Department of Data Science - Data and Visual Analytics Lab

Lab1.Red Wine Quality Data Analytics using NumPy Part-I

Objectives

In this lab, you will learn the basics of NumPy.

How to Use This Jupyter Notebook

For each question, you should write NumPy statements in the "In[]" Cell and the expected output "Out[]" is already shown just below all In[] cells.

```
In [1]:  '''
         Wine quality dataset 11 input features and 1 output feature

         1 - fixed acidity
         2 - volatile acidity
         3 - citric acid
         4 - residual sugar
         5 - chlorides
         6 - free sulfur dioxide
         7 - total sulfur dioxide
         8 - density
         9 - pH
         10 - sulphates
         11 - alcohol
         Output variable (based on sensory data):
         12 - quality (score between 0 and 10)'''
```

```
Out[1]:  '\nWine quality dataset 11 input features and 1 output feature\n\n1 - fixed a
         cidity\n2 - volatile acidity\n3 - citric acid\n4 - residual sugar\n5 - chlori
         des\n6 - free sulfur dioxide\n7 - total sulfur dioxide\n8 - density\n9 - pH\n
         10 - sulphates\n11 - alcohol\nOutput variable (based on sensory data):\n12 -
         quality (score between 0 and 10)'
```

import modules for numpy

```
In [2]:  import numpy as np
```

```
In [3]:  wines = np.genfromtxt("winequality-red.csv", delimiter=";", skip_header=1)
```

What is its size?

```
In [4]:  Wines.shape
Out[4]:  (1599, 12)
```

How many wine data rows here?

```
In [5]:  Wines.Shape [0]
Out[5]:  1599
```

How many wine data columns here?

```
In [6]:  Wines.shape [1]
Out[6]:  12
```

How many dimensions?

```
In [7]:  Wines.ndim
Out[7]:  2
```

What is the type of wines?

```
In [8]:  type (Wines)
Out[8]:  numpy.ndarray
```

What is the data type of wines data?

```
In [9]:  Wines. dtype
Out[9]:  dtype('float64')
```

Show top 5 rows

```
In [10]: wines[:5, :]
```

Scanned with CamScanner

What is the value at 3rd row, 4th column of wine data?

```
In [11]: wines[2,3]
Out[11]: 2.3
```

Select first 3 items in 4th column

```
In [12]: wines[:3,3]
Out[12]: array([1.9, 2.6, 2.3])
```

Show 1st column

```
In [13]: wines[:,0]
Out[13]: array([7.4, 7.8, 7.8, ..., 6.3, 5.9, 6. ])
```

Show 2nd row

```
In [14]: wines[1,:]
Out[14]: array([ 7.8   ,  0.88  ,  0.    ,  2.6   ,  0.098 , 25.    , 67.    ,
                 0.9968,  3.2   ,  0.68  ,  9.8   ,  5.    ])
```

Select items from rows 1 to 3 and 5th column

```
In [15]: wines[1:4,4]
Out[15]: array([0.098, 0.092, 0.075])
```

Select entire array

```
In [16]: wines[:,:]
Out[16]: array([[ 7.4  ,  0.7  ,  0.   , ...,  0.56 ,  9.4  ,  5.   ],
                [ 7.8  ,  0.88 ,  0.   , ...,  0.68 ,  9.8  ,  5.   ],
                [ 7.8  ,  0.76 ,  0.04 , ...,  0.65 ,  9.8  ,  5.   ],
                ...,
                [ 6.3  ,  0.51 ,  0.13 , ...,  0.75 , 11.   ,  6.   ],
                [ 5.9  ,  0.645,  0.12 , ...,  0.71 , 10.2  ,  5.   ],
                [ 6.   ,  0.31 ,  0.47 , ...,  0.66 , 11.   ,  6.   ]])
```

Change 1st value in wines to 100

```
In [17]: # show actual value wines[0,0]

Out[17]: 7.4
```

```
In [18]: # update wines[0,0] =100
```

```
In [19]: # show updated value ~~wines[0,0] = 7.4~~

Out[19]: 100.0
```

# change it back to 7.4 and print

```
In [20]: wines[0,0] =7.4
```

1-Dimensional Numpy Arrays

Select 4th row all column values

```
In [21]: third_wines = wines[3,:]
```

display its value

```
In [22]: third_wine

Out[22]: array([11.2  ,  0.28 ,  0.56 ,  1.9  ,  0.075, 17.   , 60.   ,  0.998,
                 3.16 ,  0.58 ,  9.8  ,  6.   ])
```

show 2nd value

```
In [23]: third_wine[1]

Out[23]: 0.28
```

Convert wine data to integer values and show it

```
In [24]: #convert to int wines.astype(int)

Out[24]: array([[ 7,  0,  0, ...,  0,  9,  5],
                [ 7,  0,  0, ...,  0,  9,  5],
                [ 7,  0,  0, ...,  0,  9,  5],
                ...,
                [ 6,  0,  0, ...,  0, 11,  6],
                [ 5,  0,  0, ...,  0, 10,  5],
                [ 6,  0,  0, ...,  0, 11,  6]])
```

Vectorization Operations

Increase wine quality score (output variable) by 10

```
In [25]: # check values first wines[:,11]

Out[25]: array([5., 5., 5., ..., 6., 5., 6.])
```

Increase by 10

```
In [26]: wines[:,11] += 10
```

Display update score

```
In [28]: wines[:, 11]

Out[28]: array([15., 15., 15., ..., 16., 15., 16.])
```

Multiply alcohol of all wine data by 3 times

```
In [29]: wines[:,10] *= 3
```

Show updated alcohol column

```
In [30]: wines[:, 10]

Out[30]: array([28.2, 29.4, 29.4, ..., 33. , 30.6, 33. ])
```

Add quality column by itselt

```
In [31]: # It will produce a new array wines[:, 11] + wines[:, 11]

Out[31]: array([30., 30., 30., ..., 32., 30., 32.])
```

Multiply alcohol and wine quality columns. It will perform element wise multiplication

```
In [32]: Wines [:, 10] * wines [:, 11]

Out[32]: array([423., 441., 441., ..., 528., 459., 528.])
```

Broadcasting

Add every row of wines data with a random array of values

```
In [33]: rand_array = np.random.rand(12)
```

Show rand_array

```
In [34]: rand_array

Out[34]: array([0.72587682, 0.9600024 , 0.17236312, 0.56655827, 0.20321856,
                0.47046892, 0.88980548, 0.40083145, 0.97884246, 0.2112331 ,
                0.33194581, 0.72612594])
```

add wines and rand_array

```
In [35]: Wines+rand_array

Out[35]: array([[ 8.12587682,  1.6600024 ,  0.17236312, ...,  0.7712331 ,
                 28.53194581, 15.72612594],
                [ 8.52587682,  1.8400024 ,  0.17236312, ...,  0.8912331 ,
                 29.73194581, 15.72612594],
                [ 8.52587682,  1.7200024 ,  0.21236312, ...,  0.8612331 ,
                 29.73194581, 15.72612594],
                ...,
                [ 7.02587682,  1.4700024 ,  0.30236312, ...,  0.9612331 ,
                 33.33194581, 16.72612594],
                [ 6.62587682,  1.6050024 ,  0.29236312, ...,  0.9212331 ,
                 30.93194581, 15.72612594],
                [ 6.72587682,  1.2700024 ,  0.64236312, ...,  0.8712331 ,
                 33.33194581, 16.72612594]])

In [ ]:
```