movie plot Movie_Tag_Prediction_all_tags_ml no verbose

December 4, 2019

1 Importing Required Libraries

[0]:
```
!pip install --upgrade jupyterhub
```

[0]:
```
# !pip install Opencv-python
# !pip install seaborn
# !pip install sklearn
# !pip install nltk
# !pip install wordcloud
# !pip install prettytable
```

[0]:
```
import os
import re
import cv2
import sys
import nltk
import shutil
import string
import pickle
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from prettytable import from_html_one
from PIL import Image
from tqdm import tqdm
from zipfile import ZipFile
from prettytable import PrettyTable
from bs4 import BeautifulSoup
from scipy.sparse import hstack
from wordcloud import WordCloud
from nltk.corpus import stopwords
from nltk.stem.snowball import SnowballStemmer
from nltk import Counter # Used to count number of times a word repeated
# from sqlalchemy import create_engine
from sklearn.model_selection import GridSearchCV, train_test_split
```

```
from sklearn import metrics
from sklearn.linear_model import SGDClassifier, LogisticRegression
from sklearn.naive_bayes import MultinomialNB, GaussianNB
from sklearn.multiclass import OneVsRestClassifier
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, hamming_loss, precision_score,␣
↪recall_score,f1_score
```

[0]:
```
from google.colab import drive
drive.mount('gdrive',force_remount=True)
```

2 Business Problem

2.1 Discription

Social tagging of movies reveals a wide range of heterogeneous information about movies, like the genre, plot structure, soundtracks, metadata, visual and emotional experiences. Such information can be valuable in building automatic systems to create tags for movies. Automatic tagging systems can help recommendation engines to improve the retrieval of similar movies as well as help viewers to know what to expect from a movie in advance. In this paper, we set out to the task of collecting a corpus of movie plot synopses and tags. We describe a methodology that enabled us to build a fine-grained set of around 70 tags exposing heterogeneous characteristics of movie plots and the multi-label associations of these tags with some 14K movie plot synopses. We investigate how these tags correlate with movies and the flow of emotions throughout different types of movies. Finally, we use this corpus to explore the feasibility of inferring tags from plot synopses. We expect the corpus will be useful in other tasks where analysis of narratives is relevant.

2.2 Problem Statemtent

Suggest the tags based on the synopses and title the movie.

2.3 Source

Kaggle :https://www.kaggle.com/cryptexcode/mpst-movie-plot-synopses-with-tags

2.4 Real World Objectives:

Predicting as many tags as possible for a movie

Predicting wrong tags may lead to bad coustomer experience

No time restrictions

3 Machine Lerning Problem

3.1 Data

[0]:
```
# Reading Data

data = pd.read_csv('mpst_full_data.csv')
# data.to_csv('mpst_full_data.csv')
data.to_csv('data.csv', index = False)
```

[0]:
```
data.head()
```

[0]:
```
      imdb_id  … synopsis_source
0  tt0057603  …             imdb
1  tt1733125  …             imdb
2  tt0033045  …             imdb
3  tt0113862  …             imdb
4  tt0086250  …             imdb

[5 rows x 6 columns]
```

Column Discription

imdb_id:- IMDB Movie Id

title:- Name of the Movie

plot_synopsis:- Summary of the movie

tags:- Tags of the movie

split:- Represents train, test or validation data

synopsis_source:- From where summary of the movie is collected from

3.2 Type of Machine Learning Problem

This is a kind of multilabel classification problem, multilabel means each data point will have a bunch of labels, unlike each data point, having one label, for example, each movie will have many tags like violence, cult, gothic, cruelty, sadist, feel-good, revenge, inspiring, romantic, stupid.

3.3 Performance Metric

Micro-Averaged F1-Score (Mean F Score) : The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

F1 = 2 * (precision * recall) / (precision + recall)

In the multi-class and multi-label case, this is the weighted average of the F1 score of each class.

'Micro f1 score': Calculate metrics globally by counting the total true positives, false negatives and false positives. This is a better metric when we have class imbalance.

'Macro f1 score': Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

https://sebastianraschka.com/faq/docs/multiclass-metric.html

http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html Hamming loss : The Hamming loss is the fraction of labels that are incorrectly predicted. https://en.wikipedia.org/wiki/Multi-label_classification#Statistics_and_evaluation_metrics

4 Exploratory Data Analysis

4.1 Loading Data

[0]:
```
data = pd.read_csv('data.csv')
```

[0]:
```
data.head()
```

[0]:
```
      imdb_id  … synopsis_source
0  tt0057603  …            imdb
1  tt1733125  …            imdb
2  tt0033045  …            imdb
3  tt0113862  …            imdb
4  tt0086250  …            imdb

[5 rows x 6 columns]
```

Checking number of duplicates in each data set

[0]:
```
print(pd.DataFrame(data.duplicated().values, columns=['d']).d.value_counts())
```

```
False    14828
Name: d, dtype: int64
```

Data sets do not have duplicate rows

[0]:
```
train_size,test_size,val_size  = data.split.value_counts()
print('Total number of data points in train data :', train_size)
print('Total number of data points in test data :', test_size)
print('Total number of data points in validation data :', val_size)
```

```
Total number of data points in train data : 9489
Total number of data points in test data : 2966
Total number of data points in validation data : 2373
```

4.2 Tag Analysis

[0]:
```
print(data.tags[0])
print(data.tags[3])
print(data.tags[10])
```

```
cult, horror, gothic, murder, atmospheric
inspiring, romantic, stupid, feel-good
revenge, neo noir, murder, violence, flashback
```

Each tag is separated by cama (',')

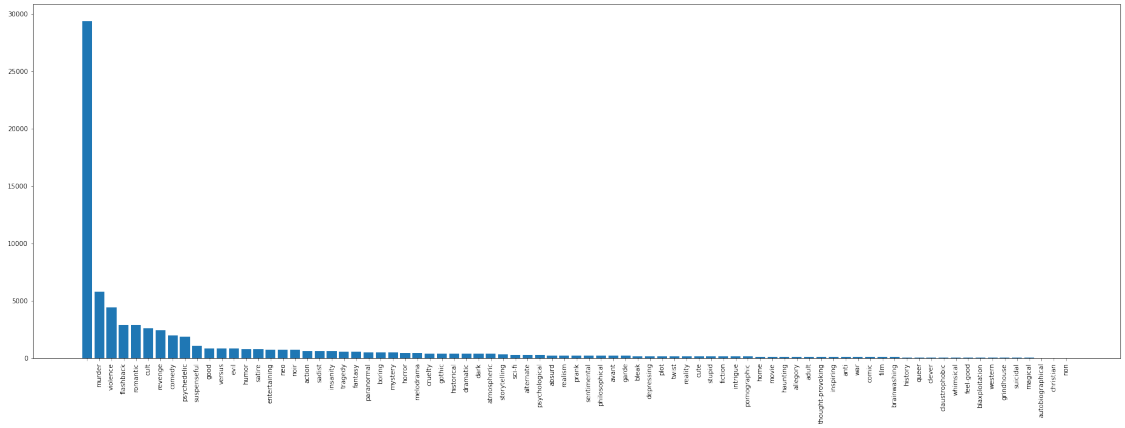
Get tokenized tags

Some of the tags are with '-', ' ' which must be replaced by '_'

Count frequency of each tag

[0]:
```
#Taking frequency of each tag


tag_count = Counter()

for i in tqdm(data.tags):
      for j in i.split():
            # Updating count of each tag in every itteration
            tag_count.update(j.split(','))

# Sorting based on number of times  tag occured
tag_count = dict(sorted(tag_count.items(), key = lambda x:x[1], reverse=True))


# Plotting tag count
tag_count = dict(tag_count)
plt.figure(figsize=(30,10))
plt.bar((tag_count.keys()), tag_count.values())
plt.xticks(rotation = 90)
plt.show()
plt.savefig('tag_count.png')
```

```
100%|      | 14828/14828 [00:00<00:00, 104666.13it/s]
```



```
<Figure size 432x288 with 0 Axes>
```

There is a tag called ”sci-fi”, if we clean the data this tag may be divided into two tags so we shall replace this sci-fi into science_fiction

And tag for first histogram is missing, something went wrong? Lets try to print these tags and their frequencies

(’absurd’, ’absurd’) (’action’, ’action’) (’adult comedy’, ’adult’) (’allegory’, ’allegory’) (’alternate history’, ’alternate’) (’alternate reality’, ’anti’) (’anti war’, ’atmospheric’) (’atmospheric’, ’autobiographical’) (’autobiographical’, ’avant’) (’avant garde’, ’blaxploitation’) (’blaxploitation’, ’bleak’) (’bleak’, ’boring’) (’boring’, ’brainwashing’) (’brainwashing’, ’christian’) (’christian film’, ’claustrophobic’)

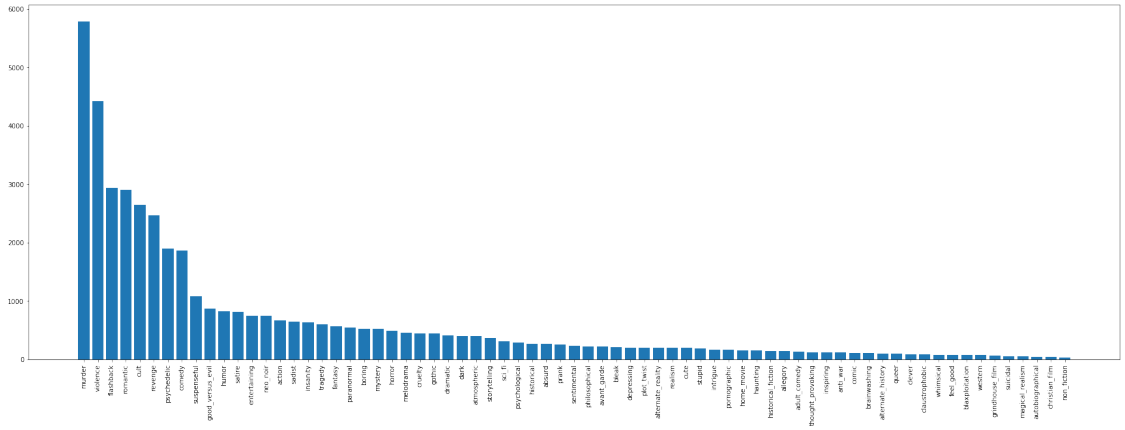
[0]:
```
tag_count
```

So, thats the problem, even space is consedered as a tag, so we must clean the data, we shall use reguler expression library to clean the data

[0]:
```
cl_tag = []
for i in data.tags:
    #Substuting (, ) as (,)
    cl_tg = re.sub('-','_', i)

    cl_tg = re.sub(', ',',',cl_tg)
    cl_tg = re.sub(' ','_',cl_tg)
    cl_tag.append(cl_tg)


tag_count = Counter()
for i in tqdm(cl_tag):
  for j in i.split():
    tag_count.update(j.split(','))
```

```
tag_count = dict(sorted(tag_count.items(), key = lambda x:x[1], reverse=True))

tag_count = dict(tag_count)
plt.figure(figsize=(30,10))
plt.bar((tag_count.keys()), tag_count.values())
plt.xticks(rotation = 90)
plt.show()
plt.savefig('cl_tag_count.png')
```

```
100%|      | 14828/14828 [00:00<00:00, 282988.08it/s]
```




```
<Figure size 432x288 with 0 Axes>
```

[0]:
```
# with open('tag_count.pkl','wb') as f:
#   pickle.dump(tag_count,f)
```

[0]:
```
with open('tag_count.pkl', 'rb') as f:
  tag_count = pickle.load(f)
```

[0]:
```
#Putting this cleaned taggs into data frame
data['clean_tag'] = cl_tag
data.to_csv('data.csv', index=False)
```

4.2.1 Observations:

Most of the tags appear less than 1000 times

Only eight of the tags are appeared more than 1000 times

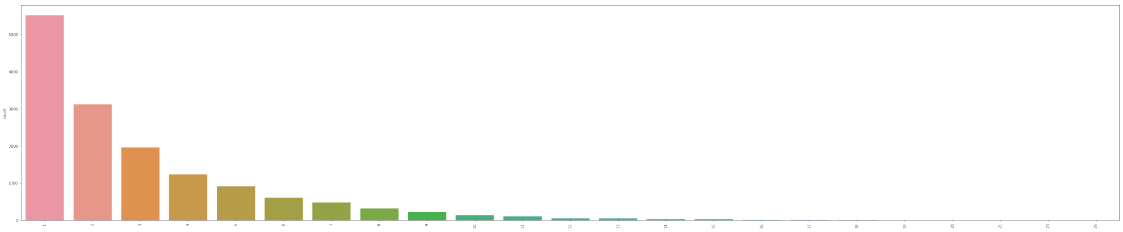
Murder is the tag that appeared max number of times i.e almost 6000 times

[0]:
```
print('Total number of unique tags:', len(tag_count))
```

```
Total number of unique tags: 71
```

4.3 Tags Per Each Movie

[0]:
```
no_tags = []

for i in data.clean_tag:

      no_tags.append(len(i.split(',')))

plt.figure(figsize=(50,10))
sns.countplot(no_tags)
plt.xticks(rotation = 90)
plt.show()
plt.savefig('no_tags.png')
```



```
<Figure size 432x288 with 0 Axes>
```

[0]:
```
# with open('no_tags.pkl','wb') as f:
#   pickle.dump(no_tags,f)
```

[0]:
```
with open('no_tags.pkl','rb') as f:
  no_tags = pickle.load(f)
```

4.4 Observations:

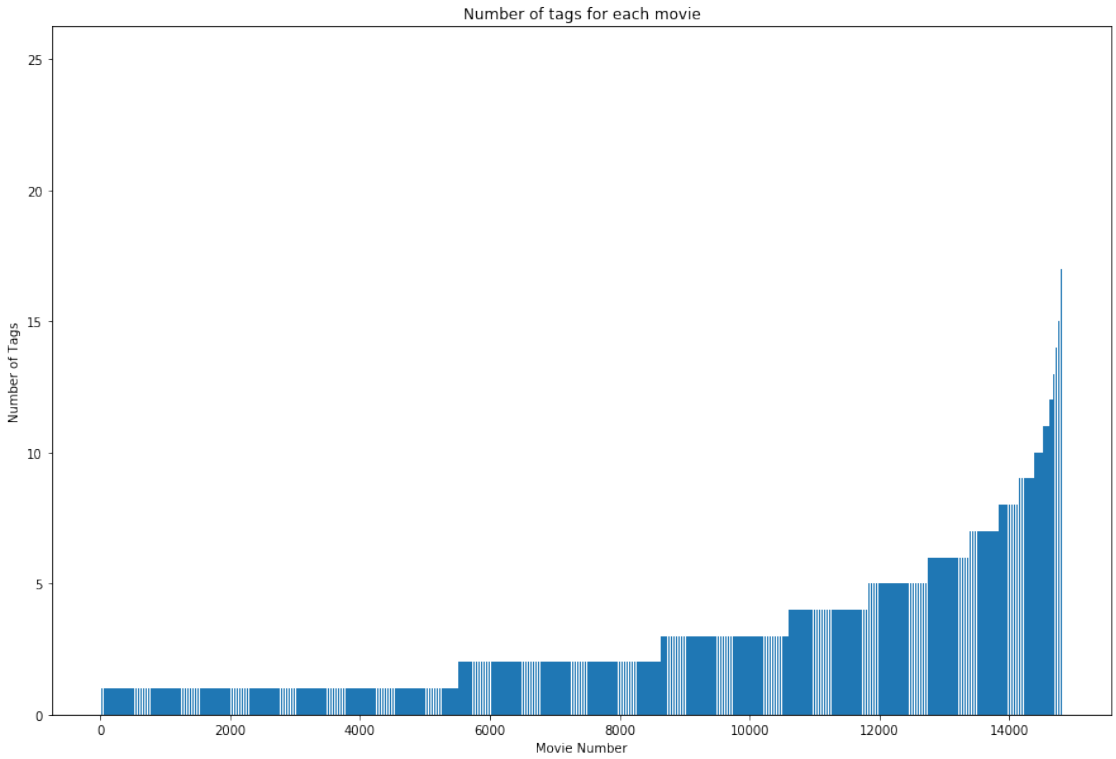
Most of the movies have 1 or 2 number of tags

Almost above 5000 movies have only one tag

Maximum number of tags is 25

Minimum number of tags is 1

[0]:
```
plt.figure(figsize=(15,10))
plt.title('Number of tags for each movie')
plt.xlabel('Movie Number')
plt.ylabel('Number of Tags')
plt.bar(np.array([i for i in range(len(no_tags))]),sorted(no_tags))
plt.show()
```



4.5 Observations:

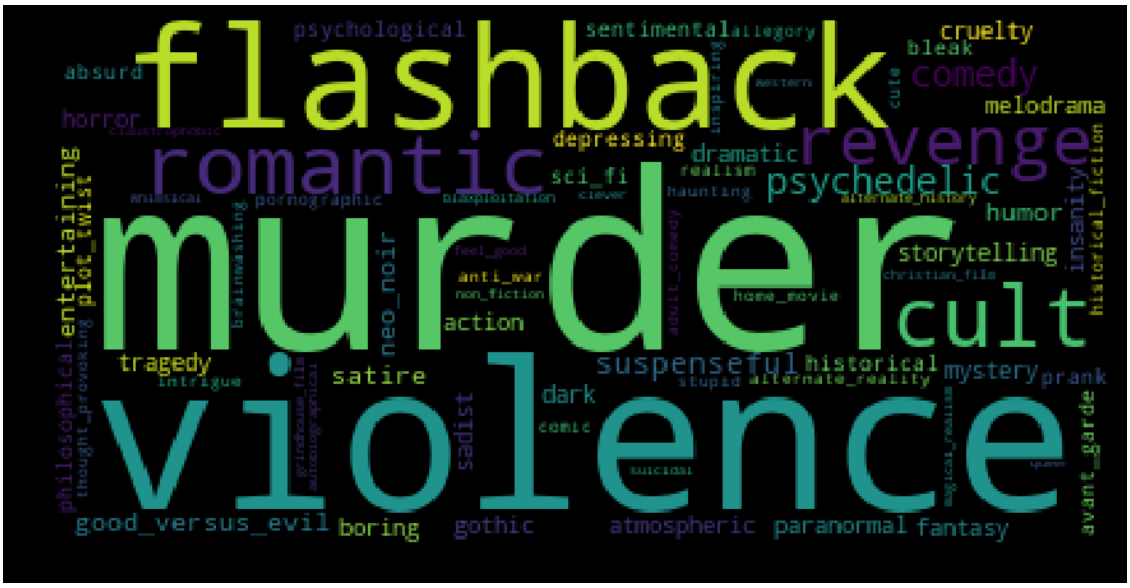
Almost 5500 movies have only one tag

Nearly 3500 movies have two tags

[0]:
```
print('Maximum number of tags is',max(no_tags))
print('Mainimum number of tags is',min(no_tags))
print('Average number of tags is ', sum(no_tags)/len(no_tags))
```

```
Maximum number of tags is 25
Mainimum number of tags is 1
Average number of tags is  2.9812516859994607
```

[0]:
```
plt.figure(figsize=(15,10), facecolor='k')

plt.imshow(WordCloud(contour_color='firebrick').
 ↪generate_from_frequencies(tag_count))
plt.show()
plt.savefig('word_cloud.png')
```




```
<Figure size 432x288 with 0 Axes>
```

4.6 Observations

Tag that is with most frequency is murder and violence

Next important tags are flashback, romantic, revenge, cult and comedy

[0]:
```

```

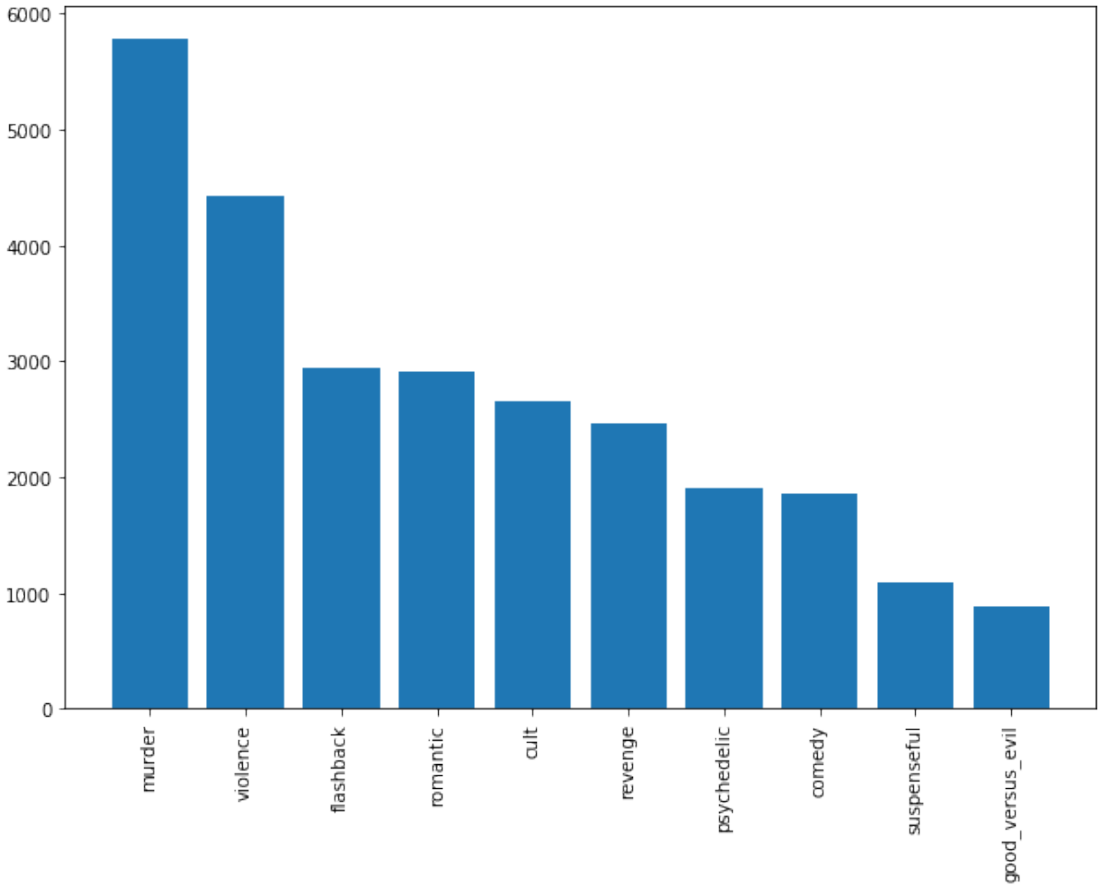
[0]:
```
# sum(no_tags)/len(no_tags)
data['no_tags'] = no_tags
# Saving data frame
data.to_csv('data.csv', index=False)
data.head()
```

[0]:
```
      imdb_id  … no_tags
0  tt0057603  …       5
1  tt1733125  …       1
2  tt0033045  …       1
```

```
3  tt0113862  …       4
4  tt0086250  …      10

[5 rows x 8 columns]
```

4.7 Top 10 Tags

[0]:
```
plt.figure(figsize=(10,7))
plt.bar(list(tag_count.keys())[:10], list(tag_count.values())[:10])
plt.xticks(rotation = 90)
plt.show()
plt.savefig('top_10_tag.png')
```




```
<Figure size 432x288 with 0 Axes>
```

4.8 Cleaning & Preprocessing Data

4.8.1 Finding Out Special Charecters Present In Our Data

[0]:
```
t =[i for i in string.ascii_lowercase]
t.extend([i for i in string.ascii_uppercase])

chars = []
for i in data.plot_synopsis:
  for j in i:
    if j not in t:
        chars.append(j)
        temp = j


special_chars = list(set(chars))
```

[0]:
```
special_chars
```

We found chainese characters in the special charecters, so we shall try to replace those chinese characters with english meaning

[0]:
```
!pip install googletrans
```

[0]:
```
from googletrans import Translator
```

[0]:
```
# https://pypi.org/project/googletrans/

trans = Translator() # Google Translator

special_chars_meaning = {}
for i in tqdm(special_chars, position=0):

    special_chars_meaning[i] = trans.translate(i).text
    # print(trans.translate(i).text)
```

```
100%|      | 605/605 [00:39<00:00, 14.71it/s]
```

[0]:
```
special_chars_meaning
```

[0]:
```
nltk.download('stopwords')

stopwords = stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
```

[0]:
```
','.join(stopwords)
```

[0]:
```
"i,me,my,myself,we,our,ours,ourselves,you,you're,you've,you'll,you'd,your,yours,
yourself,yourselves,he,him,his,himself,she,she's,her,hers,herself,it,it's,its,it
self,they,them,their,theirs,themselves,what,which,who,whom,this,that,that'll,the
se,those,am,is,are,was,were,be,been,being,have,has,had,having,do,does,did,doing,
a,an,the,and,but,if,or,because,as,until,while,of,at,by,for,with,about,against,be
tween,into,through,during,before,after,above,below,to,from,up,down,in,out,on,off
,over,under,again,further,then,once,here,there,when,where,why,how,all,any,both,e
ach,few,more,most,other,some,such,no,nor,not,only,own,same,so,than,too,very,s,t,
can,will,just,don,don't,should,should've,now,d,ll,m,o,re,ve,y,ain,aren,aren't,co
uldn,couldn't,didn,didn't,doesn,doesn't,hadn,hadn't,hasn,hasn't,haven,haven't,is
n,isn't,ma,mightn,mightn't,mustn,mustn't,needn,needn't,shan,shan't,shouldn,shoul
dn't,wasn,wasn't,weren,weren't,won,won't,wouldn,wouldn't"
```

[0]:
```
stemmer = SnowballStemmer('english')
```

[0]:
```
print(data.plot_synopsis.iloc[5])
print('='*50)
print(data.plot_synopsis.iloc[15])
print('='*50)
print(data.plot_synopsis.iloc[70])
print('='*50)
print(data.plot_synopsis.iloc[1620])
```

```
George Falconer (Colin Firth) approaches a car accident in the middle of a snow-
white scenery. There is a bloodied man there and he kisses him. He wakes up: he
was dreaming about the moment when his partner of 16 years, Jim (Mathew Goode),
died--though he was not there with him because Jim was visiting his disapproving
family on his own. George remembers the phone ringing on that fateful day, when
Jim's cousin told him about the fatal accident, and how George was not welcome
to attend the funeral, because of the family's homophobia (common for the period
and later). George remembers breaking down to Charley (Julianne Moore) that day,
his best friend from his life in London, who had also relocated to LA; once
briefly sexually attached to George before he was completely honest with
himself, she may still feel attracted to him.George showers and dresses. It's
November 30, 1962, the eve of the Cuban missile crisis. Though British, he is
now a professor of English at UCLA. He is depressed, never having recovered from
his loss; and when he leaves for work, he packs a gun in his briefcase.He tells
his cleaning lady Alva (Paulette Lamori) that she has always been wonderful - in
spite of her having forgotten to take out the bread from the fridge. George hugs
her, which leaves her utterly confused.On campus, George notices a couple of
students, chain-smoking Lois (Nicole Steinwedell) and a boy. One of the
secretaries (Keri Lynn Pratt) tells him that she has given his address to some
nice new student; it turns out to be this boy, Kenny Potter (Nicholas Hoult),
who talks to him after class about the speech George has just given out in the
classroom concerning minorities and fear. Kenny discusses recreational drug use
```

with Kenny who tells him that he had never heard George express himself so openly in class as he had that day. He buys George a pencil sharpener as a token of gratitude for George's talking with him.George phones Charley, who is dressing for the dinner they have planned at her home. George gets into his car, and picks his gun after having cleaned up his office. However, Kenny appears once again, and invites him to go for a drink, observing George's depression and having noticed that he has cleaned out the desk in his office. George tells him it will have to be some other time. He goes to the bank to pick up various things from his safe deposit box, and when looking at a photo of his deceased lover, recalls a conversation with him on the beach.After buying some bullets, he goes to a convenience store. There, Carlos (Jon Kortajarena) bumps onto him, breaking the bottle of Scotch he has just bought. George buys a new bottle of Scotch and they talk. They smoke a few cigarettes and drink a bottle of gin together. George leaves, refusing Carlos' offer of company, saying that this is a serious day for him and that he's trying to get over an old love.At home, he puts on a record and remembers a conversation with Jim while each one was reading a different book on a couch. He pretends shooting himself as practice for later that night, but in a semi-comic scene, can't find the best position in which to accomplish it. Charley calls to remind him of their dinner plans, which he grudgingly attends after leaving a note and some money for Alva. They dance and talk about London, life, Charley's ex-husband's abandonment, and she offends George by suggesting that they might have had a "normal" life together if he hadn't been a "poof." Charley says George doesn't look well, reminding him of the heart attack he suffered near the time of Jim's death. Charley tries to convince George to spend the night at her home, but he leaves.The scene flashes back to 1946 when Jim and George had met when at a bar. Jim was on leave from the Army, right after the second world war. Returning to1962, we see George returning to the same bar, near his home; now a quiet place where he asks for a Scotch.Kenny has followed him there. They talk and then go to the beach and swim naked. They go to George's place. As George's forehead is bleeding, Kenny tends to it, and sees in the medicine's cabinet a nude photo of Jim. George sees Kenny strip off his wet clothes, but does nothing. Kenny says that he and Lois are not romantically involved. Not unlike George and Charley in the distant past, Kenny explains that they had a brief sexual liason. Kenny and George do not have sex, and Kenny stays on the couch, given the very late hour.George wakes in a few hours, and finds his gun under Kenny's covers and removes it, locking it up as Kenny sleeps. When he returns to bed, George dies of a heart attack, seeing the image of Jim kissing his forehead.

==========

Hours after the end of the previous game and the death of the traitorous General Shepard, the remnants of Task Force 141 (TF141) Captain John Price (Billy Murray), John Soap MacTavish (Kevin McKidd) and Prices pilot/informant Nikolai have fled Afghanistan to a safe house belonging to Nikolai's allies. Soap is critically wounded and the whole team is wanted for war crimes they didn't commit. Shortly after they arrive, forces under the command of Ultranationalist terrorist Vladimir Makarov (Roman Varshavsky) attack to try to kill them. Nikolais best soldier, a man named Yuri (Brian Bloom) , joins the team and helps Price and Nikolai evacuate the team to safetyMeanwhile, in New York City, a

Delta Force unit named Team Metal is activated. The team is led by a squad leader code-named "Sandman" (William Fichtner) and Sgt. Derek "Frost" Westbrook . They are tasked with taking out as Russian jammer tower on the roof of the New York Stock Exchange which is preventing American air attacks. The team fights it's way through the city, into the trading floor and up onto the roof where they destroy the tower. They are evacuated from the roof by a Blackhawk gunship and they fight their way to safety, destroying several Mi-24 Hind gunships along the way. The American Air Force then establishes air superiority over the city. Next, Team Metal rendezvous with a team of Navy SEALS to infiltrate a Russian Oscar II submarine being used as a command vessel. They force it to surface and take control of the boat. Sandman and Frost use the sub's weapons against the Russian fleet in the harbor and then escape the vessel. These twin American victories, together with victories in the Washington DC area (from the previous game) prompt the Russians to evacuate the United StatesTwo months later, Russian President Boris Vorshevsky proposes a ceasefire. He heads to a summit meeting with the American Vice President in Hamburg, Germany. This action enrages the hardliners in his government and they side with Makarov. The President's plane is hijacked on the way to the summit by Makarov's forces and the President is captured. Makarov demands that Vorshevsky give him the nuclear launch codes, which he refuses.Makarovs actions prompt TF141 to take action. Yuri's intelligence reported that Makarov was moving some cargo from Sierra Leone to Europe. Makarov was using a local militia as security and building his weapons in Africa. TF141, consisting of Price, a recovered Soap and Yuri sneak into the village. They investigate the factory, only to find it abandoned. The militia then attacks, both with technical and Mortar teams. TF141 neutralizes the enemies and heads to the militia HQ, a local church. They are able to eliminate all the enemies there but they just miss the shipment. Information at the militia headquarters reveals that Makarov was smuggling WMDs using a company called Fregata EnterprisesWestern intelligence learns that one of Makarov's shipments is headed to London. A SAS team led by Sgt. Wallcroft and Sgt. Marcus Burns tries to intercept the shipment, first at the docks and then in a harrowing chase through the London Underground. They are able to catch up to the truck, only to learn it was a decoy. A mile or so away, an American tourist family are on their way to see Big Ben. As the child runs up to the corner, a white truck come stops at the curb and two men run away from the vehicle. Moments late the chemical weapons bomb in the truck exploded. All caught on the tourist camcorder. This would be just one of a series of chemical attacks in Western capitals that day with the intent of crippling Western response to Russian aggression. When the smoke cleared, it was determined that thousands had died in a terrorist attack even worse than 9-11. Taking advantage of the chaos, the Russian army attacked Western Europe, occupying much of the continentNATO responds by sending a force to liberate Hamburg. Task Force Metal was sent along with them to do another mission--rescue the American Vice President, who was being held by Makarovs forces. Frost and the rest of Team Metal fight alongside the invasion force and push their way into the city. At that point Frost joins a tank crew and takes the vehicle into a parking garage fighting all the way. At the end the team get out onto the streets, searching for the Vice President convoy. They find it a few blocks away, with no one there. They head into

another building where they rescue the Vice President.Price contacts his former commander, MacMillan, now in command of the SAS. MacMillan gives Price intel on the attacks, namely that Makarov's African contact, a Somali warlord named Waraabe was responsible for providing the chemical weapons and he was at his compound in Bosasso, Somalia. Price and his team head into Waraabe's compound just ahead of a huge sandstorm. The team enters the compound engaging the warlord's forces. Nikolai gives Yuri remote command of a mini-gun on the chopper. He uses it to cut down many of the enemy forces. Price Soap and Yuri fight their way into Waraabe's headquarters. They breach the door to his office and take out the guards, capturing Waraabe.Price and the team put on their gas masks and Price threatens Waraabe with the gas. He makes Waraaabe reveal his contact, an arms dealer named Volk, the man behind Fregatta Enterprises. Price then kills Waraabe for his role in killing so many thousands of people. When the team tries to leave, they are attacked by the rest of the militia. With the sandstorm bearing down, they push forward to the landing zone. The sandstorm comes too fast and the chopper goes down. The team fights their way to the crash site to rescue Nikolai. The storm reduces visibility to a few feet. The team makes it to the crash site and they flee to a vehicle to escape the compoundArmed with this information, Price contacts Sandman and tells him about Volk. Team Metal then heads to Paris to link up with French Intelligence. Team Metal and the French forces fought through the streets of Paris, still filled with clouds of toxic gas. They used a couple of airstrikes to take out the vehicles in their way and made it to the catacombs under the city. The joint team slipped through the passageways until they found Volk's headquarters. Volk fled when the team arrived with Frost in hot pursuit. They emerged onto the streets and Volk fled away in his car. The team got into their vehicle and chased Volk. They caught up to Volk and ran his car off the road, capturing him.Because the city was a war zone, Team Metal needed to fight their way to the landing zone so they could escape with their prisoner. The team moved for several blocks only to be blocked by an armor column. They are given air support by an AC-130 gunship which clears their way to the US Embassy. The team is pinned down by an implanted machine gun nest, but Frost moves to the flanks and takes out the enemy with a sniper rifle. The team fights on and then make a mad dash through the city getting support from the gunship until they reached the Eiffel Tower. There, they fought a pitched battle with the enemy. With the help of air support the team was able to clear the way to the landing zone. The Tower fell during the battleVolk revealed the whole conspiracy under questioning. He revealed that Makarov was holding a war council with his allies in a hotel in Prague, Czech Republic. Sandman gave the intel to Price who was already in Prague. Price intended to assassinate Makarov at the meeting. Yuri and Soap slipped into the city. Because the city was under Russian lockdown, they needed extreme stealth. They got into the center of the city, but couldn't get to their target because of the enemy forces. At that point, local Resistance fighters launched a diversionary attack (it was arranged by Price and his ally Kamarov). The team fought their way through the city and got to their target, a church overlooking the hotel where the meeting was scheduled.Hours later, Price was at the hotel and Yuri and Soap were waiting across the street in the church with sniper rifles, waiting for Makarov. The convoy showed up, but Makarov wasnt

there. He anticipated the attack and booby trapped both the church and hotel (he strapped the explosives to the captured Kamarov). Makarov sent a message "You shouldnt have come here, Yuri". Soap realized that Makarov and Yuri knew each other. The bombs went off Price managed to escape, but when the bomb went off in the church, Soap and Yuri fell to the ground. Soap was badly injured.Yuri and Price fought their way to the safehouse in a frantic attempt to get aid for the severely wounded Soap. They survived to make it to safety. They tried to save Soap, but he died. In his dying words, he gasped "Makarov knows Yuri". Price and Yuri left the room and Price punched Yuri down the nearby stairs, jamming a Desert Eagle pistol in Yuris face "Tell me how the bloody hell Makarov knows you"Yuri began his story, telling of how he and Makarov were working for Imran Zakhaev since the 1996 incident in Pripyat--near the site of the Chernobyl power plant (when Price tried to kill Zakhaev, but just wounded him) and the moment when Makarov ordered the detonation of the Nuke that destroyed a city, and killed over 30,000 American soldiers. Yuri also told of how he tried to stop Makarov's terrorist attack on the Moscow airport--he revealed the plot to Russian security services and was shot by Makarov for doing this. During the massacre he followed Makarov, despite his wounds in an attempt to stop him but he couldn't. At that point Price decides to still believe Yuri--for now.Price and Yuri go to the only other place in the area where Makarov's people are believed to be, a medieval castle. They slip into the compound, planting C4 on various parts of the building and infiltrate the command center. They continue on and slip up to the main office of the building. They learn that Makarov is still pressuring President Vorshevsky for the launch codes and that Makarov has learned where Vorshevsky's daughter Alena was being hidden in Berlin. Price and Yuri escape from the castle and send a message to Team Metal about Makarovs plans to capture Alena to force Vorshevsky to give up the codes.Team Metal and other American forces contact Alena Vorshevsky and she reveals where she is hiding and that her protection detail has been killed. Metal fights their way to the roof of a nearby building with the plan of providing cover for Team Granite, who was going to perform the rescue. Frost as his team take control of the roof and Frost used a sniper rifle to clear the adjoining rood. Team Granite arrives on the roof and moves to enter the building, while a column of Russian tanks approach the building Frost calls an A-10 strike on the column to protect the American forces. The tanks are destroyed, but Team Granite is killed by additional enemy forces in the building. Sandman then ordered Metal to go to the hotel, knowing they would have to rescue Alena themselves. They fought their way through the city, supported by a German armor column. They nearly reach the building only to have the Russians destroy another building, dropping the rubble on the team. Team Metal slips through the collapsed building and gets to the hotel where Alena was hiding. They fight through the building, climbing up to the fifth floor, only to learn Alena wasn't there anymore. Frost and Sandman climb up to the roof they rush to save Alena, but they are too late--Makarovs men took her away by chopper and Frost was badly wounded.Knowing that Vorshevsky wouldnt be able to hold out now that Makarov had his daughter, Team Metal (minus the wounded Frost) and TF 141 launch a joint rescue mission. They knew that Makarov was holding his hostages in a diamond mind in Siberia. The two teams fight their way through the building supporting the diamond mine and get to the

outside. They fight their way through the construction area supported by Predator strikes until they reach the command center. They go through the building until they reach the room where Makarov's men were torturing Alena. Team metal rescues a wounded Alena, who tells them Makarov's men took her father deeper into the mine. The two teams press on deeper into the mine, where they find the team guarding President Vorshevsky. Yuri shoots the terrorist holding Vorshevsky with a sniper shot and the teams move him to the rescue chopper. The terrorists counterattack, Price and Yuri get the president into the chopper as Team Metal covers their escape. Price wanted to help them escape also, but Sandman tells him to save the President. TF141 escapes with the Presient while the mine collapses killing Team MetalAfter President Vorshevsky is rescued, the summit goes on as planned and the war comes to an end. In Moscow, Makarov's political allies are purged out of the government and Vorshevsky forms an Ultranationalist-Loyalist coalition government. As a result of their heroics in saving Vorshevsky and the revelation of General Shepard's duplicity, Task Force 141 is cleared of all charges. This clears the task force to do one last mission --Kill Makarov. They discover that Makarov and his remaining forces have set up shop in Dubai. Price and Yuri arrive in Dubai and head to Makarov's HQ--the Hotel Oasis. Equipped with Juggernaut suits they storm into the hotel lobby, taking out Makarov's guards as they go. They head up in the elevator only to have it attacked by a helicopter. Their armor is damaged and they have to shed it. Price and Yuri head up to the restaurant on the top floor. They fight Makarov's guards and Yuri is severely wounded as Makarov flees to the roof. Yuri implores Price to leave him and go after Makarov. Price sprints onto the roof and jumps onto Makarov's chopper just as it was leaving. Price kills the two pilots and takes the chopper. It crashes onto the roof, wounding both Price and Makarov. Makarov moves to kill Price, but Yuri clambers up onto the roof to stop him. Makarov kills Yuri, giving Price the distraction he needed. Price wraps the cable around Makarov's neck and hangs him. Price then has a victory cigar as he watches Makarovs corpse hang in front of him

==========

The film is set in a small Oregon town, where a brutal serial killer nicknamed the 'Oregon undertaker' has been murdering and mutilating young women. Charlie (David Schwimmer) is an ex-teacher turned disaffected call center employee who is fired on his first day. Distraught at being unable to provide for his daughter Emily and policewoman wife Penelope (Natascha McElhone), he is approached by former colleague Gus (Simon Pegg), an aspiring scam artist who presents Charlie with a seemingly snag-free plan to make some cash: blackmail Reverend Smalls, who is listed in the company database of visitors to illegal porn websites. Gus plans on extorting money from Reverend Smalls, with the intention of publicly exposing his secret shame should he refuse to pay. The normally cautious Charlie reluctantly agrees to play a part in the scam, confident that with the money he will garner from the deal he would be able to support his family. A teenage pageant queen Josie McBroom (Alice Eve), Gus's scheming one-night stand, forces herself into the scheme. Josie convinces Charlie and Gus that she should make the call to Smalls on the grounds that if either of them called, the police could trace their voices and connect them to the job.

The plan goes into action, and Gus goes to Smalls' house where he is surprised by the Reverend wielding a gun. From outside the house two shots are heard. At the same time, Charlie arrives at the bar to explain his and Gus' alibi, only to find out that the blind owner of the gas station where Gus has supposedly gone is at the bar celebrating his 80th birthday. Charlie runs away and goes to Smalls' house, where he finds the Reverend dead. Scared, he drags the body outside, and dumps him into a nearby septic tank. He goes back inside the house, tries to call Josie. He then finds Gus and learns that the Reverend shot Gus in the leg, explaining the blood near the corpse. After the first shot, Gus hit the Reverend over the head with a vase, leaving him alive but unconscious. Gus and Charlie leave the house to escape the scene of the crime, but Charlie remembers that he left Josie's card inside. They go back and get it, but in the process Gus stumbles upon a hidden DVD collection of the Reverend. They put one in, and find a video of the Reverend killing and torturing a young girl. They try to flee the house, but are immediately met by a deputy police officer outside the door. The policeman explains that the Reverend was found dead, with three bullet holes in his head.

Charlie lets in the policeman, who notices some blood on the floor. Charlie, in the kitchen, grabs a knife and cuts himself, and then goes back to the policeman to explain that he cut himself on a vase. Right before leaving, the policeman decides to see what Charlie had been watching, much to Charlie's protests. He presses play, and watches the beginning of a children's movie Gus had secretly switched in. Satisfied, he leaves the house, but finds marks in the ground that look like someone had been dragged. Charlie follows the policeman to the end of the drag marks, the septic tank. As Charlie is about to open it, Gus smashes a vase against the head of the deputy, who is promptly dragged inside.

Inside, Charlie panics about the murder of the Reverend and the kidnapping of a policeman and leaves, where he is met by the Reverend's wife (Mimi Rogers) who has a gun pointed at Charlie. They go back inside, where she explains that she shot the Reverend and that she was going to meet her lover, Max, at the house so they could collect their $2,000,000 the real Reverend had left. Gus and Charlie explain that Max will not be seeing her, and she asks them where the money is, pointing a gun at the tied-up policeman. They frantically try to say that they don't know anything about the money, and right before she is about to shoot the deputy, Josie comes in and lodges an axe in the wife's head. As Charlie and Gus talk over what's been happening, Josie finds the money hidden in the Reverend's oven, and calls Charlie and Gus over. As they are looking at the stacks of hundred dollar bills, they hear cries of "help" from outside. The deputy has escaped through the front door, and, rolling himself along, is soon caught by the three. In his rolling, the deputy drops his badge, which Charlie picks up and puts in his pocket. They agree to dispose the body of the wife. The policeman, still alive, asks to use the toilet. Inside, he tries to escape through a window, but slips and kills himself by breaking his skull on the toilet.

Charlie, Gus, and Josie hide the bodies in suitcases and drive away to dispose them, but soon get into an argument. Charlie reveals that the reason why he has been reeling off random facts is because of a neurological disorder. This problem with the neurotransmitter acetylcholine means that eventually his mind

will become blank and explains why he was fired from his teaching position. After driving for a while Charlie realises that Gus, who had originally said he needed the money for his daughter's cornea operation, does not really have a daughter, and punches him. In this small fight, they nearly get into an accident with a fat man, who tries to call the police, but is persuaded by Gus to not do so. Charlie, Gus and Josie drive away and get to their disposal point, only to find that one of the bodies is missing. They drive back and hit the Reverend's wife, who had jumped out of the car and was trying to get help. As they look over the body, two police officers arrive, one of them being Charlie's wife, and quickly see the body. Charlie's wife tries to call her deputy, but it goes to voicemail. Josie hurriedly makes up a story, but the three are taken to the station where a special agent is waiting.

Agent Hymes (Jon Polito), the fat man the three almost got into an accident with, examines the body with Gus and Josie, seeming to understand the earlier events. However, it turns out not to be the case and he lets them go. In the waiting room, Charlie finds his sleeping daughter, who could not be left alone at the house and was brought by his wife, and gives her his coat. Charlie, Gus and Josie drive to a tar pit, where they plan to dispose of the bodies, but they find that the special agent has been following them. He gets angry at Gus for calling him fat before, and Gus swipes and stabs him with an insulin needle in the foot. The agent throws the gun up, which is caught by Charlie, who then points the gun at the agent. The agent then reveals that Josie is the Wyoming Widow; a murderer who befriended men and killed them with whiskey laced with highly concentrated thallium. She disregards it as nonsense, but Charlie and Gus make her empty her pockets, where they find the tell-tale flask of poisoned whiskey. They make her drink some, and she pretends to die, but soon begins laughing at their foolishness as it is not poisoned. They check for the agent, but as he has disappeared they go looking for him. Gus goes back to the car and tries to hide the money, but is caught by the agent, who complains of his lack of payment for what he does. He shoots Gus twice, killing him, and gets the money. In the mean time, Charlie's wife finds the badge of her deputy in her husband's coat, but drives to a bridge and throws it off, removing the evidence. The agent runs to his car, but is surprised by Josie, who was waiting in the back seat. They make him eat a large sugary lollipop, dangerous because of his diabetes, and leave him for dead. Charlie remarks on what monsters they have become, and is then faced by Josie, who has a gun pointed at his head. She explains that she really is the Wyoming Widow, and then gives him the choice of the bullet or the poisoned whiskey (from her second flask). Charlie tells her not to spend all the money in one place, and drinks the whiskey, dying quickly, but not before he happily sees Josie discover that the bag is filled with nothing but his daughter's stuffed animals. At home, Charlie receives a message on his phone from a publishing firm regarding his book and his dream job and an office. Also, his daughter is seen drawing with marker on some of the hundred dollar bills next to several large stacks of money.

Josie tries to hitch a ride away from Oregon, and finally gets one from an old man. The old man goes to the back to "double-check on something", and Josie takes out the poisoned whiskey. The old man covers a bloody leg with a tarpaulin (where it is revealed he is the Oregon undertaker), and goes back into the truck

to drive away with Josie.
==========
We open with a long shot of a rocket awaiting launch. Captain Neal Patterson (Eric Fleming), Lt. Mike Cruze (Dave Willock) and Lt. Larry Turner (Patrick Waltz) wait outside the Colonel's office. When admitted they are introduced to Professor Konrad (Paul Birch) and told their next assignment is to take Konrad to the Space Station. They express disappointment at such an unimportant assignment, but Col. Ramsey (an uncredited Guy Prescott) cryptically tells them, "There are indications of serious trouble up there. Professor Konrad will tell you about it when you're underway."The crew prepares for launch when Konrad arrives. He explains the importance of the Space Station as a launching point for all future space missions. Outside, Larry is saying goodbye to his girlfriend (an uncredited Joi Lansing). With Larry back on board, Prof. Konrad is strapped into his bed for launch. The crew grimaces with the G-forces of acceleration until they reach orbit. They navigate towards the Space Station. Konrad tells Patterson about the true nature of the mission, "They've made some disturbing observations from the Space Station. Apparently we have some deadly neighbors in outer space. The Earth may be in mortal danger." At that a ray of energy from an unknown origin flashes across the screen. Multiple shots get closer to the Space Station until one makes a glancing blow. The next shot is a direct hit and the wheel-shaped craft is destroyed. The beams are approaching the spaceship, Starfire. They try to outrun the energy force and are struck. The force accelerates the craft to an incredible speed. Later they will discover the speedometer broken above the maximum pegged value. Capt. Patterson was injured by a piece of flying debris.Title and credits roll [15 minutes into the movie] over a star field. They approach a planet and crash land in the snow. The crew awakens to take stock of their situation. They deduce they are on the planet Venus and discover the air is breathable. Below the snowline they enter a jungle with strange plants and no sound at all. A strange pulsating sound pierces the silence and Prof. Konrad correctly observes, "My guess is that was an electronic signal of some kind." They build a camp fire and bed down for the night. Larry takes the first watch. Mike took the watch that brought morning, but he dozed off. As a result, he didn't notice until too late the crew were surrounded by the inhabitants of the area. Women in short shirts, clear plastic high-heel shoes, and carrying ray guns take them hostage. They are marched back to the city. Once inside the city they get the impression that strangers, and particularly male strangers are not welcomed. They are brought into the Council Chamber for an audience with the rulers of Venus.A curtain parts and Queen Yllana (Laurie Mitchell) and her council, all wearing masks, take their seats at a table on a raised platform. Yllana introduces herself, "I am Yllana, ruler of the city of Kadir and queen of this world." Next she asks, "Why are you here?" Patterson introduces himself and his colleagues, then explains, "We came from Earth on a peaceful mission, but our ship was thrown off course, and we crashed here." The Queen does not believe the Earthmen. She and her council adjourn to decide their fate. Motiya (Lisa Davis), the blond woman that Larry found so captivating, leaves to tell Talleah (Zsa Zsa Gabor) of the Earthmen. Talleah, the leader of the resistance movement on Venus, decides to visit the Earthmen and decide for herself if they can be trusted to help her cause. The Earthmen

are returned to the Council Chamber to hear the decision. The Queen pronounces that, "You came here to spy on us, to prepare for invasion. What is the plan of attack? You will die!" They are returned to the prison quarters.Konrad voices his concern, "I have a sense of foreboding about them. A feeling of something monstrous, evil." Patterson adds, "I'm beginning to think our being here isn't an accident." He concludes the energy beam originated on Venus. Talleah visits the Earthmen. She tells them their lives are in great danger. She confides that she is one of many who want to overthrow the Queen and makes a deal, "If you help us, we are going to help you." Then she drops the bombshell, "The Earth is going to be destroyed." Talleah tells of an interplanetary war Venus had ten years earlier with Mordo. Most of the cities were destroyed and lost to the jungle. Yllana led the revolt and took over from the men. Many were killed, but the men she needed are kept on a prison colony moon, Tyrus, that orbits Venus.An escort detail comes to take Capt. Patterson to see the Queen. Konrad noticed the Queen had an interest in Patterson and advises he try to charm her. Betraying her true feeling for Patterson, a jealous Talleah says, "I hate her! I hate that Queen!" Yllana joins Patterson in her bed chamber. She tells him, "Bring me a glass of wine, Captain, and one for yourself." She sits, provocatively on the edge of the bed, showing a lot of leg. They drink. The charm appears initially to be working until she warns, "If you do not give us a truthful explanation of your visit here, the Council sentence will be carried out to the letter." She does provide an escape provision for the handsome Captain. She admits to a desire for him. He asks her to take off her mask, but she refuses. By way of warning, she shows him preparations of the Beta Disintegrator that will be used to destroy the Earth. In a moment of weakness she allows Patterson to unmask her. He is shocked by the visage. Severe radiation burns have left the angry Queen with a scarred face of blue and black skin, with open red sores. "For that you will die," she proclaims. Yllana replaces her mask and calls her guards to escort Patterson back to his cell. When alone, she removes her mask, looks at her face in the mirror and sobs uncontrollably. Patterson explains to his colleagues, "It was atomic radiation that disfigured her. I think its affected her mind." The four are taken to see Talleah in her lab. She introduces another of her circle, Kaeel (Barbara Darrow). Talleah tells Patterson they have no time to waste. She has learned that the Queen plans to destroy the Earth in two days. They plan to walk through the jungle to the site of the Beta Disintegrator and destroy it. Before they can leave, the guards search the lab. Since Talleah is still considered loyal she continues to work as the others hide. When the guards leave, Talleah frees her comrades and the Earthmen and they make their way through the city and escape outside.Odeena (Marilyn Buferd), who brought the Earthmen to see Talleah, has been captured and brought before the Queen. She protests her innocence, and is lulled into a sense of safety when the Queen reassures her, "Odeena has always been loyal to her queen. Release her. You may go." As she turns to leave, Yllana pulls a ray gun from a guard and blasts her into smoke. Outside the city, Talleah, Patterson and the others are under radar and searchlight detection. They conclude they are safest in a cave. The guards are searching in the jungle for them. Larry leaves the safety of the group to explore the cave system. Larry is observed by something that watches from a rock ledge above him. A giant red-eyed spider pounces down on him. He screams and

fights it off. The commotion draws the rest to his aid. A blast with a ray gun soon reduces the spider to a smoldering shadow on the ground. They build a fire and wait in the cave. Patterson, Larry and Mike get to know their female companions on a more intimate basis. Konrad remains unattached, so he volunteers to go gather more wood for the fire. Outside he discovers the search party of guards have found their trail and are closing in fast. They devise a plan to have Talleah pretend to capture and march them back to the city.The Queen oversees final preparations on her Beta Disintegrator. She returns to the city to confront the Earthmen, unaware Talleah is no longer loyal. But Talleah reveals herself, "I don't want your reward. One move and I kill you." Patterson orders Yllana to give orders to suspend all work on the Beta Disintegrator. Talleah adds an order of her own, "You will contact the guards on Tyrus and tell them to free all the men." Yllana pretends to comply, goes to her bed and pulls a ray gun out from under a pillow, but is quickly disarmed. Patterson pulls her mask off and gives it to Talleah to masquerade as the Queen to give the orders. They tie up the Queen and hide her behind a screen. Motiya and Kaeel leave to warn their partisans at the weapon site. The masked guards enter the Queen's chamber. Talleah orders them out, but Yllana kicks over the screen revealing the plot. Talleah is disarmed and unmasked. The Queen is restored to power. Yllana delays the execution of the traitors until after she destroys the Earth, but promises, "For her treachery, Talleah will die last, and most horribly of all." Yllana makes one more appeal to Patterson for his love and attention, but her disfigured face even repells her guards.The Earthmen are taken to the jungle site of the weapon. While Yllana gloats, Motiya and Kaeel signal their mission is completed. On a giant screen the Earth is seen suspended in space. The Queen threatens, "It took untold millions of years to create the planet you call Earth. Watch it closely Earthmen. It will be destroyed in a matter of seconds. Watch it!" Yllana presses the button to start the machine. It whirrs up to speed, then makes an unexpected sound of distress. The Queen presses the button again, but all that happens is a screeching sound. She pounds on the button but still nothing happens. The machine explodes and a small fire erupts inside. The Queen enters the main operating room to work it manually. Every lever she throws only makes things worse. More explosions and more fires. A fight between the loyalists and resistance breaks out. A final fireworks display reduces the Queen to a burnt out carcass.Sometime later the Earthmen and citizens of the city assemble in the Council Chambers. A new Queen, Talleah, and her court assume their places on the dais. Talleah announces that the ship has been repaired and the Earthmen can now leave for home. They prepare to depart, but are delayed by a message for the Queen, "The electronic tele-viewer is working. Earth answers us." Colonel Ramsey tells the crew that their orders are, "Not to attempt a return flight in the Starfire. I will not risk his life or the lives of his men in a patched-up ship. Therefore, Captain, you and your crew will remain on the planet Venus until a relief expedition can reach you." We close with Patterson and Talleah embracing, then each principle cast member gets a credit and close-up.

[0]:
```
c = data.shape[0]
def clean_data(string_):

      '''This is a function that removes special characters, apply stemming
            and    convert into lower case and return the cleaned string'''
      global c
      sys.stdout.write('Remaining {}'.format(c))

      test_st = BeautifulSoup(string_).get_text()

      # print('Length of str before', len(test_st))

      nn = []
      # sc_in_str = []
      for j in test_st.split():

            word = re.sub(r"won't", "will not", j)
            word = re.sub(r"n\'t", " not", word)
            word = re.sub(r"\'ve", " have", word)
            word = re.sub(r"can\'t", "can not", word)
            word = re.sub(r"\'re", " are", word)
            word = re.sub(r"\'s", " is", word)
            word = re.sub(r"\'d", " would", word)
            word = re.sub(r"\'ll", " will", word)
            word = re.sub(r"\'t", " not", word)

            word = re.sub(r"\'m", " am", word)

            for i in special_chars:
                if i in word:
                    try:
                        temp =  word
                        word =  word.replace(i,special_chars_meaning[i])
                        word =  word.lower()
                        # sc_in_str.append(i)

                        # print(temp,j)
                        if temp == word:
                          word = word.replace(i,'')
                    except:
                      pass

            if word not in stopwords:
                  nn.append(stemmer.stem(word))

      c -=1
```

```
        sys.stdout.write('\r')

        return ' '.join(nn)
```

4.8.2 Cleaning Plot Synopsis (Replacing Chinese Characters with their english meaning)

[0]:
```
clean_plot_synopsis = data.plot_synopsis.apply(clean_data)
```

[0]:
```
print(clean_plot_synopsis.iloc[5])
print('='*50)
print(clean_plot_synopsis.iloc[15])
print('='*50)
print(clean_plot_synopsis.iloc[70])
print('='*50)
print(clean_plot_synopsis.iloc[1620])
```

```
georg falcon (colin firth) approach car accid middl snowwhit sceneri there
bloodi man kiss he wake dream moment partner  year jim (mathew goode) diedthough
jim visit disapprov famili georg rememb phone ring fate day jimi cousin told
fatal accid georg welcom attend funer familyi homophobia common period later
georg rememb break charley (juliann moore) day best friend life london, also
reloc la; briefli sexual attach georg complet honest may still feel attract
him.georg shower dress iti novemb   eve cuban missil crisi though british,
professor english ucla. he depress never recov loss leav work pack gun
briefcase.h tell clean ladi alva (paulett lamori) alway wonder  spite forgotten
take bread fridg georg hug leav utter confused.on campus georg notic coupl
student chainsmok loi (nicol steinwedell) boy one secretari (keri lynn pratt)
tell given address nice new student turn boy kenni potter (nichola hoult) talk
class speech georg given classroom concern minor fear kenni discuss recreat drug
use kenni tell never heard georg express open class day he buy georg pencil
sharpen token gratitud georgei talk him.georg phone charley, dress dinner plan
home georg get car pick gun clean offic however, kenni appear invit go drink
observ georgei depress notic clean desk offic georg tell time he goe bank pick
various thing safe deposit box look photo deceas lover recal convers beach.aft
buy bullet goe conveni store there, carlo (jon kortajarena) bump onto break
bottl scotch bought georg buy new bottl scotch talk they smoke cigarett drink
bottl gin togeth georg leav refus carlo offer compani say serious day hei tri
get old love.at home put record rememb convers jim one read differ book couch he
pretend shoot practic later night semicom scene canot find best posit accomplish
charley call remind dinner plan grudg attend leav note money alva. they danc
talk london, life charleyi exhusbandi abandon offend georg suggest might normal
life togeth hadnot poof charley say georg doesnot look well remind heart attack
```

suffer near time jimi death charley tri convinc georg spend night home leaves.th scene flash back jim georg met bar jim leav army, right second world war return see georg return bar near home quiet place ask scotch.kenni follow they talk go beach swim nake they go georgei place as georgei forehead bleed kenni tend see medicinei cabinet nude photo jim. georg see kenni strip wet cloth noth kenni say loi romant involv not unlik georg charley distant past kenni explain brief sexual liason kenni georg sex kenni stay couch given late hour.georg wake hour find gun kennyi cover remov lock kenni sleep when return bed georg die heart attack see imag jim kiss forehead

==========

hour end previous game death traitor general shepard, remnant task forc tf) captain john price (billi murray) john soap mactavish (kevin mckidd) price pilotinform nikolai fled afghanistan safe hous belong nikolaii alli soap critic wound whole team want war crime didnot commit short arriv forc command ultranationalist terrorist vladimir makarov (roman varshavsky) attack tri kill nikolai best soldier man name yuri (brian bloom) join team help price nikolai evacu team safetymeanwhile, new york city, delta forc unit name team metal activ the team led squad leader codenam "sandman" (william fichtner) sgt. derek "frost" westbrook they task take russian jammer tower roof new york stock exchang prevent american air attack the team fight iti way citi trade floor onto roof destroy tower they evacu roof blackhawk gunship fight way safeti destroy sever mi- hind gunship along way the american air forc establish air superior citi next, team metal rendezv team navi seal infiltr russian oscar ii submarin use command vessel they forc surfac take control boat sandman frost use subi weapon russian fleet harbor escap vessel these twin american victori togeth victori washington dc area previous game prompt russian evacu unit statestwo month later russian presid bori vorshevski propos ceasefir he head summit meet american vice presid hamburg, germany. this action enrag hardlin govern side makarov. the presidenti plane hijack way summit makarovi forc presid captur makarov demand vorshevski give nuclear launch code refuses.makarov action prompt tf11 take action yurii intellig report makarov move cargo sierra leon europe. makarov use local militia secur build weapon africa. tf11 consist price, recov soap yuri sneak villag they investig factori find abandon the militia attack technic mortar team tf11 neutral enemi head militia hq, local church they abl elimin enemi miss shipment inform militia headquart reveal makarov smuggl wmds use compani call fregata enterpriseswestern intellig learn one makarovi shipment head london. a sas team led sgt. wallcroft sgt. marcus burn tri intercept shipment first dock harrow chase london underground. they abl catch truck learn decoy a mile away american tourist famili way see big ben. as child run corner white truck come stop curb two men run away vehicl moment late chemic weapon bomb truck explod all caught tourist camcord this would one seri chemic attack western capit day intent crippl western respons russian aggress when smoke clear determin thousand die terrorist attack even wors take advantag chao russian armi attack western europe, occupi much continentnato respond send forc liber hamburg. task forc metal sent along anoth missionrescu american vice president, held makarov forc frost rest team metal fight alongsid invas forc push way citi at point frost join tank crew take vehicl park garag fight way at end team get onto street search vice presid convoy they find block away one they head anoth

build rescu vice president.pric contact former command macmillan, command sas. macmillan give price intel attack name makarovi african contact somali warlord name waraab respons provid chemic weapon compound bosasso, somalia. price team head waraabei compound ahead huge sandstorm the team enter compound engag warlordi forc nikolai give yuri remot command minigun chopper he use cut mani enemi forc price soap yuri fight way waraabei headquart they breach door offic take guard captur waraabe.pric team put gas mask price threaten waraab gas he make waraaab reveal contact arm dealer name volk, man behind fregatta enterprises. price kill waraab role kill mani thousand peopl when team tri leav attack rest militia with sandstorm bear push forward land zone the sandstorm come fast chopper goe the team fight way crash site rescu nikolai. the storm reduc visibl feet the team make crash site flee vehicl escap compoundarm inform price contact sandman tell volk. team metal head pari link french intelligence. team metal french forc fought street paris, still fill cloud toxic gas they use coupl airstrik take vehicl way made catacomb citi the joint team slip passageway found volki headquart volk fled team arriv frost hot pursuit they emerg onto street volk fled away car the team got vehicl chase volk. they caught volk ran car road captur him.becaus citi war zone team metal need fight way land zone could escap prison the team move sever block block armor column they given air support ac1 gunship clear way us embassy. the team pin implant machin gun nest frost move flank take enemi sniper rifl the team fight make mad dash citi get support gunship reach eiffel tower. there, fought pitch battl enemi with help air support team abl clear way land zone the tower fell battlevolk reveal whole conspiraci question he reveal makarov hold war council alli hotel prague, czech republic. sandman gave intel price alreadi prague. price intend assassin makarov meet yuri soap slip citi becaus citi russian lockdown need extrem stealth they got center citi couldnot get target enemi forc at point local resist fighter launch diversionari attack arrang price alli kamarov) the team fought way citi got target church overlook hotel meet scheduled.hour later price hotel yuri soap wait across street church sniper rifl wait makarov. the convoy show makarov wasnt he anticip attack boobi trap church hotel strap explos captur kamarov) makarov sent messag "you shouldnt come yuri" soap realiz makarov yuri knew the bomb went price manag escap bomb went church soap yuri fell ground soap bad injured.yuri price fought way safehous frantic attempt get aid sever wound soap. they surviv make safeti they tri save soap, die in die word gasp "makarov know yuri" price yuri left room price punch yuri nearbi stair jam desert eagl pistol yuri face "tell bloodi hell makarov know you"yuri began stori tell makarov work imran zakhaev sinc incid pripyat--near site chernobyl power plant price tri kill zakhaev, wound moment makarov order deton nuke destroy citi kill american soldier yuri also told tri stop makarovi terrorist attack moscow airporth reveal plot russian secur servic shot makarov dure massacr follow makarov, despit wound attempt stop couldnot at point price decid still believ yuri--for now.pric yuri go place area makarovi peopl believ mediev castl they slip compound plant c4 various part build infiltr command center they continu slip main offic build they learn makarov still pressur presid vorshevski launch code makarov learn vorshevskyi daughter alena hidden berlin. price yuri escap castl send messag team metal makarov plan captur alena forc vorshevski give codes.team metal american forc contact alena vorshevski reveal hide protect detail kill metal

fight way roof nearbi build plan provid cover team granite, go perform rescu frost team take control roof frost use sniper rifl clear adjoin rood team granit arriv roof move enter build column russian tank approach build frost call a1 strike column protect american forc the tank destroy team granit kill addit enemi forc build sandman order metal go hotel know would rescu alena they fought way citi support german armor column they near reach build russian destroy anoth build drop rubbl team team metal slip collaps build get hotel alena hide they fight build climb fifth floor learn alena wasnot anymor frost sandman climb roof rush save alena, late--makarov men took away chopper frost bad wounded.know vorshevski wouldnt abl hold makarov daughter team metal minus wound frost) tf launch joint rescu mission they knew makarov hold hostag diamond mind siberia. the two team fight way build support diamond mine get outsid they fight way construct area support predat strike reach command center they go build reach room makarovi men tortur alena. team metal rescu wound alena, tell makarovi men took father deeper mine the two team press deeper mine find team guard presid vorshevsky. yuri shoot terrorist hold vorshevski sniper shot team move rescu chopper the terrorist counterattack price yuri get presid chopper team metal cover escap price want help escap also sandman tell save president. tf11 escap presient mine collaps kill team metalaft presid vorshevski rescu summit goe plan war come end in moscow, makarovi polit alli purg govern vorshevski form ultranationalist-loyalist coalit govern as result heroic save vorshevski revel general shepardi duplic task forc clear charg this clear task forc one last mission--kil makarov. they discov makarov remain forc set shop dubai. price yuri arriv dubai head makarovi hq--the hotel oasis. equip juggernaut suit storm hotel lobbi take makarovi guard go they head elev attack helicopt their armor damag shed price yuri head restaur top floor they fight makarovi guard yuri sever wound makarov flee roof yuri implor price leav go makarov. price sprint onto roof jump onto makarovi chopper leav price kill two pilot take chopper it crash onto roof wound price makarov. makarov move kill price, yuri clamber onto roof stop makarov kill yuri, give price distract need price wrap cabl around makarovi neck hang price victori cigar watch makarov corps hang front

==========

the film set small oregon town brutal serial killer nicknam oregon undertak murder mutil young women charli (david schwimmer) exteach turn disaffect call center employe fire first day distraught unabl provid daughter emili policewoman wife penelop (natascha mcelhone) approach former colleagu gus (simon pegg) aspir scam artist present charli seem snagfre plan make cash blackmail reverend smalls, list compani databas visitor illeg porn websit gus plan extort money reverend smalls, intent public expos secret shame refus pay the normal cautious charli reluct agre play part scam confid money garner deal would abl support famili a teenag pageant queen josi mcbroom (alic eve) gusi scheme onenight stand forc scheme josi convinc charli gus make call small ground either call polic could trace voic connect job the plan goe action gus goe small hous surpris reverend wield gun from outsid hous two shot heard at time charli arriv bar explain gus alibi find blind owner gas station gus suppos gone bar celebr th birthday charli run away goe small hous find reverend dead scared, drag bodi outsid dump nearbi septic tank he goe back insid hous tri call josie. he find gus learn reverend shot gus leg explain blood near corps after first shot gus

hit reverend head vase leav aliv unconsci gus charli leav hous escap scene crime charli rememb left josiei card insid they go back get process gus stumbl upon hidden dvd collect reverend. they put one find video reverend kill tortur young girl they tri flee hous immedi met deputi polic offic outsid door the policeman explain reverend found dead three bullet hole head charli let policeman notic blood floor charlie, kitchen grab knife cut goe back policeman explain cut vase right leav policeman decid see charli watch much charliei protest he press play watch begin childreni movi gus secret switch satisfied, leav hous find mark ground look like someon drag charli follow policeman end drag mark septic tank as charli open gus smash vase head deputi prompt drag insid inside, charli panic murder reverend kidnap policeman leav met reverendi wife (mimi rogers) gun point charlie. they go back insid explain shot reverend go meet lover max, hous could collect real reverend left gus charli explain max see ask money point gun tiedup policeman they frantic tri say donot know anyth money right shoot deputi josi come lodg axe wifei head as charli gus talk whati happen josi find money hidden reverendi oven call charli gus as look stack hundr dollar bill hear cri help outsid the deputi escap front door roll along soon caught three in roll deputi drop badg charli pick put pocket they agre dispos bodi wife the policeman still aliv ask use toilet inside, tri escap window slip kill break skull toilet charlie, gus, josi hide bodi suitcas drive away dispos soon get argument charli reveal reason reel random fact neurolog disord this problem neurotransmitt acetylcholin mean eventu mind becom blank explain fire teach posit after drive charli realis gus, origin said need money daughteri cornea oper realli daughter punch in small fight near get accid fat man tri call polic persuad gus charlie, gus josi drive away get dispos point find one bodi miss they drive back hit reverendi wife jump car tri get help as look bodi two polic offic arriv one charliei wife quick see bodi charliei wife tri call deputi goe voicemail josi hurri make stori three taken station special agent wait agent hyme (jon polito) fat man three almost got accid examin bodi gus josie, seem understand earlier event however, turn case let go in wait room charli find sleep daughter could left alon hous brought wife give coat charlie, gus josi drive tar pit plan dispos bodi find special agent follow he get angri gus call fat gus swipe stab insulin needl foot the agent throw gun caught charlie, point gun agent the agent reveal josi wyom widow; murder befriend men kill whiskey lace high concentr thallium she disregard nonsens charli gus make empti pocket find telltal flask poison whiskey they make drink pretend die soon begin laugh foolish poison they check agent disappear go look gus goe back car tri hide money caught agent complain lack payment he shoot gus twice kill get money in mean time charliei wife find badg deputi husbandi coat drive bridg throw remov evid the agent run car surpris josie, wait back seat they make eat larg sugari lollipop danger diabet leav dead charli remark monster becom face josie, gun point head she explain realli wyom widow, give choic bullet poison whiskey second flask charli tell spend money one place drink whiskey die quick happili see josi discov bag fill noth daughteri stuf anim at home charli receiv messag phone publish firm regard book dream job offic also, daughter seen draw marker hundr dollar bill next sever larg stack money josi tri hitch ride away oregon, final get one old man the old man goe back doublecheck someth josi take poison whiskey the old man cover bloodi leg tarpaulin reveal oregon undertak goe back truck drive away

josie.
==========
we open long shot rocket await launch captain neal patterson (eric fleming) lt. mike cruze (dave willock) lt. larri turner (patrick waltz) wait outsid colon offic when admit introduc professor konrad (paul birch) told next assign take konrad space station. they express disappoint unimport assign col. ramsey uncredit guy prescott) cryptic tell "there indic serious troubl professor konrad tell youar underway"th crew prepar launch konrad arriv he explain import space station launch point futur space mission outside, larri say goodby girlfriend uncredit joi lansing) with larri back board prof. konrad strap bed launch the crew grimac g-forc acceler reach orbit they navig toward space station. konrad tell patterson true natur mission "theyhav made disturb observ space station. appar dead neighbor outer space the earth may mortal danger at ray energi unknown origin flash across screen multipl shot get closer space station one make glanc blow the next shot direct hit wheelshap craft destroy the beam approach spaceship starfire. they tri outrun energi forc struck the forc acceler craft incred speed later discov speedomet broken maximum peg valu capt. patterson injur piec fli debris.titl credit roll minut movi star field they approach planet crash land snow the crew awaken take stock situat they deduc planet venus discov air breathabl below snowlin enter jungl strang plant sound a strang pulsat sound pierc silenc prof. konrad correct observ "mi guess electron signal kind they build camp fire bed night larri take first watch mike took watch brought morn doze as result didnot notic late crew surround inhabit area women short shirt clear plastic highheel shoe carri ray gun take hostag they march back citi onc insid citi get impress stranger particular male stranger welcom they brought council chamber audienc ruler venus.a curtain part queen yllana (lauri mitchell) council wear mask take seat tabl rais platform yllana introduc "i yllana, ruler citi kadir queen world next ask "whi patterson introduc colleagu explain "we came earth peac mission ship thrown cours crash the queen believ earthmen. she council adjourn decid fate motiya (lisa davis) blond woman larri found captiv leav tell talleah (zsa zsa gabor) earthmen. talleah, leader resist movement venus, decid visit earthmen decid trust help caus the earthmen return council chamber hear decis the queen pronounc "you came spi us prepar invas what plan attack you die they return prison quarters.konrad voic concern "i sens forebod a feel someth monstrous evil patterson add "iam begin think isnot accid he conclud energi beam origin venus. talleah visit earthmen. she tell live great danger she confid one mani want overthrow queen make deal "if help us go help then drop bombshel "the earth go destroy talleah tell interplanetari war venus ten year earlier mordo. most citi destroy lost jungl yllana led revolt took men mani kill men need kept prison coloni moon tyrus, orbit venus.an escort detail come take capt. patterson see queen. konrad notic queen interest patterson advis tri charm betray true feel patterson, jealous talleah say "i hate i hate queen" yllana join patterson bed chamber she tell "bring glass wine captain, one she sit provoc edg bed show lot leg they drink the charm appear initi work warn "if give us truth explan visit council sentenc carri letter she provid escap provis handsom captain. she admit desir he ask take mask refus by way warn show prepar beta disintegr use destroy earth. in moment weak allow patterson unmask he shock visag sever radiat burn left angri

queen scar face blue black skin open red sore "for die proclaim yllana replac mask call guard escort patterson back cell when alon remov mask look face mirror sob uncontrol patterson explain colleagu "it atom radiat disfigur i think affect mind the four taken see talleah lab she introduc anoth circl kaeel (barbara darrow) talleah tell patterson time wast she learn queen plan destroy earth two day they plan walk jungl site beta disintegr destroy befor leav guard search lab sinc talleah still consid loyal continu work other hide when guard leav talleah free comrad earthmen make way citi escap outside.odeena (marilyn buferd) brought earthmen see talleah, captur brought queen. she protest innoc lull sens safeti queen reassur "odeena alway loyal queen releas you may go as turn leav yllana pull ray gun guard blast smoke outsid citi talleah, patterson other radar searchlight detect they conclud safest cave the guard search jungl larri leav safeti group explor cave system larri observ someth watch rock ledg a giant redey spider pounc he scream fight the commot draw rest aid a blast ray gun soon reduc spider smolder shadow ground they build fire wait cave patterson, larri mike get know femal companion intim basi konrad remain unattach volunt go gather wood fire outsid discov search parti guard found trail close fast they devis plan talleah pretend captur march back city.th queen overse final prepar beta disintegrator. she return citi confront earthmen, unawar talleah longer loyal but talleah reveal "i donot want reward one move i kill patterson order yllana give order suspend work beta disintegrator. talleah add order "you contact guard tyrus tell free men yllana pretend compli goe bed pull ray gun pillow quick disarm patterson pull mask give talleah masquerad queen give order they tie queen hide behind screen motiya kaeel leav warn partisan weapon site the mask guard enter queeni chamber talleah order yllana kick screen reveal plot talleah disarm unmask the queen restor power yllana delay execut traitor destroy earth, promis "for treacheri talleah die last horribl yllana make one appeal patterson love attent disfigur face even repel guards.th earthmen taken jungl site weapon while yllana gloat motiya kaeel signal mission complet on giant screen earth seen suspend space the queen threaten "it took untold million year creat planet call earth. watch close earthmen. it destroy matter second watch yllana press button start machin it whirr speed make unexpect sound distress the queen press button happen screech sound she pound button still noth happen the machin explod small fire erupt insid the queen enter main oper room work manual everi lever throw make thing wors more explos fire a fight loyalist resist break a final firework display reduc queen burnt carcass.sometim later earthmen citizen citi assembl council chambers. a new queen, talleah, court assum place dai talleah announc ship repair earthmen leav home they prepar depart delay messag queen, "the electron teleview work earth answer us colonel ramsey tell crew order "not attempt return flight starfire. i risk life live men patchedup ship therefore, captain, crew remain planet venus relief expedit reach we close patterson talleah embrac principl cast member get credit closeup

Once again checking for special characters since only chinese characters were replaced previously

[0]:
```
c = data.shape[0]
def normal_clean_data(string_):

```

```
    '''This is a function that removes special characters, apply stemming
          and    convert into lower case and return the cleaned string'''
    global c
    sys.stdout.write('Remaining {}'.format(c))

    test_st = BeautifulSoup(string_).get_text()

    # print('Length of str before', len(test_st))

    nn = []
    # sc_in_str = []
    for j in test_st.split():

          for i in special_char:
              if i in j:
                      j =  j.replace(i,'')
                      j =  j.lower()
                      # sc_in_str.append(i)



          nn.append(j)

    c -=1

    sys.stdout.write('\r')

    return ' '.join(nn)
```

[0]:
```
chars = []
for i in clean_plot_synopsis:
  for j in i:
    if j not in t:
        chars.append(j)
        temp = j


special_char = list(set(chars))
special_char
```

Cleaning Special Characters

[0]:
```
clean_plot_synopsis = clean_plot_synopsis.apply(normal_clean_data)
```

[0]:
```
chars = []
for i in clean_plot_synopsis:
  for j in i:
    if j not in t:
        chars.append(j)
        temp = j


special_char = list(set(chars))
special_char
```

[0]:
```
[' ']
```

As we observe above there are no any special characters in the data exept sapce

[0]:
```
data['clean_plot_synopsis'] = clean_plot_synopsis
data.to_csv('data.csv', index=False)
data.head()
```

[0]:
```
     imdb_id  …                                clean_plot_synopsis
0  tt0057603  …  note synopsi orgin italian releas segment cert…
1  tt1733125  …  two thousand year ago nhagruul foul sorcer rev…
2  tt0033045  …  matuschekis gift store budapest workplac alfr …
3  tt0113862  …  glenn holland morn person anyonei standard wok…
4  tt0086250  …  in may cuban man name toni montana al pacino c…

[5 rows x 9 columns]
```

4.8.3 Cleaning Titles

[0]:
```
clean_title = data.title.apply(clean_data)
clean_title = clean_title.apply(normal_clean_data)
```

[0]:
```
data['clean_title'] = clean_title
data.to_csv('data.csv', index=False)
data.head()
```

[0]:
```
     imdb_id  …                         clean_title
0  tt0057603  …             i tre volti della paura
1  tt1733125  …  dungeon dragons: the book vile dark
2  tt0033045  …               the shop around corner
3  tt0113862  …                    mr. hollandi opus
4  tt0086250  …                              scarfac
```

```
[5 rows x 10 columns]
```

4.9 Splitting Data (Based on split column in dataset)

[0]:
```
data = pd.read_csv('data.csv')
```

[0]:
```
data.head()
```

[0]:
```
     imdb_id  …                          clean_title
0  tt0057603  …              i tre volti della paura
1  tt1733125  …  dungeon dragons: the book vile dark
2  tt0033045  …               the shop around corner
3  tt0113862  …                    mr. hollandi opus
4  tt0086250  …                              scarfac

[5 rows x 10 columns]
```

[0]:
```
d1,d2,d3 = data.groupby('split')
```

Saving those files into csv format

[0]:
```
d1[1].to_csv('' + d1[0]+'.csv', index = False)
d2[1].to_csv('' + d2[0]+'.csv', index = False)
d3[1].to_csv('' + d3[0]+'.csv', index = False)
```

[0]:
```
train = pd.read_csv('train.csv')
test  = pd.read_csv('test.csv')
val   = pd.read_csv('val.csv')
```

[0]:
```
print('Total number of data points in train data:-  {} i.e {}%'.format(train.
 ↪shape[0],int(train.shape[0]/data.shape[0]*100)))
print('Total number of data points in test  data:-  {} i.e {}%'.format(test.
 ↪shape[0],int(test.shape[0]/data.shape[0]*100)))
print('Total number of data points in val   data:-  {} i.e {}%'.format(val.
 ↪shape[0],int(val.shape[0]/data.shape[0]*100)))
```

```
Total number of data points in train data:-  9489 i.e 63%
Total number of data points in test  data:-  2966 i.e 20%
Total number of data points in val   data:-  2373 i.e 16%
```

4.10 Machine Learning Models

4.11 Featurizing Data

4.12 Featurizing Multilabels

[0]:
```
multiclass = CountVectorizer(tokenizer= lambda x:x.split(','),␣
 ↪vocabulary=list(tag_count.keys()))
ytrain = multiclass.fit_transform(train.clean_tag)
ytest  = multiclass.transform(test.clean_tag)
yval   = multiclass.transform(val.clean_tag)
```

[0]:
```
print('Shape of train data:-',ytrain.shape)
print('Shape of test data :-',ytest.shape)
print('Shape of val  data :-',yval.shape)
```

```
Shape of train data:- (9489, 71)
Shape of test data :- (2966, 71)
Shape of val  data :- (2373, 71)
```

https://arxiv.org/pdf/1802.07858.pdf

It is strongly said in above paper that if n in ngram is 1,2,3 it extracts word and if n in ngram is 3,4 it extracts character

4.13 Vectorizing Plot Synopsis

4.13.1 Applying TFIDF Vectorizer on Plot_synopsis with uni grams

[0]:
```
tfidf_vect_uni  = TfidfVectorizer(ngram_range=(1,1), min_df=10,␣
 ↪max_features=20000)
tfidf_train_uni = tfidf_vect_uni.fit_transform(train.clean_plot_synopsis)
tfidf_test_uni  = tfidf_vect_uni.transform(test.clean_plot_synopsis)
tfidf_val_uni    = tfidf_vect_uni.transform(val.clean_plot_synopsis)
```

[0]:
```
print('Shape of Train TFIDF', tfidf_train_uni.shape)
print('Shape of Test TFIDF ', tfidf_test_uni.shape)
print('Shape of val TFIDF  ', tfidf_val_uni.shape)
```

```
Shape of Train TFIDF (9489, 15606)
Shape of Test TFIDF  (2966, 15606)
Shape of val TFIDF   (2373, 15606)
```

4.13.2 Applying TFIDF Vectorizer on Plot_synopsis with bi grams

[0]:
```
tfidf_vect_bi  = TfidfVectorizer(ngram_range=(2,2), min_df=10,␣
 ↪max_features=20000)
tfidf_train_bi = tfidf_vect_bi.fit_transform(train.clean_plot_synopsis)
tfidf_test_bi  = tfidf_vect_bi.transform(test.clean_plot_synopsis)
tfidf_val_bi    = tfidf_vect_bi.transform(val.clean_plot_synopsis)
```

[0]:
```
print('Shape of train tfidf bi gram', tfidf_train_bi.shape)
print('Shape of test tfidf bi gram ', tfidf_test_bi.shape)
print('Shape of val  tfidf bi gram', tfidf_val_bi.shape)
```

```
Shape of train tfidf bi gram (9489, 20000)
Shape of test tfidf bi gram  (2966, 20000)
Shape of val  tfidf bi gram (2373, 20000)
```

[0]:
```
```

4.13.3 Applying TFIDF Vectorizer on Plot_synopsis with tri grams

[0]:
```
tfidf_vect_tri  = TfidfVectorizer(ngram_range=(3,3), min_df=10,␣
 ↪max_features=20000)
tfidf_train_tri = tfidf_vect_tri.fit_transform(train.clean_plot_synopsis)
tfidf_test_tri  = tfidf_vect_tri.transform(test.clean_plot_synopsis)
tfidf_val_tri    = tfidf_vect_tri.transform(val.clean_plot_synopsis)
```

[0]:
```
print('Shape of train tfidf tri gram', tfidf_train_tri.shape)
print('Shape of test tfidf tri gram ', tfidf_test_tri.shape)
print('Shape of val  tfidf tri gram', tfidf_val_tri.shape)
```

```
Shape of train tfidf tri gram (9489, 1594)
Shape of test tfidf tri gram  (2966, 1594)
Shape of val  tfidf tri gram (2373, 1594)
```

[0]:
```
```

4.13.4 Applying TFIDF Vectorizer on Plot_synopsis with char 3 grams

[0]:
```
tfidf_vect_char3  = TfidfVectorizer(ngram_range=(3,3),␣
 ↪analyzer='char',min_df=10, max_features=20000)
tfidf_train_char3 = tfidf_vect_char3.fit_transform(train.clean_plot_synopsis)
tfidf_test_char3  = tfidf_vect_char3.transform(test.clean_plot_synopsis)
tfidf_val_char3    = tfidf_vect_char3.transform(val.clean_plot_synopsis)
```

[0]:
```
print('Shape of train tfidf char3 gram', tfidf_train_char3.shape)
print('Shape of test tfidf char3 gram ', tfidf_test_char3.shape)
print('Shape of val  tfidf char3 gram', tfidf_val_char3.shape)
```

```
Shape of train tfidf char3 gram (9489, 7944)
Shape of test tfidf char3 gram  (2966, 7944)
Shape of val  tfidf char3 gram (2373, 7944)
```

4.13.5 Applying TFIDF Vectorizer on Plot_synopsis with char 4 grams

[0]:
```
tfidf_vect_char4  = TfidfVectorizer(ngram_range=(4,4),␣
 ↪analyzer='char',min_df=10, max_features=20000)
tfidf_train_char4 = tfidf_vect_char4.fit_transform(train.clean_plot_synopsis)
tfidf_test_char4  = tfidf_vect_char4.transform(test.clean_plot_synopsis)
tfidf_val_char4    = tfidf_vect_char4.transform(val.clean_plot_synopsis)
```

[0]:
```
print('Shape of train tfidf char4 gram', tfidf_train_char4.shape)
print('Shape of test tfidf char4 gram ', tfidf_test_char4.shape)
print('Shape of val  tfidf char4 gram', tfidf_val_char4.shape)
```

```
Shape of train tfidf char4 gram (9489, 20000)
Shape of test tfidf char4 gram  (2966, 20000)
Shape of val  tfidf char4 gram (2373, 20000)
```

4.13.6 Applying TFIDF Vectorizer on Plot_synopsis with char 3,4 grams

[0]:
```
tfidf_vect_char34  = TfidfVectorizer(ngram_range=(3,4),␣
 ↪analyzer='char',min_df=10, max_features=20000)
tfidf_train_char34 = tfidf_vect_char34.fit_transform(train.clean_plot_synopsis)
tfidf_test_char34  = tfidf_vect_char34.transform(test.clean_plot_synopsis)
tfidf_val_char34    = tfidf_vect_char34.transform(val.clean_plot_synopsis)
```

[0]:
```
print('Shape of train tfidf char34 gram', tfidf_train_char34.shape)
print('Shape of test tfidf char34 gram ', tfidf_test_char34.shape)
print('Shape of val  tfidf char34 gram', tfidf_val_char34.shape)
```

```
Shape of train tfidf char34 gram (9489, 20000)
Shape of test tfidf char34 gram  (2966, 20000)
Shape of val  tfidf char34 gram (2373, 20000)
```

4.13.7 Joining Uni + Bi + Tri

4.13.8 Tfidf

[0]:
```
tfidf_train_ubt = hstack((tfidf_train_uni, tfidf_train_bi, tfidf_train_tri)).
 ↪tocsr()
tfidf_test_ubt  = hstack((tfidf_test_uni, tfidf_test_bi, tfidf_test_tri)).
 ↪tocsr()
tfidf_val_ubt   = hstack((tfidf_val_uni, tfidf_val_bi, tfidf_val_tri)).tocsr()
```

[0]:
```
print('Shape of train ubt',tfidf_train_ubt.shape)
print('Shape of test  ubt',tfidf_test_ubt.shape)
print('Shape of val   ubt',tfidf_val_ubt.shape)
```

```
Shape of train ubt (9489, 37200)
Shape of test  ubt (2966, 37200)
Shape of val   ubt (2373, 37200)
```

4.13.9 Joining Char 3 + Char 4

4.13.10 Tfidf

[0]:
```
tfidf_train_c34 = hstack((tfidf_train_char3, tfidf_train_char4)).tocsr()
tfidf_test_c34  = hstack((tfidf_test_char3, tfidf_test_char4)).tocsr()
tfidf_val_c34   = hstack((tfidf_val_char3, tfidf_val_char4)).tocsr()
```

[0]:
```
print('Shape of train c34',tfidf_train_c34.shape)
print('Shape of test  c34',tfidf_test_c34.shape)
print('Shape of val   c34',tfidf_val_c34.shape)
```

```
Shape of train c34 (9489, 27944)
Shape of test  c34 (2966, 27944)
Shape of val   c34 (2373, 27944)
```

4.13.11 Joining Uni + Bi + Tri + Char3 + Char4

4.13.12 Tfidf

[0]:
```
tfidf_train_ubtc34 = hstack((tfidf_train_ubt, tfidf_train_c34))
tfidf_test_ubtc34  = hstack((tfidf_test_ubt, tfidf_test_c34))
tfidf_val_ubtc34   = hstack((tfidf_val_ubt, tfidf_val_c34))
```

[0]:
```
print('Shape of train ubtc34',tfidf_train_ubtc34.shape)
print('Shape of test ubtc34 ',tfidf_test_ubtc34.shape)
print('Shape of val ubtc34  ',tfidf_val_ubtc34.shape)
```

```
Shape of train ubtc34 (9489, 65144)
Shape of test ubtc34  (2966, 65144)
Shape of val ubtc34   (2373, 65144)
```

[0]:
```
# #Making Copy of pickle files
# for i in os.listdir(''):
#   if i[-3:] == 'pkl':
#       shutil.copyfile('' + i,'copy_of_pkls/'+i)
#       # os.remove('' + i)
# print('done')
```

5 Applying Models

5.1 Logistic Regression(SGD)

5.1.1 On Uni Gram Vectors

5.1.2 On TFIDF Vectors

[0]:
```
final_tab = PrettyTable(['n-grams','Vector','Model','Tags','Alpha','Hammming␣
↪Loss','F1Score'])
```

[0]:
```
from prettytable import from_html_one
```

[0]:
```
with open('final_tab.html','r') as f:

  final_tab = from_html_one(f.read())
```

[0]:
```
def apply_algo(xtrain,xtest,xval,ytr,yte,yval,loss = 'log',algo =␣
↪'sgd',ngm='Uni',vec='Tfidf',tag = 'Top3'):

        '''This is a sgd function that helps to hyperparameter tune and get␣
↪f1score for SVM and Logistic Regression
           with SGD and appends hamming loss, f1score for final table
           Parameters:
               xtrain:- Train Data
               xtest :- Test Data
               xval  :- Validation Data
               ytr   :- Y Train Data
               yte   :- Y Test Data
               yval  :- Y Validation Data
               loss  :-"log" for logistic regression
                       "sgd" for svm

               algo  :- "sgd" to use SGD algorithm
```

```
                    "<other than sgd> to use normal Logistic Regression␣
↪algorithem"
              ngm   :-While appending f1scores to final table we can represent␣
↪what kind of data is used to get
                    this f1score (Uni,Bi,Tri,Char,Char3,Char4,Char34...etc)
              vec   :-While appending f1scores to final table we can represent␣
↪what kind of data is used to get
                    this f1score (Tfidf,Countvect...etc)
              tag   :-While appending f1scores to final table we can represent␣
↪what kind of target data
                    is used to get this f1score (Top3,Top5...etc)'''


      alpha = [0.000001,0.00001,0.0001,0.001,0.01,0.1,1,10,100,1000]



      # for i in alpha:

      #     model = OneVsRestClassifier(SGDClassifier(loss=loss, alpha=i,␣
↪verbose=0, class_weight='balanced'))
      #     model.fit(xtrain,ytrain)
      #     # ytrainpred = model.predict(xtrain)
      #     ypred = model.predict(xval)
      #     hl = hamming_loss(yval, ypred)
      #     p_score = precision_score(yval,ypred, average='micro')
      #     recall   = recall_score(yval, ypred, average='micro')
      #     f1      = f1_score(yval, ypred, average='micro')
      #     # f1train = f1_score(ytrain,ytrainpred, average='micro')

      #     print('Alpha:{} Hamming Loss:{} Precision:{} Recall:{} f1score:
↪{}'.format(i,hl,p_score,recall,f1))
      #     hyper_tab.add_row([i,hl,p_score, recall,f1])
      # print(hyper_tab)
      # del hyper_tab


      if algo == 'sgd':

          if loss == 'hinge':
            mod = 'SVM'
          elif loss == 'log':
            mod = 'Logistic Regression(SGD)'


          hyper = {'estimator__alpha':alpha,'estimator__penalty':['l1','l2']}
```

```
            classifier = OneVsRestClassifier(SGDClassifier(loss=loss,␣
↪penalty='l2', class_weight='balanced', n_jobs=-1, verbose=10))

        else:
            mod = 'Logistic Regression'
            print('\nApplying Normal Logistic Regression')
            hyper = {'estimator__C':alpha,'estimator__penalty':['l1','l2']}
            classifier =␣
↪OneVsRestClassifier(LogisticRegression(class_weight='balanced', n_jobs=-1))




        GS = GridSearchCV(classifier, hyper, verbose=10, scoring='f1_micro',␣
↪n_jobs=-1)
        sys.stdout.write('Fitting with Gridsearch...\n')
        # print(xtrain.shape)
        # print(ytrain.shape)
        GS.fit(xtrain, ytr)
        # predictions = GS.predict (xval)
        print('*'*50)
        print(GS.best_estimator_)
        print('*'*50)
        sys.stdout.write('****************************Fitting with best␣
↪estimator**********************')
        bgs = GS.best_estimator_
        bgs.fit(xtrain, ytr)
        sys.stdout.write('\npredicting')
        predictions = bgs.predict(xval)
        print("\nVal Accuracy :",metrics.accuracy_score(yval, predictions))
        print("\nVal Hamming loss ",metrics.hamming_loss(yval,predictions))
        print("\nVal f1-score", metrics.f1_score(yval, predictions,␣
↪average='micro'))


        precision = precision_score(yval, predictions, average='micro')
        recall = recall_score(yval, predictions, average='micro')
        f1 = f1_score(yval, predictions, average='micro')

        print("\nMicro-average on test data")
        print("\nPrecision: {:.4f}, Recall: {:.4f}, F1-measure: {:.4f}".
↪format(precision, recall, f1))

        test_pred = bgs.predict(xtest)
        test_hl   = metrics.recall_score(yte,test_pred, average='micro')
```

```
        test_f1   = metrics.f1_score(yte,test_pred, average='micro')
        print (metrics.classification_report(yte, test_pred))


        if mod == 'Logistic Regression':
          hyp = bgs.estimator.C
        else:
          hyp = bgs.estimator.alpha

        print(test_f1)

        final_tab.add_row([ngm,vec,mod,tag,hyp,test_hl,test_f1])
        return bgs
```

[0]:
```
logistic_model = apply_algo(tfidf_train_uni,␣
↪tfidf_test_uni,tfidf_val_uni,ytrain,ytest,yval)
# logistic_model.fit(tfidf_train_plot_syn_uni,ytrain)
```

5.2 On Bi-Grams

5.3 On TFIDF Vector

[0]:
```
logistic_model_bi = apply_algo(tfidf_train_bi,␣
↪tfidf_test_bi,tfidf_val_bi,ytrain,ytest,yval, ngm='Bi')
```

5.4 On Tri-Grams

5.4.1 On TFIDF Vector

[0]:
```
logistic_model_tri = apply_algo(tfidf_train_tri,␣
↪tfidf_test_tri,tfidf_val_tri,ytrain,ytest,yval,ngm='Tri')
```

5.5 On Char 3

5.5.1 On TFIDF Vector

[0]:
```
logistic_model_char = apply_algo(tfidf_train_char3,␣
↪tfidf_test_char3,tfidf_val_char3,ytrain,ytest,yval,ngm='Char 3')
```

5.6 On Char 4

5.6.1 On TFIDF Vector

[0]:
```
logistic_model_char = apply_algo(tfidf_train_char4,␣
↪tfidf_test_char4,tfidf_val_char4,ytrain,ytest,yval,ngm='Char 4')
```

[0]:
```
print(final_tab)
```

5.7 On Char 3,4

5.7.1 On TFIDF Vector

[0]:
```
logistic_model_char = apply_algo(tfidf_train_char34,␣
↪tfidf_test_char34,tfidf_val_char34,ytrain,ytest,yval,ngm='Char 34')
```

5.8 On Uni + Bi + Tri

5.8.1 On TFIDF Vector

[0]:
```
logistic_model_char = apply_algo(tfidf_train_ubt,␣
↪tfidf_test_ubt,tfidf_val_ubt,ytrain,ytest,yval,ngm='U+B+T')
```

5.9 On Uni + Bi + Tri + Char3 + Char 4

5.9.1 On TFIDF Vector

[0]:
```
logistic_model_char = apply_algo(tfidf_train_ubtc34,␣
↪tfidf_test_ubtc34,tfidf_val_ubtc34,ytrain,ytest,yval,ngm='U+B+T+C3+C4')
```

[0]:
```
# with open('final_tab.html','w') as f:
#   f.write(final_tab.get_html_string())

with open('final_tab.html','r')  as f:
  print(from_html_one(f.read()))
```

```
+-------------+--------+--------------------------+--------+--------------------
-+----------------------+
|   n-grams   | Vector |          Model           | Alpha  |    Hammming Loss
|       F1Score        |
+-------------+--------+--------------------------+--------+--------------------
-+----------------------+
|     Uni     | Tfidf  | Logistic Regression(SGD) | 0.0001 |  0.4401463090223897
```

```
|  0.3434823977164605 |
|      Bi     | Tfidf  | Logistic Regression(SGD) | 0.001  | 0.35446685878962536
| 0.34948909895634117 |
|     Tri     | Tfidf  | Logistic Regression(SGD) | 0.001  |  0.3563511416537353
| 0.16809139152484773 |
|    Char 3   | Tfidf  | Logistic Regression(SGD) | 0.0001 |  0.4830414542230104
| 0.31824156564918943 |
|    Char 4   | Tfidf  | Logistic Regression(SGD) | 0.0001 |  0.4454666371092884
|  0.3476643598615917 |
|   Char 34   | Tfidf  | Logistic Regression(SGD) | 0.0001 | 0.46209266238084684
| 0.33784440842787683 |
|    U+B+T    | Tfidf  | Logistic Regression(SGD) | 0.001  | 0.42928397251163825
|  0.362742343354875  |
| U+B+T+C3+C4 | Tfidf  | Logistic Regression(SGD) | 0.001  |  0.454555530924407
|  0.3701430569971569 |
+-------------+--------+--------------------------+--------+--------------------
-+----------------------+
```

6 Normal Logistic Regression

6.0.1 On Uni Gram Vectors

6.0.2 On TFIDF Vectors

[0]:
```
l_model_uni = apply_algo(tfidf_train_uni,␣
↪tfidf_test_uni,tfidf_val_uni,ytrain,ytest,yval, algo = 'n')
```

6.1 On Bi-Grams

6.2 On TFIDF Vector

[0]:
```
l_model_bi = apply_algo(tfidf_train_bi,␣
↪tfidf_test_bi,tfidf_val_bi,ytrain,ytest,yval,ngm='Bi',algo='n')
```

6.3 On Tri-Grams

6.3.1 On TFIDF Vector

[0]:
```
l_model_tri = apply_algo(tfidf_train_tri,␣
↪tfidf_test_tri,tfidf_val_tri,ytrain,ytest,yval,ngm='Tri',algo = 'n')
```

6.4 On Char 3

6.4.1 On TFIDF Vector

[0]:
```
l_model_char = apply_algo(tfidf_train_char3,␣
↪tfidf_test_char3,tfidf_val_char3,ytrain,ytest,yval,ngm='Char 3',algo = 'n')
```

[0]:
```
```

6.5 On Char 4

6.5.1 On TFIDF Vector

[0]:
```
l_model_char = apply_algo(tfidf_train_char4,␣
↪tfidf_test_char4,tfidf_val_char4,ytrain,ytest,yval, loss = 'hinge',ngm='Char␣
↪4',algo = 'n')
```

6.6 On Char 3,4

6.6.1 On TFIDF Vector

[0]:
```
l_model_char = apply_algo(tfidf_train_char34,␣
↪tfidf_test_char34,tfidf_val_char34,ytrain,ytest,yval,ngm='Char 34',loss =␣
↪'hinge',algo = 'n')
```

6.7 On Uni + Bi + Tri

6.7.1 On TFIDF Vector

[0]:
```
l_model_char = apply_algo(tfidf_train_ubt,␣
↪tfidf_test_ubt,tfidf_val_ubt,ytrain,ytest,yval,ngm='U+B+T',loss=␣
↪'hinge',algo = 'n')
```

```
Applying Normal Logistic Regression
Fitting with Gridsearch...
Fitting 3 folds for each of 20 candidates, totalling 60 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 tasks      | elapsed:   12.1s
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed:   22.6s
[Parallel(n_jobs=-1)]: Done  16 tasks      | elapsed:   41.4s
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed:  1.0min
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:  2.0min
```

```
[Parallel(n_jobs=-1)]: Done  45 tasks      | elapsed:  4.0min
[Parallel(n_jobs=-1)]: Done  52 out of  60 | elapsed:  5.8min remaining:   53.9s
[Parallel(n_jobs=-1)]: Done  60 out of  60 | elapsed:  8.5min finished

**************************************************
OneVsRestClassifier(estimator=LogisticRegression(C=0.1, class_weight='balanced',
                                                 dual=False, fit_intercept=True,
                                                 intercept_scaling=1,
                                                 l1_ratio=None, max_iter=100,
                                                 multi_class='warn', n_jobs=-1,
                                                 penalty='l2',
                                                 random_state=None,
                                                 solver='warn', tol=0.0001,
                                                 verbose=0, warm_start=False),
                    n_jobs=None)
**************************************************
****************************Fitting with best estimator**********************
predicting
Val Accuracy : 0.03876949009692372

Val Hamming loss  0.06148394793540001

Val f1-score 0.37100006072014086

Micro-average on test data

Precision: 0.3282, Recall: 0.4266, F1-measure: 0.3710
              precision    recall  f1-score   support

           0       0.65      0.70      0.68      1155
           1       0.54      0.69      0.61       911
           2       0.32      0.44      0.37       596
           3       0.41      0.59      0.48       587
           4       0.32      0.50      0.39       551
           5       0.33      0.50      0.40       507
           6       0.32      0.41      0.36       395
           7       0.25      0.40      0.31       368
           8       0.20      0.39      0.27       228
           9       0.30      0.52      0.38       190
          10       0.15      0.22      0.18       172
          11       0.20      0.23      0.21       173
          12       0.19      0.25      0.21       159
          13       0.24      0.56      0.34       145
          14       0.23      0.43      0.30       129
          15       0.21      0.26      0.23       148
          16       0.18      0.25      0.21       136
          17       0.12      0.10      0.11       119
          18       0.46      0.40      0.43       113
```

19	0.23	0.30	0.26	129
20	0.14	0.19	0.16	115
21	0.11	0.19	0.14	117
22	0.28	0.52	0.37	92
23	0.11	0.16	0.13	93
24	0.09	0.10	0.10	97
25	0.29	0.49	0.37	83
26	0.08	0.11	0.09	83
27	0.09	0.13	0.11	85
28	0.09	0.13	0.11	79
29	0.11	0.09	0.10	66
30	0.23	0.38	0.29	66
31	0.09	0.06	0.07	65
32	0.11	0.16	0.13	64
33	0.06	0.02	0.03	56
34	0.12	0.08	0.09	51
35	0.09	0.08	0.08	51
36	0.10	0.06	0.07	36
37	0.12	0.09	0.10	45
38	0.00	0.00	0.00	40
39	0.03	0.02	0.03	42
40	0.10	0.05	0.06	44
41	0.19	0.11	0.14	35
42	0.06	0.03	0.04	37
43	0.07	0.02	0.04	41
44	0.12	0.05	0.07	39
45	0.00	0.00	0.00	40
46	0.12	0.07	0.09	28
47	0.00	0.00	0.00	29
48	0.27	0.23	0.25	26
49	0.14	0.20	0.16	25
50	0.07	0.05	0.06	22
51	0.29	0.07	0.11	28
52	0.17	0.04	0.06	27
53	0.25	0.06	0.10	32
54	0.24	0.23	0.24	30
55	0.17	0.07	0.10	27
56	0.00	0.00	0.00	18
57	0.29	0.33	0.31	18
58	0.00	0.00	0.00	15
59	0.00	0.00	0.00	13
60	0.00	0.00	0.00	15
61	0.00	0.00	0.00	13
62	0.00	0.00	0.00	11
63	0.50	0.09	0.15	11
64	0.40	0.29	0.33	14
65	0.11	0.08	0.10	12
66	0.00	0.00	0.00	8

```
           67       0.00      0.00      0.00         5
           68       0.00      0.00      0.00         8
           69       0.00      0.00      0.00         9
           70       0.00      0.00      0.00         5

   micro avg       0.33      0.42      0.37      9022
   macro avg       0.17      0.19      0.16      9022
weighted avg       0.32      0.42      0.36      9022
 samples avg       0.33      0.45      0.33      9022

0.36810650600068023
```

6.8 On Uni + Bi + Tri + Char3 + Char 4

6.8.1 On TFIDF Vector

[0]:
```
l_model_char = apply_algo(tfidf_train_ubtc34,␣
↪tfidf_test_ubtc34,tfidf_val_ubtc34,ytrain,ytest,yval,ngm='U+B+T+C3+C4',loss='hinge',algo␣
↪= 'n')
```

```
Applying Normal Logistic Regression
Fitting with Gridsearch…
Fitting 3 folds for each of 20 candidates, totalling 60 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done   2 tasks      | elapsed:  1.4min
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed:  3.5min
[Parallel(n_jobs=-1)]: Done  16 tasks      | elapsed:  6.1min
[Parallel(n_jobs=-1)]: Done  25 tasks      | elapsed:  9.2min
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed: 17.0min
[Parallel(n_jobs=-1)]: Done  45 tasks      | elapsed: 28.2min
[Parallel(n_jobs=-1)]: Done  52 out of  60 | elapsed: 45.1min remaining:  6.9min
[Parallel(n_jobs=-1)]: Done  60 out of  60 | elapsed: 66.4min finished

**************************************************
OneVsRestClassifier(estimator=LogisticRegression(C=0.1, class_weight='balanced',
                                                 dual=False, fit_intercept=True,
                                                 intercept_scaling=1,
                                                 l1_ratio=None, max_iter=100,
                                                 multi_class='warn', n_jobs=-1,
                                                 penalty='l2',
                                                 random_state=None,
                                                 solver='warn', tol=0.0001,
                                                 verbose=0, warm_start=False),
                    n_jobs=None)
**************************************************
**************************Fitting with best estimator**********************
```

```
predicting
Val Accuracy : 0.033291192583227984

Val Hamming loss  0.06202999709169471

Val f1-score 0.38082824811896443

Micro-average on test data

Precision: 0.3308, Recall: 0.4488, F1-measure: 0.3808
              precision    recall  f1-score   support

           0       0.65      0.71      0.68      1155
           1       0.55      0.69      0.61       911
           2       0.32      0.45      0.37       596
           3       0.42      0.60      0.49       587
           4       0.34      0.54      0.42       551
           5       0.31      0.49      0.38       507
           6       0.33      0.47      0.39       395
           7       0.26      0.43      0.32       368
           8       0.21      0.44      0.28       228
           9       0.30      0.55      0.39       190
          10       0.18      0.27      0.22       172
          11       0.21      0.29      0.25       173
          12       0.20      0.29      0.24       159
          13       0.24      0.56      0.34       145
          14       0.22      0.46      0.30       129
          15       0.21      0.29      0.24       148
          16       0.19      0.29      0.23       136
          17       0.15      0.11      0.13       119
          18       0.47      0.42      0.45       113
          19       0.25      0.36      0.30       129
          20       0.10      0.15      0.12       115
          21       0.13      0.22      0.17       117
          22       0.25      0.55      0.35        92
          23       0.12      0.18      0.14        93
          24       0.10      0.10      0.10        97
          25       0.29      0.58      0.39        83
          26       0.08      0.08      0.08        83
          27       0.13      0.20      0.16        85
          28       0.09      0.14      0.11        79
          29       0.08      0.06      0.07        66
          30       0.26      0.50      0.34        66
          31       0.16      0.12      0.14        65
          32       0.13      0.17      0.15        64
          33       0.05      0.02      0.03        56
          34       0.19      0.16      0.17        51
          35       0.10      0.08      0.09        51
```

```
           36       0.16      0.08      0.11        36
           37       0.22      0.13      0.17        45
           38       0.00      0.00      0.00        40
           39       0.12      0.05      0.07        42
           40       0.14      0.05      0.07        44
           41       0.20      0.11      0.15        35
           42       0.08      0.05      0.06        37
           43       0.00      0.00      0.00        41
           44       0.12      0.05      0.07        39
           45       0.18      0.05      0.08        40
           46       0.21      0.11      0.14        28
           47       0.00      0.00      0.00        29
           48       0.31      0.35      0.33        26
           49       0.14      0.20      0.17        25
           50       0.12      0.05      0.07        22
           51       0.19      0.11      0.14        28
           52       0.29      0.07      0.12        27
           53       0.33      0.06      0.11        32
           54       0.21      0.30      0.25        30
           55       0.12      0.07      0.09        27
           56       0.25      0.06      0.09        18
           57       0.35      0.39      0.37        18
           58       0.00      0.00      0.00        15
           59       0.00      0.00      0.00        13
           60       0.00      0.00      0.00        15
           61       0.00      0.00      0.00        13
           62       0.00      0.00      0.00        11
           63       1.00      0.18      0.31        11
           64       0.36      0.29      0.32        14
           65       0.10      0.08      0.09        12
           66       0.00      0.00      0.00         8
           67       0.00      0.00      0.00         5
           68       0.00      0.00      0.00         8
           69       0.00      0.00      0.00         9
           70       0.00      0.00      0.00         5

   micro avg       0.33      0.44      0.38      9022
   macro avg       0.19      0.21      0.18      9022
weighted avg       0.33      0.44      0.37      9022
 samples avg       0.34      0.48      0.34      9022

0.37805514688434344
```

6.9 Conclusions & Observations

6.9.1 For Tfidf Models

[0]:
```
with open('lfinal_tab.html','r')  as f:
  print(from_html_one(f.read()))

with open('final_tab.html','r')  as f:
  print(from_html_one(f.read()))
```

```
+-------------+--------+---------------------+-------+---------------------+----
-----------------+
|   n-grams   | Vector |        Model        | Alpha |    Hammming Loss    |
F1Score       |
+-------------+--------+---------------------+-------+---------------------+----
-----------------+
|     Uni     | Tfidf  | Logistic Regression |   1   |  0.4340500997561516 |
0.35008045771500085 |
|      Bi     | Tfidf  | Logistic Regression |  0.01 | 0.40279317224562183 |
0.3334403817039042 |
|     Tri     | Tfidf  | Logistic Regression |   1   | 0.33274218576812237 |
0.16952311037072593 |
|    Char 3   | Tfidf  | Logistic Regression |   1   |  0.4730658390600754 |
0.32093845170507956 |
|    Char 4   | Tfidf  | Logistic Regression |   1   |  0.4424739525604079 |
0.35379093366419995 |
|   Char 34   | Tfidf  | Logistic Regression |   1   | 0.46575038794058965 |
0.3469000247667795 |
|    U+B+T    | Tfidf  | Logistic Regression |  0.1  |  0.4198625581910885 |
0.36810650600068023 |
| U+B+T+C3+C4 | Tfidf  | Logistic Regression |  0.1  |  0.4414763910441144 |
0.37805514688434344 |
+-------------+--------+---------------------+-------+---------------------+----
-----------------+
+-------------+--------+-------------------------+--------+--------------------
-+----------------------+
|   n-grams   | Vector |          Model          | Alpha  |    Hammming Loss
|       F1Score        |
+-------------+--------+-------------------------+--------+--------------------
-+----------------------+
|     Uni     | Tfidf  | Logistic Regression(SGD) | 0.0001 |  0.4401463090223897
|  0.3434823977164605 |
|      Bi     | Tfidf  | Logistic Regression(SGD) | 0.001  | 0.35446685878962536
| 0.34948909895634117 |
|     Tri     | Tfidf  | Logistic Regression(SGD) | 0.001  |  0.3563511416537353
| 0.16809139152484773 |
|    Char 3   | Tfidf  | Logistic Regression(SGD) | 0.0001 |  0.4830414542230104
```

```
| 0.31824156564918943 |
|    Char 4   | Tfidf  | Logistic Regression(SGD) | 0.0001 |  0.4454666371092884
|  0.3476643598615917 |
|   Char 34   | Tfidf  | Logistic Regression(SGD) | 0.0001 | 0.46209266238084684
| 0.33784440842787683 |
|    U+B+T    | Tfidf  | Logistic Regression(SGD) | 0.001  | 0.42928397251163825
|  0.362742343354875  |
| U+B+T+C3+C4 | Tfidf  | Logistic Regression(SGD) | 0.001  |  0.454555530924407
|  0.3701430569971569 |
+-------------+--------+--------------------------+--------+--------------------
-+---------------------+
```

6.9.2 Observations

```
<li>The max f1score that is obtained is 37.805% ~ 38%</li>
<li>Almost all the vectors are having above 30% f1score exepth Tri gram vector</li>
<li>Normal logistic regression is getting highest f1score compared with sgd logistic regressio
<li>The best alpha value  is 0.001 for logistic regression and 0.1 for sgd logistic regression
```

6.10 Steps Followed

Note:Only few machine learning models or vectors were listed in this documented, ML models or vectors that were with low f1score is removed from this document

Getting Data: Data is obtained from Kaggle

```
<li>After getting data it saved to local disk for future use</li>
<li>Data is split based on split  column in data frame</li>
```

Preprocessing:

```
    <li>There were other language characters in the data, which are supposed to be removed o
    <li>Special Characters like î,¿,,,´,-,*, ,",®,ą were removed</li>
</ul>
```

EDA:There where 71 unique tags and most of the plot synopsis had 2 or 3 or 5 tags

Vectorizing:Our tags are transformed to multilabels and our data is vectorized with following techniques

```
<li>Tfidf Uni Gram</li>
<li>Tfidf Bi Gram</li>
<li>Tfidf Tri Gram</li>
<li>Tfidf Char 3</li>
<li>Tfidf Char 4</li>
<li>Tfidf Char 3 + 4</li>
<li>Tfidf Uni + Bi + Tri + Char 3 + Char 4</li>
```

Algorithms Used

```
<li>Logistic Regression</li>
<li>SGD with Log loss</li>
```