Problem set 3: Uncertainty and Bayes Network

Jacob Mongold

October 2025

Problem 1: Inference using Full Joint Distributions

Q1: Inference using Full Joint Distributions

Given the full joint distribution of the Boolean variables *Toothache*, *Catch*, and *Cavity*:

Toothache	Catch	Cavity	P
T	T	T	0.108
F	T	T	0.012
T	F	T	0.072
F	F	T	0.008
T	T	F	0.016
F	T	F	0.064
T	F	F	0.144
F	F	F	0.576

(a) $P($**toothache**$)$

$$\begin{aligned} P(\text{toothache}) &= \sum_{catch} \sum_{cavity} P(\text{toothache}, catch, cavity) \\ &= (0.108 + 0.072) + (0.016 + 0.144) \\ &= 0.34 \end{aligned}$$

$$P(\neg\text{toothache}) = 1 - 0.34 = 0.66$$

(b) $P(\textbf{cavity})$

$$\begin{aligned} P(\text{cavity}) &= \sum_{toothache} \sum_{catch} P(toothache, catch, cavity) \\ &= 0.108 + 0.012 + 0.072 + 0.008 \\ &= 0.20 \end{aligned}$$

$$P(\neg\text{cavity}) = 0.80$$

(c) $P(\textbf{toothache} \mid \textbf{cavity})$

$$\begin{aligned} P(\text{toothache} \mid \text{cavity}) &= \frac{P(\text{toothache}, \text{cavity})}{P(\text{cavity})} \\ &= \frac{\sum_{catch} P(toothache, catch, cavity)}{0.20} \\ &= \frac{0.108 + 0.072}{0.20} = 0.90 \end{aligned}$$

(d) $P(\textbf{cavity} \mid \textbf{toothache} \vee \textbf{catch})$

$$P(\text{cavity} \mid \text{toothache} \vee \text{catch}) = \frac{P(\text{cavity}, \text{toothache} \vee \text{catch})}{P(\text{toothache} \vee \text{catch})}$$

$$P(\text{cavity}, \text{toothache} \vee \text{catch}) = 0.108 + 0.012 + 0.072 = 0.192$$

$$P(\text{toothache} \vee \text{catch}) = 0.108 + 0.012 + 0.072 + 0.016 + 0.064 + 0.144 = 0.416$$

$$P(\text{cavity} \mid \text{toothache} \vee \text{catch}) = \frac{0.192}{0.416} \approx 0.46$$

(e) $P(\textbf{catch} \mid \textbf{cavity})$

$$P(\text{catch} \mid \text{cavity}) = \frac{P(\text{catch}, \text{cavity})}{P(\text{cavity})}$$
$$= \frac{0.108 + 0.012}{0.20} = 0.60$$

(f) $P(\textbf{catch} \mid \textbf{cavity} \wedge \textbf{toothache})$

$$P(\text{catch} \mid \text{cavity} \wedge \text{toothache}) = \frac{P(\text{catch}, \text{cavity}, \text{toothache})}{P(\text{cavity}, \text{toothache})}$$
$$= \frac{0.108}{0.108 + 0.072} = \frac{0.108}{0.18} = 0.60$$

Thus, $P(\text{catch} \mid \text{cavity} \wedge \text{toothache}) = P(\text{catch} \mid \text{cavity}) = 0.6$, indicating conditional independence of *Catch* and *Toothache* given *Cavity*.

Problem 2: Naive Bayes Classification Modeling

Q2: Calculating Player Characteristics for Short-listing

Each instance is described by four categorical attributes: League $\in \{\text{SerieA}, \text{LaLiga}, \text{PremierLeague}\}$, Position $\in \{\text{LW}, \text{CF}, \text{RW}\}$, Preferred Foot $\in \{\text{Left}, \text{Right}\}$, and Capped $\in \{\text{Yes}, \text{No}\}$.

The target variable is Shortlisted $\in \{\text{True}, \text{False}\}$.

Class Priors

$$P(\text{True}) = \frac{7}{14} = 0.5, \qquad P(\text{False}) = 0.5$$

Conditional Probabilities

$$P(\text{SerieA} \mid T) = \tfrac{3}{7}, \qquad P(\text{SerieA} \mid F) = \tfrac{2}{7},$$

$$P(\text{LaLiga} \mid T) = \tfrac{3}{7}, \qquad P(\text{LaLiga} \mid F) = \tfrac{1}{7},$$

$$P(\text{Premier} \mid T) = \tfrac{1}{7}, \qquad P(\text{Premier} \mid F) = \tfrac{4}{7};$$

$$P(\text{LW} \mid T) = \tfrac{1}{7}, \qquad P(\text{LW} \mid F) = \tfrac{2}{7},$$

$$P(\text{CF} \mid T) = \tfrac{3}{7}, \qquad P(\text{CF} \mid F) = \tfrac{2}{7},$$

$$P(\text{RW} \mid T) = \tfrac{3}{7}, \qquad P(\text{RW} \mid F) = \tfrac{3}{7};$$

$$P(\text{Left} \mid T) = \tfrac{2}{7}, \qquad P(\text{Left} \mid F) = \tfrac{4}{7},$$

$$P(\text{Right} \mid T) = \tfrac{5}{7}, \qquad P(\text{Right} \mid F) = \tfrac{3}{7};$$

$$P(\text{Capped=yes} \mid T) = \tfrac{4}{7}, \quad P(\text{Capped=yes} \mid F) = \tfrac{3}{7},$$

$$P(\text{Capped=no} \mid T) = \tfrac{3}{7}, \quad P(\text{Capped=no} \mid F) = \tfrac{4}{7}.$$

Prediction 1

$$x_1 = (\text{LaLiga}, \text{RW}, \text{Right}, \text{Yes})$$

$$P(T \mid x_1) \propto 0.5(3/7)(3/7)(5/7)(4/7) = 0.0375,$$

$$P(F \mid x_1) \propto 0.5(1/7)(3/7)(3/7)(3/7) = 0.0056.$$

Hence,

$$P(T \mid x_1) > P(F \mid x_1) \;\Rightarrow\; \text{Shortlisted} = \text{True}.$$

Prediction 2

$$x_2 = (\text{PremierLeague}, \text{LW}, \text{Left}, \text{No})$$

$$P(T \mid x_2) \propto 0.5(1/7)(1/7)(2/7)(3/7) = 0.00125,$$

$$P(F \mid x_2) \propto 0.5(4/7)(2/7)(4/7)(4/7) = 0.0267.$$

Thus,

$$P(T \mid x_2) < P(F \mid x_2) \;\Rightarrow\; \text{Shortlisted} = \text{False}.$$

Interpretation

Player 1 aligns strongly with the distribution of shortlisted players (LaLiga, right-footed, capped), while Player 2 matches patterns typical of non-shortlisted players (Premier League, uncapped, left-footed). Given balanced priors, the model's predictions are intuitive and interpretable.

Q3: Implementation Summary (Naive Bayes from scratch)

I implemented a Multinomial Naive Bayes text classifier entirely from scratch in Python with NumPy only. Each document is tokenized by lowercasing, removing non-alphanumeric characters except apostrophes, and splitting on whitespace. Training computes class priors $P(c)$ and per-class token counts; inference uses log-space:

$$\log_2 P(c \mid x) \;\propto\; \log_2 P(c) + \sum_{w \in V} \text{count}_x(w)\, \log_2 P(w \mid c).$$

At test time, out-of-vocabulary tokens (not seen in the training vocabulary) are ignored, satisfying the OOV requirement. Providing two likelihood variants: (i) maximum-likelihood (no smoothing), $P(w \mid c) = \frac{\text{count}(w,c)}{\sum_{v \in V} \text{count}(v,c)}$, and (ii) Laplace (+1) smoothing, $P(w \mid c) = \frac{\text{count}(w,c)+1}{\sum_{v \in V} \text{count}(v,c)+|V|}$. We then return accuracy, macro-precision, macro-recall, and confusion matrices (rows = true, columns = predicted).

Q4: Results (Test Sets), Metrics and Confusion Matrices

Movie Reviews (pos/neg)

Setting	Accuracy	Macro Precision	Macro Recall
No smoothing	0.7067	0.7225	0.7067
Laplace (+1)	0.1933	0.1928	0.1933

Confusion matrices (rows = true, cols = pred):

No smoothing: $\begin{bmatrix} 126 & 24 \\ 64 & 86 \end{bmatrix}$ Laplace (+1): $\begin{bmatrix} 32 & 118 \\ 124 & 26 \end{bmatrix}$

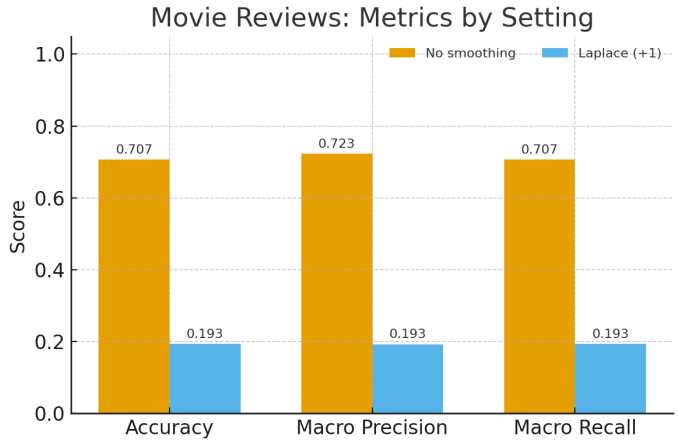
Twenty Newsgroups (atheism/christian/misc)

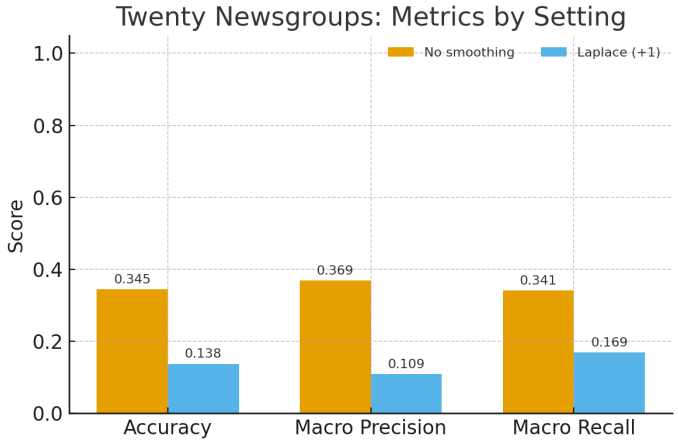
Setting	Accuracy	Macro Precision	Macro Recall
No smoothing	0.3450	0.3690	0.3408
Laplace (+1)	0.1375	0.1092	0.1691

Confusion matrices (rows = true, cols = pred):

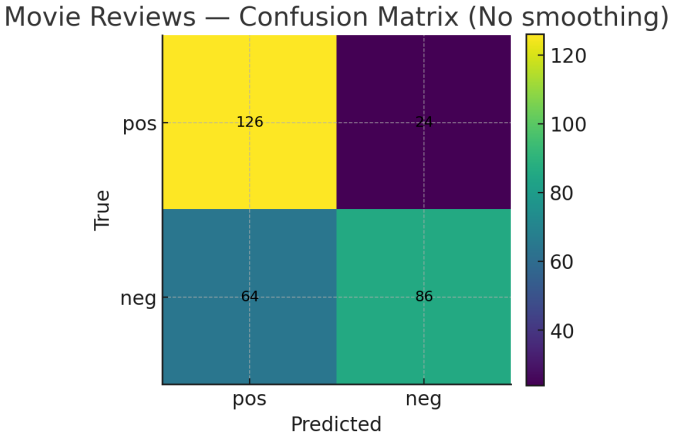
$$\text{No smoothing: } \begin{bmatrix} 119 & 15 & 8 \\ 151 & 23 & 4 \\ 89 & 14 & 6 \end{bmatrix} \qquad \text{Laplace (+1): } \begin{bmatrix} 11 & 26 & 105 \\ 66 & 3 & 109 \\ 47 & 17 & 45 \end{bmatrix}$$

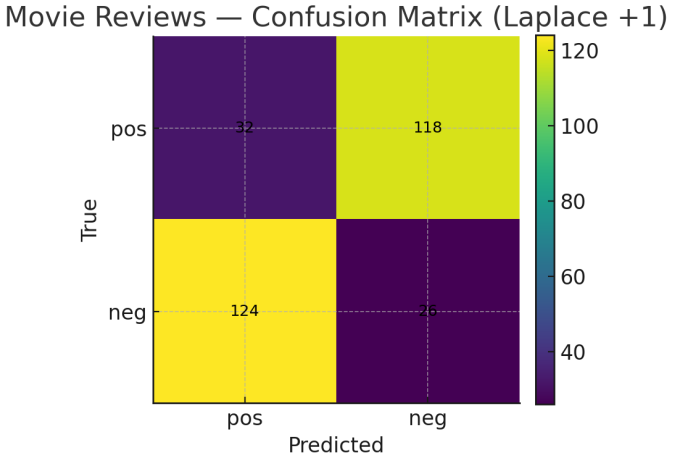
The results show that Laplace smoothing (+1) severely reduces performance across both datasets.





For the Movie Reviews task, the model without smoothing performs reasonably well, with accuracy and recall both around 0.71. It can distinguish between positive and negative reviews with fair reliability. When Laplace smoothing is applied, accuracy collapses to 0.19. The confusion matrices make this drop obvious; the smoothed model predicts the wrong class for most samples, effectively inverting many predictions.

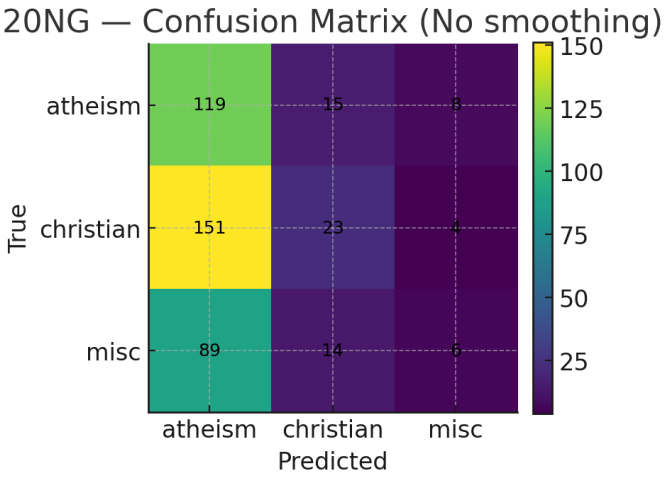


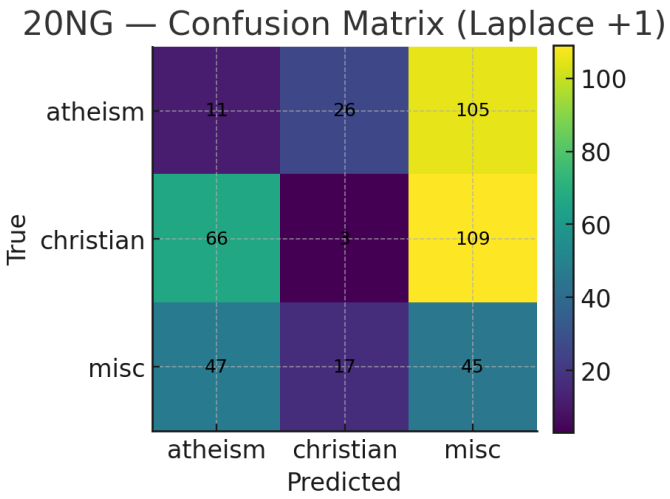


This suggests that adding +1 to every count overly flattens the probabili-

ties and wipes out useful distinctions between word frequencies in positive and negative reviews.

For the Twenty Newsgroups task, accuracy falls from 0.35 to 0.14 with Laplace smoothing, along with similar drops in precision and recall. Without smoothing, the model often confuses atheism and christian posts, which makes sense given how much their language overlaps, while the "misc" class is under-predicted. When smoothing is applied, predictions become scattered and mostly wrong; the model loses its ability to separate the classes.





Overall, Laplace smoothing hurts both models by flattening the probability estimates too much. In these text classification tasks, it blurs the differences between word probabilities so the model can't separate the classes effectively. The unsmoothed versions aren't perfect, but they keep useful distinctions between classes and make far more consistent predictions.

Q5: Movie Reviews vs. Newsgroups

The Naive Bayes classifier performs much better on the Movie Review dataset (accuracy ≈ 0.71) than on the Twenty Newsgroups dataset (accuracy ≈ 0.35). The gap comes down to task complexity, vocabulary size, and the independence assumption. Movie reviews are a binary sentiment task where common words strongly tie to positive or negative tone, so the vocabulary is small and focused. The Newsgroups task, on the other hand, covers three overlapping topics: athe-

ism, christian, and misc, where word usage is broader and less distinct. That makes likelihood estimates sparse and breaks the independence assumption more severely. The larger vocabulary also makes Laplace smoothing worse, flattening probabilities and wiping out class contrast. Naive Bayes works well for sentiment analysis with smaller, balanced vocabularies but struggles on bigger, more diverse text where context between words matters.

Q6: Advantages and Drawbacks of Naive Bayes

Movie Reviews

Without smoothing, Naive Bayes performs well (Accuracy = 0.7067; Macro P = 0.7225; Macro R = 0.7067). The binary sentiment task uses a relatively compact vocabulary, and many words appear in both classes, so maximum-likelihood estimates stay reliable and discriminative.

Newsgroups

Without smoothing, performance is moderate (Accuracy = 0.3450; Macro P = 0.3690; Macro R = 0.3408). The larger, sparser topic vocabulary and the extra class make this a tougher setting, leading to more overlap and ambiguity between categories.

Effect of Laplace (+1)

On both datasets, add-one smoothing sharply reduces performance (Reviews = 0.1933; Newsgroups = 0.1375). In high-dimensional text data, add-one is too

aggressive because the denominator $\sum_v \text{count}(v, c) + |V|$ grows large when $|V|$ is big. This drives $P(w \mid c)$ toward a near-uniform distribution and erases useful contrast between classes.

Interpretation

Naive Bayes is appealing because it is simple, fast, and efficient for both training and inference. Its strong inductive bias makes it reliable even with small datasets, and it often produces well-calibrated class rankings despite the unrealistic independence assumption. The model is also easy to interpret since token weights and log-odds can be inspected directly to see what drives predictions. The same assumptions that make it efficient also limit its accuracy. Conditional independence rarely holds in text data, which can lead to poorly estimated likelihoods. Add-one smoothing with large vocabularies can flatten probabilities too much and blur class boundaries. The bag-of-words approach also drops word order and context, losing useful information. Naive Bayes performance tends to level off compared to linear SVMs or modern neural models unless preprocessing and feature selection are carefully tuned.

Q7: Laplace Smoothing and Updated Results

Smoothing formula.

For class c and token w:

$$P(w \mid c) \;=\; \frac{\text{count}(w, c) + 1}{\sum_{v \in V} \text{count}(v, c) + |V|}.$$

Compared to the unsmoothed MLE, the numerator adds +1 for every token (preventing zeros), and the denominator adds $|V|$ to conserve probability mass. This ensures $P(w \mid c) > 0$ even for unseen tokens, avoiding posterior collapse when a test document contains an unseen word for class c.

Updated metrics with Laplace (+1).

We recomputed the test metrics using Laplace smoothing:

Movie Reviews:

Accuracy = 0.1933, Macro Precision = 0.1928, Macro Recall = 0.1933.

$$\begin{bmatrix} 32 & 118 \\ 124 & 26 \end{bmatrix}$$

Newsgroups:

Accuracy = 0.1375, Macro Precision = 0.1092, Macro Recall = 0.1691.

$$\begin{bmatrix} 11 & 26 & 105 \\ 66 & 3 & 109 \\ 47 & 17 & 45 \end{bmatrix}$$

Add-one smoothing can be too strong for large vocabularies: adding +1 to all tokens per class (and $+|V|$ to denominators) flattens likelihoods toward uniform, reducing discriminability.

Problem 4: Bayes Network - Sprinkler, Rain, and Wet Grass

Network structure

The graph consists of four Boolean variables:

$$\text{Cloudy} \to \begin{cases} \text{Sprinkler} \\ \\ \text{Rain} \end{cases} \quad \text{and} \quad \{\text{Sprinkler}, \text{Rain}\} \to \text{WetGrass}.$$

The full joint distribution factorizes as

$$P(C, S, R, W) = P(C)\, P(S \mid C)\, P(R \mid C)\, P(W \mid S, R),$$

where C =Cloudy, S =Sprinkler, R =Rain, and W =WetGrass.

Q8: Full Joint Probability Distribution

The conditional probability tables (CPTs) are:

$P(S, R \mid C)$

C	$P(S{=}T \mid C)$	$P(R{=}T \mid C)$
T	0.10	0.80
F	0.50	0.20

$P(W{=}T \mid S, R)$

S	R	$P(W{=}T \mid S, R)$
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.01

$$P(C{=}T) = 0.5, \qquad P(C{=}F) = 0.5.$$

Each joint probability entry is computed by substituting the appropriate terms into

$$P(C, S, R, W) = P(C)\, P(S \mid C)\, P(R \mid C)\, P(W \mid S, R).$$

For example,

$$P(C{=}T, S{=}T, R{=}T, W{=}T) = 0.5 \times 0.1 \times 0.8 \times 0.99 = 0.0396.$$

The complete joint distribution over all $2^4 = 16$ worlds is:

C	S	R	W	$P(C, S, R, W)$
T	T	T	T	0.0396
T	T	T	F	0.0004
T	T	F	T	0.0090
T	T	F	F	0.0010
T	F	T	T	0.3240
T	F	T	F	0.0360
T	F	F	T	0.0009
T	F	F	F	0.0891
F	T	T	T	0.02475
F	T	T	F	0.00025
F	T	F	T	0.0225
F	T	F	F	0.0025
F	F	T	T	0.08100
F	F	T	F	0.00900
F	F	F	T	0.00025
F	F	F	F	0.05625
				$\sum = 1.0000$

Q9. Posterior inference: $P(\textbf{Rain} \mid \textbf{WetGrass})$ vs. $P(\textbf{Sprinkler} \mid \textbf{WetGrass})$

We first compute the total probability of wet grass:

$$\begin{aligned} P(W{=}T) &= \sum_{C,S,R} P(C, S, R, W{=}T) \\ &= 0.0396 + 0.0090 + 0.3240 + 0.0009 + 0.02475 + 0.0225 + 0.08100 + 0.00025 \\ &= 0.5020. \end{aligned}$$

(a) Probability that it rained given wet grass.

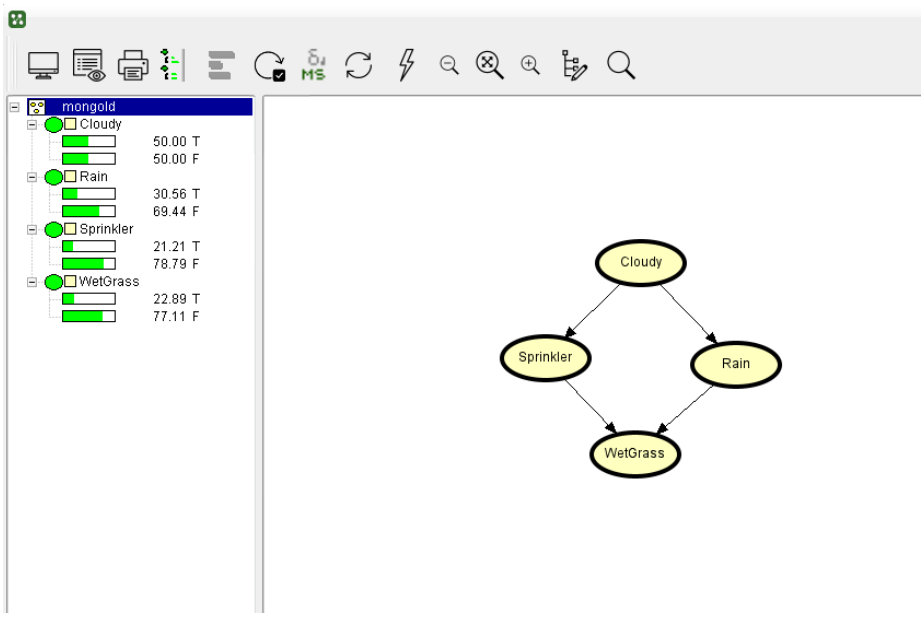
$$\begin{aligned} P(R{=}T, W{=}T) &= \sum_{C,S} P(C, S, R{=}T, W{=}T) \\ &= 0.0396 + 0.3240 + 0.02475 + 0.08100 = 0.46935, \\ P(R{=}T \mid W{=}T) &= \frac{0.46935}{0.5020} \approx 0.935. \end{aligned}$$

(b) Probability that the sprinkler was on given wet grass.

$$\begin{aligned} P(S{=}T, W{=}T) &= \sum_{C,R} P(C, S{=}T, R, W{=}T) \\ &= 0.0396 + 0.0090 + 0.02475 + 0.0225 = 0.09585, \\ P(S{=}T \mid W{=}T) &= \frac{0.09585}{0.5020} \approx 0.191. \end{aligned}$$

Given that the grass is wet, it is far more likely that it rained ($P \approx 0.94$) than that the sprinkler was on ($P \approx 0.19$).

Q10. Verification using *Hugin-Lite*



The probabilities displayed in the Hugin model are:

$$P(\text{Cloudy}) = 0.50 \text{ (T)}, \quad 0.50 \text{ (F)}$$

$$P(\text{Rain}) = 0.3056 \text{ (T)}, \quad 0.6944 \text{ (F)}$$

$$P(\text{Sprinkler}) = 0.2121 \text{ (T)}, \quad 0.7879 \text{ (F)}$$

$$P(\text{WetGrass}) = 0.2289 \text{ (T)}, \quad 0.7711 \text{ (F)}$$

When evidence for WetGrass = True is entered, Hugin updates the posterior probabilities for Rain and Sprinkler. The probability of Rain increases more sharply than that of the Sprinkler, indicating that wet grass is more likely due to rain than sprinkler use. This is consistent with the network's structure and the numerical relationships between the variables.

The conditional independencies represented in this network are:

- Sprinkler and Rain are conditionally independent given Cloudy.
- WetGrass depends on both Sprinkler and Rain.
- Cloudy affects WetGrass indirectly through Rain and Sprinkler.

These independencies justify the compact factorization:

$$P(C, S, R, W) = P(C)\, P(S \mid C)\, P(R \mid C)\, P(W \mid S, R),$$

which reduces the number of parameters from $2^4 - 1 = 15$ to only 9. In Hugin, this structure is visible by the missing connection between *Sprinkler* and *Rain*, showing that their relationship is mediated entirely through *Cloudy*.

Q11. Conditional independence analysis

The Bayesian network encodes several key conditional independencies:

1. $\text{Sprinkler} \perp \text{Rain} \mid \text{Cloudy}$

   Knowing whether it is cloudy renders the sprinkler and rain independent.

2. $\text{WetGrass} \perp \text{Cloudy} \mid (\text{Sprinkler}, \text{Rain})$

   Once we know whether the sprinkler was on and whether it rained, the probability of the grass being wet is unaffected by cloudiness.

3. $\text{Cloudy} \perp \text{WetGrass} \mid (\text{Sprinkler}, \text{Rain})$

   The observation of wet grass provides no additional information about cloudiness beyond that conveyed by sprinkler and rain.

These independencies allow the network to be factorized compactly as

$$P(C, S, R, W) = P(C)\, P(S \mid C)\, P(R \mid C)\, P(W \mid S, R),$$

which reduces the total number of parameters from $2^4 - 1 = 15$ to only 9. In Hugin, this conditional independence structure is shown visually by the missing connection between *Sprinkler* and *Rain*, meaning their relationship is explained entirely through the *Cloudy* variable.