Justin Pham

SID: 862114467

Email: [jpham079@ucr.edu](mailto:jpham079@ucr.edu)

Dec 9, 2021

Project 2 Machine Learning with Dr. Eamonn Keogh

My code is heavily based on the MatLab code provided by Dr. Keogh. All code is original except for functions from following Python libraries.

* **csv**, **os**, **pandas** and **numpy** were used to parse the provided text file
* **sys** and **math** were used to perform cross validation and calculate Euclidean distance respectively
* **time** was used to record how long the program took to run

I also used the Cython compiler to speed up my search. I used the template from this website to compile my Python code: <https://neurohackademy.github.io/high-performance-python/03-compiling/>. The code is compiled by running “python source/compile.py build\_ext --inplace” from the root directory. Run the code with “python run.py” from the root directory..

Link to my code: <https://github.com/Pjsrcool/CS-170-Project-2>

**Premise**

Dr. Keogh assigned each student a small data set and a large data set. Our task is to use Forward Selection and Backward Elimination to find the most relevant features in the data sets.

**Results for Small Data**

I was assigned Ver\_2\_CS170\_Fall\_2021\_Small\_data\_\_91.txt.

In Figure 1 below, we see the results of running Forward Selection on the Small data. For the empty set, I reported the Default rate, which is 84.0%. The Greedy Search started by selecting [6] with an accuracy of 86.2%. The next iteration finds features [6,2], then it finds [6,2,5], and so on. After all the search iterations, we find that the features [6,2] would be the most accurate at 97.4 % accuracy. I believe it may be over-fit. Therefore, I believe features [6,2,5] to be the best features, with accuracy of 95.8 %. It is only 1.6% less accurate, but has more features. This means that it maintains basically the same accuracy while using more features, so there is less chance if it being an over-fit.

In Figure 2 below, we see the results of running Backward Elimination on the Small data. I started by reporting the accuracy using All Features, which is 79.0%. Then after removing Feature 1, we have an improved accuracy of 81.2%. After all the iterations complete, I find that we have the same results as Forward Selection, which is features [2,5,6] is the best answer with 95.8%, and that features [2,6] is most likely and over-fit with an accuracy of 97.4%.

**Conclusion for Small Data:**

My final answer for the Small Data set is that features [2,5,6] is the best.Using features [2,6] may result in an over-fitting because it is marginally more accurate. It may make a really good model for the data we have, but it may also perform worse when deployed in the real world.

**Results for LARGE Data**

I was assigned Ver\_2\_CS170\_Fall\_2021\_LARGE\_data\_\_91.txt.

In Figure 3 below, we see the results of running Forward Selection on the LARGE Data.