Sentimental Analysis of Mme Bovary blog comments

ARIANA AYAVIRI

Madame Bovary" is a novel written by the French author Gustave Flaubert. The novel was first published in 1857 and is considered one of the greatest works of literature. The full title of the novel is "Madame Bovary: Mœurs de province" (Madame Bovary: Provincial Customs). The story is set in provincial France in the 19th century and follows the life of Emma Bovary, a young woman who marries a country doctor named Charles Bovary.

WE WILL ANALYSE REVIEWS FROM THE WEBSITE:

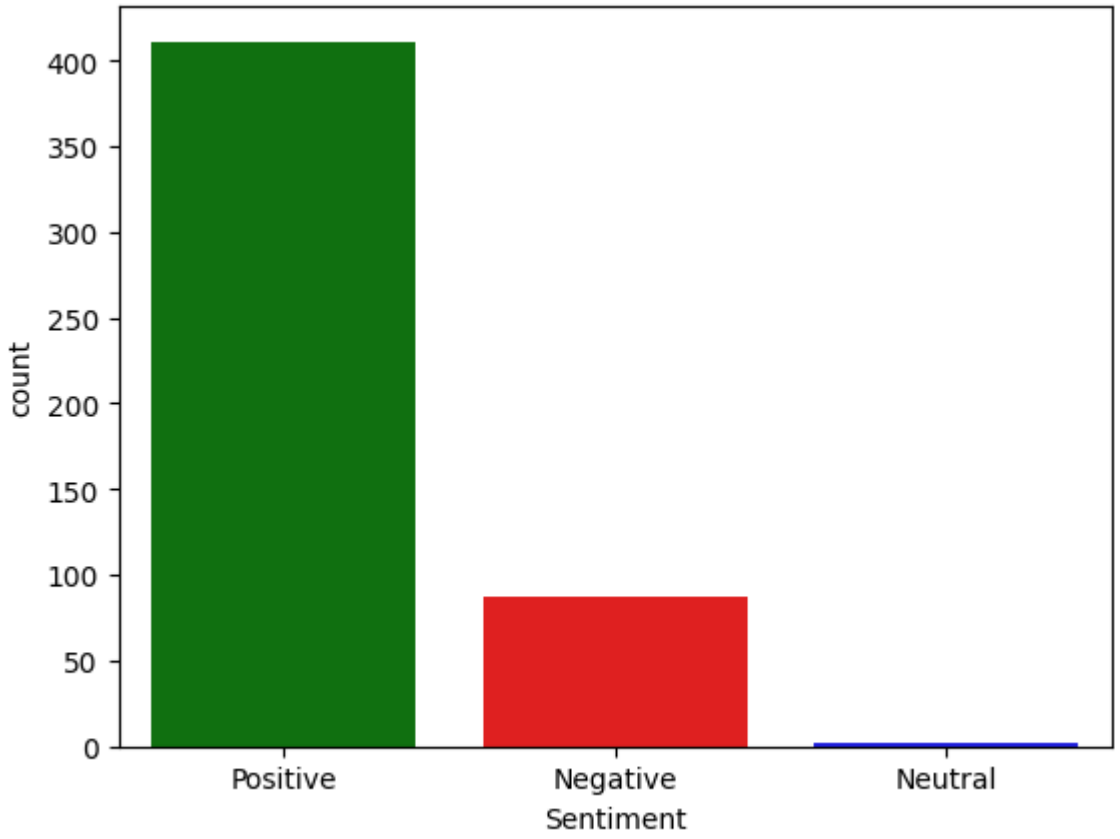
HTTPS://WWW.BABELIO.COM/LIVRES/FLAUBERT-MADAME-BOVARY/894329/CRITIQUES

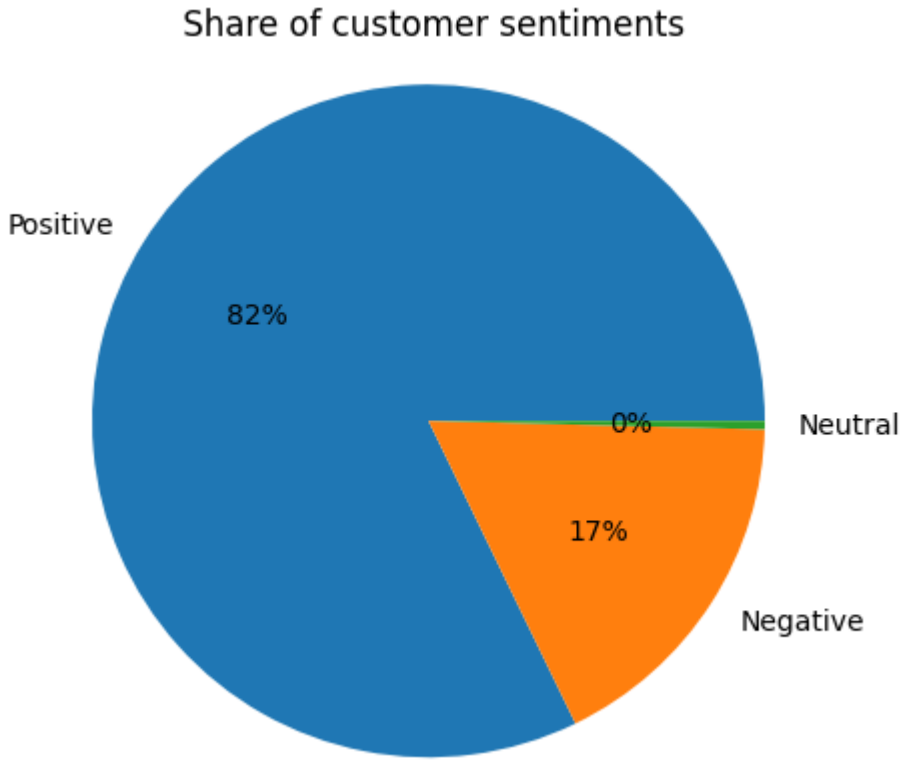
Blobber sentiment Classification analysis

```
# Sentiment analyser library
tb = Blobber(pos_tagger=PatternTagger(), analyzer=PatternAnalyzer())
```

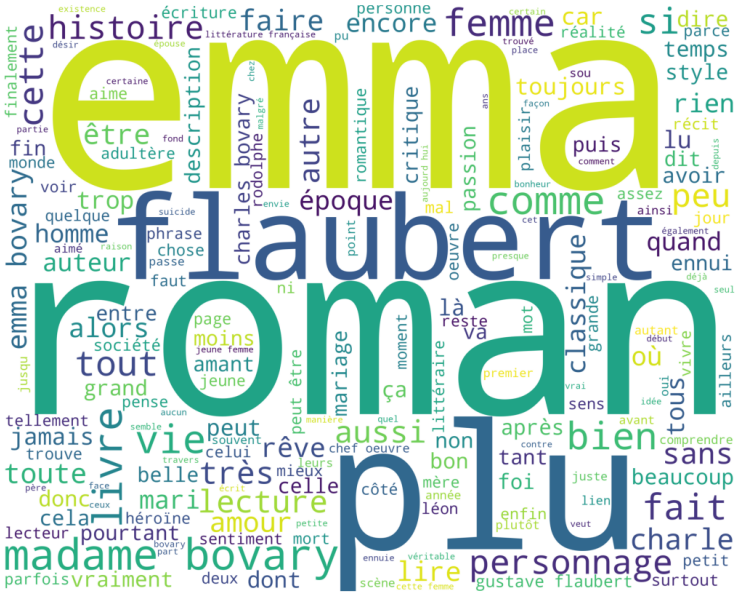
```
# Sentiment score
sentiment_score_polarity = []
for text in df["Clean_reviews_keywords"]:
    vs = tb(text).sentiment[0]
    sentiment_score_polarity.append(vs)

df['Sentiment_score'] = sentiment_score_polarity
df.head()
```

Negative, positive and neutral reviews from Mme Bovary



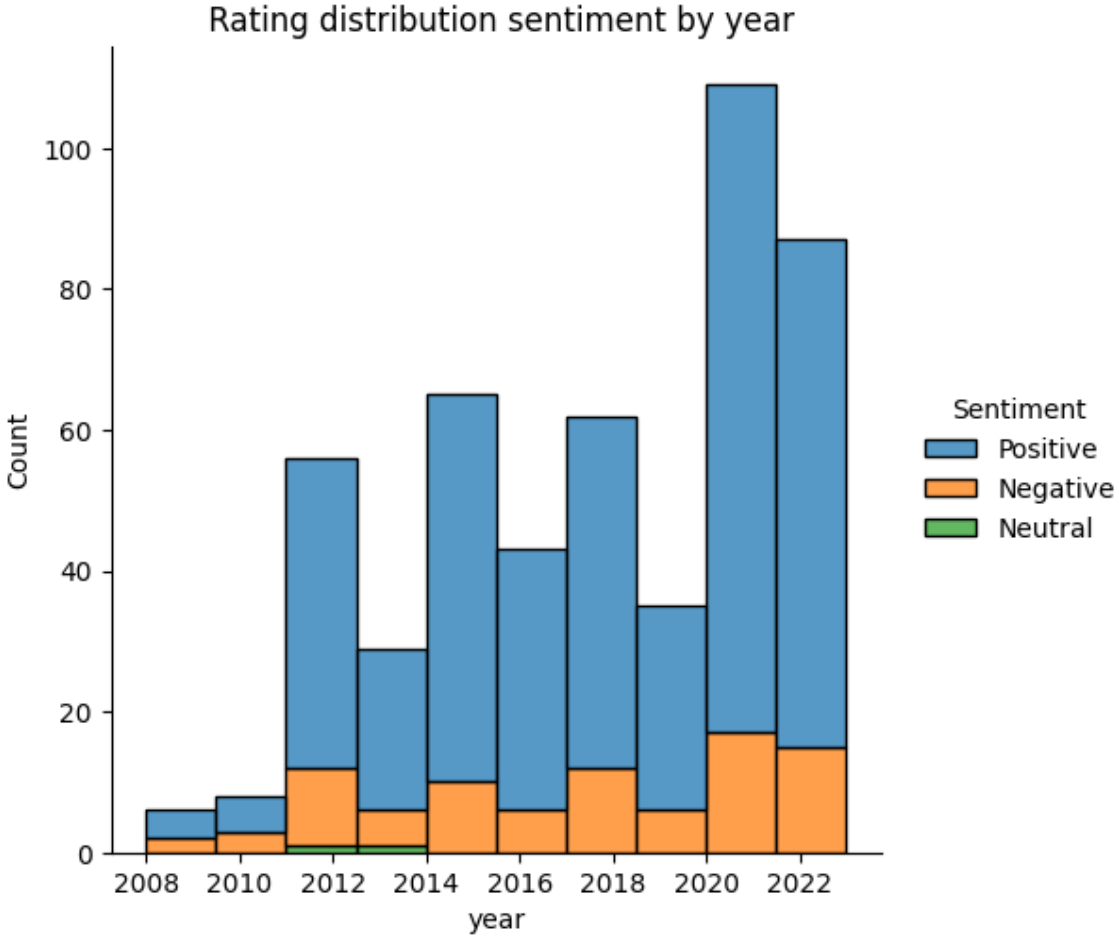


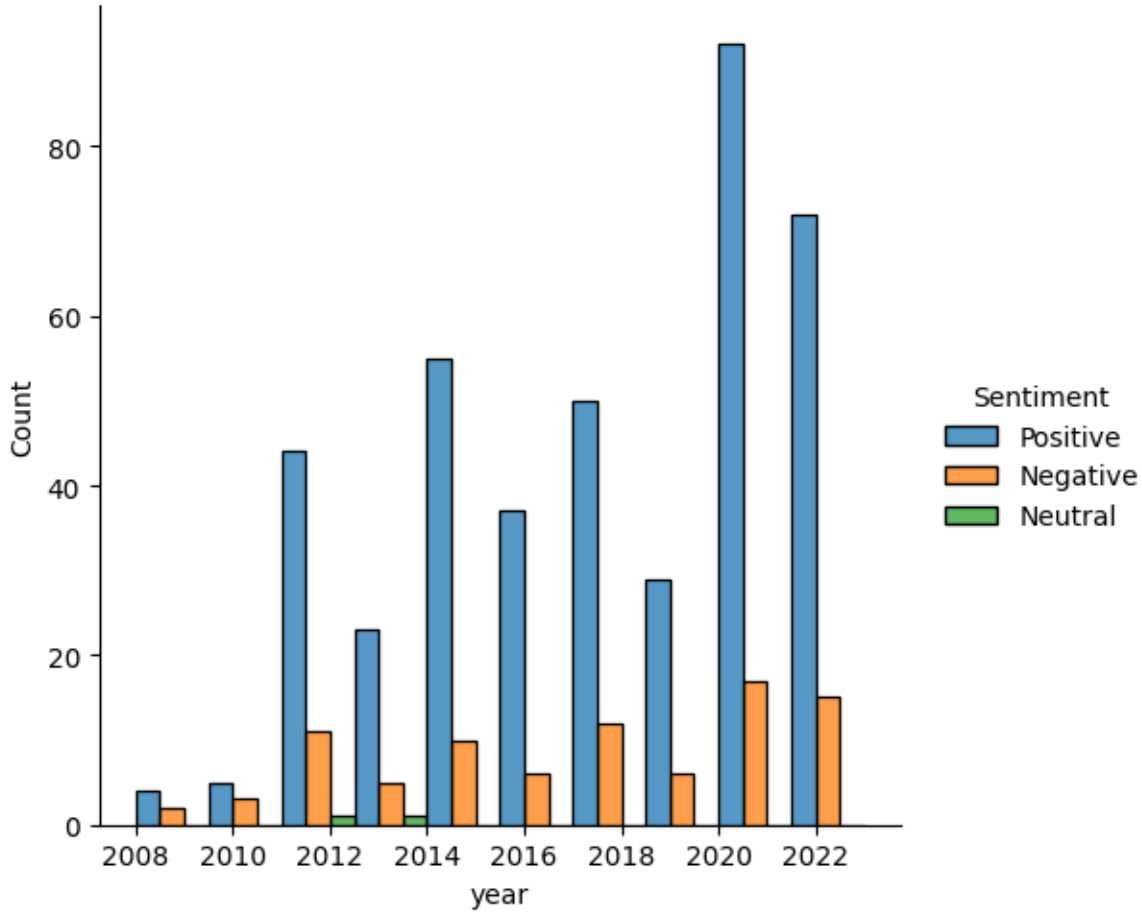
Positive words

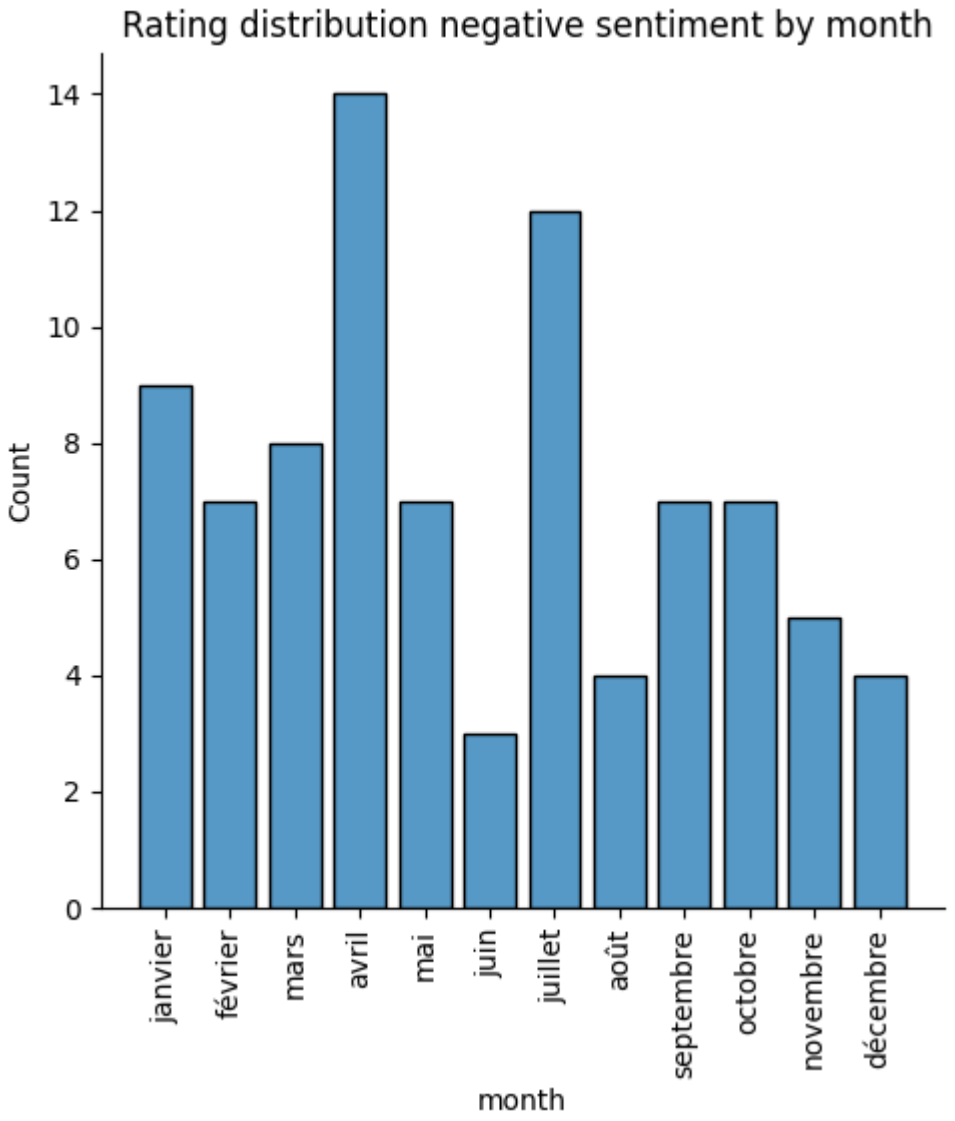


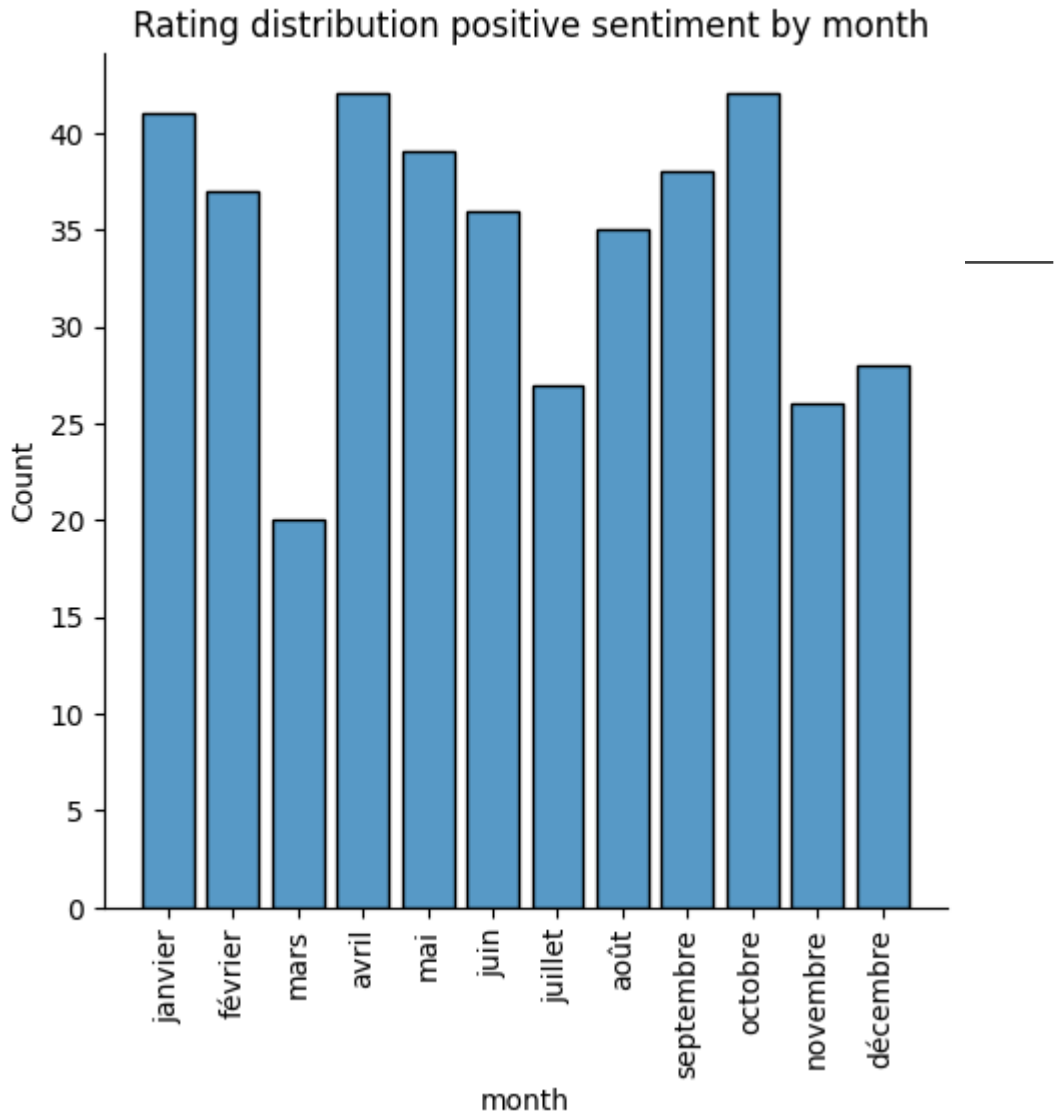
Neutral words

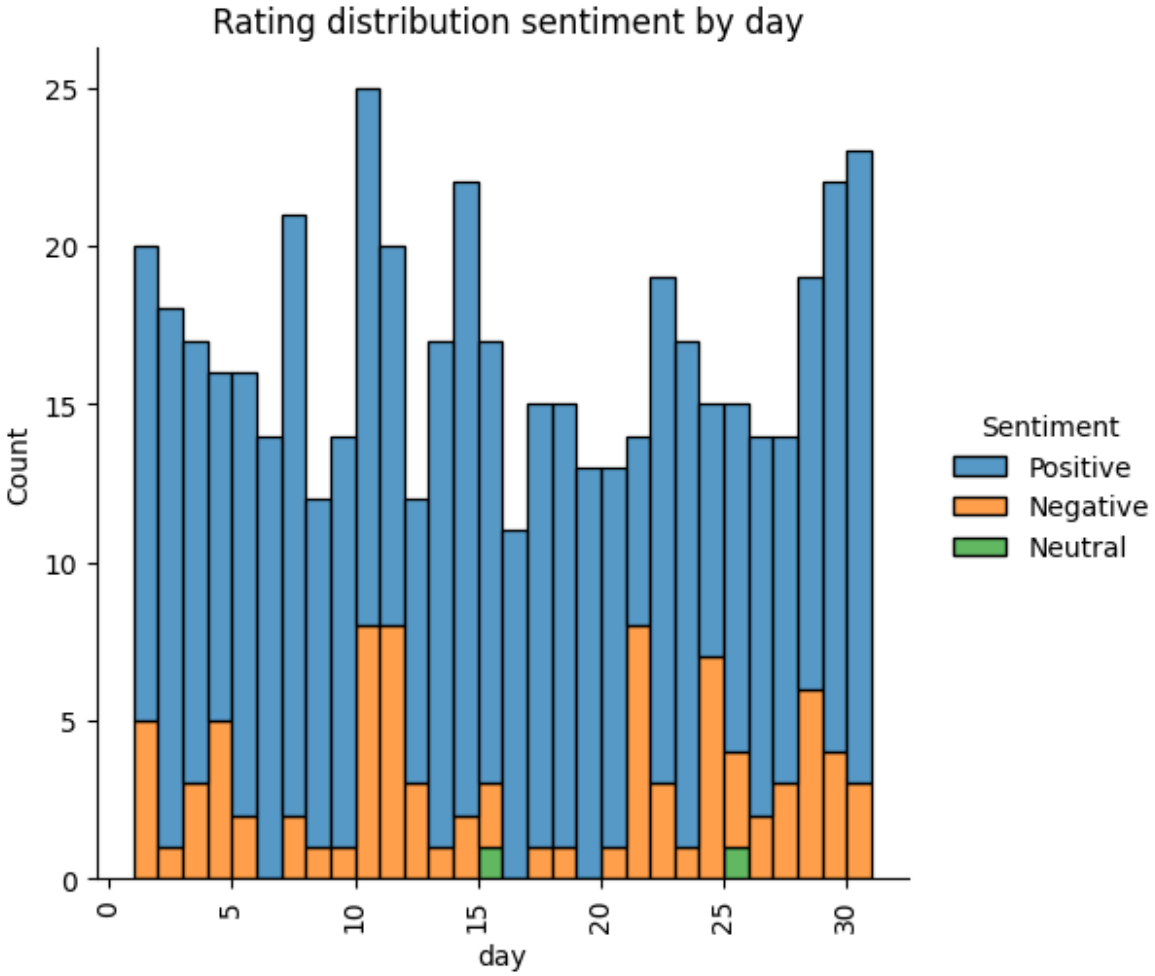
Negative words

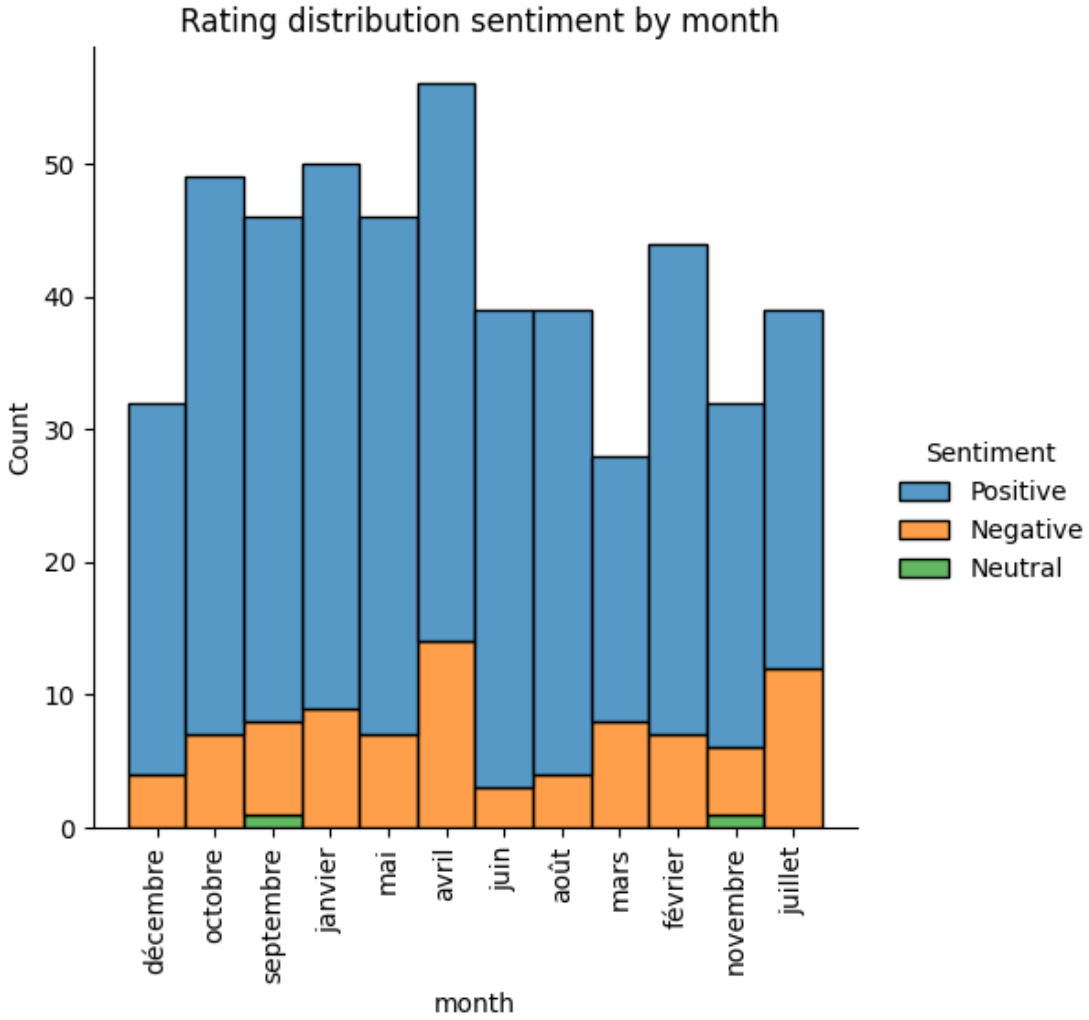

Rating distribution sentiment by year
Count
year
Sentiment
Positive
Negative
Neutral

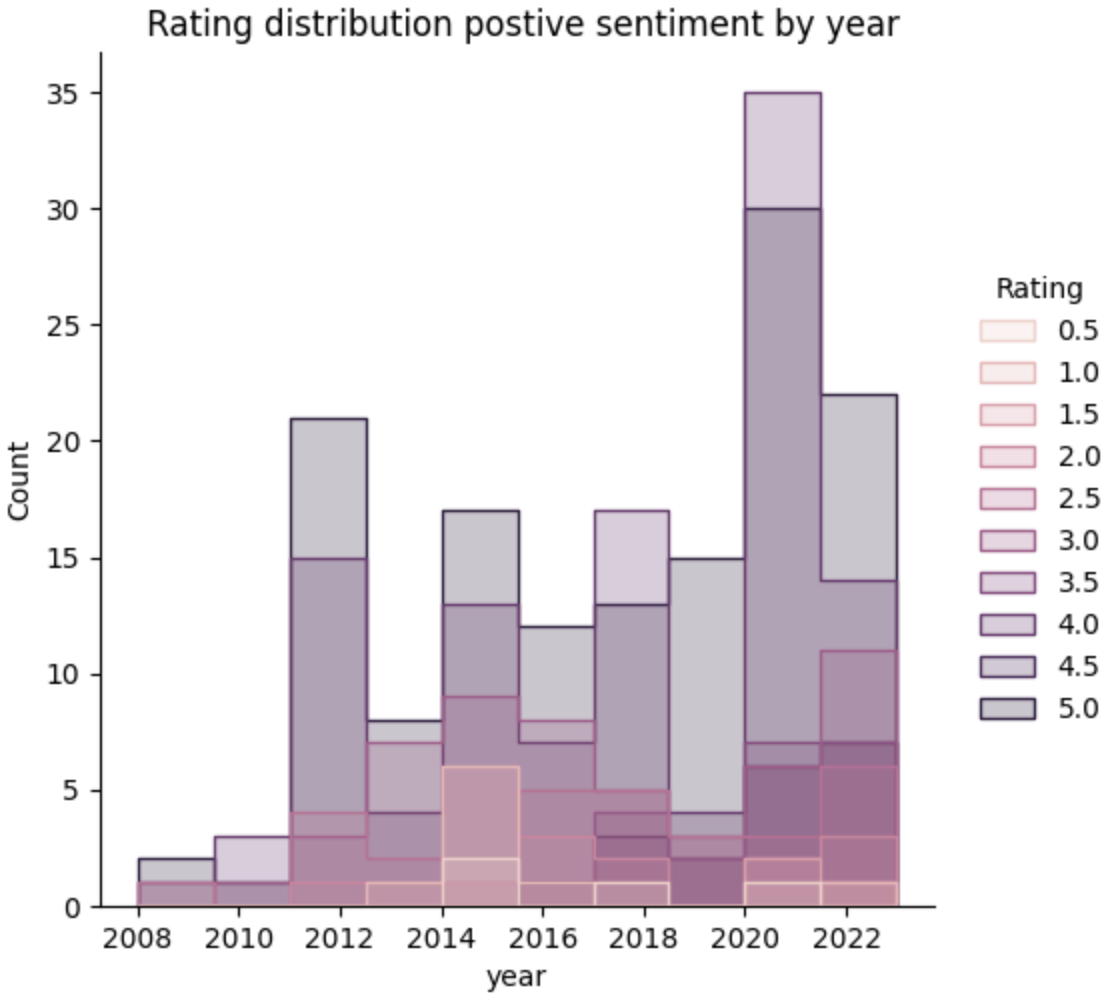

Count
year
Sentiment
Positive
Negative
Neutral

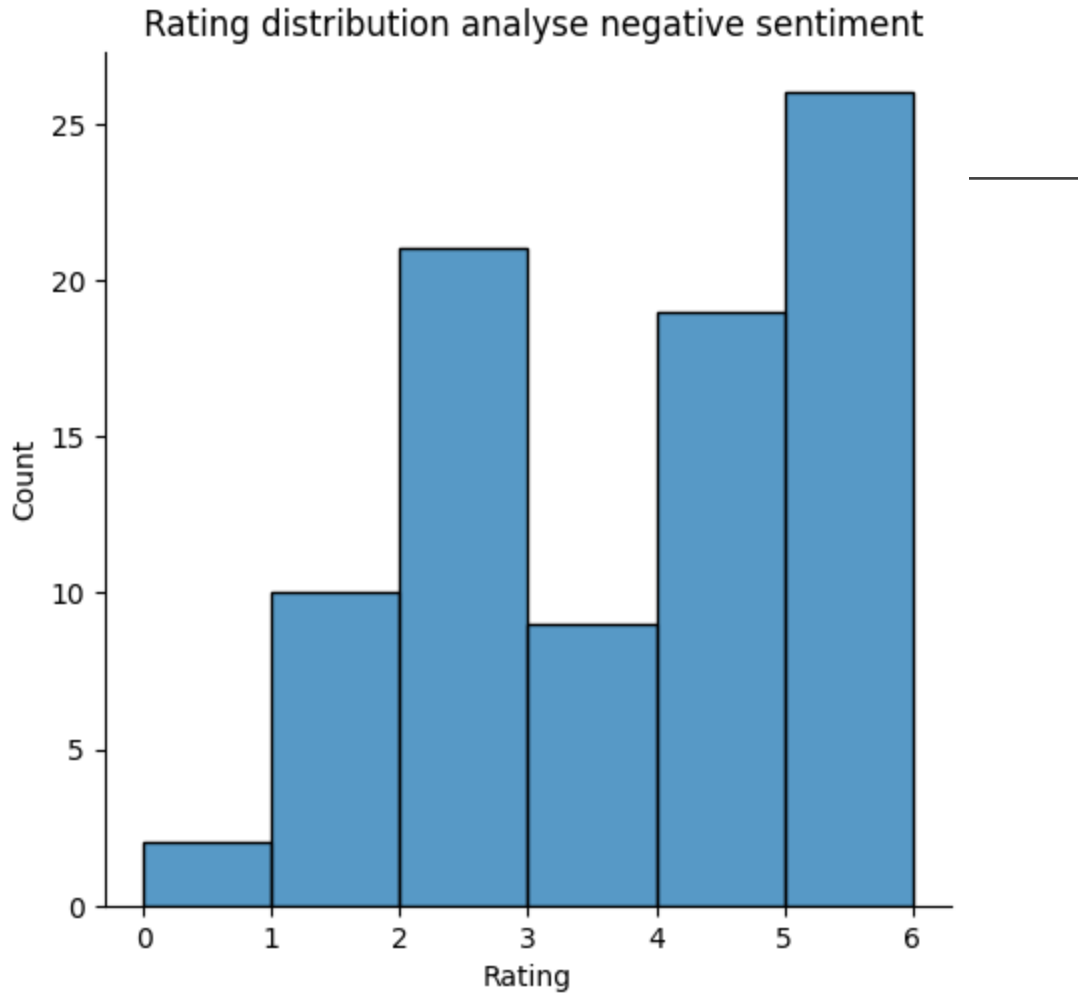

Rating distribution negative sentiment by month
Count
0
2
4
6
8
10
12
14
janvier
février
mars
avril
mai
juin
juillet
août
septembre
octobre
novembre
décembre
month

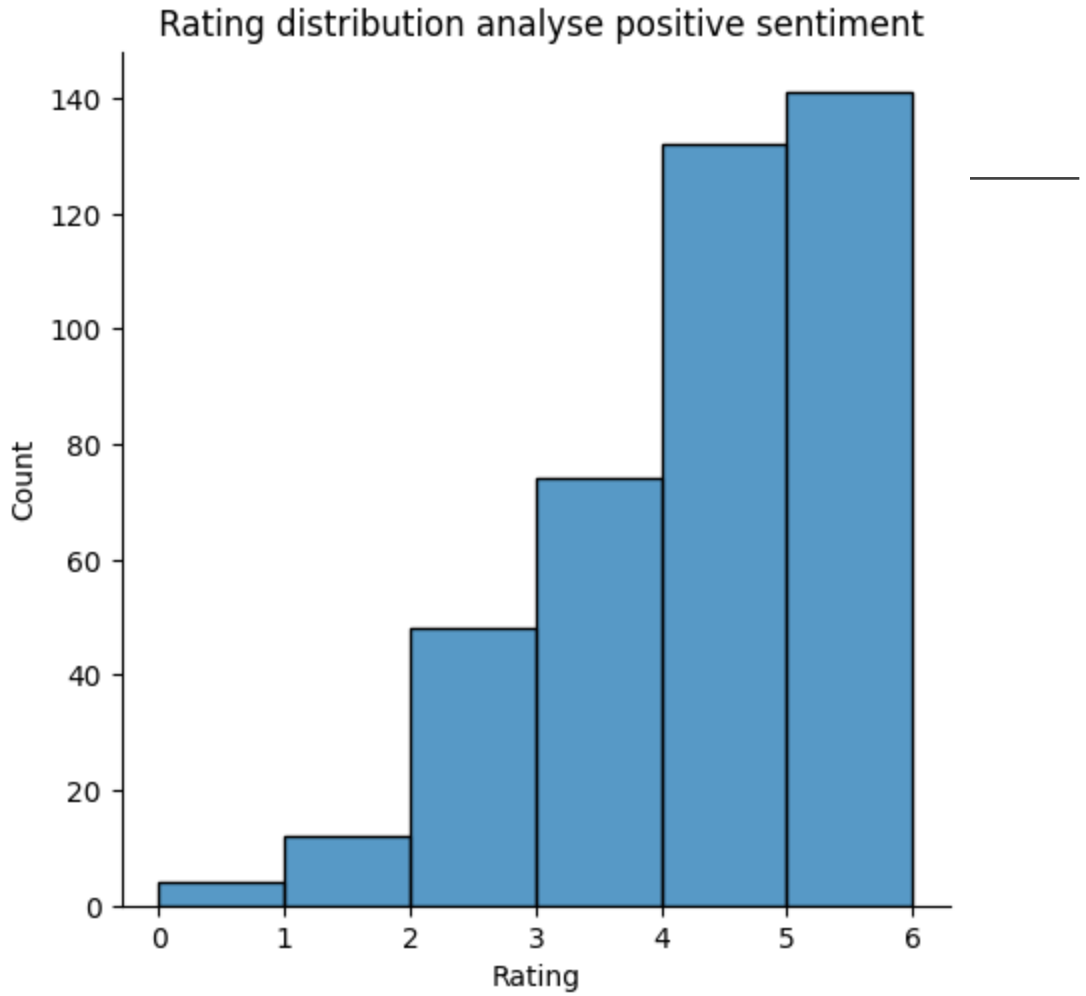

Rating distribution positive sentiment by month
Count
0
5
10
15
20
25
30
35
40
janvier
février
mars
avril
mai
juin
juillet
août
septembre
octobre
novembre
décembre
month


Rating distribution sentiment by day
Count
day
Sentiment
Positive
Negative
Neutral

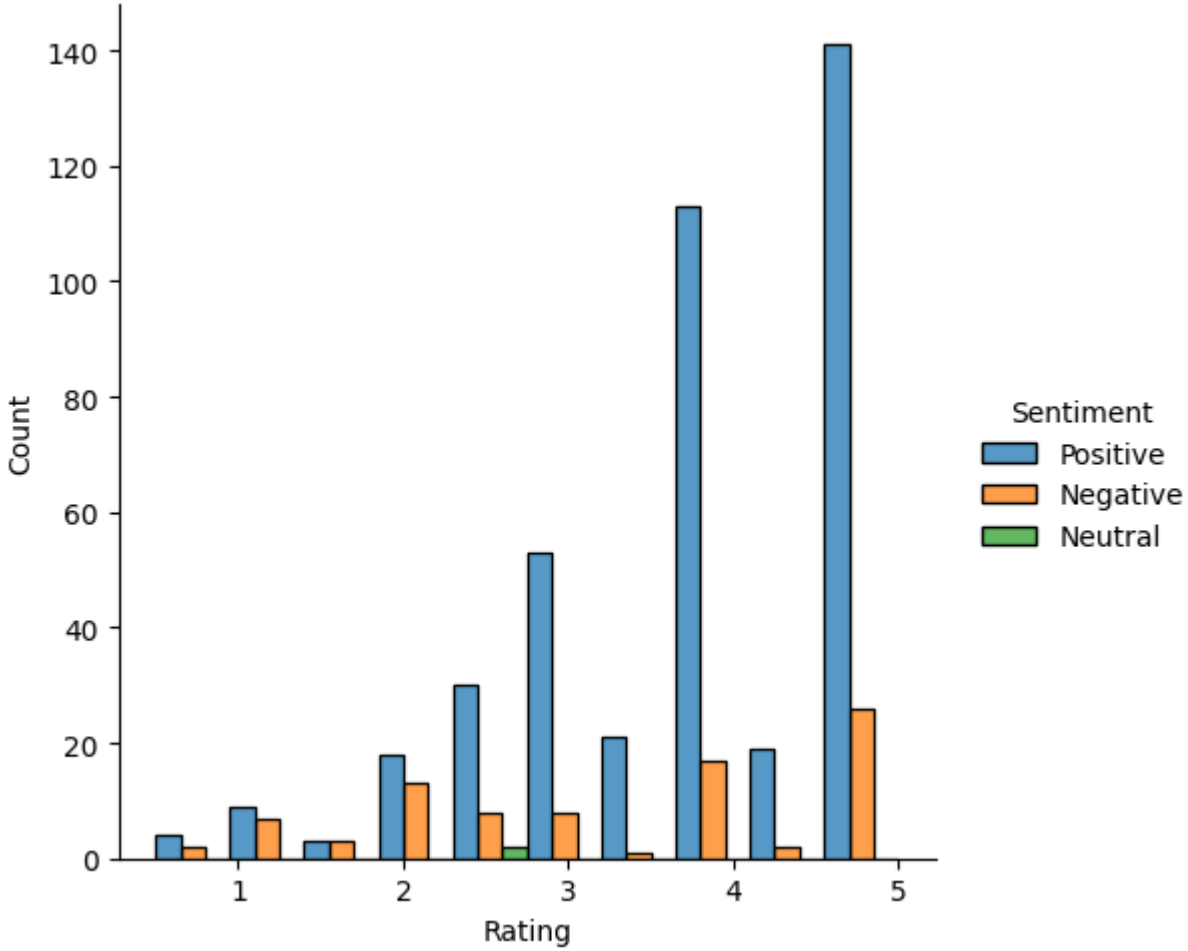

Rating distribution sentiment by month
Count
month
décembre
octobre
septembre
janvier
mai
avril
juin
août
mars
février
novembre
juillet
Sentiment
Positive
Negative
Neutral


Rating distribution postive sentiment by year
Count
year
0
5
10
15
20
25
30
35
2008
2010
2012
2014
2016
2018
2020
2022
Rating
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Rating distribution analyse negative sentiment
Count
Rating
0
5
10
15
20
25
0
1
2
3
4
5
6

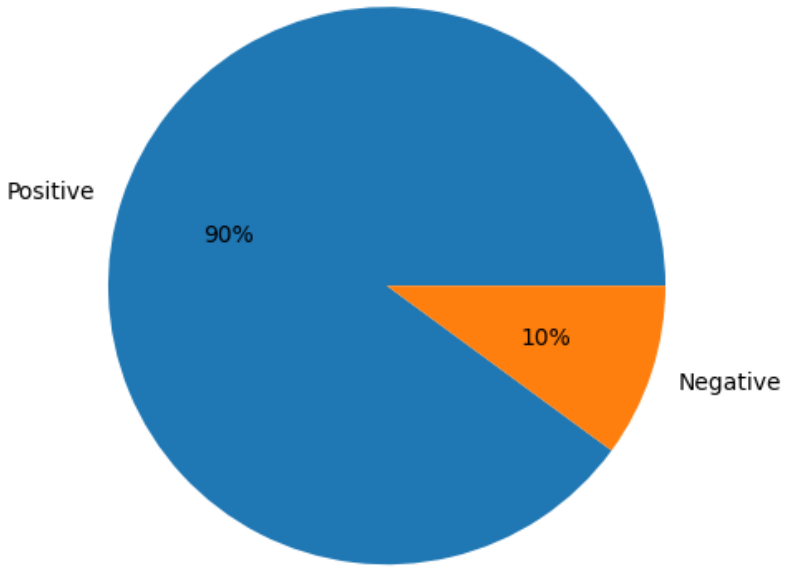

Rating distribution analyse positive sentiment
Count
Rating
0
20
40
60
80
100
120
140
0
1
2
3
4
5
6

Rating by sentiment

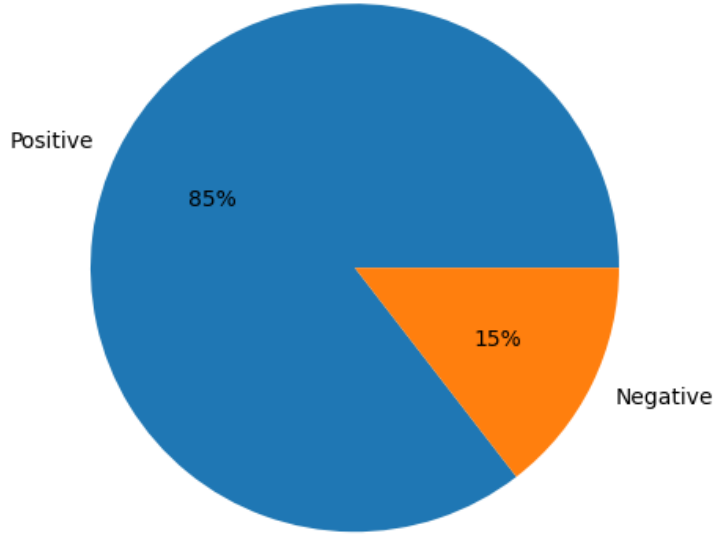


Cathegorize with key words the comments

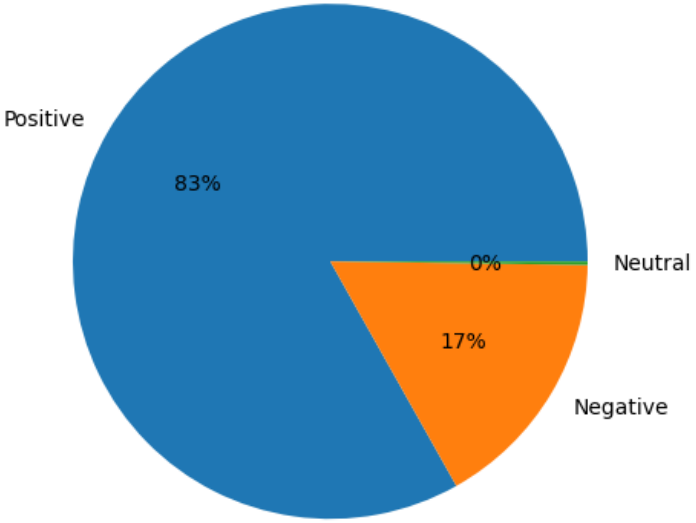
Sentiments in the comments that talk about the time line



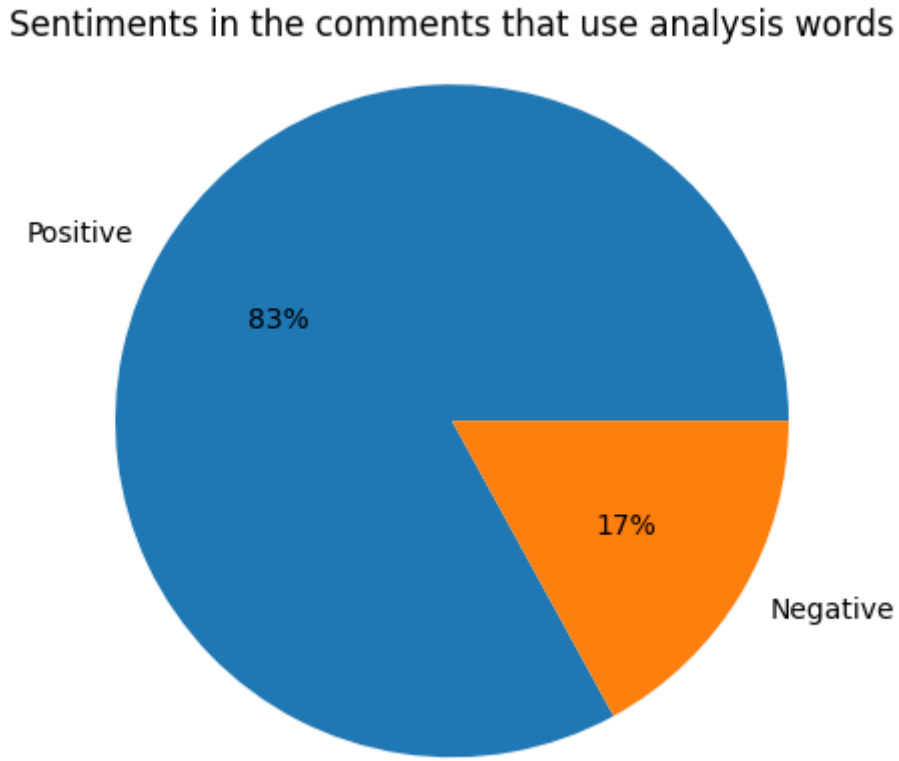
Sentiments in the comments that talk about the autor

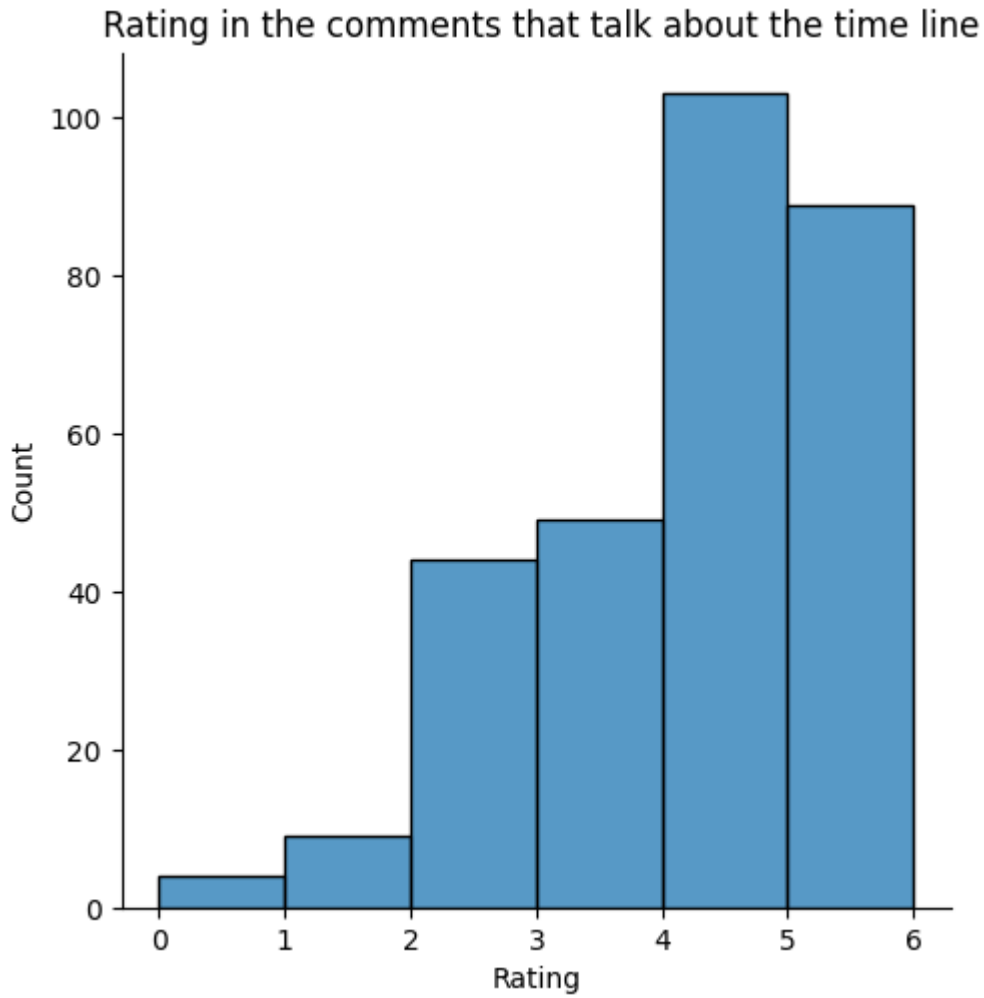


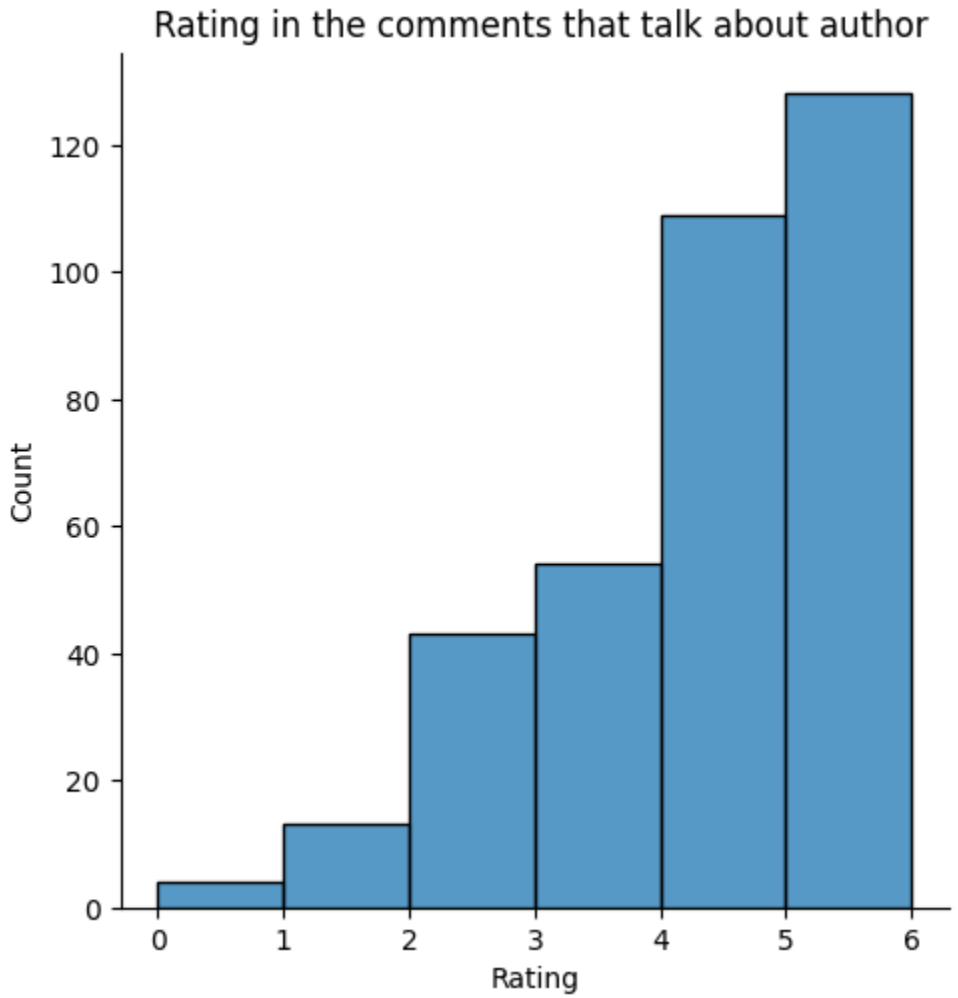
Sentiments in the comments that talk about characters

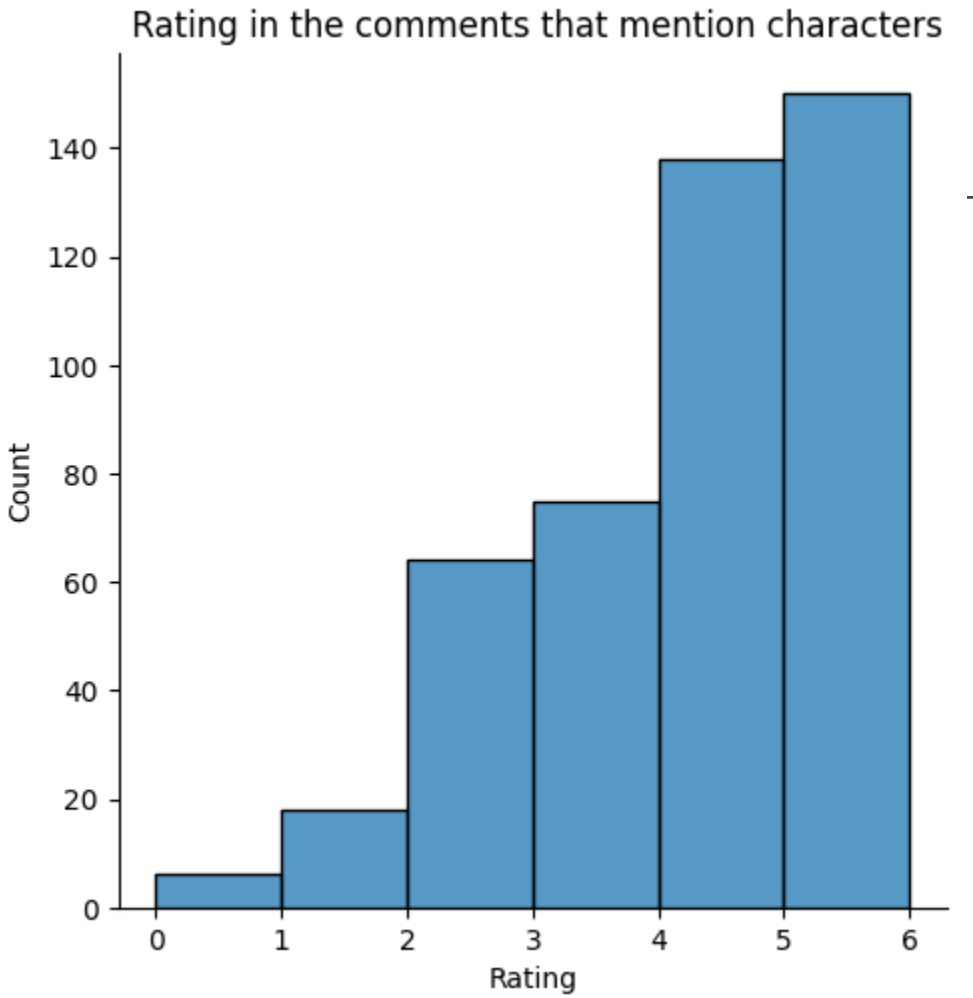


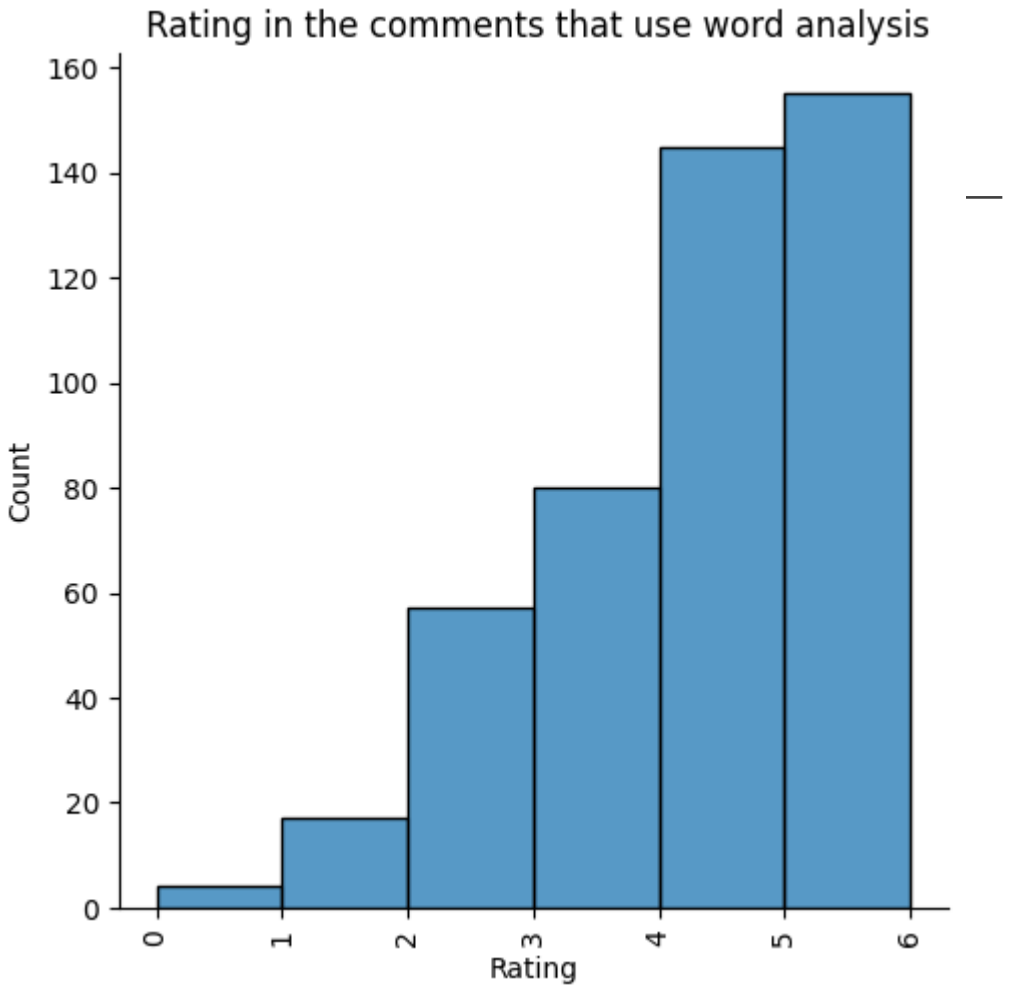
We create a cathegory that makes more Depth análisis with some commun adejctives



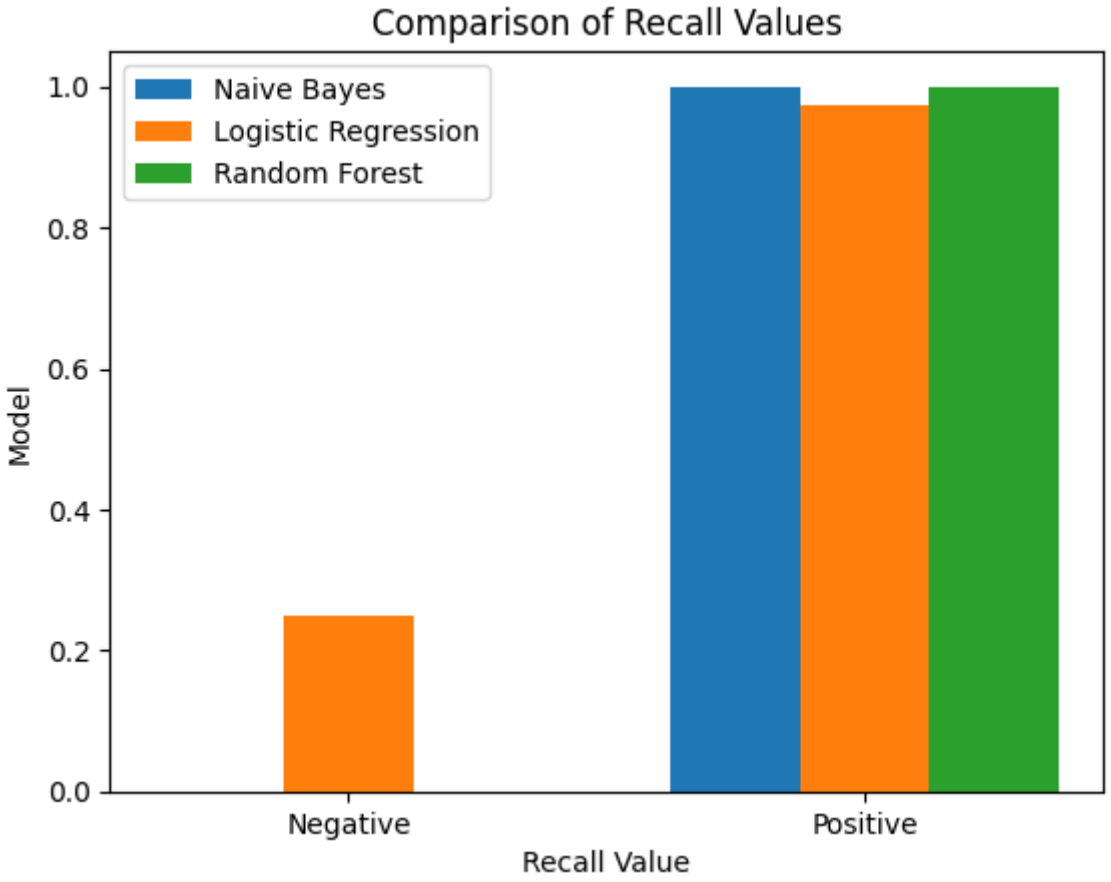

Rating in the comments that talk about the time line
Count
Rating
0
20
40
60
80
100
0
1
2
3
4
5
6

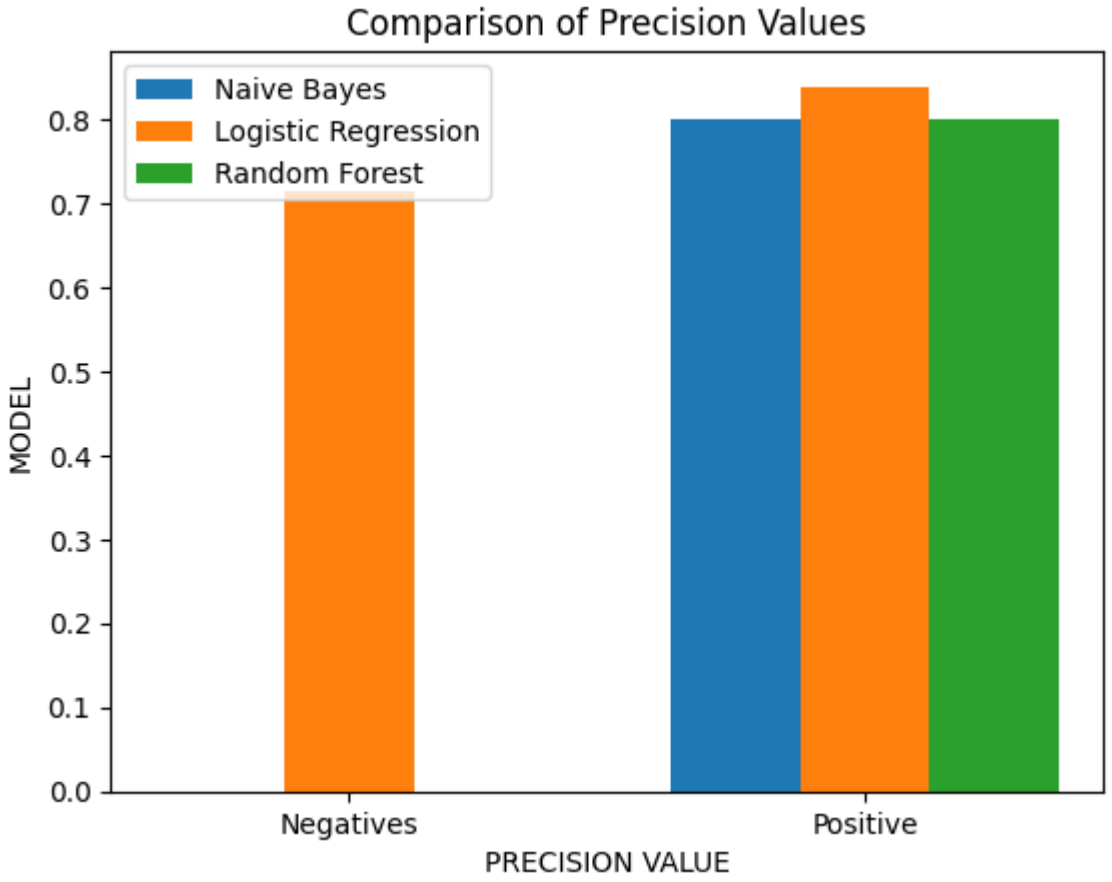

Rating in the comments that talk about author
Count
Rating
0
20
40
60
80
100
120
0
1
2
3
4
5
6

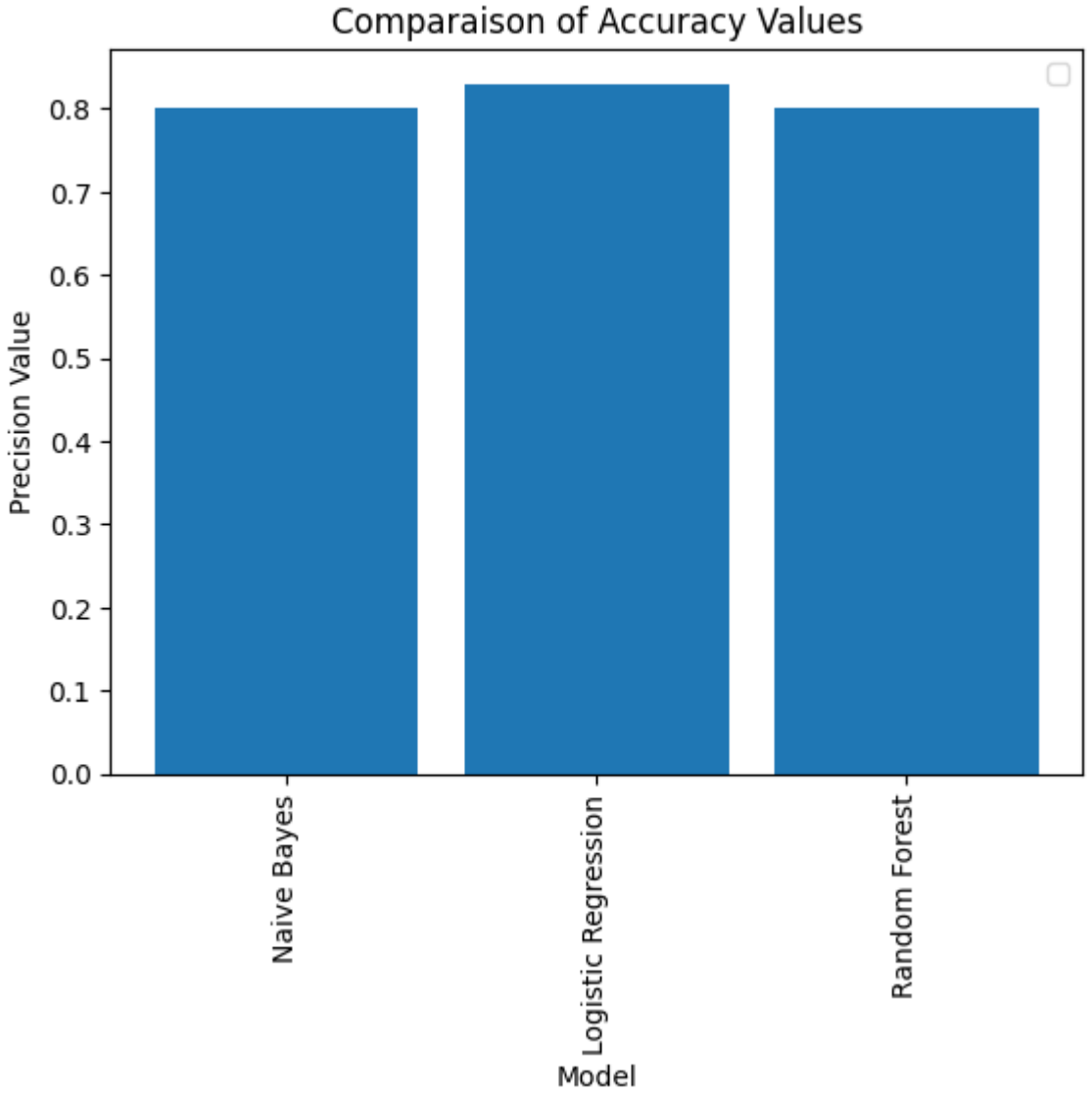

Rating in the comments that mention characters
Count
Rating
0
20
40
60
80
100
120
140
0
1
2
3
4
5
6


Rating in the comments that use word analysis
Count
Rating
0
20
40
60
80
100
120
140
160
0
1
2
3
4
5
6

We use Machine Learning to predict sentiment






Comparaison of Accuracy Values
Precision Value
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Naive Bayes
Logistic Regression
Random Forest
Model

Performing accuracy with LSTM model

```
model = Sequential()
model.add(Embedding(input_dim=num_words, output_dim=embedding_dim, input_length=max_length))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(lstm_units))
model.add(Dense(1, activation='sigmoid'))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/imdb.npz
17464789/17464789 [==============================] - 0s 0us/step
Epoch 1/4
625/625 [==============================] - 69s 103ms/step - loss: 0.4696 - accuracy: 0.7706 - val_loss: 0.3205 - val_accuracy: 0.8712
Epoch 2/4
625/625 [==============================] - 35s 56ms/step - loss: 0.2709 - accuracy: 0.8932 - val_loss: 0.2918 - val_accuracy: 0.8818
Epoch 3/4
625/625 [==============================] - 24s 38ms/step - loss: 0.1948 - accuracy: 0.9276 - val_loss: 0.3885 - val_accuracy: 0.8380
Epoch 4/4
625/625 [==============================] - 29s 47ms/step - loss: 0.2239 - accuracy: 0.9194 - val_loss: 0.3350 - val_accuracy: 0.8680
782/782 [==============================] - 9s 11ms/step
Accuracy: 0.85784
```