E. In your PDF, show the resulting one-rule threshold classifier code.
What was your one-rule for this assignment?
Which attribute did it use?
It is just an if-statement.
Copy and paste your code from your output classifier file.
Please use black ink on a white background.
Do not use a screen capture, the fonts do not scale. (2)

```
    # Aggressive drivers are > the threshold
    n_aggressive = np.sum(the_data.iloc[:, THE_IMPORTANT_ATTRIBUTE] >
THE_IMPORTANT_THRESHOLD)
    n_behaving = np.sum(the_data.iloc[:, THE_IMPORTANT_ATTRIBUTE] <=
THE_IMPORTANT_THRESHOLD)

    # Aggressive drivers are <= the threshold
    n_aggressive = np.sum(the_data.iloc[:, THE_IMPORTANT_ATTRIBUTE] <=
THE_IMPORTANT_THRESHOLD)
    n_behaving = np.sum(the_data.iloc[:, THE_IMPORTANT_ATTRIBUTE] >
THE_IMPORTANT_THRESHOLD)
```

F. Run your code to produce your classifier program.
Your code must produce the resulting classifier. (2)
It runs and produces classifier

G. Run your resulting classifier on the supplied test suite.
Your resulting classifier must run. (2)
Runs

H. Report how many Aggressive drivers did your classifier routine find in the test suite?
Have your classifier print out the number of cars <= the selected threshold.
Have your classifier print out the number of cars > the selected threshold.
Report these numbers in your write-up. (2)
n_behaving_well = 87
n_aggressive = 213

I. Conclusion: Write up what you learned here using at least three paragraphs. (2)

What did you discover? Were the results what you expected? What was surprising?

Was there anything particularly challenging? Did anything go wrong?

Once again I mostly learned how to meta program from this which is an interesting tool. I haven't had to do this in any other class or project, and I wonder what applications this is useful for.

I discovered that the noise removal definitely helps the classifier be more accurate, I forgot to do this at first and my results were all out of whack. Mostly I found this task to be similar to the last HW, and I was able to reuse a lot of code.

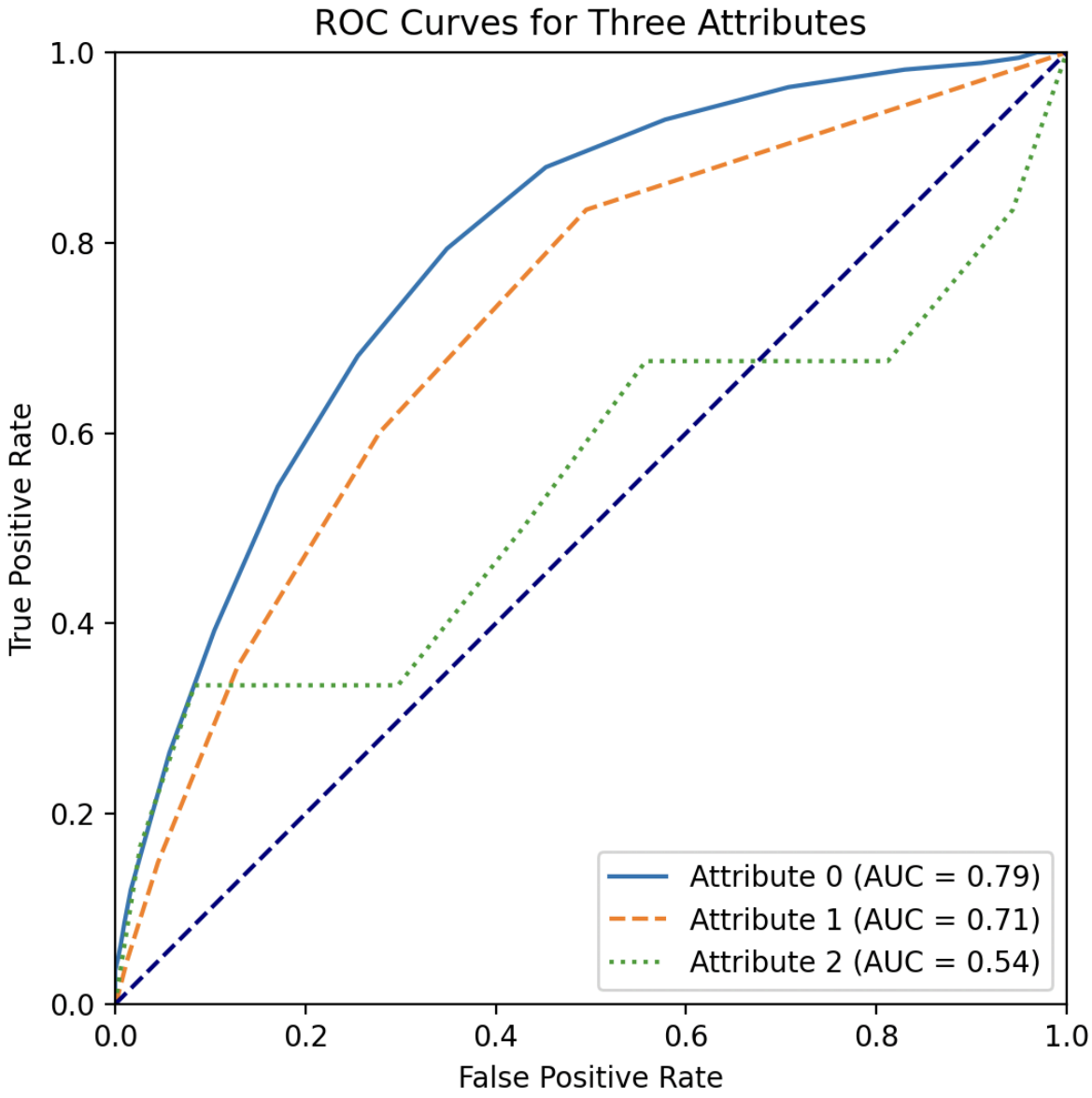
The results were pretty much expected. The optimal threshold of 62 is pretty consistent with the results of previous homeworks and also matches a speed that makes sense to separate aggressive/non-aggressive drivers. The added layer of determining whether aggressive drivers are above or below the threshold made the consideration more robust, and I think overall is accurate.

I was surprised by how sensitive the classifier is to changes in the threshold, so even slight variations in how the data is split, if more aggressive drivers found on one side vs the other, shifted the error rates.

J. Totally Optional Bonus Problem: (+1)
If you are the kind of person who is stressed doing anything unnecessary, do not do this part.
This is just for those who enjoy a challenge, and want to exercise their abilities.

On one graph, plot the three different ROC curves for the three different possible threshold classifiers.
Use different line styles (solid line, dashed line, dotted line) for each attribute.
Use a legend or label so that we can see which attribute is associated with each curve.

Make sure you use square axes, so that it looks correct.
Given these ROC curves, does the resulting attribute selection make sense to you?



Evaluating the ROC curves, attribute 0 has the best performance, attribute 1 is not too far behind but attribute 2 has poor performance. Overall It makes sense that attribute 0 was selected as it performs the best.