

Team: Decimal Binary EN43 -(Round 3)

Problem Statement: Weather Data Analysis and Prediction model

Description: Use weather datasets to predict temperature, rainfall, or other conditions for specific regions. This can help in planning for agricultural or travel needs.

“Prediction model Building and its Deployment using Flask”

Description:

Prediction Model Building and its Deployment using Flask involves creating a machine learning model to predict weather conditions like temperature or rainfall using historical weather data. The model is then deployed as a web application using Flask, allowing users to input parameters and receive real-time predictions for specific region.

Accuracy:

```
'dew': 15,
'precipprob': 20,
'uvindex': 5
}
|
# Make prediction with sample data
prediction = predict_weather(sample_input)
print("\nSample Prediction:")
print(f"Maximum Temperature: {prediction['temperature']}°C")
print(f"Weather Condition: {prediction['condition']}")

# Display unique weather conditions
print("\nPossible weather conditions in the dataset:")
print(df['conditions'].unique())
```

17	cloudcover	517 non-null	float64
18	visibility	509 non-null	float64
19	solarrradiation	517 non-null	float64
20	solarenergy	517 non-null	float64
21	uvindex	517 non-null	int64
22	conditions	517 non-null	object
23	description	517 non-null	object
24	icon	517 non-null	object
25	stations	503 non-null	object

```
dtypes: float64(19), int64(2), object(5)
memory usage: 105.1+ KB
None

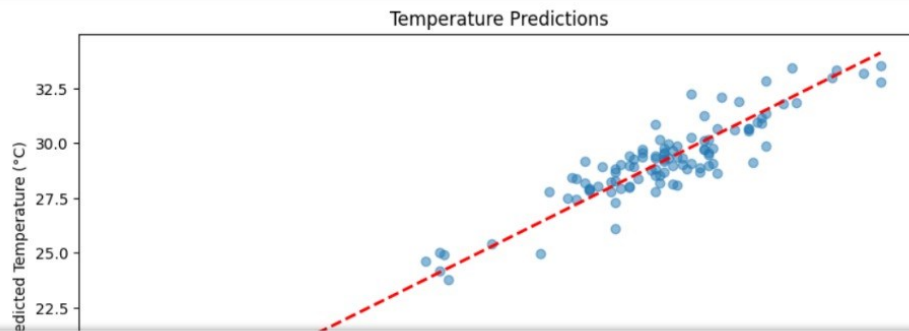
Temperature Prediction Results:
RMSE: 0.90°C

Weather Condition Prediction Results:
Accuracy: 99.04%
```

Actual vs Predicted temperature graph:

```
# Make prediction with sample data
prediction = predict_weather(sample_input)
print("\nSample Prediction:")
print(f"Maximum Temperature: {prediction['temperature']]°C")
print(f"Weather Condition: {prediction['condition']}")

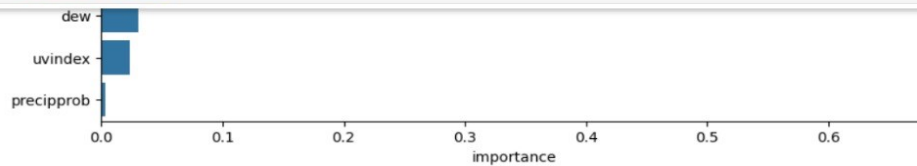
# Display unique weather conditions
print("\nPossible weather conditions in the dataset:")
print(df['conditions'].unique())
```



Prediction of Weather Condition:

```
# Make prediction with sample data
prediction = predict_weather(sample_input)
print("\nSample Prediction:")
print(f"Maximum Temperature: {prediction['temperature']]°C")
print(f"Weather Condition: {prediction['condition']}")

# Display unique weather conditions
print("\nPossible weather conditions in the dataset:")
print(df['conditions'].unique())
```



```
Sample Prediction:
Maximum Temperature: 28.48°C
Weather Condition: Partially cloudy

Possible weather conditions in the dataset:
['Clear' 'Partially cloudy' 'Rain, Partially cloudy' 'Overcast'
 'Rain, Overcast' 'Rain']
```

Deployment:

Deployment of this ML model involves integrating a trained model into a production environment for real-world use.

This includes saving the model, developing a Webpage or App for interaction, containerizing the model for consistency, and deploying it on cloud platforms for scalability. The goal is to automate predictions and provide real-time insights in applications.

The image displays two screenshots of a web application titled "Weather Prediction App". The top screenshot shows the input form where users can select a city and enter various weather parameters. The bottom screenshot shows the output of the prediction, including the city name, maximum temperature, and weather condition.

Weather Prediction App

Weather Predictor

Select City: Nagpur

Temperature (°C): 45

Humidity (%): 20

Windspeed (km/h): 23

Cloud Cover (%): 65

Sea Level Pressure (hPa): 1013

© 2025 Weather Prediction App | Developed by Decimal Binary

Sea Level Pressure (hPa): 1013

Feels Like (°C): 45

Dew Point (°C): 12

Precipitation Probability (%): 35

UV Index: 5

Predict

Prediction Results

City: Nagpur

Maximum Temperature: 33.04°C

Weather Condition: Partially cloudy

© 2025 Weather Prediction App | Developed by Decimal Binary

This is the application showing prediction of the weather condition based on specific regions.