Лабораторная работа №4

Тема: “Лабораторная работа: Bayesian Networks на примере датасета Mushroom Classification”

Таланкин Кирилл Александрович
М8О-309Б-23

Bayesian Networks

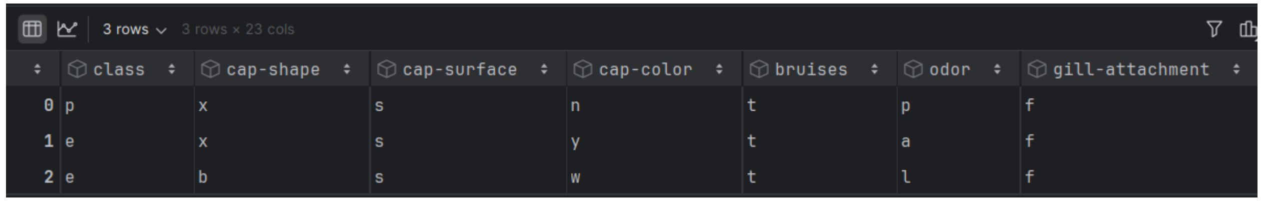
Bayesian Network - это вероятностная графовая модель, у которой узлы это переменные, причино-следственные зависимости обозначаются стрелками. Количественные зависимости задаются таблицей условных вероятностей

Bayesian Network используется для:

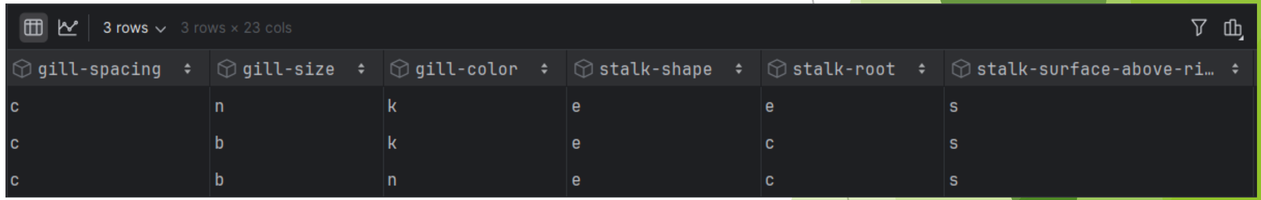
- Классификации
- Прогнозирования
- Моделирования неопределенности
- Объяснения зависимости между признаками

Описание датасета

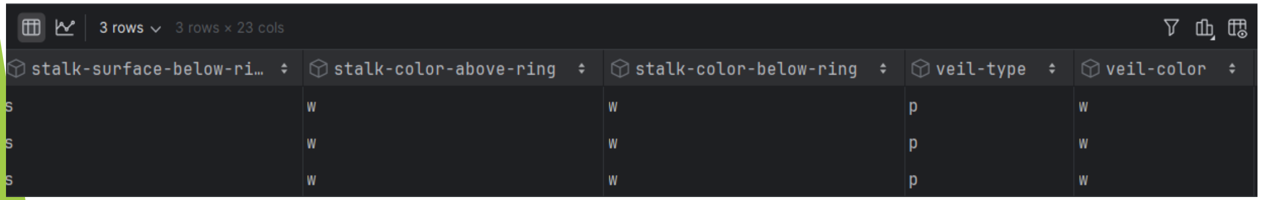
- Датасет: Mushroom Classification. Он содержит 8124 строки, 23 столбца, из которых class и 22 признака
- Признаки: все категориальные
- Классы: съедобные 52%, ядовитые 48%



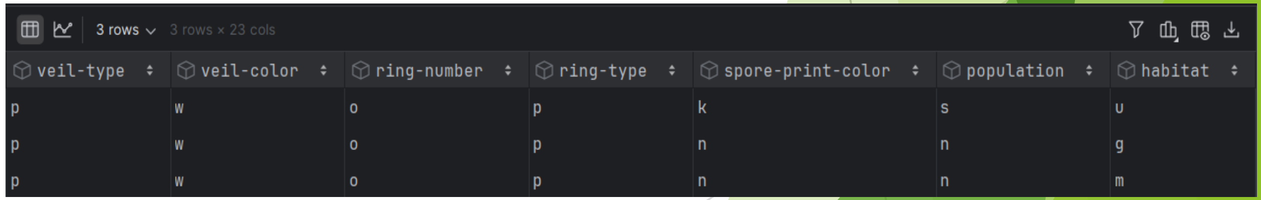
	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment
0	p	x	s	n	t	p	f
1	e	x	s	y	t	a	f
2	e	b	s	w	t	l	f



gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surface-above-ri…
c	n	k	e	e	s
c	b	k	e	c	s
c	b	n	e	c	s

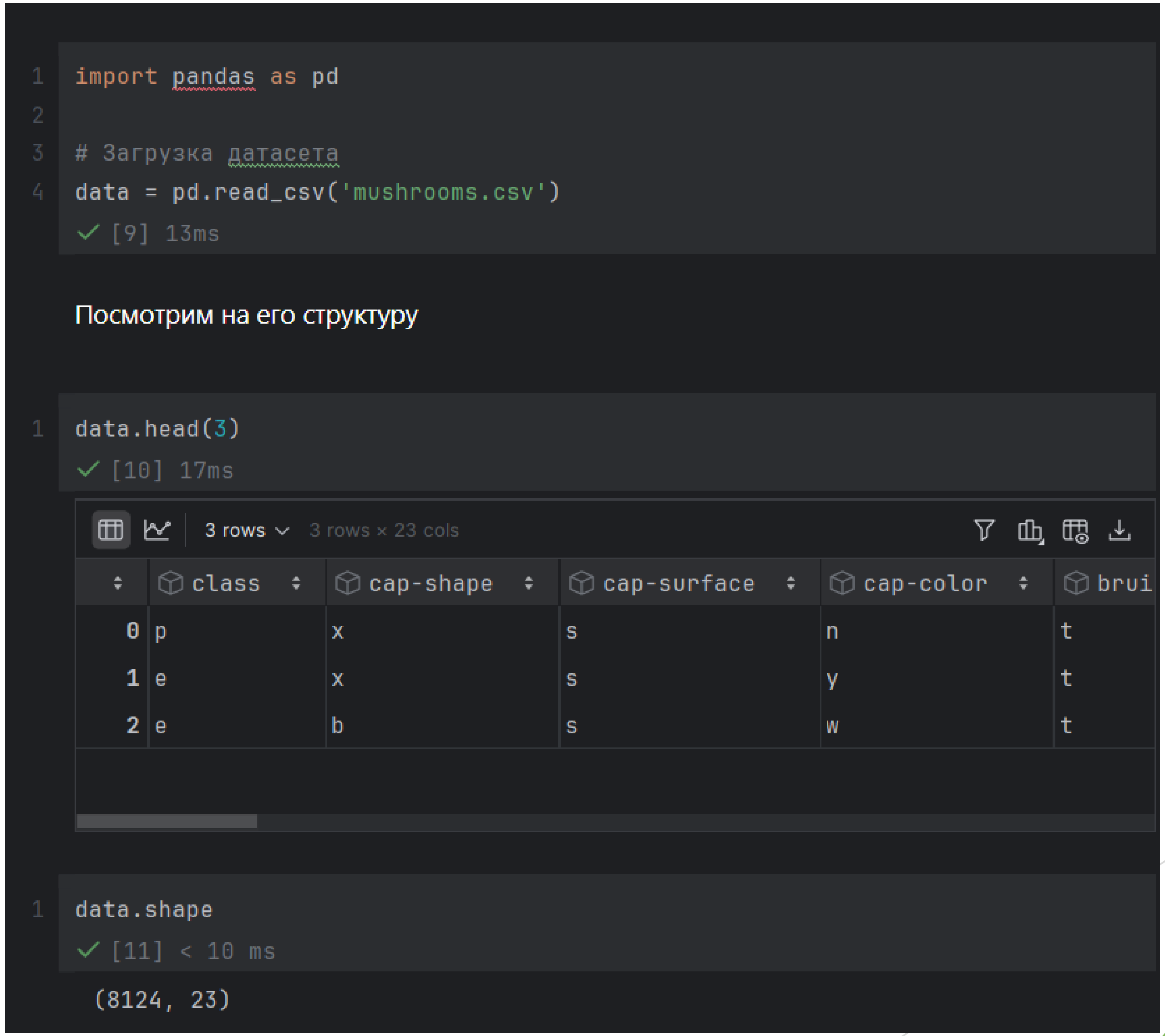


stalk-surface-below-ri…	stalk-color-above-ring	stalk-color-below-ring	veil-type	veil-color
s	w	w	p	w
s	w	w	p	w
s	w	w	p	w



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

Загрузка датасета



```python
import pandas as pd

# Загрузка датасета
data = pd.read_csv('mushrooms.csv')
```
✓ [9] 13ms

Посмотрим на его структуру

```python
data.head(3)
```
✓ [10] 17ms

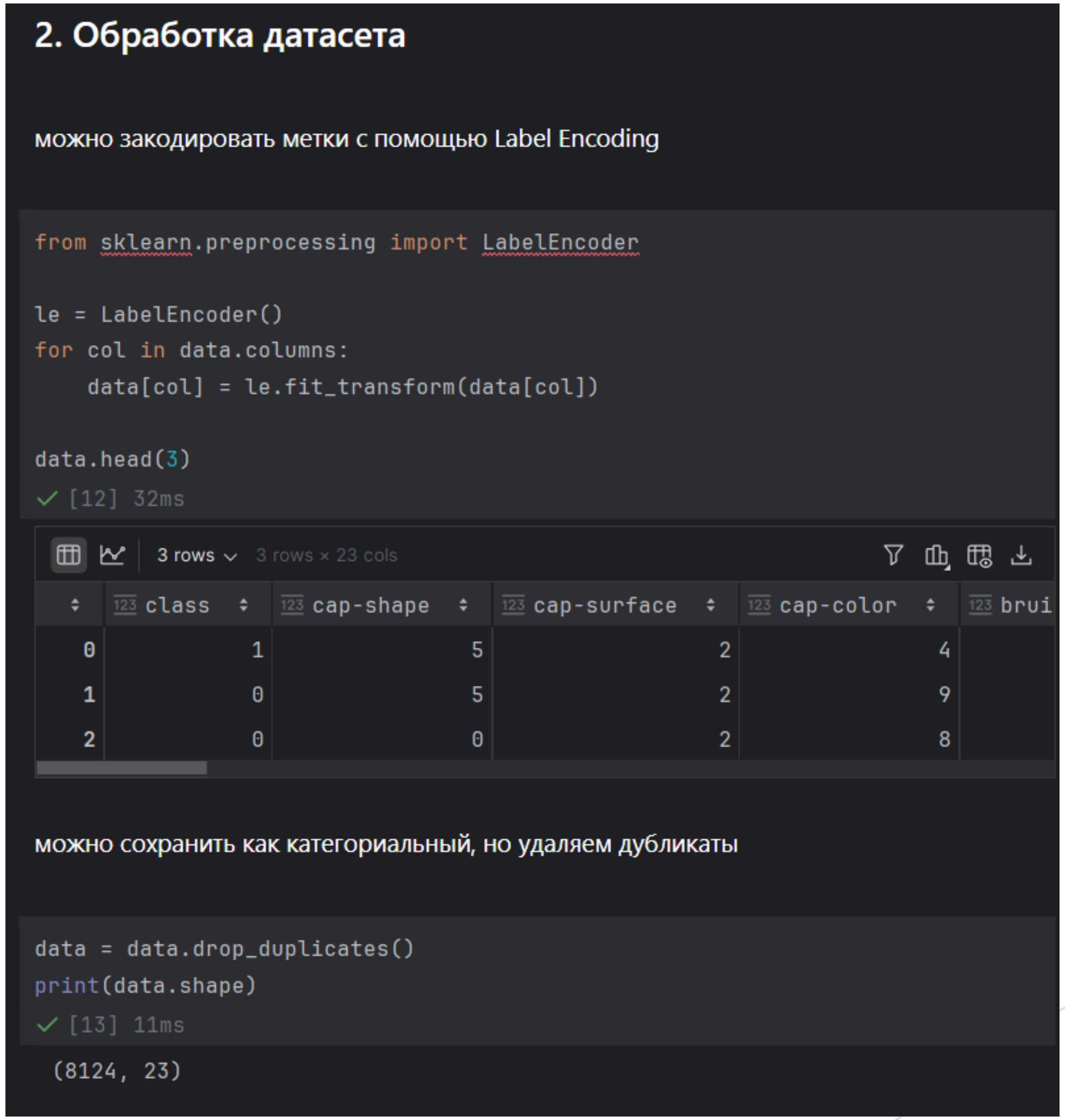
3 rows 3 rows × 23 cols

	class	cap-shape	cap-surface	cap-color	brui
0	p	x	s	n	t
1	e	x	s	y	t
2	e	b	s	w	t

```python
data.shape
```
✓ [11] < 10 ms

```
(8124, 23)
```

Обработка датасета



2. Обработка датасета

можно закодировать метки с помощью Label Encoding

```
from sklearn.preprocessing import LabelEncoder

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

data.head(3)
```

✓ [12] 32ms

3 rows 3 rows × 23 cols

	class	cap-shape	cap-surface	cap-color	brui
0	1	5	2	4	
1	0	5	2	9	
2	0	0	2	8	

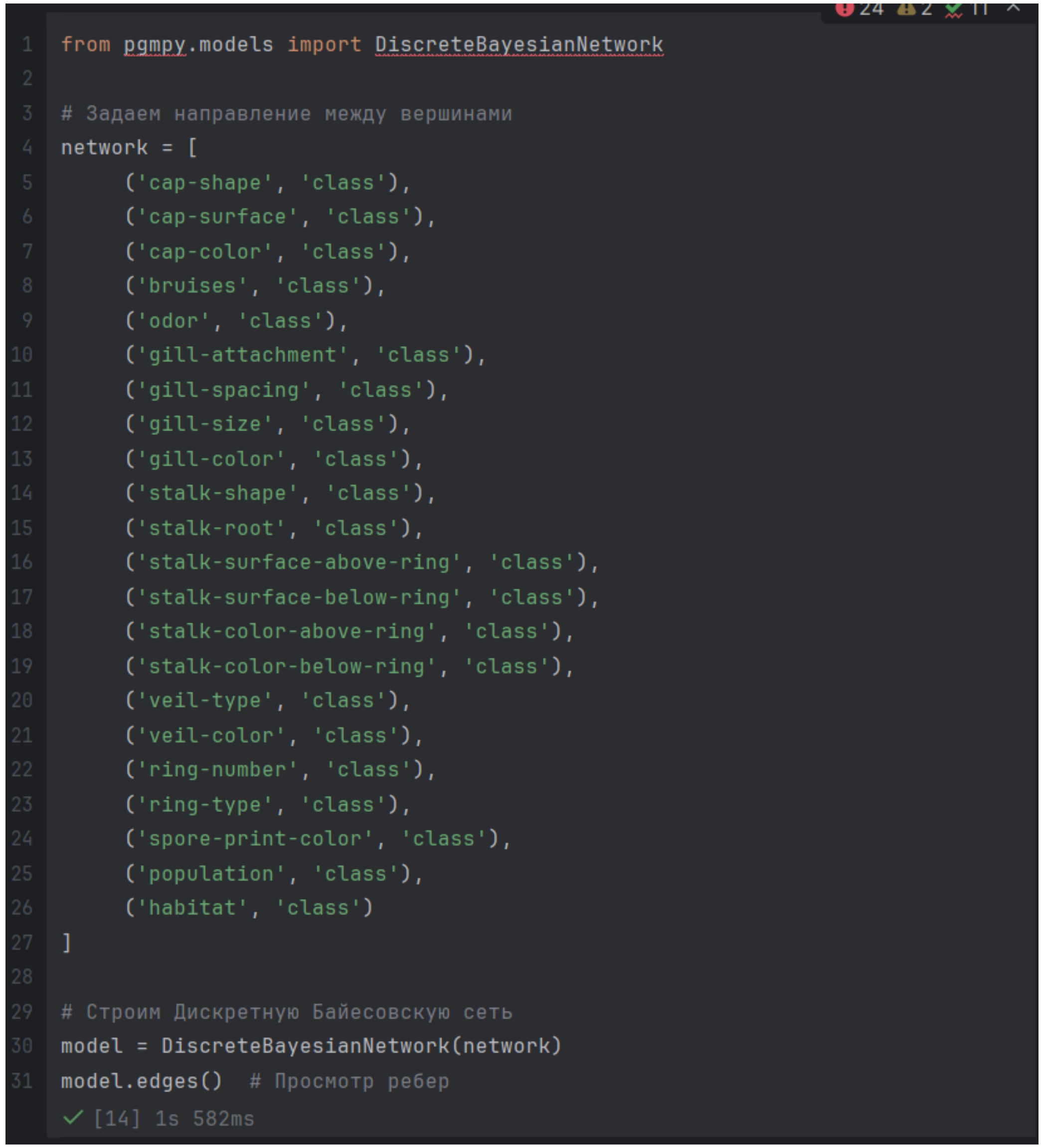
можно сохранить как категориальный, но удаляем дубликаты

```
data = data.drop_duplicates()
print(data.shape)
```

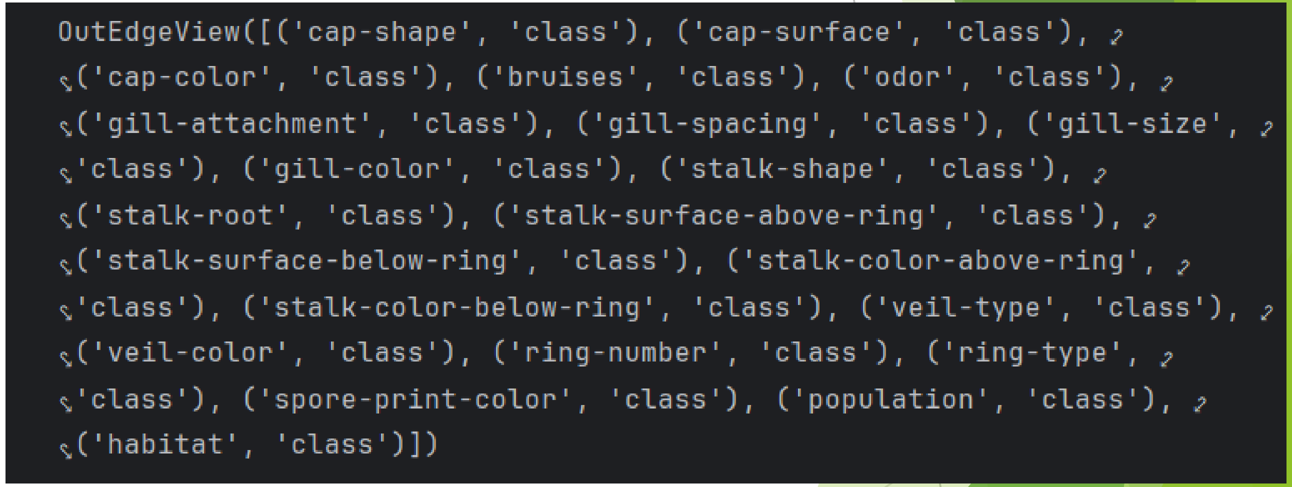
✓ [13] 11ms

```
(8124, 23)
```

Построение Bayesian Network

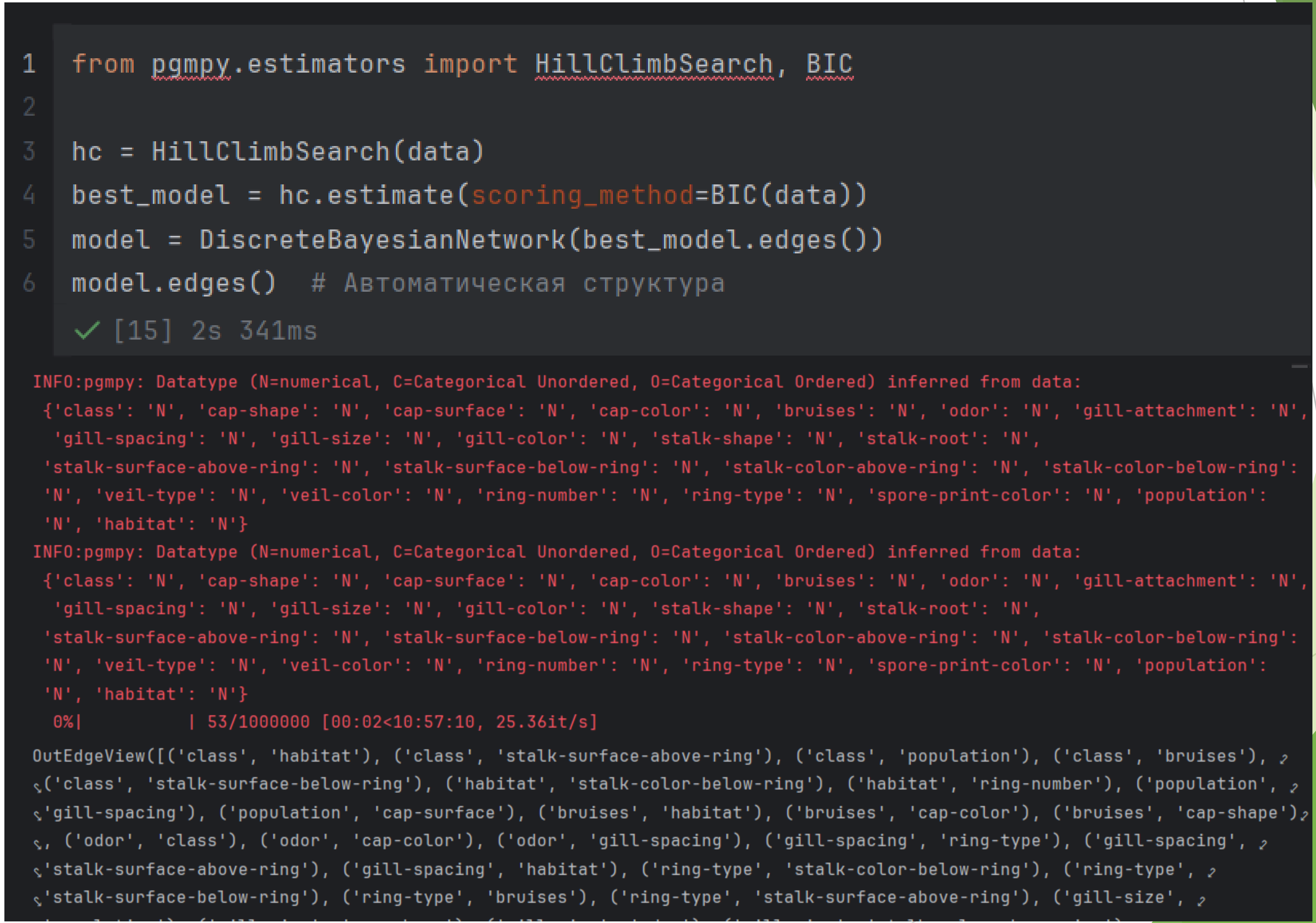


```python
from pgmpy.models import DiscreteBayesianNetwork

# Задаем направление между вершинами
network = [
     ('cap-shape', 'class'),
     ('cap-surface', 'class'),
     ('cap-color', 'class'),
     ('bruises', 'class'),
     ('odor', 'class'),
     ('gill-attachment', 'class'),
     ('gill-spacing', 'class'),
     ('gill-size', 'class'),
     ('gill-color', 'class'),
     ('stalk-shape', 'class'),
     ('stalk-root', 'class'),
     ('stalk-surface-above-ring', 'class'),
     ('stalk-surface-below-ring', 'class'),
     ('stalk-color-above-ring', 'class'),
     ('stalk-color-below-ring', 'class'),
     ('veil-type', 'class'),
     ('veil-color', 'class'),
     ('ring-number', 'class'),
     ('ring-type', 'class'),
     ('spore-print-color', 'class'),
     ('population', 'class'),
     ('habitat', 'class')
]

# Строим Дискретную Байесовскую сеть
model = DiscreteBayesianNetwork(network)
model.edges()  # Просмотр ребер
✓ [14] 1s 582ms
```



```
OutEdgeView([('cap-shape', 'class'), ('cap-surface', 'class'), 
('cap-color', 'class'), ('bruises', 'class'), ('odor', 'class'), 
('gill-attachment', 'class'), ('gill-spacing', 'class'), ('gill-size', 
'class'), ('gill-color', 'class'), ('stalk-shape', 'class'), 
('stalk-root', 'class'), ('stalk-surface-above-ring', 'class'), 
('stalk-surface-below-ring', 'class'), ('stalk-color-above-ring', 
'class'), ('stalk-color-below-ring', 'class'), ('veil-type', 'class'), 
('veil-color', 'class'), ('ring-number', 'class'), ('ring-type', 
'class'), ('spore-print-color', 'class'), ('population', 'class'), 
('habitat', 'class')])
```

Построение Bayesian Network

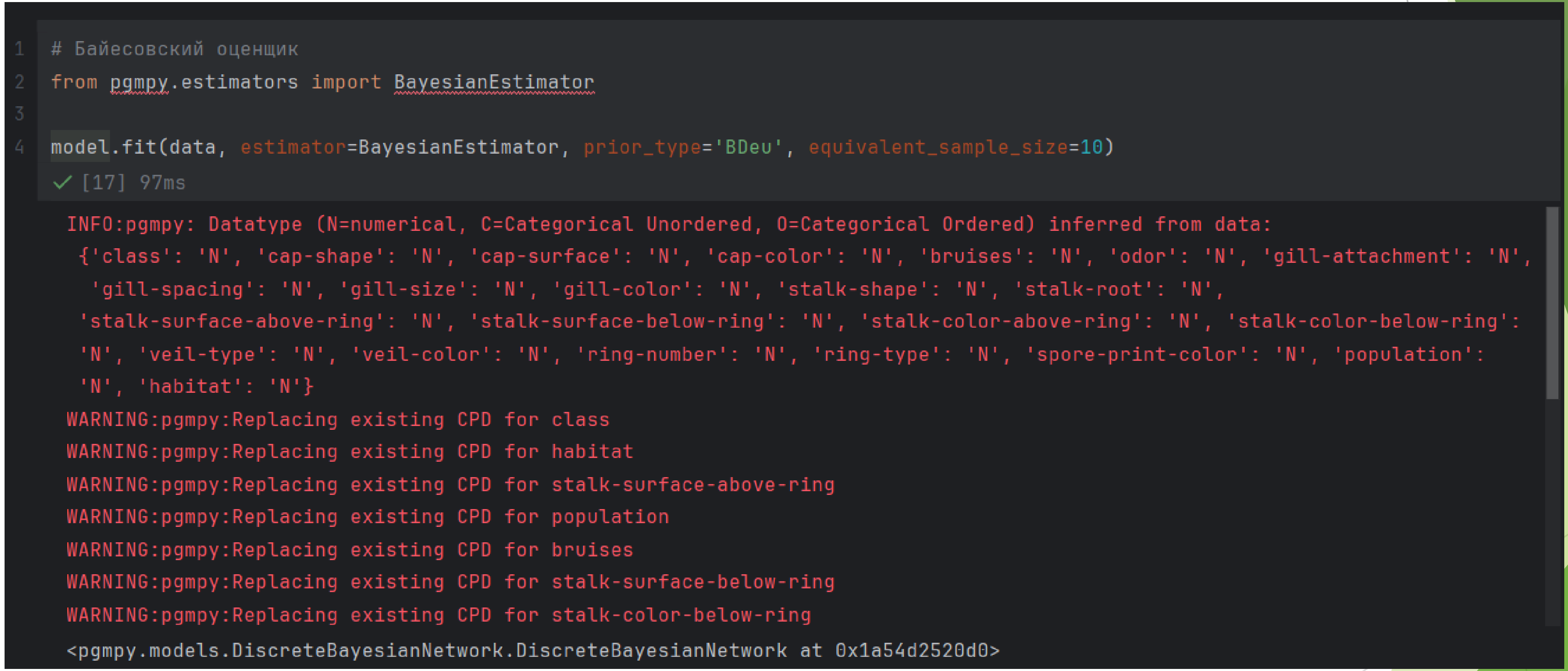


```python
from pgmpy.estimators import HillClimbSearch, BIC

hc = HillClimbSearch(data)
best_model = hc.estimate(scoring_method=BIC(data))
model = DiscreteBayesianNetwork(best_model.edges())
model.edges()  # Автоматическая структура
```

✓ [15] 2s 341ms

```
INFO:pgmpy: Datatype (N=numerical, C=Categorical Unordered, O=Categorical Ordered) inferred from data:
 {'class': 'N', 'cap-shape': 'N', 'cap-surface': 'N', 'cap-color': 'N', 'bruises': 'N', 'odor': 'N', 'gill-attachment': 'N',
  'gill-spacing': 'N', 'gill-size': 'N', 'gill-color': 'N', 'stalk-shape': 'N', 'stalk-root': 'N',
 'stalk-surface-above-ring': 'N', 'stalk-surface-below-ring': 'N', 'stalk-color-above-ring': 'N', 'stalk-color-below-ring':
 'N', 'veil-type': 'N', 'veil-color': 'N', 'ring-number': 'N', 'ring-type': 'N', 'spore-print-color': 'N', 'population':
 'N', 'habitat': 'N'}
INFO:pgmpy: Datatype (N=numerical, C=Categorical Unordered, O=Categorical Ordered) inferred from data:
 {'class': 'N', 'cap-shape': 'N', 'cap-surface': 'N', 'cap-color': 'N', 'bruises': 'N', 'odor': 'N', 'gill-attachment': 'N',
  'gill-spacing': 'N', 'gill-size': 'N', 'gill-color': 'N', 'stalk-shape': 'N', 'stalk-root': 'N',
 'stalk-surface-above-ring': 'N', 'stalk-surface-below-ring': 'N', 'stalk-color-above-ring': 'N', 'stalk-color-below-ring':
 'N', 'veil-type': 'N', 'veil-color': 'N', 'ring-number': 'N', 'ring-type': 'N', 'spore-print-color': 'N', 'population':
 'N', 'habitat': 'N'}
  0%|          | 53/1000000 [00:02<10:57:10, 25.36it/s]

OutEdgeView([('class', 'habitat'), ('class', 'stalk-surface-above-ring'), ('class', 'population'), ('class', 'bruises'), ↩
↪('class', 'stalk-surface-below-ring'), ('habitat', 'stalk-color-below-ring'), ('habitat', 'ring-number'), ('population', ↩
↪'gill-spacing'), ('population', 'cap-surface'), ('bruises', 'habitat'), ('bruises', 'cap-color'), ('bruises', 'cap-shape')↩
↪, ('odor', 'class'), ('odor', 'cap-color'), ('odor', 'gill-spacing'), ('gill-spacing', 'ring-type'), ('gill-spacing', ↩
↪'stalk-surface-above-ring'), ('gill-spacing', 'habitat'), ('ring-type', 'stalk-color-below-ring'), ('ring-type', ↩
↪'stalk-surface-below-ring'), ('ring-type', 'bruises'), ('ring-type', 'stalk-surface-above-ring'), ('gill-size', ↩
```

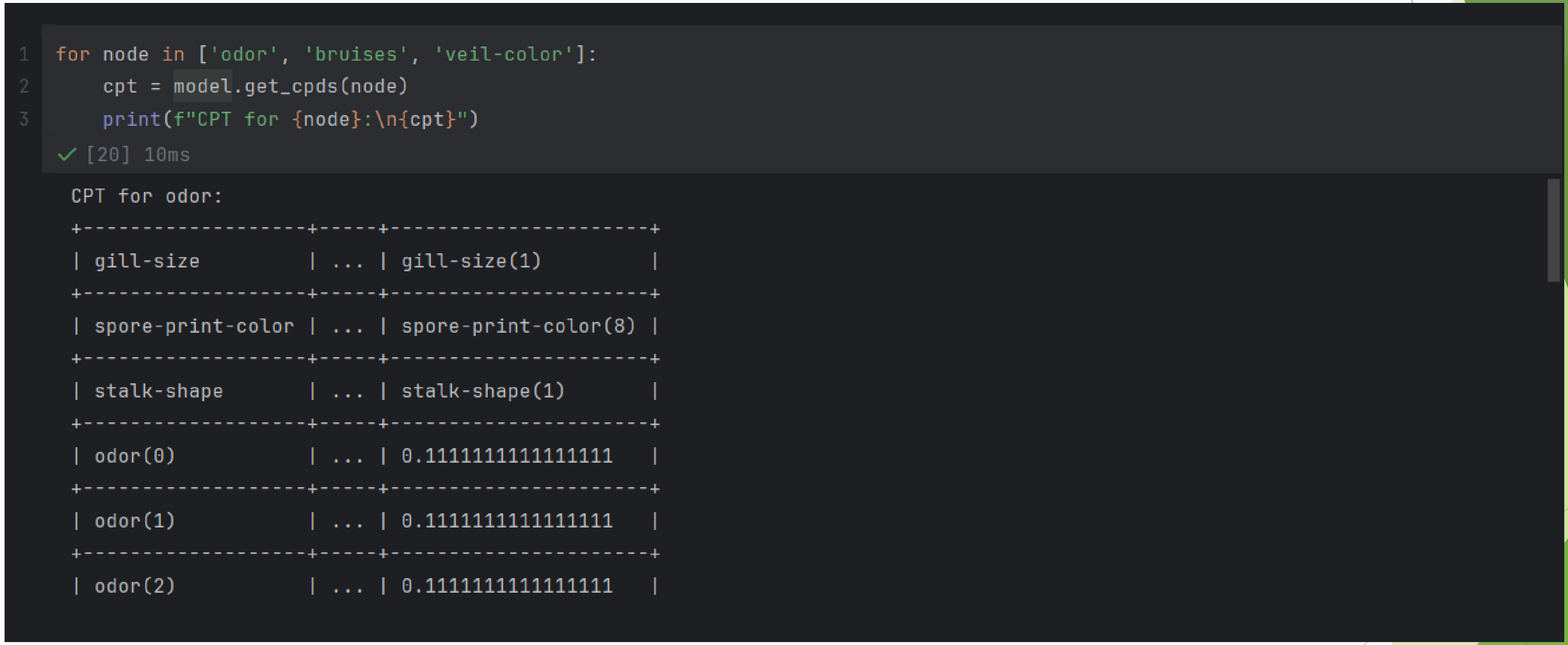
Оценка параметров и CPT



```
# Байесовский оценщик
from pgmpy.estimators import BayesianEstimator

model.fit(data, estimator=BayesianEstimator, prior_type='BDeu', equivalent_sample_size=10)
✓ [17] 97ms
```

```
INFO:pgmpy: Datatype (N=numerical, C=Categorical Unordered, O=Categorical Ordered) inferred from data:
 {'class': 'N', 'cap-shape': 'N', 'cap-surface': 'N', 'cap-color': 'N', 'bruises': 'N', 'odor': 'N', 'gill-attachment': 'N',
  'gill-spacing': 'N', 'gill-size': 'N', 'gill-color': 'N', 'stalk-shape': 'N', 'stalk-root': 'N',
 'stalk-surface-above-ring': 'N', 'stalk-surface-below-ring': 'N', 'stalk-color-above-ring': 'N', 'stalk-color-below-ring':
 'N', 'veil-type': 'N', 'veil-color': 'N', 'ring-number': 'N', 'ring-type': 'N', 'spore-print-color': 'N', 'population':
 'N', 'habitat': 'N'}
WARNING:pgmpy:Replacing existing CPD for class
WARNING:pgmpy:Replacing existing CPD for habitat
WARNING:pgmpy:Replacing existing CPD for stalk-surface-above-ring
WARNING:pgmpy:Replacing existing CPD for population
WARNING:pgmpy:Replacing existing CPD for bruises
WARNING:pgmpy:Replacing existing CPD for stalk-surface-below-ring
WARNING:pgmpy:Replacing existing CPD for stalk-color-below-ring
<pgmpy.models.DiscreteBayesianNetwork.DiscreteBayesianNetwork at 0x1a54d2520d0>
```

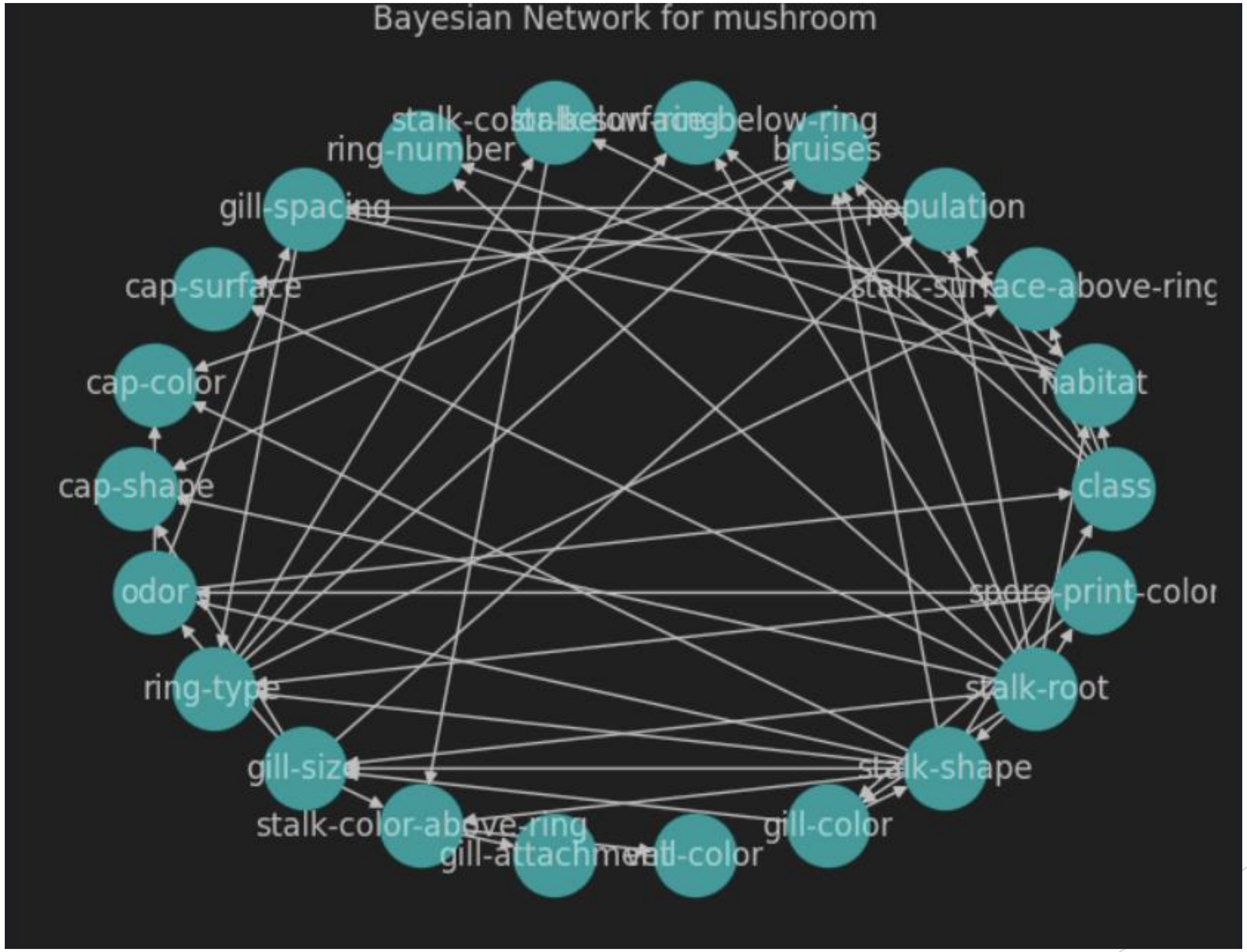
Оценка параметров и CPT



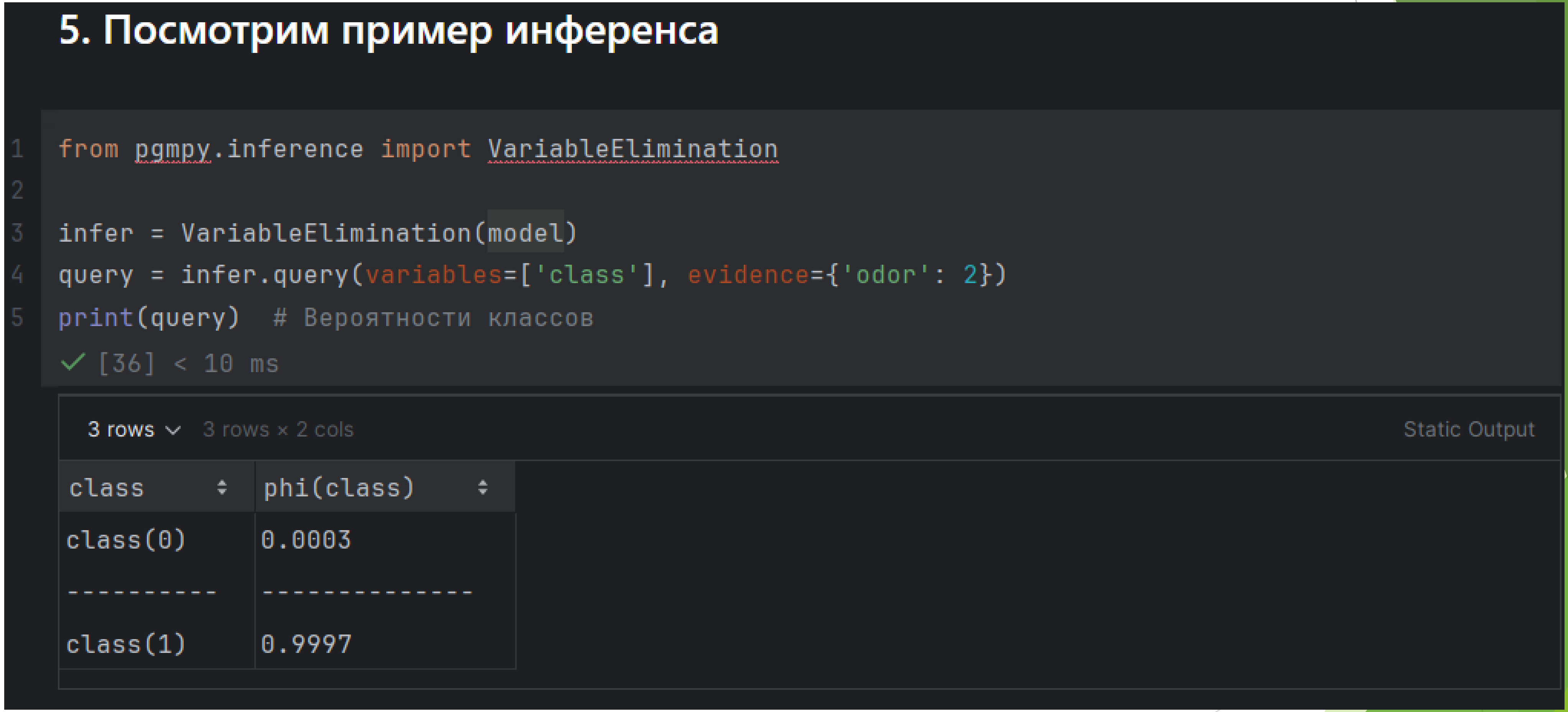
```
for node in ['odor', 'bruises', 'veil-color']:
    cpt = model.get_cpds(node)
    print(f"CPT for {node}:\n{cpt}")
✓ [20] 10ms
```

```
CPT for odor:
+--------------------+-----+----------------------+
| gill-size          | ... | gill-size(1)         |
+--------------------+-----+----------------------+
| spore-print-color  | ... | spore-print-color(8) |
+--------------------+-----+----------------------+
| stalk-shape        | ... | stalk-shape(1)       |
+--------------------+-----+----------------------+
| odor(0)            | ... | 0.1111111111111111   |
+--------------------+-----+----------------------+
| odor(1)            | ... | 0.1111111111111111   |
+--------------------+-----+----------------------+
| odor(2)            | ... | 0.1111111111111111   |
```

Визуализация сети



Пример интерфейса



5. Посмотрим пример инференса

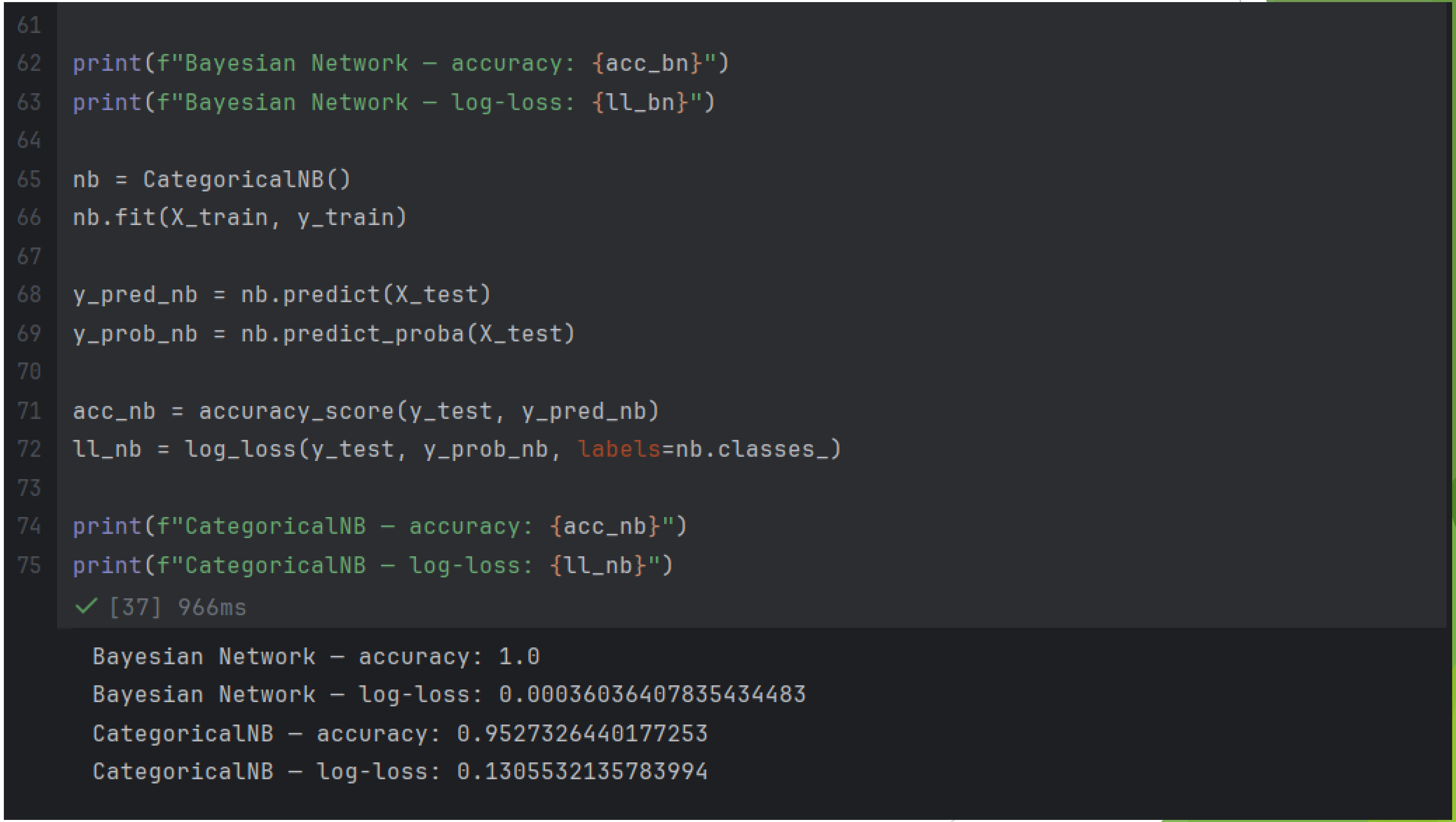
```python
from pgmpy.inference import VariableElimination

infer = VariableElimination(model)
query = infer.query(variables=['class'], evidence={'odor': 2})
print(query)  # Вероятности классов
```

✓ [36] < 10 ms

3 rows 3 rows × 2 cols Static Output

```
class        phi(class)
class(0)     0.0003
----------   --------------
class(1)     0.9997
```

Оценка с Baseline



```python

print(f"Bayesian Network — accuracy: {acc_bn}")
print(f"Bayesian Network — log-loss: {ll_bn}")

nb = CategoricalNB()
nb.fit(X_train, y_train)

y_pred_nb = nb.predict(X_test)
y_prob_nb = nb.predict_proba(X_test)

acc_nb = accuracy_score(y_test, y_pred_nb)
ll_nb = log_loss(y_test, y_prob_nb, labels=nb.classes_)

print(f"CategoricalNB — accuracy: {acc_nb}")
print(f"CategoricalNB — log-loss: {ll_nb}")
```

✓ [37] 966ms

```
Bayesian Network — accuracy: 1.0
Bayesian Network — log-loss: 0.00036036407835434483
CategoricalNB — accuracy: 0.9527326440177253
CategoricalNB — log-loss: 0.1305532135783994
```