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.

In [115…
```
import numpy as np
from sklearn.naive_bayes import BernoulliNB, MultinomialNB, GaussianNB
import pandas as pd
```

In [116…
```
df = pd.read_csv("~/Documents/ML_VLU/lab2/Data/Education.csv")
```

In [117…
```
df.head(5)
```

Out[117…

| | Text | Label |
|---|---|---|
| 0 | The impact of educational reforms remains unce... | positive |
| 1 | Critics argue that recent improvements in the ... | negative |
| 2 | Innovative teaching methods have led to unexpe... | positive |
| 3 | Despite budget constraints, the school has man... | positive |
| 4 | The true effectiveness of online learning plat... | negative |

In [118…
```
text, label = df['Text'], df['Label']
```

In [119…
```
label.head(5)
```

Out[119…
```
0    positive
1    negative
2    positive
3    positive
4    negative
Name: Label, dtype: object
```

In [159…
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(text, label, test_siz
```

In [160…
```
# Convert data into numerical features
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(stop_words='english')
X_train_vect = vectorizer.fit_transform(X_train)
X_test_vect = vectorizer.transform(X_test)
```

In [161…
```
X_train_vect = X_train_vect.toarray()
X_test_vect = X_test_vect.toarray()
```

In [162…
```
Bernoulli, Multinomial = BernoulliNB(), MultinomialNB()
Bernoulli.fit(X_train_vect, y_train)
Multinomial.fit(X_train_vect, y_train)
```

Out[162…
```
▾  MultinomialNB i ?

MultinomialNB()
```

In [163…
```
from sklearn.metrics import accuracy_score, classification_report
```

In [164…
```
print(accuracy_score(Multinomial.predict(X_test_vect),y_test))
print(accuracy_score(Bernoulli.predict(X_test_vect),y_test))
```

```
0.6666666666666666
0.5
```

In [165…
```
Multinomial_rp = classification_report(y_test, Multinomial.predict(X_test
Bernoulli_rp = classification_report(y_test, Bernoulli.predict(X_test_vec
```

In [166…
```
print(Bernoulli_rp)
print(Multinomial_rp)
```

```
              precision    recall  f1-score   support

        Posi       0.40      1.00      0.57         2
        Nega       1.00      0.25      0.40         4

    accuracy                           0.50         6
   macro avg       0.70      0.62      0.49         6
weighted avg       0.80      0.50      0.46         6

              precision    recall  f1-score   support

        Posi       0.50      1.00      0.67         2
        Nega       1.00      0.50      0.67         4

    accuracy                           0.67         6
   macro avg       0.75      0.75      0.67         6
weighted avg       0.83      0.67      0.67         6
```

In [168…
```
!streamlit run app.py
```

```
  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://10.7.167.86:8501

^C
  Stopping...
```

In [ ]:
```
```

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.

In [2]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confus
```

In [49]:
```
df = pd.read_csv("../Data/drug200.csv")
df
```

Out[49]:

| | Age | Sex | BP | Cholesterol | Na_to_K | Drug |
|---|---|---|---|---|---|---|
| **0** | 23 | F | HIGH | HIGH | 25.355 | DrugY |
| **1** | 47 | M | LOW | HIGH | 13.093 | drugC |
| **2** | 47 | M | LOW | HIGH | 10.114 | drugC |
| **3** | 28 | F | NORMAL | HIGH | 7.798 | drugX |
| **4** | 61 | F | LOW | HIGH | 18.043 | DrugY |
| **...** | ... | ... | ... | ... | ... | ... |
| **195** | 56 | F | LOW | HIGH | 11.567 | drugC |
| **196** | 16 | M | LOW | HIGH | 12.006 | drugC |
| **197** | 52 | M | NORMAL | HIGH | 9.894 | drugX |
| **198** | 23 | M | NORMAL | NORMAL | 14.020 | drugX |
| **199** | 40 | F | LOW | NORMAL | 11.349 | drugX |

200 rows × 6 columns

In [50]:
```
df_dummy = pd.get_dummies(df[['Age','Sex','BP','Cholesterol','Na_to_K']],
df_dummy.replace({False: 0, True: 1}, inplace=True)
df_dummy.head()
```

```
/tmp/ipykernel_71216/3981645735.py:2: FutureWarning: Downcasting behavior
in `replace` is deprecated and will be removed in a future version. To ret
ain the old behavior, explicitly call `result.infer_objects(copy=False)`.
To opt-in to the future behavior, set `pd.set_option('future.no_silent_dow
ncasting', True)`
  df_dummy.replace({False: 0, True: 1}, inplace=True)
```

Out[50]:

| | Age | Na_to_K | Sex_M | BP_LOW | BP_NORMAL | Cholesterol_NORMAL |
|---|---|---|---|---|---|---|
| **0** | 23 | 25.355 | 0 | 0 | 0 | 0 |
| **1** | 47 | 13.093 | 1 | 1 | 0 | 0 |
| **2** | 47 | 10.114 | 1 | 1 | 0 | 0 |
| **3** | 28 | 7.798 | 0 | 0 | 1 | 0 |
| **4** | 61 | 18.043 | 0 | 1 | 0 | 0 |

In [51]:
```
y = df['Drug']
X = df_dummy
```

In [35]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.3,rand
```

In [36]:
```
model = GaussianNB().fit(X_train, y_train)
```

In [60]:
```
accuracy_score(y_test, model.predict(X_test))
```

Out[60]: 0.8333333333333334