TRƯỜNG ĐẠI HỌC HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC

NHẬP MÔN HỌC MÁY

NGÀNH: CÔNG NGHỆ THÔNG TIN

Đề tài:

TÌM HIỂU VÀ CÀI ĐẶT ỨNG DỤNG BÀI TOÁN NAIVE BAYES CLASSIFIER

SV: Trần Hữu Luân - 2274802010520
Lê Duy Khang – 2274802010374

GVHD: TRẦN NGỌC VIỆT

THÀNH PHỐ HỒ CHÍ MINH – NĂM 2025

LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin gửi lời cảm ơn chân thành đến thầy Trần Ngọc Việt. Trong quá trình học tập và tìm hiểu bộ môn Nhập môn học máy, chúng em đã nhận được sự quan tâm, giúp đỡ và hướng dẫn tận tình, tâm huyết của thầy. Thầy đã truyền đạt cho chúng em những kiến thức quý báu, giúp chúng em hiểu rõ hơn về các thuật toán và ứng dụng thực tế của học máy.

Nhóm thực hiện báo cáo

MỤC LỤC

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT – THUẬT TOÁN NAIVE BAYES CLASSIFIER

1.1. Lý do chọn đề tài

Trong thời đại công nghệ số hiện nay, trí tuệ nhân tạo và học máy đóng vai trò quan trọng trong nhiều lĩnh vực, từ y tế, tài chính, giáo dục cho đến nhận dạng hình ảnh và xử lý ngôn ngữ tự nhiên. Một trong những thuật toán cơ bản nhưng hiệu quả trong phân loại dữ liệu là **Naïve** Bayes Classifier. Trong đề tài này, nhóm chúng em lựa chọn thuật toán Naïve Bayes Classifier để phân loại động vật dựa trên các đặc điểm sinh học. Việc nghiên cứu và ứng dụng thuật toán này giúp nhóm hiểu rõ hơn về nguyên lý hoạt động của định lý Bayes, cách tiếp cận phân loại dựa trên xác suất, cũng như khả năng ứng dụng thực tế trong các lĩnh vực khác nhau.

1.2. Tổng quan về thuật toán Naive Bayes Classifier

A. Giới thiệu về Naive Bayes

Naïve Bayes là một thuật toán phân loại dựa trên định lý Bayes và giả định rằng các thuộc tính trong dữ liệu là độc lập với nhau. Thuật toán Naïve Bayes thường được sử dụng trong các bài toán phân loại như:

- Phân loại văn bản
- Nhận diện khuôn mặt và hình ảnh
- Dự đoán bệnh lý trong y khoa

Mặc dù giả định tính độc lập giữa các thuộc tính không phải lúc nào cũng đúng trong thực tế, nhưng Naïve Bayes vẫn hoạt động tốt trong nhiều tình huống nhờ tính đơn giản, tốc độ cao và hiệu quả ngay cả với tập dữ liệu nhỏ.

B. Định lý Navie Bayes

Định lý Bayes là nền tảng lý thuyết của thuật toán Naïve Bayes, giúp tính xác suất xảy ra của một sự kiện dựa trên thông tin có sẵn từ các sự kiện liên quan.

Công thức tổng quát của định lý Bayes:

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

Trong đó:

- P(A): Xác suất tiên nghiệm của A (trước khi biết B).
- P(B): Xác suất xảy ra của B.
- P(B|A): Xác suất xảy ra B khi biết A đã xảy ra.
- P(A|B): Xác suất hậu nghiệm của A khi biết B đã xảy ra.

C. Một số kiểu mô hình phổ phiến

❖ **Gaussian Naïve Bayes (GNB)**: Áp dụng cho dữ liệu có phân phối chuẩn (Gaussian)***, t***hường được sử dụng trong bài toán phân loại dữ liệu liên tục.

❖ **Multinomial Naïve Bayes (MNB)*:*** Áp dụng cho dữ liệu rời rạc, thường dùng trong xử lý ngôn ngữ tự nhiên (NLP), như phân loại văn bản. Phù hợp với dữ liệu đếm tần suất, chẳng hạn như số lần xuất hiện của từ trong văn bản.

❖ **Bernoulli Naïve Bayes (BNB)**: Áp dụng cho dữ liệu nhị phân (0 hoặc 1). Thường được dùng trong bài toán phân loại tài liệu có hoặc không có một đặc trưng nào đó

D. Ứng dụng thuật toán

❖ Nhận diện khuôn mặt và hình ảnh: Phân loại đối tượng trong hình ảnh (ví dụ: phân loại động vật, phương tiện giao thông...), nhận diện chữ viết tay.

- ❖ Chẩn đoán y khoa***:*** Phát hiện bệnh dựa trên các triệu chứng (ví dụ: chẩn đoán ung thư, tiểu đường...). Hỗ trợ ra quyết định cho bác sĩ bằng cách phân tích dữ liệu bệnh nhân.

- ❖ Phát hiện gian lận: Chuyển đổi giọng nói thành văn bản (Speech-to-Text). Ứng dụng trong trợ lý ảo như Siri, Google Assistant.

1.3 Kết luận

Naïve Bayes là một thuật toán phân loại đơn giản nhưng mạnh mẽ, dựa trên Định lý Bayes với giả định về tính độc lập giữa các đặc trưng. Nhờ vào tính toán nhanh, hiệu quả với tập dữ liệu nhỏ, dễ triển khai và khả năng hoạt động tốt ngay cả khi dữ liệu có nhiễu, Naïve Bayes được ứng dụng rộng rãi trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, phát hiện gian lận, phân tích y khoa, và nhận diện hình ảnh. Mặc dù có một số hạn chế như giả định độc lập không luôn đúng trong thực tế, nhưng với việc áp dụng đúng loại mô hình (Gaussian, Multinomial, Bernoulli, Complement, Categorical), Naïve Bayes vẫn là một trong những công cụ hữu ích để giải quyết các bài toán phân loại.

CHƯƠNG 2. ÁP DỤNG THUẬT TOÁN NAIVE BAYES TRONG PHÂN LOẠI CHIM

2.1 Phát biểu bài toán:

Trong thực tế, việc phân loại các loài chim dựa trên các đặc điểm sinh học là một bài toán quan trọng trong nghiên cứu động vật học và bảo tồn thiên nhiên. Với sự phát triển của khoa học dữ liệu, thuật toán Naïve Bayes đã được ứng dụng rộng rãi trong phân loại dữ liệu, đặc biệt là trong nhận dạng và phân loại động vật. Bài toán đặt ra là xây dựng một mô hình phân loại chim dựa trên thuật toán **Naïve Bayes Classifier**. Mô hình này sẽ sử dụng các đặc điểm đặc trưng của loài chim để dự đoán và xác định nhóm mà một loài chim thuộc về. Các đặc trưng có thể bao gồm số lượng chân, khả năng bay, có lông hay không, khả năng đẻ trứng, môi trường sống (nước, trên cạn), và các yếu tố sinh học khác.

|2.2 Minh họa bài toán.

Class B:

d = |V| = 7

⇨ $N_B = 12$

$19 = N_B + |V|$

- Test:

E6 = [2, 1, 1, 0, 1, 0, 1, 1]

$$P(B|x6) \propto P(B)\ \Pi^{d}_{i=1} P(x_i / B) = \frac{5}{12} * \frac{7}{32} * \left(\frac{6}{32}\right)^2 * \left(\frac{3}{32}\right)^1 * \left(\frac{4}{32}\right)^1 * \left(\frac{2}{32}\right)^1$$

$$=2{,}35.\ \mathbf{10^{-6}}$$

$$\mathbf{P(e6|N)} = 3.91 \times 10^{-6}$$

⇨ **P(e6|B) < P(e6|N) nên e6 thuộc lớp N**

Xác suất:

E6 sẽ thuộc lớp N. Với kết quả này, mô hình Naïve Bayes phân loại chim chính xác theo dữ liệu trong mã nguồn.

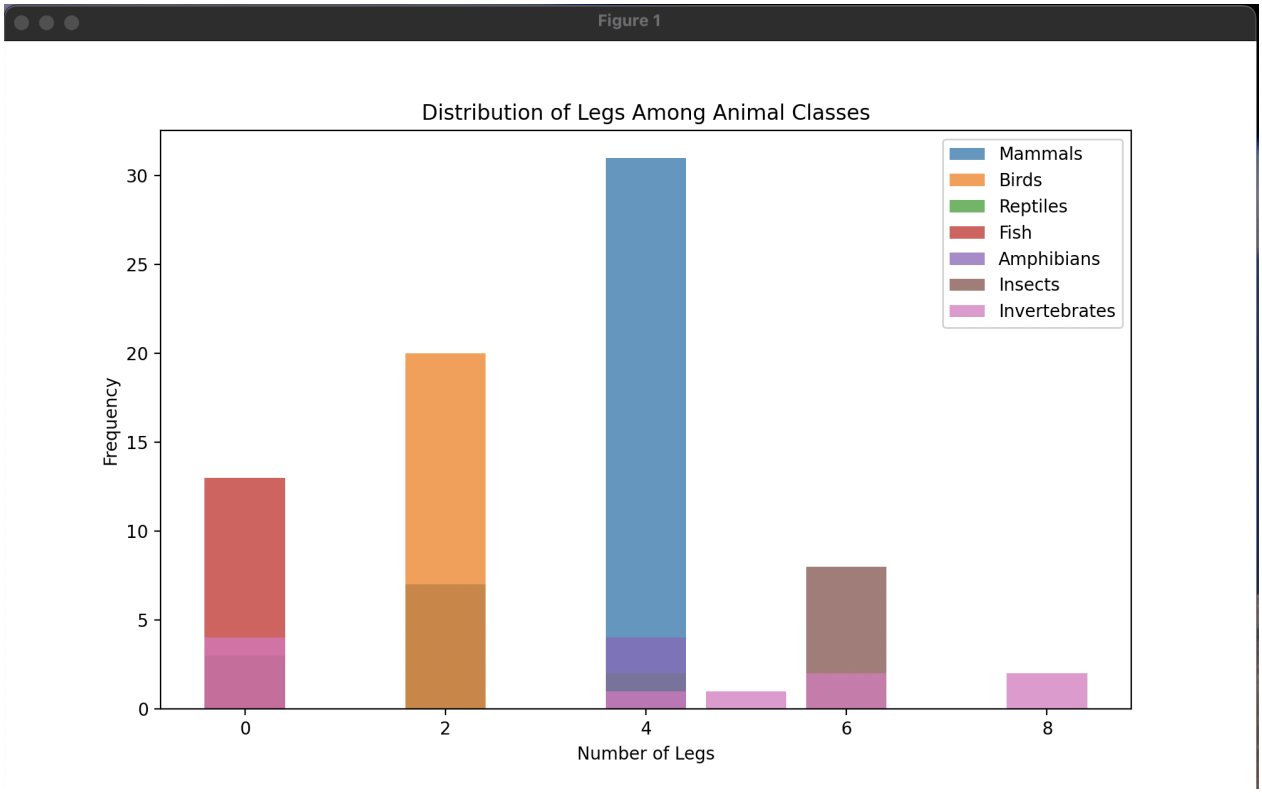
2.3 Mã nguồn

Phần này sẽ áp dụng thuật toán Naïve Bayes Classifier trên về phân loại chim trong những đoạn mã Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
from sklearn.model_selection import train_test_split

# Đọc dữ liệu từ zoo.csv
data = pd.read_csv("zoo.csv")

# Xác định các nhóm động vật
class_labels = {
    1: "Mammals", 2: "Birds", 3: "Reptiles", 4: "Fish", 5: "Amphibians", 6: "Insects", 7: "Invertebrates"
}
genders = sorted(data['class_type'].unique())  # Các loại động vật

# Chia dữ liệu theo nhóm động vật
animals = {gender: data[data['class_type'] == gender] for gender in genders}

# Các đặc trưng sử dụng để phân loại
selected_features = ['legs', 'hair', 'feathers', 'eggs', 'milk', 'airborne', 'aquatic', 'predator']

class Feature:
    def __init__(self, data, name=None, bin_width=None):
        self.name = name
        self.bin_width = bin_width
        self.freq_dict = dict(Counter(data))
        self.freq_sum = sum(self.freq_dict.values())
        self.unique_values = len(set(data))

    def frequency(self, value, smoothing=1):
        return (self.freq_dict.get(value, 0) + smoothing) / (self.freq_sum + smoothing * self.unique_values)

# Lựa chọn thuộc tính để phân loại
fts = {gender: {feature: Feature(animals[gender][feature]) for feature in selected_features} for gender in genders}

# Vẽ biểu đồ phân bố số chân của từng nhóm động vật
plt.figure(figsize=(10, 6))
for gender in genders:
    frequencies = sorted(fts[gender]['legs'].freq_dict.items(), key=lambda x: x[0])
    if frequencies:
        X, Y = zip(*frequencies)
        plt.bar(X, Y, label=f'{class_labels[gender]}', alpha=0.75)
```

```
plt.legend(loc='upper right')
plt.title("Distribution of Legs Among Animal Classes")
plt.xlabel("Number of Legs")
plt.ylabel("Frequency")
plt.show()

class NBclass:
    def __init__(self, name, features):
        self.features = features
        self.name = name

    def log_probability(self, data_point):
        log_prob = 0
        for feature_name in selected_features:
            log_prob += np.log(self.features[feature_name].frequency(data_point[feature_name]))
        return log_prob

cls = {gender: NBclass(class_labels[gender], fts[gender]) for gender in genders}

class Classifier:
    def __init__(self, nbclasses):
        self.nbclasses = nbclasses

    def predict(self, data_point, show_all=True):
        log_probs = [(nbclass.log_probability(data_point), nbclass.name) for nbclass in self.nbclasses]
        log_probs.sort(reverse=True, key=lambda x: x[0])  # Sắp xếp theo xác suất giảm dần

        # Chuyển đổi log-prob thành xác suất thực tế
        exp_probs = np.exp([p[0] for p in log_probs])
        prob_sum = np.sum(exp_probs)
        probabilities = [(exp_probs[i] / prob_sum, log_probs[i][1]) for i in range(len(log_probs))]

        return probabilities if show_all else max(probabilities)
# Chia dữ liệu thành tập huấn luyện và kiểm tra
# Tạo feature vector và target label
X = data[selected_features]
y = data['class_type']

# Chia dữ liệu thành 80% huấn luyện và 20% kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Huấn luyện Naive Bayes Classifier
train_fts = {gender: {feature: Feature(animals[gender][feature]) for feature in selected_features} for gender in genders}
train_cls = {gender: NBclass(class_labels[gender], train_fts[gender]) for gender in genders}
classifier = Classifier([train_cls[g] for g in genders])
```

```python
# Đánh giá trên tập kiểm tra
correct_predictions = 0
for i in range(len(X_test)):
    test_data = X_test.iloc[i].to_dict()  # Chuyển dữ liệu test thành dictionary
    actual_class = y_test.iloc[i]
    predicted_probabilities = classifier.predict(test_data, show_all=False)
    predicted_class = predicted_probabilities[1]  # Lấy class có xác suất cao nhất

    # Kiểm tra kết quả
    if predicted_class == class_labels[actual_class]:
        correct_predictions += 1

accuracy = correct_predictions / len(X_test)
print(f'Accuracy on test set: {accuracy * 100:.2f}%')

# Kiểm tra với một mẫu dữ liệu
sample_data = {'legs': 4, 'hair': 1, 'feathers': 0, 'eggs': 0, 'milk': 1, 'airborne': 0, 'aquatic': 0, 'predator': 1}
print(f'Predicted probabilities for sample: {classifier.predict(sample_data, show_all=True)}')
```



Accuracy on test set: 85.71%

Predicted probabilities for sample:

[(np.float64(0.9893162971097392), 'Mammals'),
(np.float64(0.0063891423078448 42), 'Reptiles'),
(np.float64(0.003927124654370679), 'Amphibians'),
(np.float64(0.00021129932067679677), 'Invertebrates'),
(np.float64(0.0001515094388260294), 'Insects'),
(np.float64(4.563673077032027e-06), 'Fish'),
(np.float64(6.349546535648552e-08), 'Birds')]

2.4 Kết luận

Mô hình Naive Bayes là mô hình phân lớp đơn giản dễ cài đặt, có tốc độ xử lý nhanh. Tuy nhiên có hạn chế là giả định các đặc trưng độc lập với nhau, điều này không phải lúc nào cũng đúng trong thực tế. Nếu các thuộc tính có mối quan hệ mạnh với nhau, mô hình có thể không đạt hiệu suất tốt nhất.

Ngoài ra, với dữ liệu có số lượng mẫu nhỏ hoặc không cân bằng giữa các lớp, Naive Bayes có thể gặp khó khăn trong việc đưa ra dự đoán chính xác. Tuy nhiên, với các bài toán có đặc điểm phù hợp, như phân loại văn bản, nhận diện email spam, hoặc phân loại động vật như trong bài toán này, Naive Bayes vẫn là một lựa chọn hiệu quả.

CHƯƠNG 3. KẾT LUẬN

Mô hình Naive Bayes đã được áp dụng để phân loại các loài động vật dựa trên các đặc trưng sinh học. triển khai mô hình Naive Bayes, chúng em nhận thấy rằng đây là một thuật toán phân loại dựa trên xác suất với hiệu suất tính toán cao và khả năng áp dụng linh hoạt trong nhiều lĩnh vực. Nhờ vào cấu trúc đơn giản, mô hình có thể xử lý dữ liệu nhanh chóng, phù hợp với những bài toán có số lượng lớn mẫu nhưng cần ra quyết định tức thời. Tuy nhiên, một trong những nhược điểm quan trọng của Naive Bayes là giả định rằng các đặc trưng trong dữ liệu không có sự phụ thuộc lẫn nhau. Trên thực tế, điều này không phải lúc nào cũng đúng, đặc biệt là với những tập dữ liệu có mối quan hệ phức tạp giữa các thuộc tính. Khi giả định này bị vi phạm, độ chính xác của mô hình có thể bị ảnh hưởng đáng kể. Một hạn chế khác của Naive Bayes là nó phụ thuộc nhiều vào chất lượng dữ liệu đầu vào. Nếu dữ liệu không đồng đều, có sự mất cân bằng giữa các lớp, hoặc xuất hiện nhiều giá trị chưa từng có trong tập huấn luyện, mô hình có thể đưa ra những dự đoán thiếu chính xác. Ngoài ra, việc xử lý các đặc trưng liên tục trong mô hình này cũng yêu cầu những phương pháp tiếp cận khác nhau, chẳng hạn như sử dụng Gaussian Naive Bayes để mô hình hóa dữ liệu theo phân phối chuẩn. Dù còn tồn tại những điểm yếu, nhưng với các bài toán có đặc điểm phù hợp như phân loại văn bản, phát hiện thư rác, nhận diện cảm xúc hay phân loại động vật, Naive Bayes vẫn chứng tỏ được tính hiệu quả và khả năng ứng dụng rộng rãi. Mô hình này đặc biệt hữu ích trong những trường hợp cần đưa ra dự đoán nhanh mà không đòi hỏi quá nhiều tài nguyên tính toán.

TÀI LIỆU THAM KHẢO

1. Naive Bayes Classifier Tutorial using Scikit-learn https://www.datacamp.com/tutorial/naive-bayes-scikit-learn Ngày 3/3/2023.

2. Naive Bayes Classifiers: https://www.geeksforgeeks.org/naive-bayes-classifiers/ Ngày 13/1/2025

3. Naive Bayes Classifier Explained with practical problems: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ Ngày 1/1/2025