Machine Learning Homework 1

Question 1)

a)

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

Probabilities, which is the probability mass function of Z.

b)

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

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

Also integrate over Z on to get .

|  |  |  |  |
| --- | --- | --- | --- |
| Y | -1 | 0 | 1 |
|  | 0.45 | 0.25 | 0.3 |

Lastly, integrate over Y on to get .

|  |  |  |
| --- | --- | --- |
| Z | 1 | 2 |
|  | 0.3 | 0.7 |

For independence we need to hold.

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  *,* thus, we can conclude these normal variables are not independent.

Question 2)

a)

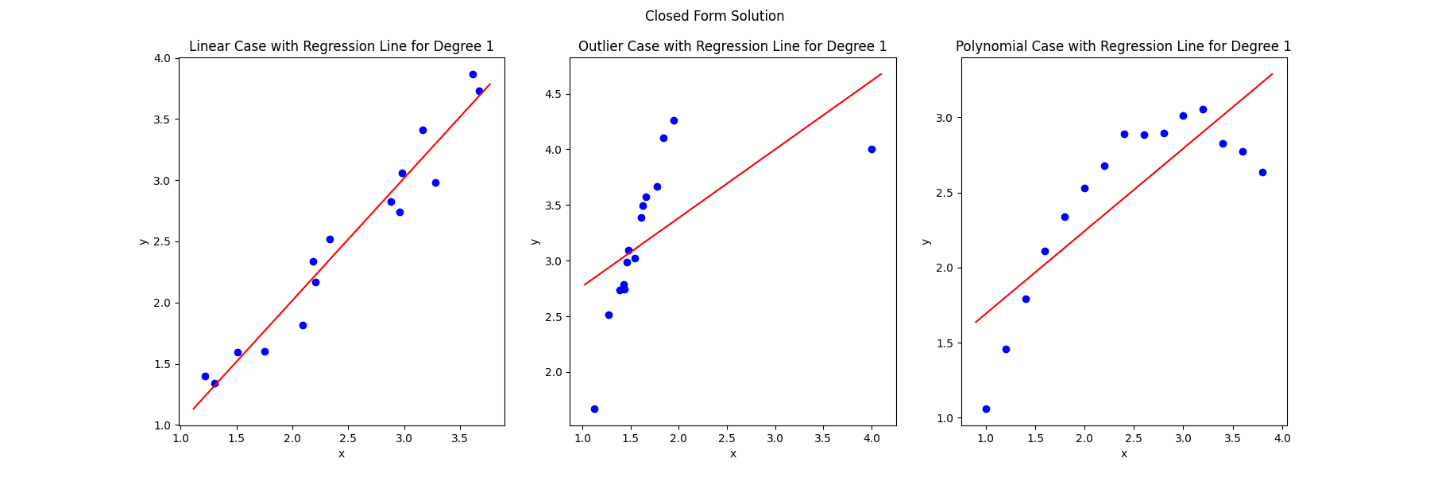


Figure 1: Closed form solution for linear regression

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)

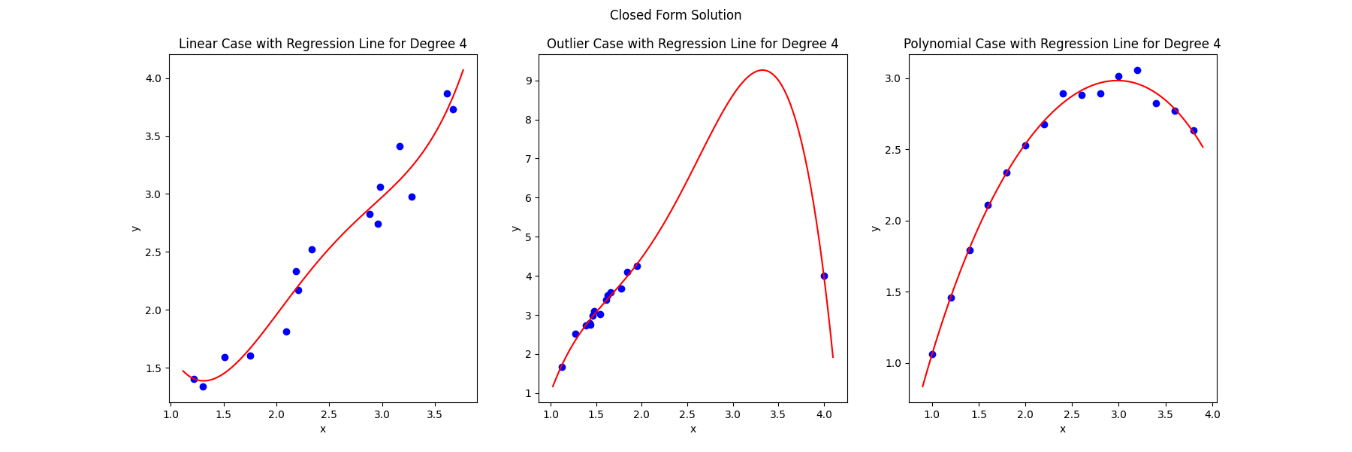


Figure 2: Closed form solution for fourth order polynomial

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)

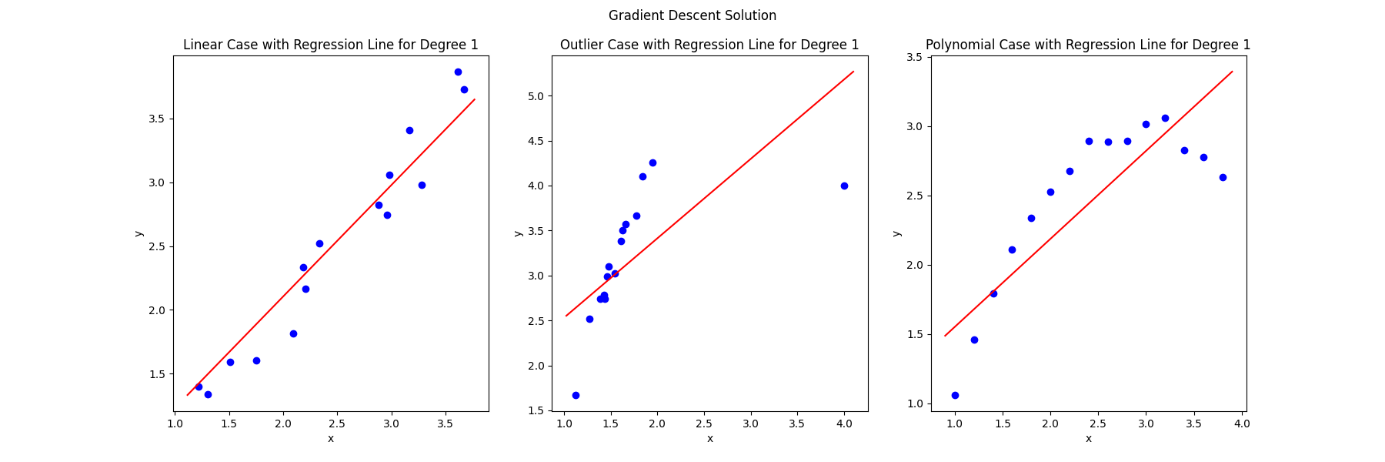


Figure 3: Linear regression with gradient descent

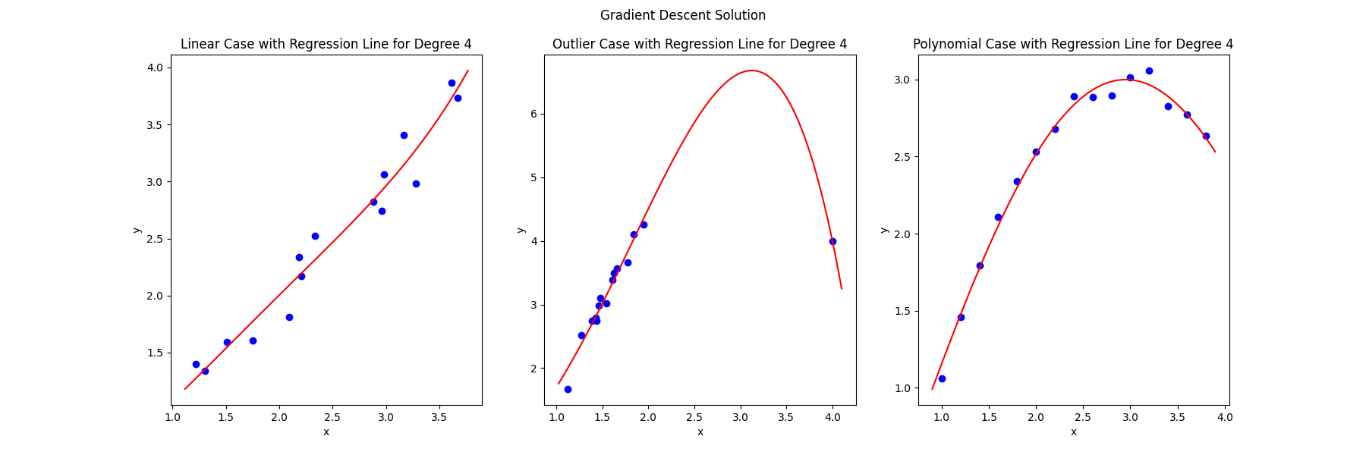


Figure 4: Fourth order regression with gradient descent

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

|  |  |  |  |
| --- | --- | --- | --- |
| 1st 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 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4th 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)

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

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 can be calculated with

Based on this, selecting makes the most sense.

Question 5)

a)

For a random variable x, which has Laplace distribution,

Laplace distribution can be seen in Figure 5.

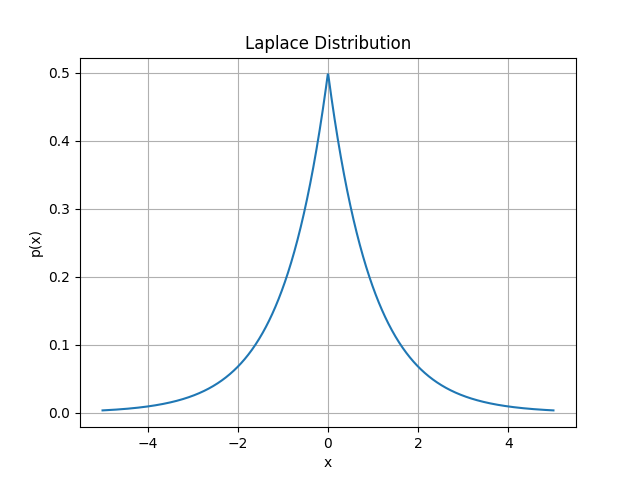


Figure 5: Laplace distribution

The probability that can be calculated with integrating from 2 to , that is since x > 0,

b)

For a random variable x, which has binomial distribution,

# Appendix

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

<https://github.com/Bubuyson/ml_hws>