CS460G MACHINE LEARNING PROJECT 2 – REGRESSIONS WRITEUP

Part 1: Linear Regression with Gradient Descent

Implementation Choices:

- Continuous red wine feature values are normalized by $x_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$ with:
  + x_i is the i-th value of a feature (there are 11 features),
  + $\max(x)$ is the maximum value of that feature,
  + $\min(x)$ is the minimum value of that feature.
  Thus, all normalized feature values are within the range of 0 and 1.
- Then, I add the bias feature column to the beginning of the feature set as x_0 feature with all of it values equal 1.
- The weights/theta values are randomly initialized in the range of 0 and 1.
- The weights are updated using full batch gradient descent method (batch size = number of examples).
- The stopping criteria for the regression model is time, or the number of epochs, which in this part is 3000.

Mean Squared Error:

I ran the code file ***linearRegression.py*** and got these following results:

Dataset {DATA_FILE}	Learning rate {ALPHA}	Iterations {NUM_EPOCHS}	Mean Squared Error	Side-by-side Prediction vs Key Comparison File
winequality-red	0.01	3000	0.5345894595911682	classified_winequality-red.csv

The weights (theta values) of the linear regression model are:

[[2.99012564], [1.02063512], [0.03487777], [0.56491746], [0.14429516], [0.52750981], [0.58006872], [0.2043399], [0.52768081], [1.60225567], [1.30083497], [2.12935034]]

Part 2: Polynomial Regression Using Basis Expansion

Implementation Choices:

- The implementation for this dataset is similar to the winequality-red dataset, except for the data preparation. For these two synthetic datasets, I do not normalize the feature values and I expand the dataset using by basis expansion (adding "new" features by raising the original feature to the k-order). So:
  + For the 2nd-order polynomial, I added 1 new column of feature X values squared.
  + For the 3rd-order polynomial, I added 2 new column of feature X values raised to the power of 2 and 3.

+ For the 3rd-order polynomial, I added 4 new column of feature X values raised to the power of 2, 3, 4, and 5.
Then, I added the bias feature column (all values equal 1) to the beginning of the expanded dataset and proceeded the regression algorithm with full batch gradient descent as the previous part.

- The weights/theta values are randomly initialized in the range of -3.0 and 3.0.
- The stopping criteria for the regression model is time, or the number of epochs, which in this part is 2000.

Mean Squared Error:

I ran the code file ***polynomialRegression.py*** for each of the synthetic data files. For each synthetic dataset**, I changed the initial variables *DATA_FILE* and *ALPHA* at the beginning of the code file before running**.

For each run, the program would call the Regression class 3 times consecutively and **stop between each order model** due to the pop-up graph **then continue creating the next model after the pop-up graph is closed manually**. After changing the parameter values, I got these following results:

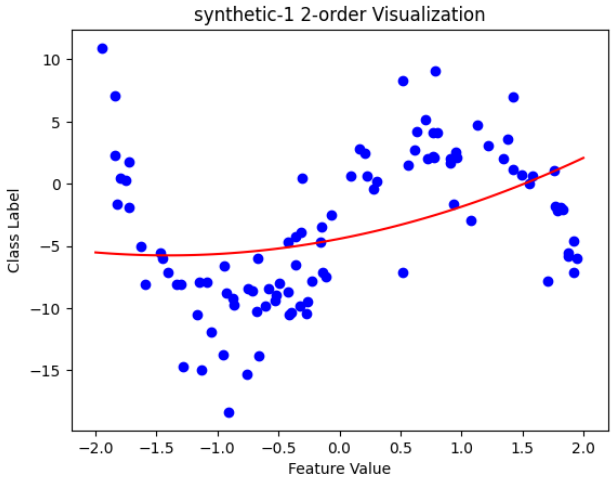
Dataset {DATA_FILE}	Learning rate {ALPHA}	Iterations {NUM_ EPOCHS}	Order	Weights & Mean Squared Error (MSE)	Side-by-side Prediction vs Key Comparison File
synthetic-1	0.01	2000	2	Weights: [[-4.41814572], [1.90014917], [0.67588914]] MSE = 30.712509552923958	classified_synthetic-1_2-order.csv
			3	Weights: [[-3.98071887], [10.95617737], [0.87372379], [-3.76078947]] MSE = 9.025855345207804	classified_synthetic-1_3-order.csv
			5	Weights: [[-3.18691497], [7.83417605], [-1.65791317], [-0.04956681], [0.77800501], [-0.85336794]] MSE = 9.056736109147046	classified_synthetic-1_5-order.csv
synthetic-2	0.007	2000	2	Weights: [[0.36303167], [-0.04931537], [-0.17333306]] MSE = 0.3307816002746618	classified_synthetic-2_2-order.csv
			3	Weights: [[0.33868555], [-0.30064793],	classified_synthetic-2_3-order.csv

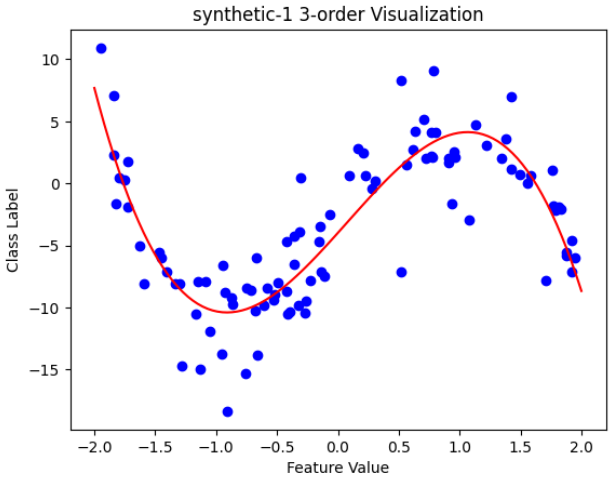
				[-0.16018777], [0.0976336]] MSE = 0.3425315877890235	
			5	Weights: [[0.49694341], [-0.97541396], [-0.54265167], [0.9225592], [0.10948869], [-0.18999385]] MSE = 0.3315644390489069	classified_synthetic-2_5-order.csv

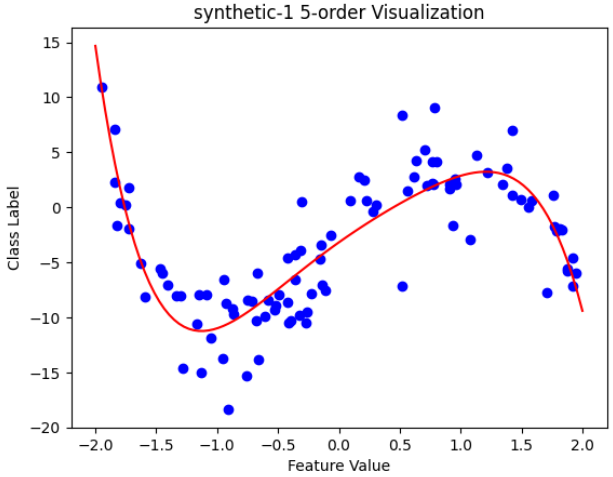
Part 3: Plot Your Regression Lines

Synthetic data visualizations as scatter plots using *matplotlib.pyplot* with the data points in blue and regression lines in red:

1. synthetic-1.csv models of 2, 3, and 5 order respectively







2. synthetic-2.csv models of 2, 3, and 5 order respectively