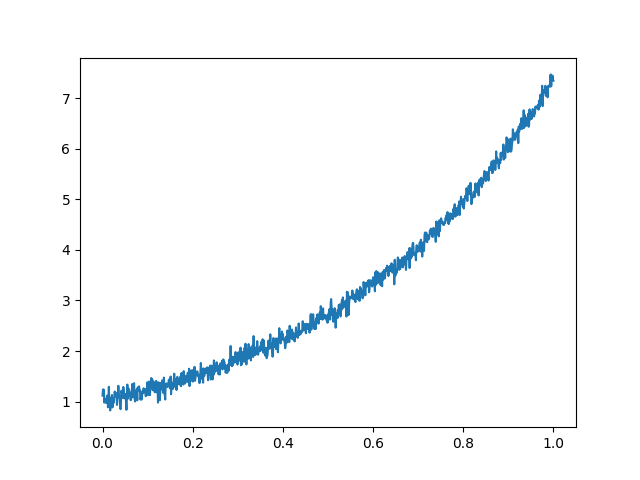
1112 Deep Learning – Homework 1 112753207 張詠軒 資碩計一

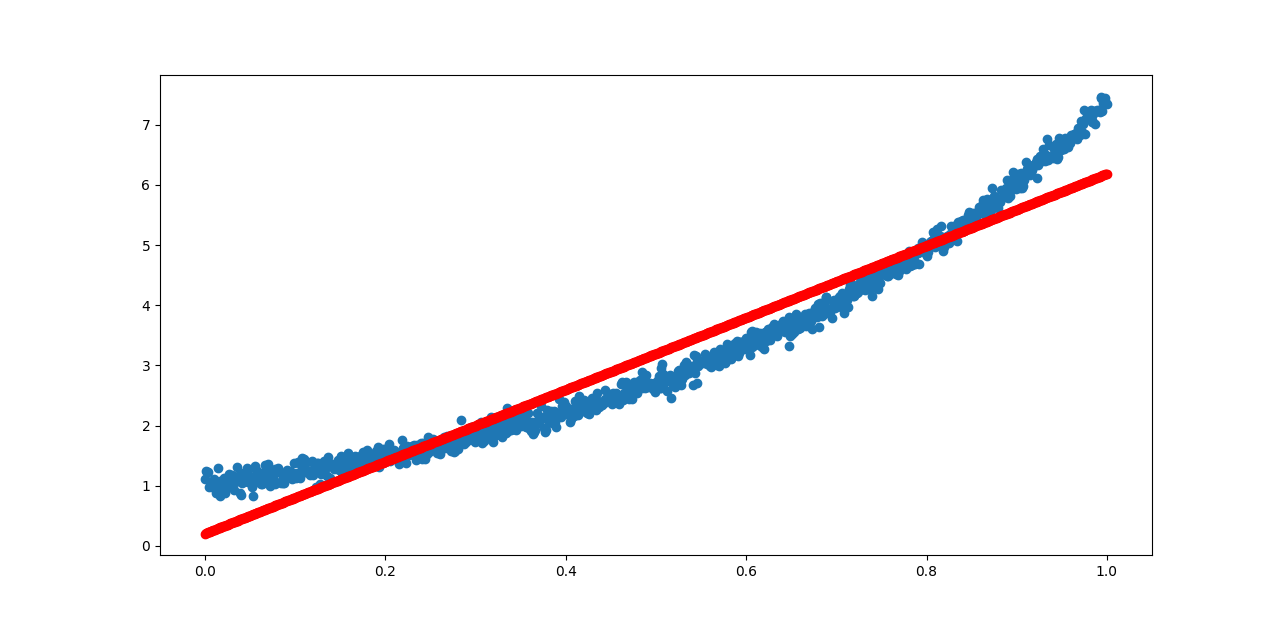
Due: 10/22, 2023, 11:59pm

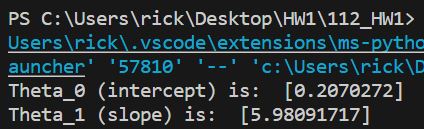
For the following questions, please upload the source code to moodle and explain the results in your report. **If you choose to implement the machine learning models by yourself (no built-in APIs), you will get extra 10% bonus for each question.**

1. Please load ‘data.mat’ into your Python code, where you will find 𝑥, 𝑦 ∈ 𝑅1001. Now do the following procedures.
   1. (**5%**) Plot the data using plot function.



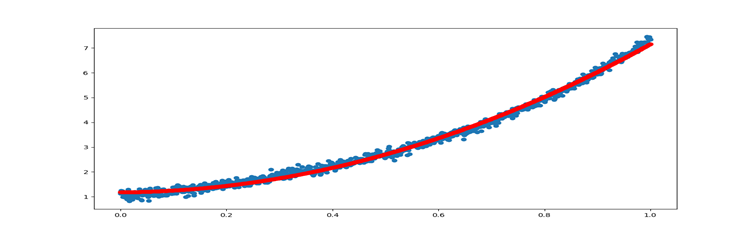
* 1. (**5%**) Compute the least square line 𝑦 = 𝜃𝟎 + 𝑥𝜃𝟏 using the given data and overlay the line over the given data.

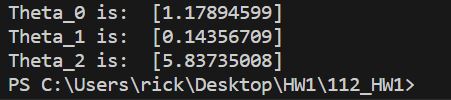




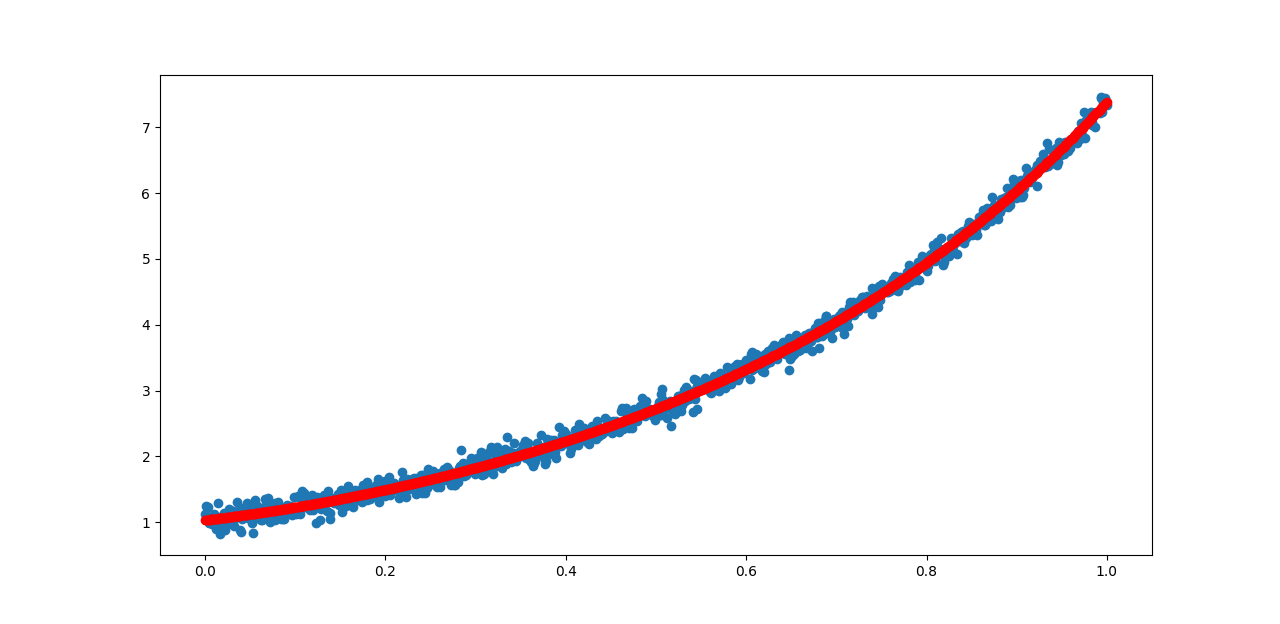
* 1. (**5%**) Compute the least square parabola (i.e. second order polynomial 𝑦 = 𝜃𝟎 +

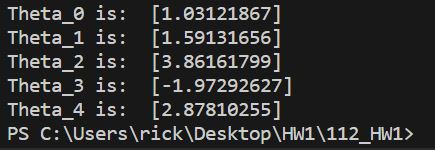
𝑥𝜃𝟏 + 𝑥2𝜃𝟐) to fit the data.



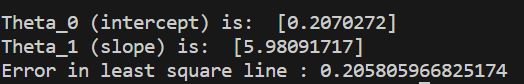


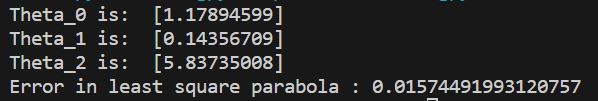
* 1. (**5%**) Compute the least square quartic curve (𝑦 = 𝜃𝟎 + 𝑥𝜃𝟏 + 𝑥2𝜃𝟐 + 𝑥3𝜃3 + 𝑥4𝜃4) to fit the data.

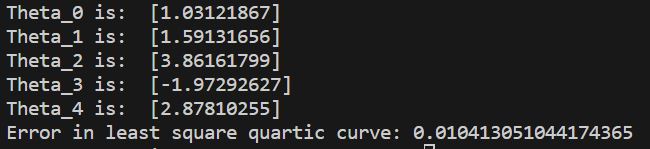




* 1. (**5%**) Explain which formulation (line, parabola, **quartic plane curve**) is more suitable for this dataset and why (please calculate the mean square error for these two fitting equations)?





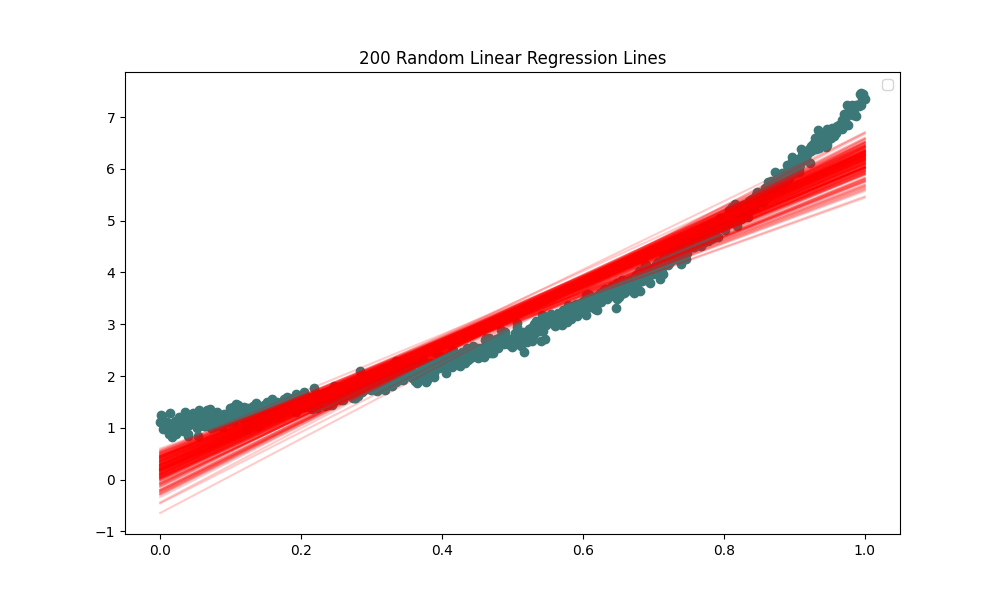


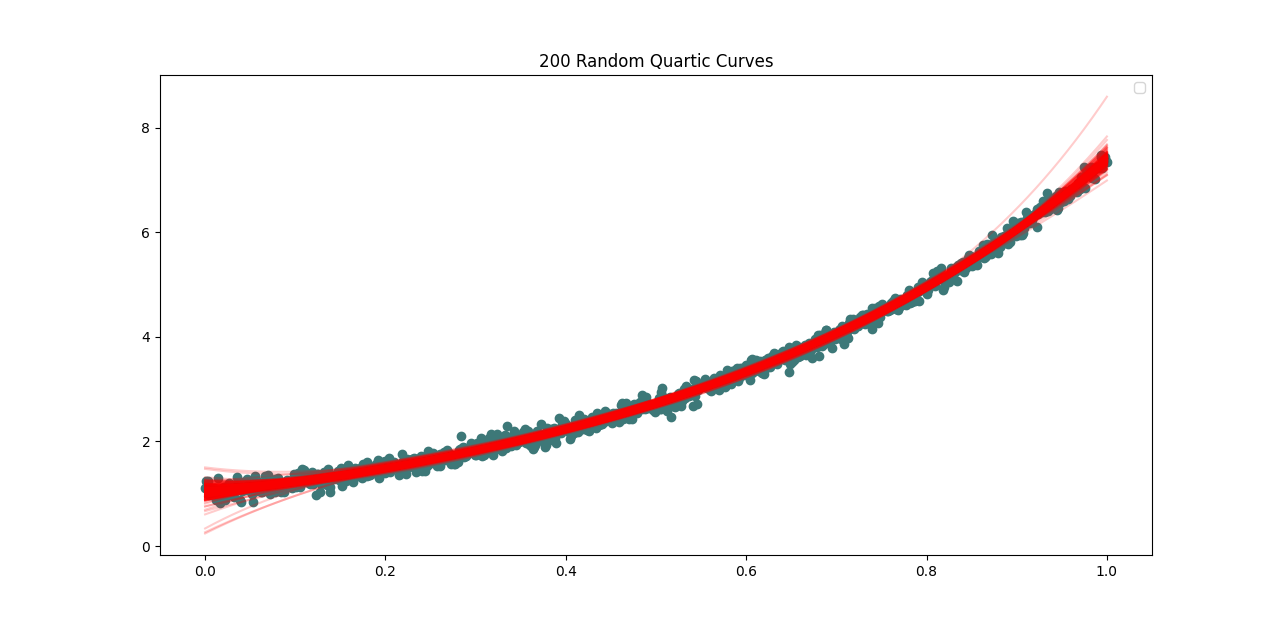
從上面資料可以看到quartic plane curve的誤差值比較小，所以使用quartic plane curve來做預測分析比較好。

1. (**25%**) Following the previous two questions, please randomly select 30 data samples for 200 times and plot these 200 lines (𝑦 = 𝜃𝟎 + 𝑥𝜃𝟏) and quartic curves (𝑦 = 𝜃𝟎 + 𝑥𝜃𝟏 +

𝑥2𝜃𝟐 + 𝑥3𝜃3 + 𝑥4𝜃4**)** in two separate figures, one for lines and the other for quartic

curves. Explain these visualizations based on the bias and variance.





1. (**15%**) In ‘train.mat,’ you can find 2-D points X=[x1, x2] and their corresponding labels

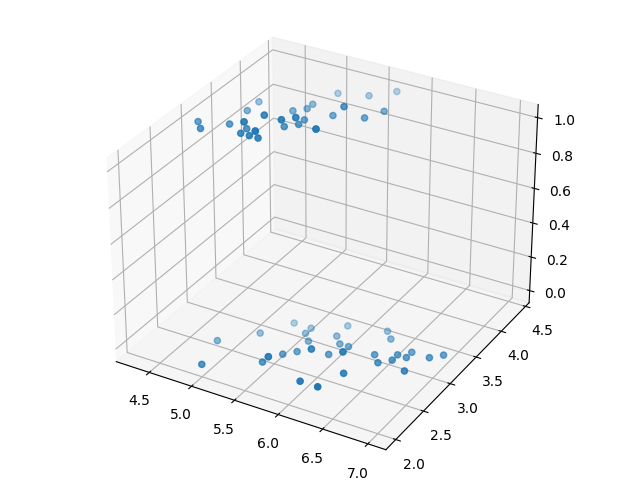
Y=y. Please use logistic regression ℎ(𝜽) = 1

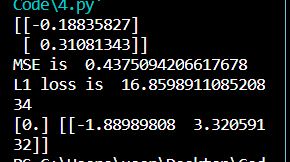
1+𝑒−𝜽𝑇𝑥

to find the decision boundary

(optimal 𝜽∗) based on ‘train.mat.” Please report the test error on the test dataset

‘test.mat.’ (percentage of misclassified test samples)





1. Download the MNIST dataset using the following example code:

##############################################

from future import print\_function import keras

from keras.datasets import mnist

# input image dimensions 28x28 img\_rows, img\_cols = 28, 28

# the data, split between train and test sets

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

x\_train = x\_train.astype('float32') x\_test = x\_test.astype('float32') x\_train /= 255

x\_test /= 255 ##############################################

Please randomly choose 5,000 different handwritten images from either the training or the testing dataset to construct your own dataset, where each digit has 500 data samples.

* 1. (**5%**) Use the following code to show 50 images in your own dataset.

##############################################

import numpy as np

import matplotlib.pyplot as plt amount= 50

lines = 5

columns = 10

number = np.zeros(amount)

for i in range(amount): number[i] = y\_test[i] # print(number[0])

fig = plt.figure()

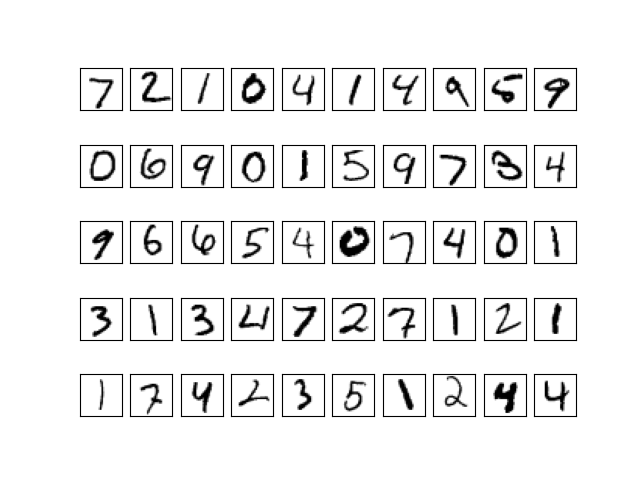
for i in range(amount):

ax = fig.add\_subplot(lines, columns, 1 + i) plt.imshow(x\_test[i,:,:], cmap='binary') plt.sca(ax)

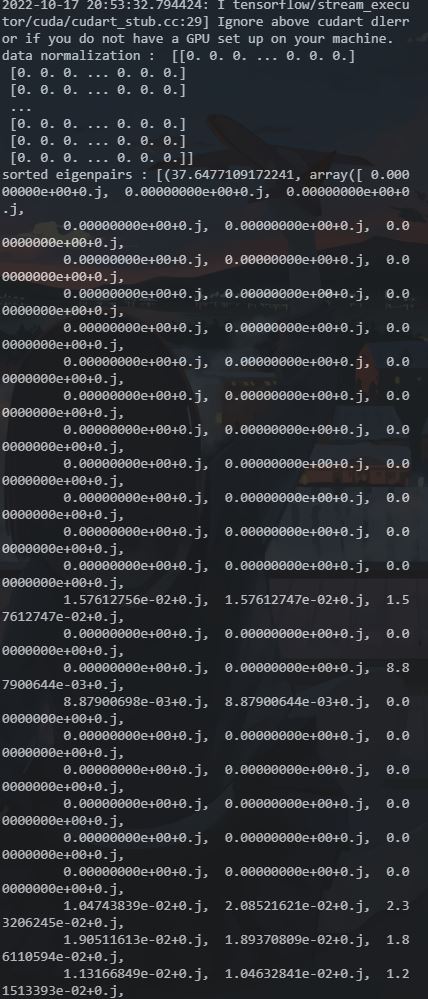
ax.set\_xticks([], [])

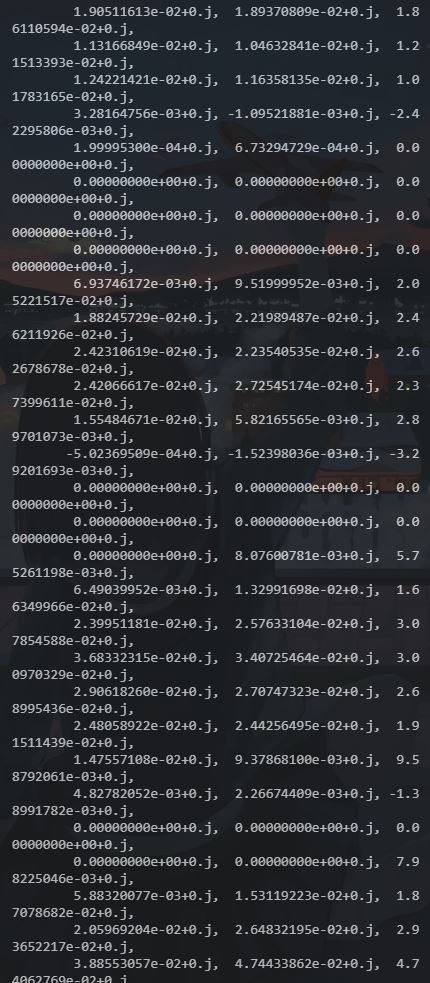
ax.set\_yticks([], [])

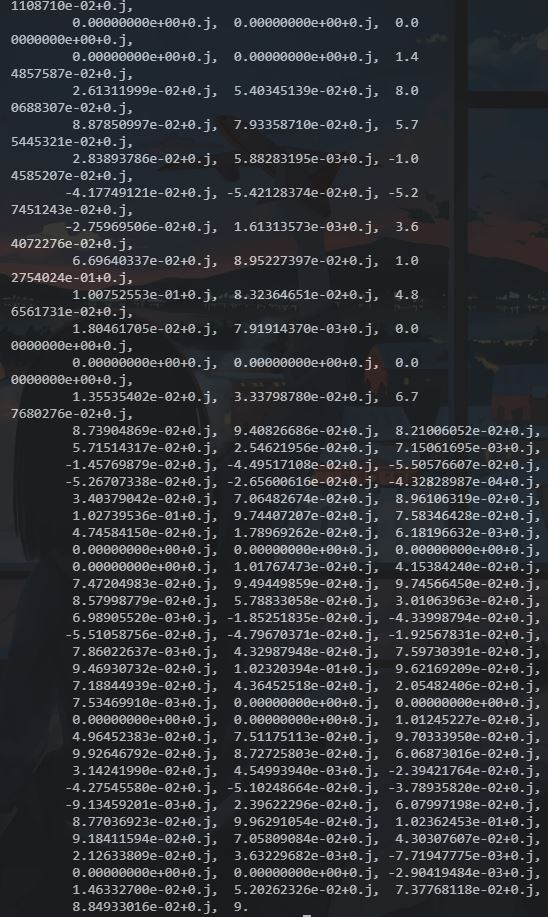
plt.show() ##############################################



* 1. (**15%**) Normalize the data (subtracting the mean from it and then dividing it by the standard deviation) and compute the eigenpairs for the covariance of the data (sorted in a descending order based on eigenvalues).







* 1. (**15%**) Please use PCA to reduce the 784 dimensional data to that with 500, 300, 100, and 50 dimensions, and then show 10 decoding results for each digit, respectively. How do you interpret these results?

