**IMPERIAL COLLEGE BUSINESS SCHOOL**

**Professional Certificate in Machine Learning and Artificial Intelligence**

**CAPSTONE PROJECT**

**Project Overview**

This capstone project aims to optimize a set of 8 functions using advanced optimization techniques.

I use the approach of Random Search and I also explore Bayesian Optimization with an Upper Confidence Bound acquisition function utilizing the GaussianProcessRegressor from the sklearn library.

The project involves an evaluation of eight functions with different dimensions. Considering all submissions there were some functions that had some improvement except for some which were stuck in a certain area

**Structure**

* input\_data.csv
* output\_data.csv
* notebooks/
* optimization\_results.ipynb
* results
* random\_search\_results.csv
* bayesian\_optimization\_results.csv

**Functions**

The project includes 8 functions that are defined in functions.py. Each function represents a different mathematical expression or model we aim to optimize.

The inputs for each query were the following:

* + Function one: 2-dimensional
  + Function two: 2-dimensional
  + Function three: 3-dimensional
  + Function four: 4-dimensional
  + Function five: 4-dimensional
  + Function six: 5-dimensional
  + Function seven: 6-dimensional
  + Function eight: 8-dimensional

**Optimization Methods**

My focus were in the 2 optimization as below , I also tried to exclude the oldest set of values in my input file and considering the new one however this approach was decommissioned as I had more knowledge about the aim of the project.

**1. Random Search**

Random Search is a simple yet effective optimization technique that involves sampling random points in the parameter space and evaluating the objective function at those points.

* ***Random Search****. Define a search space as a bounded domain of hyperparameter values and randomly sample points in that domain.*
* *Random search is appropriate for discovering new hyperparameter values or new combinations of hyperparameters, often resulting in better performance, although it may take more time to complete.*

At first glance, random searches may seem unappealing. After all, if you can’t test every hyperparameter combination, you are unlikely to find the best one.

However, this approach does come with certain perks.

Firstly, since the random search tests fewer model architectures, it requires less time and less computation to obtain results.

Although the random search may not necessarily find the best possible set of hyperparameters, it can provide a model that comes close to the ideal model in terms of performance and I could see through the leaderboard that they didn’t perform so bad.

**2. Bayesian Optimization**

Bayesian Optimization builds a probability model of the objective function and uses it to select hyperparameter to evaluate in the true objective function.

Bayesian optimization is an approach to optimizing objective functions that take a long time (minutes or hours) to evaluate. It is best suited for optimization over continuous domains of less than 20 dimensions and tolerates stochastic noise in function evaluations. It builds a surrogate for the objective and quantifies the uncertainty in that surrogate using a Bayesian machine learning technique, Gaussian process regression, and then uses an acquisition function defined from this surrogate to decide where to sample. In this capstone the approach uses a Gaussian Process (GP) to model the objective function and an acquisition function to decide where to sample next, in this case it was considered an Upper Confidence Bound acquisition function.