ONCFM

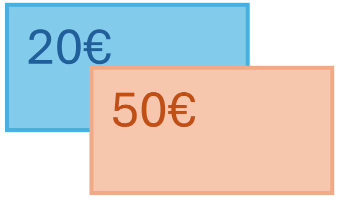
Organisation nationale de lutte contre le faux-monnayage

Counterfeight notes detection based on geometric features

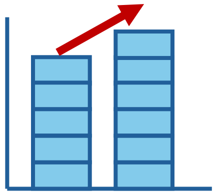
Vivien Pichon
05.2025

Euro counterfeiting on the rise

Euro market is well controlled
18 fake notes per million



Most counterfeit notes are in denominations of 20€ and 50€

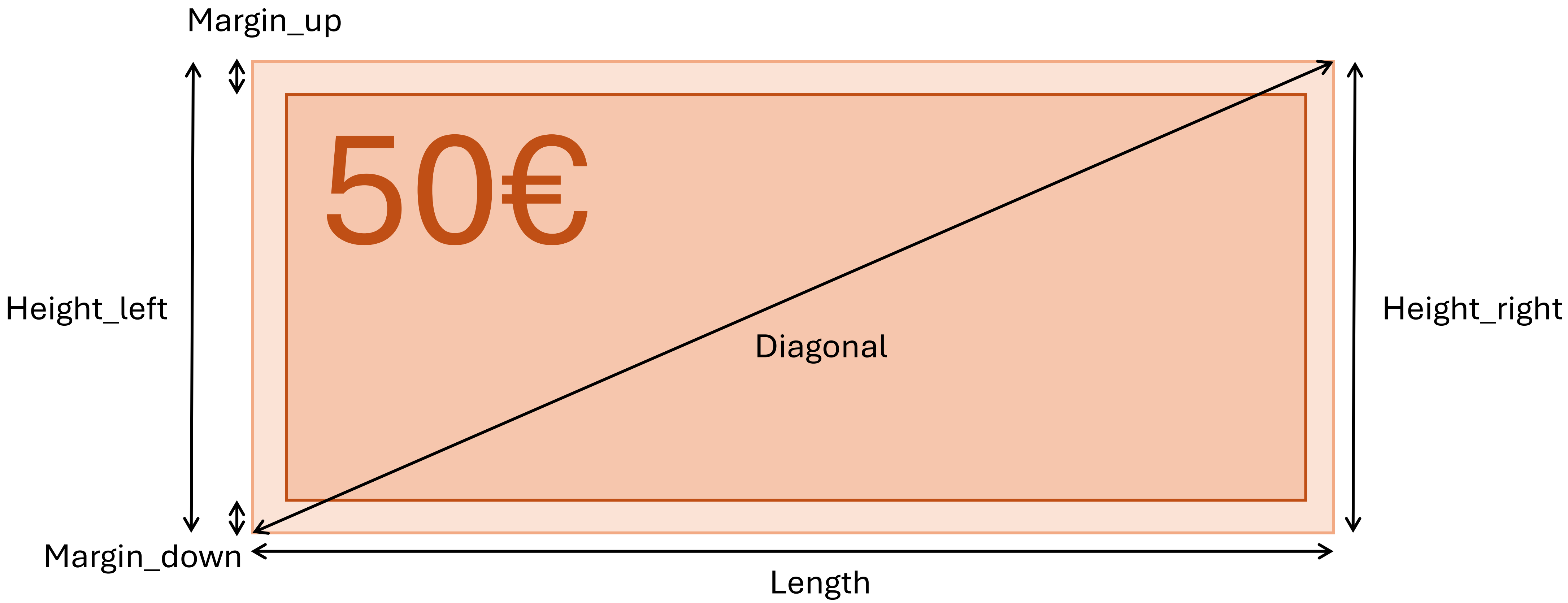


554 000 counterfeit notes seized in 2024
+20% compared to 2023

→ Counterfeight rise must be controlled now
→ We need efficient, scalable methods

Assess counterfeiting with geometric data

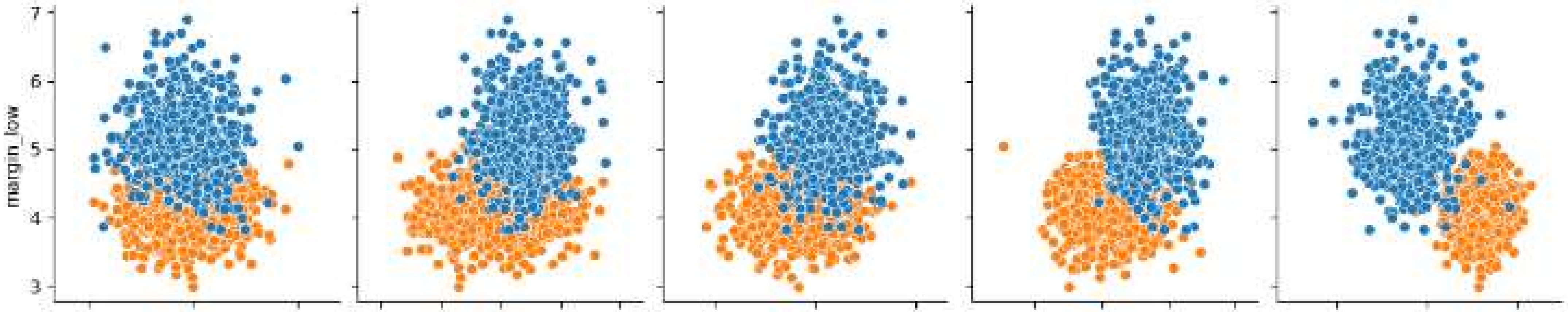
Can we detect fake notes based on geometric data?



Fill missing values with linear regression

The assumptions of the multiple linear regression are not met

1) Linear relationship between dependent and explanatory variables



diagonal	Height_left	Height_right	Margin_up	Length
R-squared 0.012	R-squared 0.09	R-squared 0.15	R-squared 0.18	R-squared 0.44

→ We take the variables with R-squared > 0.1

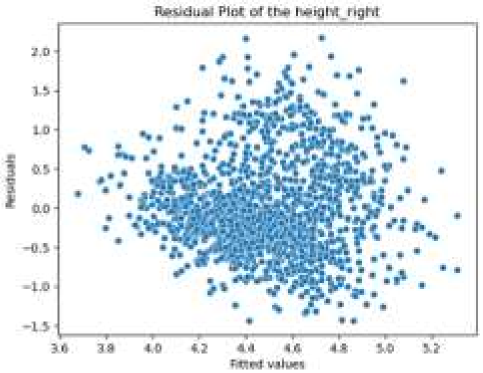
Fill missing values with linear regression

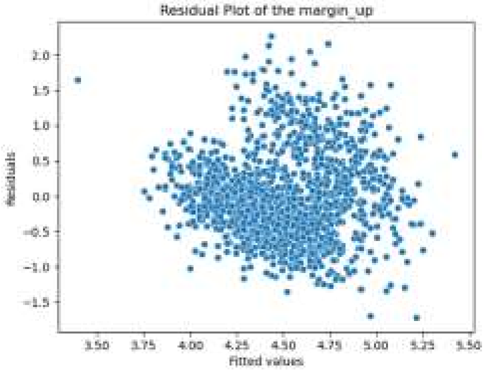
The assumptions of the multiple linear regression are not met

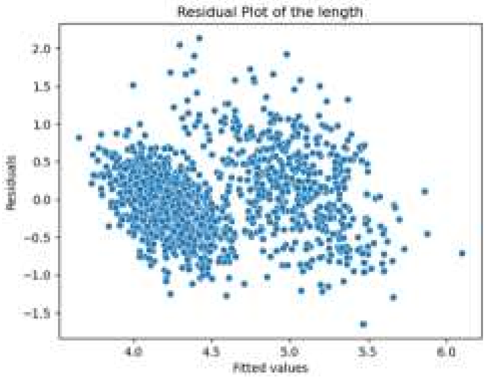
2) Independence of the residuals: Durbin-Watson statistic

Height_right	Margin_up	Length
1.18	1.32	1.86
Positive autocorrelation	Positive autocorrelation	

3) Homoscedasticity of the residuals







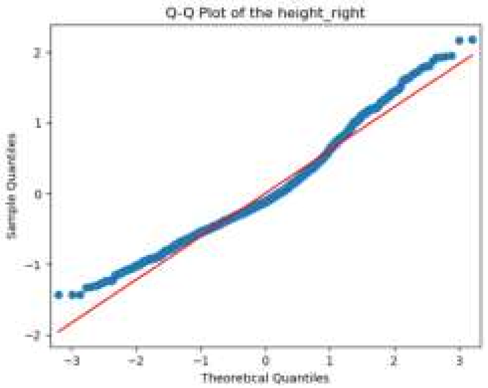
We see clear patterns on every plot

Fill missing values with linear regression

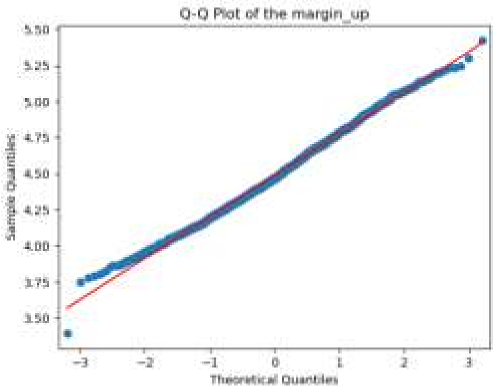
The assumptions of the multiple linear regression are not met

4) Normality of the residuals

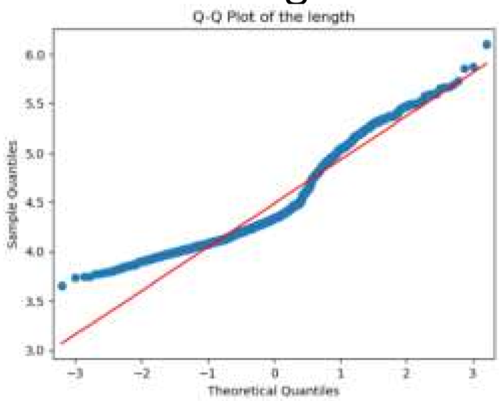
Height_right



Margin_up



Length



The residuals of the length diverge strongly from normality.

Results

No variable is fit for a linear regression with margin_low

If we do the linear regression anyway: R-squared = 0.44

Can we do better?

Fill missing values with Random Forest

Default Random Forest cannot be used

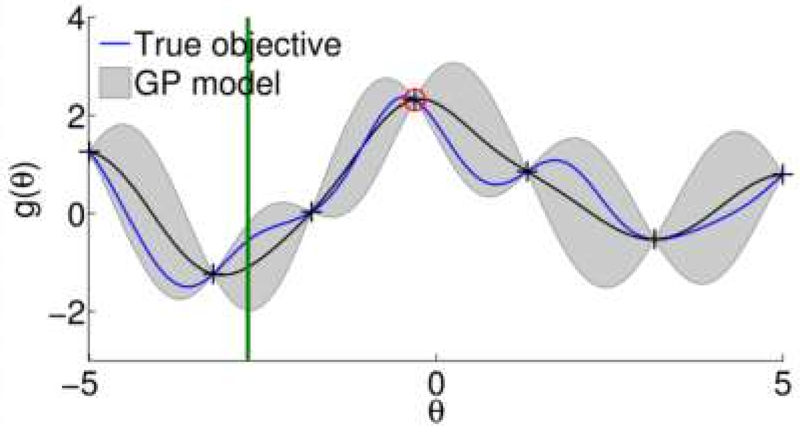
Default parameters results
R-squared on training set: **0.94**
R-squared on validation set: **0.56**

Better than linear regression
Badly overfitted

Random Forest hyperparameters

Max_depth
Min_sample_leaf
Min_sample_split
N_estimators

Bayesian search



Metric to maximise: R-squared

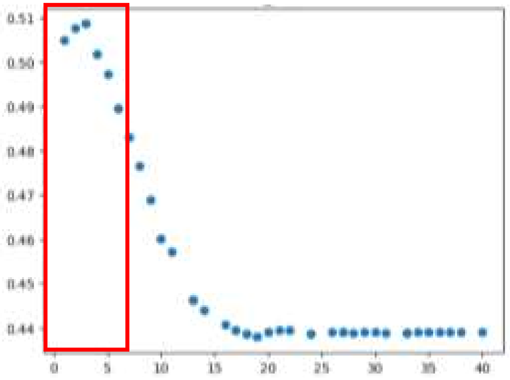
Filling data missing values

NA prediction is challenging with both methods, but the RF is better

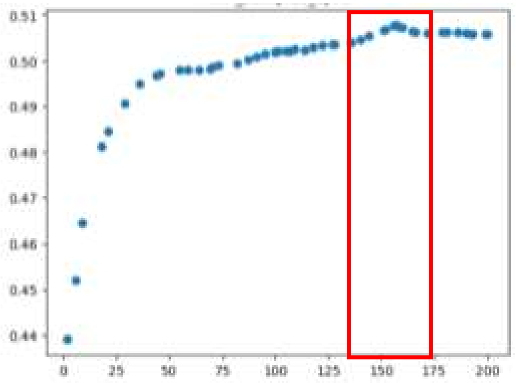
Bayesian Search in 2 steps

1 – Scan the parameters separately

Max depth



Min sample split



R^2

'sweet range'

2 – Scan all parameters over the 'sweet ranges'

Best parameters:
Max_depth = 2
Min_sample_leaf = 81
Min_sample_split = 100
N_estimators = 100

Results

R-squared on training set: **0.62**
R-squared on validation set: **0.62**

Not overfitted
6% better than before

Counterfeight notes detection

List of algorithm tested

Supervised machine learning

- K-Nearest Neighbors
- Random Forest
- Logistic Regression
- Extreme Gradient Boosting

"Glassbox" model

- Explainable boosting

Unsupervised machine learning

- Gaussian Mixture Model

Accuracy metrics used

$$\boldsymbol{Precision} = \frac{TP}{TP + FP}$$

Higher values reduces false positives.

$$\boldsymbol{Recall} = \frac{TP}{TP + FN}$$

Higher values reduces false negatives.

$$\boxed{\boldsymbol{F1} = 2 * \frac{Precision * Recall}{Precision + Recall}}$$

Balances precision and recall.

Used for model optimisation

Optimization strategy for each model type

Supervised machine learning	**Explainable boosting**	**Gaussian Mixture Model**
Hyperparameter tuning (2-step Bayesian Search)		Define number of clusters

Model fitness

Cluster ID interpretation

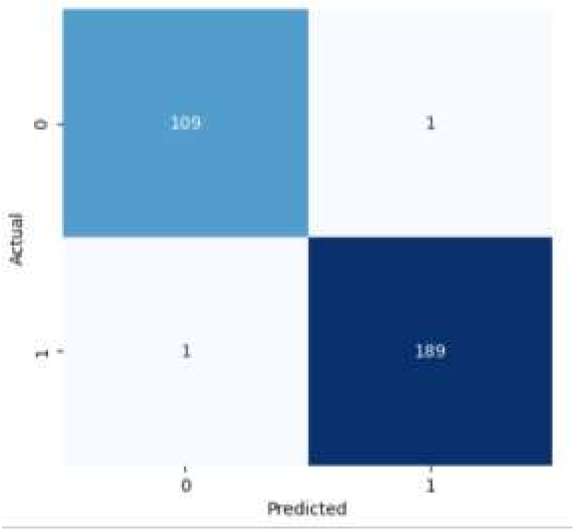
Accuracy assessment

F1 of training set vs F1 of validation set

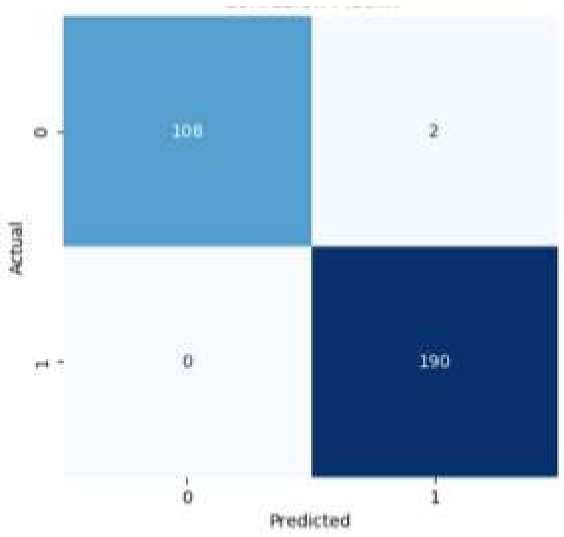
Confusion matrix

Confusion matrices for each model

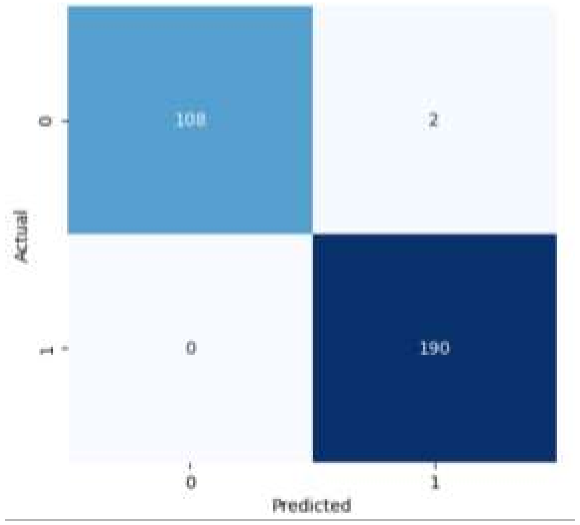
KNN



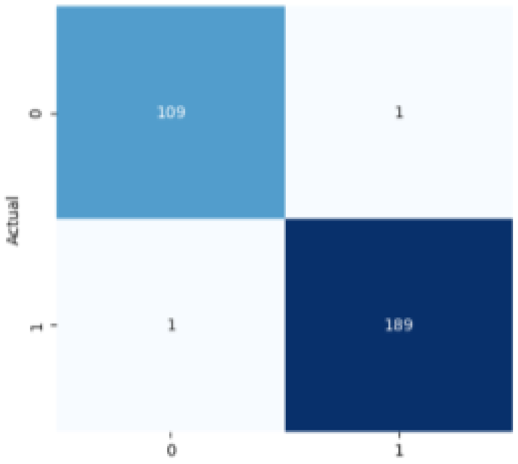
Random Forest



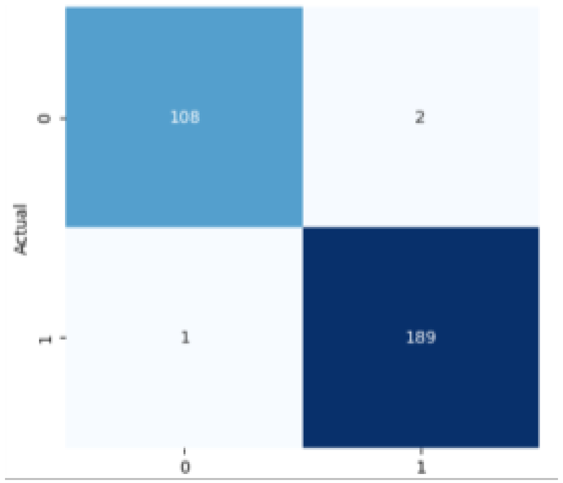
Logistic regression



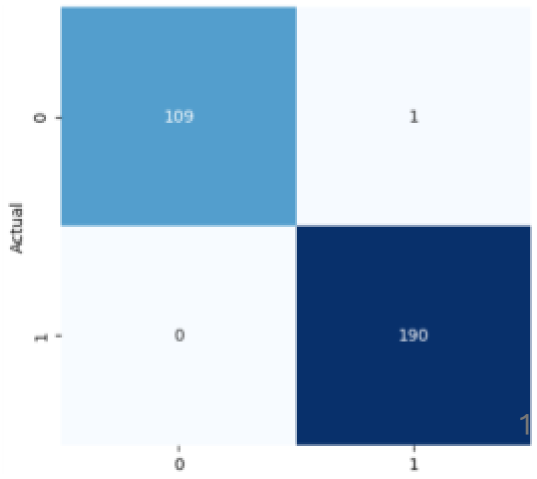
XGBoost



Explainable boosting

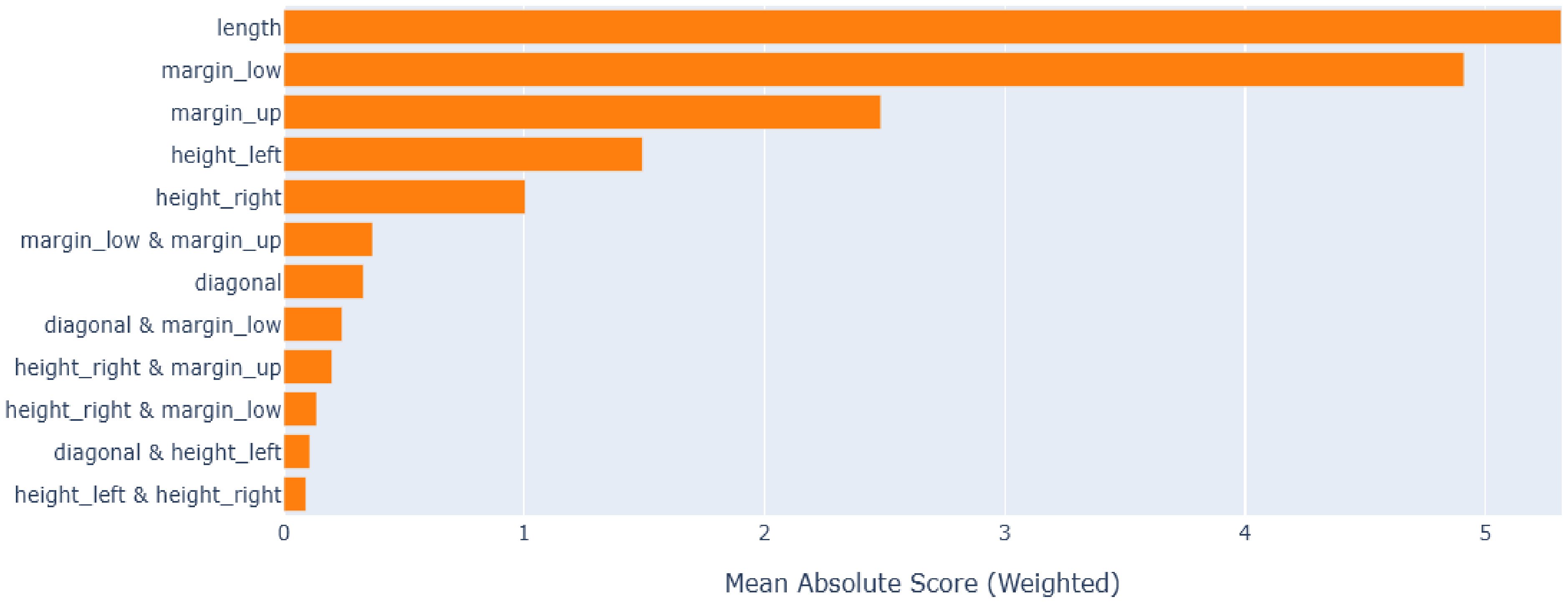


Gaussian Mixture Model



Relative importance of each parameter

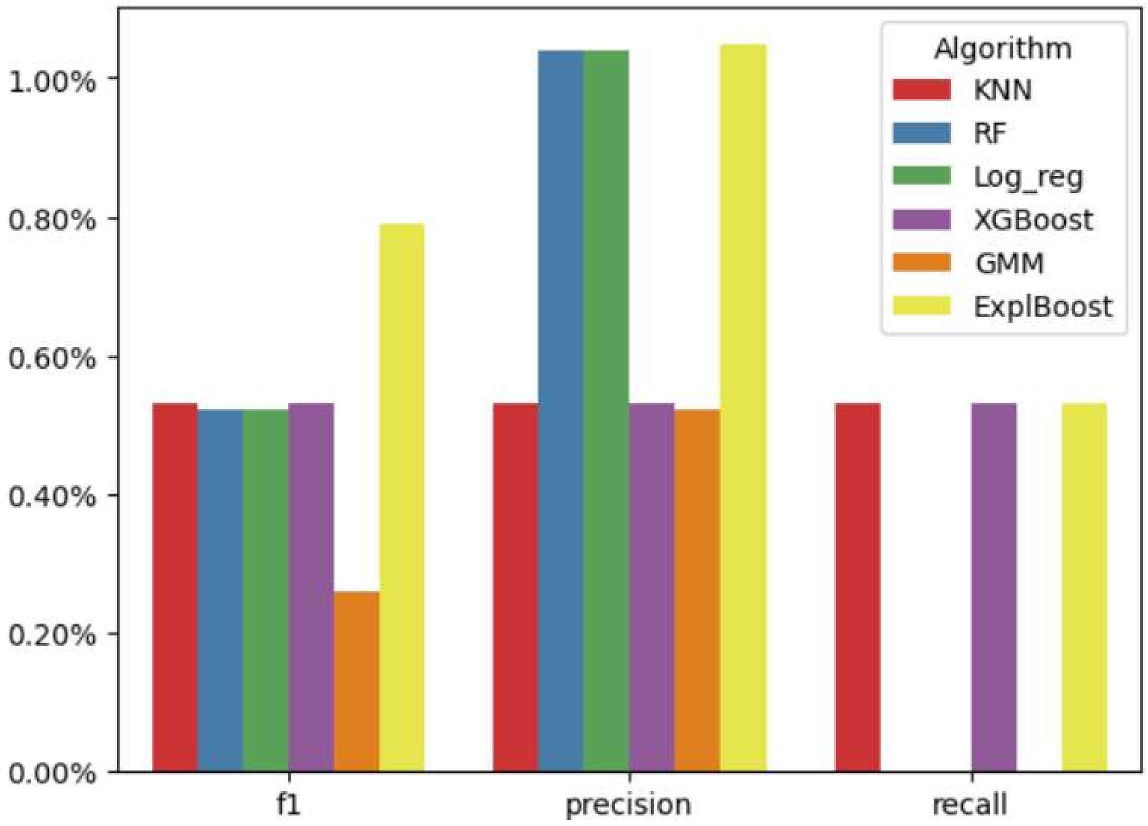
The length and margin_low are by far the most important predictive variables



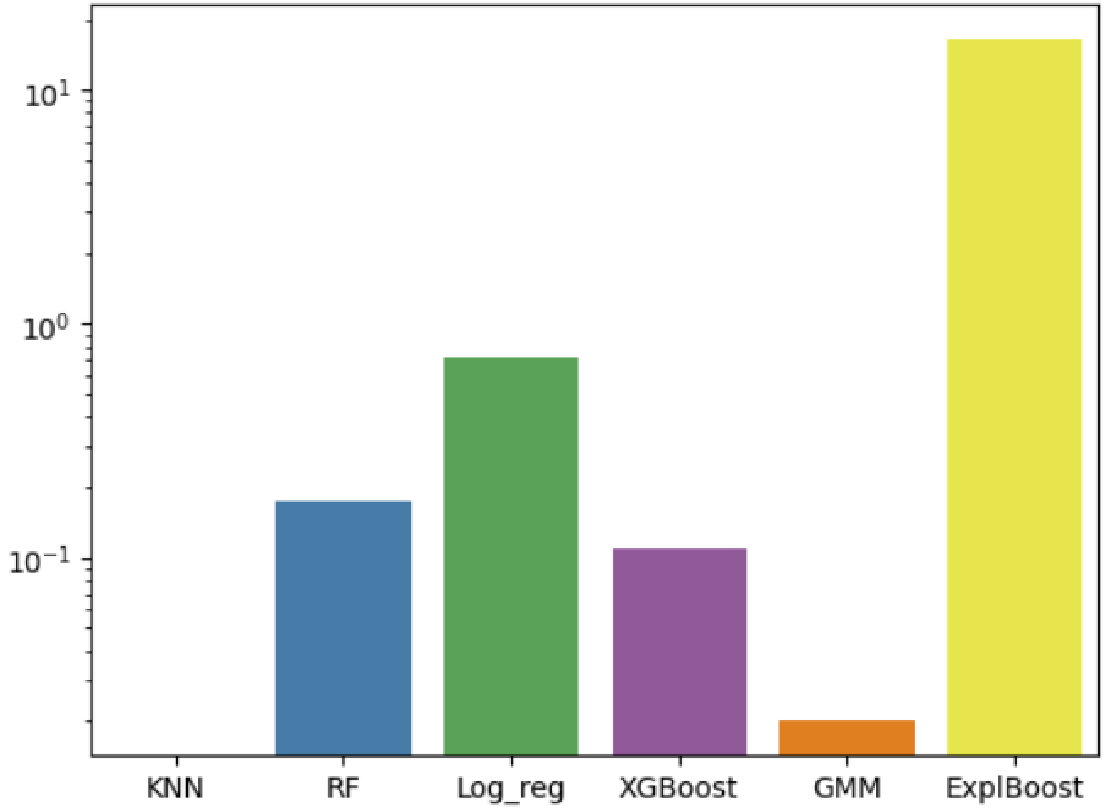
Model comparison

The Gaussian Mixture Model is very fast and accurate

Inaccuracy of the model (1 - metric)



Fitness time in seconds



Thank you for your attention