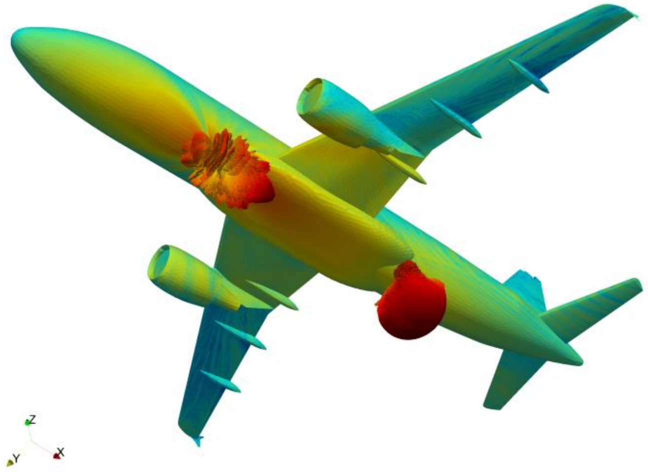
Learning for predicting the rank of hierarchical matrices

Théo Briquet

Joint work with Pierre Benjamin, Luc Giraud, Sofiane Haddad, Paul Mycek and Guillaume Sylvand,

June 17th 2024

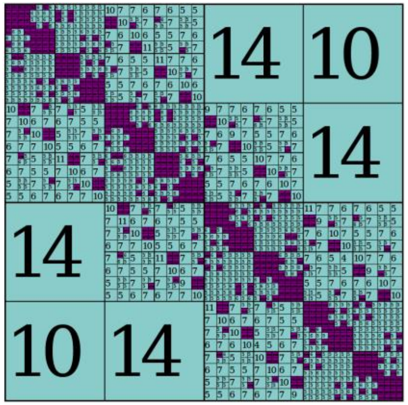
Context



- Wave propagation problems in acoustics and electromagnetism.
- $Ax = b$ with A dense and large with around 5 million unknowns (for design studies).
- LU factorisation on Frontier: (1^{st} TOP 500 : 1,2 Exaflops)
  - > 83 secondes using more of 9 millions of cores.
  - > Airbus platform : 128 cores : **60 days** !

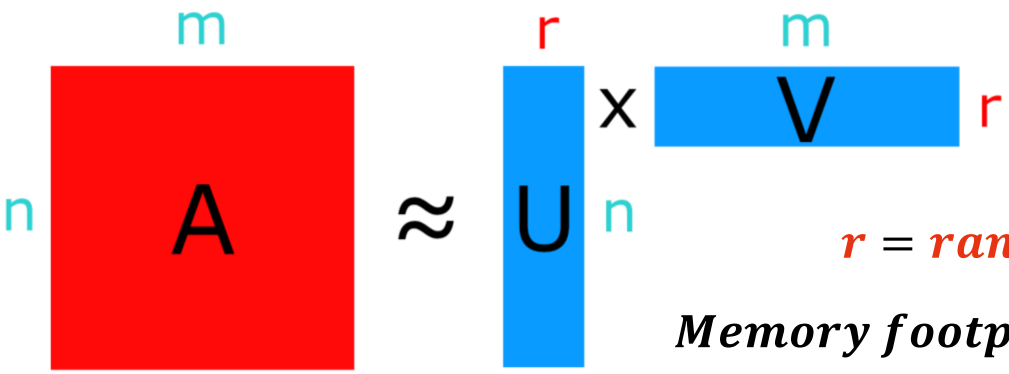
➢ Too costly in terms of time and energy.

Context



- <u>Solution</u> : Use of specialized solvers such as Hierarchical matrices.

- <u>Advantages of Hierarchical matrices</u>:
  - > Format allowing dense matrices to be stored in compressed form : low-rank blocks.



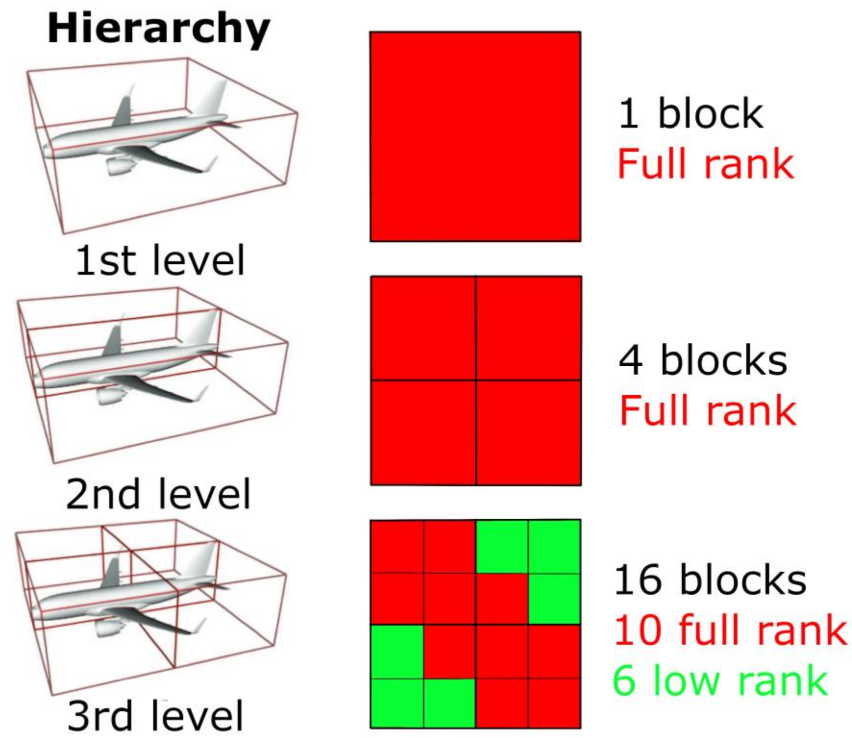
$$r = rank \text{ of } A(n, m)$$

$$Memory\ footprint = rank * (n + m)$$

  - > Algebra encoding (matrix-vector product, factorization, etc.) already implemented.

Context

<u>Bounding box:</u> the **smallest rectangular** parallelepiped capable of fully containing the object in 3D space



- Each off diagonal entry of the matrix represents the **interaction** with its neighbors.
- Each block of the matrix is associated with **two bounding boxes** (row and columns unknows).

Objectives

1. Estimate whether a block of a hierarchical matrix **fits in memory or not** (this depends on the rank). I will use a classification model.

2. **Predict the rank** of the blocks of a hierarchical matrix, allowing to know the memory footprint of blocks. I will use a regression model.

Stakes

- Avoid **memory** (jobs) **crashes**. Currently, Airbus takes significant margins to estimate the rank.

- Knowing the skeleton of the matrix in advance, which helps **optimize compression**.

- **Improve performance** and consume **less energy**.

Outline

1. ML Introduction
2. Classification
3. Regression
4. Conclusion

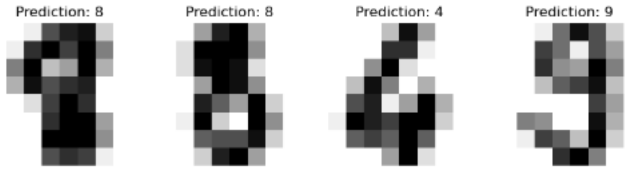
01

Quick Machine Learning Introduction

Introduction to Machine Learning

Classification :

- Classification is a machine learning technique used to **predict the class or category** of an object (dataset) based on its features.
- It is based on learning from pre-labeled examples.
- Classification is often used for pattern recognition and **decision making**.

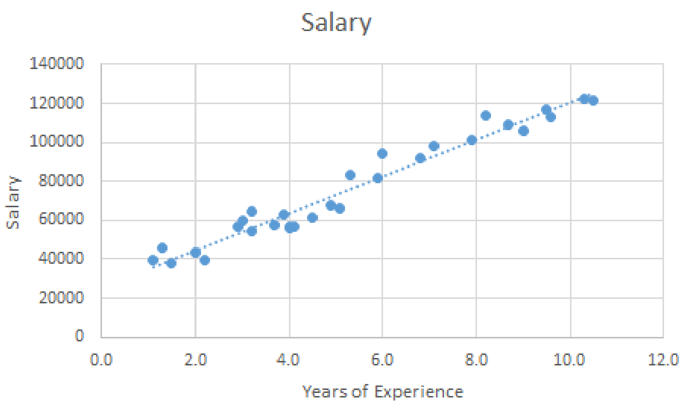


Example : Digit recognition in handwritten text

Introduction to Machine Learning

Regression :

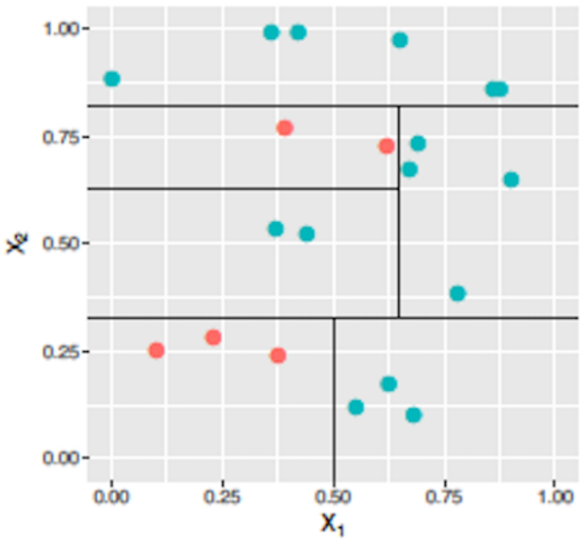
- Regression is a machine learning technique used to **predict a numerical value** of an object (dataset) based on its features.
- There are several types of regression: linear, polynomial, logistic, etc
- Regression is often used for financial forecasting such as stock price prediction, economics to study the relationships between economic variables, social sciences etc.



Example : Linear regression to predict the salary vs experience

Random Forest

The random forest model is an **ensemble of decision trees**.



Principle of a decision tree: Successive binary divisions of the feature space to obtain a partition of the space into subspaces **where the data have the same label** (here : blue color – orange color).

Principle of a Random Forest : For each point, each tree predicts a class (which can differ from tree to tree). **The final class chosen** by the forest is the one that has received the **highest number of votes** among all the trees (**majority vote**).

Dataset Presentation and Evaluation Metrics

Dataset Overview:

- Derived from an F22 aircraft
- Number of instances : **almost 3 millions** of hierarchical blocks
- Split into 80% of training data and 20% of test data

Note : more of 99% where rank < 20 -> **Few high-rank data**.

Evaluation Metrics :

1. **Classification Score** : Measures the mean accuracy of the model's predictions. Range: 0 to 1 (higher is better).
2. **R2 Score** (Regression) : Measures the proportion of variance in the dependent variable that is predictable from the independent variables. Max value : 1 (higher is better).

02

Classification

Objective :

Create a classification model to predict whether a block fits in memory (True) or not (False). A characterization of this condition could be to check if :

Memory footprint : $rank * (m + n) \leq threshold$

Note: 7% of the data **exceed** the chosen $threshold$ of 5000 .

The 9 features :

1. Center (x, y, z) of the two bounding boxes (6 features).
2. Distance between the two bounding boxes (1 feature).
3. Diameter (x,y) of the two bounding boxes : the longest diagonal segment (2 features).

Results :

Random Forest model :

- Train Score : **1** (perfect)
- Test Score : over **0.99** (≃2000 misclassified out of 550,000)

Model analysis

Let us denote by $\boldsymbol{p}$ **the proportion** of trees that classify **True** (the block fits in memory) in the random forest.

Consequently, (1-p) is the proportion of trees that classify False (the block does not fit in memory).

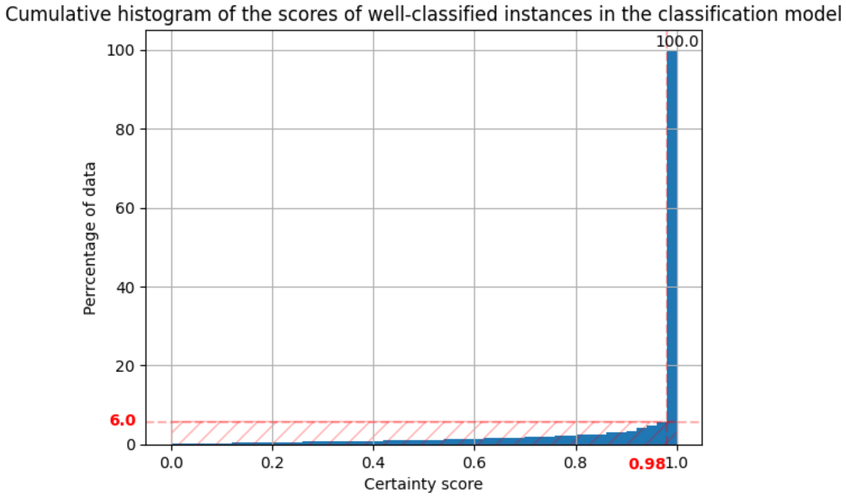
Given this, the prediction of the random forest for a particular block is determined by the majority vote of the trees.

1. If $p > 0.5$, the block is classified as True (fits in memory).
2. If $p \leq 0.5$, the block is classified as False (does not fit in memory)

To achieve better visualization, we define the Certainty score as:

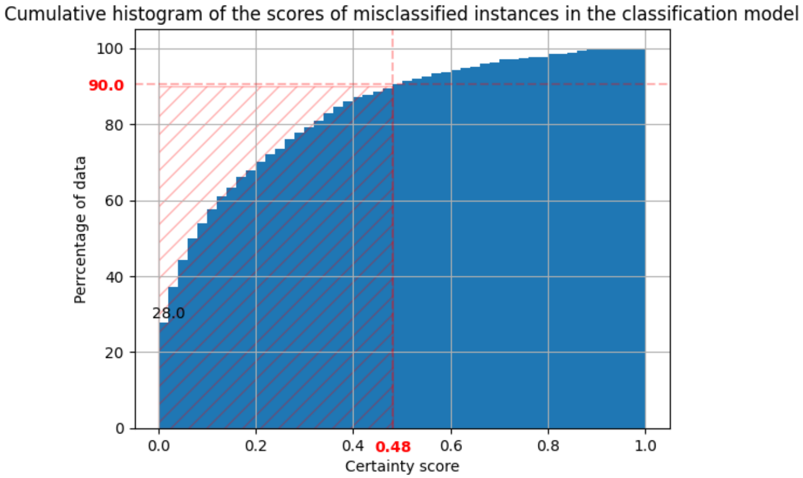
$$Certainty\ score = 1 - 4\, p\, (1 - p)$$

Well-classified



- When the model classifies correctly, it is almost always confident.

Misclassified



- When the model misclassifies, it is generally not confident, with 28% of data with high uncertainty.

Model improvement

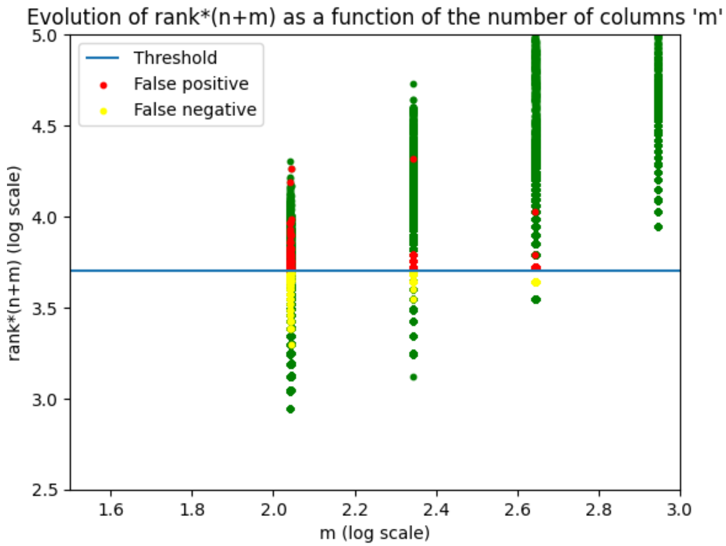
Problem : Predicting that a block fits in memory when in reality it does not (False positive) is considered a **serious error** that we want to minimize because it cand lead to memory crashes.

Idea: In the predicting phase, we retain the prediction of True (the block fits in memory) only if our model is **confident**. Otherwise, we decide that the block does not fit in memory.

Adjusted voting : We retain the True prediction only if at least 95 % of the trees have voted True , i.e., if **p ≥ 0.95**.

Note : 95% of trees is the most balanced choice.

Baseline model



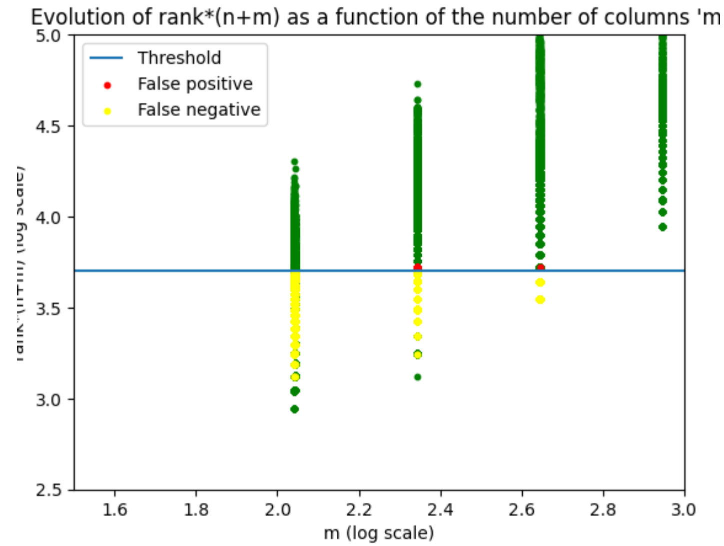
<u>Test data :</u>

2019 misclassified with :

- False positive : 952
- False negative : 1067

Score : over 0.99

Adjusted model



<u>Test data :</u>

7687 misclassified with :

- False positive : 14
- False negative : 7673

Score : over 0.98

<u>Objective achieved:</u> fewer false positives
and the remaining ones are close to the threshold

03

Regression

Objective :

Create a regression model to predict the rank of the blocks of hierarchical matrices.

The 10 features :

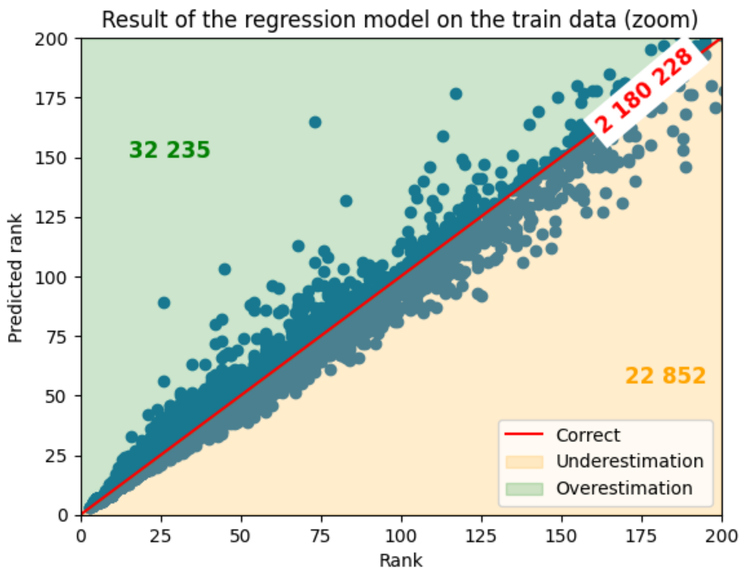
The 9 previous features + m : number of columns in the block.

Results :

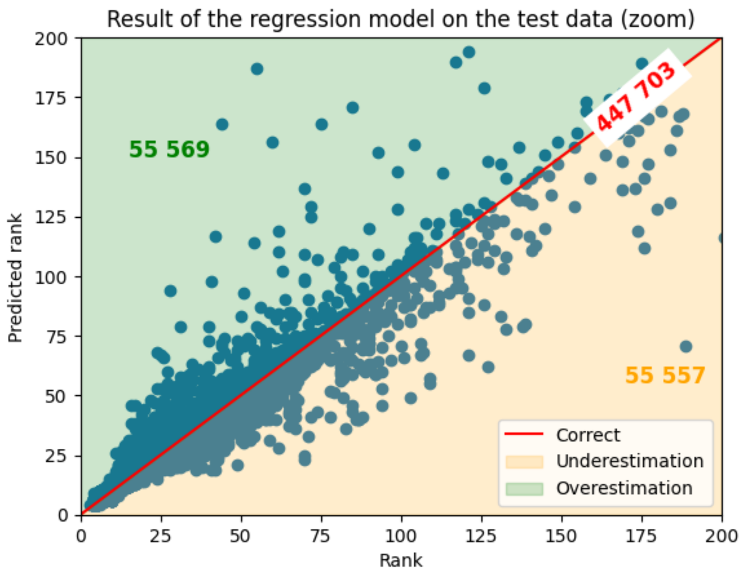
Random Forests model :

- Train R2 Score : over **0.98**
- Test R2 Score : over **0.86**

Train data



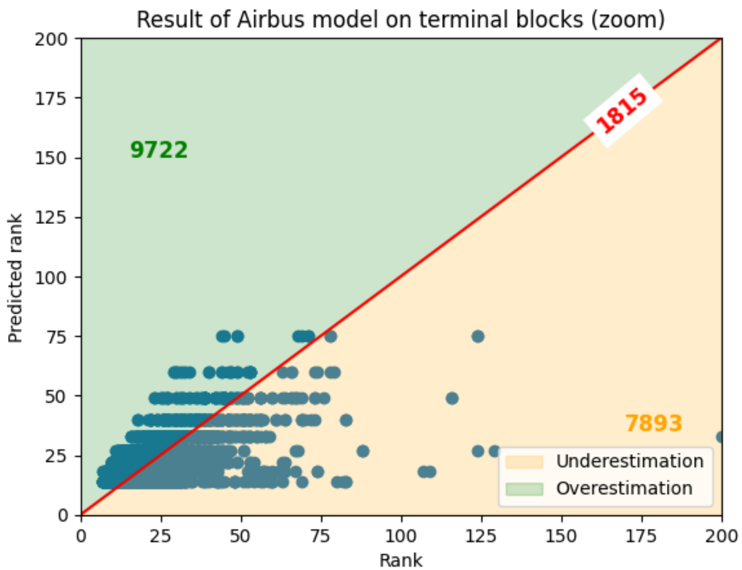
Test data



- Over **97%** of the data where the rank was correctly predicted.
- Prediction errors are relatively low, and there are few serious errors

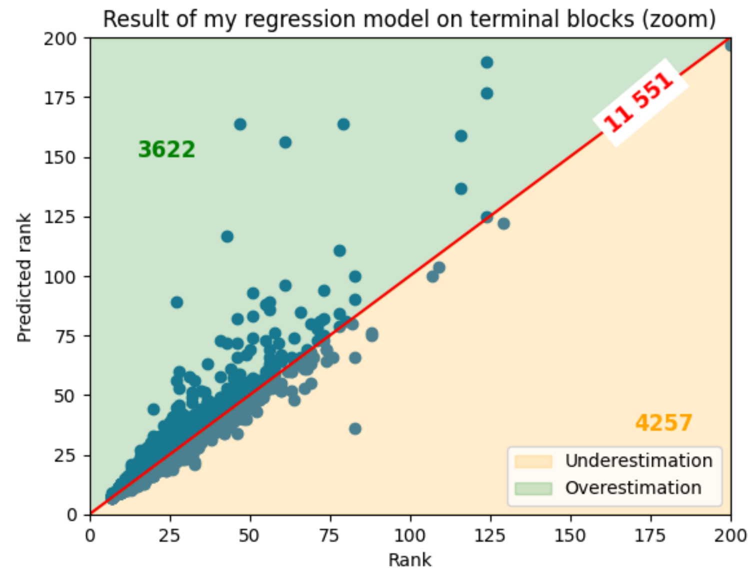
- Over **80%** of the data where the rank was correctly predicted.
- Prediction errors are more significant

Naive Linear Regression



- R2 Score : around -0.59
- R2 score with rank < 100 : 0.07697946308900216

Proposed model



- R2 Score : over **0.92**
- Best Model : More points with accurately predicted ranks

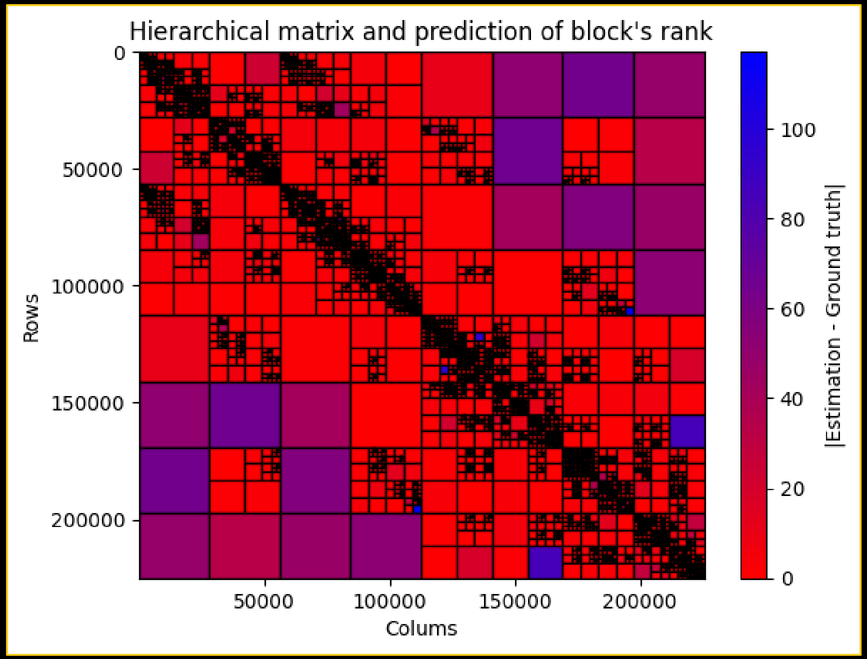
Conclusion

- Two objectives achieved :
  1. A highly performant classification model to determine if a block fits in memory
  2. A moderately high-performing regression model to predict the rank of a block

- Next steps :
  1. Try my models on more realistic and complex datasets (Airbus airplane)
  2. Further improve the regression model (try other models)
  3. How to handle ranges of rank where only few training data points are available?

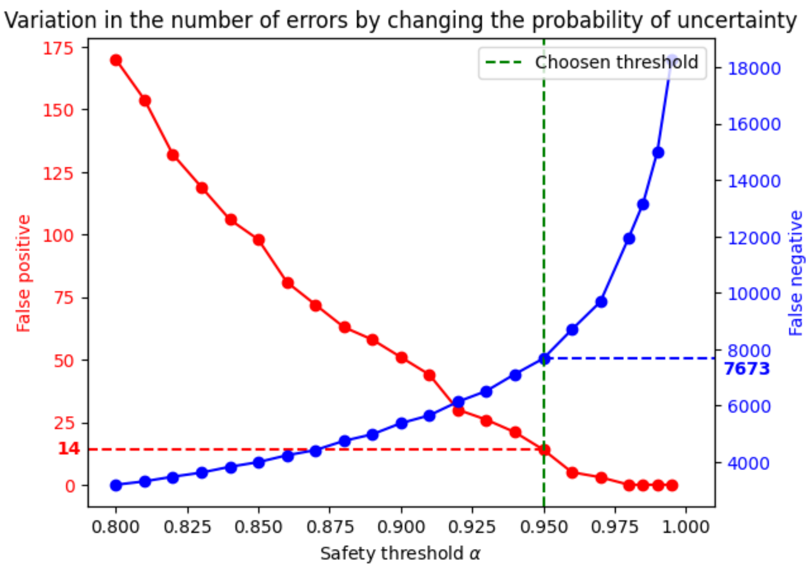
Inria

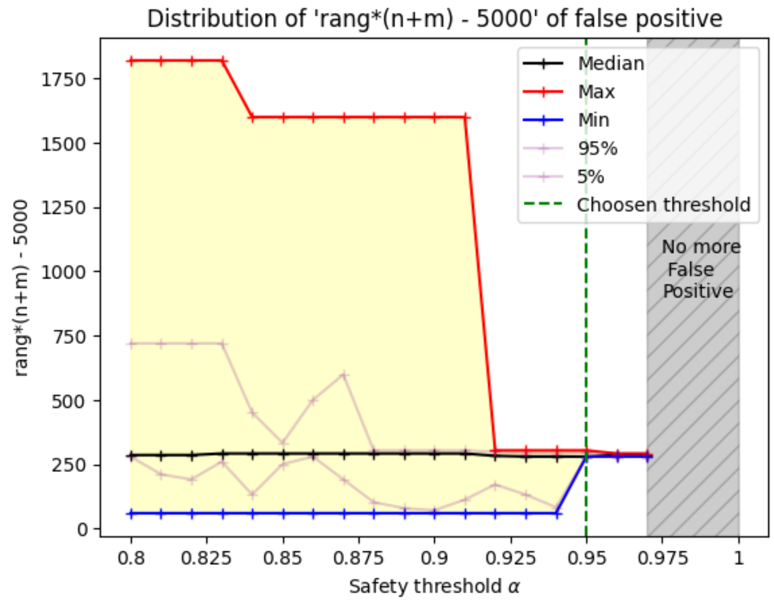
Thank you !



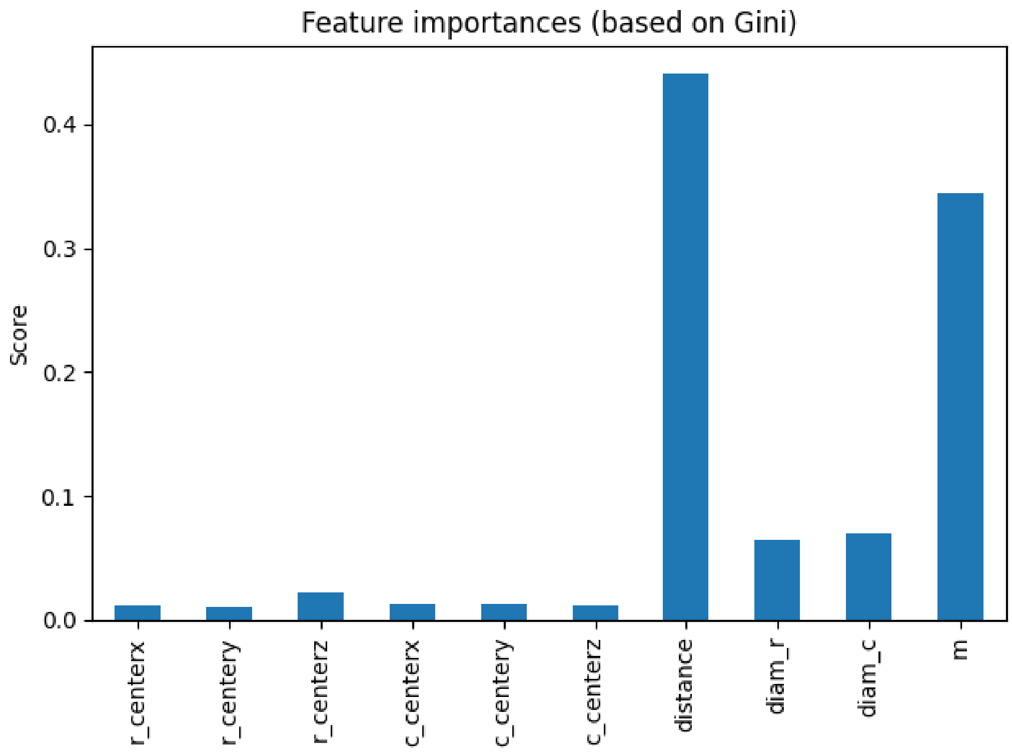
theo.briquet@inria.fr

Inria





- The more the threshold α increases, the number of false positives decreases, and the number of false negatives increases.
- The higher the threshold, the lower the maximum $rank \ * (n+m)$ decreases



Distance and m are the most important features in the model.