PROJET 10

Détectez les faux billets avec Python

Contexte du projet

ONCFM

Institution qui a pour objectif de mettre en place des méthodes d'identification des contrefaçons des billets en euros.

ONCFM

Méthode

Mise en place d'une modélisation qui serait capable d'identifier automatiquement les vrais des faux billets. A partir du jeu de données mis à notre disposition.

1. Analyse descriptive et exploratoire des données (valeurs manquantes, extrêmes, matrice de corrélation, Test…)
2. Régression linéaire / Imputation des valeurs manquantes (résidus, homoscédasticité)
3. Partitionnement(ACP, Kmeans, matrice de confusion)
4. Régression logistique (métriques, courbe ROC,matrice de confusion)
5. Methode KNN (métriques ,matrice de confusion)
6. Test de l'algorithme

ONCFM

Analyse Descriptive

Le dataframe regroupe des données **géométriques** de plusieurs billets (Des vrais et des faux).

```
# Importer le dataset billets
data = pd.read_csv('billets.csv',sep = ';')
billets=data.copy()
billets.head()
```

	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	171.46	103.36	103.66	3.77	2.99	113.09
2	True	172.69	104.48	103.50	4.40	2.94	113.16
3	True	171.36	103.91	103.94	3.62	3.01	113.51
4	True	171.73	104.28	103.46	4.04	3.48	112.54

- Length : la longueur du billet (en mm)
- Height_left : la hauteur du billet (mesurée sur le côté gauche, en mm) ;
- Height_right : la hauteur du billet (mesurée sur le côté droit, en mm)
- Margin_up : la marge entre le bord supérieur du billet et l'image de celui-ci (en mm)
- Margin_low : la marge entre le bord inférieur du billet et l'image de celui-ci (en mm)
- Diagonal : la diagonale du billet (en mm).
- Is_genuine : True ou False

ONCFM

Analyse Descriptive

Aperçu rapide du jeu de données et recherche des valeurs manquantes

1500 billets par variables

```
# Imprimons un sommaire de notre dataframe
billets.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
```

Sauf Margin_low a 1463 donc 37 valeurs manquantes

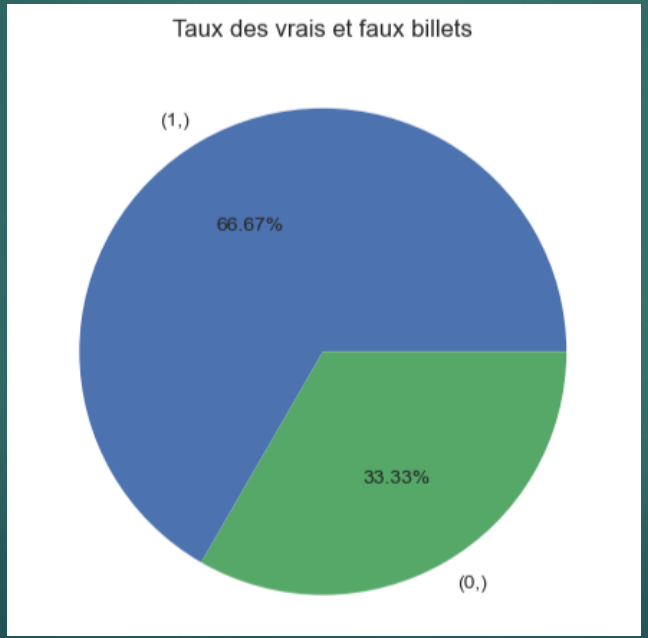
```
# vérifions les valeurs manquantes
billets.isnull().sum()
```

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

ONCFM

Analyse Descriptive

Répartition des billets présent dans le jeu de données

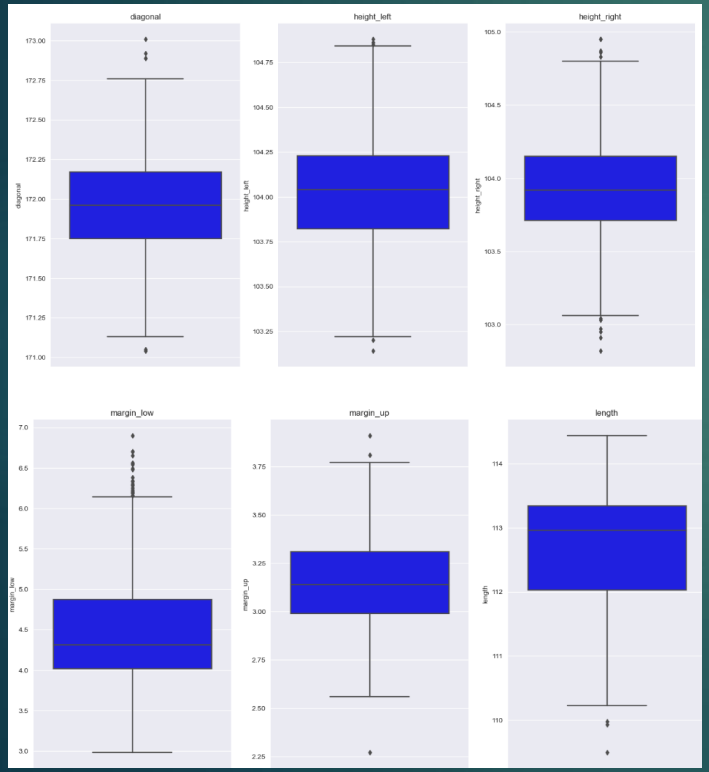


Sur les 1500 billets 33 % de billets sont faux et 67 % sont authentiques.

ONCFM

Analyse Descriptive

Aperçu rapide des valeurs aberrantes



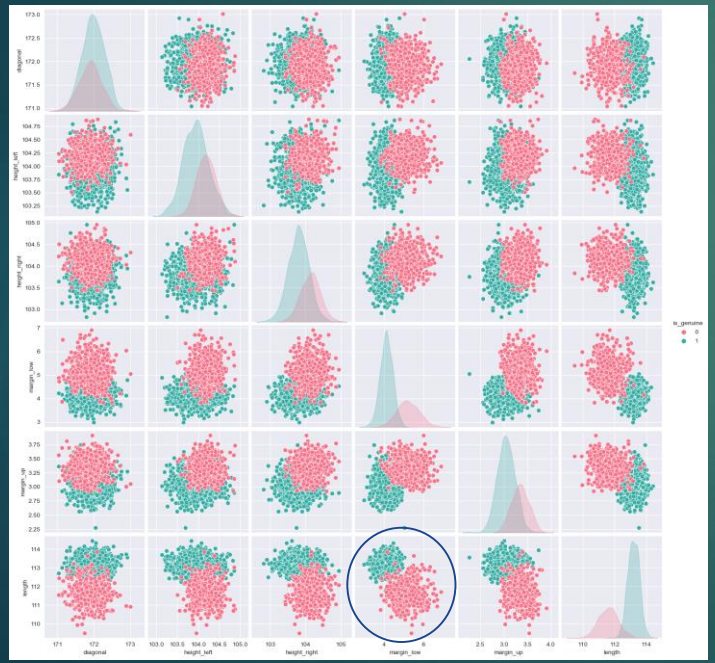
Les 6 variables comportent toutes des observations aberrantes

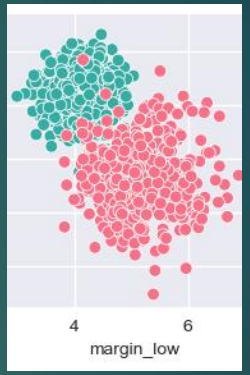
'margin_low' et 'height_right en ont plus

ONCFM

Analyse Descriptive

Un Pairplot est une visualisation de données qui trace les relations par paires entre toutes les variables d'un ensemble de données.



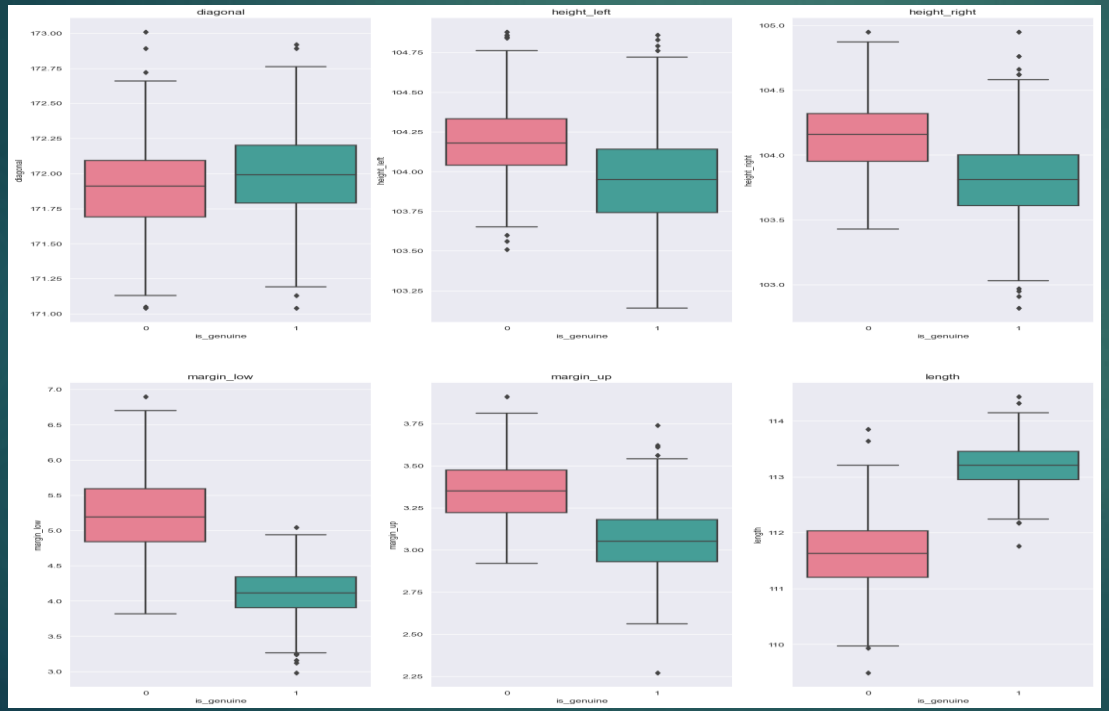


On observe une frontière, un trait qui sépare les vrais et les faux billets sur le croisement entre les 2 variables 'length' et 'margin_low'

ONCFM

Analyse Descriptive

Boxplot des variables en fonction du type de billets ‘ is_genuine’



VRAI

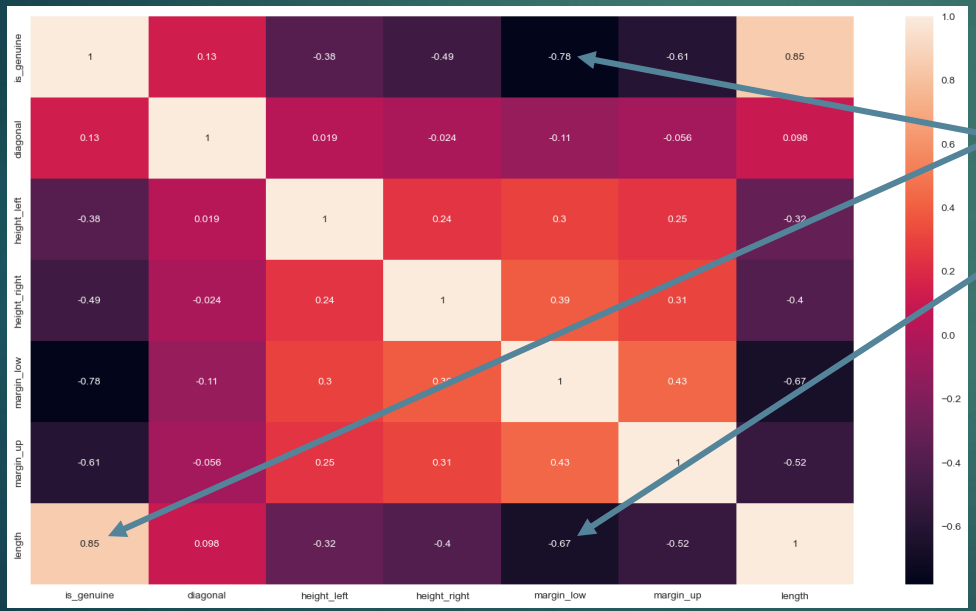
- Diagonale plus élevée
- Longueur plus élevée

FAUX

- Hauteur gauche plus élevée
- Hauteur droite plus élevée
- Marge de basse plus élevée
- Marge haute plus élevée

Analyse Descriptive

La matrice de corrélation indique les valeurs de corrélation, qui mesurent le degré de relation linéaire entre chaque paire de variables.



Les variables fortement corrélées

is-genuine - Lenght = **0.85**

is-genuine - margin_low **-0.78**

Lenght - margin_low = **-0.67**

'Length' permet de prédire la valeur de 'margin_low'

Résumé de l'analyse descriptive

La description des données nous indique :

Le dataset contient des valeurs aberrantes, des valeurs nulles (37 dans la variable 'Margin_Low')

La répartition des vrais/faux billets est un peu inégale 67%/33%.

Les corrélations semblent être bien établies.

Length et Margin Low.

Analyse Exploratoire

Test Shapiro-Wilk : Test de la **normalité** des distributions.

```
['diagonal',
 'height_left',
 'height_right',
 'margin_low',
 'margin_up',
 'length']
```

```
#ShapiroTest:
alpha = 0.05
```

```
from scipy import stats

for i in DSN.iloc[:,:]:
    stat, pvalue = stats.shapiro(donnees_sans_nan[i])
    print(pvalue)
    if pvalue > alpha:
        print("Les données suivent une loi normale")
    else:
        print("Nous rejetons l'hypothèse nulle, les données ne suivent pas une loi normale")
```

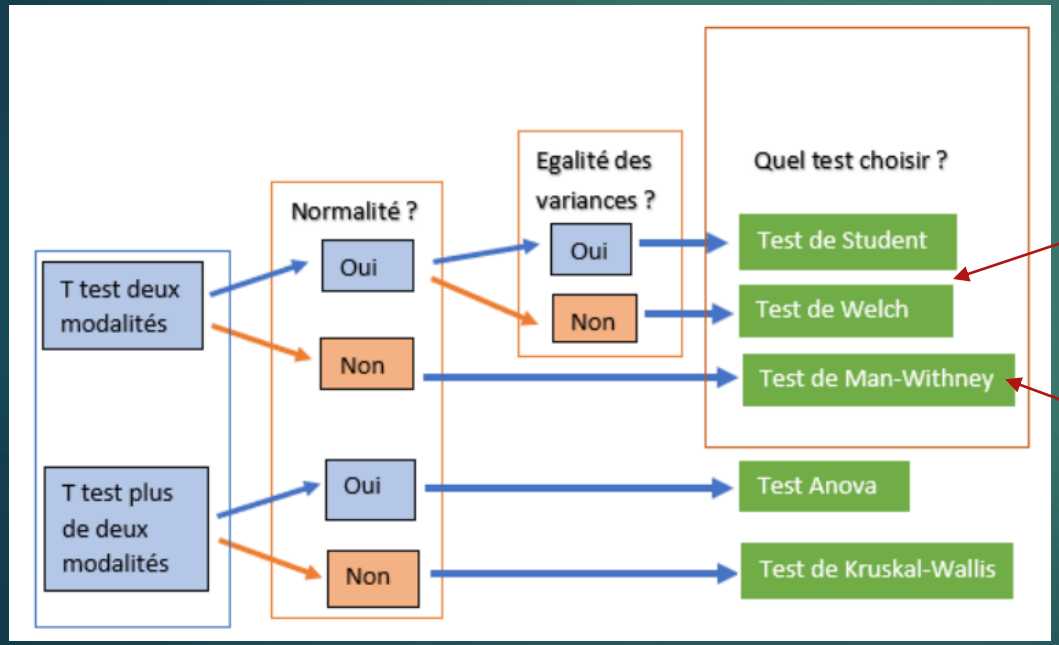
```
0.2687021493911743
Les données suivent une loi normale
0.05763062462210655
Les données suivent une loi normale
0.9576431512832642
Les données suivent une loi normale
2.8283876088209786e-24
Nous rejetons l'hypothèse nulle, les données ne suivent pas une loi normale
0.0004760113952215761
Nous rejetons l'hypothèse nulle, les données ne suivent pas une loi normale
1.0767076021107087e-27
Nous rejetons l'hypothèse nulle, les données ne suivent pas une loi normale
```

Diagonal, Height left et Height Right suivent une loi normale

Margin Low, Margin Up et Length ne suivent pas une loi normale

Analyse Exploratoire

Test de l'égalité des variances sur une distribution normale: Test de LEVENE



Diagonal, Height left et Height Right

Margin Low, Margin Up et Length

Analyse Exploratoire

Test de l'égalité des variances sur une distribution normale: Test de LEVENE

```
Variable : diagonal
Statistique de Levene : 0.12571686270898524
P-valeur : 0.7229661175157234
Pas suffisamment de preuves pour rejeter l'hypothèse nulle (égalité des variances)

Variable : height_left
Statistique de Levene : 42.350386932887965
P-valeur : 1.0457802387852748e-10
Les variances sont statistiquement différentes.

Variable : height_right
Statistique de Levene : 1.329665358267531
P-valeur : 0.24905337703204727
Pas suffisamment de preuves pour rejeter l'hypothèse nulle (égalité des variances)
```

Observation :

Nous allons utiliser le test de Student pour "diagonal" et "height_right" car les variances ne sont pas statistiquement différentes.

Nous allons utiliser le test de Welch pour "height_left" car les variances sont statistiquement différentes.

Diagonal = Test de Student

Height Left = Test de Welch

Height Right = Test de Student

Analyse Exploratoire

Test de STUDENT

```
Test de Student pour la variable 'diagonal':
Statistique de test : 5.1967708778045365
P-valeur : 2.3146624660826948e-07

Test de Student pour la variable 'height_right':
Statistique de test : -21.32295346909739
P-valeur : 4.708598379129022e-88
```

Influence Significative sur Is_Genuine

Analyse Exploratoire

Test de WELCH

```
Test de Welch pour la variable 'height_left':
Statistique de test : -16.918693070906215
P-valeur : 4.851593599662053e-58
```

Influence Significative sur Is_Genuine

Analyse Exploratoire

Test de WELCH

```
Test de Mann-Whitney pour la variable 'margin_low':
Statistique de test : 19094.0
P-valeur : 3.0465930299427695e-182

Test de Mann-Whitney pour la variable 'margin_up':
Statistique de test : 59858.0
P-valeur : 1.328995693686527e-121

Test de Mann-Whitney pour la variable 'length':
Statistique de test : 470034.5
P-valeur : 2.0735965473634398e-201
```

Influence Significative sur Is_Genuine

ONCFM

Résumé de l’analyse exploratoire

L’exploration des données nous indique :

Toutes les variables ont significativement une influence sur ‘Is_Genuine’

ONCFM

Regression linéaire / Imputation

La régression OLS (moindres carrés ordinaires) est une technique pour estimer les coefficients d'une régression linéaire qui décrivent les relations entre une ou plusieurs variables quantitatives et une variable dépendante

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:	Wed, 18 Oct 2023	Prob (F-statistic):	4.75e-299
Time:	15:39:50	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	-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

Environ 61,5 % de la variance 'Margin Low' est expliqué par les variables indépendantes

F-Stastistic elevé + Prob Faible = Modèle statistiquement significatif

Prob Omnibus = 0,000 la distribution des résidus ne semblent pas etre normale

Durbin-Watson est une mesure pour l'auto corrélation des résidus 2,038 = Autocorrélation faiblement positive

Jarque-Bera (JB) sert à tester si la distribution est normale. Prob(JB) est la probabilité que la valeur de Jarque-Bera dépasse 5% 5,62e-09 < 0.05
La distribution ne suit pas une loi normale

P-valeurs supérieur à 5% = Certaines variables sont non significatives

Regression linéaire / Imputation

Methode descendante ou backward pour les variables non signifivatives (P-Valeurs inférieur à 5%)

OLS Regression Results

Dep. Variable:	margin_low	R-squared (uncentered):	0.992
Model:	OLS	Adj. R-squared (uncentered):	0.992
Method:	Least Squares	F-statistic:	5.879e+04
Date:	Mon, 30 Oct 2023	Prob (F-statistic):	0.00
Time:	15:52:11	Log-Likelihood:	-774.71
No. Observations:	1463	AIC:	1555.
Df Residuals:	1460	BIC:	1571.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
is_genuine	-1.1496	0.028	-40.428	0.000	-1.205	-1.094
height_left	0.0569	0.002	30.004	0.000	0.053	0.061
margin_up	-0.2117	0.059	-3.609	0.000	-0.327	-0.097

Omnibus:	22.504	Durbin-Watson:	2.036
Prob(Omnibus):	0.000	Jarque-Bera (JB):	39.317
Skew:	0.060	Prob(JB):	2.90e-09
Kurtosis:	3.794	Cond. No.	595.

99,2% de la variance 'Margin Low' est expliqué par les variables indépendantes

F-Stastistic elevé + Prob Faible = Modèle statistiquement significatif

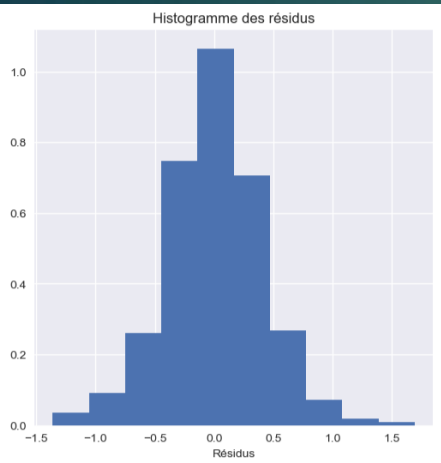
P-valeurs inférieur à 5% = les variables sont significatives

ONCFM

RÉGRESSION LINÉAIRE / IMPUTATION

Hypothèses de validité du modèle

Test de la normalité des résidus



L'allure de l'histogramme est assez sysmetrique.

Si l'on veut tester la normalité des résidus, on peut faire un test de Shapiro-Wilk.

```
]: shapiro(reg_multi.resid)
]: ShapiroResult(statistic=0.9935745596885681, pvalue=5.671638973581139e-06)
```

Ici, l'hypothèse de normalité est remise en cause (p-value = 5.67e-06 < 0.05).

Néanmoins, l'observation des résidus, 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, même si le résidu n'est pas considéré comme étant gaussien.

L'homoscédasticité (test de la contance de la variance des résidus)

```
Test de Breusch-Pagan:
LM Statistic: 164.6597687524693
LM P-Value: 6.098452163335932e-33
F-Statistic: 30.775783000171042
F P-Value: 6.11040412097926e-35
Rejet de l'hypothèse nulle. Il y a des signes d'hétéroscédasticité.
```

Dans le cas d'un "grand échantillon", les propriétés asymptotiques des estimateurs corrigent un manque "raisonnable" de normalité des résidus : le modèle linéaire est dit robuste vis à vis de cette hypothèse.

P-value < 0,5

Rejet de l'hypothèse nulle. Il y a des signes d'hétéroscédasticité,

Lien chercheuse CNRS expliquant le commentaire ci-dessus : Non-respect des hypothèses du modèle linéaire (ANOVA, régression): c'est grave, docteur?? - R-atique (ens-lyon.fr)

ONCFM

RÉGRESSION LINÉAIRE / IMPUTATION

Imputations des valeurs manquantes

```
# Sélectionner les lignes avec des valeurs manquantes dans 'margin_low'
nan_rows = billets[billets['margin_low'].isnull()]

# Sélectionner uniquement les variables indépendantes pour ces lignes
nan_data = nan_rows[['is_genuine', 'height_left', 'margin_up']]

# Ajouter une constante aux données NaN
nan_data = sm.add_constant(nan_data)

# Utiliser le modèle pour faire des prédictions
predictions = reg_multi.predict(nan_data)

print(predictions)
```

```
72      4.070783
99      4.104309
151     4.114361
197     3.979364
241     4.133697
251     4.098470
284     4.071605
334     4.112908
410     4.088638
413     4.076757
445     4.151739
481     4.164640
505     4.132498
611     4.048253
654     4.175857
675     4.214598
710     4.113443
739     4.096128
742     4.082568
780     4.099602
798     4.118753
844     4.140049
845     4.163086
871     4.119130
895     4.116005
919     4.200695
945     4.110948
946     4.065411
981     4.145168
1076    5.250466
1121    5.265191
1176    5.278182
```

```
# Créer une copie du DataFrame original
df = billets.copy()

# Remplacer les valeurs manquantes dans 'margin_low' par les prédictions
df.loc[df['margin_low'].isnull(), 'margin_low'] = predictions

# Afficher les informations sur le nouveau DataFrame
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   int32
 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: float64(6), int32(1)
memory usage: 76.3 KB
```

```
Les valeurs manquantes ont bien été remplacées.
```

Les valeurs manquantes ont bien été remplacées

ONCFM

RÉGRESSION LINÉAIRE / IMPUTATION

Résumé de la partie

Le modèle par la régression linéaire n'est pas forcément pertinent parce que les 2 hypothèses ne sont pas vérifiés à savoir :

- Normalité de la distribution des résidus : Avec JB, Omnibus la distribution ne suit pas une loi normale,

- Homoscédasticité : L'hypothèse d'homoscédasticité de notre régression linéaire est rejeté : **P-value** < 0,5

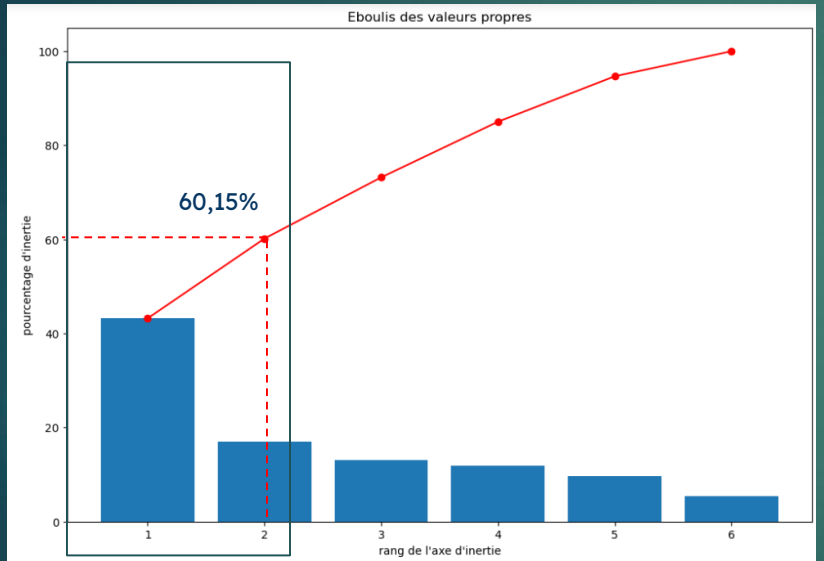
Cependant les explications données plus haut et l'article expliqué par la statisticienne du CNRS permettent de valider le modèle.

ONCFM

Analyse des composantes principales (ACP)

Eboulis des valeurs propres

Définir le nombre d'axes principaux d'analyse afin de définir les axes principaux à analyser




```
print(varexpl)
```

```
[43.20991135 16.95662162 13.01889584 11.82489513  9.66492449  5.32475157]
```

```
scree_cum = varexpl.cumsum().round()
scree_cum
```

```
array([ 43.,  60.,  73.,  85.,  95., 100.])
```

Les 2 premières composantes
expliquent plus de la moitié (60,15 %) de la variabilité

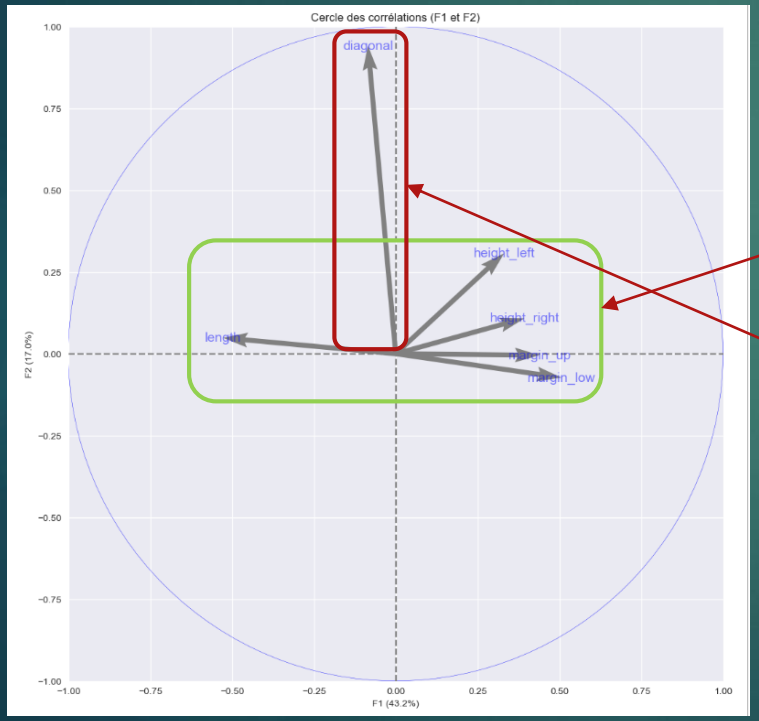
Une analyse sur les deux premiers axes est donc cohérente et suffisante

ONCFM

Analyse des composantes principales (ACP)

Cercle des corrélations

Projection des variables initiales sur un plan à deux dimensions constitué par les deux premiers facteurs



FACTEUR (F1)

Length
Height_left
Height_right
Margin_up
Margin_low

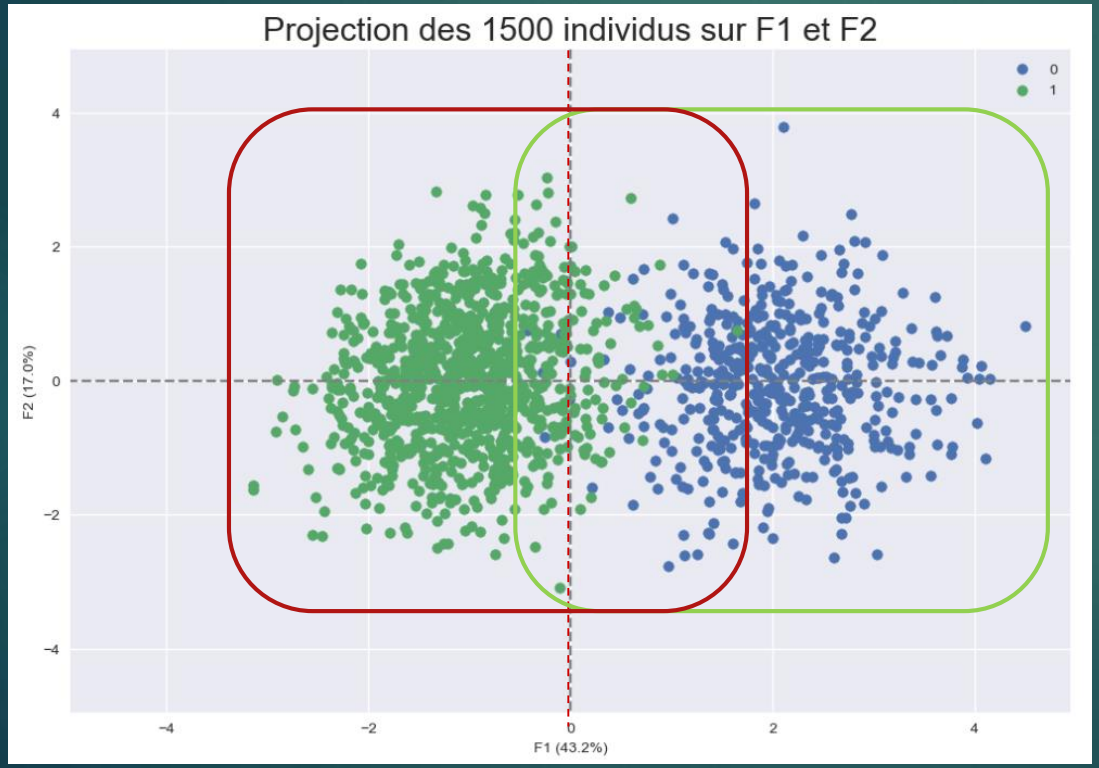
FACTEUR (F2)

Diagonal

ONCFM

Analyse des composantes principales (ACP)

Projection des individus



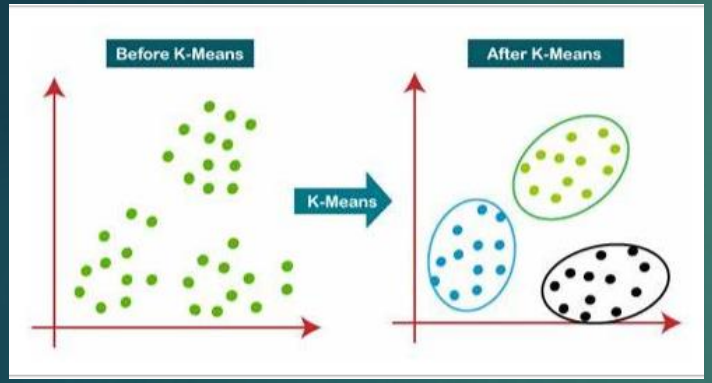
2 Classes bien distinctes en fonction de 'Is-genuine' avec des billets mal classés de part et d'autre

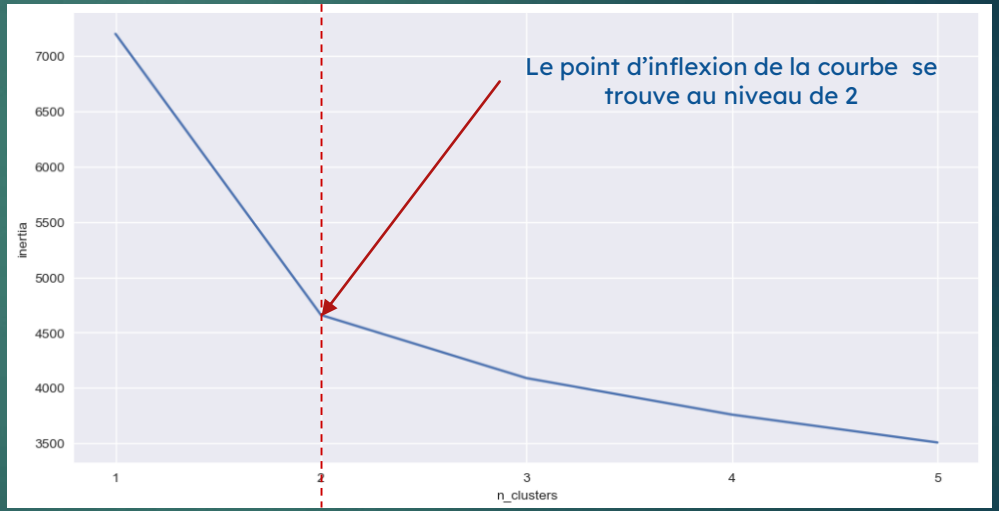
ONCFM

Le K-MEANS

K-MEANS - Nombre de clusters

C'est l'un des algorithmes de clustering les plus répandus. Il permet d'analyser un jeu de données caractérisées par un ensemble de descripteurs, afin de regrouper les données "similaires" en groupes (ou clusters).





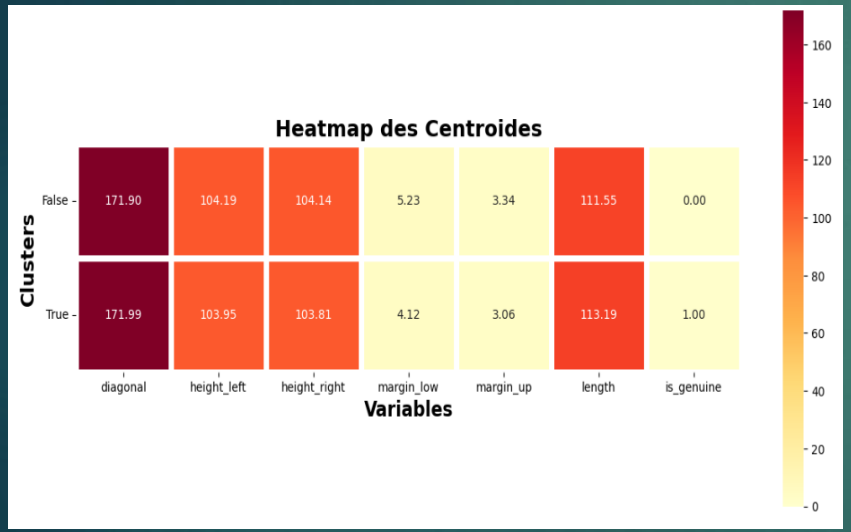
K-means a regroupé 2 Clusters avec les mêmes similitudes

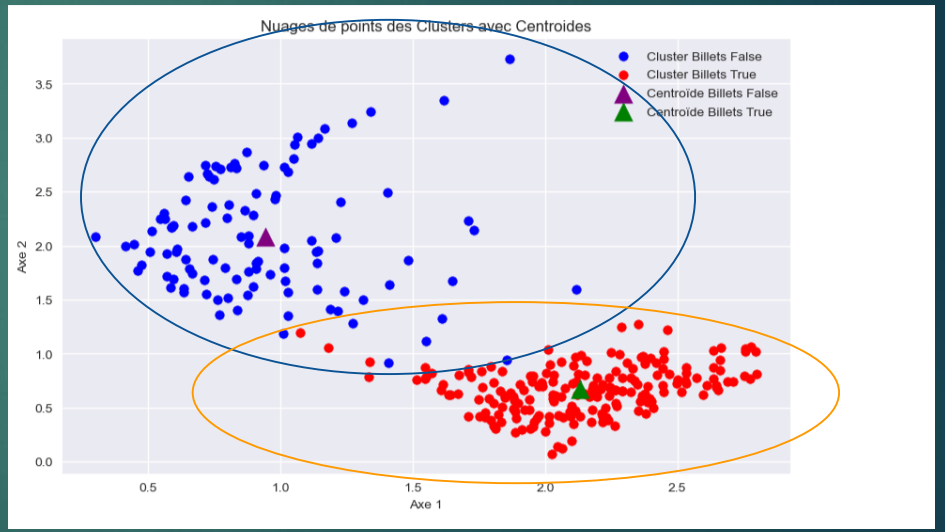
ONCFM

Le K-MEANS

K-MEANS – HeatMap et Projection

C'est l'un des algorithmes de clustering les plus répandus. Il permet d'analyser un jeu de données caractérisées par un ensemble de descripteurs, afin de regrouper les données "similaires" en groupes (ou clusters).





2 Clusters

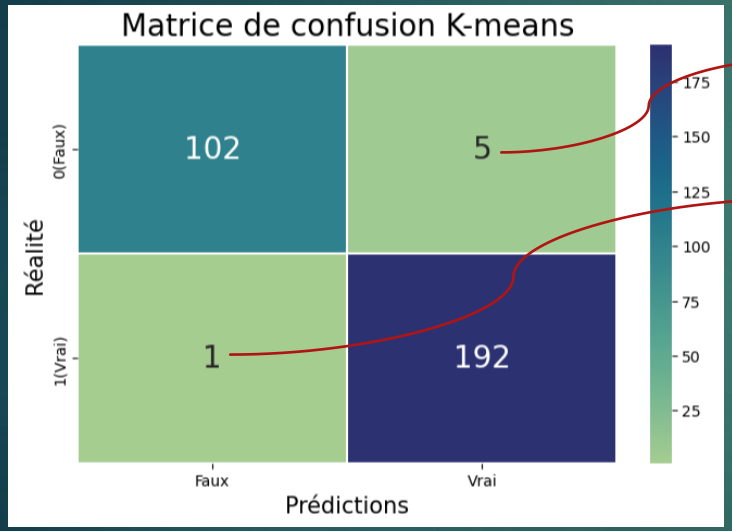
Faux billets = Cluster 0
Vrais billets = Cluster 1

ONCFM

Le K-MEANS

Matrice de confusion et métriques

La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification



5 Vrais billets parmi les faux billets

1 Faux billets parmi les vrais billets

Accuracy

Le modèle est précis à 98 % pour faire une bonne prédiction

Precision

Le modèle est précis à 97 % lorsqu'il indique qu'un billet est faux.

Recall

Le modèle a correctement classé 98 % des échantillons positifs

ONCFM

ACP / K-MEANS

Résumé de la partie

Les deux nuages de points en fonction de la variable is_genuine sont bien identifiés

En toute logique 2 clusters semblent la meilleure solution

Quelques vrais billets (5) considérés comme faux par le k-means

La classification par K-means est fidèle à 98%

Une marge d'erreur de 2%

ONCFM

RÉGRESSION LOGISTIQUE

La regression logiqitique est souvent utilisé pour la classification et l'analyse prédictive. La régression logistique estime la probabilité qu'un événement se produise, tel que voter ou ne pas voter, sur la base d'un ensemble de données donné de variables indépendantes.

Generalized Linear Model Regression Results

Dep. Variable:	is_genuine	No. Observations:	1200
Model:	GLM	Df Residuals:	1193
Model Family:	Binomial	Df Model:	6
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-29.117
Date:	Wed, 18 Oct 2023	Deviance:	58.234
Time:	15:42:00	Pearson chi2:	4.14e+03
No. Iterations:	11	Pseudo R-squ. (CS):	0.7037
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-188.0712	290.214	-0.648	0.517	-756.879	380.737
diagonal	-0.4782	1.343	-0.356	0.722	-3.111	2.154
height_left	-1.3153	1.237	-1.063	0.288	-3.741	1.110
height_right	-2.5342	1.238	-2.046	0.041	-4.961	-0.107
margin_low	-6.6439	1.291	-5.147	0.000	-9.174	-4.114
margin_up	-11.1034	2.768	-4.011	0.000	-16.529	-5.678
length	6.5435	1.144	5.720	0.000	4.301	8.786

L'équation de la droite de régression est capable de déterminer 70,37% de la distribution des points

'diagonal' et height_left' sont non significatives car p_ valeurs supérieures à 5 %

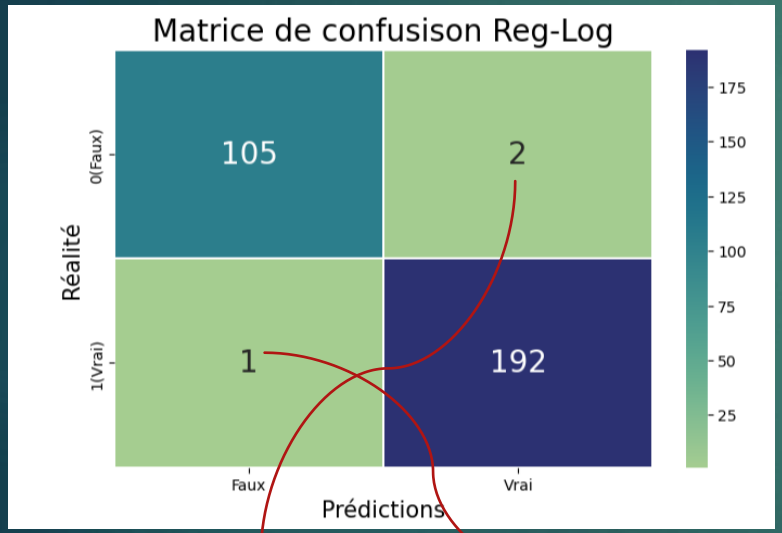
Après avoir retirer les deux variables non significatives (à savoir 'diagonal' et 'height_left') nous avons procéder au test et fait apparaitre la matrice de confusion.

ONCFM

RÉGRESSION LOGISTIQUE

Matrice de confusion et mesure de la performance

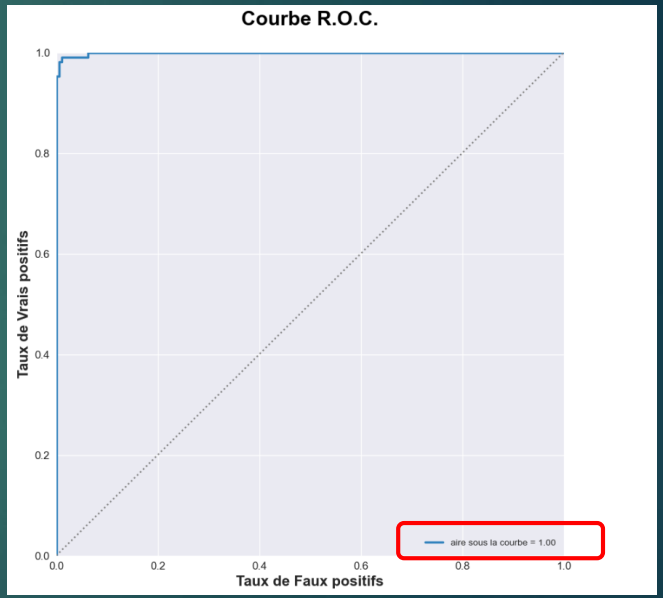
La courbe ROC est une représentation graphique des performances de tout modèle de classification à tous les seuils de classification.



Accuracy = 99 %

Precision = 99 %

Recall = 99 %



2 Vrais billets parmi les faux billets

1 Faux billets parmi les vrais billets

MODÈLE PRESQUE PARFAIT

ONCFM

RÉGRESSION LOGISTIQUE

Résumé de la partie

R2 est significatifs malgré la suppression variables non significatives

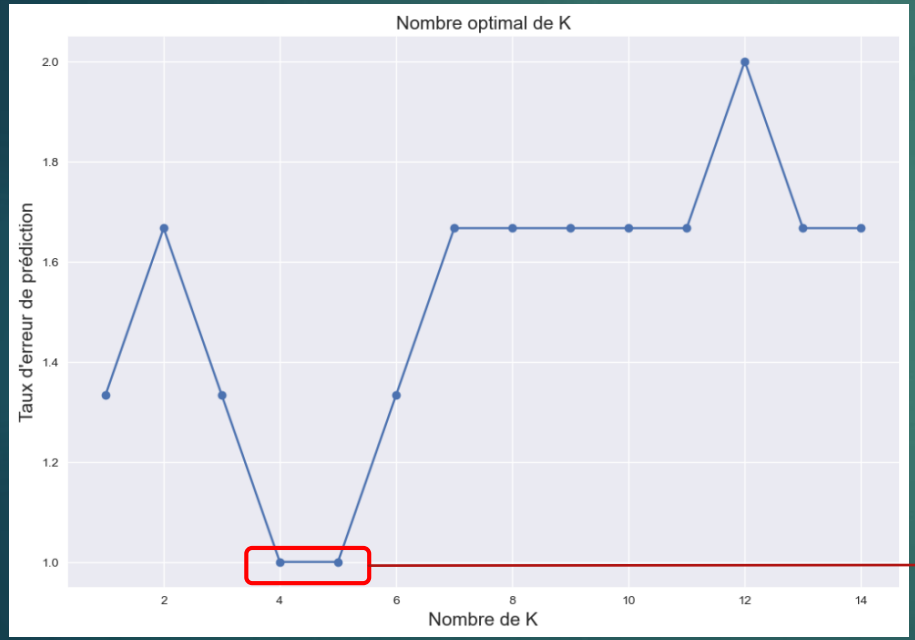
Les métriques et courbe R.O.C montre un modèle preque parfait

ONCFM

K Nearest Neighbors (KNN)

Choisir le nombre optimal de K

C'est un algorythme qui peut servir autant pour la classsification que la regression. Il est surnommée Nearest Neighbors car le principe de ce modèle consiste à choisir les K données les plus proches du point etudié afin d'en prédire les valeurs



	Erreur en pourcentage	nombre de k
3	1.000000	4
4	1.000000	5
0	1.333333	1
2	1.333333	3
5	1.333333	6

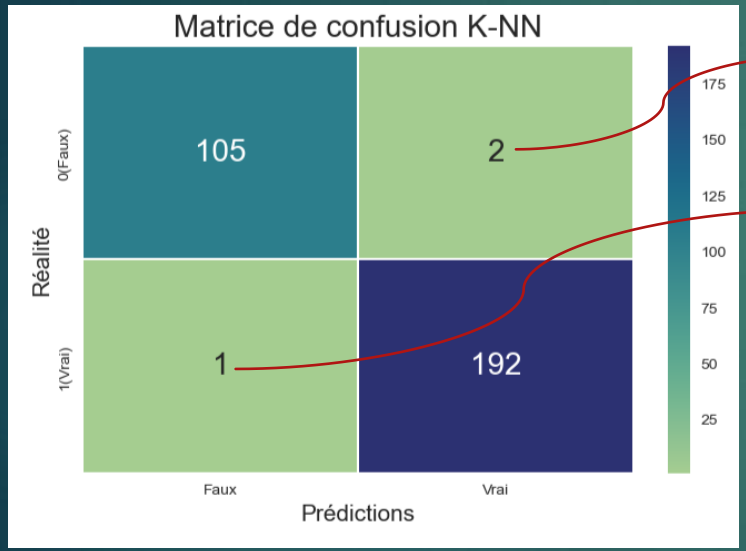
KNN le plus performant = 4 ou 5

ONCFM

Le KNN

Matrice de confusion et métriques

La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification



2 Vrais billets parmi les faux billets

1 Faux billets parmi les vrais billets

Accuracy

Le modèle est précis à 99 % pour faire une bonne prédiction

Precision

Le modèle est précis à 99 % lorsqu'il indique qu'un billet est faux.

Recall

Le modèle a correctement classé 99 % des échantillons positifs

ONCFM

CONCLUSION SUR LES 3 MODELES

La précisison des modèles

	Précision en %
K-means	97.4619
kNN	98.9691
Régression logistique	98.9691

Le KNN et la Régression Logistique sont les modèles les plus performants (99%)

Le K-Means obtient un resultat satisfaisant malgré tout (97,5%)

ONCFM

TEST DE L'ALGORYTHME

Test des 3 modèles sur le jeu de données « billets production »

Fonction de détection les vrais des faux billets

```
def prog_vérification_billets(csv):
    billet_test= pd.read_csv(csv)
    billet_value=billet_test.drop('id', axis=1)
    billet_test['Prédiction RegLog'] = reg_log_multi.predict(billet_value) >= 0.5
    billet_test['Prob Faux']=(1-(reg_log_multi.predict(billet_value))).round(3)
    billet_test["Prédiction K-means"] = kmeans.predict(billet_value)
    billet_test['Prédiction K-means'].replace([1,0],[True,False], inplace=True)
    billet_test['Prédiction K-nn'] = knn.predict(billet_value)
    billet_test['Prédiction K-nn'].replace([1,0],[True,False], inplace=True)

    billets_predict = billet_test[['id','Prob Faux','Prédiction RegLog','Prédiction K-means','Prédiction K-nn']].set_index("id")
    return billets_predict
```

```
# Tester la fonction
prog_vérification_billets('billets_production.csv')
```

id	Prob Faux	Prédiction RegLog	Prédiction K-means	Prédiction K-nn
A_1	1.000	False	False	False
A_2	1.000	False	False	False
A_3	1.000	False	False	False
A_4	0.003	True	True	True
A_5	0.000	True	True	True

ONCFM

Merci pour votre attention