Trevor Buck
Assignment 3
Machine Learning

1.A

Logistic Regression did extremely well for dataset A, but awful for dataset B.
Naive Bayes was somewhat consistent but got worse as the dataset got more complicated.
SVM(SVC) performed with each dataset. By far the best overall model.
The Decision Tree Classifier was consistent throughout the dataset with a good, not great, score.
Random Forest was slightly higher than the decision tree.

2.B

The more linearly separable the dataset is, the better any classifier will perform. This is especially true when it comes to the Logistic Regression model. It is also helpful if the dataset has clear decision boundaries (ie. the space between data points is bigger). This makes it easier for the classifier and produces a better score.

2A.1

Models: (mean), (std)

Logistic Regression:	[0.85387115, 0.06528647]
Naive Bayes:	[0.7817391, 0.05166311]
SVM(SVC):	[0.542004, 0.0040329]
Decision Tree:	[0.79794318, 0.04623075]
Random Forest:	[0.8459888, 0.04985234]
Multi-Layer Processor:	[0.73185194, 0.07729417]
Perceptron:	[0.5180072, 0.08635789]

2A.2

My best model was the Random Forest model. I chose this classifier because it has the second highest mean, with the second lowest standard deviation. Meaning that it is very accurate across all models and is therefore consistent. I chose this over the logistic regression model (highest mean) because it had almost half the standard deviation.

2A.3

MLP: Consists of 3 layers – an Input Layer, a Hidden Layer, and an Output Layer. The last two layers rely on some sort of activation function. These then relay the information back to the previous layer (back propagation).
Perceptron: Very similar to the MLP, except that it can only learn linearly separable patterns. It takes inputs, applies weights/functions to them, and produces an output. Can only produce binary classifiers. (good vs bad)

2B.1

I used the Exhaustive Grid Search. This type of search generates all the possible combinations from a 'grid' of parameters. For example, if you have 5 values for 'C' and 5 values for 'gamma', then it would run 25 different searches, each utilizing a different combination of the 2 parameters.

2B.2

C: The penalty parameter. Determines how much error is allowed by the search. (margin)
Gamma: Kernel coefficient. While normally used for non-linear hyperplanes, the gamma value affects the 'curviness' of the classifier. Higher values lead to overfitting.
Max Depth: Represents how many levels or splits are in each tree. Like the gamma value, the higher the depth, the more specific the classifier is to the training set.
N Estimators: Represents the number of trees in the forest. The more trees you have usually result in a better score; while the tradeoff is time complexity.

2B.3

Yes, but only slightly…

SVM(SVC): [0.584571, 0.0041876]
Random Forest: [0.8842351, 0.05243681]