Exercise 3-1, a), b): KNN Regression

Felix Wittich, 28.06.2022

```
clear; close all; clc;
```

1ab) KNN Regression with different k

```
rng(7); % Fix random number generator for reproducible results

% Generate data using the generate_nonlin_data_1D() function. Make
% predictions using your implementation of the kNN algorithm.
k = 10;
N = 50;
noise_level = 1;
[x,y] = generate_nonlin_data_1D(N,noise_level);

xplot = linspace(0,1,500)';
yplot = zeros(500,1);
for i = 1:length(xplot)
    yplot(i) = knn_regression(x,y,xplot(i),k);
end

% Plot results
figure
hold on
plot(x,y,'rx') % Training data
plot(xplot,yplot,'b') % Predicted data
legend('Training Data','Prediction')
xlabel('x')
ylabel('y')

% Implementation of the basic kNN algorithm
function y_pred = knn_regression(x_train,y_train,x0,k)

x = x_train;
y = y_train;

% Calculate pairwise euclidean distance
dist = sum(sqrt((x - x0).^2),2);
% dist = pdist([repmat(x0,size(x,1),1) x],'euclidean');

% Sort the distances in ascending order (use the in-built function 'sort')
[sdist,ind] = sort(dist);

% Find the k closest output values
ynn = y(ind(1:k),:);

% Calculate the mean of the k closest output values
y_pred = (1/k)*(sum(ynn));
```

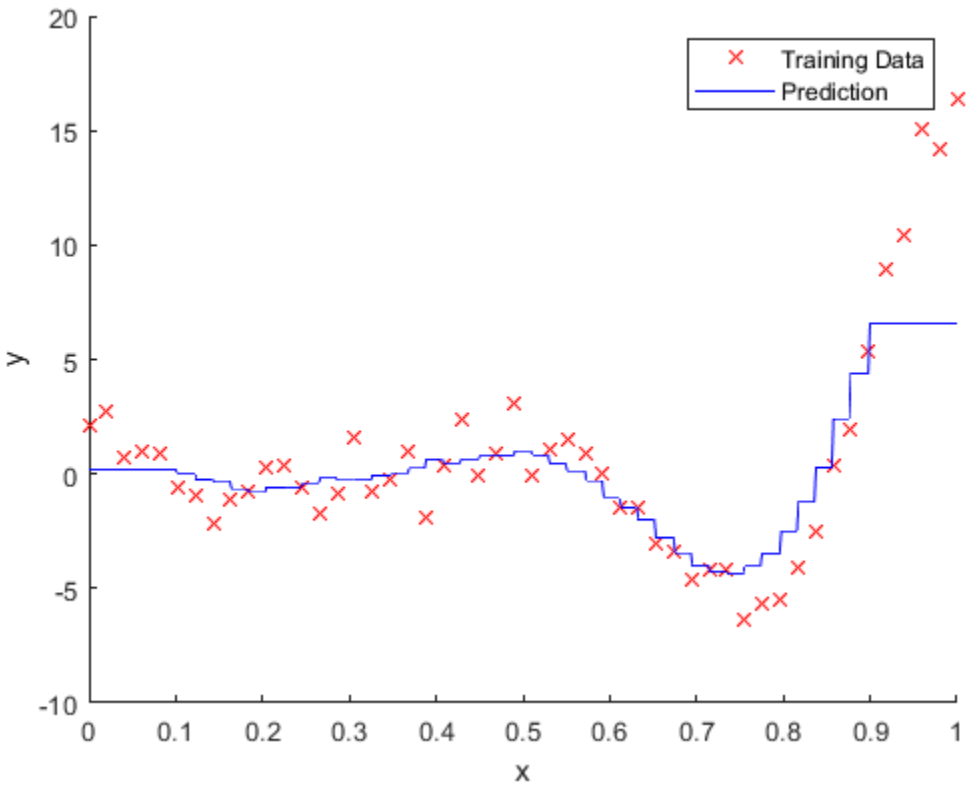
```
end

% Data generating function
function [x,y] = generate_nonlin_data_1D(N,nl)

% 1D Nonlinear test function (http://www.sfu.ca/~ssurjano/forretal08.html)

x = linspace(0,1,N)';
y = ((6*x-2).^2).*sin(12*x-4) + nl*randn(N,1);

end
```



Published with MATLAB® R2021b