LOVELY PROFESSIONAL UNIVERSITY

Artificial Intelligence Project Report

Of

Reasoning System For Weather

Name : Gajendra Gangwar(11810542)

Roll Number: 26

Section : K18KY

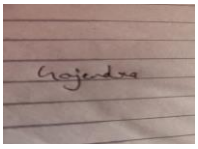
Name: M.Anish(11805869)

Roll Number: 47

Submitted To : Shabnam

Student Declaration

- This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be copied, we shall take full responsibility for it.



Gajendra gangwar

M.Anish

Place: Phagwara , Punjab

Date: 08/04/20**20**

Background and objectives

- The reasoning is the mental process of deriving logical conclusion and making predictions from available knowledge, facts, and beliefs. Or we can say, "Reasoning is a way to infer facts from existing data." It is a general process of thinking rationally, to find valid conclusions.

- In artificial intelligence, the reasoning is essential so that the machine can also think rationally as a human brain and can perform like a human.

- We have used deductive reasoning for the same, Deductive reasoning is deducing new information from logically related known information. It is the form of valid reasoning, which means the argument's conclusion must be true when the premises are true.

- Deductive reasoning is a type of propositional logic in AI, and it requires various rules and facts. It is sometimes referred to as top-down reasoning, and contradictory to inductive reasoning.

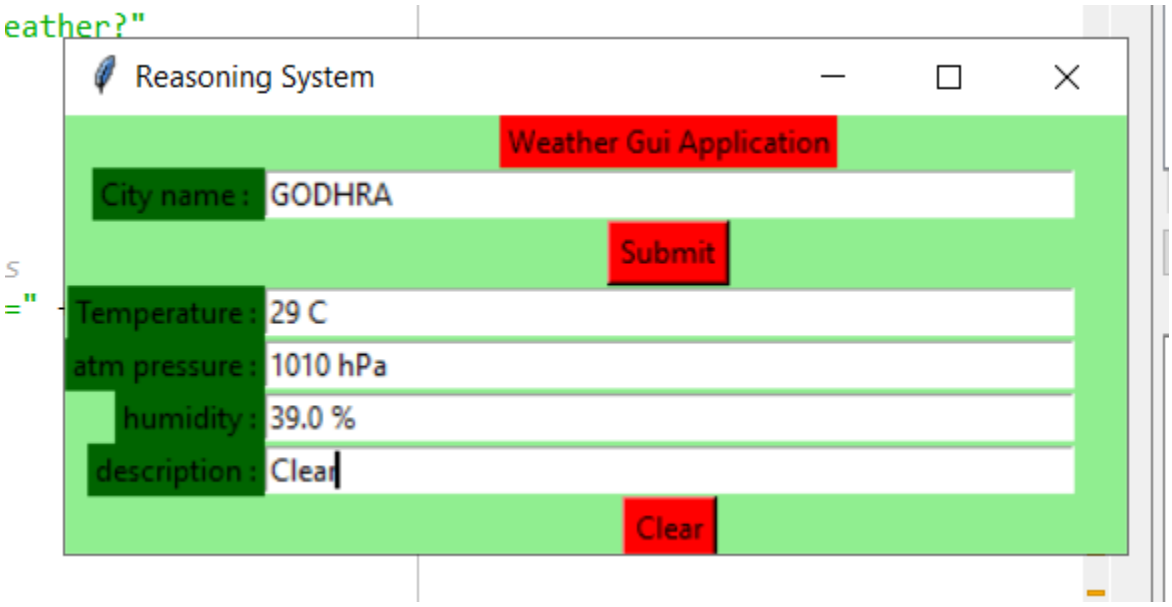
- In deductive reasoning, the truth of the premises guarantees the truth of the conclusion.
- **Objective:** To reason according to a set of logical and objective standards, while subjective thinking refers to reasoning without objective standards. Objective reasoning means reasoning that is independent of the specific subjective context, not influenced by personal characteristics, feelings or opinions of the subject.

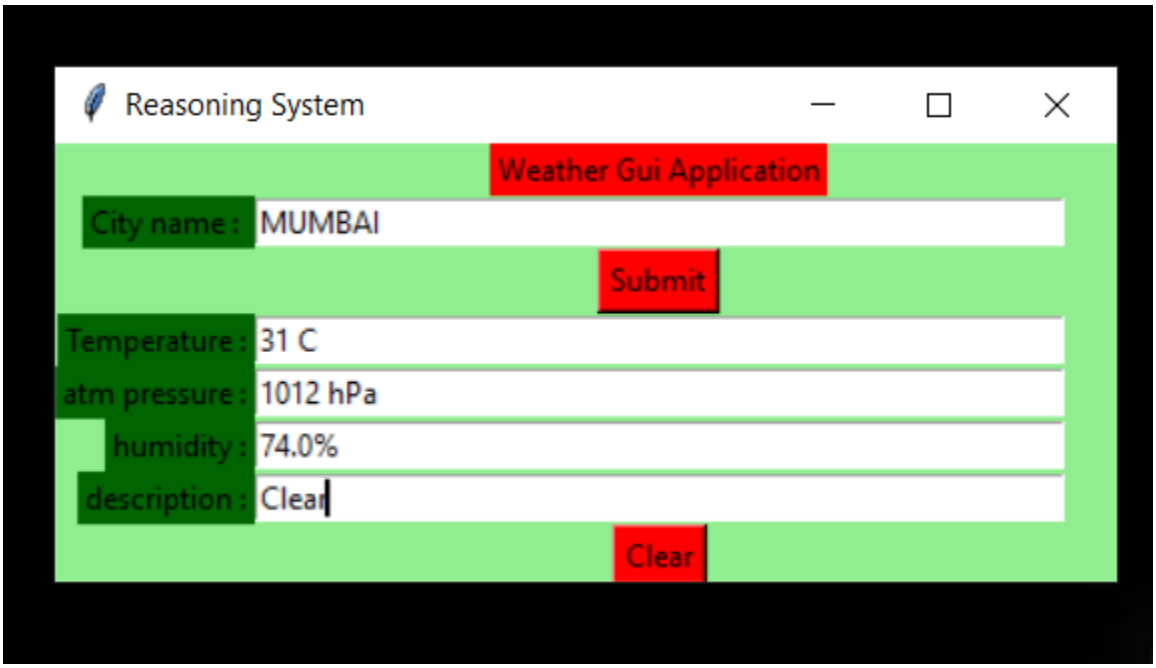
Description of project

- Reasoning system on weather which allows us to enter a city and then our reasoning system provides us about the temperature, pressure, humidity and also gives us a vague description about the weather.

- **MODULE USED:**
  GUI:

The Graphical user Interface (GUI) is a form of user interface that allows user to interact with electronic devices through graphical icons and audio indicators such as primary notations, instead of a text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces, which require commands to be typed on a computer keyboard.
The actions in the GUI is usually performed through direct manipulation of the graphical elements. Beyond computers, GUIs are used in many handled mobile devices such as MP3 players, portable media players, gaming devices , smartphones and smaller households , office and industrial controls.





Code

```
from tkinter import *
from tkinter import messagebox

# function to find weather details
# of any city using openweathermap api
def tell_weather() :

    # import required modules
    import requests, json
```

```
        # enter your api key here
        api_key = "f5fbc1231f407cb52d27e6d891272a5b"

        # base_url variable to store url
        base_url =
"http://api.openweathermap.org/data/2.5/weather?"


        # take a city name from city_field entry box
        city_name = city_field.get()

        # complete_url variable to store complete url address
        complete_url = base_url + "appid =" + api_key + "&q =" +
city_name

        # get method of requests module
        # return response object
        response = requests.get(complete_url)

        # json method of response object convert
        # json format data into python format data
        x = response.json()

        # now x contains list of nested dictionaries
        # we know dictionary contains key value pair
        # check the value of "cod" key is equal to "404"
        # or not if not that means city is found
        # otherwise city is not found
        if x["cod"] != "404" :
```

```
            # store the value of "main" key in variable y
            y = x["main"]

            # store the value corresponding to the "temp" key of y
            current_temperature = y["temp"]

            # store the value corresponding to the "pressure" key
of y
            current_pressure = y["pressure"]

            # store the value corresponding to the "humidity" key
of y
            current_humidiy = y["humidity"]

            # store the value of "weather" key in variable z
            z = x["weather"]

            # store the value corresponding to the "description"
key
            # at the 0th index of z
            weather_description = z[0]["description"]

            # insert method inserting the
            # value in the text entry box.
            temp_field.insert(15, str(current_temperature) + "
Kelvin")
            atm_field.insert(10, str(current_pressure) + " hPa")
            humid_field.insert(15, str(current_humidiy) + " %")
            desc_field.insert(10, str(weather_description) )
```

```
        # if city is not found
        else :

                # message dialog box appear which
                # shows given Error meassgae
                messagebox.showerror("Error", "City Not Found \n"
                                              "Please enter valid city
name")

                # clear the content of city_field entry box
                city_field.delete(0, END)


# Function for clearing the
# contents of all text entry boxes
def clear_all() :
        city_field.delete(0, END)
        temp_field.delete(0, END)
        atm_field.delete(0, END)
        humid_field.delete(0, END)
        desc_field.delete(0, END)

        # set focus on the city_field entry box
        city_field.focus_set()


# Driver code
if __name__ == "__main__”:

        # Create a GUI window
```

```
root = Tk()

# set the name of tkinter GUI window
root.title("Reasoning System")

# Set the background colour of GUI window
root.configure(background = "light green")

# Set the configuration of GUI window
root.geometry("425x175")

# Create a Weather Gui Application label
headlabel = Label(root, text = "Weather Gui Application",
                          fg = 'black', bg = 'red')

# Create a City name : label
label1 = Label(root, text = "City name : ",
                     fg = 'black', bg = 'dark green')

# Create a City name : label
label2 = Label(root, text = "Temperature :",
                     fg = 'black', bg = 'dark green')

# Create a atm pressure : label
label3 = Label(root, text = "atm pressure :",
                     fg = 'black', bg = 'dark green')

# Create a humidity : label
label4 = Label(root, text = "humidity :",
                     fg = 'black', bg = 'dark green')
```

```
# Create a description :label
label5 = Label(root, text = "description :",
                        fg = 'black', bg = 'dark green')


# grid method is used for placing
# the widgets at respective positions
# in table like structure .
headlabel.grid(row = 0, column = 1)
label1.grid(row = 1, column = 0, sticky ="E")
label2.grid(row = 3, column = 0, sticky ="E")
label3.grid(row = 4, column = 0, sticky ="E")
label4.grid(row = 5, column = 0, sticky ="E")
label5.grid(row = 6, column = 0, sticky ="E")


# Create a text entry box
# for filling or typing the information.
city_field = Entry(root)
temp_field = Entry(root)
atm_field = Entry(root)
humid_field = Entry(root)
desc_field = Entry(root)

# grid method is used for placing
# the widgets at respective positions
# in table like structure .
# ipadx keyword argument set width of entry space .
city_field.grid(row = 1, column = 1, ipadx ="100")
```

```
temp_field.grid(row = 3, column = 1, ipadx ="100")
atm_field.grid(row = 4, column = 1, ipadx ="100")
humid_field.grid(row = 5, column = 1, ipadx ="100")
desc_field.grid(row = 6, column = 1, ipadx ="100")

# Create a Submit Button and attached
# to tell_weather function
button1 = Button(root, text = "Submit", bg = "red",
                                fg = "black", command = tell_weather)

# Create a Clear Button and attached
# to clear_all function
button2 = Button(root, text = "Clear", bg = "red",
                                fg = "black", command = clear_all)

# grid method is used for placing
# the widgets at respective positions
# in table like structure .
button1.grid(row = 2, column = 1)
button2.grid(row = 7, column = 1)

# Start the GUI
root.mainloop()
```

Work Division

Gajendra Gangwar (11810542) – Research on reasoning system how it's works and analysis it. Worked on the implementation of code.

Gajendra Gangwar (11810542) – Worked on Implementation of code and GUI of code. Collect the information from the weather forecast website.

M.Anish (11805869) – Worked on the GUI of the reasoning system. Find the method that how to implement reasoning system on weather.

M.Anish (11805869) – helped in the collecting the information from the website.

BONAFIDE CERTIFICATE

Certified that this report “Reasoning System for weather” is the bonafide work of “Gajendra gangwar, M.Anish” who carried out the project work under supervision.

Gajendra