

**TSM – Machine Learning**

**PW 09 : Decision Trees**

16 novembre 2023

Daniel Ribeiro Cabral – Bastien Veuthey

*MSE | Data Science and Computer Science*

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

## Classification Trees

### Q1.1: At which frequencies does the electrical activity mostly occur?

The electrical activity, as represented by the amplitude of the spectrum, predominantly occurs at the lower frequencies has we can see in the graph. Specifically, we can observe significant activity in the range up to around 2 to 10 Hz. Each state shows a concentration of amplitude within specific frequency bands. For instance, the non-REM sleep (n) state appears to have activity at the very low-frequency range, while REM sleep (r) and awake (w) states show activity extending into slightly higher frequencies.

### Q1.2: Can you easily distinguish between classes visually? What can you say about the inter- vs intra-class variability?

While it can be a bit difficult to visually differentiate between the states based on the EEG spectra, certain distinctions are noticeable:

* **The non-REM sleep (n)** spectra exhibit a uniform pattern, with a pointed peak at lower frequencies, implying a low variability within this state (low intra-class variability). This consistency in the non-REM sleep state's spectrum could serve as a visual cue for identification.
* **The awake (w)** state also demonstrates a consistent spectral pattern, similar to the non-REM sleep, marked by low variability within this state (low intra-class variability). However, discerning the awake state from the non-REM sleep solely on visual inspection is complex (high inter-class variability), as both states lack significant peaks and maintain a lower amplitude throughout the spectrum.
* In contrast, **the REM sleep (r)** state shows a much higher degree of variation in its spectral profile, signifying greater variability among different instances of REM sleep (high intra-class variability). Visually, this state is more distinct compared to the other two, with notable and more prominent peaks in the spectrum (clear inter-class differentiation)

### Q1.3: Describe both of these criteria. What does a gini impurity of 0 means? What does an entropy of 1 mean?

Has seen in the theory class:

* **Gini impurity:** It's a measure that expresses the likelihood of incorrect classification if you randomly pick an item and classify it according to the distribution of classes in the dataset. It ranges from 0 to 0.5 (where 0.5 denotes a 50/50 split in a binary classification), with 0 being the most desirable score. A Gini impurity of 0 means that the set is perfectly homogeneous, that is, all elements belong to the same class.
* **Entropy:** Almost the same concept has gini. It is a concept that measures the level of disorder or uncertainty in the data. It ranges from 0 to 1 in the case of binary classification (the range is 0 to log2(n) for a classification problem with n classes). An entropy of 0 means that there is no surprise in the data (perfect purity), while an entropy of 1 (in binary classification) indicates that the data is evenly split between the classes, representing the maximum level of disorder or uncertainty. An entropy of 1 would signify that the data within a node is perfectly split between the classes, meaning the information provided by the node is no better than a random guess.

### Une image contenant capture d’écran Description générée automatiquementQ1.4: This problem suffers from a common issue in machine learning. What is this problem called? What could be its causes? How can it be resolved?

The problem found in this system is the **overfitting**. Overfitting occurs when a model learns the training data too well, capturing noise and details that do not generalize to new, unseen data. Generally, the system arrives to a 100% of accuracy on the train set. Then in the test set the accuracy drops significantly.

Some causes of this problem:

* Training the model for too many iterations
* Having a model that is too complex relative to the simplicity of the data we have.
* Having a small data

Solving this problem:

* Collecting more data or augmenting the dataset artificially (data augmentation)
* Using techniques like cross-validation
* Implementing early stopping during training, so the model doesn’t continue to learn the noise in the training data.

### Q1.5: Use the visualization of this tree to show and explain:

**What is a node? What is an edge? What is a leaf?**

* **Node**: Is a point in the decision tree where the data is split. In this tree, each rectangle represents a node. Each node will test an attribute and split the data according to the outcome of the test.
* **Edge**: An edge is the line that connects one node to another.
* **Leaf:** A leaf is a terminal node that does not split any further. This is where a prediction is made.

**What are the two additional hyperparameters doing? Do you think that both are necessary in this particular case (min\_samples\_leaf = 20, max\_depth= 4)? Why?**

* **min\_samples\_leaf :** This parameter specifies the minimum number of samples required to be at a leaf node. For example, if a leaf results in a result lower than the number specified it will not create it. That’s a good way to avoid overfitting.
* **max\_depth :** This hyperparameter controls the maximum depth of the tree. The depth of a tree is the length of the longest path from the root to a leaf.

In our case, the parameters are controlling the complexity of the tree. The example we had in the Q1.4 doesn’t have any limit that will avoid overfitting. So in our, case the parameters will do it and maximize our chances of doing good predictions on a new input. Both can be necessary, but their necessity values would usually be determined through a process of experimentation and cross-validation to see how changes in these parameters affect model performance.

**What does the color of each node represent?**

The color in each node represents the majority class in that node after the split, with different colors corresponding to different classes. In this case, they play with the intensity of the color to say if there are more or less portions of the class.

### Q1.6: Choose one of the nodes. Explain precisely the information given on each line of text in this node.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquementThe first value is the condition to split the data. If x[13] is lower then 0.07. The testing value will go left, if not it goes right. The second line explains the gini that we had (Q1.3). The samples line equal to the total number of samples that were throw this node. The value represents the distribution of the classes within the samples at this node (class 0, 1 and 2). The class line indicates the majority class among this node.

### Q1.7: Does model 2 still have the same problem as model 1? Explain based on the classification reports and the confusion matrices

### Q1.8: One of the class seems more difficult to predict than others? Which one? Where could this difficulty come from in your opinion?

### Q1.9: What does this hyperparameter do? Explain giving examples from this dataset.

### Q1.10: Compare results from model 2 and model 3. What are the pros and cons of each of them?

## Random Forest

### Q2.1: For each of the hyperparameter: Is there a range of value giving particularly good results? Or particularly bad results?

### Q2.2: These representations give valuable information about hyperparameters. It is nevertheless insufficient. What are/is the main problem(s) with those graphs in your opinion?

### Q2.3: What do the following plots represent?

### Q2.4: What do the white spots (=empty spots) in the heatmaps mean?

### Q2.5: How do those plots address the limitations of the previous visualizations?

### Q2.6: What is grid search? Explain by giving real examples from this specific task.

### Q2.7: Use the plots above to narrow the range of hyperparameters you want to explore. Which values did you choose to test for each parameter? Justify your choices.

### - Exercise 2.1: Once you know which values of hyperparameters you want to explore, complete the following code to perform a grid search on those values. Remember that the more values you choose to test, the longer the computational time required.

### - Exercise 2.2: Complete the code in order to choose one final value for each of the hyperparameters and train the model.

### Q2.8: Which value did you choose for each hyperparameter? Why?

### Q2.10: The test set should be used only at this stage, and it is theoretically important not to change the hyperparameters based on the performance on the test set. Why?

### Q2.11: Comment your results. -> How well does the model generalize on unseen data? Is a random forest better than a single classification tree in this case? What is the main challenge of this dataset? …

### Q2.11: How is this importance calculate?

### Q2.12: What can you conclude from this graph?

## Gradient Boosting for classification

### Q3.1: Two additional hyperparameters were added compared to the RandomForestClassifier. What are these hyperparameters, and what roles do they play?

### Q3.2: Comment the results. Compare these results with the ones obtained with the RandomForestClassifier. Compare more specifically the precision, the recall and the f1-score of the 'r' class obtained with GradientBoostingClassifier and RandomForestClassifier. What are your conclusions?