**BLACKBOXAI**

To modify the provided code to use Bayesian optimization without the black-box function and to include the results of the best point in the dataset, the next best point to be explored, and the expected output, we can follow these steps:

1. Remove the black-box function and replace it with a method to evaluate the output based on the existing dataset.
2. Implement the logic to find the next best point to explore based on the Gaussian Process predictions.
3. Plot the results of the optimization process.

Here’s the modified code:

python94 lines

Click to expand

import numpy as np

import pandas as pd

...

**Key Changes Made:**

1. **Removed the Black-box Function**: The black-box function is no longer used. Instead, we predict the output using the Gaussian Process.
2. **Next Best Point**: The next point to explore is determined by the acquisition function, and its predicted output is calculated using the GP model.
3. **Plotting**: A plot is added to visualize the evaluated points and the best score achieved during the optimization process.

Make sure to replace the placeholder for the DataFrame **df** with your actual data before running the code.

the best point should be as close as zero as possible, the outputs are negatives, so cather for that and change the code

**BLACKBOXAI**

The objective is to minimize the absolute value of the sum of negative scores, i.e., find points where the output is as close to zero from below (negative outputs). The current code uses y and evaluates minimum y, but since all outputs are negative, we want the best point that has the output closest to zero (maximum output value since negative values are less than zero).

Therefore, I need to change:

* The target for best point: instead of minimizing y, we want to maximize y (closest to zero, but outputs are negative or zero). So best\_y should track the maximum output (maximum y) since y is negative (e.g. -0.1 > -2.0).
* The Expected Improvement calculation: currently, it assumes minimization. We want to maximize output closer to zero (which means maximizing output because outputs are negative and closer to zero is larger value). So EI should promote points with larger predicted value.
* Predicted outputs from GP are still negative values. We want to maximize them (closest to zero).
* The acquisition function expected\_improvement should be adjusted accordingly to select points maximizing output (maximizing the target).
* Update print statements for best points and the minimum sum of negative scores (which actually becomes maximum output).
* The final plot should reflect outputs on y-axis and show the maximum output as best.

I'll adjust the expected\_improvement function to be for maximization and update the rest accordingly.

After those changes, the results will find the best point with output closest to zero (max output). Also, I will keep all other conditions same.

I will write full complete code with the requested plot and output.