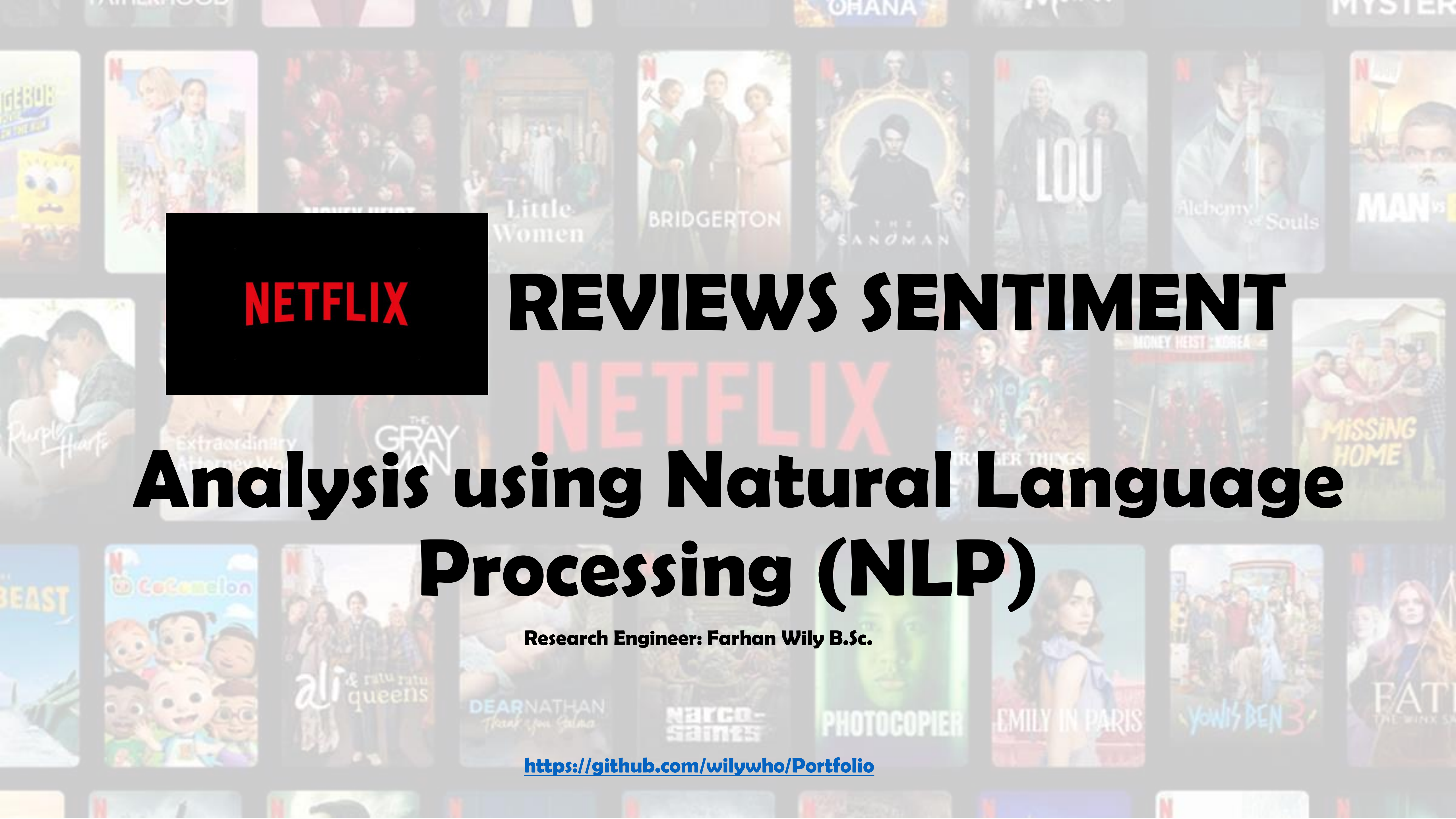


# NETFLIX REVIEWS SENTIMENT Analysis using Natural Language Processing (NLP)

**Research Engineer: Farhan Wily B.Sc.**

https://github.com/wilywho/Portfolio

# What is Natural Language Processing?

- A branch of artificial intelligence (AI) that deals with the interaction between computers and humans using natural language.

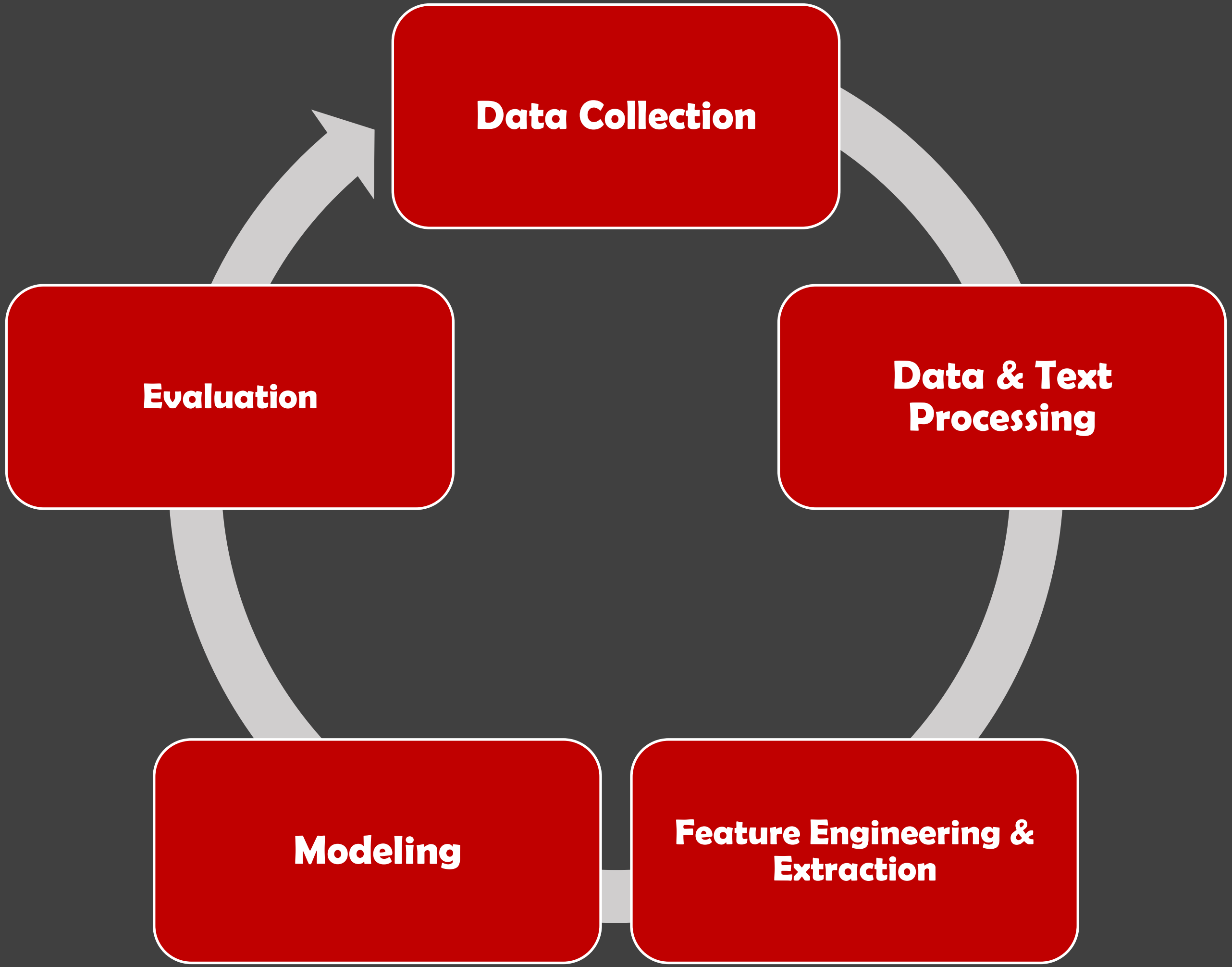
- The goal is to enable computers to understand, interpret, and generate human language in a useful way. NLP encompasses various techniques and methods for analyzing text and spoken language.

# Key Concepts in NLP

- **Tokenization**: Breaking down text into smaller units like words or sentences.
- **Stemming and Lemmatization**: Reducing words to their base form (e.g., "running" becomes "run").
- **Part-of-Speech Tagging (POS Tagging)**: Tagging each word in a text with its part of speech, such as noun, verb, etc.
- **Named Entity Recognition (NER)**: Identifying and classifying named entities like names of people, places, organizations in text.
- **Parsing**: Analyzing the grammatical structure of a sentence.
- **Sentiment Analysis**: Determining the sentiment or emotion contained in a text (positive, negative, neutral).
- **Word Embeddings**: Representations of words in numerical vector form that allow processing by machine learning algorithms (e.g., Word2Vec, GloVe).
- **Topic Modeling**: Identifying the main topics discussed in a collection of documents (e.g., LDA).

# NLP Workflow

- In this NLP project, we use several steps to achieve the research objectives



Data Collection

Data & Text Processing

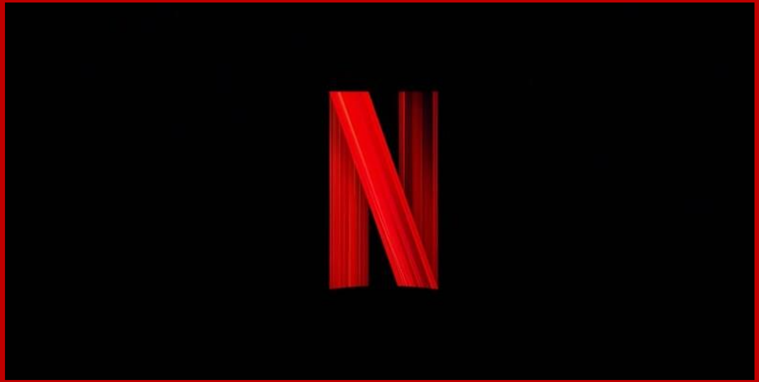
Feature Engineering & Extraction

Modeling

Evaluation

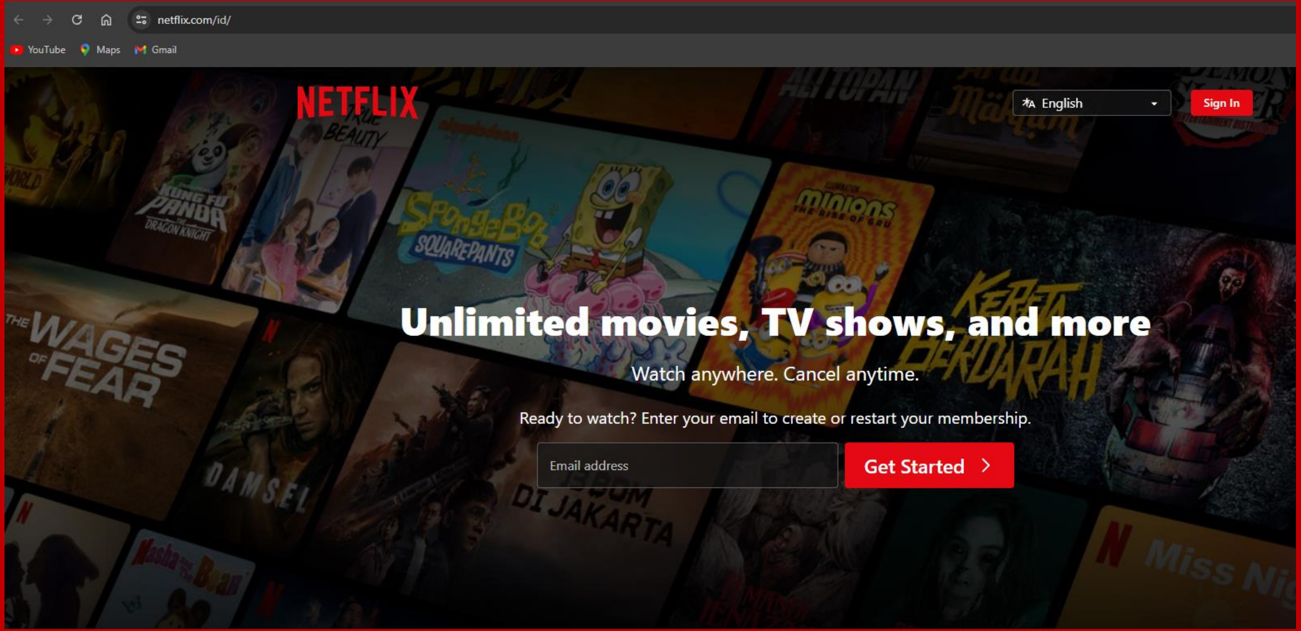
# What is Netflix?

**Netflix** is a subscription-based streaming service that allows its members to watch TV shows and movies on Internet-connected devices.



## Reviews Sentiment Analysis

In this project, we aim to analyze the sentiment of reviews from members regarding the use of Netflix services and categorize them into **positive, negative,** and **neutral** reviews to gauge member satisfaction with the services provided by **Netflix**.



Unlimited movies, TV shows, and more

Watch anywhere. Cancel anytime.

Ready to watch? Enter your email to create or restart your membership.

# Data Collection

- In this project, we use data from Kaggle through this link:

https://www.kaggle.com/datasets/ashishkumarak/netflix-reviews-playstore-daily-updated/data

The data in CSV format. Here is the data info,

### Data Info

| Column | Non-Null Count | Data Type |
|---|---|---|
| reviewId | 113622 | object |
| username | 113620 | object |
| content | 113620 | object |
| score | 113622 | int64 |
| thumbsUpCount | 113622 | int64 |
| reviewCreatedVersion | 96983 | object |
| at | 113622 | object |
| appVersion | 96983 | object |

- Data size : 6.9+ MB
- Range Index : 113622

- From the data info, we get to know that the data contains 8 features: **reviewId**, **username**, **content**, **score**, **thumbsUpCount**, **reviewCreatedVersion**, **at**, and **appVersion**.

# Data Collection

## Features Info

1. **reviewId:**

- **Description**: A unique ID that identifies each review.
- **Function**: Used as a unique identifier for each review entry. It helps in tracking and managing individual reviews.

2. **userName:**

- **Description:** The username of the person who provided the review**.**
- **Function:** Identifies the user who gave the review. Can be used for analyzing review patterns based on users.

3. **content:**

- **Description:** The text of the review given by the user**.**
- **Function:** The core of the review where users express their opinions. Used fot sentiment analysis and other text processing**.**

4. **score:**

- **Description:** The rating given by the user in numeric form, usually on a scale of 1-5.
- **Function:** Measures user satisfaction with the service or content. Used for sentiment classification (e.g., positive, negative, or neutral reviews).

# Data Collection

## Features Info

5. **thumbsUpCount:**

- **Description:** The number of “thumbs up” or “likes” that the review has received.
- **Function**: Measures how useful or agreeable the review is according to other users. Can be used to assess the influence or popularity of the review.

6. **reviewCreatedVersion:**

- **Description:** The version of the app at the time the review was created.
- **Function:** Indicates the app version used by the user when giving the review. Useful for tracking specific issues or feedback related to certain versions.

7. **at:**

- **Description:** The date and time when the review was created.
- **Function:** Indicates when the review was given. Useful for time trend analysis and understanding changes in user satisfaction over time.

8. **appVersion:**

- **Description:** The app version currently used by the user.
- **Function**: Identifies the app version in use at the time of the review. Can help in analyzing app versions in relation to user reviews.

# Data & Text Processing

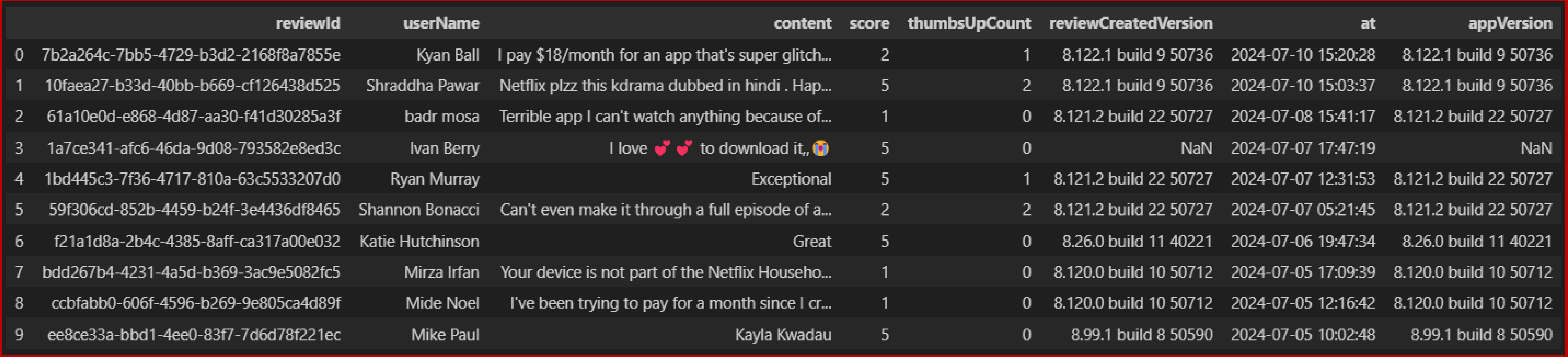
## Read Dataset

**We use Pandas from Python Library to read the data in CSV format through this code:**

- `# Read the dataset`
- `df = pd.read_csv(‘Your Disk:/Your Source Folder/netflix_reviews.csv')`

**Use this code to show the dataframe:**

- `# Show the dataset`
- `df.head(10)`
- Here is the output,



| | reviewId | userName | content | score | thumbsUpCount | reviewCreatedVersion | at | appVersion |
|---|---|---|---|---|---|---|---|---|
| 0 | 7b2a264c-7bb5-4729-b3d2-2168f8a7855e | Kyan Ball | I pay $18/month for an app that's super glitch... | 2 | 1 | 8.122.1 build 9 50736 | 2024-07-10 15:20:28 | 8.122.1 build 9 50736 |
| 1 | 10faea27-b33d-40bb-b669-cf126438d525 | Shraddha Pawar | Netflix plzz this kdrama dubbed in hindi . Hap... | 5 | 2 | 8.122.1 build 9 50736 | 2024-07-10 15:03:37 | 8.122.1 build 9 50736 |
| 2 | 61a10e0d-e868-4d87-aa30-f41d30285a3f | badr mosa | Terrible app I can't watch anything because of... | 1 | 0 | 8.121.2 build 22 50727 | 2024-07-08 15:41:17 | 8.121.2 build 22 50727 |
| 3 | 1a7ce341-afc6-46da-9d08-793582e8ed3c | Ivan Berry | I love 💕 💕 to download it,, 😭 | 5 | 0 | NaN | 2024-07-07 17:47:19 | NaN |
| 4 | 1bd445c3-7f36-4717-810a-63c5533207d0 | Ryan Murray | Exceptional | 5 | 1 | 8.121.2 build 22 50727 | 2024-07-07 12:31:53 | 8.121.2 build 22 50727 |
| 5 | 59f306cd-852b-4459-b24f-3e4436df8465 | Shannon Bonacci | Can't even make it through a full episode of a... | 2 | 2 | 8.121.2 build 22 50727 | 2024-07-07 05:21:45 | 8.121.2 build 22 50727 |
| 6 | f21a1d8a-2b4c-4385-8aff-ca317a00e032 | Katie Hutchinson | Great | 5 | 0 | 8.26.0 build 11 40221 | 2024-07-06 19:47:34 | 8.26.0 build 11 40221 |
| 7 | bdd267b4-4231-4a5d-b369-3ac9e5082fc5 | Mirza Irfan | Your device is not part of the Netflix Househo... | 1 | 0 | 8.120.0 build 10 50712 | 2024-07-05 17:09:39 | 8.120.0 build 10 50712 |
| 8 | ccbfabb0-606f-4596-b269-9e805ca4d89f | Mide Noel | I've been trying to pay for a month since I cr... | 1 | 0 | 8.120.0 build 10 50712 | 2024-07-05 12:16:42 | 8.120.0 build 10 50712 |
| 9 | ee8ce33a-bbd1-4ee0-83f7-7d6d78f221ec | Mike Paul | Kayla Kwadau | 5 | 0 | 8.99.1 build 8 50590 | 2024-07-05 10:02:48 | 8.99.1 build 8 50590 |

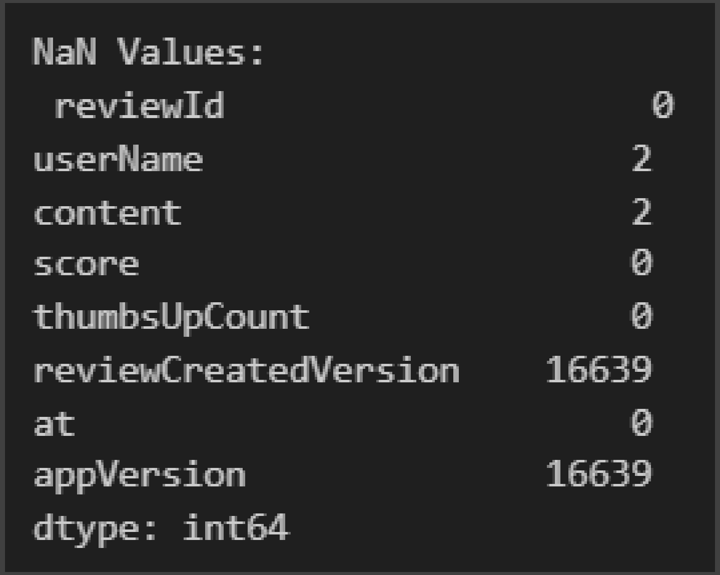
# Data & Text Processing

## NaN and Null Values Check

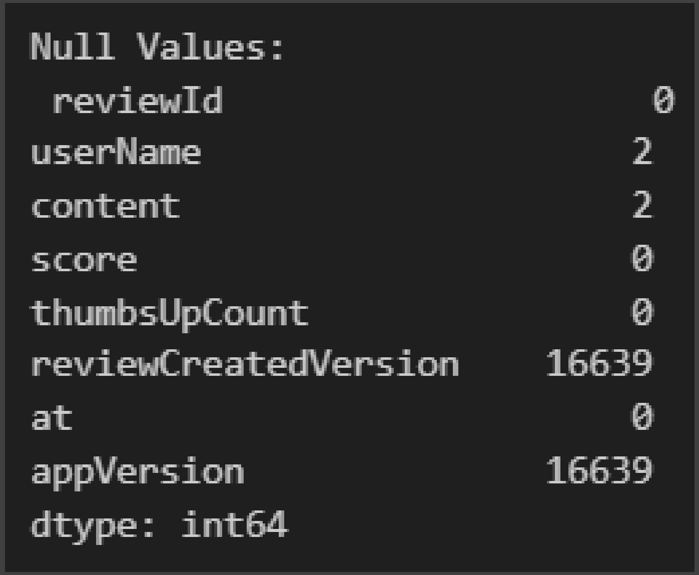
**We use Pandas from Python Library to CHECK NaN and Null values from the data in CSV format through this code:**

- `# NaN and Null values`
- `check_nan = df.isna().sum()`
- `check_null = df.isnull().sum()`
- `print('NaN Values:\n', check_nan, '\n')`
- `print('Null Values:\n', check_null, '\n')`

### Output



```
NaN Values:
 reviewId                 0
userName                 2
content                  2
score                    0
thumbsUpCount            0
reviewCreatedVersion 16639
at                       0
appVersion           16639
dtype: int64
```



```
Null Values:
 reviewId                 0
userName                 2
content                  2
score                    0
thumbsUpCount            0
reviewCreatedVersion 16639
at                       0
appVersion           16639
dtype: int64
```

# Data & Text Processing

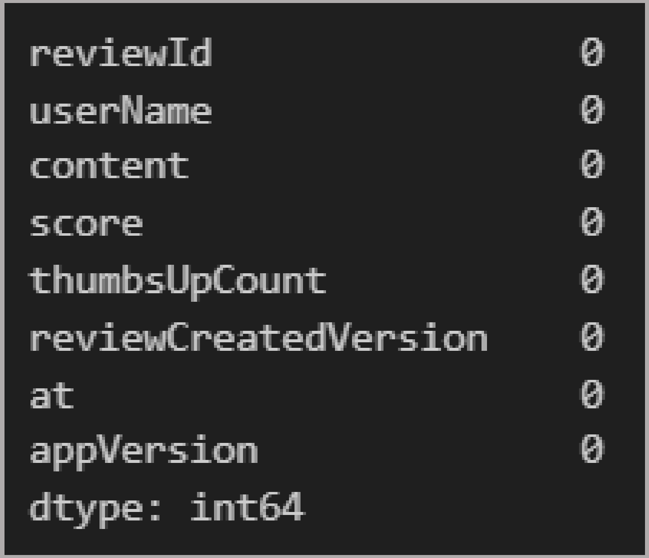
## Cleaning the NaN and Null Values

We need to fill the NaN & Null in the userName, content, reviewCreatedVersion, and appVersion column with ‘ ‘.

```
# Resolve NaN/Null values with empty strings
df['userName'].fillna('', inplace=True)
df['content'].fillna('', inplace=True)
df['reviewCreatedVersion'].fillna('', inplace=True)
df['appVersion'].fillna('', inplace=True)
```

## Output

```
# Validate results
print(df.isnull().sum())
```



```
reviewId                0
userName                0
content                 0
score                   0
thumbsUpCount           0
reviewCreatedVersion    0
at                      0
appVersion              0
dtype: int64
```

So, we have cleaning the NaN and Null values from the data

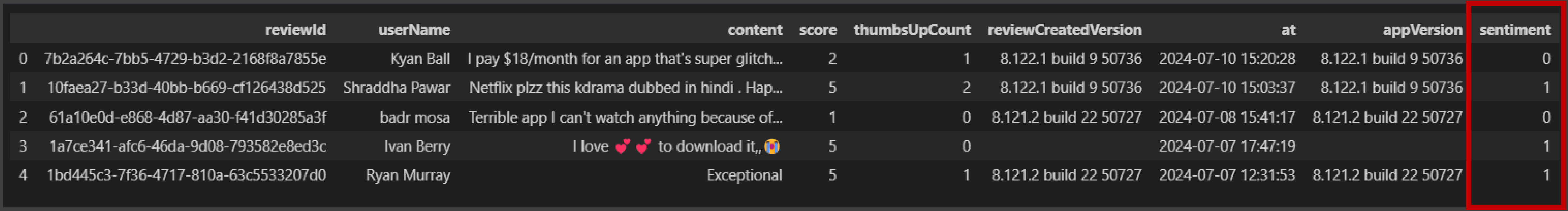
# Data & Text Processing

## Created Sentiment Column as Sentiment Analysis Parameter

The sentiment column is created based on the score given by Netflix users regarding their satisfaction with the content and services provided. The sentiment column is created using binary values 0 and 1 to determine whether the user is satisfied or not. The number 0 represents the 'not satisfied' category, and the number 1 represents the 'satisfied' category. Here is the code that we use to create sentiment column,

-
  ```
  df['sentiment'] = df['score'].apply(lambda x: 1 if x >= 3 else 0)
  ```

## Output



| | reviewId | userName | content | score | thumbsUpCount | reviewCreatedVersion | at | appVersion | sentiment |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 7b2a264c-7bb5-4729-b3d2-2168f8a7855e | Kyan Ball | I pay $18/month for an app that's super glitch... | 2 | 1 | 8.122.1 build 9 50736 | 2024-07-10 15:20:28 | 8.122.1 build 9 50736 | 0 |
| 1 | 10faea27-b33d-40bb-b669-cf126438d525 | Shraddha Pawar | Netflix plzz this kdrama dubbed in hindi . Hap... | 5 | 2 | 8.122.1 build 9 50736 | 2024-07-10 15:03:37 | 8.122.1 build 9 50736 | 1 |
| 2 | 61a10e0d-e868-4d87-aa30-f41d30285a3f | badr mosa | Terrible app I can't watch anything because of... | 1 | 0 | 8.121.2 build 22 50727 | 2024-07-08 15:41:17 | 8.121.2 build 22 50727 | 0 |
| 3 | 1a7ce341-afc6-46da-9d08-793582e8ed3c | Ivan Berry | I love 💕 💕 to download it,, 😭 | 5 | 0 | | 2024-07-07 17:47:19 | | 1 |
| 4 | 1bd445c3-7f36-4717-810a-63c5533207d0 | Ryan Murray | Exceptional | 5 | 1 | 8.121.2 build 22 50727 | 2024-07-07 12:31:53 | 8.121.2 build 22 50727 | 1 |

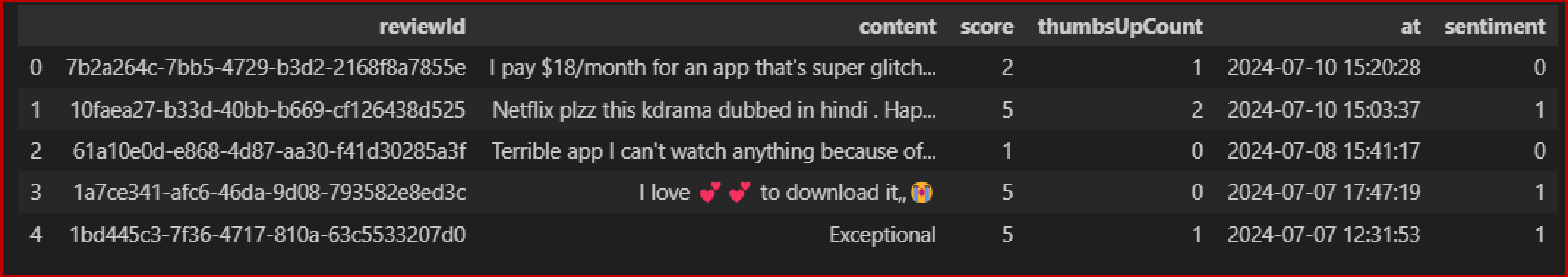
# Feature Engineering & Extraction

### Feature Selection

We only need features **reviewId**, **content**, **score**, **thumbsUpCount** and **at** which use to the next analysis**.** So, we drop another column with use this code,

```
# Delete columns
df.drop(['userName', 'reviewCreatedVersion', 'appVersion'], axis = 1, inplace = True)
```

### Output



| | reviewId | content | score | thumbsUpCount | at | sentiment |
|---|---|---|---|---|---|---|
| 0 | 7b2a264c-7bb5-4729-b3d2-2168f8a7855e | I pay $18/month for an app that's super glitch... | 2 | 1 | 2024-07-10 15:20:28 | 0 |
| 1 | 10faea27-b33d-40bb-b669-cf126438d525 | Netflix plzz this kdrama dubbed in hindi . Hap... | 5 | 2 | 2024-07-10 15:03:37 | 1 |
| 2 | 61a10e0d-e868-4d87-aa30-f41d30285a3f | Terrible app I can't watch anything because of... | 1 | 0 | 2024-07-08 15:41:17 | 0 |
| 3 | 1a7ce341-afc6-46da-9d08-793582e8ed3c | I love 💕 💕 to download it,, 😭 | 5 | 0 | 2024-07-07 17:47:19 | 1 |
| 4 | 1bd445c3-7f36-4717-810a-63c5533207d0 | Exceptional | 5 | 1 | 2024-07-07 12:31:53 | 1 |

# Feature Engineering & Extraction

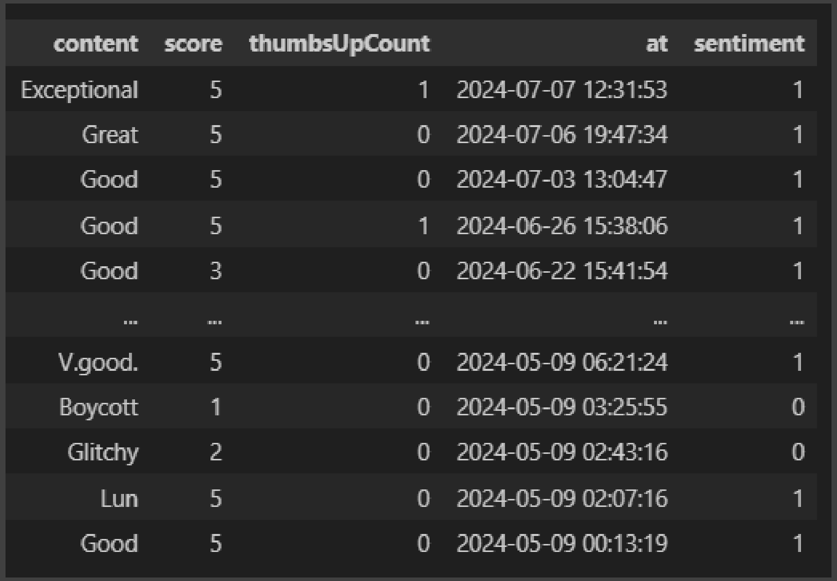
## Feature Engineering

In this section, we want to identify the **maximum**, **minimum**, and **average number** of words in the review sentences in the content column to know how many words containing in the longest review, and then we want to identified the content column has only one word and only emojis as review.

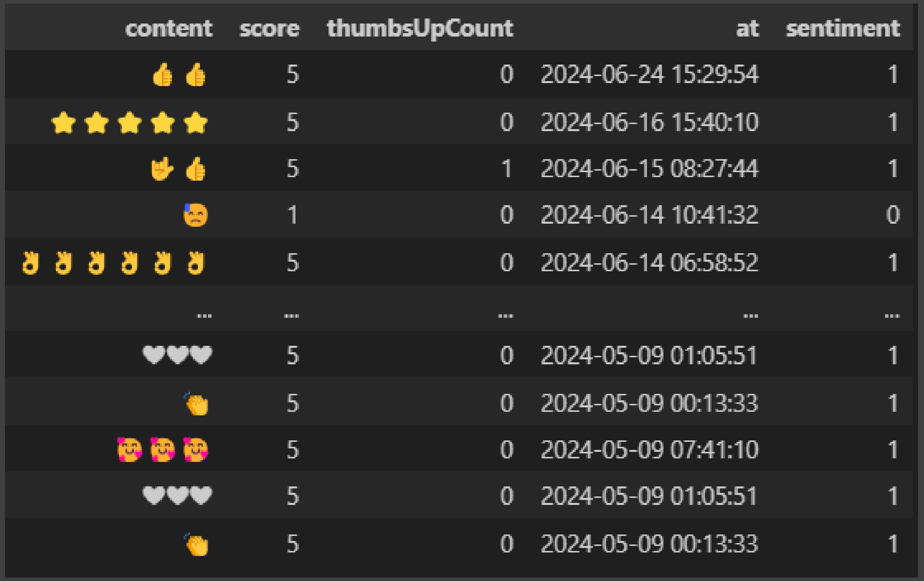
### Count of Words Information

| Info | Count of words |
|---|---|
| Maximum | 331 words |
| Minimum | 0 words |
| Average | 30 words |

### Content that contains only one word and only emojis



| | content | score | thumbsUpCount | at | sentiment |
|---|---|---|---|---|---|
| | Exceptional | 5 | 1 | 2024-07-07 12:31:53 | 1 |
| | Great | 5 | 0 | 2024-07-06 19:47:34 | 1 |
| | Good | 5 | 0 | 2024-07-03 13:04:47 | 1 |
| | Good | 5 | 1 | 2024-06-26 15:38:06 | 1 |
| | Good | 3 | 0 | 2024-06-22 15:41:54 | 1 |
| | ... | ... | ... | ... | ... |
| | V.good. | 5 | 0 | 2024-05-09 06:21:24 | 1 |
| | Boycott | 1 | 0 | 2024-05-09 03:25:55 | 0 |
| | Glitchy | 2 | 0 | 2024-05-09 02:43:16 | 0 |
| | Lun | 5 | 0 | 2024-05-09 02:07:16 | 1 |
| | Good | 5 | 0 | 2024-05-09 00:13:19 | 1 |

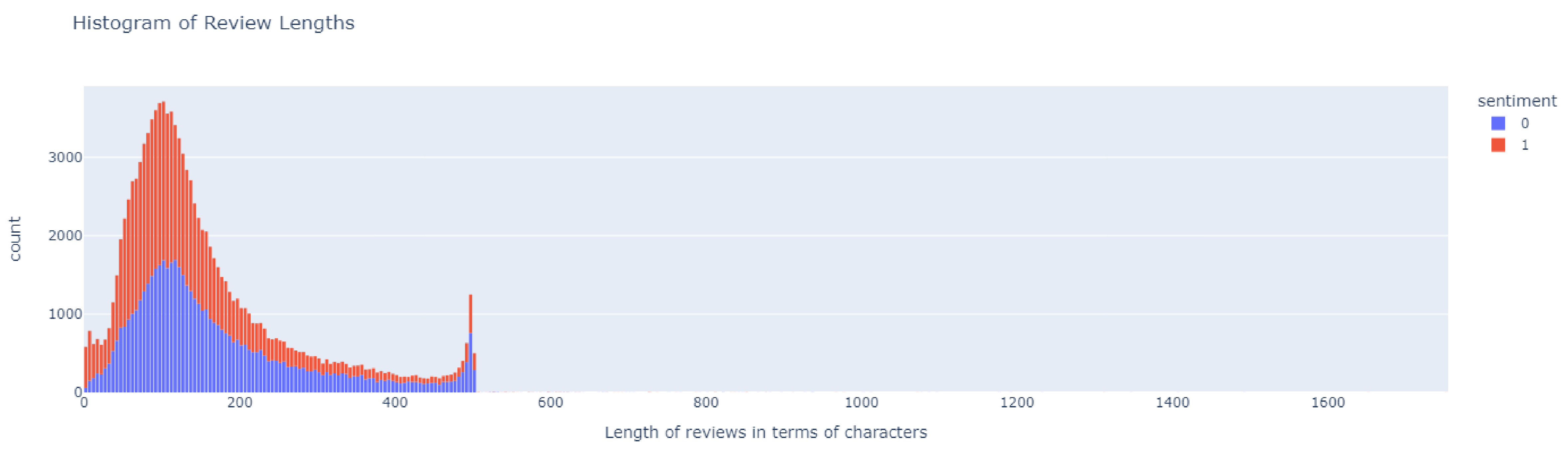


| content | score | thumbsUpCount | at | sentiment |
|---|---|---|---|---|
| 👍 👍 | 5 | 0 | 2024-06-24 15:29:54 | 1 |
| ⭐ ⭐ ⭐ ⭐ ⭐ | 5 | 0 | 2024-06-16 15:40:10 | 1 |
| 🤟 👍 | 5 | 1 | 2024-06-15 08:27:44 | 1 |
| 😓 | 1 | 0 | 2024-06-14 10:41:32 | 0 |
| 👌 👌 👌 👌 👌 👌 | 5 | 0 | 2024-06-14 06:58:52 | 1 |
| ... | ... | ... | ... | ... |
| 🤍🤍🤍 | 5 | 0 | 2024-05-09 01:05:51 | 1 |
| 👏 | 5 | 0 | 2024-05-09 00:13:33 | 1 |
| 🥰🥰🥰 | 5 | 0 | 2024-05-09 07:41:10 | 1 |
| 🤍🤍🤍 | 5 | 0 | 2024-05-09 01:05:51 | 1 |
| 👏 | 5 | 0 | 2024-05-09 00:13:33 | 1 |

# Feature Engineering & Extraction

## Feature Engineering

## Data Visualization using Plotly and Matplotlib



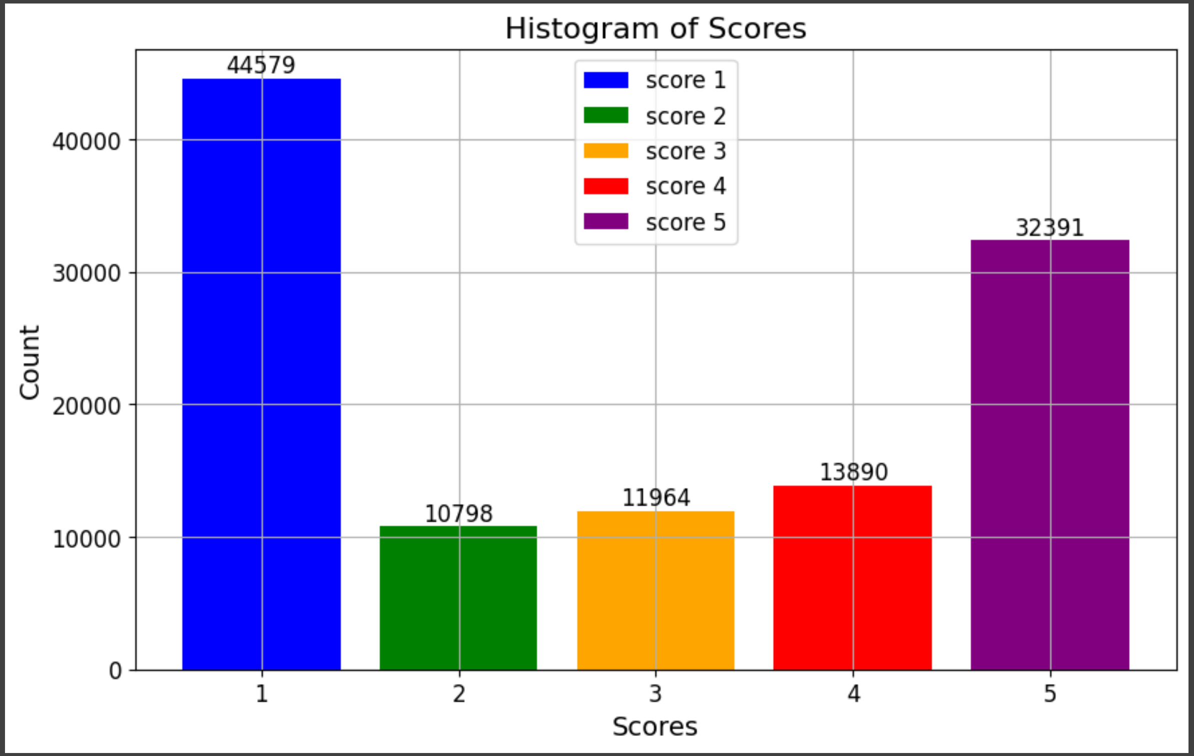
**Count of Words Information**

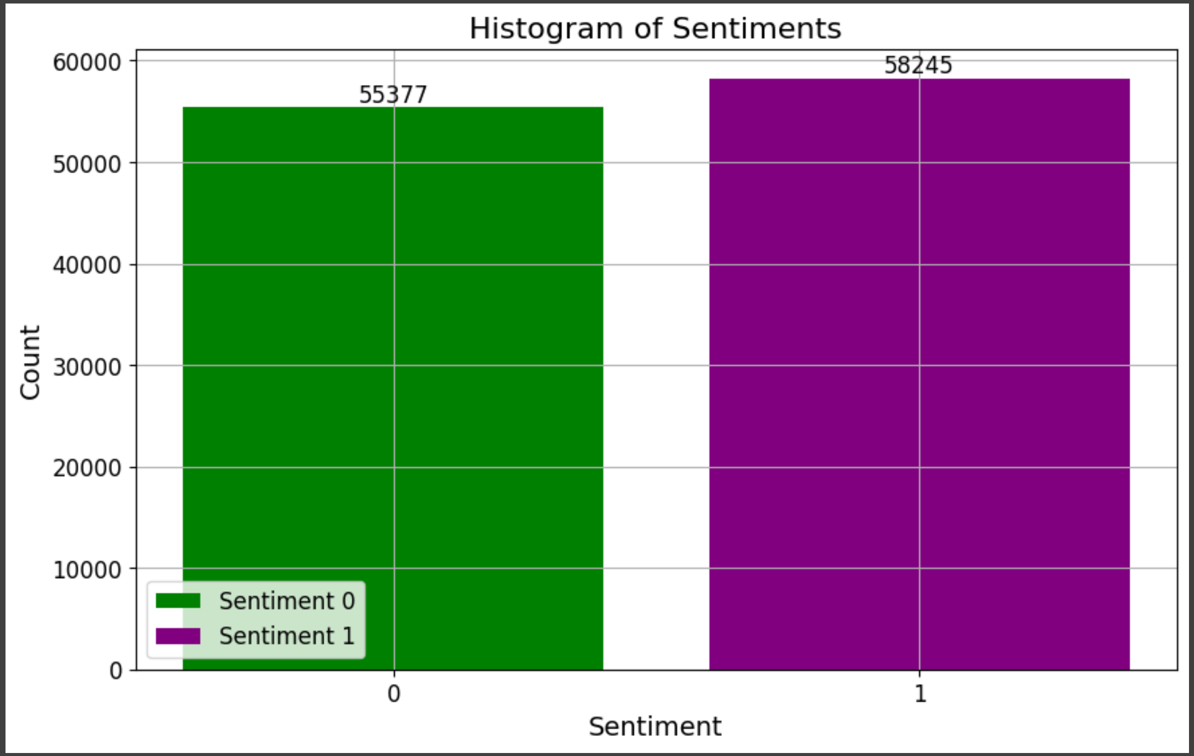
| Info | Count of words |
|---|---|
| Maximum | 331 words |
| Minimum | 0 words |
| Average | 30 words |

# Feature Engineering & Extraction

## Feature Engineering

## Data Visualization using Plotly and Matplotlib





Histogram of Scores to show the length of data from each scores

Histogram of Sentiments to show the length of data from each sentiments

# Feature Engineering & Extraction

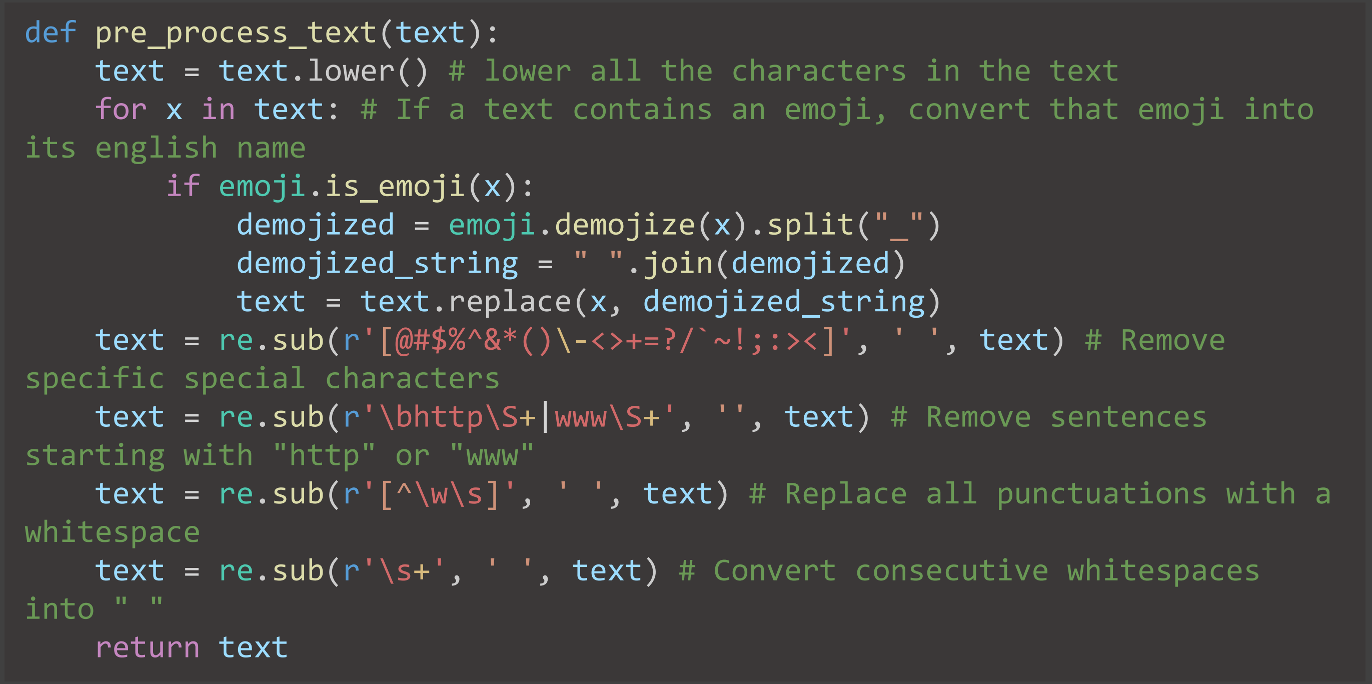
## Feature Engineering

### Convert Text in Content Column

This function takes as input a text and performs the following preprocessing on it:

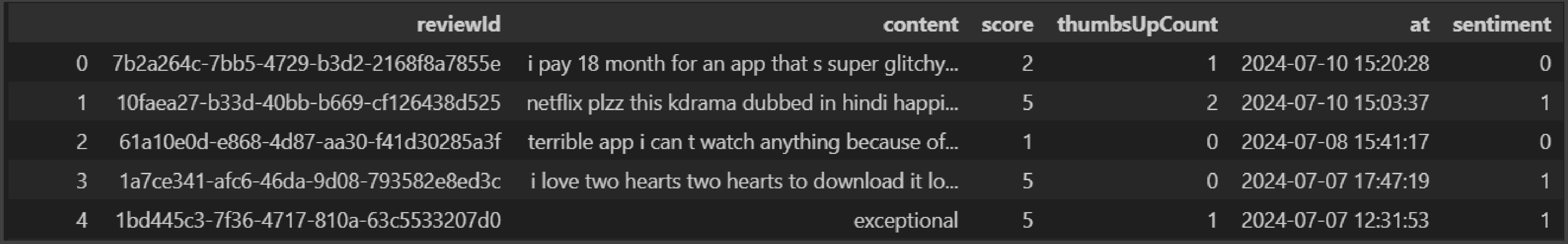
1. lower the text.
2. convert emojis into their english names.
3. remove special characters and punctuations.
4. remove urls.
5. remove all punctuations.
6. remove extra whitespaces.

### Function Code:



```
def pre_process_text(text):
    text = text.lower() # lower all the characters in the text
    for x in text: # If a text contains an emoji, convert that emoji into
its english name
        if emoji.is_emoji(x):
            demojized = emoji.demojize(x).split("_")
            demojized_string = " ".join(demojized)
            text = text.replace(x, demojized_string)
    text = re.sub(r'[@#$%^&*()\-<>+=?/`~!;:><]', ' ', text) # Remove
specific special characters
    text = re.sub(r'\bhttp\S+|www\S+', '', text) # Remove sentences
starting with "http" or "www"
    text = re.sub(r'[^\w\s]', ' ', text) # Replace all punctuations with a
whitespace
    text = re.sub(r'\s+', ' ', text) # Convert consecutive whitespaces
into " "
    return text
```

### Output



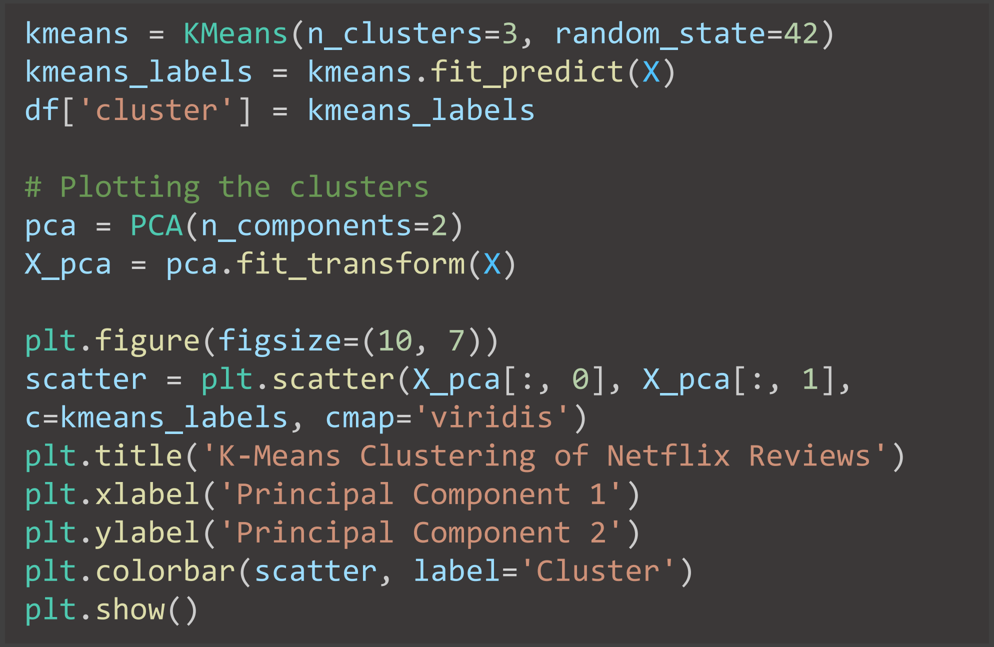
| | reviewId | content | score | thumbsUpCount | at | sentiment |
|---|---|---|---|---|---|---|
| 0 | 7b2a264c-7bb5-4729-b3d2-2168f8a7855e | i pay 18 month for an app that s super glitchy... | 2 | 1 | 2024-07-10 15:20:28 | 0 |
| 1 | 10faea27-b33d-40bb-b669-cf126438d525 | netflix plzz this kdrama dubbed in hindi happi... | 5 | 2 | 2024-07-10 15:03:37 | 1 |
| 2 | 61a10e0d-e868-4d87-aa30-f41d30285a3f | terrible app i can t watch anything because of... | 1 | 0 | 2024-07-08 15:41:17 | 0 |
| 3 | 1a7ce341-afc6-46da-9d08-793582e8ed3c | i love two hearts two hearts to download it lo... | 5 | 0 | 2024-07-07 17:47:19 | 1 |
| 4 | 1bd445c3-7f36-4717-810a-63c5533207d0 | exceptional | 5 | 1 | 2024-07-07 12:31:53 | 1 |

# Feature Engineering & Extraction

## Feature Engineering

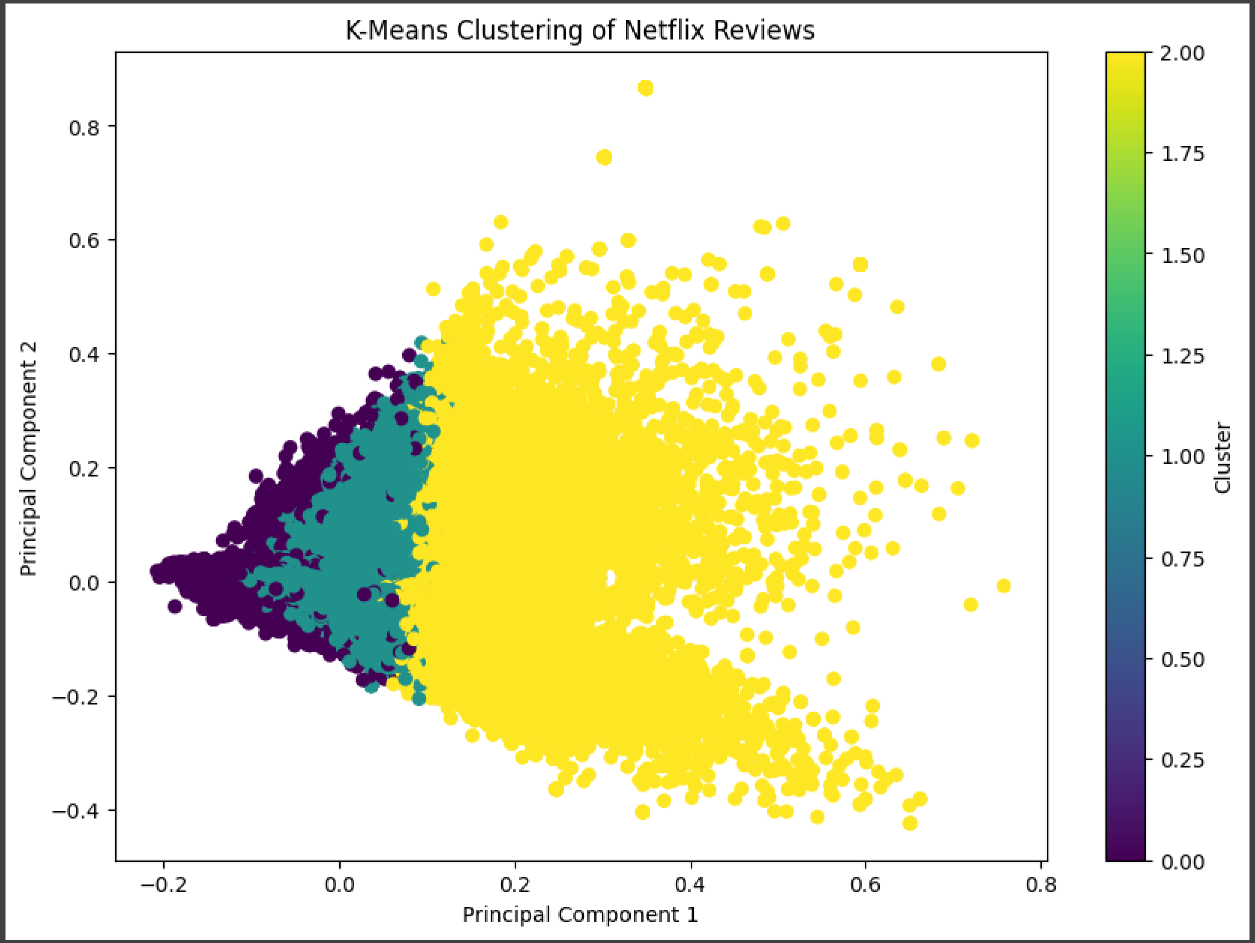
## Clustering with K-Means and Dimensionality Reduction using Principal Component Analysis (PCA)

### Function Code:



```
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans_labels = kmeans.fit_predict(X)
df['cluster'] = kmeans_labels

# Plotting the clusters
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)

plt.figure(figsize=(10, 7))
scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1],
c=kmeans_labels, cmap='viridis')
plt.title('K-Means Clustering of Netflix Reviews')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.colorbar(scatter, label='Cluster')
plt.show()
```

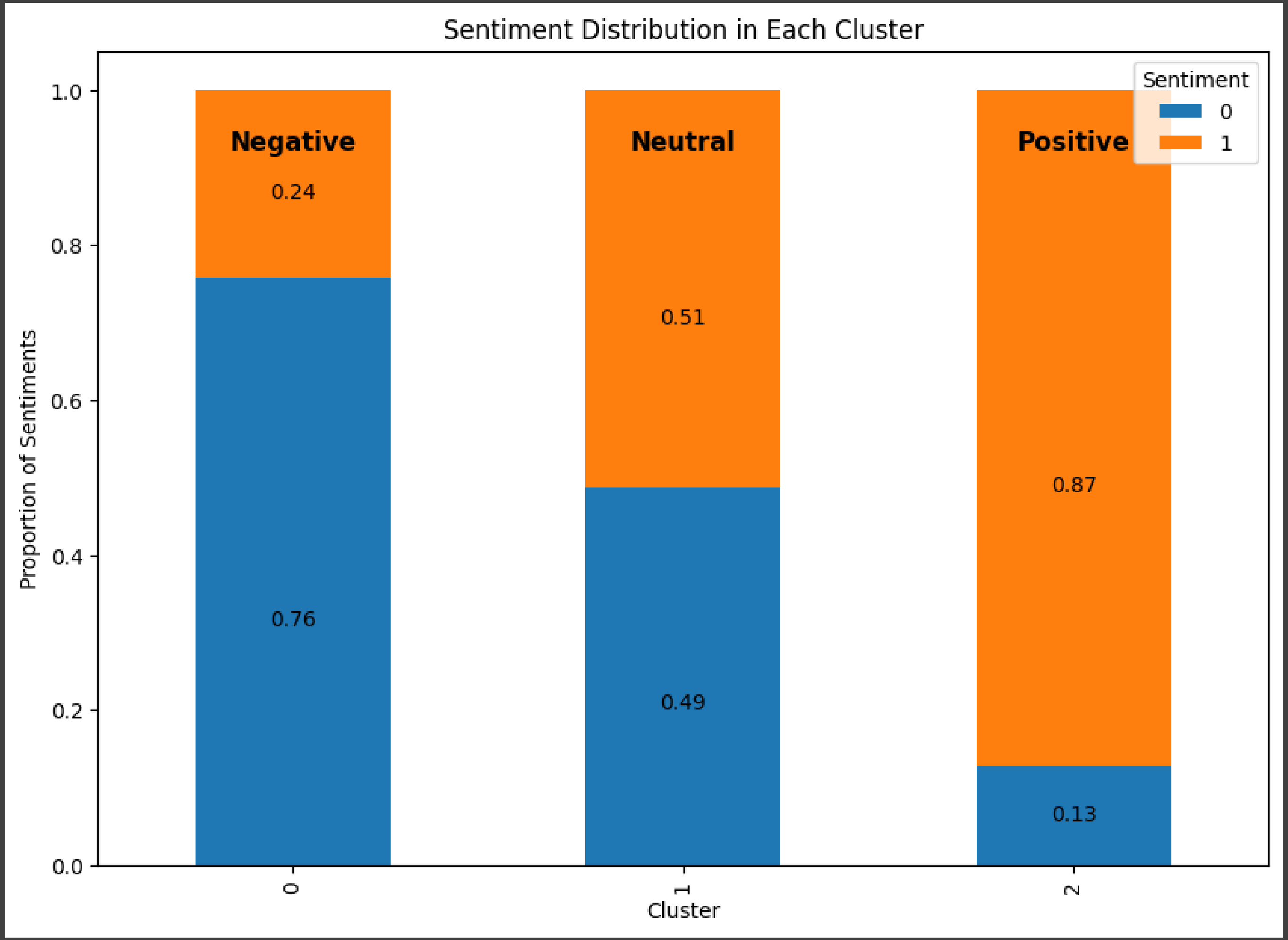
### Output



# Feature Engineering & Extraction

### Feature Engineering

### Visualize the sentiment distribution in each cluster using bar plots



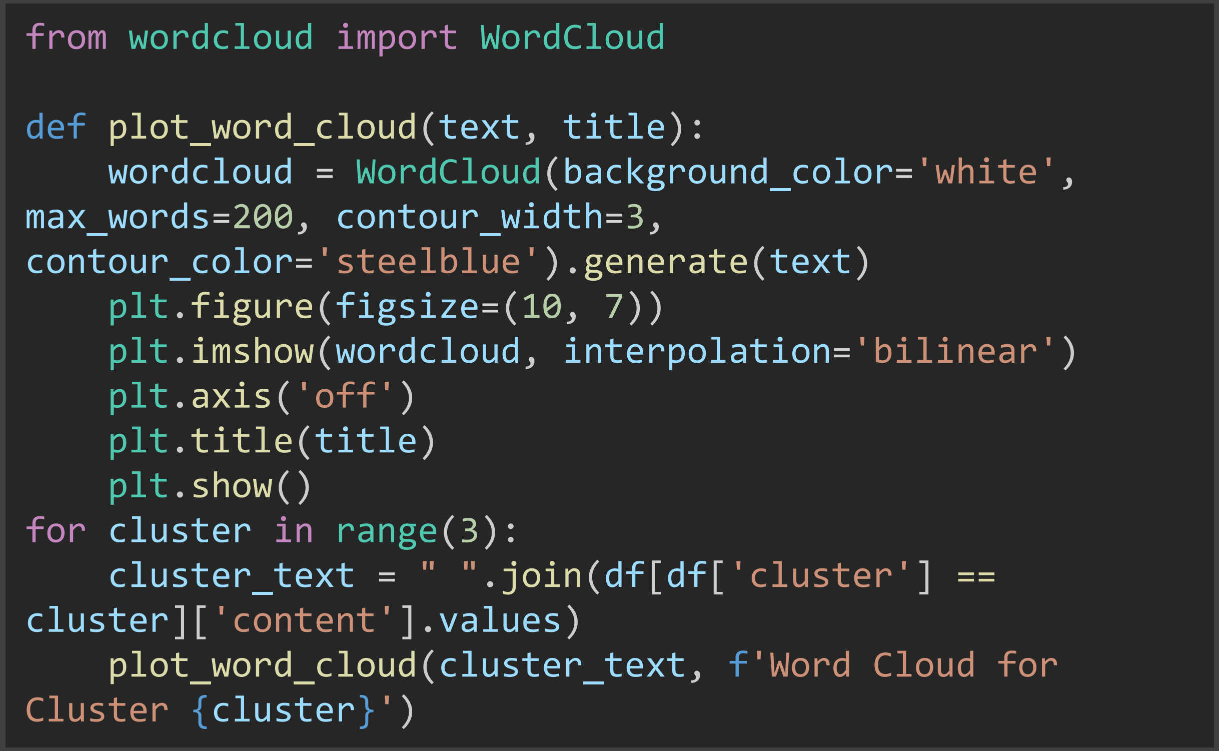
We want to divide clusters into 3 review categories: negative, neutral, and positive. Clusters that contain more sentiment values of 0 (negative) are labeled as negative review categories, while clusters that contain more sentiment values of 1 (positive) are labeled as positive review categories. Similarly, the neutral review category is assigned to clusters with a relatively balanced ratio.

# Feature Engineering & Extraction

## Feature Engineering

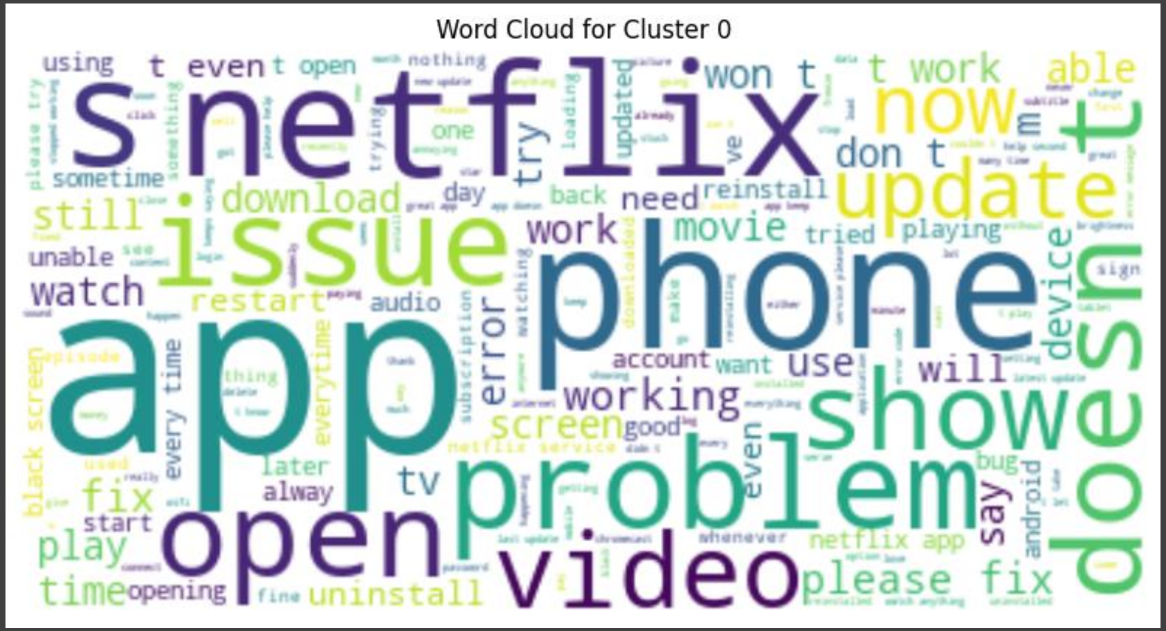
### Analysis using WordCloud

**Function Code**



```
from wordcloud import WordCloud

def plot_word_cloud(text, title):
    wordcloud = WordCloud(background_color='white',
max_words=200, contour_width=3,
contour_color='steelblue').generate(text)
    plt.figure(figsize=(10, 7))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(title)
    plt.show()
for cluster in range(3):
    cluster_text = " ".join(df[df['cluster'] ==
cluster]['content'].values)
    plot_word_cloud(cluster_text, f'Word Cloud for
Cluster {cluster}')
```

**Output**

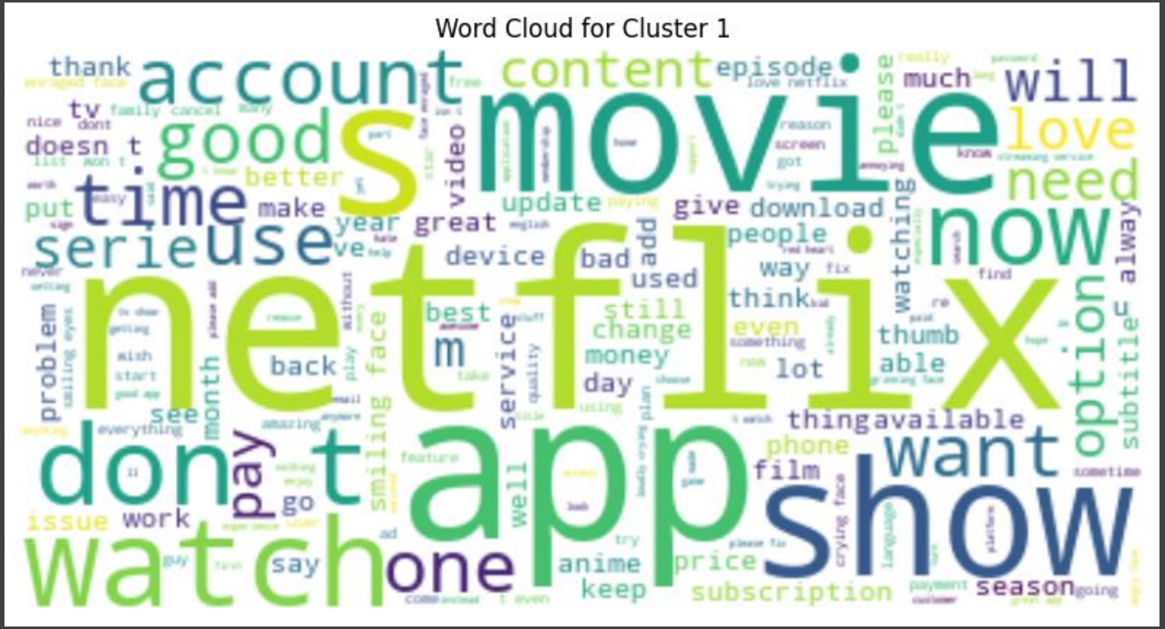


Word Cloud for Cluster 0

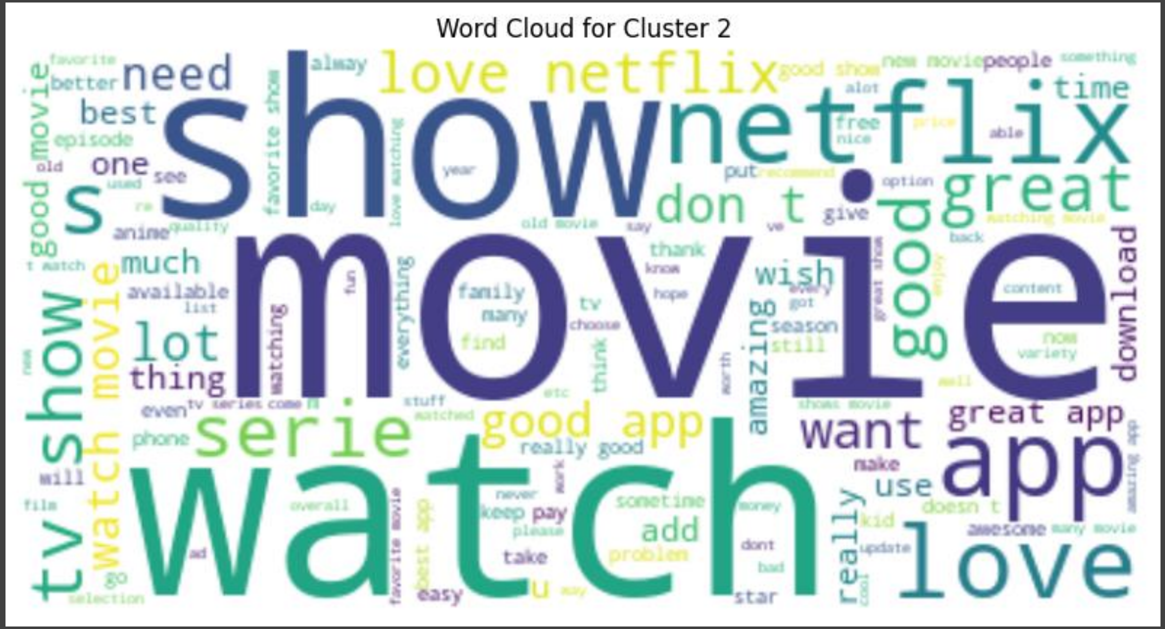
# Feature Engineering & Extraction

## Feature Engineering

### Analysis using WordCloud



Word Cloud for Cluster 1



Word Cloud for Cluster 2

From the word cloud output, we can see which words are most frequently contained in the reviews based on their size. The larger the word, the more often it appears in the reviews. For example, the word 'problem' in negative reviews in cluster 0.

# Modeling using Naïve Bayes

### Import Necessary Libraries

```
import pandas as pd
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, classification_report
```

### Split Data

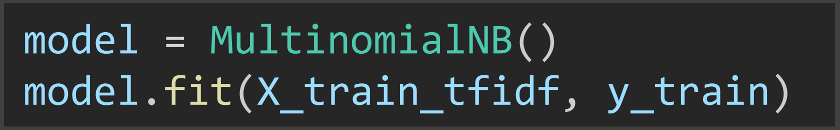
```
X = df['content']
y = df['sentiment']

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

### TF – IDF Vectorization

```
tfidf = TfidfVectorizer(stop_words='english', max_features=10000)
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)
```

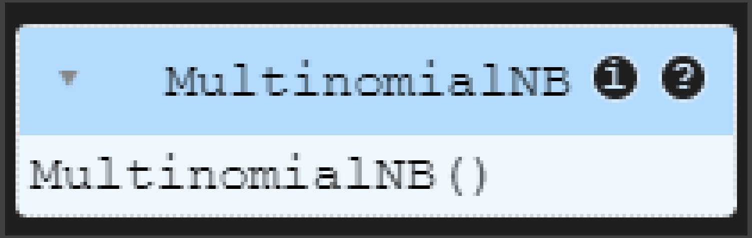
# Modeling using Naive Bayes

### Create Model using Naive Bayes



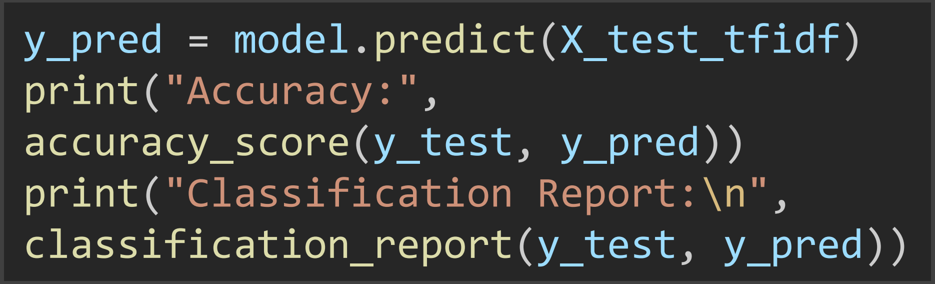
```
model = MultinomialNB()
model.fit(X_train_tfidf, y_train)
```

### Output



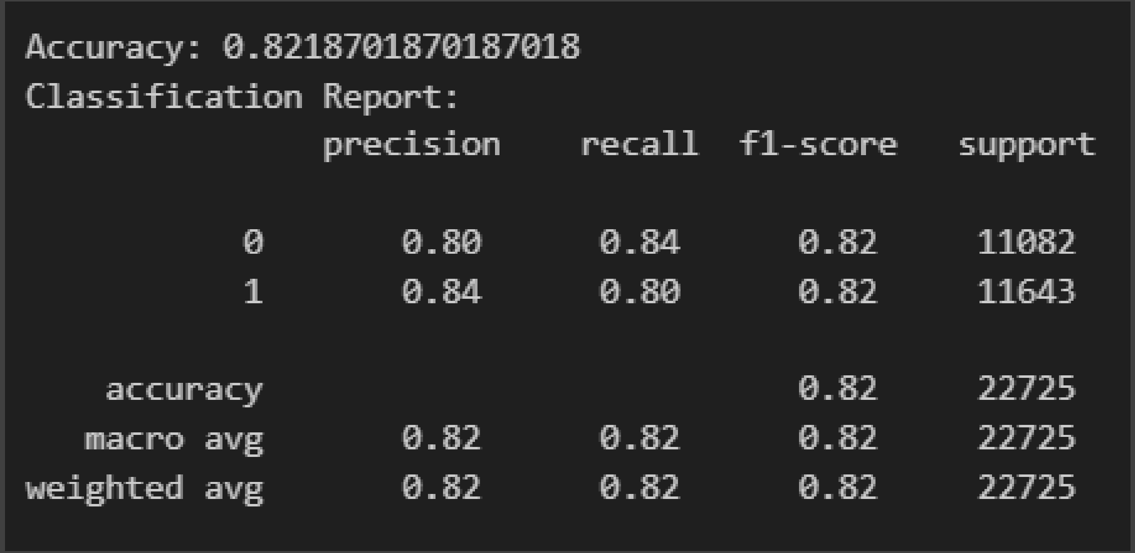
```
MultinomialNB()
```

### Model Evaluation



```
y_pred = model.predict(X_test_tfidf)
print("Accuracy:",
accuracy_score(y_test, y_pred))
print("Classification Report:\n",
classification_report(y_test, y_pred))
```

### Output



```
Accuracy: 0.8218701870187018
Classification Report:
               precision    recall  f1-score   support

           0       0.80      0.84      0.82     11082
           1       0.84      0.80      0.82     11643

    accuracy                           0.82     22725
   macro avg       0.82      0.82      0.82     22725
weighted avg       0.82      0.82      0.82     22725
```

**Accuracy : 0.82187**

**Great!!!**

# Evaluation

## Model Evaluation

**Accuracy is 82.19%**, which indicates that the model can correctly classify review sentiments with an overall accuracy of 82.19%.

## Classification Report

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| 0 | 0.80 | 0.84 | 0.82 | 11082 |
| 1 | 0.84 | 0.80 | 0.82 | 11643 |

**Precision**

The percentage of positive predictions that are actually positive.

- Class 0 (Negative): 80%
- Class 1 (Positive): 84%

A higher precision value for class 1 indicates that the model produces fewer false positives for positive reviews compared to negative reviews.

# Evaluation

**Recall**

The percentage of all positive instances that were correctly predicted.

- Class 0 (Negative): 84%
- Class 1 (Positive): 80%

A higher recall value for class 0 indicates that the model produces fewer false negatives for negative reviews compared to positive reviews.

**F1-Score**

The harmonic mean of precision and recall, providing a balanced measure of both.

- Class 0 (Negative): 0.82
- Class 1 (Positive): 0.82

Equal F1-Score values indicate that the model has a good balance between precision and recall for both classes.

**Support**

The number of actual instances for each class.

- Class 0 (Negative): 11,082
- Class 1 (Positive): 11,643

# Conclusion

**Key Conclusions:**

- **Model Performance:**

The sentiment analysis model built demonstrates good performance with an accuracy of 82.19%. The balanced precision, recall, and F1-score indicate that the model is capable of classifying positive and negative reviews well.

- **Balanced Precision and Recall:**

The model shows a good balance between precision and recall for both classes, which is important in sentiment analysis where both false positives and false negatives can be equally detrimental.

- **Data Distribution:**

The support shows that the dataset has a fairly balanced distribution between positive and negative reviews, which helps in building a model that is not biased towards one class.

With these results, the model can be relied upon for sentiment analysis in the Netflix review dataset, but there may still be room for improvement with further optimization or by using more advanced techniques.

# Github

https://github.com/wilywho/Portfolio