Clement Chen, cchen606

https://github.com/cchen606/cs170_project2

Source code in github

CS170 Project 2 Final Report

Results

- Group: Clement Chen – cchen606
- Small Dataset Results:
    - Forward: Feature Subset: {5, 3}, Acc: 92.0%
    - Backward: Feature Subset: {3, 5} Acc: 92.0%
- Large Dataset Results:
    - Forward: Feature Subset: {27, 1}, Acc: 95.5%
    - Backward: Feature Subset: {1, 27}, Acc: 95.5%
- Titanic Dataset Results:
    - Forward: Feature Subset: {2}, Acc: 78.01%
    - Backward: Feature Subset: {2}, Acc: 78.01%

(Trace for small dataset and titanic given at end of report)

Contributions: I worked on this project alone. I am not in a group

Challenges: The hardest part of this project for me was figuring out how to implement the feature search algorithm. The bulk of the time I spent on the project was in Part I. I felt that the classifier and Validation classes were more straightforward.

Design:

When implementing this project, I tried to follow the instructions for each part of the project closely to try to not to make it any harder than I had to. In part 1, I simply made two functions to complete the forward and backward search functions (not counting the dummy evaluation function and main function). I thought that I would be updating the evaluation function later on, but when I got to part 2, the instructions recommended that I implement the validation and classification as classes, and create objects instead of using functions directly, so I did this.

Analysis:

At one point I realized that the search function took a long time to run on the large dataset that was provided. So I tried to optimize some of my code to make it run faster. I tried to be more smart about how I handled the vectors, and tried to make sure that I was not copying and pasting vectors for no reason.

Across the datasets the backwards elimination algorithm was slightly faster than the forward selection. However, the forward selection algorithm personally made more sense to me. Another interesting thing I noticed was just how much longer it took for the large test database to run. Although it is a much bigger database, I was surprised that the runtime was close to 80 times longer than the small database for both algorithms.

References:

Resources used: cplusplus.com, cppreference.com, stackoverflow.com, simplilearn.com, geeksforgeeks.org

Traces:

Trace (running on small-test-dataset.txt)

First run is Forward Selection, second run is Backwards Elimination

Welcome to Clement Chen's Feature Selection Algorithm.

Type in the name of the file to test: titanic_clean.txt

Type the number of the algorithm you want to run:

1. Forward Selection

2. Backward Elimination

1

Beginning search...

Using feature(s) { 1 } accuracy is 66.39%

Using feature(s) { 2 } accuracy is 78.01%

Using feature(s) { 3 } accuracy is 56.02%

Using feature(s) { 4 } accuracy is 42.16%

Using feature(s) { 5 } accuracy is 60.08%

Using feature(s) { 6 } accuracy is 66.25%

Feature set { 2 } was best, accuracy is 78.01%

Using feature(s) { 2 1 } accuracy is 77.73%

Using feature(s) { 2 3 } accuracy is 68.49%

Using feature(s) { 2 4 } accuracy is 76.89%

Using feature(s) { 2 5 } accuracy is 77.87%

Using feature(s) { 2 6 } accuracy is 75.07%

Feature set { 2 5 } was best, accuracy is 77.87%

Using feature(s) { 2 5 1 } accuracy is 76.75%

Using feature(s) { 2 5 3 } accuracy is 67.93%

Using feature(s) { 2 5 4 } accuracy is 77.03%

Using feature(s) { 2 5 6 } accuracy is 74.79%

Feature set { 2 5 4 } was best, accuracy is 77.03%

Using feature(s) { 2 5 4 1 } accuracy is 76.19%

Using feature(s) { 2 5 4 3 } accuracy is 70.45%

Using feature(s) { 2 5 4 6 } accuracy is 73.81%

Feature set { 2 5 4 1 } was best, accuracy is 76.19%

Using feature(s) { 2 5 4 1 3 } accuracy is 75.77%

Using feature(s) { 2 5 4 1 6 } accuracy is 74.09%

Feature set { 2 5 4 1 3 } was best, accuracy is 75.77%

Using feature(s) { 2 5 4 1 3 6 } accuracy is 73.95%

Feature set { 2 5 4 1 3 6 } was best, accuracy is 73.95%

Finished search!! The best feature subset is { 2 }, which has an accuracy of 78.01%

PS C:\Users\Clem\Desktop\local code\170\proj2\output> & .\'proj2_part3.exe'

Welcome to Clement Chen's Feature Selection Algorithm.

Type in the name of the file to test: titanic_clean.txt

Type the number of the algorithm you want to run:

1. Forward Selection

2. Backward Elimination

2

Using all features { 1 2 3 4 5 6 } accuracy is 73.95%

Beginning search.

Using feature(s) { 2 3 4 5 6 } accuracy is 72.97%

Using feature(s) { 1 3 4 5 6 } accuracy is 62.18%

Using feature(s) { 1 2 4 5 6 } accuracy is 74.09%

Using feature(s) { 1 2 3 5 6 } accuracy is 74.09%

Using feature(s) { 1 2 3 4 6 } accuracy is 75.35%

Using feature(s) { 1 2 3 4 5 } accuracy is 75.77%

Feature set { 1 2 3 4 5 } was best, accuracy is 75.77%

Using feature(s) { 2 3 4 5 } accuracy is 70.45%

Using feature(s) { 1 3 4 5 } accuracy is 63.31%

Using feature(s) { 1 2 4 5 } accuracy is 76.19%

Using feature(s) { 1 2 3 5 } accuracy is 75.35%

Using feature(s) { 1 2 3 4 } accuracy is 75.77%

Feature set { 1 2 4 5 } was best, accuracy is 76.19%

Using feature(s) { 2 4 5 } accuracy is 77.03%

Using feature(s) { 1 4 5 } accuracy is 41.18%

Using feature(s) { 1 2 5 } accuracy is 76.75%

Using feature(s) { 1 2 4 } accuracy is 77.31%

Feature set { 1 2 4 } was best, accuracy is 77.31%

Using feature(s) { 2 4 } accuracy is 76.89%

Using feature(s) { 1 4 } accuracy is 43.14%

Using feature(s) { 1 2 } accuracy is 77.73%

Feature set { 1 2 } was best, accuracy is 77.73%

Using feature(s) { 2 } accuracy is 78.01%

Using feature(s) { 1 } accuracy is 66.39%

Feature set { 2 } was best, accuracy is 78.01%

Finished search!! The best feature subset is { 2 }, which has an accuracy of 78.01%

PS C:\Users\Clem\Desktop\local code\170\proj2\output>

Trace (running on small-test-dataset.txt)

First run is Forward Selection, second run is Backwards Elimination

Welcome to Clement Chen's Feature Selection Algorithm.

Type in the name of the file to test: small-test-dataset.txt

Type the number of the algorithm you want to run:

1. Forward Selection

2. Backward Elimination

1

Beginning search...

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 set { 5 } was best, accuracy is 75.00%

Using feature(s) { 5 1 } accuracy is 75.00%

Using feature(s) { 5 2 } accuracy is 80.00%

Using feature(s) { 5 3 } accuracy is 92.00%

Using feature(s) { 5 4 } accuracy is 76.00%

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 81.00%

Feature set { 5 3 } was best, accuracy is 92.00%

Using feature(s) { 5 3 1 } accuracy is 85.00%

Using feature(s) { 5 3 2 } accuracy is 80.00%

Using feature(s) { 5 3 4 } accuracy is 82.00%

Using feature(s) { 5 3 6 } accuracy is 82.00%

Using feature(s) { 5 3 7 } accuracy is 91.00%

Using feature(s) { 5 3 8 } accuracy is 79.00%

Using feature(s) { 5 3 9 } accuracy is 84.00%

Using feature(s) { 5 3 10 } accuracy is 85.00%

Feature set { 5 3 7 } was best, accuracy is 91.00%

Using feature(s) { 5 3 7 1 } accuracy is 88.00%

Using feature(s) { 5 3 7 2 } accuracy is 83.00%

Using feature(s) { 5 3 7 4 } accuracy is 78.00%

Using feature(s) { 5 3 7 6 } accuracy is 86.00%

Using feature(s) { 5 3 7 8 } accuracy is 79.00%

Using feature(s) { 5 3 7 9 } accuracy is 81.00%

Using feature(s) { 5 3 7 10 } accuracy is 84.00%

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

Using feature(s) { 5 3 7 1 2 } accuracy is 78.00%

Using feature(s) { 5 3 7 1 4 } accuracy is 75.00%

Using feature(s) { 5 3 7 1 6 } accuracy is 85.00%

Using feature(s) { 5 3 7 1 8 } accuracy is 74.00%

Using feature(s) { 5 3 7 1 9 } accuracy is 72.00%

Using feature(s) { 5 3 7 1 10 } accuracy is 74.00%

Feature set { 5 3 7 1 6 } was best, accuracy is 85.00%

Using feature(s) { 5 3 7 1 6 2 } accuracy is 77.00%

Using feature(s) { 5 3 7 1 6 4 } accuracy is 72.00%

Using feature(s) { 5 3 7 1 6 8 } accuracy is 75.00%

Using feature(s) { 5 3 7 1 6 9 } accuracy is 70.00%

Using feature(s) { 5 3 7 1 6 10 } accuracy is 70.00%

Feature set { 5 3 7 1 6 2 } was best, accuracy is 77.00%

Using feature(s) { 5 3 7 1 6 2 4 } accuracy is 73.00%

Using feature(s) { 5 3 7 1 6 2 8 } accuracy is 65.00%

Using feature(s) { 5 3 7 1 6 2 9 } accuracy is 73.00%

Using feature(s) { 5 3 7 1 6 2 10 } accuracy is 68.00%

Feature set { 5 3 7 1 6 2 4 } was best, accuracy is 73.00%

Using feature(s) { 5 3 7 1 6 2 4 8 } accuracy is 64.00%

Using feature(s) { 5 3 7 1 6 2 4 9 } accuracy is 69.00%

Using feature(s) { 5 3 7 1 6 2 4 10 } accuracy is 68.00%

Feature set { 5 3 7 1 6 2 4 9 } was best, accuracy is 69.00%

Using feature(s) { 5 3 7 1 6 2 4 9 8 } accuracy is 69.00%

Using feature(s) { 5 3 7 1 6 2 4 9 10 } accuracy is 72.00%

Feature set { 5 3 7 1 6 2 4 9 10 } was best, accuracy is 72.00%

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

Feature set { 5 3 7 1 6 2 4 9 10 8 } was best, accuracy is 68.00%

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

PS C:\Users\Clem\Desktop\local code\170\proj2\output> & .\'proj2_part3.exe'

Welcome to Clement Chen's Feature Selection Algorithm.

Type in the name of the file to test: small-test-dataset.txt

Type the number of the algorithm you want to run:

1. Forward Selection

2. Backward Elimination

2

Using all features { 1 2 3 4 5 6 7 8 9 10 } accuracy is 68.00%

Beginning search.

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

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 70.00%

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

Using feature(s) { 1 2 3 4 6 7 8 9 10 } accuracy is 66.00%

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

Using feature(s) { 1 2 3 4 5 6 8 9 10 } accuracy is 62.00%

Using feature(s) { 1 2 3 4 5 6 7 9 10 } accuracy is 72.00%

Using feature(s) { 1 2 3 4 5 6 7 8 10 } accuracy is 63.00%

Using feature(s) { 1 2 3 4 5 6 7 8 9 } accuracy is 69.00%

Feature set { 1 2 3 4 5 6 7 9 10 } was best, accuracy is 72.00%

Using feature(s) { 2 3 4 5 6 7 9 10 } accuracy is 73.00%

Using feature(s) { 1 3 4 5 6 7 9 10 } accuracy is 67.00%

Using feature(s) { 1 2 4 5 6 7 9 10 } accuracy is 70.00%

Using feature(s) { 1 2 3 5 6 7 9 10 } accuracy is 69.00%

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

Using feature(s) { 1 2 3 4 5 7 9 10 } accuracy is 70.00%

Using feature(s) { 1 2 3 4 5 6 9 10 } accuracy is 66.00%

Using feature(s) { 1 2 3 4 5 6 7 10 } accuracy is 68.00%

Using feature(s) { 1 2 3 4 5 6 7 9 } accuracy is 69.00%

Feature set { 2 3 4 5 6 7 9 10 } was best, accuracy is 73.00%

Using feature(s) { 3 4 5 6 7 9 10 } accuracy is 64.00%

Using feature(s) { 2 4 5 6 7 9 10 } accuracy is 68.00%

Using feature(s) { 2 3 5 6 7 9 10 } accuracy is 67.00%

Using feature(s) { 2 3 4 6 7 9 10 } accuracy is 66.00%

Using feature(s) { 2 3 4 5 7 9 10 } accuracy is 74.00%

Using feature(s) { 2 3 4 5 6 9 10 } accuracy is 71.00%

Using feature(s) { 2 3 4 5 6 7 10 } accuracy is 76.00%

Using feature(s) { 2 3 4 5 6 7 9 } accuracy is 74.00%

Feature set { 2 3 4 5 6 7 10 } was best, accuracy is 76.00%

Using feature(s) { 3 4 5 6 7 10 } accuracy is 68.00%

Using feature(s) { 2 4 5 6 7 10 } accuracy is 77.00%

Using feature(s) { 2 3 5 6 7 10 } accuracy is 75.00%

Using feature(s) { 2 3 4 6 7 10 } accuracy is 68.00%

Using feature(s) { 2 3 4 5 7 10 } accuracy is 80.00%

Using feature(s) { 2 3 4 5 6 10 } accuracy is 79.00%

Using feature(s) { 2 3 4 5 6 7 } accuracy is 77.00%

Feature set { 2 3 4 5 7 10 } was best, accuracy is 80.00%

Using feature(s) { 3 4 5 7 10 } accuracy is 72.00%

Using feature(s) { 2 4 5 7 10 } accuracy is 79.00%

Using feature(s) { 2 3 5 7 10 } accuracy is 79.00%

Using feature(s) { 2 3 4 7 10 } accuracy is 69.00%

Using feature(s) { 2 3 4 5 10 } accuracy is 82.00%

Using feature(s) { 2 3 4 5 7 } accuracy is 78.00%

Feature set { 2 3 4 5 10 } was best, accuracy is 82.00%

Using feature(s) { 3 4 5 10 } accuracy is 79.00%

Using feature(s) { 2 4 5 10 } accuracy is 71.00%

Using feature(s) { 2 3 5 10 } accuracy is 77.00%

Using feature(s) { 2 3 4 10 } accuracy is 68.00%

Using feature(s) { 2 3 4 5 } accuracy is 83.00%

Feature set { 2 3 4 5 } was best, accuracy is 83.00%

Using feature(s) { 3 4 5 } accuracy is 82.00%

Using feature(s) { 2 4 5 } accuracy is 77.00%

Using feature(s) { 2 3 5 } accuracy is 80.00%

Using feature(s) { 2 3 4 } accuracy is 67.00%

Feature set { 3 4 5 } was best, accuracy is 82.00%

Using feature(s) { 4 5 } accuracy is 76.00%

Using feature(s) { 3 5 } accuracy is 92.00%

Using feature(s) { 3 4 } accuracy is 67.00%

Feature set { 3 5 } was best, accuracy is 92.00%

Using feature(s) { 5 } accuracy is 75.00%

Using feature(s) { 3 } accuracy is 68.00%

Feature set { 5 } was best, accuracy is 75.00%

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

PS C:\Users\Clem\Desktop\local code\170\proj2\output>