Machine Learning
(Unsupervised Learning)

Hello!

I am Eslam Ahmed

I am a software engineer.

You can find me at jeksogsa@gmail.com

Hello!

I am Eman Ehab

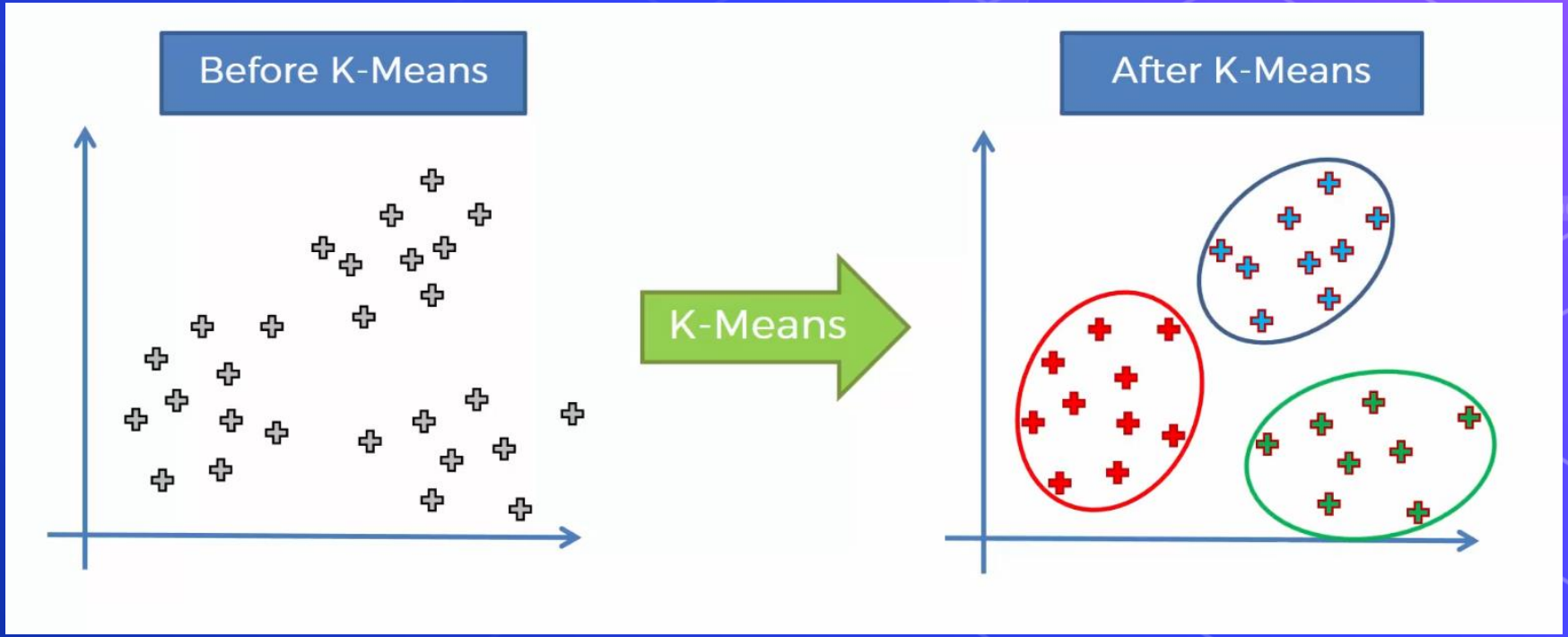
I am a ML research engineer.

You can find me at
emanehab.ieee@gmail.com

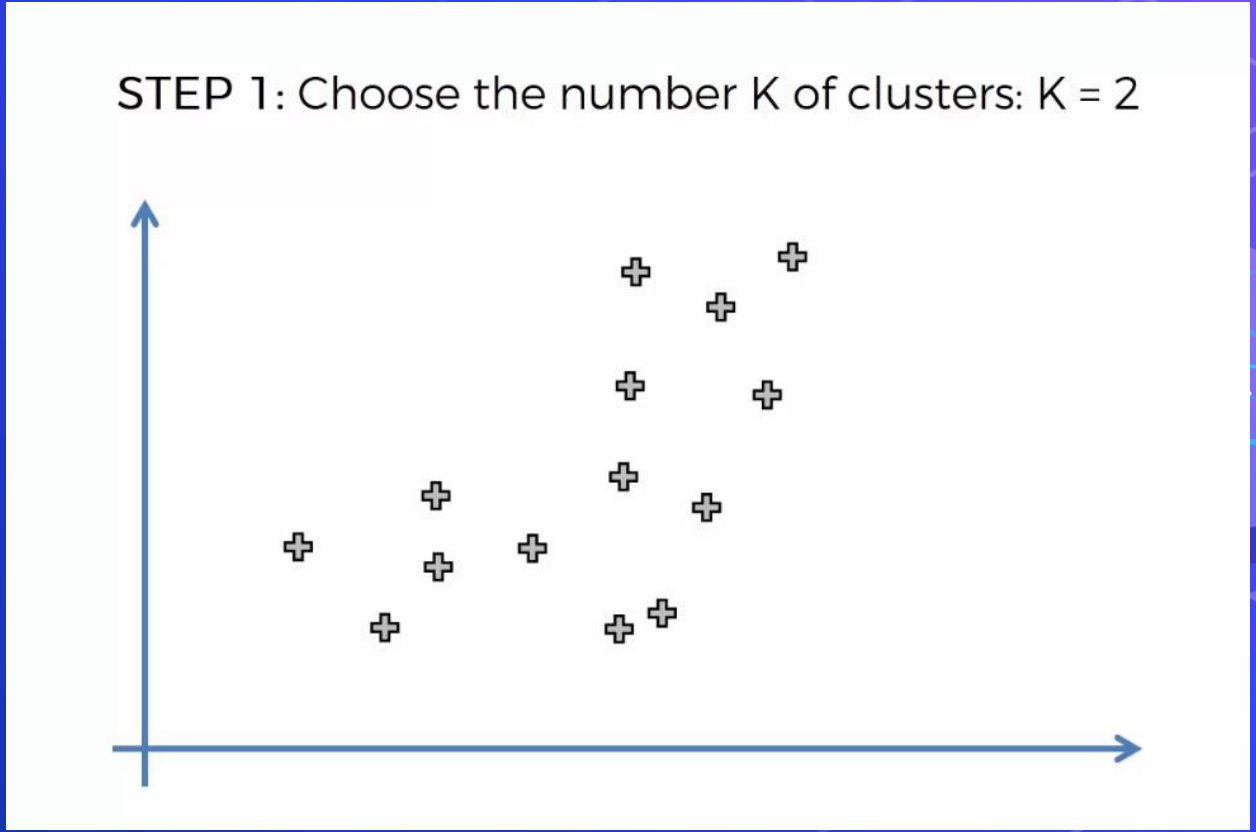
- Machine Learning
- Supervised Learning
  - Regression
    - Simple Linear Regression
    - Multiple Linear Regression
    - Polynomial Regression
    - Evaluating Model Performance
  - Classification
    - Logistic Regression
    - K-Nearest Neighbors (KNN)
    - Naive Bayes
    - SVM
    - Decision Trees
    - Ensemble Methods
      - What is Bagging & Boosting
      - Random Forests
      - XGBoost
    - Evaluating Model Performance
- Unsupervised Learning
  - Clustering
    - KMeans
    - Hierarchical Clustering
    - Density Based Clustering - DBSCAN
  - Association rule mining
    - Apriori
  - Dimension Reduction
    - PCA
    - LDA
  - Evaluating Model Performance
- Model Selection & Evaluation
  - Cross Validation
  - Hyperparameter Tuning
    - Grid Search
    - Randomized Search
- Recommendation Systems

- Machine Learning
- Supervised Learning
  - Regression
    - Simple Linear Regression
    - Multiple Linear Regression
    - Polynomial Regression
    - Evaluating Model Performance
  - Classification
    - Logistic Regression
    - K-Nearest Neighbors (KNN)
    - Naive Bayes
    - SVM
    - Decision Trees
    - Ensemble Methods
      - What is Bagging & Boosting
      - Random Forests
      - XGBoost
    - Evaluating Model Performance
- **Unsupervised Learning**
  - **Clustering**
    - **KMeans**
    - Hierarchical Clustering
    - Density Based Clustering - DBSCAN
  - Association rule mining
    - Apriori
  - Dimension Reduction
    - PCA
    - LDA
  - Evaluating Model Performance
- Model Selection & Evaluation
  - Cross Validation
  - Hyperparameter Tuning
    - Grid Search
    - Randomized Search
- Recommendation Systems

KMeans

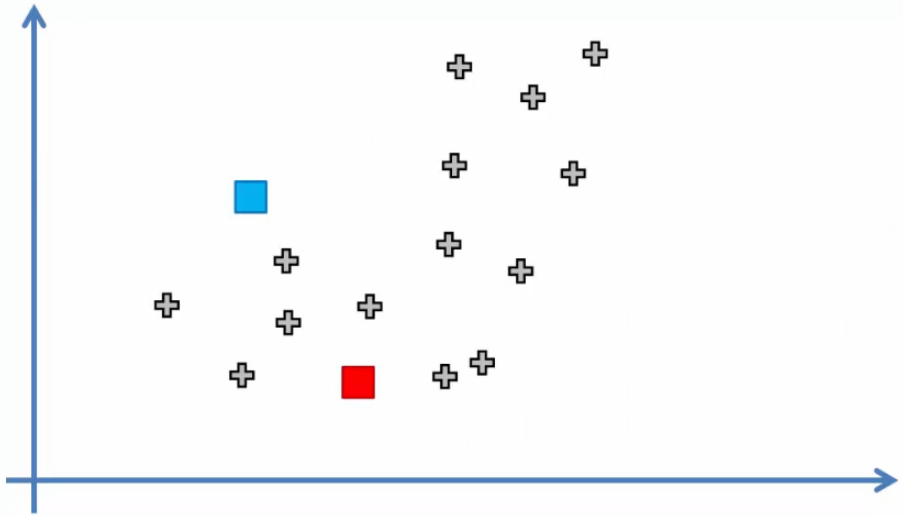


KMeans



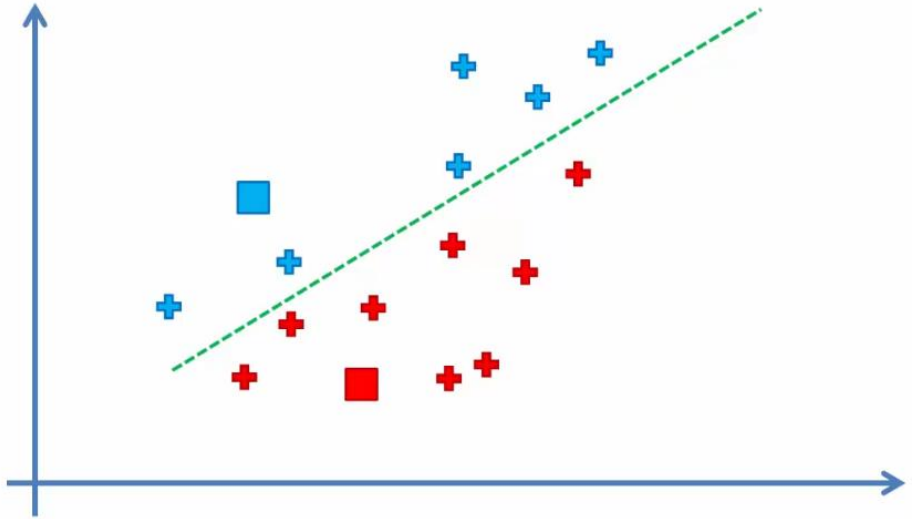
KMeans

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)



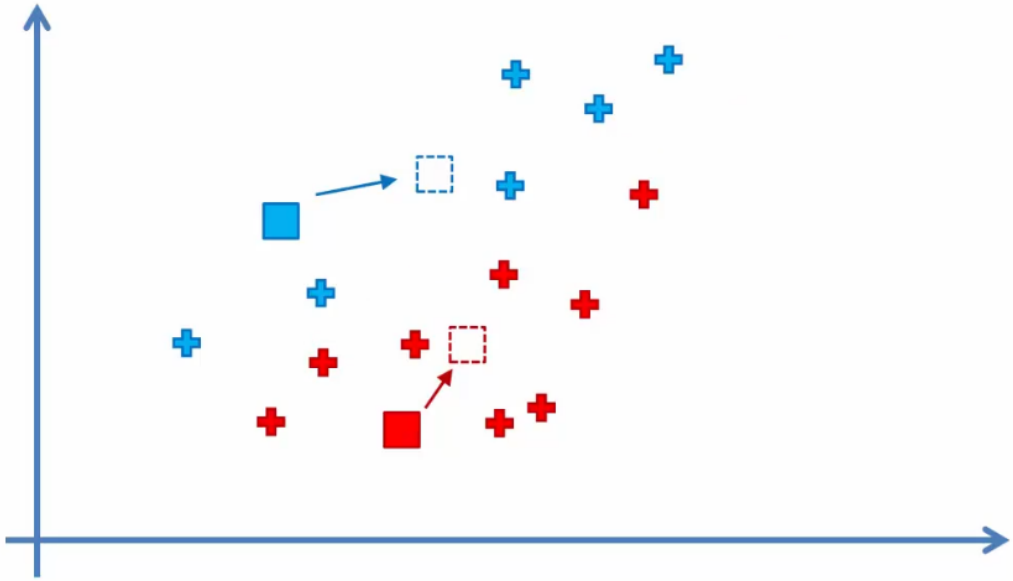
KMeans

STEP 3: Assign each data point to the closest centroid ➡ That forms K clusters



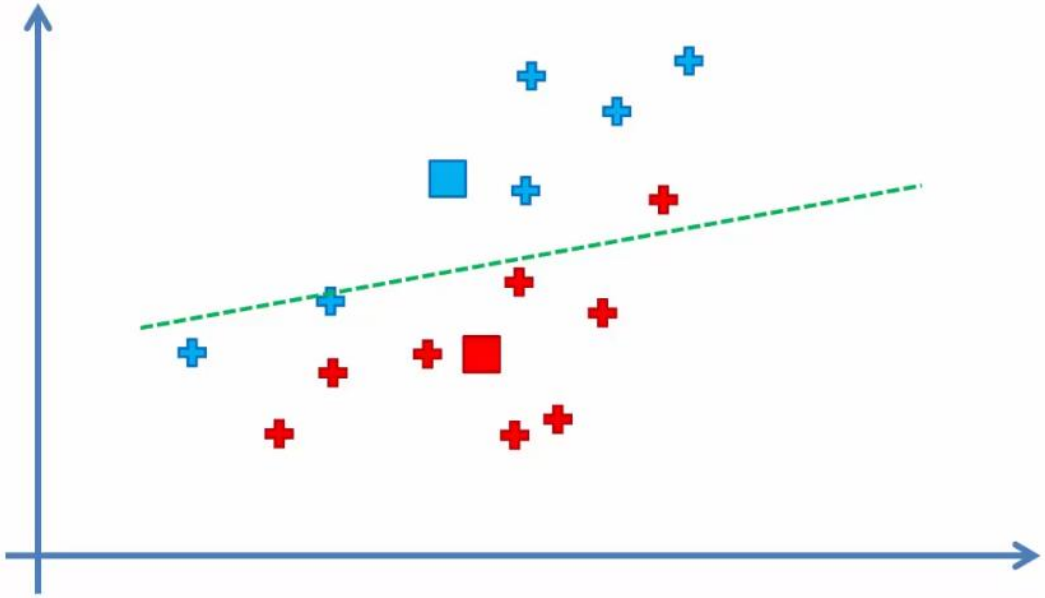
KMeans

STEP 4: Compute and place the new centroid of each cluster



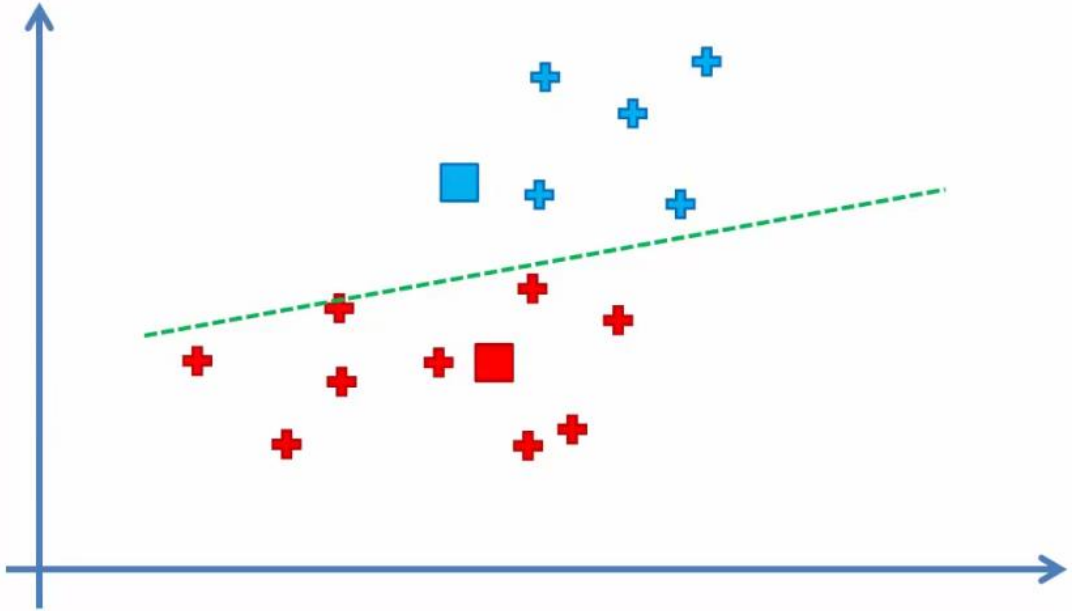
KMeans

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

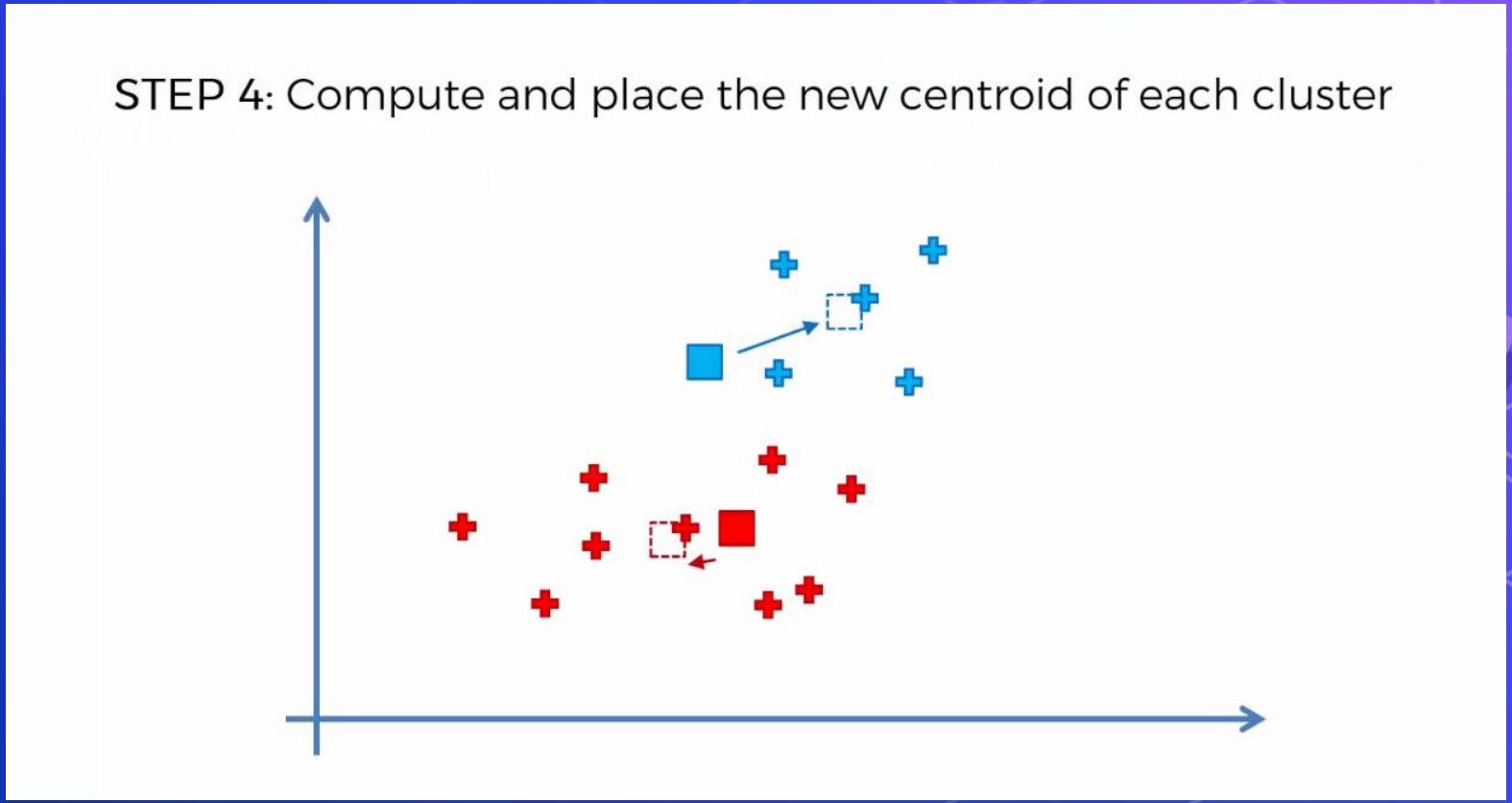


KMeans

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

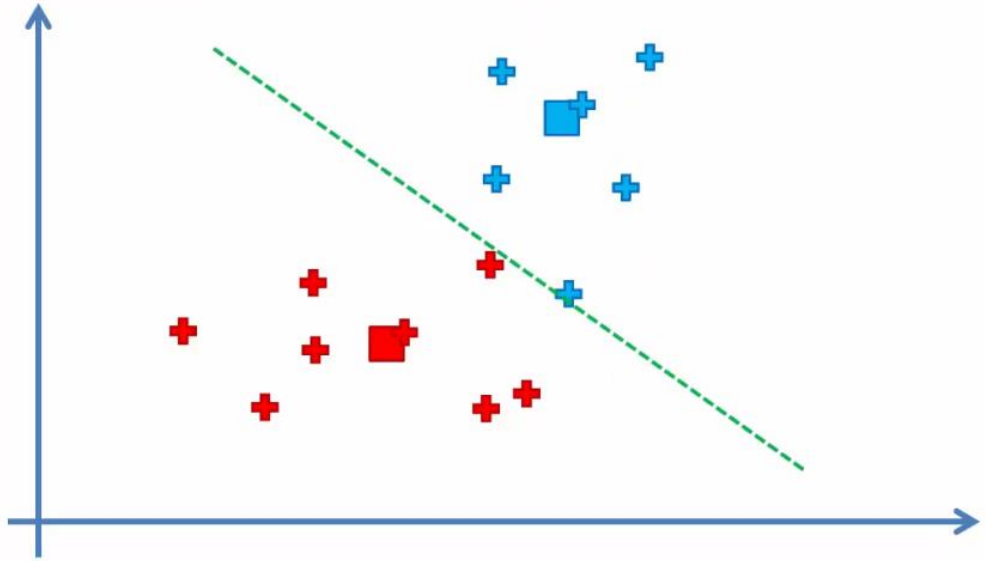


KMeans



KMeans

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took place, go to STEP 4, otherwise go to FIN.

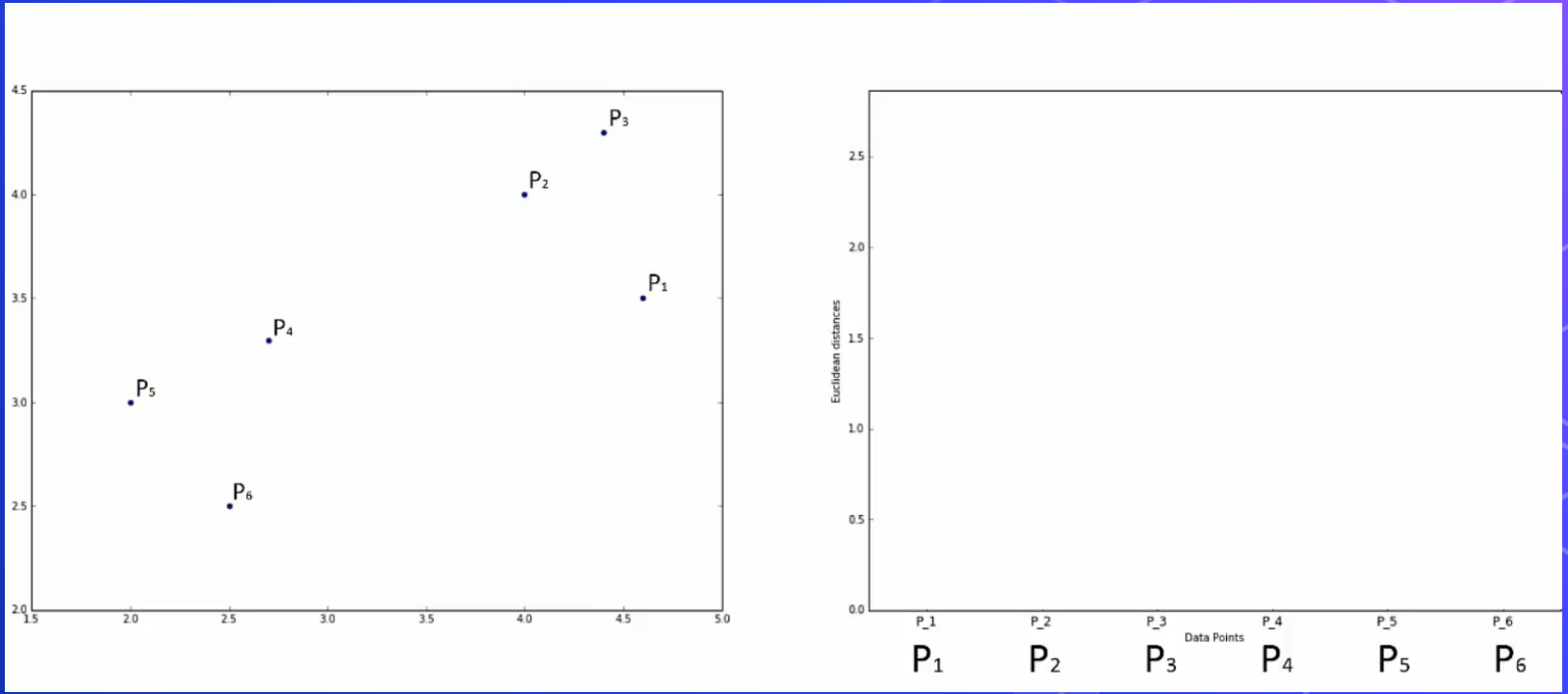


KMeans

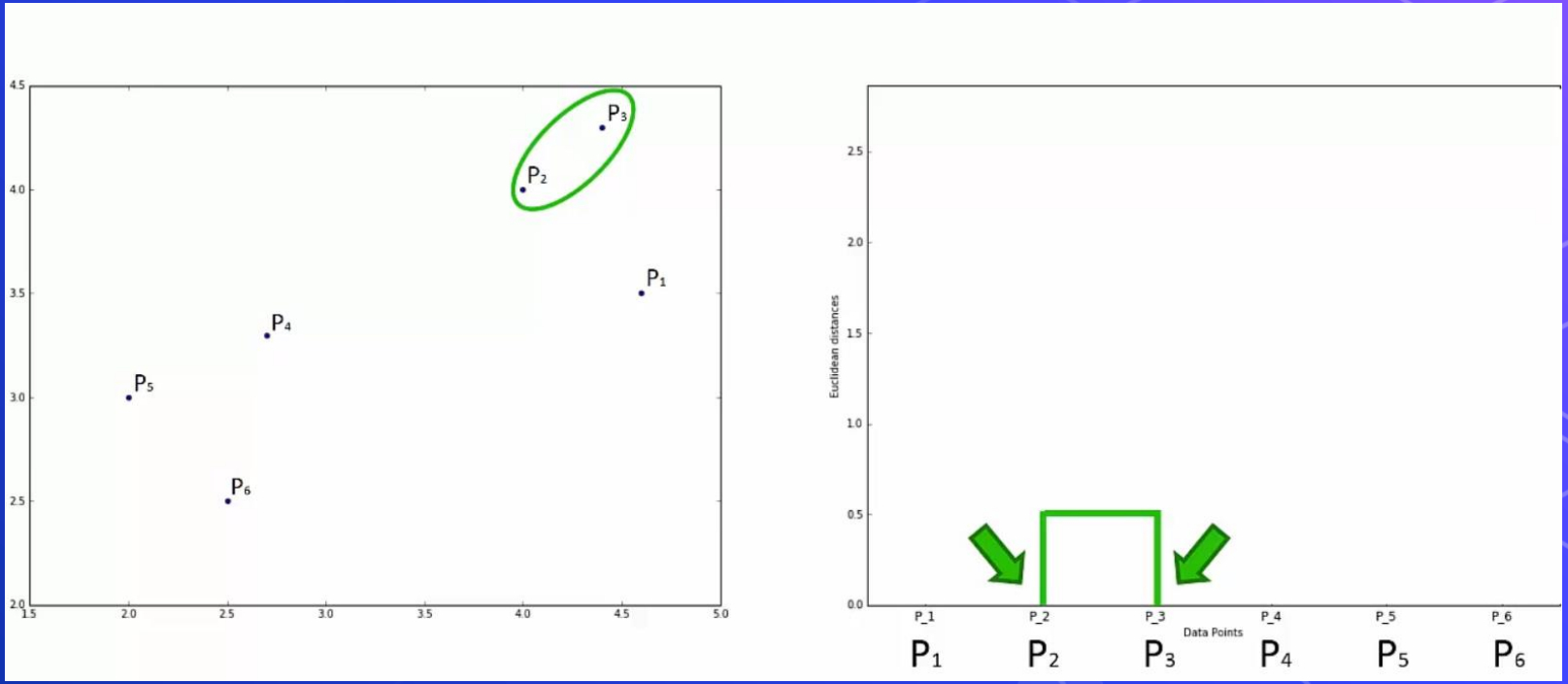
```
1 from sklearn.cluster import KMeans
2
3 kmeans = KMeans(n_clusters=3)
4 kmeans.fit(X)
5 kmeans.predict(X)
```

- Machine Learning
- Supervised Learning
  - Regression
    - Simple Linear Regression
    - Multiple Linear Regression
    - Polynomial Regression
    - Evaluating Model Performance
  - Classification
    - Logistic Regression
    - K-Nearest Neighbors (KNN)
    - Naive Bayes
    - SVM
    - Decision Trees
    - Ensemble Methods
      - What is Bagging & Boosting
      - Random Forests
      - XGBoost
    - Evaluating Model Performance
- **Unsupervised Learning**
  - **Clustering**
    - KMeans
    - **Hierarchical Clustering**
    - Density Based Clustering - DBSCAN
  - Association rule mining
    - Apriori
  - Dimension Reduction
    - PCA
    - LDA
  - Evaluating Model Performance
- Model Selection & Evaluation
  - Cross Validation
  - Hyperparameter Tuning
    - Grid Search
    - Randomized Search
- Recommendation Systems

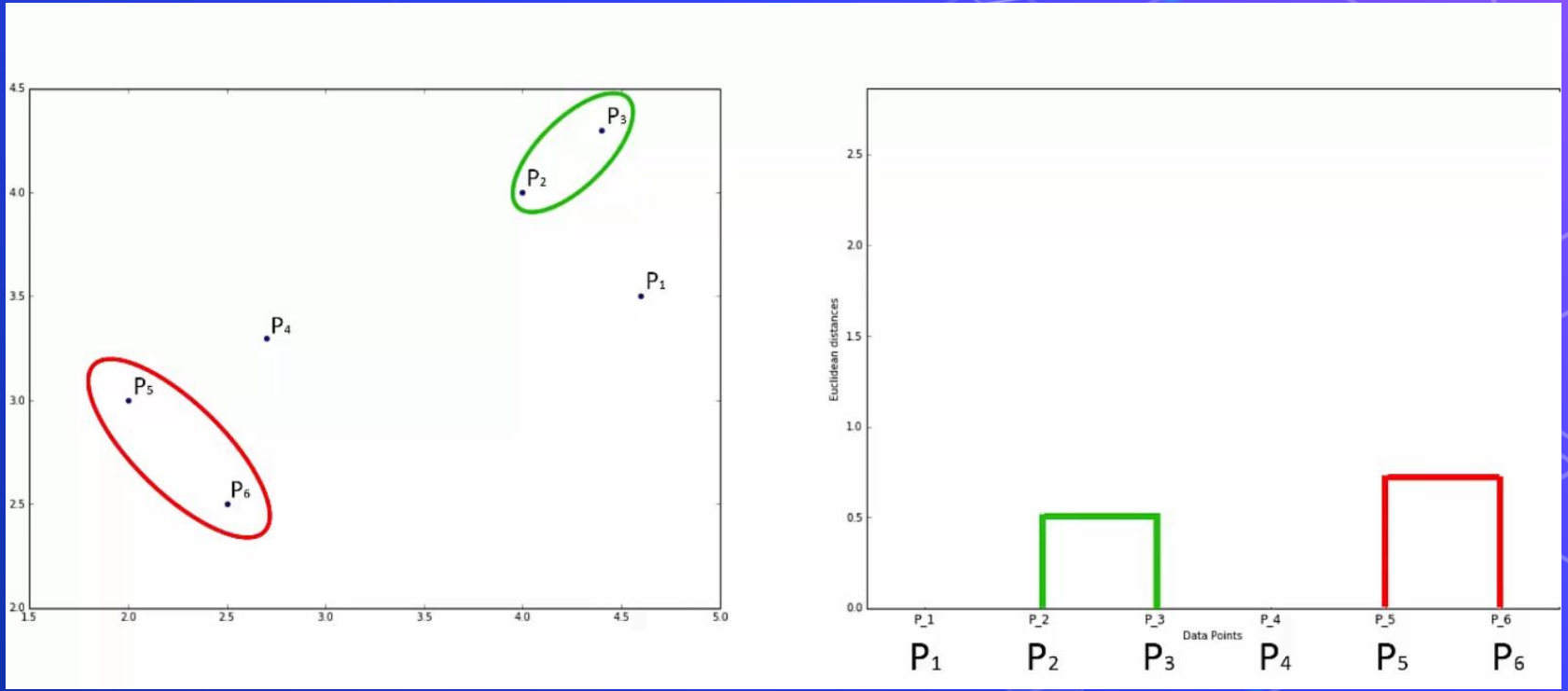
Hierarchical Clustering with dendrogram



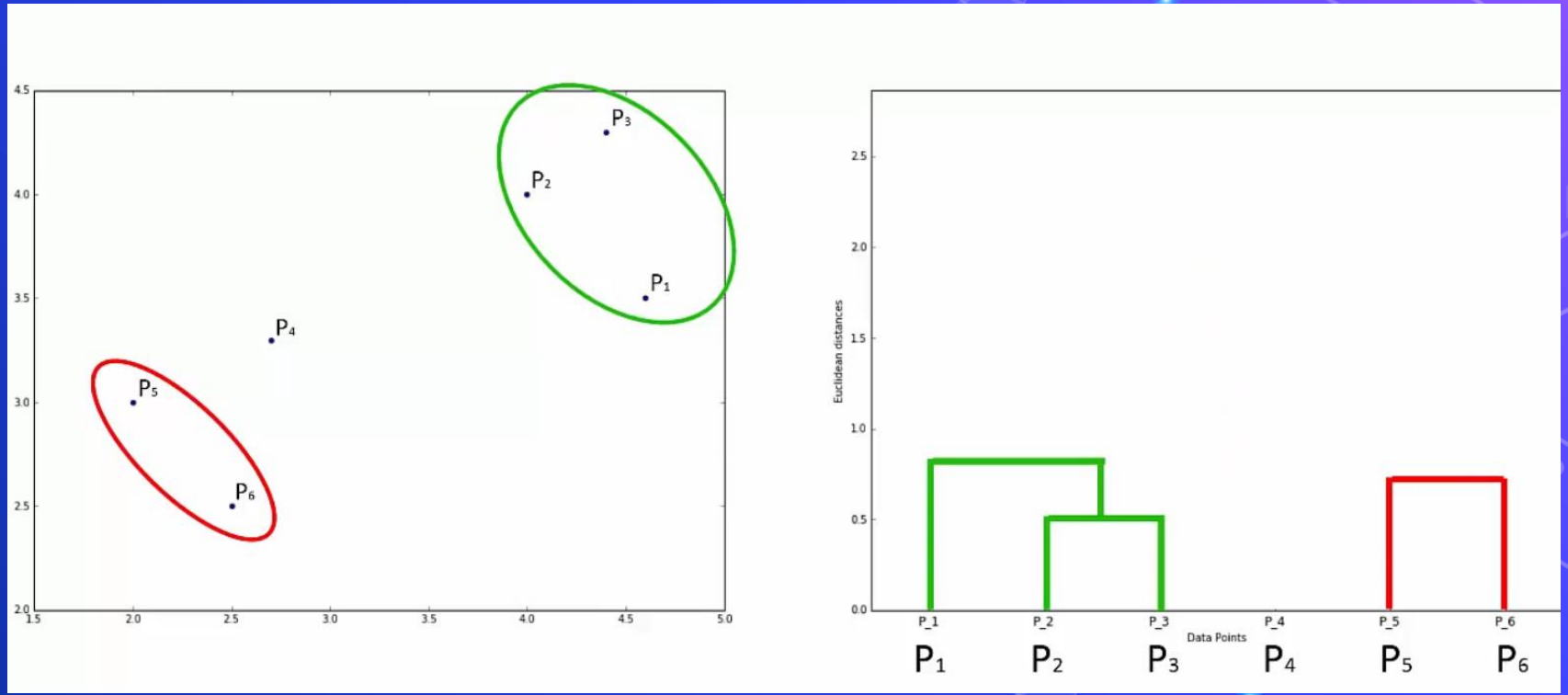
Hierarchical Clustering with dendrogram



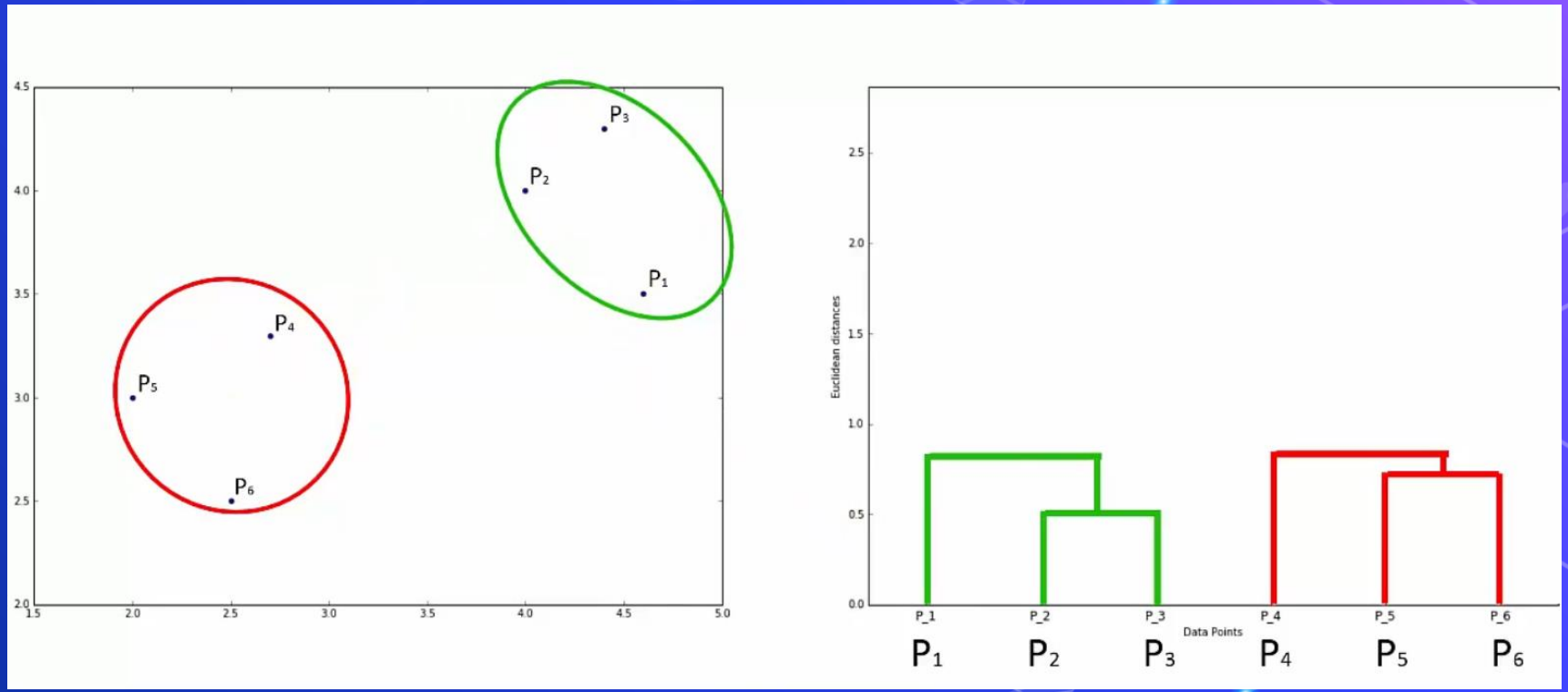
Hierarchical Clustering with dendrogram



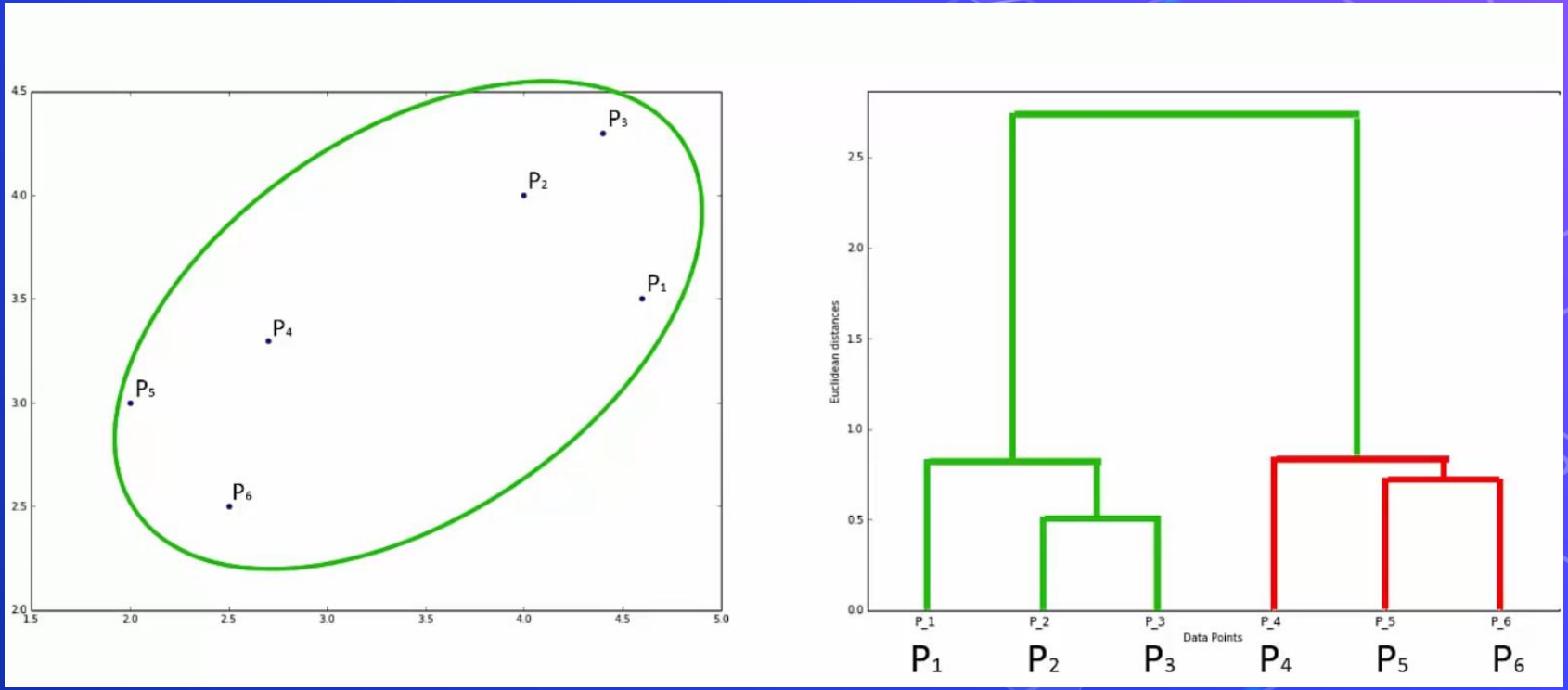
Hierarchical Clustering with dendrogram



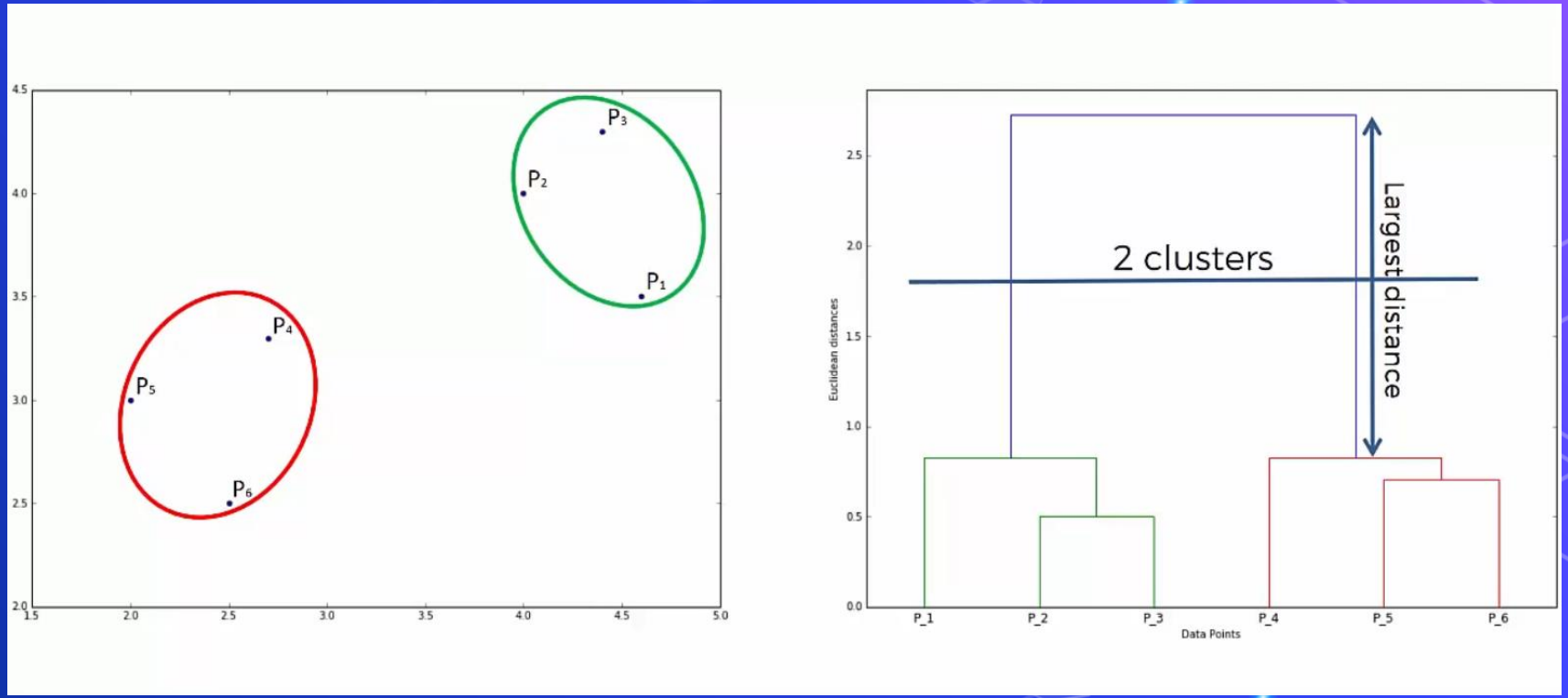
Hierarchical Clustering with dendrogram



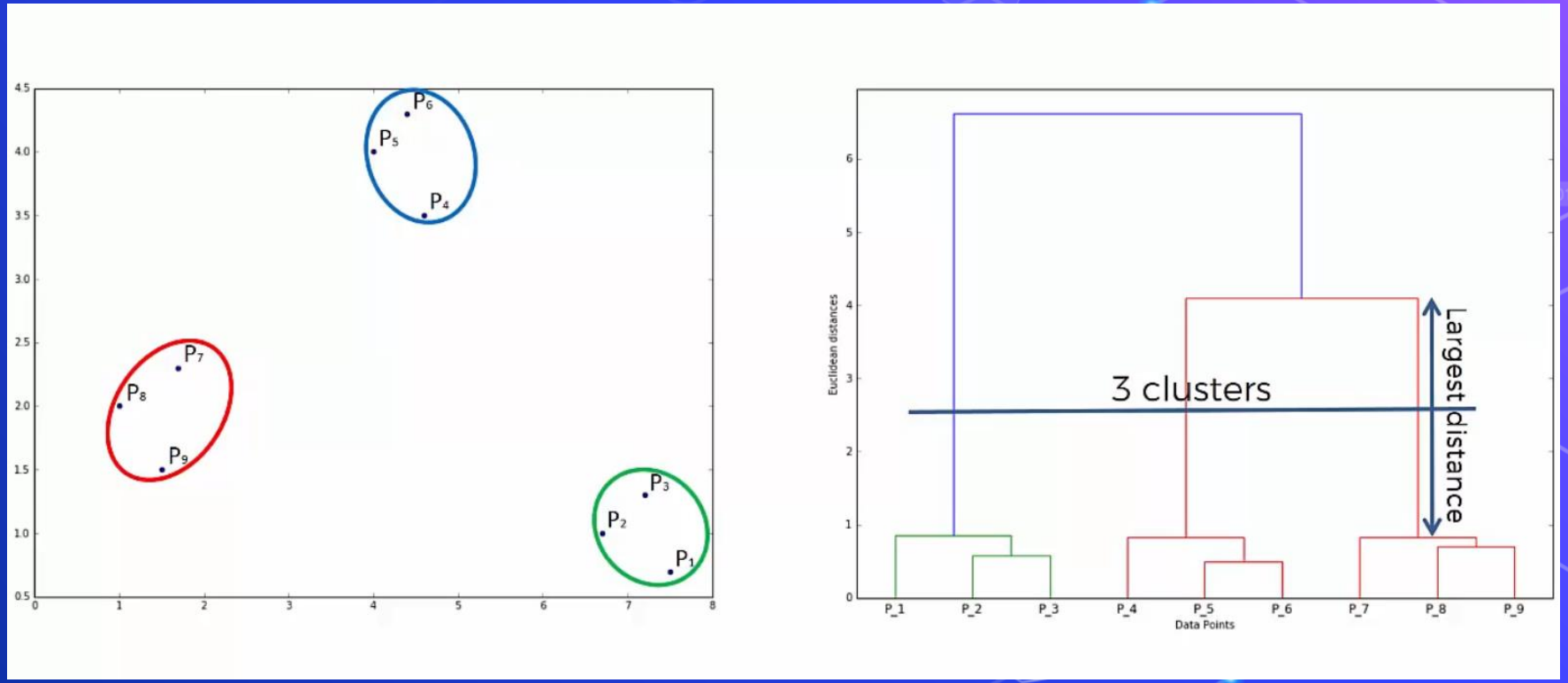
Hierarchical Clustering with dendrogram



Hierarchical Clustering with dendrogram



Hierarchical Clustering with dendrogram

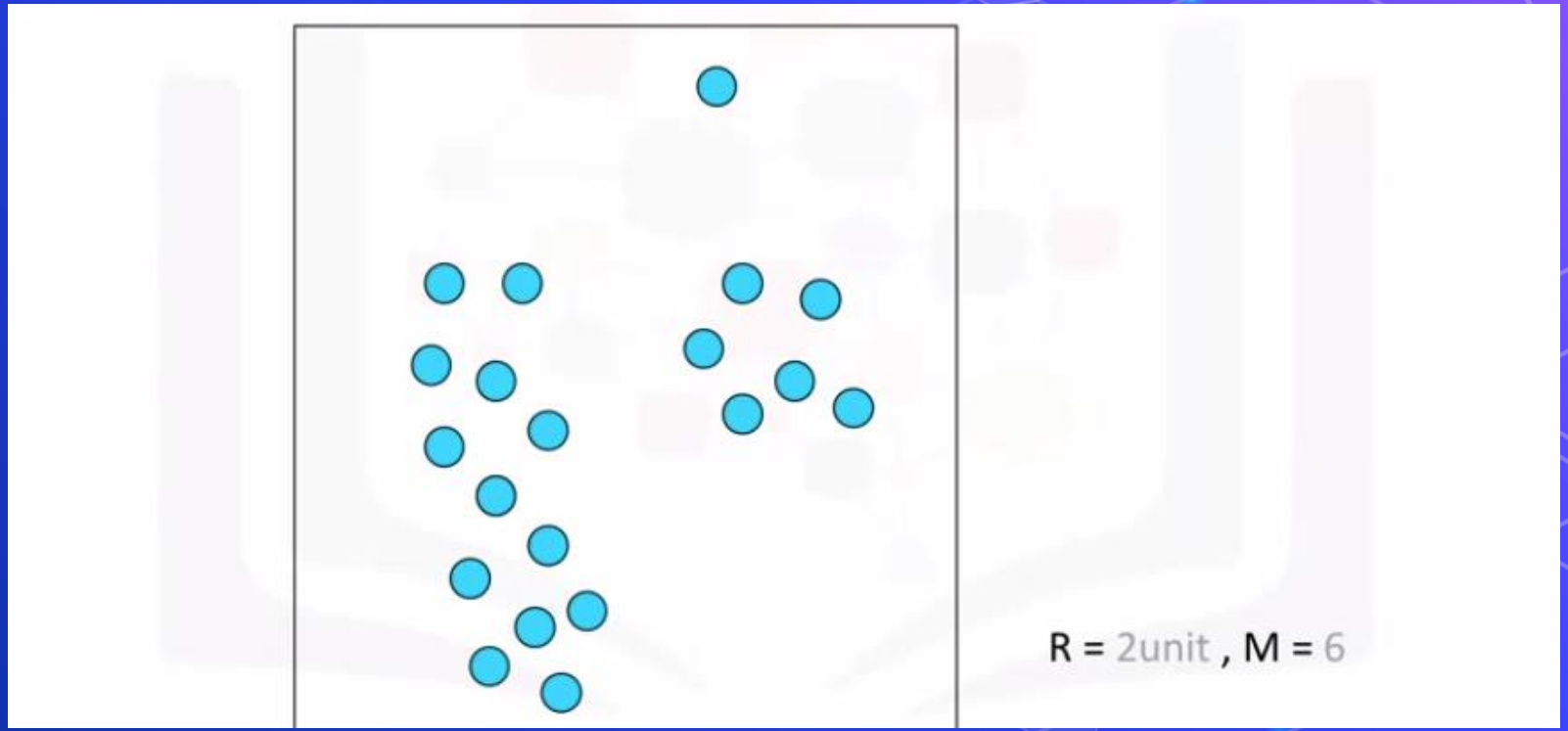


Hierarchical Clustering with dendrogram

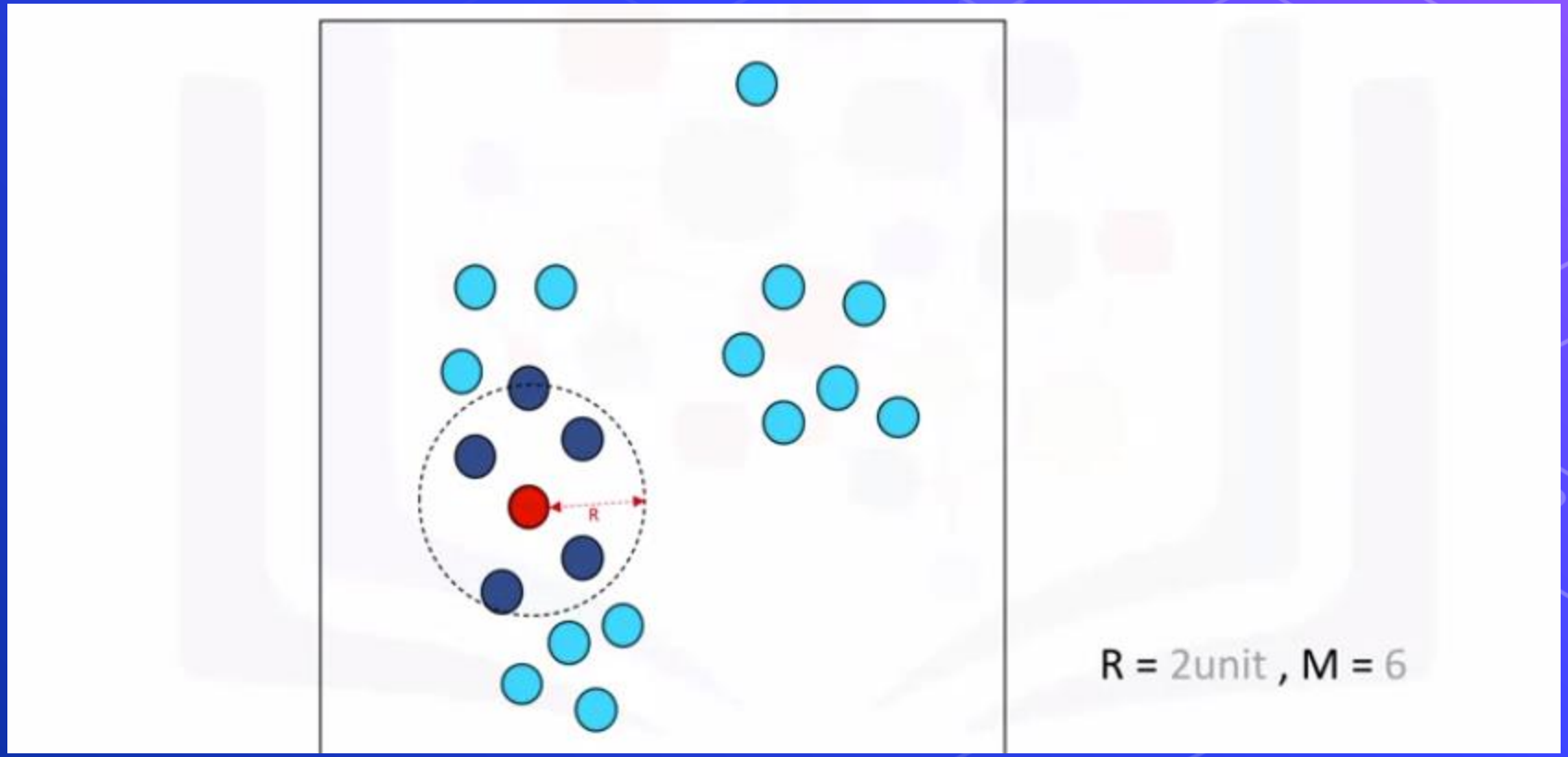
```python
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering

# visualize dendrogram
dendrogram = sch.dendrogram(sch.linkage(x, method='ward'))

# train model
model = AgglomerativeClustering(n_clusters=3)
y_labels = model.fit_predict(x)
```

- Machine Learning
- Supervised Learning
  - Regression
    - Simple Linear Regression
    - Multiple Linear Regression
    - Polynomial Regression
    - Evaluating Model Performance
  - Classification
    - Logistic Regression
    - K-Nearest Neighbors (KNN)
    - Naive Bayes
    - SVM
    - Decision Trees
    - Ensemble Methods
      - What is Bagging & Boosting
      - Random Forests
      - XGBoost
    - Evaluating Model Performance
- **Unsupervised Learning**
  - **Clustering**
    - KMeans
    - Hierarchical Clustering
    - **Density Based Clustering - DBSCAN**
  - Association rule mining
    - Apriori
  - Dimension Reduction
    - PCA
    - LDA
  - Evaluating Model Performance
- Model Selection & Evaluation
  - Cross Validation
  - Hyperparameter Tuning
    - Grid Search
    - Randomized Search
- Recommendation Systems

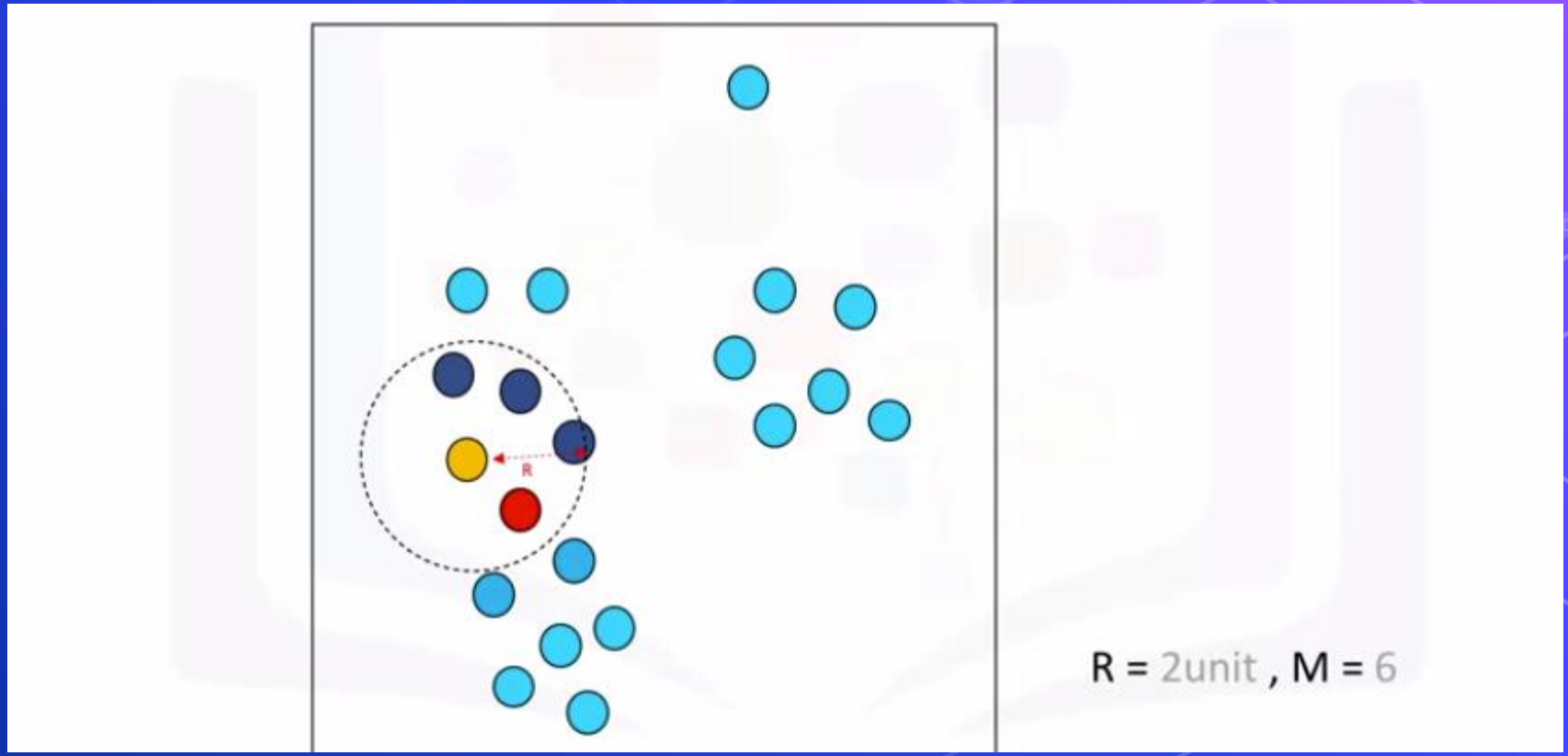
Density Based Clustering - DBSCAN



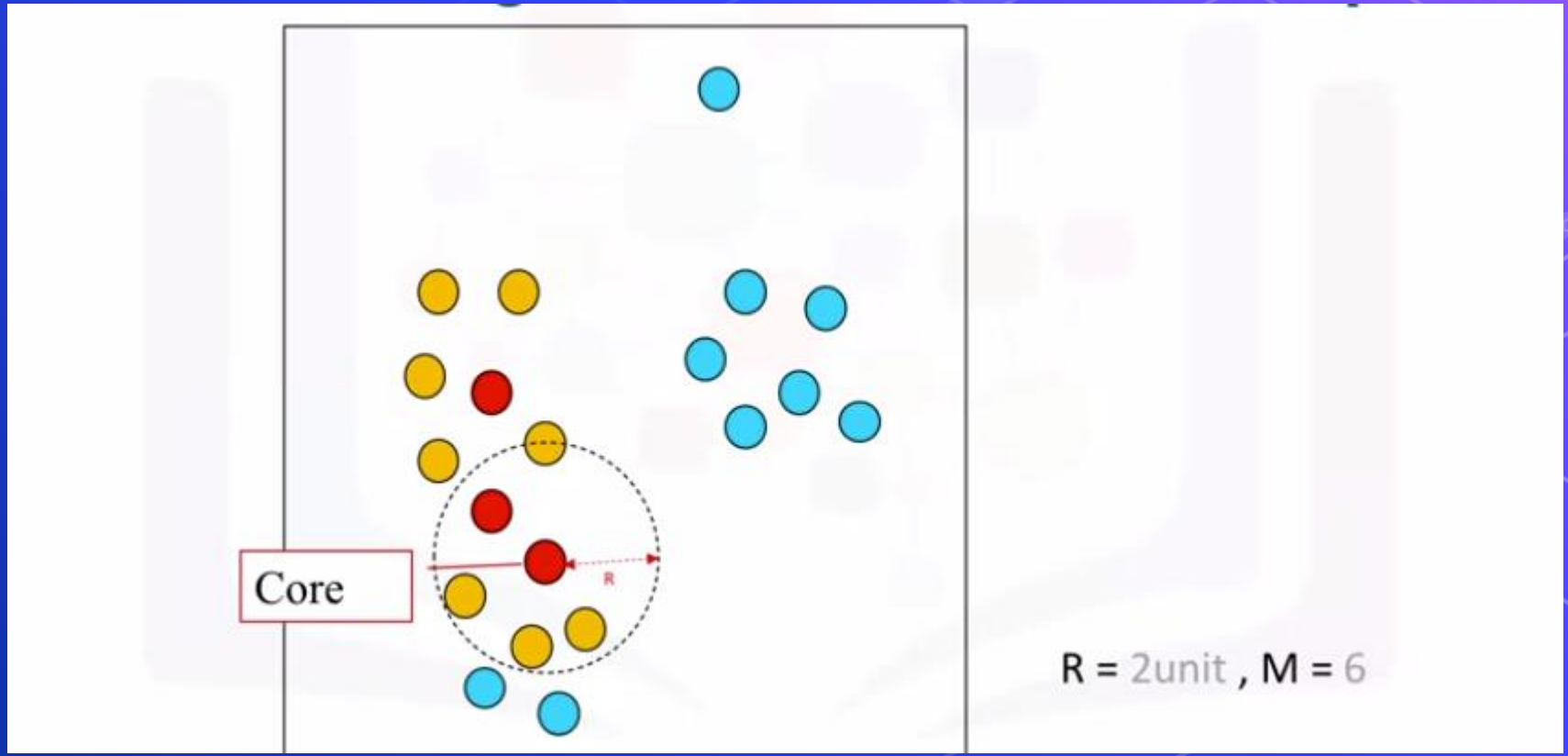
Density Based Clustering – DBSCAN (Core point)



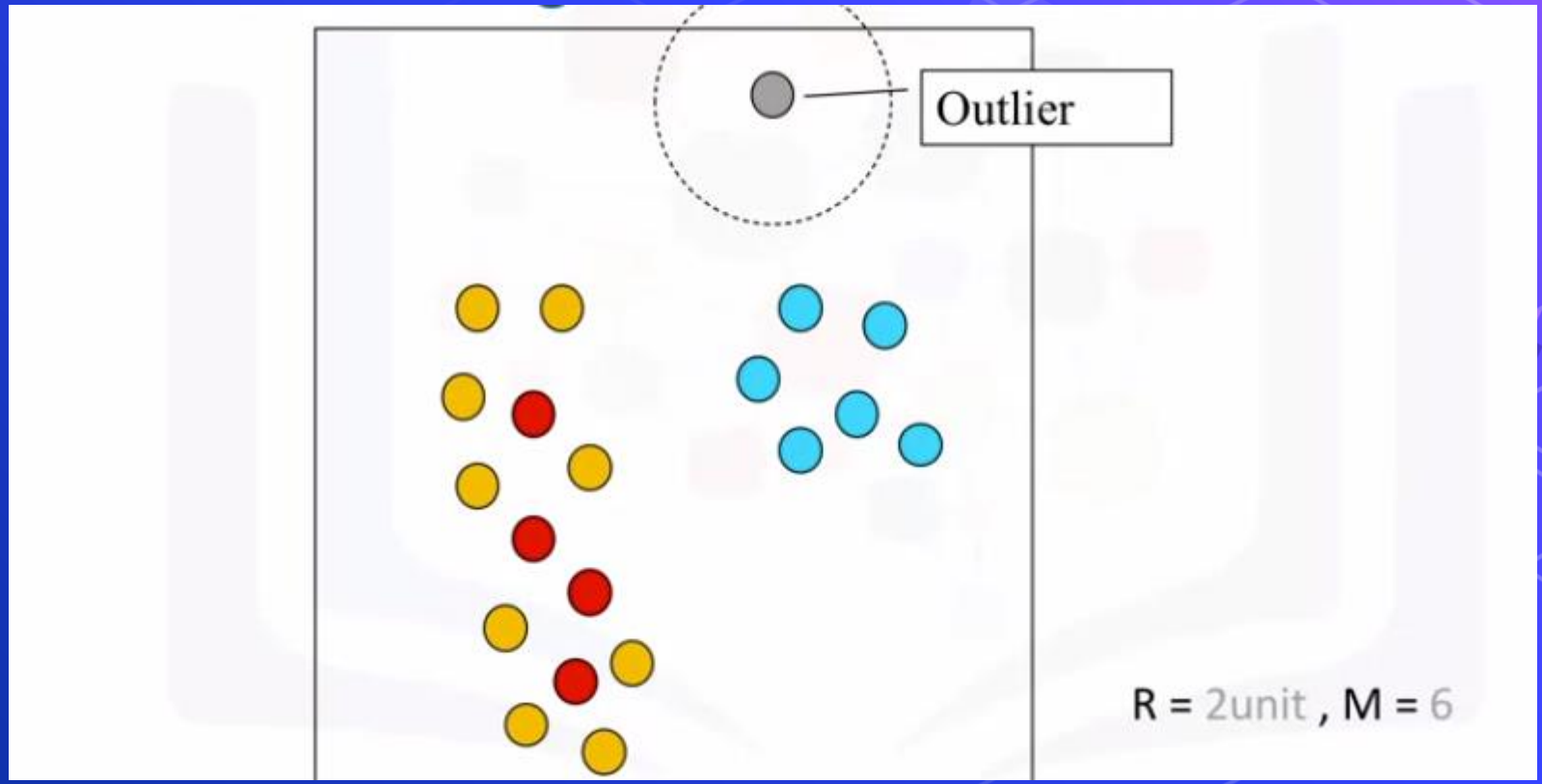
Density Based Clustering – DBSCAN (Border point)



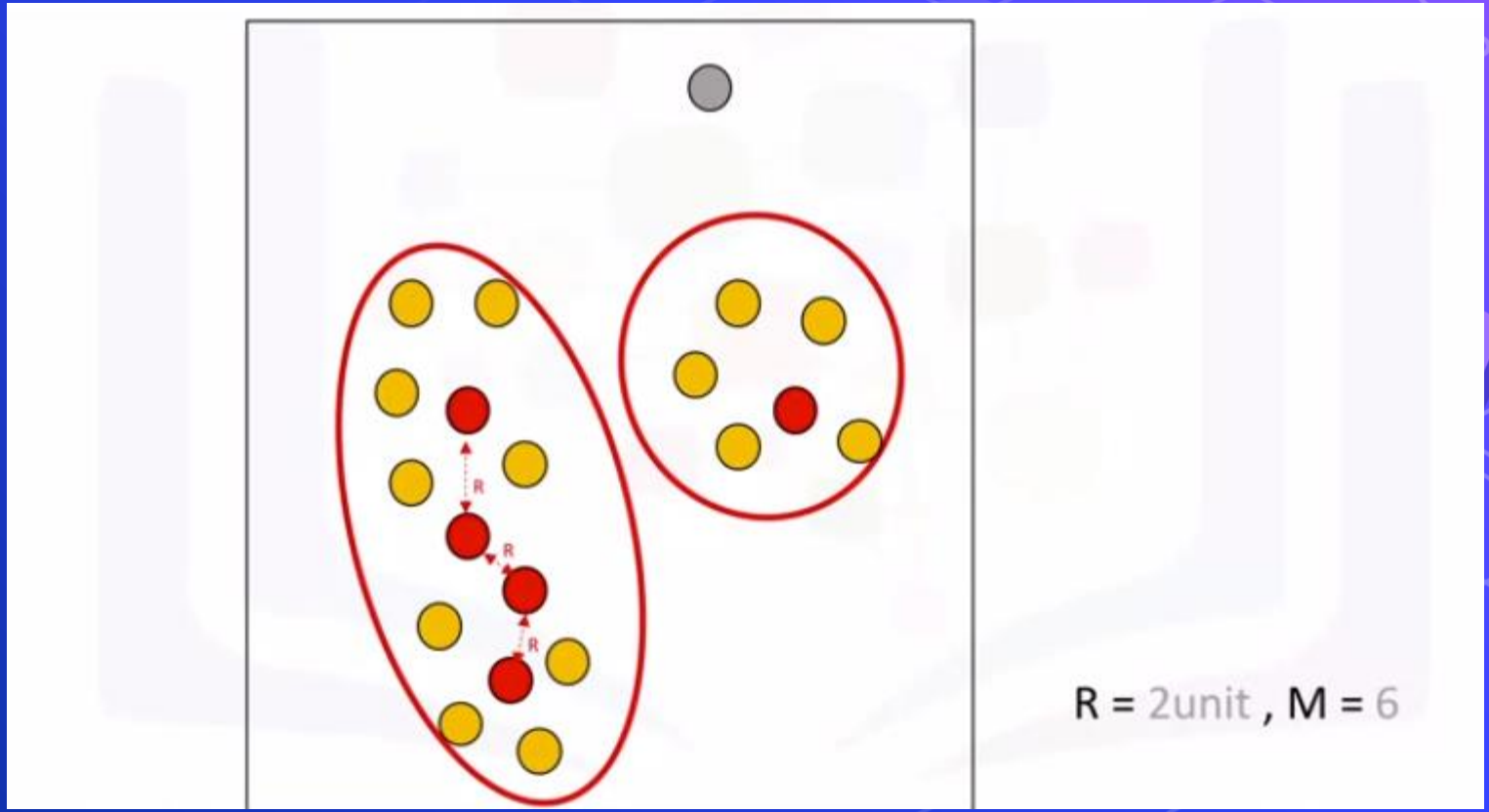
Density Based Clustering - DBSCAN



Density Based Clustering - DBSCAN



Density Based Clustering - DBSCAN

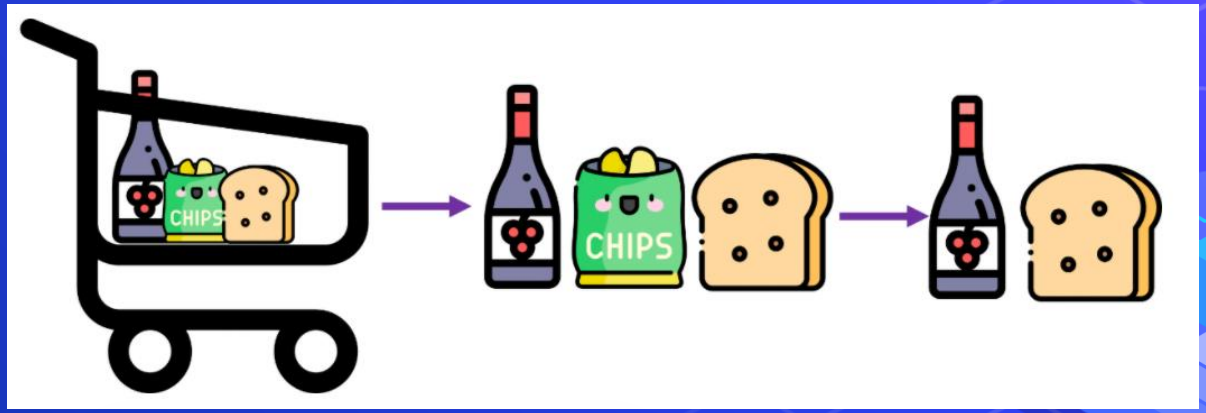


Density Based Clustering - DBSCAN

```
1 from sklearn.cluster import DBSCAN
2
3 model.DBSCAN(eps=0.3, min_samples=10)
4 y_labels = model.fit_predict(x)
```

- Machine Learning
- Supervised Learning
  - Regression
    - Simple Linear Regression
    - Multiple Linear Regression
    - Polynomial Regression
    - Evaluating Model Performance
  - Classification
    - Logistic Regression
    - K-Nearest Neighbors (KNN)
    - Naive Bayes
    - SVM
    - Decision Trees
    - Ensemble Methods
      - What is Bagging & Boosting
      - Random Forests
      - XGBoost
    - Evaluating Model Performance
- Unsupervised Learning
  - Clustering
    - KMeans
    - Hierarchical Clustering
    - Density Based Clustering - DBSCAN
  - Association rule mining
    - Apriori
  - Dimension Reduction
    - PCA
    - LDA
  - Evaluating Model Performance
- Model Selection & Evaluation
  - Cross Validation
  - Hyperparameter Tuning
    - Grid Search
    - Randomized Search
- Recommendation Systems

Apriori




```
1 >_ pip install apyori
```

Apriori

```
1 from apyori import apriori
2
3 records = []
4 for i in range(0, rows):
5     records.append([str(df.values[i,j]) for j in range(0, columns)])
6
7 association_rules = apriori(records)
8 association_results = list(association_rules)
```

Questions ?!

Thanks!

>_ Live long and prosper