Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology

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: Shaik Mohammed Shahid

VTU.NO: 24858

REG.NO: 23VECL0055

BRANCH: CSE (AIML)

YEAR/SEM: 3rd | 5th

SLOT: S12 L4

Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology

NAAC A++

CATEGORY-1 DEEMED TO BE UNIVERSITY by the UGC

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

ACADEMIC YEAR 2025-26 (SUMMER SEMESTER)

BONAFIDE CERTIFICATE

NAME	: Shaik Mohammed Shahid	BRANCH	: CSE(AIML)
VTU NO.	: 24858	REG.NO.	: 23UECL0055
YEAR/SEM	: 3rd / 5th	SLOT NO.	: S12 L1

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).

R.T. 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**

INDEX

NAME: Mohammed Shahid ROLLNO.:

STD.: DIV./ SEC.: SUBJECT:

INDEX

Date	Title	Page No.	Marks	Faculty Signature
21/07/25	Exploration of Data Visualization Tools like Tableau, Python libraries, D3.js	1	16	21/7
28/07/25	To visualize and perform Univariate analysis using continuous and categorical data	3	16	28/7
04/08/25	To visualize and perform Bivariate analysis using continuous and categorical data	9	16	4/8
11/08/25	To visualize and perform Multivariate analysis using Multiple variables involving Multiple measures	12	16	11/8
18/08/25	To design and perform visualization for Trees	15	16	18/8
25/08/25	To design and perform visualization for Graphs and Networks	19	18	25/8
08/09/25	To generate insight using Text Network Analysis and Visualization	22	17	8/9
15/09/25	To analyze and visualize Spatial and Geospatial data	26	17	15/9
29/09/25	To analyze and visualize Time Oriented Data	29	18	29/9
13/10/25	Use Case: ~~Earthquake~~ and Geospatial Data Analysis	32	17	12/10
27/10/25	Use Case: ~~2~~ Performance of different company Department over year.	36	18	27/10

complete

Total Marks: 184/220

B.T. Thakur

Signature of Faculty

the oriented Data Analysis to identify systemic.

Task-1 21/7/25

Mango Plant Disease Visualization

Exploration of Data Visualization Tools - Tableau/ Python Ge

* connecting Data Set
* Preparation of Data

Aim: To visualize the distribution of differen mango Plant disease and Explore their metadata Using Python or other visualization tods.

Algorithm:

① import Dataset:

load the data set Containing folders or records of mango Plant images classified by disease condition.

② Extract metabase:

* List all image files from each disease class folder. Eg:- Anthracnose, sooty mold.
* Create a structured table with attributes like, image name, class label

Data set table for mango plant:

Plant name	Type	Height (cm)	Sunlight	water Needs	Health condition
MangoTree1	fruit plant	150	high	Medium	Anthracnose
MangoTree2	fruit plant	140	high	high	Sooty mold
MangoTree3	fruit plant	130	Medium	Medium	Powdery Mildew
MangoTree4	fruit plant	160	high	low	Healthy
MangoTree5	fruit plant	155	Medium	high	Bacterial canker
Tree 6	fruit plant	145	low	Medium	Black spot
Tree7	fruit plant	135	high	Medium	Sooty mold
Tree 8	fruit plant	125	Medium	low	Healthy
Tree 9	fruit plant	165	high	high	Powdery mildew
Tree 10	fruit plant	120	low	Medium	Anthracnose

Preprocess and clean data:

Ensure only valid image formats are used (jpg, png)

Create Summary Table:

count the number of images in each class.

Display sample records in tabular format

Visualization (optional):

Use bar charts to show the number of images per diseases class using Python

Program:

```
import pandas as pd
import numpy as np
a=pd.read_csv("data_2023-01-...csv")
Print(a)
Sh=a.shape
Print(sh)
a.info()
a.describe()
Print(a.head())
pd.isna(a).s
a.dropna()
a.duplicated()
a.drop_du
```

VEL TECH	
EX No.	1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	4
TOTAL (20)	16
SIGN WITH DATE	

Result: Thus, to visualization the distribution of different mango plant disease and explore their meta data using Python or other visualization tools was executed successfully

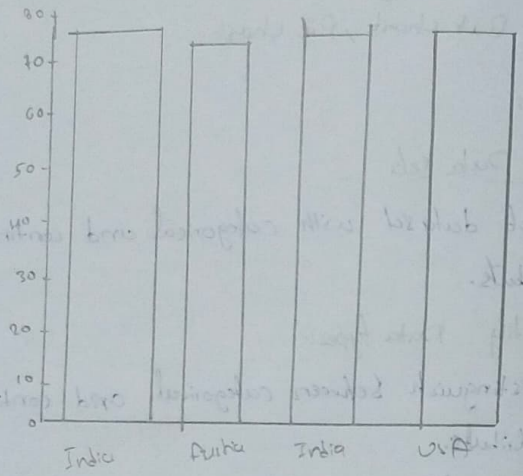
Task-2

28/7/25

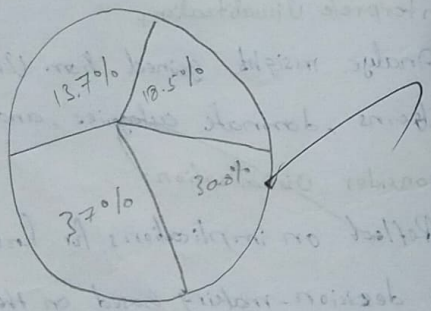
Aim: To visualize and Perform Univariate analysis using continuous and categorical data - Bar chart, Pie chart.

Algorithm:

1. Choose Data set
   * Select dataset with categorical and continuous attributes.
2. Identify Data types:
   * Distinguish between categorical and continuous attributes.
3. construct scatter, lines, strip, swarm plots:-
   * Utilize appropriate plots for continuous data analysis, considering relationships, trends, & distributions.
4. Interprete Visualization:
   * Analyse insights gained from visualization, identify patterns, dominate categories, and correlations.
5. consider visualization:
   * Reflect on implications for further exploration or decision-making based on the analysis.
6. Document finding:
   * Summarize finding and observation for reporting or Presentation Purposes.



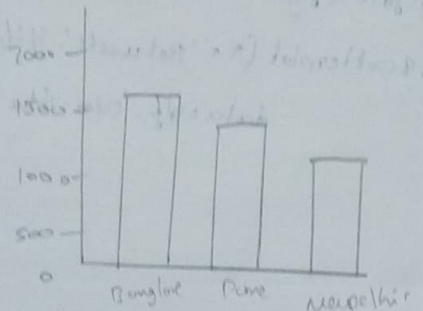
o/p



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')
        ('Likes').sum().sort_values(ascending
                =False).head(5)

plt.figure(figsize=(4,4))
Platform_likes_top5.plot(kind='pie', autopct='%1.1
        f%%', startangle=140, colours=['skyblue',)

plt.title('Top 5 Platforms by Total Likes')
plt.ylabel('')
plt.show()
top5 countries = df.nlargest(5, 'Likes')

#create a ~~chan~~ bar chart
plt.figure(figsize=(10,6))
plt.bar(top5_countries['country'], tops=countries
    ['Likes'], color=['sky blue', 'light coral', 'lightgreen',
        'lightsalmon', 'light steelblue'])
plt.ylabel('Total Likes')
plt.xlabel('country')
plt.title('Top 5 countries by total likes')
plt.show().
```

```
data = {
    'Text': ['enjoying a beautiful day at the park!'],
    'Retweets': [15, 5, 2, 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)
```



0/1

Task - 2(b)

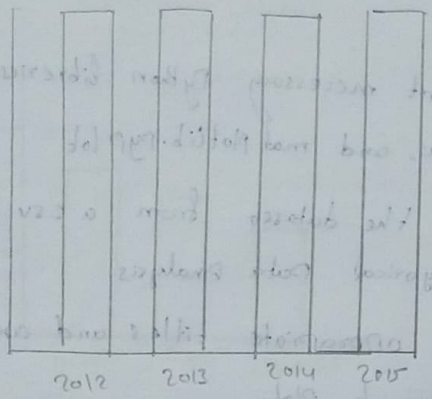
Aim: To Perform Univariate analysis on a dataset containing emp details by identifying categorical and continuous data attributes.

Algorithm:

1. Start
2. Import necessary Python libraries: Pandas, and matplotlib.pyplot
3. Load the dataset from a csv file.
4. categorical Data Analysis
5. Add appropriate 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=catogiry_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 [continow_colum], bins=10,
bins = blue bors
plt.title (f{continuous_colum} Distribution')
plt.xlable (continous_colum)
plt.show()
import Pandas pd
Import seaborn as sns
import matplotlib.Pyplot.us plt
file_path = 'employee.csv'
df = pd.read_csv(file_path)
continuous_column= 'joining year'
```

```
sns.rugplot(df[continuous_cdam ], height=0.5)
plt.title (f'Rugplot of h continous Columg')
plt.ltx.lable (continous.column)
plt.show()
```

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

28/1/24

Result: Thus visualizing and Performing univariate analysis Using contineous and categorical data - Barchart, Pie chart done Successfully.

4/8/25

Task-3

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 variations Using various statical Plots.

Algorithm:

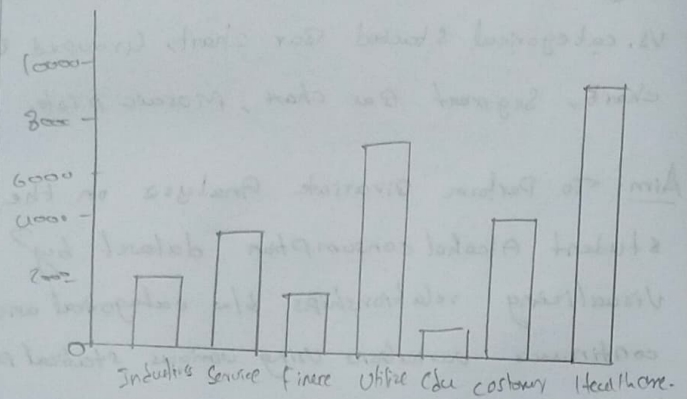
1.) Select Dataset choose a dataset countaining both catagoriacal and continuous variables.

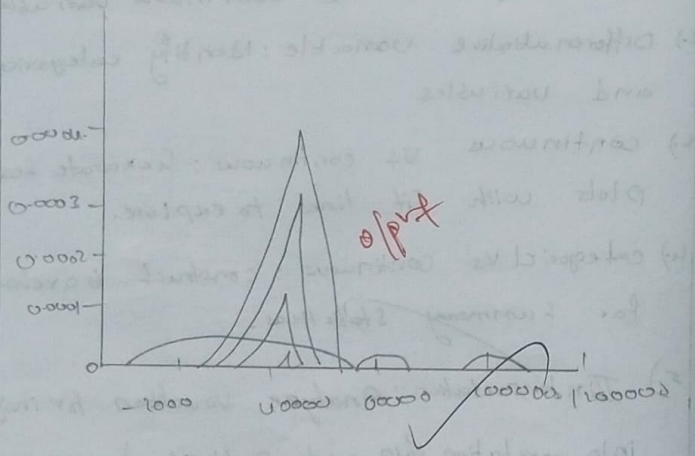
2.) Differentiative variable: Identify categorical and variables

3.) continuous Vs continuous: Genarate Scatter Plots with Fit lines to explore.

4.) categoricl Vs continuous: construct barchart for summeary Statisticas.

5.) Interpretation: Analyse visualtion for insight into relationship and Patterns, drowing conclosins for Further analysis ordecision making.





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_ colour = sky, blue)
  plt.xlabel ('Industry')
  plt.ylabel ('Income')
  plt.show()
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt.
df = pd.read_csv
categorical_column = 'Ethnicity'
plt.xlabel (categorical_column)
plt.ylabel (continuous_column)
plt.show()
```

```
import pandas as pd
import seaborn as ns
import matplotlib pyplot as plt.
df = pd.read_csv ('/counter /clean_dataset.
                   csv")

categorical_coloum = 'ethicity'
continuous_colom = 'Income'
```

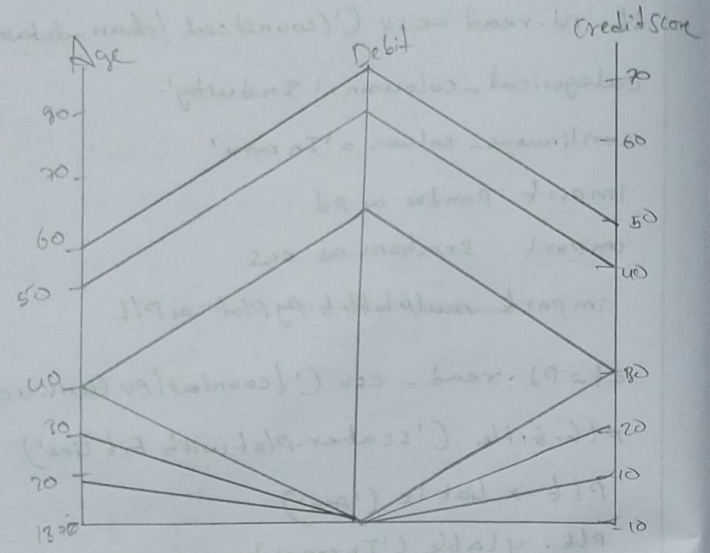
```
plt.show()
import pandas as pd
from joypy import joyplot.
import matplotlib.pyplot as plt
df= pd.read_csv ('/counstcat/clean_dataset.csv')
categorical_coloumn='Industry'.
continuous_colum = 'Income'
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt.
df=pd.read_csv ('/countcat/pv task.ucsv')
plt.title ('scater.plot with Fit line')
plt.xliable ('Age')
plt.ylable ('Income')
plt.show()
```

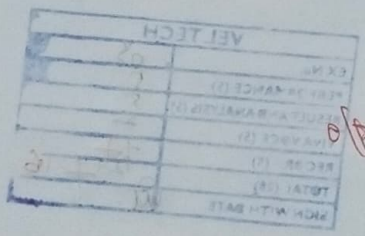
VEL TECH	
EX No.	03
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	4
TOTAL (20)	12+4=16
SIGN WITH DATE	

4/8

Result: To Visualize and Perform Bivariate Analysis using continous and categorical data is sucessfully implemented.

output





o/p ✓

Task-4

12/8/25

Aim: To visualize and perform multivariate analysis using multiple variable involving multiple measures.

Algorithm:

1) Import necessary libraries includes Numpy Pandas, seaborn.

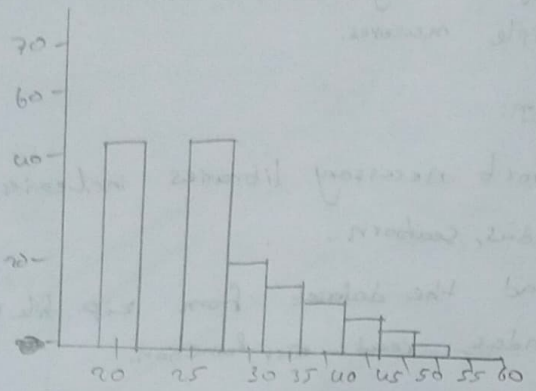
2) Load the dataset from zip file using Pandas, read_csv function.

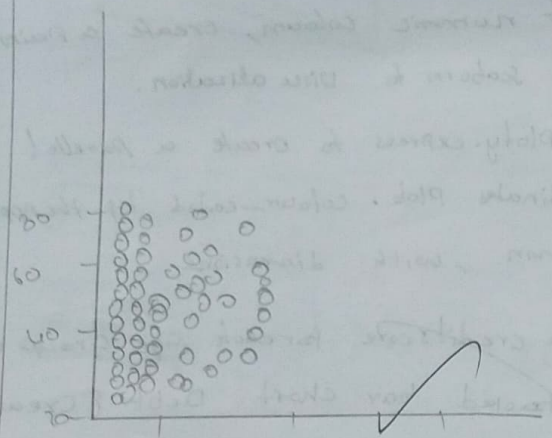
3) Print the loaded dataset to inspect its structure.

4) Select numeric coloum, create a PairPlot using seaborn to visualization.

5) Use Ploty.express to create a Parellel coordinates plot, colour_coded by the Approved colourmn, with dimensions.

6) The ~~credit~~ score for each Age Group create a ~~stacked~~ bar chart Debt & Create score for each group & display the chal





Program:

```
import numpy as np
import pandas as pd
df = pd.read_csv ("/content/archive (1).zip)
print(df)
import seaborn as sns
import matplotlib.pyplot as plt
numeric = subset = df[['Age', 'Debt', 'years
                       Employed']
plt.show()
import pandas as pd
import plotly.express as px.
attributes = ("Age", "Debt", "credit score")
fig = px. parellel_coordinate (df, colour=
                    Approved". fig.show().
plt.plot (age, dept. lable = "Debt.colour
                    = blue)
plt.xlabel ('Age')
plt.ylabel ('value")
plt.title ('line graph', Age Debt
           credit_score)
plt.legend()
plt.goind (True)
plt.show().
```

```
grouped = data.plot (kind = 'bar', stacked = True, figsize=(12,8))
plt.xlabel (Age)
plt.ylabel ('value')
plt.title ('stacked Bar chart : Debt & Credit score
                                by Age')
plt.legend (title = ('Attribute'))
plt.show()
```

VEL TECH	
EX No.	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	3
VIVA VOCE (5)	2
RECORD (5)	4
TOTAL (20)	14
SIGN WITH DATE	

Result: To visualize & Perform multiple analysis using melting variable involving multiple measures done successfully.

Task-5A 18/8/25

Aim: construct a tree map display on a real-world dataset.

Algorithm:

1.) Install the squarify package using pip.rify

2) Import necessary libraries such as pandas, matplotlib.pyplot & squarify.

3) Read the data set from /content USA.csv

4) Group the Data Frame df

5) create the figure size 12x12 inches for the tree map

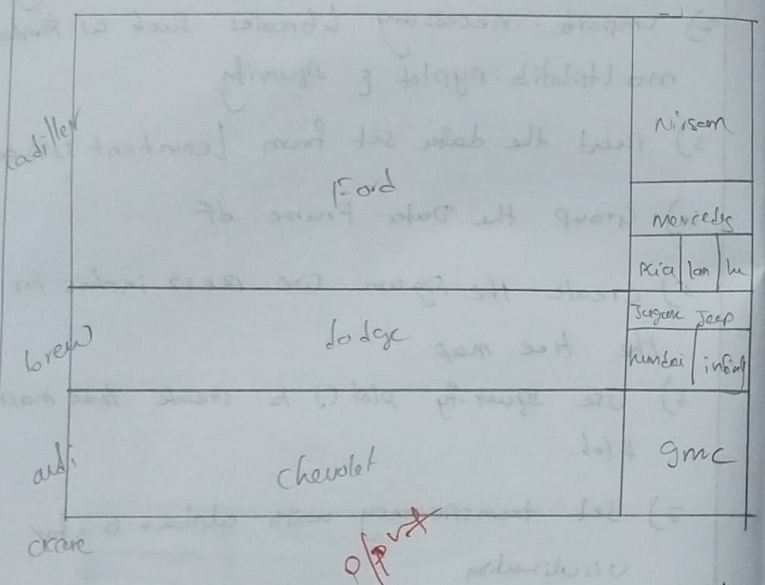
6) Use squarify.plot() to create tree map plot.

7) Set transparency with alpha = 0.6 for visualization.

8) Turn off the axis

9) Display plot using plt.show().

output



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.group by ('brand')
plt.figure(figsize = (13, 12))
plt.axis (off)
plt.title ('Tree map of total price by brand')
plt.show().
```

Task-5b

Aim: Build a sunburst display using above Program dataset.

Algorithm:

1.) Install the required package 'squarify' & plotly using pip

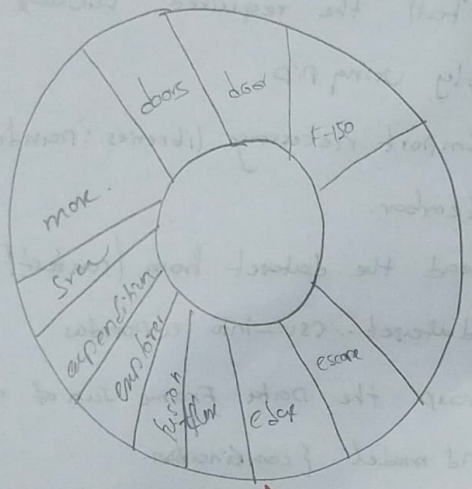
2) Import necessary libreries: pandas; numpy; seaborn.

3) Read the dataset from /content/USA_cars_dataset.csv into a pandas

4) Group the Data Frame sum of df brand and model combination.

5) Display the plot using fig.show();

output:



o/p

Program:

```
Pip install squarify
Pip install plotly
import pandas as Pd
import numpy as np
import matplotlip.pyplot as plt.
import seaborn as sns.
import squarify.
impor Plotly express as Px

df=Pd.read_csv('(content/USA-car
                -dataset=csv')

Print(df)

Sun burst-data: df.Group by (brand'model)

sun(), reset_index()

fig.show()
```

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

Result: To design & Perform visualization for trees is Successfuly done.

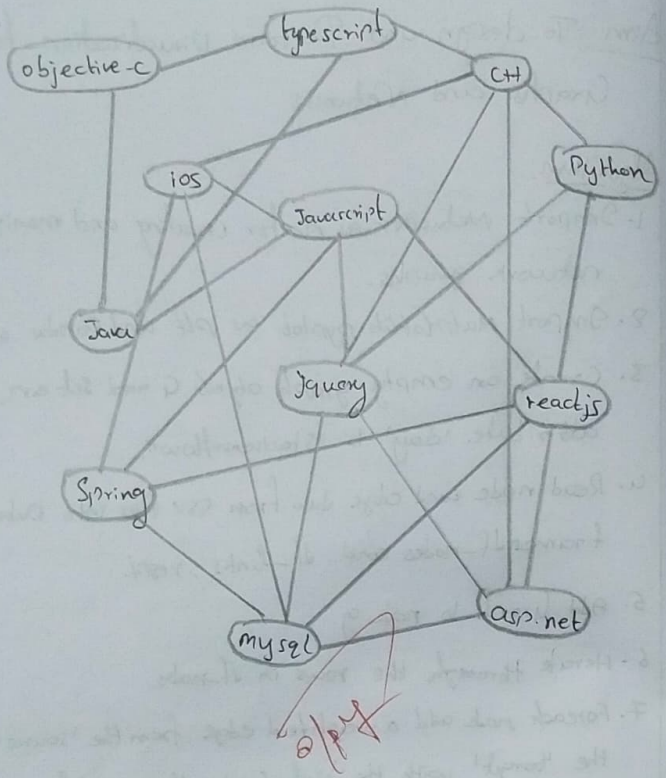
Task-6: 25/8/25

Aim: To design and Perform visualization for Graphs and Networks.

Algorithm:

1. Import Networkx as nx for creating and manipulating network graphs.
2. Import Matplotlib.pyplot as plt and Pandas as pd.
3. Create an empty graph object G and set an attribute 'day' to "Stackoverflow".
4. Read node and edge data from csv files into Data frames df_nodes and df_links, respt.
5. Add Graph to node g.
6. Iterate through the rows in df_nodes.
7. Foreach node add a weighted edge from the 'source' to the 'target' with the 'value' as the weight.
8. Iterate through the rows in df_links.
9. Define visualization options like edge color, width, node labels, and font weight.
10. Determine colors for nodes based on their 'group' attributes.
11. Set the edge colour to a specified color.
12. Show the graph using plt.show()
13. End of the algorithm.

output



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_weighted_edges_from([[row['source'], row['target'],
                            row['value']]])

plt.figure(figsize=(15,15))
option = {
    'edge_color': '#FFDEA2',
    'width': 1.5,
    'with_labels': True,
    'font_weight': 'regular',
}
colors = [color_map[G.nodes[node]['group']] for node
                                    in G]
sizes = [G.nodes[node]['nodesize']*25 for node in G]
nx.draw(G, node_color=colors, node_size=sizes,
        pos=nx.spring_layout(G, k=1.5, iterations=15),
                            **options)
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)	4
RECORD (5)	4
TOTAL (20)	24 → 18
SIGN WITH DATE	

Result: Thus, designing and Performing Visualization for Graphs and Networks done Successfully.

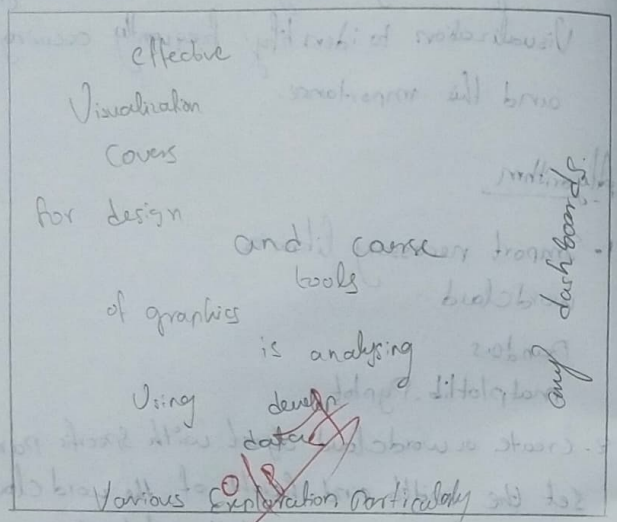
Task-7

8/9/25

Aim: To generate insight from text data using Text network Analysis and Visualation tool Such as Tagcroud and the word cloud Package in Python, by creating word clouds and graphical Visualization to identify frequently occuring terms and this importance.

Algorithm

1. Import necessary libraries.
   wordcloud
   Pandas
   matplotlib.Pyplot
2. Create a wordcloud object with specific Parameters:
   Set the width and height of the word cloud image
   Provide a list of stopword to exclude common words.
3. Display the word cloud image:
   Use Plt.imshow to display the word cloud. Turn off the axis labels. Adjust the layout for minimize Padding
4. End of Algorithm.



Program

```
from word cloud 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, practice, tools particularly
    for analysing and presenting businessse data!"
wordcloud = wordcloud (width = 800 height = 800,
                       background_color = 'white',
plt.figure (figsize = (5,5), facolor = none)
plt.imshow (word cloud)
plt.axis ("off")
plt.tight_layout (pad=0)
plt.show()
```

Result: To generate insight from text data using text network Analysis and Visualization tool such as Tagarad and the word cloud Package is done successfully.

Task-7b

Aim: To generate the word tree graph that shows how words in a text are connected providing insights into word relationship.

Algorithm:

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

Program:

```
import nltk
import matplotlib.pyplot as plt
import network as nx
from nltk corpus import stopwords
from nltk to tenice import word_to tenice
    nltk download ("punkt")
    nltk download ("stepwords")
text : Data analysis is import Data visualization. Clean
```

Output:

cloud word Analysis

Text Visualization

Tagcrowd

generate help insights

simplest.

tokens = word_tokenize(text.lower())

stop_words = set(stopwords.word('english'))

bigrams = list(nltk.bigrams(f.(literal.tokens)))

read_word = "data"

G = nx.DiGraph()

for w, w2 in bigrams

plt.figure(figsize=(10

nx.draw_network x(G, with_labels = True, node_

colour = skyblue

plt.title("word Tree Graph for root word:

(root_word)", font size=14)

plt.show()

VEL TECH	
EX No.	4
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	25
SIGN WITH DATE	

8/9

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

Task-8

15/9/25

Aim: To analyze and visualize Tamil Nadu Cities Population data across census years using Spatial and geospatial methods, revealing trends, growth patterns, and clusters with Python, QGIS, or Tableau.

Algorithm:

1. Data Preparation: Download Population statistics and Spatial boundary data for TN.
2. Spatial Join: Link city/town population data to geographic features align with tabular data.
3. Mapping: Map population values using choropleth maps and plot city/town Points/areas.
4. Spatial Analysis: Calculate growth rates for cities from census years.
5. Visualization: Generate maps showing population distribution and growth clusters.

output

Name	Status	District	POP_1991	2001	Growth	latitude
Chennai	Municipal Corp	Chennai	3,841,396	4,343,645	839,691	13.082
Coimbatore	MC	coimbatore	816,321	930,882	234,400	11.016
Madurai	MC	Madurai	477,256	928,869	39,009	9.9252
Salem	MC	Salem	490,280	696,760	332,987	11.664
Tiruchirappalli	MC	Tiruchirapalli	711,104	752,066	205,753	10.790

Program :

```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
pop_df = pd.read_csv('cities_in_tamil_nadu_population.csv')
cities_gdf = gpd.read_file('tamilnad_cities_shapfile.shp')
merged_gdf = cities_gdf.merge(pop_df, left_on='Name',
                              right_on='Name')
merged_gdf['growth'] = merged_gdf['2011-03-01'] - merged
                                  -gdf['1991-03-01']
merged_gdf.plot(column='2011-03-01', cmap='OrRd',
                                  legend=True)
plt.title('Population of Tamil Nadu Cities (2011 census)')
plt.show()
merged_gdf.plot(column='growth', cmap='YlGnBu',
                                  legend=True)
plt.title('Population Growth of Tamil Nadu Cities
                                  (1991-2011)')
plt.show()
```

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	
TOTAL (20)	17
SIGN WITH DATE	

15/9

Result: Thus, analyzing and visualizing TN cities Population data across census years using spatial and geospatial methods, revealing trends, growth Patterns, clusters done Successfully.

29/9/25

Task:9:- To analyze and Visualization Time oriented data.

Aim: To analyze and Visualize Time oriented Data using line graph, trend lines, area chart

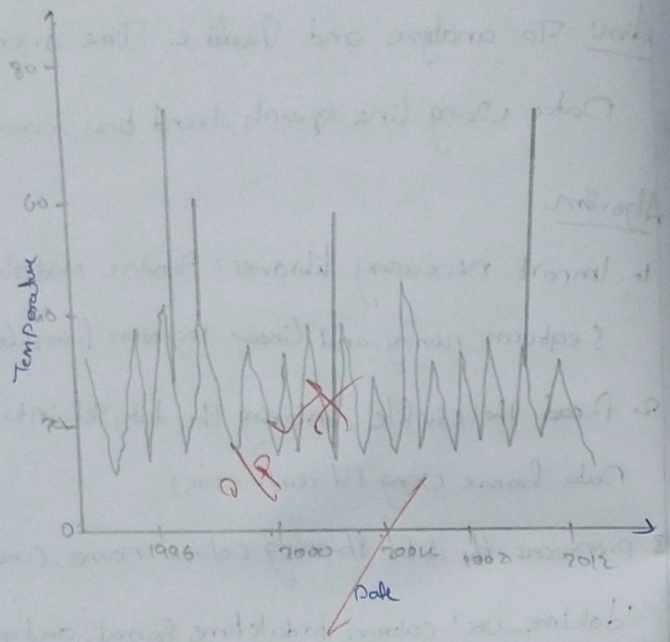
Algorithm:

1. Import necessary libraries: Pandas, matplotlib, Seaborn, numpy, and Linear regression from sklearn.
2. Read the csv file containing the dataset into a Data frame using Pd.read_csv().
3. Preprocess the data Stripping column names, Converting 'datetime_utc' column to datetime format, and setting it as the index.
4. Plot the temprature data overtime using Sns.lineplot()
5. Customize Plot by adding a title, labes for the x{y axis enabling grid lines, rolling x-axis labes plot using plt.show().

Program:

```
import pandas as pd
import matplotlib as mpl pyplot as plt.
import seaborn as sns
import numpy as np
```

output




```
from sklearn.linear_model import LinearRegression
data = pd.read_csv('/content/testset.csv')
data['datetime_utc'] = pd.to_datetime(data['datetime_utc'])

plt.figure(figsize=(10,6))
sns.lineplot(data=data, x=data.index, y='_tempm')

plt.title('Line Graph of Temperature over Time', 'Data', 'Temperature')
plt.fig_layout()
plt.show()
```

Trend Liner

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
data = pd.read_csv('/content/testset.csv')
data = ('_tempm').fillna(data['_tempm'].mean(), inplace=True)
x = data.index.astype(int).values.reshape(-1,1)
y = data['_tempm'].values
model = LinearRegression()
model.fit(x,y)
slope = model.coef_[0]
intercept = model.intercept_
plt.scatter(data.index.data['_tempm'], color='blue',
```

label='Temprature Data')

plt.title('Temp Trend Line')
plt.ylable('Temp(°C)')
plt.grid(True)

Area chart

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('/content/test set.csv')
data['date time_utc', inplace=True)
plt.figure(figsize=(10,6))
plt.xlabels('Date')
plt.ylabel('Temp')
plt.fig_root()
plt.show().
```

VEL TECH	
EX No.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
RECORD (5)	5
TOTAL (20)	18
SIGN WITH DATE	

Result: Thus, we successfully implemented the linegram trendline, Area chart by Time oriented Data and out is verified.

13/10/25

USE CASE – 1

Earthquake and Geospatial Data Analysis

Batch Mates:

1. VTU24801 JINKALA SHASHIDHAR
2. VTU24802 SHAIK JAKEER HUSSAIN
3. VTU24858 SHAIK MOHAMMED SHAHID
4. VTU24867 KUSUMARAJU BUKARI BABA
5. VTU24905 SHAIK RUBIYA
6. VTU24923 MALLU RUDRA TEJESHWAR REDDY
7. VTU25011 RAAVI NAVADEEP CHOWDARY
8. VTU25104 DONGALA JAYANDAR
9. VTU25106 MADAKKA GARI KRISHNA KOWSHIK
10. VTU25112 SEELAM NAVEEN

Aim:

To analyze and visualize earthquake occurrences using geospatial data.
The goal is to identify regions with the highest seismic activity, visualize earthquake magnitudes on maps,
and observe temporal patterns to support disaster risk assessment and preparedness.

Algorithm:

1. Load the dataset (**earthquake_data.csv**) containing Date, Latitude, Longitude, Magnitude, Depth, Region.
2. Convert **Date** into datetime format and handle missing or invalid entries.
3. Filter earthquakes based on magnitude thresholds (e.g., ≥4.0 for moderate and above).
4. Compute:
    - Average magnitude per region
    - Total earthquakes per region
    - Maximum depth and strongest quake recorded
5. Visualize data using:
    - Geographic Scatter Map (Latitude vs Longitude) color-coded by magnitude
    - Bar chart showing top earthquake-prone regions
    - Line plot showing monthly earthquake counts

OUTPUT:

=== Yearly Average Performance (Descending) ===

Sales	70.30
Finance	57.68
Marketing	54.05
R&D	49.08
HR	46.83

=== Best and Worst Month per Department ===

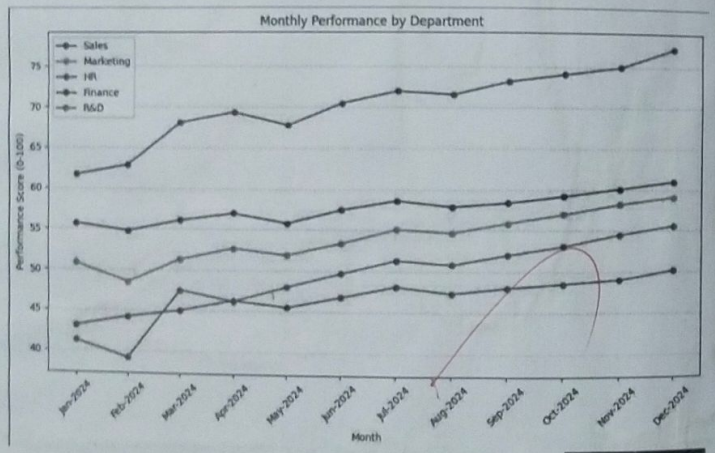
Sales: Best = Dec-2024, Worst = Jan-2024

Marketing: Best = Dec-2024, Worst = Feb-2024

HR: Best = Dec-2024, Worst = Jan-2024

Finance: Best = Dec-2024, Worst = Feb-2024

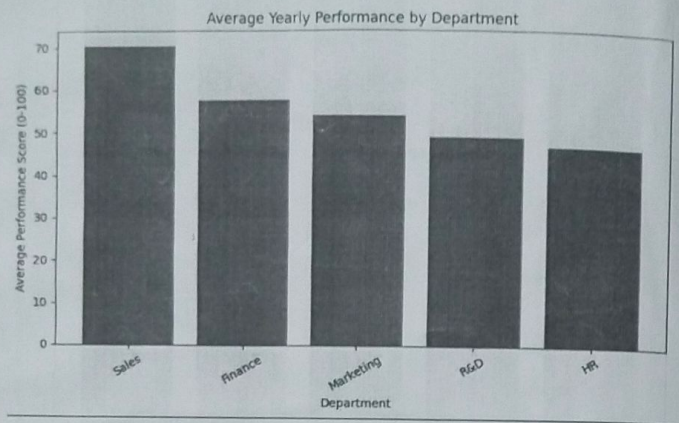
R&D: Best = Dec-2024, Worst = Feb-2024



6. Generate summary metrics (average magnitude, deepest quake, most active region).
7. Export visualizations and summary results as CSV/PNG files if needed.

Program:

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px

df = pd.read_csv("earthquake_data.csv")

df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df['Magnitude'] = pd.to_numeric(df['Magnitude'], errors='coerce')
df = df.dropna(subset=['Latitude', 'Longitude', 'Magnitude'])

avg_mag = df['Magnitude'].mean().round(2)
max_mag = df['Magnitude'].max()
most_active_region = df['Region'].value_counts().idxmax()

print("\n=== Earthquake Data Summary ===")
print(f"Average Magnitude     : {avg_mag}")
print(f"Strongest Earthquake  : {max_mag}")
print(f"Most Active Region    : {most_active_region}")

fig = px.scatter_geo(df, lat='Latitude', lon='Longitude', color='Magnitude',
                     title='Global Earthquake Distribution by Magnitude',
                     color_continuous_scale='Reds', size='Magnitude', projection='natural earth')
fig.show()
```




```
region_counts = df['Region'].value_counts().head(10)
plt.figure(figsize=(10,6))
plt.bar(region_counts.index, region_counts.values, color='orange')


plt.title("Top 10 Most Earthquake-Prone Regions")
plt.xlabel("Region")
plt.ylabel("Number of Earthquakes")
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()

monthly = df.groupby(df['Date'].dt.to_period('M')).size()
monthly.plot(kind='line', figsize=(10,5), color='green')
plt.title("Monthly Earthquake Occurrence Trend")
plt.xlabel("Month")
plt.ylabel("Number of Earthquakes")
plt.grid(True)
plt.tight_layout()
plt.show()
```

VELTECH	
EX No.	UC1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	5
TOTAL (25)	17
SIGN WITH DATE	

13/10

Result:

Thus, earthquake occurrences were successfully analyzed and visualized using geospatial data. The analysis revealed regions with the highest seismic activity, identified the strongest earthquakes, and showcased temporal patterns in frequency, enabling better understanding of global earthquake behavior and aiding disaster management planning.

27/10/25

USE CASE – 2

Performance of Different Company Departments Over Year

Batch Mates :

1. VTU24801 JINKALA SHASHIDHAR
2. VTU24802 SHAIK JAKEER HUSSAIN
3. VTU24858 SHAIK MOHAMMED SHAHID
4. VTU24867 KUSUMARAJU BUKARI BABA
5. VTU24905 SHAIK RUBIYA
6. VTU24923 MALLU RUDRA TEJESHWAR REDDY
7. VTU25011 RAAVI NAVADEEP CHOWDARY
8. VTU25104 DONGALA JAYANDAR
9. VTU25106 MADAKKA GARI KRISHNA KOWSHIK
10. VTU25112 SEELAM NAVEEN

Aim:

To analyze and visualize yearly department performance using time-series plots and summary metrics to identify trends, best/worst months, and overall strengths or weaknesses.

Algorithm: 1. Define departments and the time dimension (months for the year).

2. Acquire or generate monthly performance metrics for each department (normalized scores, KPIs, etc.).
3. Build a tidy DataFrame: rows = months, columns = departments.
4. Compute derived metrics: yearly average per department, month-over-month changes, best/worst months.
5. Visualize:

   Line chart: performance over months (one line per department). Bar chart: average yearly performance per department.
6. Export charts and data for reporting (PNG/CSV/PDF).

Program:

```
import numpy as np import
pandas as pd import
matplotlib.pyplot as plt
```

Output:

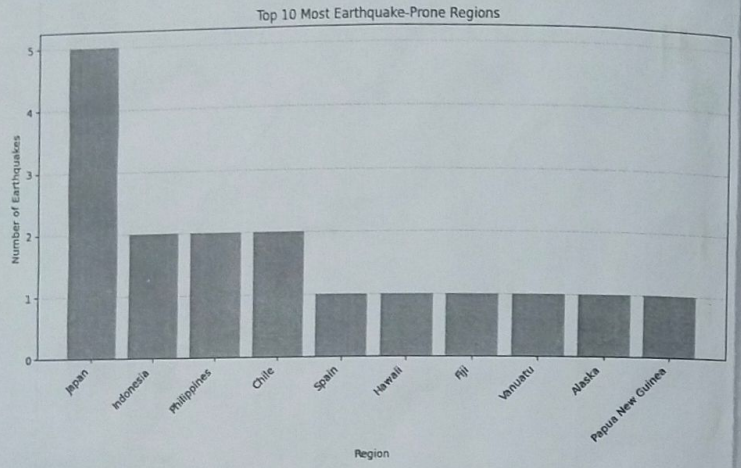
=== Earthquake Data Summary ===

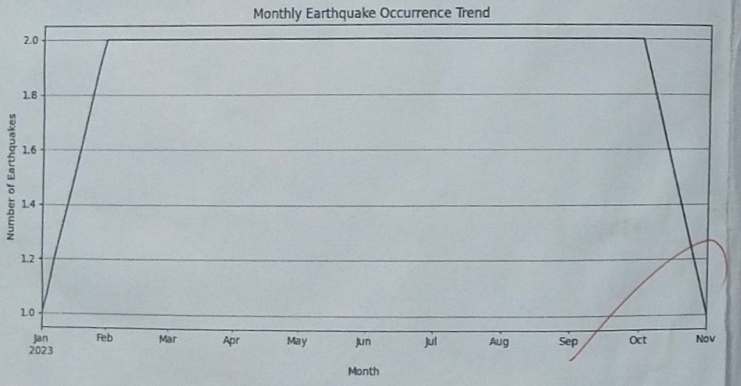
Average Magnitude : 5.98

Strongest Earthquake : 8.3

Most Active Region : Japan

4.555.566.577.58MagnitudeGlobal Earthquake Distribution by Magnitude






```
df=pd.read_csv("/content/drive/MyDrive/ColabNotebooks/department_performance_2024.csv")
df.set_index('Month', inplace=True) yearly_avg =
df.mean().round(2).sort_values(ascending=False)

best_month = df.idxmax() worst_month
= df.idxmin() monthly_changes =
df.diff().round(2)

print("\n=== Yearly Average Performance (Descending) ===")
print(yearly_avg) print("\n=== Best and Worst Month per Department ===")
for dept in df.columns: print(f"{dept}: Best = {best_month[dept]}, Worst =
{worst_month[dept]}")

plt.figure(figsize=(10,6)) for
dept in df.columns:
  plt.plot(df.index, df[dept], marker='o', linewidth=2, label=dept)
plt.title("Monthly Performance by Department")
plt.xlabel("Month") plt.ylabel("Performance Score (0-100)")
plt.xticks(rotation=45) plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.legend() plt.tight_layout() plt.show()

plt.figure(figsize=(8,5))
plt.bar(yearly_avg.index, yearly_avg.values)
plt.title("Average Yearly Performance by Department")
plt.xlabel("Department") plt.ylabel("Average
Performance Score (0-100)") plt.xticks(rotation=30)
plt.tight_layout() plt.show()
```

VEL TECH	
EX No.	12
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RD (5)	
TAL	18
DATE	

27/10

Result:

Thus, the yearly department performance was successfully analyzed and visualized, revealing key trends, best and worst months, and insights for data-driven decisions.