Weather Forecasting using Quadratic Model with Agile–Prototype Process Model

PV Hemanth Kumar
2023001468

Weather Forecasting using Quadratic Model with Agile–Prototype Process Model

A weather forecasting system using a quadratic model can be developed using the Agile–Prototype process model by following an iterative and incremental approach. Instead of completing all phases sequentially, prototypes are built quickly, shown to users for feedback, and improved in multiple cycles until the system meets accuracy and usability requirements.

Scope

The objective of this project is to develop a weather prediction system that estimates temperature (°C) and humidity (%) across a 24-hour period using a quadratic regression approach. Prototypes will be built for input handling, model development, forecast generation, and visualization, which are refined sprint by sprint with user validation.

Inputs

- **Time (in hours):** [0, 4, 8, 12, 16, 20]
- **Temperature (°C):** [18, 22, 28, 33, 29, 21]
- **Humidity (%):** [80, 70, 60, 55, 65, 78]

Agile–Prototype Process

Iterative Prototype Cycles

1. **Prototype 1 – Input & Data Handling**
    - Collect sample time, temperature, and humidity data.
    - Build a simple input module to handle arrays.
    - Feedback: Ensure input format is user-friendly.
2. **Prototype 2 – Quadratic Model Development**
    - Fit quadratic regression models for temperature & humidity.
    - Print initial equations for user inspection.
    - Feedback: Compare with expected daily trends.
3. **Prototype 3 – Forecast Generation**
    - Generate hourly predictions (0–24 hrs).
    - Display values in tabular format.
    - Feedback: Validate values against real-world weather patterns.
4. **Prototype 4 – Visualization**
    - Plot graphs for predicted temperature & humidity trends.
    - Feedback: Check readability and accuracy of graphs.
5. **Prototype 5 – Testing & Refinement**
    - Validate accuracy using metrics (MAE, RMSE).

    - Fine-tune equations or input preprocessing if needed.
    - Feedback: Final user acceptance testing (UAT).

System Architecture

- **Input:** Time, temperature, humidity arrays.
- **Processing:** Quadratic regression applied on datasets.
- **Output:** Tabular forecast + Graphical plots.
- **Feedback Loop:** Users provide corrections at each prototype cycle.

Prototype Model Design

We use the quadratic regression formula:

$$y = a \cdot t^2 + b \cdot t + c$$

Where:

- **y** = predicted weather parameter (temperature or humidity)
- **t** = time (in hours)
- **a, b, c** = regression coefficients from training data

Implementation (Python Code)

```
import numpy as np


# Step 1: Input Data
time = np.array([0, 4, 8, 12, 16, 20])            # Time in hours
temperature = np.array([18, 22, 28, 33, 29, 21])   # Temperature in °C
humidity = np.array([80, 70, 60, 55, 65, 78])      # Humidity in %


# Step 2: Fit quadratic models
temp_coeff = np.polyfit(time, temperature, 2)
hum_coeff = np.polyfit(time, humidity, 2)


a_t, b_t, c_t = temp_coeff
a_h, b_h, c_h = hum_coeff


print("\nQuadratic Model for Temperature:")
print(f"T(t) = {a_t:.4f}t² + {b_t:.4f}t + {c_t:.4f}")


print("\nQuadratic Model for Humidity:")
print(f"H(t) = {a_h:.4f}t² + {b_h:.4f}t + {c_h:.4f}")


# Step 3: Predictions for 24 hours
```

```
t_values = np.arange(0, 25, 1)

predicted_temp = a_t * t_values**2 + b_t * t_values + c_t

predicted_hum = a_h * t_values**2 + b_h * t_values + c_h


print("\nPredicted Weather Forecast (24 Hours):")

for t, temp, hum in zip(t_values, predicted_temp, predicted_hum):

    print(f"{t:02d}:00 hrs -> Temperature: {temp:.2f} °C, Humidity: {hum:.2f} %")


# Step 4: Visualization

import matplotlib.pyplot as plt


plt.figure(figsize=(12,6))


# Temperature Plot

plt.subplot(1,2,1)

plt.scatter(time, temperature, color='red', label='Original Temperature Data')

plt.plot(t_values, predicted_temp, 'b--', label='Predicted Temperature')

plt.xlabel('Time (Hours)')

plt.ylabel('Temperature (°C)')

plt.title('Temperature Forecast (Quadratic Model)')

plt.legend()
```

```
plt.grid(True)


# Humidity Plot

plt.subplot(1,2,2)

plt.scatter(time, humidity, color='green', label='Original Humidity
Data')

plt.plot(t_values, predicted_hum, 'orange', linestyle='--',
label='Predicted Humidity')

plt.xlabel('Time (Hours)')

plt.ylabel('Humidity (%)')

plt.title('Humidity Forecast (Quadratic Model)')

plt.legend()

plt.grid(True)


plt.tight_layout()

plt.show()
```

Output

```
Quadratic Model for Temperature:
T(t) = -0.1116t² + 2.5250t + 16.2857

Quadratic Model for Humidity:
H(t) = 0.2176t² + -4.5670t + 81.7500

Predicted Weather Forecast (24 Hours):
00:00 hrs -> Temperature: 16.29 °C, Humidity: 81.75 %
01:00 hrs -> Temperature: 18.70 °C, Humidity: 77.40 %
02:00 hrs -> Temperature: 20.89 °C, Humidity: 73.49 %
03:00 hrs -> Temperature: 22.86 °C, Humidity: 70.01 %
04:00 hrs -> Temperature: 24.60 °C, Humidity: 66.96 %
05:00 hrs -> Temperature: 26.12 °C, Humidity: 64.36 %
06:00 hrs -> Temperature: 27.42 °C, Humidity: 62.18 %
07:00 hrs -> Temperature: 28.49 °C, Humidity: 60.45 %
08:00 hrs -> Temperature: 29.34 °C, Humidity: 59.14 %
09:00 hrs -> Temperature: 29.97 °C, Humidity: 58.28 %
10:00 hrs -> Temperature: 30.37 °C, Humidity: 57.84 %
11:00 hrs -> Temperature: 30.56 °C, Humidity: 57.85 %
12:00 hrs -> Temperature: 30.51 °C, Humidity: 58.29 %
13:00 hrs -> Temperature: 30.25 °C, Humidity: 59.16 %
14:00 hrs -> Temperature: 29.76 °C, Humidity: 60.47 %
15:00 hrs -> Temperature: 29.05 °C, Humidity: 62.21 %
16:00 hrs -> Temperature: 28.11 °C, Humidity: 64.39 %
17:00 hrs -> Temperature: 26.96 °C, Humidity: 67.01 %
18:00 hrs -> Temperature: 25.57 °C, Humidity: 70.06 %
19:00 hrs -> Temperature: 23.97 °C, Humidity: 73.54 %
20:00 hrs -> Temperature: 22.14 °C, Humidity: 77.46 %
21:00 hrs -> Temperature: 20.09 °C, Humidity: 81.82 %
22:00 hrs -> Temperature: 17.82 °C, Humidity: 86.61 %
23:00 hrs -> Temperature: 15.32 °C, Humidity: 91.84 %
24:00 hrs -> Temperature: 12.60 °C, Humidity: 97.50 %
```