ONCFM

13/06/2024

Objectif du projet et grandes étapes

Objectif

Mettre en place un algorithme qui soit capable de différencier automatiquement les vrais des faux billets pour l'appliquer à un nouveau jeu de données

Sommaire

Les Etapes

- 1ère étape : Préparation des données
- 2ème étape : Analyse descriptive du jeu de données
- 3ème étape : Algorithme de classification Kmeans
- 4ème étape : Algorithme de classification de régression logistique

Etat des données

```
(1500, 7)
0
```

	dtype	missing_values	unique_values	count	min	max	moy	q1	med	q3
is_genuine	bool	0	2	1500	NaN	NaN	NaN	NaN	NaN	NaN
diagonal	float64	0	159	1500	171.04	173.01	171.96	171.750	171.96	172.17
height_left	float64	0	155	1500	103.14	104.88	104.03	103.820	104.04	104.23
height_right	float64	0	170	1500	102.82	104.95	103.92	103.710	103.92	104.15
margin_low	float64	37	285	1463	2.98	6.90	4.49	4.015	4.31	4.87
margin_up	float64	0	123	1500	2.27	3.91	3.15	2.990	3.14	3.31
length	float64	0	336	1500	109.49	114.44	112.68	112.030	112.96	113.34

	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length
0	True	171.81	104.86	104.95	4.52	2.89	112.83
1	True	171.46	103.36	103.66	3.77	2.99	113.09
2	True	172.69	104.48	103.50	4.40	2.94	113.16
3	True	171.36	103.91	103.94	3.62	3.01	113.51
4	True	171.73	104.28	103.46	4.04	3.48	112.54

Notre jeu de données comporte :

- 1500 lignes
- 7 variables dont 6 float et 1 bool
- 37 valeurs manquantes dans la variables ‘margin_low’

Etape 1 : Détection des outliers

Les outliers sont des valeurs ou observations « distantes » des autres observations effectuées. Elles contrastent avec les valeurs « normalement » mesurées.

Les algorithmes de ML sont sensibles aux données d'entrainement ainsi qu' à leurs distributions. Avoir des outliers dans ces données d'entrainement peut biaiser le modèle.

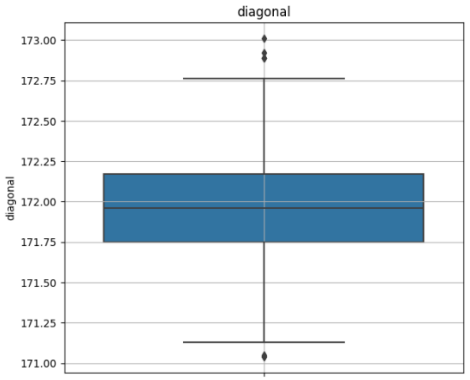
2 méthodes pour les calculer :

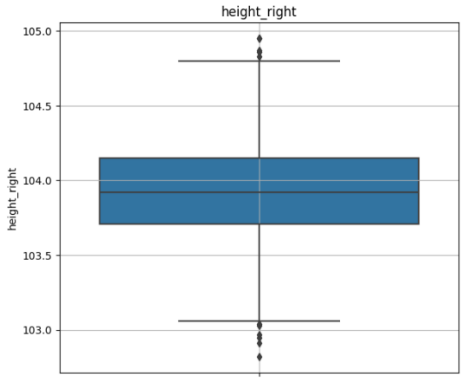
- Méthode Interquartile (IQR) : Une valeur est considérée comme aberrante si la valeur absolue de l'écart avec Q1 ou Q3 est supérieure à plus de 1,5 x Ecart interquartile.

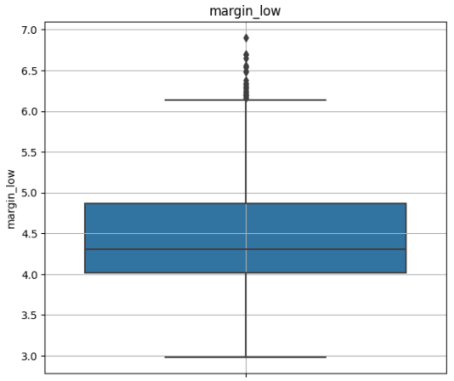
Avec cette méthode nous détectons 54 outliers dans notre jeu de données

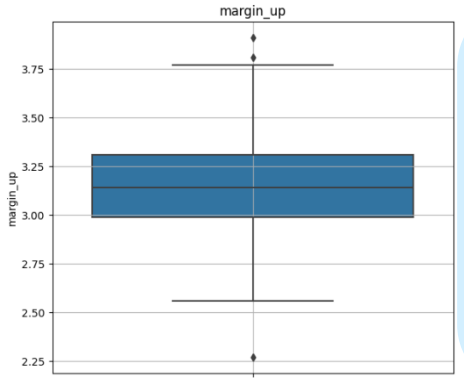
- Méthode z-score : Elle est basée sur l'écart type. Le score Z indique à combien d'écarts types une donnée se trouve par rapport à la moyenne.

Avec cette méthode nous détectons 24 outliers dans notre jeu de données









j'ai fait le choix de supprimer les 24 outliers de la méthode z-score.

C'est une décision modérée qui évite de supprimer trop de données.

Etape 1 : Remplacement des nan par Régression Linéaire

Notre jeu de données comporte 36 valeurs null dans la variable 'margin_low'.

Nous cherchons à « prédire » ces valeurs à l'aide d'une régression linéaire

Plusieurs étapes :

- Numérisation de la variable 'is_genuine'
- On sous échantillonne les lignes qui ont des nan pour entrainer notre modèle sans les nan. Les lignes nan serviront à la validation.
- Dans une variable 'y' on met notre variable à prédire et dans une variable 'X' nos variables prédictives
- On normalise nos données
- On entraine notre modèle en utilisant stat model
- On affine le modèle en ne gardant que les variables influentes (celles qui ont une p_value < 0,05) via une procédure backward
- Notre modèle de régression linéaire multiple prend en compte le type de billet et le margin_up.

La performance de notre modèle est plutôt bonne :

- L'erreur **RMSE** est de **0,40**
- Le score **R^2** est de **0,62**
- Le score **Mape** est de **0,07**

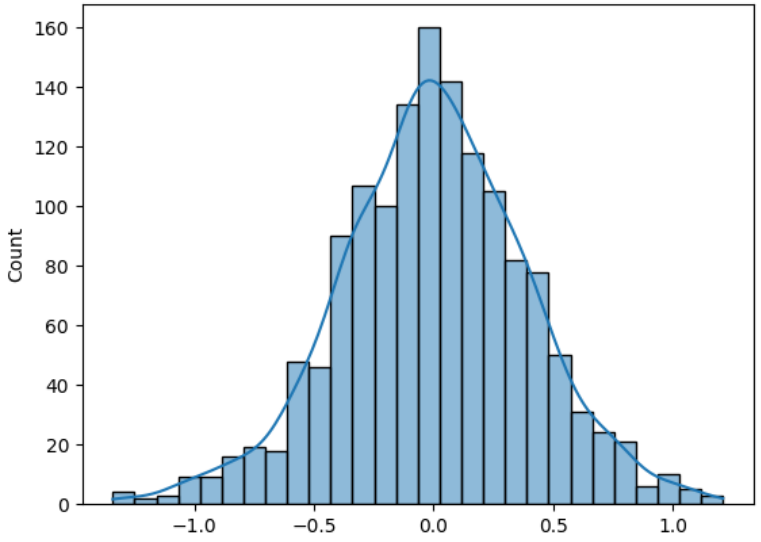
Etape 1 : Remplacement des nan par Régression Linéaire

Analyses de notre modèle

On s'assure de la bonne qualité de notre modèle en analysant les résidus. Les résidus sont les écarts entre les valeurs observées et les valeurs estimées

1) Homoscédasticité des résidus

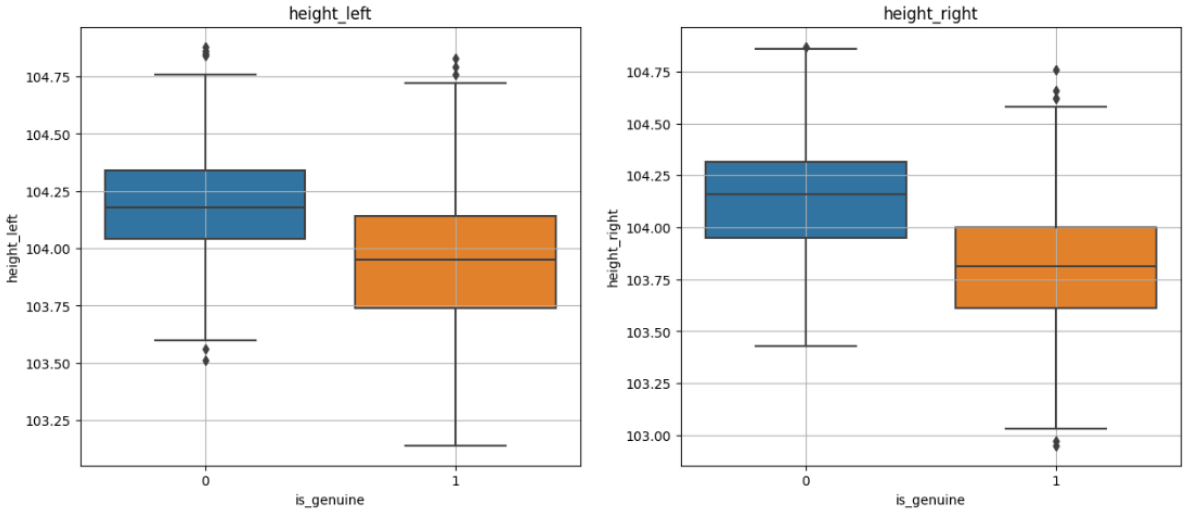
On teste la constance de la variance des résidus. Cette variance ne semble pas normale. Néanmoins, avec l'observation des résidus, on pense que la distribution est symétrique et la moyenne est proche de zéro.

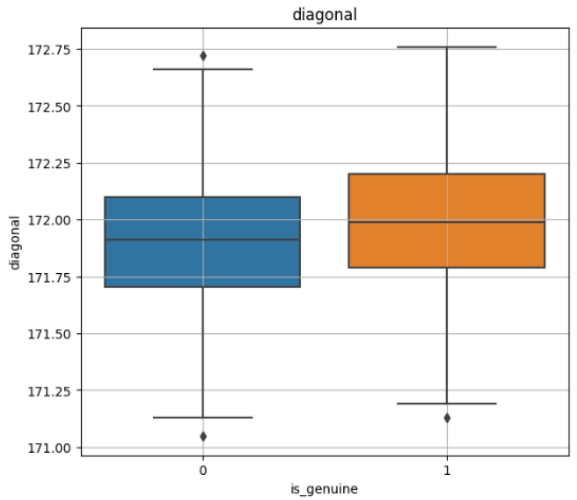


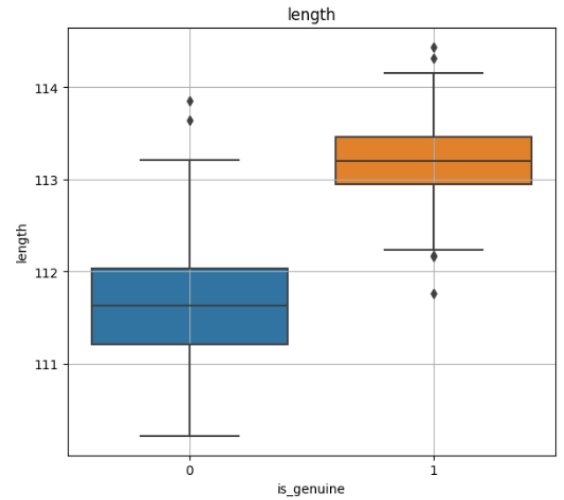
On considère que notre modèle de régression linéaire est correcte.
On prédit nos valeurs manquantes et on les remplace dans notre jeu de données

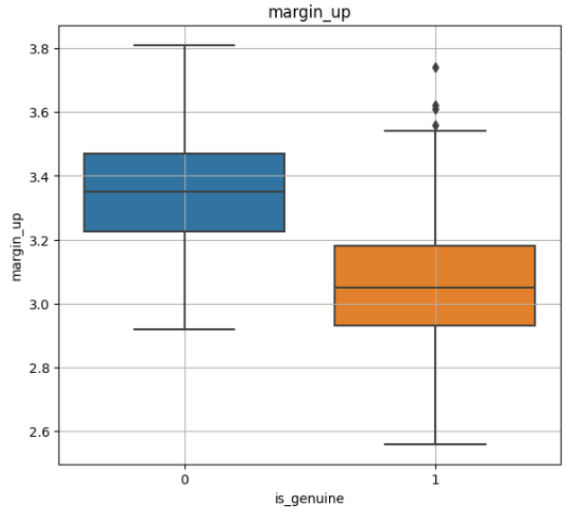
Etape 2 : Analyse descriptive du jeu de données

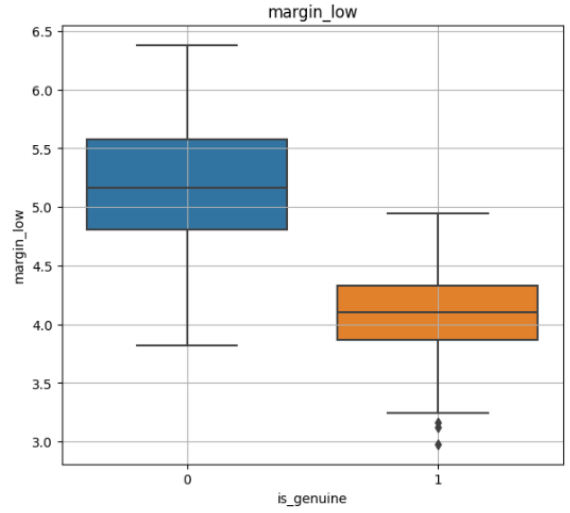
On représente la distribution de chaque variable en fonction de son type (vrai billet en orange, faux en bleu)









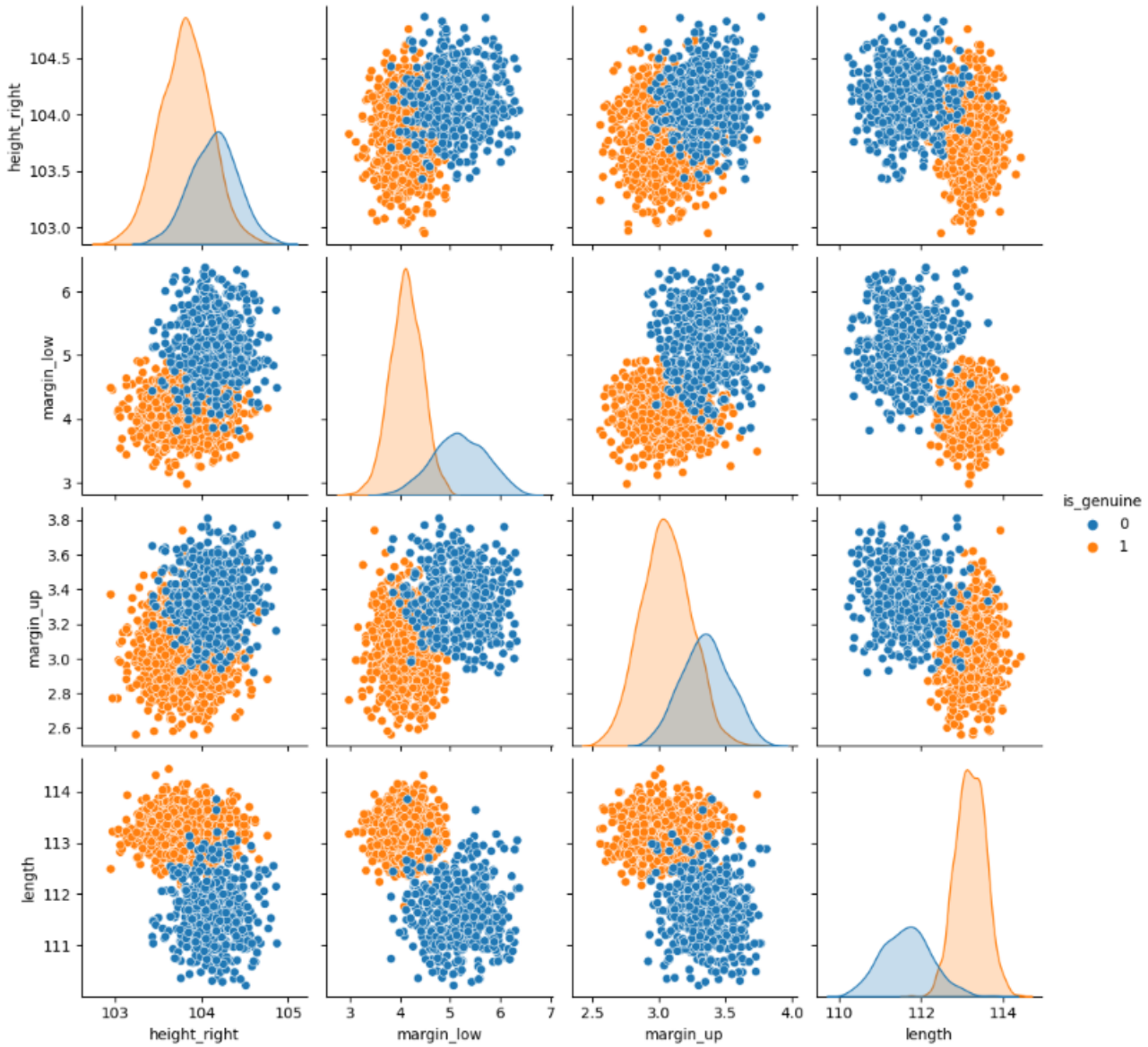


Visuellement on constate déjà des différences de distribution entre les faux et vrais billets sur des variables.

Les variables lenght, margin_low, margin_up et margin_right semble les plus concernées

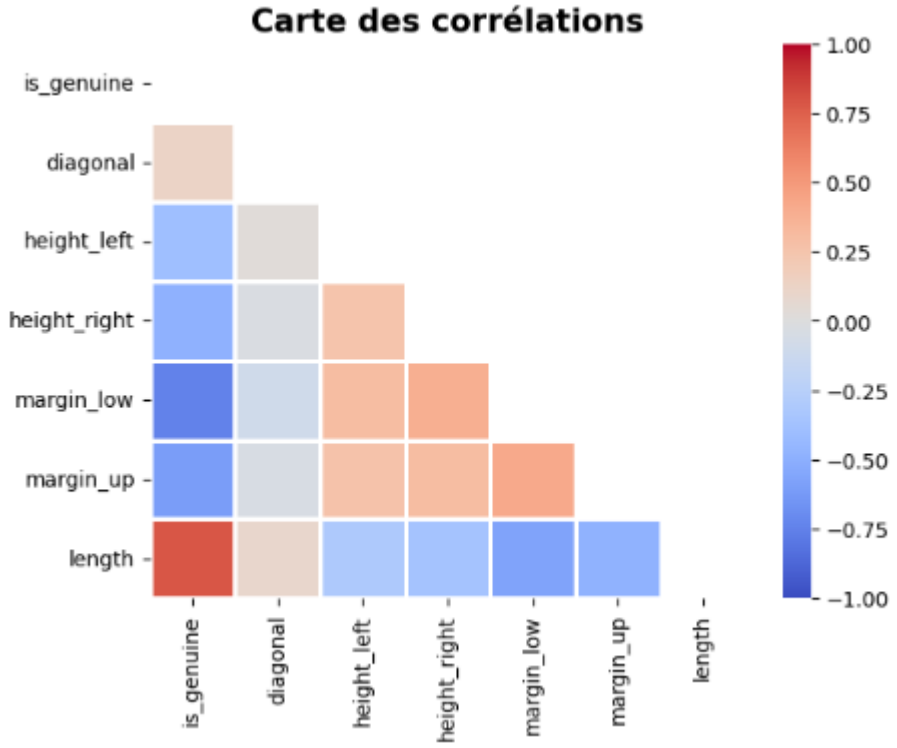
Etape 2 : Analyse descriptive du jeu de données

La séparation des groupes est encore plus prononcée quand on croise ces 4 variables



Etape 2 : Analyse descriptive du jeu de données

On peut constater cette corrélation via une **heatmap**



	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length
is_genuine	1.000000	0.124970	-0.396120	-0.488828	-0.742255	-0.609167	0.786153
diagonal	0.124970	1.000000	0.024052	-0.029485	-0.095292	-0.044873	0.088636
height_left	-0.396120	0.024052	1.000000	0.252056	0.303380	0.260672	-0.306357
height_right	-0.488828	-0.029485	0.252056	1.000000	0.384226	0.299289	-0.365103
margin_low	-0.742255	-0.095292	0.303380	0.384226	1.000000	0.415167	-0.572789
margin_up	-0.609167	-0.044873	0.260672	0.299289	0.415167	1.000000	-0.477305
length	0.786153	0.088636	-0.306357	-0.365103	-0.572789	-0.477305	1.000000

Plus le chiffre tend vers 1 (bleu foncé) plus la corrélation positive entre les 2 variables est forte.

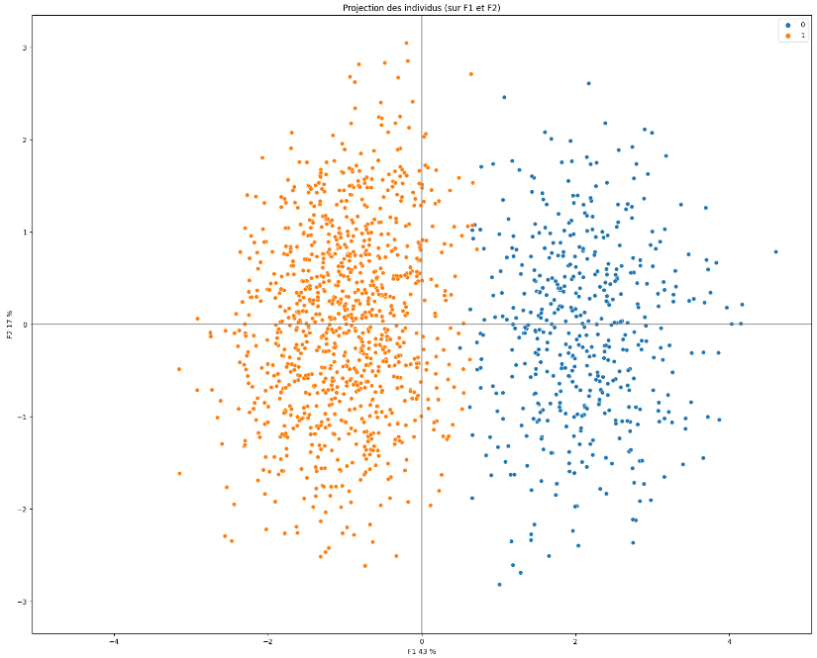
Inversement plus le chiffre tend vers -1 (rouge foncé) la corrélation est négative

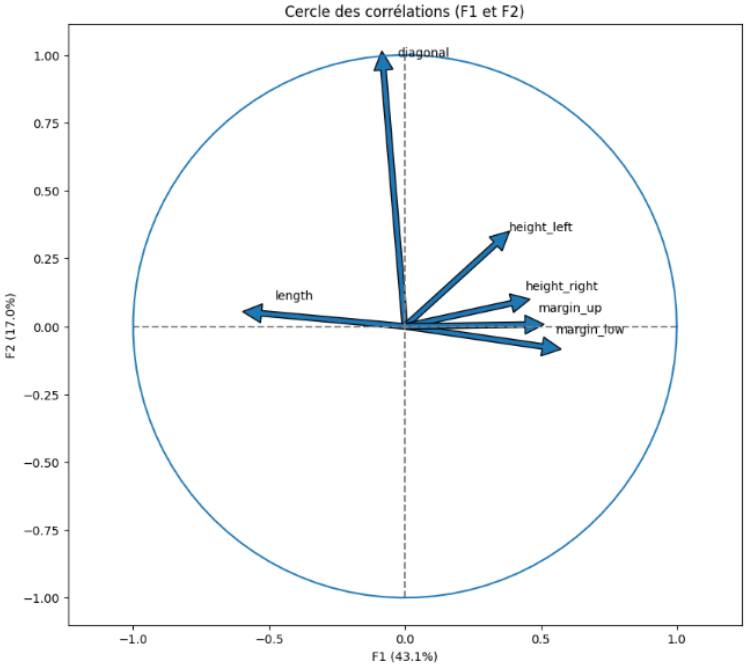
Le type de billet semble **corréler aux 4 variables** : lenght, margin_low, margin_up et margin_right

Etape 3 : Classification kmeans

ETAPES :

- On ne garde que les variables quantitatives
- Normalisation des données
- 2 clusters détectés via la 'elbow method'
- On entraine notre modèle
- On réalise une acp : le cercle des corrélations confirment notre impression de nos 4 variables impactantes sur le type de billet
- On projette nos points : les types de billets semblent bien repérés par la classification kmeans





Le rapport de classification

	precision	recall	f1-score	support
0	0.98	0.97	0.98	483
1	0.99	0.99	0.99	993
accuracy			0.99	1476
macro avg	0.99	0.98	0.98	1476
weighted avg	0.99	0.99	0.99	1476

Notre matrice de confusion

	pred_0	pred_1
test_0	470	13
test_1	8	985

Notre rapport de classification montre des scores très bon de notre modèle

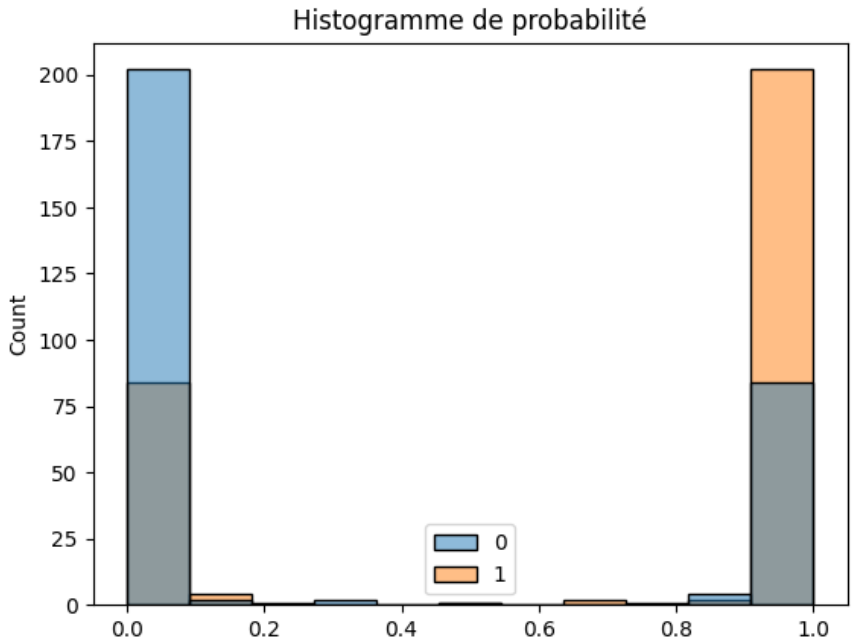
Notre matrice de confusion montre 470 vrai faux billets trouvés (sur 483) et 985 vrai billets (sur 993)

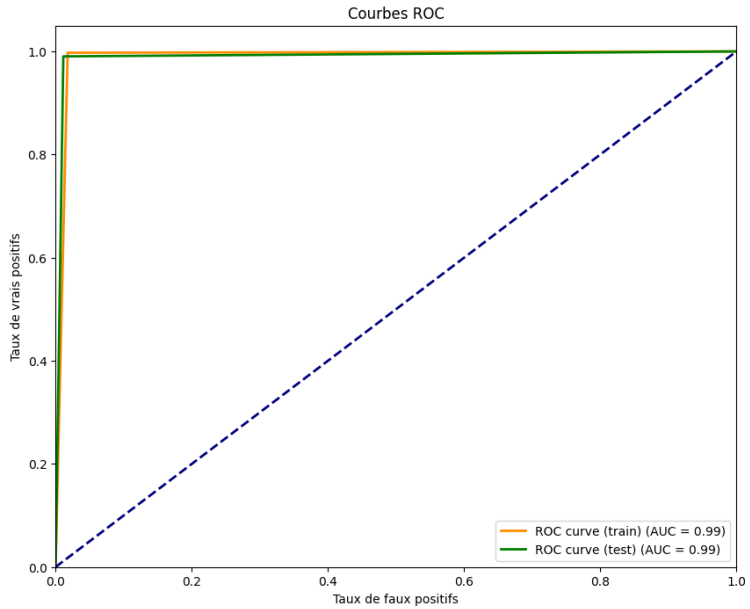
En revanche 13 faux billets ont été classés comme vrai et 8 vrai billets comme faux.

Etape 4 : Classification Régression logistique

ETAPES :

- On split nos données entre variables prédictives (X) et variable target (y). Puis on split en train & test
- Normalisation des données
- On entraine notre modèle avec scikit learn
- L'histogramme de probabilité nous montre que notre modèle semble précis à classer un billet en vrai ou faux
- On améliore notre modèle en choisissant de meilleurs paramètres





Le rapport de classification

	precision	recall	f1-score	support
0	0.99	0.99	0.99	88
1	1.00	1.00	1.00	208
accuracy			0.99	296
macro avg	0.99	0.99	0.99	296
weighted avg	0.99	0.99	0.99	296

Notre matrice de confusion

	pred_0	pred_1
test_0	87	1
test_1	1	207

Notre rapport de classification montre des scores très bon de notre modèle

Notre matrice de confusion est très bonne :

1 seul faux billet a été classé en vrai et 1 seul vrai billet a été classé en faux

Conclusion

Notre modèle de régression logistique est performant et meilleur que celui du kmeans.

Nous proposons celui-ci comme algorithme de détection de faux billets