ITEC 3040 Assignment 2

Iason Kotakis 214617732

I have read and understood the Academic Honesty Statement specified in the course outline, and I have adhered fully at all time to the academic honesty rules and policies laid by the instructor, the School of Information Technology and York University Senate's Academic Integrity Policy.

Question 1 solution

```
clc;clear all;close all;
% Question 1
% Implement a KNN classiffier
[~,data]=xlsread('dataset.xlsx');
data=data(2:end,2:end);
Outlook=data(:,1);
Temperature=data(:,2);
Humidity=data(:,3);
wind=data(:,4);
PlayTennis=data(:,5);
k=3;
X=[1:14]';
disp('--------------------------For Outlook--------------------------')
mdl = fitcknn(X,Outlook,'NumNeighbors',k,'Standardize',1)
% Predict the classifications for Outlook with minimum, mean, and maximum characteristics
Xnew = [min(X);mean(X);max(X)];
[label,score,cost] = predict(mdl,Xnew)
disp('--------------------------For Temperature--------------------------')
mdl = fitcknn(X,Temperature,'NumNeighbors',k,'Standardize',1)
% Predict the classifications for Temperature with minimum, mean, and maximum characteristics
Xnew = [min(X);mean(X);max(X)];
[label,score,cost] = predict(mdl,Xnew)
disp('--------------------------For Humidity--------------------------')
mdl = fitcknn(X,Humidity,'NumNeighbors',k,'Standardize',1)
% Predict the classifications for Humidity with minimum, mean, and maximum characteristics
Xnew = [min(X);mean(X);max(X)];
[label,score,cost] = predict(mdl,Xnew)
disp('--------------------------For Wind--------------------------')
mdl = fitcknn(X,wind,'NumNeighbors',k,'Standardize',1)
% Predict the classifications for Wind with minimum, mean, and maximum characteristics
Xnew = [min(X);mean(X);max(X)];
[label,score,cost] = predict(mdl,Xnew)
```

```
--------------------------For Outlook--------------------------

mdl =

  ClassificationKNN
             ResponseName: 'Y'
    CategoricalPredictors: []
               ClassNames: {'Overcast'  'Rainy'  'Sunny'}
           ScoreTransform: 'none'
          NumObservations: 14
                 Distance: 'euclidean'
             NumNeighbors: 3



label =
```

```
  3×1 cell array

    {'Sunny'   }
    {'Overcast'}
    {'Overcast'}


score =

      0.33333            0      0.66667
      0.33333      0.33333      0.33333
      0.66667      0.33333            0


cost =

      0.66667            1      0.33333
      0.66667      0.66667      0.66667
      0.33333      0.66667            1

--------------------------For Temperature---------------------------

mdl =

  ClassificationKNN
             ResponseName: 'Y'
    CategoricalPredictors: []
               ClassNames: {'Cool'  'Hot'  'Mild'}
           ScoreTransform: 'none'
          NumObservations: 14
                 Distance: 'euclidean'
             NumNeighbors: 3



label =

  3×1 cell array

    {'Hot' }
    {'Cool'}
    {'Mild'}


score =

            0            1            0
      0.66667            0      0.33333
            0      0.33333      0.66667


cost =

            1            0            1
```

```
      0.33333            1      0.66667
            1      0.66667      0.33333

--------------------------For Humidity--------------------------

mdl =

  ClassificationKNN
              ResponseName: 'Y'
    CategoricalPredictors: []
               ClassNames: {'High'  'Normal'}
           ScoreTransform: 'none'
          NumObservations: 14
                 Distance: 'euclidean'
             NumNeighbors: 3



label =

  3×1 cell array

    {'High'  }
    {'Normal'}
    {'High'  }


score =

            1            0
      0.33333      0.66667
      0.66667      0.33333


cost =

            0            1
      0.66667      0.33333
      0.33333      0.66667

--------------------------For Wind--------------------------

mdl =

  ClassificationKNN
              ResponseName: 'Y'
    CategoricalPredictors: []
               ClassNames: {'Strong'  'Weak'}
           ScoreTransform: 'none'
          NumObservations: 14
                 Distance: 'euclidean'
             NumNeighbors: 3
```

```
label =

  3×1 cell array

    {'Weak'  }
    {'Strong'}
    {'Strong'}


score =

      0.33333      0.66667
      0.66667      0.33333
      0.66667      0.33333


cost =

      0.66667      0.33333
      0.33333      0.66667
      0.33333      0.66667
```

Question 2 solution

```
clc;clear all;close all;
% Question 2
format short g
x = [56 60 65 73 78 84 89 95 98]';
y = [1773 1699 1660 1651 1642 1634 1628 1603 1583]';
X = [ones(length(x),1) x];
b = X\y;                        % The \ operator performs a least-squares regression.
y2 = X*b;
fprintf('y=%.2f+(%.2f)x\n',b(1),b(2))
[p,err] = polyfit(x,y,1);       % First order polynomial
y_fit = polyval(p,x,err);       % Values on a line
y_dif = y - y_fit;            % y value difference (residuals)
SSdif = sum(y_dif.^2);          % Sum square of difference
SStot = (length(y)-1)*var(y);   % Sum square of y taken from variance
r = -sqrt(1-SSdif/SStot);       % since the correlation is negative, r is a negative number
fprintf('The correlation "r" of the data is %f\n',r)

figure;
scatter(x,y)
hold on
plot(x,y2,'--')
xlabel('Year');
ylabel('Time');
grid on
```

```
y=1912.57+(-3.35)x
The correlation "r" of the data is -0.909744
```

Iason Kotakis