



Sistema de Classificação de Animais com KNN



Eduardo



Guilherme



Nadianne



Rafael



Valdir



Verenilson



Nadianne

Apresentação do problema



Aquático

Altura média

2 patas

Pena

Tom de pele Escura

Noturno

Baixo

Velocidade Baixa



Guilherme



Introdução ao KNN

- Algoritmo de aprendizado de máquina simples e intuitivo
- Objetos com características semelhantes tendem a estar próximos uns dos outros no espaço de características.



Eduardo



Por que utilizamos o KNN para esse projeto?

- Lida bem com Diversidade de Espécies
- Preserva as Relações entre Características
- Tolerância a Mudanças nos Dados



Valdir

Entradas de Dados

Nome	Habitat	Revestimento	Tonalidade	Tamanho	Velocidade	Nº Patas	Hab. N	Dieta
Leão	Terrestre	Pelo	Escuro	Médio	Alta	4	Não	Carnívoro
Águia	Aéreo	Pena	Claro	Médio	Alta	2	Não	Carnívoro
Hipopótamo	Aquático	Pele	Claro	Grande	Baixa	4	Não	Herbívoro
Lagarto	Terrestre	Escama	Claro	Pequeno	Média	4	Não	Onívoro
Lince	Terrestre	Pelo	Escuro	Médio	Alta	4	Sim	Carnívoro
Pinguim	Aquático	Pena	Claro	Médio	Baixa	2	Não	Carnívoro
Crocodilo	Terrestre	Escama	Escuro	Grande	Alta	4	Não	Carnívoro
Tigre	Terrestre	Pelo	Escuro	Grande	Alta	4	Não	Carnívoro
Girafa	Terrestre	Pelo	Claro	Grande	Média	4	Não	Herbívoro
Elefante	Terrestre	Pelo	Claro	Grande	Baixa	4	Não	Herbívoro
Rinoceronte	Terrestre	Pelo	Claro	Grande	Baixa	4	Não	Herbívoro

```

public List<Animal> animais = new List<Animal>
{
    new Animal("Leao", "Terrestre", "Pelo", true, "Médio", "Alta", 4, false, "Carnívoro"),
    new Animal("Águia", "Aéreo", "Pena", false, "Médio", "Alta", 2, false, "Carnívoro"),
    new Animal("Hipopótamo", "Aquático", "Pele", false, "Grande", "Baixa", 4, false, "Herbívoros"),
    new Animal("Lagarto", "Terrestre", "Escama", false, "Pequeno", "Média", 4, false, "Onívoro"),
    new Animal("Lince", "Terrestre", "Pelo", true, "Médio", "Alta", 4, true, "Carnívoro"),
    new Animal("Pinguim", "Aquático", "Pena", false, "Médio", "Baixa", 2, false, "Carnívoro"),
    new Animal("Crocodilo", "Aquático", "Escama", true, "Grande", "Alta", 4, false, "Carnívoro"),
    new Animal("Tigre", "Terrestre", "Pelo", true, "Grande", "Alta", 4, false, "Carnívoro"),
    new Animal("Girafa", "Terrestre", "Pelo", false, "Grande", "Média", 4, false, "Herbívoros"),
    new Animal("Elefante", "Terrestre", "Pelo", false, "Grande", "Baixa", 4, false, "Herbívoros"),
    new Animal("Rinoceronte", "Terrestre", "Pelo", false, "Grande", "Baixa", 4, false, "Herbívoros"),
    new Animal("Ornitorrinco", "Aquático", "Pelo", true, "Pequeno", "Baixa", 4, true, "Onívoro"),
    new Animal("Capivara", "Terrestre", "Pelo", false, "Médio", "Baixa", 4, false, "Herbívoros"),
    new Animal("Lobo", "Terrestre", "Pelo", true, "Médio", "Alta", 4, false, "Carnívoro"),
    new Animal("Gorila", "Terrestre", "Pelo", true, "Grande", "Baixa", 2, false, "Herbívoros"),
    new Animal("Raposa", "Terrestre", "Pelo", false, "Médio", "Alta", 4, false, "Carnívoro"),
    new Animal("Bixo-preguiça", "Terrestre", "Pelo", true, "Médio", "Baixa", 4, true, "Herbívoros"),
    new Animal("Panda", "Terrestre", "Pelo", false, "Médio", "Baixa", 4, false, "Herbívoros"),
    new Animal("Garça", "Aquático", "Pena", false, "Médio", "Média", 2, false, "Carnívoro"),
    new Animal("Flamingo", "Aquático", "Pena", false, "Grande", "Média", 2, false, "Herbívoros"),
    new Animal("Aracnídeo", "Terrestre", "Escama", false, "Pequeno", "Baixa", 8, false, "Carnívoro"),
    new Animal("Besouro", "Terrestre", "Escama", false, "Pequeno", "Baixa", 6, false, "Herbívoros"),
    new Animal("Cobra", "Terrestre", "Escama", false, "Médio", "Média", 0, false, "Carnívoro"),
    new Animal("Escorpião", "Terrestre", "Escama", false, "Pequeno", "Média", 10, false, "Carnívoro")
};//

```

Nome	Habitat	Revest	Ton	Tam	Vel	NºP	Hab.N	Dieta
------	---------	--------	-----	-----	-----	-----	-------	-------



Rafael

Normalização das Entradas

Nome	Habitat	Revestimento	Tonalidade	Tamanho	Velocidade	Nº Patas	Hab. N	Dieta
Leão	0	0	1	0.5	1	4	0	0
Águia	1	0.66	0	0.5	1	2	0	0
Hipopótamo	0.5	1	0	1	0	4	0	0.5
Lagarto	0	0.33	0	0	0.5	4	0	1
Lince	0	0	1	0.5	1	4	1	0
Pinguim	0.5	0.66	0	0.5	0	2	0	0
Crocodilo	0	0.33	1	1	1	4	0	0
Tigre	0	0	1	1	1	4	0	0
Girafa	0	0	0	1	0.5	4	0	0.5
Elefante	0	0	0	1	0	4	0	0.5
Rinoceronte	0	0	0	1	0	4	0	0.5



Rafael

Normalização das Entradas

```
if (Habitat == "Terrestre")
    HabitatNormalized = 0f;
else if (Habitat == "Aquático")
    HabitatNormalized = 0.5f;
else //Aereo
    HabitatNormalized = 1f;

if (Revestimento == "Pelo")
    RevestimentoNormalized = 0;
else if (Revestimento == "Escama")
    RevestimentoNormalized = 0.33f;
else if (Revestimento == "Pena")
    RevestimentoNormalized = 0.66f;
else //Pele lisa
    RevestimentoNormalized = 1f;
```

```
if (Tamanho == "Pequeno")
    TamanhoNormalized = 0f;
else if (Tamanho == "Médio")
    TamanhoNormalized = 0.5f;
else //Grande
    TamanhoNormalized = 1f;

if (Velocidade == "Baixa")
    VelocidadeNormalized = 0f;
else if (Velocidade == "Média")
    VelocidadeNormalized = 0.5f;
else
    VelocidadeNormalized = 1f;
```




Rafael

Normalização das Entradas

```
if (alimentacao == "Carnívoro")  
    AlimentacaoNormalized = 0f;  
else if (alimentacao == "Herbívoros")  
    AlimentacaoNormalized = 0.5f;  
else  
    AlimentacaoNormalized = 1f;
```

```
//0 = claro | 1 = escuro
```

```
TonalidadePeleNormalized = TonalidadePele ? 0 : 1;
```

```
//1 para Noturno 0 para Não Noturno
```

```
HabitoNoturnoNormalized = HabitoNoturno ? 1 : 0;
```



Verenilson

Normalização

```
int maxPatas = animais.Max(animal => animal.NumeroDePatas);
int minPatas = animais.Min(animal => animal.NumeroDePatas);
maxPatas = newAnimal.NumeroDePatas > maxPatas ? newAnimal.NumeroDePatas : maxPatas;
minPatas = newAnimal.NumeroDePatas < minPatas ? newAnimal.NumeroDePatas : minPatas;
newAnimal.NumeroDePatasNormalized = ((float)newAnimal.NumeroDePatas - minPatas) / (maxPatas - minPatas);
foreach (Animal animal in animais)
{
    animal.NumeroDePatasNormalized = ((float)animal.NumeroDePatas - minPatas) / (maxPatas - minPatas);
    animal.Proximidade = Math.Sqrt((
        Math.Pow(animal.HabitatNormalized - newAnimal.HabitatNormalized, 2) +
        Math.Pow(animal.RevestimentoNormalized - newAnimal.RevestimentoNormalized, 2) +
        Math.Pow(animal.TonalidadePeleNormalized - newAnimal.TonalidadePeleNormalized, 2) +
        Math.Pow(animal.TamanhoNormalized - newAnimal.TamanhoNormalized, 2) +
        Math.Pow(animal.VelocidadeNormalized - newAnimal.VelocidadeNormalized, 2) +
        Math.Pow(animal.HabitoNoturnoNormalized - newAnimal.HabitoNoturnoNormalized, 2) +
        Math.Pow(animal.NumeroDePatasNormalized - newAnimal.NumeroDePatasNormalized, 2)
    ));
}
animais.Sort((animal1, animal2) => animal1.Proximidade.CompareTo(animal2.Proximidade));
```



Verenilson



Algoritmo KNN

Vamos ver na prática!

Gratos pela atenção!

