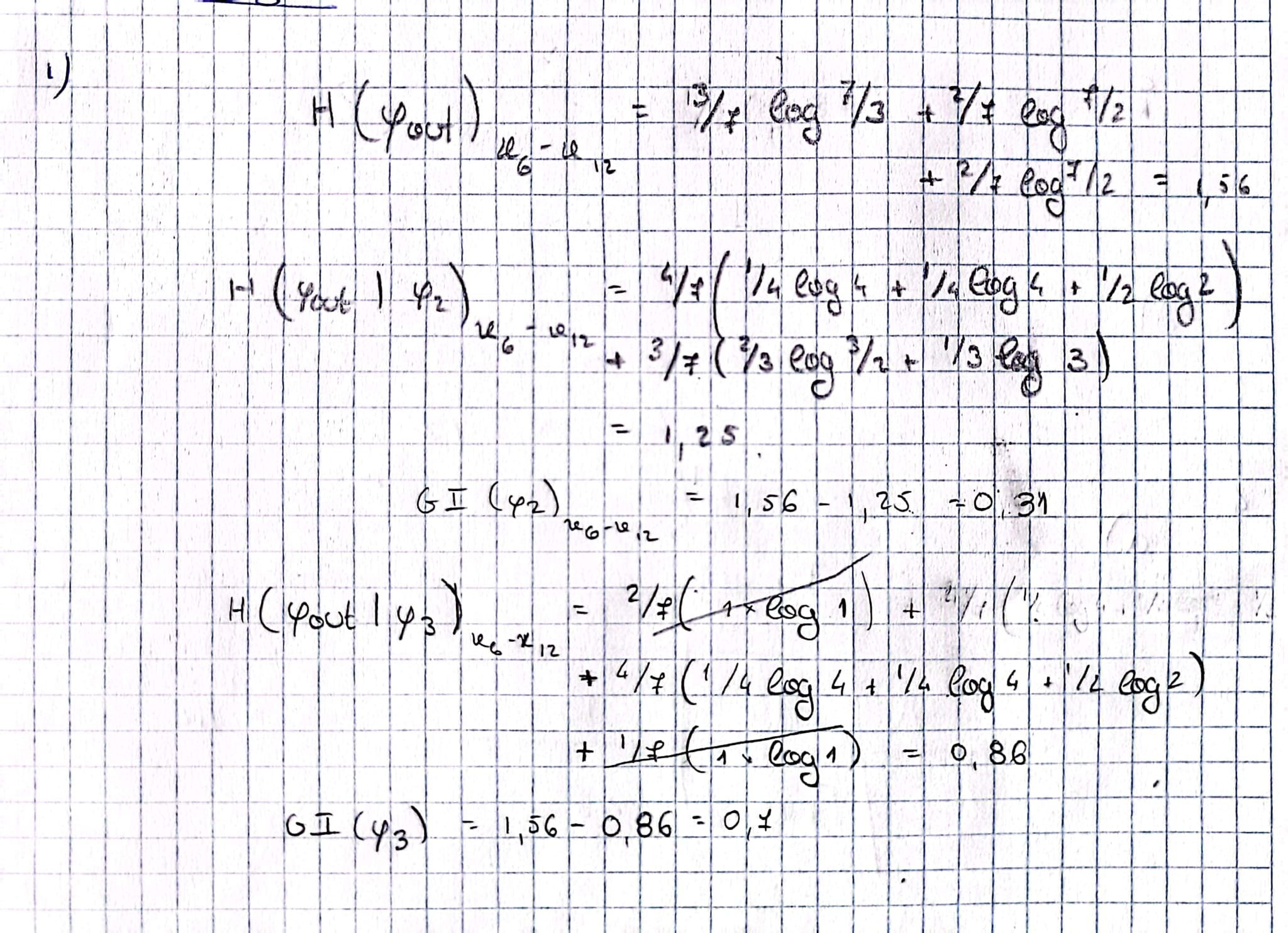
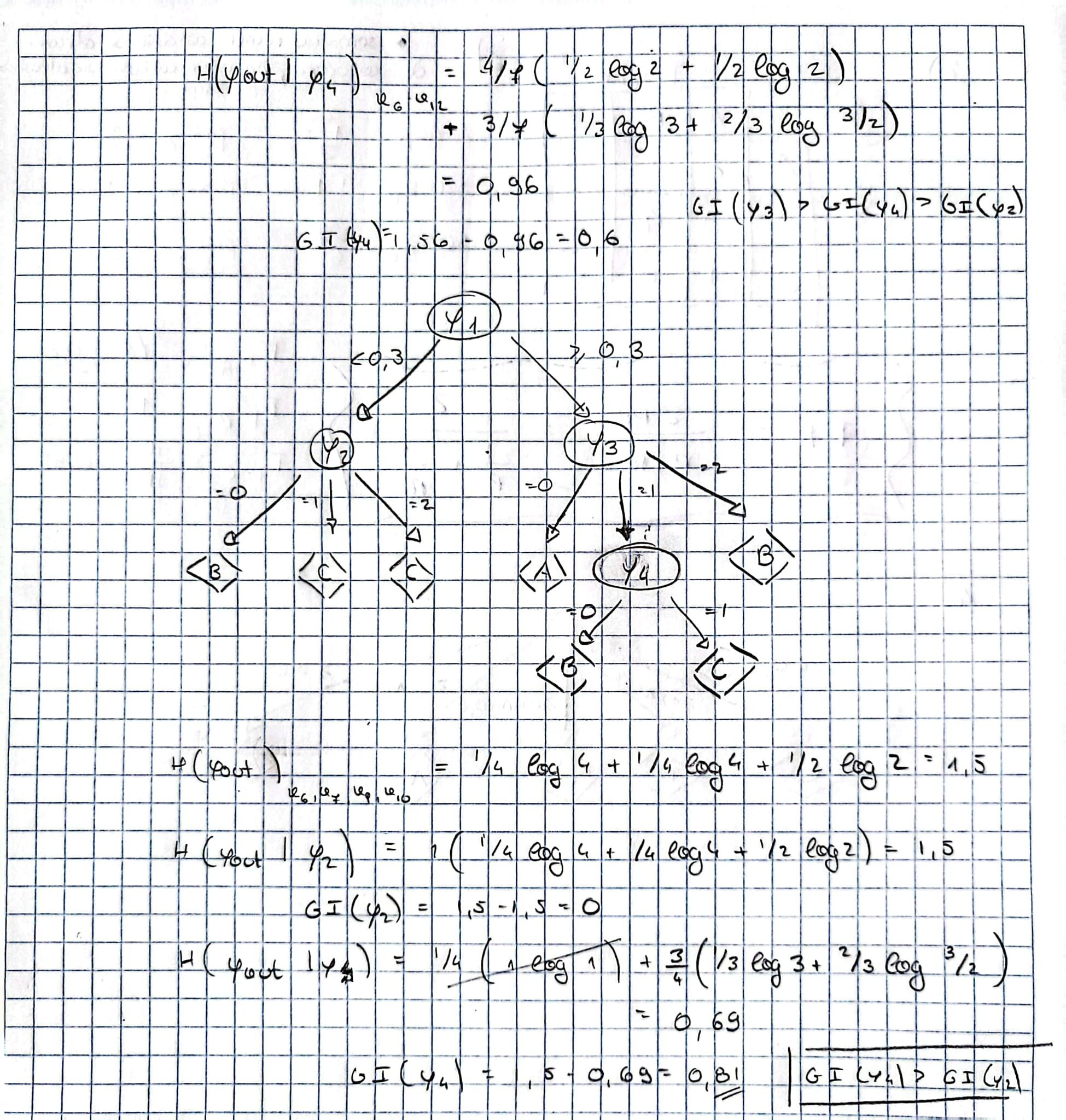
**I. Pen-and-paper**

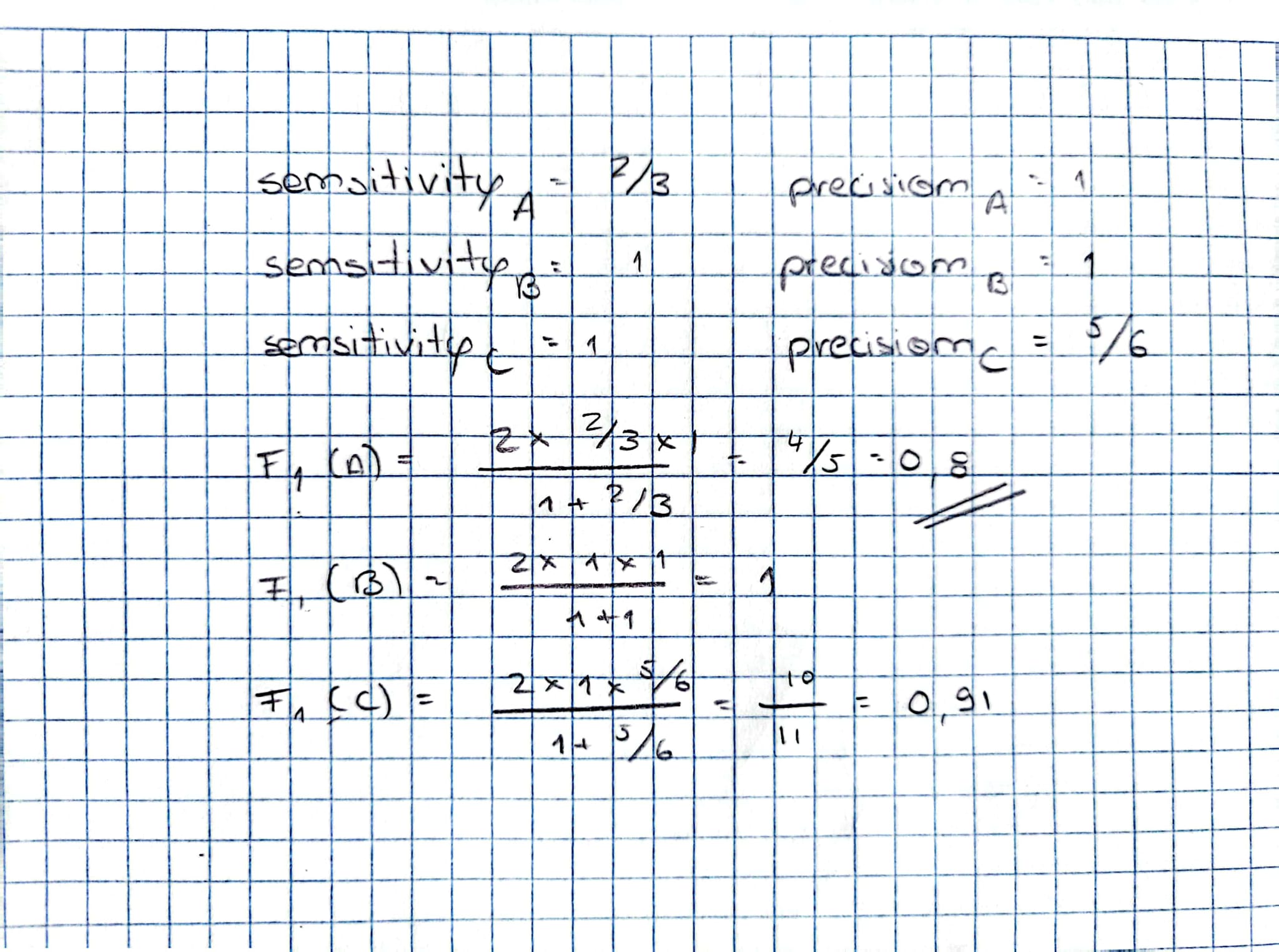




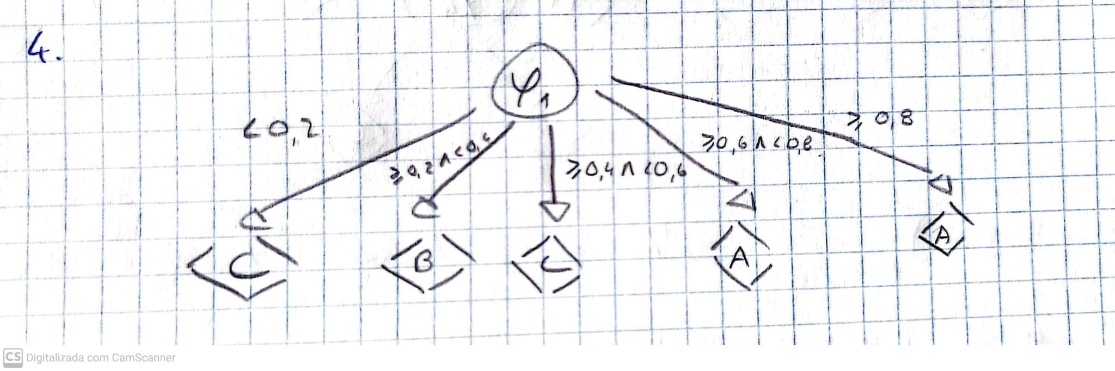
**Resposta:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Predictions | Truth | | | |
|  |  | A | B | C |
| A | 2 | 0 | 0 |
| B | 0 | 4 | 0 |
| C | 1 | 0 | 5 |

1. 

**Resposta:** Class A



**II. Programming and critical analysis**

1. Answer 5
2. Answer 6
3. For better analysis, we can separate the results in two different ranges in the X axis (Min Sample Split).

* Min Sample Split < ≈30

The train accuracy starts at 100% for a Min Sample Split of 2. That makes perfect sense as it means that we always split the decision tree whenever we get two different outcomes for differencing attribute values. This will inevitably result in a perfect classifying decision tree for every instance in the training data set, explaining the perfect accuracy score. However, it scores lower than 70% accuracy in unseen data (test data).

As the Min Sample Split setting increases along this range, the train accuracy decreases and the test accuracy increases due to a better generalization of the results.

We can come to the conclusion that the decision trees created from smaller Min Sample Split values suffer from overfitting. They become too complex, overly learning patterns and noise in the training data and then scoring poorly in the test data.

* Min Sample Split > ≈30

As the setting for Min Sample Split values increase from 30, the test accuracy starts slowly decreasing along with the train accuracy. By this point the decision trees created are starting to become too simple, translating in worse accuracies for both data sets. They now indicate underfitting problems.

The best balance between complexity and generalization capacity is found setting the Min Sample Split value at around 30, where the maximum test accuracy of about 77% is reached. At this point, the model catches enough patterns in the data to perform well with unseen data, without overly fitting the training set.

1. Answer 8

**Note**: do not forget to also submit your Jupyter notebook

**END**