Eliott Loop
Grégoire de Crombrugghe

TP-04 DataMining

Introduction

Lors du deuxième TP, il nous a été demandé d'analyser une seconde fois les différents attributs mais, cette fois-ci, avec une approche quantitative. En effet, les consignes était d'implémenter le code R calculant l'Entropie, l'Entropie Conditionnel et l'information gain. Et d'ensuite analyser certaines données.

Pour ce quatrième TP, nous devons utiliser la méthode de classification Naive Bayes. Cette méthode permet de construire un modèle de prédiction. Le but de ce TP est de comprendre comment Naive Bayes fonctionne. Sur base de cette méthode, nous devons construire 5 modèles avec une séparation, du jeux de données, différentes. Ainsi que d'appliquer la méthode sur la totalité des données.

Description du Problème

Une Banque, nous demande de réaliser un modèle qui permettra de déterminer quel produit d'investissement à proposer à un client, suivant son profil. La Banque nous met à disposition une série de données, qui doit être analysée afin de déterminer les attributs dits "utiles" et de pouvoir construire ce modèle.

Attribut continu: SE1,BA1-BA7
Attribut discret: SE2,PE1-PE15, IA1-IA3
Variable cible: InvType

Eliott Loop
Grégoire de Crombrugghe

Données Obtenu

Modèle à partire de Naive Bayes	Accuracy	Default_Classifier	Accuracy-Default_Classifier
modèle 1	0.5760456	0.5038023	0.07224335
modèle 2	0.5931559	0.5104563	0.08269962
modèle 3	0.5963245	0.5101394	0.08618504
modèle 4	0.5893536	0.5072877	0.08206591
modèle 5	0.5766793	0.5139417	0.06273764
	Average Accuracy:	0.5863118	

En moyenne,les différents modèles prédisent la bonne classe dans 58,63% des cas.

Pour pouvoir comparer nos modèles nous les comparons au *default classifier ,*qui est la probabilité de la classe majoritaire du jeu de données d'entraînement.
Par exemple pour notre premier modèle on constate que le modèle Naive Bayes a une fiabilité supérieure de 7,22% comparé au *default classifier.*

Analyse modèle final

Pour expliquer le modèle, Naive bayes, appliqué sur la totalité des données, nous utiliserons 1 attribut discret et 1 attribut continu.

Attribut discret

PE15	I0	I1
C0	0.95470233	0.04529767
C1	0.92508278	0.07491722

Pour chaque attribut discret, Naive Bayes va calculer les probabilité Conditionnel. Cette probabilité conditionnelle permet à Naive Bayes, de calculer la formule de bayes ($P(A|B) = P(B|A) * P(A)/P(B)$, B étant l'attribut cible. Avec $P(B|A) = P(b_1|A) * P(b_2|A)$ *…). L'utilité de cette formule sera développée plus tard dans ce rapport.

Eliott Loop
Grégoire de Crombrugghe

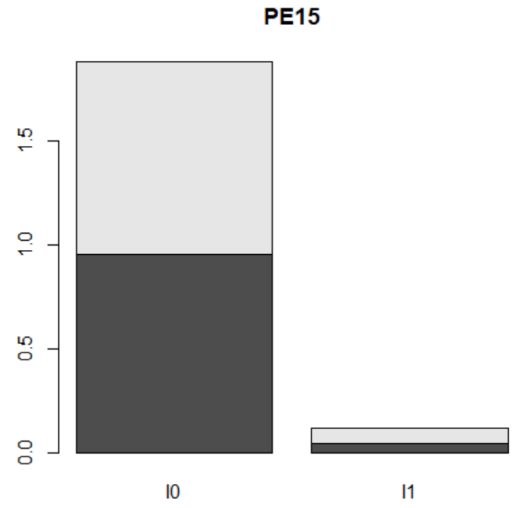
Attribut Continu

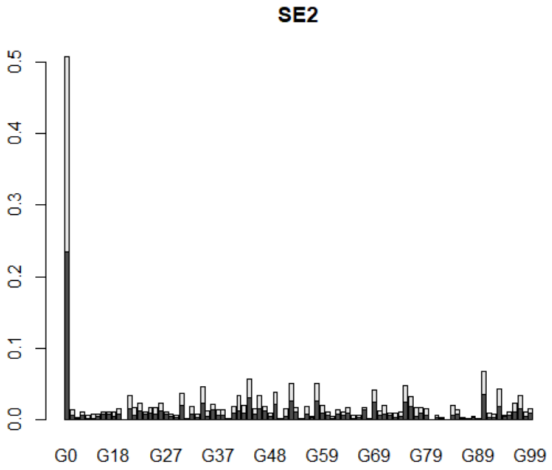
BA1	conditional mean	écart-type
C0	5.393874	8.26418
C1	4.335265	6.59633

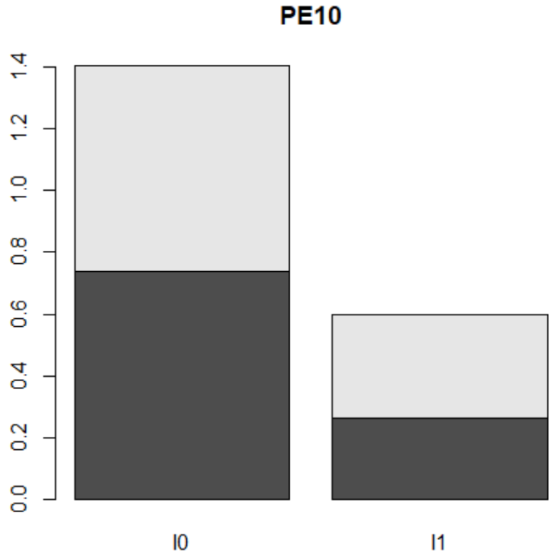
Pour chaque attribut continu, Naive Bayes va calculer le *conditionnal mean* et l'écart-type. Ceux-ci permettent à naive Bayes, d 'estimer une probabilité en utilisant la fonction de densité: $N\ (x|m,\ s)\ =\ (1/(s\ \sqrt{2\pi})\ )\ *\ e^{-(x-m)^2/2s}$. L'utilité de cette formule sera développée plus tard dans ce rapport. (Dans la Suite du rapport, lorsque sera utilisé le mot moyenne nous ferons référence à conditional mean)

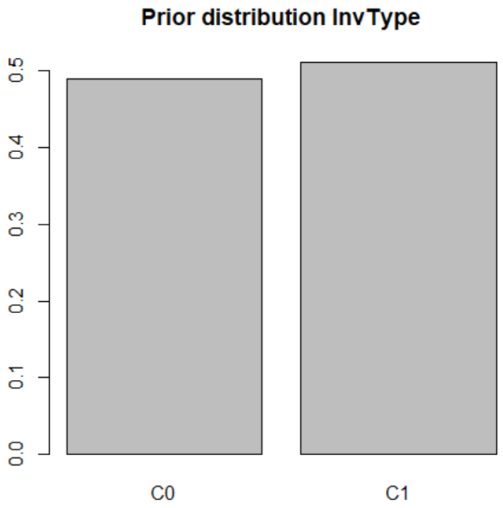
En plus des statistiques spécifiques pour les attributs discrets et continus, la probabilité antérieure de la classe est aussi utilisée pour classifier une instance.

Explication de la distribution conditionnelle pour la classificati









Eliott Loop
Grégoire de Crombrugghe

Gris clair = C1 & Gris foncé = C0

Les graphiques représentent la distribution conditionnelle des trois attributs discrets. Lors de la classification d'une instance , Naive Bayes va multiplier la probabilité conditionnelle de chaque attribut. Mais nous pouvons observer que si, par exemple nous avions eu une distribution pour l'attribut PE10 de 100% C0 et 0% C1 (pour PE10=I0). Dès qu'une nouvelle instance avec comme valeur I0 pour PE1 est introduite, Naive Bayes prédit automatiquement C0, car la méthode Naive Bayes multiplie les probabilités et une multiplication par 0 est égale à 0.

Classification d'une instance

Voici comment, nous pensons que Naive Bayes fonctionne. Pour aider la compréhension de notre explication nous prenons comme exemple la première ligne de donnée comme instance :

```
myData[1,]
SE1 SE2 BA1 BA2  BA3 BA4 BA5 BA6 BA7 PE1 PE2 PE3 PE4 PE5 PE6 PE7 PE8 PE9
45 G29  12   0 5934   0   0   0   0  I0  I0  I0  I0  I0  I0  I0  I0 I0
PE10 PE11 PE12 PE13 PE14 PE15 IA1 IA2 IA3 InvType
I0   I0   I0   I0   I0 1    I0   0   0   1      C1
```

Naive Bayes va multiplier les probabilités de chaque attribut pour les différentes classes de la variable cible.

=> P(C1)*P(SE1=45|C1)*P(SE2=G29|C1)*P(BA1=12|C1)*...=X (ligne1)

=>P(C0)*P(SE1=45|C0)*P(SE2=G29|C0)*P(BA1=12|C0)*...=Y (ligne2)

Lorsque l'attribut est un attribut discret c'est la formule de bayes qui est utilisée pour calculer la probabilité. Si au contraire, l'attribut est continu, la méthode utilisera la fonction de densité. Cette fonction utilise la moyenne et l'écart-type pour calculer la probabilité. La méthode multiplie ensuite toutes les probabilités, des attributs pour un produit, ainsi que la *prior probability* de la variable cible. En effet, cette dernière est aussi utilisée pour classifier une instance.

Si X>Y alors Naive Bayes prédit le produit d'investissement C1. Et inversement, si X<Y, alors c'est le produit C0 que Naive Bayes prédit.

Eliott Loop
Grégoire de Crombrugghe

Conclusion

En conclusion, la méthode Naive Bayes doit, suivant le type de l'attribut, calculer ou bien la formule de Bayes (et donc la probabilité conditionnel) ou bien la fonction de densité (et donc la moyenne et l'écart-type). Après l'exécution de notre programme, celui-ci "sort" 5 modèles différents de classification avec une précision de 58.63% en moyenne.