Project 2: Feature Selection with Nearest Neighbor

Teammate 1 Name: Josh McIntyre SID: 862054277

Teammate 2 Name: Edwin Leon SID: 862132870

Solution:

Dataset	Best Feature Set	Accuracy
Small Number: 41	Forward Selection = {4, 5}	0.94
	Backward Elimination = {4, 5}	0.94
	Custom Algorithm = Not implemented	
Large Number: 41	Forward Selection = {5,14}	0.96
	Backward Elimination = {18, 5}	0.85
	Custom Algorithm = Not implemented	

In completing this project, I consulted following resources:

statistics — Mathematical statistics functions — Python 3.9.5 documentation

I. Introduction

This project demonstrates the sensitivity of the nearest neighbor classifier using two greedy algorithms. The algorithms used are the forward selection algorithm and the backwards elimination algorithm. Moreover, we are going to use the leave-one-out validation evaluation function alongside the NN classifier to test 2 data sets. The program will prompt the user to input the number of features to use and the algorithm he/she wants to use. The program will output the set of features with the best accuracy.

II. Challenges

During the first test of our project, we realized something was wrong because our results did not match the given results for the small dataset. After careful debugging, the normalization is what was incorrect. After correcting the normalization code, the results for the first two tests matched.

III. Code Design

The code is designed in object oriented Python. The two classes we created are Node and Classifier. About 50 lines of code went into prompts and input validation.

The Node class encapsulates the validator function. The Node class requires a feature subset to be passed into its constructor. The validator function in Node takes no input, and uses the Node's feature subset member in performing leave-one-out validation to calculate and return the accuracy of the feature subset.

The Classifier class has a function called Train which takes a file name. Train then reads the dataset specified by the file name, normalizes the data, and initializes Classifier's data member function to be the normalized data. Classifier also has a Test function which takes an instance of the dataset, performs the nearest neighbor algorithm on this instance, and returns the classifier based on the determined closest neighbor.

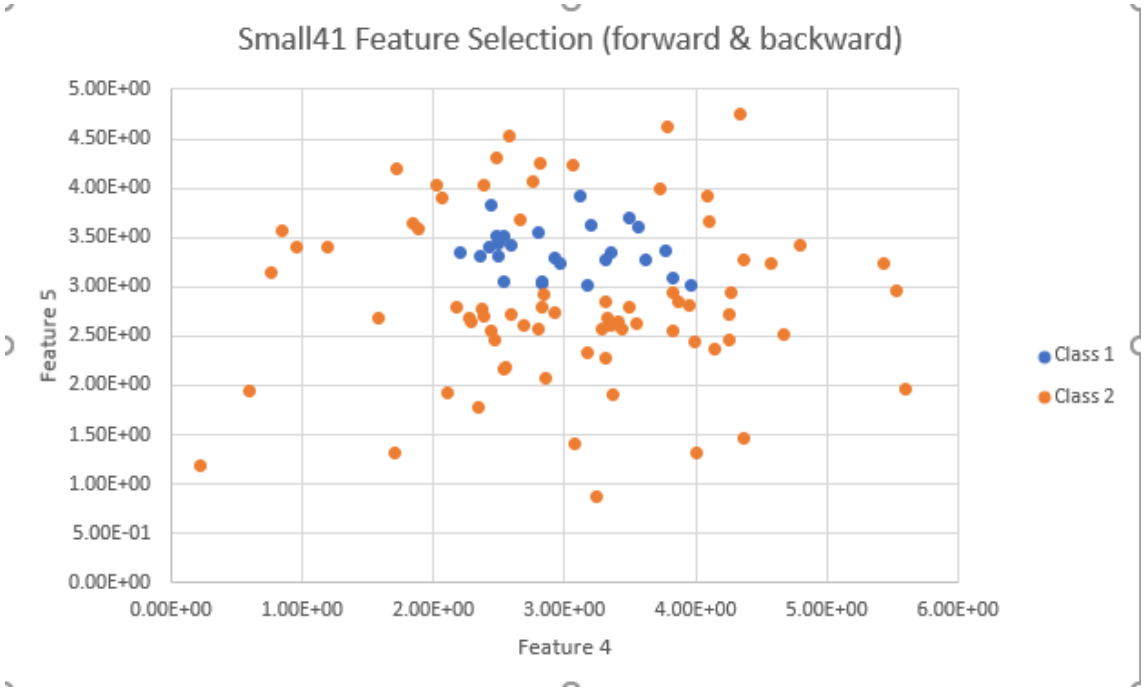
The Classifier function is used inside the Node class's validator function in order to test each instance. Inside Node's validator function there is an instance of Classifier which calls Train to read the specified dataset file, initialize a new dataset variable containing only the features specified by Node's feature subset member variable, and calls Test for each instance in that dataset variable in order to calculate and return the accuracy.

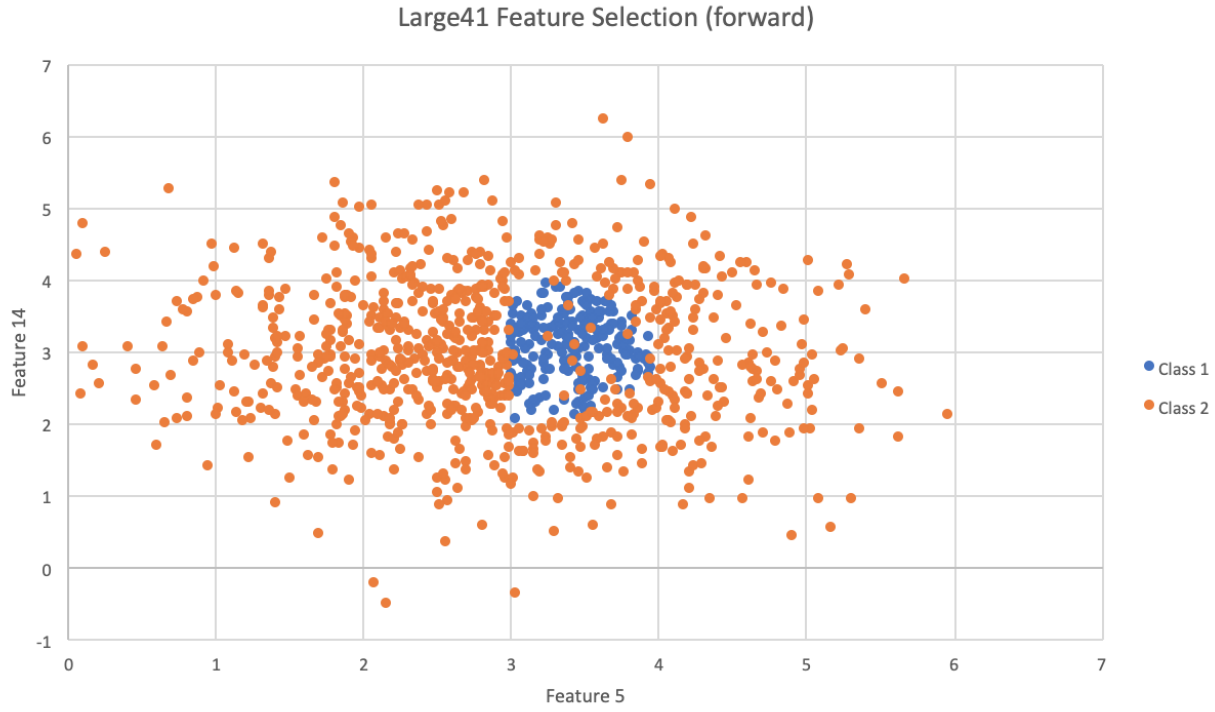
The main execution of the program contains all of the prompting, input validation, and the core implementation of the forward and backward selection algorithms.

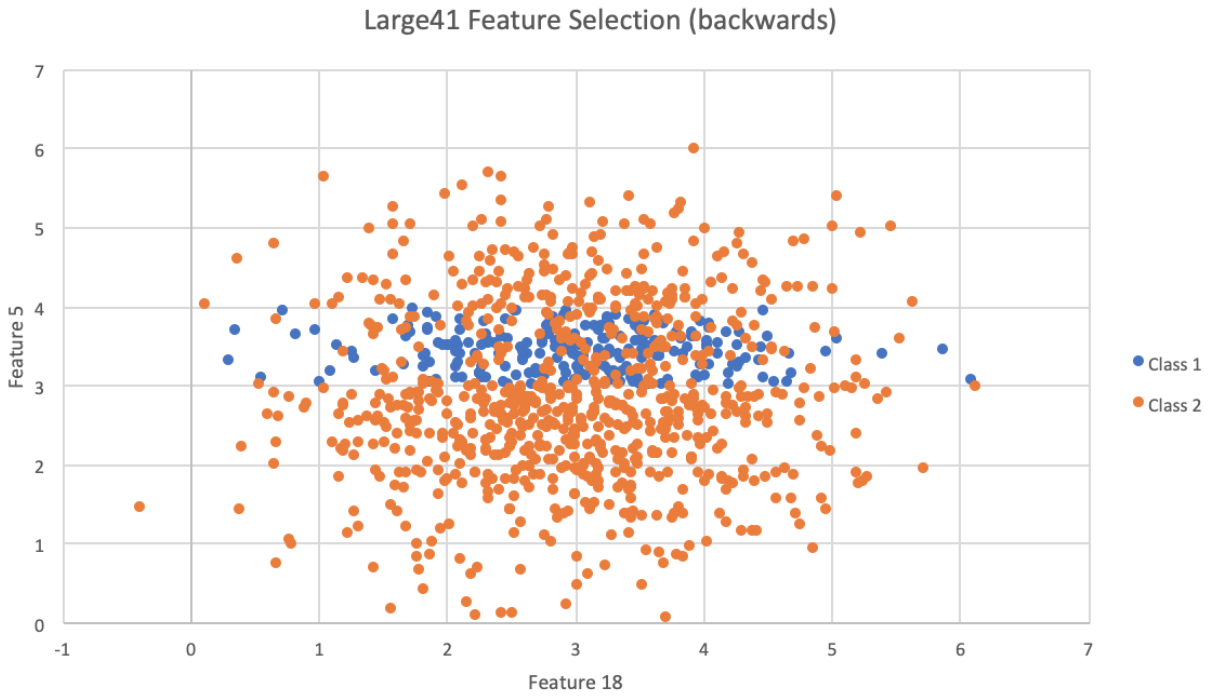
IV. Dataset details

Small Dataset: 10, 100

Large Dataset: 40, 1000







V. Algorithms

1. Forward Selection: The forward selection algorithm starts by checking the accuracy of the set/state with the least amount of features possible which would be 0 features. Next, it increments the number of features by 1 and checks all the states. Then it selects the set/state with the highest accuracy. Once again, the number of features is incremented by 1 and checks all the states whose parent is the set/state with the highest accuracy. The set/state with the highest accuracy is selected. The process repeats until the maximum number of features is reached. The algorithm keeps track of the set/state with the highest accuracy.

2. Backward Elimination: The backwards elimination algorithm starts by checking the accuracy of the set/state with the maximum amount of features possible (Only 1 state will have all features). Next, it decrements the number of features by 1 and checks all the states. Then it selects the set/state with the highest accuracy. Once again, the number of features is decremented by 1 and checks all the states that are a parent of the set/state with the highest accuracy. The set/state with the highest accuracy is selected. The process repeats until the set/state with no features is reached. The algorithm keeps track of the set/state with the highest accuracy.

VI. Analysis

Comparing Forward Selection vs Backward Elimination:

Using the small41 data set, both the forward selection and backwards elimination algorithms output the same results. The best feature set was {4,5} which gave an accuracy of 94%.

Using the large41 data set, both the forward selection and backwards elimination algorithms output different results. The best feature set was {5,14} for forward selection and the best subset was {18, 5} for backward elimination with accuracies of 96% and 85% respectively.

Forward selection had an average accuracy of 95% and backward elimination had an average accuracy of 90%.

VII. Conclusion

Our findings demonstrate that forward selection and backward selection are not optimal algorithms; however, they did find solutions with average accuracies in the nineties. The forward selection algorithm feature subsets were {4, 5} for small and {5, 14} for large. The backward selection algorithm features subsets were {4, 5} for small and {18, 5} for large. The respective accuracies for small and large average accuracies was 0.94 and 0.905.

One code optimization that can take place is moving the code that reads the dataset file into a larger scope, so that it does not get called as often. Reading from the file is expensive, so reducing the times it is called will increase the performance of the software.

VIII. Trace of Small Data Set

```
Welcome to Edwin Leon and Josh McIntyre's Feature Selection Algorithm
Please enter the total number of features
10
Type the number of the algorithm you want to run
1. Forward Selection
2. Backward Elimination
1
Using no features and "random" evaluation, we get an accuracy of 0.0%
Beginning search.
Using feature(s) {1} accuracy is 62.0%
Using feature(s) {2} accuracy is 56.00000000000001%
Using feature(s) {3} accuracy is 64.0%
Using feature(s) {4} accuracy is 56.99999999999999%
Using feature(s) {5} accuracy is 83.0%
Using feature(s) {6} accuracy is 52.0%
```

Using feature(s) {7} accuracy is 70.0%
Using feature(s) {8} accuracy is 51.0%
Using feature(s) {9} accuracy is 63.0%
Using feature(s) {10} accuracy is 60.0%

Feature set {5} was best, accuracy is 83.0%

Using feature(s) {1, 5} accuracy is 76.0%
Using feature(s) {2, 5} accuracy is 80.0%
Using feature(s) {3, 5} accuracy is 77.0%
Using feature(s) {4, 5} accuracy is 94.0%
Using feature(s) {5, 6} accuracy is 81.0%
Using feature(s) {5, 7} accuracy is 79.0%
Using feature(s) {8, 5} accuracy is 74.0%
Using feature(s) {9, 5} accuracy is 71.0%
Using feature(s) {10, 5} accuracy is 83.0%

Feature set {4, 5} was best, accuracy is 94.0%

Using feature(s) {1, 4, 5} accuracy is 84.0%
Using feature(s) {2, 4, 5} accuracy is 86.0%
Using feature(s) {3, 4, 5} accuracy is 88.0%
Using feature(s) {4, 5, 6} accuracy is 87.0%
Using feature(s) {4, 5, 7} accuracy is 90.0%
Using feature(s) {8, 4, 5} accuracy is 85.0%
Using feature(s) {9, 4, 5} accuracy is 88.0%
Using feature(s) {10, 4, 5} accuracy is 89.0%

Feature set {4, 5, 7} was best, accuracy is 90.0%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)

Using feature(s) {1, 4, 5, 7} accuracy is 84.0%
Using feature(s) {2, 4, 5, 7} accuracy is 90.0%
Using feature(s) {3, 4, 5, 7} accuracy is 84.0%
Using feature(s) {4, 5, 6, 7} accuracy is 85.0%
Using feature(s) {8, 4, 5, 7} accuracy is 87.0%
Using feature(s) {9, 4, 5, 7} accuracy is 83.0%
Using feature(s) {10, 4, 5, 7} accuracy is 88.0%

Feature set {2, 4, 5, 7} was best, accuracy is 90.0%

Using feature(s) {1, 2, 4, 5, 7} accuracy is 81.0%
Using feature(s) {2, 3, 4, 5, 7} accuracy is 79.0%
Using feature(s) {2, 4, 5, 6, 7} accuracy is 77.0%
Using feature(s) {2, 4, 5, 7, 8} accuracy is 78.0%

Using feature(s) {2, 4, 5, 7, 9} accuracy is 80.0%
Using feature(s) {2, 4, 5, 7, 10} accuracy is 85.0%

Feature set {2, 4, 5, 7, 10} was best, accuracy is 85.0%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)

Using feature(s) {1, 2, 4, 5, 7, 10} accuracy is 73.0%
Using feature(s) {2, 3, 4, 5, 7, 10} accuracy is 78.0%
Using feature(s) {2, 4, 5, 6, 7, 10} accuracy is 80.0%
Using feature(s) {2, 4, 5, 7, 8, 10} accuracy is 70.0%
Using feature(s) {2, 4, 5, 7, 9, 10} accuracy is 78.0%

Feature set {2, 4, 5, 6, 7, 10} was best, accuracy is 80.0%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)

Using feature(s) {1, 2, 4, 5, 6, 7, 10} accuracy is 77.0%
Using feature(s) {2, 3, 4, 5, 6, 7, 10} accuracy is 77.0%
Using feature(s) {2, 4, 5, 6, 7, 8, 10} accuracy is 68.0%
Using feature(s) {2, 4, 5, 6, 7, 9, 10} accuracy is 69.0%

Feature set {2, 3, 4, 5, 6, 7, 10} was best, accuracy is 77.0%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)

Using feature(s) {1, 2, 3, 4, 5, 6, 7, 10} accuracy is 69.0%
Using feature(s) {2, 3, 4, 5, 6, 7, 8, 10} accuracy is 64.0%
Using feature(s) {2, 3, 4, 5, 6, 7, 9, 10} accuracy is 73.0%

Feature set {2, 3, 4, 5, 6, 7, 9, 10} was best, accuracy is 73.0%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)

Using feature(s) {1, 2, 3, 4, 5, 6, 7, 9, 10} accuracy is 67.0%
Using feature(s) {2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 60.0%

Feature set {1, 2, 3, 4, 5, 6, 7, 9, 10} was best, accuracy is 67.0%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)

Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 66.0%
(Warning, Accuracy has decreased! Continuing search in case of local maxima)

Finished search!! The best feature subset is {4, 5}, which has an accuracy of 94.0%