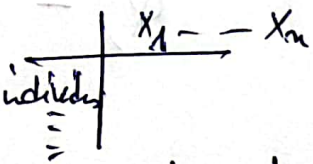
Preparation des données

I/ Collecte des données :

→ Données structurées : un tableau individus X Variable



→ Données non structurées : tout ce qui n'est pas organisé sous forme d'un tableau de données :

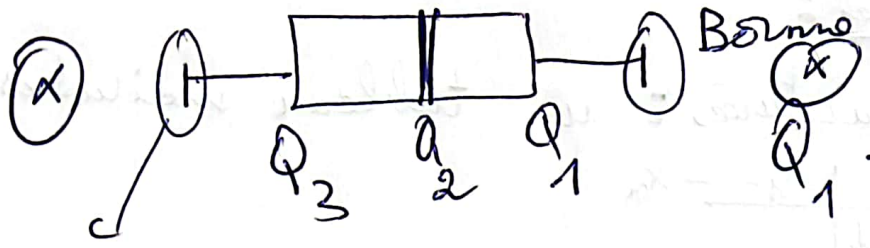
la messagerie, les images, les vidéos

II/ Nettoyage des données :

- détection des données manquantes, des données aberrants
  - détection des données manquantes
    - Résumé
      - summary
    - na.fail()
  - détection des données aberrants
    - moy de Tout
    - boîte à moustaches
- traitement des données aberrants et des données manquantes
  - traitement des données manquantes
    - calcul des tx des NA
      - sum(is.na(data))/ Prod(dim(data))
      - Si tx de NA < 5% on supprime les
      - Imputation simple
        - Generalized imputation
        - Similar case imputation
  - traitement des données aberrants

→ détection des valeurs aberrantes :

boîte à moustache



Borne inférieure

$Q_1 - 1,5 * (Q_3 - Q_1)$

Borne supérieure

$Q_3 + 1,5 * |Q_3 - Q_1|$

→ traitement des valeurs manquantes :

1 Generalized imputation : on remplace les données manquantes par : la moyenne, la médiane } variable quantitative

le mode variable qualitative

library(Hmisc)

> impute(data$, fun=mean)

2 Similar case imputation

- le plus proche voisin
  - ↓
  - library(VIM)
  - kNN(data [, c(" ", " ")])
- aléatoire
  - ↓
  - impute(data$, "random")

III Transformation des données

- encodage des variables catégorielles
  - encodage nominale
    - ~~Escot~~
    - → label encoding
    - → one hot encoding
  - encodage ordinal
    - → ordinal encoding
    - → target encoding
- Normalisation / Standardisation

scale(, center = min, scale =

$$X' = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad \text{ou} \quad X' = \frac{X_i - \mu}{\sigma}$$

Z-score

$$x' = \frac{x_i - \mu}{\sigma}$$

transformation des données

1 encodage des variables catégorielles

l'objectif de l'encodage: l'encodage est la conversion des variables catégorielles en format numérique.

Pourquoi encoder les variable catégorielle ?

- les algorithmes de machine learning nécessitent des entrées numériques
- Meilleure performance des modèles : Amélioration de la performance des modèles en facilitant leur capacité à identifier les relations dans les données

Méthodes d'encodage

- encodage nominal (Variable nominal)
  - One hot encoding
  - Target encoding (encodage par la moyenne)
- encodage ordinal (variable ordinal)
  - label encoding (encodage des étiquettes)
  - ordinal guidé par la cible

Exemples

* Encodage nominal : transforme chaque modalité en une colonne qui contient des valeurs binaire

ID	couleur
1	Rouge
2	Vert
3	Bleu
4	Rouge
5	Bleu

sous R

```
> data = data.frame (ID = 1:5,
couleur = c("Rouge", "Vert", "Bleu", "Rouge",
"Bleu" ))
> one_hot_encoded =
model.matrix (~ couleur - 1, data = data
```

Bleu	Rouge	Vert
0	1	0
0	0	1
1	0	0
0	1	0
1	0	0

(-1) retire la colonne (ID)

* Encodage ordinal : label encoding : transforme les catégories en valeurs numérique selon leur ordre

ID	Niveau
1	faible
2	Moyen
3	élevé
4	Moyen
5	faible
6	élevé

Sous R

```
> data = data.frame(ID = 1:6,
Niveau = c("Faible", "Moyen", "élevé",
"Moyen", "faible", "élevé"))
> data $ Niveau_encodé =
as.numeric(factor(data $ Niveau,
levels = c("Faible", "Moyen", "Elevé")))
```

ID	Niveau	Niveau_encodé
1	faible	1
2	Moyen	2
3	élevé	3
4	Moyen	2
5	faible	1
6	élevé	3

2 la normalisation et la standardisation :

la normalisation et la standardisation sont deux techniques pour transformer les valeurs des données afin de les rendre comparables et améliorer les performances des modèles d'apprentissage automatique.

→ **Normalisation** : la normalisation consiste à transformer les données afin qu'elles se situent dans une plage spécifique, généralement entre 0 et 1. elle est utilisée lorsque vous voulez mettre toutes les caractéristiques sur une même échelle.

Min - Max

$$X_{normalisé} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Sous R : scale( )

utilisée pour les algorithmes basés sur des distances tels que : les K-plus proches voisins (KNN) ou les réseaux neuronaux, où les différences d'échelle peuvent influencer les résultats.

→ **standardisation** : la standardisation consiste à transformer les données pour qu'elles aient une moyenne 0 et un écart-type 1, autrement dit pour que les données suivent une distribution normale centrée réduite.

Z - Score

$$X_{standardisé} = \frac{X - \mu}{\sigma}$$

utilisée avec des algorithmes qui supposent une distribution normale :

→ **Normalisation** : la normalisation consiste à transformer les données afin qu'elles se situent dans une plage spécifique, généralement entre 0 et 1. Elle est utilisée lorsque vous voulez mettre toutes les caractéristiques sur une même échelle.

Min - Max

$$X_{normalisé} = \frac{X - X_{min}}{\underbrace{X_{max} - X_{min}}_{\text{étendue}}}$$

Sous R : scale(data, center = min, scale = max-min)

utilisée pour les algorithmes basés sur des distances tels que : les k-plus proches voisins (KNN) ou les réseaux neuronaux, où les différences d'échelle peuvent influencer les résultats.

→ **standardisation** : la standardisation consiste à transformer les données pour qu'elles aient une moyenne 0 et un écart-type 1, autrement dit pour que les données suivent une distribution normale centrée réduite.

Z - Score

$$X_{standardisé} = \frac{X - \mu}{\sigma}$$

scale(data, center = mean, scale = sd)

utilisée avec des algorithmes qui supposent une distribution normale :