Assignment 2: The Raven Test

In Search for Group Differences

Data Mining, University of Aveiro
2019

Filipe Pires, 85122
DETI, MSc. Informatics Engineering

João Alegria, 85048
DETI, MSc. Informatics Engineering

***Abstract*—The search for trustworthy methodologies of determining a measurable intelligence index has been a quest of our brightest minds for decades. Raven matrices tests have proved to have widespread practical use as a measure of intelligence. They are a source of data for many studies on the general population as they seem promising tools for contexts such as psychometric tests or clinical assessment.**

In one study on the application of these matrices to groups of students from different backgrounds, questions emerged regarding the possibility of clear differences between Multimedia and Informatics students. In this paper we present the statistical analysis applied to the tests results with the help of ML classification techniques in search for determining whether any of the two groups showed significant advantages over the other.

***Index Terms*—Raven Matrices, Psychometric Tests, Intelligence Measure, Support Vector Machine, Multi-Layer Perceptron, Decision Tree, K-Nearest-Neighbors**

I. Introduction

Raven's Advanced Progressive Matrices (RAPM) is a non-verbal group test typically present in educational or clinical settings, as it is used in measuring abstract reasoning and regarded as a non-verbal estimate of fluid intelligence [1]. Examples of related test are Naglieri Nonverbal Ability or Spacial Ability Tests. Their practical use is very extended, and applicable to both adults and children. Nevertheless, studies that resort to them usually focus on populations containing groups with specific differences in order to draw conclusions from these differences. Examples of these studies are on different military sections, or different mental disabilities.

In the study whose collected data was used for our analysis [2], the aim was to compare university students from different fields in terms of learning styles effectiveness. Several tests such as Kolb and VAK or Hermann dominances allow to distinguish some learning styles like: Accommodator, Assimilator, Auditory, Convergent, Divergent, Kinesthetic and Visual. But beyond this, the researchers also applied the RAPM tests to reach more robust conclusions, and combine all results in a meaningful way. In this paper we focus only on the data related to the second set of tests.

The population that conducted the Raven tests was a group of 45 university students, 21 of Design and Multimedia and 24 of Informatics Engineering. 48 problems were presented to the distinct populations and were divided into two phases: during the first 12, the participant would receive a feedback about his/her answer; for the remaining 36 no feedback was given. During the test execution electroencephalographic (EEG) signals were registered while the participants performed the tasks, using Enobio 8 EEG recording headset and 8 channels: *F3, F4, T7, C3, Cz, C4, T8* and *Pz*.

Our aim was to determine, solely from this estimated measurement of intelligence, whether both groups hold characteristics significantly different from each other by building classification algorithms that interpret the EEG signals and other time-related metrics as features and attempt to predict which class of students a new entry belongs to. We also intended to compare our conclusions with those obtained by the original researchers.

II. Dataset & Feature Extraction

The dataset provided for the completion of this assignment had data both related to individual participants and aggregations with averages. For our intentions, we were interested only in the individual results, so we used the data from the "Trial" folder, containing 4 XML spreadsheets: two for the Informatics students (referred to as DEI), two for the Multimedia students (referred to as ESEC). Departments were split as one file contains the right answers of the students on the Raven tests and the other contains the wrong answers.

Each file contains information about 36 task executions, with an entry for each student and a column for each metric. The relevant marks for the signal analysis are around:

- Problem Display
- Possible Solutions Display
- Student's Answer

On the problem and possible solutions displays, the considered signal processing windows were [-75 500] ms. On the student's answer, the window was [-500 500] ms.

Some of the considered metrics were:

- Peak & Latency for P100 PZ, P100 CZ, P300 CZ signals
- F3, F4 Stress signals

P100 and P300 were chosen because of attentional and relationship characteristics that have sometimes been attributed to them. They are defined in the first time interval and their latency and amplitudes are stored, considering only the relevant channels. This data was used as time features for the classification algorithms.

For the frequency features, the energy of the characteristic bands is estimated in all defined windows. This Engergy (E) is then used to compute other higher level features called energy ratios, considering several signal channels:

- Stress Index
- Mental Fatigue
- Alpha Lateralization
- Immersion Index

All of these features are present in the dataset spreadsheets.

As in any classification challenge, we needed to divide the data into training and testing entries. The way we divided the results was 80% for the training of the classifiers and 20% for testing, due to the relatively reduced size of the dataset.

III. Data Quality & Normalization

The first thing we did with the raw data was to correlate every possible pair of features through 2-Dimensional projections. This would help us understand how was the data distributed in the multidimensional space and whether there were clear separations between the two classes (DEI and ESEC). Table 1 is the visual representation of this correlation as an organized collection of:

- Histograms - presented in the diagonal composed by the feature related with itself and describing the value distribution inside that given feature.
- Scatter Plots - presented in all other entries and describing the value distribution between feature pairs.

Through the analysis of this table, we can gather some important conclusions. The most clear characteristic detected was the presence of outliers, one that was already expected and needed to be taken care of. The second was the fact that the data in all pairs of features doesn't appear to be separable, i.e. there should be something resembling two groups, one for each class, or point clouds in some of the projections. If this was visible, the possibility of actually existing a difference between groups would be likelier to be true. However, that does not seem to be the case, since in most cases we are presented with uniform distributions, gaussian distributions or simply the grouping of all data in a unique central point. It is important to mention that this does not automatically mean that a solution could not be found, as the group distinction we were looking for could depend on more than two variables and so could indeed not be visible in the 2-Dimensional projections.

Once familiarized with the data, we proceeded to preprocessing it so that the models we intended to to apply would have the best chances of finding the distinguishing factors. To do this, we centered the data and removed the detected outliers. Equation 1 contains the formula of the *Z-Score* algorithm applied to each feature to center all values. Then, we removed any data entry that contained a *Null* value, since that would cause problems for the models' processing. Finally we removed the still existing outliers with the help of equation 2, that ensures that entries with very distant values from the standard deviation are not considered as they do not have valuable information to offer.

$$z_i = \frac{x_i - \mu}{\delta} \qquad (1) \qquad\qquad |x_i - \mu| < 3 \times \delta \qquad (2)$$

IV. Classifiers

In this section we will present and give a brief summary of all the machine learning models used in this study.

A. Support Vector Machine

Commonly denoted as SVM, this supervised machine learning model with the supplied training set, by mapping the several data entries in the a multidimensional space in a way that the the widest possible gap exist between the data of each class. Based in that gap a model is created and a decision rule defined that when feed new data tries to assign a class to that entry by using the decision rule already mentioned.

B. MultiLayer Perceptron

This class of feedforward artificial neural network is another supervised machine learning model. MLP is sometimes strictly refers to networks composed of multiple layers of perceptrons. Perceptron is also a supervised learning algorithm capable of doing binary classifications according to a linear predictor function. The power of the MLP is the junction of several of those perceptrons that individually don't have the a good performance, but the combine performance can reach significant values.

This junction normally follows a standard architecture divided in layers, firstly a input layer, followed by one or more hidden layers and an output layer. The input and the output layers are the ones that interact with the exterior, being the first, as the name indicates teh one where the data should be injected at and the second where the result will be presented. The hidden layers is where the main learning occurs and where the most amount of fine-tunning is applied. In this entire architecture the number of layers and the number of neurons(singular cell inside a layer) can be tunning to improve the model performance.

C. Decision Tree

As the name suggests, this model uses a tree like structure to support the data classification decision. The construction of this auxiliary structure is quite simple and straightforward, consisting in assigning a condition to each new branch, causing that the structure will converge with having only one class in leaf nodes, defining the conditions required to identify the existing classes.

D. Random Forest

E. K Nearest Neighbors

V. Models Fine Tuning

A. Performance Evaluation

Once the classification models were implemented, we needed to measure their performance. This was accomplished through the confusion matrix, a matrix that correlates a model's predictions with the true values and gives us the number of: true positives (TP) and true negatives (TN), the correct predictions of wether an entry belongs to 1 of the classes or not; false positives (FP) and false negatives (FN), the incorrect predictions. The class defined as true can be any of the two, as the process is equivalent when inversed. In our case this was something done internally to the tools used, beyond our reach.

With these values, we were able to compute 4 widely used performance metrics:

- Accuracy, given by $A = \frac{TP+TN}{TP+TN+FP+FN}$
- Precision, given by $P = \frac{TP}{TP+FP}$
- Recall, given by $R = \frac{TP}{TP+FN}$
- F1-Score, given by $F = \frac{2\times P\times R}{P+R}$

During our analysis we only consider *A* and *F*, as accuracy alone is not enough to evaluate the models (not robust to several aspects) and F1 correlates precision and recall very effectively.

B. Parameters Variation

In this section we present the manipulation of the hyper-parameters related to each of the implemented algorithms, in search for better performances.

For the SVM, using the default parameters with the Radial Basis Function (RBF) kernel, the accuracy was around 55%. The sigmoid kernel proved to be more appropriate for the dataset as it increased both the accuracy to 57% and the F1-Score from 68% to about 73%. The remaining parameters such as the kernel coefficient (altered between ”auto” and ”scale”), tolerance for stopping criterion (ranged from 1.0×10^{-3} to 0.1), decision function shape or maximum number of iterations proved to have no significant effect when carefully altered.

Moving on to MLP, the default parameters of the neural network were the following: ”relu” activation layer, ”adam” solver for weight optimization, L2 penalty and tolerance both at 1.0×10^{-4}, ”constant” learning rate and an architecture with 3 hidden layers of 13 nodes. This configuration resulted in an accuracy of 54%. The best found configuration turned out to be the following: ”identity” activation layer (although not very different from the original), ”lbfgs” solver (that proved to be more effective than ”adam” for our dataset), a simples architecture with only 2 hidden layers of 10 and 5 nodes (that had the same effect as the more complex one), and the remaining parameters with the default values (as changing them only reduced at least one of the performance metrics). The final performance was of 56% of accuracy and 67% of F1-Score.

The DTs showed worse results in all metrics. Changing the criterion (function to measure the quality of a split) from ”gini” to ”entropy” showed no improvements, nor did tweaking the trees' maximum depth or minimum number of samples required to split an internal node. Defining the weights associated with classes made precision fall 2 percentiles. The best accuracy found was 53% with F1 at 58%. The addition of Random Forests was an attempt to improve these values, as RFs use several DTs to return a combined answer theoretically better than one considering only one DT. However, from our studies, Random Forests were only able to improve F1-Score, and only slightly with a maximum value of 60%.

Finally we dedicated our attention to KNN. Reducing the number of K nearest neighbors from 5 to 3 seemed to have a considerably good effect on its performance, raising 2% of the model's accuracy. Also, defining the algorithm used for computing the nearest neighbors was futile, as all alternatives had no effect whatsoever. Altering the distance calculation formula from euclidean to any other reduced the precision slightly. The best configuration gave an accuracy of 53%, with F1 at 60%.

C. Feature Manipulation

Lorem ipsum ...

VI. Results Discussion

Lorem ipsum ...

VII. Conclusions & Future Work

Lorem ipsum ...

References

[1] Warren B. Bilker et al., ”Development of Abbreviated Nine-item Forms of the Raven's Standard Progressive Matrices Test”, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4410094, accessed in December 2019.

[2] Felisa M. Córdova et al., ”Identifying Problem Solving Strategies for Learning Styles in Engineering Students Subjected to Intelligence Test and EEG Monitoring”, https://www.sciencedirect.com/science/article/pii/S1877050915014787, Procedia Computer Science 55 (2015), accessed in December 2019.

[3] A. Tomé, ”Data Mining Assignment”, https://elearning.ua.pt/pluginfile.php/1496406/mod_resource/content/3/ED_HCT_Raven.pdf, accessed in December 2019.

APPENDIX

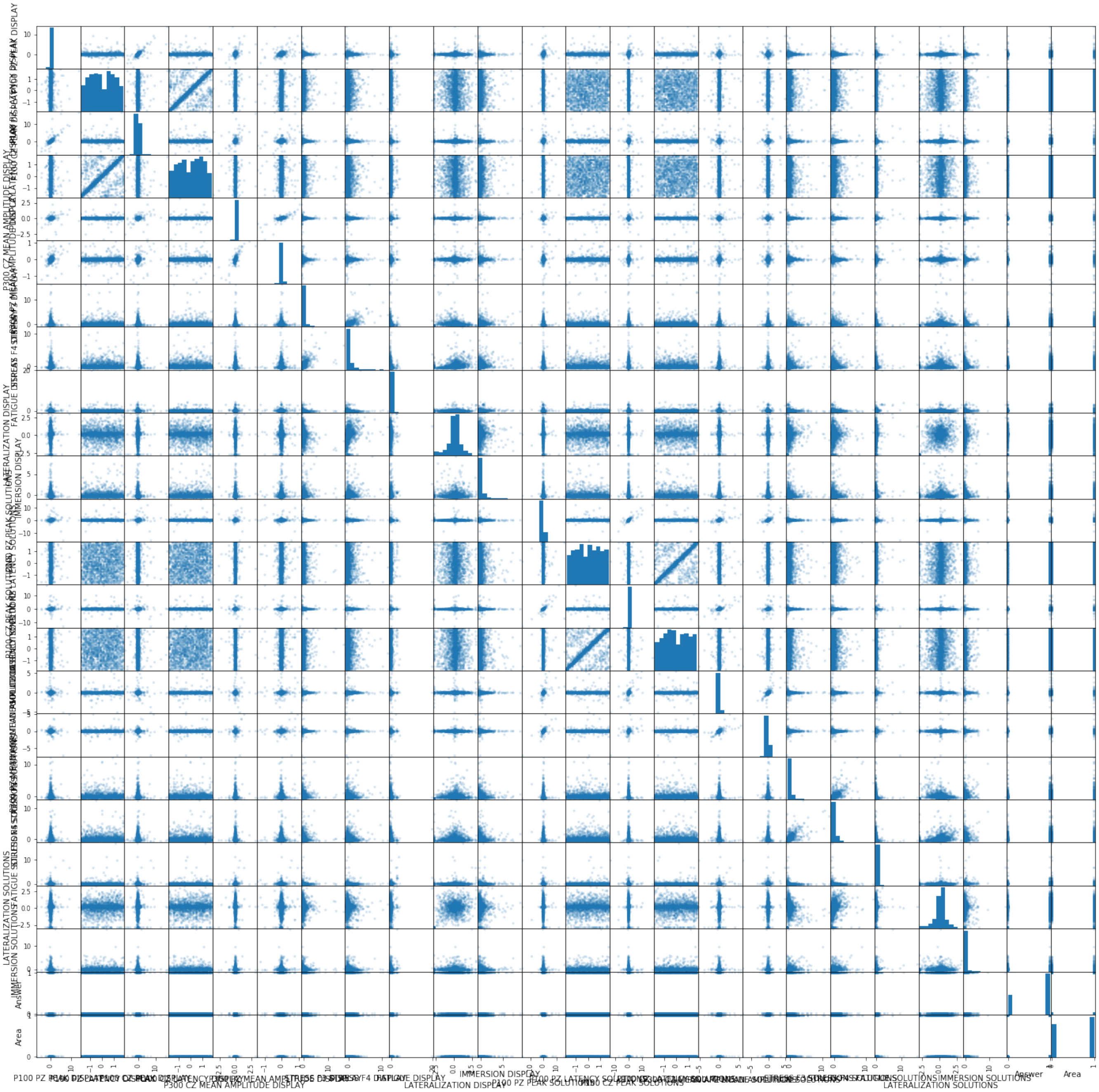

Fig. 1. Feature Correlations