Assignment 2 **CS170: Introduction to Artificial Intelligence, Dr. Eamonn Keogh**

Cruz Ramirez
SID 862175850
Email crami119@ucr.edu
Date Dec-8-22

In completing this assignment I consulted:

- https://www.tutorialspoint.com/list-methods-in-python-in-not-in-len-min-max
- https://www.geeksforgeeks.org/python-infinity/
- https://www.dropbox.com/sh/rltooq0t3khobuj/AAA3MYkZc8gb1RLa3tNSnsrga?dl=0
  Dr. Eamonn Keogh Lecture Materials
- My colleague and close friend Karan Bhogal

All important code is original. Unimportant sorting and helper functions are

- Python standard Library
- Numpy python library

Outline of this report:

- Cover page:(this page)
- My report: Pages 2-5
- Sample trace of small dataset Forward selection: Page 6
- URL to GitHub with access to code: https://github.com/Qrooz/CS170_Project2

CS 170: Assignment 2: Feature Selection with Nearest Neighbor
Cruz Ramirez, SID 862175850 Dec-8-2022

Introduction:

This report details the application and results found of the Nearest Neighbor classification algorithm used in feature selection during Dr. Eamonn Keogh's Introduction to AI class at the University of California Riverside during the fall quarter of 2022. The Nearest Neighbor Classification algorithm works by first having a dataset of known object classes, such as trying to classify cats and dogs apart from each other, with corresponding features that are quantified such as weight and height. Now using a data set of known heights and weights that are known from measured cats and dogs, a new object can be classified by taking its height and weight (quantified features) and finding the nearest neighboring values of a known object in the dataset to compare with, thus classifying the unknown as the same type. An example of this would be knowing that an animal to be classified had a large weight and medium height, and referencing the dataset its nearest neighbor that has a close height and weight is a dog, which would classify our unknown object as a dog. However, one of the greatest weaknesses of the Nearest Neighbor classification algorithm is that it is very sensitive to irrelevant features, such as if you used the amount of legs a dog or cat has to classify them. An irrelevant feature is one that does not actually give any useful information for the classification of the object. The goal of this project is to be able to utilize the Nearest Neighbor Algorithm in conjunction with Leave One Out validation with either forward selection or backward elimination in order to narrow down which combinations of features results in the best accuracy, which is a method of circumventing the weakness of Nearest Neighbors.

Forward Selection:

Forward selection works by first starting the working set of features as empty, and only adding a feature to the working set if it is the greatest accuracy found at that level of searching. This means at the first level when there are no features in the working set, only the feature with the greatest accuracy of success from checking all the features is put in the working set. Then on level two, the feature that was chosen for the working set from level one is then combined with all the other features again to find the highest accuracy of two features. This process is repeated until there are no combinations of features to add to the working set, but only the working set with the highest accuracy of all the different combinations is kept. The working set that is received with the highest accuracy of correct classification may have any number of different features from the features tested, but most commonly only two to four features are truly relevant for classifications.

Conclusions for Forward Selection:

From running the assigned small dataset 113, the results can be observed in Figure 1 below. The highest accuracy of 96.8% was observed from the combination of features 3 and 4 which was found to be the best possible choices of features to use from this dataset, the next best being the additional feature 2 being added to the set and receiving an accuracy of 92.4% .

For running large dataset 122, the most crucial results can be found in Figure 2. A maximum accuracy of 96% was observed through using features 13 and 2 which are deemed as the best features to use for this set. An additional feature 17 was close to having the max accuracy, with a calculated 95.8%.

Backwards Elimination:

Backward elimination works oppositely from Forward selection such that the working set starts off containing every single feature in the dataset and proceeds on level one to remove one feature and test the accuracy, and whichever feature that is tested when removed that results in the highest accuracy is then removed permanently from the working set. This process is continued by removing one feature at each level until there are no more features left in the working set and whichever working set of features that resulted in the highest level of accuracy of all tests is taken as the features that are the best for this classification.

Conclusions for Backwards Elimination:

From running and documenting the results of backwards elimination with small dataset 113 in reference to Figure 3, the highest recorded accuracy was 96.8% using features 3 and 4 which is in agreement with the forward selection case, and it can be concluded that the best features for small dataset 113 from both forward selection and backward elimination are features 3 and 4.

Unfortunately from running backwards elimination on the large dataset 122, a highest recorded accuracy of 84% was obtained using more than half of the features given which is not in agreement with the forward selection results. The first and last few levels of the search are shown in Figure 4, and interestingly the last feature to be removed from the working set is feature 13 which is one of the relevant features that was kept in Forward Selection.

Charts and Datatables:

Figure 1: Chart depicts the accuracy percentage of each combination of features selected at each level using the small dataset 113 and Forward Selection, default rate assumed to be 50%

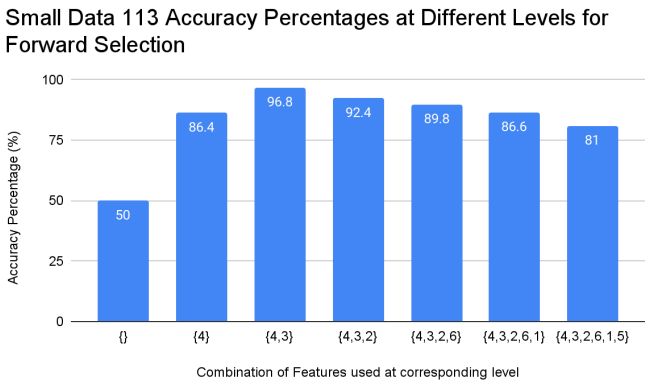

Figure 2: Chart depicts the accuracy percentage of each combination of features selected at each level using the large dataset 122 and Forward Selection, gap between {13,2,17} to ending sets for brevity, {...} denotes a current set too large to show on axis, default rate assumed to be 50%

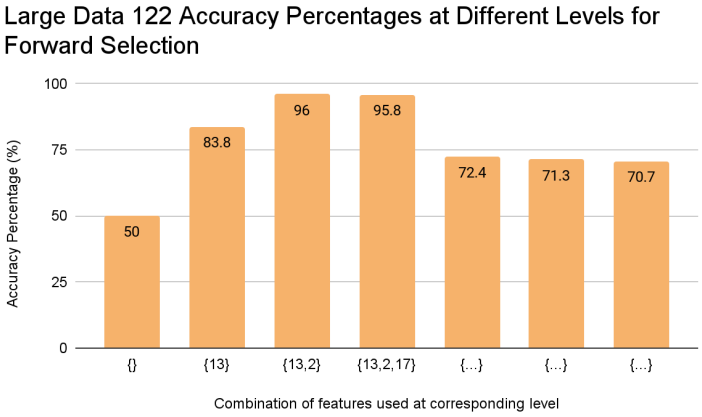

Figure 3: Chart depicts the accuracy percentage of each combination of features selected at each level using the small dataset 113 and Backward Elimination

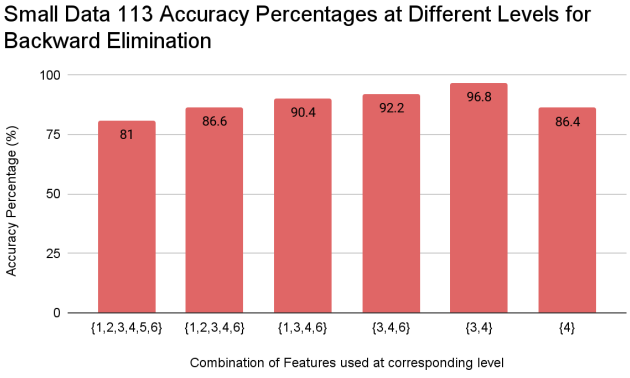

Figure 4: Chart depicts the accuracy percentage of each combination of features selected at each level using the large dataset 122 and Backward Elimination, gap between third {...} and {13,38,25} for brevity, {...} denotes a current set too large to show on chart axis

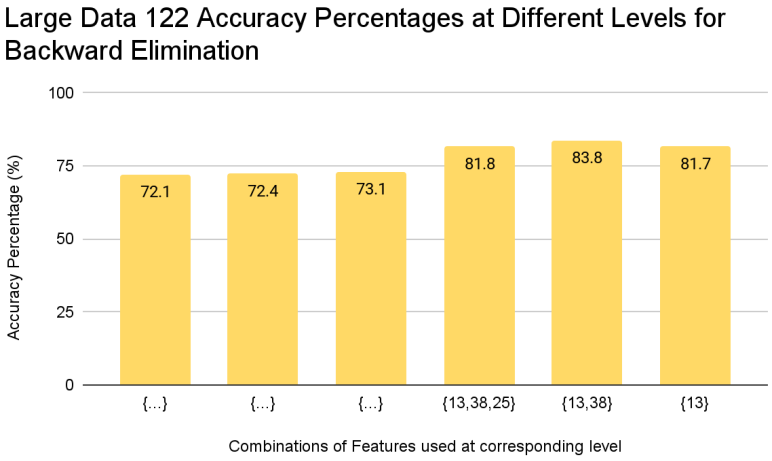

Figure 5: Table Depicts the running times of the above instances of both forward selection and backward elimination with both the small dataset 113 and large dataset 122 in minutes

	Small Dataset113	Large Dataset 122
Forward Selection	0.22 min	144.4 min
Backward Elimination	0.21 min	150.5 min

Conclusion:

Through the completion of this project, it is shown that the Nearest Neighbor algorithm is a powerful classification algorithm, and its weakness against irrelevant features can be compensated by weeding them out through leave one out validation and possibly many other types of validation. Through personal testing and running of the algorithm, the best features found for small dataset 113 were features 3 and 4, whereas the best for large dataset 122 were features 13 and 2. It can be deduced that between Forward Selection and Backwards Elimination that Forward selection is the better option because it adds the best features onto the set very early in its runtime and commonly classification problems generally rely on two to four relevant features in order to achieve an extremely high accuracy. Through looking at the runtimes displayed in Figure 5, Backwards Elimination can only get worse and worse with larger datasets when trying to find only a couple of features that are relevant enough to make big changes to the accuracy of classification.

The following is a traceback of Forward Selection on Small Dataset 113:

```
Welcome to Cruz Ramirez's Forward Feature Selection Algorithm.
Type in a '1' for searching the small dataset 113 or a '2' for searching the large dataset 122:
1
On the  1 th level of the search tree
--Considering adding the  1  feature
Returned with an Accuracy of  0.746
--Considering adding the  2  feature
Returned with an Accuracy of  0.748
--Considering adding the  3  feature
Returned with an Accuracy of  0.776
--Considering adding the  4  feature
Returned with an Accuracy of  0.864
--Considering adding the  5  feature
Returned with an Accuracy of  0.75
--Considering adding the  6  feature
Returned with an Accuracy of  0.724
On level  1 I added feature  4  to current set
On the  2 th level of the search tree
--Considering adding the  1  feature
Returned with an Accuracy of  0.854
--Considering adding the  2  feature
Returned with an Accuracy of  0.84
--Considering adding the  3  feature
Returned with an Accuracy of  0.968
--Considering adding the  5  feature
Returned with an Accuracy of  0.836
--Considering adding the  6  feature
Returned with an Accuracy of  0.846
On level  2 I added feature  3  to current set
On the  3 th level of the search tree
--Considering adding the  1  feature
Returned with an Accuracy of  0.91
--Considering adding the  2  feature
Returned with an Accuracy of  0.924
--Considering adding the  5  feature
Returned with an Accuracy of  0.914
--Considering adding the  6  feature
Returned with an Accuracy of  0.922
On level  3 I added feature  2  to current set
On the  4 th level of the search tree
--Considering adding the  1  feature
Returned with an Accuracy of  0.89
--Considering adding the  5  feature
Returned with an Accuracy of  0.86
--Considering adding the  6  feature
Returned with an Accuracy of  0.898
On level  4 I added feature  6  to current set
On the  5 th level of the search tree
--Considering adding the  1  feature
Returned with an Accuracy of  0.866
--Considering adding the  5  feature
Returned with an Accuracy of  0.84
On level  5 I added feature  1  to current set
On the  6 th level of the search tree
--Considering adding the  5  feature
Returned with an Accuracy of  0.81
On level  6 I added feature  5  to current set
The best accuracy is  0.968 with features  [4, 3]
PS C:\Users\Cruz\OneDrive\Desktop\CS170_Proj2>
```

##URL to GitHub with project code: https://github.com/Qrooz/CS170_Project2