Artificial Intelegence

ASSIGNMENT # 4

Submitted by : Muhammad Umar

Reg no. : 712

Dep. : BS.SE 4TH

Submitted To : Mr. Zubair

Submission Date : 15.June 2024

Code for data analysis and visualization with Pandas, along with the source code references for the libraries used:

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns  # Optional, for more advanced visualizations

# Replace 'your_dataset.csv' with the actual path or URL to your dataset
df = pd.read_csv('your_dataset.csv')

# Data Cleaning (Handle missing values, duplicates, etc.)
print(df.isnull().sum())  # Check for missing values
# Handle missing values (e.g., impute, remove rows)

# Remove duplicates (if necessary)
df = df.drop_duplicates()

# Data Visualization (Explore trends and relationships)

# Univariate Analysis (Exploring Single Features)

# Numerical features (Histogram)
df['column_name'].hist(bins=10, edgecolor='black')
plt.xlabel('column_name')
plt.ylabel('Frequency')
plt.title('Distribution of column_name')
plt.grid(True)
plt.show()

# Numerical features (Boxplot)
sns.boxplot(
    x = "categorical_column",
    y = "numerical_column",
    showmeans=True,  # Display mean values as points
    data=df
)
plt.xlabel('categorical_column')
plt.ylabel('numerical_column')
plt.title('Distribution of numerical_column across categories')
plt.show()

# Categorical features (Bar chart)
df['categorical_column'].value_counts().plot(kind='bar')
plt.xlabel('categorical_column')
plt.ylabel('Count')
plt.title('Distribution of categorical_column')
plt.show()

# Bivariate Analysis (Exploring Relationships Between Features)

# Numerical vs. Numerical (Scatter plot)
plt.scatter(df['column1'], df['column2'])
plt.xlabel('column1')
plt.ylabel('column2')
```

```
plt.title('Relationship between column1 and column2')
plt.grid(True)
plt.show()

# Numerical vs. Categorical (Boxplot grouped by category)
sns.boxplot(
    x = "categorical_column",
    y = "numerical_column",
    showmeans=True,
    data=df
)
plt.xlabel('categorical_column')
plt.ylabel('numerical_column')
plt.title('Distribution of numerical_column across categories')
plt.show()

# Additional Visualizations (Consider these based on your data)

# Heatmap (correlations between features)
sns.heatmap(df.corr(), annot=True)  # Annotate with correlation values
plt.title('Correlation Matrix')
plt.show()

# Pie chart (proportion of categories)
df['categorical_column'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.title('Distribution of categorical_column')
plt.show()
```

Source Code References:

- **pandas:** https://github.com/pandas-dev/pandas (Open-source library, hosted on GitHub)
- **matplotlib:** https://github.com/matplotlib/matplotlib (Open-source library, hosted on GitHub)
- **seaborn:** https://github.com/mwaskom/seaborn (Built on top of Matplotlib, hosted on GitHub)

Explanation:

1. **Import libraries:** This code imports `pandas` for data manipulation, `matplotlib.pyplot` for basic plotting, and `seaborn` for more advanced visualizations (optional).
2. **Data Loading:** It loads the dataset using `pd.read_csv()`. Replace `'your_dataset.csv'` with the actual path or URL to your dataset.
3. **Data Cleaning:** This section includes a placeholder to check for missing values and handle them appropriately (e.g., impute, remove rows). You'll need to fill in the specific data cleaning steps based on your dataset.
4. **Data Visualization:** The code demonstrates various visualizations for exploring your data:
    - Univariate analysis: Histograms and boxplots for numerical features, bar charts for categorical features.
    - Bivariate analysis: Scatter plots and boxplots to examine relationships between features.
    - Additional visualizations (optional): Heatmaps for correlations, pie charts for category proportions.
5. **Remember:** Replace `'column_name'` with the actual names of your columns throughout the code.