Assignment 4 Report

When the notebook is run, accuracy may be slightly different from the accuracies listed in this report. However, they should be close. Same goes for the times and sizes. The chart is at the end, and conveniently lists answers 3-6 for all 16 trials.

# **Car Data**

1. I partitioned the data into training and test datasets using the scikit-learn library’s train\_test\_split function, with 70/30 train/test split, as recommended by the assignment.

## Naive Bayes

1. Parameters tuned: Used categorical model because it was most accurate
2. Accuracy: 88%
3. Size: 5.34 Kb
4. Prediction time: 0.36s
5. Train time: < 0.1s

## Decision Tree

1. Parameters tuned: None
2. Accuracy: 96.5%
3. Size: 14.9 Kb
4. Prediction time: < 0.1s
5. Train time: < 0.1s

## Support Vector Machine

1. Parameters tuned: None
2. Accuracy: 97%
3. Size: 126.02 Kb
4. Prediction time: 0.12s
5. Train time: 0.17s

## Neural Net

1. Parameters tuned: None
2. Accuracy: 98%
3. Size: 92.1 Kb
4. Prediction time: <0.1s
5. Train time: 3.3s

# **Abalone**

1. Encoded the non-numerical columns of data using scikit-learn’s label encoder, and split using train\_test\_split with a 70/30 split, same as Car data. Then sorted ring counts into 3 classes (1-8, 9-10, 11+), as recommended by studies citing this set.

## Naive Bayes

1. Parameters tuned: Used Gaussian model because it was the most accurate
2. Accuracy: 59%
3. Size: 1.1 Kb
4. Prediction time: <0.1s
5. Train time: 0.2s

## Decision Tree

1. Parameters tuned: None
2. Accuracy: 56%
3. Size: 119.66 Kb
4. Prediction time: <0.1s
5. Train time: 0.1s

## Support Vector Machine

1. Parameters tuned: None
2. Accuracy: 65%
3. Size: 219.49 Kb
4. Prediction time: 0.7s
5. Train time: 1.0s

## Neural Net

1. Parameters tuned: None
2. Accuracy: 66.7%
3. Size: 46.68 Kb
4. Prediction time: <0.1s
5. Train time: 6.2s

# **Madelon Data**

1. Used the designated training data, dropped missing values ready in by CSV, partitioned using train\_test\_split with 70/30 split the same way as the first 2 datasets

## Naive Bayes

1. Parameters tuned: Used Gaussian model
2. Accuracy: 58.2%
3. Size: 16.7 Kb
4. Prediction time: <0.1s
5. Train time: <0.1s

## Decision Tree

1. Parameters tuned: None
2. Accuracy: 70.2%
3. Size: 18.52 Kb
4. Prediction time: <0.1s
5. Train time: 0.8s

## Support Vector Machine

1. Parameters tuned: None
2. Accuracy: 64.7%
3. Size: 5134.88 Kb
4. Prediction time: 0.69s
5. Train time: 0.82s

## Neural Net

1. Parameters tuned: None
2. Accuracy: 50.8%
3. Size: 1611.79 Kb
4. Prediction time: <0.1s
5. Train time: 0.63s

# **KDD Cup Data**

1. Used 10% of the data, as Dobra said we may have to, because it still yielded accurate results while taking significantly less time to compute. Used train\_test\_split in the same way as I did for other datasets, but used a 10/90 train/test split because I found this to be the best compromise of efficiency and accuracy due to the large size of the data.

## Naive Bayes

1. Parameters tuned: Used Gaussian model
2. Accuracy: 96.6%
3. Size: 13.89 Kb
4. Prediction time: 58.3s
5. Train time: 5.0s

## Decision Tree

1. Parameters tuned: None
2. Accuracy: 99.9%
3. Size: 39.33 Kb
4. Prediction time: 42.6s
5. Train time: 3.84s

## Support Vector Machine

1. Parameters tuned: Used Linear SVC model after testing found this to be efficient
2. Accuracy: 98.5%
3. Size: 7.92 Kb
4. Prediction time: 45.5s
5. Train time: 36.4s

## Neural Net

1. Parameters tuned: None
2. Accuracy: 98.6%
3. Size: 201.38 Kb
4. Prediction time: 41.5s
5. Train time: 19.2s

# **Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Car Data** | **Abalone Data** | **Madelon Data** | **KDD Cup Data** |
| **Naive Bayes** | Train: <0.1s  Predict: 0.36s  Accuracy: 88%  Size: 5.34 Kb | Train: 0.2s  Predict: <0.1s  Accuracy: 59%  Size: 1.1 Kb | Train: <0.1s  Predict: <0.1s  Accuracy: 58.2%  Size: 16.7 Kb | Train: 5.0s  Predict: 58.3s  Accuracy: 96.6%  Size: 13.89 Kb |
| **Decision Tree** | Train: <0.1s  Predict: <0.1s  Accuracy: 96.5%  Size: 14.9 Kb | Train: 0.1s  Predict: <0.1s  Accuracy: 56%  Size: 119.66 Kb | Train: 0.8s  Predict: <0.1s  Accuracy: 70.2%  Size: 18.52 Kb | Train: 3.84s  Predict: 42.6s  Accuracy: 99.9%  Size: 39.33 Kb |
| **Support Vector Machine** | Train: 0.17s  Predict: 0.12s  Accuracy: 97%  Size: 126.02 Kb | Train: 1.0s  Predict: 0.7s  Accuracy: 65%  Size: 219.49 Kb | Train: 0.82s  Predict: 0.69s  Accuracy: 64.7%  Size:  5134.88 Kb | Train: 36.4s  Predict: 45.5s  Accuracy: 98.5%  Size: 7.92 Kb |
| **Neural Network** | Train: 3.3s  Predict: <0.1s  Accuracy: 98%  Size: 92.1 Kb | Train: 6.2s  Predict: <0.1s  Accuracy: 66.7%  Size: 46.68 Kb | Train: 0.63s  Predict: <0.1s  Accuracy: 50.8%  Size: 1611.79 Kb | Train: 19.2s  Predict: 41.5s  Accuracy: 98.6%  Size: 201.38 Kb |