Randall Rogers
DSC 255: Machine Learning Fundamentals
Homework 2
Spring 2025

## Question 1

*Consider the two points*

$$x = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}, x' \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

(a) *What is the $L_2$ distance between them*

(b) *What is the $L_1$ distance between them*

(c) *What is the $L_\infty$ distance between them*

**Solution (a)**

---

The $L_2$ distance is defined as:

$$L_2 = \sqrt{\sum_{i=1}^{n} (x_i - x_i')^2}$$

Let, $n = 4$, $x_1 = -1$, $x_2 = 1$, $x_3 = -1$, $x_4 = 1$, $x_1' = 1$, $x_2' = 1$, $x_3' = 1$, $x_4' = 1$.

Use the $L_2$ equation and do the following:

- substitute $n$, expand the summation and subsitute values for $x_1$,..., $x_4'$

$L_2 = \sqrt{((-1-1)^2 + (1-1)^2 + (-1-1)^2 + (1-1)^2)}$

$L_2 = \sqrt{((-2)^2 + (0)^2 + (-2)^2 + (0)^2)}$

$L_2 = \sqrt{(4+4)}$

$L_2 = \sqrt{(8)}$

$\therefore$ the $L_2$ distance between $x$ and $x'$ is $\sqrt{8}$.

---

**Solution (b)**

---

The $L_1$ distance is defined as:

$$L_1 = \sum_{i=1}^{n} |x_i - x_i'|$$

Let, $n = 4$, $x_1 = -1$, $x_2 = 1$, $x_3 = -1$, $x_4 = 1$, $x'_1 = 1$, $x'_2 = 1$, $x'_3 = 1$, $x'_4 = 1$.

Use the $L_1$ equation and do the following:

- substitue $n$, expand the summation and substitute values for $x_1$,..., $x'_4$

$L_1 = |(-1-1)| + |(1-1)| + |(-1-1)| + |(1-1)|$

$L_1 = |(-2)| + |(0)| + |(-2)| + |(0)|$

$L_1 = 2 + 0 + 2 + 0$

$L_1 = 4$

$\therefore$ the $L_1$ distance between $x$ and $x'$ is 4.

---

**Solution (c)**

---

The $L_\infty$ distance is defined as:

$$L_\infty = \max_{i=1,2,\ldots,n} |x_i - x'_i|$$

Let, $n = 4$, $x_1 = -1$, $x_2 = 1$, $x_3 = -1$, $x_4 = 1$, $x'_1 = 1$, $x'_2 = 1$, $x'_3 = 1$, $x'_4 = 1$.

Use the $L_\infty$ equation and do the following:

- calculate the absolute differences for each component and find the maximum

$L_\infty = \max\{|(-1-1)|, |(1-1)|, |(-1-1)|, |(1-1)|\}$

$L_\infty = \max\{|(-2)|, |(0)|, |(-2)|, |(0)|\}$

$L_\infty = \max\{2, 0, 2, 0\}$

$L_\infty = 2$

$\therefore$ the $L_\infty$ distance between $x$ and $x'$ is 2.

---

---

## Question 2

*For the point* $\begin{bmatrix}1\\2\\3\end{bmatrix} \in \mathbb{R}^3$ *, compute the following*

(a) $\|x\|_1$

(b) $\|x\|_2$

(c) $\|x\|_\infty$

**Solution (a)**

---

The $L_1$ is defined as:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

Let, $n = 3$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$

Use the $L_1$ equation and do the following:

- substitute $n$, expand the summation and substitute values for $x_1$, $x_2$, $x_3$

$\|x\|_1 = \sum_{i=1}^{3} |x_i|$

$\|x\|_1 = |1| + |2| + |3|$

$\|x\|_1 = 6$

$\therefore$ the $L_1$ norm of the vector $x$ is 6.

---

**Solution (b)**

---

The $L_2$ is defined as:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Let, $n = 3$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$

Use the $L_2$ equation and do the following:

- substitute $n$, expand the summation and substitute values for $x_1$, $x_2$, $x_3$

$\|x\|_2 = \sqrt{\sum_{i=1}^{3} x_i^2}$

$\|x\|_2 = \sqrt{1^2 + 2^2 + 3^2}$

$\|x\|_2 = \sqrt{1 + 4 + 9}$

$\|x\|_2 = \sqrt{14}$

$\therefore$ the $L_2$ norm of the vector $x$ is $\sqrt{14}$.

---

**Solution (c)**

---

The $L_\infty$ is defined as:

$$\|x\|_\infty = \max_{i=1,2,\ldots,n} |x_i|$$

Let, $n = 3$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$

Use the $L_\infty$ equation and do the following:

- substitute $n$, find the maximum absolute value among $x_1$, $x_2$, $x_3$

$\|x\|_\infty = \max_{i=1,2,3} |x_i|$

$\|x\|_\infty = \max\{|1|, |2|, |3|\}$

$\|x\|_\infty = \max\{1, 2, 3\}$

$\|x\|_\infty = 3$

$\therefore$ the $L_\infty$ norm of the vector $x$ is 3.

---

---

## Question 3

*The following table specifies a distance on the space $\chi = \{A, B, C, D\}$. Is this a metric? Justify your answer.*

| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 2 | 1 | 5 |
| B | 2 | 0 | 4 | 3 |
| C | 1 | 4 | 0 | 2 |
| D | 5 | 3 | 2 | 0 |

Table 1: Table that specifies a distance function for $\chi$

**Solution**

---

To determine if the given distance function is a metric, it needs to satisfiy the properties of a meteric.

The four properties of a metric are:

1. **Non-negativity**: $d(x, y) \geq 0$ for all $x, y \in \chi$
2. **Identity of Indiscernibles**: $d(x, y) = 0$ if and only if $x = y$
3. **Symmetry**: $d(x, y) = d(y, x)$ for all $x, y \in \chi$
4. **Triangle Inequality**: $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in \chi$

Check the first property, Non-negativity.

$0, 2, 1, 5, 2, 0, 4, 3, 1, 4, 0, 2, 5, 3, 2, 0 \geq 0$

Hence, all values are all non-negative and the first property is satisfied.

Now,check the second property, Identity of Indiscernibles.

The diagonal elements are $(A, A)$, $(B, B)$, $(C, C)$, and $(D, D)$.

$$d(A, A) = 0, d(B, B) = 0, d(C, C) = 0, d(D, D) = 0$$

Hence, all diagonal elements are zero and the second property is satisfied.

Next,check the third property, Symmetry.

The symmetry elements are: $(A, B)$ and $(B, A)$; $(A, C)$ and $(C, A)$; $(A, D)$ and $(D, A)$; $(B, C)$ and $(C, B)$; $(B, D)$ and $(D, B)$; $(C, D)$ and $(D, C)$.

| $d(x, y)$ | $d(y, x)$ | Distance |
|---|---|---|
| $d(A, B)$ | $d(B, A)$ | 2 |
| $d(A, C)$ | $d(C, A)$ | 1 |
| $d(A, D)$ | $d(D, A)$ | 5 |
| $d(B, C)$ | $d(C, B)$ | 4 |
| $d(B, D)$ | $d(D, B)$ | 3 |
| $d(C, D)$ | $d(D, C)$ | 2 |

Table 2: Table that compares distance for $d(x, y)$ and $d(y, x)$ for $\chi$.

Hence, all symmetry elements are equal and the third property is satisfied

Lastly,check the fourth property, Triangle Inequality.

Check if $d(x, z) \leq d(x, y) + d(y, z)$ for all possible combinations of $x$, $y$, and $z$.

Let $x = A$, $y = B$ and $z = C$

Subsitute, $x$, $y$, $z$ into the triangle inequality and evaluate using *Table 2*:

$$d(A, C) \leq d(A, B) + d(B, C) \rightarrow 1 \leq 2 + 4 \rightarrow 1 \leq 6$$

Hence, the Triangle Inequality holds for $x = A$, $y = B$ and $z = C$

Let $x = A$, $y = C$ and $z = D$

Subsitute, $x$, $y$, $z$ into the triangle inequality and evaluate using *Table 2*:

$$d(A, D) \leq d(A, C) + d(C, D) \rightarrow 5 \leq 1 + 2 \rightarrow 5 \nleq 3$$

Hence, for $x = A$, $y = C$ and $z = D$ the Triangle Inequality is not satisfied.

$\therefore$ a distance on the space $\chi$ is not a metric

---

## Question 4

*The following vectors p and q specify probability distributions over a set of five out comes. What is the KL divergence between them, $K(p, q)$*

$$p = \left[\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16}\right], q = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right]$$

**Solution**

---

The Kullback-Leibler (KL) divergence is defined as:

$$K(p, q) = \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right)$$

Let $n = 5$,

$p_1 = \frac{1}{2}$, $p_2 = \frac{1}{4}$, $p_3 = \frac{1}{8}$, $p_4 = \frac{1}{16}$, $p_5 = \frac{1}{16}$

$q_1 = \frac{1}{4}$, $q_2 = \frac{1}{4}$, $q_3 = \frac{1}{6}$, $q_4 = \frac{1}{6}$, $q_5 = \frac{1}{6}$

- substitute $n$, expand the summation and subsitute values for $p_1$,..., $q_5$ in the KL divergence equation

$$K(p, q) = \sum_{i=1}^{5} p_i \log\left(\frac{p_i}{q_i}\right)$$

$K(p, q) = p_1 \log\left(\frac{p_1}{q_1}\right) + p_2 \log\left(\frac{p_2}{q_2}\right) + p_3 \log\left(\frac{p_3}{q_3}\right) + p_4 \log\left(\frac{p_4}{q_4}\right) + p_5 \log\left(\frac{p_5}{q_5}\right)$

$K(p, q) = \frac{1}{2} \log\left(\frac{\frac{1}{2}}{\frac{1}{4}}\right) + \frac{1}{4} \log\left(\frac{\frac{1}{4}}{\frac{1}{4}}\right) + \frac{1}{8} \log\left(\frac{\frac{1}{8}}{\frac{1}{6}}\right) + \frac{1}{16} \log\left(\frac{\frac{1}{16}}{\frac{1}{6}}\right) + \frac{1}{16} \log\left(\frac{\frac{1}{16}}{\frac{1}{6}}\right)$

$K(p, q) = \frac{1}{2} \log(2) + \frac{1}{4} \log(1) + \frac{1}{8} \log\left(\frac{3}{4}\right) + \frac{1}{16} \log\left(\frac{3}{8}\right) + \frac{1}{16} \log\left(\frac{3}{8}\right)$

$K(p, q) \approx 0.082$

$\therefore$ the KL divergence between p and q is approximately 0.082

---

---

## Question 5

*For each of the following prediction tasks, state whether it is best thought of as a classification problem or a regression problem.*

(a) *Based on sensors in a person's cell phone, predict whether they are walking, sitting, or running.*

(b) *Based on sensors in a moving car, predict the speed of the car directly in front.*

(c) *Based on a student's high-school SAT score, predict their GPA during freshman year of college.*

(d) *Based on a student's high-school SAT score, predict whether or not they will complete college.*

**Solution (a)**

---

We are attempting to predict a categorical variable (walking, sitting, or running).

$\therefore$ This is best thought as a classification problem.

---

**Solution (b)**

---

We are attempting to predict a continuous numberical variable (speed of a car).

$\therefore$ This is best thought as a regression problem.

**Solution (c)**

---

We are attempting to predict a continuous numberical variable (GPA).

$\therefore$ This is best thought as a regression problem.

---

**Solution (d)**

---

We are attempting to predict a categorical variable (pass or not pass).

$\therefore$ This is best thought as a classification problem.

---

## Question 6

*Variance examples. In each of the following cases, compute the variance.*

(a) *$X$ takes on values -1 and 1 with equal probability.*

(b) *$X$ always takes on same value.*

(c) *$X \in \{0, 1\}$ and $X$ is 1 with probability $\frac{1}{4}$.*

**Solution (a)**

---

The variance of a random variable $X$ is defined as:

$$\text{Var}(X) = E[(X - \mu)^2] = E[X^2] - (E[X])^2$$

- $E[X]$ is the expected value (mean) of $X$
- $\mu = E[X]$

Since the random variable $X$ takes on values -1 or 1 with equal probability, the probability is 0.50 or $\frac{1}{2}$.

Let $n = 2$, $x_1 = -1$, $x_2 = 1$, $P(X = x_1) = \frac{1}{2}$, $P(X = x_2) = \frac{1}{2}$.

Calculate the expected value $E[X]$.

$$E[X] = \sum_{i=1}^{n} x_i P(X = x_i)$$

Subsitute $n$, $x_1$, $x_2$, $P(X = x_1)$, $\frac{1}{2}$, $P(X = x_2)$ and expand summation.

$$E[X] = \sum_{i=1}^{2} x_i \cdot P(X = x_i) = (-1) \cdot \frac{1}{2} + (1) \cdot \frac{1}{2} = -\frac{1}{2} + \frac{1}{2} = 0$$

Calculate $E[X^2]$.

Subsitute $n$, $x_1$, $x_2$, $P(X = x_1)$, $\frac{1}{2}$, $P(X = x_2)$ and expand summation.

$$E[X^2] = \sum_{i=1}^{2} x_i^2 \cdot P(X = x_i) = (-1)^2 \cdot \frac{1}{2} + (1)^2 \cdot \frac{1}{2} = \frac{1}{2} + \frac{1}{2} = 1$$

Calculate the variance.

Subsitute $E[X^2]$ and $E[X]$ into the variance equation.

$$\text{Var}(X) = E[X^2] - (E[X])^2 = 1 - (0)^2 = 1 - 0 = 1$$

$\therefore$ the variance of $X$ is 1.

**Solution (b)**

The variance of a random variable $X$ is defined as:

$$\mathrm{Var}(X) = E[(X-\mu)^2] = E[X^2] - (E[X])^2$$

- $E[X]$ is the expected value (mean) of $X$
- $\mu = E[X]$

Since the random variable $X$ always takes on the same value, $\exists\, x \in \mathbb{R}$ such that $P(X = x) = 1$.
Calculate the expected value $E[X]$.

$$E[X] = \sum_{i=1}^{n} x_i P(X = x_i)$$

Subsitute $n = 1$, $x_1 = x$, $P(X = x_1) = 1$ and expand summation.

$$E[X] = \sum_{i=1}^{1} x_i \cdot P(X = x_i) = (x) \cdot 1 = x$$

Calculate $E[X^2]$.
Subsitute $n = 1$, $x_1 = x$, $P(X = x_1) = 1$ and expand summation.

$$E[X^2] = \sum_{i=1}^{1} x_i^2 \cdot P(X = x_i) = (x)^2 \cdot 1 = x^2$$

Calculate the variance.

Subsitute $E[X^2]$ and $E[X]$ into the variance equation.

$$\mathrm{Var}(X) = E[X^2] - (E[X])^2 = x^2 - x^2 = 0$$

$\therefore$ the variance of $X$ is 0.

**Solution (c)**

The variance of a random variable $X$ is defined as:

$$\text{Var}(X) = E[(X-\mu)^2] = E[X^2] - (E[X])^2$$

- $E[X]$ is the expected value (mean) of $X$
- $\mu = E[X]$

Let $n = 2$, $x_1 = 1$, $x_2 = 0$, $P(X = x_1) = \frac{1}{4}$, $P(X = x_2) = \frac{3}{4}$.

Calculate the expected value $E[X]$.

$$E[X] = \sum_{i=1}^{n} x_i P(X = x_i)$$

Subsitute $n$, $x_1$, $x_2$, $P(X = x_1)$, $P(X = x_2)$ and expand summation.

$$E[X] = \sum_{i=1}^{2} x_i \cdot P(X = x_i) = (1) \cdot \frac{1}{4} + (0) \cdot \frac{3}{4} = \frac{1}{4}$$

Calculate $E[X^2]$.

Subsitute $n$, $x_1$, $x_2$, $P(X = x_1)$, $P(X = x_2)$ and expand summation.

$$E[X^2] = \sum_{i=1}^{2} x_i^2 \cdot P(X = x_i) = (1)^2 \cdot \frac{1}{4} + (0)^2 \cdot \frac{3}{4} = \frac{1}{4}$$

Calculate the variance.

Subsitute $E[X^2]$ and $E[X]$ into the variance equation.

$$\text{Var}(X) = E[X^2] - (E[X])^2 = \frac{1}{4} - (\frac{1}{4})^2 = \frac{1}{4} - \frac{1}{16} = \frac{3}{16}$$

$\therefore$ the variance of $X$ is $\frac{3}{16}$.

## Question 7

*Independence and uncorrelatedness. Random variables X; Y take on values in the range* $\{-1, 0, 1\}$ *and have the following joint distribution.*

| $(X \downarrow, Y \rightarrow)$ | -1 | 0 | 1 |
|---|---|---|---|
| -1 | 0 | 0 | 1/3 |
| 0 | 0 | 1/3 | 0 |
| 1 | 1/3 | 0 | 0 |

Table 3: Joint distribution for random variables X and Y.

(a) *What is the covariance between X and Y ?*

(b) *What is the correlation between X and Y ?*

**Solution (a)**

---

Covariance for random variables $A$ and $B$ is defined as:

$$Cov(A, B) = E[AB] - E[A]E[B]$$

The expected value of a random variable $A$ is defined as:

$$E[A] = \sum_{i=1}^{n} a_i P(A = a)$$

The expected value of $AB$ where A and B are random variables is defined as:

$$E[AB] = \sum_{a \in P(A)} \sum_{b \in P(B)} ab \cdot P(A = a, B = b)$$

Calculate the expected value $E[X]$.

Let $x_1 = -1$, $x_2 = 0$, $x_3 = 1$, and $P(X = -1) = P(X = 0) = P(X = 1) = \frac{1}{3}$.

$E[X] = (-1) \cdot P(X = -1) + (0) \cdot P(X = 0) + (1) \cdot P(X = 1)$

$E[X] = (-1) \cdot \frac{1}{3} + (0) \cdot \frac{1}{3} + (1) \cdot \frac{1}{3}$

$E[X] = -\frac{1}{3} + 0 + \frac{1}{3}$

$E[X] = 0$

Calculate the expected value $E[Y]$.

Let $y_1 = -1$, $y_2 = 0$, $y_3 = 1$, and $P(Y = -1) = P(Y = 0) = P(Y = 1) = \frac{1}{3}$.

$E[Y] = (-1) \cdot P(Y = -1) + (0) \cdot P(Y = 0) + (1) \cdot P(Y = 1)$

$E[Y] = (-1) \cdot \frac{1}{3} + (0) \cdot \frac{1}{3} + (1) \cdot \frac{1}{3}$

$E[Y] = -\frac{1}{3} + 0 + \frac{1}{3}$

$E[Y] = 0$

Calculate the expected value $E[XY]$.

- Since we have a diagonal matrix our $E[XY]$ equation becomes the following:

$E[XY] = x_1y_1 \cdot P(X=-1, Y=1) + x_2y_2 \cdot P(X=0, Y=0) + x_3y_3 \cdot P(X=1, Y=1)$

$E[XY] = (-1)(1) \cdot \frac{1}{3} + (0)(0) \cdot \frac{1}{3} + (1)(-1) \cdot \frac{1}{3}$

$E[XY] = -\frac{1}{3} + 0 - \frac{1}{3}$

$E[XY] = -\frac{2}{3}$

Calculate the covariance.

Subsitute $E[XY]$, $E[X]$ and $E[Y]$ into the covariance equation.

$$Cov(X,Y) = E[XY] - E[X]E[Y] = -\frac{2}{3} - (0)(0) = -\frac{2}{3}$$

$\therefore$ the covariance between $X$ and $Y$ is $-\frac{2}{3}$.

---

**Solution (b)**

---

**Solution (b)**

---

Correlation for random variables $A$ and $B$ is defined as:

$$Corr(A,B) = \frac{Cov(A,B)}{\sqrt{Var(A) \cdot Var(B)}}$$

The variance of a random variable $A$ is defined as:

$$Var(A) = E[A^2] - (E[A])^2$$

From part (a), we know that $Cov(X,Y) = -\frac{2}{3}$ and $E[X] = E[Y] = 0$.

Calculate the variance of $X$.

First, calculate $E[X^2]$.

$E[X^2] = (-1)^2 \cdot P(X=-1) + (0)^2 \cdot P(X=0) + (1)^2 \cdot P(X=1)$

$E[X^2] = (1) \cdot \frac{1}{3} + (0) \cdot \frac{1}{3} + (1) \cdot \frac{1}{3}$

$E[X^2] = \frac{1}{3} + 0 + \frac{1}{3}$

$E[X^2] = \frac{2}{3}$

Now, calculate $Var(X)$.

$Var(X) = E[X^2] - (E[X])^2 = \frac{2}{3} - (0)^2 = \frac{2}{3}$

Calculate the variance of $Y$.

First, calculate $E[Y^2]$.

$E[Y^2] = (-1)^2 \cdot P(Y = -1) + (0)^2 \cdot P(Y = 0) + (1)^2 \cdot P(Y = 1)$

$E[Y^2] = (1) \cdot \frac{1}{3} + (0) \cdot \frac{1}{3} + (1) \cdot \frac{1}{3}$

$E[Y^2] = \frac{1}{3} + 0 + \frac{1}{3}$

$E[Y^2] = \frac{2}{3}$

Now, calculate $Var(Y)$.

$Var(Y) = E[Y^2] - (E[Y])^2 = \frac{2}{3} - (0)^2 = \frac{2}{3}$

Calculate the correlation.

Substitute $Cov(X, Y)$, $Var(X)$, and $Var(Y)$ into the correlation equation.

$Corr(X, Y) = \frac{Cov(X,Y)}{\sqrt{Var(X) \cdot Var(Y)}} = \frac{-\frac{2}{3}}{\sqrt{\frac{2}{3} \cdot \frac{2}{3}}} = \frac{-\frac{2}{3}}{\sqrt{\frac{4}{9}}} = \frac{-\frac{2}{3}}{\frac{2}{3}} = -1$

$\therefore$ the correlation between $X$ and $Y$ is $-1$.

---

---

## Question 8

*Independence and uncorrelatedness. Random variables X; Y take on values in the range* $\{-1, 0, 1\}$ *and have the following joint distribution.*

| $(X \downarrow, Y \rightarrow)$ | -1 | 0 | 1 |
|---|---|---|---|
| -1 | 1/6 | 0 | 1/6 |
| 0 | 0 | 1/3 | 0 |
| 1 | 1/6 | 0 | 1/6 |

Table 4: Joint distribution for random variables X and Y.

(a) *Are X and Y independent?*

(b) *Are X and Y uncorrelated?*

**Solution (a)**

---

Two random variables $A$ and $B$ are independent if:

$$P(A, B) = P(A) \cdot P(B)$$

Calculate probabilities $P(X = x)$ from *Table 4*.

$P(X = -1) = P(X = -1, Y = -1) + P(X = -1, Y = 0) + P(X = -1, Y = 1)$

$P(X = -1) = \frac{1}{6} + 0 + \frac{1}{6}$

$P(X = -1) = \frac{1}{3}$

$P(X = 0) = P(X = 0, Y = -1) + P(X = 0, Y = 0) + P(X = 0, Y = 1)$

$P(X = 0) = 0 + \frac{1}{3} + 0$

$P(X = 0) = \frac{1}{3}$

$P(X = 1) = P(X = 1, Y = -1) + P(X = 1, Y = 0) + P(X = 1, Y = 1)$

$P(X = 1) = \frac{1}{6} + 0 + \frac{1}{6}$

$P(X = 1) = \frac{1}{3}$

Calculate probabilities $P(Y = y)$ from *Table 4*.

$P(Y = -1) = P(X = -1, Y = -1) + P(X = 0, Y = -1) + P(X = 1, Y = -1)$

$P(Y = -1) = \frac{1}{6} + 0 + \frac{1}{6}$

$P(Y = -1) = \frac{1}{3}$

$P(Y = 0) = P(X = -1, Y = 0) + P(X = 0, Y = 0) + P(X = 1, Y = 0)$

$P(Y = 0) = 0 + \frac{1}{3} + 0$

$P(Y = 0) = \frac{1}{3}$

$P(Y = 1) = P(X = -1, Y = 1) + P(X = 0, Y = 1) + P(X = 1, Y = 1)$

$P(Y = 1) = \frac{1}{6} + 0 + \frac{1}{6} = \frac{1}{3}$

$P(Y = 1) = \frac{1}{3}$

Check if $P(X, Y) = P(X) \cdot P(Y)$ for all combinations:

For $(X = -1, Y = -1)$:

$P(X = -1, Y = -1) = \frac{1}{6}$

$P(X = -1) \cdot P(Y = -1) = \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}$

$$\frac{1}{6} \neq \frac{1}{9} \rightarrow P(X = -1, Y = -1) \neq P(X = -1) \cdot P(Y = -1)$$

$\therefore$ $X$ and $Y$ are not independent.

---

**Solution (b)**

---

Two random variables $A$ and $B$ are uncorrelated if and only if their covariance is zero:

$$\text{Cov}(A, B) = 0$$

The covariance between $A$ and $B$ is defined as:

$$\text{Cov}(A, B) = E[AB] - E[A]E[B]$$

Calculate the expected value $E[X]$.

$E[X] = \sum_{x \in P(X)} x \cdot P(X = x)$

$E[X] = (-1) \cdot \frac{1}{3} + (0) \cdot \frac{1}{3} + (1) \cdot \frac{1}{3}$

$E[X] = -\frac{1}{3} + 0 + \frac{1}{3}$

$E[X] = 0$

Calculate the expected value $E[Y]$.

$E[Y] = \sum_{y \in P(Y)} y \cdot P(Y = y)$

$E[Y] = (-1) \cdot \frac{1}{3} + (0) \cdot \frac{1}{3} + (1) \cdot \frac{1}{3}$

$E[Y] = -\frac{1}{3} + 0 + \frac{1}{3}$

$E[Y] = 0$

Calculate the expected value $E[XY]$.

$E[XY] = \sum_{x \in P(X)} \sum_{y \in P(Y)} xy \cdot P(X = x, Y = y)$

$E[XY] = (-1)(-1) \cdot \frac{1}{6} + (-1)(0) \cdot 0 + (-1)(1) \cdot \frac{1}{6} + (0)(-1) \cdot 0 + (0)(0) \cdot \frac{1}{3} + (0)(1) \cdot 0 + (1)(-1) \cdot \frac{1}{6} + (1)(0) \cdot 0 + (1)(1) \cdot \frac{1}{6}$

$E[XY] = \frac{1}{6} + 0 + (-\frac{1}{6}) + 0 + 0 + 0 + (-\frac{1}{6}) + 0 + \frac{1}{6}$

$E[XY] = \frac{1}{6} - \frac{1}{6} - \frac{1}{6} + \frac{1}{6}$

$E[XY] = 0$

Calculate the covariance.

$\text{Cov}(X, Y) = E[XY] - E[X]E[Y] = 0 - (0)(0) = 0$

Since $\text{Cov}(X, Y) = 0$, the variables $X$ and $Y$ are uncorrelated.

$\therefore$ $X$ and $Y$ are uncorrelated.

---

---

## Programming Exercises

*Before starting on this section, download the archive* ***hw2.zip*** *from the course website.*

- *In these problems, you are asked to perform nearest neighbor classification with different distance functions and to calculate error using a hold-out set or by cross-validation. The code for these tasks is neither lengthy nor complex, so you should be able to write up your own routines. Alternatively, you may invoke modules from* ***sklearn****.*

## Question 9

*Classifying back injuries. In this problem, you will use nearest neighbor to classify patients' back injuries based on measurements of the shape and orientation of their pelvis and spine. The data set spine-data.txt contains information from 310 patients. For each patient, there are: six numeric features (the x) and a label (the y): 'NO' (normal), 'DH' (herniated disk), or 'SL' (spondilolysthesis). We will divide this data into a training set with 250 points and a separate test set of 60 points.*

- *Make sure you have the data set spine-data.txt. You can load it into Python using the following.*
  import numpy as np
  # Load data set and code labels as 0 = 'NO', 1 = 'DH', 2 = 'SL'
  labels = [b'NO', b'DH', b'SL']
  data = np.loadtxt('spine-data.txt', converters={6: lambda s: labels.index(s)})

- *Split the data into a training set, consisting of the first 250 points, and a test set, consisting of the remaining 60 points.*

- *Code up a nearest neighbor classifier based on this training set. Try both $\ell_2$ and $\ell_1$ distance. Recall $x, x' \in \mathbb{R}^d$*

$$\|x - x'\|_2 = \sqrt{\sum_{i=1}^{d} (x_i - x'_i)^2}$$

$$\|x - x'\|_1 = \sum_{i=1}^{d} |x_i - x'_i|$$

*Now do the following exercises, to be turned in.*

(a) *What error rates do you get on the test set for each of the two distance functions?*

(b) *For each of the two distance functions, give the confusion matrix of the NN classifier. This is a $3 \times 3$ table of the form:*

| | NO | DH | SL |
|---|---|---|---|
| NO | | | |
| DH | | | |
| SL | | | |

Table 5: he entry at row DH, column SL, for instance, contains the number of test points whose correct label was DH and got classified as SL.

**Solution (a)**

Python code for nearest neighbor classifier using $\ell_1$, $\ell_2$, and error rates:

```python
## import libraries
import numpy as np

## initialize labels for data
data_labels = ['NO','DH','SL']

## load data from spine-data.txt and relabel the 7th column (NO:1, DH:1, SL:2)
data = np.loadtxt('spine-data.txt', converters={6: lambda s: data_labels.index(s)})

## split data into features and labels
features = data[:, :-1]
labels = data[:, -1]

## split data into training and test sets
features_train = features[:250]
features_test = features[250:]
labels_train = labels[:250]
labels_test = labels[250:]

## define l1 distance function
def l1_distance(x, x_prime):
    l1_dist = np.sum(np.abs(x - x_prime))
    return l1_dist

## define l2 distance function
def l2_distance(x, x_prime):
    l2_dist = np.sqrt(np.sum((x - x_prime) ** 2))
    return l2_dist

## define nn_classifier function
def nn_classifier(features_train, features_test, labels_train,distance_function):
    """
    1. loop through each test point in the features_test list
    2. get the distance between the test point and each training point for every
       training point
    3. find the index of the training point with the smallest distance
    4. use the index to get the label of the training point
    5. append the label to the predictions list
    6. return the predictions list
    """
    predictions = []
    for test_point in features_test:
        distances = [distance_function(test_point, train_point) for train_point in
            features_train]
        nearest_neighbor_index = np.argmin(distances)
        predictions.append(labels_train[nearest_neighbor_index])
    predictions = np.array(predictions)
    return predictions

## define error rate function
def get_error_rate(predictions, labels):
    correct = sum(p == l for p, l in zip(predictions, labels))
    accuracy = correct / len(labels)
    er = 1 - accuracy
    return er

## evaluate l1 distance classifier and get error rate
predictions_l1 = nn_classifier(features_train, features_test, labels_train, l1_distance)
error_rate_l1 = get_error_rate(predictions_l1, labels_test)

## evaluate l2 distance classifier and get error rate
predictions_l2 = nn_classifier(features_train, features_test, labels_train, l2_distance)
error_rate_l2 = get_error_rate(predictions_l2, labels_test)
```

$\therefore$ the error rates for the two distance functions are: $ER_{\ell_1} = 0.183$ and $ER_{\ell_2} = 0.117$

## Solution (b)

Python code for confusion matrix for nearest neighbor using $\ell_1$, $\ell_2$, and confusion matrix visualization:

Note: Utilize functions and variables defined in **Solution (a)**

```python
## define confusion matrix function
def get_confusion_matrix(predictions, labels):
    cm = confusion_matrix(predictions,labels)
    return cm

## visualize confusion matrix function
def plot_confusion_matrix(l1_er, cm1, l2_er, cm2, labels):
    fig, ax = plt.subplots(1,2,figsize=(10, 8),sharey=True)
    ax = ax.flatten()
    sns.heatmap(cm1, annot=True, fmt='d', cmap=sns.color_palette("rocket",
        as_cmap=True), xticklabels=labels, yticklabels=labels, ax=ax[0])
    ax[0].set_title(f'L1 Distance Confusion Matrix\nError Rate: {l1_er:.3f}')
    sns.heatmap(cm2, annot=True, fmt='d', cmap=sns.color_palette("rocket",
        as_cmap=True), xticklabels=labels, yticklabels=labels, ax=ax[1])
    ax[1].set_title(f'L2 Distance Confusion Matrix\nError Rate: {l2_er:.3f}')
    ax[0].set_xlabel('Predicted')
    ax[1].set_xlabel('Predicted')
    ax[0].set_ylabel('True')
    plt.tight_layout()
    plt.show()

## get confusion matrix for l1 and l2 distance classifier
cm_l1 = get_confusion_matrix(predictions_l1, labels_test)
cm_l2 = get_confusion_matrix(predictions_l2, labels_test)

## visualize confusion matrix for l1 and l2 distance classifier
plot_confusion_matrix(error_rate_l1, cm_l1, error_rate_l2, cm_l2, data_labels)
```

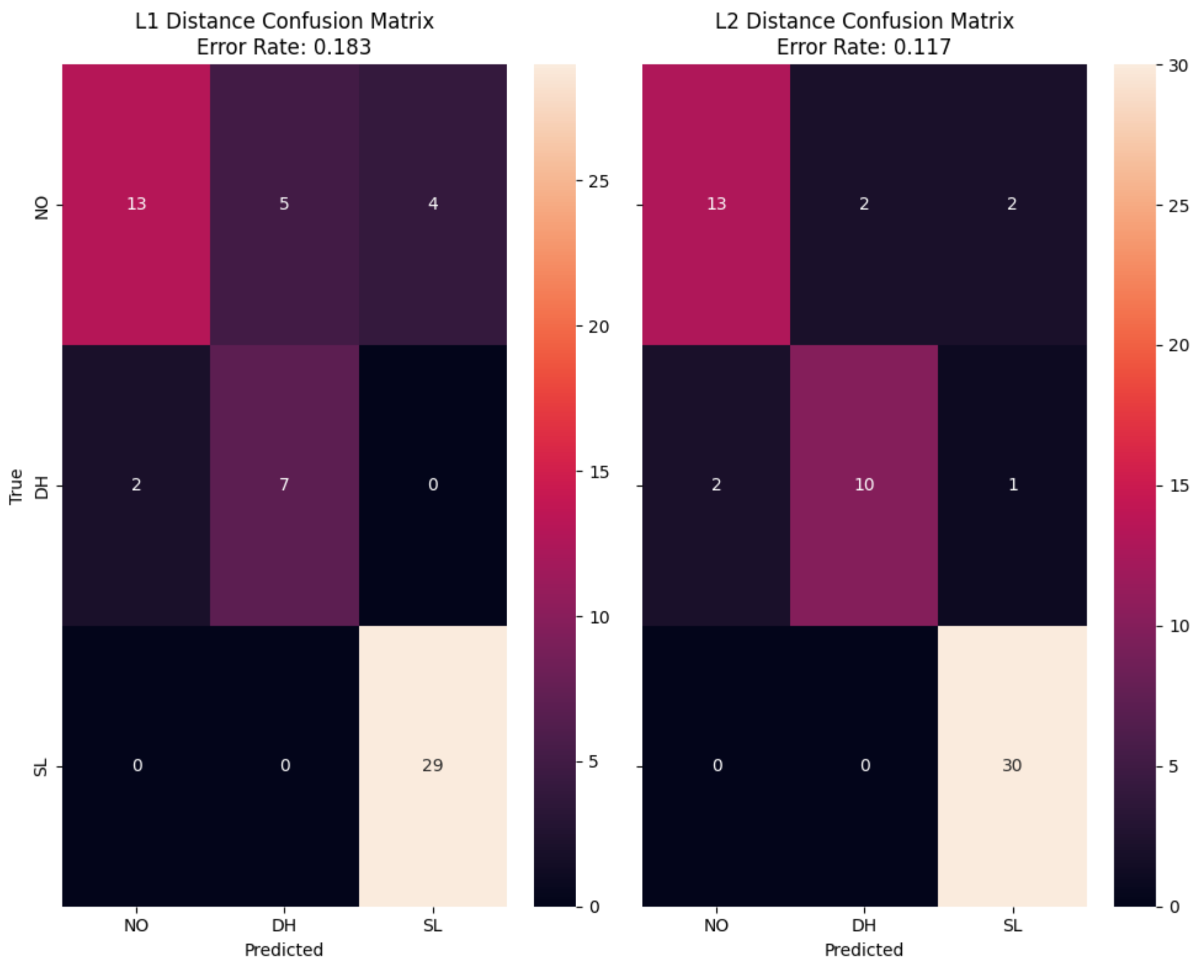


Figure 1: Confusion matrix for $\ell_1$ and $\ell_2$ nearest neighbor classifiers.

## Question 10

*Cross-validation for nearest neighbor classification. The wine.data data set is described in detail at:*

***https://archive.ics.uci.edu/ml/datasets/wine***

*This small data set has 178 observations. Each data point $x$ consists of 13 features that capture visual and chemical properties of a bottle of wine. The label $y \in \{1, 2, 3\}$ indicates which of three wineries the bottle came from. The goal is to use the data to learn a classifier that can predict $y$ from $x$.*

*Suppose we use the entire data set of 178 points as the training set for 1-NN classification with Euclidean distance. We would like to estimate the quality of this classifier.*

(a) *Use leave-one-out cross-validation (LOOCV) to estimate the accuracy of the classifier and also to estimate the $3 \times 3$ confusion matrix.*

(b) *Estimate the accuracy of the 1-NN classifier using k-fold cross-validation using 20 different choices of k that are fairly well spread out across the range 2 to 100. Plot these estimates: put k on the horizontal axis and accuracy estimate on the vertical axis.*

(c) *The various features in this data set have different ranges.Perhaps it would be better to normalize them so as to equalize their contributions to the distance function. There are many ways to do this; one option is to linearly rescale each coordinate so that the values lie in $[0, 1]$ (i.e. the minimum value on that coordinate maps to 0 and the maximum value maps to 1). Do this, and then re-estimate the accuracy and confusion matrix using LOOCV. Did the normalization helpperformance?*

### Solution (a)

---

Python code for reading *wine.DATA*, and splitting data

```python
## import libraries
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import sklearn

## read wine.DATA file
df = pd.read_csv('wine.data', header=None)

## name columns
df.columns = ['Class', 'Alcohol', 'Malic Acid', 'Ash', 'Alcalinity of Ash',
    'Magnesium', 'Total Phenols', 'Flavanoids', 'Nonflavanoid Phenols',
    'Proanthocyanins', 'Color Intensity', 'Hue', 'OD280/OD315 of Diluted Wines',
    'Proline']

## define wine labels
wine_labels = ['Class 1', 'Class 2', 'Class 3']

## split data into features and labels
features = df.iloc[:, 1:].values
labels = df.iloc[:, 0].values
```

Python code for estimating accuracy and confusion matrix for classifier

```python

## initalize 1-NN classifier
NN_1 = sklearn.neighbors.KNeighborsClassifier(n_neighbors=1, algorithm='brute',
    metric='euclidean')

## initalize LOOCV
```

```python
loocv = sklearn.model_selection.LeaveOneOut()

## get predictions from cross-validation
predictions = sklearn.model_selection.cross_val_predict(NN_1, features, labels,
    cv=loocv)

## estimate accuracy
accuracy = sklearn.metrics.accuracy_score(labels, predictions)
print(f'Accuracy: {accuracy:.3f}')
## get confusion matrix
cm = sklearn.metrics.confusion_matrix(labels, predictions)

## visualize confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap=sns.color_palette("rocket", as_cmap=True),
    xticklabels=wine_labels, yticklabels=wine_labels)
plt.title(f'Wine Classification Confusion Matrix\n Accuracy: {accuracy:.3f}')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.tight_layout()
plt.show()
```

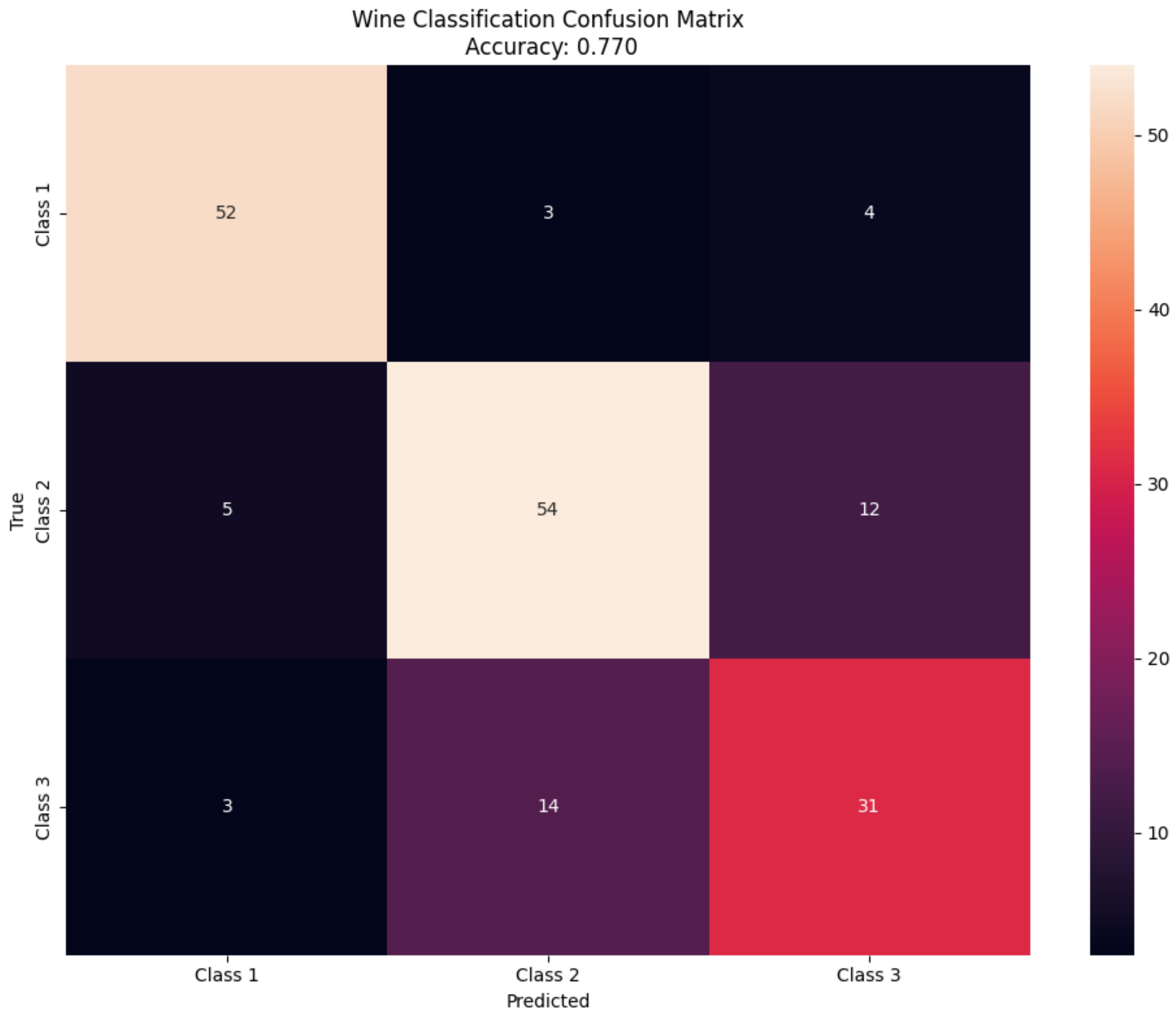


Figure 2: Estimate of confusion matrix using leave-one-out cross-validation (LOOCV) and 1-NN classification with Euclidean distance.

**Solution (b)**

Python code for estimating accuracy using k-fold cross-validation and visualization of the estimates

```python
## initialize k values such that we have  20 k's spread out across the range 2 to 100
k_values = np.linspace(2, 100, 20, dtype=int)

## initialize accuracy list
accuracies = []

## perform k-fold cross-validation for each k
for k in k_values:
    cv = sklearn.model_selection.KFold(n_splits=k, shuffle=True, random_state=42)
    scores = sklearn.model_selection.cross_val_score(NN_1, features, labels, cv=cv)
    accuracies.append(np.mean(scores))

## plot results
plt.figure(figsize=(10, 6))
plt.plot(k_values, accuracies, marker='o', linestyle='-')
plt.xlabel('Number of Folds (k)')
plt.ylabel('Accuracy Estimate')
plt.title('Accuracy Estimates for Different Values of k in k-Fold Cross-Validation')
plt.grid(True)
plt.show()
```

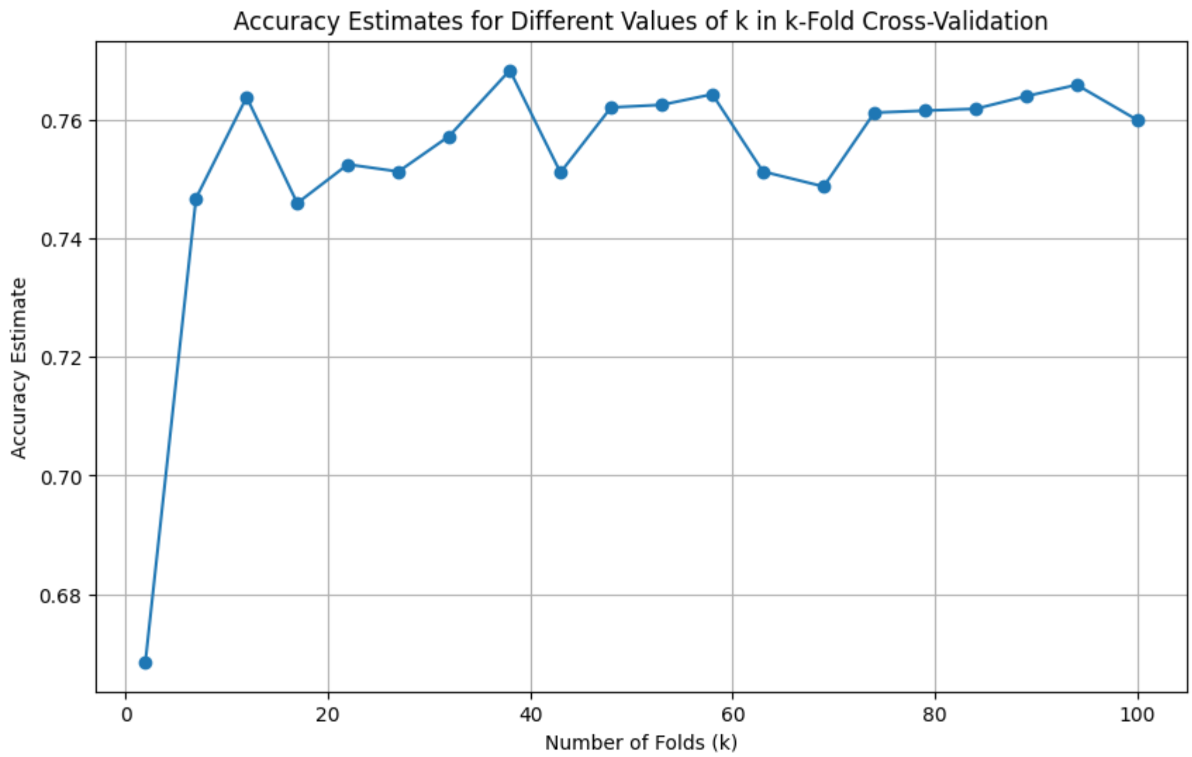


Figure 3: Estimates of k-fold cross validation accuracies.

**Solution (c)**

Python code for normalizing data and estimating accuracy and confusion matrix using LOOCV

```python
## normalize the features
norm_features = sklearn.preprocessing.normalize(features,norm='max')

## initalize 1-NN classifier
NN_1_n = sklearn.neighbors.KNeighborsClassifier(n_neighbors=1, algorithm='brute', metric='euclidean')

## initalize LOOCV
loocv_n = sklearn.model_selection.LeaveOneOut()

## get predictions from cross-validation
predictions_n = sklearn.model_selection.cross_val_predict(NN_1_n, norm_features, labels, cv=loocv_n)

## estimate accuracy
accuracy_n = sklearn.metrics.accuracy_score(labels, predictions_n)
print(f'Accuracy: {accuracy:.3f}')

## get confusion matrix
cm_n = sklearn.metrics.confusion_matrix(labels, predictions_n)

## compare confusion matrices
def compare_confusion_matrix(acc, cm, acc_norm, cm_n, labels):
    fig, ax = plt.subplots(1,2,figsize=(10, 8),sharey=True)
    ax = ax.flatten()
    sns.heatmap(cm, annot=True, fmt='d', cmap=sns.color_palette("rocket", as_cmap=True), xticklabels=labels, yticklabels=labels, ax=ax[0])
    ax[0].set_title(f'Part (a) Confusion Matrix\nAccuracy: {acc:.3f}')
    sns.heatmap(cm_n, annot=True, fmt='d', cmap=sns.color_palette("rocket", as_cmap=True), xticklabels=labels, yticklabels=labels, ax=ax[1])
    ax[1].set_title(f'Confusion Matrix for Normalized Features\nAccuracy: {acc_norm:.3f}')
    ax[0].set_xlabel('Predicted')
    ax[1].set_xlabel('Predicted')
    ax[0].set_ylabel('True')
    plt.tight_layout()
    plt.show()

  compare_confusion_matrix(accuracy, cm, accuracy_n, cm_n, wine_labels)
```

∴ from *Figure 4* normalizing data improved the performance of the classifier.

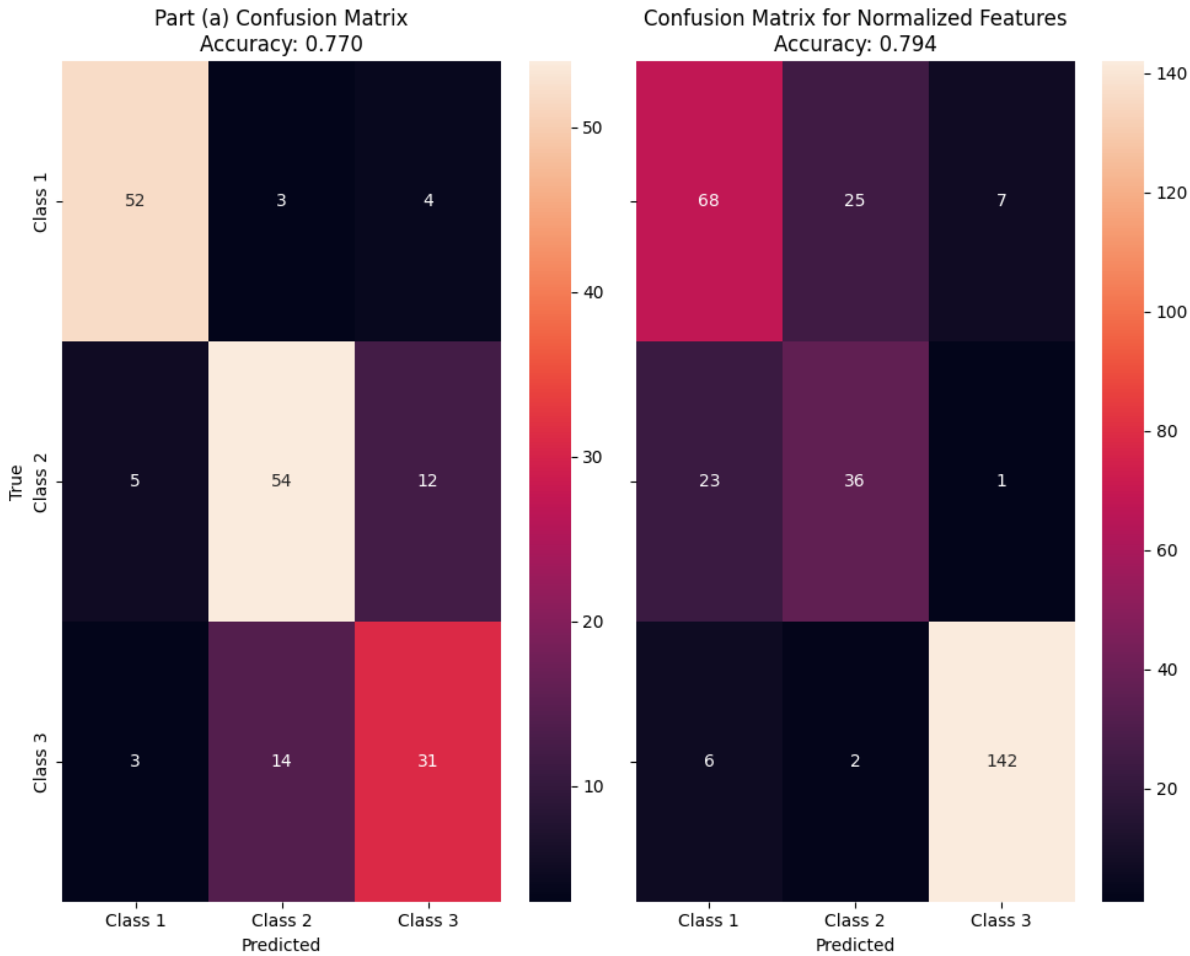


Figure 4: Estimate of confusion matrix for orginal data and normalized data using leave-one-out cross-validation (LOOCV) and 1-NN classification with Euclidean distance.