**AI\_6 Aim: Design a Fuzzy based application using Python.**

 Durgesh Rajdev Vishwakarma.

 MSc IT Part 2 - Sem 3

 PRN: 2015430016

 VPM's B. N. Bandodkar College of Science, Thane.

In below example I have used Fuzzy logic to determine Fan Speed based on temperature or city.

In Fuzzy Logic objects are not always in one of two states (true or false), but rather in several states at one time.  
First, we'll review temperature and fan speed graph to understand need of fuzzy logic in this application.

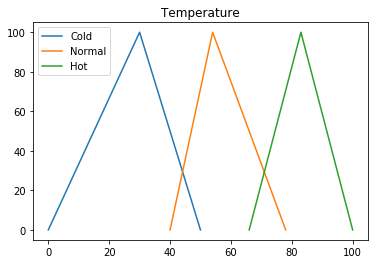
As you can see in below graph that we are unable to solve this problem with traditional binary logic.

**1. Temperature Graph**

In [1]:

|  |
| --- |
| **import** matplotlib.pyplot **as** plt  **%**matplotlib inline  ​  *#temprature*  cold **=** [0, 30, 50]  normal **=** [40, 54, 78]  hot **=** [66, 83, 100]  percentage **=** [0, 100, 0]  ​  plt.plot(cold, percentage)  plt.plot(normal, percentage)  plt.plot(hot, percentage)  plt.legend( ["Cold", "Normal", "Hot"] )  plt.title("Temperature")  plt.show() |

**Output:**

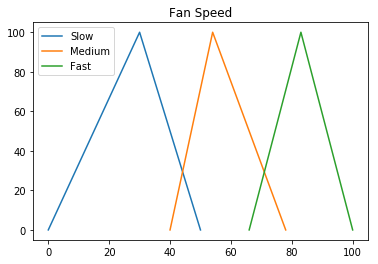


**2. Fan Speed Graph**

In [2]:

|  |
| --- |
| *#Fan Speed*  slow **=** [0, 30, 50]  medium **=** [40, 54, 78]  fast **=** [66, 83, 100]  ​  percentage **=** [0, 100, 0]  ​  plt.plot(cold, percentage)  plt.plot(medium, percentage)  plt.plot(fast, percentage)  plt.legend( ["Slow", "Medium", "Fast"] )  plt.title("Fan Speed")  plt.show() |

**Output:**



**3. Function definition**

I have defined below functions in this application:

 **Membership(x,List):**Function determines membership of x in the List.

 **Is(x,List):**Function is used to find whether x is member of List or not.

 **findSpeed(temp):**Function determines fan speed based on temperature.

In [3]:

|  |
| --- |
| *#FUNCTIONS*  **def** Membership(x,List):  "Returns the membership of a value in a list."  top**=**(float(x)**-**List[0])  bottom**=**(List[**-**1]**-**List[0])  M**=** top**/**bottom  **return** M  *#print( "Membership: " + str(M) )*  ​  **def** Is(x,List):  "Returns true if a value is in the value range of a list"  *#If a value is greater than the first item in a list..*  **if** x **>=** List[0]:  *#And if it is smaller than the last item in the list...*  **if** x**<=** List[**-**1]:  *#print the membership of the item in the list...*  Membership(x,List)  *#And return True*  **return** "Yes"  *#No else statement is needed since the return statement will exit the function.*  *#Print the membership and return False if the above condition is false.*  Membership(x,List)  **return** "No"  ​  *#Determine Fan speed based on temperature*  **def** findSpeed(temp):  "Determine Fan Speed based temperature or city"    temp\_cold **=** Membership(temp, cold)  temp\_medium **=** Membership(temp, normal)  temp\_hot **=** Membership(temp, hot)    *#Fuzzy logic values lies between 0 & 1 (if membership is less than 0 then make it 2 & filter them)*  **if** temp\_hot **<** 0:  temp\_hot **=** 2  **if** temp\_medium **<** 0:  temp\_medium **=** 2  **if** temp\_cold **<** 0:  temp\_cold **=** 2    **if**( temp\_cold **<=** temp\_medium):  **return** "Slow"  **elif**( temp\_medium **<=** temp\_hot):  **return** "Medium"  **else**:  **return** "Fast"  ​  **def** document(f):  *#Print the name of a function and its document string.*  print( f.\_\_name\_\_ **+**': '**+** f.\_\_doc\_\_**+**'\n' ) |

In [4]:

|  |
| --- |
| *# Function Documentation*  document(Is)  document(Membership)  document(findSpeed) |

**Output:**

|  |
| --- |
| Is: Returns true if a value is in the value range of a list  Membership: Returns the membership of a value in a list.  findSpeed: Determine Fan Speed based temperature or city |

**4. Variable declaration**

I have defined a range of temperature of cold, normal and hot as below, which we'll use to determine whether given temperature is cold, normal or hot.  
Based on which we'll take next action (decide fan speed: slow, normal or fast).

In [5]:

|  |
| --- |
| *#temprature range*  cold **=** (0, 50)  normal **=** (40, 78)  hot **=** (66, 100)  ​  *# Average temperature of cities*  Mumbai **=** 88  Pune **=** 64  Shimla **=** 17  Jaipur **=** 90 |

**5. Verify function with sample cities**

Here we'll verify functionality of Is() function before combining it with findSpeed() function.  
As you can see application is able to determine cold, normal or hot based given temperature.

In [6]:

|  |
| --- |
| print( "Is Shimla cold: " **+** str( Is(Shimla, cold) ))  print( "Is Shimla normal: " **+** str( Is(Shimla, normal) ))  print( "Is Shimla hot: " **+** str( Is(Shimla, hot) )) |

**Output:**

|  |
| --- |
| Is Shimla cold: Yes  Is Shimla normal: No  Is Shimla hot: No |

In [7]:

|  |
| --- |
| print( "Is Pune cold: " **+** str( Is(Pune, cold) ))  print( "Is Pune normal: " **+** str( Is(Pune, normal) ))  print( "Is Pune hot: " **+** str( Is(Pune, hot) )) |

**Output:**

|  |
| --- |
| Is Pune cold: No  Is Pune normal: Yes  Is Pune hot: No |

In [8]:

|  |
| --- |
| print( "Is Mumbai cold: " **+** str( Is(Mumbai, cold) ))  print( "Is Mumbai normal: " **+** str( Is(Mumbai, normal) ))  print( "Is Mumbai hot: " **+** str( Is(Mumbai, hot) )) |

**Output:**

|  |
| --- |
| Is Mumbai cold: No  Is Mumbai normal: No  Is Mumbai hot: Yes |

**6. Verify Fuzzy Logic with city & temperature**

I have used 3 diffrent cities and temperatures values to verify our application which we have created using Fuzzy logic.

In [9]:

|  |
| --- |
| print("Fan speed for Shimla: " **+** str( findSpeed(Shimla) ) )  print("Fan speed for Pune: " **+** str( findSpeed(Pune) ) )  print("Fan speed for Jaipur: " **+** str( findSpeed(Jaipur) ) )  print("\n")  print("Fan speed at 20: " **+** str( findSpeed(20) ) )  print("Fan speed at 60: " **+** str( findSpeed(60) ) )  print("Fan speed at 90: " **+** str( findSpeed(90) ) ) |

**Output:**

|  |
| --- |
| Fan speed for Shimla: Slow  Fan speed for Pune: Medium  Fan speed for Jaipur: Fast  Fan speed at 20: Slow  Fan speed at 60: Medium  Fan speed at 90: Fast |

**Conclusion:**

* We have successfully designed a Fuzzy based application using Python.
* For more details, please review PPT of this.