Assignment-1 (SMAI)

Romil Aggarwal, 201330112

Dataset Used

1. Iris Dataset - The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Sir Ronald Fisher (1936) as an example of discriminant analysis. The data set contains 3 classes of 50 instances each (i.e. 150 instances in total), where each class refers to a type of iris plant. It has four attributes :
   - Sepal length in cm
   - Sepal width in cm
   - Petal length in cm
   - Petal width in cm
   - Class:
   -- Iris Setosa (marked as 1)
   -- Iris Versicolour (marked as 2)
   -- Iris Virginica (marked as 3)
   (Source: Already exists in Matlab)

2. Seed Dataset - The examined group comprised kernels belonging to three different varieties of wheat, 70 elements each (i.e. 210 instances in total), randomly selected for the experiment. The data set can be used for the tasks of classification and cluster analysis. To construct the data, seven geometric parameters of wheat kernels were measured:

   - Area A,

   - Perimeter P,

   - Compactness C = 4*pi*A/P^2,

   - Length of kernel,

   - Width of kernel,

   - Asymmetry coefficient

   - Length of kernel groove.

- Class:

-- Kama (Marked as 1)

-- Rosa (Marked as 2)

-- Canadian (Marked as 3)

(Source: Taken from https://archive.ics.uci.edu/ml/datasets/seeds)

3. Wine Dataset-These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituent found in each of the three types of wines. There are total of 178 instances (59 for class One, 71 for class Two and 48 for class Three). The attributes are :

   - Alcohol

   - Malic acid

   - Ash

   - Alcalinity of ash

   - Magnesium

   - Total phenols

   - Flavanoids

   - Nonflavanoid phenols

   - Proanthocyanins

   - Color intensity

   - Hue

   - OD280/OD315 of diluted wines

   - Proline

   The last column (the 14^{th} attribute is class identifier [1-3])

   (Source: Taken from http://archive.ics.uci.edu/ml/datasets/Wine)

4. Breast Cancer - Number of Instances in the dataset are 699 (458 for class One and 241 for class Two). There are 9 class attributes in total. Attribute Information: (last column depicts the class)

   - Clump Thickness

   - Uniformity of Cell Size

   - Uniformity of Cell Shape

   - Marginal Adhesion

   - Single Epithelial Cell Size

   - Bare Nuclei

   - Bland Chromatin

   - Normal Nucleoli

   - Mitoses

   - Class (1 for benign, 2 for malignant)

   (Source: Taken from http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)

Distance Function Used

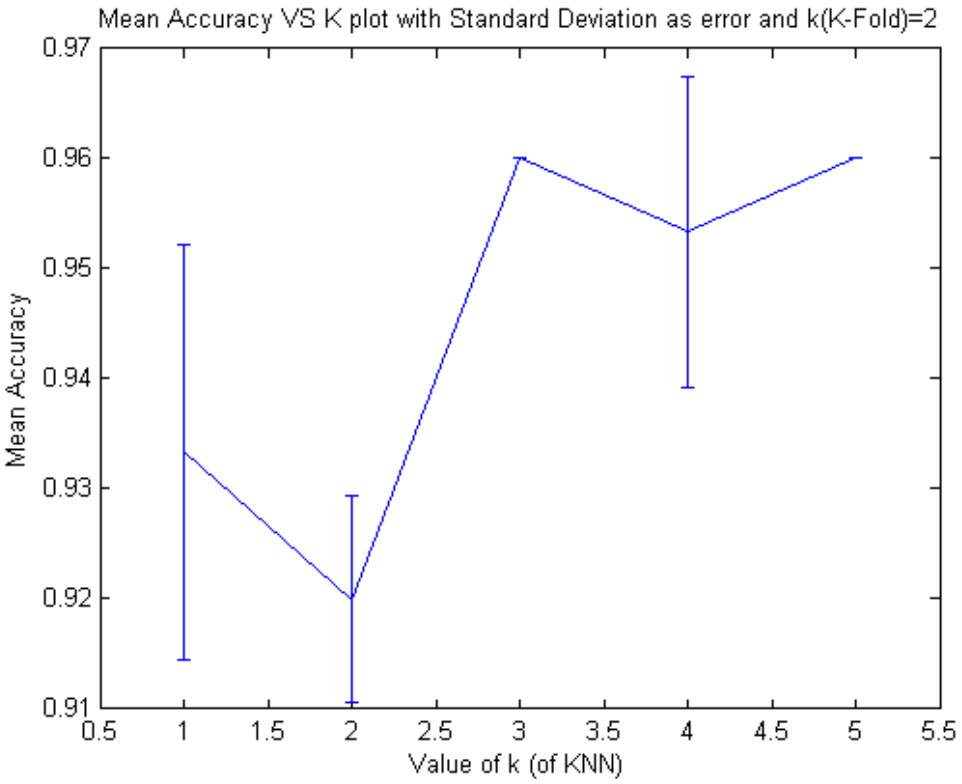
Euclidian Distance - This system gave the best results and hence is being used. Another added benefit of using this model was that it was fairly easy to implement.

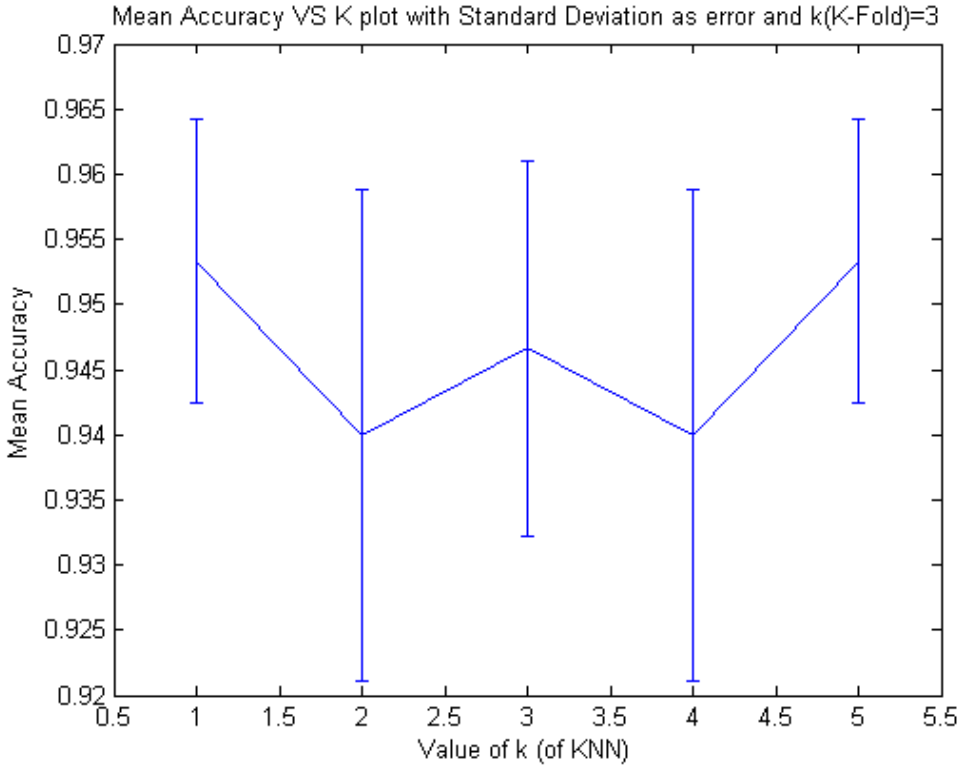
Weighted Distance - This was a good candidate for using as distance function but fixing weights was a difficult task and achieving the right weights was not possible due to lack of information.

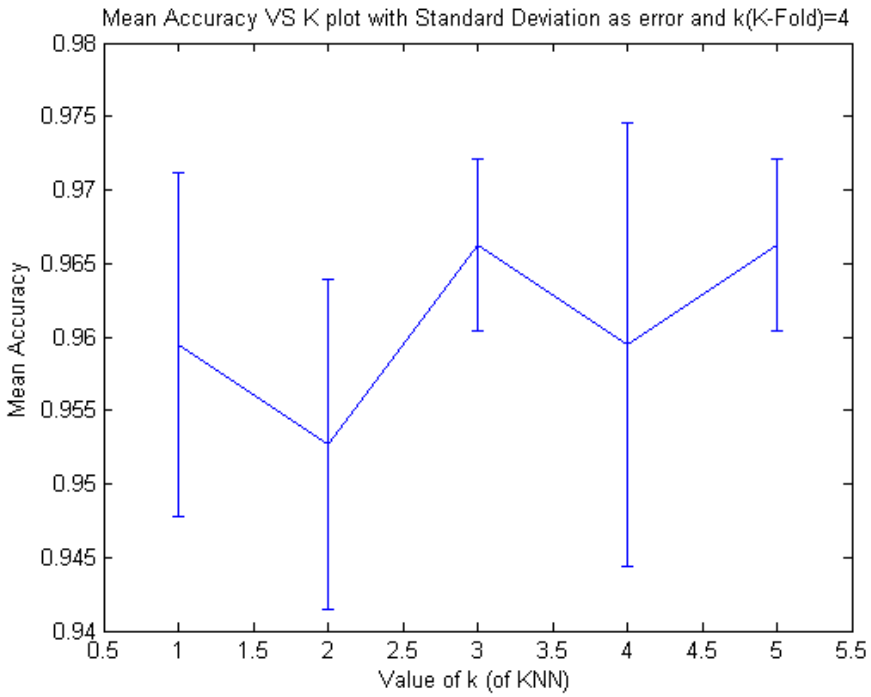
Inverse of Distance as Weighted Distance - This system provided fairly good results but similar (and sometimes better) results were coming from Euclidian distance model so Euclidian system model was preferred over this model.

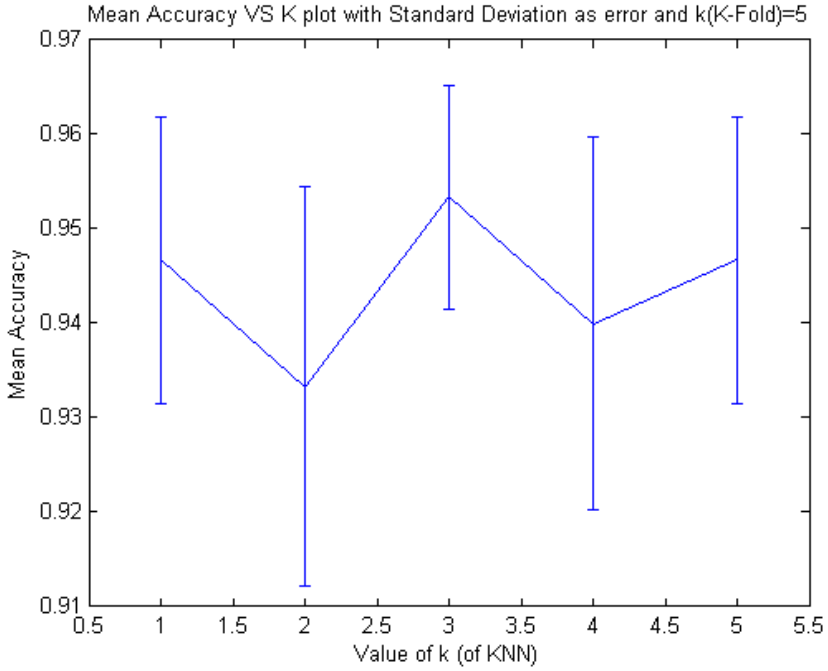
Observations

For Iris Dataset



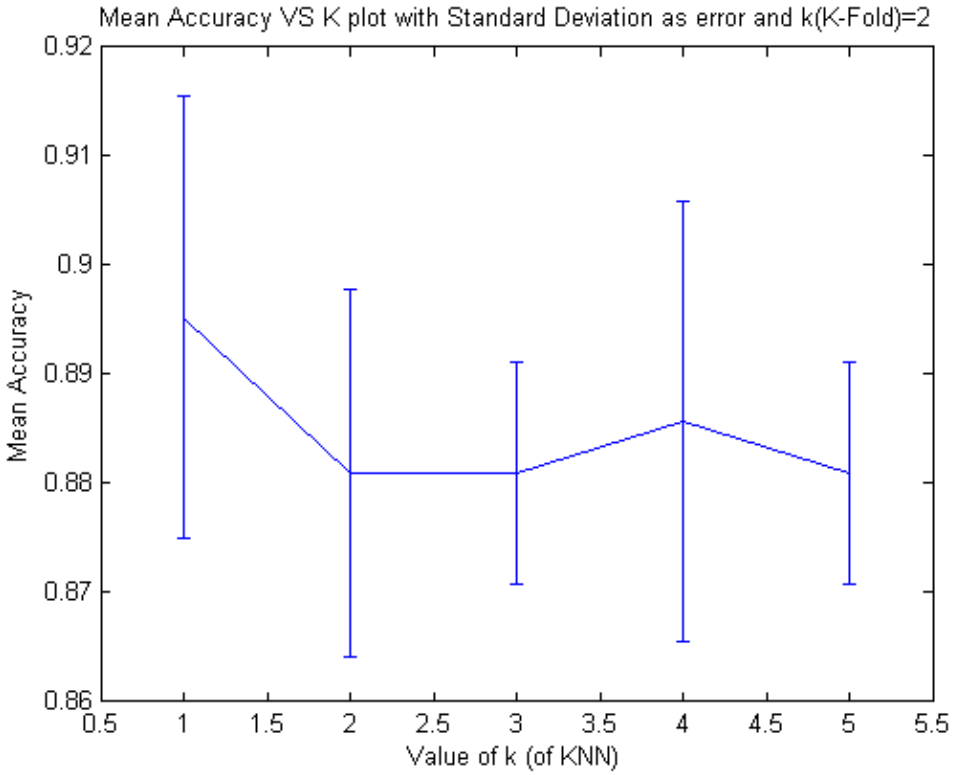


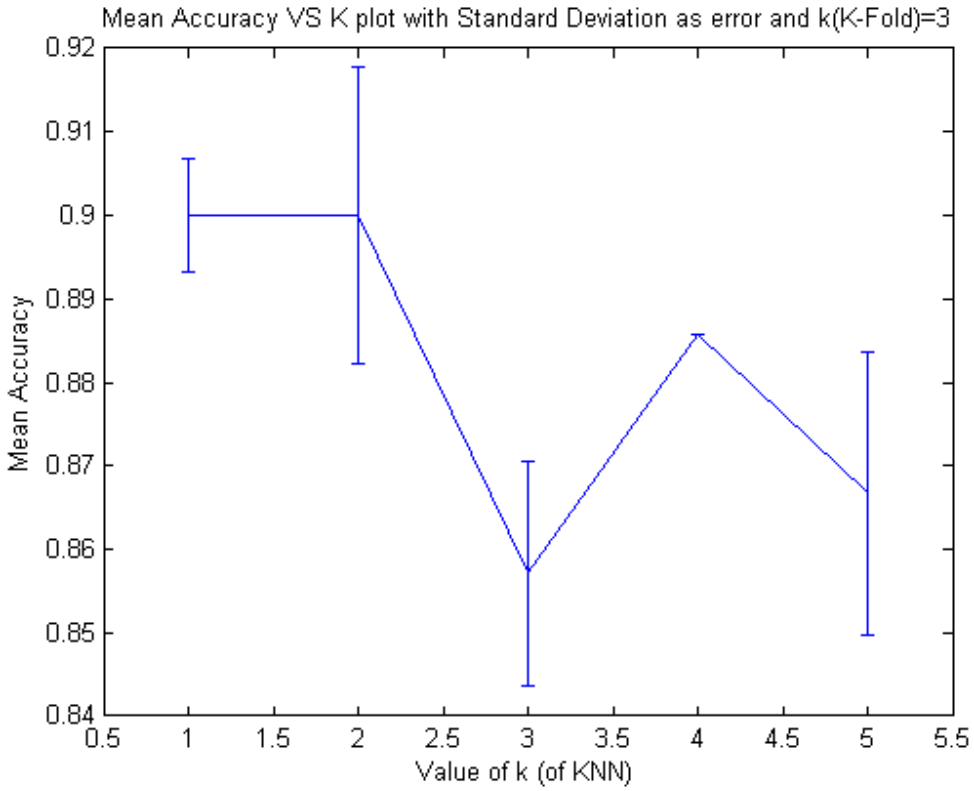


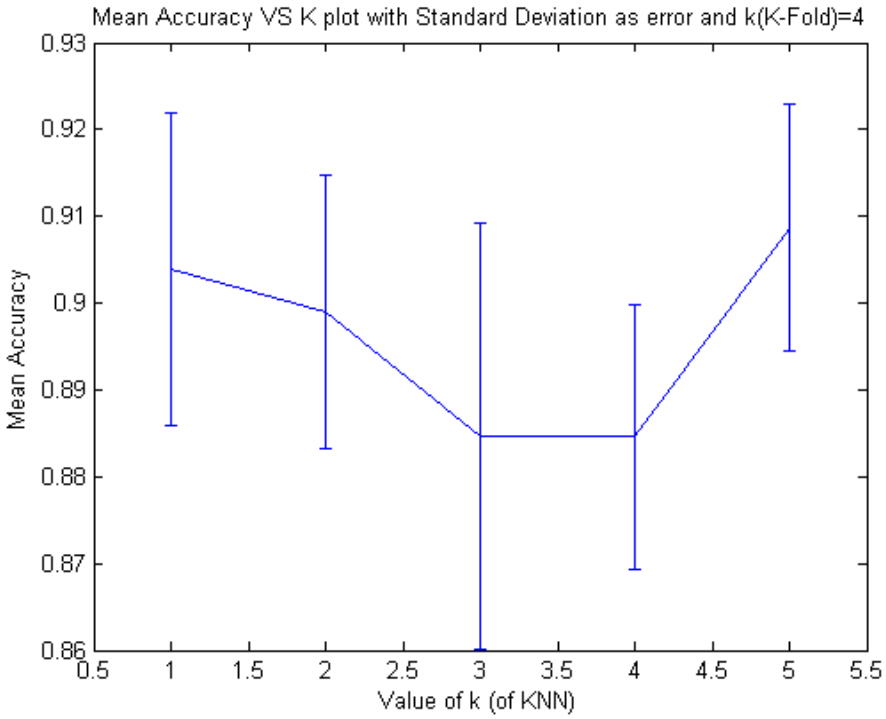


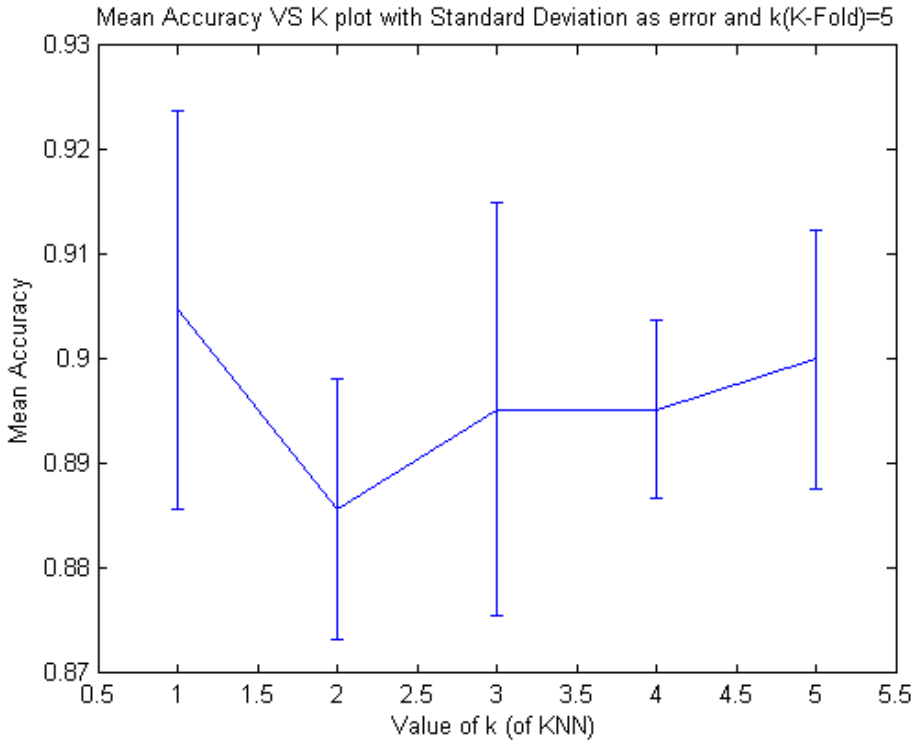
As one can see, high accuracy is attained and most of the points are correctly classified. The best K (of KNN) comes out to be 3 for this dataset. The accuracy first increases with increase in K (of K Fold Cross Validation) but suddenly decreases for K = 5.

For Seeds Dataset



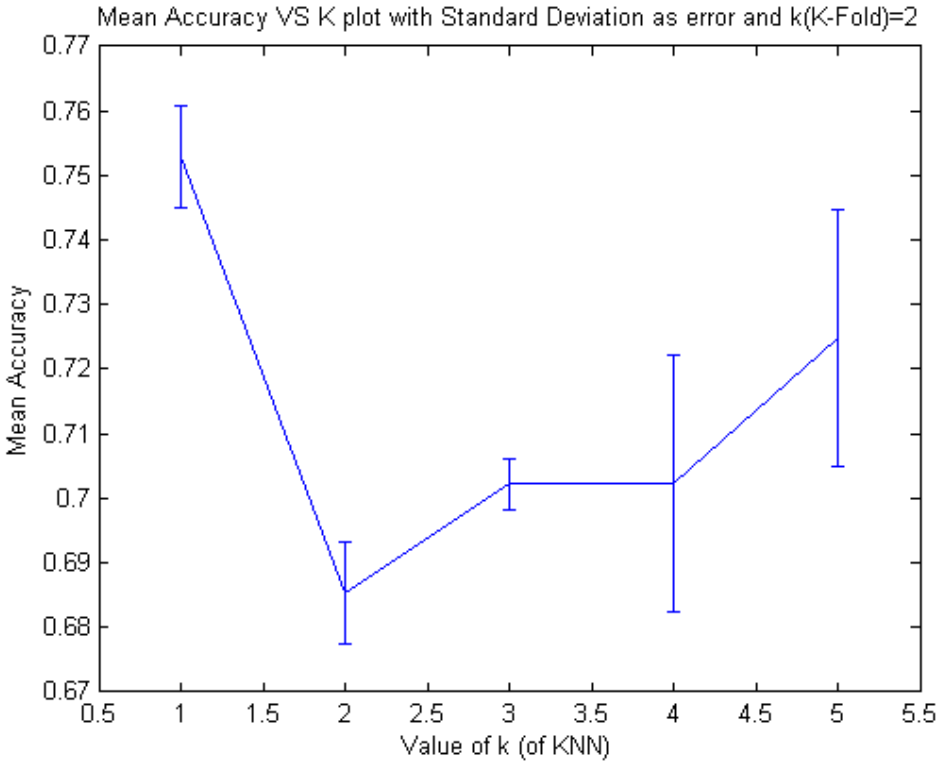


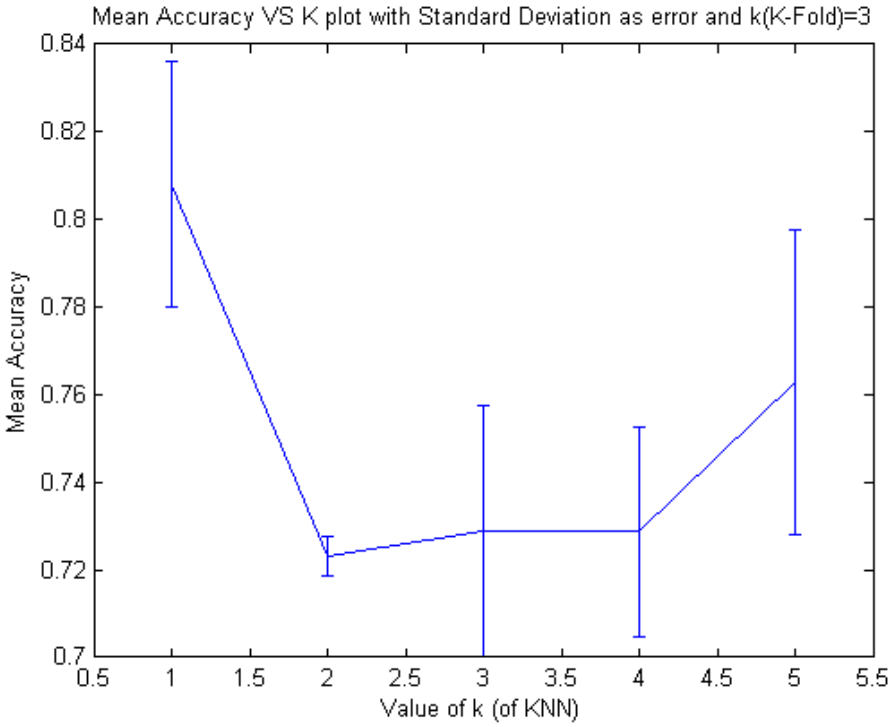


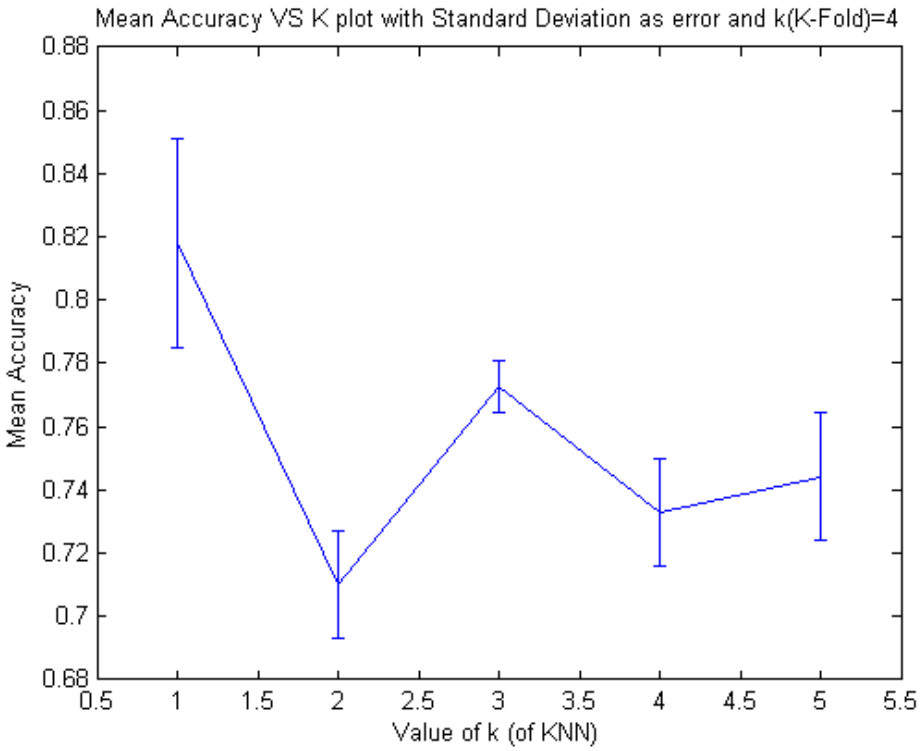


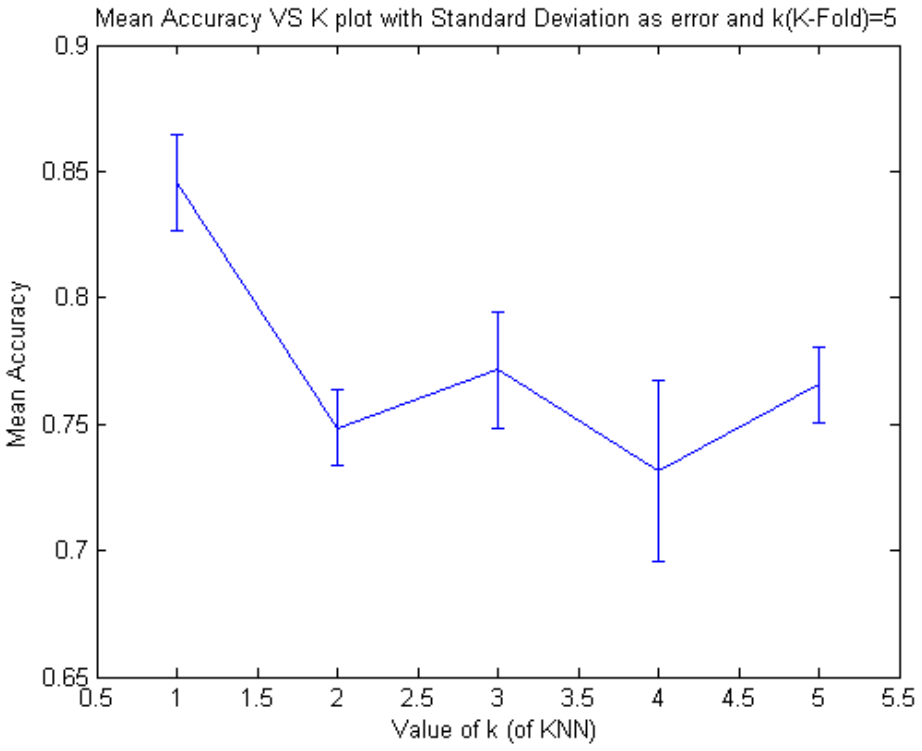
As one can see, high accuracy is attained and most of the points are correctly classified. The best K (of KNN) comes out to be 1 for this dataset. In this dataset accuracy seems to increase with increase in k (of K Fold Cross Validation).

Wine Dataset



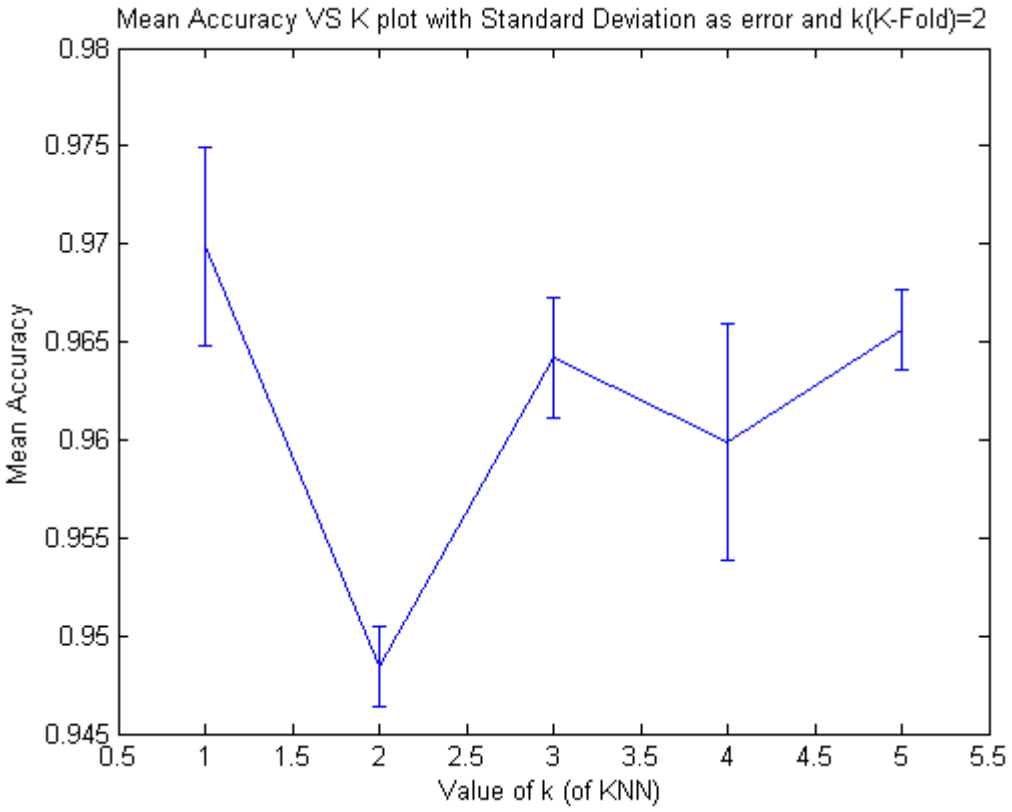


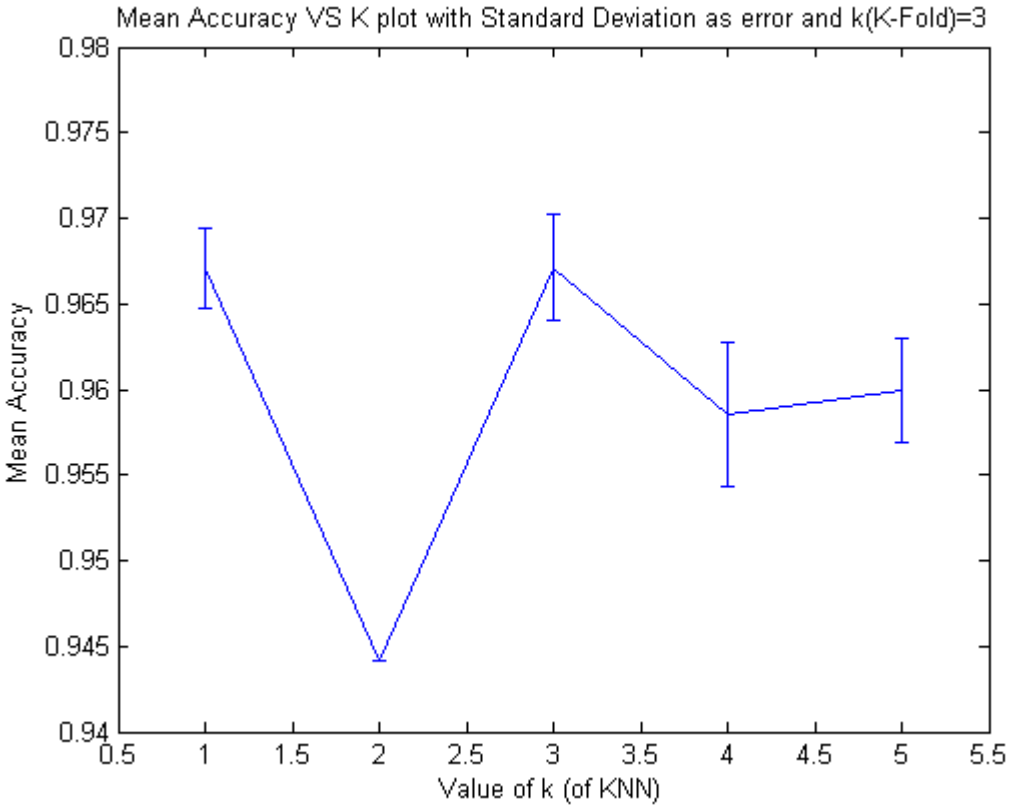


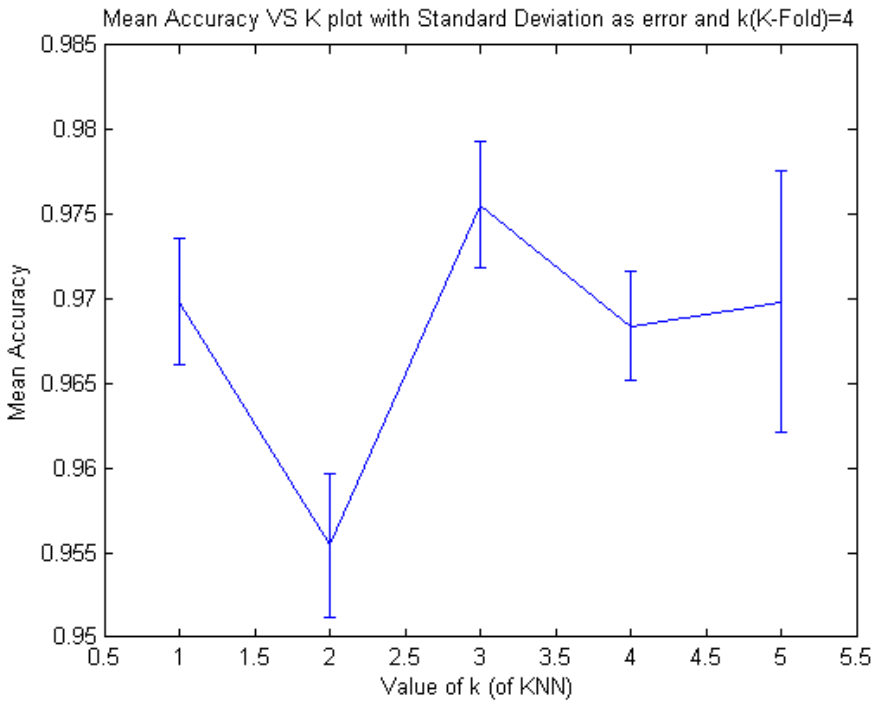


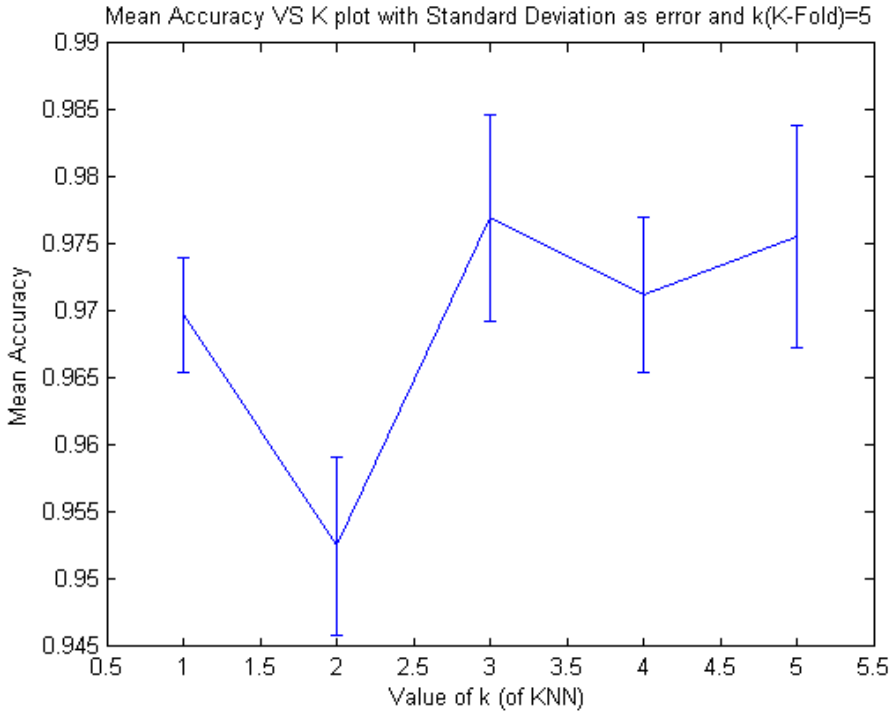
For this dataset, accuracy is not high enough and only some moderate amount of the points are correctly classified. The best K (of KNN) comes out to be 1 for this dataset. Also one can see that as K (of K Cross Validation) increases, the accuracy increases.

Breast Cancer









As one can see, high accuracy is attained and most of the points are correctly classified. The best K (of KNN) comes out to be 3 for this dataset. In this dataset accuracy seems to increase with increase in k (of K Fold Cross Validation).

Code Snippet

KNN Function (Written in Matlab)

```
function [ wrong ] = Knn( train,test,k )
    s1 = size(test);
    s2 = size(train);
    wrong = zeros(s1(1),1);
    for i = 1:s1(1)
        aa = test(i,:);
        a = repmat(aa,s2(1),1);
        b = sqrt((train(:,1:end-1) - a(:,1:end-1)).^2);
        c = sum(b(:,1:end)'); %c is row vector and not a colomn
vector
        temp = [train, c'];
        temp = sortrows(temp,s2(2)+1);
        final = temp(1:k,end-1);
        u = unique(final);
        s3 = size(u);
        max = 0;
        class = 0;
        for j = 1:s3
            d = length(find(final == u(j)));
            if(d>max)
                max = d;
                class = u(j);  %class is the assigned class
            end
        end
        if (class ~= a(1,end)) %a(1,end) has the actual class
            wrong(i) = 1;
        else
            wrong(i) = 2;
        end
    end
end
```

Main Program (Written in Matlab)

```
clc;
clear;
load ./Dataset/datasets.mat
% a = seed;        %last atribute for each entry depicts their actual class
a = irisdataset; %last atribute for each entry depicts their actual class
% a = wine;        %last atribute for each entry depicts their actual class
% a = bcancer;       %last atribute for each entry depicts their actual class
% uncomment the dataset you wish to use
temp = randperm(length(a),length(a)); %no. of samples more than atributes
for p = 2:5
    tot = 0;
    for k = 1:5
        for i = 1:p
            n = length(a)/p;   %let no. of samples be more than atributes
            ti = temp(floor((i-1)*n+1):floor(i*n));
            test = a(ti,:);
            b = ones(size(a));
            b(ti) = 0;
            tri = find(b(:,1) > 0);
            train = a(tri,:);
            l = Knn(train,test,k);
             %wrong is storing the correctly classified;change 2 to 1
             %and then it will have wrong ones
            wrong(p-1,k,i) = double(length(find(l == 2))/floor(n));
            tot = tot+1;
        end
        xyz(k) = sum(wrong(p-1,k,:))/p; %mean of accuracy
        abc(k) = sqrt(sum((wrong(p-1,k,:)-xyz(k)).^2))/p; %standard deviation
    end
    figure,errorbar([1:5],xyz,abc),
    xlabel('Value of k (of KNN)'),    ylabel('Mean Accuracy'),
    title(strcat('Mean Accuracy VS K plot with Standard Deviation as error and
k(K-Fold)=',num2str(p)));
    men1(p-1) = (sum(sum(wrong(p-1,:,:))))/tot;          %mean across each fold
    standard_dev(p-1) = sqrt((sum(sum((wrong(p-1,:,:)-men1(p-1)).^2)))/tot);
    %standard deviation across each fold
end
```