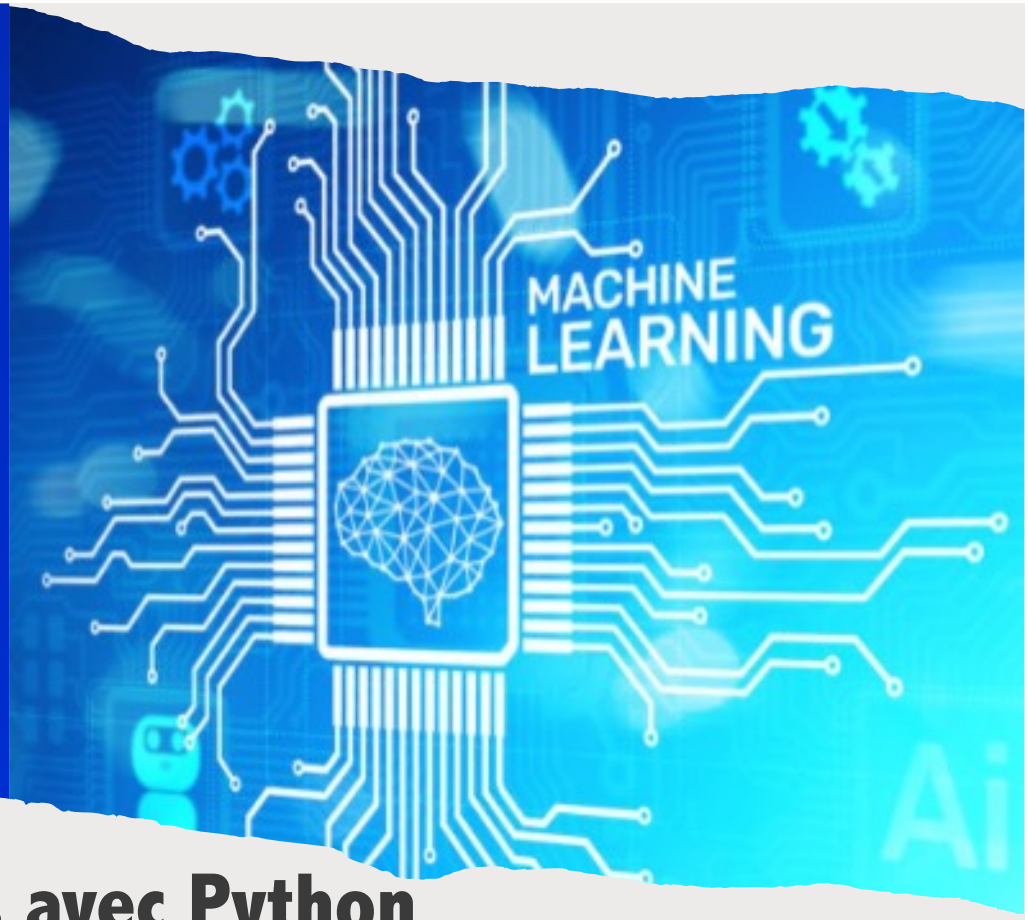
ONCFM



Détecter des faux billets avec Python

Caroline
Data Analyst

14/12/2024

Déroulé

- Contexte
  - Traitement des données
    - Mise en concurrence de modèles d'apprentissage
      - Résultats de l'analyse
        - Test pour mise en production

Contexte

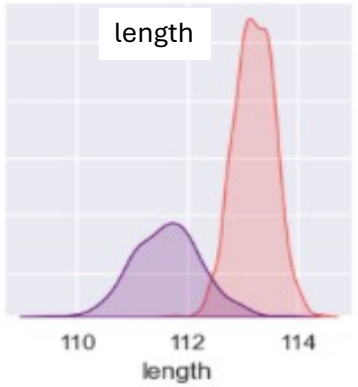
- L'organisation nationale de lutte contre le faux-monnayage souhaite mettre en place des méthodes d'identification des faux billets en euros pour lutter contre la contrefaçon

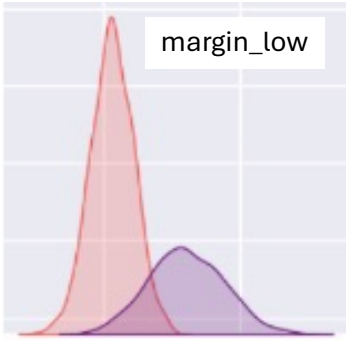
- mettre à disposition des équipes une application de machine learning à partir de scan de billets

Traitement des données

- Jeu de données : dimensions de 1500 billets scannés
  (Identifiés vrai/faux)

- 37 valeurs manquantes (margin_low)





Traitement des données

Données manquantes

-> Régression linéaire multiple

On constate que certains paramètres ne sont pas significativement différents de 0, car leur p-valeur n'est pas inférieure à 5 %, le niveau de test que nous souhaitons

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:	Tue, 03 Dec 2024	Prob (F-statistic):	4.75e-299
Time:	17:40:39	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

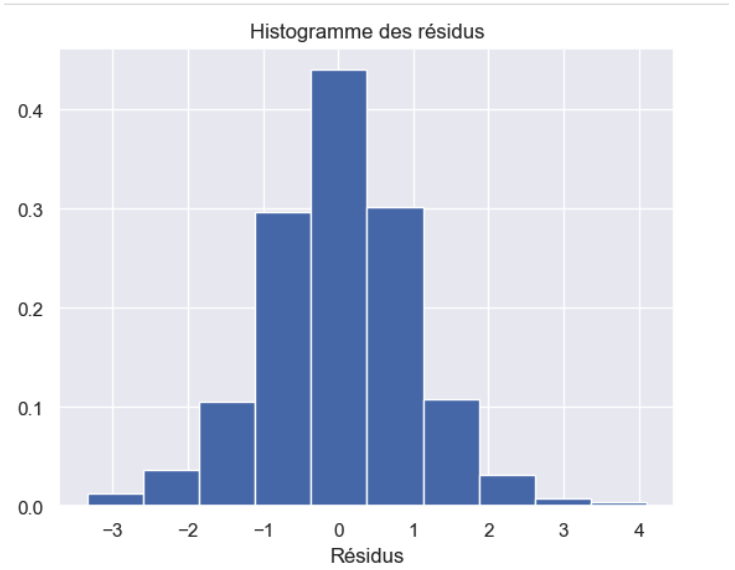
Traitement des données

Non colinéarité des variables ✓

Homoscédasticité des résidus ✗✓

H0 rejetée (car p-valeur <<5%) pour les tests :
- Breusch Pagan
- Shapiro-Wilk

Cependant, 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



Mise en concurrence de modèles d'apprentissage

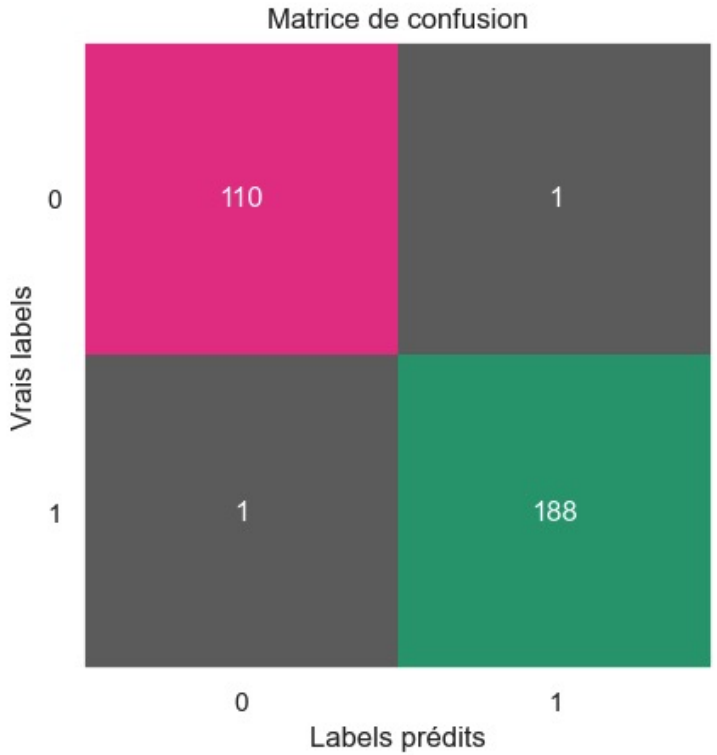
- Régression logistique
- KNN
- K-Means
- Random Forest

Régression logistique

Standardisation des données

Validation croisée stratifiée

- Nombre de vrais positifs : 188
- Nombre de vrais négatifs : 110
- Nombre de faux positifs : 1
- Nombre de faux négatifs : 1

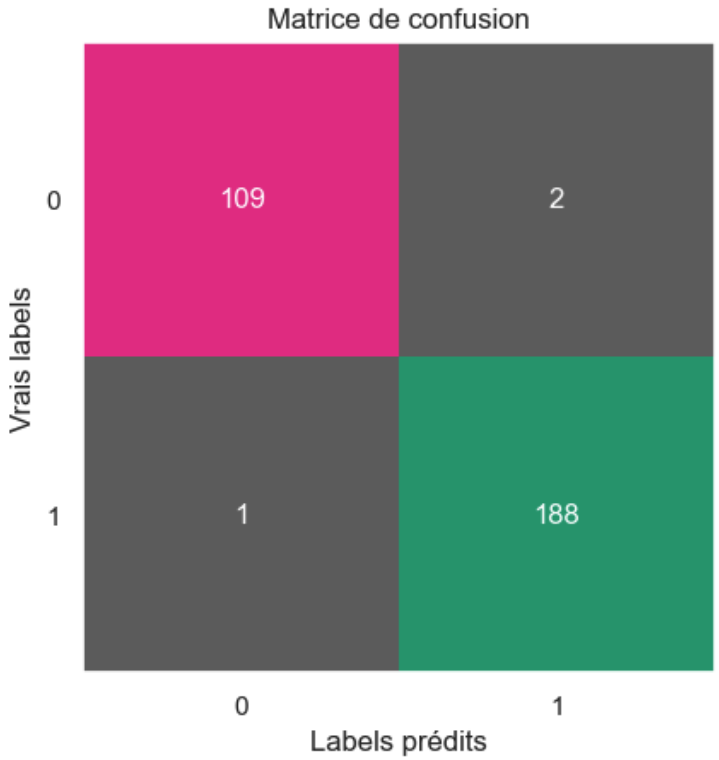


KNN – Algorithme des k plus proches voisins

Standardisation des données

\+ Validation croisée stratifiée

- Nombre de vrais positifs : 188
- Nombre de vrais négatifs : 109
- Nombre de faux positifs : 2
- Nombre de faux négatifs : 1

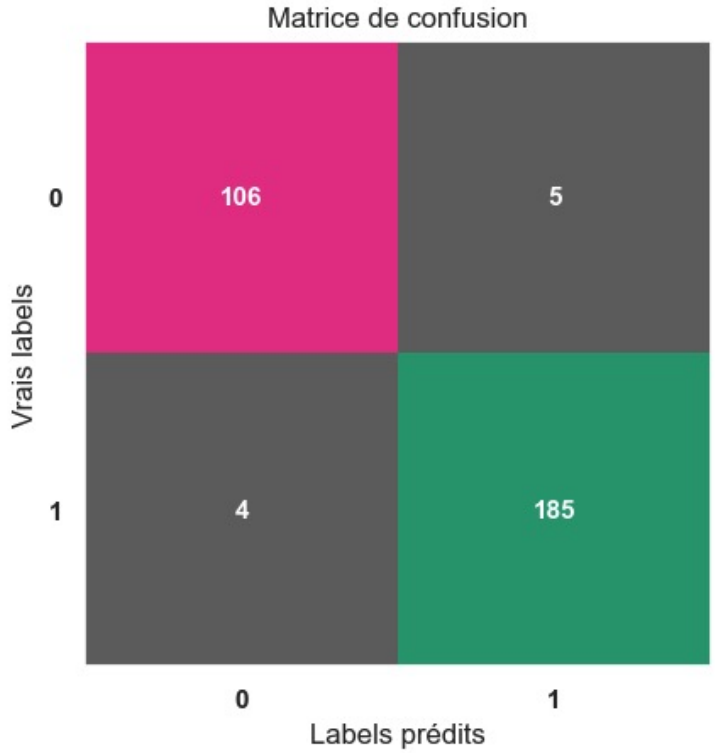


Le nombre de faux positifs est plus élevé, alors que nous cherchons à minimiser surtout ce paramètre = augmenter la précision

K-Means

+ Standardisation des données
+ 2 clusters souhaités

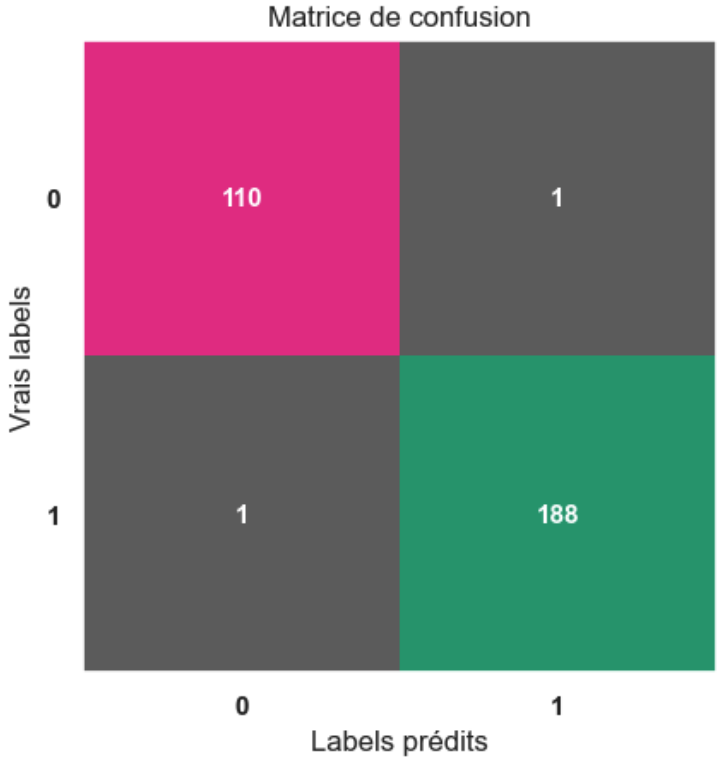
- Nombre de vrais positifs : 185
- Nombre de vrais négatifs : 106
- Nombre de faux positifs : 5
- Nombre de faux négatifs : 4



Random Forest

+Validation croisée stratifiée

- Nombre de vrais positifs : 188
- Nombre de vrais négatifs : 110
- Nombre de faux positifs : 1
- Nombre de faux négatifs : 1



Résultats

Modèle	Accuracy	Précision	F-Score
Régression Logistique	0.9933	0.9947	0.99
KNN	0.9900	0.9947	0.99
K-Means	0.9700	0.9736	0.97
Random Forest	0.9933	0.9947	0.99

Test pour mise en production

Application basée sur la régression logistique