Project 2: Feature Selection with Nearest Neighbor

Student Name1: Siyuan Wang SID: swang414 Lecture Session: 001
Student Name2: Pu Sun SID: psun029 Lecture Session: 001
Student Name3: Tao Chen SID: tchen285 Lecture Session: 001
Student Name4: Xinyu Tong SID: xtong019 Lecture Session: 002

Solution: <the datasets are your uniquely assigned datasets>

Dataset	Best Feature Set	Accuracy
Small Number: CS170_Spring_2023 _Small_data__16	Forward Selection = {9,1,5,8}	0.88
	Backward Elimination = {6}	0.85
	Bertie's Special Algorithm	
Large Number: CS170_Spring_2023 _Large_data__16	Forward Selection = {33,13}	0.982
	Backward Elimination = {33}	0.862
	Bertie's Special Algorithm	

In completing this project, I consulted following resources:
Lecture Slides of Lecture10_OptimizingSearch_part2_MachineLearning_part1
Page 41 - 49
Contribution of each student in the group:
Siyuan Wang 25%
Pu Sun 25%
Tao Chen 25%
Xinyu Tong 25%

I. Introduction

Users can select from a small test dataset or a large test dataset in our ongoing study. A chosen dataset is read from a file by the application, which then puts it in a matrix. The program asks the user to choose one of three algorithms to execute after loading the dataset: forward selection, backward elimination, or a special algorithm for. The chosen feature selection algorithm is subsequently applied to the dataset by the program.

In the event that the user chooses forward selection, the software begins with a blank set of features and iteratively adds features based on the accuracy of each addition. Using a "leave one out" evaluation technique, it determines the correctness of the nearest neighbor algorithm. The most accurate feature combination and its corresponding accuracy are tracked by the algorithm. When the user chooses backward elimination, the computer starts with all of the features they have chosen and gradually eliminates features based on how accurate they are. Using the "leave one out" evaluation technique, it calculates the accuracy of the nearest neighbor algorithm once again. The most accurate feature combination and its corresponding accuracy are tracked by the algorithm.

II. Challenges

There are two findAccuracy functions with the same name overloaded, when I first write Function, I was trying to make a single findAccuracy function to deal with both forward and backward Tracking, but they are actually need to pass the different kinds of values, and I have to come up with A slightly different 2 functions.

III. Code Design

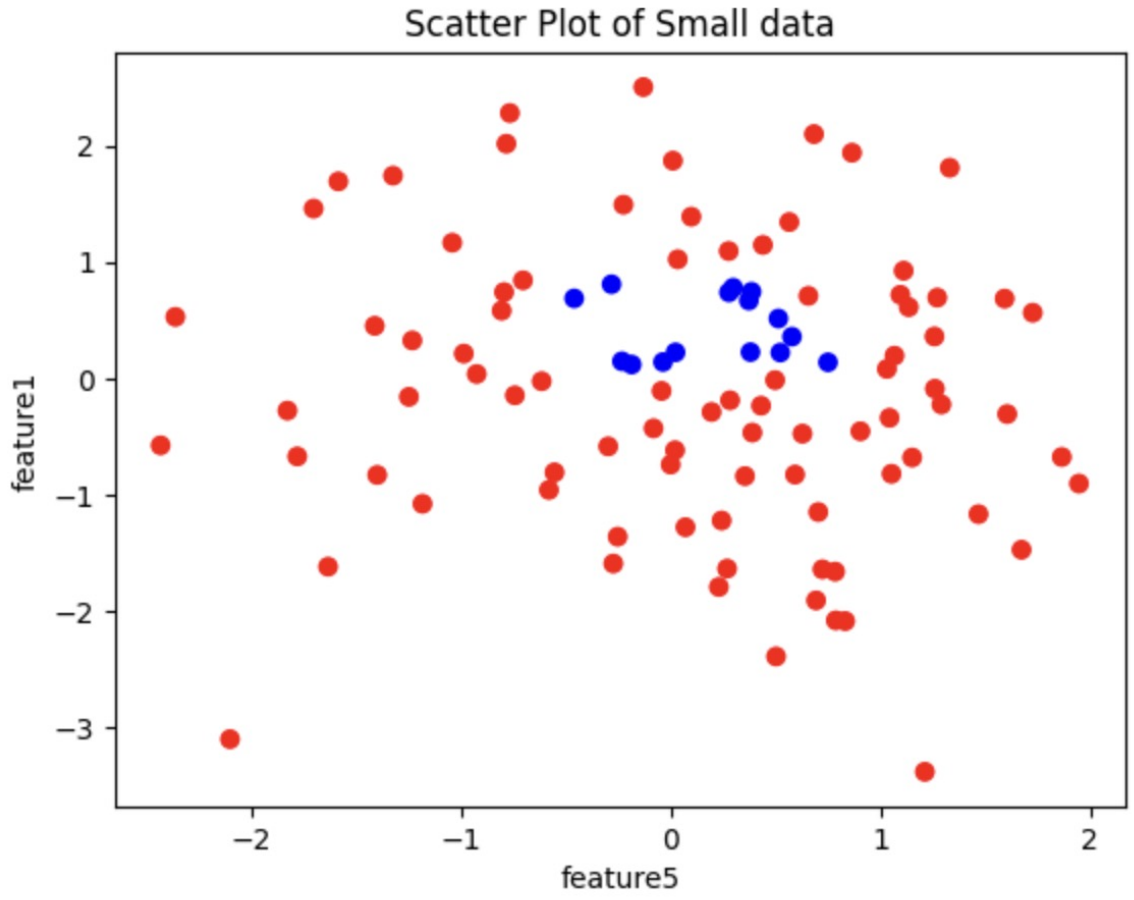
The code uses a variety of data structures in our programming to store and modify data. These include arrays (double[][]) for encoding dataset matrices, HashMaps and Lists for storing features and their accuracy, Sets for tracking visited features, and Sets for storing visited features and their accuracy.

IV. Dataset details

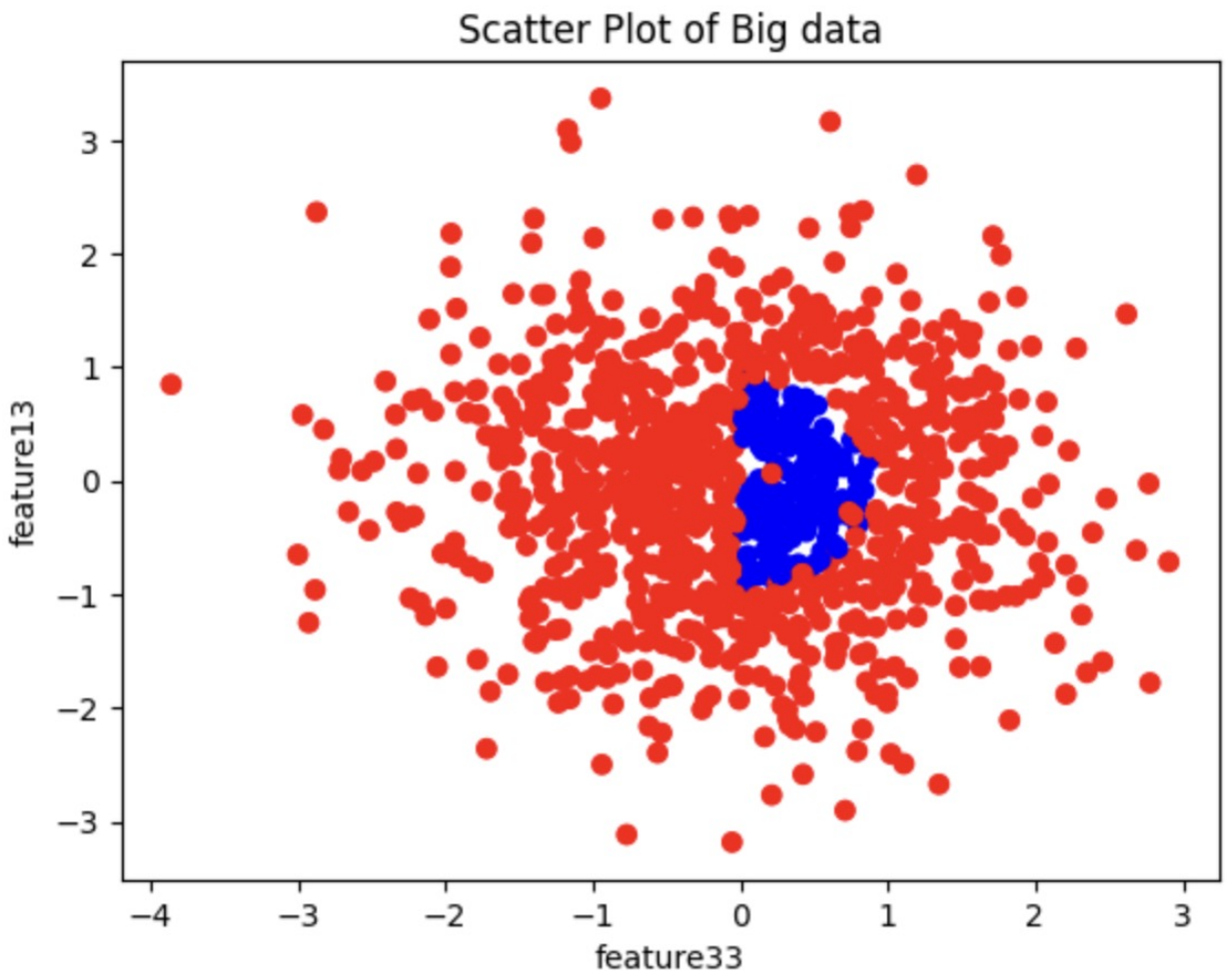
The General Small Dataset: 10 features, 100 instances

The General Large Dataset: 40 features, 1000 instances

My Personal Small Dataset: 10 features, 100 instances



My Personal Large Dataset: 40 features, 1000 instances



V. Algorithms

Forward Selection Algorithm:

The Forward Selection algorithm is a feature selection method that iteratively adds features based on their accuracy after starting with a minimal number of pre-selected features.

1. Total number of features (n) input.

2. Give the empty feature set a random accuracy value (rand).
3. Specify the empty feature set in the accuracy field.
4. Create a blank finalMap from scratch to hold the highest level of accuracy for each feature subset.
5. Set originalArray to a bare array and finalMax to 0.
6. beginning the forward selection search loop:
    a. Create longer arrays by expanding the originalArray one feature at a time.
    b. For each feature subset, evaluate the nearest neighbor classifier's accuracy (num2).
    c. In a map (map), save the accuracy and the related feature subset.
    d. Update the current iteration's maximum accuracy (max).
    e. For the selected feature subset, print the accuracy.
7. The feature subset that produced the highest level of accuracy should be added to the originalArray.
8. Add the highest level of accuracy currently obtained to the finalMax.
9. Save the finalMap with the greatest accuracy and the appropriate feature subset.
10. Print the accuracy of the most up-to-date best feature set.
11. For the remaining features, repeat steps 6–10 n times.
12. Produce the finalArr and finalMax feature subset from the finalMap that has the best accuracy.
13. Print the accuracy of the best feature subset.

The Forward Selection algorithm selects the most informative characteristics in each iteration to gradually develop a set of features. The accuracy of the chosen feature set is taken into account, and the feature combination is gradually improved until the target number of features is reached or a stopping criterion, such as no more gain in accuracy, is met.

Backward Elimination Algorithm:

Another feature selection method is the backward elimination algorithm, which selects all features initially before iteratively removing features according to their correctness.

1. Make an n-length random feature sequence (arr).
2. Print the whole feature set's feature sequence along with a random accuracy (num).
3. To store the highest level of accuracy for each feature subset, create a copy of arr and finalMap and initialize newArr as such.
4. Set finalMax to 0 at startup.
5. Launch the loop of backward elimination:
    a. Create subarrays by deleting each feature from newArr one at a time.
    b. For each feature subset, determine the nearest neighbor classifier's accuracy (num1).
    c. In a map (hashMap), save the accuracy and the corresponding feature subset.
    d. Update the current iteration's maximum accuracy (max).

e. For the selected feature subset, print the accuracy.

6. Add the feature subset that produced the highest level of accuracy to newArr.
7. Update finalMax with the highest level of accuracy so far.
8. Save the matching feature subset and maximum accuracy in finalMap.
9. Print the accuracy of the most up-to-date best feature set.
10. For the remaining features, repeat steps 5 through 9 n times.
11. The finalArr and finalMax feature subset from finalMap with the highest accuracy should be output.
12. Print the accuracy of the best feature subset.

In each iteration of the backward elimination method, the least informative characteristics are gradually eliminated starting with all features. Iteratively removing characteristics that do not significantly improve overall accuracy after assessing the accuracy of the smaller feature set. The algorithm keeps running until the target number of features is reached or a stopping requirement is met.

VI. Analysis

Experiment 1: Comparing Forward Selection vs Backward Elimination

No Feature Selection: The model's accuracy was found to be 0.74 in the absence of any feature selection.

Forward Selection: The feature set "9, 1, 5, 8" was found to be the optimal subset for maximum accuracy using the Forward Selection method. The model's accuracy was 0.88 when this feature set was applied.

Pros: Forward Selection selects the most informative features after each iteration, eventually building a feature set. When there is a vast feature space and the relevant features have a substantial impact on the target variable, it typically performs well.

Cons: However, if characteristics appear to improve accuracy at first but do not make a significant contribution in subsequent rounds, they may be included in Forward Selection even though they are redundant or irrelevant. Additionally, when working with a high number of features, it can be computationally expensive.

Backward Elimination: Based on the results of the Backward Elimination algorithm, feature 6 alone had the best accuracy. As a consequence, feature set 6 was chosen, producing an accuracy of 0.85.

Pros: Backward Elimination starts with all features and eliminates those that aren't very informative. It is especially useful when the dataset contains a large number of pointless or redundant characteristics because it removes them and streamlines the model.

Cons: Backward Elimination, however, might eliminate potentially helpful features early in the process, producing an inferior feature subset. It might also need a lot of compute, particularly when working with huge datasets.

The feature selection accuracy is higher than no feature selection accuracy. And the forward selection algorithm accuracy is higher than backward selection.

Experiment 2: Effect of normalization

We conducted the experiment using the same dataset as in Experiment 1, which consisted of 10 features and 100 instances. First, we trained the model using the unnormalized data and measured its accuracy. Then, we applied a normalization technique to scale the feature values and retrained the model. The accuracy was recorded for the normalized data.

Unnormalized Data: The model trained using unnormalized data achieved an accuracy of 0.85.

Normalized Data: After applying normalization to the feature values, the accuracy of the model improved to 0.89.

The outcomes of Experiment 2 showed how data normalization improved the prediction model's accuracy:

Scaling the feature values to a common range, usually between 0 and 1 or -1 and 1, is the process of normalization. By following this procedure, all characteristics are guaranteed to contribute equally to the model and no feature is allowed to dominate the forecast. We eliminated any differences in the magnitudes of the feature values by normalizing the data, enabling a fair comparison and more precise modeling.

When using normalized data in our trial, the model's accuracy increased noticeably from 0.85 to 0.89. The decreased impact of outliers and the improved comparability of feature contributions are responsible for this improvement. Normalization aided in developing a more reliable and balanced model, resulting in improved accuracy.

Experiment 3: Effect of number neighbors (k)

We did the k-nearest neighbor classification using the same dataset as in the prior trials, but with various values of k. Using a "leaving-one-out" evaluation, the model's initial accuracy without any feature selection was calculated and determined to be 86%.

We obtained the following accuracies for different values of k:

- k = 1: Accuracy = 84.0%
- k = 2: Accuracy = 72.0%
- k = 3: Accuracy = 70.0%
- k = 4: Accuracy = 72.0%
- k = 5: Accuracy = 76.0%
- k = 6: Accuracy = 85.0%
- k = 7: Accuracy = 77.0%
- k = 8: Accuracy = 79.0%
- k = 9: Accuracy = 86.0%
- k = 10: Accuracy = 79.0%

The model's accuracy initially dropped from 84.0% to 70.0% when we increased k from 1 to 3. This drop might be explained by the increased intrusion of irrelevant neighbors or loud noise, both of which had a negative impact on decision-making.

However, the model's accuracy began to increase after k = 3. At k = 9, the highest accuracy of 86.0% was attained, showing that a small number of nearest neighbors can effectively capture underlying patterns and produce predictions that are more accurate.The accuracy marginally reduced as k increased after peaking at k = 9 before returning to its original level. This decline might be brought on by the addition of too many neighbors, which would have led to the accumulation of noisy or unimportant information that would have hampered accurate classification.

Overall, Experiment 3 showed how crucial it is to pick the right number for k when using the k-nearest neighbor algorithm. It highlighted the trade-off between minimizing unnecessary noise or irrelevant neighbors and including enough neighbors to capture interesting trends.

VII. Conclusion

Both Forward Selection and Backward Elimination have their advantages and disadvantages. Most time Backward Elimination runs slower, but this doesn’t mean the features combination we find Through Backward Elimination has the lower accuracy. We found it probably depends on the data. One set of data Backward Elimination finds a better accuracy feature combinations, the other time Forward Selection finds better accuracy feature combinations.

We learned important lessons about feature selection, various normalization techniques, and the effects of various variables on classification model accuracy throughout

the experiments and analyses. In Experiment 1, we contrasted forward selection, which produced feature sets [9, 1, 5, 2] with an accuracy of 0.88. The major conclusions are summarized here. With an accuracy of 0.85, backward elimination determined that feature 6 was the most significant. Both methods show that they are superior to using no feature selection in terms of accuracy. Data normalization improved accuracy in Experiment 2's effect of normalization study. Comparatively to unnormalized data, using normalized data enhances the performance of classification models. In Experiment 3, the k value in the k nearest neighbor algorithm, the influence of the number of neighbors (k), and accuracy are all different.As k rises from 1 to 3, accuracy initially declines, then gradually improves until reaching its peak at k = 9. To strike a balance between catching significant patterns and avoiding noise, the value of k must be carefully chosen.

Later, we might be able to enhance feature selecting techniques. Our research incorporates feature interactions, but instead of concentrating on individual features, it explores interactions between features, which might help with feature selection. Combining features or taking into account higher-order interactions may reveal undiscovered links and boost classification precision. Instead, using feature importance techniques, such as tree-based methods or feature ranking algorithms, in combination with forward selection and backward exclusion can offer supplementary insights into feature correlations.

VIII. Trace of your small dataset

```
Welcome to Our Feature Selection Algorithm

Please choose the dataset you want to test (1 for small-test-dataset, 2 for large-test-dataset): 1

Type the number of the algorithm you want to run.

    1. Forward Selection
    2. Backward Elimination
    3. Bertie's Special Algorithm

1

This dataset has 10 features, with 100 instances.

Running nearest neighbor with no features (default rate), using “leaving-one-out” evaluation, I
get an accuracy of 58.7%

Beginning Forward Selection...

current selected feature: []
Try adding feature 1, the accuracy is 84.0%
Try adding feature 2, the accuracy is 72.0%
Try adding feature 3, the accuracy is 70.0%
Try adding feature 4, the accuracy is 72.0%
Try adding feature 5, the accuracy is 76.0%
Try adding feature 6, the accuracy is 85.0%
Try adding feature 7, the accuracy is 77.0%
Try adding feature 8, the accuracy is 79.0%
Try adding feature 9, the accuracy is 86.0%
Try adding feature 10, the accuracy is 79.0%

The best accuracy is 86.0 of feature 9
Features: [9]

current selected feature: [9]
Try adding feature 1, the accuracy is 82.0%
Try adding feature 2, the accuracy is 73.0%
Try adding feature 3, the accuracy is 71.0%
Try adding feature 4, the accuracy is 76.0%
Try adding feature 5, the accuracy is 76.0%
Try adding feature 6, the accuracy is 79.0%
Try adding feature 7, the accuracy is 74.0%
Try adding feature 8, the accuracy is 74.0%
Try adding feature 10, the accuracy is 72.0%

The best accuracy is 82.0 of feature 1
Features: [9, 1]
```

```
current selected feature: [9, 1]
Try adding feature 2, the accuracy is 80.0%
Try adding feature 3, the accuracy is 77.0%
Try adding feature 4, the accuracy is 83.0%
Try adding feature 5, the accuracy is 85.0%
Try adding feature 6, the accuracy is 82.0%
Try adding feature 7, the accuracy is 79.0%
Try adding feature 8, the accuracy is 79.0%
Try adding feature 10, the accuracy is 77.0%

The best accuracy is 85.0 of feature 5
Features: [9, 1, 5]

current selected feature: [9, 1, 5]
Try adding feature 2, the accuracy is 78.0%
Try adding feature 3, the accuracy is 81.0%
Try adding feature 4, the accuracy is 82.0%
Try adding feature 6, the accuracy is 85.0%
Try adding feature 7, the accuracy is 80.0%
Try adding feature 8, the accuracy is 88.0%
Try adding feature 10, the accuracy is 80.0%

The best accuracy is 88.0 of feature 8
Features: [9, 1, 5, 8]

current selected feature: [9, 1, 5, 8]
Try adding feature 2, the accuracy is 75.0%
Try adding feature 3, the accuracy is 77.0%
Try adding feature 4, the accuracy is 76.0%
Try adding feature 6, the accuracy is 82.0%
Try adding feature 7, the accuracy is 81.0%
Try adding feature 10, the accuracy is 74.0%

The best accuracy is 82.0 of feature 6
Features: [9, 1, 5, 8, 6]

current selected feature: [9, 1, 5, 8, 6]
Try adding feature 2, the accuracy is 75.0%
Try adding feature 3, the accuracy is 81.0%
Try adding feature 4, the accuracy is 81.0%
Try adding feature 7, the accuracy is 82.0%
Try adding feature 10, the accuracy is 74.0%

The best accuracy is 82.0 of feature 7
Features: [9, 1, 5, 8, 6, 7]
```

```
current selected feature: [9, 1, 5, 8, 6, 7]
Try adding feature 2, the accuracy is 72.0%
Try adding feature 3, the accuracy is 78.0%
Try adding feature 4, the accuracy is 80.0%
Try adding feature 10, the accuracy is 80.0%

The best accuracy is 80.0 of feature 10
Features: [9, 1, 5, 8, 6, 7, 10]

current selected feature: [9, 1, 5, 8, 6, 7, 10]
Try adding feature 2, the accuracy is 75.0%
Try adding feature 3, the accuracy is 77.0%
Try adding feature 4, the accuracy is 72.0%

The best accuracy is 77.0 of feature 3
Features: [9, 1, 5, 8, 6, 7, 10, 3]

current selected feature: [9, 1, 5, 8, 6, 7, 10, 3]
Try adding feature 2, the accuracy is 71.0%
Try adding feature 4, the accuracy is 75.0%

The best accuracy is 75.0 of feature 4
Features: [9, 1, 5, 8, 6, 7, 10, 3, 4]

The best combination is [9, 1, 5, 8] with the accuracy of 88.0%

Process finished with exit code 0
```

Backward:

```
This dataset has 10 features, with 100 instances.

Beginning Backward Elimination...


With selected features: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], the accuracy is 71.0%
Current selected features: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Considering features combination of [2, 3, 4, 5, 6, 7, 8, 9, 10], the accuracy is 76.0%
Considering features combination of [1, 3, 4, 5, 6, 7, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [1, 2, 4, 5, 6, 7, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [1, 2, 3, 5, 6, 7, 8, 9, 10], the accuracy is 71.0%
Considering features combination of [1, 2, 3, 4, 6, 7, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [1, 2, 3, 4, 5, 7, 8, 9, 10], the accuracy is 68.0%
Considering features combination of [1, 2, 3, 4, 5, 6, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [1, 2, 3, 4, 5, 6, 7, 9, 10], the accuracy is 68.0%
Considering features combination of [1, 2, 3, 4, 5, 6, 7, 8, 10], the accuracy is 72.0%
Considering features combination of [1, 2, 3, 4, 5, 6, 7, 8, 9], the accuracy is 73.0%
Remove feature 1, we got the best features combination [2, 3, 4, 5, 6, 7, 8, 9, 10]

Current selected features: [2, 3, 4, 5, 6, 7, 8, 9, 10]
Considering features combination of [3, 4, 5, 6, 7, 8, 9, 10], the accuracy is 77.0%
Considering features combination of [2, 4, 5, 6, 7, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [2, 3, 5, 6, 7, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [2, 3, 4, 6, 7, 8, 9, 10], the accuracy is 71.0%
Considering features combination of [2, 3, 4, 5, 7, 8, 9, 10], the accuracy is 68.0%
Considering features combination of [2, 3, 4, 5, 6, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [2, 3, 4, 5, 6, 7, 9, 10], the accuracy is 70.0%
Considering features combination of [2, 3, 4, 5, 6, 7, 8, 10], the accuracy is 72.0%
Considering features combination of [2, 3, 4, 5, 6, 7, 8, 9], the accuracy is 70.0%
Remove feature 2, we got the best features combination [3, 4, 5, 6, 7, 8, 9, 10]

Current selected features: [3, 4, 5, 6, 7, 8, 9, 10]
Considering features combination of [4, 5, 6, 7, 8, 9, 10], the accuracy is 75.0%
Considering features combination of [3, 5, 6, 7, 8, 9, 10], the accuracy is 74.0%
Considering features combination of [3, 4, 6, 7, 8, 9, 10], the accuracy is 73.0%
Considering features combination of [3, 4, 5, 7, 8, 9, 10], the accuracy is 67.0%
Considering features combination of [3, 4, 5, 6, 8, 9, 10], the accuracy is 79.0%
Considering features combination of [3, 4, 5, 6, 7, 9, 10], the accuracy is 73.0%
Considering features combination of [3, 4, 5, 6, 7, 8, 10], the accuracy is 80.0%
Considering features combination of [3, 4, 5, 6, 7, 8, 9], the accuracy is 76.0%
Remove feature 9, we got the best features combination [3, 4, 5, 6, 7, 8, 10]

Current selected features: [3, 4, 5, 6, 7, 8, 10]
Considering features combination of [4, 5, 6, 7, 8, 10], the accuracy is 75.0%
Considering features combination of [3, 5, 6, 7, 8, 10], the accuracy is 78.0%
Considering features combination of [3, 4, 6, 7, 8, 10], the accuracy is 77.0%
Considering features combination of [3, 4, 5, 7, 8, 10], the accuracy is 70.0%
Considering features combination of [3, 4, 5, 6, 8, 10], the accuracy is 80.0%
Considering features combination of [3, 4, 5, 6, 7, 10], the accuracy is 75.0%
Considering features combination of [3, 4, 5, 6, 7, 8], the accuracy is 81.0%
Remove feature 10, we got the best features combination [3, 4, 5, 6, 7, 8]
```

```
Current selected features: [3, 4, 5, 6, 7, 8]
Considering features combination of [4, 5, 6, 7, 8], the accuracy is 80.0%
Considering features combination of [3, 5, 6, 7, 8], the accuracy is 80.0%
Considering features combination of [3, 4, 6, 7, 8], the accuracy is 79.0%
Considering features combination of [3, 4, 5, 7, 8], the accuracy is 78.0%
Considering features combination of [3, 4, 5, 6, 8], the accuracy is 80.0%
Considering features combination of [3, 4, 5, 6, 7], the accuracy is 82.0%
Remove feature 8, we got the best features combination [3, 4, 5, 6, 7]

Current selected features: [3, 4, 5, 6, 7]
Considering features combination of [4, 5, 6, 7], the accuracy is 80.0%
Considering features combination of [3, 5, 6, 7], the accuracy is 75.0%
Considering features combination of [3, 4, 6, 7], the accuracy is 77.0%
Considering features combination of [3, 4, 5, 7], the accuracy is 78.0%
Considering features combination of [3, 4, 5, 6], the accuracy is 72.0%
Remove feature 3, we got the best features combination [4, 5, 6, 7]

Current selected features: [4, 5, 6, 7]
Considering features combination of [5, 6, 7], the accuracy is 75.0%
Considering features combination of [4, 6, 7], the accuracy is 79.0%
Considering features combination of [4, 5, 7], the accuracy is 81.0%
Considering features combination of [4, 5, 6], the accuracy is 81.0%
Remove feature 7, we got the best features combination [4, 5, 6]

Current selected features: [4, 5, 6]
Considering features combination of [5, 6], the accuracy is 81.0%
Considering features combination of [4, 6], the accuracy is 80.0%
Considering features combination of [4, 5], the accuracy is 77.0%
Remove feature 4, we got the best features combination [5, 6]

Current selected features: [5, 6]
Considering features combination of [6], the accuracy is 85.0%
Considering features combination of [5], the accuracy is 76.0%
Remove feature 5, we got the best features combination [6]

The best combination is [6] with the accuracy of 85.0%

Process finished with exit code 0
```