VSIT

Applied Artificial Intelligence Practical # 6

Name	Ninad Karlekar	**Roll Number**	22306A1012
Subject/Course:	Applied Artificial Intelligence	**Class**	M.Sc. IT – Sem III
Topic	Implement a Fuzzy based application.	**Batch**	1

Topic Design a Fuzzy based application.

a) **AIM: Design a Fuzzy based operations using Python / R.**

DESCRIPTION:

What is Fuzzy Set ?

Fuzzy refers to something that is unclear or vague . Hence, Fuzzy Set is a Set where every key is associated with value, which is between 0 to 1 based on the certainty .This value is often called as degree of membership. Fuzzy Set is denoted with a Tilde Sign on top of the normal Set notation.

Operations on Fuzzy Set with Code :-

1. Union :

Consider 2 Fuzzy Sets denoted by A and B, then let's consider Y be the Union of them, then for every member of A and B, Y will be:

degree_of_membership(Y)= max(degree_of_membership(A), degree_of_membership(B))

2. Intersection :

Consider 2 Fuzzy Sets denoted by A and B, then let's consider Y be the Intersection of them, then for every member of A and B, Y will be:

degree_of_membership(Y)= min(degree_of_membership(A), degree_of_membership(B))

3. Complement :

Consider a Fuzzy Sets denoted by A , then let's consider Y be the Complement of it, then for every member of A , Y will be:

degree_of_membership(Y)= 1 - degree_of_membership(A)

4. Difference :

Consider 2 Fuzzy Sets denoted by A and B, then let's consider Y be the Intersection of them, then for every member of A and B, Y will be:

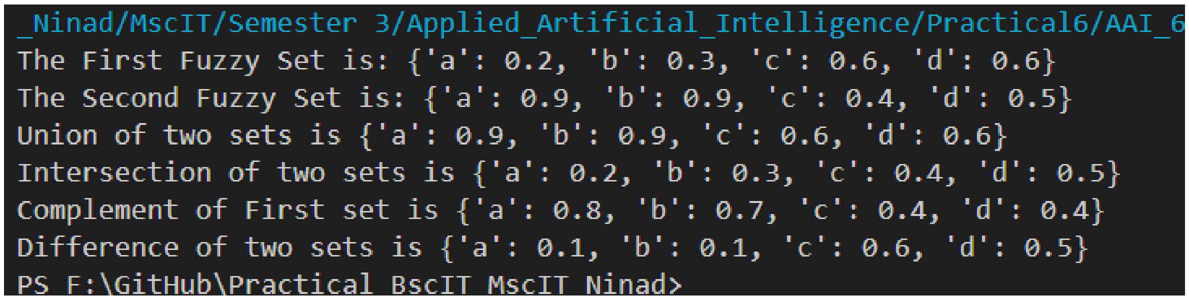
degree_of_membership(Y)= min(degree_of_membership(A), 1-degree_of_membership(B))

Code:

```
A = dict()
```

```
B = dict()
Y = dict()
# Initialize the dictionaries for fuzzy sets A, B, and the result
A = {"a": 0.2, "b": 0.3, "c": 0.6, "d": 0.6}
B = {"a": 0.9, "b": 0.9, "c": 0.4, "d": 0.5}
result = {}
# Display the fuzzy sets A and B
print('The First Fuzzy Set is:', A)
print('The Second Fuzzy Set is:', B)
# Fuzzy Set Union
for i in A:
    if A[i] > B[i]:
        result[i] = A[i]
    else:
        result[i] = B[i]
print("Union of two sets is", result)
# Fuzzy Set Intersection
result = {}
for i in A:
    if A[i] < B[i]:
        result[i] = A[i]
    else:
        result[i] = B[i]
print("Intersection of two sets is", result)
# Fuzzy Set Complement
result = {}
for i in A:
    result[i] = round(1 - A[i], 2)
print("Complement of First set is", result)
# Fuzzy Set Difference
result = {}
for i in A:
    result[i] = round(min(A[i], 1 - B[i]), 2)
print("Difference of two sets is", result)
```

Output:



```
_Ninad/MscIT/Semester 3/Applied_Artificial_Intelligence/Practical6/AAI_6
The First Fuzzy Set is: {'a': 0.2, 'b': 0.3, 'c': 0.6, 'd': 0.6}
The Second Fuzzy Set is: {'a': 0.9, 'b': 0.9, 'c': 0.4, 'd': 0.5}
Union of two sets is {'a': 0.9, 'b': 0.9, 'c': 0.6, 'd': 0.6}
Intersection of two sets is {'a': 0.2, 'b': 0.3, 'c': 0.4, 'd': 0.5}
Complement of First set is {'a': 0.8, 'b': 0.7, 'c': 0.4, 'd': 0.4}
Difference of two sets is {'a': 0.1, 'b': 0.1, 'c': 0.6, 'd': 0.5}
PS F:\GitHub\Practical_BscIT_MscIT_Ninad>
```

b) AIM: Design a Fuzzy based application using Python / R.

DESCRIPTION:

FuzzyWuzzy is a library of Python which is used for string matching. Fuzzy string matching is the process of finding strings that match a given pattern. Basically it uses Levenshtein Distance to calculate the differences between sequences.
FuzzyWuzzy has been developed and open-sourced by SeatGeek, a service to find sport and concert tickets.

FuzzyWuzzy Functions:
1] fuzz.ratio(s1, s2): This calculates the simple ratio similarity between s1 and s2.
2] fuzz.partial_ratio(s1, s2): This function computes a similarity ratio that considers partial
matches between s1 and s2.
3] fuzz.token_sort_ratio(s1, s2): This ratio considers the similarity of words in the two strings after sorting them alphabetically.
4] fuzz.token_set_ratio(s1, s2): This ratio considers the intersection and union of words (tokens) between s1 and s2. fuzz.WRatio(s1, s2): This ratio is like fuzz.ratio, but it tries to account for differences in capitalization, word ordering, and some other factors by using a weighted algorithm.

Code:

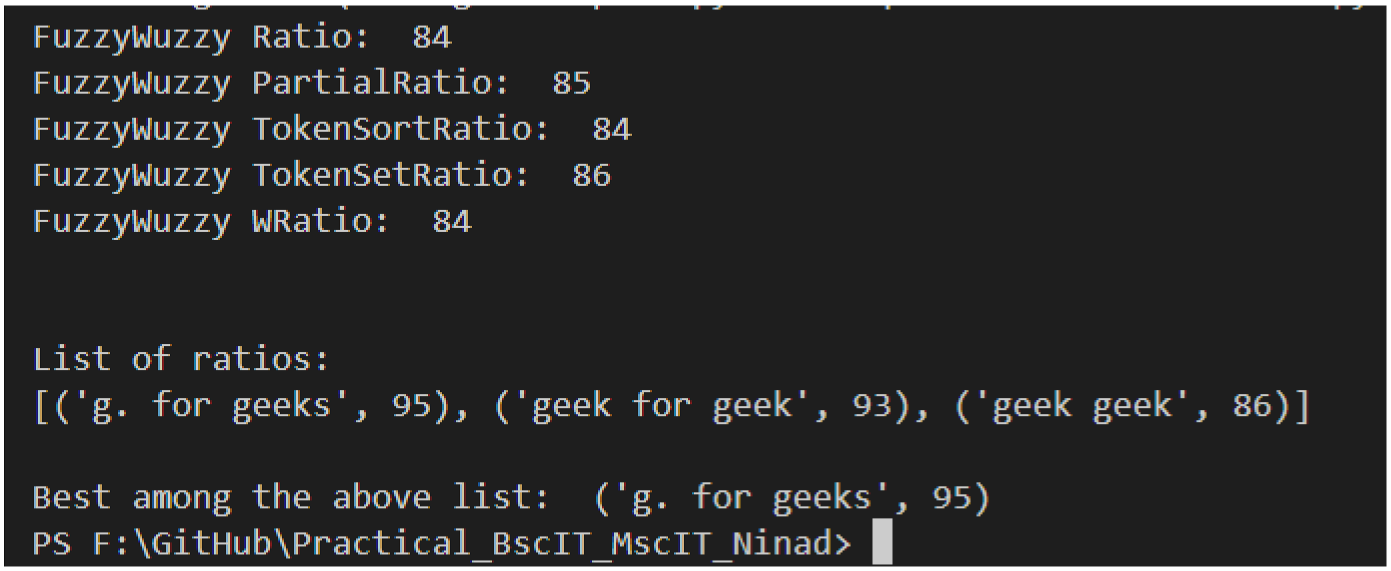
```
# AAI 6B: AIM: Design a Fuzzy based application using Python / R.

# !pip install fuzzywuzzy
from fuzzywuzzy import fuzz
from fuzzywuzzy import process

s1 = "I love GeeksforGeeks"
s2 = "I am loving GeeksforGeeks"
print("FuzzyWuzzy Ratio: ", fuzz.ratio(s1, s2))
print("FuzzyWuzzy PartialRatio: ", fuzz.partial_ratio(s1, s2))
print("FuzzyWuzzy TokenSortRatio: ", fuzz.token_sort_ratio(s1, s2))
print("FuzzyWuzzy TokenSetRatio: ", fuzz.token_set_ratio(s1, s2))
print("FuzzyWuzzy WRatio: ", fuzz.WRatio(s1, s2), "\n\n")

# for process library,
query = "geeks for geeks"
choices = ["geek for geek", "geek geek", "g. for geeks"]
print("List of ratios: ")
print(process.extract(query, choices), "\n")
print("Best among the above list: ", process.extractOne(query, choices))
```

Output:



```
FuzzyWuzzy Ratio:  84
FuzzyWuzzy PartialRatio:  85
FuzzyWuzzy TokenSortRatio:  84
FuzzyWuzzy TokenSetRatio:  86
FuzzyWuzzy WRatio:  84


List of ratios:
[('g. for geeks', 95), ('geek for geek', 93), ('geek geek', 86)]

Best among the above list:  ('g. for geeks', 95)
PS F:\GitHub\Practical_BscIT_MscIT_Ninad>
```