Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956)

NAAC A++

CATEGORY-1
DEEMED TO BE UNIVERSITY
by the
UGC University Grants Commission

School of Computing
Department of Computer Science & Engineering
(Artificial Intelligence and Machine Learning)

ACADEMIC YEAR 2025-26 (SUMMER SEMESTER)

LAB RECORD NOTEBOOK

10212CA214 - DATA VISUALIZATION

NAME: K. Sanjay

VTU.NO: 26162

REG.NO: 23UECL0029

BRANCH: CSE(AIML)

YEAR/SEM: 3rd year | 5th sem

SLOT: $S_{12}L_1$

Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act 1956)

NAAC A++

Recognised as CATEGORY-1 DEEMED TO BE UNIVERSITY by the UGC University Grants Commission

School of Computing
Department of Computer Science & Engineering
(Artificial Intelligence and Machine Learning)

ACADEMIC YEAR 2025-26 (SUMMER SEMESTER)

BONAFIDE CERTIFICATE

NAME	: K. Sanjay	BRANCH	: CSE(AIML)
VTU NO.	: 26162	REG.NO.	: 23UECL0029
YEAR/SEM	: 3rd/5th sem	SLOT NO.	: S12L1

Certified that this is a bonafide record of work done by above student in the **"10212CA214 - DATA VISUALIZATION LABORATORY"** during the year 2025-2026 (Summer Semester).

29/10/25

SIGNATURE OF LAB HANDLING FACULTY **SIGNATURE OF HOD**

Submitted for the Semester Practical Examination held on **03.11.25** at Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ME: [illegible]   SEC: Roll No:

INDEX

completed

Total Marks: 189/220

Signature of Faculty

21/07

TASK-1

→ Exploration of Data visulization Tools Tableau, Python libraries

- Connecting Dataset
- Preparation of Data

Aim:-

To visualize the distribution of different grape leaf Conditions and explore their metabata using Python (or) other visualization tools..

Algorithm:-

1. Import Dataset:
   - Load the dataset Containing folders of grape vine leaf images classified by Condition..
2. Extract Metadata:
   - list all images fiels from each class folder.
   - Create a structured format with two attributes: Images, Name, class, Label..
3. Preprocess and clean Data:
   - Ensure only valid image files are includes.
   - Remove duplicates if any..

Dataset Table for Grape vine Plants:-

Plant Name	Type	Height	Sunlight	water Need	Health Condition
Grape vine 1	Fruit Plant	110	medium	medium	Black Spot
Grape vine 2	Fruitplant	120	High	medium	Healthy
Grape vine 3	fruitPlant	121	low	High	Healthy
G.P4	fruit Plant	130	High	low	Healthy
G.P$_5$	F.P	136	low	medium	Healthy
G.PV$_6$	F.P	140	medium	low	Healthy
G.V$_7$	FP	139	low	High	Healthy
G.V$_8$	FP	105	High	low	Black Spots
G.V$_9$	FP	111	High	low	Blackspots
G.V$_{10}$	FP	100	low	High	Black Spots

o/p ✓

4. Create Summary Table:
- Count number of images in each class.
- Display sample records in tabular form..

5. visualization:
- Generate a bar chart to show the distribution of images per class using Python..

#code:-

```
import Pandas as Pd
import Numpy as np
a=Pd.read_csv ("Plant_disease.csv")
Print (a)
sh = a.shape
Print (sh)
a.info()
a.describe()
Print(a.head())
Pd.isna(a).sum()
a.dropna()
a.duplicated()
a.drop_duplicates()
```

VEL TECH	
EX No.	
PERFORMANCE (5)	4
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	2
TOTAL (20)	~~20~~
SIGN WITH DATE	12+2=14

4/7/25

Result:- Thus, to visualize the distribution of different grape leaf conditions and explore their metadata using Python (or) other visualization tools was executed successfully..

28/07

TASK-2

→To visualize and Perform Univariate analysis using Continuous and categorical data.

Tools: Tableau, Python..

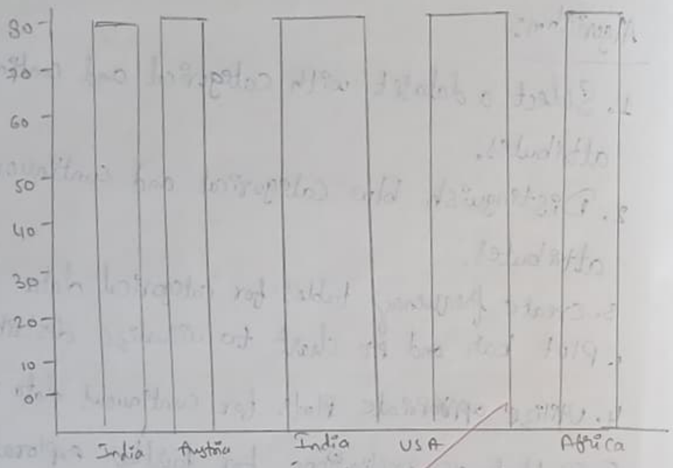
Aim:- Constuct a Bar chart, Pie chart using Univariate analysis of categorical data using identified data..

Algorithm:-

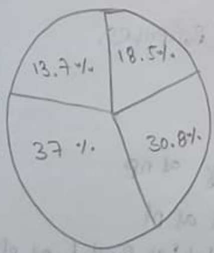
1. Select a dataset with categorical and continuous attributes.
2. Distinguish b/w categorical and continuous attributes.
3. Create frequency tables for categorical data then =Plot bar and pie chart to visualize distribution..
4. Utilize appropriate Plots for continuous data analysis.
5. Reflect on implications for further exploration (or) decision-making based on the analysis..
6. Summarize findings and observations for reporting (or) Presentation purposes.

Program:-

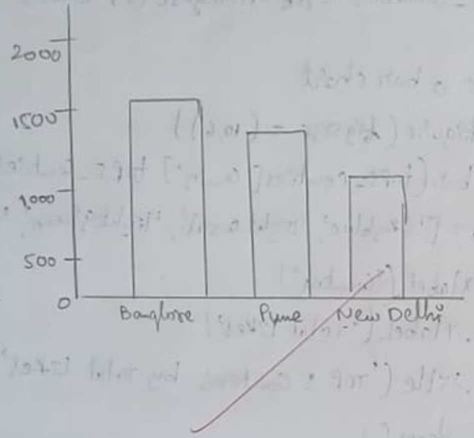
```
import numpy as np
import Pandas as pd
import matplotlib.pyplot.as plt
import seaborn as sns
df = pd.read_csv("/content/sentimentdataset.csv")
Platform-likes_top5=df.groupby("platform")
```



o/p



```
# create a pie chart
  plt.figure(figsize=(4,4))
  platform_likes_top5.plot(kind='pie', autopct='%1.1f%%',
    startangle=140, colors=['skyblue', 'light coral'])
  plt.title('Top 5 platforms by Total likes')
  plt.ylabel('')
  plt.show()
  top5_countries = df.nlargest(5, 'likes')

# create a bar chart
  plt.figure(figsize = (10,6))
  plt.bar(top5_countries['country'], top5_countries['likes'],
  color = ['skyblue', 'lightcoral', 'lightgreen', 'lightsteelblue']
  plt.xlabel('country')
  plt.Ylabel('Total likes')
  plt.title('Top 5 countries by total likes')
  plt.show()

  data = {
    'Text': ['Enjoying a beautiful day at the park!']
      'Retweets': [15, 5, 20, 8, 12],
      'Likes': [30, 10, 40, 15, 25]
    }
  df = pd.Dataframe(data)
  sns.set(style="whitegrid")
  plt.figure(figsize = (10,6))
  sns.scatterplot(x='Retweets', y='Likes', data=df,
        color = 'blue', alpha = 0.7)
```



28/07

TASK- 2[B]

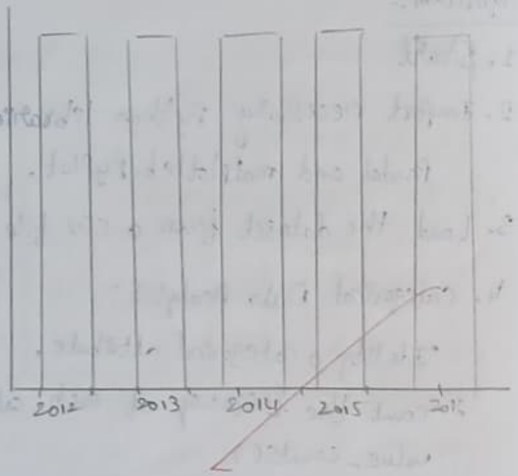
Aim:- To Perform univariate analysis on a dataset containing employee details by identifying categorical and continuous data attributes and visualizing them using Barchart, Pie chart, Histogram..

Algorithm:-

```
1. Start
2. Import necessary Python libraries:
      Pandas and matplotlib.pyplot.
3. Load the dataset from a csv file.
4. Categorical Data Analysis:
      •Identify a categorical attribute.
      •Count the frequency of each category using
       value_counts().
      •Plot the Bar chart using plt.bar ()
      •Plot the Pie chart using plt.pie ().
5. Add appropiate titles and axis labels to each
   Plot.
6. End..
```

Program:-

```
import Pandas as pd
import matplotlib.pyplot as plt
file_path = 'Employee.csv'
df = pd.read_csv(file_path)
categorical_column = 'Gender'
```



```
#Create a Pie Chart
plt.Pie (category_Counts, labels = category_Count.index])
plt.title (f Pie chart of {categorical_Column}')
plt.Show ()
import Pandas as Pd
import matplotlib.Pyplot as plt
file_Path = 'employee.csv'
df = Pd.read (file_Path)
categorical_Column = 'City'

import Pandas as Pd
import matplotlib.Pyplot as plt
Continuous_Column = 'Age'
plt.hist (df [continuous_Column], bins=10,
bins = blue bars
plt.title (f {Continuous_Column} Distribution')
plt.xlabel (Continuous_column)
plt.Show ()
import seaborn as sns
file_Path = 'Employee.csv'
df = Pd.read_csv(file_Path)
continuous_column= 'Age'
sns.kdeplot (df [continuous_Column], fill = True)
plt.title (f'Density Plot of {continuous_Column}')
plt.xlabel (Continuous_Column)
plt.Ylabel ('Density')
plt.Show ()
```

```
import Pandas Pd
import seaborn as sns
import matplotlib.pyplot as Plt
file_Path = 'employee.csv'
df = Pd.read_csv(file_Path)
Continuous_Column = 'Joining Year'
sns.rugPlot(df[Continuous_Column], height = 0.5)
Plt.title(f'Rug Plot of {continuous_Column}')
Pltx.label(continuous_Column)
Plt.show ()
```

VEL TECH	
EX No.	2
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	4
TOTAL (20)	12+3=16
SIGN WITH DATE	28/7/25

Result:- Thus, visualizing and Performing univariate analysis using continous and categorical data Piechart done Successfully..

04/08/25

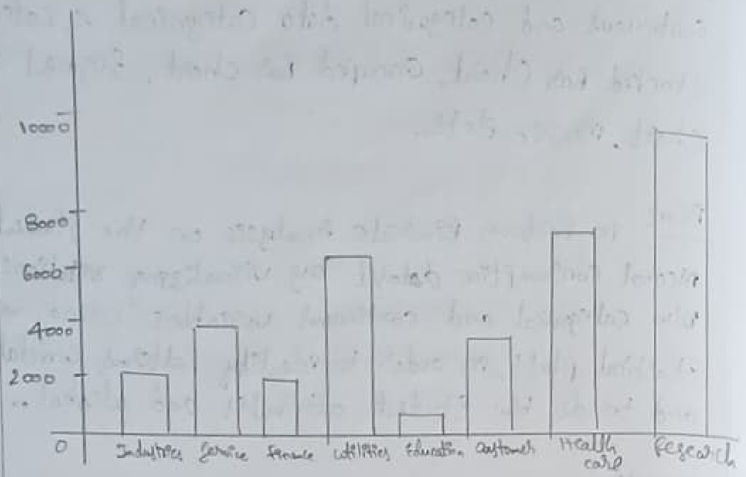
TASK-03

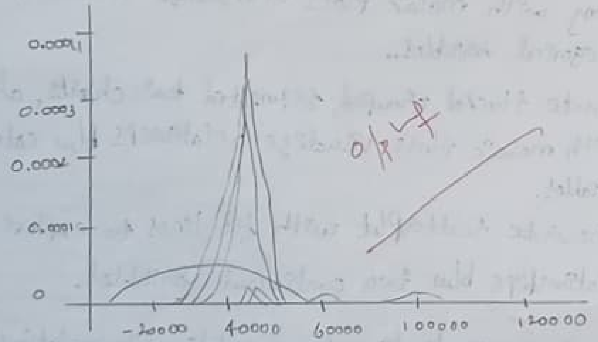
→ To visualize and Perform Bivariate analysis using Continuous and categorical data categorical vs. categorical Stacked Bar Chart, Grouped Bar chart, Segment Bar Chart, mosaic Plots..

Aim:- To Perform Bivariate Analysis on the Student Alcohol Consumption dataset by visualizing relationships b/w categorical and continuous variables using various Stastical plots, in order to identify Patterns, correlations and trends b/w Student attributes and alcohol..

Algorithm:-

1. Select Dataset: Choose a dataset containing both categorical and continuous variables..
2. Differentiate variables: Create Stacked, grouped, along with mosaic Plots to visualize relationships b/w categorical variables..
3. create Stacked, grouped, segmented bar charts, along with mosaic Plots visualize relationsps b/w categorical variables.
4. Generate Scatterplot with fit lines to explore relationships b/w two continuous variables.
5. Analyse visualization for insights into relationships distributors and patterns, drawing conclusions for further analysis (or) decision-making..





O/R

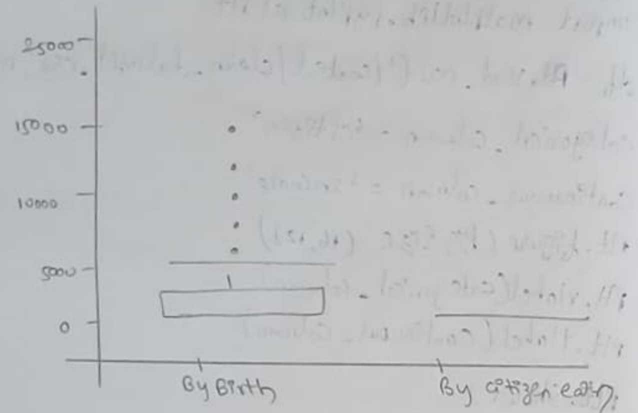
Program:-

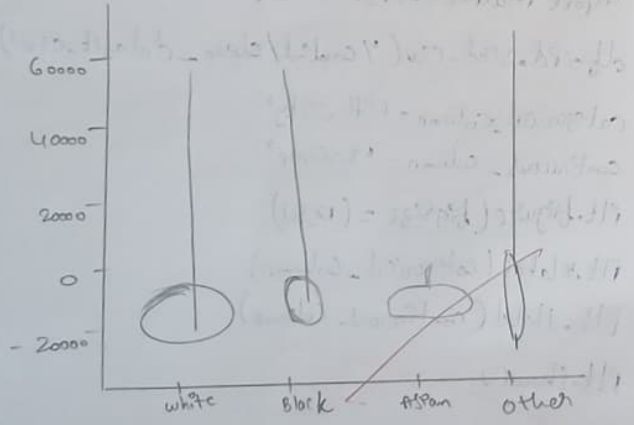
```
⇒ import Pandas as pd
  import matplotlib.pyplot as plt
  df = pd.read_csv('/content/clean_dataset.csv')
  industry = df ['Industry']
  income = df ['Income']
  plt. figure (figsize = (10,6))
  plt.bar(industry, income, color = 'skyblue')
  plt.xlabel ('industry')
  plt. ylabel ('income')
  plt.title ('income by industry')
  plt. xticks (rotation = 45)
  plt. show ()
```

```
⇒ Import Pandas as pd
  import seaborn as sns
  import matplotlib.pyplot as plt
  df = pd.read_csv('/content/clean_dataset.csv')
  categorical_column = 'ethnicity'
  continuous_column = 'Income'
  plt. figure (figsize = (12, 8))
  plt.xlabel (continuous_coulmn)
  plt.legend (title = categorical_column)
  plt.show()
```

```
⇒ import Pandas as pd
  import seaborn as sns
  import matplotlib.pyplot as plt
  df = Pd.read_csv ('/content/clean_dataset.csv', nrows =
  categorical_column = 'citizen'
  Continuous_column = 'Income'
  Plt.figure (fig size =(16,12))
  Plt.xlabel(categorical_column)
  plt.Ylabel (Continuous_Column)
  Plt.Show()
```

```
⇒ import Pandas as pd
  import seaborn as sns
  import matplotlib.Pyplot as plt
  df=Pd.read-csv('/content/clean_dataset.csv')
  categorical_column='Ethnicity'
  continuous_column = 'Income'
  Plt.figure(figsize = (12,8))
  Plt.xlabel (categorical_column)
  Plt.ylabel (continuous_Column)
  Plt.show()
```





Program:-

```
import Pandas as Pd
from joypy import joy plot
import matplotlib.pyplot as plt
df = pd.read_csv('/content/clean-dataset.csv')
categorical_column = 'Industry'
Continuous_column = 'Income'
plt.figure (figsize =(12,8))
joyplot(
        data = df,
        by= categorical_column,
        column = continuous_column,
        kind = 'kde',
        fill= True,
        linecolor = "black",
        grid = True,
         linewidth = 1,
         legend = True,
    )
plt.xlabel(continuous_column)
plt.ylabel (categorical_column)
plt.title (f Ridgement Plot for {continuous_column})
plot show ()
```

Program:-

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('/content/ DV task-4.csv')
```

```
snf.lmplot(data = df, x = 'Age', y = 'Income', height = 6,
line_kws = {'color': 'red'})
plt.xlabel('Age')
plt.ylabel('Income')
plt.show()
```

11/8/25

VEL TECH	
EX No.	1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	3
TOTAL (20)	13+3=16
.. N WITH DATE	

Result:- To visualize and perform Bivariate Analysis using continuous and categorical data is successfully implemented.

TASK-04

→ To visualize and Perform Multivariate analysis using Multiple variables involving multiple measures Scatterplot, Parallel coordinates, Line Graph, Stacked Bar chart..

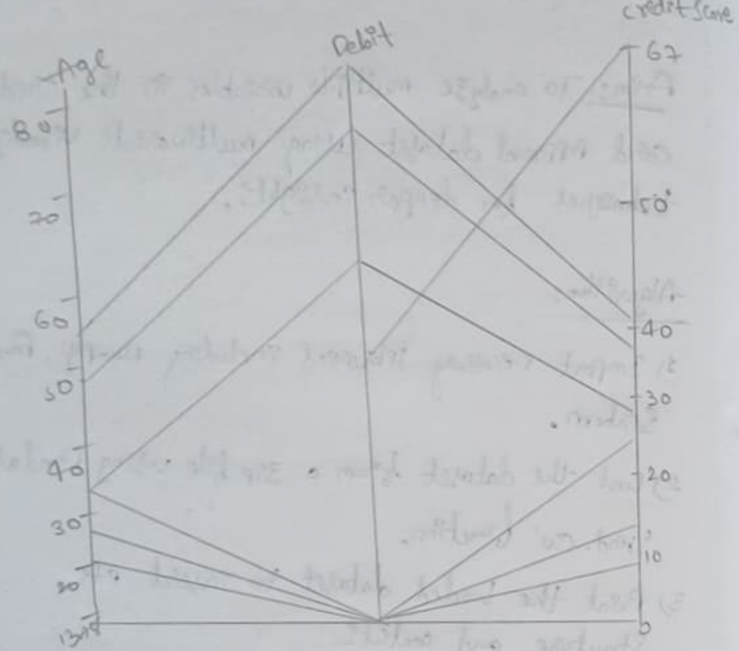
Aim:- To analyze multiple variables in the credit card Approval dataset using multivariate visualization techniques for deeper insights.

Algorithm:-

1) Import necessary libraries including Numpy, Pandas Seaborn.
2) Load the dataset from a zip file using Pandas 'read.csv' function.
3) Print the loaded dataset to insert its Structure and contents.
4) Select numeric columns (Age, Debt, YearEmployed), Create a Pairplot using Seaborn to visualize Pairwise relationships and display the Plot.
5) Use Plotly express to create a Parallel coordinat Plot.
6) Extra the first 20 entries of the dataset and Plot the age against Debt and Credit Score using a line Plot with different colours for Debt.

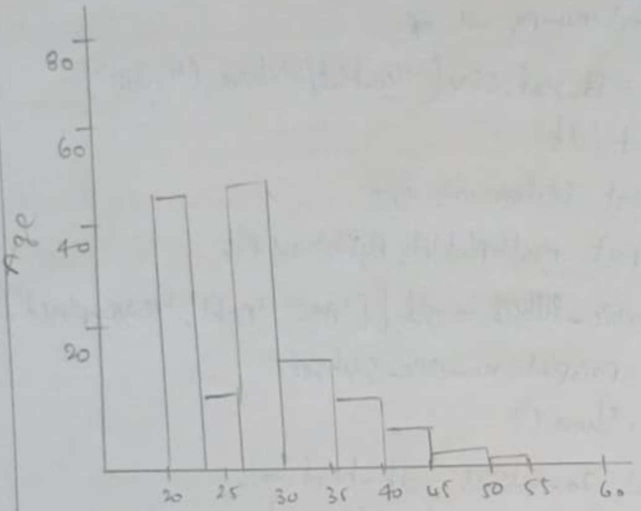
7) Group the first 20 entries by Age, summing up Debt and creditscore.

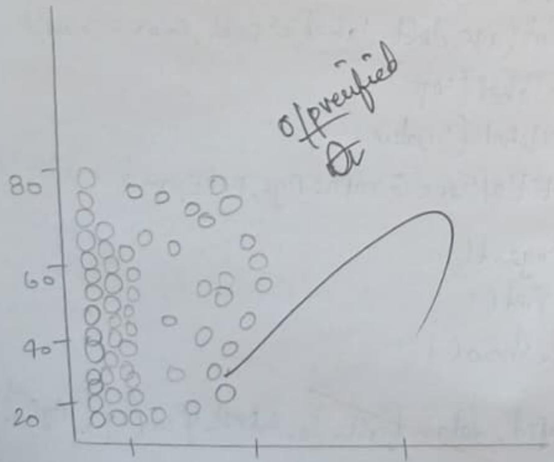
#output:-



Program:-

```
import Pandas as Pd
import numpy as np
df = Pd.read.csv("/content/archive (4).zip")
Print (df)
import seaborn as sns
import mathplotlib.Pyplot as plt
numeric_subset = df[['Age', 'Debt', 'YearsEmployed']]
sns.Pairplot(numeric_subset)
Plt.show ()
first_20_entries = df.head(20)
age = first_20_entries['Age']
debt = first_20_entries['Debt']
credit_score = first_20_entries['credit sore']
Plt.figure(figsize=(10,6))
Plt.Plot(age, debt, label='Debt', color='blue')
Plt.xlabel('Age')
Plt. ylabel ('value')
Plt.title('Line Graph: Age, Debt, and credit score')
Plt.legend()
Plt.grid()
Plt.show()

grouped_data = first_20_entries.groupby('Age').sum()
grouped_data.Plot(kind='bar', stacked=True,
                  figsize=(12,8))
Plt.xlabel('Age')
```





O/p verified

```
Plt.Ylabel('value')
Plt.title('Stacked Bar chart: Debt and credit Score)
Plt.Legend(title='Attribute')
Plt.Show()
```

VEL TECH	
EX NO.	04
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	3
TOTAL (20)	12+3=15
SIGN WITH DATE	18/8/25

Result:- To visualize and Perform multivariate Analysis using multiple variables involving multiple measures is successfully implemented

TASK-5(A)

→ To design and Perform visualization for Trees.

Tools : Tableau, Language : python..

Aim:- To design and visualize a Treemap of Car brands and their total Prices using the car dataset..

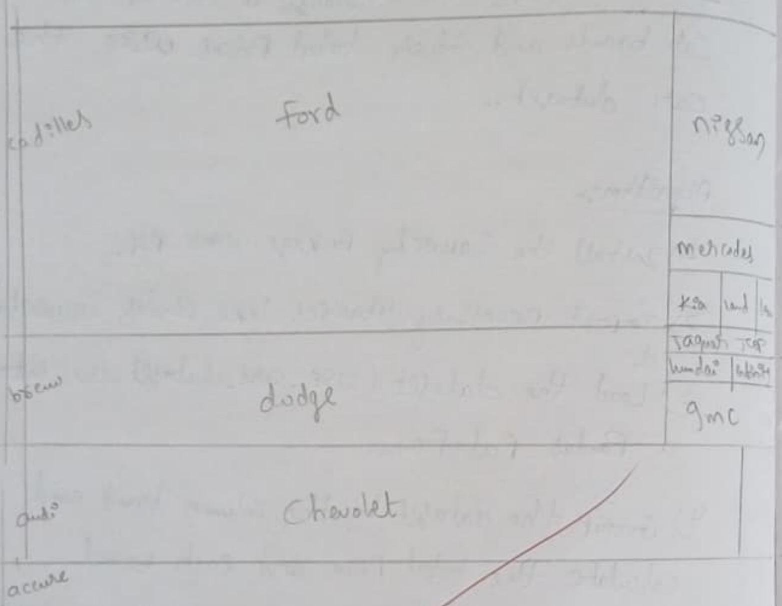
Algorithm:-

1) Install the squarify Package using pip.
2) Import necessary libraries like Pandas, Squarify..
3) Load the dataset (USA_cars_dataset.csv) into a Pandas Dataframe.
4) Group the dataset by the column brand and calculate the total Price and each brand..
5) create a figure for visualization using matplotlib.
6) Adjust transparency (alpha = 0.4), remove axes,
7) Display the Plot using plt.show().

Code:-

```
pip install squarify
import Pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

#output:-



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

18/08

TASK - 5(B)

Aim:- 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', 'Seaborn', 'Squarify' and 'plotly.express'..

3) Read the dataset from '/content/USA-car-dataset.csv' into Pandas Dataframe 'df'.

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

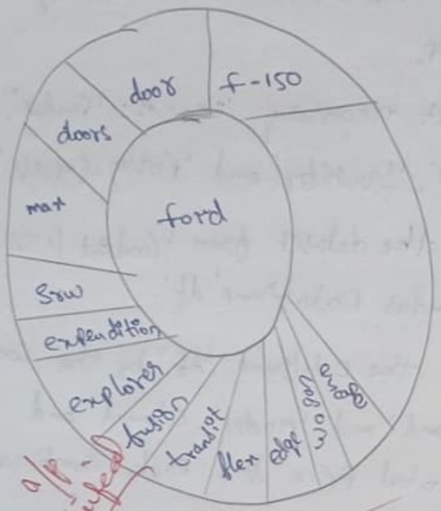
5) Create a Sunburst plot using Plotly Express ('px.sunburst()'):
   - Specify the Dataframe 'sunburst_data'.
   - Set the title of the plot as 'sunburst'.

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

Program:-

```
pip install squarify
pip install plotly
import pandas as pd
import numpy as np
```

#output:-



o/p - verified

```
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'])
fig = px.sunburst(sunburst_data.path = ['brand','model'],
fig.show()
```

VELTECH	
EX No.	04
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	4
TOTAL (20)	13+4=17
SIGN WITH DATE	18/8/21

Result:- To design and perform visualization for trees is successfully completed..

25/06

TASK-06

→ To design and Perform visualization for Graph and Networks..

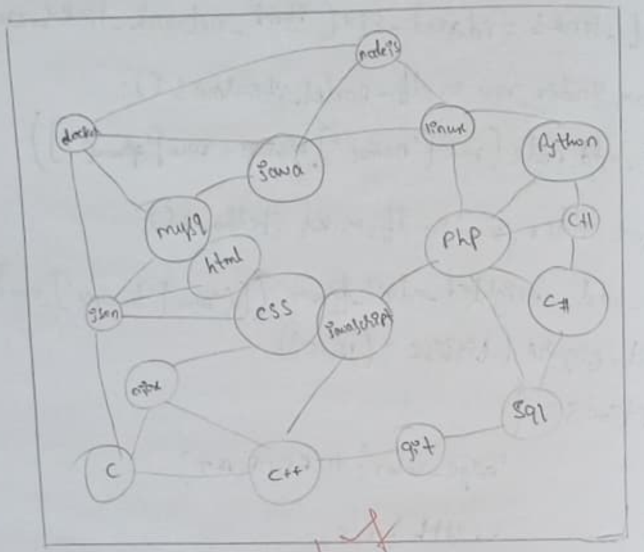
Aim:- To visualize and analyse the Stack overflow Tag Network using a force-based layout in Python Study relationships and importance of technical tags..

Algorithm:-

- Import Pandas libraries,
  Import matplotlib.pyplot as plt for data visualization
  create an empty graph object G and set an attribute.
- Add nodes to the graph G:
  Iterate through the rows in df-nodes.
  For each row, add a row to the 'name', 'group', 'node'.
- Add weighted edges to the graph G:
  Iterate through rows in df-nodes.
  For each row, add a node to the graph with a 'name', 'group' and 'node sise' attributes.
- create a figure with a Specified size.
  Define visualization option like edge color, width, node labels and font weight.
- Draw graph network:
  use nx.draw to visualize the graph.
  customise the node colors and sizes based on the Calculated values.

Code:-

```
import networkx as nx
import matplotlib.pyplot as plt
import pandas as pd
G= nx. Graph (day = "Stackoverflow")
df-nodes = pd.read-csv ('Stack_network_nodes. csv")
df-links = pd.read-csv('Stack_network_links. csv')
for index, row in df-nodes. iterrows ():
G.add_node (row['name'], group = row['group'])

for index, row in df_links. iterrows ():
G. add-weighted-edges-from ([(row['source'], row [target]
plt. figure (figsize = (15,15))
options = {
            'edge-color': # FFDEA2',
            'width': 1.5,
            'with_labels': True,
            'font-weight': 'regular',
            }
colors = [color-map [G.nodes [node] [group]] for node

sizes = [G.nodes[node]['nodesize'] * 25 for node G]

nx.draw(G, node_color = colors, node-size = sizes,
          pos = nx. spring_layout (G, k = 1.5, iteration = 1.5)
```



8/8

```
ax = plt.gca()
ax.collections[0].set_edgecolor("#555555")
plt.show()
```

VEL TECH	
EX No.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	5
TOTAL (20)	18
SIGN WITH DATE	

Result:- To design and perform visualization for graphs and networks force based layout..

08/09

TASK-7[A]

Aim:-

To genrate insights from text data using Text Network Analysis and visualization tool such as Tagcrowd and the word cloud Package in Python, by creating word clouds and graphical visualization to identify frequently occuring terms and their importance.

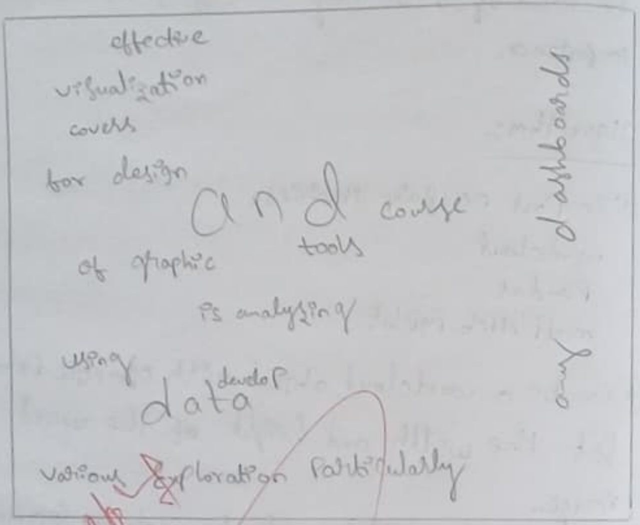
Algorithm:-

⊛ Import necessary libraries:
  wordcloud
  Pandas
  matplottib.pyplot

⊛ Create a wordcloud object with specific Parameters, set the width and height of the word cloud image..
Provide a list of stopwards to exclude common words.

⊛ Display the word cloud image:
use plt.imshow to display the word cloud. Turn off the axis labels. Adjust the layout to minimise padding..

⊛ End of Algorithm;



Code:-

```
from wordcloud import word cloud
import matplotlib.pyplot as plt
details = "Data visualization is the visual and interative
exploration and graphic representation of data of
any type. This course covers data visualization
concepts, practices, tools particularly for analysing
and presenting business data."
wordcloud = wordcloud (width = 800, height = 800,
                       background_color = 'white',

plt.figure (figsize = (5,5), facecolor = None)
plt.imshow (word cloud)
plt.axis ("off")
plt.tight_layout (pad = 0)
plt.show ()
```

VEL TECH	
EX No.	
PERFORMANCE (5)	[illegible]
RESULT AND ANALYSIS (5)	[illegible]
VIVA VOCE (5)	[illegible]
RECORD (5)	[illegible]
TOTAL (20)	[illegible]
...TH DATE	

Result:- To generate insight using Text network analysis and visualization.

TASK - 7[b]

Aim:- TO generate insight from text shows how words in a text are connected providing insights into word relationship..

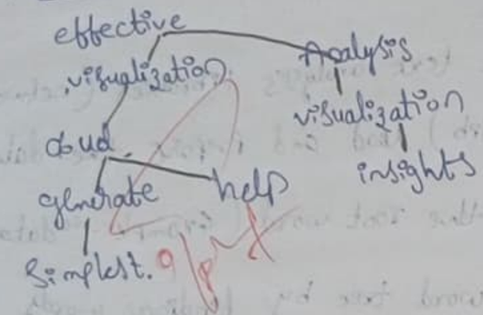
Algorithm:-

- (*) Import text analysis libraries (network, matplotlib) load and prepare the dataset.
- (*) Select the root word (example - "data")
- (*) Build a word tree by finding words that appear before, after the root word.
- (*) create a graph structure using network data visualize the tree graph using matplothib.

Program:-

```
import nlt
import matplotlib.PyPlot as plt
import network as nx
from nltk corpus import stop words
from nltk to kenize - import word - to kenize
nltk download ("Pankt")
nltk download ("Stepwords")

text = Data analysis " is import Data
visualization clear
```

output:-



```
token = word - tokenize (text toward ())
stop_words = set (stopwords, words (english))
bigrams = list (nltk, bigrams (filtered_tokens))
root_word = "data"
G = nx. Digraph ()
for w1, w2 in biagrams
plt. figure (fig size = (36))
nx. draw_network x (G, with_labels = True, node -
                                        color = skyblue,
plt. title ("word Tree Graph for rootword: (root_word)",
  font size = 14)
plt. show ()
```

VEL TECH	
EX No.	7a
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	5
TOTAL (20)	18
SIGN WITH DATE	

8/9

Result:- Thus, the generate right insight using network Analysis and visualization utilizing word tree is done and executed successfully.

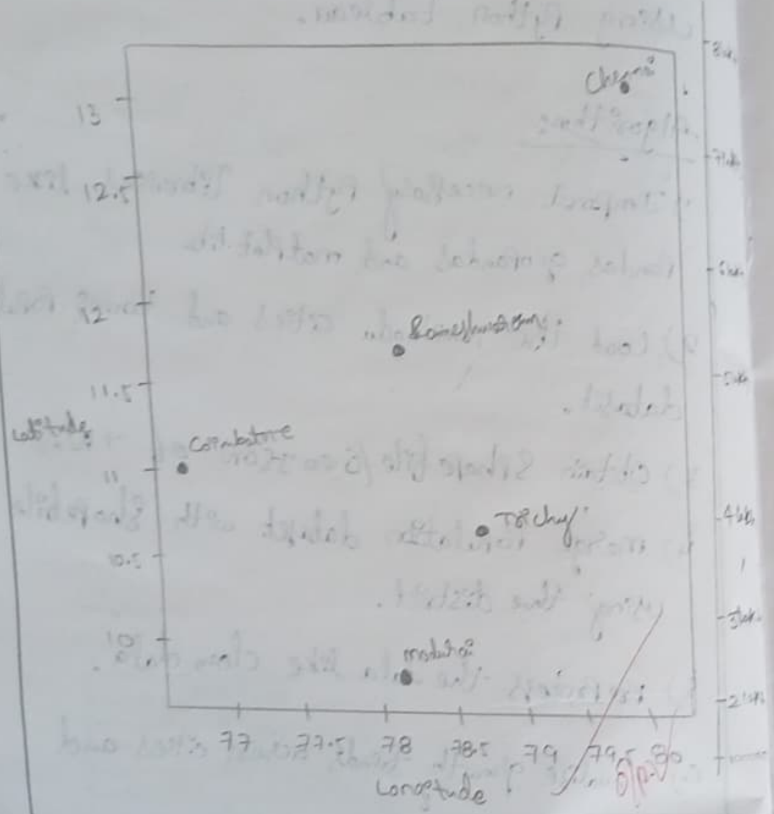
TASK-8

→ Spatial and Geospatial Data-Analysis and visualization.

Aim:- To analyse and visualize Spatial and Geospatial data of cities and towns in TN, using Python tableau.

Algorithm:-

1) Import necessary Python libraries like pandas, geopandas and matplot lib.
2) Load the Tamilnadu cities and towns Population dataset.
3) Obtain Shapefile/GeoJSON of TN.
4) merge Population dataset with Shapefile using the district.
5) PreProcess the data like clean data.
6) visualize growth trends across cities and districts.
7) Generate insights (which cities grew fastest, spatial clusters of Population growth).



Program:-

```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt

data = pd.read_csv ("tn_population.csv")

tn_map = gdp.read_file ("tamilnadu_districts.shp")

merged = tn_map.merge(data, left_on="District",
                     right_on="Dist of city")

merged.plot(column="2011-03-01", cmap="OrRd",
            legend=True, figsize=(10,6),
            edgecolor="black")

plt.title ("Tamil Nadu Population Distribution")

plt.show()
```

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	

Result:- Thus Analysis shows how population distribution in TN has changed over census years, we can visualize these trends interactively and understand regional growth patterns across TN..

29/09

TASK-09

→ TO Analyse and visualize Time oriented D Analysis to identify systemic patterns in data that helps to form trends, cycles and to forecast the data..

Aim:- TO Analyse and visualize time-oriented data in order to identify systemic patterns such as trends, cycles, (or) seasonal variances and to forecast future data points using Line graph..

Algorithm:-

1) Import the required libraries (Pandas, matplotlib, seaborn).

2) Load the time-oriented dataset (montly sales)..

3) Convert the date column into a date time object and set it as index.

4) Perform Exploratory Data Analysis:
   - Plot line graphs to identify trends.
   - Add trend lines for long-term direction.

5) Decompose the time series into trend.

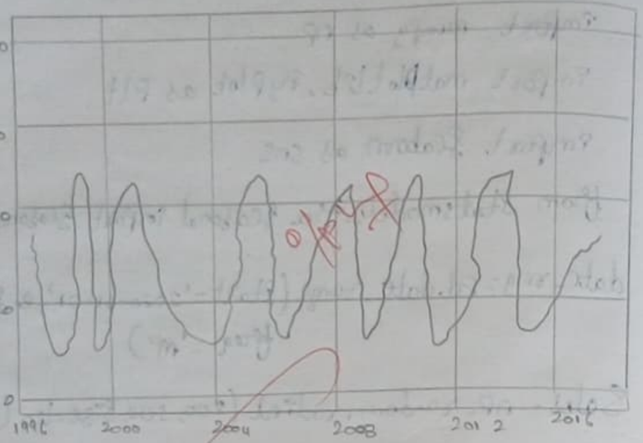
6) visualize and interprets results..

7) Conclude based on observed Patterns and forecast accuracy..

#Program:-

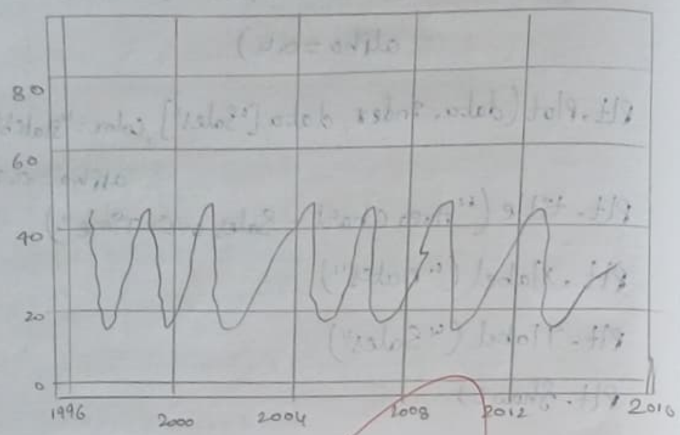
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.seasonal import seasonal-decomp

date_rng=pd.date_range(start='2020-01-01', end='23-12-01
                        freq='M')

Sales=np.random.radint(200,500,size=len(date_rng)
                        +np.linspace(10,200,len(date_rng))

date=pd.Dataframe({'Date': date_rng,'Sales': Sales})
date.set_index('Date', inplace=True)
plt.figure(fig size=(10,5))
plt.plot(data.index,data['Sales'],marker='o', label=
                        'monthly Sales')
plt.title("monthly Sales over Time")
plt.xlabel("Date")
plt.ylabel("Sales")
plt.legend()
plt.grid()
plt.show()
```

Line graphe of Temp over time.



```
Plt. figure (fig size =(10,5))
Plt. fill_between(data.index, data['Sales'], Color = "Skyblue",
                    alpha =0.4)
Plt. Plot(data. index, data['Sales'], color = "Slateblue",
                                        alpha = 0.7)
Plt. title ("Area Chart - Sales over Time")
Plt. xlabel ("Dates")
Plt. Ylabel ("Sales")
Plt. Show ()
result= Seasonal_decompose(data['Sales'], model=
                    'addictive', Period =12)
result.Plot()
Plt. Show ( )
data ['forecast'] = data ['Sales'].rolling (window=6).
                                                mean ()
Plt. figure (fig size = (10,5))
Plt. Plot (data['Sales'], label = "Actual Sales")
Plt. Plot (data ['forecast'], color = 'red' label = "Moving
                                        Avg Forecast")
Plt. title (" Sales forecast using moving Average")
```

Temperature trend with Trend line



```
Plt.xlabel ("Date")
Plt.ylabel (" Sales")
Plt.legend ()
Plt.Show ()
```

VEL TECH	
EX No.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	5
TOTAL (20)	17
SIGN WITH DATE	

29/7

Result:- Thus, the Program was executed Successfully and analyses time-oriented data reveals a Clear upward trend in Sales across the year..

Batch Mates :

1. KURAKUL SANJAY SIVA KUMAR
2. GORANTLA ABHINAY
3. BUSI VENKATA SUNIL KUMAR REDDY
4. MALEPATI YESHWANTH CHOWDARY
5. NUKALA SAI SRI VIVEK
6. CHAPPIDI MUNENDRA NAIDU
7. KOTHA VENKATA NAGA BALA MANIKANTA
8. NALLABOTHULA VENKATA ROHITH
9. KONETI HARISH
10. KUSAM ANANDA REDDY

13/10

USE CASE – 1

Earthquake and Geospatial Data Analysis

Aim:

To analyze earthquake data stored locally, visualize their global distribution, and identify high-risk seismic zones using Python's geospatial and visualization libraries

Algorithm:

1. **Download Dataset:**
   - Visit: USGS Earthquake Data (CSV)
   - Save it in the same folder as your Python script.
   - Rename it to `earthquake_data.csv`
2. **Load the CSV file using `pandas`:**
   Clean and filter relevant columns (`latitude`, `longitude`, `mag`, `place`).
3. **Perform Analysis:**
   - Compute average, maximum, and minimum magnitudes.
   - Identify the strongest earthquake location.
4. **Visualize Data:**
   - Use `matplotlib` to plot histogram of magnitudes.
   - Use `folium` to create an interactive world map with earthquake markers.
5. **Save Outputs:**
   - Display stats in console.
   - Save the interactive map as `earthquake_map.html`.

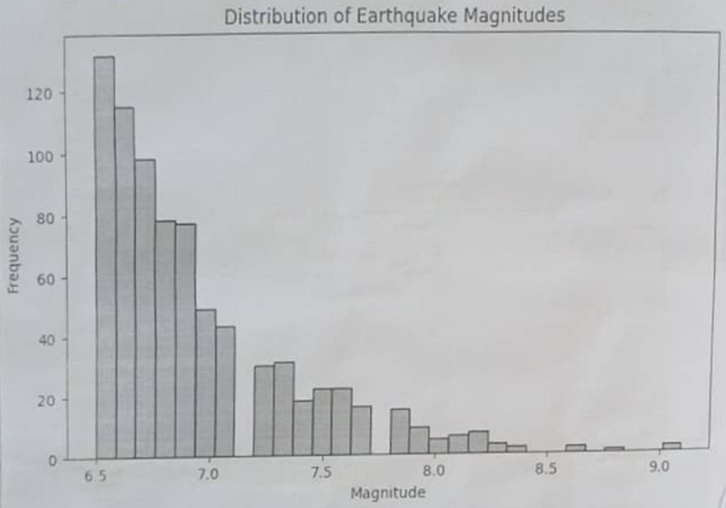
Program:

```
import pandas as pd
import folium
import matplotlib.pyplot as plt

# Load the dataset
file_path = "earthquake_data.csv"
data = pd.read_csv(file_path)
```

```
# Display first 5 records
print("First 5 Records:")
print(data[['title', 'magnitude', 'date_time', 'latitude', 'longitude']].head())

# Drop missing values
data = data.dropna(subset=['magnitude', 'latitude', 'longitude', 'title'])

# Basic statistics
avg_mag = data['magnitude'].mean()
max_mag = data['magnitude'].max()
min_mag = data['magnitude'].min()
strongest_quake = data.loc[data['magnitude'].idxmax()]

print(f"\nAverage Magnitude: {avg_mag:.2f}")
print(f"Maximum Magnitude: {max_mag:.2f}")
print(f"Minimum Magnitude: {min_mag:.2f}")
print(f"Strongest Earthquake: {strongest_quake['title']}")

# Plot histogram of magnitudes
plt.figure(figsize=(8, 5))
plt.hist(data['magnitude'], bins=30, color='lightcoral', edgecolor='black')
plt.title("Distribution of Earthquake Magnitudes")
plt.xlabel("Magnitude")
plt.ylabel("Frequency")
plt.grid(alpha=0.3)
plt.show()

# Create interactive map
m = folium.Map(location=[0, 0], zoom_start=2, tiles="CartoDB positron")

for _, row in data.iterrows():
  folium.CircleMarker(
    location=[row['latitude'], row['longitude']],
    radius=max(row['magnitude'], 1),
    color='crimson' if row['magnitude'] >= 5 else 'orange',
    fill=True,
    fill_opacity=0.7,
    popup=f"Location: {row['title']}\nMagnitude: {row['magnitude']}"
  ).add_to(m)

# Save interactive map
m.save("earthquake_map.html")
print("\nInteractive map saved as 'earthquake_map.html'")
```

OUTPUT:

Strongest Earthquake: M 9.1 - 2011 Great Tohoku Earthquake, Japan



VEL TECH	
EX NO.	uc
PERFORMANCE (5)	4
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	2
TOTAL (20)	5
SIGN WITH DATE	17

13/10

Result:

Thus, the earthquake data was successfully analyzed and visualized, showing that most earthquakes were minor to moderate, with the strongest near Japan, and highlighting active zones around the Pacific Ring of Fire.

27/10

USECASE – 2

Interactive dashboard, to convey the data that had been collected over the financial year

AIM:

To create an interactive dashboard using Python to visualize and analyze financial year data trends such as revenue, expenses, and profit across months. The dashboard will help stakeholders make data-driven decisions.

ALGORITHM:

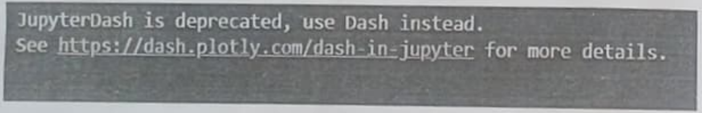
1. Load or simulate financial data with columns: Month, Revenue, Expenses, and calculate Profit.
2. Use pip to install Plotly, Dash, and pyngrok in Colab.
3. Design layout with dropdowns, graphs, and interactive components.
4. Host the dashboard using ngrok and display public URL in Colab.
5. User can select different views of the data using dropdowns.

PROGRAM:

```
!pip install dash==2.14.1 jupyter-dash pyngrok pandas plotly --quiet

import pandas as pd

from dash import dcc, html, Input, Output

from jupyter_dash import JupyterDash

import plotly.express as px

from pyngrok import ngrok

df = pd.DataFrame({

  "Month": ["Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec", "Jan",
"Feb", "Mar"],

  "Revenue": [12000, 15000, 13000, 17000, 16000, 14000, 18000, 17500, 16500,
15500, 16000, 19000],

  "Expenses": [8000, 9000, 8500, 9500, 10000, 9200, 9700, 9400, 9100, 8800, 8600,
9300]

})

df["Profit"] = df["Revenue"] - df["Expenses"]
```

OUTPUT:

See https://dash.plotly.com/dash-in-jupyter for more details.



```
JupyterDash is deprecated, use Dash instead.
See https://dash.plotly.com/dash-in-jupyter for more details.
```

```
app = JupyterDash(__name__)
app.layout = html.Div([
    html.H2(" Financial Dashboard", style={'textAlign': 'center'}),
    dcc.Dropdown(
        id='metric', options=[{'label': m, 'value': m} for m in ['Revenue', 'Expenses',
'Profit']],
        value='Revenue', clearable=False, style={'width': '50%', 'margin': 'auto'}
    ),
    dcc.Graph(id='chart'),
    html.Div(id='total', style={'textAlign': 'center', 'marginTop': 20})
])
@app.callback(
    Output('chart', 'figure'), Output('total', 'children'),
    Input('metric', 'value')
)
def update(metric):
    fig = px.line(df, x='Month', y=metric, title=f'Monthly {metric}')
    return fig, f"Total {metric}: ₹{df[metric].sum():,.2f}"
url = ngrok.connect(8050)
print(f"Dashboard URL: {url}")
app.run_server(mode='external', port=8050)
```

VEL TECH	
EX NO.	[illegible]
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	5
TOTAL (20)	18
SIGN WITH DATE	

RESULT:

Thus, To create an interactive dashboard using Python to visualize and analyze financial year data trends such as revenue, expenses, and profit across months. The dashboard will help stakeholders make data-driven decisions was successfully completed.