DÉTECTION AUTOMATIQUE DE FAUX BILLETS

CRÉATION D'UNE APPLICATION DE DÉTECTION EN MACHINE LEARNING

Gabriel Gwynn 08/2025

MISSION ONCFM – DÉTECTION AUTOMATIQUE DE FAUX BILLETS

- Mission confiée par l'Organisation Nationale de Lutte contre le Faux-Monnayage
- Rôle : développer un algorithme prédictif + application fonctionnelle
- Objectif : identifier rapidement la nature d'un billet à partir de ses caractéristiques

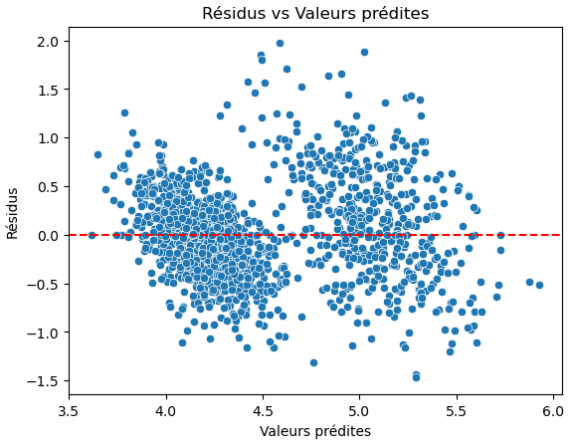
ENJEUX & CONTRAINTES

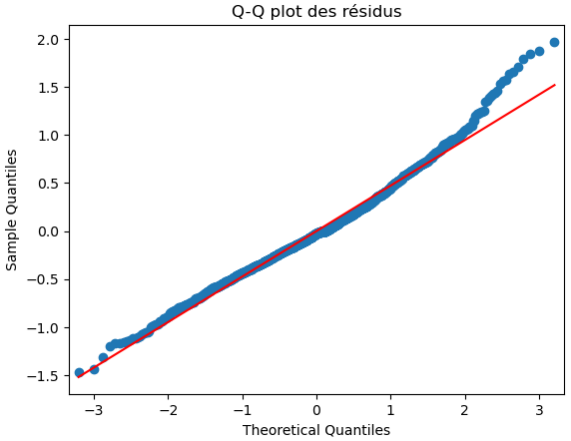
- Minimiser les faux négatifs = éviter que des faux billets passent pour vrais
- Assurer une détection rapide et fiable
- Intégration simple dans les processus opérationnels

JEU DE DONNÉES INITIAL

- 1500 billets scannés (1000 vrais, 500 faux)
- 6 caractéristiques géométriques :diagonal, height_left, height_right, margin_low, margin_up, length
- Cible : is_genuine (True = vrai billet, False = faux billet)

TRAITEMENT DES VALEURS MANQUANTES





- 37 valeurs manquantes sur margin_low
- Imputation via régression linéaire multiple
- Vérification des hypothèses de la régression :
  - Relation linéaire ✅
  - Indépendance des observations ✅ (supposé vrai car n'est pas une série temporelle)
  - Homoscedasticité ❌ non respecté mais permet quand même de faire une imputation car les coefficients de la régression restent non biaisés)
  - Normalité des erreurs ❌ non respecté , les résidues extrèmes ne respectent pas la loi normale
  - Absence de multicolinéarité ✅ (les Variance Inflation Factor de chaque variables sont inférieur à 5)

IMPUTATION ET COEFFICIENT DE DÉTERMINATION

- Le coefficient de détermination noté R^2 de notre modèle de régression est de 0,477
- Pour 1 => prédiction parfaite pour 0 => le modèle ne fait pas mieux qu'une moyenne
- 0,477 => notre modèle explique 47,7% de la variabilité de la variable
  - Ce n'est pas excellent mais suffisant pour simplement une imputation de valeurs manquantes.

- Cette imputation par régression n'étant pas parfaite, nous garderons deux dataframes, l'un avec les données imputées l'autre avec données originales emputés des lignes qui avaient pour margin_low null

TEST DE T STUDENT INDEPENDANT ET EFFET DE L'IMPUTATION

H0 (hypothèse nulle) : La moyenne (ou la distribution) de la variable est identique pour les vrais billets et les faux billets.
Pas de différence significative.

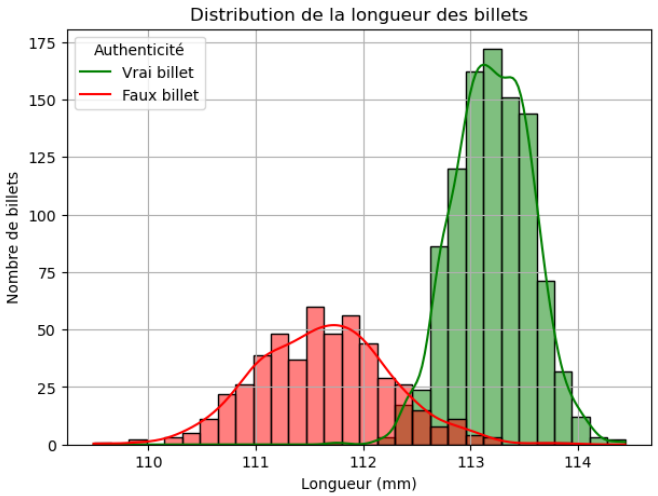
H1 (hypothèse alternative) : La moyenne (ou la distribution) de la variable est différente entre les vrais billets et les faux billets.
Il y a une différence significative.

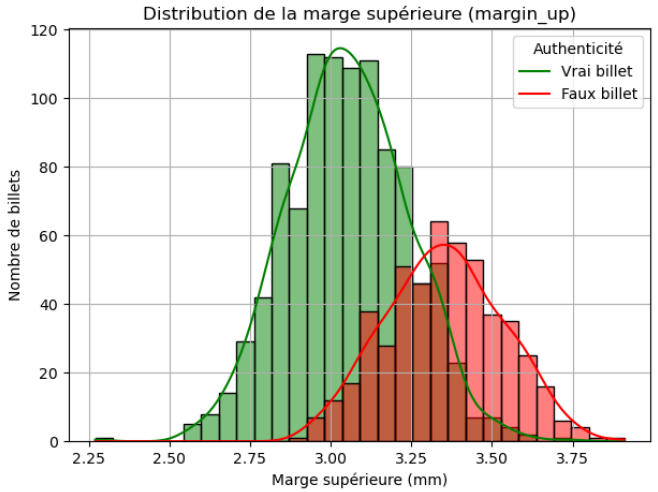
conclusion : peu importe l'imputation par régression linéaire ou la suppression des lignes avec valeurs manquantes, les p-values sont toutes extrêmement petites (≪ 0,05)

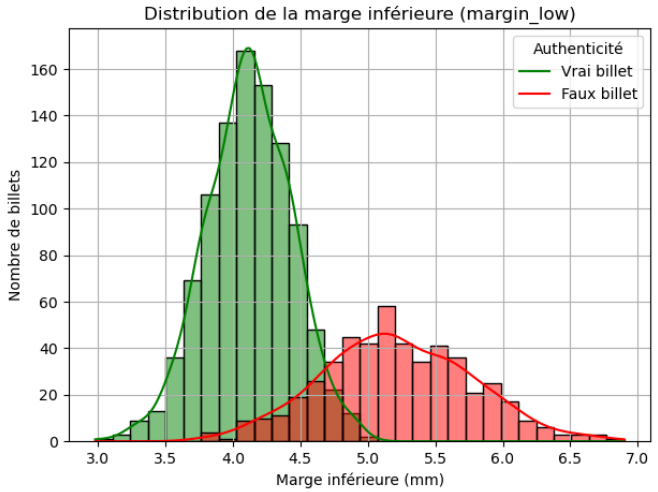
L'imputation n'a pas biaisé la significativité des tests. Toutes les variables sont pertinentes pour discriminer vrais/faux billets.

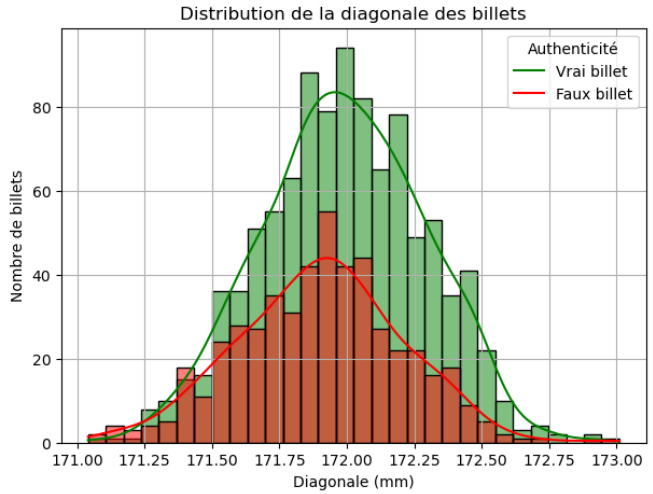
Variable / p-value	Jeu imputé	Jeu sans valeurs manquantes
Diagonal	3,1869e-0,7	2,7818e-07
Height_left	1,4154e-61	4,8516e-58
Height_right	9,2876e-89	8,3481e-87
Margin_low	6,5485e-186	1,4332e-182
Margin_up	2,9274e-141	1,3451e-140
length	1,4700e-241	1,4202e-237

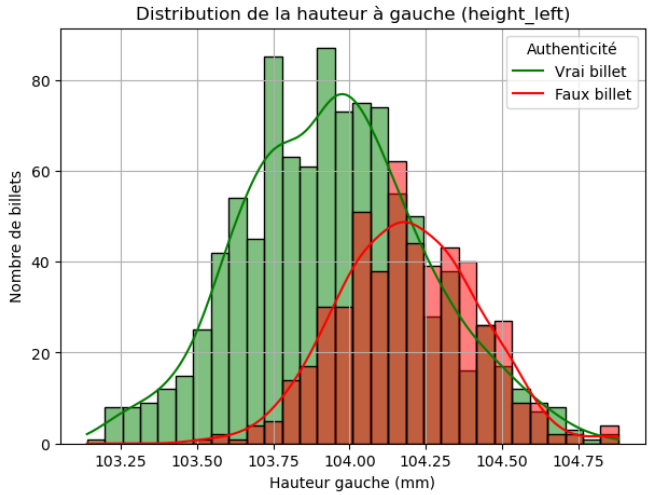
ANALYSE EXPLORATOIRE

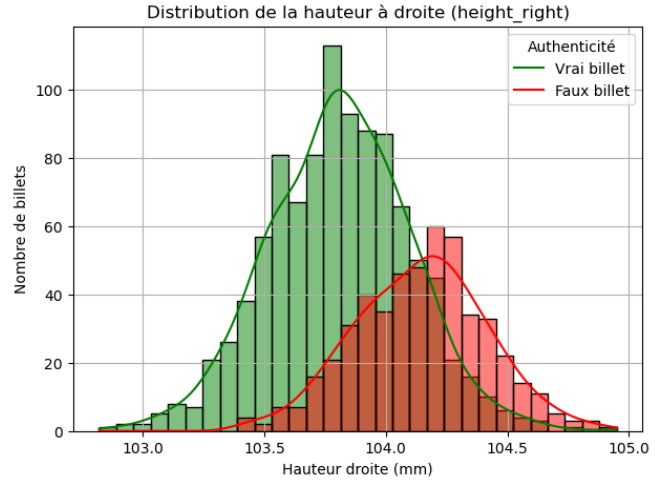






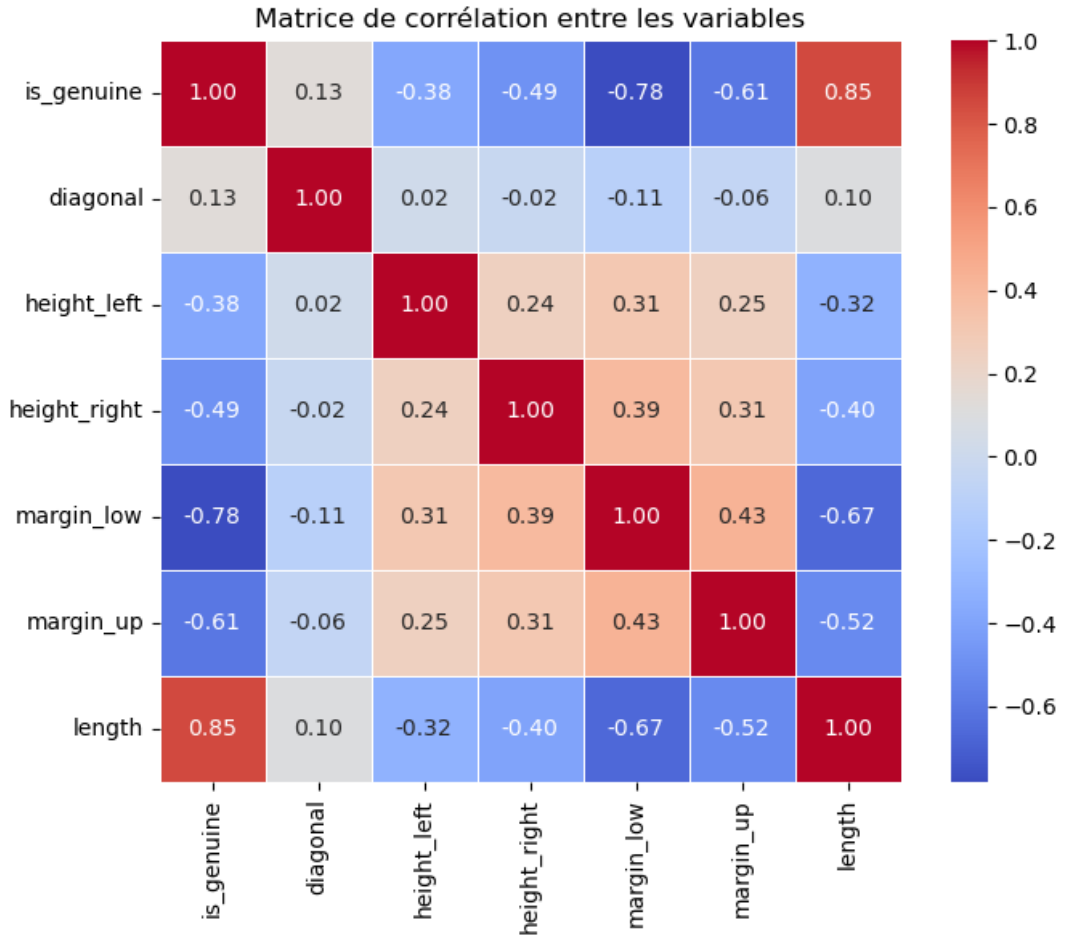






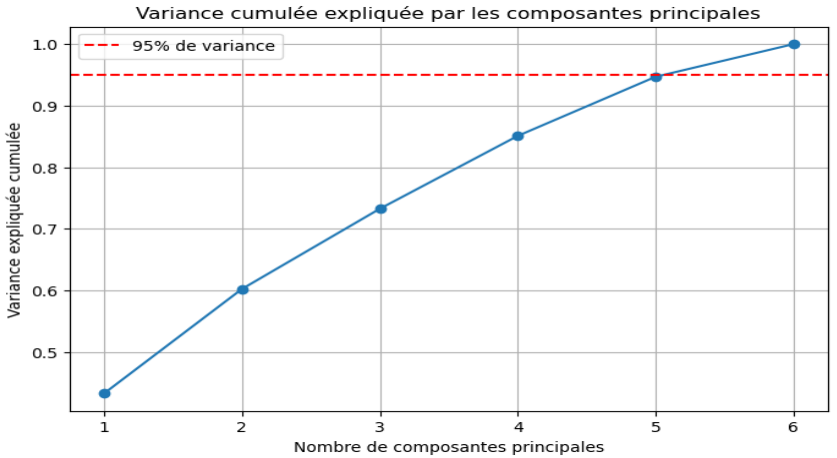
MATRICE DE CORRÉLATION

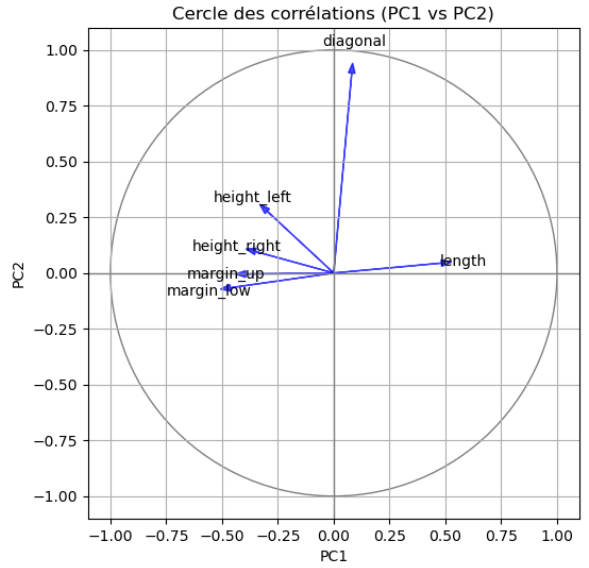
- Les variables prédictives pour détecter l'authenticité des billets sont par ordre d'importance :
  length (0.85)
  margin_low (-0.78)
  margin_up (-0.61)
  height_right (-0.49)
  height_left (-0.38)
  diagonal (+0.13)
- Top 3 des correlation entre variable :
  margin_low et length (-0.67)
  margin_up et length (-0.52)
  margin_low et margin_up (+0.43)
- Attention a la colinéarité pour les regression logistique, preferer les modèles en arbe



TEST DE COLINÉARITÉ ET ACP

- Calcule du VIF => Problème : Tous les VIF sont hors limites acceptables. Les variables sont très corrélées entre elles, ce qui empêche d'utiliser une régression linéaire ou logistique de façon fiable sans traitement.
- Solution : Passer par une ACP car moins sensible à la colinéarité
- On retiendra 5 composantes principales pour expliquer presque 95% de la variance
- PC1 représente le gabarit global du billet
- PC2 représente la mesure de la géométrie oblique



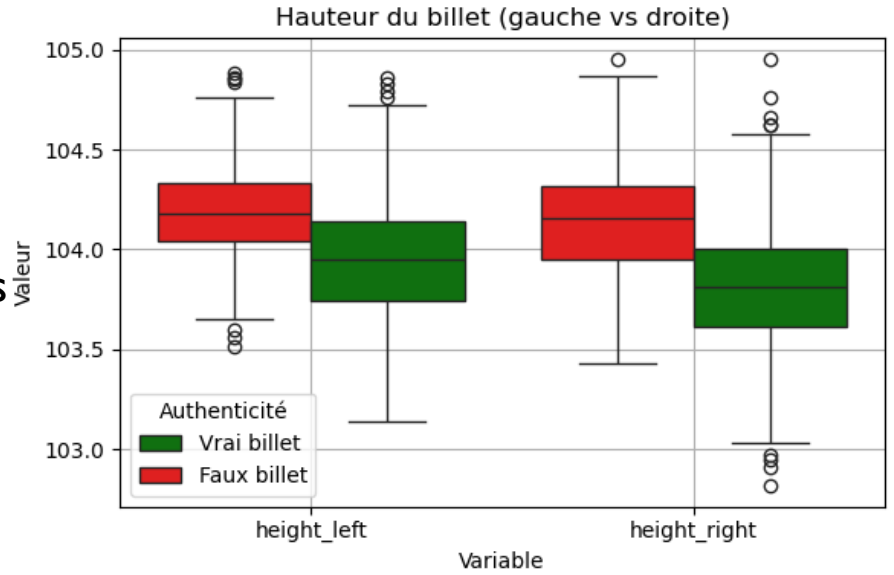


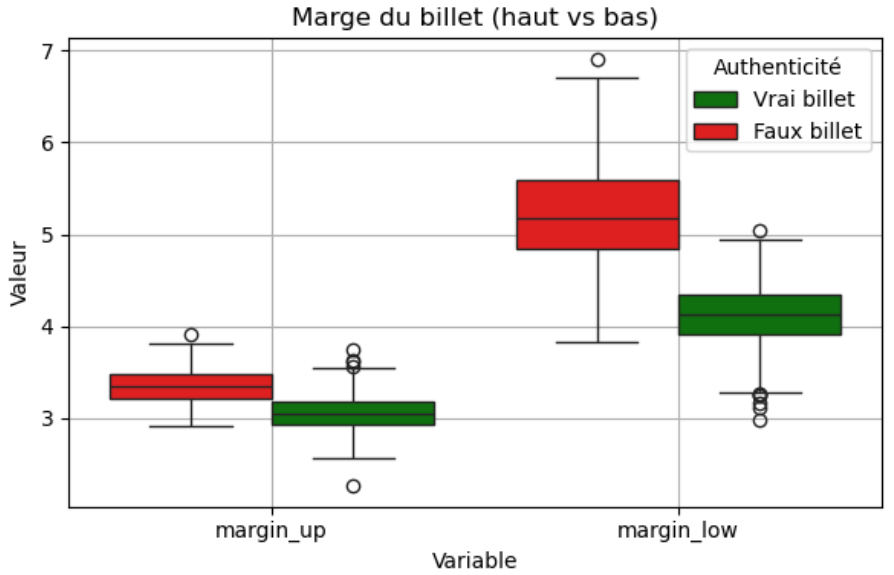
BOXPLOT : DISTRIBUTION DES VARIABLES ET VISUALISATION OULIERS

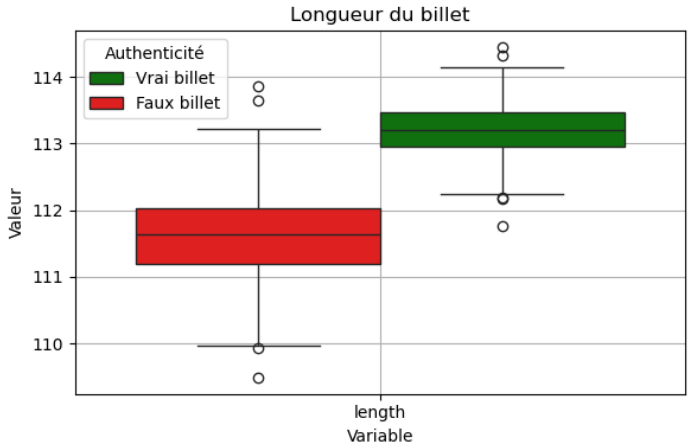
Bilan : on remarque des différences entre les vrais et faux billet dans chacune des catégories particulierement marqué en length et margin

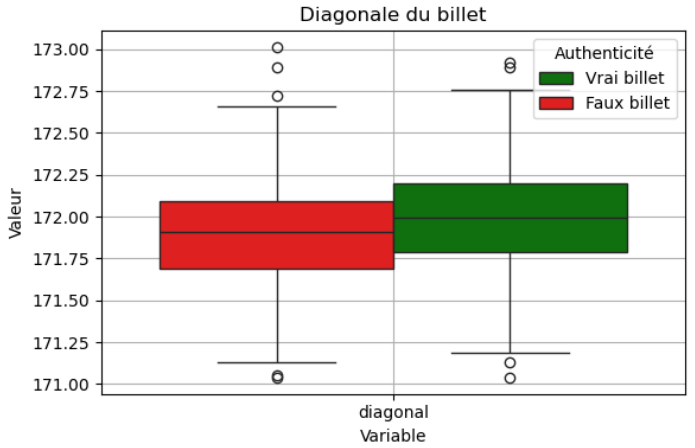
il semble également y avoir des outliers dans chacune des catégories

Note : modèles robuste aux outliers (arbres de decision, random forest) modèles sensibles aux outliers (regression, KNN...)









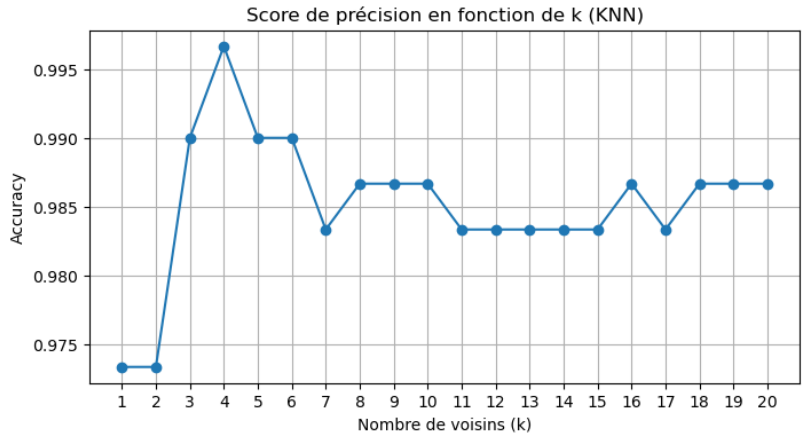
DISTRIBUTION DES OUTLIERS

- Pour margin_low et length les outliers sont un indicateur de faux billets, il est important de les garder
- Cependant pour les autres variables ils sont plutôt répartis entre les deux classes.
- Décision de garder les outliers car il ne sont ni nombreux ni concentrés sur une classe. Ils reflètent certainement une variabilité normal des billets. Les supprimer pourrait créer un biais ou réduire la diversité réelle des données.
- Ces indices nous invites à préferer par la suite une modélisation robuste.

	Total	Vrai billets	Faux billets
diagonal	7	3	4
height_left	6	3	3
height_right	11	7	4
margin_low	24	0	24
margin_up	3	1	2
length	3	0	3

ALGORITHMES RETENUS

- Régression logistique (avec ACP)
- KNN (ACP, k=4)
- Random Forest (avec et sans ACP)
- K-means (clustering non supervisé)



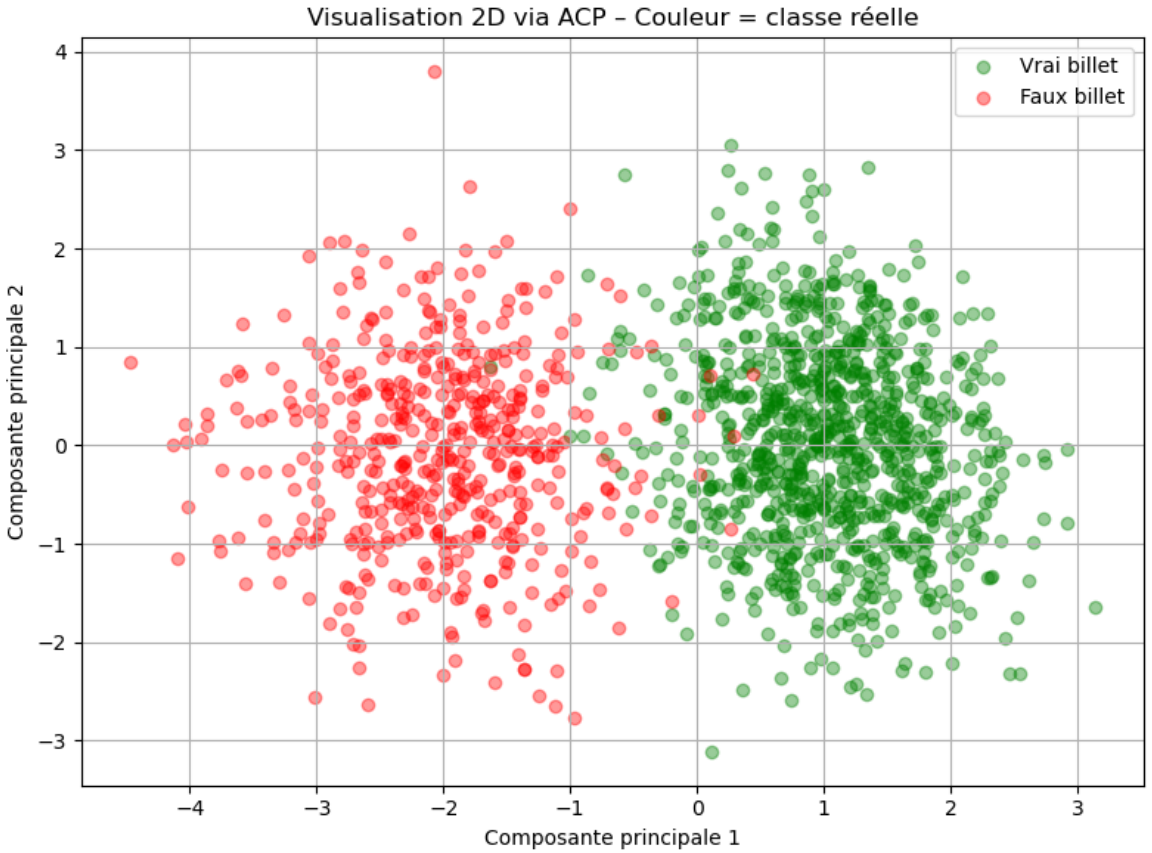
- Les scores de performances sont excellents pour la regression logistique , le KNN et le random forest. Il est légèrement moins pour pour le K-means. Nous utiliserons le k-means surtout dans un but exploratoire.

- Une analyse plus pousser via une validation croisée sera effectué sur les modèle de régression logistique, KNN et random forest

K-MEANS

- Score ARI : 0,94 (proche de 1 cad très bon)
- Silhouette score : 0,34 (groupe bien déterminé mais avec limites aux frontières)

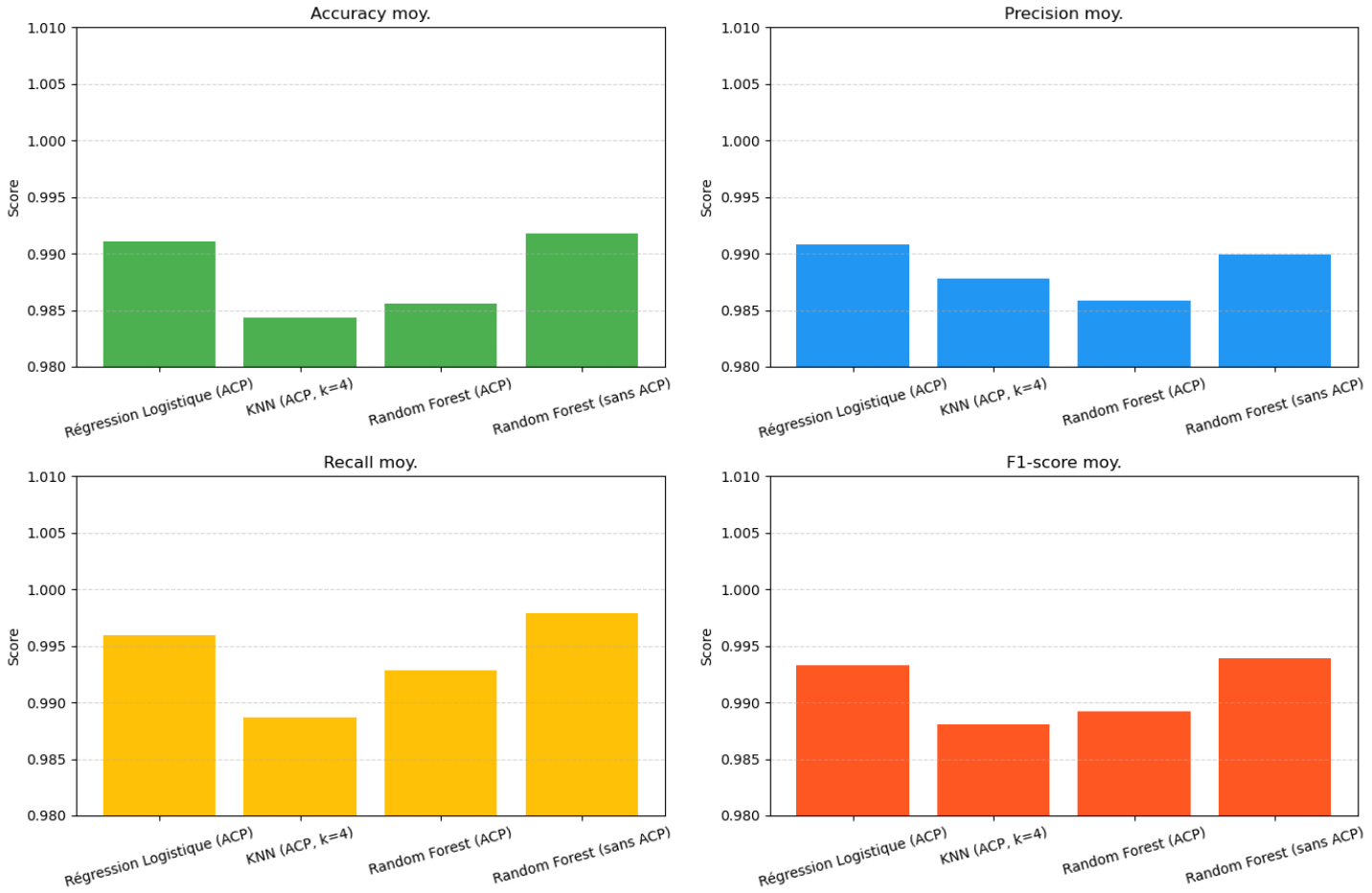
- Ces résultats confirment que les caractéristiques mesurées discriminent efficacement les deux types de billets, même sans supervision, renforçant la confiance dans la capacité des modèles supervisés à généraliser sur ce problème.



VALIDATION CROISÉE ET SCORES

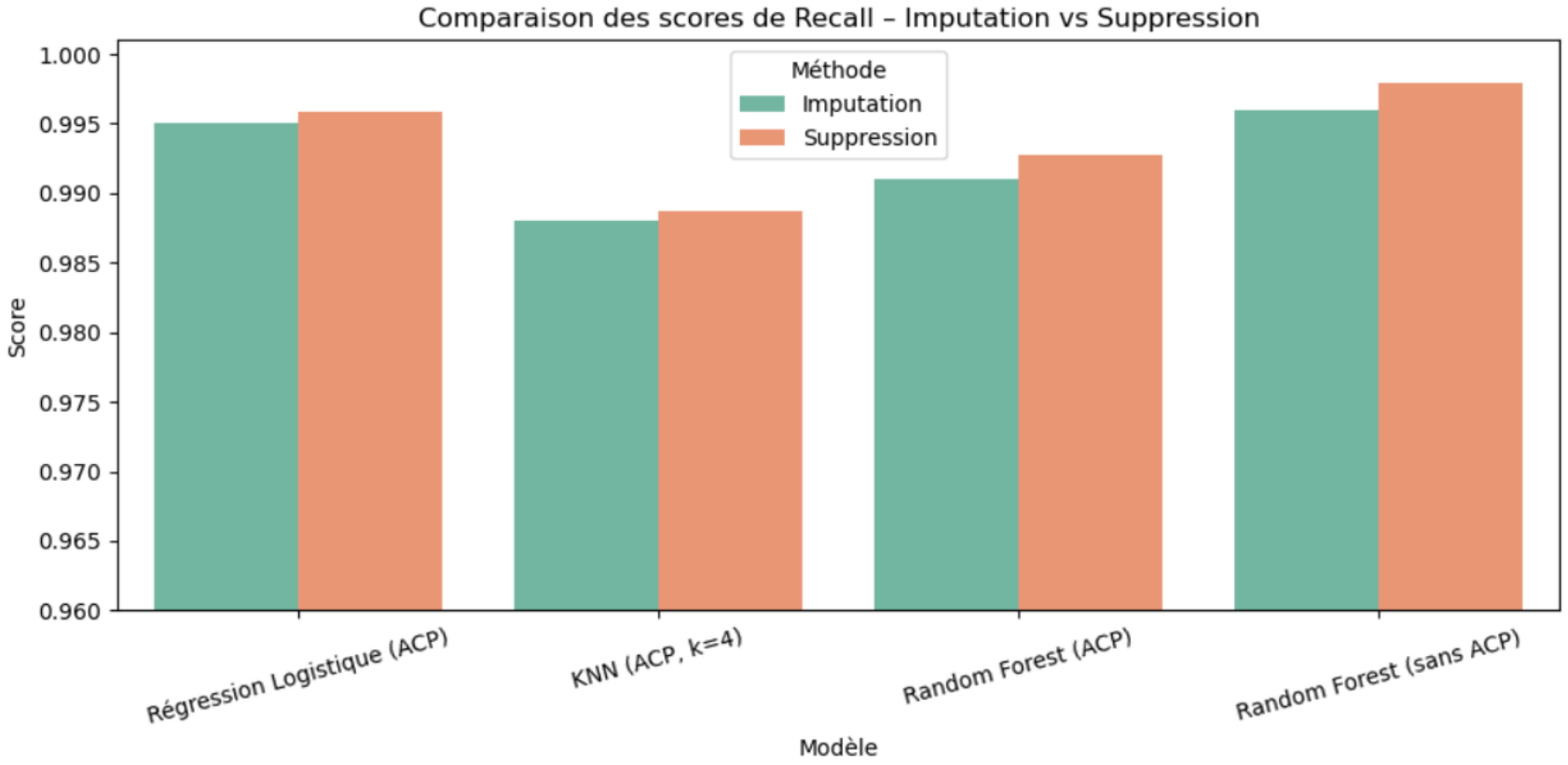
- Accuracy : Part des bonnes prédictions sur l'ensemble des observations
- Precision: Parmi les billets prédits comme faux, combien sont réellement faux ?
- Recall : Parmi les billets vraiment faux, combien ont été bien détectés comme tels ?
- F1-score : Moyenne harmonique entre précision et rappel (équilibre entre les deux)

 Scores moyens par modèle (validation croisée 5-fold)



CHOIX DE LA DATA : IMPUTATION VS SUPPRESSION

- Meilleure performance avec entrainement sur les données avec les lignes à margin_low null supprimées.
- Meilleur performance en accuracy et f1 score.
- Exception pour la precision qui est meilleure à ~0,002



CONCLUSION CHOIX DU MODÈLE = RANDOM FOREST SANS ACP

- Le modèle Random Forest sans ACP a été retenu car :
- il offre un excellent compromis entre précision, rappel et robustesse. Avec un recall de 99,79 %, il minimise drastiquement les faux négatifs (c'est-à-dire les faux billets classés à tort comme vrais), ce qui est critique dans un contexte de détection de fraude. Par ailleurs, il conserve une précision élevée (98,99 %), ce qui limite aussi les faux positifs, évitant ainsi de rejeter des billets authentiques.
- L'absence d'ACP permet de conserver l'intégralité de l'information portée par les variables d'origine, ce qui améliore l'interprétabilité des résultats et facilite la mise en production. Enfin, la nature même de la Random Forest lui confère une grande robustesse face aux données bruitées et aux valeurs extrêmes, assurant des performances stables sur de nouveaux jeux de données.
- En résumé : la Random Forest sans ACP est non seulement la plus performante mais aussi la plus robuste, et donc le meilleur choix pour un déploiement opérationnel dans ce contexte.

ANNEXE 1 : RANDOM FOREST

```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# 1. Définir les variables explicatives et cible
features = ['diagonal', 'height_left', 'height_right', 'margin_low', 'margin_up', 'length']
X = df_complete[features]
y = df_complete['is_genuine']

# 2. Séparation train/test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

# 3. Entraînement du modèle Random Forest
rf_complete = RandomForestClassifier(n_estimators=100, random_state=42)
rf_complete.fit(X_train, y_train)

# 4. Prédiction
y_pred_complete = rf_complete.predict(X_test)

# 5. Rapport de classification
print("📊 Rapport de classification - Random Forest sur df_complete :")
print(classification_report(y_test, y_pred_complete, target_names=["Faux billet", "Vrai billet"]))

# 6. Matrice de confusion
cm = confusion_matrix(y_test, y_pred_complete)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Faux billet", "Vrai billet"])
disp.plot(cmap="Greens")
plt.title("Matrice de confusion - Random Forest (df_complete)")
plt.grid(False)
```

ANNEXE 2 : APPLICATION

```python
# 4.1 Probabilités de "vrai billet" (classe True) -> colonne 1 si y est booléen
proba_true = pipe.predict_proba(df_feat)[:, 1]

# 4.2 Classe booléenne (seuil 0.5 par défaut)
pred_bool = proba_true >= 0.5

# Seuil de décision : 0.5 par défaut. Si, métier, on veut encore moins de faux négatifs,
# on peut baisser le seuil (ex. 0.45) pour être plus “strict” sur la détection des faux.

# 4.3 Étiquette lisible
pred_label = np.where(pred_bool, "Vrai billet", "Faux billet")

# 4.4 Réattacher au DataFrame d'entrée filtré
df_out = df_clean_in.copy()
df_out["proba_true"] = proba_true.round(4)
df_out["prediction_bool"] = pred_bool
df_out["prediction_label"] = pred_label

print("Aperçu des prédictions :")
display(df_out.head())
```

Aperçu des prédictions :

	diagonal	height_left	height_right	margin_low	margin_up	length	id	proba_true	prediction_bool	prediction_label
0	171.76	104.01	103.54	5.21	3.30	111.42	A_1	0.0033	False	Faux billet
1	171.87	104.17	104.13	6.00	3.31	112.09	A_2	0.0000	False	Faux billet
2	172.00	104.58	104.29	4.99	3.39	111.57	A_3	0.0000	False	Faux billet
3	172.49	104.55	104.34	4.44	3.03	113.20	A_4	0.9733	True	Vrai billet
4	171.65	103.63	103.56	3.77	3.16	113.33	A_5	1.0000	True	Vrai billet