



Projet 6

Création Algorithme de Détection de Faux-billets

Introduction



Ce projet a été réalisé :

- Dans le cadre de la **formation DATA ANALYST** d'OpenClassRooms
- Afin de concevoir un Modèle/Programme de Détection de faux-billets sous la forme d'un algorithme de machine learning.
- À partir d'un jeu de données fourni par Openclassrooms, contenant des caractéristiques métriques de vrais et de faux billets.
- Avec l'aide/support de **M. César Clavé** (Mentor Openclassrooms)



Sommaire

- Introduction (remerciements)
- Data Import - Analyse descriptive
 - Excel & Jupyter Notebook : Data Cleaning
 - Analyse descriptive du jeu de données (univariée, bivariée)
 - Tests Statistiques
 - Cas de la variable « diagonale »
- ACP (Analyse en Composantes Principales)
 - Eboulis Valeurs Propres (Scree Plot) – Variance Expliquée
 - Visualisations sur 1^{er} Plan Factoriel
 - Choix de réduction du nombre de composantes : 4
- Analyse avec Algorithme de Classification
 - Outil K-Means - Comparaison des Clusters avec Datas Originelles
- Régression Logistique & Création Programme final
 - Application d'un modèle de régression logistique sur Données après ACP
 - Matrice de confusion & Courbe ROC
 - Crédit à la création du programme sur cette base.
- Vérification de billets
 - Test sur fichier « billets.csv » fourni et présentation des résultats
- Questions / Contact
- Annexes



Data Import - Analyse descriptive



Nettoyage données

- Excel (1^{er} Contrôle visuel rapide)

Screenshot of Microsoft Excel showing a filter dialog and a data table.

Filter Dialog (Left):

- 1 is_genuine
- Trier de A à Z
- Trier de Z à A
- Irer par couleur
- Effacer le filtre de « is_genuine »
- Filtrer par couleur
- Filtres textuels
- Rechercher
- (Sélectionner tout)
- False
- True

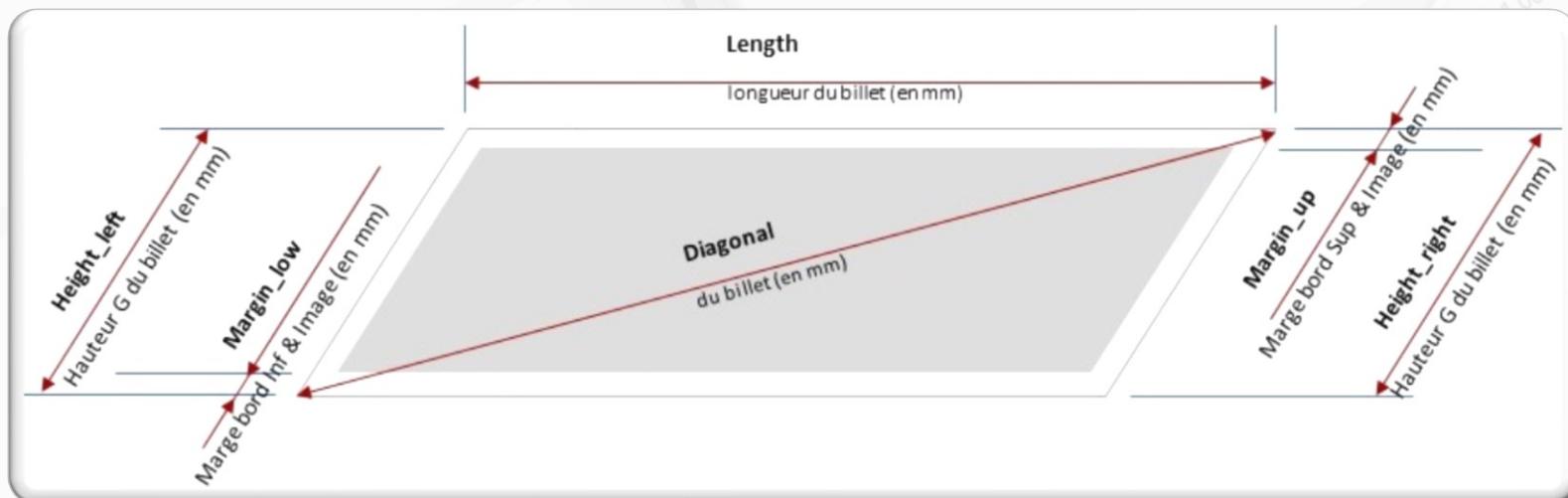
Excel Window (Right):

notes.xlsx - Excel

A101

True

	A	B	C	D	E	F	G	H
1	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length	
100	True		172.1	103.98	103.86	4.47	3.06	113
101	True		171.81	103.96	103.47	4	3	113.1
102	False		171.45	104.03	104.26	4.88	3.44	111.92
103	False		171.97	104.38	104.18	5.59	3.47	110.98
104	False		171.94	104.21	104.1	4.28	3.47	112.23
105	False		172.04	104.34	104.48	4.88	3.28	112.15



Data Import - Analyse descriptive



Nettoyage données

- Python pour le «cleaning» détaillé dans un notebook jupyter

```
df.info()
print(df.shape)
df.describe(include="all")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170 entries, 0 to 169
Data columns (total 7 columns):
is_genuine    170 non-null bool
diagonal      170 non-null float64
height_left   170 non-null float64
height_right  170 non-null float64
margin_low    170 non-null float64
margin_up     170 non-null float64
length        170 non-null float64
dtypes: bool(1), float64(6)
memory usage: 8.2 KB
(170, 7)
```

	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length
count	170	170.00	170.00	170.00	170.00	170.00	170.00
unique	2	nan	nan	nan	nan	nan	nan
top	True	nan	nan	nan	nan	nan	nan
freq	100	nan	nan	nan	nan	nan	nan
mean	NaN	171.94	104.07	103.93	4.61	3.17	112.57
std	NaN	0.31	0.30	0.33	0.70	0.24	0.92
min	NaN	171.04	103.23	103.14	3.54	2.27	109.97
25%	NaN	171.73	103.84	103.69	4.05	3.01	111.85
50%	NaN	171.94	104.06	103.95	4.45	3.17	112.84
75%	NaN	172.14	104.29	104.17	5.13	3.33	113.29
max	NaN	173.01	104.86	104.95	6.28	3.68	113.98

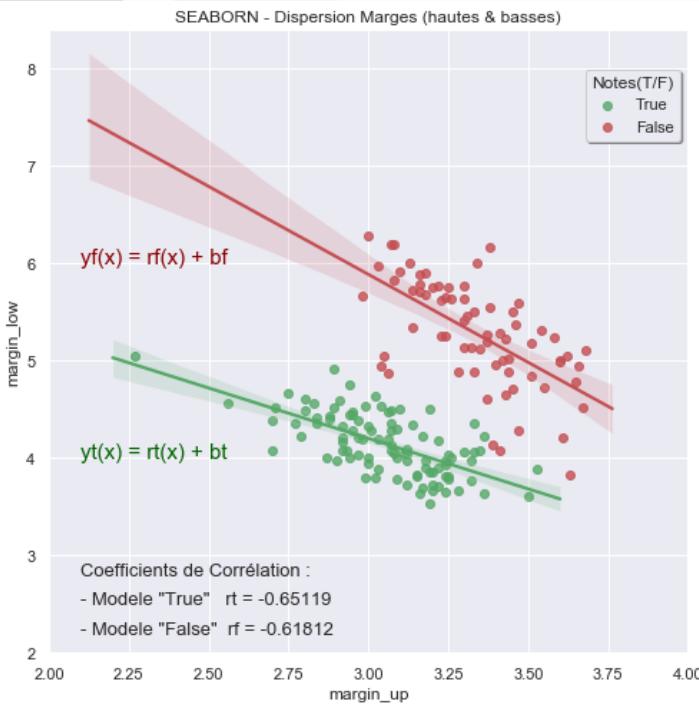
Pas de valeurs aberrantes constatées de prime abord :

- Hauteur : entre 103 & 105 mm
- Marges : entre 2,2 et 6,3 mm
- Longueur : entre 110 et 114mm
- Diagonale : entre 171 et 173 mm
- 70 billets flaggées «faux»
- 100 billets flaggés «vrais»

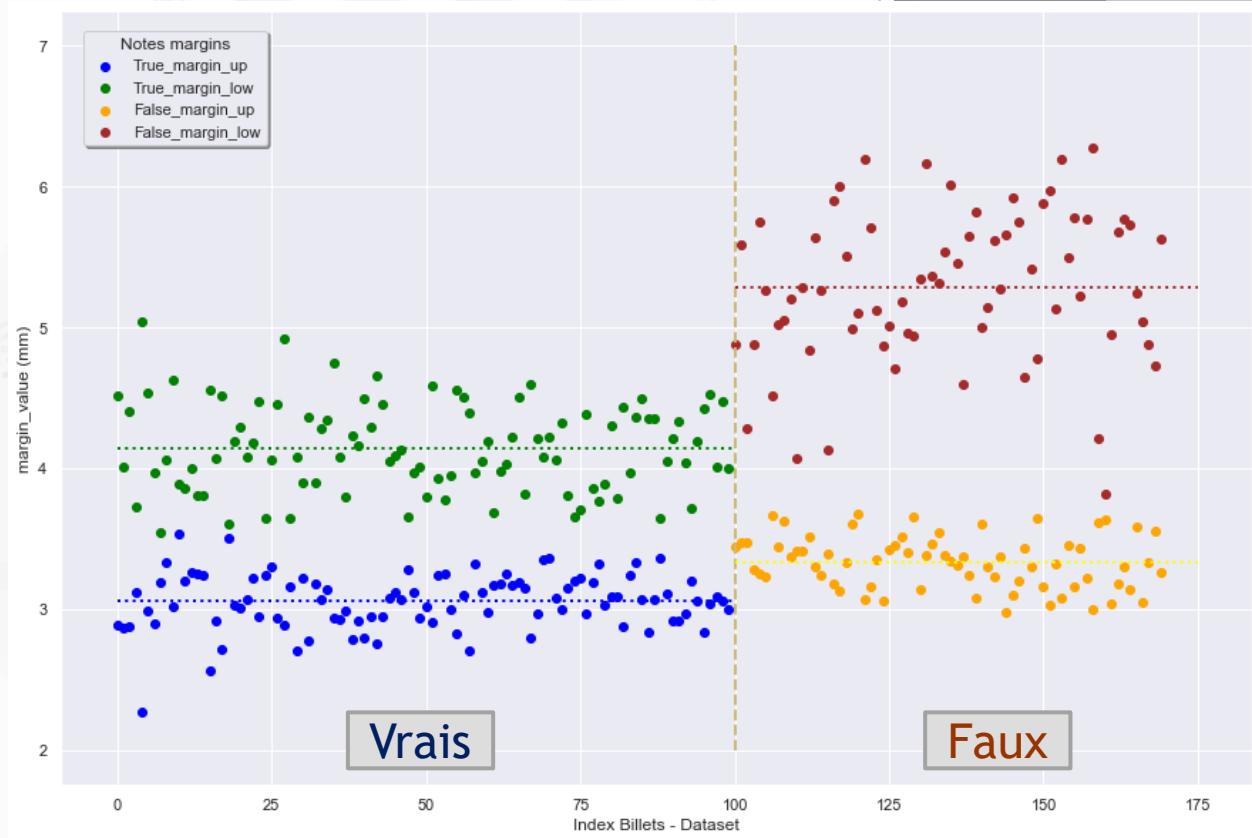
Data Import - Analyse descriptive

ANALYSE DES MARGES (hautes & basses) sur les Billets

Annexe



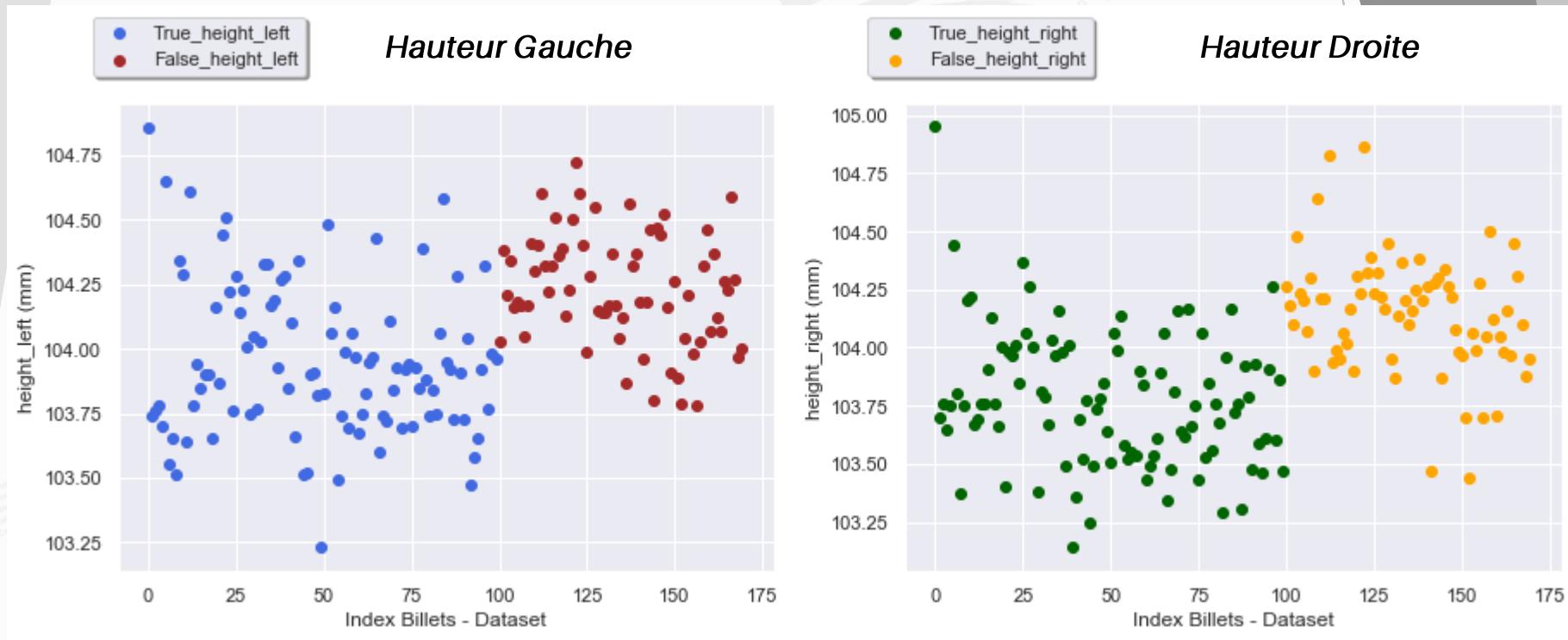
Corrélation linéaire négative entre marge Haute et marge basse. Le calcul du coefficient (rf & rt) nous indique qu'elle est serait légèrement plus significative (proche de -1) pour les VRAIS billets que pour les FAUX.



Vrais Billets → Index de 0 à 100
 Faux Billets → Index de 101 à 170
 Les valeurs des marges hautes & basses des Faux Billets sont respectivement supérieures « en Moyenne » à celles des Vrais Billets
Notamment, la variable « margin_low » (marge basse).

Data Import - Analyse descriptive

ANALYSE DES HAUTEURS sur les Billets



Ici difficile de conclure par une affirmation.

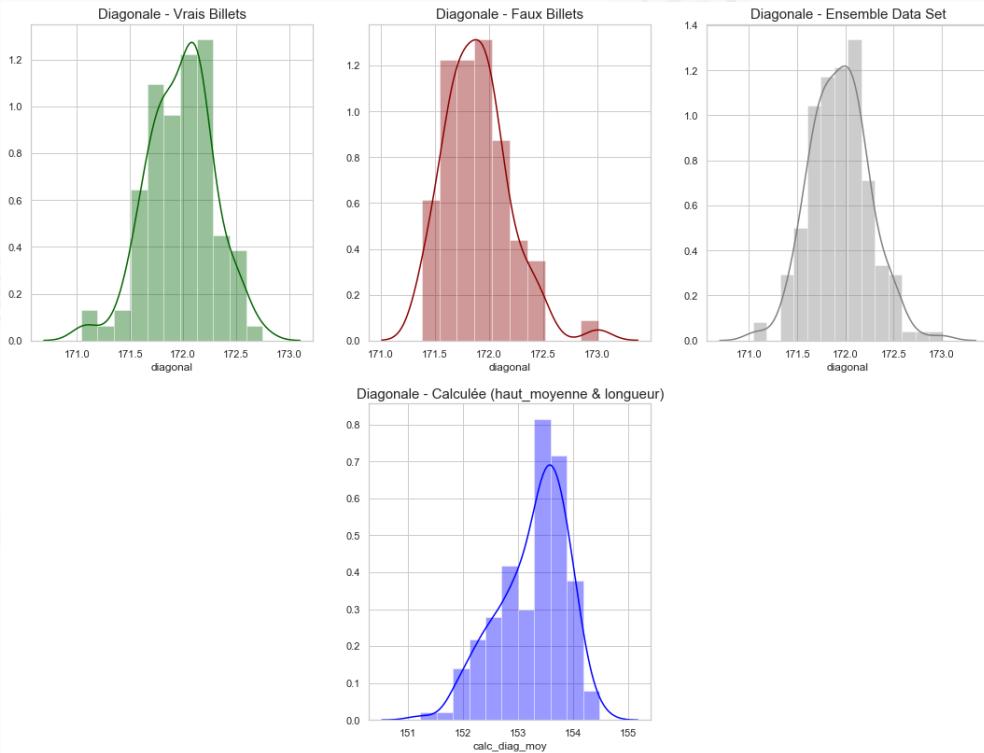
Les données sont réparties de façon assez homogène entre vrais et faux billets.

Ainsi la hauteur dans ce dataset n'est pas un critère évident de discrimination de billet.

Data Import - Analyse descriptive

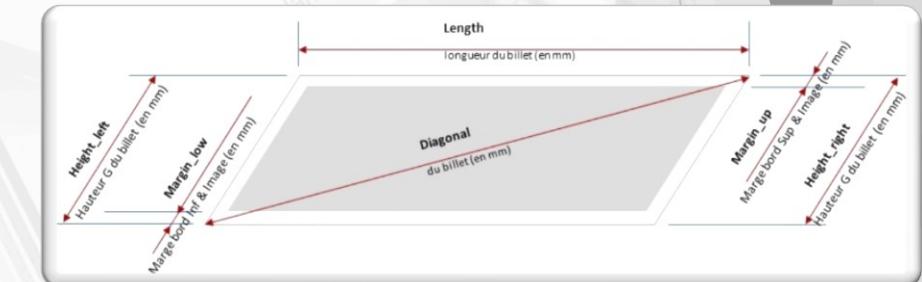
Contrôle de la valeur de la « diagonale »

is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length	calc_diag_moy	calc_diag_moy_avec_marg	
0	True	171.81	104.86	104.95	4.52	2.89	112.83	154.06	156.61
1	True	171.67	103.74	103.70	4.01	2.87	113.29	153.60	155.94
2	True	171.83	103.76	103.76	4.40	2.88	113.84	154.03	156.51
3	True	171.80	103.78	103.65	3.73	3.12	113.63	153.85	156.18
4	True	172.05	103.70	103.75	5.04	2.27	113.55	153.79	156.28



calc_diag_moy

$$\sqrt{\left(\frac{height_{left} + height_{right}}{2}\right)^2 + length^2}$$



Principe

- Construire un **nouveau** (système de représentation) du **tableau de données** (toutes quantitatives) qui permet synthétiser l'information sans trop de pertes d'informations.

Objectifs principaux :

- Analyse des relations/proximités entre les individus/lignes du dataframe.
- Analyse des relations entre les colonnes/variables de l'échantillon

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	171.67	103.74	103.70	4.01	2.87 113.29
2	True	171.83	103.76	103.76	4.40	2.88 113.84
3	True	171.80	103.78	103.65	3.73	3.12 113.63
4	True	172.05	103.70	103.75	5.04	2.27 113.55

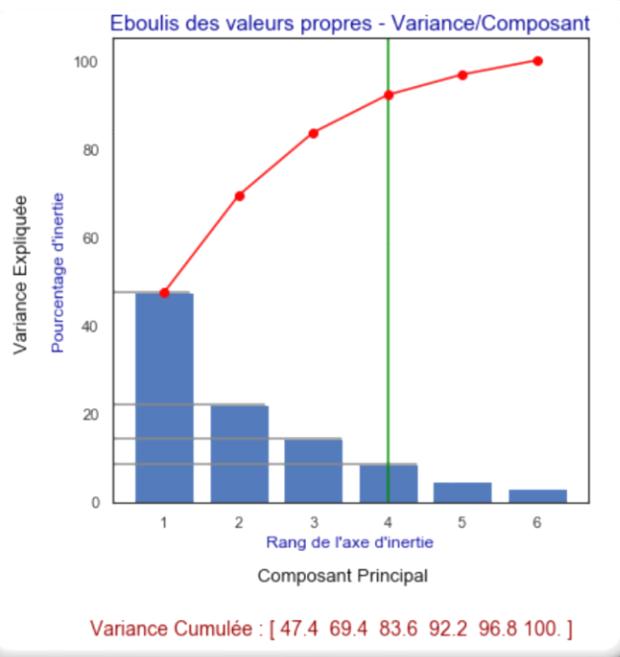
Prenons les 6 variables qualitatives comme nombre de composantes dans le calcul de l'ACP (slides suivants)

ACP (Analyse en Composantes Principales)

CALCUL DES COMPOSANTES PRINCIPALES (ACP) sur colonnes/variables

Nouveau Tableau des variables synthétiques (après centrage/réduction)

	F1	F2	F3	F4	F5	F6
0	2.15	1.60	1.79	2.43	0.70	-1.27
1	-2.11	-0.53	0.54	0.34	0.07	-0.54
2	-1.97	-0.05	0.86	0.37	-0.42	0.08
3	-2.06	-0.09	-0.53	0.52	-0.03	-0.04
4	-2.40	0.41	3.32	-0.84	-0.42	-0.45



sklearn.decomposition.PCA

```
# choix du nombre de composantes à calculer
n_comp = 6

# Import de l'échantillon & Selection des colonnes à prendre en compte dans l'ACP
data_pca = df.loc[:, df.columns != 'is_genuine']

# préparation des données pour l'ACP
# Il est fréquent de remplacer les valeurs inconnues par la moyenne de la variable
data_pca = data_pca.fillna(data_pca.mean())
X = data_pca.values

# Centrage et Réduction
std_scale = preprocessing.StandardScaler().fit(X)
X_scaled = std_scale.transform(X)

# Calcul des composantes principales
pca = decomposition.PCA(n_components=n_comp)
pca.fit(X_scaled)
X_projected = pca.fit_transform(X_scaled)

# Constitution d'un Dataframe "Resultat" pour l'ACP
dfacp = pd.DataFrame(X_projected, index=data_pca.index,
                      columns=["F"+str(i+1) for i in range(n_comp)])
dfacp.head()

# Stockage des "Libellés" pour utilisation ultérieure (Cercle Corrélation, Plan Fact.)
names = data_pca.index
features = data_pca.columns
```

Eboulis Valeurs Propres (scree plot)

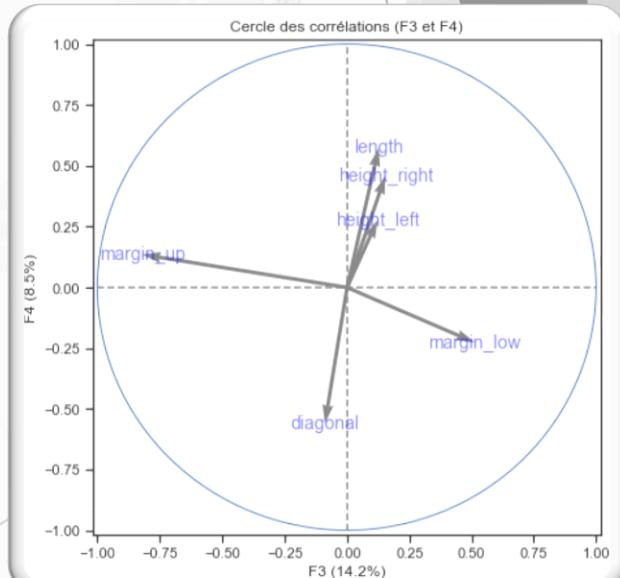
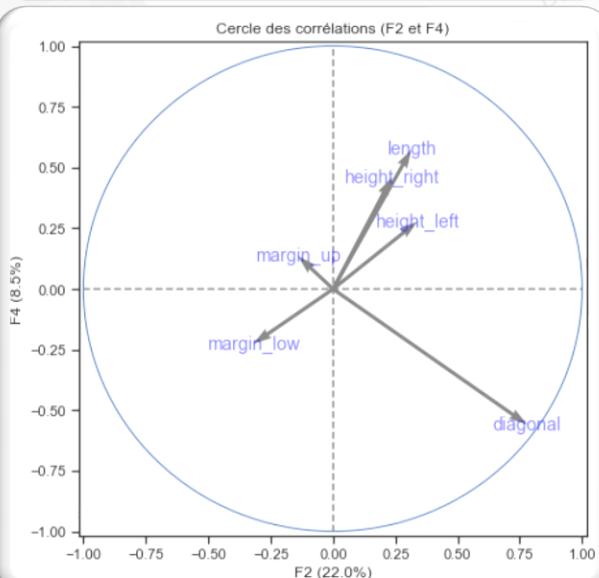
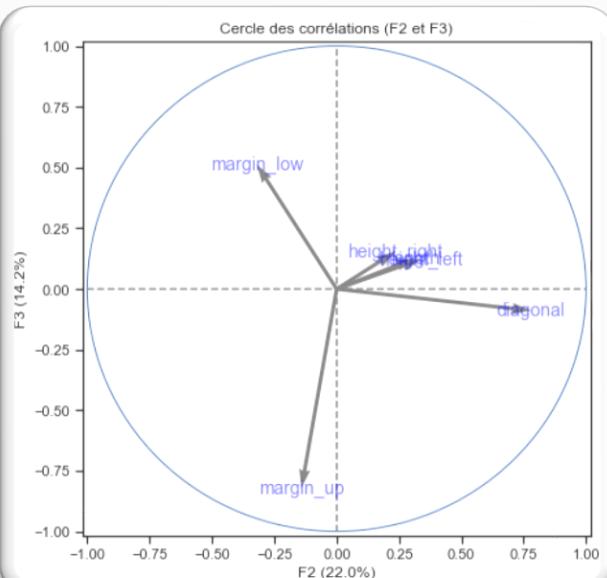
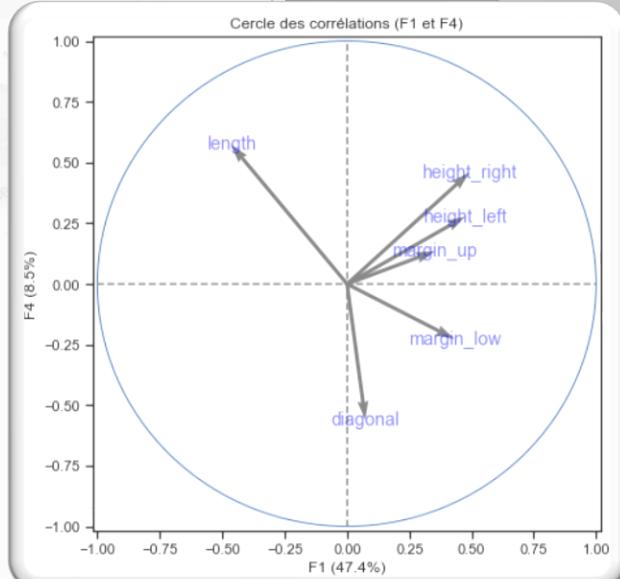
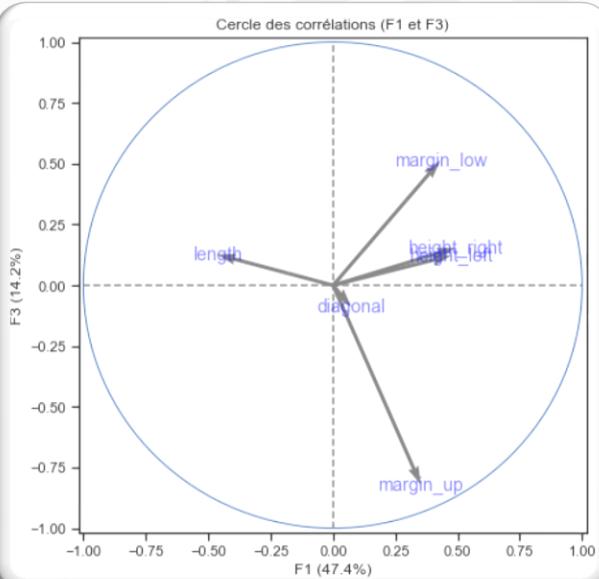
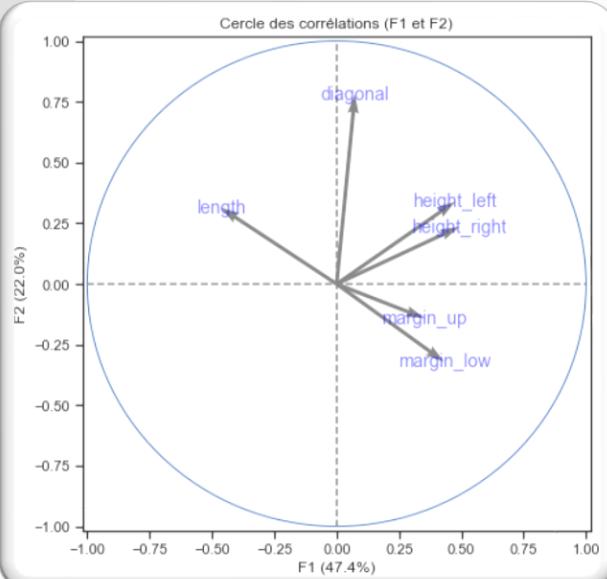
Fixons un seuil de 90% à atteindre pour expliquer un maximum de variance

Les 4 premiers composants (92.2%)

Sont suffisants pour expliquer plus de 90% de la variance cumulée (notre seuil de départ)

ACP (Analyse en Composantes Principales)

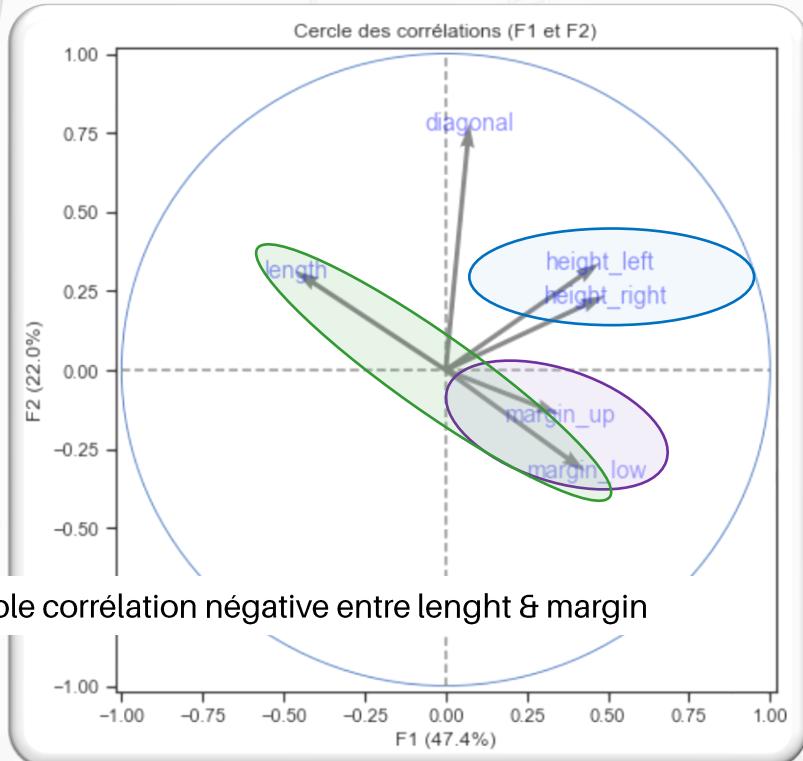
Visualisations des Variables sur les CERCLE DES CORRELATIONS



ACP (Analyse en Composantes Principales)

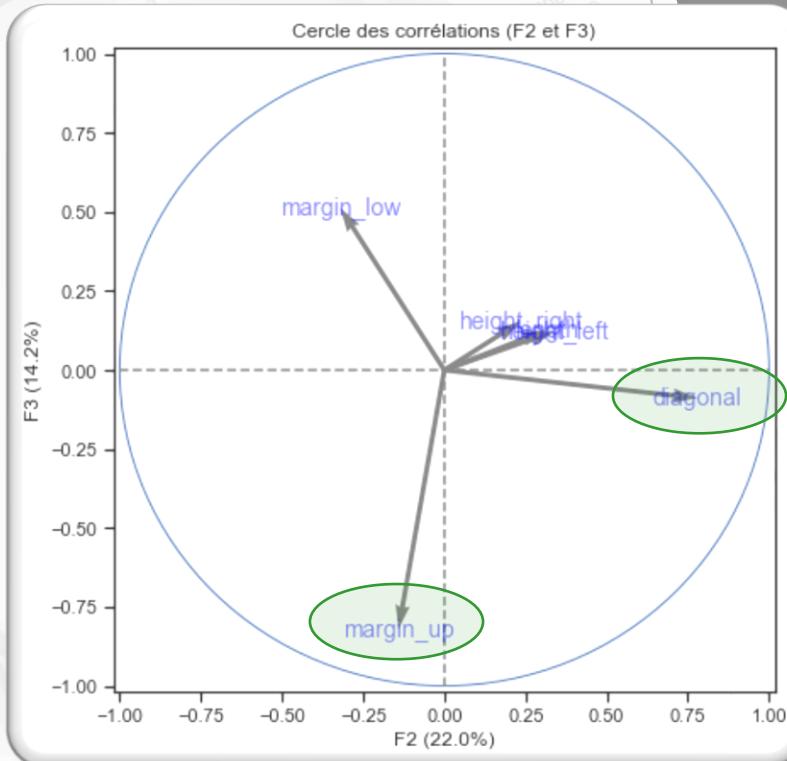
Visualisations des Variables sur les CERCLES DES CORRELATIONS

Sur le 1^{er} Plan Factoriel (F1 – F2) , Corrélation positive entre deux groupes de variables



Sur un 2^{ème} Plan Factotiel (F2 – F3)

La diagonale est bien représentée sur l'axe F2
La marge haute est bien représentée sur l'axe F3



Interprétation des variables synthétiques

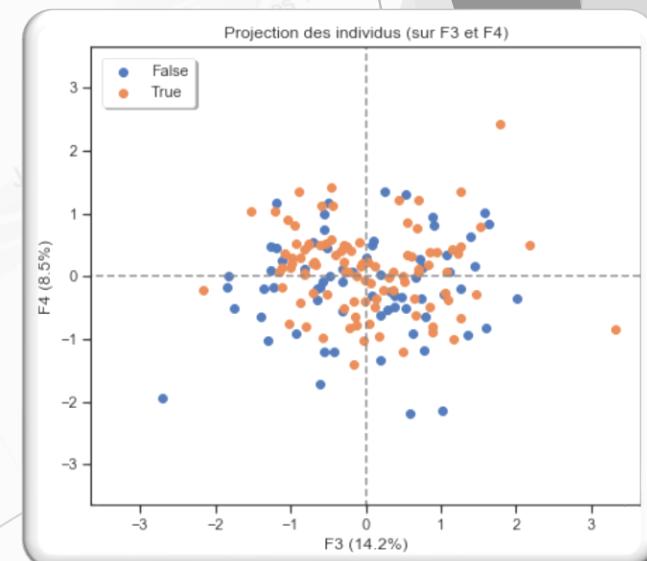
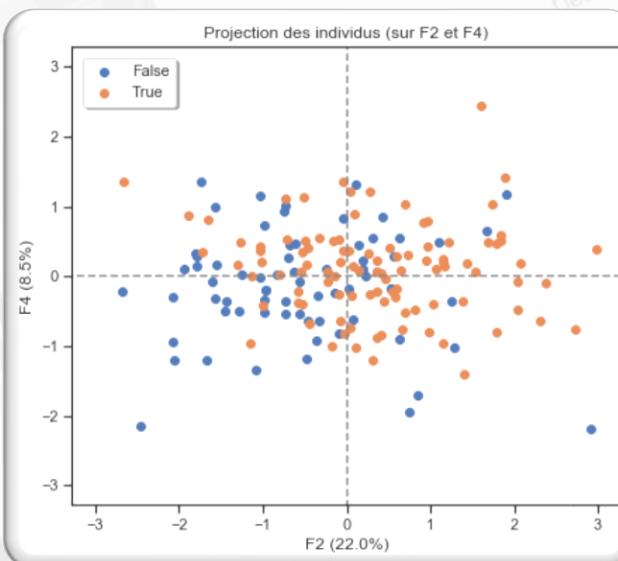
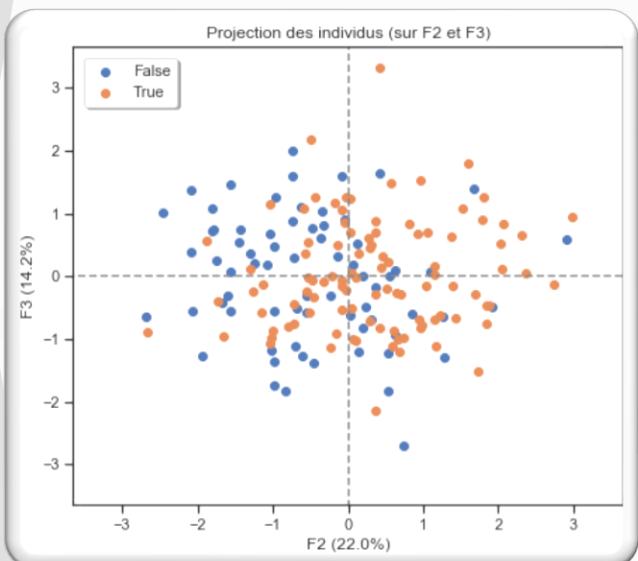
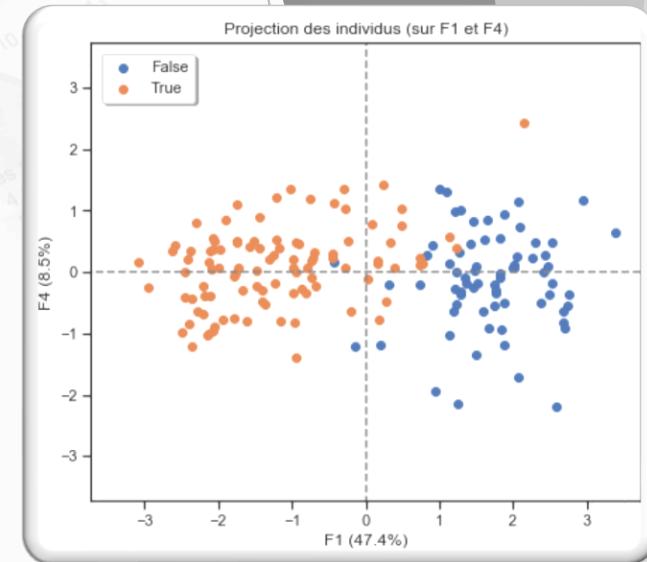
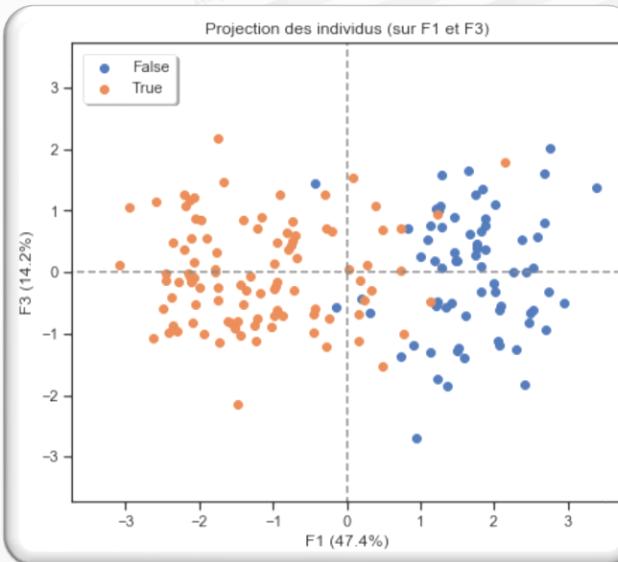
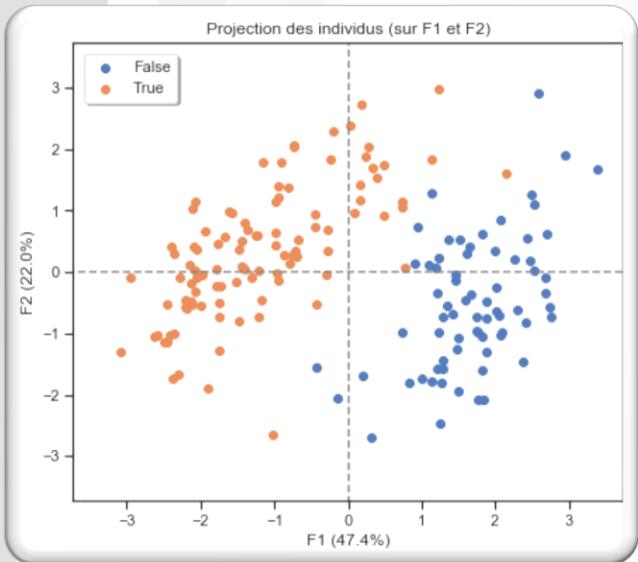
F1 – Représente les Hauteurs(left & right), et les marge(low & up). Dans une moindre mesure pour la (marge_up).

F2 – Représente la Diagonale

F3 – Traduit bien la Marge (up)

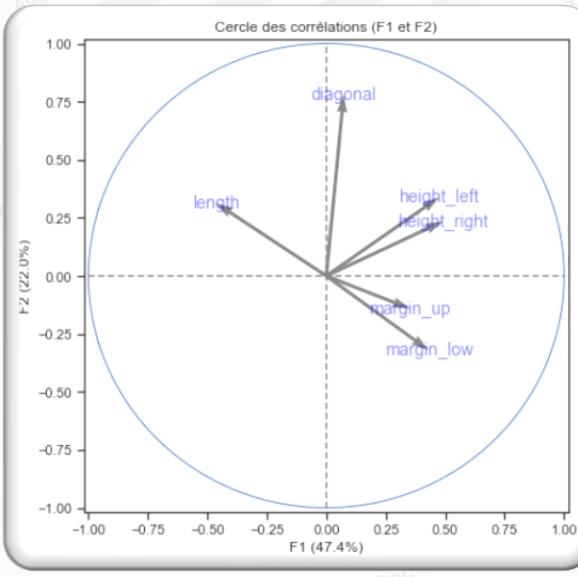
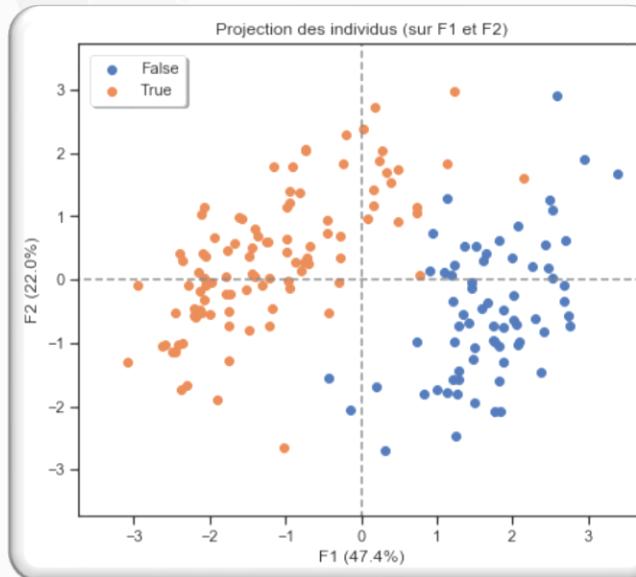
ACP (Analyse en Composantes Principales)

Projection des Billets (vrais/faux) sur les Plans Factoriel obtenus par ACP (F1-F2-F3-F4)



ACP (Analyse en Composantes Principales)

Interprétation de l'ACP



Axe F1

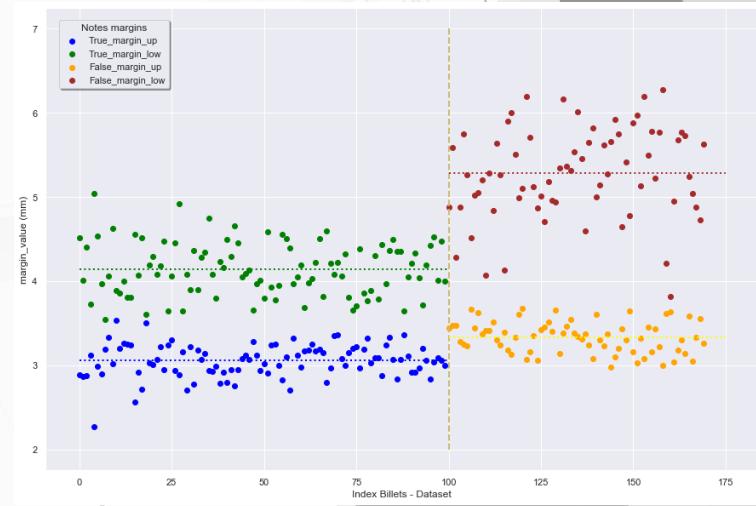
Question : qu'est-ce qui différencie les billets qui ont grande abscisse de ceux qui en ont une petite ?

Se déplacer le long des abscisses dans le sens croissant, c'est un peu se déplacer vers des billets "faux" avec des valeurs élevées en Hauteur et en Marges. Et se déplacer vers F1 dans le sens décroissant nous rapproche des billets « vrais » (avec des valeurs faibles en longueur.)

Axe F2

Question : qu'est-ce qui différencie les billets qui ont une abscisse grande de ceux qui en ont une petite ?

Se déplacer le long des ordonnées dans le sens croissant, c'est un peu se déplacer vers des billets avec une grande valeur en diagonale (mm) mais sans pour autant en savoir plus sur la nature du billets (vrai/faux)



Analyse - Algorithme de Classification

Choix : *Algorithme de Classification non supervisé → K-means*

Il permettra de regrouper en « k » clusters (ici 2) les observations de notre DataSet de billets.

Ceci afin de répondre à la question ultérieure :

Est-ce que les groupes k-means correspondent aux Groupes "Vrai/Faux" billets ?

Dataset pour K-means (issues de l'ACP)

k0 = 76 / k1 = 94

	F1	F2	F3	F4	is_genuine	clusterk
0	2.15	1.60	1.79	2.43	1	0
1	-2.11	-0.53	0.54	0.34	1	1
2	-1.97	-0.05	0.86	0.37	1	1
3	-2.06	-0.09	-0.53	0.52	1	1
4	-2.40	0.41	3.32	-0.84	1	1

Résultat Classification k_means

- 94 Vrais billets (« 1 »)
- 76 Faux billets (« 0 »)

Choix Paramètres K-means:

- n_clusters = 2
- n_init = 10
- max_iter = 300
- init = k-means++



sklearn.cluster.KMeans

```
# k-means sur Les données centrées et réduites issues de L'ACP
kmeans = cluster.KMeans(n_clusters=2, init='k-means++', n_init=10, max_iter=300, tol=0.0001)
kmeans.fit(dfk)
```

```
# Permet d'éviter l'erreur suivante :
# SettingWithCopyWarning in Pandas:A value is trying to be set on a copy of a slice from a DataFrame
pd.options.mode.chained_assignment = None # default='warn'
```

```
dfk['is_genuine'] = df.is_genuine
dfk['is_genuine'] = dfk['is_genuine'].map({True: 1, False: 0})
dfk['is_genuine'] = dfk['is_genuine'].astype(int)
dfk['clusterk'] = kmeans.labels_
```

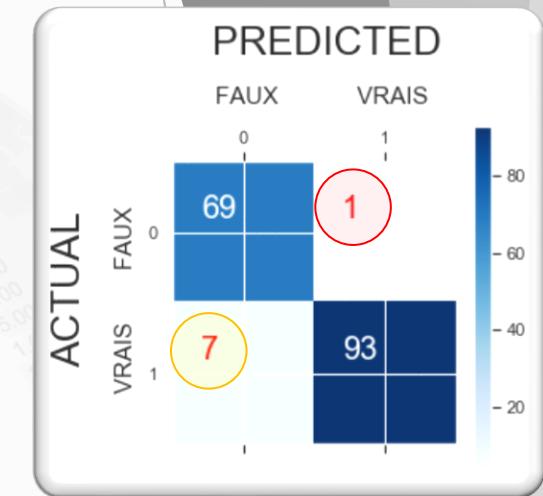
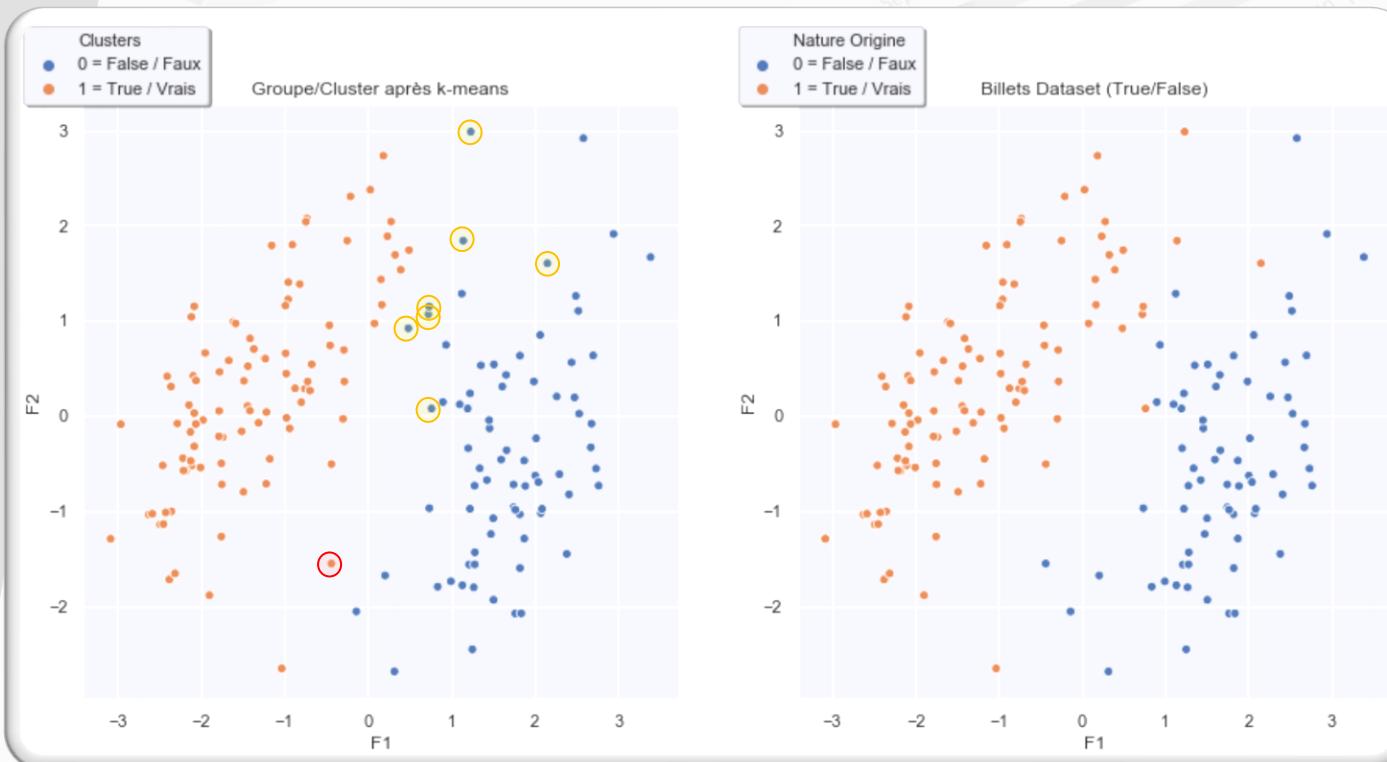
```
# Comme nous savons que le nb de clusters kmeans "vrais" est > nb de clusters kmeans "faux"
# on lui assigne le bon n° de cluster afin de comparer avec le DF original
```

```
k0 = dfk[dfk["clusterk"] == 0][['clusterk']].count()
k1 = dfk[dfk["clusterk"] == 1][['clusterk']].count()
if k0 > k1:
    dfk['clusterk'] = dfk['clusterk'].map({0: 1, 1: 0})
```

```
pd.options.mode.chained_assignment = 'warn' # reset to default
print("k0 = ", k0, " / k1 = ", k1)
dfk.head()
```

Analyse - Algorithme de Classification

Est-ce que les groupes k-means correspondent aux Groupes "Vrai/Faux" billets ?



Matrice de Confusion - Scikit-Learn

n = 170

		FAUX	VRAIS
ACTUAL	FAUX	True Negative (69)	False Positive (1)
	VRAIS	False Negative (7)	True Positive (93)
		(76)	(100)

ACC - La Précision (Accuracy) - Somme Resultats 'true' / Total -- 95.294 (%)

TPR - La Sensitivité (True Positive Rate) - Détection de Faux Billets -- 98.571 (%)

FPR - Le 'Fall Out' (False positive rate) - Probabilité de fausse alarme -- 9.211 (%)

$$\text{accuracy (ACC)} \\ \text{ACC} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}$$

$$\text{true positive rate (TPR) or sensitivity} \\ \text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{fall-out or false positive rate (FPR)} \\ \text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

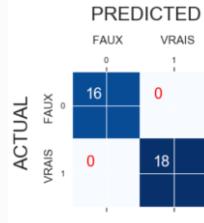
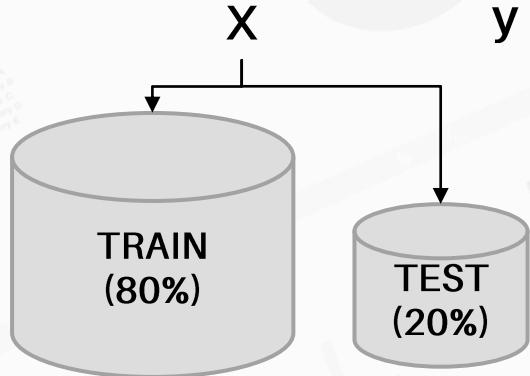
Modèle – Régression Logistique



`sklearn.linear_model.LogisticRegression`

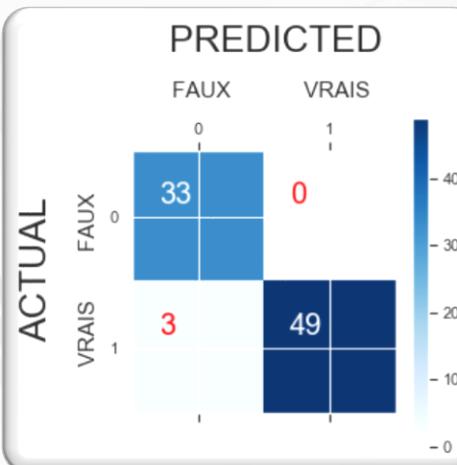
Data - Training

	F1	F2	F3	F4	is_genuine	clusterk
0	2.15	1.60	1.79	2.43	1	0
1	-2.11	-0.53	0.54	0.34	1	1
2	-1.97	-0.05	0.86	0.37	1	1
3	-2.06	-0.09	-0.53	0.52	1	1
4	-2.40	0.41	3.32	-0.84	1	1



1. Préparation (SPLIT) des Bases de données "Training" et "Prédictions" à partir de notre Dataset.
Préconisation :
 - o 80 % pour le training set
 - o 20 % pour le testing set.
2. Construction du modèle de régression logistique sur la base "TRAIN" en utilisant:
- La librairie Scikit-Learn
3. Calcul des prédictions.
Matrice de confusion sur la Base "TEST".

Ici pour provoquer plus d'erreur, je choisis une répartition de 50% - 50%



Modèle – Régression Logistique

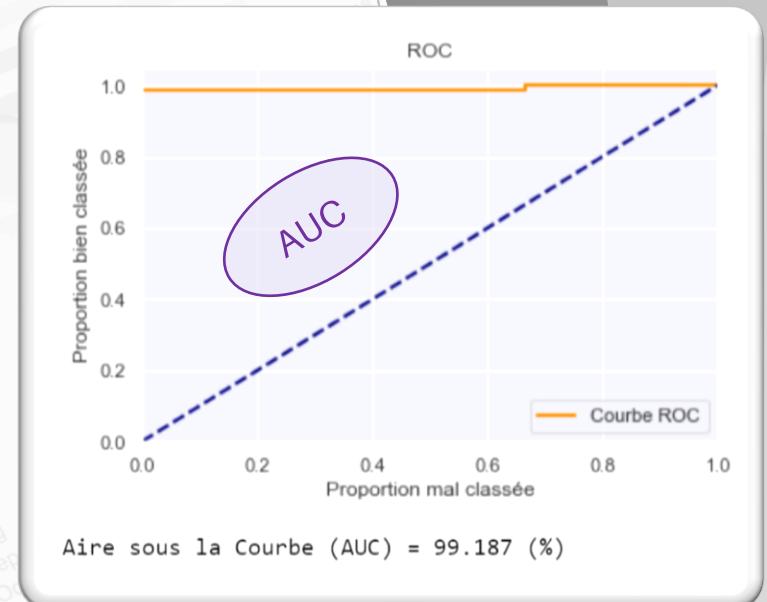
4. Calcul des probabilités sur chaque ligne de la Base "TEST"
5. Construction de la courbe ROC montrant la répartition de ces probabilités (prédictions) bien classées et mal classées.

6. Détermination de l'aire sous courbe AUC

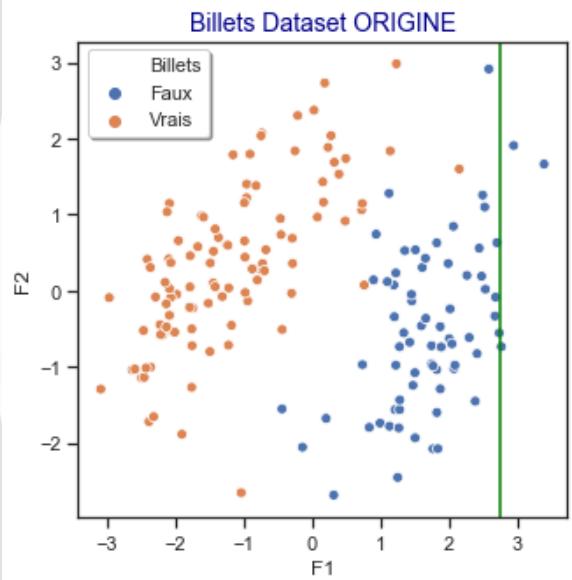
(Area Under the Curve) qui montre la pertinence du modèle.

Plus la courbe est haute, meilleur le modèle est.

Proportion d'exemples bien classés par rapport à la proportion d'exemple mal classés.



VERIFICATION MODELE



dfacp[dfacp['F1'] > 2.75]						
	F1	F2	F3	F4	F5	F6
112	2.95	1.91	-0.49	1.17	-0.70	-0.07
122	3.39	1.66	1.39	0.64	-0.42	-0.10
158	2.77	-0.74	2.01	-0.36	-0.34	-0.49

Après application du modèle de régression sur ces 3 billets, le résultat est conforme à nos attentes.

Nombre de billets : 3

	diagonal	height_left	height_right	margin_low	margin_up	length	id	Probabilité d'authenticité(%)	Prediction
112	172.32	104.60	104.83	4.84	3.51	112.55	A_1	2.20	False
122	172.29	104.72	104.86	5.71	3.16	112.15	A_2	0.97	False
158	171.84	104.32	104.50	6.28	3.00	111.06	A_3	0.16	False

Test sur 3 billets « faux » connus d'avance que l'on retire Dataset avant entrainement du modèle de régression.

Vérification de Billets

Testons avec votre jeu de données ...

1°) - Déposons votre fichier « csv » dans le répertoire approprié



2°) - Lancement du modèle de vérification



Indiquez le nom de votre fichier « csv »

(et le répertoire s'il n'est pas à la racine de ce programme)

Veuillez entrer le nom de votre fichier csv (sans l'extension) : DATA/example

Nombre de billets : 5

	diagonal	height_left	height_right	margin_low	margin_up	length	id	predict	Probabilité d'authenticité(%)
0	171.76	104.01	103.54	5.21	3.30	111.42	A_1	False	2.64
1	171.87	104.17	104.13	6.00	3.31	112.09	A_2	False	48.98
2	172.00	104.58	104.29	4.99	3.39	111.57	A_3	True	97.51
3	172.49	104.55	104.34	4.44	3.03	113.20	A_4	True	100.00
4	171.65	103.63	103.56	3.77	3.16	113.33	A_5	False	0.88

3°) – Résultat de la prédiction



Un fichier Excel est généré dans le même répertoire que le fichier csv a analyser :

nom_fichier_RESULTAT_ANALYSE.xlsx

Projet 6

Création Algorithme de Détection de Faux-billets

Merci de votre attention !



QUESTIONS ???

Pour plus d'informations, vous pouvez me contacter
en suivant le lien directement ci-dessous

frederic.boissy@gmail.com

Programme (python)

jupyter pj6_m3program.py ✓ vendredi dernier à 15:43

File Edit View Language

```
1 # -*- coding: utf8 -*-
2 from init_libraries import *
3
4 def ctrl_note(input_file):
5     df = pd.read_csv("OUTFILES/df.csv")
6     # Chargement du fichier à Evaluer et Préformatte (acp) avant application Modèle
7     ex = input_file
8     ex1 = ex.drop(columns = 'id').values
9     # Préparation Données pour modèle Regression Logistique
10    x = df.copy()
11    x = df.drop(columns = 'is_genuine').values
12    y = df['is_genuine'].values
13    y.astype(int)
14
15    # Centrage et Réduction
16    std_scale = preprocessing.StandardScaler().fit(x)
17    x_scaled = std_scale.transform(x)
18    ex_scaled = std_scale.transform(ex1)
19
20    # Calcul des composantes principales
21    n_comp = 4                                # choix du nombre de composantes à calculer
22    pca = decomposition.PCA(n_components=n_comp)
23    pca.fit(x_scaled)
24    x_projected = pca.fit_transform(x_scaled)
25    ex_projected = pca.fit_transform(ex_scaled)
26
27    # Création du Modèle de regression Logistique sur notre Dataset
28    logreg = LogisticRegression(solver='lbfgs')
29    logreg.fit(x_projected, y)
30
31    # Traitement Prédictions sur fichier Test (example) et sortie dans un Dataframe
32    e = ex.count()
33    if e["id"] > 0 :
34        y_pred = logreg.predict(ex_projected)
35        ex["predict"] = y_pred
36        tx_conf = logreg.predict_proba(ex_projected)
37        i = tx_conf[:,0]
38        ex["Probabilité d'authenticité(%)" ] = np.round((100-i*100), 2)
39        ex = ex.replace({'predict' : 1}, "Le billet est Faux")
40        ex = ex.replace({'predict' : 0}, "Le billet est Vrai")
41    else :
42        print("Fichier vide")
43
44    print("Nombre de billets : ",e["id"])
45    return ex
```



Annexes

Sources en Python

Data Import - Analyse descriptive – ACP – Kmeans

[Lien](#)

Modèle de Régression Logistique

[Lien](#)

Programme de Détection

[Lien](#)

Data Import - Analyse descriptive

Etude des différences de « moyennes » entre les « Vrais » et « Faux » billets

Ici la Question induite serait :
Est-ce que cet écart de moyenne est « Statistiquement » significatif ?

DIFFÉRENCES DES MOYENNES ENTRE VRAIS ET FAUX BILLETS

- Ecart diagonal => 0.08624285714290636
- Ecart height_left => -0.27892857142857963
- Ecart height_right => -0.369671428571408
- Ecart margin_low => -1.1380714285714255
- Ecart margin_up => -0.2790714285714291
- Ecart length => 1.5464857142857085

1°) Vérifions **SI** les variables sont "Gaussiennes" (si elles suivent une loi normale)

- Test Shapiro, avec Hypothèse Nulle H0 de normalité qui devient fausse si la p-value est < a seuil fixé à 5% (0.05)

Toutes les variables sont gaussiennes, avec des p-values élevées (toutes > 95%)

2°) On peut donc faire :

- un Test de Student sur les moyennes
 (Hypothèse nulle H0 = Egalité des moyennes/variances des variables sur les groupes True & False)

TEST t STUDENT - Comparaison des Moyennes - Datasets TRUE vs FALSE

```
La p-value de la variable ['diagonal'] du Test (t) de Student sur les Groupes True & False est : 0.07018967008887296
La p-value de la variable ['height_left'] du Test (t) de Student sur les Groupes True & False est : 2.3342002888499904e-10
La p-value de la variable ['height_right'] du Test (t) de Student sur les Groupes True & False est : 6.665246409290165e-15
La p-value de la variable ['margin_low'] du Test (t) de Student sur les Groupes True & False est : 3.940145276272617e-39
La p-value de la variable ['margin_up'] du Test (t) de Student sur les Groupes True & False est : 7.567386063614238e-17
La p-value de la variable ['length'] du Test (t) de Student sur les Groupes True & False est : 1.2348226459862946e-43
```

Les "p-value" (hormis la diagonal) nous permettent de rejeter (très facilement) l'hypothèse nulle H0 même avec un risque alpha = 1%

L'égalité des moyennes de toutes les variables des deux échantillons de billets (vrais et faux) n'est donc pas vérifiée.

Annexes

Guide de Choix d'un Test Statistique (Excel)

Lien

Guide Choix Tests Statistiques.xlsx - Excel

	B	C	D	E	F	G
1	Données	Hypothèse nulle (H0)	Exemples	Tests paramétriques "Echantillons Appariés" (mêmes individus)	Conditions de validité (tests paramétriques)	Tests non-paramétriques "Echant. Non appariés"
2	mesures sur 1 échantillon ; moyenne théorique (1 chiffre)	moyenne observée = moyenne théorique	Comparaison à une norme d'un taux de pollution mesuré	Test t (Student) pour un échantillon	2	Test de Wilcoxon pour un échantillon
3	mesures sur 2 échantillons	Les positions* sont identiques	Comparaison de notes d'étudiants entre deux classes	Test t (Student) pour échantillons indépendants	1 ; 3 ; 5	Mann-Whitney
4	mesures sur plusieurs échantillons	Les positions* sont identiques	Comparaison du rendement de maïs selon 4 engrains différents	ANOVA	1 ; 3 ; 4 ; 6	Kruskal-Wallis
5	deux séries de mesures quanti sur les mêmes individus (avant-après)	Les positions* sont identiques	Comparaison du taux d'hémoglobine moyen avant / après l'application d'un traitement sur un groupe de patients	Test t (Student) pour échantillons appariés	10	Wilcoxon
6	Plusieurs séries de mesures quanti sur les mêmes individus (avant-après)	Les positions* sont identiques	Suivi de la concentration d'un élément trace au cours du temps au sein d'un groupe de plantes	ANOVA à mesures répétées; modèles mixtes	10 ; Sphérique	Friedman
7	Plusieurs séries de mesures binaires sur les mêmes individus (avant-après)	Les positions* sont identiques	Différents juges évaluent la présence/l'absence d'un attribut sur différents produits			Test Q de Cochran
8	Mesures sur deux échantillons	variance(1) = variance(2)	Comparaison de la dispersion naturelle de la taille de 2 variétés d'un fruit	Test de Fisher		
9	Mesures sur plusieurs échantillons	variance(1) = variance(2) = variance(n)	Comparaison de la dispersion naturelle de la taille de plusieurs variétés d'un fruit	Test de Levene Test de Bartlett		
10	une proportion observée ; son effectif associé ; une proportion théorique	proportion observée = proportion théorique	Comparaison de la proportion de femelles à une proportion de 0.5 dans un échantillon	Test pour une proportion (chi²)		
11	Effectif de chaque catégorie	proportion(1) = proportion(2) = proportion(n)	Comparaison des proportions de 3 couleurs d'yeux dans un échantillon	chi²		
12	Proportion théorique et effectif associés à chaque catégorie	proportions observées = proportions théoriques	Comparer les proportions de génotypes obtenus par croisement FixFix à des proportions mendéliennes (1/2, 1/4, 1/2)	Test d'ajustement multinomial		
13	Tableau de contingence	variable 1 et variable 2 sont indépendantes	La présence d'un attribut est-elle liée à la présence d'un autre attribut?	chi² sur un tableau de contingence	1 ; 9	Test exact de Fisher ; méthode de Monte Carlo
14	mesures de deux variables sur un échantillon	variable 1 et variable 2 sont indépendantes	La biomasse de plante change-t-elle avec la concentration de Pb?	Corrélation de Pearson (& Neyman)	7 ; 8	Corrélation de Spearman
	Mesures d'une variable quantitative sur un échantillon; paramètres de la distribution	Les distributions observée et théorique sont les mêmes	Les salaires d'une société suivent-ils une distribution normale de moyenne 2500 et			Kolmogorov-Smirnov