Prudhvi Manukonda
SID: 862325916
Email:pmanu005@ucr.edu
June 10th 2022

Project 2 for CS 205 Fall 2022 with Dr Eammon Keogh.

All Code is Original.

In selecting features we can use two greedy approaches one is Forward selection and Backward Elimination.

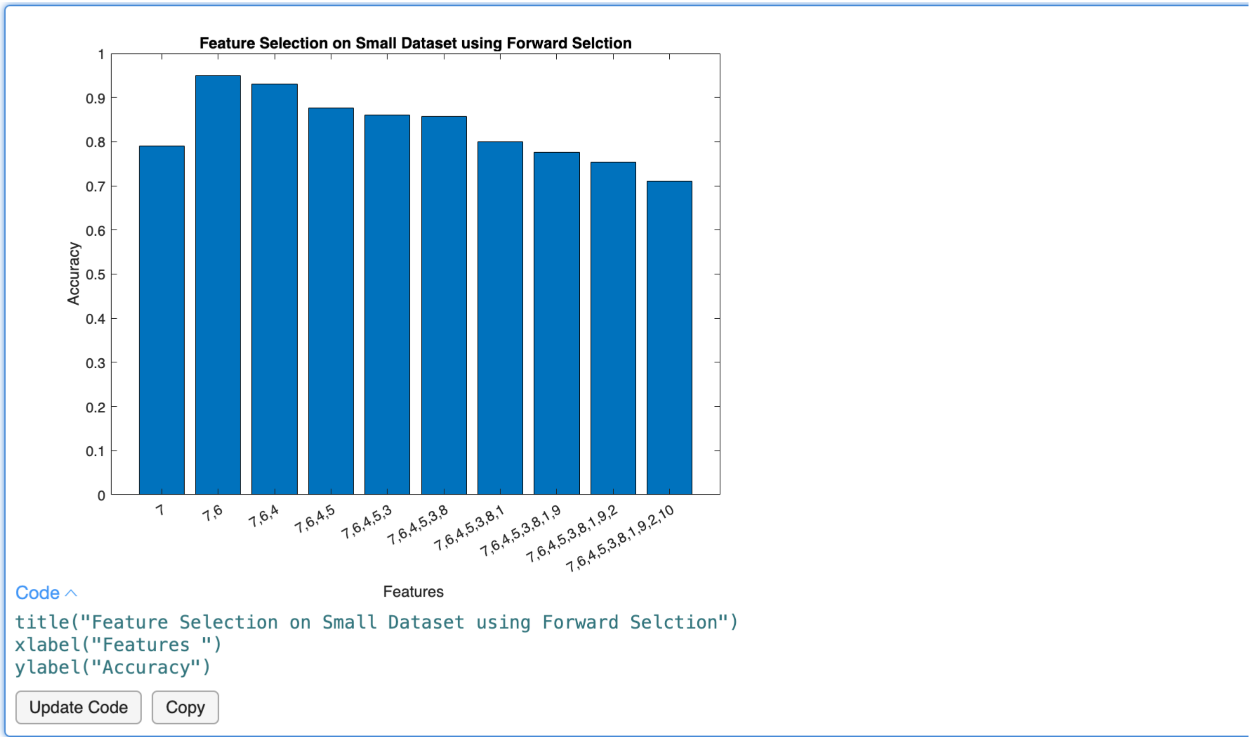
Forward Selection:Forward selection is nothing but selecting the best feature in current level of features and then in next level selecting the next best feature finally including the best features in the whole features.

Backward Elimination:Backward Elimination is nothing but removing the weak feature in current level of features and then in the next level removing the next weak feature finally excluding the weak features from the whole features.

K nearest Neighbour : Nearest Neigbour algorithm states that K nearest neighbour will have the same labels.

In this project I was assigned small dataset 26 and large dataset 44 for feature selection using forward selection, backward elimination techniques using K nearest neighbor algorithm.I have implemented K nearest algorithm with k=1 using matlab used euclidean distance as metric.

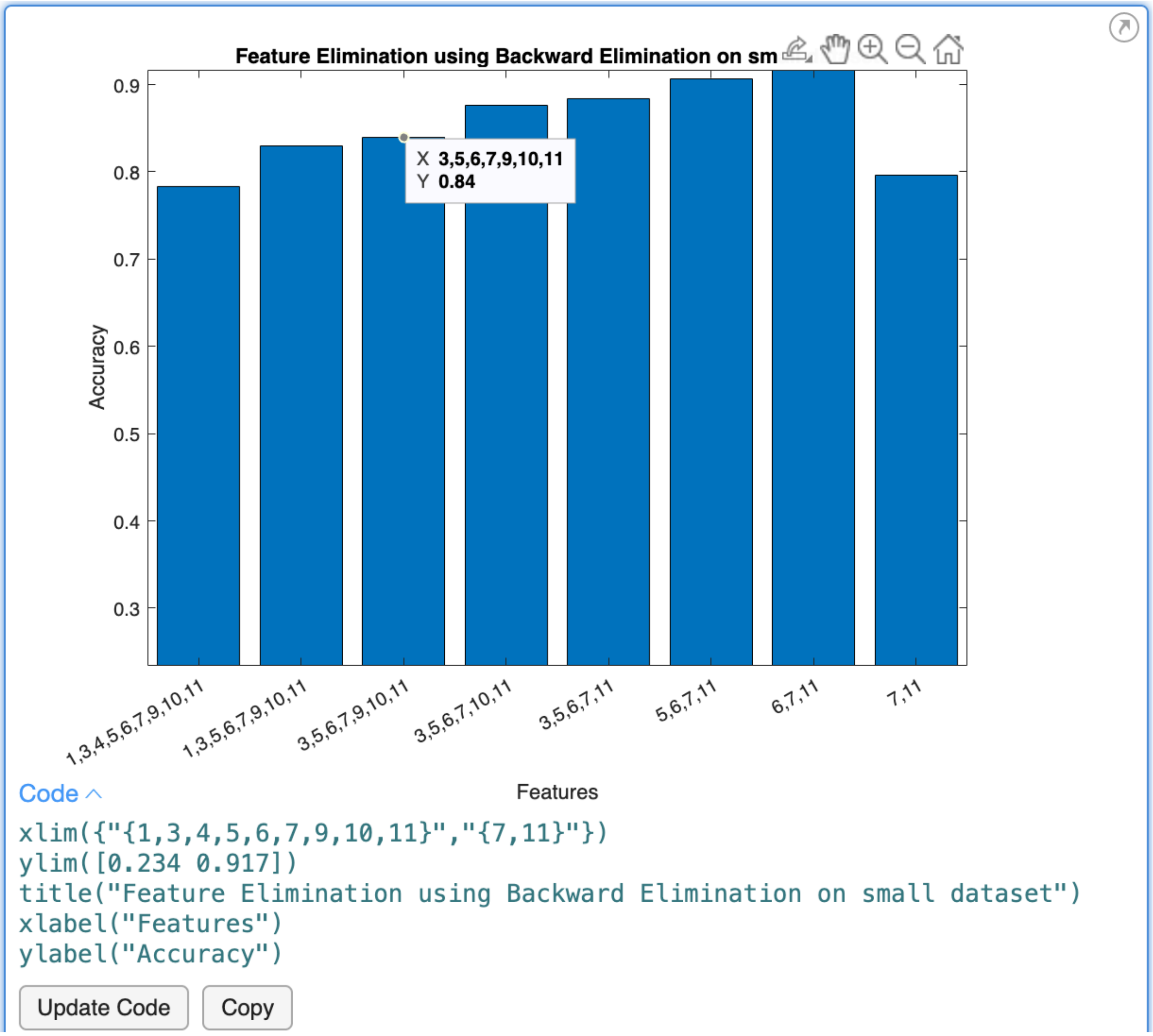
In Figure 1 shows the forward selection algorithm results for the small dataset which has **95%** accuracy using **{6,7}** strong features.



```
title("Feature Selection on Small Dataset using Forward Selction")
xlabel("Features ")
ylabel("Accuracy")
```

Fig 1)

In Figure 2 shows the backward elimination algorithm results for the small dataset which has **95%** accuracy using **{6,7,11}** strong features.



```
xlim({"{1,3,4,5,6,7,9,10,11}","{7,11}"})
ylim([0.234 0.917])
title("Feature Elimination using Backward Elimination on small dataset")
xlabel("Features")
ylabel("Accuracy")
```

Figure 2)

Conclusion For Small Dataset: Using the both forward selection and backward elimination we got 95% accuracy but using forward selection we got **{6,7}** whereas using backward selection we get **{6,7,11}** as best features.

In Figure 3 shows the forward selection algorithm results for the large dataset which has 97% accuracy..In the below figure 3) where ylabel indicates the accuracy at the corresponding features, and x label indicates the index of the feature_array which contains features up to that level. So by seeing the graph we got to know that at i=2 it has maximum value .So at corresponding i=2 in the feature_array shown in figure 4) represents**{37,7}** as the best features. Current_features_array shown in Fig 4.From the Graph we got to know that at index x=2 the maximum accuracy occurs ,so index 2 in current_features array indicates **{37,7}** so they are the strong features.

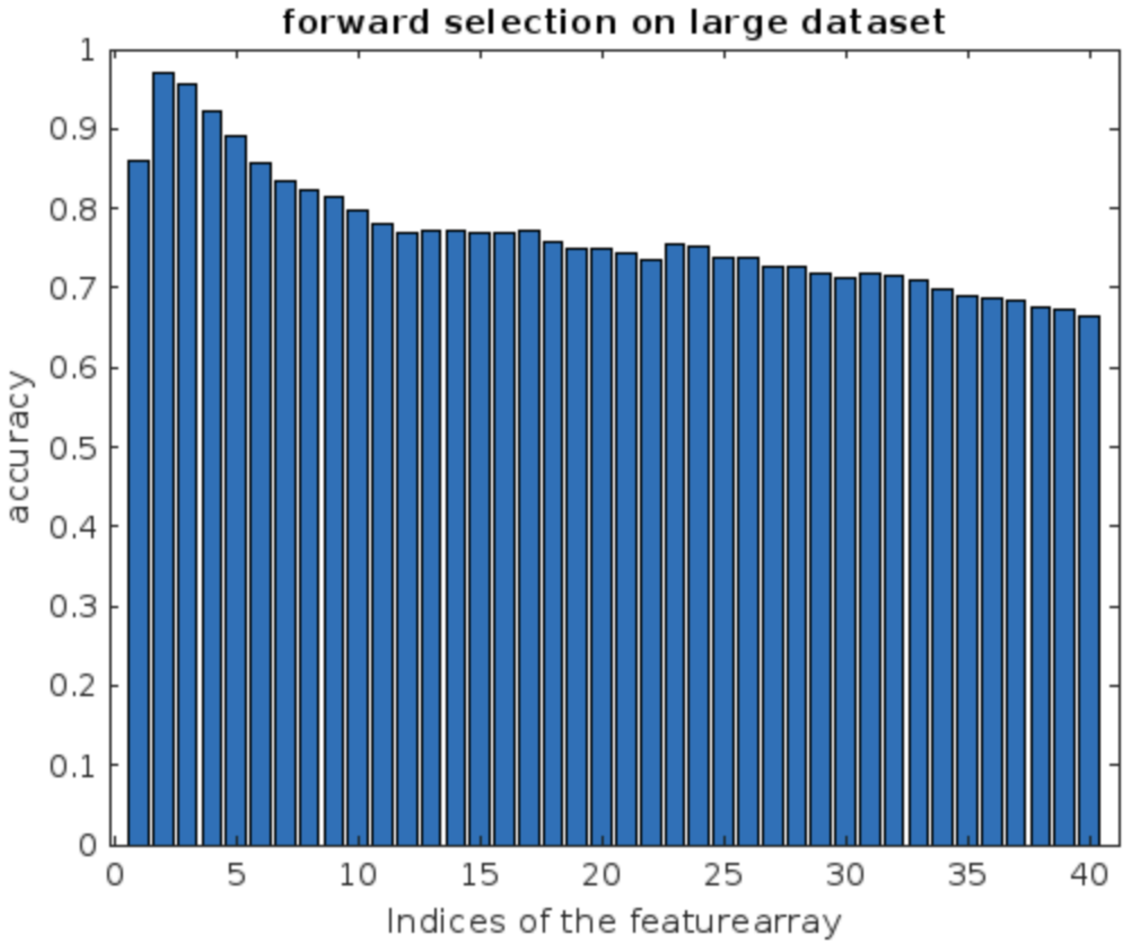
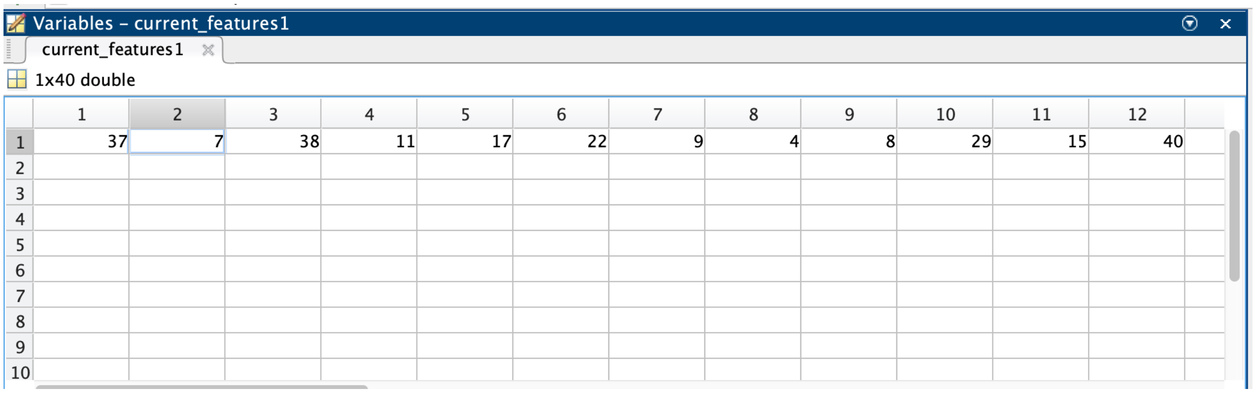
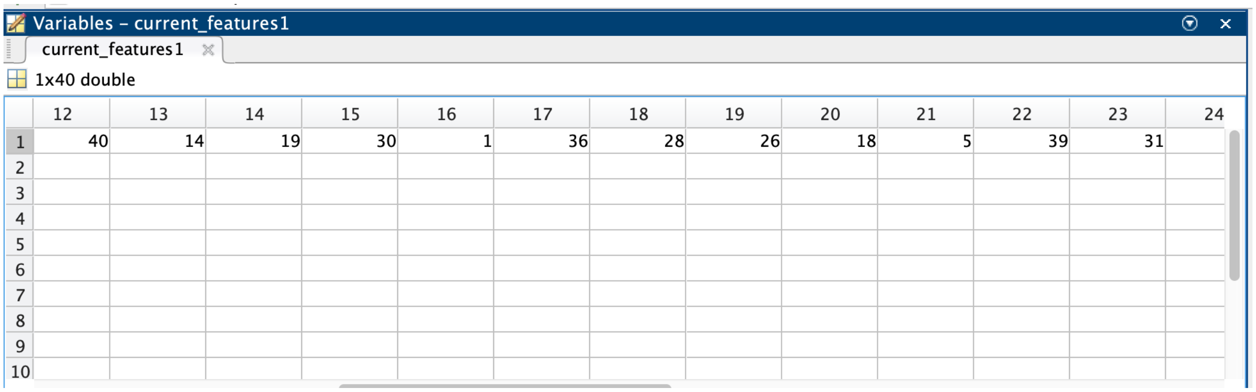


Fig 3)



1x40 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	37	7	38	11	17	22	9	4	8	29	15	40
2												
3												
4												
5												
6												
7												
8												
9												
10												

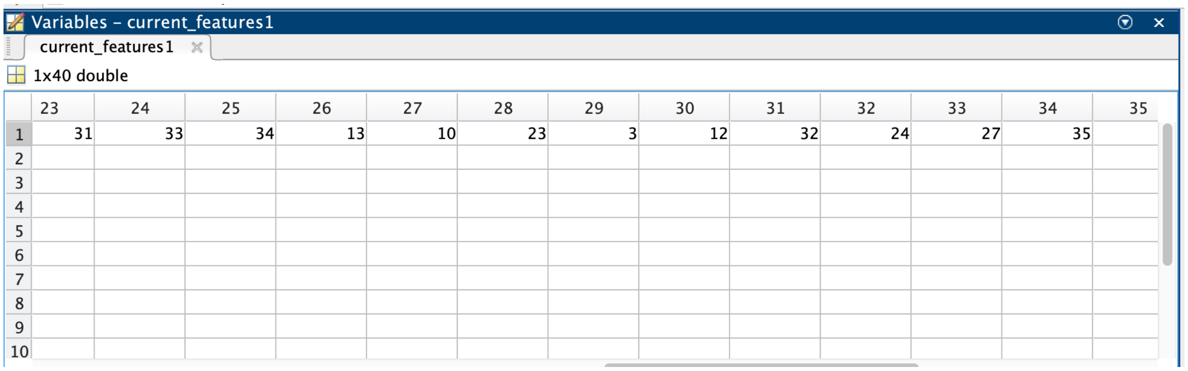


Variables – current_features1

current_features1

1x40 double

	12	13	14	15	16	17	18	19	20	21	22	23	24
1	40	14	19	30	1	36	28	26	18	5	39	31	
2													
3													
4													
5													
6													
7													
8													
9													
10													

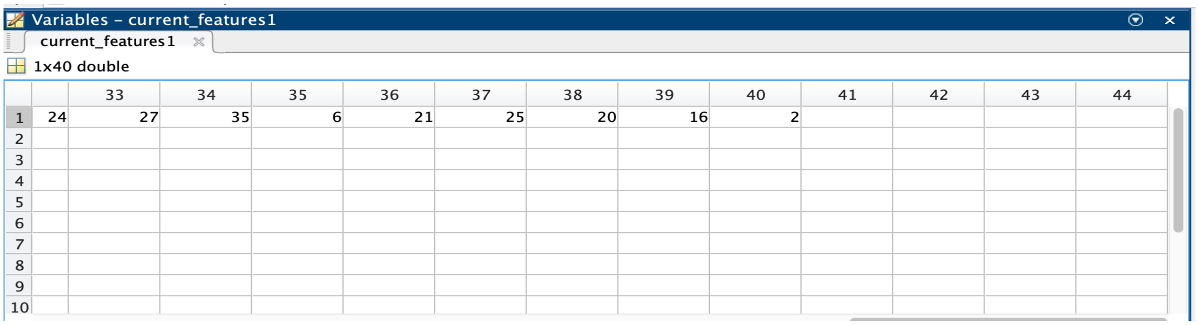


Variables – current_features1

current_features1

1x40 double

	23	24	25	26	27	28	29	30	31	32	33	34	35
1	31	33	34	13	10	23	3	12	32	24	27	35	
2													
3													
4													
5													
6													
7													
8													
9													
10													



Variables – current_features1

current_features1

1x40 double

		33	34	35	36	37	38	39	40	41	42	43	44
1	24	27	35	6	21	25	20	16	2				
2													
3													
4													
5													
6													
7													
8													
9													
10													

Fig 4)

In Figure 5 shows the backward elimination algorithm results for the large dataset which has -accuracy using {} this strong features.In the below figure where ylabel indicates the accuracy at the corresponding features, and x label indicates the index of the feature_removed_array which contains features removed up to that level. So by seeing the graph we got to know that at i=38 it has maximum value .So at index 38 in the feature_removed array shown in the figure 6) represents{9,8,16,24,22,35,14,25,2,27,21,18,31,29,17,26,13,5,19,33,36,30,10,39,6,23,15,11,3,4,32,12,34,20,1,28,40,38}} features. By removing those features from the whole features(1-40) we got **{7,37}** as strong features.

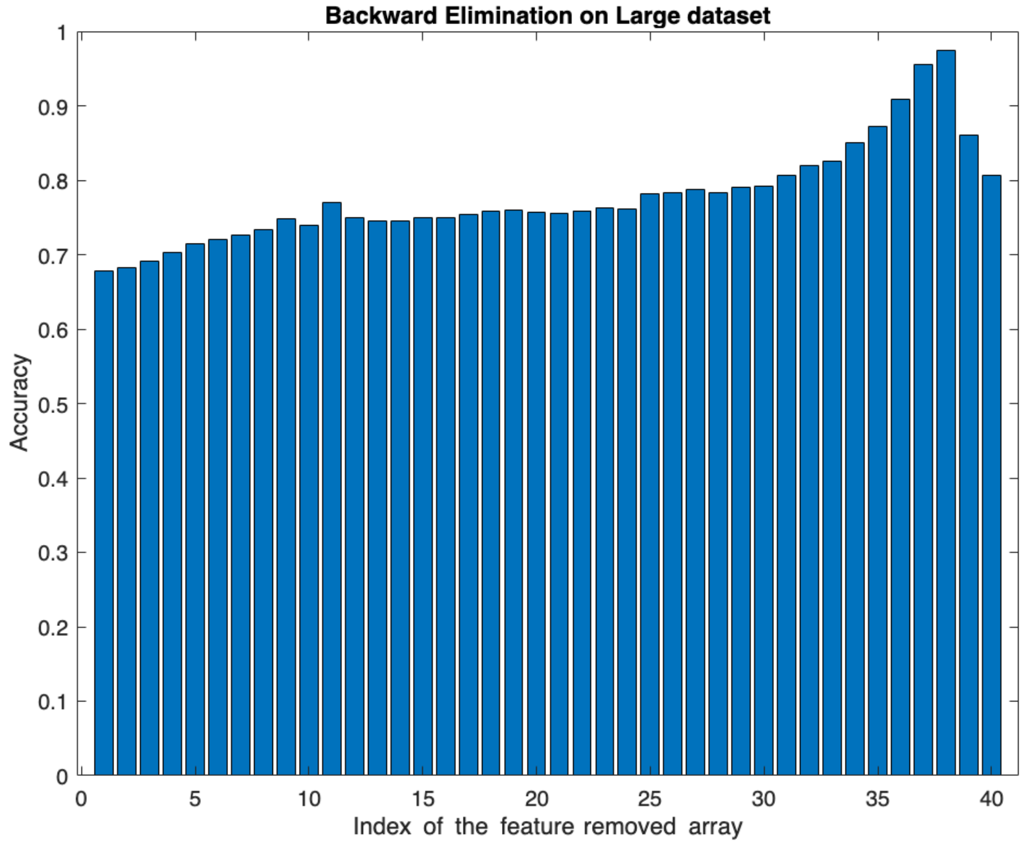
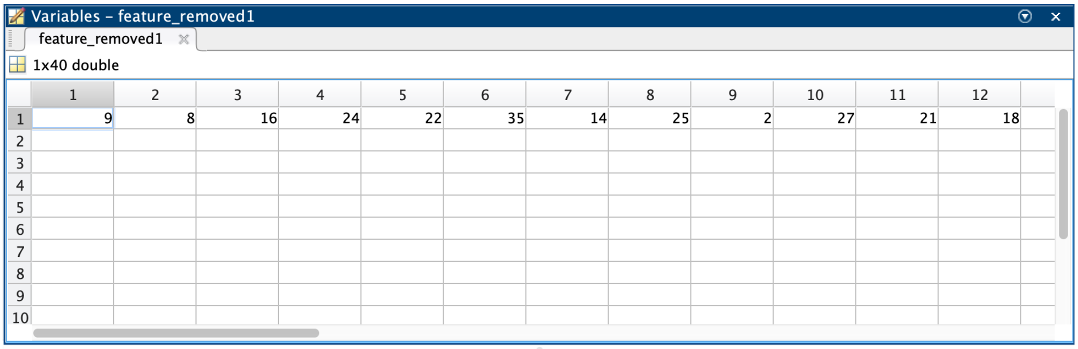
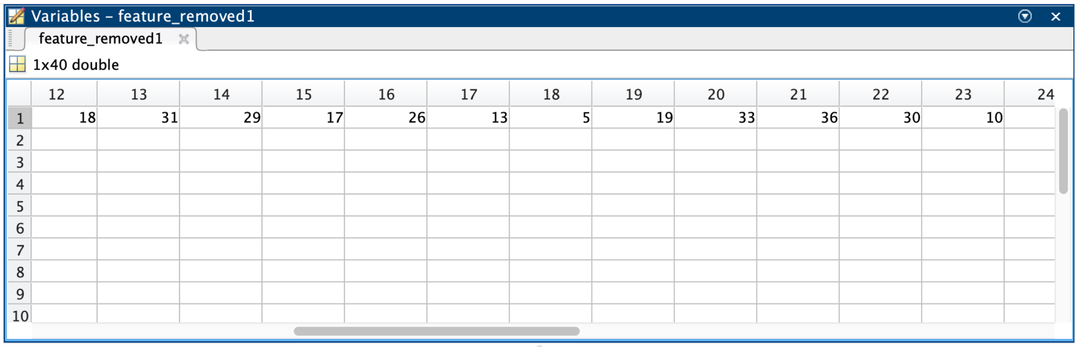

Backward Elimination on Large dataset
Accuracy
Index of the feature removed array
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25
30
35
40

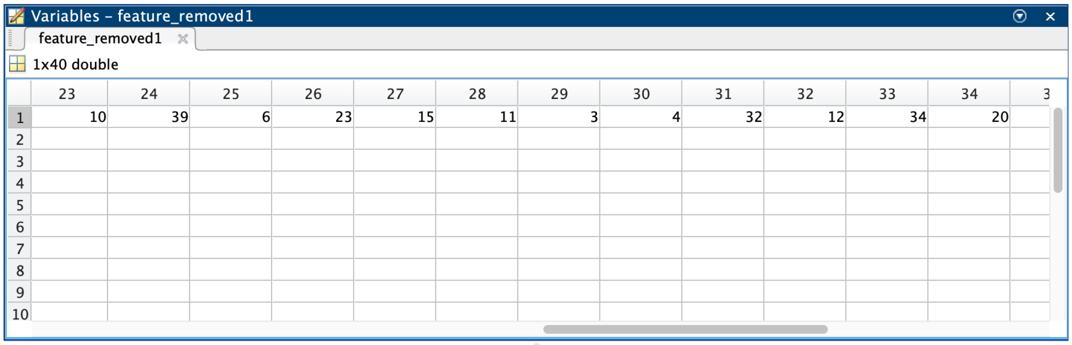
Fig 5)



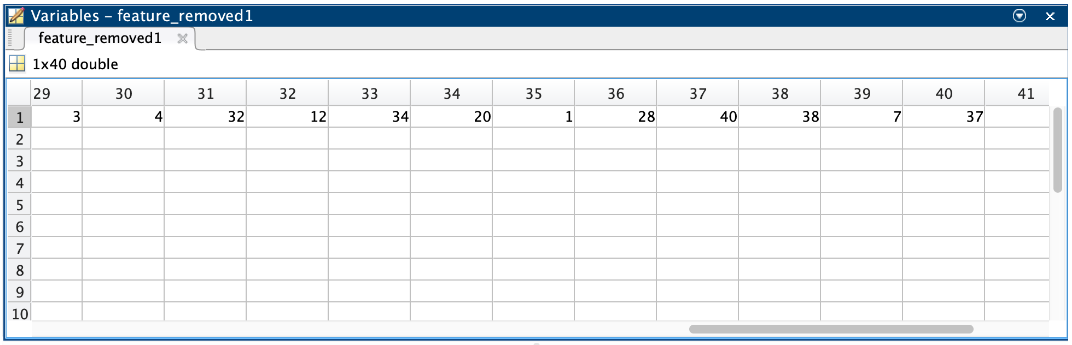
	1	2	3	4	5	6	7	8	9	10	11	12
1	9	8	16	24	22	35	14	25	2	27	21	18
2												
3												
4												
5												
6												
7												
8												
9												
10												



	12	13	14	15	16	17	18	19	20	21	22	23	24
1	18	31	29	17	26	13	5	19	33	36	30	10	
2													
3													
4													
5													
6													
7													
8													
9													
10													



	23	24	25	26	27	28	29	30	31	32	33	34	3
1	10	39	6	23	15	11	3	4	32	12	34	20	
2													
3													
4													
5													
6													
7													
8													
9													
10													

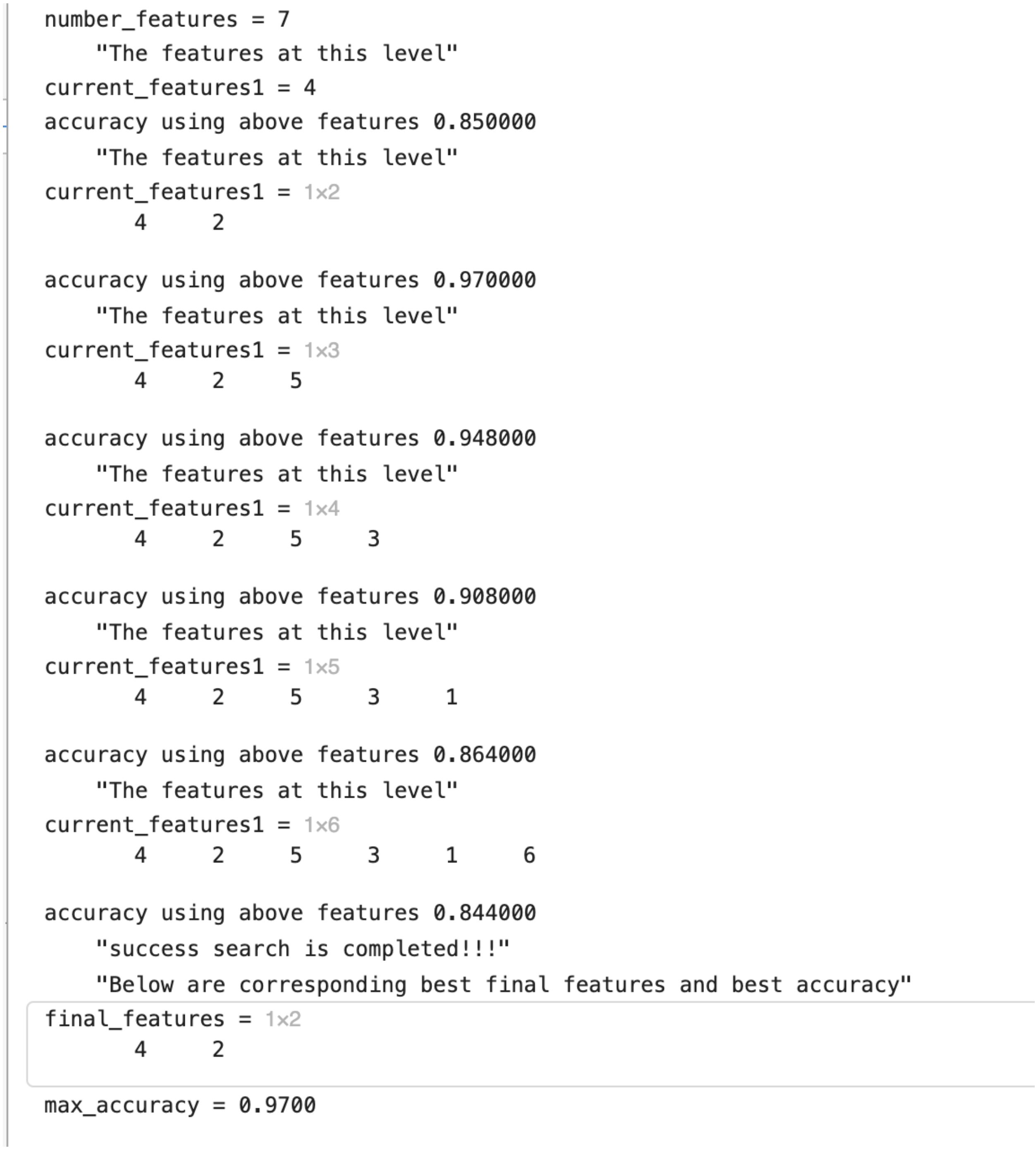


	29	30	31	32	33	34	35	36	37	38	39	40	41
1	3	4	32	12	34	20	1	28	40	38	7	37	
2													
3													
4													
5													
6													
7													
8													
9													
10													

Fig 6)

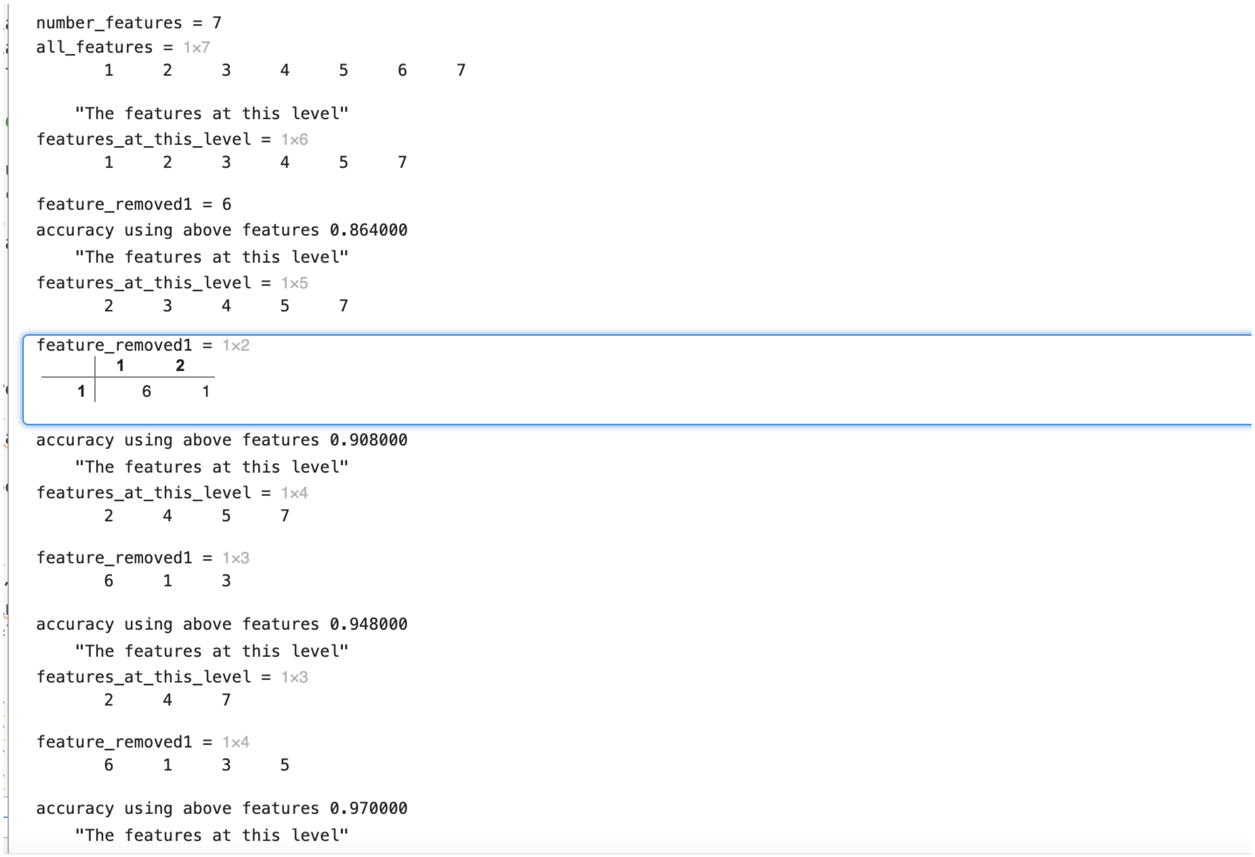
Conclusion On Large Data Set:
Both Forward Selection and Backward Elimination worked well on large datasets got accuracy about 97.4 and 97 accuracy respectively. Both forward selection and backward elimination declared **{7,37}** as best features.

To check my algorithm I have implemented my algorithm on cs205_calibration dataset1. Below shown the trace of the algorithm on calibration data using forward selection.

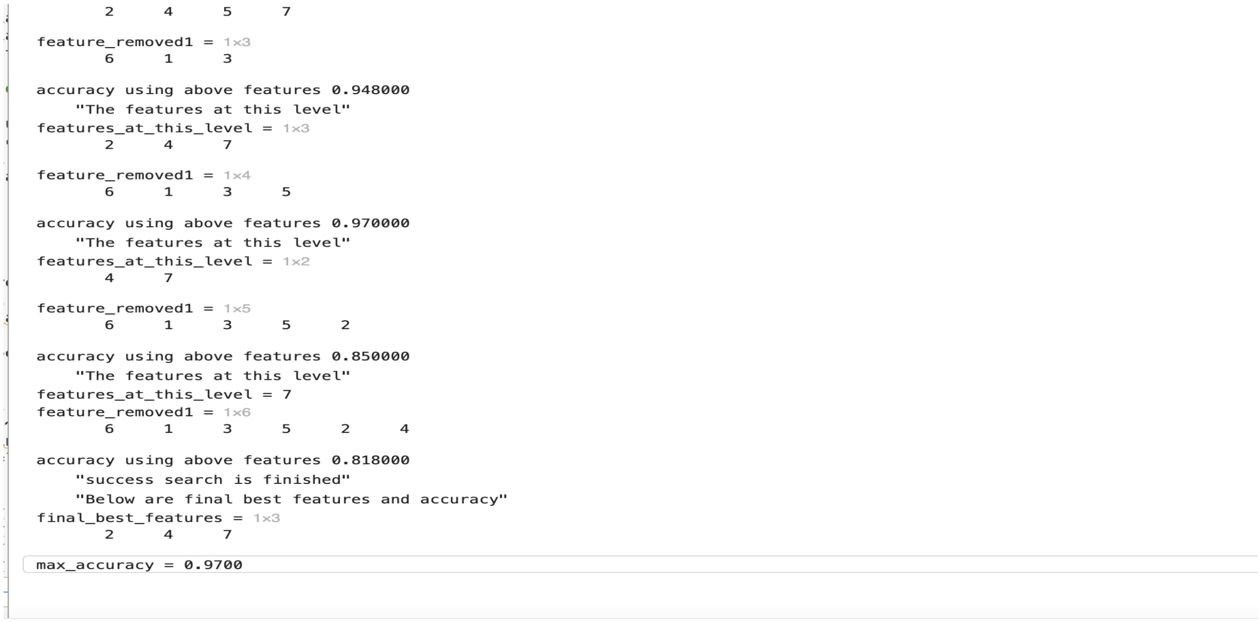


```
number_features = 7
    "The features at this level"
current_features1 = 4
accuracy using above features 0.850000
    "The features at this level"
current_features1 = 1×2
       4     2

accuracy using above features 0.970000
    "The features at this level"
current_features1 = 1×3
       4     2     5

accuracy using above features 0.948000
    "The features at this level"
current_features1 = 1×4
       4     2     5     3

accuracy using above features 0.908000
    "The features at this level"
current_features1 = 1×5
       4     2     5     3     1

accuracy using above features 0.864000
    "The features at this level"
current_features1 = 1×6
       4     2     5     3     1     6

accuracy using above features 0.844000
    "success search is completed!!!"
    "Below are corresponding best final features and best accuracy"
final_features = 1×2
       4     2

max_accuracy = 0.9700
```

Below shown the trace of the algorithm on calibration data using Backward Elimination.



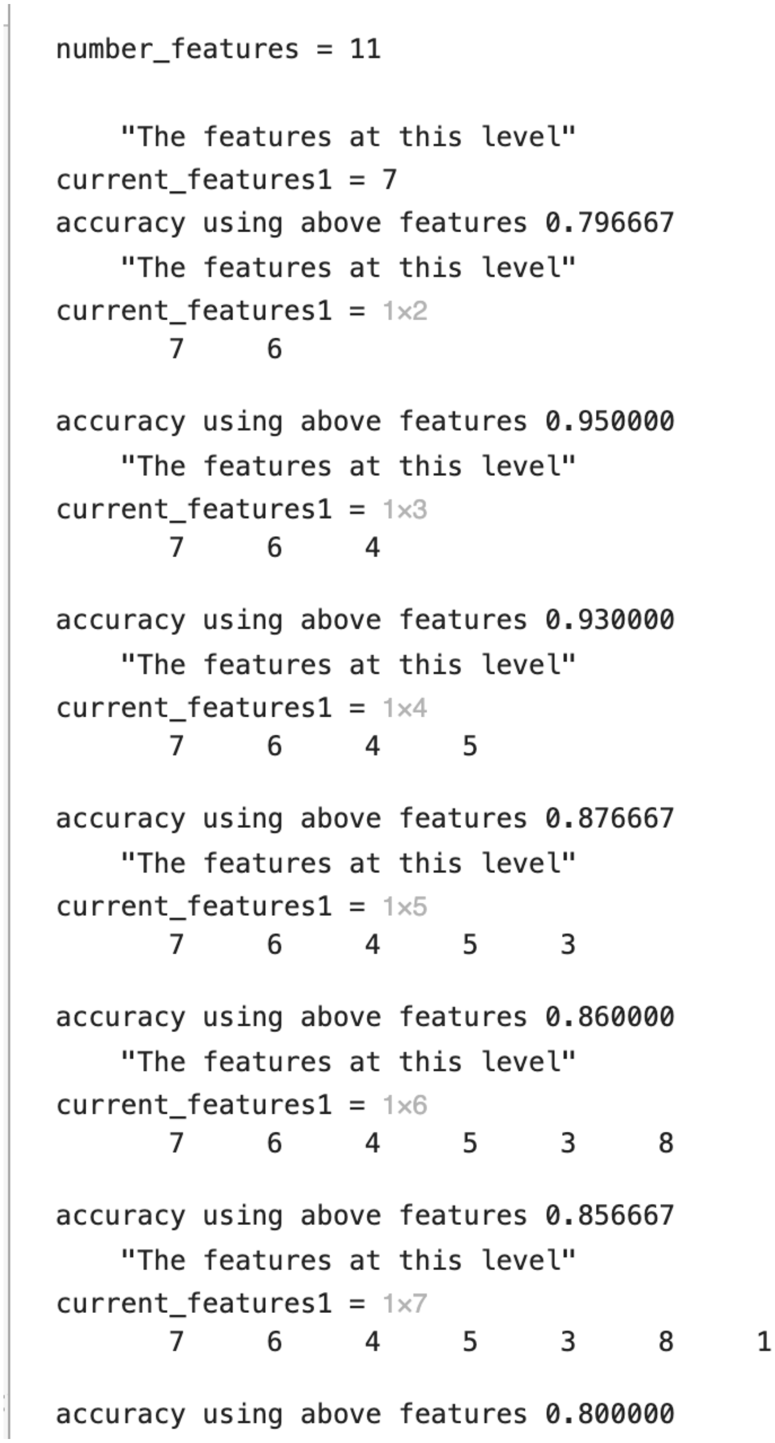
```
number_features = 7
all_features = 1×7
       1     2     3     4     5     6     7

    "The features at this level"
features_at_this_level = 1×6
       1     2     3     4     5     7

feature_removed1 = 6
accuracy using above features 0.864000
    "The features at this level"
features_at_this_level = 1×5
       2     3     4     5     7

feature_removed1 = 1×2
          1     2
     1    6     1

accuracy using above features 0.908000
    "The features at this level"
features_at_this_level = 1×4
       2     4     5     7

feature_removed1 = 1×3
       6     1     3

accuracy using above features 0.948000
    "The features at this level"
features_at_this_level = 1×3
       2     4     7

feature_removed1 = 1×4
       6     1     3     5

accuracy using above features 0.970000
    "The features at this level"
```



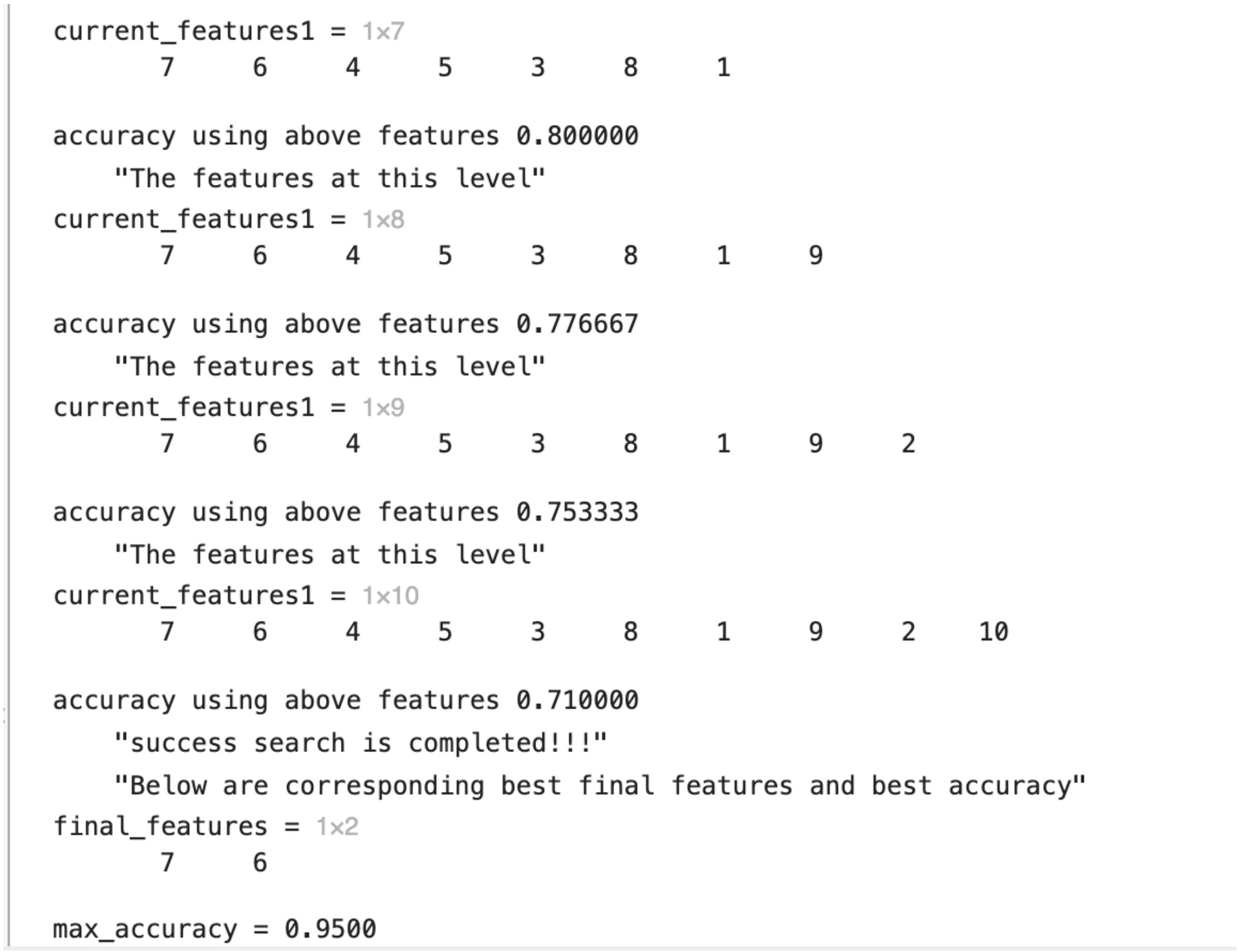
```
       2     4     5     7

feature_removed1 = 1×3
       6     1     3

accuracy using above features 0.948000
    "The features at this level"
features_at_this_level = 1×3
       2     4     7

feature_removed1 = 1×4
       6     1     3     5

accuracy using above features 0.970000
    "The features at this level"
features_at_this_level = 1×2
       4     7

feature_removed1 = 1×5
       6     1     3     5     2

accuracy using above features 0.850000
    "The features at this level"
features_at_this_level = 7
feature_removed1 = 1×6
       6     1     3     5     2     4

accuracy using above features 0.818000
    "success search is finished"
    "Below are final best features and accuracy"
final_best_features = 1×3
       2     4     7

max_accuracy = 0.9700
```

Conclusion on Calibration Data: **Got 97 percent accuracy with {4,2} using forward selection and {4,2,7} using Backward Elimination features which matched with the given calibration features.**

Trace on small data set using forward selection technique:

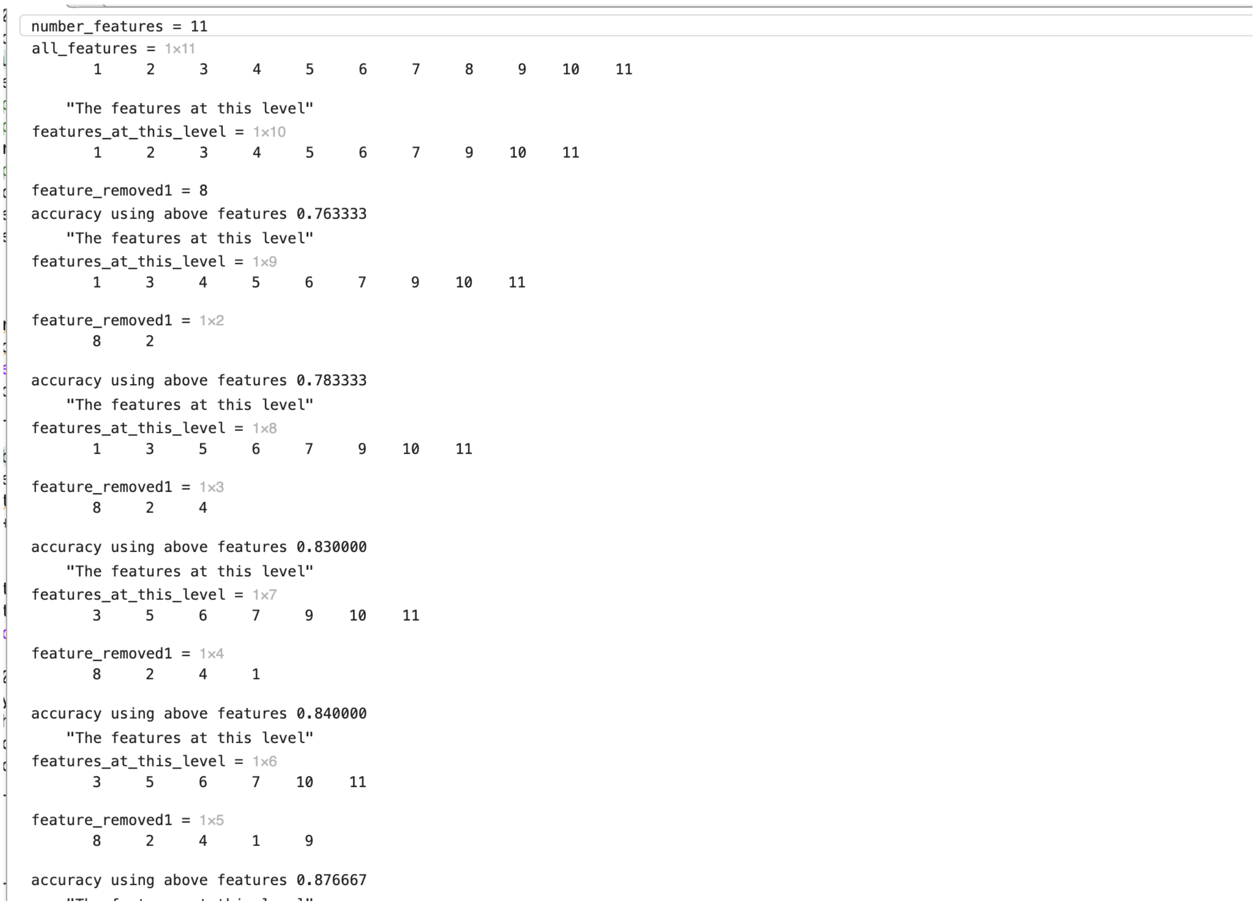


```
number_features = 11

    "The features at this level"
current_features1 = 7
accuracy using above features 0.796667
    "The features at this level"
current_features1 = 1×2
       7     6

accuracy using above features 0.950000
    "The features at this level"
current_features1 = 1×3
       7     6     4

accuracy using above features 0.930000
    "The features at this level"
current_features1 = 1×4
       7     6     4     5

accuracy using above features 0.876667
    "The features at this level"
current_features1 = 1×5
       7     6     4     5     3

accuracy using above features 0.860000
    "The features at this level"
current_features1 = 1×6
       7     6     4     5     3     8

accuracy using above features 0.856667
    "The features at this level"
current_features1 = 1×7
       7     6     4     5     3     8     1

accuracy using above features 0.800000
```

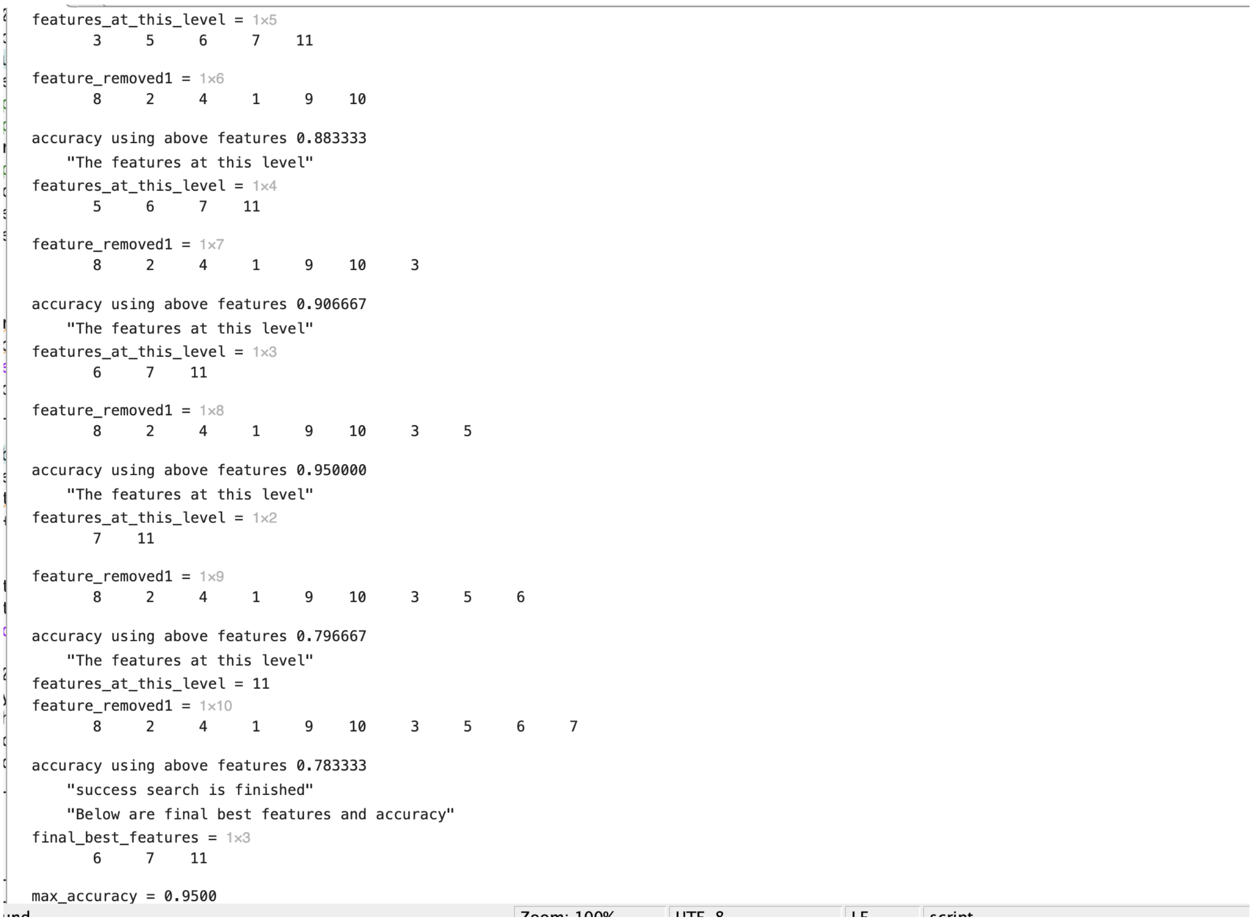


```
current_features1 = 1×7
       7     6     4     5     3     8     1

accuracy using above features 0.800000
    "The features at this level"
current_features1 = 1×8
       7     6     4     5     3     8     1     9

accuracy using above features 0.776667
    "The features at this level"
current_features1 = 1×9
       7     6     4     5     3     8     1     9     2

accuracy using above features 0.753333
    "The features at this level"
current_features1 = 1×10
       7     6     4     5     3     8     1     9     2    10

accuracy using above features 0.710000
    "success search is completed!!!"
    "Below are corresponding best final features and best accuracy"
final_features = 1×2
       7     6

max_accuracy = 0.9500
```

Trace Of Algorithm on small dataset using Backward Elimination



```
number_features = 11
all_features = 1×11
       1     2     3     4     5     6     7     8     9    10    11

    "The features at this level"
features_at_this_level = 1×10
       1     2     3     4     5     6     7     9    10    11

feature_removed1 = 8
accuracy using above features 0.763333
    "The features at this level"
features_at_this_level = 1×9
       1     3     4     5     6     7     9    10    11

feature_removed1 = 1×2
       8     2

accuracy using above features 0.783333
    "The features at this level"
features_at_this_level = 1×8
       1     3     5     6     7     9    10    11

feature_removed1 = 1×3
       8     2     4

accuracy using above features 0.830000
    "The features at this level"
features_at_this_level = 1×7
       3     5     6     7     9    10    11

feature_removed1 = 1×4
       8     2     4     1

accuracy using above features 0.840000
    "The features at this level"
features_at_this_level = 1×6
       3     5     6     7    10    11

feature_removed1 = 1×5
       8     2     4     1     9

accuracy using above features 0.876667
    "The features at this level"
```



```
features_at_this_level = 1×5
       3     5     6     7    11

feature_removed1 = 1×6
       8     2     4     1     9    10

accuracy using above features 0.883333
    "The features at this level"
features_at_this_level = 1×4
       5     6     7    11

feature_removed1 = 1×7
       8     2     4     1     9    10     3

accuracy using above features 0.906667
    "The features at this level"
features_at_this_level = 1×3
       6     7    11

feature_removed1 = 1×8
       8     2     4     1     9    10     3     5

accuracy using above features 0.950000
    "The features at this level"
features_at_this_level = 1×2
       7    11

feature_removed1 = 1×9
       8     2     4     1     9    10     3     5     6

accuracy using above features 0.796667
    "The features at this level"
features_at_this_level = 11
feature_removed1 = 1×10
       8     2     4     1     9    10     3     5     6     7

accuracy using above features 0.783333
    "success search is finished"
    "Below are final best features and accuracy"
final_best_features = 1×3
       6     7    11

max_accuracy = 0.9500
```

On a small dataset using backward accuracy got 95 percent accuracy using {6,7,11} accuracy.

Conclusion: On the whole we can conclude ,If we have applied K Nearest algorithm without feature selection it may have led to bad accuracy and bad results ,By forward selection and backward Elimination we got to know the strong features which improved the accuracy of K Nearest Algorithm.

Code Link:

Below is the link to github for this code:

https://github.com/PMANU005/CS-205-Project

Below is the code for the above algorithm. I have used matlab to solve this code.

```
ata=load("CS205_SP_2022_SMALLtestdata__26.txt");

%data=load("CS205_CalibrationData__1.txt");

%data=[[1,3,4,4];[1,4,3,3];[2,3,0,0];[2,4,0,0];];

%defined current_features empty array to track current features

current_features=[];

%defined accuracy_array empty array to track accuracies
```

```
accuracy_array=[];

number_features=length(data(1,:))

for i =1:number_features-1

 %Initializing best accuracy with 0.

   best_so_far_accuracy=0;

   accuracy=0;

   feature_to_add_this_level=0;

   for k=1:number_features-1

       %checking this condition to eliminate duplicates

       if isempty(intersect(current_features,k+1))

        %display(current_features)

        %display(k+1)

        %getting accuracy by calling crossvalidation function

        accuracy=cross_validation(data,current_features,k+1);

        %display(accuracy)

        %finding best accuracy in current level

        if accuracy>best_so_far_accuracy

           best_so_far_accuracy=accuracy;

           feature_to_add_this_level=k+1;

        end

       end

   end

   %current_features empty array to track current features

   %accuracy_array empty array to track accuracies

   accuracy_array(i)=best_so_far_accuracy;

   current_features(i)=feature_to_add_this_level;

   display("The features at this level")

   current_features1=current_features-1;
   display(current_features1);
   fprintf("accuracy using above features %f",best_so_far_accuracy)
end
max_accuracy=0;
index_accuracy=0;
%getting best accuracy from accuracy array and best features
for i=1:length(accuracy_array)
   if accuracy_array(i)>max_accuracy
       max_accuracy=accuracy_array(i);
       index_accuracy=i;
   end
end
final_features=[];
for j=1:index_accuracy
   final_features(j)=current_features1(j);
end
display("success search is completed!!!");
display("Below are corresponding best final features and best accuracy");
display(final_features);
display(max_accuracy);
```

```
%implement the cross validation
function accuracy=cross_validation(data,current_features,k1)
data1=data;
accuracy=0;
%display(k1);
%display(length(current_features));
```

```
number_features=length(data(1,:));
%display(number_features)
for i=1:number_features-1
%making other feature columns as zero.
if ~ismember(i+1,current_features) && i+1~=k1
 data1(:,i+1)=0;
end
end
%display(data1)
number_correctly_classified=0;
for i=1:size(data1,1)
  object_to_classify=data1(i,2:end);
  label_object_to_classify=data1(i,1);
  nearest_neighbour_location=inf;
  nearest_neighbour_distance=inf;
  for k=1:size(data1,1)
     %checking this condition to eliminate comparision with itself.
     if k~=i
        distance=sqrt(sum((object_to_classify-data1(k,2:end)).^2));
        %display(i);
        %display(distance)
        %getting nearest neighbour location and label
        if distance <nearest_neighbour_distance
           nearest_neighbour_distance=distance;
           nearest_neighbour_location=k;
           nearest_neighbour_label=data1(k,1);

           %display(nearest_neighbour_label);
```

```
      end
    end
  end
  if label_object_to_classify == nearest_neighbour_label
    number_correctly_classified=number_correctly_classified+1;
  end
end
accuracy=number_correctly_classified/size(data,1);
%display(accuracy);
end
```

Code For Backward Elimination

```
data=load("CS205_SP_2022_SMALLtestdata__26.txt");
```

```
%data=[[1,3,4,4];[1,4,3,3];[2,3,0,0];[2,4,0,0];];
%defining to keep track of feature removed at current level
feature_removed=[];
accuracy_array=[];
number_features=length(data(1,:))
all_features=[1:1:number_features]
for i =1:number_features-1
   best_so_far_accuracy=0;
   accuracy=0;
   feature_to_add_this_level=0;
   for k=1:number_features-1
```

```
      if isempty(intersect(feature_removed,k+1))
       %display(feature_removed)
       %display(k+1)
       accuracy=cross_validation(data,k+1,feature_removed);
       %display(accuracy)
       if accuracy>best_so_far_accuracy
          best_so_far_accuracy=accuracy;
          feature_to_remove_this_level=k+1;
       end
      end
  end
accuracy_array(i)=best_so_far_accuracy;
feature_removed(i)=feature_to_remove_this_level;
display("The features at this level")
%to get real features from removed features
feature_removed1=feature_removed-1;
features_at_this_level=[];
j=1;
for i=1:number_features
    %checking if current feature is in removed list
    if ~ismember(i,feature_removed1)
        %If it is not present only then adding into the list
        features_at_this_level(j)=i;
        j=j+1;
    end
end
display(features_at_this_level);
display(feature_removed1);
```

```
 fprintf("accuracy using above features %f",best_so_far_accuracy)
end
%to find the maximun accuracy and index from the accuracy array by
iterating the
%array.
max_accuracy=0;
index_accuracy=0;
for i=1:length(accuracy_array)
   if accuracy_array(i)>max_accuracy
       max_accuracy=accuracy_array(i);
       index_accuracy=i;
   end
end
%To get the removed features up to this level.
final_removed_features=[];
for j=1:index_accuracy
   final_removed_features(j)=feature_removed1(j);
end
final_best_features=[];
l=1;
%To get the real features from the removed features.
for j=1:number_features
   if ~ismember(j,final_removed_features)
    final_best_features(l)=j;
    l=l+1;
   end
end
display("success search is finished")
display("Below are final best features and accuracy")
display(final_best_features);
display(max_accuracy);
function accuracy=cross_validation(data,k1,feature_removed)
data1=data;
accuracy=0;
%display(k1);
%display(length(current_features));
number_features=length(data(1,:));
%display(number_features)
for i=1:number_features-1
 %Making zeros if the value is present in the removed array
if ismember(i+1,feature_removed) || i+1 == k1
 data1(:,i+1)=0;
end
end
%display(data1)

number_correctly_classified=0;

for i=1:size(data1,1)
   object_to_classify=data1(i,2:end);
   label_object_to_classify=data1(i,1);
   nearest_neighbour_location=inf;
   nearest_neighbour_distance=inf;
   for k=1:size(data1,1)
       if k~=i
           distance=sqrt(sum((object_to_classify-data1(k,2:end)).^2));
           %display(i);

           %display(distance)

           if distance <nearest_neighbour_distance

               nearest_neighbour_distance=distance;

               nearest_neighbour_location=k;

               nearest_neighbour_label=data1(k,1);


               %display(nearest_neighbour_label);

           end

       end

   end

   if label_object_to_classify == nearest_neighbour_label

       number_correctly_classified=number_correctly_classified+1;

   end

end

accuracy=number_correctly_classified/size(data,1);

%display(accuracy);


end
```