data_loading_clean

December 4, 2019

In [2]:
```
# Import Necessary Libraries
import pandas as pd
import numpy as np
import json
import nltk
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
en_stop = set(nltk.corpus.stopwords.words('english'))
import re
import csv
import matplotlib.pyplot as plt
import seaborn as sns
from tqdm import tqdm

%matplotlib inline
pd.set_option('display.max_colwidth', 300)
```

0.1 Data Loading

0.1.1 Getting into required format

In [3]:
```
# Read in data files
```

In [4]:
```
# Metadata file :
colnames = ["MovieID","1","MovieName","3","4","5","6","7","Genre"]
movie_metadata = pd.read_csv("MovieSummaries/movie.metadata.tsv", names = colnames, sep
movie_metadata = movie_metadata.reindex(columns=["MovieID","1","MovieName","3","4","5"
movie_metadata.head()
```

Out[4]:

	MovieID	1	MovieName	3	\
0	975900	/m/03vyhn	Ghosts of Mars	2001-08-24	
1	3196793	/m/08yl5d			
2	28463795	/m/0crgdbh			
3	9363483	/m/0285_cd			
4	261236	/m/01mrr1			

```
1  Getting Away with Murder: The JonBenét Ramsey Mystery  2000-02-16
2                                          Brun bitter        1988
3                                       White Of The Eye        1987
4                                      A Woman in Flames        1983

             4      5                                      6  \
0  14010832.0   98.0  {"/m/02h40lc": "English Language"}
1         NaN   95.0  {"/m/02h40lc": "English Language"}
2         NaN   83.0  {"/m/05f_3": "Norwegian Language"}
3         NaN  110.0  {"/m/02h40lc": "English Language"}
4         NaN  106.0   {"/m/04306rv": "German Language"}

                                           7  \
0  {"/m/09c7w0": "United States of America"}
1  {"/m/09c7w0": "United States of America"}
2                      {"/m/05b4w": "Norway"}
3              {"/m/07ssc": "United Kingdom"}
4                     {"/m/0345h": "Germany"}


0  {"/m/01jfsb": "Thriller", "/m/06n90": "Science Fiction", "/m/03npn": "Horror", "/m/0
1                                                                                  {"/m/02r
2
3
4
```

In [5]:
```
# Extract distinct Genres from the Genre Column and update the column
# initiate an empty list to store extracted genre values
genres = []

# extract genres
for i in movie_metadata['Genre']:
    genres.append(list(json.loads(i).values()))

# update column in dataframe
movie_metadata['Genre'] = genres

# remove movies that have no genres assigned
movie_metadata_new = movie_metadata[~(movie_metadata['Genre'].str.len() == 0)]

# Convert datatype of column to be string
movie_metadata_new = movie_metadata.astype(str)
movie_metadata_new['Genre'] = movie_metadata['Genre']
movie_metadata_new.head(2)
```

Out[5]:
```
    MovieID          1                                                MovieName  \
0   975900   /m/03vyhn                                            Ghosts of Mars
1  3196793   /m/08yl5d  Getting Away with Murder: The JonBenét Ramsey Mystery
```

```
                3           4     5                                      6  \
0  2001-08-24  14010832.0  98.0  {"/m/02h40lc": "English Language"}
1  2000-02-16         nan  95.0  {"/m/02h40lc": "English Language"}

                                             7  \
0  {"/m/09c7w0": "United States of America"}
1  {"/m/09c7w0": "United States of America"}

                                                                                Genre
0  [Thriller, Science Fiction, Horror, Adventure, Supernatural, Action, Space western]
1                                    [Mystery, Biographical film, Drama, Crime Drama]
```

In [6]:
```
# Plot Summaries file :
# Initiate empty list to store plot summaries
plot_text = []

with open("MovieSummaries/plot_summaries.txt", 'r') as f:
    reader = csv.reader(f, dialect='excel-tab')
    for summary in tqdm(reader):
        plot_text.append(summary)
plot_text[0]
```

```
42303it [00:01, 40167.69it/s]
```

Out[6]:
```
['23890098',
 "Shlykov, a hard-working taxi driver and Lyosha, a saxophonist, develop a bizarre love
```

In [7]:
```
# Split the text obtained into Movie IDs and Movie Summaries
# Initiate empty list to store  movie Ids and plot summaries
movie_id = []
movie_sum = []

for i in tqdm(plot_text):
    movie_id.append(i[0])
    movie_sum.append(i[1])

# create dataframe
summaries = pd.DataFrame({"MovieID": movie_id, "Plot": movie_sum})
x = summaries.reindex(columns = ["MovieID","Plot"])
summaries = summaries.astype(str)
summaries.head(2)
```

```
100%|| 42303/42303 [00:00<00:00, 1388299.69it/s]
```

Out[7]:
```
     MovieID  \
0  23890098
```

```
1  31186339


0
1  The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As pu
```

In [8]:
```
# merge the metadata dataframe with the summaries dataframe
movies = summaries.merge(movie_metadata_new,on = "MovieID")
movies = movies[["MovieID","MovieName","Genre","Plot"]]
movies.head(5)
```

Out[8]:
```
     MovieID           MovieName  \
0  23890098          Taxi Blues
1  31186339    The Hunger Games
2  20663735          Narasimham
3   2231378  The Lemon Drop Kid
4    595909   A Cry in the Dark

                                                                Genre  \
0                                                [Drama, World cinema]
1                [Action/Adventure, Science Fiction, Action, Drama]
2                                [Musical, Action, Drama, Bollywood]
3                                          [Screwball comedy, Comedy]
4  [Crime Fiction, Drama, Docudrama, World cinema, Courtroom Drama]


0
1  The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As pu
2  Poovalli Induchoodan  is sentenced for six years prison life for murdering his class
3  The Lemon Drop Kid , a New York City swindler, is illegally touting horses at a Flo
4  Seventh-day Adventist Church pastor Michael Chamberlain, his wife Lindy, their two s
```

In [9]:
```
movies.shape
```

Out[9]:
```
(42204, 4)
```

0.2 Data Exploration : Summary Statistics

In [10]:
```
# Create a list of all genres
all_genres = sum(genres,[])
len(set(all_genres))
```

Out[10]:
```
363
```

In [11]:
```
# Create a dictionary of genres and their occurrence count across the dataset using n
genre_freq = nltk.FreqDist(all_genres)

# Create a dataframe to represent the frequency for each genre
# create dataframe
```

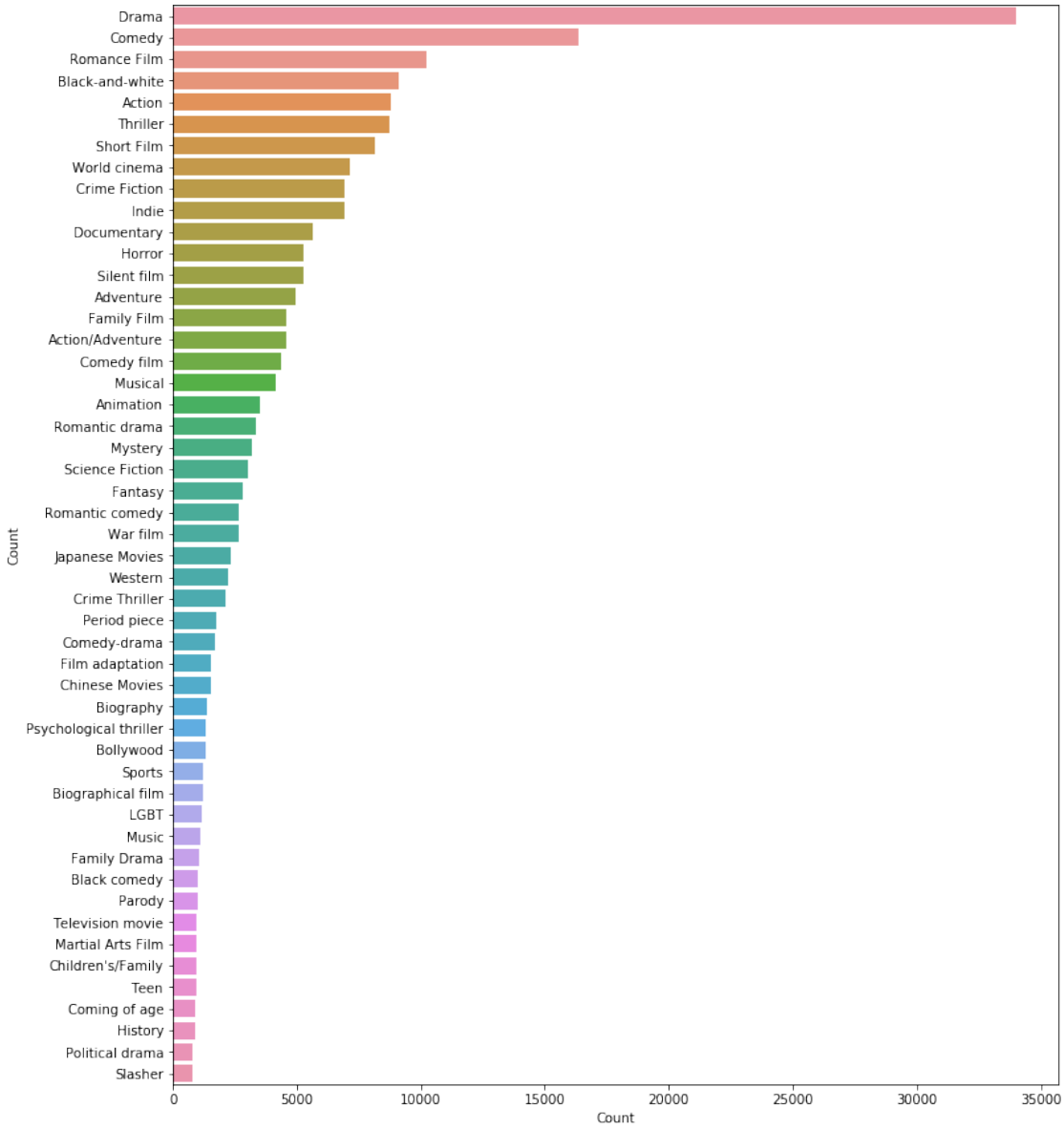
```
genre_freq_df = pd.DataFrame({'Genre': list(genre_freq.keys()),
                              'Count': list(genre_freq.values())})

# top 10 genres by frequency
genre_freq_df.sort_values("Count", ascending = False)[0:10]
```

Out[11]:

	Genre	Count
9	Drama	34007
18	Comedy	16349
29	Romance Film	10234
17	Black-and-white	9094
5	Action	8798
0	Thriller	8744
14	Short Film	8141
21	World cinema	7155
11	Crime Fiction	6948
16	Indie	6897

In [12]:
```
# Visualize the genre frequencies
freq = genre_freq_df.nlargest(columns="Count", n = 50)
plt.figure(figsize=(12,15))
ax = sns.barplot(data=freq, x= "Count", y = "Genre")
ax.set(ylabel = 'Count')
plt.show()
```



0.3 Subset Data to keep relevant genres

In [13]:
```
# Replacing all sub-types of comedy movie genres with the overarching general 'comedy
movies['Genre'] = movies['Genre'].apply(lambda x :  ['Comedy' if gen == 'Comedy film'
```

In [23]:
```
# subset dataframe by only selecting genres that appear in 98th percentile of the dat
percentiles = np.percentile(genre_freq_df['Count'], 98)
genre_selected_df = genre_freq_df[genre_freq_df['Count'] >= percentiles]
len(genre_selected_df)
```

Out[23]: 8

```
In [24]: percentiles

Out[24]: 7105.319999999998

In [25]: # Remove genres that donot satisfy the 98th percentile from the movies dataframe
         genres_to_remove = genre_freq_df[genre_freq_df['Count'] <= percentiles]['Genre'].to_li
         movies['Genre'] = movies['Genre'].apply(lambda x : [gen for gen in x if gen not in gen

         # Remove movies which do not belong to any of the selected genres i.e. length of genre
         movies['Num_Genres'] = movies['Genre'].apply(lambda x : len(x))
         movies = movies[movies['Num_Genres'] != 0]

In [26]: movies.shape

Out[26]: (35523, 6)

In [27]: movies.head(5)

Out[27]:     MovieID           MovieName                   Genre  \
         0  23890098          Taxi Blues  [Drama, World cinema]
         1  31186339    The Hunger Games        [Action, Drama]
         2  20663735          Narasimham        [Action, Drama]
         3   2231378  The Lemon Drop Kid       [Comedy, Comedy]
         4    595909   A Cry in the Dark  [Drama, World cinema]


         0
         1  The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As p
         2  Poovalli Induchoodan  is sentenced for six years prison life for murdering his clas
         3  The Lemon Drop Kid , a New York City swindler, is illegally touting horses at a Flo
         4  Seventh-day Adventist Church pastor Michael Chamberlain, his wife Lindy, their two


            Num_Genres  \
         0           2
         1           2
         2           2
         3           2
         4           2


         0
         1  nation panem consist wealthy capitol twelve poorer district punishment past rebelli
         2  poovalli induchoodan sentence six year prison life murder classmate induchoodan son
         3  lemon drop kid new york city swindler illegally tout horse florida racetrack severa
         4  seventh day adventist church pastor michael chamberlain wife lindy two son nine wee
```

0.4 Cleaning of text in plot summaries

```
In [28]: def preprocess_text(document):
```

```
        #now = datetime.datetime.now()

        # Remove all the special characters
        document = re.sub(r'\W', ' ', str(document))

        # remove all single characters
        document = re.sub(r'\s+[a-zA-Z]\s+', ' ', document)

        # Remove single characters from the start
        document = re.sub(r'\^[a-zA-Z]\s+', ' ', document)

        # Substituting multiple spaces with single space
        document = re.sub(r'\s+', ' ', document, flags=re.I)

        # Removing prefixed 'b'
        document = re.sub(r'^b\s+', '', document)

        # Converting to Lowercase
        document = document.lower()

        tokens = document.split()

        #### Remove stopwords
        words = [w for w in tokens if w not in stopwords.words('english')]
        words = [word for word in words if word not in en_stop]

        #### Lemmatize tokens obtained after removing stopwords
        wnl = WordNetLemmatizer()
        tagged = nltk.pos_tag(words)
        lem_list = []
        for word, tag in tagged:
            wntag = tag[0].lower()
            wntag = wntag if wntag in ['a', 'r', 'n', 'v'] else None
            if not wntag:
                lemma = word
            else:
                lemma = wnl.lemmatize(word, wntag)
            lem_list.append(lemma)

        preprocessed_text = ' '.join(lem_list)
        #lem_text = " ".join(lemma for lemma in lem_list)
        #print("Took %s"%(datetime.datetime.now()-now))

        return preprocessed_text, lem_list
```

In [29]:
```
# Clean all plot text summaries and append as a new column
movies['clean_plot_text'] = movies['Plot'].apply(lambda x: preprocess_text(x)[0])
```

In [ ]:
```
movies['clean_plot_tokens'] = movies['Plot'].apply(lambda x: preprocess_text(x)[1])
```

In [21]:
```
movies.head(5)
```

Out[21]:

	MovieID	MovieName	Genre	\
0	23890098	Taxi Blues	[Drama, World cinema]	
1	31186339	The Hunger Games	[Action/Adventure, Action, Drama]	
2	20663735	Narasimham	[Musical, Action, Drama]	
3	2231378	The Lemon Drop Kid	[Comedy, Comedy]	
4	595909	A Cry in the Dark	[Crime Fiction, Drama, World cinema]	

	0
1	The nation of Panem consists of a wealthy Capitol and twelve poorer districts. As
2	Poovalli Induchoodan is sentenced for six years prison life for murdering his cla
3	The Lemon Drop Kid , a New York City swindler, is illegally touting horses at a Fl
4	Seventh-day Adventist Church pastor Michael Chamberlain, his wife Lindy, their two

	Num_Genres	\
0	2	
1	3	
2	3	
3	2	
4	3	

	0
1	nation panem consist wealthy capitol twelve poorer district punishment past rebell
2	poovalli induchoodan sentence six year prison life murder classmate induchoodan so
3	lemon drop kid new york city swindler illegally tout horse florida racetrack sever
4	seventh day adventist church pastor michael chamberlain wife lindy two son nine we

In [30]:
```
# write prepared dataset to a csv for future use
movies.to_csv("movies_small_subset_df.csv")
```

In [42]:
```
# read the data in from the csv
movies = pd.read_csv("movies_small_subset_df.csv")
```