
RÉPUBLIQUE FRANÇAISE

*Liberté*
*Égalité*
*Fraternité*

# Détectez des faux billets

Parcours Data Analyste

MARCHAND Emmanuel – 11/2024

# Agenda

- **Partie 1 :** Préparation des données
- **Partie 2 :** Test de différents modèles
- **Partie 3 :** Application finale

# **Partie 1 :** Préparation des données

# Description des données source

Fichier billet


**Structure du fichier :**

- **1** variable **target** : Is_genuine.
- **5** variables **prédictives** :
  - Diagonal
  - Height_left
  - Height_right — *Dimensions du billet*
  - Margin_low
  - Margin_up — *Marges du billet*
  - length

**Contenu du fichier :**

- **1500** billets
- **37** données de Margin_low manquantes
- 66,67% de Vrais billets
- 33,33% de Faux billets

# Prédiction des valeurs manquantes de 'margin_law'

Fichier billet

.csv

Données en entrée des tests : pas de margin_law à Nan **(1463 éléments)**

On teste différentes régressions :

- Régression 1 : Y ~ diagonal + margin_up
- Régression 2 : Y ~ diagonal + height_left + height_right + margin_up
- Régression 3 : Y ~ diagonal + height_left + height_right + margin_up + length

- Entrainement du modèle
- Métriques de performance du modèle

| | Model | R^2 | RMSE | MAE | MAPE |
|---|---|---|---|---|---|
| 0 | Régression 1 : Y ~ diagonal + margin_up | 0.193707 | 0.595859 | 0.460748 | 0.101883 |
| 1 | Régression 2 : Y ~ diagonal + height_left + height_right + margin_up | 0.293681 | 0.557696 | 0.43196 | 0.0956943 |
| 2 | Régression 3 : Y ~ diagonal + height_left + height_right + margin_up + length | 0.477337 | 0.479742 | 0.372236 | 0.0825518 |

On utilise la troisième régression qui a **le meilleur $R^2$ et le plus faible coût**

Données en entrée du modèle : Uniquement margin_law Nan

**(37 éléments)**

```
Coefficients du modèle:
[-0.03391242  0.05514522  0.08333082  0.0592785  -0.35711543]
Intercept du modèle:
4.485967190704031
```

# Validation du modèle de régression linéaire

**ETAPE 1:**
**Ajustement du modèle** de régression linéaire.

- Ajout d'une colonne de constante (des 1)
- Ajustement du modèle avec la méthode OLS (Ordinary Least Squares) de la librairie 'statsmodel'

**ETAPE 2 : Vérification des hypothèses de validité** du modèle.

On utilise plusieurs trois méthodes différentes.

### Colinéarité des variables.

On utilise le facteur d'inflation de la variance (VIF).

| | feature | VIF |
|---|---|---|
| 0 | const | 591443.076346 |
| 1 | diagonal | 1.012790 |
| 2 | height_left | 1.145295 |
| 3 | height_right | 1.229263 |
| 4 | margin_up | 1.403517 |
| 5 | length | 1.574765 |

**Résultat :**

Comme les valeurs de VIF sont inférieures à 10, il n'y a pas de colinéarité problématique. Ce qui est une bonne chose pour notre modèle de régression linéaire.

### Vérification de l'homoscédasticité.

On utilise le test de Breusch-Pagan

```
p-value: 3.5301066698453908e-16
```

**Résultat :**

La p-value est nettement inférieure à 0.05, on rejette l'hypothèse nulle ce qui signifie qu'il y **a une hétéroscédasticité présente**.

### Vérification de la normalité des résidus.

On utilise le test de Shapiro-Wilk

```
p-value: 1.8450612565557023e-11
```

**Résultat :**

La valeur p est inférieure à 0.05, donc on rejette l'hypothèse nulle. Cela signifie qu'il y a une preuve suffisante pour conclure que les données **ne suivent pas une distribution normale.**

**ETAPE 3 : Test de la performance du modèle**

On utilise a validation croisée K-Fold

```
Mean Squared Error (MSE) pour chaque pli: [0.19144955 0.
Moyenne des MSE pour tous les plis: 0.2322388732106774
Ecart type des MSE: 0.03154612239480658
```

**Résultats :**

- La MSE quantifie la différence moyenne entre les valeurs prédites par le modèle et les valeurs réelles observées. Pour chaque pli, celle-ci est faible il y a peu d'erreur.
- La moyenne des MSE est faible donc le modèle à une bonne performance globale.
- L'écart type des MSE est faible donc le modèle est stable et ses performances ne varient pas beaucoup d'un pli à l'autre.

**CONCLUSION GENERALE**

Comme il n'y a pas de colinéarité problématique et que le modèle généralise bien **on va conserver les données de 'margin_low' imputés par régression linéaire.**

# Identification des outliers

## Résumé statistique des variables du Fichier billet corrigé

| | diagonal | height_left | height_right | margin_low | margin_up | length |
|---|---|---|---|---|---|---|
| count | 1500.000000 | 1500.000000 | 1500.000000 | 1500.000000 | 1500.000000 | 1500.00000 |
| mean | 171.958440 | 104.029533 | 103.920307 | 4.483475 | 3.151473 | 112.67850 |
| std | 0.305195 | 0.299462 | 0.325627 | 0.659632 | 0.231813 | 0.87273 |
| min | 171.040000 | 103.140000 | 102.820000 | 2.980000 | 2.270000 | 109.49000 |
| 25% | 171.750000 | 103.820000 | 103.710000 | 4.020000 | 2.990000 | 112.03000 |
| 50% | 171.960000 | 104.040000 | 103.920000 | 4.310000 | 3.140000 | 112.96000 |
| 75% | 172.170000 | 104.230000 | 104.150000 | 4.870000 | 3.310000 | 113.34000 |
| max | 173.010000 | 104.880000 | 104.950000 | 6.900000 | 3.910000 | 114.44000 |

On note une bonne répartition des données,

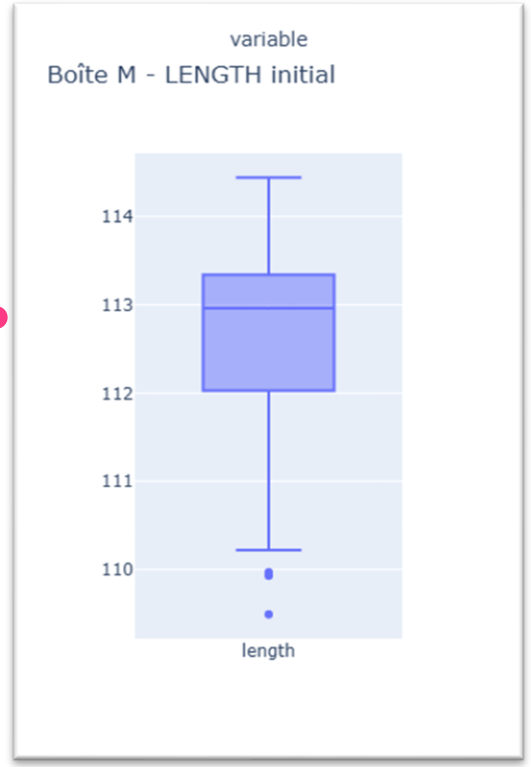
seul les colonnes **'margin_low' et 'lenght'** ont **des écarts type supérieurs** au reste des variables.

# Analyse des outliers





Outliers au dessous de la **moustache haute**




Outliers en dessous de la **moustache basse**

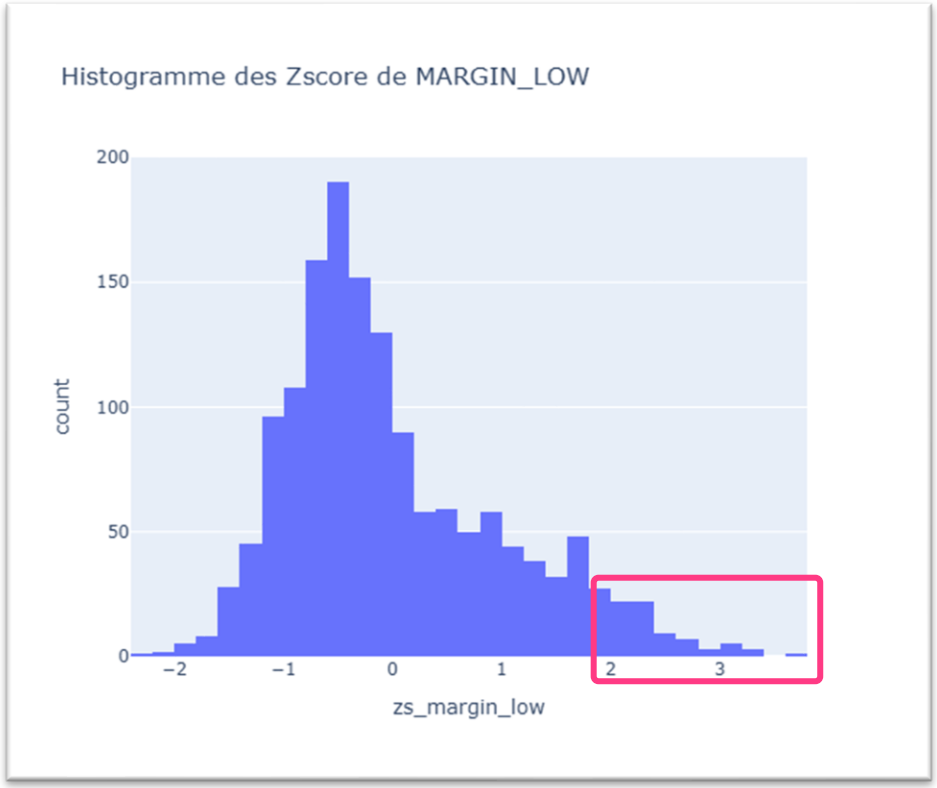
# Identification des outliers

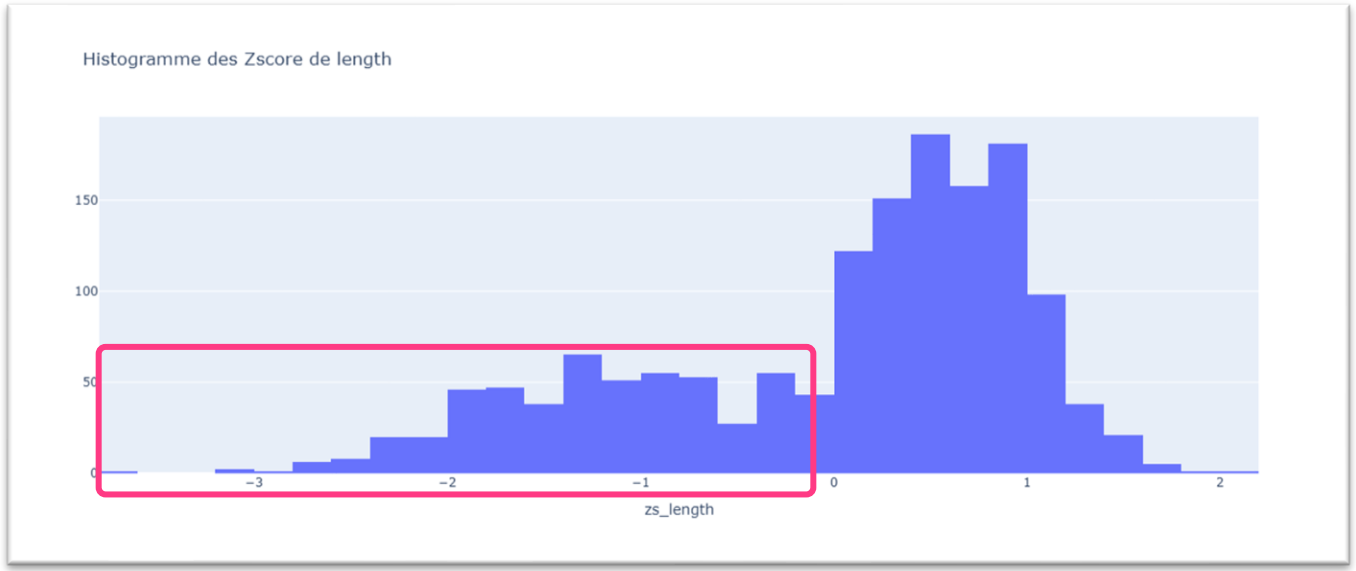
On identifie les Outliers en utilisant le **Zscore.**

Le z-score mesure de combien d'écarts types une valeur est éloignée de la moyenne de la variable. On considère **qu'un Z-score supérieur à 2 ou 3** correspond à un Outlier.

## MARGIN_LOW

On voit **quelques outliers** dont le Zscore est supérieur à 2.




## LENGTH

Il y a seulement un outliers à peine supérieur à 2.
PAR CONTRE on note **de nombreux outliers avec un Zscore négatif**.

# Transformation des outliers

On doit traiter les Outliers des deux variables **'margin_low'** et **'length',** car ils auront une influence (disproportions de valeurs) sur les résultats des modèles de machine learning que nous allons utiliser.

Pour ne pas perdre trop d'informations , on ne va pas supprimer les outliers, mais leur **appliquer une transformation.**

**D'un point de vue statistique** il est recommandé d'appliquer la log +1 sur les Outliers, pour réduire la dispersion de la distribution et l'influence des outliers. Cela est valable si la distribution contient des Min et des Max très éloignés. Si ce n'est pas le cas, en appliquant cette méthode statistique nous risquons d'introduire d'autre outliers (exemple log de 115 est 2,06).

**D'un point de vue métier** nous savons que les dimensions des billets sont comprises entre des seuils que nous pouvons fixer, au dela de ces seuils le billet est forcément Faux. Dans ce cas nous pouvons supprimer l'Outlier car nous ne perdons pas d'information, nous avons l'information que le billet est Faux.

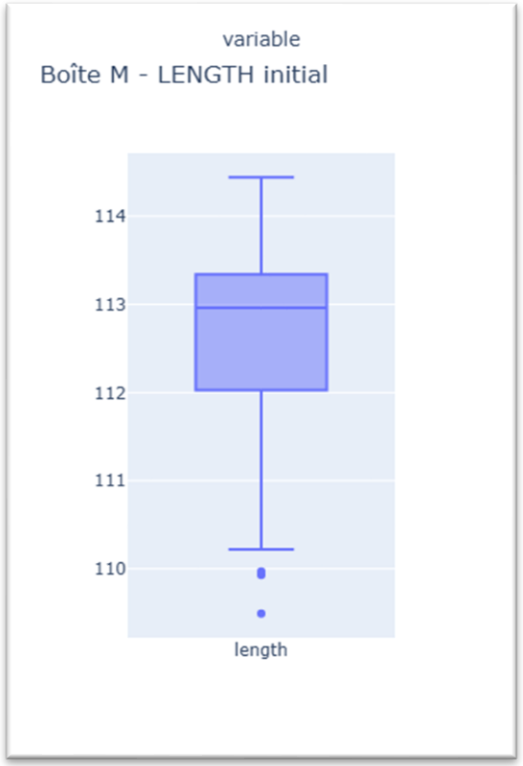
**REGLES :**

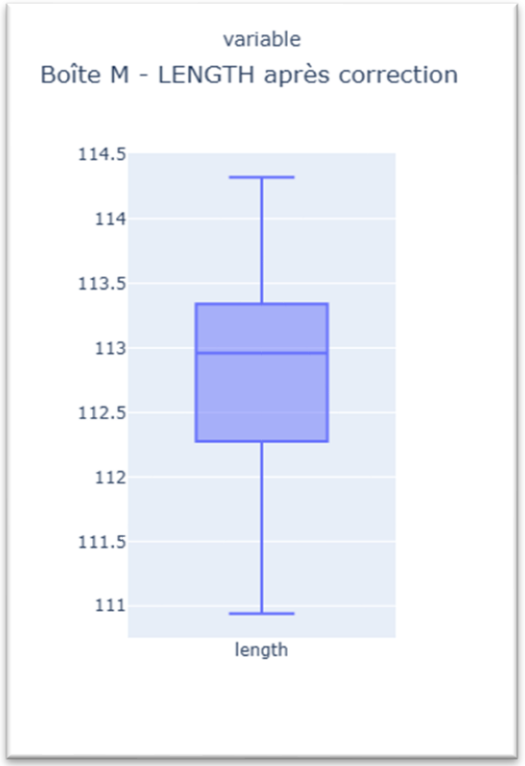
Les valeurs identifiées comme Outliers seront transformées :

- ❑ pour les dimensions incluses dans les seuils on leur attribuera **la valeur médiane de la distribution**
- ❑ pour les dimensions en dehors des seuils **on les supprime.**

# Résultats de la transformation de Length et Margin_low

## LENGTH



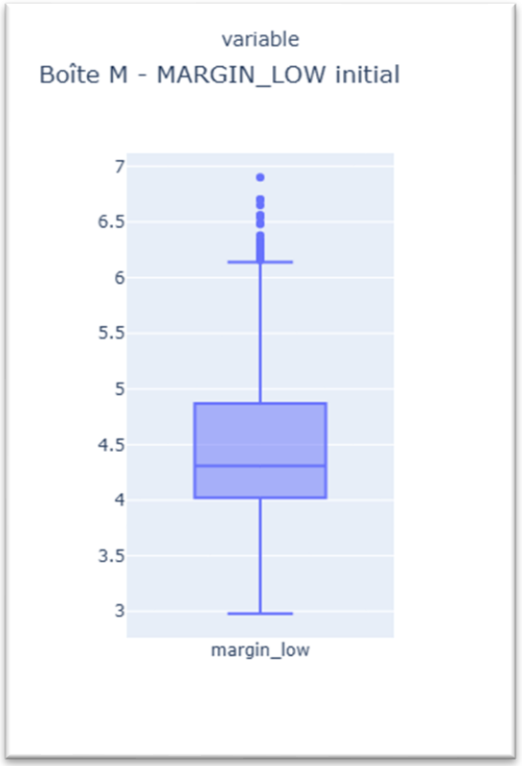

**Avant** **Après**

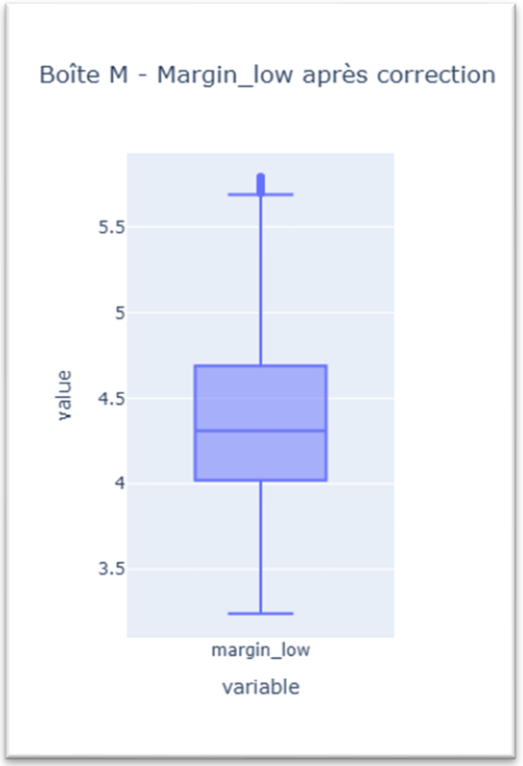
**Règles :**

Si ZScore <-2 ou > 2
- si **length** n'est pas compris entre 110 et 115 => delete
- si **length** est compris entre 110 et 115 => médiane de la distribution

- nb de lignes lues : 1500
- nb de lignes supprimées : 3
- nb de lignes modifiées : 56

## MARGIN_LOW




**Avant** **Après**

**Règles :**

Si ZScore <-2 ou > 2
- si **margin_low** n'est pas compris entre 3 et 7 => delete
- si **margin_low** est compris entre 3 et 7 => médiane de la distribution

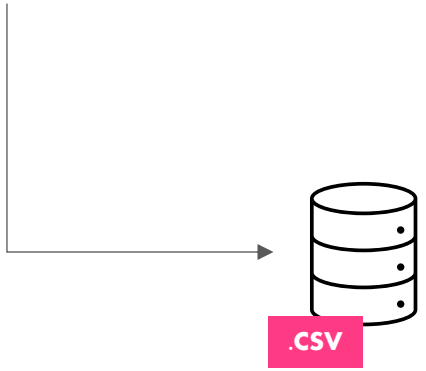
- nb de lignes lues : 1497
- nb de lignes supprimées : 1
- nb de lignes modifiées : 76

# Bilan de la préparation des données

**37** Margin_low calculées par régression Linéaire

**4** Outliers supprimés et considérés comme des billets faux

**132** Outliers modifiés pour ne pas perdre de l'information




**1496**
éléments

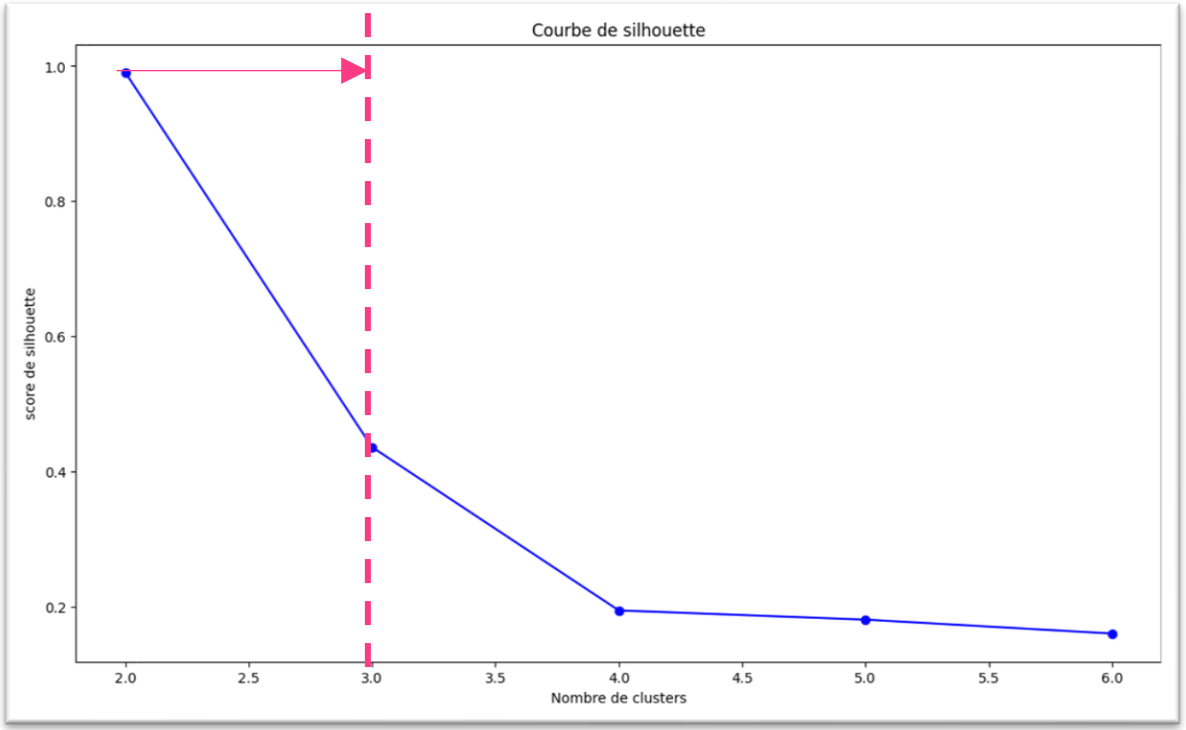
Nous disposons d'un fichier adapté pour trouver le modèle le plus fiable pour détecter les faux billets.

Il servira également de données d'entrainement pour le modèle qui sera retenu (en mode production).

**Partie 2 :** Tests de différents modèles.

# Test du modèle Kmeans

Il y a une nette différence entre 2 et 3 clusters. On **va garder 2 clusters**, ce qui correspond à un **cluster avec les vrais** billets et **un autre avec les faux billets**.




*On choisit arbitrairement 6 clusters et pour chacun d'eux on calcule le score de silhouette.*

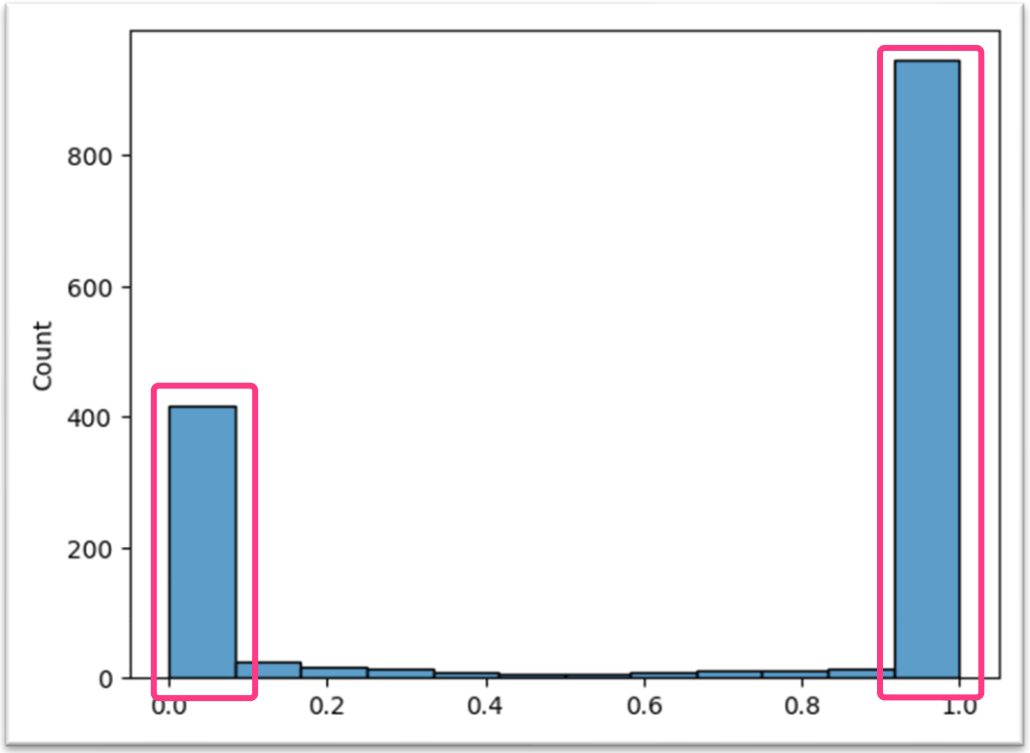
**Résultats du Kmeans avec Deux clusters**.

```
Matrice de confusion:
[[431  66]
 [  6 993]]

accuracy_score :  0.9518716577540107
```

- La précision est **de 95,2% ce qui est un très bon score.**.
- **1424** résultats corrects sur les 1496 entrées.
- Le silhouette_score est de **47%** ce qui est faible.

# Test du modèle Régression logistique

Le modèle **est assez fiable** au niveau des prédictions, la plupart des prédictions ont une probabilité proche de O ou de 1..



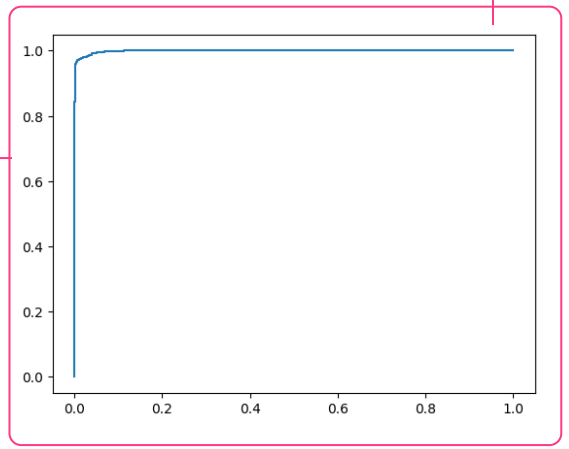

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.94 | 0.96 | 497 |
| 1 | 0.97 | 0.99 | 0.98 | 999 |
| accuracy | | | 0.97 | 1496 |
| macro avg | 0.97 | 0.96 | 0.97 | 1496 |
| weighted avg | 0.97 | 0.97 | 0.97 | 1496 |

**Rapport de classification du modèle Régression logistique.**

**Résultats de la Régression Logistique.**

```
Matrice de confusion:
[[467  30]
 [ 10 989]]

accuracy_score 0.9732620320855615
```

- La précision est **de 97,3% ce qui est excellent**.
- Le modèle a trouvé **1456 bonnes réponses** sur les 1496 entrées**.**




*Le tracé du ROC confirme que le modèle est bon, la courbe monte rapidement vers le classificateur optimal (1 sur l'axe des y)*

# Test du modèle KNN – K-Nearest Neighbors

**Résultats du modèle KNN.**

```
Matrice de confusion :
[[100   7]
 [  4 189]]
```

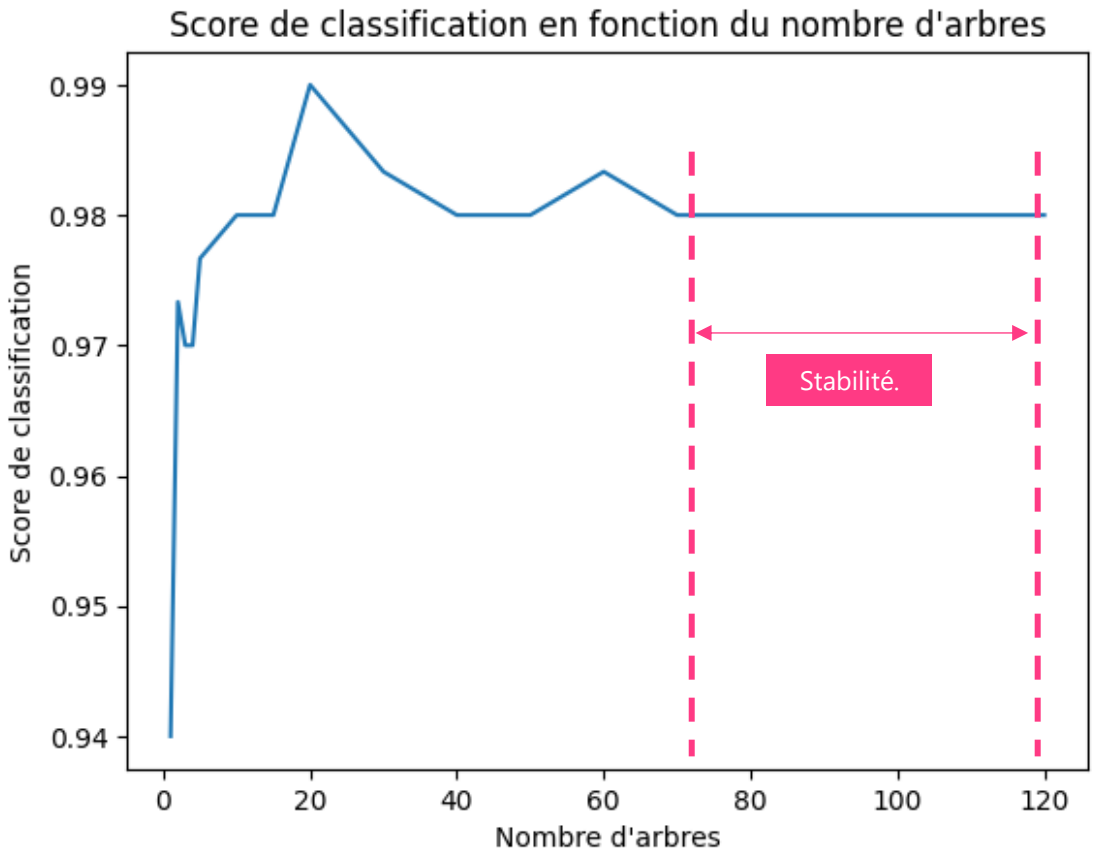
- Le modèle a trouvé **289 bonnes réponses** sur les 300 entrées**.**

**Rapport de classification du modèle KNN.**

```
              precision    recall  f1-score   support

           0       0.96      0.93      0.95       107
           1       0.96      0.98      0.97       193

    accuracy                           0.96       300
   macro avg       0.96      0.96      0.96       300
weighted avg       0.96      0.96      0.96       300

Train score 0.975752508361204
Test score 0.9633333333333334
```

- Le modèle a une précision de **96%** pour détecter **les faux billets et les vrais billets.**
- Il **détecte mieux les vrais billets** (recall de 98%)
- La précision est **96%.**
- Le modèle colle aux données d'entraînement (le score(train) est excellent) mais il ne reproduit pas la même performance sur les données de test. Il y a un **léger overfitting**

# Test du modèle Random Forest




Variation du score de classification du modèle en fonction du nombre d'arbres.

**Résultats du Random Forest avec 80 Arbres.**

```
Accuracy: 0.9633333333333334
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.92      0.95       107
           1       0.95      0.99      0.97       193

    accuracy                           0.96       300
   macro avg       0.97      0.95      0.96       300
weighted avg       0.96      0.96      0.96       300


Train score 0.9707357859531772
Test score 0.9633333333333334
```

- La précision est **de 98%** pour **les faux billets** et **95% pour les vrais.**
- Le modèle détecte **99% des vrais billets** et **92% des faux**.
- La précision est **de 96,3%**
- Le modèle colle aux données d'entraînement (le score(train) est excellent) mais il ne reproduit pas la même performance sur les données de test. Il y a un **léger overfitting**

# Choix du modèle

| Modèle | Classe | Précision | Recall | F1-Score | Support |
|---|---|---|---|---|---|
| Kmeans | 0 | 95,2% | na | na | na |
| | 1 | | na | na | na |
| Régression logistique | 0 | 98% | 94% | 96% | 497 |
| | 1 | 97% | 99% | 98% | 999 |
| K-Nearest Neighbors | 0 | 96% | 93% | 95% | 107 |
| | 1 | 96% | 98% | 97% | 193 |
| Random Forest | 0 | 98% | 92% | 95% | 107 |
| | 1 | 95% | 99% | 97% | 193 |

Modèle non supervisé.

Nous retenons le modèle **Régression logistique**, il obtient les meilleurs score pour la précision et le recall des faux billets ce qui est notre priorité.

**Partie 3 :** Application finale.

# Processus d'analyse du fichier

1

Chargement du fichier

- Nom fichier

2

Détection faux billets

- Entrainement avec le fichier du prototype
- Test sur le fichier production

3

Résultat

- Dataframe de résultat
- Ajout colonne 'Is_genuine' qui indique si le billet est authentique (1) ou non (0)
- Et une colonne 'probability' qui indique la probabilité de la réponse.

| | diagonal | height_left | height_right | margin_low | margin_up | length | id | is_genuine | probability |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 172.09 | 103.95 | 103.73 | 4.39 | 3.09 | 113.19 | B_1 | 1 | 0.981143 |
| 1 | 171.52 | 104.17 | 104.03 | 5.27 | 3.16 | 111.82 | B_2 | 0 | 0.003307 |
| 2 | 171.78 | 103.80 | 103.75 | 3.81 | 3.24 | 113.39 | B_3 | 1 | 0.997898 |
| 3 | 172.02 | 104.08 | 103.99 | 5.57 | 3.30 | 111.10 | B_4 | 0 | 0.000075 |
| 4 | 171.79 | 104.34 | 104.37 | 5.00 | 3.07 | 111.87 | B_5 | 0 | 0.005462 |

RÉPUBLIQUE
FRANÇAISE

*Liberté*
*Égalité*
*Fraternité*


# MERCI