Experiment 3: Exploratory Data Analysis (EDA) using Seaborn

Title:

Exploratory Data Analysis (EDA) using Seaborn in Python

Aim:

To perform Exploratory Data Analysis (EDA) using the Seaborn library for understanding dataset structure, relationships, and patterns through visualizations.

Objectives:

- Understand the importance of EDA in the ML pipeline.
- Use Seaborn to visualize data distributions and relationships.
- Identify outliers, correlations, and trends in data.
- Gain insights that help in data preprocessing and model selection.

Theory:

Exploratory Data Analysis (EDA) is the process of examining datasets to summarize their main characteristics using both **statistical** and **visual** methods.
EDA helps to:

- Detect missing or inconsistent data.
- Identify patterns and correlations.
- Decide which features are relevant for modeling.

Seaborn is a Python data visualization library built on top of **matplotlib**, providing a high-level interface for attractive and informative statistical graphics.

Common Seaborn Plot Types:

Plot Type	Purpose
`distplot()` / `histplot()`	Show data distribution
`boxplot()`	Detect outliers and compare categories
`pairplot()`	Visualize pairwise relationships
`heatmap()`	Show correlation between features
`countplot()`	Show frequency of categorical variables
`scatterplot()`	Show relationship between two numeric features

Algorithm / Steps:

1. Import required libraries (pandas, seaborn, matplotlib).
2. Load a sample dataset (e.g., Iris or Titanic).
3. Display dataset information and summary statistics.
4. Use Seaborn to plot:
   - Distributions
   - Boxplots
   - Pairplots
   - Heatmaps
5. Observe and interpret the graphs.
6. Draw conclusions based on visual findings.

Sample Python Code:

```
# Experiment 3: Exploratory Data Analysis using Seaborn

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

# Load a sample dataset
df = sns.load_dataset('iris')

# 1. Display basic information
print("Dataset Info:")
print(df.info())
print("\nSummary Statistics:")
print(df.describe())

# 2. Distribution plot of one feature
sns.histplot(df['sepal_length'], kde=True, color='skyblue')
plt.title("Distribution of Sepal Length")
plt.show()

# 3. Boxplot for outlier detection
sns.boxplot(x='species', y='sepal_width', data=df, palette='Set2')
plt.title("Boxplot of Sepal Width by Species")
```

```
plt.show()

# 4. Pairplot to visualize relationships between features
sns.pairplot(df, hue='species', palette='husl')
plt.suptitle("Pairplot of Iris Dataset", y=1.02)
plt.show()

# 5. Correlation Heatmap
corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation Heatmap")
plt.show()
```

Expected Output:

1. **Histogram** showing the distribution of *sepal_length*.
2. **Boxplot** comparing *sepal_width* across species — helps detect outliers.
3. **Pairplot** showing pairwise relationships between all numerical features.
4. **Heatmap** showing correlation coefficients between variables.

Sample Insights:

- Sepal length and petal length are positively correlated.
- Some species (e.g., *setosa*) have distinctly different feature distributions.
- Few outliers exist in *sepal_width*.

Result:

The experiment successfully demonstrated how to perform Exploratory Data Analysis using Seaborn. Students learned how to visualize data distribution, detect outliers, and identify relationships between variables.

Viva Questions:

1. What is the purpose of EDA?
2. What is the difference between histogram and boxplot?
3. How can you detect outliers visually?
4. What does a correlation heatmap represent?
5. What function is used in Seaborn to show pairwise relationships?

Additional Practice (Optional):

Use the **Titanic dataset** (`sns.load_dataset('titanic')`) and perform:

- Countplot of passenger class vs survival.
- Heatmap for missing values (`sns.heatmap(df.isnull())`).
- Boxplot of age vs class.