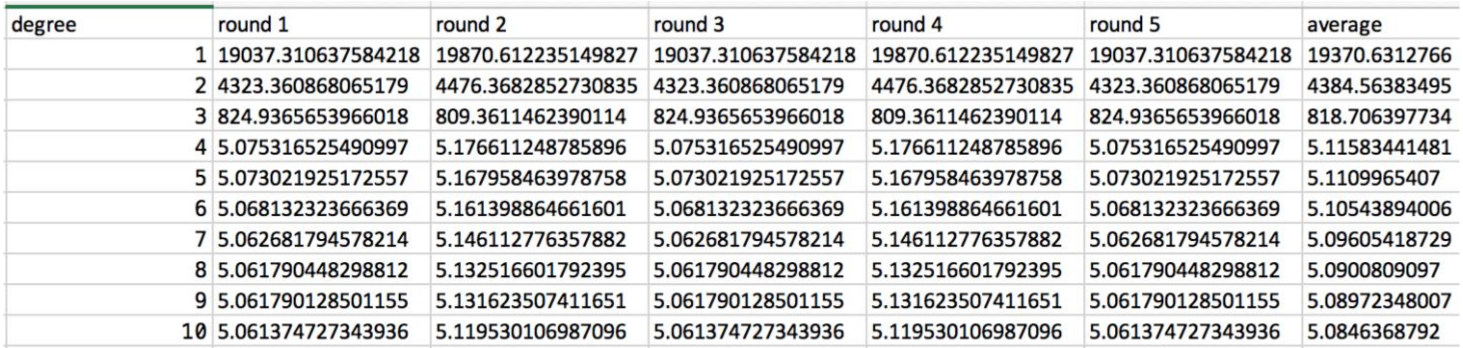
Erin Fago
HW 2- Regression – A Least Squares Approach

Machine Learning Report

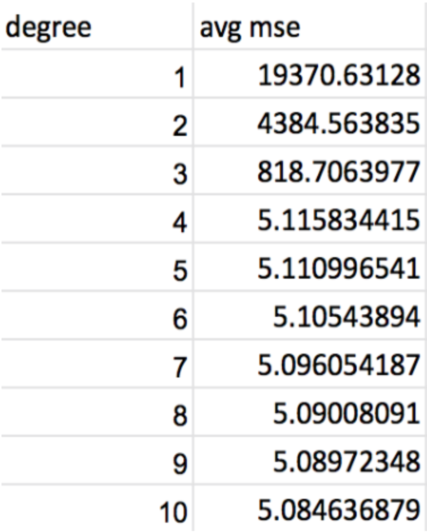
PART 2 - FINDINGS AND NOTES:

Once running the script, nscript.py in part 2 (which calls reg.py from part 1) we can begin to compare the data. Below you will find a chart that shows all the findings from the each round from the 5-fold cross validation for n from 1-10, including the average of all those values at the end.

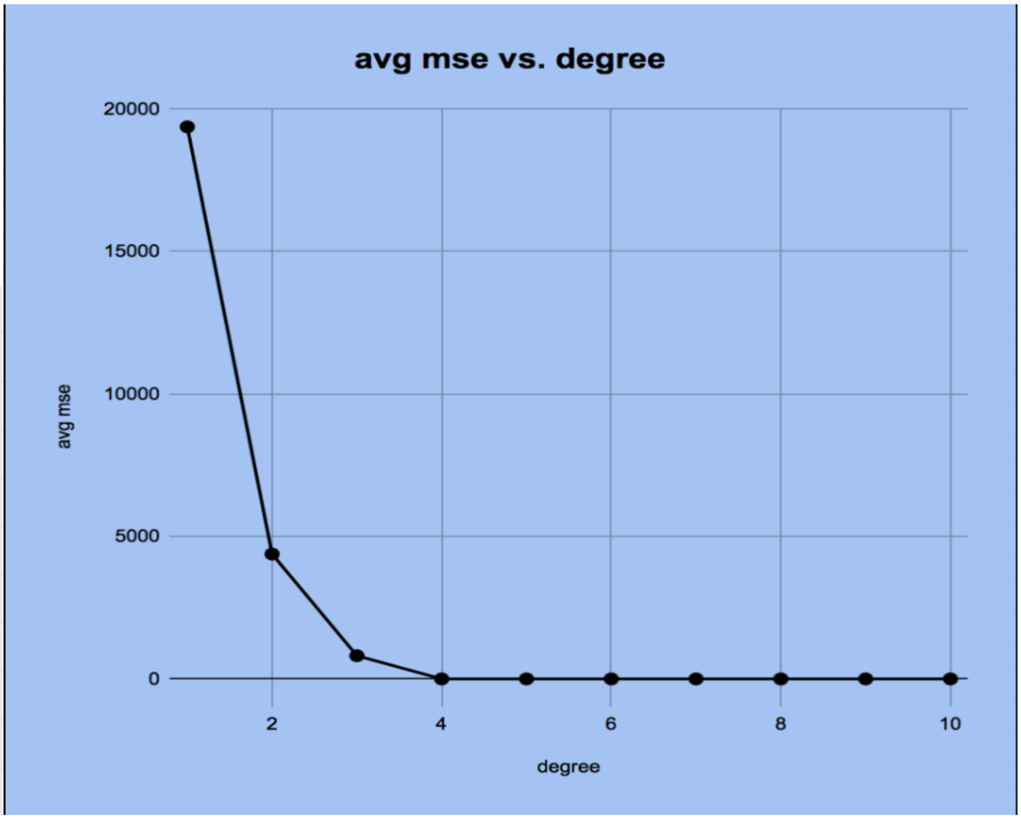


degree	round 1	round 2	round 3	round 4	round 5	average
1	19037.310637584218	19870.612235149827	19037.310637584218	19870.612235149827	19037.310637584218	19370.6312766
2	4323.360868065179	4476.3682852730835	4323.360868065179	4476.3682852730835	4323.360868065179	4384.56383495
3	824.9365653966018	809.3611462390114	824.9365653966018	809.3611462390114	824.9365653966018	818.706397734
4	5.075316525490997	5.176611248785896	5.075316525490997	5.176611248785896	5.075316525490997	5.11583441481
5	5.073021925172557	5.167958463978758	5.073021925172557	5.167958463978758	5.073021925172557	5.1109965407
6	5.068132323666369	5.161398864661601	5.068132323666369	5.161398864661601	5.068132323666369	5.10543894006
7	5.062681794578214	5.146112776357882	5.062681794578214	5.146112776357882	5.062681794578214	5.09605418729
8	5.061790448298812	5.132516601792395	5.061790448298812	5.132516601792395	5.061790448298812	5.0900809097
9	5.061790128501155	5.131623507411651	5.061790128501155	5.131623507411651	5.061790128501155	5.08972348007
10	5.061374727343936	5.119530106987096	5.061374727343936	5.119530106987096	5.061374727343936	5.0846368792

To lay out the data more clearly, below is a chart and a graph from part 2, comparing just the average MSE's from the 5-fold cross-validation for all the degrees from 1-10.



degree	avg mse
1	19370.63128
2	4384.563835
3	818.7063977
4	5.115834415
5	5.110996541
6	5.10543894
7	5.096054187
8	5.09008091
9	5.08972348
10	5.084636879



As demonstrated by the graph, as the degree gets bigger, the average MSE gets smaller. Based on the dataset, the two seem to have a inverse relationship. However, it is important to note that, while the MSE does get smaller as the degree increases, once you get to the 4th degree it does not change much after that. The difference between the 4th degree and the 10th degree is around .03, which is marginal compared to the difference between the 3rd and 4th degree, which is around 813! As the degree increases, a lot more heavy lifting is created for the program and you risk overfitting. If the regression function is overfit then it will not accurately be able to fit future data samples and risks taking erroneous data and outliers too much into effect when determining its curve. Since there is very minimal difference between the degrees from 4 – 10, I believe it makes the most sense to select 4 as our n, even though 10 technically gives you the lowest average MSE across N validation datasets.

PART 3 - FINDINGS AND NOTES:

After running reg.py with syn_train.txt as the training data, synthdata.txt as the validation data, and 4 as the degree we can find the final polynomial function of such order and plot the curve with all samples.

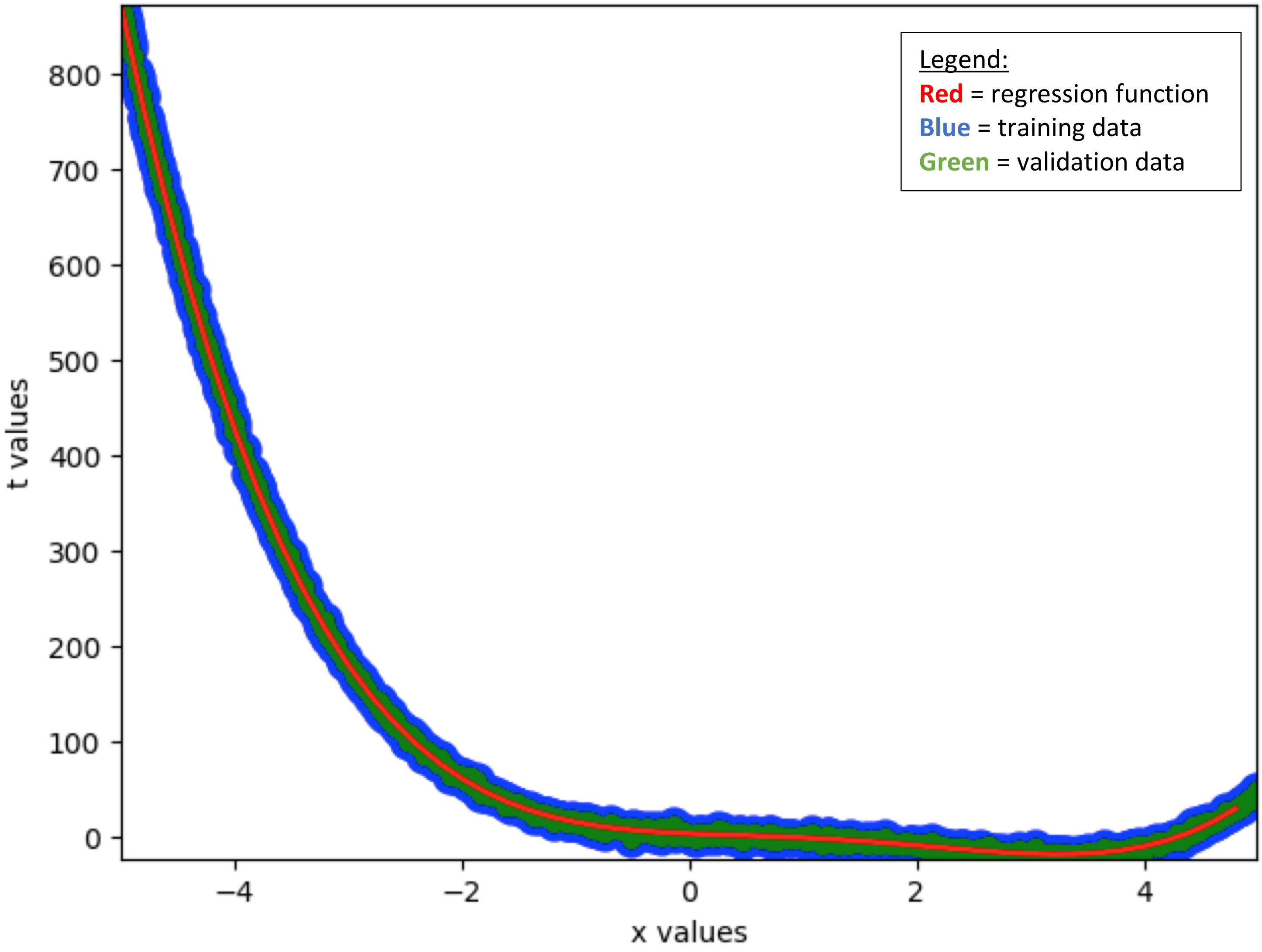
Degree: 4

MSE: 4.969142623955408

Final function: $3.8363638676225515 - 25.376241213289727x + 82.02952516343069x^2 - 387.972869508442x^3 + 373.3638909685668x^4$

Rounded function: $3.84 - 25.38x + 82.03x^2 - 387.97x^3 + 373.36x^4$

Below you will find a graph comparing t vs x values, where the red line is the regression function, the green line is the validation data, and the blueline is the training data for n = 4. The regression function does a great job of learning from the training data while remaining general enough to also be a good fit for the validation data.


Legend:
Red = regression function
Blue = training data
Green = validation data
t values
800
700
600
500
400
300
200
100
0
x values
−4
−2
0
2
4