Homework 03

Justin Lad

CSCI 720 — Big Data Analytics

February 17, 2019

Problem 1

A.

If we are trying to maximize safety on roads, we should break ties by selecting the lower speed threshold when deciding if the person is driving aggressively. In order to do this, we would use < when scanning potential threshold. This will work because we start with the lowest possible threshold, and keep increasing it to the maximum. So by mandating that the new threshold has to be exclusively less than our current value, our program will select the lower speed threshold in the case of a tie. This ensures maximum safety on the road because we are more likely to pull someone over as an aggressive driver based on the speed.

B.

If we are now trying to maximize trust in police, we should select the higher speed threshold. This will reduce the number of false positives. In order to achieve this, we would use $\leq$ when scanning potential thresholds.

C.

The best threshold to set the police scanner at is 62.5 MPH

D.

If we now change the cost function to be: cost = 2*False Alarm + 1*Missed Speeder, I think the threshold will decrease. I think this because I printed the number of false alarms and missed speeders at the best threshold, and false alarms was the higher variable (11 false alarms vs 104 false negatives). So misses are the dominant factor with this cost function. So if false negatives are twice as costly, the number of false negatives in our best threshold will decrease. And based on this data, reducing the threshold speed will reduce the number of false negatives (misses)

The new threshold is 57.5. This makes sense because now the number of false alarms (116) is much greater than number of false negatives (22)

E.

The temporary cost function can be decomposed into objective function and regularization. Objective function is the # False Alarms. This makes sense because we do not want to pull over innocent drivers and erode trust in the police. The regularization is # Missed Speeders, because it would be nice if we didn't miss many speeders, but isn't crucial to minimize for our objective of keeping trust in the police.

0.1 F.

Number of Aggressive drivers let through (False Negative, aka miss) was 104

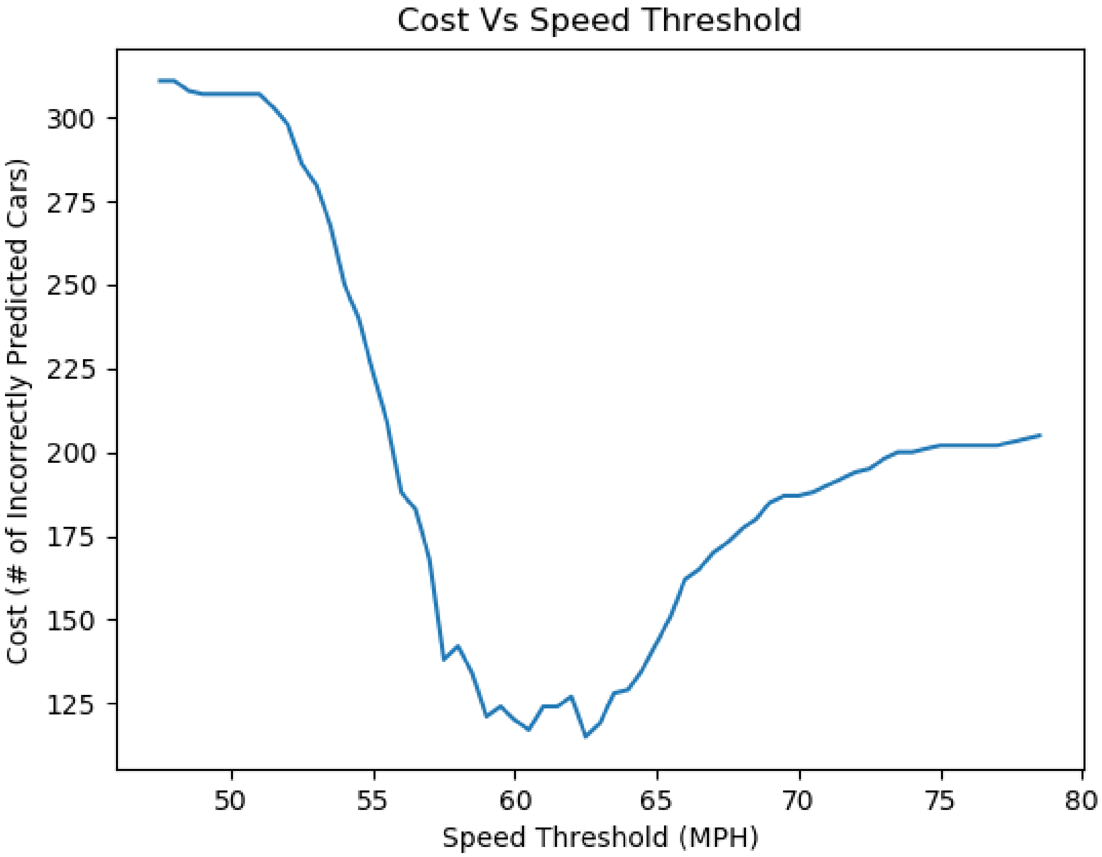
G.

Number of non-reckless drivers pulled over (False Positive, aka false alarm) was 11

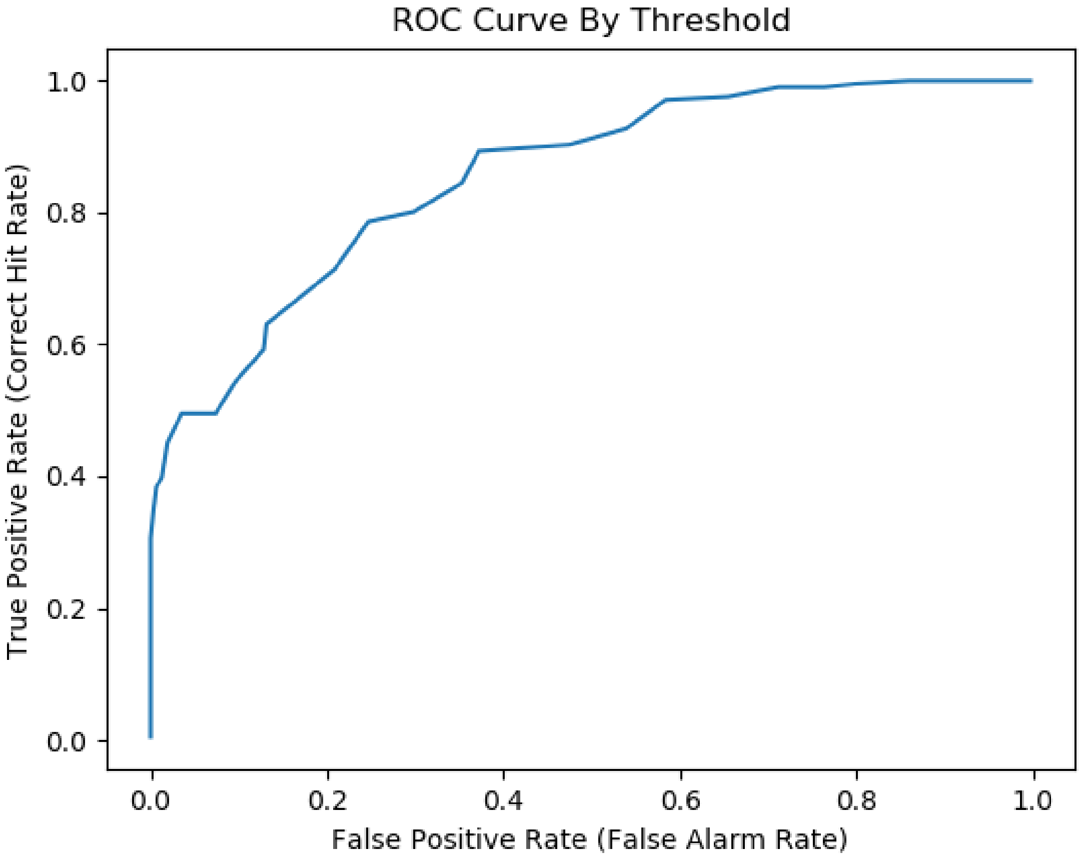
H.

Otsu's method returned a threshold of 61 MPH. It's actually surprisingly close to our classifier's threshold of 62.5 MPH. I found this interesting because Otsu's method is a form of unsupervising learning, that found a similar threshold to this supervised learning model.

I.


Cost Vs Speed Threshold
Cost (# of Incorrectly Predicted Cars)
300
275
250
225
200
175
150
125
50
55
60
65
70
75
80
Speed Threshold (MPH)

J.



K.

I learned about how the cost function can be manipulated to perform better for a given target or task. For example, when we aim to maximize police trust, the main objective is to design a reasonable classifier while also minimizing number of innocent people (false alarm) we pull over. We can also plot the data and see if it is a reasonable classifier. Conversely, if we're in a strict town our objective function should minimize the number of missed speeders. We accomplished this by redefining our cost function as an objective function with cost = 2 * # missed speeders and a regularization function, cost = 1 * # false positives. It was interesting to see how easily we could build a simple classifier based on having access to supervised data and a cost function that minimizes weighted(if desired) sums of misses and false alarms. We then searched the space of possible thresholds from the minimum to maximum value to see which had the lowest cost.

It also taught me how to consider how breaking ties based upon the task or objective we are trying to maximize. When thresholds tie, which should select the tie that benefits the task. For example, when we are trying to maximize trust in police, we should use the lower threshold by requiring an exclusively lower value at higher thresholds. Ties also occurred when deciding if our classifier should label a speed data point as class 0 or not. Again, this depends on our task and if we are trying to maximize trust in police, we should assume class 0, not a speeder. That being said, ties are somewhat infrequent so it usually doesn't have a significant impact on the result. It reduced the threshold by 0.5 MPH in each cost function.

For multi-dimensional data, we would have to sweep possible values of the second (or third, etc) dimensions, to see which had the minimal cost function. This does not scale well, as we would have $O(n^d)$, where d is the number of features. We would then check if our data point was less than or equal to that feature's threshold.

The only tricky thing was sorting the data while preserving the associated column. At first, it was sorting the second column into one's and zero's, so my sort corrupted the data and messed up the data. I knew the correct value based on testing the unsorted data, so I knew I had to find a better way to sort. Turns out, this is what lexsort is for.