1.

The unsupervised Naive Bayes classification works on the principal factor that the randomiser used to generate class probabilities are done with respects to a non-uniform distribution. This method makes the assumption that for each instance, there is a random non-uniform chance of each class occurring. Although this might be true in most real-world scenarios, certain classification of things could still have uniform distributions, equal amount of probability for each class to occur.

Out of the data-sets given, the mushroom data-set is the closest to being nearly uniformly distributed with the chance of each class being 1/n (where n is the total number of classes). The data-set for Car has two of the classes ‘good’ and ‘vgood’ at a similar probability percentage and hence has a lower accuracy when predicted by the unsupervised Naive Bayes. The other data-sets remain relatively non-uniformly distributed with their classes hence producing a better accuracy with evaluation using the unsupervised Naive Bayes Classifier.

The classifier works well with non-uniform random class distributions and horribly with uniform ones with similar probabilities for each possible class.

Car

3.99% is of class ‘good’

3.76% is of class ‘vgood’

Mushroom

51.8% is of class ‘e’

48.2% is of class ‘p’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | breast-cancer.csv | car.csv | hypothyroid.csv | Mushroom.csv |
| Run 1 (%) | 64.9122 | 19.6531 | 93.3649 | 52.4307 |
| Run 2 (%) | 70.1754 | 70.5202 | 95.1026 | 50.0307 |
| Run 3 (%) | 29.8245 | 68.7861 | 95.7345 | 52.0615 |
| Run 4 (%) | 66.6666 | 70.2312 | 95.5766 | 51.4461 |
| Run 5 (%) | 70.1754 | 69.6531 | 95.1026 | 49.7846 |
| Average (%) |  |  |  |  |

5.

Deterministically labelling the instances would make the predictions rely solely on the randomiser algorithm used to generate those classes. The use of randomly generated non-uniform distributions of classes per instance in the original unsupervised Naive Bayes Classifier allows the random distributions made from the randomiser algorithm to be more spread out across the data set over each instance and create a somewhat realistic distribution over many iterations.

The best case scenario for deterministically labelling would be the evaluation score of 1/n where n is the total number of classes in the dataset. This would only occur when a uniform randomiser algorithm is applied when deterministically labelling, in that case the class predictions would be biased and directly proportional to the number of classes.

Non-uniform randomiser algorithms (deterministic()) used specially without iterations would rely solely on one distribution generated from the random.randint() function and thus would not be normally distributed enough to resemble any real-case scenario and make accurate predictions.