Evan Smith

Assignment 4

**Problem 1:**

Result from my program is: A picture containing text

Description automatically generated

Code is contained in this zip file, written in C#.

**Problem 2:**

I used an excel table to speed up calculations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fruit | Red | Has Leaves | Has Seeds | Total |
| Strawberry | 0.86 | 0.71 | 0.57 | 0.29 |
| Apple | 0.67 | 0.17 | 0.50 | 0.50 |
| Pear | 0.40 | 0.20 | 0.80 | 0.21 |
| Total | 0.67 | 0.33 | 0.58 | 1200.00 |

Full formulas:

Text, letter

Description automatically generated

**Problem 3:**

1. I can say that this classifier is better than random, since the curve is above a 45% line. A point on the curve corresponds to the ratio of true positives to false positives from the model. We can see that if the sensitivity of the model is below 25%, it has a functionally 0% false positive.
2. Of these two ROC curves, model A is better, since it has more AUC. This implies that it has a higher TP rate than the other model at the same FP rate.
3. For a random guess, the ROC curve would be approximately a 45-degree line. It looks this way because for any given point the likelihood of a true positive or true negative is equal, so we expect the points to converge on the diagonal.

**Problem 4:**

I would select the green, Ts2-norm algorithm. It appears to minimize the AUC, which in this case is preferable since we are graphing the FN and TN rates. If we set the threshold for false acceptance at around 25%, then we would expect to have about 12.5% of total matches be imposters, but this algorithm has only about a 5% false negative rate at this point, so only 2.5% of legitimate customers would be turned away. This could be pushed even higher if we so wish, but this strikes a nice balance between 10$ per theft and 100$ lost per false rejection.

**Problem 5:**

|  |  |
| --- | --- |
| RQ | 0.2 |
| RP' | 0.4 |
| RP | 0.6 |
| RL' | 0.9 |
| RS | -0.4 |
| RS' | -0.4 |
| RT | -0.9 |
| RT' | -1.1 |