**Use Case # 1:**

**Optimize material performances** – The design of materials consists of minimizing, maximizing or targeting specific properties. For example, a goal in alloys design is to determine a set of compositions, temperature and pressure that maximizes functions which depend on phases, volume fractions and on the thermodynamic, structural, dynamics, thermal transport and surface properties.

|  |  |
| --- | --- |
| **User:** | Material Scientist/Engineer/Researcher who has knowledge of the design space/parameters and may have a basic knowledge of Python |
| **Function:** | User feeds the initial data (temperature, pressure, composition, etc.) and parameter bounds into the selected optimizer and runs the program (repeats the process iteratively with newly obtained data) |
| **Results:** | Optimizer is executed iteratively by comparing previously obtained solutions until an optimum solution is found by reducing the distribution over possible functions (posterior). |
| **Component:** | Bayes\_Opt - Bayesian Optimization can be used to guide the choice of experiments during materials design and discovery to find good material designs in as few experiments as possible. Bayesian optimization incorporates prior belief about the unknown function and updates the prior with samples drawn from this unknown function to get a posterior that better approximates the unknown function. The model used for approximating the objective function is called *surrogate model* (Gaussian processes, or GP, is used as the surrogate model for Bayes\_Opt). Bayesian optimization also uses an *acquisition function* that directs sampling to areas where an improvement over the current best observation is likely (for Bayes\_Opt, Upper Confidence Bound, or UCB is used). |

**Use Case # 2:**

|  |  |
| --- | --- |
| **User:** | Material Scientist/Engineer/Researcher who wants to optimize an experiment and prioritizes speed vs accuracy. |
| **Function:** | User selects an optimizer that prioritizes speed vs accuracy, and the algorithm will select the correct optimizer that fits the user’s needs |
| **Results:** | The algorithm will be trained with a few optimizers, so it will pick the one that fits the user’s needs. In this case it will most likely be the random optimizer. |
| **Component:** | The random optimizer samples datapoints and ranks them according to the highest output. It is fast but not accurate.  The algorithm (KNN, SVM, or multilinear regression) that chooses the best optimizer has been trained with data from several optimizers. It will determine which one fits the users needs. |

**Use Case # 3:**

**Use Case # 4:**

**Use Case # 5:**