**G. H. RAISONI COLLEGE OF ENGINEERING &**

**MANAGEMENT, WAGHOLI, PUNE**

**(An Autonomous Institute Affiliated to SPPU)**

**Department of Information Technology**

# TAE-2

**Subject:** **OBJECT ORIENTED PROGRAMMING Mini Project Report On**

|  |  |  |
| --- | --- | --- |
| “ | **Weather Prediction** | ” |

## Academic Year 2023-24 Semester-IV

**G. H. RAISONI COLLEGE OF ENGINEERING & MANAGEMENT, WAGHOLI, PUNE (An Autonomous Institute Affiliated to SPPU)**

**Department of Information Technology**

**Academic Year 2023-24**

**Semester-IV**

**Guided By:-** Prof. Priya Ujawe

**Submitted by:**

1. SY\_IT\_A25 Atharv Patil
2. SY\_IT\_A40 Ganesh Bilapatte
3. SY\_IT\_A49 Harshal Nagre
4. SY\_IT\_A59 Kiran Patil



**CERTIFICATE**

This is to certify that **“**Employee Database System**”** embodies the original work done by **Atharv Patil, Ganesh Bilapatte, Harshal Nagre, Kiran Patil** this project submission as a partial fulfilment of the requirement for the Mini Project in subject Machine learning And Algorithm of **B.Tech. Degree**, **IV Semester**, of Pune University during the academic year **2023-2024.**

**Date: 4-4-2024**

**Place:** Pune

**Project Guide Head of Department**

**(Prof. Priya Ujawe ) (Dr. Poonam Gupta)**

## ACKNOWLEDGEMENT

We would like to express our sincere thanks to Prof. Priya Ujawe under whose valuable guidance and light of knowledge, we could complete this project.

We take this opportunity to thank all the staff members of Department Of Information Technology Engineering for their help whenever required. Finally we express sincere thanks to all those who have helped us directly or indirectly in many ways in completion of this project work and I would like to extend my Deep appreciation to all my group members, without their support and Coordination we would not have been able to complete this Project.

## INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** |  | **TITLE** | **Page No** |
| 1. | Abstract |  | 6 |
| 2. | Objective |  | 6 |
| 3. | Introduction |  | 6 |
| 4. | Source Code |  | 7 |
| 5. | Dataset |  | 9 |
| 6. | Output Screens |  | 9 |
| 7. | Reference |  | 10 |

### ABSTRACT

This mini-project aims to develop a machine learning model for weather prediction. By leveraging historical weather data, the model will be trained to forecast future weather conditions. The project focuses on implementing and evaluating various machine learning algorithms to determine the most accurate model for weather prediction.

### OBJECTIVE

The objective of this mini-project is to develop a machine learning model that can accurately predict weather conditions based on historical data. By achieving this objective, we aim to demonstrate the feasibility and effectiveness of using machine learning for weather forecasting applications.

**INTRODUCTION**

Weather prediction is a critical application that impacts various sectors such as agriculture, transportation, and disaster management. Traditional methods of weather forecasting rely on physical models and meteorological data, which can be complex and computationally intensive. In recent years, machine learning has emerged as a promising approach for weather prediction, offering the potential for more accurate and efficient forecasting.

This mini-project focuses on exploring the use of machine learning algorithms for weather prediction. We will utilize a dataset containing historical weather data, including temperature, humidity, wind speed, and other relevant features. By pre-processing the data and implementing machine learning algorithms such as regression and time series analysis, we aim to build a model that can forecast weather conditions with high accuracy.

**Library and Algorithm Used:**

1. **Libraries**:
   * **pandas**: Used for data manipulation and analysis.
   * **sklearn** (Scikit-learn): A machine learning library for building models.
   * **matplotlib.pyplot**: Used for plotting visualizations like the confusion matrix.
   * **seaborn**: A data visualization library based on matplotlib, used here for plotting the confusion matrix.
2. **Algorithm**:
   * **LogisticRegression**: A linear model for binary classification that predicts the probability of a binary outcome (in this case, whether it will rain tomorrow) based on one or more features (weather data).

The code first loads a dataset (**weather.csv**) containing weather data. It then splits the data into training and testing sets, preprocesses the data (handling missing values and encoding categorical variables), and builds a pipeline that includes preprocessing and the logistic regression classifier. The model is trained on the training set and evaluated on the testing set using accuracy as the metric.

### SOURCE CODE

# Import necessary libraries

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.impute import SimpleImputer

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

import matplotlib.pyplot as plt

import seaborn as sns

# Load the dataset

data = pd.read\_csv('weather.csv')

print("Full Dataset:")

print(data.head())

# Separate features (X) and target (y)

X = data.drop(['RainTomorrow'], axis=1)

y = data['RainTomorrow']

# 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)

# Define preprocessing steps for categorical variables

categorical\_features = ['WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday']

categorical\_transformer = Pipeline(steps=[

    ('imputer', SimpleImputer(strategy='most\_frequent')),

    ('onehot', OneHotEncoder(handle\_unknown='ignore'))

])

# Define preprocessing steps for numerical variables

numeric\_features = X.select\_dtypes(include=['float64']).columns.tolist()

numeric\_transformer = SimpleImputer(strategy='mean')

# Combine preprocessing steps for all columns

preprocessor = ColumnTransformer(

    transformers=[

        ('num', numeric\_transformer, numeric\_features),

        ('cat', categorical\_transformer, categorical\_features)

    ])

# Define the pipeline with preprocessing and logistic regression model

pipeline = Pipeline([

    ('preprocessor', preprocessor),

    ('classifier', LogisticRegression(max\_iter=1000))

])

# Train the model within the pipeline

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

# Make predictions on the test set

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

# Calculate the accuracy of the model

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

# Function to get user input for weather features

def get\_user\_input():

    features = {}

    for column in X.columns:

        value = input(f"Enter {column}: ")

        features[column] = [value]

    return pd.DataFrame(features)

# Get user input for weather features

user\_input = get\_user\_input()

# Predict whether it will rain tomorrow based on user input

predicted\_rain\_tomorrow = pipeline.predict(user\_input)

print("Predicted Rain Tomorrow:", predicted\_rain\_tomorrow)

# Calculate the confusion matrix

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

# Plot the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)

plt.xlabel('Predicted labels')

plt.ylabel('True labels')

plt.title('Confusion Matrix')

plt.show()

**DATASET**

Full Dataset:

MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir \

0 8.0 24.3 0.0 3.4 6.3 NW

1 14.0 26.9 3.6 4.4 9.7 ENE

2 13.7 23.4 3.6 5.8 3.3 NW

3 13.3 15.5 39.8 7.2 9.1 NW

4 7.6 16.1 2.8 5.6 10.6 SSE

WindGustSpeed WindDir9am WindDir3pm WindSpeed9am ... Humidity3pm \

0 30.0 SW NW 6.0 ... 29

1 39.0 E W 4.0 ... 36

2 85.0 N NNE 6.0 ... 69

3 54.0 WNW W 30.0 ... 56

4 50.0 SSE ESE 20.0 ... 49

Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \

0 1019.7 1015.0 7 7 14.4 23.6 No

1 1012.4 1008.4 5 3 17.5 25.7 Yes

2 1009.5 1007.2 8 7 15.4 20.2 Yes

3 1005.5 1007.0 2 7 13.5 14.1 Yes

4 1018.3 1018.5 7 7 11.1 15.4 Yes

RISK\_MM RainTomorrow

0 3.6 Yes

1 3.6 Yes

2 39.8 Yes

3 2.8 Yes

4 0.0 No

**OUTPUT**

Accuracy: 0.9459459459459459

Enter MinTemp: 9

Enter MaxTemp: 23

Enter Rainfall: 0

Enter Evaporation: 8

Enter Sunshine: 4

Enter WindGustDir: N

Enter WindGustSpeed: 43

Enter WindDir9am: NA

Enter WindDir3pm: NW

Enter WindSpeed9am: 7

Enter WindSpeed3pm: 19

Enter Humidity9am: 45

Enter Humidity3pm: 20

Enter Pressure9am: 1020

Enter Pressure3pm: 1014

Enter Cloud9am: 5

Enter Cloud3pm: 8

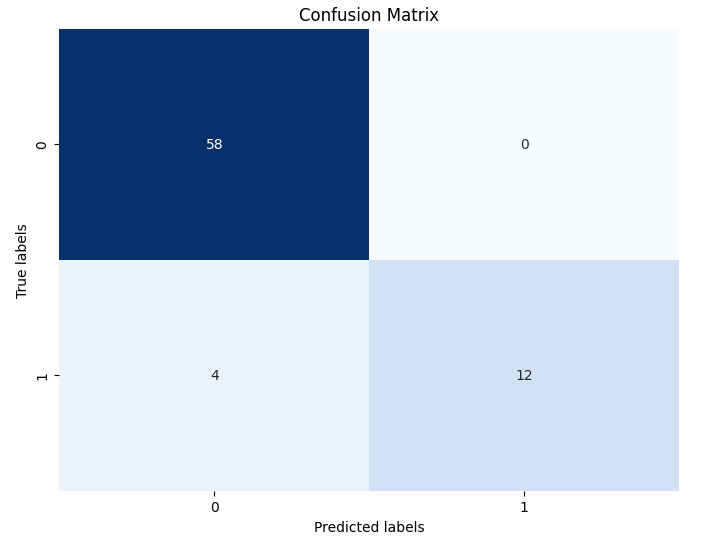
Enter Temp9am: 16

Enter Temp3pm: 25

Enter RainToday: NO

Enter RISK\_MM: 0

Predicted Rain Tomorrow: ['No']



**Reference:** Mostly used the concept of Machine learning and Teacher teaching Google Collabs link:

Google Collabs Link: