المدرسة الوطنية للعلوم التطبيقية - بني ملال

ⵜⴰⵙⴷⴰⵡⵜ ⵜⴰⵏⴰⵎⵓⵔⵜ ⵏ ⵜⵙⵙⵏⴰⵡⵉⵏ ⵜⵉⵜⵜⴰⵡⵉⵏ - ⴱⵏⵉ ⵎⵍⴰⵍ

Ecole Nationale des Sciences Appliquées - Béni Mellal

ANALYSE DES DONNÉES

Filière : Transformation Digitale Industrielle (TDI)

Pr. ESSWIDI AYOUB

A.U. 2024-2025

PLAN PRÉVISIONNEL ET OBJECTIFS

- Analyse de données
  - Prétraitement et nettoyage des données
  - Visualisation des données (EDA).
  - Feature Selection, Feature engineering, …
  - Analyse des données multimédia, Analyse du Web et des réseaux sociaux
- Science de données et Data Mining
  - Introduction et types d'apprentissage
  - Algorithmes de ML
  - Algorithmes de Deep Learning
  - Exemple d'application en IA générative
- Big Data
  - Introduction au Big Data: définition, caractéristiques, Historique, domaine d'application.
  - Architectures du Big Data
  - Ecosystèmes : Hadoop et Map-Reduce, Apache spark, kafka, HBASE, …

PRÉTRAITEMENT ET NETTOYAGE DES DONNÉES

- Échantillonnage
- Nettoyage des données
  - Gestion des données manquantes
  - Gestion des données aberrantes
  - Gestion des doublons
  - Gestion des données Incohérentes
- Gestion des données catégorielles
- Discrétisation des données
- Normalisation et standardisation des données
- Gestion les données déséquilibrées
- Gestion Des Attributs Temporels

3

DATA PREPROCESSING (PRÉPARATION DES DONNÉES)

- Le prétraitement des données comprend plusieurs activités essentielles avant l'analyse, telles que le nettoyage des données, …etc. Ces étapes permettent d'assurer la qualité et la cohérence des données utilisées pour des modèles d'analyse ou de ML.

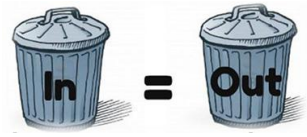
- Nettoyage des données est un sous-ensemble de prétraitement, principalement concerné par **l'identification** et **la correction** des erreurs et des incohérences au sein de l'ensemble de données.

- La préparation des données englobe l'ensemble du processus depuis le moment où elles sont collectées jusqu'au moment où elles sont introduites dans les outils analytiques.
  - la préparation des données comprend la collecte, l'intégration le nettoyage et l'ingénierie des données

4

QUALITÉ DES DONNÉES

« Garbage in → Garbage out »



La qualité + La fiabilité = les analyses, modèles et décisions basées sur ces données soient précis et pertinents

5

QUALITÉ DES DONNÉES

Dimensions de la qualité des données

6

QUALITÉ DES DONNÉES

Dimensions de la qualité des données

- **Exactitude (Accuracy) :**
  - Les données reflètent-elles correctement la réalité ?
  - **Exemple** : Les adresses des clients dans une base de données sont-elles à jour et correctes ?

- **Complétude (Completeness) :**
  - Les données contiennent-elles toutes les informations nécessaires ?
  - **Exemple** : Une fiche client sans numéro de téléphone est incomplète.

- **Cohérence (Consistency) :**
  - Les données sont-elles uniformes entre les différentes sources ?
  - **Exemple** : Les données d'un client dans le CRM et dans la base comptable doivent correspondre.

7

QUALITÉ DES DONNÉES

Dimensions de la qualité des données

- **Unicité (uniqueness) :**
  - Certaines informations doivent être enregistrées de manière unique.
  - **Exemple**: le Numéro de téléphone d'un client ne peut pas être enregistrée deux fois dans la base de données avec des ID client différents.

- **Actualité (Timeliness) :**
  - Les données sont-elles mises à jour régulièrement ?
  - **Exemple** : Les prix des produits sur un site e-commerce doivent être actualisés pour refléter les promotions.

- **Validité (conformité) :**
  - Les données doivent respecter les règles et contraintes définies pour leur format ou leur contenu.
  - **Exemple** : un champ "âge" accepte des valeurs négatives ou supérieures à 120 ans.

8

QUALITÉ DES DONNÉES

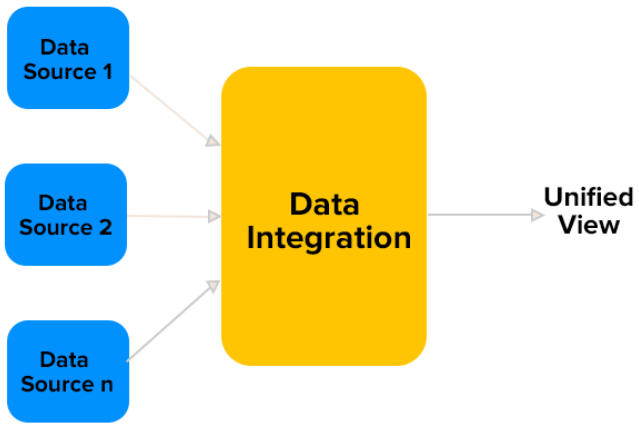
Dimensions de la qualité des données

- **Pertinence (Relevance) :**
  - Les données sont-elles adaptées aux objectifs de l'analyse ?
  - **Exemple** : Collecter des données sur les habitudes alimentaires pour prédire les ventes d'équipements sportifs pourrait être non pertinent.

- **Accessibilité (Accessibility) :**
  - Les données sont-elles facilement accessibles pour ceux qui en ont besoin ?
  - **Exemple** : Une base de données protégée par un système d'accès complexe peut limiter son utilisation.

9

L'INTÉGRATION DES DONNÉES



10

L'INTÉGRATION DES DONNÉES

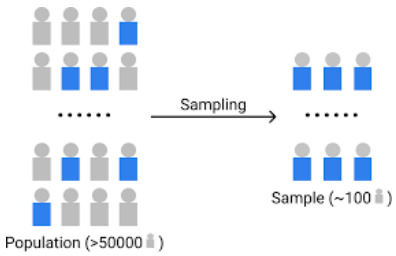
- L'intégration des données est le processus de combinaison de données provenant de plusieurs sources pour fournir aux organisations une vue unifiée pour :
  - des informations améliorées,
  - une prise de décision éclairée
  - une compréhension cohérente de leurs opérations commerciales.

- Problèmes :
  - Sources différentes => schémas de BD différents
  - Identifier les entités du monde réel à partir de multiples sources de données, par exemple, Beni Mellal = Bni Mellal
  - Conflits de valeurs de données (l'heure dans différents serveurs par exemple, les métriques et les unités de mesures)

11

ÉCHANTILLONNAGE

- Il est souvent difficile de collecter suffisamment de données pour un projet d'apprentissage automatique. Cependant, parfois il y a trop de données, et vous devez sélectionner un sous-ensemble d'exemples pour l'entrainement d'un modèle



- En fin de compte, la réponse dépend du problème: que nous voulons prédire et quelles fonctionnalités voulons-nous?

12

ÉCHANTILLONNAGE (SIMPLING)

TYPES d'Échantillonnage

- **Échantillonnage aléatoire simple :** Il y a une probabilité égale de sélectionner un élément particulier
- **Échantillonnage sans remplacement :** Au fur et à mesure que chaque élément est sélectionné, il est retiré de la population
- **En échantillonnage avec remise**, le même objet peut être ramassé plusieurs fois
- **Échantillonnage stratifié :** Divisez les données en plusieurs partitions ; puis tirer des échantillons aléatoires de chaque partition

13

NETTOYAGE DES DONNÉES

Gérer les valeurs manquantes

Id Client	Code Produit	Sexe	Revenu	Âge	Marié	Montant
1001	75000	M	75000	C	M	5000
1002	4000	F	-40000	40	V	4000
1003	92100		1000000	45	V	7000
1004	6260	M	50000	0	C	1000
1005	29000	F	99999	30	D	3000

Y a-t-il des valeurs manquantes dans ce tableau ou autres problèmes ?
data.isna().sum()

14

NETTOYAGE DES DONNÉES

Gérer les valeurs manquantes

Id Client	CP	Sexe	Revenu	Âge	Marié	Montant
1001	75000	M	75000	C	M	5000
1002	4000	F	-40000	40	V	4000
1003	92100		1000000	45	V	7000
1004	6260	M	50000	0	C	1000
1005	29000	F	99999	30	D	3000

- CP : 4000 ? 6260 ?
- Sexe : il y a un champ manquant.
- Revenu : -40000 ? Négatif ! 1000000 ? C'est beaucoup. 99999 (c'est très précis), unité
- Age : C ? 0 ?
- Marié : que signifient les symboles ?
- Montant : le problème de la monnaie.

15

NETTOYAGE DES DONNÉES

Gérer les valeurs manquantes

solutions possibles :

- Ignorer les tuples : opération faite d'habitude quand l'étiquette de classe manque
- Remplir les valeurs manquantes manuellement : opération ennuyeuse et infaisable
- Employer une constante globale pour remplir les valeurs manquantes : par exemple, "inconnu", ce qui engendre la création d'une nouvelle classe
- Remplacer les valeurs numériques par une constante par exemple 0 en utilisant les fonctions **replace() ou fillna()** de Pandas
- Utiliser une valeur statistique (e.g., moyenne) pour remplir les valeurs manquantes.

16

NETTOYAGE DES DONNÉES

Gérer les valeurs manquantes

Solutions possibles :

- Dans le cas des données numériques non continues, remplacer toute donnée manquante par le mode de la distribution statistique (la valeur la plus fréquente) de l'attribut concerné, si ce mode existe.
- On peut également chercher à estimer ces valeurs manquantes par des méthodes d'induction comme des modèles de ML.
- Remplissez avec la valeur la plus probable. Pour ce faire, utilisez une technique d'inférence, par exemple bayésien ou arbre décision

17

NETTOYAGE DES DONNÉES

Gérer les valeurs manquantes

Cas d'utilisation avec la bibliothèque scikit-klearn:

❑ **SimpleImputer:** remplace les valeurs manquantes par des valeurs statistiques

- Missing_values : np.nan | -1 | 999 | inconnue |…
- Strategy : mean | median | most_frequence | constant

```
From sklearn.impute import SimpleImpute

imputer = SimpleImputer(
      missing_values = np.nan,
      strategy = 'mean'
)

Imputer.fit_transform(Data)
```

❑ **KNNImputer:** remplace les valeurs manquantes par des valeurs des plus proches voisins

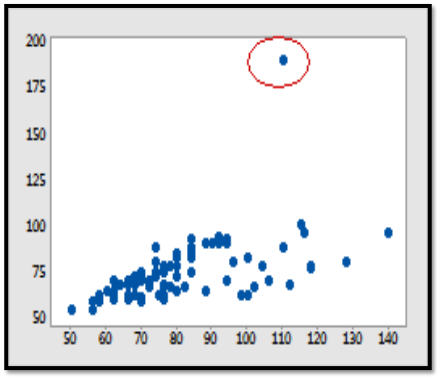
❑ Pour plus d'information https://scikit-learn.org/1.6/modules/impute.html

18

NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

- Les valeurs aberrantes sont des points de données qui diffèrent considérablement des autres observations dans l'ensemble de données et peuvent fausser l'analyse statistique ou les modèles de ML.

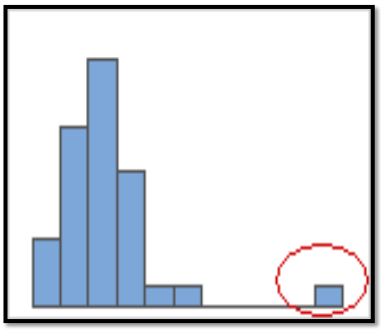


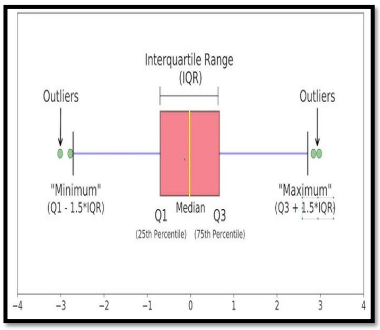
19

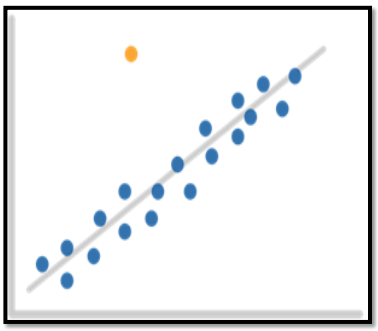
NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

- Méthodes de détection des valeurs aberrantes
  - méthodes graphiques :







  - Techniques de filtrage basées sur des seuils statistiques pour supprimer / remplacer les valeurs aberrantes.

20

NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

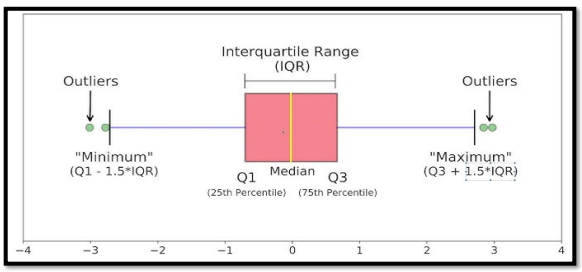
- Méthodes de détection des valeurs aberrantes
  - Clustering : détecter les exceptions
  - Par partitionnement (binning)
    - Trier et partitionner les données
    - Lisser les partitions par la moyenne, la médiane, les bornes, …
  - Inspection humaine et informatique combinée : détection des valeurs suspectes a l'aide d'une vérification humaine

21

NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

- Méthodes de détection des valeurs aberrantes
  - Calculez des statistiques récapitulatives comme la moyenne, l'écart-type, les quartiles et l'intervalle interquartile (IQR).
  - Les valeurs aberrantes sont souvent définies comme des points de données inférieurs à $Q_1 - 1{,}5 * IQR$ ou supérieurs à $Q_3 + 1{,}5 * IQR$.

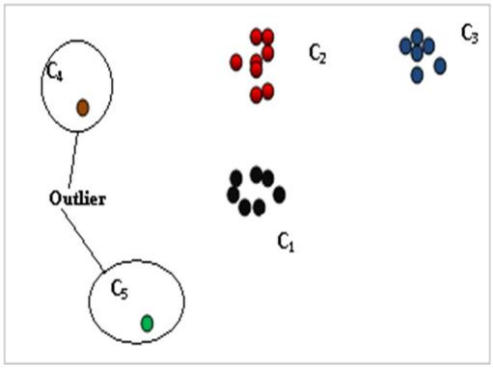


22

NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

- Méthodes de détection des valeurs aberrantes
  - Clustering : détecter les exceptions



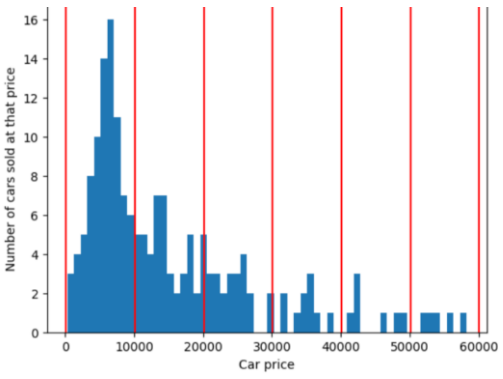
23

NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

- Méthodes de détection des valeurs aberrantes
- Par partitionnement (binning):
- Partitionnement à largeur égale (distance) :
  - on divise la plage en N intervalles de taille égale
  - si A et B sont les valeurs les plus basses et les plus élevées de l'attribut, la largeur des intervalles sera :

$$W = \frac{B-A}{N}.$$

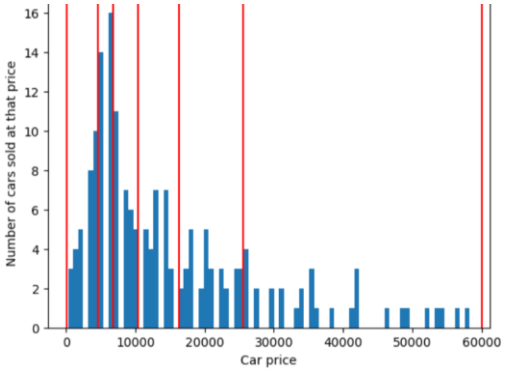


24

NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

- Méthodes de détection des valeurs aberrantes
- Par partitionnement (binning):
- Partitionnement à profondeur égale (fréquence) :
  - On divise la plage en N intervalles, chacun contenant approximativement le même nombre d'échantillons



25

NETTOYAGE DES DONNÉES

Gestion des valeurs aberrantes

- Méthodes de détection des valeurs aberrantes
- Solutions :
  - ✓ On supprime l'individu.
    - Le risque est de supprimer par la même occasion des valeurs utiles. Si la population est nombreuse, ce risque devient très faible.
  - ✓ On remplace la valeur aberrante par une autre valeur : la valeur statistique, une valeur prise au hasard dans la distribution des valeurs.
    - Le risque est de produire une incohérence par rapport aux autres valeurs (aberration croisée).
  - ✓ Imputation

26

DATA PREPROCESSING (PRÉTRAITEMENT DES DONNÉES)

Gestion des données dupliquées

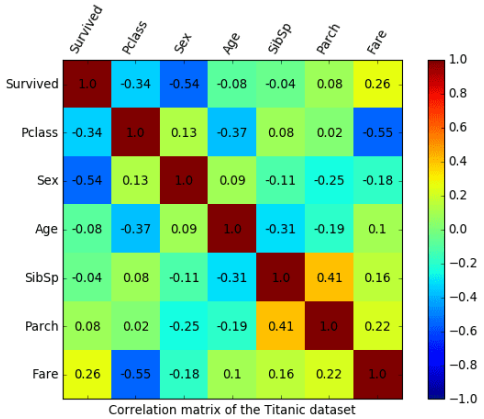
- Utilisez des fonctions comme **duplicated()** dans pandas pour identifier les lignes en double en fonction de colonnes spécifiques ou de la ligne entière.
- Si les enregistrements en double sont redondants et ne fournissent aucune information supplémentaire, vous pouvez les supprimer à l'aide de la fonction **drop_duplicates()** de pandas ou des méthodes similaires dans d'autres bibliothèques.
- Dans certains cas, des doublons peuvent se produire en raison de plusieurs entrées, mais ont des identifiants uniques. Assurez-vous de conserver des identifiants uniques ou des colonnes clés qui distinguent les enregistrements en double.

27

DATA PREPROCESSING (PRÉTRAITEMENT DES DONNÉES)

Gestion des données dupliquées

- **Les attributs redondants** peuvent être détectés par analyse de corrélation et d'analyse de covariance **df.corr()**



Correlation matrix of the Titanic dataset

28

DATA PREPROCESSING (PRÉTRAITEMENT DES DONNÉES)

DONNÉES INCOHÉRENTES

❑ Exemple :
Les valeurs d'une colonnes ville dans une Dataset sont:
Rabat, Beni mellal, Casa, bni mellal, bm, casablanca, Beni-mellal, rabat, ….

❑ Solutions :

✓ Correction manuelle à l'aide de références externes

✓ Semi-automatique à l'aide de divers outils Pour détecter la violation des dépendances fonctionnelles et des contraintes de données connues

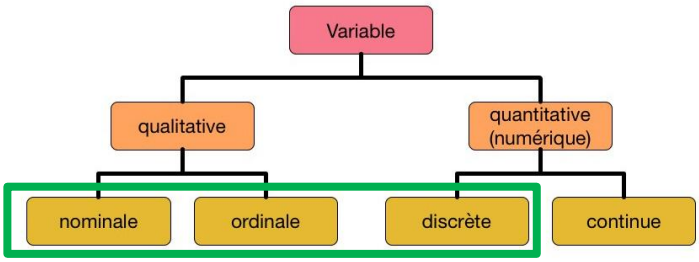
✓ Calcule de similarité

29

GESTION DES DONNÉES CATÉGORIELLES

❑ Les variables catégorielles représentent des groupes ou des catégories et sont souvent de nature non numérique, ce qui pose des difficultés pendant l'entraînement du modèle, notamment :

- Représentation non numérique
- Variables ordinales et nominales

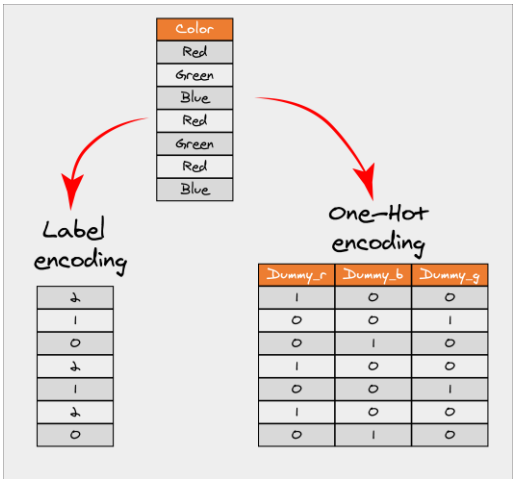


30

GESTION DES DONNÉES CATÉGORIELLES

Traitement

- L'encodage **one-hot** et l'encodage **ordinal** sont deux techniques utilisées pour traiter les données catégorielles.
- **L'encodage one-hot** transforme chaque catégorie en une colonne binaire distincte.
- **l'encodage ordinal** attribue un ordre numérique aux catégories, ce qui est utile lorsque l'ordre des données est pertinent.



31

GESTION DES DONNÉES CATÉGORIELLES

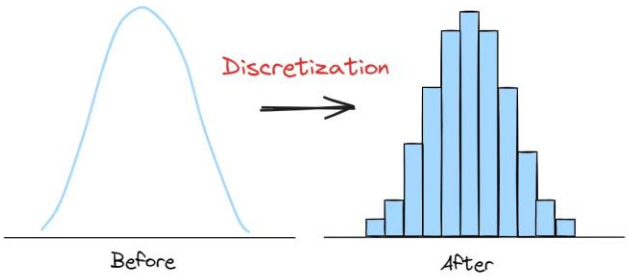
Exemple

```
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder, OneHotEncoder

# méthode 1
encoder = LabelEncoder()
encoder.fit_transform(y)

# méthode 2
encoder = OrdinalEncoder()
encoder.fit_transform(X)

# méthode 3
encoder = OneHotEncoder()
encoder.fit_transform(X)
```

32

DISCRÉTISATION DE DONNÉE

- La discrétisation des données consiste à transformer des données continues en catégories ou intervalles discrets
- Réduire le nombre de valeurs pour un attribut continu donné en partitionnant la plage de l'attribut en intervalles,
- Les étiquettes d'intervalle remplacent les valeurs d'attribut réelles



33

DISCRÉTISATION DE DONNÉE

- Méthodes:
  - Discrétisation (Binning)
  - Analyse de clusters
  - Discrétisation basée sur les technique statistique (l'entropie, khi2,….)

34

DISCRÉTISATION DE DONNÉE

Techniques de discrétisation

1. **Discrétisation égale en largeur (Equal-width binning) :**
   Divise l'intervalle de la variable en intervalles de largeur égale.
   Exemple : Si les valeurs vont de 0 à 100 et qu'on crée 5 catégories, chaque intervalle sera de 20 unités : [0-20], [21-40], ...

2. **Discrétisation égale en fréquence (Equal-frequency binning) :**
   Chaque intervalle contient un nombre égal d'exemples.
   Exemple : Si on a 100 données et 5 catégories, chaque catégorie contiendra 20 données.

3. **Basée sur des seuils prédéfinis :**
   Utilise des seuils spécifiques définis par l'utilisateur ou basés sur la connaissance métier.
   Exemple : Catégoriser les âges en [0-18] (enfants), [19-60] (adultes), [60+] (seniors).

35

DISCRÉTISATION DE DONNÉE

Techniques de discrétisation

4. **Discrétisation par clustering :**
   Utilise des algorithmes comme K-means pour regrouper des données similaires en catégories.

5. **Entropie minimale (MDLP - Minimum Description Length Principle) :**
   Utilise des méthodes basées sur l'entropie pour diviser les données là où la distribution des classes change significativement.

36

DISCRÉTISATION DE DONNÉE

Techniques de discrétisation

Exemples:

```python
import numpy as np
import pandas as pd

# Exemple de données
data = {'Valeurs': [1.2, 2.5, 3.8, 4.1, 5.7, 6.9, 8.3, 9.0, 10.5]}
df = pd.DataFrame(data)

# Discrétisation égale en largeur (3 intervalles)
df['Discret_Equal_Width'] = pd.cut(df['Valeurs'], bins=3, labels=['Bas', 'Moyen', 'Haut'])

# Discrétisation égale en fréquence (3 intervalles)
df['Discret_Equal_Frequency'] = pd.qcut(df['Valeurs'], q=3, labels=['Bas', 'Moyen', 'Haut'])

print(df)
```

```
   Valeurs Discret_Equal_Width Discret_Equal_Frequency
0      1.2                Bas                      Bas
1      2.5                Bas                      Bas
2      3.8                Bas                      Bas
3      4.1                Bas                    Moyen
4      5.7              Moyen                    Moyen
5      6.9              Moyen                    Moyen
6      8.3               Haut                     Haut
7      9.0               Haut                     Haut
8     10.5               Haut                     Haut
```

37

NORMALISATION ET STANDARDISATION DES DONNÉES

Normalisation

- La mise à l'échelle des fonctionnalités est une méthode utilisée pour normaliser la plage de variables indépendantes ou de fonctionnalités de données.
- La normalisation signifie généralement que les valeurs sont redimensionnées dans une plage de [0,1].

Min-Max	$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$
Max	$x' = \frac{x}{x_{max}}$
Decimal	$x' = \frac{x}{10^j}$
Sigmoid	$x' = \frac{1}{1 + e^{-a}}, \forall a = \frac{x - x_{min}}{x_{std}}$
Softmax	$x' = \frac{1 - e^{-a}}{1 + e^{-a}}, \forall a = \frac{x - x_{min}}{x_{std}}$

38

NORMALISATION ET STANDARDISATION DES DONNÉES

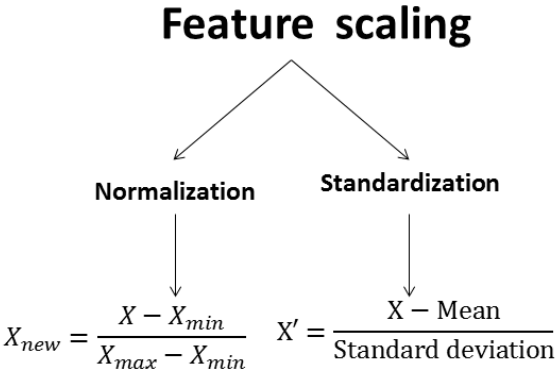
Standardisation

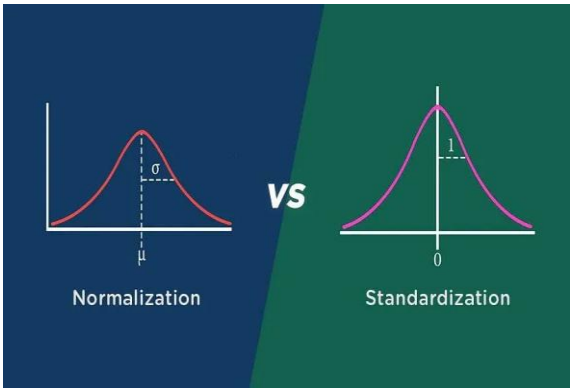
- La standardisation transforme les valeurs d'une variable pour qu'elles aient une moyenne $\mu = 0$ et un écart-type $\sigma = 1$.
- La standardisation ne fixe pas de plage spécifique pour les valeurs transformées..

39

NORMALISATION ET STANDARDISATION DES DONNÉES

Normalisation VS Standardisation

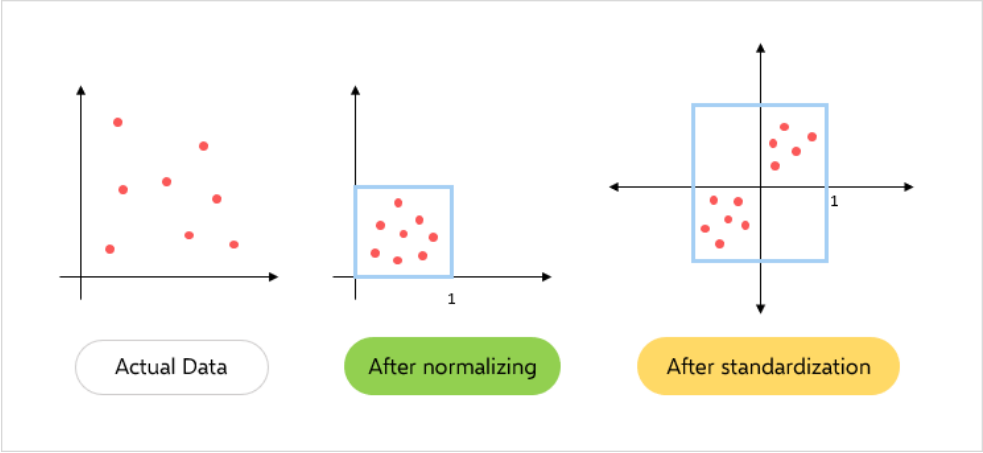




40

NORMALISATION ET STANDARDISATION DES DONNÉES

Normalisation VS Standardisation



41

NORMALISATION ET STANDARDISATION DES DONNÉES

Normalisation VS Standardisation

from sklearn.preprocessing **import** MinMaxScaler, StandardScaler,RobustScaler	
X_minmax = MinMaxScaler().fit_transform(X)	$X_{minmax} = \frac{X - X_{min}}{X_{max} - X_{min}}$
X_stdscl = StandardScaler().fit_transform(X)	$X_{stdscl} = \frac{X - \mu}{\sigma}$
X_robust = RobustScaler().fit_transform(X)	$X_{robust} = \frac{X - median}{IQR}$ où $IQR = interquartile\ range = Q_3 - Q_1$

42

GESTION DES DONNÉES TEMPORELLE

date/temps

- Un fichier CSV est importé et qu'un cadre de données est créé, les objets Date/Heure du fichier sont lus comme un objet chaîne (String) plutôt que comme un objet Date/Heure.

- Vérifiez le format des dates (par exemple :YYYY-MM-DD, DD/MM/YYYY, etc.).
- Identifiez si les données incluent également l'heure (HH:MM:SS).
- Déterminez si les données sont au format string ou déjà converties en un type de donnée spécifique comme datetime
  (en Python, cela peut être pd.Timestamp ou datetime.datetime).

43

GESTION DES DONNÉES TEMPORELLE

Date/time

```
import pandas as pd

df['date_col'] = pd.to_datetime(df['date_col'],
                                format='%Y-%m-%d %H:%M:%S',
                                errors='coerce')
```

- Le paramètre **format** permet de spécifier le format exact des données.
- L'option **errors**='coerce' convertit les valeurs non valides en NaT (Not a Time).

44

GESTION DES DONNÉES TEMPORELLE

Extraction de caractéristiques temporelles

- df['year'] = df['date_col'].dt.year
- df['month'] = df['date_col'].dt.month
- df['day'] = df['date_col'].dt.day
- df['weekday'] = df['date_col'].dt.weekday # 0 = Lundi, 6 = Dimanche
- df['hour'] = df['date_col'].dt.hour
- df['is_weekend'] = df['weekday'].isin([5, 6]).astype(int) # 1 si c'est un week-end

45

Pandas Cheatsheet

Imports

```
import pandas as pd
import altair as alt
import datetime
```

Creating DataFrames

```
df1 = pd.read_csv(  # From file
    'world_countries.csv')
df2 = pd.DataFrame({  # From Python dict
  'col0': [0, 1, 2],  'col1': [3, 4, 5],
  'col2': ['ab', 'cd', 'ef'],
  'col3': [datetime.datetime.now()] * 3})
```

Inspecting DataFrames

```
df1.head()      # First 5 rows
df1.tail()      # Last 5 rows
df1.columns     # Columns names
len(df1)        # Number of rows
df1.shape       # Number of rows and cols
df1.describe()  # Stats about each column
df1.info()      # Summary info
```

Summarizing columns

```
# Rename a column
df1 = df1.rename(
  columns={'Population': 'Pop'})
df1.Pop.sum()     # Sum
df1.Pop.mean()    # Average
df1.Pop.std()     # Standard deviation
df1.Pop.median()  # Median
df1.Pop.min()     # Minimum
df1.Pop.max()     # Maximum
```

Filtering rows

```
df1[5:11]      # Select rows 5 through 10
# Rows with Spain in the Country column
df1[df1.Country == "Spain"]
# Removing nulls
df1 = df1[~df1.Pop.isnull()]
# Convert strings to integers
df1.ConSal = df1.Pop.astype('int64')
# Booleans operators are &, | and ~
df1[(df1.Pop > 100) &
    ~(df1.Area.isnull())]
```

Column manipulations

```
# Arithmetic operations on columns
df2['col0'] + df2['col1']
# Even if they're strings
df2['col2'] + df2['col2']
# Create new column from the result
df2['col4'] = df2['col0'] / df2['col1']
# String methods and attributes can be
# accessed via .str.
df2['col2'].str.replace('a', 'b')
# And datetime methods and attributes
# via .dt.
df2.col3.dt.date
# Select just some columns from DataFrame
df1[['Country', 'Pop']]
```

Dealing with missing values

```
# Drop rows with any missing values
df1.dropna()
# Drop columns with any missing value
df1.dropna(axis=1)
# Fill missing values with 0s
df1.fillna(0)
# Fill missing values with ''
df1.fillna('')
```

Grouping

```
# Get the average salary for each country
df1.groupby('Country').agg(
    {'Pop': 'mean'})
# Get the average and minimum salary
df1.groupby('Country').agg(
    {'Pop': ['mean', 'min']})
# Keep grouping column as a column
df1.groupby(
    'Country', as_index=False).agg(
    {'Pop': ['mean', 'min']})
```

Miscellaneous

```
# Reorder from top salary to lowest
df1.sort_values('Pop',
                ascending=False)
# Remove a column
df1.drop(columns='Phones')
# Randomly select a sample of 45 rows
df1 = df1.sample(45)
```

Merging

```
df3 = pd.DataFrame(
    {'col5': ['ab', 'cd', 'ef'],
     'col6': [100, 200, 300]})
# Create a new DataFrame matching col2
# of df2 and col5 of df3.
df2.merge(df3, left_on='col2',
          right_on='col5')
```

Graphing

```
alt.Chart(df1).mark_point().encode(
  x='Country', y='Area', size='Pop',
  color='Birthrate')
```

46