



An abstract background featuring a dense, colorful pattern of overlapping circles in shades of yellow, green, blue, and orange, creating a heatmap-like effect.

ANALYSE ET VISUALISATION DES DONNÉES

Alexandra-Maria DOBRESCU

| Cours 5 |

Collection et préparation des données

- SOURCES ET ACQUISITION DES DONNÉES
- NETTOYAGE ET PRÉTRAITEMENT DES DONNÉES
- TYPES DE DONNÉES
- LA MESURE

Exercice Classique



...

Attributes				Classes
Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Source [3]

Metttons les dernières touches

ALGORITHMES

Arbres de décision

$$\text{Entropie}(\text{PlayGolf}, \text{Outlook}) = 0.693$$

$$\text{Entropie}(\text{PlayGolf}, \text{Temperature}) = 0.91$$

$$\text{Entropie}(\text{PlayGolf}, \text{Humidity}) = 0.79$$

$$\text{Entropie}(\text{PlayGolf}, \text{Wind}) = 0.89$$

$$\text{Entropie}(\text{PlayGolf}) = 0.94$$



Metttons les dernières touches

ALGORITHMES

Arbres de décision

$$\text{Entropie}(\text{PlayGolf}, \text{Outlook}) = 0.693$$

$$\text{Entropie}(\text{PlayGolf}, \text{Temperature}) = 0.91$$

$$\text{Entropie}(\text{PlayGolf}, \text{Humidity}) = 0.79$$

$$\text{Entropie}(\text{PlayGolf}, \text{Wind}) = 0.89$$

$$\text{Entropie}(\text{PlayGolf}) = 0.94$$

Attribut	Entropie	Gain
Outlook	0.69	0.25
Temperature	0.91	0.03
Humidity	0.79	0.15
Wind	0.89	0.05

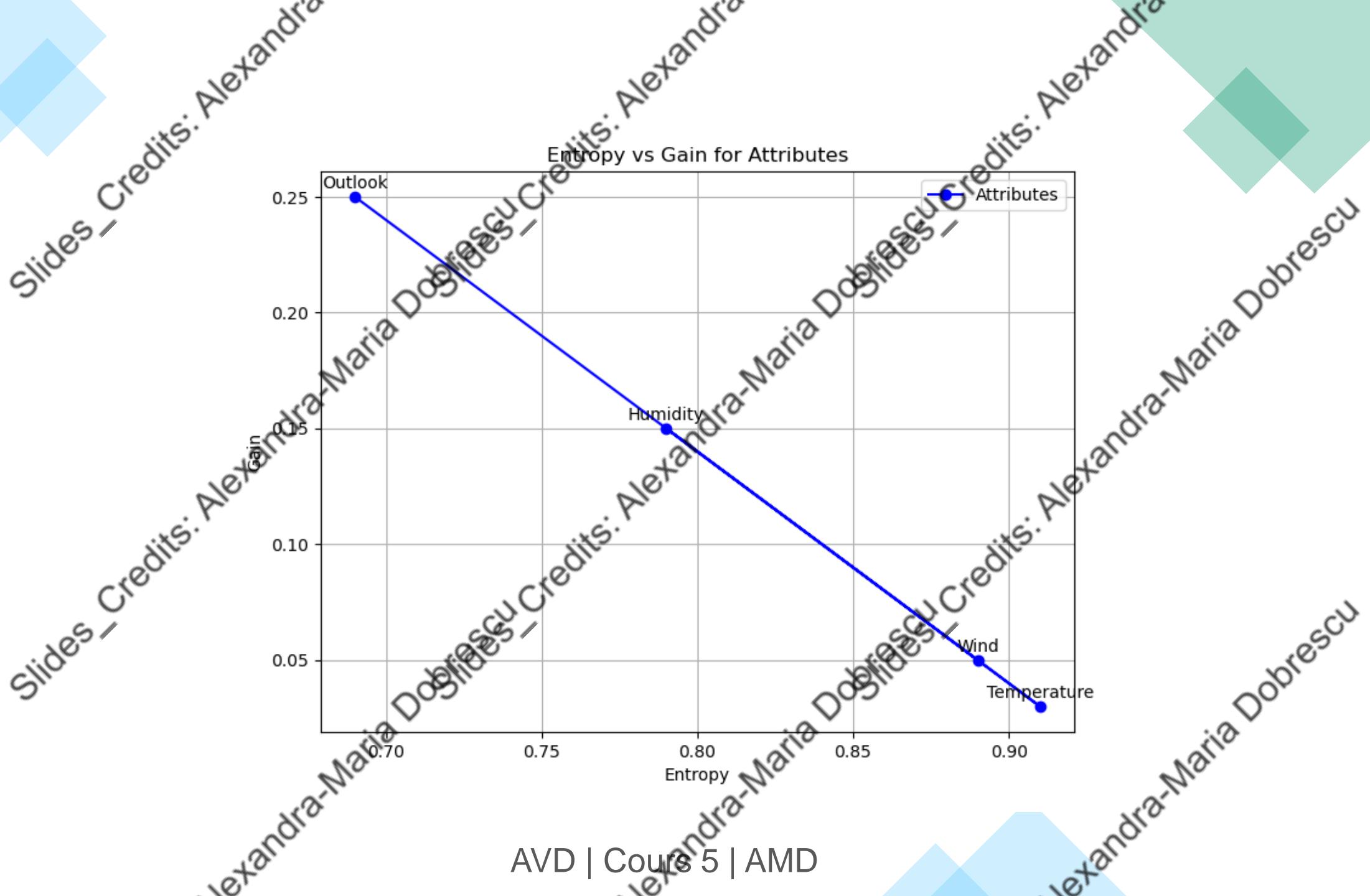
Metttons les dernières touches

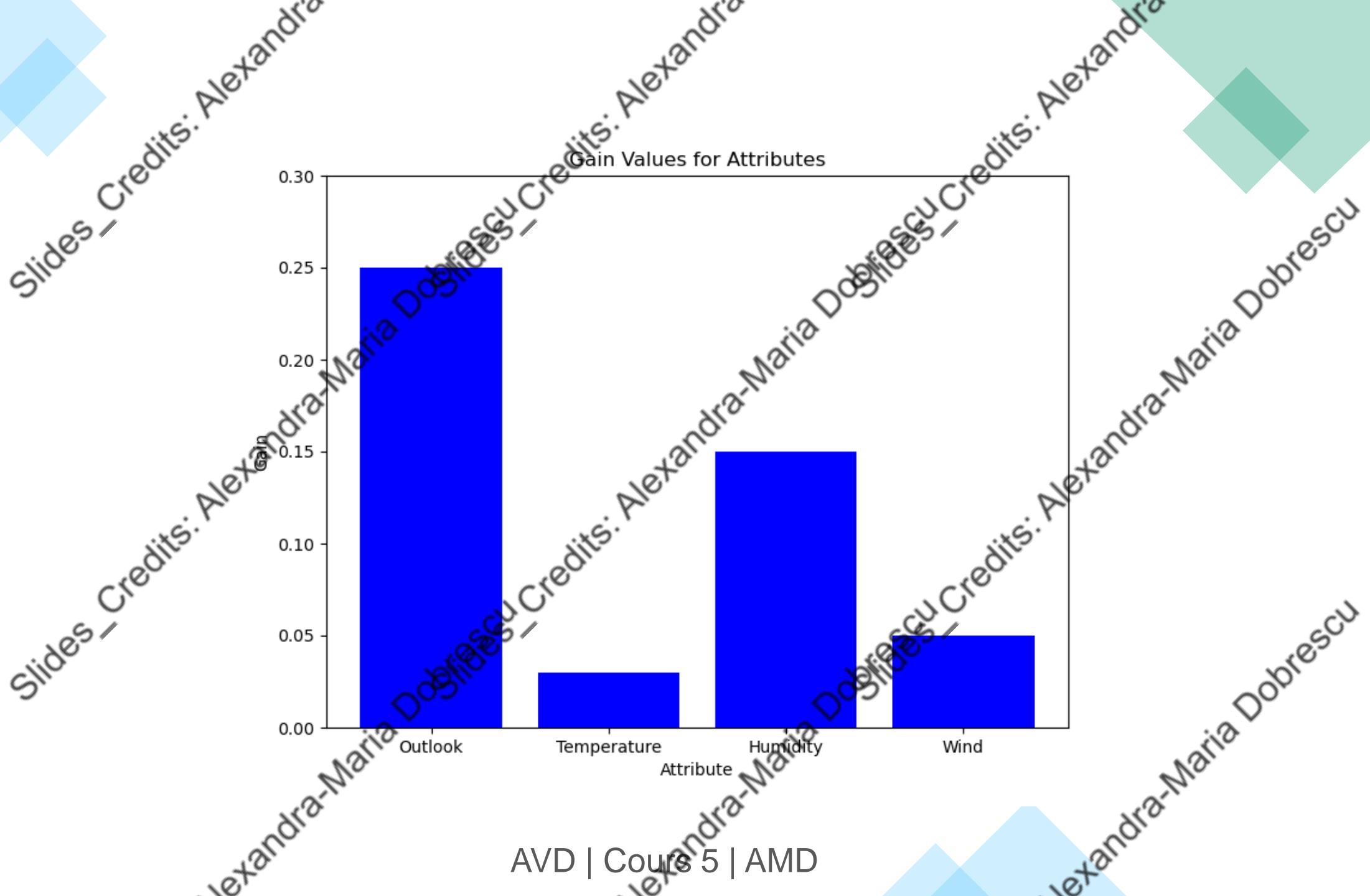
ALGORITHMES

Arbres de décision

Attribut	Entropie	Gain
Outlook	0.69	0.25
Temperature	0.91	0.03
Humidity	0.79	0.15
Wind	0.89	0.05

?





Conclusions de l'exercice

ALGORITHMES

Arbres de décision

SEULEMENT EN ANALYSANT LES VALEURS D'ENTROPIE ET DE GAIN, IL EST POSSIBLE DE NE PAS OBTENIR DE MODÈLES CLAIRS À PARTIR DE CES VALEURS.

CEPENDANT, CES VALEURS SONT GÉNÉRALEMENT UTILISÉES DANS LE CONTEXTE DES ARBRES DE DÉCISION OU DE L'APPRENTISSAGE AUTOMATIQUE AFIN DE SÉLECTIONNER LE MEILLEUR ATTRIBUT POUR DIVISER LES DONNÉES ET PRENDRE DES DÉCISIONS.

Conclusions de l'exercice

ALGORITHMES

Arbres de décision

Les attributs ayant un gain plus élevé et une entropie plus faible sont plus informatifs et donnent lieu à de meilleures répartitions dans les algorithmes d'arbre de décision.

Gain plus élevé: Les attributs dont le gain est élevé (plus proche de 1) fournissent plus d'informations lorsqu'ils sont utilisés pour diviser une décision. Cela signifie qu'ils séparent mieux les données en différentes classes ou catégories.

Entropie plus faible: Une entropie plus faible (proche de 0) indique qu'un attribut entraîne moins d'incertitude ou de caractère aléatoire dans les données. Dans la terminologie des arbres de décision, les attributs à faible entropie sont meilleurs pour les divisions « pures », où la plupart des données des sous-ensembles résultants appartiennent à la même classe.

Conclusions de l'exercice

ALGORITHMES

Arbres de décision

Discussion : Comment fournissons-nous nos informations ?

- Parmi les valeurs fournies, « Température » a le gain le plus faible (0,03) et l'entropie la plus élevée (0,91), ce qui suggère qu'il pourrait s'agir de l'attribut le moins informatif pour la prise de décision.
- D'autre part, « Outlook » a le gain le plus élevé (0,25) et une entropie relativement plus faible (0,69), ce qui indique qu'il pourrait être plus utile pour diviser les données de manière efficace.

Discrétisation des données

ALGORITHMES

La forêt aléatoire (Random Forest)

{ RANDOM FOREST, UN ALGORITHME D'APPRENTISSAGE ENSEMBLISTE, PEUT TRAITER À LA FOIS DES DONNÉES CONTINUES ET DISCRÈTES. CEPENDANT, LA DISCRÉTISATION PEUT ÊTRE UTILE DANS LES CAS OÙ IL EST DIFFICILE DE SAISIR DES MODÈLES COMPLEXES AVEC DES DONNÉES CONTINUES. }

Construit plusieurs arbres de décision au cours de la formation et combine leurs prédictions afin d'obtenir une précision et une robustesse plus élevées.

Il convient à la fois aux tâches de régression (prédiction de valeurs numériques) et de classification (classement des données en classes).

Discrétisation des données

ALGORITHMES

La forêt aléatoire
(Random Forest)

{ LA FORÊT ALÉATOIRE EST UN CLASSIFICATEUR D'ENSEMBLE COMPOSÉ D'UN ENSEMBLE D'ARBRES DE DÉCISION. LA SORTIE DU CLASSIFICATEUR FINAL EST LA VALEUR MODALE DES CLASSES PRODUITES PAR CHAQUE ARBRE. }

Chaque arbre de la forêt est construit en utilisant un sous-ensemble aléatoire des données et un sous-ensemble aléatoire des caractéristiques, ce qui réduit le risque de *overfitting* et améliore la généralisation.

La prédiction finale dans une forêt aléatoire est généralement faite en prenant un vote majoritaire (pour la classification) ou une moyenne (pour la régression) des prédictions des arbres individuels.

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \dots, (x_n, y_n)$, features F , and number of trees in forest B .

```
1 function RANDOMFOREST( $S, F$ )
2    $H \leftarrow \emptyset$ 
3   for  $i \in 1, \dots, B$  do
4      $S^{(i)} \leftarrow$  A bootstrap sample from  $S$ 
5      $h_i \leftarrow$  RANDOMIZEDTREELEARN( $S^{(i)}, F$ )
6      $H \leftarrow H \cup \{h_i\}$ 
7   end for
8   return  $H$ 
9 end function
10 function RANDOMIZEDTREELEARN( $S, F$ )
11   At each node:
12      $f \leftarrow$  very small subset of  $F$ 
13     Split on best feature in  $f$ 
14   return The learned tree
15 end function
```

Choisir B - nombre d'arbres « à cultiver »

Source [1]

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \dots, (x_n, y_n)$, features F , and number of trees in forest B .

```
1 function RANDOMFOREST( $S, F$ )
2    $H \leftarrow \emptyset$ 
3   for  $i \in 1, \dots, B$  do
4      $S^{(i)} \leftarrow$  A bootstrap sample from  $S$ 
5      $h_i \leftarrow$  RANDOMIZEDTREELEARN( $S^{(i)}, F$ )
6      $H \leftarrow H \cup \{h_i\}$ 
7   end for
8   return  $H$ 
9 end function
10 function RANDOMIZEDTREELEARN( $S, F$ )
11   At each node:
12      $f \leftarrow$  very small subset of  $F$ 
13     Split on best feature in  $f$ 
14   return The learned tree
15 end function
```

F - ensemble de caractéristiques

Choisissez $f << F$ - nombre de variables utilisées pour diviser chaque nœud.

Source [1]

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \dots, (x_n, y_n)$, features F , and number of trees in forest B .

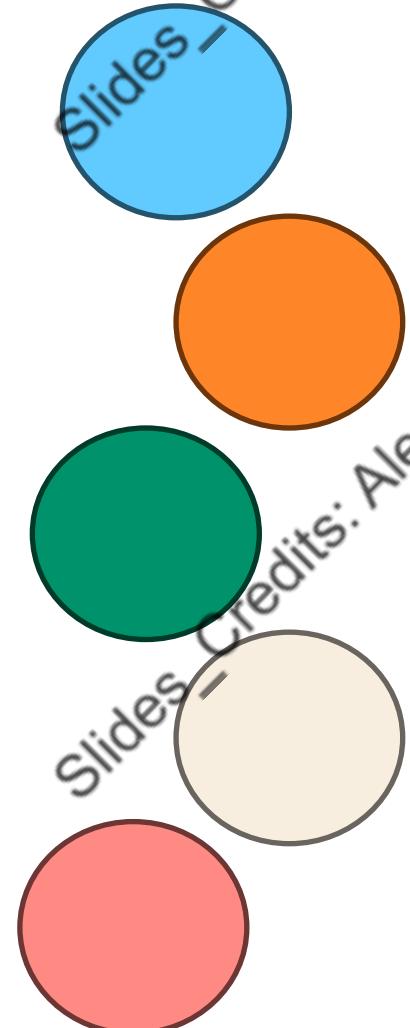
```
1 function RANDOMFOREST( $S, F$ )
2    $H \leftarrow \emptyset$ 
3   for  $i \in 1, \dots, B$  do
4      $S^{(i)} \leftarrow$  A bootstrap sample from  $S$ 
5      $h_i \leftarrow$  RANDOMIZEDTREELEARN( $S^{(i)}, F$ )
6      $H \leftarrow H \cup \{h_i\}$ 
7   end for
8   return  $H$ 
9 end function
10 function RANDOMIZEDTREELEARN( $S, F$ )
11   At each node:
12      $f \leftarrow$  very small subset of  $F$ 
13     Split on best feature in  $f$ 
14   return The learned tree
15 end function
```

Construisez un échantillon **bootstrap** à partir des données d'apprentissage **avec remplacement**.

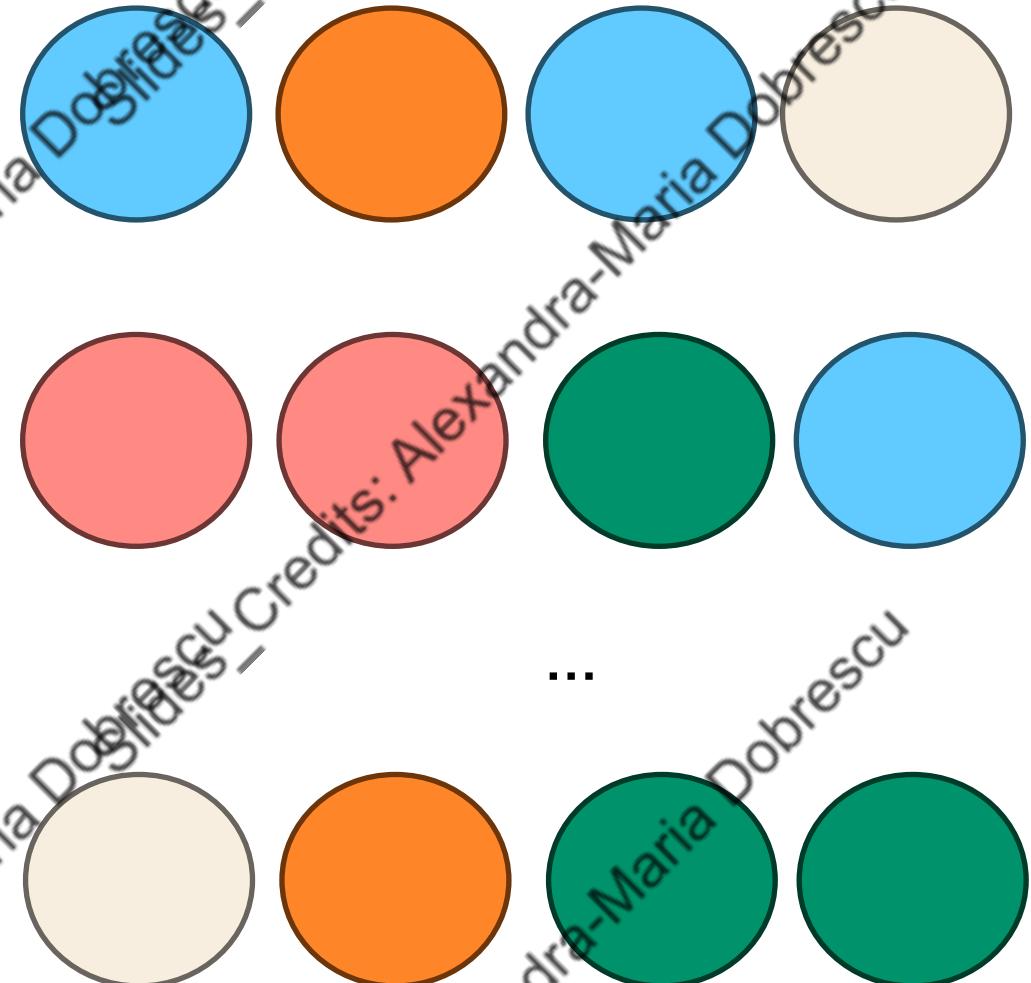
Source [1]

Bootstrap avec Remplacement

Données observées



Échantillons bootstrapés



Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \dots, (x_n, y_n)$, features F , and number of trees in forest B .

```
1 function RANDOMFOREST( $S, F$ )
2    $H \leftarrow \emptyset$ 
3   for  $i \in 1, \dots, B$  do
4      $S^{(i)} \leftarrow$  A bootstrap sample from  $S$ 
5      $h_i \leftarrow$  RANDOMIZEDTREELEARN( $S^{(i)}, F$ )
6      $H \leftarrow H \cup \{h_i\}$ 
7   end for
8   return  $H$ 
9 end function
10 function RANDOMIZEDTREELEARN( $S, F$ )
11   At each node:
12      $f \leftarrow$  very small subset of  $F$ 
13     Split on best feature in  $f$ 
14   return The learned tree
15 end function
```

Lors de la croissance d'un arbre, à chaque nœud, sélectionnez f variables au aléatoire et utilisez-les pour trouver la meilleure division.

Source [1]

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \dots, (x_n, y_n)$, features F , and number of trees in forest B .

```
1 function RANDOMFOREST( $S, F$ )
2    $H \leftarrow \emptyset$ 
3   for  $i \in 1, \dots, B$  do
4      $S^{(i)} \leftarrow$  A bootstrap sample from  $S$ 
5      $h_i \leftarrow$  RANDOMIZEDTREELEARN( $S^{(i)}, F$ )
6      $H \leftarrow H \cup \{h_i\}$ 
7   end for
8   return  $H$ 
9 end function
10 function RANDOMIZEDTREELEARN( $S, F$ )
11   At each node:
12      $f \leftarrow$  very small subset of  $F$ 
13     Split on best feature in  $f$ 
14   return The learned tree
15 end function
```

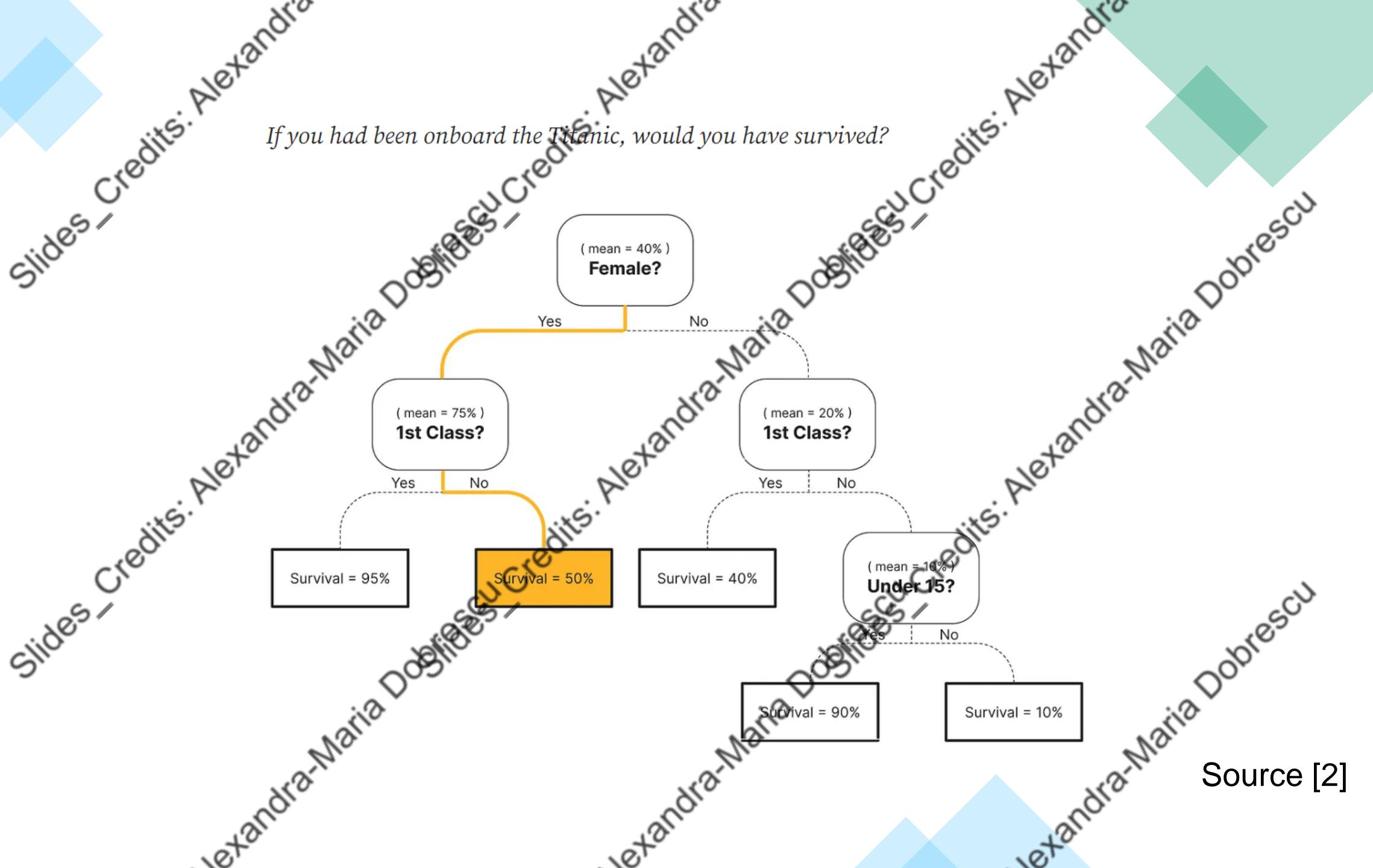
Faire pousser l'arbre au maximum.
Il n'y a pas de réduction de la taille (*pruning*).

Algorithm 1 Random Forest

Precondition: A training set $S := (x_1, y_1), \dots, (x_n, y_n)$, features F , and number of trees in forest B .

```
1 function RANDOMFOREST( $S, F$ )
2    $H \leftarrow \emptyset$ 
3   for  $i \in 1, \dots, B$  do
4      $S^{(i)} \leftarrow$  A bootstrap sample from  $S$ 
5      $h_i \leftarrow$  RANDOMIZEDTREELEARN( $S^{(i)}, F$ )
6      $H \leftarrow H \cup \{h_i\}$ 
7   end for
8   return  $H$ 
9 end function
10 function RANDOMIZEDTREELEARN( $S, F$ )
11   At each node:
12      $f \leftarrow$  very small subset of  $F$ 
13     Split on best feature in  $f$ 
14   return The learned tree
15 end function
```

Prédire de nouvelles données en agrégeant les prédictions des arbres (Les votes majoritaires pour la classification, moyenne pour la régression).



Discrétisation des données

ALGORITHMES

La forêt aléatoire
(Random Forest)

UTILISATION DE DONNÉES DISCRÈTES AVEC RANDOM FOREST

Random Forest peut traiter des données discrètes et continues.

La capacité de Random Forest à diviser les données en sous-ensembles et à évaluer les caractéristiques le rend bien adapté aux problèmes de types de données mixtes.

Lorsque vous travaillez avec des données discrètes, vous pouvez coder les caractéristiques catégorielles sous forme de valeurs numériques ou utiliser un codage à un seul point (one-hot encoding) pour représenter les différentes catégories.

Il est important de prétraiter et de préparer vos données afin de s'assurer que l'encodage des variables discrètes est effectué correctement et de manière cohérente..

Discrétisation des données

ALGORITHMES

Algorithme Apriori
(extraction de règles d'association)

{ IL EST BASÉ SUR LE PRINCIPE D'APRIORI QUI STIPULE QUE TOUT SOUS-ENSEMBLE D'UN ENSEMBLE FRÉQUENT EST ÉGALEMENT UN ENSEMBLE FRÉQUENT. }

La formulation Étant donné un ensemble de n éléments, noté n -itemset, et en utilisant le principe d'Apriori, nous savons que ce n -itemset est la réunion de $(n-1)$ -itemsets.

- Cela nous permet de déterminer les ensembles fréquents de dimension n en examinant uniquement les ensembles fréquents de dimension $(n-1)$.
- Si $\{a,b,c,d\}$ est un ensemble fréquent, ses 4 sous-ensembles à 3 valeurs sont également fréquents.

Discrétisation des données

ALGORITHMES

Algorithme Apriori
(extraction de règles d'association)

Exemple : L'Algorithme Apriori peut être utilisé pour l'analyse du panier de la ménagère. La discrétisation des montants d'achat peut aider à trouver des associations entre les articles de manière plus efficace.

Scénario : Magasin d'alimentation en ligne

Ensemble de données : Nous disposons de données de transaction provenant d'une boutique d'alimentation en ligne. Chaque transaction représente l'achat d'un client et contient la liste des articles achetés lors de cette transaction.

Discrétisation des données

ALGORITHMES

Algorithme Apriori
(extraction de règles d'association)

Objectif : Identifier les associations d'articles pour améliorer les recommandations de produits et les dispositions des magasins.

Exemples de transactions :

- 1) Transaction 1 : Pain, lait, œufs
- 2) Transaction 2 : Pain, fromage, bière, lait
- 3) Transaction 3 : lait, fromage, bière, chips
- 4) Transaction 4 : Pain, lait, fromage, bière
- 5) Transaction 5 : Pain, œufs, lait, chips

T	Panier
1	p,l,o
2	p,f,b,l
3	l,f,b,c
4	p,l,f,b
5	p,o,l,c

Discrétisation des données

ALGORITHMES

Algorithme Apriori
(extraction de règles d'association)

Application de l'algorithme Apriori

Étape 1 - Identifier les éléments fréquents : Comptez la fréquence de chaque élément (par exemple, le pain, le lait, le fromage, la bière, les œufs, les chips) dans l'ensemble de données.

Étape 2 - Génération d'ensembles d'éléments candidats : Créer des ensembles candidats initiaux de taille 2 (paires d'éléments) en utilisant les éléments fréquents de l'étape 1.

Étape 3 - Élimination (pruning) des ensembles peu fréquents : Calculer le support (fréquence) des itemsets candidats. Éliminez les items dont le support est inférieur à un seuil prédéfini (par exemple, 2).

Discrétisation des données

ALGORITHMES

Algorithme Apriori
(extraction de règles d'association)

Le Support: la probabilité de base qu'un événement se produise. Il est mesuré par la proportion de transactions dans lesquelles un ensemble d'éléments apparaît.

En d'autres termes, le support (A) est le nombre de transactions qui incluent A divisé par le nombre total de transactions.

$$S(A) = \frac{T(A)}{\sum_i^n T(n)}$$

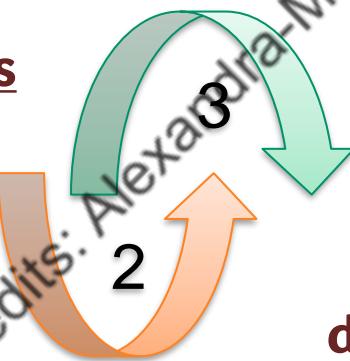
Où $T(A)$ - est le nombre de transactions qui incluent A;
 $\sum_i^n T(n)$ - le nombre total de transactions.

Discrétisation des données

ALGORITHMES

Algorithme Apriori
(extraction de règles d'association)

Répétez les étapes



Continuez à générer des ensembles d'éléments candidats plus importants en joignant des ensembles d'éléments fréquents ($n-1$).

Éliminez les ensembles peu fréquents jusqu'à ce qu'il ne soit plus possible de générer d'ensembles fréquents.

Discrétisation des données

ALGORITHMES

Algorithme Apriori
(extraction de règles d'association)

Résultats :

- L'algorithme Apriori identifie des ensembles d'articles fréquents, tels que {Pain, Lait}, indiquant les articles souvent achetés ensemble.
- Ces associations peuvent être utilisées pour améliorer les recommandations de produits ou l'agencement des magasins afin d'augmenter les ventes.
- L'algorithme aide les entreprises à comprendre le comportement des clients et à prendre des décisions fondées sur des données pour améliorer leurs stratégies.

Bibliographie

- [1] <https://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf>
- [2] <https://towardsdatascience.com/building-a-random-forest-classifier-c73a4cae6781>
- [3] https://miro.medium.com/v2/resize:fit:1100/format:webp/0*TtRjZOvYr7uSuWQn.jpg
- [4] Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.