CMM201 Week 10: Importing and Wrangling Data

1 Importing and Wrangling Data in Python

1.1 Aims of the Lecture

- Learn how to import numerical data to Python from different sources.
- Understand how to select certain parts of the imported data.

1.2 Example

1.2.1 Loading Data from a Module

- Python has a module called **scikit-learn** or *sklearn* which contains several datasets commonly used in data science and business analytics.
- For this excercise, we will use the **IRIS** database contained in this module.
- This dataset contains the sepal and petal lengths and widths from 150 samples of 3 different types of the iris flower.

Fig 1: Iris dataset.

- Unlike last week, we will **NOT** work with the actual images, but rather with the numerical information extracted from samples.
- First, we need to install **sklearn**:

```
In [ ]: !pip install sklearn
```

- Then, we can load the iris dataset:

```
In [ ]: ## Load iris dataset
        from sklearn import datasets
        iris = datasets.load_iris()
        print(type(iris))
```

- The dataset is contained on a **dictionary-like** structure referred to as **sklearn.utils.Bunch**.
- If you print it, you will see a lot of things contained:

```
In [ ]: print(iris)
```

- Therefore, we need to extract each index of this dictionary into a different variables to understand and analyse them separately.
- First, we will import the data:

```
In [ ]: data = iris['data']
        print(data, type(data), data.shape)
```

- The data is stored in a *numpy array* of 150 rows and 4 columns, each corresponding to the measurements of a flower.
- Then, we will import the headers of the data:

```
In [ ]: header = iris['feature_names']
        print(header, type(header))
```

- **Why do you think the data and the header are stored separately?**
- Afterwards, we will import the **class/target**:

```
In [ ]: target = iris['target']
        print(target, type(target), target.shape)
```

- The class/target is a *numpy array* which contains the **category** of each flowers.
- Each sample is labelled as 0, 1 or 2 instead of the iris type since the labels can be better used as numbers.
- A separate key called **target_names** contains the name corresponding to each numerical label.

```
In [ ]: target_names = iris['target_names']
        print(target_names, type(target_names), target_names.shape)
```

- Finally, just in case you are interested, there is an entry containing the description of the dataset (a string):

```
In [ ]: iris['DESCR']
```

1.2.2 Wrangling Data

- Accessing an individual entry of the dataset (along with its class/target):

```
In [ ]: print(data[0], target[0])
```

- Creating a table for each iris type ("manually")

```
In [ ]: setosa = data[0:50]
        print(setosa, setosa.shape)
```

```
In [ ]: ## Use this cell to create and print versicolor and virginica (with the shape)
```

- Creating a table for each iris type ("automatically")

```
In [ ]: ## In case that data is not in order or you don't want to count,
        ## we can use this alternative:
        import numpy as np
        setosa2 = data[np.where(target==0)]
        print(setosa, setosa.shape)
```

```
In [ ]: ## Verify that we get the same
        setosa == setosa2
```

- Creating a new table with "less" columns (by column number):

```
In [ ]: ## creating a "reduced" table
        ## with ony the first two columns
        data_red1 = data[:,:2]
        print(data_red1,data_red1.shape)
```

```
In [ ]: ## Use this cell to create a new dataset called data_red2
        ## with the last two columns
```

```
In [ ]: ## Use this cell to create a new dataset called data_red3
        ## with the first and the third columns
```

```
In [ ]: ## creating a "reduced" table with ony the first column
        col_0 = data[:,0]
        print(col_0,col_0.shape)
```

- Getting a column by it's name:

```
In [ ]: sepal_length = data[:,header.index('sepal length (cm)')]
        print(sepal_length,sepal_length.shape)
```

1.3 Importing YOUR data

- For the coursework output 2, you will need to import the data from a **.csv** file.
- For instance, the IRIS dataset would look something like this:

	A	B	C	D	E	F
1	flower_id	sepal_leng	sepal_widt	petal_leng	petal_widt	variety
2	45	5.1	3.5	1.4	0.2	0
3	88	4.9	3	1.4	0.2	0
4	100	4.7	3.2	1.3	0.2	0
5	133	4.6	3.1	1.5	0.2	0
6	160	5	3.6	1.4	0.2	0

Fig 1: Iris dataset in csv

- Your datasets will have a **first column** with the id of each entry (**NOT** the same as the row index).
- Your dataset will have the class/target in the **last column**.
- The **first row** contains the header.
- You need to find a pre-existing module that lets you import data from a csv file into a numpy array.
- Try to import the header in a different variable as the data.
- Since the classes/targets are numeric for all datasets, you can leave them on the same numpy array as the data.
- You don't need the target names, just work with the numbers!