untitled6-1

August 5, 2024

[ ]:
```
import pandas as pd
import numpy as np
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 sklearn.preprocessing import StandardScaler as sc
from sklearn.svm import LinearSVC
```

[ ]:
```
df = pd.read_csv("/content/train_data.txt",sep=":::
 ↪",names=['NO','MOVIE_NAME','GENRE','DESCRIPTION'])
```

```
<ipython-input-37-b1a88c6dadca>:1: ParserWarning: Falling back to the 'python'
engine because the 'c' engine does not support regex separators (separators > 1
char and different from '\s+' are interpreted as regex); you can avoid this
warning by specifying engine='python'.
  df = pd.read_csv("/content/train_data.txt",sep=":::",names=['NO','MOVIE_NAME',
'GENRE','DESCRIPTION'])
```

[ ]:
```
print(df)
```

```
          NO                                  MOVIE_NAME          GENRE  \
0          1               Oscar et la dame rose (2009)          drama
1          2                           Cupid (1997)       thriller
2          3        Young, Wild and Wonderful (1980)          adult
3          4                  The Secret Sin (1915)          drama
4          5                  The Unrecovered (2007)          drama
…        …                                      …              …
54209  54210                       "Bonino" (1953)         comedy
54210  54211            Dead Girls Don't Cry (????)         horror
54211  54212  Ronald Goedemondt: Ze bestaan echt (2008)    documentary
54212  54213                Make Your Own Bed (1944)         comedy
54213  54214  Nature's Fury: Storm of the Century (2006)        history
```

```
                                                DESCRIPTION
0       Listening in to a conversation between his do…
1       A brother and sister with a past incestuous r…
2       As the bus empties the students for their fie…
3       To help their unemployed father make ends mee…
4       The film's title refers not only to the un-re…
…                                                     …
54209   This short-lived NBC live sitcom centered on …
54210   The NEXT Generation of EXPLOITATION. The sist…
54211   Ze bestaan echt, is a stand-up comedy about g…
54212   Walter and Vivian live in the country and hav…
54213   On Labor Day Weekend, 1935, the most intense …

[54214 rows x 4 columns]
```

[ ]:
```
test_data=pd.read_csv("/content/test_data.txt",sep=":::
↪",names=['NO','MOVIE_NAME','DESCRIPTION'])
```

```
<ipython-input-22-083b3032d583>:1: ParserWarning: Falling back to the 'python'
engine because the 'c' engine does not support regex separators (separators > 1
char and different from '\s+' are interpreted as regex); you can avoid this
warning by specifying engine='python'.
  test_data=pd.read_csv("/content/test_data.txt",sep=":::",names=['NO','MOVIE_NA
ME','DESCRIPTION'])
```

[ ]:
```
test_data_solution=pd.read_csv("/content/test_data_solution.txt",sep=":::
↪",names=['NO','MOVIE_NAME','GENRE','DESCRIPTION'])
```

```
<ipython-input-39-439b5a6c1fa0>:1: ParserWarning: Falling back to the 'python'
engine because the 'c' engine does not support regex separators (separators > 1
char and different from '\s+' are interpreted as regex); you can avoid this
warning by specifying engine='python'.
  test_data_solution=pd.read_csv("/content/test_data_solution.txt",sep=":::",nam
es=['NO','MOVIE_NAME','GENRE','DESCRIPTION'])
```

[ ]:
```
print(test_data)
```

```
          NO                          MOVIE_NAME  \
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)
…      …                                    …
54195  54196   "Tales of Light & Dark" (2013)
54196  54197      Der letzte Mohikaner (1965)
54197  54198               Oliver Twink (2007)
```

```
54198  54199                Slipstream (1973)
54199  54200        Curitiba Zero Grau (2010)

                                          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…
…                                                     …
54195   Covering multiple genres, Tales of Light & Da…
54196   As Alice and Cora Munro attempt to find their…
54197   A movie 169 years in the making. Oliver Twist…
54198   Popular, but mysterious rock D.J Mike Mallard…
54199   Curitiba is a city in movement, with rhythms …

[54200 rows x 3 columns]
```

[ ]:
```
print(test_data_solution)
```

```
          NO                     MOVIE_NAME          GENRE  \
0          1          Edgar's Lunch (1998)       thriller
1          2      La guerra de papá (1977)         comedy
2          3   Off the Beaten Track (2010)    documentary
3          4        Meu Amigo Hindu (2015)          drama
4          5             Er nu zhai (1955)          drama
…      …                             …             …
40123  40124                 'Doc' (1971)        western
40124  40125              "AN.X.O" (2015)         action
40125  40126   Bachke Rehna Re Baba (2005)         comedy
40126  40127         Sukeban gerira (1972)         action
40127  40128                 Na dne (2014)          drama

                                          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…
…                                                     …
40123   One night of 1881, Doc Holliday, a famous pok…
40124   AN.X.O is a web series with 8 minute episodes…
40125   Rukmini is well into her early 40s and is sti…
40126   Miko Sugimoto is the leader of the Red Helmet…
40127   More than a century ago a play about "former …

[40128 rows x 4 columns]
```

[ ]:

TRAINING AND TESTING OF DATA

[ ]:
```
df['DESCRIPTION'].fillna("",inplace=True)
```

[ ]:
```
print(df['DESCRIPTION'])
```

```
0         Listening in to a conversation between his do…
1         A brother and sister with a past incestuous r…
2         As the bus empties the students for their fie…
3         To help their unemployed father make ends mee…
4         The film's title refers not only to the un-re…
                              …
54209     This short-lived NBC live sitcom centered on …
54210     The NEXT Generation of EXPLOITATION. The sist…
54211     Ze bestaan echt, is a stand-up comedy about g…
54212     Walter and Vivian live in the country and hav…
54213     On Labor Day Weekend, 1935, the most intense …
Name: DESCRIPTION, Length: 54214, dtype: object
```

[ ]:
```
test_data['DESCRIPTION'].fillna("",inplace=True)
```

[ ]:
```
print(test_data["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…
                              …
54195     Covering multiple genres, Tales of Light & Da…
54196     As Alice and Cora Munro attempt to find their…
54197     A movie 169 years in the making. Oliver Twist…
54198     Popular, but mysterious rock D.J Mike Mallard…
54199     Curitiba is a city in movement, with rhythms …
Name: DESCRIPTION, Length: 54200, dtype: object
```

[ ]:
```
vectorizer=TfidfVectorizer(stop_words='english',max_features=1000)
```

[ ]:
```
X_train=vectorizer.fit_transform(df["DESCRIPTION"])
X_test=vectorizer.fit_transform(test_data["DESCRIPTION"])
```

[ ]:
```
le=LabelEncoder()
Y_train=le.fit_transform(df["GENRE"])
Y_test=le.transform(test_data_solution["GENRE"])
```

[ ]:
```
x_train,x_test,y_train,y_test=train_test_split(X_train,Y_train,test_size=0.
↪2,random_state=42)
```

[ ]:
```
v=LinearSVC()
v.fit(x_train,y_train)
prediction=v.predict(x_test)
print("accuracy:",accuracy_score(y_test,prediction))
print("classification report:",classification_report(y_test,prediction))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/svm/_classes.py:32:
FutureWarning: The default value of `dual` will change from `True` to `'auto'`
in 1.5. Set the value of `dual` explicitly to suppress the warning.
  warnings.warn(

accuracy: 0.5391496818223739
classification report:                precision    recall  f1-score   support

           0       0.35      0.18      0.24       263
           1       0.42      0.27      0.33       112
           2       0.27      0.09      0.13       139
           3       0.25      0.11      0.15       104
           4       0.00      0.00      0.00        61
           5       0.46      0.47      0.46      1443
           6       0.29      0.06      0.09       107
           7       0.65      0.82      0.72      2659
           8       0.53      0.73      0.61      2697
           9       0.42      0.12      0.19       150
          10       0.10      0.01      0.02        74
          11       0.56      0.50      0.53        40
          12       0.00      0.00      0.00        45
          13       0.49      0.55      0.52       431
          14       0.50      0.45      0.47       144
          15       0.40      0.08      0.13        50
          16       0.15      0.04      0.06        56
          17       0.43      0.09      0.15        34
          18       0.46      0.23      0.31       192
          19       0.22      0.01      0.03       151
          20       0.36      0.22      0.28       143
          21       0.43      0.24      0.31      1045
          22       0.40      0.27      0.32        93
          23       0.39      0.20      0.26        81
          24       0.29      0.07      0.11       309
          25       0.33      0.10      0.15        20
          26       0.68      0.73      0.70       200

    accuracy                           0.54     10843
   macro avg       0.36      0.25      0.27     10843
weighted avg       0.50      0.54      0.50     10843
```

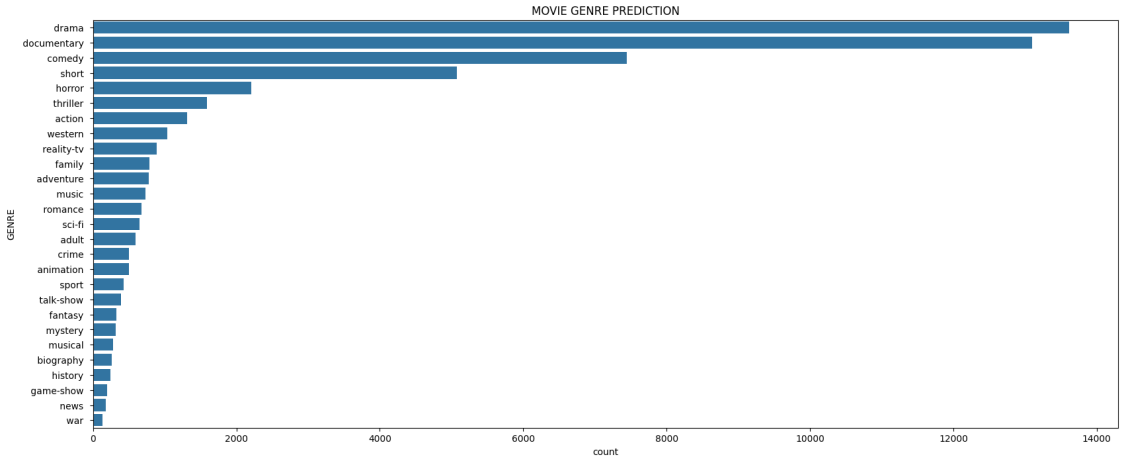
[ ]:
```
naive=MultinomialNB()
naive.fit(X_train,Y_train)
```

[ ]:
```
MultinomialNB()
```

[ ]:
```
naive.predict(X_test)
```

[ ]:
```
array([8, 8, 7, …, 8, 8, 7])
```

[ ]:
```
plt.figure(figsize=(20,8))
sns.countplot(y=df["GENRE"],order=df["GENRE"].value_counts().index)
plt.title("MOVIE GENRE PREDICTION")
plt.xlabel=("x-axis")
plt.ylabel=("y-axis")
```



[ ]:
```
def movie_pred(description):
  vectorizer_1=vectorizer.transform([description])
  prediction=v.predict(vectorizer_1)
  return le.inverse_transform(prediction)[0]
sample="A group of friends accidentally mix up their vacation plans and end up␣
 ↪at a tropical resort meant for retirees. Hilarity ensues as they try to fit␣
 ↪in with the elderly guests while navigating romantic mishaps and slapstick␣
 ↪antics"
print(movie_pred(sample))
```

```
comedy
```