Task 5: To design and perform visualization for Trees

• TreeMap, Sun Burst Display

Tools: Tableau, Language :Python

5.a) Construct a treeMap display on a real-world dataset. You can download the dataset in the given link. Cars_US_dataset | Kaggle. US Cars'data was scraped from AUCTION EXPORT.com. This dataset included Information about 28 brands of clean and used vehicles for sale in US. Twelve features were assembled for each car in the dataset. In the data set you can take any attributes,Values and visualize it .

Aim:

To Construct a treeMap display on a real-world dataset.

Alogrithm:

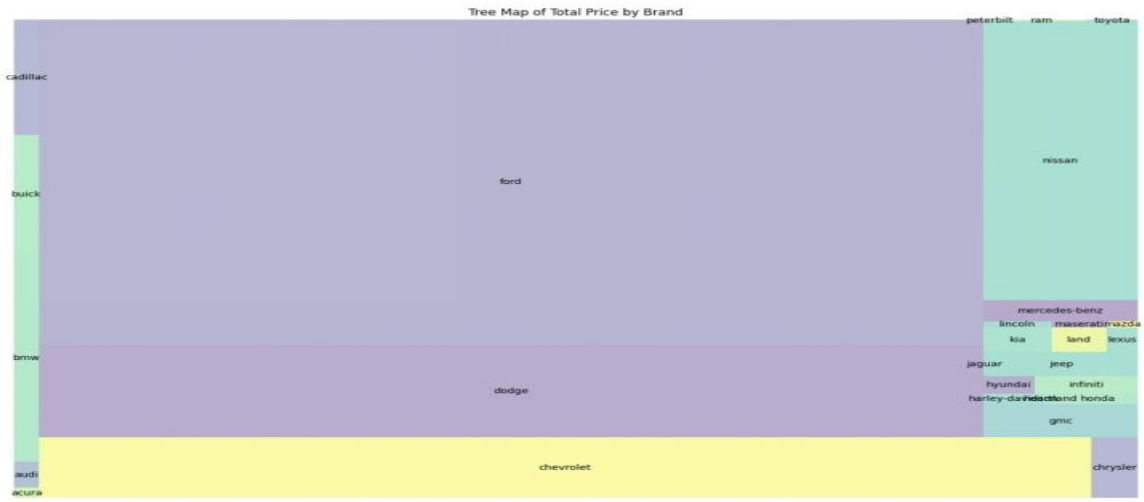
1. Install the `squa` package using pip.rify
2. Import necessary libraries such as `pandas`, `matplotlib.pyplot`, and `squarify`.
3. Read the dataset from '/content/USA_cars_datasets.csv' into a pandas DataFrame `df`.
4. Group the DataFrame `df` by the 'brand' column and calculate the total price for each brand.
5. Create a figure of size 18x12 inches for the tree map visualization.
6. Use `squarify.plot()` to create a tree map plot.
7. Pass the sizes of the rectangles as the total price of each brand (`brand_total_price['price']`) and labels **as** the brand names (`brand_total_price['brand']`).
8. Set transparency with alpha=0.4 for better visualization.
9. Turn off the axis.
10. Set the title of the plot as 'Tree Map of Total Price by Brand'.
11. Display the plot using plt.show()

Program

```
pip install squarify

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

```
import squarify
df=pd.read_csv('/content/USA_cars_datasets.csv')
print(df)
brand_total_price = df.groupby('brand')['price'].sum().reset_index()
plt.figure(figsize=(18, 12))
squarify.plot(sizes=brand_total_price['price'], label=brand_total_price['brand'], alpha=0.4)
plt.axis('off')
plt.title('Tree Map of Total Price by Brand')
plt.show()
```

Output:



Result:

A TreeMap was successfully created showing how different **car models** contribute to the **overall price distribution** within each **brand**. It provides an intuitive visual comparison of brands and their models based on price.

5 b.)Build a sunburst display using above program dataset.

Algorithm:

1. Install the required packages `squarify` and `plotly` using pip.

2. Import necessary libraries: `pandas`, `numpy`, `matplotlib.pyplot`, `seaborn`, `squarify`, and `plotly.express`.

3. Read the dataset from '/content/USA_cars_datasets.csv' into a pandas DataFrame `df`.

4. Group the DataFrame `df` by the combination of 'brand' and 'model' columns and calculate the total price for each combination.

5. Create a sunburst plot using Plotly Express (`px.sunburst()`):

   - Specify the DataFrame `sunburst_data` as the data source.

   - Set the path for the sunburst plot to follow the hierarchy of 'brand' and 'model'.

   - Define 'price' as the values to be represented.

   - Set the title of the plot as 'Sunburst Display of Total Price by Brand and Model'.

6. Display the plot using `fig.show()`.

Program:

```
pip install squarify

pip install plotly

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import squarify

import plotly.express as px

df=pd.read_csv('/content/USA_cars_datasets.csv')

print(df)

sunburst_data = df.groupby(['brand', 'model']).sum().reset_index()

fig = px.sunburst(sunburst_data, path=['brand', 'model'], values='price', title='Sunburst Display of Total Price by Brand and Model')

fig.show()
```

Output:

Sunburst Display of Total Price by Brand and Model

Result:

A **Sunburst chart** was successfully created showing the hierarchical relationship between **car brands and models**. The visualization clearly represents each brand's contribution to total price, making it easy to compare model-level distributions within each brand.