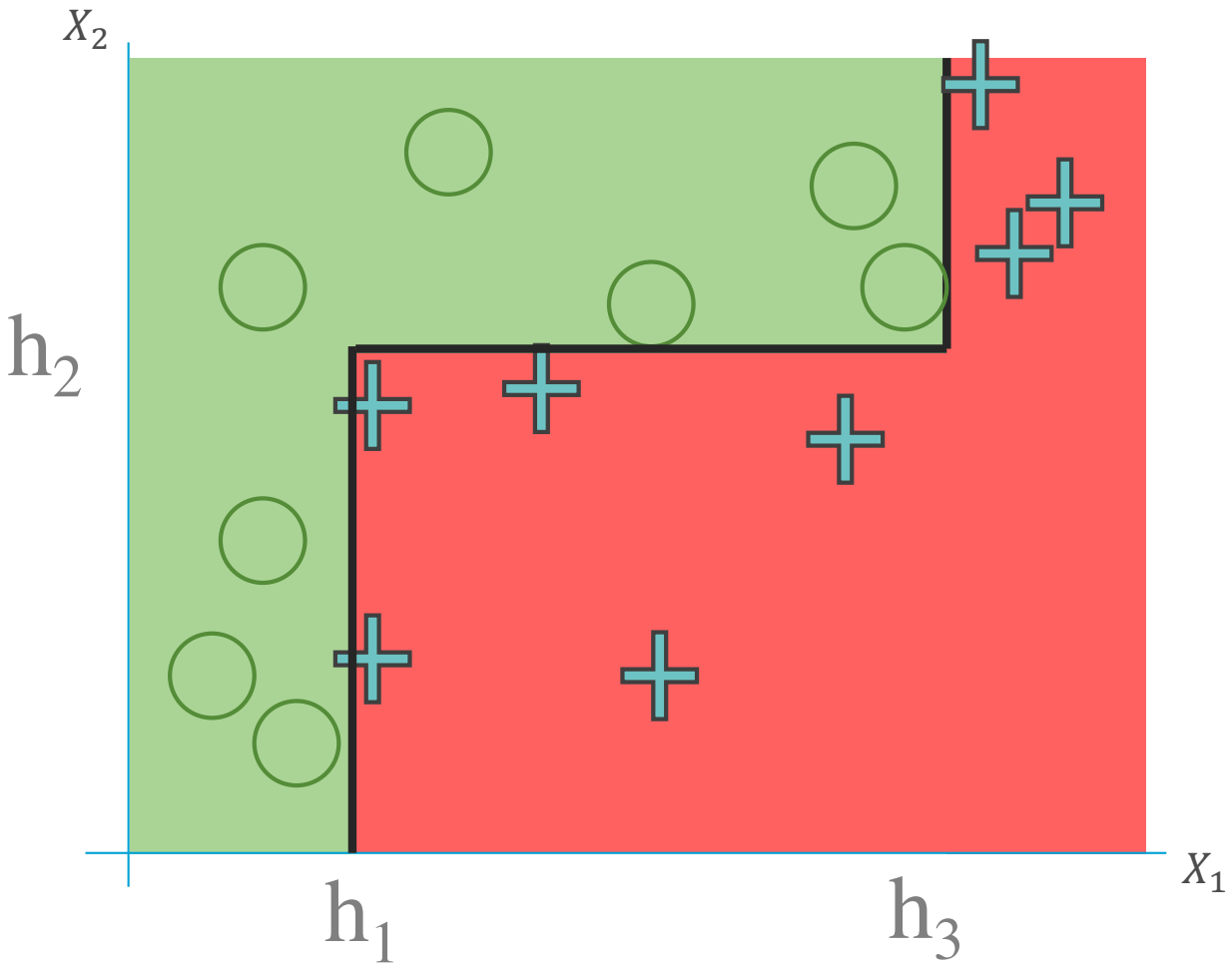
ADABOOST – WEAK LEARNERS – ROUND 3

But second weak learner also needs help…



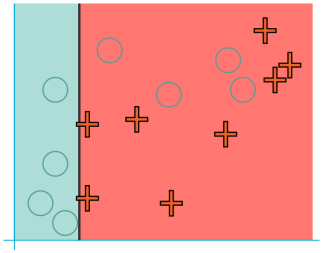
BOOSTING– WEAK LEARNERS – ROUND FINAL

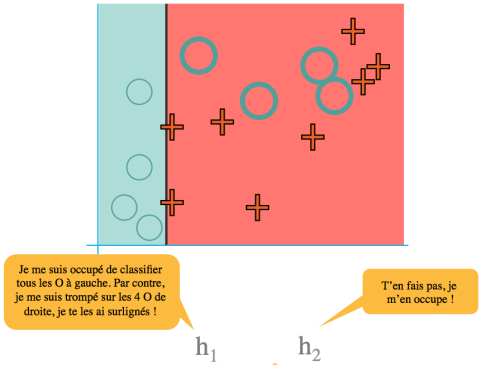
Altogether the three weak learners make a strong learner



$$H = \sum \alpha_i h_i \qquad \alpha_i = \frac{1}{2} \ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$$

GRADIENT BOOSTING





- At the first stage, the overall hypothesis H is equal to h1
- What is the role of h2 ? Make sure that h1 + h2 is equal to y, i.e

$$\forall\, i \in (1, \dots, n) \qquad H(x_i) + h_2(x_i) \approx y_i$$

$$\forall\, i \in (1, \dots, n) \qquad h_2(x_i) \approx y_i - H(x_i)$$

SOLUTION $\quad h_2.fit(X, y - H)$

With,

$$j\big(y, H(x)\big) = \frac{(y - H(x))^2}{2}$$

Then,

$$H(x_i) := H(x_i) - \frac{\partial J}{\partial H(x_i)} \quad \forall\, i \in (1, \dots, n)$$

ROC AUC – A METRIC FOR CLASSIFICATION

Definition of the confusion matrix

		Real value: Positive	Real value: Negative
Prediction	Positive	True Positive	False Positive
Prediction	Negative	False Negative	True Negative
		$TPR = \frac{\sum True\ positive}{\sum Postive\ Condition}$	$FPR = \frac{\sum False\ positive}{\sum NegativeCondition}$

ROC CURVE - A PARAMETRIC CURVE WITH THE THRESHOLD

How to post process your predictions ?

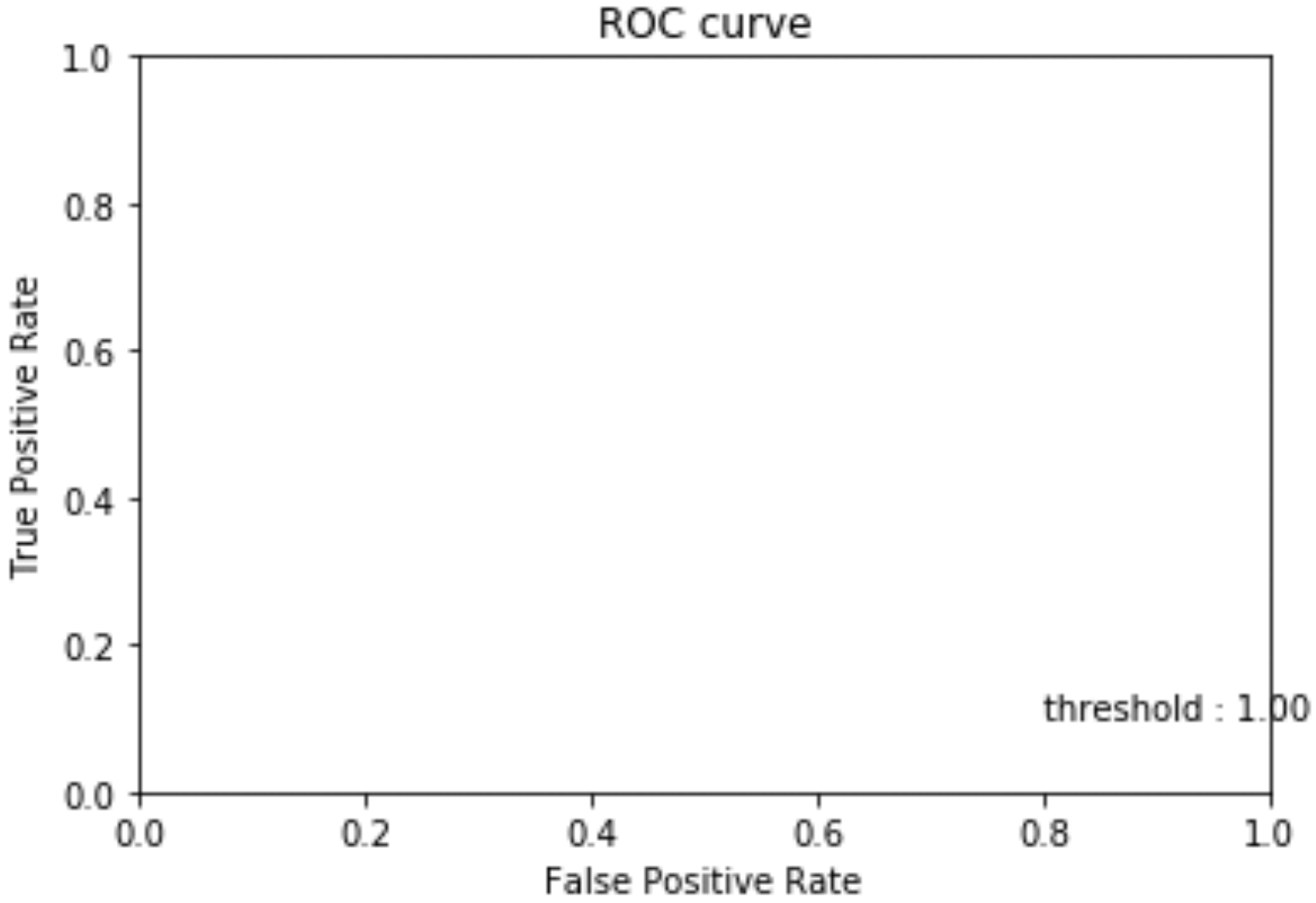
	pred	true
43123	0.229203	0
24080	0.232772	0
70632	0.236211	0
15958	0.239722	0
51413	0.243677	0
61074	0.248302	0
63564	0.253800	0
72346	0.260722	1
44533	0.270871	0
68135	0.280703	0
58454	0.298216	0
13160	0.314317	1
66162	0.333712	1
73577	0.365904	1
44743	0.410899	1
55067	0.469812	1
29718	0.541483	1
71963	0.617832	1
47659	0.713352	1
18209	0.828930	1

61074	0.248302	0
63564	0.253800	0
72346	0.260722	1
44533	0.270871	0
68135	0.280703	0
58454	0.298216	0
13160	0.314317	1
66162	0.333712	1

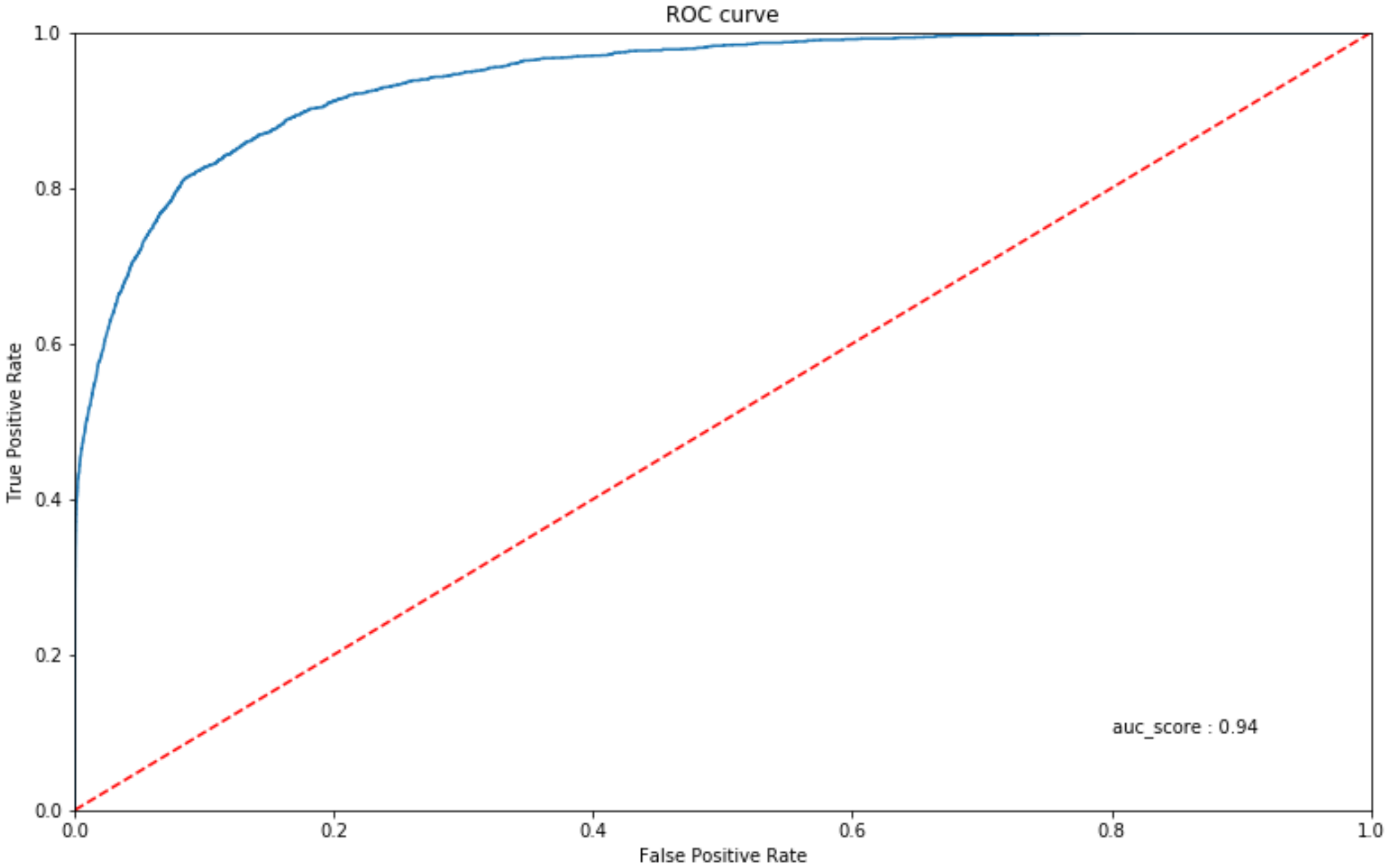
- According to the **threshold on the predicted probability**, I am going to generate False Positive, or False Negative

ROC CURVE - A PARAMETRIC CURVE WITH THE THRESHOLD

We can plot the TPR wrt to the FPR according to the threshold



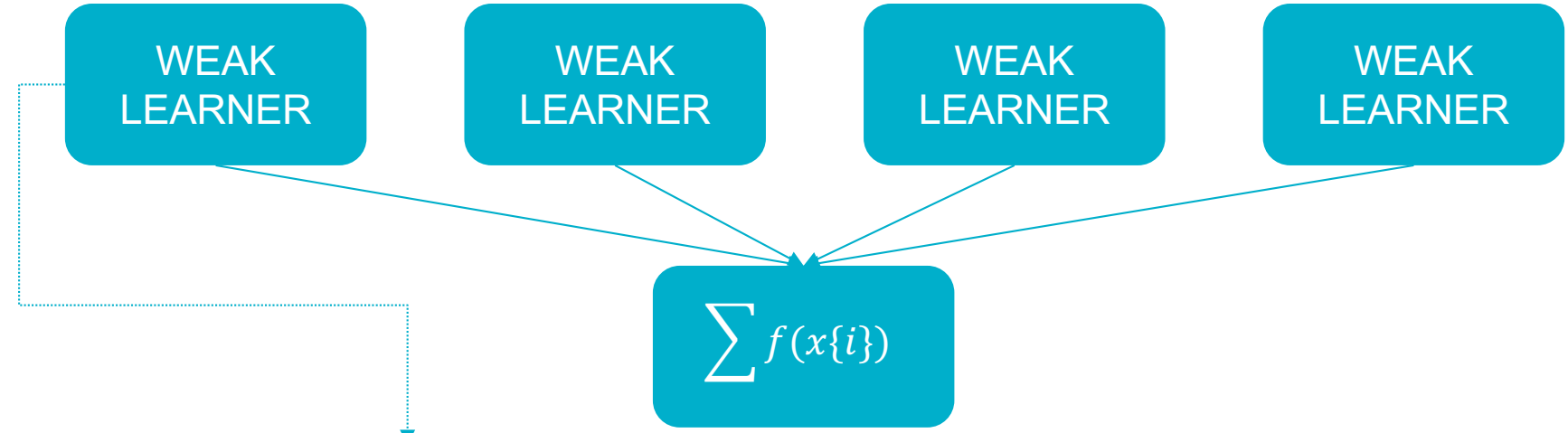
ROC AUC – AREA UNDER THE CURVE

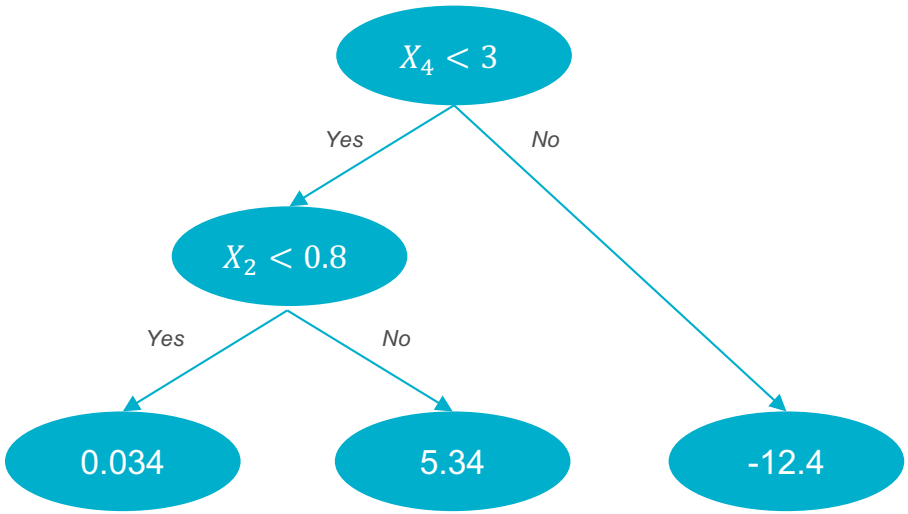


01 > FEATURE IMPORTANCE & ENSEMBLE METHODS

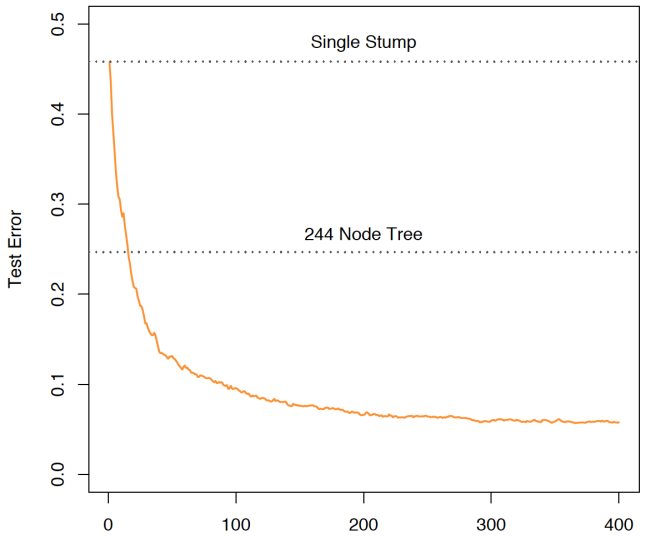
ENSEMBLE METHODS : JOIN THE PACK

A collection of weak learners is better than a unique large learner



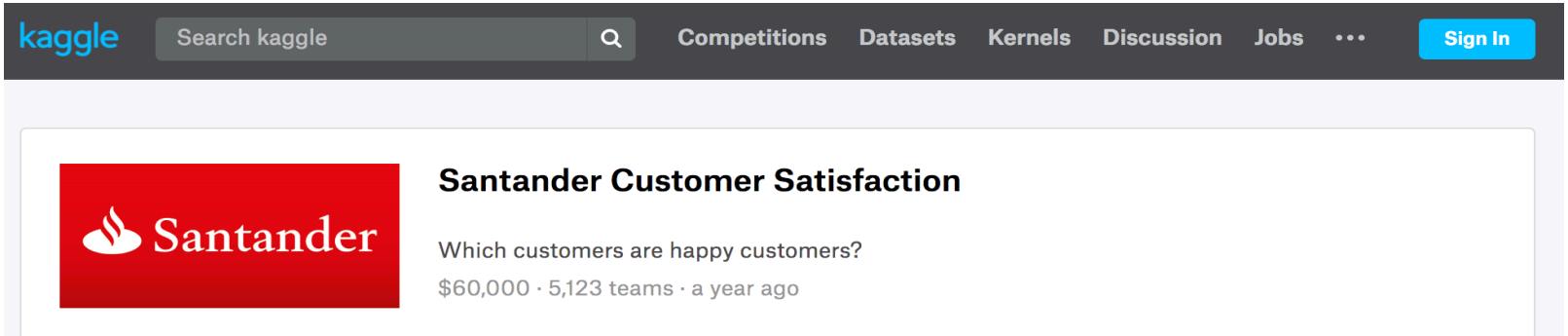


From Elements of statistical learning, Hastie

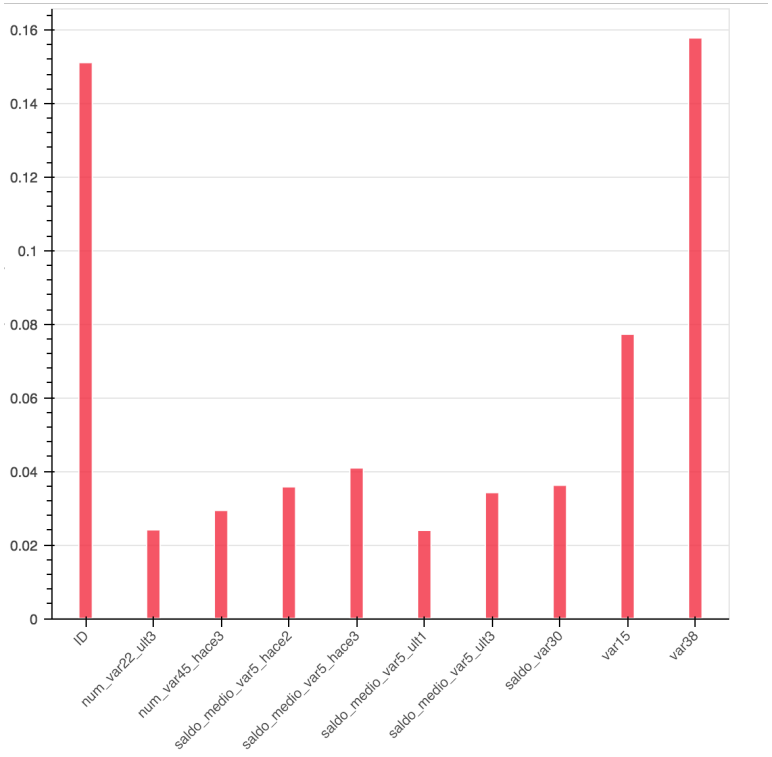


FEATURE IMPORTANCE ? JUST COUNT THEM UP !

Each split is equally important, so feature importance is just a cumulative sum



- Xgboost binary classification
- 0.83 ROCAUC (cross validated)
- Feature importance computation

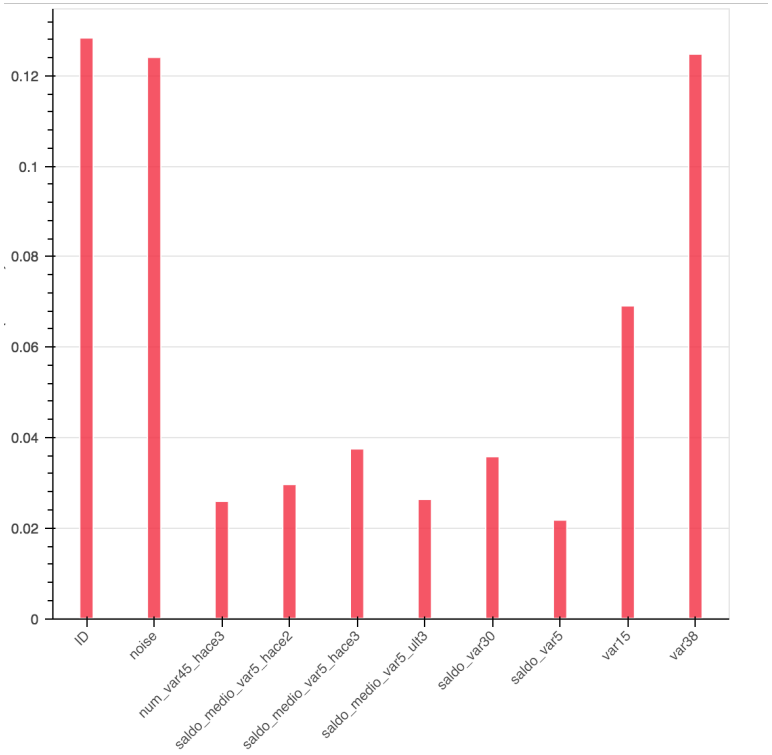


CUMULATIVE SUM IS NOT ROBUST

Adding a noise variable changes the importance of features

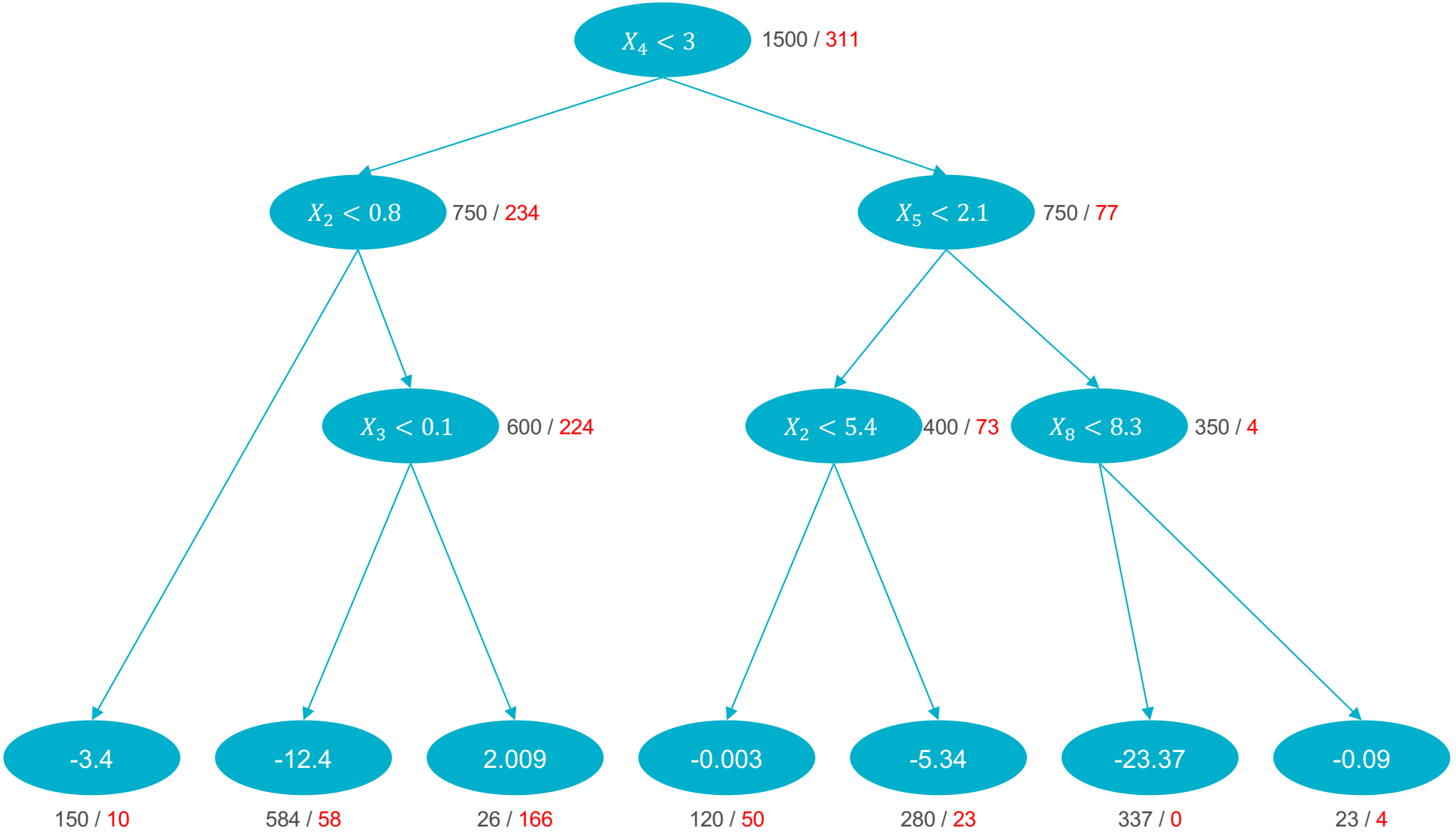
- Without any feature engineering, add random noise (uniform distribution between 0-1)

- Xgboost binary classification
- SAME PERFORMANCE (0.83)
- Feature importance computation



TREES ARE HIERARCHICAL

A feature importance is associated with the number of occurrences it handles



02 > GRADIENT BOOSTING : TREECEPTION

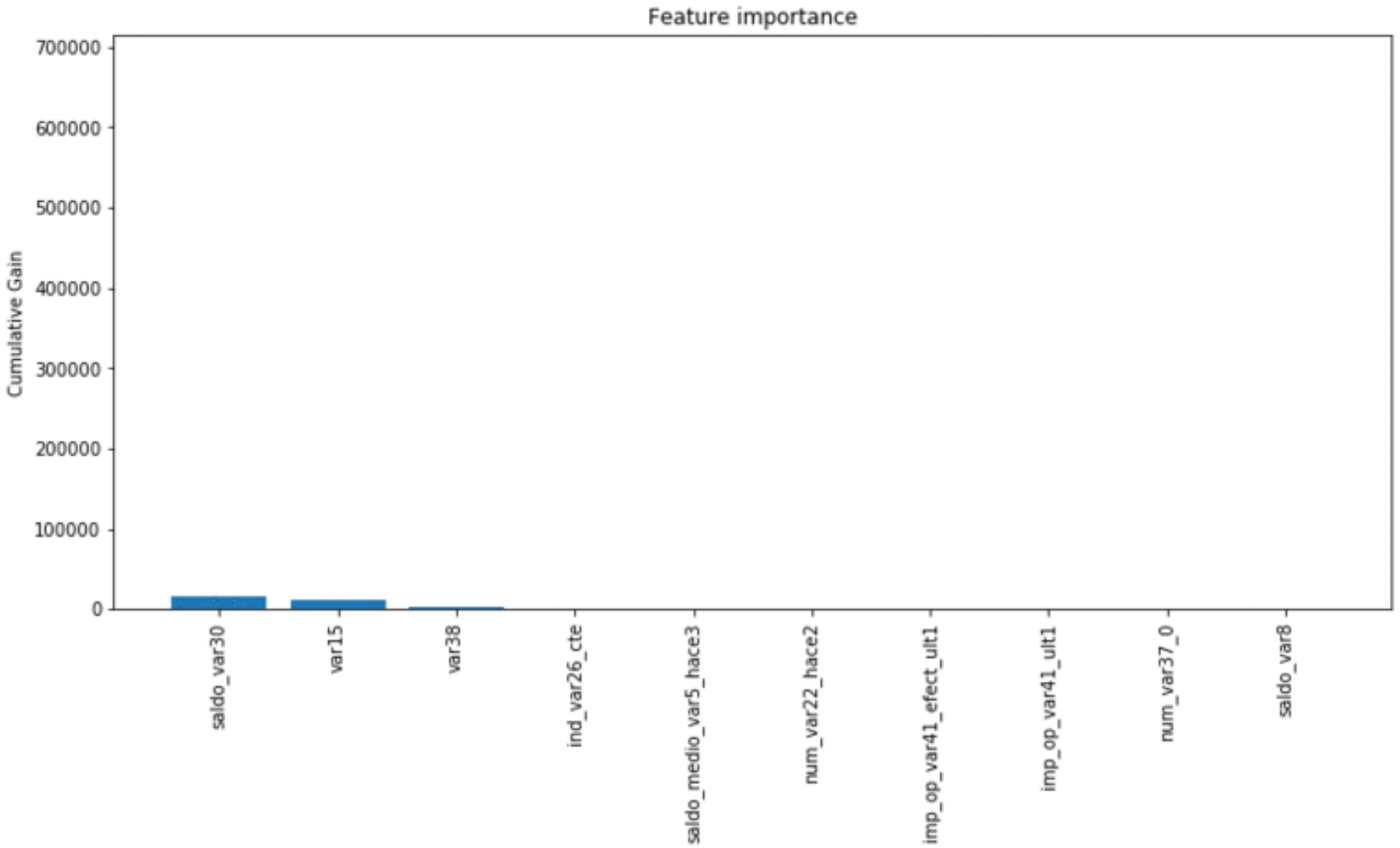
GRADIENT BOOSTING STRUCTURE

Trees make mistakes to train subsequent trees

```
booster[0]:
0:[saldo_var30<2.955] yes=1,no=2,missing=1,gain=223.785,cover=15204
    1:[var15<28] yes=3,no=4,missing=3,gain=301.257,cover=4276.25
        3:[saldo_var8<2.085] yes=7,no=8,missing=7,gain=3.61861,cover=2239
            7:[var38<13396] yes=15,no=16,missing=15,gain=3.6216,cover=2238
                15:leaf=-0,cover=1
                16:[imp_op_var39_comer_ult3<1613.43] yes=27,no=28,missing=27,gain=2.73829,cover=2237
                    27:[var15<26] yes=45,no=46,missing=45,gain=1.65054,cover=2235.75
                        45:[noise<0.00132532] yes=71,no=72,missing=71,gain=0.635187,cover=1888.5
                            71:[noise<0.00100589] yes=99,no=100,missing=99,gain=0.854701,cover=2.25
                                99:leaf=-0.111111,cover=1.25
                                100:leaf=-0,cover=1
                            72:leaf=-0.192529,cover=1886.25
                        46:[ID<151300] yes=73,no=74,missing=73,gain=5.9299,cover=347.25
                            73:[var38<47875.7] yes=101,no=102,missing=101,gain=3.33744,cover=346
                                101:leaf=-0.133333,cover=27.5
                                102:leaf=-0.183412,cover=318.5
                            74:leaf=0.0222222,cover=1.25
                    28:leaf=-0.0222222,cover=1.25
            8:leaf=-0,cover=1
        4:[var38<117321] yes=9,no=10,missing=9,gain=55.9796,cover=2037.25
            9:[var38<73981.9] yes=17,no=18,missing=17,gain=17.467,cover=1539.25
                17:[saldo_medio_var5_hace3<0.435] yes=29,no=30,missing=29,gain=15.3777,cover=636
                    29:[ind_var30_0<1] yes=47,no=48,missing=47,gain=5.02247,cover=584.5
                        47:[imp_op_var40_comer_ult1<695.445] yes=75,no=76,missing=75,gain=0.141027,cover=12
                            75:leaf=-0.183333,cover=11
                            76:leaf=-0.05,cover=1
                        48:[imp_op_var41_efect_ult3<405] yes=77,no=78,missing=77,gain=5.86103,cover=572.5
                            77:[var15<33] yes=103,no=104,missing=103,gain=4.56021,cover=569.5
                                103:leaf=-0.12487,cover=142.75
                                104:leaf=-0.101812,cover=426.75
                            78:[saldo_medio_var8_ult1<11.58] yes=105,no=106,missing=105,gain=4.75,cover=3
                                105:leaf=0.1,cover=2
                                106:leaf=-0.1,cover=1
                    30:[num_op_var41_hace2<43] yes=49,no=50,missing=49,gain=5.43074,cover=51.5
                        49:[noise<0.0232753] yes=79,no=80,missing=79,gain=1.47009,cover=50.25
                            79:leaf=-0.04,cover=1.5
                            80:[imp_op_var41_efect_ult3<2850] yes=107,no=108,missing=107,gain=0.0875811,cover=48.75
                                107:leaf=-0.177436,cover=47.75
                                108:leaf=-0.05,cover=1
                        50:leaf=0.0222222,cover=1.25
```

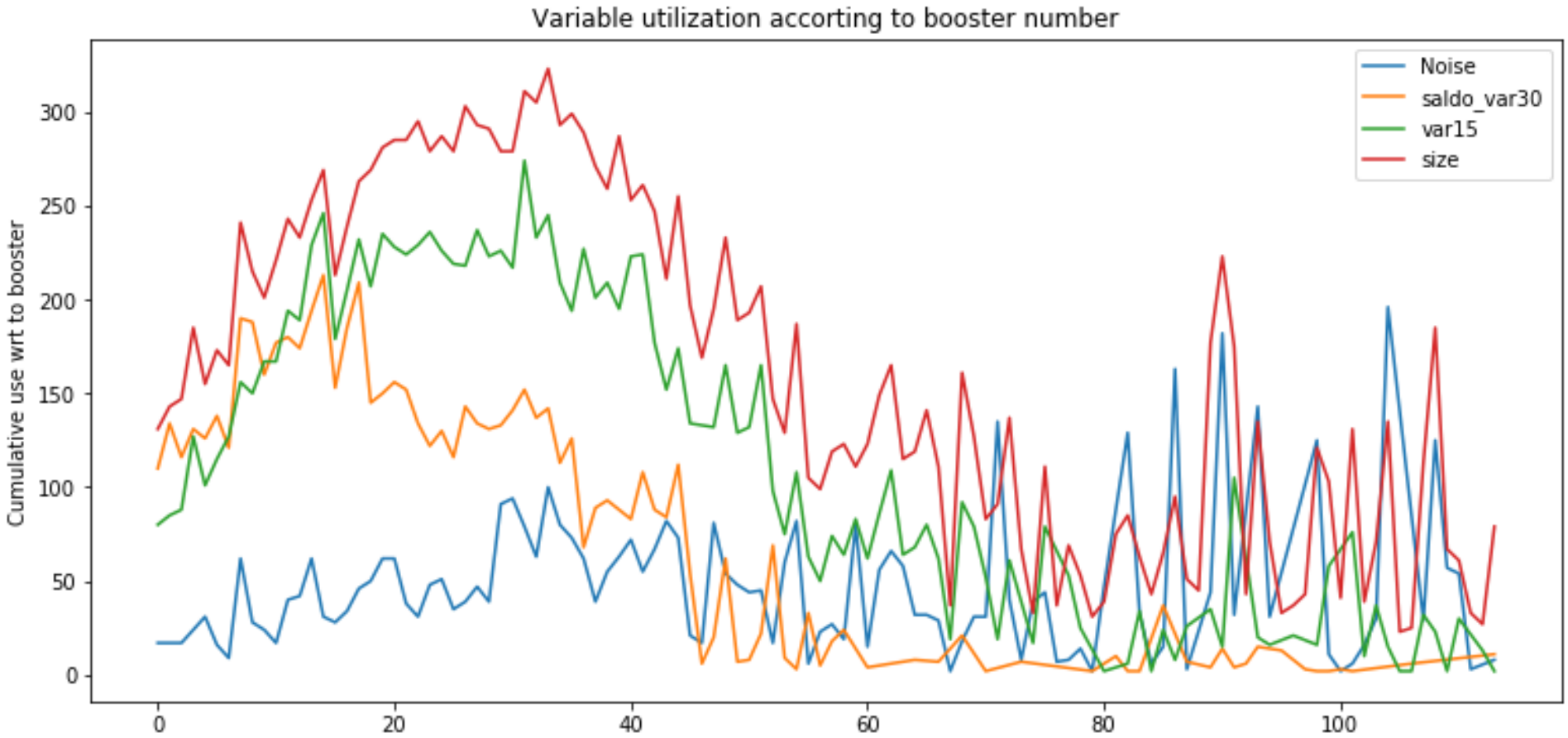
EVOLUTION OF VARIABLES REGARDING ITERATIONS

Cumulative gain is stable through iterations



NOISE ADDITION IN GRADIENT BOOSTING PREDICTION

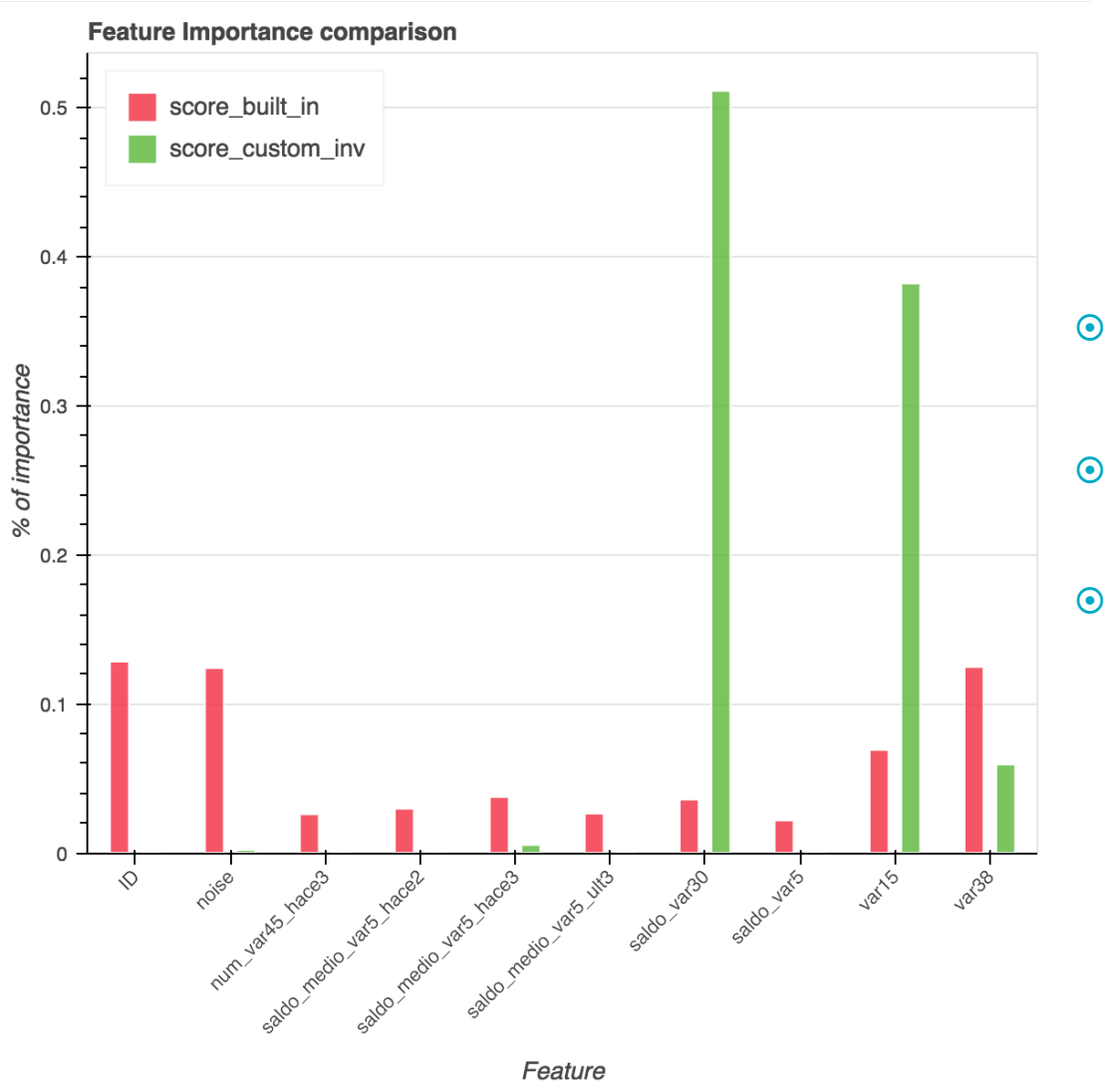
Noise is used only deeply in the trees with small discrimination power



Depth	2	3	4	5	6	7	8
%	0.02	0.12	0.65	2.03	6.5	16.26	74.4
Part-2	100	90	88.67	75.30	91.79	86.75	89.6

A ROBUST METRIC NOISE-PROOF

It cancels noise, it must be good !

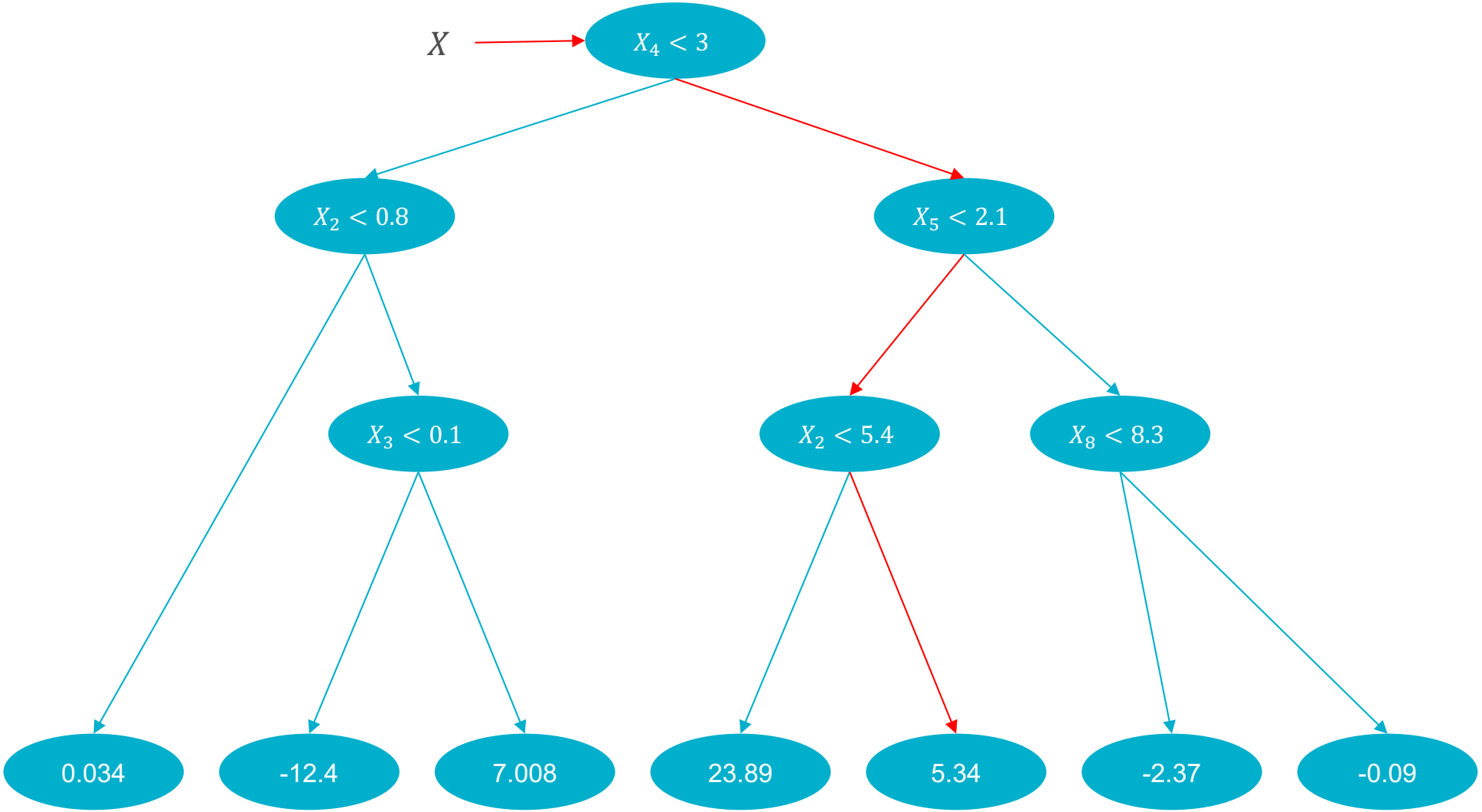


- $Importance = \sum_{every\ tree} \frac{Gain}{\#\ booster^2}$
- Noise is ranked 9th
- Equivalent performances with 8 features (ROC AUC .82)

03 > POINT FEATURE IMPORTANCE ?

IS IT REALLY IMPORTANT FOR EVERYONE ?

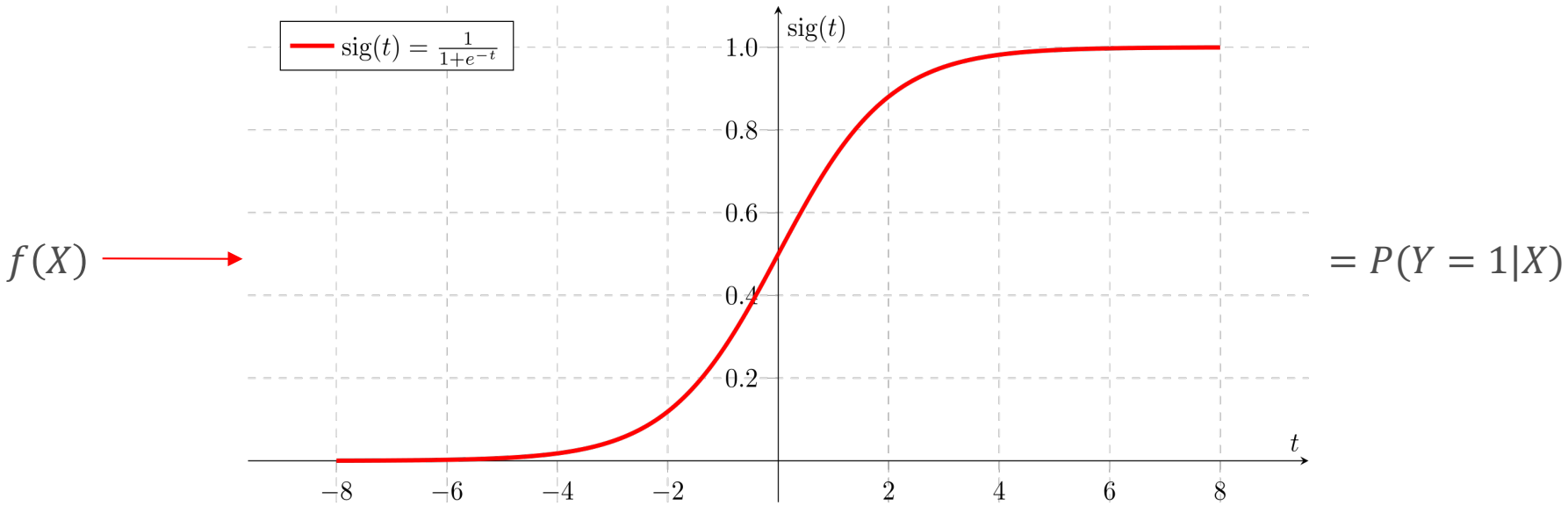
A data point does not cross the entireness of the Forest



GRADIENT BOOSTING PREDICTION IS A SUM

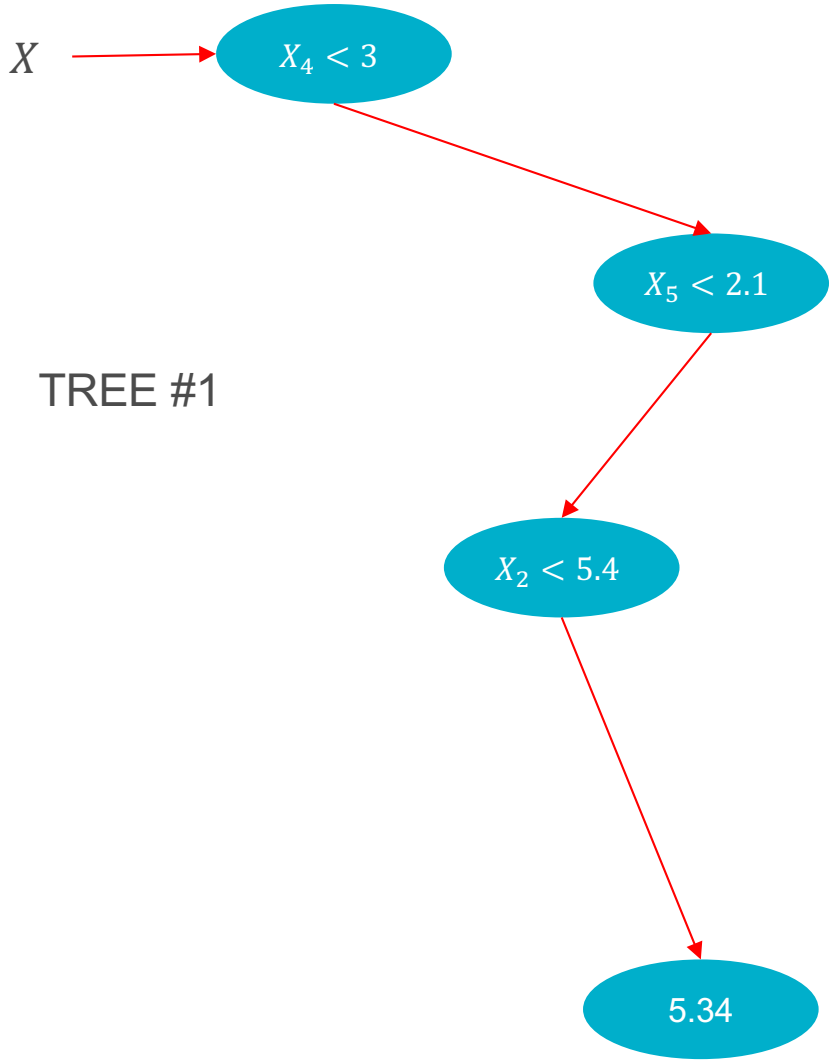
Aggregation of the scores for each leaf

$f(X) =$ 7.008 + 23.89 + … + -0.35 + -13



.PREDICT : FOLLOW THE PATHS

The score of a leaf is obtained by going through the « gates » of the splits



TREE #1

$$score_{tree} = \sum_{splits \in tree} \frac{1}{\#features} * score_{tree}$$

$$score_{feature} = \sum_{\substack{tree \in trees \\ i, feature\ in\ tree}} w_i * score_{tree}$$

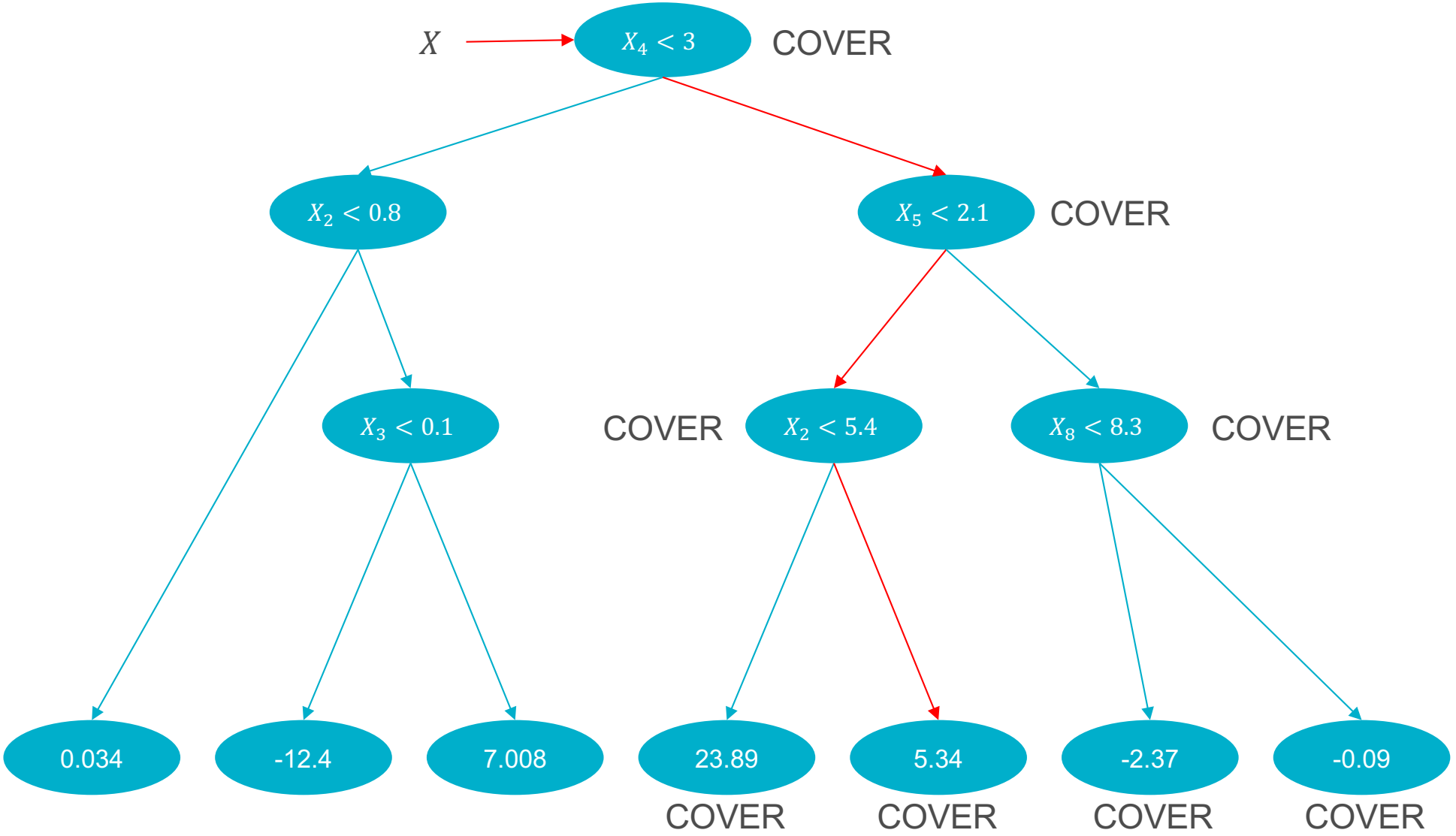
AGGREGATION OF SEVERAL POINTS

Residual interpretations

- Test
  - **10 most False Positives** : not a high probability of being zero
  - **Top 5 contributers** toward a positive value (misclassification)
- Results
  - **Two variables come for 4 of the 10 data points**: saldo_medio_var5_hace3, var3
  - **ID** is in the top : generation of noise
  - Lead for investigation **BUT** does not tell what threshold is faulty

HOW MUCH A NODE IS RESPONSIBLE FOR THE FINAL SCORE ?

Using the data point distribution in a leaf, we can back propagate the score



.PREDICT : STILL THE SAME THE PATHS

Same same, but different, but still same

$$score_{node} = \sum_{\substack{children \\ of\ node}} \frac{cover_{child}}{\sum_{children} cover} * score_{child}$$

$$score_{feature} = \sum_{\substack{tree \in trees \\ feature\ in\ paths}} score_{node}$$

AGGREGATION OF SEVERAL POINTS

Residual interpretations

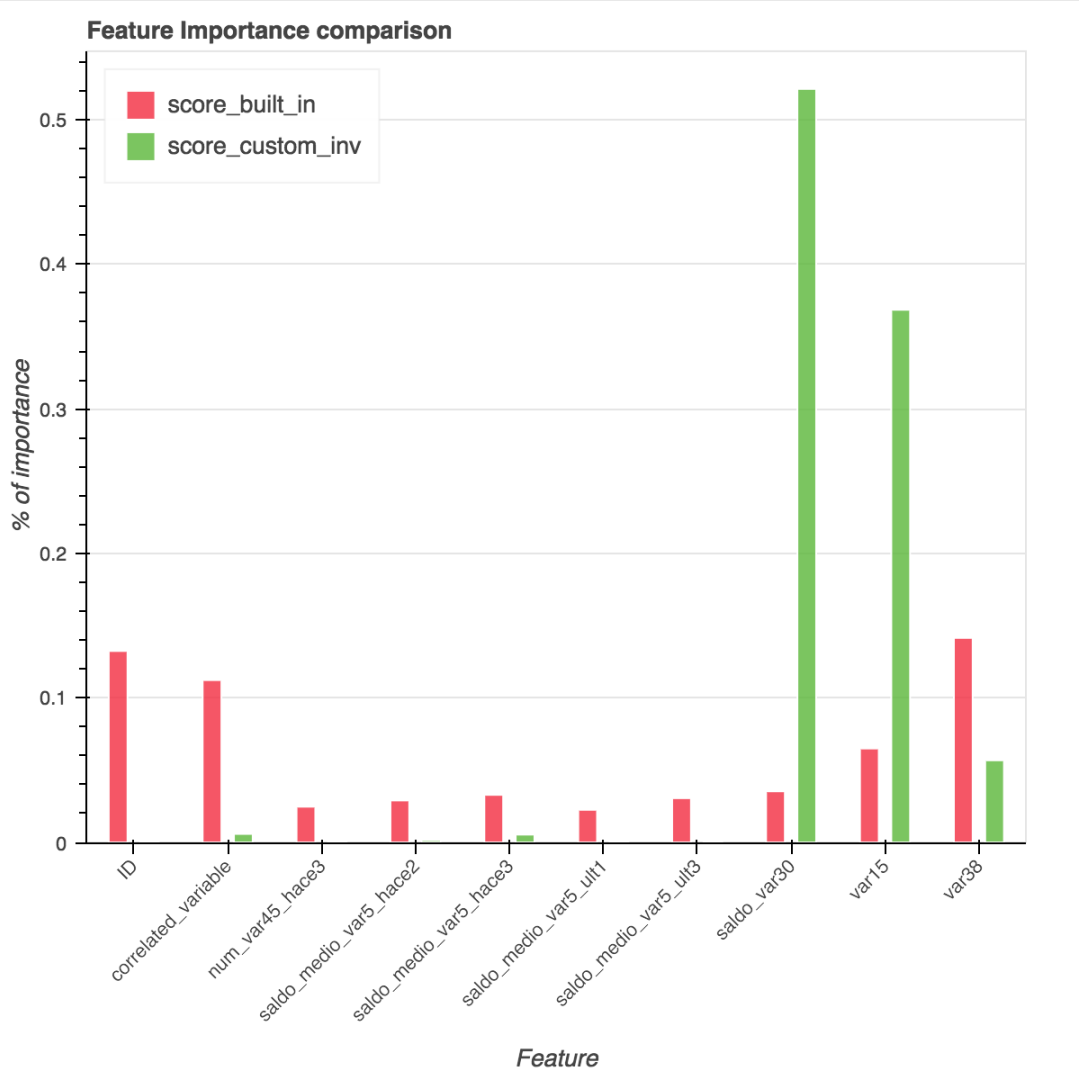
- Test
  - **10 most False Positives** : not a high probability of being zero
  - **Top 5 contributers** toward a positive value (misclassification)
- Results
  - **Not the same variables pointed out**
  - Killer Features are main contributers
  - Maybe look at the hypercube generated in terms of separation in space

04 > LIMITS & WAYS FORWARD

CORRELATED VARIABLES

Machine Learning is just correlation, not causality

- Without any feature engineering, add highly correlated variable to var15 (95 %)



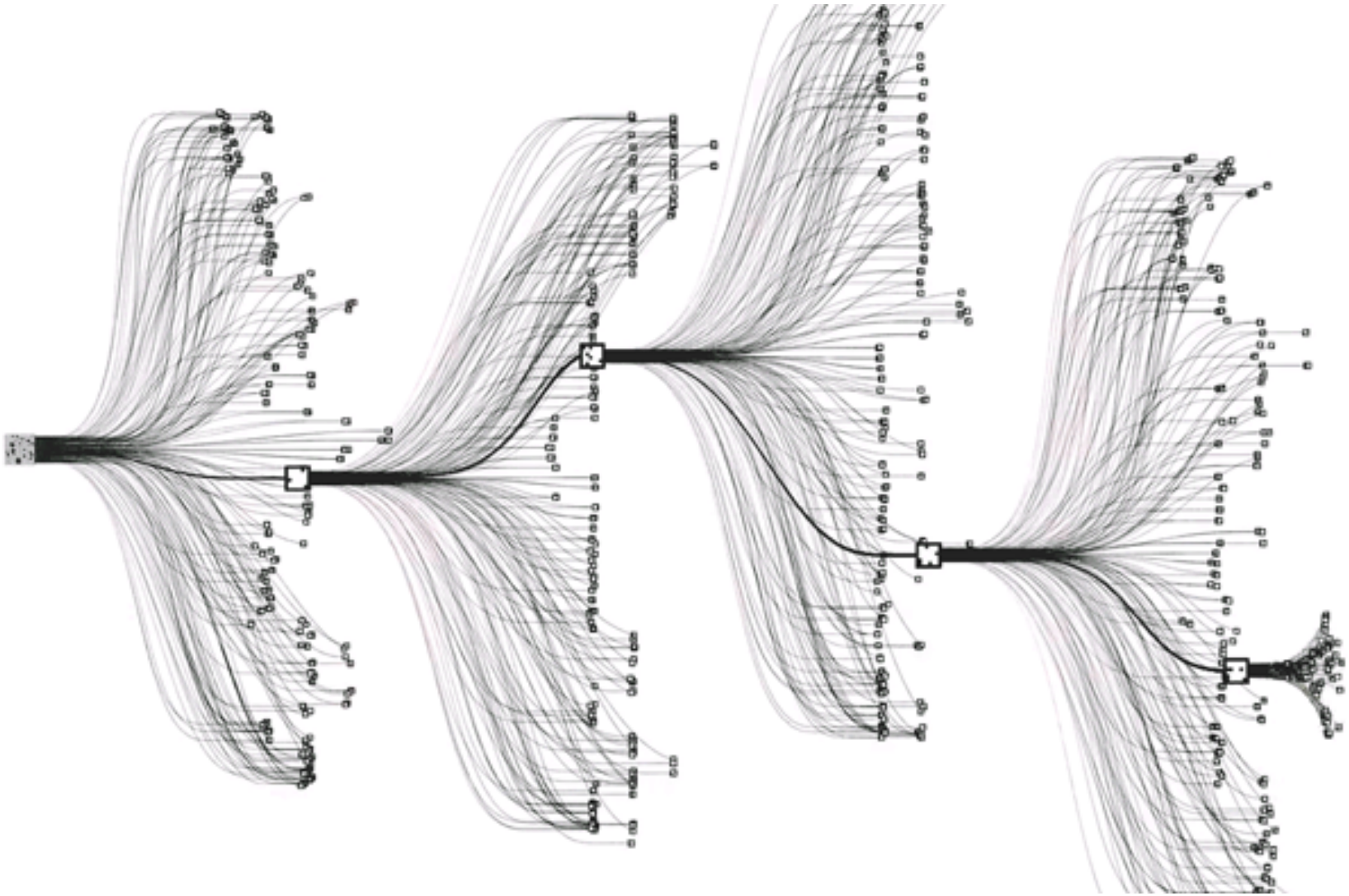
INTERPRETATION IS NOT A MIRACLE

Prediction is achievable, but not prescription

« There will always be a compromise between predictive power and interpretabiliy »

INTERPRETATION IS MORE IMPORTANT THAN PERFORMANCE

A 90% accuracy interpretable model is better than 99% accuracy black box



Alphago – Deep Mind

MULTI CLASS / REGRESSION / FINE TUNING

Few leads for the next step

- Multi-class classificiation generates a huge amount of trees with gradient boosting
- Regression split gains are different
- Regression scores are inherently different
- What is the influence of the cost function (L1 – L2) ?
- Taking the threshold into account could lead to hypercube defintions for data segmentation
- Everything is on my github : **Cnstant/feature_importance_gbm**

THANKS FOR YOUR ATTENTION

- Questions ?

05 > ANNEXES

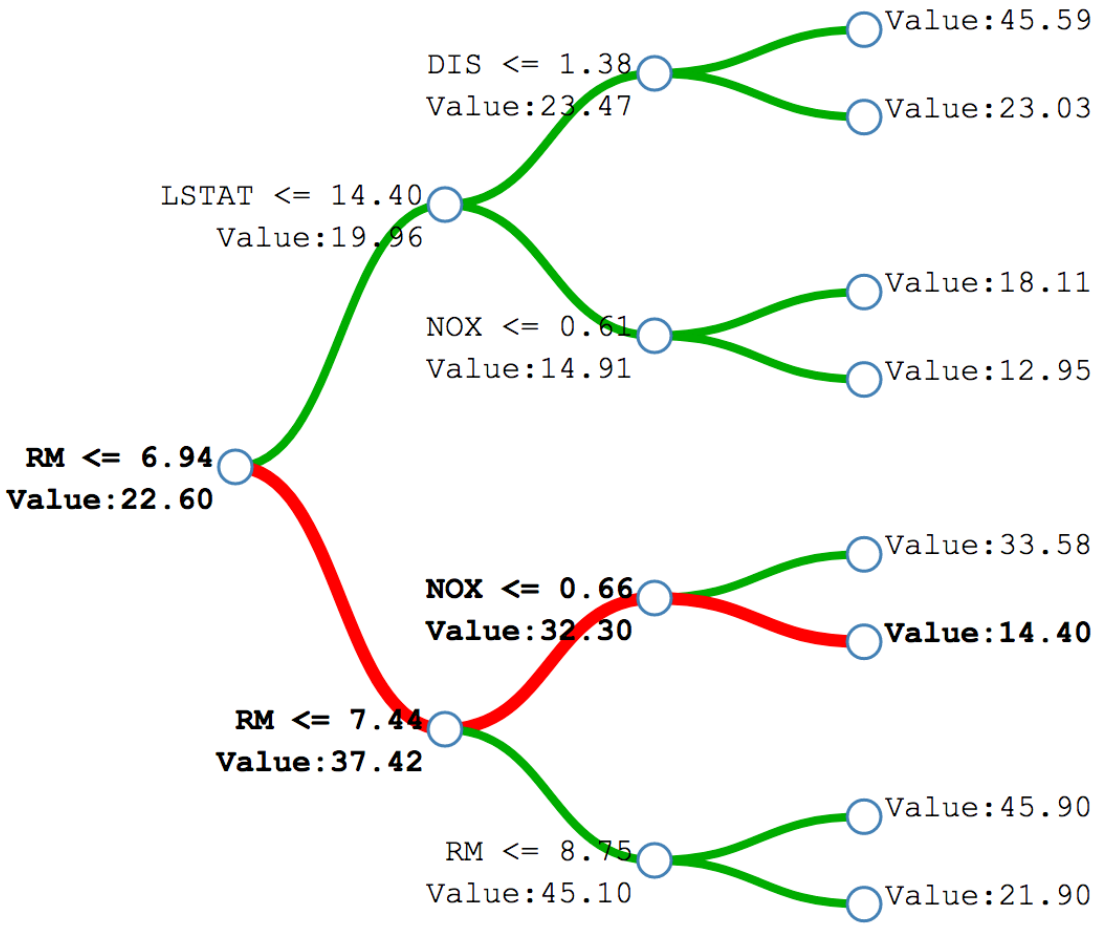
RECURSIVITY FOR PATH ENUMERATION

Implementation is to be open sourced

```
def get_paths(node, path=None):
    #import pdb; pdb.set_trace()
    path = path or []

    # Copy of the actual path if there is a right child
    if node.right:
        right_path = copy(path)

    # Left child inherits the current path
    path.append(node)
    paths = get_paths(result[node.left], path) if node.left else [path]

    # Right child is second and extend the result
    if node.right:
        right_path.append(node)
        right_paths = get_paths(result[node.right], right_path)
        paths.extend(right_paths)

    # Pass the results back
    return paths
```

TREE INTERPRETER FOR RANDOM FOREST

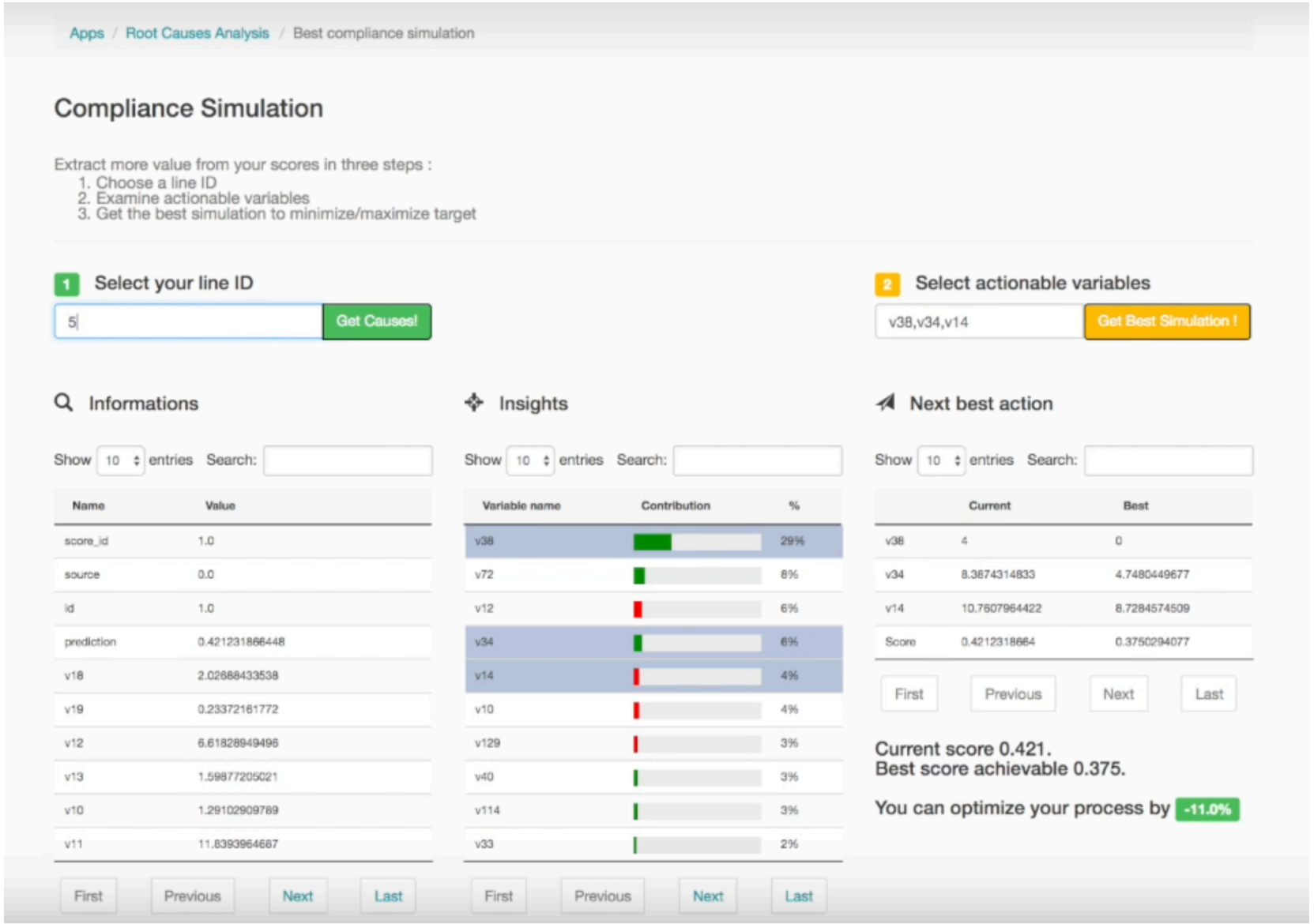
Random Forest are only horizontal



Prediction: **14.40** ≈ **22.60** (trainset mean) + **14.82**(gain from RM) - **5.12**(loss from RM) - **17.9**(loss from NOX)

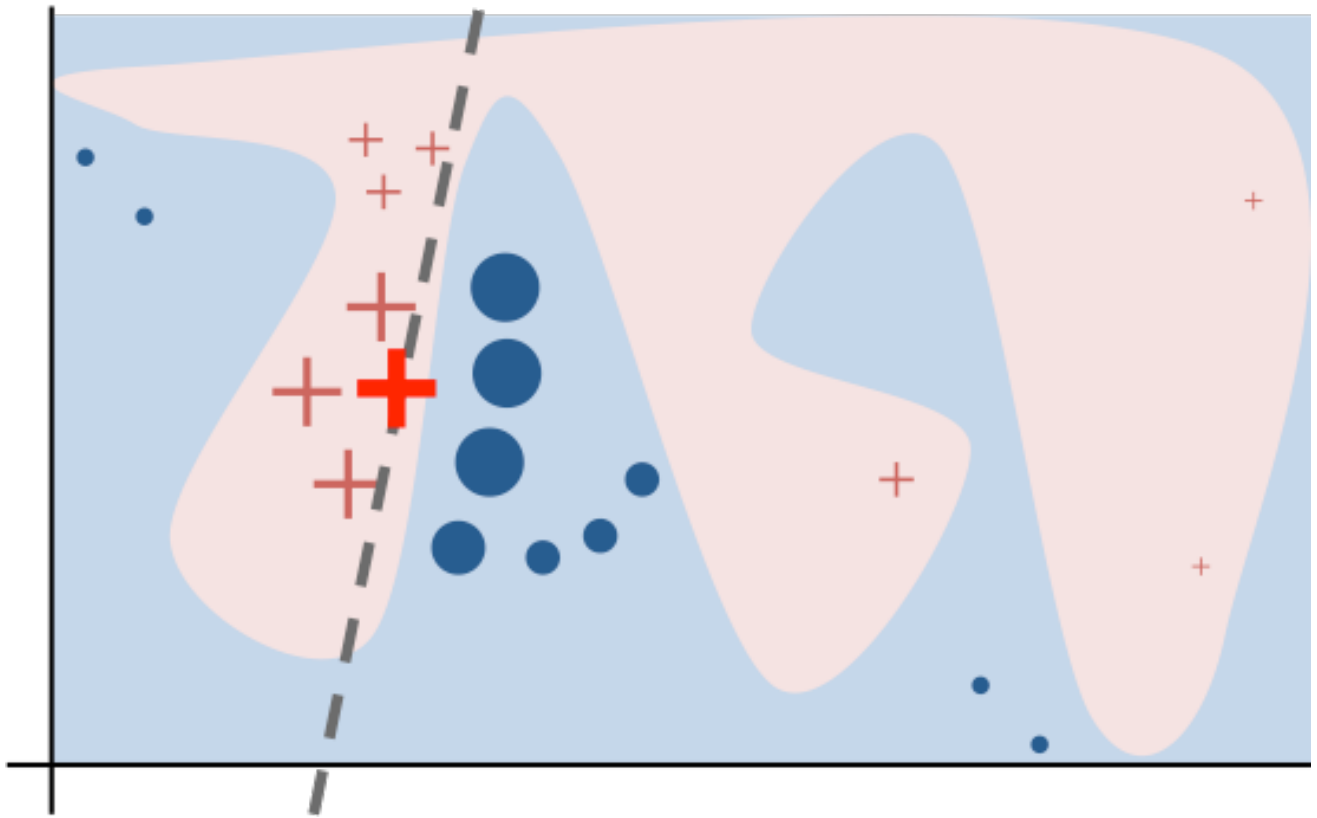
TREE INTERPRETER FOR RANDOM FOREST

Can we turn prediction into prescription ?



LIME

Don’t interpret the model, interpret the stability of the prediction



LIME

Feature importance is how much you need to change your value to change class