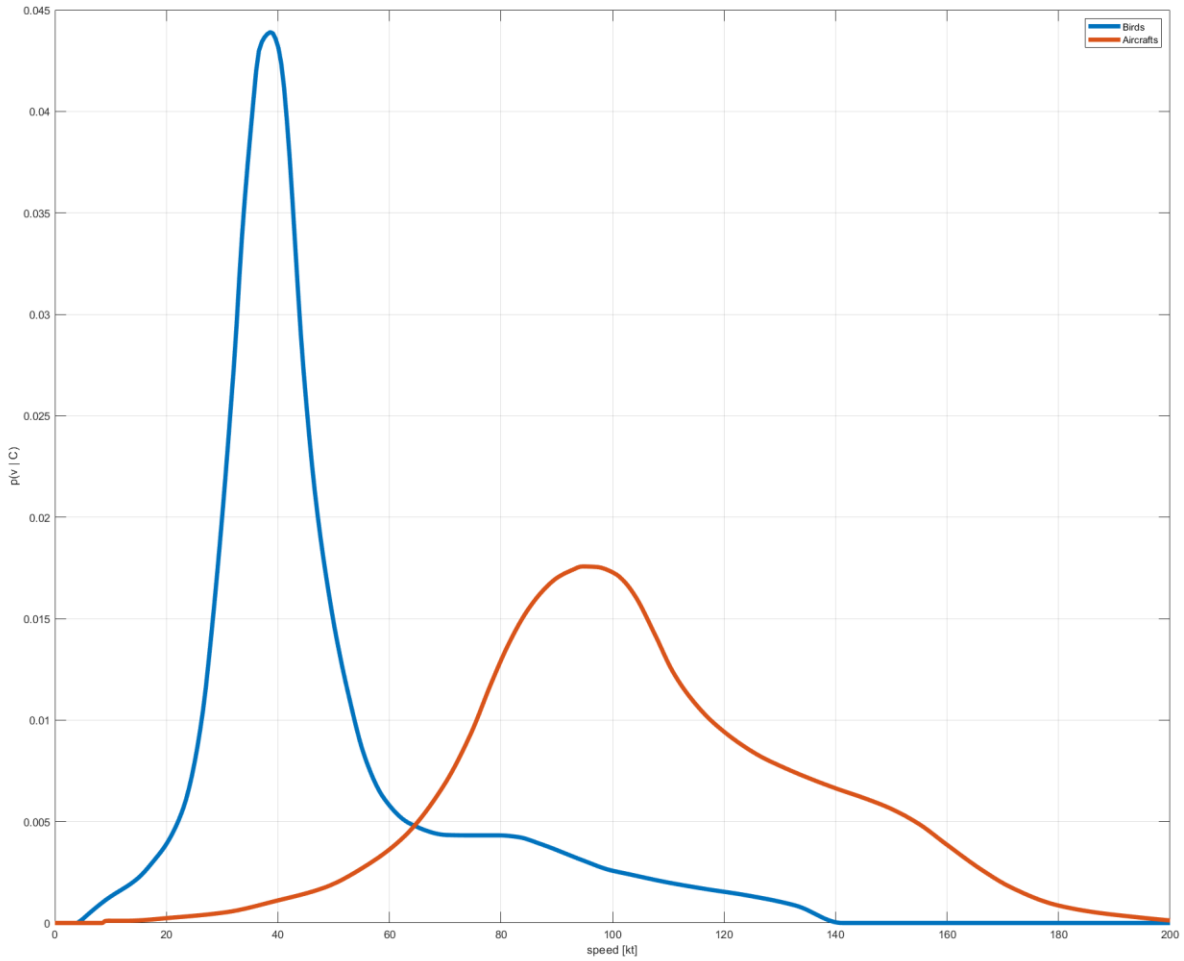
Naïve Bayesian Classification

A RADAR TRACE CLASSIFIER (100 Points)

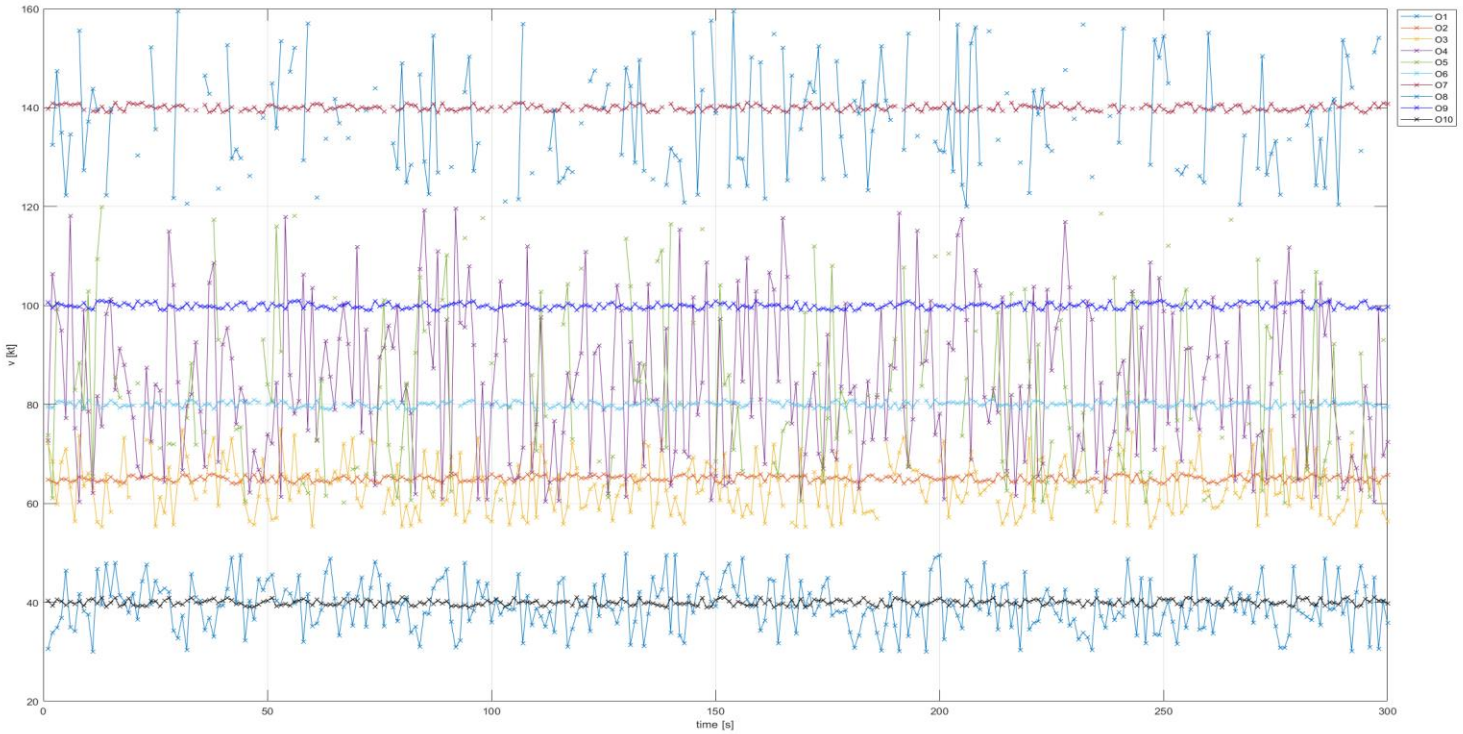
A frequent problem at airports is the collision between aircrafts and birds. You are to solve this problem classifying radar tracks at your disposal into two classes: birds and aircrafts. Using a Naïve Recursive Bayesian classifier, your job is to calculate and report the probability that the object belongs to one of the two classes for each datapoint provided.

For your classification, you are given the following data:

a) The PDF of a specified speed for the two categories of objects as represented here:



b) 10 tracks representing the velocity of the unidentified flying object as it was measured by a military-grade radar (1s sampling frequency for a total length of 300s). If the radar was not able to acquire the target and perform the measurement, the corresponding datapoint will be a **`NaN`** value:



While testing your application, you must consider that in certain cases the speed of the object alone is not sufficient to make a reasonable determination. This might manifest itself with a classification probability close to a $P(O_i = C \mid S) = 0.5$ for $C \in \{bird, aircraft\}$.

Also, assume that the classifier is conservative when transitioning between classes of objects. A probability of transition $P(C_{t+1} = bird \mid C_t = bird) = 0.9$ and $P(C_{t+1} = aircraft \mid C_t = aircraft) = 0.9$ should be sufficient. However, feel free to change these values as appropriate.

As initial probabilities for the classes, it is normal practice to start the classification from equally distributed values (for two classes, it would be 0.5 for each class). Expect these values to change as the classifier acquires more information from the signals.

Could you extract an additional feature from the data to improve the classification? If yes, can you modify your original solution to also include this feature in the classifier? Make sure to explain your rationale in the README file.

SUBMISSION

Python or C++ are the preferred implementation languages. If you are writing in C++, please include a Makefile as well as any other instructions for compilation. For Python, simply provide a plain PY file (no Jupyter notebook).

Your solution may make use of any numerical libraries for pre-processing, fundamental calculations (i.e., linear algebra) and visualization. However, the core portion of your solution must be implemented from scratch.

Submit your solution via Canvas and include a README file that clearly explains its assumptions.

SOLUTION

$O_1 = \boldsymbol{bird}, O_2 = \boldsymbol{aircraft}, O_3 = \boldsymbol{bird}, O_4 = \boldsymbol{bird}, O_5 = \boldsymbol{bird}, O_6 = \boldsymbol{aircraft},$
$O_7 = \boldsymbol{aircraft}, O_8 = \boldsymbol{aircraft}, O_9 = \boldsymbol{aircraft}, O_{10} = \boldsymbol{bird}$