Machine Learning

DÉTECTEZ DES FAUX BILLETS

Elodie Mendes

ONCFM

MISSION

Lutte contre le faux -monnayage avec des méthodes d' identification de faux billets.

OBJECTIF

Application de machine learning pour une détection plus rapide et efficace.

MÉTHODOLOGIE

Détection des faux billets

Nettoyage de données → Analyse exploratoire des données → Mise en place des algorithmes

Machine Learning

NETTOYAGE DE DONNÉES

PRÉPARATION DES DONNÉES

Analyse exploratoire

DONNÉES INITIALES

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
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_low    1463 non-null   float64
 5   margin_up     1500 non-null   float64
 6   length        1500 non-null   float64
dtypes: bool(1), float64(6)
```

VALEURS MANQUANTES

```
df.isnull().sum()
```

```
is_genuine       0
diagonal         0
height_left      0
height_right     0
margin_low      37
margin_up        0
length           0
dtype: int64
```

PRÉPARATION DES DONNÉES

Régression linéaire

1. **Déterminer les valeurs prédictives et la valeur cible**
2. **Division des données pour Entraînement/Test**
3. **Prédictions**
4. **Evaluation de la performance du modèle**

PRÉPARATION DES DONNÉES

Régression linéaire

EVALUATION DE LA PERFORMANCE DU MODELE

```
print(f"RMSE: {mean_squared_error(y_test, y_pred_test)}")
print(f"MAPE: {mean_absolute_percentage_error(y_test, y_pred_test)}")
```

```
RMSE: 0.1373945287178082
MAPE: 0.0649607417221153
```

```
df.info()
```

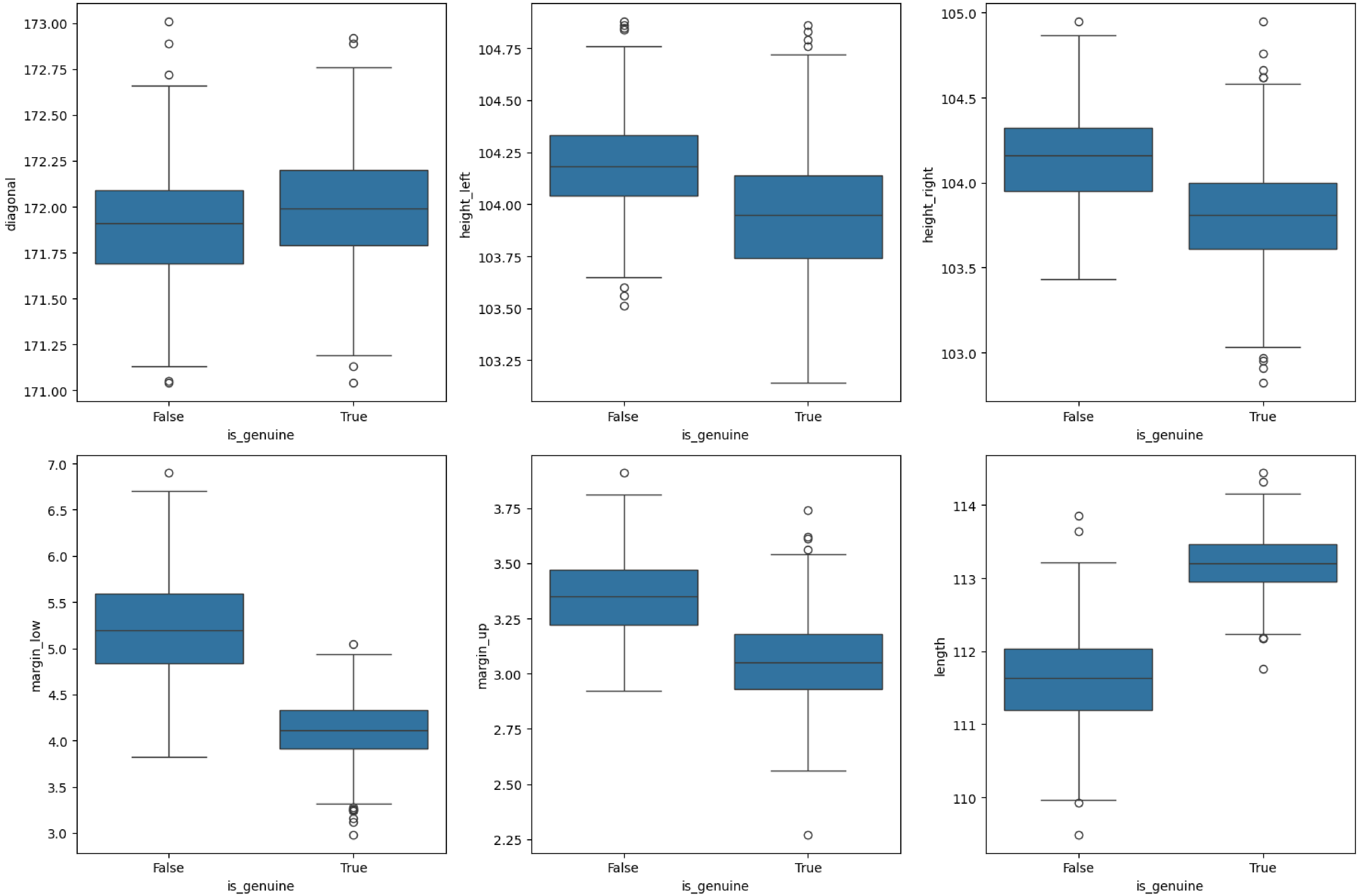
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   is_genuine    1500 non-null   bool   
 1   diagonal      1500 non-null   float64
 2   height_left   1500 non-null   float64
 3   height_right  1500 non-null   float64
 4   margin_low    1500 non-null   float64
 5   margin_up     1500 non-null   float64
 6   length        1500 non-null   float64
dtypes: bool(1), float64(6)
```

Machine Learning

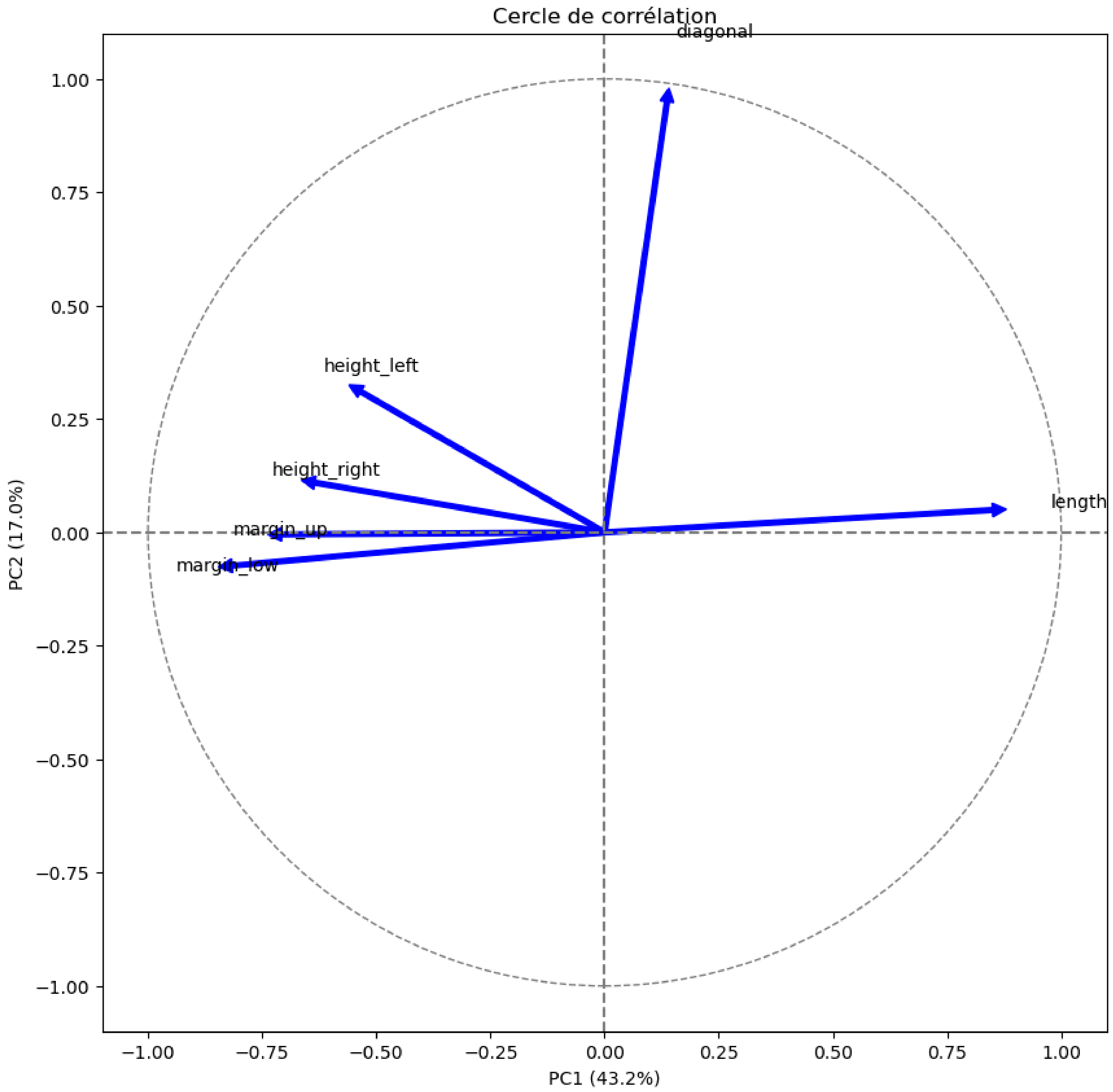
EXPLORATIONS DES DONNÉES

EXPLORATION DES DONNÉES

Caractéristiques des billets



Réduction des dimensions: ACP



1er axe factoriel: Longueur du billet

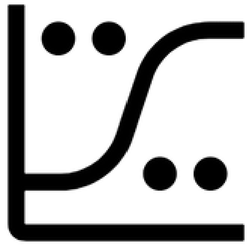
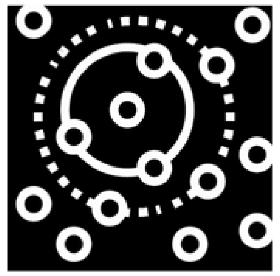
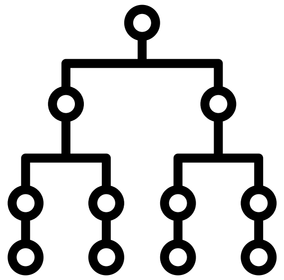
2ème axe factoriel: Diagonal du billet

Corrélations entre marge et hauteur d’un billet

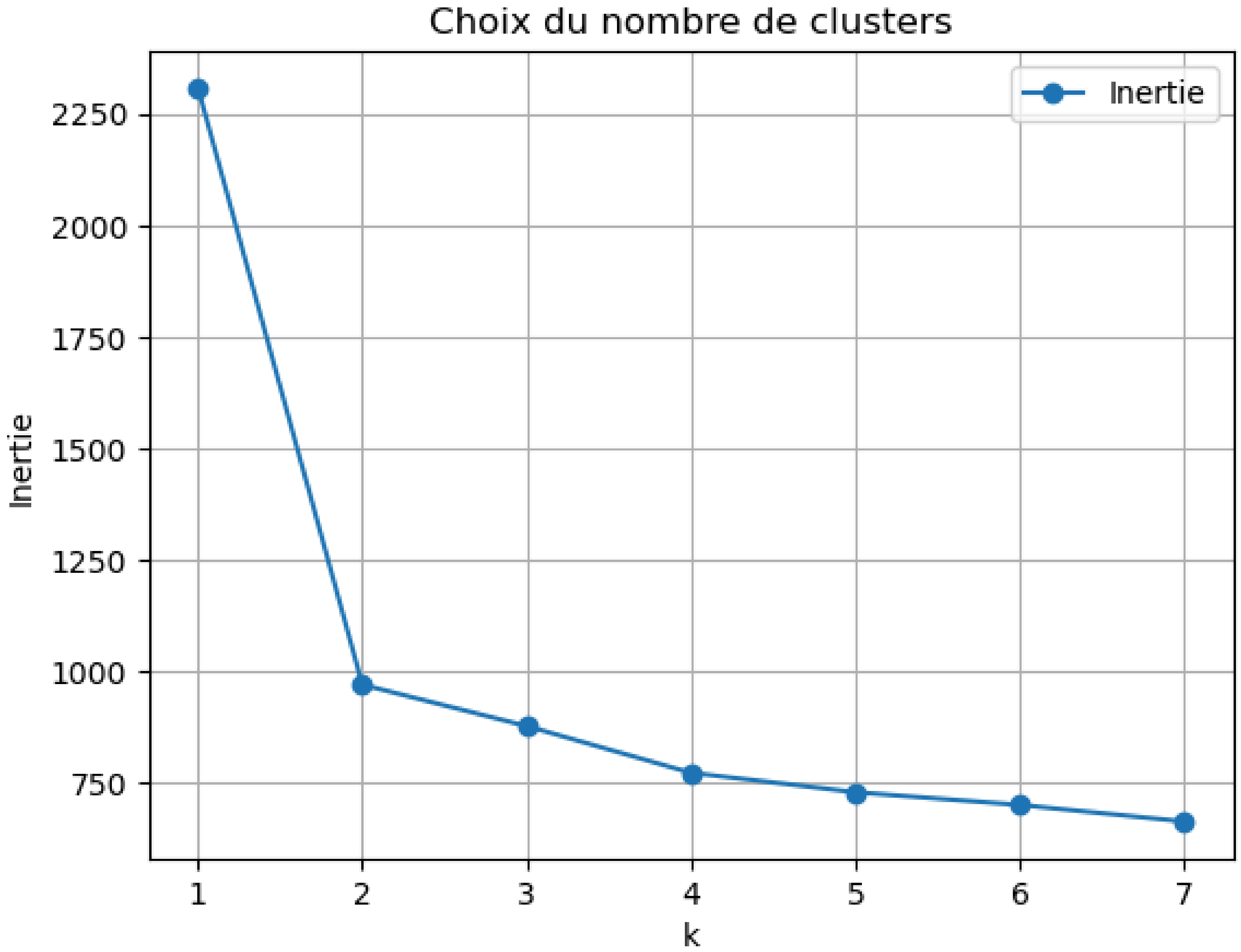
Machine Learning

MISE EN PLACE DES ALGORITHMES

Test de 4 algorithmes

			
Régression logistique	K-NN	K-means	Random Forest

Algorithme non supervisée : K-means

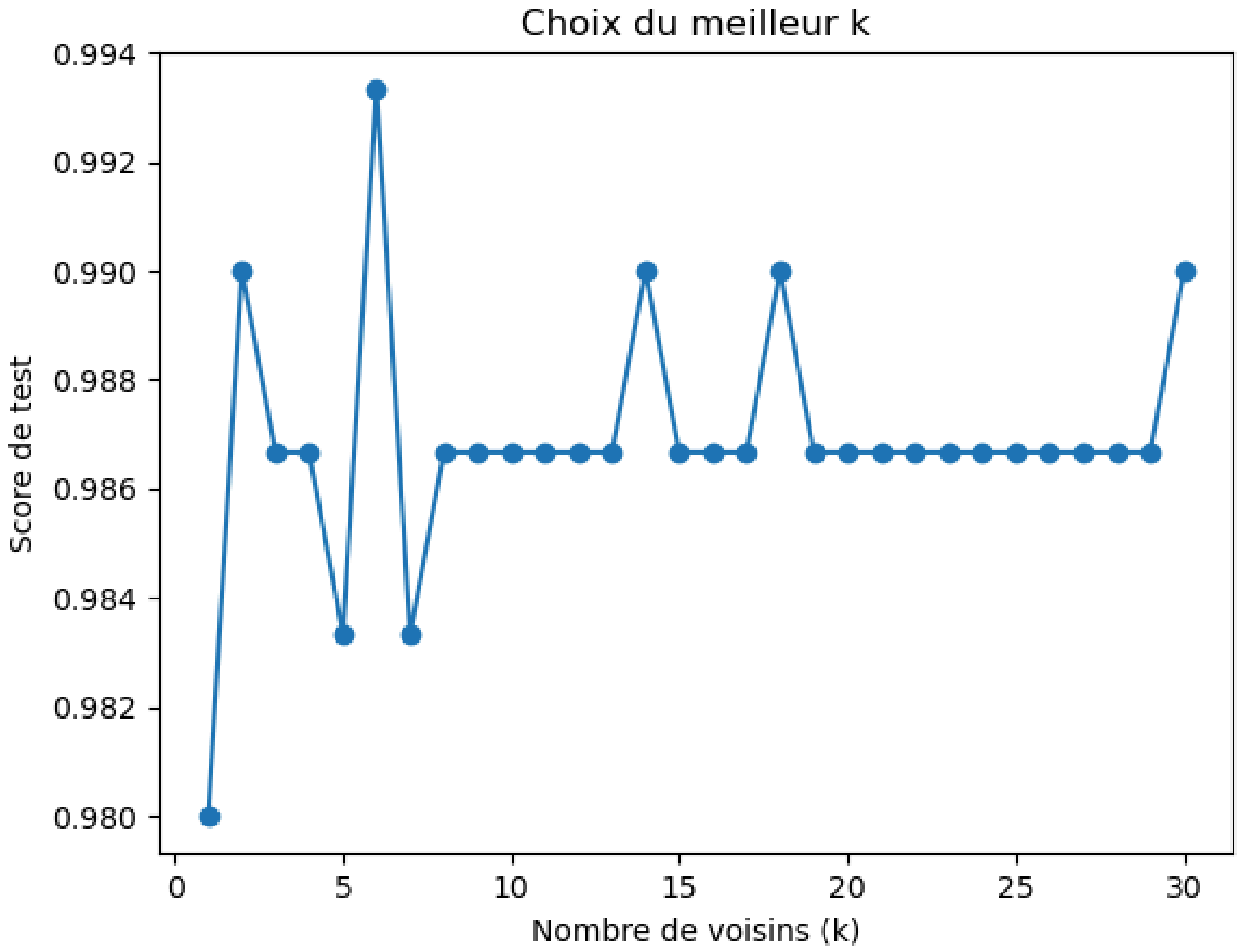


Performance

Performance du K-means:

	precision	recall	f1-score	support
False	1.00	0.95	0.97	110
True	0.97	1.00	0.98	190
accuracy			0.98	300

Algorithme supervisée : K-nearest neighbors



Performance

Résultat du K-nearest_neighbors:

	precision	recall	f1-score	support
False	1.00	0.96	0.98	110
True	0.98	1.00	0.99	190
accuracy			0.99	300

Méthode supervisée

Performance

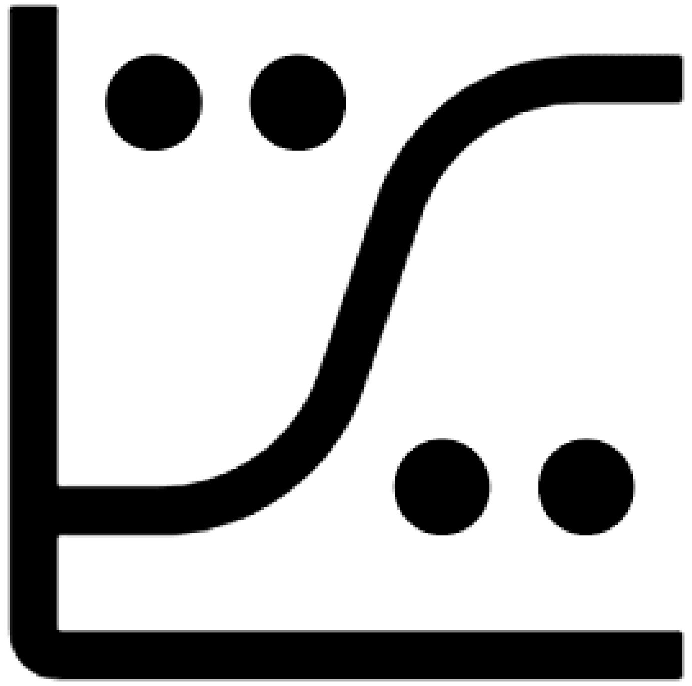
Résultat du Random forest :

	precision	recall	f1-score	support
False	1.00	0.97	0.99	110
True	0.98	1.00	0.99	190
accuracy			0.99	300

Random Forest

Méthode supervisée

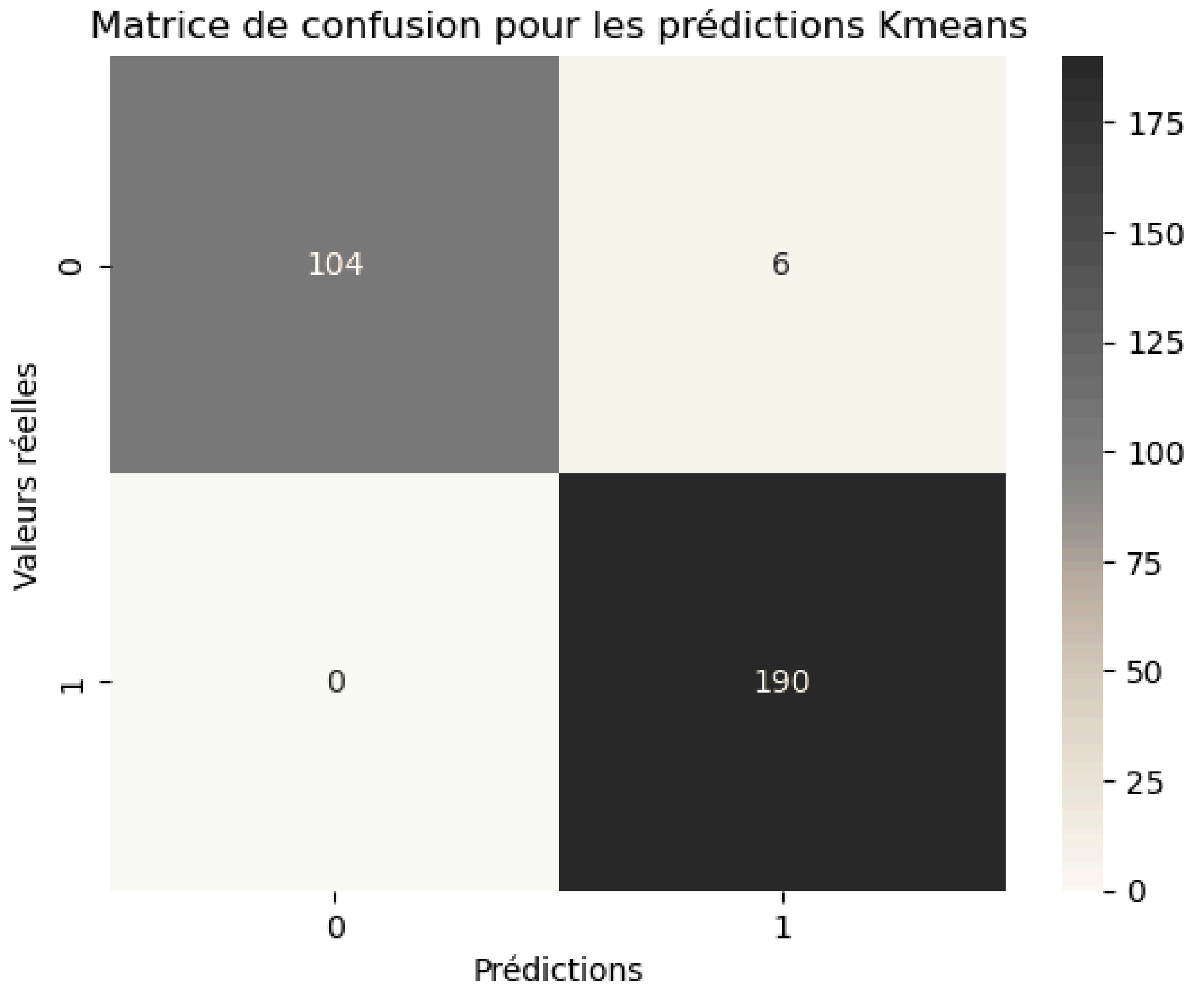
Performance

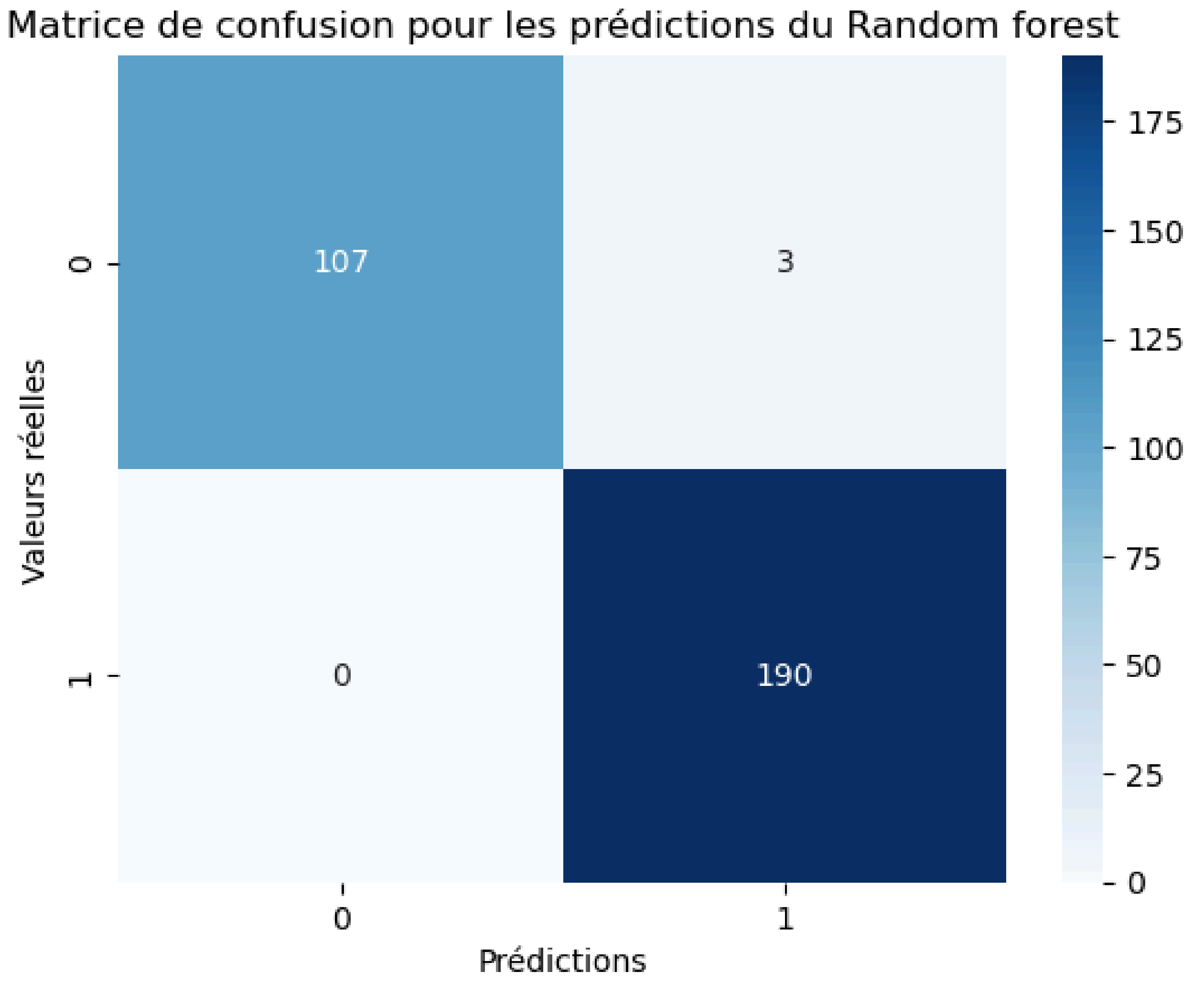


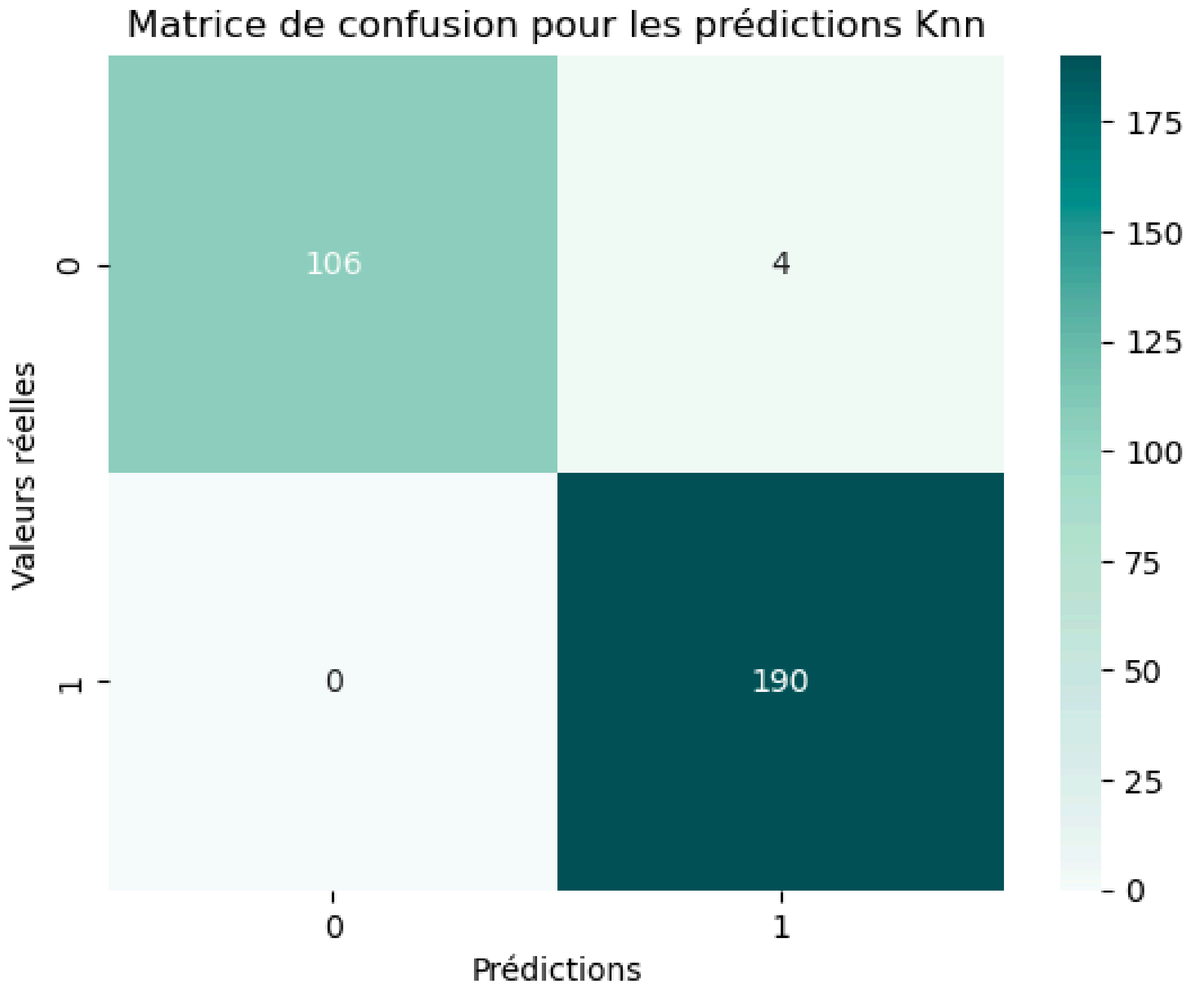
Résultat de la Régression logistique:

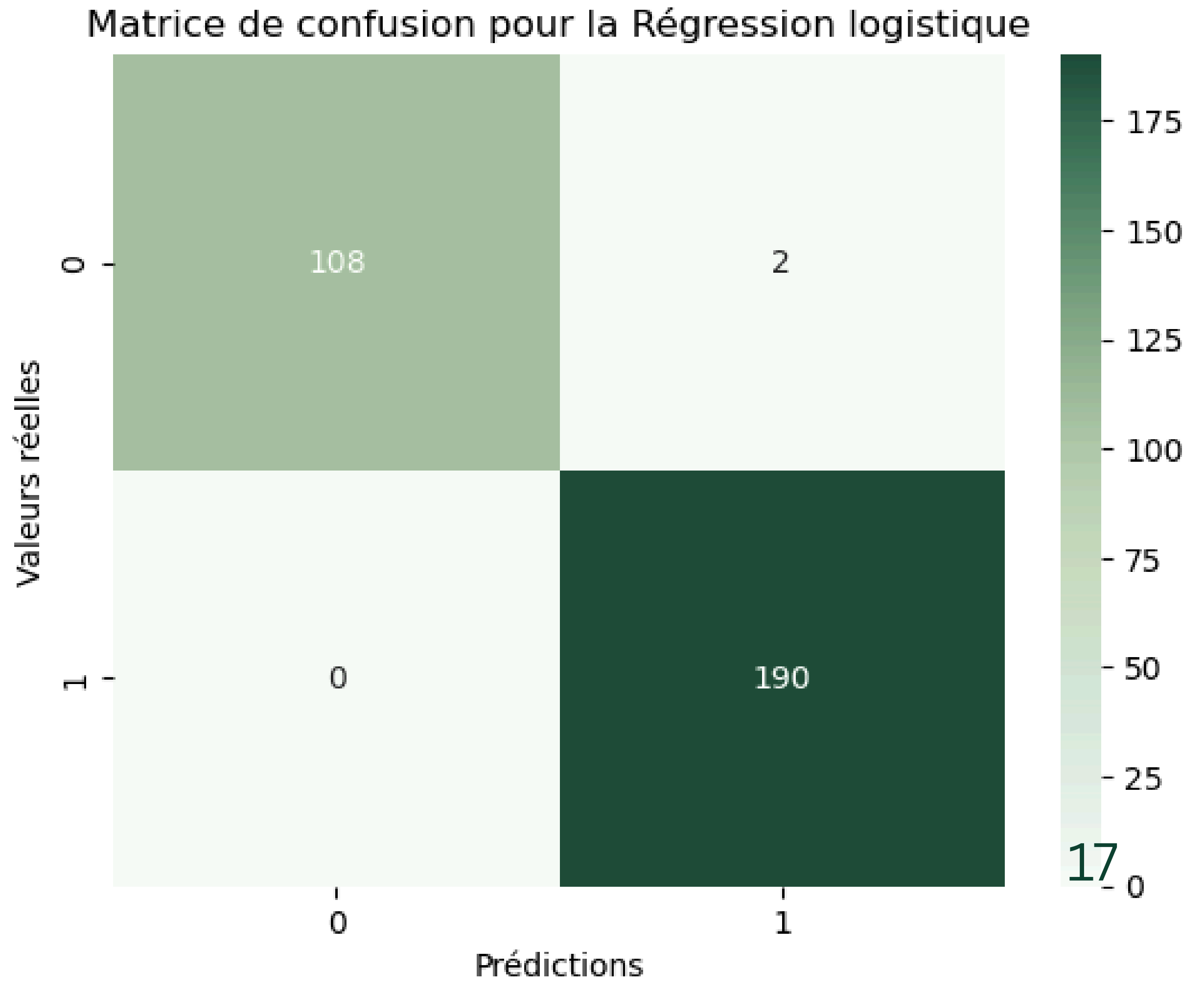
	precision	recall	f1-score	support
False	1.00	0.98	0.99	110
True	0.99	1.00	0.99	190
accuracy			0.99	300

Comparaison des matrices de confusion









MODÈLE RETENU: LA RÉGRESSION LOGISTIQUE

- Le plus performant : 98% de faux billets détectés et seulement 2 faux billets prédits comme vrais.
- Robuste et simple d'interprétation
- Création de l'application "**Prédire_billets**"

```
resultats = predire_billets("billets_pro.csv")
resultats
```

```
Prédictions terminées. Fichier enregistré sous 'resultats_prediction_billets.csv'
```

	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length	prediction	probabilité
0	True	171.81	104.86	104.95	4.52	2.89	112.83	True	0.998871
1	True	171.46	103.36	103.66	3.77	2.99	113.09	True	0.999999
2	True	172.69	104.48	103.50	4.40	2.94	113.16	True	0.999994
3	True	171.36	103.91	103.94	3.62	3.01	113.51	True	1.000000
4	True	171.73	104.28	103.46	4.04	3.48	112.54	True	0.979807
...	...	...	...	...	...	...	...	...	...
1495	False	171.75	104.38	104.17	4.42	3.09	111.28	True	0.532321
1496	False	172.19	104.63	104.44	5.27	3.37	110.97	False	0.999767
1497	False	171.80	104.01	104.12	5.51	3.36	111.95	False	0.970593
1498	False	172.06	104.28	104.06	5.17	3.46	112.25	False	0.909426