GOVERNMENT POLYTECHNIC NAGAMANGALA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vth Semester Diploma

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

**Assignment: 02**

**NAME: KUSUMA D.R**

**ROLL NO: 158CS22023**

AIML (20CS51)

ASSIGNMENT – WEEK 02

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

1. Aggregate functions.
2. Use Map, Filter, Reduce, and Lambda Functions with Pandas data frames
3. Visualize the data set (At least 6 different plots).
4. 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.)
   1. Create a Git Repository for following the Regression Project ML / deep learning.
   2. Classification Project – ML / deep learning
   3. Clustering project – ML / deep learning
   4. Natural Language Processing – ML / deep learning

**Important Note:**

1. **Last Date for Submission:17-07-2024.**
2. **Everyone must perform the above operation using different datasets.**
3. **Submit the report to the email** [**aimlgtn@gmail.com**](mailto:aimlgtn@gmail.com)
4. Download any two datasets from the internet and perform the following operations.

**DATASET**:01 (“/content/iris (1).csv.xlsx”)

**Description:**

* Sepal.length
* Sepal.width
* Petal.width
* Petal.length
* Variety

**(a)Aggregate functions**.

* **HEAD**

import pandas as pd

path="/content/iris (1).csv.xlsx"

df=pd.read\_excel(path)

df.head(5)

**OUTPUT:**

| **sepal.length** | **sepal.width** | **petal.length** | **petal.width** | **variety** |
| --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | Setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | Setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | Setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | Setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | Setosa |

* **TAIL**

df.tail(10)

**OUTPUT:**

| **sepal.length** | **sepal.width** | **petal.length** | **petal.width** | **variety** |
| --- | --- | --- | --- | --- |
| **140** | 6.7 | 3.1 | 5.6 | 2.4 | Virginica |
| **141** | 6.9 | 3.1 | 5.1 | 2.3 | Virginica |
| **142** | 5.8 | 2.7 | 5.1 | 1.9 | Virginica |
| **143** | 6.8 | 3.2 | 5.9 | 2.3 | Virginica |
| **144** | 6.7 | 3.3 | 5.7 | 2.5 | Virginica |
| **145** | 6.7 | 3.0 | 5.2 | 2.3 | Virginica |
| **146** | 6.3 | 2.5 | 5.0 | 1.9 | Virginica |
| **147** | 6.5 | 3.0 | 5.2 | 2.0 | Virginica |
| **148** | 6.2 | 3.4 | 5.4 | 2.3 | Virginica |
| **149** | 5.9 | 3.0 | 5.1 | 1.8 | Virginica |

* **SUM**

df.sum()

**Output:**

sepal.length 876.5

sepal.width 458.6

petal.length 563.7

petal.width 179.9

variety SetosaSetosaSetosaSetosaSetosaSetosaSetosaSeto...

dtype: object

* **COUNT**

df.count()

**Output:**

sepal.length 150

sepal.width 150

petal.length 150

petal.width 150

variety 150

dtype: int64

* **MINIMUM**

df.min()

**Output:**

sepal.length 4.3

sepal.width 2.0

petal.length 1.0

petal.width 0.1

variety Setosa

dtype: object

* **MAXIMUM**

df.max()

**Output:**

sepal.length 7.9

sepal.width 4.4

petal.length 6.9

petal.width 2.5

variety Virginica

dtype: object

* **GROUPEDBY**

grouped =df.groupby('sepal.length')

print(grouped.agg({'sepal.length':'count'}))

**Output:**

sepal.length

sepal.length

4.3 1

4.4 3

4.5 1

4.6 4

4.7 2

4.8 5

4.9 6

5.0 10

5.1 9

5.2 4

5.3 1

5.4 6

5.5 7

5.6 6

5.7 8

5.8 7

5.9 3

6.0 6

6.1 6

6.2 4

6.3 9

6.4 7

6.5 5

6.6 2

6.7 8

6.8 3

6.9 4

7.0 1

7.1 1

7.2 3

7.3 1

7.4 1

7.6 1

7.7 4

* 1. 1

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

* **MAP**

import pandas as pd

d =pd.read\_excel('/content/iris (1).csv.xlsx')

d['petal.length']=d['petal.length'].map(lambda x: x\*2)

print(d)

**Output:**

sepal.length sepal.width petal.length petal.width variety

0 5.1 3.5 2.8 0.2 Setosa

1 4.9 3.0 2.8 0.2 Setosa

2 4.7 3.2 2.6 0.2 Setosa

3 4.6 3.1 3.0 0.2 Setosa

4 5.0 3.6 2.8 0.2 Setosa

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

145 6.7 3.0 10.4 2.3 Virginica

146 6.3 2.5 10.0 1.9 Virginica

147 6.5 3.0 10.4 2.0 Virginica

148 6.2 3.4 10.8 2.3 Virginica

149 5.9 3.0 10.2 1.8 Virginica

[150 rows x 5 columns]

* **FILTER**

filtered\_d = d[d['sepal.length'].map(lambda x:x>=5)]

print(filtered\_d)

**Output:**

sepal.length sepal.width petal.length petal.width variety

0 5.1 3.5 2.8 0.2 Setosa

4 5.0 3.6 2.8 0.2 Setosa

5 5.4 3.9 3.4 0.4 Setosa

7 5.0 3.4 3.0 0.2 Setosa

10 5.4 3.7 3.0 0.2 Setosa

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

145 6.7 3.0 10.4 2.3 Virginica

146 6.3 2.5 10.0 1.9 Virginica

147 6.5 3.0 10.4 2.0 Virginica

148 6.2 3.4 10.8 2.3 Virginica

149 5.9 3.0 10.2 1.8 Virginica

[128 rows x 5 columns]

* **REDUCE**

from functools import reduce

if not filtered\_d['sepal.length'].empty:

  total = reduce(lambda x, y: x+y,filtered\_d['sepal.length'])

  print(total)

else:

  print("the filtered DataFrame is empty,cannot calculate total sepal.length")

**output:**

773.3000000000002

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

* Line plot

import pandas as pd

d=pd.read\_excel('/content/iris (1).csv.xlsx')

import matplotlib.pyplot as plt

plt.plot(d['sepal.width'],d['petal.length'], color="black")

plt.title("line plot")

plt.xlabel('sepal.width')

plt.ylabel('petal.length')

plt.show()

**output:**

* **BAR PLOT**

plt.bar(d['sepal.width'],d['petal.length'],edgecolor='green')

plt.title("bar plot")

plt.xlabel('sepal.width')

plt.ylabel('petal.length')

plt.show()

**Output:**

* **HISTOGRAM**

plt.hist(d['sepal.width'],edgecolor='black')

plt.title("histogram ")

plt.xlabel('sepal.width')

plt.ylabel('frequency')

plt.show()

**Output:**

* **SCATTER Plot**

plt.scatter(d['sepal.width'],d['petal.length'], edgecolor='green')

plt.title("scatter plot")

plt.xlabel('sepal.width')

plt.ylabel('petal.length')

plt.show()

**Output:**

* **BOX plot**

d.boxplot(column=['petal.width'])

plt.title("box plot")

plt.ylabel('petal.width')

plt.show()

**Output:**

* **Pie Chart**

plt.pie(d['sepal.width'],labels=d['petal.length'])

plt.title("pie chart")

plt.show()

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |  |