**Exploratory Data Analysis (EDA) on a Public Dataset**

TASK : Choose a simple, publicly available dataset (e.g., Iris, Titanic, or a dataset from Kaggle).

Clean the data, handle missing values, and perform basic statistical analysis.

Visualize data distributions, correlations, and key patterns using chart

and graphs.

Tools: Python (Pandas, Matplotlib, Seaborn), Jupyter Notebook.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Configure visualizations

sns.set(style="whitegrid")

# Load the dataset

iris = sns.load\_dataset('iris')

# Display the first few rows

iris.head()

# Shape of the dataset

print(f"Dataset contains {iris.shape[0]} rows and {iris.shape[1]} columns.")

# Basic information about the dataset

iris.info()

# Check for missing values

print("Missing values per column:")

print(iris.isnull().sum())

# Summary statistics

iris.describe()

plt.figure(figsize=(8, 6))

sns.histplot(iris['sepal\_length'], kde=True, color="blue")

plt.title("Distribution of Sepal Length")

plt.xlabel("Sepal Length")

plt.ylabel("Frequency")

plt.show()

plt.figure(figsize=(8, 6))

sns.countplot(data=iris, x='species', palette='Set2')

plt.title("Count of Each Species")

plt.xlabel("Species")

plt.ylabel("Count")

plt.show()

plt.figure(figsize=(8, 6))

sns.boxplot(data=iris, x='species', y='sepal\_width', palette='coolwarm')

plt.title("Boxplot of Sepal Width by Species")

plt.xlabel("Species")

plt.ylabel("Sepal Width")

plt.show()

plt.figure(figsize=(8, 6))

sns.scatterplot(data=iris, x='petal\_length', y='petal\_width', hue='species', palette='Set1')

plt.title("Petal Length vs Petal Width by Species")

plt.xlabel("Petal Length")

plt.ylabel("Petal Width")

plt.show()

sns.pairplot(iris, hue='species', palette='Set2', diag\_kind='kde', height=2)

plt.show()

# Compute correlation matrix

correlation\_matrix = iris.iloc[:, :-1].corr()

# Plot the heatmap

plt.figure(figsize=(8, 6))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

plt.title("Correlation Matrix")

plt.show()

iris.to\_csv('cleaned\_iris.csv', index=False)











