ilorn001@ucr.edu

CS170: Introduction to Artificial Intelligence

Dr. Eamonn Keogh

14 March 2021

Project 2 Report

Table of Contents

Introduction

In real life implementations of A.I., datasets are used for teaching machines to recognize patterns of input into a logical output. The input parameters are called ‘features.’ It is common that not all features that we initially consider end up being useful for obtaining optimal results.

Feature Selection is a process in which we attempt to weed out unnecessary features from our data set. This project implements the concept of feature selection to showcase its effectiveness by implementing search algo.

This project was conducted using Matlab. All code is original except for the graph search function provided by the professor.

Feature Selection Via Search Algorithm

This project is meant to explore different implementations of search algorithms for rating and selecting features of a dataset. This report will contain the results of a comparison of two search algorithms that both attempt to find the set of features that captures the highest accuracy rate.

Comparison of Algorithms

This project is meant to explore different implementations of search algorithms for rating and selecting features of a dataset. This report will contain the results of a comparison of two search algorithms that both attempt to find the set of features that captures the highest accuracy rate.

The two search algorithms we will be using are

- Forward Selection
- Backwards Elimination

These two search algorithms may sound opposite to each other, but they operate almost identically.

Forward Selection

Forward Selection is a search algorithm that can be used in feature selection as follows. Starting from the empty set, test the accuracies of all possible adjacent sets by adding a single feature. Out of all the possible adjacent sets (sets that contain one more unique feature than the

current set), expand the one that scores the highest accuracy. This continues until we have the full set of features. Figure 1 below visualizes this algorithm.

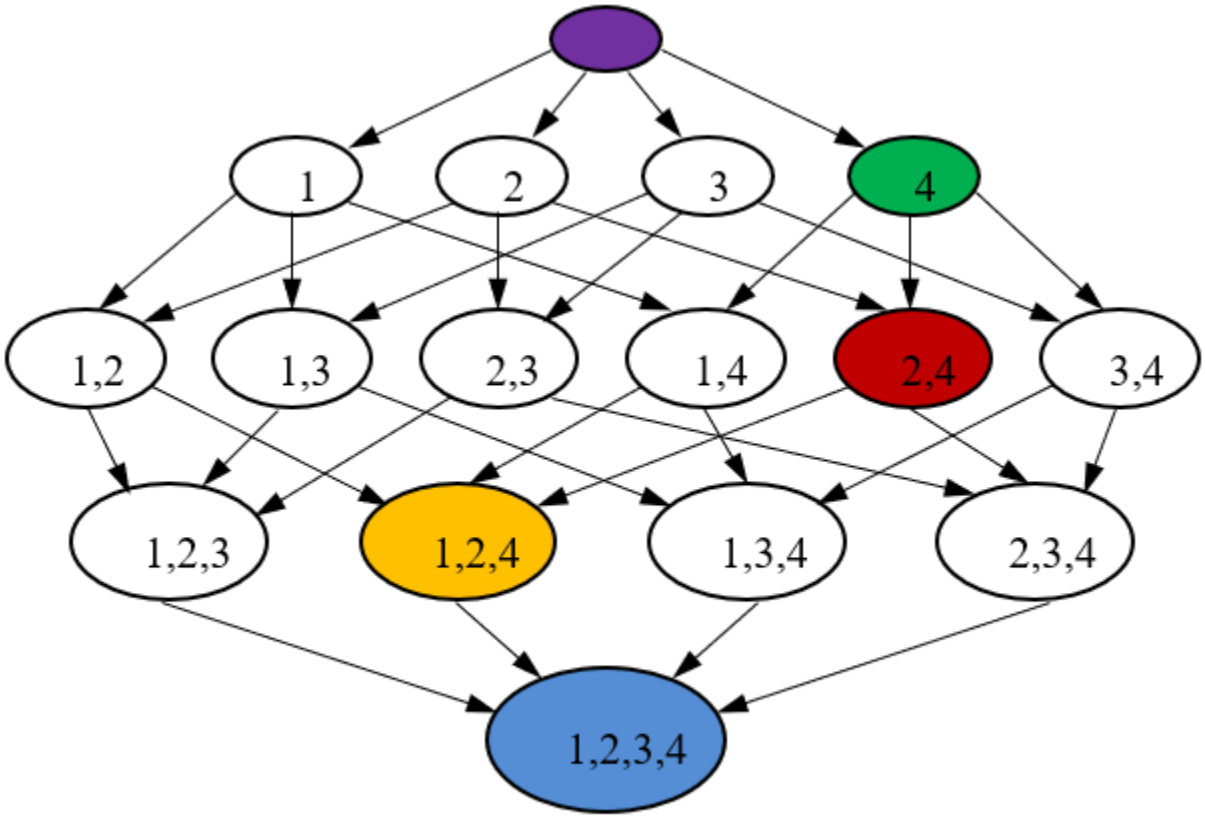


Figure 1: Forward Search on a System of Four Features
Credit: Project_2_Briefing Slides, Eamonn Keogh

In this example, feature 4 is determined to be the best performing single feature. Then, all sets on the next step must contain the previous set, which is just 4. This continues until we reach the full set of features. From there, we compare our resulting performances at each step and select the one that performs best overall.

Backwards Elimination

Backwards Elimination performs almost exactly the same as Forward Selection.The key difference lies in the graph traversal. Backwards Elimination starts with the full set of features, as opposed to the empty set. It then compares how performance changes as it considers removing each feature one at a time. On each level, the algorithm removes the feature that it performs the best without, and continues to evaluate the sets adjacent to the new set with one less feature. This continues until we reach the empty set, and then evaluate all performances and select the best one overall.

Example of Algorithms on Small Data Set

The following will be snippets of my code's generated output for a small dataset. This dataset is the same as contained in the provided file, "CS170_SMALLtestdata__42.txt".

The following is the initial output, The whole output is hidden for space convenience.

```
Reading Data Set from file "CS170_SMALLtestdata__42.txt"
Read file of 300 data points with 10 features each.
> Starting forward search <
On level 1 of search tree
--Considering adding feature 1
----Accuracy: 0.64667
--Considering adding feature 2
----Accuracy: 0.67
--Considering adding feature 3
----Accuracy: 0.69
--Considering adding feature 4
----Accuracy: 0.68
--Considering adding feature 5
----Accuracy: 0.69
--Considering adding feature 6
----Accuracy: 0.61
--Considering adding feature 7
----Accuracy: 0.65333
--Considering adding feature 8
----Accuracy: 0.63667
--Considering adding feature 9
Elapsed time is 19.543502 seconds.
> Starting backward search <
--Considering dropping feature 7
----Accuracy: 0.68
--Considering dropping feature 8
----Accuracy: 0.69333
--Considering dropping feature 9
----Accuracy: 0.66667
--Considering dropping feature 10
----Accuracy: 0.69
On level 1, dropped feature 6.
On level 2 of search tree
--Considering dropping feature 1
----Accuracy: 0.7
--Considering dropping feature 2
----Accuracy: 0.69667
--Considering dropping feature 3
----Accuracy: 0.69333
--Considering dropping feature 4
----Accuracy: 0.71
Elapsed time is 25.224727 seconds.
```

The entire output for Forward Selection and Backward Search is shown below.

1.
```
On level 1 of search tree
--Considering adding feature 1
----Accuracy: 0.64667
--Considering adding feature 2
----Accuracy: 0.67
--Considering adding feature 3
----Accuracy: 0.69
--Considering adding feature 4
----Accuracy: 0.68
--Considering adding feature 5
----Accuracy: 0.69
--Considering adding feature 6
----Accuracy: 0.61
--Considering adding feature 7
----Accuracy: 0.65333
--Considering adding feature 8
----Accuracy: 0.63667
--Considering adding feature 9
----Accuracy: 0.83333
--Considering adding feature 10
----Accuracy: 0.65
On level 1, added feature 9.
On level 2 of search tree
--Considering adding feature 1
----Accuracy: 0.8
--Considering adding feature 2
----Accuracy: 0.79333
--Considering adding feature 3
----Accuracy: 0.81333
--Considering adding feature 4
----Accuracy: 0.76667
--Considering adding feature 5
----Accuracy: 0.91333
--Considering adding feature 6
----Accuracy: 0.82333
```

2.
```
--Considering adding feature 7
----Accuracy: 0.78333
--Considering adding feature 8
----Accuracy: 0.80333
--Considering adding feature 10
----Accuracy: 0.78667
On level 2, added feature 5.
On level 3 of search tree
--Considering adding feature 1
----Accuracy: 0.87333
--Considering adding feature 2
----Accuracy: 0.84
--Considering adding feature 3
----Accuracy: 0.85333
--Considering adding feature 4
----Accuracy: 0.84667
--Considering adding feature 6
----Accuracy: 0.82333
```

3.
```
--Considering adding feature 7
----Accuracy: 0.86
--Considering adding feature 8
----Accuracy: 0.85333
--Considering adding feature 10
----Accuracy: 0.85333
On level 3, added feature 1.
On level 4 of search tree
--Considering adding feature 2
----Accuracy: 0.78667
--Considering adding feature 3
----Accuracy: 0.80333
--Considering adding feature 4
----Accuracy: 0.79667
--Considering adding feature 6
----Accuracy: 0.83
--Considering adding feature 7
```

4.
```
----Accuracy: 0.79
```

```
--Considering adding feature 8
----Accuracy: 0.82333
--Considering adding feature 10
----Accuracy: 0.80333
On level 4, added feature 6.
On level 5 of search tree
--Considering adding feature 2
----Accuracy: 0.74667
--Considering adding feature 3
----Accuracy: 0.77667
--Considering adding feature 4
----Accuracy: 0.78
--Considering adding feature 7
----Accuracy: 0.75667
--Considering adding feature 8
----Accuracy: 0.80667
--Considering adding feature 10
----Accuracy: 0.81333
```

5.

```
On level 5, added feature 10.
On level 6 of search tree
--Considering adding feature 2
----Accuracy: 0.76
--Considering adding feature 3
----Accuracy: 0.73667
--Considering adding feature 4
----Accuracy: 0.75
--Considering adding feature 7
----Accuracy: 0.75333
--Considering adding feature 8
----Accuracy: 0.78667
On level 6, added feature 8.
On level 7 of search tree
--Considering adding feature 2
----Accuracy: 0.71333
--Considering adding feature 3
----Accuracy: 0.73
```

6.

```
--Considering adding feature 4
----Accuracy: 0.75333
--Considering adding feature 7
----Accuracy: 0.75333
On level 7, added feature 4.
On level 8 of search tree
--Considering adding feature 2
----Accuracy: 0.71333
--Considering adding feature 3
----Accuracy: 0.68667
--Considering adding feature 7
----Accuracy: 0.72333
On level 8, added feature 7.
On level 9 of search tree
--Considering adding feature 2
----Accuracy: 0.67333
--Considering adding feature 3
```

7.

```
--Considering adding feature 3
----Accuracy: 0.7
On level 9, added feature 3.
On level 10 of search tree
--Considering adding feature 2
----Accuracy: 0.67333
On level 10, added feature 2.
```

8.

The following is the whole printed output for Backwards Elimination.

1.
```
On level 1 of search tree
--Considering dropping feature 1
----Accuracy: 0.69
--Considering dropping feature 2
----Accuracy: 0.7
--Considering dropping feature 3
----Accuracy: 0.67333
--Considering dropping feature 4
----Accuracy: 0.69667
--Considering dropping feature 5
----Accuracy: 0.67
--Considering dropping feature 6
----Accuracy: 0.70333
--Considering dropping feature 7
----Accuracy: 0.68
--Considering dropping feature 8
----Accuracy: 0.69333
--Considering dropping feature 9
----Accuracy: 0.66667
--Considering dropping feature 10
----Accuracy: 0.69
On level 1, dropped feature 6.
```

2.
```
On level 2 of search tree
--Considering dropping feature 1
----Accuracy: 0.7
--Considering dropping feature 2
----Accuracy: 0.69667
--Considering dropping feature 3
----Accuracy: 0.69333
--Considering dropping feature 4
----Accuracy: 0.71
--Considering dropping feature 5
----Accuracy: 0.7
--Considering dropping feature 7
----Accuracy: 0.73
--Considering dropping feature 8
----Accuracy: 0.73667
--Considering dropping feature 9
----Accuracy: 0.66667
--Considering dropping feature 10
----Accuracy: 0.69667
On level 2, dropped feature 8.
```

3.
```
On level 3 of search tree
--Considering dropping feature 1
----Accuracy: 0.68
--Considering dropping feature 2
----Accuracy: 0.72
--Considering dropping feature 3
----Accuracy: 0.71
--Considering dropping feature 4
----Accuracy: 0.72667
--Considering dropping feature 5
----Accuracy: 0.71333
--Considering dropping feature 7
----Accuracy: 0.71333
--Considering dropping feature 9
----Accuracy: 0.67333
--Considering dropping feature 10
----Accuracy: 0.70667
On level 3, dropped feature 4.
```

4.
```
On level 4 of search tree
--Considering dropping feature 1
----Accuracy: 0.74667
--Considering dropping feature 2
----Accuracy: 0.73333
--Considering dropping feature 3
----Accuracy: 0.71333
--Considering dropping feature 5
----Accuracy: 0.69333
--Considering dropping feature 7
----Accuracy: 0.75
```

```
--Considering dropping feature 9
----Accuracy: 0.69
--Considering dropping feature 10
----Accuracy: 0.71333
On level 4, dropped feature 7.
On level 5 of search tree
--Considering dropping feature 1
----Accuracy: 0.76
--Considering dropping feature 2
----Accuracy: 0.77
--Considering dropping feature 3
----Accuracy: 0.77333
--Considering dropping feature 5
----Accuracy: 0.70667
--Considering dropping feature 9
----Accuracy: 0.70333
--Considering dropping feature 10
```

5. ----Accuracy: 0.74333

```
On level 5, dropped feature 3.
On level 6 of search tree
--Considering dropping feature 1
----Accuracy: 0.8
--Considering dropping feature 2
----Accuracy: 0.80333
--Considering dropping feature 5
----Accuracy: 0.74333
--Considering dropping feature 9
----Accuracy: 0.74
--Considering dropping feature 10
----Accuracy: 0.78667
On level 6, dropped feature 2.
On level 7 of search tree
--Considering dropping feature 1
----Accuracy: 0.85333
```

6. --Considering dropping feature 5

```
--Considering dropping feature 9
----Accuracy: 0.73333
--Considering dropping feature 10
----Accuracy: 0.87333
On level 7, dropped feature 10.
On level 8 of search tree
--Considering dropping feature 1
----Accuracy: 0.91333
--Considering dropping feature 5
----Accuracy: 0.8
--Considering dropping feature 9
----Accuracy: 0.69667
On level 8, dropped feature 1.
On level 9 of search tree
--Considering dropping feature 5
----Accuracy: 0.83333
--Considering dropping feature 9
```

7. ----Accuracy: 0.69

```
On level 9, dropped feature 5.
On level 10 of search tree
--Considering dropping feature 9
----Accuracy: 0.78667
```

8. On level 10, dropped feature 9.

Figure 2 below shows the best accuracies at each level of the forward search.

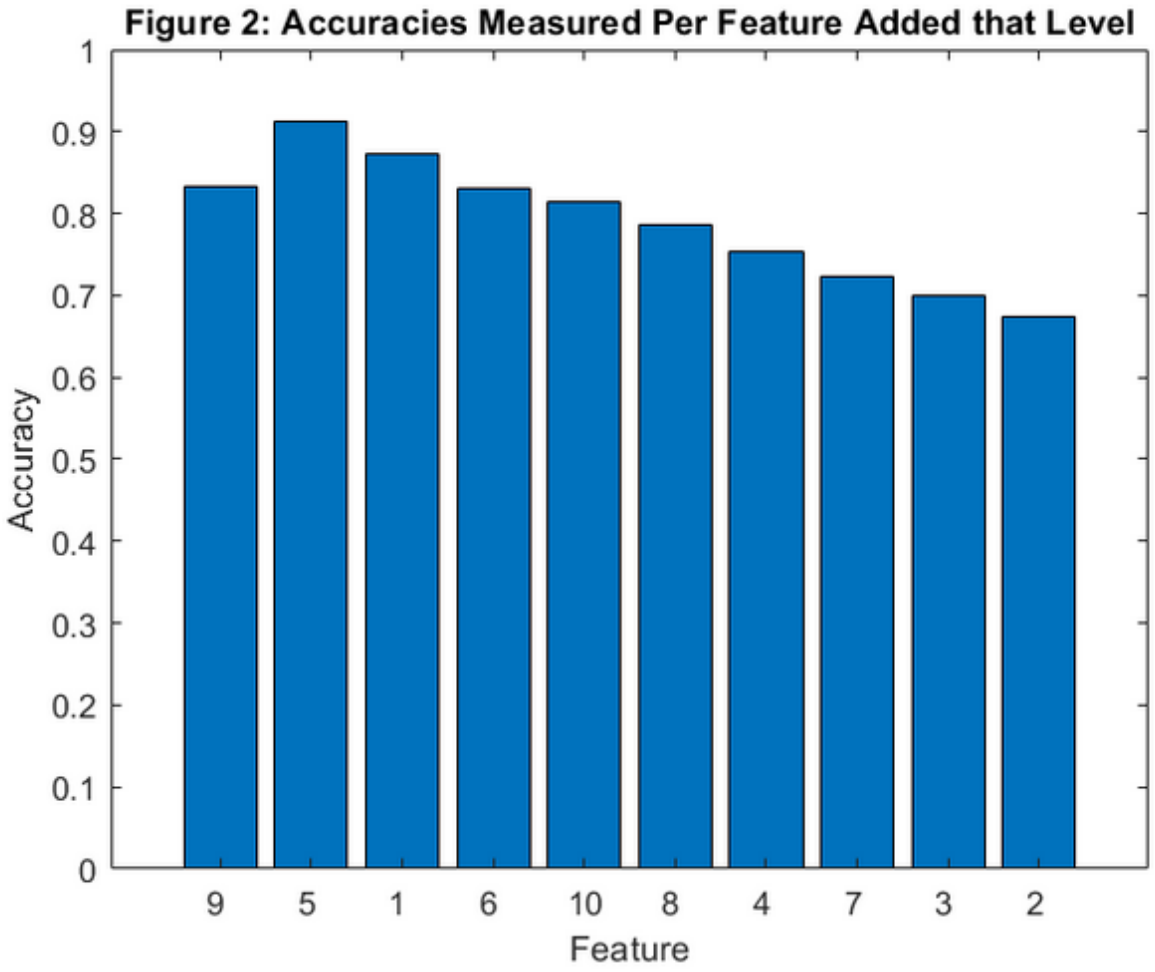
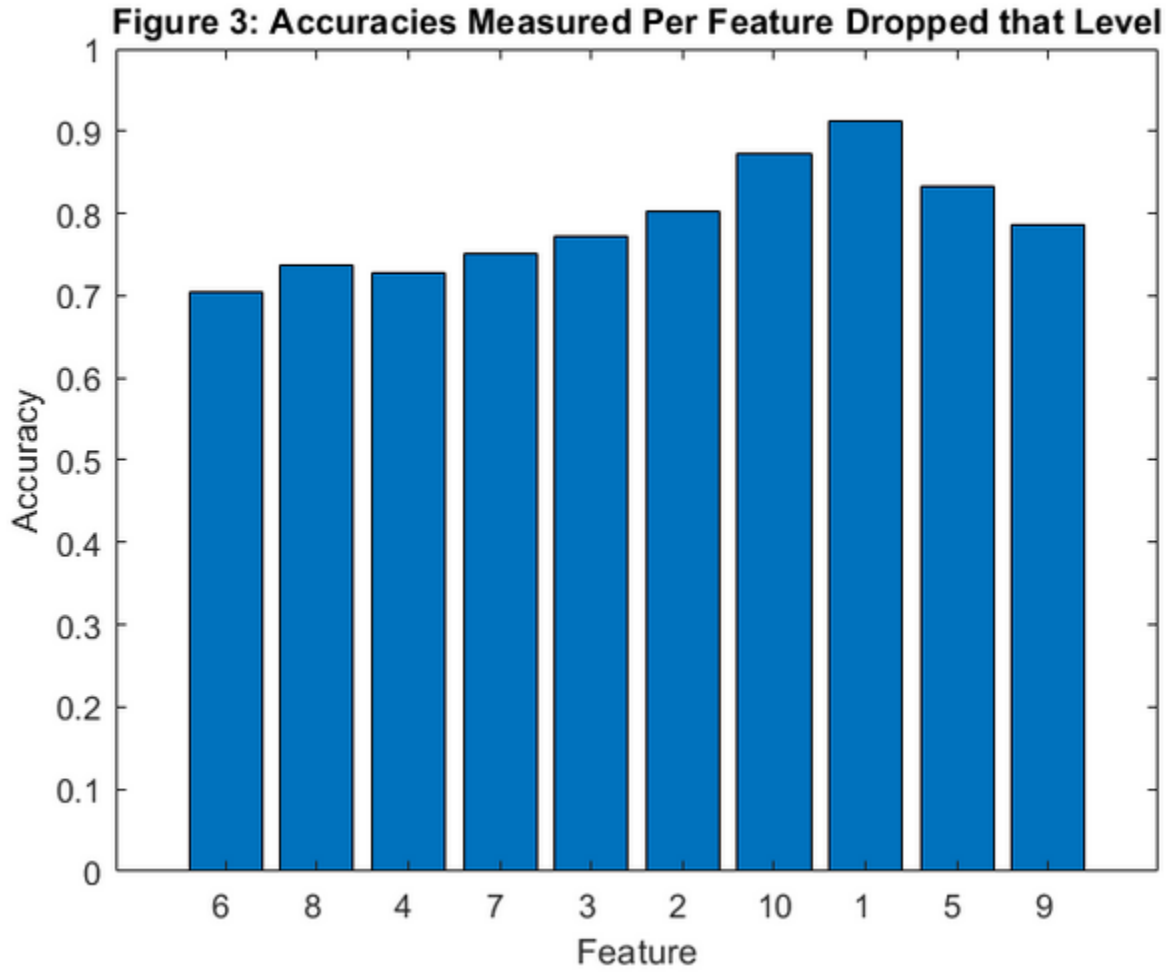


Figure 3 shows the best accuracies of each level of the backwards deletion. Notice how the last two deleted features are the first two added in forward selection.



The following are the produced results of both algorithms.

```
Best Features found by Forward Selection: 9  5
Performance: 0.91333
Best Features found by Backwards Elimination: 5  9
Performance: 0.91333
Best Features found Overall: 9  5
Best Performance Overall: 0.91333
```

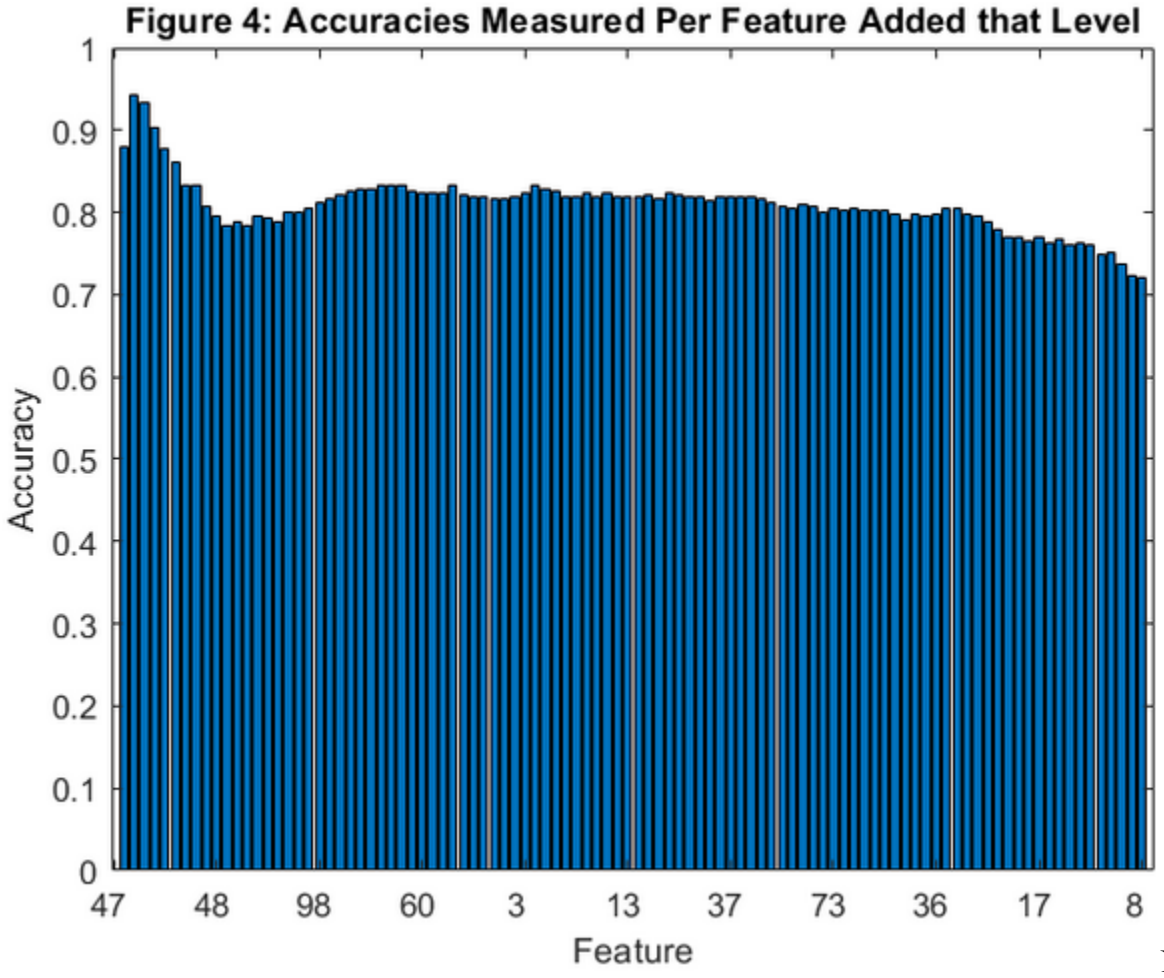
Example of Algorithms on Large Data Set

The following will be snippets of my code's generated output for a small dataset. This dataset is the same as contained in the provided file, "CS170_largetestdata__2.txt".

The following generated output is snipped because the Matlab editor constrains the amount of output that can be printed at once. It will show the end of the Forward Selection, and the beginning of Backwards Elimination, as well as their turnaround times.

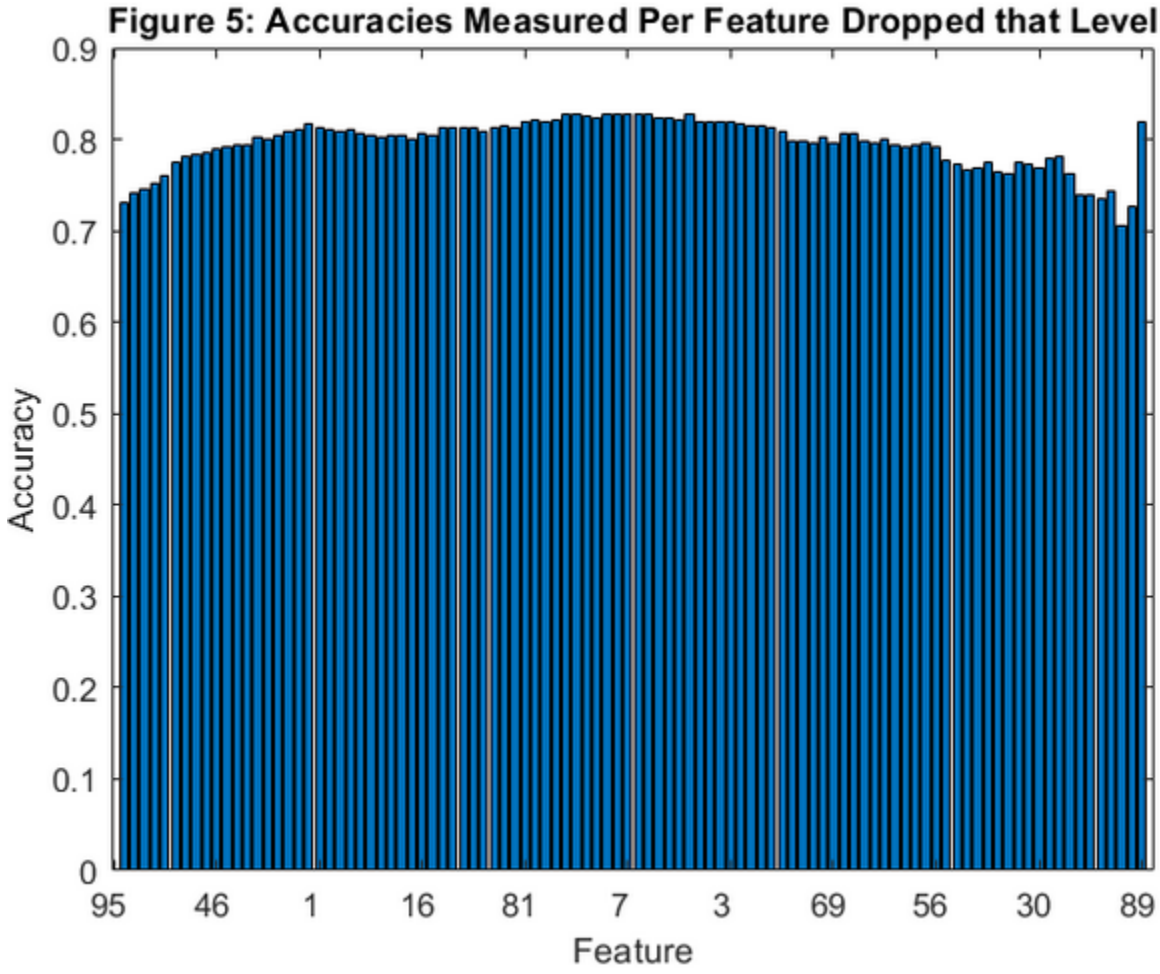
```
On level 98 of search tree
--Considering adding feature 25
----Accuracy: 0.734
--Considering adding feature 75
----Accuracy: 0.736
--Considering adding feature 81
----Accuracy: 0.726
On level 98, added feature 75.
On level 99 of search tree
--Considering adding feature 25
----Accuracy: 0.722
--Considering adding feature 81
----Accuracy: 0.708
On level 99, added feature 25.
On level 100 of search tree
--Considering adding feature 81
----Accuracy: 0.72
On level 100, added feature 81.
Elapsed time is 5681.857275 seconds.
> Starting backward search <
On level 1 of search tree
--Considering dropping feature 1
----Accuracy: 0.714
--Considering dropping feature 2
----Accuracy: 0.704
--Considering dropping feature 3
----Accuracy: 0.716
--Considering dropping feature 4
----Accuracy: 0.698
--Considering dropping feature 5
----Accuracy: 0.704
--Considering dropping feature 6
----Accuracy: 0.708
--Considering dropping feature 7
----Accuracy: 0.712
--Considering dropping feature 8
----Accuracy: 0.716
--Considering dropping feature 9
Elapsed time is 5597.386119 seconds.
```

Notice how in Figure 4, the most prominent features are identified immediately.



N

Notice how in Figure 5, the maximum value is not the peak on the right, but rather one of the values on top of the middle hill.



The results are expanded on below. This output snippet shows the final result of our feature selection algorithm. Note that the second list of features should be from Backwards Selection, I apologize for the typo.

```
Best Features found by Forward Selection: 47  48
Performance: 0.942
Best Features found by Forward Selection: 12  31  18  94  55  97  34   4  61  38  58  29   8  14  32  65  57  75

Performance: 0.828


Best Features found Overall: 47  48
Best Performance Overall: 0.942
```

Comparison of Results

On the small dataset, both algorithms come to the same result. This isn't unexpected. When certain features are incredibly strong, they will be picked up by either algorithm.

On the large dataset, both algorithms come up with completely different results. The results of this dataset showcase the differences between the two algorithms regardless of their similarities. The forwards selection produces a result that returns a much better accuracy, and is much more realistic with only 2 features than the backward elimination's 58 features that make up its best feature combination.

The runtimes for each algorithm on each dataset is recorded below.

Turnaround time:	Forward Selection	Backwards Elimination
Small dataset	19.54 sec	25.22 sec
Large dataset	94.7 min	93.283 min

Code

All the code can be found on Github:
https://github.com/IvanLorna/FeatureSelectionWithNN

Note that the .mlx and .m files are identical. I work on the live script file, but save it as a regular script file as well, so that it can be viewed in the github file viewer.

Conclusion

This report details my findings and observations on the two search algorithms that solve the same problem. Their results may differ but may also be the same. This is because there is a semantic difference between adding the best scoring to the set of considered features, and removing the feature that returns the best scores after elimination.