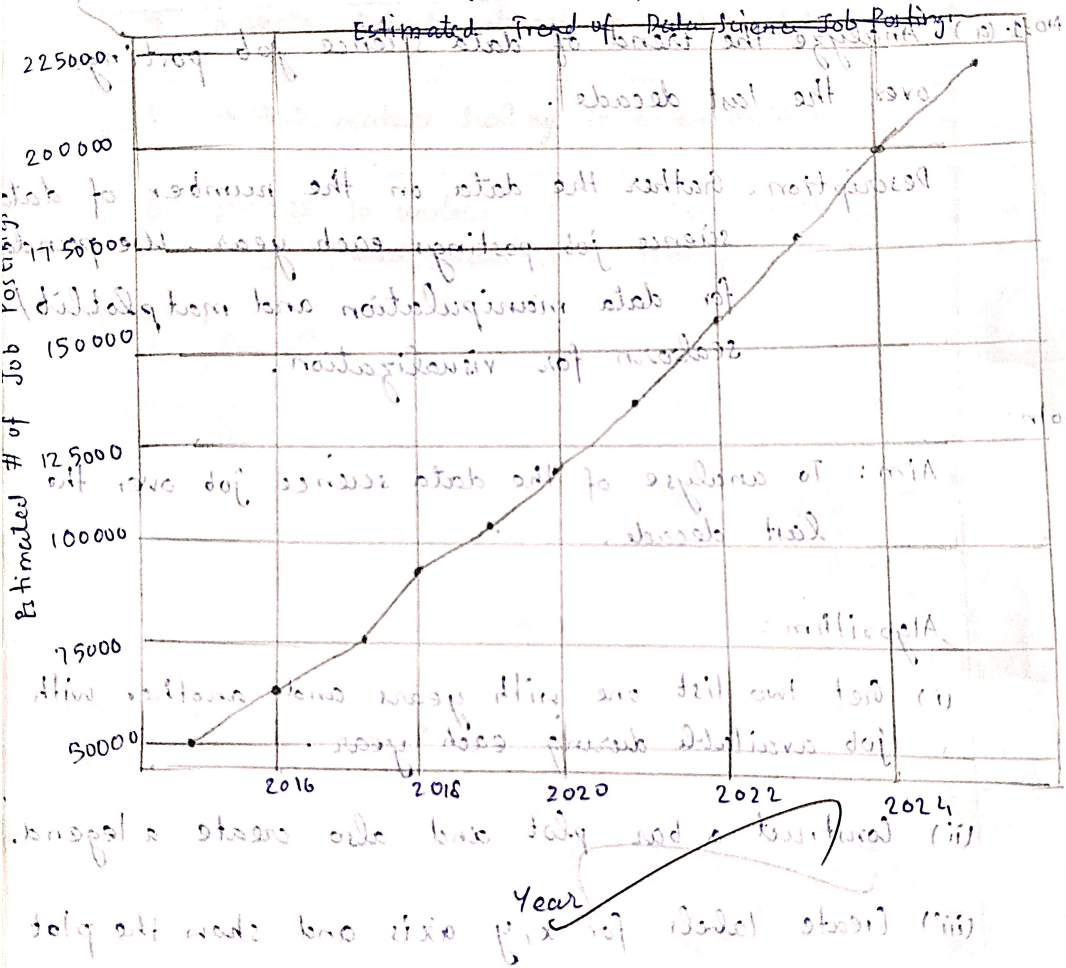
Estimated Trend of Data Science Job Postings
(2015 - 2025)



11-08-25

Experiment 1

Exp No:1. (a) Analyze the trend of data science job postings over the last decade.

Description: Gather the data on the number of data science job postings each year. Use pandas for data manipulation and matplotlib/ seaborn for visualization.

Soln:

Aim: To analyse of the data science job over the last decade.

Algorithm:

(i) Get two list one with years and another with job available during each year.

(ii) Construct a bar plot and also create a legend.

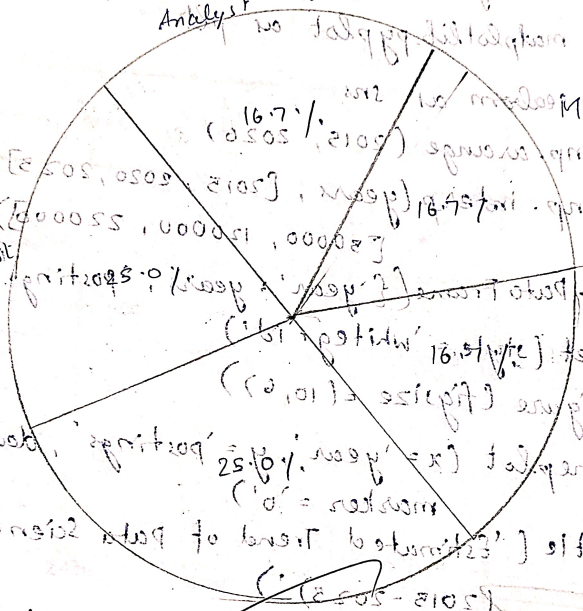
(iii) Create labels for x, y axis and show the plot.

Program:

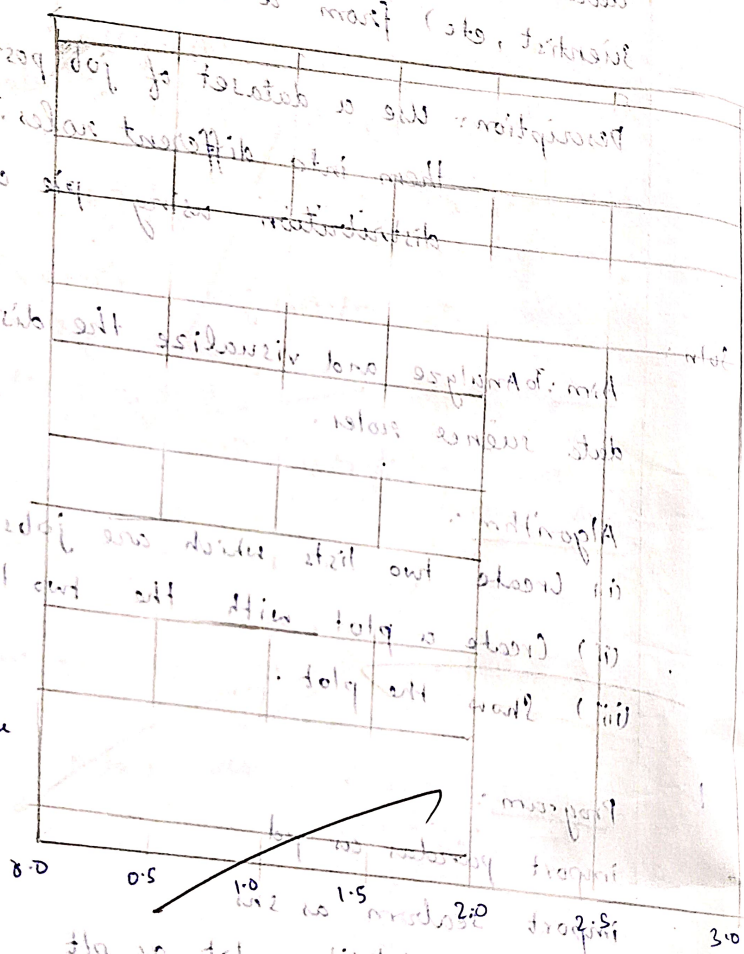
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
years = np.arange (2015, 2026)
posts = np. interp (years, [2015, 2020, 2025],
                    [30000, 120000, 220000])
df = pd. DataFrame({'year': years, 'postings': posts})
sns. set (style = 'whitegrid')
plt. figure (figsize = (10, 6))
sns. lineplot (x = 'year', y = 'postings', data = df,
               marker = 'o')
plt. title ('Estimated Trend of Data Science Job Postings
            (2015 - 2025)')
plt. ylabel ('Estimated # of Job Postings').
plt. xlabel ('Year')
plt. show ()
```

Result: Thus, the experiment of estimation of trends of data science job postings in years is successfully plotted and verified.

Distribution of Data science Roles in Job Posting



Distribution of Data Science Roles in Job Postings



21-07-2026

Exp No: 1.(b) Analyze and visualize the distribution of various data science roles (Data Analyst, Data Engineer, Data Scientist, etc) from a dataset.

Description: Use a dataset of job postings and categorize them into different roles. Visualize the distribution using pie charts of bar plots.

Soln:

Aim: To Analyze and visualize the distribution of various data science roles.

Algorithm:

(i) Create two lists which are jobs and no. of jobs.

(ii) Create a plot with the two lists

(iii) Show the plot.

Program:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.DataFrame({'year': years, 'postings': posts})
sns.set_style('darkgrid')
plt.figure(figsize=(12,6))
sns.lineplot(data=df, x='year', y='postings', lw=2,
             marker='o')
plt.xlabel('Year')
plt.ylabel('No. of Job Postings (approx.)')
plt.xticks(years)
plt.show()
```

Program

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = {
        'job_title': ['Data Scientist', 'Data Analyst', 'Data Engineer',
        'Machine Learning Engineer', 'Business Intelligence
        Analyst', ~~'Data Scientist'~~,
        ]
}

df = pd.DataFrame(data)
def categorize_role(title):
    title = title.lower()
    if 'machine learning' in title or 'ml engineer'
    in title:
        return 'Machine Learning Engineer'
    elif 'scientist' in title:
        return 'Data Scientist'
    elif 'engineer' in title:
        return 'Data Engineer'
    elif 'analyst' in title:
        if 'business intelligence' in title
            return 'Business Intelligence Analyst'
        return 'Data Analyst'
    else:
        return 'other'
```

```
df['role'] = df['job_title'].apply(categorize_role)
role_counts = df['role'].value_counts().reset_index()
role_counts.columns = ['Role', 'Count']
sns.set(style='white grid')
plt.figure(figsize=(8,6))
sns.barplot(data=role_counts, x='Count', y='Role', palette='viridis')
plt.title('Distribution of Data Sience Roles in Job Postings')
plt.xlabel('Role')
plt.show()
plt.figure(figsize=(7,7))
plt.pie(role_counts['Count'], labels=role_counts['Role'], autopct='%1.1f%%', startangle=140, colors=sns.color_palette('viridis', len(role_counts))
plt.title('Distribution of Data Science Roles in Job Postings')
plt.axis('equal')
plt.show()
```

Result: Thus, the experiment for distribution of data sience roles in job postings are plotted and visualized and verified successfully.

8-07-2027

Exp No:1.(c) Conduct an experiment to differentiate Structured, Un-structured and Semi structured data.

Description: Create small datasets for each type and explain their characteristics.

Soln:

Aim:

To conduct an experiment to differentiate structure~~structure~~ and unstructured data.

Algorithm:

(i) Read the csv file and extract
(ii) Create two different bar plot.
(iii) Segregate the necessary data required to draw the plot
(iv) Show the plot.

Program: (structured Data)

```
import pandas as pd
structured_data = pd.DataFrame({
    'Employee ID': [101, 102, 103],
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'Department': ['HR', 'Engineering', 'Finance']
})
print("Structured Data (Tabular):")
print(structured_data)
```

Output : (Structured Data) :

Structured Data (Tabular) :

	Employee ID	Name	Age	Department
0	101	Alice	25	HR
1	102	Bob	30	Engineering
2	103	Charlie	25	Finance

Program : (Semi-structured Data)

```
import json
semi_structured_data = [
    {
        "Employee ID": 101,
        "Name": "Alice",
        "Age": 25,
        "Department": "HR",
        "Skills": ["Communication", "Recruitment"]
    },
    {
        "Employee ID": 102,
        "Name": "Bob",
        "Age": 30,
        "Department": "Engineering",
        "Skills": ["Python", "Databases"]
    }
]
print("Semi-structured Data (JSON):")
print(json.dumps(semi_structured_data, indent
```

Output: (Semi-structured data)

Semi-structured Data (JSON):


```
[
    {
        "Employee ID": 101,
        "Name": "Alice",
        "Age": 25,
        "Department": "HR",
        "Skills": [
            "Communication",
            "Recruitment"
        ]
    },
    {
        "Employee ID": 102,
        "Name": "Bob",
        "Age": 30,
        "Department": "Engineering",
        "Skills": [
            "Python",
            "Databases"
        ]
    }
]
```

Program : (Unstructured data)

```
unstructured data = """
Alice is a great team player with excellent
communication skills. Bob has strong expertise
in Python and database management.
"""
print ("Unstructured Data (Text) :")
print (unstructured_data)
```

Output : (Unstructured data)

```
Unstructured Data (Text) :

Alice is a great team player with excellent
communication skills. Bob has strong expertise
in Python and database management.
```

Result : Thus, the python program to analyze and visualize data of structured, unstructured and semi-structured data has been executed and verified successfully.

28-07-2025

Expt No 1. (d) : Conduct an experiment to encrypt and decrypt given sensitive data.

Description : Use the cryptography library to encrypt and decrypt a piece of data.

Soln: Aim: To encrypt and decrypt a given piece of data using the cryptography library in Python.

Algorithm:

1) To import the Fernet class from the cryptography fernet module.
2) Generate a secret key using Fernet.generate_key().
3) Create a Fernet object with the generated key.
4) Take a sensitive data as input and convert it into bytes.
5) Encrypt the data using the encrypt() function.
6) Decrypt the encrypted data using the decrypt() function.
7) Display the original data, encrypted text, and decrypted text.

Program:

```
from cryptography.fernet import Fernet
key = Fernet.generate_key()
cipher = Fernet(key)
print("Generated Key", key.decode())
data = "My Password@123"
print("\n Original Data:", data)
encrypted_data = cipher.encrypt(data.encode())
print("\n Encrypted Data:", encrypted_data.decode
decrypted_data = cipher.decrypt(encrypted_data).decod
print("\n Decrypted Data:", decrypted_data)
```

Output :

Generated key :

A2-F2-5vS-2V1-pWYmv4VwZ-YcFQFYc1y3qSkrwSkLy

H&M =

Original Data : My Password@123

Encrypted Data :

gAAAAABjPjDKBGuqFqfMZZy-3qAKpSQnEmFvSgv

1hznjyzknsRg4bbZDC2mZkOnpQzxc-1wjlsFeoiz

shCMbt7vvBMQ8NN2q=

Decrypted Data : My Password@123.

Result : Thus, the given sensitive data was success fully encrypted and decrypted using the cryptography library.