**Capstone Project Competition  
Student name: Victor Acosta  
Student ID: 534**

**Overview**

For the first 7 weeks I used the default parameters and same algorithm for all functions. The code file had one variable file to set the function number, this variable was used to assign different settings and sampling to each function

During these weeks I could observe in the chart the progress of the discovery process and even beta was set for balanced the results were leaning to the (0.9, 0.9) coordinates of the chart.

During weeks 8 and 9 I introduced new kernels to play around with linear and RBF kernels, also I added the option to choose beta (the outputs were shown as explorative, balanced and exploitative) from which I manually chose one based on previous results observations and sent for the weekly submission.

On week 10 and last submission I used the knowledge acquired in Module 24 and switched to Bayesian optimisation giving much better and diverse results.

All details are listed below

**Week 1 to 7**

All default values were used. Default hyperparameters for the GaussianProcessRegressor() object and BETA set to balanced with 1.96. Even the model was using a balanced BETA but we can still see it had a tendency for exploitation in the upper right region.

The following images show the 2D model with can be easy to observe what is going on. For upper dimensions no charts are plotted.

|  |  |
| --- | --- |
| Week 2  A white graph with yellow dots and numbers  Description automatically generated | Week 3  A white sheet with numbers and dots  Description automatically generated |
| Week 5  A white sheet with numbers and dots  Description automatically generated | Week 7 |

The search grid has a radius of 20% from the last results, this is represented in code as discovery\_radius variable, I have the feeling no wider or narrower radius was required as the results were targeting an area

**Week 8**

**Introduced beta changes** for exploitation with Beta = 1.0, balanced Beta = 1.96 and exploration Beta = 2.5  
All results are now displayed, and one will be chosen for the weekly submission form with the following criteria:

* If balanced and exploitation are close, then **balanced** is chosen
* If balanced and exploitation are not close, then **exploratory** results is chosen

**Week 9**

Used only exploration and changed to **polynomial kernel** to see if we can find a different area in the grid with better results. Results are similar, no significant changes in function 1 and 2

A white sheet with numbers and dots

Description automatically generated

**Week 10 and last submission**

After some research and completed Module 24 I moved from manual tuning to **Bayes optimisation** using the **skopt** library.   
I finally achieved independence of same tuning for all, instead each function has its own kernel according to the data gathered.  
As this is the last submission, the approach was all exploitative, but the results have improved specially those previously aiming the corners of the (0,0)-(1,1) space, see results below

0.877228-0.876638 (exploitative / expSineSquared kernel)  
0.681479-0.997980 (exploitative / RationalQuadratic kernel)  
0.396957-0.465740-0.427848 (exploitative / Matern kernel)  
0.387169-0.394453-0.379160-0.425423 (exploitative / RBF kernel)  
0.999999-0.999999-0.999999-0.999999 (exploitative / ExpSineSquared kernel)  
0.376786-0.256616-0.618076-0.796831-0.093563 (exploitative / Matern kernel)  
0.035570-0.201389-0.228024-0.301528-0.326184-0.684137 (exploitative / ExpSineSquared kernel)  
0.086298-0.100834-0.078870-0.088948-0.666280-0.449861-0.207286-0.708571 (exploitative / ExpSineSquared)

A white sheet with numbers and dots

Description automatically generated

**The future of this model**

Variables, parameters and hyperparameters.  
Having the last submission leveraging Bayesian optimisation to find the best kernel and hyperparameters, the rest of the fine tunning of the model goes to the bayes search itself and other variables in the codebase.

If we continue to develop this model the following variables and settings can be adjusted for better results:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable/setting | Current | Next Possible | Impact |
| BayesSearchCV(… n\_iter) | 25 | 1000+ | This will increase running time by possibly hours |
| discovery\_radius | 0.2 | 0.3 | Increase the radius to 30% for each data point |
| sizes | [2, 100],  [3, 80],  [4, 60],  [5, 30],  [6, 15],  [7, 10],  [8, 8] | [2, 256],  [3, 256],  [4, 128],  [5, 128],  [6, 64],  [7, 32],  [8, 16] | The first element in the array is the dimension number. The second element is the number of search point in the “discovery\_radius” range.  Impact: Potential Memory crash |
| beta | 1.0, 1.96 and 2.5 | 0.75, 1.96 and 3.0 | More degree of exploiting and exploring |

**Reflexion if I have start over again**

* Start with more exploration than exploitation, followed by hammering with exploration the best area. (in other words, the opposite of what I did in the project)
* Fine tune the sizes to get more sampling in the search radius and give more time to compute (perhaps hours instead of minutes)
* Move to Bayesian optimisation earlier, perhaps not at the beginning but with enough data to make more meaningful optimisations