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

Name:Harshavardhini K

Rollno:235229108

Import modules for numpy

In [22]:
```
import numpy as np
```

In [23]:
```
wines = np.genfromtxt("winequality-red.csv", delimiter=";", skip_header=1)
```

What is its size?

In [24]:
```
wines.size
```

Out[24]: 19188

How many wine data rows here?

In [25]:
```
wines.shape[0]
```

Out[25]: 1599

How many wine data columns here?

In [26]:
```
wines.shape[0]
```

Out[26]: 1599

How many dimensions?

In [27]:
```
wines.ndim
```

Out[27]: 2

What is the type of data?

In [28]:
```
type(wines)
```

Out[28]:
```
numpy.ndarray
```

What is the data type of wines data?

In [29]:
```
wines.dtype
```

Out[29]:
```
dtype('float64')
```

Show top 5 rows

In [31]:
```
wines[:5, :]
```

Out[31]:
```
array([[7.400e+00, 7.000e-01, 0.000e+00, 1.900e+00, 7.600e-02, 1.100e+01,
        3.400e+01, 9.978e-01, 3.510e+00, 5.600e-01, 9.400e+00, 5.000e+00],
       [7.800e+00, 8.800e-01, 0.000e+00, 2.600e+00, 9.800e-02, 2.500e+01,
        6.700e+01, 9.968e-01, 3.200e+00, 6.800e-01, 9.800e+00, 5.000e+00],
       [7.800e+00, 7.600e-01, 4.000e-02, 2.300e+00, 9.200e-02, 1.500e+01,
        5.400e+01, 9.970e-01, 3.260e+00, 6.500e-01, 9.800e+00, 5.000e+00],
       [1.120e+01, 2.800e-01, 5.600e-01, 1.900e+00, 7.500e-02, 1.700e+01,
        6.000e+01, 9.980e-01, 3.160e+00, 5.800e-01, 9.800e+00, 6.000e+00],
       [7.400e+00, 7.000e-01, 0.000e+00, 1.900e+00, 7.600e-02, 1.100e+01,
        3.400e+01, 9.978e-01, 3.510e+00, 5.600e-01, 9.400e+00, 5.000e+00]])
```

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

In [32]:
```
wines[2,3]
```

Out[32]:
```
2.3
```

Select first 3 items in 4th column

In [33]:
```
wines[:3, 3]
```

Out[33]:
```
array([1.9, 2.6, 2.3])
```

Show 1st column

In [34]:
```
wines[:]
```

Out[34]:
```
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.   ]])
```

Show 2nd row

In [35]:
```
wines[1, :]
```

Out[35]:
```
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 [36]:
```
wines[1:4, 4]
```

Out[36]:
```
array([0.098, 0.092, 0.075])
```

Select entire array

In [37]:
```
wines[:,:]
```

Out[37]:
```
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 [38]:
```
wines[0,0]
```

Out[38]:
```
7.4
```

In [41]:
```
wines[0,0] = 100
```

In [40]:
```
wines[0,0]
```

Out[40]:
```
100.0
```

Change it back to 7.4 and print

In [42]:
```
wines[0,0] = 7.4
```

In [43]:
```
wines[0,0]
```

Out[43]:
```
7.4
```

1-Dimensional Numpy Arrays

Select 4th row all column values

In [44]:
```
fourth_row = wines[3,]
```

In [45]:
```
fourth_row
```

Out[45]:
```
array([11.2  ,  0.28 ,  0.56 ,  1.9  ,  0.075, 17.   , 60.   ,  0.998,
        3.16 ,  0.58 ,  9.8  ,  6.   ])
```

In [46]:
```
fourth_row[1]
```

Out[46]:
```
0.28
```

Convert wine data to integer values and show it

In [47]:
```
wines.astype(int)
```

Out[47]:
```
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 Operation

Increase wine quality score (output variable) by 10

In [48]:
```
wines[:,11]
```

Out[48]:
```
array([5., 5., 5., ..., 6., 5., 6.])
```

Increase by 10

In [49]:
```
wines[:, 11] += 10
```

Display update score

In [50]:
```
wines[:, 11]
```

Out[50]:
```
array([15., 15., 15., ..., 16., 15., 16.])
```

Multiply alcohol of all wine data by 3 times

In [51]:
```
wines[:, 10] *= 3
```

Show updated alcohol column

In [52]:
```
wines[:, 10]
```

Out[52]:
```
array([28.2, 29.4, 29.4, ..., 33. , 30.6, 33. ])
```

Add quality column by itselt

In [53]:
```
wines[:, 11] + wines[:, 11]
```

Out[53]:
```
array([30., 30., 30., ..., 32., 30., 32.])
```

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

In [54]:
```
wines[:,10] * wines[:,11]
```

Out[54]:
```
array([423., 441., 441., ..., 528., 459., 528.])
```

Broadcasting

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

In [55]:
```
rand_array=np.random.rand(12)
```

Show rand_array

In [56]:
```
rand_array
```

Out[56]:
```
array([0.57013749, 0.41723486, 0.07048208, 0.63356327, 0.40004755,
       0.23290089, 0.14470312, 0.79693345, 0.44543633, 0.88587652,
       0.68711421, 0.77106973])
```

add wines and rand_array

In [57]:
```
wines+rand_array
```

Out[57]:
```
array([[ 7.97013749,  1.11723486,  0.07048208, ...,  1.44587652,
         28.88711421, 15.77106973],
       [ 8.37013749,  1.29723486,  0.07048208, ...,  1.56587652,
         30.08711421, 15.77106973],
       [ 8.37013749,  1.17723486,  0.11048208, ...,  1.53587652,
         30.08711421, 15.77106973],
       ...,
       [ 6.87013749,  0.92723486,  0.20048208, ...,  1.63587652,
         33.68711421, 16.77106973],
       [ 6.47013749,  1.06223486,  0.19048208, ...,  1.59587652,
         31.28711421, 15.77106973],
       [ 6.57013749,  0.72723486,  0.54048208, ...,  1.54587652,
         33.68711421, 16.77106973]])
```

In [ ]:
```

```