Homework 1
By: Jaron Ho (1005011)

Answers rounded to 5 s.f. where required.

Q1
(a)
No such classifier exists, because there is a positive example and a negative example that are of the same distance from the origin ($(-1,-1)$ and $(1,-1)$ respectively). As such, both of these examples will always be classified in the same group, whether they are outside a circle where $r<1$, or inside a circle where $r>1$. This would result in at least 1 error, no matter what value of r is used.

(b)
If we consider positive targets to lie strictly below the classifier line (i.e. points along the classifier line are negative), then $\theta=(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$ is the normal vector to the classifier line. This classifier line runs through both of the negative examples $(1,-1)$ and $(-2,2)$, so they are correctly classified as negative. Meanwhile, both of the positive examples $(-1,-1)$ and $(-2,1)$ lie below the classifier line, so they are correctly classified as positive.

Q2
(a)
$\theta=[-2.1296,-5.1108]$
$\theta_0=0.0$
Accuracy = 98.390%

(b)
$\theta=[-0.67839,-9.9954]$
$\theta_0=2.0$
Accuracy = 97.988%

(c)
Package versions:
- Python: 3.10.9
- Jupyter notebook: 6.5.2
- Numpy: 1.24.1
- Matplotlib: 3.6.2

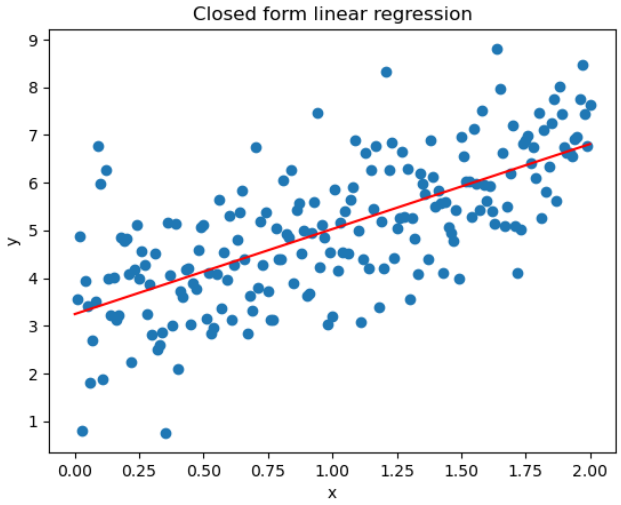
Steps:
1. Unzip Homework1.zip to extract Homework1.ipynb.
2. Download HW1_data.zip from eDimension into the directory containing Homework1.ipynb.
3. Unzip HW1_data.zip.
4. Open Homework1.ipynb with Jupyter notebook.
5. Run all cells.

Q3

(a)

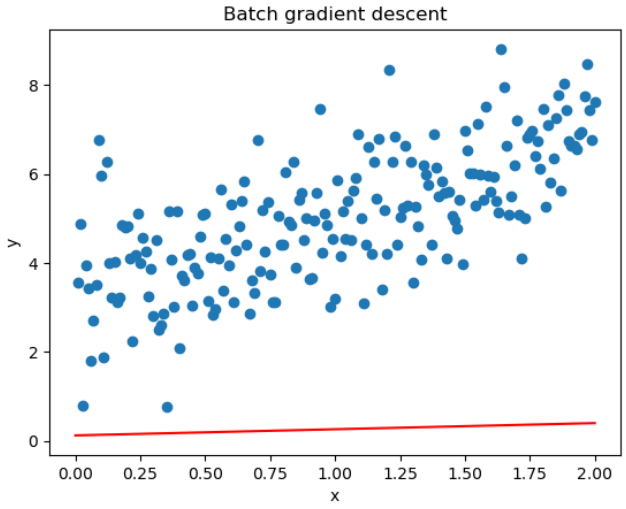
$\theta = [1.7816, 3.2447]$

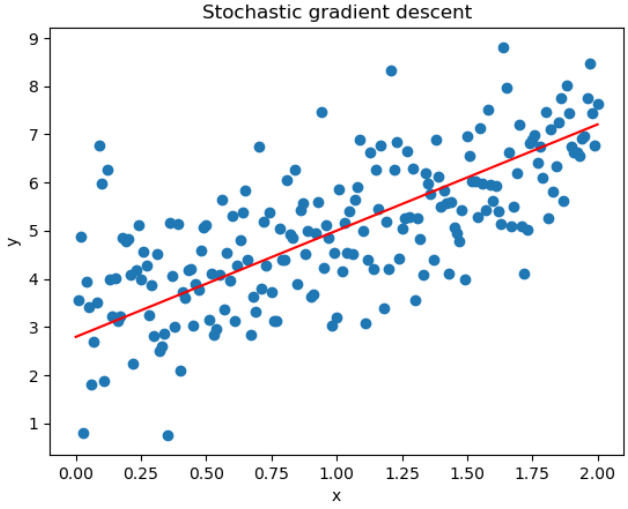
Training error = 0.58126



(b)

	Batch gradient descent	**Stochastic gradient descent**
θ	[0.13823, 0.12323]	[2.1861, 2.7383]
Miminum training error	12.422	0.61352



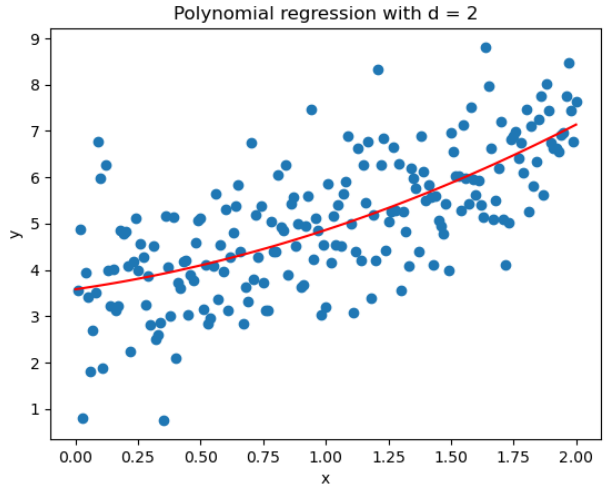


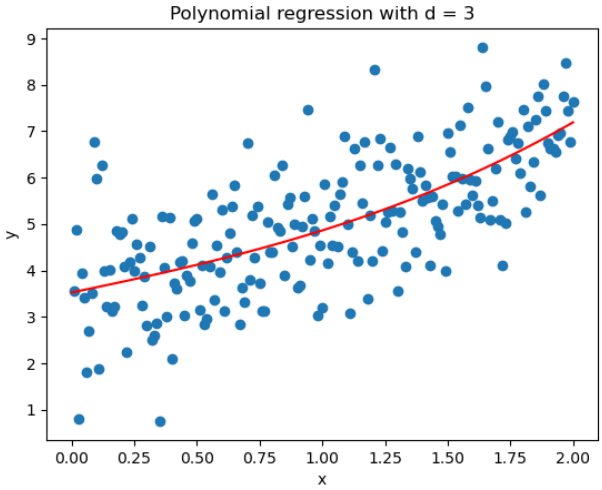
(c)

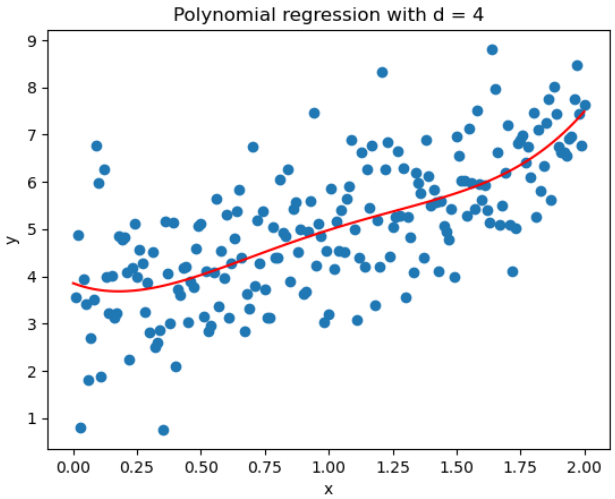
d	**Training error**
2	0.57030
3	0.57008
4	0.56462
5	0.56390
6	0.56134
7	0.55624
8	0.55378
9	0.55292
10	0.55259
11	0.63022
12	29.170
13	3.6260
14	3.2676
15	18.869

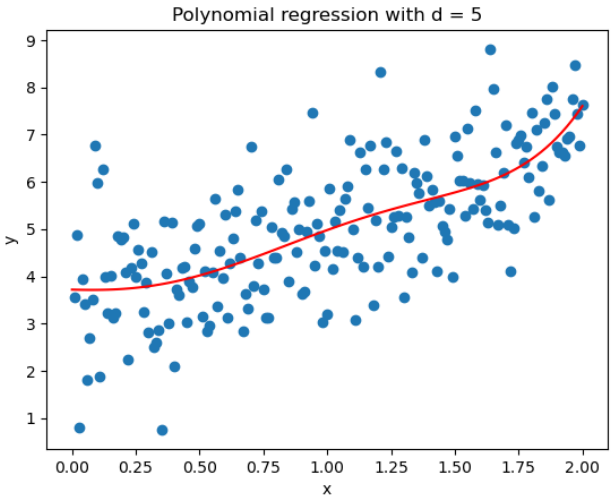
There are two minima at $d = 10$ and $d = 14$. Since the training error is lower when $d = 10$, it means that the training error is minimised at $d = 10$.

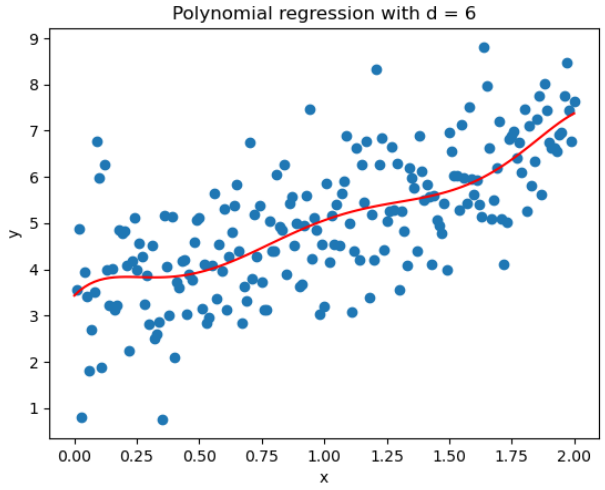
Hence, the error got worse after the 10^{th} order fit.

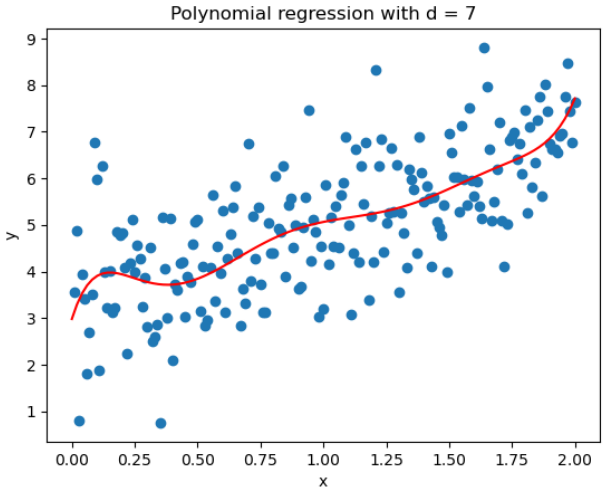


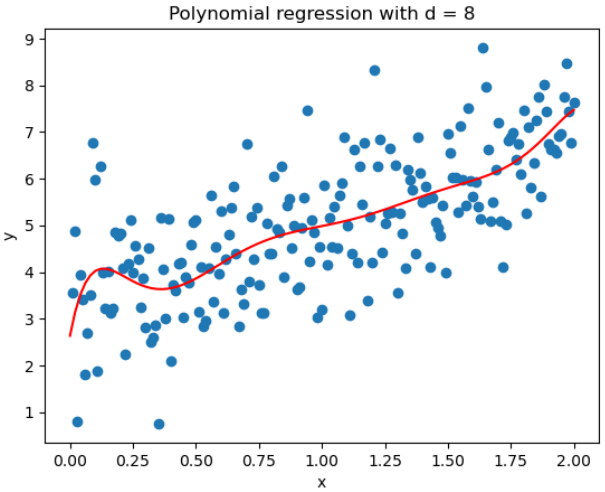


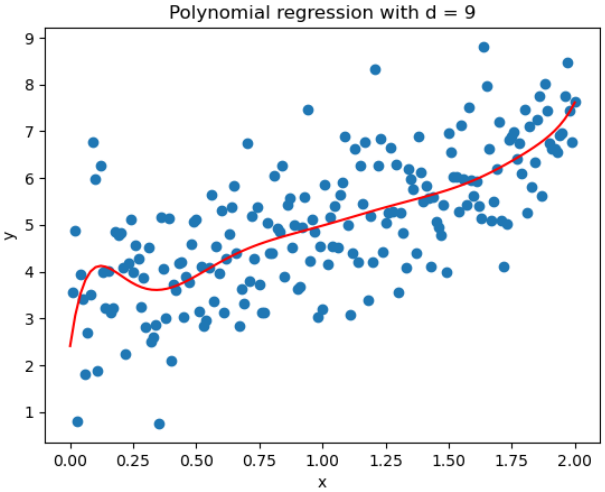


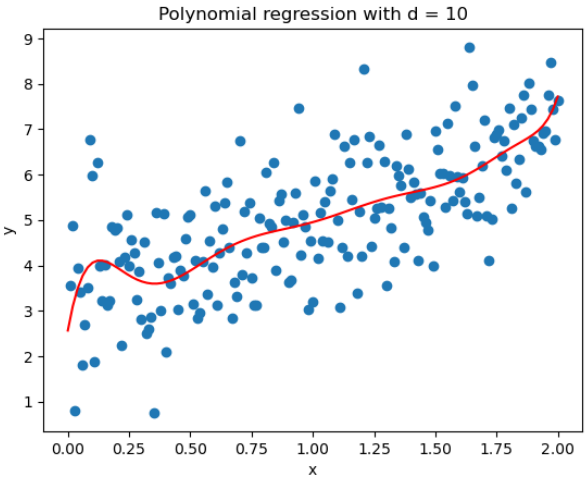


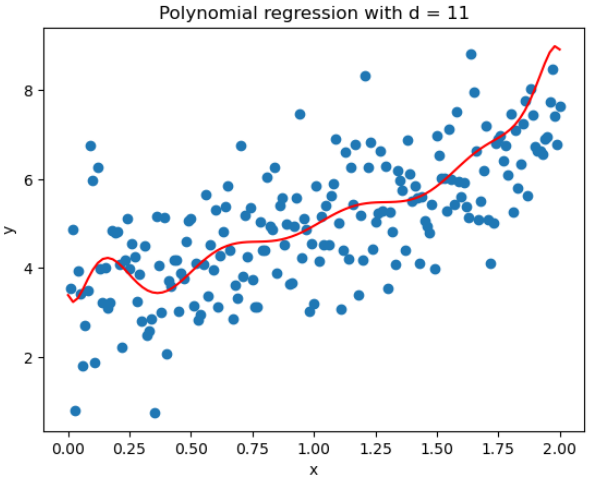

Polynomial regression with d = 6
x
y

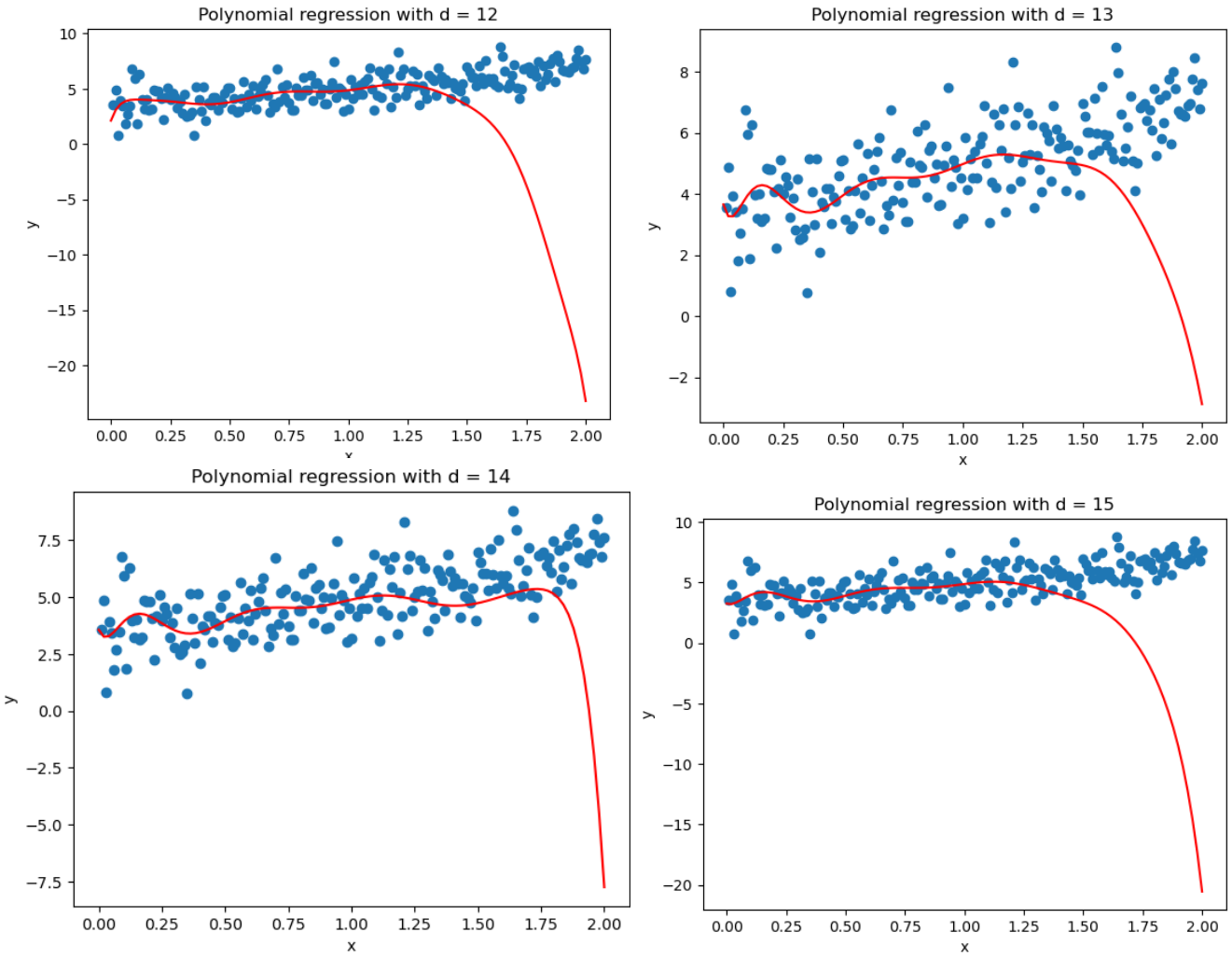

Polynomial regression with d = 7
x
y


Polynomial regression with d = 8
x
y


Polynomial regression with d = 9
x
y


Polynomial regression with d = 10
x
y


Polynomial regression with d = 11
x
y



Q4

(a)

$\theta = [-0.39897, 0.63808, -0.010299, -0.89998]$

(b)

Since the validation loss is a strictly decreasing function over $i \in [0, 4.9]$ where $\lambda = 10^{-i}$, the minimum validation loss occurs when i is maximised. Hence,

$\lambda = 10^{-4.9}$