Universidad de Córdoba
Escuela Politécnica Superior de Córdoba

Computer Science Engineering Degree
Mention: Computation
Fourth year. First quadrimester

Introduction to Computational Models

Assignment 3: Radial basis functions Neural Networks

Ventura Lucena Martínez
i72lumav@uco.es
31008689C

Academic Year 2021-2022
Cordoba, November 30, 2021

Contents

List of Figures

List of Tables

Abstract

This laboratory task serves as an approach to radial basis function neural networks (RBF). In this way, an RBF neural network has been developed, using Python and the *scikit-learn* [1] library. In this sense, it has also served as familiarization with external libraries, widely used in the field of machine learning. An algorithm with the above characteristics has been carried out and in this report the effect of different parameters on a given set of real world data will be analyzed.

1 Implementation of the RBF neural network training algorithm

1.1 Architecture

The RBF neural network models have the following architecture:

- An input layer with as many neurons as input variables the dataset has.
- A hidden layer with a number of neurons specified by the user. We are considering just one hidden layer. The type of all the neurons in the hidden layer will be RBF.
- An output layer with as many neurons as output variables the dataset has.
  - When considering **regression** datasets, all the output neurons will be linear.
  - When considering **classification** datasets, all the output neurons will be softmax. The softmax function is already implemented by the logistic regression algorithm used for adjusting the weights of the output layer.

1.2 Training

1. Application of a clustering algorithm that will serve to establish the centers of the RBF (input-to-output layer weights).
   - For **classification** problems, the centroids initialisation will be random and stratified[1], n_1 patterns.
   - For **regression** problems, n_1 will be randomly selected. After initialising the centroids, the `sklearn.cluster.KMeans` class will be used, with only one centroid initialisation `(n_init)` and a maximum of 500 iterations `(max_iter)`.
2. To adjust the radium of the RBF, a simple heuristic will be applied (the half of the distance average to the rest of the centres). This is, the radium of the j-th neuron will be:

$$\sigma_j = \frac{1}{2 \cdot (n_1 - 1)} \sum_{i \neq j} \|c_j - c_i\| = \frac{1}{2 \cdot (n_1 - 1)} \sum_{i \neq j} \sqrt{\sum_{d=1}^{n} (c_{jd} - c_{id}^2)} \quad (1)$$

3. Learning the weights from hidden-to-output layer:
   - For **regression** problem, it is done using the Moore-Penrose [2] pseudo-inverse. This is:

$$\beta^T_{(n_1+1)\times k} = (\mathbf{R}^+)_{(n_1+1)\times N}\ \mathbf{Y}_{(N\times k)} = \quad (2)$$

[1]For this, `sklearn.model_selection.train_test_split` method has been used.

$$= \left(\mathbf{R}^T_{(n_1+1)\times N} \times \mathbf{R}^T_{N\times(n_1+1)}\right)^{-1} \mathbf{R}^T_{(n_1+1)\times N}\ \mathbf{Y}_{(N\times k)} \quad (3)$$

where $\mathbf{R}$ is the matrix containing the outputs of the RBF neurons, β is a matrix containing a vector of parameters for each of the outputs to be predicted, and $\mathbf{Y}$ is a matrix with the target outputs. To perform these operations, we will use the matrix functions of `numpy` [3], which is a dependence of `scikit-learn`.

- For **classification** problems, it is done using a logistic regression linear model. Using the `sklearn.linear_model.LogisticRegression` class, providing a value for the C parameter in order to apply regularization. Note that in this library what we are specifying is the cost value C (importance of the approximation error versus the regularization error), in such a way that $\eta = \frac{1}{C}$. We will use the $L2$ regularization [4] and the `liblinear` optimisation algorithm.

2 Datasets

We will test different configurations of the neural network and execute each configuration with five seeds (1, 2, 3, 4 and 5). Based on the results obtained, the average and standard deviation of the error will be obtained. For the regression problems, only the *MSE* will be shown. However, for classification problems, the *CCR* (the percentage of correct classified patterns) will be shown, as well as the *MSE*[2].

To assess how the implemented algorithm works, we will run it on:

2.1 Regression datasets

- ***Sin-function* dataset:** this dataset is composed by 120 training patterns and 41 testing patterns. It has been obtained adding some random noise to the sine function.
- ***Quake* dataset [5]:** this dataset is composed by 1633 training patterns and 546 testing patterns. It corresponds to a database in which the objective is to find out the strength of an earthquake (measured on the Richter scale). As input variables, we use the depth of focus, the latitude and longitude at which it occurs.
- ***Parkinsons* dataset [6]:** this dataset is composed by 4406 training patterns and 1469 testing patterns. It contains, as inputs or independent variables, a series of clinical data from patients with Parkinson's disease, including biometric measurement data from their voice. Furthermore, as output or dependent variables, it includes the motor value and the UPDRS (Unified Parkinson's Disease Rating Scale).

[2] In classification, the *MSE* should be obtained converting the class labels to binary values and comparing them against the predicted probabilities, which could be obtained by the predict `proba_method`.

2.2 Classification datasets

- ***Divorce* dataset [7]:** contains 127 training patterns and 43 test patterns. The dataset contains the answer to a series of questions belonging to surveys, with the aim of predicting the divorce of a partner. The answers to the questions are provided in the Likert scale with values from 0 to 4. All the input variables are numerically considered. Two examples of questions are as follows:

  - *I know my spouse's favourite food.*
  - *I can tell you what kind of stress my spouse is facing in her/his life.*

  The dataset contains a total of 54 questions (therefore, 54 input variables) and two categories (0 if there is no divorce, 1 if there is a divorce).

- ***noMNIST* dataset [8]:** originally, this dataset was composed by 200000 training patterns and 10000 test patterns, with a total of 10 classes. Nevertheless, for this lab assignment, the size of the dataset has been reduced in order to reduce the computational cost. In this sense, the dataset is composed by 900 training patterns and 300 test patterns. It includes a set of letters (from a to f) written with different typologies or symbols. They are adjusted to a squared grid of 28×28 pixels. The images are in grey scale in the interval $[-1.0; +1.0]$. Each of the pixels is an input variable (with a total of $28 \times 28 = 784$ input variables) and the class corresponds to a written letter (a, b, c, d, e y f, with a total of 6 classes).

3 Metrics

- **Regression:** average and standard deviation of training and testing *MSE*.

- **Classification:** average and standard deviation of training and testing *CCR*, along with the average and standard deviation of training and testing *MSE*. The *MSE* requested for the classification task is the one obtained when comparing the target output (0 for the wrong classes and 1 for the correct ones) against the predicted probabilities by the model.

4 Experiments

As a guideline, the training and generalisation errors achieved by a linear regression over the three regression datasets; and the training *CCR* and the test *CCR* achieved by a logistic regression (using *Weka*) over the two classification datasets is shown:

- ***Sin* dataset:** $MSE_{train} = 0.02968729$; $MSE_{test} = 0.03636649$.

- ***Quake* dataset:** $MSE_{train} = 0.03020644$; $MSE_{test} = 0.02732409$.

- ***Parkinsons* dataset:** $MSE_{train} = 0.043390$; $MSE_{test} = 0.046354$.
- ***Divorce* dataset:** $CCR_{train} = 90,5512\%$; $CCR_{test} = 90,6977\%$.
- ***noMNIST* dataset:** $CCR_{train} = 80,4444\%$; $CCR_{test} = 82,6667\%$.

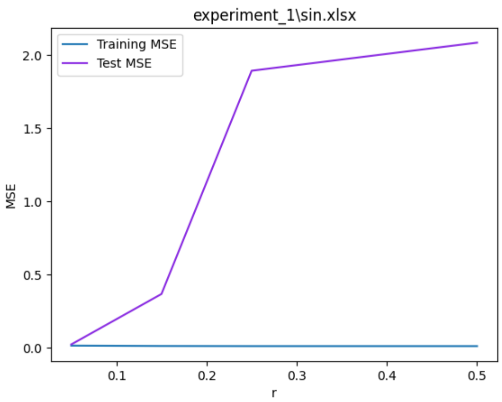
4.1 Experiment 1

For all the datasets, we are considering a number hidden neurons (n_1) equal to the 5%, 15%, 25% and 50% of the total number of patterns of the dataset. In this stage, for classification problems we are using $L1$ regularization and $\eta = 10^{-5}$:

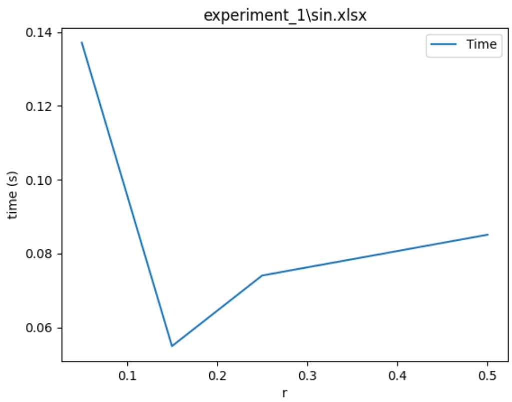
4.1.1 *Sin* dataset

r	**MSE$_{train}$**	**MSE$_{test}$**	**CRR$_{train}$**	**CRR$_{test}$**	**Time (s)**
0.05	0.013817 +- 0.000189	0.022182 +- 0.000245	0.00% +- 0.00%	0.00% +- 0.00%	0.137131929397583
0.15	0.011216 +- 0.000068	0.367074 +- 0.079658	0.00% +- 0.00%	0.00% +- 0.00%	0.054924964904785156
0.25	0.010371 +- 0.000024	1.894359 +- 0.387985	0.00% +- 0.00%	0.00% +- 0.00%	0.07405519485473633
0.5	0.010357 +- 0.000002	2.086239 +- 0.423911	0.00% +- 0.00%	0.00% +- 0.00%	0.08510398864746094

Table 1: Experiment 1 with *Sin* dataset.



(a) *MSE* test w.r.t radius.



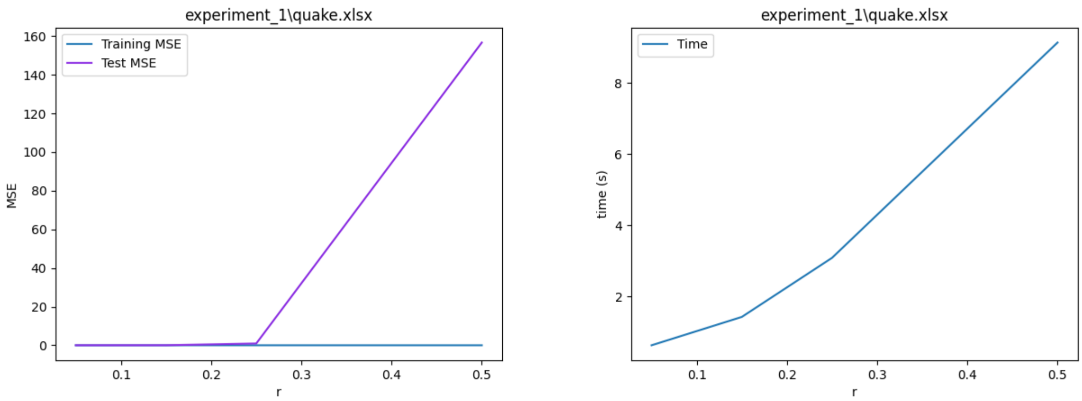
(b) *Timing* test w.r.t radius.

Figure 1: Experiment 1 with *Sin* dataset.

4.1.2 *Quake* dataset

r	MSE$_{train}$	MSE$_{test}$	CRR$_{train}$	CRR$_{test}$	Time (s)
0.05	0.028397 +- 0.000056	0.028377 +- 0.000262	0.00% +- 0.00%	0.00% +- 0.00%	0.6291153430938721
0.15	0.025373 +- 0.000101	0.040941 +- 0.002148	0.00% +- 0.00%	0.00% +- 0.00%	1.4259967803955078
0.25	0.022197 +- 0.000078	0.945664 +- 0.298850	0.00% +- 0.00%	0.00% +- 0.00%	3.0869994163513184
0.5	0.018451 +- 0.000080	156.676762 +- 58.884170	0.00% +- 0.00%	0.00% +- 0.00%	9.141394138336182

Table 2: Experiment 1 with *Quake* dataset.



(a) *MSE* test w.r.t radius.

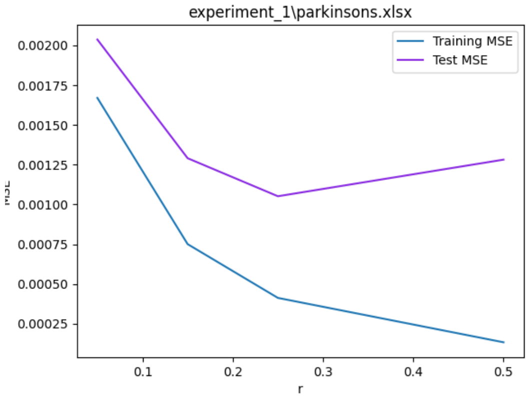
(b) *Timing* test w.r.t radius.

Figure 2: Experiment 1 with *Quake* dataset.

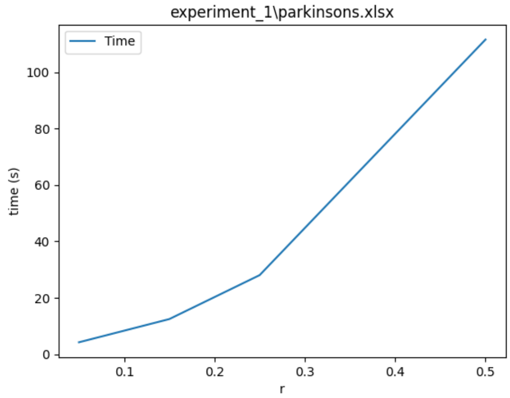
4.1.3 *Parkinsons* dataset

r	MSE$_{train}$	MSE$_{test}$	CRR$_{train}$	CRR$_{test}$	Time (s)
0.05	0.001670 +- 0.000067	0.002037 +- 0.000055	0.00% +- 0.00%	0.00% +- 0.00%	4.264132738113403
0.15	0.000751 +- 0.000010	0.001291 +- 0.000087	0.00% +- 0.00%	0.00% +- 0.00%	12.483939409255981
0.25	0.000413 +- 0.000016	0.001052 +- 0.000057	0.00% +- 0.00%	0.00% +- 0.00%	28.056097745895386
0.5	0.000134 +- 0.000002	0.001282 +- 0.000140	0.00% +- 0.00%	0.00% +- 0.00%	111.52655291557312

Table 3: Experiment 1 with *Parkinsons* dataset.



(a) *MSE* test w.r.t radius.



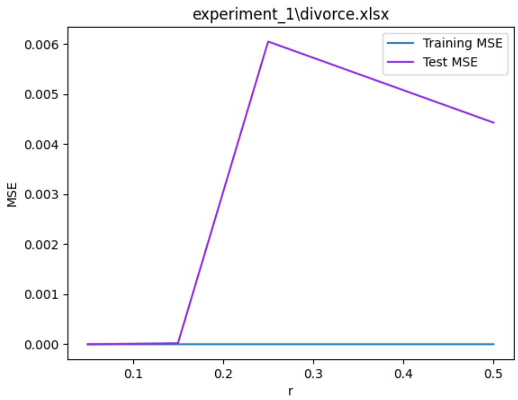
(b) *Timing* test w.r.t radius.

Figure 3: Experiment 1 with *Parkinsons* dataset.

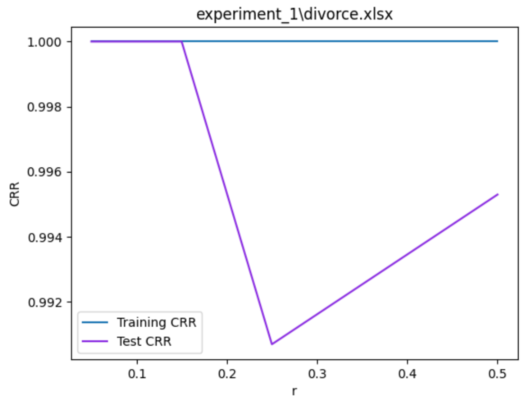
4.1.4 *Divorce* dataset

r	**MSE**$_{train}$	**MSE**$_{test}$	**CRR**$_{train}$	**CRR**$_{test}$	**Time (s)**
0.05	0.000000 +- 0.000000	0.000001 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.14513468742370605
0.15	0.000001 +- 0.000000	0.000022 +- 0.000013	100.00% +- 0.00%	100.00% +- 0.00%	0.1379995346069336
0.25	0.000001 +- 0.000001	0.006049 +- 0.006768	100.00% +- 0.00%	99.07% +- 1.14%	0.1910412311553955
0.5	0.000001 +- 0.000001	0.004429 +- 0.004324	100.00% +- 0.00%	99.53% +- 0.93%	0.18889474868774414

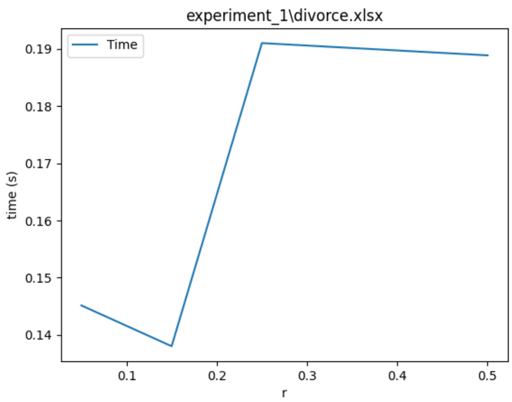
Table 4: Experiment 1 with *Divorce* dataset.



(a) *MSE* test w.r.t radius.



(b) *CRR* test w.r.t radius.



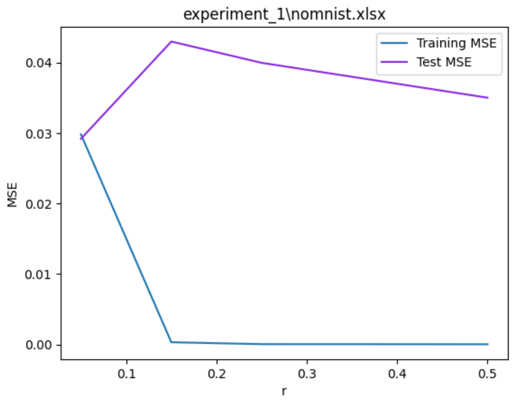
(c) Timing test w.r.t radius.

Figure 4: Experiment 1 with *Divorce* dataset.

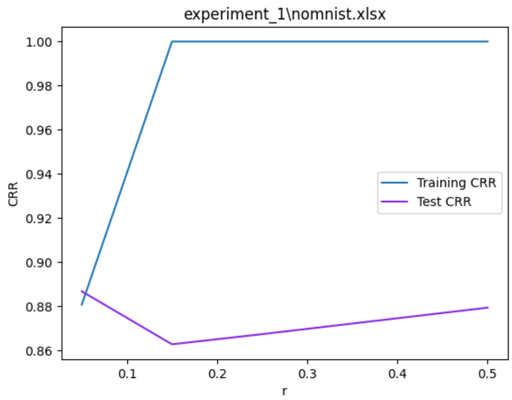
4.1.5 *noMNIST* dataset

r	**MSE**$_{train}$	**MSE**$_{test}$	**CRR**$_{train}$	**CRR**$_{test}$	**Time (s)**
0.05	0.029820 +- 0.001445	0.029193 +- 0.001339	88.07% +- 0.53%	88.67% +- 0.87%	6.4926722049713135
0.15	0.000313 +- 0.000183	0.043002 +- 0.004671	100.00% +- 0.00%	86.27% +- 1.83%	129.31005573272705
0.25	0.000041 +- 0.000009	0.039980 +- 0.004401	100.00% +- 0.00%	86.73% +- 1.58%	59.30211877822876
0.5	0.000021 +- 0.000001	0.035029 +- 0.001868	100.00% +- 0.00%	87.93% +- 0.65%	39.9752733707428

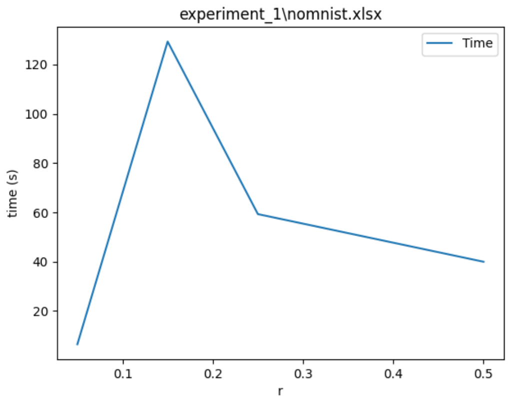
Table 5: Experiment 1 with *noMNIST* dataset.



(a) MSE test w.r.t radius.



(b) CRR test w.r.t radius.



(c) Timing test w.r.t radius.

Figure 5: Experiment 1 with $noMNIST$ dataset.

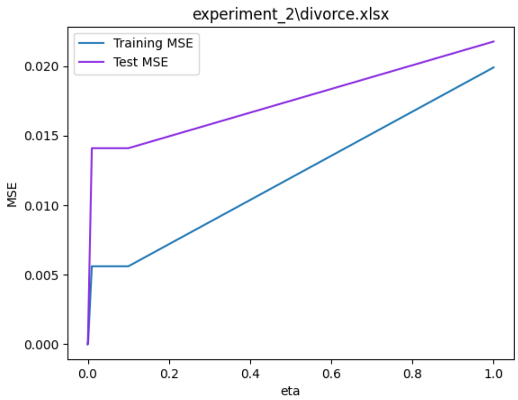
4.2 Experiment 2

For the classification problems with the best architectures, we are trying the following values of η: $\eta = 1$, $eta = 0.1$, ..., $\eta = 10^{-10}$, along with the two types of regulatization ($L1$ and $L2$).

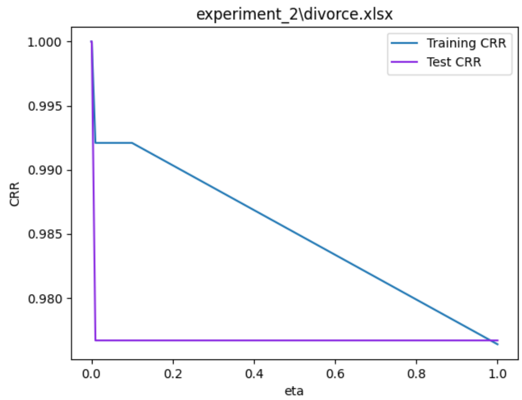
4.2.1 *Divorce* dataset

$\eta/L1$	$\textbf{MSE}_{train}$	$\textbf{MSE}_{test}$	$\textbf{CRR}_{train}$	$\textbf{CRR}_{test}$	**Time (s)**
1	0.019916 +- 0.000734	0.021780 +- 0.000447	97.64% +- 0.00%	97.67% +- 0.00%	0.12507343292236328
10^{-1}	0.005612 +- 0.001197	0.014107 +- 0.001736	99.21% +- 0.00%	97.67% +- 0.00%	0.12192440032958984
10^{-2}	0.005612 +- 0.001197	0.014107 +- 0.001736	99.21% +- 0.00%	97.67% +- 0.00%	0.15794897079467773
10^{-3}	0.000214 +- 0.000110	0.001031 +- 0.001534	100.00% +- 0.00%	100.00% +- 0.00%	0.12201786041259766
10^{-4}	0.000009 +- 0.000007	0.000069 +- 0.000124	100.00% +- 0.00%	100.00% +- 0.00%	0.1210176944732666
10^{-5}	0.000000 +- 0.000000	0.000001 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.14513468742370605
10^{-6}	0.000000 +- 0.000000	0.000000 +- 0.000000	100.00% +- 0.00%	100.00% +- 0.00%	0.12701869010925293
10^{-7}	0.000000 +- 0.000000	0.000000 +- 0.000000	100.00% +- 0.00%	100.00% +- 0.00%	0.1210176944732666
10^{-8}	0.000000 +- 0.000000	0.000000 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.12288355827331543
10^{-9}	0.000000 +- 0.000000	0.000000 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.12214875221252441
10^{-10}	0.000000 +- 0.000000	0.000000 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.12604951858520508

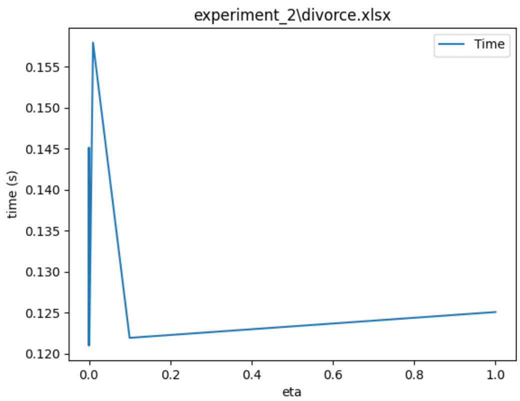
Table 6: Experiment 2 with *Divorce* dataset with $L1$ regularization.



(a) *MSE* test w.r.t η.



(b) *CRR* test w.r.t η.

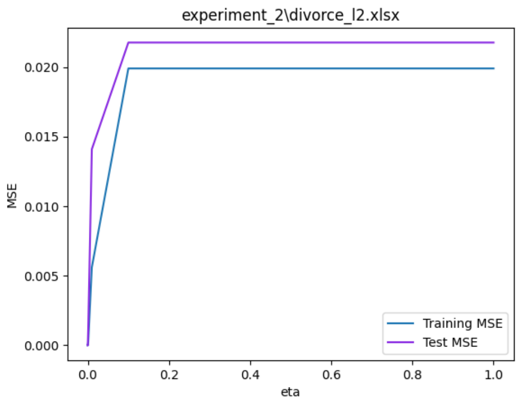


(c) Timing test w.r.t η.

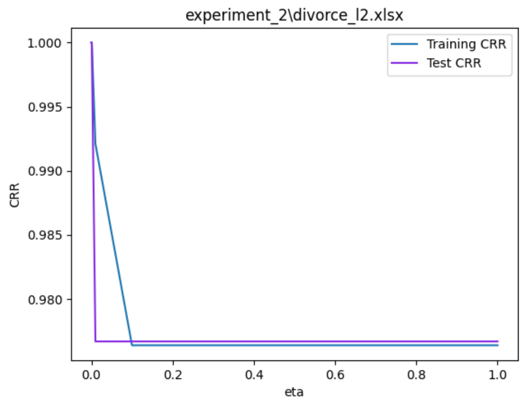
Figure 6: Experiment 2 with *Divorce* dataset with $L1$ regularization.

$\eta/L2$	$\mathbf{MSE}_{train}$	$\mathbf{MSE}_{test}$	$\mathbf{CRR}_{train}$	$\mathbf{CRR}_{test}$	**Time (s)**
1	0.019916 +- 0.000734	0.021780 +- 0.000447	97.64% +- 0.00%	97.67% +- 0.00%	0.19418668746948242
10^{-1}	0.019916 +- 0.000734	0.021780 +- 0.000447	97.64% +- 0.00%	97.67% +- 0.00%	0.16300725936889648
10^{-2}	0.005612 +- 0.001197	0.014107 +- 0.001736	99.21% +- 0.00%	97.67% +- 0.00%	0.16310596466064453
10^{-3}	0.000214 +- 0.000110	0.001031 +- 0.001534	100.00% +- 0.00%	100.00% +- 0.00%	0.1790013313293457
10^{-4}	0.000009 +- 0.000007	0.000069 +- 0.000124	100.00% +- 0.00%	100.00% +- 0.00%	0.13800668716430664
10^{-5}	0.000000 +- 0.000000	0.000001 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.163071870803833
10^{-6}	0.000000 +- 0.000000	0.000000 +- 0.000000	100.00% +- 0.00%	100.00% +- 0.00%	0.16487574577331543
10^{-7}	0.000000 +- 0.000000	0.000001 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.1410045623779297
10^{-8}	0.000000 +- 0.000000	0.000001 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.14603614807128906
10^{-9}	0.000000 +- 0.000000	0.000001 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.153167724609375
10^{-10}	0.000000 +- 0.000000	0.000001 +- 0.000001	100.00% +- 0.00%	100.00% +- 0.00%	0.13900136947631836

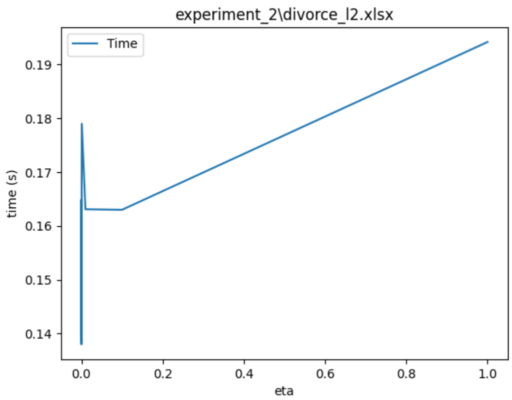
Table 7: Experiment 2 with *Divorce* dataset with $L2$ regularization.



(a) *MSE* test w.r.t η.



(b) *CRR* test w.r.t η.



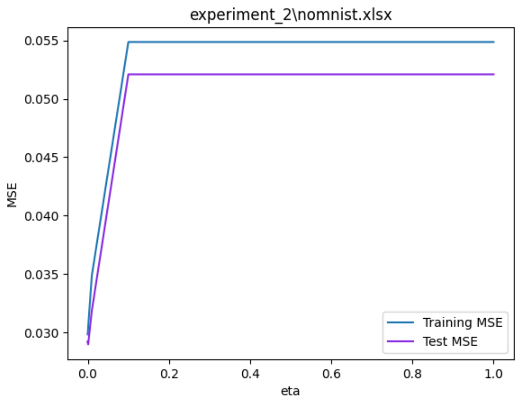
(c) Timing test w.r.t η.

Figure 7: Experiment 2 with *Divorce* dataset with $L2$ regularization.

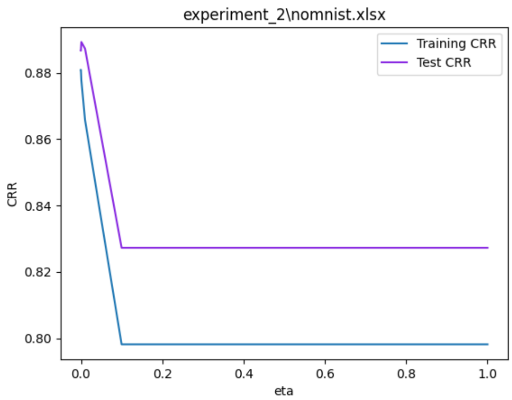
4.2.2 *noMNIST* dataset

$\eta/L1$	**MSE**$_{train}$	**MSE**$_{test}$	**CRR**$_{train}$	**CRR**$_{test}$	**Time (s)**
1	0.054875 +- 0.001214	0.052100 +- 0.001604	79.82% +- 1.02%	82.73% +- 1.65%	2.700671434402466
10^{-1}	0.054875 +- 0.001214	0.052100 +- 0.001604	79.82% +- 1.02%	82.73% +- 1.65%	2.69450044631958
10^{-2}	0.034871 +- 0.001263	0.031852 +- 0.001660	86.58% +- 0.73%	88.73% +- 0.71%	3.3368122577667236
10^{-3}	0.030462 +- 0.001368	0.028968 +- 0.001372	87.78% +- 0.60%	88.93% +- 0.77%	5.009470701217651
10^{-4}	0.029876 +- 0.001435	0.029148 +- 0.001345	88.04% +- 0.52%	88.73% +- 0.90%	5.469960451126099
10^{-5}	0.029820 +- 0.001445	0.029193 +- 0.001339	88.07% +- 0.53%	88.67% +- 0.87%	6.4926722049713135
10^{-6}	0.029815 +- 0.001447	0.029202 +- 0.001339	88.09% +- 0.54%	88.67% +- 0.87%	5.64203667640686
10^{-7}	0.029814 +- 0.001447	0.029203 +- 0.001339	88.09% +- 0.54%	88.67% +- 0.87%	5.796874046325684
10^{-8}	0.029814 +- 0.001447	0.029202 +- 0.001338	88.09% +- 0.54%	88.67% +- 0.87%	5.662917852401733
10^{-9}	0.029814 +- 0.001447	0.029202 +- 0.001338	88.09% +- 0.54%	88.67% +- 0.87%	5.739234447479248
10^{-10}	0.029814 +- 0.001447	0.029202 +- 0.001338	88.09% +- 0.54%	88.67% +- 0.87%	5.772911071777344

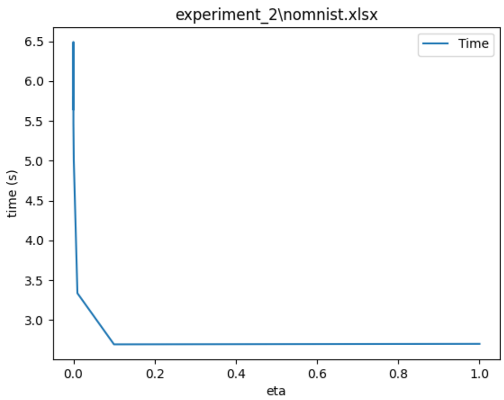
Table 8: Experiment 2 with *noMNIST* dataset with $L1$ regularization.



(a) *MSE* test w.r.t η.



(b) *CRR* test w.r.t η.

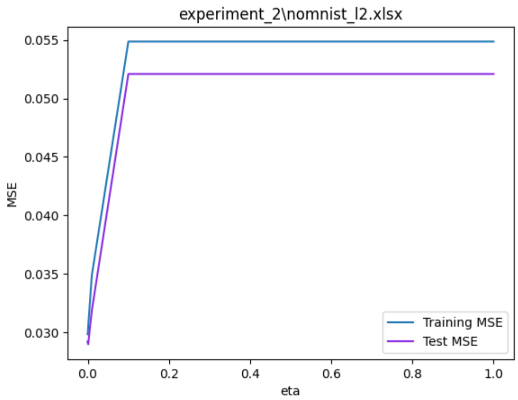


(c) Timing test w.r.t η.

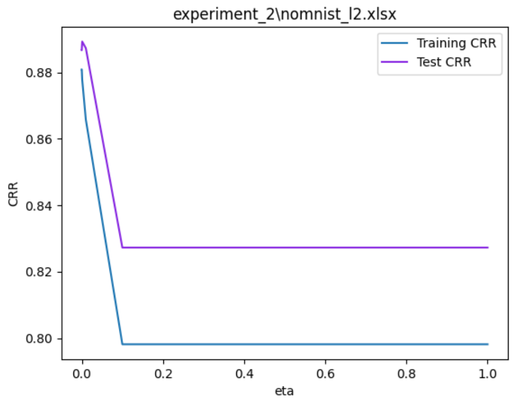
Figure 8: Experiment 2 with *noMNIST* dataset with $L1$ regularization.

$\eta/L2$	**MSE**$_{train}$	**MSE**$_{test}$	**CRR**$_{train}$	**CRR**$_{test}$	**Time (s)**
1	0.054875 +- 0.001214	0.052100 +- 0.001604	79.82% +- 1.02%	82.73% +- 1.65%	3.26466965675354
10^{-1}	0.054875 +- 0.001214	0.052100 +- 0.001604	79.82% +- 1.02%	82.73% +- 1.65%	3.1870884895324707
10^{-2}	0.034871 +- 0.001263	0.031852 +- 0.001660	86.58% +- 0.73%	88.73% +- 0.71%	3.8533949851989746
10^{-3}	0.030462 +- 0.001368	0.028968 +- 0.001372	87.78% +- 0.60%	88.93% +- 0.77%	5.787272214889526
10^{-4}	0.029876 +- 0.001435	0.029148 +- 0.001345	88.04% +- 0.52%	88.73% +- 0.90%	6.589781761169434
10^{-5}	0.029820 +- 0.001445	0.029193 +- 0.001339	88.07% +- 0.53%	88.67% +- 0.87%	6.54505467414856
10^{-6}	0.029815 +- 0.001447	0.029202 +- 0.001339	88.09% +- 0.54%	88.67% +- 0.87%	6.644064903259277
10^{-7}	0.029814 +- 0.001447	0.029203 +- 0.001339	88.09% +- 0.54%	88.67% +- 0.87%	6.761331081390381
10^{-8}	0.029814 +- 0.001447	0.029202 +- 0.001338	88.09% +- 0.54%	88.67% +- 0.87%	6.769020318984985
10^{-9}	0.029814 +- 0.001447	0.029202 +- 0.001338	88.09% +- 0.54%	88.67% +- 0.87%	6.605666875839233
10^{-10}	0.029814 +- 0.001447	0.029202 +- 0.001338	88.09% +- 0.54%	88.67% +- 0.87%	6.680197238922119

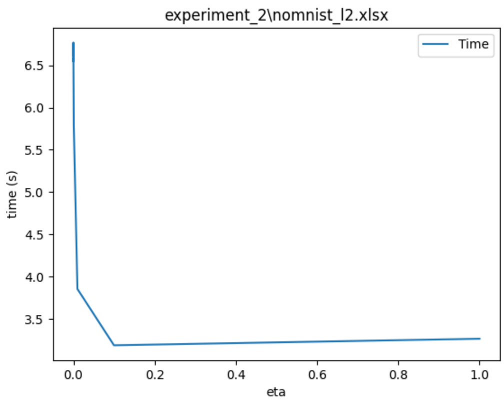
Table 9: Experiment 2 with *noMNIST* dataset with $L2$ regularization.



(a) *MSE* test w.r.t η.



(b) *CRR* test w.r.t η.



(c) Timing test w.r.t η.

Figure 9: Experiment 2 with *noMNIST* dataset with $L2$ regularization.

As we can see in the *noMNIST* data, both regularization, $L1$ and $L2$, reach to the same values, except the time, where we can see an increse of 15.41% on the $L2$:

$$\overline{L1}_{time} = 4.938005599(s) \tag{4}$$

$$\overline{L2}_{time} = 5.698867473(s) \tag{5}$$

$$\overline{R}_{time} = \frac{\overline{L2}_{time}}{\overline{L1}_{time}} = \frac{5.698867473(s)}{4.938005599(s)} = 1.154082829 = 15.41\% \tag{6}$$

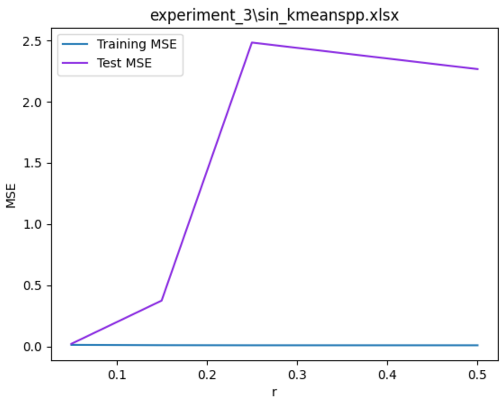
4.3 Experiment 3

For both, regression and classification problems, we are comparing the results obtained using the initialisation proposed for the `sklearn.cluster.KMeans` algorithm (using both the best architecture and the configuration for the logistic regression) according to the `"k-means++"` initialisation.

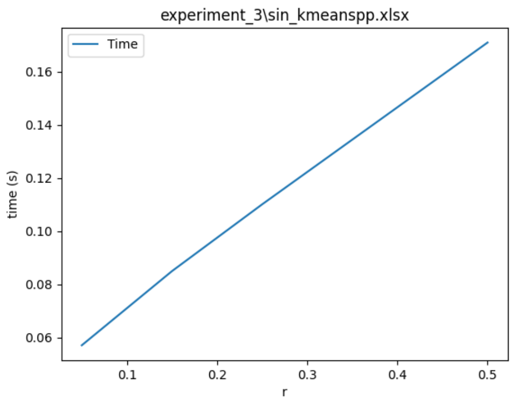
4.3.1 *Sin* dataset

r/kmeans++	MSE$_{train}$	MSE$_{test}$	CRR$_{train}$	CRR$_{test}$	Time (s)
0.05	0.013800 +- 0.000108	0.022301 +- 0.000292	0.00% +- 0.00%	0.00% +- 0.00%	0.05713152885437012
0.15	0.011197 +- 0.000017	0.375319 +- 0.022855	0.00% +- 0.00%	0.00% +- 0.00%	0.08500051498413086
0.25	0.010361 +- 0.000001	2.482909 +- 0.397155	0.00% +- 0.00%	0.00% +- 0.00%	0.1101071834564209
0.5	0.010359 +- 0.000003	2.266105 +- 0.308700	0.00% +- 0.00%	0.00% +- 0.00%	0.17093110084533691

Table 10: Experiment 3 with *Sin* dataset and `k-means++`.



(a) *MSE* test w.r.t radius.



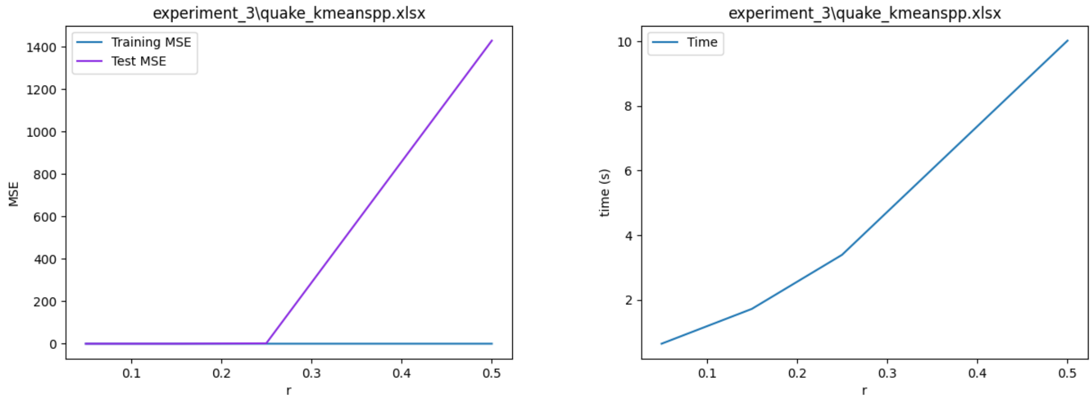
(b) Timing test w.r.t radius.

Figure 10: Experiment 3 with *Sin* dataset and `k-means++`.

4.3.2 *Quake* dataset

r/kmeans++	MSE_{train}	MSE_{test}	CRR_{train}	CRR_{test}	Time (s)
0.05	0.028221 +- 0.000047	0.028686 +- 0.000103	0.00% +- 0.00%	0.00% +- 0.00%	0.6498491764068604
0.15	0.025518 +- 0.000077	0.050181 +- 0.002693	0.00% +- 0.00%	0.00% +- 0.00%	1.7251231670379639
0.25	0.022101 +- 0.000099	1.763023 +- 0.424420	0.00% +- 0.00%	0.00% +- 0.00%	3.3961498737335205
0.5	0.017828 +- 0.000051	1429.013203 +- 715.031778	0.00% +- 0.00%	0.00% +- 0.00%	10.017149209976196

Table 11: Experiment 3 with *Quake* dataset and `k-means++`.



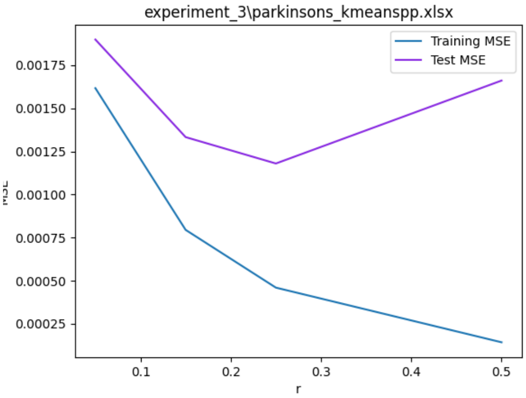
(a) *MSE* test w.r.t radius. (b) Timing test w.r.t radius.

Figure 11: Experiment 3 with *Quake* dataset and `k-means++`.

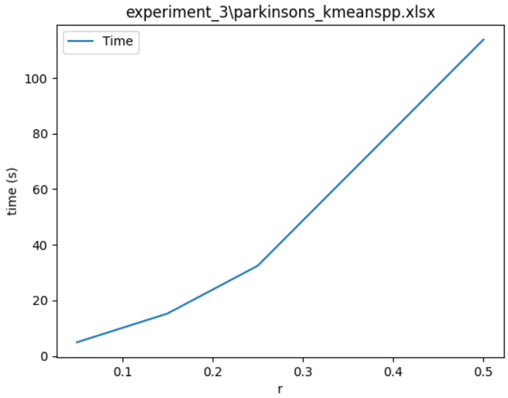
4.3.3 *Parkinsons* dataset

r/kmeans++	MSE_{train}	MSE_{test}	CRR_{train}	CRR_{test}	Time (s)
0.05	0.001617 +- 0.000068	0.001899 +- 0.000073	0.00% +- 0.00%	0.00% +- 0.00%	4.923632621765137
0.15	0.000795 +- 0.000019	0.001333 +- 0.000027	0.00% +- 0.00%	0.00% +- 0.00%	15.2344331741333
0.25	0.000460 +- 0.000015	0.001180 +- 0.000065	0.00% +- 0.00%	0.00% +- 0.00%	32.43939661979675
0.5	0.000143 +- 0.000003	0.001661 +- 0.000210	0.00% +- 0.00%	0.00% +- 0.00%	113.83208703994751

Table 12: Experiment 3 with *Parkinsons* dataset and `k-means++`.



(a) *MSE* test w.r.t radius.



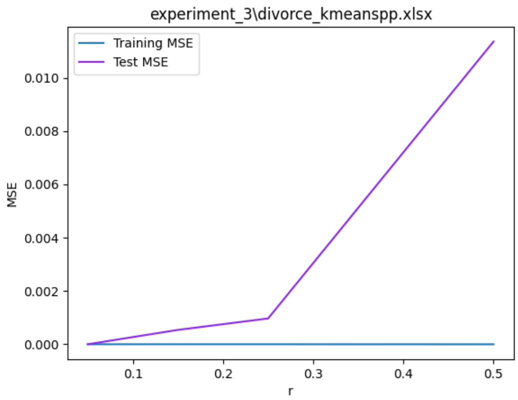
(b) Timing test w.r.t radius.

Figure 12: Experiment 3 with *Parkinsons* dataset and `k-means++`.

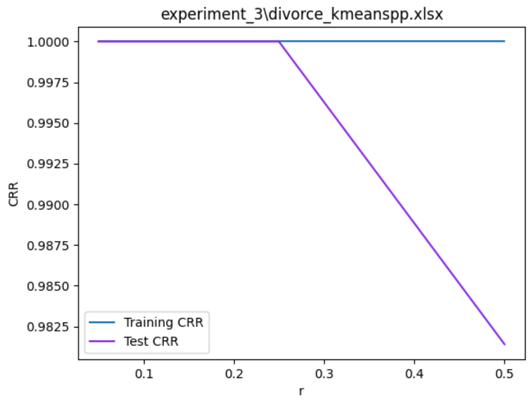
4.3.4 *Divorce* dataset

r/kmeans++	$\mathbf{MSE}_{train}$	$\mathbf{MSE}_{test}$	$\mathbf{CRR}_{train}$	$\mathbf{CRR}_{test}$	Time (s)
0.05	0.000001 +- 0.000000	0.000001 +- 0.000003	100.00% +- 0.00%	100.00% +- 0.00%	0.1320028305053711
0.15	0.000001 +- 0.000001	0.000540 +- 0.000990	100.00% +- 0.00%	100.00% +- 0.00%	0.16803383827209473
0.25	0.000001 +- 0.000001	0.000969 +- 0.000817	100.00% +- 0.00%	100.00% +- 0.00%	0.2030014991760254
0.5	0.000000 +- 0.000000	0.011357 +- 0.004014	100.00% +- 0.00%	98.14% +- 0.93%	0.2810065746307373

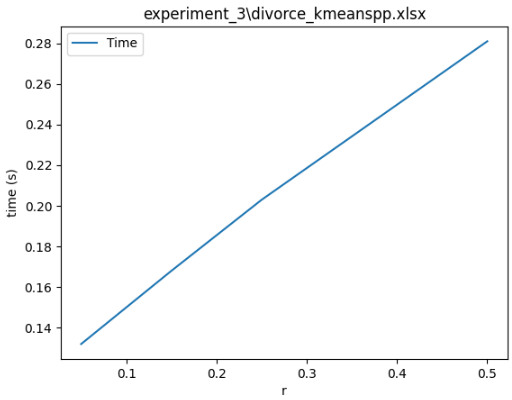
Table 13: Experiment 3 with *Divorce* dataset and `k-means++`.



(a) *MSE* test w.r.t radius.



(b) *MSE* test w.r.t radius.



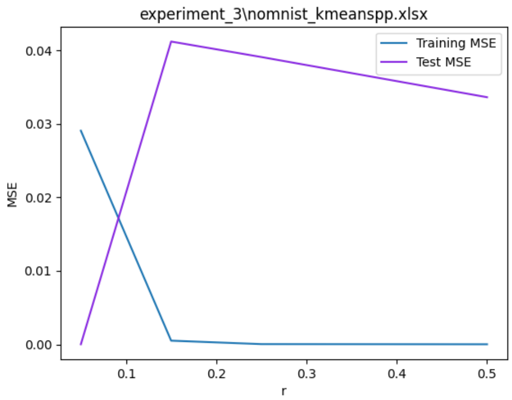
(c) Timing test w.r.t radius.

Figure 13: Experiment 3 with *Divorce* dataset and `k-means++`.

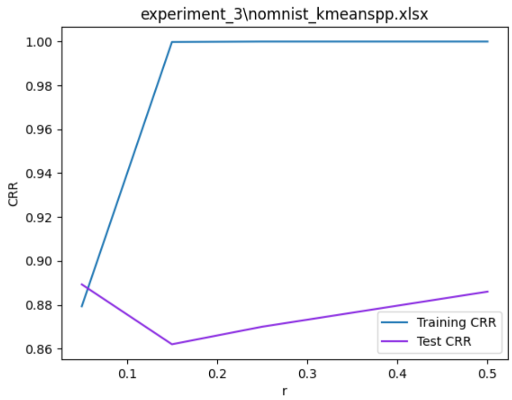
4.3.5 *noMNIST* dataset

r/kmeans++	**MSE**$_{train}$	**MSE**$_{test}$	**CRR**$_{train}$	**CRR**$_{test}$	**Time (s)**
0.05	0.029083 +- 0.000598	0.029126 +- 0.000645	87.93% +- 0.31%	88.93% +- 0.83%	6.534244537353516
0.15	0.000513 +- 0.000314	0.041217 +- 0.003090	99.98% +- 0.04%	86.20% +- 0.81%	129.389488697052
0.25	0.000041 +- 0.000004	0.039108 +- 0.003161	100.00% +- 0.00%	87.00% +- 0.87%	68.76031231880188
0.5	0.000019 +- 0.000001	0.033640 +- 0.001637	100.00% +- 0.00%	88.60% +- 0.65%	45.85815215110779

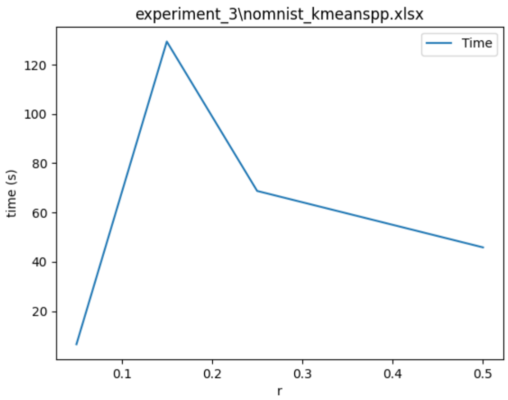
Table 14: Experiment 3 with *noMNIST* dataset and `k-means++`.



(a) *MSE* test w.r.t radius.



(b) *MSE* test w.r.t radius.



(c) Timing test w.r.t radius.

Figure 14: Experiment 3 with *noMNIST* dataset and `k-means++`.

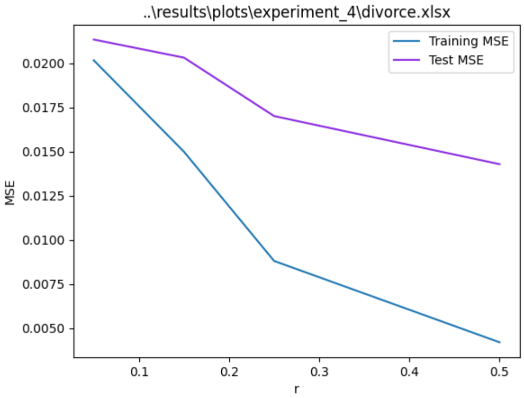
4.4 Experiment 4

Finally, for any of the classification problems, we are running the script considering the problem as if it was regression (i.e. the classification parameter is *False* and compute the *CCR* rounding the predictions to the closest integer).

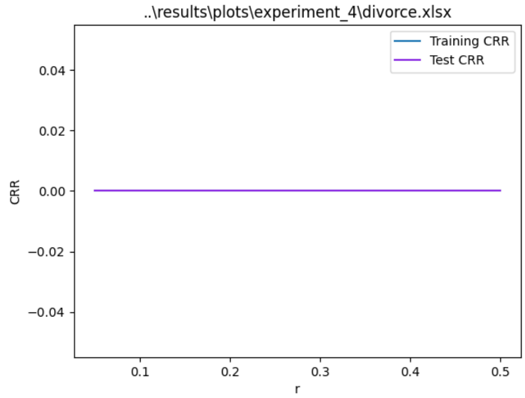
4.4.1 *Divorce* dataset

r	$\mathbf{MSE}_{train}$	$\mathbf{MSE}_{test}$	$\mathbf{CRR}_{train}$	$\mathbf{CRR}_{test}$	Time (s)
0.05	0.020157 +- 0.001765	0.021333 +- 0.001161	0.00% +- 0.00%	0.00% +- 0.00%	0.13402986526489258
0.15	0.014985 +- 0.001344	0.020309 +- 0.000922	0.00% +- 0.00%	0.00% +- 0.00%	0.1472311019897461
0.25	0.008801 +- 0.001986	0.017001 +- 0.001192	0.00% +- 0.00%	0.00% +- 0.00%	0.11499977111816406
0.5	0.004199 +- 0.001354	0.014279 +- 0.001674	0.00% +- 0.00%	0.00% +- 0.00%	0.15999817848205566

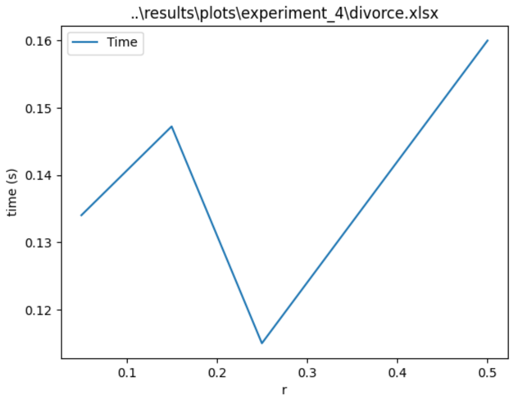
Table 15: Experiment 4 with *Divorce* dataset.



(a) *MSE* test w.r.t radius.



(b) *CRR* test w.r.t radius.



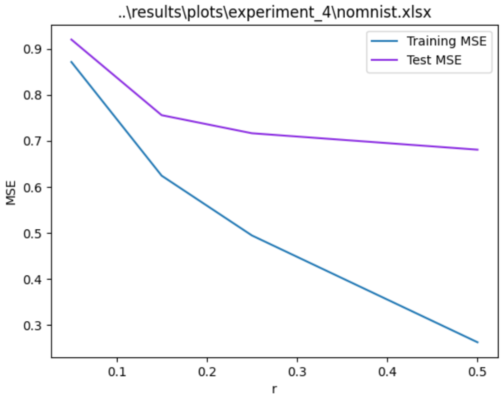
(c) Timing test w.r.t radius.

Figure 15: Experiment 4 with *Divorce* dataset.

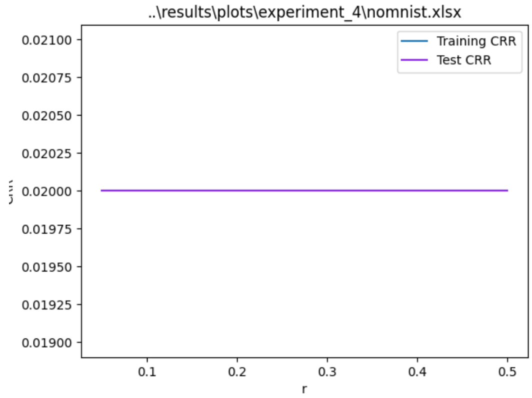
4.4.2 *noMNIST* dataset

r	MSE_{train}	MSE_{test}	CRR_{train}	CRR_{test}	Time (s)
0.05	0.871545 +- 0.032224	0.920240 +- 0.043723	2.00% +- 0.00%	2.00% +- 0.00%	2.7128288745880127
0.15	0.624652 +- 0.021231	0.756030 +- 0.033329	2.00% +- 0.00%	2.00% +- 0.00%	3.082817316055298
0.25	0.494868 +- 0.026250	0.716757 +- 0.010562	2.00% +- 0.00%	2.00% +- 0.00%	3.6642508506774902
0.5	0.262882 +- 0.014401	0.681099 +- 0.014404	2.00% +- 0.00%	2.00% +- 0.00%	6.402103424072266

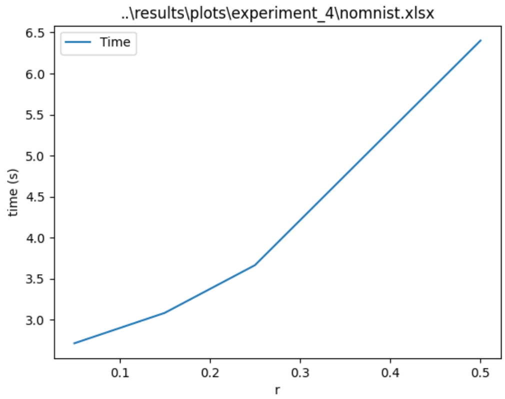
Table 16: Experiment 4 with *noMNIST* dataset.



(a) *MSE* test w.r.t radius.



(b) *CRR* test w.r.t radius.



(c) Timing test w.r.t radius.

Figure 16: Experiment 4 with *noMNIST* dataset.

5 Conclusion

The experiments carried out are generally good. The MSE and CRR rate are lower and over, repectively, than the Weka's given thresholds, and they are also reached in what we could consider a good timing, all of this discarding some outliers as we can see in tables 2, 16 and 11 (all of them marked with red colour).

Once obtained these results, we have selected the best ones of each experiment and dataset, considering time as a determining value in case the metrics have similar values. For experiment 1 (section 4.1) and 3 (section 4.3), we have considered the following architectures as the best ones in which the value of $r = 0.05$ appears the 90%:

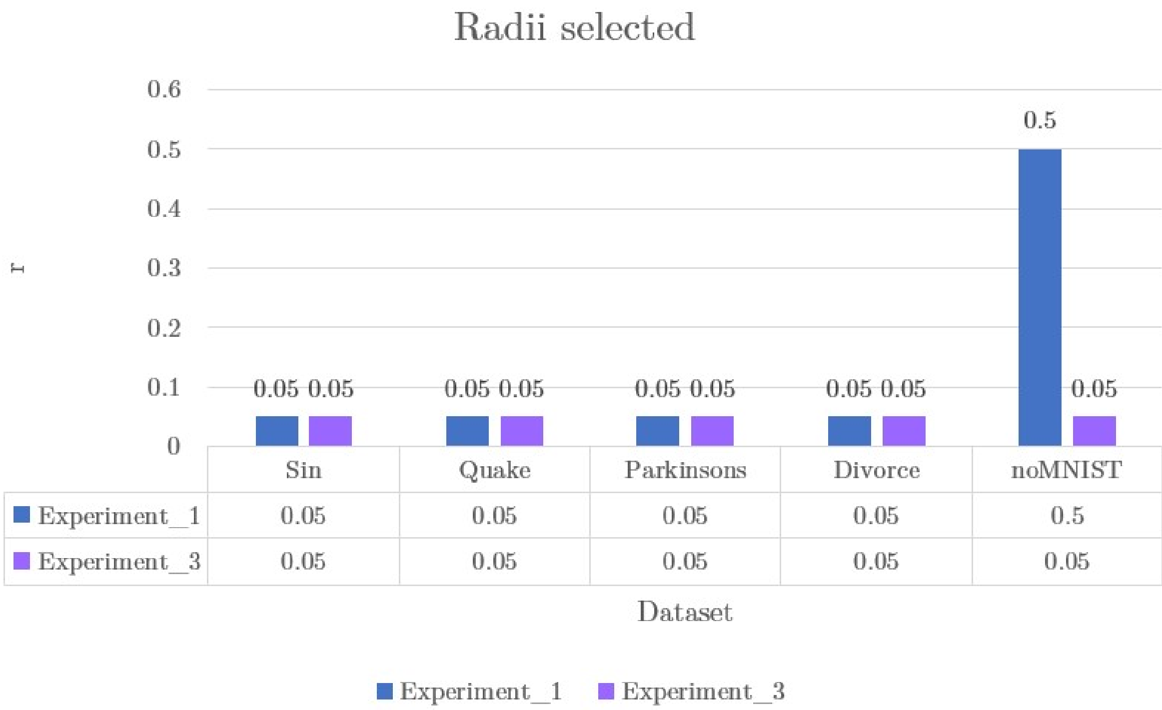


Figure 17: Experiments 1 and 3: Best architectures.

In the case of *noMNIST* dataset, the best confusion matrix according to experiment 1 (section 4.1) is as follows:

$$CM = \begin{array}{cc} & \begin{matrix} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} & \mathbf{e} & \mathbf{f} \end{matrix} \\ \begin{matrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \\ \mathbf{e} \\ \mathbf{f} \end{matrix} & \begin{bmatrix} 46 & 1 & 1 & 0 & 0 & 2 \\ 1 & 44 & 0 & 2 & 3 & 0 \\ 0 & 0 & 45 & 1 & 3 & 1 \\ 2 & 6 & 0 & 42 & 0 & 0 \\ 0 & 0 & 1 & 0 & 47 & 2 \\ 2 & 1 & 0 & 0 & 1 & 46 \end{bmatrix} \end{array}$$

- **Training MSE:** 0.028227.
- **Test MSE:** 0.026723.
- **Training CRR:** 88.22%.
- **Test CRR:** 90.00%.
- **Time:** 6.4926722049713135 s.

Comparing the computational time between experiments of *noMIST* dataset of assignment 2 and 3 (the best architectures):

- **Assignment 2:** 721.434852521 s.
- **Assignment 3:** 6.4926722049713135 s.

Which is an increase of 11116.17% or 111.1616 times higher.

Moving to experiment 2 (section 4.2), we can observe that the values are not quite scattered, reaching the best values when $\eta <= 10^{-5}$ with *Divorce* dataset and $\eta <= 10^{-3}$ with *noMNIST* dataset. As we commented before, regularization $L2$ has a time increase of 15.41% with respect to regularization $L1$, this means that $L1$ is faster.

References

[1] F. Pedregosa **andothers**. “Scikit-learn: Machine Learning in Python”. **in***Journal of Machine Learning Research*: 12 (2011), **pages** 2825–2830.

[2] Wikipedia contributors. *Moore–Penrose inverse — Wikipedia, The Free Encyclopedia.* [Online; accessed 27-November-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Moore%E2%80%93Penrose_inverse&oldid=1048805043.

[3] Charles R. Harris **andothers**. “Array programming with NumPy”. **in***Nature*: 585.7825 (**september** 2020), **pages** 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[4] James McCaffrey. *Test Run - L1 and L2 Regularization for Machine Learning.* Microsoft. 2015. URL: https://docs.microsoft.com/en-us/archive/msdn-magazine/2015/february/test-run-l1-and-l2-regularization-for-machine-learning.

[5] *Quake dataset.* Knowledge Extraction based on Evolutionary Learning. URL: https://sci2s.ugr.es/keel/dataset.php?cod=75.

[6] Athanasios. Tsanas **and** Max. Little. *Parkinsons Telemonitoring Data Set.* URL: http://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring.

[7] M. Yöntem, K. Adem, T. İlhan **and** S. Kılıçarslan. *Divorce prediction using correlation based feature selection and artificial neural networks.* Nevşehir Hacı Bektaş Veli University SBE Dergisi. 2019. URL: https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set.

[8] *notMNIST dataset.* 2011. URL: http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html.