Câu 1: Phân phối Bernoulli và Multinomial

Cho tập dữ liệu Education.csv [https://drive.google.com/file/d/1Gn6YWHXRuPbTUXY5HFxM5C_tJHuZxCka/view?usp=sharing]

- Trong đó:
    - Text: Chứa đoạn văn bản liên quan đến chủ đề giáo dục.
    - Label: Chứa nhãn cảm xúc của văn bản [Tích cực (Positive)/Tiêu cực (Negative)].
- Yêu cầu: Áp dụng thuật toán Naive Bayes (phân phối bernoulli và phân phối Multinomial) để dự đoán cảm xúc của văn bản là tích cực hay tiêu cực và so sánh kết quả của hai phân phối đó.

Câu 2: Phân phối Gaussian

Cho tập dữ liệu Drug.csv [https://drive.google.com/file/d/1_G8oXkLlsauQkujZzJZJwibAWu5PgBXK/view?usp=sharing]

- Trong đó:
    - Age: Tuổi của bệnh nhân
    - Sex: Giới tính của bệnh nhân
    - BP: Mức huyết áp
    - Cholesterol: Mức cholesterol trong máu
    - Na_to_K: Tỷ lệ Natri và Kali trong máu
    - Drug: Loại thuốc [A/B/C/X/Y]
- Yêu cầu: Áp dụng thuật toán Naive Bayes (phân phối Gaussian) để dự đoán kết quả loại thuốc phù hợp với bệnh nhân.

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import BernoulliNB, MultinomialNB, GaussianNB
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder

# Đọc dữ liệu Education.csv và Drug.csv từ đường dẫn
education_df = pd.read_csv('D:/Tai luu (thanh)/baitapthu/lab2/Education.csv')
drug_df = pd.read_csv('D:/Tai luu (thanh)/baitapthu/lab2/drug200.csv')
```

```
# Chuẩn bị dữ liệu
X = education_df['Text']  # Đoạn văn bản
y = education_df['Label']  # Nhãn cảm xúc
```

Double-click (or enter) to edit

```
# Chuyển đổi văn bản thành các đặc trưng dạng số (Bag of Words)
vectorizer = CountVectorizer(binary=True)  # Sử dụng binary cho Bernoulli
X_vectorized = vectorizer.fit_transform(X)

# Chia dữ liệu thành tập huấn luyện và kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.3, random_state=42)

# Áp dụng Bernoulli Naive Bayes
bernoulli_nb = BernoulliNB()
bernoulli_nb.fit(X_train, y_train)
y_pred_bernoulli = bernoulli_nb.predict(X_test)
```

```
# Đánh giá kết quả Bernoulli Naive Bayes
print("Kết quả Bernoulli Naive Bayes")
print("Độ chính xác:", accuracy_score(y_test, y_pred_bernoulli))
print("Báo cáo chi tiết:\n", classification_report(y_test, y_pred_bernoulli, zero_division=1))
```

```
Kết quả Bernoulli Naive Bayes
Độ chính xác: 0.4375
Báo cáo chi tiết:
               precision    recall  f1-score   support
```

```
    negative       0.36      0.67      0.47         6
    positive       0.60      0.30      0.40        10

    accuracy                           0.44        16
   macro avg       0.48      0.48      0.44        16
weighted avg       0.51      0.44      0.43        16
```

```
# Áp dụng Multinomial Naive Bayes
multinomial_nb = MultinomialNB()
multinomial_nb.fit(X_train, y_train)
y_pred_multinomial = multinomial_nb.predict(X_test)


# Đánh giá kết quả Multinomial Naive Bayes
print("\nKết quả Multinomial Naive Bayes")
print("Độ chính xác:", accuracy_score(y_test, y_pred_multinomial))
print("Báo cáo chi tiết:\n", classification_report(y_test, y_pred_multinomial, zero_division=1))
```

```
Kết quả Multinomial Naive Bayes
Độ chính xác: 0.625
Báo cáo chi tiết:
               precision    recall  f1-score   support

    negative       0.50      0.50      0.50         6
    positive       0.70      0.70      0.70        10

    accuracy                           0.62        16
   macro avg       0.60      0.60      0.60        16
weighted avg       0.62      0.62      0.62        16
```

```
# Chuẩn bị dữ liệu
X_drug = drug_df[['Age', 'Na_to_K']]  # Sử dụng các đặc trưng liên tục (Age và Na_to_K)
y_drug = drug_df['Drug']


# Mã hóa nhãn (Drug) thành số
label_encoder = LabelEncoder()
y_drug_encoded = label_encoder.fit_transform(y_drug)

# Chia dữ liệu thành tập huấn luyện và kiểm tra
X_train_drug, X_test_drug, y_train_drug, y_test_drug = train_test_split(X_drug, y_drug_encoded, test_size=0.3, random_state=42)

# Áp dụng Gaussian Naive Bayes
gaussian_nb = GaussianNB()
gaussian_nb.fit(X_train_drug, y_train_drug)
y_pred_gaussian = gaussian_nb.predict(X_test_drug)


# Đánh giá kết quả Gaussian Naive Bayes cho Drug.csv
print("\nKết quả Gaussian Naive Bayes cho Drug.csv")
print("Độ chính xác:", accuracy_score(y_test_drug, y_pred_gaussian))
print("Báo cáo chi tiết:\n", classification_report(y_test_drug, y_pred_gaussian, target_names=label_encoder.classes_, zero_division=1))
```

```
Kết quả Gaussian Naive Bayes cho Drug.csv
Độ chính xác: 0.7166666666666667
Báo cáo chi tiết:
               precision    recall  f1-score   support

       DrugY       1.00      1.00      1.00        26
       drugA       0.33      0.29      0.31         7
       drugB       0.40      0.67      0.50         3
       drugC       0.00      0.00      0.00         6
       drugX       0.57      0.72      0.63        18

    accuracy                           0.72        60
   macro avg       0.46      0.53      0.49        60
weighted avg       0.66      0.72      0.68        60

c:\Users\nhoxt\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\nhoxt\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\nhoxt\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\metrics\_classification.py:1531: UndefinedMetricWarning
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```