

### Text Generation Model

Choose a model

gpt-3.5-turbo-16k

Temperature

0.00

1.00

Use cache

### Data Summarization

Choose a dataset

Choose a dataset

Select a dataset

Upload your own data

Choose a CSV or JSON file

Drag and drop file here  
Limit 200MB per file • CSV, JSON

Browse files

combined-2.csv X

Choose a summarization method

Choose a method

llm

Uses the LLM to generate annotate the default summary, adding details such as semantic types for columns and dataset description

**Goal Selection**

Number of goals to generate

1

10

Add Your Own Goal

Describe Your Goal

show humidity and temperature and plc

**Visualization Library**

Choose a visualization library

# PBL: Adaptive Dashboards Powered by Generative LLMs

*Guide : Dr. Deepak Dharrao*

Dr. Deepak Dharrao

*By: Nouman Jinabade (R&A), Shivansh Chutani (E&TC), Rika Mallika (E&TC)*

Nouman Jinabade

Rika Mallika

Shivansh Chutani

## Summary

		num_unique_values	semantic_type	description	std
0	1/23/23']	362	date		None
1	5.742291666666667	350	temperature		3.165
2	3.958333333333333	304	humidity		13.7547
3		0			None
4		4	disease		None
5		4	humidity		None
6		4	temperature		None
7		4	condition		None
8		4	disease		None

## Goals (4)

Choose a generated goal

show humidity and temperature and plot thresholds >85% humidity and plot temperature 18°C...

```

goals[selected_goal_index] Goal Goal(question='show humidity and temperature
and plot thresholds >85% humidity and plot temperature 18°C - 25°C',
visualization='show humidity and temperature and plot thresholds >85% humidity
and plot temperature 18°C - 25°C', rationale='', index=0)

A visualization goal

index int 0

question str 'show humidity and temperature and plot thresholds >85%
humidity and plot temperature 18°C - 25°C'

rationale str ''

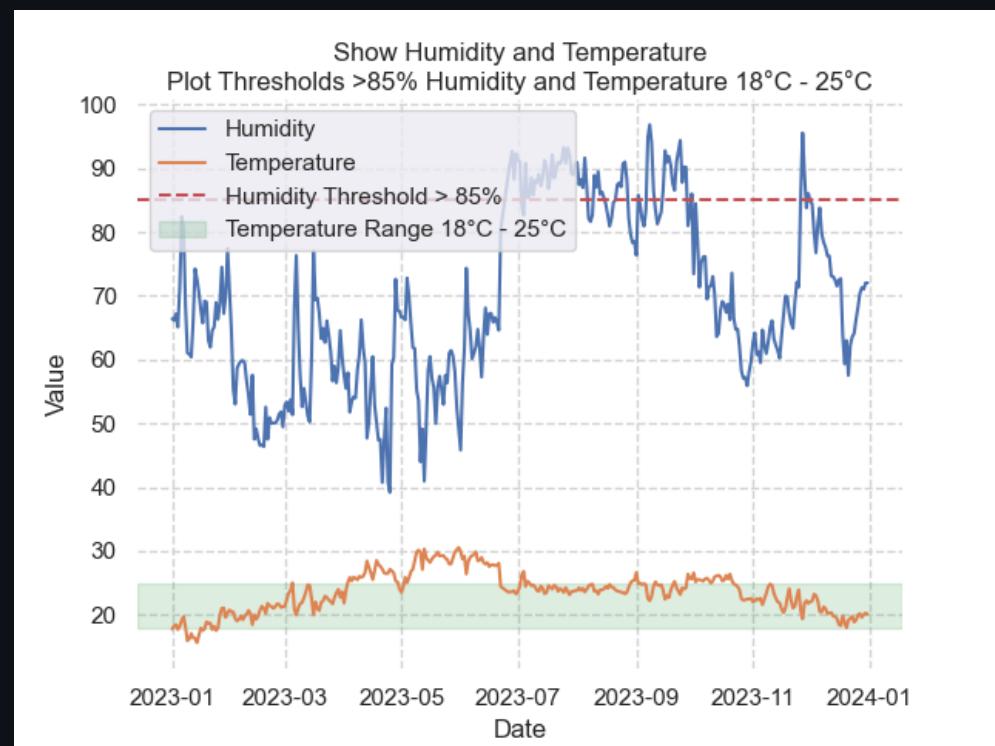
visualization str 'show humidity and temperature and plot thresholds >85%
humidity and plot temperature 18°C - 25°C'

```

## Visualizations

Choose a visualization

Visualization 1



Visualization 1

## Visualization Code

```

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

# solution plan
# i. Convert date field to datetime type
data['Date'] = pd.to_datetime(data['Date'], errors='coerce')

# ii. Drop rows with missing date values
data = data[pd.notna(data['Date'])]

```

```

def plot(data: pd.DataFrame):
    sns.set(style="darkgrid")

    # Plot humidity
    sns.lineplot(x='Date', y='Relative_humidity__Mean_', data=data, label='Humidity')

    # Plot temperature
    sns.lineplot(x='Date', y='Temperature__Mean_', data=data, label='Temperature')

    # Plot threshold for humidity > 85%
    plt.axhline(y=85, color='r', linestyle='--', label='Humidity Threshold > 85%')

    # Plot temperature range 18°C - 25°C
    plt.axhspan(18, 25, color='g', alpha=0.2, label='Temperature Range 18°C - 25°C')

    plt.xlabel('Date')
    plt.ylabel('Value')
    plt.title('Show Humidity and Temperature\nPlot Thresholds >85% Humidity and Te')
    plt.legend()

    return plt

chart = plot(data)

```

## Customize Visualization using NLP

Enter customization instructions (e.g., 'Change color to red')

change graph name to Downey Mildew , and plot only the months of June - October

**Apply Customization**

```

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# solution plan
# i. Convert date field to datetime type
data['Date'] = pd.to_datetime(data['Date'], errors='coerce')

# ii. Drop rows with missing date values
data = data[pd.notna(data['Date'])]

def plot(data: pd.DataFrame):
    sns.set(style="darkgrid")

    # Filter data for months June - October
    data = data[(data['Date'].dt.month >= 6) & (data['Date'].dt.month <= 10)]

    # Plot humidity
    sns.lineplot(x='Date', y='Relative_humidity__Mean_', data=data, label='Humidity')

    # Plot temperature
    sns.lineplot(x='Date', y='Temperature__Mean_', data=data, label='Temperature')

    # Plot threshold for humidity > 85%

```

```
plt.axhline(y=85, color='r', linestyle='--', label='Humidity Threshold > 85%')

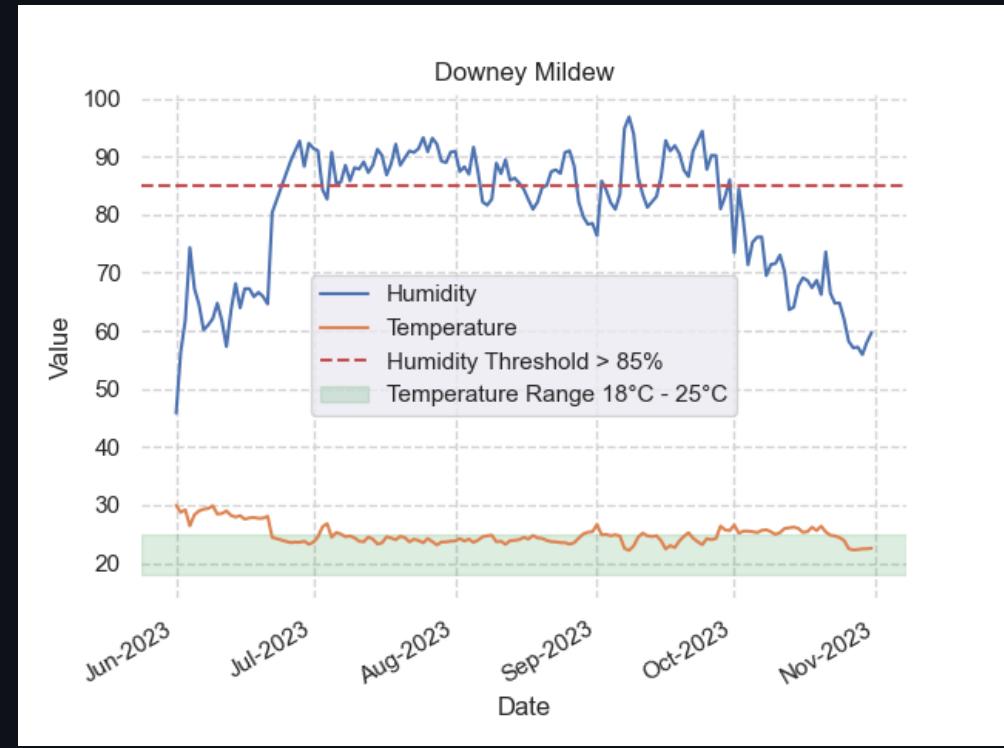
# Plot temperature range 18°C - 25°C
plt.axhspan(18, 25, color='g', alpha=0.2, label='Temperature Range 18°C - 25°C')

plt.xlabel('Date')
plt.ylabel('Value')
plt.title('Downey Mildew', wrap=True)
plt.legend()

# Format x-axis labels as month-year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%b-%Y'))
plt.gcf().autofmt_xdate()

return plt

chart = plot(data)
```



Edited Visualization