Assignment 1

Task 1

This solution takes major inspiration from the example posted on Canvas, but without the Residual part since this is not required in the assignment. My implementation also differs from the example on Canvas since it splits the dataset into a training and a testing dataset (This is done on line 12 and 14). I'm not sure if it is smart, but I also randomly sort the rows of the data before every execution.

As can be seen by the plotted graph in the end of the code. The final regression line ends up not being that accurate in accordance with the data, but since this task was more about exploring the data and doing some basic plotting etc. I don't see it as a major issue.

In the final graph the data used to make the regression is plotted with the blue dots, and the data used for training is plotted as the green dots.

Task 2

Task two is far more complex than task one and the solution I ended up with utilizes mostly external libraries for my solutions. To begin with the dataset in the assignment was available from the sklearn library, I loaded this dataset and split it using the train_test_split function from sklearn. I was uncertain if using a consistent seed with this function was correct or not. I concluded that using a random seed for every running of the code was more "realistic".

Given that the training and testing dataset was randomized at every execution of the code. Picking the appropriate n_neighbours value for the KNeighborsClassifier function was difficult. I therefore made a loop that loops through trying and scoring different values for n_neighbours between 1 and 20. (The process usually lands somewhere between 2 and 7 for the best value). This is displayed as a graph to the user.

When the n_neighbours graph with the best score is decided the Classifier is rerun then the program attempts to predict the y_lables of the entire test dataset. This testing accuracy and results are calculated and reported in the last two lines of the program, in my testing this usually results in a 36(94%) 37(97%) or 38(100%) out of 38 accuracy. Given this, I believe that my solution to the task is fairly adequate and very accurate.