

# INT216 CAPSTRONE

- PROJECT ON

Project Title: WEATHER PREDICTION

Section: K21MD

Submitted by: **Gorlamari Srujan**

kumar Reddy

Reg. No: 12111003

Roll No: RK21MDB45

Under the Guidance of (WASEEM UD DIN WANI)  
School of Computer Science and Engineering



**L**OVELY  
**P**ROFESSIONAL  
**U**NIVERSITY

# ***DECLARATION***

We here by declare that the project work entitled(“weather prediction”) is an authentic record of our own work carried out as requirements of Capstone Project for the award of B. Tech degree in Python Project from Lovely Professional University, Phagwara, under the guidance of Waseem Ud Din Wani , during January to May 2023. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Name of Student: SRUJAN REDDY

Registration Number: 12111003

# ***CERTIFICATE***

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfilment of the conditions for the award of B.Tech degree in from Lovely Professional University, Phagwara.

**Signature and Name of the Mentor  
Designation**

School of Computer Science and Engineering,  
Lovely Professional University, Phagwara, Punjab.

**DATE:**

# TABLE OF CONTENTS

SNO	TITLE	PAGE NUMBER
1	DECLARATION	2
2	CERTIFICATE	3
3	ABSTRACT	5
4	INTRODUCTION	6
5	OBJECTIVE	7
6	DESCRIPTION	8
7	CODE	9
8	OUTPUT	14
9	SCOPE OF PROJECT	15
10	FUTURE DEVELOPMENT OF PROJECT	16
11	CONCLUSION	17

# ***ABSTRACT***

>This is a Python code that creates a graphical user interface (GUI) using the tkinter library to predict the weather for the future day based on the input data of yesterday's temperature and humidity and today's temperature and humidity.

>It then displays the predicted temperature and humidity for the future day, as well as some precautionary measures based on the predicted temperature.

>The program uses the Linear Regression algorithm from the scikit-learn library to train the model and make predictions. It also sets a background image for the GUI window.

>The program starts the GUI loop to display the GUI window and allow the user to input the weather data and get the predictions.

# ***INTRODUCTION***

>This code is a Python program using the Tkinter library that creates a GUI for a simple contact list application.

>It uses the tkinter library to create the GUI window and widgets such as labels, text entry fields, and buttons.

> The weather predictor function takes inputs for yesterday's and today's temperature and humidity and predicts the values for tomorrow using linear regression.

> It also provides suggestions for precautions based on the predicted temperature range.

>The GUI window has a background image and the widgets are placed in specific locations using grid() method.

>The script runs continuously in a loop until the user exits the GUI window.

>I am trying to create a weather predictor using a linear regression model.

# ***OBJECTIVE***

The objective of the code is to create a graphical user interface (GUI) for a weather predictor application. The user can input the temperature and humidity data for two consecutive days and then the application uses a linear regression model to predict the weather for the next day. The predicted temperature and humidity are displayed on the GUI along with a precaution message depending on the predicted temperature.

The GUI also has a background image and various labels and buttons to display and input data.

I can say simply the objectives of the code are:

- >To create a GUI that allows the user to input weather data for two days (yesterday and today).
- >To calculate the difference in weather data between the two days and add it to the weather data for day 2 to get the weather data for the future day.
- >To train a linear regression model using the input weather data and predict the weather for the future day.
- >To display the predicted weather data for the future day in the GUI.
- >To provide some recommendations/precautions to the user based on the predicted temperature

# ***DISCRIPTION***

>This is a Python script that creates a GUI window using the tkinter library. The GUI window is titled "Weather Predictor" and has a size of 2500x1665 pixels. The script defines a function called "predict\_weather" that predicts the weather for a future day based on the weather data for the previous two days provided by the user.

>The function calculates the difference between the weather data for the two days, adds that difference to the weather data for the second day, creates a dataset using the temperature and humidity data for the two days, trains a linear regression model on that dataset, and uses the model to predict the weather for the future day.

>The function also displays the predicted temperature and humidity for the future day, and a precaution message based on the predicted temperature.

>The script then creates the GUI widgets, including labels, text entry fields, and buttons, and places them in the GUI window using the grid method. When the user clicks the "Tomorrow Weather" button, the predict\_weather function is called and the predicted weather data is displayed in the GUI.

>Finally, the script starts the GUI loop, which waits for user input and responds to events such as button clicks.



# Code :

```
import tkinter as tk

from sklearn.linear_model import LinearRegression
from PIL import ImageTk, Image

# Create the GUI window

root = tk.Tk()
root.title("Weather Predictor")
root.geometry("1275x762+100+100")


# Define the function to predict the weather

def predict_weather():
    # Get the data from the user
    yesterday_temp = float(yesterday_temp_entry.get())
    yesterday_humidity = float(yesterday_humidity_entry.get())
    today_temp = float(today_temp_entry.get())
    today_humidity = float(today_humidity_entry.get())
```

**# Calculate the difference in weather data between the two days**

**temp\_diff = today\_temp - yesterday\_temp**

**humidity\_diff = today\_humidity - yesterday\_humidity**

**# Add the difference to the weather data for day 2 to get the weather data for the future day**

**tomorrow\_temp = today\_temp + temp\_diff**

**tomorrow\_humidity = today\_humidity + humidity\_diff**

**# Create the dataset**

**X = [[yesterday\_temp, yesterday\_humidity], [today\_temp, today\_humidity]]**

**y = [tomorrow\_temp, tomorrow\_humidity] # Example output, replace with actual values**

**# Train the linear regression model**

**model = LinearRegression()**

**model.fit(X, y)**

**# Predict the weather for the future day**

**tomorrow\_weather\_label.config(**

**text=f"The predicted temperature for tomorrow is {round(tomorrow\_temp, 2)}°C & humidity is {round(tomorrow\_humidity, 2)}%"**

```
if tomorrow_temp < 10:
    precaution_label.config(text="you are in cool
temperature so please eat hot foods like :\\n Soup\\n
ChiliRoasted vegetables\\n Hot beverages")

elif tomorrow_temp < 30:
    precaution_label.config(text="you are in moderate
temperature feel the nature")

elif tomorrow_temp < 50:
    precaution_label.config(text="you are in hot temperature
so please drink cool jucices like :\\n Watermelon juice\\n
Lemonade\\n Cucumber juice\\n Pineapple juice\\n refreshing
drink")

elif tomorrow_temp < 70:
    precaution_label.config(text="you are in very hot
temperature so please drink cool jucices like :\\n Watermelon
juice\\n Lemonade\\n Cucumber juice\\n Pineapple juice\\n
refreshing drink")

elif tomorrow_temp < -30:
    precaution_label.config(text="hlo bro please enter valid
temperatures")

else:
    precaution_label.config(text="hlo bro please enter valid
temperatures")
```

**# Set the background image**

**image\_file = Image.open("Backgroundimage.png")**

**resized\_image = image\_file.resize((1375,762))**

**tk\_image = ImageTk.PhotoImage(resized\_image)**

**image\_label = tk.Label(root, image=tk\_image)**

**image\_label.place(relwidth=1, relheight=1)**

**# Create the GUI widgets**

**yesterday\_temp\_label = tk.Label(root, text="Yesterday  
Temperature (°C): ")**

**yesterday\_temp\_entry = tk.Entry(root)**

**yesterday\_humidity\_label = tk.Label(root, text="Yesterday  
Humidity (%): ")**

**yesterday\_humidity\_entry = tk.Entry(root)**

**today\_temp\_label = tk.Label(root, text="Today Temperature  
(°C): ")**

**today\_temp\_entry = tk.Entry(root)**

**today\_humidity\_label = tk.Label(root, text="Today Humidity  
(%): ")**

**today\_humidity\_entry = tk.Entry(root)**

**predict\_button = tk.Button(root, text="Tomorrow Weather",  
command=predict\_weather)**

**tomorrow\_weather\_label = tk.Label(root)**

**precaution\_label = tk.Label(root)**

**# Place the widgets in the GUI window**

**yesterday\_temp\_label.grid(row=0, column=0, padx=(300,0),  
pady=(150,0))**

**yesterday\_temp\_entry.grid(row=0, column=0, padx=(700,0),  
pady=(150,0))**

**yesterday\_humidity\_label.grid(row=0, column=0,  
padx=(300,0), pady=(200,0))**

**yesterday\_humidity\_entry.grid(row=0, column=0,  
padx=(700,0), pady=(200,0))**

**today\_temp\_label.grid(row=0, column=0, padx=(300,0),  
pady=(250,0))**

**today\_temp\_entry.grid(row=0, column=0, padx=(700,0),  
pady=(250,0))**

**today\_humidity\_label.grid(row=0, column=0, padx=(300,0),  
pady=(300,0))**

**today\_humidity\_entry.grid(row=0, column=0, padx=(700,0),  
pady=(300,0))**

**predict\_button.grid(row=0, column=0, padx=(700,0),  
pady=(350,0))**

**tomorrow\_weather\_label.grid(row=0, column=0,  
columnspan=2, padx=(950,0), pady=(150,0))**

```
precaution_label.grid(row=0, column=0, columnspan=3,  
rowspan=1, padx=(950,0), pady=(300,0))
```

```
# Start the GUI loop
```

```
root.mainloop()
```

# OUTPUT OF THE PROJECT:





# WEATHER

Yesterday Temperature (°C):

45

The predicted temperature for tomorrow is 43.0°C &amp; humidity is 27.0%

Yesterday Humidity (%):

23

Today Temperature (°C):

44

Today Humidity (%):

23

Tomorrow Weather

you are in hot temperature so please drink cool juices like :

Watermelon juice

Lemonade

Cucumber juice

Pineapple juice

refreshing drink

Activate Windows

Go to Settings to activate Windows



Type here to search



21:44

02-05-2023





# ***SCOPE OF PROJECT***

>Based on the provided code, this is a GUI weather predictor application that uses linear regression to predict the temperature and humidity of the following day based on the user-provided weather data from the previous day and current day.

>The GUI window contains input fields for the weather data of the previous day and current day, a button to predict the weather for the following day, and labels to display the predicted weather and precautions based on the predicted temperature.

>The linear regression model is trained using the data provided by the user, and the predicted temperature and humidity for the following day are displayed in the GUI window.

>Additionally, based on the predicted temperature, the application provides recommendations for precautions that should be taken to stay healthy and comfortable, such as drinking cool juices or eating hot foods

# FUTURE DEVELOPMENT OF PROJECT

1.Data validation: Add data validation to ensure that the user enters valid temperature and humidity values. For example, you can check that the temperature is within a reasonable range (e.g., -50°C to 50°C) and the humidity is between 0 and 100%.

2.More accurate prediction model: The current implementation uses a simple linear regression model, but you can experiment with other machine learning models to improve the accuracy of the predictions. For example, you can try using a decision tree or a neural network.

3.Historical data analysis: You can add a feature to the app that allows the user to input historical weather data and analyze it to identify trends and patterns. This can provide valuable insights into how weather conditions change over time

.

4.User interface improvements: The current implementation uses a basic user interface, but you can make it more visually appealing and user-friendly. For example, you can add icons, animations, and other visual elements to enhance the user experience.

5.Integration with weather APIs: You can integrate the app with weather APIs to get real-time weather data for the user's location.

>This can provide more accurate predictions and make the app more useful for users who need to plan their activities based on weather conditions.

## ***CONCLUSION***

>My code creates a GUI window that allows the user to enter weather data for two days (yesterday and today), and predicts the weather data for the future day using linear regression. The predicted temperature and humidity values are displayed on the GUI window along with a message suggesting the precautions to take based on the predicted temperature

>The code also loads a background image and places various labels and entry widgets in the GUI window using the `grid()` method. Finally, the code starts the GUI loop using the `mainloop()` method.



Submitted by:

**Gorlamari Srujan**

**Kumar Reddy**

**Reg no : 12111003**