DETECTEUR DE FAUX BILLETS

OpenClassrooms - Parcours Data Analyst V2

Projet 10 - Valérie Chadeau

SOMMAIRE

- Le contexte
- Partie 1 : Chargement, exploration & Nettoyage des données
- Partie 2 : La Mission
    - Analyses univariées
    - Analyse bivariées
    - Les modélisations
        - Méthode apprentissage non supervisé : Kmeans
        - Méthode apprentissage supervisé : régression logistique
- Test du programme de détecteur de faux billets

Contexte

La mission

Consultant Data Analyst dans une entreprise spécialisée dans la data, je suis en prestation pour **l'Organisation nationale de lutte contre le faux-monnayage (ONCFM)** qui a pour objectif la mise en place des méthodes d'identification des contrefaçons des billets en euros à partir de certaines caractéristiques du billet (dimensions, positionnement des marges).

Objectif

Mise en place d'un algorithme capable de différencier automatiquement les vrais des faux billets à partir de caractéristiques physiques du billet relevées par une machine.

Cet algorithme sera programmé en Python.

PARTIE 1

Chargement, exploration

Nettoyage des données

DONNEES INITIALES

Données issues du fichier “billets.csv” :

billets.csv

Liste de billets avec leurs caractéristiques physiques + indicateur billet vrai ou faux

contient

- **1500** lignes
- **7** variables (6 quantitatives, 1 qualitative)
- **37 valeurs manquantes pour la variable "margin_low"** (environ 2,4% des données)

Description des variables

Nom	Description	Type	Unité
is_genuine	Indicateur billet vrai ou faux	qualitative (booléen)	
length	Longueur	quantitative	mm
height_left	Hauteur sur le côté gauche	quantitative	mm
height_right	Hauteur sur le côté droit	quantitative	mm
margin_up	Marge supérieure	quantitative	mm
margin_low	Marge inférieure	quantitative	mm
diagonal	Diagonale	quantitative	mm

TRAITEMENT DES DONNEES MANQUANTES

➔ **Méthode imputation données manquantes : régression linéaire multiple**

Remplacement des valeurs manquantes par des valeurs prédites grâce à une régression linéaire multiple avec :

* **variable réponse :**
  margin_low

* **variables explicatives :**
  X1=diagonal
  X2=height_left
  X3=height_right
  X4=margin_up
  X5=length

Le dataframe initial a été divisé en 2 parties :

* un **dataframe sans valeur manquantes** (pour entrainer le modèle de régression linéaire)
* un **dataframe ne contenant que les lignes avec les valeurs manquantes** sur lequel a été appliqué le modèle

Régression testée avec les 2 librairies StatsModels et Scikit-Learn avec un niveau de risque α=5 %.

➔ Méthode imputation données manquantes : régression linéaire multiple

* Suppression des variables non significatives afin d’optimiser le modèle

* **R² ajusté (présence de plusieurs variables) = 0,473** qui n'est pas très élevé (seul 47,3% des observations peuvent être expliqués par le modèle)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             margin_low   R-squared:                       0.477
Model:                            OLS   Adj. R-squared:                  0.476
Method:                 Least Squares   F-statistic:                     266.1
Date:                Mon, 25 Jul 2022   Prob (F-statistic):          2.60e-202
Time:                        09:35:42   Log-Likelihood:                -1001.3
No. Observations:                1463   AIC:                             2015.
Df Residuals:                    1457   BIC:                             2046.
Df Model:                           5
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       22.9948      9.656      2.382      0.017       4.055      41.935
diagonal        -0.1111      0.041     -2.680      0.007      -0.192      -0.030
height_left      0.1841      0.045      4.113      0.000       0.096       0.272
height_right     0.2571      0.043      5.978      0.000       0.173       0.342
margin_up        0.2562      0.064      3.980      0.000       0.130       0.382
length          -0.4091      0.018    -22.627      0.000      -0.445      -0.374
==============================================================================
Omnibus:                       73.627   Durbin-Watson:                   1.893
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               95.862
Skew:                           0.482   Prob(JB):                     1.53e-21
Kurtosis:                       3.801   Cond. No.                     1.94e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.94e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Suppression des variables non significatives avec p-value < 0,05

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             margin_low   R-squared:                       0.475
Model:                            OLS   Adj. R-squared:                  0.473
Method:                 Least Squares   F-statistic:                     329.5
Date:                Mon, 25 Jul 2022   Prob (F-statistic):          4.80e-202
Time:                        09:35:42   Log-Likelihood:                -1004.9
No. Observations:                1463   AIC:                             2020.
Df Residuals:                    1458   BIC:                             2046.
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept        5.2656      7.048      0.747      0.455      -8.560      19.091
height_left      0.1779      0.045      3.971      0.000       0.090       0.266
height_right     0.2550      0.043      5.917      0.000       0.170       0.340
margin_up        0.2588      0.064      4.012      0.000       0.132       0.385
length          -0.4136      0.018    -22.930      0.000      -0.449      -0.378
==============================================================================
Omnibus:                       75.632   Durbin-Watson:                   1.885
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               99.244
Skew:                           0.489   Prob(JB):                     2.81e-22
Kurtosis:                       3.820   Cond. No.                     1.04e+05
==============================================================================
```

TRAITEMENT DES DONNEES MANQUANTES

➔ **Validation pré-requis régression linéaire**

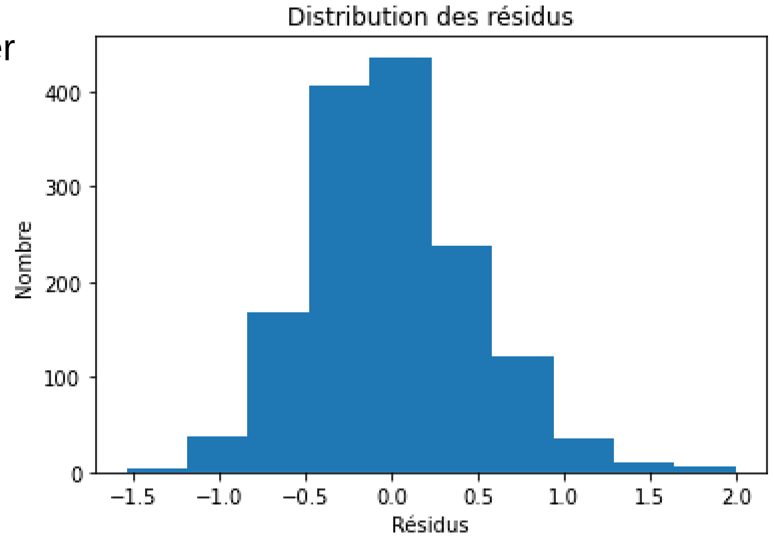
<u>1. Normalité des résidus</u>

L'objectif est de s'assurer que les résidus suivent une loi normale via **le test de Shapiro-Wilk (échantillon** < 5000 observations). On pose :

H0 : Les résidus suivent une loi normale si p-value > 5%
H1 : Les résidus ne suivent pas une loi normale si p-value < 5%

Comme p-value = 5.929039204044528e-11 < 5% ==> on rejette H0 : **les résidus ne suivent pas une loi normale.**

D'autres tests ont été utilisés (**QQPlot, Test Jarque-Berra**) afin de valider ce résultat.



2. L’égalité des variances (homoscédasticité)

La variance des résidus doit être constante : on vérifie cette condition via le test statistique **Breusch-Pagan.**

On pose :

H0 : Les résidus présentent une égalité de variance (il y a homoscédasticité) si p-value > 5%
H1 : Les résidus ne présentent pas une égalité de variance si p-value < 5%

➔ Comme p_value= 7.759535216174283e-16 < α (= 5%), on peut donc rejeter l'hypothèse H0 : il y a **hétéroscédasticité**.

3. Vérification colinéarité des variables

On la réalise en étudiant **le facteur d’inflation de variance(VIF) pour chaque variable indépendante qui va permettre d'évaluer si les facteurs sont corrélés les uns aux autres.**

```
[1.1352454885217662, 1.2296897649621907, 1.4040824245306425, 1.5630988358433224]
```

➔Tous les coefficients < 10 : **il n'y a donc pas de problème de colinéarité.**

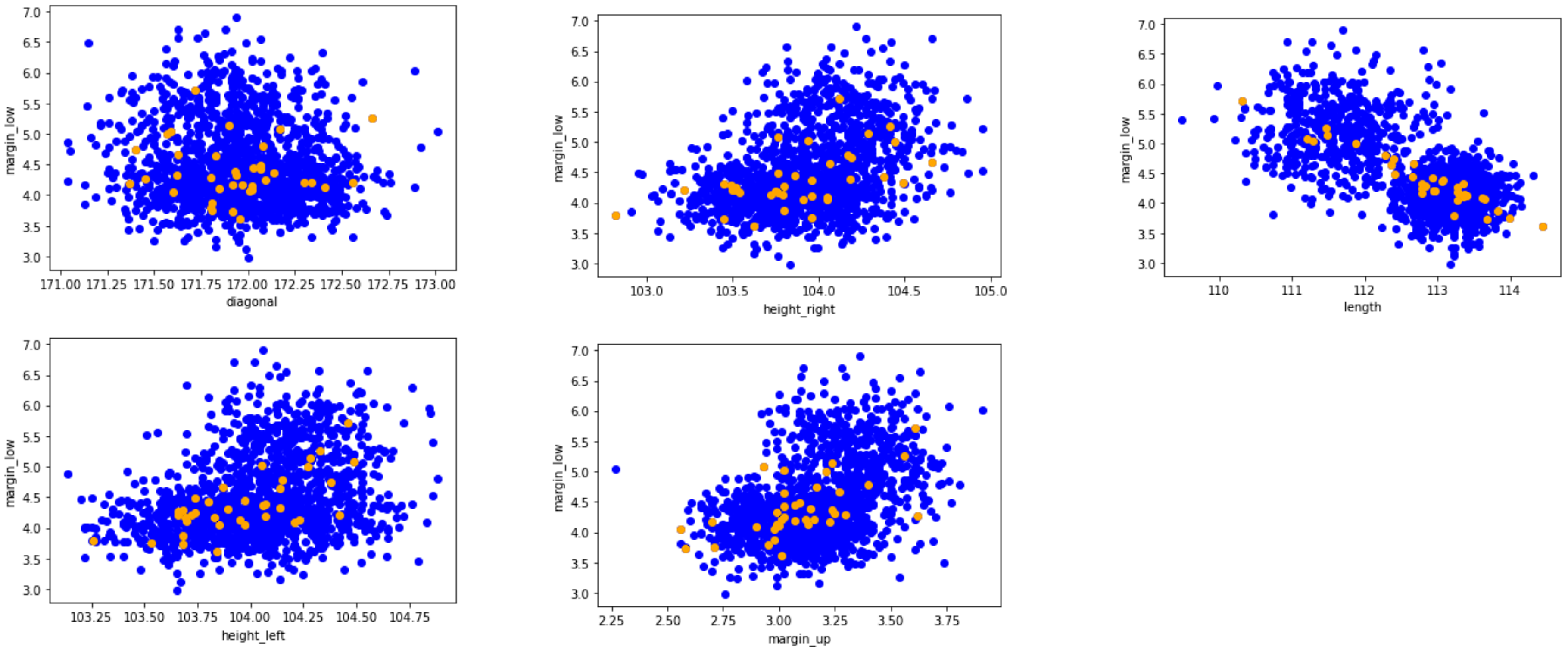
<u>CONCLUSION</u>

La normalité des résidus et l'égalité des variances <u>ne sont pas vérifiées</u> dans notre échantillon.

Cependant, l'observation des résidus, le fait qu'ils ne soient pas très différents d'une distribution symétrique, et le fait que l'échantillon soit de taille suffisante (supérieure à 30) permettent de dire que les résultats obtenus par le modèle linéaire gaussien ne sont pas absurdes, même si le résidu n'est pas considéré comme étant gaussien.

Nous pouvons donc valider l'utilisation du modèle de régression linéaire afin de déterminer les valeurs manquantes de notre jeu de données

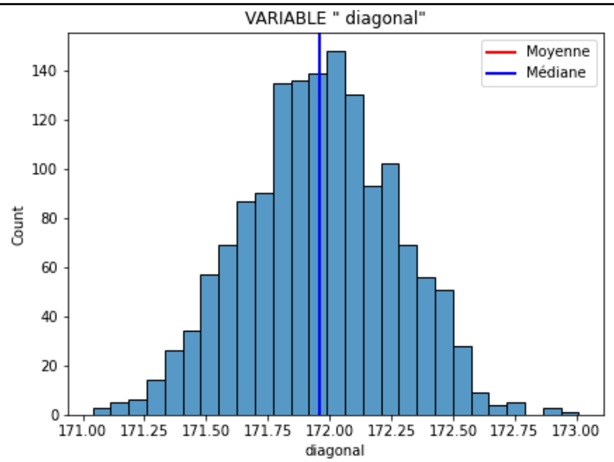
Visualisation par graphique nuage de points des données créées avec la régression linéaire (variable 'margin_low' VS les autres variables).

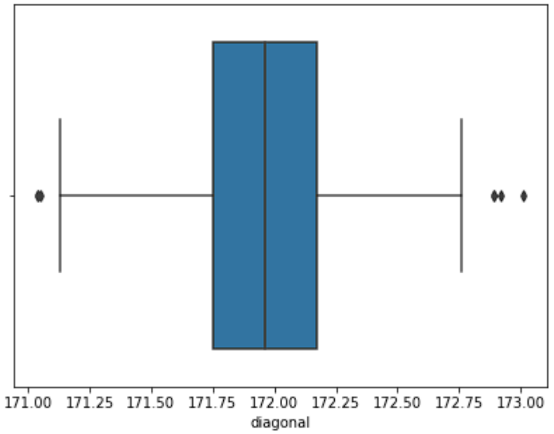


PARTIE 2

La mission

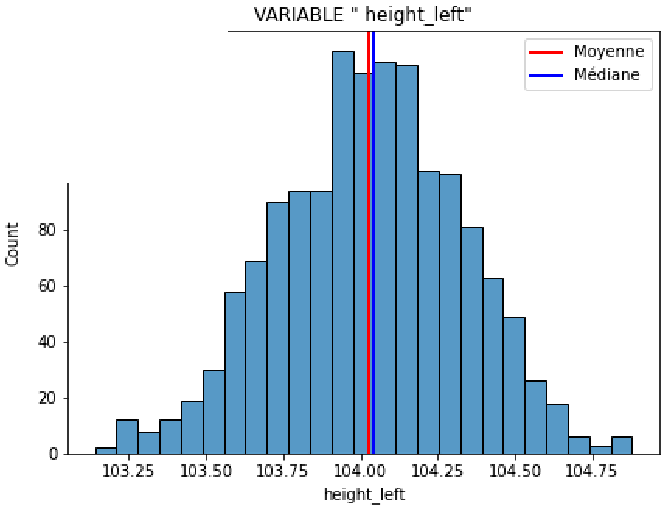
ANALYSES UNIVARIEES

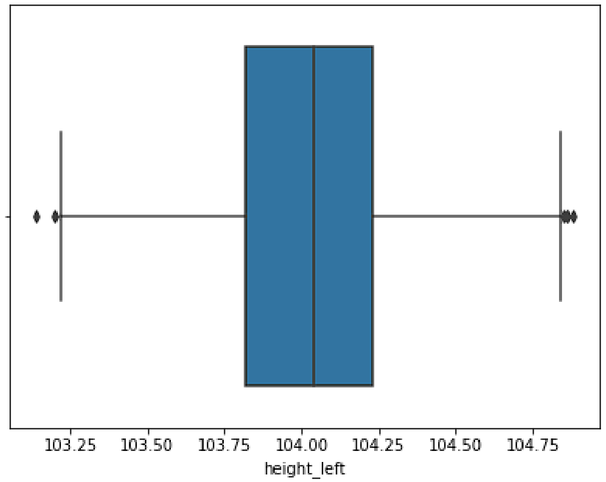




Variable ‘diagonal’

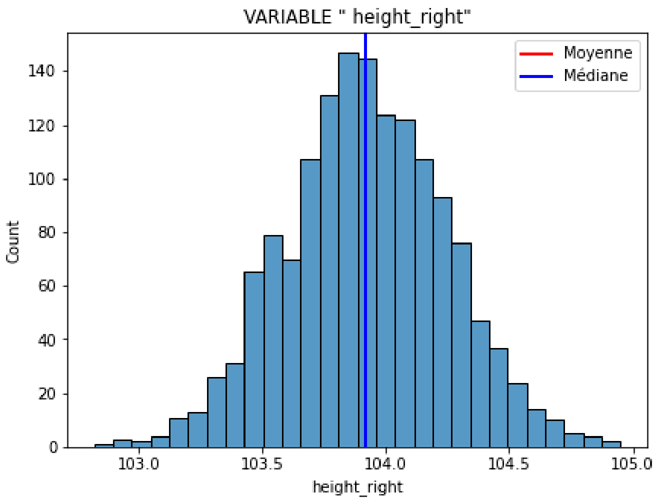
```
Moyenne :  171.96
Médiane :  171.96
Kurtosis :  -0.13
Écart-type :  -0.13
Test de Shapiro-Wilk / p-value :   0.32
avec H0 : La variable suit une loi normale si p-value > 5%
==> La variable ' diagonal 'suit une loi normale
```

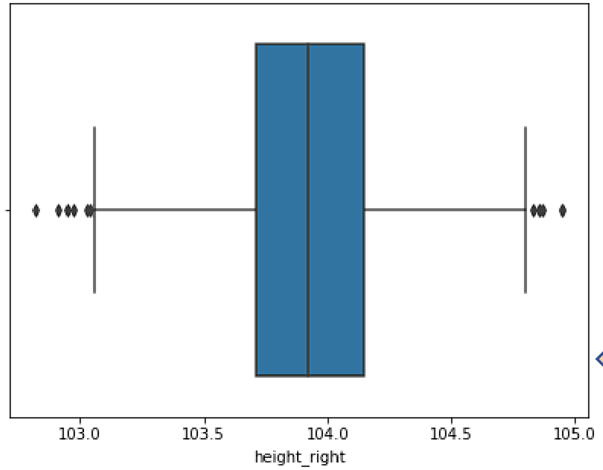




Variable ‘height_left’

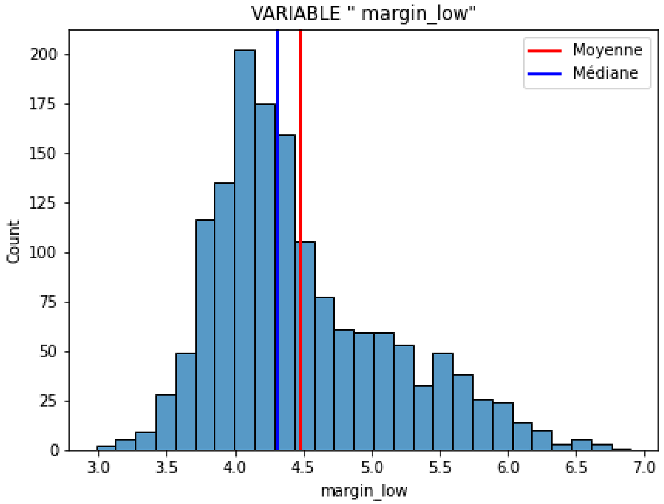
```
Moyenne :  104.03
Médiane :  104.04
Kurtosis :  -0.2
Écart-type :  -0.2
Test de Shapiro-Wilk / p-value :   0.05
avec H0 : La variable suit une loi normale si p-value > 5%
==> La variable ' height_left 'suit une loi normale
```

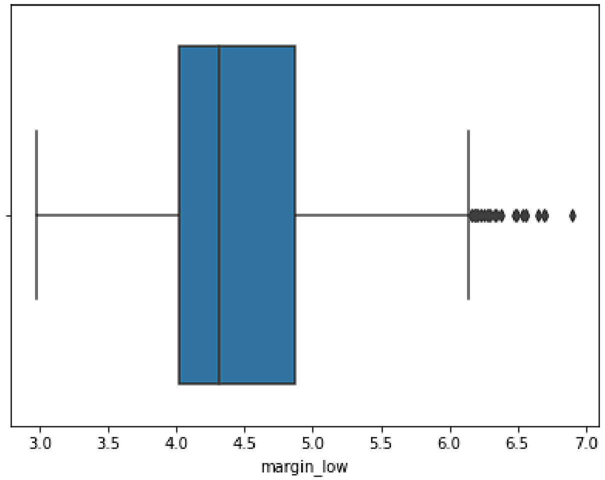




Variable ‘height_right’

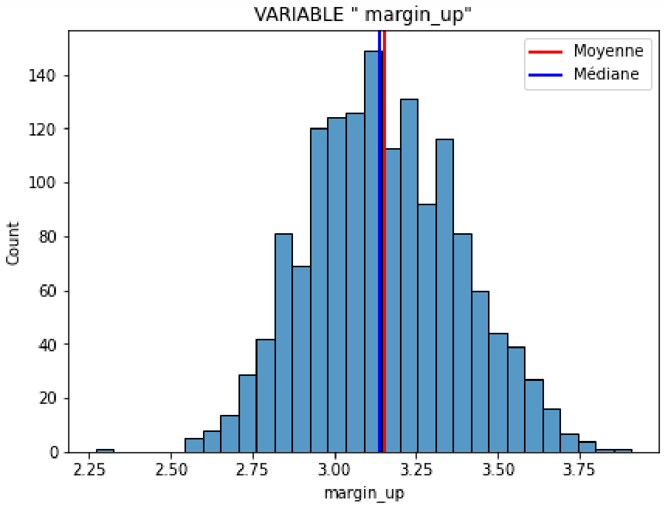
```
Moyenne :  103.92
Médiane :  103.92
Kurtosis :  -0.03
Écart-type :  -0.03
Test de Shapiro-Wilk / p-value :   0.98
avec H0 : La variable suit une loi normale si p-value > 5%
==> La variable ' height_right 'suit une loi normale
```

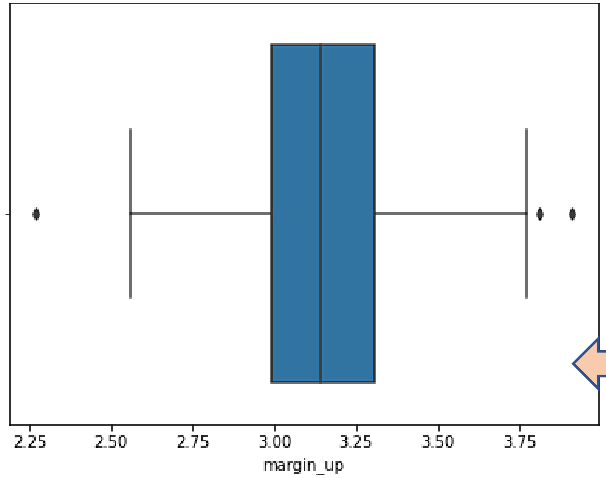




Variable ‘margin_low’

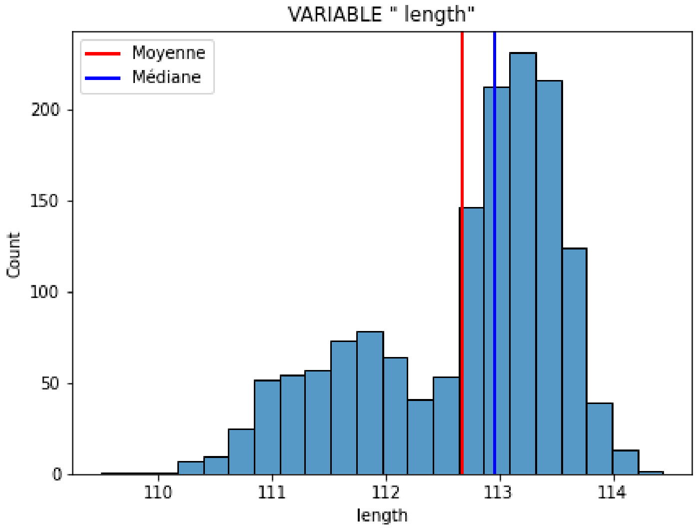
```
Moyenne :  4.48
Médiane :  4.31
Kurtosis :  0.25
Écart-type :  0.25
Test de Shapiro-Wilk / p-value :   0.0
avec H0 : La variable suit une loi normale si p-value > 5%
==> La variable  margin_low  ne suit pas une loi normale
```

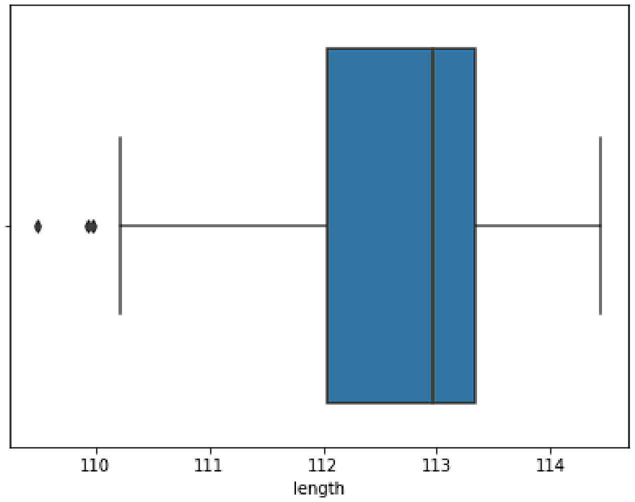




Variable 'margin_up'

```
Moyenne :  3.15
Médiane :  3.14
Kurtosis :  -0.25
Écart-type :  -0.25
Test de Shapiro-Wilk / p-value :   0.0
avec H0 : La variable suit une loi normale si p-value > 5%
==> La variable  margin_up  ne suit pas une loi normale
```





Variable 'length'

```
Moyenne :  112.68
Médiane :  112.96
Kurtosis :  -0.28
Écart-type :  -0.28
Test de Shapiro-Wilk / p-value :   0.0
avec H0 : La variable suit une loi normale si p-value > 5%
==> La variable  length  ne suit pas une loi normale
```

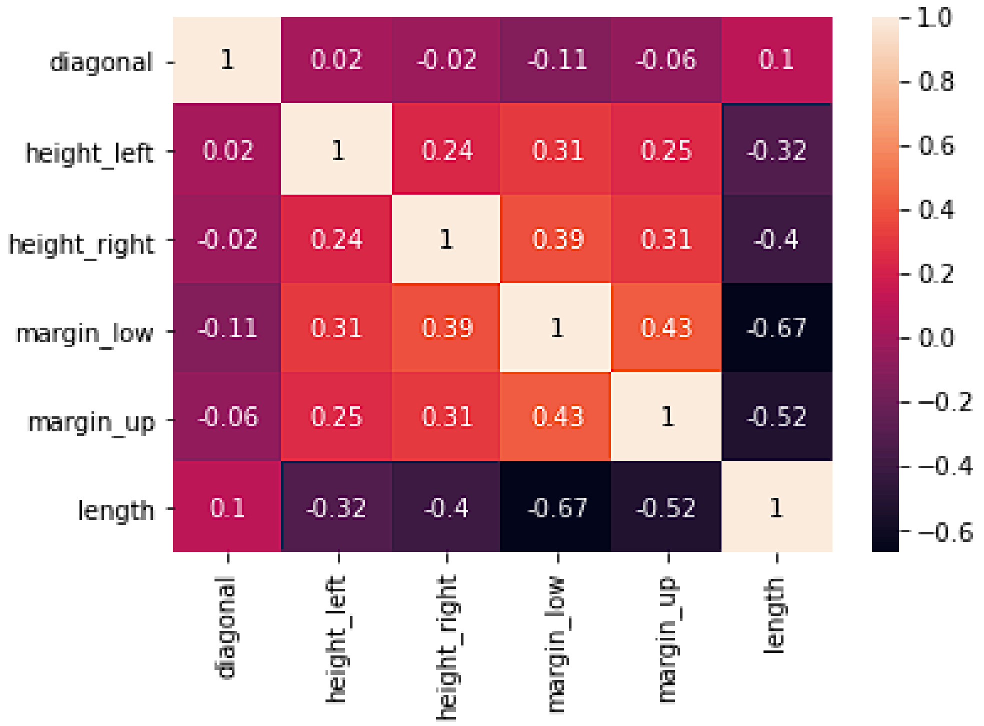
<u>EN SYNTHESE :</u>

- Distribution normale : "diagonal", "height_left", "height_right", "margin_up"
- Distribution qui ne suit pas une loi normale : "margin_low" (étalement vers la droite) et "length" (étalement vers la gauche)
- Présence de quelques outliers pour l'ensemble des variables mais pas de valeurs aberrantes : ces outliers sont conservés afin d'éviter le risque d'**underfitting** du modèle.

➔ Corrélation entre variables

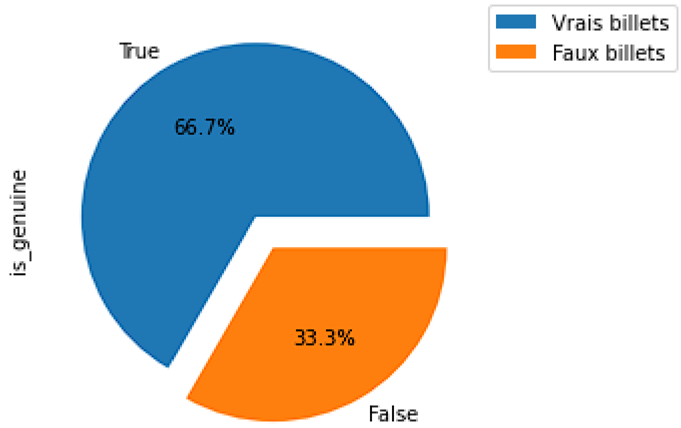
Corrélation négative significative entre marges (haute ou basse) et longueur

* **margin_low / length : -0.67**
*** margin_up / length : -0.52**



ANALYSES BIVARIEES

- **Analyse billets vrais et faux :** répartition des billets de l'échantillon en fonction des 2 modalités de la variable 'is_genuine' et portait 'type' des billets vrais et faux (utilisation de la moyenne des variables)

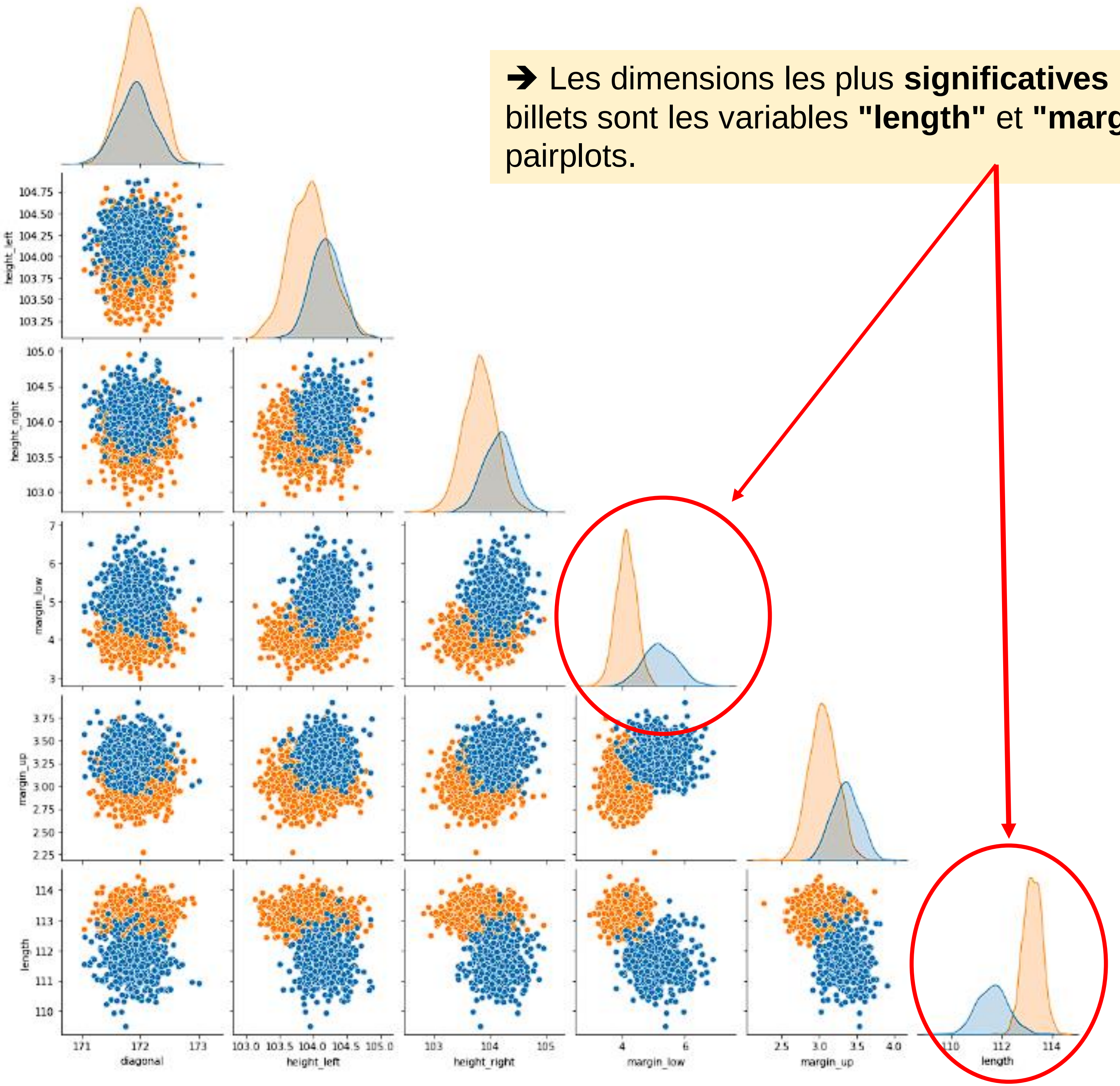


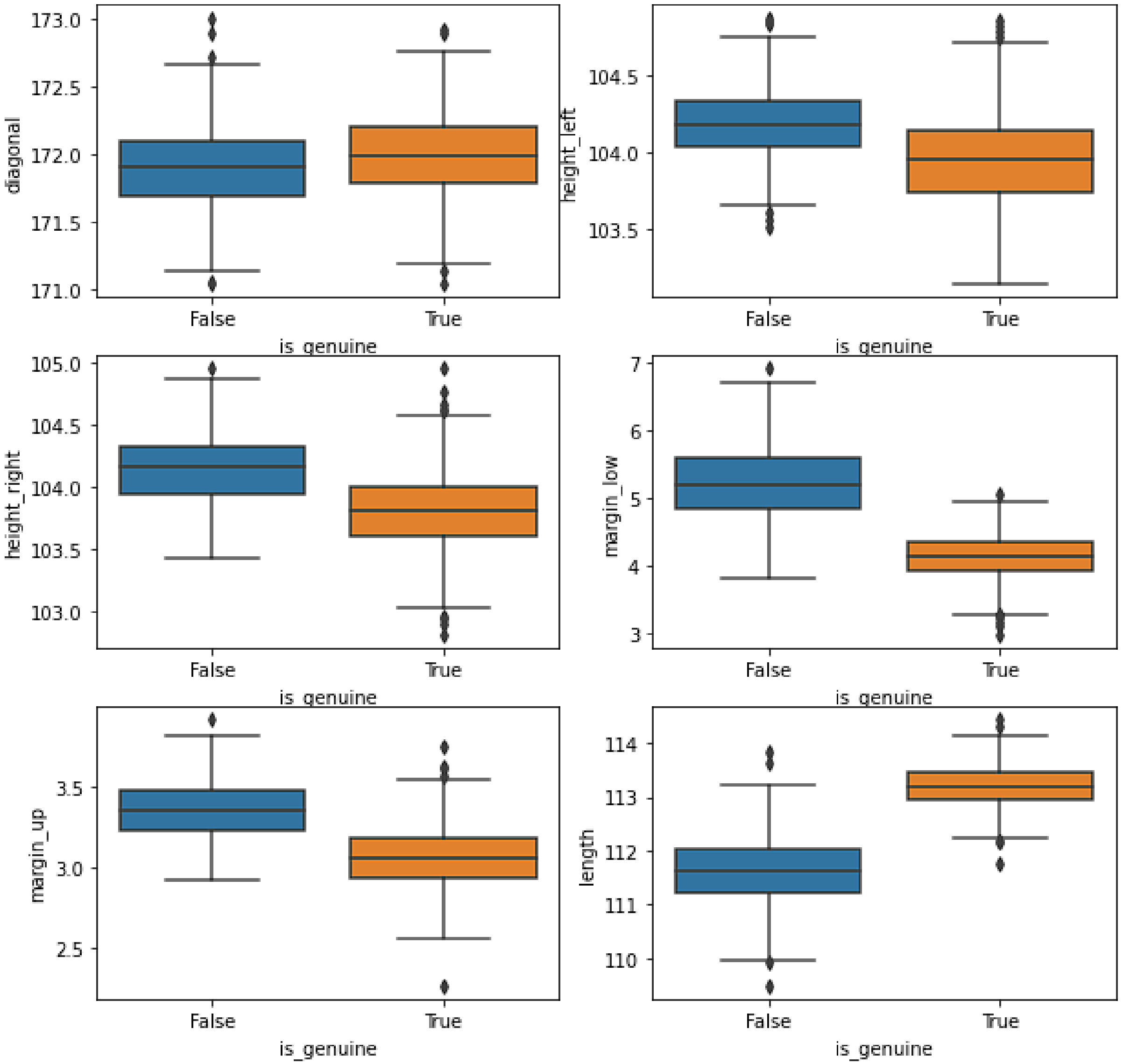
	Billet vrai ('is_genuine'=True)	Billet faux ('is_genuine'=False)
Length	113.20 mm	111.63 mm
height_left	103.94 mm	104.19 mm
height_right	103.80 mm	104.14 mm
margin_up	3.05 mm	3.35 mm
margin_low	4.11 mm	5.21 mm
diagonal	171.98 mm	171.90 mm

soit 1000 billets vrais et 500 billets faux

Un faux billet est un billet plus court et plus haut qu'un billet vrai.

➔ Les dimensions les plus **significatives** qui permettent de distinguer les vrais des faux billets sont les variables **"length"** et **"margin_low"** lorsqu'on observe les distributions via des pairplots.





➔ En revanche la distribution des données via des boxplots montrent des **différences significatives entre les différentes variables** suivant le type de billet à l'exception de la variable 'diagonal'.

En effet, les ordres de grandeur étant similaires entre les 2 types de billets, on remarque assez rapidement les différences même minimes entre les billets vrais et faux.

LES MODELISATIONS

➔ Réalisation de classifications automatiques pour le partitionnement des données.

➔ 2 types de classification ont été testés :

- Apprentissage non supervisé : algorithme de clustering **KMEANS**
- Apprentissage supervisé : **régression logistique**

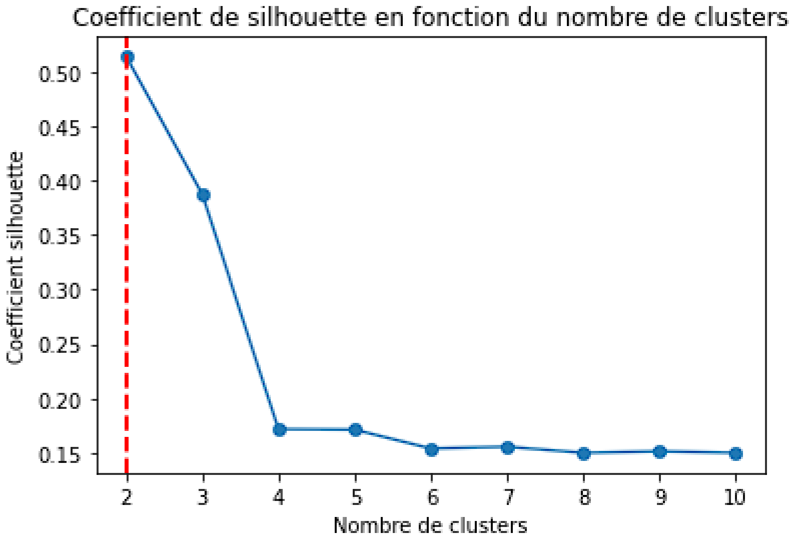
D'autres types de classification auraient pu être utilisés comme l'algorithme **KNN** (k plus proches voisins).

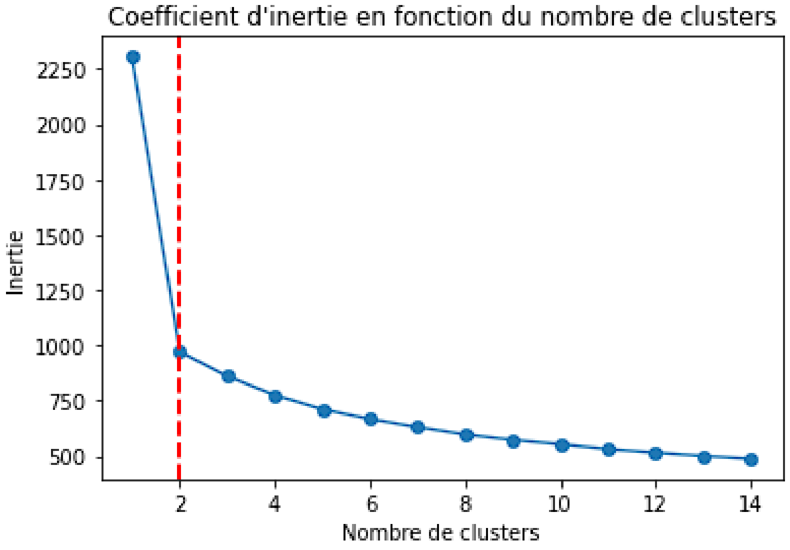
➔ **Pré-requis :** Standardisation des données afin de leur donner le même 'poids' (**centrage et réduction** des données avec la méthode « Standardscaler »). Les modèles ont été testés avec et sans standardisation. Les modèles les plus performants sont ceux avec standardisation des données.

Apprentissage non supervisé : algorithme de clustering : KMEANS

Calcul du nombre de clusters à choisir avec 2 procédés différents:

- calcul du coefficient de silhouette
- calcul du coefficient d'inertie.



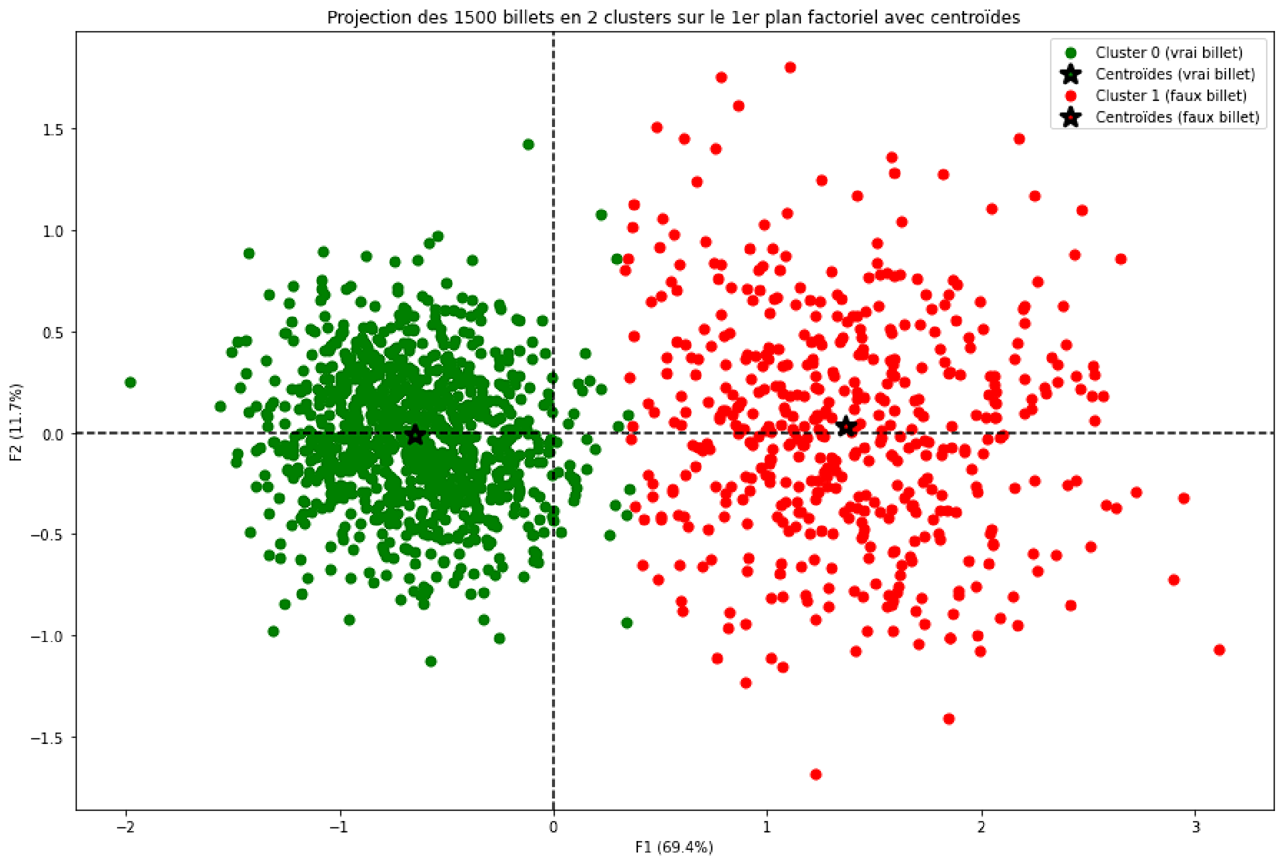


Résultat identique pour les 2 procédés : **répartition en 2 clusters**

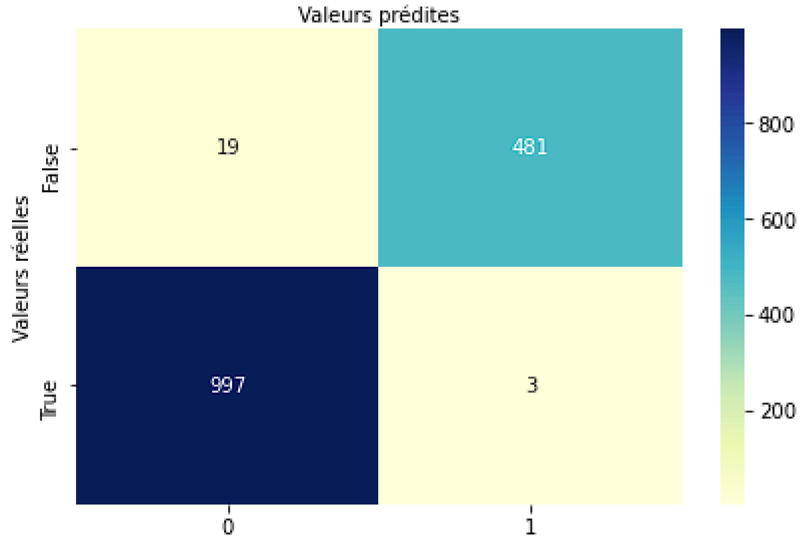
➔ SYNTHESE RESULTATS

```
Vrais billets (= cluster 0) : 997
Faux billets  (= cluster 1) : 481
```

Précision du modèle : 0.985



Matrice de confusion :



➔ Prédictions avec centroïdes des 2 clusters :

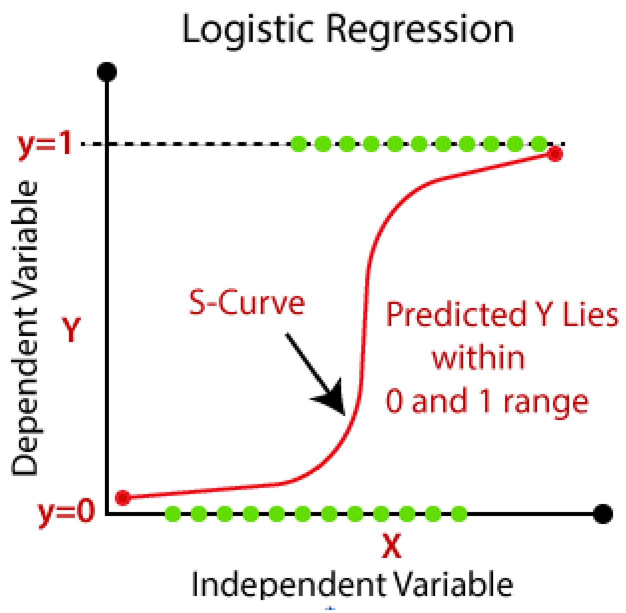
	diagonal	height_left	height_right	margin_up	margin_low	length	cluster_pred
0	171.986998	103.951654	103.813337	3.058219	4.124064	113.196152	0
1	171.898492	104.193017	104.144855	3.347231	5.237659	111.591860	1

Les clusters prédits pour les centroïdes correspondent bien à leur cluster de rattachement.

Apprentissage supervisé : REGRESSION LOGISTIQUE

- Réalisation d’une classification avec la variable catégorielle 'is_genuine' (variable expliquée qualitative=target) à partir des variables explicatives quantitatives (=features).

- Utilisation de la validation croisée stratifiée afin d'obtenir une évaluation plus robuste du modèle avec n_splits = 5

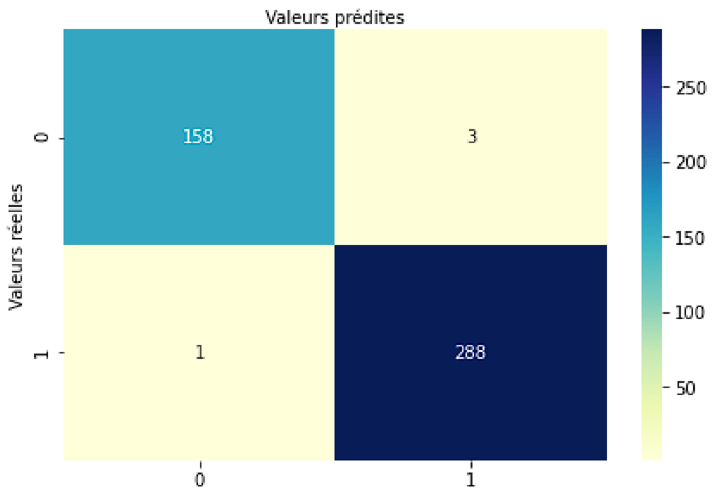


➔ SYNTHESE RESULTATS

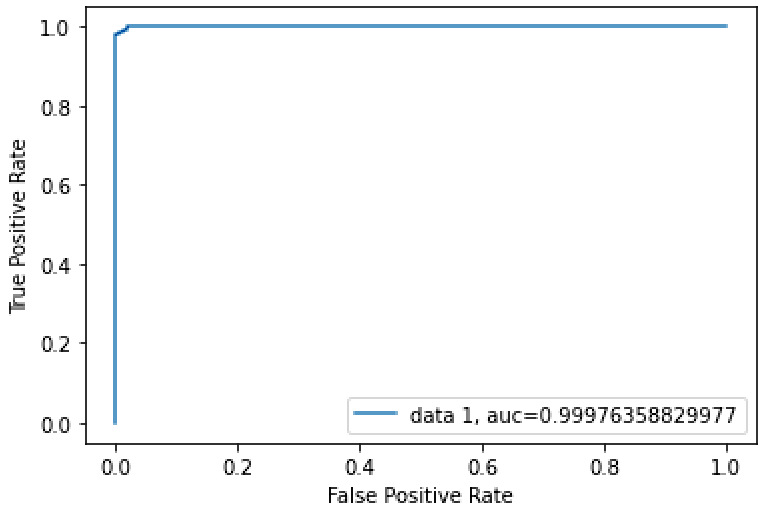
```
Validation croisée régression logistique avec standardisation des données :
---------------------------------------------------------------------------
[0.99666667 0.99 0.99 0.99666667 0.99 ]
```

Mean accuracy = 0.99266667

```
Matrice de confusion :
----------------------
```



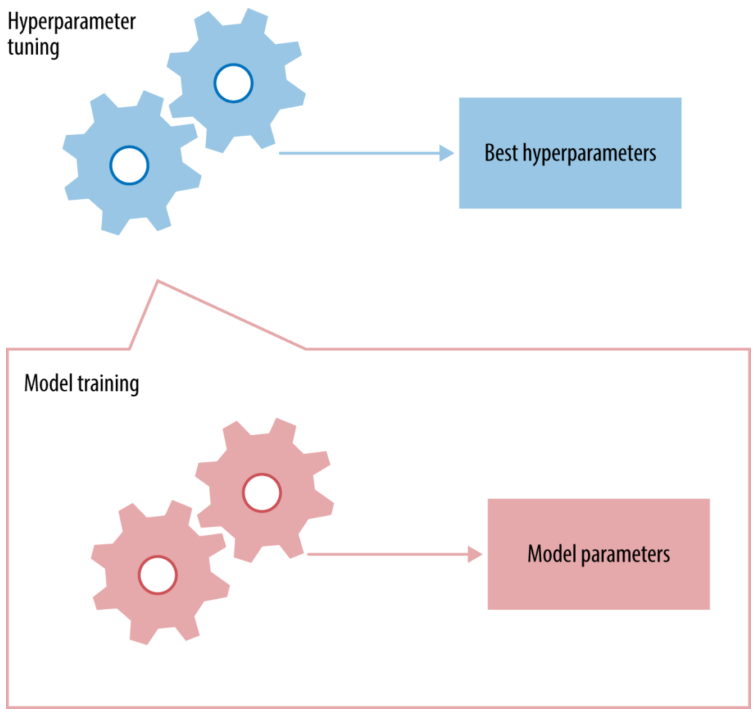

```
Courbe ROC :
------------
```



➔ La moyenne de précision du modèle est très bonne (99.26%) tout comme l'AUC qui est de 0.99
➔ Vérification si possibilité d'optimiser le modèle

OPTIMISATION DU MODELE DE REGRESSION LOGISTIQUE

Utilisation de la **validation croisée** et de la fonction **GridSearchCV** afin d'optimiser **les hyperparamètres** de la régression logistique.

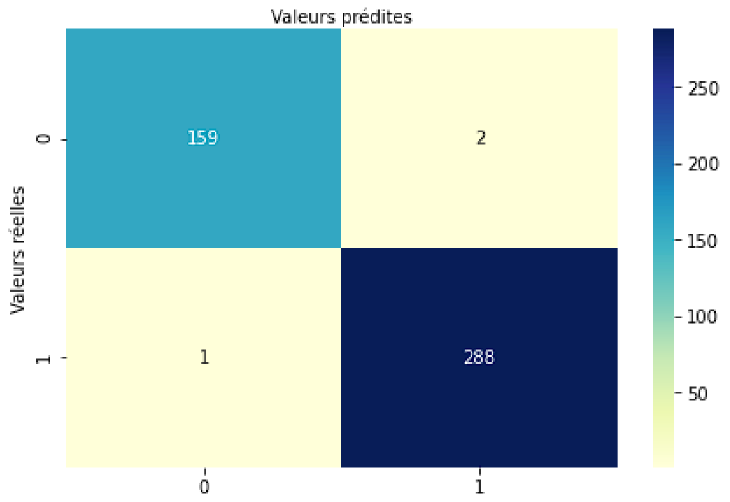



```
#Préparation de l'estimateur
model = Pipeline([('scaler', StandardScaler()), ('logistic', LogisticRegression())])

params = {
    'logistic__solver': ["lbfgs", "liblinear", "sag", "saga",'newton-cg',],
    'logistic__C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}
model_opti = GridSearchCV(model, param_grid=params, cv=cross_validation )
model_opti.fit(X_train, y_train)
```

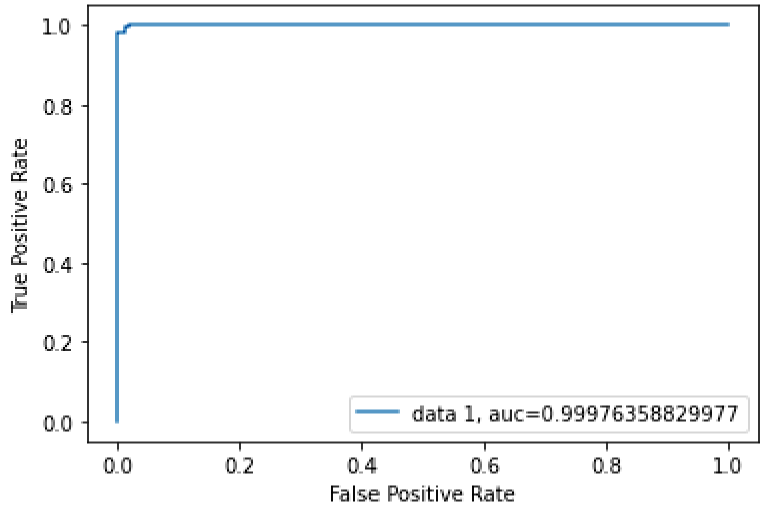
➔ SYNTHESE RESULTATS

```
Paramètres optimaux GridsearchCV :
-----------------------------------
Pipeline(steps=[('scaler', StandardScaler()), ('logistic',
LogisticRegression(C=0.1, solver='liblinear'))])

Test score : 99.33 %
```

```
Matrice de confusion :
----------------------
```




```
Courbe ROC :
--------------
```

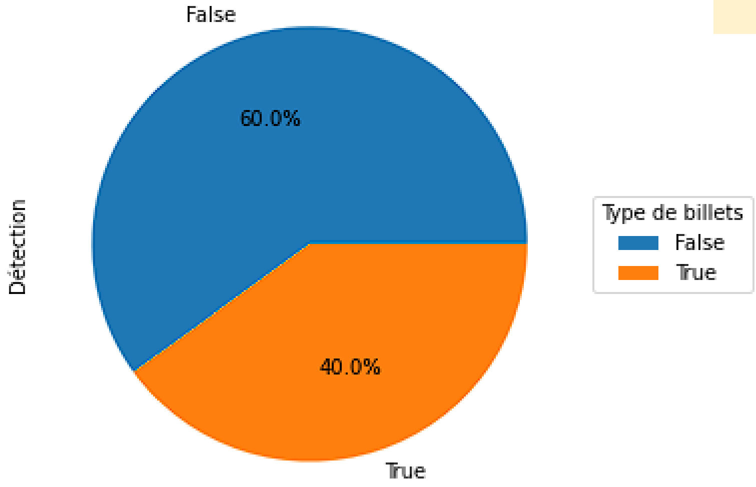


- La matrice de confusion montre une légère amélioration de la précision **(99.33%)**
- Pas de différence significative au niveau de la courbe ROC avec une AUC de **0.99**

<u>CONCLUSION :</u>
Avec 99,33% de précision, le modèle "model_opti" (librairie sklearn) est le modèle le plus robuste qui servira à l'élaboration du programme de détection de faux billets.

Test algorithme de détection de faux billets

```
SYNTHESE FICHIER
-----------------
```




```
5 billets dans le fichier dont :

- 2 billets vrais

- 3 billets faux

==> Liste id billets vrais :
---------------------------
['A_4', 'A_5']


==> Liste id billets faux :
---------------------------
['A_1', 'A_2', 'A_3']
```

➔ Tests réalisés avec le fichier **'billets_production.csv'** fourni, le modèle de données **'model_opti'** et un **seuil de 0.5**.

	id	Détection	Proba billet faux	Proba billet vrai
0	A_1	False	0.964837	0.035163
1	A_2	False	0.991812	0.008188
2	A_3	False	0.990659	0.009341
3	A_4	True	0.138143	0.861857
4	A_5	True	0.005344	0.994656