COURS DE *DATA SCIENCE*

2è Partie: Préparation et prétraitement des données

Dr. Pegdwendé Nicolas Sawadogo

sawadogonicholas44@gmail.com

PROGRAMME DE LA SÉANCE 2

1. Enjeux des traitements préliminaires
2. Exploration des données
3. Nettoyage et préparation des données

ENJEUX DES TRAITEMENTS PRELIMINAIRES

SECTION 1

UN DELUGE DE DONNÉES

1. Avant de concevoir un modèle de *machine learning,* le *data scientist* doit choisir un modèle les données qui conviennent.

2. Mais avec la diversité de données, on s'y perd presque (Souvenez-vous: V comme variété)
   - Quelles données choisir?
   - Quels prétraitements réaliser pour rendre les données adéquates?
   - Quelles données complémentaires rechercher?

UNE DIVERSITÉ DES SOURCES DE DONNÉES

Avec la tendance *big data,* de plus en plus de sources de données sont accessibles:

- Systèmes d'information de production
  - Les données de transactions de ventes et d'achats de produits
  - Les données des interactions du service client
- Les entrepôts de données comportementales
  - Les fichiers logs, qui tracent les interactions entre utilisateurs et outils numériques
  - Les données issues des capteurs et objets connectés
  - Les données issues des réseaux sociaux

UNE DIVERSITÉ DES SOURCES DE DONNÉES

Avec la tendance *big data,* de plus en plus de sources de données sont accessibles:

- Les données géographiques
  - Des informations de géolocalisation sur des points d'intérêt
  - Les données météorologiques
  - Les données socio-économique: revenu par habitant, trafic routier, etc.
- Les données ouvertes (*open data*)
  - Des données gouvernementales d'intérêt publique
  - Par exemple, évolution des prix des céréales

UNE DIVERSITÉ DE FORMATS

Le data scientist doit travailler avec une diversité de formats de données, liée à la diversité de sources de données

- Fichiers classiques
  - Fichiers textuels standards (CSV, TXT, TSV)
  - Fichiers aux formats objets (JSON, XML)
  - Fichiers d'info géographiques (Shapefile)
  - Fichiers multimédias (images, vidéos, sons)
- Données de bases de données
  - Bases de données relationnelles (Oracle, PostgreSQL, MySQL, SQL Server)
  - Bases de données NoSQL (Neo4J, MongoDB, Cassandra, Hbase, ElasticSearch)

UNE DIVERSITÉ DE QUALITÉ

La qualité d'un modèle dépend de celle des données utilisées. Les critères de qualité incluent:

- L'exhaustivité
  - Moins il y'a de données manquantes, mieux c'est
  - Il faut (idéalement) à la fois avoir toutes les données possibles sur tous les individus possibles
- La granularité
  - C'est le degré de finesse des données (spatiales, temporelles)
  - Par exemple, pour des analyses par pays, il ne faut pas inclure des données au niveau continent.

UNE DIVERSITÉ DE QUALITÉ

La qualité d'un modèle dépend de celle des données utilisées. Les critères de qualité incluent:

- L'exactitude
  - Il faut veiller à ce que les données utilisées soient exactes
  - En traquant les incohérences, valeurs aberrantes, etc.
- La fraicheur
  - Vérifier que les données sont toujours d'actualité
  - Définir une politique de mise à jour des données, et s'y tenir

SOLUTIONS : EXPLORATION DES DONNÉES

Cela permet au *data scientist* de se préparer lui-même en ayant un bon aperçu des données.

- Utiliser les statistiques descriptives
  - C'est une arme simple, mais efficace pour connaitre et comprendre ses données
  - Moyenne, médiane, mode, quantiles, quartiles, ecart-type, etc.
  - Tableaux croisés dynamiques
- Visualiser les données
  - Construire des graphiques permettant d'avoir une idée du comportement des données
  - Identification aisée des valeurs extrêmes/manquantes
  - Aperçu de potentielles corrélations entre des variables
  - Boxplot, histogrammes, barcharts, etc.

SOLUTIONS : PREPARATION DES DONNÉES

Il s'agit de constituer un jeu de données de qualité, et homogène.

- Nettoyer les données
  - Pour chaque colonne, dépendant du type de données attendu, on va corriger certaines valeurs, ou supprimer.
  - La suppression peut concerner la cellule, ou l'individu
  - On peut aussi remplacer la valeur par la moyenne, la médiane, ou la valeur la plus courante.
- Transformer les données
  - Créer de nouvelles variable à partir d'autres
  - Agréger des valeurs: Master = M1 + M2; Licence = L1 + L2

SOLUTIONS : PREPARATION DES DONNÉES

Il s'agit de constituer un jeu de données de qualité, et homogène.

- Enrichir les données
  - Joindre de nouvelles sources de données
  - Ajouter des géocodages pour représenter une adresse géographique ou une adresse IP
  - Associer des informations juridiques à des entreprises
  - Associer des informations sur les jours fériés
  - Ajouter des données météorologiques

OUTILS DE PREPARATION DES DONNÉES

- Tableurs
  - Ce sont des outils essentiellement graphiques qui permettent de façon intuitive de réaliser des transformations
  - Ils sont généralement bien connus des utilisateurs: Excel
  - Nécessitent une MAJ manuelle, et limité en cas de gros volumes
- Outils ETL
  - Extract-Transform-Load: conçus à la base pour le entrepôts de données
  - Ils utilisent des workflows d'exécution, assurant une traçabilité
- Programmation
  - Utilisation de scripts personnalisés : R, Python, Java
  - Supporte plus ou moins de gros volumes de données

N° de la question : 926

PETIT POINT D'ETAPE…

La visualisation des données (avant la construction d'un modèle) peut être assimilée à:

A. L'exploration de données

B. La préparation de données

https://toreply.univ-lille.fr/

PROGRAMME DE LA SÉANCE 2

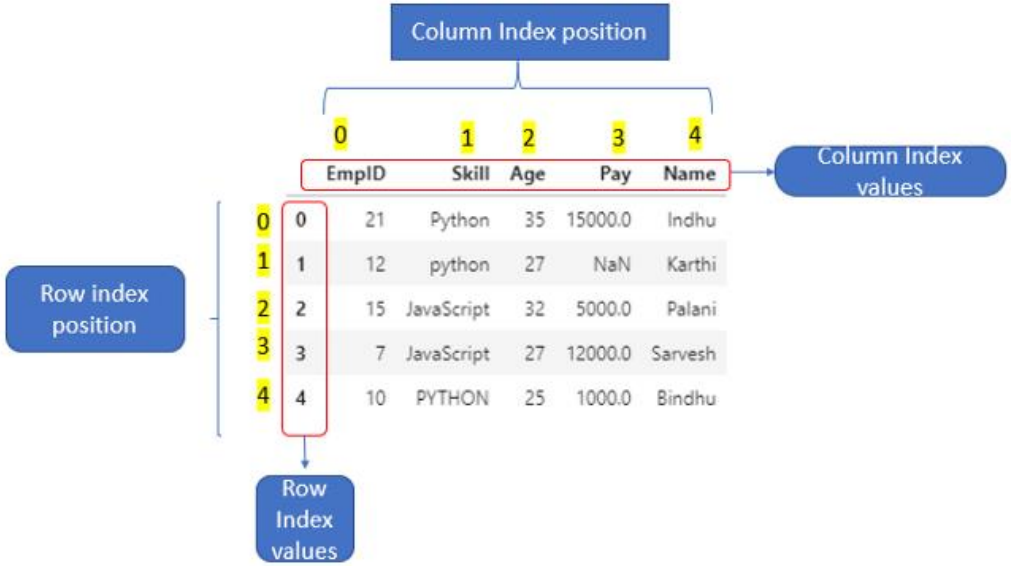
1. Enjeux des traitements préliminaires
2. Exploration des données
3. Nettoyage et préparation des données

EXPLORATION DES DONNEES SOUS PYTHON

SECTION 2

CHARGEMENT DES DONNÉES

- Dataframe
  - Ce sont des tableaux qui permettent de manipuler des données
  - Sous Python, ils sont gérés grâce au package pandas.



CHARGEMENT DES DONNÉES

- Dataframe
  - Ce sont des tableaux qui permettent de manipuler des données
  - Sous Python, ils sont gérés grâce au package pandas.

- Chargement à partir d'un fichier excel

```
df = pd.read_excel("fichier.xlsx", "feuil1", ...)
```

- Chargement à partir d'un fichier csv

```
df = pd.read_csv("fichier.csv", ";", ...)
```

DESCRIPTION DES DONNÉES

Statistiques descriptives à partir d'un dataframe

- Moyenne

```
df.mean(axis=0) #Moyennes en lignes
df.mean(axis=1) #Moyennes en colonnes
```

C'est le même principe pour la variance, l'écart type, etc.

- Mode

```
df.mode(axis=0) #Mode en lignes
df.mode(axis=1) #Mode en colonnes
```

- Valeurs fréquentes

```
df['var1'].value_counts()
```

DESCRIPTION DES DONNÉES

Statistiques descriptives à partir d'un dataframe

- Aperçu des données

```
df.head() #Premières lignes
df.tail() #Dernières lignes
```

- Description générale du dataframe

```
df.dtypes #Types de données
df.describe() #Distribution des données
```

DESCRIPTION DES DONNÉES

Statistiques descriptives à partir d'un dataframe

- Tableau croisé simple

```
pd.crosstab(df['TABAC'], df['RONFLE'])
```

RONFLE	non	oui
TABAC		
0	44	20
1	21	15

- Tableau croisé en pourcentages

```
pd.crosstab(df['TABAC'], df['RONFLE'], normalize=True)
```

RONFLE	non	oui
TABAC		
0	0.44	0.20
1	0.21	0.15

GROUPEMENTS ET TRIS

- Groupement avec agrégation simple

```
df.groupby('SEXE')['TAILLE'].mean()
```

```
SEXE
0    181.253333
1    180.640000
```

- Groupement avec agrégations multiples

```
df.groupby('SEXE').agg({'AGE':'max', 'POIDS':'mean',
'TABAC':'count'})
```

SEXE	AGE	POIDS	TABAC
0	74	90.773333	75
1	68	89.320000	25

GROUPEMENTS ET TRIS

- Filtrage

```
df[df['RONFLE']=='oui']
#Ind. dont la variable RONFLE="oui"

df[(df['RONFLE']=='oui') & (df['POIDS'] < 70)]
#ind. de moins de 70kg qui ronflent
```

- Tri

```
df.sort_values(by = ['AGE', 'POIDS'], ascending = True)
#Tri par ordre asc. de l'age, puis du poids

df.sort_values(by = ['AGE', 'POIDS'],
ascending = [True, False], inplace=True)
#Tri par ordre asc. de l'age, puis desc. du poids
```

TESTS STATISTIQUES

- Test de normalité: Shapiro-Wilk

```
from scipy.stats import shapiro
stat, pvalue = shapiro(df.TAILLE)
```

- Test du chi deux

```
from scipy.stats import chi2_contingency
table = pd.crosstab(df.RONFLE, df.SEXE)
stat, pvalue, dof, expected = chi2_contingency(table)
```

- Test du chi deux

```
from scipy.stats import chi2_contingency
table = pd.crosstab(df.RONFLE, df.SEXE)
stat, pvalue, dof, expected = chi2_contingency(table)
```

N° de la question : 392

PETIT POINT D'ETAPE…

Les notation « df.col1 » et « df['col1'] sont équivalentes (avec « df » un dataframe et « col1 » un nom de colonne):

A. Vrai

B. Faux

https://toreply.univ-lille.fr/

PROGRAMME DE LA SÉANCE 2

1. Enjeux des traitements préliminaires
2. Exploration des données
3. Nettoyage et préparation des données

NETTOYAGE ET PREPARATION DES DONNEES SOUS PYTHON

SECTION 3

RECODAGE DE VARIABLES

- Discrétisation de variable quantitative

```
df["int_effectif"]=pd.qcut(df.effectif, 2,
labels=["petit-eff","grand-eff"])
```

- Regroupement de modalités

```
referentiel = {"Licence 3": "Licence",
"Master 1": "Master",
"Master 2": "Master"}
df["diplome_prepare"] = df["niveau"].map(referentiel)
```

	intitule	niveau	responsable	effectif	int_effectif
0	L3 MIASH-IDS	Licence 3	JV	53	grand-eff
1	M1 Info	Master 1	FB	70	grand-eff
2	M2 SISE	Master 2	RR	24	petit-eff
3	M2 OPSIE	Master 2	NH	20	petit-eff
4	M2 BIBD	Master 2	OB	22	petit-eff

GESTION DES VALEURS MANQUANTES

Dans les dataframes, les valeurs manquantes sont représentées par le code « Na » ou « NaN ». Elles sont:

- déduites automatiquement des valeurs vides

```
df = pd.read_csv(...)
```

- ou explicitement définies

```
df = pd.read_csv(..., na_values="-")
```

	intitule	niveau	responsable	effectif
0	L3 MIASH-IDS	Licence 3	JV	53.0
1	M1 Info	Master 1	FB	70.0
2	M2 SISE	Master 2	RR	24.0
3	M2 OPSIE	Master 2	NH	NaN
4	M2 BIBD	NaN	OB	22.0

GESTION DES VALEURS MANQUANTES

- Elimination des lignes/colonnes incomplètes

```
df = df.dropna(axis=0) # sup. des lignes incomplètes
```

- Remplacement par une valeur fixe

```
df.effectif=df.effectif.fillna(20)
```

- Remplacement par le mode

```
df.niveau.fillna(df.niveau.mode()[0], inplace=True)
```

- Remplacement par la médiane

```
df.effectif=df.effectif.fillna(df.effectif.median())
```

EXTRACTION DE DONNÉES : ILOC

Sélection à base d'indices

- Sélection en lignes uniquement

```
df.iloc[0] # première ligne
df.iloc[-1] # dernière ligne
df.iloc[0:10] # 10 premières lignes
df.iloc[[0,2,4]] #1ère, 3ème et 5ème lignes
```

- Sélection simultanée en lignes et colonnes

```
df.iloc[:,0] # toutes les lignes, première colonne
df.iloc[1:3,2:4] # lignes 2 à 3, colonnes 3 à 4
df.iloc[[0,3],[1,2]] # lignes 1 et 4, colonnes 2 et 3
```

EXTRACTION DE DONNÉES : LOC

Sélection à base de labels et de conditions

- Sélection par labels

```
df.loc[["M2 SISE", "M2 BIBD"]]
```

	intitule	niveau	responsable	effectif
M2 SISE	M2 SISE	Master 2	RR	24.0
M2 BIBD	M2 BIBD	Master 2	OB	22.0

```
df.loc["M2 SISE":"M2 BIBD", ['niveau','effectif']]
```

	niveau	effectif
M2 SISE	Master 2	24.0
M2 OPSIE	Master 2	NaN
M2 BIBD	Master 2	22.0

- Sélection par conditions

```
df.loc[df['effectif']<30, ['niveau','effectif']]
```

RESTRUCTURATION DE DONNÉES

Désagréger un tableau de contingence

	Formation	2017	2018	2019
0	BIBD	19	22	21
1	OPSIE	25	27	24
2	SISE	26	24	25

```
pd.melt(df, id_vars='Formation',
value_vars=['2017','2018','2019'])
```

	Formation	variable	value
0	BIBD	2017	19
1	OPSIE	2017	25
2	SISE	2017	26
3	BIBD	2018	22
4	OPSIE	2018	27
5	SISE	2018	24
6	BIBD	2019	21
7	OPSIE	2019	24
8	SISE	2019	25

RECODAGE DISJONCTIF BINAIRE

Transformer des variables qualitatives en quantitatives

```
df2  = pd.get_dummies(df)
```

	effectif	intitule_L3 MIASH-IDS	intitule_M1 Info	intitule_M2 BIBD	intitule_M2 OPSIE	intitule_M2 SISE
0	53	1	0	0	0	0
1	70	0	1	0	0	0
2	24	0	0	0	0	1
3	20	0	0	0	1	0
4	22	0	0	1	0	0

INTERROGATION DE BD AVEC SQLALCHEMY

Compatible aux SGBD relationnels

- Création d'une passerelle

```
from sqlalchemy import create_engine
engine = create_engine('sqlite:///test.db')
```

- Création d'une table à partir d'un dataframe

```
df.to_sql(name="table_name", con=engine,
if_exists='replace')
```

- Création d'un dataframe à partir d'une requête

```
df = pd.read_sql_query("select * from table_name",
con=engine)
```

JOINTURES DE DATAFRAMES

Jointure interne

	etudiant	formation
0	Bob	M1 Info
1	Nina	M2 SISE
2	Nico	M2 SISE
3	Mary	M1 Info
4	Emma	M1 Info

	intitule	niveau	responsable
0	L3 MIASH-IDS	Licence 3	JV
1	M1 Info	Master 1	FB
2	M2 SISE	Master 2	RR
3	M2 OPSIE	Master 2	NH
4	M2 BIBD	Master 2	OB

```
pd.merge(etudiants, formations, left_on='formation',
right_on='intitule')
```

	etudiant	formation	intitule	niveau	responsable	effectif	int_effectif	diplome_prepare
0	Bob	M1 Info	M1 Info	Master 1	FB	70	grand-eff	Master
1	Mary	M1 Info	M1 Info	Master 1	FB	70	grand-eff	Master
2	Emma	M1 Info	M1 Info	Master 1	FB	70	grand-eff	Master
3	Nina	M2 SISE	M2 SISE	Master 2	RR	24	petit-eff	Master
4	Nico	M2 SISE	M2 SISE	Master 2	RR	24	petit-eff	Master

JOINTURES DE DATAFRAMES

- Jointure gauche

```
pd.merge(etudiants, formations, left_on='formation',
right_on='intitule', how='left')
```

- Jointure droite

```
pd.merge(etudiants, formations, left_on='formation',
right_on='intitule', how='right')
```

- Jointure externe

```
pd.merge(etudiants, formations, left_on='formation',
right_on='intitule', how='outer')
```

- Concaténation

```
pd.concat([df1, df2], axis=1)
```

NORMALISATION DE DONNÉES

Centrer-réduire les données

	poids	taille	revenu
0	59	1.75	1400
1	75	1.82	1750
2	80	1.72	2300
3	79	1.91	1600

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit_transform(df)
```

```
array([[-1.68893523, -0.68358593, -1.08453851],
       [ 0.2074131 ,  0.27343437, -0.03739788],
       [ 0.80002195, -1.09373748,  1.60810882],
       [ 0.68150018,  1.50388904, -0.48617243]])
```

N° de la question : 815

PETIT POINT D'ETAPE…

Le recodage disjonctif binaire s'applique:

A. Aux variables catégorielles uniquement

B. Aux variables numériques uniquement

C. Aux variables catégorielles et quantitatives

https://toreply.univ-lille.fr/

QUELQUES RESSOURCES

- https://www.kdnuggets.com/2019/06/select-rows-columns-pandas.html
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html
- https://machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet/