|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | 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.** | | | |
| 1. **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:** | | | |
| 1. **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](https://en.wikipedia.org/wiki/Levenshtein_distance" \t "_blank) to calculate the differences between sequences. [FuzzyWuzzy](https://github.com/seatgeek/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:** | | | |