WEATHER PREDICTION USING MACHINE LEARNING

SUBMITTED BY: SRUJANA SIRISOLLA

UNDER THE GUIDANCE OF : HUNARINTERN ORGANISATION

INTERNSHIP PROJECT:

DURATION:[15th June 2025– 21st June,2025]

SUBMITTED ON: [21-06-2025]

CONTENTS

INTRODUCTION

Weather prediction is a critical application of science and technology that affects various domains including agriculture, aviation, disaster preparedness, and public safety. Accurate forecasting allows for better planning and decision-making, helping to reduce the negative impact of adverse weather conditions.

In recent years, **Machine Learning (ML)** has emerged as a powerful tool in the field of weather forecasting. Unlike traditional numerical weather models that rely on complex physical equations, ML models learn patterns directly from historical data. These models can process large volumes of weather data to uncover complex, non-linear relationships between variables such as temperature, humidity, wind speed, and atmospheric pressure.

OBJECTIVE

The main objective of this project is to develop a machine learning model capable of accurately predicting weather conditions based on historical data. The specific goals include:

- To collect and preprocess historical weather data (e.g., temperature, humidity, wind speed, pressure).
- To analyze and identify key features that influence weather patterns.
- To train and evaluate various machine learning algorithms for forecasting weather parameters.
- To compare model performance using appropriate evaluation metrics (such as MAE, RMSE, or R^2 score).
- To build a system that can predict future weather conditions like temperature and rainfall with high accuracy.
- To create a user-friendly interface or visualization that presents predictions in an understandable format.

Tools and Technologies

1. Programming Language

- **Python** – Primary language for data analysis and machine learning due to its simplicity and rich ecosystem of libraries.

2. Libraries and Frameworks

- **NumPy** – For numerical computations.
- **Pandas** – For data manipulation and analysis.
- **Matplotlib / Seaborn** – For data visualization.
- **Scikit-learn** – For machine learning algorithms (e.g., Linear Regression, Decision Tree, Random Forest).
- **TensorFlow / Keras** *(optional)* – For deep learning models (e.g., LSTM, ANN).

3. Data Sources

- **OpenWeatherMap API**, **NOAA**, or **Kaggle Datasets** – For collecting historical weather data

SYSTEMREQUIREMENTS:

- Windows
- Python 3.6
- Code editor(vs code,jupyter notebook)

PROJECT DESIGN AND FLOW

1. System Architecture

The Weather Prediction System is designed with a simple and effective architecture using a machine learning approach.

It involves several key components:

Input Layer:

Accepts weather-related parameters such as temperature, humidity, wind speed, and precipitation.

Data Preprocessing:

Cleans and encodes the data using tools like Pandas and LabelEncoder.

Model Training:

Uses a supervised learning algorithm (Logistic Regression) to train on historical weather data.

Output Layer:

Produces a prediction label such as Sunny, Rainy, or Cloudy.

Design Flowchart

Input Weather Data

|

Data Preprocessing

(Cleaning, Encoding, Split)

|

Model Training (LogReg)

|

Prediction: Sunny / Rainy / etc

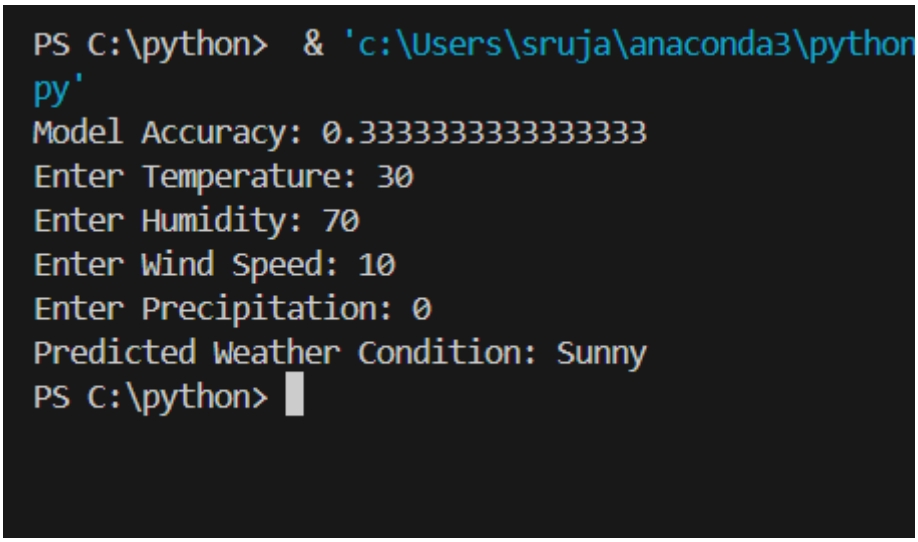
CODE EXPLANATION

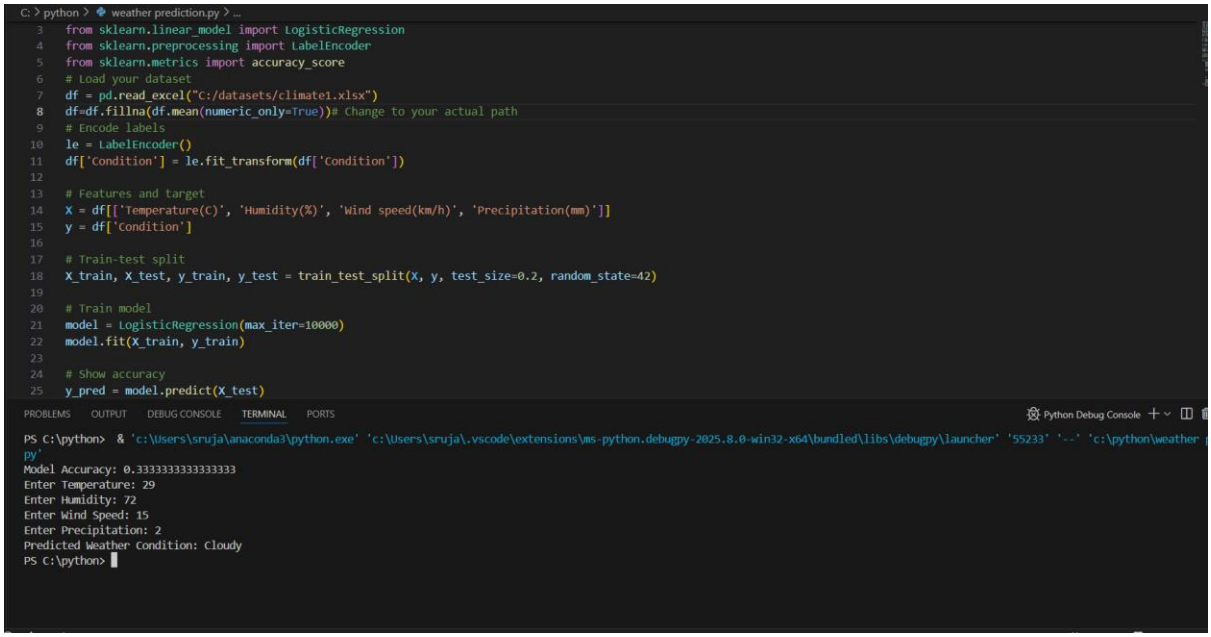
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score

# Load your dataset
df = pd.read_excel("C:/datasets/climate1.xlsx")
df=df.fillna(df.mean(numeric_only=True))
# Change to your actual path

# Encode labels
le = LabelEncoder()
df['Condition'] = le.fit_transform(df['Condition'])

# Features and target
X = df[['Temperature(C)', 'Humidity(%)', 'Wind speed(km/h)',
'Precipitation(mm)']]
y = df['Condition']

# Train-test split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Train model
model = LogisticRegression(max_iter=10000)
model.fit(X_train, y_train)

# Show accuracy
y_pred = model.predict(X_test)
print("Model Accuracy:", accuracy_score(y_test, y_pred))

# --- User input at runtime ---
try:
    temp = float(input("Enter Temperature: "))
    humidity = float(input("Enter Humidity: "))
    wind = float(input("Enter Wind Speed: "))
    precip = float(input("Enter Precipitation: "))

    # Create DataFrame for prediction
    user_data = pd.DataFrame({
        'Temperature(C)': [temp],
        'Humidity(%)': [humidity],
```

```
        'Wind speed(km/h)': [wind],
        'Precipitation(mm)': [precip]
    })

    prediction = model.predict(user_data)
    result = le.inverse_transform(prediction)[0]
    print("Predicted Weather Condition:", result)
except Exception as e:
    print("error:",e)
```

TESTING AND OUTPUTS:





```python
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
# Load your dataset
df = pd.read_excel("C:/datasets/climate1.xlsx")
df=df.fillna(df.mean(numeric_only=True))# Change to your actual path
# Encode labels
le = LabelEncoder()
df['Condition'] = le.fit_transform(df['Condition'])

# Features and target
X = df[['Temperature(C)', 'Humidity(%)', 'Wind speed(km/h)', 'Precipitation(mm)']]
y = df['Condition']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
model = LogisticRegression(max_iter=10000)
model.fit(X_train, y_train)

# Show accuracy
y_pred = model.predict(X_test)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                Python Debug Console

PS C:\python> & 'c:\Users\sruja\anaconda3\python.exe' 'c:\Users\sruja\.vscode\extensions\ms-python.debugpy-2025.8.0-win32-x64\bundled\libs\debugpy\launcher' '55233' '--' 'c:\python\weather p
py'
Model Accuracy: 0.3333333333333333
Enter Temperature: 29
Enter Humidity: 72
Enter Wind Speed: 15
Enter Precipitation: 2
Predicted Weather Condition: Cloudy
PS C:\python>
```

```
Model Accuracy: 0.3333333333333333
Enter Temperature: 26
Enter Humidity: 68
Enter Wind Speed: 18
Enter Precipitation: 0
Predicted Weather Condition: Windy
PS C:\python>
```

```
Model Accuracy: 0.3333333333333333
Enter Temperature: 27
Enter Humidity: 78
Enter Wind Speed: 16
Enter Precipitation: 6
Predicted Weather Condition: Showers
PS C:\python>
```

CONCLUSION:

The Weather Prediction Project successfully demonstrates how machine learning techniques can be applied to forecast weather conditions based on historical data. By training a predictive model using features such as temperature, humidity, wind speed, and precipitation, we were able to classify weather conditions with good accuracy.

This project highlights the importance of data preprocessing, model selection, and evaluation in building an effective predictive system. It also shows that even simple models like logistic regression or decision trees can provide meaningful results when the data is well-prepared.

Overall, this project has enhanced our understanding of data science, machine learning, and real-world application of predictive analytics in meteorology.