

- A dataset for wines will be loaded into the Python environment from sklearn. This data set contains **classification data** to classify the wines into 3 classes based on few features (attributes).

- The dataset contains few NumPy nd-arrays. Important ones are `data` and `target`. The snippet is shown below (zoom-in to see the details):

- For the given data set, the *data* is of size 178x13. There are 178 rows with 13 attributes (excluding the class). The names of these attributes are called *feature names*.

- For the given data set, the *target* is of size 178. There are 178 class values; one for each record. The names of these target are called *target_names*. These are the names of different classes.

[1]

- Many times in the area of Data Science, all attributes except the class is called the X attribute and the class or the target attribute is called the Y class. They can be assigned as follows before processing further:

```
X = wineData.data
Y = wineData.target
```

- The wine data which is stored in the variable *wineData* can be transported into Pandas DataFrames as shown below. First a data frame is created using the *data* part of the data set and then the class is added from the *target*.

```
data = pd.DataFrame(wineData.data, columns=wineData.feature_names)
data['class'] = wineData.target
data
```

alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	class
15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0

- Portion of the data frame is shown above. You would appreciate that it appears more organized.
- The unique count of different classes can be summarised below using the *value_counts()* function. This is helpful if one wants to know how many records are present in the classes.

```
print (data['class'].value_counts())
```

```
1    71
0    59
2    48
Name: class, dtype: int64
```

- There is a function called *groupby()* that can be used to group the data and perform some calculations based on that.

```
data.groupby('class').mean()
```

class	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline
0	13.744746	2.010678	2.455593	17.037288	106.338983	2.840169	2.982373	0.290000	1.899322	5.528305	1.062034	3.92	1065.0
1	12.278732	1.932676	2.244789	20.238028	94.549296	2.258873	2.080845	0.363662	1.630282	3.086620	1.056282	3.40	1050.0
2	13.153750	3.333750	2.437083	21.416667	99.312500	1.678750	0.781458	0.447500	1.153542	7.396250	0.682708	3.17	1185.0

- Note that the grouping is done based on the 3 classes and then the mean for each class per attribute is shown.

Exercise:

Load *breast_cancer* classification data from sklearn and convert that into a Pandas data frame. How many classes are there? How many rows in each class? What are the class names?