<u>ML Lab (CS360)</u>

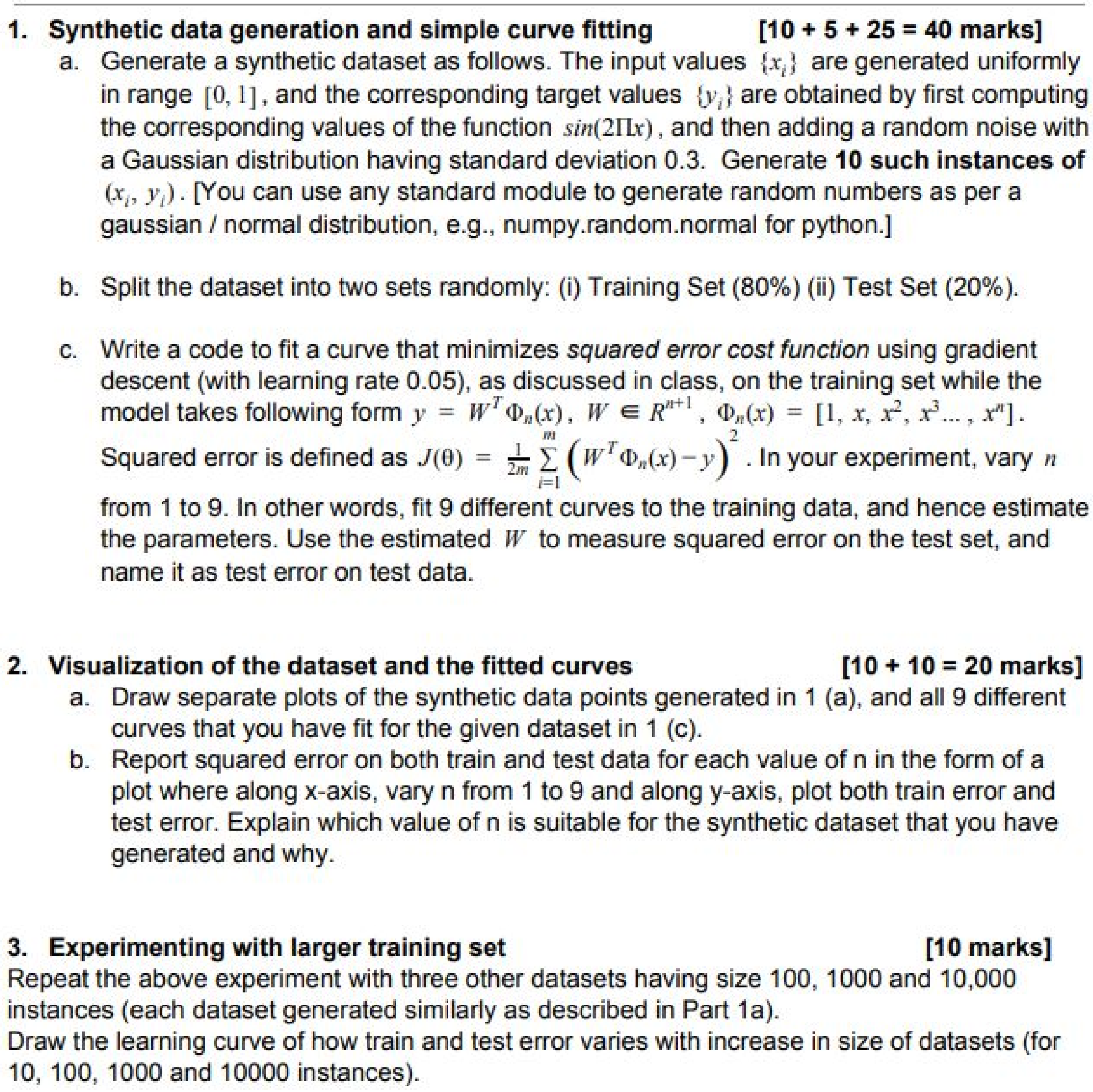
Assignment 4

Total Marks: 80

Submission deadline: 13-Sept-2020 (5PM)

Submission mode: Google Form

<u>Linear Regression:</u> Reference



1. **Synthetic data generation and simple curve fitting** **[10 + 5 + 25 = 40 marks]**
    a. Generate a synthetic dataset as follows. The input values $\{x_i\}$ are generated uniformly in range $[0, 1]$, and the corresponding target values $\{y_i\}$ are obtained by first computing the corresponding values of the function $sin(2\Pi x)$, and then adding a random noise with a Gaussian distribution having standard deviation 0.3. Generate **10 such instances of** (x_i, y_i). [You can use any standard module to generate random numbers as per a gaussian / normal distribution, e.g., numpy.random.normal for python.]

    b. Split the dataset into two sets randomly: (i) Training Set (80%) (ii) Test Set (20%).

    c. Write a code to fit a curve that minimizes *squared error cost function* using gradient descent (with learning rate 0.05), as discussed in class, on the training set while the model takes following form $y = W^T \Phi_n(x)$, $W \in R^{n+1}$, $\Phi_n(x) = [1, x, x^2, x^3 ..., x^n]$. Squared error is defined as $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(W^T \Phi_n(x) - y \right)^2$. In your experiment, vary n from 1 to 9. In other words, fit 9 different curves to the training data, and hence estimate the parameters. Use the estimated W to measure squared error on the test set, and name it as test error on test data.

2. **Visualization of the dataset and the fitted curves** **[10 + 10 = 20 marks]**
    a. Draw separate plots of the synthetic data points generated in 1 (a), and all 9 different curves that you have fit for the given dataset in 1 (c).
    b. Report squared error on both train and test data for each value of n in the form of a plot where along x-axis, vary n from 1 to 9 and along y-axis, plot both train error and test error. Explain which value of n is suitable for the synthetic dataset that you have generated and why.

3. **Experimenting with larger training set** **[10 marks]**

Repeat the above experiment with three other datasets having size 100, 1000 and 10,000 instances (each dataset generated similarly as described in Part 1a).
Draw the learning curve of how train and test error varies with increase in size of datasets (for 10, 100, 1000 and 10000 instances).

Logistic Regression (Marks: 10)

1. Implements the logistic regression over the breast cancer wisconsin dataset (Hint: Load dataset using **“sklearn.datasets.load_breast_cancer”**)
    a. Compute training and testing accuracy on the table mentioned below by varying training samples (from 10% to 60%). Moreover, please save the table as a CSV file.

1. Amount of randomly selected training data	2. Training accuracy	3. Testing accuracy
10%		
20%		
30%		
40%		
50%		
60%		