From DTU Learn download the scripts for this week called 02450Toolbox_Python_Week02.zip and copy them to the folder where you stored the scripts from last week.

Summary statistics, similarity and visualization with PYTHON

Objective: The overall objective is to get a basic understanding for measures of similarity as well as summary statistics. Upon completing this exercise it is expected that you:

- Understand how to calculate summary statistics such as mean, variance, median, range, covariance and correlation.
- Understand the various measures of similarity such as Jaccard and Cosine similarity and apply similarity measures to query for similar observations.
- Get an understanding of the many ways data can be visualized including histograms, boxplots, and scatter plots.

Material: Lecture notes *"Introduction to Machine Learning and Data Mining"* as well as the files provided from DTU Learn.

2.1 Summary Statistics

The goal is to recap a few basic summary statistics and make sure you are comfortable computing these by hand and your favorite programming language. You may need to look up the definitions in the lecture notes.

2.1.1 Using pen and paper and a `basic` electronic calculator (e.g. on your computer), calculate the (empirical) mean, standard deviation (unbiased), median, and range of the following set of numbers collected in a vector:
$\mathbf{x} = [-0.68, -2.11, 2.39, 0.26, 1.46, 1.33, 1.03, -0.41, -0.33, 0.47]$

2.1.2 Consult the script `ex2_1_2.py` which achieves the same using the built-in functions in your programming language. Script details:

- *Look at the help page of the functions* `mean()`, `std()`, `median()`, `min()` *and* `max()` *of NumPy* `array` *class.*

2.1.3 Modify/extend the script `ex2_1_2.py` to compute both the unbiased and biased estimate of the standard deviation using `std()`. Are the two estimates the same? What is the difference? Script details:

- *Consider what the argument* `ddof=1` *means.*

2.2 Measures of similarity

We will use a subset of handwritten digits dataset transformed to images of size 16×16 pixels, we will attempt to find digits in the data base that are the most similar to a given query face. Each attribute corresponds to a particular pixel in the image where a value of 1 is white and 0 is black. To measure similarity we will consider the following measures: SMC, Jaccard, Cosine, ExtendedJaccard, and Correlation. These measures of similarity are given by:

$$\begin{aligned}
\text{SMC}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\text{Number of matching attribute values}}{\text{Number of attributes}} \\
\text{Jaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\text{Number of 11 matching attributes}}{\text{Number of attributes not involved in 00 matches}} \\
\text{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} \\
\text{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - \boldsymbol{x}^\top \boldsymbol{y}} \\
\text{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\text{cov}(\boldsymbol{x}, \boldsymbol{y})}{\text{std}(\boldsymbol{x})\text{std}(\boldsymbol{y})}
\end{aligned}$$

where $\text{cov}(\boldsymbol{x}, \boldsymbol{y})$ denotes the covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$ and $\text{std}(\boldsymbol{x})$ denotes the standard deviation of $\boldsymbol{x}$.

Notice that the SMC and Jaccard similarity measures are only defined for binary data, i.e., data that takes values of $\{0, 1\}$. As the data we analyze is non-binary, we will transform the data to be binary when calculating these two measures of similarity by setting

$$x_i = \begin{cases} 0 & \text{if } x_i < \text{median}(\boldsymbol{x}) \\ 1 & \text{otherwise.} \end{cases}$$

Note that, depending on the situation, it can be incorrect to encode information in a single binary attribute—and this is true for binary attributes in general. If the meaning behind the value 0 is not specifically non-presence of an attribute, it can be erroneous. For instance, if male/female is encoded in one binary attribute (male: 0, female: 1), some measures will not model the information carried in being male, and a one-of-out-K encoding would be a proper representation.

For the next step, we will look at the USPS handwritten digit database. The digits dataset contains 9298 16x16 handwritten (single) digits images in greyscale.

2.2.1 Inspect and run the script `ex2_2_1.py`. The script loads the digits dataset, computes the similarity between a selected query image and all others, and display the query image, the 5 most similar images, and the 5 least similar images. The value of the used similarity measure is shown below each image. Try changing the query image and the similarity measure and see what happens.

2.2.2 We will investigate how scaling and translation impact the following three similarity measures: Cosine, ExtendedJaccard, and Correlation. Let α and β be two constants. Determine which of the following statements are correct (you may need to use pen and paper). You can check your answers with the script `ex2_2_2.py`

$$\begin{aligned} \text{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Cosine}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\ \text{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \text{ExtendedJaccard}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\ \text{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Correlation}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\ \text{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Cosine}(\beta + \boldsymbol{x}, \boldsymbol{y}) \\ \text{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \text{ExtendedJaccard}(\beta + \boldsymbol{x}, \boldsymbol{y}) \\ \text{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Correlation}(\beta + \boldsymbol{x}, \boldsymbol{y}) \end{aligned}$$

Script details:

- *Type* `help(similarity)` *to learn about the Python function that is used to compute the similarity measures.*
- *Even though a similarity measure is theoretically invariant e.g. to scaling, it might not be exactly invariant numerically.*

2.2.3 Discuss the pratical implications of similarity measures that are translation and/or scaling invariant. You can base the discusison on the image digits dataset but think also of non-image example (e.g. retrieving documents based on the bag-of-words representation from the previous exercise).

2.3 Visualizing and measuring distance in the Fisher's Iris data

In this exercise we will reproduce most of the figures in "Introduction to Data Mining." section 7.1 using the Iris flower dataset that was also introduced in Exercise 1.

2.3.1 The Iris data set is available in the Excel file `Data/iris.xls`. Load the data into Python and generate all the variables described in *Representation of data in Python* using the script `ex2_3_1.py`.

Script details:

- *You can use the package* `xlrd` *which you have used before.*

2.3.2 Inspect and run `ex2_3_2.py`. The script plots a histogram of each of the four attributes in the Iris data.

Script details:

- *You can use the command* `hist` *to plot a histogram.*
- *Use indexing to extract each attribute. For example,* `X[:,m-1]` *extracts the* `m`*'th attribute.*
- *For multiple plots in one figure window you can use the command* `subplot(n,m,i)`*.*

Show on the graph that the petal length is either between 1 and 2 cm. or between 3 and 7 cm. but that no flowers in the data set have a petal length between 2 and 3 cm. Do you think this could be useful to discriminate between the different types of flowers?

2.3.3 Inspect and run `ex2_3_3.py`. The script produces a boxplot of the four attributes in the Iris data as shown in Figure 3.11 in the book.

Script details:

- *Take a look at the function* `boxplot`*.*
- *Type* `help(boxplot)`*" to see how you can adjust the boxplot and add labels.*

This boxplot shows the same information as the histogram in the previous exercise. Discuss the advantages and disadvantages of the two types of plots.

2.3.4 Inspect and run `ex2_3_4.py`. The scripts produces a boxplot for each attribute for each class as shown in Figure 7.2 in the book.

Script details:

- *Use the functions* `subplot()` *and* `boxplot()`*.*
- *The variable* `y` $\in \{0, 1, \ldots, C-1\}$ *contains the class labels. To extract the data objects belonging to, say, class* `c`*, you can use* `y` *to index into* `X` *like this:* `X[(y==c), :]`
- *It is easier to compare the boxplots if they are all on the same axis. To do this, you can use the function* `ylim()`*.*

Show on the graph that all the Iris-setosa in this data set have a petal length between 1 and 2 cm.

2.3.5 Inspect and run `ex2_3_5.py`. The scripts produces a matrix of scatter plots of each combination of two attributes against each other as shown in Figure 7.6 in the book.

Script details:

- *To make a scatter plot, you can use the function* `plot(x,y,s)` *where* `x` *and* `y` *specify the coordinates and* `s` *is a string that specifies the line style and plot symbol, e.g.,* `s='.'` *to make dots.*
- *To extract the data values for the* `m`*'th attribute in the* `c`*'th class, you can write* `X[(y==c),m]`*.*
- *You can use the command* `hold all` *to plot multiple plots on top of each other.*

Say you want to discriminate between the three types of flowers using only the length and width of either sepal or petal. Show on the graph why it would be better to use petal length and width rather than sepal length and width.

2.3.6 Inspect and run `ex2_3_6.py`. The script produces a 3-dimensional scatter plot of three attributes as shown in Figure 7.7 in the book.

Script details:

- *Read more about plotting in 3 dimensions:* `matplotlib.sourceforge.net/mpl_toolkits/mplot3d/tutorial.html`
- *To plot in 3 dimensions you need to import* `matplotlib.pyplot` *as earlier, and additionally* `Axes3D` *from* `mpl_toolkits.mplot3d`.

Try rotating the data. Can you find an angle where the three types of flower are separated in the plot? Discuss the pros and cons of visualizing data in 2 and 3 dimensions, respectavely? How would you plot data that is inherently 4 dimensional or higher?

2.3.7 Use the script `ex2_3_7.py` to plot the data matrix as an image. The data matrix should be standardized to have zero mean and unit standard deviation. What does this plot indicate? (Hint: the class-labels are sorted).

Script details:

- *You can use the function* `imagesc()` *to plot an image.*
- *By default, the image will be smoothed, what is not always desired when you look at the data. Use parameter* `interpolation='None'` *to display raw data.*
- *The function* `zscore()` *can be used to standardize the data matrix.*

You are welcome to try out other plotting methods for the data. Matplotlib online repository is a good source of inspiration:
`https://matplotlib.org/stable/gallery/index.html`

2.4 Visualizing Wine Data

We will in this part of the exercise consider two datasets related to red and white variants of the Portuguese "Vinho Verde" wine, the data has been downloaded from `http://archive.ics.uci.edu/ml/datasets/Wine+Quality`. Only physicochemical and sensory attributes are available, i.e., there is no data about grape types, wine brand, wine selling price, etc. The data has the following attributes:

#	Attribute	Unit
1	Fixed acidity (tartaric)	g/dm^3
2	Volatile acidity (acetic)	g/dm^3
3	Citric acid	g/dm^3
4	Residual sugar	g/dm^3
5	Chlorides	g/dm^3
6	Free sulfur dioxide	mg/dm^3
7	Total sulfur dioxide	mg/dm^3
8	Density	g/cm^3
9	pH	pH
10	Sulphates	g/dm^3
11	Alcohol	% vol.
12	Quality score	0–10

Attributes 1–11 are based on physicochemical tests and attribute 12 on human judging. Later in the course we will attempt to predict whether a wine is a red or white wine based on these attributes and we will also attempt to predict the wine quality. Unfortunately, the data set has

many observations that can be considered outliers and in order to carry out analyses later it is important to remove the corrupt observations.

The aim of this exercise is to use visualization and distance measurements to identify outliers and remove these outliers from the data (we will return to these aspects later in the course as well). It might be necessary to remove some outliers before other outlying observations become visible. Thus, the process of finding and removing outliers is often iterative. The wine data is stored in a Matlab file, `Data/wine.mat`.

2.4.1 Inspect and run the script `ex2_4_1.py`. The scripts loads the data into Python using the `scipy.io.loadmat()` function, as in previous exercises. This dataset contains many observations that can be considered outliers and the visualization tools you have worked with in the previous exercise is used to identify the outliers in the data set. How many outliers are identified by the script? How are the identified outliers removed from the data set?

Script details:

- *You can use your solutions to the previous exercise as a starting point for making your visualizations.*
- *Say you want to find all data objects for which the alcohol percentage (attribute number 11) is not greater than 100%. You can mask them simply as* `mask=(X[:,10]<=100)`.
- *You can use the mask to eliminate the outlier observations (rows of data matrix). For instance you can write* `X=X[mask,:]` *where* `mask` *indicates the data objects that should be maintained. Remember also to remove them from the class index vector,* `y=y[mask,1]` *and to recompute* `N`.

We will later in the course attempt to classify the type of wine (white or red) as well as predict the quality of wine based on the physicochemical tests. Visual inspection of the data can give an indication of the difficulty of these tasks.

2.4.2 Inspect and run the script `ex2_4_2.py`. Are there any of the measurements that seem to be well suited in order to discriminate between red and white wines? What plots are particular useful in order to investigate this?

2.4.3 Does any of the 12 attributes appear to correlate with each other? What plots are well suited to investigate this? Script details:

- *You can validate your findings by computing correlations in Python. Use:*
  `from dtuimldmtools import similarity` *and*
  `similarity(att1, att2, 'Correlation')`

2.4.4 Can you identify any clear relationship between the various physicochemical measurements of the wines and the quality of the wines as rated by human judges?

So far we have used visualization of distrbutions and points to identify outliers implicitly consider the distance from some point to others, but typically one attribute at the time

In this part, we will see how distances measure between the observations in 12 dimensions can be used to can insight about the Wine dataset and problem of disciminating white vs. red wines based on the available attributes.

We consider the Lp distances discussed in the book chapter 4, and specifically the L1 norm, L2 norm (Euclidian) and Linf (max norm). Please look up the definitions of these before attempting this exercise.

Note this is a small experiment at creating slightly different exercise type than we normally run in the course; please let us know if you would like more of these. Also, note that the script uses the `#%%` whcih in VScode result in an interactive session if you run the cell; you can still run the full script as normal if you want.

2.4.5 Consider the script in `ex2_4_3.py` and work your way through the subtask within. You will need to provide some code yourself! The solution will be provided by the TAs). Consult your TA if you get stuck. Script details:

- *You need to replace the the* `raise NotImplementedError()` *with your own code before moving on.* ***Do not use CoPilot, ChatGPT or other AI tools for this - you will be doing a massive disfavour to yourself!***
- *Read about VScode's interactive mode here* `https://code.visualstudio.com/docs/python/jupyter-support-py`

2.5 Tasks for the report

Provide the basic summary statistics of your attributes preferable in a table and consider if attributes are correlated, see also the functions numpy.cov() and numpy.corrcoef(). Specifically address the questions:

1. Include relevant summary statistics of the attributes. Reflect on the values.
2. **Data visualization(s) based on suitable visualization techniques.**
   Touch upon the following aspects, use visualizations when it appears sensible.
   Keep in mind the ACCENT principles and Tufte's guidelines when you visualize the data.
   - Are there issues with extreme values or outliers in the data?
   - How are the individual attributes distributed (e.g. normally distributed)?
   - Are the attributes correlated?

References