BRL Assignment 1

Arne Lescrauwaet (852617312)

May 2024

1 Introduction

My primary mode of transportation is my bike, and living in Belgium makes checking weather forecast essential, with websites like "Buienradar" and "Accuweather" being my go-to resources. This inspired me to develop a Bayesian network to predict the probability of rain based on observed variables. Predicting precipitation falls within the broader field of weather forecasting, which involves anticipating atmospheric conditions for a specific location and time. Therefore, the central question for this assignment is: "Can I create a Bayesian network that accurately ($\geq$80%) predicts the probability of precipitation given the necessary observed variables?"

2 Data

Originally, I intended to use a dataset from local weather stations, but I could not find a suitable one. Therefore, I turned to Kaggle and decided to use the weather dataset, which is a slightly modified version of the Weather Prediction Dataset by Huber Florian.

The dataset was derived from the European Climate Assessment & Dataset project, offering daily observations from meteorological stations across Europe and the Mediterranean. These recordings span the years 2000 to 2010 and cover 18 locations. The dataset comprises 3654 observations and includes 165 variables.

Locations

1. Basel (Switzerland)
2. Budapest (Hungary)
3. Dresden (Germany)
4. Düsseldorf (Germany)
5. Kassel (Germany)
6. München (Germany)
7. De Bilt (Netherlands)
8. Maastricht (Netherlands)
9. Heathrow (UK)
10. Ljubljana (Slovenia)
11. Malmo (Sweden)
12. Stockholm (Sweden)
13. Montélimar (France)
14. Perpignan (France)
15. Tours (France)
16. Oslo (Norway)
17. Roma (Italy)
18. Sonnblick (Austria)

Variables and Units

Variable	**Unit**
Month	-
Mean Temperature	°C
Max Temperature	°C
Min Temperature	°C
Cloud Cover	oktas
Wind Speed	m/s
Wind Gust	m/s
Humidity	Fraction of 100%
Pressure	1000 hPa
Global Radiation	100 W/m^2
Precipitation	cm
Sunshine	Hours

I chose to focus on De Bilt since this was the closest weather station and utilized the following variables: The months from the month column were mapped to their corresponding seasons, creating a new variable called seasons. Cloud coverage was converted from oktas to percentages using the formula $\frac{x}{8} * 100$ and then categorized based on whether it was higher or lower than 85%. This threshold was adopted from the Bayesian Network paper by S. P. Khabarov et al [3] for precipitation forecasting. The humidity variable was converted to percentages and categorized based on its quantiles. Pressure was converted to

hPa and categorized based on whether it was above or below its mean. The paper by S. P. Khabarov et al. used a threshold of 760 hPa, but their observations were from South Russia, and the values differed significantly from mine thus I decided not to adopt this value. Precipitation was renamed to rain and categorized into rain or no rain based on the precipitation value.

Category	**Type**	**Values**
Season	Ordinal	Summer, Autumn, Winter, Spring
Cloud Coverage	Ordinal	Low, Medium, High
Humidity	Ordinal	Low, High
Pressure	Ordinal	Low, High
Temperature	Ordinal	Low, High
Rain	Ordinal	No Rain, Rain

3 Bayesian Network

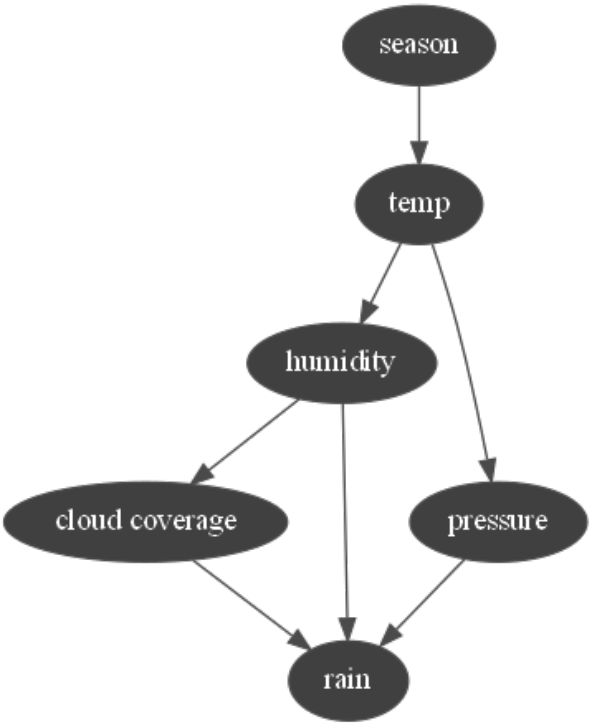

Figure 1: Bayesian Network

4 Development

Weather forecasting is complex and outside my area of expertise. Therefore, before starting development, I reviewed similar weather prediction models and the variables they used [3, 2, 4, 1]. I selected variables that were present in my dataset, commonly used in the literature, and observable via weather report websites. After selecting the variables, the most challenging part was determining their relationships. Through Google searches to understand the influence of variables on weather systems, I quickly discovered that these systems are highly interconnected, and their relationships can significantly impact the outcome. Ultimately, I settled on the network shown in the previous section.

My rationale behind these connections is as follows: The connection between season and temperature is straightforward. Temperature is linked to humidity because warm air can hold more water vapor

than cold air, resulting in higher relative humidity when the air is cooler and lower relative humidity when the air is warmer. Temperature also affects pressure, as atmospheric pressure decreases when the temperature increases. Humidity is connected to cloud coverage because cloud coverage increases with higher humidity. Humidity directly influences the probability of rain since higher humidity means more water vapor, increasing the likelihood of precipitation. Atmospheric pressure affects the probability of rain because low-pressure systems usually lead to precipitation, while high-pressure systems typically result in calmer weather. The impact of cloud coverage on rain is quite straightforward.

5 PyAgrum

1. gum.BayesNet() to initialise the network
2. gum.LabelizedVariable to define a variable with a set of labeled states
3. bn.add() to add a LabelizedVariable to the network
4. bn.addArc() to connect two variables
5. gimg.export() export the bn to an image
6. bn.cpt() select the conditional probability table of a variable in the network
7. cpt.fillWith() to add values to a conditional probability table of a variable in the network
8. gum.LazyPropagation() is used to create an instance of LazyPropagation, which is an inference engine for Bayesian networks. There are several inference engines in pyAgrum: "LazyPropagation", "Variable Elimination", and "Shafer-Shenoy". I opted for the LazyPropagation because the network is moderately connected, not extremely dense, and resembles a tree structure which are properties that work well with LazyPropagation.
9. gnb.showInference(bn, evs, targets) is used to visualize the results of an inference. The "evs" argument is the observed variables and the "targets" variable is the nodes you want to predict probabilities for.
10. ie.setEvidence() sets the evidence used in the showInference function
11. ie.makeInference() is used to explicitly trigger the inference computations within an inference engine
12. ie.posterior() is used to compute and return the posterior probability distribution of a specified variable

The complete implementation can be found on Github.

6 Results

I computed a confusion matrix on the test set (10% of the data), which showed satisfactory results and achieved a 65% accuracy score. Additionally, I tested the network on several weather observations. In the first situation, with an observed temperature of 19 degrees Celsius, 40% cloud coverage, and 71% humidity, the model predicted a 36.83% probability of rain, which aligns with the website's predicted 20% probability of rain. In the second situation, with an observed temperature of 18 degrees Celsius, 98% cloud coverage, and 75% humidity, the model predicted a 67.48% probability of rain, which corresponds to the website's predicted 93% probability of rain. While the prediction class is generally correct, the exact probabilities are not that accurate. This accuracy could likely be improved by including additional variables in the network, such as proximity to the sea or mountains, elevation above sea level, latitude, and longitude. However, datasets containing all these variables are more challenging to find. The connections between the variables also have a big impact on the result of the network as can be seen in table 3 of [4]. Thus a better knowledge of weather systems or building the network algorithmically i.e. using hill climb, General 2-phase Restricted Maximization, ... such as done in [4] would probably help improve the model. In the end, I would say I was not entirely successful in finding the answer to my initial question "Can I create a Bayesian network that accurately predicts the probability of precipitation given the necessary observed variables?". My network manages to predict if there will be precipitation but not with enough accuracy for my liking.

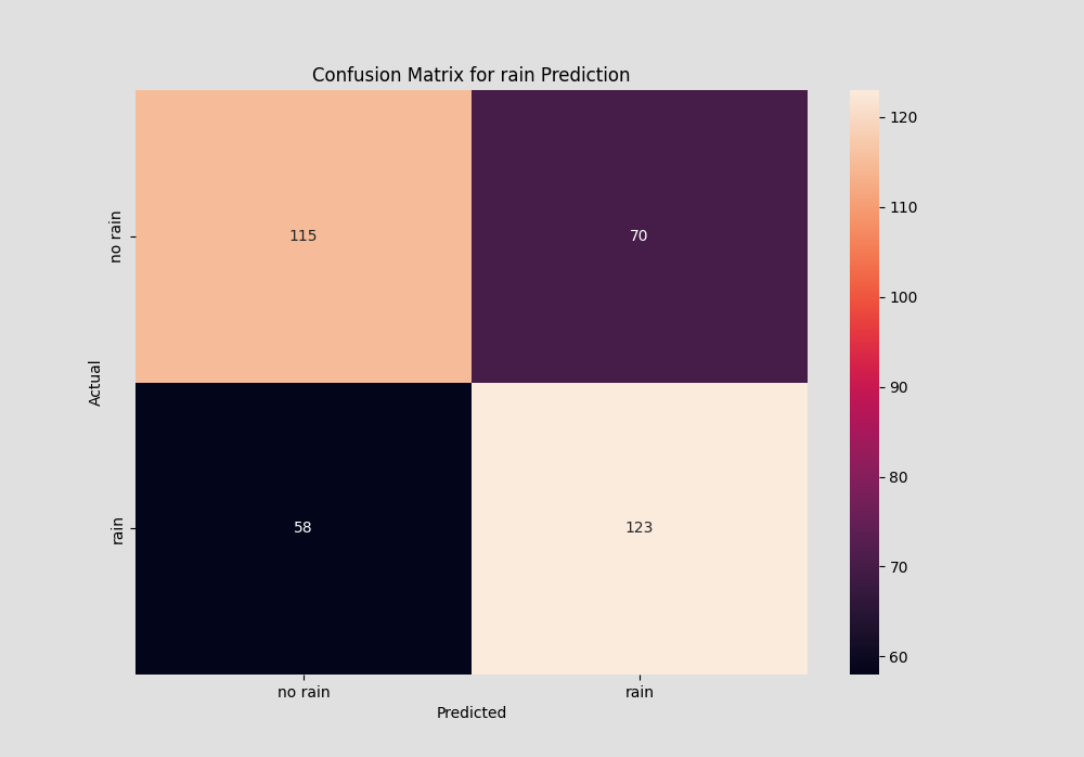

Figure 2: Confusion matrix

References

[1] Jingjing Chang et al. "Dynamic Bayesian networks with application in environmental modeling and management: A review". In: *Environmental Modelling Software* 170 (2023), p. 105835. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2023.105835. URL: https://www.sciencedirect.com/science/article/pii/S1364815223002219.

[2] Prabal Das and Kironmala Chanda. "Bayesian Network based modeling of regional rainfall from multiple local meteorological drivers". In: *Journal of Hydrology* 591 (2020), p. 125563. ISSN: 0022-1694. DOI: https://doi.org/10.1016/j.jhydrol.2020.125563. URL: https://www.sciencedirect.com/science/article/pii/S0022169420310234.

[3] S Khabarov, M Shilkina, and N Vasiliev. "Precipitation forecast based on the Bayesian Network". In: *IOP Conference Series: Earth and Environmental Science* 806 (Aug. 2021), p. 012016. DOI: 10.1088/1755-1315/806/1/012016.

[4] Salwa Rizqina Putri and Arie Wahyu Wijayanto. "Learning Bayesian Network for Rainfall Prediction Modeling in Urban Area using Remote Sensing Satellite Data (Case Study: Jakarta, Indonesia)". In: *Proceedings of The International Conference on Data Science and Official Statistics* 2021.1 (Jan. 2022), pp. 77–90. DOI: 10.34123/icdsos.v2021i1.37. URL: https://proceedings.stis.ac.id/icdsos/article/view/37.