Discrétisation des variables

Eunice KOFFI, Diakité GAOUSSOU, Camil ZAHI

2022-11-27

Il est important de discrétiser nos variables pour permettre une meilleure performance de notre modèle de classification. Aussi, dans la suite de notre analyse, une discrétisation facilitera la création de notre grille de score.

Librarie R scorecard

La librairie scorecard est celle qui nous permettra d'avoir notre grille de score à l'issu de notre modélisation. Nous l'importons donc pour l'utiliser.

```
library(scorecard)
```

Importation des bases de données

Nous importons les bases de données issues du traitement préalable que nous avons fait sous python.

Nous avons remarqué que certaines variables ne sont pas au bon format. Il est donc important de les mettre au bon format pour la suite de notre analyse.

```
# Modification du type de variables
# Certaines variables ne sont pas au bon format.

base_up$defaut_36mois               <- as.character(base_up$defaut_36mois)
base_up$TYP_CNT_TRA_MAX_BRP         <-
as.character(base_up$TYP_CNT_TRA_MAX_BRP)
base_up$ASU_BIEN_FIN_BRP            <- as.character(base_up$ASU_BIEN_FIN_BRP)
base_up$NAT_BIEN_FIN_BRP            <- as.character(base_up$NAT_BIEN_FIN_BRP)
base_up$COD_ETA_BIEN_CRI            <- as.character(base_up$COD_ETA_BIEN_CRI)
base_up$QUA_INT_MAX_BRP             <- as.character(base_up$QUA_INT_MAX_BRP)
base_up$TOP_PRET_RELAIS_BRP         <-
as.character(base_up$TOP_PRET_RELAIS_BRP)
base_up$top_exist_conso_revo_BRP    <-
as.character(base_up$top_exist_conso_revo_BRP)
base_up$TOP_NAT_FR_CRI              <- as.character(base_up$TOP_NAT_FR_CRI)
base_up$top_locatif                 <- as.character(base_up$top_locatif)
```

Discrétisation avec WOE et sélection de variables avec IV

Le poids de la preuve (WOE) et la valeur de l'information (IV) sont des techniques simples mais puissantes pour effectuer une transformation et une sélection variables. Cet algorithme permet de faire à la fois la discrétisation des variables continues et un regroupement des modalités des variables catégorielles de manière optimale. En effet, nous lui demandons de se baser sur le taux de défaut afin de s'assurer qu'on ait au moins 5% d'effectif dans chaque classe et que le taux de défaut dans chaque classe soit bien différencié. On parle de binning optimal. L'algorithme utilise des méthodes de segmentation arborescente ou le Chi-deux pour fusionner des modalités. C'est une mesure du pouvoir prédictif d'une variable indépendante par rapport à la variable cible. L'analyse des résultats nous indique donc dans quelle mesure une variable peut différencier les clients sains des clients en défaut. L'un des avantages de l'utilisation de cet algorithme est la gestion des valuers manquantes et des valeurs aberrantes. La formule pour calculer le WoE est la suivante : W oE = ln(%def aut %non – def aut) (1) Un WoE positif signifie que la proportion de bons clients est supérieure à celle des mauvais clients. De la même manière, un WoE négatif signifie que la proportion de mauvais clients est supérieure à celle des bons clients.

L'IV

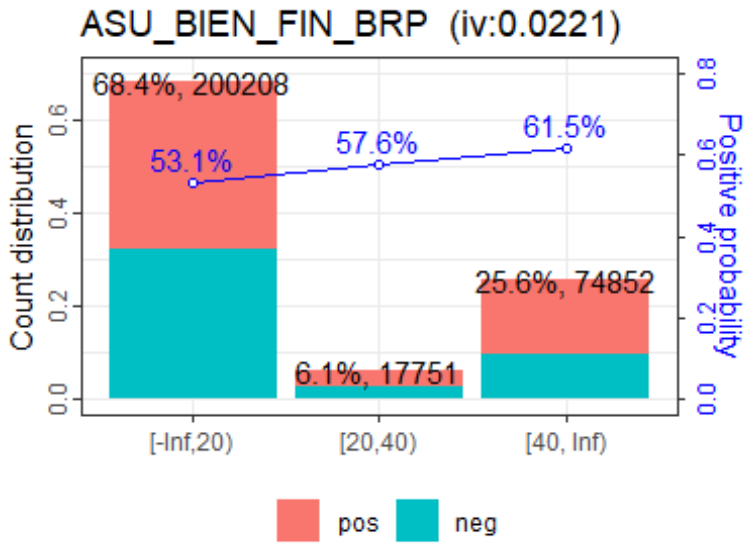
Information Value	Variable Predictiveness
Moins de 0.02	Non prédictif
0.02 de 0.1	Faible pouvoir prédictif
0.1 de 0.3	Pouvoir prédictif moyen
0.3 de 0.5	Fort pouvoir prédictif
0.5	Trop grand pour être vrai

Discrétisation

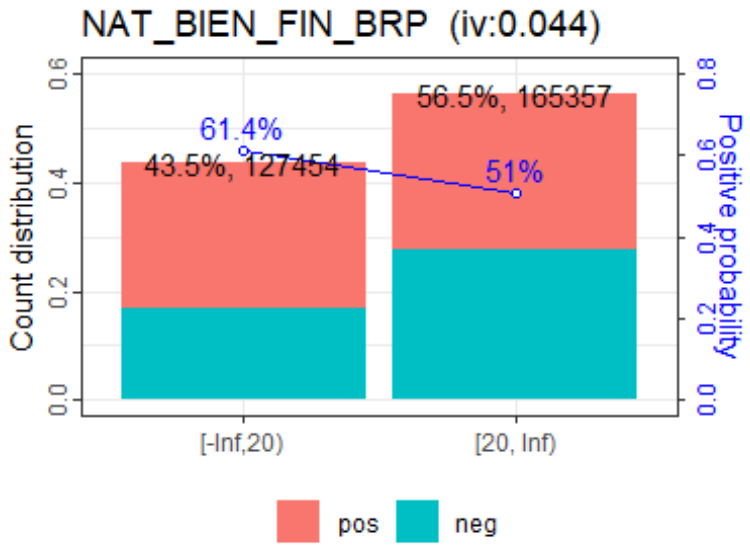
```
bins= woebin(base_up, y='defaut_36mois', positive =0, method="tree", var_skip
= 'date_debloc_avec_crd')

## ✓ Binning on 292811 rows and 27 columns in 00:00:06
```

Cette méthode nous permet également d'avoir

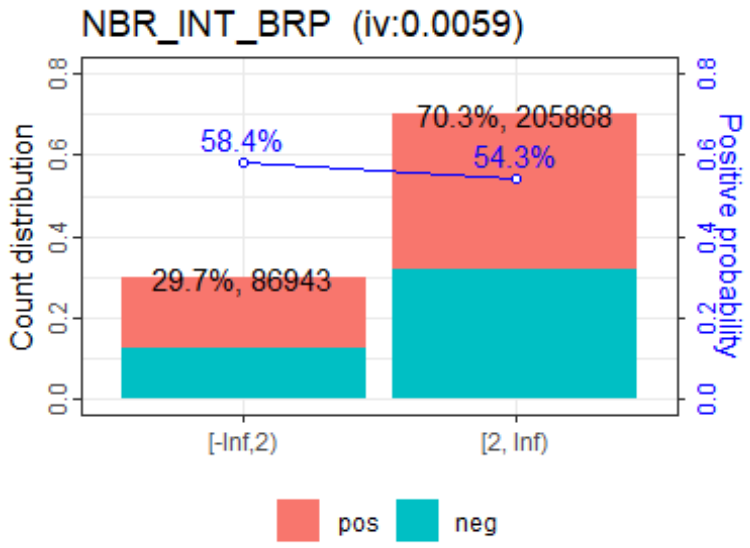
```
# Poids des variables pour chaque observation dans la base de données
base_up_woe <- woebin_ply(base_up, bins)

## ✓ Woe transformating on 292811 rows and 25 columns in 00:00:05

# Affichage des graphiques
woebin_plot(bins)

## $ASU_BIEN_FIN_BRP
```



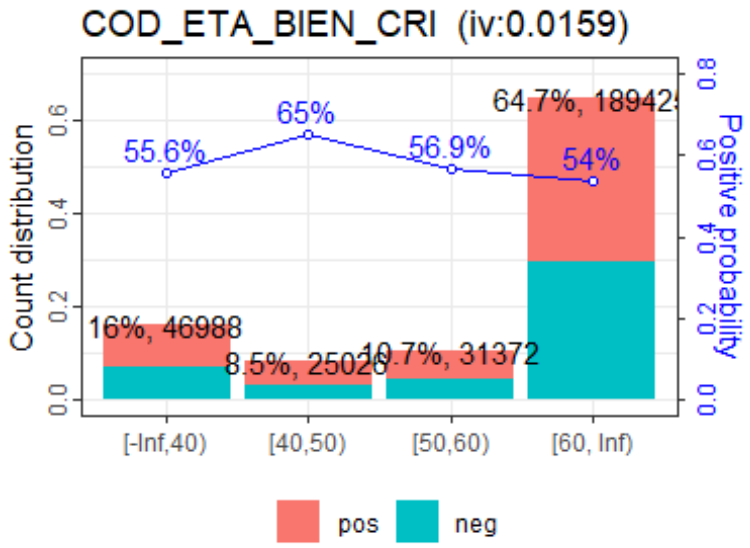

```
##
## $NAT_BIEN_FIN_BRP
```



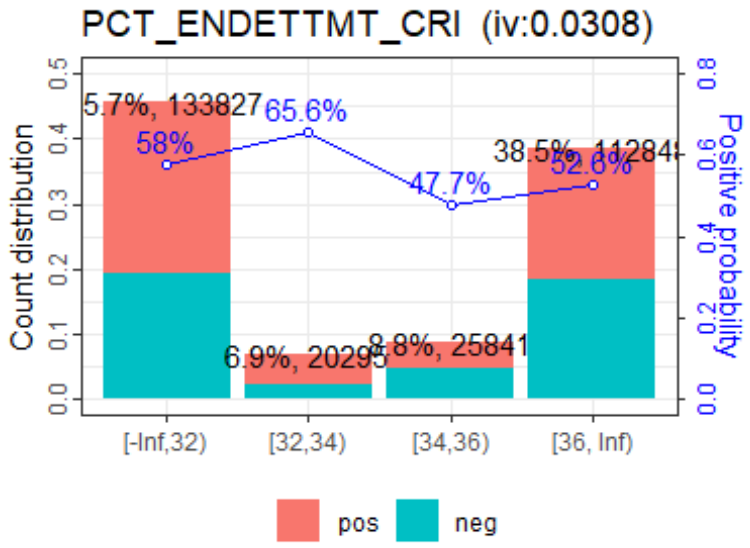

```
##
## $NBR_INT_BRP
```



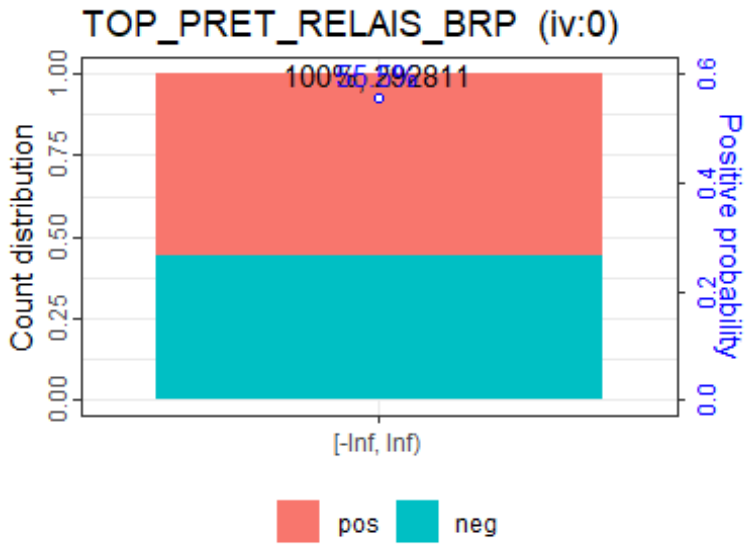

```
##
## $COD_ETA_BIEN_CRI
```



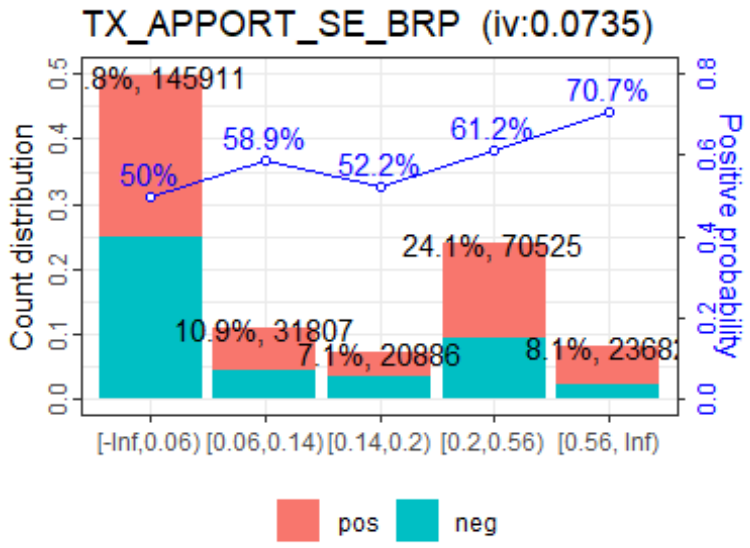

```
##
## $PCT_ENDETTMT_CRI
```



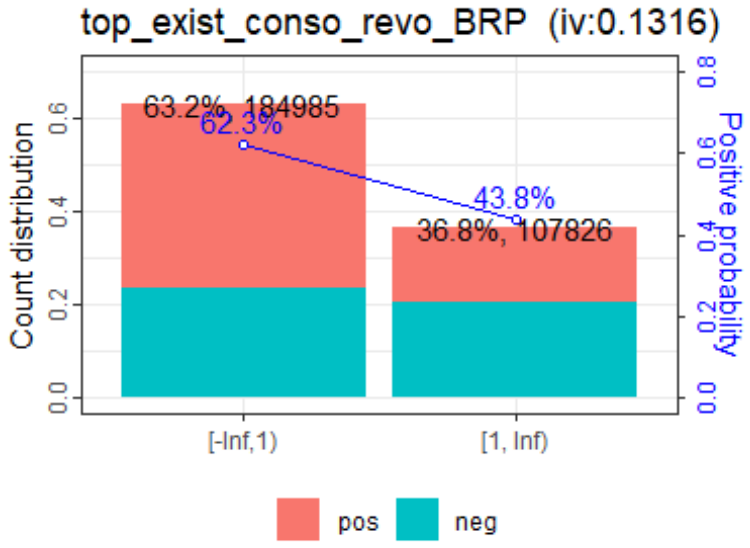

```
##
## $TOP_PRET_RELAIS_BRP
```



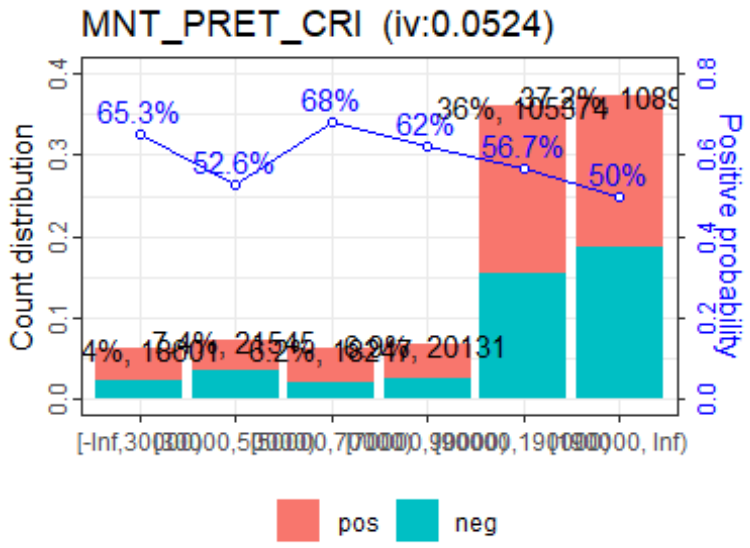

```
##
## $TX_APPORT_SE_BRP
```



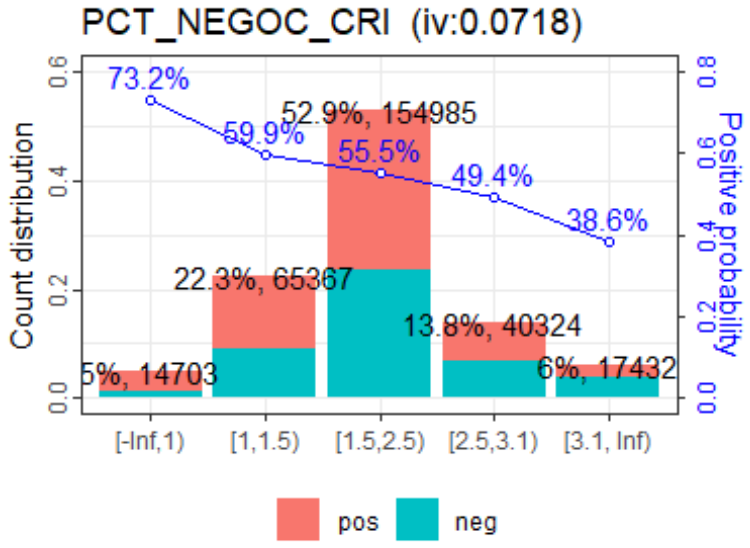

```
##
## $top_exist_conso_revo_BRP
```



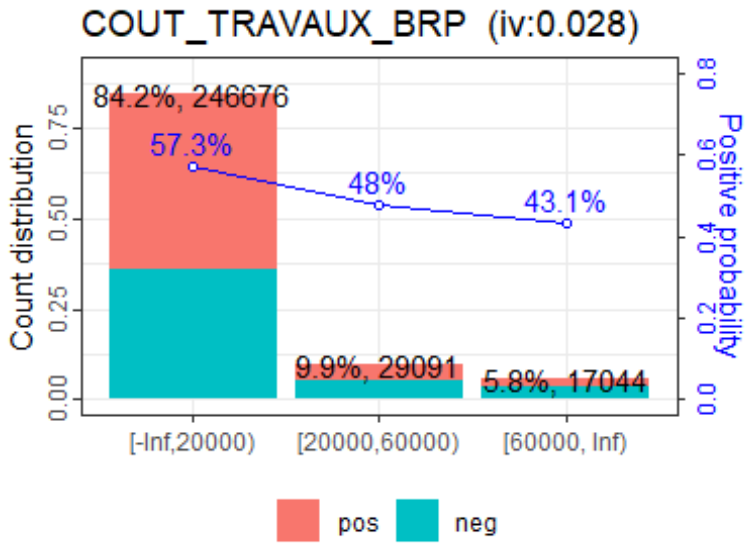

```
##
## $MNT_PRET_CRI
```



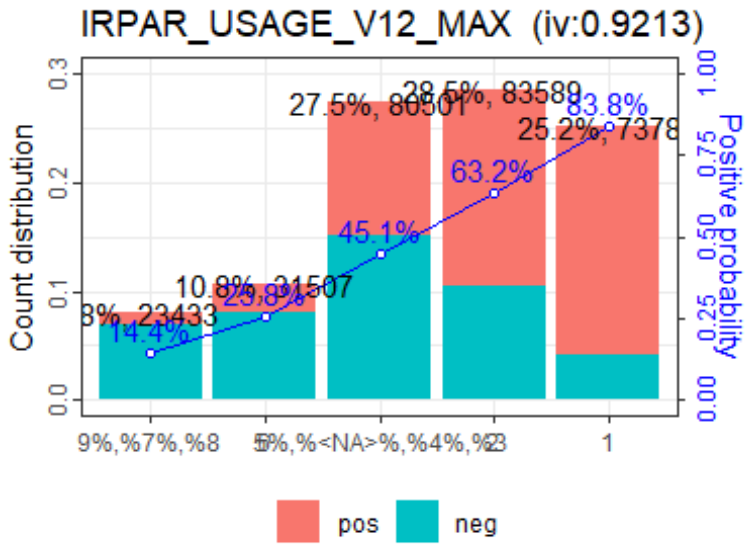

```
##
## $PCT_NEGOC_CRI
```



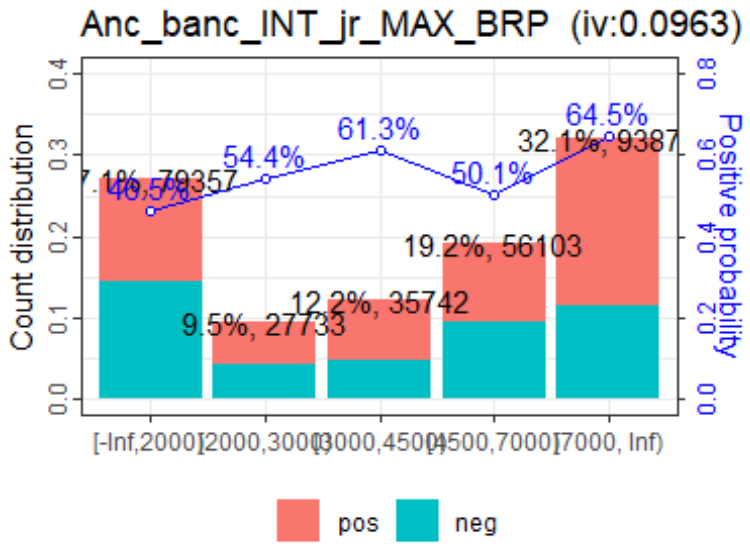

```
##
## $COUT_TRAVAUX_BRP
```



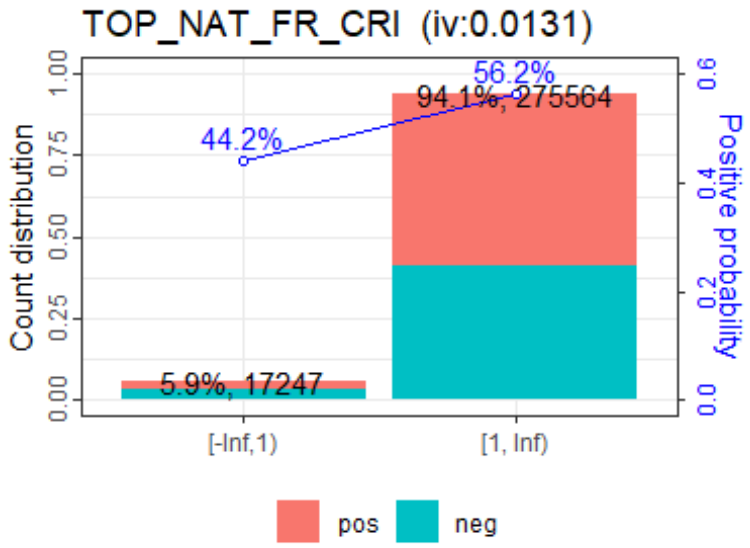

```
##
## $IRPAR_USAGE_V12_MAX
```



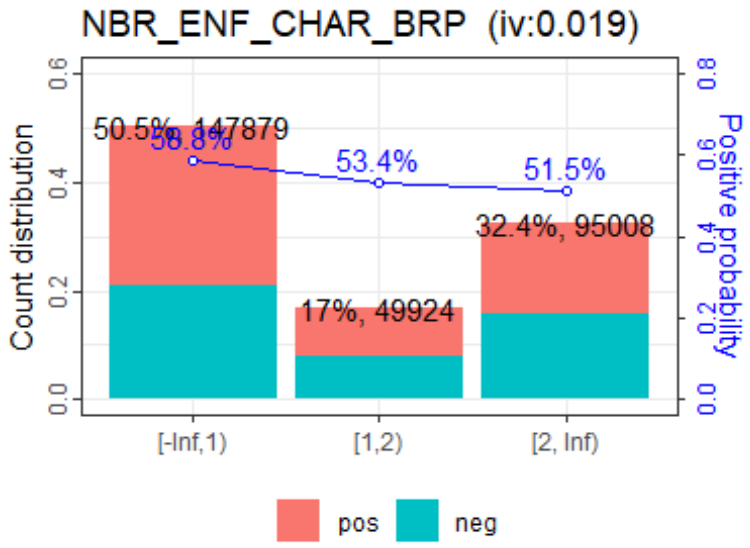

```
##
## $Anc_banc_INT_jr_MAX_BRP
```



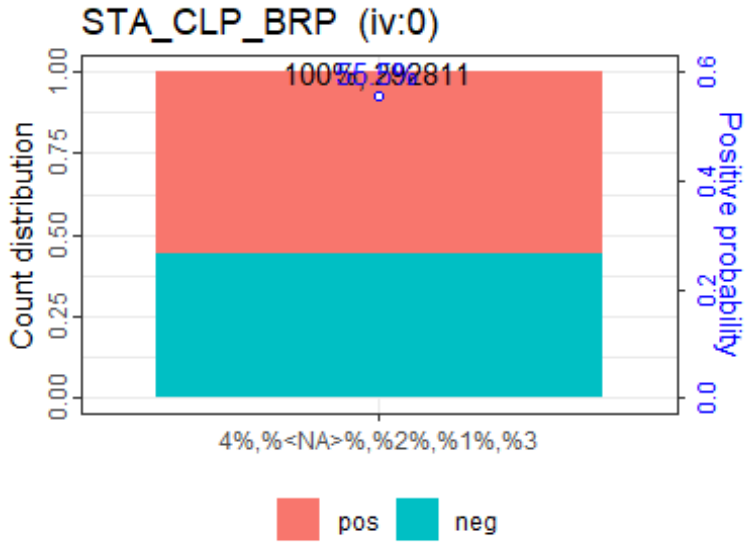

```
##
## $TOP_NAT_FR_CRI
```



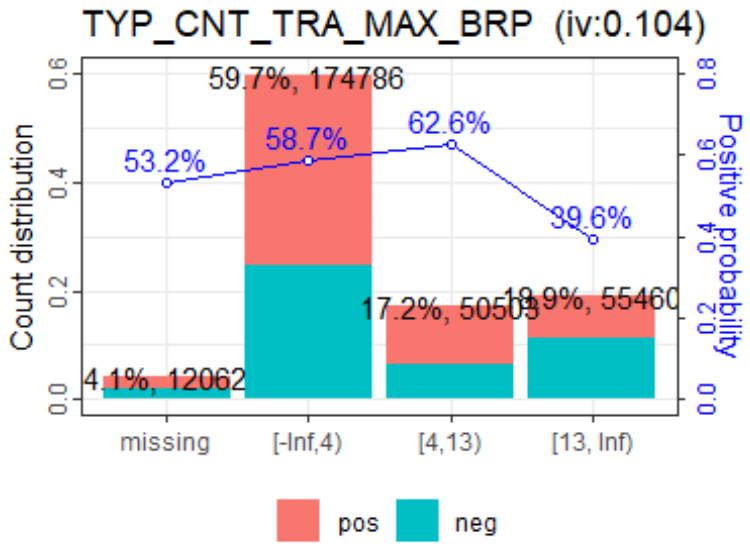

```
##
## $NBR_ENF_CHAR_BRP
```



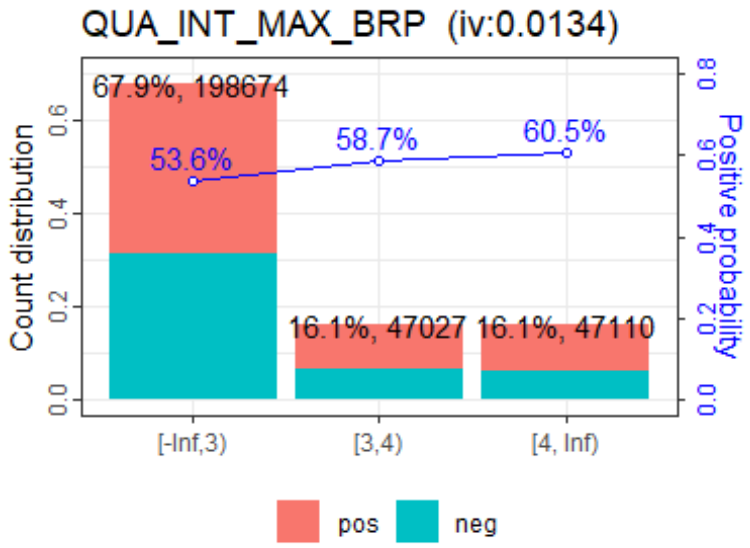

```
##
## $STA_CLP_BRP
```



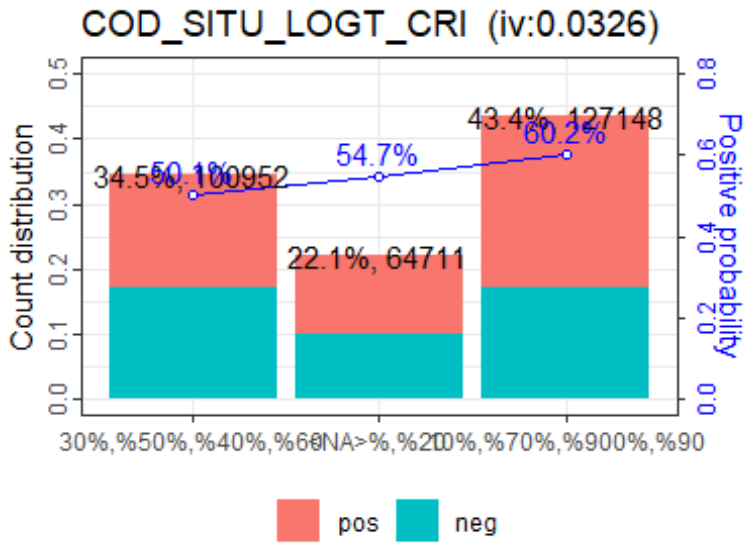

```
##
## $TYP_CNT_TRA_MAX_BRP
```



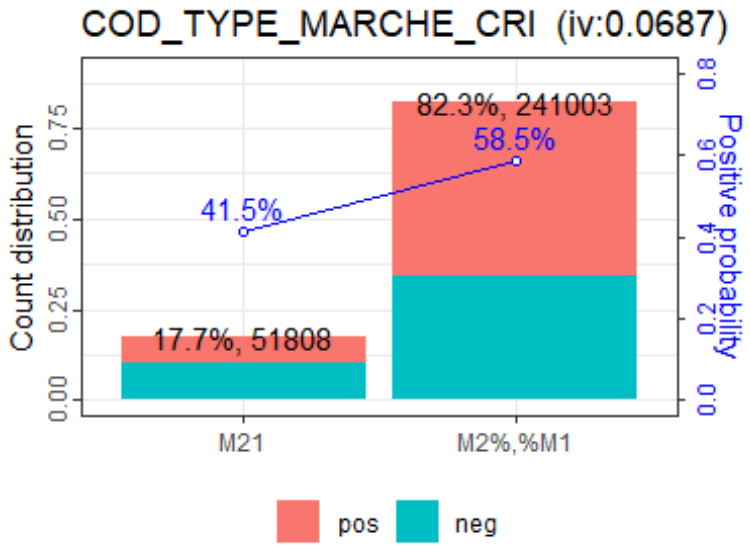

```
##
## $QUA_INT_MAX_BRP
```



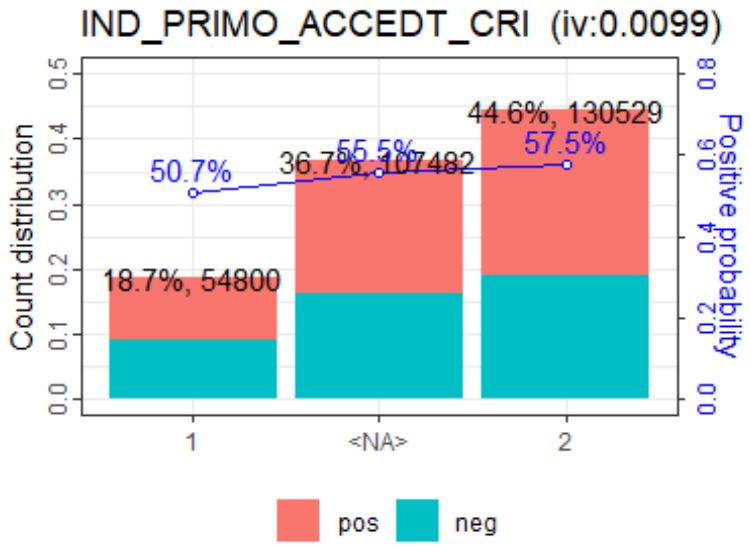

```
##
## $COD_SITU_LOGT_CRI
```



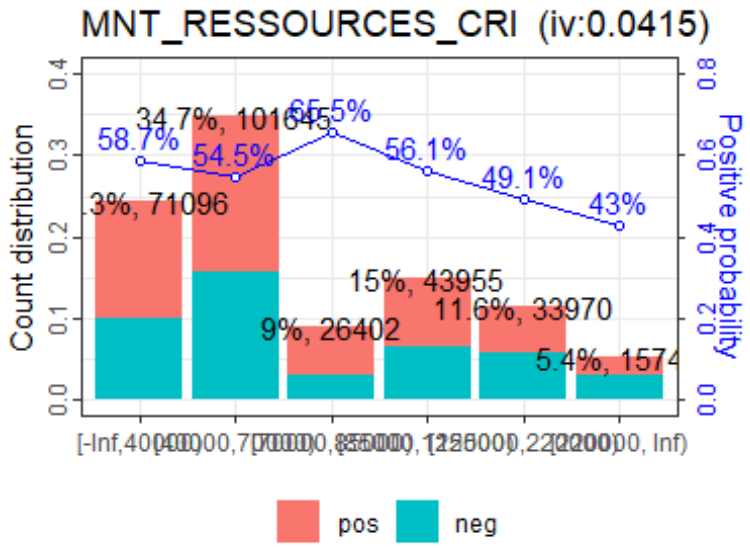

```
##
## $COD_TYPE_MARCHE_CRI
```



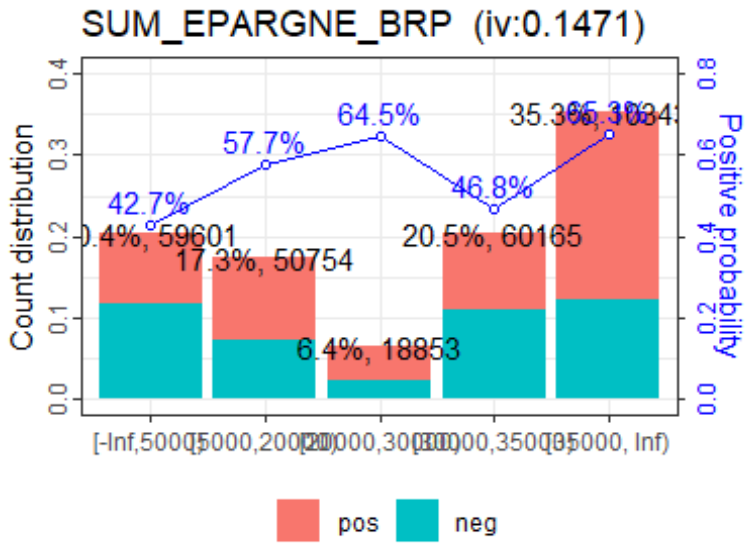

```
##
## $IND_PRIMO_ACCEDT_CRI
```



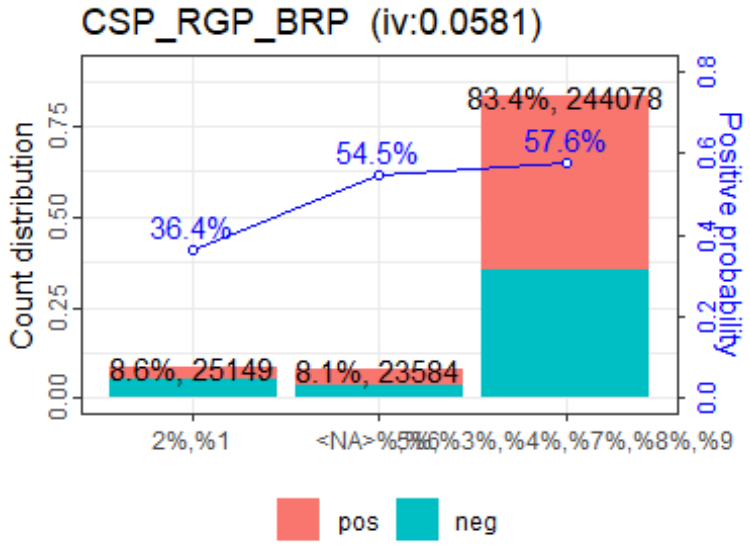

```
##
## $MNT_RESSOURCES_CRI
```



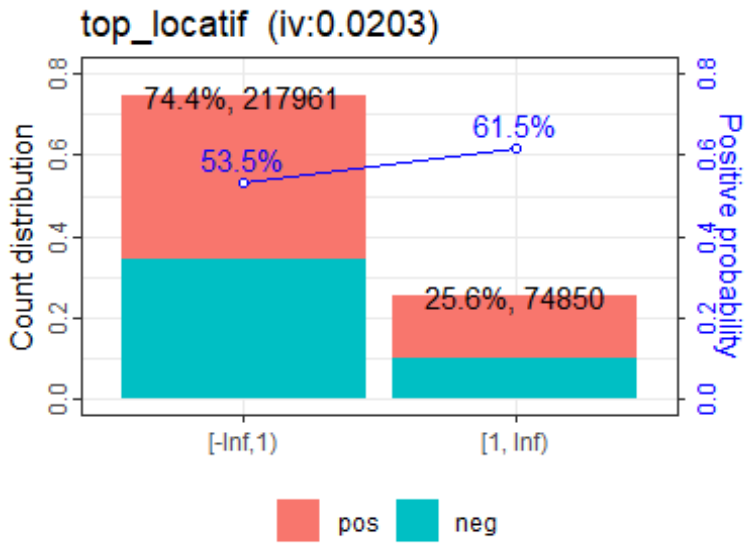

```
##
## $SUM_EPARGNE_BRP
```




```
##
## $CSP_RGP_BRP
```




```
##
## $top_locatif
```


top_locatif (iv:0.0203)
74.4%, 217961
53.5%
61.5%
25.6%, 74850
Count distribution
Positive probability
0.0
0.2
0.4
0.6
0.8
[-Inf,1)
[1, Inf)
pos
neg

```r
# Enregistrement des nouvelles variables discrétisées
bins_df = data.table::rbindlist(bins)
kable(distinct(bins_df[ ,  c("variable", "total_iv")]))
```

variable	total_iv
ASU_BIEN_FIN_BRP	0.0221174
NAT_BIEN_FIN_BRP	0.0440220
NBR_INT_BRP	0.0058552
COD_ETA_BIEN_CRI	0.0159109
PCT_ENDETTMT_CRI	0.0308087
TOP_PRET_RELAIS_BRP	0.0000000
TX_APPORT_SE_BRP	0.0734922
top_exist_conso_revo_BRP	0.1315756
MNT_PRET_CRI	0.0524446
PCT_NEGOC_CRI	0.0717984
COUT_TRAVAUX_BRP	0.0279640
IRPAR_USAGE_V12_MAX	0.9212856
Anc_banc_INT_jr_MAX_BRP	0.0962552
TOP_NAT_FR_CRI	0.0130991
NBR_ENF_CHAR_BRP	0.0189544
STA_CLP_BRP	0.0000000
TYP_CNT_TRA_MAX_BRP	0.1040163
QUA_INT_MAX_BRP	0.0133738
COD_SITU_LOGT_CRI	0.0326329
COD_TYPE_MARCHE_CRI	0.0686509
IND_PRIMO_ACCEDT_CRI	0.0098971
MNT_RESSOURCES_CRI	0.0414620
SUM_EPARGNE_BRP	0.1470644
CSP_RGP_BRP	0.0581021
top_locatif	0.0203101