Machine Learning Project 1
Total Points: 100

In this project, you will develop a regression algorithm for learning a predictive model. You will develop a Matlab function with the following I/O and saved as a file called myregression.m:

```
[pred] = function myregression(trainX, testX, noutput)
```

The descriptions of the I/O variables are:

- pred – [ntest x noutput] array of predictions where ntest is the number of input feature vectors in the testing data and noutput is the number of outputs
- trainX – [ntrain x (nfeature + noutput)] training data; array of features and outputs, where the first nfeature columns are the feature values and the last noutput columns are the output values
- testX – [ntest x nfeature] test data
- noutput – the number of output value columns in the training data

For example, if I have two training data vectors, each with four features and two outputs, each row of the trainX array would look like this:

```
trainX(1,:) = [x11 x12 x13 x14 t11 t12];
trainX(2,:) = [x21 x22 x23 x24 t21 t22];
```

where x are your feature values and t are your observations. If I then want to predict the outputs for two test data features, then testX would look like this:

```
testX(1,:) = [x11 x12 x13 x14];
testX(2,:) = [x21 x22 x23 x24];
```

Notice that there are no observations given (as we wouldn't have those for the test data).

Your submission will be your myregression.m file and that is all. I will run your code for three data sets that I will provide (with random cross-validation folds) and one data set that I have sequestered. Your submission will be graded on the following criteria:
(30 points) Does it run?
(50 points) Does it produce a reasonable output?
(20 points) What is the squared error of the prediction? (For these points, I will compare your squared error with my results.)

Here are the rules of the assignment:

- Your code must perform regression as we have learned in class. That is, it must learn a linear regression model $y(x|w) = \Phi w$. (You can use different basis functions, as appropriate.)
- No other inputs are allowed. My script will only provide as input the training data `trainX`, the testing data `testX`, and the number of outputs `noutput`.

- You cannot use the testing data to learn your model (note that my code will not provide the true output for the testing data in the input).
- Your code can perform data scaling (scaling features to be on the interval [0,1] or subtract mean / divide by standard deviation, etc.), data transformation by using basis functions, cross-validation on the training data, or model selection.

I highly recommend that you consider using cross-validation with some sort of model selection and then using the chosen model to predict the output(s) of the testing data.