LUTTE CONTRE LE FAUX-MONNAYAGE

NGUYEN Huynh Thanh Qui
Février 2022

Cliquer sur ce boton pour faire apparaître/disparaître le code

Show Code

PARTIE 2

Analyse de données

[Information générale de nos données]

```
<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)
memory usage: 71.9 KB
```

Out[4]:

```
(1500, 7)
```

- Dans le contexte de mettre en place une modélisation qui serait capable d'identifier automatiquement les vrais des faux billets, j'utilise d'abord un méthode de classification supervisée qui s'appelle **Régression logistique**
- *"L'objectif de la régression logistique est de modéliser, de classifier, une variable binaire prenant ses valeurs dans {0,1} en fonction de variables explicatives quantitatives (et potentiellement qualitatives)"* (source : OpenClassrooms (https://openclassrooms.com/fr/courses/4525326-realisez-des-modelisations-de-donnees-performantes/5754148-apprehendez-le-fonctionnement-de-la-regression-logistique)). Dans ce cas, le variable à expliquer qualitative signifie des vrais et faux billets, qui sont présentés respectivement par 1 et 0
- Cependant, afin de vérifier la fiabilité de cette méthode ainsi que sa performance de prédiction, j'utilise également un autre méthode de classification non-supervisée qui s'appelle **Partitionnement (classification) K-MEANS**. Le but est de mettre en concurrence deux méthodes de prédiction pour sélectionner l'algorithme le plus performant possible pour identifier un maximum de faux billets au sein de la masse de billets analysés
- Pour construire les modèles de prédictions, je divise les données en 2 parties qui seviront aux données d'apprentissages et de test (pour vérifier la capacité prédictive des modèles) (à noter que toutes données utilisées sont normalisées en 0 et 1). La division est aléatoire et suit une proportion de 80/20 (apprentissage/test)

Préparation des données d'apprentissage (80%) et de test (20%)

Données d'apprentissage (avec variable dépendante ('is_genuine')) et les types de billets correspondants

Nombre de billets d'apprentissage: 1200

	is_genuine	diagonal	height_left	height_right	margin_low	margin_up	length
105	1.0	0.837563	0.425287	0.563725	0.329082	0.567073	0.767361
199	1.0	0.654822	0.597701	0.553922	0.334184	0.439024	0.949242
926	1.0	0.370558	0.281609	0.333333	0.137755	0.640244	0.672191
1497	0.0	0.253968	0.946317	0.664181	0.555521	1.000000	0.000000
306	1.0	0.649746	0.488506	0.539216	0.316327	0.414634	0.737753

(1200, 7)

L'étiquette de données 'test':

```
105     1.0
199     1.0
926     1.0
1497    0.0
306     1.0
Name: is_genuine, dtype: float64
```

Données de test (sans variable dépendante ('is_genuine')) et les types de billets correspondants

Nombre de billets d'apprentissage : 300

	diagonal	height_left	height_right	margin_low	margin_up	length
831	0.421320	0.465517	0.299020	0.244898	0.457317	0.790625
416	0.441624	0.488506	0.289216	0.109694	0.396341	0.725063
358	0.345178	0.580460	0.411765	0.285714	0.439024	0.771591
88	0.456853	0.201149	0.446078	0.321429	0.560976	0.731408
1223	0.284264	0.436782	0.690141	0.798469	0.780488	0.258586

(300, 6)

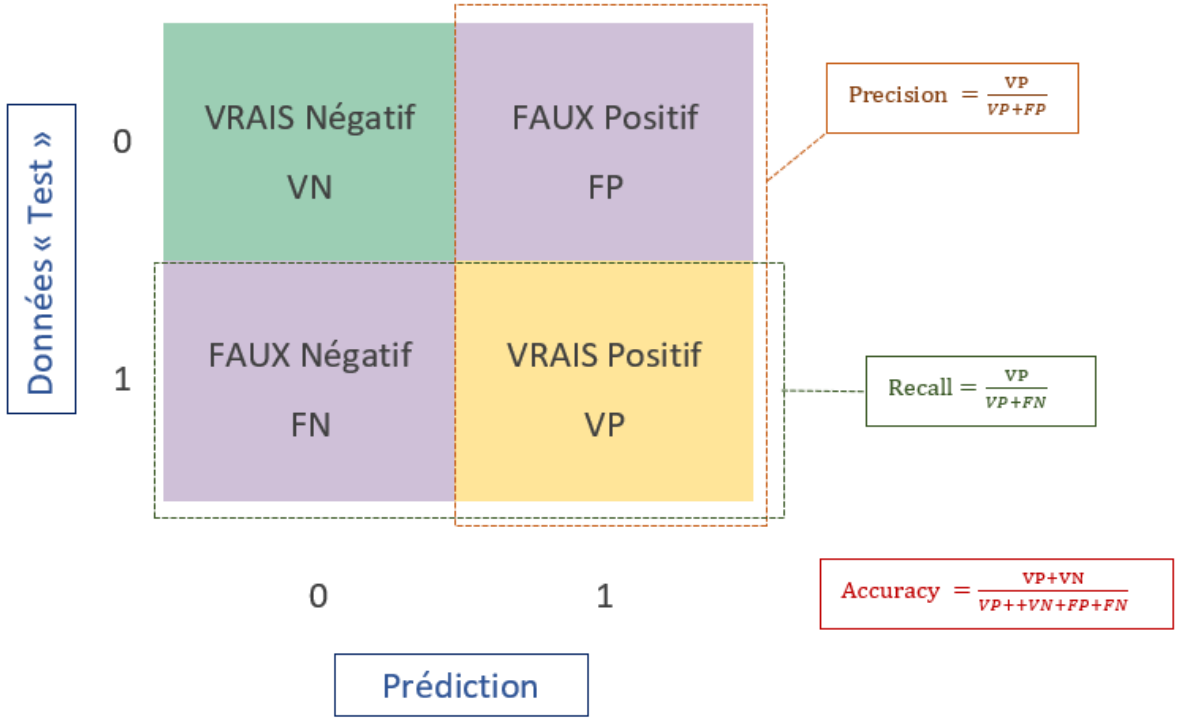
L'étiquette de données 'test':

```
831     1.0
416     1.0
358     1.0
88      1.0
1223    0.0
Name: is_genuine, dtype: float64
```

Vérification la performance de prédiction

Matrice de confusion

- La fiabilité des modèles sera vérifiée par la matrice de confusion et les indicateurs présentés en-dessous



- En générale, ce matice évalue la cohérence entre les étiquettes générées par les modèles (0 et 1) et celles de données "test" (régression logistique)/données brutes (partitionnement K-MEANS)
- Les données qui **sont prédites correctement** sont classées dans le groupe nommé **"VRAIS"** (et l'inverse pour le groupe "FAUX"). Les étiquettes **1** et **0** sont nommées **"POSITIF"** et **"NEGATIF"**

I. Régression logistique

- La modélisation de la variable binaire (catégorielle dans ce cas) en utilisant la régression logistique demande une donnée d'apprentissage complète (données contenant de variable dépendante "is_genuine"). C'est pour cette raison, j'utilise les données **x_training**
- Le résultat de cette régression se trouve toujours entre 0 et 1, ce qui correspond à la probabilité d'apparaître de l'événement qu'on souhaite à analyser. Dans notre cas, j'utilise le seuil de la **probabilité d'acceptable de vrais billets à 50%** pour la prédiction. Plus précisément, si le résultat est suppérieur ou égale à 0,5, ce billet est classé comme un vrai et l'inverse pour les faux billets

1. Génération du modèle de la régression logistique en utilisant les données d'apprentissage

```
                 Generalized Linear Model Regression Results
==============================================================================
==
Dep. Variable:              is_genuine   No. Observations:                 12
00
Model:                             GLM   Df Residuals:                     11
93
Model Family:                Binomial   Df Model:
6
Link Function:                   logit   Scale:                          1.00
00
Method:                           IRLS   Log-Likelihood:                -16.9
69
Date:                Sat, 12 Feb 2022   Deviance:                       33.9
38
Time:                        17:22:50   Pearson chi2:                     6
7.4
No. Iterations:                     11
Covariance Type:            nonrobust
==============================================================================
====
                  coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
Intercept      12.1239      6.239      1.943      0.052      -0.104      2
4.352
margin_low    -38.2589      9.689     -3.949      0.000     -57.249     -1
9.269
diagonal       -1.2494      3.569     -0.350      0.726      -8.245
5.746
height_left    -3.1794      3.056     -1.040      0.298      -9.169
2.810
height_right   -5.5270      3.537     -1.563      0.118     -12.459
1.406
margin_up     -20.8092      6.025     -3.454      0.001     -32.617      -
9.001
length         33.6925      6.657      5.061      0.000      20.646      4
6.739
==============================================================================
====
```

Modèle de la régression logistique

model_GLM = model_glm.predcit()

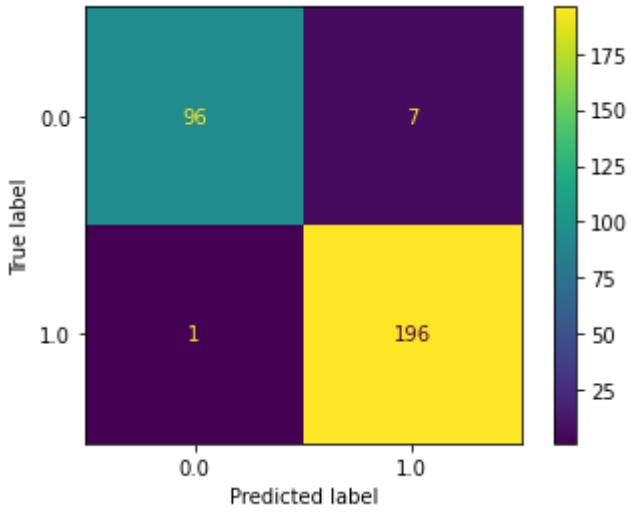
2. Matrice de confusion entre les données "test" et les résultats prédictifs de la régression logistique

```
              precision    recall  f1-score   support

         0.0       0.99      0.93      0.96       103
         1.0       0.97      0.99      0.98       197

    accuracy                           0.97       300
   macro avg       0.98      0.96      0.97       300
weighted avg       0.97      0.97      0.97       300

[[ 96   7]
 [  1 196]]
```



II. Clustering K-MEAN

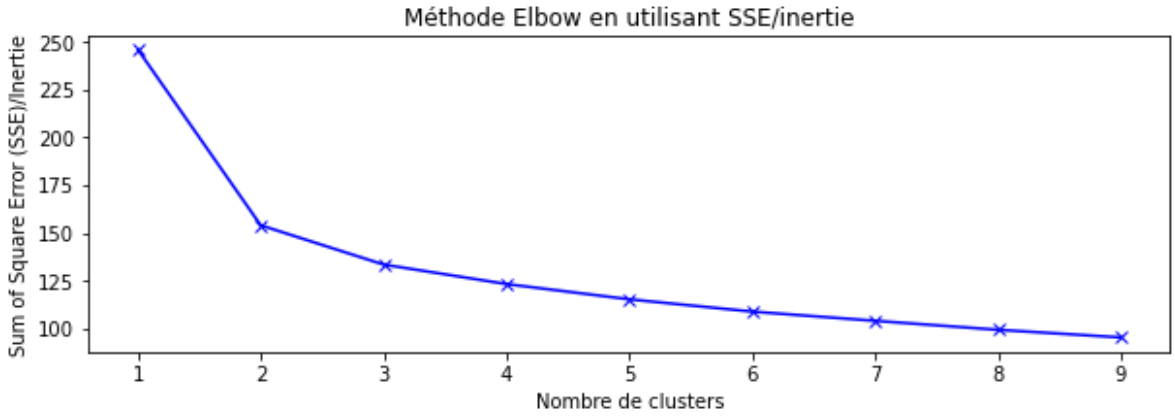
- Contrairement à la régression logistique, le technique de partitionnement K-MEANS n'utilise pas le variable à expliquer ('is_genuine') car il correspond à une classification **non-paramétrique**
- La performance de cette méthode sera évaluée par l'indice de **l'homogénéite** et **la matrice de confusion** entre les clusters prédits et les types de billets issus de données brutes (et complètes)
- Le choix du nombre de clusters K-MEANS est basé sur le résultat de la méthode du coude

Données normalisées pour la classification K-MEANS

	margin_low	diagonal	height_left	height_right	margin_up	length
0	0.392857	0.390863	0.988506	1.000000	0.378049	0.684880
1	0.201531	0.213198	0.126437	0.367647	0.439024	0.739867
2	0.362245	0.837563	0.770115	0.289216	0.408537	0.754672
3	0.163265	0.162437	0.442529	0.504902	0.451220	0.828693
4	0.270408	0.350254	0.655172	0.269608	0.737805	0.623548

```
(1500, 6)
```

1. Choix du nombre de cluster K-MEANS: méthode du coude



Nombre de groupes optimal choisi pour la méthode K-MEANS est 2

2. Classification K-MEANS

Homogénéite entre les données brutes et celles issues de la classification K-MEANS est 88.44%

Nombre de billets par cluster K-MEANS

Cluster	Nombre de billet
0	495
1	1005

Grâce à la valeur de l'homogénéité citée en dessus, on peut déduire que pres que 90% des clusters de K-MEANS sont similaires aux clusters de données brut es, malgré une différenciation dans la nommenclature du nom de clusters

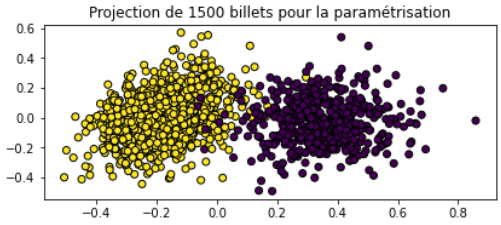
Modèle de la classification K-MEANS

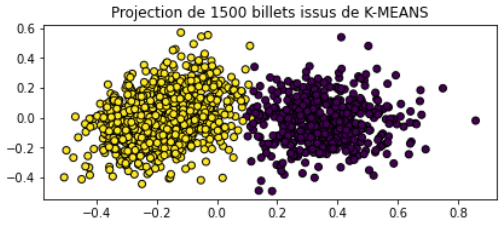
model_k_means = model_km.predict()

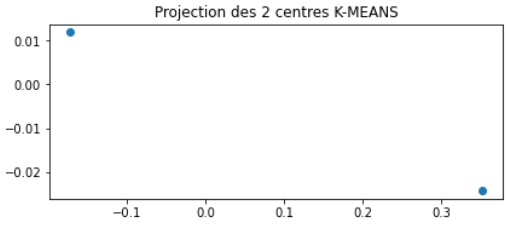
2. Comparaison (par PCA) la répartition de clusters de K-MEANS et de données brutes sur un plan 2D

Out[20]:

```
Text(0.5, 1.0, 'Projection des 2 centres K-MEANS')
```







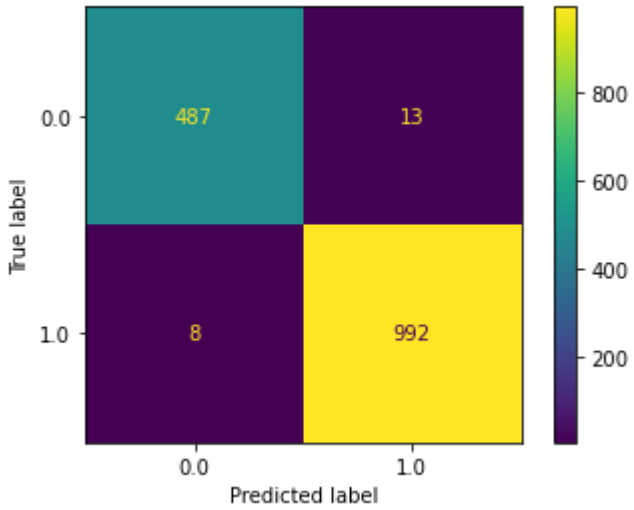
3. Matrice de confusion entre les données brutes et les clusters de K_MEANS

```
              precision    recall  f1-score   support

         0.0       0.98      0.97      0.98       500
         1.0       0.99      0.99      0.99      1000

    accuracy                           0.99      1500
   macro avg       0.99      0.98      0.98      1500
weighted avg       0.99      0.99      0.99      1500

[[487  13]
 [  8 992]]
```



Remarque (si les indicateurs affichés en-dessus ayant une valeur inférieure à 0,15)

- La précision et le rappel (ou autres indicateurs) du matrice de confusion obtenus dans le cas de la classification K-MEANS peuvent avoir des valeurs très faibles **(inférieurs à 0,05)**. Ce phénomène ne signifie pas qu'elle n'est pas fiable ou n'a pas de capacité prédictive car **presque 90% des données classifiées sont cohérentes aux données brutes** (partie II.2)

- Cette situation dû à la **nomenclature automatique de clusters** issus de la **classification non-supervisée**. Pour avoir les bons résultats, il faut les recalculer en utilisant ce formule: **(1 - Valeur d'Indicateur)**
- **Les indicateurs doivent normalement ayant une valeur suppérieure à 0,95 (ou 95%)**

III. Choix du modèle

- La **précision** signifie **le pourcentage de valeurs positives/négatives prédites** qui sont **réellement positives/négatives**
- Le **rappel (recall)** signifie **le pourcentage de valeurs positives/négatives qui sont été correctement prédites sur tous les enregistrements positifs/négatives**
- Afin de choisir le modèle le plus performant, je vais comparer les valeurs des indicateurs cités en-dessus; plus qu'elles sont grandes, plus le modèle est fiable
- En comparant les résultats de matrices de confusion issus des 2 méthodes (appliqués sur les mêmes données 'test'), on peut déduire que **la régression logistique** est **plus performante** que la classification K-MEANS. De plus, un des points négatifs du partitionnenement K-MEANS est la nomenclature des clusters. En fait, c'est une classification non paramétrique, donc, la génération du nom de clusters est aléatoire et peut causer une incohérence du nom de billets prédictifs

PARTIE 3

Test du modèle

Copier et remplacer le nom dans les parenthèses par le nom du dataframe contenant les données de billets à analyser

identification_billet_RL(data_a_verifie)

Exemple d'un test de l'algorithme

```
identification_billet_RL(billets_production)
```

Out[23]:

	Classe_RL	diagonal	height_left	height_right	margin_low	margin_up	length	id
0	False	171.76	104.01	103.54	5.21	3.30	111.42	A_1
1	False	171.87	104.17	104.13	6.00	3.31	112.09	A_2
2	False	172.00	104.58	104.29	4.99	3.39	111.57	A_3
3	True	172.49	104.55	104.34	4.44	3.03	113.20	A_4
4	True	171.65	103.63	103.56	3.77	3.16	113.33	A_5