WEEK-1(Extract data from different file formats and display the summary statistics)

AIM: Extract data from different file formats and display the summary statistics.

1.JSON: Java Script Object Notation (JSON) is one of the most widely used data interchange formats across the digital realm. JSON is a lightweight alternative to legacy formats like XML

```
import pandas as pd
df = pd.read_json(r"iris.json")
print(df.head())
```

```
   sepalLength  sepalWidth  petalLength  petalWidth species
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
```

```
import json
with open(r"iris.json",'r') as file:
    df = json.load(file)
print(df)
```

```
[{'sepalLength': 5.1, 'sepalWidth': 3.5, 'petalLength': 1.4, 'petalWidth': 0.2, 'spec
```

2)A CSV (Comma-Separated Values) file is a plain text file that stores tabular data in a structured format, with each line representing a row and each value separated by commas. It is commonly used for data exchange between different applications or for data storage in a simple, human-readable format.

```
import pandas as pd
df = pd.read_csv(r"iris.csv")
print(df.head())
```

```
   150    4  setosa  versicolor  virginica
0  5.1  3.5     1.4         0.2          0
1  4.9  3.0     1.4         0.2          0
2  4.7  3.2     1.3         0.2          0
3  4.6  3.1     1.5         0.2          0
4  5.0  3.6     1.4         0.2          0
```

```
l=[]
with open(r"cereal.csv",'r')as file:
    csv_reader = csv.reader(file)
    for row in csv_reader:
        l.append(row)
```

3)A TSV (Tab-Separated Values) file is a plain text file that uses tabs to separate data values. It is similar to a CSV (Comma-Separated Values) file, but instead of commas, tabs are used as the delimiter. Each line in the TSV file represents a row of data, and the values within each row are separated by tabs, allowing for structured data storage and easy parsing.

```
import pandas as pd
df = pd.read_csv(r"iris.tsv",delimiter="\t")
print(df.head())
```

```
   # sepal_length:lin  sepal_width:lin  petal_length:lin  petal_width:lin  \
0                 5.1              3.5               1.4              0.2
1                 4.9              3.0               1.4              0.2
2                 4.7              3.2               1.3              0.2
3                 4.6              3.1               1.5              0.2
4                 5.0              3.6               1.4              0.2

     label:nom
0  Iris-setosa
1  Iris-setosa
2  Iris-setosa
3  Iris-setosa
4  Iris-setosa
```

```
import csv
l=[]
with open(r"iris.tsv",'r')as file:
    tsv_reader = csv.reader(file,delimiter="\t")
    for row in tsv_reader:
        l.append(row)
```

4)From XML XML or eXtensible Markup Language is a markup language that defines rules for encoding data/documents to be shared across the Internet. Like JSON, XML is also a text format that is human readable. Its design goals involved strong support for various human languages (via Unicode), platform independence, and simplicity. XMLs are widely used for representing data of varied shapes and sizes. XMLs are widely used as configuration formats by different systems, metadata, and data representation format for services like RSS, SOAP, and many more.XML is a language with syntactic rules and schemas defined and refined over the years.The most import components of an XML are as follows: • Tag: A markup construct denoted by strings enclosed

components of an XML are as follows: • Tag: A markup construct denoted by strings enclosed with angled braces (“<” and “>”). • Content: Any data not marked within the tag syntax is the content of the XML file/object. • Element: A logical construct of an XML. An element may be defined with a start and an end tag with or without attributes, or it may be simply an empty tag. • Attribute: Key-value pairs that represent the properties or attributes of the element in consideration. These are enclosed within a start or an empty tag Xml files can be extracted in two ways 1) ElementTree 2) Minidom

```
import xml.etree.ElementTree as et


tree = et.parse(r"Books.xml")
root = tree.getroot()


print(root)
print(len(root))
```

```
     <Element 'bookstore' at 0x0000027814441F80>
     3
```

```
print(tree)
print(root[0])
```

```
     <xml.etree.ElementTree.ElementTree object at 0x00000278142E9A20>
     <Element 'book' at 0x0000027814441FD0>
```

```
for child in root:
    print(child[0].tag)
    print(child[0].text)
print(tree)
print(root[0].findall('title')[0].text)
```

```
     author
     Gambardella, Matthew
     author
     Ralls, Kim
     author
     Corets, Eva
     author
     Corets, Eva
     author
     Corets, Eva
     author
     Randall, Cynthia
     author
     Thurman, Paula
     author
     Knorr, Stefan
```

```
author
Kress, Peter
author
O'Brien, Tim
author
O'Brien, Tim
author
Galos, Mike
<xml.etree.ElementTree.ElementTree object at 0x000001FB82C66760>
XML Developer's Guide
```

5)HTML: An HTML (Hypertext Markup Language) file is a standard file format used for creating web pages. It contains structured elements and tags that define the structure and content of a web page. These tags are interpreted by web browsers to display the page's text, images, links, and other media.

```
import requests
from bs4 import BeautifulSoup
url = "https://www.google.com"
page = requests.get(url)
soup = BeautifulSoup(page.content,'html.parser')
a = soup.prettify()
#print(soup.prettify)
#print(page.content)
print(a[:225])
```

```
<!DOCTYPE html>
<html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN">
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <meta content="/logos/doodles/2023/zarina-hashmis-86th-bir
```

```
import subprocess

def convert_ipynb_to_pdf(notebook_path, output_path):
    command = f"jupyter nbconvert --to pdf {notebook_path} --output {output_path}"
    subprocess.call(command, shell=True)

# Usage example
notebook_path = 'week.ipynb'
output_path = 'safsdga\output.pdf'

convert_ipynb_to_pdf(notebook_path, output_path)
```

Colab paid products - Cancel contracts here