1. **[40%] Case: Simulated Annealing and Genetic Algorithm**

Given the following function:

Find its global minimum value and the point (, ) that produces it using **Simulated Annealing** and **Genetic Algorithm**. You can use any library or publicly available code to implement it (please cite the source if you do). Alternatively, you may create your own program. Conduct a number of experiments by tuning its hyperparameters in order to achieve the most optimum result you can get. Please be careful in defining the decision space (search space). Otherwise, you might end up getting result that is far away from optimum. Compare and analyze the results produced by the two algorithms. Submit the code along with your explanation regarding the code and the experimental results, along with your analysis and conclusions. (*Code without explanation and analysis won’t get full mark even if the code produces the correct answer*)

**Answer:**

1. **Simulated Annealing**

**Source :** [**https://machinelearningmastery.com/simulated-annealing-from-scratch-in-python/**](https://machinelearningmastery.com/simulated-annealing-from-scratch-in-python/)

**Import Library Needed**

Graphical user interface, text, application

Description automatically generated

**Defined Objective Function**

Text

Description automatically generated

**Defined Simulated Annealing Function**

Graphical user interface, text

Description automatically generated

**Defined Parameters that can be Tuned**

Text, letter

Description automatically generated

**Example Test of Above Parameter**

Graphical user interface, text, application, email

Description automatically generated

1. **Genetic Algorithm**

**Source :** [**https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/**](https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/)

**Import Library Needed**

Graphical user interface, text, application

Description automatically generated

**Defined Objective Function**

Graphical user interface, text, application

Description automatically generated

**Defined Decode Function**

Text

Description automatically generated

**Defined Selection Function (Using Tournament Selection)**

Text

Description automatically generated

**Defined Cross Over Function**

Text

Description automatically generated

**Defined Mutation Function**

Text

Description automatically generated

**Defined Genetic Algorithm Function**

Text

Description automatically generated

**Defined Parameters that can be Tuned**

Text

Description automatically generated

**Example Test of Above Parameter**

Graphical user interface, text, application, email

Description automatically generated

1. **Experiment Result (Simulated Annealing)**
2. **Bounds [[-1.0, 1.0]]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iterations** | **Temp = 10** | **Temp = 100** | **Temp = 1000** | **Temp = 10000** |
| 100 | -6.249982 | -6.249941 | -6.249941 | -6.249941 |
| 200 | -6.249992 | -6.249941 | -6.249941 | -6.249941 |
| 300 | -6.249996 | -6.249994 | -6.249941 | -6.249941 |
| 400 | -6.249996 | -6.249997 | -6.249941 | -6.249941 |
| 500 | -6.249996 | -6.249997 | -6.249941 | -6.249941 |

1. **Bounds [[-2.0, 2.0]]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iterations** | **Temp = 10** | **Temp = 100** | **Temp = 1000** | **Temp = 10000** |
| 100 | -6.249943 | -6.249924 | -6.231724 | -6.231724 |
| 200 | -6.249990 | -6.249924 | -6.249905 | -6.249905 |
| 300 | -6.249996 | -6.250000 | -6.249905 | -6.249905 |
| 400 | -6.249999 | -6.250000 | -6.249905 | -6.249905 |
| 500 | -6.249999 | -6.250000 | -6.249905 | -6.249905 |

1. **Experiment Result (Genetic Algorithm)**
2. **Bounds [[-1.0, 1.0]], crossover = 0.9, and mutation = 0.5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iterations** | **pop = 100** | **pop = 200** | **pop = 300** | **pop = 400** |
| 100 | -6.250000 | -6.250000 | -6.250000 | -6.250000 |
| 200 | - | -6.250000 | -6.250000 | -6.250000 |
| 300 | - | - | -6.250000 | -6.250000 |
| 400 | - | - | - | -6.250000 |
| 500 | - | - | - | - |

1. **Bounds [[-2.0, 2.0]], crossover = 0.9, and mutation = 0.5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iterations** | **pop = 100** | **pop = 200** | **pop = 300** | **pop = 400** |
| 100 | -6.250000 | -6.250000 | -6.250000 | -6.250000 |
| 200 | - | -6.250000 | -6.250000 | -6.250000 |
| 300 | - | - | -6.250000 | -6.250000 |
| 400 | - | - | - | -6.250000 |
| 500 | - | - | - | - |

1. **Conclusion**

|  |  |  |  |
| --- | --- | --- | --- |
| **Best Parameter** | **Algorithm** | **Point (x) value** | **Global Minimum Value** |
| Bounds [[-2.0, 2.0]], Iterations = 300,  Temp = 100 | Simulated Annealing | -0.25005408 | -6.250000 |
| Bounds [[-1.0, 1.0]], Iterations = 100,  pop = 100 | Genetic  Algorithm | -0.25 | -6.250000 |

**Simulated Annealing Best Parameter Improvement Plotting**

Chart, line chart

Description automatically generated

**Genetic Algorithm Best Result Plotting Improvement Plotting**

Chart, line chart

Description automatically generated

**Analysis & Conclusion :**

Although simulated annealing and genetic algorithm produce the same Global Minimum value, Simulated Annealing requires a wider boundary to achieve this value while Genetic Algorithm is not required. From there, it can be said that if the search is wider, the computation will take longer and will take up more memory as well. It is the same with the number of iterations, where Simulated Annealing requires 300 iterations whereas Genetic only needs 100 times to get the Global Minimum value. **Therefore, Simulated Annealing and Genetic Algorithm can both find the Global Minimum value with the Genetic Algorithm more efficiently and quickly in finding the value**.

1. **[10%] Essay: Artificial Neural Network**

Artificial Neural Network (ANN) can be used for classification and regression. Describe a number of aspects that highlight the architectural differences between ANN used for classification and that used for regression!

**Answer:**

**Source Code :** [**https://machinelearningmastery.com/neural-network-models-for-combined-classification-and-regression/**](https://machinelearningmastery.com/neural-network-models-for-combined-classification-and-regression/)

The task of estimating a mapping function (f) from input variables (X) to discrete output variables (y) is known as classification predictive modeling. Labels or categories are terms used to describe the output variables. The mapping function determines which class or category an observation belongs to.

The task of estimating a mapping function (f) from input variables (X) to continuous output variables (y) is known as regression predictive modeling. Real value are terms used to describe the output variables.

**Import Library Needed**

Text

Description automatically generated

**Read Dataset**

Table

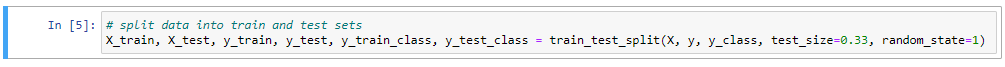
Description automatically generated with medium confidence

**Prepare for Training Model**

Text

Description automatically generated

**Split Dataset into Training and Testing**



**Defined Neural Networks architecture**

Text

Description automatically generated

**Neural Networks Regression Model**

Graphical user interface, text, application

Description automatically generated

**Neural Networks Classification Model**

Graphical user interface, text, application

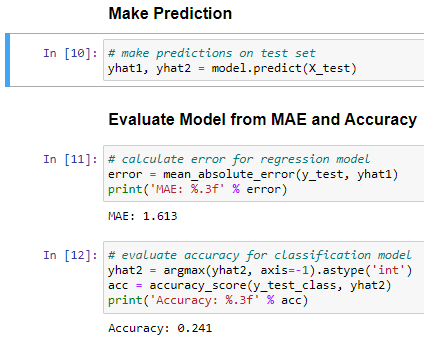
Description automatically generated

**Fitting Model**

Graphical user interface

Description automatically generated with medium confidence

**Make Prediction and Evaluate Model**



**Final Architecture for Regression and Classification**

Diagram

Description automatically generated

From final architecture above, we can see that Regression Model only output 1 output because it’s model for output continuous value like integer. In other hand, Classification Model can output more than 1 output, in this case output 28 classification result because classification model can be more than one type of problem such as “Email Spam”, “Not Spam”, “Promotion”, “Education”, and many more.

1. **[35%] Case: Cluster Analysis Using Fuzzy C-Means**

You are given a dataset obtained from the results of measurements of alcohol in five different concentrations passed through the two-channel QCM sensors, so the dataset has 10 attributes. The dataset can be accessed via the address: <https://archive.ics.uci.edu/ml/machine-learningdatabases/00496/QCM%20Sensor%20Alcohol%20Dataset.zip>.

The dataset consist of 5 types of dataset: QCM3, QCM6, QCM7, QCM10, and QCM12. In each of dataset, there is alcohol classification of five types: 1-octanol, 1-propanol, 2-butanol, 2-propanol, 1-isobutanol. Based on this data, can the sensor be used to identify the five types of alcohol even though they are at different concentrations? For this purpose, you are asked to perform a cluster analysis using fuzzy c-means (FCM). Write down your codes in Jupyter Notebook or Google Colab along with its explanation. You may employ the relevant Libraries. Try your best to do clustering parameter tuning to find the best clustering result. Evaluate the results of the clustering by performing cluster validity, using silhouette plots, purity index, estimating the number of clusters with the appropriate method, and PCA for visualization of data distribution. Report your experiment l, perform analysis on them, and withdraw some relevant conclusions.

**Answer:**

**Source Code References :** [**https://github.com/Amirhossein-Rajabpour/Fuzzy-C-Means-Clustering/blob/main/CI\_Project2.ipynb**](https://github.com/Amirhossein-Rajabpour/Fuzzy-C-Means-Clustering/blob/main/CI_Project2.ipynb)

**Import Library Needed**

Graphical user interface, text

Description automatically generated

**Read Dataset**

Table

Description automatically generated

**Combine Dataset**

Graphical user interface, text, application

Description automatically generated

**Drop Unused Column**

Graphical user interface

Description automatically generated

**Change Header Name into Index Value**

Table

Description automatically generated

**Save Final Combine Dataset Into CSV File**

Graphical user interface, text, application, email

Description automatically generated

**Defined Function to Get Maximum Value in Dataset**

Text

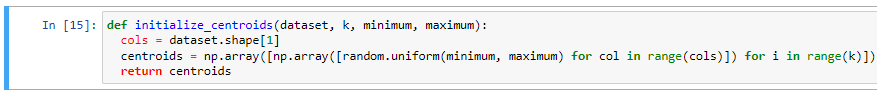
Description automatically generated

**Defined Function to Get Minimum Value in Dataset**

Text

Description automatically generated

**Defined Function to Initialize Centroid**

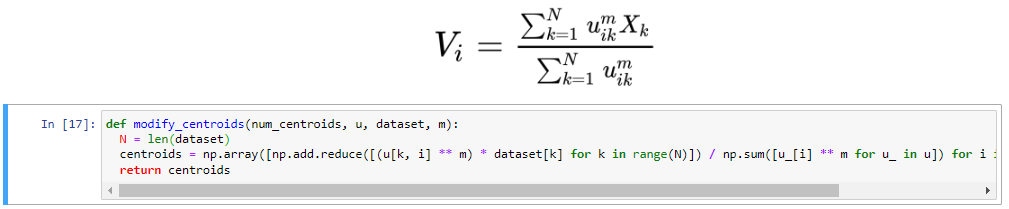


**Defined Function to Calculate fuzziness for Each Point**

Graphical user interface, text, application

Description automatically generated

**Defined Function to Update Centroid Value**



**Defined Function to Calculating Errors**

Graphical user interface, text

Description automatically generated

**Defined Function to Create Fuzzy C-Means Clustering Model**

Text

Description automatically generated

**Defined Function to Plotting Result**

Text, letter

Description automatically generated

**Looping and Plotting Errors for N-Cluster to Find Best Cluster Number**

Graphical user interface

Description automatically generated

**See Plotting Result of Cluster = 3**

Chart, scatter chart

Description automatically generated

**Looping and Plotting Errors for M-Number to Find Best M-Value**

Graphical user interface

Description automatically generated

**See Plotting Result of Fuzzy Clustering**

Chart, scatter chart

Description automatically generated

**Experiment Result ( Hyperparameter Tuning Cluster Number Value)**

|  |  |
| --- | --- |
| **n\_clusters** | **Error Value** |
| 1 | 5349251.173192 |
| 2 | 1322018.370411783 |
| 3 | 552013.7781680705 |
| 4 | 322045.7441320725 |
| 5 | 228161.06340106565 |
| 6 | 175950.1655450468 |
| 7 | 146237.9299553237 |
| 8 | 130563.51580720895 |
| 9 | 101614.2262083058 |

**Experiment Result ( Hyperparameter Tuning M-Value)**

|  |  |
| --- | --- |
| **M\_Value** | **Error Value** |
| 2 | 552013.7781680705 |
| 3 | 273675.31972060935 |
| 4 | 109522.60841735757 |
| 5 | 40286.058213249526 |
| 6 | 14326.817848193461 |
| 7 | 4933.596429214704 |
| 8 | 1683.2837036939657 |
| 9 | 570.9975853004663 |
| 10 | 192.94227544508362 |
| 11 | 65.01872774728781 |

**Analysis & Conclusion :**

Although Fuzzy C-Means Clustering produce High Error value, but with hyperparameter tuning number of cluster and M-Value we can decrease the error value. From the result, we can conclude that the more cluster given, the least error is produced also same conclusion with tuning m-value. Because it more decreased, I choose the error value that much different from before that is when number of cluster is 3 and M-Value is 3. **Therefore, Fuzzy C-Means Clustering can cluster the given dataset with least error with higher value of cluster and m-value**.

1. **[15%] Case: Clustering Using Genetic Algorithm**

Observe thoroughly unlabeled data points as in the figure below. If the genetic algorithm (GA) is used for clustering the data into 3 clusters: (a) write the initial centroids in the representation of a chromosome; (b) Select the best solutions (with the highest fitness values) as parents. You can calculate the fitness value using the objective function as kMeans.

**Answer:**

**Source Code References :** [**https://github.com/anugraharief/Clustering-GeneticAlgorithm/blob/main/Clustering\_GeneticAlgorithm.ipynb**](https://github.com/anugraharief/Clustering-GeneticAlgorithm/blob/main/Clustering_GeneticAlgorithm.ipynb)

**Import Library Needed**

Graphical user interface, text, application

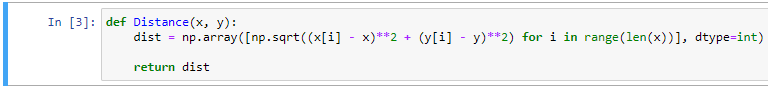
Description automatically generated

**Defined Coordinates Based on Question**

Chart, scatter chart

Description automatically generated

**Defined Function to Get Distance of Each Point**



**Defined Fitness Function to be Improved with Genetic Algorithm**

Text

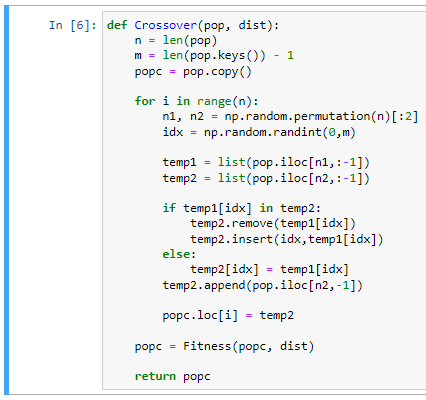
Description automatically generated with medium confidence

**Defined Function to Create Population**

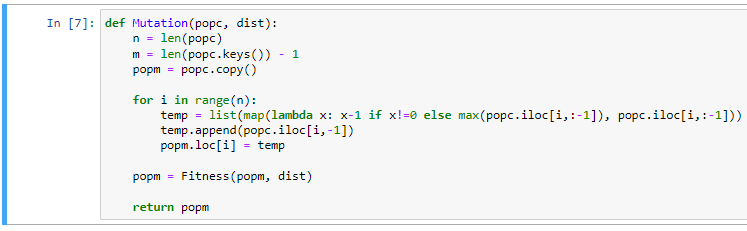
Graphical user interface, text, application

Description automatically generated

**Defined Function to Crossover in Genetic Algorithm**



**Defined Function to Mutation in Genetic Algorithm**



**Defined Function to Selection in Genetic Algorithm**

Text, letter

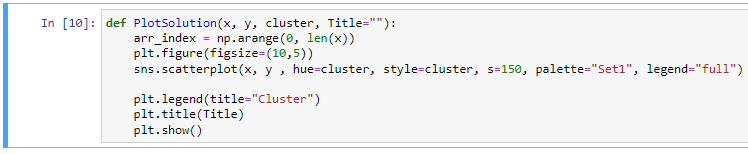
Description automatically generated

**Defined Function to Predict Cluster in Genetic Algorithm**

Graphical user interface, text, application

Description automatically generated

**Defined Function to Plotting Solution**



**Defined Function to Clustering based Genetic Algorithm**

Text

Description automatically generated

**Example Testing**

Chart, scatter chart

Description automatically generated

1. **Write the initial centroids in the representation of a chromosome.**

Graphical user interface, text, application

Description automatically generated

As we can see above, Fitness Function use sum of distance as a metric representation to cluster those point coordinates.

Graphical user interface, text, application, email

Description automatically generated

And distance got from above formula. So we can assume that the Initial Centroid in this case is the Distance between one point to another point coordinates.

1. **Select the best solutions (with the highest fitness values) as parents**

**Experiment Result ( Hyperparameter Tuning Value of Chromosome)**

|  |  |
| --- | --- |
| **n\_chrom** | **Parent Coordinates** |
| 100 | (4, 3) |
| 200 | (1, 1) |
| 300 | (4, 1) |
| 400 | (1, 1) |
| 500 | (4, 1) |

From the result above, we can conclude that best solution as parent either point (1, 1) or point (4, 1).