Yuteng Wu
SID 862217627
Email ywu352@ucr.edu
Date Dec-6-2022

In completing this assignment I have consulted:

Slides from the lecture

- 6__MachineLearning001.pdf
- 7__MachineLearning001.pdf
- 8__MachineLearning001.pdf
- 9__MachineLearning001.pdf

Outline of this report:

- Cover page: (this page)
- My report: (2-4)
- Sample output and trace back: (5-6)
- URL of my code: CS170-Project2

CS170: Project 2: The Feature Selection

Yuteng Wu, SID:862217627 Dec-6-2022

Introduction

In project 2 we are tasked to implement the Nearest Neighbor Algorithm we learned from the lecture, then utilize it with two different but similar searches, Forward Selection and Backward Elimination to find out which subset (features) has the best accuracy rate out of the given data.

Searching Result Summary

	CS170_Small_Data__107.txt	**CS170_Large_Data__124.txt**
Forward Selection	8.1 second	86 minutes
Backward Elimination	9.8 second	227 minutes
Best Accuracy (Forward)	[3,1] 97%	[39,13] 96.3%
Best Accuracy (Backward)	[3,1] 97%	[39,17] 84.6%

Searching Report

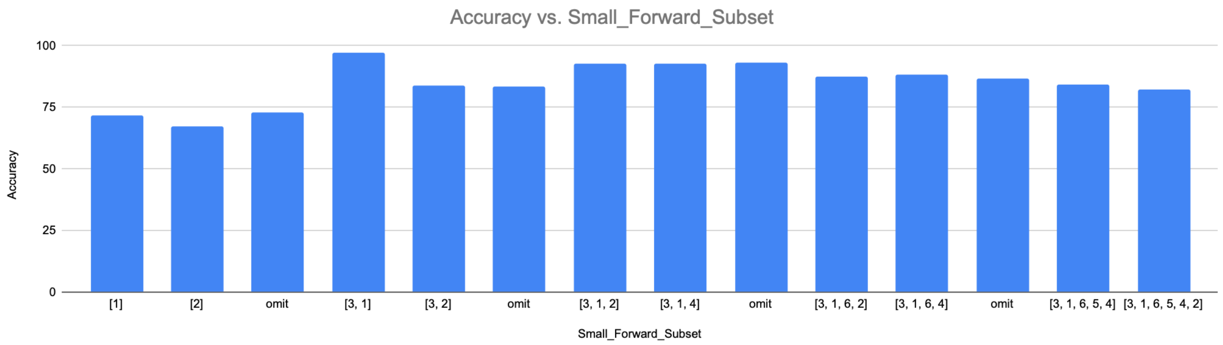


Figure 1: Accuracy vs. Small_Forward_Subset

In *Figure 1, it is the graph of accuracy vs forward selection running on **CS170_Small_Data__107.txt**. It starts from a single feature and gradually increases till all the features are included at the end. The “omit” represents the average accuracy of a few subsets due to the spacing. We can see that at first I added single features and the accuracy rate is relatively low. Then we have a sudden increase in accuracy after adding two features, which is the best accuracy with features [3,1] and a 97% accuracy rate. After that, we can see the graph went into a small dip and went back up, then gradually decrease as we adding more features.

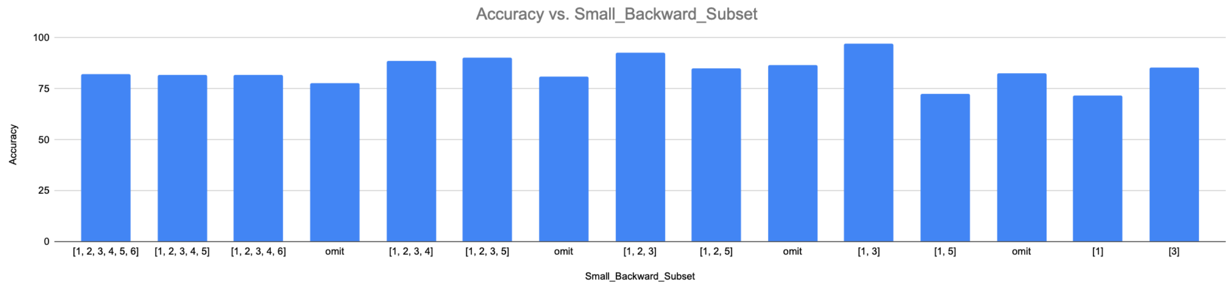


Figure 2: Accuracy vs. Small_Backward_Subset

In *Figure 2, it is the graph of accuracy vs backward elimination running on **CS170_Small_Data__107.txt**. Comparing *Figure 2 with the previous forward selection (*Figure 1), it has the same result but slightly different route of searching. For backward elimination, it starts at all features with a low accuracy rate, then you can see it is slowly increasing even though it has experienced a few dips before it reaches back to the highest accuracy with two features [1,3] and a 97% accuracy rate. After that, the accuracy rate dropped dramatically.

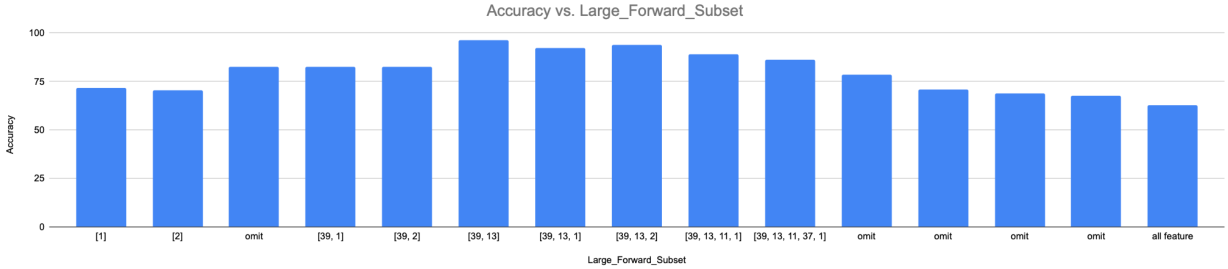


Figure 3: Accuracy vs. Large_Forward_Subset

In *Figure 3, it is the graph of accuracy vs forward elimination running on **CS170_Large_Data__124.txt**. For this file, it is significantly larger than the small data. Therefore, the trend of the graph is more clear and easier to tell. It starts at a single feature with low accuracy, generally increases till the highest point, the best accuracy subset with feature [39,13] and a 96.3% accuracy rate. After that it gradually decrease till all feature subset.

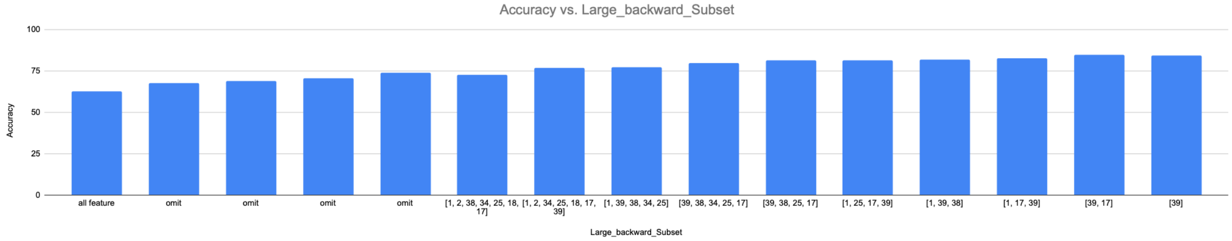


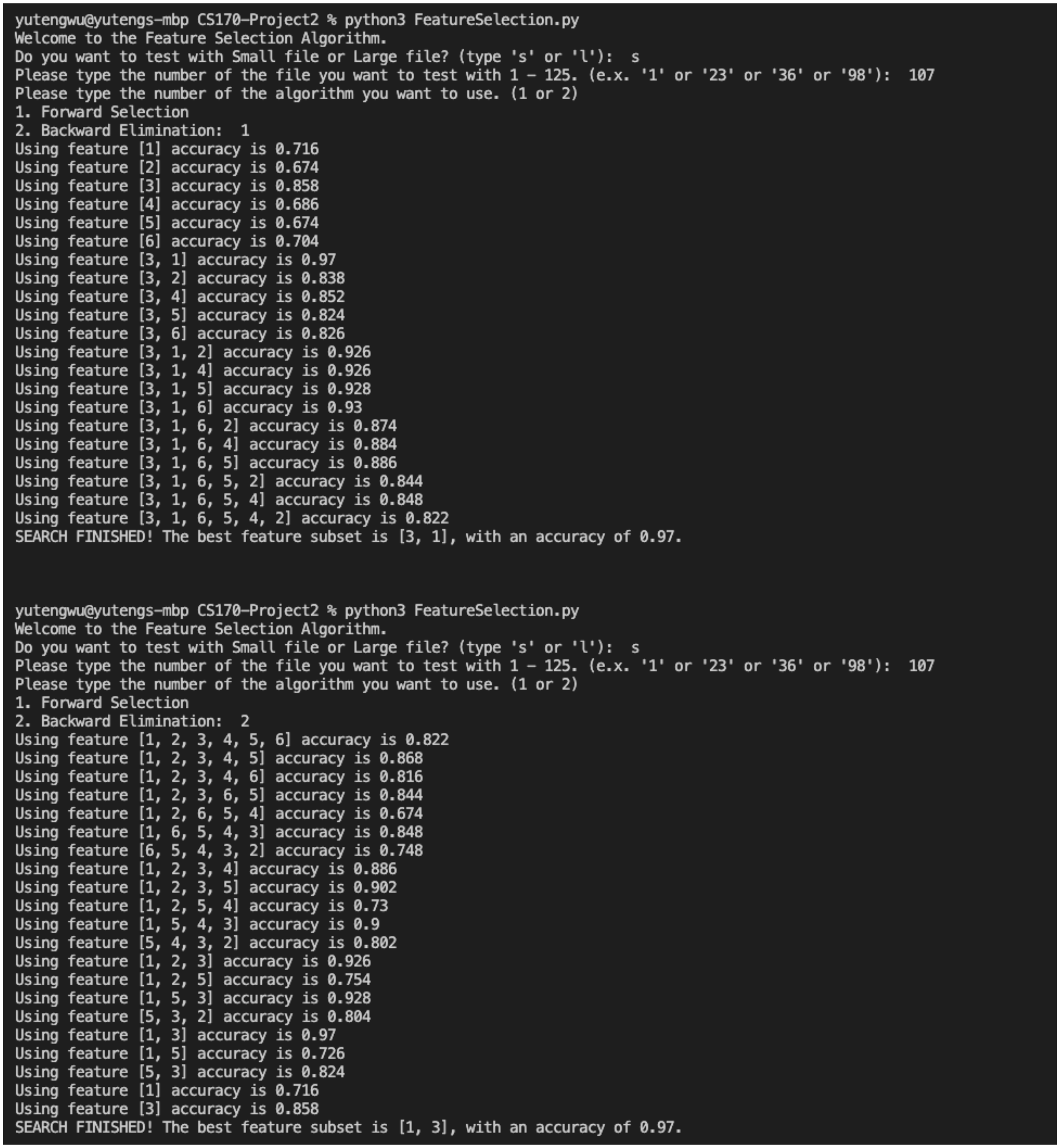
Figure 4: Accuracy vs. Large_Backward_Subset

In *Figure 4, it is the graph of accuracy vs backward elimination running on **CS170_Large_Data__124.txt**. Interestingly, the result does not match with the forward selection run on the same file. As we could see from the graph above, the accuracy rate started very low with all features, then gradually increasing like a linear line. The highest accuracy happened at the very end with two features [39,17] and a 84.6% accuracy rate.

Conclusion

Based on the result of the Feature Selection Program on these two files, I assume that the forward selection and the backward elimination does not always give out the same result despite testing on the same file. I think the result could be affected by the same accuracy rate returned from different subsets, and the forward selection and the backward elimination chose to run with different subsets, then end up with different results.

Screenshot for the Forward Selection and Backward Elimination on CS170_Small_Data__107.txt



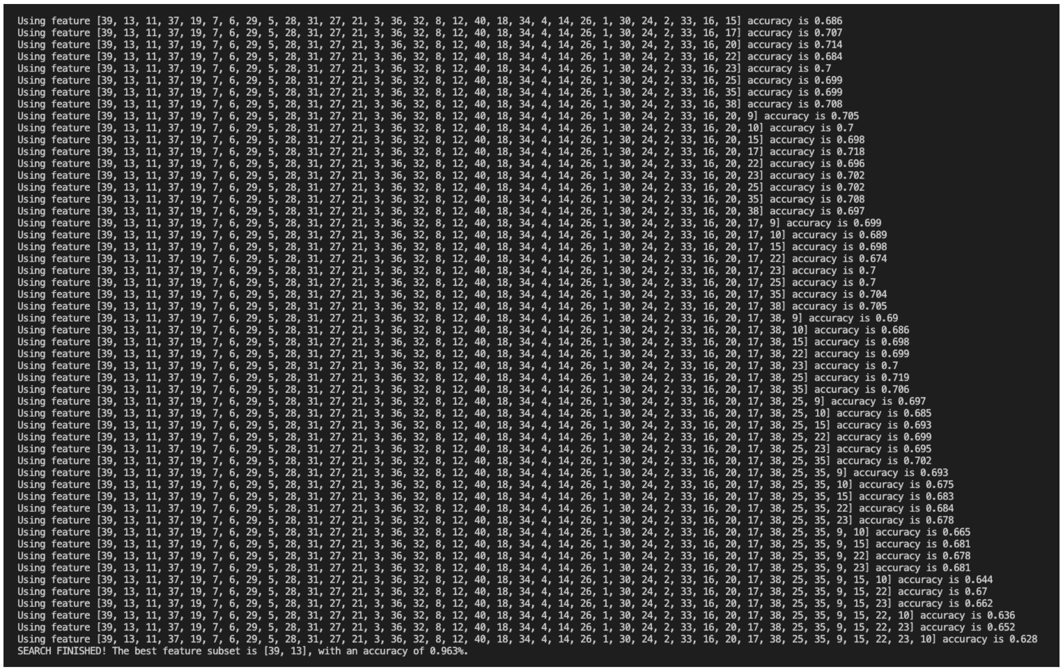
```
yutengwu@yutengs-mbp CS170-Project2 % python3 FeatureSelection.py
Welcome to the Feature Selection Algorithm.
Do you want to test with Small file or Large file? (type 's' or 'l'):  s
Please type the number of the file you want to test with 1 - 125. (e.x. '1' or '23' or '36' or '98'):  107
Please type the number of the algorithm you want to use. (1 or 2)
1. Forward Selection
2. Backward Elimination:  1
Using feature [1] accuracy is 0.716
Using feature [2] accuracy is 0.674
Using feature [3] accuracy is 0.858
Using feature [4] accuracy is 0.686
Using feature [5] accuracy is 0.674
Using feature [6] accuracy is 0.704
Using feature [3, 1] accuracy is 0.97
Using feature [3, 2] accuracy is 0.838
Using feature [3, 4] accuracy is 0.852
Using feature [3, 5] accuracy is 0.824
Using feature [3, 6] accuracy is 0.826
Using feature [3, 1, 2] accuracy is 0.926
Using feature [3, 1, 4] accuracy is 0.926
Using feature [3, 1, 5] accuracy is 0.928
Using feature [3, 1, 6] accuracy is 0.93
Using feature [3, 1, 6, 2] accuracy is 0.874
Using feature [3, 1, 6, 4] accuracy is 0.884
Using feature [3, 1, 6, 5] accuracy is 0.886
Using feature [3, 1, 6, 5, 2] accuracy is 0.844
Using feature [3, 1, 6, 5, 4] accuracy is 0.848
Using feature [3, 1, 6, 5, 4, 2] accuracy is 0.822
SEARCH FINISHED! The best feature subset is [3, 1], with an accuracy of 0.97.


yutengwu@yutengs-mbp CS170-Project2 % python3 FeatureSelection.py
Welcome to the Feature Selection Algorithm.
Do you want to test with Small file or Large file? (type 's' or 'l'):  s
Please type the number of the file you want to test with 1 - 125. (e.x. '1' or '23' or '36' or '98'):  107
Please type the number of the algorithm you want to use. (1 or 2)
1. Forward Selection
2. Backward Elimination:  2
Using feature [1, 2, 3, 4, 5, 6] accuracy is 0.822
Using feature [1, 2, 3, 4, 5] accuracy is 0.868
Using feature [1, 2, 3, 4, 6] accuracy is 0.816
Using feature [1, 2, 3, 6, 5] accuracy is 0.844
Using feature [1, 2, 6, 5, 4] accuracy is 0.674
Using feature [1, 6, 5, 4, 3] accuracy is 0.848
Using feature [6, 5, 4, 3, 2] accuracy is 0.748
Using feature [1, 2, 3, 4] accuracy is 0.886
Using feature [1, 2, 3, 5] accuracy is 0.902
Using feature [1, 2, 5, 4] accuracy is 0.73
Using feature [1, 5, 4, 3] accuracy is 0.9
Using feature [5, 4, 3, 2] accuracy is 0.802
Using feature [1, 2, 3] accuracy is 0.926
Using feature [1, 2, 5] accuracy is 0.754
Using feature [1, 5, 3] accuracy is 0.928
Using feature [5, 3, 2] accuracy is 0.804
Using feature [1, 3] accuracy is 0.97
Using feature [1, 5] accuracy is 0.726
Using feature [5, 3] accuracy is 0.824
Using feature [1] accuracy is 0.716
Using feature [3] accuracy is 0.858
SEARCH FINISHED! The best feature subset is [1, 3], with an accuracy of 0.97.
```

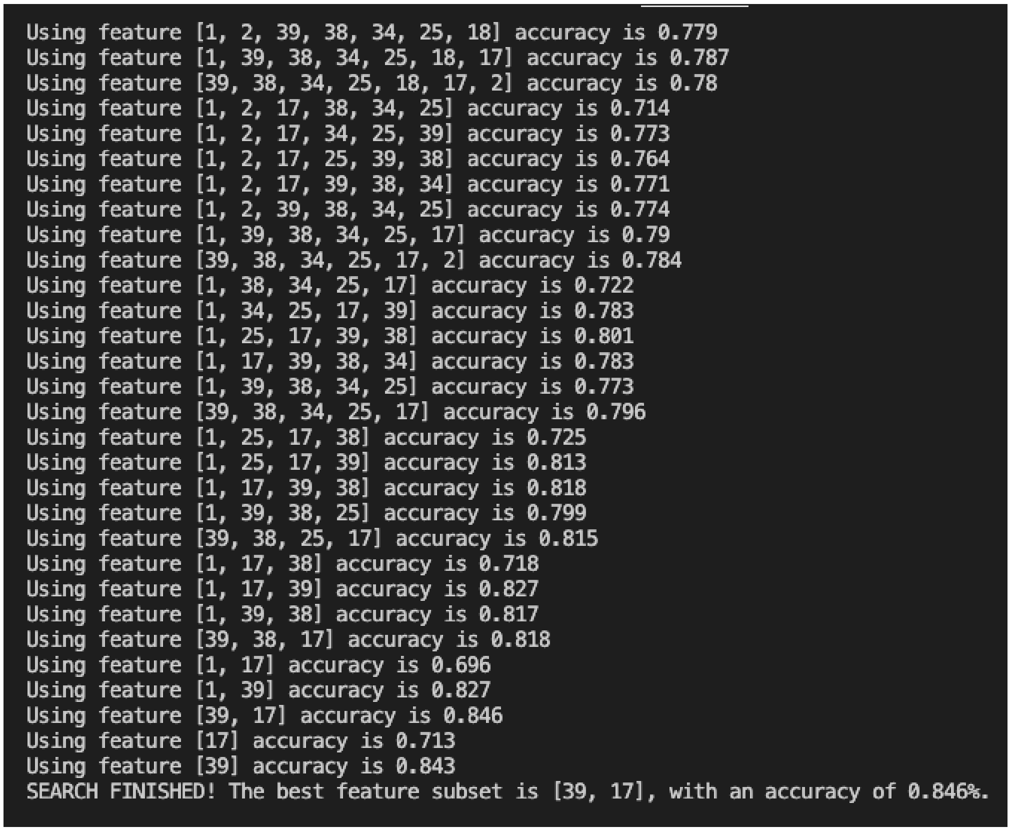
Screenshot for the Forward Selection and Backward Elimination on

CS170_Large_Data__124.txt

{Dear Dr. Eamonn Keogh, sorry for the extra “%” sign typo at the result. I spotted this error at the last minute and I don’t have enough time to rerun the program.}



```
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 15] accuracy is 0.686
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 17] accuracy is 0.707
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20] accuracy is 0.714
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 22] accuracy is 0.684
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 23] accuracy is 0.7
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 25] accuracy is 0.699
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 35] accuracy is 0.699
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 38] accuracy is 0.708
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 9] accuracy is 0.705
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 10] accuracy is 0.7
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 15] accuracy is 0.698
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17] accuracy is 0.718
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 22] accuracy is 0.696
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 23] accuracy is 0.702
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 25] accuracy is 0.702
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 35] accuracy is 0.708
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 38] accuracy is 0.697
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 9] accuracy is 0.699
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 10] accuracy is 0.689
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 15] accuracy is 0.698
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 22] accuracy is 0.674
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 23] accuracy is 0.7
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 25] accuracy is 0.7
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 35] accuracy is 0.704
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38] accuracy is 0.705
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 9] accuracy is 0.69
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 10] accuracy is 0.686
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 15] accuracy is 0.698
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 22] accuracy is 0.699
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 23] accuracy is 0.7
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25] accuracy is 0.719
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 35] accuracy is 0.706
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 9] accuracy is 0.697
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 10] accuracy is 0.685
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 15] accuracy is 0.693
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 22] accuracy is 0.699
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 23] accuracy is 0.695
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35] accuracy is 0.702
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9] accuracy is 0.693
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 10] accuracy is 0.675
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 15] accuracy is 0.683
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 22] accuracy is 0.684
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 23] accuracy is 0.678
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 10] accuracy is 0.665
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 15] accuracy is 0.681
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 22] accuracy is 0.678
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 23] accuracy is 0.681
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 15, 10] accuracy is 0.644
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 15, 22] accuracy is 0.67
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 15, 23] accuracy is 0.662
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 15, 22, 10] accuracy is 0.636
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 15, 22, 23] accuracy is 0.652
Using feature [39, 13, 11, 37, 19, 7, 6, 29, 5, 28, 31, 27, 21, 3, 36, 32, 8, 12, 40, 18, 34, 4, 14, 26, 1, 30, 24, 2, 33, 16, 20, 17, 38, 25, 35, 9, 15, 22, 23, 10] accuracy is 0.628
SEARCH FINISHED! The best feature subset is [39, 13], with an accuracy of 0.963%.
```



```
Using feature [1, 2, 39, 38, 34, 25, 18] accuracy is 0.779
Using feature [1, 39, 38, 34, 25, 18, 17] accuracy is 0.787
Using feature [39, 38, 34, 25, 18, 17, 2] accuracy is 0.78
Using feature [1, 2, 17, 38, 34, 25] accuracy is 0.714
Using feature [1, 2, 17, 34, 25, 39] accuracy is 0.773
Using feature [1, 2, 17, 25, 39, 38] accuracy is 0.764
Using feature [1, 2, 17, 39, 38, 34] accuracy is 0.771
Using feature [1, 2, 39, 38, 34, 25] accuracy is 0.774
Using feature [1, 39, 38, 34, 25, 17] accuracy is 0.79
Using feature [39, 38, 34, 25, 17, 2] accuracy is 0.784
Using feature [1, 38, 34, 25, 17] accuracy is 0.722
Using feature [1, 34, 25, 17, 39] accuracy is 0.783
Using feature [1, 25, 17, 39, 38] accuracy is 0.801
Using feature [1, 17, 39, 38, 34] accuracy is 0.783
Using feature [1, 39, 38, 34, 25] accuracy is 0.773
Using feature [39, 38, 34, 25, 17] accuracy is 0.796
Using feature [1, 25, 17, 38] accuracy is 0.725
Using feature [1, 25, 17, 39] accuracy is 0.813
Using feature [1, 17, 39, 38] accuracy is 0.818
Using feature [1, 39, 38, 25] accuracy is 0.799
Using feature [39, 38, 25, 17] accuracy is 0.815
Using feature [1, 17, 38] accuracy is 0.718
Using feature [1, 17, 39] accuracy is 0.827
Using feature [1, 39, 38] accuracy is 0.817
Using feature [39, 38, 17] accuracy is 0.818
Using feature [1, 17] accuracy is 0.696
Using feature [1, 39] accuracy is 0.827
Using feature [39, 17] accuracy is 0.846
Using feature [17] accuracy is 0.713
Using feature [39] accuracy is 0.843
SEARCH FINISHED! The best feature subset is [39, 17], with an accuracy of 0.846%.
```