A4:Implement a Naive Bayes classifier using sci-kit-learn to classify text documents based on their content.

In [14]:
```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
```

In [40]:
```
# Small toy dataset
docs = [
    "I love space science and astronomy",
    "The galaxy is full of stars",
    "Computer graphics are amazing",
    "Rendering 3D models is fun",
    "Politics create conflicts between nations",
    "The government passed a new law",
    "Sun is a Star"
]
```

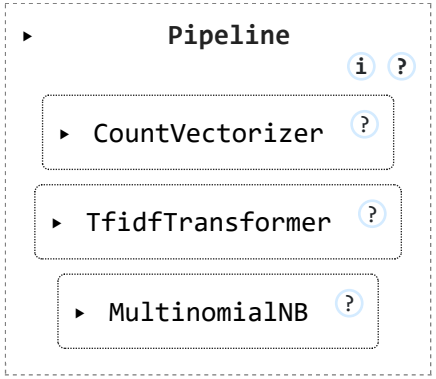
In [41]:
```
labels = [
    "sci", "sci",
    "comp", "comp",
    "politics", "politics", "sci"
]
```

In [42]:
```
# 2. Build a pipeline for vectorization + TF-IDF + Naive Bayes
model = Pipeline([
    ('vect', CountVectorizer()),          # Convert text to token counts
    ('tfidf', TfidfTransformer()),        # Convert counts to TF-IDF
    ('clf', MultinomialNB()),             # Naive Bayes classifier
])
```

In [43]:
```
# Split into train/test sets
X_train, X_test, y_train, y_test = train_test_split(docs, labels, test_size=0.33, random_state=42, stratify=labels)
```

In [44]:
```
# Train
model.fit(X_train, y_train)
```

Out[44]:



In [45]:
```
# Predict
y_pred = model.predict(X_test)
```

In [46]:
```
# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.3333333333333333

Classification Report:

              precision    recall  f1-score   support

        comp       0.00      0.00      0.00         1
    politics       0.00      0.00      0.00         1
         sci       0.33      1.00      0.50         1

    accuracy                           0.33         3
   macro avg       0.11      0.33      0.17         3
weighted avg       0.11      0.33      0.17         3
```

```
c:\Users\Venkat\AppData\Local\Continuum\anaconda3\envs\myenv_upgrade\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1706: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
c:\Users\Venkat\AppData\Local\Continuum\anaconda3\envs\myenv_upgrade\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1706: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
c:\Users\Venkat\AppData\Local\Continuum\anaconda3\envs\myenv_upgrade\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1706: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
```

In [ ]: