Bandit class:

Add more detailed docstrings to the Bandit class's methods, outlining their purpose and usage.
Consider updating variable nomenclature for clarity and readability.
Check that the class structure follows the PEP 8 requirements.

Epsilon Greedy:

Check that the epsilon decay is working properly.
Give detailed explanations of how epsilon decay is calculated and how it affects exploration vs. exploitation.

Thompson Sampling:

Document how accuracy influences Thompson's Sampling behavior.
Check the precision numbers to make sure they are within acceptable limits.

Utilize Visualization:

Comment out the purpose of each method in the Visualization class.

Handle Errors:

Include exception handling to handle potential errors such as division by zero or improper inputs.

Enhance Logging:

Add meaningful log messages to aid debugging and comprehension of the code's execution flow.

Document the Code:

To document your code, use tools like Pyment or write docstrings.(Optional)
Make sure each class, method, and variable has a detailed description.

Import Statements:

Import statements should be sorted and grouped for better readability.

Unit Testing:

Create unit tests to ensure that each component of your code works properly.

Optimization and Efficiency:

Consider optimizing parts of the code for improved performance.

Example Code Usage:

Provide a detailed example of how to utilize the code.

Final Review:

Go over the code thoroughly to ensure it fits all standards and is well-documented.