EDS PROJECT SALARY OF EMPLOYEES

Guided by Prajakta Ugale

Presented By :-
Netra Patil : 202201040172
Sneha Kurhade : 202201070137
Sanskruti Shetiya : 202201070106

INTRODUCTION

- Data science is a field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from data. It is a multidisciplinary field that combines elements of statistics, computer science, mathematics, and business.

- NumPy is a Python library that provides a high-performance multidimensional array object. It is the fundamental package for scientific computing in Python.

- Pandas is a Python library that provides high-performance, easy-to-use data structures and data analysis tools.

- Linear regression is a statistical method that is used to model the relationship between two or more variables.

MOTIVATION

- A salary dataset is a collection of data that includes information about salaries, such as the job title, department,name of the employees, and the salary itself.

- Salary datasets are interesting to use in projects because they can provide insights into a variety of topics, such as the job market, the salaries etc.They can also be used to identify trends in salaries over time.

- Salary datasets can be used to create an analysis of the factors that affect salaries etc.

DETAILS OF DATASET

Name : Salary Dataset

Number of Features : 6

          Employee Id

          First Name

          Last Name

          Job Title

          Department
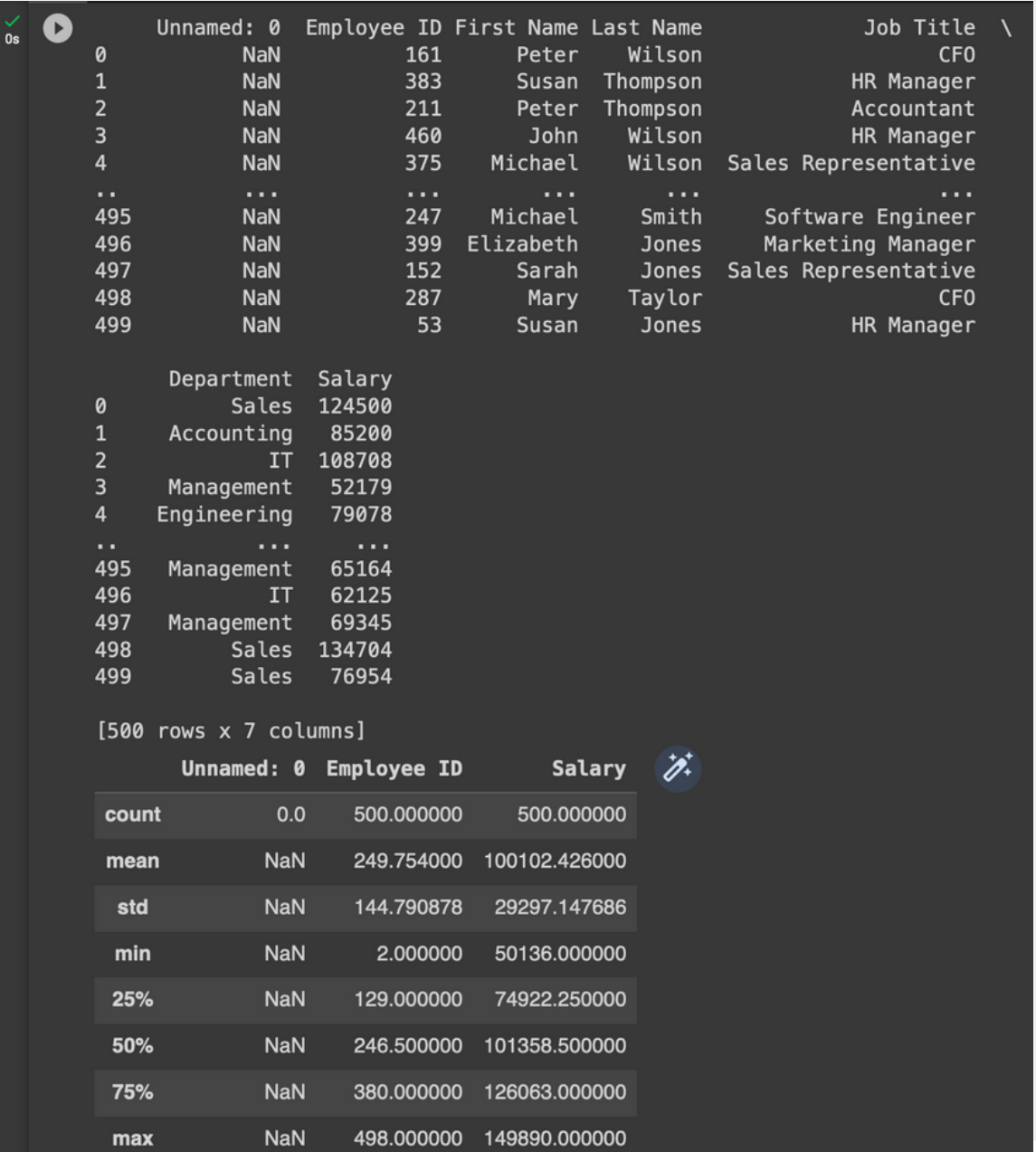
          Salary

Number of Records : Rows - 500

          Columns - 6

DATA MANIPULATION

```python
import pandas as pd
df =pd.read_csv("/content/employee-records.xlsx - Sheet1.csv")
#print all records of dataset
print(df)
# save DataFrame to a CSV file
df1.to_csv("Salary.csv",index=True)
# print all record through salary_data
salary_data=pd.read_csv('/content/employee-records.xlsx - Sheet1.csv')

# compute basic summary statistics of salary_data
salary_data.describe()
```

# #selecting salary>10000
print(df.loc[df['Salary']>10000])

# #apply multiple aggregation functions
df.groupby('Salary').agg(['mean','max','min'])

```
[500 rows x 7 columns]
     Unnamed: 0  Employee ID First Name Last Name           Job Title  \
0           NaN          161      Peter    Wilson                 CFO
1           NaN          383      Susan  Thompson          HR Manager
2           NaN          211      Peter  Thompson          Accountant
3           NaN          460       John    Wilson          HR Manager
4           NaN          375    Michael    Wilson Sales Representative
..          ...          ...        ...       ...                 ...
495         NaN          247    Michael     Smith   Software Engineer
496         NaN          399  Elizabeth     Jones   Marketing Manager
497         NaN          152      Sarah     Jones Sales Representative
498         NaN          287       Mary    Taylor                 CFO
499         NaN           53      Susan     Jones          HR Manager

       Department  Salary
0           Sales  124500
1      Accounting   85200
2              IT  108708
3      Management   52179
4     Engineering   79078
..            ...     ...
495    Management   65164
496            IT   62125
497    Management   69345
498         Sales  134704
499         Sales   76954

[500 rows x 7 columns]
```

	Unnamed: 0			Employee ID		
	mean	max	min	mean	max	min
Salary						
50136	NaN	NaN	NaN	431.0	431	431
50387	NaN	NaN	NaN	7.0	7	7
50493	NaN	NaN	NaN	473.0	473	473
51118	NaN	NaN	NaN	328.0	328	328
51200	NaN	NaN	NaN	194.0	194	194
...	...	...	...	...	...	...
148943	NaN	NaN	NaN	117.0	117	117
149012	NaN	NaN	NaN	226.0	226	226
149272	NaN	NaN	NaN	299.0	299	299
149676	NaN	NaN	NaN	271.0	271	271
149890	NaN	NaN	NaN	196.0	196	196

499 rows × 6 columns

#selecting salary>10000

```
print(df.groupby("Salary").get_group(141000)).
```

#compute the correlation between columns

```
print(df.corr())
```

```
     Unnamed: 0  Employee ID First Name Last Name              Job Title
278         NaN           93       Mary    Wilson      Marketing Manager

     Department  Salary
278  Accounting  141000
```

```
             Unnamed: 0  Employee ID    Salary
Unnamed: 0          NaN          NaN       NaN
Employee ID         NaN     1.000000 -0.037357
Salary              NaN    -0.037357  1.000000
```

DATA VISUALIZATION

```python
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_csv("/content/employee-records.xlsx - Sheet1.csv")
df.head()

#sample data
x=[5,3,15,7,19] #x-axis values(Employee Id)
y=[90000,65000,150000,60000,200000] #y-axis values(Salary)

#creation of line chart
plt.plot(x,y,marker='o')

#customization of the chart
plt.title("Line Chart Example")
plt.xlabel("Employee Id")
plt.ylabel("Salary")

#display the chart
plt.show()
```
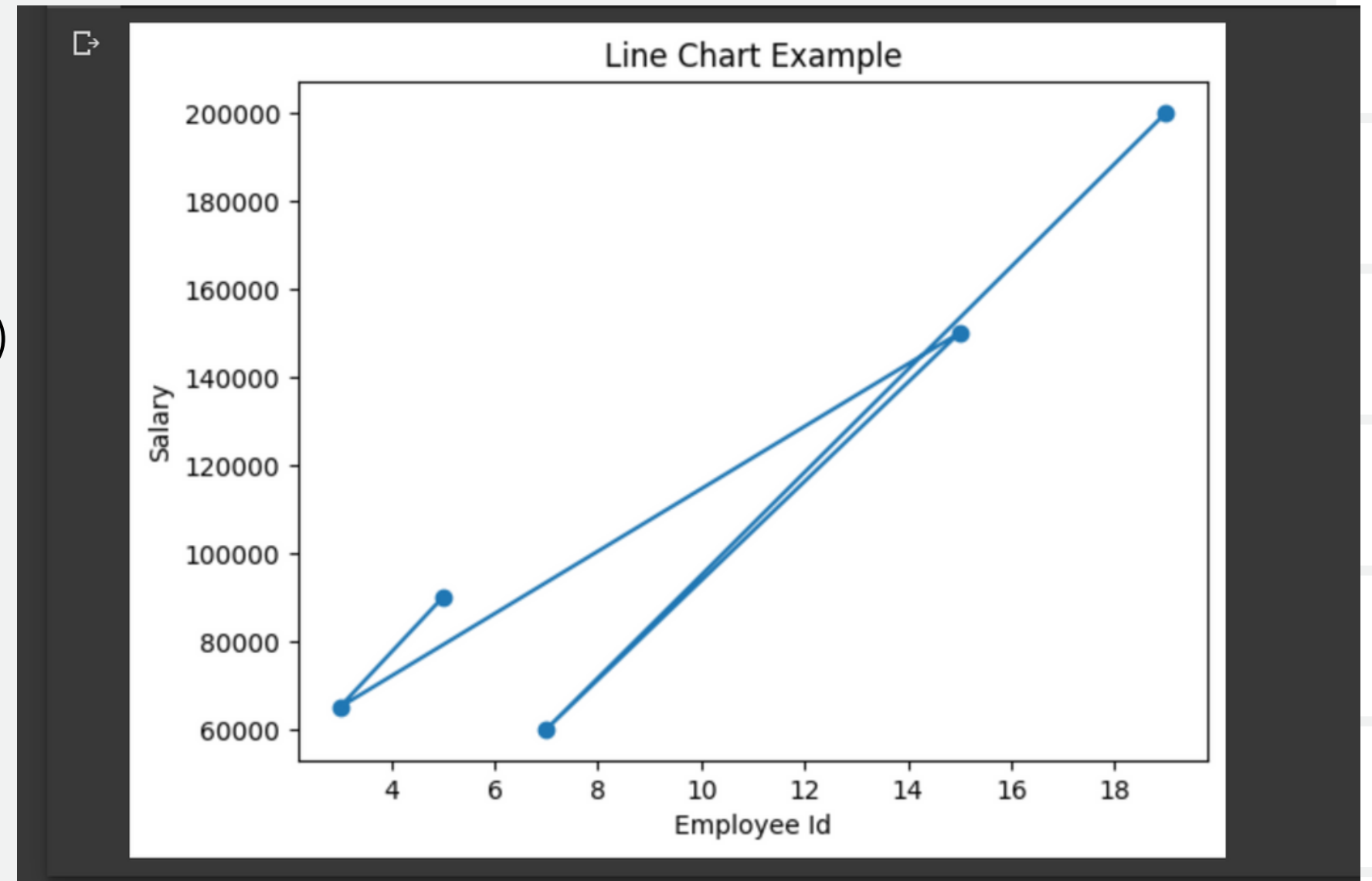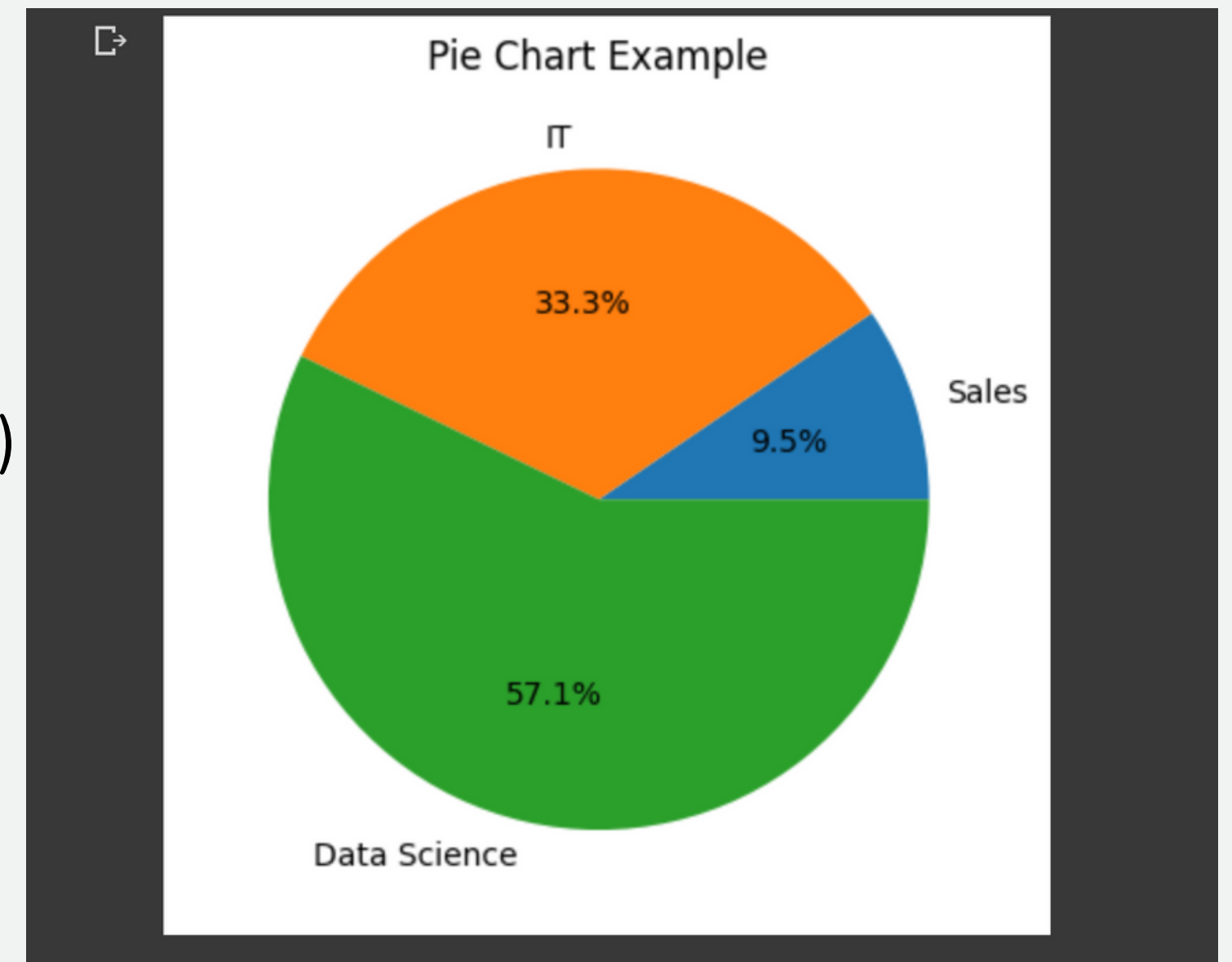
```python
import matplotlib.pyplot as plt

# Sample data
Department = ['Sales', 'IT','Data Science']
Salary = [10000, 35000, 60000]

# Plotting of Pie Chart
plt.pie(Salary, labels=Department, autopct='%1.1f%%')

# Addition of title
plt.title('Pie Chart Example')

# Display the chart
plt.show()
```

```python
import matplotlib.pyplot as plt

#Sample Data
Salary=['10000','300000','150000','50000']
JobTitle=['CEO','Software Engineer','Accountant','Data Scientist']

#Poltting of bar
plt.bar(Salary,JobTitle)
plt.xlabel("Salary")
plt.ylabel("JobTitle")
plt.title("Bar Plot Example")

#Display the plot
plt.show()
```
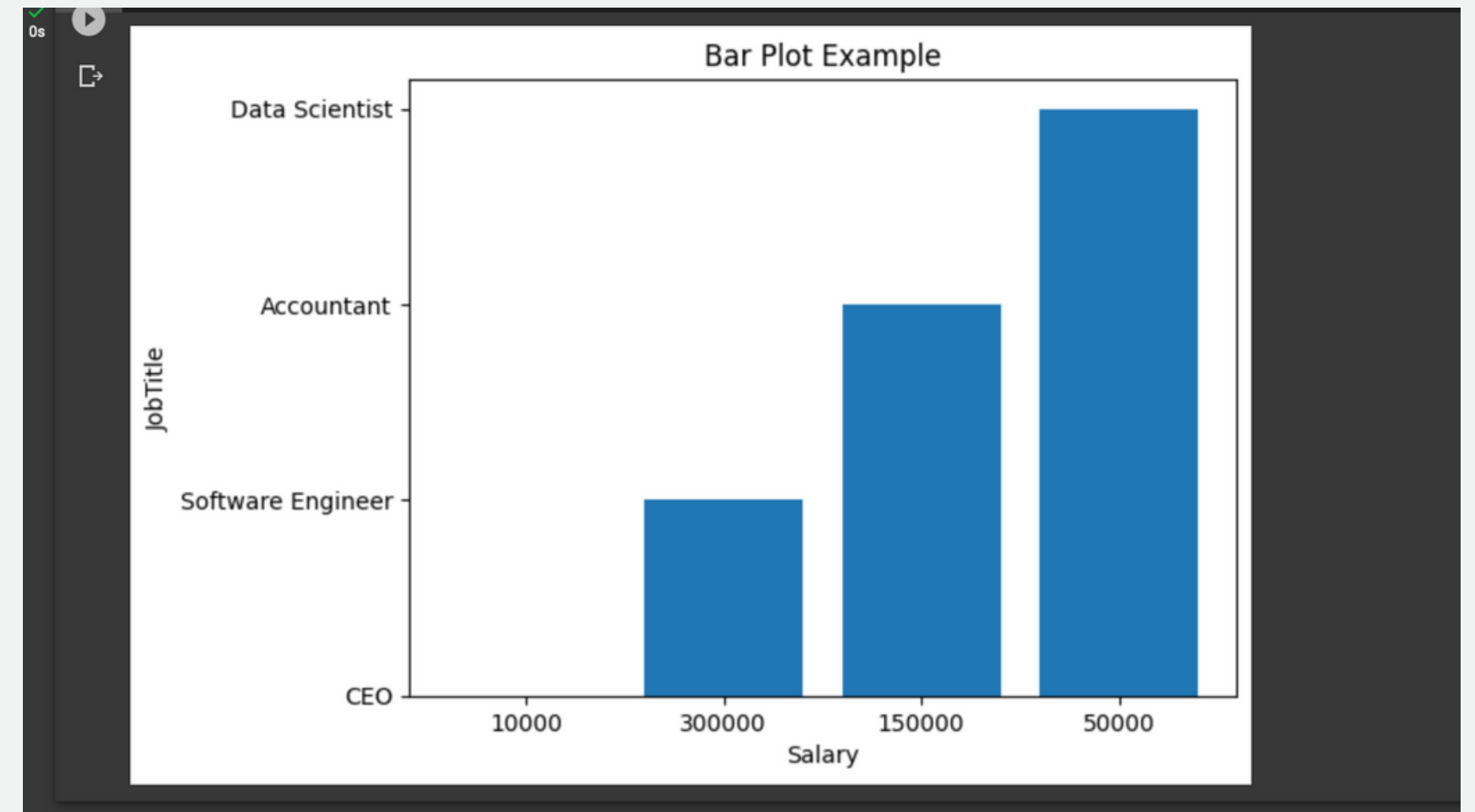
```python
import matplotlib.pyplot as plt

#Sample data
data=[45,28,45,36,52,40]

#Plotting of histogram
plt.hist(data,bins=5,edgecolor="black")

#Customization
plt.xlabel('Employee Id')
plt.ylabel('Salary')
plt.title('Histogram Example')

#Display the histogram
plt.show()
```
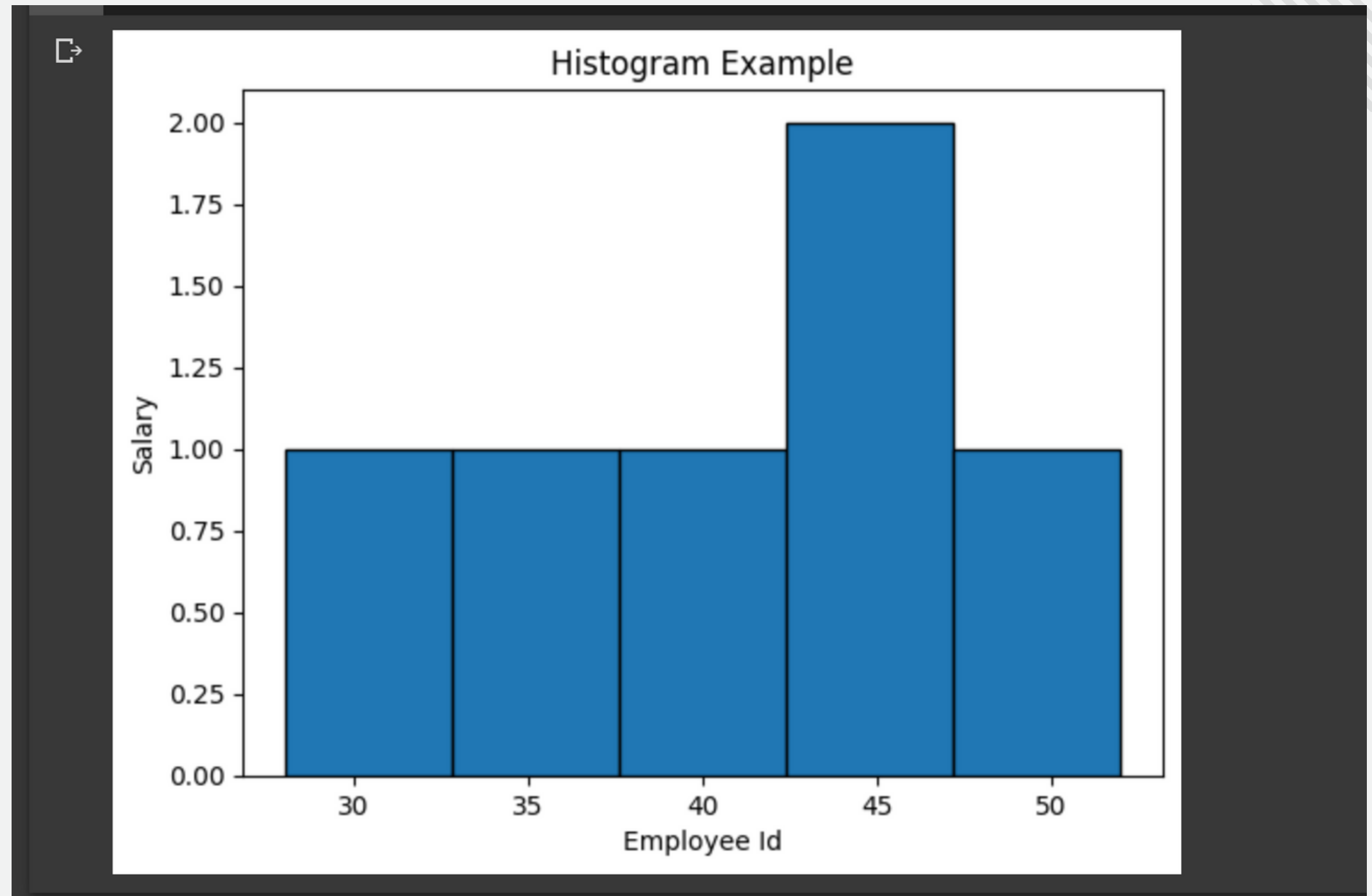
```python
import matplotlib.pyplot as plt
import numpy as np

#Generate some random data
data=np.random.randn(1000)

#Create density plot
plt.hist(data,density=True,bins=30,alpha=0.5)

#Addition of labels and title
plt.xlabel('Employee Id')
plt.ylabel('Salary')
plt.title('Density Plot Example')

#Display the plot
plt.show()
```
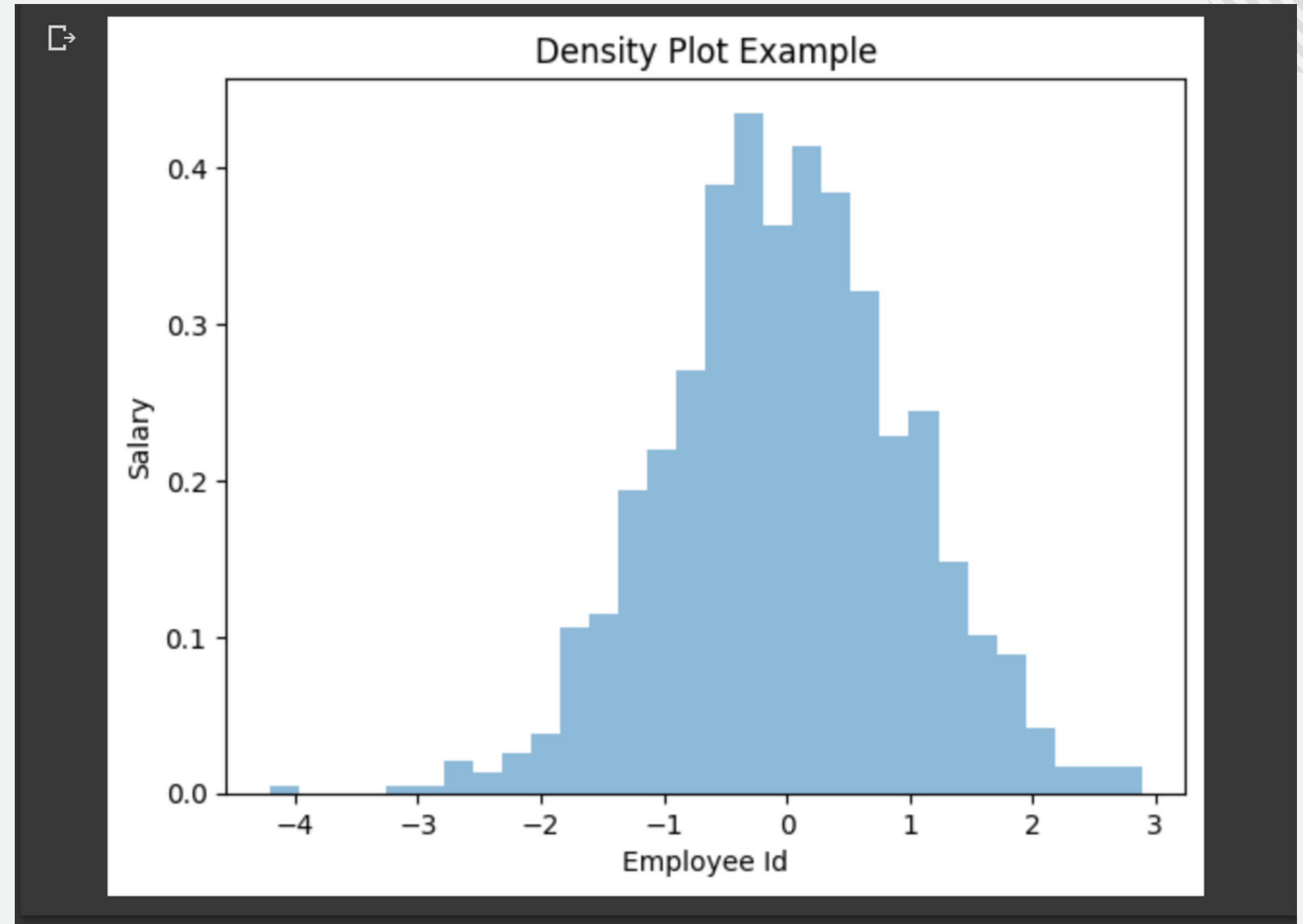
PREDICTIVE TECHNIQUE

```python
import pandas as pd
df=pd.read_csv('/content/employee-records.xlsx - Sheet1.csv')
print(df)
df1=df.groupby('Salary').max()
print(df1)
plt.plot(df1.index,df1['Department'],marker='o')

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
X=df['Salary']
df=df.dropna()
Y=df['Department']

X=np.array(df['Salary']).reshape(-1,1)
Y=np.array(df['Department']).reshape(-1,1)

#Dropping any rows with Nan Values
X_train,X_test,y_train, y_test = train_test_split(X,Y,test_size=0.25)

#Splitting data into training and testing data
regr=LinearRegression()
regr.fit(X_train,y_train)
print(regr.score(X_test,y_test))
```

`0.5600001 1243`

APPLICATION

- Data manipulation is a powerful tool that can be used to improve the quality, accuracy, and usability of data. It is a key part of many data science and machine learning workflows.
- Data visualization is the process of transforming data into a visual format that makes it easier to understand and interpret. It is a powerful tool that can be used to communicate insights from data to a wide range of audiences.
- Plots such as line,bar,histogram,pie chart can provide visual representations.
- After performing data manipulation, visualizing the data, and clustering using Kmeans,the resulting clusters can serve as new features for predictive modeling.
- The cluster labels can be used as input features to build a classification model to predict survival or any other relevant outcome.

CONCLUSION

- Our analysis of the Salary dataset has provided valuable insights into the Employee's information.

- We discovered significant correlations between Employee and salaries such as Employee Id, Job Title,Department and Salary.

- Through data cleaning, preprocessing, visualization, and modeling, we were able to extract meaningful information.

THANK YOU