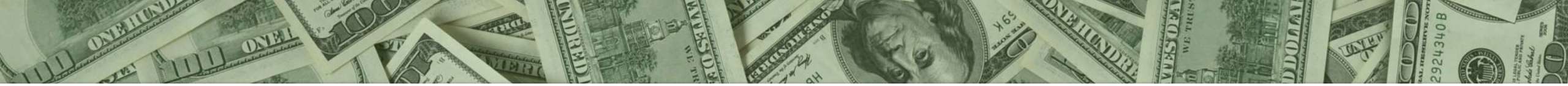
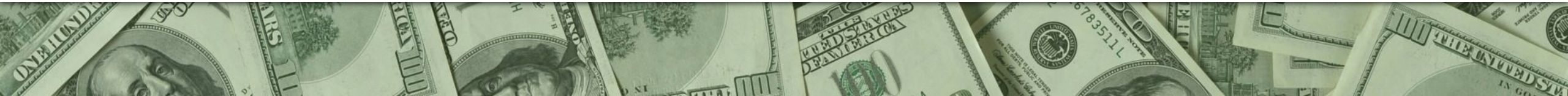


Objectifs

Construire un algorithme qui, à partir des caractéristiques géométriques d'un billet, serait capable de définir si ce dernier est un vrai ou un faux billet.

Sommaire

- Analyse des billets
- Régression linéaire
- K-means
- Knn
- Régression logistique
- Conclusion




Analyse

Analyse :

Présentation des données:

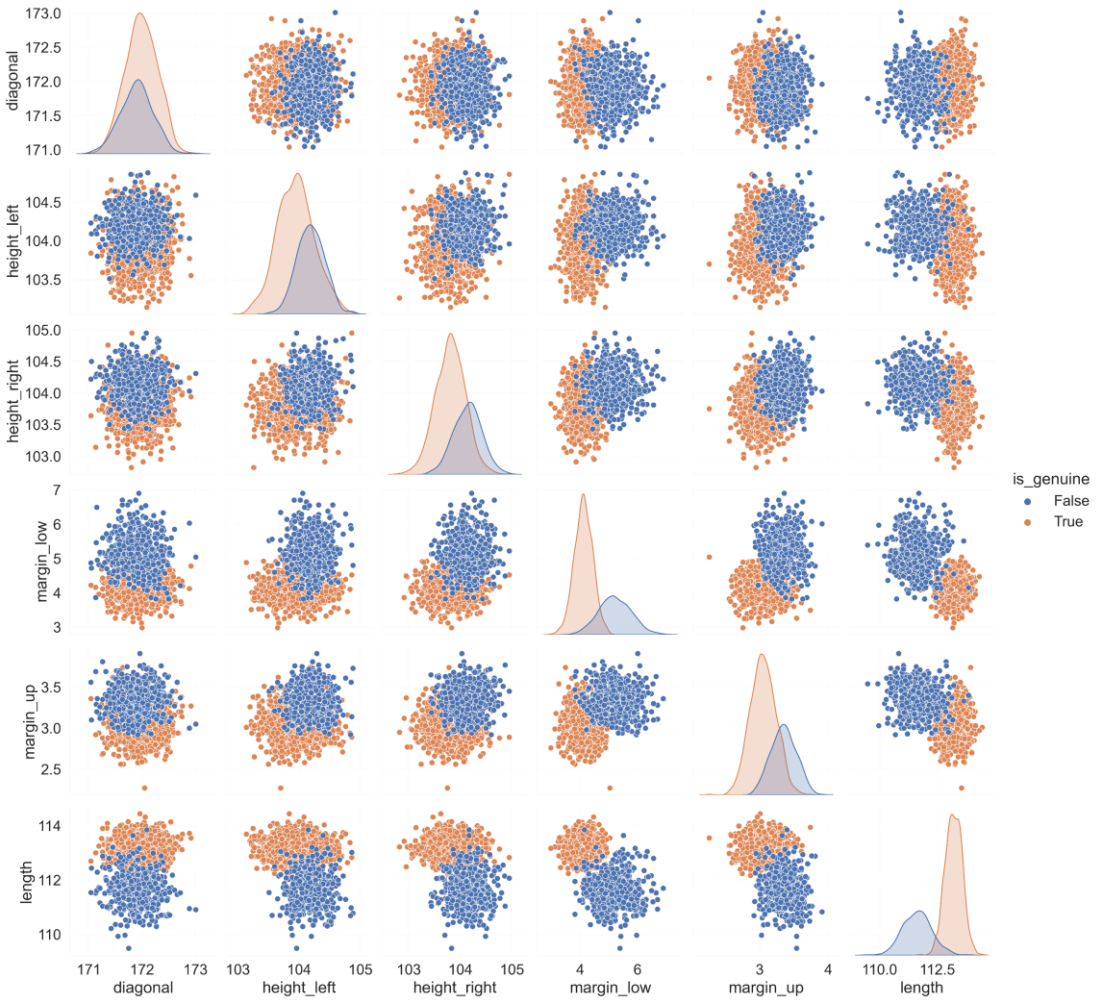
	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length
0	True	171.81	104.86	104.95	4.52	2.89	112.83
1	True	171.46	103.36	103.66	3.77	2.99	113.09
2	True	172.69	104.48	103.50	4.40	2.94	113.16
3	True	171.36	103.91	103.94	3.62	3.01	113.51
4	True	171.73	104.28	103.46	4.04	3.48	112.54

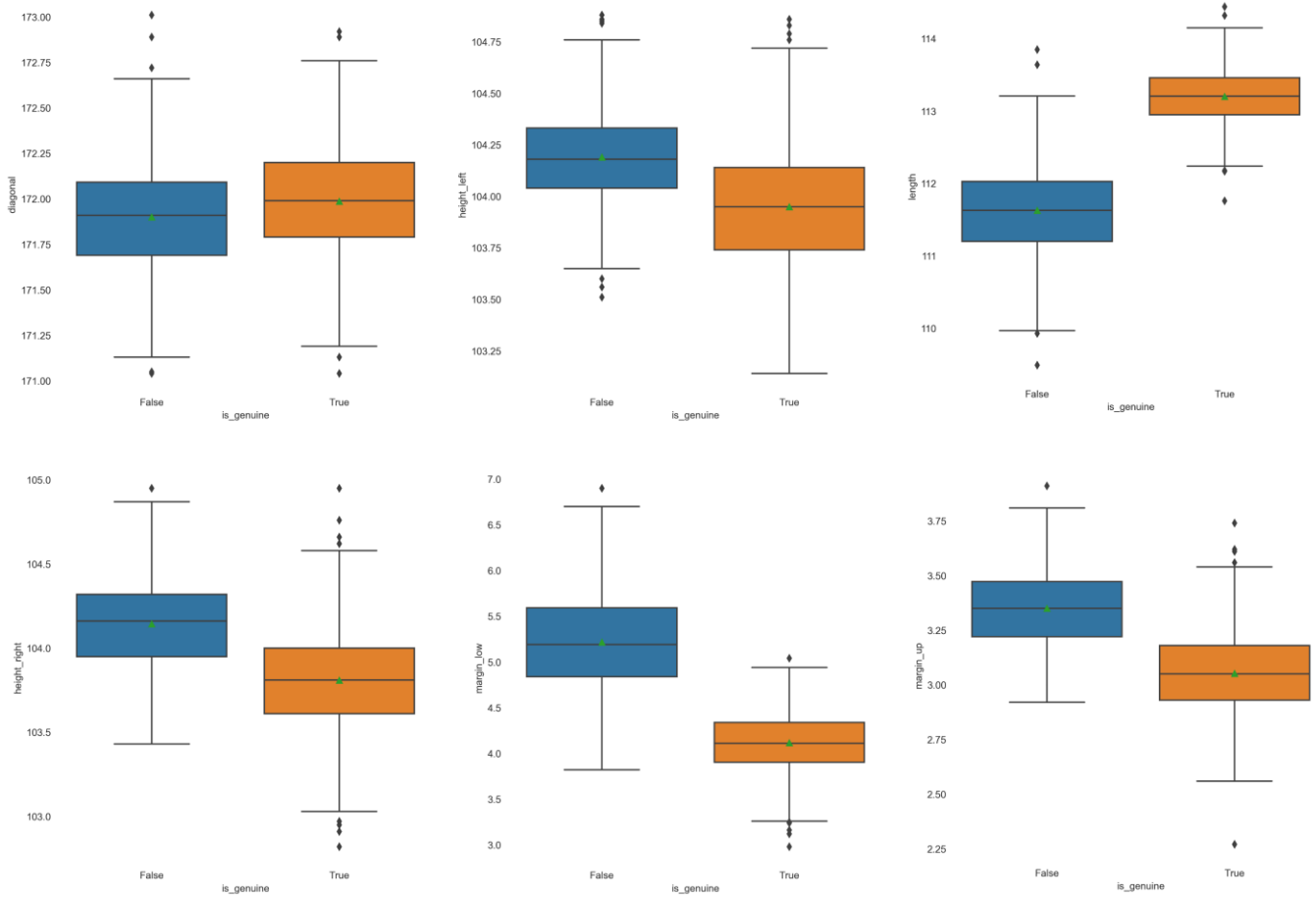
Valeurs manquantes :

Des valeurs manquantes dans la colonne margin_low

Vrai billet : 29 valeurs manquantes
Faux billet : 8 valeurs manquantes
Total billet : 37 valeurs manquantes

Analyse des billets :





Régression linéaire

Combler les valeurs manquantes :

Régression linéaire :

OLS Regression Results

Dep. Variable:	margin_low	R-squared:	0.617
Model:	OLS	Adj. R-squared:	0.615
Method:	Least Squares	F-statistic:	390.7
Date:	Sun, 27 Nov 2022	Prob (F-statistic):	4.75e-299
Time:	18:17:12	Log-Likelihood:	-774.14
No. Observations:	1463	AIC:	1562.
Df Residuals:	1456	BIC:	1599.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.8668	8.316	0.345	0.730	-13.445	19.179
is_genuine[T.True]	-1.1406	0.050	-23.028	0.000	-1.238	-1.043
diagonal	-0.0130	0.036	-0.364	0.716	-0.083	0.057
height_left	0.0283	0.039	0.727	0.468	-0.048	0.105
height_right	0.0267	0.038	0.701	0.484	-0.048	0.102
margin_up	-0.2128	0.059	-3.621	0.000	-0.328	-0.098
length	-0.0039	0.023	-0.166	0.868	-0.050	0.042

Omnibus:	21.975	Durbin-Watson:	2.038
Prob(Omnibus):	0.000	Jarque-Bera (JB):	37.993
Skew:	0.061	Prob(JB):	5.62e-09
Kurtosis:	3.780	Cond. No.	1.95e+05

Test de colinéarité :

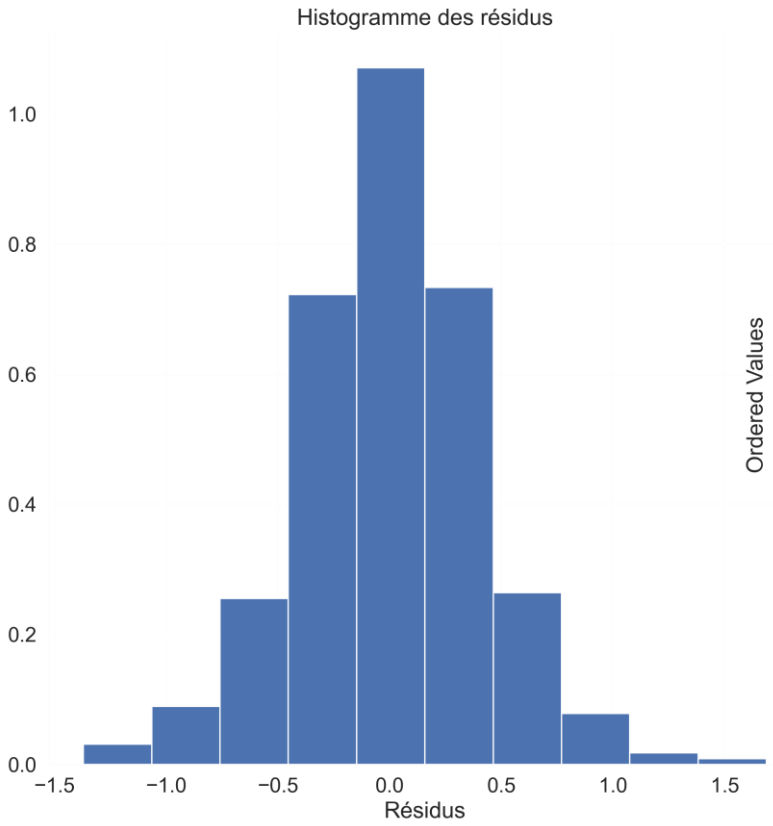
```
[1.5938854494007755, 1.5938854494007748]
```

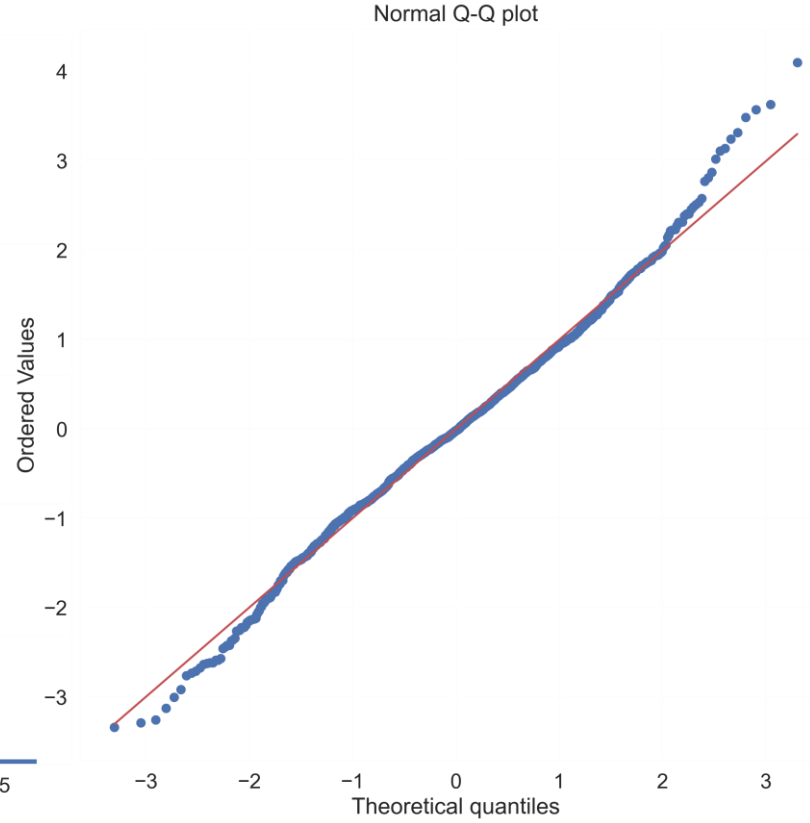
Test autocorrélation Durbin Watson :

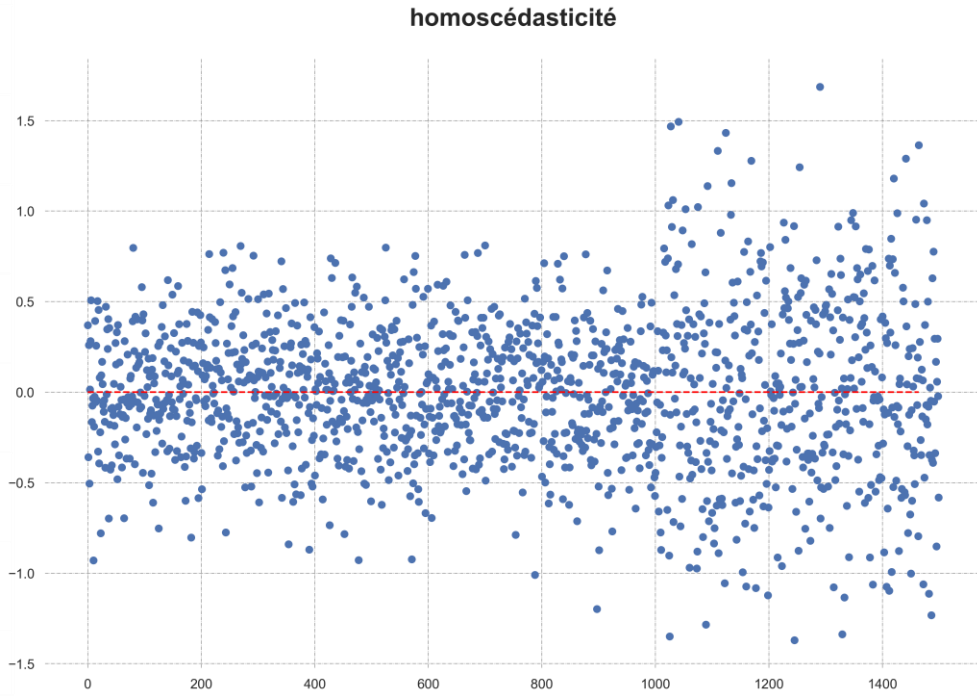
```
2.0410819121411503
```

R^2 : 0,617

MSE : 0,16317666353515042
Contenant tout mes nul : data_nul
Sans aucun nul : data_clean







Test Shapiro :

```
ShapiroResult(statistic=0.9936248064041138, pvalue=6.209427738212 98e-06)
```

Test d'homoscédasticité :

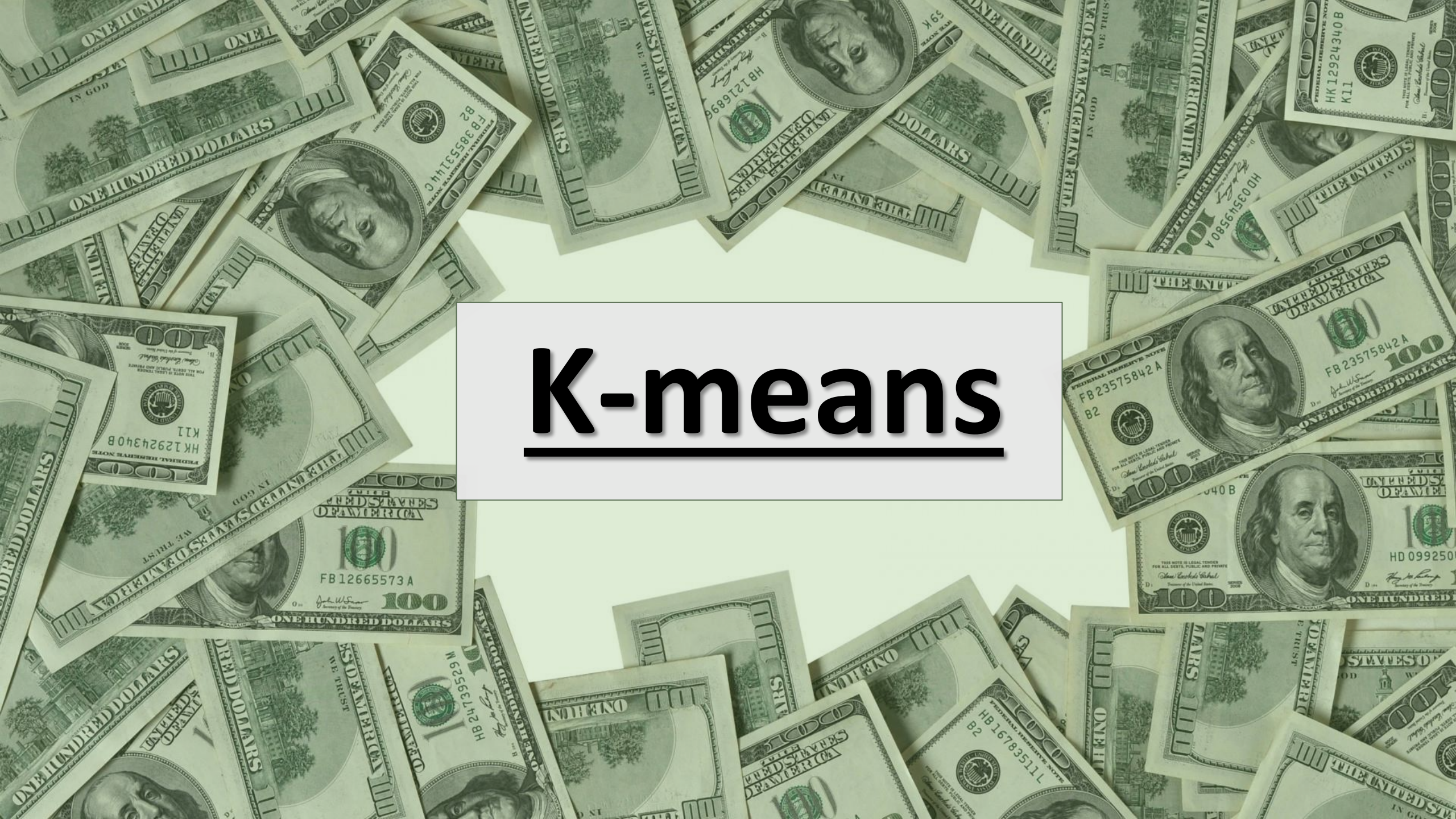
```
[('Bresuch-Pagan test', 163.45772873027045),
 ('p-value', 3.2033559115836335e-36),
 ('f-value', 91.82013129631463),
 ('f p-value', 2.7456283593639 73e-38)]
```

Dataframe Final :

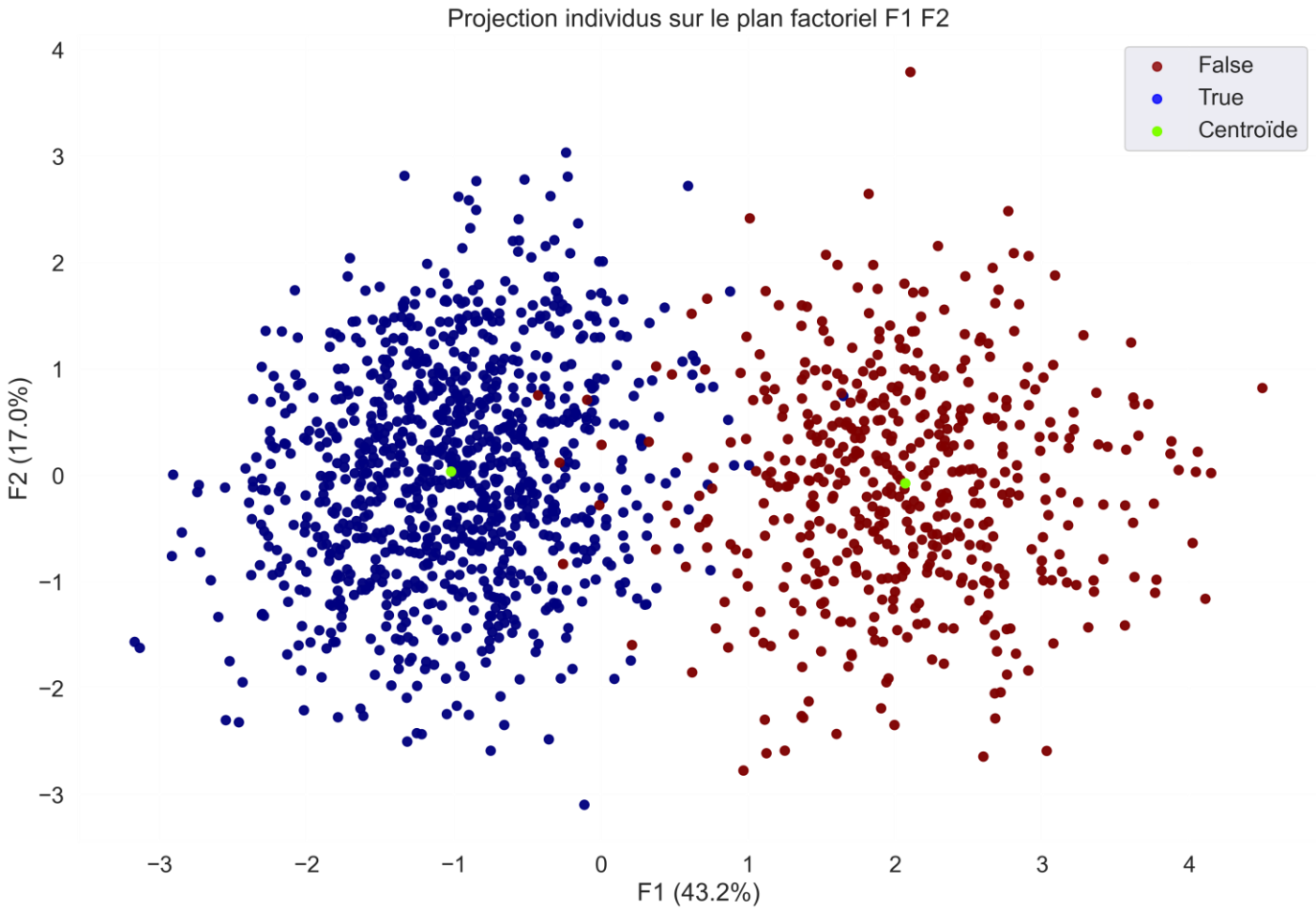
```
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   is_genuine    1500 non-null   bool   
 1   diagonal      1500 non-null   float64
 2   height_left   1500 non-null   float64
 3   height_right  1500 non-null   float64
 4   margin_up     1500 non-null   float64
 5   length        1500 non-null   float64
 6   margin_low    1500 non-null   float64
```

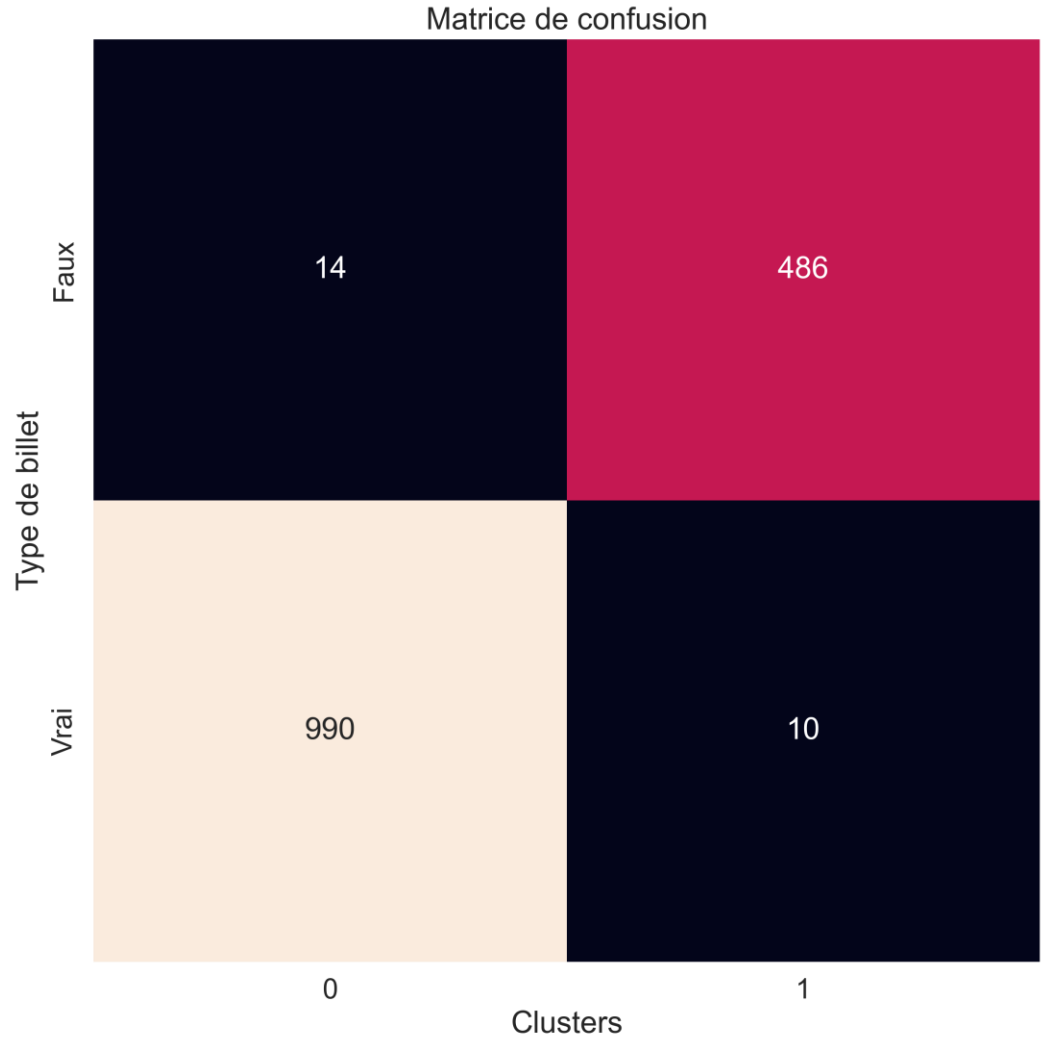
Données prédites :

```
nombre de données : 37
[4.049773   4.11192914 4.1373371  4.00530275 4.1147059  4.08365665
 4.07824963 4.13218571 4.07625982 4.06887804 4.09270537 4.20072639
 4.14831867 4.04177712 4.15764266 4.21536775 4.0999635  4.08385344
 4.08141511 4.10015599 4.13999692 4.16301741 4.16528771 4.11328527
 4.13919147 4.19682249 4.09476648 4.09908047 4.12776084 5.27901631
 5.2890952  5.27799711 5.28667512 5.23597327 5.15374018 5.19627059
 5.25639737]
```


K-means

K-means

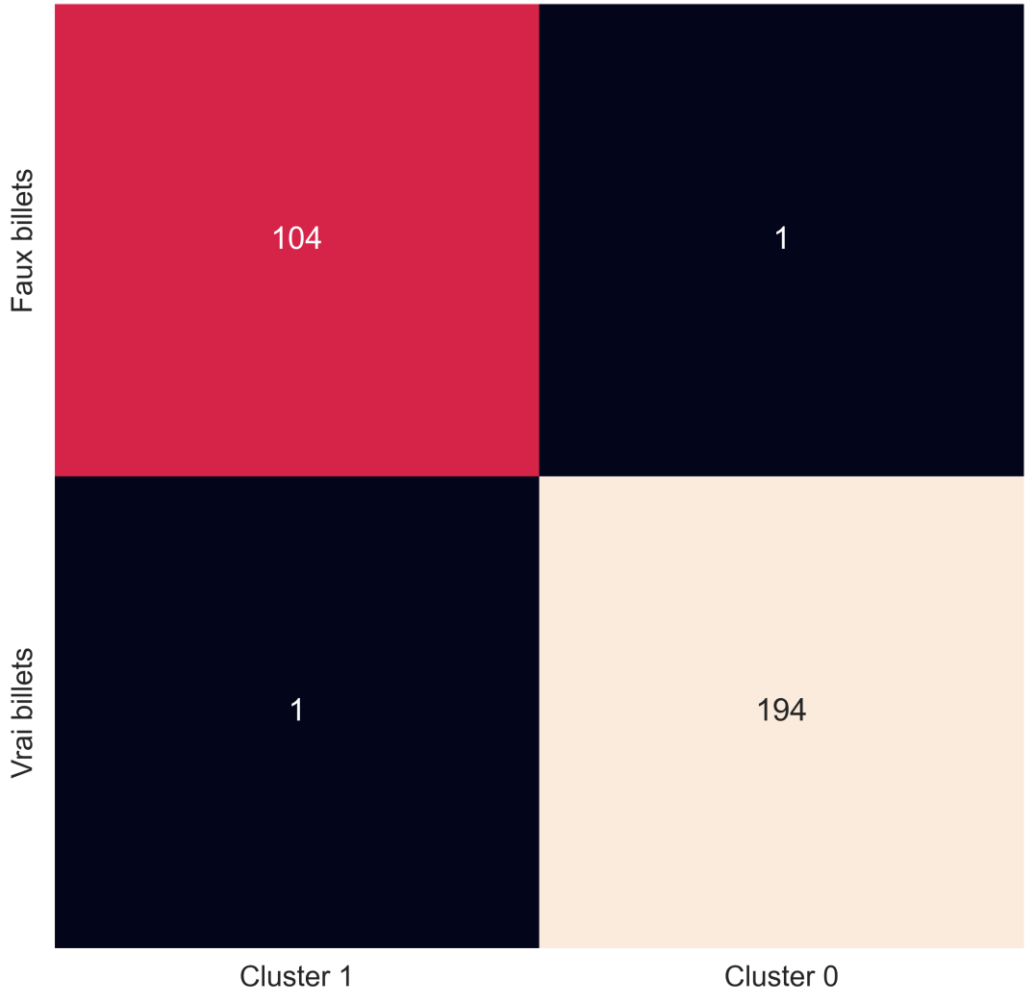

Projection individus sur le plan factoriel F1 F2
False
True
Centroïde
F2 (17.0%)
F1 (43.2%)
4
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
4


Matrice de confusion
Type de billet
Faux
Vrai
14
486
990
10
0
1
Clusters


K-nn

Knn:

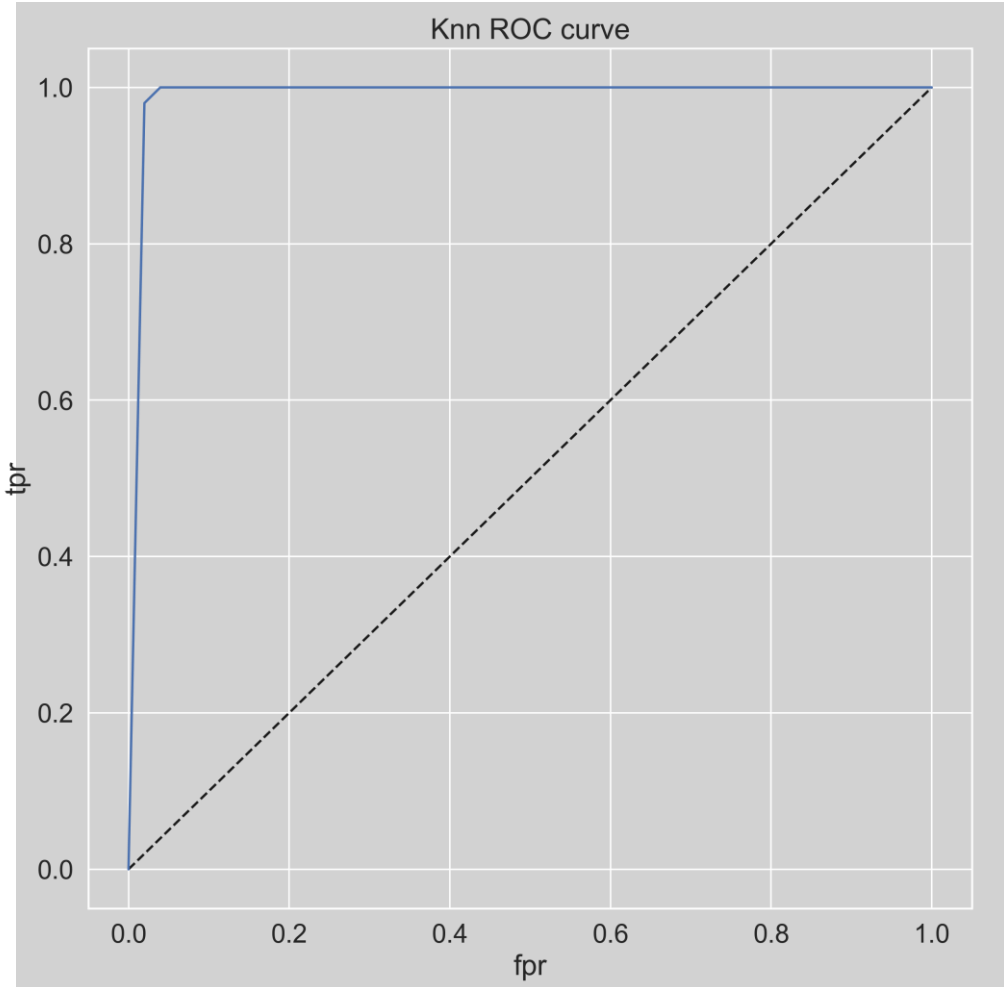
Données prédite :



99% de réussite pour le modèle knn

	precision	recall	f1-score	support
0	0.99	0.99	0.99	105
1	0.99	0.99	0.99	195
accuracy			0.99	300
macro avg	0.99	0.99	0.99	300
weighted avg	0.99	0.99	0.99	300

K-nn roc:



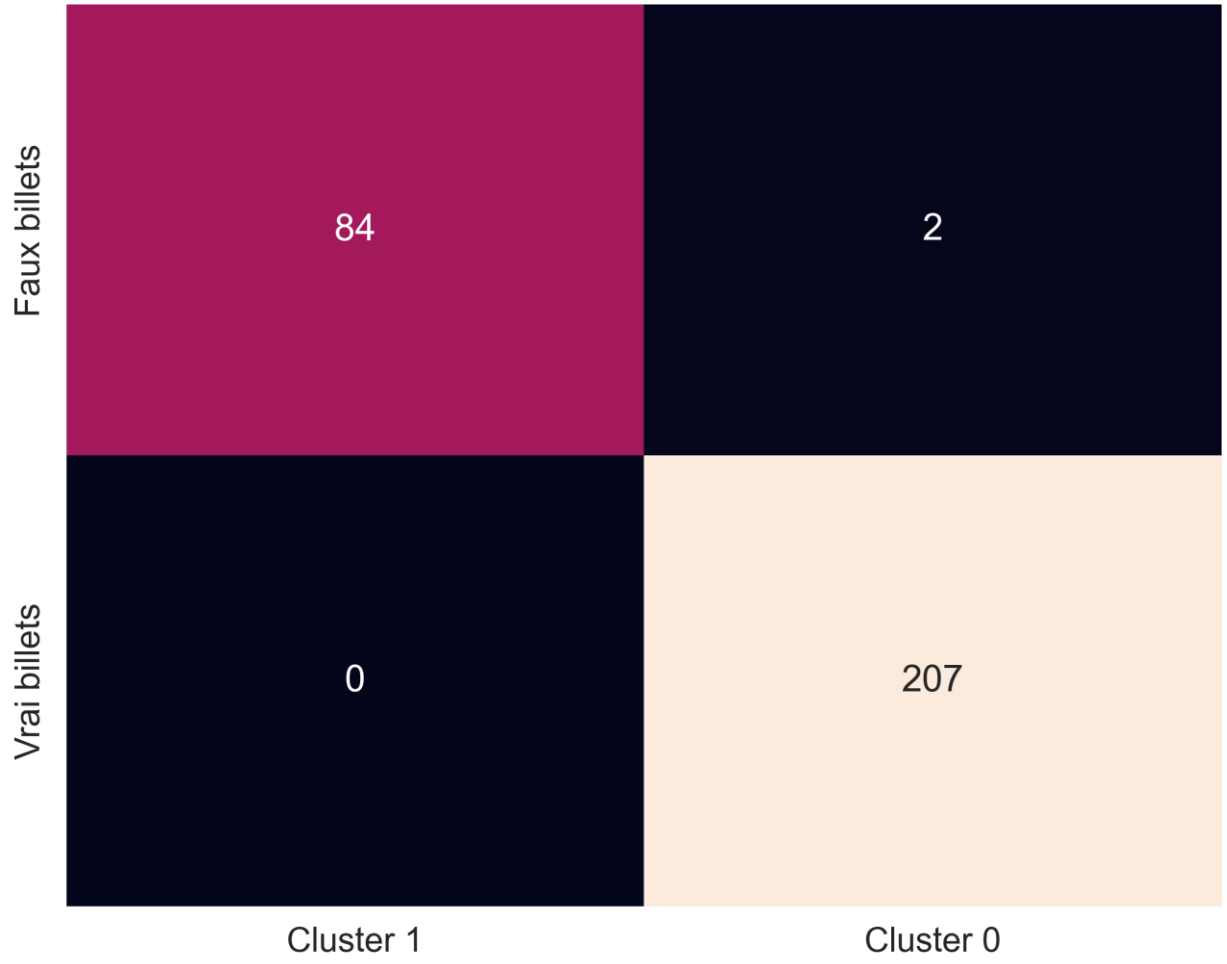
98% score AUC

```
0.989700980148266
```

Régression logistique

Régression logistique :

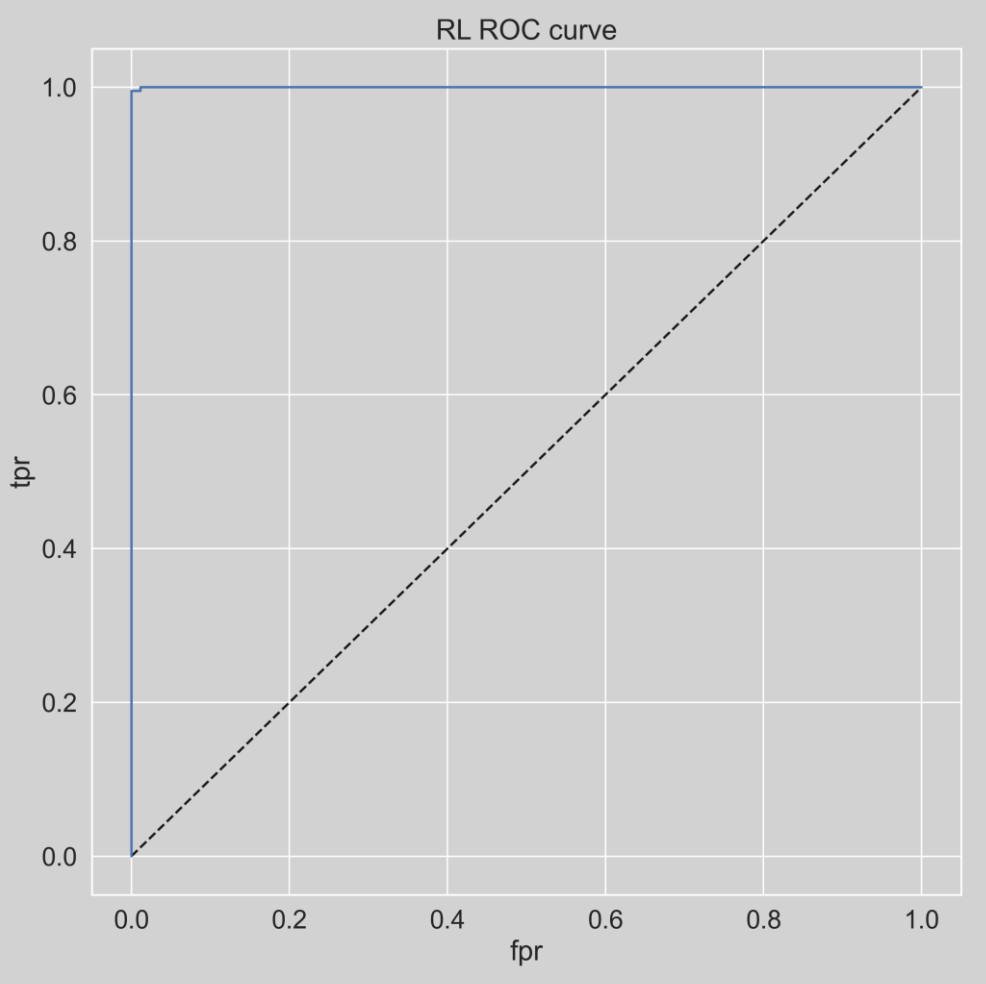
Données prédite :



99% de réussite pour la Régression logistique

	precision	recall	f1-score	support
False	1.00	0.98	0.99	86
True	0.99	1.00	1.00	207
accuracy			0.99	293
macro avg	1.00	0.99	0.99	293
weighted avg	0.99	0.99	0.99	293

Régression logistique roc:



99% score AUC

```
0.9994515357000399
```


Conclusion

Conclusion

Des 3 algorithmes utiliser le KNN et la régression logistique sont ceux avec les meilleurs résultat.

KNN

- Lent quand jeu de données conséquent
- N’est pas demander dans le projet

Régression logistique

- Peut être utiliser sur un jeu de données conséquent
- Demander dans le projet