



# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



## Practical – 1

### Knowledge Representation (Prolog programming)

#### 1.1 Write a program in Prolog to implement simple facts and Queries

- **Software requirements:** GNU Prolog
- **Theory:**

A **fact** is a predicate expression that makes a declarative statement about the problem domain. Whenever a variable occurs in a Prolog expression, it is assumed to be **universally quantified**.

```
likes(john, susie). /* John likes Susie */
```

The Prolog interpreter responds to **queries** about the facts and rules represented in its database. The database is assumed to represent what is true about a particular problem domain. In making a query you are asking Prolog whether it can prove that your query is true. If so, it answers "yes" and displays any **variable bindings** that it made in coming up with the answer. If it fails to prove the query true, it answers "No".

Whenever you run the Prolog interpreter, it will **prompt** you with ?-.

- **Code:**

```
male(dhaval).  
male(roi).  
male(jenish).  
male(manish).  
male(jeet).
```

```
female(meena).  
female(henna).  
female(tina).  
female(sumbul).  
female(priti).
```

- **Output:**

```
compiling C:/GNU-Prolog/bin/pr1.pl for byte code...
C:/GNU-Prolog/bin/pr1.pl compiled, 10 lines read - 841 bytes written,
yes
| ?- male(dhaval).

yes
| ?- female(priti).

yes
| ?- male(jen).

no
| ?- male(X).

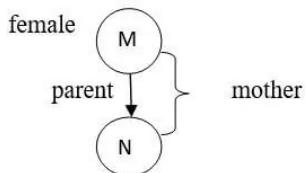
X = dhaval ? ;
X = roi ? ;
X = jenish ? ;
X = manish ? ;
X = jeet
(47 ms) yes
| ?- |
```

- **Conclusion:** We learned to implement simple program in Prolog.

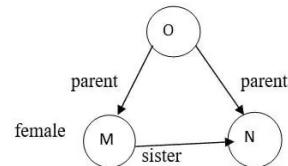
**1.2 Design a Family Tree for Your Family:** Write a program which contains three predicates: male, female, parent. Make rules for following family relations: father, mother, grandfather, grandmother, brother, sister, uncle, aunt, nephew and niece, cousin.

- Software requirements: GNU Prolog
- Theory:

The prolog syntax for mother-sister relationship is given as below:



Mother Relationship



Sister Relationship

mother(M,N): parent(M,N), female(M).

sister(M,N): parent(O,M), parent(O,N), female(M), M\!=\!=N.

We can also define:

- father(M,N): parent(M,N), female(M).
- haschild(M): parent(M,\_).
- brother(M,N): parent(O,M), parent(O,N), male(M), M\!=\!=N.

Where \_(underscore) indicates that it is an anonymous variable.

We can also write the syntax for some other relationships as follows:

- grandparent(M,N): parent(M,O), parent(O,N).
- grandmother(M,O): mother(M,N), parent(N,O).
- grandfather(M,O): father(M,N), parent(N,O).
- wife(M,N): parent(M,O), parent(N,O), female(M), male(N).
- uncle(M,O): brother(M,N), parent(N,O).

The program in prolog specifies the relationship between objects and the properties of objects; the family trees tell us how to construct a database of family.

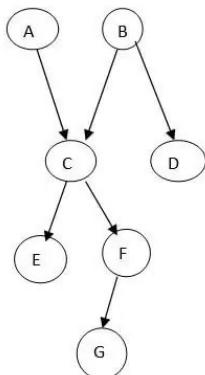
**Brother relationship:**

- If they both are male.
- If they have the same parent.

Suppose we have some clauses to illustrate the relationship:

- Let us see the working of the family tree by considering an example that can be formed from the prolog family tree.

The sample family tree is given below:



- **Code**

```

male(gumansinhji).
male(dharmendrasinh).
male(narendrasinh).
male(dhruvrajsinh).
male(pratipalsinh).
female(mohanba).
female(rajanba).
female(naynaba).
female(krupaba).
spouse(narendrasinh,naynaba).
parent(gumansinhji,dharmendrasinh).
parent(mohanba,dharmendrasinh).

parent(gumansinhji,narendrasinh).
parent(mohanba,narendrasinh).
  
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



parent(dharmendrasinh,dhruvrajsinh).  
parent(rajanba,dhruvrajsinh).

parent(dharmendrasinh,pratipalsinh).  
parent(rajanba,pratipalsinh).

parent(dharmendrasinh,krupaba).  
parent(rajanba,krupaba).  
father(X,Y) :- male(X),parent(X,Y).  
mother(X,Y) :- female(X),parent(X,Y).  
son(X,Y) :- male(X),parent(Y,X).  
daughter(X,Y) :- female(X),parent(X,Y).

grandfather(X,Y) :- male(X),parent(X,Somebody),parent(Somebody,Y).  
grandmother(X,Y) :- female(X),parent(X,Somebody),parent(Somebody,Y).

brother(X,Y) :- male(X),parent(Somebody,X),parent(Somebody,Y).  
sister(X,Y) :- female(X),parent(Somebody,X),parent(Somebody,Y).

uncle(X,Y) :- male(X),brother(X,Somebody),parent(Somebody,Y).  
aunty(X,Y) :- female(Y),uncle(Somebody,X),spouse(Somebody,Y).

• **Output**



# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



```
| ?-
father(X,Y).

X = gumansinhji
Y = dharmendrasinh ?
Action (: for next solution, a for all solutions, RET to stop) ?
Action (: for next solution, a for all solutions, RET to stop) ?
Action (: for next solution, a for all solutions, RET to stop) ?

yes
| ?- father(X,Y).

X = gumansinhji
Y = dharmendrasinh ? ;

X = gumansinhji
Y = narendrasinh ? ;

X = dharmendrasinh
Y = dhruvrajsinh ? ;

X = dharmendrasinh
Y = pratipalsinh ? ;

X = dharmendrasinh
Y = krupaba ? ;

(31 ms) no
| ?-
```



# Charotar University of Science and Technology

Devang Patel Institute of Advance Technology and Research

Department of Computer Engineering



File Edit Terminal Prolog Help

```
| ?- mother(X,Y).  
  
X = mohanba  
Y = dharmendrasinh ? ;  
  
X = mohanba  
Y = narendrasinh ? ;  
  
X = rajanba  
Y = dhruvrajsinh ? ;  
  
X = rajanba  
Y = pratipalsinh ? ;  
  
X = rajanba  
Y = krupaba ? ;  
  
no  
| ?- grandfather(X,Y).  
  
X = gumansinhji  
Y = dhruvrajsinh ? ;  
  
X = gumansinhji  
Y = pratipalsinh ? ;  
  
X = gumansinhji  
Y = krupaba ? ;  
  
no  
| ?-
```

- **Conclusion:** We learned to form functions for relations and ‘and’ & ‘or’ operations.



### **1.3Aim: Design a Medical Diagnosis Expert System**

**Software requirements:** GNU Prolog

**Theory:** Make function for every type of yes and no to convert into y and n.

Then make function ask for cold, cough and fever accordingly give results.

**Code:** answer(Inp,X):-((Inp='Yes',X='y');(Inp='yes',X='y');(Inp='y',X='y');(Inp='Y',X='y');(Inp='YES',X='y'));((Inp='NO',X='n');(Inp='no',X='n');(Inp='N',X='n');(Inp='n',X='n')).

```
mainask:-write('Write your
name :'),read(Name),cold(C),cough(F),fever(V),fun(C,F,V,Ans),write(Name),write(Ans).
```

```
fever(V):-write('Fever ? '),read(Inp),answer(Inp,V).
```

```
cough(F):-write('Cough ? '),read(Inp),answer(Inp,F).
```

```
cold(C):-write('Cold ? '),read(Inp),answer(Inp,C).
```

```
fun(C,F,V,Ans):-(C='y',F='y',V='y',Ans=' has viral few.')(C='n',F='n',V='y',Ans=' has viral
fever.');
```

```
(C='y',F='n',V='n',Ans=' has normal cold.')(C='n',F='n',V='n',Ans=' is alright!');
```

```
(C='n',F='y',V='n',Ans=' has normal cough.')(C='y',F='y',V='n',Ans=' has normal cold and
cough.');
```

**Output:**

```
uncaught exception: error(existence_error(proced
| ?- mainask.
Write your name :dhruv
.
Cold ? Y
.
Cough ? N.
Fever ? N.
dhruv has viral few.

true ?

yes
| ?-
```



**Conclusion:** We learned to perform arithmetic operations on prolog, found sum of 1 to n integers and applied recursion.

#### **1.4 Write a Prolog program to demonstrate arithmetic operations and find addition of 1 to N numbers. Also demonstrate examples of recursion.**

- **Software requirements:** GNU Prolog
- **Theory:**

In arithmetic expression, any variables must already be bound. The value of these variables must be numerical. The value of arithmetic expression bounds the variable of first argument. If it is not, an error message will be generated as result.

In arithmetic expression, + - \* / symbols are special type of infix operator, and these operators are also known as arithmetic operators. In Prolog, operators are used as predicates but here operators are functions and these operators return a numerical value.

Arithmetic expressions can include variables, numbers, operators, and arithmetic functions. These will be written in parentheses with their arguments. These will return numerical values just like the arithmetic operators.

The minus(-) arithmetic operator is used as a binary infix operator, which is used to describe the difference of two numerical values like A - 2. It is also used as a unary prefix operator, which is used to describe the negative of a numerical value.

Looping is used to enable a set of instructions to be repeatedly executed either a fixed number of times or until a given condition met. Prolog has no looping facility, but we can obtain a similar effect. Using this effect, we can evaluate a sequence of goals repeatedly.

- **Code:**

```
sumto(1, 1).
sumto(N, M) :- N>1, N1 is N-1, sumto(N1, M1), M is M1+N.

loop(0).
loop(N) :- N>0, write('value of N is: '), write(N), nl,
S is N-1, loop(S).
```



# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



- **Output:**

```
compiling C:/Users/dell/Desktop/SEM 6/AI/P1/p1(4).pl for byte code..
C:/Users/dell/Desktop/SEM 6/AI/P1/p1(4).pl compiled, 5 lines read -
| ?- A is 5.7 + 2.9 * 3.

A = 14.39999999999999

yes
| ?- B is sqrt(25).

B = 5.0

yes
| ?- A is 7, B is -A - 3.

A = 7
B = -10

yes
| ?- 15 is 9 + 6 - 13 + 20
.

no
```

```
| ?- sumto(50,N).
N = 1275 ?

yes
| ?- loop(5).
value of N is: 5
value of N is: 4
value of N is: 3
value of N is: 2
value of N is: 1

true ? |
```

- **Conclusion:** We learned to perform arithmetic operations on prolog, found sum of 1 to n integers and applied recursion.

### 1.5 Write a program to display Fibonacci series in prolog.

- **Software requirements:** GNU Prolog
- **Theory:**

The **Fibonacci sequence**, in which each number is the sum of the two preceding ones. The sequence commonly starts from 0 and 1.

The Fibonacci number  $I$  is defined as the sum of the Fibonacci numbers for  $I-1$  and  $I-2$ , while the Fibonacci number of 0 and 1 are both defined to be 1.

The Fibonacci numbers may be defined by the recurrence relation

$F_0=0$  and  $F_1=1$ ,  $F=F_{-1}+F_{-2}$ ;

for  $n > 1$ .

- **Code:**

fab1(1,1).

fab1(2,1).

fab1(N,T):-N>2,N1 is N-1,N2 is N-2,fab1(N1,T1),fab1(N2,T2),T is (T1+T2).

- **Output:**

```

yes
| ?- fab1(4,T).

T = 3 ?

yes
| ?-

```

- **Conclusion:** We compiled the Fibonacci series.

## 1.6 Write a program to find factorial of a number in prolog using recursion.

- **Software requirements:** GNU Prolog
- **Theory:**

Prolog Factorial is the product of an integer and the other integers below the given number i.e., 5! is represented as 5\*4\*3\*2\*1 which is equal to 120.

Here in Prolog factorial,

- Prolog Factorial function definition is also similar to a normal factorial function.
- Factorial(0,1) i.e., factorial of 0 is generally 1.
- Factorial(N,M), if any temporary value N1 is assigned to N-1.
- Factorial(N1,M1), and is factorial of N1 is M1.
- M is NM1 i.e., assigning M to N\*M1, then value of N is M.

The above happens to be the recursive relation between N and factorial M. It reviews rules for particular relation in the top to bottom order.

- **Code:**

```
fact(0,1).
fact(N,ANS):- N>0,
N1 is N-1,
fact(N1,A),
ANS is A*N.
```

- **Output:**

```
compiling C:/Users/dell/Desktop/SEM 6/AI/P1/p1(6).PL for byte code..
C:/Users/dell/Desktop/SEM 6/AI/P1/p1(6).PL compiled, 4 lines read - :
| ?- fact(5,ANS).

ANS = 120 ?
```

- **Conclusion:** We implemented factorial.

### 1.7 Write predicates one converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.

- **Software requirements:** GNU Prolog
- **Theory:**

Here are the formulae for converting between degrees Fahrenheit and degrees Celsius:

$$\text{DegreesF} = (9 / 5 * \text{DegreesC}) + 32$$

$$\text{DegreesC} = 5 / 9 * (\text{DegreesF} - 32)$$

convert(C, F) that first checks which temperature (C or F) is given, and then does the appropriate conversion. That is, when C is given, convert(C, F) uses c2f(C, F).

- **Code**

```
convert(F,C) :- C is (F-32)*(5/9).
c_to_f(C,F) :- F is C*(9/5) + 32.
frrezinf(F) :- F=<32.
```

- **Output**



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
(10 mo) no
| ?- [pr3].
compiling C:/GNU-Prolog/bin/pr3.pl for byte code...
C:/GNU-Prolog/bin/pr3.pl compiled, 2 lines read - 1019 byte

yes
| ?- convert(32,C).

C = 0.0

yes
| ?- convert(33,C).

C = 0.5555555555555558

yes
| ?- convert(33,C).

C = 0.5555555555555558

yes
| ?- convert(36,C).

C = 2.222222222222223

yes
| ?- |
```

- **Conclusion:** We converted degree of temperatures into different units using prolog.

**1.8 Write a program in prolog to implement phone list which store name, phone number and birthdays of friends and family members. Write a query to get birthday a list of people whose birthdays are in the current month.**

- **Software requirements:** GNU Prolog
- **Theory:**

We first take the data in form of fact.

Then we create a query for finding the appropriate data with condition.

A single underscore ( \_ ) denotes an anonymous variable and means "any term".

Unlike other variables, the underscore does not represent the same value everywhere it occurs within a predicate definition.

We then define needed variable and get the answer.

- **Code**

```
phone_list(person(dhruvrajsinh,gohil),'123456789',bdate(day(5),month(12),year(2002))
)).
phone_list(person(ravi,mishra),'123456789',bdate(day(5),month(11),year(2002))).
phone_list(person(shubh,patel),'123456798',bdate(day(4),month(10),year(2001))).
phone_list(person(rajsinh,gohil),'123475689',bdate(day(6),month(2),year(2002))).
phone_list(person(dhruvraj,gohil),'123457789',bdate(day(7),month(1),year(2003))).
phone_list(person(dhruv,gohil),'133456789',bdate(day(2),month(3),year(2001))).
```

- **Output:**

```
| ?- phone_list(person(shubh,palte).
B = '123456798'
X = 2001
Y = 4 ?-
yes
| ?-
```

- **Conclusion:** We implemented phone list query.

## 1.9 Implement Water Jug Problem in Prolog

**Variation1:A Water Jug Problem:** You are given two jugs, a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug, and the ground on which water may be poured. Neither jug has any measuring markings on it. How can you get exactly 2 gallons of water in the 4-gallon jug?

**Variation2:** You have to defuse a bomb by placing exactly 4 gallons (15 L) of water on a sensor. The problem is, you only have a 5-gallon (18.9 L) jug and



a 3 gallons (11 L) jug on hand! Neither jug has any measuring markings on it.

**Variation3:** A milkman carries a full 12-liter container of milk. He needs to deliver exactly liters. However, the customer only has 8 and 5-liter jugs. Neither jug has any measuring markings on it.

**Code:**

**Variant-1:**

```
from queue import Queue
```

```
def water_jug_2_gallons():
```

```
    visited = set()
```

```
    q = Queue()
```

```
    q.put((0, 0))
```

```
    visited.add((0, 0))
```

```
    while not q.empty():
```

```
        jug4, jug3 = q.get()
```

```
        if jug4 == 2:
```

```
            return True
```

```
        # fill the 4-gallon jug
```

```
        if (4, jug3) not in visited:
```

```
            q.put((4, jug3))
```

```
            visited.add((4, jug3))
```

```
        # fill the 3-gallon jug
```

```
        if (jug4, 3) not in visited:
```

```
            q.put((jug4, 3))
```

```
            visited.add((jug4, 3))
```



**Charotar University of Science  
and Technology**

**Devang Patel Institute of Advance  
Technology and Research**

**Department of Computer Engineering**



```
# empty the 4-gallon jug
if (0, jug3) not in visited:
    q.put((0, jug3))
    visited.add((0, jug3))

# empty the 3-gallon jug
if (jug4, 0) not in visited:
    q.put((jug4, 0))
    visited.add((jug4, 0))

# pour water from 4-gallon to 3-gallon jug
if jug4 > 0 and jug3 < 3:
    amount = min(jug4, 3 - jug3)
    if (jug4 - amount, jug3 + amount) not in visited:
        q.put((jug4 - amount, jug3 + amount))
        visited.add((jug4 - amount, jug3 + amount))

# pour water from 3-gallon to 4-gallon jug
if jug3 > 0 and jug4 < 4:
    amount = min(jug3, 4 - jug4)
    if (jug4 + amount, jug3 - amount) not in visited:
        q.put((jug4 + amount, jug3 - amount))
        visited.add((jug4 + amount, jug3 - amount))

return False
```

**Varient-2:**

**CODE:**

```
From queue import Queue
def defuse_bomb():
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
visited = set()
q = Queue()
q.put((0, 0))
visited.add((0, 0))
while not q.empty():
    jug5, jug3 = q.get()
    if jug5 == 4:
        return True
    # fill the 5-gallon jug
    if (5, jug3) not in visited:
        q.put((5, jug3))
        visited.add((5, jug3))
    # fill the 3-gallon jug
    if (jug5, 3) not in visited:
        q.put((jug5, 3))
        visited.add((jug5, 3))
    # empty the 5-gallon jug
    if (0, jug3) not in visited:
        q.put((0, jug3))
        visited.add((0, jug3))
    # empty the 3-gallon jug
    if (jug5, 0) not in visited:
        q.put((jug5, 0))
        visited.add((jug5, 0))
    # pour water from 5-gallon to 3-gallon jug
    if jug5 > 0 and jug3 < 3:
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
amount = min(jug5, 3 - jug3)
if (jug5 - amount, jug3 + amount) not in visited:
    q.put((jug5 - amount, jug3 + amount))
    visited.add((jug5 - amount, jug3 + amount))

# pour water from 3-gallon to 5-gallon jug
if jug3 > 0 and jug5 < 5:
    amount = min(jug3, 5 - jug5)
    if (jug5 + amount, jug3 - amount) not in visited:
        q.put((jug5 + amount, jug3 - amount))
        visited.add((jug5 + amount, jug3 - amount))

return False
```

**Variant 3:**

**CODE:**

```
from queue import Queue
```

```
def deliver_milk(target):
    visited = set()
    q = Queue()
    q.put((0, 0, 12))
    visited.add((0, 0, 12))
    while not q.empty():
        jug8, jug5, jug12 = q.get()
        if jug8 == target:
            return True
        # fill the 8-liter jug
```



**Charotar University of Science  
and Technology**

**Devang Patel Institute of Advance  
Technology and Research**

**Department of Computer Engineering**



```
if (8, jug5, jug12) not in visited:  
    q.put((8, jug5, jug12))  
    visited.add((8, jug5, jug12))  
  
    # fill the 5-liter jug  
  
    if (jug8, 5, jug12) not in visited:  
        q.put((jug8, 5, jug12))  
        visited.add((jug8, 5, jug12))  
  
    # empty the 8-liter jug  
  
    if (0, jug5, jug12) not in visited:  
        q.put((0, jug5, jug12))  
        visited.add((0, jug5, jug12))  
  
    # empty the 5-liter jug  
  
    if (jug8, 0, jug12) not in visited:  
        q.put((jug8, 0, jug12))  
        visited.add((jug8, 0, jug12))  
  
    # pour water from the 8-liter jug to the 5-liter jug  
  
    if jug8 > 0:  
        space_in_jug5 = 5 - jug5  
        amount_to_pour = min(jug8, space_in_jug5)  
        if (jug8 - amount_to_pour, jug5 + amount_to_pour, jug12) not in visited:  
            q.put((jug8 - amount_to_pour, jug5 + amount_to_pour, jug12))  
            visited.add((jug8 - amount_to_pour, jug5 + amount_to_pour, jug12))  
  
        # pour water from the 5-liter jug to the 8-liter jug  
  
        if jug5 > 0:  
            space_in_jug8 = 8 - jug8  
            amount_to_pour = min(jug5, space_in_jug8)
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
if (jug8 + amount_to_pour, jug5 - amount_to_pour, jug12) not in visited:  
    q.put((jug8 + amount_to_pour, jug5 - amount_to_pour, jug12))  
    visited.add((jug8 + amount_to_pour, jug5 - amount_to_pour, jug12))  
return False
```

### 1.10 Write a program to perform following operations on lists in prolog

- ✓ Create a List
- ✓ Write in List
- ✓ Check Membership
- ✓ Length of a List
- ✓ Reverse a List
- ✓ Concatenation
- ✓ Add an item
- ✓ Delete an item
- ✓ Sub list
- ✓ Permutations
- ✓ Append list

#### Finding nth element

- Software requirements: GNU Prolog
- Theory:

In Prolog, lists are inbuilt data structures. Lists can be used to represent sets, stacks, queues, linked lists, and several complex data structures such as trees, graphs, etc.

#### Basic Notation and Properties of Lists:

- A list in Prolog is an ordered collection of items denoted as [i<sub>1</sub>, i<sub>2</sub>, ..., i<sub>n</sub>].
- Unlike arrays in other programming languages where we can directly access any element of the array, prolog lists allow direct access of the first element only which is denoted as Head. Therefore we can write a prolog list as: [Head | Rest], where Rest is the rest of the list excluding the first element Head.
- Prolog lists allow nonhomogeneous data types of list items.
- Nested lists of arbitrary depths are also allowed in prolog.

A list can be either **empty** or **non-empty**. In the first case, the list is simply written as a Prolog atom, []. In the second case, the list consists of two things as given below –

- The first item, called the **head** of the list;
- The remaining part of the list, called the **tail**.

Suppose we have a list like: [red, green, blue, white, dark]. Here the head is red and tail is [green, blue, white, dark]. So the tail is another list.

### Basic Operations on Lists

Following table contains various operations on prolog lists –

Operations	Definition
Membership Checking	During this operation, we can verify whether a given element is member of specified list or not?
Length Calculation	With this operation, we can find the length of a list.
Concatenation	Concatenation is an operation which is used to join/add two lists.
Delete Items	This operation removes the specified element from a list.
Append Items	Append operation adds one list into another (as an item).
Insert Items	This operation inserts a given item into a list.

- **Code:**

```
list_mem(X,[X|_]).
```

```
list_mem(X,[_|TAIL]) :- list_mem(X,TAIL).
```

```
list_len([],0).
```

```
list_len([_|TAIL],N) :- list_len(TAIL,N1), N is N1 + 1.
```

```
list_con([],L,L).
```

```
list_con([X1|L1],L2,[X1|L3]) :- list_con(L1,L2,L3).
```

```
list_remove(X, [X], []).
```

```
list_remove(X,[X|L1], L1).
```

```
list_remove(X, [Y|L2], [Y|L1]) :- list_remove(X,L2,L1).
```

```
list_append(A,T,T) :- list_mem(A,T),!.
```

```
list_append(A,T,[T|A]).
```



# Charotar University of Science and Technology

Devang Patel Institute of Advance Technology and Research

Department of Computer Engineering



- Output:

```
|compiling C:/GNU-Prolog/bin/p1(10).pl for byte code...
C:/GNU-Prolog/bin/p1(10).pl compiled, 1 lines read - 426 bytes written, 12 ms
| ?- list_mem(b,[a,b,c]).  
true ?  
  
yes  
| ?- list_mem(b,[a,[b,c]]).  
  
no  
| ?- list_mem([b,c],[a,[b,c]]).  
true ?  
  
yes  
| ?- list_mem(d,[a,b,c]).  
  
no  
| ?- |
```

```
|compiling C:/GNU-Prolog/bin/p1(10).pl for byte code...
C:/GNU-Prolog/bin/p1(10).pl compiled, 6 lines read - 929 bytes written, 11 ms
| ?- list_len([],Len).  
Len = 0  
  
yes  
| ?- list_len([a,b,c,d,e,f,g,h,i,j],Len).  
Len = 10  
  
yes  
| ?- list_len([[a,b],[c,d],[e,f]],Len).  
Len = 3  
  
yes  
| ?- |
```



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



```
compiling C:/GNU-Prolog/bin/p1(10).pl for byte code...
C:/GNU-Prolog/bin/p1(10).pl compiled, 11 lines read - 1825 bytes written, 12 ms
| ?- list_rev([a,b,c,d,e],NewList).
```

```
NewList = [e,d,c,b,a]
```

```
yes
| ?- list_rev([a,b,c,d,e],[e,d,c,b,a]).
```

```
yes
| ?- list_rev([a,b,c,d,e],[e,d,c,b,x]).
```

```
no
| ?- |
```

```
NewList = [1,2,a,b,c]
```

```
yes
| ?- list_con([], [a,b,c], NewList).
```

```
NewList = [a,b,c]
```

```
yes
| ?- list_concat([[1,2,3],[p,q,r]],[a,b,c],NewList).
uncaught exception: error(existence_error(procedure,list_concat/3),top_level/0)
| ?- list_con([[1,2,3],[p,q,r]],[a,b,c],NewList).
```

```
NewList = [[1,2,3],[p,q,r],a,b,c]
```

```
yes
| ?- |
```

```
NewList = [a,e,i,o,u]
```

```
yes
| ?- list_append(e, [e,i,o,u], NewList).
```

```
NewList = [e,i,o,u]
```

```
yes
| ?- list_append([a,b], [e,i,o,u], NewList).
```

```
NewList = [[a,b],e,i,o,u]
```

```
yes
| ?- |
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
| ?- list_delete(a,[a,e,i,o,u],NewList).  
NewList = [e,i,o,u] ?  
yes  
| ?- list_delete(a,[a],NewList).  
NewList = [] ?  
yes  
| ?- list_delete(X,[a,e,i,o,u],[a,e,o,u]).  
X = i ? ;  
no  
| ?-
```

```
| ?- list_insert(a,[e,i,o,u],NewList).  
NewList = [a,e,i,o,u] ? a  
NewList = [e,a,i,o,u]  
NewList = [e,i,a,o,u]  
NewList = [e,i,o,a,u]  
NewList = [e,i,o,u,a]  
NewList = [[e,i,o,u,a]]  
(15 ms) no  
| ?-
```

```
| ?- list_perm([a,b,c,d],X).
X = [a,b,c,d] ? a
X = [a,b,c,d]
X = [a,b,d,c]
X = [a,b,d,c]
X = [a,b,d,c]
X = [a,b,d,c]
X = [a,c,b,d]
X = [a,c,b,d]
X = [a,c,d,b]|
X = [a,c,d,b]
X = [a,c,d,b]
X = [a,c,d,b]
X = [a,d,b,c]
```

```
X = [d,b,c,a]
X = [d,c,a,b]
X = [d,c,a,b]
X = [d,c,b,a]
X = [d,c,b,a]
X = [d,c,b,a]
X = [d,c,b,a]
X = [d,c,a,b]
X = [d,c,a,b]
X = [d,c,b,a]
X = [d,c,b,a]
X = [d,c,b,a]
X = [d,c,b,a]
(93 ms) no
```

- **Conclusion:** We implemented various operations.



# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



## 1.11 Write a program to demonstrate cut and fail in prolog.

- Software requirements: GNU Prolog
- Theory:

When one statement is true, another one must be false. In such cases we can use the cut.

We can also define a predicate where we use the two cases using disjunction (OR logic). So when first one satisfies, it does not check for the second one, otherwise, it will check for the second statement.

The fail predicate simply fails the rule.

The fail forces backtracking in an attempt to unify with another clause.

- **Code:**

```
max(X,Y,X) :- X >= Y, !.  
max(X,Y,Y) :- X < Y.  
max_find(X,Y,Max) :- X>=Y,! , Max = X; Max = Y.  
a(X):- b(X),c(X),fail.  
a(X):-d(X).  
b(1).  
b(4).  
c(1).  
c(3).  
d(4).
```

- **Output:**

```
C:/GNU-Prolog/bin/p1(11).pl compiled, 2 lines read - 1169 bytes written, 8 ms  
| ?- max(10,20,Max).  
Max = 20  
yes  
| ?- max(20,10,Max).  
Max = 20  
(16 ms) yes  
| ?- |
```



# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



```
|compiling C:/Users/dell/Desktop/SEM 6/AI/P1/p1(11).pl for byte code...
C:/Users/dell/Desktop/SEM 6/AI/P1/p1(11).pl compiled, 11 lines read -
| ?- a(X).
X = 4
yes
| ?- |
```

- **Conclusion:** We learned cut and fail function.

## 1.12 Write a program to solve Monkey Banana problem in Prolog

- Software requirements: GNU Prolog
- Theory:

Suppose the problem is as given below –

- A hungry monkey is in a room, and he is near the door.
- The monkey is on the floor.
- Bananas have been hung from the center of the ceiling of the room.
- There is a block (or chair) present in the room near the window.
- The monkey wants the banana, but cannot reach it.

So if the monkey is clever enough, he can come to the block, drag the block to the center, climb on it, and get the banana. Below are few observations in this case –

- Monkey can reach the block, if both of them are at the same level.
- If the block position is not at the center, then monkey can drag it to the center.
- If monkey and the block both are on the floor, and block is at the center, then the monkey can climb up on the block. So the vertical position of the monkey will be changed.
- When the monkey is on the block, and block is at the center, then the monkey can get the bananas.

We will create some predicates:

We have some predicates that will move from one state to another state, by performing action.

- When the block is at the middle, and monkey is on top of the block, and monkey does not have the banana (i.e. **has not** state), then using the **grasp** action, it will change from **has not** state to **have** state.
- From the floor, it can move to the top of the block (i.e. **on top** state), by performing the action **climb**.
- The **push** or **drag** operation moves the block from one place to another.
- Monkey can move from one place to another using **walk** or **move** clauses.

Another predicate will be canget(). Here we pass a state, so this will perform move predicate from one state to another using different actions, then perform canget() on state 2. When we have reached to the state '**has>**', this indicates '**has banana**'. We will stop the execution.

- **Code:**

```
move(state(middle,onbox,middle,hasnot), grasp, state(middle,onbox,middle,has)).  
move(state(P,onfloor,P,H), climb, state(P,onbox,P,H)).  
move(state(P1,onfloor,P1,H), drag(P1,P2), state(P2,onfloor,P2,H)).  
move(state(P1,onfloor,B,H), walk(P1,P2), state(P2,onfloor,B,H)).  
canget(state(_,_,_,has)).  
canget(State1) :- move(State1,_,State2), canget(State2).
```

- **Output:**

```
| ?- canget(state(atdoor, onfloor, atwindow, hasnot)).  
true ?  
yes  
| ?- trace  
. The debugger will first creep -- showing everything (trace)  
yes  
{trace}  
| ?- canget(state(atdoor, onfloor, atwindow, hasnot)).  
1 1 Call: canget(state(atdoor, onfloor, atwindow, hasnot)) ?  
2 2 Call: move(state(atdoor, onfloor, atwindow, hasnot), _71, _111) ?  
2 2 Exit: move(state(atdoor, onfloor, atwindow, hasnot), walk(atdoor, _99), state(_99, onfloor, atwindow, hasnot)) ?  
3 2 Call: canget(state(_99, onfloor, atwindow, hasnot)) ?  
4 3 Call: move(state(_99, onfloor, atwindow, hasnot), _129, _169) ?  
4 3 Exit: move(state(atwindow, onfloor, atwindow, hasnot), climb, state(atwindow, onbox, atwindow, hasnot)) ?  
5 3 Call: canget(state(atwindow, onbox, atwindow, hasnot)) ?  
6 4 Call: move(state(atwindow, onbox, atwindow, hasnot), _184, _224) ?  
6 4 Fail: move(state(atwindow, onbox, atwindow, hasnot), _184, _212) ?  
5 3 Fail: canget(state(atwindow, onbox, atwindow, hasnot)) ?  
4 3 Redo: move(state(atwindow, onfloor, atwindow, hasnot), climb, state(atwindow, onbox, atwindow, hasnot)) ?  
4 3 Exit: move(state(atwindow, onfloor, atwindow, hasnot), drag(atwindow, _157), state(_157, onfloor, _157, hasnot)) ?  
5 3 Call: canget(state(_157, onfloor, _157, hasnot)) ?  
6 4 Call: move(state(_157, onfloor, _157, hasnot), _187, _227) ?  
6 4 Exit: move(state(_157, onfloor, _157, hasnot), climb, state(_157, onbox, _157, hasnot)) ?  
7 4 Call: canget(state(_157, onbox, _157, hasnot)) ?  
8 5 Call: move(state(_157, onbox, _157, hasnot), _242, _282) ?  
8 5 Exit: move(state(middle, onbox, middle, hasnot), grasp, state(middle, onbox, middle, has)) ?  
9 5 Call: canget(state(middle, onbox, middle, has)) ?  
9 5 Exit: canget(state(middle, onbox, middle, has)) ?  
7 4 Exit: canget(state(middle, onbox, middle, hasnot)) ?  
5 3 Exit: canget(state(middle, onfloor, middle, hasnot)) ?  
3 2 Exit: canget(state(atwindow, onfloor, atwindow, hasnot)) ?  
1 1 Exit: canget(state(atdoor, onfloor, atwindow, hasnot)) ?
```

- **Conclusion:** We solved the monkey banana problem with prolog.



## **Charotar University of Science and Technology**

**Devang Patel Institute of Advance  
Technology and Research**

**Department of Computer Engineering**



**1.13:Implement Missionaries and Cannibals Problem Solution in Prolog**  
Missionaries and Cannibals is a problem in which 3 missionaries and 3 cannibals want to cross from the left bank of a river to the right bank of the river. There is a boat on the left bank, but it only carries at most two people at a time (and can never cross with zero people). If cannibals ever outnumber missionaries on either bank, the cannibals will eat the missionaries.

**Code:**



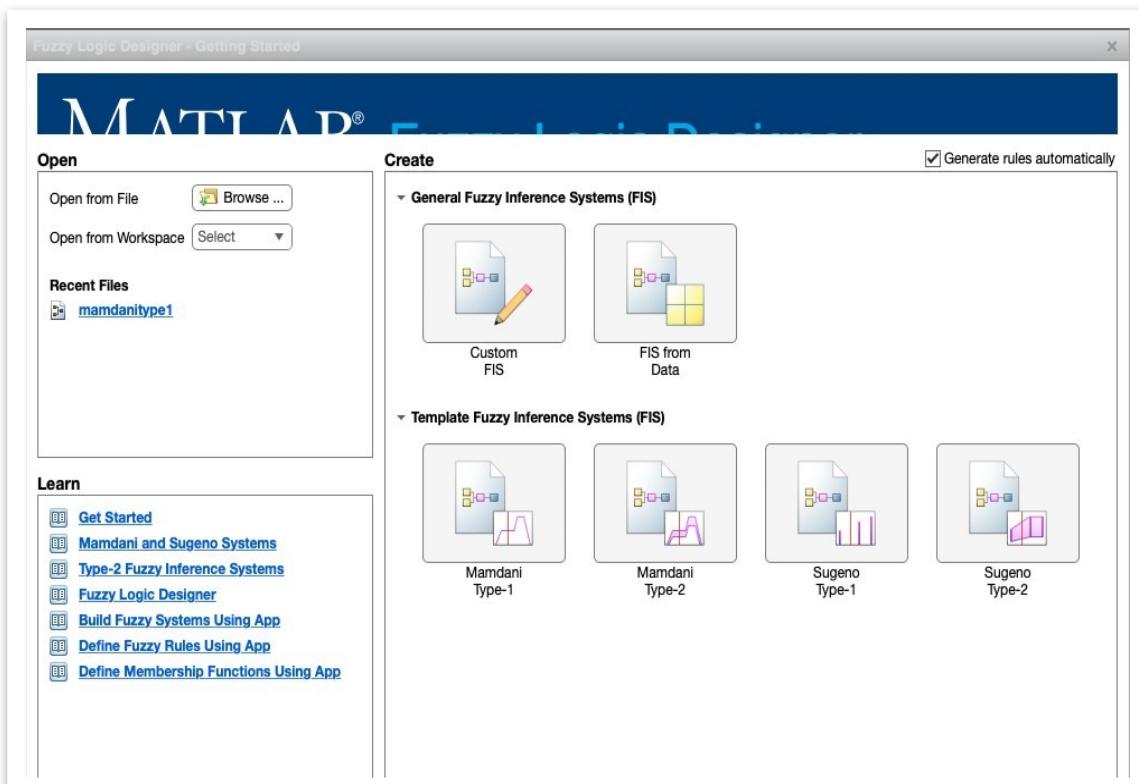
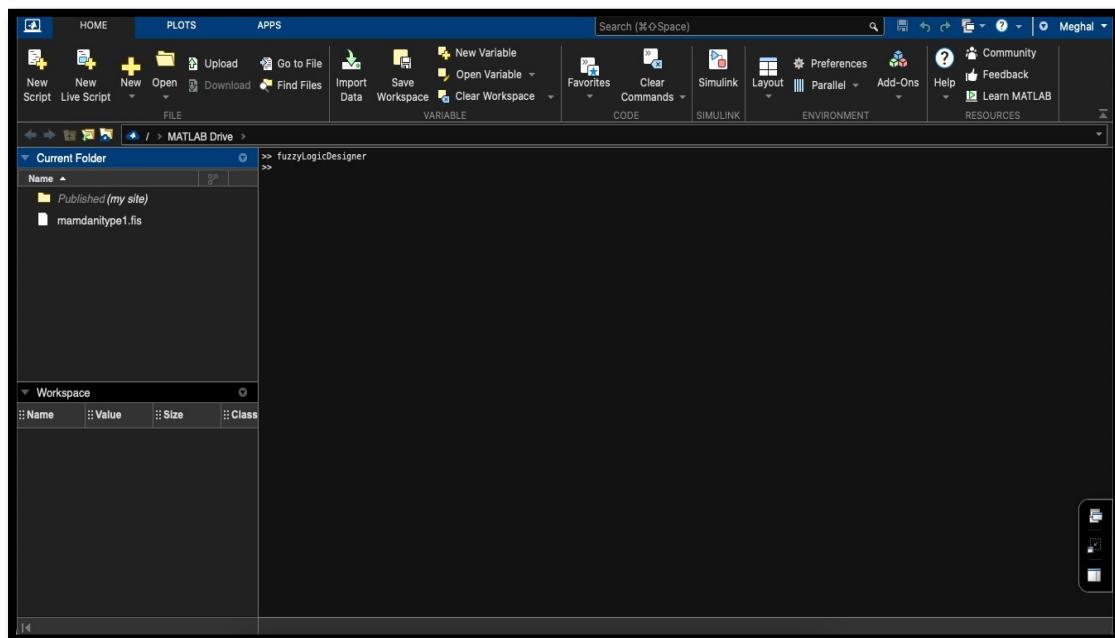
**2 Aim:** Design a controller to determine wash time of a domestic washing machine. Assume the input is dirt and grease on clothes. Use three descriptors for input variables and five descriptors for output variables. Derive the set of rules for controller action and defuzzification. The design should be supported by the figure wherever possible. Show that if the clothes are solid to a larger degree the wash time will be more and vice versa.

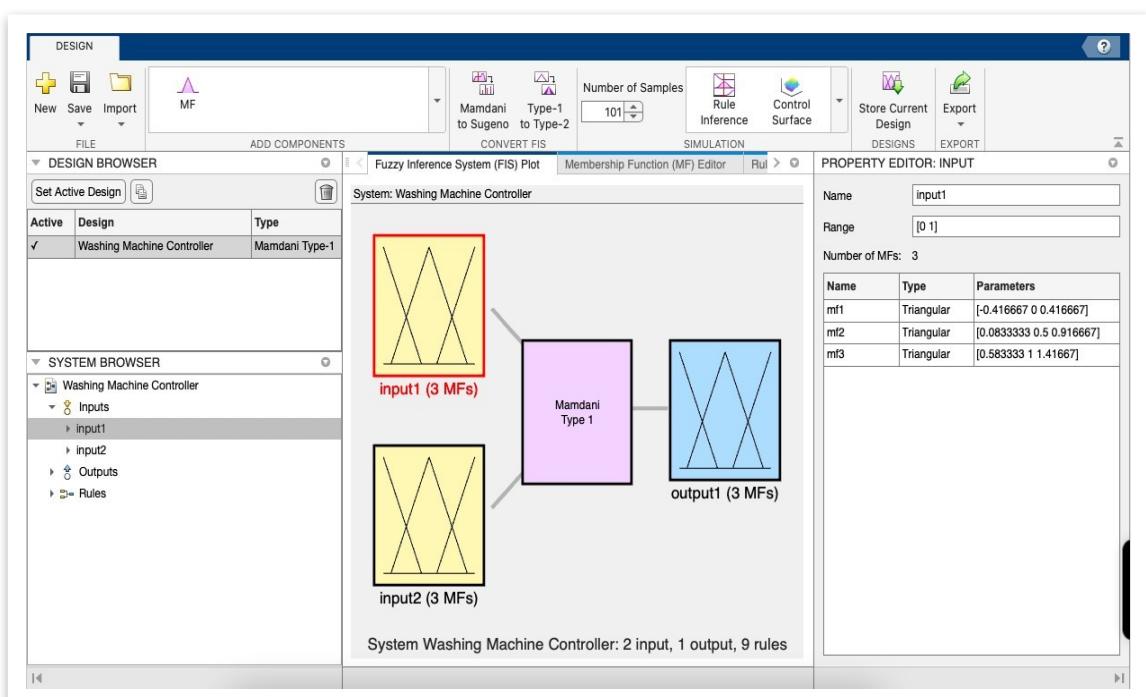
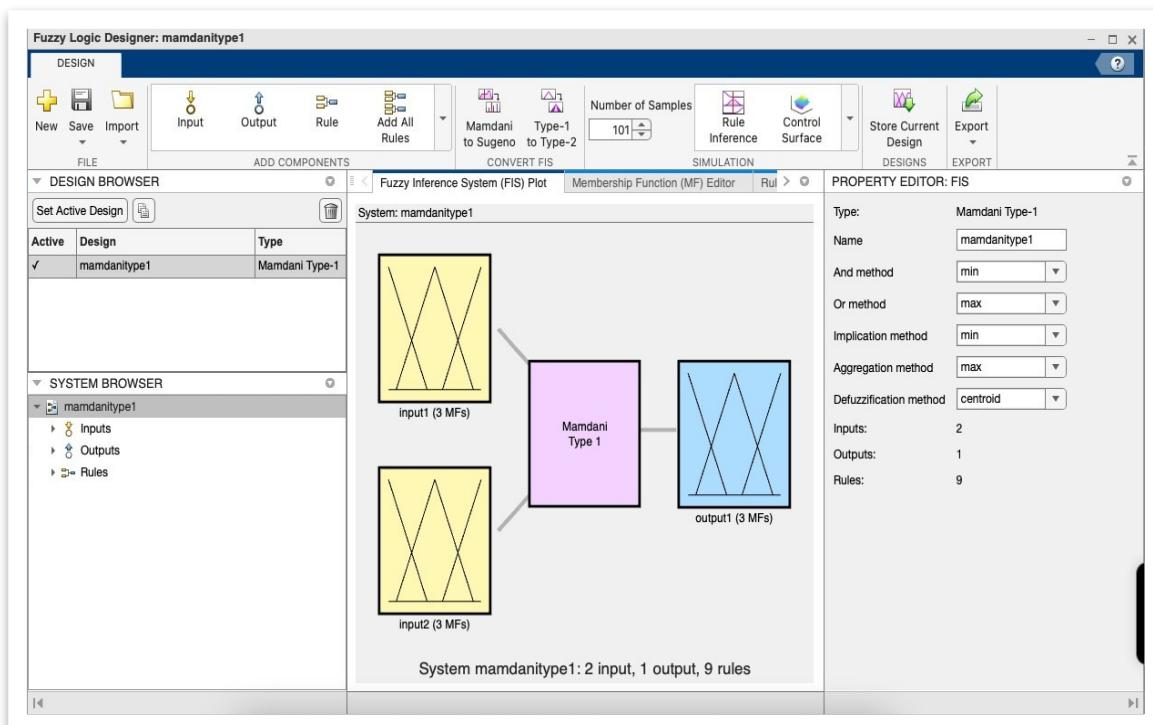
Software requirements: Matlab, Fuzzy Logic Designer

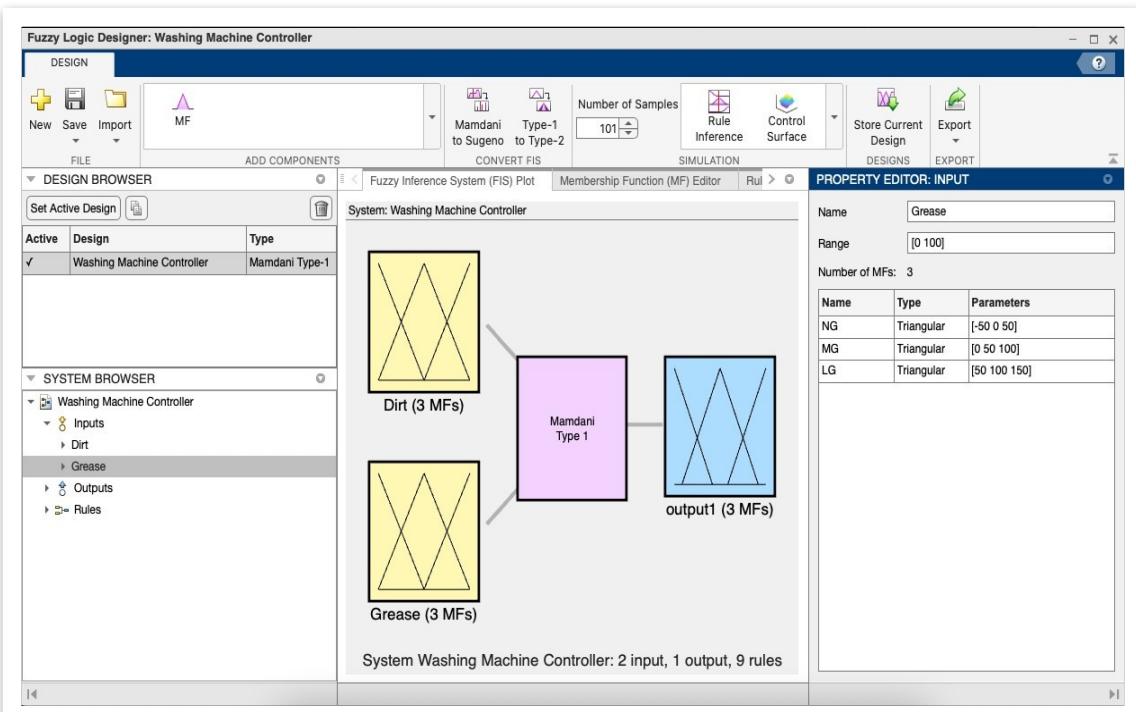
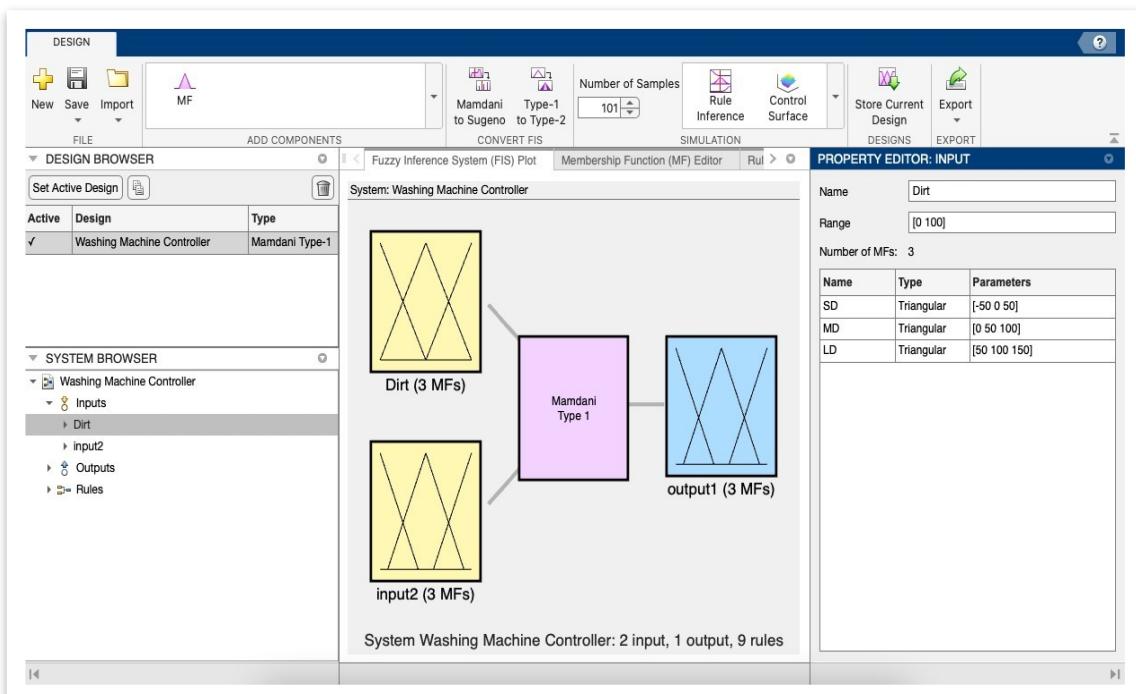
Steps:



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**





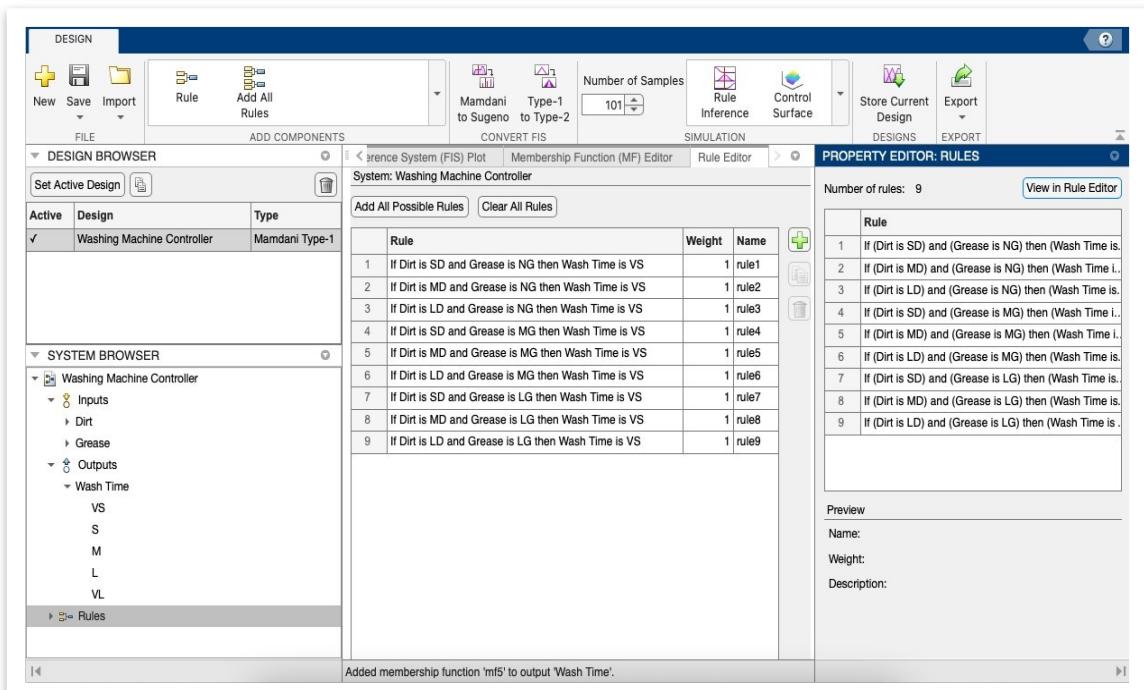
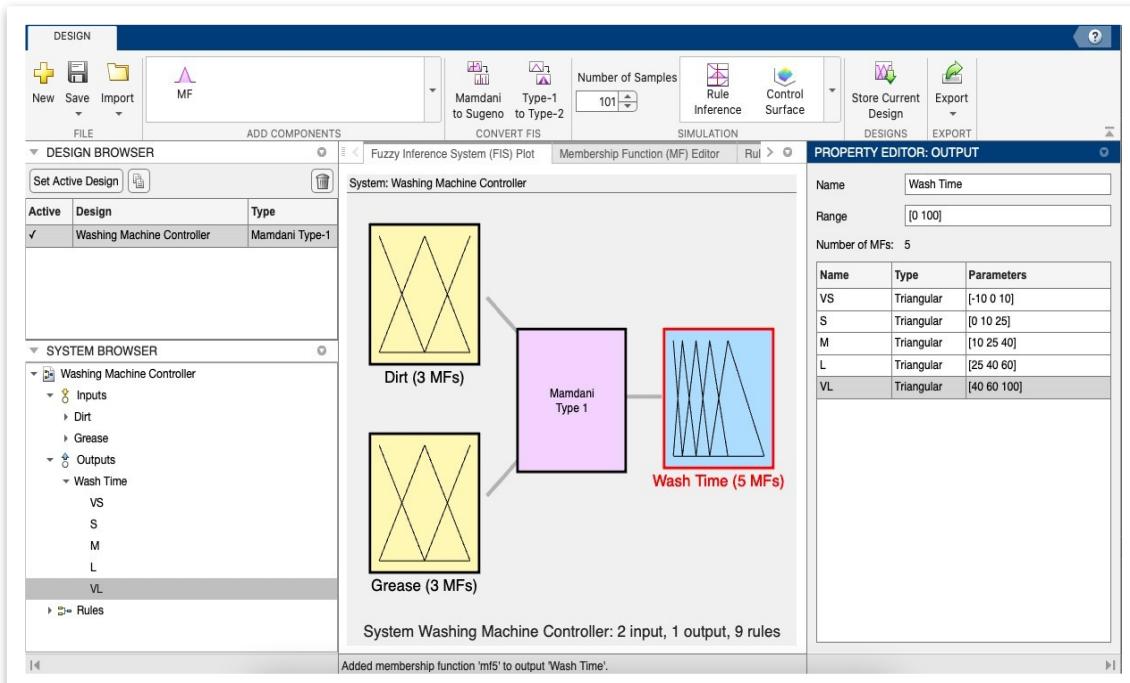




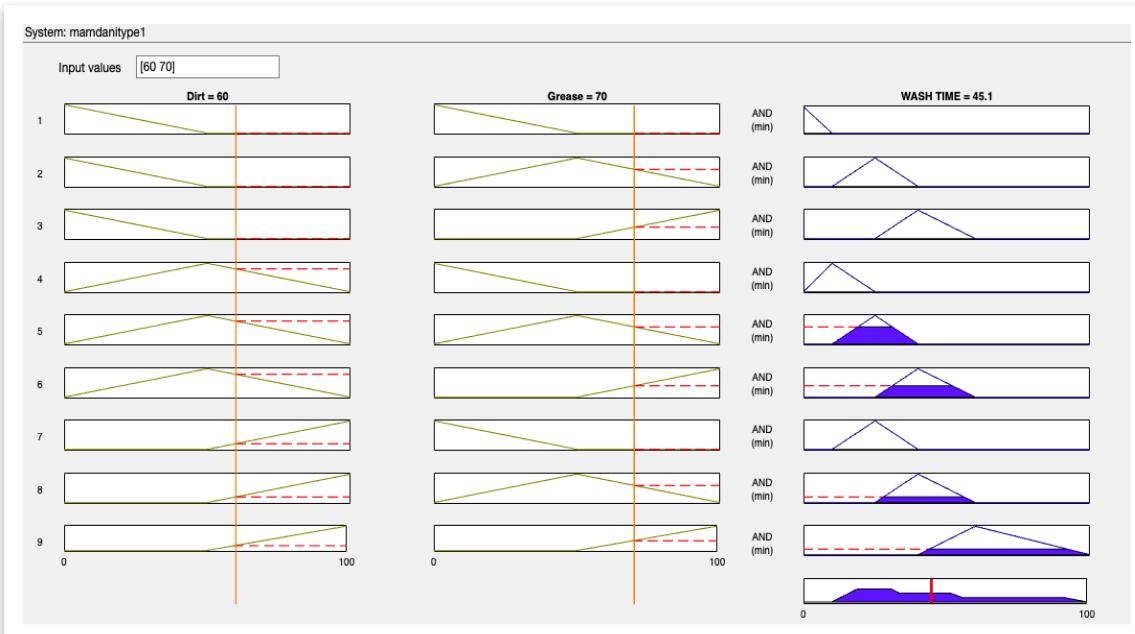
# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



	Rule	Weight	Name
1	If Dirt is SD and Grease is NG then WASH TIME is VS	1	rule1
2	If Dirt is SD and Grease is MG then WASH TIME is M	1	rule2
3	If Dirt is SD and Grease is LG then WASH TIME is L	1	rule3
4	If Dirt is MD and Grease is NG then WASH TIME is S	1	rule4
5	If Dirt is MD and Grease is MG then WASH TIME is M	1	rule5
6	If Dirt is MD and Grease is LG then WASH TIME is L	1	rule6
7	If Dirt is LD and Grease is NG then WASH TIME is M	1	rule7
8	If Dirt is LD and Grease is MG then WASH TIME is L	1	rule8
9	If Dirt is LD and Grease is LG then WASH TIME is VL	1	rule9



- Conclusion: The final Wash time = 45.1 minutes.



**Charotar University of Science  
and Technology**

**Devang Patel Institute of Advance  
Technology and Research**

**Department of Computer Engineering**



## **Practical – 3**

### **Constraint Satisfaction Problem**

#### **3.1: Constraint Satisfaction Problem –Crypt Arithmetic**

**[SEND + MORE = MONEY]**

#### **Code:**

```
def isSolvable(words, result):
    mp = [-1]*(26)
    used = [0]*(10)
    Hash = [0]*(26)
    CharAtfront = [0]*(26)
    uniq = ""
    for word in range(len(words)):
        for i in range(len(words[word])):
            ch = words[word][i]
            Hash[ord(ch) - ord('A')] += pow(10, len(words[word]) - i - 1)
            if mp[ord(ch) - ord('A')] == -1:
                mp[ord(ch) - ord('A')] = 0
                uniq += str(ch)
            if i == 0 and len(words[word]) > 1:
                CharAtfront[ord(ch) - ord('A')] = 1
    for i in range(len(result)):
        ch = result[i]
        Hash[ord(ch) - ord('A')] -= pow(10, len(result) - i - 1)
        if mp[ord(ch) - ord('A')] == -1:
            mp[ord(ch) - ord('A')] = 0
            uniq += str(ch)
        if i == 0 and len(result) > 1:
            CharAtfront[ord(ch) - ord('A')] = 1
    mp = [-1]*(26)
    return True

def solve(words, i, S, mp, used, Hash, CharAtfront):
    if i == len(words):
```



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



```
return S == 0
ch = words[i]
val = mp[ord(words[i]) - ord('A')]
if val != -1:
    return solve(words, i + 1, S + val * Hash[ord(ch) - ord('A')], mp, used, Hash,
CharAtfront)
x = False
for l in range(10):
    if CharAtfront[ord(ch) - ord('A')] == 1 and l == 0:
        continue
    if used[l] == 1:
        continue
    mp[ord(ch) - ord('A')] = l
    used[l] = 1
    x |= solve(words, i + 1, S + l * Hash[ord(ch) - ord('A')], mp, used, Hash,
CharAtfront)
    mp[ord(ch) - ord('A')] = -1
    used[l] = 0
return x
arr = [ "SIX", "SEVEN", "SEVEN" ]
S = "TWENTY"
if isSolvable(arr, S):
    print("Yes")
else:
    print("No")
```

#### Output:

**Conclusion:** In this practical we learned about . Constraint Satisfaction problem in ai.



## Practical – 4

### Natural Language Processing

**4.1: Regular Expression Perform Natural Language Processing Tasks [Text Reading, Text Analysis, Text Pre-processing, Text Classification, EDA, Stemming, Lemmatization] using NLTK using Python Programming.**

#### Code:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

data = pd.read_csv("Corona_NLP_train.csv",encoding='latin1')
df = pd.DataFrame(data)
df.head()

plt.figure(figsize=(10,5))
sns.countplot(x='Sentiment', data=df, order=['Extremely Negative', 'Negative', 'Neutral', 'Positive', 'Extremely Positive'], )

df.info()

reg = re.compile("(@[A-Za-z0-9]+)|([#A-Za-z0-9]+)|([^\w\A-Za-z\s])|(w+://S+)")
tweet = []
for i in df["OriginalTweet"]:
    tweet.append(reg.sub(" ", i))
df = pd.concat([df, pd.DataFrame(tweet, columns=["CleanedTweet"])], axis=1, sort=False)

df.head()
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
from sklearn.feature_extraction.text import TfidfVectorizer
stop_words = set(stopwords.words('english')) # make a set of stopwords
vectoriser = TfidfVectorizer(stop_words=None)

X_train = vectoriser.fit_transform(df["CleanedTweet"])
# Encoding the classes in numerical values
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
y_train = encoder.fit_transform(df['Sentiment'])
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train, y_train)

test_data = pd.read_csv("Corona_NLP_test.csv",encoding='latin1')
test_df = pd.DataFrame(test_data)
test_df.head()

reg1 = re.compile("@[A-Za-z0-9]+|[#[A-Za-z0-9]+]|([^\wA-Za-z ])|(w+://S+)" )

tweet = []

for i in test_df["OriginalTweet"]:
    tweet.append(reg1.sub(" ", i))
test_df = pd.concat([test_df, pd.DataFrame(tweet, columns=["CleanedTweet"])], axis=1,
sort=False)
test_df.head()

X_test = vectoriser.transform(test_df["CleanedTweet"])
y_test = encoder.transform(test_df["Sentiment"])
# Prediction
y_pred = classifier.predict(X_test)

len(y_pred)
```



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



```
pred_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
pred_df.head()

from sklearn import metrics
# Generate the roc curve using scikit-learn.
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred, pos_label=1)
plt.plot(fpr, tpr)
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')
plt.show()
# Measure the area under the curve. The closer to 1, the "better" the predictions.
print("AUC of the predictions: {0}".format(metrics.auc(fpr, tpr)))

import matplotlib.pyplot as plt
import numpy
from sklearn import metrics

confusion_matrix = metrics.confusion_matrix(y_test, y_pred)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix)

cm_display.plot()
plt.show()

Accuracy = metrics.accuracy_score(y_test, y_pred)
Accuracy
```

#### Output:

UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment	CleanedTweet
0	3799	48751	London	16-03-2020 @MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral	Gahan https t co iFz9FAn2Pa and https ...
1	3800	48752	UK	16-03-2020 advice Talk to your neighbours family to excha...	Positive	advice Talk to your neighbours family to excha...
2	3801	48753	Vagabonds	16-03-2020 Coronavirus Australia: Woolworths to give elde...	Positive	Coronavirus Australia Woolworths to give elde...
3	3802	48754	NaN	16-03-2020 My food stock is not the only one which is emp...	Positive	My food stock is not the only one which is emp...
4	3803	48755	NaN	16-03-2020 Me, ready to go at supermarket during the #COV...	Extremely Negative	Me ready to go at supermarket during the ou...



# Charotar University of Science and Technology

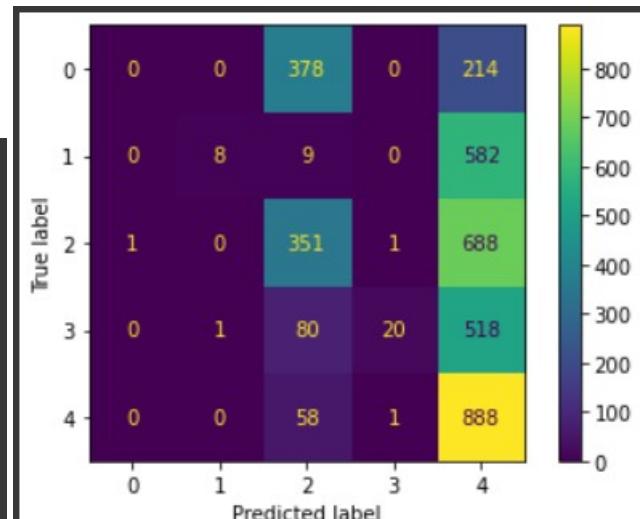
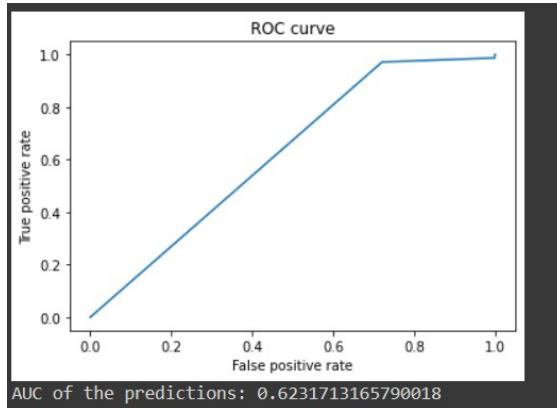
## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment	CleanedTweet
0	1	44953	NYC	02-03-2020 TRENDING: New Yorkers encounter empty supermar...	Extremely Negative	TRENDING New Yorkers encounter empty supermar...
1	2	44954	Seattle, WA	02-03-2020 When I couldn't find hand sanitizer at Fred Me...	Positive	When I couldn t find hand sanitizer at Fred Me...
2	3	44955	NaN	02-03-2020 Find out how you can protect yourself and love...	Extremely Positive	Find out how you can protect yourself and love...
3	4	44956	Chicagoland	02-03-2020 #Panic buying hits #NewYork City as anxious sh...	Negative	buying hits City as anxious shoppers stock...
4	5	44957	Melbourne, Victoria	03-03-2020 #toiletpaper #dunnypaper #coronavirus #coronav...	Neutral	19 One week everyone...

Actual	Predicted	
0	0	4
1	4	4
2	1	4
3	2	2
4	3	2



0.33359662980516064

**Conclusion:** In this practical we learned about NLP in AI.



## Practical – 5

### Chatbot Application

#### 5.1: Chatbot Building with SAP Conversational AI Track.

##### Output:

Step 1

##### Create SAP Conversational AI account

Go to <https://cai.tools.sap/>, and click Sign Up in the upper-right corner.

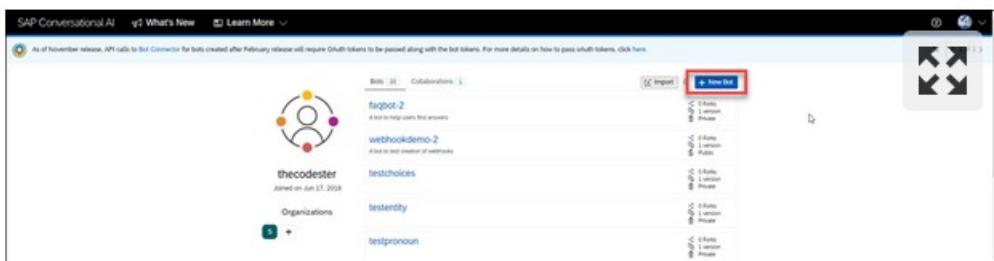
Follow the instructions for creating an account.

The screenshot shows the SAP Conversational AI registration interface. The main form is titled "Registration" and asks for "Tell Us About Yourself" with fields for First Name, Last Name, and E-Mail. Below that is a "Set Password" section with fields for Password and Re-Enter Password. At the bottom, there are "Terms and Conditions" and a checkbox for accepting them. A "Required" indicator is shown next to the password field. A "Register" button is at the bottom right. In the background, there's a "SAP" logo and a "Create SAP ID" link. A sidebar on the left has a "Create SAP ID" button and a note about existing accounts. The top navigation bar includes links for "What's New" and "Learn More".

## Step 2

### Create new bot project

Click **New Bot**.



Fill in the following:

- Select **Perform Actions**.

With a standard **Perform Actions** bot, the developer is responsible for creating entities and intents with expressions. The developer is also responsible for building and managing the conversational flow that pulls information from back-end systems to help simplify processes for the chatbot end user.

An **FAQ** bot retrieves answers to users' questions from one or more documents (.csv files) that you upload. The document must include predefined pairs of questions and answers. This allows your bot to map the user's query to the best match and retrieve an answer without interpreting the intent of the question.

To ease the complexity of the FAQ bot, the intents and entities are predefined and hidden, and the bot includes a set of predefined skills. However, you can design the bot responses as per your business needs.

- For the predefined skills, choose **Greetings**.
- In the **Create your bot** section, enter the following:

#### Field Name Value

Bot name **my-first-bot**

Description **A bot that likes to tell jokes and have a little fun**

- For **Data Policy**, select the following:
  - o **Non-personal**.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



- o **Store.**
- o **Non-vulnerable.**
- For Bot visibility, select **Public**.

The screenshot shows the SAP Conversational AI setup interface. Step 1: "What do you want your chatbot to do?" offers two options: "Perform Actions using conversational flows" and "Retrieve Answers from FAQ documents". Step 2: "Select predefined skills for your bot" lists five skills: Greetings, Small Talk, Weather, Customer Satisfaction, and Set Alarm. Step 3: "Create your bot" shows a workspace with a code editor containing "thecoderster / my-first-bot", a description field ("A bot that likes to tell jokes and have a little fun"), and a topics section.

Step 3

#### The 4 stages of a bot's life

There are 4 stages in your bot's life:

- **Train:** Teach your bot what it needs to understand.

Here you will create **intents** – ideas your bot will recognize – using **expressions** that the bot should recognize coming from the user.



- **Build:** Create your conversational flow with our Bot Builder tool. Here you give your bot skills it can perform.

Here you will create **skills** – things your bot can do – and define when they will be triggered. Your skills are defined by **triggers, requirements** (information your bot must collect), and the **actions**.

- **Connect:** Ship your bot to one or several messaging platforms.
- **Monitor:** See how users are communicating with your bot, check if the bot is understanding users correctly, and make updates to the bots intents/entities. Monitoring also lets you see what your users want, and gives you ideas of what additional capabilities you could add to the bot.

The 4 phases are represented in the bot project by the 4 tabs.

A screenshot of the SAP Conversational AI interface. At the top, there's a navigation bar with a back arrow, the project name 'thecoder / my-first-bot-2', a version dropdown set to 'v1', a 'development' button, and a dropdown menu. Below the navigation is a summary section with 'Created today at 15:41' and a description 'A bot that likes to tell jokes and have a little fun'. There's a 'Edit' button next to the description. Below this is a 'Add Topics' button. At the bottom is a horizontal navigation bar with four tabs: 'Train' (which is highlighted with a red border), 'Build', 'Connect', and 'Monitor'.

#### Step 4

##### Fork Intent

Now we will work in the **Train** tab, where we will build intents. Remember, intents are the things people want to tell our bot. Within an intent, we indicate all the possible expressions that a person might use to communicate that intent.

As SAP Conversational AI is collaborative, you do not have to re-create each intent every time. You can “fork” an intent someone has already created and clone it right into your bot.

1. Go to the **Train** tab.
2. In the **Search** box, enter **joke**, and press **Search**.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



SAP Conversational AI    What's New    Learn More

As of November release, API calls to Bot Connector for bots created after February release will require OAuth tokens to be passed along with the bot tokens. For more details on how to pass OAuth tokens, click [here](#).

theicodester / my-first-bot-2    v1    development

Created today at 15:41  
A bot that likes to tell jokes and have a little fun

Add Topics

Train    Build    Connect    Monitor    Settings

Intents    Entities

Your training Today 10/18/21 at 15:41:31 finished successfully on Today 10/18/21 at 15:41:34

★ TIPS  
Hey there! I can help you to improve your bot, please follow the instructions to train your bot well!

Create and manage your intents

Your intents    What's an intent?

joke    Search    + New Intent

Find an intent within your list...

@ goodbye    # Number ★    # Organization ★    # Duration ★    more... (52)

@ greetings    # Person ★    # DateTime ★    # Pronoun ★    (48)

You will get a list of intents to fork.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



The screenshot shows a search interface with a search bar containing 'joke'. Below the search bar are three search results:

- lucero-davi / sales-cloud-assistant-cai-redesign / @ask-joke**  
Ask for a funny joke  
A joke for me • Can you tell me a joke ? • just be funny?  
Language: en → Fork
- srvenkatz5 / joke-bot-7 / @ask-joke**  
Ask for a funny joke  
Botty bot, can you tell me a joke please? • Tell me a joke • Can you tell me a joke ?  
Language: en Fork
- vitalii\_romanenko-epam-com / sendlinks / @ask-joke**  
Ask for a funny joke  
Tell me a joke • Can you tell me a joke ? • A joke please  
Language: en Fork

Click **Fork** for the first @ask-joke intent.

There are many existing intents and the choices are always changing.

Explore the intent by clicking on it.

In the intent we forked, there are 2 expressions to discern if someone wants to be told a joke.  
In yours, you may receive more expressions.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



A screenshot of the Rasa Platform interface. At the top, there's a navigation bar with 'Train' (highlighted with a yellow arrow), 'Fork' (0), 'Export' (0), and other options like 'Settings'. Below the navigation, the path 'thecodester / my-first-bot-2 / v1 / development' is shown. A red arrow points to the 'my-first-bot-2 / intents / @ask-joke' section. A tooltip says 'Ask for a funny joke'. In the 'Intent overview' section, there are counts for 25 entities and 25 pronouns. The 'Expression' section shows two examples: 'Something fun?' and 'Can you share with me a kinky joke?'. Both examples are highlighted with a red border. There are also 'Import Expressions' and 'Create' buttons.

#### Step 5 Create new intents

You will be able to reuse many, many intents created by others. But there are times you will want to create your own.

1. Click the **Train** tab.
2. Create an intent for recognizing good reactions to jokes.
  - o Click **Create**.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



Created today at 15:41  
A bot that likes to tell jokes and have a little fun

Add Topics

Train Build Connect Monitor Settings

Intents Entities

Your training Today 10/18/21 at 15:41:31 finished successfully on Today 10/18/21 at 15:41:34 but is outdated. Additional changes in your dataset have been made during the training. Please try again.

★ TIPS Some of your intents have less than 50 expressions. Ensure your intent has at least 50 expressions to improve your dataset. Add expressions with different sentences but the same meaning and be sure to have only one sentence by expression

Create and manage your intents

Your intents What's an intent?

Search and fork an intent from the community + New Intent

Find an intent within your list

- For the name of the intent, enter **laughs**.
- For the description, enter **A natural reaction to our awesome jokes**.

2.

3.

Let's create your intent

thecoder / my-first-bot-2 / laughs

Your intent description (optional)

A natural reaction to our awesome jokes.

Matching Strictness

0 50 100

Cancel Create Intent

- o Click **Create Intent**.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



4. Create a second intent, this time for bad reactions to jokes.
  - o Click **Create**.
  - o For the name of the intent, enter **lame**.
  - o For the description, enter **You can't succeed every time**.
  - o Click **Create Intent**.

#### Step 6

Now that you have created 2 intents, we need to add expressions – things people might say if they had that intent. A golden rule would be to add **at least 30 expressions** to an intent, and ideally more than 50.

Put yourself in the shoes of the people talking to your bot. What could they possibly ask?

For this tutorial, you do not have to come up with 50 expressions but you will add 4 to each intent.

1. Click on the @laughs intent.
2. In the expression field, enter a sentence you want your bot to understand, then press **Enter**.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



The screenshot shows the Microsoft Bot Framework Composer interface. At the top, it says 'Created today at 15:41' and 'A bot that likes to tell jokes and have a little fun'. Below this, there's a 'Train' tab, a 'Build' tab, a 'Connect' tab, and a 'Monitor' tab. A 'Settings' button is also present. In the main area, it shows the intent 'my-first-bot-2 / intents / @laughs'. A tip message says 'Hey there! I can help you to improve your bot, please follow the instructions to train your bot well!'. Below this, it says 'Train your intent using expressions'. Under 'Intent overview', it says 'Intent overview is available after adding expressions...'. In the 'Expression' section, there's a British flag icon, a '+ Add Language' button, and a text input field containing the expression 'I haven't laughed that much in a long time!'. A red box highlights this input field. To the right of the input field is a blue 'Import Expressions' button. Below the input field, it says 'No expressions added' and 'Start by adding expressions above to build your intent.'

Here are some examples for the laughs intent you added:

Hahaha that's hilarious  
ROFL you're good!  
That, my friend, was an amazing joke.  
I haven't laughed that much in a long time!

For a production bot, you want 30-50 expressions, all the ways someone is likely to express their intent. In addition, after deploying the bot, you can review what users are writing and add those.

Do the same for the @lame intent (you'll have to navigate to the intent by click **Train** tab or navigating the breadcrumb).

Here are some examples for the lame intent:

You have no sense of humor whatsoever.  
That's both terrible and offensive.  
What the heck was that?  
Try harder, that was a very bad joke.



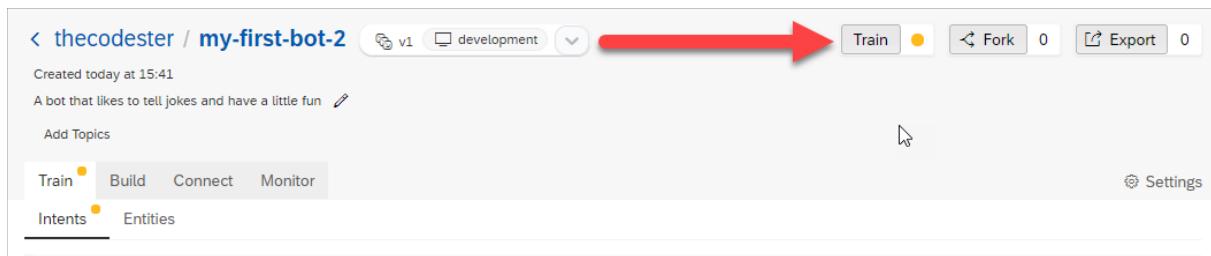
3.

#### Step 7

We want to now test to see if the bot can detect when a user says something matching the intent we created.

1. Click the **Train** button at the top-right, so that the bot trains itself based on the latest intents and expressions (sometimes referred as your dataset).

The button indicates the current status of training: Yellow is out of date, blue is training, green is up to date..



By default, your bot does not train itself but you must click Train for it to understand the latest intents and expressions.

You can change the setting so the bot automatically trains itself at **Settings > Version > (current version dropdown)**, and scroll down to .



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



the codeester / my-first-bot-2 v1 development

Train Fork 0 Export 0

Created today at 15:41  
A bot that likes to tell jokes and have a little fun

Add Topics

Train Build Connect Monitor Settings

Bot Options Use versioning to track progress, iterate or manage your bot [How does this work?](#)

Collaborators

System Aliases

Tokens

Data Policy

Danger Zone

Versions 2

Environments

Version 1

+ New Version

3

Tokens

Request token

337c595008c157abbf600b7e83da6192

You can use the request token above to make runtime API requests on:

<https://api.cai.tools.sap/build/v1/dialog>  
to use bot builder ([API Reference](#))  
<https://api.cai.tools.sap/v2/request>  
to analyze text ([API Reference](#))

## NLP

### Context management

#### Resolve pronouns

Map pronouns to entities in the conversation history

#### Resolve descriptions

Map superlatives to list replies in the conversation history

### Training Mode

#### Automatic

Training is automatically triggered by any change to the bot

#### Manual

You decide when you want to update your training

At the bottom of the panel, open the **Expression Analysis** tab.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



Create and manage your intents

Your intents What's an intent?

Search and fork an intent from the community  + New Intent

Find an intent within your list...

Edit mode

Intent	Entities	Count
@ lame	# Pronoun ★ # Number ★	4
@ laughs	# Number ★ # Pronoun ★	4
@ ask-joke	# Pronoun ★ # N...	...
@ goodbye	# Number ★ # Organization ★ # Duration ★	more...

Enter something that you think should match the intent, like:

Text  
Copy  
Botty bot, can you tell me a joke please?

If the intent is successfully trained, the test will show that the bot recognized the intent (as well as entities within the expression).

Step 9

Now let's build a skill to tell a joke.

1. Go back to the **Build** tab – where you see the canvas.  
2. Click **Add skill**.

There are several types of skills:

- o **Business** skills reflect the core purpose of your bot.
- o **Floating** skills complement your bot's core business skills (for example, small talk).
- o The **Initialize** skill (there can only be one in a chatbot) is triggered when the conversation with the user starts.

3. Call your skill **tell-me-a-joke**, set the type as **Business**, and click **Add**.





# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



A W

all rk

1y E

nd M

ec I

Add Skill

Name

tell-me-a-joke 241

ex. book-flight, checkout

Type

Business Floating Initialize

Business skills reflect the core purpose of your bot.

Activation (On)

Your skill is active. If you deactivate it, your bot will no longer use it when chatting.

Title

Your bot automatically uses this skill title when disambiguating during chat. If you don't specify one, your bot will display: tell-me-a-joke

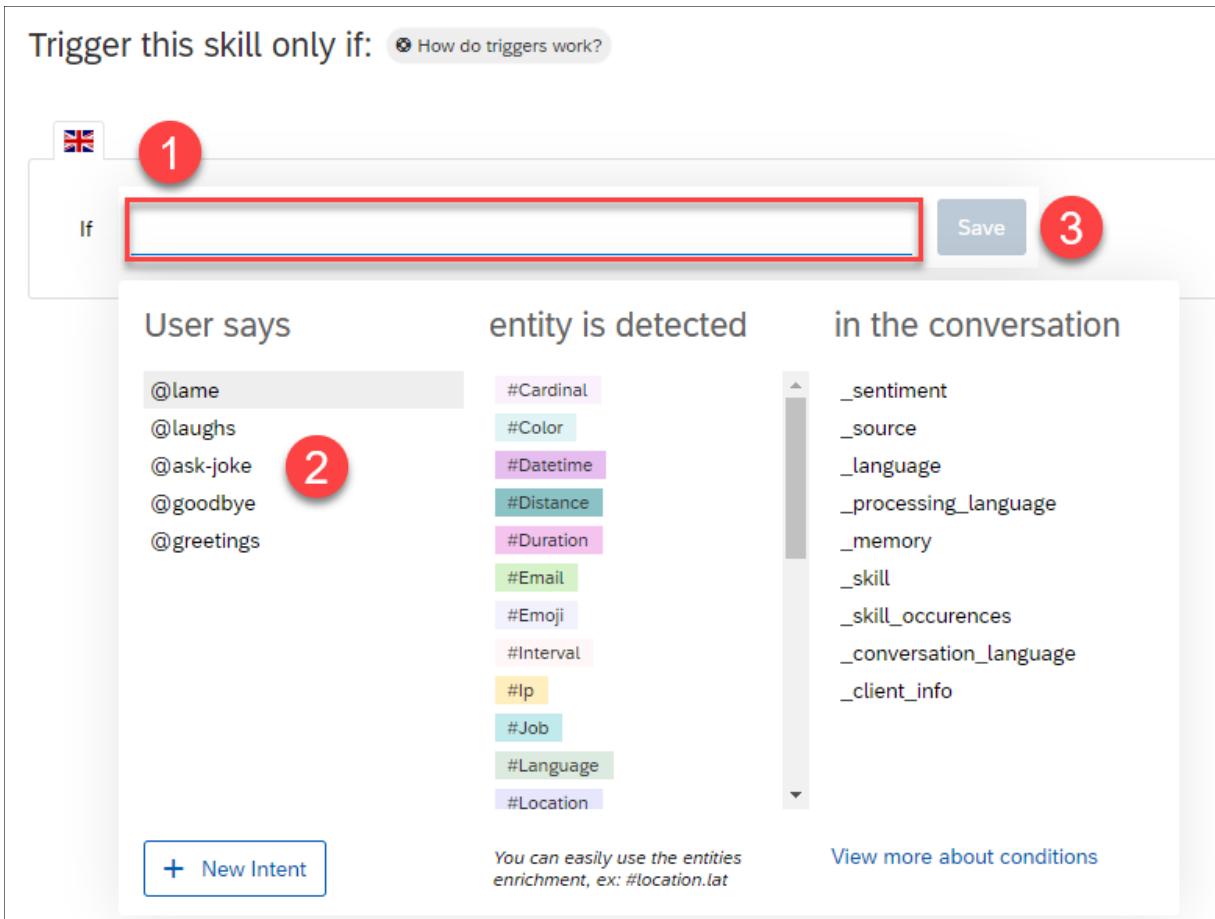
Type a short title for this skill's quick reply button...

Cancel Add

Click the new **tell-me-a-joke** skill, and open the **Triggers** tab.

Add 3 **If** statements, one for each of the @ask-joke, @laugh, and @lame intents.

- Click in the empty space right after If, select the @ask-joke intent, and then click **Save**.



Trigger this skill only if: [How do triggers work?](#)

If  Save

User says entity is detected in the conversation

@lame	#Cardinal	_sentiment
@laughs	#Color	_source
@ask-joke	#Datetime	_language
@goodbye	#Distance	_processing_language
@greetings	#Duration	_memory
	#Email	_skill
	#Emoji	_skill_occurrences
	#Interval	_conversation_language
	#Ip	_client_info
	#Job	
	#Language	
	#Location	

[+ New Intent](#) You can easily use the entities enrichment, ex: #location.lat View more about conditions

- Click on the + sign – **Add a new list of conditions** – and repeat the above for @laughs and @lame intents.
- Change the logical conditions to Or between each If statement, simply by clicking And.

It should now look like this:



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



Trigger this skill only if: [How do triggers work?](#)



Go to the **Actions** tab.

Click **New Action Group**, then **Add Condition**.

After the If, select the @ask-joke intent, and click **Save**.

Click **Choose Message Type**, choose the **Text** format, and type in a really good joke, like:

Joke

Copy

I just flew into town, and boy are my arms tired.

You can define additional jokes, and one of the jokes from the set will be displayed randomly.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



8.

When all requirements are met, craft your bot's response What's an action?

A screenshot of a bot builder interface. At the top, there is a condition: "If @ask-joke is-present". Below this, a message card is shown with a placeholder icon and the text "I just flew into town, and boy are my arms tired.". A red arrow points to the placeholder icon. At the bottom of the message card, there is a note: "Markdown syntax is disabled." Below the message card, there are four buttons: "Choose Message Type", "Connect External Service", "Connect Fallback Channel", and "Update Conversation".

You can also set a delay (optional) between two messages, up to 5 seconds. This might be useful when the messages your bot sends are quite long and need time to be read by the user.

9. Click **Save**.

Step 10

Before releasing your bot to your audience, chat with it in real situation. This will help show you how the bot will behave in a “real” conversation.

Double-check if there are any errors in the conversation flow (e.g., bad answers, fallback errors, weird behavior) and correct them before sending the bot out.

Chat with your bot as often as you can while you build it, which will make it easier to find problems.

Click on the bottom-right yellow button **Chat Preview** and start sending messages. Use the clear icon



at the top of the panel to refresh the chat.





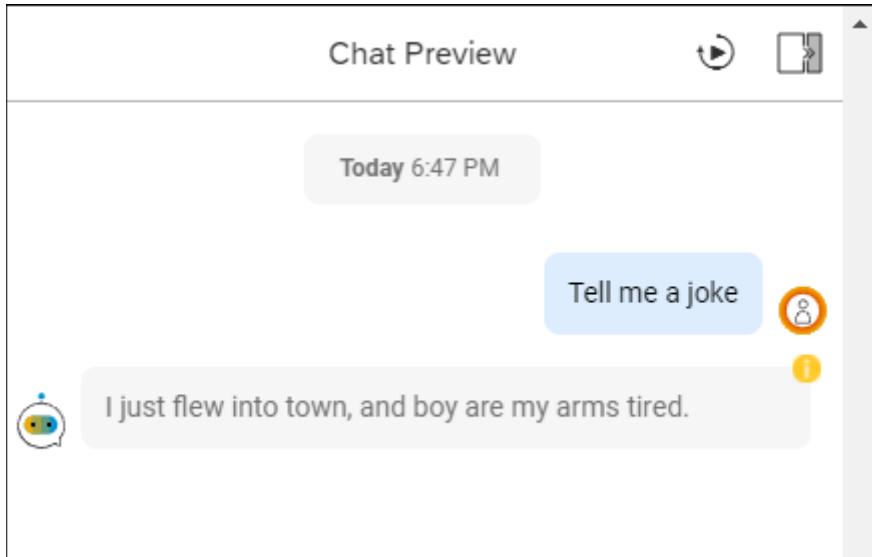
# Charotar University of Science and Technology

Devang Patel Institute of Advance Technology and Research

Department of Computer Engineering



- If you say **Tell me a joke**, the bot will tell you a joke.
- Try other phrases and see how the bot responds. If the bot does not respond properly, then try improving your intent with additional expressions.





# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



User is referring to:

@ask-joke

In the expression there are:

- # Pronoun ★ you
- # Pronoun ★ me
- # Number ★ a

See more in the JSON view

Botty bot, can you tell me a joke please?

3.

If the bot did not recognize the intent (or selected the wrong intent), you will need to train your bot some more by adding additional expressions (Step 6).

#### Step 8

Now that your bot knows how to understand people who talk to it, it's time to give your robot some skills.

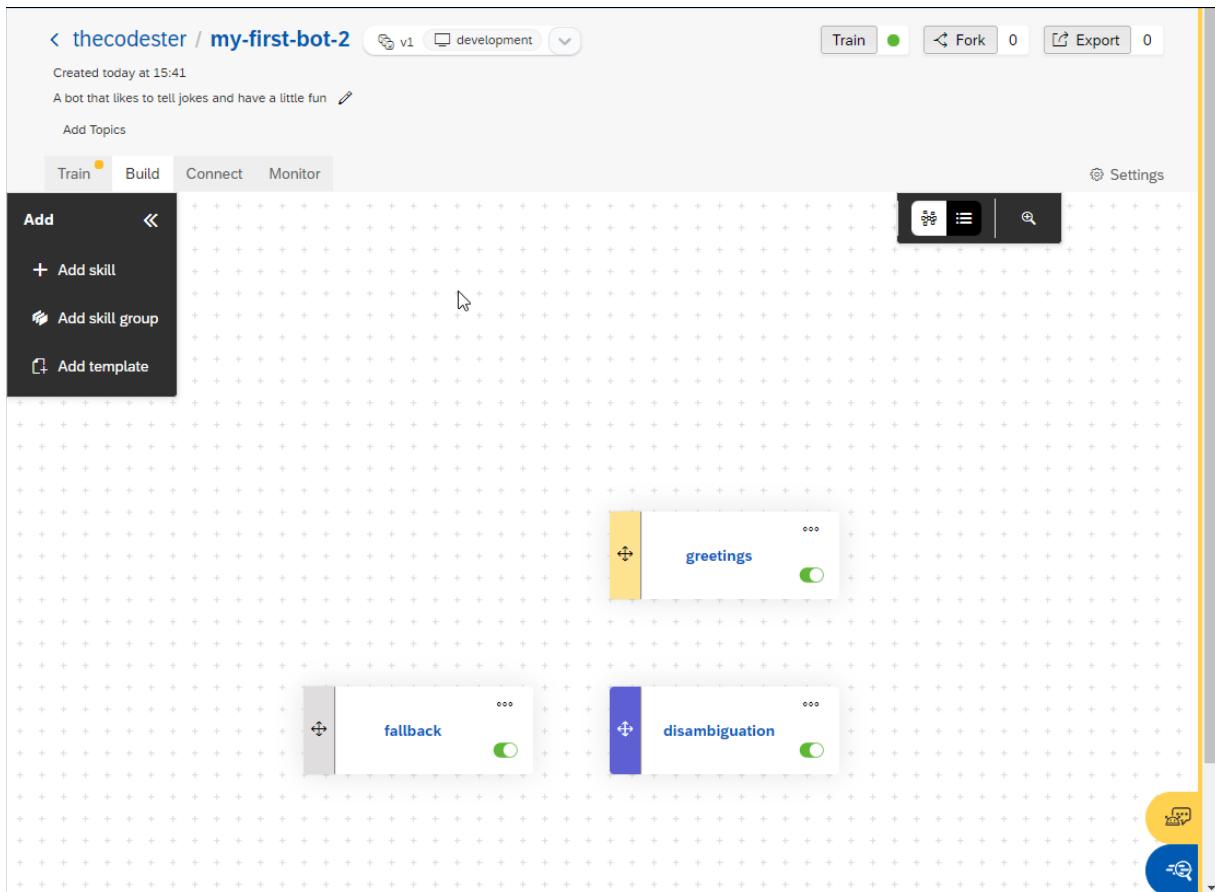
Open the **Build** tab.



# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



Inside, you'll find the **Bot Builder**, which helps you construct the conversation flow of your bot.

## What is a skill?

Each skill represents one thing that your bot knows how to do. Your skill can be complicated (e.g., manage payment by credit card) or quite simple (e.g., answer a basic question).

Just like intents, you can create a skill from scratch or inherit skills from other bots you've created or from other people's bots.

In our project, choose the predefined skill **Greetings**.

The **Greetings** skill – like all skills – has 4 tabs:

- **README.md:** A description of the purpose of your skill
- **Triggers:** The conditions that must occur – generally the intents that the user must express – for the skill to be executed



**Charotar University of Science  
and Technology**

**Devang Patel Institute of Advance  
Technology and Research**

**Department of Computer Engineering**



- **Requirements:** Information that must be collected in order for the skill to be executed
- **Actions:** The action to take (basically, this is the skill)

If you navigate through the tabs, you'll see that the **Greetings** skill is structured as follows:

- It is triggered if either the intent @greetings or @goodbye is matched.
- It has no requirements because it does not need to collect additional information. That means that it will execute actions directly after being triggered.
- It has two possible actions: If the @greetings intent is matched, it sends a random welcoming message chosen from a list. If the @goodbye intent is matched, it does the same thing, but picks the message from a different list

**Conclusion:** In this practical we learned how to create a chatbot using SAP tool online.

## Practical – 6

### Various Tools for ML Techniques

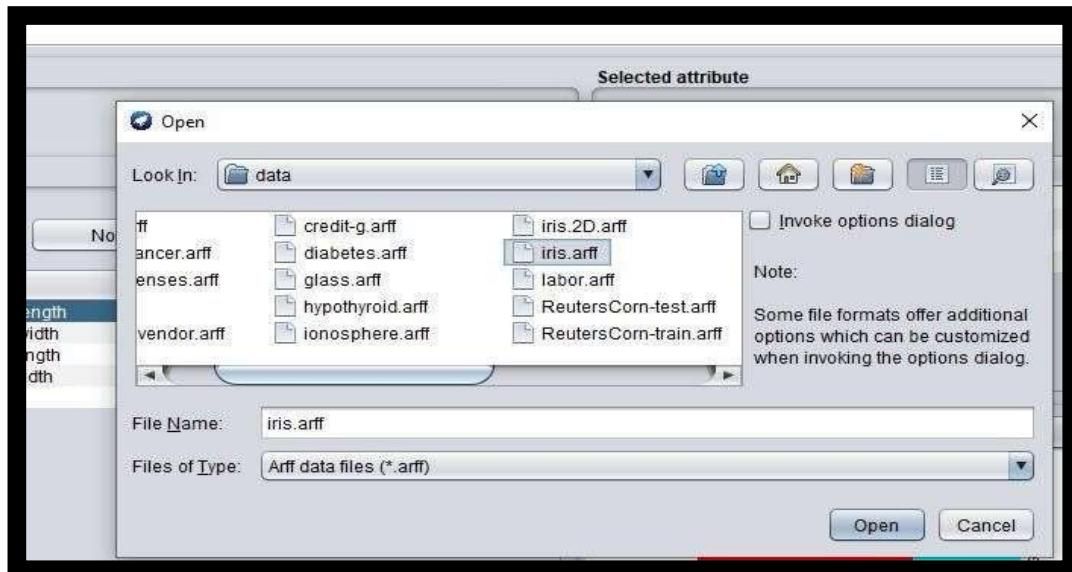
**6.1: Perform classification on Iris dataset using neural network tools such as WEKA, ORANGE**

**Output:**

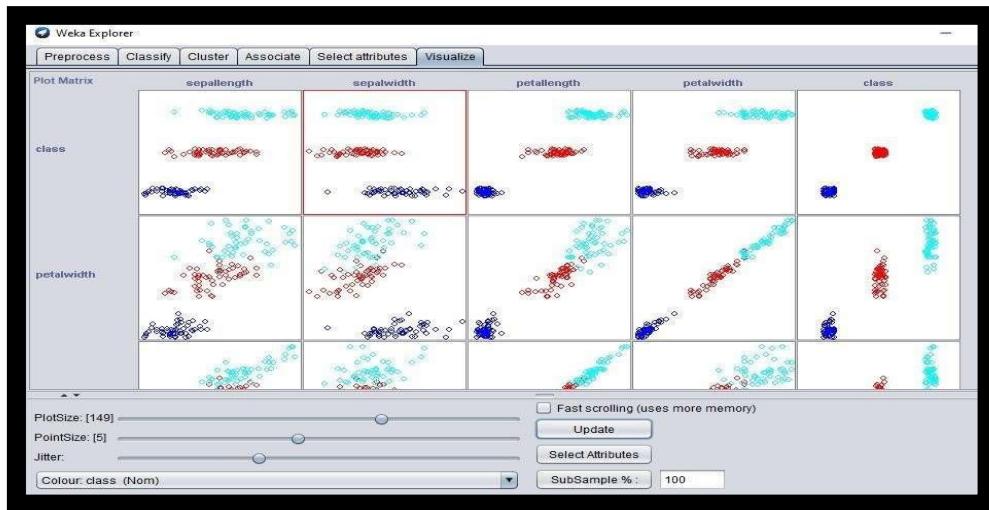
**Using WEKA**

Classification on Iris Dataset:

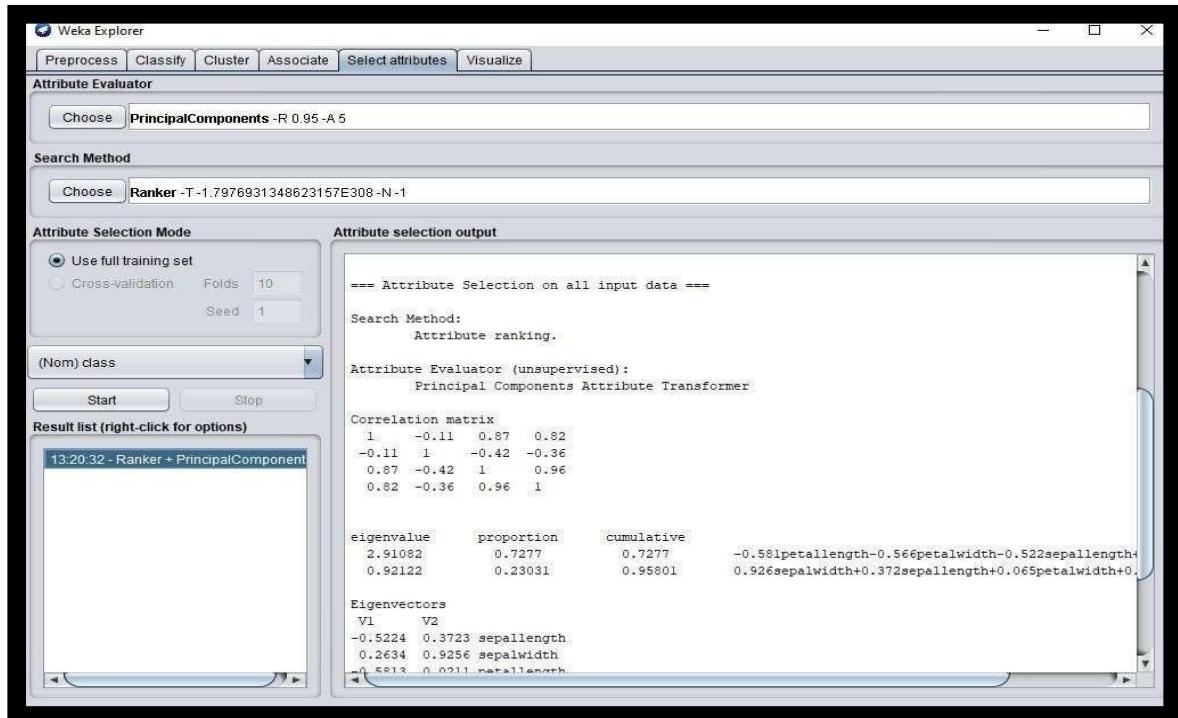
1. Loading the dataset



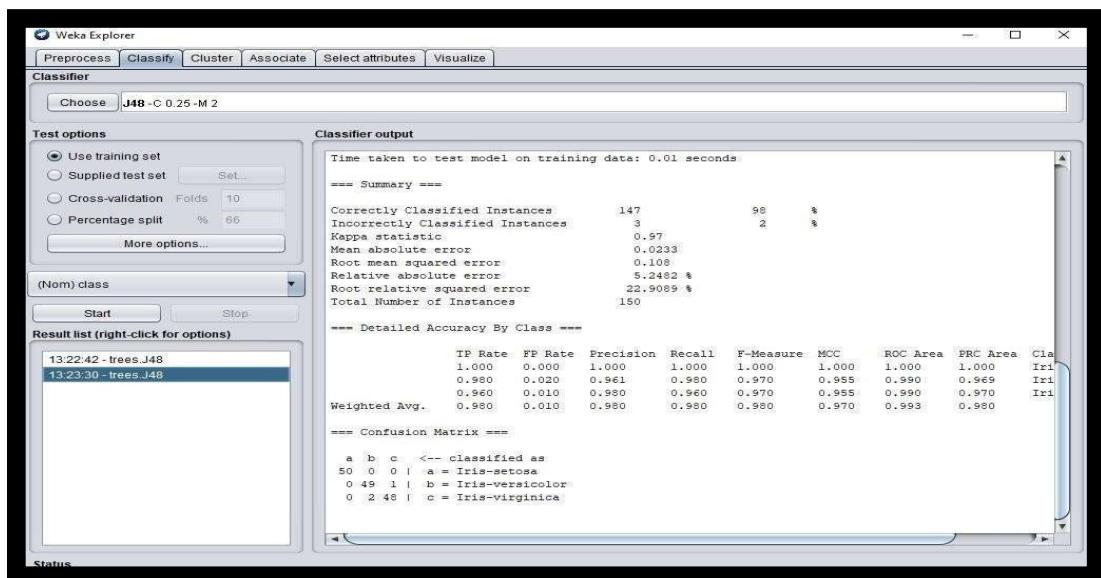
2. Editing the dataset



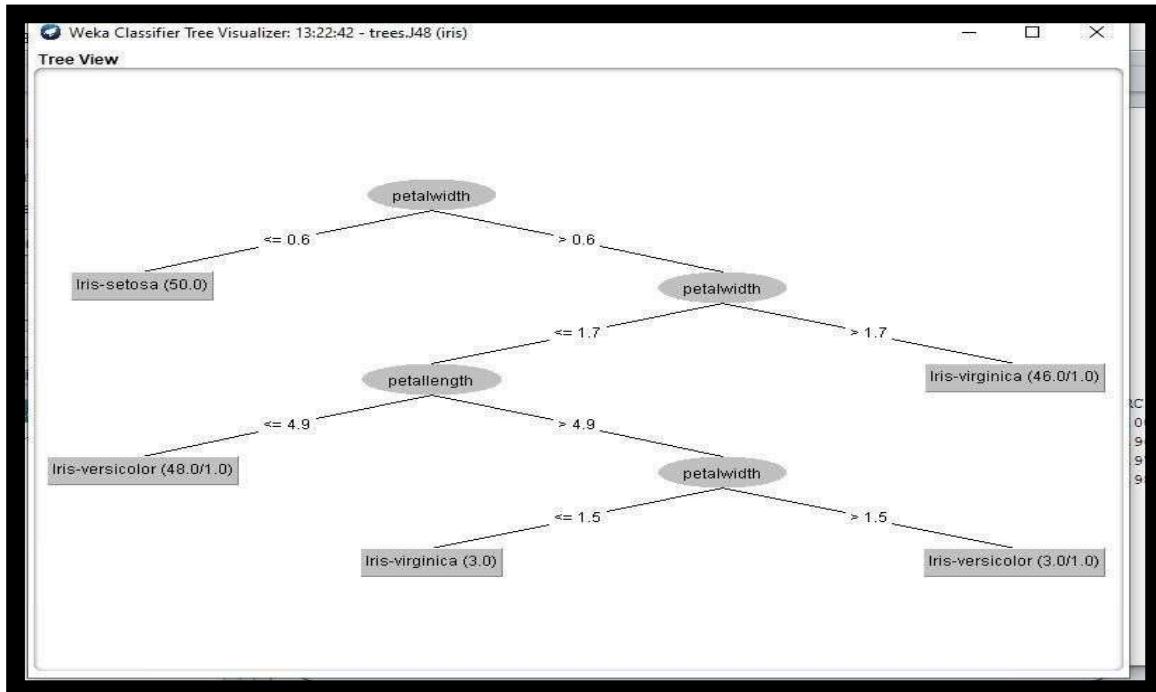
### 3. Selecting the Attributes



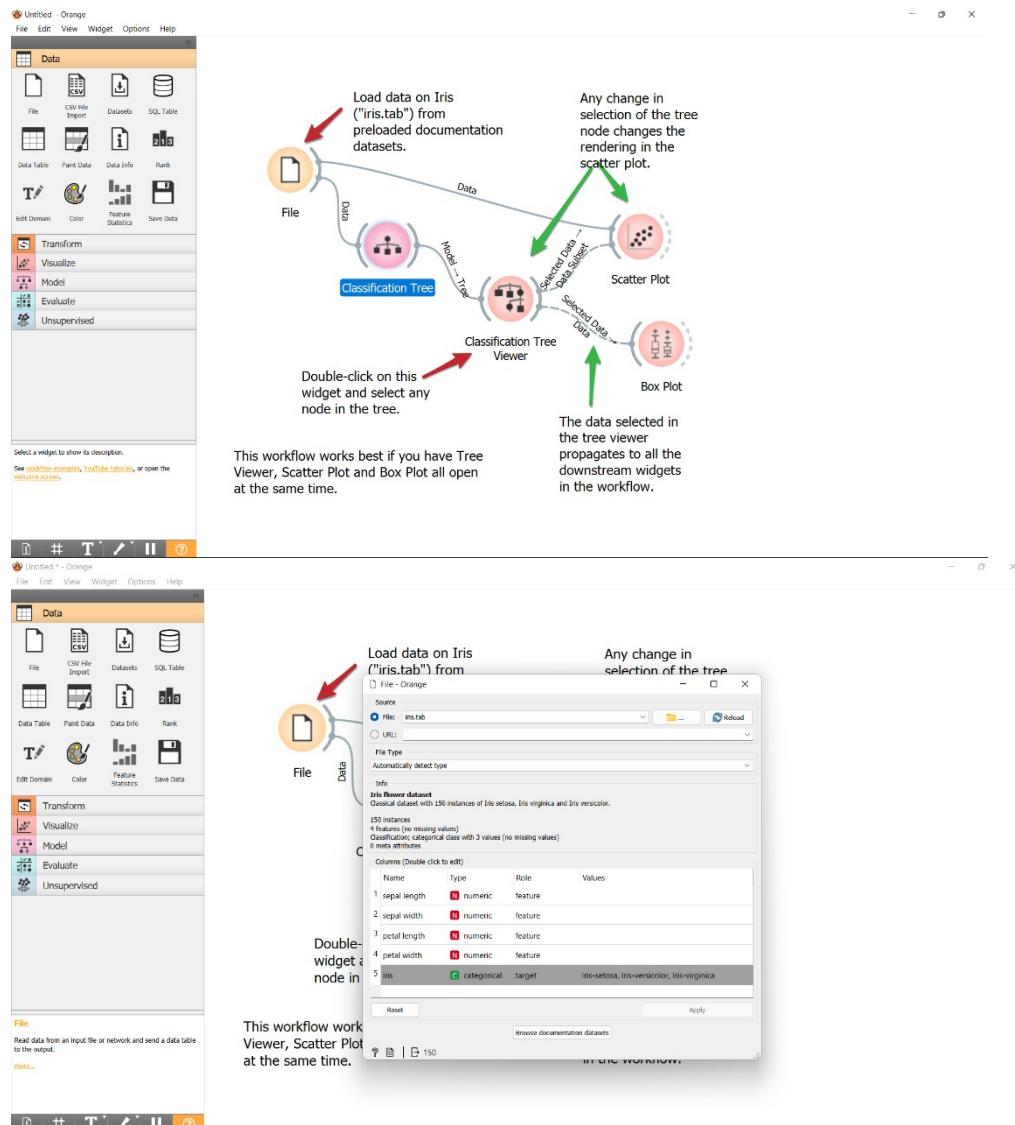
### 4. Classifying the dataset - J48 – C 0.25 – M 2



5. Classification of Iris Dataset in Tree View



#### Using Orange:



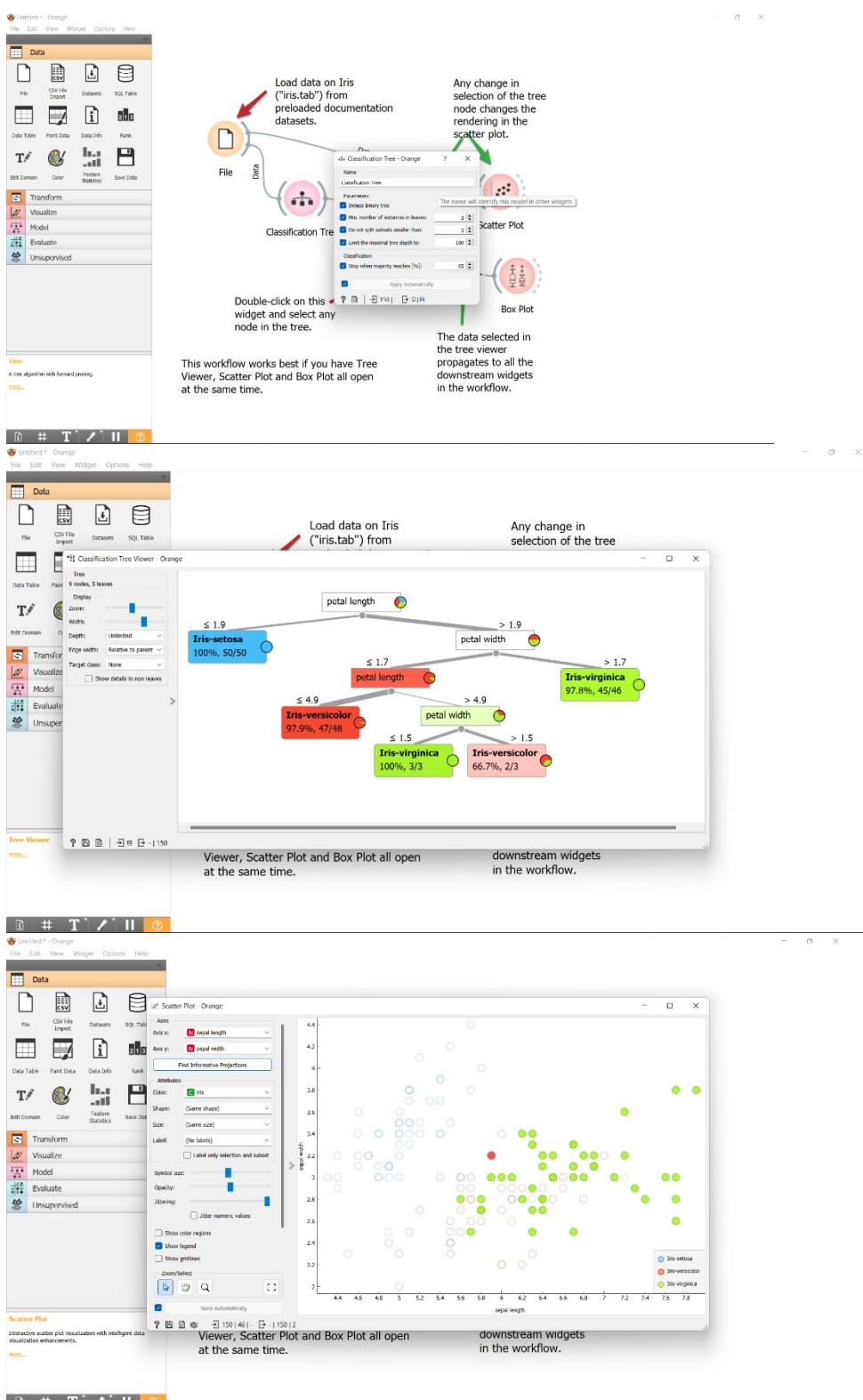
The screenshot shows the Orange data mining software interface with two main windows. The top window displays a workflow diagram for analyzing the Iris dataset. The bottom window shows the 'Data' tab of the 'File' dialog, where the 'iris.tab' file is selected as the source.

**Workflow Diagram Labels:**

- Load data on Iris ("iris.tab") from preloaded documentation datasets.
- Any change in selection of the tree node changes the rendering in the scatter plot.
- Double-click on this widget and select any node in the tree.
- This workflow works best if you have Tree Viewer, Scatter Plot and Box Plot all open at the same time.
- Classification Tree
- Data
- Model - Tree
- Classification Tree Viewer
- Scatter Plot
- Box Plot
- Selected Data
- Selected Data Subset
- Selected Data Dialog
- Histogram

**Data Tab Labels:**

- File
- Data
- Source
- File Type
- Info
- Iris flower dataset
- 150 instances
- 4 features (no missing values)
- Classification/ Categorical class with 3 values (no missing values)
- 0 metas attributes
- Columns (Double click to edit)
- Name
- Type
- Role
- Values
- sepal length
- sepal width
- petal length
- petal width
- iris
- target
- Iris-setosa, Iris-versicolor, Iris-virginica
- Reset
- Apply
- Browse documentation datasets



Untitled - Orange

File Edit View Widget Options Help

Data

File CSV File Import Datasets SQL Table

Data Table Paint Data Data Info Rank

T Color Feature Statistics Save Data

Transform Visualize Model Evaluate Unsupervised

Box Plot

Visualize the distribution of feature values in a box plot.

more...

Load data on Iris Any change in

Variable Filter... iris sepal length sepal width petal length petal width

Subgroups Filter... None iris

Order by relevance to subgroups

Order by relevance to variable

Display Annotate No comparison Compare medians Compare means

Box Plot - Orange

Iris-versicolor:  $5.9 \pm 0.00$

Iris-virginica:  $6.642 \pm 0.59$

At least one group has just one instance, cannot compute significance

This workflow works best if you have Tree Viewer, Scatter Plot and Box Plot all open at the same time.

propagates to all the downstream widgets in the workflow.

Untitled - Orange

File Edit View Widget Options Help

Data

File CSV File Import Datasets SQL Table

Data Table Paint Data Data Info Rank

T Color Feature Statistics Save Data

Transform Visualize Model Evaluate Unsupervised

File Read data from an input file or network and send a data table to the output.

more...

Choose class-labeled dataset. Say, "iris.tab" from documentation datasets.

Data Table

It's always a good idea to check out the data first.

File

Several learners can be scored in cross-validation at the same time.

Logistic Regression Random Forest Classification SVM

Learner Learner Learner

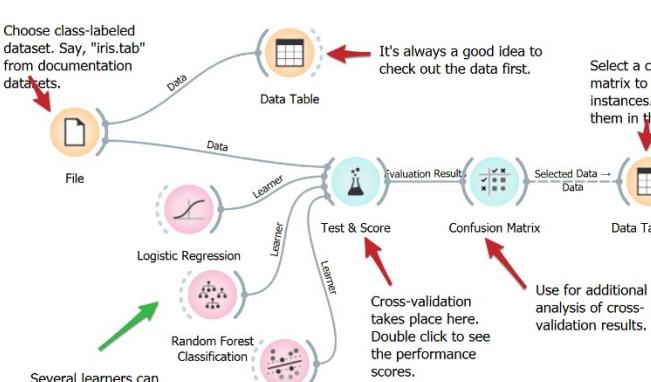
Evaluation Results Test & Score Confusion Matrix

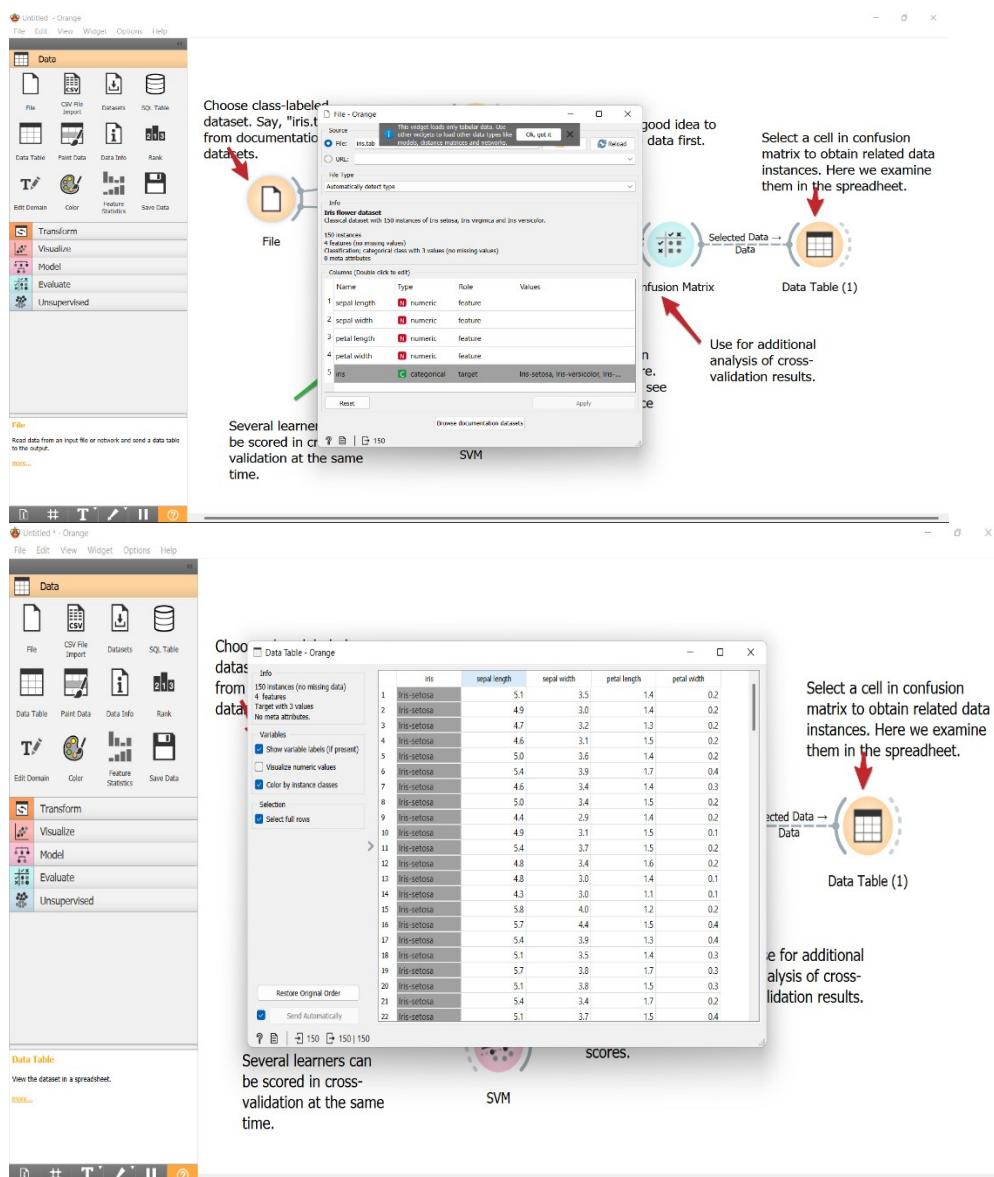
Selected Data Data Table (1)

Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.

Cross-validation takes place here. Double click to see the performance scores.

Use for additional analysis of cross-validation results.





**Choose class-labeled dataset.** Say, "Iris.t" from documentation datasets.

good idea to data first.

Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.

Selected Data → Data Table (1)

Use for additional analysis of cross-validation results.

**Several learners can be scored in cross-validation at the same time.**

**SVM**

**Choose data from data.**

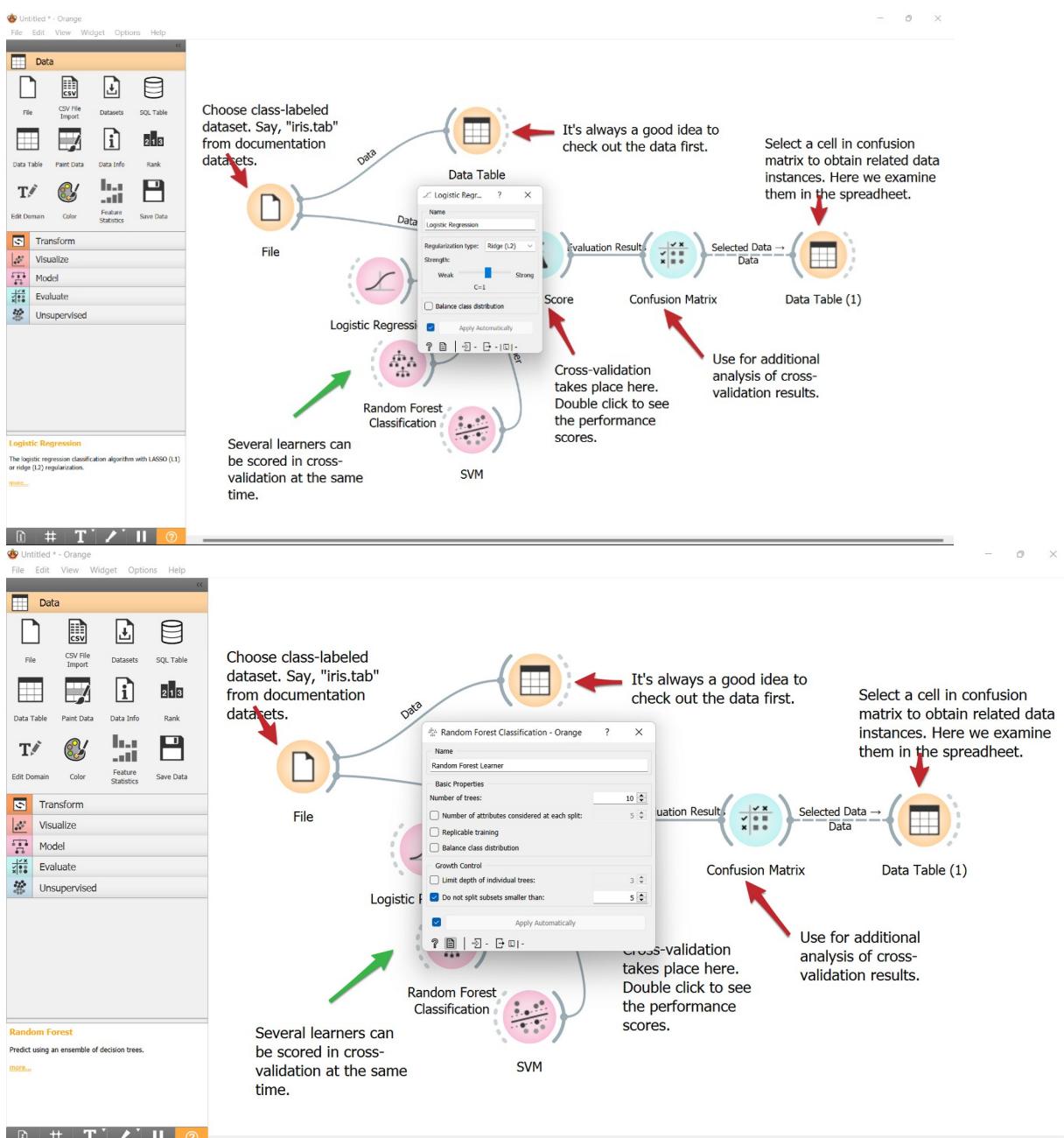
Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.

Selected Data → Data Table (1)

Use for additional analysis of cross-validation results.

**Scores.**

**SVM**



The screenshot displays two separate workflows in the Orange data mining software:

**Top Workflow (Logistic Regression):**

- Start:** Choose class-labeled dataset, say, "iris.tab" from documentation datasets.
- Process:** Data → Data Table (Logistic Regression) → Evaluation Result → Selected Data → Data Table (1).
- Widgets:** Data Table, File, Transform, Visualize, Model, Evaluate, Unsupervised.
- Notes:**
  - It's always a good idea to check out the data first.
  - Several learners can be scored in cross-validation at the same time.
  - Cross-validation takes place here. Double click to see the performance scores.
  - Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.
  - Use for additional analysis of cross-validation results.

**Bottom Workflow (Random Forest):**

- Start:** Choose class-labeled dataset, say, "iris.tab" from documentation datasets.
- Process:** Data → Data Table (Random Forest Classification) → Evaluation Result → Selected Data → Data Table (1).
- Widgets:** Data Table, File, Transform, Visualize, Model, Evaluate, Unsupervised.
- Notes:**
  - It's always a good idea to check out the data first.
  - Several learners can be scored in cross-validation at the same time.
  - Cross-validation takes place here. Double click to see the performance scores.
  - Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.
  - Use for additional analysis of cross-validation results.

Untitled \* - Orange

File Edit View Widget Options Help

Data

- File
- CSV File Import
- Datasets
- SQL Table
- Data Table
- Paint Data
- Data Info
- Rank
- Edit Domain
- Color
- Feature Statistics
- Save Data

Transform

Visualize

Model

Evaluate

Unsupervised

SVM

Support Vector Machines map inputs to higher-dimensional feature spaces.

more...

File

Choose class-labeled dataset. Say, "iris.tab" from documentation datasets.

Logistic Regression

Random Classification

SVM

Several learners can be scored in cross-validation at the same time.

It's always a good idea to check out the data first.

Evaluation Results

Confusion Matrix

Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.

Selected Data → Data

Data Table (1)

Use for additional analysis of cross-validation results.

Test & Score - Orange

Choose class-labeled dataset. Say, "iris.tab" from documentation datasets.

It's always a good idea to check out the data first.

Cross validation

Number of folds: 10

Stratified

Cross validation by feature

Random sampling

Repeat train/test: 10

Training set size: 66 %

Stratified

Leave one out

Test on train data

Test on test data

Classification

SVM

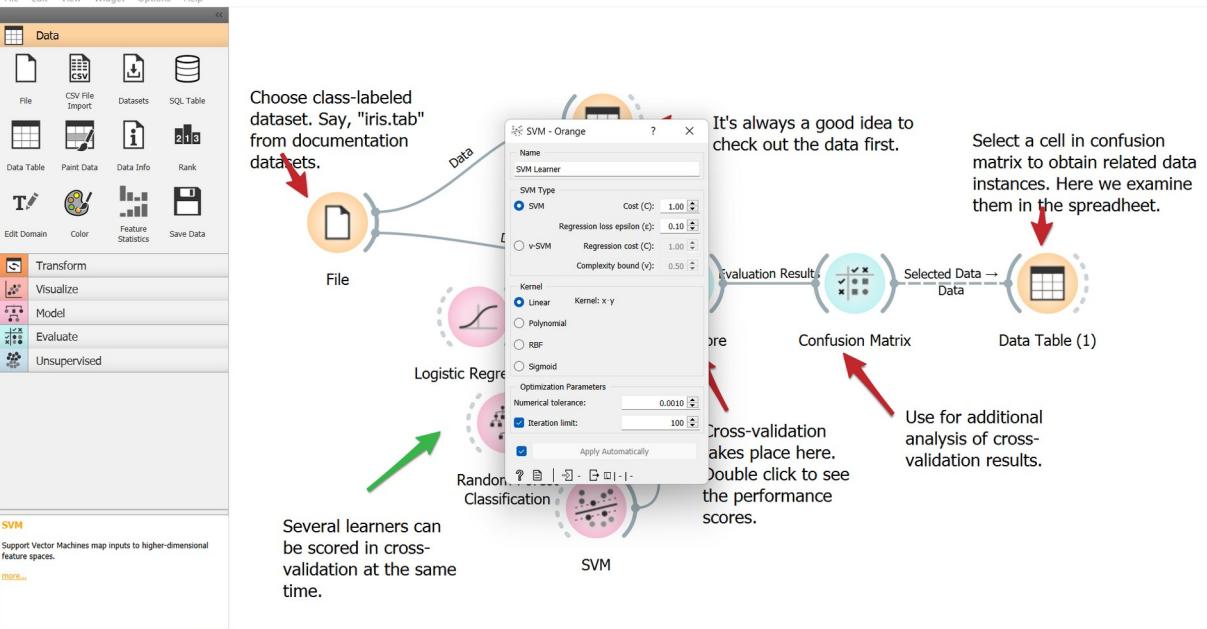
Several learners can be scored in cross-validation at the same time.

the performance scores.

Selected Data → Data

Data Table (1)

Use for additional analysis of cross-validation results.



Model	AUC	CA	F1	Precision	Recall
SVM Learner	0.996	0.947	0.947	0.947	0.947
Random Forest Learner	0.983	0.953	0.953	0.953	0.953
Logistic Regression	0.908	0.667	0.667	0.667	0.667

Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.



# Charotar University of Science and Technology

## Devang Patel Institute of Advance Technology and Research

### Department of Computer Engineering



**Untitled - Orange**

File Edit View Widget Options Help

Data

- File
- CSV File Import
- Datasets
- SQL Table
- Data Table
- Paint Data
- Data Info
- Rank
- Edit Domain
- Color
- Feature Statistics
- Save Data
- Transform
- Visualize
- Model
- Evaluate
- Unsupervised

Choose class-labeled dataset. See "iris.tab".

frc da

Learners

- Logistic Regression
- Random Forest Learner
- SVM Learner

Output

- Predictions
- Probabilities
- Apply Automatically

Clicking on cells or in headers outputs the corresponding data instances

Ok, got it

Show: Number of instances

Predicted

Actual	Iris-setosa	Iris-versicolor	Iris-virginica	$\Sigma$
Iris-setosa	50	0	0	50
Iris-versicolor	0	47	3	50
Iris-virginica	0	2	48	50
$\Sigma$	50	49	51	150

Select Correct Select Misclassified Clear Selection

3x150 150 can be scored in cross-validation at the same time.

SVM

Matrix

Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.

Selected Data → Data

Data Table (1)

Use for additional analysis of cross-validation results.

**Untitled - Orange**

File Edit View Widget Options Help

Data

- File
- CSV File Import
- Datasets
- SQL Table
- Data Table
- Paint Data
- Data Info
- Rank
- Edit Domain
- Color
- Feature Statistics
- Save Data
- Transform
- Visualize
- Model
- Evaluate
- Unsupervised

Choose data from data

Data Table (1) - Orange

Info

2 instances (no missing data)  
4 features  
Target with 3 values  
1 meta attribute

Variables

- Show variable labels (if present)
- Visualize numeric values
- Color by instance classes

Selection

- Select full rows

Restore Original Order

Send Automatically

iris Iris/Logistic Regressor sepal length sepal width petal length petal width

1	iris-virginica	Iris-versicolor	4.9	2.5	4.5
2	iris-virginica	Iris-versicolor	6.0	2.2	5.0

Several learners can be scored in cross-validation at the same time.

SVM

Matrix

Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadsheet.

Selected Data → Data

Data Table (1)

Use for additional analysis of cross-validation results.

Untitled - Orange

File Edit View Widget Options Help

**Data**

- File
- CSV File Import
- Datasets
- SQL Table
- Data Table
- Paint Data
- Data Info
- Rank
- Edit Domain
- Color
- Feature Statistics
- Save Data

**Transform**

**Visualize**

**Model**

**Evaluate**

**Unsupervised**

Read the data. Try this schema with the "brown-selected" data (from datasets that come with Orange). 

the data is in a heat

Choose any part of the clustering dendrogram in Hierarchical Clustering. Then, observe the selected data in a data table, or in any other analysis widget. Open both Hierarchical Clustering and Data Table (1) widget to turn this schema into interactive data analysis.

Compute the distances between the data samples

Selected Data Data

Data Table (1)

Box Plot

Any change in selection in hierarchical clustering will propagate to the Data Table and Box Plot widgets.

File - Orange

Source: File: brown-selected.tab

File Type: Automatically detect type

Info: Brown dataset  
Gene expression of baker's yeast.  
186 instances  
79 features (1.5% missing values)  
Classification: categorical class with 3 values (no missing values)  
1 meta attribute

Columns (Double click to edit)

Name	Type	Role	Values
alpha 0	numeric	feature	
alpha 7	numeric	feature	
alpha 14	numeric	feature	
alpha 21	numeric	feature	
alpha 28	numeric	feature	

Reset Apply

Browse documentation datasets

?

186

**Data Table - Orange**

Info: 186 instances, 79 features (1.5 % missing data), Target with 3 values, 1 meta attribute

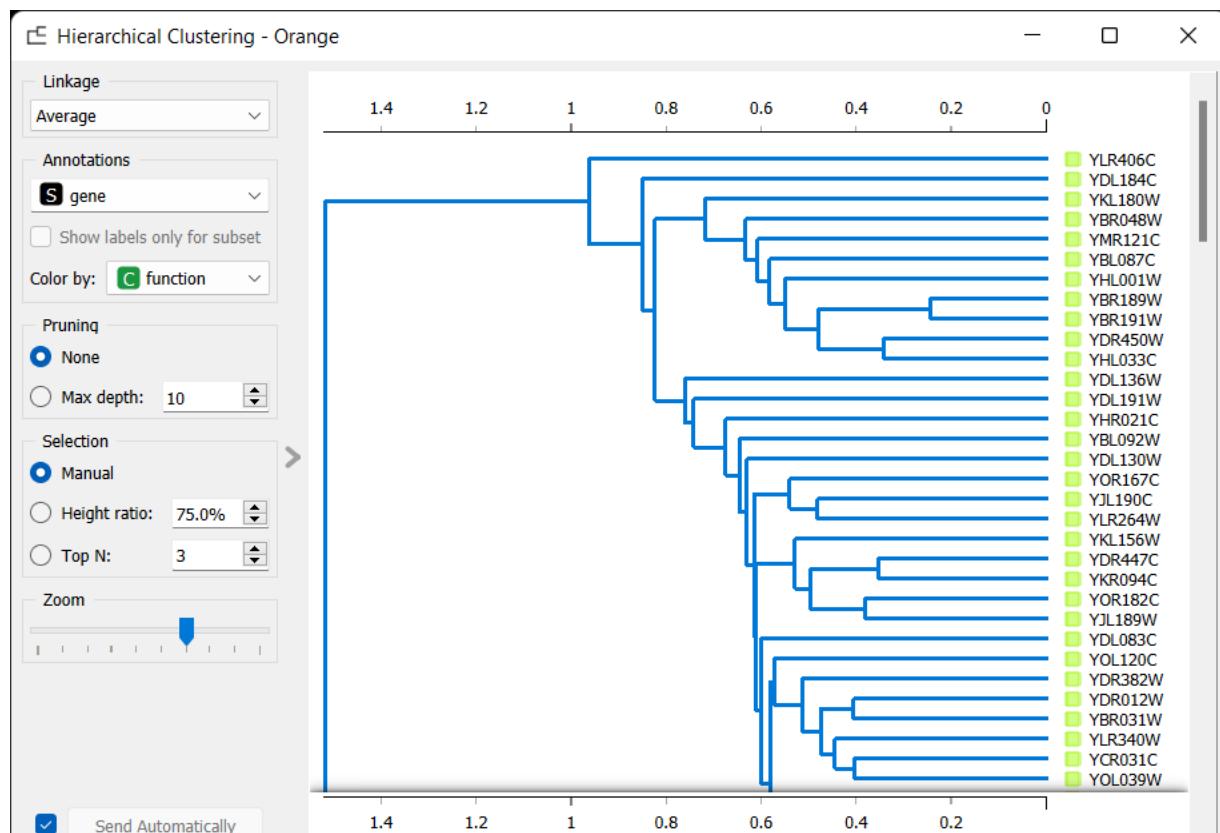
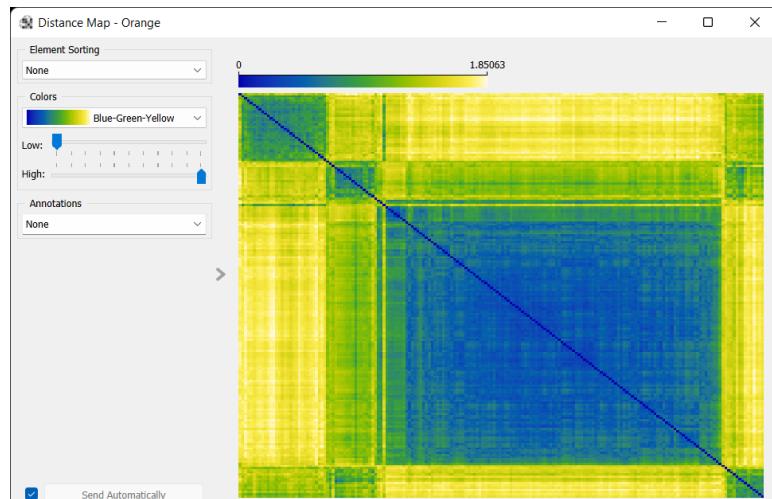
Variables: Show variable labels (if present), Visualize numeric values, Color by instance classes

Selection: Select full rows

Restore Original Order

Send Automatically

	function	gene	alpha 0	alpha 7	alpha 14	alpha 21
1	Proteas	YGR270W	?	-0.023	0.057	
2	Proteas	YIL075C	-0.031	-0.031	-0.060	
3	Proteas	YDL007W	-0.013	?	0.067	
4	Proteas	YER094C	0.003	0.025	0.067	
5	Proteas	YFR004W	-0.068	-0.003	-0.041	
6	Proteas	YDR427W	-0.012	-0.009	-0.009	
7	Proteas	YKL145W	0.012	0.008	-0.006	
8	Proteas	YGL048C	0.067	-0.064	0.011	
9	Proteas	YFR050C	0.093	0.027	0.044	
10	Proteas	YDL097C	0.062	0.002	0.050	
11	Proteas	YOR259C	-0.037	-0.122	0.030	
12	Proteas	YPR108W	-0.016	-0.051	0.073	
13	Proteas	YER021W	0.012	0.008	0.043	
14	Proteas	YGR253C	-0.053	0.167	-0.072	
15	Proteas	YGL011C	0.011	-0.017	0.045	
16	Proteas	YMR314W	-0.022	-0.048	-0.041	
17	Proteas	YGR135W	-0.002	-0.009	-0.022	
18	Proteas	YER012W	0.045	0.041	0.056	
19	Proteas	YPR103W	-0.002	-0.048	0.017	
20	Proteas	YJL001W	0.014	0.002	-0.009	
21	Proteas	YOR362C	-0.042	0.062	-0.030	

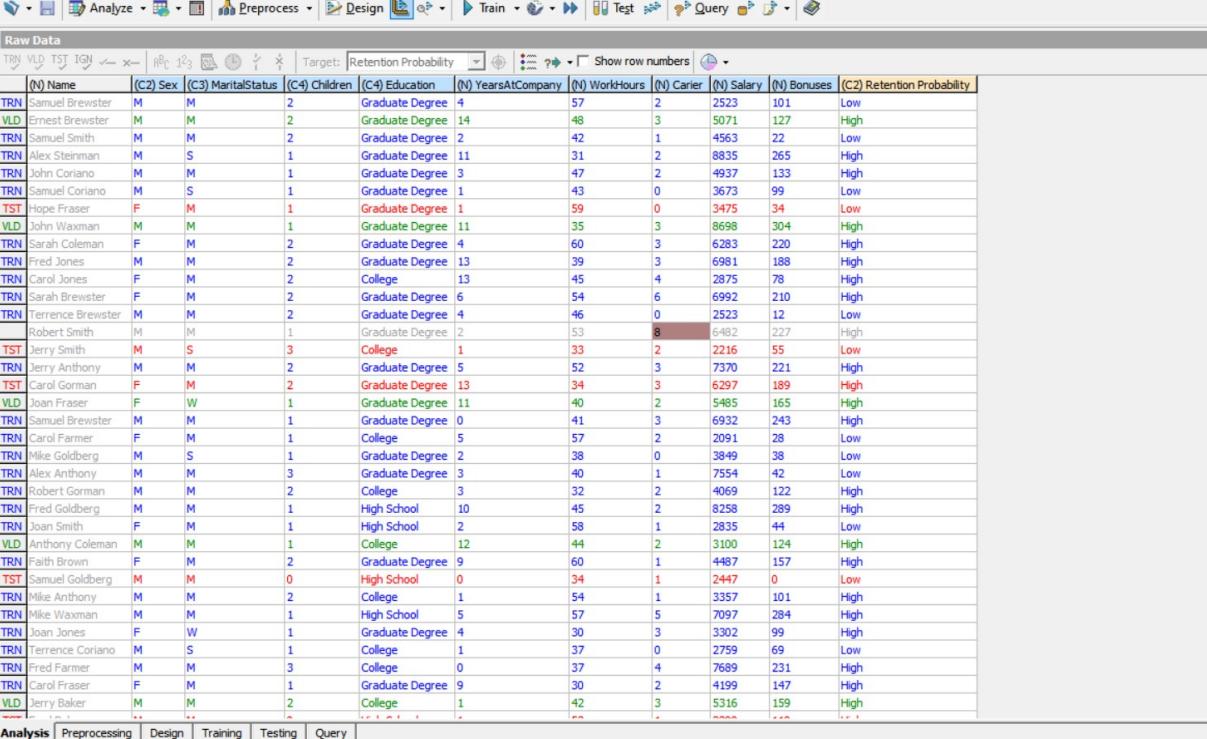


**Conclusion:** In this practical we performed classification on Iris dataset using neural network tools such as WEKA, ORANGE.

## 6.2: Show the uses of NEUROINTELLIGENCE to make prediction for faculty retention ratio.

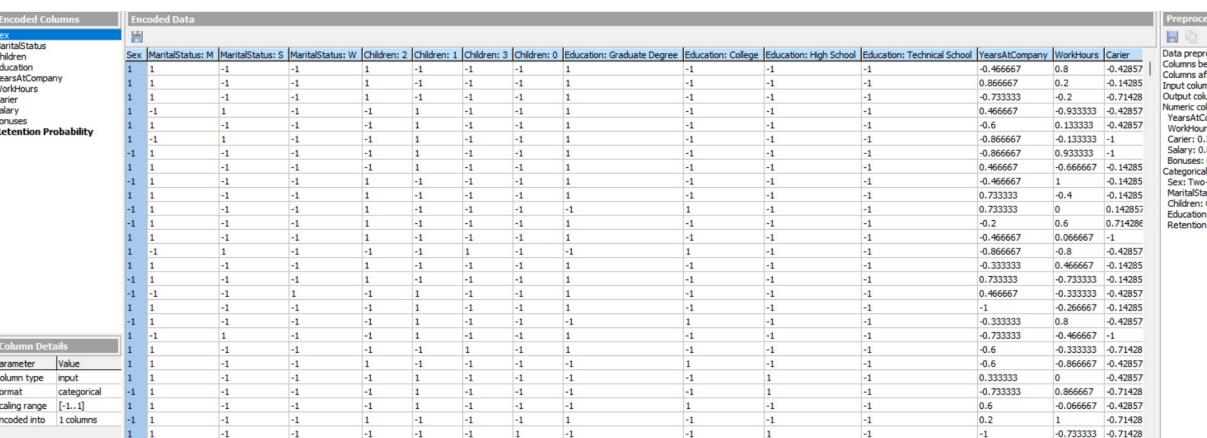
### Output:

#### Analysis:



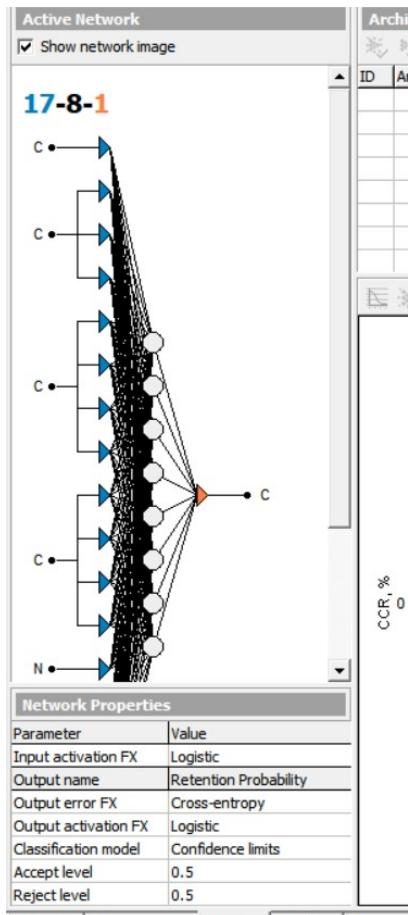
The screenshot shows the Orange data mining interface. The top menu bar includes options like Analyze, Preprocess, Design, Train, Test, Query, etc. Below the menu is a toolbar with various icons. The main area displays a data table titled 'Raw Data'. The table has columns: (N) Name, (C2) Sex, (C3) MaritalStatus, (C4) Children, (C4) Education, (N) YearsAtCompany, (N) WorkHours, (N) Carier, (N) Salary, (N) Bonuses, and (C2) Retention Probability. The data rows represent various employees with their details and retention status. At the bottom of the data table, there are buttons for Analysis, Preprocessing, Design, Training, Testing, and Query.

### Pre-processing:

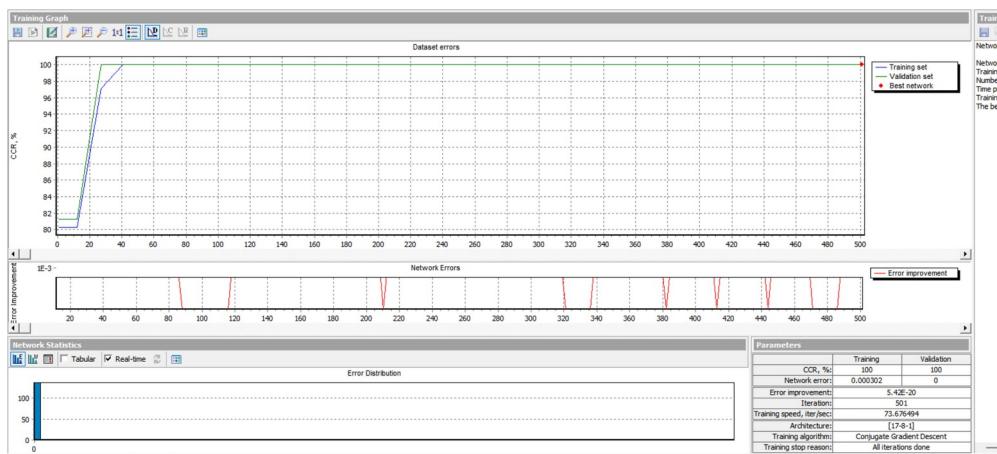


The screenshot shows the 'Preprocess' tab of the Orange interface. It displays a large matrix of encoded data corresponding to the 'Raw Data' table. The columns are labeled with various features and their encoded values. To the left, there is a 'Column Details' panel showing parameters like 'Parameter' and 'Value' for each column. The columns include Sex, MaritalStatus, Children, Education, YearsAtCompany, WorkHours, and Carier. The matrix contains binary values (0 or 1) representing the encoded state of each feature for each data point.

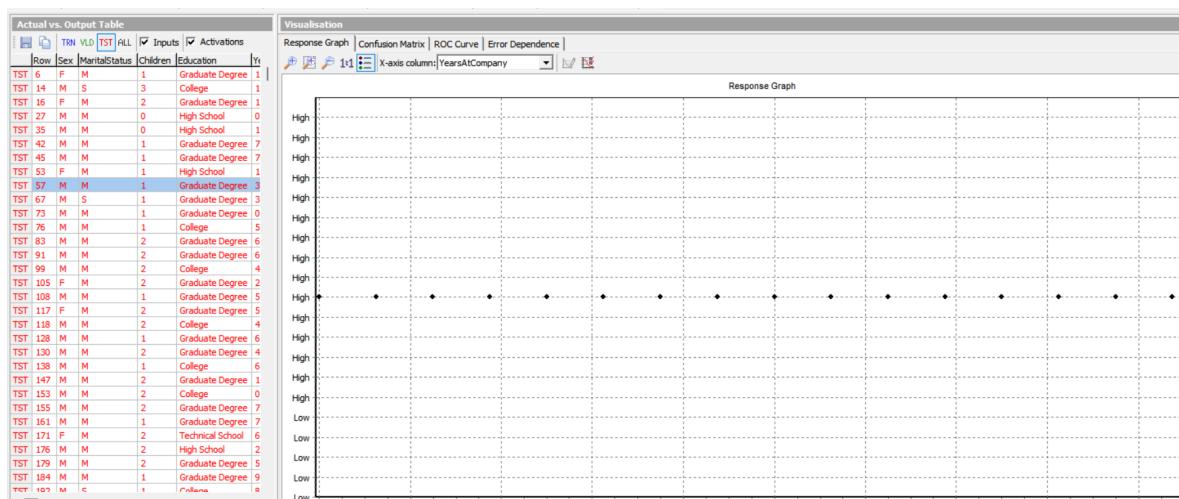
**Design:**



**Training:**



### Testing:



### Query:

Manual Query

Sex	MaritalStatus	Children	Education	YearsAtCompany	WorkHours	Carier	Salary	Bonuses
S	0	High Schoo	1	35	1	7000	0	
max: n/a	max: n/a	max: n/a	max: n/a	max: 15	max: 60	max: 7	max: 8975	max: 354
min: n/a	min: n/a	min: n/a	min: n/a	min: 0	min: 30	min: 0	min: 2004	min: 0

Results Table

Sex	MaritalStatus	Children	Education	YearsAtCompany	WorkHours	Carier	Salary	Bonuses	* Retention Probability
F	S	0	High Schoo	1	35	1	7000	0	Low
M	M	0	Graduate	5	10	4	4500	135	High
F	W	1	Graduate	12	45	7	5000	30	High
M	S	1	Graduate	12	45	7	2006	30	High

**Conclusion:** In this practical we learned about NEUROINTELLIGENCE tool and how to work on it.



## **Practical –7**

### **Game Playing**

**7.1: Apply MINIMAX Algorithm to solve Tic-Tac-Toe game in python.  
Design your solution using alpha-beta pruning.**

#### **Code:**

```
from random import choice
from math import inf
board = [[0, 0, 0],
          [0, 0, 0],
          [0, 0, 0]]
def Gameboard(board):
    chars = {1: 'X', -1: 'O', 0: ' '}
    for x in board:
        for y in x:
            ch = chars[y]
            print(f'| {ch} |', end="")
        print('\n' + '-----')
        print('=====')
def Clearboard(board):
    for x, row in enumerate(board):
        for y, col in enumerate(row):
            board[x][y] = 0
def winningPlayer(board, player):
    conditions = [[board[0][0], board[0][1], board[0][2]],
                  [board[1][0], board[1][1], board[1][2]],
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
[board[2][0], board[2][1], board[2][2]],  
[board[0][0], board[1][0], board[2][0]],  
[board[0][1], board[1][1], board[2][1]],  
[board[0][2], board[1][2], board[2][2]],  
[board[0][0], board[1][1], board[2][2]],  
[board[0][2], board[1][1], board[2][0]]]
```

```
if [player, player, player] in conditions:  
    return True  
return False  
  
def gameWon(board):  
    return winningPlayer(board, 1) or winningPlayer(board, -1)  
  
def printResult(board):  
    if winningPlayer(board, 1):  
        print('X has won! ' + '\n')  
    elif winningPlayer(board, -1):  
        print('O\'s have won! ' + '\n')  
    else:  
        print('Draw' + '\n')  
  
def blanks(board):  
    blank = []  
    for x, row in enumerate(board):  
        for y, col in enumerate(row):  
            if board[x][y] == 0:  
                blank.append([x, y])  
    return blank
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
def boardFull(board):
    if len(blanks(board)) == 0:
        return True
    return False

def setMove(board, x, y, player):
    board[x][y] = player

def playerMove(board):
    e = True
    moves = {1: [0, 0], 2: [0, 1], 3: [0, 2],
             4: [1, 0], 5: [1, 1], 6: [1, 2],
             7: [2, 0], 8: [2, 1], 9: [2, 2]}
    while e:
        try:
            move = int(input('Enter a number between 1-9: '))
            if move < 1 or move > 9:
                print('Invalid Move! Try again!')
            elif not (moves[move] in blanks(board)):
                print('Invalid Move! Try again!')
            else:
                setMove(board, moves[move][0], moves[move][1], 1)
                Gameboard(board)
                e = False
        except(KeyError, ValueError):
            print('Enter a number!')
def getScore(board):
    if winningPlayer(board, 1):
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
return 10

elif winningPlayer(board, -1):
    return -10

else:
    return 0

def abminimax(board, depth, alpha, beta, player):
    row = -1
    col = -1
    if depth == 0 or gameWon(board):
        return [row, col, getScore(board)]
    else:
        for cell in blanks(board):
            setMove(board, cell[0], cell[1], player)
            score = abminimax(board, depth - 1, alpha, beta, -player)
            if player == 1:
                # X is always the max player
                if score[2] > alpha:
                    alpha = score[2]
                    row = cell[0]
                    col = cell[1]
            else:
                if score[2] < beta:
                    beta = score[2]
                    row = cell[0]
                    col = cell[1]
    setMove(board, cell[0], cell[1], 0)
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



```
if alpha >= beta:  
    break  
if player == 1:  
    return [row, col, alpha]  
else:  
    return [row, col, beta]  
def o_comp(board):  
    if len(blanks(board)) == 9:  
        x = choice([0, 1, 2])  
        y = choice([0, 1, 2])  
        setMove(board, x, y, -1)  
        Gameboard(board)  
    else:  
        result = abminimax(board, len(blanks(board)), -inf, inf, -1)  
        setMove(board, result[0], result[1], -1)  
        Gameboard(board)  
  
def x_comp(board):  
    if len(blanks(board)) == 9:  
        x = choice([0, 1, 2])  
        y = choice([0, 1, 2])  
        setMove(board, x, y, 1)  
        Gameboard(board)  
    else:  
        result = abminimax(board, len(blanks(board)), -inf, inf, 1)  
        setMove(board, result[0], result[1], 1)
```



**Charotar University of Science  
and Technology**  
**Devang Patel Institute of Advance  
Technology and Research**  
**Department of Computer Engineering**



Gameboard(board)

```
def makeMove(board, player, mode):
```

```
    if mode == 1:
```

```
        if player == 1:
```

```
            playerMove(board)
```

```
        else:
```

```
            o_comp(board)
```

```
    else:
```

```
        if player == 1:
```

```
            o_comp(board)
```

```
        else:
```

```
            x_comp(board)
```

```
def pvc():
```

```
    while True:
```

```
        try:
```

```
            order = int(input('Enter to play 1st or 2nd: '))
```

```
            if not (order == 1 or order == 2):
```

```
                print('Please pick 1 or 2')
```

```
            else:
```

```
                break
```

```
        except(KeyError, ValueError):
```

```
            print('Enter a number')
```

Clearboard(board)

```
if order == 2:
```

```
    currentPlayer = -1
```

else:

```

    currentPlayer = 1

    while not (boardFull(board) or gameWon(board)):

        makeMove(board, currentPlayer, 1)

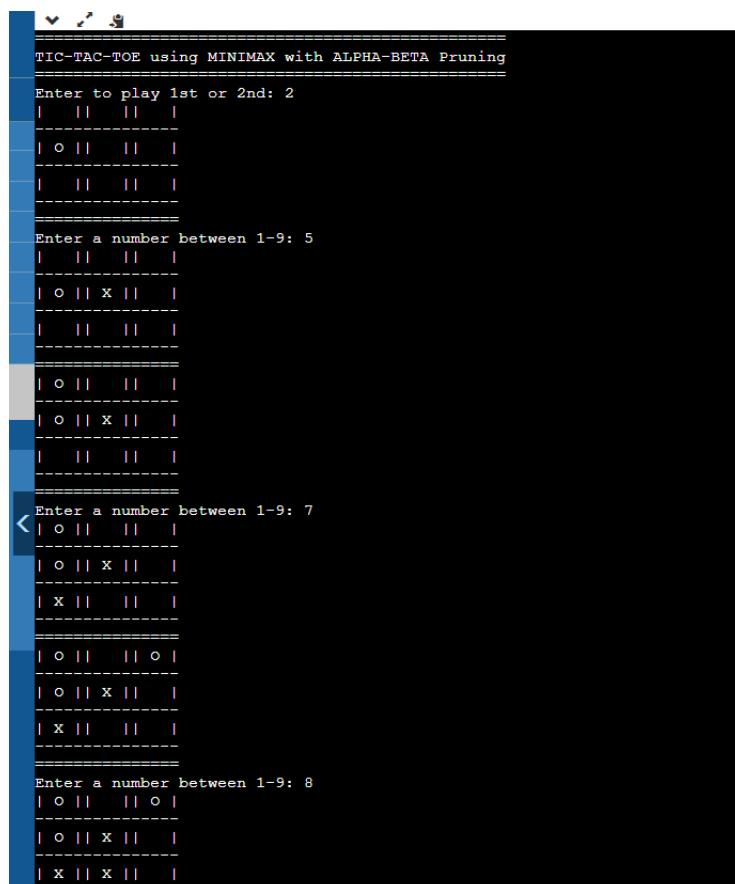
        currentPlayer *= -1

        printResult(board)

    print("=====")
    print("TIC-TAC-TOE using MINIMAX with ALPHA-BETA Pruning")
    print("=====")
    pvc()

```

**Output:**



```

=====
TIC-TAC-TOE using MINIMAX with ALPHA-BETA Pruning

Enter to play 1st or 2nd: 2
| | | |
| o | | |
| | | |
=====

Enter a number between 1-9: 5
| | | |
| o | | |
| | | |
=====

| o | | |
| o | | x |
| | | |
=====

Enter a number between 1-9: 7
<| o | | |
| o | | x |
| x | | |
=====

| o | | o |
| o | | x |
| x | | |
=====

Enter a number between 1-9: 8
| o | | o |
| o | | x |
| x | | x |
=====

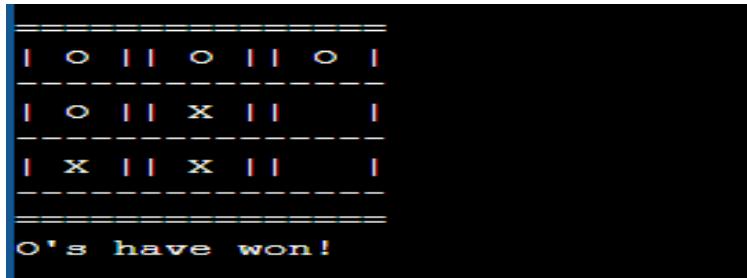

```



# Charotar University of Science and Technology

Devang Patel Institute of Advance  
Technology and Research

Department of Computer Engineering



**Conclusion:** In this practical we learned how to Apply MINIMAX Algorithm to solve Tic-Tac-Toe game in python.

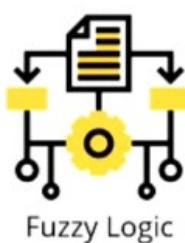
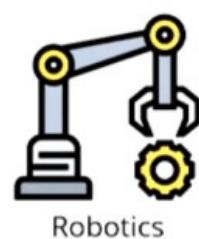
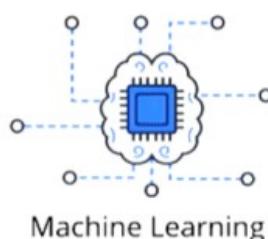
## Practical –8

### Game Playing

#### 8.1: Presentation on AI Domains, Applications, Problems and its Explanation

##### Domains of AI:

- Machine Learning
- Deep Learning
- Robotics
- Expert systems
- Fuzzy Logic



**Domains  
of  
Artificial Intelligence**

##### Machine Learning:

Machine Learning is basically the science of getting machines to interpret processes and analyze the data to solve real world problems. Machine learning supervised and unsupervised in enforcement learning.



### **Deep Learning:**

Deep Learning is the process of implementing neural networks on high dimensional data to gain insights and form solutions, basically logic behind in the face verification algorithms on Facebook it is the logic behind self-driving cars, virtual assistant and Alexa.

Then we have natural language processing first to science to draw insights from natural human languages and grow the businesses.

Example: Twitter and Amazon

Twitter uses NLP to filter out terrorizing words in the twitter

Amazon uses NLP to understand the consumer behavior and user experiences

### **Robotics:**

Robotics is a branch of artificial intelligence which focuses on different branches and applications of robots. AI robots are artificial environments to produce results by taking some accountable actions.

Example: Sofia robot

### **Expert Systems:**

Expert systems are AI based computer systems that learn and reciprocate the decision-making ability of human experts. Expert systems use IF/Then logics mentioned in order to solve any complex problems. They do not relay conventional proceeding programming.

Expert systems mainly used in information management seem to use virus detection, medical and hospital records and so on.

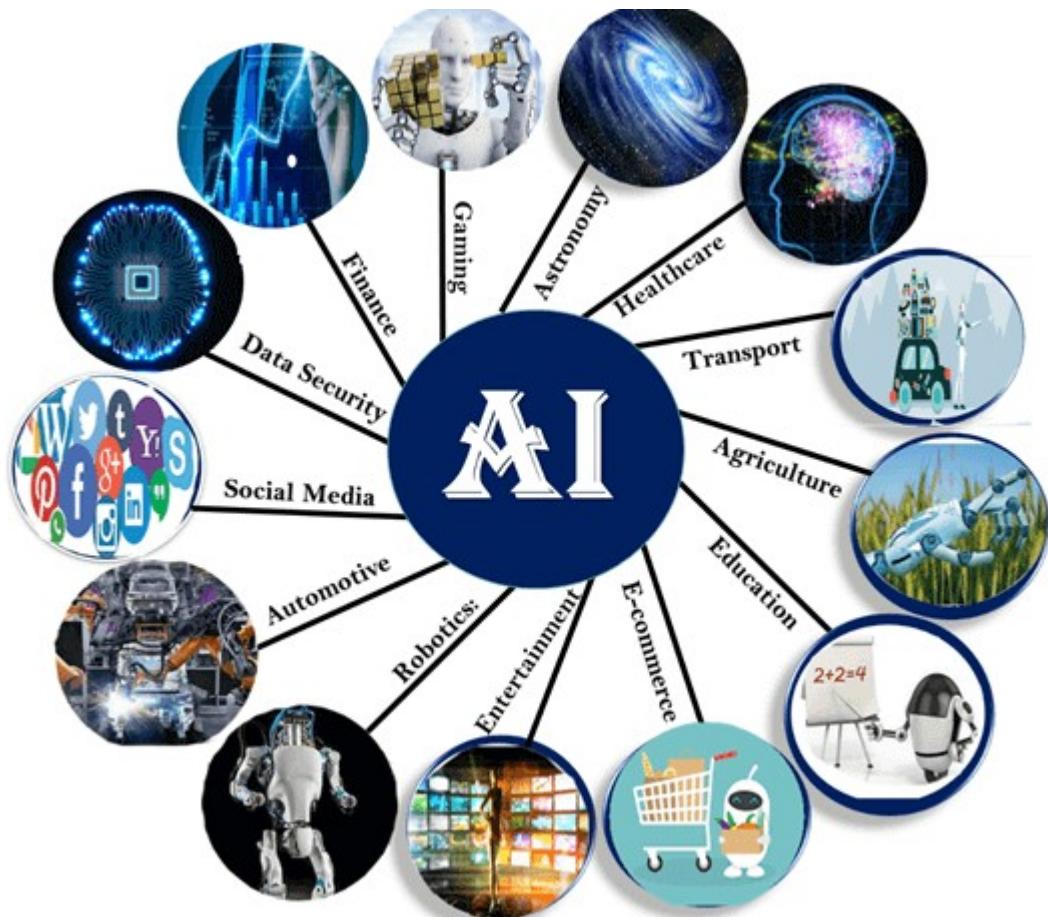
### **Fuzzy Logic:**

Fuzzy logic is computing approach that is based on principles degree of truth instead of usual modern logic that we used which is basically Boolean logic. Fuzzy logic used in the medical field to solve complex problems which involve decision making is also used in automated gas systems in cars.

## Application of AI

Artificial Intelligence has various applications in today's society. It is becoming essential for today's time because it can solve complex problems with an efficient way in multiple industries, such as Healthcare, entertainment, finance, education, etc. AI is making our daily life more comfortable and fast.

Following are some sectors which have the application of Artificial Intelligence:





# **Charotar University of Science and Technology**

**Devang Patel Institute of Advance  
Technology and Research**

**Department of Computer Engineering**



## **1. AI in Astronomy**

- Artificial Intelligence can be very useful to solve complex universe problems. AI technology can be helpful for understanding the universe such as how it works, origin, etc.

## **2. AI in Healthcare**

- In the last, five to ten years, AI becoming more advantageous for the healthcare industry and going to have a significant impact on this industry.
- Healthcare Industries are applying AI to make a better and faster diagnosis than humans. AI can help doctors with diagnoses and can inform when patients are worsening so that medical help can reach to the patient before hospitalization.

## **3. AI in Gaming**

- AI can be used for gaming purpose. The AI machines can play strategic games like chess, where the machine needs to think of a large number of possible places.

## **4. AI in Finance**

- AI and finance industries are the best matches for each other. The finance industry is implementing automation, chatbot, adaptive intelligence, algorithm trading, and machine learning into financial processes.

## **5. AI in Data Security**

- The security of data is crucial for every company and cyber-attacks are growing very rapidly in the digital world. AI can be used to make your data more safe and secure. Some examples such as AEG bot, AI2 Platform, are used to determine software bug and cyber-attacks in a better way.

## **6. AI in Social Media**

- Social Media sites such as Facebook, Twitter, and Snapchat contain billions of user profiles, which need to be stored and managed in a very efficient way. AI can organize and manage massive amounts of data. AI can analyze lots of data to identify the latest trends, hashtag, and requirement of different users.

## **7. AI in Travel & Transport**



# Charotar University of Science and Technology

Devang Patel Institute of Advance Technology and Research

Department of Computer Engineering



- AI is becoming highly demanding for travel industries. AI is capable of doing various travel related works such as from making travel arrangement to suggesting the hotels, flights, and best routes to the customers. Travel industries are using AI-powered chatbots which can make human-like interaction with customers for better and fast response.

## 8. AI in Automotive Industry

- Some Automotive industries are using AI to provide virtual assistant to their user for better performance. Such as Tesla has introduced TeslaBot, an intelligent virtual assistant.
- Various Industries are currently working for developing self-driven cars which can make your journey more safe and secure.

## 9. AI in Robotics:

- Artificial Intelligence has a remarkable role in Robotics. Usually, general robots are programmed such that they can perform some repetitive task, but with the help of AI, we can create intelligent robots which can perform tasks with their own experiences without pre-programmed.
- Humanoid Robots are best examples for AI in robotics, recently the intelligent Humanoid robot named as Erica and Sophia has been developed which can talk and behave like humans.

## 10. AI in Entertainment

- We are currently using some AI based applications in our daily life with some entertainment services such as Netflix or Amazon. With the help of ML/AI algorithms, these services show the recommendations for programs or shows.

## 11. AI in Agriculture

- Agriculture is an area which requires various resources, labor, money, and time for best result. Now a day's agriculture is becoming digital, and AI is emerging in this field. Agriculture is applying AI as agriculture robotics, solid and crop monitoring, predictive analysis. AI in agriculture can be very helpful for farmers.

## 12. AI in E-commerce

- AI is providing a competitive edge to the e-commerce industry, and it is becoming more demanding in the e-commerce business. AI is helping



shoppers to discover associated products with recommended size, color, or even brand.

### **13. AI in education:**

- AI can automate grading so that the tutor can have more time to teach. AI chatbot can communicate with students as a teaching assistant.
- AI in the future can be work as a personal virtual tutor for students, which will be accessible easily at any time and any place.

### **Top Common Challenges in AI**

There are several Artificial Intelligence problems, and we are going to address these challenges and how to solve them.

#### **1. Computing Power**

The amount of power these power-hungry algorithms use is a factor keeping most developers away. Machine Learning and Deep Learning are the stepping stones of this Artificial Intelligence, and they demand an ever-increasing number of cores and GPUs to work efficiently. There are various domains where we have ideas and knowledge to implement deep learning frameworks such as asteroid tracking, healthcare deployment, tracing of cosmic bodies, and much more.

They require a supercomputer's computing power, and yes, supercomputers aren't cheap. Although, due to the availability of Cloud Computing and parallel processing systems developers work on AI systems more effectively, they come at a price. Not everyone can afford that with an increase in the inflow of unprecedented amounts of data and rapidly increasing complex algorithms.



## 2. Trust Deficit

One of the most important factors that are a cause of worry for the AI is the unknown nature of how deep learning models predict the output. How a specific set of inputs can devise a solution for different kinds of problems is difficult to understand for a layman.

Many people in the world don't even know the use or existence of Artificial Intelligence, and how it is integrated into everyday items they interact with such as smartphones, Smart TVs, Banking, and even cars (at some level of automation).

**Must Read:** [Free nlp online course!](#)

## 3. Limited Knowledge

Although there are many places in the market where we can use Artificial Intelligence as a better alternative to the traditional systems. The real problem is the knowledge of Artificial Intelligence. Apart from technology enthusiasts, college students, and researchers, there are only a limited number of people who are aware of the potential of AI.

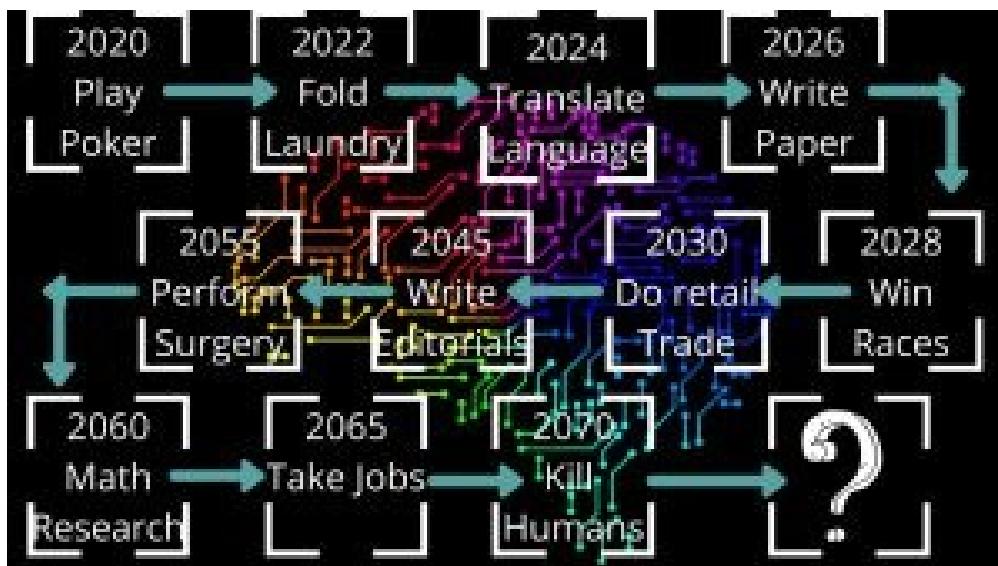
For example, there are many **SMEs (Small and Medium Enterprises)** which can have their work scheduled or learn innovative ways to increase their production, manage resources, sell and manage products online, learn and understand consumer behavior and react to the market effectively and efficiently. They are also not aware of service providers such as Google Cloud, Amazon Web Services, and others in the tech industry.

## 4. Human-level

This is one of the most important challenges in AI, one that has kept researchers on edge for AI services in companies and start-ups. These companies might be boasting of above 90% accuracy, but humans can do better in all of these scenarios. For example, let our model predict whether the image is of a dog or a cat. The human can predict the correct output nearly every time, mopping up a stunning accuracy of above 99%.

For a deep learning model to perform a similar performance would require unprecedented finetuning, hyperparameter optimization, large dataset, and a well-defined and accurate algorithm, along with robust computing power, uninterrupted training on train data and testing on test data. That sounds a lot of work, and it's actually a hundred times more difficult than it sounds.

One way you can avoid doing all the hard work is just by using a service provider, for they can train specific deep learning models using pre-trained models. They are trained on millions of images and are fine-tuned for maximum accuracy, but the real problem is that they continue to show errors and would really struggle to reach human-level performance.



## 5. Data Privacy and Security

The main factor on which all the deep and machine learning models are based on is the availability of data and resources to train them. Yes, we have data, but as this data is generated from millions of users around the globe, there are chances this data can be used for bad purposes.

For example, let us suppose a medical service provider offers services to 1 million people in a city, and due to a cyber-attack, the personal data of all the one million users fall in the hands



**Charotar University of Science  
and Technology**

**Devang Patel Institute of Advance  
Technology and Research**

**Department of Computer Engineering**



of everyone on the dark web. This data includes data about diseases, health problems, medical history, and much more. To make matters worse, we are now dealing with planet size data. With this much information pouring in from all directions, there would surely be some cases of data leakage.

Some companies have already started working innovatively to bypass these barriers. It trains the data on smart devices, and hence it is not sent back to the servers, only the trained model is sent back to the organization.

**Conclusion:** In this practical we did in depth analysis in AI.