Projet n°6 – Détectez les faux billets

JÉRÉMY VANGANSBERG - OPENCLASSROOMS

Introduction

Introduction

- Analyser un jeu de données sur des billets de banque
- Chaque billet est décrit par des variables dimensionnels
- Le jeu de données est **étiqueté**, certains billets sont vrais et d'autres sont faux
- A partir de ces informations, les objectifs sont :
  - Analyser le jeu de données
  - Utiliser des algorithmes de classification
  - Effectuer des prédictions

Sommaire

A. Caractéristiques du jeu de données
B. Analyse univariée et bivariée
C. Analyse en composantes principales
D. Détails des algorithmes
E. Kmeans
F. Régression logistique

Caractéristiques du jeu de données

Jeu de données (*Dataset*)

- Caractéristiques du dataset :
  - **170 entrées** décrivant des billets de banque avec 7 caractéristiques :
    - « diagonal »
    - « height_left »
    - « height_right »
    - « margin_low »
    - « margin_up »
    - « length »

    Dimensions en mm
    - « is_genuine » : cette variable est un ***label*** qui indique si le billet est vrai ou faux.

Analyse univariée et bivariée

Nettoyage et description

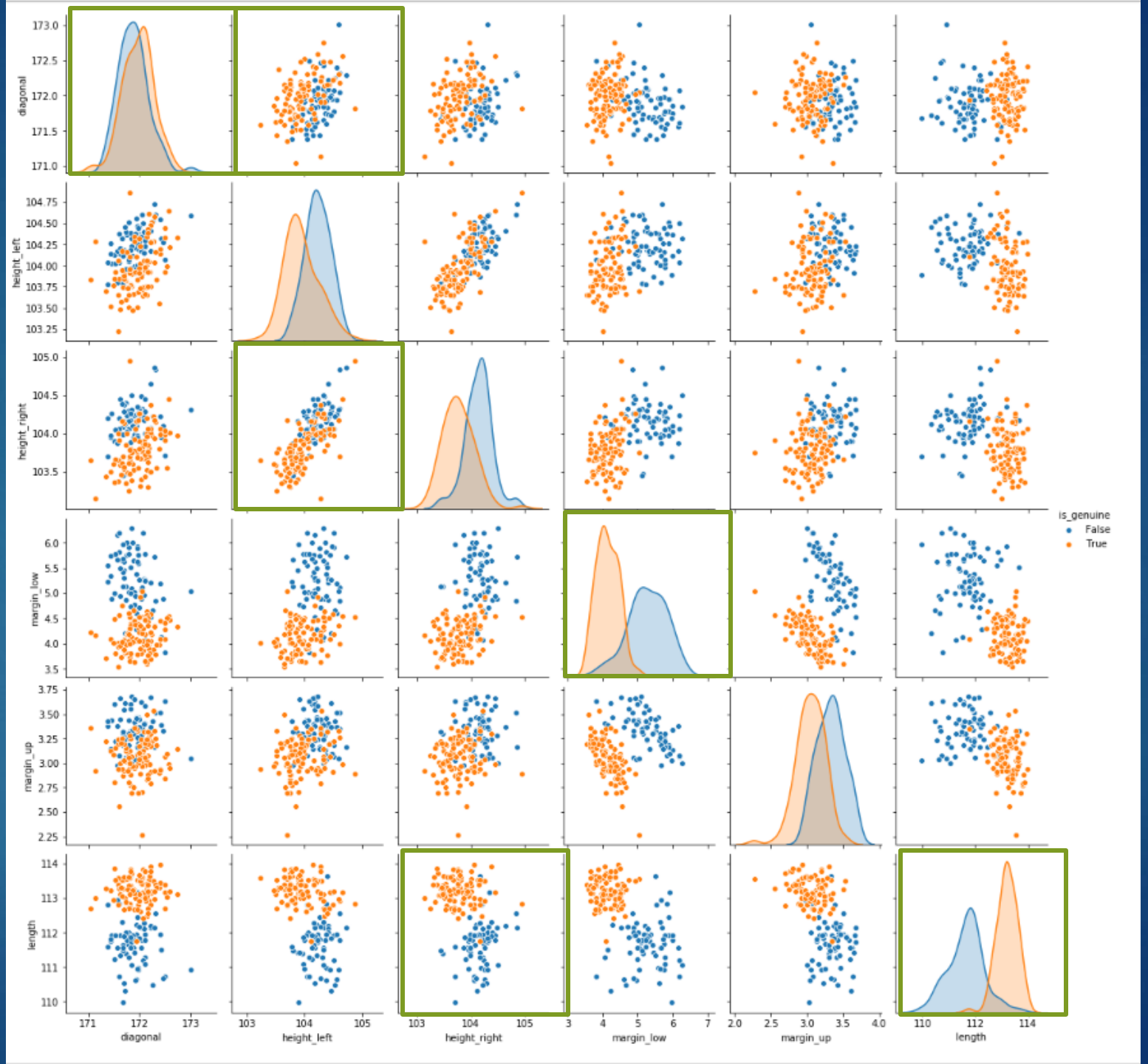
- Il n'y a aucune valeur manquante, atypique ou aberrante dans le dataset

Description des variables :

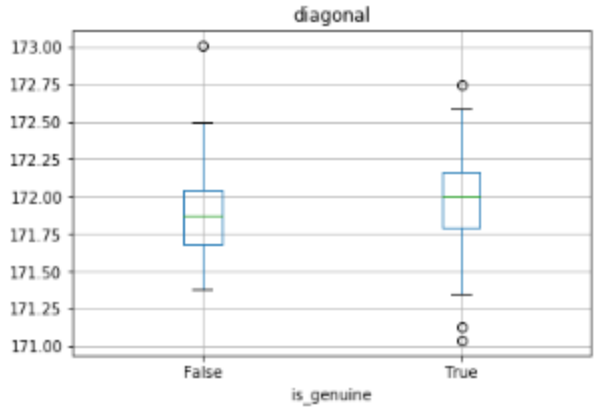
	diagonal	height_left	height_right	margin_low	margin_up	length
count	170.000000	170.000000	170.000000	170.000000	170.000000	170.000000
mean	171.940588	104.066353	103.928118	4.612118	3.170412	112.570412
std	0.305768	0.298185	0.330980	0.702103	0.236361	0.924448

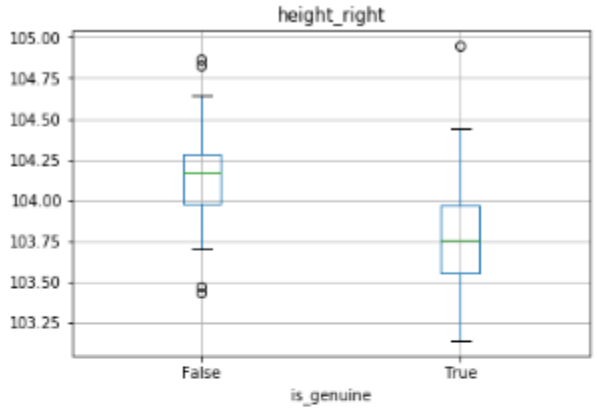
- Même si les variables ont la même unité, il faudra procéder à une mise à l'échelle pour certains des algorithmes

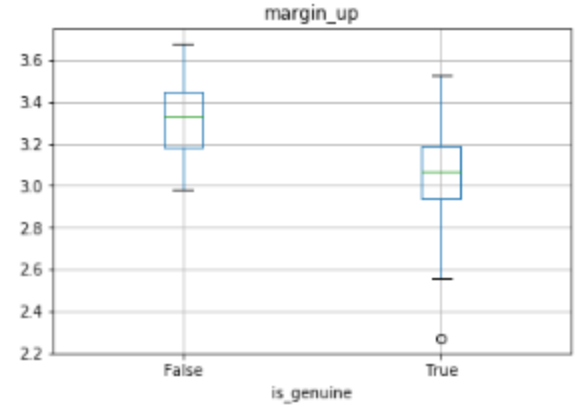
Analyse univariée et bivariée

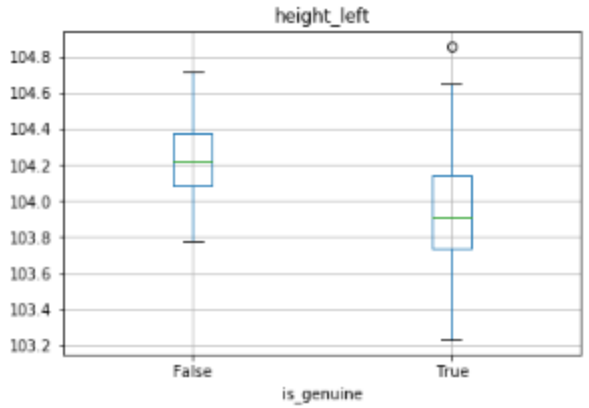


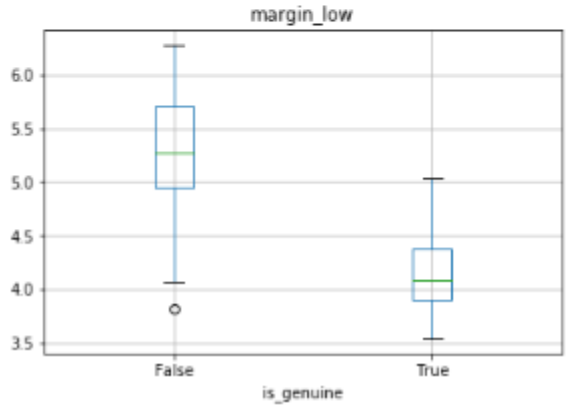
Boîtes à moustache

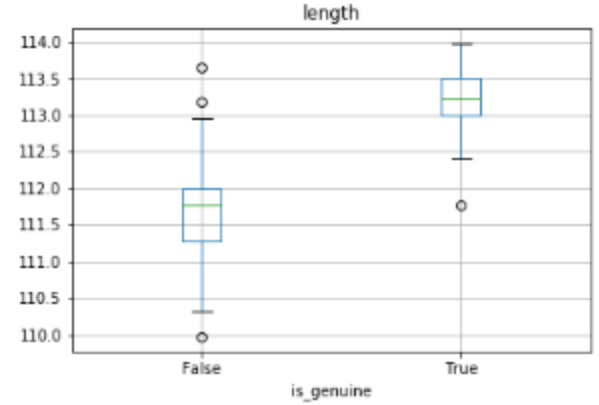












Analyse en composantes principales

Qu'est-ce que l'ACP ?

Objectifs :

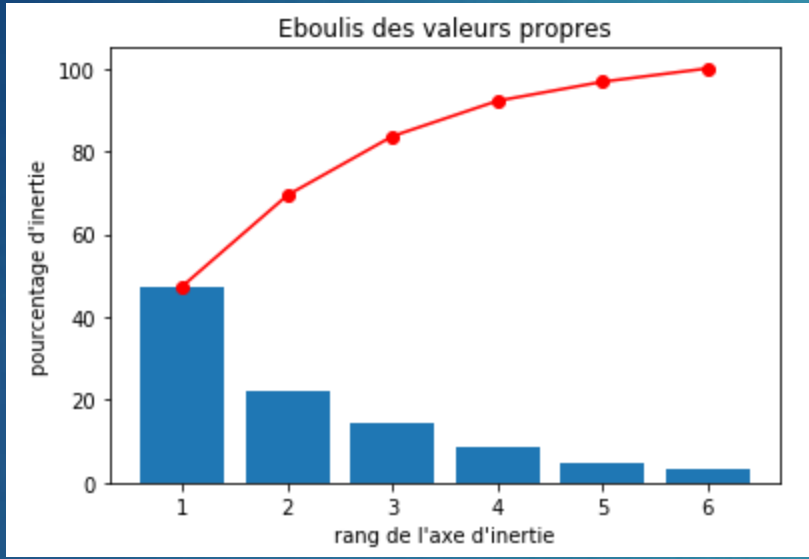
- Etudier la variabilité des individus
- Etudier les liaisons entre les variables

Principe :

Définir **de nouvelles variables qui sont des combinaisons linéaires de variables initiales** qui feront perdre le moins d'information possible

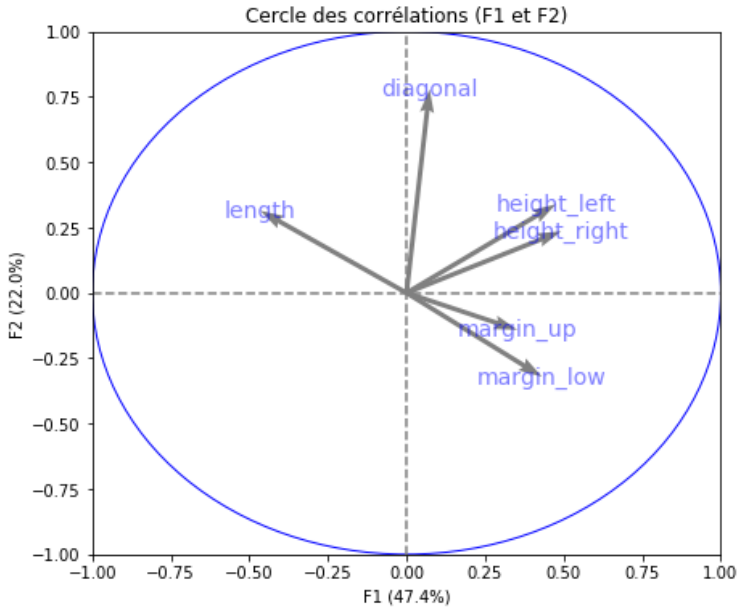
C'est un outil d'aide à l'interprétation. L'ACP permet de visualiser des espaces à dimensions > 3, en projetant un nuage de points sur des plans à deux dimensions.

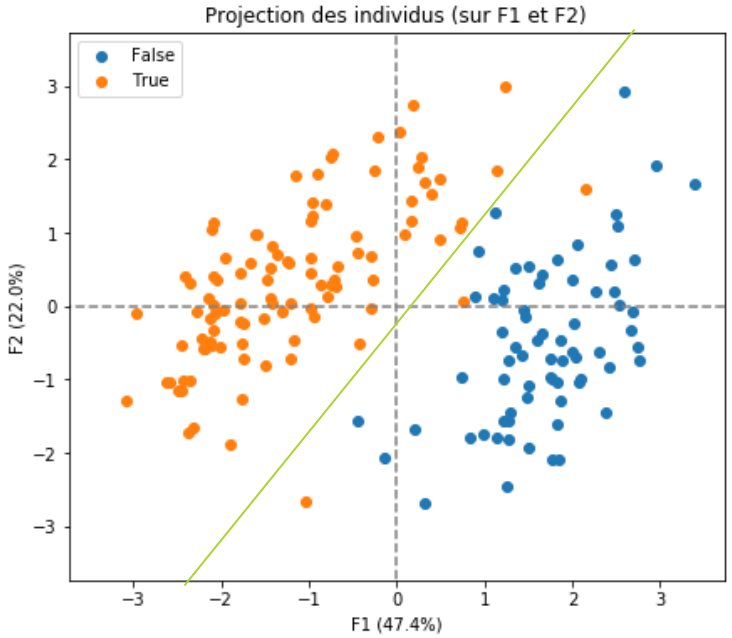
Eboulis des valeurs propres



- Le graphique nous indique que le pourcentage d'information expliquée est de :
  - Axe 1 : + de 40%
  - Axe 2 : + de 20%
- Le premier plan factoriel explique donc + de 60% de l'information
- Je choisis donc de limiter mon analyse à celui-ci

Analyse du premier plan factoriel

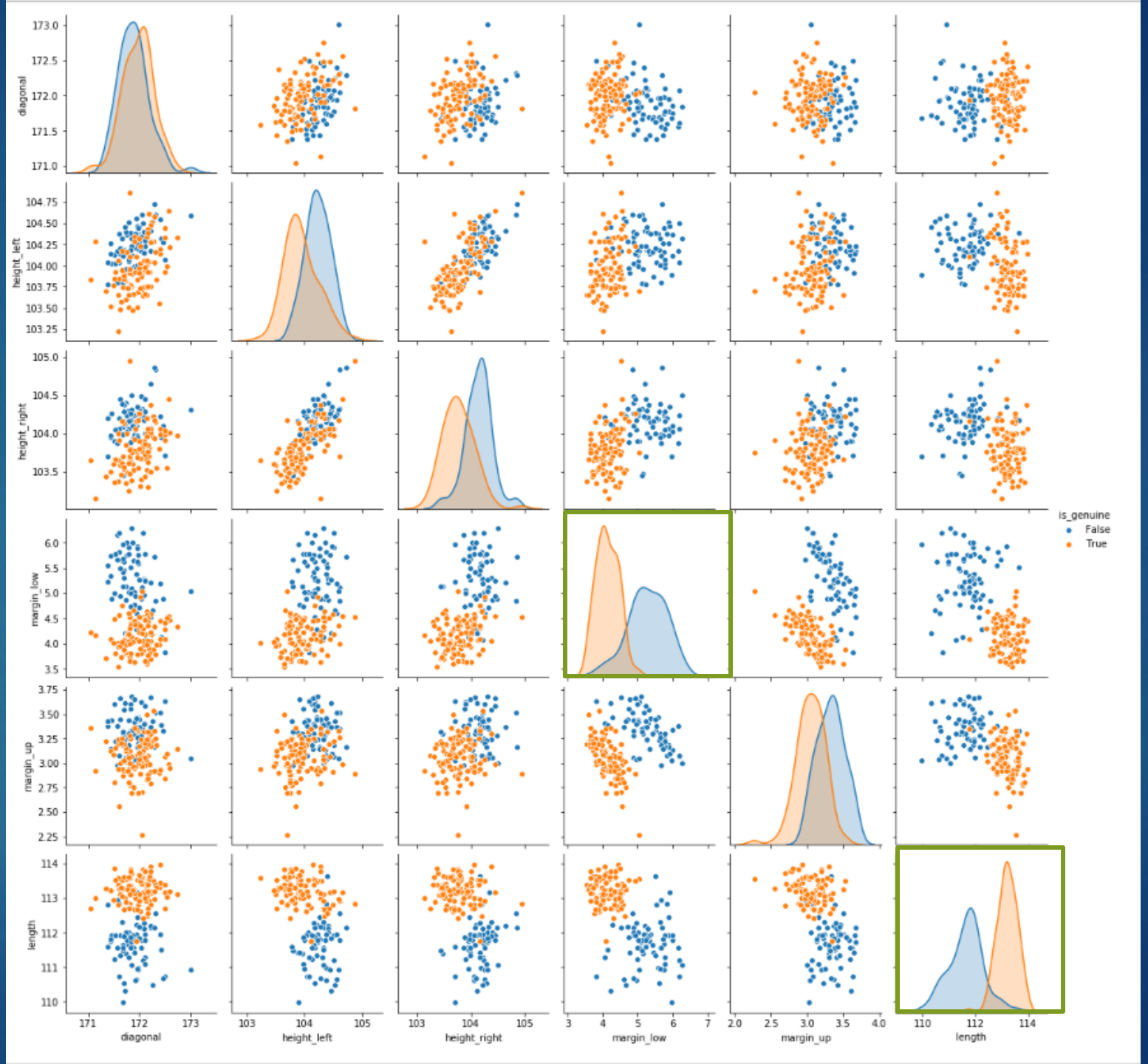




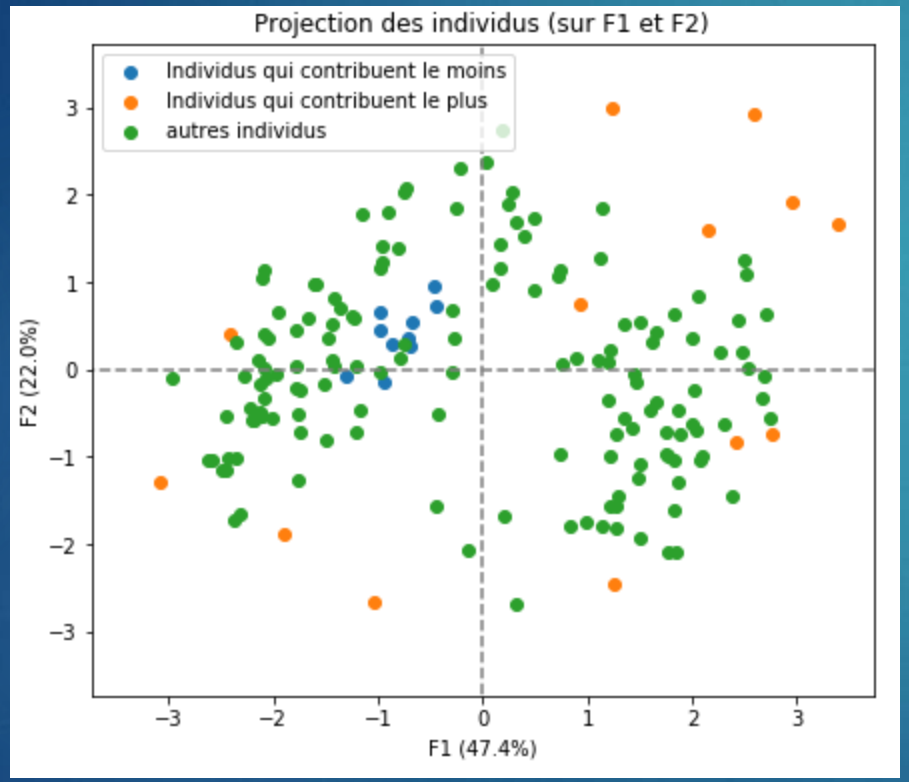
Variables	F1	F2
diagonal	0.073275	0.779639
height_left	0.475502	0.339250
height_right	0.491821	0.235543
margin_low	0.431027	-0.320537
margin_up	0.352540	-0.141120
length	-0.465373	0.314536

« margin_low » et « length » sont les variables qui séparent le mieux les clusters :

Rappel : Analyse univariée et bivariée



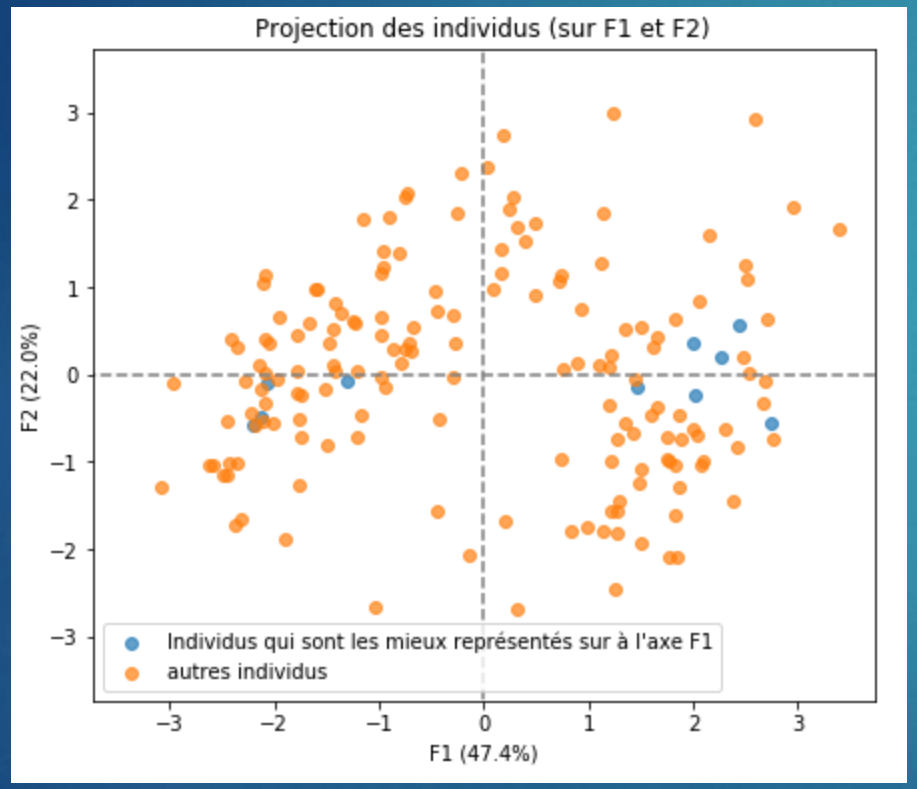
Contribution à l'inertie totale

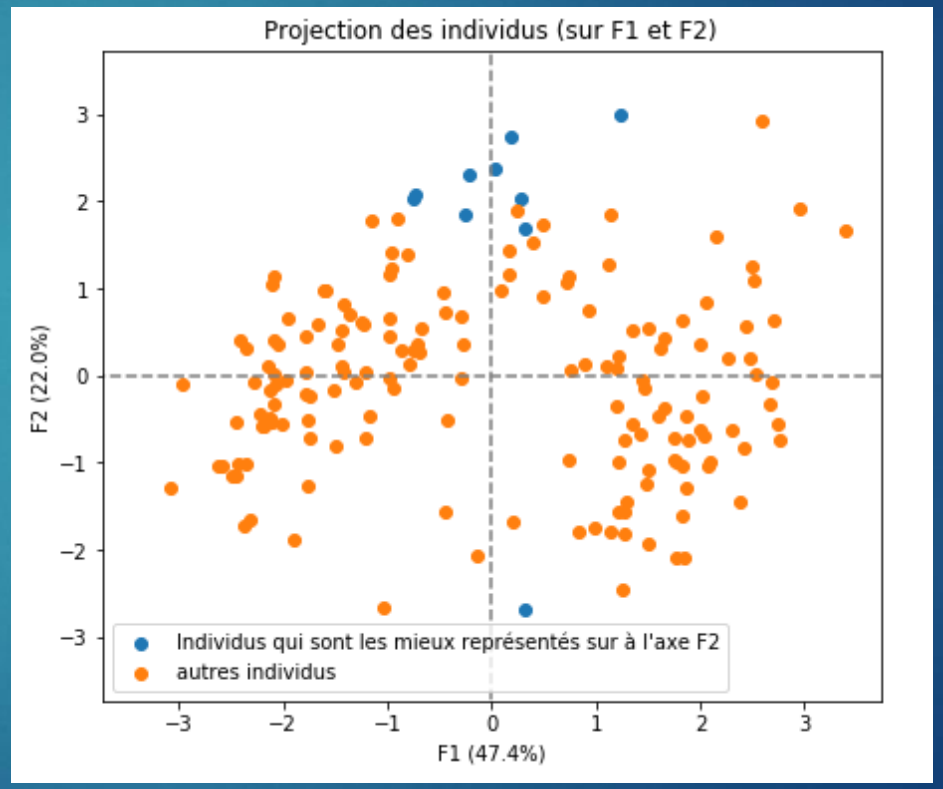


- Information nécessaire au calcul d'autres données
- Les individus qui contribuent le plus à l'inertie totale sont ceux qui se démarquent le plus des autres
- Formule :

$$d_i^2 = \sum_{j=1}^{p} z_{ij}^2$$

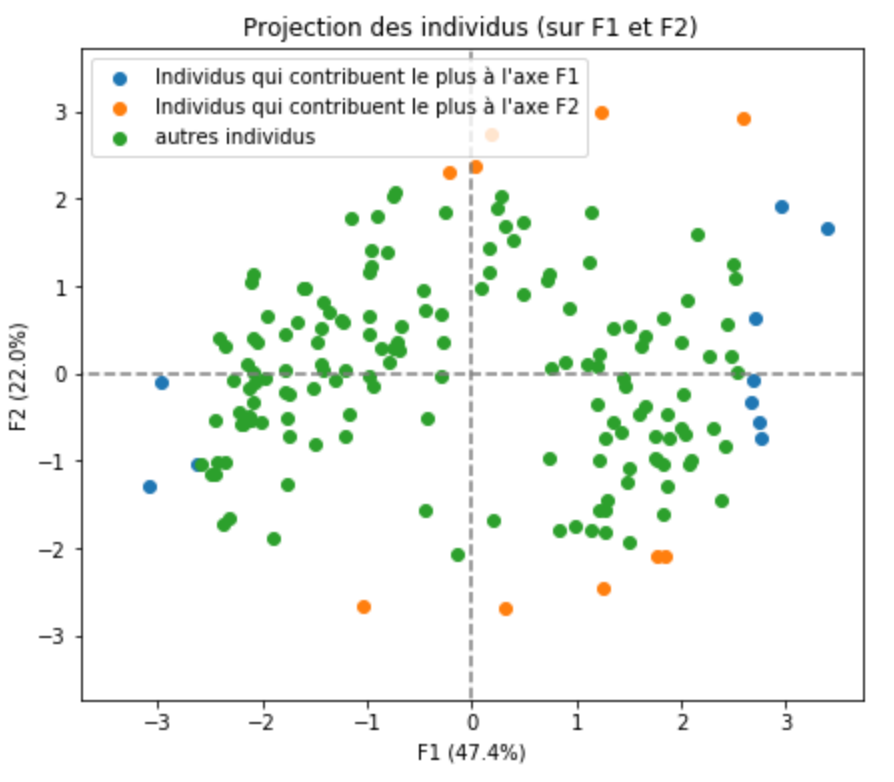
Qualité de représentation des individus par rapport aux axes





- Formule $cos^2_{ik} = \frac{F^2_{ik}}{d^2_i}$

Contribution des individus aux axes



- Les individus aux extrémités du nuage de points contribuent le plus aux axes
- Sur l'axe F1, le top 10 des contributions: 7 faux billets et 3 vrais billets.
- Formule :

$$CTR_{ik} = \frac{F_{ik}^2}{n \times \lambda_k}$$

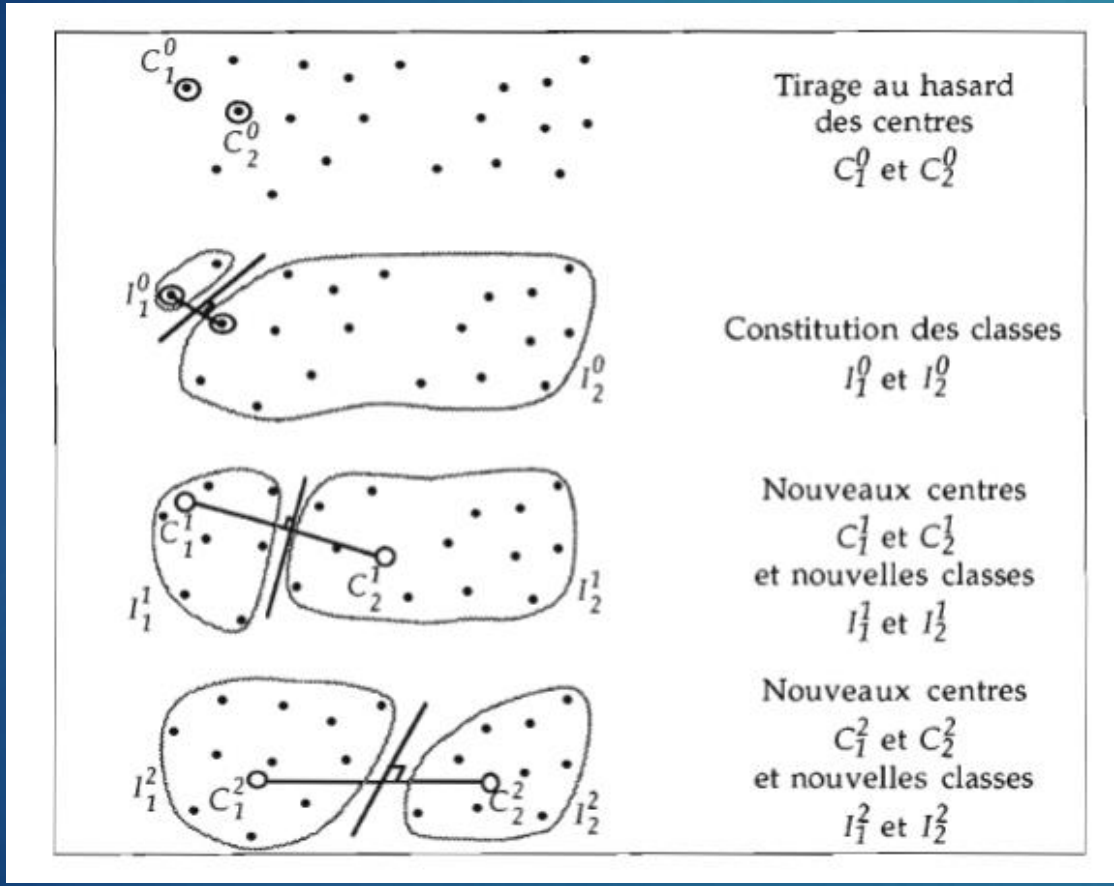
Détails des algorithmes

Comparatif des modèles

	Kmeans	**Régression logistique**
Méthode	Non-supervisée	Supervisée
Objectif	Analyse des clusters	Classifier des données
Avantage (non-exhaustif)	Fonctionne avec des données non-étiquetées	Approche probabiliste de la prédiction de classe
Inconvénient (non-exhaustif)	Donne des résultats inconsistants à chaque exécution de l'algorithme	Les données doivent pouvoir être séparer avec une régression logistique (le contre exemple du XOR)

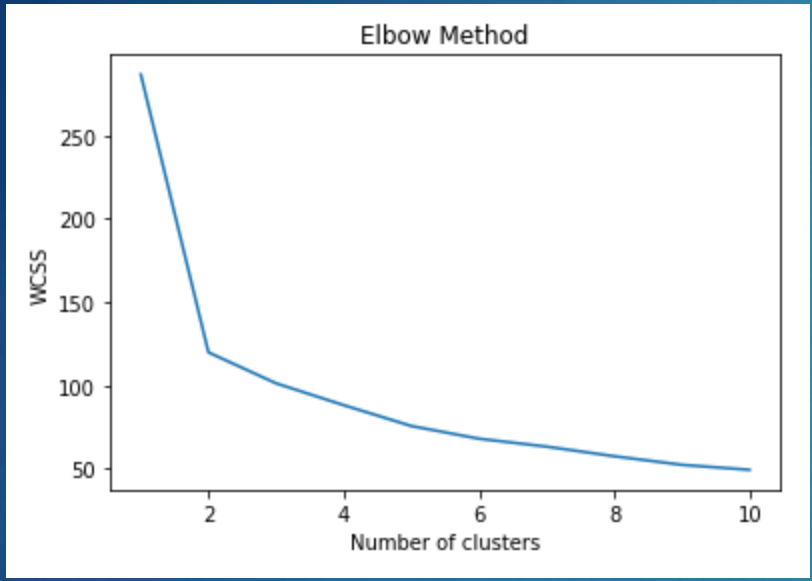
Algorithme de classification : Kmeans

Kmeans : schématisation



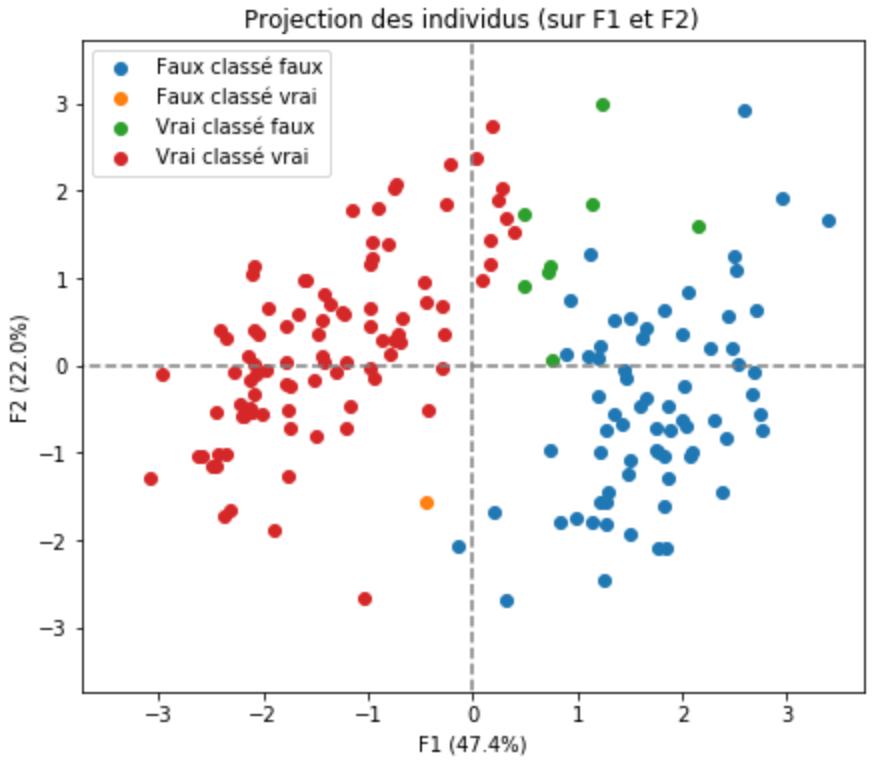
- Nécessite de déterminer le nombre de clusters au préalable
- Le processus minimise implicitement l'inertie intra-classe
- Répété jusqu'à convergence
- Sous Python, j'utiliserai la méthode Kmeans de la librairie sklearn avec le paramètre 'k-means++' qui est un algorithme d'initialisation qui permet d'améliorer la probabilité de trouver une solution optimale

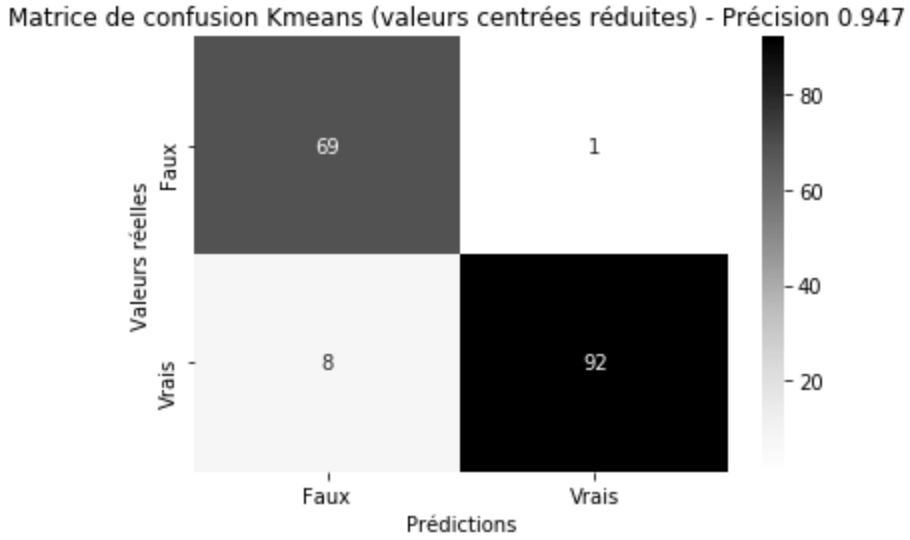
Méthode du coude



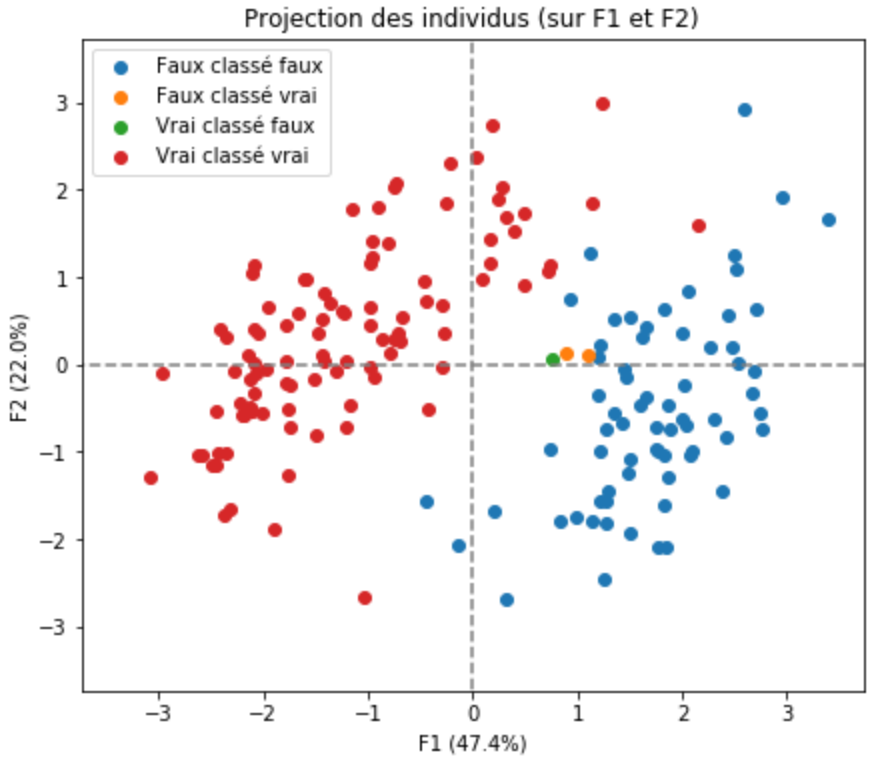
- Même si les paramètres de l'exercice nous suggère clairement qu'il y a deux clusters (les billets vrais et les billets faux), il est intéressant d'utiliser la méthode du coude pour savoir si cette information est corroboré.

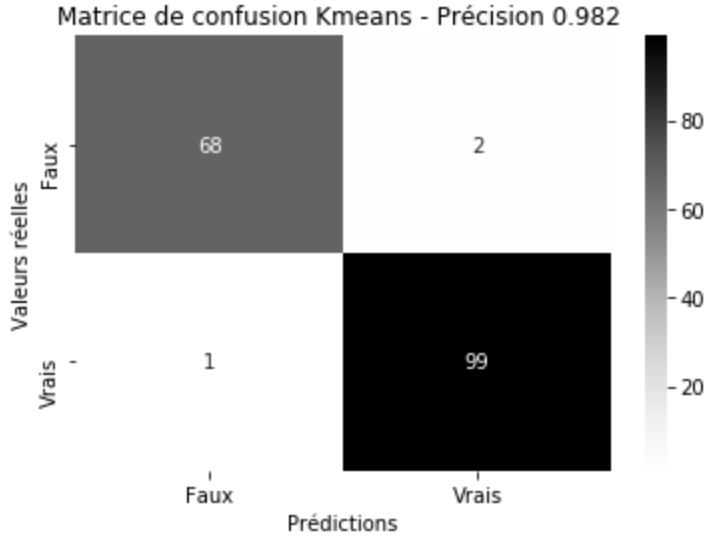
Kmeans valeurs centrées et réduites





Kmeans valeurs non centrées et réduites





Algorithme de classification : Régression logistique

Régression logistique (1)

1ère étape : entrainement du modèle

- Le modèle dispose de données avec des *labels* **y ∈ {0,1}**
- Calcul d'une variable **z**, qui correspond au vecteur théta (les coefficients attribué à chaque variable) multiplié par les variables

$$z = \theta^T x$$

- La variable **z** peut prendre n'importe quelle valeur, or la valeur de y étant 0 ou 1, la régression logistique va utiliser la fonction sigmoïde qui produit un résultat borné entre 0 et 1

$$g(z) = \frac{1}{1 + e^{-z}}$$

- On passe la variable **z** dans la fonction sigmoïde

$$h_\theta(x) = g(\theta^T x)$$

Régression logistique (2)

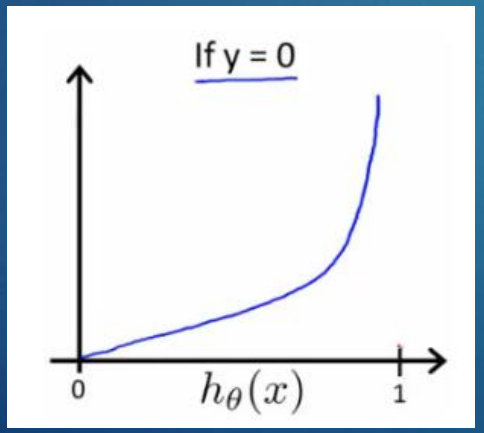
- Minimisation de la fonction coût à l'aide de la descente du gradient (ou d'un autre algorithme plus complexe)

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\ \log(h_\theta(x^{(i)})) + (1-y^{(i)})\ \log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Quand y =1

Quand y =0

Régularisation pour éviter l'overfitting



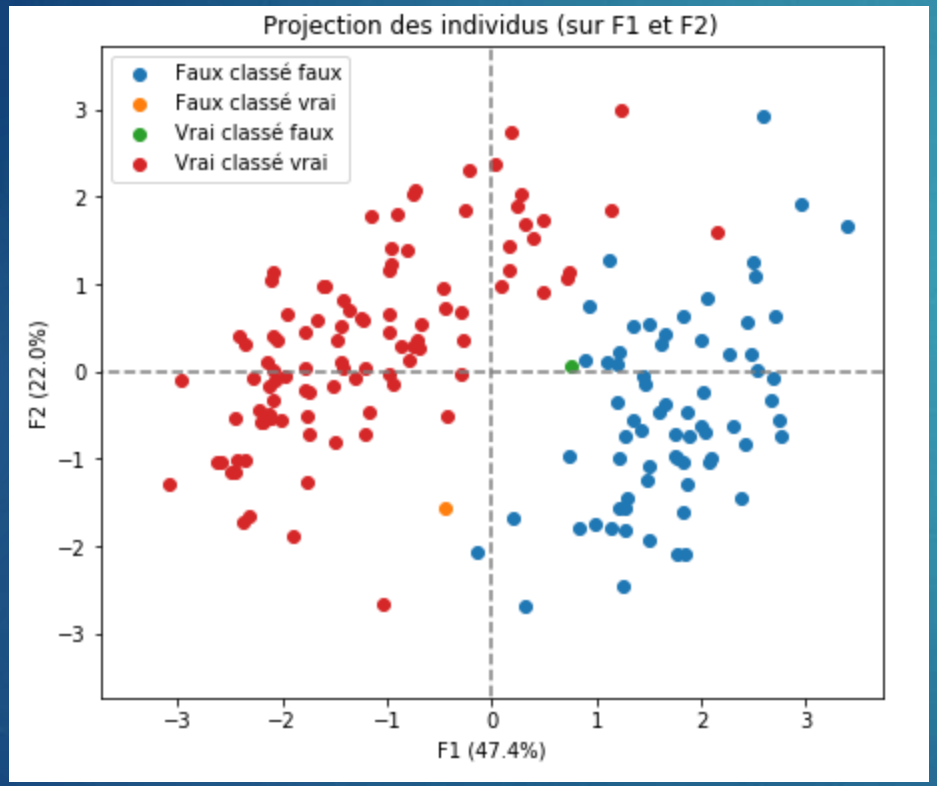
Régression logistique

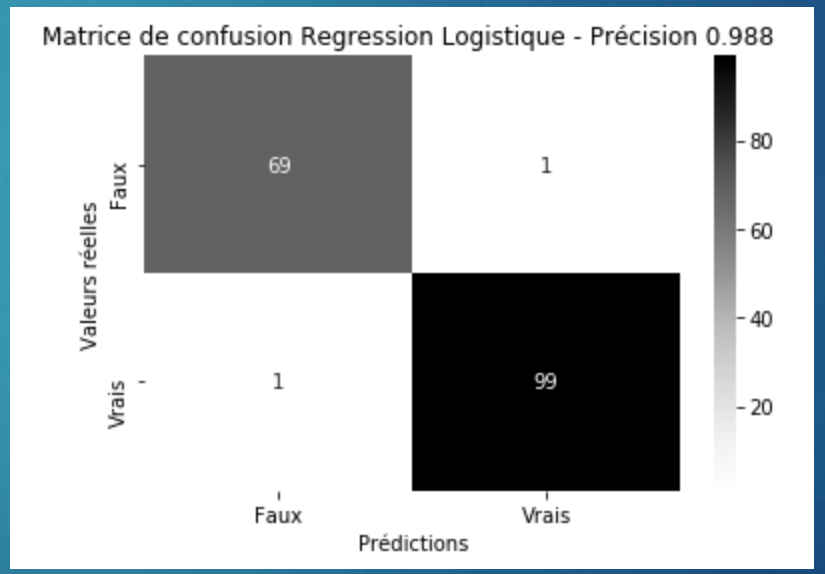
Afin de limiter l'overfitting, l'entraînement de la régression logistique a été effectué sur 75% du dataset.

2ème étape : Appliquer le modèle sur le dataset et apprécier le taux de précision

3ème étape : Appliquer le modèle sur d'autres jeux de données

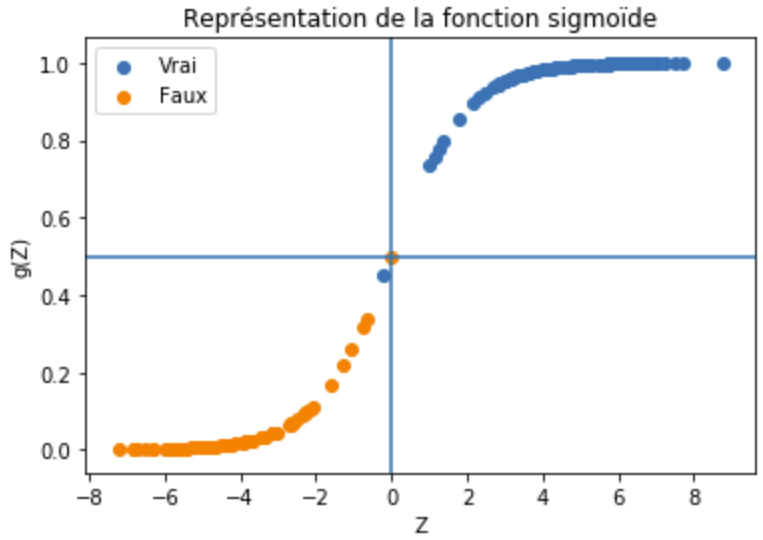
Visualisation de la régression logistique - *Données « scales »*





Fonction sigmoïde

Données « scales »

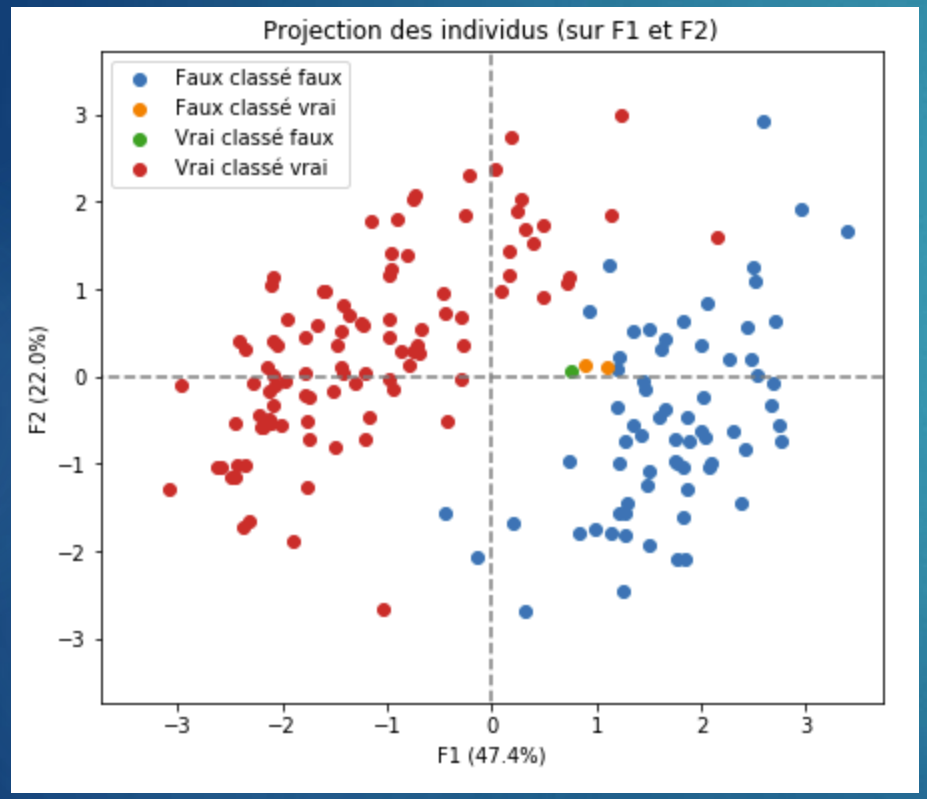


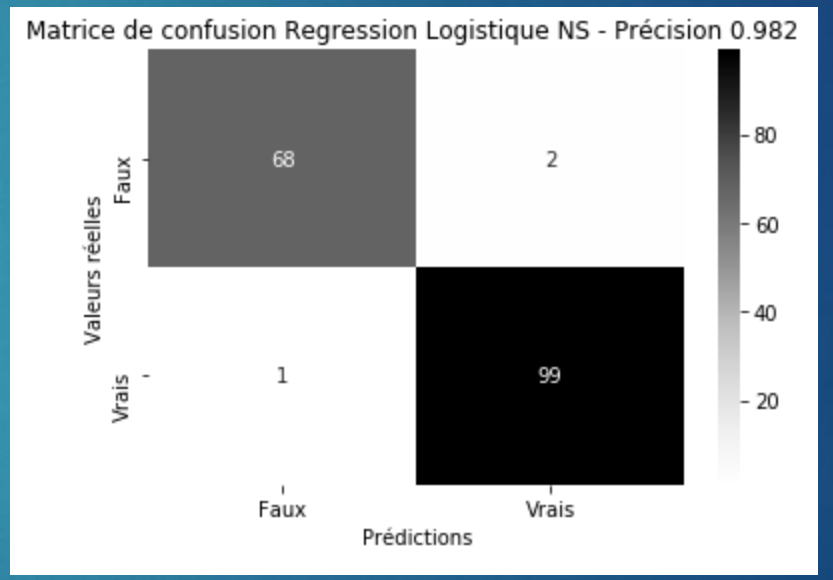
$$h_\theta(x) = g(\theta^T x)$$

$$z = \theta^T x$$

$$g(z) = \frac{1}{1+e^{-z}}$$

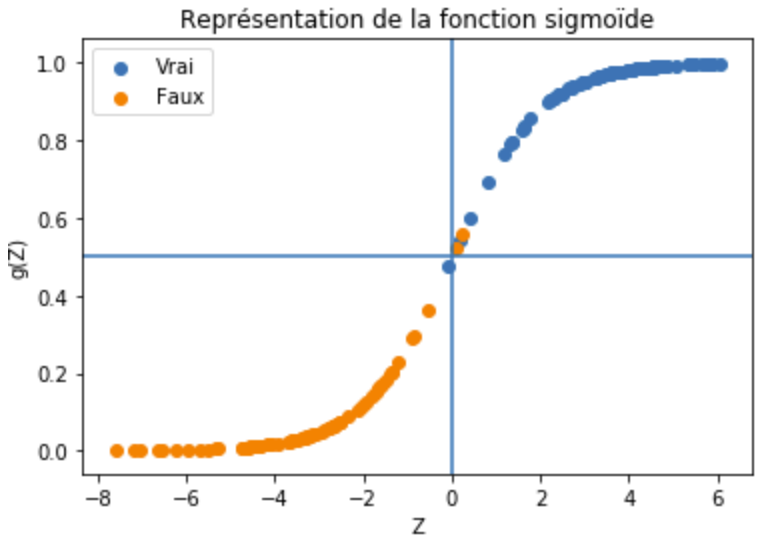
Visualisation de la régression logistique - *Données « non-scales »*





Fonction sigmoïde

Données « non-scales »



$$h_\theta(x) = g(\theta^T x)$$

$$z = \theta^T x$$

$$g(z) = \frac{1}{1+e^{-z}}$$

Conclusion

Comparatif des taux de précision

	Kmeans	Régression logistique
Données « scale »	0,947	0,988
Données « non-scale »	0,982	0,982

- Les modèles ont les mêmes performances sur les données brutes
- La régression logistique sur les données mise à l'échelle à le meilleur résultat avec un taux de précision qui s'élève à 98,8%