SENTIMENT ANALYSIS

GROUP 6 PRESENTATION:

PRESENTED BY:

Kumar Shivam - n01592878
Rudransh Sharma - n01544993
Anish Nepal - n01571685
Harshal Dhuria - n01526164

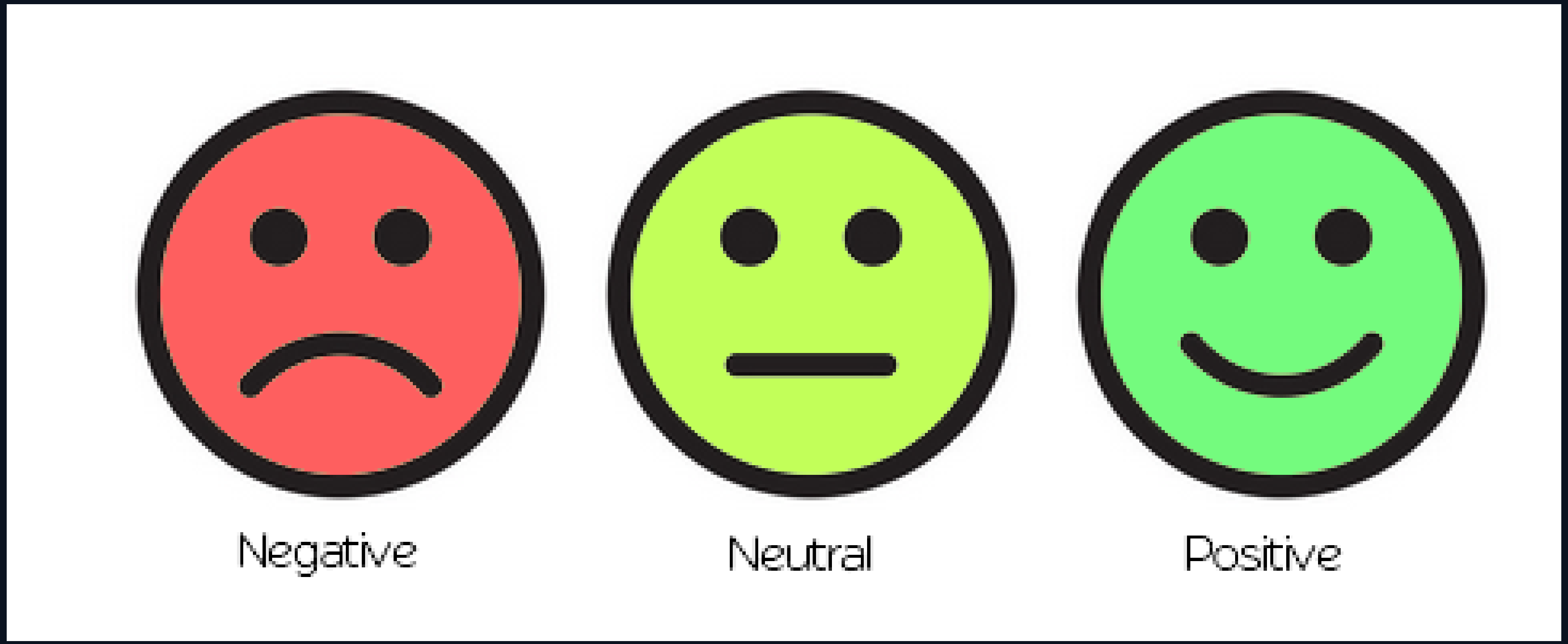
PROBLEM STATEMENT

A newly opened restaurant in Las Vegas is looking to enhance its fine dining experience based on its customer reviews.

Opened	Investment	Total Customer Reviews
2023	**$200K**	**2400**

GOAL

The main goal of this project is to perform sentiment analysis, calculate polarity scores for the customer reviews, and uncover the underlying sentiment in each piece of content.



LIBRARIES USED:

MODEL SELECTION

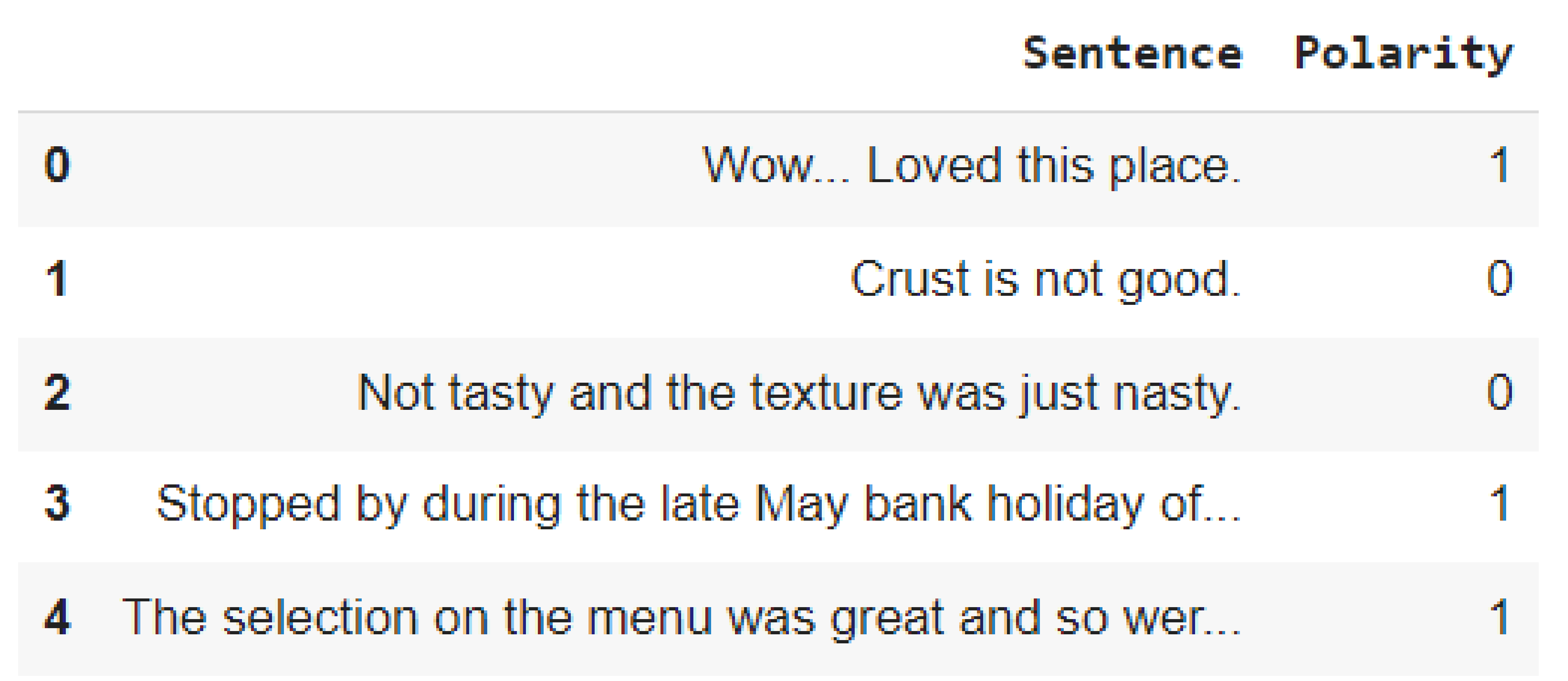
We have used two models to do the Sentiment Analysis:

1. VADER (Valence Aware Dictionary and Sentiment Reasoner)
- a rule-based sentiment analyzer to find if a review is positive or negative

2. LSTM - (Long Short Term Memory)
- helps computers understand and remember information for a longer time, specially designed to work on long texts

Comparitiverly, VADER is handy for quick assessments, it may not catch the nuances or deeper meanings in language unlike Long Short-Term Memory (LSTM).

DATASET DESCRIPTION



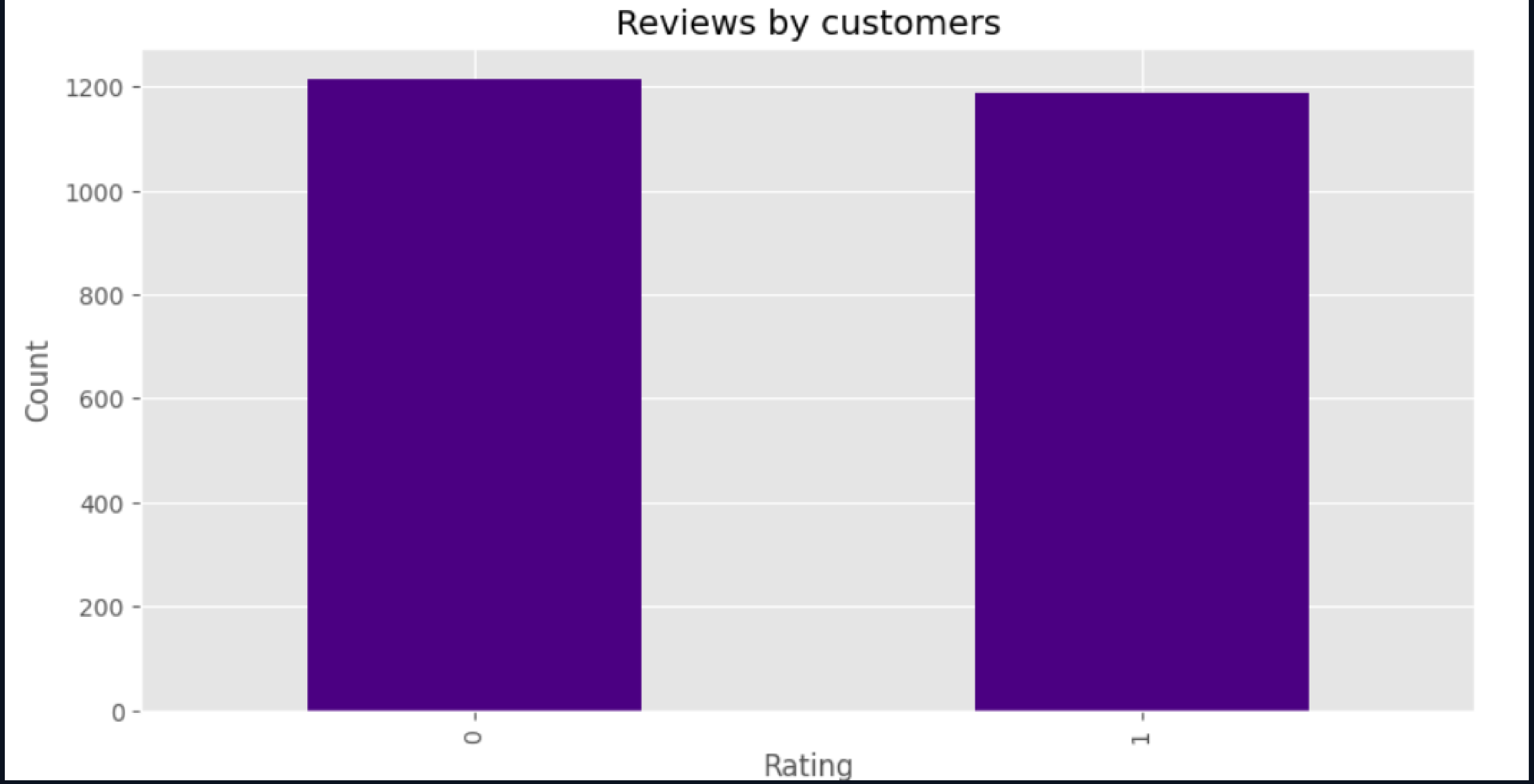
	Sentence	Polarity
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer...	1

Two Features:

1. Sentence - Actual Customer Reviews.
2. Polarity - Classified as 0 and 1 based on the reviews given.

There are no NULL values in the dataset meaning quality of the dataset is good.

DATA EXPLORATION

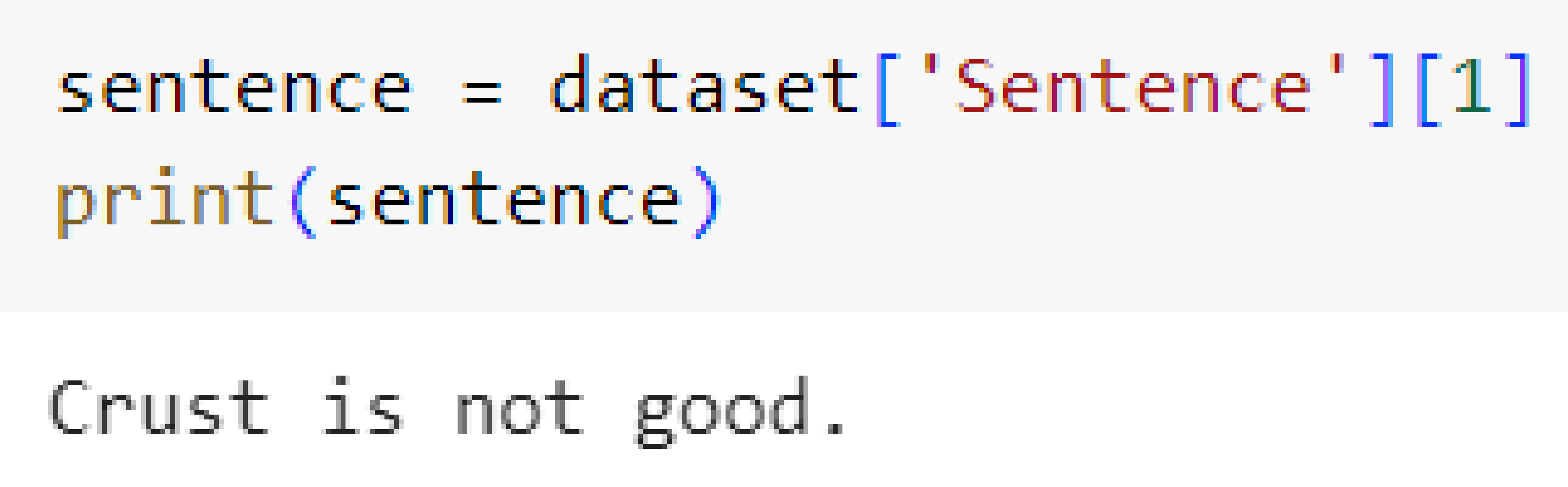


Analyzing Reviews of Customers:
Here 0 means ‘Negative Review’ and 1 means ‘Positive Review’

DATA PREPROCESSING

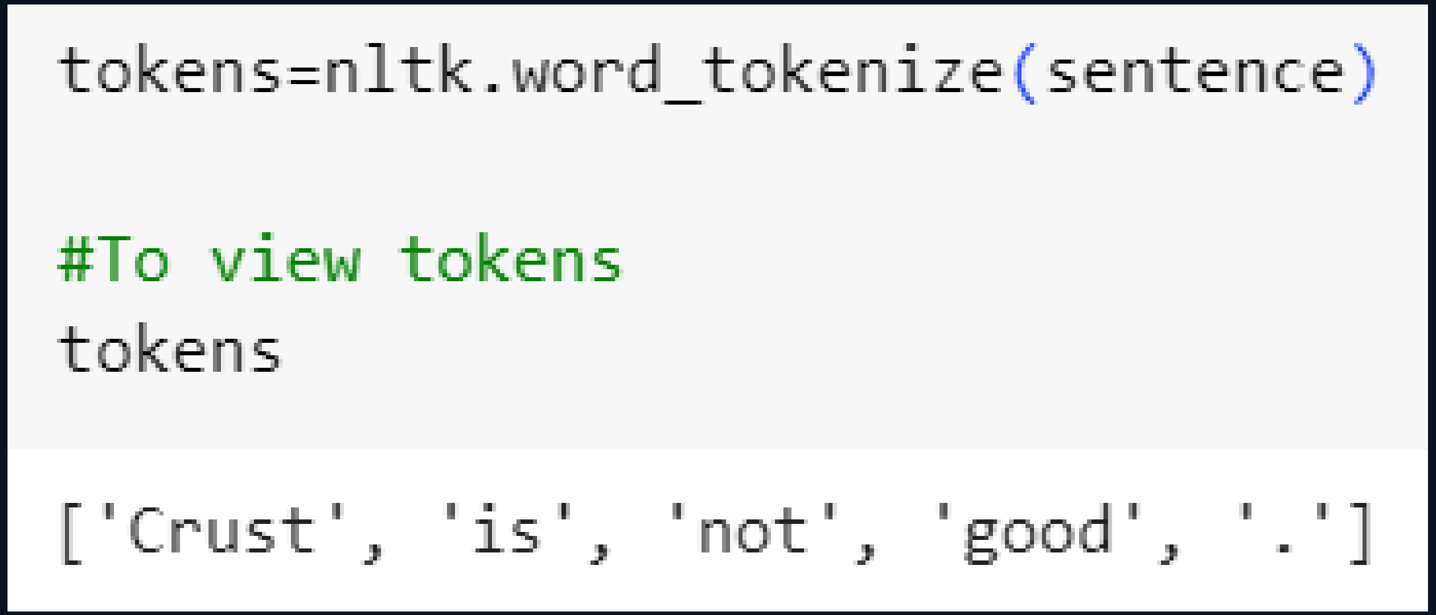
STEP-1: Tokenization

- Initial stage in NLTK text analytics
- Process of breaking down a paragraph into simple components.
- Easier for computers to process and analyze languages.



```
sentence = dataset['Sentence'][1]
print(sentence)
```

```
Crust is not good.
```



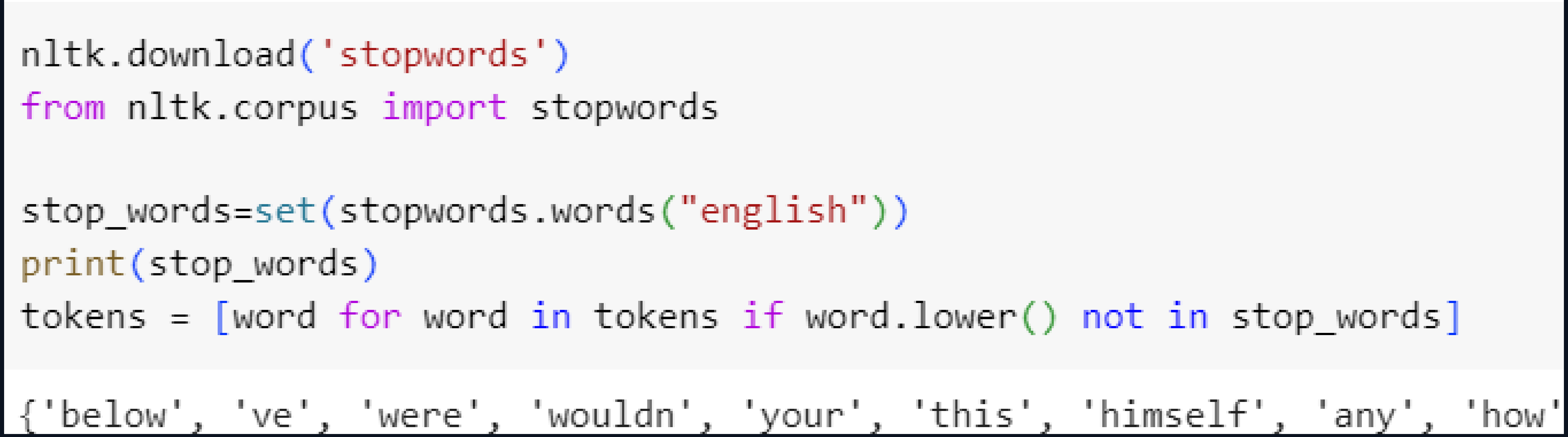
```
tokens=nltk.word_tokenize(sentence)

#To view tokens
tokens
```

```
['Crust', 'is', 'not', 'good', '.']
```

DATA PREPROCESSING

STEP-2: Stop Word

- Textual noise
- Text may include stopping words like “is,” “am,” “are,” “this,” “a,” “an,” “the,” etc.
- Built a list of stop words and filtered out from the tokens list.



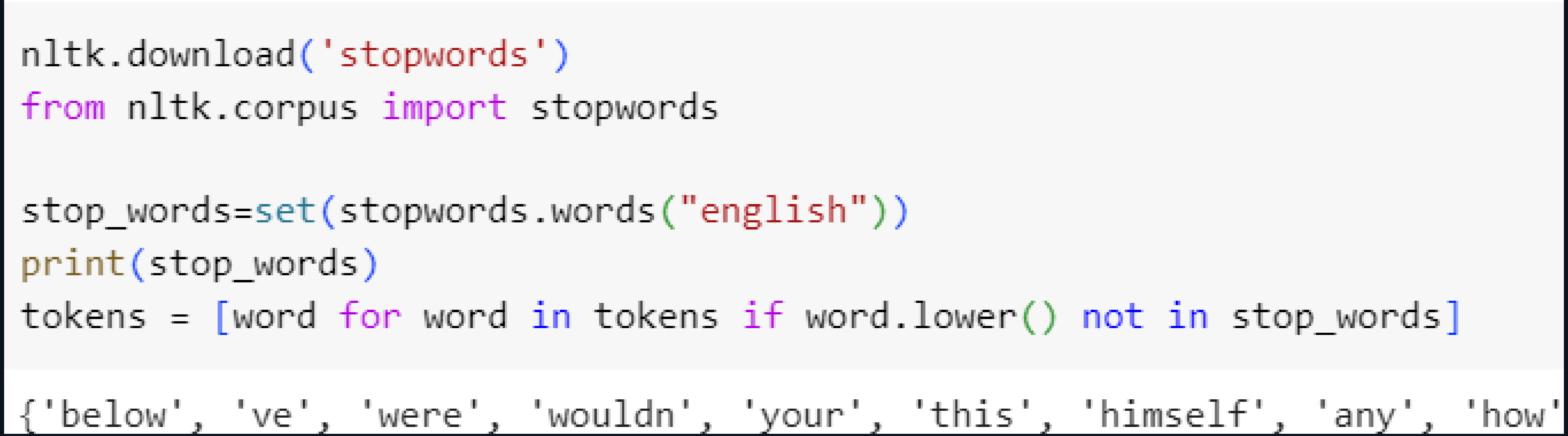
```
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words=set(stopwords.words("english"))
print(stop_words)
tokens = [word for word in tokens if word.lower() not in stop_words]
```

```
{'below', 've', 'were', 'wouldn', 'your', 'this', 'himself', 'any', 'how'
```

DATA PREPROCESSING

STEP-2: Stop Word

- Textual noise
- Text may include stopping words like “is,” “am,” “are,” “this,” “a,” “an,” “the,” etc.
- Built a list of stop words and filtered out from the tokens list.



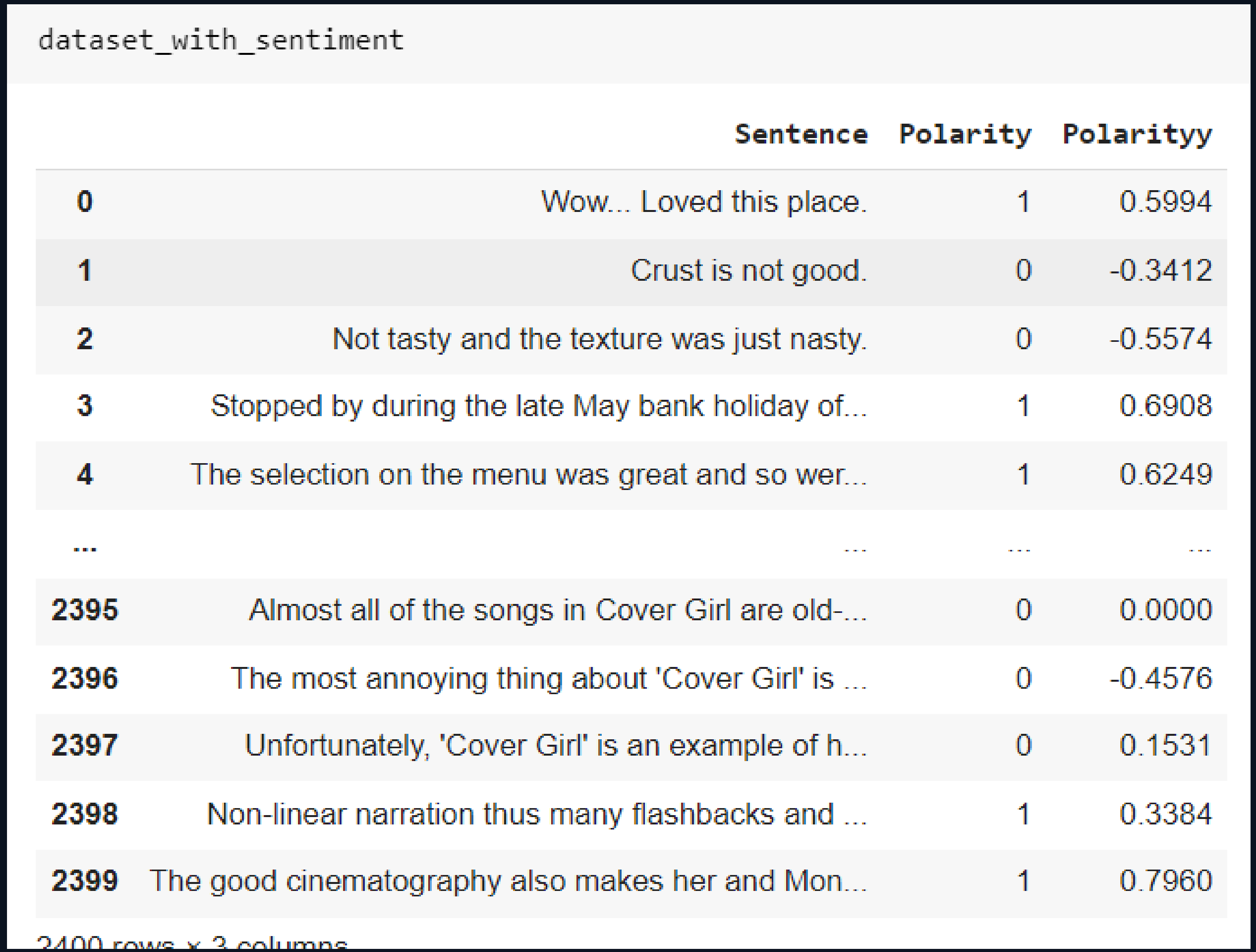
```
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words=set(stopwords.words("english"))
print(stop_words)
tokens = [word for word in tokens if word.lower() not in stop_words]
```

```
{'below', 've', 'were', 'wouldn', 'your', 'this', 'himself', 'any', 'how'
```

SENTIMENT ANALYSIS - VADER

- A tool that reads through sentences, assigns a score to express how positive, negative, or neutral the text is.
- It's like a quick judge for understanding emotions in text.
- It calculates a compound polarity score ('Polarity') for each sentence:
  positive sentiment : (compound score >= 0.05)
  neutral sentiment : (compound score > -0.05) and (compound score < 0.05)
  negative sentiment : (compound score <= -0.05)

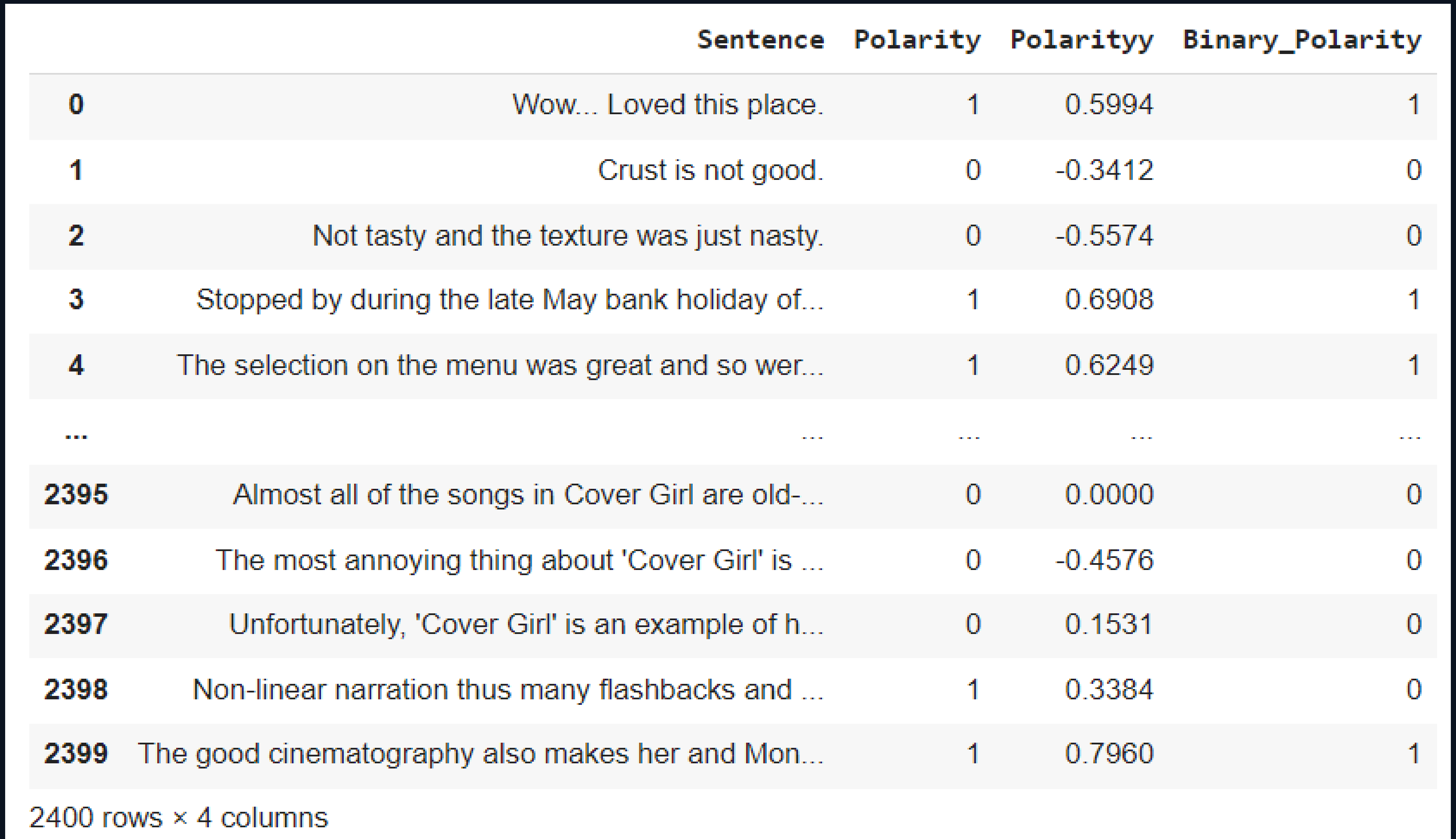
SENTIMENT ANALYSIS - VADER



```
dataset_with_sentiment
```

	Sentence	Polarity	Polarityy
0	Wow... Loved this place.	1	0.5994
1	Crust is not good.	0	-0.3412
2	Not tasty and the texture was just nasty.	0	-0.5574
3	Stopped by during the late May bank holiday of...	1	0.6908
4	The selection on the menu was great and so wer...	1	0.6249
...	...	...	...
2395	Almost all of the songs in Cover Girl are old-...	0	0.0000
2396	The most annoying thing about 'Cover Girl' is ...	0	-0.4576
2397	Unfortunately, 'Cover Girl' is an example of h...	0	0.1531
2398	Non-linear narration thus many flashbacks and ...	1	0.3384
2399	The good cinematography also makes her and Mon...	1	0.7960

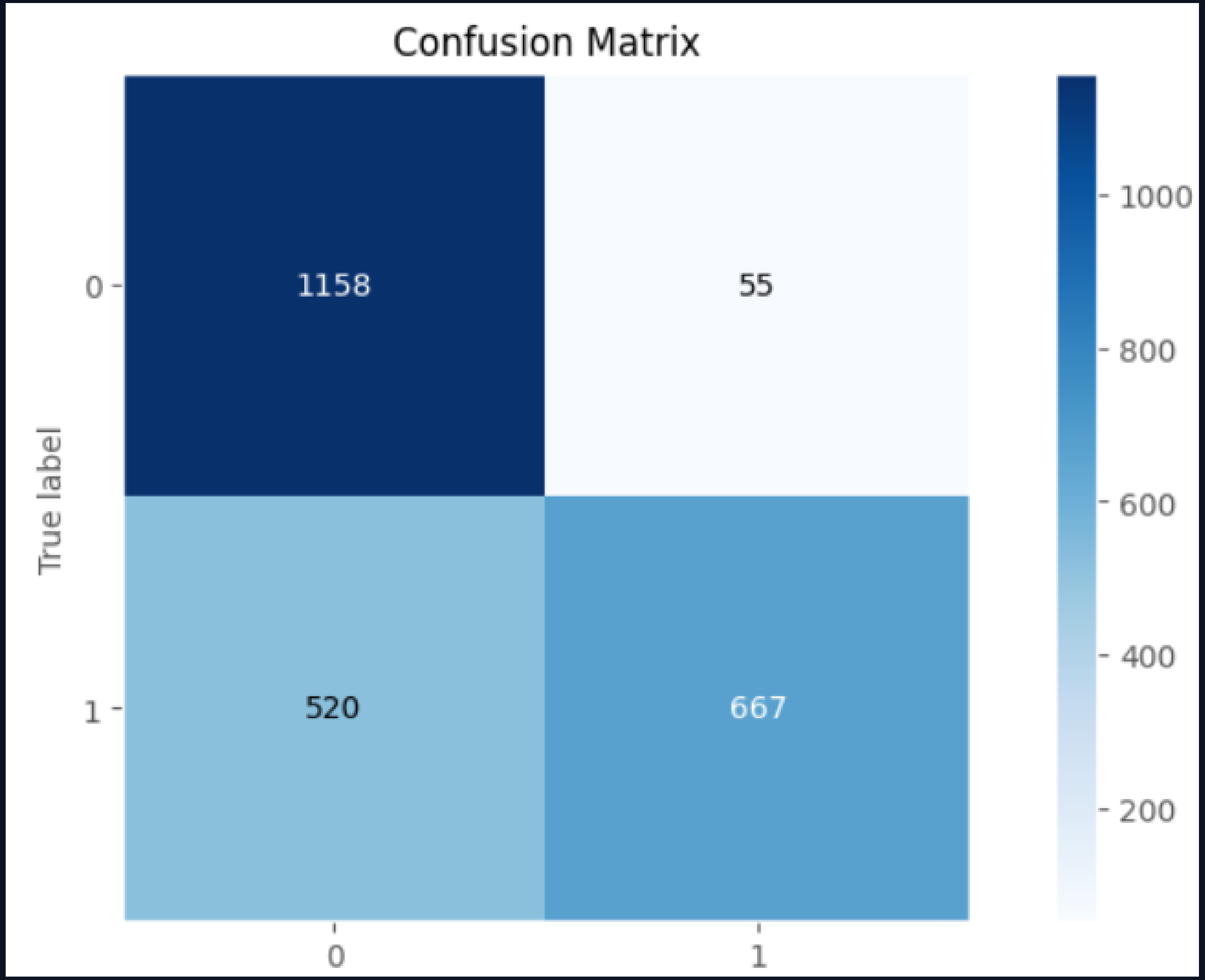
2400 rows × 3 columns

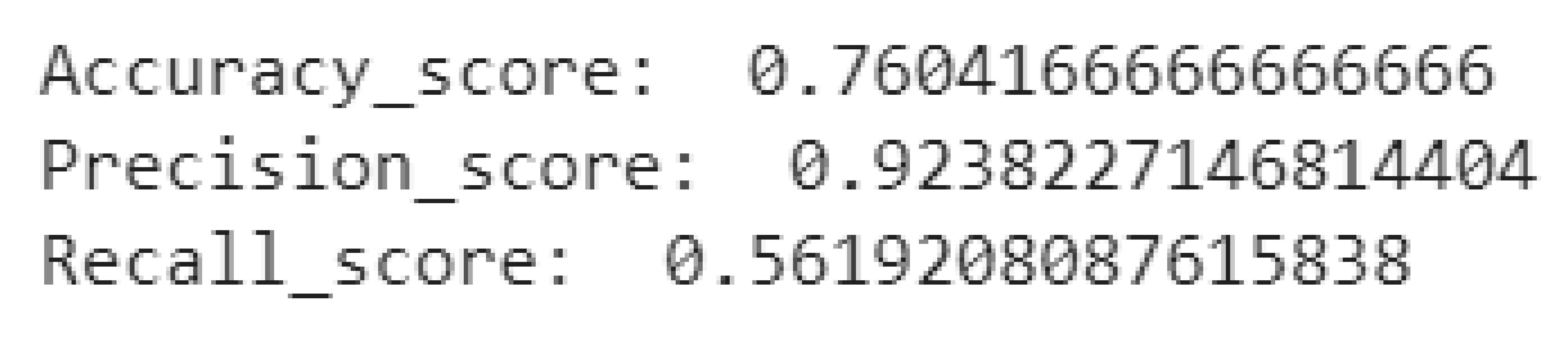


	Sentence	Polarity	Polarityy	Binary_Polarity
0	Wow... Loved this place.	1	0.5994	1
1	Crust is not good.	0	-0.3412	0
2	Not tasty and the texture was just nasty.	0	-0.5574	0
3	Stopped by during the late May bank holiday of...	1	0.6908	1
4	The selection on the menu was great and so wer...	1	0.6249	1
...	...	...	...	...
2395	Almost all of the songs in Cover Girl are old-...	0	0.0000	0
2396	The most annoying thing about 'Cover Girl' is ...	0	-0.4576	0
2397	Unfortunately, 'Cover Girl' is an example of h...	0	0.1531	0
2398	Non-linear narration thus many flashbacks and ...	1	0.3384	0
2399	The good cinematography also makes her and Mon...	1	0.7960	1

2400 rows × 4 columns

ACCURACY - VADER



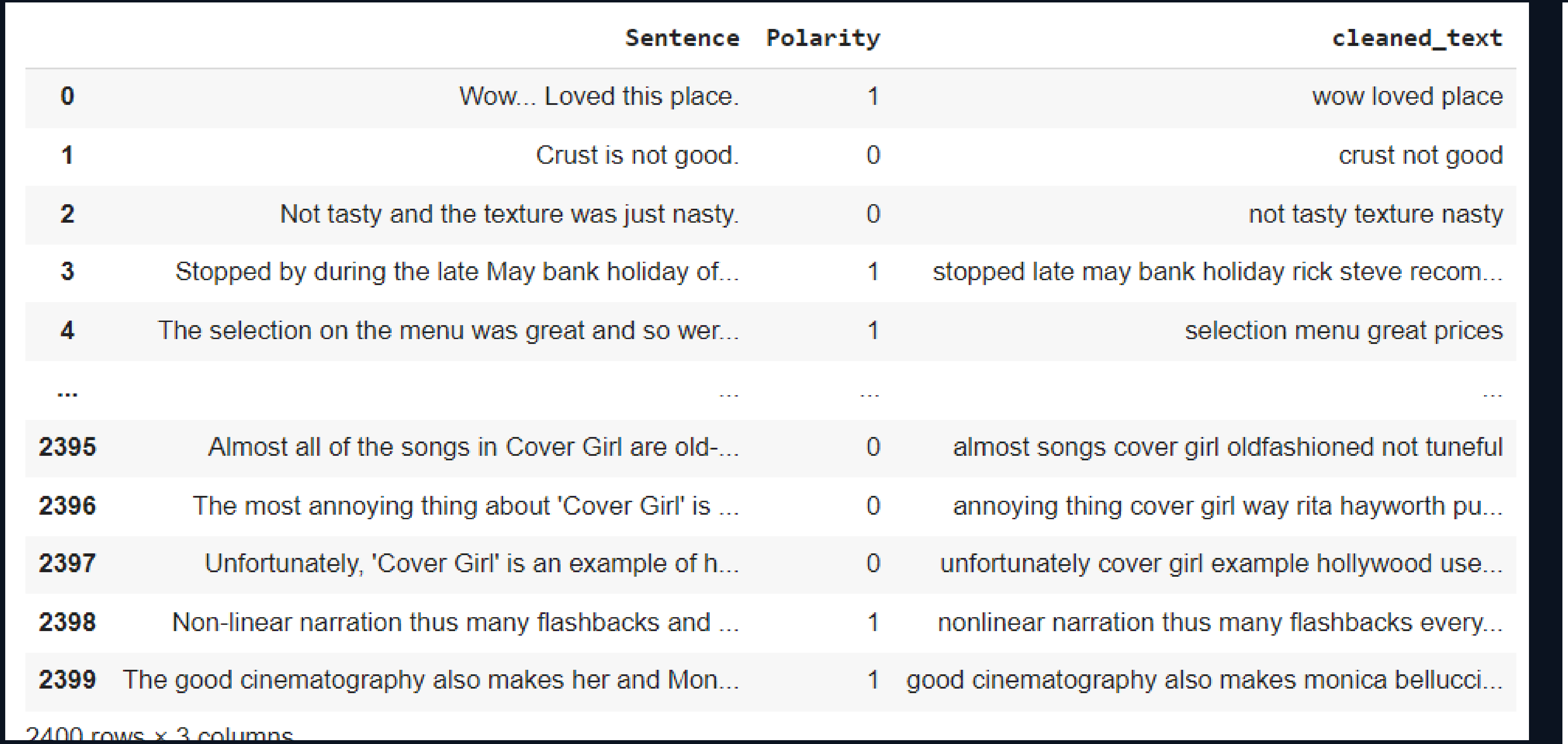


```
Accuracy_score:  0.7604166666666666
Precision_score:  0.9238227146814404
Recall_score:  0.5619208087615838
```

DATA PREPROCESSING - LSTM

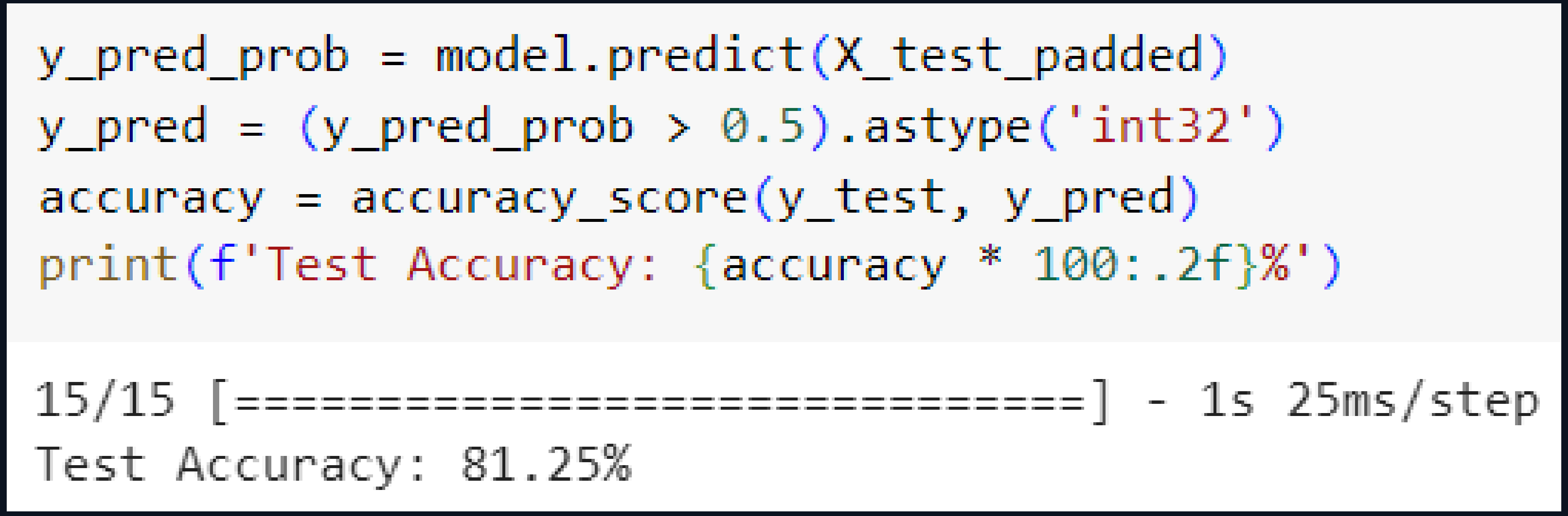
1. Training a sentiment analysis model using a neural network (LSTM) with Keras.
2. For text processing: "re" for regular expressions, "word_tokenize" for breaking text into words, and "stopwords" for excluding non-contributing words.
3. creating a custom text preprocessing function called preprocess_text1 that converts text to lowercase, removes special characters and numbers, tokenizes the text, and excludes common English stopwords with the exception of custom stop words like 'not' and 'never'.
4. splitting the data into training and testing sets for machine learning: **X** contains cleaned text, **y** contains sentiment labels, and 80% is used for training (**X_train** and **y_train**) while 20% is reserved for testing (**X_test** and **y_test**).
5. using the Bag of Words approach for feature extraction, converting the cleaned text into a feature vector using CountVectorizer. The data is then split into training and testing sets for model training and evaluation.
6. preparing the text data for input into a neural network model. It uses a Tokenizer to convert the text into sequences of numerical values and ensures that all sequences have the same length by padding or truncating them to a maximum length of 110. This step is crucial for training a neural network on text data.
7. defining a neural network model using Keras with an embedding layer for representing words, an LSTM layer for sequence processing, a dropout layer to prevent overfitting, and a final dense layer with a sigmoid activation for binary sentiment classification.
8. It runs for 6 epochs, dividing the data into batches of size 80 and validating the model's performance on a 10% subset of the training data.
9. code predicts sentiment labels for the test data using the trained neural network model, compares the predictions to the actual labels, and calculates the accuracy of the model on the test set, displaying the result as a percentage.

MODEL EVALUATION - LSTM



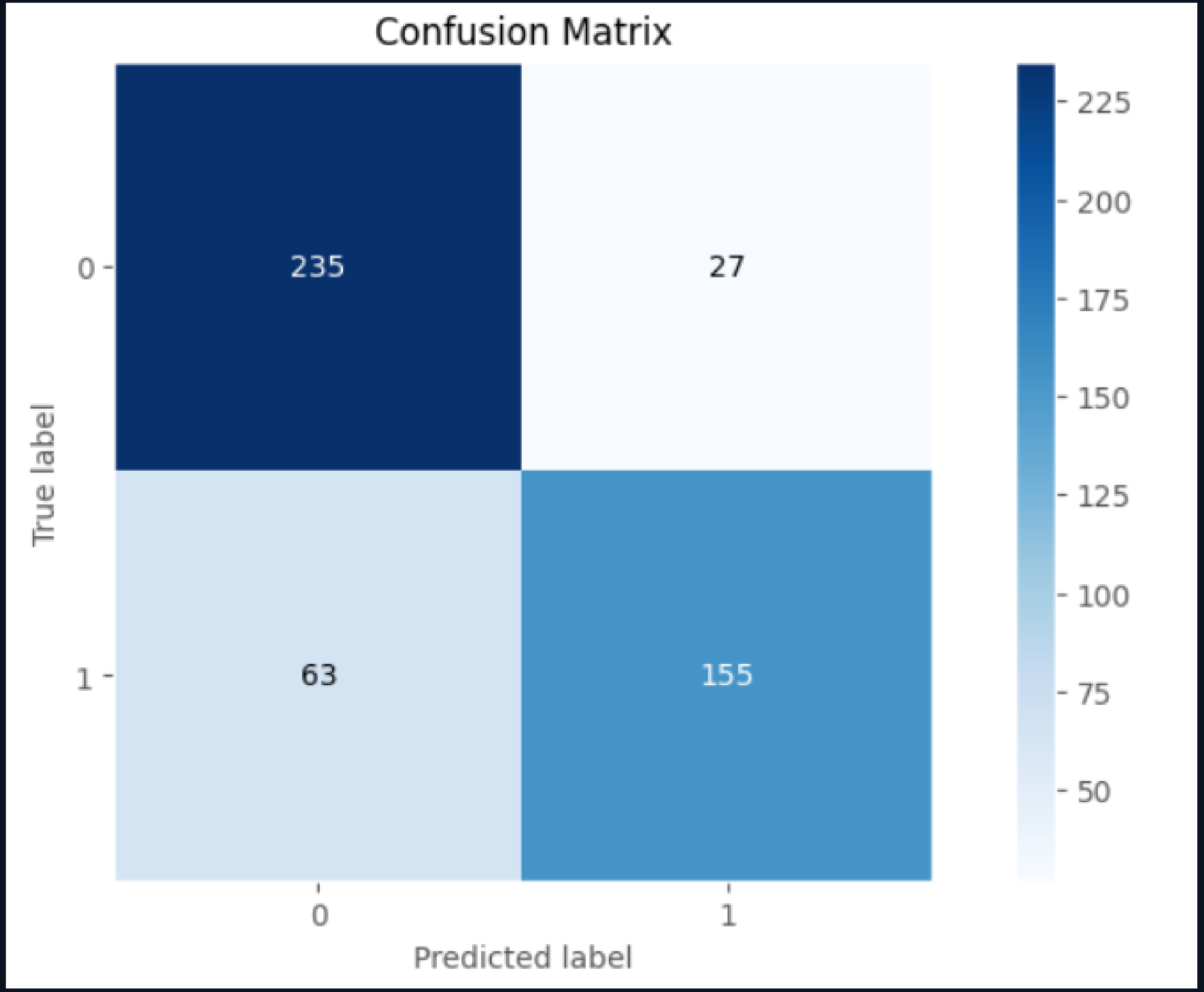
	Sentence	Polarity	cleaned_text
0	Wow... Loved this place.	1	wow loved place
1	Crust is not good.	0	crust not good
2	Not tasty and the texture was just nasty.	0	not tasty texture nasty
3	Stopped by during the late May bank holiday of...	1	stopped late may bank holiday rick steve recom...
4	The selection on the menu was great and so wer...	1	selection menu great prices
...	...	...	...
2395	Almost all of the songs in Cover Girl are old-...	0	almost songs cover girl oldfashioned not tuneful
2396	The most annoying thing about 'Cover Girl' is ...	0	annoying thing cover girl way rita hayworth pu...
2397	Unfortunately, 'Cover Girl' is an example of h...	0	unfortunately cover girl example hollywood use...
2398	Non-linear narration thus many flashbacks and ...	1	nonlinear narration thus many flashbacks every...
2399	The good cinematography also makes her and Mon...	1	good cinematography also makes monica bellucci...

2400 rows × 3 columns



```
y_pred_prob = model.predict(X_test_padded)
y_pred = (y_pred_prob > 0.5).astype('int32')
accuracy = accuracy_score(y_test, y_pred)
print(f'Test Accuracy: {accuracy * 100:.2f}%')
```

```
15/15 [==============================] - 1s 25ms/step
Test Accuracy: 81.25%
```



SHORTCOMINGS

- Polarity & subjectivy analysis
- Stop wording challenges
- Wrong predictions
- Hyperparameter tuning balancing
- Insufficient input data

CONCLUSION

- Our models carefully tuned hyper parameters providing valuable insights.
- Shows distribution of positive and negative sentiments.
- Since the model is trained in small dataset, it has limitations to work on large datasets.
- The project successfully underscores the significance of sentiment analysis in understanding customer perception.
- All insights provided will help business in making data-driven decisions for improvement.