DataScientest

Challenge Rakuten

Rapport d’exploration et de data visualisation

Présenté par l’équipe projet :

CHELOUAH Slimane
LOLO MVOUMBI Simplice
MORMICHE Nicolas
ZIDI Riadh

Table des matières

1 - Introduction du projet

1.1 - Contexte

Ce projet s'inscrit dans le cadre d'un challenge Rakuten qui porte sur le thème de la classification multimodale (texte et image) de produits à e-commerce à grande échelle où l'objectif est de prédire le code type de chaque produit tel que défini dans le catalogue de Rakuten France. D'un point de vue économique, le Multimodal Learning est une technique utile pour les e-commerces car il permet de catégoriser automatiquement des produits en fonction des images et des informations fournies par les commerçants.

1.2 - Périmètre

Étude des classificateurs multimodales qui peuvent être adaptés à la détermination de catégorie de produit à partir d'image et de texte. On se limite ici aux données fournies par Rakuten concernant les produits et leurs images associées afin de vérifier son efficacité.

Nous prendrons comme base le modèle de référence fourni par Rakuten :

- Pour les données d'image, une version du modèle *Residual Networks (ResNet)*
- Pour les données texte, un classificateur CNN simplifié

1.3 - Objectifs

Nous devons fournir un fichier y_test.csv, dans le même format que le fichier y_train.csv, en associant chaque identifiant produit ("Unnamed: 0") au code type ("prdtypecode") associé :

1	Unnamed: 0	Un identifiant entier pour le produit. Cet identifiant permet d'associer le produit à son code type de produit correspondant.
2	prdtypecode	Code type de produit (catégorie)

On utilisera ici un classificateur multimodal pour rattacher un produit à son code type où les données à disposition comportent 27 catégories pour 84916 produits et 84916 images.

Rakuten ayant déjà réalisé une classification multimodale à l'aide de divers modèles, le but est de proposer une alternative en testant différents modèles (ex : CNN ou autre) afin d'obtenir un meilleur score F1 pondéré que celui proposé par Rakuten actuellement :

- Classification des textes avec CNN : 0.8113
- Classification des images avec ResNet50 : 0,5534

1.4 - Notions importantes

Le Multimodal Learning est une notion centrale du projet. Il existe plusieurs méthodes pour fusionner les informations textuelles et visuelles :

- Les réseaux de neurones multimodaux
  (modèle de fusion CNN / RNN / TNN)

- L'apprentissage par transfert
  (transférer des connaissances apprises d'une modalité à une autre)

- Les techniques de co-apprentissage ou d'apprentissage adversarial (représentation des données multimodales)

- Les réseaux de neurones auto-encodeurs
  (reconstruire les données d'entrée en passant par une représentation latente)

Concernant la classification des données multimodales (textes et images), nous aurons plusieurs choix de modèle à traiter afin de vérifier le score F1 pondéré de chacun :

- Classification des textes (ex : BERT ou CNN)

- Classification des images (ex : CNN, RNN ou TNN)

Pour former et entraîner un modèle de Multimodal Learning, on peut utiliser des techniques de Deep Learning tel que les réseaux de neurones où l'on utilise des encodeurs :

- Former un modèle multimodal à double encodeur (ex : BERT et ResNet-50)

- Entraîner un modèle multimodal à encodeur commun (ex : ALBEF)

2 - Exploration des données

Rakuten met à disposition les datasets X_train, y_train et X_test (sans les données targets y_test correspondantes). Nous avons donc décidé de fusionner X_train et y_train pour l'exploration des données.

- **Importer les fichiers CSV**

```
df_X = pd.read_csv("X_train.csv", sep = ',', index_col = 0)
df_y = pd.read_csv("Y_train.csv", sep = ',', index_col = 0)
```

- **Fusionner les DataFrames X (variables explicatives) et y (variable cible)**

```
fusion = pd.merge(df_X, df_y, left_index=True, right_index=True)
```

- **Afficher les premières lignes du DataFrame**

```
fusion.head()
```

- **Afficher les informations du DataFrame**

```
fusion.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 84916 entries, 0 to 84915
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   designation  84916 non-null  object
 1   description  55116 non-null  object
 2   productid    84916 non-null  int64 
 3   imageid      84916 non-null  int64 
 4   prdtypecode  84916 non-null  int64 
dtypes: int64(3), object(2)
memory usage: 5.9+ MB
```

- **Afficher les statistiques descriptives des variables quantitatives**

```
fusion.describe()
```

	productid	imageid	prdtypecode
count	8.491600e+04	8.491600e+04	84916.000000
mean	2.555468e+09	1.152691e+09	1773.219900
std	1.588656e+09	1.751427e+08	788.179885
min	1.839120e+05	6.728400e+04	10.000000
25%	6.760519e+08	1.056269e+09	1281.000000
50%	3.190506e+09	1.213354e+09	1920.000000
75%	3.995599e+09	1.275646e+09	2522.000000
max	4.252012e+09	1.328824e+09	2905.000000

- **Afficher les valeurs uniques prises par prdtypecode :** 27 catégories au total

```
fusion.prdtypecode.nunique()
```

- **Vérifier les doublons :** aucun doublon

```
fusion.duplicated().sum()
```

- **Vérifier les valeurs nulles :** description compte 29800 NaN sur 84916 soit 35%

```
fusion.isna().sum()
```

```
designation        0
description    29800
productid          0
imageid            0
prdtypecode        0
dtype: int64
```

- **Exploration des données**

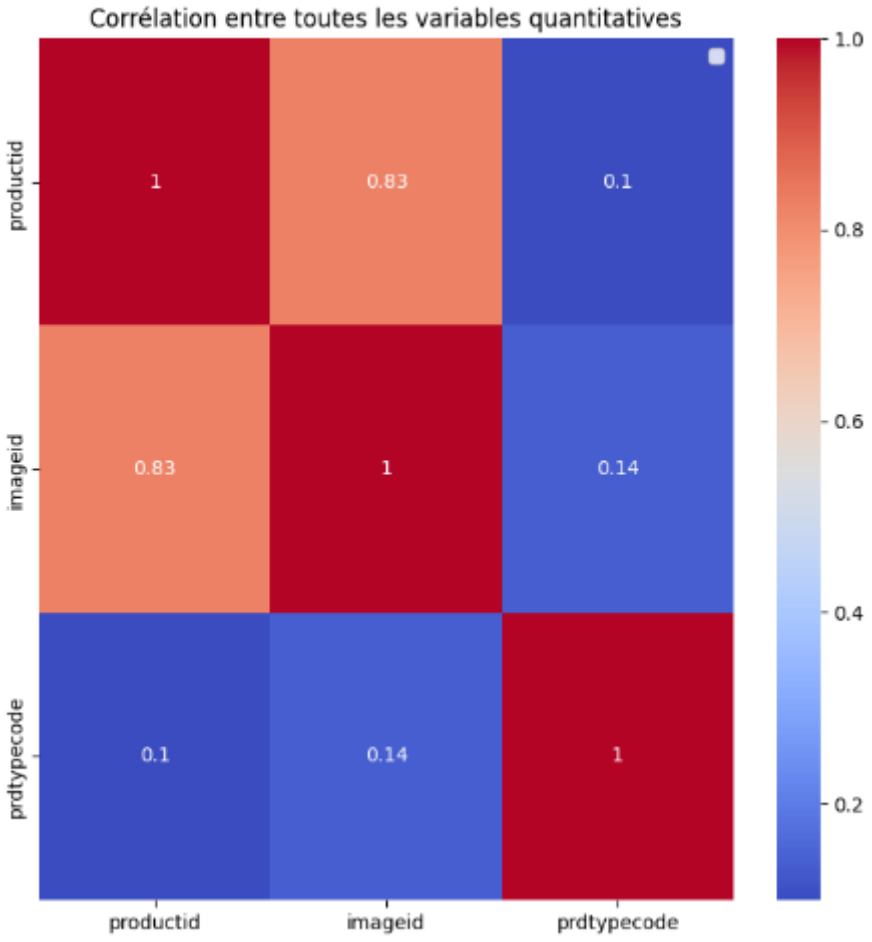
Vous trouverez le tableau d'exploration des données ici : exploration des données

- Onglet Fusion pour les données texte
- Onglet X_train_images pour les données image

3 - Visualisation des données

Heatmap : corrélation entre toutes les variables quantitatives

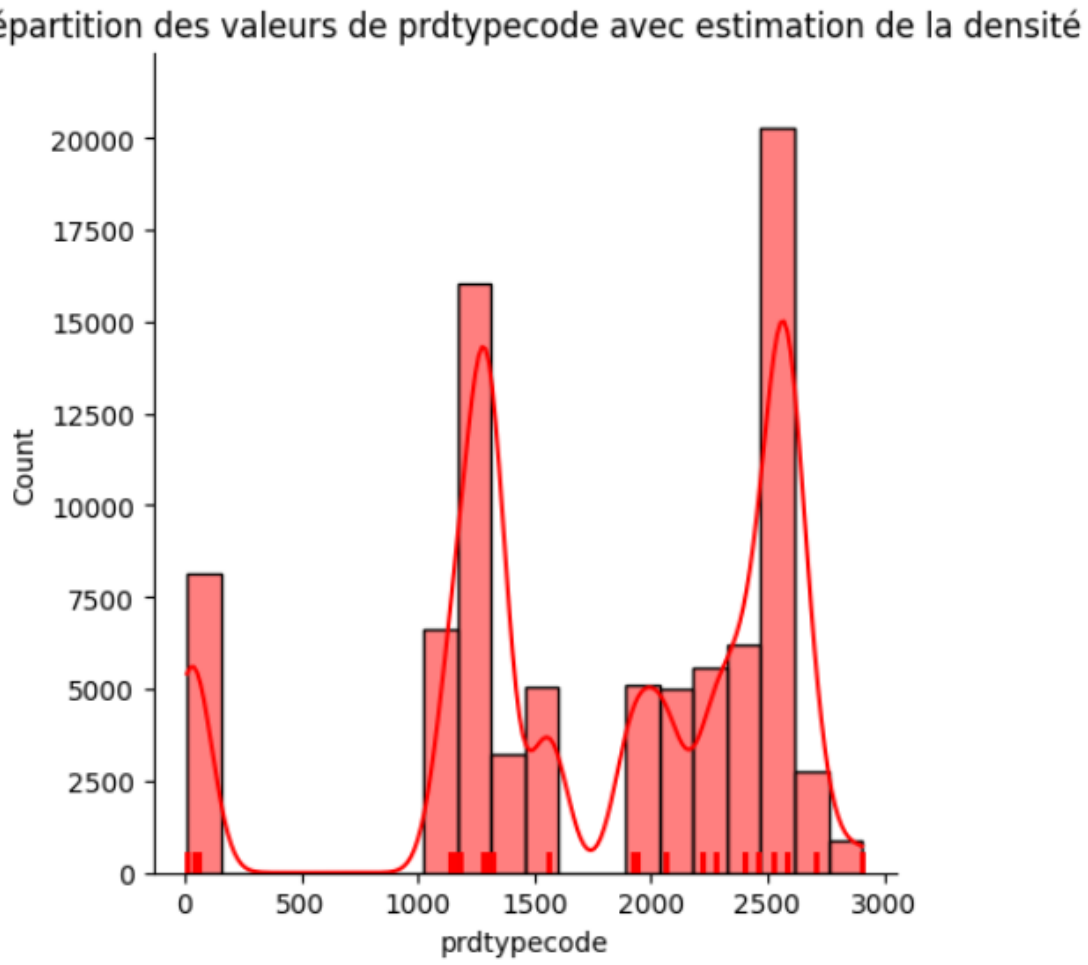
```
fusion = fusion.drop(['description'], axis=1)
fusion = fusion.drop(['designation'], axis=1)
fig, ax = plt.subplots(figsize = (8,8))
sns.heatmap(fusion.corr(), ax = ax, cmap = "coolwarm", annot=True)
plt.title('Corrélation entre toutes les variables quantitatives')
plt.savefig("reports/figures/heatmap.png", bbox_inches='tight')
```



Conclusion : on ne retrouve aucune corrélation intéressante entre les variables quantitative mis à part entre productid et imageid qui ne semble pas être significatif pour le projet

Histogramme avec estimation de la densité : prdtypecode

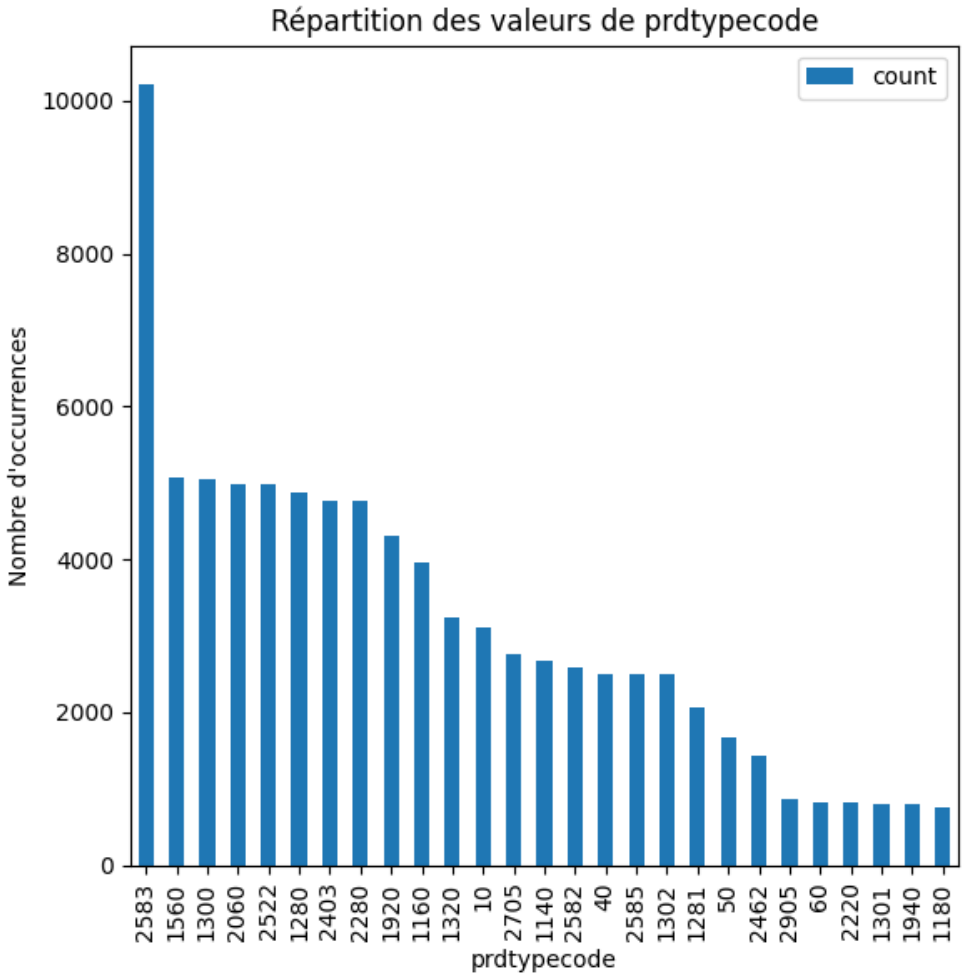
```
sns.displot(fusion.prdtypecode, bins=20, kde = True, rug=True,
color="red")
plt.title('Répartition des valeurs de prdtypecode avec estimation de la
densité')
plt.savefig("reports/figures/histogramme_avec_estimation_densite.png",
bbox_inches='tight')
```



Conclusion : on constate que les valeurs de codes type produits se répartissent sur 3 plages de valeurs principales.

Histogramme : prdtypecode

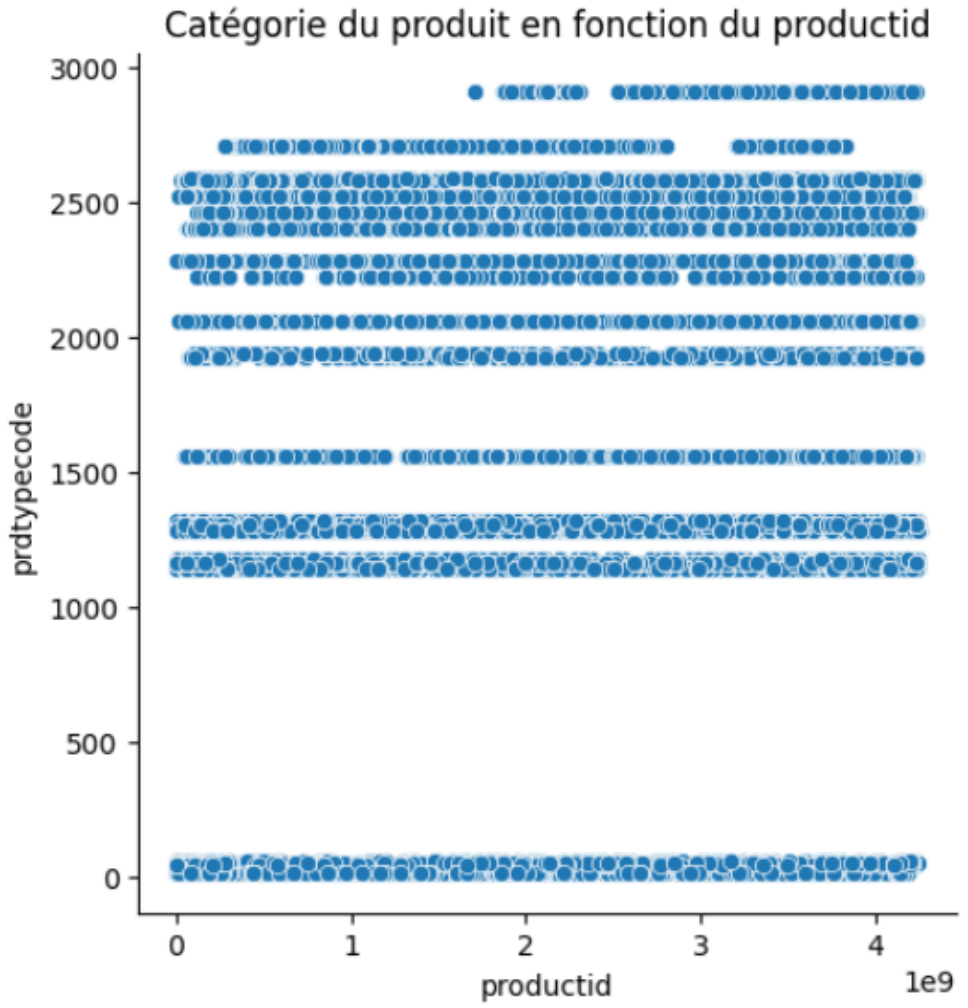
```
distribution = fusion['prdtypecode'].value_counts()
distribution_df = distribution.reset_index()
distribution_df.columns = ['prdtypecode', 'count']
distribution_df.plot(kind='bar', x='prdtypecode', y='count')
plt.xlabel('prdtypecode')
plt.ylabel('Nombre d\'occurrences')
plt.title('Répartition des valeurs de prdtypecode')
plt.savefig("reports/figures/historamme.png", bbox_inches='tight')
```



Conclusion : graphe qui reprend les représentations des 27 catégories de type produits ordonnés par ordre décroissants. La catégorie 2583 se détache fortement en terme de représentation.

Nuage de point : productid et prdtypecode

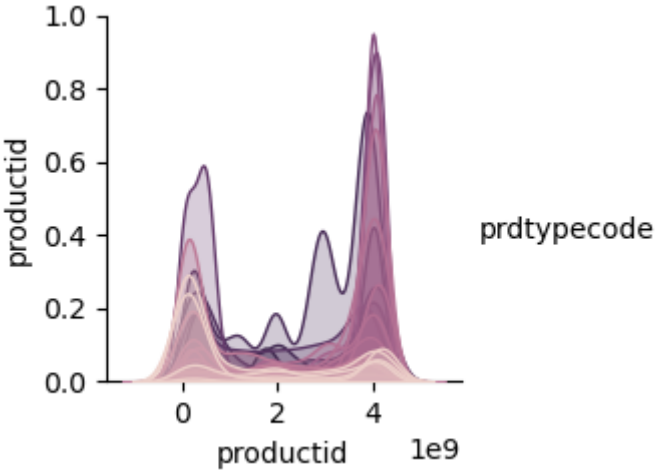
```
sns.relplot(x=fusion.productid, y=fusion.prdtypecode)
plt.title('Catégorie du produit en fonction du productid')
plt.savefig("reports/figures/scatterplot.png", bbox_inches='tight')
```



Conclusion : les code type produits se répartissent par plage discrètes de valeurs. D'où la répartition en lignes. Dans la majorité des cas, pour chaque code type produit, les codes produits s'étendent sur les plages entière des productid

Pairplot : corrélation entre productid et prdtypecode

```
sns.pairplot(fusion[['productid', 'prdtypecode']],
hue="prdtypecode",diag_kind="kde")
plt.title('Catégorie du produit en fonction du productid')
plt.savefig("reports/figures/pairplot.png", bbox_inches='tight')
```



Conclusion : il y a des distributions similaires de productid selon le code type produit : accumulation en début et fin de numérotations