**Machine Learning Model for Learning Trait Similarities**

const calculateJaccardSimilarity = require('./calculateJaccardSimilarity');

// Sample employee data

const employee1 = {

name: "John Doe",

hard\_skills: ["JavaScript", "React", "Node.js", "MongoDB"],

soft\_skills: ["Communication", "Teamwork", "Problem Solving"],

spiritual\_gifts: ["Teaching", "Leadership", "Service"]

};

const employee2 = {

name: "Jane Smith",

hard\_skills: ["JavaScript", "React", "Python", "SQL"],

soft\_skills: ["Communication", "Teamwork", "Creativity"],

spiritual\_gifts: ["Service", "Mercy", "Encouragement"]

};

// Calculate Jaccard similarity between the two employees

const similarityScore = calculateJaccardSimilarity(employee1, employee2);

console.log("Overall similarity score:", similarityScore);

// Track similarity for individual traits

const traitSimilarity = {

hard\_skills: calculateJaccardSimilarity(employee1.hard\_skills, employee2.hard\_skills),

soft\_skills: calculateJaccardSimilarity(employee1.soft\_skills, employee2.soft\_skills),

spiritual\_gifts: calculateJaccardSimilarity(employee1.spiritual\_gifts, employee2.spiritual\_gifts)

};

console.log("Trait similarity:");

console.log(traitSimilarity);

// This model first calculates the overall similarity score between the two employees using the

// calculateJaccardSimilarity function. Then, it tracks the similarity for individual traits (hard skills, soft skills,

// spiritual gifts) separately and logs the results. Adjust the employee data as needed and analyze the trait similarities

// to identify which traits have higher similarity ratings between the two employees.