```
# YouTube Sentiment Analysis Project
# Import Libraries
import pandas as pd
import numpy as np
import re
import string
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Step 1: Load Dataset
df = pd.read_csv("youtube_comments.csv")  # change to your dataset path
print(df.head())



# Step 2: Data Preprocessing
def clean_text(text):
    text = text.lower()
    text = re.sub(r'http\S+', '', text)  # remove URLs
    text = re.sub(r'@\w+', '', text)     # remove mentions
    text = re.sub(r'#\w+', '', text)     # remove hashtags
    text = re.sub(r'[^\w\s]', '', text)  # remove punctuation
    text = re.sub(r'\d+', '', text)      # remove numbers
    return text

df['cleaned'] = df['comment'].apply(clean_text)

# Step 3: Split Data
X = df['cleaned']
y = df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Feature Extraction (TF-IDF)
tfidf = TfidfVectorizer(max_features=5000)
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

```
X_test_tfidf = tfidf.transform(X_test)

# Step 5: Train Model
model = MultinomialNB()
model.fit(X_train_tfidf, y_train)

# Step 6: Predictions
y_pred = model.predict(X_test_tfidf)

# Step 7: Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Step 8: Try New Predictions
sample_comments = [
    "I love this video! It’s awesome ",
    "This is terrible, I dislike it.",
    "It's okay, not too good or bad."
]

sample_cleaned = [clean_text(x) for x in sample_comments]
sample_tfidf = tfidf.transform(sample_cleaned)
preds = model.predict(sample_tfidf)

for c, p in zip(sample_comments, preds):
    print(f"Comment: {c} --> Sentiment: {p}")
```

```
                                        comment     label
0  This video is amazing! Loved every part of it.  positive
1                I really hate the sound quality.  negative
2              The editing was great and smooth.  positive
3                  Not bad, but could be better.   neutral
4                Worst video I’ve seen this week.  negative
Accuracy: 0.0

Confusion Matrix:
 [[0 0 1]
 [1 0 0]
 [2 0 0]]
```

```
Classification Report:
               precision    recall  f1-score   support

    negative       0.00      0.00      0.00       1.0
     neutral       0.00      0.00      0.00       1.0
    positive       0.00      0.00      0.00       2.0

    accuracy                           0.00       4.0
   macro avg       0.00      0.00      0.00       4.0
weighted avg       0.00      0.00      0.00       4.0

Comment: I love this video! It’s awesome  --> Sentiment: negative
Comment: This is terrible, I dislike it. --> Sentiment: negative
Comment: It's okay, not too good or bad. --> Sentiment: neutral
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defi
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defi
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defi
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```