PRACTICE 03 : STEP 2. DATA PREPROCESSING

"EXPLORATORY DATA ANALYSIS"

1. Aim of the practice

- Data understanding with statistics
- Data understanding with visualization

2. Data understanding with statistics

From the data of the heart.csv.

2.1. Explore the top 20 rows and the last 20 rows

2.2. Getting each attribute's Data Type by df.dtypes

2.3. Display the statistical summary of the data by df.describe(): Count, Mean, Standard Deviation, Minimum Value, Maximum value, 25th Percentile, Median Percentile, 75th Percentile.

2.4. Reviewing Class Distribution according to the "age" by using: count_class = df.groupby("classname").size()

2.5. Reviewing Correlation between Attributes by: corr_matrix=df.corr()

3. Data understanding with visualization

```
import seaborn as sns
import matplotlib.pyplot as plt
```

3.1. Univariate Plots

3.1.1. Histogram: Use df.hist(), sns.histplot(df["Columnname"]), then

```
graph=sns.countplot(x='cp', data=df)
graph.set_xticklabels(['cp 0','cp 1', 'cp 2', 'cp 3' ])
graph.set_title('Title')
```

3.1.2. Density Plots to visualize the distribution of continuous numerical data:

```
df.plot(kind="density", subplots=True, layout=(4, 4), sharex=False, figsize=(15, 12))
plt.tight_layout()  # Adjust layout for better visibility
plt.show()pyplot.show()
```

3.2. Multivariate Plots

3.2.1. Visualizing the Correlation Matrix with a Heatmap

```
corr_matrix=df.corr()
import seaborn as sns
import matplotlib.pyplot as plt

# Set figure size
plt.figure(figsize=(10, 8))

# Create heatmap
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

# Show plot
plt.title("Correlation Matrix Heatmap")
plt.show()
```

3.2.2. Scatter Plot (Relationship Between Two Numeric Variables)

```
plt.scatter(df["age"], df["chol"])

plt.xlabel("Age")

plt.ylabel("Cholesterol")

plt.title("Scatter Plot: Age vs Cholesterol")

plt.show()
```

3.2.3. Cluster Plot (K-Means Clustering Visualization)

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3)

df["cluster"] = kmeans.fit_predict(df[["age", "chol"]])

sns.scatterplot(x=df["age"], y=df["chol"], hue=df["cluster"], palette="viridis")

plt.title("Cluster Plot: Age vs Cholesterol")

plt.show()
```