Task ML(iti)

Classify for KNN

k-nearest neighbors(KNN) : is a simple supervised learning algorithm used for classification and sometimes regression

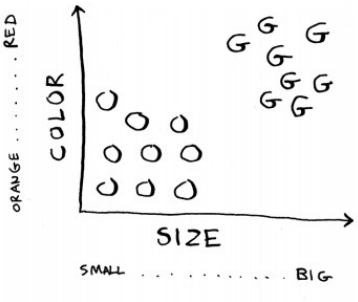
Main idea

To classify a new point, look at its K closest points in the training data and assign the label that is most common among them.

Example

If you have a fruit and you're not sure whether it's an orange or a grapefruit, you can look at its features (size and color).

- Grapefruits are usually larger and redder
- Oranges are smaller and less red

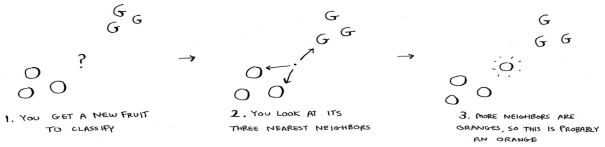


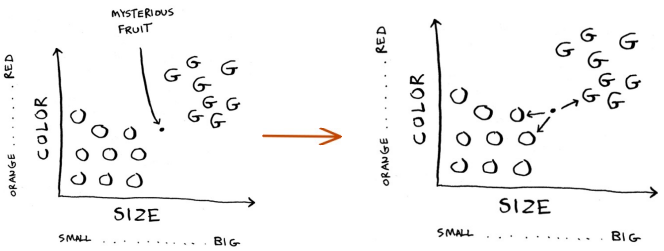
If the new fruit is big and red, you might guess it's a grapefruit

But if it's unclear, check the K nearest neighbors (for example, K=3).

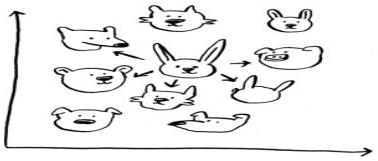
⟶ If most of them are oranges → the fruit is likely an orange

⟶ If most are grapefruits → the fruit is likely a grapefruit





Building a Recommendation System



Scenario Netflix wants to recommend movies to users

Finding Neighbors Plot each user on a graph based on similarity in taste users with similar preferences are closer together on the graph.

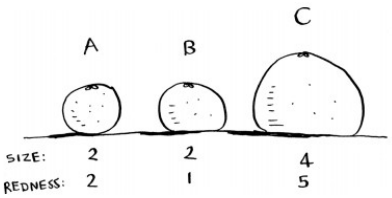
For a target user (e.g., Priyanka), find the 5 nearest users ⟶ Example: Justin, JC, Joey, Lance, and Chris

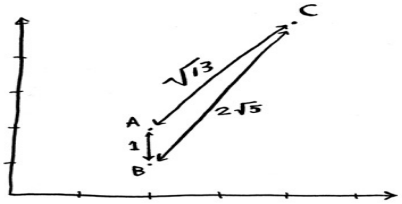
Since they have similar movie tastes, recommend to Priyanka the movies they like

Remaining Challenge how to measure similarity between users to place them correctly on the graph?

Feature extraction In the fruit example, size and color are the features used for comparison Each fruit can be represented by a set of numbers (features) By graphing the fruits, you can visually see similarities

→ To measure similarity precisely, use the Pythagorean distance formula Example: Distance between fruit A and B is 1, confirming they are similar





Regression for KNN Regression = Predicting a numerical value (e.g., rating, quantity)

Works like KNN classification but instead of voting for a category, you average the values of the K nearest neighbors. K is flexible (can be 2, 5, 10, etc.)

Example A small bakery in Berkeley bakes fresh bread daily and wants to predict how many loaves to make today.

Features Used

1. Weather (scale 1–5, where 1 = bad, 5 = great)
2. Weekend/Holiday (1 if yes, 0 if no)
3. Game Day (1 if yes, 0 if no)

A. (5, 1, ∅) = 300 LOAVES B. (3, 1, 1) = 225 LOAVES

C. (1, 1, ∅) = 75 LOAVES D. (4, ∅, 1) = 200 LOAVES

E. (4, ∅, ∅) = 150 LOAVES F. (2, ∅, ∅) = 50 LOAVES

Approach (KNN Regression, k = 4)

Use historical sales data with the same features. Today's conditions: Weekend + Good Weather.

Find the 4 closest past days (A, B, D, E) based on feature similarity (4, 1, ∅) = ?

Prediction Average the loaves sold on those 4 days. Result = 218.75 loaves → bake about 219 loaves today.

A. 1 ←
B. 2 ←
C. 9
D. 2 ←
E. 1 ←
F. 5

Picking good features

A feature is a measurable property or characteristic used by a machine learning model to make predictions or recommendations

Key Idea In KNN, choosing the right features is critical for accurate predictions/recommendations.

Bad features → Poor performance. Good features → Strong, relevant predictions.