Anika Sood
SID#: 862283943
Email: asood008@ucr.edu
June 7th, 2025

Project 2 for CS 205 Spring 2025, with Dr. Eamonn Keogh.

My Code:
https://github.com/anikkasood/cs205-project2

All code is original except:

1. I structured the code similar to a project I did in CS 170, where I worked in a team of 3. I reused some functions for normalization/reading data/evaluating. Here is a reference to the code for that project: https://github.com/anikkasood/CS170-Project2

Part 1 Description:
The datasets provided are ASCII text w/ IEEE standard for 8 floating point numbers. To load this data into my program, I use a 'DataRow' struct to store each row of my data (label and features). Column 1 is class and the rest are features (that aren't normalized) . I used the small dataset 'CS205_small_Data__25.txt' and large dataset 'CS205_larger_Data_15.txt'. I implement the nearest neighbor classifier within 2 search algorithms, forward selection and backward elimination.

Forward Selection and Backward Elimination Background:
Forward selection begins with an empty feature set, and then adds features to the set that will improve the accuracy of the search. On the other hand, Backwards Elimination will begin with a full set of features and then remove one at a time until the feature set is as accurate as possible.

As Himanshi Singh [1] explains, to perform forward selection, the following steps are taken:

"1. Train n model using feature (n) individually and check the performance.
2. Choose the variable which gives the best performance.
3. Repeat the process and add one variable at a time.

4. Variable Producing the highest improvement is retained.
5. Repeat the entire process until there is no significant improvement in the model's performance."

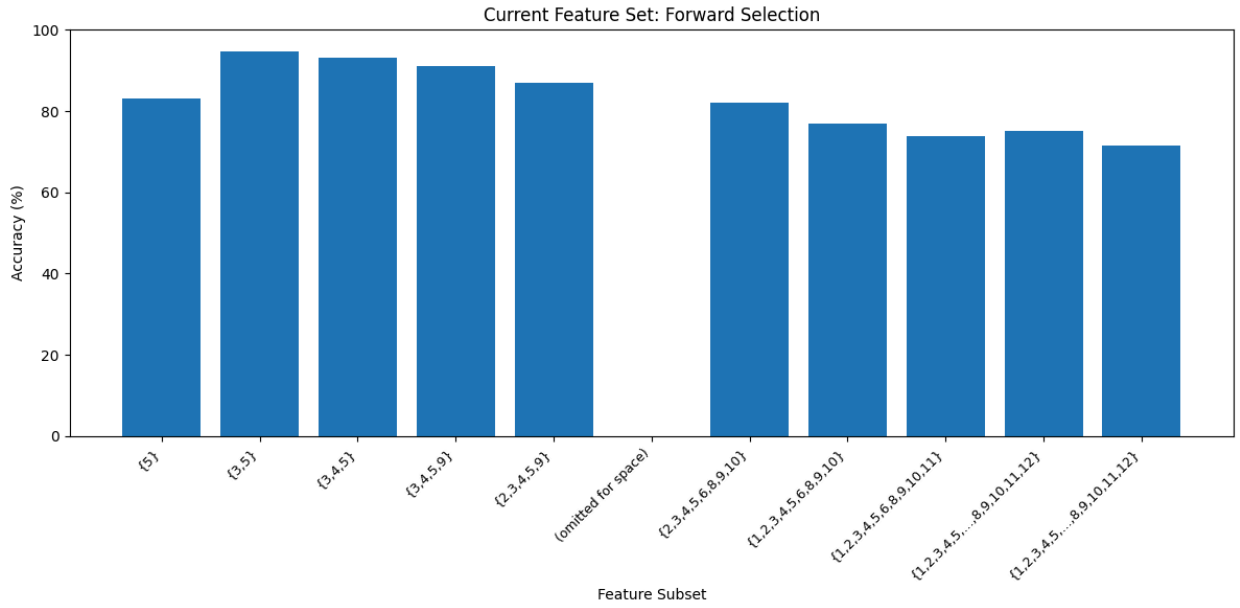
Similarly, backwards elimination does a similar process but instead of beginning with an empty set and adding features that work well, it begins with a full set of all features and then removes the ones that perform the worst.

Part 1 Implementation:

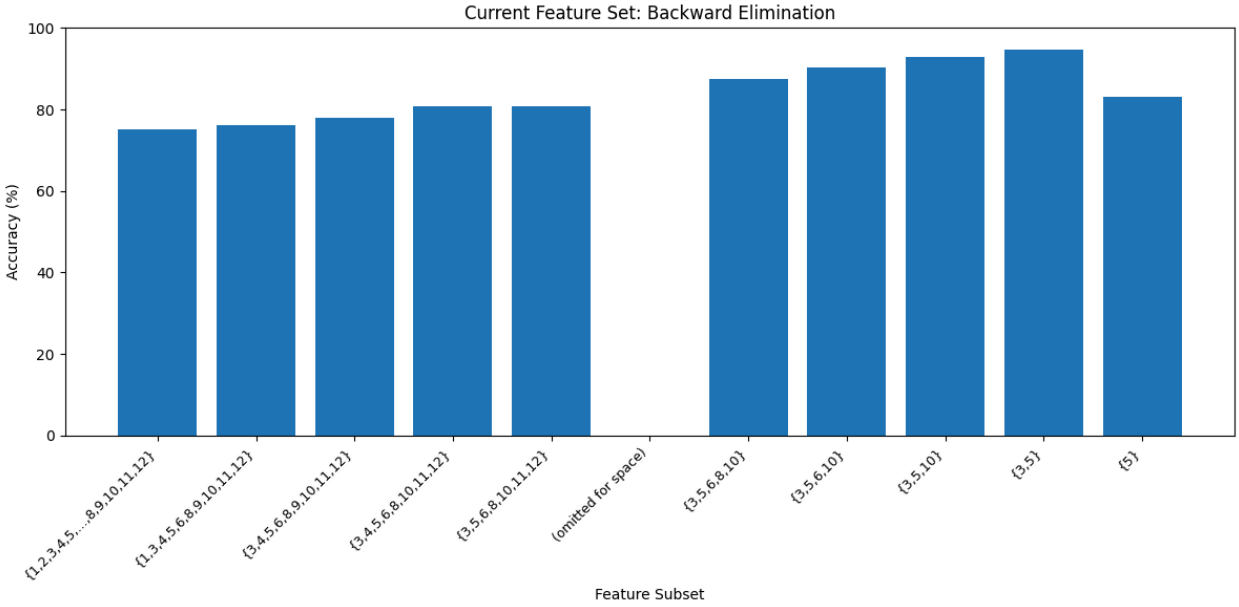
For the implementation of this part of the project, I created 3 classes: Data, Search, and Classifier. In data, I read in the data from .txt files and normalized it with helper functions. Search contained implementations of the forward selection and backward elimination algorithms. In the Classifier class, I implement the nearest neighbor algorithm and perform training/testing.

Part 1 Results Small Dataset:

Forward Selection: In figure 1, we can see the result of running forward selection on CS205_small_Data__25.txt. It begins with an empty set {} (this is not plotted below), and adds each feature in one at a time. We can see how the accuracy increases with the addition of features.



Backward Elimination: In figure 2, we can see the result of running forward selection on CS205_small_Data__25.txt. It begins with a set of all features and removes irrelevant features. We can see how the accuracy increases with the removal of features.



Part 1 Small Dataset Conclusion

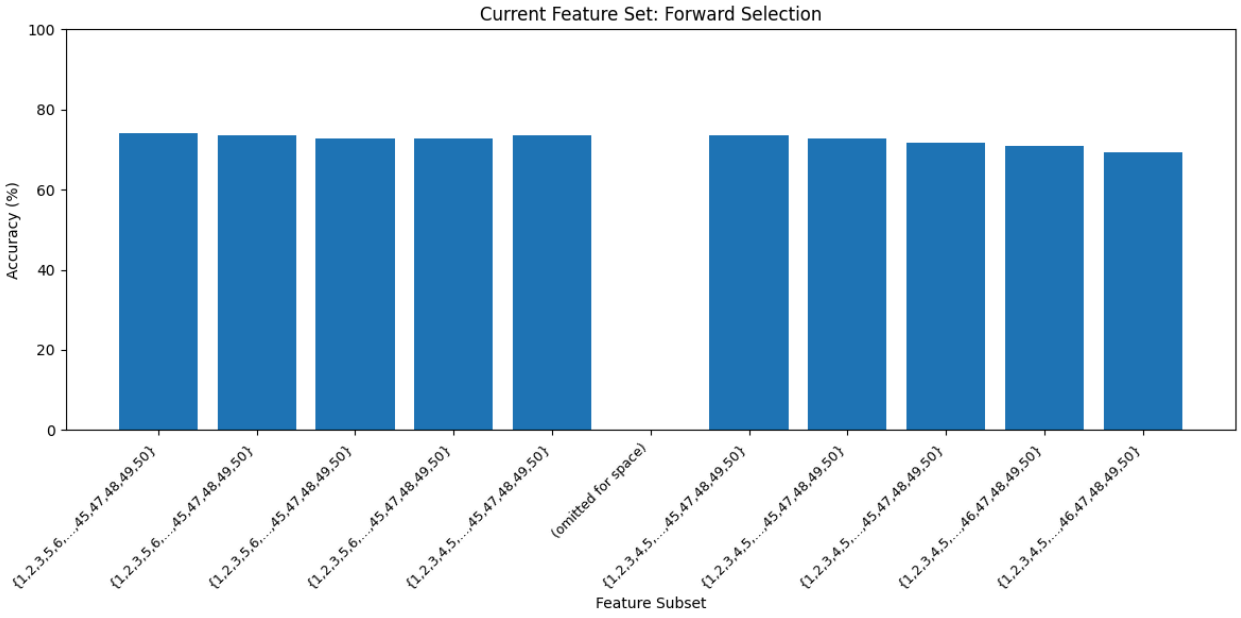
The results of this dataset pass the sanity check: both algorithms produce 71.6% accuracy for the full dataset. This was a relatively small dataset, with 12 features and 500 instances. Forward selection resulted in the best feature subset of {3,5}, which has an accuracy of 94.6%, and backward elimination had the same best result of {3,5} and 94.6%. Below is the computational effort of this dataset. The similarity in results suggest that, due to the small search space, both likely evaluate similar subsets, which allows both algorithms to converge on the same optimal subset. Additionally, this suggests that the data could have less noise/less complex feature interactions. Below is the computational effort for both algorithms on this dataset:

	Small dataset 25 (12 features and 500 instances)
Forward Selection	15495168 μs = **0.26 minutes**

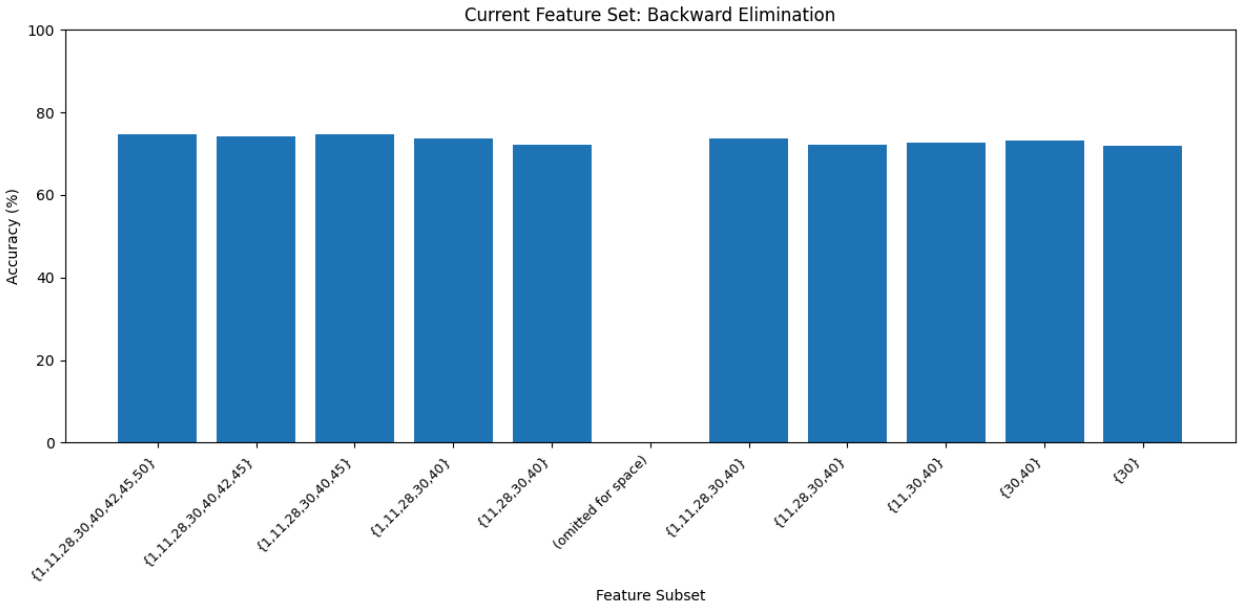
Backward Elimination	17602184 μs = **0.29 minutes**

Part 1 Results Large Dataset:

Forward Selection: In figure 1, we can see the result of running forward selection on CS205_large_Data__15.txt. It begins with an empty set {} (this is not plotted below), and adds each feature in one at a time. We can see how the accuracy increases with the addition of features.



Backward Elimination: In figure 2, we can see the result of running forward selection on CS205_large_Data__15.txt. It begins with a set of all features and removes irrelevant features. We can see how the accuracy slightly increases with the removal of features.



Part 1 Large Dataset Conclusion

The results of this dataset pass the sanity check: both algorithms produce 69.2% accuracy for the full dataset. This was a relatively large dataset, with 50 features and 1000 instances. Forward selection resulted in the best feature subset of {7,26}, which has an accuracy of 96.4%. Backward elimination's best subset was {1,2,5,7,8,9,10,11,13,14,17,19,23,25,27,28,30,31,33,34,35,37,38,39,40,41,42,43,45,46,47,50}, which has an accuracy of 79.2%. As we can see, forward selection chose a smaller number of features and had a higher accuracy compared to backward elimination. This shows how the greedy strategy used in forward selection, where the best performing individual feature is chosen first, can have a large impact on the chosen set and overall performance of the model. This data shows how forward selection might scale better to data with higher dimensions.

	Large Dataset 15 (50 features & 1000 instances)
Forward Selection	1400161618 µs = **23.3 minutes**
Backward	1854464469 = **30**

Elimination	**minutes**

*Traces for part 1 included at the bottom of this report, for readability.

Part 2 Implementation:

For part 2, I selected the Breast Cancer Wisconsin Dataset. [2]. The first column is an ID number, so I have removed it because it has no meaning. The labels are ‘M’ for malignant and ‘B’ for benign. For my purposes, I replaced M with 1, and B with 0. The rest of the features in this dataset are the following, as explained in [2]:

a) radius (mean of distances from center to points on the perimeter)

b) texture (standard deviation of gray-scale values)

c) perimeter

d) area

e) smoothness (local variation in radius lengths)

f) compactness (perimeter^2 / area - 1.0)

g) concavity (severity of concave portions of the contour)

h) concave points (number of concave portions of the contour)

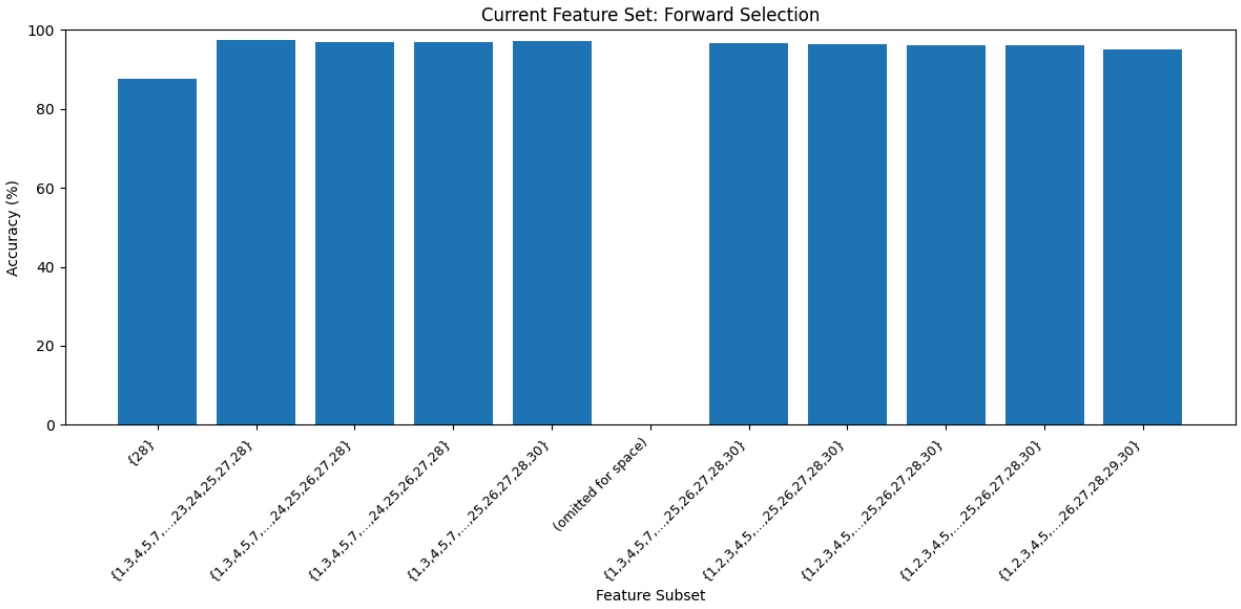
i) symmetry

j) fractal dimension ("coastline approximation" - 1)

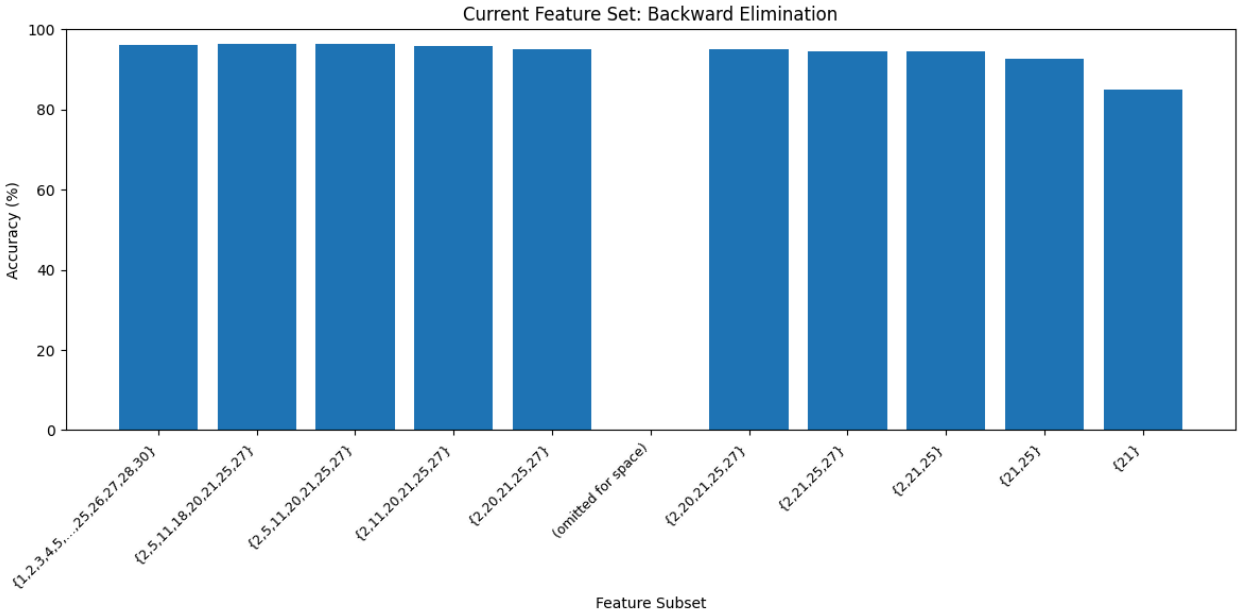
I normalized this data and then ran forward selection and backwards elimination on these, similar to part 1 of this project.

Part 2 Results:

Forward Selection: In figure 1, we can see the result of running forward selection on breast_cancer_wisconsin.txt. It begins with an empty set {} (this is not plotted below), and adds each feature in one at a time. We can see how the accuracy increases with the addition of features.



Backward Elimination: In figure 2, we can see the result of running forward selection on breast_cancer_wisconsin.txt. It begins with a set of all features and removes irrelevant features. We can see how the accuracy slightly increases with the removal of features.



Part 2 Conclusion:

The results of this dataset pass the sanity check: both algorithms produce 95.1% accuracy for the full dataset. This was a relatively medium-sized dataset, with 30 features and 569 instances. Forward selection resulted in the best feature subset of {7,8,14,16,18,20,21,22,23,24,25,28}, which has an accuracy of 98.1%. Backward elimination's best subset was

{1,2,3,4,5,7,8,11,12,13,14,16,17,18,19,20,21,24,25,26,27,28,30}, which has an accuracy of 97.5%. As we can see, forward selection chose a smaller number of features and had a higher accuracy compared to backward elimination. This is similar to the results of the large dataset and shows how forward selection might scale better to data with higher dimensions. The lower accuracy of the larger subsets (in this example and also with the large dataset) could be due to the curse of dimensionality, which means that too many features can result in noise or overfitting in the model. Below are the computational results of this dataset.

	Breast Cancer Wisconsin Dataset (30 features & 569 instances)
Forward Selection	150954441 μs = **2.5 minutes**
Backward Elimination	182838014 μs = **3 minutes**

Why First Features Added are The Best Features:

When beginning a search with an empty set of features, the first features that are added are the best. With forward selection, at each step a single feature is added that will improve the model's accuracy. The first feature selected has the highest accuracy on its own. As seen in the traces for all datasets, the first step of the search is to see the accuracy of each feature individually. The highest accuracy is added to the solution set. This is the greedy choice made by the forward selection algorithm. The features chosen after this initial selection will be evaluated on how well they perform with the first selection of features. Hence, the features selected at the start have the most positive impact on the overall accuracy of all selected features.

*Traces for part 2 included at the bottom of this report, for readability.

Computational Efforts - Part 1 and 2

I ran all of these searches in C++ on a laptop with 1.4 GHz Quad Core Intel Core i% and 8 GB of main memory.

Computation Time For Each Dataset (features/attributes) using Forward Selection and Backward Elimination

	Small dataset 25 (12 features and 500 instances)	**Large Dataset 15** (50 features & 1000 instances)	**Breast Cancer Wisconsin Dataset** (30 features & 569 instances)
Forward Selection	15495168 μs = **0.26 minutes**	1400161618 μs = **23.3 minutes**	150954441 μs = **2.5 minutes**
Backward Elimination	17602184 μs = **0.29 minutes**	1854464469 = **30 minutes**	182838014 μs = **3 minutes**

In the table above, we can see the computational efforts needed for forward selection and backward elimination on all three datasets explored in this project. The small dataset had the smallest amount of features and instances (12 and 500), and we can see that the computation time was the least for both methods out of all three. Forward selection (0.26 minutes) was a bit quicker than backward elimination (0.29 minutes). For the large dataset assigned, there were 50 features and 1000 instances. We can see the same pattern: forward selection was notably quicker than backward elimination (23.3 minutes vs 30 minutes). For the last dataset I chose, the same is true: forward selection is 2.5 minutes and backward selection is 3 minutes. The larger datasets took more computation time and it appears that for these cases backward elimination took more time than forward selection. This intuitively makes sense because backward selection begins with a set of all features at the start, so at the beginning each step is slower (compared to forward selection) because it's evaluating more data. The relationship becomes more noticeable with more data.

Traces:

1. Small Dataset - 25

Below I show a single trace of my algorithm. I am only showing Forward Selection on the small dataset

Welcome to the Anika Sood Feature Selection Algorithm.

Type in the name of the file to test :

CS205_small_Data__25.txt

Type the number of the algorithm you want to run.

(1) Forward Selection

(2) Backward Elimination

1

This dataset has 12 features (not including class attribute), with 500 instances.

Forward Selection:

Running nearest neighbor with all 4 features, using “leaving-one-out” evaluation, I get an accuracy of 82.4%

Beginning search.

Using feature(s) {1} accuracy is 70.8%

Using feature(s) {2} accuracy is 76.2%

Using feature(s) {3} accuracy is 74.4%

Using feature(s) {4} accuracy is 70.0%

Using feature(s) {5} accuracy is 83.0%

Using feature(s) {6} accuracy is 68.2%

Using feature(s) {7} accuracy is 73.2%

Using feature(s) {8} accuracy is 73.2%

Using feature(s) {9} accuracy is 70.4%

Using feature(s) {10} accuracy is 74.4%

Using feature(s) {11} accuracy is 71.2%

Using feature(s) {12} accuracy is 69.4%
Feature set {5} was best, accuracy is 83.0%

Using feature(s) {1,5} accuracy is 83.6%
Using feature(s) {2,5} accuracy is 81.6%
Using feature(s) {3,5} accuracy is 94.6%
Using feature(s) {4,5} accuracy is 82.0%
Using feature(s) {5,6} accuracy is 81.0%
Using feature(s) {5,7} accuracy is 85.0%
Using feature(s) {5,8} accuracy is 80.6%
Using feature(s) {5,9} accuracy is 84.4%
Using feature(s) {5,10} accuracy is 85.2%
Using feature(s) {5,11} accuracy is 84.2%
Using feature(s) {5,12} accuracy is 81.2%
Feature set {3,5} was best, accuracy is 94.6%

Using feature(s) {1,3,5} accuracy is 91.0%
Using feature(s) {2,3,5} accuracy is 90.8%
Using feature(s) {3,4,5} accuracy is 93.2%
Using feature(s) {3,5,6} accuracy is 92.0%
Using feature(s) {3,5,7} accuracy is 91.6%
Using feature(s) {3,5,8} accuracy is 90.6%
Using feature(s) {3,5,9} accuracy is 93.2%
Using feature(s) {3,5,10} accuracy is 92.8%
Using feature(s) {3,5,11} accuracy is 89.6%
Using feature(s) {3,5,12} accuracy is 91.4%
Feature set {3,4,5} was best, accuracy is 93.2%

Using feature(s) {1,3,4,5} accuracy is 87.8%
Using feature(s) {2,3,4,5} accuracy is 89.8%
Using feature(s) {3,4,5,6} accuracy is 87.2%

Using feature(s) {3,4,5,7} accuracy is 90.2%
Using feature(s) {3,4,5,8} accuracy is 88.0%
Using feature(s) {3,4,5,9} accuracy is 91.2%
Using feature(s) {3,4,5,10} accuracy is 89.2%
Using feature(s) {3,4,5,11} accuracy is 88.0%
Using feature(s) {3,4,5,12} accuracy is 90.4%
Feature set {3,4,5,9} was best, accuracy is 91.2%

Using feature(s) {1,3,4,5,9} accuracy is 83.4%
Using feature(s) {2,3,4,5,9} accuracy is 87.0%
Using feature(s) {3,4,5,6,9} accuracy is 85.6%
Using feature(s) {3,4,5,7,9} accuracy is 86.0%
Using feature(s) {3,4,5,8,9} accuracy is 86.4%
Using feature(s) {3,4,5,9,10} accuracy is 85.8%
Using feature(s) {3,4,5,9,11} accuracy is 84.2%
Using feature(s) {3,4,5,9,12} accuracy is 84.4%
Feature set {2,3,4,5,9} was best, accuracy is 87.0%

Using feature(s) {1,2,3,4,5,9} accuracy is 82.0%
Using feature(s) {2,3,4,5,6,9} accuracy is 85.0%
Using feature(s) {2,3,4,5,7,9} accuracy is 81.4%
Using feature(s) {2,3,4,5,8,9} accuracy is 83.2%
Using feature(s) {2,3,4,5,9,10} accuracy is 82.2%
Using feature(s) {2,3,4,5,9,11} accuracy is 80.6%
Using feature(s) {2,3,4,5,9,12} accuracy is 82.8%
Feature set {2,3,4,5,6,9} was best, accuracy is 85.0%

Using feature(s) {1,2,3,4,5,6,9} accuracy is 82.6%
Using feature(s) {2,3,4,5,6,7,9} accuracy is 79.2%
Using feature(s) {2,3,4,5,6,8,9} accuracy is 84.2%
Using feature(s) {2,3,4,5,6,9,10} accuracy is 82.8%

Using feature(s) {2,3,4,5,6,9,11} accuracy is 79.4%
Using feature(s) {2,3,4,5,6,9,12} accuracy is 79.2%
Feature set {2,3,4,5,6,8,9} was best, accuracy is 84.2%

Using feature(s) {1,2,3,4,5,6,8,9} accuracy is 80.0%
Using feature(s) {2,3,4,5,6,7,8,9} accuracy is 79.8%
Using feature(s) {2,3,4,5,6,8,9,10} accuracy is 82.0%
Using feature(s) {2,3,4,5,6,8,9,11} accuracy is 81.4%
Using feature(s) {2,3,4,5,6,8,9,12} accuracy is 78.8%
Feature set {2,3,4,5,6,8,9,10} was best, accuracy is 82.0%

Using feature(s) {1,2,3,4,5,6,8,9,10} accuracy is 77.0%
Using feature(s) {2,3,4,5,6,7,8,9,10} accuracy is 75.4%
Using feature(s) {2,3,4,5,6,8,9,10,11} accuracy is 76.2%
Using feature(s) {2,3,4,5,6,8,9,10,12} accuracy is 75.2%
Feature set {1,2,3,4,5,6,8,9,10} was best, accuracy is 77.0%

Using feature(s) {1,2,3,4,5,6,7,8,9,10} accuracy is 73.2%
Using feature(s) {1,2,3,4,5,6,8,9,10,11} accuracy is 73.8%
Using feature(s) {1,2,3,4,5,6,8,9,10,12} accuracy is 71.8%
Feature set {1,2,3,4,5,6,8,9,10,11} was best, accuracy is 73.8%

Using feature(s) {1,2,3,4,5,6,7,8,9,10,11} accuracy is 70.6%
Using feature(s) {1,2,3,4,5,6,8,9,10,11,12} accuracy is 75.0%
Feature set {1,2,3,4,5,6,8,9,10,11,12} was best, accuracy is 75.0%

Using feature(s) {1,2,3,4,5,6,7,8,9,10,11,12} accuracy is 71.6%
Feature set {1,2,3,4,5,6,7,8,9,10,11,12} was best, accuracy is 71.6%

Finished search!! The best feature subset is {3,5}, which has an accuracy of 94.6%

2. Large Dataset - 15 : Shortened Trace

Below I show a single trace of my algorithm. I am only showing Backward Elimination on the large dataset

Backward Elimination:
Using all features and 'random' evaluation, I get an accuracy of 69.2%
Beginning search.
Using feature(s)
{2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.3%
Using feature(s)
{1,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.3%
Using feature(s)
{1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 71.1%
Using feature(s)
{1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 68.8%
Using feature(s)
{1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.2%
Using feature(s)
{1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 68.4%
Using feature(s)
{1,2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 70.4%
Using feature(s)
{1,2,3,4,5,6,7,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.7%

Using feature(s)
{1,2,3,4,5,6,7,8,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.1%
Using feature(s)
{1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 67.2%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.2%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.5%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 70.1%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.7%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.6%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 70.0%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 70.2%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.4%

Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 71.0%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.5%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.3%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 70.9%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 70.2%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.0%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.1%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.3%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 68.5%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 68.9%

Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.3%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.0%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 70.5%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.6%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.0%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.5%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 67.7%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,37,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.3%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,38,39,40,41,42,43,44,45,46,47,48,49,50} 69.5%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,39,40,41,42,43,44,45,46,47,48,49,50} 69.8%

Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,40,41,42,43,44,45,46,47,48,49,50} 68.7%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,41,42,43,44,45,46,47,48,49,50} 68.4%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,42,43,44,45,46,47,48,49,50} 68.3%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,41,43,44,45,46,47,48,49,50} 70.0%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,41,42,44,45,46,47,48,49,50} 69.9%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,41,42,43,45,46,47,48,49,50} 68.7%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,41,42,43,44,46,47,48,49,50} 69.3%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,41,42,43,44,45,47,48,49,50} 68.7%
Using feature(s)
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
33,34,35,36,37,38,39,40,41,42,43,44,45,46,48,49,50} 69.0%
...
... [omitted 1227 lines] ...
...
Using feature(s) {1,11,28,40,42,45,47,50} 72.7%

Using feature(s) {1,11,28,30,42,45,47,50} 74.1%

Using feature(s) {1,11,28,30,40,45,47,50} 74.0%

Using feature(s) {1,11,28,30,40,42,47,50} 74.0%

Using feature(s) {1,11,28,30,40,42,45,50} 74.6%

Using feature(s) {1,11,28,30,40,42,45,47} 74.5%

Feature set {1,11,28,30,40,42,45,50} was best, accuracy is 74.6%

Using feature(s) {11,28,30,40,42,45,50} 71.3%

Using feature(s) {1,28,30,40,42,45,50} 71.9%

Using feature(s) {1,11,30,40,42,45,50} 73.4%

Using feature(s) {1,11,28,40,42,45,50} 71.5%

Using feature(s) {1,11,28,30,42,45,50} 73.3%

Using feature(s) {1,11,28,30,40,45,50} 73.7%

Using feature(s) {1,11,28,30,40,42,50} 73.2%

Using feature(s) {1,11,28,30,40,42,45} 74.3%

Feature set {1,11,28,30,40,42,45} was best, accuracy is 74.3%

Using feature(s) {11,28,30,40,42,45} 72.9%

Using feature(s) {1,28,30,40,42,45} 70.8%

Using feature(s) {1,11,30,40,42,45} 72.6%

Using feature(s) {1,11,28,40,42,45} 70.7%

Using feature(s) {1,11,28,30,42,45} 73.3%

Using feature(s) {1,11,28,30,40,45} 74.7%

Using feature(s) {1,11,28,30,40,42} 72.5%

Feature set {1,11,28,30,40,45} was best, accuracy is 74.7%

Using feature(s) {11,28,30,40,45} 73.4%

Using feature(s) {1,28,30,40,45} 70.9%

Using feature(s) {1,11,30,40,45} 72.4%

Using feature(s) {1,11,28,40,45} 72.7%

Using feature(s) {1,11,28,30,45} 73.4%

Using feature(s) {1,11,28,30,40} 73.7%

Feature set {1,11,28,30,40} was best, accuracy is 73.7%

Using feature(s) {11,28,30,40} 72.2%

Using feature(s) {1,28,30,40} 70.9%

Using feature(s) {1,11,30,40} 70.2%

Using feature(s) {1,11,28,40} 70.6%

Using feature(s) {1,11,28,30} 71.9%

Feature set {11,28,30,40} was best, accuracy is 72.2%

Using feature(s) {28,30,40} 72.5%

Using feature(s) {11,30,40} 72.6%

Using feature(s) {11,28,40} 70.7%

Using feature(s) {11,28,30} 71.4%

Feature set {11,30,40} was best, accuracy is 72.6%

Using feature(s) {30,40} 73.2%

Using feature(s) {11,40} 71.4%

Using feature(s) {11,30} 72.5%

Feature set {30,40} was best, accuracy is 73.2%

Using feature(s) {40} 71.1%

Using feature(s) {30} 71.8%

Feature set {30} was best, accuracy is 71.8%

Finished search!! The best feature subset is {1,2,5,7,8,9,10,11,13,14,17,19,23,25,27,28,30,31,33,34,35,37,38,39,40,41,42,43,45,46,47,50}, which has an accuracy of 79.2%

3. Chosen Dataset - Breast Cancer Wisconsin : Shortened Trace

Below I show a single trace of my algorithm. I am only showing Backward Elimination on the large dataset

Forward Selection:

Running nearest neighbor with all 4 features, using “leaving-one-out” evaluation, I get an accuracy of 37.3%

Beginning search.
Using feature(s){1} 80.8
Using feature(s){2} 62.0
Using feature(s){3} 82.6
Using feature(s){4} 81.4
Using feature(s){5} 61.9
Using feature(s){6} 73.6
Using feature(s){7} 81.4
Using feature(s){8} 84.0
Using feature(s){9} 58.3
Using feature(s){10} 56.2
Using feature(s){11} 75.0
Using feature(s){12} 54.0
Using feature(s){13} 72.9
Using feature(s){14} 78.6
Using feature(s){15} 52.0
Using feature(s){16} 59.4
Using feature(s){17} 63.8
Using feature(s){18} 64.9
Using feature(s){19} 51.7
Using feature(s){20} 54.5
Using feature(s){21} 85.1
Using feature(s){22} 63.6
Using feature(s){23} 87.0
Using feature(s){24} 87.5
Using feature(s){25} 60.8
Using feature(s){26} 72.4
Using feature(s){27} 79.3
Using feature(s){28} 87.7
Using feature(s){29} 59.6
Using feature(s){30} 64.7

Feature set {28} was best, accuracy is 87.7%

Using feature(s){1,28} 92.3

Using feature(s){2,28} 90.0

Using feature(s){3,28} 90.2

Using feature(s){4,28} 91.2

Using feature(s){5,28} 88.6

Using feature(s){6,28} 86.1

Using feature(s){7,28} 86.1

Using feature(s){8,28} 86.6

Using feature(s){9,28} 85.2

Using feature(s){10,28} 89.8

Using feature(s){11,28} 88.0

Using feature(s){12,28} 86.5

Using feature(s){13,28} 85.6

Using feature(s){14,28} 93.1

Using feature(s){15,28} 87.5

Using feature(s){16,28} 86.5

...

... [omitted 399 lines] ...

...

Using feature(s){1,3,5,7,8,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28} 97.0

Using feature(s){1,3,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 97.2

Using feature(s){1,3,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28,29} 96.3

Using feature(s){1,3,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28,30} 97.4

Feature set {1,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28} was best, accuracy is 97.4%

Using feature(s){1,2,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28} 96.8

Using feature(s){1,3,4,5,6,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28} 96.8

Using feature(s){1,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28} 96.8

Using feature(s){1,3,4,5,7,8,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28} 96.7

Using feature(s){1,3,4,5,7,8,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28} 96.7

Using feature(s){1,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 97.0
Using feature(s){1,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28,29} 96.3
Using feature(s){1,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,27,28,30} 97.0
Feature set {1,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} was best, accuracy is 97.0%
Using feature(s){1,2,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 96.7
Using feature(s){1,3,4,5,6,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 96.7
Using feature(s){1,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 97.0
Using feature(s){1,3,4,5,7,8,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 96.3
Using feature(s){1,3,4,5,7,8,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 97.0
Using feature(s){1,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29} 96.3
Using feature(s){1,3,4,5,7,8,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.7
Feature set {1,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} was best, accuracy is 97.0%
Using feature(s){1,2,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 96.5
Using feature(s){1,3,4,5,6,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 96.5
Using feature(s){1,3,4,5,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 96.3
Using feature(s){1,3,4,5,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28} 96.8
Using feature(s){1,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29} 96.1
Using feature(s){1,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 97.2

Feature set {1,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} was best, accuracy is 97.2%

Using feature(s){1,2,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.5

Using feature(s){1,3,4,5,6,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.1

Using feature(s){1,3,4,5,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.3

Using feature(s){1,3,4,5,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.7

Using feature(s){1,3,4,5,7,8,9,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} 96.1

Feature set {1,3,4,5,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} was best, accuracy is 96.7%

Using feature(s){1,2,3,4,5,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.5

Using feature(s){1,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.3

Using feature(s){1,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 95.8

Using feature(s){1,3,4,5,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} 95.8

Feature set {1,2,3,4,5,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} was best, accuracy is 96.5%

Using feature(s){1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.1

Using feature(s){1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.0

Using feature(s){1,2,3,4,5,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} 95.6

Feature set {1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} was best, accuracy is 96.1%

Using feature(s){1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} 96.0

Using feature(s){1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} 95.4

Feature set {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30} was best, accuracy is 96.0%

Using feature(s){1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} 95.1

Feature set {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30} was best, accuracy is 95.1%

Finished search!! The best feature subset is {7,8,14,16,18,20,21,22,23,24,25,28}, which has an accuracy of 98.1%

Citations:

1. https://www.analyticsvidhya.com/blog/2021/04/forward-feature-selection-and-its-implementation/#h-what-is-forward-feature-selection-in-machine-learning

2. Wolberg, W., Mangasarian, O., Street, N., & Street, W. (1993). Breast Cancer Wisconsin (Diagnostic) [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.