CS205: Artificial Intelligence, Dr. Eamonn Keogh

Project 2

Name: Princekumar Manishbhai Savsaviya

SID: 862548420

E-Mail: psavs001@ucr.edu

Date: 06/07/2025

Preface

About Code:

- Language: Python 3.12
- Libraries (subroutines) used:
    - NumPy
    - time
    - matplotlib
    - sklearn
    - sys
    - os
- All Other Code are my Original.
- Note :: All the index in graphs are zero indexed.

Material Used:

- Lecture slides

Table of Contents

CS205: Project 2

Part 1

Introduction

In this project, We are tasked with implementing two search methods

1) Forward Selection
2) Backward Elimination

for finding the best feature groups among the given features. We are asked to use the nearest-neighbor algorithm to train the model on the dataset.

Algorithms

I) Forward Selection

It is a greedy feature‑selection technique that starts with an empty set of predictors and then repeatedly adds the single feature whose inclusion most improves a chosen model's performance (for example, highest cross‑validation accuracy). At each step, it evaluates the model using each candidate feature not yet selected, picks the one that gives the largest performance gain, and adds it permanently. This process continues until no remaining feature can further improve the model, yielding a compact subset of variables tailored to maximize predictive accuracy.

II) Backward Elimination

It is a greedy feature‑selection method that begins with the full set of all candidate variables and then repeatedly removes the single feature whose exclusion most improves—or least degrades—a chosen model's performance (for example, highest cross‑validation accuracy). At each iteration, it evaluates the model without each remaining feature in turn, identifies which removal yields the best result, and permanently drops that feature. This process continues until removing any additional variable would worsen the model, leaving a streamlined subset optimized for predictive power.

Results

I tested the above algorithms on two datasets assigned to me CS205_small_Data__22.txt and CS205_large_Data__18.txt, below are the results I obtained from the algorithm.

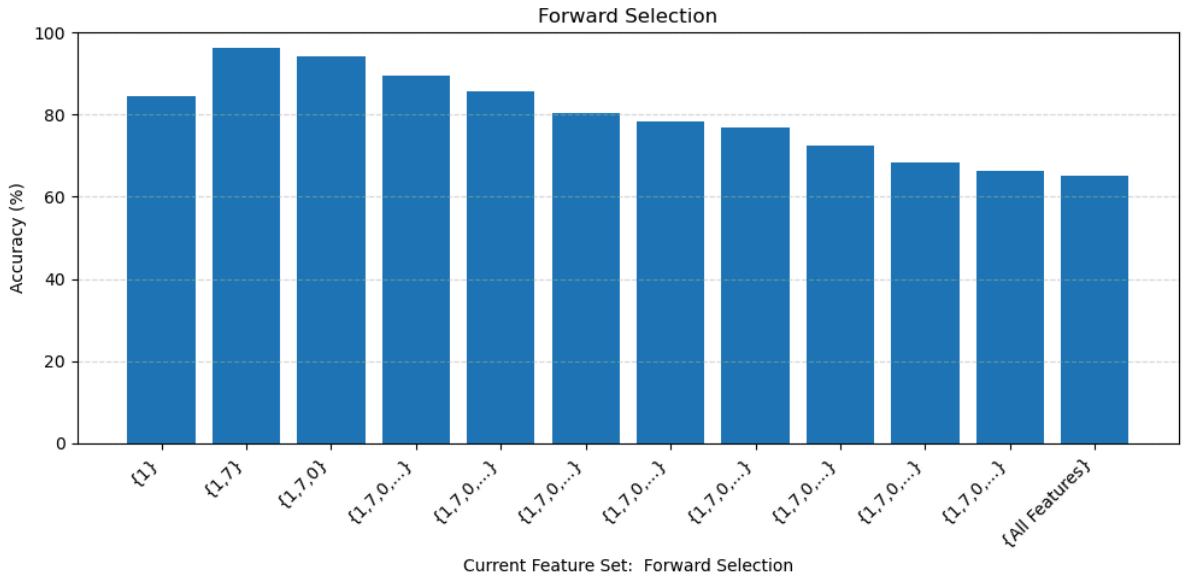


Figure 1 : Accuracy of increasingly largesubsets of features discovered by forward selection.

Figure 1 is the result of running the forward selection on the small dataset, which has 500 instances and 12 features. Default rate of this dataset is around 82.4%. Adding the feature 1 improved the accuracy to 84.4%. and adding another feature 7 to it increases the accuracy to 96.2%, but when third feature 0 is added, accuracy dropped to 94% and it keeps decreasing as we add the new features and at last with all-features we got the accuracy 65.20%.

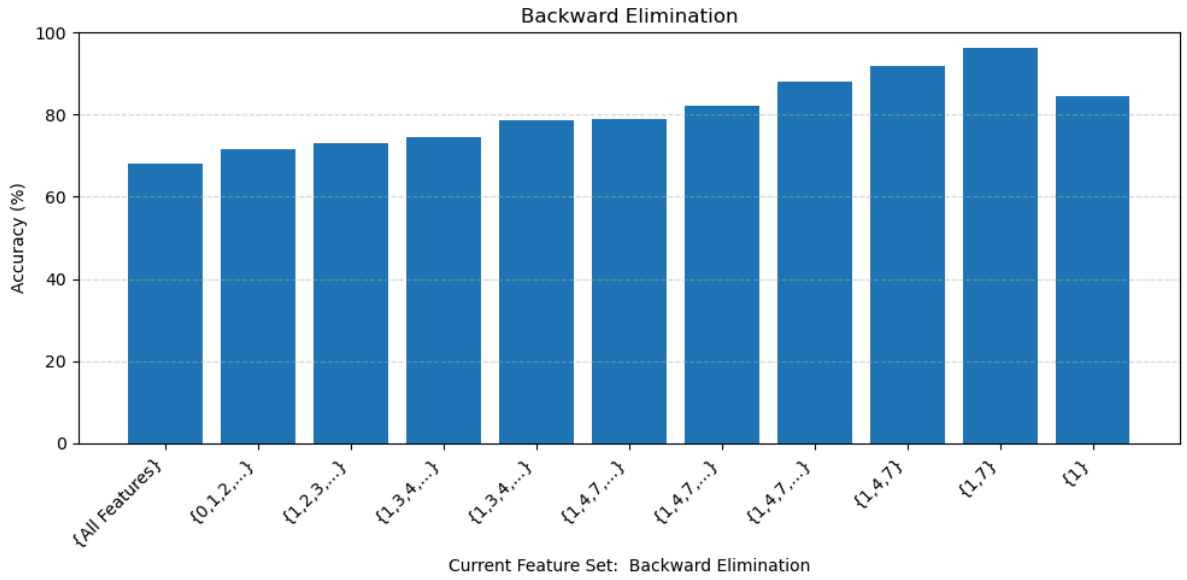


Figure 2 : Accuracy of increasingly small subsets of features discovered by backward Elimination

Figure 2 represents the backward elimination, in this I started with all the features and got the accuracy of 65.20%, and as we remove the features from the full feature set, accuracy start to improve and hit the maximum value of 96.2% with two features {1,7} and then again started to decrease as features are being removed.

Conclusion for small dataset :: From above two results, I come to the conclusion that feature {1,7} is the most useful feature as they give the highest possible accuracy in both of the algorithms.

For the Large Dataset, Figure 3 and Figure 4 are given below. Large Dataset consist of 50 features and 1000 samples.

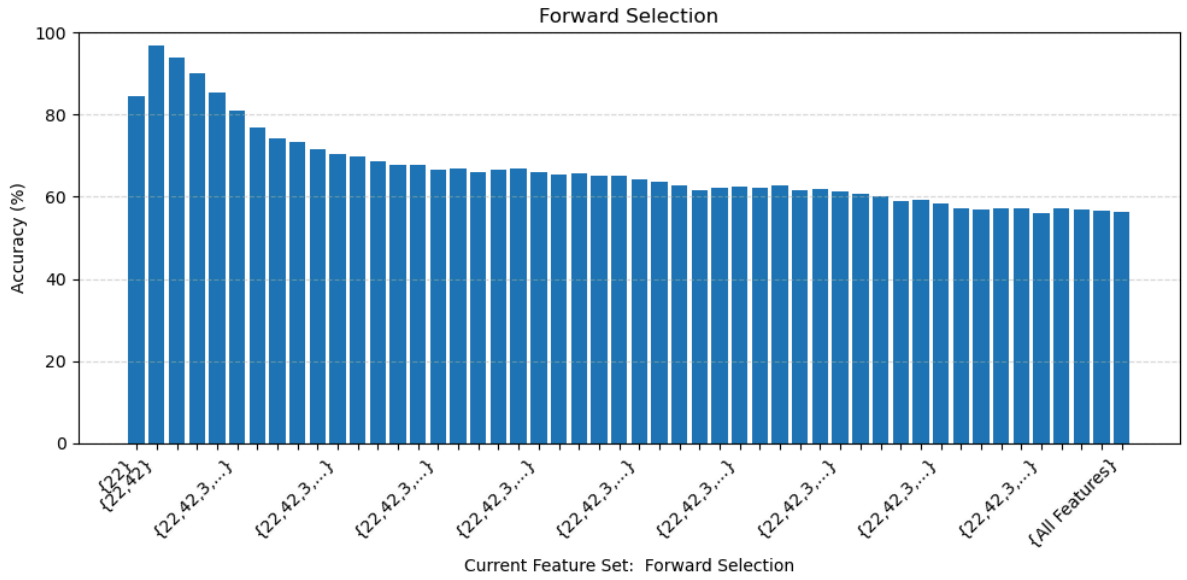


Figure 3 : Accuracy of increasingly large subsets of features discovered by forward selection

Figure 3, is for the forward selection on the large dataset, and as feature 22 is added to the selected feature set, accuracy improved to 84.6% from 79.40% and when second feature 42 is added, accuracy drastically improved to 96.9%, but when more features are added then accuracy started to decline form 96.9% to around 58% when all the features are considered.

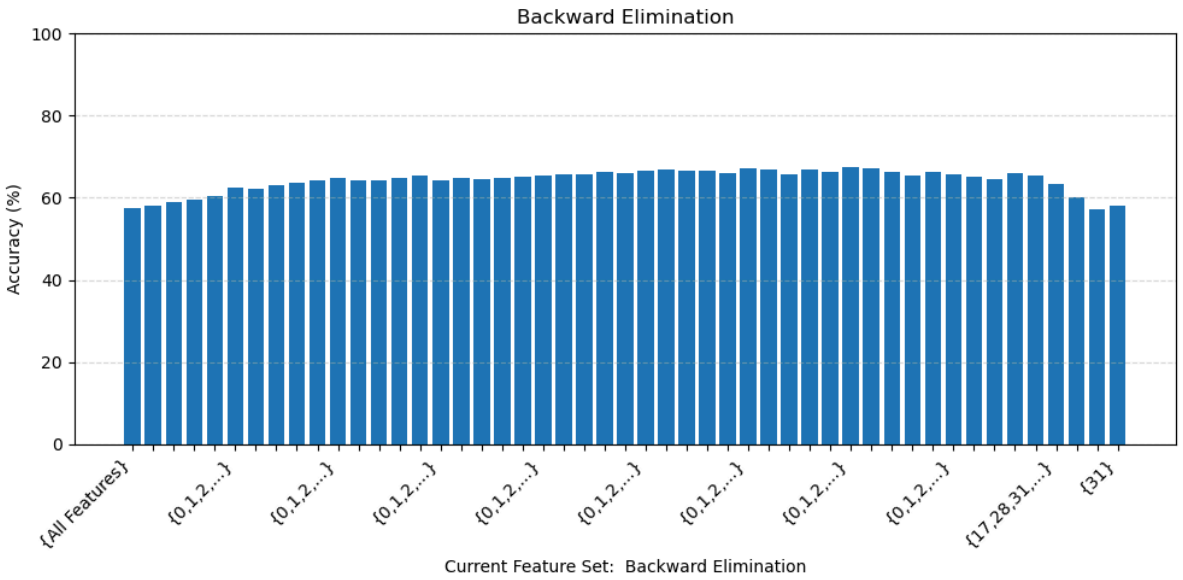


Figure 4 : Accuracy of increasingly small subsets of features discovered by backward Elimination

For large dataset, backward elimination gives completely different results than the forward selection, as shown in Figure 4, accuracy started almost at 58% for all 50 features and started gradually increasing with some irregularities in

between as features are removed and accuracy peaked around 64% when there are 20 features and then again started decreasing which is still worse than the default rate.

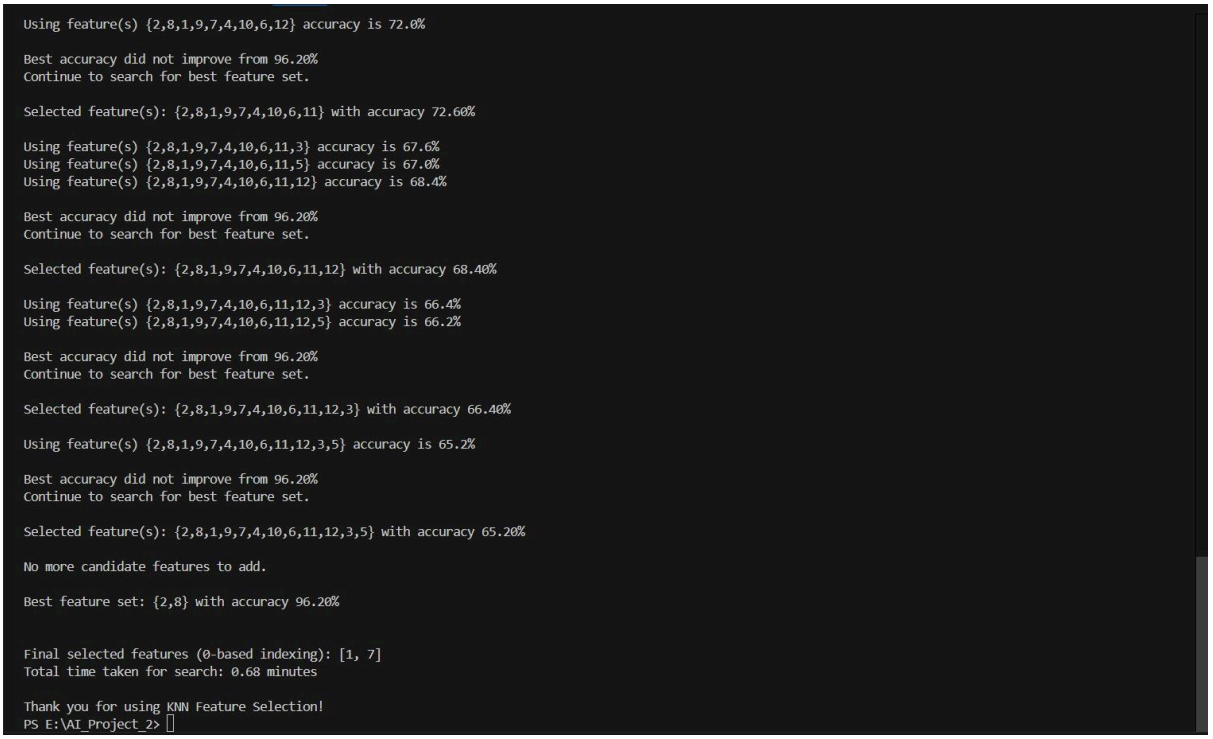
Conclusion for large dataset : So from both the results, I believe that feature {22,42} are the most relevant feature and for this specific dataset backward elimination has kind of no effect on selection of features as it gives accuracy lower than the default rate.

Computational Efforts : I run the algorithms on intel i7-10th generation CPU with 32 GB memory.

	Forward Selection	**Backward Elimination**
Small Dataset	0.6 min	0.59 min
Large Dataset	14.47 min	15.02 min

Working of my Code ::

```
PS E:\AI_Project_2> & E:/anaconda3/python.exe e:/AI_Project_2/KNN.py
Welcome to KNN Feature Selection!
Please enter the data file name (e.g., CS205_small_Data__22.txt): CS205_small_Data__22.txt
Select algorithm
 1) forward selection
 2) backward elimination
 Enter 1 or 2 :: 1
This Dataset has 12 features and 500 samples.

 You can press 'q' at any time to quit the search early. Search ends after the completion of current iteration.

Baseline accuracy with 12 features: 65.20%

Starting forward selection...
Using feature(s) {1} accuracy is 62.6%
Using feature(s) {2} accuracy is 84.4%
Using feature(s) {3} accuracy is 56.0%
Using feature(s) {4} accuracy is 56.2%
Using feature(s) {5} accuracy is 60.6%
Using feature(s) {6} accuracy is 62.0%
Using feature(s) {7} accuracy is 57.8%
Using feature(s) {8} accuracy is 69.6%
Using feature(s) {9} accuracy is 64.2%
Using feature(s) {10} accuracy is 59.0%
Using feature(s) {11} accuracy is 64.0%
Using feature(s) {12} accuracy is 57.4%

Selected feature(s): {2} with accuracy 84.40%

Using feature(s) {2,1} accuracy is 80.8%
Using feature(s) {2,3} accuracy is 82.6%
Using feature(s) {2,4} accuracy is 80.6%
Using feature(s) {2,5} accuracy is 77.6%
Using feature(s) {2,6} accuracy is 82.6%
Using feature(s) {2,7} accuracy is 81.4%
Using feature(s) {2,8} accuracy is 96.2%
Using feature(s) {2,9} accuracy is 82.6%
Using feature(s) {2,10} accuracy is 80.6%
Using feature(s) {2,11} accuracy is 80.2%
Using feature(s) {2,12} accuracy is 77.8%
```



Note :: Here's the demo of a small dataset on forward selection. In this intermediate steps are skipped due to more number of images but starting and ending results of forward selection on small dataset is fully shown.

Part 2

Introduction

In this part, I am asked to implement forward selection on any of the dataset of my choosing, so I choose the Wine Quality Dataset from the UCI's machine learning Repository.

This dataset is real numbered, with 11 features and 4898 instances.

Result

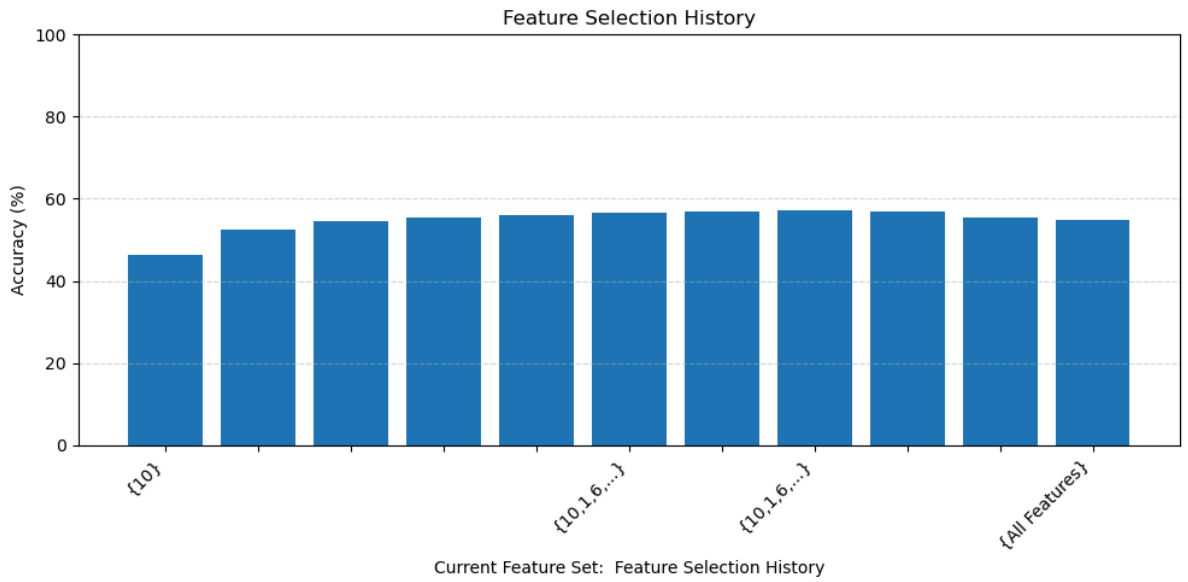


Figure 5 : Accuracy of increasingly large subsets of features discovered by forward selection on wine dataset

For the wine dataset, default rate is around 43.08%, selection of one feature gives us the accuracy of 46.32% and as we add more features, accuracy keeps increasing till 8 features are included then again accuracy started to decrease very slowly. So with 8 features we got the best accuracy which is 57.15%.

Conclusion for Wine Dataset : First feature which is considered is the alcohol level in the wine, second feature picked by forward selection is volatile acidity and the third feature picked is total sulfur dioxide which is the most important feature according to Kurtanjek, Ž [3], so our algorithm works fine.

Working Code ::

```
[Running] python -u "e:\AI_Project_2\Wine-Quality.py"
e:\AI_Project_2\Wine-Quality.py:28: SyntaxWarning: invalid escape sequence '\w'
  red_wine = pd.read_csv('wine+quality\winequality-red.csv', sep=';')
e:\AI_Project_2\Wine-Quality.py:29: SyntaxWarning: invalid escape sequence '\w'
  white_wine = pd.read_csv('wine+quality\winequality-white.csv', sep=';')
Default classification rate (most common class): 6 (43.08%)

Baseline accuracy with 11 features: 54.76%

Starting forward selection...
Using feature(s) {1} accuracy is 36.3%
Using feature(s) {2} accuracy is 44.6%
Using feature(s) {3} accuracy is 39.2%
Using feature(s) {4} accuracy is 42.6%
Using feature(s) {5} accuracy is 42.5%
Using feature(s) {6} accuracy is 39.5%
Using feature(s) {7} accuracy is 40.9%
Using feature(s) {8} accuracy is 45.1%
Using feature(s) {9} accuracy is 39.3%
Using feature(s) {10} accuracy is 39.0%
Using feature(s) {11} accuracy is 46.3%

Selected feature(s): {11} with accuracy 46.32%

Using feature(s) {11,1} accuracy is 49.4%
Using feature(s) {11,2} accuracy is 52.4%
Using feature(s) {11,3} accuracy is 48.9%
Using feature(s) {11,4} accuracy is 51.1%
Using feature(s) {11,5} accuracy is 49.9%
Using feature(s) {11,6} accuracy is 52.3%
Using feature(s) {11,7} accuracy is 49.8%
Using feature(s) {11,8} accuracy is 50.6%
Using feature(s) {11,9} accuracy is 50.3%
Using feature(s) {11,10} accuracy is 49.0%

Selected feature(s): {11,2} with accuracy 52.43%
```

```
Best accuracy did not improve from 57.15%
Continue to search for best feature set.

Selected feature(s): {11,2,7,10,4,3,6,5,1,8} with accuracy 55.55%

Using feature(s) {11,2,7,10,4,3,6,5,1,8,9} accuracy is 54.8%

Best accuracy did not improve from 57.15%
Continue to search for best feature set.

Selected feature(s): {11,2,7,10,4,3,6,5,1,8,9} with accuracy 54.76%

No more candidate features to add.

Best feature set: {11,2,7,10,4,3,6,5} with accuracy 57.15%

Test accuracy with selected features: 55.69%
```

URL To My Code :: https://github.com/princesavsaviya/AI_Project_2

References

[1] Numpy Documentation :: https://numpy.org/doc/

[2] Sklearn Documentation :: https://scikit-learn.org/0.21/documentation.html

[3] Kurtanjek, Ž. (2023). *Wine quality analysis by the structural causal model (SCM)*. *Croatian Journal of Food Science and Technology, 15*(2), 173–184. https://doi.org/10.17508/CJFST.2023.15.2.05

Dataset

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Wine Quality [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T.