**Aim**

Our aim is to prove the functionality, purpose, reliability and applicability of Bayesian Networks. For that, we are performing an inference test through the likelihood weighting method using our own algorithm on a Bayes Network previously given, and an exact inference test using the open source program JavaBayes on a different Bayes Network given. We are also proving a property of a specific type of Bayes Network through mathematical calculations. Finally, we are creating an example of a Bayesian Network to address a generic problem.

This problem is important to understand better the properties and implications of a Bayesian Network, a method to determine probabilities widely used in the artificial intelligence field.

**Methods**

**Results and Discussion**

This problem is important to understand better the properties and implications of a Bayesian Network, a method to determine probabilities widely used in the artificial intelligence field.

* Task 1
* Task 2
* Task 3
* Task 3 -1

Since there is no in the function, the independence is proven.

* Task 3 - 2
* Task 4

SD

WD

T

WO

HS

Work = 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sun | Mon | Tue | Wed | Thu | Fri | Sat |
| H | 0.05 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.05 |
| B | 0.05 | 0.2 | 0.8 | 0.8 | 0.8 | 0.2 | 0.05 |
| N | 0.05 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.5 |

Work = 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sun | Mon | Tue | Wed | Thu | Fri | Sat |
| H | 0.95 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.95 |
| B | 0.95 | 0.8 | 0.2 | 0.2 | 0.2 | 0.8 | 0.95 |
| N | 0.95 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.5 |

Stay Home = 1

|  |  |  |  |
| --- | --- | --- | --- |
| Work = 1 | H | N | C |
| H | 0.02 | 0.05 | 0.75 |
| B | 0.02 | 0.02 | 0.25 |
| N | 0.05 | 0.05 | 0.25 |
| Work = 0 | H | N | C |
| H | 0.05 | 0.25 | 0.95 |
| B | 0.1 | 0.25 | 0.75 |
| N | 0.4 | 0.5 | 0.9 |

Stay Home = 0

|  |  |  |  |
| --- | --- | --- | --- |
| Work = 1 | H | N | C |
| H | 0.98 | 0.95 | 0.25 |
| B | 0.98 | 0.98 | 0.75 |
| N | 0.95 | 0.95 | 0.75 |
| Work = 0 | H | N | C |
| H | 0.95 | 0.75 | 0.05 |
| B | 0.8 | 0.75 | 0.25 |
| N | 0.6 | 0.5 | 0.1 |

**Conclusions**

By the end of these experiments, we could see that algorithms that are more complex may have better results for classification, although that is not true for all the algorithms. We can also say that for this particular problem, the CFS approach had a negative influence in our classifiers and a positive influence for the classifiers from Weka. Overall, the findings were good and we can say that our classifiers are satisfactory and reliable, if we consider a comparison with the ZeroR algorithm. Although if we think that we are dealing with a classification problem that involves life risk, 79.1667% of accuracy may not be the best of results.

To improve this experiment we propose the following future work:

* Have a better understanding of the CFS tool as to why it didn’t help much our algorithms;
* Acquire more data for better classification;
* Implementation of other classifiers like the MLP.

With special consideration on the last one, because our algorithms had far better performance than Weka’s we can infer that implementing our own MLP (or similar) we would be able to achieve even better results.

**Reflection**

Throughout this assignment, we achieved a better understanding of the implications of a couple of classifying algorithms, specifically Naïve Bayes and K Nearest Neighbour. We also had a grasp into a type of feature selection tool, and how it relates with the classifiers. By running our codes several times during the experiment, we also had an idea of the influence that stratification has in the classification. The most interesting thing we learned was that algorithms of the same type implemented differently could have completely different results. Moreover, we also learned about the range of applicability of these classifying algorithms, and also that for each problem a different classifier may have a different – seldom better- rank in the performance rating than other more complex classifiers.

**Appendix**