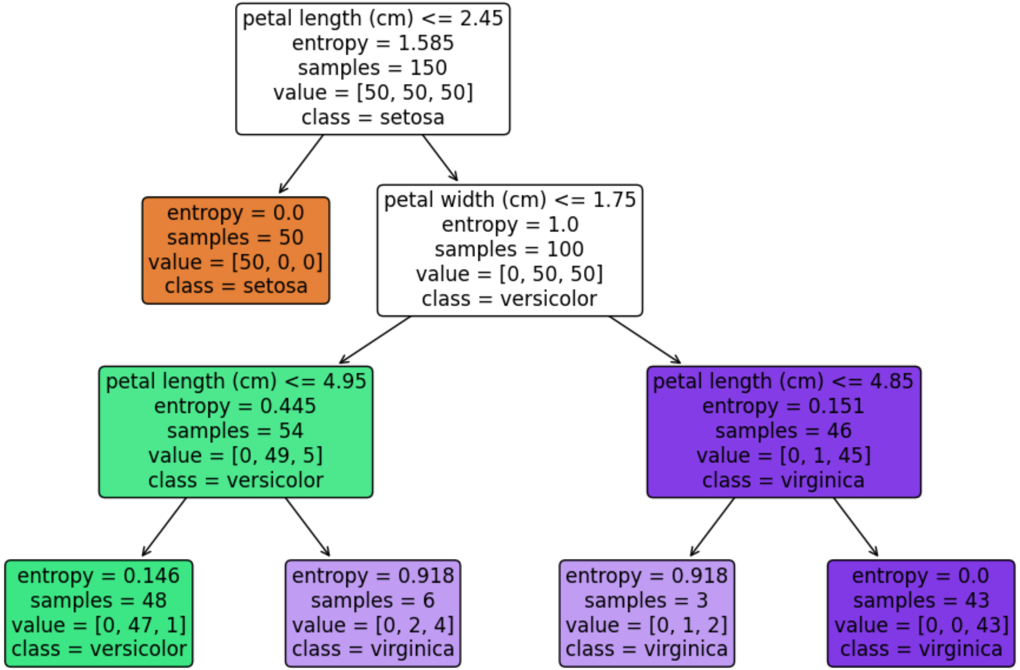
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Algorithms for Data Science

Decision Making (Decision Tree / Bayesian Networks)

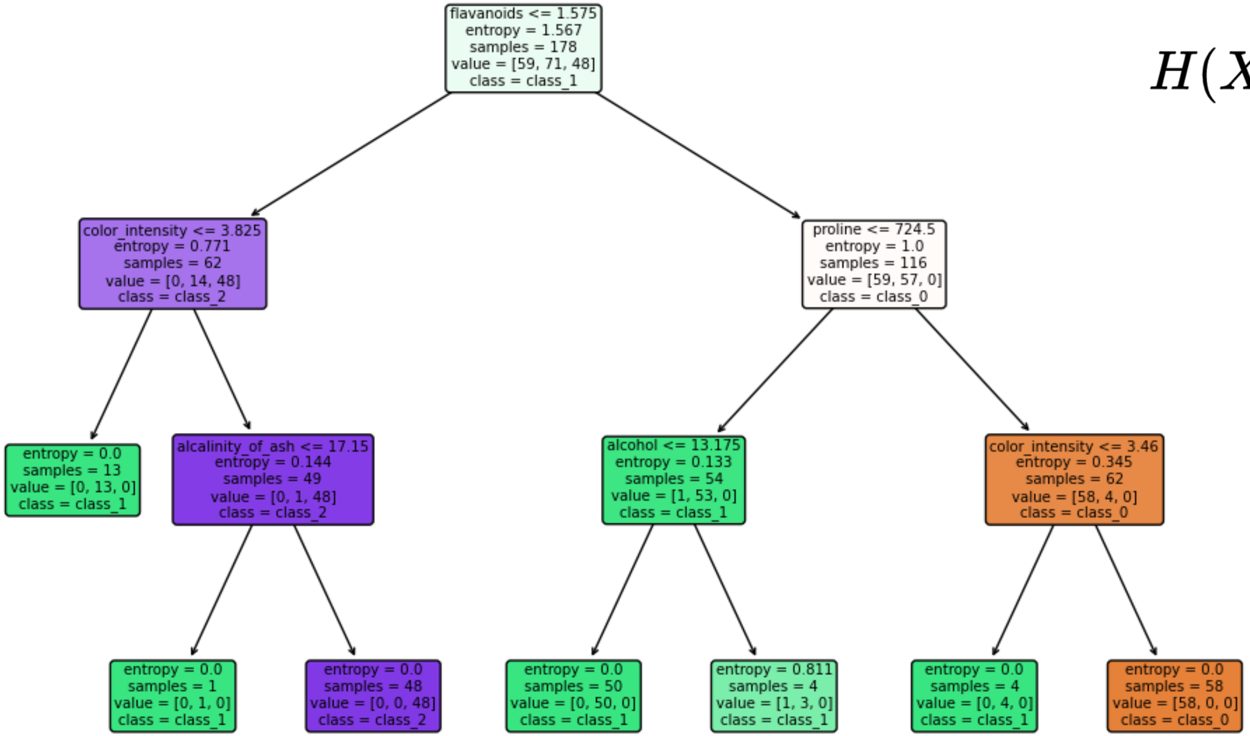
Decision Trees: Learning from Choices



Why Decision Trees?

- **Interpretable** – Easy to understand.
- **Handles both** categorical & numerical data.
- Works well for **rule-based** decision-making.

How Do Decision Trees Make Splits?



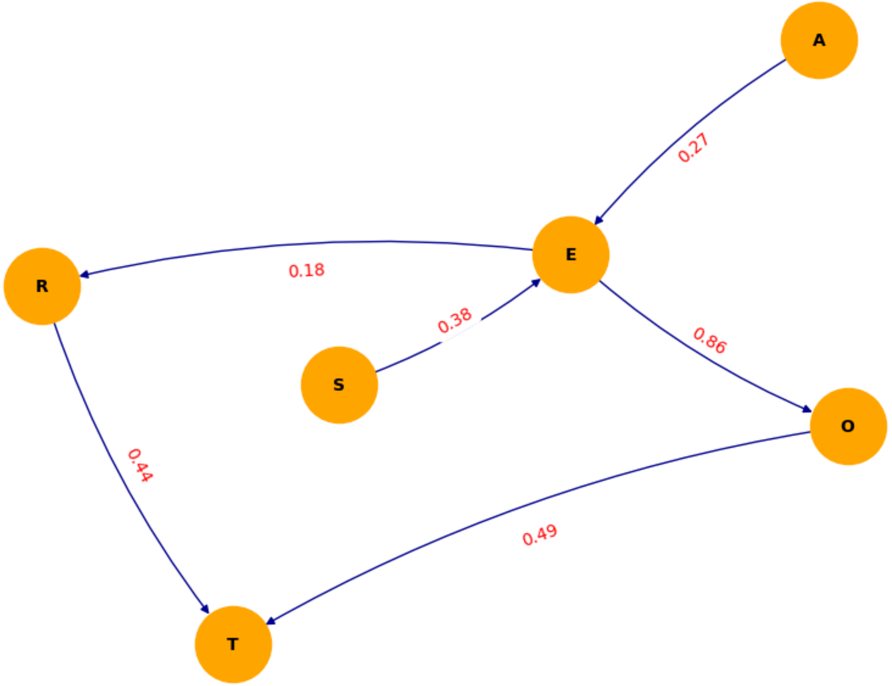
$$H(X) = -\sum p_i \log_2 p_i$$

- **Entropy = 0** → The node is pure (only one class is present).
- **Entropy = 1** → The node is evenly split between **two** classes (e.g., 50% vs. 50%).
- **Entropy > 1** → Happens when there are **more than two classes**.

Bayesian Networks: Probabilistic Reasoning

Bayesian Networks

- A **Directed Acyclic Graph (DAG)** that represents **probabilistic relationships**.
- Can **model uncertainty** and causal relationships.
- Useful for medical diagnosis, fraud detection, and AI reasoning.



Bayesian Inference & Conditional Probability

```python
from pgmpy.models import BayesianNetwork
from pgmpy.inference import VariableElimination
from pgmpy.factors.discrete import TabularCPD

# Define the structure
model = BayesianNetwork([('Flu', 'Fever')])

# Define Conditional Probability Distributions
cpd_flu = TabularCPD(variable='Flu', variable_card=2, values=[[0.9], [0.1]])
cpd_fever = TabularCPD(variable='Fever', variable_card=2,
                       values=[[0.15, 0.85], [0.85, 0.15]],
                       evidence=['Flu'], evidence_card=[2])

model.add_cpds(cpd_flu, cpd_fever)

# Perform Inference
inference = VariableElimination(model)
result = inference.query(variables=['Flu'], evidence={'Fever': 1})
print(result)
```

✓ 0.0s

```
+--------+------------+
| Flu    |   phi(Flu) |
+========+============+
| Flu(0) |     0.9808 |
+--------+------------+
| Flu(1) |     0.0192 |
+--------+------------+
```

Bayes Theorem:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Prior Probability:

$$P(Flu) = 0.10$$

Conditional Probability:

$$P(\text{Fever} \mid \text{Flu}) = 0.85$$

Compute

$$P(Flu|Fever)$$

Variable Elimination vs Gibbs Sampling

Feature	Exact	Complexity	Best Use Case
Variable Elimination	☑ Yes	$O(n^2)$ to $O(n^3)$	Small Bayesian Networks
Gibbs Sampling	✕ No	$O(k)$ (for large k)	Large Bayesian Networks

Variable Elimination

Mathematically, if we have the joint distribution *P(A,B)*, **we can eliminate** *B* **by summing over all values of** *B*

$$P(A) = \sum_B P(A,B)$$

Example:

A	B	P(A,B)
0	0	0.2
0	1	0.3
1	0	0.4
1	1	0.1

$$P(A=0) = P(A=0, B=0) + P(A=0, B=1)$$

$$= \mathbf{0.2 + 0.3 = 0.5}$$

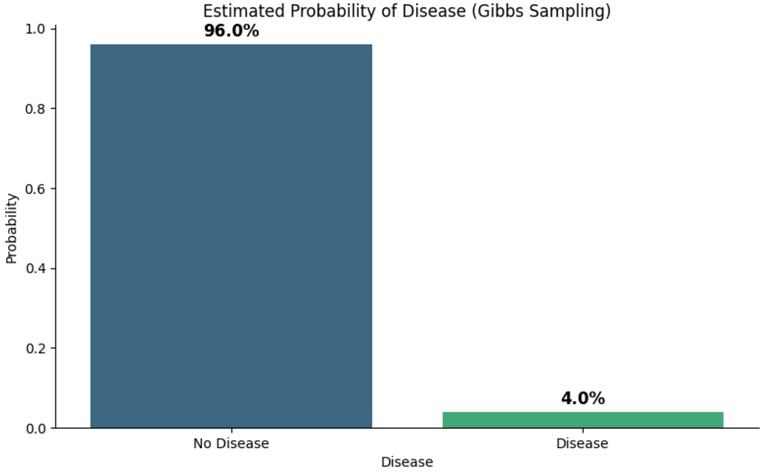
$$P(A=1) = P(A=1, B=0) + P(A=1, B=1)$$

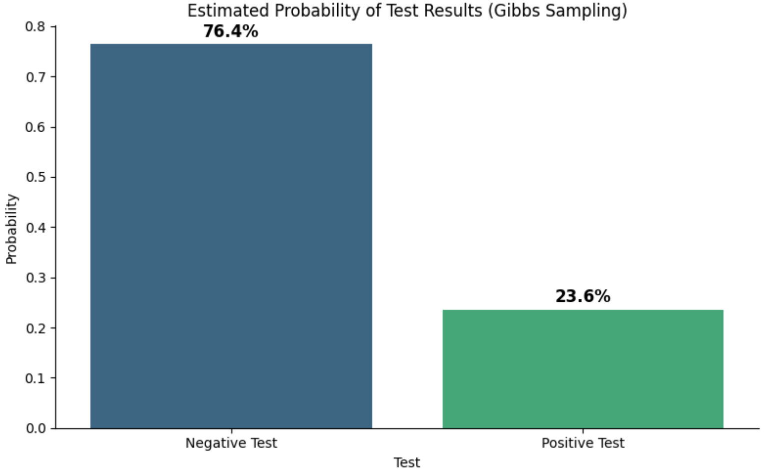
$$= \mathbf{0.4 + 0.1 = 0.5}$$

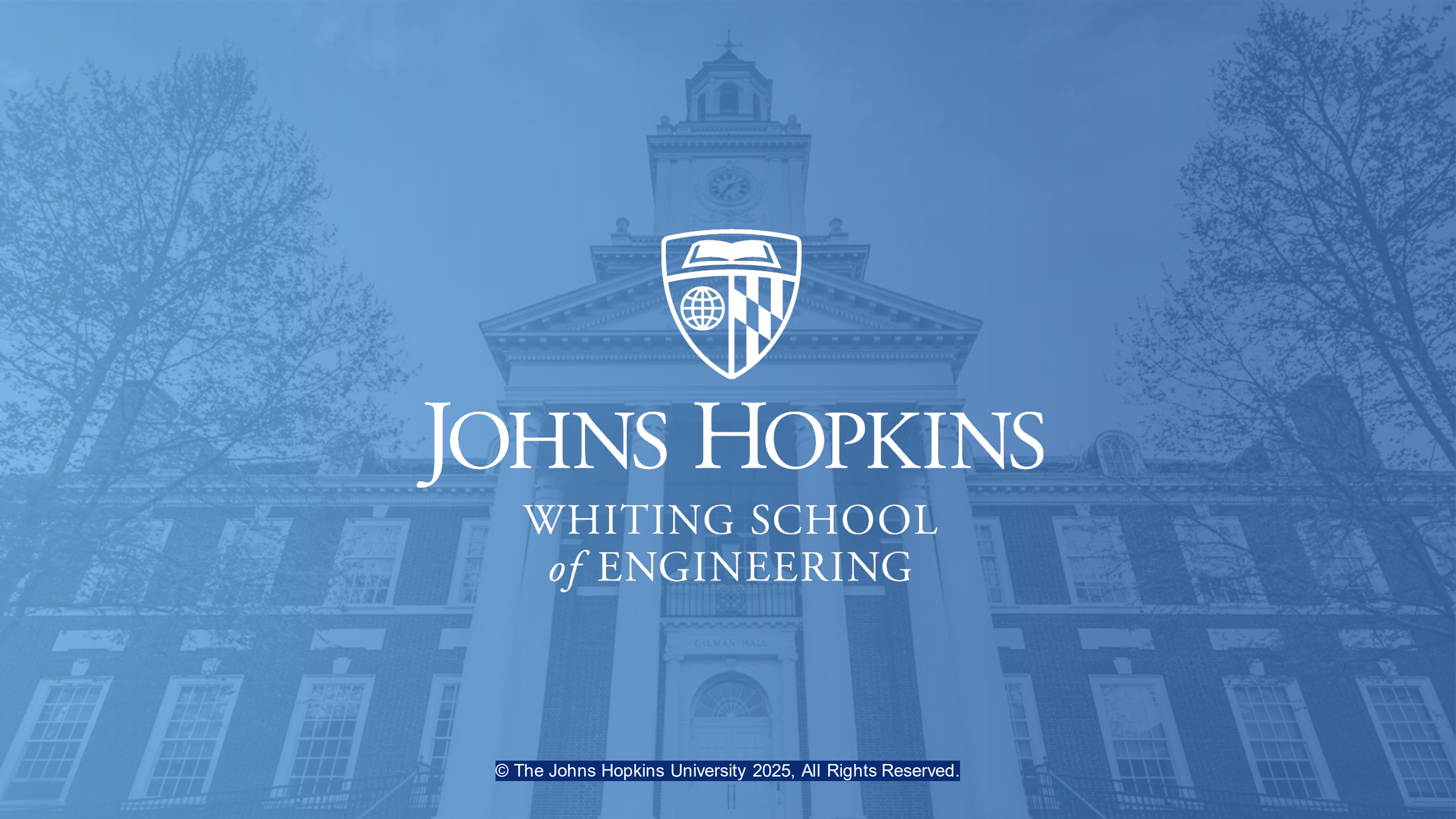
A	P(A)
0	0.5
1	0.5

Gibbs Sampling

```python
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pgmpy.models import BayesianNetwork
from pgmpy.sampling import GibbsSampling
from pgmpy.factors.discrete import TabularCPD

# Define the Bayesian Network Structure: Disease -> Test
model = BayesianNetwork([('Disease', 'Test')])

# Define Conditional Probability Distributions (CPDs)
cpd_disease = TabularCPD(variable='Disease', variable_card=2, values=[[0.95], [0.05]])  # P(D)
cpd_test = TabularCPD(variable='Test', variable_card=2,
                      values=[[0.80, 0.10],  # P(T=0 | D=0), P(T=0 | D=1)
                              [0.20, 0.90]],  # P(T=1 | D=0), P(T=1 | D=1)
                      evidence=['Disease'], evidence_card=[2])

# Add CPDs to the model
model.add_cpds(cpd_disease, cpd_test)

# Verify the model
assert model.check_model()

# Perform Gibbs Sampling
inference = GibbsSampling(model)
samples = inference.sample(size=500)

# Compute frequency of outcomes
disease_counts = samples['Disease'].value_counts(normalize=True)
test_counts = samples['Test'].value_counts(normalize=True)
```







JOHNS HOPKINS

WHITING SCHOOL of ENGINEERING