Exploratory Data Analysis (EDA) Assignment

Submitted by: Sasha S

Institution: Purdue University

Date: 30/6/25

---

1. Introduction

Exploratory Data Analysis (EDA) is a critical step in data analysis that involves summarizing the main characteristics of a dataset, often using visual methods. This assignment focuses on performing EDA on the Iris dataset to uncover insights, detect patterns, and prepare the data for further analysis.

Objectives:

- Load and inspect the dataset.
- Clean and preprocess the data.
- Perform univariate and multivariate analysis.
- Identify outliers and anomalies.
- Derive meaningful insights.

---

2. Dataset Description

Dataset Source: Kaggle

Dataset Name: Iris Dataset

Features:

- Numerical Features: sepal_length,sepal_width,petal_length,petal_width

- Categorical Features: None
- Target Variable (if applicable): species : categorical: Iris-setosa, Iris-versicolor, Iris-virginica

Dataset Size:

- Rows: 150
- Columns:5

---

3. Methodology

Tools & Libraries Used:

- Python (Pandas, NumPy, Matplotlib, Seaborn)
- Jupyter Notebook

Steps Performed:

1. Data Loading & Initial Inspection
2. Data Cleaning (Handling Missing Values, Duplicates)
3. Univariate Analysis (Distributions, Count Plots)
4. Bivariate/Multivariate Analysis (Correlation, Scatter Plots)
5. Outlier Detection & Treatment
6. Feature Engineering (if applicable)
7. Key Insights & Conclusions

---

4. Implementation & Results

4.1 Data Loading & Initial Inspection

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
```

```
df = pd.read_csv("iris.data.csv", header=None)
df.columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
'species']


# Display first 5 rows
print(df.head())

# Check dataset shape
print(f"Dataset Shape: {df.shape}") # will print (150,5)

# Check data types and missing values
print(df.info()) # in our case is none

# Summary statistics
print(df.describe())  #prints statistics for each column
```

Observations:

- The dataset contains **150 rows** and **5 columns**.
- Missing values found in columns: None.
- Data types: **Numerical:** sepal_length, sepal_width, petal_length, petal_width ,**Categorical:** species

---

4.2 Data Cleaning

Handling Missing Values

```
# Fill missing numerical values with mean

df['sepal_length'].fillna(df['sepal_length'].mean(), inplace=True)

df['sepal_width'].fillna(df['sepal_width'].mean(), inplace=True)

df['petal_length'].fillna(df['petal_length'].mean(), inplace=True)

df['petal_width'].fillna(df['petal_width'].mean(), inplace=True)
```

```
# Fill missing categorical values with mode

df['species'].fillna(df['species'].mode()[0], inplace=True)

# Remove duplicate rows

df.drop_duplicates(inplace=True)

print("The missing values: " ,df.isnull().sum()) # Check for any remaining missing values

print(df.shape) # Check the shape of the DataFrame after cleaning
```

Observations:

- 0 missing values were filled.
- 3 duplicate rows were removed.

4.3 Univariate Analysis

Sepal Length Distribution

plt.hist(df['sepal_length'], bins=20)

plt.title("Sepal Length Distribution")

plt.xlabel("Sepal Length")

plt.ylabel("Frequency")

plt.show()

Sepal Width Distribution

```
plt.hist(df['sepal_width'], bins=20)

plt.title("Sepal Width Distribution")

plt.xlabel("Sepal Width")

plt.ylabel("Frequency")

plt.show()


# Petal Length Distribution

plt.hist(df['petal_length'], bins=20)

plt.title("Petal Length Distribution")

plt.xlabel("Petal Length")

plt.ylabel("Frequency")

plt.show()


# Petal Width Distribution

plt.hist(df['petal_width'], bins=20)

plt.title("Petal Width Distribution")

plt.xlabel("Petal Width")

plt.ylabel("Frequency")
```

plt.show()

Observation:

- Most petal lengths are between 1.0 and 2.0 cm .
- Sepal widths are mostly between 2.5 and 3.5 cm.
- Sepal lengths are mostly between 6 and 7 cm
- Petal widths are mostly between 0 and 0.5 cm.

Categorical Features (Count Plot)

```
sns.countplot(x='species', data=df)

plt.title("Species Distribution")

plt.xlabel("Species")

plt.ylabel("Count")

plt.show()
```

Observation:

- **Iris-setosa**: 48 samples
- **Iris-versicolor**: 50 samples
- **Iris-virginica**: 49 samples
- The dataset is **almost perfectly balanced**, with an equal number of samples across all three Iris species.

4.4 Bivariate/Multivariate Analysis

Scatter Plot (Numerical vs Numerical)

```
sns.scatterplot(x='sepal_length', y='petal_length', hue='species', data=df)

plt.title("Sepal Length vs Petal Length")

plt.xlabel("Sepal Length")

plt.ylabel("Petal Length")

plt.show()
```

Observation:

- Petal length tends to increase with sepal length. The three species form distinct clusters, especially **Iris-setosa**, which is clearly separated.

Reg Plot

```
import seaborn as sns

import matplotlib.pyplot as plt



# Sepal Length vs Petal Length with regression line

sns.regplot(x='sepal_length', y='petal_length', data=df)

plt.title("Sepal Length vs Petal Length (Line of Best Fit)")

plt.xlabel("Sepal Length")

plt.ylabel("Petal Length")

plt.show()



# Petal Length vs Petal Width with regression line
```

```
sns.regplot(x='petal_length', y='petal_width', data=df)

plt.title("Petal Length vs Petal Width (Line of Best Fit)")

plt.xlabel("Petal Length")

plt.ylabel("Petal Width")

plt.show()
```

Observation:

- **Sepal Length vs Petal Length:** Shows a clear upward trend. Longer sepals are associated with longer petals.
- **Petal Length vs Petal Width:** Strong linear relationship ,one increases with the other.

4.5 Outlier Detection

Boxplot Analysis

```
sns.boxplot(x=df['sepal_length'])

plt.title("Sepal Length Distribution with Outliers")

plt.show()



sns.boxplot(x=df['sepal_width'])

plt.title("Sepal Width Distribution with Outliers")

plt.show()
```

```
sns.boxplot(x=df['petal_length'])

plt.title("Petal Length Distribution with Outliers")

plt.show()



sns.boxplot(x=df['petal_width'])

plt.title("Petal Width Distribution with Outliers")

plt.show()



(No dots outside any of the box plots)
```

Observation:

- Outliers not detected in any column

Outlier Treatment (IQR Method)

```
Not aplicable
```

Observation:

- No outliers were removed.

---

5. Key Insights

1. **The dataset is clean and well-structured, with 150 flower samples evenly split across three species.**

There were no missing values, and only 3 duplicate rows were removed during cleaning.

2. **Petal length and petal width show a strong positive correlation, meaning they increase together.**

   This relationship helps visually separate the species when plotted, especially in scatter plots and regression plots

3. **No visible outliers were found in any of the numerical features based on boxplots.**

   This confirms the dataset is ideal for analysis and doesn't need further cleaning.

---

6. Conclusion

This EDA assignment helped in understanding the iris dataset by identifying trends, cleaning inconsistencies, and visualizing relationships. Further steps could include predictive modeling or advanced statistical analysis.

---

7. References

- [https://www.kaggle.com/datasets/vikrishnan/iris-dataset?resource=download]
- [Matplotlib Documentation]
- [Seaborn Tutorials]