Intelligence Artificielle pour l'Analyse de Données

Data Preprocessing

Akka Zemmari
zemmari@u-bordeaux.fr

LaBRI, Université de Bordeaux

2019-2020

De quoi allons nous parler dans ce cours

Pré-traitement des données avant leur exploitation :

- Charger des données
- Extraire des sous ensembles de données
- Faire des calculs sur ces données
- Traiter les données manquantes et les valeurs aberrante
- "Feature scaling" : Normaliser, standardiser les données
- ...

Bibliothèques Python pour l'analyse de données

Les bibliothèques les plus populaires

- NumPy
- SciPy
- Pandas
- SciKit-Learn

Visualization des données :

- matplotlib
- Seaborn

SciPy

- Ensemble d'algorithmes pour l'algèbre linéaire, les équations différentielles, le calcul d'intégrales, les statistiques, etc
- Basé sur NumPy
- Site officiel : `https://www.scipy.org/scipylib/`

Pandas

- Définit des structures de données et des outils pour travailler sur des données en tables,
- fournit un ensemble d'outils pour manipuler des données : chargement, fusion, "reshaping", tri, aggrégation, etc
- permet de gérer les données manquantes,
- site officiel : `https://pandas.pydata.org/`

Pandas, manipulations de base

Etape 1. Charger les données

Avant tout, il faut charger les bibliothèques Python nécessaires :

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pa
```

Dans le reste de ce cours, on considère que les données sont stockées dans un fichier `my_data.csv`.

Pandas

Etape 1. Charger les données

Ensuite, on charge les données avec Pandas :

```
1  my_dataset = pa.read_csv('./my_data.csv', skiprows=1, low_memory=False)
```

Pandas

Etape 1. Charger les données

Ensuite, on charge les données avec Pandas :

```
1  my_dataset = pa.read_csv('./my_data.csv', skiprows=1, low_memory=False)
```

Cette instruction :

1. charge le contenu du fichier dans un `DataFrame` pandas,

Pandas

Etape 1. Charger les données

Ensuite, on charge les données avec Pandas :

```
1  my_dataset = pa.read_csv('./my_data.csv', skiprows=1, low_memory=False)
```

Cette instruction :

1. charge le contenu du fichier dans un `DataFrame` pandas,
2. supprime la première ligne du fichier, ce qui peut être très utile si celle-ci contient des informations sur le contenu du fichier et non les noms des colonnes,

Pandas

Etape 1. Charger les données

Ensuite, on charge les données avec Pandas :

```
1  my_dataset = pa.read_csv('./my_data.csv', skiprows=1, low_memory=False)
```

Cette instruction :

1. charge le contenu du fichier dans un `DataFrame` pandas,
2. supprime la première ligne du fichier, ce qui peut être très utile si celle-ci contient des informations sur le contenu du fichier et non les noms des colonnes,

On peut également supprimer les colonnes dans lesquelles on a moins de 50% de données manquantes :

```
1  my_dataset = my_dataset.dropna(thresh=half_count,axis=1)
```

Etape 2. Explorer les données

Afficher les 5 premières lignes :

```
my_dataset.head()
```

Visualiser les types des données :

```
#Check a particular column type
my_dataset['nom_col'].dtype
#Check types for all the columns
my_dataset.dtypes
```

Pandas

Méthodes usuelles :

`my_dataset.method()`	Description
`head[n], tail([n])`	first/last n rows
`describe()`	generate descriptive statistics (for numeric columns only)
`max(), min()`	return max/min values for all numeric columns
`mean(), median()`	return mean/median values for all numeric columns
`std()`	standard deviation
`sample([n])`	returns a random sample of the data frame
`dropna()`	drop all the records with missing values

Sélectionner une colonne en particulier :

```
#Subset the data frame using column name:
my_dataset['nom_col']

#Use the column name as an attribute:
my_dataset.nom_col
```

Imputation des données manquantes

Données manquantes : Un grand problème en analyse de données.

- S'il manque beaucoup de données, on supprime simplement la colonne
- Sinon, on utilise une des stratégies suivantes pour remplir les données manquantes :
  - `mean` : les données manquantes sont remplacées par la moyenne de la variable. Il s'agit de la méthode par défaut
  - `median` : on utilise la valeur médiane de la variable
  - `most_frequent`, on utilise la valeur la plus fréquente. Attention, cette stratégie n'a pas de sens si la variable est continue
  - ...

Feature scaling

La plupart du temps, les données proviennent avec des ordres de grandeurs différents. Cette différence d'échelle peut conduire à des performances moindres an analyse de données et en machine learning.

Pour palier à cela, des traitements préparatoires sur les données existent. Notamment le Feature Scaling qui comprend la Standardisation et la Normalisation.

Feature scaling

Normalisation

La normalisation peut- être effectuée par la technique du Min-Max Scaling. La transformation se fait grâce à la formule suivante :

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

Feature scaling

Standardisation

La standardisation peut- être appliquée quand les input features répondent à des distributions normales avec des moyennes et des écart-types différents.

Par conséquent, cette transformation aura pour impact d'avoir toutes nos features répondant à la même loi normale $\mathcal{N}(0, 1)$. La transformation se fait grâce à la formule suivante :

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

Feature scaling

En pratique :

```
1 from sklearn.preprocessing import MinMaxScaler
2 ...
```