



PORT CITY INTERNATIONAL UNIVERSITY

Report Name: Data Preparation and Basic Python Operations for Pattern Recognition

Course Code: CSE 332

Course Title: Pattern Recognition Sessional

SUBMITTED TO

Name of Teacher: Ratul Barua

Designation: Lecturer

Department: CSE

SUBMITTED BY

Name of Student: Shaik Abdul Ahad

Program: BSc. in CSE

ID No.: CSE 029 07666

Batch No.: CSE-29-E-A

Experiment No: 01

Experiment Name: Data Preparation and Basic Python Operations for Pattern Recognition

Objective

- To familiarize with basic Python programming for data handling.
- To practice data preparation techniques such as list creation, random data generation, and DataFrame manipulation using Pandas for future use in Pattern Recognition tasks.
- To generate datasets for medical applications suitable for Pattern Recognition analysis.

Introduction

Pattern Recognition involves the classification or description of observations. Before applying machine learning algorithms or recognition models, data preparation is essential. This experiment focuses on the preparation of structured datasets using Python. Techniques such as handling user input, creating lists and dictionaries, random data generation, and using Pandas DataFrames are demonstrated to build foundational skills for advanced Pattern Recognition experiments.

Tools and Technologies Used

Programming Language: Python 3.x

Libraries:

- pandas: for data manipulation and DataFrame creation
- random: for generating random numbers

Platform: Jupyter Notebook

Experiment Procedure

- Initialize basic variables and demonstrate data types.
- Collect numeric and string data from user inputs.
- Use loops to populate lists with values.
- Create structured datasets (dictionary and Pandas DataFrame).
- Generate random data using the random module for simulation of real-world scenarios.
- Create and export datasets suitable for Pattern Recognition tasks.
- Display all intermediate outputs to verify correctness.

Code Implementation and Output

#1 Basic python

```
A = 6
```

```
A
```

```
6
```

#2 Variable with string value

```
a = '10'
```

```
a
```

```
'10'
```

#3 Input from user (requires input)

```
b = int(input())
```

```
b
```

```
12
```

```
12
```

#4 Empty list creation

```
a = []
```

```
a
```

```
[]
```

#5 Array of strings input

```
arr = []
```

```
n = int(input("Enter the number of string elements: "))
```

```
for i in range(n):
```

```
    element = input("Enter a string: ")
```

```
    arr.append(element)
```

```
print("Array of strings:", arr)
```

```
Enter the number of string elements: 3
```

```
Enter a string:
```

```
Shaik
```

```
Enter a string: abdul
```

```
Enter a string:
```

```
ahad
```

```
Array of strings: ['Shaik', 'abdul', 'ahad']
```

#7 Collecting student data: names, IDs, CGPAs

```
names, ids, cgs = [], [], []
```

```
for i in range(3):
```

```
    a = input("Enter a name: ")
```

```

names.append(a)
a = input("Enter an id: ")
ids.append(a)
a = input("Enter a CGPA: ")
cgs.append(a)
print("Names:", names)
print("IDs:", ids)
print("CGPAs:", cgs)

```

```

Enter a name: Shaik
Enter an id: 666
Enter a CGPA: 3.95
Enter a name: Abdul
Enter an id: 444
Enter a CGPA: 3.90
Enter a name: Ahad
Enter an id: 555
Enter a CGPA: 3.92
Names: ['Shaik', 'Abdul', 'Ahad']
IDs: ['666', '444', '555']
CGPAS: ['3.95', '3.90', '3.92']

```

#8 Creating dictionary from collected data

```

dict = {'Name': names, 'ID': ids, 'CGPA': cgs}
dict

```

```

{'Name': ['sjaol', 'abdul', 'ahad'], 'ID': ['34', '45', '89'], 'CGPA': ['6.33', '3.22', '4.66']}

```

#9 Creating pandas DataFrame

```

import pandas as pd
df = pd.DataFrame(data=dict)
df

```

```

   Name  ID CGPA
0  Shaik 666  3.95
1  Abdul 444  3.90
2  Ahad  555  3.92

```

#10 Export DataFrame to CSV file

```

df.to_csv('Testdata.csv')

```

#11 Random integer generation

```
import random
```

```
x = random.randint(25,30)
```

```
x
```

```
27
```

#12 Adding random age data to DataFrame

```
age = []
```

```
for i in range(0,3):
```

```
    x = random.randint(25,30)
```

```
    age.append(x)
```

```
age
```

```
df["AGE"] = age
```

```
df
```

```
[27,25,25]
```

#13 Generating random dataset for medical parameters

```
age, hemoglobin, tg, cholestro = [], [], [], []
```

```
for i in range(10):
```

```
    x = random.randint(25,30)
```

```
    age.append(x)
```

```
    a = random.uniform(10,18)
```

```
    hemoglobin.append(a)
```

```
    a = random.uniform(5,40)
```

```
    tg.append(a)
```

```
    a = random.uniform(10,60)
```

```
    cholestro.append(a)
```

```
dict = {'AGE': age, 'Hemoglobin': hemoglobin, 'TG': tg, 'Cholestro': cholestro}
```

```
df = pd.DataFrame(data=dict)
```

```
df
```

	AGE	Hemoglobin	TG	Cholestro
0	28	(11.926475927945287, 2)	16.423498	58.785715
1	27	(16.954310043638575, 2)	8.622159	49.251482
2	27	(11.634577073797473, 2)	26.925844	46.860351
3	30	(17.83763079710879, 2)	19.774321	48.526809
4	27	(17.060469439125665, 2)	32.893163	36.114842
5	26	(14.444378736014716, 2)	18.605102	13.978866
6	29	(17.468263406086585, 2)	22.240301	23.218538
7	26	(12.108470350071347, 2)	12.359030	12.056346
8	25	(16.955475574896028, 2)	9.012955	23.758119
9	28	(12.985720524888292, 2)	7.498624	24.532987

#14 Generating a random float between 0 and 1 rounded to 2 decimal places

```
x = round(random.random(), 2)
```

x

0.51

Discussion / Explanation

- The experiment successfully demonstrated basic Python syntax and data operations relevant to Pattern Recognition.
- Random data generation for medical parameters (Age, Hemoglobin, TG, Cholesterol) simulates real-life datasets for further analysis in machine learning or pattern classification tasks.
- Exporting datasets to CSV enables easy reuse for future experiments.
- This exercise reinforced the importance of data preparation, which is often more critical than the recognition algorithm itself.

Conclusion

This experiment provided essential hands-on practice in preparing datasets using Python, which is crucial for Pattern Recognition and Machine Learning. By generating structured datasets, the groundwork has been laid for applying advanced analytical and recognition techniques in future experiments.