GOVERNMENT POLYTECHNIC NAGAMANGALA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vth Semester Diploma

**Artificial Intelligence and Machine Learning (20CS51)**

**Assignment: 01**

**NAME: LOCHANA G**

**ROLL NO: 158CS22025**

AIML (20CS51)

ASSIGNMENT – WEEK 02

1. Download any two datasets from the internet and perform the following operations.

* Aggregate functions.
* Use Map, Filter, Reduce, and Lambda Functions with Pandas data frames
* Visualize the data set (At least 6 different plots).
* How do you create a project plan and product backlog for an AI project? (Everyone chooses the area you want to work on or do the research work. Give a brief introduction to the work carried out and the final report to be submitted at the end of the course.)
* Create a Git Repository for following the Regression Project ML / deep learning.
* Classification Project – ML / deep learning
* Clustering project – ML / deep learning
* Natural Language Processing – ML / deep learning

**Important Note:**

* **Last Date for Submission:17-07-2024.**
* **Everyone must perform the above operation using different datasets.**
* **Submit the report to the email** [**aimlgtn@gmail.com**](mailto:aimlgtn@gmail.com)

1. Download any two datasets from the internet and perform the following operations.

**DATASET:0.1**("/content/drive/MyDrive/drive/fifa-world-cup (1).csv")

**DESCRIPTION**

1. EDITION
2. YEAR
3. LOCATION
4. WINNER
5. TEAMS
6. MATCHES
7. GOALS
8. AVERAGE\_GOALS
9. AVERAGE\_ATTENDANCE

(**a) Aggregate functions.**

* **HEAD**

import pandas as pd

path="/content/drive/MyDrive/drive/fifa-world-cup (1).csv"

df=pd.read\_csv(path)

df.head(6)

**output**

**EDITION YEAR LOCATION WINNER TEAMS MATCHES GOALS AVERAGE\_GOALS AVERAGE\_ATTENDANCE**

**0 2014 FIFA World Cup Brazil ™ 2014 Brazil Germany 32 64 171 2.7 52918**

**1 2010 FIFA World Cup South Africa ™ 2010 South Africa Spain 32 64 145 2.3 49669**

**2 2006 FIFA World Cup Germany ™ 2006 Germany Italy 32 64 147 2.3 52491**

**3 2002 FIFA World Cup Korea/Japan ™ 2002 Korea/Japan Brazil 32 64 161 2.5 42268**

**4 1998 FIFA World Cup France ™ 1998 France France 32 64 171 2.7 43517**

**5 1994 FIFA World Cup USA ™ 1994 USA Brazil 24 52 141 2.7 68991**

* **TAIL**

import pandas as pd

path="/content/drive/MyDrive/drive/fifa-world-cup (1).csv"

df=pd.read\_csv(path)

df.tail(6)

**output**

EDITION YEAR LOCATION WINNER TEAMS MATCHES GOALS AVERAGE\_GOALS AVERAGE\_ATTENDANCE

14 1958 FIFA World Cup Sweden ™ 1958 Sweden Brazil 16 35 126 3.6 23423

15 1954 FIFA World Cup Switzerland ™ 1954 Switzerland Germany 16 26 140 5.4 29561

16 1950 FIFA World Cup Brazil ™ 1950 Brazil Uruguay 13 22 88 4.0 47511

17 1938 FIFA World Cup France ™ 1938 France Italy 15 18 84 4.7 20872

18 1934 FIFA World Cup Italy ™ 1934 Italy Italy 16 17 70 4.1 21352

19 1930 FIFA World Cup Uruguay ™ 1930 Uruguay Uruguay 13 18 70 3.9 32808

\

* **sum**

import pandas as pd

path="/content/drive/MyDrive/drive/fifa-world-cup (1).csv"

df=pd.read\_csv(path)

df['YEAR'].SUM()

**output**

39496

* **MINIMUM**

df.min()

**output**

EDITION 1930 FIFA World Cup Uruguay ™

YEAR 1930

LOCATION Argentina

WINNER Argentina

TEAMS 13

MATCHES 17

GOALS 70

AVERAGE\_GOALS 2.2

AVERAGE\_ATTENDANCE 20872

dtype: object

* **MAXIMUM**

df.max()

**output**

EDITION 2014 FIFA World Cup Brazil ™

YEAR 2014

LOCATION Uruguay

WINNER Uruguay

TEAMS 32

MATCHES 64

GOALS 171

AVERAGE\_GOALS 5.4

AVERAGE\_ATTENDANCE 68991

dtype: object

* **COUNT**

df.count()

**output**

EDITION 20

YEAR 20

LOCATION 20

WINNER 20

TEAMS 20

MATCHES 20

GOALS 20

AVERAGE\_GOALS 20

AVERAGE\_ATTENDANCE 20

dtype: int64

* **GROUPBY**

df=pd.read\_csv('/content/drive/MyDrive/drive/fifa-world-cup (1).csv')

grouped = df.groupby('YEAR')['GOALS'].sum()

print(grouped)

**output**

YEAR

1930 70

1934 70

1938 84

1950 88

1954 140

1958 126

1962 89

1966 89

1970 95

1974 97

1978 102

1982 146

1986 132

1990 115

1994 141

1998 171

2002 161

2006 147

2010 145

2014 171

Name: GOALS, dtype: int64

(b)Use Map, Filter, Reduce, and Lambda Functions with Pandas data frames

* **MAP**

import pandas as pd

#read csv file into Dataframe

df=pd.read\_csv('/content/drive/MyDrive/drive/fifa-world-cup (1).csv')

df['YEAR'] = df['YEAR'].map(lambda x:x\*3.10)

print(df)

**output**

EDITION YEAR LOCATION WINNER \

0 2014 FIFA World Cup Brazil ™ 6243.4 Brazil Germany

1 2010 FIFA World Cup South Africa ™ 6231.0 South Africa Spain

2 2006 FIFA World Cup Germany ™ 6218.6 Germany Italy

3 2002 FIFA World Cup Korea/Japan ™ 6206.2 Korea/Japan Brazil

4 1998 FIFA World Cup France ™ 6193.8 France France

5 1994 FIFA World Cup USA ™ 6181.4 USA Brazil

6 1990 FIFA World Cup Italy ™ 6169.0 Italy Germany

7 1986 FIFA World Cup Mexico ™ 6156.6 Mexico Argentina

8 1982 FIFA World Cup Spain ™ 6144.2 Spain Italy

9 1978 FIFA World Cup Argentina ™ 6131.8 Argentina Argentina

10 1974 FIFA World Cup Germany ™ 6119.4 Germany Germany

11 1970 FIFA World Cup Mexico ™ 6107.0 Mexico Brazil

12 1966 FIFA World Cup England ™ 6094.6 England England

13 1962 FIFA World Cup Chile ™ 6082.2 Chile Brazil

14 1958 FIFA World Cup Sweden ™ 6069.8 Sweden Brazil

15 1954 FIFA World Cup Switzerland ™ 6057.4 Switzerland Germany

16 1950 FIFA World Cup Brazil ™ 6045.0 Brazil Uruguay

17 1938 FIFA World Cup France ™ 6007.8 France Italy

18 1934 FIFA World Cup Italy ™ 5995.4 Italy Italy

19 1930 FIFA World Cup Uruguay ™ 5983.0 Uruguay Uruguay

TEAMS MATCHES GOALS AVERAGE\_GOALS AVERAGE\_ATTENDANCE

0 32 64 171 2.7 52918

1 32 64 145 2.3 49669

2 32 64 147 2.3 52491

3 32 64 161 2.5 42268

4 32 64 171 2.7 43517

5 24 52 141 2.7 68991

6 24 52 115 2.2 48388

7 24 52 132 2.5 46039

8 24 52 146 2.8 40571

9 16 38 102 2.7 40678

10 16 38 97 2.6 49098

11 16 32 95 3.0 50124

12 16 32 89 2.8 48847

13 16 32 89 2.8 27911

14 16 35 126 3.6 23423

15 16 26 140 5.4 29561

16 13 22 88 4.0 47511

17 15 18 84 4.7 20872

18 16 17 70 4.1 21352

19 13 18 70 3.9 32808

* **FILTER**

filtered\_df = df[df['MATCHES'] > 20]

print(filtered\_df)

**output**

EDITION YEAR LOCATION WINNER TEAMS \

0 2014 FIFA World Cup Brazil ™ 2014 Brazil Germany 32

1 2010 FIFA World Cup South Africa ™ 2010 South Africa Spain 32

2 2006 FIFA World Cup Germany ™ 2006 Germany Italy 32

3 2002 FIFA World Cup Korea/Japan ™ 2002 Korea/Japan Brazil 32

4 1998 FIFA World Cup France ™ 1998 France France 32

5 1994 FIFA World Cup USA ™ 1994 USA Brazil 24

6 1990 FIFA World Cup Italy ™ 1990 Italy Germany 24

7 1986 FIFA World Cup Mexico ™ 1986 Mexico Argentina 24

8 1982 FIFA World Cup Spain ™ 1982 Spain Italy 24

9 1978 FIFA World Cup Argentina ™ 1978 Argentina Argentina 16

10 1974 FIFA World Cup Germany ™ 1974 Germany Germany 16

11 1970 FIFA World Cup Mexico ™ 1970 Mexico Brazil 16

12 1966 FIFA World Cup England ™ 1966 England England 16

13 1962 FIFA World Cup Chile ™ 1962 Chile Brazil 16

14 1958 FIFA World Cup Sweden ™ 1958 Sweden Brazil 16

15 1954 FIFA World Cup Switzerland ™ 1954 Switzerland Germany 16

16 1950 FIFA World Cup Brazil ™ 1950 Brazil Uruguay 13

MATCHES GOALS AVERAGE\_GOALS AVERAGE\_ATTENDANCE

0 64 171 2.7 52918

1 64 145 2.3 49669

2 64 147 2.3 52491

3 64 161 2.5 42268

4 64 171 2.7 43517

5 52 141 2.7 68991

6 52 115 2.2 48388

7 52 132 2.5 46039

8 52 146 2.8 40571

9 38 102 2.7 40678

10 38 97 2.6 49098

11 32 95 3.0 50124

12 32 89 2.8 48847

13 32 89 2.8 27911

14 35 126 3.6 23423

15 26 140 5.4 29561

16 22 88 4.0 47511

* **REDUCE**

from functools import reduce

if not filtered\_df['MATCHES'].empty:

total = reduce(lambda x,y:x+y,filtered\_df['MATCHES'])

print(total)

else:

print("The filtered DataFrame is empty,cannot calculate total MATCHES.")

**output**

783

(c)Visualize the data set (At least 6 different plots).

1. **LINE PLOT**

import pandas as pd

d=pd.read\_csv('/content/drive/MyDrive/drive/fifa-world-cup (1).csv')

import matplotlib.pyplot as plt

plt.plot(d['YEAR'],d['GOALS'],marker='o',linestyle='-',color='red')

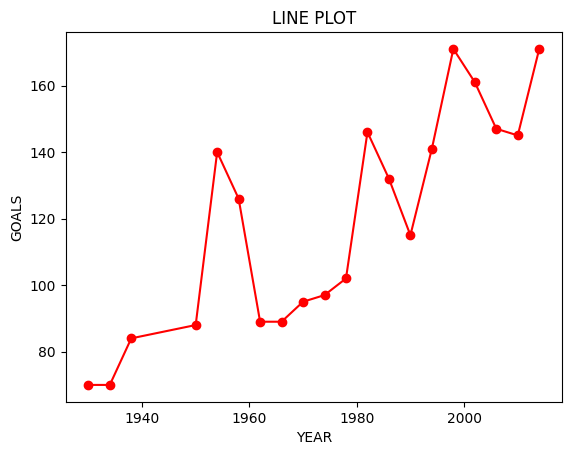
plt.title("LINE PLOT")

plt.xlabel('YEAR')

plt.ylabel('GOALS')

plt.show()

**output**

**2.BAR PLOT**

import pandas as pd

df=pd.read\_csv('/content/drive/MyDrive/drive/fifa-world-cup (1).csv')

import matplotlib.pyplot as plt

df.groupby('MATCHES')['GOALS'].sum().plot(kind='bar')

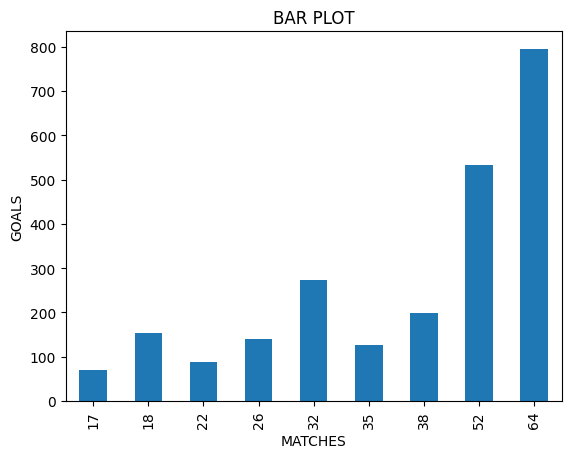
plt.title("BAR PLOT")

plt.xlabel('MATCHES')

plt.ylabel('GOALS')

plt.show()

**output**



**3.HISTOGRAM**

plt.hist(df['TEAMS'],bins=5,edgecolor='black')

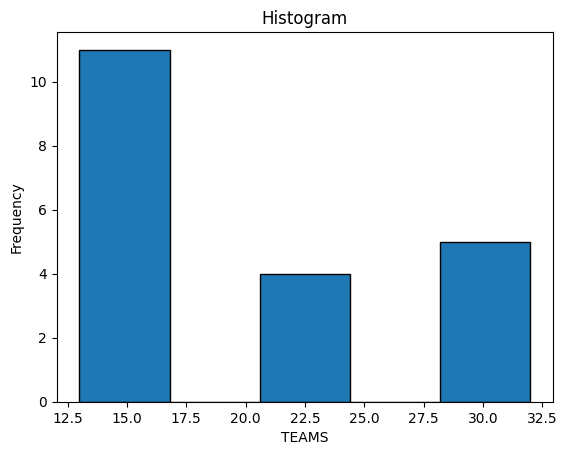
plt.title('Histogram')

plt.xlabel('TEAMS')

plt.ylabel('Frequency')

plt.show()

**output**



**4.SCATTER PLOT**

plt.scatter(d['YEAR'],d['GOALS'])

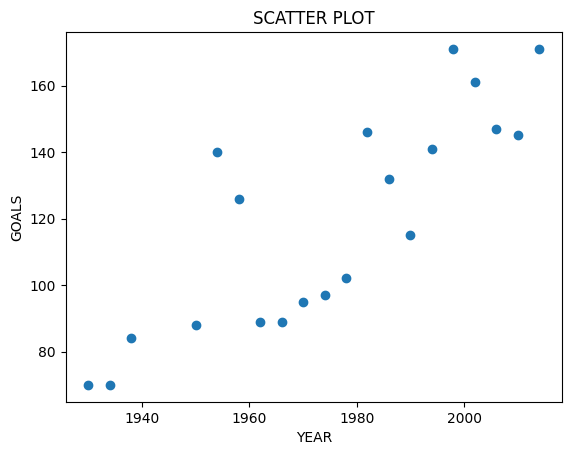
plt.title("SCATTER PLOT")

plt.xlabel('YEAR')

plt.ylabel('GOALS')

plt.show()

**output**



**5.BOX PLOT**

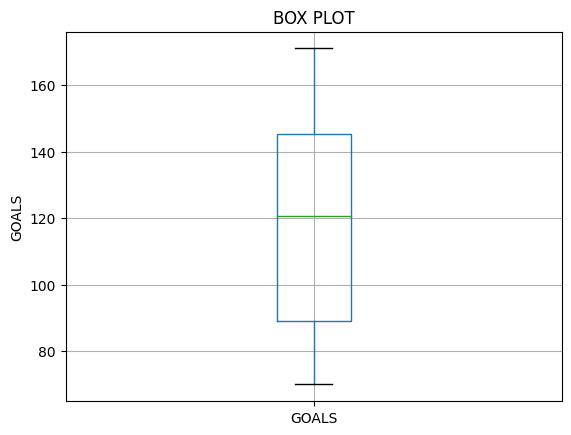
df.boxplot(column='GOALS')

plt.title("BOX PLOT")

plt.ylabel('GOALS')

plt.show()

**output**



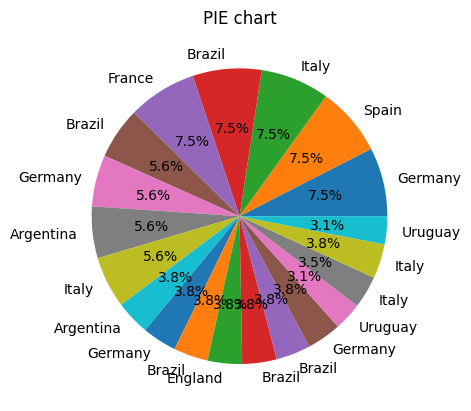
**6.PIE CHART**

plt.pie(d['TEAMS'],labels=d['WINNER'],autopct='%1.1f%%')

plt.title('PIE chart')

plt.show()

**output**



**DATASET:0.2**('/content/drive/MyDrive/drive/restaurants.csv')

**DESCRIPTION**

1. name
2. zipCode
3. neighborhood
4. councilDistrict
5. policeDistrict
6. Location 1

(**a) Aggregate functions.**

* **HEAD**

import pandas as pd

path='/content/drive/MyDrive/drive/restaurants.csv'

df=pd.read\_csv(path)

df.head(6)

**output**

name zipCode neighborhood councilDistrict policeDistrict Location 1

0 410 21206 Frankford 2 NORTHEASTERN 4509 BELAIR ROAD\nBaltimore, MD\n

1 1919 21231 Fells Point 1 SOUTHEASTERN 1919 FLEET ST\nBaltimore, MD\n

2 SAUTE 21224 Canton 1 SOUTHEASTERN 2844 HUDSON ST\nBaltimore, MD\n

3 #1 CHINESE KITCHEN 21211 Hampden 14 NORTHERN 3998 ROLAND AVE\nBaltimore, MD\n

4 #1 chinese restaurant 21223 Millhill 9 SOUTHWESTERN 2481 frederick ave\nBaltimore, MD\n

5 19TH HOLE 21218 Clifton Park 14 NORTHEASTERN 2722 HARFORD RD\nBaltimore, MD\n

* **TAIL**

import pandas as pd

path='/content/drive/MyDrive/drive/restaurants.csv'

df=pd.read\_csv(path)

df.tail(6)

**output**

name zipCode neighborhood councilDistrict policeDistrict Location 1

1321 ZEEBA LOUNGE 21230 Federal Hill 10 SOUTHERN 916 LIGHT ST\nBaltimore, MD\n

1322 ZEN WEST ROADSIDE CANTINA 21212 Rosebank 4 NORTHERN 5916 YORK RD\nBaltimore, MD\n

1323 ZIASCOS 21231 Washington Hill 1 SOUTHEASTERN 1313 PRATT ST\nBaltimore, MD\n

1324 ZINK'S CAF 21213 Belair-Edison 13 NORTHEASTERN 3300 LAWNVIEW AVE\nBaltimore, MD\n

1325 ZISSIMOS BAR 21211 Hampden 7 NORTHERN 1023 36TH ST\nBaltimore, MD\n

1326 ZORBAS 21224 Greektown 2 SOUTHEASTERN 4710 EASTERN Ave\nBaltimore, MD\n

* **sum**

df['zipCode'].sum()

**output**

28112213

* **MINIMUM**

df.min()

**output**

name #1 CHINESE KITCHEN

zipCode -21226

neighborhood Abell

councilDistrict 1

policeDistrict CENTRAL

Location 1 1 BIDDLE ST\nBaltimore, MD\n

dtype: object

* **MAXIMUM**

df.max()

**output**

name wozi lounge

zipCode 21287

neighborhood Wyman Park

councilDistrict 14

policeDistrict WESTERN

Location 1 Hopkins Pl\nBaltimore, MD\n

dtype: object

* **COUNT**

df.count()

**output**

name 1327

zipCode 1327

neighborhood 1327

councilDistrict 1327

policeDistrict 1327

Location 1 1327

dtype: int64

* **GROUPBY**

df=pd.read\_csv('/content/drive/MyDrive/drive/restaurants.csv')

grouped = df.groupby('name')['zipCode'].sum()

print(grouped)

**output**

name

#1 CHINESE KITCHEN 21211

#1 chinese restaurant 21223

1919 21231

19TH HOLE 21218

3 KINGS 21205

...

serdenes restaurant 21205

smiley's tavern 21223

subway 21202

v.f.w. post 2916 21225

wozi lounge 21212

Name: zipCode, Length: 1277, dtype: int64

(b)Use Map, Filter, Reduce, and Lambda Functions with Pandas data frames

* **MAP**

import pandas as pd

#read csv file into Dataframe

df=pd.read\_csv('/content/drive/MyDrive/drive/restaurants.csv')

df['councilDistrict'] = df['councilDistrict'].map(lambda x:x\*3.10)

print(df)

**output**

name zipCode neighborhood councilDistrict \

0 410 21206 Frankford 6.2

1 1919 21231 Fells Point 3.1

2 SAUTE 21224 Canton 3.1

3 #1 CHINESE KITCHEN 21211 Hampden 43.4

4 #1 chinese restaurant 21223 Millhill 27.9

... ... ... ... ...

1322 ZEN WEST ROADSIDE CANTINA 21212 Rosebank 12.4

1323 ZIASCOS 21231 Washington Hill 3.1

1324 ZINK'S CAF 21213 Belair-Edison 40.3

1325 ZISSIMOS BAR 21211 Hampden 21.7

1326 ZORBAS 21224 Greektown 6.2

policeDistrict Location 1

0 NORTHEASTERN 4509 BELAIR ROAD\nBaltimore, MD\n

1 SOUTHEASTERN 1919 FLEET ST\nBaltimore, MD\n

2 SOUTHEASTERN 2844 HUDSON ST\nBaltimore, MD\n

3 NORTHERN 3998 ROLAND AVE\nBaltimore, MD\n

4 SOUTHWESTERN 2481 frederick ave\nBaltimore, MD\n

... ... ...

1322 NORTHERN 5916 YORK RD\nBaltimore, MD\n

1323 SOUTHEASTERN 1313 PRATT ST\nBaltimore, MD\n

1324 NORTHEASTERN 3300 LAWNVIEW AVE\nBaltimore, MD\n

1325 NORTHERN 1023 36TH ST\nBaltimore, MD\n

1326 SOUTHEASTERN 4710 EASTERN Ave\nBaltimore, MD\n

[1327 rows x 6 columns]

* **FILTER**

filtered\_df = df[df['zipCode'] > 20]

print(filtered\_df)

**output**

name zipCode neighborhood councilDistrict \

0 410 21206 Frankford 6.2

1 1919 21231 Fells Point 3.1

2 SAUTE 21224 Canton 3.1

3 #1 CHINESE KITCHEN 21211 Hampden 43.4

4 #1 chinese restaurant 21223 Millhill 27.9

... ... ... ... ...

1322 ZEN WEST ROADSIDE CANTINA 21212 Rosebank 12.4

1323 ZIASCOS 21231 Washington Hill 3.1

1324 ZINK'S CAF 21213 Belair-Edison 40.3

1325 ZISSIMOS BAR 21211 Hampden 21.7

1326 ZORBAS 21224 Greektown 6.2

policeDistrict Location 1

0 NORTHEASTERN 4509 BELAIR ROAD\nBaltimore, MD\n

1 SOUTHEASTERN 1919 FLEET ST\nBaltimore, MD\n

2 SOUTHEASTERN 2844 HUDSON ST\nBaltimore, MD\n

3 NORTHERN 3998 ROLAND AVE\nBaltimore, MD\n

4 SOUTHWESTERN 2481 frederick ave\nBaltimore, MD\n

... ... ...

1322 NORTHERN 5916 YORK RD\nBaltimore, MD\n

1323 SOUTHEASTERN 1313 PRATT ST\nBaltimore, MD\n

1324 NORTHEASTERN 3300 LAWNVIEW AVE\nBaltimore, MD\n

1325 NORTHERN 1023 36TH ST\nBaltimore, MD\n

1326 SOUTHEASTERN 4710 EASTERN Ave\nBaltimore, MD\n

[1326 rows x 6 columns]

* **REDUCE**

from functools import reduce

if not filtered\_df['councilDistrict'].empty:

total = reduce(lambda x,y:x+y,filtered\_df['councilDistrict'])

print(total)

else:

print("The filtered DataFrame is empty,cannot calculate total councilDistrict.")

**output**

29552.299999999846

(c)Visualize the data set (At least 6 different plots).

1. **LINE PLOT**

import pandas as pd

d=pd.read\_csv('/content/drive/MyDrive/drive/restaurants.csv')

import matplotlib.pyplot as plt

plt.plot(d['zipCode'],d['councilDistrict'],marker='o',linestyle='-',color='purple')

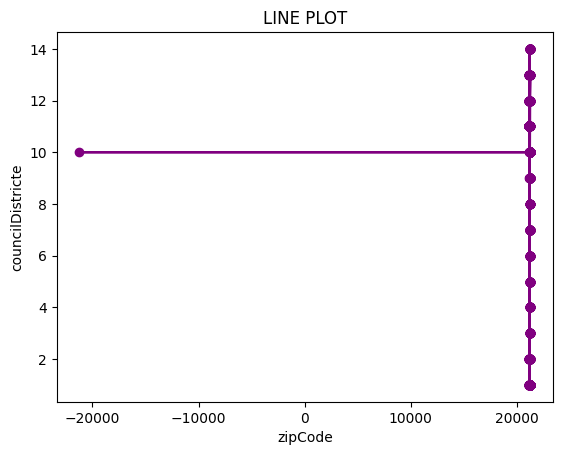
plt.title("LINE PLOT")

plt.xlabel('zipCode')

plt.ylabel('councilDistricte')

plt.show()

**output**



**2.BAR PLOT**

import pandas as pd

df=pd.read\_csv('/content/drive/MyDrive/drive/restaurants.csv')

import matplotlib.pyplot as plt

df.groupby('councilDistrict')['zipCode'].sum().plot(kind='bar')

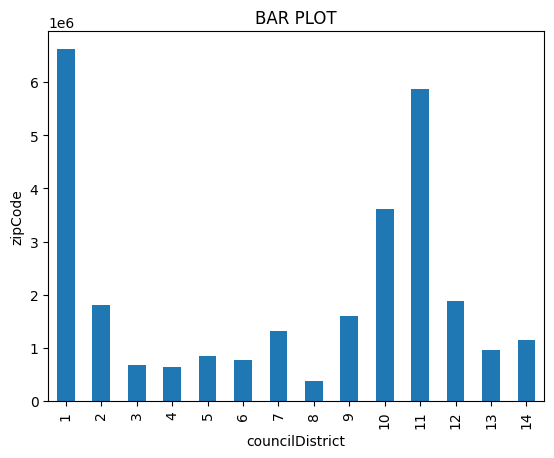
plt.title("BAR PLOT")

plt.xlabel('councilDistrict')

plt.ylabel('zipCode')

plt.show()

**output**



**3.HISTOGRAM**

plt.hist(df['councilDistrict'],bins=5,edgecolor='black')

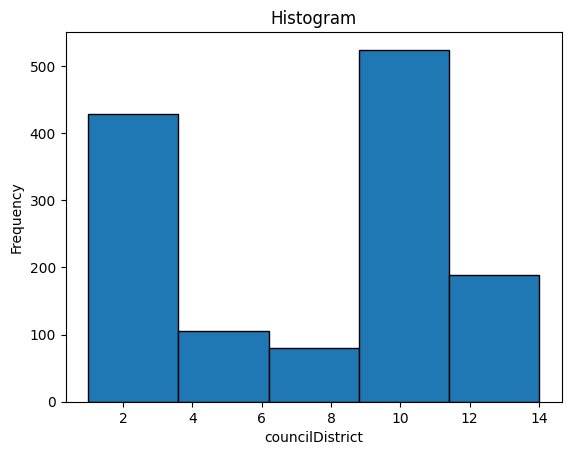
plt.title('Histogram')

plt.xlabel('councilDistrict')

plt.ylabel('Frequency')

plt.show()

**output**



**4.SCATTER PLOT**

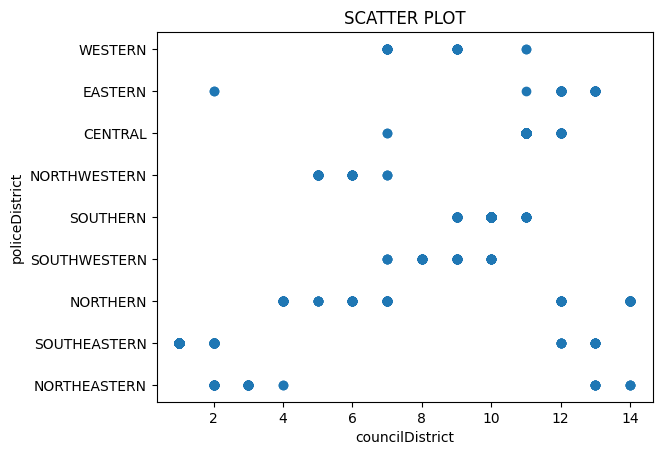
plt.scatter(d['councilDistrict'],d['policeDistrict'])

plt.title("SCATTER PLOT")

plt.xlabel('councilDistrict')

plt.ylabel('policeDistrict')

plt.show()

**output**

**5.BOX PLOT**

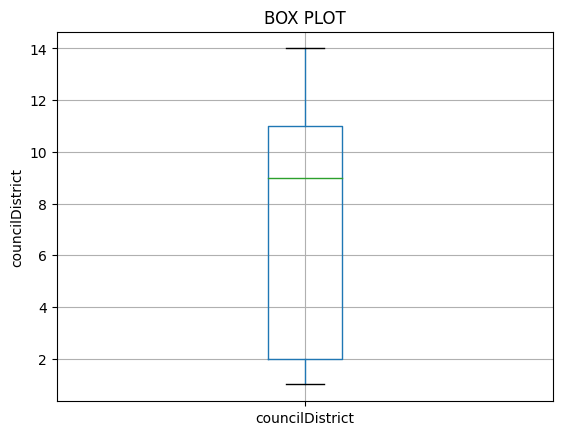
df.boxplot(column='councilDistrict')

plt.title("BOX PLOT")

plt.ylabel('councilDistrict')

plt.show()

**output**



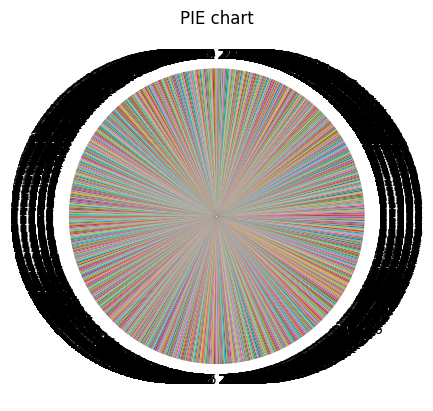
**6.PIE CHART**

plt.pie(d['councilDistrict'],labels=d['zipCode'])

plt.title('PIE chart')

plt.show()

**output**



**(d)How do you create a project plan and product backlog for an AI project?**

* + **classification Project – ML / deep learning**

Creating a project plan and product backlog for an AI project, specifically a Classification Project using Machine Learning (ML) or Deep Learning, involves several key steps to ensure clarity, feasibility, and successful execution. Here’s a structured approach:

**1. Define Project Objectives and Scope**

* **Objectives:** Clearly articulate what the project aims to achieve (e.g., improve accuracy of classification, develop a scalable model).
* **Scope:** Define the boundaries of the project (e.g., types of data to be classified, specific algorithms to be used).

**2. Identify Stakeholders and Resources**

* **Stakeholders:** Identify who will be involved (e.g., data scientists, domain experts, project managers).
* **Resources:** List the tools, technologies, and datasets required for the project.

**3. Create a High-Level Project Plan**

* **Milestones:** Define major phases (e.g., data collection, preprocessing, model selection, training, evaluation).
* **Timeline:** Estimate timeframes for each phase, considering dependencies and potential risks.

**4. Develop the Product Backlog**

* **User Stories:** Capture functional requirements from the perspective of users (e.g., data analysts, end-users of the classification results).
* **Prioritization:** Rank user stories based on importance and dependencies.
* **Epics and Tasks:** Break down larger functionalities (epics) into smaller, manageable tasks.

**Example Product Backlog Items:**

1. **Data Collection and Preprocessing**
   * Identify sources of data (epic)
   * Implement data scraping script (task)
   * Clean and preprocess data (task)
2. **Model Development**
   * Select appropriate ML algorithms (epic)
   * Implement baseline model (task)
   * Experiment with hyperparameters (task)
3. **Evaluation and Testing**
   * Define evaluation metrics (epic)
   * Develop test cases (task)
   * Conduct cross-validation (task)
4. **Deployment and Monitoring**
   * Integrate model with deployment pipeline (epic)
   * Implement monitoring for mod
   * el performance (task)
   * Develop documentation for end-users (task)

**5. Iterate and Refine**

* **Sprints:** Plan iterative development cycles (e.g., using Agile methodologies) to continuously improve the model and address feedback.
* **Review and Adapt:** Regularly review progress against the plan, adapt backlog items based on new insights or changes in requirements.

**Additional Considerations:**

* **Risk Management:** Identify potential risks (e.g., data quality issues, algorithm performance) and plan mitigation strategies.
* **Communication:** Ensure clear communication channels among team members and stakeholders.
* **Ethical Considerations:** Address ethical implications of data usage and model outputs.