Project 2: Feature Selection with Nearest Neighbor

Student Name 1: Ryan Le SID: rle026 Lecture Session: 001
Student Name 2: Aidan Lopez SID: alope396 Lecture Session: 001
Student Name 3: Brandon Tran SID: btran117 Lecture Session: 002
Student Name 4: Vy Vo SID: vvo025 Lecture Session: 001

Solution:

Dataset	Best Feature Set	Accuracy
Small Number (general)	Forward Selection = {3, 5}	92%
	Backward Elimination = {2, 4, 5, 7, 10}	82%
	Custom Algorithm = Not implemented	NA
Large Number (general)	Forward Selection = {1, 27}	95.5%
	Backward Elimination = {27}	84.70%
	Custom Algorithm = Not implemented	NA
Small Number: 1	Forward Selection = {5, 7}	97%
	Backward Elimination = {5, 7, 9}	94%
	Custom Algorithm = Not implemented	NA
Large Number: 1	Forward Selection = {12, 25}	96.8%
	Backward Elimination = {} (empty)	84.7%
	Custom Algorithm = Not implemented	NA

Resources:

We did not use any outside resources besides looking into how to read data from the dataset and the C++ standard libraries.
https://www.geeksforgeeks.org/how-to-split-a-string-in-cc-python-and-java/#

Contribution of each student in the group:

Ryan Le: replaced dummy evaluation function in the feature search algorithm to the leave-one-out validator (part 3) and contributed to writing the report

Aidan Lopez: helped with debugging the classifier and validator (part 2), helped with replacing the dummy evaluation function in the feature search algorithm to the leave-one-out validator (part 3), and contributed to writing the report.

Brandon Tran: completed classifier (leave-one-out) and validator (loading data and calculating accuracy) (part 2) and contributed to writing the report

Vy Vo: completed search feature using forward selection and backward elimination (part 1) and contributed to writing report

I. Introduction:

In this project, we will find the best feature subset given a dataset of instances with many features. We will implement a greedy search that searches for the best accuracy of a feature subset using forward selection and backward elimination. The accuracy of subsets will be found by using a nearest neighbor classifier and the real evaluation function (leave-one-out).

Since the nearest neighbor classifier is very sensitive to irrelevant features, we will need to select these features carefully and normalize them. Using the inputted data set, the program will search for the feature subset(s) that will yield the highest accuracy in classifying each instance.

II. Challenges:

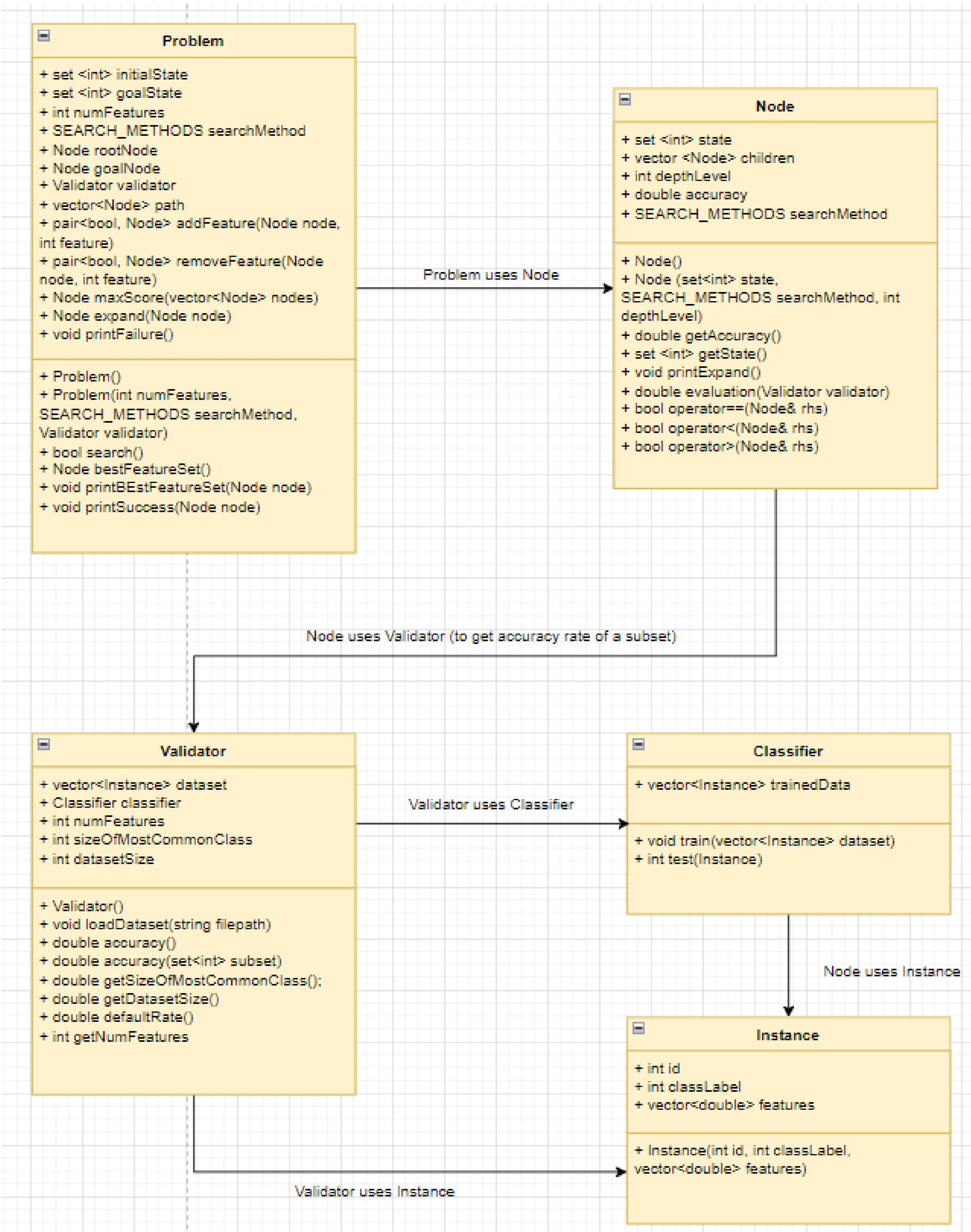
For part 1, a challenge that we faced is tracing through the nodes to make sure that the best node at each depth was selected and expanded on (no backtracking once that node is selected). In addition to this, we had to keep track of a complete path of each best node at each level to find the final highest accuracy feature subset which includes the initial and final state. Lastly, we had to write general code that will work with both forward selection and backward elimination, especially when considering the different operators for forward vs. backward.

For part 2, a challenge that we had encountered was providing the correct instance data to the classifier for leave-one-out. We normalized the data before training the classifier with it, but forgot to normalize the data before testing it as well. This caused us a lot of headaches in debugging. There were also some slight challenges in normalizing the data as well. The process itself was not too difficult, but we had to keep track of a large amount of variables and reiterate over the data various times. This led us to have minor simple mistakes such as forgetting to convert the sums to averages and printing the data as ints rather than floats.

For part 3, a challenge we struggled with was achieving the correct accuracy. When testing with the data, we would get results that were obviously incorrect, which led to a lot of debugging statements. We first figured out that the data set was not being loaded properly, causing our values to be off after the program's first iteration. However, when we had thought we loaded it properly, we were still getting incorrect results. This was because when loading the data, we were unintentionally loading it twice, which affected our results.

III. Code Design:

The flowing block diagram below is the structure/flow of our code.



Problem class searches and expands the tree from initial state to goal state to find the subset with the highest accuracy. The Node class is needed as it contains information about each state (feature subset). The Problem class is where nodes are expanded based on the search method and operators (add or remove features) and at each depth level, the subset with the highest accuracy within the same level is added to the solution path. The overall highest accuracy is found based on a greedy search (no backtracking) from the initial state to goal state.

The Node class contains the state (feature subset) which is used to get the accuracy rate by passing the Node's feature subset to the Validator. It also has an accuracy variable that will be used to make a comparison to get the node (subset) with the highest accuracy rate.

The Validator class is used by the Node class to test the accuracy of a given feature subset. It does so by reading in the feature data, normalizing the data, and storing it. Afterwards, based on the given feature subset, the Validator will trim the data and train the Classifier with this data. Afterwards, the Validator can test all of the instances using the Classifier to determine the accuracy rate.

The Classifier is used by the Validator in order to determine accuracies. The Classifier stores the trimmed data (data with a specific feature subset) and can test various instances against this trimmed data using the leave-one-out method. The result of the test is passed back to the Validator, and the Validator will conclude whether the test was correct or not.

The Instance class is simply a data class that contains each row of data read from the input file. It stores every instance along with the class label, class ID, and feature data.

IV. Dataset Details:

The General Small Dataset:
- # of features: 10
- # of instances: 100

The General Large Dataset:
- # of features: 40
- # of instances: 1000

The Personal Small Dataset:
- # of features: 10
- # of instances: 100

The Personal Large Dataset:
- # of features: 40
- # of instances: 1000

V. Algorithms:

For both forward selection and backward elimination, we will use the general search algorithm to implement the feature search to find the highest accuracy using greedy search.

Forward Selection:

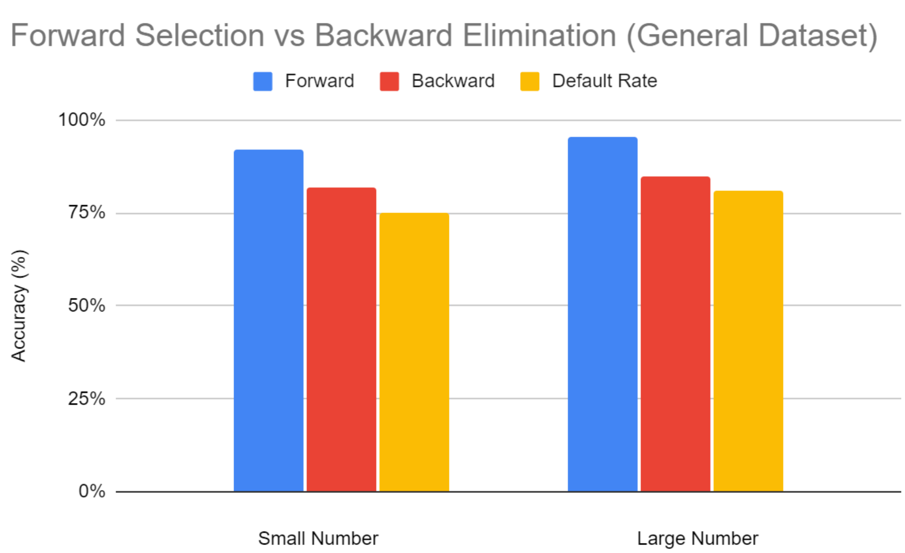
The initial state is empty and the goal state is the complete set with all of the features. At each depth level, as we go down the tree, we will add one more feature to our subset (starting with empty) and then compare all of the expanded nodes to get the feature subset with the highest accuracy. We then add this feature subset to the frontier and then pop and expand this node to get this feature subset's children (which is this feature subset plus one more added feature). Once we reach the goal state, at the end, we will compare all of the subsets that we found at each depth level and get the subset with the highest accuracy rate.

Backward Elimination:

The initial state is the complete set with all of the features and the goal state is empty. At each depth level, as we go down the tree, we will remove one feature from our subset (starting with the full set) and then compare all of the expanded nodes to get the feature subset with the highest accuracy. We then add this feature subset to the frontier and then pop and expand this node to get this feature subset's children (which is this feature subset minus one feature). Once we reach the goal state, at the end, we will compare all of the subsets that we found at each depth level and get the subset with the highest accuracy rate.

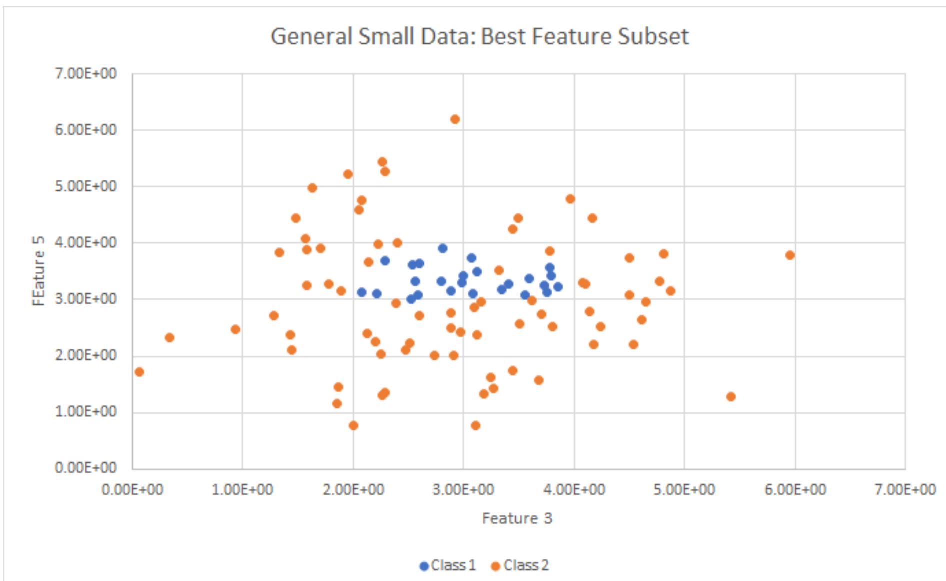
VI. Analysis:

General Test Data:



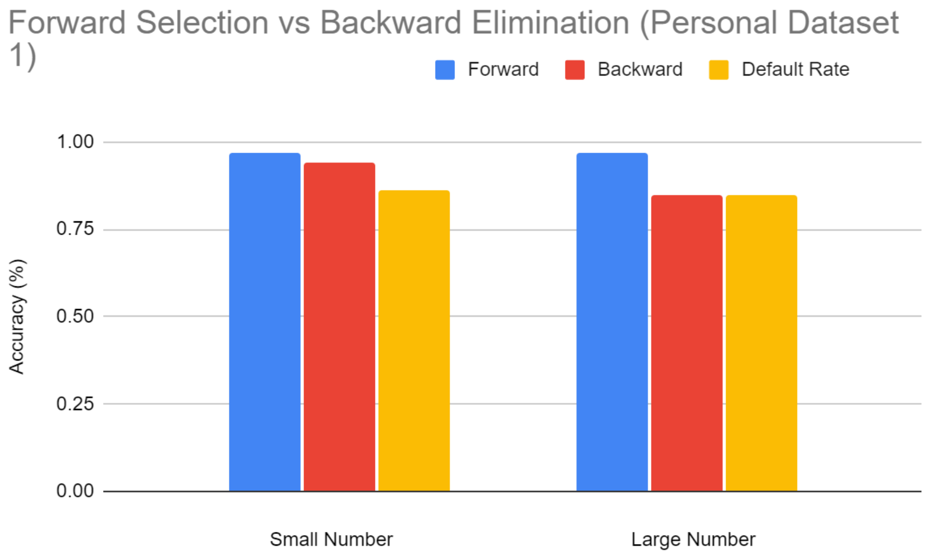
The results of the general test data showed a higher accuracy when using forward selection as compared to backward elimination. When using the small number data, forward selection outperformed backward elimination, resulting in an accuracy of 92% as compared to the 82% accuracy achieved from backward elimination. A similar outcome was reached again when using the large number data, seeing a higher accuracy of 95.5% from forward selection and an accuracy of 84.7% from backward elimination.

When looking at the impact of using feature selection, we can see there is a noticeable increase in accuracy when comparing the accuracy of forward selection and using no features (default rate). Using the small number data, the forward selection classified each instance with an accuracy of 92% as compared to the default rate which has an accuracy of 75%. A similar result is achieved with the large number data, where forward selection yields an accuracy of 95.5% and the default rate yields an accuracy of 81%. Based on this data we can conclude that the use of feature selection creates a higher accuracy when classifying both large and small datasets.



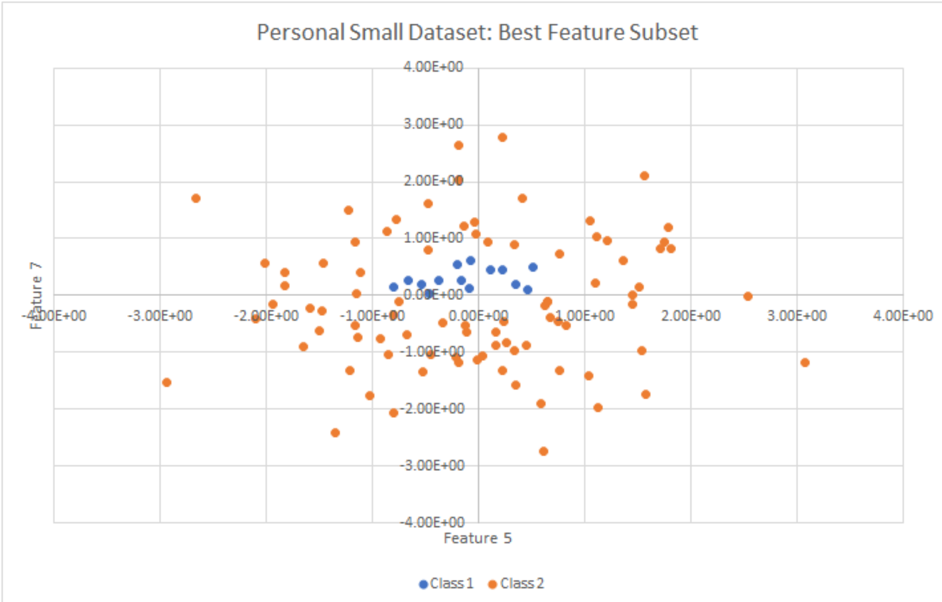
The above plot shows the classification of each instance with the feature subset {3, 5}. When looking at the graph, the data points of class one are all clustered into the center of the graph, being surrounded by class 2. Since all the points are clustered together, using the nearest neighbor approach would result in a high accuracy when classifying the instances (as seen by our results).

Personal Test Data:



The results of the Personal test data displayed another higher accuracy when using forward selection to classify each instance as compared to backwards elimination. Looking at the small number data, forward selection pulls ahead with an accuracy of 97% whereas backward elimination only has an accuracy of 94%. This outcome is repeated when using the large number data, seeing again a higher percentage of 96.8% accuracy from forward selection and an accuracy of 84.7% from backward elimination.

Taking a look at the impact of using feature selection (this time for the personal dataset), we can see there is once again a noticeable increase in accuracy when comparing the accuracy of forward selection and using no features (default rate). When inputting the small number data, the forward selection classified each instance with an accuracy of 97% as compared to the default rate which has an accuracy of 86%. A similar result is achieved with the large number data, where forward selection yields an accuracy of 96.8% and the default rate yields an accuracy of 84.7%. Based on this data we can conclude that the use of feature selection creates a higher accuracy when classifying both large and small datasets. However, something to note is that the default rate and backward elimination accuracies are the same when testing the large number data set. This is due to the fact that the features found by the backward elimination algorithm yielded an accuracy that was lower than that of the default rate, and thus resorted to using an empty set of features.



The above plot shows the classification of each instance with the feature subset {5, 7}. When looking at the graph, the data points of class one are all clustered into the center of the graph, being surrounded by class 2. Since all the points are clustered together, using the nearest neighbor approach would result in a high accuracy when classifying the instances (as seen by our results).

VII. Conclusion:

In part 1, we implemented the greedy search algorithm to find the features subset with the highest accuracy rate using forward section method or backward elimination method. In part 2, we implemented the validator and classification aspects of the code and were able to determine how accurately the program could classify the data set given an inputted set of features. Combining these aspects in part 3 let the program determine which features to use that would yield the highest accuracy when classifying the data sets.

After testing and comparing the two algorithms with varying datasets, both small and large, we found that the forward selection algorithm does a better job of selecting the features to use to classify the data. When testing both large and small for both personal and general data sets, forward selection always allowed for a higher accuracy when classifying each instance. This suggests that the forward selection algorithm is better suited for searching for the best features compared to backward elimination.

VIII. Trace of Datasets

Trace of personal small dataset: FORWARD SELECTION

```
Welcome to Ryan Le (rle026), Aidan Lopez (alope396), Brandon Tran (btran117), and Vy Vo
(vvo025) Project 2.
Select feature algorithm:
1. Forward Selection
2. Backward Elimination
3. Special Algorithm
1
search method = 1
Please enter the file path for your data:
C:\Users\vyvanvo\Documents\UCR\Courses\CS170\Project\CS170-Projects\project2\CS170_
Spring_2023_Small_data__1.txt

This dataset has 10 features (not including the class attribute), with 100 instances
Please wait while we normalize the data... Done!

Running nearest neighbor with no features (default rate), using "leaving-one-out" evaluation,
we get an accuracy of 86.00

Beginning search...

Initial state:  Using feature(s) {} accuracy is 86.00

        Using feature(s) {1} accuracy is 84.00
        Using feature(s) {2} accuracy is 79.00
        Using feature(s) {3} accuracy is 76.00
        Using feature(s) {4} accuracy is 77.00
        Using feature(s) {5} accuracy is 78.00
        Using feature(s) {6} accuracy is 75.00
        Using feature(s) {7} accuracy is 90.00
        Using feature(s) {8} accuracy is 80.00
        Using feature(s) {9} accuracy is 77.00
        Using feature(s) {10} accuracy is 79.00

Feature subset {7} was best, accuracy is 90.00

        Using feature(s) {1, 7} accuracy is 79.00
        Using feature(s) {2, 7} accuracy is 85.00
        Using feature(s) {3, 7} accuracy is 89.00
        Using feature(s) {4, 7} accuracy is 83.00
        Using feature(s) {5, 7} accuracy is 97.00
```

Using feature(s) {6, 7} accuracy is 85.00
Using feature(s) {7, 8} accuracy is 86.00
Using feature(s) {7, 9} accuracy is 84.00
Using feature(s) {7, 10} accuracy is 84.00

Feature subset {5, 7} was best, accuracy is 97.00

Using feature(s) {1, 5, 7} accuracy is 91.00
Using feature(s) {2, 5, 7} accuracy is 90.00
Using feature(s) {3, 5, 7} accuracy is 97.00
Using feature(s) {4, 5, 7} accuracy is 87.00
Using feature(s) {5, 6, 7} accuracy is 96.00
Using feature(s) {5, 7, 8} accuracy is 90.00
Using feature(s) {5, 7, 9} accuracy is 94.00
Using feature(s) {5, 7, 10} accuracy is 93.00

Feature subset {3, 5, 7} was best, accuracy is 97.00

Using feature(s) {1, 3, 5, 7} accuracy is 88.00
Using feature(s) {2, 3, 5, 7} accuracy is 86.00
Using feature(s) {3, 4, 5, 7} accuracy is 88.00
Using feature(s) {3, 5, 6, 7} accuracy is 92.00
Using feature(s) {3, 5, 7, 8} accuracy is 83.00
Using feature(s) {3, 5, 7, 9} accuracy is 94.00
Using feature(s) {3, 5, 7, 10} accuracy is 87.00

Feature subset {3, 5, 7, 9} was best, accuracy is 94.00

Using feature(s) {1, 3, 5, 7, 9} accuracy is 82.00
Using feature(s) {2, 3, 5, 7, 9} accuracy is 85.00
Using feature(s) {3, 4, 5, 7, 9} accuracy is 86.00
Using feature(s) {3, 5, 6, 7, 9} accuracy is 89.00
Using feature(s) {3, 5, 7, 8, 9} accuracy is 80.00
Using feature(s) {3, 5, 7, 9, 10} accuracy is 85.00

Feature subset {3, 5, 6, 7, 9} was best, accuracy is 89.00

Using feature(s) {1, 3, 5, 6, 7, 9} accuracy is 81.00
Using feature(s) {2, 3, 5, 6, 7, 9} accuracy is 82.00
Using feature(s) {3, 4, 5, 6, 7, 9} accuracy is 84.00

Using feature(s) {3, 5, 6, 7, 8, 9} accuracy is 77.00
Using feature(s) {3, 5, 6, 7, 9, 10} accuracy is 87.00

Feature subset {3, 5, 6, 7, 9, 10} was best, accuracy is 87.00

Using feature(s) {1, 3, 5, 6, 7, 9, 10} accuracy is 81.00
Using feature(s) {2, 3, 5, 6, 7, 9, 10} accuracy is 79.00
Using feature(s) {3, 4, 5, 6, 7, 9, 10} accuracy is 79.00
Using feature(s) {3, 5, 6, 7, 8, 9, 10} accuracy is 75.00

Feature subset {1, 3, 5, 6, 7, 9, 10} was best, accuracy is 81.00

Using feature(s) {1, 2, 3, 5, 6, 7, 9, 10} accuracy is 73.00
Using feature(s) {1, 3, 4, 5, 6, 7, 9, 10} accuracy is 79.00
Using feature(s) {1, 3, 5, 6, 7, 8, 9, 10} accuracy is 76.00

Feature subset {1, 3, 4, 5, 6, 7, 9, 10} was best, accuracy is 79.00

Using feature(s) {1, 2, 3, 4, 5, 6, 7, 9, 10} accuracy is 75.00
Using feature(s) {1, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 77.00

Feature subset {1, 3, 4, 5, 6, 7, 8, 9, 10} was best, accuracy is 77.00

Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 76.00

Feature subset {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} was best, accuracy is 76.00

Finished search!! The best feature subset is {5, 7} , which has an accuracy of 97.00

Trace of personal dataset: BACKWARD SELECTION

Welcome to Ryan Le (rle026), Aidan Lopez (alope396), Brandon Tran (btran117), and Vy Vo (vvo025) Project 2.
Select feature algorithm:
1. Forward Selection
2. Backward Elimination
3. Special Algorithm
2
search method = 2
Please enter the file path for your data:

```
C:\Users\vyvanvo\Documents\UCR\Courses\CS170\Project\CS170-Projects\project2\CS170_
Spring_2023_Small_data__1.txt

This dataset has 10 features (not including the class attribute), with 100 instances
Please wait while we normalize the data... Done!

Running nearest neighbor with no features (default rate), using "leaving-one-out" evaluation,
we get an accuracy of 86.00

Beginning search...

Initial state:  Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 76.00

        Using feature(s) {2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 75.00
        Using feature(s) {1, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 77.00
        Using feature(s) {1, 2, 4, 5, 6, 7, 8, 9, 10} accuracy is 81.00
        Using feature(s) {1, 2, 3, 5, 6, 7, 8, 9, 10} accuracy is 72.00
        Using feature(s) {1, 2, 3, 4, 6, 7, 8, 9, 10} accuracy is 74.00
        Using feature(s) {1, 2, 3, 4, 5, 7, 8, 9, 10} accuracy is 82.00
        Using feature(s) {1, 2, 3, 4, 5, 6, 8, 9, 10} accuracy is 72.00
        Using feature(s) {1, 2, 3, 4, 5, 6, 7, 9, 10} accuracy is 75.00
        Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 10} accuracy is 77.00
        Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9} accuracy is 80.00

Feature subset {1, 2, 3, 4, 5, 7, 8, 9, 10} was best, accuracy is 82.00

        Using feature(s) {2, 3, 4, 5, 7, 8, 9, 10} accuracy is 78.00
        Using feature(s) {1, 3, 4, 5, 7, 8, 9, 10} accuracy is 80.00
        Using feature(s) {1, 2, 4, 5, 7, 8, 9, 10} accuracy is 85.00
        Using feature(s) {1, 2, 3, 5, 7, 8, 9, 10} accuracy is 69.00
        Using feature(s) {1, 2, 3, 4, 7, 8, 9, 10} accuracy is 78.00
        Using feature(s) {1, 2, 3, 4, 5, 8, 9, 10} accuracy is 79.00
        Using feature(s) {1, 2, 3, 4, 5, 7, 9, 10} accuracy is 81.00
        Using feature(s) {1, 2, 3, 4, 5, 7, 8, 10} accuracy is 81.00
        Using feature(s) {1, 2, 3, 4, 5, 7, 8, 9} accuracy is 79.00

Feature subset {1, 2, 4, 5, 7, 8, 9, 10} was best, accuracy is 85.00

        Using feature(s) {2, 4, 5, 7, 8, 9, 10} accuracy is 87.00
        Using feature(s) {1, 4, 5, 7, 8, 9, 10} accuracy is 84.00
        Using feature(s) {1, 2, 5, 7, 8, 9, 10} accuracy is 79.00
        Using feature(s) {1, 2, 4, 7, 8, 9, 10} accuracy is 82.00
        Using feature(s) {1, 2, 4, 5, 8, 9, 10} accuracy is 83.00
        Using feature(s) {1, 2, 4, 5, 7, 9, 10} accuracy is 79.00
        Using feature(s) {1, 2, 4, 5, 7, 8, 10} accuracy is 86.00
        Using feature(s) {1, 2, 4, 5, 7, 8, 9} accuracy is 87.00
```

Feature subset {2, 4, 5, 7, 8, 9, 10} was best, accuracy is 87.00

Using feature(s) {4, 5, 7, 8, 9, 10} accuracy is 85.00
Using feature(s) {2, 5, 7, 8, 9, 10} accuracy is 79.00
Using feature(s) {2, 4, 7, 8, 9, 10} accuracy is 81.00
Using feature(s) {2, 4, 5, 8, 9, 10} accuracy is 78.00
Using feature(s) {2, 4, 5, 7, 9, 10} accuracy is 79.00
Using feature(s) {2, 4, 5, 7, 8, 10} accuracy is 85.00
Using feature(s) {2, 4, 5, 7, 8, 9} accuracy is 83.00

Feature subset {4, 5, 7, 8, 9, 10} was best, accuracy is 85.00

Using feature(s) {5, 7, 8, 9, 10} accuracy is 81.00
Using feature(s) {4, 7, 8, 9, 10} accuracy is 81.00
Using feature(s) {4, 5, 8, 9, 10} accuracy is 73.00
Using feature(s) {4, 5, 7, 9, 10} accuracy is 87.00
Using feature(s) {4, 5, 7, 8, 10} accuracy is 83.00
Using feature(s) {4, 5, 7, 8, 9} accuracy is 84.00

Feature subset {4, 5, 7, 9, 10} was best, accuracy is 87.00

Using feature(s) {5, 7, 9, 10} accuracy is 90.00
Using feature(s) {4, 7, 9, 10} accuracy is 79.00
Using feature(s) {4, 5, 9, 10} accuracy is 73.00
Using feature(s) {4, 5, 7, 10} accuracy is 85.00
Using feature(s) {4, 5, 7, 9} accuracy is 88.00

Feature subset {5, 7, 9, 10} was best, accuracy is 90.00

Using feature(s) {7, 9, 10} accuracy is 83.00
Using feature(s) {5, 9, 10} accuracy is 77.00
Using feature(s) {5, 7, 10} accuracy is 93.00
Using feature(s) {5, 7, 9} accuracy is 94.00

Feature subset {5, 7, 9} was best, accuracy is 94.00

Using feature(s) {7, 9} accuracy is 84.00
Using feature(s) {5, 9} accuracy is 85.00
Using feature(s) {5, 7} accuracy is 97.00

Feature subset {5, 7} was best, accuracy is 97.00

Using feature(s) {7} accuracy is 90.00
Using feature(s) {5} accuracy is 78.00

```
Feature subset {7} was best, accuracy is 90.00

        Using feature(s) {} accuracy is 86.00

Feature subset {} was best, accuracy is 86.00

Finished search!! The best feature subset is {5, 7} , which has an accuracy of 97.00
```

Trace of general small dataset: FORWARD SELECTION

```
Welcome to Ryan Le (rle026), Aidan Lopez (alope396), Brandon Tran (btran117), and Vy Vo
(vvo025) Project 2.
Select feature algorithm:
1. Forward Selection
2. Backward Elimination
3. Special Algorithm
1
search method = 1
Please enter the file path for your data:
C:\Users\vyvanvo\Documents\UCR\Courses\CS170\Project\CS170-Projects\project2\data\sma
ll-test-dataset.txt

This dataset has 10 features (not including the class attribute), with 100 instances
Please wait while we normalize the data... Done!

Running nearest neighbor with no features (default rate), using "leaving-one-out" evaluation,
we get an accuracy of 75.00

Beginning search...

Initial state:  Using feature(s) {} accuracy is 75.00

        Using feature(s) {1} accuracy is 57.00
        Using feature(s) {2} accuracy is 54.00
        Using feature(s) {3} accuracy is 68.00
        Using feature(s) {4} accuracy is 65.00
        Using feature(s) {5} accuracy is 75.00
        Using feature(s) {6} accuracy is 61.00
        Using feature(s) {7} accuracy is 62.00
        Using feature(s) {8} accuracy is 60.00
        Using feature(s) {9} accuracy is 66.00
        Using feature(s) {10} accuracy is 64.00

Feature subset {5} was best, accuracy is 75.00
```

Using feature(s) {1, 5} accuracy is 76.00
Using feature(s) {2, 5} accuracy is 80.00
Using feature(s) {3, 5} accuracy is 92.00
Using feature(s) {4, 5} accuracy is 75.00
Using feature(s) {5, 6} accuracy is 79.00
Using feature(s) {5, 7} accuracy is 80.00
Using feature(s) {5, 8} accuracy is 77.00
Using feature(s) {5, 9} accuracy is 73.00
Using feature(s) {5, 10} accuracy is 82.00

Feature subset {3, 5} was best, accuracy is 92.00

Using feature(s) {1, 3, 5} accuracy is 83.00
Using feature(s) {2, 3, 5} accuracy is 79.00
Using feature(s) {3, 4, 5} accuracy is 84.00
Using feature(s) {3, 5, 6} accuracy is 82.00
Using feature(s) {3, 5, 7} accuracy is 89.00
Using feature(s) {3, 5, 8} accuracy is 79.00
Using feature(s) {3, 5, 9} accuracy is 82.00
Using feature(s) {3, 5, 10} accuracy is 85.00

Feature subset {3, 5, 7} was best, accuracy is 89.00

Using feature(s) {1, 3, 5, 7} accuracy is 88.00
Using feature(s) {2, 3, 5, 7} accuracy is 81.00
Using feature(s) {3, 4, 5, 7} accuracy is 78.00
Using feature(s) {3, 5, 6, 7} accuracy is 88.00
Using feature(s) {3, 5, 7, 8} accuracy is 80.00
Using feature(s) {3, 5, 7, 9} accuracy is 82.00
Using feature(s) {3, 5, 7, 10} accuracy is 84.00

Feature subset {1, 3, 5, 7} was best, accuracy is 88.00

Using feature(s) {1, 2, 3, 5, 7} accuracy is 79.00
Using feature(s) {1, 3, 4, 5, 7} accuracy is 77.00
Using feature(s) {1, 3, 5, 6, 7} accuracy is 86.00
Using feature(s) {1, 3, 5, 7, 8} accuracy is 75.00
Using feature(s) {1, 3, 5, 7, 9} accuracy is 75.00
Using feature(s) {1, 3, 5, 7, 10} accuracy is 75.00

Feature subset {1, 3, 5, 6, 7} was best, accuracy is 86.00

Using feature(s) {1, 2, 3, 5, 6, 7} accuracy is 76.00
Using feature(s) {1, 3, 4, 5, 6, 7} accuracy is 73.00
Using feature(s) {1, 3, 5, 6, 7, 8} accuracy is 78.00
Using feature(s) {1, 3, 5, 6, 7, 9} accuracy is 71.00

```
    Using feature(s) {1, 3, 5, 6, 7, 10} accuracy is 71.00

Feature subset {1, 3, 5, 6, 7, 8} was best, accuracy is 78.00

    Using feature(s) {1, 2, 3, 5, 6, 7, 8} accuracy is 68.00
    Using feature(s) {1, 3, 4, 5, 6, 7, 8} accuracy is 68.00
    Using feature(s) {1, 3, 5, 6, 7, 8, 9} accuracy is 72.00
    Using feature(s) {1, 3, 5, 6, 7, 8, 10} accuracy is 67.00

Feature subset {1, 3, 5, 6, 7, 8, 9} was best, accuracy is 72.00

    Using feature(s) {1, 2, 3, 5, 6, 7, 8, 9} accuracy is 70.00
    Using feature(s) {1, 3, 4, 5, 6, 7, 8, 9} accuracy is 64.00
    Using feature(s) {1, 3, 5, 6, 7, 8, 9, 10} accuracy is 67.00

Feature subset {1, 2, 3, 5, 6, 7, 8, 9} was best, accuracy is 70.00

    Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9} accuracy is 70.00
    Using feature(s) {1, 2, 3, 5, 6, 7, 8, 9, 10} accuracy is 68.00

Feature subset {1, 2, 3, 4, 5, 6, 7, 8, 9} was best, accuracy is 70.00

    Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 65.00

Feature subset {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} was best, accuracy is 65.00

Finished search!! The best feature subset is {3, 5} , which has an accuracy of 92.00
```

Trace of general smalldataset: BACKWARD ELIMINATION

```
Welcome to Ryan Le (rle026), Aidan Lopez (alope396), Brandon Tran (btran117), and Vy Vo
(vvo025) Project 2.
Select feature algorithm:
1. Forward Selection
2. Backward Elimination
3. Special Algorithm
2
search method = 2
Please enter the file path for your data:
C:\Users\vyvanvo\Documents\UCR\Courses\CS170\Project\CS170-Projects\project2\data\sma
ll-test-dataset.txt

This dataset has 10 features (not including the class attribute), with 100 instances
Please wait while we normalize the data... Done!
```

Running nearest neighbor with no features (default rate), using "leaving-one-out" evaluation, we get an accuracy of 75.00

Beginning search...

Initial state: Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 65.00

Using feature(s) {2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 69.00
Using feature(s) {1, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 66.00
Using feature(s) {1, 2, 4, 5, 6, 7, 8, 9, 10} accuracy is 75.00
Using feature(s) {1, 2, 3, 5, 6, 7, 8, 9, 10} accuracy is 68.00
Using feature(s) {1, 2, 3, 4, 6, 7, 8, 9, 10} accuracy is 71.00
Using feature(s) {1, 2, 3, 4, 5, 7, 8, 9, 10} accuracy is 69.00
Using feature(s) {1, 2, 3, 4, 5, 6, 8, 9, 10} accuracy is 61.00
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 9, 10} accuracy is 70.00
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 10} accuracy is 67.00
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9} accuracy is 70.00

Feature subset {1, 2, 4, 5, 6, 7, 8, 9, 10} was best, accuracy is 75.00

Using feature(s) {2, 4, 5, 6, 7, 8, 9, 10} accuracy is 73.00
Using feature(s) {1, 4, 5, 6, 7, 8, 9, 10} accuracy is 74.00
Using feature(s) {1, 2, 5, 6, 7, 8, 9, 10} accuracy is 71.00
Using feature(s) {1, 2, 4, 6, 7, 8, 9, 10} accuracy is 64.00
Using feature(s) {1, 2, 4, 5, 7, 8, 9, 10} accuracy is 77.00
Using feature(s) {1, 2, 4, 5, 6, 8, 9, 10} accuracy is 65.00
Using feature(s) {1, 2, 4, 5, 6, 7, 9, 10} accuracy is 70.00
Using feature(s) {1, 2, 4, 5, 6, 7, 8, 10} accuracy is 74.00
Using feature(s) {1, 2, 4, 5, 6, 7, 8, 9} accuracy is 68.00

Feature subset {1, 2, 4, 5, 7, 8, 9, 10} was best, accuracy is 77.00

Using feature(s) {2, 4, 5, 7, 8, 9, 10} accuracy is 77.00
Using feature(s) {1, 4, 5, 7, 8, 9, 10} accuracy is 74.00
Using feature(s) {1, 2, 5, 7, 8, 9, 10} accuracy is 63.00
Using feature(s) {1, 2, 4, 7, 8, 9, 10} accuracy is 59.00
Using feature(s) {1, 2, 4, 5, 8, 9, 10} accuracy is 69.00
Using feature(s) {1, 2, 4, 5, 7, 9, 10} accuracy is 78.00
Using feature(s) {1, 2, 4, 5, 7, 8, 10} accuracy is 74.00
Using feature(s) {1, 2, 4, 5, 7, 8, 9} accuracy is 71.00

Feature subset {1, 2, 4, 5, 7, 9, 10} was best, accuracy is 78.00

Using feature(s) {2, 4, 5, 7, 9, 10} accuracy is 76.00
Using feature(s) {1, 4, 5, 7, 9, 10} accuracy is 74.00
Using feature(s) {1, 2, 5, 7, 9, 10} accuracy is 68.00

Using feature(s) {1, 2, 4, 7, 9, 10} accuracy is 65.00
Using feature(s) {1, 2, 4, 5, 9, 10} accuracy is 74.00
Using feature(s) {1, 2, 4, 5, 7, 10} accuracy is 80.00
Using feature(s) {1, 2, 4, 5, 7, 9} accuracy is 75.00

Feature subset {1, 2, 4, 5, 7, 10} was best, accuracy is 80.00

Using feature(s) {2, 4, 5, 7, 10} accuracy is 82.00
Using feature(s) {1, 4, 5, 7, 10} accuracy is 72.00
Using feature(s) {1, 2, 5, 7, 10} accuracy is 73.00
Using feature(s) {1, 2, 4, 7, 10} accuracy is 61.00
Using feature(s) {1, 2, 4, 5, 10} accuracy is 72.00
Using feature(s) {1, 2, 4, 5, 7} accuracy is 77.00

Feature subset {2, 4, 5, 7, 10} was best, accuracy is 82.00

Feature subset {2, 5, 7} was best, accuracy is 77.00

Using feature(s) {5, 7} accuracy is 80.00
Using feature(s) {2, 7} accuracy is 55.00
Using feature(s) {2, 5} accuracy is 80.00

Feature subset {5, 7} was best, accuracy is 80.00

Using feature(s) {7} accuracy is 62.00
Using feature(s) {5} accuracy is 75.00

Feature subset {5} was best, accuracy is 75.00

Using feature(s) {} accuracy is 75.00

Feature subset {} was best, accuracy is 75.00

Finished search!! The best feature subset is {2, 4, 5, 7, 10} , which has an accuracy of 82.00