université PARIS-SACLAY

MACHINE-LEARNING (3)
CHAINE DE TRAITEMENTS

Vincent Guigue
vincent.guigue@agroparistech.fr

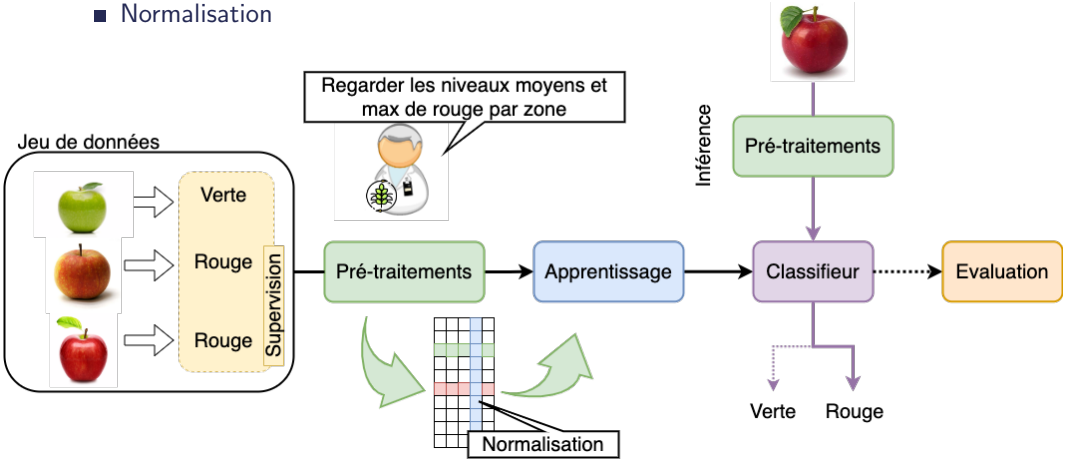
MIA PARIS-SACLAY
EKINOCS

AGROPARISTECH
Institut des Sciences et Industries du Vivant et de l'Environnement

Introduction

Chaine de traitements

- Intégration des connaissances expert/métier
  - Construction de caractéristiques
  - Sélection de caractéristiques
  - Choix des métriques
- Stabilisation numérique
  - Normalisation



Intégration de nos outils dans scikit-learn

Objectifs

- Comprendre les principaux leviers de performances en machine learning (encore !)
- Savoir les mettre en œuvre [dans scikit-learn]

Valeurs manquantes

Comment gérer les valeurs manquantes (continues) ?

	mpg	cylinders	displacement	horsepower	weight	acceleration
30	28.0	4	140.0	90	2264	15.5
31	25.0	4	113.0	95	2228	14.0
32	25.0	4	98.0	?	2046	19.0
33	19.0	6	232.0	100	2634	13.0
34	16.0	6	225.0	105	3439	15.5
35	17.0	6	250.0	100	3329	15.5
36	19.0	6	250.0	88	3302	15.5
37	18.0	6	232.0	100	3288	15.5
38	14.0	8	350.0	165	4209	12.0
39	14.0	8	400.0	175	4464	11.5
40	14.0	8	351.0	153	4154	13.5

Souvent localiser sur une (ou quelques) colonne(s)

- EM : estimation des données manquantes
- Suppression des lignes affectées
- Affectation d'une valeur arbitraire
  - Moyenne / Médiane
  - Plus proche voisin

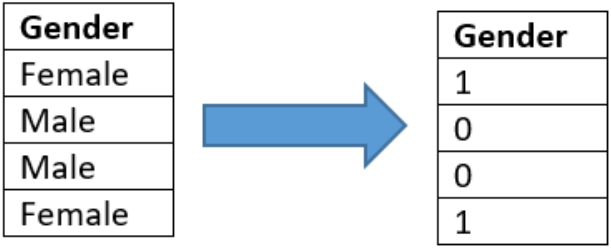
Comment gérer les valeurs manquantes (Discrètes) ?

- Valeur la plus fréquente
- Echantillonnage (multinomial)
- Plus proche voisin (sur les autres caractérstiques)

Feature Engineering

Variables discrètes

- Cas binaire / Cas n-aire

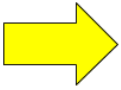


- Option intéressante : grouper les catégories peu fréquentes
  `OneHotEncoder(min_frequency=6, sparse=False)`
- Pour aller plus loin
  - ECoC
  - Embedding

Variables discrètes

- Cas binaire / Cas n-aire

Color
Red
Red
Yellow
Green
Yellow



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

- Option intéressante : grouper les catégories peu fréquentes
  `OneHotEncoder(min_frequency=6, sparse=False)`
- Pour aller plus loin
  - ECoC
  - Embedding

Ouverture vers le deep learning

C'est quoi le **deep learning** ? réponse assez ouverte...

- Apprendre des représentations (et des distances) entre éléments discrets
  - e.g. Distance sémantique et/ou grammaticale entre les mots
  - Distance entre les profils utilisateurs dans les systèmes de recommandation
  - Distance entre des graphes / des noeuds d'un graphe
- Apprendre des représentations d'objets complexes
  - e.g. en vision, projeter les données dans un espace de faible dimension sémantique
- Des architectures génératives
  - GPT, Dall-e, ...
  - Transférables d'une application à l'autre
- Architectures complexes : différents objectifs/modalités de données
  - Modélisation directe des contraintes métiers

Ouverture vers le deep learning

C'est quoi le **deep learning** ? réponse assez ouverte...

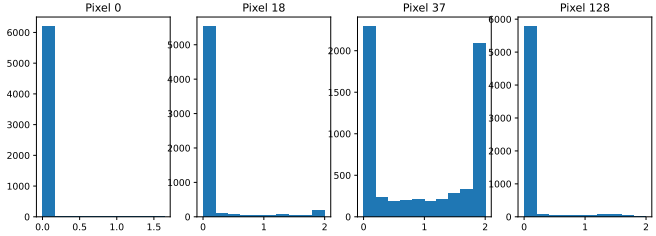
- Apprendre des représentations (et des distances) entre éléments discrets
  - e.g. Distance sémantique et/ou grammaticale entre les mots
  - Distance entre les profils utilisateurs dans les systèmes de recommandation
  - Distance entre des graphes / des noeuds d'un graphe
- Apprendre des représentations d'objets complexes
  - e.g. en vision, projeter les données dans un espace de faible dimension sémantique
- Des architectures génératives
  - GPT, Dall-e, ...
  - Transférables d'une application à l'autre
- Architectures complexes : différents objectifs/modalités de données
  - Modélisation directe des contraintes métiers

$\Rightarrow$ Apprendre la meilleure représentation
Conférence deep-learning : ICLR (Int. Conf. on Representation Learning)

Simplification des données

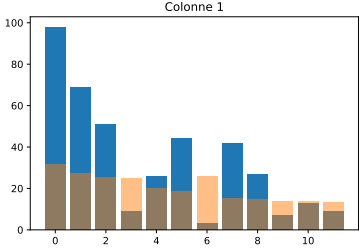
- Binarisation

- e.g. pixel dans une image usps



- Discrétisation (quantiles ou linéaires)

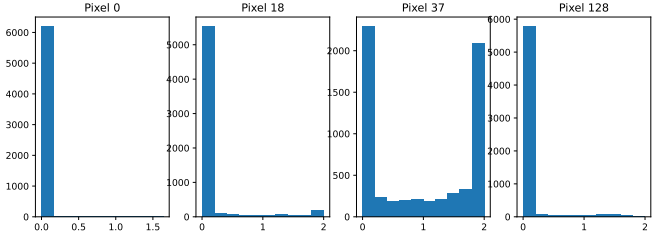
- e.g. CV (auto-mpg)



Simplification des données

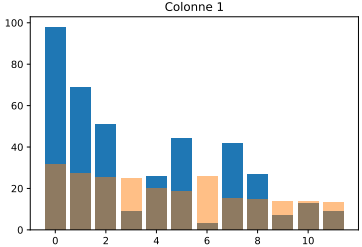
- Binarisation

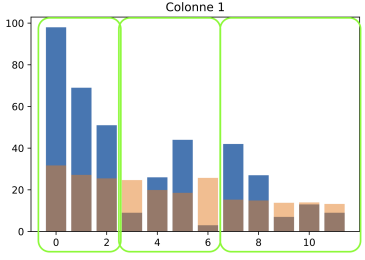
- e.g. pixel dans une image usps



- Discrétisation (quantiles ou linéaires)

- e.g. CV (auto-mpg)



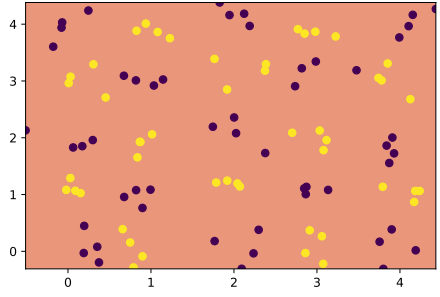


Transformations arbitraires / métiers

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
5	15.0	8	429.0	198	4341	10.0	70	1	ford galaxie 500
6	14.0	8	454.0	220	4354	9.0	70	1	chevrolet impala
7	14.0	8	440.0	215	4312	8.5	70	1	plymouth fury iii
8	14.0	8	455.0	225	4425	10.0	70	1	pontiac catalina
9	15.0	8	390.0	190	3850	8.5	70	1	amc ambassador dpl
10	15.0	8	383.0	170	3563	10.0	70	1	dodge challenger se
11	14.0	8	340.0	160	3609	8.0	70	1	plymouth 'cuda 340
12	15.0	8	400.0	150	3761	9.5	70	1	chevrolet monte carlo
13	14.0	8	455.0	225	3086	10.0	70	1	buick estate wagon (sw)
14	24.0	4	113.0	95	2372	15.0	70	3	toyota corona mark ii
15	22.0	6	198.0	95	2833	15.5	70	1	plymouth duster
16	18.0	6	199.0	97	2774	15.5	70	1	amc hornet
17	21.0	6	200.0	85	2587	16.0	70	1	ford maverick
18	27.0	4	97.0	88	2130	14.5	70	3	datsun pl510
19	26.0	4	97.0	46	1835	20.5	70	2	volkswagen 1131 deluxe sedan
20	25.0	4	110.0	87	2672	17.5	70	2	peugeot 504
21	24.0	4	107.0	90	2430	14.5	70	2	audi 100 ls

Comment gérer la dernière colonne ?

Exemple checkers

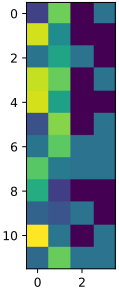


Avec un classifieur linéaire...

- Classification linéaire par défaut :

  $ScoresCV$: $[0.5, 0.5, 0.3, 0.5, 0.4]$

- Ajout de colonnes :
  intervalles binaires (pair/impair) sur les deux dimensions



$ScoresCV$: $[0.75, 0.85, 0.9, 0.8, 0.85$

Normalisation

Pourquoi normaliser ?

Par ordre d'usage :

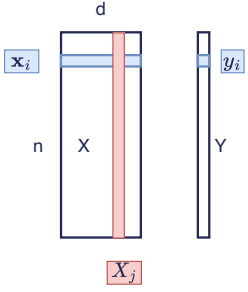
1. Pour améliorer les performances
   - Tirer parti d'informations à différentes échelles
2. Pour faciliter le réglage des hyper-paramètres
   - Mêmes ordres de grandeur $\Rightarrow$ mêmes réglages
3. Pour respecter les propriétés du modèle utilisé
   - e.g. hypothèse multinomiale
   - modèles sans biais (normalisation des y)

Normalisation gaussienne

Centrer réduire chaque variable :

$$Z_j = \frac{X_j - \mu_j}{\sigma_j}$$

- = Normalisation standard (dans scikit-learn)
- Un test à faire systématiquement (la fonction est implémentée dans tous les systèmes)



La normalisation par individus

ATTENTION :

- Lié à la nature des données

- min-max

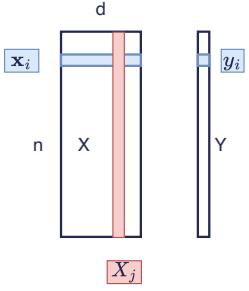
$$\tilde{\mathbf{x}_i} = \frac{\mathbf{x}_i - \min(\mathbf{x}_i)}{\max(\mathbf{x}_i) - \min(\mathbf{x}_i)} \in [0, 1]$$

- multinomiale

$$\tilde{\mathbf{x}_i} = \frac{\mathbf{x}_i}{\sum_j x_{ij}} \in [0, 1], \qquad \sum_j \tilde{x_{ij}} = 1$$

- Normalisation des produits scalaires (matrices de Gram)

$$\tilde{\mathbf{x}_i} = \frac{\mathbf{x}_i}{\sum_j x_{ij}^2}, \qquad \|\mathbf{x}_i\|^2 = 1$$



Conclusion

Exemple de la classification de signaux

Que veut certaines propriétés :

- Détection de motifs indépendamment de l'échelle
  - Normalisation par individu (min-max)
  - Somme à 1 (signaux positifs), somme des carrés à 1, somme à 0, ...
- Invariance en translation (ou pas, selon la nature des informations discriminantes)
  - Calcul des moyennes, écart-types
  - FFT, PSD

Conclusion : approche standard en ML

1. Traitements = interprétation des informations métier/expert
2. Batteries de tests classiques pour estimer les performances attendues
   - normalisation standard
   - modèles linéaires + forêts
   - validation croisée ou autre selon les cas
3. Optimisation (idéalement automatisée ou semi-automatisée)
   - Feature engineering
   - Grid-search
   - Ensembling

Beaucoup d'outils existent en sklearn $\Rightarrow$ Maitrise + documentation
Lorsqu'il manque un outil $\Rightarrow$ Penser intégration