Machine Learning I
Introduction à pandas

Nicolas Bourgeois

Télécharger

Data and Cheatsheets :

`ouralou.fr/Resources/epita/C1.zip`

Vous aurez aussi besoin de :

`mysql-python.sourceforge.net/MySQLdb.html#mysqldb`

Exercice

Exercice

Chargez le fichier `data1.csv` dans une table. Identifiez quelles sont les colonnes qui contiennent le plus de valeurs manquantes.

Exercice

Supprimez les colonnes avec valeurs manquantes et affichez les cinq premières lignes. Que se passe-t-il si vous éliminez les plutôt les lignes avec valeurs manquantes ?

Exercice

Affichez cinq lignes aléatoires (doublons autorisés).

Solution

```
import pandas as pd
from numpy.random import randint
## question 1
df1 = pd.read_csv('./C1/data1.csv')
print(df1.info())
## question 2
df2 = df1.dropna(axis=1)
print(df1.head())
print(df2.head())
## question 3
n_lignes = df1.shape[0]
rand_vect = randint(0,n_lignes,size=5)
print(df1.loc[rand_vect])
```

Exercice

Exercice

Produisez la sous-table des passagers de 1ere classe.

Exercice

Produisez la sous-table des passagers masculins d'âge compris entre 30 et 50 inclus.

Exercice

A l'aide de `groupby`, trouvez l'effectif, l'âge moyen et le taux de survie des passagers par classe.

Solution

```
import pandas as pd
from numpy.random import randint
df1 = pd.read_csv('./C1/data1.csv')
## question 1
class1 = df1[df1.pclass == 1]
print(class1.tail())
## question 2
patriarcat = df1[(df1.sex == 'male') &
    (df1.age > 29) & (df1.age < 51)]
print(patriarcat.head())
## question 3
par_classe = df1.groupby('pclass')
print(par_classe['pclass'].count())
print(par_classe['age','survived'].mean())
```

Problème

Exercice

Dans le fichier `data2.csv`, trouvez le total cumulé de développement des provinces contrôlées par Muscovy, Ryazan et Novgorod qui ne produisent pas de céréales ('Grain').

Exercice

Même question mais en appliquant préalablement une fonction qui diminue pour chaque province produisant de la fourrure ('Fur') le développement de 5, sans toutefois pouvoir le descendre en dessous de 3.

Solution - première partie

```
import pandas as pd
import numpy as np

provinces = pd.read_csv("./C1/data2.csv",sep=";",
    na_values='nan',skipinitialspace=True)
to_sum=provinces.loc[provinces.country.isin(['Muscovy',
  'Ryazan','Novgorod']) & (provinces.goods != 'Grain')]
total = to_sum.dev.sum()
print(total)
```

Solution - deuxième partie

```
import pandas as pd
import numpy as np
provinces = pd.read_csv("./C1/data2.csv",sep=";",
    na_values='nan',skipinitialspace=True)
to_sum=provinces.loc[provinces.country.isin(['Muscovy',
  'Ryazan','Novgorod']) & (provinces.goods != 'Grain')]
to_sum.loc[to_sum.goods == 'Fur',
    'dev'] = to_sum.loc[to_sum.goods == 'Fur',
    'dev'].apply(lambda x:max(3,x-5))
total = to_sum.dev.sum()
print(total)
```

Exercice

Exercice

Importez sur un serveur la base de données data3.sql. Puis avec une interface de type MySQLdb, lisez-la avec pandas sous forme d'un dictionnaire de dataframes (la clef étant le nom).

Solution

```
from MySQLdb import *
import pandas as pd
host,user,pwd='localhost','root','root'
dbname = 'MissiDominici'
try:
    db = connect(host, user, pwd, dbname)
    curseur = db.cursor()
    curseur.execute("SHOW TABLES")
    tablenames = [t[0] for t in curseur.fetchall()]
except Error as e:
    print ("Error %d: %s" % (e.args[0], e.args[1]))
tables = dict()
for t in tablenames:
    sqlr = 'SELECT * FROM '+t
    tables[t] = pd.read_sql(sqlr,con=db)
db.close()
print(tables[tablenames[-1]])
```

Exercice

Exercice

Dans la table des personnages, ne gardez que les champs `'id'`, `'name'` et `'date_start'`, puis éliminez les lignes avec valeurs manquantes.

Exercice

Faites trois joins de cette table avec les tables appropriées de façon à ajouter des colonnes mentionnant le titre complet de chaque personnage, par exemple :

"Louis le Pieux, roi, Aquitaine".

Solution - première partie

```
from MySQLdb import *
import pandas as pd
host,user,pwd='localhost','root','root'
dbname = 'MissiDominici'
db = connect(host, user, pwd, dbname)
curseur = db.cursor()
curseur.execute("SHOW TABLES")
tablenames = [t[0] for t in curseur.fetchall()]
tables = dict()
for t in tablenames:
    sqlr = 'SELECT * FROM '+t
    tables[t] = pd.read_sql(sqlr,con=db)
db.close()
persos = tables['actor'][['id','name','date_start']]
persos = persos.dropna()
print(persos)
```

Solution - deuxième partie

```
from MySQLdb import *
import pandas as pd
host,user,pwd='localhost','root','root'
dbname = 'MissiDominici'
db = connect(host, user, pwd, dbname)
curseur = db.cursor()
curseur.execute("SHOW TABLES")
tablenames = [t[0] for t in curseur.fetchall()]
tables = dict()
for t in tablenames:
    sqlr = 'SELECT * FROM '+t
    tables[t] = pd.read_sql(sqlr,con=db)
db.close()
persos = tables['actor'][['id','name','date_start']].dropna()
n_to_n = tables['actor_has_role_and_place'][['actor','role','place']]
roles = tables['role'][['id','name']].rename(columns={'name':'title'})
places = tables['place'][['id','name']].rename(columns={'name':'from'})
persos = pd.merge(left=persos,right=n_to_n,left_on='id',right_on='actor')
persos = pd.merge(left=persos,right=roles,left_on='role',right_on='id')
persos = pd.merge(left=persos,right=places,left_on='place',right_on='id')
print(persos[['name','date_start','title','from']])
```