DSCI 510: Principles of Programming for Data Science

Name: Lab Assignment 11
Due Date: November 21th, 2024 at 4pm PT

Deliverable: A python file named **run.py**

Introduction

In this lab, you will work with the Iris dataset, focusing on data pre-processing, descriptive statistics, and data analysis using the Pandas library. This dataset contains measurements for iris flowers, providing a great opportunity to practice data cleaning and exploratory data analysis.

Iris Dataset

For this assignment, we will use the Iris Dataset, which includes the following information about each flower:

- SepalLengthCm: Sepal length in centimeters
- SepalWidthCm: Sepal width in centimeters
- PetalLengthCm: Petal length in centimeters
- PetalWidthCm: Petal width in centimeters
- Species: Species of the iris (Setosa, Versicolor, Virginica)

Data Source:

Download data from the link below (use the big blue "Download (3.7 KB)" button in the top right corner). Unzip the folder with the `iris.data` file inside, this is the file that you are going to use in this lab assignment.

https://archive.ics.uci.edu/ml/datasets/iris

Assignment Tasks

1 Pre-processing (5 points)

Read the data from the filename provided in the function arguments (`iris.data` file). You can use `pd.read_csv()`. Calculate the z-scores for the `SepalLengthCm` and `SepalWidthCm` columns. Remove any samples where the z-score of `SepalLengthCm` or `SepalWidthCm` is less than -2 or greater than 2. This will help filter out outliers in the dataset. After you've filtered the dataset create an "ID" column for the filtered dataset you return that has a range from 1 to the end of the filtered dataset+1.

- **Function Name:** `preprocess_data`

- **Arguments:** (str) `input_filename`
- **Returns:** (DataFrame) `preprocessed_data` with outliers removed.

Example

```
preprocess_data('iris.data')
# Output: A DataFrame without samples where SepalLengthCm
# or SepalWidthCm has a z-score < -2 or > 2.
```

2 Descriptive Statistics (10 points)

Using the pre-processed dataset, please answer the following questions:
When you are returning float values, round them to 1 decimal place. You can use the `round()` function like so: `round(<number>, <ndigits>)`. *Note that you need to* ***return*** *the answer in these functions (do not just print it the answer)*

1. Number of samples for each species.
   Function: `species_count()`
   Return: dict
2. Average sepal length across all samples.
   Function: `average_sepal_length()`
   Return: float
3. Maximum petal width across all samples.
   Function: `max_petal_width()`
   Return: float
4. Minimum petal length across all samples.
   Function: `min_petal_length()`
   Return: float
5. Number of samples with sepal length over 5.0 cm.
   Function: `count_sepal_length_above_5()`
   Return: int

Example Outputs

```
species_count()
# Output: {'Iris-versicolor': 49, ...}

average_sepal_length()
# Output: 5.7

max_petal_width()
# Output: 2.5

min_petal_length()
# Output: 1.0
```

```
count_sepal_length_above_5()
# Output: 107
```

3 Analysis (15 points)

Based on the pre-processed dataset, provide answers to the following questions:
Note that you need to `return` *the answer in these functions (do not just print the answer)*

1. Number of samples with petal length under 2.0 cm.
   Function: `count_petal_length_below_2()`
   Return: int
2. IDs of samples with sepal width above 3.5 cm.
   Function: `get_sepal_width_above_3_5()`
   Return: list (in ascending order)
3. Number of samples for each species with petal width over 1.5 cm.
   Function: `species_count_petal_width_above_1_5()`
   Return: dict
4. IDs of samples with petal length above 6.0 cm for the species "Iris-virginica".
   Function: `get_virginica_petal_length_above_6()`
   Return: list (ascending order)
5. ID of the sample with the largest sepal width.
   Function: `get_largest_sepal_width()`
   Return: int

Example Outputs

```
count_petal_length_below_2()
# Output: 45

get_sepal_width_above_3_5()
# Output: [4, 5, 10, ...]

species_count_petal_width_above_1_5()
# Output: {'Iris-virginica': 41, ...}

get_virginica_petal_length_above_6()
# Output: [101, 103, ...]

get_largest_sepal_width()
# Output: 118
```

Notes

- Use the pre-processed dataset for all statistics and analysis for consistency.
- Make sure that you call the `preprocess_data()` function every time you conduct analysis.