Python, Wine Dataset and Pandas

Abul Hasan

Birkbeck

January 30, 2019

Introducing Numpy

First we must import the numpy module

```
import numpy as np
```

Matrix Creation:

```
a =np.array([1.2,2.5],[3.2,2.8],[1.1,4.5])
```

$$\begin{bmatrix} 1.2 & 2.5 \\ 3.2 & 2.8 \\ 1.1 & 4.5 \end{bmatrix}$$

Information

```
#object type, data type, dimension,shape
print(type(a))
print(a.dtype)
print(a.ndim)
print(a.shape)
print(a)#visualization
```

Creating matrix from a sequence of numbers

- #Creating a matrix from a list of evenly spaced values
  a =np.arange(0,10).reshape(2,5)# arange(): 0 to 9
  print(a)
- #creating a matrix from a vector
  a = np.array([2.1,3.4,6.7,8.1,3.5,7.2])
  b = a.reshape(3,2)
  print(b) # b matrix has 3 rows 2 columns
- #repeating 8 times the value 0
  #e.g. for initialization
  a =np.zeros(shape=(2,4))

Modifying the size of a matrix

- a = np.array([[1.2,2.5],[3.2,1.8],[1.1,4.3]])
  b =np.array([[4.1,2.6]])
  c = np.append(a,b,axis=0) #axis=0 is the row binding
  print(c)
- Exercises:
  - Append a column to the "a" matrix
  - Print the first row vector of the matrix. Hint:a[:,1] will print the first column
  - How can you insert a column or a row at a specific position in the matrix?
  - Can you make sense of the output from the following code?
    print(np.argmax(a,axis=0))

Linear Algebra

- Consider the following symmetric matrix:

$$\begin{bmatrix} 43.81 & 50.01 & 47.64 & 36.74 & 42.00 \\ 50.01 & 57.22 & 54.53 & 41.66 & 48.22 \\ 47.64 & 54.53 & 51.97 & 39.64 & 45.98 \\ 36.74 & 41.66 & 39.64 & 31.40 & 34.44 \\ 42.00 & 48.22 & 45.98 & 34.64 & 40.84 \end{bmatrix}$$

- This is the matrix in python.`http://www.dcs.bbk.ac.uk/~abulhasan/dsta/DSTALab02/matrix.py`
- Use transpose function in numpy to transpose the matrix.
- Find the eigenvalues and eigenvectors from this matrix using numpy function.

Wine Review Dataset

- The dataset can be found from: `https://www.kaggle.com/zynicide/wine-reviews`
- 130k wine reviews with variety, location, winery, price, and description.
- We are going to use JSON file format in this class.
- JSON is a text file where data is stored in nested key-value pair. Much analogous to the python dictionaries.
- JSON data can be loaded using json package.
- We are going to use pandas instead.

DataFrame from Pandas

- The DatFrame object in Pandas is really a dictionary with values appended as list objects. A simple example will reveal the fact.

```
import pandas as pd
new_dataframe=pd.DataFrame({'column1':[1,2,3,4,5],
                            'column2':[2.3,4.5,6.7,6.5,5.5]})
print(new_dataframe)
```

Importing json objects into the dataframe object

```
import json
import pandas as pd
#package for flattening json in pandas df
from pandas.io.json import json_normalize
#load json object
with open('winemag-data-130k-v2.json') as f:
    d = json.load(f)
#lets put the data into a pandas df
#d[0] points to the first datapoint
nycphil = json_normalize(d[0])
nycphil.head()
```

Loading a dataset from a weblink

```
 import pandas as pd
df = pd.read_json("https://www.dcs.bbk.ac.uk/"
+"~abulhasan/dsta/"+
"wine-reviews/winemag-data-130k-v2.json")
```

DataFrame Illustration

- Pandas DataFrame object takes care of missing data points in the raw data.

Figure 1: DataFrame Padding from DataCamp

- DataFrame object can be created from array and be manipulated like arrays in numpy. Try this code: `https://www.dcs.bbk.ac.uk/~abulhasan/dsta/DSTALab02/pd_dataframe.py`