

# Intro to Machine Learning Assignment

Ibrahim Mohamed

Student ID: 7790

October 12, 2024

## **Note**

This assignment was written using LaTeX. The source code for the document and the code used in this assignment can be found in the following GitHub link: <https://github.com/H1MAA/Intro-To-Machine-Assignments>

# 1 Linear Algebra

**Question 1:** Find the eigenvalues of the following matrix  $A$ :

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 4 & -17 & 8 \end{bmatrix}$$

Eigenvalues:  $\lambda_1, \lambda_2, \lambda_3$ , can be found by solving the following equation:

$$\det(A - \lambda I) = 0$$

This gives us the following:

$$\det \left( \begin{bmatrix} -\lambda & 1 & 0 \\ 0 & -\lambda & 1 \\ 4 & -17 & 8 - \lambda \end{bmatrix} \right)$$

Selecting the first row and using cofactor expansion and selecting the first column, we get:

$$\det(A - \lambda I) = -\lambda \cdot \det \begin{bmatrix} -\lambda & 1 \\ -17 & 8 - \lambda \end{bmatrix} - 1 \cdot \det \begin{bmatrix} 0 & 1 \\ 4 & 8 - \lambda \end{bmatrix} + 0 \cdot \det \begin{bmatrix} 0 & -\lambda \\ 4 & -17 \end{bmatrix}$$

First term of the equation is:

$$-\lambda \cdot \det \begin{bmatrix} -\lambda & 1 \\ -17 & 8 - \lambda \end{bmatrix} = -\lambda \cdot (-\lambda \cdot (8 - \lambda) - 1 \cdot -17)$$

Simplifying the above equation, we get:

$$-\lambda \cdot (-\lambda \cdot (8 - \lambda) - 1 \cdot -17) = \lambda^3 + 8\lambda^2 - 17\lambda$$

Second term of the equation is:

$$-1 \cdot \det \begin{bmatrix} 0 & 1 \\ 4 & 8 - \lambda \end{bmatrix} = -1 \cdot (0 \cdot (8 - \lambda) - 1 \cdot 4)$$

Simplifying the above equation, we get:

$$-1 \cdot (0 \cdot (8 - \lambda) - 1 \cdot 4) = 4$$

The third term of the equation is 0, so we can ignore it. Therefore, the equation becomes:

$$-\lambda^3 + 8\lambda^2 - 17\lambda + 4 = 0$$

Factorize:

$$(\lambda - 4)(\lambda^2 - 4\lambda + 1) = 0$$

Therefore, the eigenvalues are:

$$\lambda_1 = 4, \quad \lambda_{2,3} = 2 \pm \sqrt{3}$$

## 2 Probability

**Question 2:** Three persons A, B, and C have applied for a job in a private company. The chance of their selections is in the ratio 1:2:4. The probabilities that A, B, and C can introduce changes to improve the profits of the company are 0.8, 0.5, and 0.3, respectively. If the change does not take place, find the probability that it is due to the appointment of C.

**Given:**

$$P(A) = \frac{1}{7}, \quad P(B) = \frac{2}{7}, \quad P(C) = \frac{4}{7}$$
$$P(\text{Change} \mid A) = 0.8, \quad P(\text{Change} \mid B) = 0.5, \quad P(\text{Change} \mid C) = 0.3$$

**Find:**

**P(C | No Change)**

From the question we can deduce that:

$$P(\text{Change}) = P(A) \cdot P(\text{Change} \mid A) + P(B) \cdot P(\text{Change} \mid B) + P(C) \cdot P(\text{Change} \mid C)$$

$$P(\text{Change}) = \frac{1}{7} \cdot 0.8 + \frac{2}{7} \cdot 0.5 + \frac{4}{7} \cdot 0.3 = \frac{3}{7}$$

Therefore,

$$P(\text{No Change}) = 1 - P(\text{Change}) = \frac{4}{7}$$

And

$$P(\text{No Change} \mid C) = 1 - P(\text{Change} \mid C) = 0.7$$

We can get the probability of no change due to the appointment of C:

$$P(C \mid \text{No Change}) = \frac{P(C) \cdot P(\text{No Change} \mid C)}{P(\text{No Change})}$$

$$P(C \mid \text{No Change}) = \frac{\frac{4}{7} \cdot 0.7}{\frac{4}{7}} = 0.7$$

### 3 K-Nearest Neighbor (KNN)

**Question 3a:** Knowing that passed exam is the class label, predict the class of the following students using K-Nearest Neighbor for  $K = 3$ .

**Solution:** We Can use the Euclidean distance to find the nearest neighbors. The Euclidean distance between two points  $(x_1, y_1)$  and  $(x_2, y_2)$  is given by:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Using python we can implement the Euclidean distance and the K-Nearest Neighbor algorithm as follows:

```
import numpy as np
import pandas as pd

def getEuclideanDistance(a, b):
    return np.linalg.norm(a - b)

def getNearestNeighbors(train, test, k):
    distances = []
    for i in range(len(train)):
        distances.append((train[i], getEuclideanDistance(train[i], test)))
    distances.sort(key=lambda x: x[1])
    return distances[:k]
```

Figure 1: Euclidean Distance and Nearest Neighbors Functions

```
df = pd.read_csv('Exam Data.csv')
df.drop('ID', axis=1, inplace=True)
df.head()
```

	Programming Languages	Delivered Assignments	GPA	Passed Exam
0	3	5	3.1	1.0
1	3	5	2.0	0.0
2	2	7	3.5	1.0
3	4	9	2.5	1.0
4	3	11	3.9	0.0

Figure 2: Load data as a CSV file

Then we can use the `getNearestNeighbors` function to find the nearest neighbors:

```
k = 3
features = df[['Programming Languages', 'Delivered Assignments', 'GPA']].values
labels = df['Passed Exam'].values

# Test indices
test_indices = [8, 9, 10, 11]

for idx in test_indices:
    test_instance = features[idx]
    neighbors = getNearestNeighbors(features[:8], test_instance, k)
    print(f"Student No. {idx+1}: Nearest neighbors are: {[n[0]+1 for n in neighbors]},",
          f" their distances are: {[round(n[1], 3) for n in neighbors]}")
    neighbor_indices = [n[0] for n in neighbors]
    neighbor_labels = [labels[i] for i in neighbor_indices]
    predicted_label = max(set(neighbor_labels), key=neighbor_labels.count)
    if predicted_label == 1.0:
        print(f"Student No. {idx+1}: Predicted Passed Exam = Yes")
    else:
        print(f"Student No. {idx+1}: Predicted Passed Exam = No")
    print()
```

Figure 3: K-Nearest Neighbor Example

#### Results:

- **Student No. 9:**
  - Nearest neighbors: [6, 7, 1], Distances: [1.005, 1.792, 3.164]
  - Predicted Passed Exam: **No**
- **Student No. 10:**
  - Nearest neighbors: [6, 7, 1], Distances: [1.487, 2.1, 2.193]
  - Predicted Passed Exam: **No**
- **Student No. 11:**
  - Nearest neighbors: [2, 1, 3], Distances: [1.414, 1.792, 2.693]
  - Predicted Passed Exam: **Yes**
- **Student No. 12:**
  - Nearest neighbors: [1, 2, 3], Distances: [1.077, 1.803, 2.0]
  - Predicted Passed Exam: **Yes**

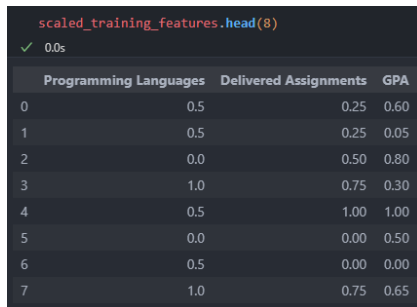
**Question 3b:** Show the effect of feature scaling by transforming each feature (column) to be in the range  $[0, 1]$ . The minimum value of the feature in the training data maps to 0, and the maximum value of the same feature maps to 1. Predict the labels again.

**Solution:**

to Scale the value of the features to be in the range  $[0, 1]$ , we can use the Min-Max scaling method. The formula for Min-Max scaling is given by:

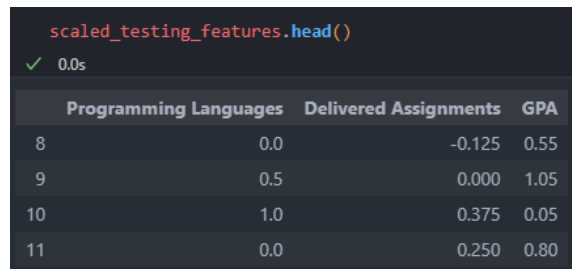
```
def minMaxScaler(data, scaler):
    return (data - scaler.min()) / (scaler.max() - scaler.min())
```

Giving Us the following results:



	Programming Languages	Delivered Assignments	GPA	
0	0.5	0.25	0.60	
1	0.5	0.25	0.05	
2	0.0	0.50	0.80	
3	1.0	0.75	0.30	
4	0.5	1.00	1.00	
5	0.0	0.00	0.50	
6	0.5	0.00	0.00	
7	1.0	0.75	0.65	

Figure 4: Training



	Programming Languages	Delivered Assignments	GPA	
8	0.0	-0.125	0.55	
9	0.5	0.000	1.05	
10	1.0	0.375	0.05	
11	0.0	0.250	0.80	

Figure 5: Testing

**Results:**

- **Student No. 9:**

- Nearest neighbors: [6, 1, 3], Distances: [0.135, 0.627, 0.673]
- Predicted Passed Exam: **Yes**

- **Student No. 10:**

- Nearest neighbors: [1, 6, 3], Distances: [0.515, 0.743, 0.75]
- Predicted Passed Exam: **Yes**

- **Student No. 11:**

- Nearest neighbors: [4, 2, 7], Distances: [0.451, 0.515, 0.627]
- Predicted Passed Exam: **No**

- **Student No. 12:**

- Nearest neighbors: [3, 6, 1], Distances: [0.25, 0.391, 0.539]
- Predicted Passed Exam: **Yes**

**Question 4:** Consider the set of training examples in the diagram below.

1. How will the point (8, 8) be classified by the 1-nearest neighbor classifier?

**Answer:** The point (8, 8) will be classified as Negative

2. How will the point (8, 1) be classified by the 1-nearest neighbor classifier?

**Answer:** Similarly, the point (8, 1) will be classified as Positive

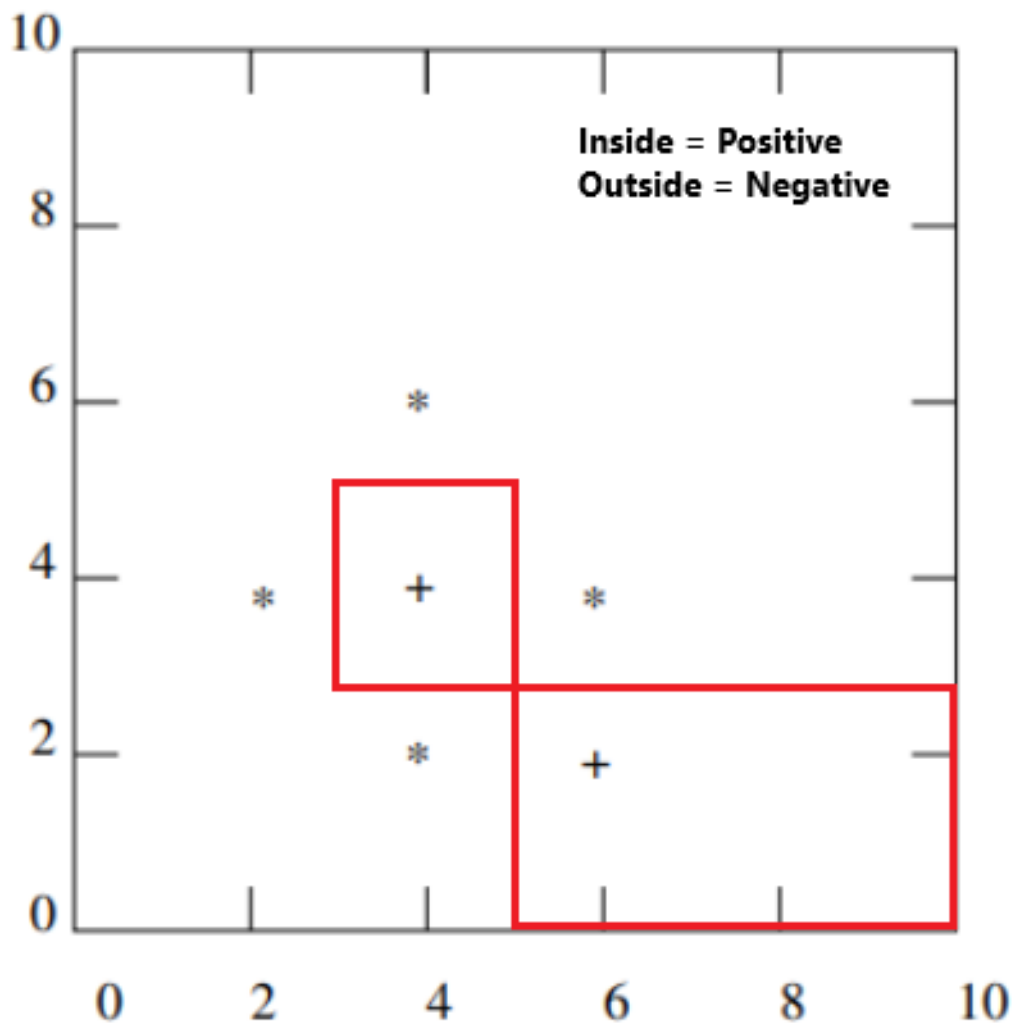


Figure 6: K-Nearest Neighbor Decision Boundary

**Question 5:** For the data given with 10 points and 2 classes:

1. What is the leave-one-out cross-validation error when using K-nearest neighbor classifier with  $K = 1$ ?
2. Which of the following values of K leads to the minimum number of leave-one-out validation errors: 3, 5, or 9?

**Solution:**

From the given data, we can deduce that the shortest distance  $K = 1$  for each point is of a different class. Therefore, the MSE for each point is 1. .  
The leave one out cross-validation error can be calculated as follows:

$$\text{Leave-One-Out Error} = \frac{1}{N} \sum_{i=1}^N \text{MSE}_i$$

Therefore, the leave-one-out cross-validation error is 1.

he following values of K leads to the minimum number of leave-one-out validation errors is 5