University of California, Riverside

CS205-
Artificial Intelligence

Winter 2024

CS205 Assignment 2: Feature Selection.

05/07/2024

Name	SID	Email	Group Number
Parth Shinde	862466119	pshin022@ucr.edu	-

I consulted the following in completing this assignment:

- CS 205 Lecture slides on ***Project-2 breifing*** by Dr. Eamonn Keogh.
- I have used the following non-trivial libraries for doing the assignment:-
    - **NumPy :** Used for loading the dataset and performing array manipulations and calculations.
    - **Sklearn:** Used for implementing K-Nearest Neigbor algorithm,finding the accuracy score and the also evaluating/validating the algorithm using LeaveOneOut algorithm.
    - **Pandas:** Used in the helper scatter-plot generation script.
    - **Matplotlib:** Used for generating the plots.
- I have used other trivial python-in built libraries that are present in the python3.9.
- I also consulted the following links for understanding sklearn and the KNN classifier implementation.
    - https://machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/
    - https://python-course.eu/machine-learning/k-nearest-neighbor-classifier-with-sklearn.php

List of contents:

Introduction

In this assignment we must implement a feature selection algorithm to find the most optimal feature set for the dataset considering, we use the K-nearest neighbor classifier as our baseline model. The two algorithms given to us to implement are Forward selection and Backward elimination.

Description of Datasets

The datasets given to us have the label in the first column of the dataset and all the other columns are the features given to us. We load this dataset using the numpy library and separate out the features and labels and pass that out to our classifier. The table below shows the shape of the dataset and some information about the dataset.

Dataset	No of Features	No of instances or rows
Small Dataset(2 labels)	12	500
Large Dataset(2 labels)	50	5000

K-Nearest-Neigbor Algorithm

K-Nearest Neighbors (KNN) is a simple, non-parametric, instance-based learning algorithm that stores training instances and makes predictions based on them at prediction time, using distance metrics like Euclidean or Manhattan to find the nearest neighbors. The parameter ***k*** determines the number of neighbors to consider: a small ***k*** captures local structure, while a large ***k*** provides more stability. For classification, KNN uses majority voting among the k-nearest neighbors to determine the class, and for regression, it predicts values based on the average or weighted average of the k-nearest neighbors. The algorithm involves storing all training data, calculating distances from the test instance to all training instances, sorting to select the k-nearest neighbors, and making predictions based on these neighbors.

Forward Selection

It is a greedy algorithm, which has a top to bottom approach to solving the search problem. It starts with no features (empty set of features) and iteratively adds the feature that improves the accuracy the most, until no more improvements are found.

Backward Elimination

Similar to forward selection, it is a greedy algorithm, but it’s a bottom to up approach. It starts with all features and iteratively removes the feature whose removal improves the accuracy the most, until no more improvements are found.

Algorithm

- A general feature search algorithm (the greedy problems) follow the algorithm given below
- Initialize a feature set (empty or full in our case)
- Initialize a the initial best accuracy of the model to a baseline value(e.g baseline accuracy could be :-using all the features given.)
- While the stopping condition is not met:

```
For each feature in the feature set to consider:
    Modify the current feature set by adding or removing the feature.
    Evaluate the model's performance (accuracy) using the modified feature set.
    If the accuracy improves:
        Update the best accuracy.
        Update the best feature set.
        (add the best feature or remove the worst feature in feature set).

If the best feature set did not change during this iteration:
    Terminate the search.
```

- Return the best feature set and its corresponding accuracy.

Timing Analysis

According to the algorithm given above, the Worst-Case Time Complexity would approximately be O(n^2 * m * T).Considering (m,T) to be the validation and time complexity of Classifier, respectively i.e in our case the KNN algorithm.

The code can be accessed here

The tracebacks are given as follows, and the user inputs are highlighted in yellow, and the final answer is green.

1. *Small Data and Forward Selection*

```
Type in the name of the file to test below:-CS205_small_Data__34.txt
Welcome to Parth's Feature Selection Algorithm.
Type the number of the algorithm you want to run.
```

```
         1.Forward Selection
         2.Backward Elimination
1

This dataset has 12 features (not including the class attribute), with 500
instances.
Running nearest neighbor with all 12 features, using “leaving-one-out”
evaluation, I get an accuracy of 73.4%
Beginning search.
Using feature(s) {0} accuracy is 68.8%
Using feature(s) {1} accuracy is 85.6%
Using feature(s) {2} accuracy is 69.0%
Using feature(s) {3} accuracy is 70.2%
Using feature(s) {4} accuracy is 69.8%
Using feature(s) {5} accuracy is 69.0%
Using feature(s) {6} accuracy is 68.8%
Using feature(s) {7} accuracy is 67.6%
Using feature(s) {8} accuracy is 72.0%
Using feature(s) {9} accuracy is 70.4%
Using feature(s) {10} accuracy is 71.0%
Using feature(s) {11} accuracy is 74.2%
Using feature(s) {0, 1} accuracy is 81.2%
Using feature(s) {1, 2} accuracy is 84.4%
Using feature(s) {1, 3} accuracy is 83.8%
Using feature(s) {1, 4} accuracy is 83.8%
Using feature(s) {1, 5} accuracy is 80.8%
Using feature(s) {1, 6} accuracy is 83.6%
Using feature(s) {1, 7} accuracy is 84.0%
Using feature(s) {8, 1} accuracy is 84.4%
Using feature(s) {1, 9} accuracy is 84.4%
Using feature(s) {1, 10} accuracy is 83.6%
Using feature(s) {1, 11} accuracy is 96.8%
Using feature(s) {0, 1, 11} accuracy is 92.4%
Using feature(s) {1, 2, 11} accuracy is 93.0%
Using feature(s) {3, 1, 11} accuracy is 94.2%
Using feature(s) {1, 11, 4} accuracy is 94.2%
Using feature(s) {1, 11, 5} accuracy is 92.6%
Using feature(s) {1, 11, 6} accuracy is 93.2%
Using feature(s) {1, 11, 7} accuracy is 94.6%
Using feature(s) {8, 1, 11} accuracy is 92.8%
Using feature(s) {1, 11, 9} accuracy is 93.2%
Using feature(s) {1, 10, 11} accuracy is 94.0%
Excuetion time using forward selection for CS205_small_Data__34.txt is:-
40.709142446517944
Graph saved as Forward Selection_CS205_small_Data__34.txt.pdf
Finished search!! The best feature subset is {1, 11}, which has an
accuracy of 96.8%
```

2. Small Data & Forward Selection

```
Type in the name of the file to test below:-CS205_small_Data__34.txt
Welcome to Parth's Feature Selection Algorithm.
Type the number of the algorithm you want to run.
         1.Forward Selection
         2.Backward Elimination
2
```

```
This dataset has 12 features (not including the class attribute), with 500
instances.
Running nearest neighbor with all 12 features, using “leaving-one-out”
evaluation, I get an accuracy of 73.4%
Beginning search.
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} accuracy is 73.6%
Using feature(s) {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} accuracy is 68.0%
Using feature(s) {0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11} accuracy is 74.6%
Using feature(s) {0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11} accuracy is 75.0%
Using feature(s) {0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11} accuracy is 70.8%
Using feature(s) {0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11} accuracy is 74.2%
Using feature(s) {0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11} accuracy is 74.6%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11} accuracy is 76.8%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11} accuracy is 73.6%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11} accuracy is 72.4%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11} accuracy is 72.6%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10} accuracy is 70.6%
Using feature(s) {1, 2, 3, 4, 5, 6, 8, 9, 10, 11} accuracy is 77.2%
Using feature(s) {0, 2, 3, 4, 5, 6, 8, 9, 10, 11} accuracy is 70.0%
Using feature(s) {0, 1, 3, 4, 5, 6, 8, 9, 10, 11} accuracy is 76.0%
Using feature(s) {0, 1, 2, 4, 5, 6, 8, 9, 10, 11} accuracy is 76.8%
Using feature(s) {0, 1, 2, 3, 5, 6, 8, 9, 10, 11} accuracy is 74.4%
Using feature(s) {0, 1, 2, 3, 4, 6, 8, 9, 10, 11} accuracy is 74.8%
Using feature(s) {0, 1, 2, 3, 4, 5, 8, 9, 10, 11} accuracy is 75.8%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 9, 10, 11} accuracy is 76.0%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 10, 11} accuracy is 75.8%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 11} accuracy is 75.4%
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10} accuracy is 71.8%
Using feature(s) {2, 3, 4, 5, 6, 8, 9, 10, 11} accuracy is 71.0%
Using feature(s) {1, 3, 4, 5, 6, 8, 9, 10, 11} accuracy is 79.0%
Using feature(s) {1, 2, 4, 5, 6, 8, 9, 10, 11} accuracy is 79.4%
Using feature(s) {1, 2, 3, 5, 6, 8, 9, 10, 11} accuracy is 78.0%
Using feature(s) {1, 2, 3, 4, 6, 8, 9, 10, 11} accuracy is 77.4%
Using feature(s) {1, 2, 3, 4, 5, 8, 9, 10, 11} accuracy is 79.0%
Using feature(s) {1, 2, 3, 4, 5, 6, 9, 10, 11} accuracy is 76.2%
Using feature(s) {1, 2, 3, 4, 5, 6, 8, 10, 11} accuracy is 76.0%
Using feature(s) {1, 2, 3, 4, 5, 6, 8, 9, 11} accuracy is 76.2%
Using feature(s) {1, 2, 3, 4, 5, 6, 8, 9, 10} accuracy is 72.8%
Using feature(s) {2, 4, 5, 6, 8, 9, 10, 11} accuracy is 69.2%
Using feature(s) {1, 4, 5, 6, 8, 9, 10, 11} accuracy is 79.0%
Using feature(s) {1, 2, 5, 6, 8, 9, 10, 11} accuracy is 80.4%
Using feature(s) {1, 2, 4, 6, 8, 9, 10, 11} accuracy is 78.2%
Using feature(s) {1, 2, 4, 5, 8, 9, 10, 11} accuracy is 78.0%
Using feature(s) {1, 2, 4, 5, 6, 9, 10, 11} accuracy is 81.4%
Using feature(s) {1, 2, 4, 5, 6, 8, 10, 11} accuracy is 80.2%
Using feature(s) {1, 2, 4, 5, 6, 8, 9, 11} accuracy is 78.0%
Using feature(s) {1, 2, 4, 5, 6, 8, 9, 10} accuracy is 74.8%
Using feature(s) {2, 4, 5, 6, 9, 10, 11} accuracy is 72.2%
Using feature(s) {1, 4, 5, 6, 9, 10, 11} accuracy is 82.6%
Using feature(s) {1, 2, 5, 6, 9, 10, 11} accuracy is 81.4%
Using feature(s) {1, 2, 4, 6, 9, 10, 11} accuracy is 81.6%
Using feature(s) {1, 2, 4, 5, 9, 10, 11} accuracy is 82.4%
Using feature(s) {1, 2, 4, 5, 6, 10, 11} accuracy is 81.6%
Using feature(s) {1, 2, 4, 5, 6, 9, 11} accuracy is 80.4%
Using feature(s) {1, 2, 4, 5, 6, 9, 10} accuracy is 76.8%
Using feature(s) {4, 5, 6, 9, 10, 11} accuracy is 72.8%
Using feature(s) {1, 5, 6, 9, 10, 11} accuracy is 83.6%
Using feature(s) {1, 4, 6, 9, 10, 11} accuracy is 85.4%
Using feature(s) {1, 4, 5, 9, 10, 11} accuracy is 84.8%
Using feature(s) {1, 4, 5, 6, 10, 11} accuracy is 82.2%
Using feature(s) {1, 4, 5, 6, 9, 11} accuracy is 83.0%
Using feature(s) {1, 4, 5, 6, 9, 10} accuracy is 81.2%
Using feature(s) {4, 6, 9, 10, 11} accuracy is 72.8%
Using feature(s) {1, 6, 9, 10, 11} accuracy is 84.8%
Using feature(s) {1, 4, 9, 10, 11} accuracy is 88.2%
Using feature(s) {1, 4, 6, 10, 11} accuracy is 88.6%
Using feature(s) {1, 4, 6, 9, 11} accuracy is 86.6%
Using feature(s) {1, 4, 6, 9, 10} accuracy is 80.6%
Using feature(s) {10, 11, 4, 6} accuracy is 74.8%
Using feature(s) {1, 10, 11, 6} accuracy is 88.8%
Using feature(s) {1, 10, 11, 4} accuracy is 90.6%
Using feature(s) {1, 11, 4, 6} accuracy is 91.8%
Using feature(s) {1, 10, 4, 6} accuracy is 82.8%
Using feature(s) {11, 4, 6} accuracy is 75.6%
Using feature(s) {1, 11, 6} accuracy is 93.2%
Using feature(s) {1, 11, 4} accuracy is 94.2%
Using feature(s) {1, 4, 6} accuracy is 83.2%
Using feature(s) {11, 4} accuracy is 71.4%
Using feature(s) {1, 11} accuracy is 96.8%
Using feature(s) {1, 4} accuracy is 83.8%
Using feature(s) {11} accuracy is 74.2%
Using feature(s) {1} accuracy is 85.6%
Excuetion time using backward elimination for CS205_small_Data__34.txt
is:-221.22128438949585
Graph saved as Backward Elimination_CS205_small_Data__34.txt.pdf
Finished search!! The best feature subset is {1, 11}, which has an
accuracy of 96.8%
```

3. *Large Data & Forward Selection*

```
Type in the name of the file to test below:-CS205_large_Data__41.txt
Welcome to Parth's Feature Selection Algorithm.
Type the number of the algorithm you want to run.
         1.Forward Selection
         2.Backward Elimination
1

This dataset has 50 features (not including the class attribute), with
5000 instances.
Running nearest neighbor with all 50 features, using "leaving-one-out"
evaluation, I get an accuracy of 70.0%
Beginning search.
Using feature(s) {0} accuracy is 70.1%
Using feature(s) {1} accuracy is 71.3%
Using feature(s) {2} accuracy is 70.4%
Using feature(s) {3} accuracy is 71.3%
Using feature(s) {4} accuracy is 72.5%
Using feature(s) {5} accuracy is 71.4%
Using feature(s) {6} accuracy is 70.3%
Using feature(s) {7} accuracy is 70.2%
Using feature(s) {8} accuracy is 71.7%
Using feature(s) {9} accuracy is 70.6%
Using feature(s) {10} accuracy is 70.8%
Using feature(s) {11} accuracy is 71.5%
Using feature(s) {12} accuracy is 70.7%
Using feature(s) {13} accuracy is 69.9%
Using feature(s) {14} accuracy is 70.4%
Using feature(s) {15} accuracy is 70.9%
Using feature(s) {16} accuracy is 70.1%
Using feature(s) {17} accuracy is 70.5%
Using feature(s) {18} accuracy is 70.9%
Using feature(s) {19} accuracy is 70.0%
Using feature(s) {20} accuracy is 85.0%
Using feature(s) {21} accuracy is 69.4%
Using feature(s) {22} accuracy is 71.2%
Using feature(s) {23} accuracy is 71.5%
Using feature(s) {24} accuracy is 71.1%
Using feature(s) {25} accuracy is 70.1%
Using feature(s) {26} accuracy is 71.6%
Using feature(s) {27} accuracy is 71.3%
Using feature(s) {28} accuracy is 70.6%
Using feature(s) {29} accuracy is 71.6%
Using feature(s) {30} accuracy is 70.4%
Using feature(s) {31} accuracy is 71.6%
Using feature(s) {32} accuracy is 70.2%
Using feature(s) {33} accuracy is 70.7%
Using feature(s) {34} accuracy is 69.9%
Using feature(s) {35} accuracy is 70.3%
Using feature(s) {36} accuracy is 71.2%
Using feature(s) {37} accuracy is 70.0%
Using feature(s) {38} accuracy is 70.7%
Using feature(s) {39} accuracy is 71.7%
Using feature(s) {40} accuracy is 71.4%
Using feature(s) {41} accuracy is 71.0%
Using feature(s) {42} accuracy is 71.3%
Using feature(s) {43} accuracy is 71.1%
Using feature(s) {44} accuracy is 70.0%
Using feature(s) {45} accuracy is 70.0%
Using feature(s) {46} accuracy is 70.5%
Using feature(s) {47} accuracy is 71.2%
Using feature(s) {48} accuracy is 70.4%
Using feature(s) {49} accuracy is 73.7%
Using feature(s) {0, 20} accuracy is 83.5%
Using feature(s) {1, 20} accuracy is 83.4%
Using feature(s) {2, 20} accuracy is 84.3%
Using feature(s) {3, 20} accuracy is 84.7%
Using feature(s) {20, 4} accuracy is 84.4%
Using feature(s) {20, 5} accuracy is 83.7%
Using feature(s) {20, 6} accuracy is 85.1%
Using feature(s) {20, 7} accuracy is 83.4%
Using feature(s) {8, 20} accuracy is 84.9%
Using feature(s) {9, 20} accuracy is 84.0%
Using feature(s) {10, 20} accuracy is 84.9%
Using feature(s) {11, 20} accuracy is 83.7%
Using feature(s) {20, 12} accuracy is 84.9%
Using feature(s) {20, 13} accuracy is 84.2%
Using feature(s) {20, 14} accuracy is 84.5%
Using feature(s) {20, 15} accuracy is 83.9%
Using feature(s) {16, 20} accuracy is 83.9%
Using feature(s) {17, 20} accuracy is 83.6%
Using feature(s) {18, 20} accuracy is 84.6%
Using feature(s) {19, 20} accuracy is 83.5%
Using feature(s) {20, 21} accuracy is 83.5%
Using feature(s) {20, 22} accuracy is 85.1%
Using feature(s) {20, 23} accuracy is 84.4%
Using feature(s) {24, 20} accuracy is 83.0%
Using feature(s) {25, 20} accuracy is 84.4%
Using feature(s) {26, 20} accuracy is 84.9%
Using feature(s) {27, 20} accuracy is 83.6%
Using feature(s) {20, 28} accuracy is 84.4%
Using feature(s) {20, 29} accuracy is 83.6%
Using feature(s) {20, 30} accuracy is 82.5%
Using feature(s) {20, 31} accuracy is 85.2%
Using feature(s) {32, 20} accuracy is 85.3%
Using feature(s) {33, 20} accuracy is 84.6%
Using feature(s) {34, 20} accuracy is 84.2%
Using feature(s) {35, 20} accuracy is 84.1%
Using feature(s) {20, 36} accuracy is 83.4%
Using feature(s) {20, 37} accuracy is 83.9%
Using feature(s) {20, 38} accuracy is 83.6%
Using feature(s) {20, 39} accuracy is 83.4%
Using feature(s) {40, 20} accuracy is 84.3%
Using feature(s) {41, 20} accuracy is 85.7%
Using feature(s) {42, 20} accuracy is 83.1%
Using feature(s) {43, 20} accuracy is 84.0%
Using feature(s) {20, 44} accuracy is 84.5%
Using feature(s) {20, 45} accuracy is 84.2%
Using feature(s) {20, 46} accuracy is 84.7%
Using feature(s) {20, 47} accuracy is 84.1%
Using feature(s) {48, 20} accuracy is 84.3%
Using feature(s) {49, 20} accuracy is 97.3%
Using feature(s) {0, 49, 20} accuracy is 95.3%
Using feature(s) {49, 20, 1} accuracy is 95.1%
Using feature(s) {49, 2, 20} accuracy is 95.1%
Using feature(s) {49, 3, 20} accuracy is 94.8%
Using feature(s) {49, 20, 4} accuracy is 95.6%
Using feature(s) {49, 20, 5} accuracy is 95.4%
Using feature(s) {49, 20, 6} accuracy is 95.2%
Using feature(s) {49, 20, 7} accuracy is 95.6%
Using feature(s) {8, 49, 20} accuracy is 95.9%
Using feature(s) {49, 20, 9} accuracy is 95.2%
Using feature(s) {49, 10, 20} accuracy is 95.1%
Using feature(s) {49, 11, 20} accuracy is 95.0%
Using feature(s) {49, 20, 12} accuracy is 94.8%
Using feature(s) {49, 20, 13} accuracy is 95.6%
Using feature(s) {49, 20, 14} accuracy is 95.4%
Using feature(s) {49, 20, 15} accuracy is 95.1%
Using feature(s) {16, 49, 20} accuracy is 95.2%
Using feature(s) {49, 20, 17} accuracy is 95.6%
Using feature(s) {49, 18, 20} accuracy is 95.3%
Using feature(s) {49, 19, 20} accuracy is 95.6%
Using feature(s) {49, 20, 21} accuracy is 95.1%
Using feature(s) {49, 20, 22} accuracy is 96.8%
Using feature(s) {49, 20, 23} accuracy is 94.9%
Using feature(s) {24, 49, 20} accuracy is 95.1%
Using feature(s) {49, 20, 25} accuracy is 95.7%
Using feature(s) {49, 26, 20} accuracy is 95.4%
Using feature(s) {49, 27, 20} accuracy is 94.9%
Using feature(s) {49, 20, 28} accuracy is 95.5%
Using feature(s) {49, 20, 29} accuracy is 95.7%
Using feature(s) {49, 20, 30} accuracy is 95.2%
Using feature(s) {49, 20, 31} accuracy is 95.2%
Using feature(s) {32, 49, 20} accuracy is 94.9%
Using feature(s) {49, 20, 33} accuracy is 95.3%
Using feature(s) {49, 34, 20} accuracy is 94.9%
Using feature(s) {49, 35, 20} accuracy is 95.3%
Using feature(s) {49, 20, 36} accuracy is 95.4%
Using feature(s) {49, 20, 37} accuracy is 95.3%
Using feature(s) {49, 20, 38} accuracy is 94.9%
Using feature(s) {49, 20, 39} accuracy is 95.5%
Using feature(s) {40, 49, 20} accuracy is 95.0%
Using feature(s) {49, 20, 41} accuracy is 95.3%
Using feature(s) {49, 42, 20} accuracy is 95.0%
Using feature(s) {49, 43, 20} accuracy is 95.1%
Using feature(s) {49, 20, 44} accuracy is 95.2%
Using feature(s) {49, 20, 45} accuracy is 94.8%
Using feature(s) {49, 20, 46} accuracy is 95.0%
Using feature(s) {49, 20, 47} accuracy is 95.9%
Using feature(s) {48, 49, 20} accuracy is 95.2%
Excuetion time using forward selection for CS205_large_Data__41.txt is:-
7283.639548301697
Graph saved as Forward Selection_CS205_large_Data__41.txt.pdf
Finished search!! The best feature subset is {49, 20}, which has an
accuracy of 97.3%
```

4. *Large Data & Backward Elimination*

```
Type in the name of the file to test below:-CS205_large_Data__41.txt
Welcome to Parth's Feature Selection Algorithm.
Type the number of the algorithm you want to run.
         1.Forward Selection
         2.Backward Elimination
2

This dataset has 50 features (not including the class attribute), with
5000 instances.
Running nearest neighbor with all 50 features, using “leaving-one-out”
evaluation, I get an accuracy of 70.0%
Beginning search.
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
```

```
Using feature(s) {0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
```

```
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
Using feature(s) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 35,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
Using feature(s) {1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 1, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
```

```
Using feature(s) {0, 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 10, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 3
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 17,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 35, 36
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36
```

```
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
Using feature(s) {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35
c:\All Data\AI\Assignment-2\new_feature.py:128: UserWarning: Tight layout
not applied. The bottom and top margins cannot be made large enough to
accommodate all axes decorations.
  fig.tight_layout()
Graph saved as Backward ElimEination_CS205_large_Data__41.txt.pdf
Finished search!! The best feature subset is {0, 1, 2, 3, 4, 5, 6, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,
29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46,
48}, which has an accuracy of 72.6%
```

Summary of results:

I ran all the above combinations for the given dataset and have found the following results. The timing results of the various dataset and algorithm combinations are presented in the following table.

I have also generated graphs for each combination. The first graph for each combination is a mix of bar and line plot representations of the accuracy and feature set chosen. The second graph is a scatter plot showing the feature set versus accuracy.

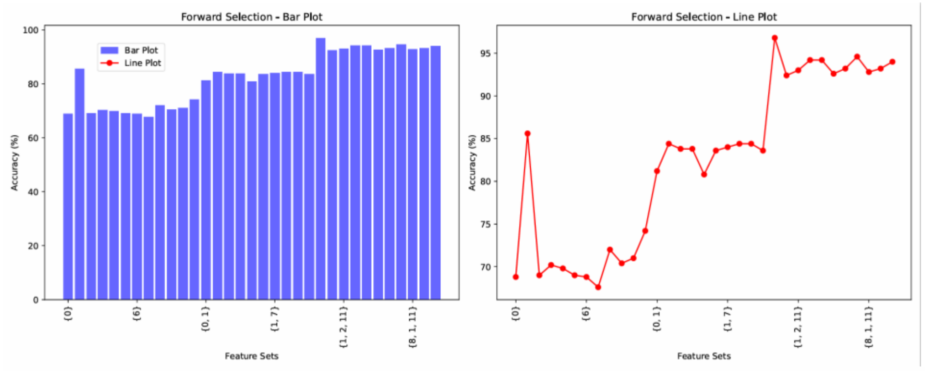
The scripts are provided in this github repo.

Timing Table:-

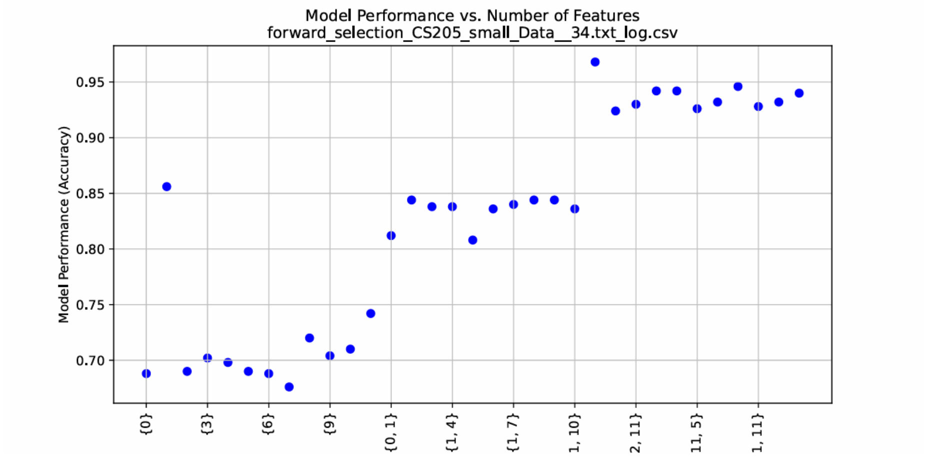
Dataset	**Feature selection technique**	**Excuetion time**
Small Dataset	Forward Selection	57 seconds
Small Dataset	Backward Elimination	221 seconds
Large Dataset	Forward Selection	7283 seconds(approx 2.02 hours)
Large Dataset	Backward Elimination	8023 seconds(approx 2.20 hours)

1. Small Data & Forward Selection.

a.

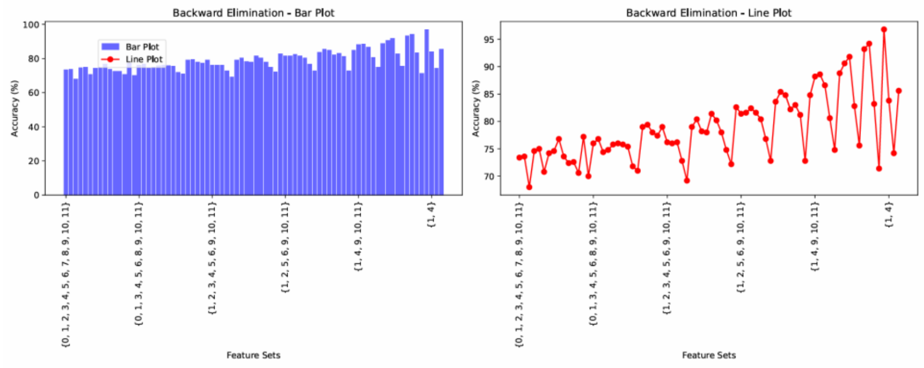


b.

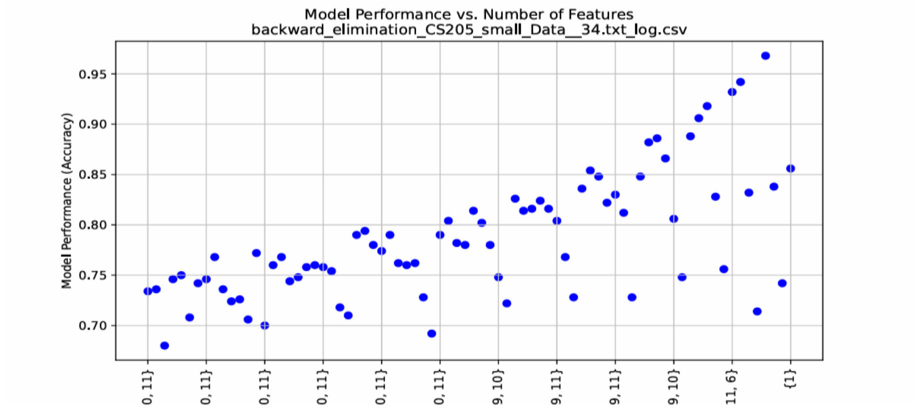


2. Small Data & Backward Elimination

a.

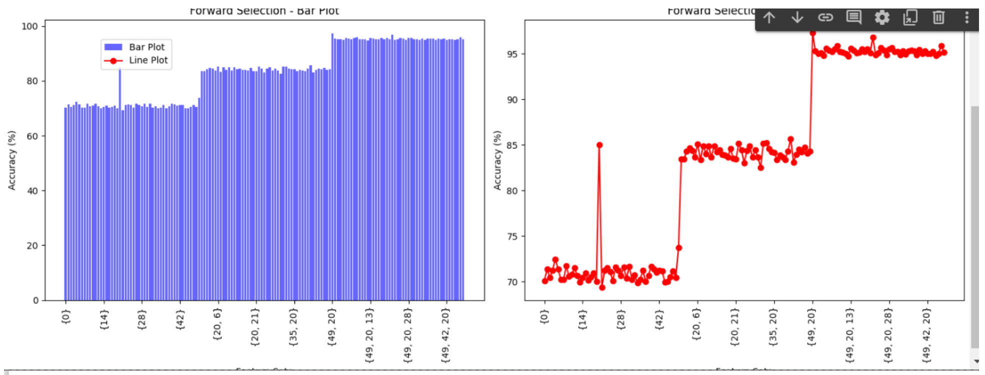


b.

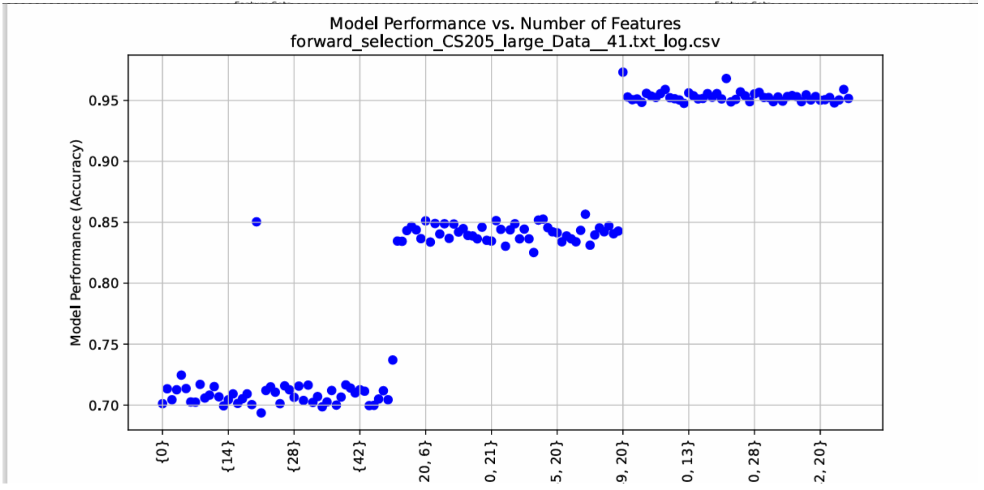


3. Large Data & Forward Selection

a.

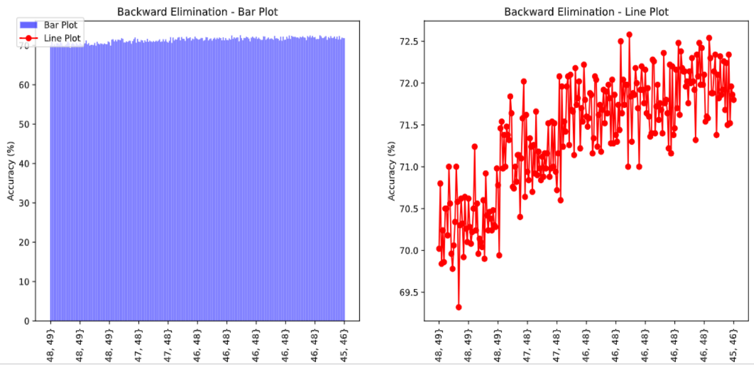


b.

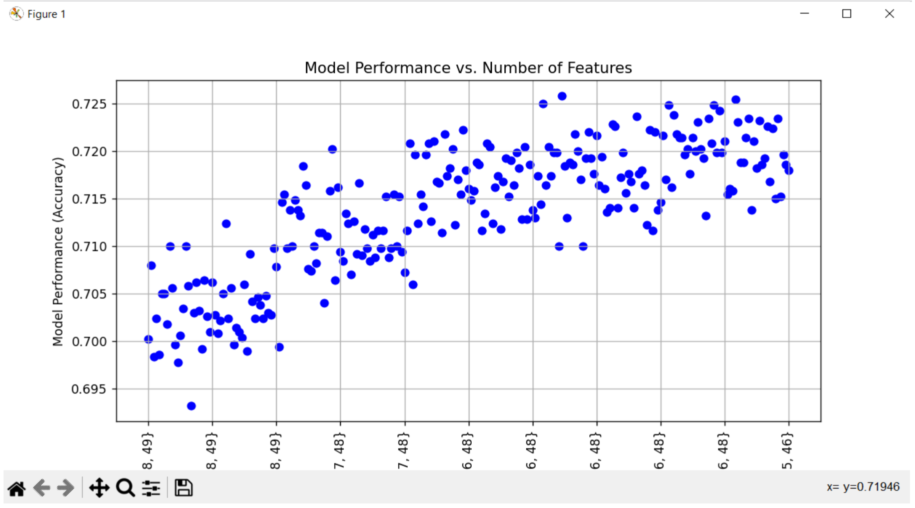


4. Large Data & Backward Elimination

a.



b.



Conclusion

- Both algorithms are complete but not optimal, as they gave similar results for the small dataset but different ones for the larger one. The backward elimination algorithm produced a worse feature set than forward selection. There may be more optimal solutions, but these were not found due to the greedy nature of the algorithms, which search for local maxima (in terms of accuracy) rather than global maxima.
- Forward selection converged faster in my case, but this may differ in other cases, considering both algorithms are greedy.
- A Bi-directional approach (using both forward selection and backward elimination) may give better results.
- The results overall seem intuitive based on the content discussed in the lectures.