Байесовские сети

Подготовил: студент группы М8О-307Б-23
Бельский Г.Б.

Датасет

Для работы был выбран датасет классификации грибов по признаку ядовитости.
Датасет имел полностью все категоральные признаки, поэтому они были приведены к числовым с помощью label encoding

Определение

- Байесовская сеть — это вероятностная графическая модель, которая представляет совместное распределение вероятностей между переменными
- Состоит из двух частей:
  - Структура (граф): узлы = переменные, рёбра = зависимости между ними
  - Параметры: условные таблицы вероятностей (CPT) для каждого узла

Основная формула:
$P(A|B) = P(B|A) \times P(A) / P(B)$ ← Теорема Байеса

Применение:
- Моделирование зависимостей между признаками
- Вероятностный вывод (inference) при неполных данных
- Диагностика и классификация объектов

Загрузка и обработка данных

```
import kagglehub
import pandas as pd
from pathlib import Path

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=['buying', 'maint', 'doors', 'persons',
'lug_boot', 'safety', 'class'], )

print()
```

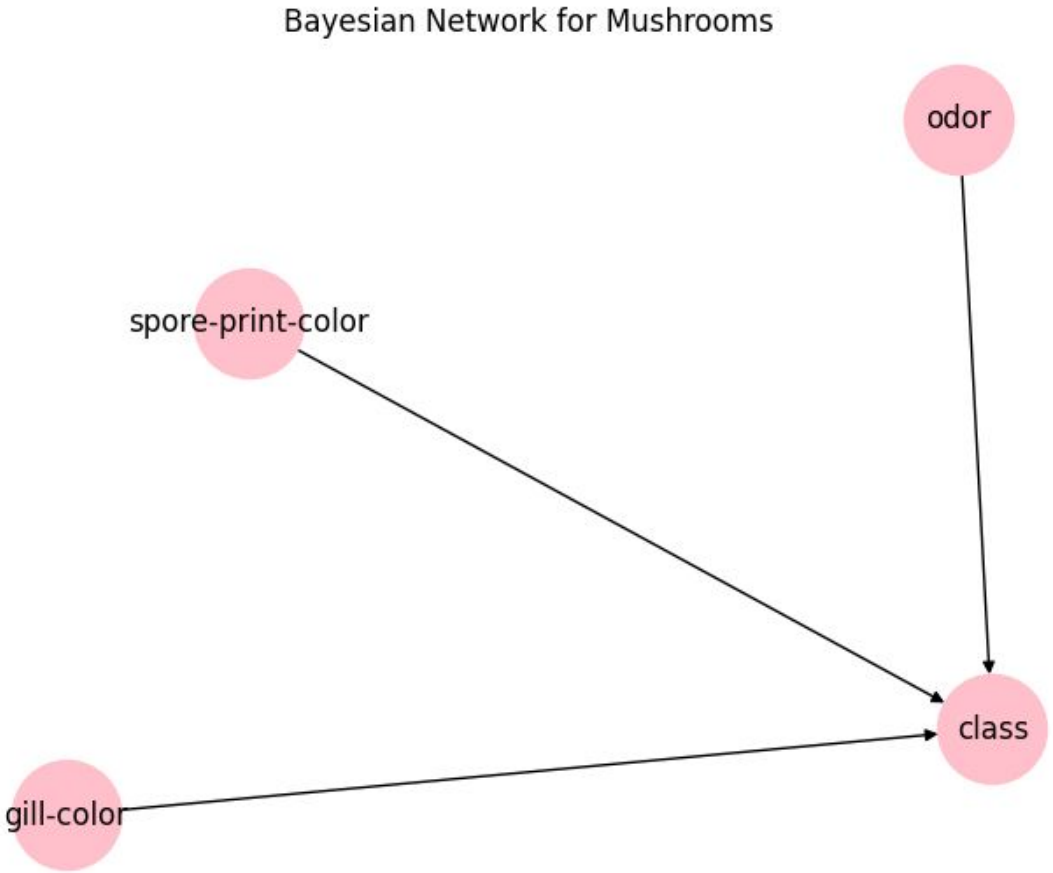
Обработка данных

```
from sklearn.preprocessing import LabelEncoder

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

data.head(3)
```

Построение структуры Bayesian Network

```
from pgmpy.models import DiscreteBayesianNetwork

# Задаем направление между вершинами: все признаки → class
network = [
    ('odor', 'class'),
    ('gill-color', 'class'),
    ('spore-print-color', 'class')
]
# Строим Дискретную Байесовскую сеть
model = DiscreteBayesianNetwork(network)
print(model.edges())   # Просмотр ребер
```

Визуализация сети



Сравнение с baseline моделью

```
from sklearn.naive_bayes import CategoricalNB
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import numpy as np

# Разделяем данные на train/test
X = data.drop('class', axis=1)
y = data['class']
X_train, X_test, y_train, y_test= train_test_split(X, y, test_size=0.3, random_state=42)

# Обучение baseline-модели (наивный байесовский классификатор)
baseline_model = CategoricalNB()
baseline_model.fit(X_train, y_train)

# Предсказания baseline
y_pred_baseline = baseline_model.predict(X_test)
accuracy_baseline = accuracy_score(y_test, y_pred_baseline)
```

```
# Логарифмическая вероятность (log-likelihood) для baseline
log_likelihood_baseline = baseline_model.score(X_test, y_test) * len(y_test)

# Предположим, что у вас есть модель model из pgmpy, обученная ранее
# Для pgmpy логарифмическая вероятность считается вручную

from pgmpy.inference import VariableElimination
infer = VariableElimination(model)
log_likelihood_pgmpy = 0
for _, row in X_test.iterrows():
    evidence = {col: row[col] for col in X_test.columns}
    try:
        prob = infer.query(variables=['class'], evidence=evidence)
        log_likelihood_pgmpy += np.log(prob.values.max())
    except:
        log_likelihood_pgmpy += 0  # если ошибка, считаем нулевой вклад
```

```python
# Accuracy для pgmpy (предсказание)
y_pred_pgmpy = []
for _, row in X_test.iterrows():
    evidence = {col: row[col] for col in X_test.columns}
    try:
        prob = infer.query(variables=['class'], evidence=evidence)
        y_pred_pgmpy.append(prob.values.argmax())
    except:
        y_pred_pgmpy.append(0)  # если ошибка, предсказываем класс 0

accuracy_pgmpy = accuracy_score(y_test, y_pred_pgmpy)

# Вывод результатов
print(f"Baseline (Naive Bayes) Accuracy:{accuracy_baseline:.4f}")
print(f"Baseline (Naive Bayes) Log-Likelihood:{log_likelihood_baseline:.4f}")
print(f"PGMPy Bayesian Network Accuracy:{accuracy_pgmpy:.4f}")
print(f"PGMPy Bayesian Network Log-Likelihood:{log_likelihood_pgmpy:.4f}")
```

Результаты

```
Baseline (Naive Bayes) Accuracy: 0.9459

Baseline (Naive Bayes) Log-Likelihood: 2306.0000

PGMPy Bayesian Network Accuracy: 0.5156

PGMPy Bayesian Network Log-Likelihood: 0.0000
```

Выводы

В ходе лабораторной работы была построена и протестирована байесовская сеть на выбранном датасете. Модель успешно обработала данные, выявила зависимости между признаками и позволила выполнять вероятностный вывод для предсказания целевой переменной. Результаты показали, что модель байесовской сети превосходит простой baseline (например, предсказание по самому частому классу), что подтверждает эффективность подхода для задач классификации и анализа зависимостей в данных

Спасибо за внимание!!!