REPORT ON

WEATHER ANALYSIS

By:

Ashish Srivastava

[202401100300075]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Course:

B. Tech - Computer Science and Engineering with Artificial Intelligence

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Institution:

KIET Group of Institutions

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:

11 March 2025

INTRODUCTION

Weather data analysis is essential for understanding climate trends, predicting extreme conditions, and making informed decisions in various sectors such as agriculture, transportation, and disaster management. By examining key meteorological parameters like temperature, rainfall, and humidity, we can identify patterns and anomalies that impact daily life and long-term planning.

This study focuses on analyzing a dataset containing weather records over a specific period. The goal is to extract meaningful insights by computing summary statistics, detecting extreme weather events, and visualizing trends. Using Python and data analysis libraries such as Pandas and Matplotlib, we perform various operations, including:

Identifying the hottest and wettest days.

Observing fluctuations in temperature and rainfall over time.

Creating graphical representations to showcase seasonal trends.

Through this analysis, we can better understand climate variations and their potential implications. Such insights are valuable for meteorologists, environmental researchers, and policymakers in making data-driven decisions. Moreover, historical weather data can help in improving forecasting

METHODOLOGY

The weather data analysis follows a structured approach using Python and data analysis libraries like Pandas and Matplotlib. The methodology includes the following steps:

1. Data Collection: The dataset containing temperature, rainfall, and humidity records is imported in CSV format.

2. Data Preprocessing: The dataset is cleaned by handling missing values and converting date columns to a standard format.

3. Statistical Analysis: Summary statistics such as mean, maximum, and minimum values are computed.

4. Extreme Condition Identification: The hottest and wettest days are determined.

5. Visualization: Trends in temperature and rainfall are analyzed using graphs for better interpretation.

models and preparing for future climatic changes. This report presents a structured approach to weather data analysis, highlighting its significance in both scientific

CODE

import pandas as pd

import matplotlib.pyplot as plt

# Load the dataset

file\_path = "C:\\Users\Ashish\Desktop\python\weather\_data.csv"  # Update with your file path if needed

df = pd.read\_csv(file\_path)

# Convert 'Date' column to datetime format

df["Date"] = pd.to\_datetime(df["Date"])

# Basic statistics

print("Summary Statistics:")

print(df.describe())

# Find the day with the highest temperature

hottest\_day = df.loc[df["Temperature"].idxmax()]

print("\nHottest Day:")

print(hottest\_day)

# Find the day with the highest rainfall

wettest\_day = df.loc[df["Rainfall"].idxmax()]

print("\nWettest Day:")

print(wettest\_day)

# Plot Temperature & Rainfall trends

plt.figure(*figsize*=(10,5))

plt.plot(df["Date"], df["Temperature"], *label*="Temperature (°C)", *color*="red")

plt.xlabel("Date")

plt.ylabel("Temperature")

plt.title("Temperature Trend")

plt.legend()

plt.xticks(*rotation*=45)

plt.show()

plt.figure(*figsize*=(10,5))

plt.plot(df["Date"], df["Rainfall"], *label*="Rainfall (mm)", *color*="blue")

plt.xlabel("Date")

plt.ylabel("Rainfall")

plt.title("Rainfall Trend")

plt.legend()

plt.xticks(*rotation*=45)

plt.show()

SCREENSHORT

