This is for Final Project CIS 242 Spring 2021

In [1]:
```
import pandas as pd
import numpy as np
pd.options.display.float_format = '{:.5f}'.format
import matplotlib.pyplot as plt
import seaborn as sns

from scipy import stats
from ast import literal_eval
import ast


# Data Preprocessing and some general dataset feel was taken from https://www
# This code is part of CIS 242 Final Project. The author of this code is Kath
```

In [2]:
```
movies = pd.read_csv("movies_metadata.csv", low_memory=False)
movies.head()
```

Out[2]:

	adult	belongs_to_collection	budget	genres	homepage	id	
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	NaN	8844	tt
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collect...	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	NaN	15602	tt
3	False	NaN	16000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	31357	tt
4	False	{'id': 96871, 'name': 'Father of the Bride Col...	0	[{'id': 35, 'name': 'Comedy'}]	NaN	11862	tt

5 rows × 24 columns

In [3]:
```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
pd.set_option('display.max_colwidth', 150000) #important for getting all the
pd.set_option('display.max_columns', 999)
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from IPython.display import Image
import pydotplus

keywords = pd.read_csv("keywords.csv")
keywords.head()
```

Out[3]:

	id	keywords
0	862	[{'id': 931, 'name': 'jealousy'}, {'id': 4290, 'name': 'toy'}, {'id': 5202, 'name': 'boy'}, {'id': 6054, 'name': 'friendship'}, {'id': 9713, 'name': 'friends'}, {'id': 9823, 'name': 'rivalry'}, {'id': 165503, 'name': 'boy next door'}, {'id': 170722, 'name': 'new toy'}, {'id': 187065, 'name': 'toy comes to life'}]
1	8844	[{'id': 10090, 'name': 'board game'}, {'id': 10941, 'name': 'disappearance'}, {'id': 15101, 'name': "based on children's book"}, {'id': 33467, 'name': 'new home'}, {'id': 158086, 'name': 'recluse'}, {'id': 158091, 'name': 'giant insect'}]
2	15602	[{'id': 1495, 'name': 'fishing'}, {'id': 12392, 'name': 'best friend'}, {'id': 179431, 'name': 'duringcreditsstinger'}, {'id': 208510, 'name': 'old men'}]
3	31357	[{'id': 818, 'name': 'based on novel'}, {'id': 10131, 'name': 'interracial relationship'}, {'id': 14768, 'name': 'single mother'}, {'id': 15160, 'name': 'divorce'}, {'id': 33455, 'name': 'chick flick'}]
4	11862	[{'id': 1009, 'name': 'baby'}, {'id': 1599, 'name': 'midlife crisis'}, {'id': 2246, 'name': 'confidence'}, {'id': 4995, 'name': 'aging'}, {'id': 5600, 'name': 'daughter'}, {'id': 10707, 'name': 'mother daughter relationship'}, {'id': 13149, 'name': 'pregnancy'}, {'id': 33358, 'name': 'contraception'}, {'id': 170521, 'name': 'gynecologist'}]

In [4]:

```
# Let's clean data because I was getting dtype errors when I imported the csv

def clean_id(x):
    try:
        x = int(x)
    except:
        x = np.NaN
    return x
```

In [5]:

```
movies['id'] = movies['id'].apply(clean_id)
movies.dropna(subset=['id'], inplace=True)

# Dropping N/A values and making sure the id gets parsed as integer
```

In [6]:

```
df = pd.merge(movies, keywords, how='inner', on='id')

# merging the two datasets on the ID so we get one single dataframe lets take
df
```

Out[6]:

	adult	belongs_to_collection	budget	genres

	adult	belongs_to_collection	budget	genres	
0	False	{'id': 10194, 'name': 'Toy Story Collection', 'poster_path': '/7G9915LfUQ2lVfwMEEhDsn3kT4B.jpg', 'backdrop_path': '/9FBwqcd9IRruEDUrTdcaafOMKUq.jpg'}	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, 'name': 'Comedy'}, {'id': 10751, 'name': 'Family'}]	http://toystory.
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, 'name': 'Fantasy'}, {'id': 10751, 'name': 'Family'}]	

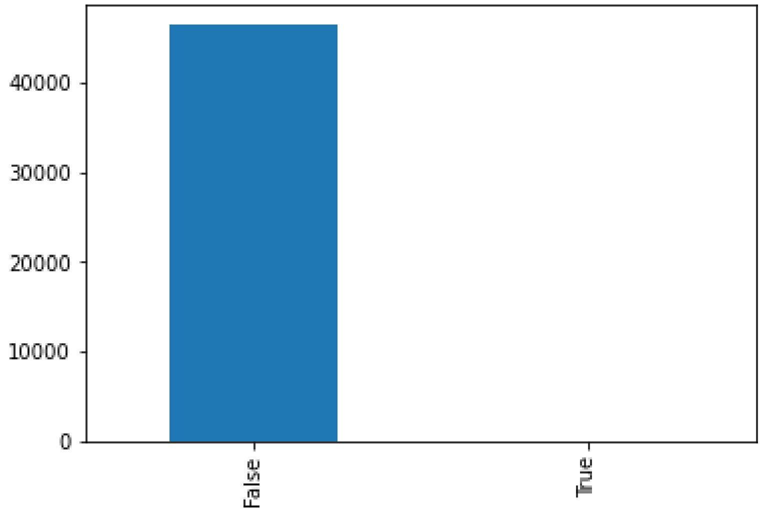
	adult	belongs_to_collection	budget	genres
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collection', 'poster_path': '/nLvUdqgPgm3F85NMCii9gVFUcet.jpg', 'backdrop_path': '/hypTnLot2z8wpFS7qwsQHW1uV8u.jpg'}	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, 'name': 'Comedy'}]
3	False	NaN	16000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'name': 'Drama'}, {'id': 10749, 'name': 'Romance'}]
4	False	{'id': 96871, 'name': 'Father of the Bride Collection', 'poster_path': '/nts4iOmNnq7GNicycMJ9pSAn204.jpg', 'backdrop_path': '/7qwE57OVZmMJChBpLEbJEmzUydk.jpg'}	0	[{'id': 35, 'name': 'Comedy'}]

	adult	belongs_to_collection	budget	genres	
...	...	...	...	...	
46477	False	NaN	0	[{'id': 18, 'name': 'Drama'}, {'id': 10751, 'name': 'Family'}]	http://www.imdb.
46478	False	NaN	0	[{'id': 18, 'name': 'Drama'}]	
46479	False	NaN	0	[{'id': 28, 'name': 'Action'}, {'id': 18, 'name': 'Drama'}, {'id': 53, 'name': 'Thriller'}]	

	adult	belongs_to_collection	budget	genres
46480	False	NaN	0	[]
46481	False	NaN	0	[]

46482 rows × 25 columns

In [7]:
```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
# get a feel for the distribution
df.adult.value_counts().plot(kind='bar')
plt.show()

# All these movies are rated UA / A which means we can just drop this column

df.adult.describe()
```



Out[7]:
```
count      46482
unique         2
top        False
freq       46473
Name: adult, dtype: object
```

In [8]:
```
df['tagline'] = df['tagline'].fillna('')
df['overview'] = df['overview'].fillna('')
df['description'] = df['overview'] + df['tagline']

df

# Cleaning up data further to make sure the model trained does not bug out, o

df = df.drop('homepage', axis=1)

# okay its gone now
```

In [9]:
```
df['genres'] = df['genres'].fillna('[]')
df['genres'] = df['genres'].apply(literal_eval)
df['genres'] = df['genres'].apply(lambda x: [i['name'] for i in x] if isinsta
```

In [10]:
```
def genres_list(df_genres):
    genres = set()

    for genres_list in df_genres:
        try:
            genres.update(genres_list)
        except AttributeError:
            pass

    return genres
```

```
genres = genres_list(df['genres'])
print(genres)
```

```
{'Fantasy', 'Science Fiction', 'History', 'Animation', 'Documentary', 'War',
'Drama', 'Horror', 'Western', 'TV Movie', 'Mystery', 'Adventure', 'Family', 'C
omedy', 'Romance', 'Action', 'Music', 'Crime', 'Thriller', 'Foreign'}
```

In [11]:

```
# Basically creating dummies and using the genres list as the dictionary to m
def split_genres(val):
    try:
        if gene in val:
            return 1
        else:
            return 0
    except AttributeError:
        return 0

# Apply function for each genre
for gene in genres:
    df[gene] = df['genres'].apply(split_genres)
```

In [12]:

```
df = df.drop('belongs_to_collection', axis = 1)
df = df.drop('imdb_id', axis = 1)
df = df.drop('original_language', axis = 1)
```

In [13]:

```
df = df.drop('poster_path', axis = 1)
```

In [14]:

```
df.columns
```

Out[14]:

```
Index(['adult', 'budget', 'genres', 'id', 'original_title', 'overview',
       'popularity', 'production_companies', 'production_countries',
       'release_date', 'revenue', 'runtime', 'spoken_languages', 'status',
       'tagline', 'title', 'video', 'vote_average', 'vote_count', 'keywords',
       'description', 'Fantasy', 'Science Fiction', 'History', 'Animation',
       'Documentary', 'War', 'Drama', 'Horror', 'Western', 'TV Movie',
       'Mystery', 'Adventure', 'Family', 'Comedy', 'Romance', 'Action',
       'Music', 'Crime', 'Thriller', 'Foreign'],
      dtype='object')
```

In [15]:

```
def clean_year(x):
    if x != np.nan:
        year = str(x).split('-')[0]
        return year
    else:
        return np.NaN

df.dropna(subset=['release_date'], inplace=True)
df['year'] = df['release_date'].apply(clean_year)
df = df.drop(['release_date'], axis=1)
```

In [16]:

```
df.dropna(subset=['year'], inplace=True)
df['year'] = df['year'].astype(int)
```

In [17]:

```
df['budget'] = pd.to_numeric(df['budget'])

df['revenue'] = pd.to_numeric(df['revenue'])

df
```

Out[17]:

	adult	budget	genres	id	original_title	overview	popularity

	adult	budget	genres	id	original_title	overview	popularity
0	False	30000000	[Animation, Comedy, Family]	862.00000	Toy Story	Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene. Afraid of losing his place in Andy's heart, Woody plots against Buzz. But when circumstances separate Buzz and Woody from their owner, the duo eventually learns to put aside their differences.	21.946943
1	False	65000000	[Adventure, Fantasy, Family]	8844.00000	Jumanji	When siblings Judy and Peter discover an enchanted board game that opens the door to a magical world, they unwittingly invite Alan -- an adult who's been trapped inside the game for 26 years -- into their living room. Alan's only hope for freedom is to finish the game, which proves risky as all three find themselves running from giant rhinoceroses, evil monkeys and other terrifying creatures.	17.015539

	adult	budget	genres	id	original_title	overview	popularity
2	False	0	[Romance, Comedy]	15602.00000	Grumpier Old Men	A family wedding reignites the ancient feud between next-door neighbors and fishing buddies John and Max. Meanwhile, a sultry Italian divorcée opens a restaurant at the local bait shop, alarming the locals who worry she'll scare the fish away. But she's less interested in seafood than she is in cooking up a hot time with Max.	11.7129
3	False	16000000	[Comedy, Drama, Romance]	31357.00000	Waiting to Exhale	Cheated on, mistreated and stepped on, the women are holding their breath, waiting for the elusive "good man" to break a string of less-than-stellar lovers. Friends and confidants Vannah, Bernie, Glo and Robin talk it all out, determined to find a better way to breathe.	3.859495

	adult	budget	genres	id	original_title	overview	popularity
4	False	0	[Comedy]	11862.00000	Father of the Bride Part II	Just when George Banks has recovered from his daughter's wedding, he receives the news that she's pregnant ... and that George's wife, Nina, is expecting too. He was planning on selling their home, but that's a plan that -- like George -- will have to change with the arrival of both a grandchild and a kid of his own.	8.387519
...	...	...	...	...	...	...	...
46476	False	0	[Drama, Action, Romance]	30840.00000	Robin Hood	Yet another version of the classic epic, with enough variation to make it interesting. The story is the same, but some of the characters are quite different from the usual, in particular Uma Thurman's very special maid Marian. The photography is also great, giving the story a somewhat darker tone.	5.683753
46478	False	0	[Drama]	111109.00000	Siglo ng Pagluluwal	An artist struggles to finish his work while a storyline about a cult plays in his head.	0.178241

	adult	budget	genres	id	original_title	overview	popularity
46479	False	0	[Action, Drama, Thriller]	67758.00000	Betrayal	When one of her hits goes wrong, a professional assassin ends up with a suitcase full of a million dollars belonging to a mob boss ...	0.903007
46480	False	0	[]	227506.00000	Satana likuyushchiy	In a small town live two brothers, one a minister and the other one a hunchback painter of the chapel who lives with his wife. One dreadful and stormy night, a stranger knocks at the door asking for shelter. The stranger talks about all the good things of the earthly life the minister is missing because of his puritanical faith. The minister comes to accept the stranger's viewpoint but it is others who will pay the consequences because the minister will discover the human pleasures thanks to, ehem, his sister- in -law... The tormented minister and his cuckolded brother will die in a strange accident in the chapel and later an infant will be born from the minister's adulterous relationship.	0.003503

	adult	budget	genres	id	original_title	overview	popularity
46481	False	0	[]	461257.00000	Queerama	50 years after decriminalisation of homosexuality in the UK, director Daisy Asquith mines the jewels of the BFI archive to take us into the relationships, desires, fears and expressions of gay men and women in the 20th century.	0.163015

46394 rows × 41 columns

In [18]:

```
df
```

Out[18]:

	adult	budget	genres	id	original_title	overview	popularity
0	False	30000000	[Animation, Comedy, Family]	862.00000	Toy Story	Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene. Afraid of losing his place in Andy's heart, Woody plots against Buzz. But when circumstances separate Buzz and Woody from their owner, the duo eventually learns to put aside their differences.	21.946943

	adult	budget	genres	id	original_title	overview	popularity
1	False	65000000	[Adventure, Fantasy, Family]	8844.00000	Jumanji	When siblings Judy and Peter discover an enchanted board game that opens the door to a magical world, they unwittingly invite Alan -- an adult who's been trapped inside the game for 26 years -- into their living room. Alan's only hope for freedom is to finish the game, which proves risky as all three find themselves running from giant rhinoceroses, evil monkeys and other terrifying creatures.	17.015539
2	False	0	[Romance, Comedy]	15602.00000	Grumpier Old Men	A family wedding reignites the ancient feud between next-door neighbors and fishing buddies John and Max. Meanwhile, a sultry Italian divorcée opens a restaurant at the local bait shop, alarming the locals who worry she'll scare the fish away. But she's less interested in seafood than she is in cooking up a hot time with Max.	11.7129

	adult	budget	genres	id	original_title	overview	popularity
3	False	16000000	[Comedy, Drama, Romance]	31357.00000	Waiting to Exhale	Cheated on, mistreated and stepped on, the women are holding their breath, waiting for the elusive "good man" to break a string of less-than-stellar lovers. Friends and confidants Vannah, Bernie, Glo and Robin talk it all out, determined to find a better way to breathe.	3.859495
4	False	0	[Comedy]	11862.00000	Father of the Bride Part II	Just when George Banks has recovered from his daughter's wedding, he receives the news that she's pregnant ... and that George's wife, Nina, is expecting too. He was planning on selling their home, but that's a plan that -- like George -- will have to change with the arrival of both a grandchild and a kid of his own.	8.387519
...	...	...	...	...	...	...	...

	adult	budget	genres	id	original_title	overview	popularity
46476	False	0	[Drama, Action, Romance]	30840.00000	Robin Hood	Yet another version of the classic epic, with enough variation to make it interesting. The story is the same, but some of the characters are quite different from the usual, in particular Uma Thurman's very special maid Marian. The photography is also great, giving the story a somewhat darker tone.	5.683753
46478	False	0	[Drama]	111109.00000	Siglo ng Pagluluwal	An artist struggles to finish his work while a storyline about a cult plays in his head.	0.178241
46479	False	0	[Action, Drama, Thriller]	67758.00000	Betrayal	When one of her hits goes wrong, a professional assassin ends up with a suitcase full of a million dollars belonging to a mob boss ...	0.903007

	adult	budget	genres	id	original_title	overview	popularity
46480	False	0	[]	227506.00000	Satana likuyushchiy	In a small town live two brothers, one a minister and the other one a hunchback painter of the chapel who lives with his wife. One dreadful and stormy night, a stranger knocks at the door asking for shelter. The stranger talks about all the good things of the earthly life the minister is missing because of his puritanical faith. The minister comes to accept the stranger's viewpoint but it is others who will pay the consequences because the minister will discover the human pleasures thanks to, ehem, his sister- in -law... The tormented minister and his cuckolded brother will die in a strange accident in the chapel and later an infant will be born from the minister's adulterous relationship.	0.003503
46481	False	0	[]	461257.00000	Queerama	50 years after decriminalisation of homosexuality in the UK, director Daisy Asquith mines the jewels of the BFI archive to take us into the relationships, desires, fears and expressions of gay men and women in the 20th century.	0.163015

46394 rows × 41 columns

In [19]:
```
#df['budget'] = df[df['budget'].between(, 800000000)]
#df['revenue'] = df[df['revenue'].between(100, 800000000)]
```

In [20]:
```
#df['budget'] = df[df['budget'] > 0]
```

In [21]:
```
df = df[df['budget'].between(100, 800000000)]
df = df[df['revenue'].between(100, 800000000)]
```

In [22]:
```
def add_movie_year_period(movie_year):
    if movie_year < 1900:
        return '1800'
    elif movie_year < 2000:
        return '1900'
    elif movie_year < 2010:
        return '2000'
    else :
        return '2010'
```

In [23]:
```
df['year_period'] = df['year'].apply(add_movie_year_period)
```

In [24]:
```
df['keywords'] = df['keywords'].apply(literal_eval)
df['keywords'] = df['keywords'].apply(lambda x: [i['name'] for i in x] if isi
```

In [25]:
```
dictionary = {}

def count_words(word_list):
    for word in word_list:
        if dictionary.get(word) == None:
            dictionary[word] = 1
        else:
            dictionary[word] += 1


df['keywords'].apply(count_words)


dictionary_copy = dictionary.copy()
for key, value in dictionary_copy.items():
    if value == 1:
        dictionary.pop(key)


dictionary_copy
```

Out[25]:
```
{'jealousy': 56,
 'toy': 9,
 'boy': 18,
 'friendship': 120,
 'friends': 53,
 'rivalry': 33,
 'boy next door': 1,
 'new toy': 2,
 'toy comes to life': 12,
 'board game': 2,
 'disappearance': 16,
 "based on children's book": 11,
 'new home': 2,
 'recluse': 1,
 'giant insect': 2,
 'based on novel': 264,
```

```
                 'interracial relationship': 12,
                 'single mother': 20,
                 'divorce': 43,
                 'chick flick': 2,
                 'robbery': 54,
                 'detective': 58,
                 'bank': 21,
                 'obsession': 45,
                 'chase': 43,
                 'shooting': 9,
                 'thief': 30,
                 'honor': 5,
                 'murder': 250,
                 'suspense': 117,
                 'heist': 30,
                 'betrayal': 29,
                 'money': 54,
                 'gang': 42,
                 'cat and mouse': 4,
                 'criminal mastermind': 3,
                 'cult film': 22,
                 'ex-con': 9,
                 'heist movie': 3,
                 'one last job': 3,
                 'loner': 5,
                 'bank job': 1,
                 'neo-noir': 29,
                 'gun fight': 3,
                 'crime epic': 2,
                 'terrorist': 34,
                 'hostage': 39,
                 'explosive': 9,
                 'vice president': 1,
                 'cuba': 11,
                 'falsely accused': 11,
                 'secret identity': 36,
                 'computer virus': 8,
                 'secret base': 6,
                 'secret intelligence service': 12,
                 'kgb': 11,
                 'satellite': 7,
                 'special car': 3,
                 'cossack': 3,
                 'electromagnetic pulse': 1,
                 'time bomb': 2,
                 'st. petersburg russia': 3,
                 'ejection seat': 2,
                 'red army': 3,
                 'white house': 12,
                 'usa president': 32,
                 'new love': 37,
                 'widower': 11,
                 'wildlife conservation': 2,
                 'presidential election': 6,
                 'watergate scandal': 2,
                 'biography': 144,
                 'government': 28,
                 'historical figure': 26,
                 'exotic island': 14,
                 'treasure': 19,
                 'map': 8,
                 'ship': 29,
                 'scalp': 2,
                 'pirate': 12,
                 'poker': 6,
                 'drug abuse': 12,
                 '1970s': 41,
                 'overdose': 8,
                 'illegal prostitution': 13,
```

```
'bowling': 4,
'servant': 5,
'country life': 5,
'jane austen': 3,
'inheritance': 6,
'military officer': 3,
'period drama': 12,
'rainstorm': 3,
'horse and carriage': 1,
'decorum': 1,
'hotel': 48,
"new year's eve": 12,
'witch': 41,
'bet': 8,
'hotel room': 16,
'sperm': 3,
'los angeles': 111,
'hoodlum': 20,
'woman director': 276,
'episode film': 1,
'africa': 24,
'indigenous': 6,
'human animal relationship': 9,
'bat': 3,
'brother brother relationship': 62,
'subway': 18,
'new york city': 64,
'new york subway': 3,
'train robbery': 1,
'gambling': 21,
'miami': 12,
'job': 10,
'travel': 19,
'mafia': 27,
'debt': 8,
'mobster': 21,
'business': 9,
'hollywood': 22,
'gangster': 45,
'crime': 25,
'violence': 200,
'drug': 92,
'producer': 3,
'con': 3,
'competition': 31,
'assassination': 38,
'cia': 50,
'cat': 23,
'mexican standoff': 14,
'seattle': 5,
'hitman': 47,
'mission of murder': 16,
'hidden camera': 8,
'rescue': 54,
'shootout': 53,
'police chase': 10,
'sniper rifle': 4,
'silencer': 9,
'double cross': 10,
'caribbean': 7,
'detroit michigan': 4,
'individual': 21,
'prostitute': 55,
'alcohol': 45,
'casino': 19,
'love at first sight': 15,
'lovesickness': 15,
'film producer': 10,
'screenwriter': 11,
```

```
'dying and death': 76,
'rage and hate': 10,
'unsociability': 18,
'alcoholism': 17,
'alcohol abuse': 7,
'attempted suicide': 2,
'female friendship': 16,
'coming of age': 31,
'gynecologist': 2,
'photocopier': 1,
'truth or dare': 1,
'clone': 14,
'dream': 45,
'island': 43,
'eye': 4,
'dystopia': 160,
'aging': 7,
'children': 22,
'girl': 6,
'childhood': 3,
'schizophrenia': 10,
'philadelphia': 7,
'cassandra syndrom': 1,
'stockholm syndrome': 1,
'time travel': 52,
'post-apocalyptic': 41,
'lethal virus': 5,
'monkey': 14,
'subterranean': 3,
'sheep': 2,
'pig': 8,
'affection': 5,
'piglet': 3,
'heroism': 4,
'talking animal': 18,
'separation': 1,
'german shepherd': 4,
'grandson': 1,
'talking pig': 3,
'prison': 91,
'rape': 59,
'socially deprived family': 4,
'penalty': 5,
'death penalty': 9,
'despair': 7,
'death row': 3,
'begnadigung': 2,
'therapist': 16,
'self-discovery': 8,
'prison cell': 6,
'court case': 26,
'death sentence': 3,
'doomed man': 4,
'sentence': 3,
'lethal injection': 4,
'forgiveness': 10,
'charity': 3,
'mercy petition': 1,
'right and justice': 12,
'court': 25,
'electric chair': 6,
'cowardliness': 19,
'martial arts': 64,
'monster': 58,
'gore': 51,
'sorcerer': 9,
'tournament': 6,
'based on video game': 36,
'hand to hand combat': 12,
```

```
'adultery': 41,
'winter': 19,
'television': 11,
'new hampshire': 3,
'narcissistic personality disorder': 1,
'wedding vows': 5,
'marriage proposal': 17,
'married couple': 16,
'monogamy': 1,
'advice': 3,
'marriage': 54,
'quilt': 1,
'love': 132,
'family holiday': 9,
'extramarital affair': 39,
'grandmother': 3,
'self-fulfilling prophecy': 3,
's.w.a.t.': 3,
'drug dealer': 38,
'evisceration': 2,
'lust and impulsiveness': 1,
'pride and vanity': 2,
'immoderateness': 1,
'insomnia': 7,
'investigation': 69,
'pension': 2,
'police': 125,
'serial killer': 71,
'culture clash': 11,
'settler': 8,
'forbidden love': 21,
'colony': 6,
'musical': 111,
'gold rush': 3,
'princess': 32,
'romance': 31,
'native american': 18,
'animation': 39,
'virginia': 3,
'star crossed lovers': 6,
'reference to pizarro': 1,
'jamestown virginia': 1,
'pug dog': 1,
'cross cultural relationship': 1,
'musket': 2,
'animate tree': 1,
'indian chief': 1,
'based on myths or folklore': 4,
'17th century': 4,
'shamanism': 1,
'songs': 1,
'indians vs. settlers': 1,
'animal sidekick': 4,
'powhatan': 1,
'land claim': 1,
'law': 9,
'relatives': 2,
'theft': 17,
'criminal': 35,
'hungarian': 2,
'sibling': 1,
'adoption': 24,
'adoptive father': 4,
'childlessness': 5,
'looking for  birth parents': 5,
'adoptive mother': 2,
'independent film': 241,
'child': 19,
'rap music': 10,
```

```
'parent child relationship': 31,
'rapper': 7,
'dancing': 27,
'sexual obsession': 8,
'showdown': 17,
'sheriff': 38,
'eroticism': 21,
'nudity': 110,
'bank robber': 13,
'vampire': 54,
'holy water': 4,
'siege': 6,
'stripper': 12,
'priest': 21,
'explosion': 56,
'extreme violence': 28,
'bank robbery': 15,
'preacher': 4,
'hostage situation': 6,
'crucifix': 5,
'recreational vehicle': 1,
'blood spurting': 2,
'convenience store': 3,
'female vampire': 3,
'mexican american border': 1,
'nightclub entertainer': 1,
'boa constrictor': 1,
'bomb': 32,
'sex': 129,
'handcuffs': 4,
'fbi': 49,
'kidnapping': 78,
'russian': 12,
'sadism': 16,
'parking garage': 3,
'pizza': 4,
'police protection': 2,
'psychopath': 50,
'remake': 78,
'car crash': 41,
'conspiracy': 41,
'on the run': 25,
'fugitive': 18,
'gunfight': 28,
'sadist': 6,
'foot chase': 16,
'police detective': 11,
'police station': 5,
'car chase': 36,
'held at gunpoint': 3,
'fbi agent': 39,
'computer hacker': 7,
'machine gun': 5,
'swat team': 6,
'grenade launcher': 2,
'underwater scene': 1,
'rogue agent': 4,
'woman lawyer': 1,
'damsel in distress': 5,
'bully': 11,
'man vs machine': 12,
'alien planet': 8,
'struggle for survival': 2,
'father son relationship': 71,
'bounty hunter': 14,
'boat': 18,
'way of life': 2,
'coffin': 8,
'denver': 2,
```

```
'godmother': 2,
'paranoia': 28,
'revenge': 141,
'diner': 15,
'blood': 56,
'sailing trip': 2,
'diary': 11,
'sailing': 10,
'survival': 41,
'teenage boy': 9,
'discipline': 3,
'squall': 1,
'sail': 1,
'ship captain': 3,
'male bonding': 2,
'seasickness': 1,
'storm at sea': 6,
'1960s': 21,
'train station': 4,
'politics': 24,
'laboratory': 11,
'jekyll and hyde': 1,
'housemaid': 2,
'19th century': 22,
'half vampire': 1,
'helicopter': 46,
'river': 21,
'captain': 9,
'underground': 8,
'countdown': 2,
'pilot': 26,
'fistfight': 18,
'canyon': 2,
'major': 3,
'train': 23,
'park ranger': 2,
'desert': 29,
'military': 26,
'nuclear device': 1,
'boxing': 5,
'stealth aircraft': 1,
'abandoned mine': 2,
'humvee': 1,
'burglar': 12,
'language barrier': 5,
'motel': 26,
'psychiatric hospital': 4,
'maid': 7,
'nervous breakdown': 2,
'escapade': 8,
'laundry room': 1,
'loss of mother': 21,
'loss of father': 27,
'golf': 6,
'sport': 125,
'taxes': 2,
'farewell': 5,
'photographer': 17,
'wife husband relationship': 53,
'iowa': 2,
"love of one's life": 48,
'mother role': 8,
'bridge': 13,
'housewife': 5,
'love letter': 7,
'peasant': 9,
'marriage crisis': 20,
'photography': 14,
'secret love': 19,
```

```
'nature documentary': 1,
'scotland': 19,
'in love with enemy': 5,
'legend': 6,
'independence': 6,
'idealism': 3,
'revolt': 5,
'tyranny': 2,
'vietnam veteran': 13,
'taxi': 20,
'night shift': 3,
'vigilante': 17,
'alienation': 2,
'misanthrope': 3,
'shot to death': 6,
'new york': 99,
'supermarket': 11,
'gang war': 9,
'disabled child': 1,
'wedding': 66,
'diamond': 5,
'bronx': 2,
'duringcreditsstinger': 302,
'twin brother': 11,
'lawyer': 53,
'estate': 7,
'santa barbara california': 2,
'role of women': 3,
'ladykiller': 15,
'success': 13,
"ladies' man": 3,
'chefin': 1,
'casanova': 3,
'womanizer': 9,
'lsd': 3,
'half-brother': 3,
'callgirl': 1,
'san francisco': 33,
'gay': 41,
'drag queen': 7,
'homophobia': 12,
'florida': 21,
'coming out': 10,
'senator': 11,
'politician': 19,
'lgbt': 18,
'crossdressing': 3,
'1990s': 5,
'airport': 22,
'strip club': 19,
'witness protection': 10,
'internal affairs': 3,
'brutality': 16,
'drug lord': 14,
'hangar': 1,
'loose cannon': 1,
'bust': 1,
'reference to skittles': 1,
'black cop': 1,
'badge': 1,
'action hero': 8,
'corrupt cop': 6,
'england': 41,
'liverpool': 1,
'theatre company': 1,
'1940s': 11,
'moon': 13,
'nasa': 14,
'spaceman': 3,
```

```
'race against time': 14,
'houston': 1,
'based on true story': 73,
'space': 30,
'disaster': 20,
'astronaut': 25,
'imax': 15,
'saturn v rocket': 1,
'courage': 5,
'hypothermia': 1,
'apollo program': 1,
'cape kennedy': 1,
'lunar mission': 1,
'spacecraft accident': 1,
'18th century': 9,
'highlands': 1,
'violent man': 1,
'riddle': 19,
'dc comics': 22,
'rose': 4,
'gotham city': 5,
'partner': 17,
'superhero': 64,
'robin': 1,
'broken neck': 4,
'psychologist': 14,
'district attorney': 4,
'millionaire': 23,
'falling down stairs': 1,
'tied up': 3,
'tommy gun': 3,
'beretta': 1,
'knocked out': 2,
'super powers': 25,
'disfigurement': 12,
'father figure': 6,
'smoking': 8,
'corner shop': 1,
'cigarette': 5,
'tobacco': 3,
'cigar': 3,
'halloween': 18,
'supernatural': 39,
'afterlife': 15,
'danger': 6,
'ghost': 35,
'disorder': 3,
'young heroes': 1,
'imaginary': 3,
'supernatural ability': 3,
'mischievous children': 1,
'drug pusher': 1,
'gorilla': 4,
'kongo': 3,
'diamond mine': 3,
'submarine': 16,
'mutiny': 11,
'russia': 17,
'missile': 11,
'nuclear missile': 10,
'embassy': 5,
'u.s. navy': 14,
'battle for power': 3,
'torpedo': 3,
'moral dilemma': 5,
'post cold war': 1,
'aircraft carrier': 2,
'chain of command': 1,
'launch code': 1,
```

```
'sonar': 1,
'gunslinger': 9,
'anti terror': 4,
'ambush': 17,
'mexico': 20,
'guitar': 9,
'bartender': 5,
'tragic hero': 4,
'mariachi': 1,
'leg brace': 2,
'concealed weapon': 1,
'flashback': 47,
'bookstore': 1,
'jazz club': 1,
'private detective': 16,
'film noir': 15,
'gold': 22,
'sequel': 130,
'deception': 30,
'simon says': 1,
'dump truck': 1,
'aqueduct': 1,
'federal reserve bank': 1,
'camelot': 2,
'knight': 15,
'king arthur': 5,
'excalibur': 1,
'knights of the round table': 2,
'female nudity': 89,
'hacker': 20,
'virtual reality': 17,
'computer': 20,
'sexual fantasy': 6,
'prank': 10,
'internet': 9,
'cyberpunk': 22,
'teenager': 108,
'secret service': 11,
'dream sequence': 6,
'brain': 4,
'childhood memory': 9,
'pharmaceutical industry': 2,
'computer chip': 2,
'courier': 1,
'cyber': 1,
'judge': 17,
'based on comic': 64,
'police officer': 21,
'based on graphic novel': 10,
'frame up': 2,
'dystopic future': 14,
'framed for murder': 6,
'gene': 1,
'prosthetic limb': 1,
'post holocaust': 1,
'dna testing': 2,
'dredd': 2,
'2000 ad': 2,
'puberty': 5,
'first time': 7,
'cinema': 8,
'filmmaker': 2,
'game show': 3,
'slacker': 6,
'shopping': 2,
'mall': 3,
'ex-boyfriend ex-girlfriend relationship': 2,
'jay and silent bob': 2,
'silent man': 1,
```

```
'coke': 1,
'bandleader': 1,
'aftercreditsstinger': 157,
'based on tv series': 30,
'tokusatsu': 3,
'superhero team': 7,
'ethnic diversity': 3,
'super sentai': 1,
'power rangers': 1,
'mighty morphin power rangers': 1,
'mmpr': 1,
'cheating': 22,
'new identity': 13,
'stalking': 12,
'library': 9,
'party': 56,
'free spirit': 3,
'librarian': 3,
'burning of witches': 2,
'puritan': 2,
'pregnancy': 31,
'go-go dancer': 1,
'spanner': 3,
'seduction': 23,
'striptease': 6,
'sexappeal': 2,
'robber': 4,
'laden': 1,
'writer': 33,
'telepathy': 11,
'dna': 8,
'genetics': 10,
'instinct': 1,
'femme fatale': 9,
'alien': 88,
'decapitation': 16,
'sexual attraction': 6,
'cocoon': 2,
'genetic engineering': 4,
'scientists': 1,
'alien dna': 2,
'pornography': 12,
'police brutality': 19,
'ex-girlfriend': 10,
'future': 54,
'bodyguard': 11,
'minidisc': 2,
'ex-cop': 15,
'pentagon': 4,
'navy seal': 6,
'world war ii': 77,
'vineyard': 4,
'harvest': 2,
'grape': 1,
'abandoned woman': 1,
'ocean': 17,
'tattoo': 11,
'mutant': 30,
'water': 12,
'doomsday': 5,
'colonel': 9,
'dialogue': 5,
'sunrise': 5,
'talking': 3,
'soulmates': 8,
'walking': 2,
'austria': 5,
'traveller': 2,
'danube': 1,
```

```
'bittersweet': 5,
'romantic': 2,
'vienna': 4,
'aids': 14,
'roommate': 14,
'homicide': 13,
'car journey': 12,
'homosexuality': 20,
'escape': 69,
'road movie': 17,
'salesclerk': 5,
'loser': 8,
'sex addiction': 6,
'mental institution': 7,
'patient': 7,
'psychiatrist': 22,
'don juan': 1,
'employee': 4,
'workplace': 2,
'sexual harassment': 3,
'intrigue': 6,
'gas station': 13,
'utah': 3,
'stupidity': 6,
'pill': 2,
'cigar smoking': 4,
'fired from the job': 2,
'clumsiness': 2,
'stepmother stepdaughter relationship': 1,
'sitting on a toilet': 9,
'aspen colorado': 1,
'parakeet': 1,
'defecation': 1,
'scooter': 3,
'hotel suite': 2,
'endangered species': 1,
'foolish': 1,
'laxative': 7,
'transsexuality': 7,
'fortune teller': 7,
'film business': 7,
'film making': 10,
'vororte': 7,
'film maker': 4,
'boxer': 19,
'film director': 6,
'oddball': 1,
'celebrity': 8,
'morphine': 1,
'movie studio': 2,
'drug addict': 9,
'cult director': 1,
'theremin': 1,
'handgun': 2,
'trick or treating': 3,
'actor': 9,
'transvestite': 1,
'chicago': 27,
'sports team': 7,
'ghetto': 10,
'narration': 18,
'college': 36,
'basketball': 11,
'high school sports': 7,
'inner city': 1,
'high school student': 10,
'mother': 19,
'secret': 48,
'literature': 10,
```

```
'fantasy': 20,
'passion': 9,
'lesbian': 17,
'true': 7,
'relationship': 38,
'paris': 59,
'plantation': 6,
'pity': 1,
'bite': 17,
'fang vamp': 16,
'france': 23,
'revolution': 13,
'president': 14,
'history': 11,
'baby': 32,
'medicine': 6,
'media': 4,
'scientist': 49,
'science': 13,
'male pregnancy': 1,
'fertility': 1,
'gynaecology': 1,
'ceasarean birth': 1,
'android': 25,
'galaxy': 5,
'hermit': 1,
'death star': 3,
'lightsaber': 3,
'jedi': 5,
'rescue mission': 8,
'empire': 1,
'rebellion': 6,
'planet': 11,
'smuggler': 3,
'the force': 2,
'space opera': 17,
'galactic war': 1,
'stormtrooper': 2,
'totalitarianism': 2,
'chocolate': 5,
'sister sister relationship': 30,
'cooking': 5,
'mexican revolution': 2,
'recipe': 2,
'magic realism': 2,
'new mexican cinema': 4,
'new latin american cinema': 1,
'montana': 6,
'world war i': 18,
'journey round the world': 4,
'child abuse': 17,
'single parent': 15,
'social work': 2,
'drug addiction': 19,
'liberation': 10,
'social worker': 5,
'self-abandonment': 3,
'custody battle': 6,
'son': 11,
'surrogate mother': 4,
'racist': 7,
'human experimentation': 13,
'frankenstein': 9,
'loss of sister': 7,
'twin sister': 3,
'autism': 11,
'lake': 14,
'kaspar-hauser-syndrom': 2,
'feral child': 7,
```

```
'forest': 32,
'north carolina': 5,
'family': 80,
'sadistic': 8,
'journalist': 56,
'mass murder': 18,
'yellow press': 2,
'trauma': 8,
'daughter': 65,
'satire': 18,
'controversy': 2,
'controversial': 6,
'young couple': 1,
'abuse': 6,
'general': 13,
'research': 8,
'army': 31,
'serum': 3,
'epidemic': 7,
'medical research': 4,
'corruption': 62,
'assassin': 48,
'loss of family': 15,
'immigrant': 12,
'training': 27,
'loneliness': 14,
'neighbor': 32,
'tragic love': 9,
'complex relationship': 1,
'complex characters': 1,
'immigration': 6,
'political prisoner': 2,
'cuban refugees': 1,
'transporter': 14,
'brothel': 14,
'massage': 1,
'stolen money': 6,
'crime boss': 5,
'dance contest': 2,
'junkyard': 2,
'kamikaze': 2,
'ambiguous ending': 2,
'briefcase': 3,
'redemption': 14,
'heirloom': 1,
'pulp fiction': 1,
'reference to al green': 1,
'prairie': 2,
'pistol': 14,
'grandfather grandson relationship': 9,
'birthday': 17,
'funeral': 28,
'vegetarian': 10,
'orphan': 33,
'deathbed': 1,
'suspicion of murder': 13,
'mannequin': 7,
'fashion photographer': 2,
'sandwich': 2,
'poodle': 1,
'reporter': 26,
'fashion': 10,
'swan': 1,
'explosives expert': 2,
'space travel': 25,
'teleportation': 8,
'uprising': 5,
'androgyny': 2,
'space western': 1,
```

```
'outer space': 21,
'timebomb': 3,
'death of son': 2,
'nuclear weapons': 5,
'invented language': 1,
'hieroglyph': 1,
'egyptian mythology': 3,
'egyptology': 3,
'holiday': 39,
'christmas party': 16,
'santa claus': 15,
'deal': 4,
'christmas tree': 8,
'christmas': 39,
'delinquent': 3,
'parole board': 2,
'escape from prison': 9,
'wrongful imprisonment': 2,
'destroy': 10,
'reincarnation': 6,
'artial arts': 1,
'death': 59,
'exploding planet': 1,
'mountain cabin': 1,
'solar system': 2,
'ku klux klan': 9,
'anthology': 7,
'evil doll': 3,
'hood': 4,
'dirty cop': 15,
'funeral home': 3,
'mentally disabled': 8,
'widow': 20,
'dysfunctional family': 19,
'artificial intelligence': 27,
'hologram': 6,
'computer program': 4,
'visual effect': 1,
'police training': 2,
'coma': 15,
'brother': 14,
"man of one's dreams": 1,
'hospital': 65,
'becoming an adult': 20,
'return': 6,
'overweight': 4,
'province': 3,
'bathing': 4,
'olympic games': 13,
'empowerment': 8,
'boredom': 4,
'dolphin': 7,
'mascot': 3,
'pets': 1,
'wheelchair': 19,
'cruise': 7,
'longing': 2,
'paralysis': 3,
'sadomasochism': 7,
'voyeurism': 10,
'spy': 47,
'sniper': 22,
'colombia': 5,
'drug traffic': 14,
'bomber': 4,
'mercenary': 18,
'insurgence': 7,
'coast guard': 5,
'spying': 3,
```

```
 'war': 55,
 'car bomb': 7,
 'drug cartel': 5,
 'infantry': 6,
 'jack ryan': 5,
 'political cover-up': 1,
 'suicide': 63,
 'principal witness ': 3,
 'search for witnesses': 1,
 'arson': 7,
 'detroit': 12,
 'manager': 6,
 'bad mother-in-law': 5,
 "family's daily life": 8,
 'stone age': 8,
 'plan': 10,
 'best friend': 48,
 'dinosaur': 18,
 ...}
```

In [ ]:

In [26]:

```
def filter_keywords(word_list):
    words = []
    for word in word_list:
        if dictionary.get(word):
            words.append(word)
    return words
```

In [27]:

```
df['keywords'] = df['keywords'].apply(filter_keywords)
```

In [28]:

```
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import wordnet
from nltk.stem.porter import PorterStemmer

import warnings; warnings.simplefilter('ignore')
%matplotlib inline
```

In [29]:

```
stemmer = PorterStemmer()
df['keywords'] = df['keywords'].apply(lambda x: [stemmer.stem(i) for i in x])
df['keywords'] = df['keywords'].apply(lambda x: [str.lower(i.replace(" ", ""))
```

In [30]:

```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
df['pstem'] = df["overview"].apply(lambda x: [stemmer.stem(y) for y in x.spli
df['pstem']= [" ".join(token) for token in df['pstem']]
```

In [31]:

```
df['popularity']
```

Out[31]:

```
0        21.946943
1        17.015539
3         3.859495
5        17.924927
8          5.23158
           ...
46184    40.796775
46267     1.323587
46425     0.903061
46428     0.121844
46438     0.039793
Name: popularity, Length: 5309, dtype: object
```

In [32]:

```
import seaborn as sns
```

```
df
```

Out[32]:

	adult	budget	genres	id	original_title	overview	popularity
0	False	30000000	[Animation, Comedy, Family]	862.00000	Toy Story	Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene. Afraid of losing his place in Andy's heart, Woody plots against Buzz. But when circumstances separate Buzz and Woody from their owner, the duo eventually learns to put aside their differences.	21.94694
1	False	65000000	[Adventure, Fantasy, Family]	8844.00000	Jumanji	When siblings Judy and Peter discover an enchanted board game that opens the door to a magical world, they unwittingly invite Alan -- an adult who's been trapped inside the game for 26 years -- into their living room. Alan's only hope for freedom is to finish the game, which proves risky as all three find themselves running from giant rhinoceroses, evil monkeys and other terrifying creatures.	17.01553

	adult	budget	genres	id	original_title	overview	popularity
3	False	16000000	[Comedy, Drama, Romance]	31357.00000	Waiting to Exhale	Cheated on, mistreated and stepped on, the women are holding their breath, waiting for the elusive "good man" to break a string of less-than-stellar lovers. Friends and confidants Vannah, Bernie, Glo and Robin talk it all out, determined to find a better way to breathe.	3.85949
5	False	60000000	[Action, Crime, Drama, Thriller]	949.00000	Heat	Obsessive master thief, Neil McCauley leads a top-notch crew on various insane heists throughout Los Angeles while a mentally unstable detective, Vincent Hanna pursues him without rest. Each man recognizes and respects the ability and the dedication of the other even though they are aware their cat-and-mouse game may end in violence.	17.92492

	adult	budget	genres	id	original_title	overview	popularity
8	False	35000000	[Action, Adventure, Thriller]	9091.00000	Sudden Death	International action superstar Jean Claude Van Damme teams with Powers Boothe in a Tension-packed, suspense thriller, set against the back-drop of a Stanley Cup game.Van Damme portrays a father whose daughter is suddenly taken during a championship hockey game. With the captors demanding a billion dollars by game's end, Van Damme frantically sets a plan in motion to rescue his daughter and abort an impending explosion before the final buzzer...	5.23158
...	...	...	...	...	...	...	..
46184	False	11000000	[Action, Crime, Mystery, Thriller]	395834.00000	Wind River	An FBI agent teams with the town's veteran game tracker to investigate a murder that occurred on a Native American reservation.	40.79677
46267	False	12000000	[Action, Comedy, Drama]	24049.00000	சிவாஜி	Corrupt police and politicians target a computer engineer for trying to better the lives of less privileged citizens.	1.32358

	adult	budget	genres	id	original_title	overview	popularity
46425	False	800000	[Comedy, Drama]	62757.00000	Dikari	The sea, August, interesting and simple people. They tan, swim, play volleyball, basketball, drink, dance and then find someone to spend the night with. Many grew out of their student phase and can afford a more comfortable holiday but when July comes they grab a tent, jump into their cars and come here. Here, nobody talks about work and the size of your wallet means nothing.	0.90306

	adult	budget	genres	id	original_title	overview	popularity
46428	False	2000000	[Romance, Drama]	63281.00000	Про любоff	У девушки Даши, приехавшей с подругой «покорять» Москву, редкая специальность — преподаватель техники речи, а жизнь — самая обыкновенная: съемная квартира, невысокие гонорары и занятия с утра до вечера. Однажды Даша получает выгодное предложение — дать уроки преуспевающему бизнесмену Владу, участвующему в политических выборах. У героев начинается бурный роман. Но случайная встреча с женой Влада заставляет Дашу взглянуть на происходящее совсем с другой стороны. И Влад оказывается совсем не героем романа и вовсе не мужчиной мечты...	0.12184
46438	False	5000000	[Action, Comedy, Crime, Foreign]	63898.00000	Антидурь	Failing to complete an important assignment without casualties, fearless crime fighters from an elite special service agency, masters of disguise and simply fun guys "Velik" and "Koshka" were demoted to serve in a department of a Drug Enforcement Agency...	0.03979

5309 rows × 43 columns

In [ ]:

In [33]:
```
df['keywords'] = df['keywords'].astype(str)
df['production_countries'] = df['production_countries'].astype(str)
df['production_companies'] = df['production_companies'].astype(str)
```

In [34]:
```
df['production_countries'] = df['production_countries'].apply(literal_eval)
df['production_companies'] = df['production_companies'].apply(literal_eval)
```

In [35]:
```
df['overview'] = df['pstem'].astype(str)

df['overview'].apply(count_words)


dictionary_copy = dictionary.copy()
for key, value in dictionary_copy.items():
    if value == 1:
        dictionary.pop(key)
```

In [36]:
```
new_movies = df.filter(['overview', 'keywords', 'popularity'], axis=1)
new_movies_copy = df.filter(['overview', 'keywords', 'popularity'], axis=1)
```

In [37]:
```
new_movies['popularity'] = pd.to_numeric(new_movies['popularity'])
new_movies_copy['popularity'] = pd.to_numeric(new_movies_copy['popularity'])
```

In [38]:
```
# pd.cut(new_movies.popularity, bins=10, right=False)
new_movies
new_movies['len'] = df.apply(lambda row: len(row.keywords), axis=1)
new_movies = new_movies[new_movies.len > 2]
new_movies.drop(columns = "len", axis=1)
```

Out[38]:

	overview	keywords	popularity
0	led by woody, andy' toy live happili in hi room until andy' birthday bring buzz lightyear onto the scene. afraid of lose hi place in andy' heart, woodi plot against buzz. but when circumst separ buzz and woodi from their owner, the duo eventu learn to put asid their differences.	['jealousi', 'toy', 'boy', 'friendship', 'friend', 'rivalri', 'newtoy', 'toycomestolif']	21.94694
1	when sibl judi and peter discov an enchant board game that open the door to a magic world, they unwittingli invit alan -- an adult who' been trap insid the game for 26 year -- into their live room. alan' onli hope for freedom is to finish the game, which prove riski as all three find themselv run from giant rhinoceroses, evil monkey and other terrifi creatures.	['boardgam', 'disappear', "basedonchildren'sbook", 'newhom', 'giantinsect']	17.01554
3	cheat on, mistreat and step on, the women are hold their breath, wait for the elus "good man" to break a string of less-than-stellar lovers. friend and confid vannah, bernie, glo and robin talk it all out, determin to find a better way to breathe.	['basedonnovel', 'interracialrelationship', 'singlemoth', 'divorc', 'chickflick']	3.85949

	overview	keywords	popularity
5	obsess master thief, neil mccauley lead a top-notch crew on variou insan heist throughout lo angel while a mental unstabl detective, vincent hanna pursu him without rest. each man recogn and respect the abil and the dedic of the other even though they are awar their cat-and-mous game may end in violence.	['robberi', 'detect', 'bank', 'obsess', 'chase', 'shoot', 'thief', 'honor', 'murder', 'suspens', 'heist', 'betray', 'money', 'gang', 'catandmous', 'criminalmastermind', 'cultfilm', 'ex-con', 'heistmovi', 'onelastjob', 'loner', 'neo-noir', 'gunfight', 'crimeep']	17.92493
8	intern action superstar jean claud van damm team with power booth in a tension-packed, suspens thriller, set against the back-drop of a stanley cup game.van damm portray a father whose daughter is suddenli taken dure a championship hockey game. with the captor demand a billion dollar by game' end, van damm frantic set a plan in motion to rescu hi daughter and abort an impend explos befor the final buzzer...	['terrorist', 'hostag', 'explos']	5.23158
...	...	...	...
45987	pete is a footbal enthusiast, who play as a goalkeep for FC heman, a team play in the lowest possibl league. hi girlfriend, anna, hate the whole sport. pete and hi teammat are plan to travel to watch the footbal world cup held in germany. anna is not excit about pete' plan to leav her alon for the summer. therefor anna decid to present a challeng to pete: she will form a team from the wive and girlfriend of the FC heman players, and then the women' team (fc venus) would play against FC heman. If the women' team wins, the men will have to give up football, and if the men' team wins, the women will never complain about their hobby.	['sport', 'malefemalerelationship', 'soccer']	0.94751
46031	the last gunslinger, roland deschain, ha been lock in an etern battl with walter o'dim, also known as the man in black, determin to prevent him from toppl the dark tower, which hold the univers together. with the fate of the world at stake, good and evil will collid in the ultim battl as onli roland can defend the tower from the man in black.	['gunsling', 'basedonnovel']	50.90359
46156	gene, a multi-expression emoji, set out on a journey to becom a normal emoji.	['smartphon']	33.69460
46184	An fbi agent team with the town' veteran game tracker to investig a murder that occur on a nativ american reservation.	['rape', 'mountain', 'gun', 'investig', 'murder', 'nativeamerican', 'shootout', 'photograph', 'violenc', 'fbiag', 'binocular', 'snowmobil']	40.79677
46267	corrupt polic and politician target a comput engin for tri to better the live of less privileg citizens.	['corruptpolitician']	1.32359

4936 rows × 3 columns

In [39]:
```
new_movies.popularity.describe()
```

Out[39]:
```
count    4936.00000
mean        9.73542
std         9.50165
min         0.03058
```

```
25%         5.95730
50%         8.69460
75%        11.84313
max       228.03274
Name: popularity, dtype: float64
```

In [40]:
```
new_movies = new_movies[new_movies['popularity'].between(0, 60)]
new_movies_copy = new_movies[new_movies['popularity'].between(0, 60)]

# This is to ensure that when this model goes into the real world new dataset
```

In [ ]:

In [41]:
```
pd.set_option('display.max_colwidth', 150000) #important for getting all the
pd.set_option('display.max_columns', 999)
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from IPython.display import Image
import pydotplus
```

In [42]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [43]:
```
import re
from sklearn.feature_extraction import text
skl_stopwords = text.ENGLISH_STOP_WORDS
my_stopwords = list(skl_stopwords) + []
# ["man", "story","finds", "takes", "hi", "ha", "thi", "come", "becom", "make
```

In [44]:
```
tfidf1 = TfidfVectorizer(use_idf=True, norm=None,  stop_words=my_stopwords, m
tf1_dm = tfidf1.fit_transform(new_movies.keywords)
print(new_movies.keywords)
pd.DataFrame(tf1_dm.toarray(),columns = tfidf1.get_feature_names())
```

```
0
['jealousi', 'toy', 'boy', 'friendship', 'friend', 'rivalri', 'newtoy', 'toyco
mestolif']
1
['boardgam', 'disappear', "basedonchildren'sbook", 'newhom', 'giantinsect']
3
['basedonnovel', 'interracialrelationship', 'singlemoth', 'divorc', 'chickflic
k']
5         ['robberi', 'detect', 'bank', 'obsess', 'chase', 'shoot', 'thief', 'h
onor', 'murder', 'suspens', 'heist', 'betray', 'money', 'gang', 'catandmous',
'criminalmastermind', 'cultfilm', 'ex-con', 'heistmovi', 'onelastjob', 'lone
r', 'neo-noir', 'gunfight', 'crimeep']
8
['terrorist', 'hostag', 'explos']

...
45987
['sport', 'malefemalerelationship', 'soccer']
46031
['gunsling', 'basedonnovel']
46156
['smartphon']
46184
['rape', 'mountain', 'gun', 'investig', 'murder', 'nativeamerican', 'shootou
t', 'photograph', 'violenc', 'fbiag', 'binocular', 'snowmobil']
46267
['corruptpolitician']
Name: keywords, Length: 4918, dtype: object
```

Out[44]:

	3d	aftercreditssting	airplan	alcohol	alien	anim	assassin	basedoncom
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

	3d	aftercreditssting	airplan	alcohol	alien	anim	assassin	basedoncom
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
...	...	...	...	...	...	...	...	...
4913	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4914	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4915	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4916	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4917	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

4918 rows × 83 columns

In [45]:

```
from sklearn.feature_extraction.text import CountVectorizer
import math

countid = CountVectorizer(binary=True, stop_words=my_stopwords, min_df = 0.01
countid_dm = countid.fit_transform(new_movies.keywords) #apply the transforma

print(type(countid_dm))
print(countid_dm.shape)
pd.DataFrame(countid_dm.toarray(),columns = countid.get_feature_names())
```

```
<class 'scipy.sparse.csr.csr_matrix'>
(4918, 84)
```

Out[45]:

	3d	aftercreditssting	aftercreditssting duringcreditssting	airplan	alcohol	alien	anim	assassin	basedc
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
4913	0	0	0	0	0	0	0	0	
4914	0	0	0	0	0	0	0	0	
4915	0	0	0	0	0	0	0	0	
4916	0	0	0	0	0	0	0	0	
4917	0	0	0	0	0	0	0	0	

4918 rows × 84 columns

In [46]:

```
tfidf2 = TfidfVectorizer(use_idf=True, norm=None, stop_words=my_stopwords, mi
tf2_dm = tfidf2.fit_transform(new_movies.overview)
print(new_movies.overview)
pd.DataFrame(tf2_dm.toarray(),columns = tfidf2.get_feature_names())
```

0

```
led by woody, andy' toy live happili in hi room until andy' birthday bring buz
z lightyear onto the scene. afraid of lose hi place in andy' heart, woodi plot
against buzz. but when circumst separ buzz and woodi from their owner, the duo
eventu learn to put asid their differences.
1
when sibl judi and peter discov an enchant board game that open the door to a
magic world, they unwittingli invit alan -- an adult who' been trap insid the
game for 26 year -- into their live room. alan' onli hope for freedom is to fi
nish the game, which prove riski as all three find themselv run from giant rhi
noceroses, evil monkey and other terrifi creatures.
3
cheat on, mistreat and step on, the women are hold their breath, wait for the
elus "good man" to break a string of less-than-stellar lovers. friend and conf
id vannah, bernie, glo and robin talk it all out, determin to find a better wa
y to breathe.
5
obsess master thief, neil mccauley lead a top-notch crew on variou insan heist
throughout lo angel while a mental unstabl detective, vincent hanna pursu him
without rest. each man recogn and respect the abil and the dedic of the other
even though they are awar their cat-and-mous game may end in violence.
8
intern action superstar jean claud van damm team with power booth in a tension
-packed, suspens thriller, set against the back-drop of a stanley cup game.van
damm portray a father whose daughter is suddenli taken dure a championship hoc
key game. with the captor demand a billion dollar by game' end, van damm frant
ic set a plan in motion to rescu hi daughter and abort an impend explos befor
the final buzzer...

...
45987    pete is a footbal enthusiast, who play as a goalkeep for FC heman, a
team play in the lowest possibl league. hi girlfriend, anna, hate the whole sp
ort. pete and hi teammat are plan to travel to watch the footbal world cup hel
d in germany. anna is not excit about pete' plan to leav her alon for the summ
er. therefor anna decid to present a challeng to pete: she will form a team fr
om the wive and girlfriend of the FC heman players, and then the women' team
(fc venus) would play against FC heman. If the women' team wins, the men will
have to give up football, and if the men' team wins, the women will never comp
lain about their hobby.
46031
the last gunslinger, roland deschain, ha been lock in an etern battl with walt
er o'dim, also known as the man in black, determin to prevent him from toppl t
he dark tower, which hold the univers together. with the fate of the world at
stake, good and evil will collid in the ultim battl as onli roland can defend
the tower from the man in black.
46156
gene, a multi-expression emoji, set out on a journey to becom a normal emoji.
46184
An fbi agent team with the town' veteran game tracker to investig a murder tha
t occur on a nativ american reservation.
46267
corrupt polic and politician target a comput engin for tri to better the live
of less privileg citizens.
Name: overview, Length: 4918, dtype: object
```

Out[46]:

	agent	american	attempt	base	battl	becom	befor	begin	best	
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.0
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.0
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.0
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.0
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	4.14900	0.00000	0.00000	0.0
...	...	...	...	...	...	...	...	...	...	
4913	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.0
4914	0.00000	0.00000	0.00000	0.00000	8.92782	0.00000	0.00000	0.00000	0.00000	0.0

	agent	american	attempt	base	battl	becom	befor	begin	best	
4915	0.00000	0.00000	0.00000	0.00000	0.00000	3.30238	0.00000	0.00000	0.00000	0.0
4916	4.43827	4.02022	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.0
4917	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.0

4918 rows × 99 columns

In [47]:

```
countid2 = CountVectorizer(binary=True, stop_words=my_stopwords, min_df = 0.0
countid2_dm = countid2.fit_transform(new_movies.overview) #apply the transfor

print(type(countid2_dm))
print(countid2_dm.shape)
pd.DataFrame(countid2_dm.toarray(),columns = countid2.get_feature_names())
```

```
<class 'scipy.sparse.csr.csr_matrix'>
(4918, 102)
```

Out[47]:

	agent	american	attempt	base	battl	becom	befor	begin	best	boy	bring	brother
0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
4913	0	0	0	0	0	0	0	0	0	0	0	0
4914	0	0	0	0	1	0	0	0	0	0	0	0
4915	0	0	0	0	0	1	0	0	0	0	0	0
4916	1	1	0	0	0	0	0	0	0	0	0	0
4917	0	0	0	0	0	0	0	0	0	0	0	0

4918 rows × 102 columns

In [ ]:

In [48]:

```
from scipy.sparse import hstack
combined_features = hstack((tf1_dm, tf2_dm))
combined_count_features = hstack((countid_dm, countid2_dm))
combined_mix_features = hstack((countid_dm, tf2_dm))
```

In [49]:

```
import math

combined_feature_names = tfidf1.get_feature_names() + tfidf2.get_feature_name
print(len(combined_feature_names))
```

```
182
```

In [50]:

```
from scipy.stats import zscore

new_movies['popularity'] = zscore(new_movies['popularity'])
```

In [51]:

```
bin_labels_5 = ['Not Popular (Trim Capital Allocation)', 'Maybe Popular (Deep
```

```
new_movies['popularity'] = pd.qcut(new_movies.popularity, q=[0, .334, .667, 1
```

In [52]:
```
score_dummy = pd.get_dummies(new_movies['popularity'])
```

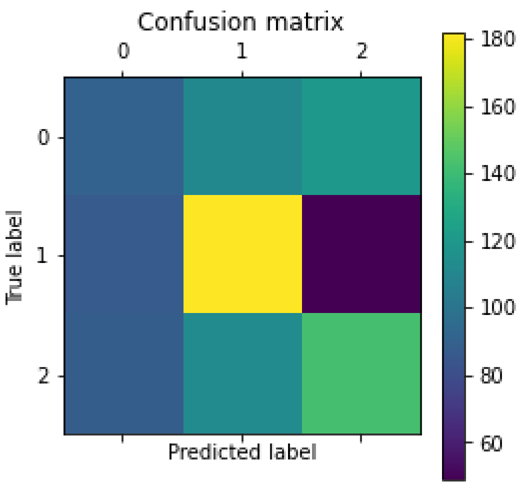
In [53]:
```
from sklearn.model_selection import train_test_split
X = combined_features.toarray()
y = new_movies['popularity'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

In [54]:
```
# define a fancy confusion matrix
def create_cm(t1, t2):
    cm = confusion_matrix(t1, t2)
    plt.matshow(cm)
    plt.title('Confusion matrix')
    plt.colorbar()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()
    print(cm)
```

In [55]:
```
# Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
# fit a Naive Bayes model to the data
model = MultinomialNB()
model.fit(X_train, y_train)


NBnews_predicted = model.predict(X_test)
NB_expected = y_test
```

In [56]:
```
print(classification_report(NB_expected, NBnews_predicted))
create_cm(NB_expected, NBnews_predicted)
```

```
                                      precision    recall  f1-score   support

                Maybe Popular (Deepdive)       0.34      0.28      0.31       322
Not Popular (Trim Capital Allocation)       0.45      0.57      0.50       318
Trending (Acquire/Increase Marketing)       0.46      0.42      0.44       344

                                accuracy                           0.42       984
                               macro avg       0.42      0.42      0.42       984
                            weighted avg       0.42      0.42      0.42       984
```




```
[[ 91 111 120]
 [ 87 182  49]
 [ 88 113 143]]
```

In [57]:
```
# combined_features = hstack((combined_features, score_dummy))
combined_features
```

Out[57]:
```
<4918x182 sparse matrix of type '<class 'numpy.float64'>'
	with 35856 stored elements in COOrdinate format>
```

In [58]:
```
#import scipy.sparse
combined_features = pd.DataFrame.sparse.from_spmatrix(combined_features)

X = combined_features
y = score_dummy

from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(criterion = "entropy", random_state = 12345, min_
dt.fit(X, y)

combined_features
```

Out[58]:

	0	1	2	3	4	5	6	7	8	
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	3.93252	0.00
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00
...	...	...	...	...	...	...	...	...	...	
4913	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00
4914	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	3.93252	0.00
4915	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00
4916	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00
4917	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00

4918 rows × 182 columns

In [59]:
```
# what is the shape of our tree
print(dt.tree_.max_depth) #number of split levels
print(dt.tree_.n_leaves) #total number of leaves
```

```
163
1932
```

In [60]:
```
# Sheeeessssh, that's a lot of levels.
from sklearn import tree
from matplotlib import pyplot as plt
plt.figure(figsize=(400, 400))
#tree.plot_tree(dt, feature_names = combined_feature_names,  filled=True)
#plt.show()
```

Out[60]:
```
<Figure size 28800x28800 with 0 Axes>
```
```
<Figure size 28800x28800 with 0 Axes>
```

In [61]:
```
from sklearn.tree import export_text
text_tree = export_text(dt, feature_names = list(combined_feature_names))
print(text_tree)
```

```
|--- independentfilm <= 2.01
|   |--- dystopia <= 2.23
|   |   |--- superhero <= 2.67
```

```
|   |   |   |--- duringcreditssting <= 1.90
|   |   |   |   |--- magic <= 2.66
|   |   |   |   |   |--- violenc <= 2.10
|   |   |   |   |   |   |--- rescu <= 2.75
|   |   |   |   |   |   |   |--- 3d <= 2.50
|   |   |   |   |   |   |   |   |--- assassin <= 2.57
|   |   |   |   |   |   |   |   |   |--- world <= 1.55
|   |   |   |   |   |   |   |   |   |   |--- anim <= 2.61
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 153
|   |   |   |   |   |   |   |   |   |   |--- anim >  2.61
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |--- world >  1.55
|   |   |   |   |   |   |   |   |   |   |--- peopl <= 2.21
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 47
|   |   |   |   |   |   |   |   |   |   |--- peopl >  2.21
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |--- assassin >  2.57
|   |   |   |   |   |   |   |   |   |--- dure <= 1.99
|   |   |   |   |   |   |   |   |   |   |--- wa <= 1.99
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |   |--- wa >  1.99
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- dure >  1.99
|   |   |   |   |   |   |   |   |   |   |--- secret <= 2.07
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- secret >  2.07
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- 3d >  2.50
|   |   |   |   |   |   |   |   |--- son <= 2.00
|   |   |   |   |   |   |   |   |   |--- becom <= 1.65
|   |   |   |   |   |   |   |   |   |   |--- fight <= 2.61
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 16
|   |   |   |   |   |   |   |   |   |   |--- fight >  2.61
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- becom >  1.65
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- son >  2.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- rescu >  2.75
|   |   |   |   |   |   |   |--- come <= 1.85
|   |   |   |   |   |   |   |   |--- past <= 2.22
|   |   |   |   |   |   |   |   |   |--- meet <= 1.96
|   |   |   |   |   |   |   |   |   |   |--- town <= 1.98
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |--- town >  1.98
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- meet >  1.96
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- past >  2.22
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- come >  1.85
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- violenc >  2.10
|   |   |   |   |   |   |--- shootout <= 2.76
|   |   |   |   |   |   |   |--- world <= 1.55
|   |   |   |   |   |   |   |   |--- alcohol <= 2.51
|   |   |   |   |   |   |   |   |   |--- daughter <= 6.22
|   |   |   |   |   |   |   |   |   |   |--- thing <= 2.22
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 29
|   |   |   |   |   |   |   |   |   |   |--- thing >  2.22
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- daughter >  6.22
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- alcohol >  2.51
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- world >  1.55
|   |   |   |   |   |   |   |   |--- run <= 2.11
|   |   |   |   |   |   |   |   |   |--- becom <= 1.65
|   |   |   |   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |   |   |   |   |--- becom >  1.65
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- run >  2.11
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- shootout >  2.76
|   |   |   |   |   |   |   |--- cia <= 2.78
|   |   |   |   |   |   |   |   |--- basedonnovel <= 1.97
|   |   |   |   |   |   |   |   |   |--- team <= 2.03
|   |   |   |   |   |   |   |   |   |   |--- remak <= 2.57
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- remak >  2.57
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- team >  2.03
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- basedonnovel >  1.97
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- cia >  2.78
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- magic >  2.66
|   |   |   |   |   |--- woman <= 1.90
|   |   |   |   |   |   |--- film <= 1.95
|   |   |   |   |   |   |   |--- evil <= 6.65
|   |   |   |   |   |   |   |   |--- sex <= 2.32
|   |   |   |   |   |   |   |   |   |--- tri <= 1.83
|   |   |   |   |   |   |   |   |   |   |--- use <= 2.16
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |--- use >  2.16
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- tri >  1.83
|   |   |   |   |   |   |   |   |   |   |--- hi <= 0.91
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- hi >  0.91
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- sex >  2.32
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- evil >  6.65
|   |   |   |   |   |   |   |   |--- befor <= 2.07
|   |   |   |   |   |   |   |   |   |--- daughter <= 2.07
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- daughter >  2.07
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- befor >  2.07
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- film >  1.95
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- woman >  1.90
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- duringcreditssting >  1.90
|   |   |   |   |--- friendship <= 2.35
|   |   |   |   |   |--- love <= 1.68
|   |   |   |   |   |   |--- decid <= 2.06
|   |   |   |   |   |   |   |--- star <= 2.15
|   |   |   |   |   |   |   |   |--- murder <= 1.93
|   |   |   |   |   |   |   |   |   |--- man <= 1.62
|   |   |   |   |   |   |   |   |   |   |--- dream <= 2.76
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 57
|   |   |   |   |   |   |   |   |   |   |--- dream >  2.76
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- man >  1.62
|   |   |   |   |   |   |   |   |   |   |--- wife <= 1.99
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- wife >  1.99
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- murder >  1.93
|   |   |   |   |   |   |   |   |   |--- robberi <= 2.76
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- robberi >  2.76
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- star >  2.15
```

```
|   |   |   |   |   |   |   |   |--- daughter <= 2.07
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- daughter >  2.07
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- decid >  2.06
|   |   |   |   |   |   |   |--- american <= 2.01
|   |   |   |   |   |   |   |   |--- newyork <= 2.45
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- newyork >  2.45
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- american >  2.01
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- love >  1.68
|   |   |   |   |   |   |--- sport <= 2.33
|   |   |   |   |   |   |   |--- tri <= 1.83
|   |   |   |   |   |   |   |   |--- just <= 2.10
|   |   |   |   |   |   |   |   |   |--- stori <= 1.78
|   |   |   |   |   |   |   |   |   |   |--- begin <= 1.98
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- begin >  1.98
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- stori >  1.78
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- just >  2.10
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- tri >  1.83
|   |   |   |   |   |   |   |   |--- stori <= 1.78
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- stori >  1.78
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- sport >  2.33
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- friendship >  2.35
|   |   |   |   |   |--- young <= 1.59
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- young >  1.59
|   |   |   |   |   |   |--- class: 0
|   |   |--- superhero >  2.67
|   |   |   |--- parti <= 2.73
|   |   |   |   |--- use <= 2.16
|   |   |   |   |   |--- stop <= 2.22
|   |   |   |   |   |   |--- highschool <= 2.57
|   |   |   |   |   |   |   |--- end <= 2.12
|   |   |   |   |   |   |   |   |--- sequel <= 2.33
|   |   |   |   |   |   |   |   |   |--- ha <= 1.42
|   |   |   |   |   |   |   |   |   |   |--- death <= 2.08
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |--- death >  2.08
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- ha >  1.42
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- sequel >  2.33
|   |   |   |   |   |   |   |   |   |--- save <= 2.01
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- save >  2.01
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- end >  2.12
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- highschool >  2.57
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- stop >  2.22
|   |   |   |   |   |   |--- class: 2
|   |   |   |   |--- use >  2.16
|   |   |   |   |   |--- power <= 6.28
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- power >  6.28
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- parti >  2.73
|   |   |   |   |--- class: 0
```

```
|   |--- dystopia >  2.23
|   |   |--- monster <= 2.71
|   |   |   |--- day <= 1.93
|   |   |   |   |--- team <= 2.03
|   |   |   |   |   |--- new <= 4.57
|   |   |   |   |   |   |--- son <= 2.00
|   |   |   |   |   |   |   |--- human <= 2.24
|   |   |   |   |   |   |   |   |--- forc <= 1.88
|   |   |   |   |   |   |   |   |   |--- sequel <= 2.33
|   |   |   |   |   |   |   |   |   |   |--- kill <= 2.07
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 16
|   |   |   |   |   |   |   |   |   |   |--- kill >  2.07
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- sequel >  2.33
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- forc >  1.88
|   |   |   |   |   |   |   |   |   |--- friend <= 3.46
|   |   |   |   |   |   |   |   |   |   |--- onli <= 5.47
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- onli >  5.47
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- friend >  3.46
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- human >  2.24
|   |   |   |   |   |   |   |   |--- vampir <= 2.75
|   |   |   |   |   |   |   |   |   |--- world <= 1.55
|   |   |   |   |   |   |   |   |   |   |--- year <= 1.66
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- year >  1.66
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- world >  1.55
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- vampir >  2.75
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- son >  2.00
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- new >  4.57
|   |   |   |   |   |   |--- soon <= 2.01
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- soon >  2.01
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- team >  2.03
|   |   |   |   |   |--- class: 0
|   |   |   |--- day >  1.93
|   |   |   |   |--- class: 0
|   |   |--- monster >  2.71
|   |   |   |--- wa <= 1.99
|   |   |   |   |--- class: 0
|   |   |   |--- wa >  1.99
|   |   |   |   |--- class: 1
|--- independentfilm >  2.01
|   |--- brotherbrotherrelationship <= 2.69
|   |   |--- womandirector <= 1.94
|   |   |   |--- dyinganddeath <= 2.58
|   |   |   |   |--- onli <= 1.82
|   |   |   |   |   |--- hi <= 8.15
|   |   |   |   |   |   |--- life <= 4.37
|   |   |   |   |   |   |   |--- film <= 1.95
|   |   |   |   |   |   |   |   |--- man <= 1.62
|   |   |   |   |   |   |   |   |   |--- timetravel <= 2.77
|   |   |   |   |   |   |   |   |   |   |--- base <= 2.20
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 24
|   |   |   |   |   |   |   |   |   |   |--- base >  2.20
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- timetravel >  2.77
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- man >  1.62
|   |   |   |   |   |   |   |   |   |--- new <= 1.52
|   |   |   |   |   |   |   |   |   |   |--- ha <= 1.42
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- ha >  1.42
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- new >  1.52
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- film >  1.95
|   |   |   |   |   |   |   |   |--- high <= 2.06
|   |   |   |   |   |   |   |   |   |--- day <= 1.93
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- day >  1.93
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- high >  2.06
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- life >  4.37
|   |   |   |   |   |   |   |--- plan <= 2.13
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- plan >  2.13
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- hi >  8.15
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- onli >  1.82
|   |   |   |   |   |--- class: 1
|   |   |   |--- dyinganddeath >  2.58
|   |   |   |   |--- hi <= 0.91
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- hi >  0.91
|   |   |   |   |   |--- class: 0
|   |   |--- womandirector >  1.94
|   |   |   |--- suicid <= 2.66
|   |   |   |   |--- duringcreditssting <= 1.90
|   |   |   |   |   |--- becom <= 4.95
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- becom >  4.95
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- duringcreditssting >  1.90
|   |   |   |   |   |--- class: 0
|   |   |   |--- suicid >  2.66
|   |   |   |   |--- class: 0
|   |--- brotherbrotherrelationship >  2.69
|   |   |--- thing <= 2.22
|   |   |   |--- class: 0
|   |   |--- thing >  2.22
|   |   |   |--- class: 0
```

In [62]:

```python
from sklearn.model_selection import train_test_split

X = combined_features
y = score_dummy

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, ra
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(3934, 182) (984, 182) (3934, 3) (984, 3)
```

In [63]:

```python
dt = DecisionTreeClassifier(criterion = "entropy")
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)

print(dt.score(X_test, y_test))
# Only 40% accurate overall.
```

```
0.3709349593495935
```

In [64]:

```python
from sklearn.ensemble import RandomForestClassifier
# Instantiate model with 100 decision trees
movies_rf = RandomForestClassifier(random_state = 12345, n_estimators = 500)
movies_rf.fit(X_train, y_train)
```

Out[64]: RandomForestClassifier(n_estimators=500, random_state=12345)

In [65]:
```
influence = pd.Series(movies_rf.feature_importances_, index = combined_featur
influence.sort_values(inplace = True, ascending = False)
print(influence[0:19])

# Some of these words don't really give much info, a good way to remove them
```

```
hi                 0.03450
life               0.01577
ha                 0.01545
new                0.01322
man                0.01305
young              0.01270
becom              0.01215
independentfilm    0.01108
year               0.01093
love               0.01072
world              0.01071
thi                0.01057
friend             0.00992
live               0.00981
stori              0.00961
help               0.00949
set                0.00945
famili             0.00911
tri                0.00908
dtype: float64
```

In [66]:
```
movies_dt = DecisionTreeClassifier(criterion = "entropy", random_state = 1234
movies_dt.fit(X_train, y_train)
dt_pred = movies_dt.predict(X_test)

print(movies_dt.score(X_test, y_test))
```

```
0.375
```

In [67]:
```
plt.rcParams["figure.figsize"] = (50,10)
```

In [68]:
```
from wordcloud import WordCloud

# Read the whole text.
intext = (str(list(influence.index[1:30])))
# Generate a word cloud image
wordcloud = WordCloud(background_color="white", min_word_length = 2).generate

# Display the generated image:
# the matplotlib way:
import matplotlib.pyplot as plt
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

In [69]:
```
new_movies_copy['popularity'] = pd.to_numeric(new_movies_copy['popularity'])
```

In [70]:
```
bin_labels_5 = ['Not Popular (Trim Capital Allocation)', 'Trending (Acquire/I

new_movies_copy['popularity'] = pd.qcut(new_movies_copy.popularity, q=[0, .5,
```

In [71]:
```
from sklearn.model_selection import train_test_split
X = combined_features
y = new_movies_copy['popularity'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

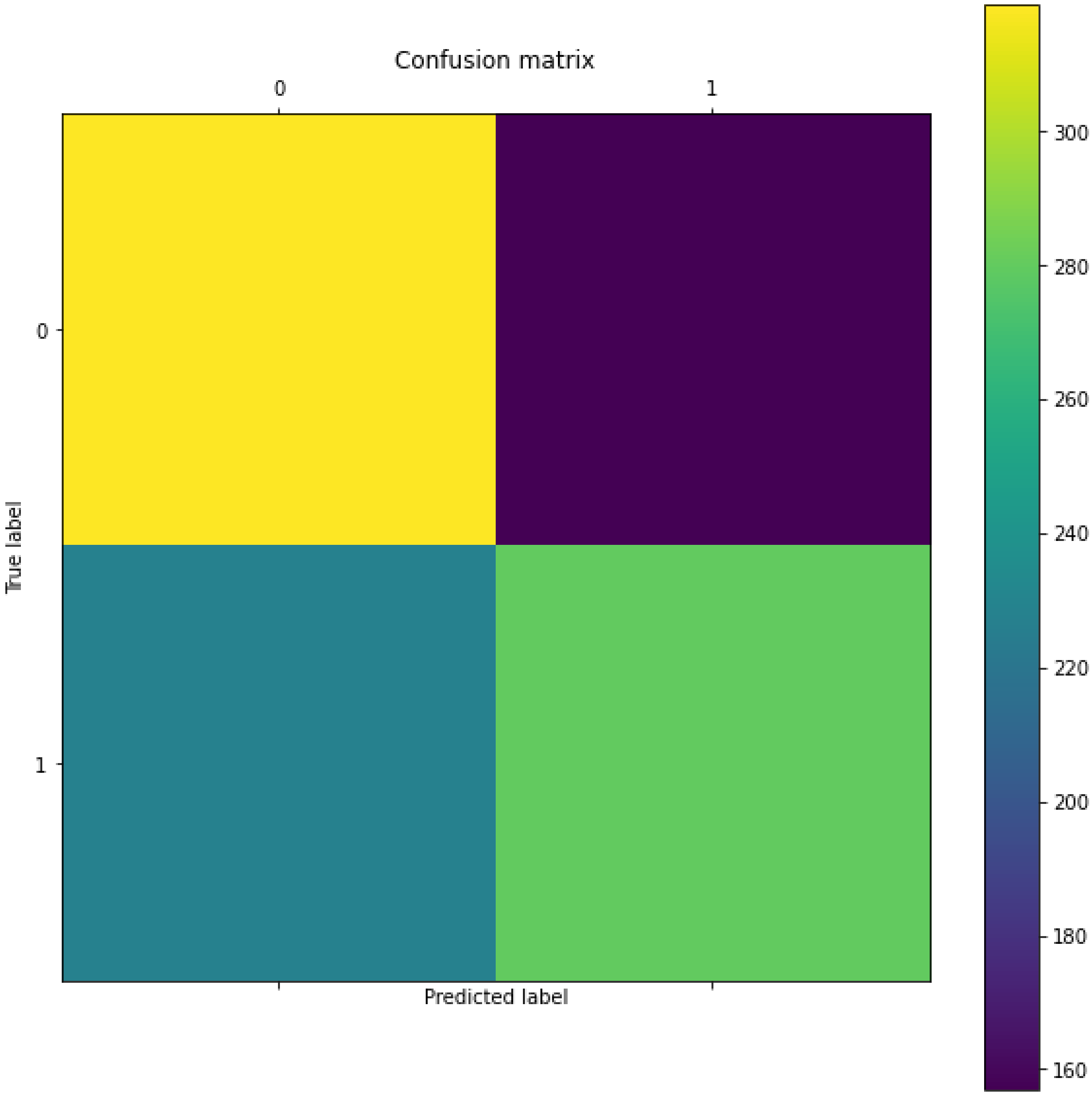
In [72]:
```
# define a fancy confusion matrix
def create_cm(t1, t2):
    cm = confusion_matrix(t1, t2)
    plt.matshow(cm)
    plt.title('Confusion matrix')
    plt.colorbar()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()
    print(cm)
```

In [73]:
```
# Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
# fit a Naive Bayes model to the data
model = MultinomialNB()
model.fit(X_train, y_train)


NBnews_predicted = model.predict(X_test)
NB_expected = y_test
```

In [74]:
```
print(classification_report(NB_expected, NBnews_predicted))
create_cm(NB_expected, NBnews_predicted)
```

```
                                        precision    recall  f1-score   support

  Not Popular (Trim Capital Allocation)      0.58      0.67      0.62       476
Trending (Acquire/Increase Marketing)      0.64      0.55      0.59       508

                               accuracy                          0.61       984
```

```
   macro avg       0.61      0.61      0.61       984
weighted avg       0.61      0.61      0.61       984
```




```
[[319 157]
 [228 280]]
```

In [75]:
```
score_dummy_new = pd.get_dummies(new_movies_copy['popularity'])
```

In [76]:
```
from sklearn.model_selection import train_test_split

X = combined_features
y = score_dummy_new

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, ra
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(3934, 182) (984, 182) (3934, 2) (984, 2)
```

In [77]:
```
from sklearn.ensemble import RandomForestClassifier
# Instantiate model with 100 decision trees
movies_rf = RandomForestClassifier(random_state = 12345)
movies_rf.fit(X_train, y_train)
```

Out[77]:
```
RandomForestClassifier(random_state=12345)
```

In [78]:
```
influence = pd.Series(movies_rf.feature_importances_, index = combined_featur
influence.sort_values(inplace = True, ascending = False)
print(influence[0:19])
```

```
hi                 0.03035
independentfilm    0.01512
life               0.01409
ha                 0.01365
new                0.01279
young              0.01268
man                0.01150
world              0.01099
love               0.01056
becom              0.01051
thi                0.01037
set                0.00995
year               0.00994
tri                0.00953
friend             0.00953
live               0.00944
film               0.00924
stori              0.00918
famili             0.00898
dtype: float64
```

In [79]:

```
from wordcloud import WordCloud

# Read the whole text.
intext = (str(list(influence.index[1:30])))
# Generate a word cloud image
wordcloud = WordCloud(background_color="white", min_word_length = 2).generate

# Display the generated image:
# the matplotlib way:
import matplotlib.pyplot as plt
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

In [80]:

```
from sklearn.model_selection import train_test_split
X = tf2_dm.toarray()
y = new_movies_copy['popularity'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

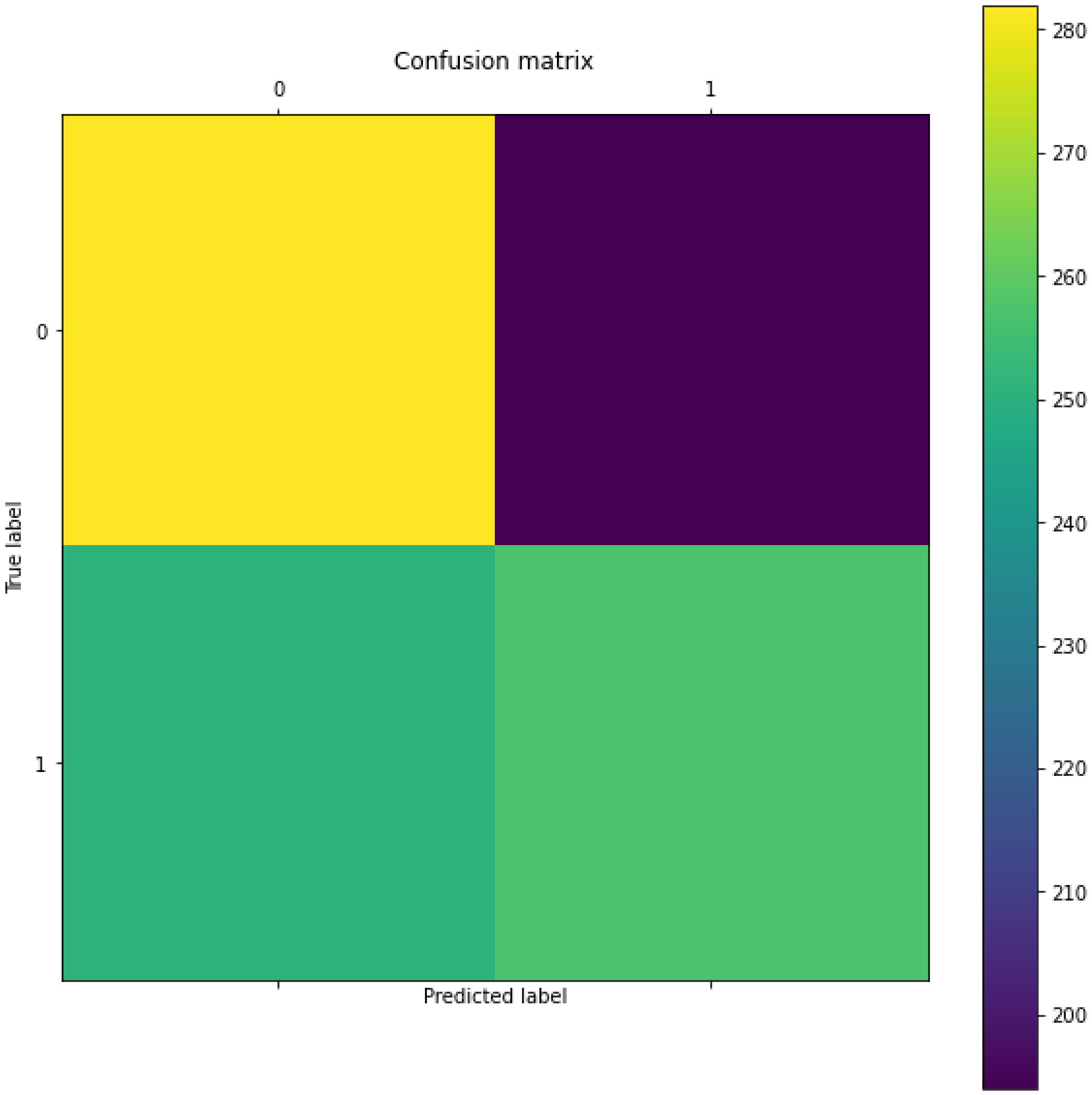
In [81]:

```
# Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
# fit a Naive Bayes model to the data
```

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


NBnews_predicted = model.predict(X_test)
NB_expected = y_test
```

In [82]:
```
print(classification_report(NB_expected, NBnews_predicted))
create_cm(NB_expected, NBnews_predicted)
```

```
                                          precision    recall  f1-score   support

Not Popular (Trim Capital Allocation)          0.53      0.59      0.56       476
Trending (Acquire/Increase Marketing)          0.57      0.51      0.54       508

                             accuracy                           0.55       984
                            macro avg          0.55      0.55      0.55       984
                         weighted avg          0.55      0.55      0.55       984
```




```
[[282 194]
 [251 257]]
```

In [83]:
```
from sklearn.model_selection import train_test_split

X = tf2_dm.toarray()
y = score_dummy_new

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, ra
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(3934, 99) (984, 99) (3934, 2) (984, 2)
```

In [84]:
```
from sklearn.ensemble import RandomForestClassifier
# Instantiate model with 100 decision trees
movies_rf = RandomForestClassifier(random_state = 12345)
movies_rf.fit(X_train, y_train)
```

Out[84]:
```
RandomForestClassifier(random_state=12345)
```

In [85]:
```
influence = pd.Series(movies_rf.feature_importances_, index = tfidf2.get_feat
influence.sort_values(inplace = True, ascending = False)
print(influence[0:19])
```

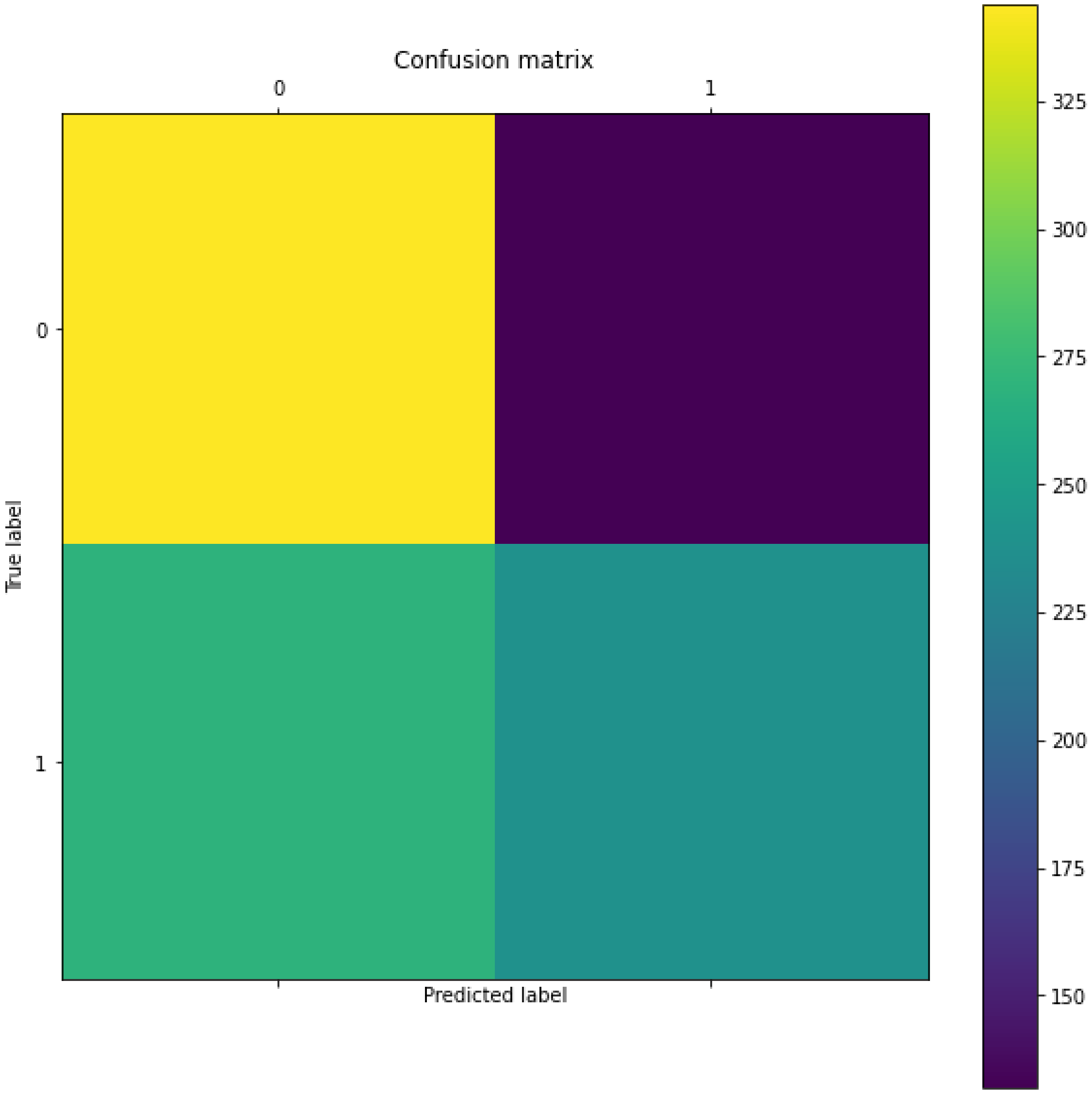
```
hi        0.04163
ha        0.02084
life      0.02006
young     0.01935
man       0.01813
new       0.01769
love      0.01536
becom     0.01536
live      0.01457
tri       0.01396
stori     0.01377
thi       0.01346
world     0.01340
famili    0.01322
year      0.01319
woman     0.01313
set       0.01256
help      0.01248
film      0.01237
dtype: float64
```

In [86]:
```
from sklearn.model_selection import train_test_split
X = tf1_dm.toarray()
y = new_movies_copy['popularity'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

In [87]:
```
# Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
# fit a Naive Bayes model to the data
model = MultinomialNB()
model.fit(X_train, y_train)


NBnews_predicted = model.predict(X_test)
NB_expected = y_test
```

In [88]:
```
print(classification_report(NB_expected, NBnews_predicted))
create_cm(NB_expected, NBnews_predicted)
```

```
                                        precision    recall  f1-score   support

   Not Popular (Trim Capital Allocation)      0.56      0.72      0.63       476
Trending (Acquire/Increase Marketing)      0.64      0.47      0.54       508

                                accuracy                          0.59       984
                               macro avg      0.60      0.60      0.59       984
                            weighted avg      0.60      0.59      0.59       984
```




```
[[344 132]
 [269 239]]
```

In [89]:
```
from sklearn.model_selection import train_test_split

X = tf1_dm.toarray()
y = score_dummy_new

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, ra
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(3934, 83) (984, 83) (3934, 2) (984, 2)
```

In [90]:
```
from sklearn.ensemble import RandomForestClassifier
# Instantiate model with 100 decision trees
movies_rf = RandomForestClassifier(random_state = 12345)
movies_rf.fit(X_train, y_train)
```

Out[90]:
```
RandomForestClassifier(random_state=12345)
```

In [91]:
```
influence = pd.Series(movies_rf.feature_importances_, index = tfidf1.get_feat
influence.sort_values(inplace = True, ascending = False)
print(influence[0:19])
```

```
independentfilm       0.04106
murder                0.02753
basedonnovel          0.02311
duringcreditssting    0.01877
dystopia              0.01759
music                 0.01731
```

```
womandirector        0.01716
newyork              0.01669
violenc              0.01648
polic                0.01647
reveng               0.01631
love                 0.01586
prison               0.01541
sequel               0.01530
highschool           0.01505
biographi            0.01449
alcohol              0.01430
sex                  0.01377
friendship           0.01370
dtype: float64
```

In [92]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="ticks", color_codes=True)
```

In [93]:

```
#xaxis = list(influence.index[0:5])
#tf2_df = pd.DataFrame.sparse.from_spmatrix(tf2_dm)
#popscores = sns.load_dataset(tf2_df)
#sns.catplot(x=xaxis, y="popularity", data=tips)
```

In [107…

```
from wordcloud import WordCloud

# Read the whole text.
intext = (str(list(influence.index[0:30])))
# Generate a word cloud image
wordcloud = WordCloud(background_color="white", min_word_length = 2).generate

# Display the generated image:
# the matplotlib way:
import matplotlib.pyplot as plt
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

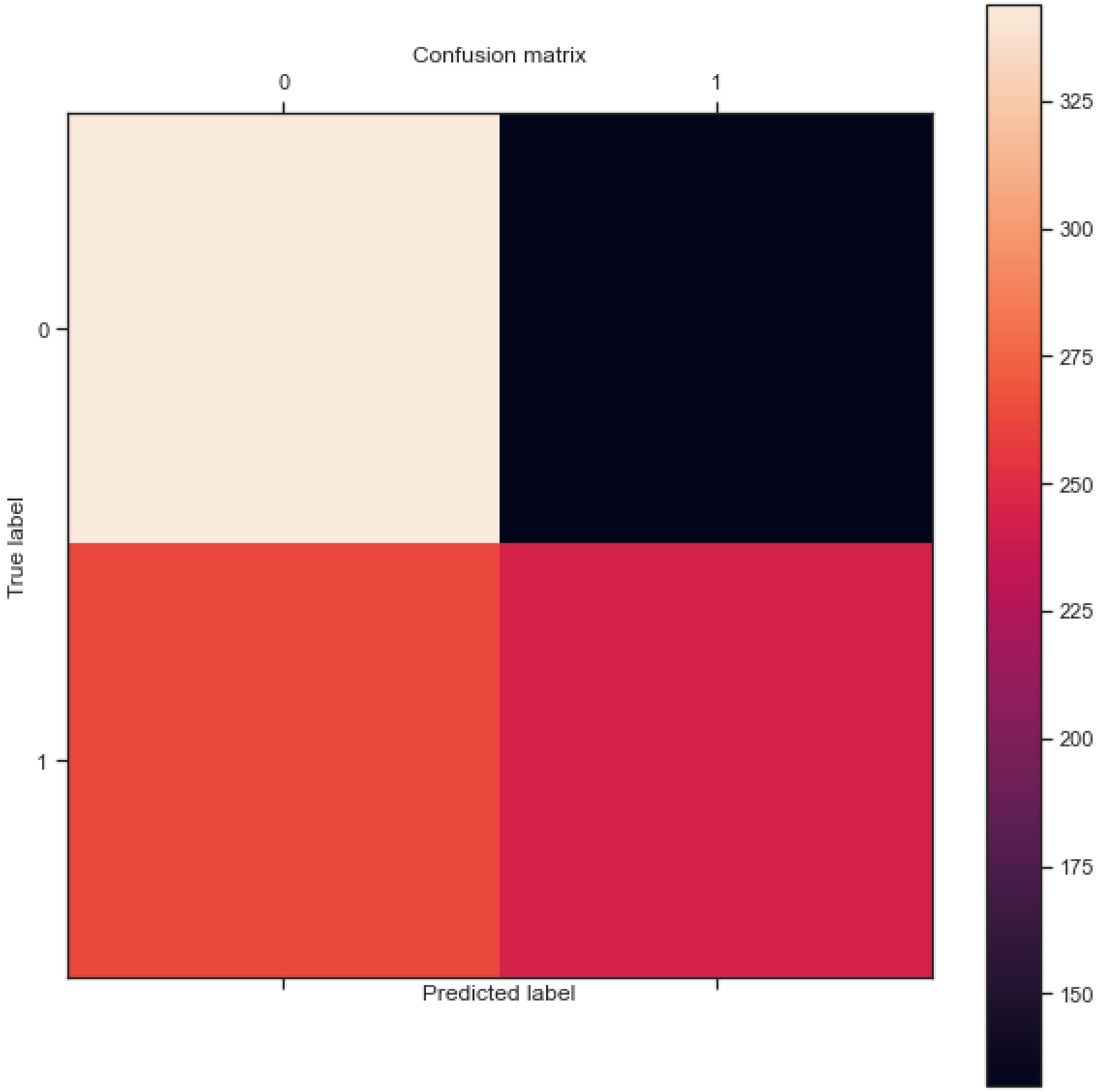
In [95]:

```
from sklearn.model_selection import train_test_split
X = countid_dm.toarray()
y = new_movies_copy['popularity'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

In [96]:
```
# Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
# fit a Naive Bayes model to the data
model = MultinomialNB()
model.fit(X_train, y_train)


NBnews_predicted = model.predict(X_test)
NB_expected = y_test
```

In [97]:
```
print(classification_report(NB_expected, NBnews_predicted))
create_cm(NB_expected, NBnews_predicted)
```

```
                                        precision    recall  f1-score   support

    Not Popular (Trim Capital Allocation)       0.57      0.72      0.64       476
Trending (Acquire/Increase Marketing)       0.65      0.48      0.55       508

                             accuracy                           0.60       984
                            macro avg       0.61      0.60      0.59       984
                         weighted avg       0.61      0.60      0.59       984
```




```
[[344 132]
 [263 245]]
```

In [98]:
```
from sklearn.ensemble import RandomForestClassifier
# Instantiate model with 100 decision trees
movies_rf = RandomForestClassifier(random_state = 12345)
movies_rf.fit(X_train, y_train)
```

Out[98]:
```
RandomForestClassifier(random_state=12345)
```

In [99]:
```
influence = pd.Series(movies_rf.feature_importances_, index = countid.get_fea
influence.sort_values(inplace = True, ascending = False)
print(influence[0:19])
```

```
independentfilm      0.04327
basedonnovel         0.03010
murder               0.02470
duringcreditssting   0.02164
dystopia             0.02124
violenc              0.01998
sequel               0.01760
polic                0.01657
womandirector        0.01637
reveng               0.01619
love                 0.01616
losangel             0.01550
biographi            0.01508
aftercreditssting    0.01422
sex                  0.01409
newyork              0.01405
prison               0.01401
friendship           0.01377
drug                 0.01366
dtype: float64
```

In [108…
```
from wordcloud import WordCloud

# Read the whole text.
intext = (str(list(influence.index[0:30])))
# Generate a word cloud image
wordcloud = WordCloud(background_color="white", min_word_length = 2).generate

# Display the generated image:
# the matplotlib way:
import matplotlib.pyplot as plt
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

In [101…
```
from sklearn.model_selection import train_test_split
X = combined_count_features.toarray()
y = new_movies_copy['popularity'].values
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

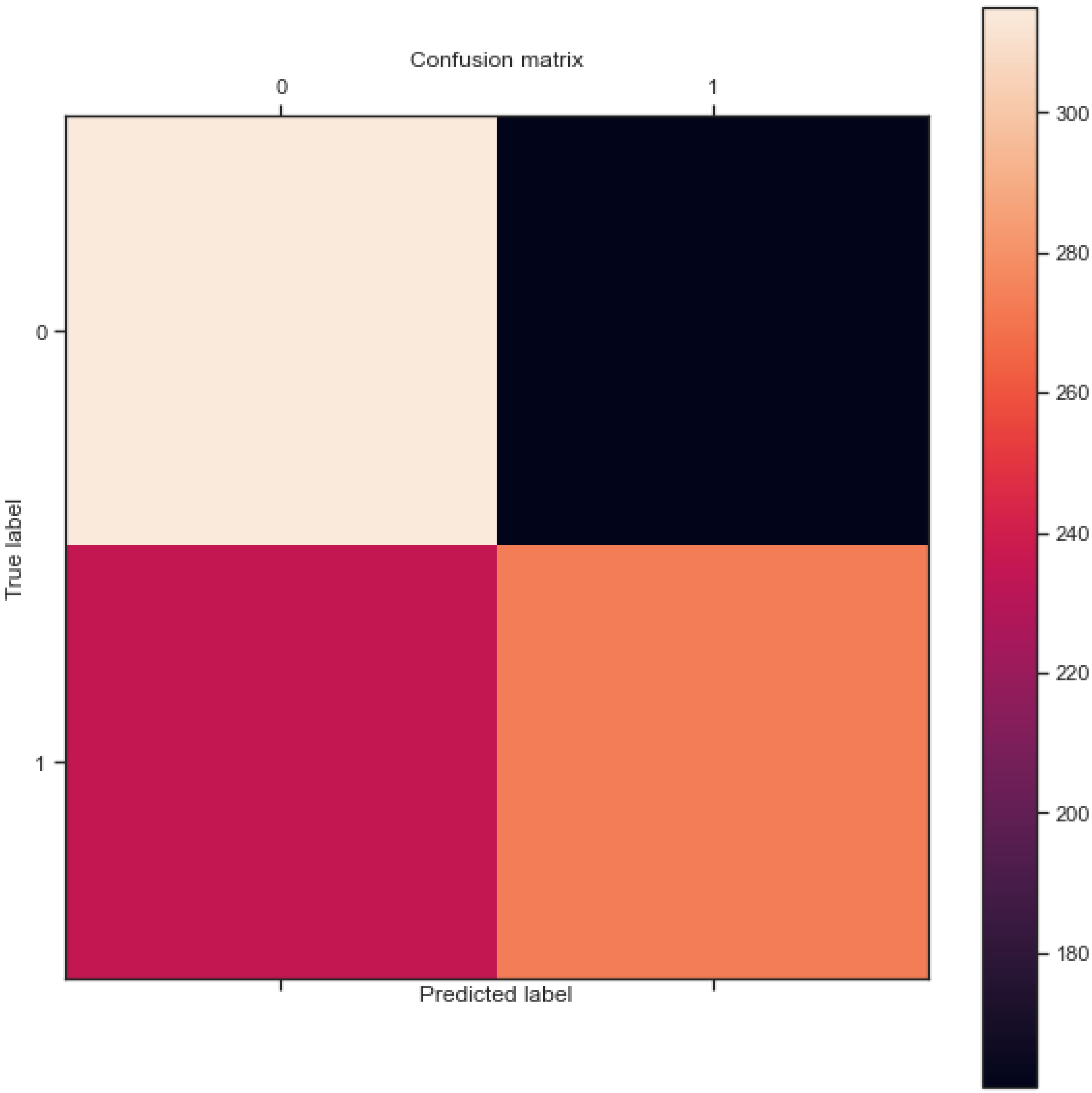
In [102…

```
# Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
# fit a Naive Bayes model to the data
model = MultinomialNB()
model.fit(X_train, y_train)


NBnews_predicted = model.predict(X_test)
NB_expected = y_test
```

In [103…

```
print(classification_report(NB_expected, NBnews_predicted))
create_cm(NB_expected, NBnews_predicted)
```

```
                                      precision    recall  f1-score   support

    Not Popular (Trim Capital Allocation)       0.57      0.66      0.61       476
Trending (Acquire/Increase Marketing)       0.63      0.54      0.58       508

                             accuracy                           0.60       984
                            macro avg       0.60      0.60      0.60       984
                         weighted avg       0.60      0.60      0.60       984
```




```
[[315 161]
 [235 273]]
```

In [104…

```
from sklearn.model_selection import train_test_split
X = combined_mix_features.toarray()
y = new_movies_copy['popularity'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

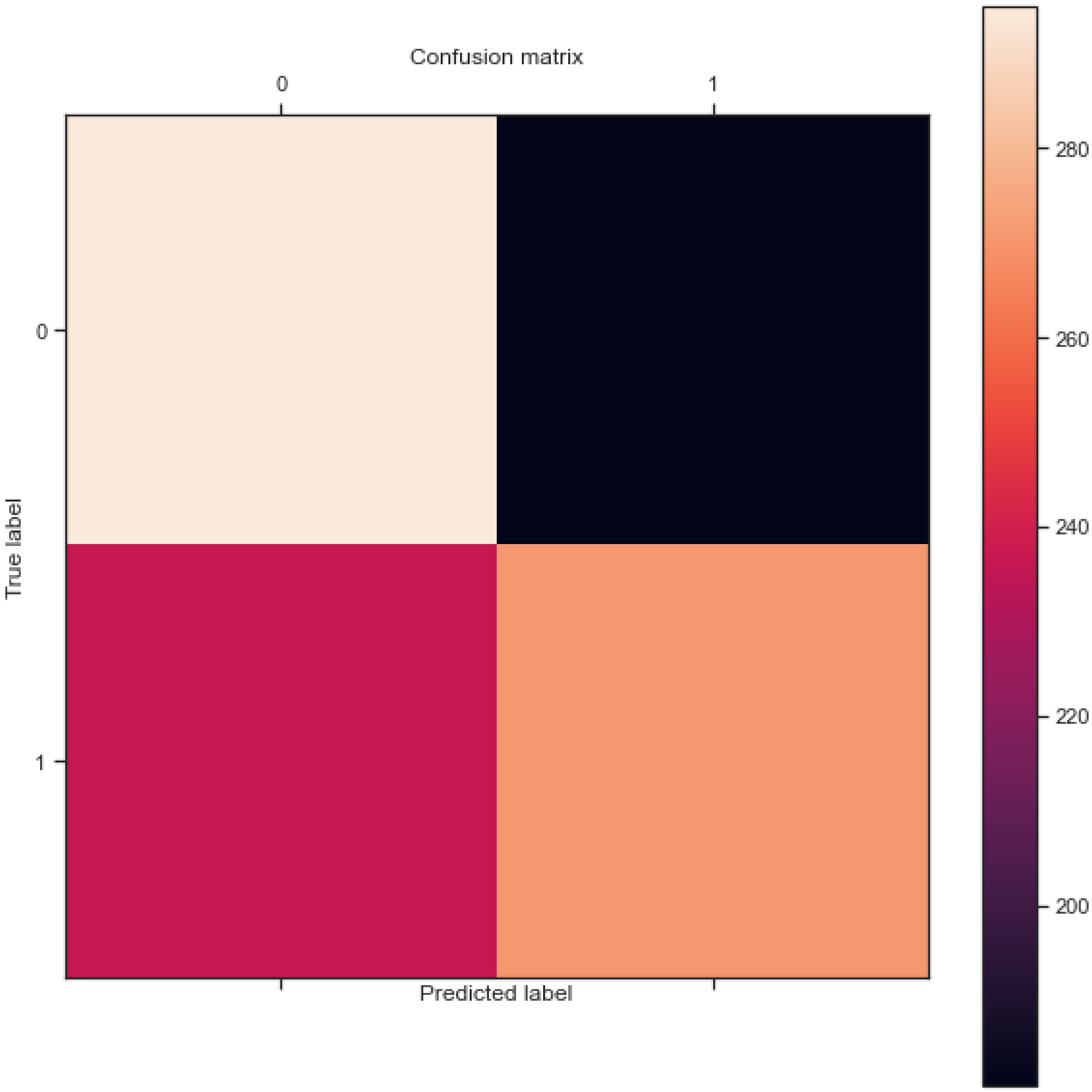
In [105…

```
# Multinomial Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
# fit a Naive Bayes model to the data
model = MultinomialNB()
model.fit(X_train, y_train)


NBnews_predicted = model.predict(X_test)
NB_expected = y_test
```

In [106…

```
print(classification_report(NB_expected, NBnews_predicted))
create_cm(NB_expected, NBnews_predicted)
```

```
                                        precision    recall  f1-score   support

Not Popular (Trim Capital Allocation)        0.55      0.62      0.59       476
  Trending (Acquire/Increase Marketing)      0.60      0.53      0.56       508

                             accuracy                           0.58       984
                            macro avg       0.58      0.58      0.57       984
                         weighted avg       0.58      0.58      0.57       984
```



```
[[295 181]
 [237 271]]
```

In [109…

```
from sklearn.ensemble import RandomForestClassifier
# Instantiate model with 100 decision trees
movies_rf = RandomForestClassifier(random_state = 12345)
movies_rf.fit(X_train, y_train)
```

Out[109…

```
RandomForestClassifier(random_state=12345)
```