Project 2: Feature Selection with Nearest Neighbor

Name: Jianeng Yang SID: 862039649

Solution: Personal dataset 102

Dataset	Best Feature Set	Accuracy
Small Number: 102	Forward Feature Selection = {4, 9}	0.93
	Backward Elimination = {1,2,3,4,5,6,7,8,9,10}	0.83
	Custom Algorithm = {4,9}	0.93
Large Number: 102	Forward Feature Selection = {8, 2}	0.972
	Backward Elimination = {1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 40}	0.797
	Custom Algorithm = {8,2}	0.972

Solution: Inital dataset(test for code correctness)

Dataset	Best Feature Set	Accuracy
Small Number: Initial dataset	Forward Selection = {3,5}	0.92
	Backward Elimination = {2, 4, 5, 7, 10}	0.83
	Custom Algorithm = {3,5}	0.92
Large Number: Initial dataset	Forward Selection = {1,27}	0.955
	Backward Elimination = {1, 2, 3, 5, 9, 10, 11, 12, 13, 15, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 35, 36, 37, 38, 40}	0.745
	Custom Algorithm = {1,27}	0.955

--------------------------------<Begin Report>-----------------------------------

In completing this project, I consulted following resources:

how to keep track of time in python

https://www.codegrepper.com/code-examples/python/python+track+time+of+execution

How to normalize the data in python(standard mean normalization, min-max normalization)

https://stackoverflow.com/questions/26414913/normalize-columns-of-pandas-data-frame

Bidirection Elimination idea

https://stats.stackexchange.com/questions/421361/how-exactly-does-bidirectional-stepwise-elimination-works

I. Introduction

For this project, we implemented 2 types of search algorithms: forward feature selection and backward elimination. For these two search algorithms, we are able to determine the best feature set for each dataset and their accuracy. The 3 types of search algorithms are custom algorithms(optional). This algorithm is a combination of forwarded feature selection and backward elimination, which do the forward feature selection while eliminating the feature that causes the accuracy drop.

II. Challenges

The most challenging part of this project is the running time of backward elimination search algorithms. It takes more than 1-2 hours to get the result of the best feature on the large dataset. So, I’m not able to compare the increase k vs the best feature accuracy on the large dataset.

III. Code Design

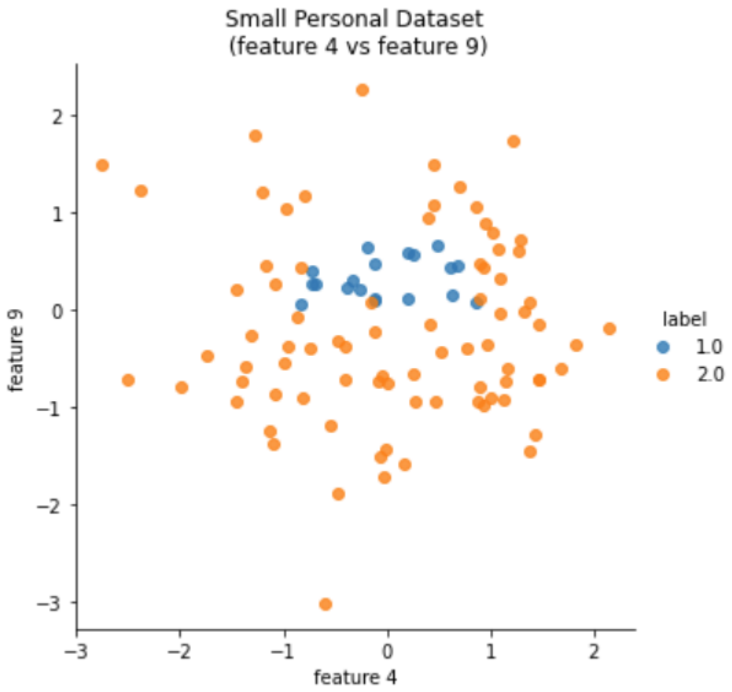
Submit with the same folder as report

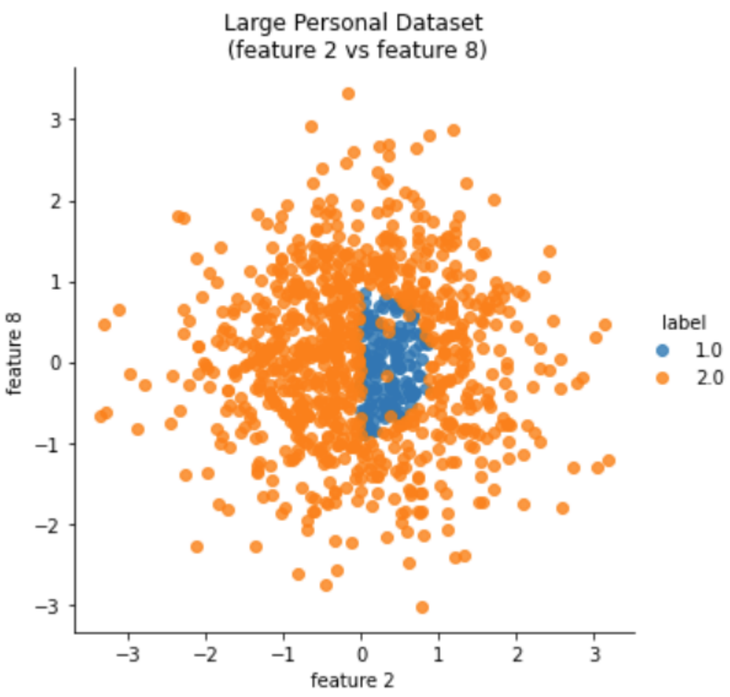
IV. Dataset details

Small Dataset: Number of features = 10, number of instances = 100

Large Dataset: Number of features = 40, number of instances = 1000

Plot some features of my personal dataset.





The above are the best feature sets that we got from the forward feature selection algorithm. By observing these two graphs, we see that class label 1 is more in the center and class label 2 is more spread out. Also, we can see that most of the data are class label 2 in both small and large datasets

V. Algorithms

1. Forward Feature Selection
2. Backward Elimination
3. Jianeng Yang algorithm (optional)

- Bi-direction Elimination search: it's a combination of forwarding Feature Selection and Backward Elimination.
- It results in the same accuracy and best feature set as the forward feature selection algorithm but it does search faster.

VI. Analysis

Experiment 1: Comparing Forward Selection vs Backward Elimination.

Dataset	Best Feature Set	Time(sec)	Accuracy
Small Number: 102	Forward Feature Selection = {4, 9}	0.6109	0.93
	Backward Elimination = {1,2,3,4,5,6,7,8,9,10}	0.5589	0.84
	Custom Algorithm = {4,9}	0.5499	0.93
Large Number: 102	Forward Feature Selection = {8, 2}	298.0437	0.972
	Backward Elimination = {1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 40}	4219.4853	0.868
	Custom Algorithm = {4,9}	190.6706	0.972

For observation tables above, forward feature selection searches have better time and accuracy than backward elimination searches in both small and large datasets. But there is one exception, the Backward Elimination on a small dataset is faster than the Forward feature selection search time but it still has lower accuracy. Because the accuracy is dropped at the beginning of the first elimination feature subsets, so we stop the search at the very beginning.For a better time, the Backward elimination uses all features first then eliminates one each time until no feature, and the forward feature selection uses one feature first and adds one each time util using all features. This cause the Backward elimination to use more "feature" overall, which it needs to spend more time calculating the euclidean distance. For example, backward use 40 feature, then we calculate euclidean distance= sqrt[(x0-y0)^2 + (x1-y1)^2 + ... + (x39-y9)^2]. For forward feature selection, we use 1 feature, then we calculate the euclidean distance= sqrt[(x0-y0)^2 + (x1-y1)^2]. Which we saw there are huge amound of time spend on search between Forward Feature Selection and Backward Elimination Algorithn. (298.0437 seconds vs

4219.45853 seconds; 4.96 mins vs 70.32 mins). For the accuracy, I observed that the accuracy decreases as we add more features during the forward feature selection search, except for the best feature subset. And the accuracy mostly increases as we eliminate features during the backward elimination search.

The custom algorithm, it's the combination of forwarding feature selection and backward elimination. So, it had the same best feature set and accuracy as the forward feature selection but it searches faster than the forward feature selection. As result, the best search algorithm is our custom algorithm because it had higher accuracy and faster search time.

Experiment 2: Effect of normalization

Forward Selection

Dataset	Best Feature Set	Accuracy
Small Number: 102	Unnormalized = {4, 9}	0.93
	Standard normalized data = {4, 9}	0.93
	Min-max normalized data ={4, 9}	0.93
Large Number: 102	Unormalized data= {8, 2}	0.972
	Standard normalized data = {8, 2}	0.972
	Min-Max normalized data = {8, 2}	0.972

Backward Elimination

Dataset	Best Feature Set	Accuracy
Small Number: 102	Unnormalized = {1,2,3,4,5,6,7,8,9,10}	0.83
	standard normalized data = {1,2,3,4,5,6,7,8,9,10}	0.81
	Min-max normalized data = {1,2,3,4,5,6,7,8,9,10}	0.82
Large Number: 102	unnormalized data = {1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 40}	0.797
	standard normalized data= {1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 40}	0.79
	Min-Max normal data = {1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 40}	0.786

Custom Algorithm Bidirection Elimination

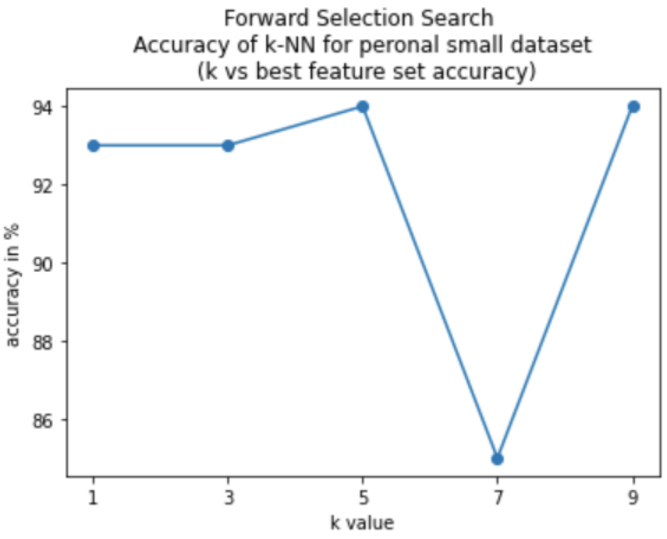
Dataset	Best Feature Set	Accuracy
Small Number: 102	Unnormalized = {4, 9}	0.93
	Standard normalized data = {4, 9}	0.93
	Min-max normalized data ={4, 9}	0.93

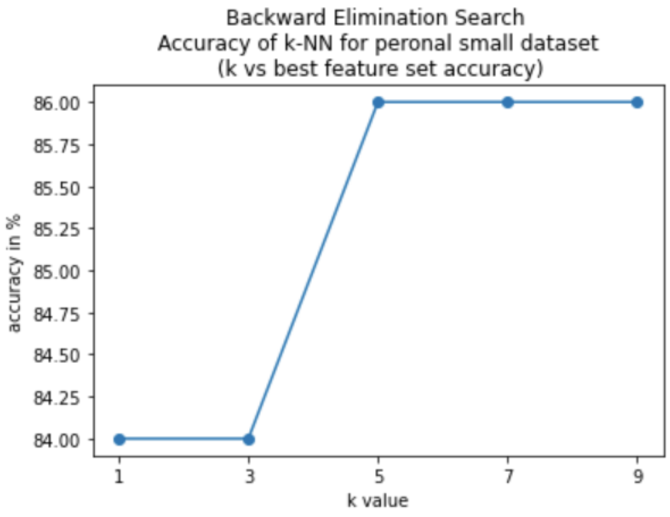
Large Number: 102	Unormalized data= {8, 2}	0.972
	Standard normalized data = {8, 2}	0.972
	Min-Max normalized data = {8, 2}	0.972

Contrast the result of the best feature set's accuracy on the unnormalized data and normalized data. From the forward selection search, the unnormalized data and the 2-class normalized data have the same best feature set and accuracy results. For the backward elimination search, all unnormalized and normalized data are had the same best feature subset, but the unnormalized data has the highest accuracy. This means the normalized data cause the accuracy drops. In last, we conclude that there is no effect of normalization in forward feature selection search. And the effect of normalization in backward elimination search causes the accuracy drop. Hence, the data of normalization would not help to improve the accuracy in this project.

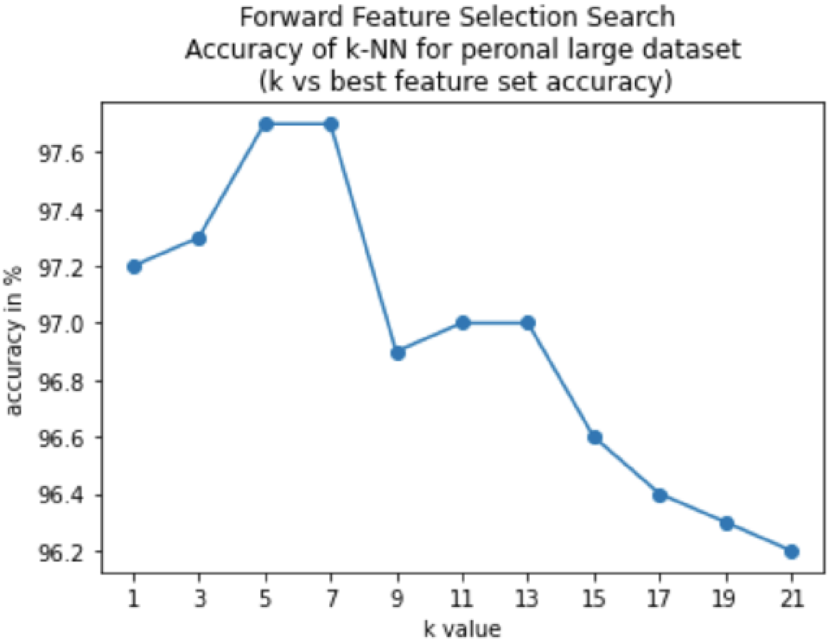
(Since our custom algorithm is the combination of forward feature selection and backward elimination. So. the statistic of our custom algorithm is the same as forward feature selection. And for the effect of normalization, we just compare the forward feature selection and backward elimination)

Experiment 3: Effect of number neighbors (k)


Forward Selection Search
Accuracy of k-NN for peronal small dataset
(k vs best feature set accuracy)


Backward Elimination Search
Accuracy of k-NN for peronal small dataset
(k vs best feature set accuracy)

The two graphs above are the k value vs best feature set accuracy in two search algorithms. We only consider the k old number value, which can prevent the tie situation happen in an even number of the k value. For forward feature selection search, the accuracy increases as the k value increases but it's only a small amount of percentage increase and there is an accuracy drop at k value = 7. So the best k-nn value for forward feature selection is 5. For the backward elimination search, the accuracy increase as the k value increase but it stops increasing at k =5. So, the best k-nn value for backward elimination search is 5. We conclude that using the current k nearest neighbor can help us improve the accuracy of the search for the small dataset. (Again, the custom algorithm has the same statistic as forward feature selection algorithm)



By observing the graph above, for my personal largest dataset, the best k-nn value is 7, because we had the highest accuracy when k = 7 and there is a shock drop of accuracy from 7 to 9. (Since). This proves that the k-nn value has an impact on improving the accuracy of our best feature set. Since the backward elimination algorithm takes more than 1-2 hours for each k value search, I'm not able to get the plot of the backward elimination. But from the small dataset, we saw an improvement in accuracy by increasing and choosing the right k-nn value in all algorithms of the small dataset. Therefore, we conclude that the k nearest neighbor had an effect on the improvement of our accuracy for both small and large datasets.

VII. Conclusion

In my conclusion, the forward feature selection algorithm has better accuracy and searches time than the backward elimination algorithm in both small and large datasets. For the data normalization, it doesn't help us to improve the accuracy, and it decreases the accuracy of the backward elimination search. For k-nn value, the right k-nn value helps us to improve the accuracy of the best subset of features among all algorithms. Moreover, there are search time-efficient problems that backward elimination has. It's the cost of the huge amount of time to find the best feature subset and their result has lower accuracy than the forward feature selection algorithms.

Although backward elimination is doing poorly than the forward feature selection in this project’s small and large dataset. But the elimination of features is a good idea. So, I created custom algorithms that combine both forward feature selection and backward elimination algorithms. This algorithm would add features as the accuracy increase, if any of the features add and cause the accuracy to drop then they would be eliminated and never be considered to be added in the rest of the search. As result, this algorithm results in the same as the forward feature selection algorithm but it had a faster search time.

VIII. Trace of your small dataset

Forward feature selection

```
Welcome to Jianeng Yang Feature Selection Algorithm

Enter 1 to chose the small dataset

Enter 2 to chose the large dataset

---- 1.7848944664001465 seconds ----

Chose type of the algorithm

Enter 1 for Forward Selection

Enter 2 for Backward Elimination

Enter 3 for Jianeng Yang(Bidirection Elimination)'s Special Algorithm.

---- 2.2506308555603027 seconds ----

Enter 1 to chose all features

Enter 2 to input your subset of features

Enter your number of neighbors(k-nn):

You chose the small dataset, this dataset has 10 features, with  100  instances. And k-nn is 1

---- 3.2999014854431152 seconds ----

Using no features and "random" evalution, I get an accuracy of  18.0 %


Beginning search (Forward Selection):

Using features(s) [1] accuracy is  70.0 %

Using features(s) [2] accuracy is  65.0 %
Using features(s) [3] accuracy is  70.0 %
Using features(s) [4] accuracy is  78.0 %
Using features(s) [5] accuracy is  69.0 %
Using features(s) [6] accuracy is  69.0 %
Using features(s) [7] accuracy is  64.0 %
Using features(s) [8] accuracy is  68.0 %
Using features(s) [9] accuracy is  80.0 %
Using features(s) [10] accuracy is  66.0 %

Feature set  [9]  was best, accuracy is  80.0 %
Using features(s) [1, 9] accuracy is  77.0 %
Using features(s) [2, 9] accuracy is  80.0 %
Using features(s) [3, 9] accuracy is  82.0 %
Using features(s) [4, 9] accuracy is  93.0 %
Using features(s) [5, 9] accuracy is  81.0 %
Using features(s) [6, 9] accuracy is  80.0 %
Using features(s) [7, 9] accuracy is  81.0 %
Using features(s) [8, 9] accuracy is  79.0 %
Using features(s) [10, 9] accuracy is  79.0 %

Feature set  [4, 9]  was best, accuracy is  93.0 %
Using features(s) [1, 4, 9] accuracy is  87.0 %
Using features(s) [2, 4, 9] accuracy is  89.0 %
Using features(s) [3, 4, 9] accuracy is  81.0 %
Using features(s) [5, 4, 9] accuracy is  82.0 %
Using features(s) [6, 4, 9] accuracy is  85.0 %
Using features(s) [7, 4, 9] accuracy is  92.0 %
Using features(s) [8, 4, 9] accuracy is  83.0 %
Using features(s) [10, 4, 9] accuracy is  86.0 %
Warning: the accuracy is not increasing!!!
Finished search!!  The best features are  [4, 9]  which has accuracy  93.0 %



---- 3.9488160610198975 seconds ----
```

Backward Elimination

```
Welcome to Jianeng Yang Feature Selection Algorithm
Enter 1 to chose the small dataset
Enter 2 to chose the large dataset
---- 2.0544769763946533 seconds ----
Chose type of the algorithm
Enter 1 for Forward Selection
Enter 2 for Backward Elimination
Enter 3 for Jianeng Yang(Bidirection Elimination)'s Special Algorithm.
---- 3.6363964080810547 seconds ----
Enter 1 to chose all features
Enter 2 to input your subset of features
Enter your number of neighbors(k-nn):
You chose the small dataset, this dataset has 10 features, with  100  instances. And k-nn is 1
---- 5.942071199417114 seconds ----
Using all feature(s) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] , I get an accuracy of  83.0 %

Beginning search (Backward Elimination):
Using features(s) [2, 3, 4, 5, 6, 7, 8, 9, 10] accuracy is  76.0 %
Using features(s) [1, 3, 4, 5, 6, 7, 8, 9, 10] accuracy is  79.0 %
Using features(s) [1, 2, 4, 5, 6, 7, 8, 9, 10] accuracy is  78.0 %
Using features(s) [1, 2, 3, 5, 6, 7, 8, 9, 10] accuracy is  80.0 %
Using features(s) [1, 2, 3, 4, 6, 7, 8, 9, 10] accuracy is  73.0 %
Using features(s) [1, 2, 3, 4, 5, 7, 8, 9, 10] accuracy is  77.0 %
Using features(s) [1, 2, 3, 4, 5, 6, 8, 9, 10] accuracy is  75.0 %
Using features(s) [1, 2, 3, 4, 5, 6, 7, 9, 10] accuracy is  77.0 %
Using features(s) [1, 2, 3, 4, 5, 6, 7, 8, 10] accuracy is  78.0 %
Using features(s) [1, 2, 3, 4, 5, 6, 7, 8, 9] accuracy is  72.0 %
Warning: the accuracy is not increasing!!!
Finished search!!  The best features are  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  which has accuracy
83.0 %



---- 6.525073289871216 seconds ----
```

Custom Algorithm(Bidirection Elimination)

```
Welcome to Jianeng Yang Feature Selection Algorithm
Enter 1 to chose the small dataset
Enter 2 to chose the large dataset
---- 1.9872992038726807 seconds ----
Chose type of the algorithm
Enter 1 for Forward Selection
Enter 2 for Backward Elimination
Enter 3 for Jianeng Yang(Bidirection Elimination)'s Special Algorithm.
---- 3.2645673751831055 seconds ----
Enter 1 to chose all features
Enter 2 to input your subset of features
Enter your number of neighbors(k-nn):
You chose the small dataset, this dataset has 10 features, with  100  instances. And k-nn is
1
```

```
---- 4.270010232925415 seconds ----
Using no features and "random" evalution, I get an accuracy of  18.0 %


Beginning search(Bidirection Elimination)
Using features(s) [1] accuracy is  70.0 %
Using features(s) [2] accuracy is  65.0 %
Using features(s) [3] accuracy is  70.0 %
Using features(s) [4] accuracy is  78.0 %
Using features(s) [5] accuracy is  69.0 %
Using features(s) [6] accuracy is  69.0 %
Using features(s) [7] accuracy is  64.0 %
Using features(s) [8] accuracy is  68.0 %
Using features(s) [9] accuracy is  80.0 %
Using features(s) [10] accuracy is  66.0 %
Number of features to be eliminated:  0


Feature set  [9]  was best, accuracy is  80.0 %
Using features(s) [1, 9] accuracy is  77.0 %
Eliminated this feature! Since this feautre(s) [ 1 ] had create the accuracy  77.0 % is lower than previous best features accuracy  80.0 %
Using features(s) [2, 9] accuracy is  80.0 %
Using features(s) [3, 9] accuracy is  82.0 %
Using features(s) [4, 9] accuracy is  93.0 %
Using features(s) [5, 9] accuracy is  81.0 %
Using features(s) [6, 9] accuracy is  80.0 %
Using features(s) [7, 9] accuracy is  81.0 %
Using features(s) [8, 9] accuracy is  79.0 %
Eliminated this feature! Since this feautre(s) [ 8 ] had create the accuracy  79.0 % is lower than previous best features accuracy  80.0 %
```

```
Using features(s) [10, 9] accuracy is  79.0 %
Eliminated this feature! Since this feautre(s) [ 10 ] had create the accuracy  79.0 % is lower than previous best features accuracy  80.0 %
Number of features to be eliminated:  3


Feature set  [4, 9]  was best, accuracy is  93.0 %
Using features(s) [2, 4, 9] accuracy is  89.0 %
Eliminated this feature! Since this feautre(s) [ 2 ] had create the accuracy  89.0 % is lower than previous best features accuracy  93.0 %
Using features(s) [3, 4, 9] accuracy is  81.0 %
Eliminated this feature! Since this feautre(s) [ 3 ] had create the accuracy  81.0 % is lower than previous best features accuracy  93.0 %
Using features(s) [5, 4, 9] accuracy is  82.0 %
Eliminated this feature! Since this feautre(s) [ 5 ] had create the accuracy  82.0 % is lower than previous best features accuracy  93.0 %
Using features(s) [6, 4, 9] accuracy is  85.0 %
Eliminated this feature! Since this feautre(s) [ 6 ] had create the accuracy  85.0 % is lower than previous best features accuracy  93.0 %
Using features(s) [7, 4, 9] accuracy is  92.0 %
Eliminated this feature! Since this feautre(s) [ 7 ] had create the accuracy  92.0 % is lower than previous best features accuracy  93.0 %
Number of features to be eliminated:  5
We used and eliminated all features!


Finished search!! The best accuracy is  The best features are  [4, 9]  which has accuracy  93.0 %



---- 4.817439794540405 seconds ----
```