REPORT FOR WASHING MACHINE(fuzzy - controller)

As a project work for Course

SOFT COMPUTING (INT 246)

Name : Deepak Kumar
Registration Number : 11914243
Name : Devyansh
Registration Number : 11907819
Program : CSE B.Tech (ML)
Semester : 5th
School : School of Computer Science and Engineering
Name of the University : Lovely Professional University

Date of submission :19th NOVEMBER 2021

Lovely Professional University
Jalandhar, Punjab, India.

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

WASHING MACHINE

ABSTRACT:-

The washing machine is widely used in the present day's society for washing and drying of clothes. Th Most of the raw materials used in the manufacturing process of the washing machine have considerable negative social and environmental impacts which include mass scale extraction of metals, releases of toxic gases during production and processing, contamination of land and health hazards caused in humans and animals. Therefor it is important to mitigate these effects while achieving the social requirements.

It is believed that providing a service to replace the domestic washing machine would be a feasible solution for this dilemma. A service can be introduced where all the dirty laundry will be collected on a weekly basis, washed, dried, ironed and then returned to the users after a certain number of days. It is suggested that these services be well organised and functioned as a service to the community than a business to individual service providing companies. Therefor a fixed charge for a fixed volume or number of clothes is strongly recommended

ACKNOWLEDGEMENT:-

I would like to thank my mentor - Prof. Sagar Pande for his advice and inputs on this project. Many thanks to my friends and seniors as well, who spent countless hours to listen and provide feedbacks.

Table of Contents

INTRODUCTION:-

1.1 Context

This project has been done as part of my course for the CSE at Lovely Professional University. Supervised by Sagar Pande, I have one months to fulfill the requirements in order to succeed the module.

1.2 Motivations

Being extremely interested in everything having a relation with the Machine Learning, the group project was a great occasion to give us the time to learn and confirm our interest for this field. The fact that we can make display result for linguistic variable and give the ability for machines . We can use Machine Learning in Finance, Medicine, almost everywhere. That's why I decided to conduct my project around the Machine Learning.

1.3 Idea:-

As a first experience, we wanted to make my project as much didactic as possible by approaching every different steps of the machine learning process and trying to understand them deeply. As our first project " CHAT BOT" the problems that are not immediate scientific interest but useful to illustrate and practice, we chose to take WASHING MACHINE.

TEAM MEMBERS:-

Deepak Kumar:-

Contributions:-

1. Coding(joined)
2. Linguistic variable convert into equation
3. Triangular rule (applied)
4. Machine lerning(joined)

Devyansh Shukla:-

Contributions:-

1. Coding(joined)
2. By using inbuilt function (numpy.arrange) create linguistic variable
3. Mamdani approach
4. Using Auto membership function
5. Machine learning(joined)

LIBRARIES:-

Numpy:-

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python.

As the whole project is based on whole complex stats ,we will use this fast calculations and provide results.

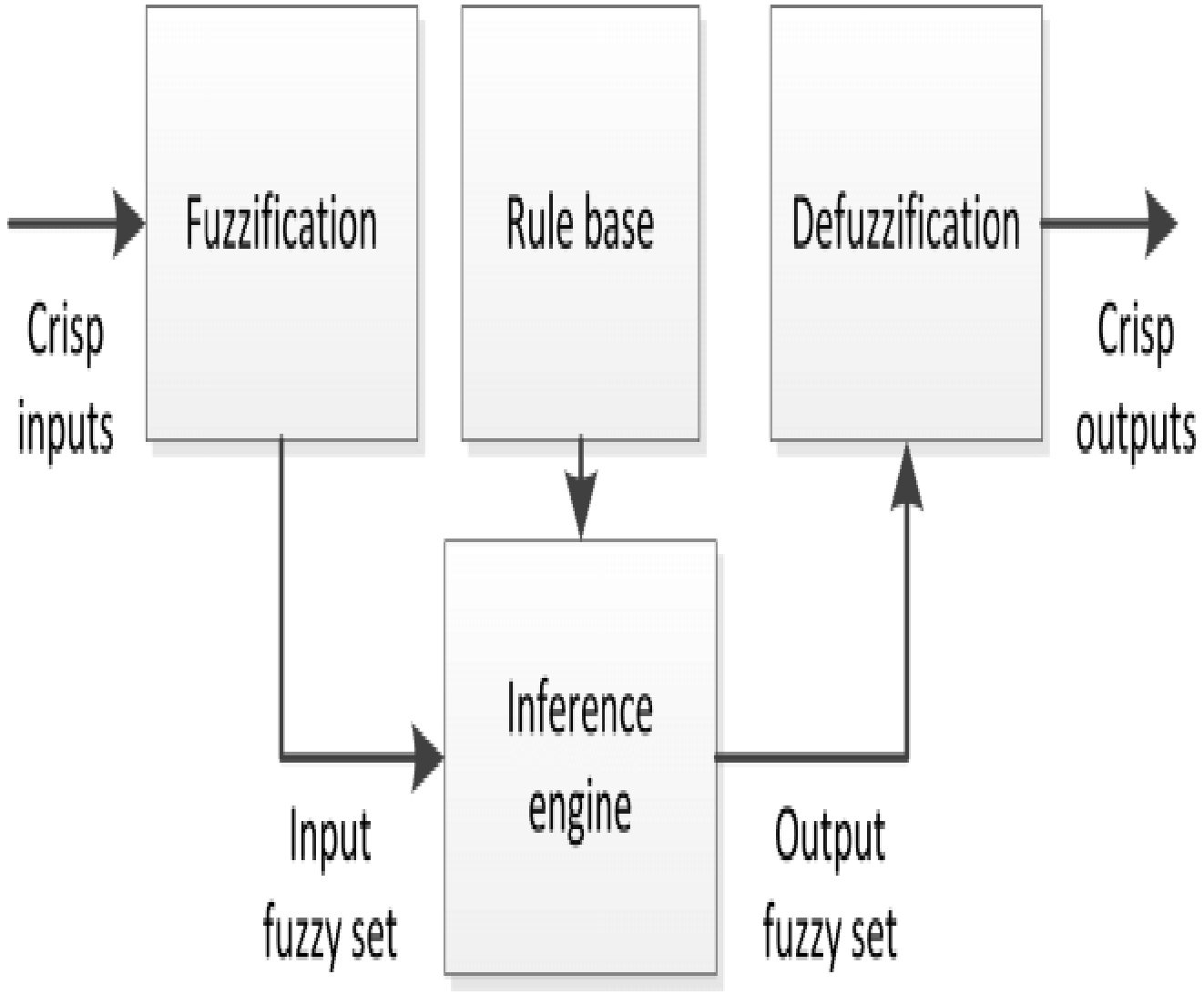
Matplotlib:-

Matplotlib tries to make easy things easy and hard things possible. We will generate plots, histograms, scatterplots, etc.,to make our project more appealing and easier to understand.

Fuzzy😊 (from skfuzzy import control as ctrl)

skfuzzy control subpackage, providing a high-level API for fuzzy system design. ... Consequent (output/control) variable for a fuzzy control system. Skfuzzy control. Control System ([rules]) Base class to contain a Fuzzy Control System.

MODULE OF WASHING MACHINE



Crisp input :- is a collection of unordered distinct elements, which are derived from Universal set. Universal set consists of all possible elements

Fuzzification:- Fuzzification is the process of converting a crisp input value to a fuzzy value that is performed by the use of the information in the knowledge base

RULES:-

If-Then Rules: Fuzzy sets depend on certain rules. The rule base is the most important requirement for the fuzzy logic. The rule base generally consists of various cases of If-Then rules. First the fuzzy sets and the membership functions are declared. Then the If-Then rules for the membership functions are decided for the particular control. The output is controlled by these rules on input. A typical If-Then rule consists of two parts. They are 1**) Antecedent** and 2) **Consequence** or Conclusion.

The 'If' statement is the Antecedent and the Then statement is the Consequence. If - (Antecedent) & Then - (Consequence). Examples: If the fan is slow, then increase the speed. If the temperature is high, then decrease the setting on an air conditioner.

Mamdani Method: Mamdani's methods of the Fuzzy interface is the most commonly used method. It was among the first control systems built using fuzzy set theory. It was first put forward by Ebrahim Mamdani as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. This inference method expects the output variable to be fuzzy sets. It is possible and also efficient to use a single spike in the output as membership function rather than a distributed fuzzy set. This is known as singleton output membership function. It enhances the Defuzzification process because it greatly simplifies the computation required by the more general Mamdani method which finds the centroid of the two dimensional function. But in the Sugeno type of inference can be used to model any inference system in which the output membership function is either linear or constant

Defuzzification:- It is used to transfer fuzzy inference results into a crisp output. In other words, defuzzification is realized by a decision-making algorithm that selects the best crisp value based on a fuzzy set.

SCREENSHOTS:-

1> FIRST APPROACH(with equation)

```
#Fuzzy controller for washing machine
#step 1 : identify i/p and o/p descriptors
#dirt           |   grease
#sd->small dirt  |   ng->no grease
#md->medium dirt |   mg->medium grease
#ld->large dirt  |   lg->large grease
#vs->very short time
#s->short time
#m->medium time
#l->long time
#vl->very long time
#-----------------------------------------------------
#step2 : membership function for i/p and o/p
#mf for dirt :-
#mf(sd)(x)=  (50-x)/50
#mf(md)(x)=  x/50        ,0<=x<=50
#           (100-x)/50  ,50<=x<=100
#mf(ld)(x)=  (x-50)/50   ,50<=x<=100
#mf for grease :-
#mf(ng)(y)=  (50-y)/50   ,0<=y<=50
#mf(mg)(y)=  y/50        ,0<=y<=50
#           (100-y)/50  ,50<=y<=100
#mf(lg)(y)=  (y-50)/50   ,50<=y<=100
#mf for wash time :-
#mf(vs)(z)=  (10-z)/10   ,0<=z<=10
```

#mf(s)(z)= z/10 ,0<=z<=10
 # (25-z)/15 ,10<=z<=25
#mf(m)(z)= (z-10)/15 ,10<=z<=25
 # (40-z)/15 ,25<=z<=40
#mf(l)(z)= (z-25)/15 ,25<=z<=40
 # (60-z)/15 ,40<=z<=60
#mf(vl)(z)= (z-40)/20 ,40<=z<=60
#----------

#step 3 : rule base

	ng	mg	lg
	0	1	2
0	sd	vs	m	l
1	md	s	m	l
2	ld	m	l	vs

```
def membership_functions(value,variable,descriptor):
    if(variable=='dirt'):
        if(descriptor=='sd' and value>=0 and value<=50):
            return (50-value)/50
        elif(descriptor=='md' and value>=0 and value<=50):
            return value/50
        elif(descriptor=='md' and value>50 and value<=100):
            return (100-value)/50
        elif(descriptor=='ld' and value>=50 and value<=100):
            return (value-50)/50
    elif(variable=='grease'):
        if(descriptor=='ng' and value>=0 and value<=50):
            return (50-value)/50
        elif(descriptor=='mg' and value>=0 and value<=50):
            return value/50
        elif(descriptor=='mg' and value>50 and value<=100):
            return (100-value)/50
        elif(descriptor=='lg' and value>=50 and value<=100):
            return (value-50)/50
```

```
def rule_base(dirt,grease):
    if(dirt=='sd'):
        dirt = 0
    elif(dirt=='md'):
        dirt = 1
    elif(dirt=='ld'):
        dirt = 2
    if(grease=='ng'):
        grease = 0
    elif(grease=='mg'):
        grease = 1
    elif(grease=='lg'):
        grease = 2
    rules = [['vs','m','l'],['s','m','l'],['m','l','vl']]
    return rules[dirt][grease]
```

```
#step 4 : rule evalutaion
def mapping_dirt(percentage):
    temp,temp1 = [],[]
    if(percentage>=0 and percentage<=50):
        # temp.append('sd')
        temp.append(membership_functions(percentage,'dirt','sd'))
        temp1.append('sd')
    if(percentage>=0 and percentage<=100):
        # temp.append('md')
        temp.append(membership_functions(percentage,'dirt','md'))
        temp1.append('md')
    if(percentage>=50 and percentage<=100):
        # temp.append('ld')
        temp.append(membership_functions(percentage,'dirt','ld'))
        temp1.append('ld')
    return temp,temp1

def mapping_grease(percentage):
    temp,temp1 = [],[]
    if(percentage>=0 and percentage<=50):
        temp.append(membership_functions(percentage,'grease','ng'))
        temp1.append('ng')
    if(percentage>=0 and percentage<=100):
        temp.append(membership_functions(percentage,'grease','mg'))
        temp1.append('mg')
    if(percentage>=50 and percentage<=100):
        temp.append(membership_functions(percentage,'grease','lg'))
        temp1.append('lg')
    return temp,temp1
```

```
def mapping_grease(percentage):
    temp,temp1 = [],[]
    if(percentage>=0 and percentage<=50):
        temp.append(membership_functions(percentage,'grease','ng'))
        temp1.append('ng')
    if(percentage>=0 and percentage<=100):
        temp.append(membership_functions(percentage,'grease','mg'))
        temp1.append('mg')
    if(percentage>=50 and percentage<=100):
        temp.append(membership_functions(percentage,'grease','lg'))
        temp1.append('lg')
    return temp,temp1
def mapping_washing_time(variable,value):
    temp = []
    if(variable=='vs'):
        temp.append((-1)*((value*10)-10))
    elif(variable=='s'):
        temp.append(value*10)
        temp.append((-1)*((value*15)-25))
    elif(variable=='m'):
        temp.append((value*15)+10)
        temp.append((-1)*((value*15)-40))
    elif(variable=='l'):
        temp.append((value*15)+25)
        temp.append((-1)*((value*20)-60))
    elif(variable=='vl'):
        temp.append((value*20)+40)
    return sum(temp)/len(temp)
```

```
    return sum(temp)/len(temp)
import itertools as it
m,n = int(input("Enter dirt percentage : ")),int(input("Enter grease percentage : "))
l1,l1_rule = mapping_dirt(m)
l2,l2_rule = mapping_grease(n)
a = list(map(min,list(it.product(l1,l2))))
b = list(it.product(l1_rule,l2_rule))
temp1,temp2 = list(map(str,b[a.index(max(a))]))
print(mapping_washing_time(rule_base(temp1,temp2),max(a)))
```

```
Enter dirt percentage : 45
Enter grease percentage : 56
25.0
```

2nd APPROACH (using inbuilt function)

```
from skfuzzy import control as ctrl
import skfuzzy as fuzz
import numpy as np

class washing_machine:

    degree_dirt = ctrl.Antecedent(np.arange(0, 101, 1), 'degree_dirt')
    type_dirt = ctrl.Antecedent(np.arange(0, 101, 1), 'type_dirt')
    wash_time = ctrl.Consequent(np.arange(0, 61, 1), 'wash_time')

    degree_names = ['Low', 'Medium', 'High']
    type_names = ['NonFat', 'Medium', 'Fat']

    #Outputing them into auto-membership functions
    degree_dirt.automf(names=degree_names)
    type_dirt.automf(names=type_names)

    # Washing Time Universe
    wash_time['very_short'] = fuzz.trimf(wash_time.universe, [0, 8, 12])
    wash_time['short'] = fuzz.trimf(wash_time.universe, [8, 12, 20])
    wash_time['medium'] = fuzz.trimf(wash_time.universe, [12, 20, 40])
    wash_time['long'] = fuzz.trimf(wash_time.universe, [20, 40, 60])
    wash_time['VeryLong'] = fuzz.trimf(wash_time.universe, [40, 60, 60])
```

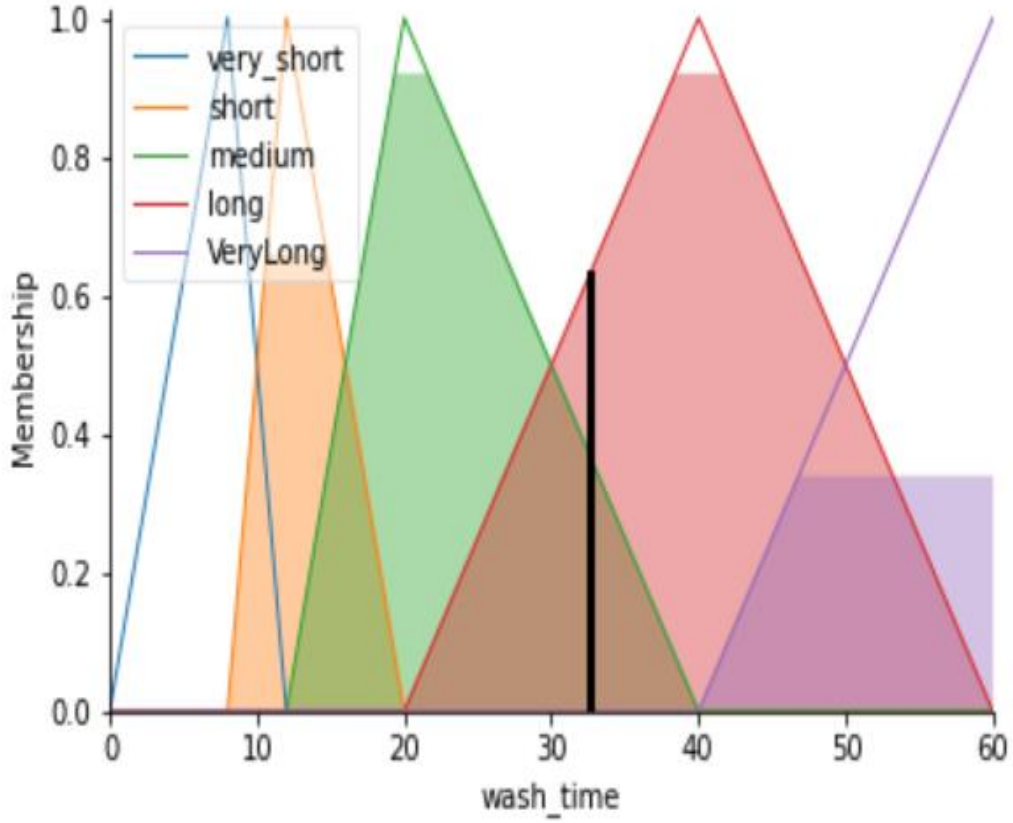
```
    # Rule Application
    rule1 = ctrl.Rule(degree_dirt['High'] | type_dirt['Fat'], wash_time['VeryLong'])
    rule2 = ctrl.Rule(degree_dirt['Medium'] | type_dirt['Fat'], wash_time['long'])
    rule3 = ctrl.Rule(degree_dirt['Low'] | type_dirt['Fat'], wash_time['long'])
    rule4 = ctrl.Rule(degree_dirt['High'] | type_dirt['Medium'], wash_time['long'])
    rule5 = ctrl.Rule(degree_dirt['Medium'] | type_dirt['Medium'], wash_time['medium'])
    rule6 = ctrl.Rule(degree_dirt['Low'] | type_dirt['Medium'], wash_time['medium'])
    rule7 = ctrl.Rule(degree_dirt['High'] | type_dirt['NonFat'], wash_time['medium'])
    rule8 = ctrl.Rule(degree_dirt['Medium'] | type_dirt['NonFat'], wash_time['short'])
    rule9 = ctrl.Rule(degree_dirt['Low'] | type_dirt['NonFat'], wash_time['very_short'])

    # Washing Control Simulation
    washing_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9])
    washing = ctrl.ControlSystemSimulation(washing_ctrl)

def fuzzify_laundry(fuzz_type,fuzz_degree):

    washing_machine.washing.input['type_dirt'] = fuzz_type
    washing_machine.washing.input['degree_dirt'] = fuzz_degree

    washing_machine.washing.compute()

    washing_machine.wash_time.view(sim=washing_machine.washing)

    return washing_machine.washing.output['wash_time']
```

```
def fuzzify_laundry(fuzz_type,fuzz_degree):

    washing_machine.washing.input['type_dirt'] = fuzz_type
    washing_machine.washing.input['degree_dirt'] = fuzz_degree

    washing_machine.washing.compute()

    washing_machine.wash_time.view(sim=washing_machine.washing)

    return washing_machine.washing.output['wash_time']

from fuzzify import *
#import fuzzify
def compute_washing_parameters(type_of_dirt,degree_of_dirt):

    if type_of_dirt < 0.0 or type_of_dirt > 100.0:
        raise Exception("Invalid Type of Dirtiness: %lf" %type_of_dirt)
    if degree_of_dirt < 0.0 or type_of_dirt > 100.0:
        raise Exception("Invalid Degree of Dirtiness: %lf" %degree_of_dirt)

    type_fuzzy = fuzzify_laundry(type_of_dirt,degree_of_dirt)

    return type_fuzzy
```

```
from fuzzify import *
#import fuzzify
def compute_washing_parameters(type_of_dirt,degree_of_dirt):

    if type_of_dirt < 0.0 or type_of_dirt > 100.0:
        raise Exception("Invalid Type of Dirtiness: %lf" %type_of_dirt)
    if degree_of_dirt < 0.0 or type_of_dirt > 100.0:
        raise Exception("Invalid Degree of Dirtiness: %lf" %degree_of_dirt)

    type_fuzzy = fuzzify_laundry(type_of_dirt,degree_of_dirt)

    return type_fuzzy



if __name__ == "__main__":
    type_of_dirt = float(input("Enter Type of Dirtiness [0-100]"))
    degree_of_dirt = float(input("Enter Degree of Dirtiness [0-100]"))
    washing_parameters = compute_washing_parameters(type_of_dirt,degree_of_dirt)
    print(washing_parameters)
```

```
Enter Type of Dirtiness [0-100]54
Enter Degree of Dirtiness [0-100]67
```

```
32.64403615569083
```



Uses of Fuzzy Logic:-

The Fuzzy rationale is utilized in different fields, for example, car frameworks, household products, condition control, and so forth. The basic applications of Fuzzy Rationale are:

- It is utilized for dynamic emotionally supportive networks and individual assessments in the huge organization business.
- It additionally controls the pH, drying, concoction refining process in the substance business.
- The fuzzy rationale is utilized in natural language handling and different serious applications in artificial intelligence.
- It is widely utilized in present-day control frameworks, for example, master frameworks.
- Fuzzy Logic impersonates how an individual would decide, just a lot quicker.

A lot of rules are then applied to the enrollment capacities to yield fresh yield esteem. How about we take a case of procedure control and comprehend fuzzy rationale.

Stage 1

Here, Temperature is the info and Fan Speed is the yield. You need to make a lot of participation capacities for each information. An enrollment work is basically a graphical portrayal of the fuzzy variable sets. We will at that point make an enrollment work for every one of three arrangements of temperature:

Stage 2

In the following stage, we will utilize three fuzzy sets for the yield, Slow, Medium, and Fast. A lot of capacities are made for each yield set similarly concerning the information sets.

Stage 3

Since we have our enrollment capacities characterized, we can make the standards that will characterize how the participation capacities will be applied to the last framework. We will make three standards for this framework:

- On the off chance that Hot, at that point Fast
- On the off chance that Warm, at that point Medium
- What's more, on the off chance that Cold, at that point Slow

These principles apply to the enrollment capacities to deliver the fresh yield an incentive to drive the framework. Along these lines, for an info estimation of 52 degrees, we converge the enrollment capacities. Here, we are applying two guidelines as the crossing point happens in the two capacities. You can stretch out the convergence focuses on the yield capacities to create a crossing point. You would then be able to shorten the yield capacities at the stature of the crossing focuses.

Conclusion

Fuzzy logic helps in understanding as to how people would perform in a dynamic environment in an unknown way. It thus acts like an Artificial Intelligence or AI that offers possibilities that could occur. Fuzzy logic helps control machines and offers a range of adequate thinking that could occur as part of the human decision-making process.

REFRENCES:-

To conduct this project the following tools have been used :

- Jupyter notebook and spyder
- **Numpy (Library)** : http://www.numpy.org/
- **Fuzzy (library)** : http://www.fuzzy.org/

1.1 Coursera:-

We have used this side for our basis knowledge gain of the methods that will be used in the project

https://www.coursera.org/learn/machine-learning-with-python

1.2 Stackoverflow:-

We have used this site for solving our different problems which has occurred during this project.

https://stackoverflow.com/questions/2620343/what-is-machine-learning

Youtube link

https://youtu.be/qdeA6OO04ZI