**A Project Report**

**On**

**WEATHER PREDICTIONS**

***Submitted in partial fulfillment of the***

***requirement for the award of the degree of***

**MASTER OF COMPUTER APPLICATION**

****

Session (2024-25)

By

**DIYAKUMARI**

**24SCSE2030284**

**SWEETY KUMARI**

**24SCSE2030462**

**ANKITA RAJ  
 24SCSE2030096**

Under the guidance of

**Mr.Srinivasan Raju**

# 

**SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY**

**GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA**

****

# CANDIDATE’S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **“WEATHER PREDICTIONS”** in partial fulfillment of the requirements for the award of the MCA (Master of Computer Application) submitted in the School of Computer Applications and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of Sept, 2024 to feb, 2025, under the supervision of Mr.Srinivasan Raju School of Computer Applications and Technology , Galgotias University, Greater Noida.

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Diya kumari (24SCSE2030284)

Sweety kumari(24SCSE2030462)

Ankita Raj(24SCSE2030096)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr.Srinivasan Raju

Assistant Professor

## CERTIFICATE

This is to certify that Project Report entitled” WEATHER PREDICTIONS” which is submitted by Diya kumari, Sweety kumari, Ankita raj in partial fulfillment of the requirement for the award of degree MCA in School of Computer Applications and Technology, Galgotias University, Greater Noida, India is a record of the candidate own work carried out by them under my supervision. The matter embodied in this project is original and has not been submitted for the award of any other degree.

**Signature of Examiner(s) Signature of Supervisor(s)**

Date: Sep, 2024

Place: Greater Noida

## ACKNOWLEDGEMENT

We are pleased to present this Project report entitled “WEATHER PREDICTION USING MACHINE LEARNING”. It is indeed a great pleasure and a moment of immense satisfaction for me to express sense of profound gratitude and indebtedness towards our guide Asst. Mr. Srinivasan Raju whose enthusiasm are the source of inspiration for me. I am extremely thankful for the guidance and untiring attention. I got to learn a lot more about this project which will be very helpful for me. We would like to thank the entire Teaching staff who are directly or indirectly involved in the various data collection and software assistance to bring forward this report. I express deep sense of gratitude towards my parents for their sustained cooperation and wishes, which have been a prime source of inspiration to take this project work. Last but not the least, we would like to thank all our MCA colleagues for their co-operation and useful suggestion and all those who have directly or indirectly helped us in completion of this project work.

**LIST OF TABLE**

|  |  |  |
| --- | --- | --- |
| **SNO** | **TOPICS** | **PAGE NO.** |
| **1** | **ABSTRACT** | **5** |
| **2** | **INTRODUCTION** | **6** |
| **3** | **TECHNOLOGY USED** | **7** |
| **4** | **SOFTWARE REQUIREMENT SPECIFICATION** | **8** |
| **5** | **CODE** | **9** |
| **6** | **RESULT** | **14** |
| **7** | **CONCLUSION** | **15** |
| **8** | **FUTURE SCOPE** | **16** |

**ABSTRACT**

Weather prediction plays a critical role in numerous sectors, including agriculture, transportation, and disaster management. Traditional meteorological models have relied on numerical weather prediction (NWP) methods, which require extensive computational resources and a deep understanding of atmospheric dynamics. Despite the promise of ML in weather prediction, challenges such as data quality, model interpretability, and computational complexity remain.

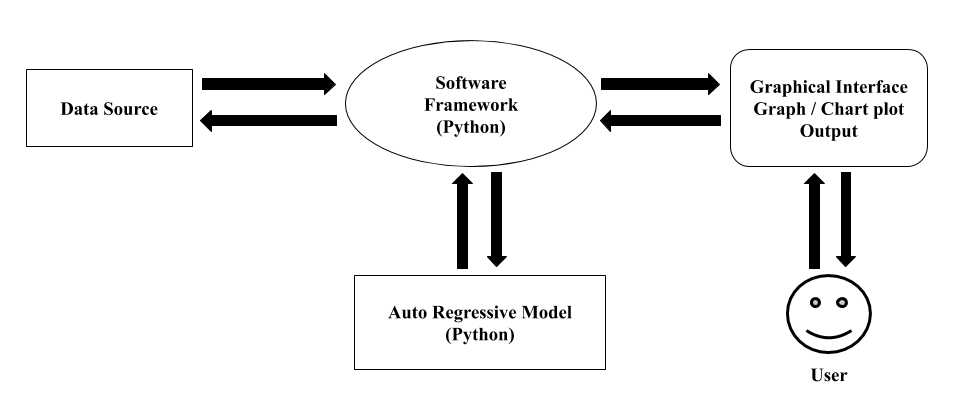
This paper also examines recent advances in ML techniques, such as neural networks and reinforcement learning, and their potential to further revolutionize weather prediction systems in the future.

**INTRODUCTION**

Weather prediction has always been a crucial task, impacting a wide range of sectors, including agriculture, aviation, transportation, disaster management, and daily life.

Traditional weather forecasting methods have relied on numerical weather prediction (NWP), which involves complex mathematical models and simulations of atmospheric dynamics. These methods, though highly accurate, require significant computational power and are limited by the quality and resolution of available data. As a result, they often face challenges in accurately predicting short-term weather patterns or sudden changes in weather conditions.

**FLOWCHART OF WEATHER PREDICTION IN PYTHON:**



**TECHNOLOGY USED**

**MACHINE LEARNING:**

In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of Machine Learning. Machine Learning is a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959. Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.



**SOFTWARE REQUIREMENTS SPECIFICATION**

* **HARDWARE REQUIREMENTS**

|  |  |
| --- | --- |
| **NUMBER** | **DESCRIPTION** |
| 1 | PC With 256 GB or more hard disk |
| 2 | Pc with 4GB RAM |
| 3 | Pc with pentinum 1 |

* **SOFTWARE REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **NUMBER** | **DESCRIPTION** | **TYPE** |
| 1 | OPERATING SYSTEM | WINDOWS HP |
| 2 | LANGUAGE | PYTHON |
| 3 | IDE | JUPYTER NOTEBOOK |
| 4 | BROWSER | GOOGLE CHROME |

**CODE**

import requests

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_absolute\_error

# Your OpenWeatherMap API key

API\_KEY = "db837532278e7cd8b73cdec9918eed93"

# List of Indian cities

cities = ["Delhi", "Mumbai", "Kolkata", "Chennai", "Bangalore"]

# Function to fetch weather data for a city

def fetch\_weather\_data(city):

url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API\_KEY}&units=metric"

response = requests.get(url)

if response.status\_code == 200:

data = response.json()

return {

"City": city,

"Temperature": data["main"]["temp"],

"Humidity": data["main"]["humidity"],

"WindSpeed": data["wind"]["speed"],

"Pressure": data["main"]["pressure"]

}

else:

print(f"Failed to fetch data for {city}")

return None

# Fetch data for all cities

weather\_data = [fetch\_weather\_data(city) for city in cities]

weather\_data = [data for data in weather\_data if data]

# Convert data to a DataFrame

df = pd.DataFrame(weather\_data)

# Features and target

X = df[["Humidity", "WindSpeed", "Pressure"]]

y = df["Temperature"]

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train a Random Forest Regressor

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate the model

mae = mean\_absolute\_error(y\_test, y\_pred)

print(f"Mean Absolute Error: {mae}")

# Predict temperature for a user-provided city

def predict\_temperature\_for\_city(city):

city\_data = fetch\_weather\_data(city)

if city\_data:

new\_data = pd.DataFrame({

"Humidity": [city\_data["Humidity"]],

"WindSpeed": [city\_data["WindSpeed"]],

"Pressure": [city\_data["Pressure"]]

})

predicted\_temp = model.predict(new\_data)[0]

print(f"Predicted Temperature for {city}: {predicted\_temp:.2f}°C")

else:

print(f"Could not fetch data for {city}.")

# Example usage

city\_to\_predict = input("Enter a city name: ")

predict\_temperature\_for\_city(city\_to\_predict)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Weather Prediction</title>

<link rel="stylesheet" href="/static/styles.css">

</head>

<body>

<div class="container">

<h1>Weather Prediction for Indian Cities</h1>

<form action="/" method="POST">

<label for="city">Enter City Name:</label>

<input type="text" id="city" name="city" placeholder="e.g., Delhi" required>

<button type="submit">Predict Temperature</button>

</form>

{% if predicted\_temp %}

<div class="result">

<h3>Predicted Temperature for {{ city }}:</h3>

<p>{{ predicted\_temp }}°C</p>

</div>

{% elif error %}

<div class="error">

<p>{{ error }}</p>

</div>

{% endif %}

</div>

</body>

</html>

body {

font-family: Arial, sans-serif;

background-color: #f0f8ff;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.container {

background: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

text-align: center;

max-width: 400px;

width: 100%;

}

h1 {

margin-bottom: 20px;

font-size: 24px;

}

form {

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 10px;

}

input {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

button {

background-color: #007BFF;

color: #fff;

border: none;

padding: 10px 20px;

border-radius: 5px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

.result, .error {

margin-top: 20px;

text-align: left;

}

.result p {

font-size: 20px;

font-weight: bold;

}

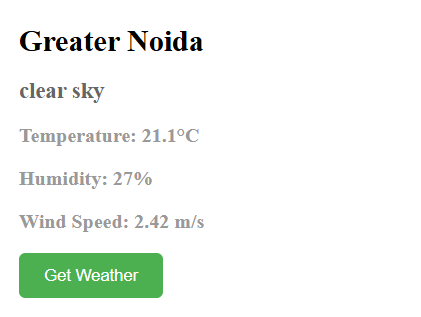
.error p {

color: red;

font-weight: bold;

}

**OUTPUT**



**RESULTS**

Machine Learning in weather prediction has led to significant advancements, resulting in improved forecasting accuracy, faster predictions, and better handling of complex weather patterns. Machine learning algorithms, particularly deep learning techniques such as convolutional neural networks (CNNs) and RNNs have demonstrated superior performance in capturing complex patterns in weather data. By analyzing vast amounts of historical data (e.g., temperature, humidity, precipitation, wind speed), ML models have been able to provide more accurate short-term and long-term weather forecasts than traditional models in some instances.

**CONCLUSION**

Machine learning (ML) has emerged as a transformative tool in the field of weather prediction, offering significant improvements over traditional forecasting methods. By leveraging large datasets, such as historical weather records, real-time sensor data, and satellite imagery, ML models have demonstrated superior capabilities in identifying complex patterns, improving the accuracy of forecasts, and predicting extreme weather events with greater precision. Techniques such as deep learning, reinforcement learning, and hybrid models combining ML with traditional numerical weather prediction (NWP) systems have shown promising results in enhancing both short-term and long-term weather forecasting.

**FUTURE SCOPE**

The future of weather prediction through machine learning (ML) holds tremendous promise, driven by continuous advancements in both data availability and computational power. As ML technologies mature and weather data becomes increasingly detailed and accessible, we can expect several transformative developments in the field. Advanced ML models, such as deep learning networks, are expected to contribute to more granular predictions, enhancing both short-term (e.g., next-hour or next-day) and long-term (e.g., seasonal or annual) forecasting.

**REFERENCE:**

A)[www.google.com](http://www.google.com)