

MSc in Software Engineering and Database Technologies

CT621 Artificial Intelligence

**Student Name:** Cristina Borges

**NUIG ID Number:** 20234955

**Course:** CT621 Artificial Intelligence

**Workshop No:** Week 5

**Assignments:** Week 5

**Date of Submission:** 20/06/2021

Contents

1.0 Assignment 2

1.1 Question 1 3

Information Gain 3

Data set 3

1.2 Question 2 5

Distance weighted nearest neighbour algorithm 5

Logical query to give paths 5

Go Instruction 6

Appendix 1 7

# Assignment

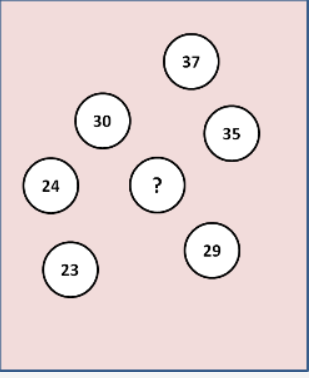
**Question 1.**

In Workshop 5 Section 1 we show how to calculate the information gain value for the outlook attribute in the Weather dataset. Calculate the information gain value for the temperature attribute in the Weather dataset. Please clearly show your workings.

**Question 2.**

Consider the classification problem depicted in the image below. The query instance is depicted by a circle with a question mark. The distance between the query instance and

* Case 23 is 4
* Case 24 is 3
* Case 30 is 2
* Case 29 is 2
* Case 35 is 3
* Case 37 is 4



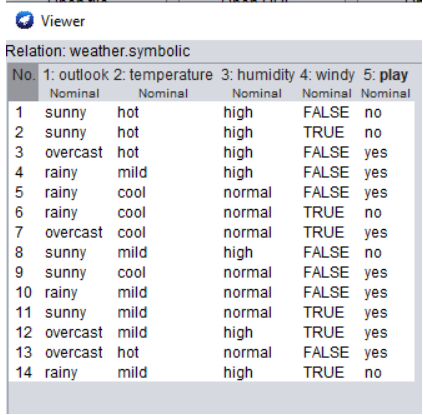
1. Using the Distance–Weighted Nearest Neighbour algorithm calculate the value assigned to the query instance. You can assume that all instances are considered in calculating the value of the query instance.
2. Using the K–Nearest Neighbour algorithm calculate the value assigned to the query instance when *K = 4.*

## 1.1 Question 1

## Information Gain

The information gain of an attribute is the reduction in entropy from partitioning the data according to that attribute (NUIG). The provide a quantitative measure of the quality of a split in the data (Zhou).

## Data set



The dataset is as set out in the table above which I extracted via Weka as part of my week 5 discussion.

The formula to calculate information gain is:

Gain(S,A) = Ent(S) -

We are looking for the information gain for the temperature attribute.

Summarised for ease in this table

|  |  |  |
| --- | --- | --- |
| Temperature | | |
| **Hot** | **Mild** | **Cool** |
| Yes | Yes | Yes |
| Yes | Yes | Yes |
| No | Yes | Yes |
| No | Yes | No |
|  | No |  |
|  | No |  |

Ent(S) = Ent([9+, 5-])

=-9/14 log(9/14,2) – 5/14log(5/14,2)

= 0.41 + 0.53

=.94

This corresponds to the calculation in our course material

Ent(S-temp= hot) = Ent([2+, 2-])

=-2/4log(2/4,2) – 2/4log(2/4,2)

=0.5 +0.5

= 1.0

Ent(S-temp= mild) = Ent([4+,2-])

=-4/6log(4/6,2) – 2/6log(2/6,2)

= 0.39 +0.52

= 0.92

Ent(S-temp= cold) = Ent([3+,1-])

= -3/4log(3/4,2) – 1/4log(1/4,2)

=0.31 + 0.50

=0.81

**Information Gain**

0.94 – 4/14 x1.0 – 6/14 x0.92 – 4/14 x0.81

= 0.02857

Similar to the windy example from the course notes, this is a very low value for the information gain which implies that the temperature attribute.

By inspection this appears to be reasonable as none of the temperature conditions appear to be very predictive of whether individual will play when you compare to the overall data set.

I also looked at if the temperature outlooks were combined to give 2 options (i.e. Hot vs Mild & Cold combined) to see if that provided a better information gain and benefit.

**All logs are base 2 and are denoted by log(number, base) in formula above**

## 2.1 Question 2

## Distance weighted algorithm

This provides each neighbour with a weighting which is inverse to the distance from the target (NUIG). This can be calculated using the following general formula

F(x) = (Sum of (Weights x f(x)) / Sum of weights

Before starting I have sorted each case by distance

|  |  |  |
| --- | --- | --- |
| Case | Distance (d) | Weighting (1/d) |
| 29 | 2 | ½ |
| 30 | 2 | ½ |
| 24 | 3 | 1/3 |
| 35 | 3 | 1/3 |
| 23 | 4 | ¼ |
| 37 | 4 | ¼ |

I have assumed all distances are Euclidian and that no scaling of values is required.

## Nearest neighbour algorithm

Nearest neighbour is Case 29 & Case 30.

Using the general formula above we get

[(½ x 29) + (½ x 30)] / (½ + ½) = 29.5

If the intention was to include all cases, the formula becomes:

[(½ x 29) + (½ x 30) + (1/3 x 24) + (1/3 x 35) + (1/4 x 23) +(1/4 x 37) ] / (½ + ½ + 1/3 + 1/3 + ¼ + 1/4) = 29.62

## K-nearest neighbour algorithm

If K=4 then we look at the 4 nearest neighbours only.

|  |  |  |  |
| --- | --- | --- | --- |
| Case | Distance (d) | Weighting (1/d) | Included for algorithm |
| 29 | 2 | ½ | Yes |
| 30 | 2 | ½ | Yes |
| 24 | 3 | 1/3 | Yes |
| 35 | 3 | 1/3 | Yes |
| 23 | 4 | ¼ | No |
| 37 | 4 | ¼ | No |

For k=4 only the four closest cases are included.

Calculation then becomes:

[(½ x 29) + (½ x 30) + (1/3 x 24) + (1/3 x 35)] / (½ + ½ + 1/3 + 1/3) = 29.50

From my additional reading, a few observations on the k-nearest neighbour algorithm are that:

1. Choosing a small number of neighbours can introduce noise – this didn’t occur in this case
2. When there are an even number of classes, we should choose an odd value for k – in this case, as there were cases that are equidistant, an odd value of k could have had a material impact on our answer i.e. if k=3 or k=5, we could have computed a value of 28.125 or 30.875 and 28.65 or 30.478 depending on the case selected

## Appendix 1

**Assignment 1**

**References**

CT621 Artificial Intelligence (2021) Machine Learning (Part 1)

Available at: NUIG Blackboard

Accessed 19th June 2021

Zhou V (2019) What Information Gain and Information Entropy are and how they're used to train Decision Trees.Available at: <https://victorzhou.com/blog/information-gain/>

Accessed 19th June 2021

CT621 Artificial Intelligence (2021) Machine Learning (Part 2)

Available at: NUIG Blackboard

Accessed 19th June 2021

Geeksforgeeks (2020) Weighted K-NN

Available at: <https://www.geeksforgeeks.org/weighted-k-nn/#:~:text=Weighted%20kNN%20is%20a%20modified,be%20more%20sensitive%20to%20outliers>.

Accessed 19th June 2021

Aditya K (2020) KNN Algorithm: When? Why? How?

Available at: <https://towardsdatascience.com/knn-algorithm-what-when-why-how-41405c16c36f>

Accessed 19th June 2021