Implement weather forecasting using Agile-Scrum process model

Name – PV Hemanth Kumar (2023001468)

Scope

The objective is to develop a simple weather prediction system that estimates rainfall (mm) and humidity (%) over a 24-hour period using a quadratic regression model. The development follows the Agile–Scrum process model, where work is divided into product backlog items and delivered in short sprints with continuous feedback.

Inputs

- Time (hours): [0, 4, 8, 12, 16, 20]
- Rainfall (mm): [0, 2, 8, 15, 10, 3]
- Humidity (%): [95, 85, 70, 60, 75, 90]

Agile–Scrum Process

Scrum Roles

- **Product Owner (PO):** Defines features & priorities.
- **Scrum Master (SM):** Facilitates Scrum process, removes blockers.
- **Development Team:** Builds data pipeline, quadratic model, graphs, and testing.
- **Stakeholders / Users:** Validate outputs, give feedback.

Product Backlog (User Stories)

1. As a user, I want to input time, rainfall, and humidity data so the model can learn patterns.
2. As a data scientist, I want to fit quadratic regression models so predictions can be generated.
3. As a user, I want forecasts displayed for each hour of the day.
4. As a user, I want results shown in graphs for better visualization.
5. As a tester, I want accuracy validated with metrics (MAE, RMSE).

Sprint Planning

- **Sprint 1:** Data input & preprocessing.
- **Sprint 2:** Implement quadratic regression model.
- **Sprint 3:** Generate hourly forecasts & tabular output.
- **Sprint 4:** Visualization with plots + accuracy check.
- **Sprint 5:** Testing, user feedback, and final deployment.

Each sprint ends with a **review demo** and **retrospective**, ensuring incremental delivery.

Definition of Done (DoD)

- Code implemented and reviewed.
- Forecasts generated for 24 hours.
- Graphs plotted successfully.
- Accuracy validated with MAE/RMSE.
- Output matches acceptance criteria.

Implementation (Python Code)

```
import numpy as np


# Step 1: Input Data
time = np.array([0, 4, 8, 12, 16, 20])      # Time in hours
rainfall = np.array([0, 2, 8, 15, 10, 3])   # Rainfall in mm
humidity = np.array([95, 85, 70, 60, 75, 90]) # Humidity in %


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


a_r, b_r, c_r = rain_coeff
a_h, b_h, c_h = hum_coeff


print("\nQuadratic Model for Rainfall:")
print(f"R(t) = {a_r:.4f}t² + {b_r:.4f}t + {c_r:.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_rainfall = a_r * t_values**2 + b_r * t_values + c_r
predicted_humidity = a_h * t_values**2 + b_h * t_values + c_h

print("\nPredicted Weather Forecast (24 Hours):")
for t, rain, hum in zip(t_values, predicted_rainfall, predicted_humidity):
    print(f"{t:02d}:00 hrs -> Rainfall: {rain:.2f} mm, Humidity: {hum:.2f} %")

# Step 4: Plotting
import matplotlib.pyplot as plt

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

# Rainfall Plot
plt.subplot(1,2,1)
plt.scatter(time, rainfall, color='blue', label='Original Rainfall Data')
plt.plot(t_values, predicted_rainfall, 'r--', label='Rainfall Prediction')
plt.xlabel('Time (Hours)')
plt.ylabel('Rainfall (mm)')
plt.title('Rainfall Prediction (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_humidity, 'orange', linestyle='--', label='Humidity Prediction')

plt.xlabel('Time (Hours)')

plt.ylabel('Humidity (%)')

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

plt.legend()

plt.grid(True)


plt.tight_layout()

plt.show()
```

Output

```
Quadratic Model for Rainfall:
R(t) = -0.0993t² + 2.3152t + -2.2500

Quadratic Model for Humidity:
H(t) = 0.2734t² + -5.9330t + 98.3929

Predicted Weather Forecast (24 Hours):
00:00 hrs -> Rainfall: -2.25 mm, Humidity: 98.39 %
01:00 hrs -> Rainfall: -0.03 mm, Humidity: 92.73 %
02:00 hrs -> Rainfall: 1.98 mm, Humidity: 87.62 %
03:00 hrs -> Rainfall: 3.80 mm, Humidity: 83.05 %
04:00 hrs -> Rainfall: 5.42 mm, Humidity: 79.04 %
05:00 hrs -> Rainfall: 6.84 mm, Humidity: 75.56 %
06:00 hrs -> Rainfall: 8.07 mm, Humidity: 72.64 %
07:00 hrs -> Rainfall: 9.09 mm, Humidity: 70.26 %
08:00 hrs -> Rainfall: 9.91 mm, Humidity: 68.43 %
09:00 hrs -> Rainfall: 10.54 mm, Humidity: 67.14 %
10:00 hrs -> Rainfall: 10.97 mm, Humidity: 66.41 %
11:00 hrs -> Rainfall: 11.20 mm, Humidity: 66.22 %
12:00 hrs -> Rainfall: 11.23 mm, Humidity: 66.57 %
13:00 hrs -> Rainfall: 11.06 mm, Humidity: 67.47 %
14:00 hrs -> Rainfall: 10.69 mm, Humidity: 68.92 %
15:00 hrs -> Rainfall: 10.13 mm, Humidity: 70.92 %
16:00 hrs -> Rainfall: 9.36 mm, Humidity: 73.46 %
17:00 hrs -> Rainfall: 8.40 mm, Humidity: 76.55 %
18:00 hrs -> Rainfall: 7.24 mm, Humidity: 80.19 %
19:00 hrs -> Rainfall: 5.88 mm, Humidity: 84.38 %
20:00 hrs -> Rainfall: 4.32 mm, Humidity: 89.11 %
21:00 hrs -> Rainfall: 2.56 mm, Humidity: 94.39 %
22:00 hrs -> Rainfall: 0.61 mm, Humidity: 100.21 %
23:00 hrs -> Rainfall: -1.55 mm, Humidity: 106.58 %
24:00 hrs -> Rainfall: -3.90 mm, Humidity: 113.50 %
```