Ivan Lytovka, 500861433
Daniyil Kotov, 500877422

Introduction

For the purpose of this assignment we will use the Dry Bean data set from UCI Machine Learning Repository [1]. The rationale to use this data set is because it has an acceptably large number of records, and the classification associated with this data set is diverse. The data set has 13611 records and 17 attributes in each record. There's 7 classes in total, which provides a solid foundation for testing the precision of different classification methods.

The 5 classification methods are: Decision Tree, K-neighbors, Support Vector, Random Forest, and ML classifier. In the next sections we will touch base on every classification method as well as the results achieved from running these methods on our data set. In particular, the key metrics that we will analyze are time and accuracy. In the closing section, we will compare the methods against each other and declare the most efficient approach.

Decision Tree Classifier

```
DT (default - max_depth=None) |
     Accuracy: 0.9992654260528894; Time: 0.06251025199890137s
DT (max_depth=3) |
     Accuracy: 0.9615572967678746; Time: 0.06737899780273438s
DT (max_depth=2) |
     Accuracy: 0.722820763956905; Time: 0.05591702461242676s
```

By default, the Decision Tree classifier has no maximum depth set. At that setting, the accuracy of the classifier is stellar in both accuracy and time. Playing around with the depth setting, we can see that at least on the given dataset, only at max depth of two the Decision Tree classifier's accuracy takes a decent hit, while not saving a lot of time.

Ivan Lytovka, 500861433
Daniyil Kotov, 500877422

K Neighbors Classifier

```
KN (default - n_neighbors=5) |
     Accuracy: 0.8143976493633692; Time: 0.8935089111328125s
KN (n_neighbors=10) |
     Accuracy: 0.7732615083251714; Time: 0.8576581478118896s
KN (n_neighbors=100) |
     Accuracy: 0.64128305582762; Time: 0.9003942012786865s
```

By default, the K Neighbors classifier has the nearest K number of neighbors set to 5, which on our dataset produces the best result. Increasing the K number to both 10 and 100 resulted in a 4% and 18% decrease in accuracy respectively. The time score discrepancy is not large enough to be considered to correlate to the selected number of neighbors.

Support Vector Classifier

```
SV (default - kernel=radial basis) |
     Accuracy: 0.6449559255631734; Time: 5.07098126411438s
SV (kernel=linear) |
     Accuracy: 0.9591087169441724; Time: 36.722963094711304s
SV (kernel=polynomial) |
     Accuracy: 0.6461802154750245; Time: 2.248913049697876s
SV (kernel=sigmoid) |
     Accuracy: 0.3188050930460333; Time: 5.821003198623657s
```

By default, the Support Vector classifier uses a radial basis kernel. At this setting, the accuracy of predictions was at the very least subpar and the runtime disappointing compared to the results of the Decision Tree and K Neighbors classifiers. Changing the kernel setting to linear produced the highest accuracy score out of all of the SVC methods. However, this accuracy came at the price of the longest runtime. The polynomial kernel presented the same accuracy rate as the radial basis one, but cut down the time in half. The sigmoid produced the worst performance in terms of accuracy to time - half the accuracy of the radial basis and polynomial at the speed of the radial basis. Support Vector classifiers in general seem to be unfit for large datasets.

Random Forest Classifier

```
RF (default - n_estimators=100) |
      Accuracy: 0.9987757100881489; Time: 1.4801712036132812s
RF (default - n_estimators=800) |
      Accuracy: 0.9990205680705191; Time: 11.987420797348022s
RF (default - n_estimators=50) |
      Accuracy: 0.9987757100881489; Time: 0.7603299617767334s
RF (default - n_estimators=10) |
      Accuracy: 0.9953476983349657; Time: 0.09253907203674316s
```

By default, the Random Forest Classifier has the number of estimators set to 100. At this setting, the classifier shows close to perfect accuracy with a subpar runtime. Tweaking the number of estimators to both larger and smaller numbers does not produce a noticeable change in accuracy. However, it does produce one in runtime. The higher the number of estimator trees, the longer the runtime. So, decreasing the number of estimators to 10 allowed us to preserve the accuracy performance, while decreasing the runtime to that of the Decision Tree classifier.

ML Classifier

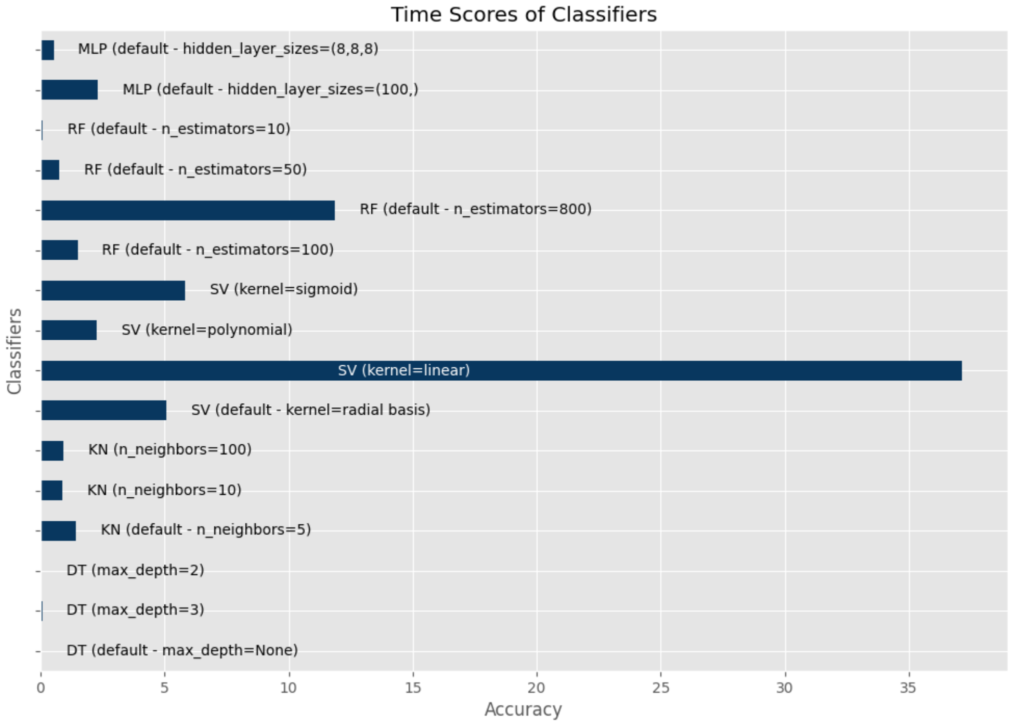
```
MLP (default - hidden_layer_sizes=(100,)

     | Accuracy: 0.6349167482859941; Time: 2.241055965423584s

MLP (default - hidden_layer_sizes=(3,3,3)

     | Accuracy: 0.2605288932419197; Time: 1.121068000793457s

MLP (default - hidden_layer_sizes=(8,8,8)

     | Accuracy: 0.33178256611165524; Time: 0.8534829616546631s
```

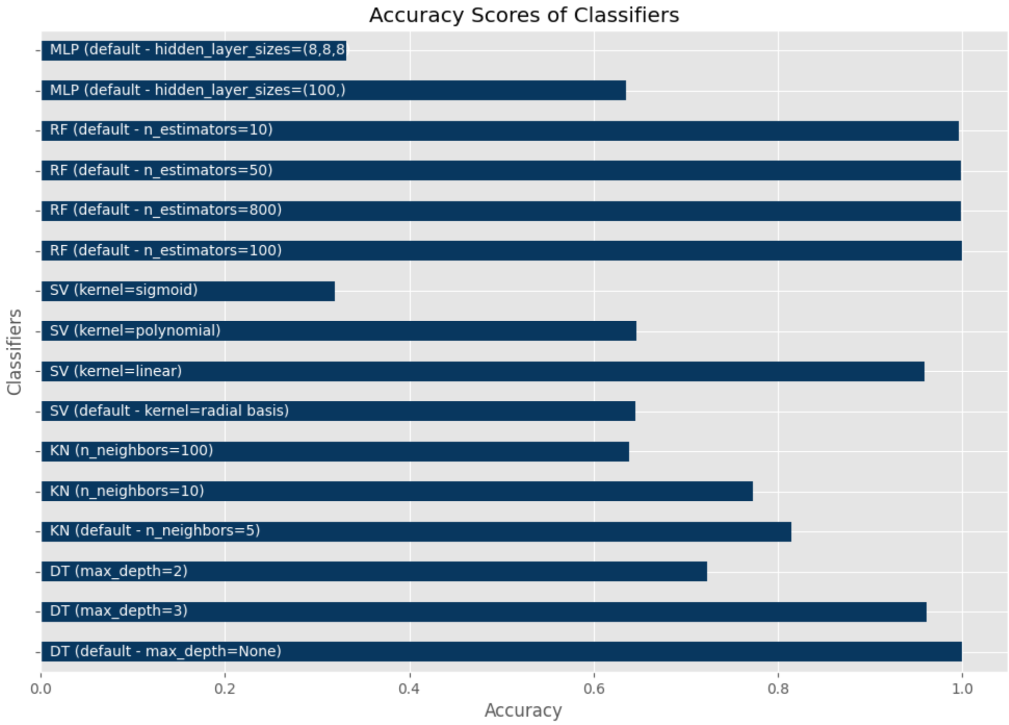
By default, the ML classifier has a hidden layer size set to 1 with 100 hidden units, which on our dataset produces the best results in terms of accuracy and the worst results in terms of time. Despite better time results. setting the hidden layer size to (x, x, x) does not generate reliable classification as the accuracy score varies between 0.26-0.33. ML classifiers in general seem to be unfit for large datasets.

Conclusion

The two graphs below compare the performance of all classification methods. The first graph plots time scores, while the second graph plots accuracy of classifiers. Based on the findings, we conclude that the Decision Tree classification with the default constructor parameters takes precedence over every other method both in terms of accuracy and time. Random Forest classifier also produces the top-notch accuracy on the given data set; however, time score is slightly inferior to Decision Tree classification. Support Vector classifier (kernel=sigmoid) and ML classifier are among outsiders for accuracy scores.

Ivan Lytovka, 500861433
Daniyil Kotov, 500877422





Ivan Lytovka, 500861433
Daniyil Kotov, 500877422

References:

[1] Dry Bean Dataset Data Set: https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset