MOVIE GENRE CLASSIFICATION ---- (TASK 1 FOR MACHINE LEARNING)

FEB Batch P33 CODSOFT INTERNSHIP PROGRAM

Author: Irfan Ullah Khan

GITHUB PROFILE | KAGGLE PROFILE | LINKEDIN PROFILE

YOUTUBE PROFILE | EMAIL CONTACT ME | WEBSITE CONTACT ME

```
%matplotlib inline
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Load Dataset

```
df = pd.read_csv("movies_genres.csv", delimiter='\t')
```

Information about Dataset

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4869 entries, 0 to 4868
Data columns (total 30 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   title        4869 non-null   object 
 1   plot         4869 non-null   object 
 2   Action       4868 non-null   float64
 3   Adult        4868 non-null   float64
 4   Adventure    4868 non-null   float64
 5   Animation    4868 non-null   float64
 6   Biography    4868 non-null   float64
 7   Comedy       4868 non-null   float64
 8   Crime        4868 non-null   float64
 9   Documentary  4868 non-null   float64
 10  Drama        4868 non-null   float64
 11  Family       4868 non-null   float64
 12  Fantasy      4868 non-null   float64
 13  Game-Show    4868 non-null   float64
 14  History      4868 non-null   float64
 15  Horror       4868 non-null   float64
 16  Lifestyle    4868 non-null   float64
 17  Music        4868 non-null   float64
 18  Musical      4868 non-null   float64
 19  Mystery      4868 non-null   float64
 20  News         4868 non-null   float64
 21  Reality-TV   4868 non-null   float64
 22  Romance      4868 non-null   float64
 23  Sci-Fi       4868 non-null   float64
 24  Short        4868 non-null   float64
 25  Sport        4868 non-null   float64
 26  Talk-Show    4868 non-null   float64
 27  Thriller     4868 non-null   float64
 28  War          4868 non-null   float64
 29  Western      4868 non-null   float64
dtypes: float64(28), object(2)
memory usage: 1.1+ MB
```

Size of Dataset

```
df.shape
```

```
(4869, 30)
```

First 5 Rows of Dataset

```
df.head()
```

	title	plot	Action	Adult	Adventure	Animation	Biography	Comedy
0	"#7DaysLater" (2013)	#7dayslater is an interactive comedy series f...	0.0	0.0	0.0	0.0	0.0	1.C
1	"#BlackLove" (2015) {Crash the Party (#1.9)}	With just one week left in the workshops, the...	0.0	0.0	0.0	0.0	0.0	0.C
2	"#BlackLove" (2015) {Making	All of the women start making	0.0	0.0	0.0	0.0	0.0	0.C

Last 5 Rows of Dataset

```
df.tail()
```

	title	plot	Action	Adult	Adventure	Animation	Biography	Comedy
4864	"American Diner Revival" (2015) {Retro Remix (...	Help arrives for the "Red Hots Coney Island" ...	0.0	0.0	0.0	0.0	0.0	0.0
4865	"American Doers" (2016) {Katie Gong (#1.7)}	Everyone needs a place to firmly plant their ...	0.0	0.0	0.0	0.0	0.0	0.0

We have a total of 47403 movies and each of them is associated with 28 possible genres. The genres columns simply contain a 1 or 0 depending of wether the movie is classified into that particular genre or not, so the one-hot-enconding schema is alreay provided in this file.

Next we are going to calculate the absolute number of movies per genre

Note: each movie can be associated with more than one genre, we just want to know which genres have more movies.

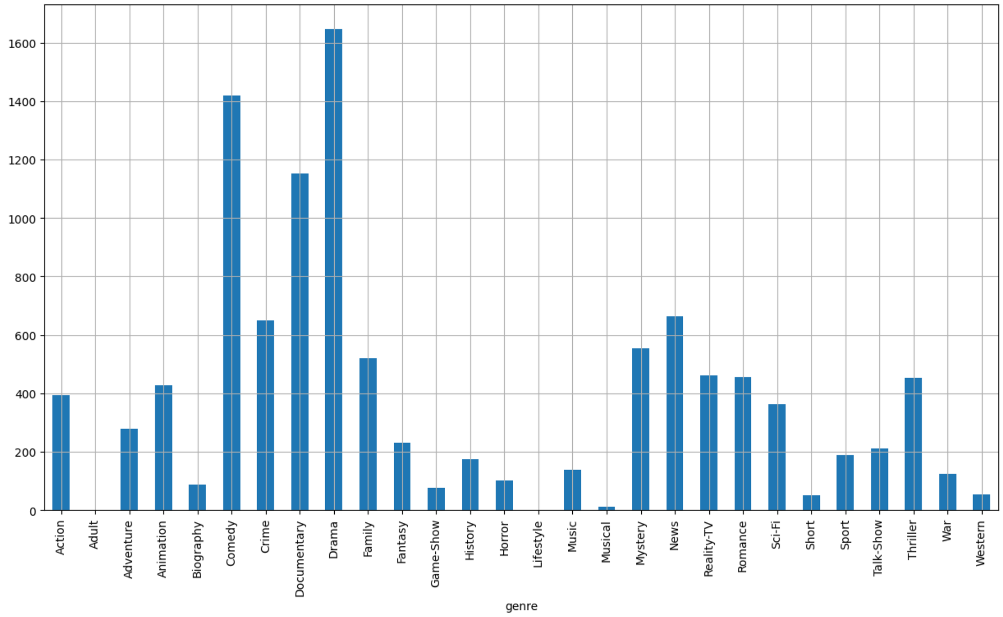
```
df_genres = df.drop(['plot', 'title'], axis=1)
counts = []
categories = list(df_genres.columns.values)
for i in categories:
    counts.append((i, df_genres[i].sum()))
df_stats = pd.DataFrame(counts, columns=['genre', '#movies'])
df_stats
```

	genre	#movies
0	Action	395.0
1	Adult	1.0
2	Adventure	278.0
3	Animation	429.0
4	Biography	88.0
5	Comedy	1418.0
6	Crime	650.0
7	Documentary	1153.0
8	Drama	1648.0
9	Family	521.0
10	Fantasy	231.0
11	Game-Show	76.0
12	History	176.0
13	Horror	102.0
14	Lifestyle	0.0
15	Music	137.0
16	Musical	12.0
17	Mystery	554.0
18	News	664.0
19	Reality-TV	461.0
20	Romance	456.0
21	Sci-Fi	362.0
22	Short	52.0
23	Sport	189.0
24	Talk-Show	210.0
25	Thriller	452.0
26	War	124.0
27	Western	54.0

Next steps: Generate code with `df_stats` View recommended plots

```
df_stats.plot(x='genre', y='#movies', kind='bar', legend=False, grid=True, figsize=(15, 8))
```

```
<Axes: xlabel='genre'>
```



Since the `Lifestyle` has 0 instances we can just remove it from the data set

```
df.drop('Lifestyle', axis=1, inplace=True)
```

One thing that notice when working with this dataset is that there are plots written in different languages. Let's use langedetect tool to identify the language in which the plots are written

```
!pip install langdetect
from langdetect import detect

df['plot_lang'] = df['plot'].apply(lambda text: detect(str(text)))
print(df['plot_lang'].value_counts())
```

```
Collecting langdetect
  Downloading langdetect-1.0.9.tar.gz (981 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 981.5/981.5 kB 11.2 MB/s eta 0:0
  Preparing metadata (setup.py) ... done
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from langdetect) (1.16.0)
Building wheels for collected packages: langdetect
  Building wheel for langdetect (setup.py) ... done
  Created wheel for langdetect: filename=langdetect-1.0.9-py3-none-any.whl size=993225 sha256=f9d59c1c905c27143
  Stored in directory: /root/.cache/pip/wheels/95/03/7d/59ea870c70ce4e5a370638b5462a7711ab78fba2f655d05106
Successfully built langdetect
Installing collected packages: langdetect
Successfully installed langdetect-1.0.9
en    4868
de       1
Name: plot_lang, dtype: int64
```

There other languages besides English, let's just keep English plots, and save this to a new file.

```
df = df[df.plot_lang.isin(['en'])]
df.to_csv("movies_genres_en.csv", sep='\t', encoding='utf-8')
```