

IFT870/BIN710

Forage de données

Thème 3 : Prétraitement et représentation des données

Aida Ouangraoua

Département d'informatique



Université de
Sherbrooke

Partie I : Théorie

Prétraitement

☐ **Auscultation (qualité des données)**

☐ **Nettoyage**

☐ **Intégration**

☐ **Réduction**

☐ **Transformation**

☐ **Discrétisation/binarisation**

Auscultation

❑ **Auscultation (qualité des données)**

- ❖ Correctitude
- ❖ Complétude
- ❖ Cohérence
- ❖ À jour
- ❖ Vraisemblable
- ❖ Interprétable

Prétraitement

☐ Auscultation (qualité des données)

☐ Nettoyage

☐ Intégration

☐ Réduction

☐ Transformation

☐ Discrétisation/binarisation

Nettoyage

□ Nettoyage : exemple de cas

- ❖ **Données incomplètes** : manque de valeurs d'attribut, ou manque de certains attributs intéressants

Exemple : Année =

- ❖ **Données bruitées**: contenant du bruit, des erreurs ou des valeurs aberrantes.

Exemple : Prix = -1000

- ❖ **Données incohérentes**: contenant des incohérences entre valeurs d'attributs.

Exemple : Age = 40 et DateNaissance = "01/01/1900";
Valeurs «1, 2, 3,A,B,C» pour attribut catégoriel;
Même valeurs pour plusieurs objets.

Nettoyage

- ❑ **Nettoyage : dans le cas de données incomplètes**
(Module `sklearn.impute : model.fit() ... model.transform()`)
 - ❖ Retirer les objets incomplets : à condition qu'il reste assez de données
 - ❖ Completer les valeurs manquantes
 - Manuellement, si on sait où les retrouver
 - Automatiquement avec :
 - La moyenne globale
 - La moyenne suivant la classe
 - La valeur la plus probable (inférence bayésienne ou arbre de décision)

Nettoyage

❑ **Nettoyage : dans le cas de données bruitées**

(Lissage : nettoyer le bruit pour réduire les irrégularités du modèle)

- ❖ Partitionner les données en parts de fréquence égale puis nettoyer par moyenne, médiane ou bornes
- ❖ Nettoyer après avoir appliqué un modèle de régression
- ❖ Détecter par clustering et nettoyer les valeurs aberrantes (extrêmes)
- ❖ Détecter et nettoyer manuellement les valeurs suspectes

Nettoyage

❑ Nettoyage : dans le cas de données incohérentes

❖ Utiliser des métadonnées

Exemple: ensemble ou intervalle des valeurs, distribution, relations

❖ Vérifier les règles d'unicité, de consécuité ou toute autre règle connue

❖ Explorer les données pour découvrir des règles et des relations, et détecter les valeurs suspectes

Exemple: corrélation et regroupement pour trouver les valeurs aberrantes

Prétraitement

- ☐ Auscultation (qualité des données)
- ☐ Nettoyage
- ☐ **Intégration**
- ☐ Réduction
- ☐ Transformation
- ☐ Discrétisation/binarisation

Intégration

❑ **Intégration : Combinaison de données de sources multiples**

- ❖ Identifier les objets ou attributs redondants

Exemple : analyse de corrélation/co-variance

- ❖ Résoudre les conflits de valeurs d'attributs

Exemple : différentes échelles, unités de mesure

Prétraitement

- ☐ Auscultation (qualité des données)
- ☐ Nettoyage
- ☐ Intégration
- ☐ Réduction
- ☐ Transformation
- ☐ Discrétisation/binarisation

Réduction

❑ **Réduction : réduire le volume des données pour accélérer/améliorer le traitement**

❑ **Deux stratégies**

❖ Réduction de dimension (attributs)

- Élimination des attributs non pertinents et réduction du bruit
- Réduction du temps et de l'espace requis pour l'analyse
- Facilitation de la visualisation

❖ Réduction des données (objets)

- Choisir une représentation compressée des données

Réduction de dimension

- ❑ **Analyse en composantes principales (PCA)**
- ❑ **Factorisation par matrices non-négatives (NMF)**
- ❑ **Manifold**
- ❑ **Sélection d'attributs** : ne conserver que les attributs les plus pertinents

Réduction de dimension par sélection d'attributs

(Module `sklearn.feature_selection : model.fit_transform()`)

- ❑ **Statistique univariée** : attributs classés en fonction de la significativité statistique de leur relation avec l'attribut cible
Exemple pour la classification : analyse de la variance ANOVA
 - ❖ Attribut considéré individuellement (**SelectKBest**, **SelectPercentile**)
- ❑ **Sélection par modèle** : Utiliser des modèles supervisés qui évaluent l'importance des attributs (*feature_importances*)(Exemple: arbres de décision et dérivés, modèles linéaires)
(**SelectFromModel**)
- ❑ **Sélection itérative par modèle** : utilisation de plusieurs modèles de façon itérative, en affinant (ajout/retrait) au fur et à mesure l'ensemble des attributs jusqu'à un critère d'arrêt
(RFE 'RecursiveFeatureElimination')

Réduction des données

- ❑ **Réduction par modèle (supervisée) : apprendre un modèle à partir de données, conserver les paramètres du modèle, et supprimer les données**

Exemple : régression linéaire, régression multiple

- ❑ **Réduction sans modèle (non-supervisé) :**

- ❖ **Avec histogramme** : ne conserver que les moyennes ou sommes des valeurs dans chaque intervalle

- ❖ **Avec clustering** : ne conserver que la représentation des clusters

Exemple : centre et rayon, ou dendrogramme pour clustering hiérarchique

- ❖ **Avec échantillonnage** : ne conserver qu'un échantillon des objets

- ❑ **Compression des données sans perte (séquence, audio, vidéo)**

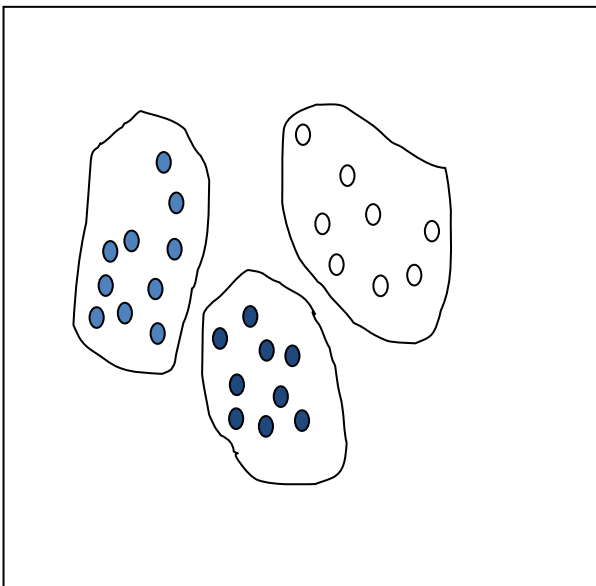
Type d'échantillonnage

☐ **Aléatoire**

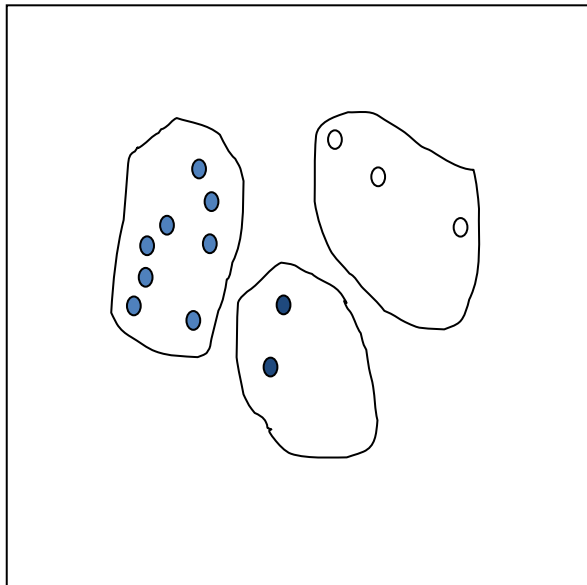
☐ **Avec remise** : un objet peut être tiré plusieurs fois

☐ **Sans remise** : un objet est tiré une seule fois

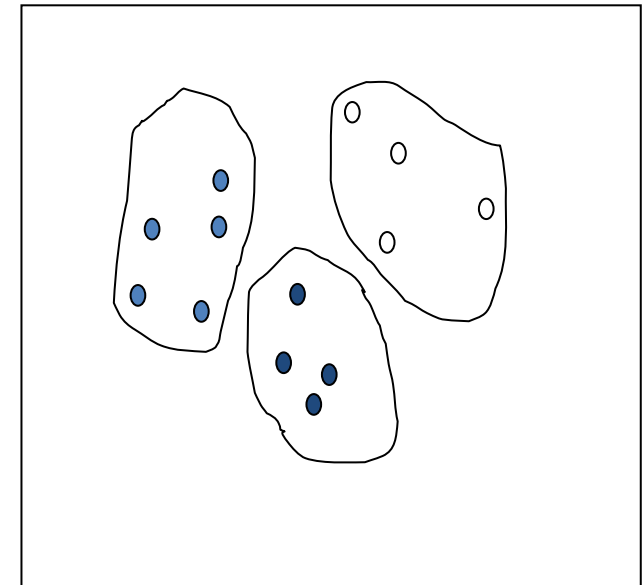
☐ **Stratifié** : partitionner les données et échantillonner dans chaque partie.



Données



Échantillonnage aléatoire



Échantillonnage stratifié

Prétraitement

- ☐ Auscultation (qualité des données)
- ☐ Nettoyage
- ☐ Intégration
- ☐ Réduction
- ☐ Transformation
- ☐ Discrétisation/binarisation

Transformation des données

(Module sklearn.preprocessing)

❑ **Transformation** : appliquer une fonction qui transforme l'ensemble des valeurs d'un attribut en un nouvel ensemble de valeurs

❑ **Fonctions:**

- ❖ **Transformation non linéaire:** appliquer des fonctions mathématiques telles que:
 - **log, exp** : aident à ajuster les échelles relatives
 - **sin, cos** : utiles pour les données à motifs périodiques
- ❖ **Normalisation:** ramener les valeurs d'un attribut à un intervalle spécifié
- ❖ **Construction d'attributs additionnels** : à partir des attributs existants (Exemple : polynôme)

Normalisation des données

(Module `sklearn.preprocessing : model.fit_transform()`)

❑ **Normalisation** : ramener les valeurs d'un attribut à un intervalle spécifié

❖ **Normalisation Min-max** : vers $[new_min_A, new_max_A]$
(**MinMaxScaler**)

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

Exemple: Pour des valeurs variant de 500 à 3000 normalisées vers l'intervalle $[0, 1]$, la valeur 650 devient: $\frac{650 - 500}{3000 - 500} (1.0 - 0) + 0 = 0.06$

❖ **Normalisation par Z-score** (μ : moyenne, σ : écart-type):
(**StandardScaler**)

$$v' = \frac{v - \mu}{\sigma}$$

Exemple: Si $\mu = 2000$ et $\sigma = 700$, la valeur 650 devient:

$$\frac{650 - 2000}{700} = -1.93$$

Construction d'attributs additionnels

(Module `sklearn.preprocessing`)

- ❑ Ajout de produits d'attributs ou attributs polynomiaux pour enrichir la représentation.
- ❖ Ajout de produits d'attributs : utile pour représenter des interactions entre les attributs
(`np.hstack([X, Y, X*Y])`)
- ❖ Ajout d'attributs polynomiaux : utile pour les modèles linéaires de régression → régression polynomiale
(`PolynomialFeatures`)

Prétraitement

- ☐ Auscultation (qualité des données)
- ☐ Nettoyage
- ☐ Intégration
- ☐ Réduction
- ☐ Transformation
- ☐ Discrétisation/binarisation

Discrétisation/binarisation (Module `sklearn.preprocessing`)

- ❑ Discrétisation: attribut à valeurs continues → attribut à k valeurs catégorielles (**KBinsDiscretizer**)
 - ❖ Diviser l'ensemble des valeurs continues en k intervalles, puis remplacer les valeurs par les identifiants des intervalles
- ❑ Binarisation: attribut à k valeurs catégorielles → k attributs binaires (**OneHotEncoder**, ou **pandas.getdummies**)
 - ❖ Supprimer l'attribut initial et remplacer par k attributs binaires, un pour chaque valeur initiale

Discretisation/binarisation (Module sklearn.preprocessing)

❑ pandas.getdummies()

Integer Feature	Categorical Feature
0	0 socks
1	1 fox
2	2 socks
3	1 box

```
display(pd.get_dummies(demo_df))
```

Integer Feature	Categorical Feature_box	Categorical Feature_fox	Categorical Feature_socks
0	0	0	1
1	1	1	0
2	2	0	1
3	1	0	0

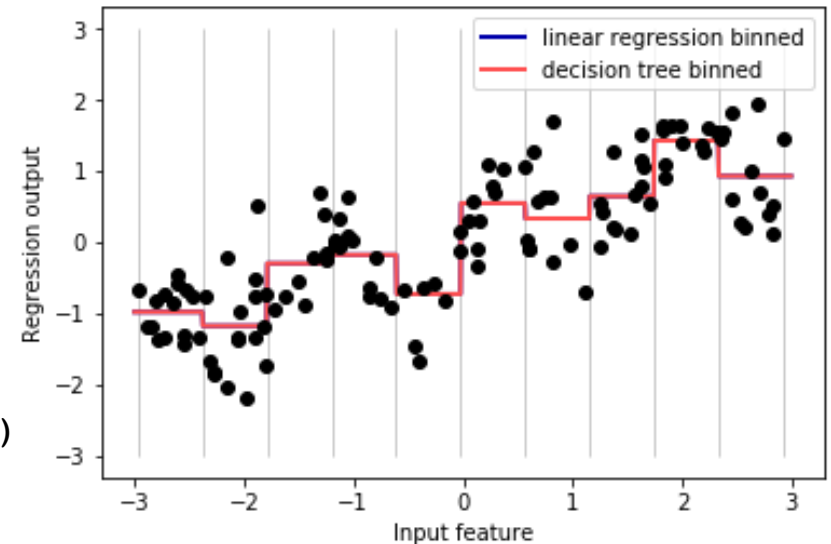
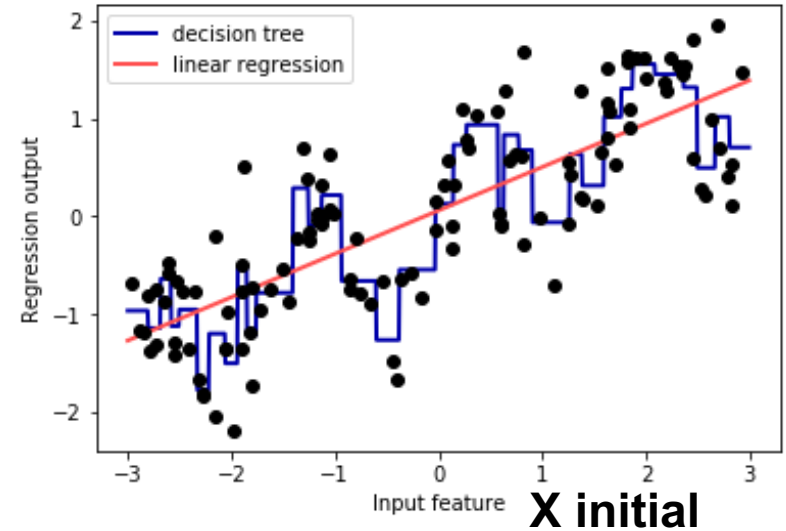
❑ OneHotEncoder

	Integer Feature_0	Integer Feature_1	Integer Feature_2	Categorical Feature_box	Categorical Feature_fox	Categorical Feature_socks
0	1	0	0	0	0	1
1	0	1	0	0	1	0
2	0	0	1	0	0	1
3	0	1	0	1	0	0

Discretisation/binarisation (Module sklearn.preprocessing)

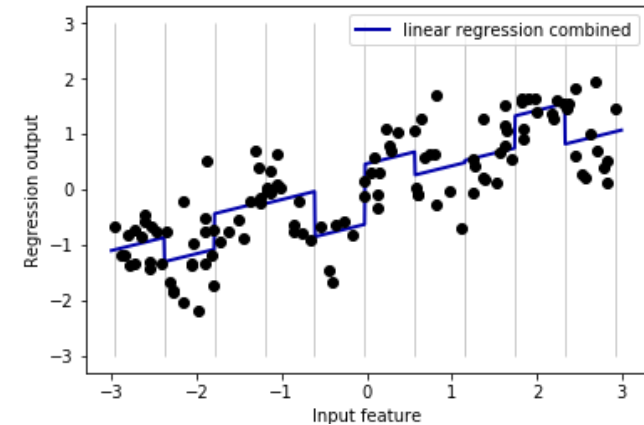
```
[[-0.753]  
 [ 2.704]  
 [ 1.392]  
 [ 0.592]  
 [-2.064]  
 [-2.064]  
 [-2.651]  
 [ 2.197]  
 [ 0.607]  
 [ 1.248]]
```

```
array([[0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],  
      [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.],  
      [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],  
      [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],  
      [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],  
      [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],  
      [1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],  
      [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],  
      [0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],  
      [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.]])
```

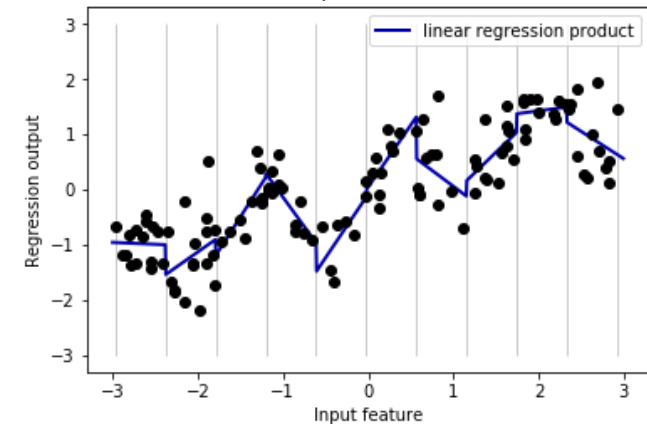


Discretisation/binarisation (Module sklearn.preprocessing)

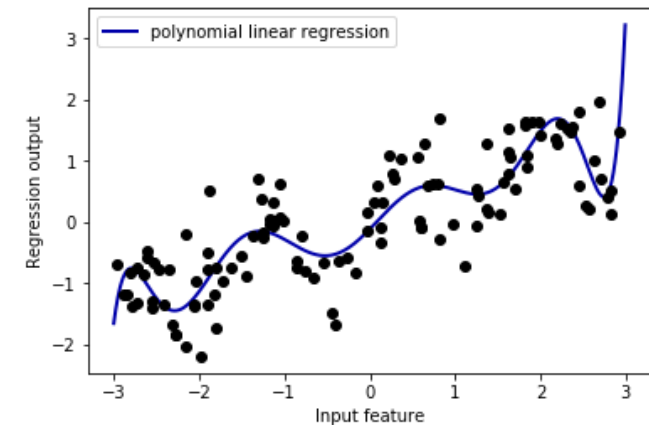
X initial + X discrétisé et binarisé



X initial * X discrétisé et binarisé



Polynômes de X



Discretisation/binarisation (Module `sklearn.preprocessing`)

❑ **ColumnTransformer, make_columntransformer**

Pour définir les transformations à appliquer à différents ensembles d'attributs

Références

- [1] PEDREGOSA et al. : *Scikit-learn : Machine Learning in Python*. JMLR 12, pp. 2825-2830. (User guide and API : [https ://scikit-learn.org/stable/](https://scikit-learn.org/stable/)), 2011.

- [2] Jiawei HAN, Micheline KAMBER, Jian PEI. *DataMining: Concepts and Techniques (Third edition)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

- [3] Adreas C. MÜLLER et Sarah GUIDO : *Introduction to Machine Learning with Python*. O'Reilly Media, Inc., Sebastopol, CA, 2017.

- [4] Pang-Ning TAN, Michael STEINBACH et Vipin KUMAR : *Introduction to Data Mining, (Second Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2018.