Xie Zhou 912143385

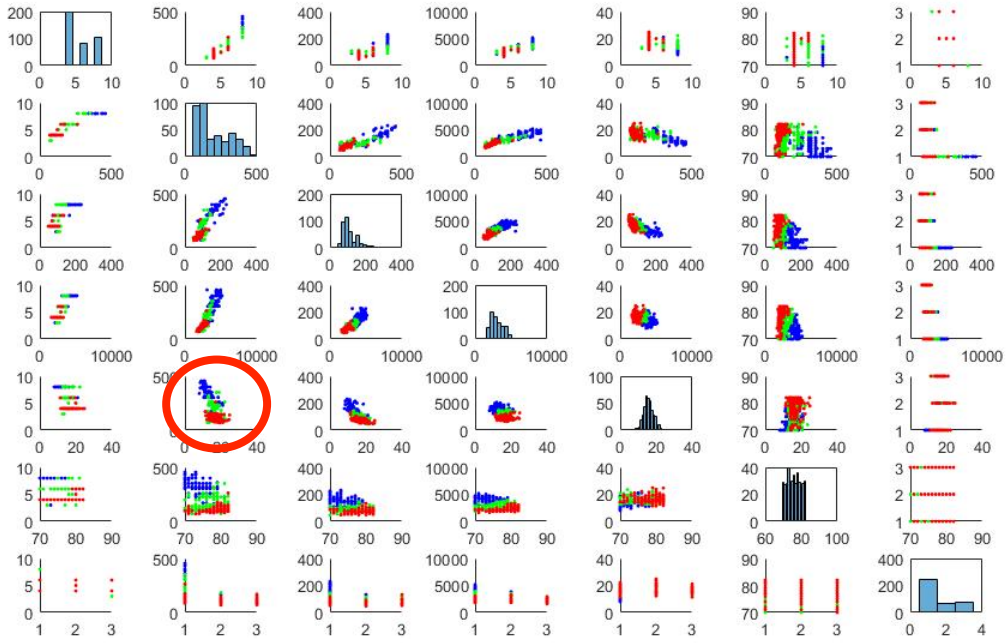
ECS171 HW1

Note: matrix rows are shuffled for randomizing data.
Problem1: files: inputdata.m classify.m; run command: classify

Answers:
1) low: 9-18.6 medium: 18.6-27 high: 27-46.6

Problem2: files: inputdata.m scatterplot.m; run command: scatterplot

Answers:
1) 2d-figure: (displayed in program also)



2) cylinder-weight pair is most informative because the color separations are relatively clear(one above another).

Problem3: files: linear1var.m; run command: none

Answers:
linear1var will be called from trylinear1var.
It takes in a training Matrix [y|x], polynomial degree(0+), and the index of feature (1-7);
It returns a w vector using OLS estimator.

Problem4: files: linear1var.m trylinear1var.m inputdata.m; <u>run command</u>: trylinear1var

Answers:

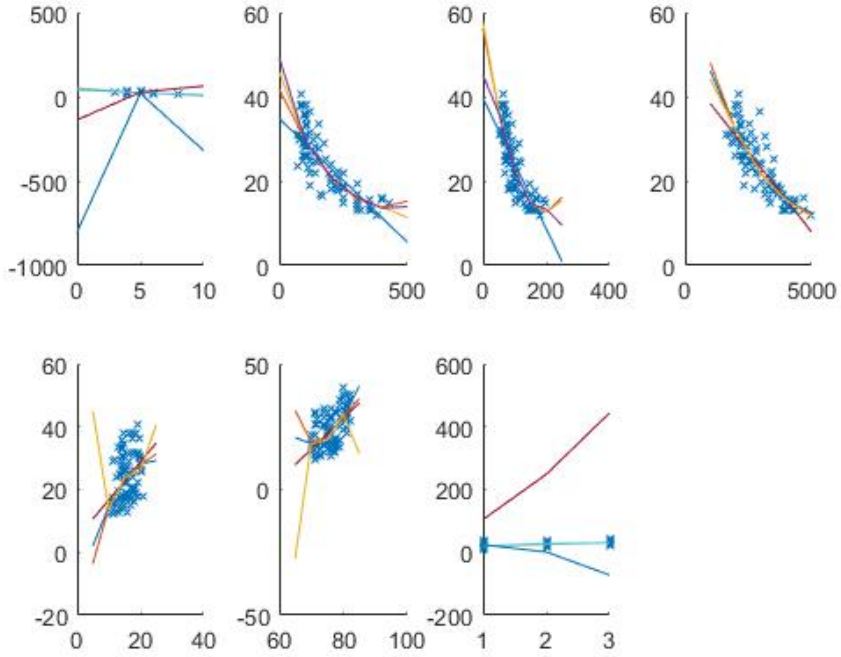
1) training MSE:

	0th	1st	2nd	3rd *(e+03)	4th *(e+03)
x1	0.0593*(e+03)	0.0222*(e+03)	0.0219*(e+03)	0.0208	0.0207
x2	0.0593*(e+03)	0.0194*(e+03)	0.0169*(e+03)	0.0169	0.0169
x3	0.0593*(e+03)	0.0228*(e+03)	0.0192*(e+03)	0.0192	0.0191
x4	0.0593*(e+03)	0.0167*(e+03)	0.0152*(e+03)	0.0152	0.0151
x5	0.0593*(e+03)	0.0497*(e+03)	0.0478*(e+03)	0.0477	0.0460
x6	0.0593*(e+03)	0.0400*(e+03)	0.0376*(e+03)	0.0376	0.0370
x7	0.0593*(e+03)	0.0400*(e+03)	0.0395*(e+03)	1.3189	7.9901

2) testing MSE:

	0th *(e+03)	1st *(e+03)	2nd *(e+03)	3rd *(e+03)	4th *(e+03)
x1	0.0658	0.0302	0.0306	0.0253	0.0273
x2	0.0658	0.0279	0.0255	0.0255	0.0255
x3	0.0658	0.0279	0.0187	0.0186	0.0189
x4	0.0658	0.0254	0.0247	0.0247	0.0247
x5	0.0658	0.0505	0.0541	0.0540	0.0553
x6	0.0658	0.0415	0.0416	0.0418	0.0420
x7	0.0658	0.0459	0.0444	1.1839	7.0909

3) Plot:(displayed in program also)



4) 2nd order performs the best as the corresponding MSE are smaller.
5) 3rd feature –displacement (least error; reliable correspondence) is the most informative.

Problem5: files: modifylinear.m trymodifylinear.m inputdata.m; run command: trymodifylinear

Answers:
1) training set MSE: (polynomial degree 0 - 2)
63.1468 10.3224 6.8007

2) testing set MSE: (polynomial degree 0 - 2)
52.8173 13.4456 9.5945

Problem6: files: logistic.m trylogistic.m inputdata.m; run command: trylogistic

Answers:
logistic.m takes in two parameters, the training matrix[y|x] and the class(0/1/2 ->low/medium/high). It uses batch gradient descent method to update w vector. The learning rate is set to 1*10^-6 to guarantee that w vector would converge, and not too slow.
In trylogistic.m, it calls the logistic function 3 times with class = low, medium, high and the data set. So w0, w1, and w2 returned. Then for each sample in training/testing set, I called the 3 logistic functions together, and compare the results (probabilities of sample i being in class low/medium/high). Set the class whose logistic function returns a higher value as the predicted value (all in a vector). correct # of predictions/total # of samples = accuracy.

accuracyTrain:	accuracyTest:
0.7219	0.7444

The accuracy varies largely. Therefore, I believe there are better methods to use the logistic model for determining multiple(2+) dependent variables.

Problem7: files: predict1.m inputdata.m; run command: predict1

Answers:
1) 2nd order multi-variate polynomial preficted mpg: 21.332083
2) logistic regression predicted category: low mpg

Problem8: files: predictfunny.m inputdata.m; run command: predictfunny

Answers:

1) Assumptions:
   #cylinders = 0
   displacement=350
   horsepower(after certain mechanical enhancement) = 180
   weight(very heavy material) = 4000
   acceleration(slow) = 10
   model year (not too old model, common in some countries): 70
   origin(random) = 1

   2nd order multi-variate polynomial preficted mpg: 8.969108