```
pwd
```

```
'/content'
```

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
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, confusion_matrix, classification_report,r2_score
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files test_data.txt.zip

- **test_data.txt.zip**(application/x-zip-compressed) - 14485180 bytes, last modified: 6/16/2024 - 100% done

Saving test_data.txt.zip to test_data.txt.zip

```
from google.colab import files
uploaded = files.upload()
```

Choose Files train_data.txt.zip

- **train_data.txt.zip**(application/x-zip-compressed) - 14617856 bytes, last modified: 6/16/2024 - 100% done

Saving train_data.txt.zip to train_data.txt.zip

```
df_test = pd.read_csv("test_data.txt.zip", sep=":::", header=0, engine='python')
df_train = pd.read_csv("train_data.txt.zip", sep=":::", header=0, engine='python')
df_train.columns = ['SN', 'movie_name', 'category', 'confession']
df_test.columns = ['SN', 'movie_name' , 'confession']
```

```
df_test.head()
```

| | SN | movie_name | confession |
|---|---|---|---|
| 0 | 2 | La guerra de papá (1977) | Spain, March 1964: Quico is a very naughty ch... |
| 1 | 3 | Off the Beaten Track (2010) | One year in the life of Albin and his family ... |
| 2 | 4 | Meu Amigo Hindu (2015) | His father has died, he hasn't spoken with hi... |
| 3 | 5 | Er nu zhai (1955) | Before he was known internationally as a mart... |
| 4 | 6 | Riddle Room (2016) | Emily Burns is being held captive in a room w... |

Next steps: Generate code with df_test View recommended plots

```
df_train.head()
```

| | SN | movie_name | category | confession |
|---|---|---|---|---|
| 0 | 2 | Cupid (1997) | thriller | A brother and sister with a past incestuous r... |
| 1 | 3 | Young, Wild and Wonderful (1980) | adult | As the bus empties the students for their fie... |
| 2 | 4 | The Secret Sin (1915) | drama | To help their unemployed father make ends mee... |
| 3 | 5 | The Homecoming (2007) | drama | The film's title refers not only to the un- |

Next steps: Generate code with df_train View recommended plots

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54213 entries, 0 to 54212
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   SN          54213 non-null  int64
 1   movie_name  54213 non-null  object
 2   category    54213 non-null  object
 3   confession  54213 non-null  object
dtypes: int64(1), object(3)
memory usage: 1.7+ MB
```

```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54199 entries, 0 to 54198
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   SN          54199 non-null  int64
 1   movie_name  54199 non-null  object
 2   confession  54199 non-null  object
dtypes: int64(1), object(2)
memory usage: 1.2+ MB
```

```
df_train.describe()
```

| | SN |
|---|---|
| **count** | 54213.000000 |
| **mean** | 27108.000000 |
| **std** | 15650.089409 |
| **min** | 2.000000 |
| **25%** | 13555.000000 |
| **50%** | 27108.000000 |
| **75%** | 40661.000000 |
| **max** | 54214.000000 |

```
df_test.describe()
```

| | SN |
|---|---|
| **count** | 54199.000000 |
| **mean** | 27101.000000 |
| **std** | 15646.047957 |
| **min** | 2.000000 |
| **25%** | 13551.500000 |
| **50%** | 27101.000000 |
| **75%** | 40650.500000 |
| **max** | 54200.000000 |

```
df_test.isnull().sum()
```

```
SN            0
movie_name    0
confession    0
dtype: int64
```

```
df_train.isnull().sum()
```

```
SN            0
movie_name    0
category      0
confession    0
dtype: int64
```

```
df_train.count()
```

```
SN            54213
movie_name    54213
category      54213
confession    54213
dtype: int64
```



```
df_test.count()
```

```
SN            54199
movie_name    54199
confession    54199
dtype: int64
```

Generate 10 random numbers using numpy Close

Generate is available for a limited time for unsubscribed users. Upgrade to Colab Pro

```
df_train.iloc[0:3]
```

| | SN | movie_name | category | confession |
|---|---|---|---|---|
| **0** | 2 | Cupid (1997) | thriller | A brother and sister with a past incestuous r... |
| **1** | 3 | Young, Wild and Wonderful (1980) | adult | As the bus empties the students for their fie... |

```
df_train.loc[0]
```

```
SN                                                    2
movie_name                                  Cupid (1997)
category                                        thriller
confession     A brother and sister with a past incestuous r...
Name: 0, dtype: object
```

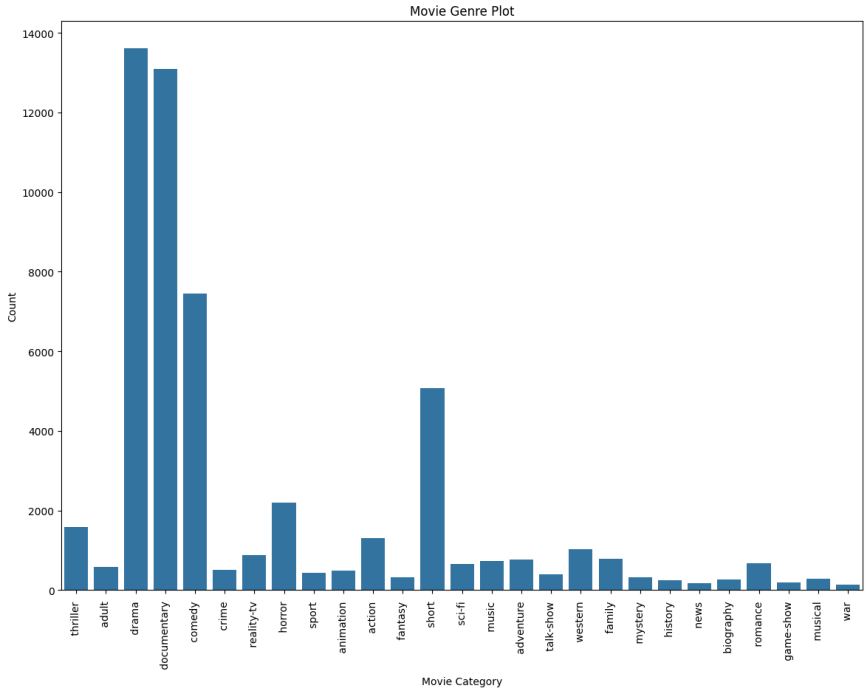
```
df_test.shape
```

```
(54199, 3)
```

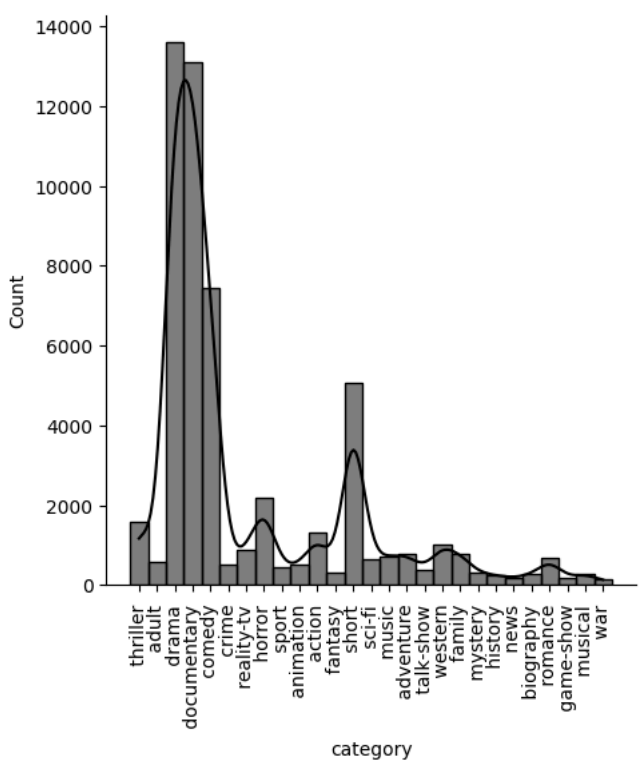
```
df_train.shape
```

```
(54213, 4)
```

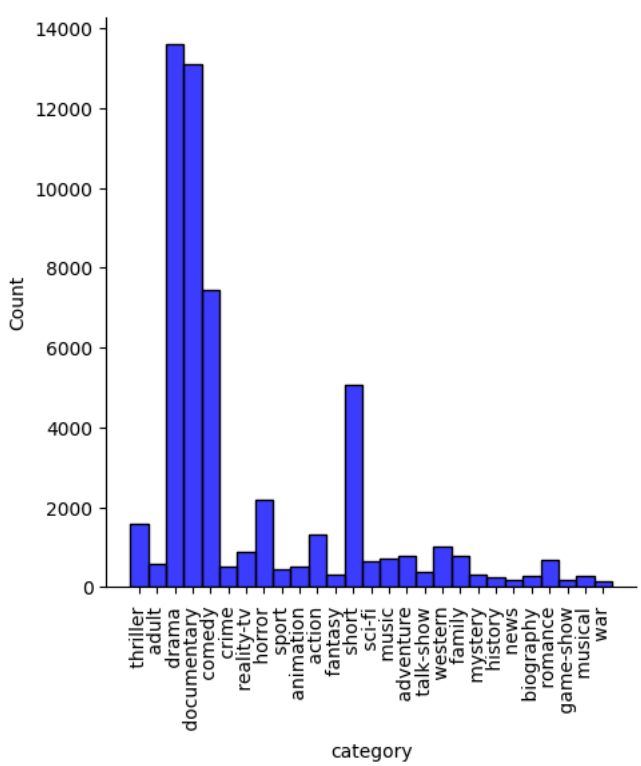
```
plt.figure(figsize=(14,10))
sns.countplot(x='category', data=df_train)
plt.xlabel('Movie Category')
plt.ylabel('Count')
plt.title('Movie Genre Plot')
plt.xticks(rotation=90);
plt.show()
```



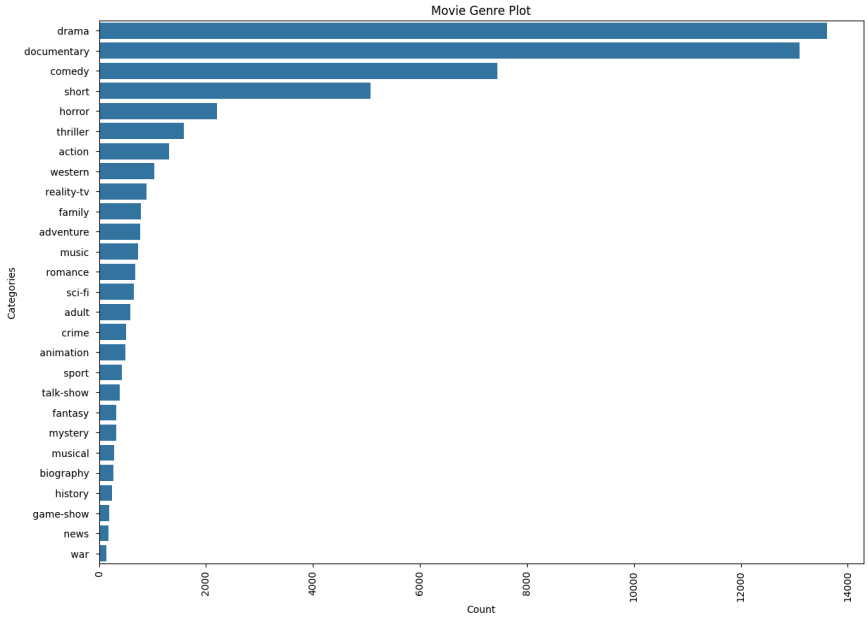

```
sns.displot(df_train.category, kde =True, color = "black")
plt.xticks(rotation=90);
```



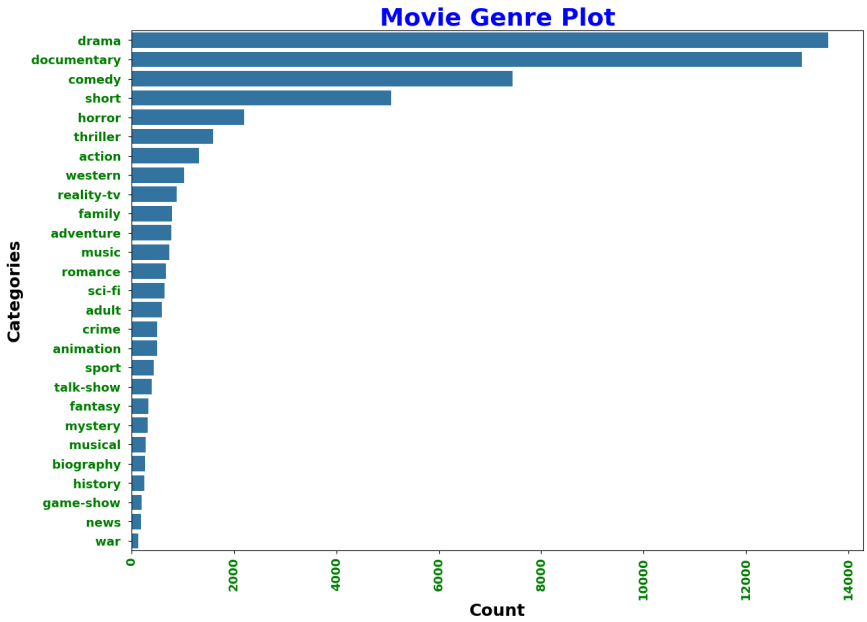

```
sns.displot(df_train.category, kde=False, color = "blue")
plt.xticks(rotation=90);
```




```
plt.figure(figsize = (14,10))
count1 = df_train.category.value_counts()
sns.barplot(x = count1, y = count1.index, orient = 'h')
plt.xlabel('Count')
plt.ylabel('Categories')
plt.title('Movie Genre Plot')
plt.xticks(rotation=90)
plt.show()
```




```
plt.figure(figsize = (14,10))
count1 = df_train.category.value_counts()
sns.barplot(x = count1, y = count1.index, orient = 'h')
plt.xlabel('Count', fontsize = 18, fontweight = 'bold')
plt.ylabel('Categories', fontsize = 18, fontweight = 'bold')
plt.title('Movie Genre Plot', fontsize = 26, fontweight = 'bold', color = 'blue')
plt.xticks(rotation=90, fontsize = 13, fontweight = 'bold', color = 'green')
plt.yticks(fontsize = 13, fontweight = 'bold', color = 'green')
plt.show()
```




```
df_combined = pd.concat([df_train, df_test], axis = 0)
df_combined.head()
```

| | SN | movie_name | category | confession |
|---|---|---|---|---|
| **0** | 2 | Cupid (1997) | thriller | A brother and sister with a past incestuous r... |
| **1** | 3 | Young, Wild and Wonderful (1980) | adult | As the bus empties the students for their fie... |
| **2** | 4 | The Secret Sin (1915) | drama | To help their unemployed father make ends mee... |
| **3** | 5 | The Unrecovered (2007) | drama | The film's title refers not only to the un- |

```
df_combined.shape
```

```
(108412, 4)
```

```
df_combined.size
```

```
433648
```

```
df_combined.isnull().any()
```

```
SN            False
movie_name    False
category       True
confession    False
dtype: bool
```

```
df_combined.count()
```

```
SN            108412
movie_name    108412
category       54213
confession    108412
dtype: int64
```

```
encoder = LabelEncoder()
df_combined["category"] = encoder.fit_transform(df_combined["category"].values)
```

```
encoder = LabelEncoder()
df_combined["movie_name"] = encoder.fit_transform(df_combined["movie_name"].values)
```

```
df_combined.head()
```

| | SN | movie_name | category | confession |
|---|---|---|---|---|
| **0** | 2 | 31219 | 24 | A brother and sister with a past incestuous r... |
| **1** | 3 | 107506 | 1 | As the bus empties the students for their fie... |
| **2** | 4 | 96119 | 8 | To help their unemployed father make ends mee... |
| **3** | 5 | 97557 | 8 | The film's title refers not only to the un-re... |
| **4** | 6 | 74516 | 7 | Quality Control consists of a series of 16mm ... |

```
df_combined.category = df_combined.category.fillna(df_combined.category.mean())
```

```
df_combined.count()
```

```
SN            108412
movie_name    108412
category      108412
confession    108412
dtype: int64
```

```
df_combined.duplicated().values.any()
```

```
False
```

## Data Processing

```
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df_combined["confession"])
df_combined.head()
```

| | SN | movie_name | category | confession |
|---|---|---|---|---|
| **0** | 2 | 31219 | 24 | A brother and sister with a past incestuous r... |
| **1** | 3 | 107506 | 1 | As the bus empties the students for their fie... |
| **2** | 4 | 96119 | 8 | To help their unemployed father make ends mee... |
| **3** | 5 | 97557 | 8 | The film's title refers not only to the un-re... |
| **4** | 6 | 74516 | 7 | Quality Control consists of a series of 16mm ... |

```
y = df_combined["category"]
```

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

## Naive Bayes Classifier

```
naive_bayes_model = MultinomialNB()
naive_bayes_model.fit(X_train, y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
nb_predictions = naive_bayes_model.predict(X_test)
print("Naive Bayes Model:")
print(confusion_matrix(y_test, nb_predictions))
print(classification_report(y_test, nb_predictions))
print("Accuracy: ", accuracy_score(y_test, nb_predictions))
print("r2_Score: ", r2_score(y_test, nb_predictions))
```

```
        0     0     0     0     0     0     0     0     0     0     0
        0     0     0    86]
[       0     0     0     0     0     1     0     0     1     0     0     0
        0     0     0     0     0     0     0     0     0     0     0     0
        0     0     0   319]
[       0     0     0     0     0     0     0     0     0     0     0     0
        0     0     0     0     0     0     0     0     0     0     0     0
        0     0     0    35]
[       0     0     0     0     0     0     0     0     0     0     0     0
        0     0     0     0     0     0     0     0     0     0     0     0
        0     0     0   218]
[       0     0     0     0     0    41     0     0    10     0     0     0
        0     0     0     0     0     0     0     0     0     0     0     0
        0     0     0 10683]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       256
           1       0.00      0.00      0.00       127
           2       0.00      0.00      0.00       146
           3       0.00      0.00      0.00        91
           4       0.00      0.00      0.00        42
           5       0.23      0.01      0.02      1488
           6       0.00      0.00      0.00        96
           7       1.00      0.00      0.00      2666
           8       0.20      0.00      0.00      2777
           9       0.00      0.00      0.00       151
          10       0.00      0.00      0.00        70
          11       0.00      0.00      0.00        51
          12       0.00      0.00      0.00        44
          13       0.00      0.00      0.00       480
          14       0.00      0.00      0.00       131
          15       0.00      0.00      0.00        49
          16       0.00      0.00      0.00        73
          17       0.00      0.00      0.00        44
          18       0.00      0.00      0.00       173
          19       0.00      0.00      0.00       158
          20       0.00      0.00      0.00       118
          21       0.00      0.00      0.00       961
          22       0.00      0.00      0.00        97
          23       0.00      0.00      0.00        86
          24       0.00      0.00      0.00       321
          25       0.00      0.00      0.00        35
          26       0.00      0.00      0.00       218
          27       0.49      1.00      0.66     10734

    accuracy                           0.49     21683
   macro avg       0.07      0.04      0.02     21683
weighted avg       0.41      0.49      0.33     21683

Accuracy:  0.49370474565327677
r2_Score:  -0.7967143521586344
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score ar
  _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 ar
  _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 ar
  _warn_prf(average, modifier, msg_start, len(result))
```

## Logistic Regression Model

```
logistic_regression_model = LogisticRegression()
logistic_regression_model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```