PROJET 10 : Détectez des faux billets avec R ou Python

Shana Husejnovic

ONCFM

Présentation du contexte

- L'Organisation nationale de lutte contre le faux-monnayage, ou **ONCFM**, est une organisation publique ayant pour objectif de mettre en place des méthodes d'identification des contrefaçons des billets en euros. Dans le cadre de cette lutte, ils souhaitent mettre en place un algorithme qui soit capable de différencier automatiquement les vrais des faux billets.
- Je suis **consultante Data Analyst** dans une entreprise spécialisée dans la data. Mon entreprise a décroché une prestation en régie au sein de l'Organisation nationale de lutte contre le faux-monnayage (ONCFM).
- **OBJECTIF** : construire un algorithme qui, à partir des caractéristiques géométriques d'un billet, serait capable de définir si ce dernier est un vrai ou un faux billet en respectant le cahier des charges qui demande l'utilisation de python ou R, et d'utiliser deux méthodes de prédiction : kmeans et la régression logistique.

ONCFM

Présentation des données

- 1 fichier csv nommé 'billets'
- Nous disposons de six informations géométriques sur un billet :
  - 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)

  Nous disposons également de la variable is_genuin qui nous permet de savoir si le billet est vrai ou faux.

ONCFM

Exploration des données : python

- On remarque des valeurs manquantes pour la variable margin_low
- Nous devons nous interroger sur le traitement de ces valeurs manquantes.
- Nous choisissons donc de les remplacer plutôt que de les supprimer, mais comment ?

```
<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
dtypes: bool(1), float64(6)
memory usage: 71.9 KB
```

ONCFM

Valeur manquantes margin_low : python

- Nous avons décidé d'utiliser des modèles d'apprentissage automatique pour remplacer les valeurs manquantes :
  - La régression linéaire
  - La régression linéaire robuste
  - Le KNN
  - Le MLE
- Nous avons ensuite comparé les différents modèles en utilisant l'évaluation de performance et fait une comparaison des différences statistiques entre les données de bases et les données remplacées par chaque modèle :
  - Nous avons opté pour le modèle MLE

ONCFM

Modèles de prédiction

- Le cahier des charges nous demandait d'utiliser la régression logistique et l'algorithme de clustering du kmeans. Toutefois le modèle du kmeans n'étant pas vraiment adapté à la prédiction d'une sortie de type booléenne ou binaire nous avons décidé d'aller plus loin.
- Nous avons donc décidé de comparer plusieurs modèles :
  - Kmeans
  - Regression logistique
  - KNN
  - Random Forest
  - Réseau de neurones
- Tout comme pour la prédiction des valeurs manquantes nous avons effectué des tests pour évaluer la performance des différents modèles et nous avons opté pour le modèle **Random Forest**

ONCFM