

Alex Kitsul
230134210
CPSC 450
Assignment 6 - Report
April 1

Pseudocode

```
main:
    k <- from file
    m <- from file
    points <- from file as format (x, y)

    centers <- LloydsAlgorithm(points, k)
    print(centers)

LloydsAlgorithm(data, k):
    centers = random k centers
    clusters = dict() with center indexes as keys and empty lists as values

    while (!converged):
        for point in data:
            closest_center = None
            for center in centers:
                if (distance(point, center) < distance(point, closest_center)):
                    closest_center = center

            clusters[closest_center].append(point)

        for center in clusters:
            new_center = mean of all points in cluster
            if centers are the same:
                converged = True
            centers[center] = new_center
            clusters[center] = []

    return centers
```

Program Code

LloydsAlgorithm.py

```
from math import dist
from numpy import mean, round_

def main():
    """
        Main driver method

        Parameters:

            None

        Returns:

            None. Side effect.
    """
    file_name = input("Enter the file name: ")

    with open(file_name, "r") as file:
        first_line = file.readline().split(" ")

        k = int(first_line[0])
        m = int(first_line[1])
        points = []
        for i in file:
            coords = tuple(float(j) for j in i.split())
            points.append(coords)

        centers = LloydsAlgorithm(points, k)
        for i in centers:
            print(i)

def LloydsAlgorithm(data, k):
    """
        Execute the Lloyds Algorithm to find k centers

        Parameters:

            data (list[tuple]): A list of tuples representing the data points
            k (int): The number of centers to find

        Returns:

            centers (list[tuple]): A list of tuples representing the centers
    """
```

```
flag = True
# Initialize centers
centers = []
clusters = dict()
# Choose random points as first centers
for i in range(k):
    centers.append(data[i])
    clusters[i] = []

# While we haven't yet converged
while(flag):
    # For each point
    for i in range(len(data)):
        closest_center_index = None
        closest_center_dist = None
        # For each center
        for j in range(len(centers)):
            # Calculate distance from point to center, if the center is closer, update
            if (closest_center_dist is None or dist(data[i], centers[j]) < \
                closest_center_dist):
                closest_center_index = j
                closest_center_dist = dist(data[i], centers[j])

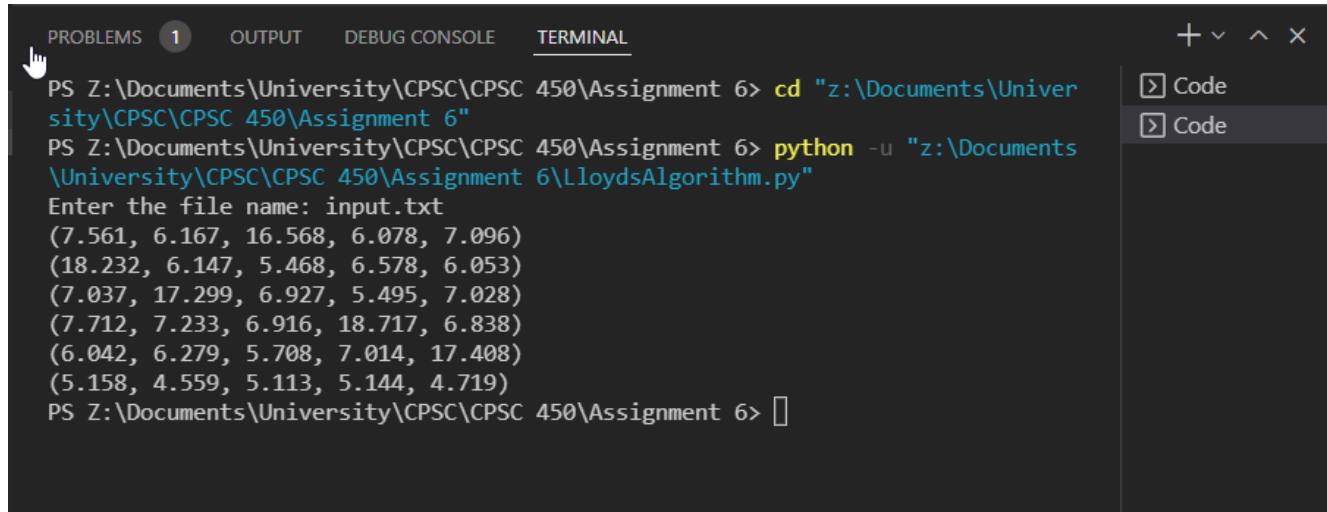
        # Add point to closest center index
        clusters[closest_center_index].append(data[i])

    # Update centers
    flag = False
    for i in clusters:
        # Calculate the mean of the points in the cluster
        new_center = tuple(round_(mean(clusters[i], axis=0), decimals=3))
        # If it's different, then we need to continue
        if (new_center != centers[i]):
            flag = True
        # Update the center
        centers[i] = new_center
        # Clear the clusters
        clusters[i] = []

return centers

if __name__ == "__main__":
    main()
```

Examples with Output



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal has tabs for PROBLEMS (1), OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal content shows the following commands and output:

```
PS Z:\Documents\University\CPSC\CPSC 450\Assignment 6> cd "z:\Documents\University\CPSC\CPSC 450\Assignment 6"
PS Z:\Documents\University\CPSC\CPSC 450\Assignment 6> python -u "z:\Documents\University\CPSC\CPSC 450\Assignment 6\LloydsAlgorithm.py"
Enter the file name: input.txt
(7.561, 6.167, 16.568, 6.078, 7.096)
(18.232, 6.147, 5.468, 6.578, 6.053)
(7.037, 17.299, 6.927, 5.495, 7.028)
(7.712, 7.233, 6.916, 18.717, 6.838)
(6.042, 6.279, 5.708, 7.014, 17.408)
(5.158, 4.559, 5.113, 5.144, 4.719)
PS Z:\Documents\University\CPSC\CPSC 450\Assignment 6> 
```

On the right side of the terminal, there are two 'Code' buttons with a magnifying glass icon, and window control buttons (+, -, ^, x) at the top right.