Experiment Number : 9 - Data Visualization Tools and Techniques

Batch: FDS-2 **Roll Number: 1601042223** **Name: Chandana Ramesh Galgali**

Aim of the Experiment: Exploration of data visualization tools and techniques

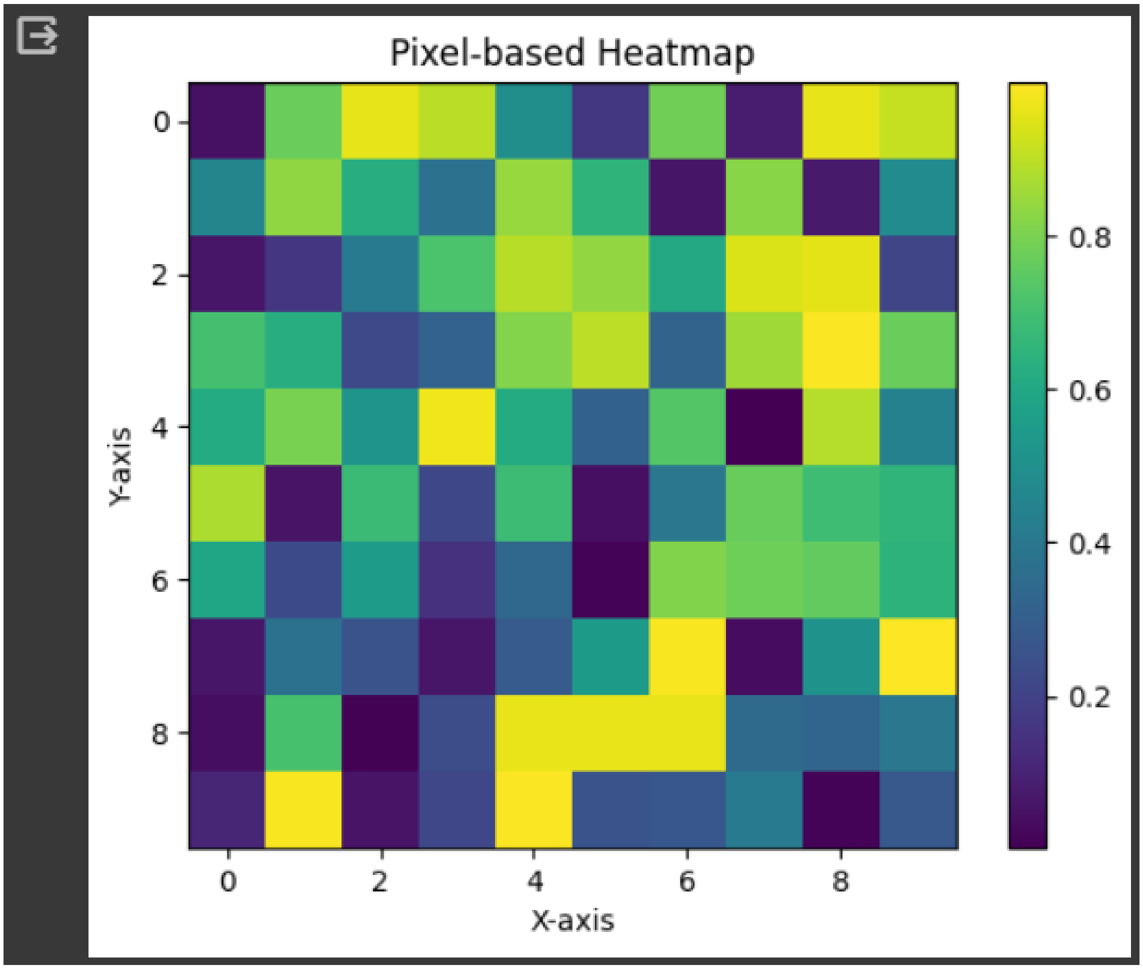
Program/ Steps:

1. Analyze the data with data visualization techniques.
2. Implement the Pixel Based data visualization techniques.
3. Implement the Geometric data visualization techniques.
4. Implement the Icon Based Visualization technique.
5. Implement the Hierarchy Based Visualization technique.
6. Explore the Any Data Visualization Tool.

Code with Output/Result:

1. Pixel Based Visualization Techniques

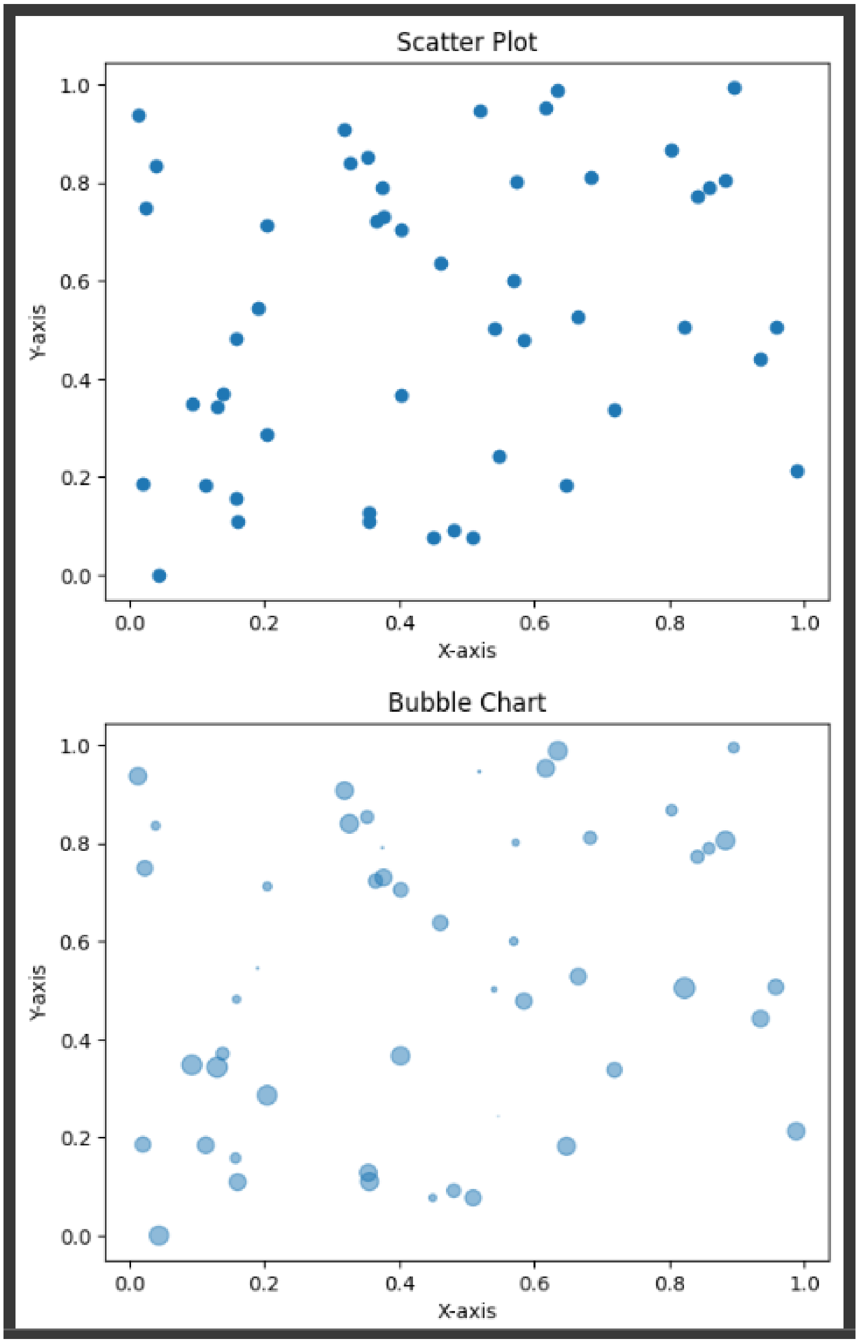
```
import numpy as np

import matplotlib.pyplot as plt

data = np.random.rand(10, 10)

plt.imshow(data, cmap='viridis', interpolation='nearest')

plt.colorbar()

plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Pixel-based Heatmap')

plt.show()
```

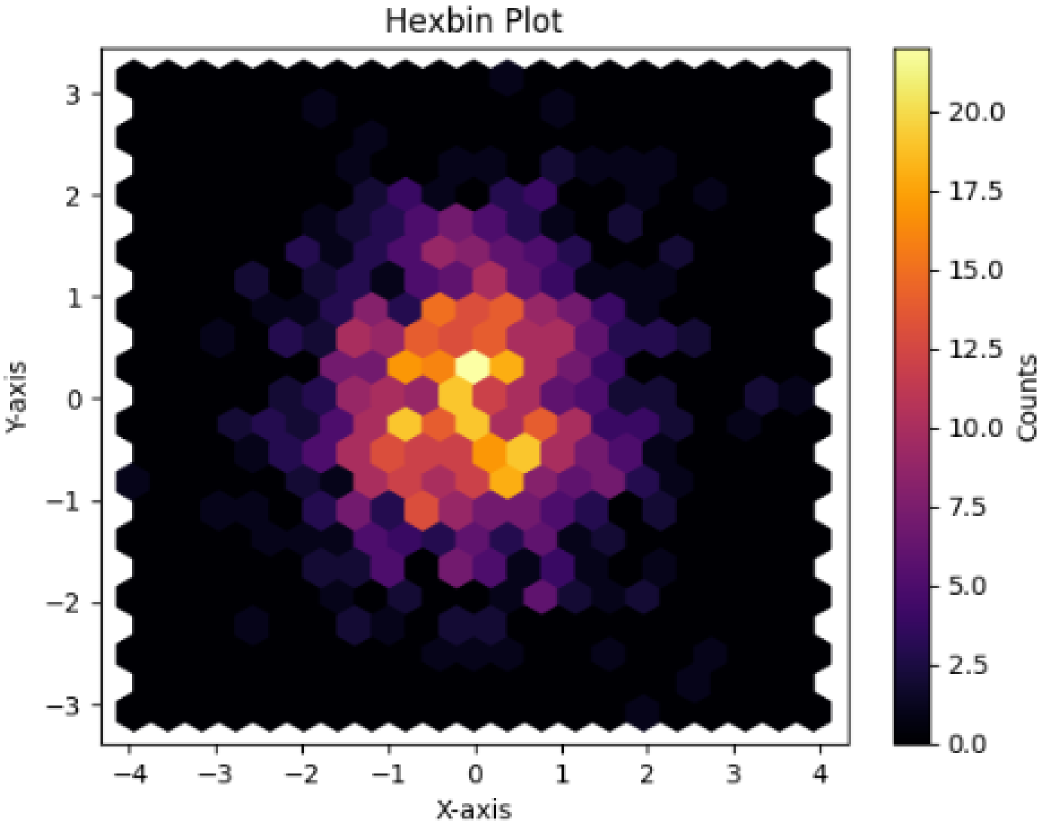


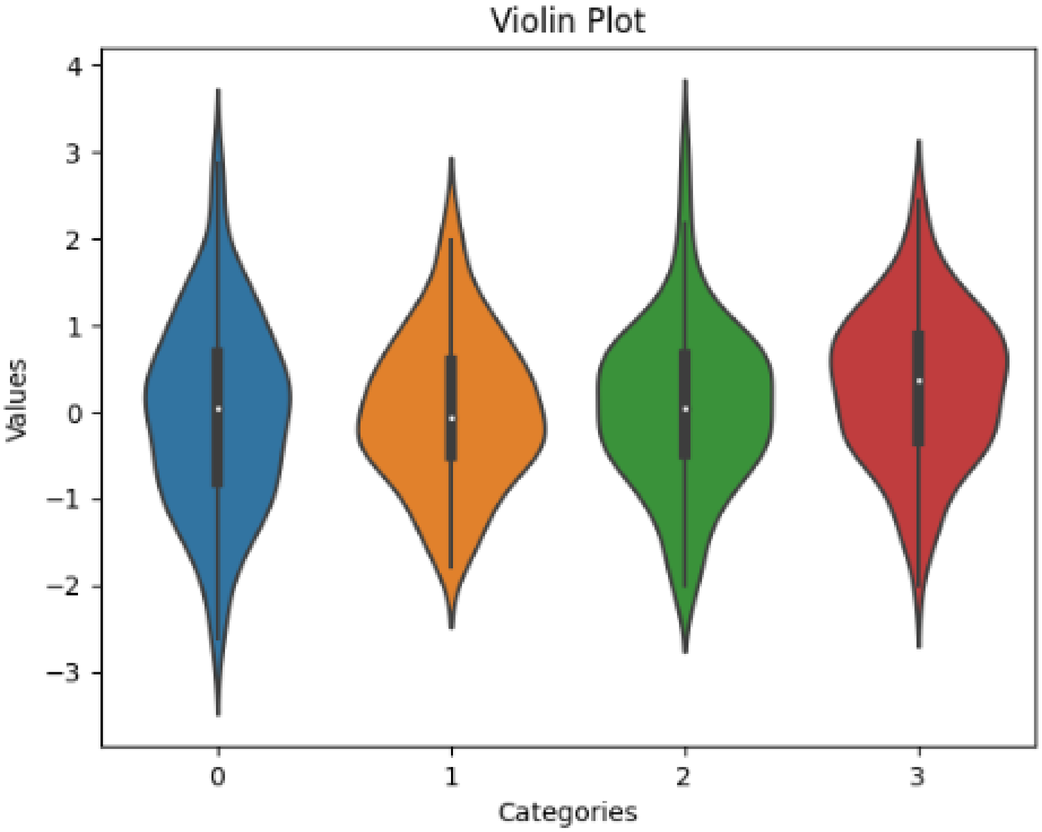
2. Geometric Based Visualization Techniques:

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
x = np.random.rand(50)
y = np.random.rand(50)
plt.scatter(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.show()
sizes = np.random.rand(50) * 100
plt.scatter(x, y, s=sizes, alpha=0.5)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Bubble Chart')
plt.show()
x = np.random.randn(1000)
y = np.random.randn(1000)
plt.hexbin(x, y, gridsize=20, cmap='inferno')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Hexbin Plot')
plt.colorbar(label='Counts')
plt.show()
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
categories = ['A', 'B', 'C', 'D']
```

```
data = [np.random.randn(100) for _ in categories]

sns.violinplot(data=data)

plt.xlabel('Categories')

plt.ylabel('Values')

plt.title('Violin Plot')

plt.show()
```



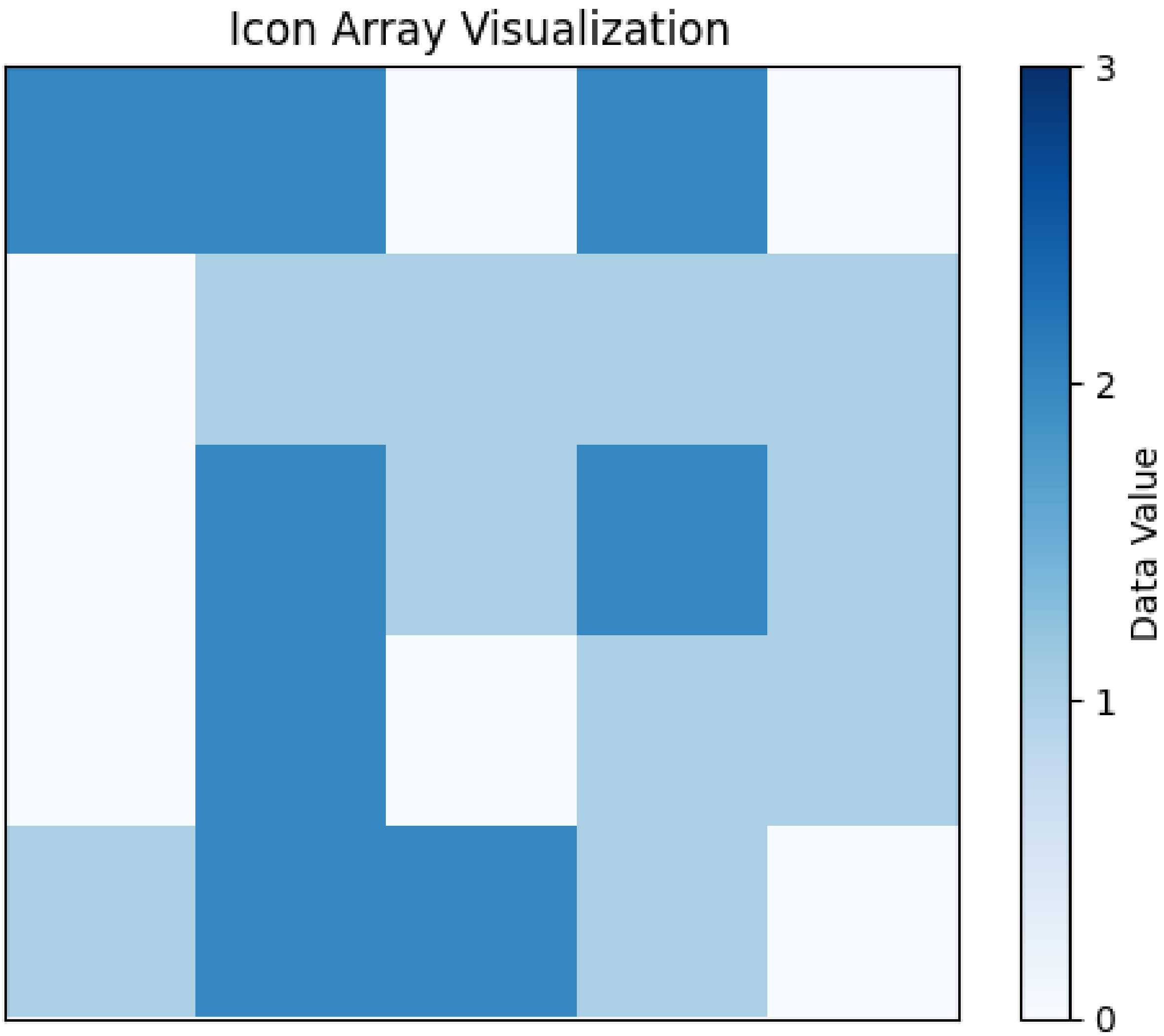

Hexbin Plot
Y-axis
3
2
1
0
−1
−2
−3
X-axis
−4
−3
−2
−1
0
1
2
3
4
Counts
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

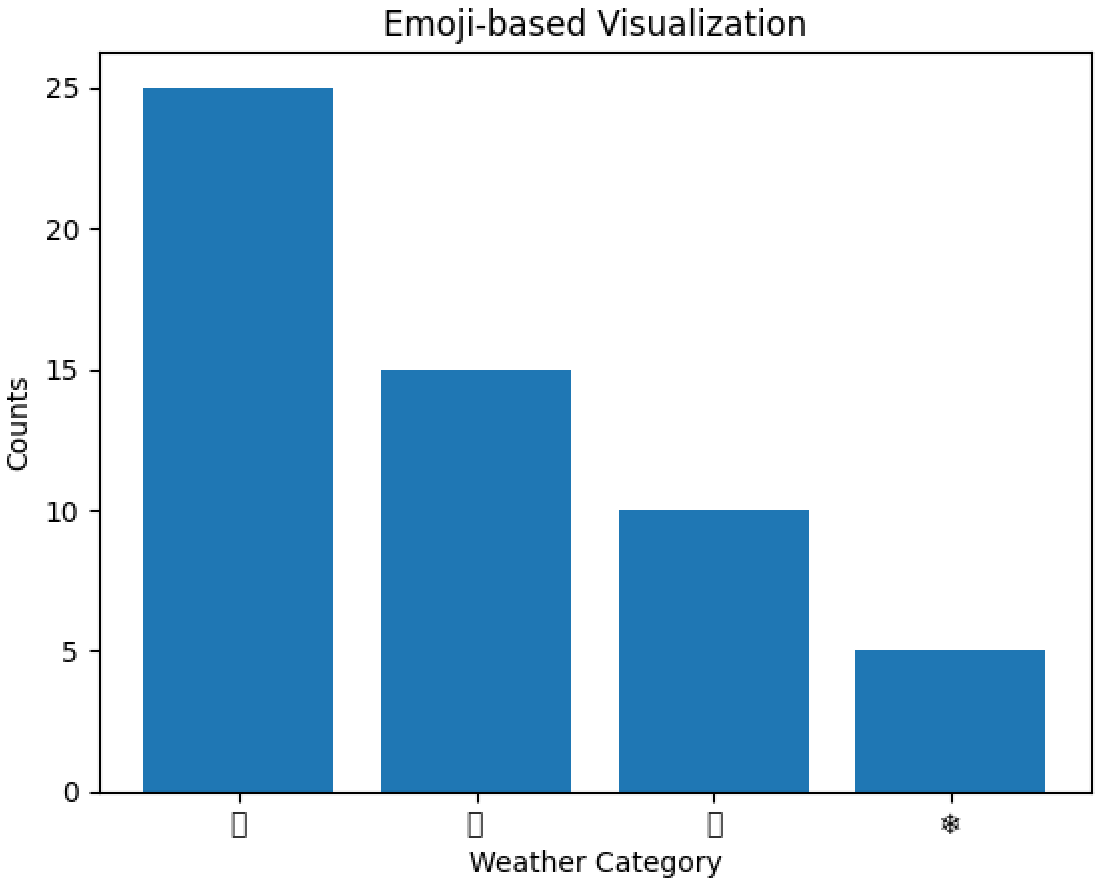

Violin Plot
Values
4
3
2
1
0
−1
−2
−3
Categories
0
1
2
3

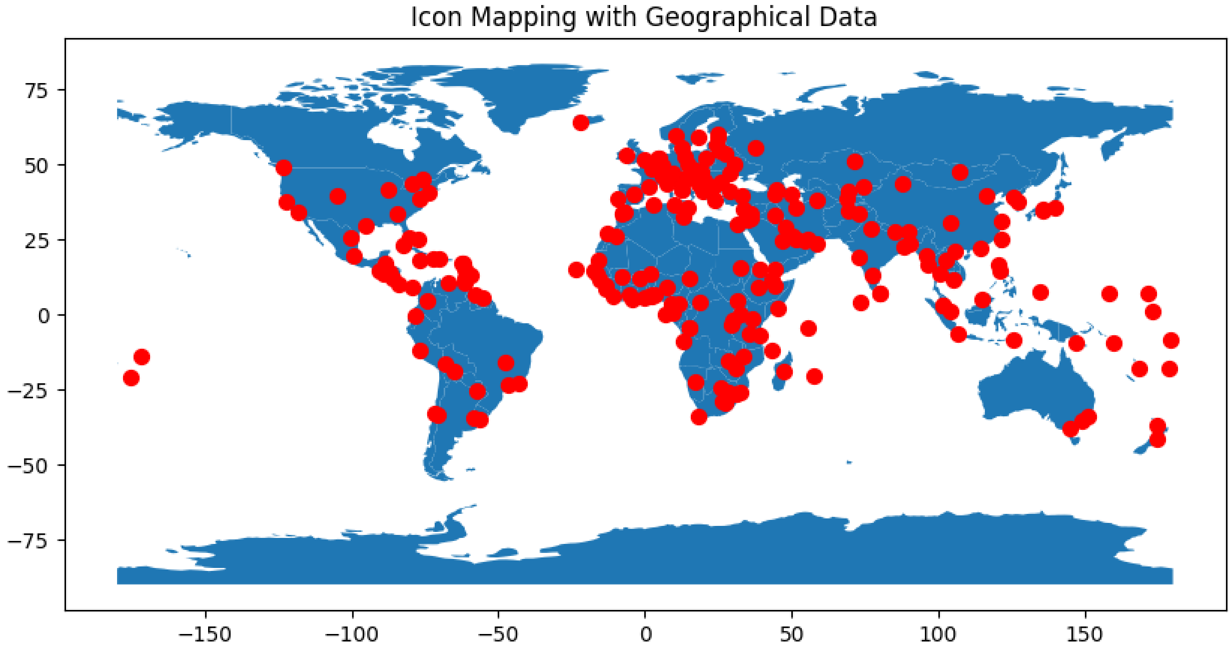
3. Icon-based visualization techniques

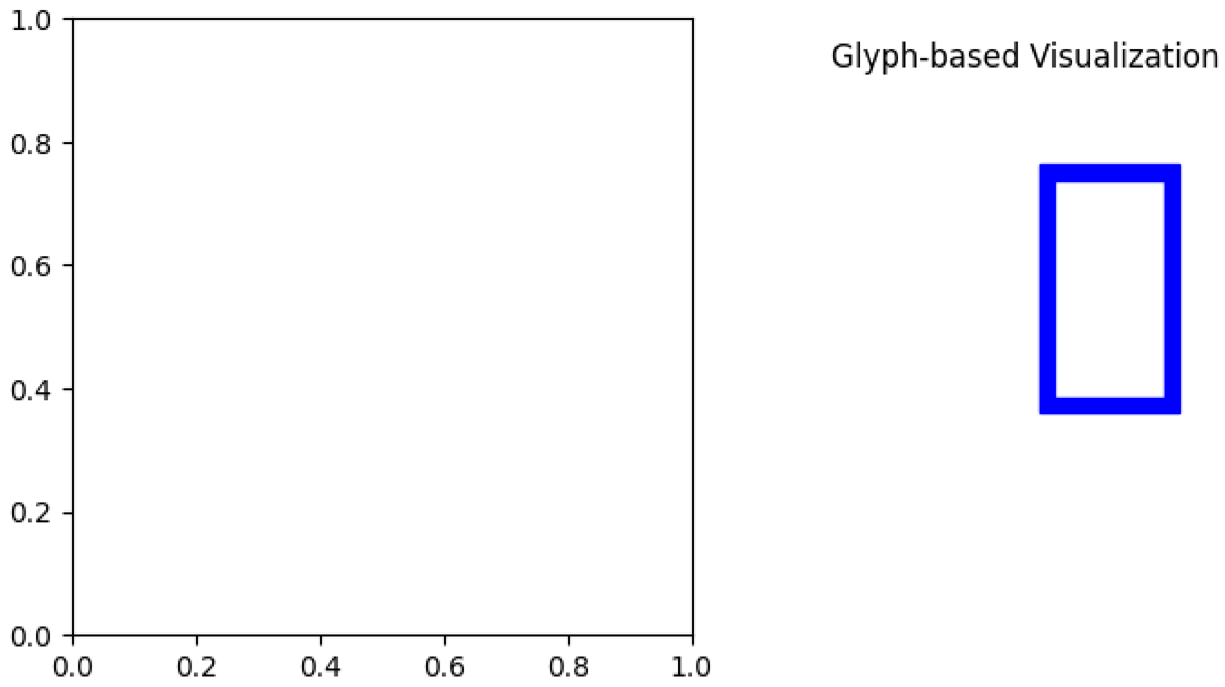
```
import matplotlib.pyplot as plt
import numpy as np
data = np.random.randint(0, 3, size=(5, 5))
plt.imshow(data, cmap='Blues', vmin=0, vmax=3)
plt.colorbar(ticks=[0, 1, 2, 3], label='Data Value')
plt.xticks([])
plt.yticks([])
plt.title('Icon Array Visualization')
plt.show()
import matplotlib.pyplot as plt
categories = ['🌞', '🌧️', '🌈', '❄️']
counts = [25, 15, 10, 5]
plt.bar(categories, counts)
plt.xlabel('Weather Category')
plt.ylabel('Counts')
plt.title('Emoji-based Visualization')
plt.show()
import geopandas as gpd
import matplotlib.pyplot as plt
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
ax = world.plot(figsize=(10, 6))
cities = gpd.read_file(gpd.datasets.get_path('naturalearth_cities'))
cities.plot(ax=ax, marker='o', color='red', markersize=50)
plt.title('Icon Mapping with Geographical Data')
plt.show()
import matplotlib.pyplot as plt
from matplotlib.text import TextPath
from matplotlib.patches import PathPatch
glyph = TextPath((0, 0), "🌟", size=100, prop='emoji', usetex=False)
glyph = TextPath((0, 0), "🌟", size=100, prop='emoji', usetex=False)
patch = PathPatch(glyph, color='blue')
```

```
fig, ax = plt.subplots(figsize=(4, 4))
fig, ax = plt.subplots(figsize=(4, 4))
ax.add_patch(patch)
ax.set_xlim(-100, 100)
ax.set_ylim(-100, 100)
plt.axis('off')
plt.title('Glyph-based Visualization')
plt.show()
```




Emoji-based Visualization
Counts
0
5
10
15
20
25
□
□
□
❄
Weather Category

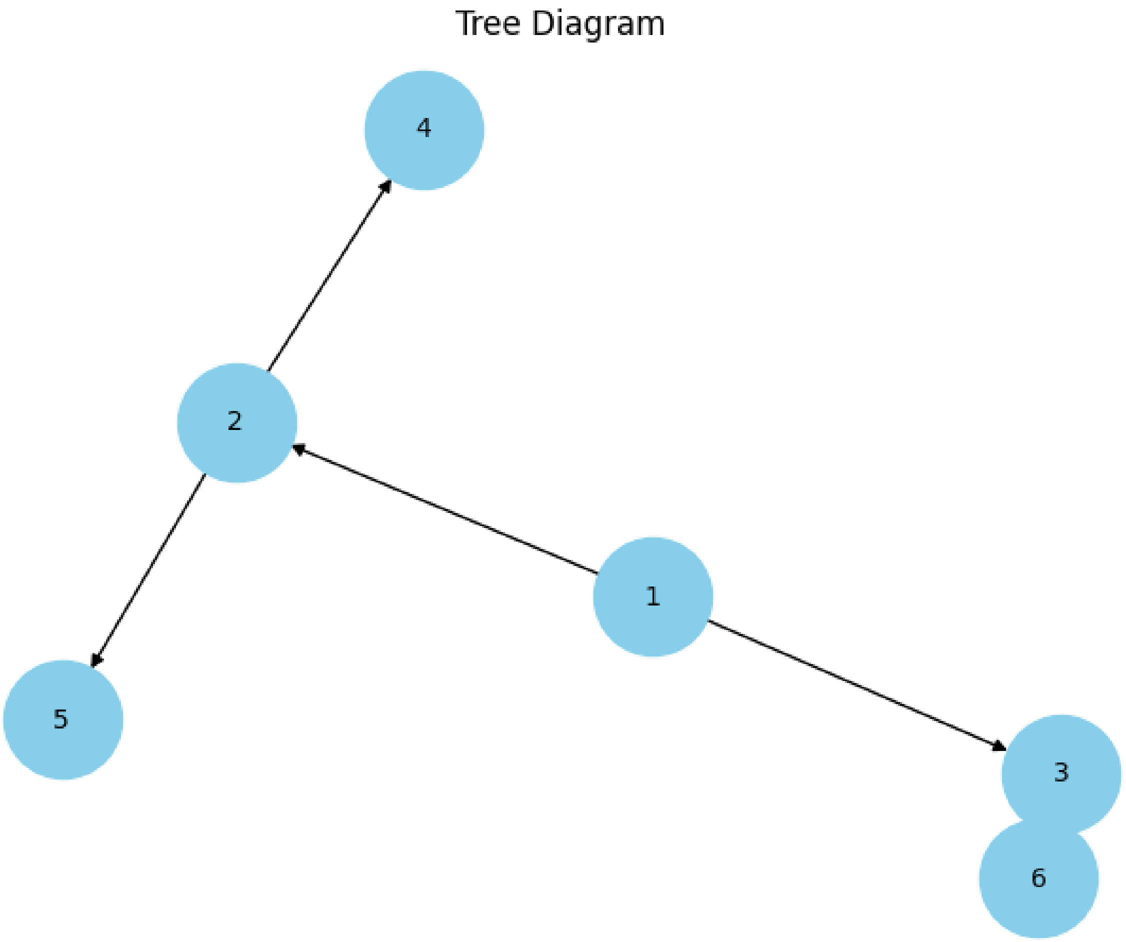

Icon Mapping with Geographical Data
75
50
25
0
−25
−50
−75
−150
−100
−50
0
50
100
150

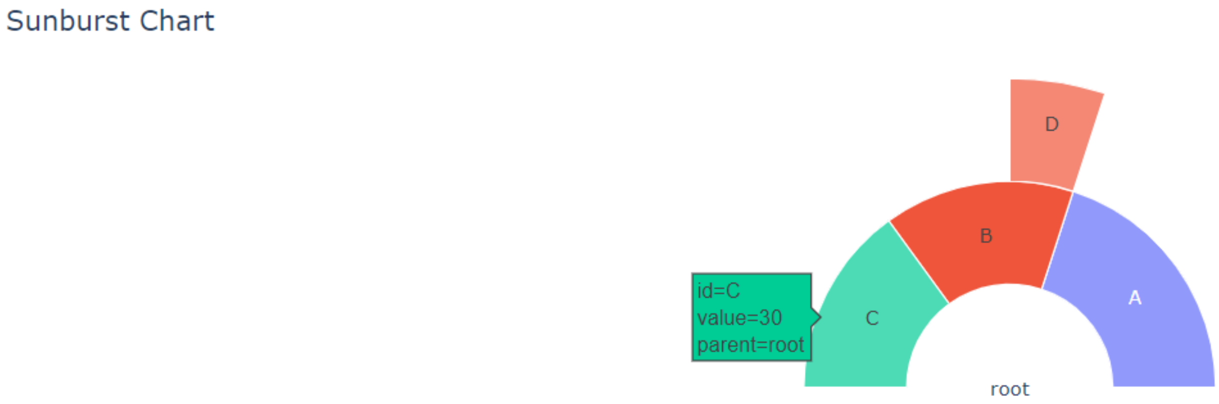


4. Hierarchical-based visualization techniques

```
import networkx as nx

import matplotlib.pyplot as plt

G = nx.DiGraph()

G.add_edges_from([(1, 2), (1, 3), (2, 4), (2, 5), (3, 6)])

pos = nx.spring_layout(G)

nx.draw(G, pos, with_labels=True, node_size=2000, node_color="skyblue",

font_size=10, font_color="black")

plt.title('Tree Diagram')

plt.show()

import plotly.express as px

data = {

'id': ['root', 'A', 'B', 'C', 'D'],

'parent': ['', 'root', 'root', 'root', 'B'],

'value': [100, 40, 20, 30, 10]

}

fig = px.sunburst(data, names='id', parents='parent', values='value')
```

```
fig.update_layout(title='Sunburst Chart')

fig.show()
```

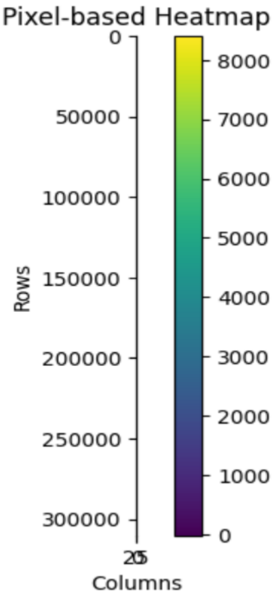


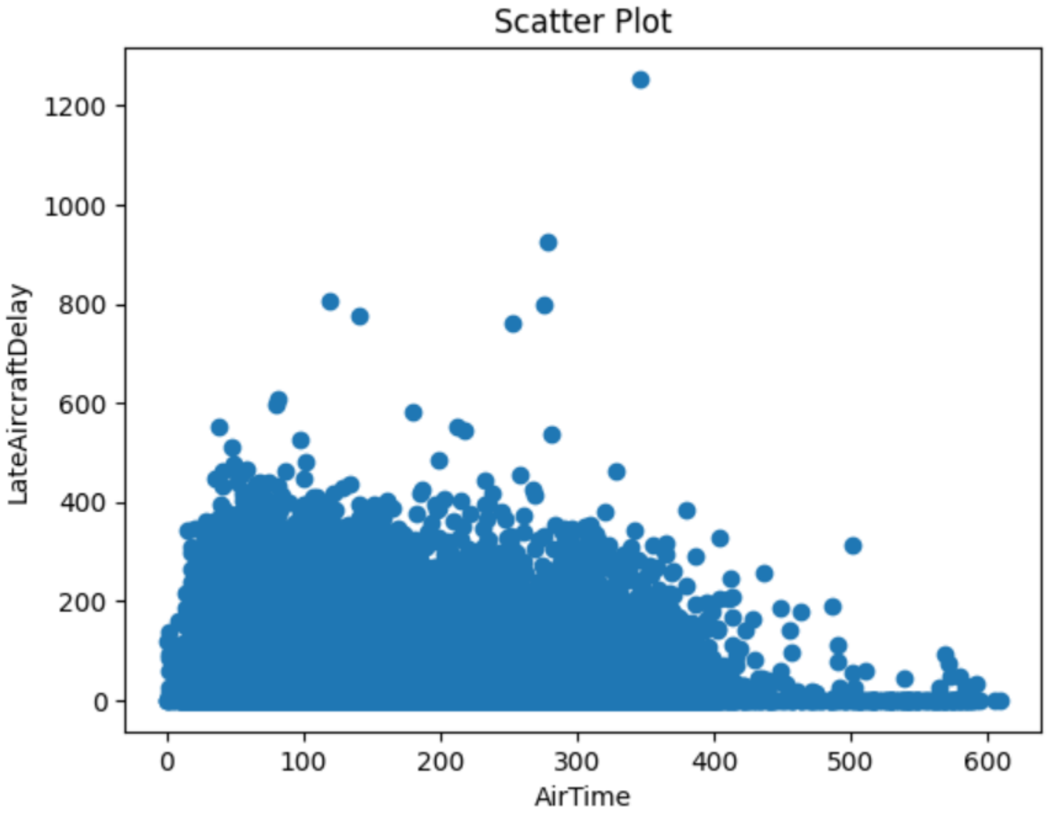


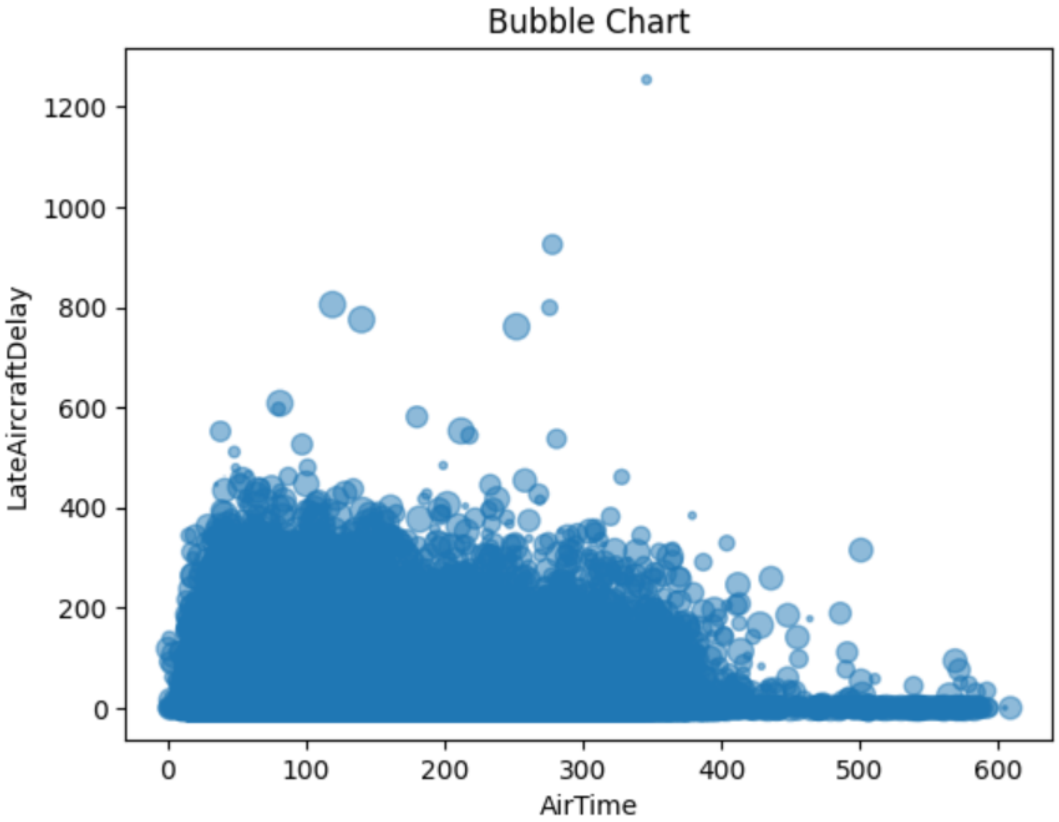
Code:

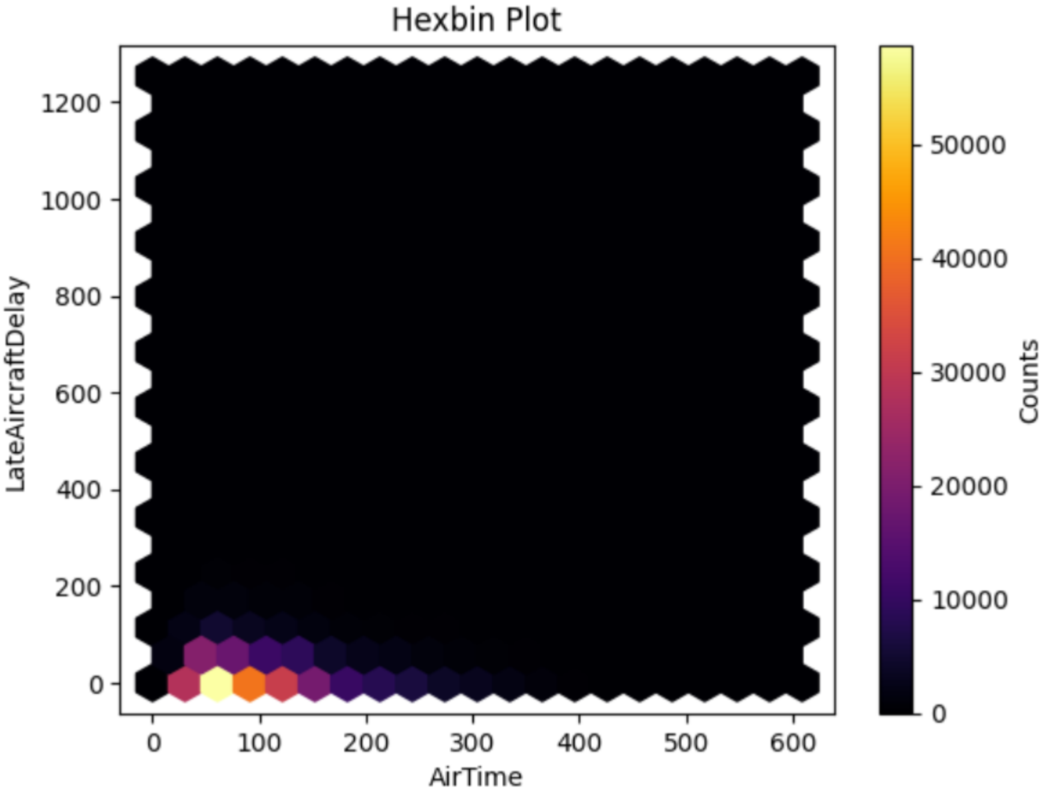
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.text import TextPath
from matplotlib.patches import PathPatch
dataframe = pd.read_csv(r'/content/Flight_delay (1).csv')
dataframe = dataframe.apply(pd.to_numeric, errors='coerce')
# Create a pixel-based heatmap
plt.imshow(dataframe.values, cmap='viridis', interpolation='nearest')
plt.colorbar()
plt.xlabel('Columns')
plt.ylabel('Rows')
plt.title('Pixel-based Heatmap')
plt.show()
x = dataframe['AirTime']
y = dataframe['LateAircraftDelay']
# Create a scatter plot
plt.scatter(x, y)
plt.xlabel('AirTime')
plt.ylabel('LateAircraftDelay')
plt.title('Scatter Plot')
plt.show()
sizes = np.random.rand(len(x)) * 100
# Create a bubble chart
plt.scatter(x, y, s=sizes, alpha=0.5)
plt.xlabel('AirTime')
plt.ylabel('LateAircraftDelay')
plt.title('Bubble Chart')
plt.show()
# Create a hexbin plot
plt.hexbin(x, y, gridsize=20, cmap='inferno')
plt.xlabel('AirTime')
plt.ylabel('LateAircraftDelay')
plt.title('Hexbin Plot')
plt.colorbar(label='Counts')
plt.show()
columns = ['AirTime', 'LateAircraftDelay']
# Create a violin plot
sns.violinplot(data=dataframe[columns])
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Violin Plot')
plt.show()
# Create an icon array visualization
plt.imshow(dataframe[columns].values, cmap='Blues', vmin=0, vmax=3)
plt.colorbar(ticks=[0, 1, 2, 3], label='Data Value')
plt.xticks([])
plt.yticks([])
plt.title('Icon Array Visualization')
plt.show()
```

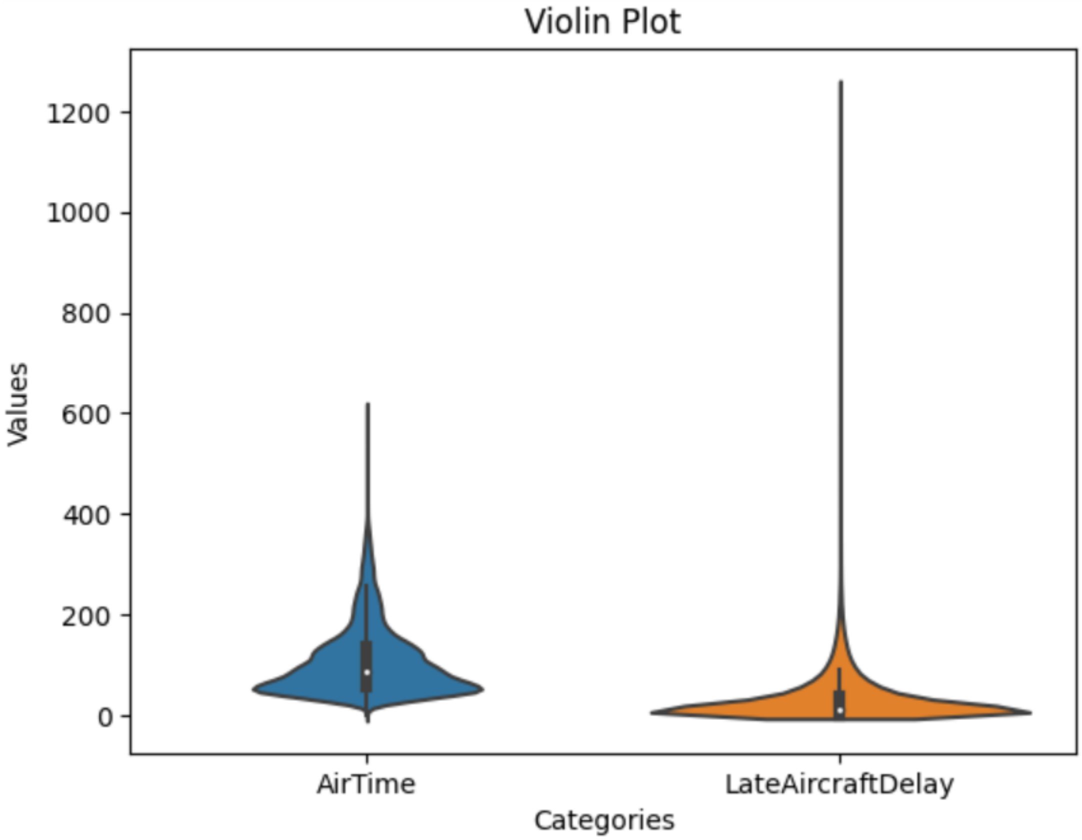
Output:

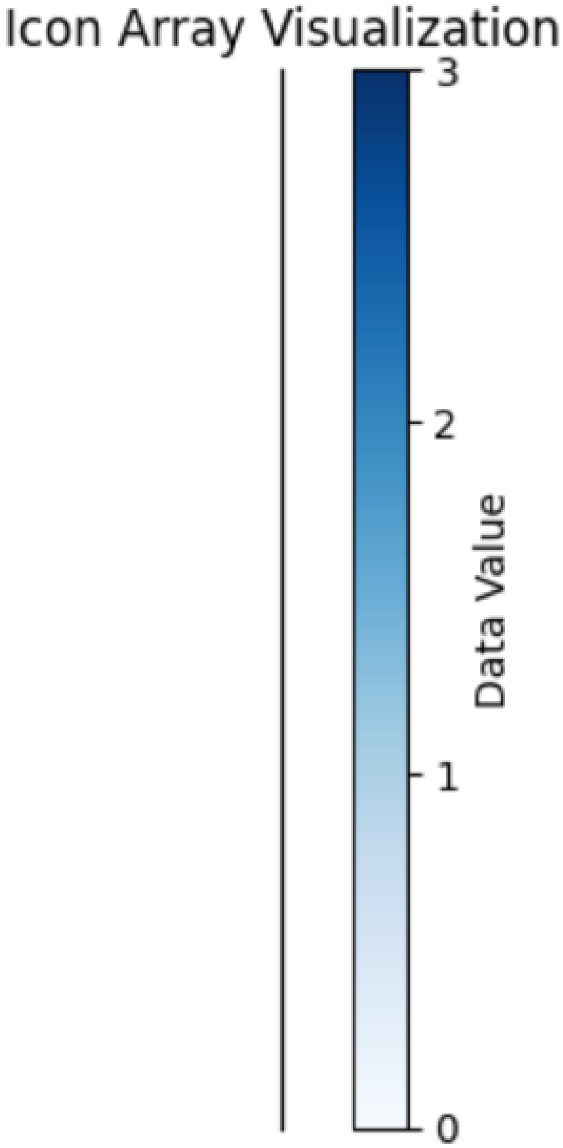



Scatter Plot
LateAircraftDelay
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
AirTime


Bubble Chart
LateAircraftDelay
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
AirTime


Hexbin Plot
LateAircraftDelay
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600
AirTime
Counts
50000
40000
30000
20000
10000
0


Violin Plot
Values
1200
1000
800
600
400
200
0
AirTime
LateAircraftDelay
Categories



Post Lab Question-Answers:

1. How data visualization plays roles in data mining?

Ans: Data visualization plays a crucial role in data mining by helping analysts and data scientists gain insights from complex datasets. Here are a few ways data visualization contributes to the data mining process:

1. Exploratory Data Analysis: Data visualization techniques, such as scatter plots, histograms, and box plots, allow analysts to explore the data visually. By visualizing the data, patterns, trends, and outliers can be easily identified, leading to a better understanding of the dataset.

2. Pattern Recognition: Data mining involves identifying patterns and relationships within the data. Visualizations, such as heatmaps, network graphs, and decision trees, can help reveal hidden patterns that may not be apparent in raw data. These visual representations make it easier to identify clusters, associations, and correlations.

3. Communication and Presentation: Data visualization is an effective way to communicate findings and insights to stakeholders. Visualizations make complex information more accessible and understandable to a wider audience. By presenting data visually, decision-makers can quickly grasp the key takeaways and make informed decisions based on the findings.

4. Model Evaluation: Data mining often involves building predictive models. Visualizations, such as ROC curves, precision-recall curves, and confusion matrices, help evaluate the performance of these models. By visualizing model metrics, analysts can assess the model's accuracy, identify areas for improvement, and compare different models.

5. Interactive Exploration: Interactive visualizations allow users to interact with the data, enabling them to drill down into specific subsets, filter data, and explore different dimensions. This interactivity enhances the data mining process by enabling analysts to ask ad-hoc questions and gain deeper insights into the data.

In summary, data visualization plays a crucial role in data mining by facilitating data exploration, pattern recognition, communication of findings, model evaluation, and interactive exploration. It helps analysts uncover insights, make informed decisions, and effectively communicate their findings to stakeholders.

2. What is chatbot? How does it differ from Dashboard?

Ans: A chatbot is a computer program designed to simulate human conversation through text or voice interactions. It uses natural language processing (NLP) and machine learning techniques to understand and respond to user queries or commands. Chatbots can be used for various purposes, such as customer support, information retrieval, and task automation.

On the other hand, a dashboard is a visual interface that displays key information, metrics, and data in a consolidated and easily understandable format. Dashboards are typically used to monitor and analyze data in real-time, providing users with a snapshot of the current state of affairs. They often include charts, graphs, tables, and other visual elements to present data in a visually appealing and informative manner.

While both chatbots and dashboards involve data interaction and visualization, they serve different purposes and have distinct characteristics:

1. Interaction Style: Chatbots engage in conversational interactions with users, where users can ask questions, provide commands, or seek assistance. Dashboards, on the other hand, are primarily designed for users to passively view and analyze data without engaging in direct conversation.

2. Data Retrieval: Chatbots are typically used to retrieve specific information or perform tasks based on user queries. They use NLP techniques to understand user intent and retrieve relevant data from various sources. Dashboards, on the other hand, display pre-defined data sets or allow users to customize the data they want to view.

3. Real-time Monitoring: Dashboards are often used for real-time monitoring of data, providing up-to-date information on key metrics. Chatbots can also provide real-time information, but their primary focus is on interactive conversations rather than continuous monitoring.

4. User Experience: Chatbots aim to provide a conversational and personalized user experience. They can adapt their responses based on user input and context. Dashboards, on the other hand, focus on providing a visual and intuitive user interface for data exploration and analysis.

In summary, chatbots and dashboards serve different purposes and have distinct interaction styles. Chatbots enable conversational interactions and data retrieval, while dashboards provide visual interfaces for data monitoring and analysis. Both can be valuable tools in different contexts, depending on the specific needs and goals of the users.

Outcomes: :Comprehend various data visualization techniques

Conclusion (based on the Results and outcomes achieved):

The exploration of data visualization tools and techniques has proven to be invaluable in the data mining process. It has enabled analysts to uncover insights, make informed decisions, and effectively communicate findings to stakeholders. By leveraging these tools and techniques, analysts have been able to extract maximum value from complex datasets, leading to improved decision-making and better outcomes.

References:

Books/ Journals/ Websites

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition

---