Projet 10 : Détection de faux billets de banque

NATHAN KELIBY

Sommaire

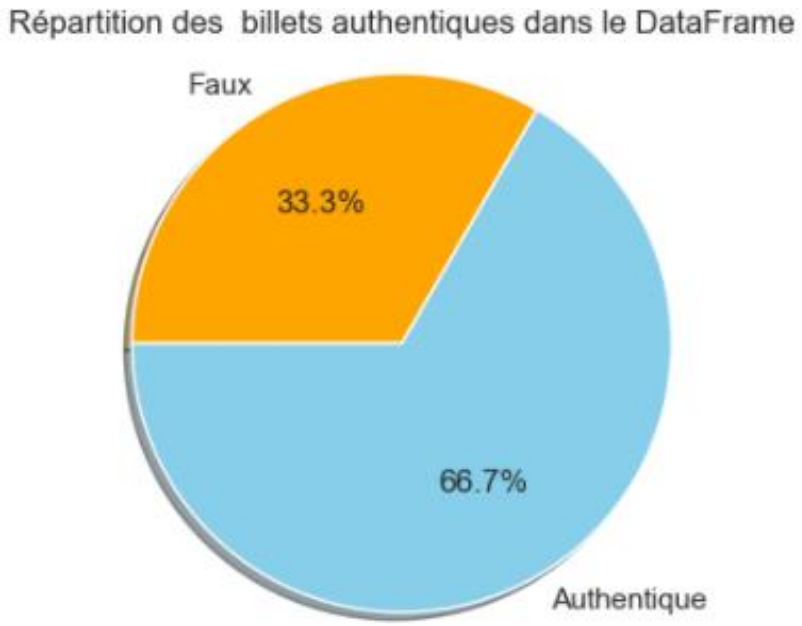
Présentation des données

1 variable qualitative :

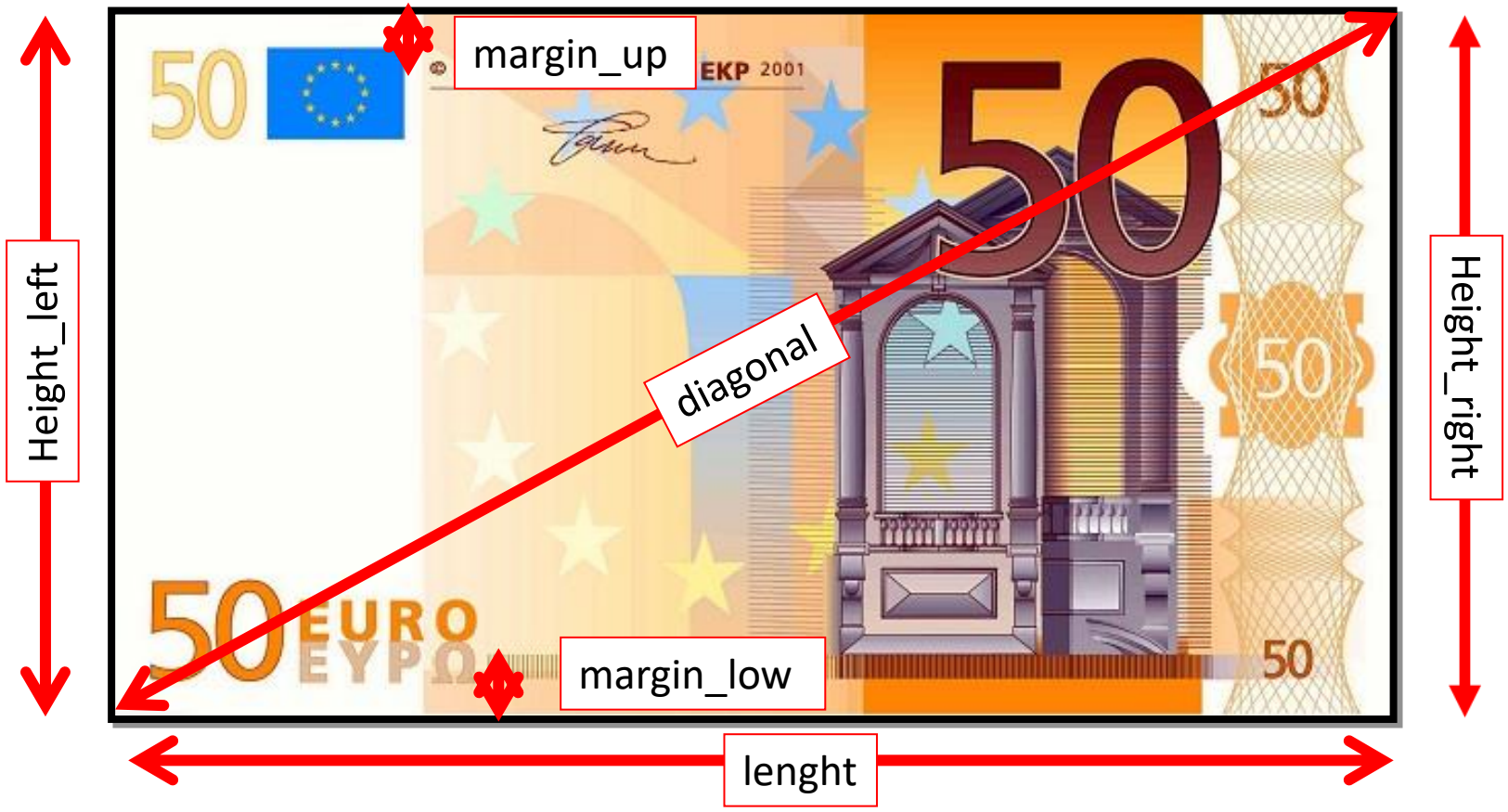
Is_guenine

True : 1000 billets

False : 500 billets



6 variables quantitatives :

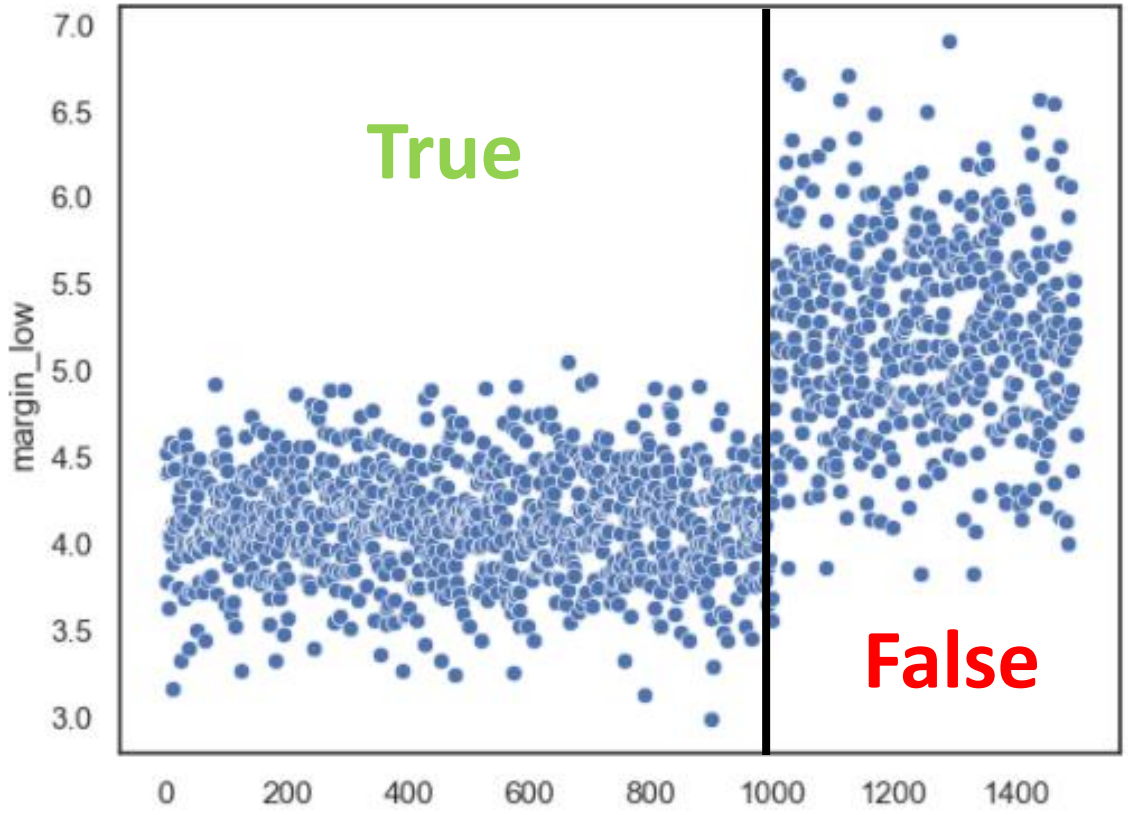


1) Préparation des données

A) Valeurs manquantes

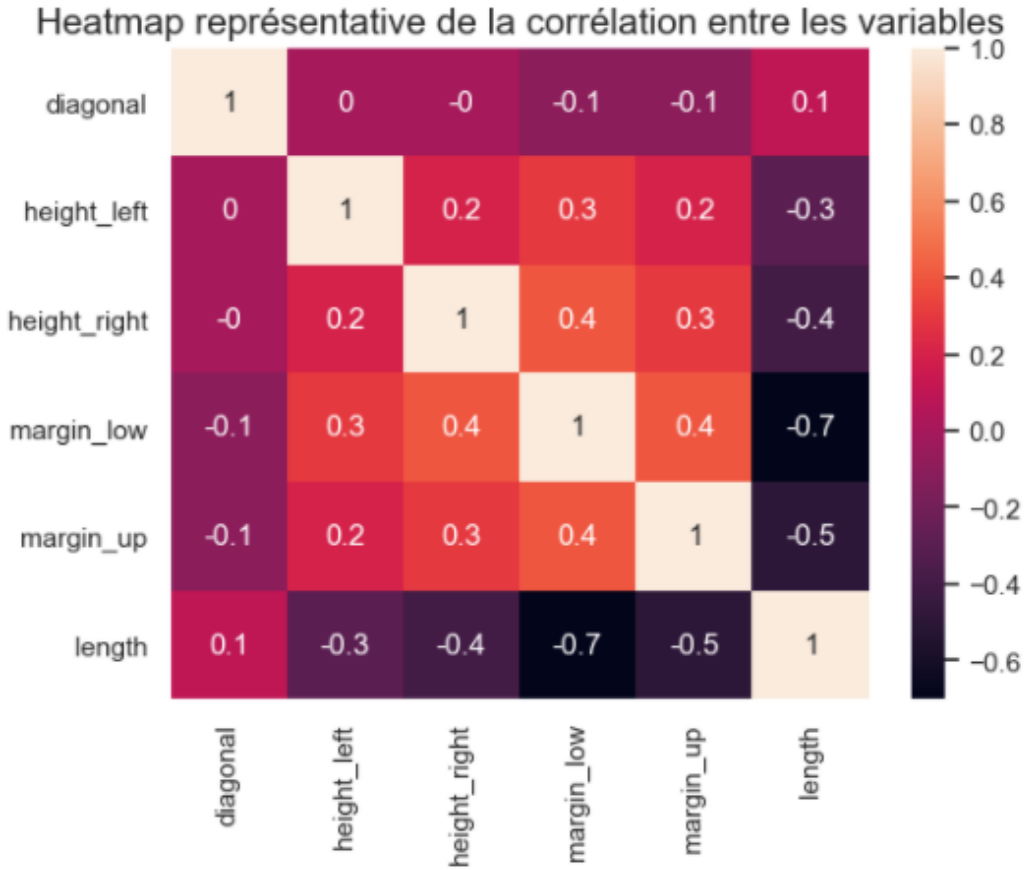
‘margin_low’ : 37 nan

	columns	nb_null	%_null
0	is_genuine	0	0.000000
1	diagonal	0	0.000000
2	height_left	0	0.000000
3	height_right	0	0.000000
4	margin_low	37	2.466667
5	margin_up	0	0.000000
6	length	0	0.000000

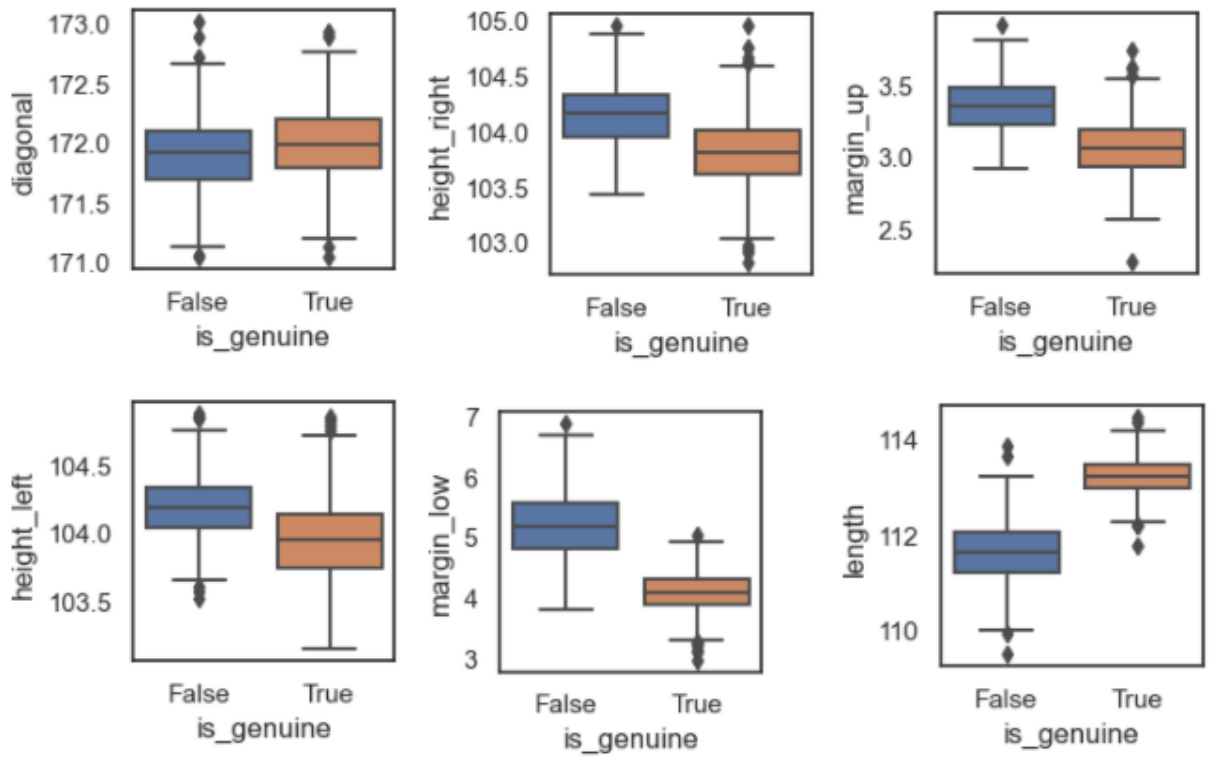


II) Statistiques descriptives

A) Analyse bivariée



Comparaison de la distribution entre les vrais et faux billets



1) Préparation des données

A) Valeurs manquantes : Régression linéaire

R^2 : 45%

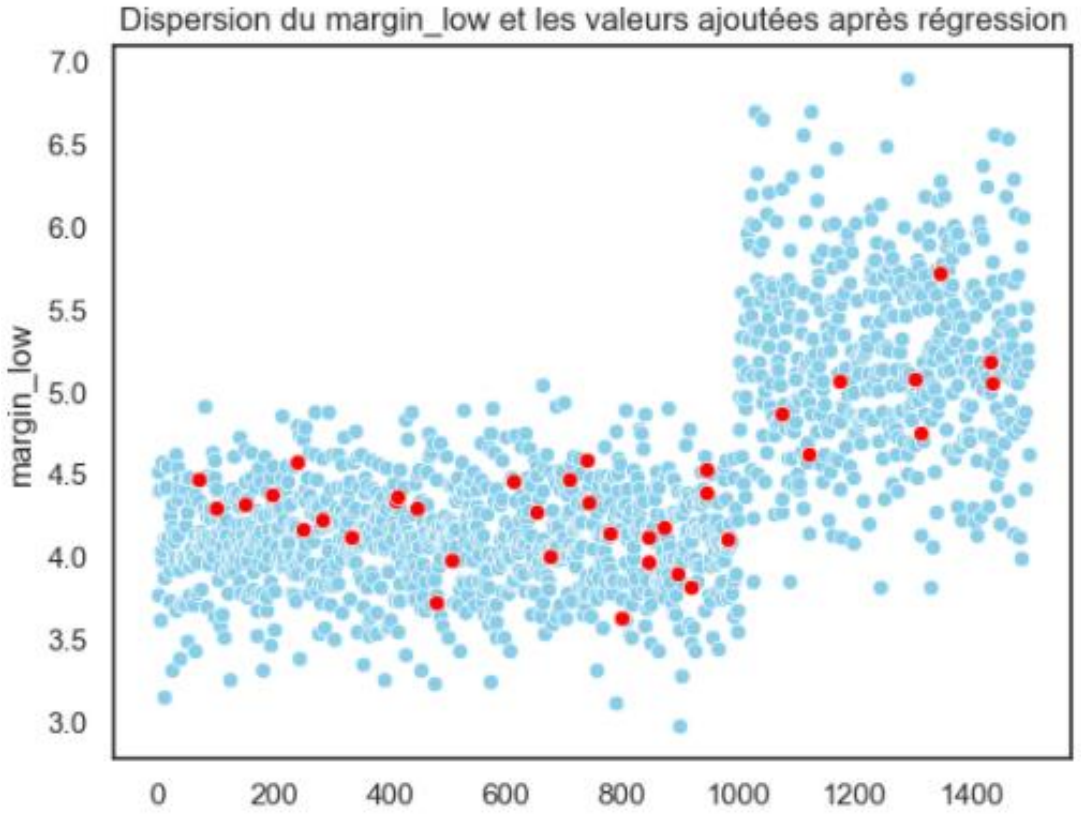
Variables explicatives choisies : length , margin_up

(voir corrélations, test de Levene et test de Shapiro)

```
La performance du modèle sur la base dapprentissage
----------------------------------------
L'erreur quadratique moyenne est 0.49152318343013973
le score R2 est 0.45355759760662573


La performance du modèle sur la base de test
----------------------------------------
L'erreur quadratique moyenne est 0.4850710768524728
le score R2 est 0.4554738163367009
```

Remarque : Le R^2(carré du coefficient de corrélation linéaire) correspond au pourcentage de détermination de la distribution des points par l'equtation de la droite



1) Préparation des données

A) Outliers et standardisation

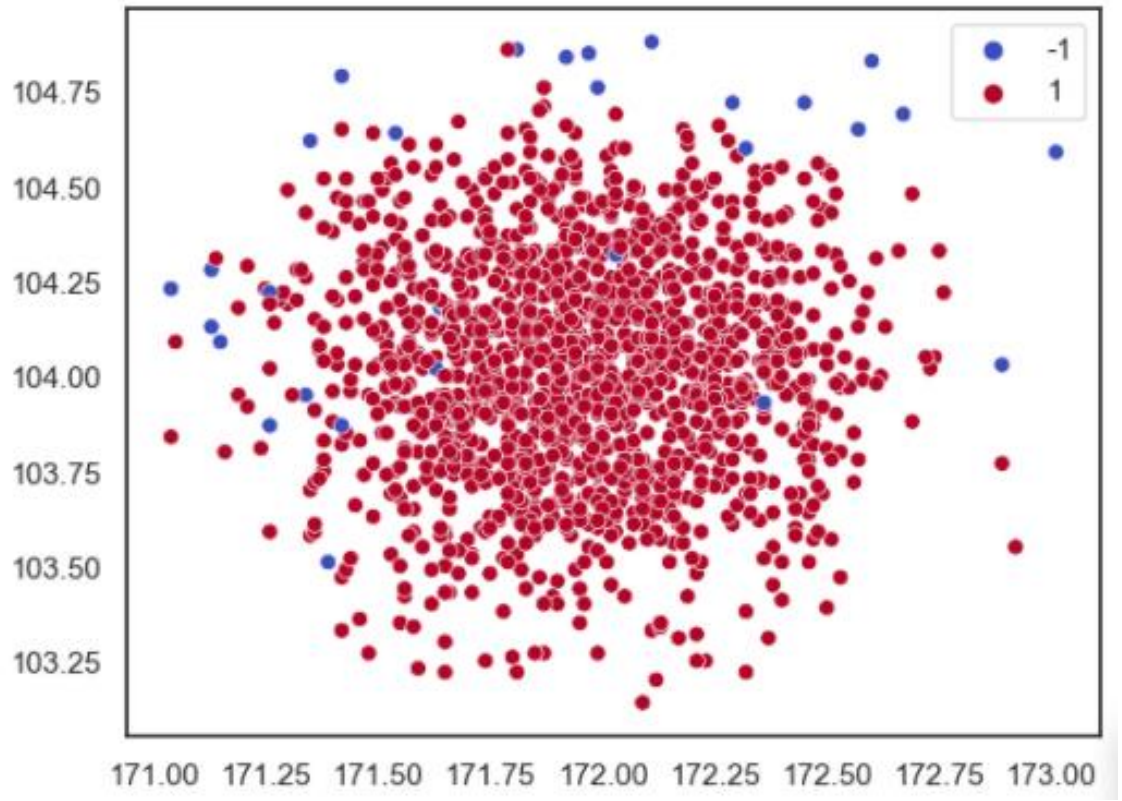
Méthode de détections des outliers :

- Interquartiles
- IsolationForest (contamination = 0.02)

Standardisation des valeurs : RobustScaler()

Remarque : Les outliers ne sont pas des valeurs aberrantes et sont tous de faux billets

Bilan : Conservation des outliers



II) Statistiques descriptives

A) Analyse univariée

Entrée [6]:
```
data.describe()
```

Out[6]:

	diagonal	height_left	height_right	margin_low	margin_up	length
count	1500.000000	1500.000000	1500.000000	1463.000000	1500.000000	1500.00000
mean	171.958440	104.029533	103.920307	4.485967	3.151473	112.67850
std	0.305195	0.299462	0.325627	0.663813	0.231813	0.87273
min	171.040000	103.140000	102.820000	2.980000	2.270000	109.49000
25%	171.750000	103.820000	103.710000	4.015000	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

Remarques:

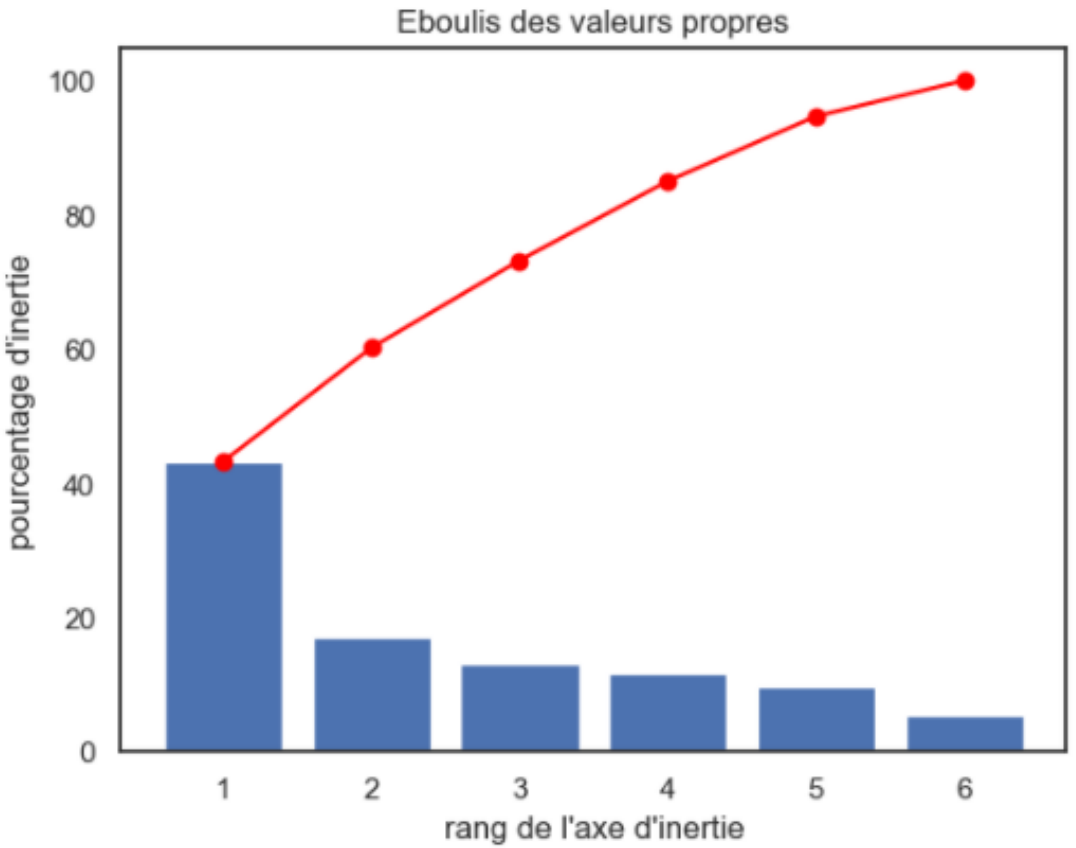
- Les données sont très précises(au centième de mm près)
- Les écart-types sont faibles (<0.9)

II) Statistiques descriptives

B) Analyse en composantes principales (ACP)

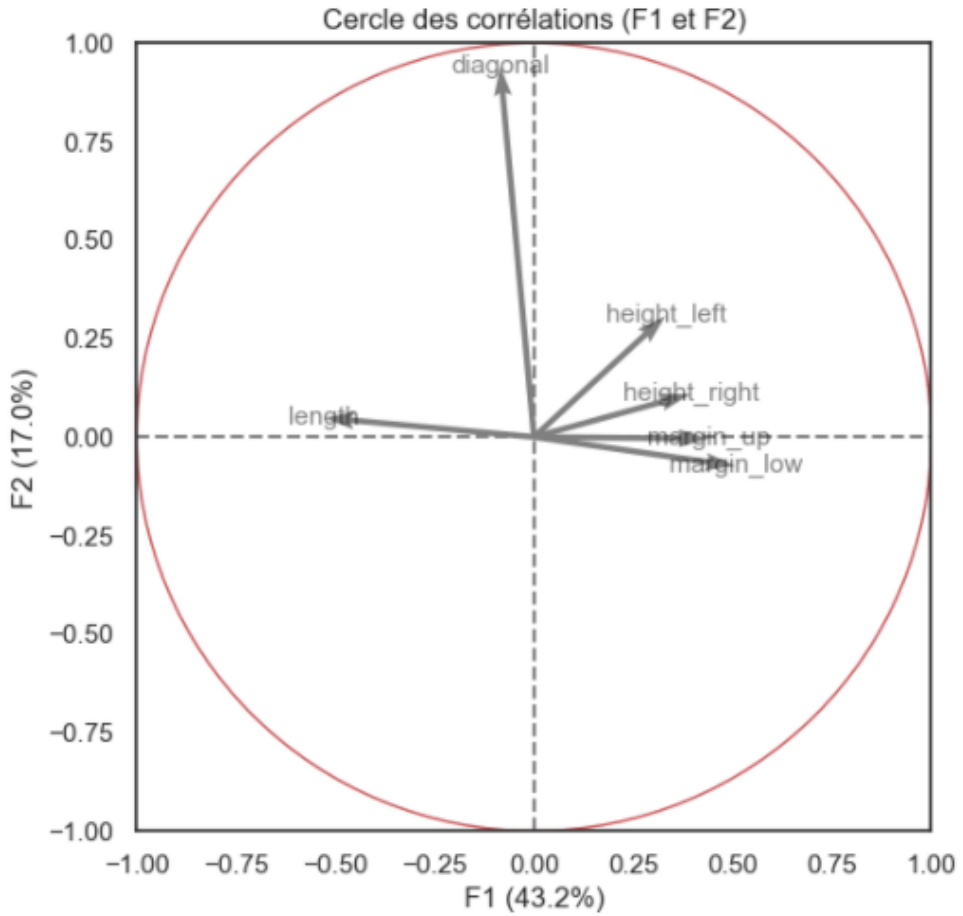
	Dimension	Variance expliquée	% variance expliquée	% cum. var. expliquée
0	F1	2.594392	43.0	43.0
1	F2	1.018153	17.0	60.0
2	F3	0.781630	13.0	73.0
3	F4	0.709942	12.0	85.0
4	F5	0.580198	10.0	95.0
5	F6	0.319688	5.0	100.0

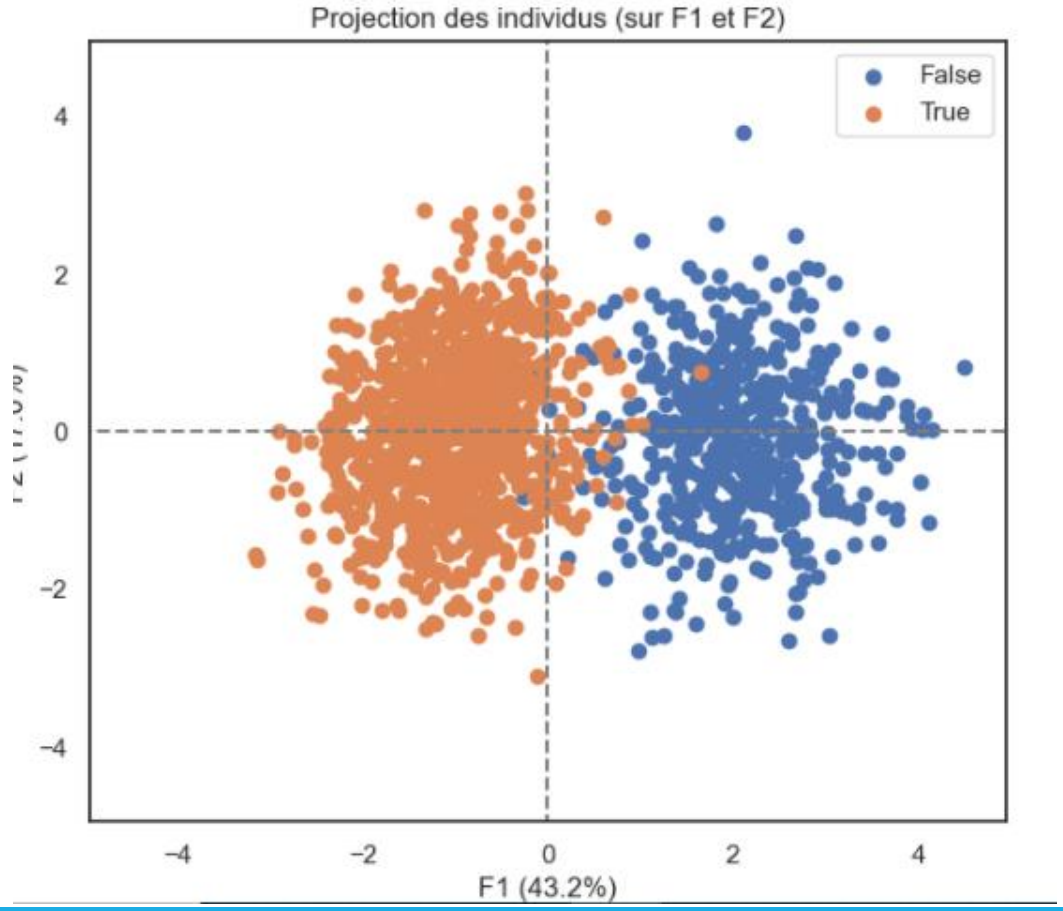
On arrive à environ 95% de variance expliquée pour 5 dimensions



II) Statistiques descriptives

B) Analyse en composantes principales (ACP)





II) Statistiques descriptives

B) Train-test split

Train-test split :

```
# X : Data
# X_norm : Data standardisé

# Variable à expliquer
y = data.is_genuine
```

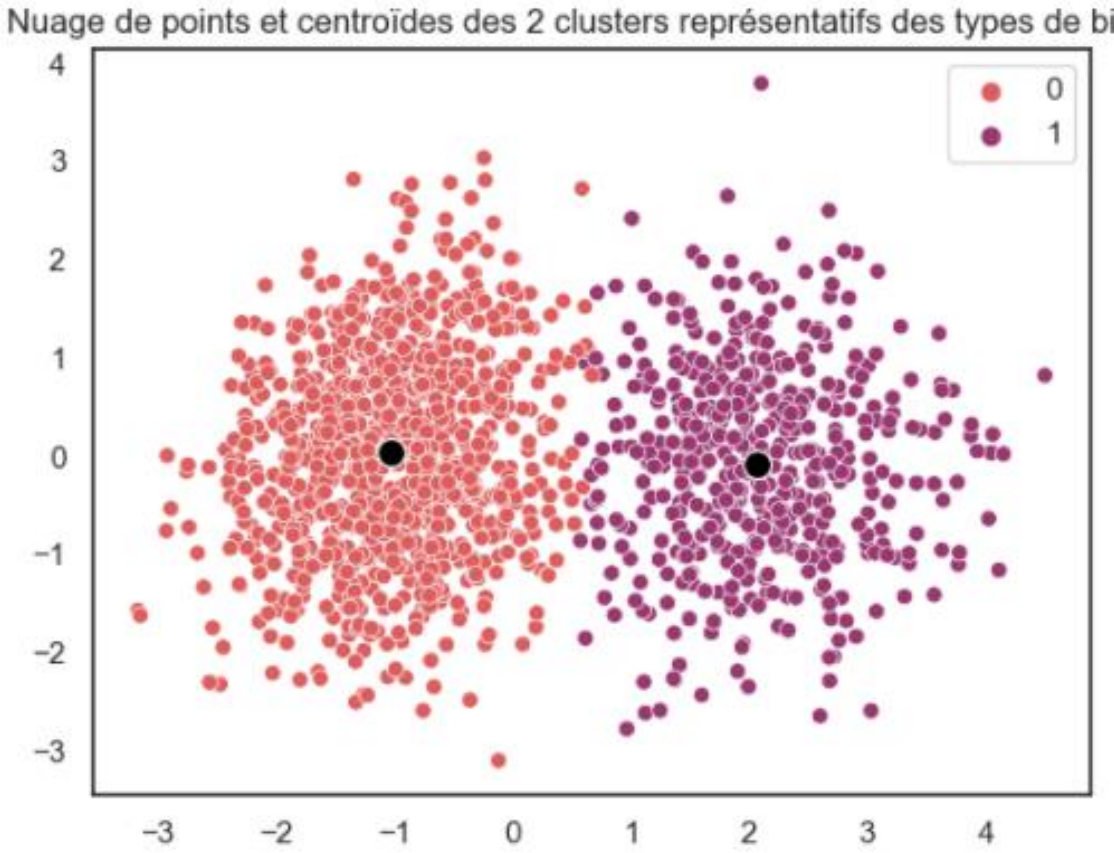
```
# Partition aléatoire du jeu de données en 80% pour créer le modèle, 20% pour tester le modèle
X_train, X_test, y_train, y_test = train_test_split(X_norm, y, test_size=0.20)
```

```
print('Train Set :',X_train.shape)
print('Train Test :',X_test.shape)
```

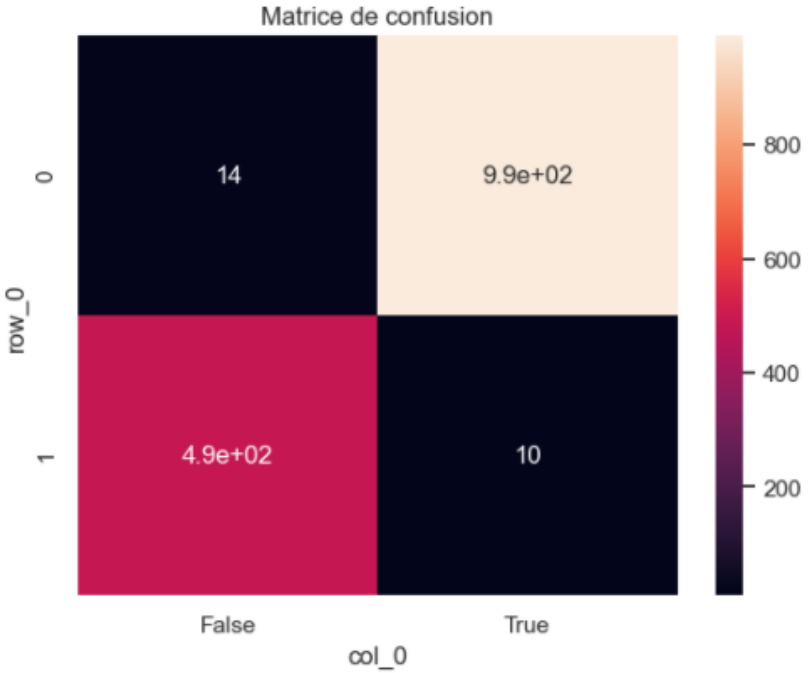
```
Train Set : (1200, 6)
Train Test : (300, 6)
```

II) Statistiques descriptives

B) Apprentissage non-supervisé : Classification k-means

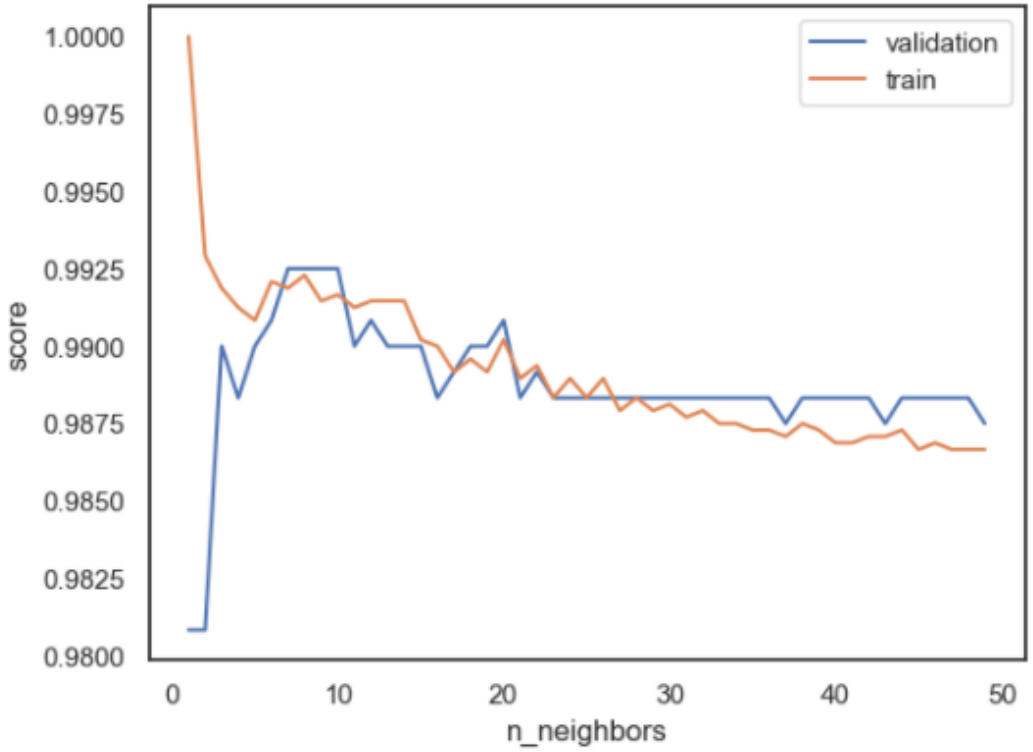


Score = 98.4%



II) Statistiques descriptives

B) Apprentissage supervisé : KN-Neighboors

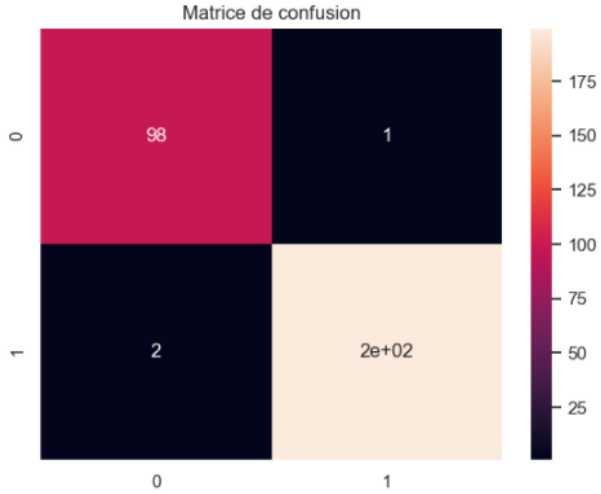


Score = 99,3%

- Cross validation : 5 split
- Validation curve : jusqu'à 50 voisins
- GridSearchCV :

  Bestparams : metric = 'manhattan'

  n_neighboors = 8



II) Statistiques descriptives

B) Apprentissage supervisé : Régression logistique

Coefficients :

Logit Regression Results

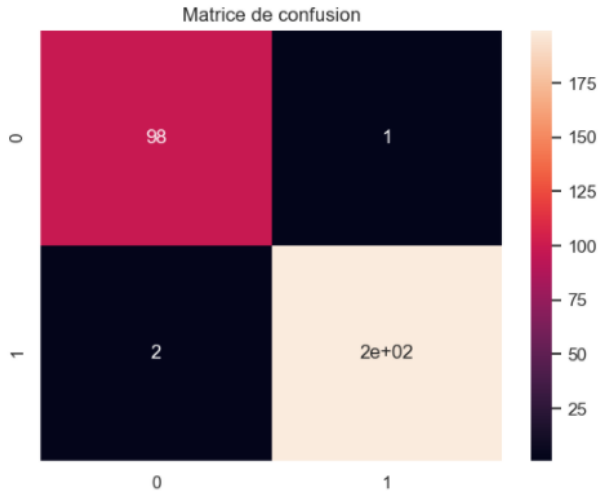
Dep. Variable:	is_genuine	**No. Observations:**	1500
Model:	Logit	**Df Residuals:**	1494
Method:	MLE	**Df Model:**	5
Date:	Wed, 22 Dec 2021	**Pseudo R-squ.:**	0.6957
Time:	13:17:09	**Log-Likelihood:**	-290.53
converged:	True	**LL-Null:**	-954.77
Covariance Type:	nonrobust	**LLR p-value:**	4.319e-285

	coef	std err	z	P>|z|	[0.025	0.975]
x1	0.4180	0.144	2.894	0.004	0.135	0.701
x2	-0.7560	0.148	-5.118	0.000	-1.045	-0.466
x3	-1.1284	0.171	-6.595	0.000	-1.464	-0.793
x4	-3.4943	0.325	-10.758	0.000	-4.131	-2.858
x5	-1.8066	0.198	-9.116	0.000	-2.195	-1.418
x6	5.1831	0.449	11.534	0.000	4.302	6.064

Possibly complete quasi-separation: A fraction 0.23 of observations can be perfectly predicted. This might indicate that there is complete quasi-separation. In this case some parameters will not be identified.

Score = 99%

- Cross validation : 5 split
- GridSearchCV :
  Bestparams : solver = ‘newton-cg’



Bilan : Choix du model final pour les tests

	K-means	KNN	Régression logistique
Score :	98.4%	99,3%	99%

Choix de la régression logistique car l'objectif étais d'opposer celle-ci avec le k-means, mais le modèle à privilégier est le KNN