NOTE: I am completing this assignment using Python, so I will try to recreate the results in SAS.

1. **Deliverable 1:** **Show at least part of the table that shows the classification rules for Serious = Y or N using the text variable of the doctor's notes. Complete the node tree shown on p 46 of the TM Tutorial Chapters 5, 6, and 7**

In order to build classification rules, I am following the tree procedure outlined by Weiss et all. That is, I am going complete a decision tree first, and extract classification rules from the tree. Using Python, I created a decision tree with the following criteria:

* Minimum samples per leaf: 200
* Maximum depth: 5

This results in the following tree. Note, the initial population has a distribution of 93.3% serious, and 6.7% non-serious. By the time we get to the leaves, the distribution is much better split into mostly serious, mostly non-serious, with one leaf that is closer to an even split.

A close up of text on a white background

Description automatically generated

Classification Rules:

**Serious**

* hospitalization > 0.012
* (hospitalization <= 0.012) & (disabling > 0.021)
* (hospitalization <= 0.012) & (disabling <= 0.021) & (admitted > 0.038)

**Non-serious:**

* (hospitalization <= 0.012) & (disabling <= 0.021) & (admitted <= 0.038)

1. **Deliverable 2:** **Use the ClassifyConfusionMatrix.xlsx spreadsheet to calculate the classification measures and report the results. In the text output SAS will provide the count of TP, TN, FP, and FN predictions. Report at least accuracy, precision, and F1-value, an overall measure of how good the classification model works on the test partition.**

Using the tree model, we can build our crosstabulation. Below, prediction is the rows, and actual is the columns. True indicates Serious, and False indicates non-serious.  
A screenshot of a cell phone

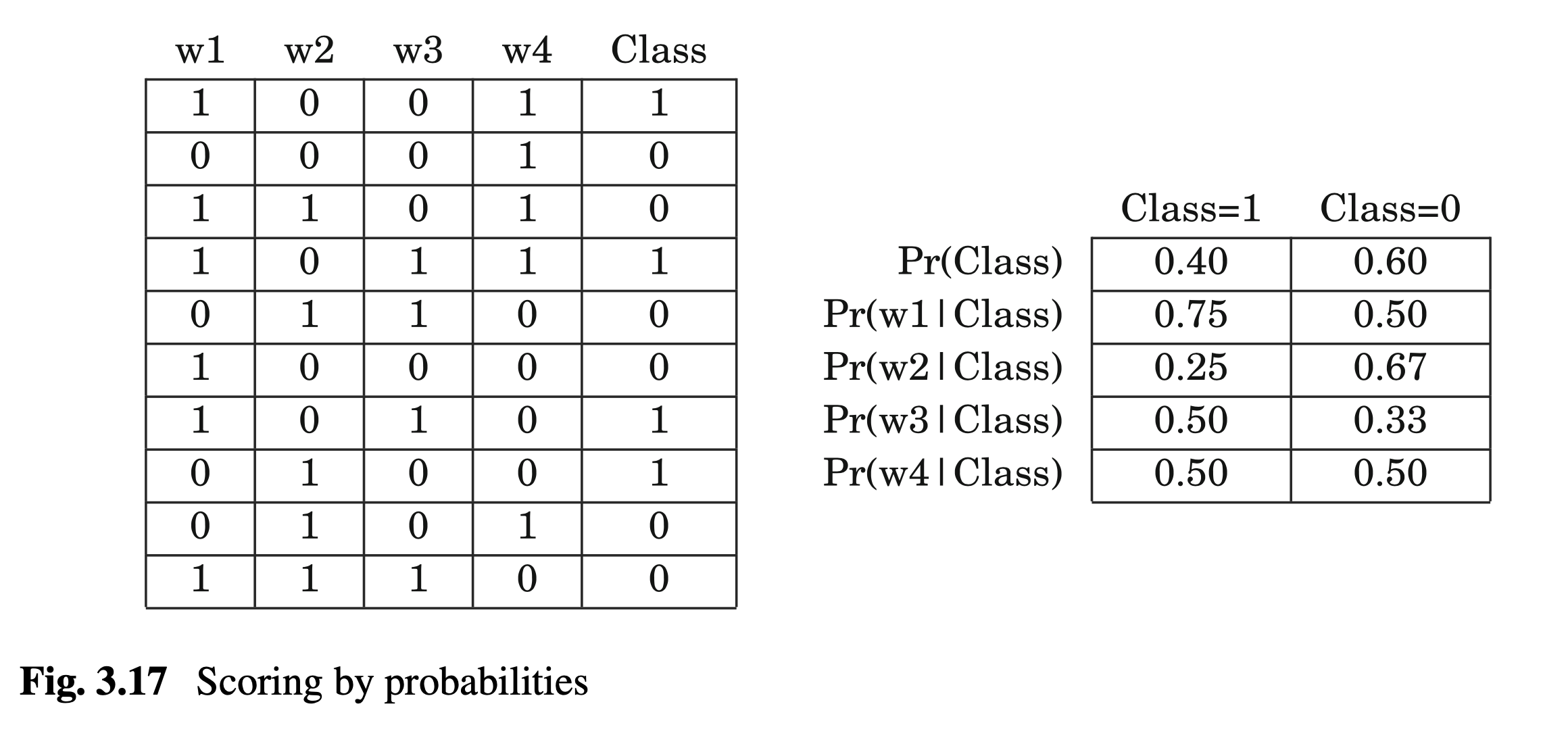
Description automatically generated

We can immediately note that we don’t have a lot of false positives, but we do have a number of false negatives. In this case, I think the false negatives are a lot more serious, as someone may not get the care that they need. In the attached ClassifyConfusionMatrix.xlsx, I’ll represent this by having false negatives have a greater cost. As shown in the attached spreadsheet, here are my calculated metrics:

* Accuracy:
* Precision: 89.5%
* F1-Score: 0.5760
* Overall Measure: Kapa = 0.95

Kapa, in this scenario, tells us that our classifier is very good. Kapa is a measurement of the agreement of our actual data and predicted data, vs random agreement. In this case, 0.95 indicates we have an extremely good classifier.

1. **Deliverable 3:** **Verify the probabilities in Fig.3.17 and compute the class for a document that has words w1 and w2.**



Given that 0.067 > 0.018, we can label this new document as Class = 0.

1. **Deliverable 4:** **Submit a brief report indicating what you did and what you found. Indicate how the results are related to the theory in the Weiss et al Report the measures that show how well the classification worked.**

For this assignment, I used a decision tree to build a set of classification rules. Using these rules, we can make predictions about new cases, whether or not they are serious. These predictions are made on the tokenized ‘Symptom Text’, with a few added columns (age, sex, cur\_ill, prior\_vax). Not terribly surprisingly, the text itself proved to be much better at making serious classifications than the four additional columns. The classifier appeared to do a good job breaking out the serious and non-serious cases, but the high number of false negatives would be a little concerning in this specific case. Not properly treating someone with serious complications could mean death, disablement, or other serious health detriments. In this case, while the Kapa, precision, and accuracy were all *good*, I think in a health scenario, we would need to have them be better. Adjusting the costs in the attached spreadsheet to punish false negatives, I actually end up with a negative expected payoff. This measure indicates we should try to build a better model.

This is where we can relate to the theory presented in Weiss. If we were looking into creating a better model, we would use a loss function of some kind, and iteratively train models to try and minimize this loss function. This cost function could punish the false negatives greater than the false positives. With this iteration, we would also likely see improvements in accuracy/precision/f1.