Machine Learning Homework 1

	Q1	Q2	Q3	Q4	Q5	Total
Maximum score	2	2	2	2	2	10
Expected score	2	2	2	2	2	10

Question 1)

a)

Given the table, we are asked to evaluate $P(Z|\ X\ =\ 0, Y\ =\ 0)$. If we look at the conditions where X and Y are 0, we see that Z can take 1 or 2 values with

$$P(Z = 1|\ X\ =\ 0, Y\ =\ 0) = \frac{0.06}{0.06 + 0.09} = 0.4$$

$$P(Z = 2|\ X\ =\ 0, Y\ =\ 0) = \frac{0.09}{0.06 + 0.09} = 0.6$$

Probabilities, which is the probability mass function of Z.

b)

To check whether Y and Z are independent, we firstly integrate over X on $P(X, Y, Z)$ to get $P(Y, Z)$.

Z\Y	-1	0	1
1	0.12	0.08	0.1
2	0.33	0.17	0.2

Also integrate over Z on $P(Y, Z)$ to get $P(Y)$.

Y	-1	0	1
	0.45	0.25	0.3

Lastly, integrate over Y on $P(Y, Z)$ to get $P(Z)$.

Z	1	2
	0.3	0.7

For independence we need $P(Y,Z) = P(Y)P(Z)$ to hold.

$P(Y)P(Z)$ can be calculated as product of individual elements which is

Z\Y	-1	0	1
1	0.135	0.075	0.09
2	0.315	0.175	0.21

Which is not the same as $P(Y,Z)$, thus, we can conclude these normal variables are not independent.

Question 2)

a)

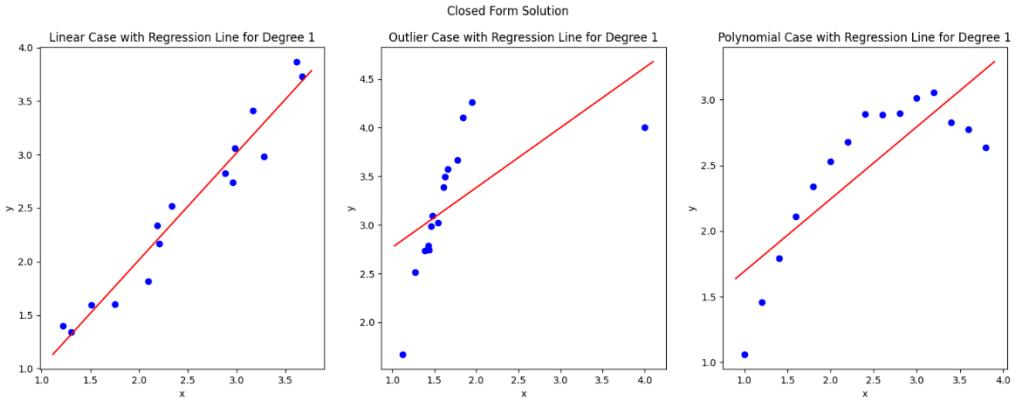

Figure 1: Closed form solution for linear regression

The expected absolute error for the Linear data using closed_form and degree 1 is 0.0312369

The expected absolute error for the Outlier data using closed_form and degree 1 is 0.2781270

The expected absolute error for the Polynomial data using closed_form and degree 1 is 0.1122481

b)

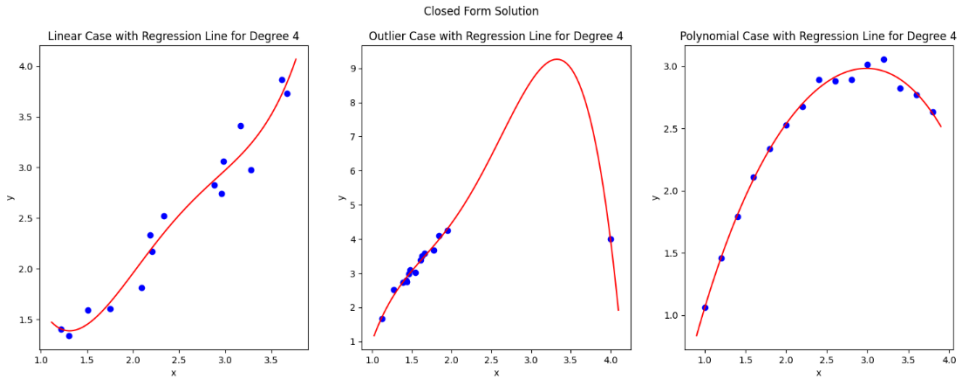

Figure 2: Closed form solution for fourth order polynomial

The expected absolute error for the Linear data using closed_form and degree 4 is 0.0250600

The expected absolute error for the Outlier data using closed_form and degree 4 is 0.0093708

The expected absolute error for the Polynomial data using closed_form and degree 4 is 0.0017873

c)

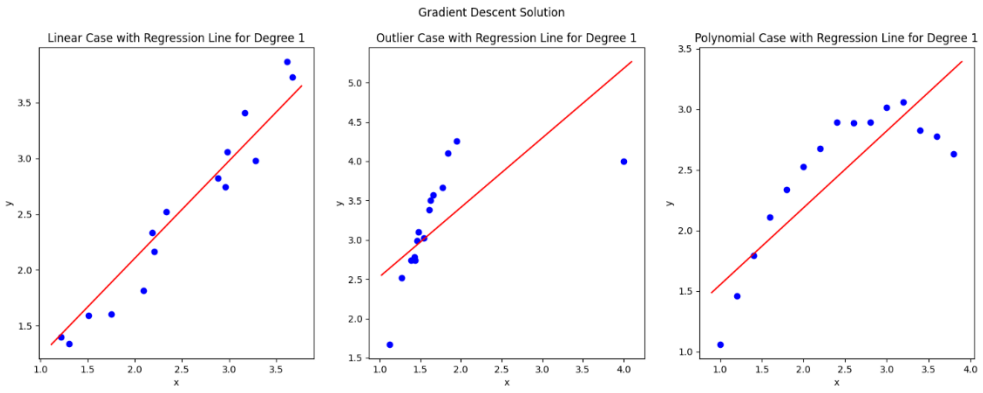

Figure 3: Linear regression with gradient descent

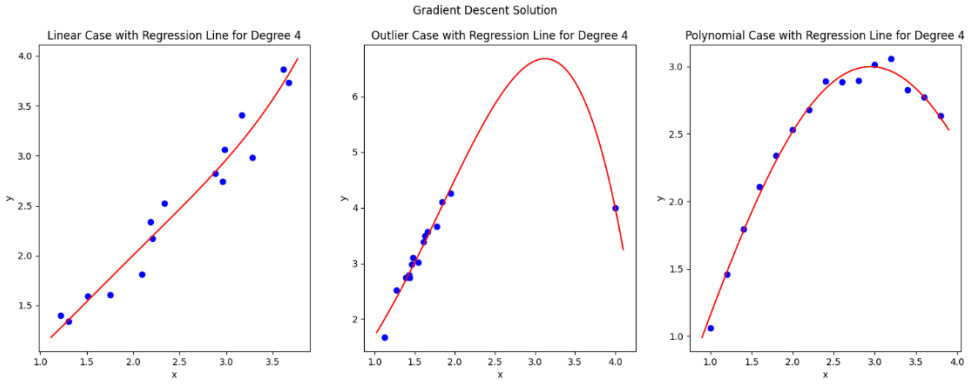

Figure 4: Fourth order regression with gradient descent

The expected absolute error for the Linear data using gradient_descent and degree 4 is 0.0321444

The expected absolute error for the Outlier data using gradient_descent and degree 4 is 0.0187191

The expected absolute error for the Polynomial data using gradient_descent and degree 4 is 0.0075464

The expected absolute error for the Linear data using gradient_descent and degree 1 is 0.0312501

The expected absolute error for the Outlier data using gradient_descent and degree 1 is 0.2826367

The expected absolute error for the Polynomial data using gradient_descent and degree 1 is 0.1126263

1^{st} degree regression	Linear data	Outlier data	Polynomial data
Gradient descent error	0.0312501	0.2826367	0.1126263
Closed form error	0.0312369	0.2781270	0.1122481

4^{th} degree regression	Linear data	Outlier data	Polynomial data
Gradient descent error	0.0321444	0.0187191	0.0075464
Closed form error	0.0250600	0.0093708	0.0017873

As anticipated, closed-form solutions yield lower errors compared to gradient descent methods (though they are quite similar). However, they may also be considered the most effective means

of overfitting the data. The fact that they produce fewer errors during training does not necessarily imply their suitability for practical applications.

Question 3)

$$H = healthy, S = sick, N = negative, P = positive$$

$$P(S) = 0.02, P(H) = 0.98$$

$$P(P|S) = 0.9$$

$$P(N|H) = 0.9, P(P|H) = 0.1$$

To find probability of being sick given the test is positive we expand using bayes formula

$$P(S|P) = \frac{P(P|S)\ x\ P(S)}{P(P)} = \frac{P(P|S)\ x\ P(S)}{P(P|S)\ x\ P(S) + P(P|H)\ x\ P(H)} = \frac{0.9\ x\ 0.02}{0.9\ x\ 0.02 + 0.1\ x\ 0.98} = 0.1552$$

Therefore, the test is approximately 155 times accurate out of 1000 times. I would not trust the test out of a single positive and try to redo it until we approach an acceptable probability.

Question 4)

Given the probabilities and utility matrix, expected utility for each class c_{pred} can be calculated with

$$E\left[U\left(c_{pred}|x\right)\right] = \sum_{c_{true}=1}^{\#classes\ =\ 3} p(c_{true}|x)\ x\ U\left(c_{true}, c_{pred}\right)$$

$$E[U(1|x)] = p(c=1|x)\ x\ U(1,1) + p(c=2|x)\ x\ U(2,1) + p(c=3|x)\ x\ U(3,1)$$
$$= 0.7\ x\ 5 + 0.2\ x\ 0 + 0.1\ x(-3) = 3.2$$

$$E[U(2|x)] = p(c=2|x)\ x\ U(1,2) + p(c=2|x)\ x\ U(2,2) + p(c=3|x)\ x\ U(3,2)$$
$$= 0.7\ x\ 3 + 0.2\ x\ 4 + 0.1\ x\ 0 = 2.9$$

$$E[U(3|x)] = p(c=1|x)\ x\ U(1,3) + p(c=2|x)\ x\ U(2,3) + p(c=3|x)\ x\ U(3,3)$$
$$= 0.7\ x\ 1 + 0.2\ x\ (-2) + 0.1\ x\ 10 = 1.3$$

Based on this, selecting $c_{pred} = 1$ makes the most sense.

Question 5)

a)

For a random variable x, which has Laplace distribution, $p(x) = \frac{1}{2}\exp\left(-|x|\right)$

Laplace distribution can be seen in Figure 5.

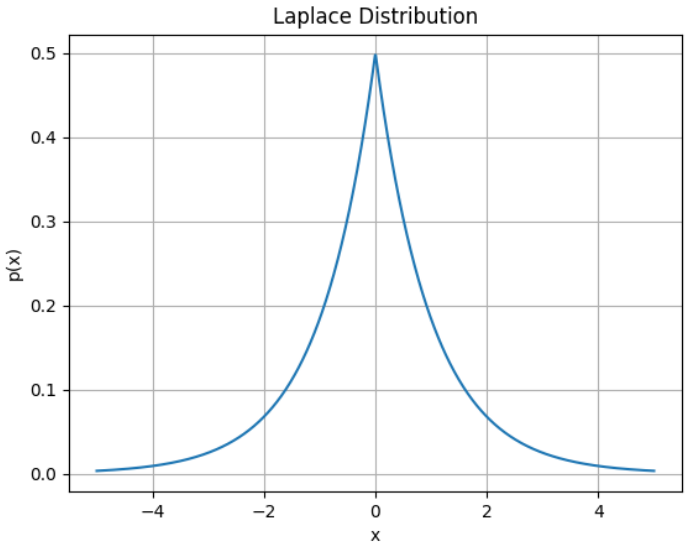

Figure 5: Laplace distribution

The probability that $x > 2$ can be calculated with integrating $p(x)$ from 2 to ∞, that is since x > 0,

$$p(x) = \frac{1}{2}\exp(-x)$$

$$\int_2^{\infty} p(x)dx = 0 + \frac{1}{2}\exp(-2) \cong 0.06767$$

b)

For a random variable x, which has binomial distribution, $p(x) = \binom{n}{x} p^x (1-p)^{n-x}$

$$E[x] = \sum_{x=0}^{n} x \binom{n}{x} p^x (1-p)^{n-x}$$

$$= \sum_{x=1}^{n} x \binom{n}{x} p^x (1-p)^{n-x}$$

$$= \sum_{x=1}^{n} n \binom{n-1}{x-1} p^x (1-p)^{n-x}$$

$$= np \sum_{x=1}^{n} \binom{n-1}{x-1} p^{x-1} (1-p)^{(n-1)-(x-1)}$$

$$= np \sum_{j=0}^{m} \binom{m}{j} p^j (1-p)^{m-j}$$

$$= np$$

$$E[x^2] = \sum_{x=0}^{n} x^2 \binom{n}{x} p^x (1-p)^{n-x}$$

$$= \sum_{x=0}^{n} nx \binom{n-1}{x-1} p^x (1-p)^{n-x}$$

$$= np \sum_{x=1}^{n} x \binom{n-1}{x-1} p^{x-1} (1-p)^{(n-1)-(x-1)}$$

$$= np \sum_{j=0}^{n} (j+1) \binom{m}{j} \; p^j (1-p)^{m-j}$$

$$= np \left(\sum_{j=0}^{n} j \binom{m}{j} \; p^j (1-p)^{m-j} + \sum_{j=0}^{n} \binom{m}{j} \; p^j (1-p)^{m-j} \right)$$

$$= np \left((n-1)p \sum_{j=1}^{n} \binom{m-1}{j-1} \; p^{j-1} (1-p)^{(m-1)-(j-1)} + \sum_{j=0}^{n} \binom{m}{j} \; p^j (1-p)^{m-j} \right)$$

$$= np\big((n-1)p + 1\big)$$

$$= n^2 p^2 + np(1-p)$$

Appendix

See the link below to access the python code used in questions 2 and 5.

https://github.com/Bubuyson/ml_hws