BTECH IV SEM – ML LAB EXERCISES – WEEK-02

Suppose we have a csv file *students.csv* and its contents are:

```
Id,Name,Course,City,Session
21,Mark,Python,London,Morning
22,John,Python,Tokyo,Evening
23,Sam,Python,Paris,Morning
32,Shaun,Java,Tokyo,Morning
```

We want to read all the rows of this csv file line by line and process each line at a time.

Also note that, here we don't want to read all lines into a list of lists and then iterate over it, because that will not be an efficient solution for large csv file i.e. file with size in GBs.

We are looking for solutions where we read & process only one line at a time while iterating through all rows of csv, so that minimum memory is utilized.

Python has a csv module, which provides two different classes to read the contents of a csv file i.e. ***csv.reader*** and ***csv.DictReader***.

With csv module's reader class object we can iterate over the lines of a csv file as a list of values, where each value in the list is a cell value.

```
from csv import reader

# open file in read mode
with open('students.csv', 'r') as read_obj:
    # pass the file object to reader() to get the reader object
    csv_reader = reader(read_obj)
    # Iterate over each row in the csv using reader object
    for row in csv_reader:
        # row variable is a list that represents a row in csv
        print(row)
```

Output:

```
['Id', 'Name', 'Course', 'City', 'Session']
['21', 'Mark', 'Python', 'London', 'Morning']
['22', 'John', 'Python', 'Tokyo', 'Evening']
['23', 'Sam', 'Python', 'Paris', 'Morning']
['32', 'Shaun', 'Java', 'Tokyo', 'Morning']
```

It iterates over all the rows of *students.csv* file. For each row it fetched the contents of that row as a list and printed that list.

How did it work ?

It performed the following steps,

1. Open the file 'students.csv' in read mode and create a file object.
2. Create a reader object (iterator) by passing file object in csv.reader() function.
3. Now once we have this reader object, which is an iterator, then use this iterator with for loop to read individual rows of the csv as list of values. Where each value in the list represents an individual cell.

This way only one line will be in memory at a time while iterating through csv file, which makes it a memory efficient solution.

Read csv file without header

suppose we want to skip the header and iterate over the remaining rows of csv file.

```
from csv import reader

# skip first line i.e. read header first and then iterate over each row od csv
as a list
with open('students.csv', 'r') as read_obj:
    csv_reader = reader(read_obj)
    header = next(csv_reader)
    # Check file as empty
    if header != None:
        # Iterate over each row after the header in the csv
        for row in csv_reader:
            # row variable is a list that represents a row in csv
```

print(row)

Output:

```
['21', 'Mark', 'Python', 'London', 'Morning']
['22', 'John', 'Python', 'Tokyo', 'Evening']
['23', 'Sam', 'Python', 'Paris', 'Morning']
['32', 'Shaun', 'Java', 'Tokyo', 'Morning']
Header was:
['Id', 'Name', 'Course', 'City', 'Session']
```

It skipped the header row of csv file and iterate over all the remaining rows of students.csv file. For each row it fetched the contents of that row as a list and printed that list. In initially saved the header row in a separate variable and printed that in end.

How did it work ?

As reader() function returns an iterator object, which we can use with Python for loop to iterate over the rows. But in the above example we called the next() function on this iterator object initially, which returned the first row of csv. After that we used the iterator object with for loop to iterate over remaining rows of the csv file.

Read csv file line by line using csv module DictReader object

With csv module's DictReader class object we can iterate over the lines of a csv file as a dictionary i.e.
for each row a dictionary is returned, which contains the pair of column names and cell values for that row.
Let's understand with an example,

```
from csv import DictReader

# open file in read mode
with open('students.csv', 'r') as read_obj:
    # pass the file object to DictReader() to get the DictReader object
    csv_dict_reader = DictReader(read_obj)
    # iterate over each line as a ordered dictionary
    for row in csv_dict_reader:
        # row variable is a dictionary that represents a row in csv
        print(row)
```

Output:

```
{'Id': '21', 'Name': 'Mark', 'Course': 'Python', 'City': 'London', 'Session': 'Morning'}
{'Id': '22', 'Name': 'John', 'Course': 'Python', 'City': 'Tokyo', 'Session': 'Evening'}
{'Id': '23', 'Name': 'Sam', 'Course': 'Python', 'City': 'Paris', 'Session': 'Morning'}
{'Id': '32', 'Name': 'Shaun', 'Course': 'Java', 'City': 'Tokyo', 'Session': 'Morning'}
```

DictReader class has a member function that returns the column names of the csv file as list.

```
from csv import DictReader

# open file in read mode
with open('students.csv', 'r') as read_obj:
    # pass the file object to DictReader() to get the DictReader object
    csv_dict_reader = DictReader(read_obj)
    # get column names from a csv file
    column_names = csv_dict_reader.fieldnames
    print(column_names)
```

Output:

```
['Id', 'Name', 'Course', 'City', 'Session']
```

Read specific columns (by column name) in a csv file while iterating row by row

```
from csv import DictReader

# iterate over each line as a ordered dictionary and print only few column by column name
with open('students.csv', 'r') as read_obj:
    csv_dict_reader = DictReader(read_obj)
    for row in csv_dict_reader:
        print(row['Id'], row['Name'])
```

Read specific columns (by column Number) in a csv file while iterating row by row

```
from csv import reader

# iterate over each line as a ordered dictionary and print only few column by column Number
with open('students.csv', 'r') as read_obj:
    csv_reader = reader(read_obj)
    for row in csv_reader:
        print(row[1], row[2])
```

WEEK-02 EXERCISES

mtcars.csv

1. Tabulate number of cylinders in the mtcars dataset table.mtcars$cylinder
2. Find the five number summary of milespergallon
3. Draw a histogram for hp
4. Box plot the miles per gallon
5. Find the avg weight of all cars
6. Find the car with the minimum displacement
7. Find the car with the maximum qsec.
8. Find the median horse power and find the car with the highest fuel efficiency
9. Find the car with the lowest fuel efficiency
10. Find the car with the best hp
11. Tabulate mpg for different number of gears
12. Do side by side boxplot of mpg for cars with am (automatic transmission) Vs cars with Manual transmission.

Cereals1 data set: First convert the .xls into ,.csv and import into Python.

1) Tabulate the following attributes: mfr, and type of cereals
2) Display the 5 number summary for all nutritional attributes starting from calories to vitamins

Note: A value of -1 in nutrients indicates a missing observation.

3) For missing values find and replace with arithmetic mean of the attributes.
4) Find and replace outliers with median.
5) Compare the 5 number summary before and after preprocessing.
6) Draw side-by-side box plots of Calories of Hot Vs cold cereals.
7) Are the attributes calories and consumer rating correlated?
8) Are the attributes mfr and consumer rating correlated?
9) Which is the best Vs worst cereal in terms of user rating?
10) Which is the best Vs worst cereal in terms of calories?
11) Rate the top 5 cereals in terms of user rating?
12) Identify the cereal with the highest sodium.
13) Identify the cereal with the lowest carbohydrate.
14) Are the variable shelf and sugar correlated?
15) Identify the manufacturer of the cereal with the highest sugar content.
16) Replace missing values with mean
17) Find outliers using 1.5 * IQR

1. Write a Python script to generate a dataset containing 3 columns as follows:
Column-1: RegNo = {A set of 20 regnos}
Column-2: Mark1 = {A set of 20 marks} #By using the rand function
Column-3: Mark2 = {A set of 20 marks} #By using the rand function
Column-3: Mark3 = {A set of 20 marks} #By using the rand function
The Columns numbered 2,3,4 should be generated using the random() or randint() function of random library.
Compute the 5 number summary of all the Marks columns and display in a tabular format.

2. Generate a Sine wave using the sine, arrange, cos and pi functions of python with appropriate labeling of X and Y-axes.

3. Write a python script file to accept the file name from the user and display the content in a csv file. The script file should perform the following functions:
(a) Write the data as comma separated values in the form of rows and columns into a .txt file containing the following details:
Column-1: SNO = {A set of 5 Integers}
Column-2: Product Names = {A set of 5 product Names =
[Soaps, Biscuits, Chocolates, WaterBottles, Pastes] }
Column-3: UnitPrice = {A set of 5 prices in rupees = [25, 35, 45, 55, 65]}
(b) Convert this text file into .csv file using either csv module or pandas library and display the .csv file.
(c) Display the number of rows and columns in the .csv file. Also extract the column names and display them.

4. Draw a scatter plot for the data given below by taking Hours-Studied as X-axis and ExamGrade as Y-axis. Also write a function to determine the correlation between these two variables.
Col1-Hours Studied: 2, 9, 5, 5, 3, 7, 1, 8, 6, 2
Col2-ExamGrade: 69, 98, 82, 77, 71, 84, 55, 94, 84, 64