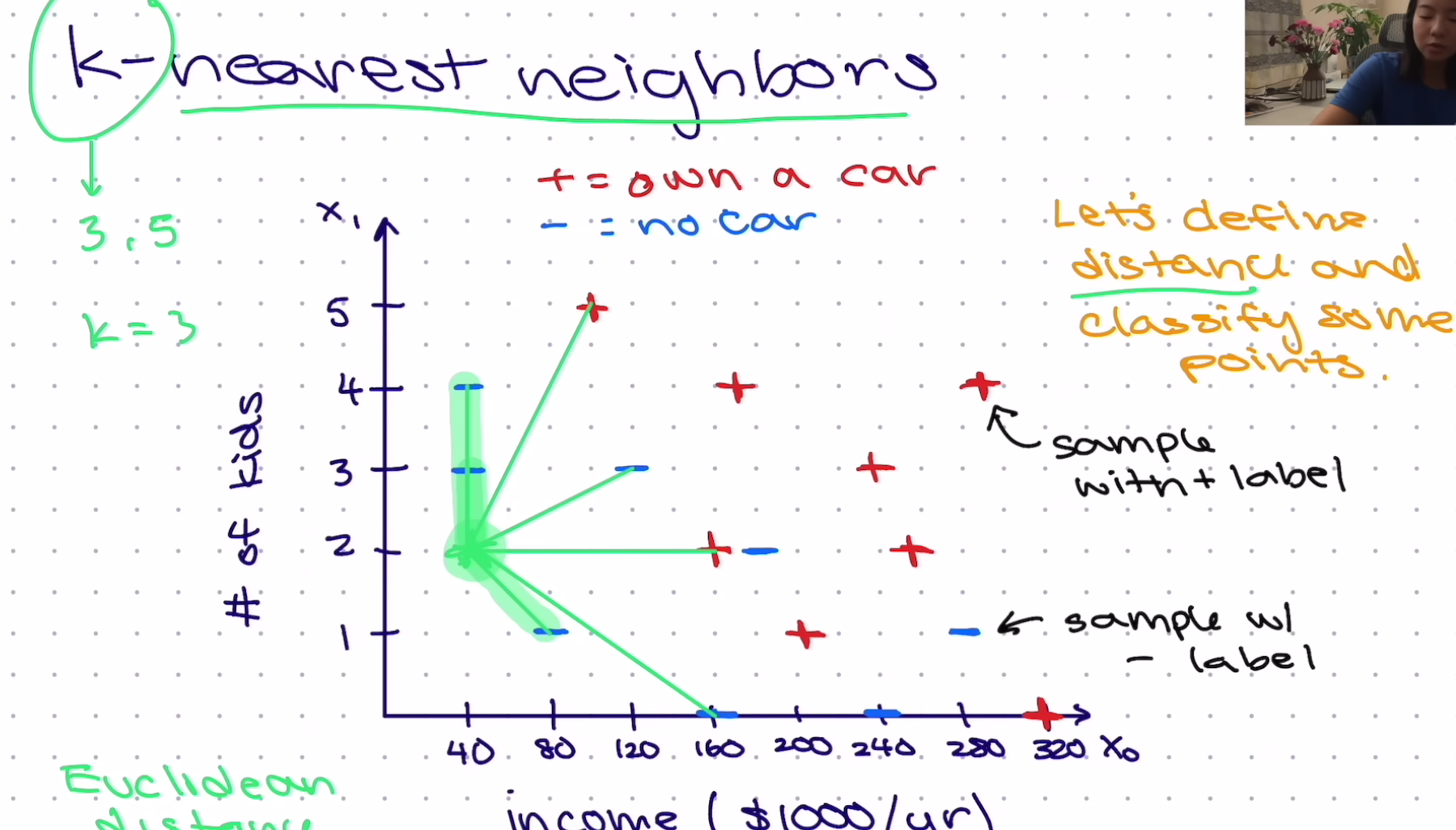
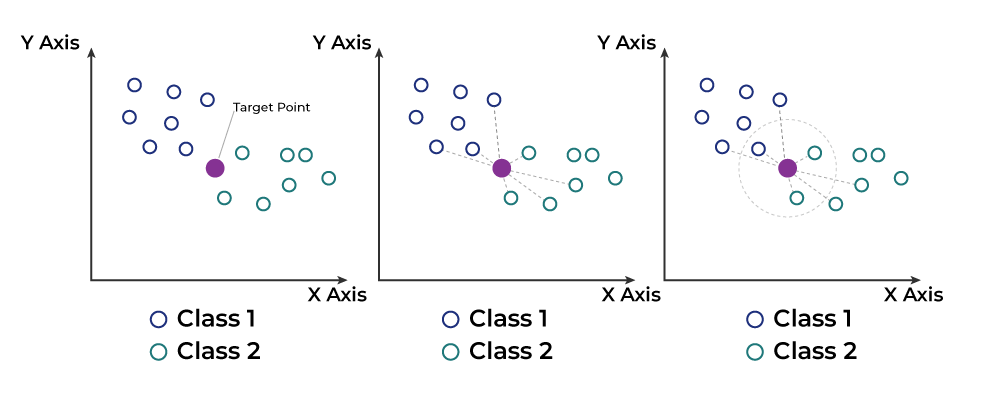
**The k-nearest neighbors (KNN) algorithm**





Euclidean



# Understanding KNN Algorithm

KNN operates on the principle of similarity, predicting the label or value of a new data point by examining the labels or values of its K nearest neighbors in the training dataset.

* **Steps Involved:**
  1. **Selecting Optimal K:** Determine the appropriate value for k, representing the number of nearest neighbors considered during prediction.
  2. **Calculating Distance:** Employ Euclidean distance to measure the similarity between the target data point and each data point in the dataset.
  3. **Identifying Nearest Neighbors:** The k data points with the smallest distances to the target point are recognized as the nearest neighbors.
  4. **Classification or Regression:**
     + *Classification:* Determine the class labels through majority voting among the neighbors, with the most frequent class becoming the predicted label.
     + *Regression:* Calculate the class label by averaging the target values of K nearest neighbors, with the resulting average value becoming the predicted output.

# Choosing the Value of K

* **Significance of K:** The value of k in KNN is pivotal as it determines the number of neighbors considered during classification or regression.
* **Data Consideration:** Selection of k should be based on the characteristics of the input data. If the data contains outliers or noise, a higher value of k is preferable.
* **Odd Value Preference:** To avoid ties in classification, it's recommended to choose an odd value for k.
* **Cross-Validation:** Utilizing cross-validation techniques aids in selecting the most suitable k value for the dataset.

# Step-by-Step Explanation of KNN Algorithm

1. **Selecting Optimal Value of K:**
   * Choose the value of k representing the number of neighbors to be considered during prediction.
2. **Calculating Distance:**
   * Utilize Euclidean distance to measure the similarity between the target data point and each data point in the dataset.
3. **Finding Nearest Neighbors:**
   * Identify the k data points with the smallest distances to the target point, which serve as the nearest neighbors.
4. **Voting for Classification or Taking Average for Regression:**
   * *Classification:* Determine class labels through majority voting among neighbors.
   * *Regression:* Calculate class label by averaging target values of K nearest neighbors.

# Disadvantages of the KNN Algorithm

* **Non-Scalability:**
  + KNN is often labeled as a Lazy Algorithm, indicating its high computational and storage requirements.
  + The algorithm demands significant computing power and data storage, resulting in time-consuming and resource-intensive operations.
* **Curse of Dimensionality:**
  + The peaking phenomenon, associated with the Curse of Dimensionality, affects KNN.
  + High-dimensional data poses challenges for the algorithm in accurately classifying data points, leading to decreased performance.
* **Prone to Overfitting:**
  + The Curse of Dimensionality also contributes to overfitting issues in KNN.
  + Overfitting occurs when the model learns noise in the training data, leading to poor generalization.
  + Mitigation strategies such as feature selection and dimensionality reduction are commonly employed to address this challenge effectively.

