Résumé :

Durant ce projet on va préparer les données pour entrainer un modèle de classification qui a pour but de détecter les supermarchés qui utilisent un système de display pour la mise en avant de leur produit. Le projet se décompose en trois parties : exploration des données, traitement des données et pour finir l'entrainement et la comparaison de modèles Random Forest.

Table des matières :

Liste des figures:

Introduction :

Le but principal de ce projet est de concevoir un modèle de classification capable d'identifier les enseignes qui utilisent des techniques de display pour la mise en avant de leurs produits, en se basant sur un ensemble de variables d'entrée spécifiques. Dans l'univers compétitif du commerce de détail, la stratégie de display joue un rôle essentiel dans la promotion des produits et l'influence sur les comportements d'achat des consommateurs. En exploitant des données telles que le volume des ventes, la valeur des ventes, le chiffre d'affaires du magasin, et d'autres indicateurs pertinents, notre modèle vise à déceler des schémas et des caractéristiques distinctifs des enseignes qui adoptent activement des pratiques de display.

I. Analyse exploratoire des données :

Commençons par voir l'allure de notre base de données :

	Display	cor_sales_in_vol	cor_sales_in_val	CA_mag	value	ENSEIGNE	VenteConv	Feature
0	No_Displ	2.0	20.20	47400	36	CORA	72.0	No_Feat
1	No_Displ	2.0	11.90	62000	24	LECLERC	48.0	No_Feat
2	No_Displ	8.0	29.52	60661	60	AUCHAN	480.0	No_Feat
3	No_Displ	2.0	16.20	59677	19	CARREFOUR	38.0	No_Feat
4	No_Displ	5.0	62.10	142602	50	CORA	250.0	No_Feat
...	...	...	...	...	...	...	...	...
25777	Displ	1.0	10.70	4033	40	CARREFOUR MARKET	40.0	No_Feat
25778	Displ	5.0	34.45	4033	35	CARREFOUR MARKET	175.0	No_Feat
25779	Displ	12.0	81.72	4033	35	CARREFOUR MARKET	420.0	No_Feat
25780	Displ	5.0	29.75	4033	24	CARREFOUR MARKET	120.0	No_Feat
25781	Displ	5.0	52.24	4033	40	CARREFOUR MARKET	200.0	No_Feat

25782 rows × 8 columns

Figure 1: Allure de la base de données

```
0   Display           25782 non-null  object
1   cor_sales_in_vol  25782 non-null  float64
2   cor_sales_in_val  25782 non-null  float64
3   CA_mag            25782 non-null  int64
4   value             25782 non-null  int64
5   ENSEIGNE          25782 non-null  object
6   VenteConv         25782 non-null  float64
7   Feature           25782 non-null  object
```

Figure 2: Infos sur les variables de la base de données

Notre base est composée de 25782 lignes et de 8 colonnes, une variable Y à prédire et 6 variables explicatives :

- Y : Display (variable cible) : l'enseigne utilise un display ou non , variable catégorielle avec deux modalités (Displ ou No_Displ).
- X1 : cor_sales_in_vol : ventes corrigées en volume, variable numérique réelle continue.
- X2 : cor_sales_in_val : ventes corrigées en valeur, variable numérique (float) continue.
- X3 : CA_mag : chiffre d'affaire de l'enseigne, variable numérique (int) continue.
- X4 : value : les valeurs, variable numérique (int) continue.
- X5 : ENSEIGNE : nom de l'enseigne, variable catégorielle à 19 modalités.
- X6 : VenteConv : les ventes converties, variable numérique (float) continue.
- X7 : Feature : si l'enseigne distribue des prospectus de ses produits, variable catégorielle à 2 modalités.

II. Prétraitement des données :

Le but de cette partie est de préparer les données pour l'entrainement du modèle. Comme on peut le constater dans la figure 2, aucune colonne n'a de valeur manquante. On va donc analyser les données pour voir si il existe des valeurs aberrantes dans la data, ensuite on va discrétiser les valeurs continues et encoder les variables catégorielle. Commençons par une étude univariée sur les variables quantitatives de la base de données :

1. Traitement des valeurs aberrantes des variables quantitatives continues :

X1, ventes corrigées en volume :

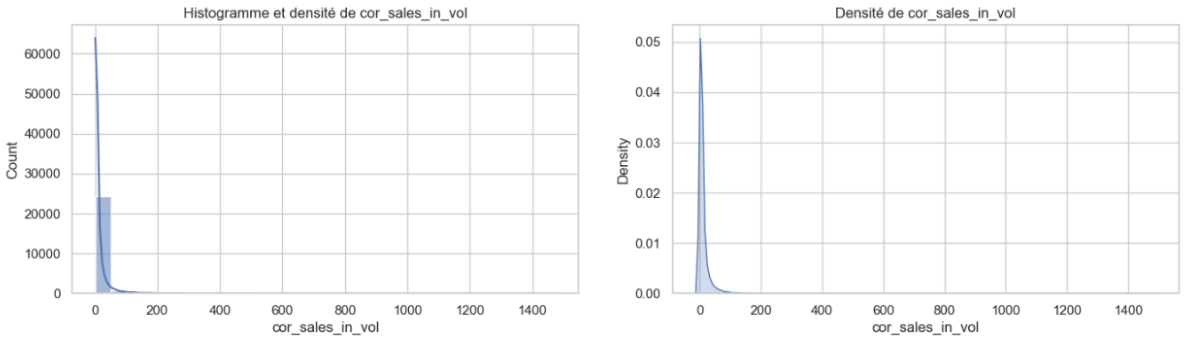

Figure 3 : histogramme et densité de X1

X2, ventes corrigées en valeur :

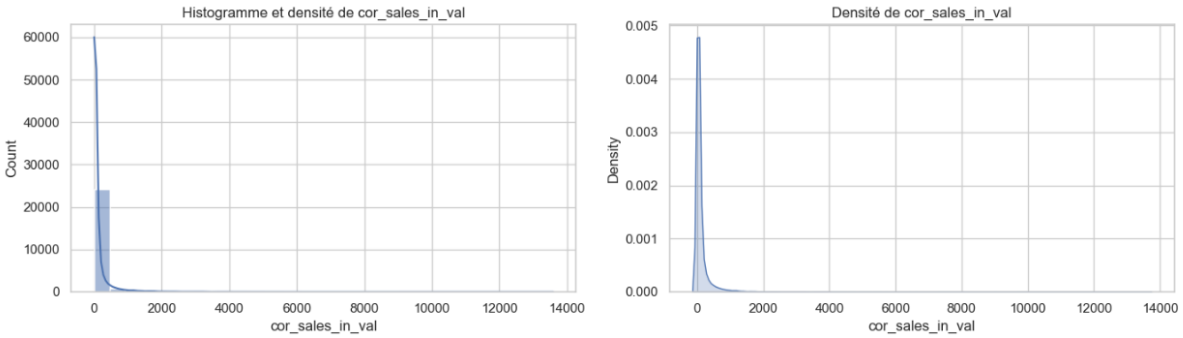

Figure 4 : histogramme et densité de X2

X3, chiffre d'affaire :

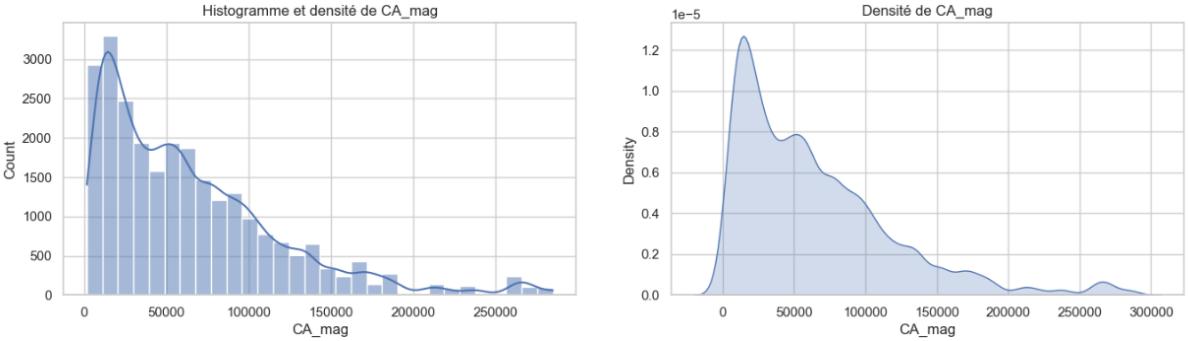

Figure 5 : histogramme et densité de X3

X4, valeur :

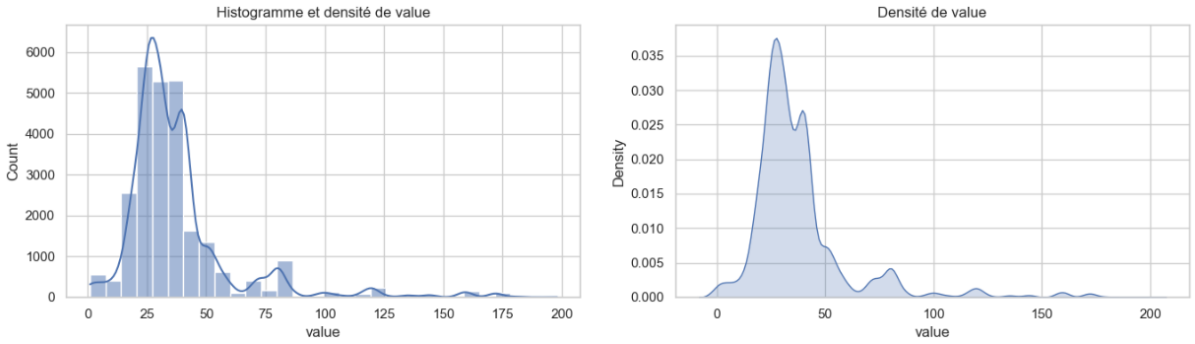

Figure 6 : histogramme et densité de X4

X6, ventes converties :

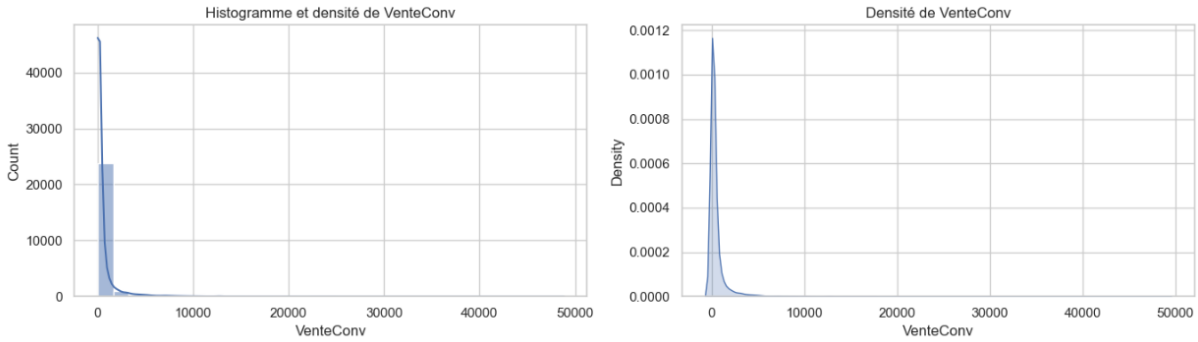

Figure 7 : histogramme et densité de X6

Interprétation : les variables ventes corrigées en volume, vente corrigées en valeur, chiffre d'affaire et ventes converties ont des histogrammes asymétriques, ce qui correspond à l'existence de valeurs aberrantes dans ces variables. La variable valeur quant à elle affiche une concentration autour d'une gamme de valeur (avec un pic quand la variable prend les valeurs entre 20 et 37.5). Cependant, elle est également asymétrique à droite. Donc il existe également des valeurs aberrantes dans la variable valeur.

Afin de mieux souligner ces valeurs aberrantes, on va utiliser des boxplots (boite à moustache) pour mieux les visualiser :

Boxplot de X1, ventes corrigées en volume :

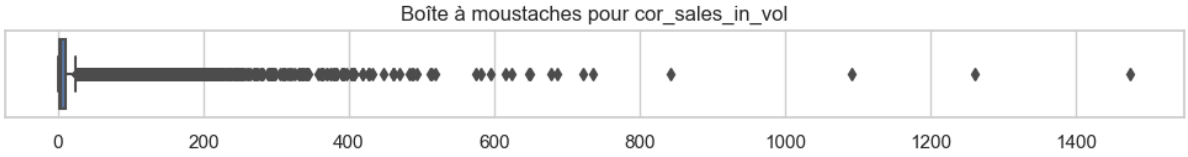

Figure 8 : boite à moustache de X1

Boxplot de X2, ventes corrigées en valeur :

Figure 9 : boite à moustache de X2

Boxplot de X3, chiffre d'affaire :

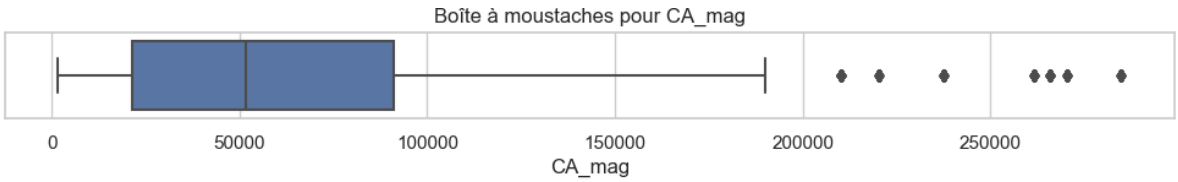

Figure 10 : boite à moustache de X3

Boxplot de X4, valeur :

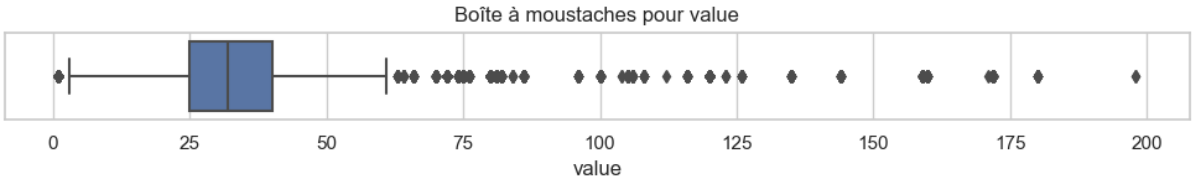

Figure 11 : boite à moustache de X4

Boxplot de X6, ventes converties :

Figure 12 : boite à moustache de X6

Interprétation : grâce aux boites à moustaches on constate clairement la présence de valeurs aberrantes. On va les corriger en utilisant la méthode de la winsorization, elle consiste à fixer une borne minimale et une borne maximale et de d'affecter la valeur de la borne minimale à toutes les valeurs qui sont en- dessous de celle-ci et d'affecter la valeur de la borne maximale à toutes les valeurs en-dessus de celle-ci, cette méthode permet de limiter l'impact des valeurs aberrantes sur le modèle sans avoir à les supprimer. Voici les résultats obtenues après la winsorization :

Résultat de la winsorisation sur X1, ventes corrigées en volume :

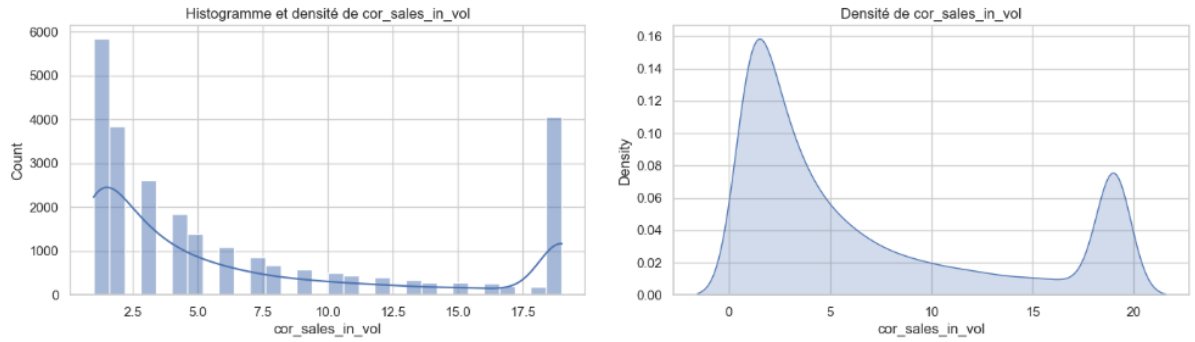

Figure 13 : histogramme et densité de X1 après la winsorisation

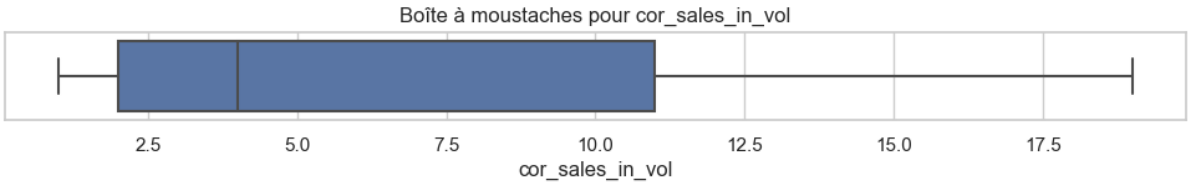

Figure 14 : boite à moustache de X1 après la winsorisation

Résultat de la winsorisation sur X2, ventes corrigées en valeur :

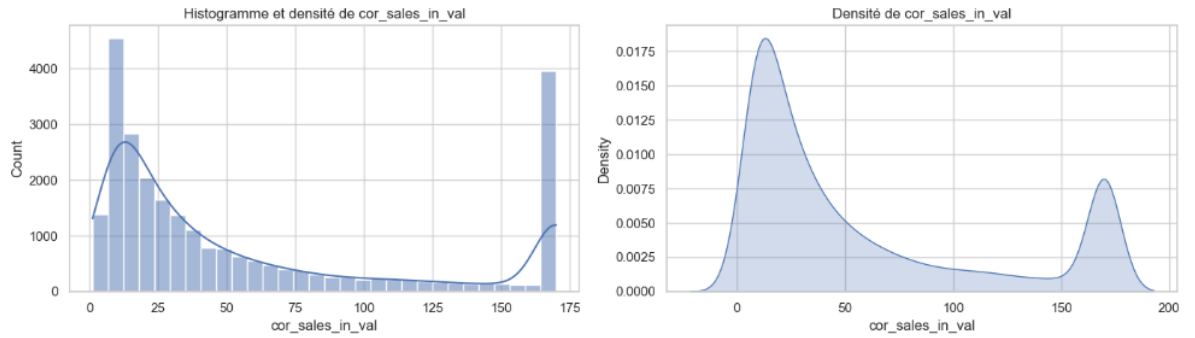

Figure 15 : histogramme et densité de X2 après la winsorisation

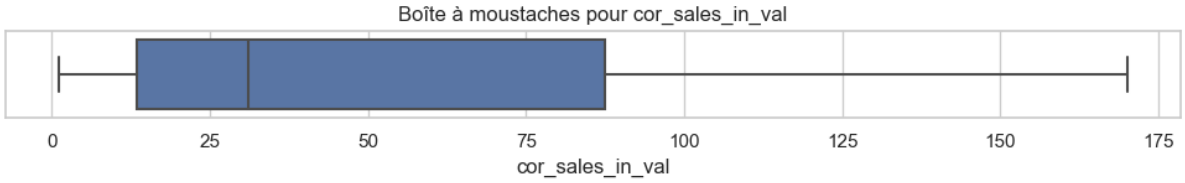

Figure 16 : boite à moustache de X2 après la winsorisation

Résultat de la winsorisation sur X3, chiffre d'affaire :

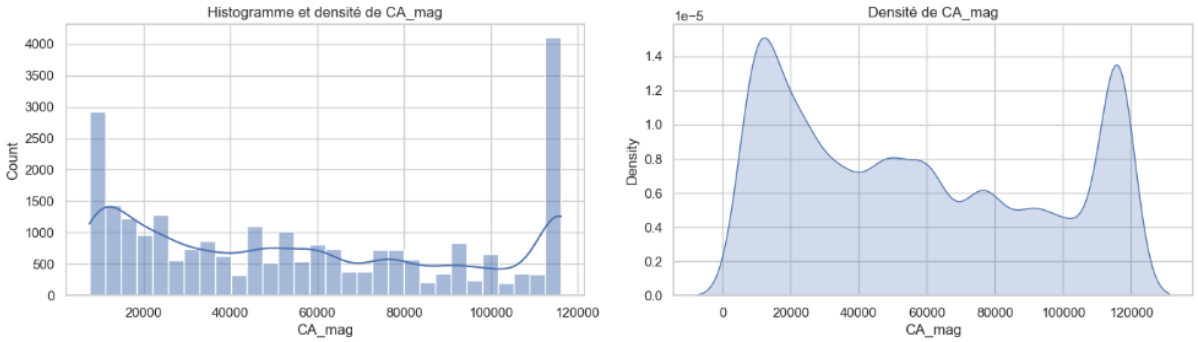

Figure 17 : histogramme et densité de X3 après la winsorisation

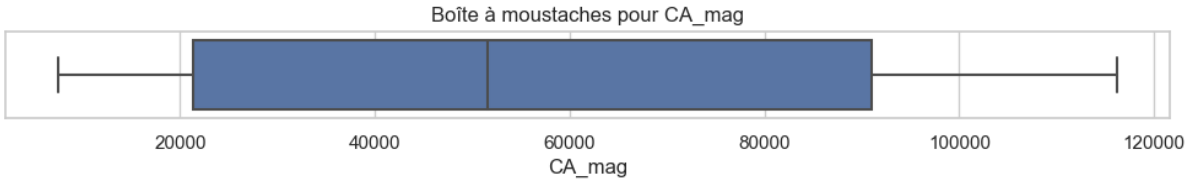

Figure 18 : boite à moustache de X3 après la winsorisation

Résultat de la winsorisation sur X4, valeur :

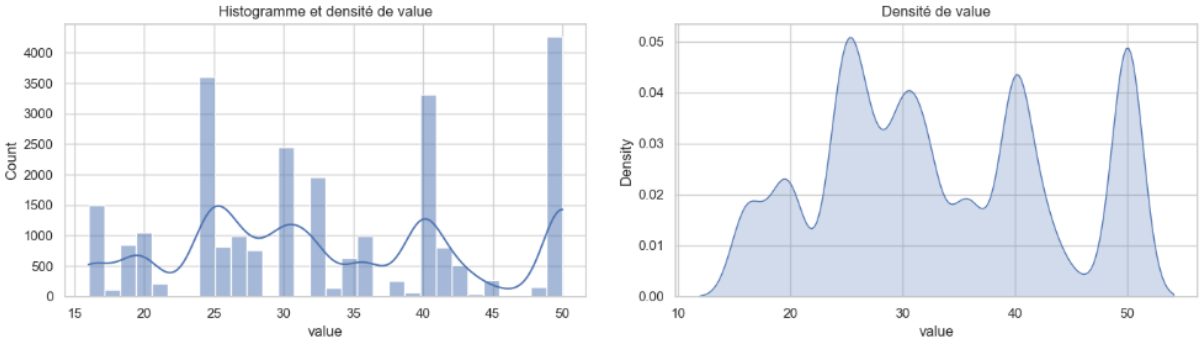

Figure 19 : histogramme et densité de X4 après la winsorisation

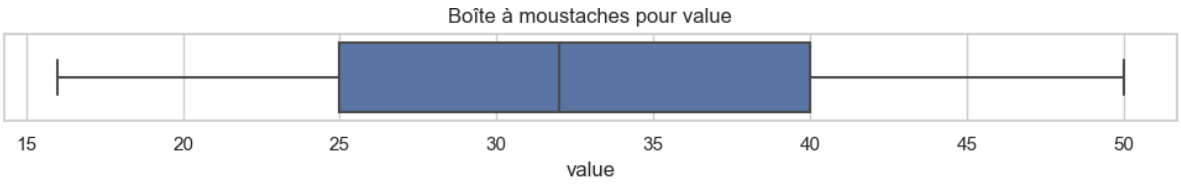

Figure 20 : boite à moustache de X4 après la winsorisation

Résultat de la winsorisation sur X6, ventes converties :

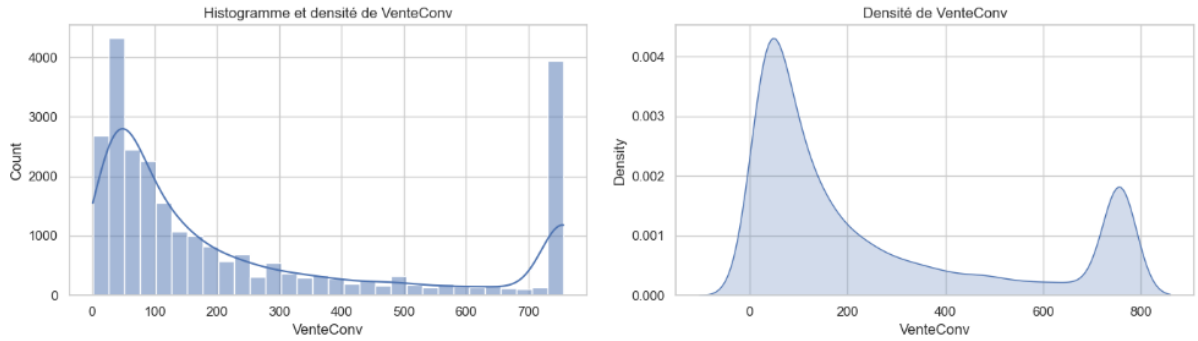

Figure 21 : histogramme et densité de X6 après la winsorisation

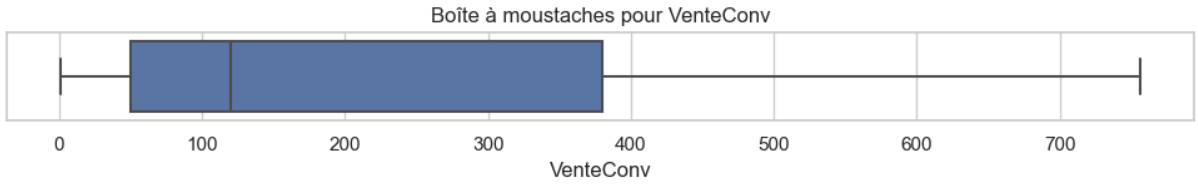

Figure 22 : boite à moustache de X6 après la winsorisation

Le résultat de la winsorisation est facilement remarquable, les données ont une meilleure répartition. On peut maintenant passer à l'étape de la discrétisation.

2. Discrétisation des valeurs continues :

Le discrétisation des valeurs continues est le fait de transformer les valeurs continues en des valeurs discrètes. Il existe plusieurs méthodes de discrétisation, on va utiliser les arbres de décision avec le 'DecisionTreeDiscretiser' de la bibliothèque 'feature-engine' de Python. Cette méthode utilise des arbres de décision pour discrétiser chaque variable continue. Pour chaque variable, il construit un arbre de décision où la variable continue est utilisée pour prédire la variable cible. Les points de division (ou "splits") de l'arbre de décision sont ensuite utilisés comme seuils de discrétisation. Cela permet de diviser la variable continue en segments qui sont informatifs par rapport à la variable cible, c'est-à-dire que chaque bin résultant a une forte association avec la variable cible. On fixe la profondeur maximale de l'arbre à 4 et l'effectif minimal dans un nœud terminal à 1750 pour éviter le sur-apprentissage.

Les variables quantitatives sont désormais des variables discrètes avec 9 modalités pour la variable X1, 7 modalités pour X2, 8 modalités pour X3, 9 modalités pour X4 et 9 modalités pour X6. Voici les arbres de décision ainsi que la nouvelle distribution des variables après l'utilisation de cette méthode :

Arbre de décision et distribution de X1, ventes corrigées en volume :

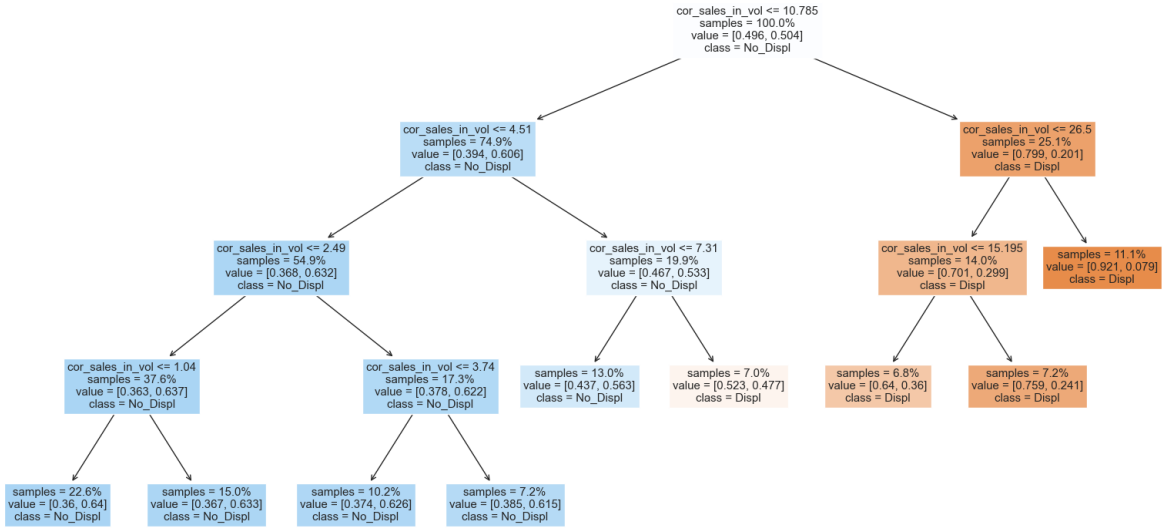

Figure 23 : Arbre de décision pour discrétiser X1

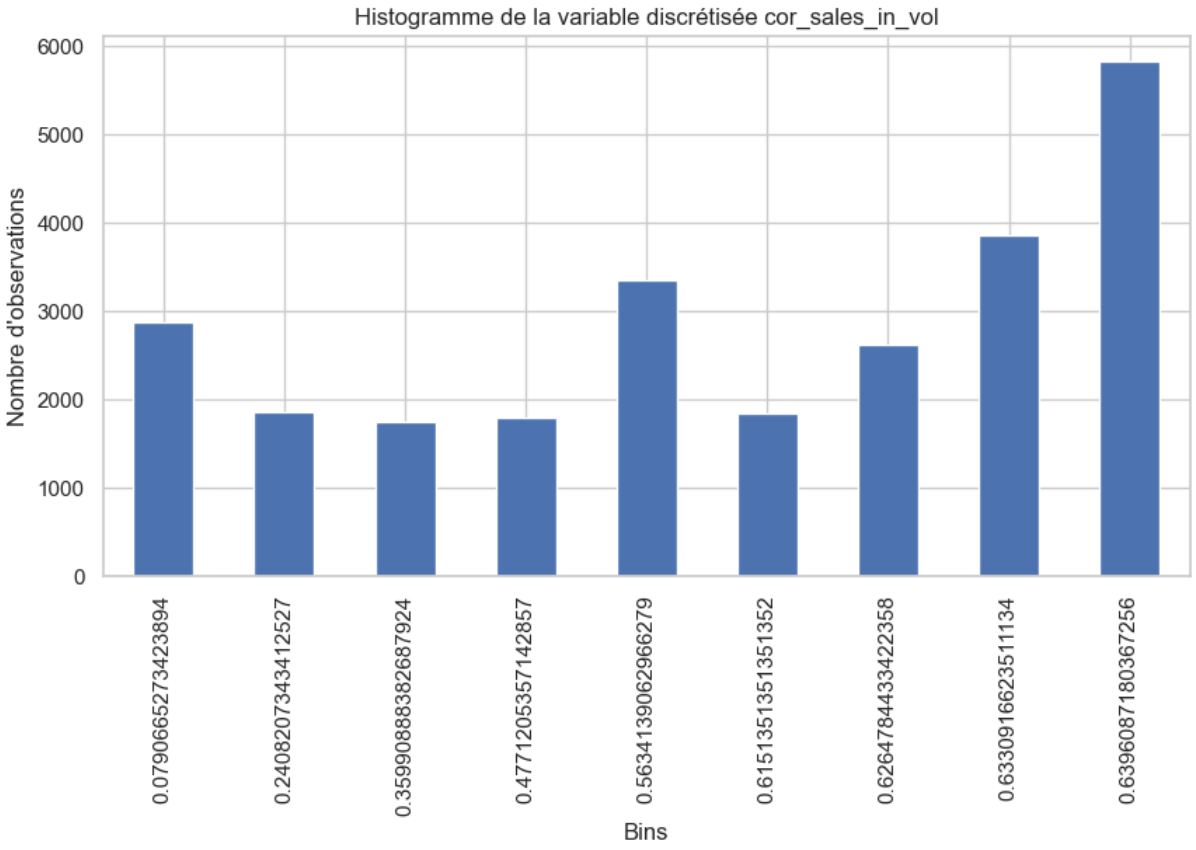

Figure 24 : distribution de X1 après la discrétisation

Arbre de décision et distribution de X2, ventes corrigées en valeur :

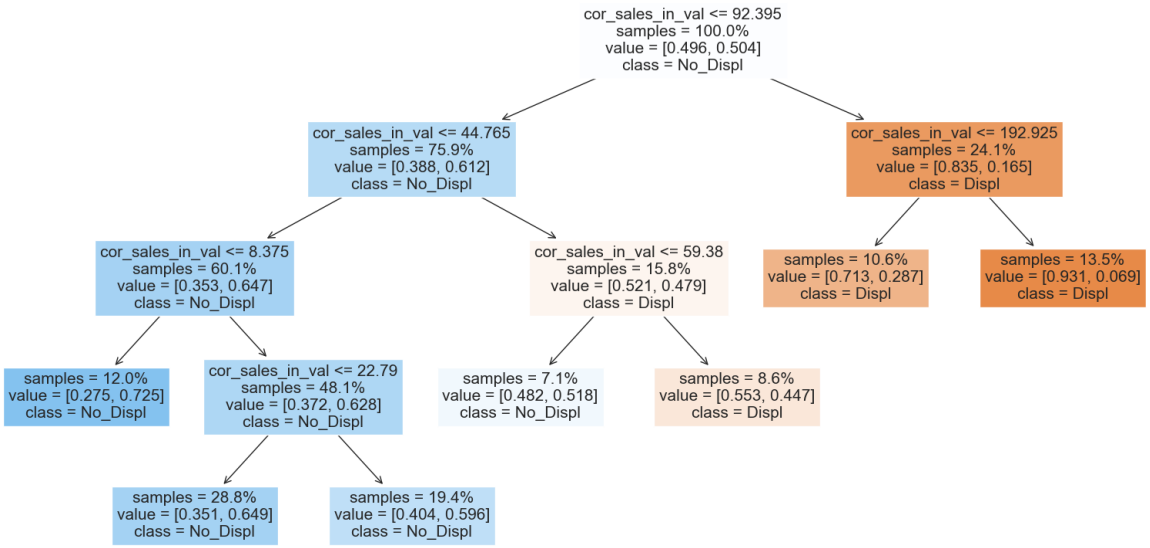

Figure 25 : Arbre de décision pour discrétiser X1

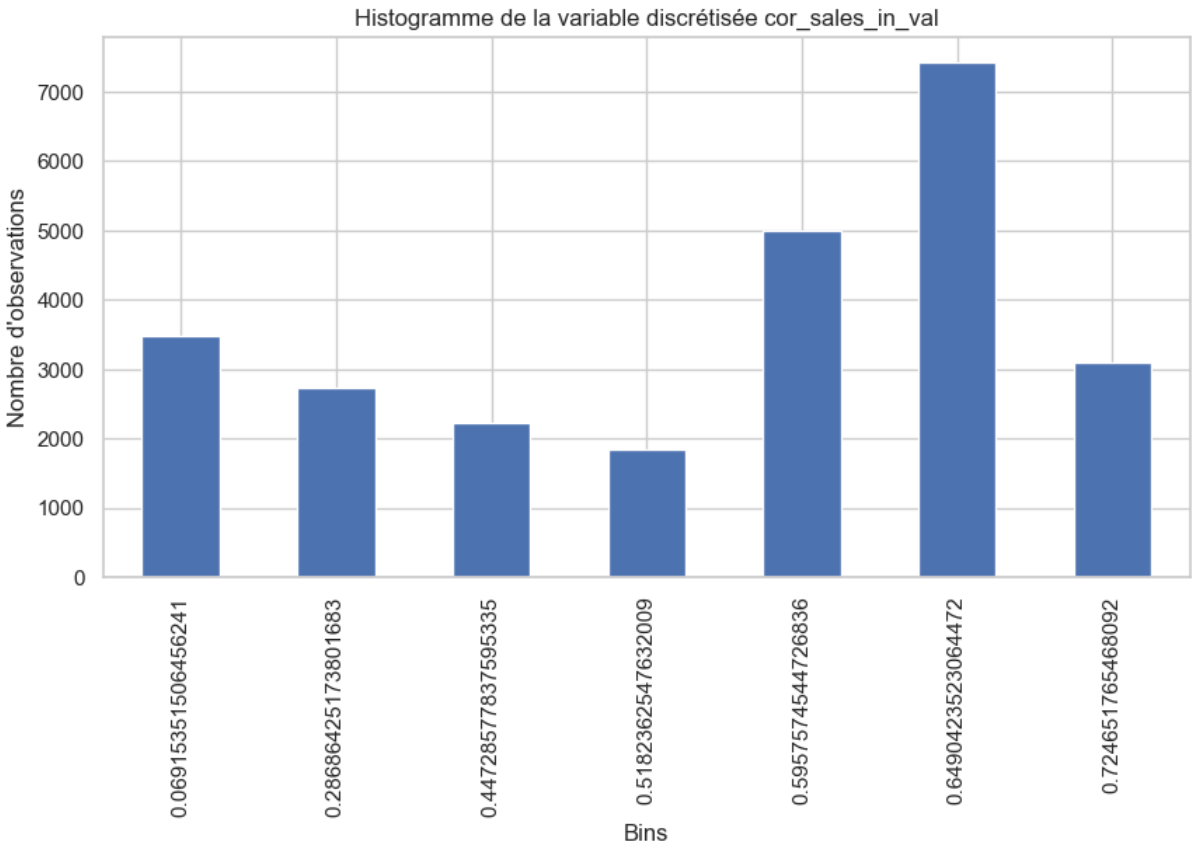

Figure 26 : distribution de X2 après la discrétisation

Arbre de décision et distribution de X3, chiffre d'affaire :

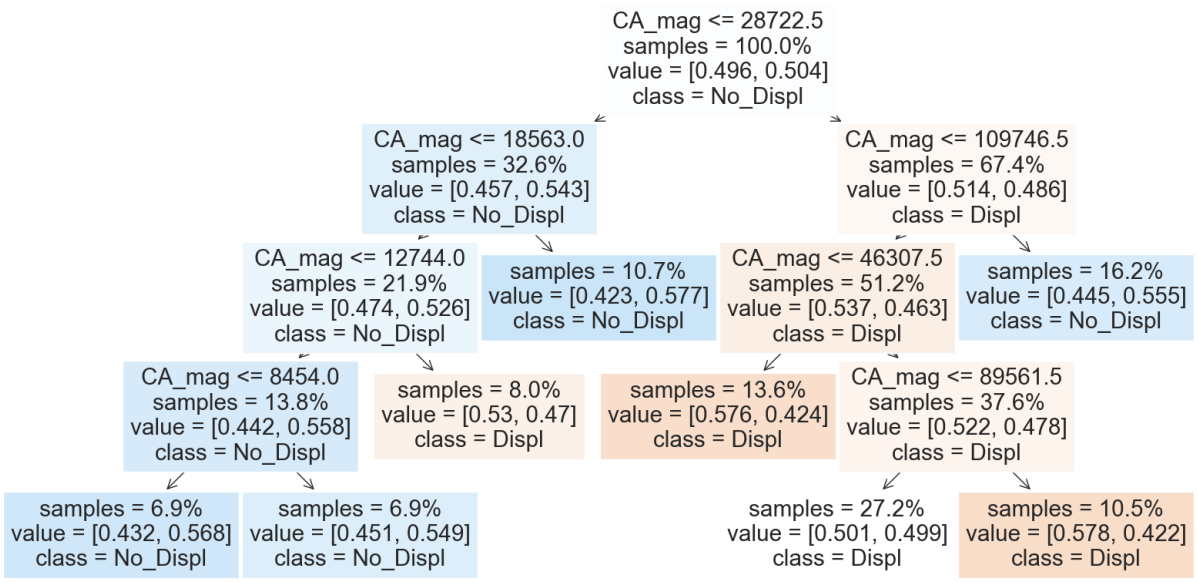

Figure 27 : Arbre de décision pour discrétiser X3

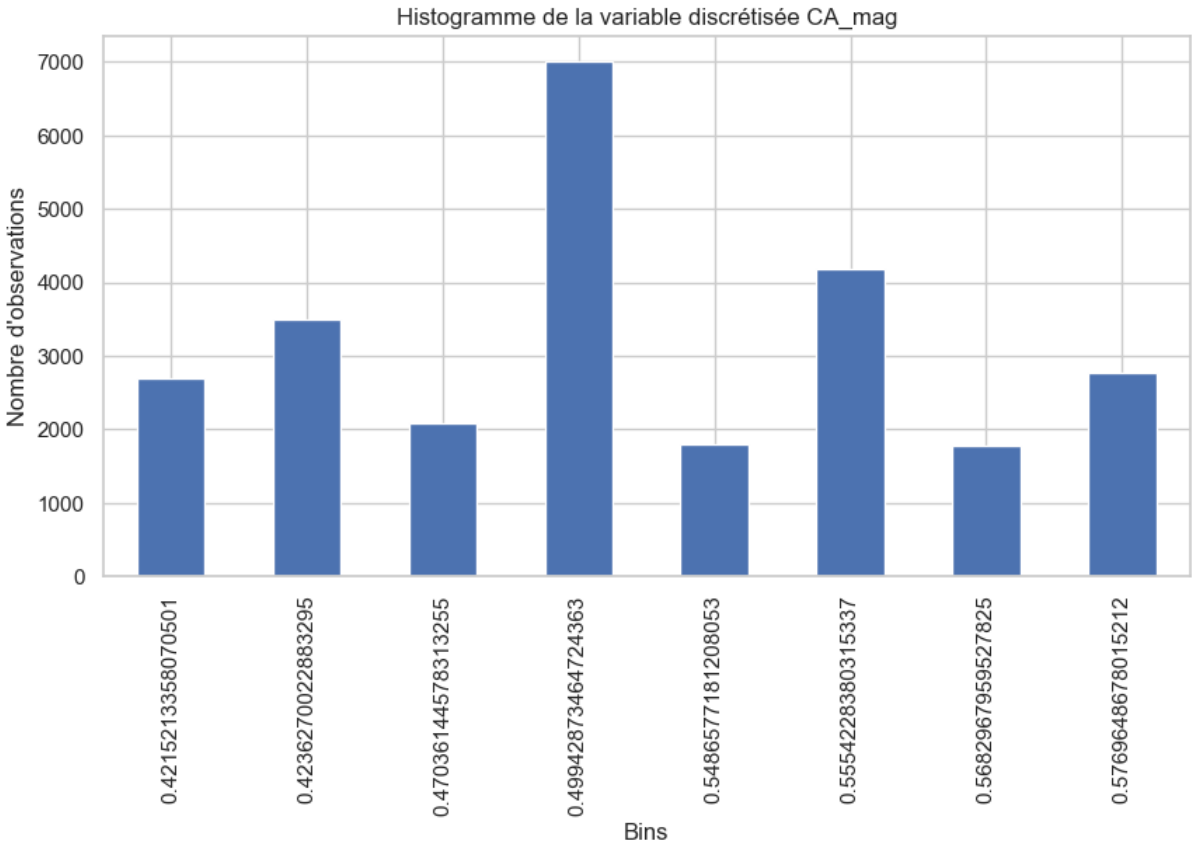

Figure 28 : distribution de X3 après la discrétisation

Arbre de décision et distribution de X4, valeur :

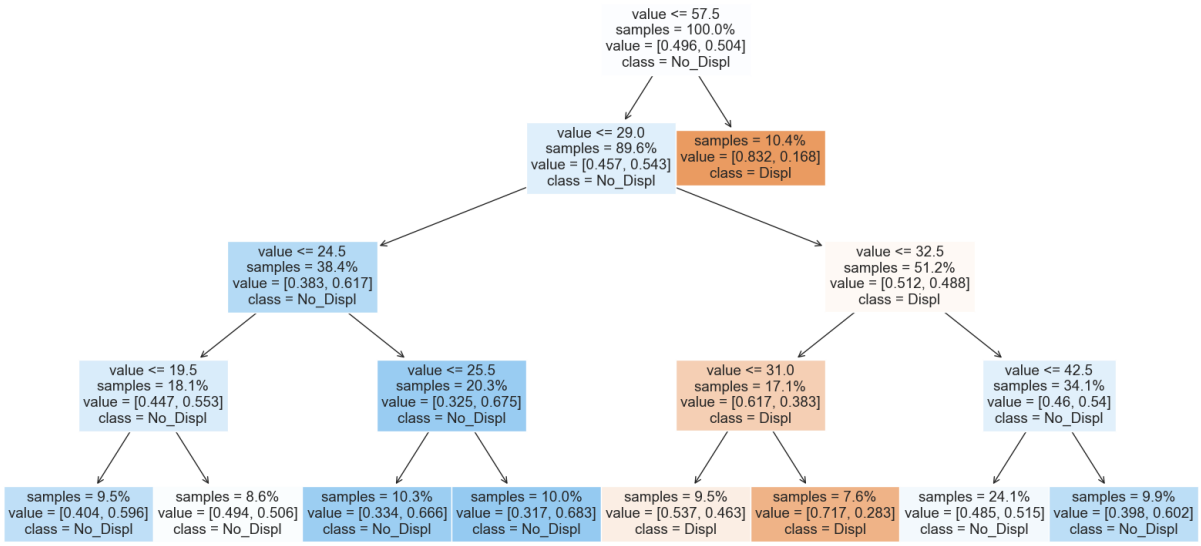

Figure 29 : Arbre de décision pour discrétiser X4

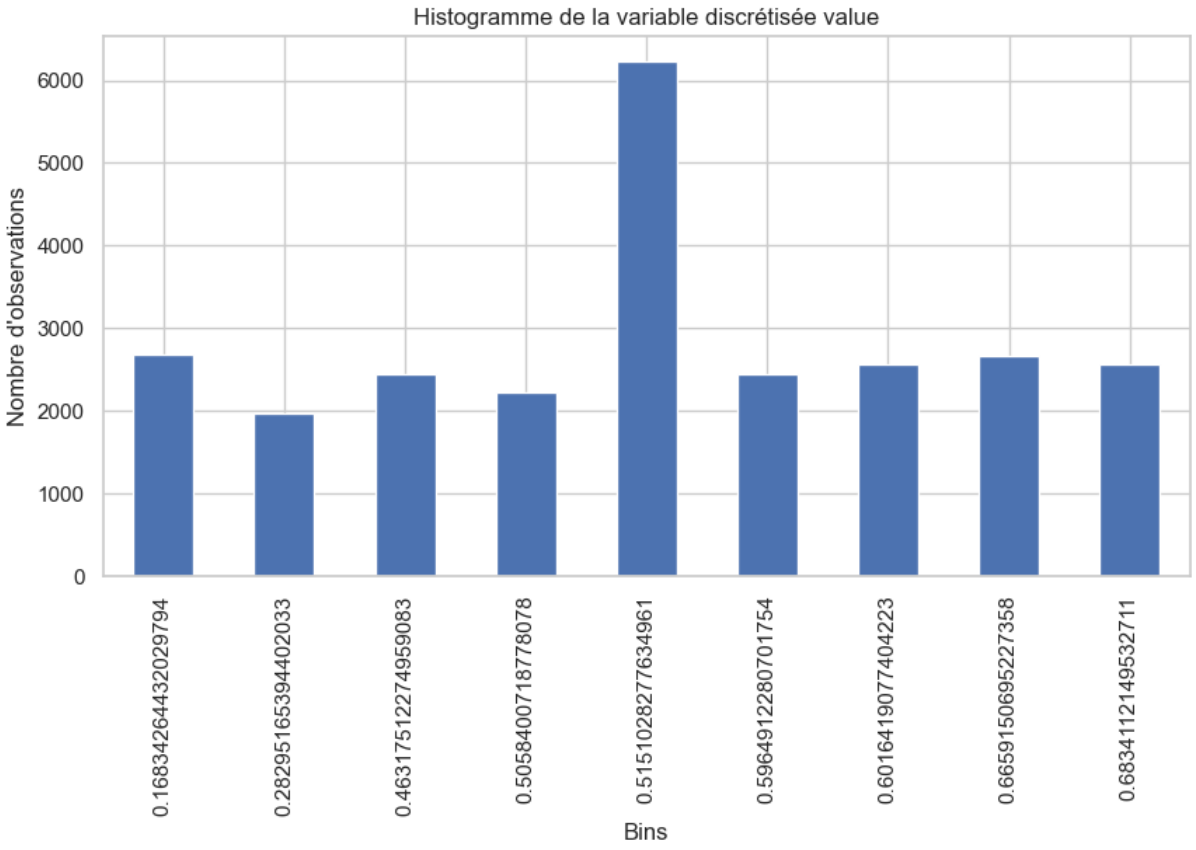

Figure 30 : distribution de X4 après la discrétisation

Arbre de décision et distribution de X6, ventes converties :

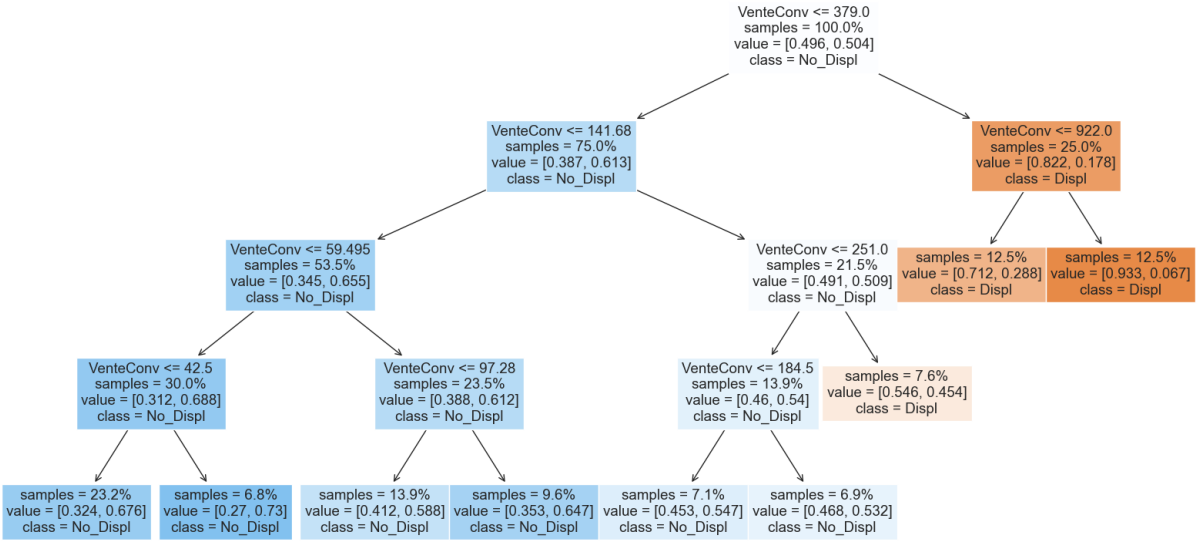

Figure 31 : Arbre de décision pour discrétiser X6

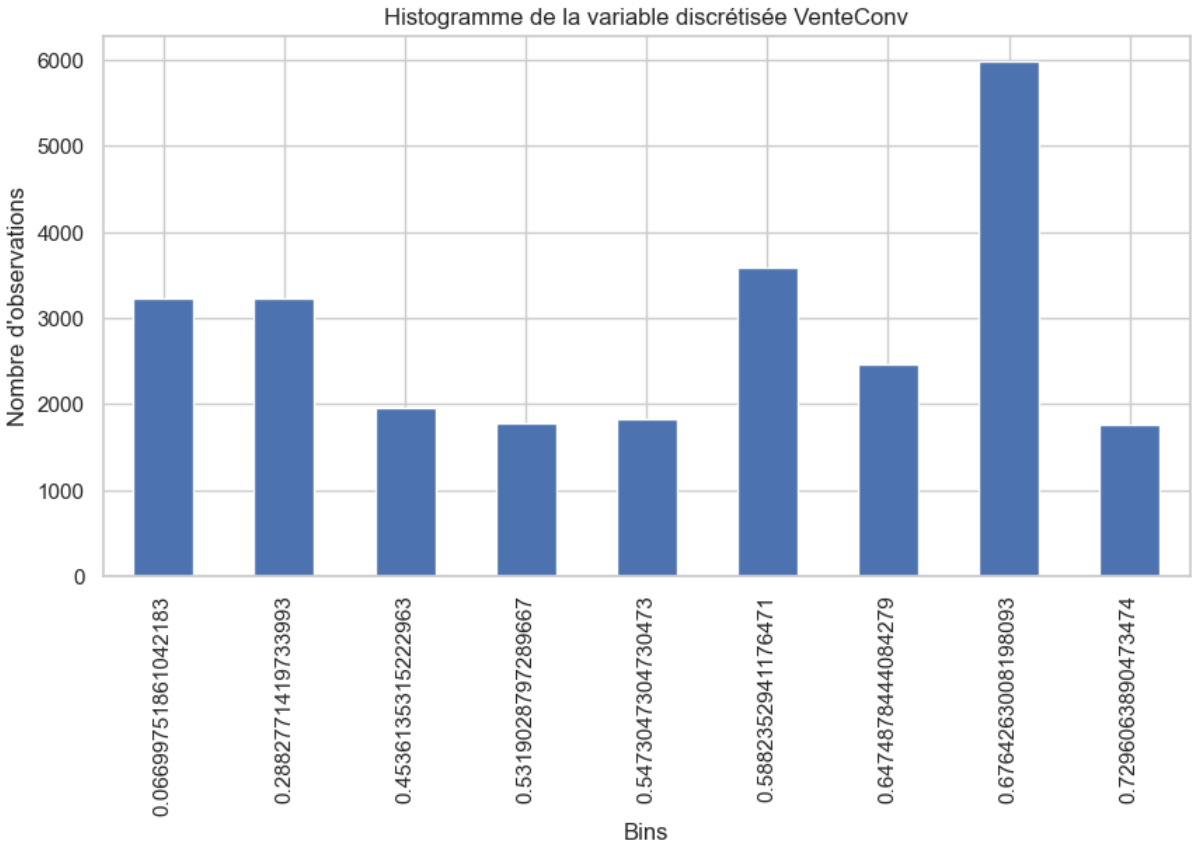

Figure 32 : distribution de X6 après la discrétisation

3. Encodage des variables catégorielles à deux modalités :

Attaquons nous maintenant au reste des variables. Il reste 3 variables catégorielles, la variable cible Y Display, la variable X5 ENSEIGNE et la variable X7 Feature. Y et X7 sont des variables catégorielles à deux modalités en va s'occuper d'eux en premier lieu. Quant à la variable enseigne, elle est composée de 19 modalités on va traiter son cas à part un peu plus tard.

Y et X7 sont des variables catégorielles avec des valeurs textuelles, on va les transformer en des variables catégorielles binaires (ne prennent que 0 et 1 pour valeurs) :

Y : Display	X7 : Feature
No_disp	No_feat
Disp	Feat

Y : Display	X7 : Feature
0	0
1	1

0	No_Displ	No_Feat
1	No_Displ	No_Feat
2	No_Displ	No_Feat
3	No_Displ	No_Feat
4	No_Displ	No_Feat
	...	...
25777	Displ	No_Feat
25778	Displ	No_Feat
25779	Displ	No_Feat
25780	Displ	No_Feat
25781	Displ	No_Feat

0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
	..	..
25777	1	0
25778	1	0
25779	1	0
25780	1	0
25781	1	0

4. Corrélation des variables et tests de significativité :

Avant de passer à l'encodage de la variable ENSEIGNE, on va étudier la corrélation entre les valeurs discrètes et la significativité entre la variable cible et les variables explicatives.

Matrice de corrélation pour les variables X1, X2, X3, X4 et X6 :

On remarque que certaines variable sont fortement corrélées entres elles, il faudra faire attention lors de la construction du modèle car des variables fortement corrélées entres elles peuvent biaisés les résultats et affectés grandement les performances du modèle. L'idéal sera d'utiliser des modèles qui ne sont pas trop affecté par la corrélation entre les variables comme les Decision Trees, le Random Forest ou encore le modèle XG Boost.

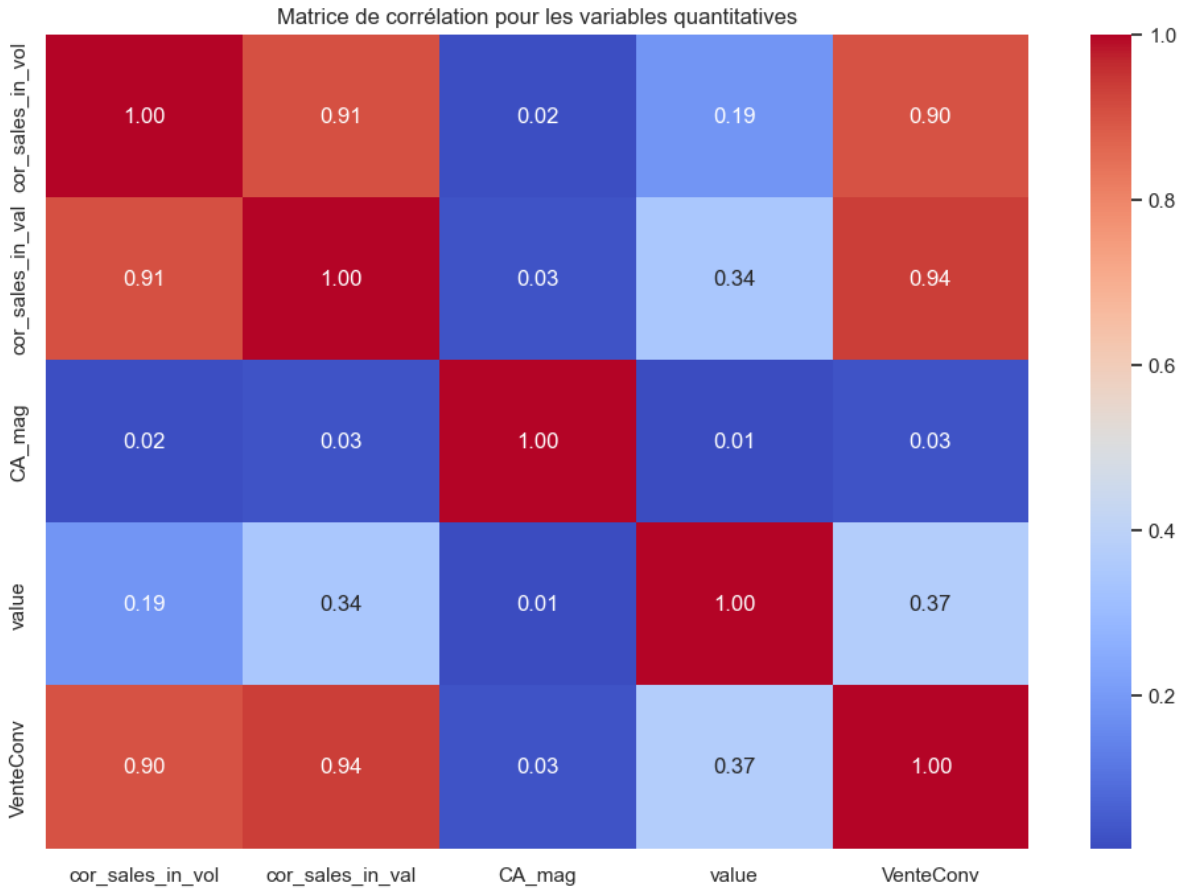

Figure 33 : matrice de corrélation des variables quantitatives

Test de significativité pour l'ensemble des variables :

L'ensemble de nos variables sont des variables catégorielles (après la discrétisation des variables quantitatives), pour tester leur significativités on va utiliser le test CHI-2 qui est adapté pour les données catégorielles :

```
Chi2 test for cor_sales_in_vol:  p-value < 0.05 => cor_sales_in_vol explique la variable cible
Chi2 test for cor_sales_in_val:  p-value < 0.05 => cor_sales_in_val explique la variable cible
Chi2 test for CA_mag:  p-value < 0.05 => CA_mag explique la variable cible
Chi2 test for value:  p-value < 0.05 => value explique la variable cible
Chi2 test for VenteConv:  p-value < 0.05 => VenteConv explique la variable cible
Chi2 test for Feature_Encoded:  p-value < 0.05 => Feature_Encoded explique la variable cible
Chi2 test for ENSEIGNE:  p-value < 0.05 => ENSEIGNE explique la variable cible
```

Figure 34 : résultat du test de CHI-2

Toutes les p-values sont inférieurs à 0.05, donc les 7 variables de notre dataset sont significative et expliquent la variable cible Display.

5. Encodage de la variable X5 ENSEIGNE :

Il ne reste plus qu'une dernière étape avant d'entrainer les modèles. Il faut catégoriser la variable X5 ENSEIGNE. Cependant, X5 est composée de 19 modalités et ce sont tous des valeurs textuelles. Utiliser un LabelEncoder risque de biaisé les résultats du modèle, on va donc se tourner vers deux autres méthodes. La première sera de scinder la base de données en plusieurs segments (plusieurs dataset). Chacun de ses segments comportera un groupe de données qui ont les mêmes caractéristiques et qui affectent la variable cible de la même manière. Pour obtenir ce résultat on va utiliser les arbres de décision.

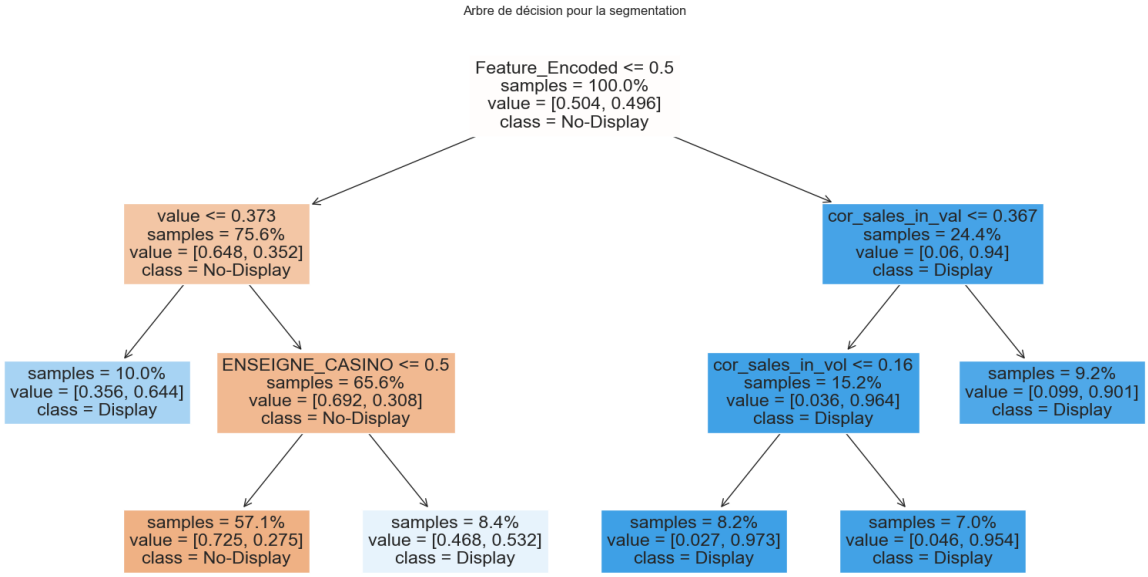

Figure 35 : arbre de décision pour la segmentation des données

L'arbre de décision obtenu possède 6 nœuds terminaux, on obtient donc 6 datasets. Voici les infos sur les nouveaux datasets :

```
---  ------            --------------  -----
 0   Display           2589 non-null   object
 1   cor_sales_in_vol  2589 non-null   float64
 2   cor_sales_in_val  2589 non-null   float64
 3   CA_mag            2589 non-null   float64
 4   value             2589 non-null   float64
 5   ENSEIGNE          2589 non-null   object
 6   VenteConv         2589 non-null   float64
 7   Feature           2589 non-null   object
 8   Display_Encoded   2589 non-null   int64
 9   Feature_Encoded   2589 non-null   int64
```

```
---  ------            --------------  -----
 0   Display           14734 non-null  object
 1   cor_sales_in_vol  14734 non-null  float64
 2   cor_sales_in_val  14734 non-null  float64
 3   CA_mag            14734 non-null  float64
 4   value             14734 non-null  float64
 5   ENSEIGNE          14734 non-null  object
 6   VenteConv         14734 non-null  float64
 7   Feature           14734 non-null  object
 8   Display_Encoded   14734 non-null  int64
 9   Feature_Encoded   14734 non-null  int64
```

```
---  ------            --------------  -----
 0   Display           2172 non-null   object
 1   cor_sales_in_vol  2172 non-null   float64
 2   cor_sales_in_val  2172 non-null   float64
 3   CA_mag            2172 non-null   float64
 4   value             2172 non-null   float64
 5   ENSEIGNE          2172 non-null   object
 6   VenteConv         2172 non-null   float64
 7   Feature           2172 non-null   object
 8   Display_Encoded   2172 non-null   int64
 9   Feature_Encoded   2172 non-null   int64
```

```
---  ------            --------------  -----
 0   Display           2110 non-null   object
 1   cor_sales_in_vol  2110 non-null   float64
 2   cor_sales_in_val  2110 non-null   float64
 3   CA_mag            2110 non-null   float64
 4   value             2110 non-null   float64
 5   ENSEIGNE          2110 non-null   object
 6   VenteConv         2110 non-null   float64
 7   Feature           2110 non-null   object
 8   Display_Encoded   2110 non-null   int64
 9   Feature_Encoded   2110 non-null   int64
```

```
---  ------           --------------  -----
 0   Display          1797 non-null   object
 1   cor_sales_in_vol 1797 non-null   float64
 2   cor_sales_in_val 1797 non-null   float64
 3   CA_mag           1797 non-null   float64
 4   value            1797 non-null   float64
 5   ENSEIGNE         1797 non-null   object
 6   VenteConv        1797 non-null   float64
 7   Feature          1797 non-null   object
 8   Display_Encoded  1797 non-null   int64
 9   Feature_Encoded  1797 non-null   int64
```

```
---  ------           --------------  -----
 0   Display          2380 non-null   object
 1   cor_sales_in_vol 2380 non-null   float64
 2   cor_sales_in_val 2380 non-null   float64
 3   CA_mag           2380 non-null   float64
 4   value            2380 non-null   float64
 5   ENSEIGNE         2380 non-null   object
 6   VenteConv        2380 non-null   float64
 7   Feature          2380 non-null   object
 8   Display_Encoded  2380 non-null   int64
 9   Feature_Encoded  2380 non-null   int64
```

La deuxième méthode consiste à décomposer la variable ENSEIGNE en plusieurs colonnes, une colonne pour chaque modalité de la variable ENSEIGNE. Les valeurs de ces nouvelles colonnes seront des variables catégorielles binaire. Pour illuster ça prenons un exemple :

	cor_sales_in_vol	cor_sales_in_val	CA_mag	value	VenteConv	Feature_Encoded	ENSEIGNE	Display_Encoded
0	0.633092	0.649042	0.499429	0.515103	0.588235	0	CORA	0
1	0.633092	0.649042	0.499429	0.505840	0.729606	0	LECLERC	0

Figure 36 : exemple pour la décomposition de la colonne ENSEIGNE

Ici les deux premières lignes de la variable ENSEINGNE ont pour valeurs CORA et LECLERC, une nouvelle colonne aura pour nom CORA_Encoded, la première ligne de la nouvelle colonne CORA_Encoded aura pour valeur 1, tendis que la seconde ligne de cette colonne aura une valeur de 0.

Au final on aura un total de 19 colonnes avec des valeurs catégorielles binaires car la variable ENSEIGNE compte 19 modalités.

```
---  ------           --------------  -----
 0   Display          25782 non-null  object
 1   cor_sales_in_vol 25782 non-null  float64
 2   cor_sales_in_val 25782 non-null  float64
 3   CA_mag           25782 non-null  float64
 4   value            25782 non-null  float64
 5   ENSEIGNE         25782 non-null  object
 6   VenteConv        25782 non-null  float64
 7   Feature          25782 non-null  object
 8   Display_Encoded  25782 non-null  int32
 9   Feature_Encoded  25782 non-null  int32
 10  CORA             25782 non-null  int32
 11  LECLERC          25782 non-null  int32
 12  AUCHAN           25782 non-null  int32
 13  CARREFOUR        25782 non-null  int32
 14  CASINO           25782 non-null  int32
 15  SUPER U          25782 non-null  int32
 16  GEANT            25782 non-null  int32
 17  CARREFOUR MARKET 25782 non-null  int32
 18  FRANPRIX         25782 non-null  int32
 19  INTERMARCHE      25782 non-null  int32
 20  ECOMARCHE        25782 non-null  int32
 21  MONOPRIX         25782 non-null  int32
 22  SIMPLY MARKET    25782 non-null  int32
 23  OTHERS           25782 non-null  int32
 24  MATCH            25782 non-null  int32
 25  PRISUNIC         25782 non-null  int32
 26  HYPER U          25782 non-null  int32
 27  SHOPI            25782 non-null  int32
 28  MARCHE U         25782 non-null  int32
```

Figure 37 : dataset après la décomposition de la variable ENSEIGNE

Notre base de données est maintenant composée de 29 colonnes, les colonnes rajoutées sont des variables catégorielles binaires.

III. Entrainement et comparaison des modèles :

Maintenant que nos données sont prêtes, on va entrainer un modèle Random Forest. Bien évidement on va entrainer différents modèles un pour chacune des bases de données dont on dispose (7 bases de données obtenues via les 2 méthodes effectuées pour traiter les données) afin de comparer les résultats des modèles selon la méthode utilisée.

1. Résultats du Random Forest sur les datasets obtenus à partir de l'arbre de décision :

Commençons par les 6 bases de données obtenues grâce à la méthode des arbres de décision :

	Accuracy	Precision	Recall	F1-score
Data 1 (10% des données)	0.70	0.59	0.55	0.57
Data 2 (57.1% des données)	0.74	0.77	0.91	0.84
Data 3 (10% des données)	0.66	0.67	0.58	0.62
Data 4 (8.4% des données)	0.97	0	0	0
Data 5 (8.2% des données)	0.95	0	0	0
Data 6 (7% des données)	0.88	0.29	0.15	0.22

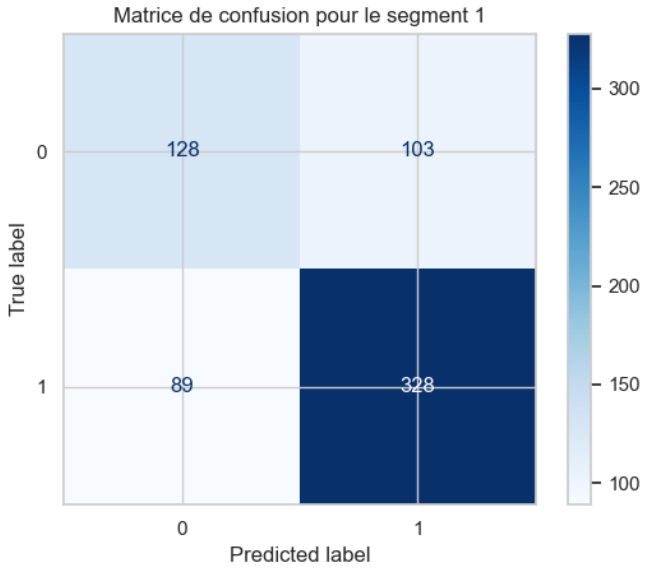

Figure 38 : matrice de confusion du premier dataset

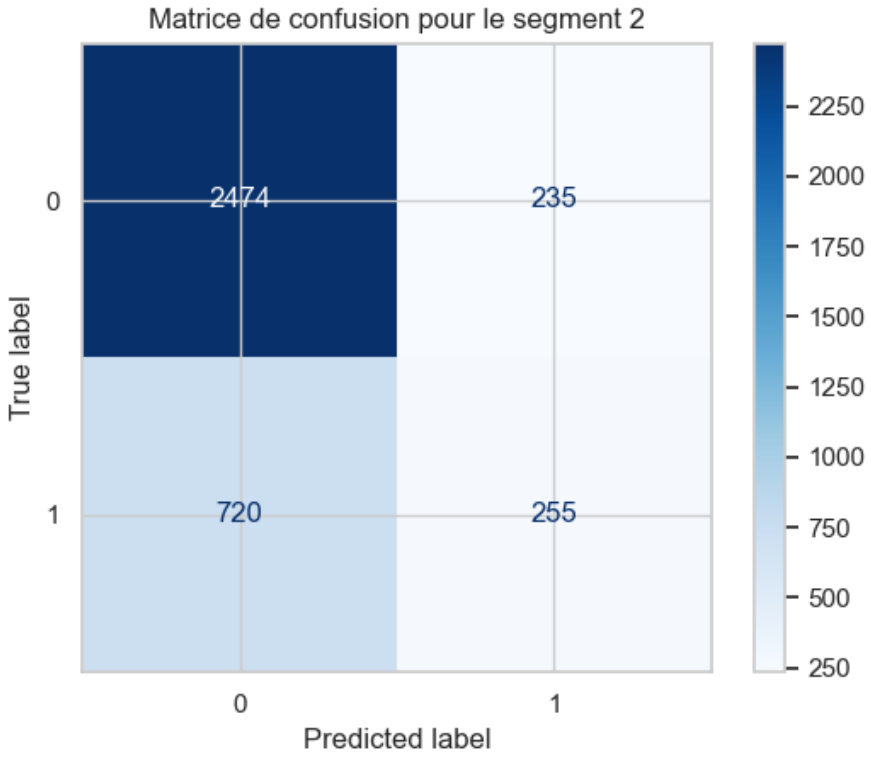

Figure 39 : matrice de confusion du deuxième dataset

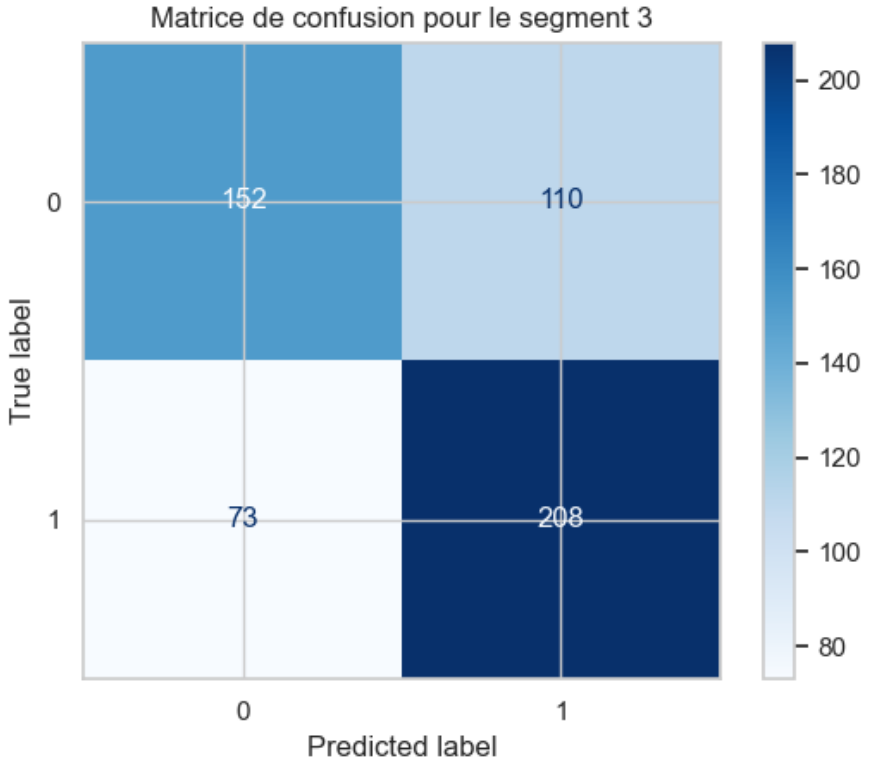

Figure 40 : matrice de confusion du troisième dataset

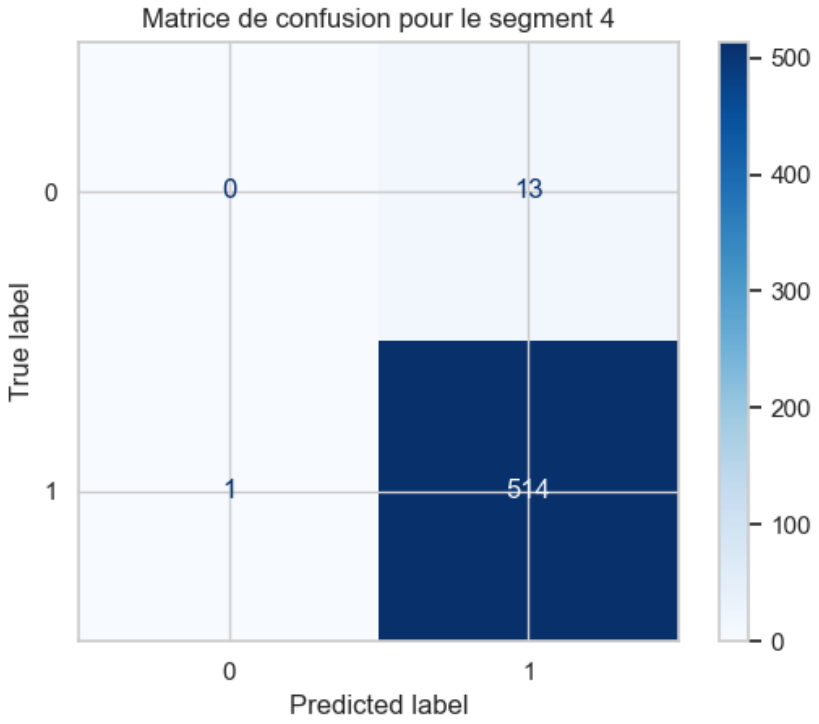

Figure 41 : matrice de confusion du quatrième dataset

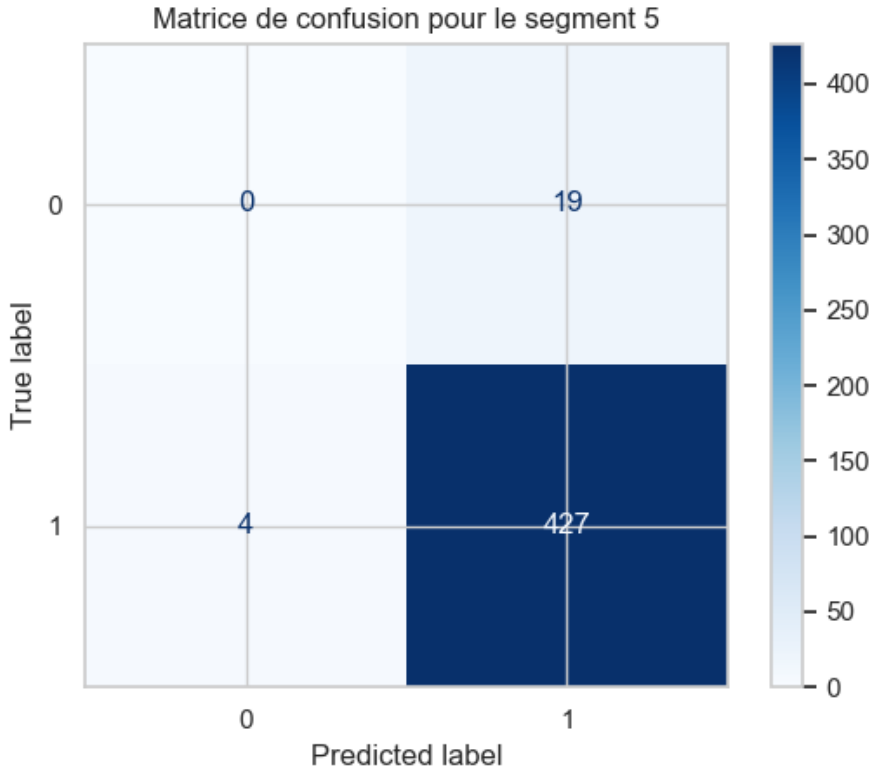

Figure 42 : matrice de confusion du cinquième dataset

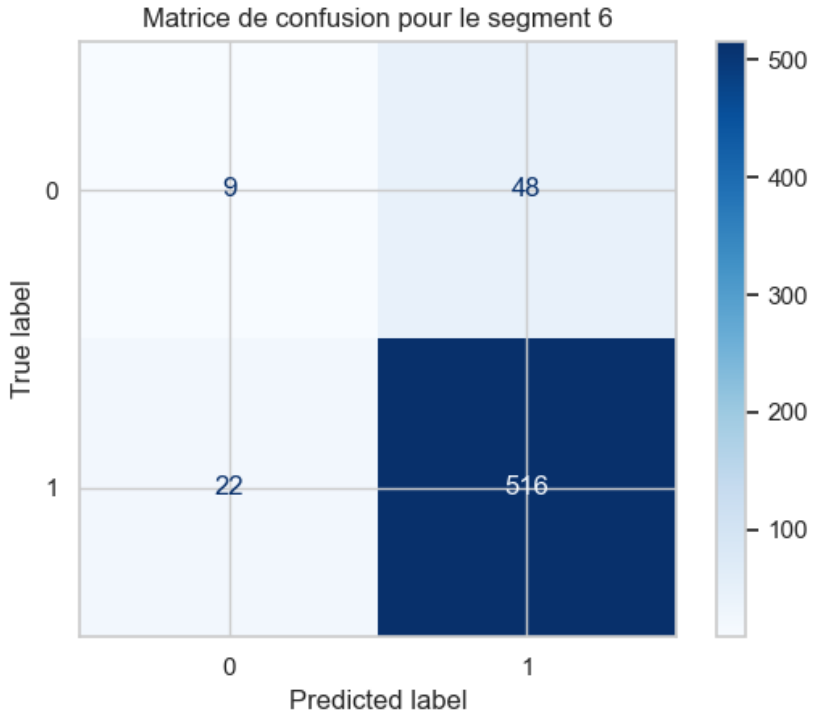

Figure 43 : matrice de confusion du sixième dataset

Le but de notre projet étant de minimiser le risque. Pour se faire, il faut que les modèles ne classent pas les valeurs ou y='No_Displ' en réalité en 'Displ'. On va donc s'intéresser particulièrement à la case en bas à gauche des matrices de confusion car elle représente les faux positives. En conclusion voici les résultats du modèle Random Forest sur la data de la méthode 1 (arbre de décision) :

- Moyenne pondérée de l'accuracy des modèles : 0.77
- Total de faux positives : 909 faux positives sur 6448 valeurs des ensembles de test.

2. Résultats du Random Forest sur le dataset obtenu à partir de la décomposition de la variable ENSEIGNE :

On entraine un modèle Random Forest avec le dataset obtenu à travers la seconde méthode de traitement des données qui consistait à décomposer la colonne ENSEIGNE en plusieurs colonnes (une nouvelle colonne pour chaque modalité de la variable ENSEIGNE). Les résultats du modèle sont les suivants :

	Accuracy	Precision	Recall	F1-score
Data 7 (28 colonnes)	0.79	0.78	0.80	0.79

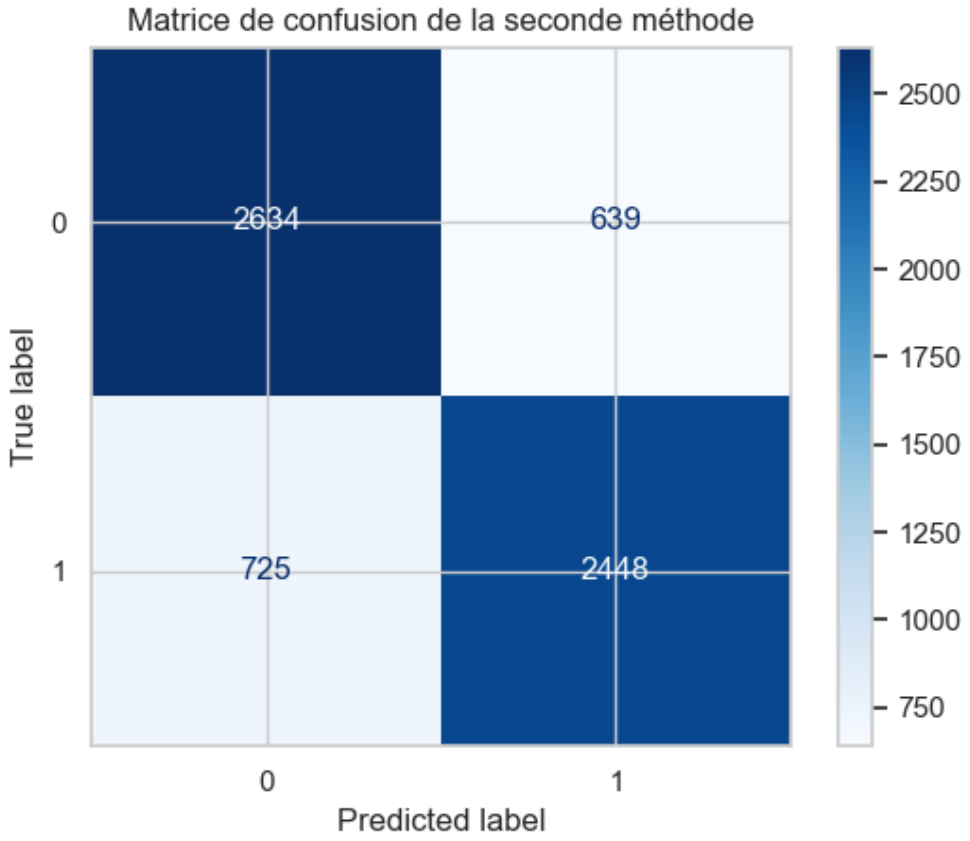

Figure 44 : matrice de confusion du septième dataset

En conclusion voici les résultats du modèle Random Forest sur la data de la méthode 2 (décomposition de la variable ENSEIGNE en plusieurs colonnes) :

- Accuracy du modèle : 0.79
- Total de faux positives : 725 faux positives sur 6446 valeurs dans l'ensemble de test.

3. Comparaison des deux méthodes de traitement des données :

	Accuracy du modèle	Faux positives	Risque
Data de la Méthode 1	77%	909/6448	14%
Data de la méthode 2	79%	725/6446	11%

Les résultats de la deuxième méthode qui consiste à décomposer la variable ENSEIGNE en plusieurs colonnes catégorielles binaires est bien meilleure que la méthode des arbres de décision. Elle offre un modèle plus performant et un risque réduit.

Conclusion :

A travers ce projet qui avait pour but de modéliser un classifieur pour détecter les supermarchés qui utilisent un système de display pour la mise en avant de leur produit, on a eu l'occasion d'analyser la distribution des données et leur densité, de traiter les valeurs manquantes avec la méthode de la winsorisation puis de discrétiser les variables continues. Ensuite on a encodé les variables catégorielles. La variable enseigne posait problème car elle contenait 19 modalités, pour résoudre ce problème on a opté pour deux méthodes. La première consiste à scinder la base de données en différent dataset qui regroupe les données ayant les mêmes caractéristiques. La seconde quant à elle consiste à décomposer la variable enseigne en 19 colonnes (une colonne pour chaque modalité de la variable enseigne).

Grâce au métriques du modèle Random Forest et aux matrices de confusion on a pu comparer les deux méthodes. La seconde méthode offre de meilleures performances et minimise le risque de prédire des faux positives par le modèle.