lab2_DVA_maha18

March 6, 2021

0.1 Red Wine Quality Data Analysis using NumPy Part-II

```
[1]: import numpy as np
```

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

0.1.1 NumPy Aggregation Methods

Find sum of all residual sugar values

```
[71]: wines[:, 3].sum()
```

```
[71]: 4059.55
```

Find sums of every feature value. There are 12 features altogether

```
[72]: wines.sum(axis=0)
```

```
[72]: array([13303.1    ,   843.985  ,   433.29   ,  4059.55   ,   139.859  ,
              25384.     , 74302.     ,  1593.79794,  5294.47   ,  1052.38   ,
              16666.35   ,  9012.     ])
```

Find sum of every row

```
[73]: wines.sum(axis=1)
```

```
[73]: array([ 74.5438 , 123.0548 ,  99.699  , …, 100.48174, 105.21547,
               92.49249])
```

```
[74]: wines.sum(axis=1).shape     # let us check its shape
```

```
[74]: (1599,)
```

What is the maximum residual sugar value in red wines data?

```
[75]: wines[:,3].astype(int)
```

```
[75]: array([1, 2, 2, …, 2, 2, 3])
```

```
[76]: np.max(wines[:,3].astype(int))
```

```
[76]: 15
```

What is the minimum residual sugar value in red wines data?

```
[77]: np.min(wines[:,3].astype(int))
```

```
[77]: 0
```

What is the average residual sugar value in red wines data?

```
[78]: np.mean(wines[:,3])
```

```
[78]: 2.53880550343965
```

What is 25 percentile residual sugar value?

```
[79]: np.percentile(wines[:,3], 25)
```

```
[79]: 1.9
```

What is 75 percentile residual sugar value?

```
[80]: np.percentile(wines[:,3], 75)
```

```
[80]: 2.6
```

Find the average of each feature value

```
[81]: wines.mean(axis=0)
```

```
[81]: array([ 8.31963727,  0.52782051,  0.27097561,  2.5388055 ,  0.08746654,
              15.87492183, 46.46779237,  0.99674668,  3.3111132 ,  0.65814884,
              10.42298311,  5.63602251])
```

0.1.2 NumPy Array Comparisons

Show all wines with quality > 5

```
[82]: wines[:, 11] > 5
```

```
[82]: array([False, False, False, …,  True, False,  True])
```

```
[83]: wines[:, 11] > 7
```

```
[83]: array([False, False, False, …, False, False, False])
```

```
[65]: np.any((wines[:, 11] > 7) == True)          # check if any value is True
```

```
[65]: True
```

Show first 3 rows where wine quality > 7

```
[84]: high_quality = wines[:, 11] > 7
```

```
[85]: high_quality
```

```
[85]: array([False, False, False, …, False, False, False])
```

```
[86]: wines[high_quality, :][:3, :]
```

```
[86]: array([[7.900e+00, 3.500e-01, 4.600e-01, 3.600e+00, 7.800e-02, 1.500e+01,
              3.700e+01, 9.973e-01, 3.350e+00, 8.600e-01, 1.280e+01, 8.000e+00],
             [1.030e+01, 3.200e-01, 4.500e-01, 6.400e+00, 7.300e-02, 5.000e+00,
              1.300e+01, 9.976e-01, 3.230e+00, 8.200e-01, 1.260e+01, 8.000e+00],
             [5.600e+00, 8.500e-01, 5.000e-02, 1.400e+00, 4.500e-02, 1.200e+01,
              8.800e+01, 9.924e-01, 3.560e+00, 8.200e-01, 1.290e+01, 8.000e+00]])
```

Show wines with a lot of alcohol > 10 and high wine quality > 7

```
[87]: high_quality_and_alcohol = (wines[:,10] > 10) & (wines[:,11] > 7)
      wines[high_quality_and_alcohol,10:]           # show only alcohol and wine␣
       ↪quality columns
```

```
[87]: array([[12.8,  8. ],
             [12.6,  8. ],
             [12.9,  8. ],
             [13.4,  8. ],
             [11.7,  8. ],
             [11. ,  8. ],
             [11. ,  8. ],
             [14. ,  8. ],
             [12.7,  8. ],
             [12.5,  8. ],
             [11.8,  8. ],
             [13.1,  8. ],
             [11.7,  8. ],
             [14. ,  8. ],
             [11.3,  8. ],
             [11.4,  8. ]])
```

0.1.3 Combining NumPy Arrays

Combine red wine and white wine data

```
[40]: white_wines = np.genfromtxt("winequality-white.csv", delimiter=";",␣
       ↪skip_header=1)
      white_wines.shape
```

```
[40]: (4898, 12)
```

```
[41]: all_wines = np.vstack((wines, white_wines))       # combine them
      all_wines.shape
```

```
[41]: (6497, 12)
```

Combine using concatenate method

[42]:
```
data2 = np.concatenate((wines, white_wines), axis=0)
```

[43]:
```
data2.shape
```

[43]:
```
(6497, 12)
```

0.1.4 Matrix Operations and Reshape

Transpose wine data

[44]:
```
np.transpose(wines).shape
```

[44]:
```
(12, 1599)
```

Convert wine data into 1D array

[45]:
```
wines.ravel()
wines.ravel().shape
```

[45]:
```
(19188,)
```

Reshape second row of wines into a 2-dimensional array with 2 rows and 6 columns

[46]:
```
wines[1,:].reshape((2,6))
```

[46]:
```
array([[ 7.8   ,  0.88  ,  0.    ,  2.6   ,  0.098 , 25.    ],
       [67.    ,  0.9968,  3.2   ,  0.68  ,  9.8   ,  5.    ]])
```

0.1.5 Sort alcohol column Ascending Order

[49]:
```
sorted_alcohol = np.sort(wines[:, 10])
sorted_alcohol
wines[:, 10].sort()       # In-place sorting
wines[:, 10]
```

[49]:
```
array([ 8.4,  8.4,  8.5, …, 14. , 14. , 14.9])
```

0.1.6 Sort alcohol column Descending Order

[50]:
```
sorted_alcohol_desc = np.sort(wines[:, 10])[::-1]
sorted_alcohol_desc
wines[:, 10]              # original data not modified
```

[50]:
```
array([ 8.4,  8.4,  8.5, …, 14. , 14. , 14.9])
```