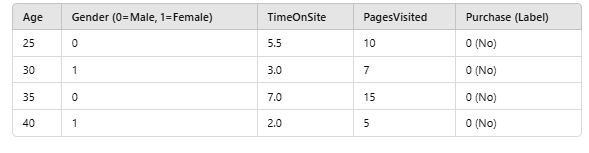
**Problem: Predicting Whether a Customer Will Buy a Product**

**Scenario:** A retail company wants to predict whether a customer visiting their online store will purchase a product based on their browsing history, time spent on the site, and other behavioral features.

**Dataset Features:**

* **Age**: Age of the customer.
* **Gender**: Gender of the customer.
* **Time on Site**: Time spent browsing the website (in minutes).
* **Pages Visited**: Number of pages visited during the session.

**Sample Data :**



import java.util.ArrayList;

import java.util.Comparator;

import java.util.List;

class Customer {

double age;

double gender; // 0: Male, 1: Female

double timeOnSite;

double pagesVisited;

int purchase; // 0: No, 1: Yes

public Customer(double age, double gender, double timeOnSite, double pagesVisited, int purchase) {

this.age = age;

this.gender = gender;

this.timeOnSite = timeOnSite;

this.pagesVisited = pagesVisited;

this.purchase = purchase;

}

}

class KNN {

private List<Customer> trainingData;

public KNN(List<Customer> trainingData) {

this.trainingData = trainingData;

}

// Calculate Euclidean Distance

private double calculateDistance(Customer a, Customer b) {

return Math.sqrt(

Math.pow(a.age - b.age, 2) +

Math.pow(a.gender - b.gender, 2) +

Math.pow(a.timeOnSite - b.timeOnSite, 2) +

Math.pow(a.pagesVisited - b.pagesVisited, 2)

);

}

public int predict(Customer newCustomer, int k) {

List<Customer> neighbors = new ArrayList<>(trainingData);

// Sort neighbors by distance to the new customer

neighbors.sort(Comparator.comparingDouble(c -> calculateDistance(c, newCustomer)));

// Count purchases in top k neighbors

int purchaseYes = 0;

int purchaseNo = 0;

for (int i = 0; i < k; i++) {

if (neighbors.get(i).purchase == 1) {

purchaseYes++;

} else {

purchaseNo++;

}

}

// Majority voting

return purchaseYes > purchaseNo ? 1 : 0;

}

}

public class KNNExample {

public static void main(String[] args) {

// Training Data

List<Customer> trainingData = new ArrayList<>();

trainingData.add(new Customer(25, 0, 5.5, 10, 1)); // Male, purchased

trainingData.add(new Customer(30, 1, 3.0, 7, 0)); // Female, not purchased

trainingData.add(new Customer(35, 0, 7.0, 15, 1)); // Male, purchased

trainingData.add(new Customer(40, 1, 2.0, 5, 0)); // Female, not purchased

/\* Age Gender (0=Male, 1=Female) TimeOnSite PagesVisited Purchase (Label)

25 0 5.5 10 0 (No)

30 1 3.0 7 0 (No)

35 0 7.0 15 0 (No)

40 1 2.0 5 0 (No)\*/

// Create KNN model

KNN knn = new KNN(trainingData);

// New customer data

Customer customer1 = new Customer(28, 0, 4.5, 9, 0); // Predict purchase (purchase label is unknown)

Customer customer2 = new Customer(28, 0, 2, 5, 0);

// Predict using k = 3

int k = 3;

int prediction1 = knn.predict(customer1, k);

int prediction2 = knn.predict(customer2, k);

// Output prediction

System.out.println("Customer1 Prediction: " + (prediction1 == 1 ? "Purchase" : "No Purchase"));

System.out.println("Customer2 Prediction: " + (prediction2 == 1 ? "Purchase" : "No Purchase"));

}

}