```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import re
import nltk
import string
from nltk.corpus import stopwords
from nltk.stem import LancasterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
```

```
train_path = "/content/train_data.txt"
train_data = pd.read_csv(train_path, sep=':::', names=['Title', 'Genre', 'Description'], engine='python')
```

```
print(train_data.describe())
```

```
                                Title    Genre  \
count                           19190    19190
unique                          19190       27
top      Oscar et la dame rose (2009)    drama
freq                                 1     4750

                                            Description
count                                             19190
unique                                            19157
top      Grammy - music award of the American academy ...
freq                                                  6
```

```
print(train_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19190 entries, 1 to 11180
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Title        19190 non-null  object
 1   Genre        19190 non-null  object
 2   Description  19190 non-null  object
dtypes: object(3)
memory usage: 599.7+ KB
None
```

```
print(train_data.isnull().sum())
```

```
Title          0
Genre          0
Description    0
dtype: int64
```

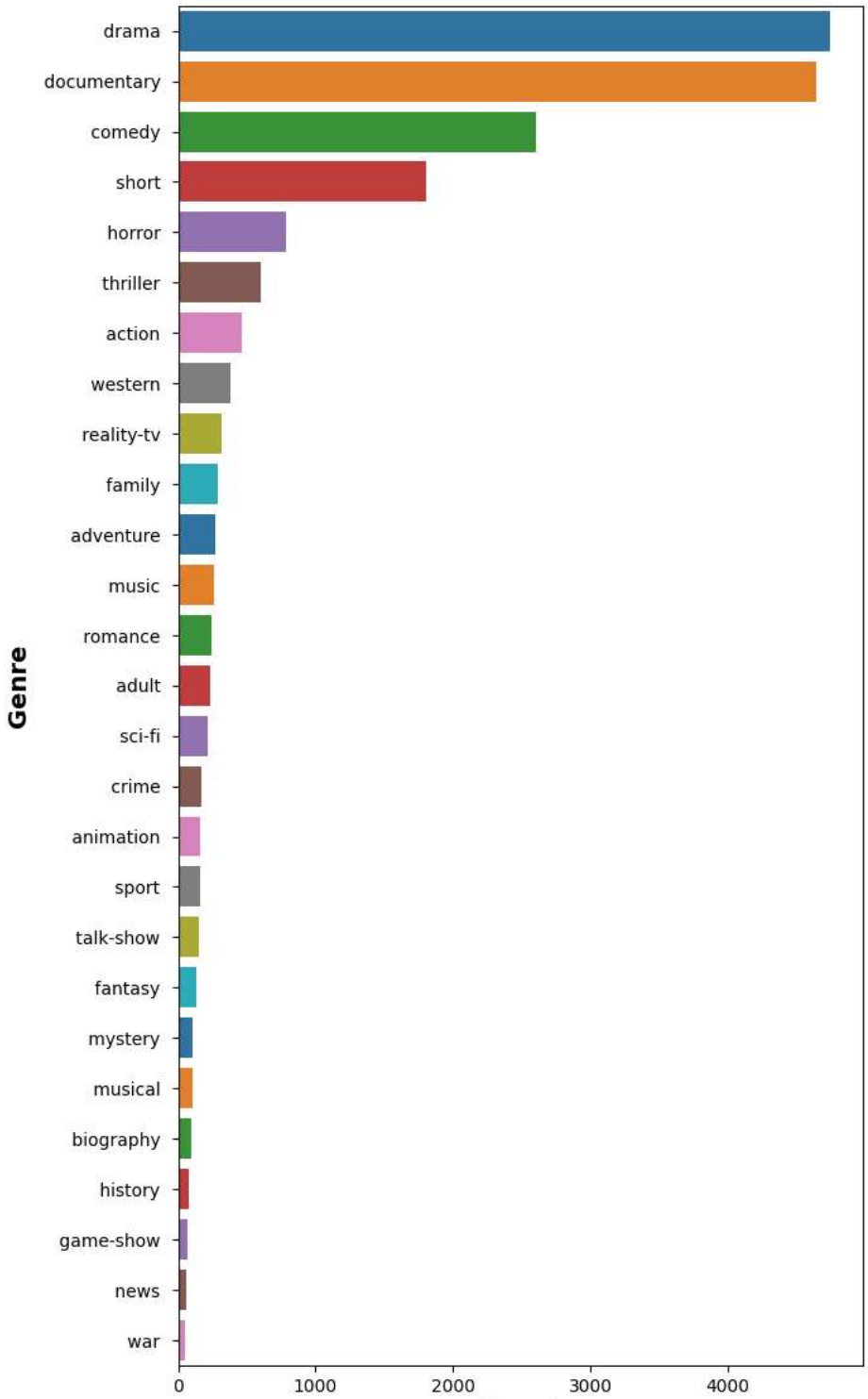
```
test_path = "/content/test_data.txt"
test_data = pd.read_csv(test_path, sep=':::', names=['Id', 'Title', 'Description'], engine='python')
test_data.head()
```

	Id	Title	Description
0	1	Edgar's Lunch (1998)	L.R. Brane loves his life - his car, his apar...
1	2	La guerra de papá (1977)	Spain, March 1964: Quico is a very naughty ch...
2	3	Off the Beaten Track (2010)	One year in the life of Albin and his family ...
3	4	Meu Amigo Hindu (2015)	His father has died, he hasn't spoken with hi...
4	5	Er nu zhai (1955)	Before he was known internationally as a mart...

Plot the distribution of genres in the training **data**

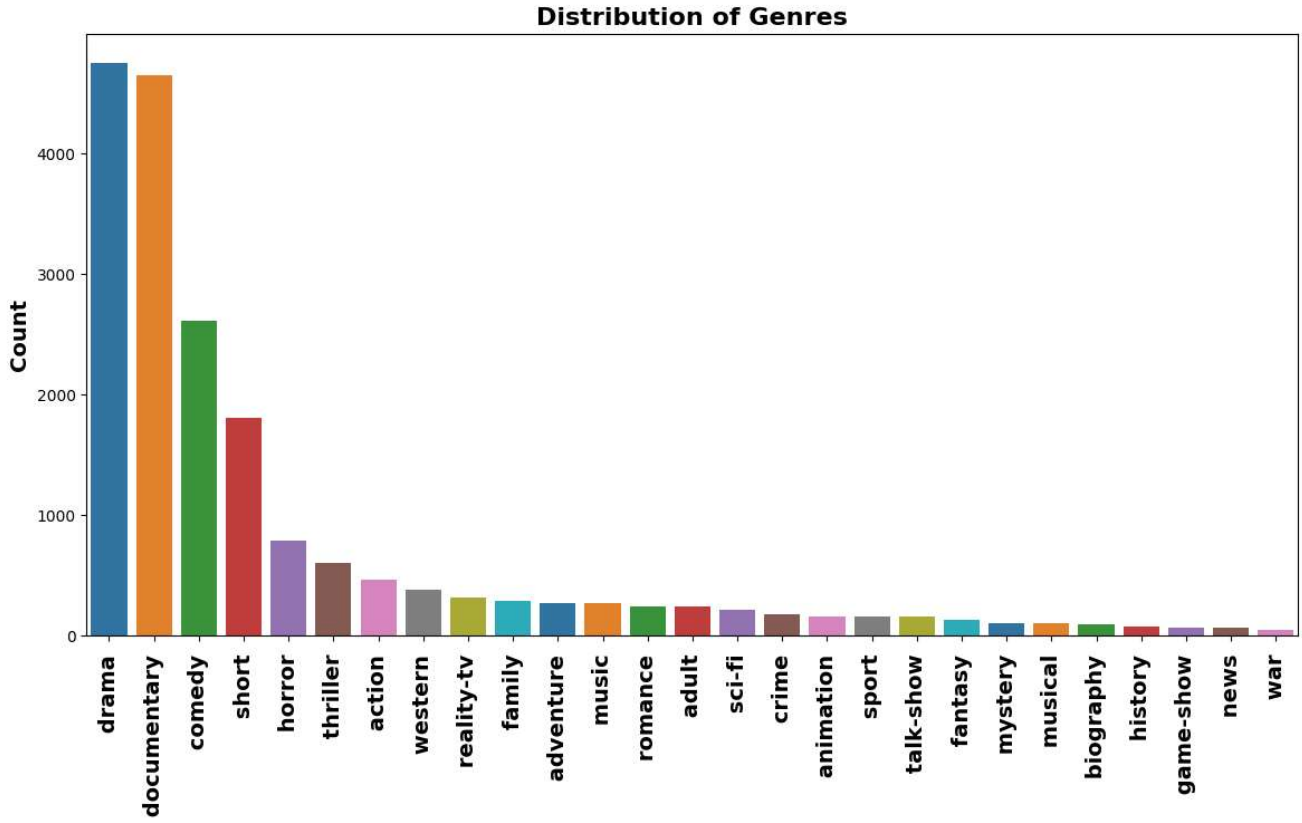
```
plt.figure(figsize=(7, 14))
sns.countplot(data=train_data, y='Genre', order=train_data['Genre'].value_counts().index, palette='tab10')
plt.xlabel('Count', fontsize=14, fontweight='bold')
plt.ylabel('Genre', fontsize=14, fontweight='bold')
```

```
Text(0, 0.5, 'Genre')
```



Plot the distribution of genres using a bar **plot**

```
plt.figure(figsize=(14, 7))
counts = train_data['Genre'].value_counts()
sns.barplot(x=counts.index, y=counts, palette='tab10')
plt.xlabel('Genre', fontsize=14, fontweight='bold')
plt.ylabel('Count', fontsize=14, fontweight='bold')
plt.title('Distribution of Genres', fontsize=16, fontweight='bold')
plt.xticks(rotation=90, fontsize=14, fontweight='bold')
plt.show()
```



Initialize the stemmer and stop **words**

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

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
stemmer = LancasterStemmer()
stop_words = set(stopwords.words('english'))
```

Define the clean_text function

```
def clean_text(text):
    text = text.lower()  # Lowercase all characters
    text = re.sub(r'@\S+', '', text)  # Remove Twitter handles
    text = re.sub(r'http\S+', '', text)  # Remove URLs
    text = re.sub(r'pic.\S+', '', text)
    text = re.sub(r"[^a-zA-Z+']", ' ', text)  # Keep only characters
    text = re.sub(r'\s+[a-zA-Z]\s+', ' ', text + ' ')  # Keep words with length > 1 only
    text = "".join([i for i in text if i not in string.punctuation])
    words = nltk.word_tokenize(text)
    stopwords = nltk.corpus.stopwords.words('english')  # Remove stopwords
    text = " ".join([i for i in words if i not in stopwords and len(i) > 2])
    text = re.sub("\s[\s]+", " ", text).strip()  # Remove repeated/leading/trailing spaces
    return text
```

Apply the clean_text function to the 'Description' column in the training and test data

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

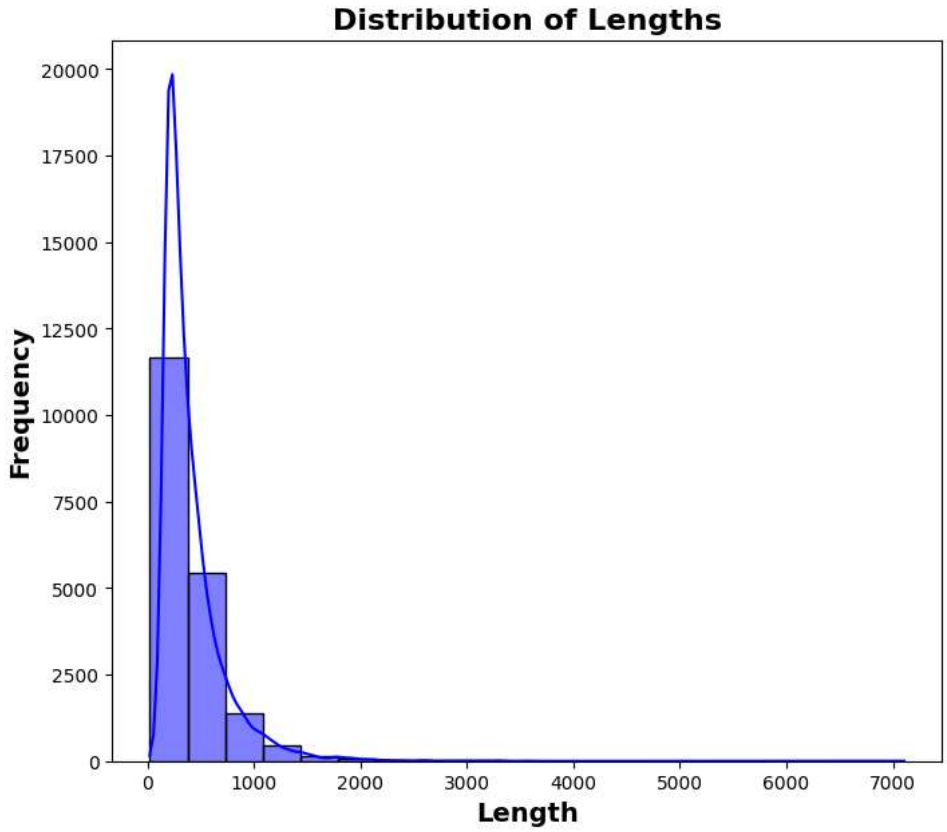
```
train_data['Text_cleaning'] = train_data['Description'].apply(clean_text)
test_data['Text_cleaning'] = test_data['Description'].apply(clean_text)
```

Calculate the length of cleaned text

```
train_data['length_Text_cleaning'] = train_data['Text_cleaning'].apply(len)
```

Visualize the distribution of text lengths

```
plt.figure(figsize=(8, 7))
sns.histplot(data=train_data, x='length_Text_cleaning', bins=20, kde=True, color='blue')
plt.xlabel('Length', fontsize=14, fontweight='bold')
plt.ylabel('Frequency', fontsize=14, fontweight='bold')
plt.title('Distribution of Lengths', fontsize=16, fontweight='bold')
plt.show()
```



Initialize the TF-IDF vectorizer

```
tfidf_vectorizer = TfidfVectorizer()
```

Fit and transform the training data

```
X_train = tfidf_vectorizer.fit_transform(train_data['Text_cleaning'])
```

Transform the test data

```
X_test = tfidf_vectorizer.transform(test_data['Text_cleaning'])
```

```
# Split the data into training and validation sets
X = X_train
y = train_data['Genre']
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train a Multinomial Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train, y_train)

# Make predictions on the validation set
y_pred = classifier.predict(X_val)

# Evaluate the performance of the model
accuracy = accuracy_score(y_val, y_pred)
print("Validation Accuracy:", accuracy)
print(classification_report(y_val, y_pred))
```

```
Validation Accuracy: 0.4351224596143825
               precision    recall  f1-score   support

      action        0.00      0.00      0.00        80
       adult        0.00      0.00      0.00        30
   adventure        0.00      0.00      0.00        58
   animation        0.00      0.00      0.00        26
   biography        0.00      0.00      0.00        17
      comedy        0.67      0.02      0.03       526
       crime        0.00      0.00      0.00        33
 documentary        0.54      0.88      0.67       968
       drama        0.36      0.86      0.51       941
      family        0.00      0.00      0.00        58
     fantasy        0.00      0.00      0.00        26
   game-show        0.00      0.00      0.00        18
     history        0.00      0.00      0.00        18
      horror        0.00      0.00      0.00       175
       music        0.00      0.00      0.00        53
     musical        0.00      0.00      0.00        28
     mystery        0.00      0.00      0.00        16
        news        0.00      0.00      0.00         8
  reality-tv        0.00      0.00      0.00        53
     romance        0.00      0.00      0.00        43
      sci-fi        0.00      0.00      0.00        42
       short        0.00      0.00      0.00       386
       sport        0.00      0.00      0.00        22
   talk-show        0.00      0.00      0.00        23
    thriller        0.00      0.00      0.00       115
         war        0.00      0.00      0.00         9
     western        0.00      0.00      0.00        66

    accuracy                           0.44      3838
   macro avg        0.06      0.07      0.04      3838
weighted avg        0.32      0.44      0.30      3838

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score a
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score a
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score a
  _warn_prf(average, modifier, msg_start, len(result))
```

Use the trained model to make predictions on the test data

```
X_test_predictions = classifier.predict(X_test)
test_data['Predicted_Genre'] = X_test_predictions
```

Save the test_data DataFrame with predicted genres to a CSV file

```
test_data.to_csv('predicted_genres.csv', index=False)
```

Display the 'test_data' DataFrame with predicted genres

```
print(test_data)
```

```
            Id                                     Title  \
0        1                       Edgar's Lunch (1998)
1        2                    La guerra de papá (1977)
2        3                 Off the Beaten Track (2010)
3        4                      Meu Amigo Hindu (2015)
4        5                           Er nu zhai (1955)
...    ...                                         ...
3255  3256                           The Doctor (1991)
3256  3257                            Space Men (1960)
3257  3258            Do You Know the Milkyway? (1985)
3258  3259                       The Collection (2005)
3259  3260   "Mutant: Leaving Humanity Behind" (2012)

                                           Description  \
0      L.R. Brane loves his life - his car, his apar...
1      Spain, March 1964: Quico is a very naughty ch...
2      One year in the life of Albin and his family ...
3      His father has died, he hasn't spoken with hi...
4      Before he was known internationally as a mart...
...                                                ...
3255   Jack McKee is a doctor with it all: he's succ...
3256   In the 22nd Century, Ray Peterson, reporter f...
3257   Kris, thought lost in the war, returns home t...
3258   Director Bruno de Almeida and a group of New ...
3259   Bodybuilder Rich Piana has become a sensation...

                                        Text_cleaning Predicted_Genre
```

```
0     brane loves life car apartment job especially ...          drama
1     spain march quico naughty child three belongin...          drama
2     one year life albin family shepherds north tra...    documentary
3     father died hasnt spoken brother years serious...          drama
4     known internationally martial arts superstar b...          drama
...                                                 ...            ...
3255  jack mckee doctor hes successful hes rich extr...          drama
3256  century ray peterson reporter interplanetary n...          drama
3257  kris thought lost war returns home discover to...          drama
3258  director bruno almeida group new york actors w...    documentary
3259  bodybuilder rich piana become sensation enormo...    documentary

[3260 rows x 5 columns]
```