Summarization

•Data Cleaning

1. Loading the Dataset

Load the Iris dataset using Pandas' read_csv() function:

```
column_names = ['id', 'sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species']
iris_data = pd.read_csv('data/Iris.csv', names= column_names, header=0)

iris_data.head()
```

2. Explore the dataset

To get insights about our dataset, we will print some basic information using the built-in functions in pandas

```
print(iris_data.info())
print(iris_data.describe())
```

The info() function is useful to understand the overall structure of the data frame, the number of non-null values in each column, and the memory usage. While the summary statistics provide an overview of numerical features in your dataset.

3. Checking Class Distribution

This is an important step in understanding how the classes are distributed in categorical columns, which is an important task for classification. You can perform this step using the value_counts() function in pandas.

```
print(iris_data['species'].value_counts())
```

4. Removing Missing Values

Since it is evident from the info() method that we have 5 columns with no missing values, we will skip this step. But if you encounter any missing values, use the following command to handle them:

```
iris_data.dropna(inplace=True)
```

5. Removing Duplicates

Duplicates can distort our analysis so we remove them from our dataset. We will first check their existence using the below-mentioned command:

```
duplicate_rows = iris_data.duplicated()
print("Number of duplicate rows:", duplicate_rows.sum())
```

6. Save the cleaned dataset

Save the cleaned dataset to the new CSV file.

```
iris_data.to_csv('cleaned_iris.csv', index=False)
```

Normalization of Float Value Columns

- Normalization

is the process of scaling numerical features to have a mean of 0 and a standard deviation of 1. This process is done to ensure that the features contribute equally to the analysis. We will normalize the float value columns for consistent scaling.

- One-Hot Encoding

For categorical analysis, we will perform one-hot encoding on the species column. This step is performed due to the tendency of Machine Learning algorithms to work better with numerical data. The one-hot encoding process transforms categorical variables into a binary (0 or 1) format.