YouTube Comments Using CRISP-DM (Sentiment Analysis)

TEAM MEMBERS:

VEERAPU RAJU----------------------BU22CSEN0100246

CHINTHA KRISHNA BALAJI-------BU22CSEN0101063

BOMMANA HARSHITHA-----------BU22CSEN0101289

TALARI LOKESH KUMAR----------BU22CSEN0101360

C. NAYUM AKTHAR -----------------BU22CSEN0101361

D.NIKHIL REDDY---------------------BU22CSEN0101541

Download Dataset

```
import pandas as pd
from textblob import TextBlob

# Load the dataset
file_path = '/content/bts_2021_1.csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
print(data.head())
```

```
  query                                          url  \
0   bts  https://www.youtube.com/watch?v=S8GpX3SAeig
1   bts  https://www.youtube.com/watch?v=S8GpX3SAeig
2   bts  https://www.youtube.com/watch?v=S8GpX3SAeig
3   bts  https://www.youtube.com/watch?v=S8GpX3SAeig
4   bts  https://www.youtube.com/watch?v=S8GpX3SAeig

                                             title           upload_date  \
0  5 Hour BTS Piano Playlist | Study & Relax with...  2021-01-01T10:58:00Z
1  5 Hour BTS Piano Playlist | Study & Relax with...  2021-01-01T10:58:00Z
2  5 Hour BTS Piano Playlist | Study & Relax with...  2021-01-01T10:58:00Z
3  5 Hour BTS Piano Playlist | Study & Relax with...  2021-01-01T10:58:00Z
4  5 Hour BTS Piano Playlist | Study & Relax with...  2021-01-01T10:58:00Z

    channel    views   likes  dislikes  comment_count  \
0  DooPiano  2444982  119269       501           3224
1  DooPiano  2444982  119269       501           3224
2  DooPiano  2444982  119269       501           3224
3  DooPiano  2444982  119269       501           3224
4  DooPiano  2444982  119269       501           3224

                                        comment_text          comment_author  \
0  ♪ Listen on Spotify!: https://spoti.fi/3gC9GfA...                DooPiano
1  My ears: *relaxing* My hands: *writing*  My le...                •ɱĭss süğą•
2  Parents: You have to make us proud  Partner: Y...  Leian Xyrielle Dayahan
3  Little boy: “Are you an angel?”  Girl: “What?”...                   Lisha
4  Reasons to live:    “Suicide doesn’t stop the ...               Grace Cho

            comment_date  comment_likes
0  2021-01-01T10:58:20Z           3884
1  2021-01-29T15:42:43Z           4077
2  2021-03-09T00:52:35Z            827
3  2021-02-12T15:48:08Z            921
4  2021-02-02T18:39:00Z           2248
```

Data Pre-processing

1. Missing Value Analysis:

```
# Check for missing values
print(data.isnull().sum())

# Drop rows with missing values (if any)
data = data.dropna(subset=['comment_text'])
```

```
query             0
url               0
title             0
upload_date       0
channel           0
views             0
likes             0
dislikes          0
comment_count     0
comment_text      0
comment_author    0
comment_date      0
comment_likes     0
dtype: int64
```

2. Sentiment Analysis (Polarity and Subjectivity):

```
# Function to calculate polarity and subjectivity
def get_sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity, blob.sentiment.subjectivity

# Apply the function to the dataset to get polarity and subjectivity
data[['polarity', 'subjectivity']] = data['comment_text'].apply(lambda x: pd.Series(get_sentiment(x)))
```

```
from textblob import TextBlob

# Function to get the sentiment polarity and subjectivity
def get_sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity, blob.sentiment.subjectivity

# Apply the function to the dataset to get polarity and subjectivity
data[['polarity', 'subjectivity']] = data['comment_text'].apply(lambda x: pd.Series(get_sentiment(x)))

# Sample output: First few rows with polarity and subjectivity
print(data[['comment_text', 'polarity', 'subjectivity']].head())
```

```
                                        comment_text  polarity  subjectivity
0  ♪ Listen on Spotify!: https://spoti.fi/3gC9GfA...  0.230549      0.491982
1  My ears: *relaxing* My hands: *writing*  My le...  0.000000      0.000000
2  Parents: You have to make us proud  Partner: Y...  0.587500      0.800000
3  Little boy: “Are you an angel?”  Girl: “What?”...  0.071312      0.650000
4  Reasons to live:    “Suicide doesn’t stop the ...  0.251724      0.563024
```

3. Categorizing Sentiment:

```
def categorize_sentiment(polarity):
    if polarity > 0:
        return 'Positive'
    elif polarity == 0:
        return 'Neutral'
    else:
        return 'Negative'

data['sentiment'] = data['polarity'].apply(categorize_sentiment)
```

4. Encoding and Vectorization:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import LabelEncoder

# Encode the sentiment labels (Positive, Neutral, Negative)
le = LabelEncoder()
data['sentiment_encoded'] = le.fit_transform(data['sentiment'])

# Vectorizing the comment_text using CountVectorizer
```

```
vectorizer = CountVectorizer(stop_words='english')
X = vectorizer.fit_transform(data['comment_text'])

# Target variable
y = data['sentiment_encoded']
```

Model Creation

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Split data into train and test sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the Logistic Regression model
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
```

```
▾     LogisticRegression
LogisticRegression(max_iter=1000)
```

Model Evaluation

```
from sklearn.metrics import classification_report, confusion_matrix

# Make predictions
y_pred = log_reg.predict(X_test)

# Classification report
classification_rep = classification_report(y_test, y_pred, target_names=le.classes_)
print("Classification Report: \n", classification_rep)

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix: \n", conf_matrix)
```

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

    Negative       0.81      0.56      0.66       443
     Neutral       0.88      0.94      0.91      2722
    Positive       0.90      0.88      0.89      2494

    accuracy                           0.88      5659
   macro avg       0.86      0.79      0.82      5659
weighted avg       0.88      0.88      0.88      5659

Confusion Matrix:
 [[ 246  106   91]
 [  19 2550  153]
 [  38  257 2199]]
```

Predicting Sentiment for New Comments

```
# Function to predict sentiment along with polarity and subjectivity for new comments
def predict_sentiment_with_polarity_subjectivity(new_comments):
    # Vectorize the new comments for sentiment classification
    new_comments_vectorized = vectorizer.transform(new_comments)

    # Predict the sentiment (encoded)
    predictions_encoded = log_reg.predict(new_comments_vectorized)

    # Convert encoded predictions back to sentiment labels
    predicted_sentiments = le.inverse_transform(predictions_encoded)

    # Get polarity and subjectivity using TextBlob
    polarity_subjectivity = [get_sentiment(comment) for comment in new_comments]

    # Combine all the information and return the results
    results = []
    for comment, sentiment, (polarity, subjectivity) in zip(new_comments, predicted_sentiments, polarity_subjectivity):
        results.append({
            'Comment': comment,
            'Predicted Sentiment': sentiment,
            'Polarity': polarity,
            'Subjectivity': subjectivity
        })

    return results

# Example new comments for testing
new_comments = [
    "I love you",
    "I didn't like this video at all, but it wasn't a waste of time",
    "I hate you"
]

# Predicting sentiment, polarity, and subjectivity for the new comments
predicted_results = predict_sentiment_with_polarity_subjectivity(new_comments)

# Display the results
for result in predicted_results:
    print(f"Comment: '{result['Comment']}'")
    print(f"Predicted Sentiment: {result['Predicted Sentiment']}")
    print(f"Polarity: {result['Polarity']}")
    print(f"Subjectivity: {result['Subjectivity']}")
    print("\n")
```

```
Comment: 'I love you'
Predicted Sentiment: Positive
Polarity: 0.5
Subjectivity: 0.6


Comment: 'I didn't like this video at all, but it wasn't a waste of time'
Predicted Sentiment: Neutral
Polarity: -0.2
Subjectivity: 0.0


Comment: 'I hate you'
Predicted Sentiment: Negative
Polarity: -0.8
Subjectivity: 0.9
```