Bryce Hills

SID: 8620730105

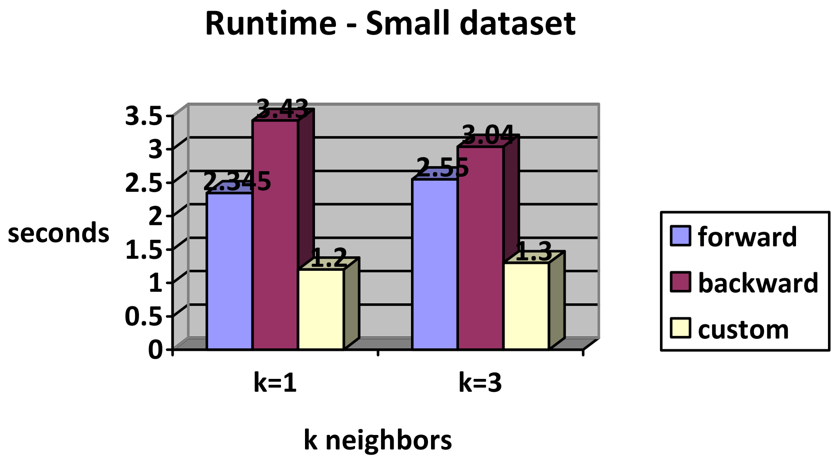
CS 170 Sect #021

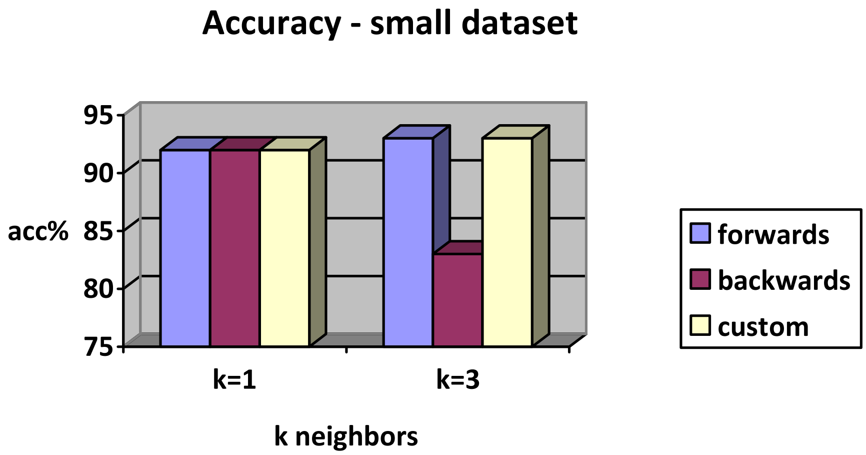
A.I. Project 2 Report

- Project Design:
  - **Feature selection**: To iterate and find the best features, I used two different algorithms
    - 1. Forward Selection – Begins with an empty set of features and adds the to a list determined by maximum score. The best score and corresponding subset of features ae saved and output by the algorithm.
    - 2. Backwards Elimination – Begins with a full set of features and iterates though each subset of features to find the best possible subset. Each iteration is scored and the best subset is output.
    - 3. Custom Algorithm – uses numba to optimize the calculation time of Euclidean distance. It still uses forward selection but performs 8 times faster than the normal forward selection, as shown in the charts on the next page. The results are the same as FS, but the runtime is significantly less.
  - **Leave One Out**:
    - This algorithm looks at compares all instances with one another to find the closest neighbor given a certain set of features. It uses Euclidean distance and KNN to determine this
    - K Nearest Neighbor – implemented within the Cross one out function, a k value is passed in by the user to classify each instance using k nearest neighbors. The neighbors are all stored in a pre sorted list and the number of instances is counted and the class is determined by the more frequent class.
    - Euclidean Distance – takes 2 numbers and a set of features to calculate the distance between two instances
  - **Functions** – general functions that make up the rest of the code:
    - 1.Normalize – normalizes the data points within the 2-dimensional array.
    - 2. Driver – creates user UI to gather input from text file and information about with algorithm to use and the k for KNN.
- Challenges:
  - One of the main challenges was fully understanding the meaning of the files that were given to us. We are told that the columns are the features and the rows are the instances. It may seem obvious, but without any applied/concrete examples, the data in the files looks meaningless. Once I understood the meaning of the class column and the rest, writing the algorithms became much easier.
  - This project has a lot of indexing. This includes columns, rows, k neighbors, and subsets and features and instances. It can become confusing and cause run time errors. I think it is important to standardize indexing from the beginning and make output very clear.

- Slow run time – It is difficult to test and debug the large data sets since running takes so long, this is why it is important to find a way to optimize the code and make it faster.

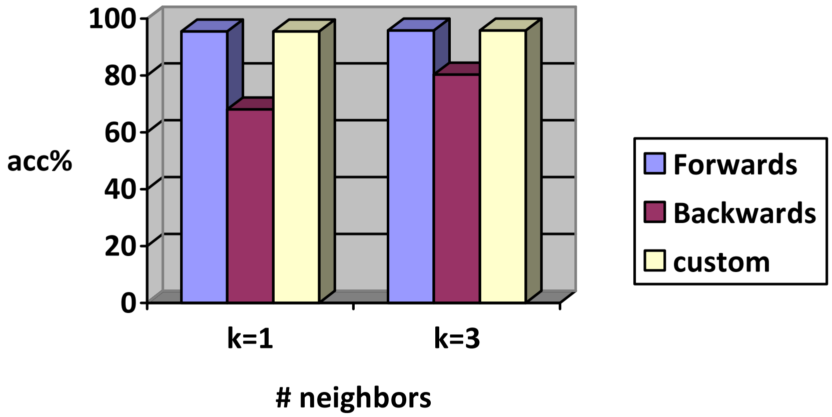
Small Dataset Results:



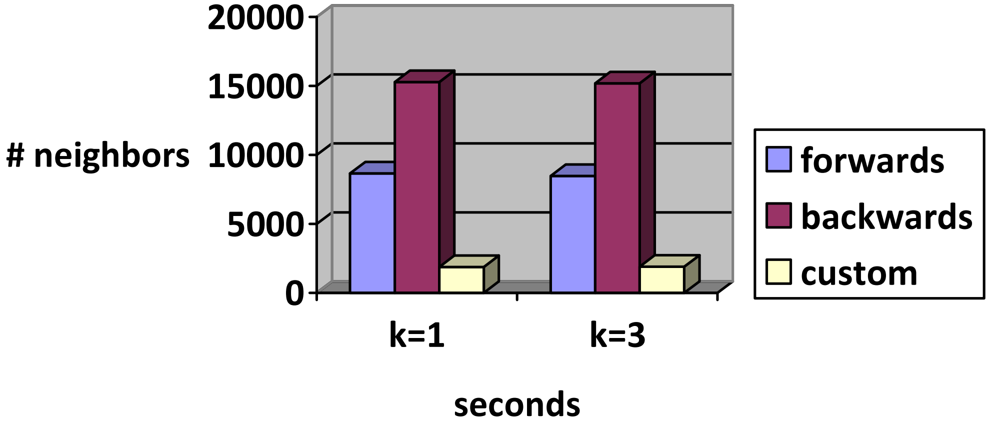


Large Dataset Results:

Accuracy Large dataset



Time - large dataset



Findings:

- 1. Forward selection seems to generate better results as far as time and accuracy are concerned. There are only 2 different data sets to test on, so it is difficult to say if this is true across the board, but given these 2 data sets, forward selection outperforms backward elimination
- The custom algorithm simply utilizes the numba library's @jit functionality to optimize the calculation time taken to use the Euclidean distance function. This simple addition to the code reduces the run time by 1/8.
- 2. It takes a very long time to run the large data set since I used python and have a linear implementation. It would be possible to increase our run time if we used multithreading or perhaps optimized the way we iterate/store features.
- 3. The best features that are consistently selected as good features are 0,4,5,7and 26. Note that my indexing starts at 0 rather than 1 so it may differ from other results by indexing.
- 4. It is very common in these data sets to have an accuracy range from 60-70%.

References:

- https://docs.scipy.org/doc/numpy/reference/generated/numpy.loadtxt.html
- https://www.geeksforgeeks.org/ml-multiple-linear-regression-backward-elimination-technique/
- https://stackoverflow.com/questions/1401712/how-can-the-euclidean-distance-be-calculated-with-numpy
- https://numba.pydata.org/numba-doc/dev/index.html
- https://stackoverflow.com/questions/46808362/how-to-use-numba-jit-with-methods
- https://stackoverflow.com/questions/2662140/how-to-measure-running-time-of-algorithms-in-python