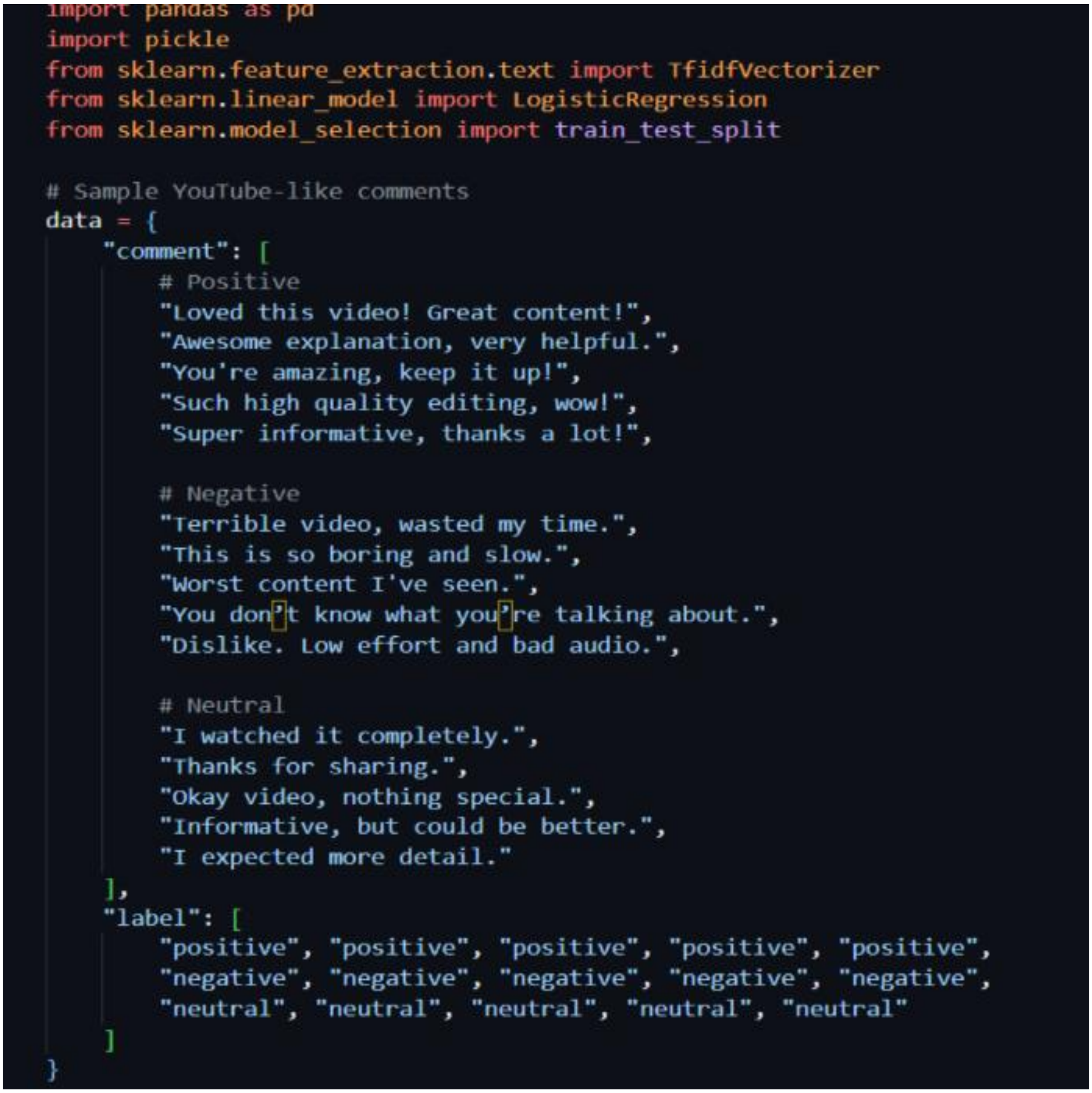
CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

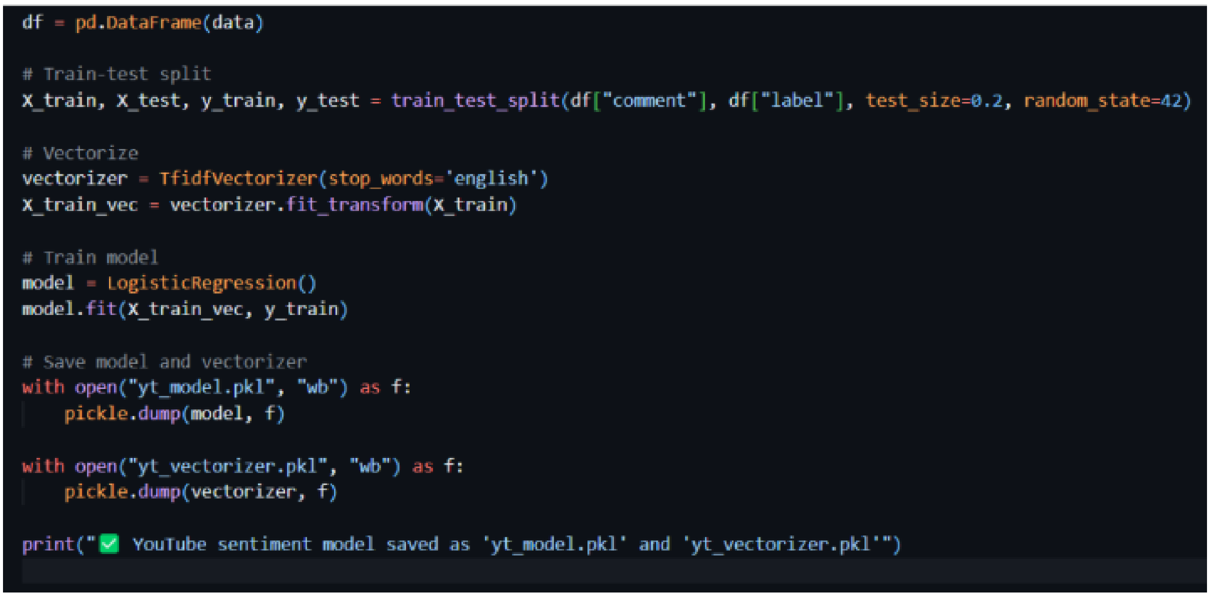
**Name:** Johnson Kumar **Uid:** 232bcs12654

**Topic:** YouTube Comment Sentiment: Scrape comments → Analyze for toxicity or sentiment.

Model training code:

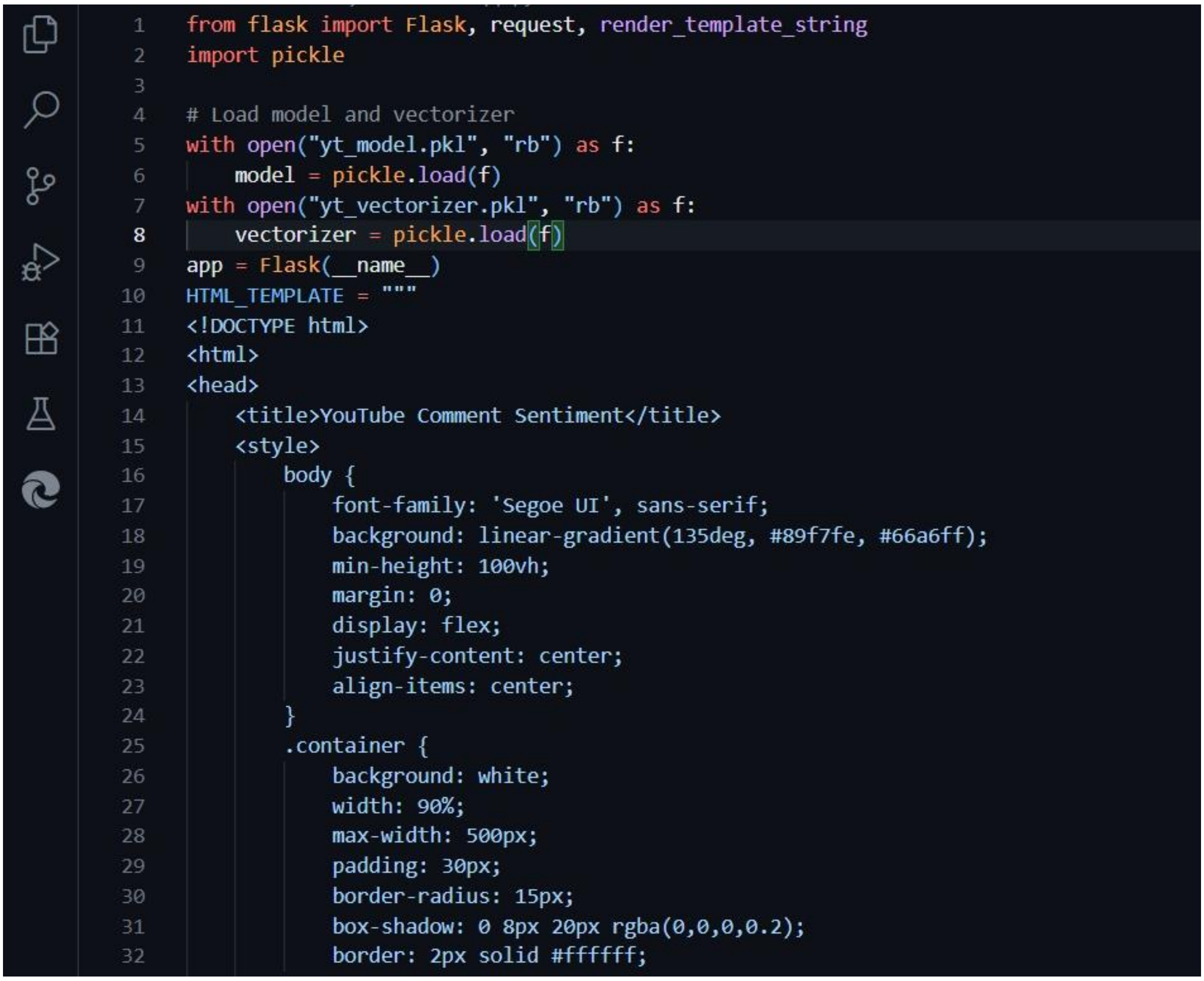


```
import pandas as pd
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

# Sample YouTube-like comments
data = {
    "comment": [
        # Positive
        "Loved this video! Great content!",
        "Awesome explanation, very helpful.",
        "You're amazing, keep it up!",
        "Such high quality editing, wow!",
        "Super informative, thanks a lot!",

        # Negative
        "Terrible video, wasted my time.",
        "This is so boring and slow.",
        "Worst content I've seen.",
        "You don’t know what you’re talking about.",
        "Dislike. Low effort and bad audio.",

        # Neutral
        "I watched it completely.",
        "Thanks for sharing.",
        "Okay video, nothing special.",
        "Informative, but could be better.",
        "I expected more detail."
    ],
    "label": [
        "positive", "positive", "positive", "positive", "positive",
        "negative", "negative", "negative", "negative", "negative",
        "neutral", "neutral", "neutral", "neutral", "neutral"
    ]
}
```

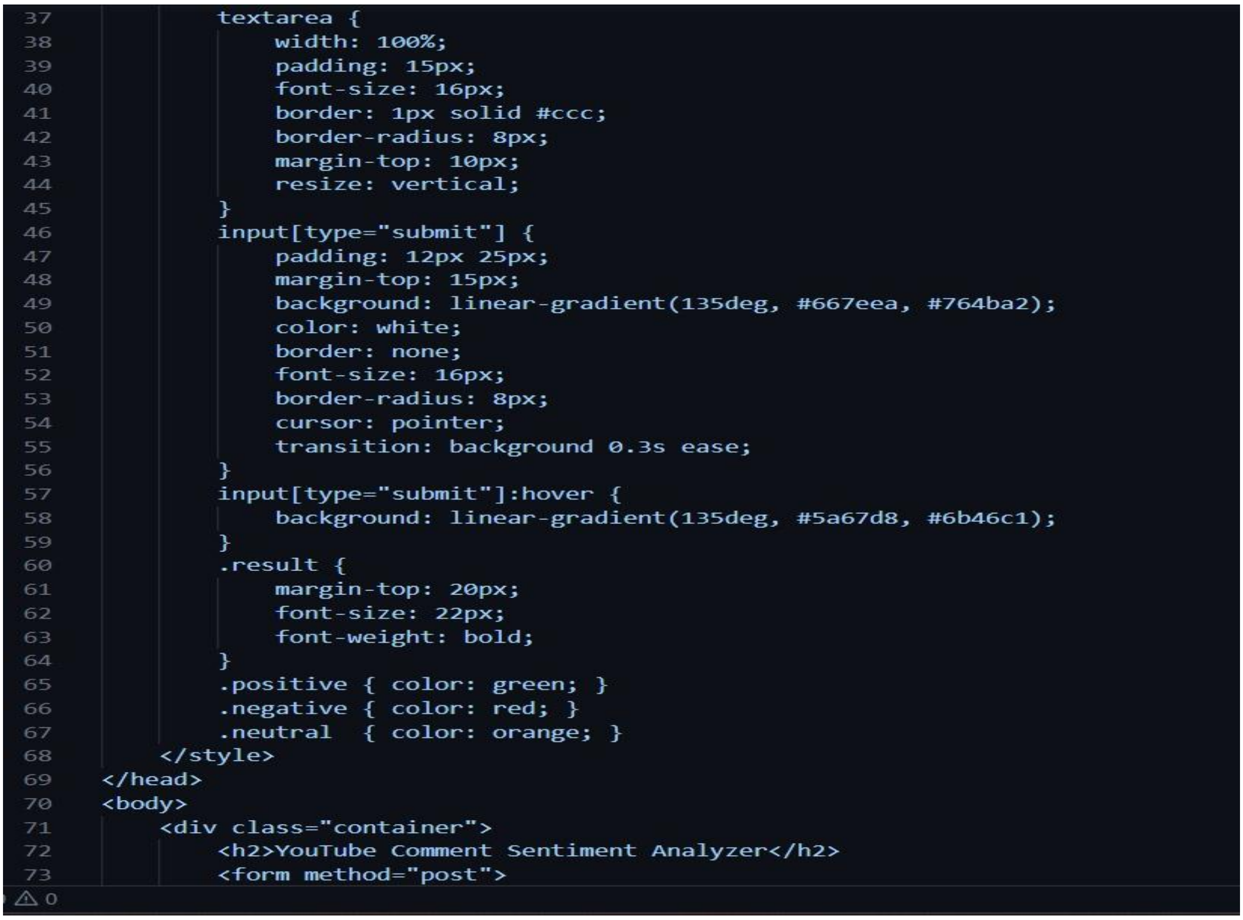


```python
df = pd.DataFrame(data)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(df["comment"], df["label"], test_size=0.2, random_state=42)

# Vectorize
vectorizer = TfidfVectorizer(stop_words='english')
X_train_vec = vectorizer.fit_transform(X_train)

# Train model
model = LogisticRegression()
model.fit(X_train_vec, y_train)

# Save model and vectorizer
with open("yt_model.pkl", "wb") as f:
    pickle.dump(model, f)

with open("yt_vectorizer.pkl", "wb") as f:
    pickle.dump(vectorizer, f)

print("✅ YouTube sentiment model saved as 'yt_model.pkl' and 'yt_vectorizer.pkl'")
```

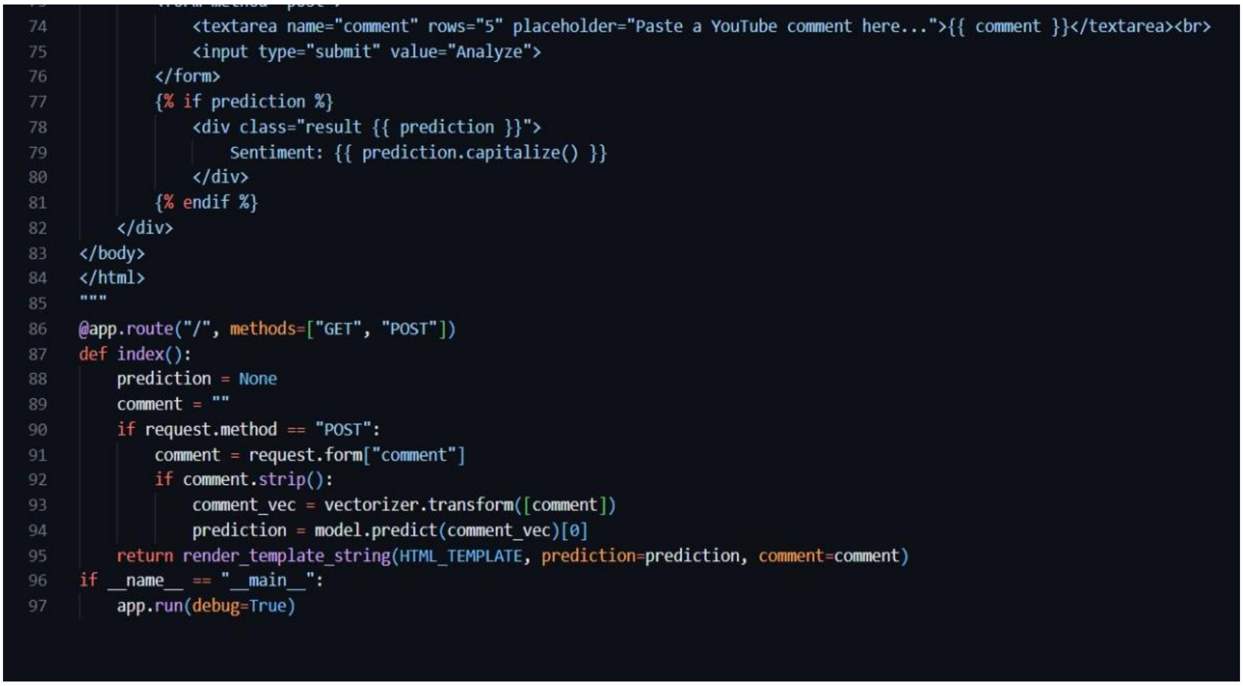
## Flask Code:



```python
from flask import Flask, request, render_template_string
import pickle

# Load model and vectorizer
with open("yt_model.pkl", "rb") as f:
    model = pickle.load(f)
with open("yt_vectorizer.pkl", "rb") as f:
    vectorizer = pickle.load(f)
app = Flask(__name__)
HTML_TEMPLATE = """
<!DOCTYPE html>
<html>
<head>
    <title>YouTube Comment Sentiment</title>
    <style>
        body {
            font-family: 'Segoe UI', sans-serif;
            background: linear-gradient(135deg, #89f7fe, #66a6ff);
            min-height: 100vh;
            margin: 0;
            display: flex;
            justify-content: center;
            align-items: center;
        }
        .container {
            background: white;
            width: 90%;
            max-width: 500px;
            padding: 30px;
            border-radius: 15px;
            box-shadow: 0 8px 20px rgba(0,0,0,0.2);
            border: 2px solid #ffffff;
```

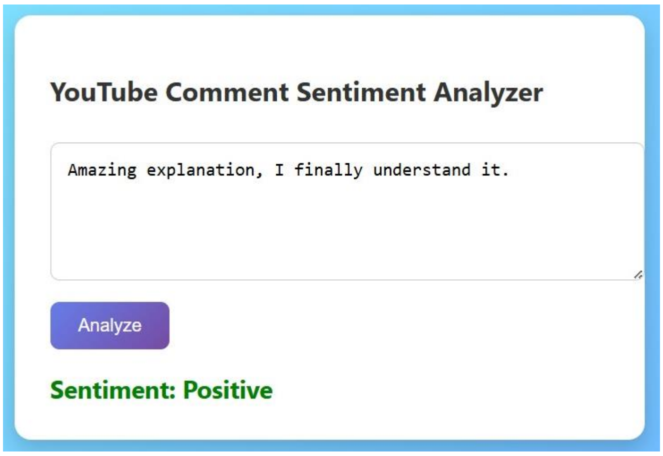


```
        textarea {
            width: 100%;
            padding: 15px;
            font-size: 16px;
            border: 1px solid #ccc;
            border-radius: 8px;
            margin-top: 10px;
            resize: vertical;
        }
        input[type="submit"] {
            padding: 12px 25px;
            margin-top: 15px;
            background: linear-gradient(135deg, #667eea, #764ba2);
            color: white;
            border: none;
            font-size: 16px;
            border-radius: 8px;
            cursor: pointer;
            transition: background 0.3s ease;
        }
        input[type="submit"]:hover {
            background: linear-gradient(135deg, #5a67d8, #6b46c1);
        }
        .result {
            margin-top: 20px;
            font-size: 22px;
            font-weight: bold;
        }
        .positive { color: green; }
        .negative { color: red; }
        .neutral  { color: orange; }
    </style>
</head>
<body>
    <div class="container">
        <h2>YouTube Comment Sentiment Analyzer</h2>
        <form method="post">
```



```
            <textarea name="comment" rows="5" placeholder="Paste a YouTube comment here...">{{ comment }}</textarea><br>
            <input type="submit" value="Analyze">
        </form>
        {% if prediction %}
            <div class="result {{ prediction }}">
                Sentiment: {{ prediction.capitalize() }}
            </div>
        {% endif %}
    </div>
</body>
</html>
"""
@app.route("/", methods=["GET", "POST"])
def index():
    prediction = None
    comment = ""
    if request.method == "POST":
        comment = request.form["comment"]
        if comment.strip():
            comment_vec = vectorizer.transform([comment])
            prediction = model.predict(comment_vec)[0]
    return render_template_string(HTML_TEMPLATE, prediction=prediction, comment=comment)
if __name__ == "__main__":
    app.run(debug=True)
```

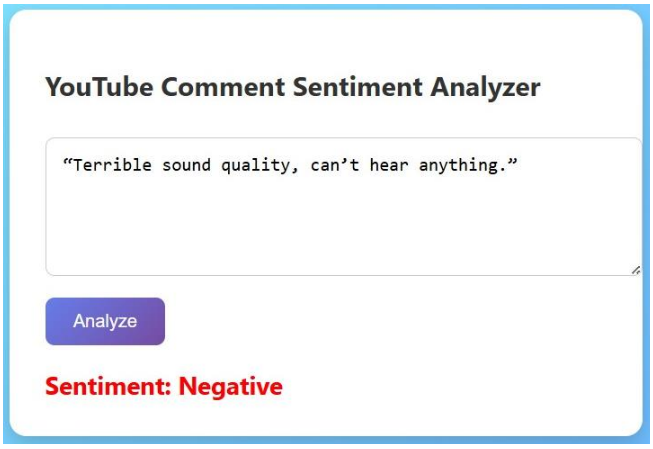
## Output:



YouTube Comment Sentiment Analyzer

```
Amazing explanation, I finally understand it.
```

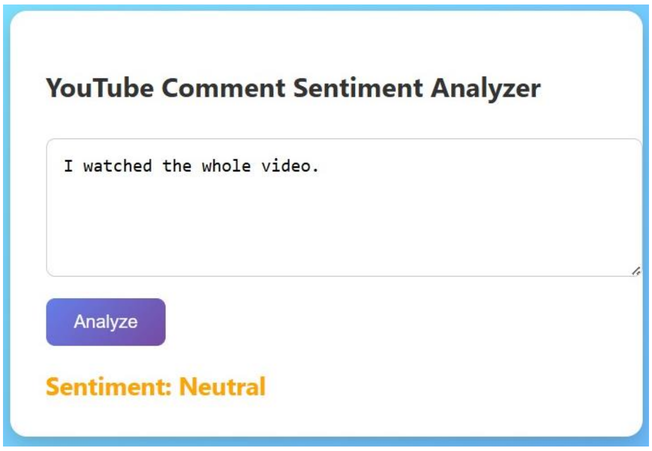
Sentiment: Positive



YouTube Comment Sentiment Analyzer

```
“Terrible sound quality, can’t hear anything.”
```

Sentiment: Negative



YouTube Comment Sentiment Analyzer

```
I watched the whole video.
```

Sentiment: Neutral