Assignment 1 (SMAI)

Viral Parekh
Roll No: 201507535

Details of Datasets

- **Dataset 1**

Name	Iris Data Set
Number of Features	4
Number of Class	3
Number of Insances	150
.Data Source	http://archive.ics.uci.edu/ml/datasets/Iris
Attribute Information	1. sepal length in cm 2. sepal width in cm 3. petal length in cm 4. petal width in cm
Modification in Database	Converted class name to value 1,2,3 for generalization

- **Dataset 2**

Name	Breast Cancer Wisconsin (Original) Data Set
Number of Features	9
Number of Class	3
Number of Insances	682
Data Source	http://archive.ics.uci.edu/ml/datasets /Breast+Cancer+Wisconsin+%28Original%29
Attribute Information	1. Clump Thickness: 1 - 10 2. Uniformity of Cell Size: 1 - 10 3. Uniformity of Cell Shape: 1 - 10 4. Marginal Adhesion: 1 - 10 5. Single Epithelial Cell Size: 1 - 10 6. Bare Nuclei: 1 - 10 7. Bland Chromatin: 1 - 10 8. Normal Nucleoli: 1 - 10 9. Mitoses: 1 - 10
Modification in Database	Removed first column of data set which was sample code ID and Removed data with missing values

- **Dataset 3**

Name	Balance Scale Data Set
Number of Features	4
Number of Class	2
Number of Insances	576
Data Source	http://archive.ics.uci.edu/ml/datasets/Balance+Scale
Attribute Information	1. Class Name: 3 (L, R) 2. Left-Weight: 5 (1, 2, 3, 4, 5) 3. Left-Distance: 5 (1, 2, 3, 4, 5) 4. Right-Weight: 5 (1, 2, 3, 4, 5) 5. Right-Distance: 5 (1, 2, 3, 4, 5)
Modification in Database	Removed class B – which had very few instances

- **Dataset 4**

Name	titanic dataset
Number of Features	4
Number of Class	2
Number of Insances	2201
Data Source	http://www.cs.toronto.edu/~delve/data/titanic/desc.html
Attribute Information	1.CLASS 2.AGE 3.SEX 4.SURVIVED
Modification in Database	N/A

Matlab Code

1. KNN_Classifier_Plot.m
2. KNNClassify.m
3. kFold.m

KNN_Classifier_Plot.m

```
% KNN Classifier Implementation
% Viral Parekh
% Roll No :201507535
% Assignment-1 (SM in AI)

clear;
```

```matlab
clc;

% Path to DataSet
Path2DataSet1='D:\MS CSE\Assignments\SMAI\Assignment1\ds1\IrisData.txt';
Path2DataSet2='D:\MS CSE\Assignments\SMAI\Assignment1\ds3_BreastCancer\BreastCancer.txt';
Path2DataSet3='D:\MS CSE\Assignments\SMAI\Assignment1\ds2_BalanceScale\BalanceScale1.txt';
Path2DataSet4='D:\MS CSE\Assignments\SMAI\Assignment1\ds4_titinic\Titanic.txt';

inp=input('Enter value to select dataset (1/2/3/4) :');
% Read Dataset as per user input
if(inp==1)
 Data= textread(Path2DataSet1) ;
elseif(inp==2)
    Data= textread(Path2DataSet2) ;
elseif(inp==3)
    Data= textread(Path2DataSet3) ;
else
    Data= textread(Path2DataSet4) ;
end

%  Initialize variable for statistical analysis
 Accuracy=zeros(5,5);
 MeanAccuracy=zeros(5,5);
 STD=zeros(5,5);
 cnt=zeros(5,5);

 for nFold=2:5
    Accuracy=zeros(nFold,5);
    for pNo=1:nFold
% Call function kFold to divide dataset in to TraningData and
% SampleData partition
        [TrainData TrueClass SampleData GroundTruth]=kFold(Data,nFold,pNo);
        for kval=1:5
% Run KNNClassify function to get prediction for sample data
           pClass=KNNClassify(TrainData,TrueClass,SampleData,kval);

% Compute Misclassified samples
           Misclassified=0;
           for i=1:size(pClass)
                if(pClass(i)~=GroundTruth(i))
                  Misclassified=Misclassified+1;
                end
           end
%  Compute Accuracy and feed the value into array for post calculation
            ptError=(Misclassified*100)/size(pClass,1);
            ptAccuracy=100-ptError;

            Accuracy(pNo,kval)=ptAccuracy;
        end
    end

    for cc=1:5
    STD(nFold,cc)=std(Accuracy(:,cc));
    MeanAccuracy(nFold,cc)=mean(Accuracy(:,cc));
    end
 end



% Plot the accuracy for 2,3,4,5 fold scenerio
x=1:1:5;
subplot(2,2,1),errorbar(x,MeanAccuracy(2,:),STD(2,:),'rx'),title(strcat('2 Fold
#MeanAccuracy=',num2str(mean(MeanAccuracy(2,:))))),ylabel('Mean Accuracy'),xlabel('K ->'),axis([0
6 50 110 ])
subplot(2,2,2),errorbar(x,MeanAccuracy(3,:),STD(3,:),'rx'),title(strcat('3 Fold
#MeanAccuracy=',num2str(mean(MeanAccuracy(3,:))))),ylabel('Mean Accuracy'),xlabel('K ->'),axis([0
6 50 110 ])
subplot(2,2,3),errorbar(x,MeanAccuracy(4,:),STD(4,:),'rx'),title(strcat('4 Fold
#MeanAccuracy=',num2str(mean(MeanAccuracy(4,:))))),ylabel('Mean Accuracy'),xlabel('K ->'),axis([0
6 50 110 ])
subplot(2,2,4),errorbar(x,MeanAccuracy(5,:),STD(5,:),'rx'),title(strcat('5 Fold
#MeanAccuracy=',num2str(mean(MeanAccuracy(5,:))))),ylabel('Mean Accuracy'),xlabel('K ->'),axis([0
6 50 110 ])


disp(strcat('2 Fold   Mean Accuracy=',num2str(mean(MeanAccuracy(2,:)))));
disp(strcat('3 Fold   Mean Accuracy=',num2str(mean(MeanAccuracy(3,:)))));
disp(strcat('4 Fold   Mean Accuracy=',num2str(mean(MeanAccuracy(4,:)))));
disp(strcat('5 Fold   Mean Accuracy=',num2str(mean(MeanAccuracy(5,:)))));
```

KNNClassify.m

```
function [ PredicatedClass ] = KNNClassify( TrainData,TrueValue,SampleData,K)

 PredicatedClass=0;

% loop for each row of sample data/test data
 for i=1:size(SampleData)

% logic for distance function for dataset
    ss = repmat(SampleData(i,:),size(TrainData,1),1);
    ss=TrainData-ss;
    ss=ss.^2;
    Dist=sum(ss,2);
    [sumValue Index]=sort(Dist);
% Determine class for K neighbours
    kNeighbours=TrueValue(Index(1:K));
    [Class Frequency]=mode(kNeighbours);
% Tie breaker logic
    if(Frequency<=K/2 && K>1)
          kNeighbours=TrueValue(Index(1:K+1));
           [Class Frequency]=mode(kNeighbours);
    end

    PredicatedClass=[PredicatedClass;Class];
 end

 PredicatedClass=PredicatedClass(2:size(PredicatedClass));

end
```

kFold.m

```
function [ TrainData TrueValue SampleData GroundTruth ] = kFold( Data,fold,pNo)

[nDataSets nFeatures]=size(Data);

% Detrimine size of partition
sPartition=floor(nDataSets/fold);

sData=zeros(1,nFeatures);
tData=zeros(1,nFeatures);
```

```
% Divide the data in to two part one for traingdata and true value other
% part sampledata and ground truth
for i=1:fold
    if(pNo==i)
        sData=[sData;Data((i-1)*sPartition+1:i*sPartition,:)];
    else
        tData=[tData;Data((i-1)*sPartition+1:i*sPartition,:)];
    end
end
% Adjust remaining data, ex if total datasets are 51 and numberof folds are
% 5 then 1 data set will be adjusted in training data (50+1)
rem=nDataSets-sPartition*fold;
if(rem~=0)
     tData=[tData;Data(nDataSets-rem:nDataSets,:)];
end


% Adjust Retrun value for the function
TrainData=tData(2:size(tData),1:nFeatures-1);
TrueValue=tData(2:size(tData),nFeatures);
SampleData=sData(2:size(sData),1:nFeatures-1);
GroundTruth=sData(2:size(sData),nFeatures);



end
```

Tie Breaker Logic :

When value of K is even and tie occurs then, K+1 neighbors are taken into account.

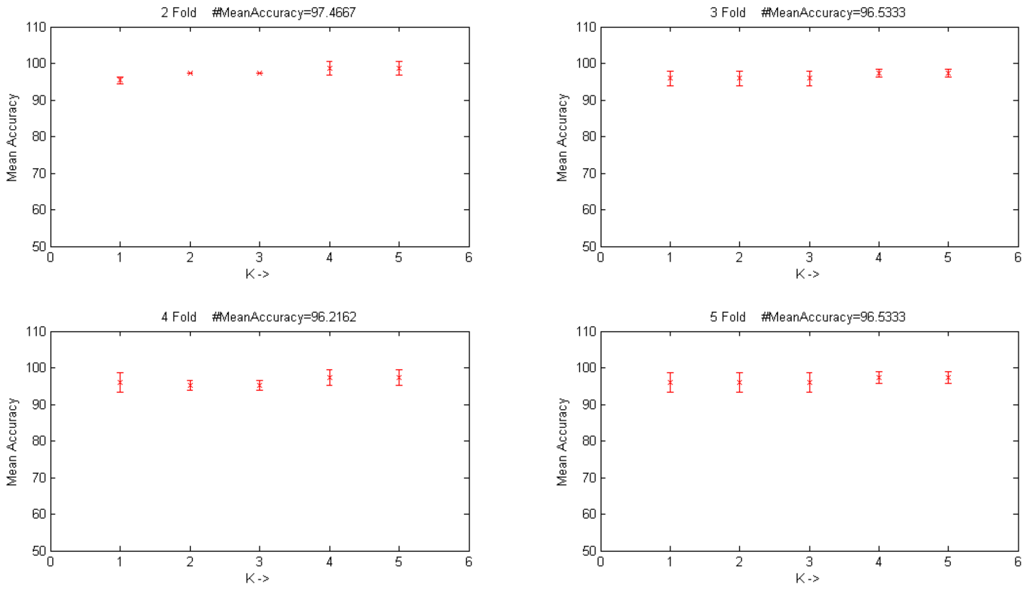
```
    kNeighbours=TrueValue(Index(1:K));
    [Class Frequency]=mode(kNeighbours);
% Tie breaker logic
    if(Frequency<=K/2 && K>1)
           kNeighbours=TrueValue(Index(1:K+1));
           [Class Frequency]=mode(kNeighbours);
    end

    PredicatedClass=[PredicatedClass;Class];
 end
```

Output Graphs

For Dataset 1:

Distance Function : Euclidian Distance

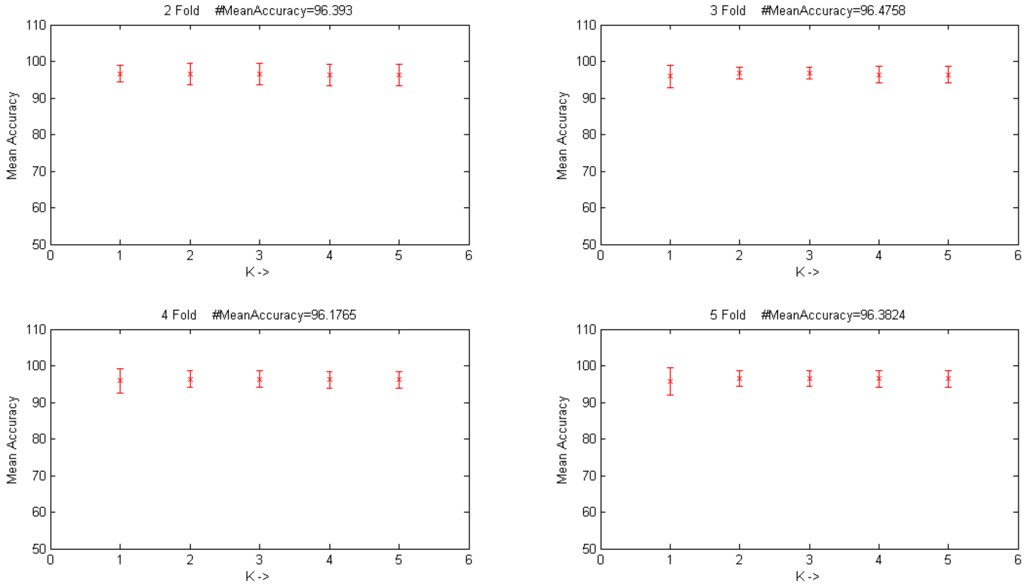
2 Fold Mean Accuracy=97.4667
3 Fold Mean Accuracy=96.5333
4 Fold Mean Accuracy=96.2162
5 Fold Mean Accuracy=96.5333



For Dataset 2:

Distance Function: Euclidian Distance

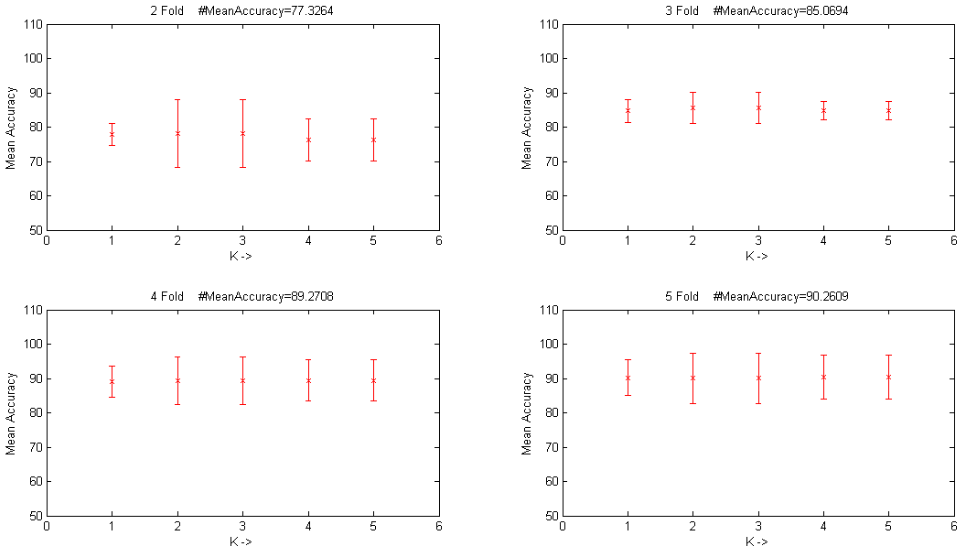
2 Fold	Mean Accuracy=96.393
3 Fold	Mean Accuracy=96.4758
4 Fold	Mean Accuracy=96.1765
5 Fold	Mean Accuracy=96.3824



For Dataset 3:

Distance Function : Euclidian Distance

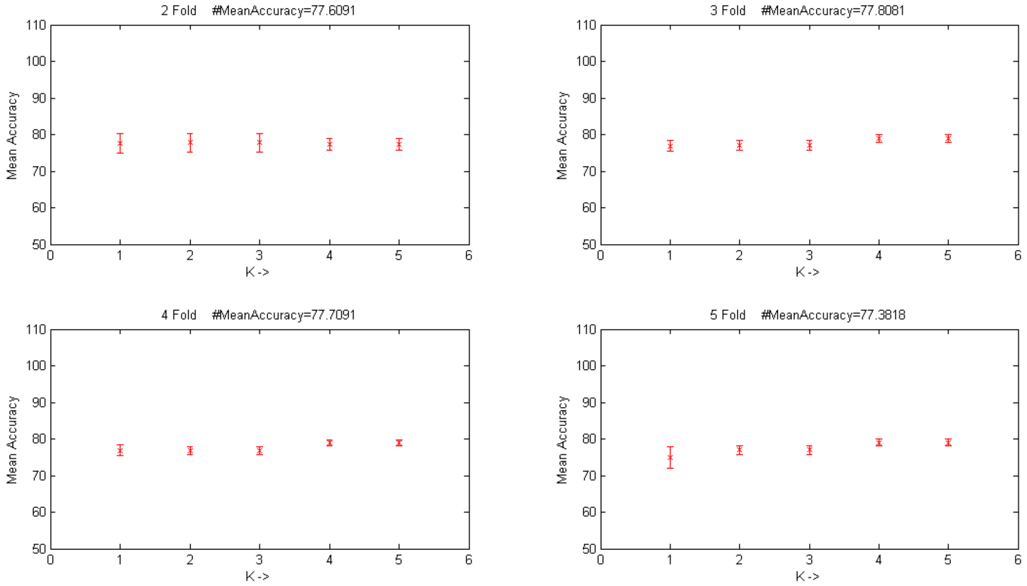
2 Fold   Mean Accuracy=77.3264
3 Fold   Mean Accuracy=85.0694
4 Fold   Mean Accuracy=89.2708
5 Fold   Mean Accuracy=90.2609



For Dataset 4:

Distance Function : Euclidian Distance

2 Fold	Mean Accuracy=77.6091
3 Fold	Mean Accuracy=77.8081
4 Fold	Mean Accuracy=77.7091
5 Fold	Mean Accuracy=77.3818



Summary:

KNN classifier is useful when all the features/attributes which determines the class of dataset are known and the classes are separable with less error in feature space

For using KNN classifier, it is desirable that Training sample is consisting of considerably large number of data for each class. If data for particular class is very less then it may lead to wrong classification for that particular data.