

cp-project-1

January 22, 2026

```
[23]: import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
[24]: df = pd.read_csv("/content/twitter_validation.csv", header=None)
df.columns = ['id', 'source', 'Sentiment', 'Text']

df = df.rename(columns={'Sentiment': 'Label'})

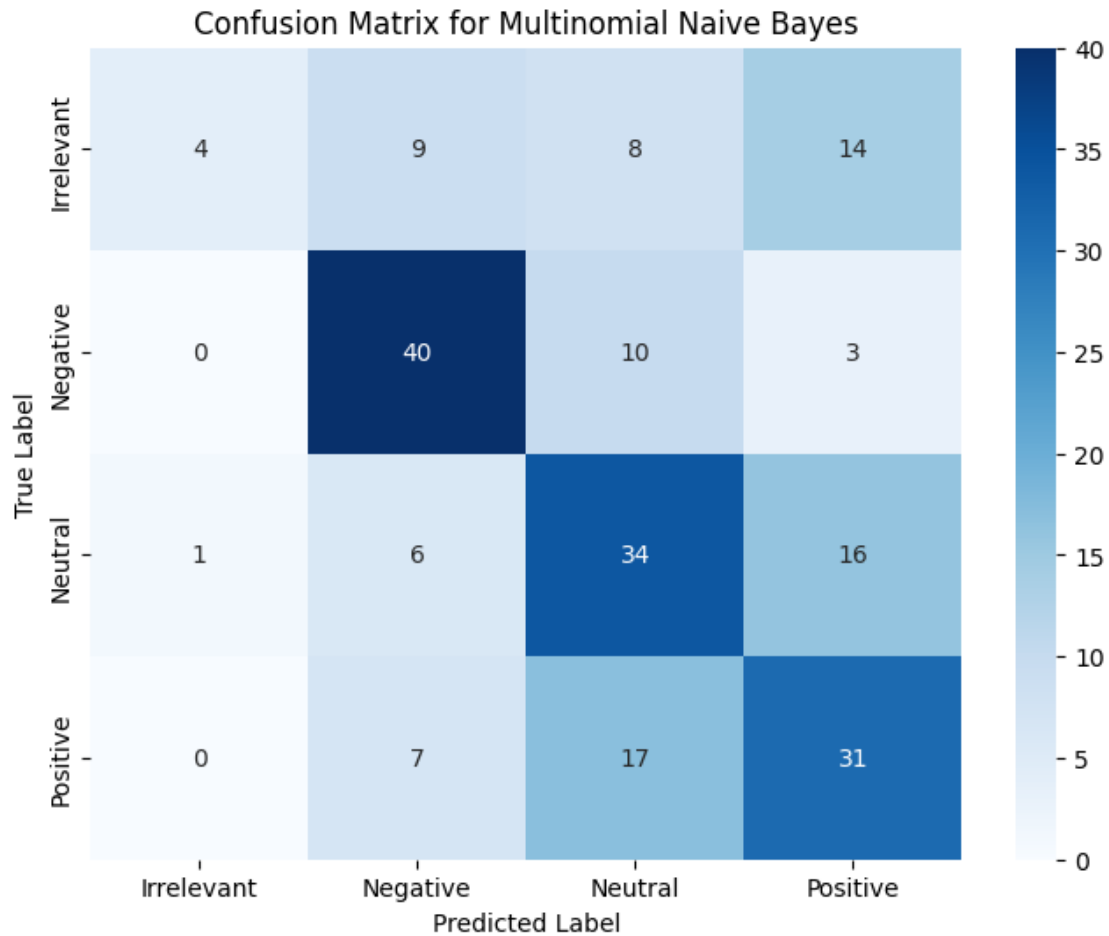
df = df[['Text', 'Label']]
```

```
[58]: from sklearn.metrics import confusion_matrix

# Re-initialize and train the MultinomialNB classifier to get its predictions
clf_nb = Pipeline([
    ('vectorizer_tri_grams', TfidfVectorizer()),
    ('naive_bayes', MultinomialNB())
])
clf_nb.fit(X_train, y_train)
y_pred_nb = clf_nb.predict(X_test)

cm = confusion_matrix(y_test, y_pred_nb)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=original_labels, yticklabels=original_labels)
plt.title('Confusion Matrix for Multinomial Naive Bayes')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```



```
[59]: import pandas as pd

# Get accuracy scores from previous executions
nb_accuracy = 0.545
rf_accuracy = 0.51

accuracy_df = pd.DataFrame({
    'Model': ['Multinomial Naive Bayes', 'Random Forest'],
    'Accuracy': [nb_accuracy, rf_accuracy]
})

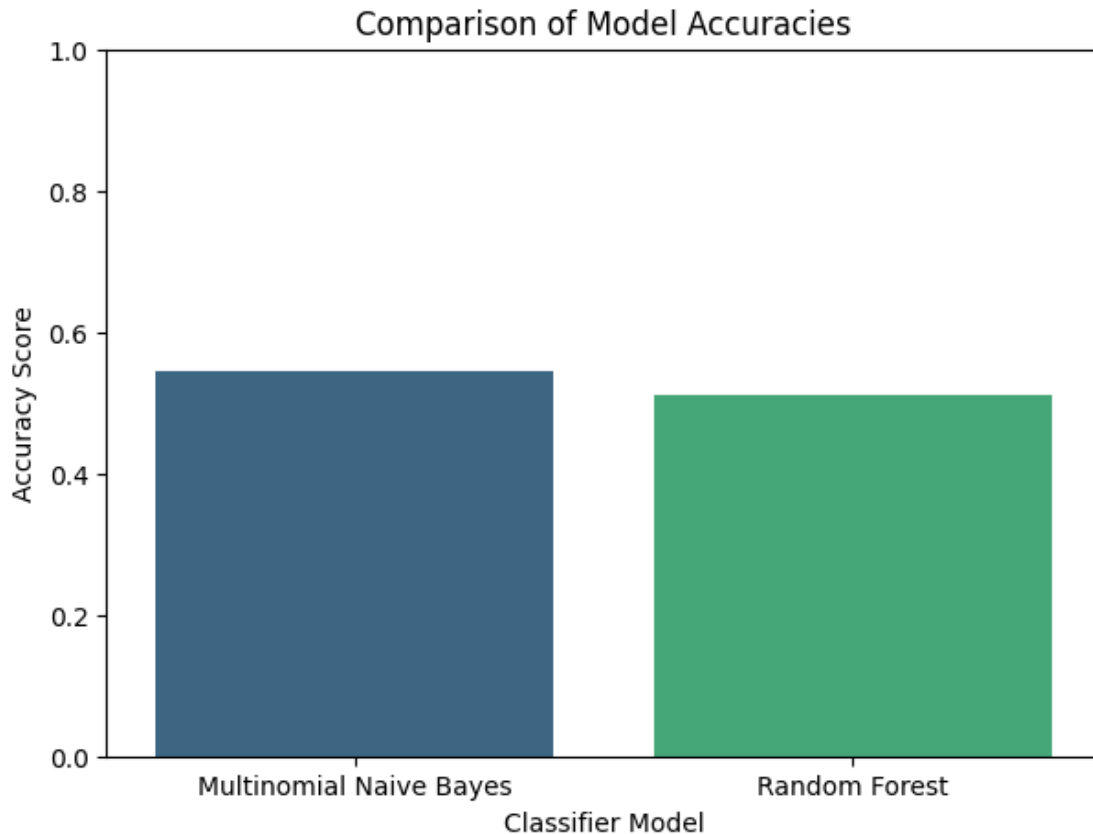
plt.figure(figsize=(7, 5))
sns.barplot(x='Model', y='Accuracy', data=accuracy_df, palette='viridis')
plt.title('Comparison of Model Accuracies')
plt.ylim(0, 1) # Set y-axis limit from 0 to 1 for accuracy
plt.ylabel('Accuracy Score')
plt.xlabel('Classifier Model')
```

```
plt.show()
```

/tmp/ipython-input-1660276792.py:13: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='Model', y='Accuracy', data=accuracy_df, palette='viridis')
```



```
[25]: print(df.shape)
```

```
(1000, 2)
```

```
[26]: df.head(10)
```

```
[26]:
```

	Text	Label
0	I mentioned on Facebook that I was struggling ...	Irrelevant
1	BBC News - Amazon boss Jeff Bezos rejects clai...	Neutral
2	@Microsoft Why do I pay for WORD when it funct...	Negative

3	CSGO matchmaking is so full of closet hacking,...	Negative
4	Now the President is slapping Americans in the...	Neutral
5	Hi @EAHelp I've had Madeleine McCann in my cel...	Negative
6	Thank you @EAMaddenNFL!! \n\nNew TE Austin Hoo...	Positive
7	Rocket League, Sea of Thieves or Rainbow Six: ...	Positive
8	my ass still knee-deep in Assassins Creed Odys...	Positive
9	FIX IT JESUS ! Please FIX IT ! What In the wor...	Negative

```
[27]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   Text    1000 non-null    object
 1   Label   1000 non-null    object
dtypes: object(2)
memory usage: 15.8+ KB
```

```
[28]: df['Label'].value_counts()
```

```
[28]: Label
Neutral      285
Positive     277
Negative     266
Irrelevant   172
Name: count, dtype: int64
```

```
[29]: for i in range(10):
        print(f"{i+1}: {df['Text'][i]} -> {df['Label'][i]}")
```

```
1: I mentioned on Facebook that I was struggling for motivation to go for a run
the other day, which has been translated by Tom's great auntie as 'Hayley can't
get out of bed' and told to his grandma, who now thinks I'm a lazy, terrible
person -> Irrelevant
2: BBC News - Amazon boss Jeff Bezos rejects claims company acted like a 'drug
dealer' bbc.co.uk/news/av/busine... -> Neutral
3: @Microsoft Why do I pay for WORD when it functions so poorly on my @SamsungUS
Chromebook? -> Negative
4: CSGO matchmaking is so full of closet hacking, it's a truly awful game. ->
Negative
5: Now the President is slapping Americans in the face that he really did commit
an unlawful act after his acquittal! From Discover on Google
vanityfair.com/news/2020/02/t... -> Neutral
6: Hi @EAHelp I've had Madeleine McCann in my cellar for the past 13 years and
the little sneaky thing just escaped whilst I was loading up some fifa points,
she took my card and I'm having to use my paypal account but it isn't working,
```

can you help me resolve it please? -> Negative
7: Thank you @EAMaddenNFL!!

New TE Austin Hooper in the ORANGE & BROWN!!

#Browns | @AustinHooper18

pic.twitter.com/GRg4xzFK0n -> Positive
8: Rocket League, Sea of Thieves or Rainbow Six: Siege? I love playing all three on stream but which is the best? #stream #twitch #RocketLeague #SeaOfThieves #RainbowSixSiege #follow -> Positive
9: my ass still knee-deep in Assassins Creed Odyssey with no way out anytime soon lmao -> Positive
10: FIX IT JESUS ! Please FIX IT ! What In the world is going on here. @PlayStation @AskPlayStation @Playstationsup @Treyarch @CallofDuty negative 345 silver wolf error code pic.twitter.com/ziRyhfrf59Q -> Negative

```
[30]: df.dropna(inplace=True)
```

```
[31]: !pip install spacy
      !python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.21.1)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.1)
```

Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)

Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)

Requirement already satisfied: pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.12.3)

Requirement already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)

Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)

Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)

Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (0.7.0)

Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (2.41.4)

Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (4.15.0)

Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy) (0.4.2)

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.4.4)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.11)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2.5.0)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2026.1.4)

Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (1.3.3)

Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (0.1.5)

Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spacy) (8.3.1)

Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy) (0.23.0)

Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy)

(7.5.0)

Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages
(from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.4.2->spacy) (2.0.1)

Collecting en-core-web-sm==3.8.0

Using cached https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0-py3-none-any.whl (12.8 MB)

Download and installation successful

You can now load the package via `spacy.load('en_core_web_sm')`

Restart to reload dependencies

If you are in a Jupyter or Colab notebook, you may need to restart Python in order to load all the package's dependencies. You can do this by selecting the 'Restart kernel' or 'Restart runtime' option.

```
[32]: import spacy
      nlp = spacy.load("en_core_web_sm")
```

```
[33]: def preprocess(text):
      # remove stop words and lemmatize the text
      doc = nlp(text)
      filtered_tokens = []
      for token in doc:
          if token.is_stop or token.is_punct:
              continue
          filtered_tokens.append(token.lemma_)

      return " ".join(filtered_tokens)
```

```
[34]: df['Preprocessed Text'] = df['Text'].apply(preprocess)
```

```
[35]: df
```

```
[35]:
```

	Text	Label \
0	I mentioned on Facebook that I was struggling ...	Irrelevant
1	BBC News - Amazon boss Jeff Bezos rejects clai...	Neutral
2	@Microsoft Why do I pay for WORD when it funct...	Negative
3	CSGO matchmaking is so full of closet hacking,...	Negative
4	Now the President is slapping Americans in the...	Neutral
..
995	Toronto is the arts and culture capital of ...	Irrelevant
996	THIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...	Irrelevant
997	Today sucked so it's time to drink wine n play...	Positive
998	Bought a fraction of Microsoft today. Small wins.	Positive
999	Johnson & Johnson to stop selling talc baby po...	Neutral

```

                                Preprocessed Text
0   mention Facebook struggle motivation run day t...
1   BBC News Amazon boss Jeff Bezos reject claim c...
2   @Microsoft pay WORD function poorly @samsungu ...
3   csgo matchmaking closet hacking truly awful game
4   President slap Americans face commit unlawful ...
..
995  Toronto art culture capital Canada wonder ...
996  ACTUALLY good TOT bring viewer \n\n people get...
997  today suck time drink wine n play borderland s...
998           buy fraction Microsoft today small win
999  Johnson Johnson stop sell talc baby powder U.S...

```

[1000 rows x 3 columns]

```

[36]: # Strip whitespace from the 'Label' column to handle potential inconsistencies
      ↪(e.g., ' Twitter' vs 'Twitter')
df['Label'] = df['Label'].str.strip()

le_model = LabelEncoder()
df['Label'] = le_model.fit_transform(df['Label'])

```

```

[37]: df.head(10)

```

```

[37]:
                                Text  Label  \
0   I mentioned on Facebook that I was struggling ...      0
1   BBC News - Amazon boss Jeff Bezos rejects clai...      2
2   @Microsoft Why do I pay for WORD when it funct...      1
3   CSGO matchmaking is so full of closet hacking,...      1
4   Now the President is slapping Americans in the...      2
5   Hi @EAHelp I've had Madeleine McCann in my cel...      1
6   Thank you @EAMaddenNFL!! \n\nNew TE Austin Hoo...      3
7   Rocket League, Sea of Thieves or Rainbow Six: ...      3
8   my ass still knee-deep in Assassins Creed Odys...      3
9   FIX IT JESUS ! Please FIX IT ! What In the wor...      1

```

```

                                Preprocessed Text
0   mention Facebook struggle motivation run day t...
1   BBC News Amazon boss Jeff Bezos reject claim c...
2   @Microsoft pay WORD function poorly @samsungu ...
3   csgo matchmaking closet hacking truly awful game
4   President slap Americans face commit unlawful ...
5   hi @EAHelp Madeleine McCann cellar past 13 yea...
6   thank @EAMaddenNFL \n\n New TE Austin Hooper O...
7   Rocket League Sea Thieves Rainbow siege  love...
8   ass knee deep Assassins Creed Odyssey way anyt...
9   fix JESUS fix world go  @playstation @askplay...

```

```
[38]: label_counts = df['Label'].value_counts()
single_instance_labels = label_counts[label_counts == 1].index

df_filtered = df[~df['Label'].isin(single_instance_labels)].copy()

X_train, X_test, y_train, y_test = train_test_split(df_filtered['Preprocessed_
↪Text'], df_filtered['Label'],
                                                    test_size=0.2,
↪random_state=42, stratify=df_filtered['Label'])
```

```
[39]: print("Shape of training data:", X_train.shape, y_train.shape)
print("Shape of testing data:", X_test.shape, y_test.shape)
```

```
Shape of training data: (800,) (800,)
Shape of testing data: (200,) (200,)
```

```
[40]: clf = Pipeline([
    ('vectorizer_tri_grams', TfidfVectorizer()),
    ('naive_bayes', (MultinomialNB()))
])
```

```
[41]: clf.fit(X_train, y_train)
```

```
[41]: Pipeline(steps=[('vectorizer_tri_grams', TfidfVectorizer()),
    ('naive_bayes', MultinomialNB())])
```

```
[42]: y_pred = clf.predict(X_test)
```

```
[43]: print(accuracy_score(y_test, y_pred))
```

```
0.545
```

```
[44]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.80	0.11	0.20	35
1	0.65	0.75	0.70	53
2	0.49	0.60	0.54	57
3	0.48	0.56	0.52	55
accuracy			0.55	200
macro avg	0.61	0.51	0.49	200
weighted avg	0.58	0.55	0.52	200

```
[45]: clf = Pipeline([
        ('vectorizer_tri_grams', TfidfVectorizer()),
        ('naive_bayes', (RandomForestClassifier()))
    ])
```

```
[46]: clf.fit(X_train, y_train)
```

```
[46]: Pipeline(steps=[('vectorizer_tri_grams', TfidfVectorizer()),
        ('naive_bayes', RandomForestClassifier())])
```

```
[47]: y_pred = clf.predict(X_test)
```

```
[48]: print(accuracy_score(y_test, y_pred))
```

0.51

```
[49]: print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	1.00	0.17	0.29	35
1	0.44	0.87	0.59	53
2	0.51	0.42	0.46	57
3	0.60	0.47	0.53	55
accuracy			0.51	200
macro avg	0.64	0.48	0.47	200
weighted avg	0.60	0.51	0.48	200

```
[52]: test_df = pd.read_csv('/content/twitter_validation.csv', header=None)
test_df.columns = ['id', 'source', 'Sentiment', 'Text'] # Assign columns as per
↳ the original structure
test_df = test_df.rename(columns={'Sentiment': 'Label'})
test_df = test_df[['Text', 'Label']]
```

```
[53]: test_text = test_df['Text'][10]
print(f"{test_text} ==> {test_df['Label'][10]}")
```

The professional dota 2 scene is fucking exploding and I completely welcome it.

Get the garbage out. ==> Positive

```
[54]: test_text_processed = [preprocess(test_text)]
test_text_processed
```

```
[54]: ['professional dota 2 scene fuck explode completely welcome \n\n garbage']
```

```
[55]: test_text = clf.predict(test_text_processed)
```

```
[56]: # Use the inverse transform of the LabelEncoder to get the original class names
      classes = le_model.classes_

      print(f"True Label: {test_df['Label'].iloc[10]}") # Using .iloc to access by_
        ↪ position
      print(f'Predict Label: {classes[test_text[0]]}')

```

True Label: Positive
Predict Label: Negative

```
[57]: import matplotlib.pyplot as plt
      import seaborn as sns

      # Get the counts of each label
      label_counts = df['Label'].value_counts().sort_index()

      # Map encoded labels back to original class names for better readability
      original_labels = le_model.inverse_transform(label_counts.index)

      plt.figure(figsize=(8, 6))
      sns.barplot(x=original_labels, y=label_counts.values, palette='viridis')
      plt.title('Distribution of Sentiment Labels')
      plt.xlabel('Sentiment')
      plt.ylabel('Count')
      plt.xticks(rotation=45)
      plt.tight_layout()
      plt.show()

```

/tmp/ipython-input-779907872.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=original_labels, y=label_counts.values, palette='viridis')
```

