**1. Combining, Bagging and Random Forests**

1a. All three doctors correct = 0.8^3 = 0.512

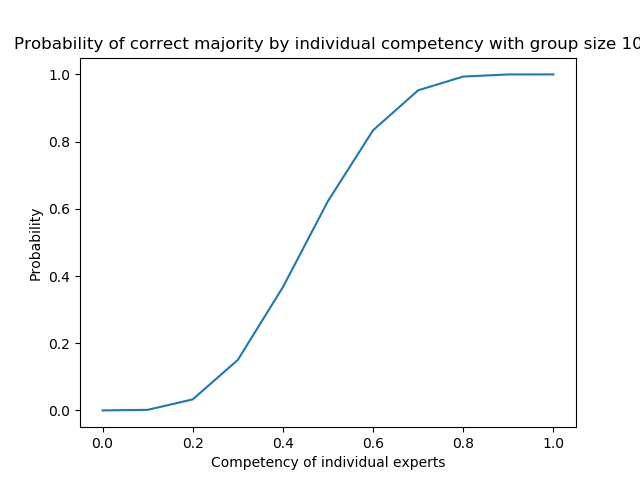
At least two doctors correct = 0.8^2 = 0.64

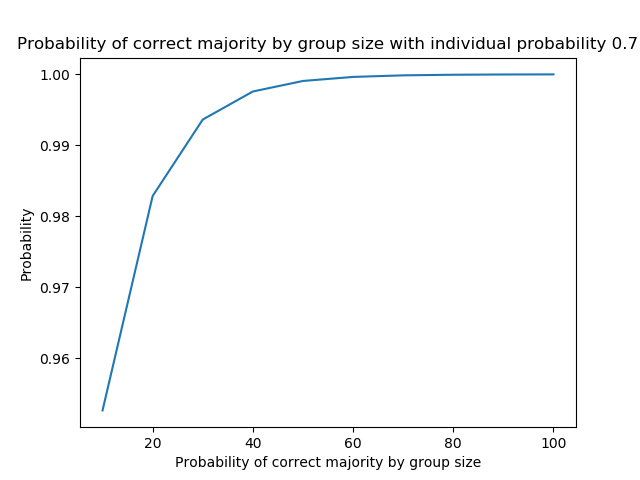
Thus, the probability that the group is correct using majority voting is 3 \* (0.2\*0.8\*0.8) + 0.8^2 = 0.896

1b. We did not find a direct formula, see answer c. for the code implementation.

1c. See the file “majority.py”. The probability that the majority of medical students in example c. are right is roughly 0.78.

1d.





1e. As of now, the three experts have the highest chance. The student group seems to have a higher probability after they gain more than roughly 26 students (see code).

1.2.

Calculated probabilities:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | | P2 | | P3 | | Mean | | Max | | Min | | Prod | |
| A | B | A | B | A | B | A | B | A | B | A | B | A | B |
| 0.9 | 0.1 | 0.9 | 0.1 | 0 | 1 | 0.6 | 0.4 | 0.9 | 1 | 0 | 0.1 | 0 | 0.01 |
| 0.9 | 0.1 | 0.9 | 0.1 | 0.3 | 0.7 | 0.7 | 0.3 | 0.9 | 0.7 | 0.3 | 0.1 | 0.243 | 0.007 |
| 0.9 | 0.1 | 0.2 | 0.8 | 0.1 | 0.9 | 0.4 | 0.6 | 0.9 | 0.9 | 0.1 | 0.1 | 0.018 | 0.072 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Decisions for each classifier:

|  |  |  |  |
| --- | --- | --- | --- |
| Mean | Max | Min | Prod |
| A | A | B | B |
| A | A | A | A |
| B | A & B | A & B | B |
| B | B | B | B |

1.3. If *N* = the original sample size, *k* is the bootstrap sample size and *t* is the amount of bootstrap samples taken, then the percentage of observations that are left out of all bootstrap samples = (1 – (*k*/*N*))^*t* \* 100%.

1.4. In bootstrap sampling, random datapoints are taken from the original sample, whereas in the random subspace method, new datapoints are created out of randomly selected features from the original sample over all datapoints.

1.5.

The importance of a feature is derived from the probability of reaching the node that makes a decision about the feature summed over all trees in the ensemble. The probability for a node is derived by counting how many samples reach that node. (https://stackoverflow.com/questions/15810339/how-are-feature-importances-in-randomforestclassifier-determined)

1.6.

To test the random forest classifier, we used the python sci-kit learn implementation on the wine quality dataset. This is a quiet famous dataset from the UCI machine learning repository. The dataset is available for red and white wines, but we only focused at the white wines. The goal for the dataset is to predict the quality of a wine on a discrete scale from 0 to 10. Each integer can be seen as its own class. The dataset has 11 variables (fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol) that can be used to predict/classify the quality of the wine. The accuracy of our classifier is 0.623, using cross-validation (test size = .4). The most important predictors are: free sulfur oxide(0.10), density(0.11) and alcohol(0.11). We compared our results with a SVM. Not only does the SVM have a lower accuracy(0.509), it also takes significantly more time than the random forest.

**2 Boosting:**

2.3.

In bagging one chooses random subsamples form the data to train the individual classifiers on. Boosting is an iterative process. Each individual classifier assigns a weight to data points. Data points that have been classified wrong get a higher weight, so the next classifier focuses more on those difficult data points.