Victoria Yong
1004455

51.504 Machine Learning (2023)
Homework 1

Due 12 Oct 2023, Thursday, 11.59pm

Question 1

Suppose A is a random variable with state space {2,3} and B is a random variable with state space {1,2,3}. Suppose that the joint probability of A and B follows $P(A = 2, B = 1) = 1/36$, $P(A = 2, B = 2) = 5/18$, $P(A = 2, B = 3) = 1/9$, $P(A = 3, B = 1) = 1/4$ and $P(A = 3, B = 2) = 5/36$. Drawing a probability table might help.

(a) [4 points] Calculate $P(A = 3, B = 3)$.

(b) [4 points] Calculate the marginal distribution of B.

(c) [8 points] Calculate the expectation and variance of B.

(d) [4 points] Calculate $P(A = 2|B = 1)$.

B \ A	2	3	Sum
1	1/36	9/36	10/36
2	10/36	5/36	15/36
3	4/36	7/36	11/36
Sum	15/36	21/36	1

a) $P(A=3, B=3) = \frac{7}{36}$

b) $P(B=1) = \frac{10}{36}$

$P(B=2) = \frac{15}{36}$

$P(B=3) = \frac{11}{36}$

$P(B) = \frac{10}{36} + \frac{15}{36} + \frac{11}{36}$

$= 1$

c) $E(B) = \sum b P(B) = 1 \cdot \frac{10}{36} + 2 \cdot \frac{15}{36} + 3 \cdot \frac{11}{36} = \frac{73}{36}$

$$\begin{aligned}
Var(B) &= E(B^2) - E(B)^2 = \sum b^2 P(b) - \left(\frac{73}{36}\right)^2 \\
&= 1\left(\frac{10}{36}\right) + 4\left(\frac{15}{36}\right) + 9\left(\frac{11}{36}\right) - \frac{5329}{36^2} \\
&= \frac{100}{1296} + \frac{3600}{1296} + \frac{9801}{1296} - \frac{5329}{1296} \\
&= \frac{755}{1296}
\end{aligned}$$

d) $P(A=2 | B=1) = \frac{P(A=2, B=1)}{P(B=1)} = \frac{1/36}{10/36} = \frac{1}{10}$

Victoria Yong
1004455

Question 2

(a) Let X be a discrete random variable with state space {1,2,3}, a hypothetical three sided die. Suppose that the probabilities associated with the state space are

$$P(X = 1) = 2\theta,$$
$$P(X = 2) = 3\theta,$$
$$P(X = 3) = 1 - 5\theta.$$

Calculate the range of possible values of the parameter θ. [4 points]

Calculate the Maximum Likelihood Estimator of X. [8 points]

(This estimator should be expressed in terms of x_1, x_2, and x_3, where x_i is the number of times i shows up in the data sample set. If 1 shows us five times, then $x_1 = 5$.)

$$L(\theta, X) = \prod_{x=1}^{3} P(x) = 2\theta \cdot 3\theta \cdot (1-5\theta)$$
$$= 6\theta^2 - 30\theta^3$$

$$\frac{d}{d\theta} L(\theta, X) = 0$$
$$12\theta - 90\theta^2 = 0$$
$$\theta(2 - 15\theta) = 0$$
$$\theta = 0 \quad \text{or} \quad \theta = \frac{2}{15}$$

$$\frac{d^2}{d\theta^2} L(\theta, X) < 0$$
$$12 - 180\theta < 0$$
$$\theta > -\frac{1}{15}$$

$$\therefore \; 0 \leq \theta \leq \frac{2}{15}$$

To find MLE,

$$\frac{d}{d\theta} P(\text{data} \mid \theta) = \frac{d}{d\theta} (2\theta)^{x_1} \cdot (3\theta)^{x_2} \cdot (1-5\theta)^{x_3}$$
$$= \frac{d}{d\theta} 2^{x_1}\theta^{x_1} 3^{x_2}\theta^{x_2} (1-5\theta)^{x_3}$$
$$= 2^{x_1}3^{x_2}\left[x_1\theta^{x_1-1}(\theta^{x_2}(1-5\theta)^{x_3}) + x_2\theta^{x_2-1}(\theta^{x_1}(1-5\theta)^{x_3}) + x_3(1-5\theta)^{x_3-1}(-5)\theta^{x_1}\theta\right] = 0$$

$$x_1\theta^{x_1-1}\theta^{x_2}(1-5\theta)^{x_3} + x_2\theta^{x_2-1}\theta^{x_1}(1-5\theta)^{x_3} - 5x_3\theta^{x_1}\theta^{x_2}(1-5\theta)^{x_3-1} = 0$$

$$x_1\theta^{-1} + x_2\theta^{-1} - 5x_3(1-5\theta)^{-1} = 0$$
$$\frac{x_1 + x_2}{\theta} = \frac{5x_3}{1-5\theta}$$
$$\frac{1}{\theta} - 5 = \frac{5x_3}{x_1 + x_2}$$
$$\hat{\theta} = \frac{x_1 + x_2}{5(x_1 + x_2 + x_3)}$$

The MLE

$$\therefore \; \hat{\theta} = \frac{x_1 +}{5(x_1 + x \quad x_3)}$$

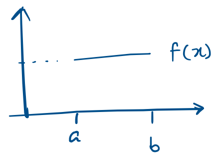
Victoria Yong
1004455

(b) [8 points] (*) Let $X_1,\ldots,X_N$ be a sample set of Uniform(a,b), where $b > a$ and a and b are unknown parameters. Find the MLE $\hat{a}$ and $\hat{b}$. (Write out the likelihood function in terms of indicator functions. An indicator function $1_S(x)$ gives output 1 when x is in the set S and outputs 0 when x is not in the set S.)

$$f(x) = \frac{1}{b-a}$$

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$



$$L(x_i ; a, b) = \prod_{i=1}^{n} \frac{1}{b-a} = \frac{1}{(b-a)^n}$$

$$\ln L(x_i ; a, b) = \ln(1) - \ln((b-a)^n)$$
$$= 0 - n\ln(b-a)$$
$$= -n\ln(b-a)$$

MLE,

$$\hat{a} = \frac{d}{da} \ln L(x_i ; a, b) = -n \frac{1}{(b-a)\ln e}(-1) = \frac{n}{b-a}$$

$$\hat{b} = \frac{d}{db} \ln L(x_i ; a, b) = -n \frac{1}{b-a\ln e}(1) = -\frac{n}{b-a}$$

In indicator function,

$$L(x ; a, b) = [\![x \in [a,b]]\!]$$

Victoria Yong
1004455

Question 3

Suppose we want to perform 2-means clustering on the data set {1,2,3,8,9,10}, and the initial clustering sets are $C_1 = \{2,8\}$ and $C_2 = \{1,3,9,10\}$

(a) [8 points] Compute the centroids μ_1 of C_1 and μ_2 of C_2.

(b) [4 points] Next, compute the new clusters formed by these centroids μ_1 and μ_2. Label the cluster associated with μ_1 as D_1, and the cluster associated with μ_2 as D_2.

(c) [8 points] Calculate the new centroids of D_1 and D_2. Is this clustering stable?

a) $\mu_1 = \frac{2+8}{2} = 5$

$\mu_2 = \frac{1+3+9+10}{4} = \frac{23}{4} = 5\frac{3}{4}$

b)
$|5-1| = 4$ $\quad |5\frac{3}{4} - 1| = 4\frac{3}{4}$ $\quad \therefore 1 \in D_1$

$|5-2| = 3$ $\quad |5\frac{3}{4} - 2| = 3\frac{3}{4}$ $\quad \therefore 2 \in D_1$

$|5-3| = 2$ $\quad |5\frac{3}{4} - 3| = 2\frac{3}{4}$ $\quad \therefore 3 \in D_1$

$|5-8| = 3$ $\quad |5\frac{3}{4} - 8| = 2\frac{1}{4}$ $\quad \therefore 8 \in D_2$

$|5-9| = 4$ $\quad |5\frac{3}{4} - 9| = 3\frac{1}{4}$ $\quad \therefore 9 \in D_2$

$|5-10| = 5$ $\quad |5\frac{3}{4} - 10| = 4\frac{1}{4}$ $\quad \therefore 10 \in D_2$

$\therefore$ $D_1 = \{1, 2, 3\}$

$D_2 = \{8, 9, 10\}$

c) $\mu_1 = \frac{1+2+3}{3} = 2$

$\mu_2 = \frac{8+9+10}{3} = 9$

$|2-1| = 1$ $\quad |9-1| = 8$ $\quad \therefore 1 \in D_1$

$|2-2| = 0$ $\quad |9-2| = 7$ $\quad \therefore 2 \in D_1$

$|2-3| = 1$ $\quad |9-3| = 6$ $\quad \therefore 3 \in D_1$

$|2-8| = 6$ $\quad |9-8| = 1$ $\quad \therefore 8 \in D_2$

$|2-9| = 7$ $\quad |9-9| = 0$ $\quad \therefore 9 \in D_2$

$|2-10| = 8$ $\quad |9-10| = 1$ $\quad \therefore 10 \in D_2$

$\therefore$ No change of clustering with updated mean.

Clustering is stable

$D_1 = \{1, 2, 3\}$

$D_2 = \{8, 9, 10\}$

Victoria Yong
1004455

Question 4

A function of X and Y can be expressed by

$$f(x,y) = \begin{cases} \frac{6}{7}\left(x^2 + \frac{xy}{2}\right), & \text{if } 0 \le x \le 1, 0 \le y \le 2; \\ 0, & \text{otherwise.} \end{cases}$$

(a) [6 points] Show that $f(x,y)$ is a joint probability density function.

(b) [6 points] Find the probability density function of X.

(c) [6 points] Hence or otherwise, find the conditional density function $f_{Y|X}(y \mid x = 1)$.

(d) [2 points] Are X and Y independent? Explain.

a) $f(x,y) = \frac{6}{7}\left(x^2 + \frac{xy}{2}\right)$

$$\int_0^2\int_0^1 f(x,y)\,dx\,dy = \frac{6}{7}\int_0^2\int_0^1 x^2 + \frac{xy}{2}\,dx\,dy$$
$$= \frac{6}{7}\int_0^2 \left[\frac{x^3}{3} + \frac{x^2 y}{4}\right]_0^1 dy$$
$$= \frac{6}{7}\int_0^2 \frac{1}{3} + \frac{y}{4}\,dy$$
$$= \frac{6}{7}\left[\frac{1}{3}y + \frac{y^2}{8}\right]_0^2 dy$$
$$= \frac{6}{7}\left[\frac{2}{3} + \frac{1}{2}\right]$$
$$= \frac{6}{7}\left[\frac{7}{6}\right]$$
$$= 1$$

$\therefore f(x,y)$ is a joint probability density function.

b) $f_X(x) = \int_0^2 \frac{6}{7}\left(x^2 + \frac{xy}{2}\right) dy = \frac{6}{7}\left[x^2 y + \frac{xy^2}{4}\right]_0^2 = \frac{6}{7}\left(2x^2 + x\right)$

c) $f_{Y|X}(y \mid x) = \dfrac{f_{X,Y}(x,y)}{f_X(x)} = \dfrac{\frac{6}{7}\left(x^2 + \frac{xy}{2}\right)}{\frac{6}{7}\left(2x^2 + x\right)} = \dfrac{x + \frac{y}{2}}{2x+1} = \dfrac{2x+y}{4x+2}$

$\therefore f_{Y|X}(y \mid x=1) = \dfrac{y+2}{6}$

Victoria Yong
1004455

d) $f_Y(y) = \int_0^2 \frac{6}{7}\left(x^2 + \frac{xy}{2}\right) dy = \frac{6}{7}\left[x^2 y + \frac{xy^2}{4}\right]_0^2$

$= \frac{6}{7}(2x^2 + x)$

$f_X(x) f_Y(Y) = \frac{6}{7}\left(\frac{1}{3} + \frac{y}{4}\right) \cdot \frac{6}{7}\left(2x^2 + x\right)$

$= \frac{6}{7}\left(\frac{4 + 3y}{12}\right) \cdot \frac{6}{7}\left(2x^2 + x\right)$

$= \frac{6}{7}\left(\frac{8x^2 + 4x + 6x^2 y + 3xy}{12}\right)$

$\neq f(x, y)$

$\therefore$ X and Y are not independent.

Victoria Yong
1004455

Question 5

This exercise will help us perform clean up, model selection and cross validation on data, and we will do so with a RIASEC data set. RIASEC is a psychological personality test that is used to evaluate people's personalities so as to determine what kind of work is suitable for them. Mathematically, each of the six letters R, I, A, S, E and C corresponds to six independent personality types, and a RIASEC score for a person's personality is therefore a six dimensional vector, of which each entry takes a value from the set $\{1,2,3,4,5\}$.

In the personality test, for each of the six personality traits, each person is asked eight questions, each also taking a score from 1 to 5. The score for each trait is calculated by summing up all the scores of the eight questions and averaging them (so dividing by 8). The data set is denoted "RIASEC.csv", and the codebook to understand the data is denoted "RIASECcodebook", which points to a website that helps you to understand how the data set is structured. It includes other information about the people who answered the questions, but for the purposes of this exercise we will ignore them and also only deal with one trait, R, which stands for "Realistic". Again you can use R or Matlab or whatever you prefer.

The vertical index for the data is the sample size, which is the list of people who answered these questions. There are about 8000 people.

(a) [6 points] (Cleaning Up) From the data set, write a code to construct a data set or matrix that reports the responses of everyone only for the eight questions related to the "Realistic" trait, which are R1 to R8. You are basically truncating the matrix here to an 8000+ by 8 matrix. Next, you will notice there are some -1 entries in the matrix, these are the people who left the answers blank or drew some squiggly smiley face that makes no sense. Write a code to get rid of these people altogether. By this I mean, you want to remove the rows with -1 on it, not launch nuclear missiles at them. This is essentially cleaning up the data. Since you won't be able to print an 8000+ by 8 matrix on paper, please submit code instead.

(b) [8 points] (Model selection) We want to see how the answers for the first question, R1, correlates with their R score. However we want to test the validity of our theory, so we will use the first 6500 people as training data. Therefore, write a code to compute the R score for each person. Treat the R score as the dependent variable and the R1 score as the independent variable, and compute the estimated regression function and the residual sum of squares for the first 6500 people.

(c) [6 points] (Validation) We now want to see if our previous model generalizes well. Therefore, using the regression model in part b, calculate the residual sum of squares for the remaining people and compare the residual sum of squares (averaged) with that (averaged as well) for part b.

51.504 Machine Learning (Professional)

HW1 Q5

Victoria Yong 1004455

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

(a) [6 points] (Cleaning Up) From the data set, write a code to construct a data set or matrix that reports the responses of everyone only for the eight questions related to the “Realistic” trait, which are R1 to R8. You are basically truncating the matrix here to an 8000+ by 8 matrix. Next, you will notice there are some −1 entries in the matrix, these are the people who left the answers blank or drew some squiggly smiley face that makes no sense. Write a code to get rid of these people altogether. By this I mean, you want to remove the rows with −1 on it, not launch nuclear missiles at them. This is essentially cleaning up the data. Since you won’t be able to print an 8000+ by 8 matrix on paper, please submit code instead.

In [2]:

```
data =  np.genfromtxt("RIASEC.csv", names=True, dtype=int, usecols=[f"R{x}" for x in range
labels = data.dtype.names

data = data.view((int, len(data.dtype.names)))
data = data[~np.any(data == -1, axis=1)]


print(labels)
print(data.shape)
print(data[:5])
```

```
('R1', 'R2', 'R3', 'R4', 'R5', 'R6', 'R7', 'R8')
(8478, 8)
[[3 1 4 2 1 2 1 1]
 [1 1 1 1 1 1 1 1]
 [3 2 1 1 1 1 2 1]
 [3 2 1 2 2 3 1 2]
 [3 1 3 4 3 4 3 3]]
```

(b) [8 points] (Model selection) We want to see how the answers for the first question, R1, correlates with their R score. However we want to test the validity of our theory, so we will use the first 6500 people as training data. Therefore, write a code to compute the R score for each person. Treat the R score as the dependent variable and the R1 score as the independent variable, and compute the estimated regression function and the residual sum of squares for the first 6500 people.

In [3]:

```
class Regression():


    def __init__(self, X, y, X_test, y_test, **kwargs):
        self.X = X
        self.y = y
        self. X_t = X_test
        self. y_t = y_test

        self.lr = kwargs.pop('lr', 0.05)
        self.epochs = kwargs.pop('epochs', 10)
        self.order = kwargs.pop('order', 15)


    def fit(self, option = 0):
        """
```

```
        Function to calculate linear regression model.

        Takes in option argument where
            0: Closed form multivariate linear regression
            1: Closed form linear regression

        Returns theta and predictions.
        """
        self.option = option
        if option == 0:
            theta, preds, error = self._closed_multivariate_linear_regression(self.X, self
            title = f"Closed Multivariate Linear Regression | Train Error: {error}"

        elif option == 1:
            theta, preds, error = self._closed_linear_regression(self.X, self.y)
            title = f"Closed Linear Regression | Train Error: {error}"

        else:
            return 'Invalid argument.'

        self._plot(self.X, self.y, preds, title)


    def test(self):

        """
        Function to calculate linear regression model on test data.
        Returns theta and predictions.
        """
        if self.option == 0:
            theta, preds, error = self._closed_multivariate_linear_regression(self.X_t, se
            title = f"Closed Multivariate Linear Regression | Test Error: {error}"

        elif self.option == 1:
            theta, preds, error = self._closed_linear_regression(self.X_t, self.y_t)
            title = f"Closed Linear Regression | Test Error: {error}"

        else:
            return 'Need to fit model first before testing.'

        self._plot(self.X_t, self.y_t, preds, title)

    def _closed_linear_regression(self, X, y):
        """
        Closed form linear regression. A linear regression can be modelled as
            theta = mean(y) - [r * (SD(y)/SD(x))] * mean(x),
            where r = 1/n * sum(xi*yi)

        X: Input vector
        y: Label vector
        Returns predictions and theta as numpy array

        Note: This didn't work well. Not sure why.
        """

        n = X.shape[0]
        r = 1/n * np.sum(np.multiply(X, y))

        theta = np.mean(y) - (r * (np.std(y)/np.std(X)) * np.mean(X))
        preds = X.dot(theta)
        error = self._sse(y, preds)

        return theta, self.preds, error


    def _closed_multivariate_linear_regression(self, X, y):
```

```
        """
        Closed form multivariate linear regression. A multivariate linear regression can b
            theta = inv(X.T dot X) dot (X.T dot y)

        X: Input vector
        y: Label vector
        Returns predictions and theta as numpy array
        """

        theta = np.dot(np.linalg.inv(X.T.dot(X)), X.T.dot(y))
        preds = X.dot(theta)
        error = self._sse(y, preds)

        return theta, preds, error

    def _sse(self, y, preds):
        """
        Averaged residual sum of squares, as requested by part c. This is basically the MS
        Formula from wikipedia: https://en.wikipedia.org/wiki/Residual_sum_of_squares
        """
        return sum(np.square((y - preds))) / y.shape[0]


    def _plot(self, X, y, preds, title="model", dim = 1):

        plt.scatter(X[:, dim], y, color='g')
        plt.plot(X[:, dim], preds, color='b')
        plt.title(title)
        plt.show()

        return
```

In [4]:

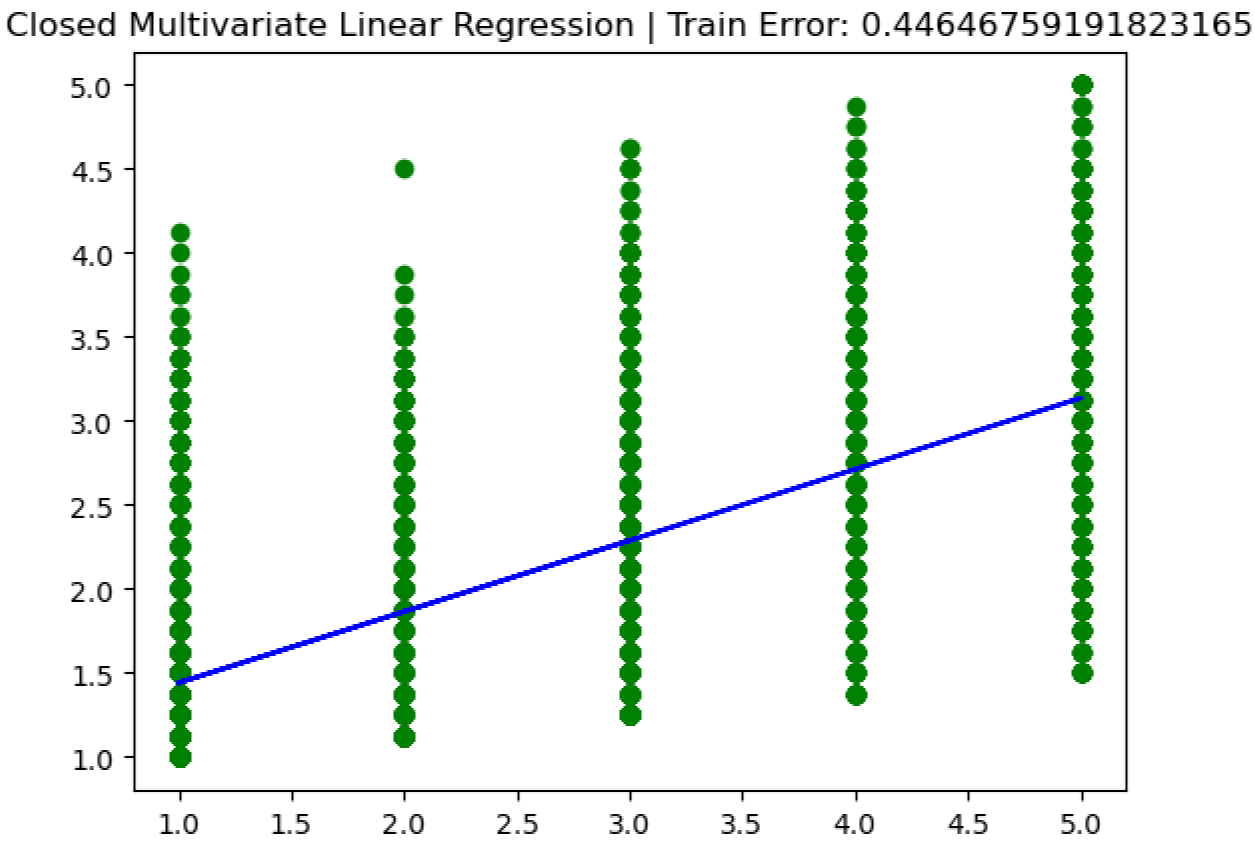
```
# Calculate R score for each row
r = np.mean(data, axis=1)

# Prepare datasets
train_set = data[:6500, 0]
train_labels = r[:6500]

test_set = data[6500:, 0]
test_labels = r[6500:]

# add row of 1s to make X suitable for dot pdt
train_set = np.stack((np.ones(shape=train_set.shape), train_set), axis=1)
test_set = np.stack((np.ones(shape=test_set.shape), test_set), axis=1)

print(train_set[:5], test_set[:5])
print(train_set.shape, test_set.shape)
```

```
[[1. 3.]
 [1. 1.]
 [1. 3.]
 [1. 3.]
 [1. 3.]] [[1. 3.]
 [1. 4.]
 [1. 1.]
 [1. 1.]
 [1. 1.]]
(6500, 2) (1978, 2)
```

In [5]:

```
## Task: R1 vs R regression model
```

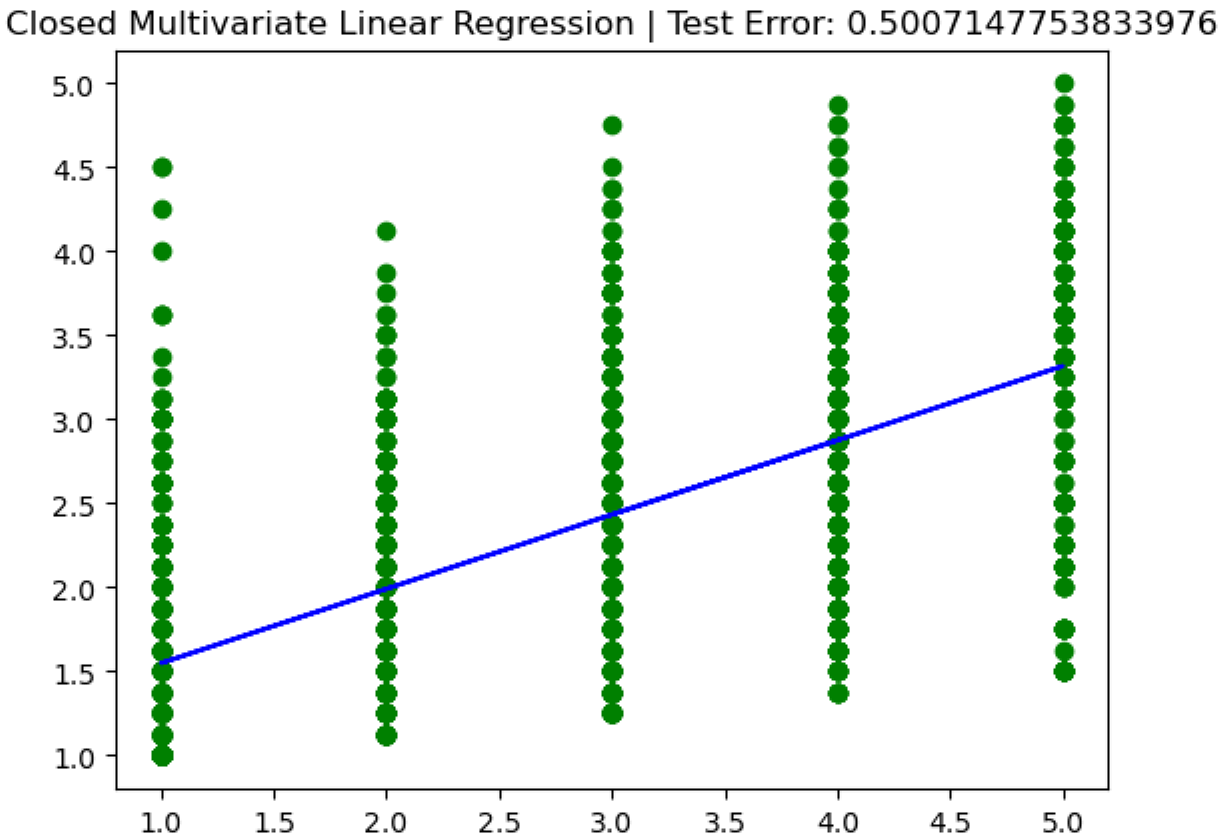
```
model = Regression(train_set, train_labels, test_set, test_labels)
model.fit()
```



(c) [6 points] (Validation) We now want to see if our previous model generalizes well. Therefore, using the regression model in part b, calculate the residual sum of squares for the remaining people and compare the residual sum of squares (averaged) with that (averaged as well) for part b.

In [6]:

```
model.test()
```



Comparing the averaged residual sum of squares (MSE) values for part b (train) and part c (test):

```
Train MSE: 0.4464
Test MSE: 0.50071
```

The overall error values are close, which indicate that the model most likely generalizes well to this distribution.