Бибиков Павел ИУ5-22М. Рубежный контроль №2

В этой тетрадке решается задача бинарной классификации текстов (lyrics) из датасета `tcc_ceds_music.csv`.
В качестве целевого признака возьмём наличие «violence» (строка `violence` > 0).
Два подхода к векторизации текстов:

- CountVectorizer
- TfidfVectorizer

Два классификатора:

1. RandomForestClassifier
2. LogisticRegression

Что такое признак violence в датасете?

В датасете tcc_ceds_music.csv каждая строка соответствует одной песне, в том числе её тексту (столбец lyrics) и различным метаданным: жанру, дате выпуска, акустическим характеристикам и т. д.

Среди этих метаданных есть тематические лексические счётчики: как часто в тексте употребляются слова, связанные с насилием (violence), любовью, дружбой, миром/жизнью и пр.

Почему именно этот признак?

Демонстрация бинарной классификации. Чётко разделить тексты на «содержат тему насилия» и «не содержат» проще всего:

- violence > 0 → класс 1 (есть насилие)
- violence = 0 → класс 0 (нет насилия)

Такой целевой признак легко интерпретировать и быстро показать работу алгоритмов RandomForestClassifier и LogisticRegression.

Однако в датасете нет violence = 0, так что возьмем медианное значение = 0.00250626582559355

In [19]:
```
# Ячейка 2: Импорт необходимых библиотек
import pandas as pd
import numpy as np
```

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

import warnings
warnings.filterwarnings('ignore')
```

In [20]:
```
# Ячейка 3: Загрузка и первичный осмотр данных
df = pd.read_csv('tcc_ceds_music.csv')
print("Форма данных:", df.shape)
print("Колонки:", df.columns.tolist())
print(df[['lyrics','violence']].head())
```

```
Форма данных: (28372, 31)
Колонки: ['Unnamed: 0', 'artist_name', 'track_name', 'release_date', 'genr
e', 'lyrics', 'len', 'dating', 'violence', 'world/life', 'night/time', 'shak
e the audience', 'family/gospel', 'romantic', 'communication', 'obscene', 'm
usic', 'movement/places', 'light/visual perceptions', 'family/spiritual', 'l
ike/girls', 'sadness', 'feelings', 'danceability', 'loudness', 'acousticnes
s', 'instrumentalness', 'valence', 'energy', 'topic', 'age']
                                              lyrics  violence
0  hold time feel break feel untrue convince spea...  0.063746
1  believe drop rain fall grow believe darkest ni...  0.096777
2  sweetheart send letter goodbye secret feel bet...  0.002770
3  kiss lips want stroll charm mambo chacha merin...  0.001548
4  till darling till matter know till dream live ...  0.001350
```

In [21]:
```
# Ячейка 4: Предобработка
# Удаляем строки без текста
df = df.dropna(subset=['lyrics'])

# Бинаризуем violence по порогу = медиане
threshold = df['violence'].median()
df['violence_flag'] = (df['violence'] > threshold).astype(int)

# Диагностика
print("Медиана violence:", threshold)
print("Распределение violence_flag:\n", df['violence_flag'].value_counts())
```

```
Медиана violence: 0.00250626582559355
Распределение violence_flag:
 violence_flag
1    14186
0    14186
Name: count, dtype: int64
```

In [22]:
```
# Ячейка 5: Формирование X и y и разбиение
X = df['lyrics']
y = df['violence_flag']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

In [23]:
```python
# Ячейка 6 (исправленная): CountVectorizer + RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import numpy as np

# Векторизация
cv = CountVectorizer(max_df=0.9, min_df=5, ngram_range=(1,2))
X_train_cv = cv.fit_transform(X_train)
X_test_cv  = cv.transform(X_test)

# Обучение
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train_cv, y_train)

# Предсказание
y_pred_rf = rf.predict(X_test_cv)

# Диагностика — убедимся, что есть оба класса
print("Уникальные метки в y_test:", np.unique(y_test))
print("Уникальные метки в y_pred_rf:", np.unique(y_pred_rf))

# Оценка
print("RandomForest + CountVectorizer:")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print(classification_report(
    y_test,
    y_pred_rf,
    labels=[0, 1],
    target_names=['No Violence', 'Violence']
))
```

```
Уникальные метки в y_test: [0 1]
Уникальные метки в y_pred_rf: [0 1]
RandomForest + CountVectorizer:
Accuracy: 0.8438766519823788
              precision    recall  f1-score   support

 No Violence       0.84      0.85      0.84      2838
    Violence       0.85      0.84      0.84      2837

    accuracy                           0.84      5675
   macro avg       0.84      0.84      0.84      5675
weighted avg       0.84      0.84      0.84      5675
```

In [24]:
```python
# Ячейка 7: TfidfVectorizer + LogisticRegression
# Векторизация
tfv = TfidfVectorizer(max_df=0.9, min_df=5, ngram_range=(1,2))
X_train_tfv = tfv.fit_transform(X_train)
X_test_tfv  = tfv.transform(X_test)

# Обучение
lr = LogisticRegression(max_iter=1000, random_state=42)
lr.fit(X_train_tfv, y_train)

# Предсказание и оценка
y_pred_lr = lr.predict(X_test_tfv)
```

```
print("LogisticRegression + TfidfVectorizer:")
print("Accuracy:", accuracy_score(y_test, y_pred_lr))
print(classification_report(y_test, y_pred_lr, target_names=['No Violence','
```

```
LogisticRegression + TfidfVectorizer:
Accuracy: 0.8176211453744493
              precision    recall  f1-score   support

 No Violence       0.78      0.88      0.83      2838
    Violence       0.86      0.76      0.81      2837

    accuracy                           0.82      5675
   macro avg       0.82      0.82      0.82      5675
weighted avg       0.82      0.82      0.82      5675
```

Ячейка 8: Выводы

- **RandomForestClassifier + CountVectorizer** показал точность *~84%*.
- **LogisticRegression + TfidfVectorizer** показал точность *~82%*.