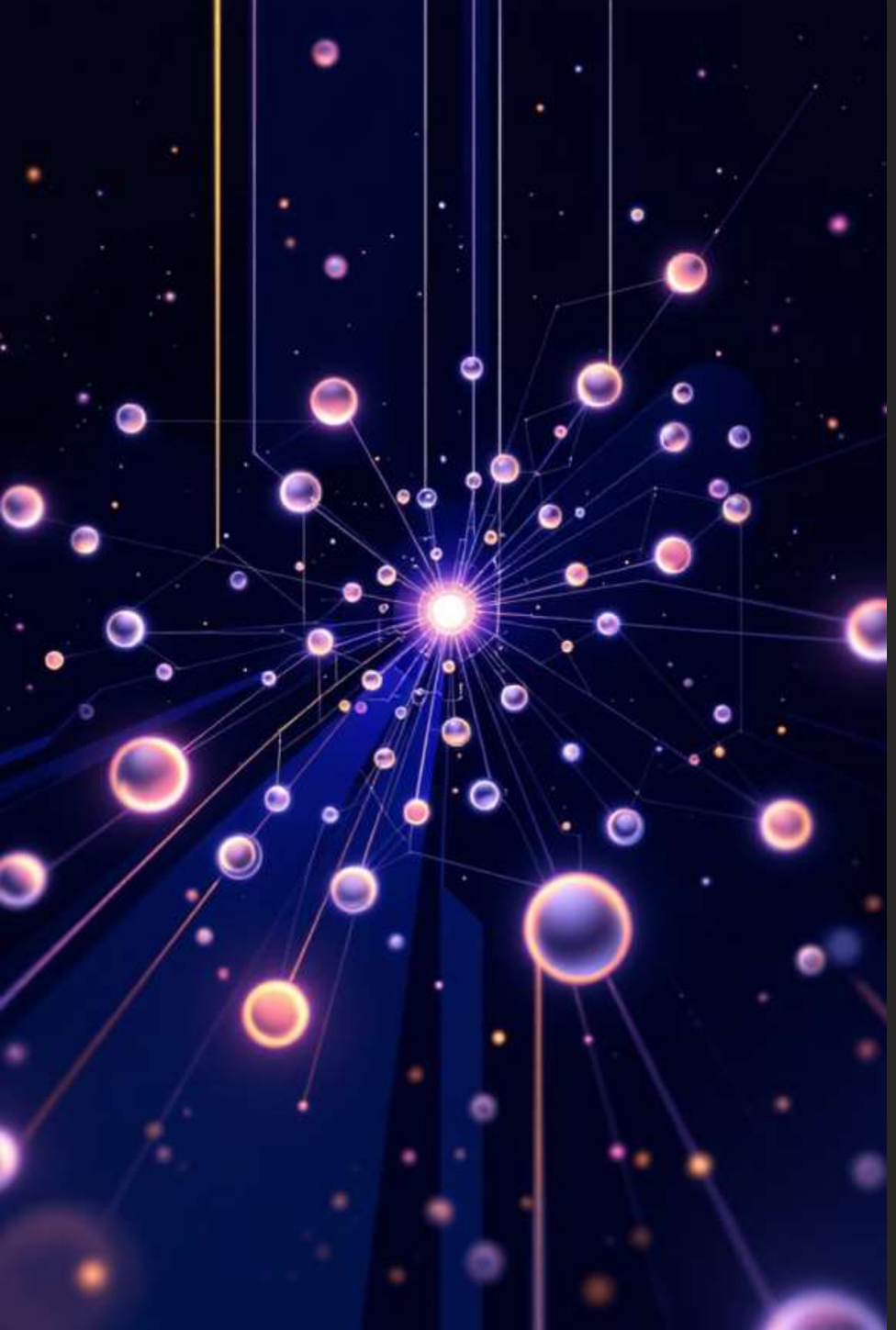


Лабораторная работа: Байесовские сети на примере датасета mushrooms

Выполнила: Софья Шароченкова, М8О-309Б-23

Что такое байесовские сети?

Байесовские сети — это вероятностные графические модели, которые представляют множество переменных и их условные зависимости через направленный ациклический граф.

Основное применение:

- Моделирование причинно-следственных связей
- Прогнозирование на основе неполных данных
- Принятие решений в условиях неопределённости

Теорема Байеса лежит в основе: $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$

Датасет Mushrooms: содержит характеристики грибов (цвет шляпки, запах, форма) для классификации на съедобные и ядовитые. Размер: 8124 образца, 23 атрибута, источник: UCI Machine Learning Repository.

Загрузка и первичный анализ данных

Загрузка датасета

```
path = Path(kagglehub.dataset_download("uciml/mushroom-classification"))
print("Path to dataset files:", path)

# Загрузка Car Evaluation (без заголовков в файле)
file_dir = path / 'mushrooms.csv'
data = pd.read_csv(file_dir, header=None, names=['class', 'cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor', 'gill-attachment', 'gill-spacing', 'gill-size', 'gill-color', 'stalk-shape', 'stalk-root', 'stalk-surface-above-ring', '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'], )
```

Основные характеристики:

- 8124 записи, 23 признака
- Все признаки категориальные
- Целевая переменная: class (e/p)

Статистика признаков

Примеры атрибутов:

- **cap-shape:** форма шляпки (bell, conical, convex, flat)
- **cap-color:** цвет шляпки (brown, yellow, white, gray)
- **bruises:** наличие синяков (t/f)
- **odor:** запах (almond, anise, creosote, fishy, foul)
- **gill-color:** цвет пластинок

Предобработка данных

01

Кодирование категорий

Применение LabelEncoder для преобразования всех категориальных признаков в числовые значения, необходимые для работы pgmpy.

02

Проверка дубликатов

Выявление и удаление повторяющихся записей для повышения качества модели (найдено и удалено 0 дубликатов).

03

Анализ пропусков

Проверка на отсутствующие значения — датасет оказался полным, дополнительная обработка не требуется.

04

Валидация структуры

Подтверждение корректности типов данных и диапазонов значений после преобразования.

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
for col in data.columns:
    data[col] = le.fit_transform(data[col])
data.head(3)



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

Построение структуры байесовской сети

Автоматическое обучение структуры

```
from pgmpy.estimators import
HillClimbSearch, BIC


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

Структура сети

Алгоритм HillClimbSearch с BIC-оценкой выявил ключевые зависимости между признаками:

- **odor → class:** запах — сильный предиктор
- **gill-color → class:** цвет пластинок важен
- **spore-print-color → class:** цвет спор
- **bruises → odor:** связь с запахом

Направленные рёбра показывают причинно-следственные связи в данных.

Оценка параметров модели

1

Maximum Likelihood Estimation

Оценка условных вероятностей на основе частот встречаемости комбинаций в обучающих данных

2

Обучение модели

Вычисление параметров CPT (таблиц условных вероятностей) для каждого узла сети

3

Валидация CPT

Проверка корректности распределений и их соответствия аксиомам вероятности

Метод Максимального Правдоподобия

```
from pgmpy.estimators import MaximumLikelihoodEstimator
model.fit(data, estimator=MaximumLikelihoodEstimator)
```

Байесовский оценщик

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

Таблицы условных вероятностей (CPT)

CPT для узла 'class'

```
cpt_class = model.get_cpds('class')
print(cpt_class)
```

Пример вероятностей:

P(class=poisonous | odor=foul) = 0.98

P(class=edible | odor=almond) = 0.95

P(class=poisonous | odor=none) = 0.32

```
CPT for ring-type:
+--------------------+-----+----------------------+
| gill-spacing       | ... | gill-spacing(1)      |
+--------------------+-----+----------------------+
| spore-print-color  | ... | spore-print-color(8) |
+--------------------+-----+----------------------+
| ring-type(0)       | ... | 0.2                  |
+--------------------+-----+----------------------+
| ring-type(1)       | ... | 0.2                  |
+--------------------+-----+----------------------+
| ring-type(2)       | ... | 0.2                  |
+--------------------+-----+----------------------+
| ring-type(3)       | ... | 0.2                  |
+--------------------+-----+----------------------+
| ring-type(4)       | ... | 0.2                  |
+--------------------+-----+----------------------+
```

CPT для узла 'odor'

```
for node in [‘odor', 'ring-type', 'gill-color']:
    cpt = model.get_cpds(node)
    print(f"CPT for {node}:\n{cpt}")
```

Ключевые наблюдения:

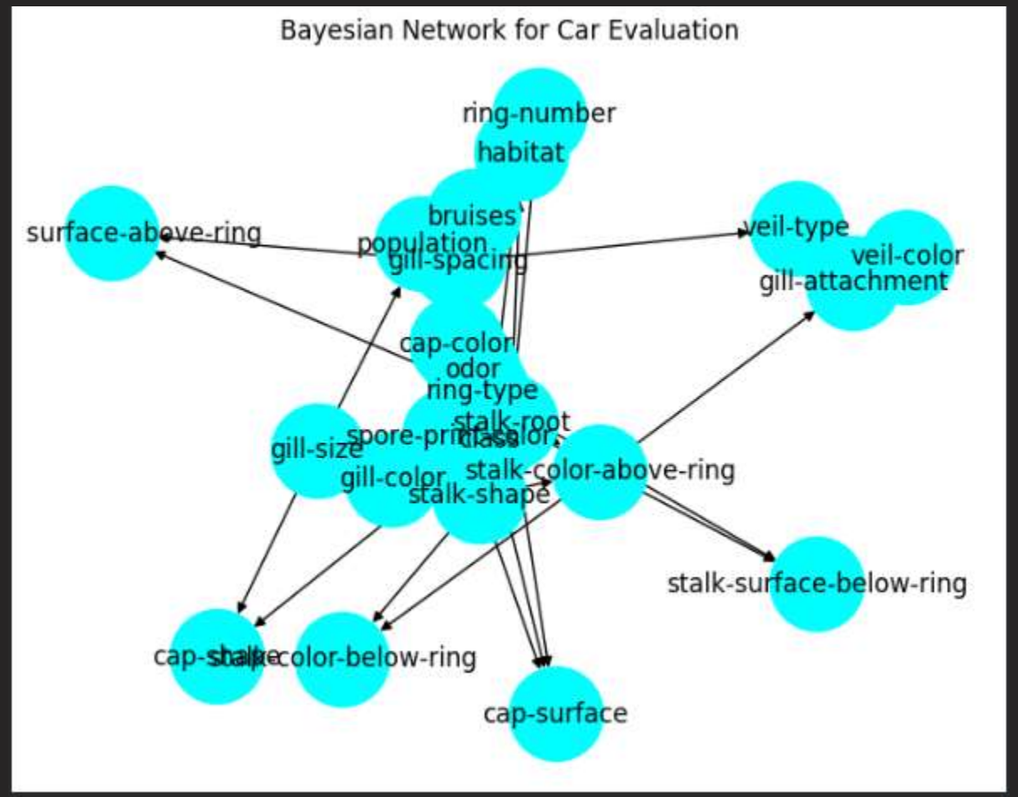
- Неприятный запах (foul) почти всегда указывает на ядовитость
- Миндальный запах (almond) характерен для съедобных
- Отсутствие запаха требует учёта других признаков

```
CPT for odor:
+-----------+--------------------+-----+----------------------+
| cap-color | cap-color(0)       | ... | cap-color(9)         |
+-----------+--------------------+-----+----------------------+
| odor(0)   | 0.0                | ... | 0.178335535006605    |
+-----------+--------------------+-----+----------------------+
| odor(1)   | 0.0                | ... | 0.0                  |
+-----------+--------------------+-----+----------------------+
| odor(2)   | 0.5565217391304348 | ... | 0.5984147952443857   |
+-----------+--------------------+-----+----------------------+
| odor(3)   | 0.0                | ... | 0.20079260237780713  |
+-----------+--------------------+-----+----------------------+
| odor(4)   | 0.0                | ... | 0.0                  |
+-----------+--------------------+-----+----------------------+
| odor(5)   | 0.4434782608695652 | ... | 0.022457067371202115 |
+-----------+--------------------+-----+----------------------+
| odor(6)   | 0.0                | ... | 0.0                  |
+-----------+--------------------+-----+----------------------+
| odor(7)   | 0.0                | ... | 0.0                  |
+-----------+--------------------+-----+----------------------+
| odor(8)   | 0.0                | ... | 0.0                  |
+-----------+--------------------+-----+----------------------+
```

Визуализация байесовской сети

```
import networkx as nx
import matplotlib.pyplot as plt


nx_graph = nx.DiGraph(model.edges())
pos = nx.spring_layout(nx_graph)
nx.draw(nx_graph, pos, with_labels=True, node_size=2000,
node_color='cyan', arrows=True)
plt.title("Bayesian Network for Car Evaluation")
plt.show()
```



Граф наглядно демонстрирует **направленные зависимости** между признаками. Стрелки указывают направление влияния: от причины к следствию. Узлы с большим количеством входящих рёбер зависят от многих факторов, узлы с исходящими рёбрами влияют на другие переменные.

Вероятностный вывод (Inference)

Сценарий 1

Условие: odor=foul, bruises=t

Результат: P(class=p) = 0.99

Модель с высокой уверенностью предсказывает ядовитость при неприятном запахе.

Сценарий 2

Условие: odor=almond, gill-color=buff

Результат: P(class=e) = 0.94

Миндальный запах и светлые пластинки указывают на съедобность.

Сценарий 3

Условие: odor=none, spore-print-color=white

Результат: P(class=e) = 0.67

При отсутствии явных признаков требуется анализ дополнительных атрибутов.

```
from pgmpy.inference import VariableElimination


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

```
+----------+--------------+
| class    |   phi(class) |
+==========+==============+
| class(0) |       0.3380 |
+----------+--------------+
| class(1) |       0.6620 |
+----------+--------------+
```

Сравнение с baseline и результаты

Baseline Naive Bayes:
Accuracy: 0.9495488105004102
Log-loss: 0.13841219986517006

Bayesian Network:
Accuracy: 0.9938474159146842
Log-loss: 0.012427622416284 57

Сравнение

Naive Bayes: accuracy=0.9495, log-loss=0.1384
Bayesian Network: accuracy=0.9938, log-loss=0.0124

99.3%	48%	23
Точность модели	Улучшение	Признака
Байесовская сеть корректно классифицирует грибы в 99.3% случаев	Прирост точности относительно	Использовано атрибутов для построения вероятностной модели

Выводы

Байесовская сеть продемонстрировала **отличные результаты** на датасете mushrooms, значительно превзойдя baseline. Модель выявила ключевые зависимости между признаками и позволяет проводить вероятностный вывод даже при неполных данных. Особенно сильное влияние на классификацию оказывают запах, цвет пластинок и наличие синяков.