

Siyi Zhan
June 4, 2020
CS170 Project 2

Challenges

In this project, my main challenge is time management. During this special time, all finals and reports are taken home. Therefore, it's hard to arrange time appropriately. However, technically, I had a hard time declaring the normalization function. After searching online, I chose a z-normalization. Reference will be attached. This time the algorithm itself is no longer hard as Project1.

Design (objects and methods)

My whole program is designing as:

Three Algorithm functions:

Nearest_neighbor

Forward_selection

Backward_selection

Two Helper functions:

Normalization

Cal_accuracy(Using leaving one out evaluation)

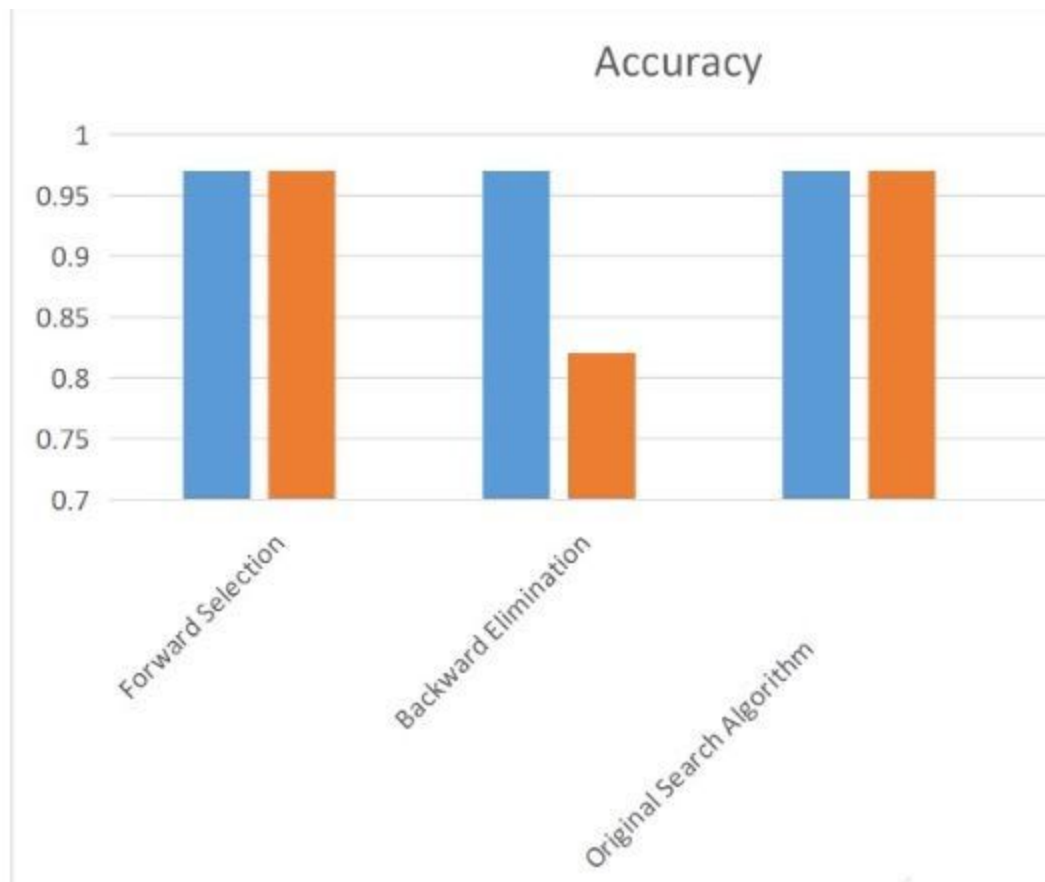
Also, a basic GUI design for users in process()

Code Optimization

I did import numpy arrays, which takes significantly less amount of memory compared to lists. It takes much less memory to store data, especially for this time when we have large data in test files. Therefore, numpy arrays can be my example to use special data structures to optimize my code.

Plots for features and analysis

- If you implemented more than one feature search algorithm (optional)
 - o comparison of different algorithms and discussion.
 - o You might want to compare running times for different search algorithms.



Elapsed Time(seconds)			
	Forward Selection	Backward Elimination	Original Search Algorithm
cs_170_small102	0.412801	0.274962	0.236899
cs_170_large102	9.919276	20.596999	4.820169

Analysis

The forward search algorithm returns the most accurate than the backward selection and leaving one out evaluation algorithm. My original algorithm turns out to be fastest!

References

https://jmotif.github.io/sax-vsm_site/morea/algorithm/znorm.html