Nome: Fernanda Fernandes Borges

Período: 8

Inteligência Artificial

O que é KNN?

R: KNN: K-nearest-neighbor (K-vizinho-próximo)

Utilizado para classificar objetos com base em exemplos de treinamento que estão mais próximos no espaço de características;

O parâmetro k é o número de vizinhos mais próximos.

Para utilizar o KNN é necessário: – Um conjunto de exemplos para treinamento; – Definir uma métrica para calcular a distância entre os exemplos de treinamento; – Definir o valor de K.

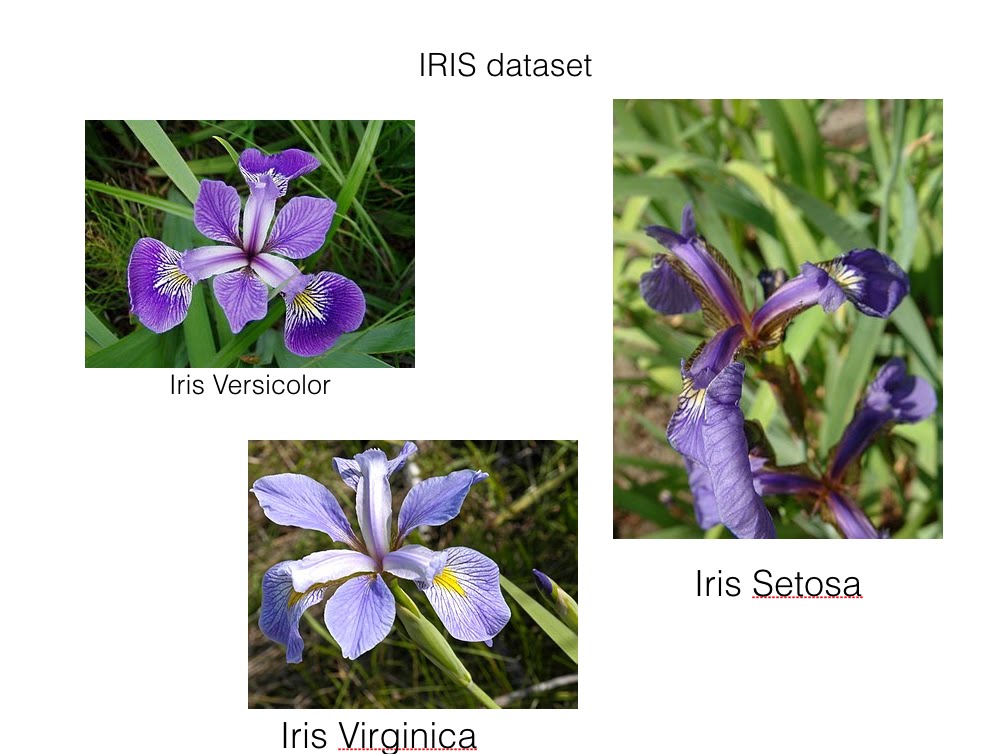
Classificar um exemplo desconhecido com o algoritmo KNN consiste em: – Calcular a distância entre o objeto desconhecido e os outros exemplos do conjunto de treinamento; – Identificar os K vizinhos mais próximos; – Utilizar o “rótulo” (nome) da classe dos vizinhos mais próximos para determinar o rótulo de classe do exemplo desconhecido (votação majoritária).

Calculo da distância entre dois pontos: – A mais simples é a distancia euclidiana: Outras formas de medir distâncias: – Distância de Mahalanobis, Minkowsky, Hamming, entre outros.

Como escolher o valor de K? – Se K for muito pequeno, a classificação fica sensível a pontos de ruído; – Se k for elevado, a vizinhança pode incluir elementos de outras classes. \*normalmente, escolhe-se um valor ímpar para K (evita empates desnecessários no cálculo).

O conjunto de dados de flores de Iris ou o conjunto de dados de Fisher Iris é um conjunto de dados multivariados introduzido pelo estatístico britânico e biologista Ronald Fisher em seu artigo de 1936. O uso de medidas múltiplas em problemas taxonômicos como exemplo de análise discriminante linear . Às vezes é chamado de conjunto de dados Iris de Anderson porque Edgar Anderson coletou os dados para quantificar a variação morfológica de flores de íris de três espécies relacionadas. Duas das três espécies foram coletadas na Península de Gaspé "todas da mesma pastagem, e colhidas no mesmo dia e medidas ao mesmo tempo pela mesma pessoa com o mesmo aparelho".

O conjunto de dados consiste em 50 amostras de cada uma das três espécies de Iris setosa , Iris virginica e Iris versicolor ). Foram medidas quatro características de cada amostra: o comprimento e a largura das sépalas e pétalas , em centimetros. Com base na combinação destas quatro características, Fisher desenvolveu um modelo discriminante linear para distinguir as espécies umas das outras.



Este trabalho foi desenvolvido um programa, que através de um algoritmo estruturado têm o objetivo de identificar novas flores .O sistema classificará as novas flores, para isso foi utilizado um Iris Dataset que é um banco de dados com as informações sobre 3 tipos de flores.

-Setosa -Virginica -Setosa

Para isso foi utilizado arquivos como o Iris Dataset que foi dividido em duas partes. Sendo que uma seria usado como base de treinamento e uma base de teste. O programa irá receber os dois arquivos e salvar em duas listas para que possa ser utilizado durante o cálculo.

Para fazer o cálculo da distância foi utilizado a fórmula Euclidiana para o cálculo. Onde você terá dois loops sendo que o primeiro será da base de teste e o segundo loop interno será o de treinamento. Dentro do loop interno será feito o cálculo, onde será comparado cada posição da lista de testes com cada posição da lista de treinamento. No final esse resultado será salvo na tabela de treinamento.Antes de interar o loop da lista de teste, ele irá ordenar a lista e irá classificar conforme o k digitado no inicio da execução conforme o resultado de K será possível classificar a classe do objeto testado.

O código da flor que pega todos os tamanhos da flor.

public class Flor {

public double [] Features;

public String Name;

public Flor(){}

//lista.add(Double.parseDouble(Classifier(flores, flor\_teste0)));

//comprimento sepala, largura da sepala, comprimento da sepala , largura da sepala, nome

Flor(String name, double sepalLength, double sepalWidth, double petalLength, double petalWidth){

Features = new double[4];

this.Name = name;

Features[0] = sepalLength;

Features[1] = sepalWidth;

Features[2] = petalLength;

Features[3] = petalWidth;

}

}

E o código do Main da classe principal.

public static void main(String[] args) {

// TODO code application logic here

List<Flor> flores = new ArrayList<Flor>();

List<Double> lista = new ArrayList<Double>();

List<Flor> lista\_itens = new ArrayList<Flor>();

//IRIS SETOSA

flores.add(new Flor("Iris-setosa",5.1 //comprimento sepala

,3.5 //largura da sepala

,1.4 //comprimento da sepala

,0.2 ));//largura da sepala

flores.add(new Flor("Iris-setosa",4.9,3.0,1.4,0.2 ));

flores.add(new Flor("Iris-setosa",4.7,3.2,1.3,0.2 ));

flores.add(new Flor("Iris-setosa",4.6,3.1,1.5,0.2 ));

// flores.add(new Flor("Iris-setosa",5.0,3.6,1.4,0.2 ));

// flores.add(new Flor("Iris-setosa",5.4,3.9,1.7,0.4 ));

// flores.add(new Flor("Iris-setosa",4.6,3.4,1.4,0.3 ));

// flores.add(new Flor("Iris-setosa",5.0,3.4,1.5,0.2 ));

// flores.add(new Flor("Iris-setosa",4.4,2.9,1.4,0.2 ));

//IRIS VERSICOLOR

flores.add(new Flor("Iris- Versicolor", 5.5, 2.4, 3.7, 1.0));

flores.add(new Flor("Iris- Versicolor", 5.8, 2.7, 3.9, 1.2));

flores.add(new Flor("Iris- Versicolor", 6.0, 2.7, 5.1, 1.6));

flores.add(new Flor("Iris- Versicolor", 5.4, 3.0, 4.5, 1.5));

// flores.add(new Flor("Iris- Versicolor", 6.0, 3.4, 4.5, 1.6));

// flores.add(new Flor("Iris- Versicolor", 6.7, 3.1, 4.7, 1.5));

// flores.add(new Flor("Iris- Versicolor", 6.3, 2.3, 4.4, 1.3));

// flores.add(new Flor("Iris- Versicolor", 5.6, 3.0, 4.1, 1.3));

// flores.add(new Flor("Iris- Versicolor", 5.5, 2.5, 4.0, 1.3));

//IRIS VIRGINICA

flores.add(new Flor("Iris- Virginica", 6.1, 3.0, 4.9, 1.8));

flores.add(new Flor("Iris- Virginica", 6.4, 2.8, 5.6, 2.1));

flores.add(new Flor("Iris- Virginica", 7.4, 2.8, 6.1, 1.9));

flores.add(new Flor("Iris- Virginica", 7.9, 3.8, 6.4, 2.0));

// flores.add(new Flor("Iris- Virginica", 7.2, 3.0, 5.8, 1.6));

// flores.add(new Flor("Iris- Virginica", 7.7, 3.0, 6.1, 2.3));

// flores.add(new Flor("Iris- Virginica", 6.3, 3.4, 5.6, 2.4));

// flores.add(new Flor("Iris- Virginica", 6.4, 2.8, 5.6, 2.2));

// flores.add(new Flor("Iris- Virginica", 6.3, 2.8, 5.1, 1.5));

// double [] textoParaOrdenar;

// //IRIS SETOSA

// //double[] flor\_teste1 = {5.1, 3.5, 1.4,0.2};

double[] flor\_teste0 = new double[4];

flor\_teste0[0] = 4.8;

flor\_teste0[1] = 3.5;

flor\_teste0[2] = 1.4;

flor\_teste0[3] = 0.2;

//

// Arrays.sort(flor\_teste0);

//

// for(double i: flor\_teste0){

// System.out.println(" Testando"+i);

// }

//

// double[] flor\_teste2 = new double[4];

// flor\_teste2[0] = 4.7;

// flor\_teste2[1] = 3.2;

// flor\_teste2[2] = 1.3;

// flor\_teste2[3] = 0.2;

//

// Arrays.sort(flor\_teste2);

// //double[] flor\_teste2 = {4.7, 3.2, 1.3,0.2};

//

// for(double i: flor\_teste2){

// System.out.println(" Testando"+i);

// }

// double[] flor\_teste1 = new double[4];

// flor\_teste1[0] = 5.1;

// flor\_teste1[1] = 3.5;

// flor\_teste1[2] = 1.4;

// flor\_teste1[3] = 0.2;

//

// Arrays.sort(flor\_teste1);

//

// for(double i: flor\_teste1){

// System.out.println(" Testando"+i);

// }

////

// int k = 5;

// System.out.println("verificando K = " + k);

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

// if (flor\_teste0[i] != 0.0) {

// System.out.println(" Resultado " + flor\_teste0[i]);

// } else {

// k = k ++;

// }

//// }

//double[] flor\_teste0 = {5.1, 3.5, 1.4,0.2};

double[] flor\_teste1 = {5.1, 3.5, 1.4,0.2};

double[] flor\_teste2 = {4.7, 3.2, 1.3,0.2};

double[] flor\_teste3 = {4.8, 3.0, 1.3,0.2};

Nesse arquivo ele tem os arquivos de base e faz a comparação com os dados de teste. E faz a conta das distâncias de cada arquivo de teste em relação a base e com base nessas distâncias ele descobre qual será o knn. Ou seja ele pega o elemento e calcula os seus vizinhos mais próximos e com base nisso descobre qual será o elemento de teste sua classificação no caso qual dos três tipos será a flor(Iris-Setosa, Iris-Versicolor, Iris Virginica). Depois dessa classificação ele descobre o elemento de teste.