Assignment #6

Assignment Overview

In this assignment you will expand your knowledge of lists, tuples, functions, dictionaries, and CSV file manipulation to perform simple **Exploratory Data Analysis (EDA)** of the Iris flower dataset.

Background

The Iris flower dataset is one of the most popular datasets in human history. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant: setosa, virginica, or versicolor. For each sample, 4 attributes are stored: petal length, petal width, sepal length, and sepal width.

See: http://archive.ics.uci.edu/ml/datasets/Iris/ and https://www.kaggle.com/uciml/iris and https://en.wikipedia.org/wiki/Iris_flower_data_set for more.

There are many examples of EDA for this dataset in many languages and frameworks. You are welcome to check them out for ideas, but refrain from using packages, libraries, etc. beyond the scope of this assignment.

Project Specification

This is a group assignment.
Students are encouraged (but not required) to work in groups of max 3 students.

Ideally, the group should be organized around three main tasks / duties:

- Design of the solution ("architect" role)
- Coding of the solution ("developer" role)
- Documentation of the solution ("reporter" role)

You are required to indicate in your submission "who did what" and document the entire process, from sketching the original plans and dividing up the tasks all the way to polishing the interface, testing the solution, and preparing the report.

In this assignment you will expand upon A5 and create a Jupyter notebook, available via Google Colab, that should contain functionality for:

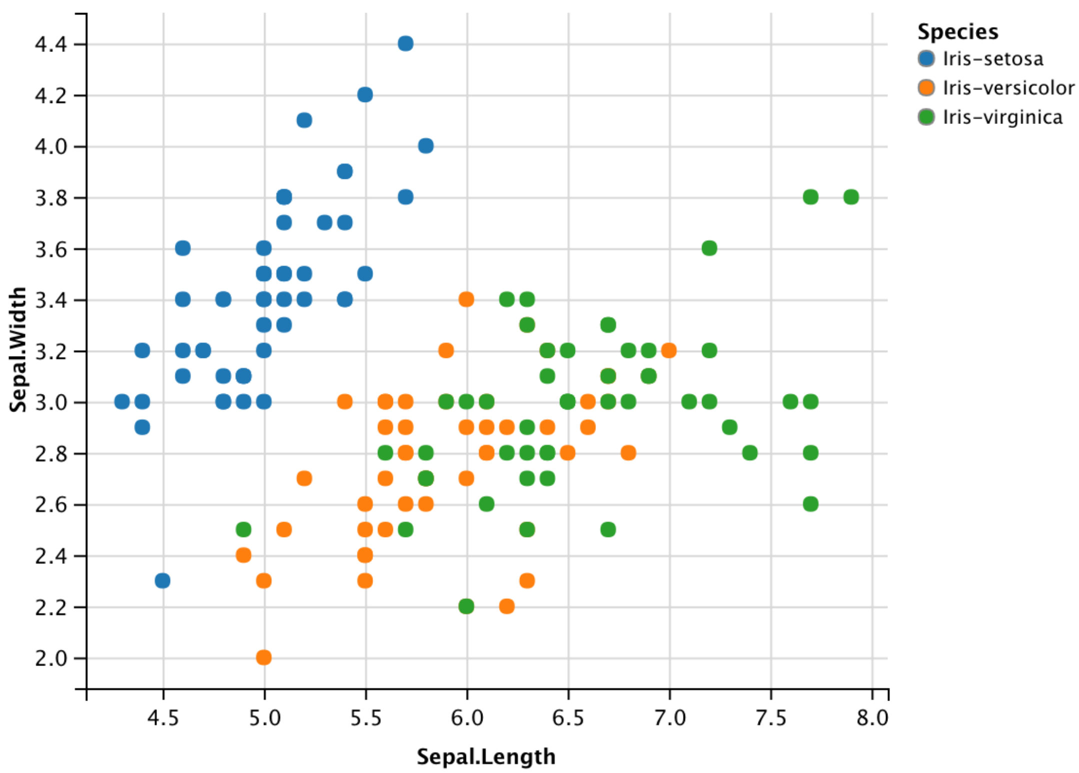
1. **Reading the data** from the `iris.csv` file.

2. **Creating suitable data structures and algorithms for storing each of the four attributes/features** (petal length, petal width, sepal length, and sepal width) for each data point **and computing the minimum, maximum, mean, and standard deviation of each attribute for each species**.

    - This is often known as producing the "summary statistics" of a dataset.
    - According to this[1], you should obtain the values below (note that quartiles are not necessary):

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

3. **Plotting meaningful scatter plots** of the data, two features at a time. Do this at least for sepal width vs. sepal length and petal width vs. petal length (see example below).

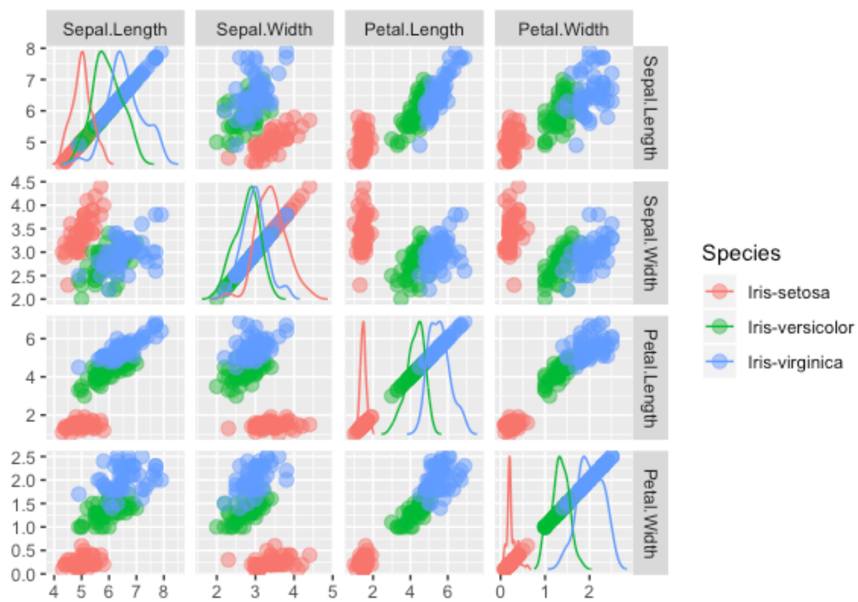


[1] https://www.kaggle.com/bharath25/descriptive-statistics-and-machine-learning-iris

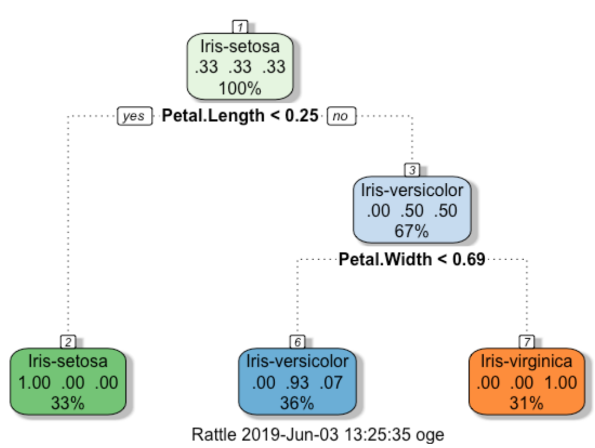
4. **Normalizing the data**, adjusting each feature in the same way across all examples. In this case, we want to limit the range of each feature to the [0..1] interval.
   - The normalized values should be like this (quartiles and median are not necessary):

```
> summary(iris_norm)
  Sepal.Length     Sepal.Width      Petal.Length     Petal.Width
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333
 Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000
 Mean   :0.4287   Mean   :0.4392   Mean   :0.4676   Mean   :0.45778
 3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
```

5. (OPTIONAL, 5% bonus) **Plotting the histograms for the features and their correlations in a single plot.**
   - Here is how they could look like (values below are before normalization):



6. Using the petal length and petal width as features/attributes, **(manually) building a decision tree classifier** (essentially a series of if-elif-else statements in the proper sequence with sensible thresholds for each decision/node).
   - This is a possible solution:



Requirements

You are required to:

1. Break down your solution / notebook into meaningful cells, organized using headings, and with proper text, code, plots, figures, links. etc.
2. Prepare a **Conclusion** cell with a summary of your insights and lessons learned.

You are allowed to use matplotlib, seaborn, pylab, or any other plotting library for Python.

You are **NOT** allowed to use pandas, numpy, scikit-learn, or any other "data science" library for Python.

Deliverables

You must submit (via Canvas):

- The **link**[2] to a Jupyter notebook on Google Colab containing your entire solution. It must include:
    - Header:
        - Team members' names, date, course name + code, assignment number
    - Your source code
    - Results (of multiple runs) + meaningful comments
    - Plots
    - Figures
    - References (including your "sources of inspiration" for the code)
    - Comments (README-like): installation instructions, dependencies, etc.
    - Project notes (describing what my TA and I cannot see by looking at your source code and/or running your program).
        - Examples: design decisions, documented limitations, future improvements, etc.
    - Your **Conclusion** with a summary of your insights and lessons learned.

Bonus opportunities:

5% extra if you implement item (5) above.

[2] When sharing the link to your Google Colab notebook, choose the 'anyone with the link can open it' option, i.e., **don't make it specific to a domain** (such as fau.edu) **or individual** (instructor or TA).