

## Inteligência Artificial

Thiago Henrique Leite da Silva, RA: 139920

### AULA9: Exercício teórico aprendizado supervisionado

1) (0,2) Aplique o KNN com  $K = 3$  para classificar o 11o elemento da tabela. Considere a distância Euclidiana e Manhattan, compare e há diferença entre elas.

	A1	A2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	5	7	?

Para resolver este item, implementei um algoritmo Knn em Java que seguirá anexo a atividade no Classroom. O algoritmo recebe como parâmetros um conjunto de elementos para treino, sendo fornecidos por meio de dois arrays de atributos double e um array de inteiros com a classe do respectivo elemento.

Feito isso, temos como treinar nosso algoritmo de duas formas, com a distância de manhattan e com a distância euclidiana:

```
public Double euclideanDistance(int i, double a, double b) {  
    return Math.pow(attributeA1[i] - a, 2) + Math.pow(attributeA2[i] - b, 2);  
}  
  
public Double manhattanDistance(int i, double a, double b) {  
    return Math.abs(attributeA1[i] - a) + Math.abs(attributeA2[i] - b);  
}
```

Terminado o treino, inserimos os atributos do elemento que queremos descobrir a classe. O resultado obtido foi o seguinte:

```

Console
<terminated> KnnAlgorithm (1) [Java Application] /usr/lib/jvm/jre1
Isira os atributos do elemento que deseja classificar:

Atributo A1: 5
Atributo A2: 7

Treinando algoritmo com distância euclideana:
Espécie: 1 >> {1=2, 2=1}

Treinando algoritmo com distância de manhattan:
Espécie: 1 >> {1=3, 2=0}

```

O hash retornado é da seguinte forma:

{classe => quantidade de ocorrências desta classe no vetor com as K menores distâncias}

Portanto, ambos os treinos nos forneceram o mesmo resultado, a diferença foi que com a distância de Manhattan, as três menores distâncias foram da Classe 1, enquanto com a distância euclidiana, tivemos dois elementos da Classe 1 e um elemento da Classe 2, dando a impressão que a distância de manhattan tem mais certeza de seu palpite. Porém, por se tratar de um teste particular, não nos diz muita coisa.

2) (0,2) Aplique o algoritmo de Bayes no problema a seguir:

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	???

Give Birth			Can Fly			Live in Water		
	mammals	non		mammals	non		mammals	non
yes	6	1	yes	1	3	yes	2	3
no	1	11	no	6	10	no	5	6
						sometimes	0	4
yes	6/7	12/13	yes	1/7	3/13	yes	2/7	3/13
no	1/7	1/13	no	6/7	10/13	no	5/7	6/13
						sometimes	0/7	4/13

	Have Legs		Class	
	mammals	non	mammals	non
yes	5	4	7	13
no	2	9		
yes	5/7	4/13	7/20	13/20
no	2/7	9/13		

Obs. O denominador foi omitido pois é sempre igual.

$P(\text{mammals} | \text{give birth, can fly, live in water, have legs})$

$P(\text{mammals} | \text{yes, no, yes, no}) = (6/7 * 6/7 * 2/7 * 2/7 * 7/20) = 0.0209912$

$P(\text{non-mammals} | \text{yes, no, yes, no}) = (12/13 * 10/13 * 3/13 * 9/13 * 13/20) = 0.0737369$

Logo, o animal com as características fornecidas pode ser classificado, pelo algoritmo de Bayes, como não mamífero.

3) (0,2) Execute árvores de decisão:

a) a AND b

b) a XOR b

c) (a AND b) OR (b AND c)

4) (0,2) Calcular a medida de entropia para os dados abaixo:

C1	0
C2	6
E=?	

C1	1
C2	5
E=?	

C1	2
C2	4
E=?	

C1	3
C2	3
E=?	

5) (0,2) Pesquise as principais diferenças entre os algoritmos de árvores de decisão: Hunt, ID3, C4.5, J4.8, C5.0, CART, Random-Forest.