

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

combined- 2.csv

combined- 2.csv
15.8KB

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

seaborn

Number of visualizations to generate

1 10

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	5.7422916e	350	temperature		3.165
2	3.9583333e	304	humidity		3.7547
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 <60% humidity and plot temperature 21°C...

```

goals[selected_goal_index] Goal Goal(question='show humidity and temperature
and plot thresholds <60% humidity and plot temperature 21°C - 32°C',
visualization='show humidity and temperature and plot thresholds <60% humidity
and plot temperature 21°C - 32°C', rationale='', index=0)

A visualization goal

index int 0

question str 'show humidity and temperature and plot thresholds <60%
humidity and plot temperature 21°C - 32°C'

rationale str ''

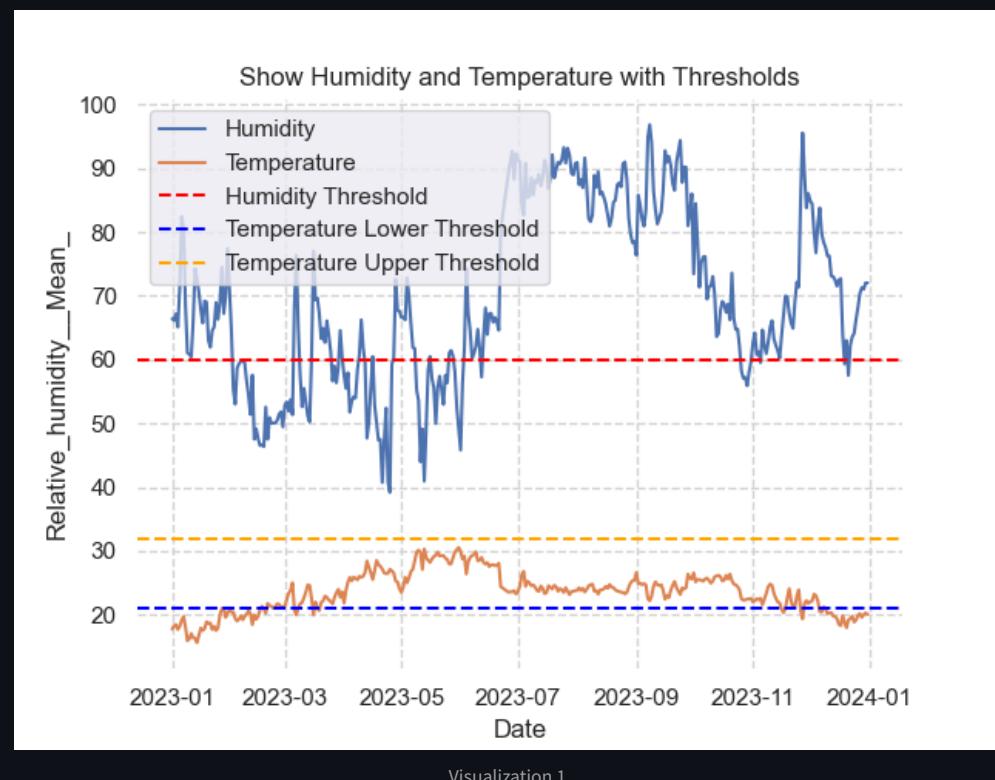
visualization str 'show humidity and temperature and plot thresholds <60%
humidity and plot temperature 21°C - 32°C'

```

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 humidity threshold
    plt.axhline(y=60, color='red', linestyle='--', label='Humidity Threshold')

    # Plot temperature thresholds
    plt.axhline(y=21, color='blue', linestyle='--', label='Temperature Lower Threshold')
    plt.axhline(y=32, color='orange', linestyle='--', label='Temperature Upper Threshold')

    plt.title('Show Humidity and Temperature with Thresholds', wrap=True)
    plt.legend()

    return plt

chart = plot(data)

```

Customize Visualization using NLP

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

change graph name to Powdery Mildew , and plot only the months of Jan - June

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 of Jan - June
    data = data[(data['Date'].dt.month >= 1) & (data['Date'].dt.month <= 6)]

    # 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 humidity threshold
    plt.axhline(y=60, color='red', linestyle='--', label='Humidity Threshold')

```

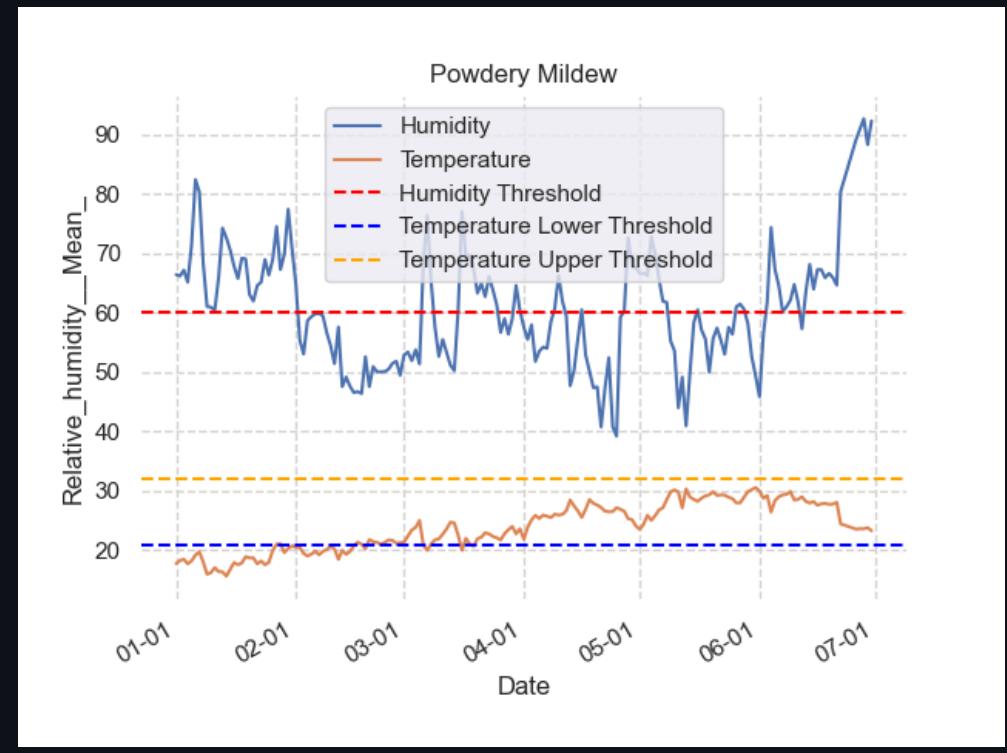
```
# Plot temperature thresholds
plt.axhline(y=21, color='blue', linestyle='--', label='Temperature Lower Thresh')
plt.axhline(y=32, color='orange', linestyle='--', label='Temperature Upper Thresh')

plt.title('Powdery Mildew', wrap=True)
plt.legend()

# Format x-axis labels as month-day
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m-%d'))
plt.gcf().autofmt_xdate()

return plt

chart = plot(data)
```



Edited Visualization