Instituto Superior Técnico

MEEC

Computação Inteligente

Projeto Intermédio

Prediction using NN and Fuzzy Systems

Trabalho realizado por:
- João Gomes, 96248, joaomgomes2001@tecnico.ulisboa.pt
- Miguel Vicente, 96288, miguel.mendes.vicente@tecnico.ulisboa.pt

2022/2023
1ºSemestre, 1º Período

1 Introduction

This project had as objective the prediction of how many people would be inside a lab with 3 workstations, using *Neural Network* and *Fuzzy Systems*.

It was made available a imbalaced dataset, acquired by installed sensors, the information about the temperature(1 sensor in each workstation), luminosity(1 sensor in each workstation), CO_2 levels, moviments detectores(2 of them) and the number of persons in the room. In this report are explained in detail the decisions that the group made while developing this project, and the reasons behind them.

2 Brute Force Approach

In a first instance, the group made a neural network with a hidden layer of 3 neurons, value selected based in the rules of thumb 1, 2, 3, given in the theoretic classes.

1. Size of Input Layer $<$ Size of Hidden Layer $>$ Size of Output Layer
2. Size of Hidden Layer $= \frac{2}{3}$ Size of Input Layer + Size of Output Layer.
3. Size of Hidden Layer $<$ 2x Size of Input Layer

To train the NN, train data (70% of the total data received) was used , and a prediction was made with the test set (the remaining 30%). As it can be observed by the Confusion Matrix (Figure 1), the results were catastrophic, with the NN predicting that would never be 1 person in the room, which is far from the truth, because a total of 468 (the least out of the 4 possible results), were there in the time period.

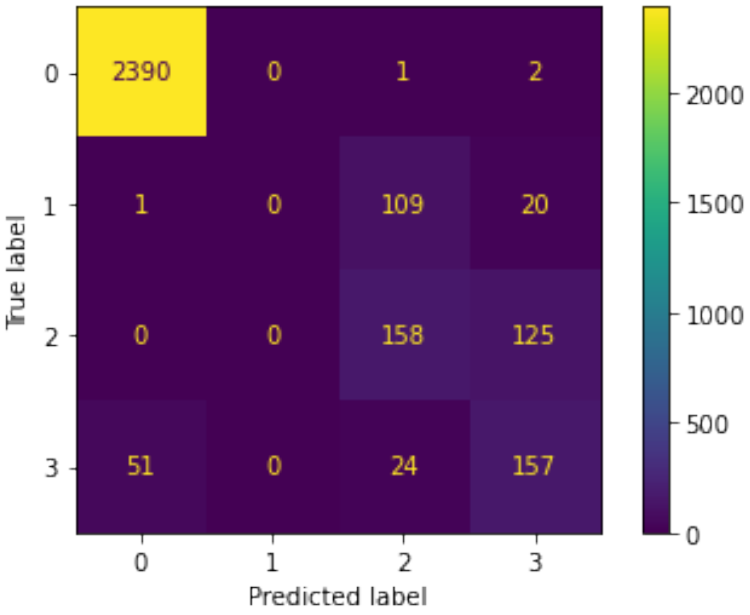

Figura 1: Confusion Matrix - Brute Force Approach

This failure in the prediction can be explained due to various problems. The data received was highly imbalanced towards 0 persons, the only pre-processing done in the data was after the removing of the 4 rows with null values, out of the 10125 left, the count of persons were:

Number of Persons	Count
0	7994
1	468
2	882
3	781

This will lead the NN to have the tendency of predict the output to be zero, as the majority of the data it has been trained on has this output. Another problem is that this NN, thanks to the sensors the data is a little bit "delayed"i.e. the levels of Co2 and temperature only increase/decrease after a time after someone enter/leaves the lab, so since this NN only has access to data that is not temporally organized in this way, it cannot predict as good as we would like. Having the following classification report:

	Precision	Recall	F1-score	Support
0	0.98	1.00	0.99	2393
1	0.00	0.00	0.00	130
2	0.54	0.56	0.55	383
3	0.52	0.68	0.59	232
Accuracy			0.89	3038
macro avg	0.51	0.56	0.53	3038
weighted avg	0.86	0.89	0.87	3038

with values which will be discussed later on this report.

3 Classification/Validation of a Model

For the kind of brute approach tried in the previous section, it was obvious that the NN was not performing well at all. However, for the rest of the project, if we wanted to tune the Hyper-parameters and see the effect of new features implemented, we had to be careful choosing the best way to classify the performance of the NN. In that sense, there were three different matters taking into consideration.

3.1 Choosing the best classification measure

Some of the usual measures for performance classification, are Accuracy, Precision and Recall. Of course the ideal was to have the maximum score in every one of them, but we decided maximizing recall was the main objective of the Hyper-parameters tuning, because we cannot dissociate the problem with reality, while precision refers to the percentage of your results which are relevant, recall refers to the percentage of total relevant results correctly classified by your algorithm. Once this data were taken during a world pandemic, if we wanted to make a system to be implemented for prevention/recall measures in the real world it would be much better to know when the students were violating the 2 persons limit, then to fail to predict it.

3.2 Average of random bias and weight initialization

The initial weights and bias of the NN can be started randomly, by setting the variable randomstate to some number. Instead of evaluating the performance of the NN for one single randomstate, which could lead to "false" results, if the NN got stuck in a bad local minimum, we took the average performance of the n random states (set by the variable RD STATE NUM)

3.3 Cross-Validation

We can risk to over-fit the model if we tune the hyper-parameters to perform well in a single test set. Since a good model is not the one that gives accurate predictions on the known data or training data but the one which gives good predictions on the new data and avoids overfitting and underfitting. We choose to use k-fold cross validation to solve this problem, after some tests we decided to go with a k=5, so the dataset will be divided into 5 equal parts and the process will run 5 times, each time with a different holdout set, as showed in Figure 6 and then we took the average of the scores, which is called the cross-validation accuracy and it will serve as our performance metric for the model.

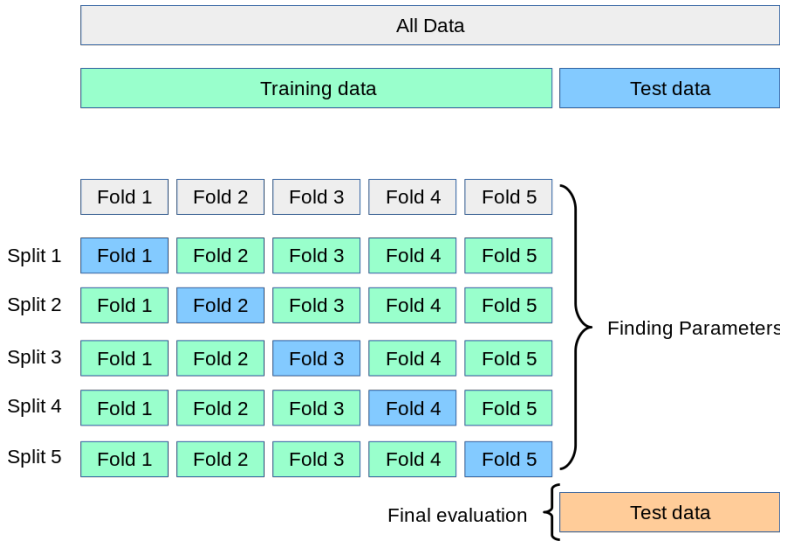

Figura 2: K-Fold Cross Validation Example

4 Creating of NN Model

We had the objective to define the best possible NN model with the best results and without overfitting or underfitting.

To do that we started pre-processing the data, removing the outliers using a function to detect and remove them, we choose to remove the ones that were $k * \sigma$ a far from the rest of the data, and choose a $k = 6$, after that we removed the rows with null values. Since the data given to us had 10129 rows, and the outliers+null values were only 8, we only removed them and did not replace them with any know technique.

The next thing we decided were to balanced the data, since having a balanced data set for a model would generate higher accuracy models, higher balanced accuracy and balanced detection rate. Hence, its important to have a balanced data set for a classification model. So after some consideration, and some testing with various values, we took the night data, i.e, the data that was taken between 8PM-Midnight and 2AM-8AM, this is only because during that time Tecnico was closed, there were no one in the room, that data were only messing with the NN statistics and guiding the model towards the wrong direction. The fact that we leave the data between Midnight and 2AM, was merely so the model could have some more insights of that occurrences(data of 0 persons in the lab), without it the model had very little of it, which ended with way better results, i.e. cross-validation score and better statistics.

Finally we balanced the data using the Min-max normalization, so it could help training our neural networks as the different features are on a similar scale, which helps to stabilize the gradient descent step, allowing us to use larger learning rates or help models converge faster for a given learning rate.

To improve even more the results and the model, we delayed the CO2 data because when someone enter the room it does not increase or decrease right a way, but it takes some time. Being 50 minutes the time that gave the better results.

After all this processing, the data were finally ready to be used by the model. We choose to use a "standart"MLP, after train test split the data. With the help of the cross validation score, explained in the last point, we were able to tune the hyper-parameters so the NN could predict the number of persons in the room very accurately. The hyper-parameters used were: Solver='adam', a Activation Function='relu', a RandomState=42 (explained before), and two hidden layers with 7 and 5 neurons respectively(following the rule of thumb mencioned earlier).

The Confusion Matrix of the model were:

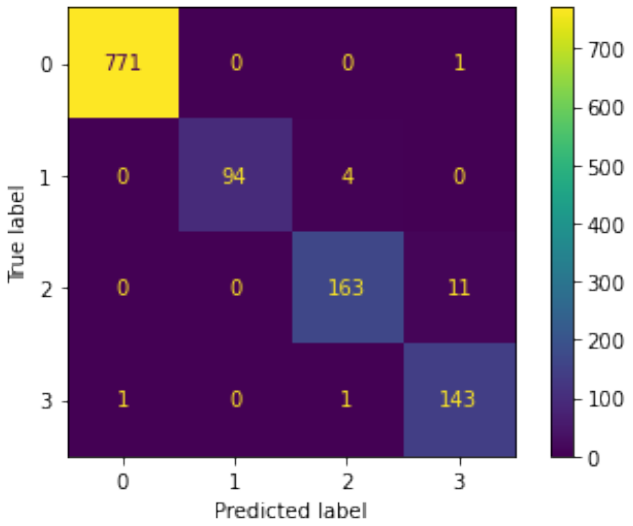

Figura 3: Confusion Matrix

The classification report:

	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	772
1	1.00	0.96	0.98	98
2	0.97	0.94	0.95	174
3	0.92	0.99	0.95	145
Accuracy			0.98	1189
macro avg	0.97	0.97	0.97	1189
weighted avg	0.98	0.98	0.98	1189

With a cross-validation score of 0.9333538638416687. All this statistics were fantastic, way better than the brute force approach, which shows the importance of all this pre-processing and analysis of the data before feeding it to the NN. With such a high score we can be almost sure that the model will not have a overfitting problem and can be used with different dataframes and have the same good results.

5 Classification using Fuzzy Systems

In this section is detailed the implementation of the Fuzzy System to predict when are more than 2 persons inside the room.

Firstly, we divide the data set in two sub-sets: analogous to changing an MLP's hyper-parameters, we should have a separate test set to ensure the system is not over-fit. So we decide to divide our data set in two parts: 80% of train set and 20% of test set.

5.1 Tuning Fuzzy sets and Rules

When we start thinking about which fuzzy sets we might create to obtain good results, we start analysing the CO2 data. After analysing that, we decided to create a new feature that gives the variation of CO2 levels in the past hour and a half. As we can see in Figure 4 the number of persons inside the room exceed two persons when the CO2 variation levels are greater than 0.15.

Another feature that we decided to implement after analysing the data from the light sensors, was the sum of the levels of light of the three sensors. Doing that, as we can see in Figure 5, the number of persons inside the room exceed two persons when the sum of Light levels are greater than 0.6.

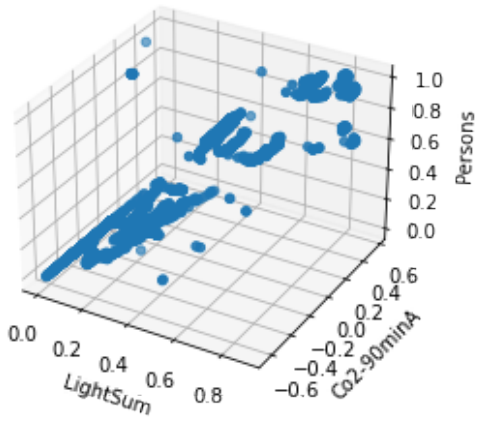

Figura 4: CO2 Variation

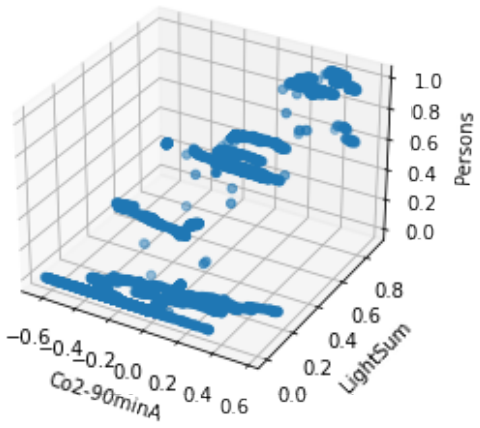

Figura 5: Sum of Light

After creating this two features we tried to implement the Fuzzy System with just this two features but we weren't achieving the values that we pretended. So we decided to search for another feature on the data set.

The last one was a feature mixing the time of the day, the light and the Co2 variation. Putting it simple we checked if the S3Light sensor, the one closer to the only window in the lab, was picking up more light than the other two together, this would mean that we were during the day(lab opened), so adding the CO2 variation, i.e, if it were rising fast that would mean that someone were in the room, probably with the lights off, using only the sun light to see, so we put that rows with a value equal to 1 and the other to 0. If it were too negative(< -1)

that would mean that someone were with the lights on in the workstations 1 and 2, and using the light of the sun near the window(S3). This would help in the cases that people dont turn the lights. After adding this feature the results improved a lot.

Membership functions were drawn for antecedents and consequences. It was advised during the theoretical classes that trapezoids with an overlap of 25% to 50% of it's base were a good starting point. The CO2 Variation Levels and Sum of Light Levels are divided in five categories and the Weather is divided in two while the final output is only divided in two, as we can see on Figure 6. A base size and an arbitrary overlap are used respecting the imposed rules and facilitating the math.

Then, the antecedents and consequences were defined. Most of the rules are trivial and the borders of 0.15 (for CO2 Variation Levels) and 0.6 (for Sum of Light Levels) are the points that deserve most of our attention.

We defined that for the CO2 Variation Levels, any value within the "Increasing a little" and "Increasing a lot" ranges would be considered more than two persons inside the room.

We defined that for the the Sum of light, any value within the "High" and "Very High" ranges would be considered more than two persons inside the room.

We also considerate that if occurs any value within the "Constant"range, for the CO2 Variation, and any value within the "Low", "Very Low", "Medium", "Very High", that are ranges defined for the Sum of light, and in addition occurs any value within "Good", that is a range defined for the Weather, would be considered more than two persons inside the room.

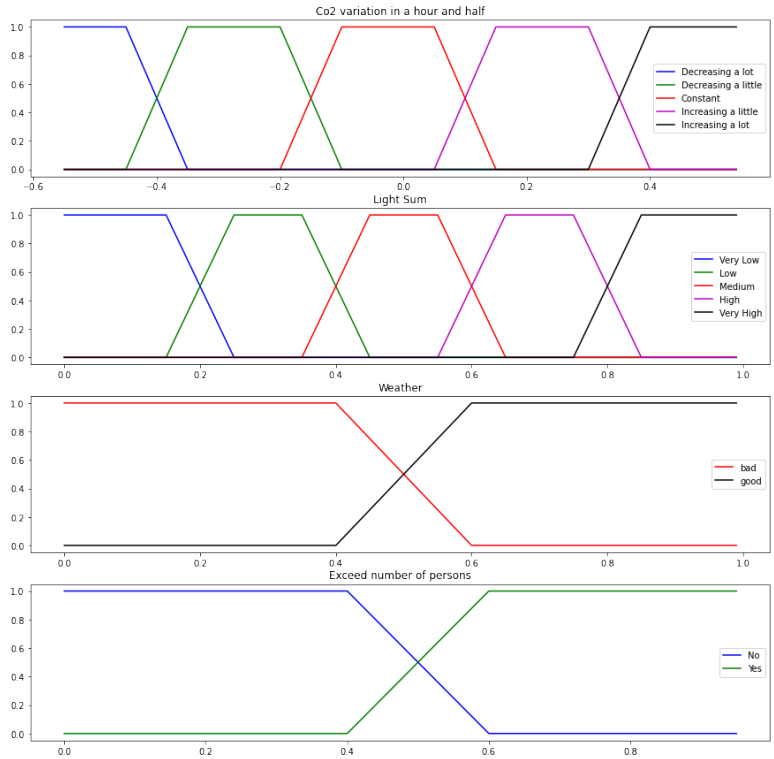

Figura 6: Membership Functions

Here are the rules for the "Yes-case"summarized:

- LightSum = "Very High"
- LightSum = "High"
- CO2Var = "Constant"**and** LightSum = "Very Low"**and** Weather = "Good"
- CO2Var = "Constant"**and** and LightSum = "Low"**and** Weather = "Good"
- CO2Var = "Constant"**and** LightSum = "Medium"**and** Weather = "Good"
- CO2Var = "Increasing a little"
- CO2Var = "Increasing a lot"

5.2 Tuning the Threshold to activate binary output

Then the train data was applied to the implemented Fuzzy system, which generated the defuzzified values from 0 to 1. Now it is still necessary to obtain a threshold to activate the final binary "exceed two persons"result (yes-1 or no-0). By observing the outputs (Figure 7), it is possible to draw a rule for the best threshold: for outputs > 0.57, the output is considered to be 1 and consequently there are more than two persons in the room. Otherwise, the output is considered to be 0 and consequently there isn't three person in the room.

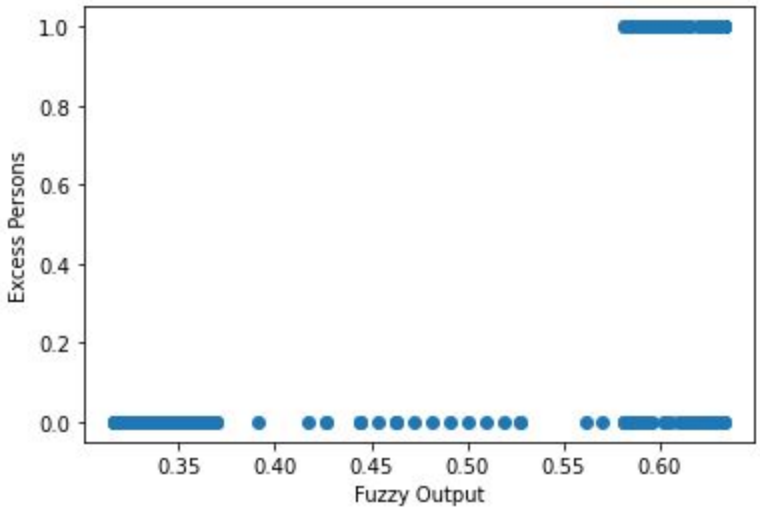

Figura 7: Generated/Real Output

5.3 Performance Evaluation

After using the first 80% of the data to formulate the Fuzzy system and tuning it, the remaining 20% was used to test the model. After running the model, the following confusion matrix was drawn, Figure 1. The system did not have a perfect output response, however we obtain a good value of Recall and decent values of F1-Score, the worse parameter was the Precision that we can only obtain a value of 0.6, as we can see in table 1.

Tabela 1: Values of Precision/Recall/F1-Score

Precision	Recall	F1-Score
0.6	1	0.749(9)

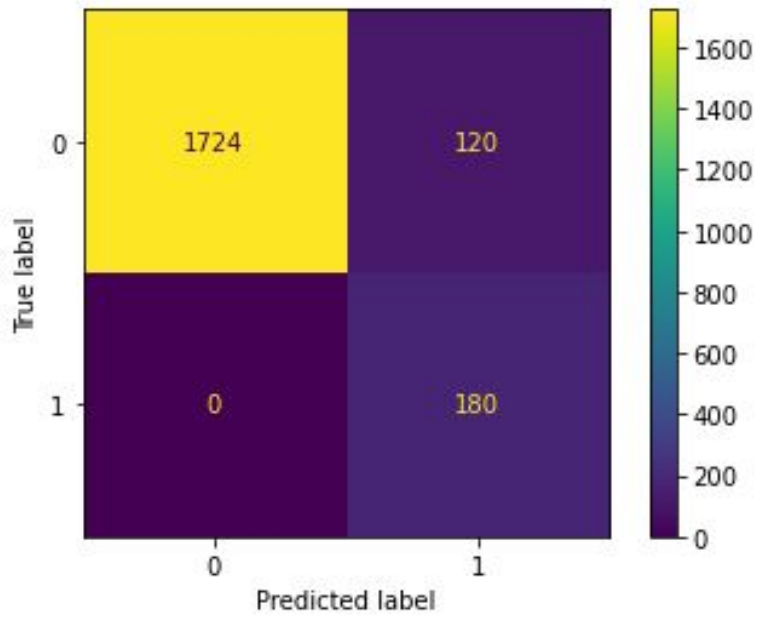

Figura 8: Confusion Matrix

6 Generalization and Conclusion

Comparing the Fuzzy System with the NN, we can see that the NN performs better, however, the results of the Fuzzy System weren't bad, having in consideration that it requires way less computational power to use this system, and it's implementation is way more perceptible to the human brain than understanding the way the NN is behaving.