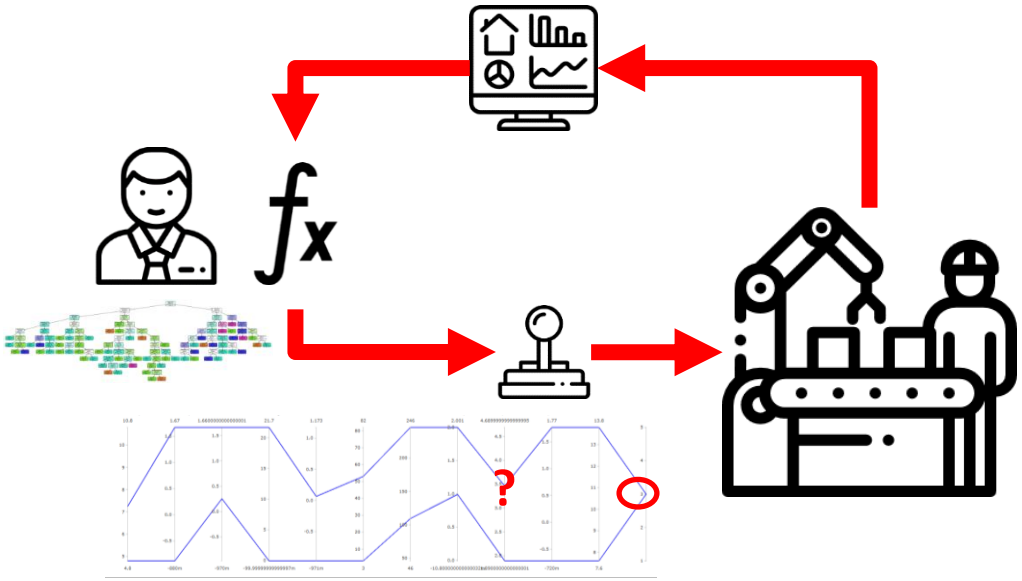
Comment améliorer votre production via la fouille de données ?

Data Science

Initiation au prétraitement de données



Wahb ZOUHRI
Jean-Yves DANTAN
Lazhar HOMRI
Alain ETIENNE

Arts et Métiers Institute of Technology

LCFC Conception Fabrication Commande

Data preprocessing
Introduction

Introduction

Le prétraitement est l'ensemble des manipulations qui transforment un jeu de données brut pour le rendre utilisable par un modèle d'**apprentissage automatique** (*machine learning*).

Le prétraitement de données est nécessaire :

- pour rendre nos données appropriées à certains modèles d'apprentissage automatique,
- pour réduire la dimensionnalité,
- pour mieux identifier les données pertinentes,
- et pour augmenter les performances du modèle.

Il s'agit de la partie la plus importante d'un projet d'apprentissage automatique et elle est fortement susceptible d'affecter le succès d'un projet.

En effet, si nous n'alimentons pas un modèle d'apprentissage automatique avec des données correctement formées, **il ne fonctionnera pas du tout**.

Introduction

Le prétraitement est l'ensemble des manipulations qui transforment un jeu de données brut pour le rendre utilisable par un modèle d'**apprentissage automatique** (*machine learning*).

- **Nettoyage**
- **Codage**
- **Transformation**
- **Mise à l'échelle**
- **Réduction de la dimensionnalité**
- **Sur-échantillonnage**

Data preprocessing

Nettoyage de données *(data cleaning)*

Introduction

Le nettoyage des données est le processus de **détection** et de **correction** (ou de suppression) des données corrompus ou inexacts d'un ensemble de données.

Le nettoyage des données est nécessaire pour remplacer les valeurs manquantes dans un jeu de données, car presque tous les modèles d'apprentissage automatique ne sont pas en mesure de les traiter.

A	B	C
12	-40	-
20	-	a
-	69	b
33	-15	b

Nettoyage

A	B	C
12	-40	**b**
20	**2**	a
21,5	69	b
33	-15	b

Cependant, il faut savoir que la façon de traiter les valeurs manquantes dépend du type de variable que l'on traite, c'est-à-dire s'il s'agit d'une variable **quantitative** ou **qualitative**.

Introduction

Variable quantitative : contient des nombres (souvent des nombres à virgule flottante).

Variable qualitative : contient des valeurs dans un ensemble discret et de taille finie (par exemple, des lettres, des couleurs).

A	B	C
12	-40	b
20	2	a
21,5	69	b
33	-15	b

Quantitative **Qualitative**

La première chose à faire est d'identifier (à l'aide de Python) les variables quantitatives et les variables qualitatives dans un jeu de données.

Identification des types de variables

Importer la librairie pandas dédiée à la manipulation et l'analyse des matrices de données.

- `import pandas as pd`

Importer le fichier de données Excel. Il suffit de remplacer ''sample_dataset'' par le chemin vers votre fichier Excel.

- `data = pd.read_excel("sample_dataset.xlsx")`

Retourner le type de données de chaque paramètre.

- `data.dtypes`

Séparer les données quantitatives des données qualitatives.

- `categorical_variables = data.select_dtypes(include =['object','category','bool'])`
- `numerical_variables = data.select_dtypes(exclude =['object','category','bool'])`

Nettoyage des variables quantitatives

- **Imputation à l'aide de valeurs (moyennes/médianes) :**

  Cela fonctionne en calculant la moyenne/médiane des valeurs non manquantes dans une colonne, puis en remplaçant les valeurs manquantes dans chaque colonne séparément et indépendamment des autres.

  Avantages :

  ✓ Facile et rapide.
  ✓ Fonctionne bien avec les petits jeux de données.

  Inconvénients :

  × Ne prend pas en compte les corrélations entre les paramètres.
  × Pas très précis.

Nettoyage des variables qualitatives

- **Imputation en utilisant la valeur la plus fréquente:**

  Une autre stratégie statistique consiste à remplacer les données manquantes par les valeurs les plus fréquentes dans chaque colonne.

  Avantages :

  ✓ Fonctionne bien avec les variables qualitatives.

  Inconvénients :

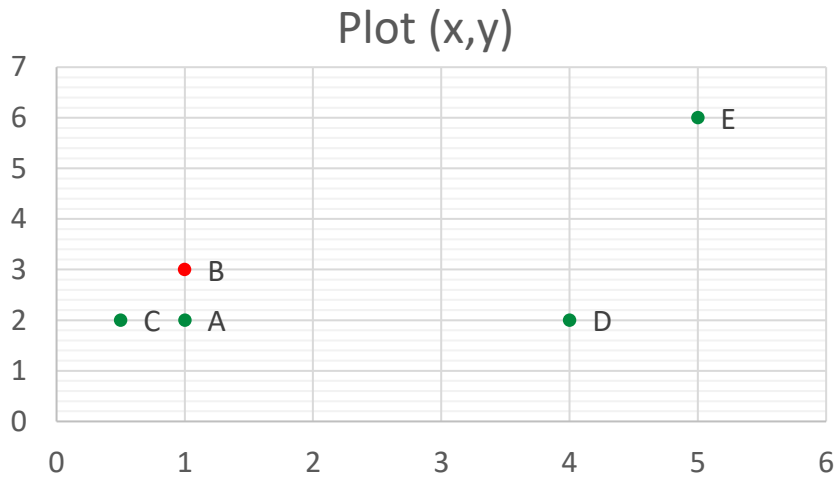
  × Ne prend pas en compte les corrélations entre les paramètres.
  × Elle peut introduire un biais dans les données.

Nettoyage des variables quantitatives & qualitatives

- **Imputation à l'aide de la méthode des K-plus proches voisins (KNN):**

  Cet algorithme est basé sur la méthode **KNN** pour remplacer la valeur manquante d'une donnée par une valeur moyenne calculée à partir de ses K plus proches voisins.

	x	y	z
A	1	2	3
B	1	3	NaN
C	0,5	2	8
D	4	2	30
E	5	6	60



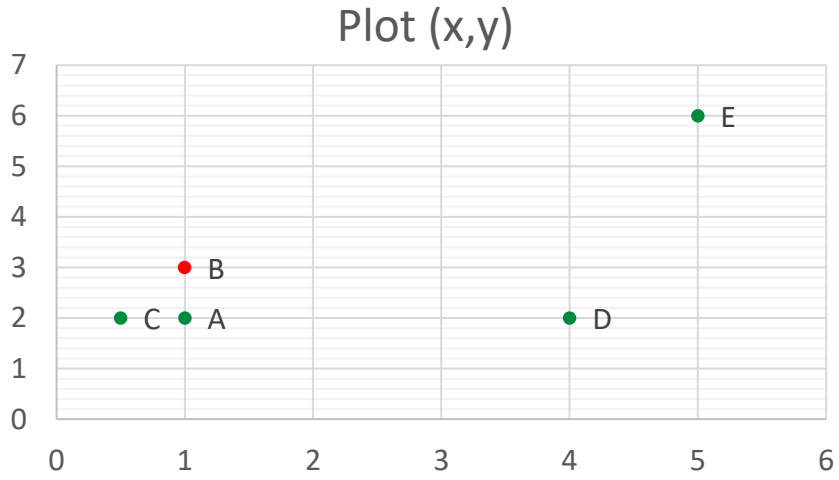
Pour un nombre de voisins **K = 2** :

- Nous allons tout d'abord identifier les deux plus proches voisins, qui sont dans ce cas les points A et C.
- Nous allons utiliser ces deux points pour estimer la valeur manquante de B, de telle sorte que : NaN est remplacé par $\frac{3+8}{2} = 5,5$

Nettoyage des variables quantitatives & qualitatives

- **Imputation à l'aide de la méthode des K-plus proches voisins (KNN):**

	x	y	z
A	1	2	3
B	1	3	NaN
C	0,5	2	8
D	4	2	30
E	5	6	60



Pour un nombre de voisins **K = 3** :

- Les trois plus proches voisins sont : A, C et D.
- NaN est remplacé dans ce cas par $\frac{3+8+30}{3} = 13,66$.

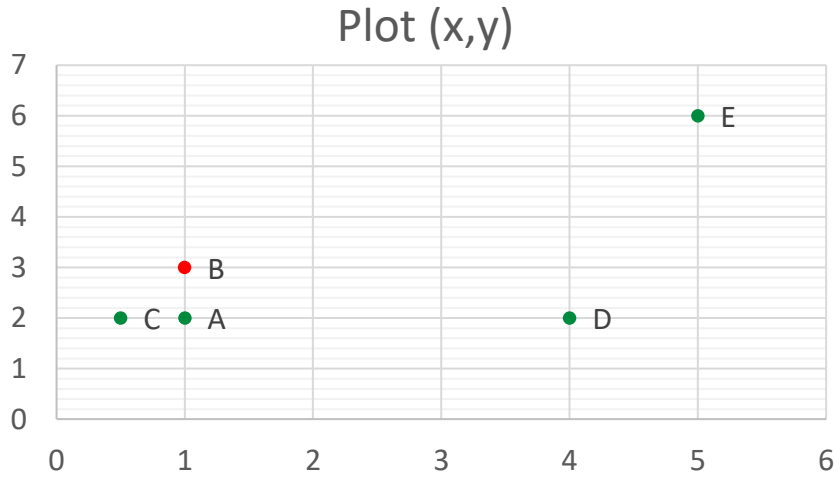
Dans cet exemple, nous pouvons remarquer que le point D est loin du point B, ce qui peut biaiser les résultats.

Pour y remédier, il est possible d'ajouter un poids à chaque valeur de sorte que plus le voisin est proche, plus le poids est élevé.

Nettoyage des variables quantitatives & qualitatives

- **Imputation à l'aide de la méthode des K-plus proches voisins (KNN):**

	x	y	z
A	1	2	3
B	1	3	NaN
C	0,5	2	8
D	4	2	30
E	5	6	60



Pour un nombre de voisins **K = 3** :

Il est possible d'ajouter un poids à chaque valeur de sorte que plus le voisin est proche, plus le poids est élevé.

Les poids sont définis par l'inverse de la distance (euclidienne) entre le point B et ses voisins :

$$z_B = \frac{\sum_{i=1}^{K} \omega_i * z_i}{\sum_{i=1}^{K} \omega_i}$$

Avec :

$$\omega_i = \frac{1}{dist(B, i^{ème}\_voisin)}$$

Nettoyage des variables quantitatives & qualitatives

- **Imputation à l'aide de la méthode des K-plus proches voisins (KNN):**

  Avantages :

  ✓ Peut être beaucoup plus précise que les méthodes d'imputation par la moyenne, la médiane ou la valeur la plus fréquente (cela dépend du jeu de données).

  Inconvénients :

  × Peut être coûteuse en termes de calcul. KNN fonctionne en stockant l'ensemble des données en mémoire.

  × K-NN est assez sensible aux valeurs aberrantes dans les données.

Imputation des valeurs manquantes

Importer les méthodes d'imputation nécessaires.

- `from sklearn.impute import SimpleImputer, KNNImputer`

Remplacer les valeurs manquantes par la valeur moyenne, médiane ou la valeur la plus fréquente.

- ```
  cleaner = SimpleImputer(strategy='mean') # mean peut être modifié par median ou most_frequent.
  New_X = cleaner.fit_transform(X)  # avec X une colonne du jeu de données.
  ```

Remplacer les valeurs manquantes par la méthode KNN.

- ```
  cleaner = KNNImputer(n_neighbors = 10, weights='uniform')  # uniform ou distance.
  New_X = cleaner.fit_transform(X)  # avec X une colonne du jeu de données.
  ```

Imputation des valeurs manquantes

Code Python : Traiter les deux types de variables en même temps.

```python
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer, make_column_selector

data = pd.read_csv("sample_dataset.csv")                    # Importer le jeu de données
numerical = data.select_dtypes(exclude = "object").columns  # Identifier les noms des variables quantitatives
categorical = data.select_dtypes(include = "object").columns  # Identifier les noms des variables qualitatives

'''
La fonction ci-dessous permet de traiter des variables qualitatives et quantitatives en même temps.
Il suffit de spécifier des 3-uplets définis par :
    - un nom,
    - une méthode d'imputation,
    - et les noms des variables concernées par cette imputation.
'''

cleaner = ColumnTransformer([
    ('numerical_transformer', SimpleImputer(strategy='mean'), numerical),
    ('categorical_transformer', SimpleImputer(strategy='most_frequent'), categorical)])

new_data = cleaner.fit_transform(data)

'''
Une autre façon de procéder :
    en utilisant la fonction make_column_selector qui spécifie le type de variables concernées par l'imputation.
'''

cleaner = ColumnTransformer([
    ('numerical_transformer', SimpleImputer(strategy='mean'), make_column_selector(dtype_exclude="object")),
    ('categorical_transformer', SimpleImputer(strategy='most_frequent'),  make_column_selector(dtype_include="object"))]
    , remainder='drop')

new_data = cleaner.fit_transform(data)
```

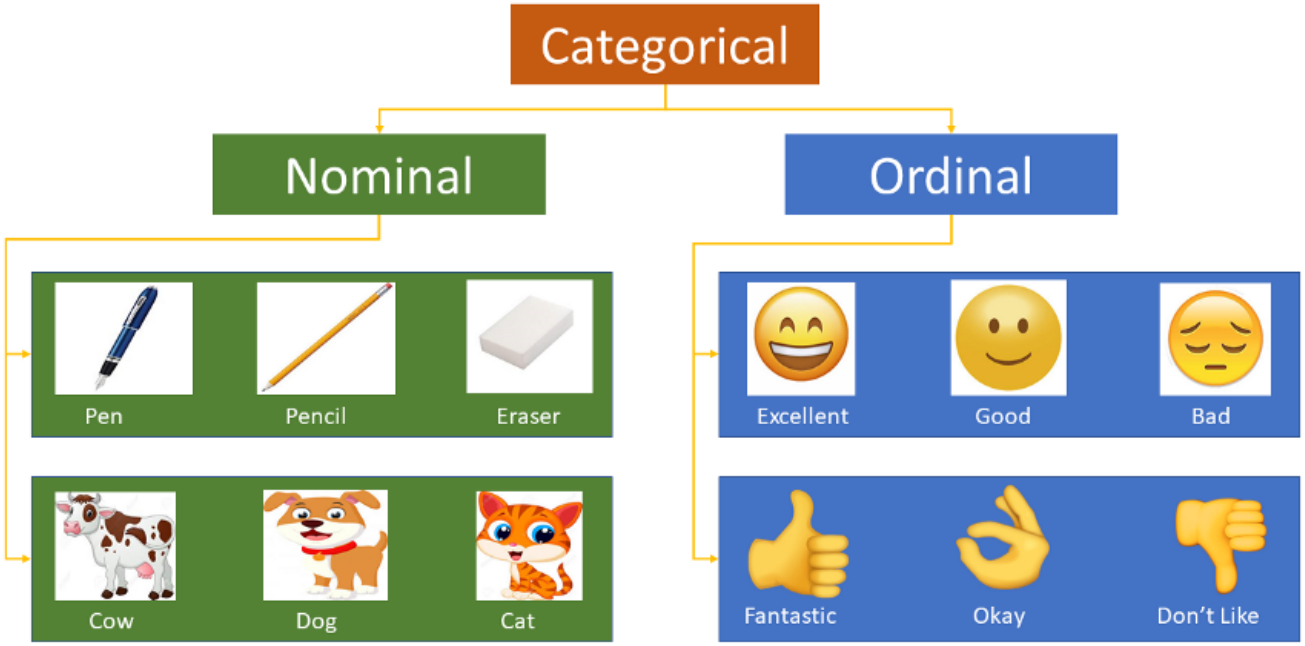
Data preprocessing

Encodage des variables qualitatives *(encoding)*

Encodage des variables quantitatives

La plupart des algorithmes d'apprentissage automatique ne peuvent pas traiter les variables qualitatives, à moins de les convertir en valeurs numériques. Les performances de nombreux algorithmes varient en fonction de la manière dont les variables catégorielles sont encodées.

Les variables catégorielles peuvent être divisées en deux catégories : Nominales (pas d'ordre particulier) et Ordinales (un certain ordre).



Encodage One-Hot

Le *one-hot encoding* est une méthode de conversion des données pour les préparer à un algorithme et obtenir une meilleure prédiction.

Il consiste à encoder une variable qualitative à **n** valeurs sur un vecteur de dimension **n** dont une seule prend la valeur 1. La dimension égale à 1 représente la valeur prise par la variable.

Color
Red
Red
Yellow
Green
Yellow

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

Les algorithmes d'apprentissage automatique traitent l'ordre des nombres comme un attribut de signification. En d'autres termes, ils liront un nombre plus élevé comme meilleur ou plus important qu'un nombre inférieur.

Cependant, les valeurs prises par une variable nominale n'ont aucun ordre/ classement, et cela peut entraîner des problèmes avec les prédictions et de mauvaises performances. C'est alors qu'un codage *one-hot* sauve la situation.

Encodage ordinal

Dans le codage ordinal, chaque valeur de catégorie unique se voit attribuer une valeur entière.

Pour certaines variables, un codage ordinal peut être suffisant. Les valeurs entières ont une relation ordonnée naturelle entre elles et les algorithmes d'apprentissage automatique peuvent être capables de comprendre et d'exploiter cette relation.

Valeur originale	Valeur encodée
mauvais	0
bon	1
très bon	2
excellent	3

Encodage des étiquettes (labels)

Dans l'encodage des étiquettes en Python, nous remplaçons la valeur qualitative par une valeur numérique comprise entre 0 et le nombre de classes moins 1.

Cet encodage est similaire à l'encodage ordinal, pourtant en pratique nous utilisons deux fonctions différentes pour encoder les paramètres d'entrée et le paramètre de sortie, car l'ordre dans ce cas peut ne pas être très important.

Valeur originale	Valeur encodée
Chat	0
Chien	1
Cheval	2
Crocodile	3

Encodage des variables qualitatives

Code python

```python
import numpy as np
from sklearn.preprocessing import OneHotEncoder, LabelEncoder, OrdinalEncoder

# One-hot encoding : codage de variables nominales
X = np.array([["A"],["A"],["B"],["C"]])  # exemple d'un vecteur X = (A,A,B,C)
enc = OneHotEncoder(sparse=False)        # définition de l'encodeur que nous souhaitons utiliser
X_enc = enc.fit_transform(X)             # encodage du vecteur X

# Ordinal encoding : codage de variables ordinale
X1 =  [["High"],["Low"],["Low"],["Medium"]] # un deuxieme exemple d'une variable ordinale
enc1 = OrdinalEncoder(categories=[["Low","Medium","High"]]) # il faut préciser l'ordre d'importance à la fonction
X1_enc = enc1.fit_transform(X1)

# Label encoding : codage de la variable cible
y = ["A","B","B","C","D"] # et un troisième exemple ou nous avons 4 classes.
enc2 = LabelEncoder()
y_enc = enc2.fit_transform(y)
```

Data preprocessing

Transformations des variables quantitatives

Transformations des variables quantitatives

La transformation des données est le processus qui consiste à convertir des données brutes dans un format ou une structure qui conviendrait mieux à la construction de modèles.

Il s'agit d'une étape impérative dans le prétraitement de données qui facilite la découverte d'informations.

Pourquoi faut-il transformer les données ?

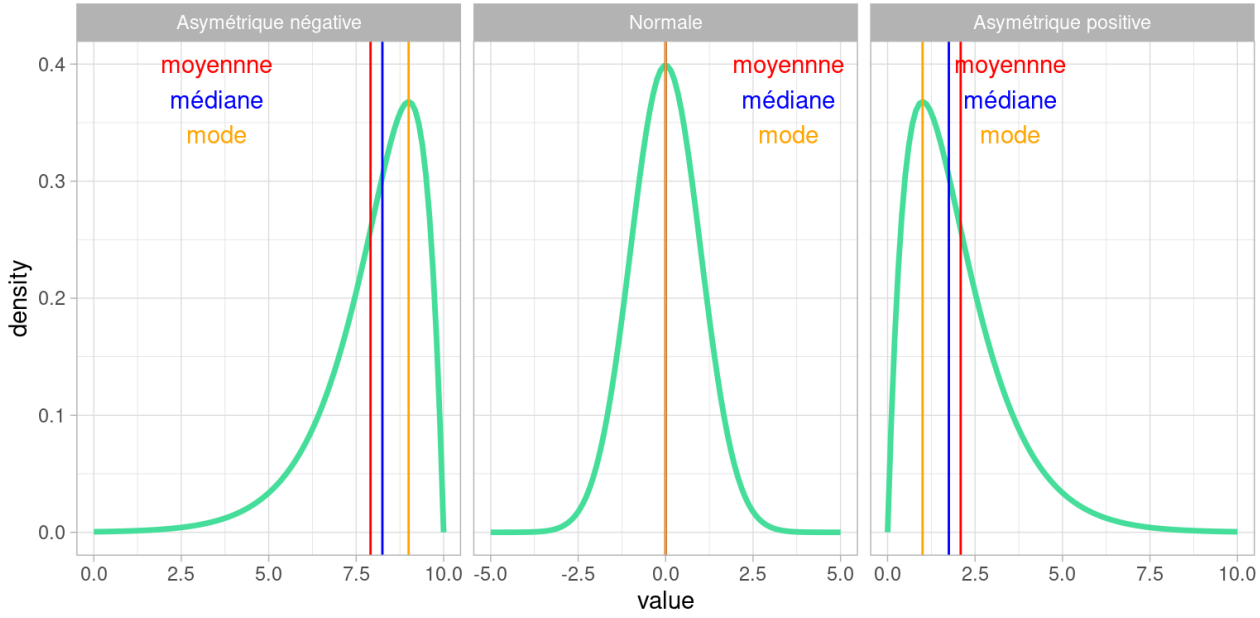
- l'algorithme est plus susceptible d'être biaisé lorsque la distribution des données est asymétrique.
- La transformation des données à la même échelle permet à l'algorithme de mieux comparer la relation relative entre les points de données.

Les transformations de puissance

Les algorithmes d'apprentissage automatique tels que la **régression linéaire** et les **réseaux bayésiens** supposent que les variables numériques ont une distribution de probabilité gaussienne.

Il se peut que vos données n'aient pas une distribution gaussienne, mais plutôt une distribution de type gaussien (presque gaussienne, mais avec des valeurs aberrantes ou une asymétrie) ou une distribution totalement différente (exponentielle,…).

Ainsi, vous pouvez obtenir de meilleures performances pour un large ensemble d'algorithmes d'apprentissage automatique en transformant les variables d'entrée et/ou de sortie pour qu'elles aient une distribution gaussienne ou plus gaussienne.



Les transformations de puissance

Les transformations de puissance telles que la transformation de **Box-Cox** et la transformation de **Yeo-Johnson** offrent un moyen d'effectuer ces transformations sur vos données.

Box-Cox exige que les données d'entrée soient strictement positives, tandis que **Yeo-Johnson** supporte les données positives ou négatives.

Box-Cox

$$B(x,\lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{si } \lambda \neq 0 \\ \log(x) & \text{si } \lambda = 0 \end{cases}$$

- Si λ >1 cela amplifie les grandes valeurs de x.
- si λ < 1 cela réduit les grandes valeurs de x.

Yeo-Johnson

$$\Psi(\lambda, y) = \begin{cases} ((y+1)^\lambda - 1)/\lambda & \text{si } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{si } \lambda = 0, y \geq 0 \\ -[(-y+1)^{(2-\lambda)} - 1]/(2-\lambda) & \text{si } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{si } \lambda = 2, y < 0 \end{cases}$$

- Il s'agit d'une amélioration de la méthode Box-Cox, qui permet de prendre en compte également les valeurs négatives.

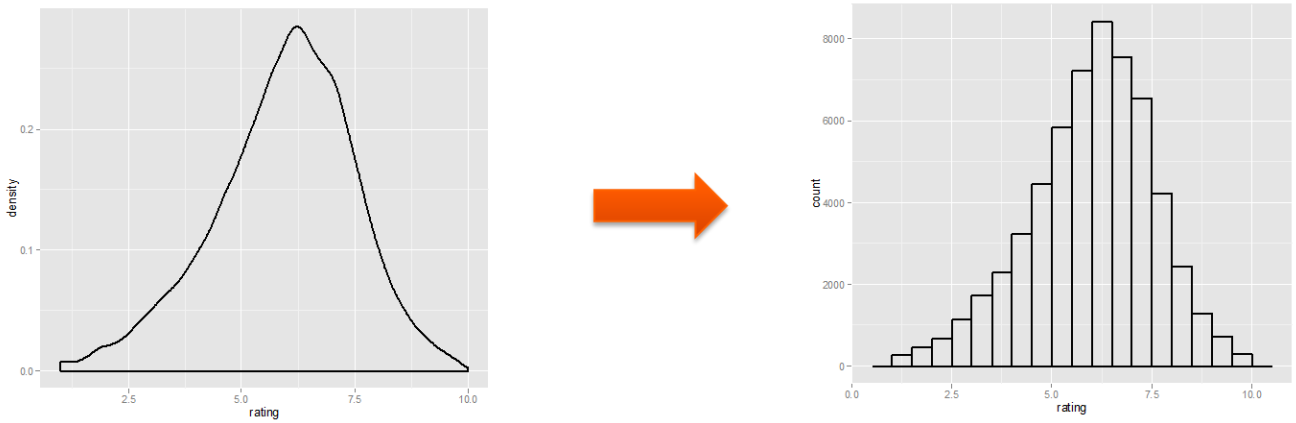
En pratique, les algorithmes s'occupent de définir la valeur de **λ** qui minimise l'asymétrie et rend les données proches d'une distribution normale.

Binning

Le ***binning*** (de l'anglais pour *"mise en récipient"*) est utilisé pour la transformation d'une variable continue en une caractéristique qualitative.

Les valeurs des données sont divisées en petits intervalles (ou catégories) appelés *bins.*

Cela a un effet de lissage sur les données d'entrée et peut améliorer les performances de quelques modèles. Il peut également être utilisé pour identifier les valeurs manquantes ou aberrantes.



Nous allons voir deux types de *binning* :

- Le *binning* à largeur égale,
- le *binning* à fréquence égale.

Binning

Le binning à largeur égale

Cet algorithme divise la variable continue en plusieurs catégories ayant des intervalles de même largeur, tel que :

$$l = \frac{max - min}{n\_bins}$$

avec :

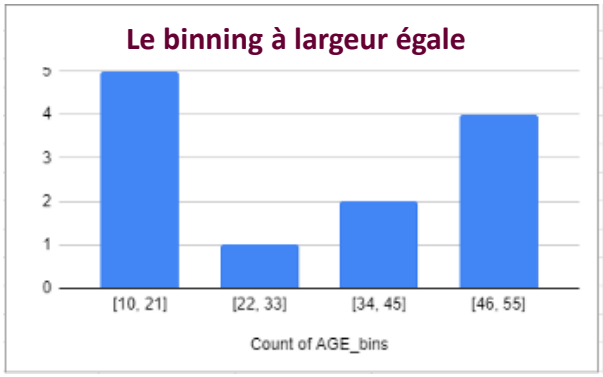
- $\boldsymbol{n\_bins}$ = nombre de catégories souhaités.
- $\boldsymbol{max}$ & $\boldsymbol{min}$ = la valeur maximale et la valeur minimale d'une variable.
- $\boldsymbol{l}$ = largeur des catégories.

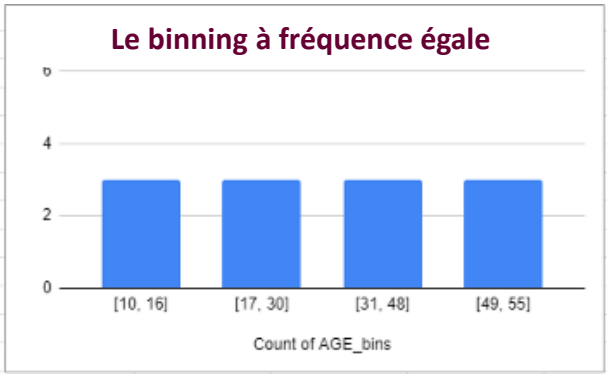
Le binning à fréquence égale

Cet algorithme divise les données en plusieurs catégories ayant approximativement le même nombre de valeurs. Le nombre de valeurs ($freq$) à inclure dans chaque catégorie est donc :

$$freq = \frac{nombre\ de\ données}{n\_bins}$$

AGE
10
15
16
18
20
30
35
42
48
50
52
55





Binarisation

La binarisation est le processus de transformation d'une variable numérique en une variable binaire.

Ceci nécessite de définir un seuil, où les valeurs de la variable supérieures au seuil seront transformées en 1, et sinon elles seront mises à 0.

Bien que facile, la binarisation est très utilisée pour convertir un problème de régression (c.-à-d., la variable cible est continue) en un problème de classification (c.-à-d., la variable cible est une classe, soit 0 ou 1).

Transformation des variables quantitatives

Code python

```
1  import pandas as pd
2  from sklearn.preprocessing import PowerTransformer, KBinsDiscretizer, Binarizer
3  import matplotlib.pyplot as plt
4
5  ''' ------Power transform-------'''
6  data = pd.read_csv("sample_dataset.csv")                    # importer le jeu de données csv
7  data['mean texture'].hist()                                 # représenter le paramètre "mean texture" sous forme d'histogrammes
8  plt.show()
9  power = PowerTransformer('yeo-johnson',standardize=False)   # définir le transformateur que vous voulez utiliser
10 trans_data = power.fit_transform(data)                      # transformer les données
11 plt.hist(trans_data[:,1])                                   # afficher le 1er paramètre des données transformées sous forme d'histogrammes
12 plt.show()
13
14 ''' ------Binning-------'''
15 binner = KBinsDiscretizer(strategy = 'uniform', n_bins = 5, encode = 'onehot-dense')    # un binning en 5 catégories de largeur fixe.
16 bins_data = binner.fit_transform(data)
17
18 ''' ------Binning-------'''
19 converter = Binarizer(threshold = 12)                       # dans ce cas, le seuil est égal à 12
20 binary_param = converter.fit_transform(data.iloc[:,-2:-1])   # convertir la dérnière colonne
```

Ligne 9 : changer ‘yeo-johnson’ par ‘box-cox’ pour appliquer la transformation Box-Cox.

Ligne 15 : changer ‘uniform’ par ‘quantile’ pour appliquer un binning à fréquence égale.

Data preprocessing

Mise à l'échelle (data scaling)

Data scaling

Introduction

La mise à l'échelle transforme les colonnes afin qu'elles aient le même ordre de grandeur. C'est parfois nécessaire car des ordres de grandeur différents peuvent affecter l'importance des caractéristiques perçues par le modèle.

De plus, ces transformations conservent la même corrélation entre les variables et entre chaque variable et la variable cible. De cette façon, le modèle ne pense pas que des variables sont plus importantes que les autres.

Les deux techniques de mise à l'échelle les plus couramment utilisées sont :

- Normalisation Z-score (*Standardization*)
- Normalisation Min-Max (*Min-Max scaler*)

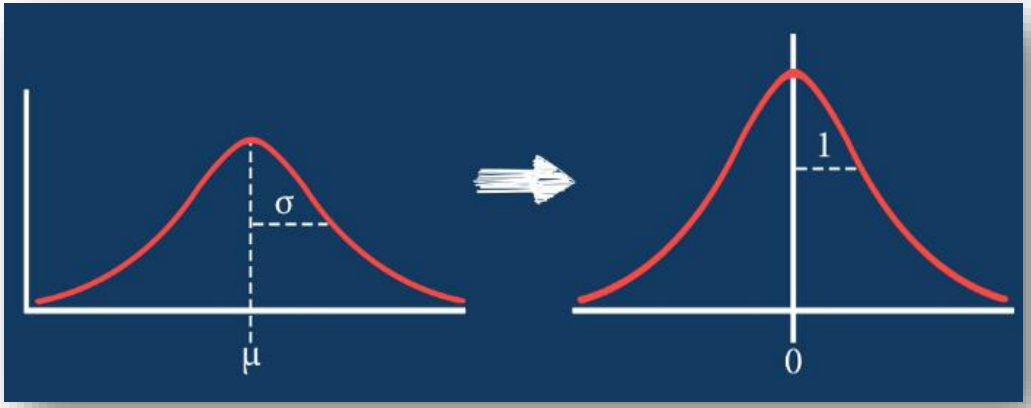
A	B	C
12	-40	9
20	1	4
25	69	8
33	-15	7

Mise à l'échelle

A	B	C
-1,54	-1,19	0,98
0,03	1,56	0,22
0,35	0,01	0,53
1,23	-0,37	0

Normalisation Z-score

La **normalisation Z-score** (*standardization*) est le processus de remise à l'échelle des variables χ afin qu'elles aient une moyenne **μ=0** et un écart-type **σ=1**.



Techniquement, cette normalisation centre et normalise les données en soustrayant la moyenne et en divisant par l'écart-type. Les données résultantes **(z)** sont appelées des **données centrées réduites** et peuvent être calculées comme suit :

$$z = \frac{\chi - \mu}{\sigma}$$

Cependant, cette approche est très sensible à la présence de valeurs aberrantes, puisqu'elles influencent à la fois le calcul de la moyenne et de l'écart-type.

Normalisation Min-Max

La normalisation **Min-Max** est une remise à l'échelle des données à afin que toutes les valeurs se situent dans une nouvelle plage de 0 et 1.

Pour chaque valeur d'une variable, une normalisation **Min-Max** consiste à soustraire la valeur minimale d'une variable, puis à la diviser par son étendue. L'étendue est la différence entre la valeur maximale et la valeur minimale.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Une normalisation **Min-Max** préserve la forme de la distribution d'origine. Elle ne modifie pas de manière significative les informations contenues dans les données initiales.

Notez qu'une normalisation **Min-Max** ne réduit pas l'importance des valeurs aberrantes.

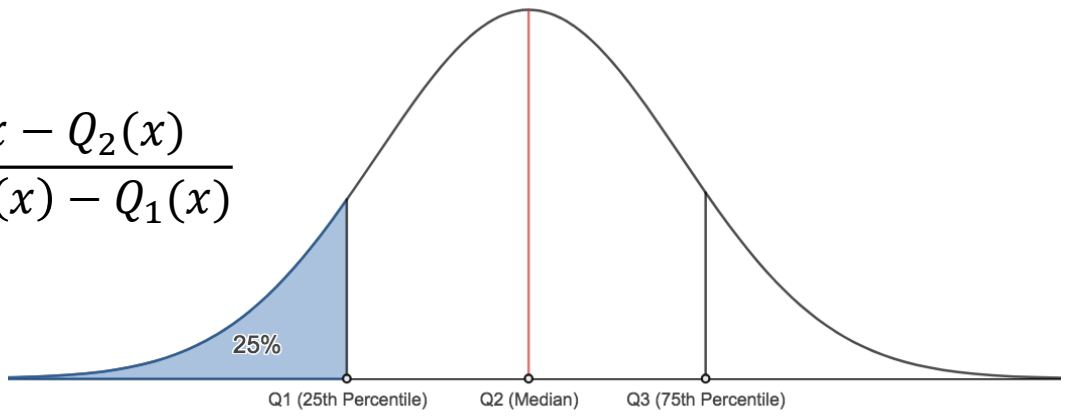
Normalisation robuste

Une approche de la normalisation des variables d'entrée en présence de valeurs aberrantes consiste à ignorer ces dernières dans le calcul de la moyenne et de l'écart-type, puis à utiliser les valeurs calculées pour mettre la variable à l'échelle.

Cette méthode est appelée normalisation robuste ou mise à l'échelle robuste des données.

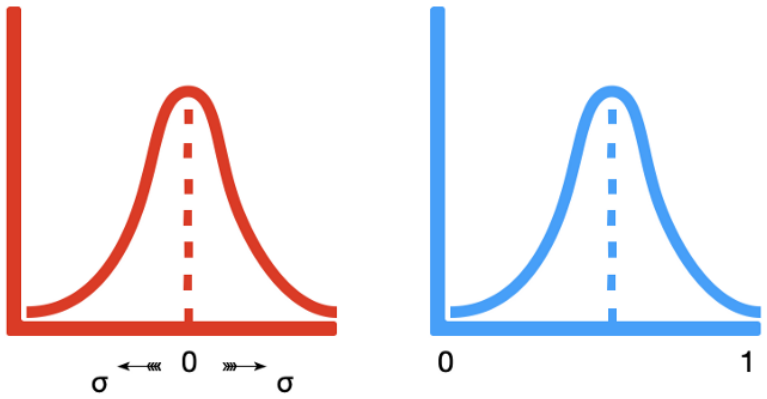
On peut y parvenir en calculant la médiane (**50e** percentile) et les **25e** et **75e** percentiles. Les valeurs de chaque variable sont ensuite soustraites de leur médiane et divisées par l'écart interquartile (**IQR**), qui est la différence entre les 75e et 25e percentile.

$$x_{scaled} = \frac{x - Q_2(x)}{Q_3(x) - Q_1(x)}$$



La variable résultante a une médiane nulle. Elle n'est pas biaisée par des valeurs aberrantes et ces dernières sont toujours présentes avec les mêmes relations au regard des autres valeurs.

Z-score vs Min-Max



- La normalisation **Min-Max** est pratiquée lorsque les données n'ont pas une distribution gaussienne, tandis que la normalisation **z-score** est préférée pour les données ayant une distribution gaussienne.
- La normalisation **Min-Max** s'étend sur une plage de [0,1]. La normalisation **z-score** n'est pas limitée par une plage.
- La normalisation **Min-Max** est plus affectée par les valeurs aberrantes par rapport à la normalisation **z-score** .
- La normalisation **Min-Max** est envisagée lorsque les algorithmes ne font pas d'hypothèses sur la distribution des données. La normalisation **z-score** est utilisée lorsque les algorithmes font des hypothèses sur la distribution des données.

Mise à l'échelle des données

Code python

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler

data = pd.read_excel("sample_dataset.xlsx")    # Importer le jeu de données

'''Normalisation z-score'''
scaler = StandardScaler()                      # préciser quelle normalisation vous souhaitez utiliser
data_scaled = scaler.fit_transform(data)       # données normalisées

'''Normalisation Min-Max'''
scaler1 = MinMaxScaler()
data_scaled = scaler1.fit_transform(data)

'''Normalisation robuste'''
scaler2 = RobustScaler()
data_scaled = scaler2.fit_transform(data)
```

Data preprocessing

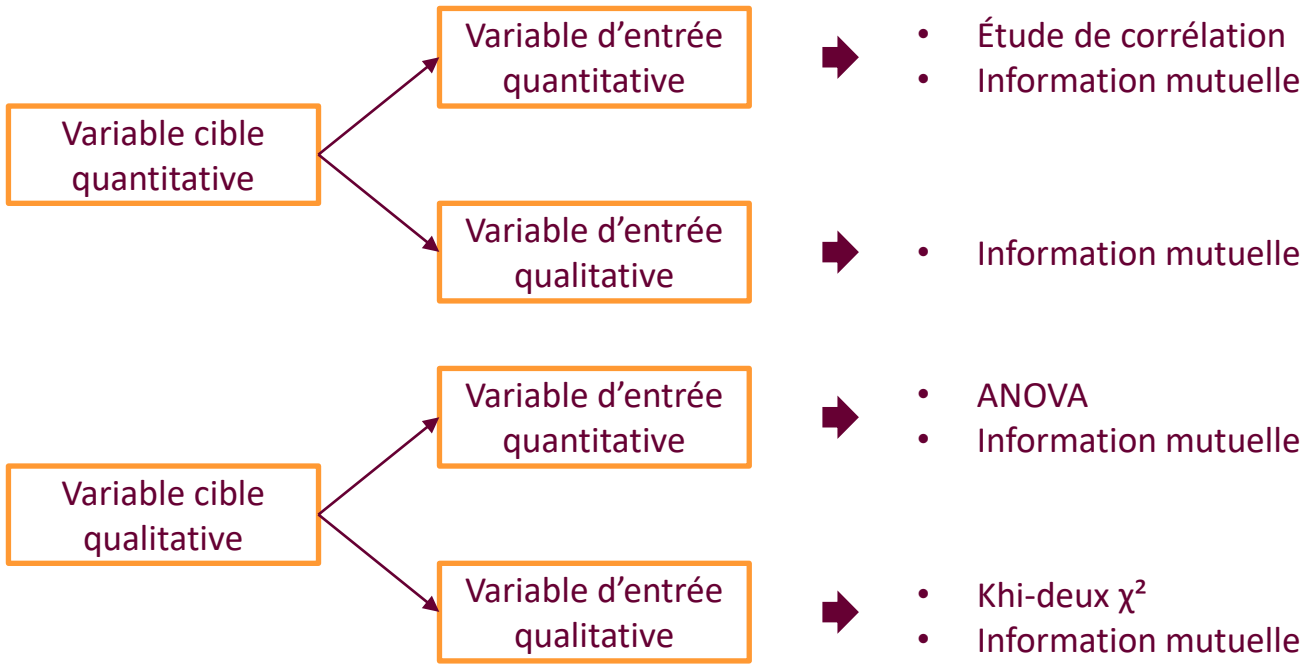
Sélection des caractéristiques (Feature selection)

Introduction

La sélection des caractéristiques est le processus qui consiste à réduire le nombre de variables d'entrée lors du développement d'un modèle de *machine learning*.

Les méthodes de sélection des caractéristiques basées sur les statistiques consistent à évaluer la relation entre chaque variable d'entrée et la variable cible et à sélectionner les variables d'entrée qui sont fortement liées à la variable cible.

Ces méthodes peuvent être rapides et efficaces, cependant le choix des mesures statistiques à utiliser dépend du type de données des variables d'entrée et de sortie.



Feature selection

Pourquoi sélectionner des caractéristiques ?

- Les modèles simples sont plus faciles à interpréter
- Des temps d'apprentissage plus courts
- Une meilleure généralisation en réduisant le phénomène de sur-apprentissage.
- Réduction du risque d'erreurs de données pendant l'utilisation du modèle
- Malédiction de la dimensionnalité : les algorithmes d'apprentissage automatique ont tendance à avoir de mauvais comportements dans les espaces de grande dimension.

Avant de commencer

Lors de la sélection des caractéristiques, une première étape consiste à effectuer un contrôle et une identification rapide de certaines variables à supprimer :

- **Variables constantes** : sont celles qui ne présentent qu'une seule valeur pour toutes les observations de l'ensemble de données.

- **Variables quasi-constantes** : caractéristiques pour lesquelles une seule valeur est partagée par la grande majorité des observations de l'ensemble de données (généralement entre 95% et 99%).

- **Variables dupliquées** : Lorsque deux variables du jeu de données présentent la même valeur pour toutes les observations, il s'agit essentiellement de la même variable.
  De plus, et c'est une chose à garder à l'esprit, les variables dupliquées peuvent apparaître après un processus qui génère de nouvelles variables à partir de variables existantes, comme le "*one hot encoding*".

Étude de corrélation

Le premier cas s'agit d'un problème de régression avec des variables d'entrée numériques.

Les techniques les plus courantes consistent à utiliser un coefficient de corrélation, tel que celui de **Pearson** pour une corrélation linéaire, ou des méthodes basées sur les rangs pour une corrélation non linéaire afin d'évaluer la relation entre une variable d'entrée et la variable de sortie.

- **Coefficient de corrélation de Pearson (linéaire).**
- Coefficient de rang de Spearman (non linéaire).

L'étude de corrélation permet aussi d'évaluer les relations entre les variables d'entrée.

Si deux ou plusieurs variables d'entrée sont fortement corrélées entre elles, cela peut signifier qu'elles expriment toutes deux la même information. Il y a donc une raison de supprimer l'une de ces variables du modèle.

Si vous n'êtes pas sûr de la variable à supprimer, vous pouvez toujours envisager de construire deux modèles, un avec chacune des variables.

Étude de corrélation

résumé du principe de base

l'hypothèse centrale de la sélection des caractéristiques par corrélation est qu'un bon ensemble de variables d'entrée, contient des variables qui sont fortement corrélées avec la variable cible mais non corrélées entre elles.

Définition . *Le* coefficient de corrélation *des variables aléatoires* X *et* Y *est défini par :*

$$\rho(X,\ Y) = \frac{\mathbf{Cov}(X,\ Y)}{\sigma(X)\,\sigma(Y)}$$

si X *et* Y *possèdent des écart-types non nuls.*

Théorème . *Si* X *et* Y *sont indépendantes leur coefficient de corrélation est nul.*

La réciproque est fausse ; le coefficient de corrélation donne donc une indication plus ou moins précise de l'indépendance de deux variables aléatoires.

Définition . *Si* $\rho(X,\ Y)$ *est nul les deux variables sont dites* non-corrélées.

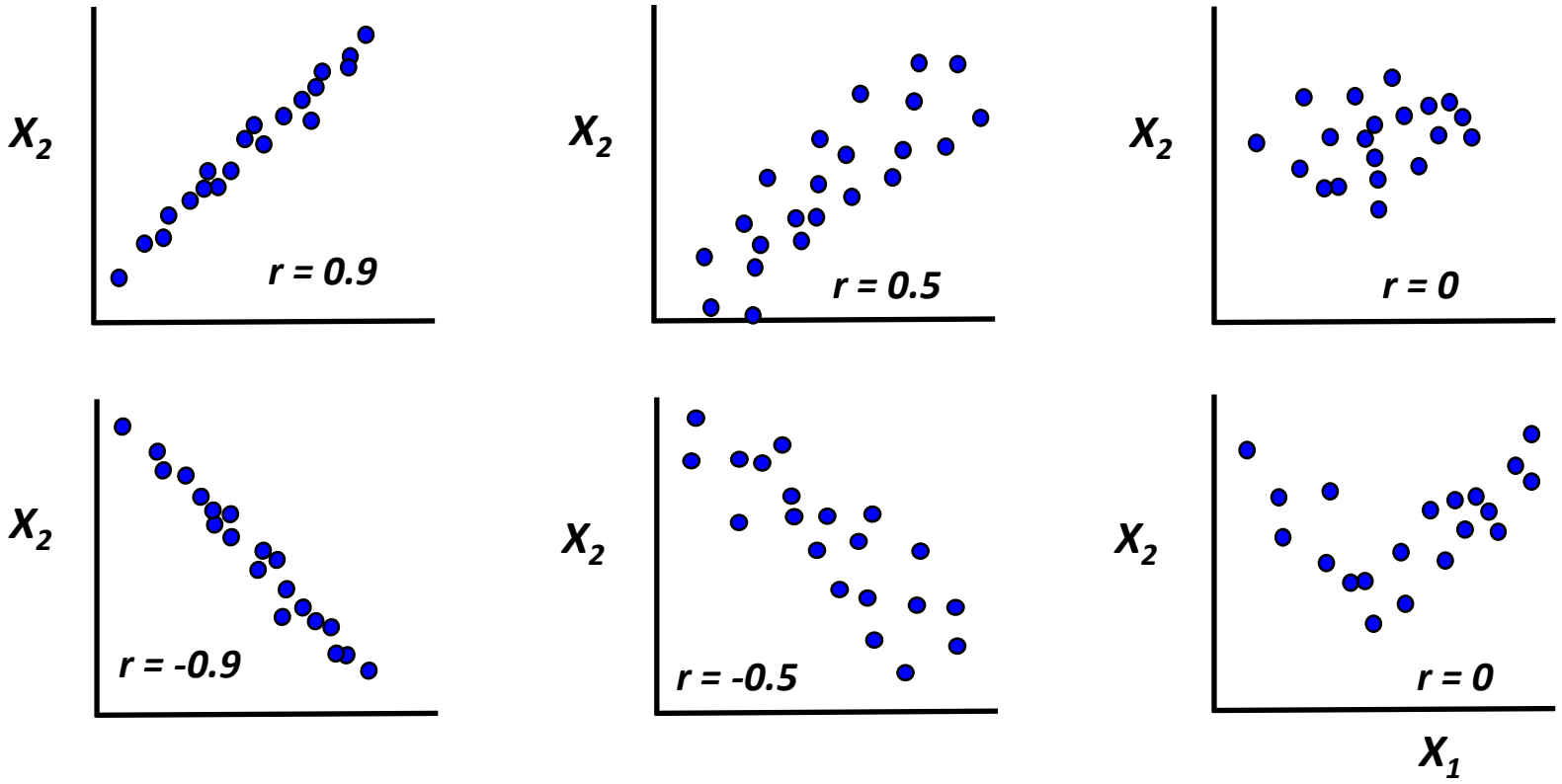
Estimation de $Cov(X,Y) = \hat{\sigma}_{xy}$

$$Cov(X,Y) = \hat{\sigma}_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)(Y_i - \overline{Y}_n)$$

Étude de corrélation

Coefficient de corrélation de Bravais-Pearson :

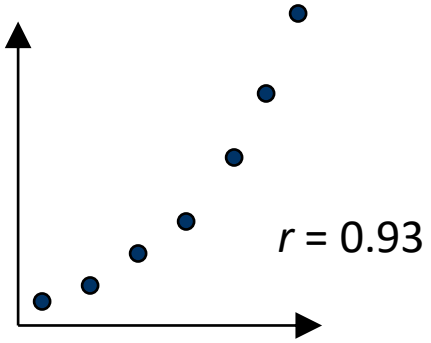
$$r = \frac{Cov(X,Y)}{s_x s_y} = \frac{s_{xy}}{s_x s_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum(x_i - \bar{X})^2}\sqrt{\sum(y_i - \bar{Y})^2}}$$

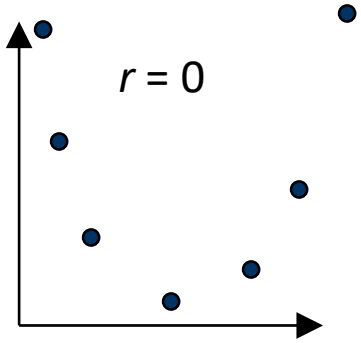


Étude de corrélation

Le coefficient r de Bravais Pearson peut prendre toutes les valeurs réelles comprises dans l'intervalle [-1, 1]. Plus la valeur absolue de r est proche de 1, plus il y a de conformité avec le modèle linéaire. Un coefficient positif indique que les deux variables « évoluent » dans le même sens. Un coefficient négatif indique qu'il existe une relation inverse entre les variables x et y. Le coefficient de corrélation est un indice indépendant de la moyenne.

Ce coefficient de corrélation doit être manié avec grande précaution.

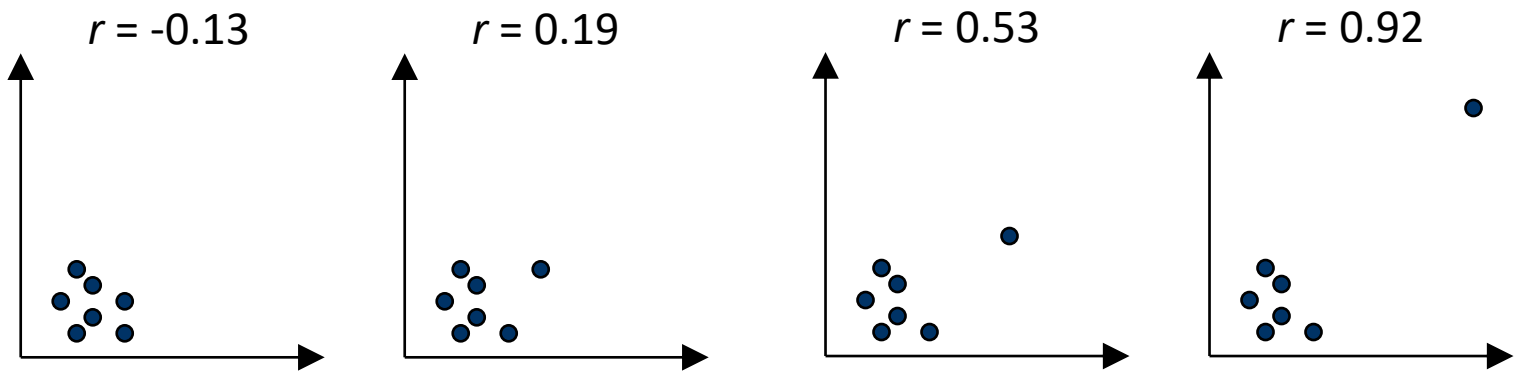




- *r* donne le degré de liaison linéaire.
- Dépendance curvilinéaire forte et r faible dans le 2eme cas.
- Le diagramme *xy* doit donc toujours être examiné en même temps que la valeur de *r*.

Étude de corrélation

Ce coefficient de corrélation doit être manié avec grande précaution



Le coefficient de corrélation peut produire de hautes valeurs si des points isolés sont présents.

Information mutuelle

L'information mutuelle est une mesure de la dépendance mutuelle de deux variables.

En d'autres termes, l'information mutuelle quantifie la "quantité d'information" obtenue sur une variable aléatoire par l'observation de l'autre variable aléatoire.

L'information mutuelle d'un couple $(\mathbf{X}; \mathbf{Y})$ de variables représente leur degré de dépendance au sens probabiliste.

On dit que deux variables sont indépendantes si la réalisation de l'une n'apporte aucune information sur la réalisation de l'autre.

Le coefficient de corrélation est une mesure du **cas particulier de dépendance** dans lequel la relation entre les deux variables est **linéaire**.

L'information mutuelle $\mathbf{I}(\mathbf{X}; \mathbf{Y})$ est nulle si et seulement si les variables sont indépendantes, et croit lorsque la dépendance augmente.

Information mutuelle

$$\mathrm{I}(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{(X,Y)}(x,y) \log \left(\frac{P_{(X,Y)}(x,y)}{P_X(x)\, P_Y(y)} \right)$$

$P_{(X,Y)}(x,y)$: est la probabilité que x et y se produisent en même temps.

$P_X(x)$: est la probabilité de l'occurrence de x.

$P_Y(y)$: est la probabilité de l'occurrence de y.

Exemple :

$P_X(x)$ est la probabilité que la température à Metz soit de 10°C.

$P_Y(y)$ est la probabilité que nous soyons en juin.

$P_{(X,Y)}(x,y)$ est la probabilité que la température à Metz soit de 10°C en juin.

Information mutuelle

Propriétés :

- $\mathbf{I(X;Y)}$ = 0 si et seulement si X et Y sont des variables aléatoires indépendantes.
- L'information mutuelle est positive ou nulle.
- L'information mutuelle est symétrique.
- Si $\mathbf{g_1}$ et $\mathbf{g_2}$ sont deux transformations (*ex. : normalisation min_max, logarithmiques, …*) alors $\mathbf{I(g_1(X);g_2(Y)) \leq I(X;Y)}$. Ceci signifie qu'aucune transformation sur les données brutes ne peut faire apparaître de l'information.

Information mutuelle

Supposons que nous ayons le tableau de distribution suivant :

	Y = 0	Y = 1	Y = 2	Marginale*
X = 0	0,2	0,1	0,2	0,5
X = 1	0,0	0,2	0,1	0,3
X = 2	0,1	0,0	0,1	0,2
Marginale*	0,3	0,3	0,4	1,0

$$\mathrm{I(X\,,Y)} = p_X(0,0) * \log\left(\frac{p_{(X,Y)}(0,0)}{p_X . p_Y}\right) + \cdots + p_X(2,2) * \log\left(\frac{p_{(X,Y)}(2,2)}{p_X . p_Y}\right)$$

$$= 0{,}2 * \log\left(\frac{0{,}2}{0{,}5 * 0{,}3}\right) + \cdots + 0{,}1 * \log\left(\frac{0{,}1}{0{,}2 * 0{,}4}\right)$$

$$= 0{,}33$$

Le calcul de l'information mutuelle peut être appliqué pour évaluer la dépendance entre tous les types de variables, qualitatives et quantitatives.

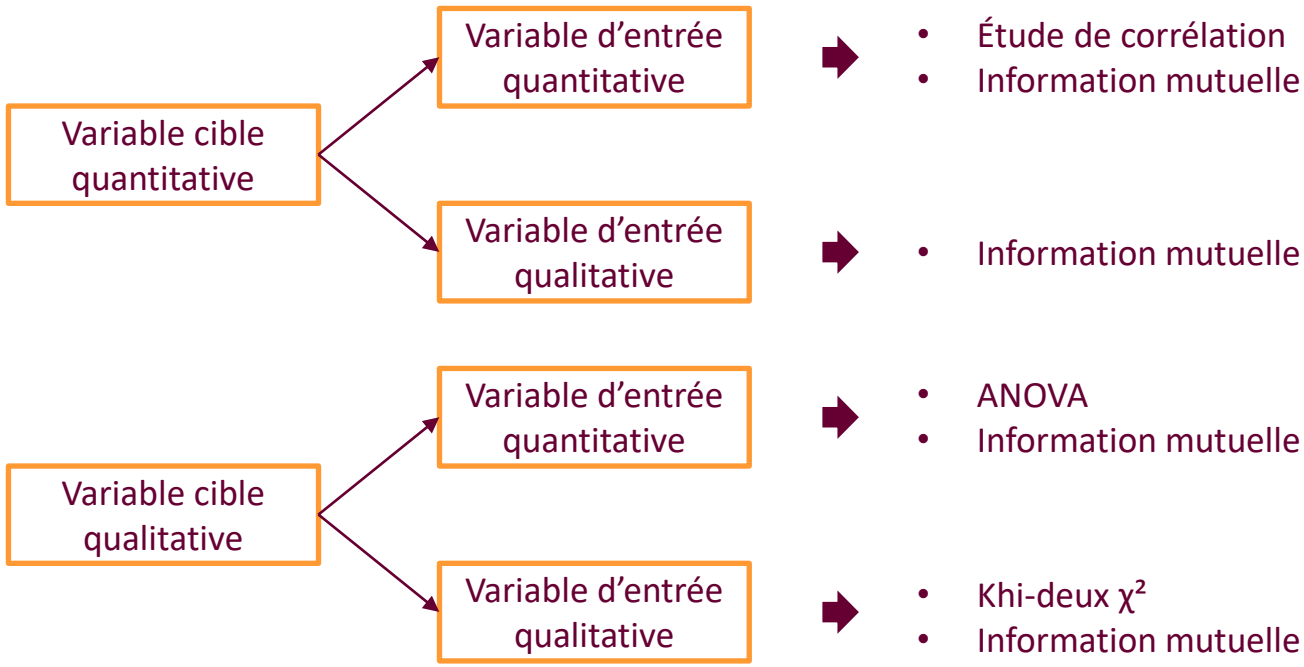
* la loi de probabilité marginale d'une variable aléatoire à plusieurs dimensions, est la loi de probabilité d'une de ses composantes.

Introduction

La sélection des caractéristiques est le processus qui consiste à réduire le nombre de variables d'entrée lors du développement d'un modèle de *machine learning*.

Les méthodes de sélection des caractéristiques basées sur les statistiques consistent à évaluer la relation entre chaque variable d'entrée et la variable cible et à sélectionner les variables d'entrée qui sont fortement liées à la variable cible.

Ces méthodes peuvent être rapides et efficaces, cependant le choix des mesures statistiques à utiliser dépend du type de données des variables d'entrée et de sortie.



ANOVA

L'ANOVA, ou l'analyse de la variance, est une méthode statistique utilisée pour vérifier s'il existe des **différences significatives entre les moyennes** de deux groupes ou plus.

Elle est utilisée pour déterminer si les variations d'une variable de réponse sont dues à des différences dans une ou plusieurs variables d'entrée qualitatives, également appelées **facteurs**.

L'ANOVA peut être utilisée pour les plans à une direction (un facteur) et à plusieurs directions (plus d'un facteur). Le résultat d'un test ANOVA peut être utilisé pour confirmer ou rejeter l'hypothèse nulle selon laquelle les moyennes de tous les groupes sont égales.

Il existe **généralement** trois types d'ANOVA :

- **ANOVA à un facteur : utilisée pour comparer les moyennes d'une variable indépendante et d'une variable dépendante.**
- ANOVA à deux facteurs : utilisée pour comparer les moyennes de deux variables indépendantes et d'une variable dépendante.
- ANOVA à trois facteurs : utilisée pour comparer les moyennes de trois variables indépendantes et d'une variable dépendante.

ANOVA à un facteur

"Prenons un exemple !!"

Présentation des deux variables à étudier :

- Les forêts : **Variable qualitative** appelée facteur, et contenant trois modalités.
- Hauteur des arbres : **Variable quantitative** ou réponse notée Y.

Forêt 1	Forêt 2	Forêt 3
23,3	18,9	22,5
24,4	21,1	22,9
24,6	21,1	23,7
24,9	22,1	24,0
25,0	22,5	24,0
26,2	23,5	24,5

Objectif :

- Etudier si l'effet de ce facteur est significatif sur la variable réponse.

ANOVA à un facteur

Etape 1 :

Calculons la moyenne $\bar{y}_i$ et la variance $s^2{}_i$ de chaque échantillon ***i*** , avec ***I = 3 et J = 6*** :

$$\bar{y}_i = \frac{1}{J}\sum_{j=1}^{J} y_{ij}, \quad i = 1,\ldots, I$$

$$s^2{}_i(y) = \frac{1}{J}\sum_{j=1}^{J} (y_{ij} - \bar{y}_i)^2, \quad i = 1,\ldots, I$$

$\bar{y}_{foret\_1}$	$\bar{y}_{foret\_2}$	$\bar{y}_{foret\_3}$
24,75	21,53	23,6

$s^2{}_{foret\_1}$	$s^2{}_{foret\_2}$	$s^2{}_{foret\_3}$
0,73	2,07	0,47

*A noter que **J** peut varier d'un échantillon à l'autre.*

Etape 2 :

La moyenne de toutes les observations est la moyenne des moyennes de chaque échantillon:

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{I} y_{ij} \quad = \mathbf{23,29}$$

Etape 3 :

Calcul de la moyenne des variances.

$$\frac{1}{I}\sum_{i=1}^{I} s^2{}_i(y) \quad = \quad \frac{1}{3}(0{,}73 + 2{,}07 + 0{,}47) = 1{,}09$$

ANOVA à un facteur

Etape 4 :

Calcul de la variance des moyennes.

$$\frac{1}{I}\sum_{i=1}^{I}(\overline{y_i}-\overline{y})^2 = \frac{1}{3}((24{,}75-23{,}29)^2+(21{,}53-23{,}29)^2+(23{,}60-23{,}29)^2)$$

$$= 1{,}75$$

Etape 5 :

La variance de toutes les observations est la somme de la variance des moyennes (Etape 4) et de la moyenne des variances (Etape 3).

$$s^2(y) = 2{,}84$$

Nous multiplions cette somme par le nombre total de données, ce qui nous permet d'avoir la relation suivante :

$$\sum_{i=1}^{I}\sum_{j=1}^{J}(y_{ij}-\overline{y})^2 = J\sum_{i=1}^{I}(\overline{y_i}-\overline{y})^2+\sum_{i=1}^{I}(\sum_{j=1}^{J}(y_{ij}-\overline{y_i})^2) = 51{,}3$$

ANOVA à un facteur

Interprétation :

Nous multiplions cette somme par le nombre total de données, ce qui nous permet d'avoir la relation suivante :

$$\sum_{i=1}^{I}\sum_{j=1}^{J}(y_{ij}-\bar{y})^2 = J\sum_{i=1}^{I}(\bar{y_i}-\bar{y})^2+\sum_{i=1}^{I}(\sum_{j=1}^{J}(y_{ij}-\bar{y_i})^2)$$

Cette relation s'écrit : $\boldsymbol{SC_{tot} = SC_F + SC_R}$

avec :

- $\boldsymbol{SC_{tot}}$ la variation totale : dispersion des données autour de la moyenne générale.
- $\boldsymbol{SC_F}$ la variation due au facteur : dispersion des moyennes autour de la moyenne générale.
- $\boldsymbol{SC_R}$ la variation résiduelle : dispersion des données à l'intérieur de chaque échantillon autour de sa moyenne.

ANOVA à un facteur

rappel du principe de base

L'ANOVA permet de confirmer si l'hypothèse nulle (H0) est vérifiée, c.-à-d. si les moyennes des différents échantillons sont égales.

H0 : $\bar{y}_1 = \bar{y}_2 = \ldots = \bar{y}_I$

H1 : Les $\bar{y}_i$ ne sont pas toutes égales.

Interprétation :

- Si (H0) est vraie alors la variation due au facteur SC_F doit être petite par rapport à la variation résiduelle SC_R.
- Si (H1) est vraie alors la variation due au facteur SC_F doit être grande par rapport à la variation résiduelle SC_R .

Pour comparer ces quantités, **Fisher** a considéré le rapport des carrés moyens, avec ***I*** le nombre d'échantillons et ***n*** le nombre total de données.

Carré moyen associé au facteur $CM_F = \frac{SC_F}{I-1}$

Carré moyen résiduel $CM_R = \frac{SC_R}{n-I}$

ANOVA à un facteur

Décision :

Pour un seuil donné $\boldsymbol{\alpha}$ (5% en général) les tables de Fisher* nous fournissent une valeur critique $\boldsymbol{c}$ telle que :

Si $\boldsymbol{F_{I-1,n-I} < c}$ → H0 est vraie

Si $\boldsymbol{F_{I-1,n-I} \geq c}$ → H1 est vraie

Avec : $\boldsymbol{F_{I-1,n-I} = \frac{CM_F}{CM_R}}$

Par exemple, pour :

- I = 3
- n = 18
- α = 0.05

On trouve la valeur $\boldsymbol{c}$ qui correspond à $\boldsymbol{F_{2,15}}$:

- c = 3,682

Critical Values of the F-Distribution: $\alpha = 0.05$

Denom. d.f.	1	2	3	4
1	161.448	199.500	215.707	224.583
2	18.513	19.000	19.164	19.247
3	10.128	9.552	9.277	9.117
4	7.709	6.944	6.591	6.388
5	6.608	5.786	5.409	5.192
6	5.987	5.143	4.757	4.534
7	5.591	4.737	4.347	4.120
8	5.318	4.459	4.066	3.838
9	5.117	4.256	3.863	3.633
10	4.965	4.103	3.708	3.478
11	4.844	3.982	3.587	3.357
12	4.747	3.885	3.490	3.259
13	4.667	3.806	3.411	3.179
14	4.600	3.739	3.344	3.112
15	4.543	3.682	3.287	3.056

* https://www.stat.purdue.edu/~lfindsen/stat511/F_alpha_05.pdf

ANOVA à un facteur

Tableau de l'ANOVA

Il permet de résumer les différentes mesures de l'analyse.

Variation	variance	SC = var*n	ddl	CM = SC/ddl	F	Probabilité critique
Due au facteur	Variance des moyennes	SC_F	I-1	CM_F	$\frac{CM_F}{CM_R}$	p-value*
Résiduelle	Moyenne des variances	SC_R	n-I	CM_R		
Totale	Variance totale	SC_{tot}	n-1			

Application à notre exemple :

Variation	variance	SC = var*n	ddl	CM = SC/ddl	F	Probabilité critique
Due au facteur	1,75	31,88	2	15.94	12.31	0.0007
Résiduelle	1,09	19,43	15	1.29		
Totale	2,84	51,31	17			

p-value < 0.05, donc les hauteurs moyennes sont significativement différentes dans chaque forêt.

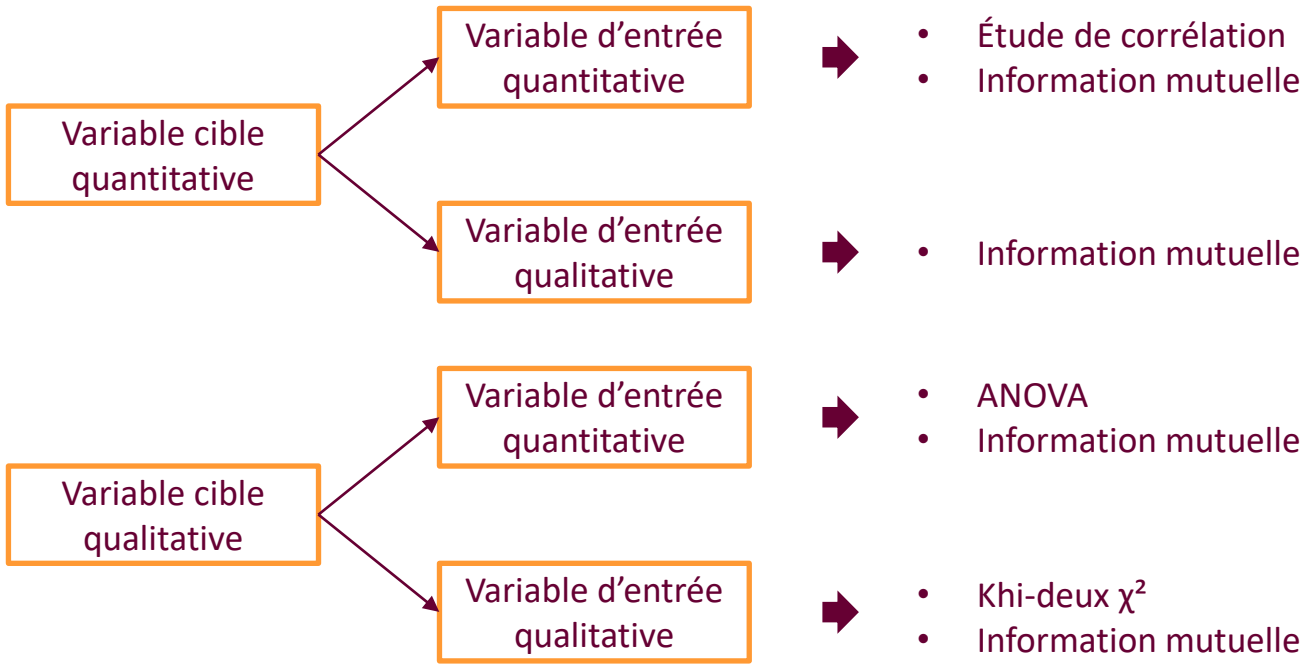
* Trouvée à partir de la valeur **c** fourni par la table de Fisher , généralement à l'aide d'un calculateur .

Introduction

La sélection des caractéristiques est le processus qui consiste à réduire le nombre de variables d'entrée lors du développement d'un modèle de *machine learning*.

Les méthodes de sélection des caractéristiques basées sur les statistiques consistent à évaluer la relation entre chaque variable d'entrée et la variable cible et à sélectionner les variables d'entrée qui sont fortement liées à la variable cible.

Ces méthodes peuvent être rapides et efficaces, cependant le choix des mesures statistiques à utiliser dépend du type de données des variables d'entrée et de sortie.



Khi-deux

Le test du **Khi-deux**, également connu sous le nom de test du **Khi-carré**, est une méthode statistique couramment utilisée pour la sélection de caractéristiques en apprentissage automatique.

Elle mesure la dépendance entre deux variable qualitative, telles qu'un paramètre d'entrée et un paramètre de sortie, en comparant les fréquences observées de leur cooccurrence aux fréquences attendues si elles étaient indépendantes.

La statistique de test résultante suit une distribution du **Khi-deux**, qui peut être utilisée pour déterminer l'importance de l'association entre la caractéristique et la sortie.

Les caractéristiques présentant des valeurs élevées du **Khi-deux** sont considérées comme plus pertinentes pour la tâche de prédiction et sont plus susceptibles d'être sélectionnées pour le modèle final.

Khi-deux

Les étapes de l'application du test du Khi-deux pour la sélection des caractéristiques dans l'apprentissage automatique sont les suivantes :

1. **Calculer la valeur du Khi-deux pour chaque variable d'entrée :** Pour chaque variable d'entrée, calculer les fréquences observées de ses valeurs dans les échantillons qui appartiennent à chaque classe de la variable cible, et les comparer aux fréquences attendues si la variable d'entrée et la variable cible étaient indépendantes.

2. **Calculez la *p-value* pour chaque variable d'entrée :** La valeur du Khi-deux suit une **distribution du Khi-deux**, qui peut être utilisée pour calculer la *p-value*, qui représente la probabilité d'observer une valeur du Khi-deux calculée.

3. **Sélectionnez les caractéristiques ayant les *p-values* les moins élevées :** Les variables d'entrée dont la *p-value* est inférieure à un certain seuil (par exemple, 0.05) sont considérées comme significatives et sont retenues pour le modèle de machine learning final.

Khi-deux

1. Calculer la valeur du Khi-deux pour chaque variable d'entrée :

Imaginons que nous menions une enquête pour déterminer s'il existe une relation entre le type de voiture qu'une personne conduit et son niveau de revenu.

Nous recueillons des données sur 370 personnes interrogées et créons un **tableau de contingence** avec les deux variables suivantes : type de voiture (économique, SUV) et niveau de revenu (faible, moyen).

	Moyen (M)	Faible (F)	
Économique (E)	200	60	260
SUV (S)	100	10	110
	300	70	**370**

Pour calculer la valeur du Khi-deux, nous devons comparer ce qui a été observé à ce qui est attendu (***O**bserved Vs **E**xpected*). Pour ce faire, nous créons un nouveau tableau, de sorte que chaque cellule représente les observations attendues en supposant que les deux variables sont indépendantes.

	Moyen (M)	Faible (F)
Économique (E)	$\frac{260 * 300}{370} = 210{,}81$	$\frac{260 * 70}{370} = 49{,}19$
SUV (S)	$\frac{110 * 300}{370} = 89{,}19$	$\frac{110 * 70}{370} = 20{,}81$

Khi-deux

1. Calculer la valeur du Khi-deux pour chaque variable d'entrée :

observé

	Moyen (M)	Faible (F)	
Économique (E)	200	60	260
SUV (S)	100	10	110
	300	70	**370**

attendu

	Moyen (M)	Faible (F)
Économique (E)	210,81	49,19
SUV (S)	89,19	20,81

La formule pour calculer la valeur du Khi-deux pour une variable d'entrée et une variable cible est la suivante :

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Où :

- O_i est la fréquence **observée** de la variable d'entrée pour la valeur ***i*** lorsque la valeur cible vaut **j**.
- E_i est la fréquence **attendue** de la variable d'entrée pour la valeur ***i*** lorsque la valeur cible vaut **j**.

Application :

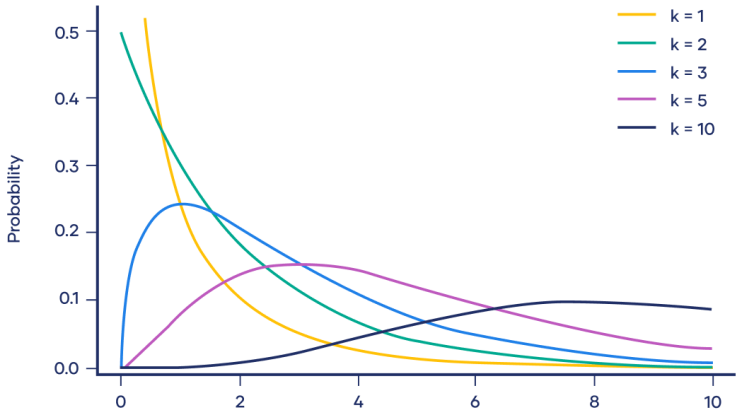
$$\chi^2 = \frac{(200 - 210{,}8)^2}{210{,}8} + \frac{(60 - 49{,}19)^2}{49{,}19} + \frac{(100 - 89{,}19)^2}{89{,}19} + \frac{(10 - 20{,}81)^2}{20{,}81} = 9{,}84$$

Khi-deux

2. Calculez la *p-value* pour chaque variable d'entrée:

Si nous échantillonnons une population plusieurs fois et calculons la statistique du test de Khi-deux de pour chaque échantillon, la statistique du test suivra une distribution du Khi-deux (figure à droite).

Nous pouvons voir comment la forme de la distribution du khi-deux change au fur et à mesure avec l'augmentation des **degrés de liberté (k).**



Pour calculer le degré de liberté, et donc savoir quelle distribution Khi-deux à utiliser pour le reste de l'analyse, il suffit d'appliquer la relation suivante :

$$\boldsymbol{k = (nombre\ de\ valeurs\ de\ la\ variable\ 1° - 1) * (nombre\ de\ valeurs\ de\ la\ variable\ 2° - 1)}$$

Dans notre exemple :

$$\boldsymbol{k = (2 - 1) * (2 - 1) = 1}$$

Nous utiliserons donc une distribution du **Khi-deux avec un degré de liberté de 1** pour confirmer ou non l'hypothèse nulle H0, tel que :

- **H0** : il n'y a pas d'association entre la voiture conduite et le niveau de revenu.
- **H1** : il existe une association entre la voiture conduite et le niveau de revenu.

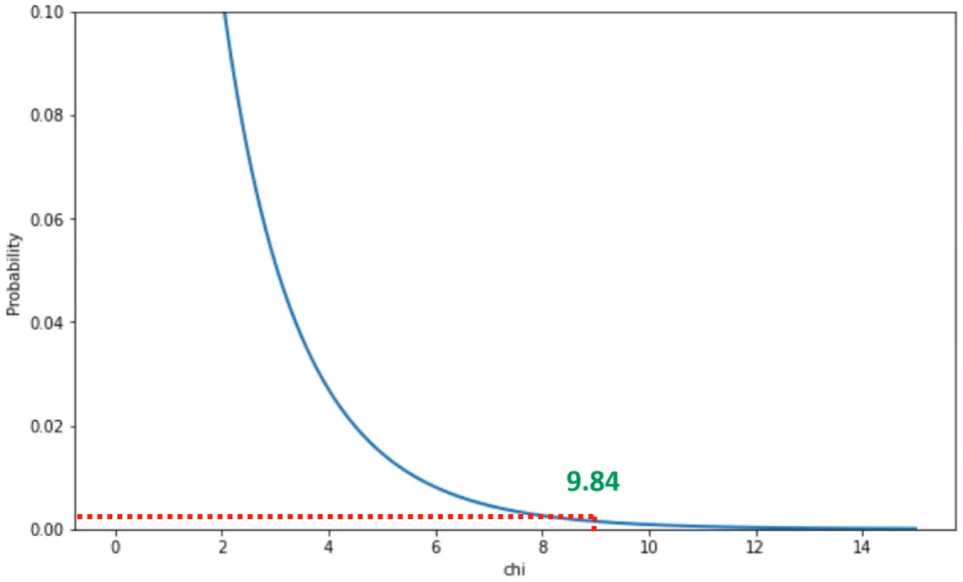
Khi-deux

2. Calculez la *p-value* pour chaque variable d'entrée:

La probabilité (*p-value*) que des personnes à faible revenu et des personnes à revenu moyen aient la même distribution de voitures économiques et de SUV est très faible.

En se fixant un seuil (généralement 0.05), Nous pouvons comparer la *p-value* trouvée avec ce seuil, et conclure s'il faut rejeter ou ne pas rejeter l'hypothèse nulle.

Nous pouvons facilement voir sur la figure à droite que la **probabilité trouvée** est inférieure à 0,05 et que, par conséquent, il existe une association entre le revenu d'une personne et la voiture qu'elle conduit (**H1 est vrai**).



3. Sélectionnez les caractéristiques ayant les *p-values* les moins élevées :

Les variables d'entrée dont la *p-value* est inférieure à un certain seuil sont considérées comme significatives et sont retenues pour le modèle de *machine learning* final.

Sélection de caractéristiques

Code python (1/2)

```python
import pandas as pd
from sklearn.feature_selection import SelectKBest, f_regression, mutual_info_regression, f_classif, mutual_info_classif, chi2
import numpy as np
from sklearn.preprocessing import OrdinalEncoder, LabelEncoder

df = pd.read_csv("sample_dataset.csv")
df = df.iloc[:,:11].dropna()                # dropna() permet de supprimer toutes les lignes avec des valeurs manquantes
X = df.iloc[:,:-1]                          # paramètres d'entrée
y = df.iloc[:,-1]                           # paramètre de sortie


''' Variables quantitatives et cible quantitative '''

# Matrice de corrélation
r = np.zeros((df.shape[1],df.shape[1]))
for i in range(df.shape[1]):
    for j in range(df.shape[1]):
        r[i,j] = np.corrcoef(df.iloc[:,i], df.iloc[:,j])[0,1]

# Sélectionner les 5 paramètres les plus corrélés avec la sortie.
selector = SelectKBest(f_regression, k = 5)
selector.fit(X,y)
X_tr = selector.transform(X)                 # les variables les plus corrélés
noms_var = X.columns[selector.get_support()]   # les noms des variables les plus corrélés

# Information mutuelle
selector = SelectKBest(mutual_info_regression, k = 5)
selector.fit(X,y)
X_s = selector.transform(X)
noms_var = X.columns[selector.get_support()]
I = selector.scores_
```

Sélection de caractéristiques

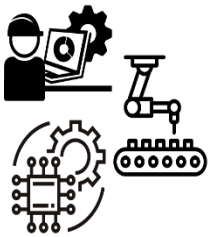
Code python (2/2)

```
32
33  ''' Variables qualitatives et cible quantitative '''
34  X = df.iloc[:,:3]
35  X = OrdinalEncoder().fit_transform(X).toarray()
36  selector = SelectKBest(lambda X,y : mutual_info_regression(X,y, discrete_features=True),k = 5)
37  X_tr = selector.fit_transform(X,y)
38  noms_var = X.columns[selector.get_support()]
39
40  ''' Variables quantitatives et cible qualitative '''
41  # ANOVA
42  selector = SelectKBest(f_classif, k = 5)
43  selector.fit(X,y)
44  X_tr = selector.transform(X)
45  noms_var = X.columns[selector.get_support()]
46  p_values = selector.pvalues_    # p_values
47
48  #Information mutuelle
49  selector = SelectKBest(mutual_info_classif, k = 5)
50  selector.fit(X,y)
51  X_tr = selector.transform(X)
52  noms_var = X.columns[selector.get_support()]
53
54  ''' Variables qualitatives et cible qualitative '''
55  X_tr = OrdinalEncoder().fit_transform(X)
56  y_tr = LabelEncoder().fit_transform(y)
57  selector = SelectKBest(chi2, k = 5)
58  X_tr = selector.fit_transform(X_tr,y_tr)
59  p_values = selector.pvalues_     # p_values
```

Comment améliorer votre production via la fouille de données ?

Data Science

Ce support de cours a été élaboré à partir des formations conçues dans le cadre de la chaire.



Chaire Arts et Métiers
de recherche industrielle
SYSTÈMES DE PRODUCTION
RECONFIGURABLES-SÛRS-PERFORMANTS

Auteurs :

Jean-Yves DANTAN
Laura WANG
Lazhar HOMRI
Augustin CROS-LE LAGADEC
Francois DELAVALLE
Suliac LEFREUVE
Tanguy COANET
Wahb ZOUHRI
Malek OURARI
Alain ETIENNE
Tanguy COLLEVILLE
Valentin CHERBONNIER

thyssenkrupp
UIMM F2i - FONDS INNOVATION LA FABRIQUE DE L'AVENIR
UIMM Lorraine LA FABRIQUE DE L'AVENIR
Grand Est ALSACE CHAMPAGNE-ARDENNE LORRAINE
UNION EUROPÉENNE Fonds Européen de Développement Régional