Nicholas Mulligan
Comp 131 - Intro to AI
Naive Bayesian Classifier

Compilation

To run the program, make sure that all data is saved in the same directory, then input:

```
>> python3 rtr.py
```

Initial Prior Probability

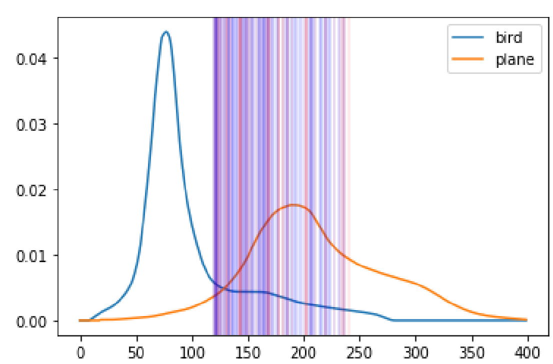
For the initial prior probability, I followed the recommendation in the spec to have a prior of 0.5, or equal probability that a given object is a plane or a bird. Of course this changed while processing the data, but this was the initial assumption.

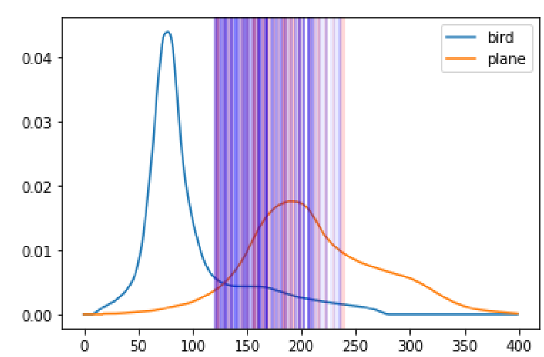
Handling Nan

If a data point is nan, the model simply ignores it, skipping to the next valid data point. I wasn't sure how else to handle these, and anything else would cause some sort of error.

Misclassified Plots

Tracks four and five are pictured below. Assuming default parameters (transition tolerance of 0.9, along with no additional features), both tracks were misclassified as aircrafts, meaning the data below represents a bird. All data points are plotted vertically. The red lines indicate the first 100 data points, and the blue lines represent the last 200. When considering the misclassification, it's clear that a majority of the data points fall on velocities where the likelihood of being an aircraft is greater than the likelihood of being a bird.





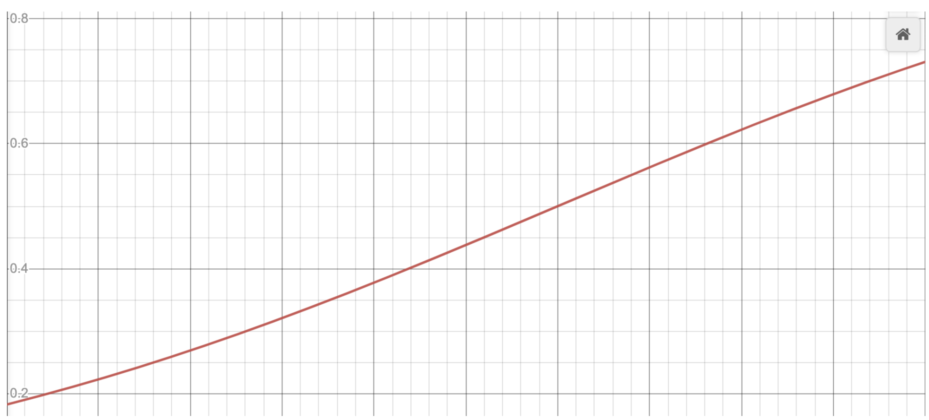
Additional Feature Selection

To decide on an additional feature, I performed various calculations to observe if there were any aspects of the data that may indicate whether a track was a bird or an aircraft. One notable feature of the data was the average change in velocity, or the sum of the total change velocity divided by the number of seconds the object was observed (with valid points of data). The total average changes for all tracks is shown below, along with the aggregate average for each respective category, birds and aircrafts:

```
curr track avg_change 1 (bird): 8.74735950384887
curr track avg_change 2 (plane): 0.5182434710935793
curr track avg_change 3 (bird): 8.895608211868911
curr track avg_change 4 (bird): 15.020233300724618
curr track avg_change 5 (bird): 15.198885862669707
curr track avg_change 6 (plane): 0.5752408603097249
curr track avg_change 7 (plane): 0.5164326830777917
curr track avg_change 8 (plane): 9.681886106700105
curr track avg_change 9 (plane): 0.777778995372013
curr track avg_change 10 (bird): 0.5710483909788877
plane avg change: 2.413916423310643
bird avg change: 9.6866270540182
```

It's clear from the data that birds, on average, have a higher average change in velocity than aircrafts. There are only two tracks that are an exception to this rule (Track 8 and Trak 10). I decided to factor this into the probability updates with a given weight. To decide how to update it, however, I had to create a likelihood function. Since the average change in velocity for a bird was ≈ 2.5, and the average change in velocity for an aircraft was ≈ 9.5, I needed a function that reported a 50% chance for each category when the average change in velocity was 6.0 (halfway between 2.5 and 9.5), and increased from 2.5 to 9.5 to represent the increase in probability of being a bird following an increase in velocity. I decided to adjust a sigmoid function to fit this, creating a smooth transition. I calculated the following equation, with the graph shown to the right:

$$\frac{1}{1+e^{-\left(\frac{1}{4}x-1.5\right)}}$$



With this equation, the network takes into account the current average change, and factors this change into the calculation for the current probability of the sample being a bird or aircraft with weight *w.* The calculations for the update with the added feature are as follows:

P(Bird) = (P(Bird without feature addition) * (1 - w)) + (P(bird | curr_average_change) * w)
P(Plane) = (P(Plane without feature addition) * (1 - w)) + (P(plane | curr_average_change) * w)

With *w* between 0.4 and 0.5 (and a transition tolerance of 0.9, as recommended), the classifier is able to correctly identify all 10 tracks. It should be noted that this feature was created using the entire data set, then tested with the same data. Since there were only 10 tracks total, it seemed pointless to me to split the tracks into training and testing. If there were more than 10 tracks, I'm sure this feature selection wouldn't be as successful in classification.