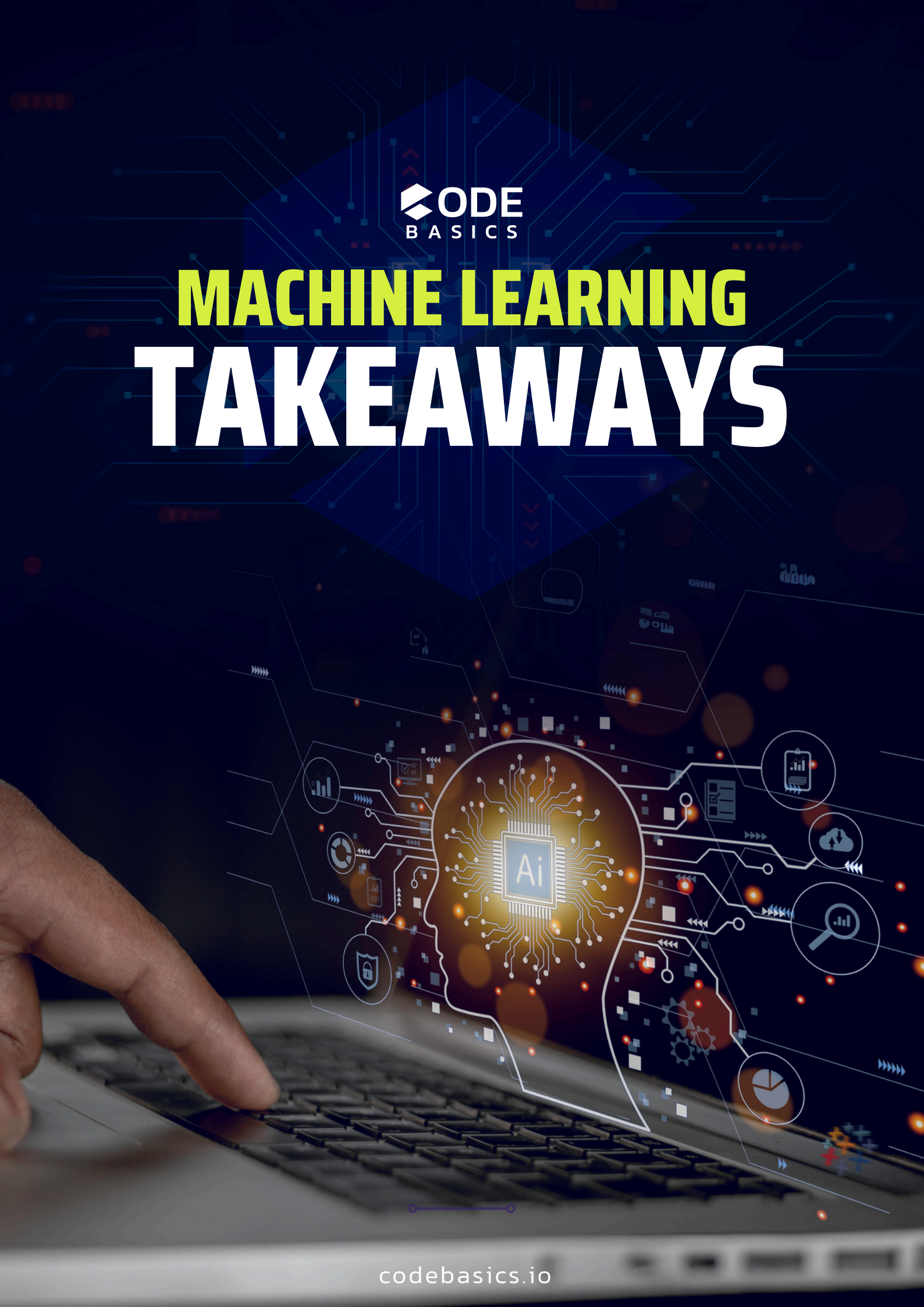
CODE BASICS

MACHINE LEARNING TAKEAWAYS



codebasics.io

Chapter:

Unsupervised Learning

K Means Clustering: Theory

1. K-means clustering is a method to group data into clusters where each piece of data is closest to the central point, or centroid, of its cluster.
    a. Start with K centroids by putting them at random place, Here k = 2 (random)
    b. Compute distance of every point from centroid and cluster them accordingly
    c. Adjust centroids so that they become center of gravity for given cluster
    d. Again re-cluster every point based on their distance with centroid
    e. Again adjust centroids
    f. Recompute clusters and repeat this till data points stop changing clusters

K Means Clustering: Theory

2 SSE – Sum of Squared Errors

3 To find the optimal number of clusters (k) using SSE, plot the sum of squared distances from each data point to its cluster's centroid. Then, select k where the decrease in SSE starts to level off, known as the "elbow point."

K Means Clustering: Customer Segmentation

1. K-means clustering might not perform well if the data is on different scales. It's recommended to preprocess the data and scale it using the min-max method or standard scaling method.

2. In real-world situations, having many features can make it difficult to determine the value of k. To address this, we compute SSE for each k and plot the elbow chart to find the optimal k.

3. In k-means, there's an API called inertia, which represents the sum of squared distances (errors).

Hierarchical Clustering: Theory

1. Hierarchical clustering is a technique that constructs a tree of clusters by grouping similar data points, beginning with individual points and progressively merging them into larger clusters.
    - Steps:
        i. Treat each data point as its own cluster.
        ii. Measure distances between clusters.
        iii. Merge the closest clusters.
        iv. Recalculate distances.
        v. Repeat until all data points form one cluster.
        vi. Optionally, you can create a dendrogram, a tree diagram showing merge sequences and distances.
    - Types:
        - Agglomerative Clustering – Popular one
        - Divisive Clustering

Hierarchical Clustering: Theory

2 Linkage methods in hierarchical clustering determine cluster distances and formation. These methods include:

a. Average Linkage: Average distance between all point pairs in two clusters.

b. Single Linkage: Shortest distance between any two points in different clusters.

c. Complete Linkage: Longest distance between any two points in different clusters.

d. Ward's Linkage: Minimizes the increase in within-cluster variance after merging.

Hierarchical Clustering: Customer Segmentation

1. A dendrogram is a tree-like diagram that illustrates the arrangement and merging levels of data points in hierarchical clustering.

2. The SciPy library provides a range of APIs specifically designed for Hierarchical Clustering.

DBSCAN: Theory

1. DBSCAN is a clustering algorithm that groups data points into clusters based on density; it identifies core points (with many neighbors), border points (fewer neighbors but close to a core point), and marks isolated points in low-density areas as outliers.
    - Steps:
        - Choose eps (max neighbor distance) and minPts (min points for a cluster).
        - Classify points with at least minPts within eps as core points.
        - Points within eps of core points but with fewer than minPts neighbors are border points.
        - Non-core and non-border points are outliers.
        - Form clusters by connecting core points and their neighbors, including border points.
        - Assign each point to a cluster or as an outlier.

DBSCAN: Theory

2 Benefits:

- Good at handling outliers
- No need to specify the number of clusters
- Faster compared to other clustering methods
- Good at handling weird shapes of data

DBSCAN: Practical Implementation

1. Best eps and min_samples can be figured out with trial and error and by examining the scatter plot for the DBSCAN method. This can be mastered with some practice.

2. For outlier detection, DBSCAN is a highly effective and straightforward solution.

3. Experiment with the parameters to determine the ones that best fit your dataset and use case.