Четене на данни в Python

За да продължим напред ще представим няколоко начина за четене на файлове, съдържащи множества от наблюдения чрез Python.

Следните редове въвеждат данни от програмния код в променливите u и v:

```
import numpy as np
u = [0, 1.2, 2.4, 3.2, 4, 5, 6, 7, 8, 9]
v = [1.5, 2.8, 2, 5, 7, 8, 8.5, 9.2, 10.4, 12.5]
```

Първата команда зарежда в паметта библиотеката numpy с краткото име np. Присвояваме на променливите u и v съответните поредици от числови стойности по указания начин. За да проверим дали правилно са въведени данните отпечатваме променливите u и v, като използваме цикъл. Първият ред от цикъла започва с командата for и завършва с две точки. Променливата i е брояч на цикъла, който се променя от 0 до 10, без да приема стойност 10. След двете точки се преминава на следващия ред и започва тялото на цикъла. Всички команди в тялото на цикъла се изписват на нов ред и започват няколко интервала по-навътре, точно под променливата на цикъла (в случая i). След приключване на тялото на цикъла се оставя празен ред за края на цикъла.

```
print (u,v)
for i in range(0,10):
    print (u[i],' ',v[i])
```

Следваща възможност за въвеждане на стойностите на променливите е чрез редовете:

```
x = np.array([0, 1.2, 2.4, 3.2, 4, 5, 6, 7, 8, 9])
y = np.array([1.5, 2.8, 2, 5, 7, 8, 8.5, 9.2, 10.4, 12.5])
```

По този начин x и y са въведени като променливи от тип “array” в Python.

В следващите редове в променливата dataset са въведени стойности като наредени двойки. Когато командата не може да се събере на един ред се натиска Enter и въвеждането продължава от там където е разположен маркерът – в случая следващата наредена двойка е точно под първата на предходния ред:

```
dataset = [[0, 1.5], [1.2, 2.8], [2.4, 2], [3.2, 5], [4, 7],
           [5, 8], [6, 8.5], [7,9.2], [8,10.4],[9, 12.5]]
z = [row[0] for row in dataset]
s = [row[1] for row in dataset]
```

След като в dataset са разположени наредените двойки, то в следващите два реда на променливите се присвояват съответните стойности от наредениет двойки, чрез специфичния цикъл за въвеждане на данни.

Сега ще покажем как се четат данни от файл с добавка “csv”. Това файлове създадени на MS Excel и се отварят пак в MS Excel.

```
import pandas
dataset = pandas.read_csv("insurance.csv")
data = dataset.values
x=data[:,0]
y=data[:,1]
```

В началото в паметта се зарежда библиотеката pandas, тъй като в нея се намира командата read_csv. Тази команда има много особености и възможности за използване и за нея може да се прочете от интернет, от съответния сайт https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html . В най-обикновен вид тя е използвана на втория ред. След прочитане данните са прехвърлени в променливата dataset, която е от тип DataFrame. За същността и особеностите на този тип трябва да се прочете описанието https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html Съгласно това описание данните се отделят чрез комадата “values”. На третия ред в променливата data приема стойностите, като двумерен масив. Променливите x и y получават съответно първа и втора колона на данните от масива data. В Python номерата на редовете и колоните винаги започват от 0, т.е. първата колона (в случая) има номер 0. А двете точки “:” означават всички елементи, от първия до последния.

По подобен начин се четат и текстови файлове с добавка “txt”. Например

```
import pandas
datasetTXT = pandas.read_csv('AutoInsurSweden.txt',
                             delim_whitespace=True, decimal=',')
data = datasetTXT.values
x=data[:,0]
y=data[:,1]
```

При четенето на текстовите файлове трябва допълнително да се укаже какви са особеностите на данните във файла, за да може правилно да бъдат

разчетени. За тези особености трябва да се прочете от описанието на командата read_csv от съответния сайт.

Още един пример:

```
import numpy
data = numpy.loadtxt('ex2data1.txt', delimiter = ',')
x = data[:,0:2]
y = data[:,2]
```

В този пример текстовият файл с данни се чете с командата loadtxt от библиотеката numpy. За това как работи тази команда може да се прочете от сайта
https://docs.scipy.org/doc/numpy/reference/generated/numpy.loadtxt.html
След прочитане на данните в променливата data те се прехвърлят към променливите x и y. В променливата x се зареждат две колони с данни – това са колона 0 и колона 1. А колона 2 не се прехвърля към x. Колона 2 се прехвърля в променливата y, както е указано на последния ред.

Със следващият пример ще покажем как след четене на данни от даден файл се коригират често срещани неудобства при провеждане на анализ на данни. Това ще направим върху файла ILPD.csv, съдържащ наблюдения на болни от чернодробни заболявания https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset) (декември 2019).

```
#------------
print
print ' Read the data Indian Liver Patient Dataset '
# for Indian Liver Patient Dataset  --- ILPD.csv
import pandas as pd
dataset = pd.read_csv("ILPD.csv", delimiter = ',',header=None)
print (dataset.shape)
```

Наблюденията са прочетени в променливата dataset, а чрез командата dataset.shape намираме размерността на променливата dataset. Променливата dataset е от тип DataFrame.

Със следващата команда откриваме наблюдения, които съдържат стойност „nan“. Променливата indexNAN съдържа номерата на редове, които съдържат „nan“.

```
#
import numpy
indexNAN = numpy.nonzero(pd.isnull(dataset.values).any(1))[0]
print (indexNAN)
```

С командата изпускаме редовете , които се съдържат в indexNAN. Размерността на променливата dataset е променена.

```
# изпускаме редове
dataset=dataset.drop([indexNAN[i] for i in range(len(indexNAN))])
print (dataset.shape)
```

Следващото неудоство на наблюденията е, че съдържат тесктова колона, а именно втората колона, която представя информация за пола на пациента. За стравян с това неудобство съществуват две възможности. Едната е просто да пренебрегнем тази колона и да работим само с числовите колони. Това става с командата

```
#
# изпускаме колона с текст Male , Female
dataset.columns = ['A0', 'A1', 'A2','A3', 'A4', 'A5','A6', 'A7','A8', 'A9','class']
datasetA1=dataset.drop('A1', axis=1)
print (datasetA1.shape)
```

Променливата datasetA1 не сдържа текстовата колона. Със следващата команда отделяме само числовите характеристики (колоните) от множеството от данни.

```
# Отделяне на числовите характеристики
dataset_num = dataset._get_numeric_data()  #keep only numeric features
#
```

Прехвърляме данните в променливите X и y и продължаваме работа с тях.

```
X = dataset_num.values[:,:9]
y = dataset_num.values[:,9]
```

Нека да отблежим, че изпускането на една или няколко характеристики може да доведе до неточности при решаване на съответната задача от анализа на наблюденията.

Другата възможност е да преобразуваме текстовата колона в числова. За тази цел в Python има определени команди. Следващите няколко реда правят това.

```
# или преобразуване на текстовата характеристика към числова
# Male   = 1
# FeMale = 0
dataset.columns = ['A0', 'A1', 'A2','A3', 'A4', 'A5','A6', 'A7','A8', 'A9','class']
from sklearn.preprocessing import LabelEncoder
labelencoder_set = LabelEncoder()
# 0= Female, 1= Male
s=dataset["A1"].values
s=labelencoder_set.fit_transform(s)
dataset["A1"]=pd.to_numeric(s)
```

След това в променливата dataset всички характеристики са само числови. Зареждаме данните в променливите X и y.

```
#
X = dataset.values[:,:10]
y = dataset.values[:,10]
```