Discovery Learning: K-Means Step-by-Step

Understanding the K-Means Algorithm Through Practice

Machine Learning for Smarter Innovation - Week 1

Learning Objectives

By completing this exercise, you will:

- Understand each step of the K-means algorithm
- Practice manual clustering to build intuition
- See how computers optimize cluster assignments
- Learn to predict convergence patterns

The K-Means Algorithm

Algorithm Steps

1. **Initialize**: Choose k initial cluster centers (randomly or using k-means++)
2. **Assign**: Assign each point to the nearest cluster center
3. **Update**: Move centers to the mean of their assigned points
4. **Repeat**: Continue until centers stop moving (convergence)

Exercise 1: Manual K-Means (k=2)

Given these 10 innovation projects with their complexity (x) and impact (y) scores:

Project	A	B	C	D	E	F	G	H	I	J
Complexity (x)	2	3	8	9	2	3	8	9	5	5
Impact (y)	2	3	8	9	8	9	2	3	5	6
Your Cluster										

Step 1: Initialize Centers

Let's start with centers at Project A (2,2) and Project C (8,8).

Your Work

Calculate distances from each point to both centers using: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Distance to Center 1 (2,2):

- A to C1: _____
- B to C1: _____
- C to C1: _____
- D to C1: _____
- E to C1: _____

Distance to Center 2 (8,8):

- A to C2: _____
- B to C2: _____
- C to C2: _____
- D to C2: _____
- E to C2: _____

Step 2: First Assignment

Assign each project to its nearest center:

Cluster Assignment

Cluster 1 members: _____
Cluster 2 members: _____

Step 3: Update Centers

Calculate new center positions (mean of assigned points):

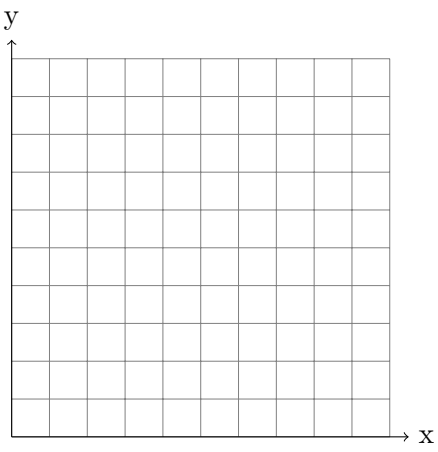
New Centers

New Center 1:
x = (sum of x-coordinates) / (number of points) = _____
y = (sum of y-coordinates) / (number of points) = _____

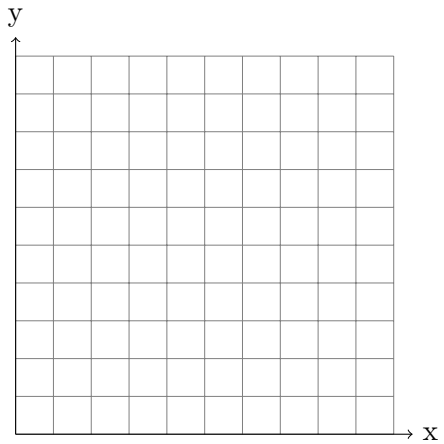
New Center 2:
x = _____
y = _____

Exercise 2: Predicting Convergence

Draw your predictions:



Initial Configuration



After Convergence

Exercise 3: Algorithm Behavior

Answer these questions based on your manual calculation:

1. How many iterations did it take to converge? _____
2. What would happen with different initial centers? _____
3. Which points might switch clusters if k=3? _____
4. What's the total within-cluster sum of squares? _____

Exercise 4: Real Implementation

Compare your manual results with sklearn's implementation:

Python Code

```
from sklearn.cluster import KMeans
import numpy as np

# Your data
X = np.array([[2,2], [3,3], [8,8], [9,9], [2,8],
              [3,9], [8,2], [9,3], [5,5], [5,6]])

# Run K-means
kmeans = KMeans(n_clusters=2, random_state=42)
labels = kmeans.fit_predict(X)
centers = kmeans.cluster_centers_

print("Cluster assignments:", labels)
print("Final centers:", centers)
print("Iterations:", kmeans.n_iter_)
```

Your observations:

- Did sklearn get the same clusters? _____
- Were the final centers identical? _____

- Why might there be differences? _____

Key Insights

- K-means is **deterministic** given the same initial centers
- Different initializations can lead to **different final clusters**
- The algorithm finds a **local optimum**, not necessarily global
- Convergence typically happens in **5-10 iterations**
- The result minimizes **within-cluster variance**

Challenge Extensions

1. Try k=3 with the same data. How do clusters change?
2. Add an outlier at (15, 15). What happens to the clusters?
3. Calculate the silhouette score for k=2 vs k=3
4. Implement one iteration in Python without using sklearn

Reflection Questions

Think Deeper

1. When would you choose a different k value in real applications?
2. How could you automate the choice of initial centers?
3. What business decisions could you make from these clusters?
4. How would you validate that your clusters are meaningful?