ANNASAHEB DANGE COLLEGE OF ENGINEERING AND TECHNOLOGY, ASHTA

Artificial Intelligence and Data Science

DATA EXPLORATION AND VISUALIZATION LAB MANUAL

Subject Code: 1ADPC302
A.Y. 2024-25

Course Coordinator

Prof. Akshay K. Mane

Asst. Professor, AI & DS

Sant Dnyaneshwer Shikshan Sanstha's

ANNASAHEB DANGE COLLEGE OF ENGINEERING AND TECHNOLOGY, ASHTA

(An Autonomous Institute affiliated to Shivaji University, Kolhapur)

Artificial Intelligence and Data Science Department

Vision

To produce exclusive software professionals who shall effectively contribute to the leveraging field of Artificial Intelligence and Data Science.

Mission

We will achieve our Vision by:

- Providing Excellent Infrastructure facilitating the students and faculty members with recent trends and technologies.
- Imparting High-Quality Education to the students also instigating them with ethical and moral values.
- Enabling students to enhance their research abilities to address various society-oriented issues through Innovative projects
- Collaborating with various Industries to make students industry ready

PO Programme Outcomes

Learners / Students of Artificial Intelligence and Data Science Engineering Programme Graduates are expected to have attained & will be able to:

01. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

02. Problem analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

03. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

04. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

05. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

06. The engineer and society: Demonstrate understanding of contemporary knowledge of engineering to assess societal, health, safety, legal and cultural issues and the consequent responsibilities.

07. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

08. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

09. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities, write effective reports, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the ability to engage in independent and life-long learning in the broadest context of technological change.

Sant Dnyaneshwer Shikshan Sanstha's

ANNASAHEB DANGE COLLEGE OF ENGINEERING AND TECHNOLOGY, ASHTA

(An Autonomous Institute affiliated to Shivaji University, Kolhapur)

Artificial Intelligence and Data Science Department

PEO Program Educational Objectives

PEO 1: Ability to understand, apply, analyse, design models and applications for all the real-world scenarios related to Artificial Intelligence and Data Science. (PO 1,2,3)
PEO 2: Practice engineering in a broader aspect and exhibit professional leadership qualities in their field. (PO 4,5,6)
PEO 3: Enhancing technological competence to withstand the challenges in the volatile IT industry. (PO 7,8,9)
PEO 4: To be committed in Life-long research and Learning activities that supports societal development. (PO 10,11,12)

PSO Program Specific Outcomes:

PSO 1: - Practically Applying the skills & knowledge acquired to various Inter/Multi/Trans disciplinary problem areas. (PEO 1,2)

PSO 2: - Enrich Leading abilities in the field of Artificial Intelligence and Data Science to qualify for employability. (PEO 3,4)

General Lab Instructions

Do

1. Join the lab session few minutes before the start time.
2. Maintain proper environment of the lab.
3. Go through the theory behind the experiments before attending each lab.
4. Complete your assignment within the given period of time.

Don'ts

1. Don't use internet for solving assignment problems.
2. Don't share your codes with other students. You will get zero marks if your codes are found copied from any online resource.
3. Don't use electronic gadgets (e.g., Mobile phone, Tab, etc.) during the lab hours. You should only use one system for solving assignment problems.

Lab Equipment

Following hardware and software are necessary to perform the experiments in data science lab.

Programming Languages:

1. Python
2. R

Programming Editor:

1. Visual Studio Code
2. Jupyter
3. Google Collab

Hardware:

1. A computer that can execute Python and ML Programs.

Operating system:

1. Windows

Content

EXPERIMENT NO. 1

Title:

Distribution of Variables and Numerical Summaries of Level and Spread on the Iris Dataset

Objective:

To analyze the distribution of variables and calculate the numerical summaries (mean, median, standard deviation, variance, and range) for each feature in the Iris dataset.

Brief Theory:

Introduction: The Iris dataset is a classic and widely used dataset in the field of machine learning and statistics. It consists of 150 observations of iris flowers, with each observation having four numerical features: sepal length, sepal width, petal length, and petal width. Additionally, each observation is classified into one of three species of iris: Iris setosa, Iris versicolor, and Iris virginica.

Distribution of Variables

The distribution of a variable shows how frequently each value occurs. It provides insights into the shape, central tendency, and variability of the data. Common ways to visualize distributions include histograms, box plots, and density plots.

Numerical Summaries

Numerical summaries provide concise statistical descriptions of data. They help in understanding the central tendency, spread, and overall nature of the data. The key numerical summaries are:

- **Mean**: The average value of the data.
- **Median**: The middle value when the data is sorted.
- **Standard Deviation**: A measure of the amount of variation or dispersion in the data.
- **Variance**: The average of the squared differences from the mean.
- **Range**: The difference between the maximum and minimum values.

Steps: -

Step 1: Import Necessary Libraries

First, we need to import the libraries required for data manipulation and visualization.

Code: -

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
```

Step 2: Load the Iris Dataset

We will load the Iris dataset using the load_iris function from scikit-learn and convert it into a pandas DataFrame.

Code: -

```
# Load the iris dataset
iris_data = load_iris()
iris = pd.DataFrame(data=iris_data.data, columns=iris_data.feature_names)
iris['species'] = iris_data.target
```

load_iris() loads the dataset, which is then converted to a DataFrame with appropriate column names.

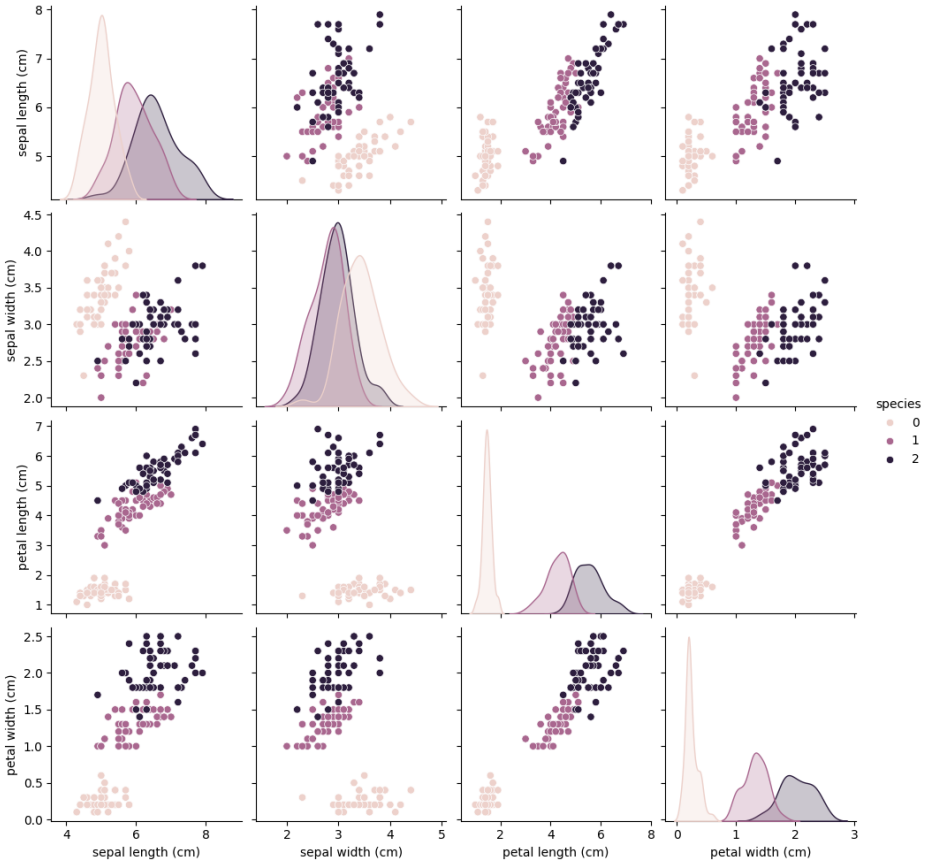
Step 3: Visualize the Distribution of Variables

We will use pair plots to visualize the distribution and relationship between the numerical features.

Code: -

```
# Distribution of Variables
sns.pairplot(iris, hue='species')
```

plt.show()



pairplot creates scatter plots for each pair of features and histograms for individual features, color-coded by species. This helps in identifying patterns and differences between species.

Step 4: Calculate Numerical Summaries

We will calculate the mean, median, standard deviation, variance, and range for each numerical feature.

Code: -

Numerical Summaries

numerical_summaries = iris.describe()
print(numerical_summaries)

```
# Numerical Summaries
numerical_summaries = iris.describe()
print(numerical_summaries)
```

```
       sepal length (cm)  sepal width (cm)  petal length (cm)  \
count         150.000000        150.000000         150.000000
mean            5.843333          3.057333           3.758000
std             0.828066          0.435866           1.765298
min             4.300000          2.000000           1.000000
25%             5.100000          2.800000           1.600000
50%             5.800000          3.000000           4.350000
75%             6.400000          3.300000           5.100000
max             7.900000          4.400000           6.900000

       petal width (cm)     species
count        150.000000  150.000000
mean           1.199333    1.000000
std            0.762238    0.819232
min            0.100000    0.000000
25%            0.300000    0.000000
50%            1.300000    1.000000
75%            1.800000    2.000000
max            2.500000    2.000000
```

describe() generates summary statistics including count, mean, standard deviation, min, 25th percentile, median (50th percentile), 75th percentile, and max.

Step 5: Interpretation

1. **Distribution Visualization**: The pair plot provides scatter plots of all pairs of features and histograms of individual features, colored by species. This helps in identifying patterns, correlations, and differences between species.
2. **Numerical Summaries**: The describe() method outputs the count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum for each feature.

Conclusion:

In this experiment, we visualized the distribution of variables and calculated numerical summaries for the Iris dataset. The pair plots revealed distinct clusters for different species, and the numerical summaries provided insights into the central tendency and spread of each feature. This foundational analysis is essential for further statistical and machine learning applications.

Practice Questions:

1. Plot histograms for each feature to analyze the distribution of the data. What do the shapes of the histograms suggest about the distribution?
2. Create boxplots for each feature in the Iris dataset. How do the boxplots help identify any outliers in the dataset?
3. Calculate the mean for each feature (sepal length, sepal width, petal length, and petal width) in the Iris dataset.
4. Write a Python script to load the Iris dataset using `pandas`. How do you display the first few rows of the dataset?

Expected Oral Questions

1. What is the Iris dataset, and why is it commonly used in data science?
2. Explain the importance of calculating descriptive statistics (mean, median, standard deviation, etc.) when analyzing a dataset?
3. Why might you choose to plot histograms for the features in the Iris dataset?
4. Why might you standardize features, and how does standardization affect the distribution of data?
5. Why is it important to understand the distribution of each feature in a dataset?

FAQs in Interviews

Q: Can you explain the purpose of calculating descriptive statistics (mean, median, standard deviation, variance) when analyzing a dataset like Iris?

A: Descriptive statistics provide a summary of the data, helping us understand the central tendency (mean, median), spread (standard deviation, variance), and overall range. These statistics give us a quick overview of the data, revealing patterns, detecting outliers, and identifying the distribution characteristics of each feature.

Q: What are the key differences between mean and median, and in what scenarios might one be more informative than the other?

A: The mean is the average of all data points, while the median is the middle value when data is ordered. The median is more informative when the data is skewed or contains outliers, as it is less affected by extreme values, providing a better central tendency measure in such cases.

Q: How would you explain the significance of the range and standard deviation in a dataset?

A: The range provides the difference between the maximum and minimum values, giving a sense of the total spread. Standard deviation measures the average distance of each data point from the mean, indicating how dispersed the data is. Together, they help in understanding the variability and consistency of the data

Q: Describe how you would load the Iris dataset in Python and calculate basic descriptive statistics for each feature.

A: You can load the Iris dataset using `pandas` or directly from `sklearn.datasets`. After loading the data into a DataFrame, you can use `df.describe()` to get an overview of the basic descriptive statistics, including mean, median, standard deviation, etc. You can also calculate specific statistics using methods like `df.mean()`, `df.median()`, and `df.std()`.

```
import pandas as pd
from sklearn.datasets import load_iris

# Load the dataset
iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

# Calculate descriptive statistics
print(df.describe())
print("Mean:", df.mean())
print("Median:", df.median())
print("Standard Deviation:", df.std())
```

EXPERIMENT NO. 2

Title:

Scaling and Standardizing the Iris Dataset

Objective:

To apply scaling and standardizing techniques to the features of the Iris dataset to ensure they have appropriate ranges and distributions for further analysis or machine learning algorithms.

Brief Theory:

Introduction: In data preprocessing, scaling and standardizing are essential steps. They help in normalizing the range of independent variables or features of data.

Scaling: Scaling transforms the data to fit within a specific range, usually 0 to 1. This is especially useful when dealing with features of different units or magnitudes.

Standardizing: Standardizing transforms the data to have a mean of 0 and a standard deviation of 1. This process ensures that each feature contributes equally to the analysis, which is particularly important for algorithms that assume normally distributed data (e.g., PCA, logistic regression).

Why Scaling and Standardizing?

Improves convergence speed: For gradient-based algorithms, scaling can improve the convergence speed.

Equal Contribution: Ensures all features contribute equally to the result.

Handles different units: Aligns the units of different features, making the comparison fair.

Steps: -

Step 1: Import Necessary Libraries

First, import the libraries required for data manipulation and preprocessing.

Code: -

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.datasets import load_iris
```

Step 2: Load the Iris Dataset

Load the Iris dataset using the load_iris function from scikit-learn and convert it into a pandas DataFrame.

Code: -

```
# Load the iris dataset
iris_data = load_iris()
iris = pd.DataFrame(data=iris_data.data, columns=iris_data.feature_names)
iris['species'] = iris_data.target
```

The load_iris() function loads the dataset, which is then converted to a DataFrame with appropriate column names.

Step 3: Standardize the Dataset

Standardize the numerical features to have a mean of 0 and a standard deviation of 1.

Code: -

```
# Standardizing the dataset
scaler = StandardScaler()
iris_scaled = pd.DataFrame(scaler.fit_transform(iris.iloc[:, :-1]), columns=iris.columns[:-1])

# Adding the species column back to the standardized dataset
iris_scaled['species'] = iris['species']
```

StandardScaler is used to standardize the features. The fit_transform() method computes the mean and standard deviation, and then scales the data accordingly.

Step 4: Scale the Dataset

Apply Min-Max scaling to the numerical features, transforming them to a range between 0 and 1.

Code: -

```
# Min-Max Scaling the dataset
minmax_scaler = MinMaxScaler()
iris_minmax_scaled = pd.DataFrame(minmax_scaler.fit_transform(iris.iloc[:, :-1]),
columns=iris.columns[:-1])

# Adding the species column back to the min-max scaled dataset
iris_minmax_scaled['species'] = iris['species']
```

MinMaxScaler is used to scale the features to a range between 0 and 1. The fit_transform() method computes the minimum and maximum values and scales the data accordingly.

Step 5: Comparison of Original, Standardized, and Scaled Data

Compare the original, standardized, and scaled data to observe the differences.

Code: -

```
# Display original, standardized, and scaled data
print("Original Data:")
print(iris.head())

print("\nStandardized Data:")
print(iris_scaled.head())

print("\nMin-Max Scaled Data:")
print(iris_minmax_scaled.head())
```

Output:-

```
Original Data:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

   species
0        0
1        0
2        0
3        0
4        0

Standardized Data:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0          -0.900681          1.019004          -1.340227         -1.315444
1          -1.143017         -0.131979          -1.340227         -1.315444
2          -1.385353          0.328414          -1.397064         -1.315444
3          -1.506521          0.098217          -1.283389         -1.315444
4          -1.021849          1.249201          -1.340227         -1.315444

   species
0        0
1        0
2        0
3        0
4        0

Min-Max Scaled Data:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0           0.222222          0.625000           0.067797          0.041667
1           0.166667          0.416667           0.067797          0.041667
2           0.111111          0.500000           0.050847          0.041667
3           0.083333          0.458333           0.084746          0.041667
4           0.194444          0.666667           0.067797          0.041667

   species
0        0
1        0
2        0
3        0
4        0
```

Original Data: The raw values of the features.

Standardized Data: Features have a mean of 0 and a standard deviation of 1.

Min-Max Scaled Data: Features are scaled to a range between 0 and 1.

Conclusion:

In this experiment, we successfully applied scaling and standardizing techniques to the Iris dataset. Standardizing ensured that each feature had a mean of 0 and a standard deviation of 1, while Min-Max scaling transformed the features to a range between 0 and 1. These preprocessing steps are crucial for ensuring that features contribute equally and appropriately to machine learning models and further statistical analysis

Practice Questions:

1. Write a Python script to load the Iris dataset using `sklearn.datasets.load_iris`. Explore the dataset by printing the feature names, target names, and the first five rows of the dataset.
2. Apply Min-Max scaling to the features of the training set to transform the data to a range between 0 and 1. Then, apply the same transformation to the test set.
3. Apply standardization to the features of the training set so that they have a mean of 0 and a standard deviation of 1. Then, apply the same transformation to the test set.
4. After applying Min-Max scaling and standardization to the training set, compare the mean and standard deviation of the features before and after applying these techniques. Discuss the effects on the dataset.

Expected Oral Questions

1. Can you describe the structure of the Iris dataset? What are the features and target variables?
2. Why is it important to explore the dataset before applying any scaling or standardization?
3. What is the difference between scaling and standardization?
4. Why is it necessary to scale or standardize features before applying machine learning algorithms?
5. What could happen if you skip scaling or standardization for certain algorithms?
6. How do you apply Min-Max scaling in Python? Could you describe the process?
7. How do you apply standardization in Python? What are the key steps?
8. After scaling or standardizing, how can you verify that the transformation was applied correctly?

FAQs in Interviews

Q: What is the difference between Min-Max scaling and standardization? When would you use each technique?

A: Min-Max scaling transforms the data by rescaling the features to a specific range, typically [0, 1]. The formula used is:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

This technique is useful when you know the exact range of the data, or when you want to preserve the relationships between the data points while ensuring they fall within a specific range.

Standardization, on the other hand, transforms the data to have a mean of 0 and a standard deviation of 1. The formula is:

$$X_{standardized} = \frac{X - \mu}{\sigma}$$

This technique is useful when the data follows a normal distribution, or when you are using algorithms that assume or benefit from normally distributed data (e.g., logistic regression, SVM).

Use Cases:

- Use **Min-Max scaling** when the algorithm you are using (e.g., KNN, neural networks) is sensitive to the magnitude of the data.
- Use **standardization** when the data is normally distributed or when you are using algorithms that rely on the distribution of the data.

Q: Why is it important to apply the same scaling or standardization to both the training and test sets?

A: It is crucial to apply the same scaling or standardization to both the training and test sets to ensure that the model is evaluated on a consistent scale. If you fit the scaler separately to the test set, it might alter the distribution, leading to inaccurate or biased model evaluation. The correct approach is to fit the scaler on the training set and then use the same transformation parameters (mean, standard deviation, min, max) to scale the test set. This maintains the integrity of the model's performance evaluation and prevents data leakage.