DJS-Compute

Data Analytics & Machine Learning

Assignment-3

Matplotlib and Seaborn Data Visualization

Dataset:

Use the "Titanic: Machine Learning from Disaster" dataset, which can be obtained from the following link:

(https://www.kaggle.com/c/titanic/data)

Resources:

1. **Matplotlib Official Documentation**
2. **Matplotlib Tutorial on DataCamp**
3. **Seaborn Official Documentation**
4. **Seaborn Tutorial on DataCamp**
5. **Seaborn Tutorial on Towards Data Science**
6. **Seaborn Cheat Sheet**

Tasks:

1. Data Loading and Exploration:

a. Load the Titanic dataset into a Pandas DataFrame.
b. Display the first few rows of the dataset.
c. Check for missing values in the dataset and handle them appropriately.
d. Calculate basic summary statistics for the numerical columns (e.g., age, fare).

2. Data Visualization with Matplotlib:

a. Create a bar chart showing the count of passengers in each passenger class (1st, 2nd, 3rd).
b. Create a histogram of passenger ages, labeling the x-axis as "Age" and the y-axis as "Count."
c. Create a pie chart to show the distribution of male and female passengers.
d. Create a box plot for the fare to visualize its distribution.

3. Data Visualization with Seaborn:

a. Create a heatmap showing the correlation between different numerical features in the dataset. Annotate the heatmap with the correlation values.
b. Create a violin plot to visualize the distribution of ages by passenger class. Each violin plot should represent a different class.

c. Create a count plot to show the number of survivors and non-survivors, differentiating by passenger class. Use different colors for each class.
d. Create a pair plot for numerical features to explore relationships between them.

4. Questions to Answer with Plots:

a. What is the distribution of passengers by class?
b. What is the age distribution of passengers on the Titanic?
c. What is the gender distribution among passengers?
d. How does fare vary on the Titanic?
e. Is there a correlation between different numerical features in the dataset?
f. How does age distribution vary by passenger class?
g. What is the survival rate for each passenger class?
h. Are there any interesting relationships between numerical features in the dataset?

Submission:

Students should submit their Jupyter Notebook or Python script containing the code for data loading, data cleaning, and the creation of various plots. Along with the code, provide explanations and interpretations of the plots in a clear and concise manner.

The deadline for completing the tasks is November 1, 2023. A discussion session, addressing any doubts and task-related topics, will be conducted in the following week.

Here are a few additional visualization libraries and a brief overview of each:

1. Bokeh:

- Bokeh is a Python library for creating interactive, web-ready visualizations. It's designed to work seamlessly with web applications, allowing you to build interactive dashboards and web-based plots.

- [Bokeh Official Documentation](https://docs.bokeh.org/en/latest/index.html)

2. Altair:

- Altair is a declarative statistical visualization library for Python. It allows you to create interactive visualizations with concise and expressive code.

- [Altair Official Documentation](https://altair-viz.github.io/)

3. Holoviews:

- HoloViews is an open-source Python library that makes data visualization easy. It provides high-level building blocks for creating interactive visualizations and dashboards.

- [HoloViews Official Documentation](http://holoviews.org/)

4. Folium:

- Folium is a Python library for creating interactive maps. It's particularly useful for geospatial data visualization and can be integrated with Jupyter Notebooks.

- [Folium Documentation](https://python-visualization.github.io/folium/)

5. Geopandas:

- Geopandas is an open-source library that simplifies working with geospatial data. It's useful for plotting maps, spatial data analysis, and geovisualization.

- [Geopandas Documentation](https://geopandas.org/en/stable/)

6. Dash:

- Dash is a Python framework for building interactive web applications with ease. It's designed for creating interactive data dashboards and visualizations.

- [Dash Documentation](https://dash.plotly.com/introduction)

7. Pygal:

- Pygal is a Python library for creating scalable vector graphics (SVG) charts. It's designed to create attractive, responsive charts that can be easily embedded in websites.

- [Pygal Documentation](http://www.pygal.org/en/stable/)