# Homework 2 Polynomial Regression 460 G

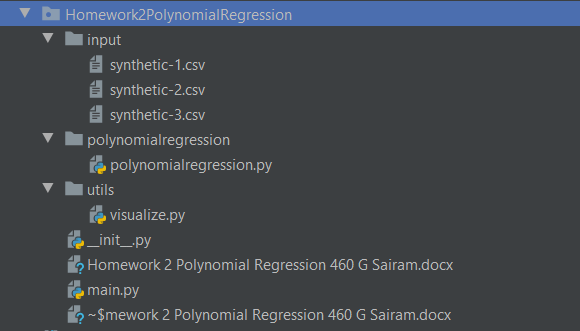
**Sairam Sri Vatsavai**

## Synthetic dataset

**Polynomial Regression** : I realized the need for polynomial regression after plotting the datasets, as none of the dataset can be approximated with linear regression. Hence, polynomial regression is better choice for the datasets. The parameters/hyperparameters play important role in reducing the MSE (mean square error) are learning rate and epochs. I have used full batch gradient descent algorithm to change the weights or theta values during training.

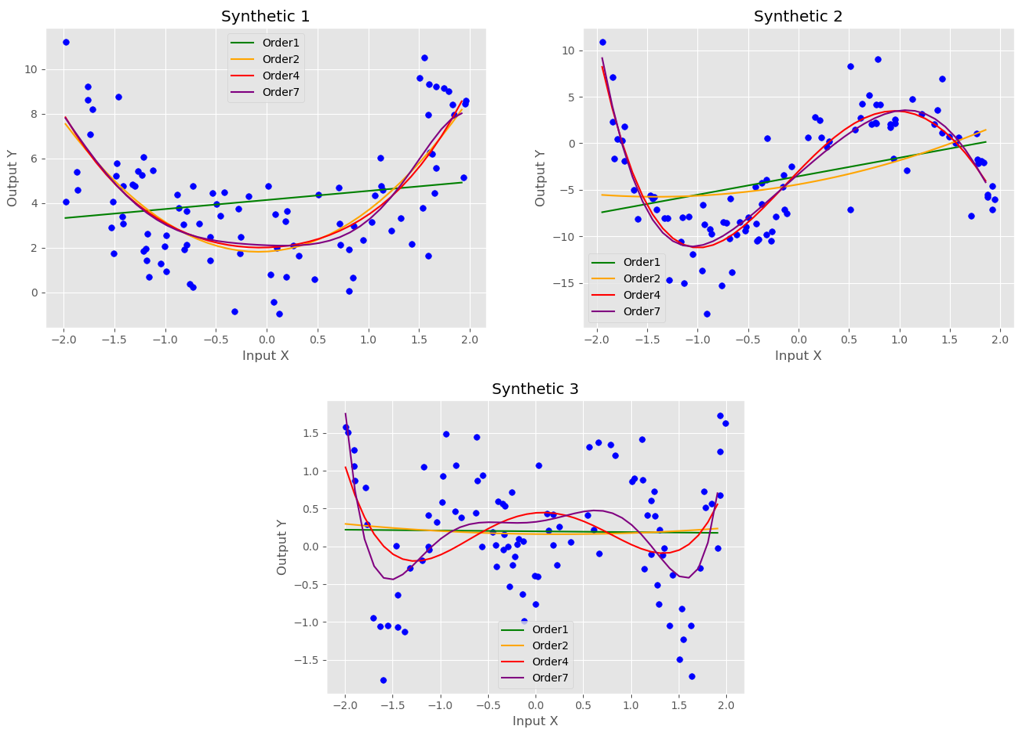
**Implementation Details:** I created a python project shown in Figure 1.

* It contains PolynomialRegression class for implementation including *fit* method for training, *mse* method for MSE calculations function.
* Input folder contains the datasets
* Utils folder contains functions for visualization of polynomial line generated after training
* main.py contains calls for training all the datasets to create polynomial lines and reporting the MSE for each of them for various degrees or order. It also creates plots to illustrate the polynomial lines.



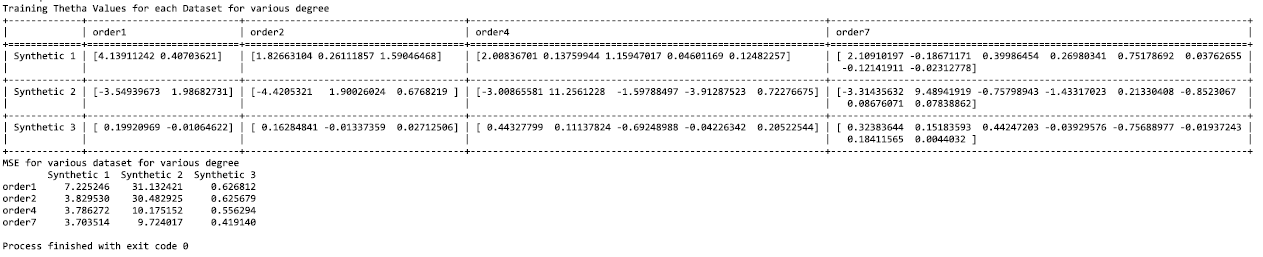
**Figure 1. Project Structure**

**Visualization :**  Figure 2. contains the plots of trained polynomial model for each synthetic datasets. The implementation is adapted from : <https://github.com/pickus91/Polynomial-Regression-From-Scratch>



**Figure 2. Polynomial Regression plots for each synthetic dataset for various order**

**Accuracy Details:** I am reporting the MSE details of each dataset in Table 1 and Figure 3.



**Figure 3. Polynomial Regression MSE and Theta values for each dataset with various order**

**Hyperparameter Configuration:**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| **Alpha/Learning Rate** | **0.001** |
| **Epochs** | **100k** |
| **Order/Degree** | **[1,2,4,7]** |

**Libraries Utilized:**

* **matplotlib**
* **pandas**
* **numpy**
* **tabulate**