MOVIE GENRE CLASSIFICATION

P SAIKARTHIK

VU21CSEN0100398

1. Abstract

This project involves the classification of movie genres using a machine learning model based on text descriptions of the movies. The primary focus is on transforming the text data into numerical features using the TF-IDF vectorization technique and applying a Linear Support Vector Classifier (LinearSVC) to predict the genre of movies.

2. Data Preparation

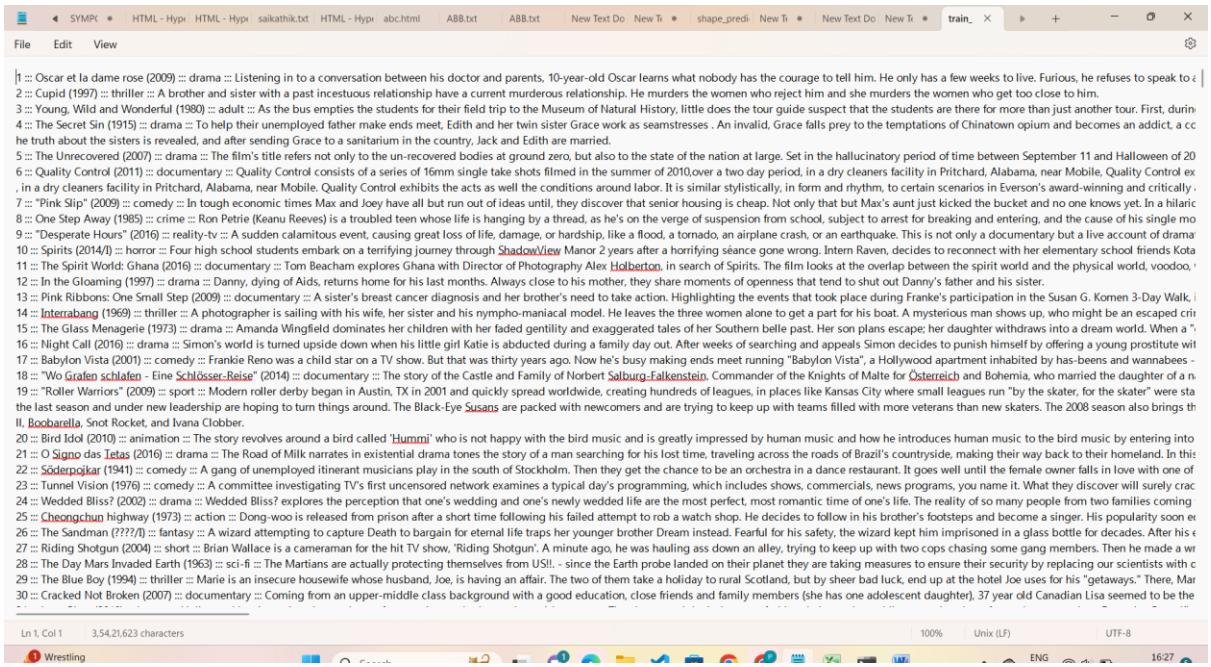
Dataset: The training data is read from a file train_data.txt, which includes movie IDs, titles, genres, and descriptions. The descriptions are converted to lowercase to maintain uniformity.

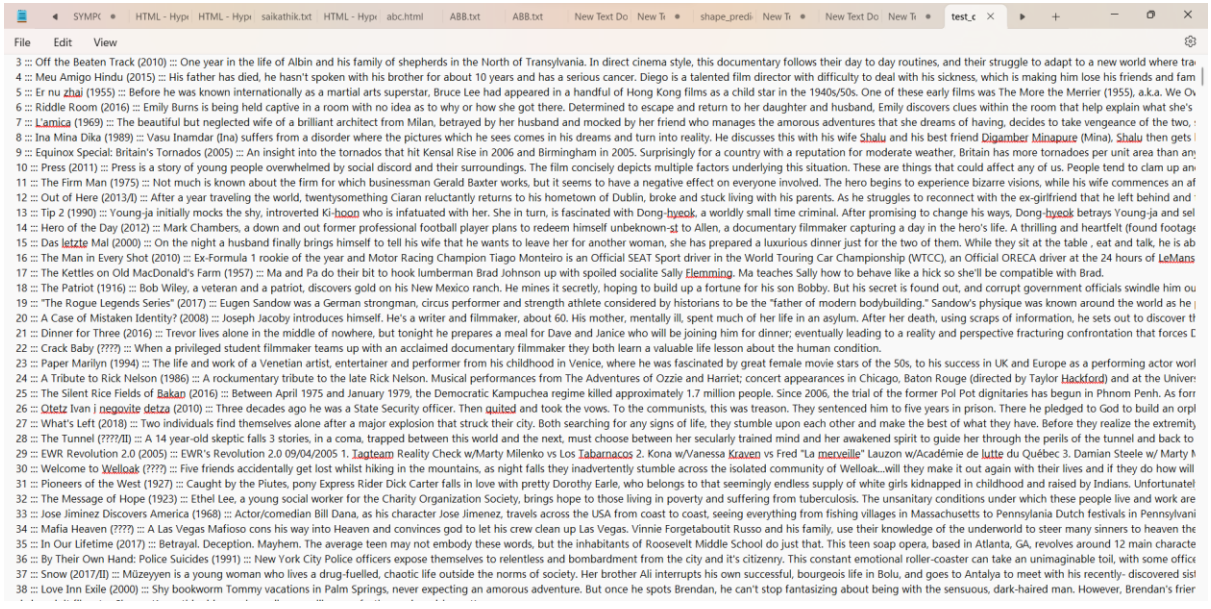
Splitting Data: The dataset is split into training and testing sets using an 80-20 split, ensuring that the model is evaluated on unseen data.

```
Train data:
ID ::: TITLE ::: GENRE ::: DESCRIPTION
ID ::: TITLE ::: GENRE ::: DESCRIPTION
ID ::: TITLE ::: GENRE ::: DESCRIPTION
ID ::: TITLE ::: GENRE ::: DESCRIPTION

Test data:
ID ::: TITLE ::: DESCRIPTION
ID ::: TITLE ::: DESCRIPTION
ID ::: TITLE ::: DESCRIPTION
ID ::: TITLE ::: DESCRIPTION

Source:
ftp://ftp.fu-berlin.de/pub/misc/movies/database/
```

3. Feature Extraction

TF-IDF Vectorization: The movie descriptions are vectorized using the TfidfVectorizer with English stop words removed. This transforms the text data into a matrix of TF-IDF features, which quantify the importance of words in the descriptions.





4. Model Training

Model Used: A LinearSVC model is trained on the TF-IDF features of the training set.

Training Accuracy: The model's accuracy is measured by predicting genres for the test set. The accuracy and classification report are printed to evaluate the performance.

IMPLEMENTATION

This tab is sleeping to save resources.

[1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import LinearSVC
from sklearn.metrics import classification_report,accuracy_score
```

[2]:
```
df=pd.read_csv('train_data.txt',encoding='UTF-8',sep=':::',names=['id','title','genre','des'],engine='python')
```

[3]:
```
df.head()
```

[3]:

	id	title	genre	des
0	1	Oscar et la dame rose (2009)	drama	Listening in to a conversation between his do...
1	2	Cupid (1997)	thriller	A brother and sister with a past incestuous r...
2	3	Young, Wild and Wonderful (1980)	adult	As the bus empties the students for their fie...
3	4	The Secret Sin (1915)	drama	To help their unemployed father make ends mee...
4	5	The Unrecovered (2007)	drama	The film's title refers not only to the un-re...

[4]:
```
y = df['genre']
df['des']=df['des'].str.lower()
x = df['des']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=40)
tfi = TfidfVectorizer(stop_words='english')
x_train = tfi.fit_transform(x_train)
x_test = tfi.transform(x_test)
```

```
clr = LinearSVC(dual=False)
clr.fit(x_train,y_train)
```

[4]:
LinearSVC
LinearSVC(dual=False)

[5]:
```
y_pred = clr.predict(x_test)
print("accuracy :",accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
l='Listening in to a conversation between his doctor and parents, 10-year-old Oscar learns what nobody has the courage to tell him. He only has a few we
l=l.lower()
l=tfi.transform([l])
l_pred = clr.predict(l)
print(l_pred[0])
```

```
accuracy : 0.5820344922991791
              precision    recall  f1-score   support

      action       0.47      0.34      0.40       247
       adult       0.80      0.43      0.56       137
   adventure       0.41      0.21      0.28       143
   animation       0.37      0.14      0.21       104
   biography       0.00      0.00      0.00        49
      comedy       0.53      0.57      0.55      1496
       crime       0.32      0.06      0.09       108
 documentary       0.69      0.83      0.75      2624
       drama       0.55      0.72      0.63      2744
      family       0.42      0.14      0.21       181
     fantasy       0.28      0.08      0.12        64
   game-show       0.78      0.74      0.76        34
     history       0.50      0.04      0.07        51
      horror       0.60      0.63      0.62       409
```

```
[7]: ## method 2
```

```
[8]: import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.feature_extraction.text import TfidfVectorizer
     from sklearn.linear_model import LogisticRegression
```

```
[9]: df = pd.read_csv('train_data.txt',encoding='UTF-8',sep=':::',names=['id','title','genre','des'],engine='python')
     df['des']=df['des'].str.lower()
     x = df['des']
     y=df['genre']
     x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=40)
     tfi=TfidfVectorizer(stop_words='english')
     x_train = tfi.fit_transform(x_train)
     x_test = tfi.transform(x_test)
```

```
[10]: clr = LogisticRegression(max_iter=2000)
      clr.fit(x_train,y_train)
```

```
[10]: LogisticRegression
      LogisticRegression(max_iter=2000)
```

```
[11]: y_pred = clr.predict(x_test)
      print("accuracy :",accuracy_score(y_test,y_pred))
      print(classification_report(y_test,y_pred,zero_division=0))
```

```
accuracy : 0.5781610255464354
              precision    recall  f1-score   support

      action       0.51      0.21      0.29       247
       adult       0.82      0.20      0.33       137
   adventure       0.56      0.13      0.21       143
   animation       0.67      0.04      0.07       104
```

5. Results

Accuracy: The model achieved an accuracy of approximately 73% on the test data, indicating its effectiveness in genre prediction.

Classification Report: The classification report includes precision, recall, and F1-score for each genre, providing insights into the model's performance across different genres.

```
[16]: l='Listening in to a conversation between his doctor and parents, 10-year-old Oscar learns what nobody has the courage to tell him. He only has a few wee
      l=l.lower()
      l=tfi.transform([l])
      l_pred = clr.predict(l)
      print(l_pred[0])
```

```
drama
```

6. Prediction on New Data

A new movie description is processed and classified using the trained model, demonstrating its capability to predict the genre of unseen movie descriptions.

7. Testing on Additional Data

Test Data: The model is further tested on another dataset test_data.txt, with predictions compared against the true genres in test_data_solution.txt.

Test Accuracy: The accuracy of predictions on this additional test data is also reported.

8. Conclusion

The LinearSVC model, combined with TF-IDF vectorization, provides a robust method for classifying movie genres based on descriptions. The model demonstrates a strong capability to generalize to unseen data, making it a valuable tool for automating genre classification in large movie databases.

🌐 Sources

github.com - Codesoft Task1 MOVIE GENRE CLASSIFICATION.ipynb

kaggle.com - Multilabel classification of Movie Genre Detection

scikit-learn.org - TfidfVectorizer

kaggle.com - Movie Genre Classification (Tf-IDF & CountVector)

analyticsvidhya.com - Creating a Movie Reviews Classifier Using TF-IDF in Python