

4:

Bayesian Networks

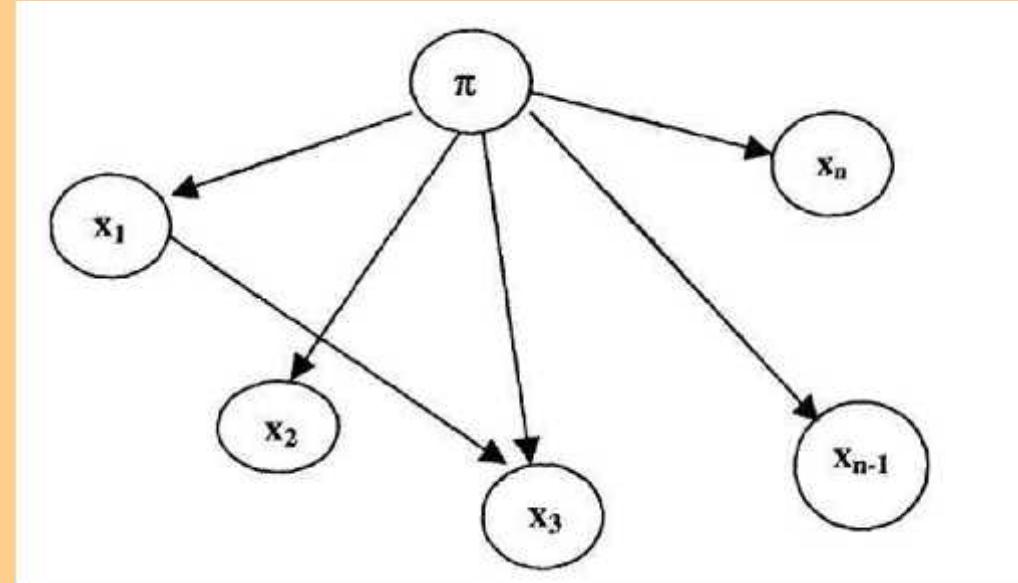
: Mushroom Classification



: 8 : -307 -23

Bayesian Networks

$$P(A|B) = P(B|A) \cdot P(A) / P(B)$$



Mushroom Classification

class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	
0	p	x	s	n	t	p	f	c	n	k
1	e	x	s	y	t	a	f	c	b	k
2	e	b	s	w	t	i	f	c	b	n

stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
s	w	w	p	w	o	p	k	s	u
s	w	w	p	w	o	p	n	n	g
s	w	w	p	w	o	p	n	n	m

: Kaggle
: 8124

: , , , , .(22)
:

: "odor"

```
import pandas as pd

path = 'mushrooms.csv'
data = pd.read_csv(path)
```

```
data.shape  
(8124, 23)
```

8124

, 23

(
object)

: class e(), p()

data.info()

data.head(3)

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color
0	p	x	s	n	t	p	f	c	n	k
1	e	x	s	y	t	a	f	c	b	k
2	e	b	s	w	t	i	f	c	b	n

stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color	ring-number	ring-type	spore-print-color	population	habitat
s	w	w	p	w	o	p	k	s	u
s	w	w	p	w	o	p	n	n	g
s	w	w	p	w	o	p	n	n	m

#	column	Non-Null Count	Dtype
0	class	8124	non-null
1	cap-shape	8124	non-null
2	cap-surface	8124	non-null
3	cap-color	8124	non-null
4	bruises	8124	non-null
5	odor	8124	non-null
6	gill-attachment	8124	non-null
7	gill-spacing	8124	non-null
8	gill-size	8124	non-null
9	gill-color	8124	non-null
10	stalk-shape	8124	non-null
11	stalk-root	8124	non-null
12	stalk-surface-above-ring	8124	non-null
13	stalk-surface-below-ring	8124	non-null
14	stalk-color-above-ring	8124	non-null
15	stalk color below ring	8124	non-null
16	veil-type	8124	non-null
17	veil-color	8124	non-null
18	ring-number	8124	non-null
19	ring-type	8124	non-null
20	population	8124	non-null
21	habitat	8124	non-null

dtypes: object(23)

1. « » :
 veil-type — 'p',
 stalk-root — 2480 «?»,

```
data = data.drop(['stalk-root', 'veil-type'], axis=1)
```

```
data.shape (8124, 21)
```

2. :
 LabelEncoder — 'x', 'f',
 'k' . . 0, 1, 2...

```
data['veil-type'].value_counts()
veil-type
p    8124
Name: count, dtype: int64
```

Пропуски по столбцам:
 stalk-root 2480

```
le = LabelEncoder()
for col in data.columns:
    data[col] = le.fit_transform(data[col])
```

```
data.head(3)
```

	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-above-ring	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-color	ring-number	ring-type	spore-print-color	population	habitat
0	1	5	2	4	1	6	1	0	1	4	...	2	2	7	7	2	1	4	2	3	5
1	0	5	2	9	1	0	1	0	0	4	...	2	2	7	7	2	1	4	3	2	1
2	0	0	2	8	1	3	1	0	0	5	...	2	2	7	7	2	1	4	3	2	3

3 rows × 21 columns

Bayesian Network

Байесовская сеть для классификации грибов (Mushroom Dataset)
Красный — class (ядовитость), Желтый — odor (запах)

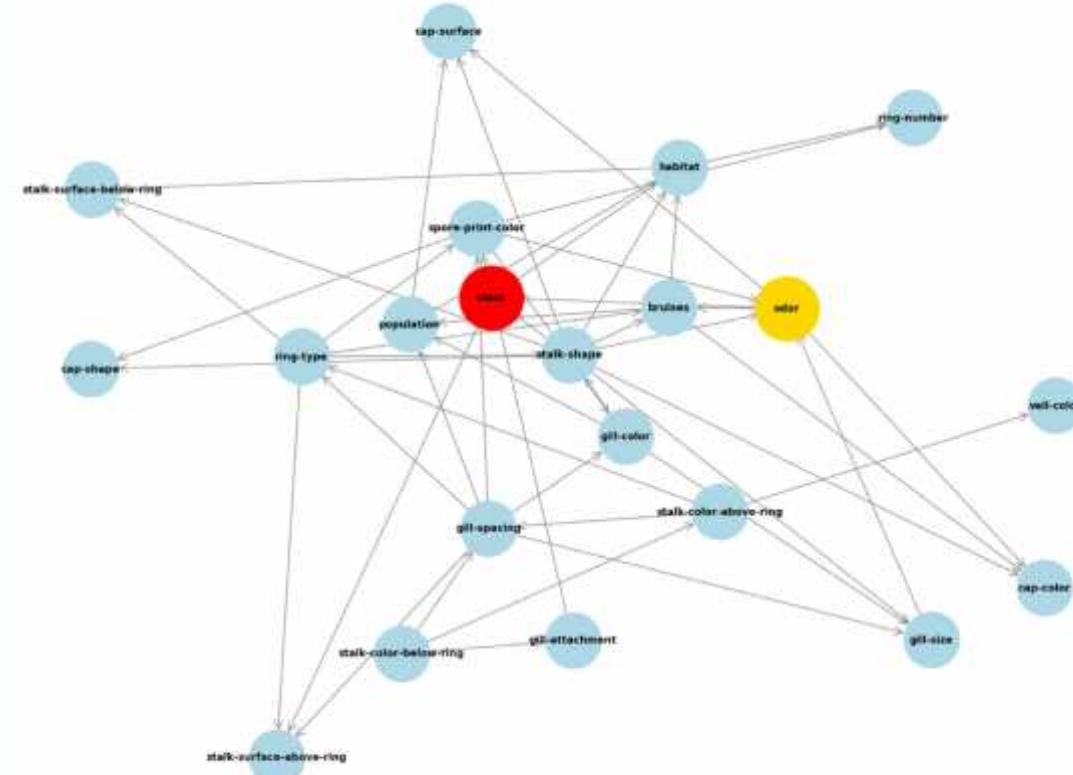
```
hc = HillClimbSearch(data)
best_model_structure = hc.estimate(scoring_method=DicScore(data))

model = BayesianNetwork(best_model_structure.edges())
```

Найдено узлов: 21

Найдено рёбер: 48

- _____ — class
(_____ / _____).
- _____ — odor (_____) —



BDeu-

BayesianEstimator

21

(CPT)

,

```
from pgmpy.estimators import BayesianEstimator  
  
model.fit(data,  
           estimator=BayesianEstimator,  
           prior_type='BDeu',  
           equivalent_sample_size=10)
```

(CPT)

```
for cpd in model.cpd:
    print(f"\n{'='*60}")
    print(f"Таблица СРТ для узла: {cpd.variable} (всего состояний: {cpd.variable_card})")
    if len(cpd.variables) > 1:
        print(f"Зависит от: {' '.join(cpd.variables[:-1])}")
    else:
        print("этот узел не имеет родителей (корневой)")
    print(f"{'='*60}")
    print(cpd)
    print("\n")
```

Таблица СРТ для узла: class (всего состояний: 2)		
Зависит от: class, odor		
odor	...	odor(8)
stalk-shape	...	stalk-shape(1)
class(0)	...	0.0004817883985353632
class(1)	...	0.9995182116014646

Таблица СРТ для узла: gill-attachment (всего состояний: 2)		
Зависит от: gill-attachment		
stalk-color-below-ring	...	stalk-color-below-ring(8)
gill-attachment(0)	...	0.022123893805309734
gill-attachment(1)	...	0.9778761061946902

```

import networkx as nx
import matplotlib.pyplot as plt

# Увеличим размер графа — у него 21 узел и 47 ребер
plt.figure(figsize=(16, 12))

# Создаем ориентированный граф
nx_graph = nx.DiGraph(nx.edges())

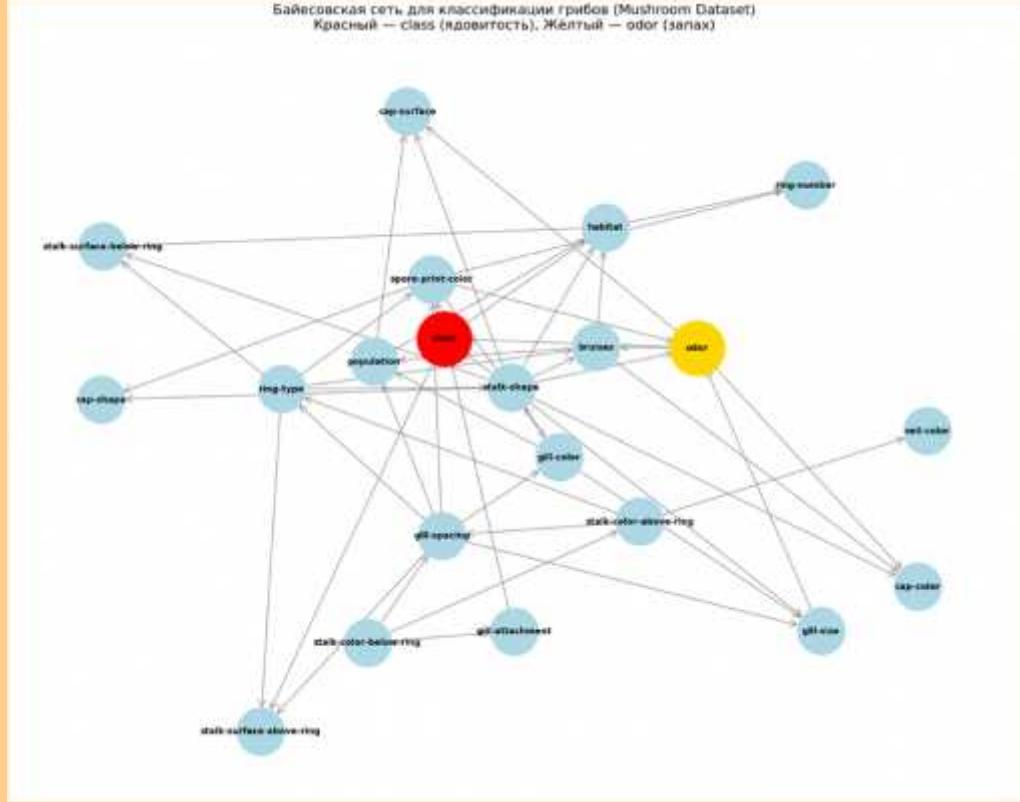
# Раскладка — spring layout с цветом каждого узла
pos = nx.spring_layout(nx_graph, seed=12, k=1, iterations=50)

# Рисуем
nx.draw(
    nx_graph,
    pos,
    with_labels=True,
    node_size=3000,
    node_color='lightblue',
    font_size=18,
    font_weight='bold',
    arrows=True,
    arrowstyle='->',
    arrowsize=20,
    edge_color='gray'
)

# Подсвечиваем ключевые узлы
nx.draw_networkx_nodes(nx_graph, pos, nodelist=['class'], node_color='red', node_size=4000)
nx.draw_networkx_nodes(nx_graph, pos, nodelist=['odor'], node_color='gold', node_size=4000)

plt.title("Байесовская сеть для классификации грибов (Mushroom Dataset)\n"
          "Красный — class (ядовитость), Желтый — odor (запах)", fontsize=16, pad=20)
plt.axis('off')
plt.tight_layout()
plt.show()

```



Inference

ПРИМЕРЫ ВЕРОЯТНОСТНОГО ВЫВОДА: определение съедобности гриба

1. Нет запаха + коричневый отпечаток спор (типичный шампиньон):

Условия: {'odor': 5, 'grole-print-color': 2}
→ P(съедобный) = 99.6498%
→ P(ядовитый) = 0.3502%
⇒ Гриб СЪЕДОБНЫЙ ✅ (уверенность 99.65%)

2. Запах foul (воняющий) – почти всегда ядовитый:

Условия: {'odor': 4}
→ P(съедобный) = 19.1835%
→ P(ядовитый) = 80.8165%
⇒ Гриб ЯДОВИТЫЙ ✗ (уверенность 80.82%)

3. Запах креозота (creosote) – 100% ядовитый в реальности:

Условия: {'odor': 2}
→ P(съедобный) = 0.0257%
→ P(ядовитый) = 99.9743%
⇒ Гриб ЯДОВИТЫЙ ✗ (уверенность 99.97%)

```
from pyprob.Inference import VariableElimination
infer = VariableElimination(model)

print("\n" * 88)
print("ПРИМЕРЫ ВЕРОЯТНОСТНОГО ВЫВОДА: определение съедобности гриба")
print("\n" * 88)

def predict_mushroom(evidence):
    result = infer.query(variables=['class'], evidence=evidence)
    prob_edible = result.values[0]
    prob_poisonous = result.values[1]

    print("\n\n" + "Условия: " + str(evidence))
    print("→ P(съедобный) = " + str(prob_edible) + "%")
    print("→ P(ядовитый) = " + str(prob_poisonous) + "%")
    if prob_poisonous >= 0.5:
        print("→ Гриб ЯДОВИТЫЙ ✗ (уверенность: " + str(prob_poisonous) + "%)")
    else:
        print("→ Гриб СЪЕДОБНЫЙ ✅ (уверенность: " + str(prob_edible) + "%)")

# Использование примеров
print("\n1. Нет запаха + коричневый отпечаток спор (типичный шампиньон):")
predict_mushroom({'odor': 5, 'grole-print-color': 2})

print("\n2. Запах foul (воняющий) – почти всегда ядовитый:")
predict_mushroom({'odor': 4})

print("\n3. Запах креозота (creosote) – 100% ядовитый в реальности:")
predict_mushroom({'odor': 2})

print("\n4. Типичный запах миндаля (almond) – что скажет сеть?")
predict_mushroom({'odor': 0})

print("\n5. Только цвет жабр бурый (buff) – что скажет сеть?")
predict_mushroom({'gill-color': 0})

print("\n6. Ничего не знаем (априорная вероятность):")
predict_mushroom({})
```

4. Запах миндаля – почти всегда съедобный:

Условия: {'odor': 0}
→ P(съедобный) = 99.8789%
→ P(ядовитый) = 0.1211%
⇒ Гриб СЪЕДОБНЫЙ ✅ (уверенность 99.88%)

5. Только цвет жабр бурый (buff) – что скажет сеть?

Условия: {'gill-color': 0}
→ P(съедобный) = 1.8578%
→ P(ядовитый) = 98.1422%
⇒ Гриб ЯДОВИТЫЙ ✗ (уверенность 98.14%)

6. Ничего не знаем (априорная вероятность):

Условия: {}
→ P(съедобный) = 52.8418%
→ P(ядовитый) = 47.1582%
⇒ Гриб СЪЕДОБНЫЙ ✅ (уверенность 52.84%)

Baseline

```
Naive Bayes (sklearn) → Accuracy: 0.95980 | Log-loss: 0.11317  
Байесовская сеть (pgmpy) → Accuracy: 1.00000 | Log-loss: 0.00147
```

ИТОГОВОЕ СРАВНЕНИЕ

Модель	Accuracy	Log-loss
Naive Bayes (sklearn)	0.95980	0.11317
Байесовская сеть	1.00000	0.00147

scikit-learn _____

: accuracy = 95.98% log-loss = 0.113.

, pgmpy, accuracy = 100.0% log-loss = 0.00147.

, (HillClimbSearch +
BIC) (, odor, spore-
print-color, gill-color class),

, (log-loss), ,