HARI PRASATH S 225229110

In [114]:

```
'''
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[114]:

```
'\nWine quality dataset 11 input features and 1 output feature\n1 - fixed ac
idity\n2 - volatile acidity\n3 - citric acid\n4 - residual sugar\n5 - chlori
des\n6 - free sulfur dioxide\n7 - total sulfur dioxide\n8 - density\n9 - pH
\n10 - sulphates\n11 - alcohol\nOutput variable (based on sensory data):\n12
- quality (score between 0 and 10)'
```

Importing numpy package

In [3]:

```
import numpy as np
```

In [4]:

```
wines = np.genfromtxt("winequality-red.csv", delimiter=";", skip_header=1)
```

size

In [5]:

```
wines.shape
```

Out[5]:

```
(1599, 12)
```

wine data rows

In [118]:

```
wines.shape[0]
```

Out[118]:

```
1599
```

wine data columns

In [119]:

```
wines.shape[1]
```

Out[119]:

```
12
```

wine dimensions

In [120]:

```
wines.ndim
```

Out[120]:

```
2
```

types of wine

In [121]:

```
type(wines)
```

Out[121]:

```
numpy.ndarray
```

datatype of wine data

In [122]:

```
wines.dtype
```

Out[122]:

```
dtype('float64')
```

printing top 5 rows

In [123]:

```
wines[:5]
```

Out[123]:

```
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]])
```

printing value at 3rd row 4th column of wine data

In [124]:

```
wines[2,3]
```

Out[124]:

```
2.3
```

selecting first 3 items in 4th column

In [125]:

```
wines[:3, 3]
```

Out[125]:

```
array([1.9, 2.6, 2.3])
```

1st column

In [153]:

```
wines[:,0]
```

Out[153]:

```
array([7.4, 7.8, 7.8, ..., 6.3, 5.9, 6. ])
```

printing 2nd row

In [127]:

```
wines[1]
```

Out[127]:

```
array([ 7.8   ,  0.88  ,  0.    ,  2.6   ,  0.098 , 25.    , 67.    ,
        0.9968,  3.2   ,  0.68  ,  9.8   ,  5.    ])
```

selecting items from rows 1 to 3 and 5th column

In [128]:

```
wines[1:4, 4]
```

Out[128]:

```
array([0.098, 0.092, 0.075])
```

selecting entire row

In [129]:

```
wines
```

Out[129]:

```
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#showing actual value

In [130]:

```
wines[0,0]
```

Out[130]:

```
7.4
```

updateing value

In [131]:

```
wines[0,0]=100
```

showing updated value

In [132]:

```
wines[0,0]
```

Out[132]:

```
100.0
```

change it back

In [133]:

```
wines[0,0] = 7.4
```

printing after value changed back

In [134]:

```
wines[0,0]
```

Out[134]:

```
7.4
```

1-Dimensional Numpy Arrays

In [135]:

```
#Select 4th row all column values
third_wine = wines[3, :]
#displaying its value
third_wine
```

Out[135]:

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

In [136]:

```
#showing 2nd value
third_wine[1]
```

Out[136]:

```
0.28
```

In [137]:

```
#Convert wine data to integer values and show it
#convert to int
wines.astype(int)
```

Out[137]:

```
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

In [ ]:

```
#Increase wine quality score (output variable) by 10
# check values first
wines[:, 11]
```

In [139]:

```
#increase by 10
wines[:, 11] += 10
```

In [140]:

```
#displaying updated score
wines[:, 11]
```

Out[140]:

```
array([15., 15., 15., ..., 16., 15., 16.])
```

In [141]:

```
#Multiply alcohol of all wine data by 3 times
wines[:, 10] *= 3
```

In [142]:

```
#showing updated alcohol column
wines[:, 10]
```

Out[142]:

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

In [143]:

```
#Add quality column by itself
# It will produce a new array
wines[:, 11] + wines[:, 11]
```

Out[143]:

```
array([30., 30., 30., ..., 32., 30., 32.])
```

In [144]:

```
#Multiply alcohol and wine quality columns. It will perform element wise multiplication
wines[:,10] * wines[:,11]
```

Out[144]:

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

Broadcasting

In [149]:

```
#Add every row of wines data with a random array of values
ran = np.random.rand(12)
```

In [150]:

```
#showing added array
ran
```

Out[150]:

```
array([0.29543068, 0.54591411, 0.98904468, 0.81862454, 0.3508722 ,
       0.28657268, 0.78057341, 0.51224889, 0.0746879 , 0.32514679,
       0.49078379, 0.68149546])
```

In [151]:

```
#adding wines and random added array
wines + ran
```

Out[151]:

```
array([[ 7.69543068,  1.24591411,  0.98904468, ...,  0.88514679,
         9.89078379,  5.68149546],
       [ 8.09543068,  1.42591411,  0.98904468, ...,  1.00514679,
        10.29078379,  5.68149546],
       [ 8.09543068,  1.30591411,  1.02904468, ...,  0.97514679,
        10.29078379,  5.68149546],
       ...,
       [ 6.59543068,  1.05591411,  1.11904468, ...,  1.07514679,
        11.49078379,  6.68149546],
       [ 6.19543068,  1.19091411,  1.10904468, ...,  1.03514679,
        10.69078379,  5.68149546],
       [ 6.29543068,  0.85591411,  1.45904468, ...,  0.98514679,
        11.49078379,  6.68149546]])
```