```
!pip install torch==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio===0.8.1 -f https://downlo
#torch 1.9.0+cu102  | 1.8.1+cu111
```

```
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.8.1+cu111
  Downloading https://download.pytorch.org/whl/cu111/torch-1.8.1%2Bcu111-cp37-cp37m-linu
     |█████████████▌                  | 834.1 MB 1.3 MB/s eta 0:14:42tcmalloc: large al
     |█████████████████               | 1055.7 MB 1.4 MB/s eta 0:11:17tcmalloc: large a
     |█████████████████████▌           | 1336.2 MB 1.3 MB/s eta 0:08:06tcmalloc: large a
     |███████████████████████████▍     | 1691.1 MB 1.2 MB/s eta 0:03:59tcmalloc: large a
     |████████████████████████████████| 1982.2 MB 1.2 MB/s eta 0:00:01tcmalloc: large al
tcmalloc: large alloc 2477727744 bytes == 0x55728ca9c000 @  0x7feb58cca615 0x557123fd802
     |████████████████████████████████| 1982.2 MB 1.2 kB/s
Collecting torchvision==0.9.1+cu111
  Downloading https://download.pytorch.org/whl/cu111/torchvision-0.9.1%2Bcu111-cp37-cp37
     |████████████████████████████████| 17.6 MB 1.2 MB/s
Collecting torchaudio===0.8.1
  Downloading torchaudio-0.8.1-cp37-cp37m-manylinux1_x86_64.whl (1.9 MB)
     |████████████████████████████████| 1.9 MB 8.2 MB/s
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from tor
Requirement already satisfied: pillow>=4.1.1 in /usr/local/lib/python3.7/dist-packages (
Installing collected packages: torch, torchvision, torchaudio
  Attempting uninstall: torch
    Found existing installation: torch 1.9.0+cu102
    Uninstalling torch-1.9.0+cu102:
      Successfully uninstalled torch-1.9.0+cu102
  Attempting uninstall: torchvision
    Found existing installation: torchvision 0.10.0+cu102
    Uninstalling torchvision-0.10.0+cu102:
      Successfully uninstalled torchvision-0.10.0+cu102
ERROR: pip's dependency resolver does not currently take into account all the packages t
torchtext 0.10.0 requires torch==1.9.0, but you have torch 1.8.1+cu111 which is incompat
Successfully installed torch-1.8.1+cu111 torchaudio-0.8.1 torchvision-0.9.1+cu111
```

execution time : 5

```
!pip install transformers requests beautifulsoup4 pandas numpy
```

```
Collecting transformers
  Downloading transformers-4.9.2-py3-none-any.whl (2.6 MB)
     |████████████████████████████████| 2.6 MB 7.5 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (2.23
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (1.1.5)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (1.19.5)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from
Collecting pyyaml>=5.1
  Downloading PyYAML-5.4.1-cp37-cp37m-manylinux1_x86_64.whl (636 kB)
     |████████████████████████████████| 636 kB 68.5 MB/s
Collecting tokenizers<0.11,>=0.10.1
  Downloading tokenizers-0.10.3-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manyli
```

```
    |████████████████████████████████| 3.3 MB 58.3 MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packa
Collecting sacremoses
  Downloading sacremoses-0.0.45-py3-none-any.whl (895 kB)
    |████████████████████████████████| 895 kB 62.5 MB/s
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from
Collecting huggingface-hub==0.0.12
  Downloading huggingface_hub-0.0.12-py3-none-any.whl (37 kB)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from sa
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from sac
Installing collected packages: tokenizers, sacremoses, pyyaml, huggingface-hub, transfor
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.0.12 pyyaml-5.4.1 sacremoses-0.0.45 tokenizers-
```

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch
import requests
from bs4 import BeautifulSoup
import re
```

```
import pandas as pd
```

Processed Data

```
df= pd.read_csv('/content/drive/MyDrive/Colab Notebooks/CapstoneGL/imdbgbprep.csv', encoding=
```

```
df.head()
```

	Unnamed: 0	Title	reviewed_by	reviews
		final fantasy the spirits within	evelyn c	capsule this very dark scifi fantasy is

```
df.drop('Unnamed: 0', axis=1, inplace=True)
```

roger ebert asks in his review of sexy

```
def clean_str(string):
  """
  String cleaning before vectorization
  """
  try:
    string = re.sub(r'^https?:\/\/<>.*[\r\n]*', '', string, flags=re.MULTILINE)
    string = re.sub(r"[^A-Za-z]", " ", string)
    words = string.strip().lower().split()
    words = [w for w in words if len(w)>=1]
    return " ".join(words)
  except:
    return ""
```

```
df['clean_reviews'] = df['reviews'].apply(clean_str)
df.head()
```

	Title	reviewed_by	reviews	clean_reviews
0	final fantasy the spirits within 2001	evelyn c leeper	capsule this very dark scifi fantasy is magnif...	capsule this very dark scifi fantasy is magnif...
1	sexy beast 2000	mark r leeper	roger ebert asks in his review of sexy beast w...	roger ebert asks in his review of sexy beast w...
2	final fantasy the spirits within 2001	robin clifford	aliens beings have taken over the earth the gr...	aliens beings have taken over the earth the gr...
		susan	susan grangers review of	susan grangers review of

```
df.loc[0, 'clean_reviews']
```

```
'capsule this very dark scifi fantasy is magnificent visually but it has a nearly incoh
erent plot final fantasy is a japaneseamerican coproduction entirely animated but with
a very real threedimensional look and with very reallooking characters in the year alie
ns that appear to us as translucent images but still very deadly creatures have invaded
earth saving the earth requires resorting to semimystical means to understand and halt
```

Instantiate Model

```
tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
```

```
model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-un
Downloading: 100%          953/953 [00:00<00:00, 25.5kB/s]

Downloading: 100%          872k/872k [00:00<00:00, 1.58MB/s]

Downloading: 100%          112/112 [00:00<00:00, 2.80kB/s]

Downloading: 100%          39.0/39.0 [00:00<00:00, 1.25kB/s]

Downloading: 100%          669M/669M [00:13<00:00, 46.9MB/s]
```

```
!pip install torchinfo
```

```
Collecting torchinfo
  Downloading torchinfo-1.5.3-py3-none-any.whl (19 kB)
Installing collected packages: torchinfo
Successfully installed torchinfo-1.5.3
```

```
from torchinfo import summary
```

```
summary(model, depth=12)
```

```
│    │    │    │    │    │    └─Linear: 7-50           590,592
│    │    │    │    │    │    └─Linear: 7-51           590,592
│    │    │    │    │    │    └─Linear: 7-52           590,592
│    │    │    │    │    │    └─Dropout: 7-53          --
│    │    │    │    │    └─BertSelfOutput: 6-44        --
│    │    │    │    │    │    └─Linear: 7-54           590,592
│    │    │    │    │    │    └─LayerNorm: 7-55        1,536
│    │    │    │    │    │    └─Dropout: 7-56          --
│    │    │    │    └─BertIntermediate: 5-23           --
│    │    │    │    │    └─Linear: 6-45                2,362,368
│    │    │    │    └─BertOutput: 5-24                 --
│    │    │    │    │    └─Linear: 6-46                2,360,064
│    │    │    │    │    └─LayerNorm: 6-47             1,536

│    │    │    │    │    └─Dropout: 6-48               --
│    │    │    └─BertLayer: 4-9                        --
│    │    │    │    └─BertAttention: 5-25              --
│    │    │    │    │    └─BertSelfAttention: 6-49     --
│    │    │    │    │    │    └─Linear: 7-57           590,592
│    │    │    │    │    │    └─Linear: 7-58           590,592
│    │    │    │    │    │    └─Linear: 7-59           590,592
│    │    │    │    │    │    └─Dropout: 7-60          --
│    │    │    │    │    └─BertSelfOutput: 6-50        --
│    │    │    │    │    │    └─Linear: 7-61           590,592
│    │    │    │    │    │    └─LayerNorm: 7-62        1,536
│    │    │    │    │    │    └─Dropout: 7-63          --
│    │    │    │    └─BertIntermediate: 5-26           --
│    │    │    │    │    └─Linear: 6-51                2,362,368
│    │    │    │    └─BertOutput: 5-27                 --
│    │    │    │    │    └─Linear: 6-52                2,360,064
│    │    │    │    │    └─LayerNorm: 6-53             1,536
│    │    │    │    │    └─Dropout: 6-54               --
│    │    │    └─BertLayer: 4-10                       --
│    │    │    │    └─BertAttention: 5-28              --
│    │    │    │    │    └─BertSelfAttention: 6-55     --
│    │    │    │    │    │    └─Linear: 7-64           590,592
│    │    │    │    │    │    └─Linear: 7-64           590,592
│    │    │    │    │    │    └─Linear: 7-65           590,592
│    │    │    │    │    │    └─Linear: 7-66           590,592
│    │    │    │    │    │    └─Dropout: 7-67          --
│    │    │    │    │    └─BertSelfOutput: 6-56      --
│    │    │    │    │    │    └─Linear: 7-68           590,592
│    │    │    │    │    │    └─LayerNorm: 7-69        1,536
│    │    │    │    │    │    └─Dropout: 7-70          --
│    │    │    │    └─BertIntermediate: 5-29           --
│    │    │    │    │    └─Linear: 6-57                2,362,368
│    │    │    │    └─BertOutput: 5-30                 --
│    │    │    │    │    └─Linear: 6-58                2,360,064
│    │    │    │    │    └─LayerNorm: 6-59             1,536
│    │    │    │    │    └─Dropout: 6-60               --
│    │    │    └─BertLayer: 4-11                       --
│    │    │    │    └─BertAttention: 5-31              --
│    │    │    │    │    └─BertSelfAttention: 6-61     --
│    │    │    │    │    │    └─Linear: 7-71           590,592
│    │    │    │    │    │    └─Linear: 7-72           590,592
│    │    │    │    │    │    └─Linear: 7-73           590,592
│    │    │    │    │    │    └─Dropout: 7-74          --
│    │    │    │    │    └─BertSelfOutput: 6-62      --
│    │    │    │    │    │    └─Linear: 7-75           590,592
│    │    │    │    │    │    └─LayerNorm: 7-76        1,536
│    │    │    │    │    │    └─Dropout: 7-77          --
│    │    │    │    └─BertIntermediate: 5-32           --
```

```
print(model)
```

```
BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(105879, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0): BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (1): BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (2): BertLayer(
          (attention): BertAttention(
```

Encode and Calculate Sentiment

```
tokens = tokenizer.encode('It was good but couldve been better. Great', return_tensors='pt')
```

```
result = model(tokens)
```

```
result.logits
```

```
tensor([[-2.7768, -1.2353,  1.4419,  1.9804,  0.4584]],
       grad_fn=<AddmmBackward>)
```

```
int(torch.argmax(result.logits))+1
```

```
4
```

Load Reviews

```
df.head()
```

	Title	reviewed_by	reviews	clean_reviews
0	final fantasy the spirits within 2001	evelyn c leeper	capsule this very dark scifi fantasy is magnif...	capsule this very dark scifi fantasy is magnif...
1	sexy beast 2000	mark r leeper	roger ebert asks in his review of sexy beast w...	roger ebert asks in his review of sexy beast w...
2	final fantasy the spirits within 2001	robin clifford	aliens beings have taken over the earth the gr...	aliens beings have taken over the earth the gr...
		susan	susan grangers review of	susan grangers review of

```
df.drop(['Title','reviewed_by','reviews',], axis=1, inplace=True)
```

```
df.head()
```

	clean_reviews
0	capsule this very dark scifi fantasy is magnif...
1	roger ebert asks in his review of sexy beast w...
2	aliens beings have taken over the earth the gr...
3	susan grangers review of jurassic park iii uni...
4	susan grangers review of final fantasy spirits...

```
df['clean_reviews'].iloc[0]
```

```
'capsule this very dark scifi fantasy is magnificent visually but it has a nearly incoh
erent plot final fantasy is a japaneseamerican coproduction entirely animated but with
a very real threedimensional look and with very reallooking characters in the year alie
ns that appear to us as translucent images but still very deadly creatures have invaded
earth saving the earth requires resorting to semimystical means to understand and halt
the enemy if this film had been done in liveaction the scenes more spectacular than tho
```

```
def sentiment_score(review):
    tokens = tokenizer.encode(review, return_tensors='pt')
    result = model(tokens)
    return int(torch.argmax(result.logits))+1
```

```
sentiment_score(df['clean_reviews'].iloc[10])
```

```
2
```

```
df['clean_reviews'].iloc[10]
```

```
'it has to be a record even with writers alison fouse greg grabianski davepolsky michae
l anthony snowden craig wayans marlon wayans and shawn wayansscary movie still couldnt
come up with a single good scene another recordmight go for the biggest drop in quality
from the original movie to the sequel scary movie was imaginative and funny but its seq
uel is neither longstretches of boredom are interrupted periodically by whispered groan
s ofyuck although outrageous physical comedy can be hilarious as theres something about
```

```
from time import time  # To time our operation
```

```
aging of the bedroom banfing scene from the exorcist they appeared tohave used the same
```

```
t = time()

df['sentiment'] = df['clean_reviews'].apply(lambda x: sentiment_score(x[:512]))

print('Time taken to build : {} mins'.format(round((time() - t) / 60, 2)))
```

```
Time taken to build : 169.0 mins
```

```
df.head()
```

	clean_reviews	sentiment
0	capsule this very dark scifi fantasy is magnif...	3
1	roger ebert asks in his review of sexy beast w...	3
2	aliens beings have taken over the earth the gr...	4
3	susan grangers review of jurassic park iii uni...	4
4	susan grangers review of final fantasy spirits...	4

```
s_counts = df['sentiment'].value_counts()
s_counts
```

```
3    9229
4    7506
2    5423
1    4212
5    1497
Name: sentiment, dtype: int64
```

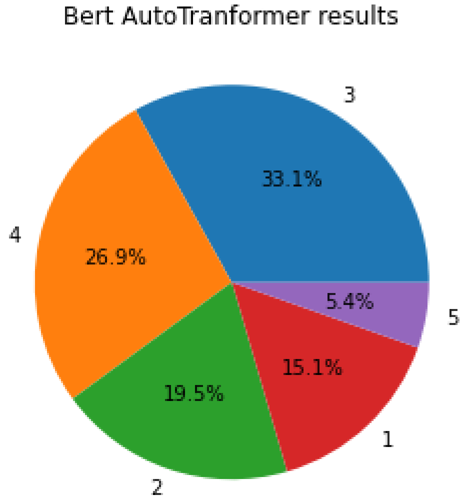
```
df.to_csv(r'/content/drive/MyDrive/Colab Notebooks/CapstoneGL/imdbautomodelgb08152021.csv', i
```

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
Bert_counts= df['sentiment'].value_counts()
```

```
plt.figure(figsize=(15,7))
plt.subplot(1,3,1)
plt.title("Bert AutoTranformer results")
plt.pie(Bert_counts.values, labels = Bert_counts.index, explode = None, autopct='%1.1f%%', sh
```

```
     ([<matplotlib.patches.Wedge at 0x7f05b3d59a90>,
       <matplotlib.patches.Wedge at 0x7f05b3d64350>,
       <matplotlib.patches.Wedge at 0x7f05b3d64c90>,
       <matplotlib.patches.Wedge at 0x7f05b3d6e4d0>,
       <matplotlib.patches.Wedge at 0x7f05b3d6ef90>],
      [Text(0.5564310655783481, 0.9488859095061662, '3'),
       Text(-1.074783013264829, 0.23418256638224508, '4'),
       Text(-0.3541337542486788, -1.0414361641991008, '2'),
       Text(0.7565579160084718, -0.7985111894800964, '1'),
       Text(1.0843723260507565, -0.18476108490499799, '5')],
      [Text(0.30350785395182617, 0.5175741324579088, '33.1%'),
       Text(-0.586245279962634, 0.12773594529940638, '26.9%'),
       Text(-0.19316386595382481, -0.5680560895631458, '19.5%'),
       Text(0.4126679541864391, -0.43555155789823435, '15.1%'),
       Text(0.5914758142095035, -0.10077877358454435, '5.4%')])
```



```
Class = { 1: 'Negative',2: 'Partially_Negative',3: 'Neutral',4: 'Partially_Positive',5: 'Posi
```

```
t = time()

df.sentiment =[Class[item] for item in df.sentiment]

print('Time taken to build : {} mins'.format(round((time() - t) / 60, 2)))
```

```
     Time taken to build : 0.0 mins
```

```
df.head()
```

	clean_reviews	sentiment
0	capsule this very dark scifi fantasy is magnif...	Neutral
1	roger ebert asks in his review of sexy beast w...	Neutral
2	aliens beings have taken over the earth the gr...	Partially_Positive

```
df['clean_reviews'].iloc[2]
```

```
'aliens beings have taken over the earth the great cities are deserted andprecious few
humans remain to repel the invaders and reclaim the world formankind aki ross voice of
mingna and her mentor dr sid voice ofdonald sutherland must develop their wave theory t
he only antidote tocounter the alien phantoms in this latest video game to become a fea
turelength movie in final fantasy the spirits within it was inevitable ever since pixar
animation brought the tin toy to lifein as the first allcomputer generated and oscarwin
```

```
df['clean_reviews'].iloc[1]
```

```
'roger ebert asks in his review of sexy beast who would have guessed that the most sava
ge maddog frothing gangster in recent movies would be played by ben kingsley my respons
e would be that anyone who has seen alan arkin in wait until dark henry fonda in once u
pon a time in the west or anthony hopkins in the silence of the lambs should have guess
ed it they should know that the way for a film to create a really creepy sociopath is c
ast someone who generally plays mild sympathetic or even ineffectual character roles th
```

```
df.sample(5)
```

	clean_reviews	sentiment
5137	all about my mother todo sobre mi madresony cl...	Partially_Positive
9626	the blair witch projectchadz rating out of ver...	Partially_Positive
85	legally blonde reese witherspoon luke wilson s...	Negative
8183	the cinema of the s will be remembered for two...	Partially_Negative
18101	batman and robingeorge clooney chris odonnell ...	Partially_Negative

```
df.to_csv(r'/content/drive/MyDrive/Colab Notebooks/CapstoneGL/imdbgb08162021bertsentiment.csv
```

```
df['clean_reviews'].iloc[85]
```

```
'legally blonde reese witherspoon luke wilson selma blair matthew davis victorgarber je
nnifer coolidge holland taylor ali larter screenplay bykaren mccullah lutz and kirsten
smith based on the novel by amandabrown directed by robert luketic minutesrated pg star
s out of five stars review by ed johnsonott nuvo newsweeklywwwnuvocomarchive reviews at
httpreviewsimdbcomreviewsbyedward johnsonottto receive reviews by email at no charge se
nd subscription requests toejohnsonottprodigynet or email ejohnsonottsubscribeonelistco
```

```
df['clean_reviews'].iloc[18101]
```

```
'batman and robingeorge clooney chris odonnell arnold schwarzenegger uma thurmanrating
and out of five stars review by ed johnsonottfor more reviews go to wwwnuvoonlinecom an
d click on film the ads for batman and robin scream the event of the summer is here and
```

```
!pip install ktrain
```

```
Collecting ktrain
  Downloading ktrain-0.27.2.tar.gz (25.3 MB)
     |████████████████████████████████| 25.3 MB 97 kB/s
Collecting scikit-learn==0.23.2
  Downloading scikit_learn-0.23.2-cp37-cp37m-manylinux1_x86_64.whl (6.8 MB)
     |████████████████████████████████| 6.8 MB 23.1 MB/s
Requirement already satisfied: matplotlib>=3.0.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: pandas>=1.0.1 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: fastprogress>=0.1.21 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages (fro
Collecting langdetect
  Downloading langdetect-1.0.9.tar.gz (981 kB)
     |████████████████████████████████| 981 kB 39.0 MB/s
Requirement already satisfied: jieba in /usr/local/lib/python3.7/dist-packages (from
Collecting cchardet
  Downloading cchardet-2.1.7-cp37-cp37m-manylinux2010_x86_64.whl (263 kB)
     |████████████████████████████████| 263 kB 61.2 MB/s
Requirement already satisfied: chardet in /usr/local/lib/python3.7/dist-packages (fro
Collecting syntok
  Downloading syntok-1.3.1.tar.gz (23 kB)
Collecting seqeval==0.0.19
  Downloading seqeval-0.0.19.tar.gz (30 kB)
Collecting transformers<=4.3.3,>=4.0.0
  Downloading transformers-4.3.3-py3-none-any.whl (1.9 MB)
     |████████████████████████████████| 1.9 MB 42.0 MB/s
Collecting sentencepiece
  Downloading sentencepiece-0.1.96-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_
     |████████████████████████████████| 1.2 MB 64.3 MB/s
Collecting keras_bert>=0.86.0
  Downloading keras-bert-0.88.0.tar.gz (26 kB)
Requirement already satisfied: networkx>=2.3 in /usr/local/lib/python3.7/dist-package
Collecting whoosh
  Downloading Whoosh-2.7.4-py2.py3-none-any.whl (468 kB)
     |████████████████████████████████| 468 kB 54.9 MB/s
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-2.2.0-py3-none-any.whl (12 kB)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: Keras>=2.2.4 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from K
Collecting keras-transformer>=0.39.0
  Downloading keras-transformer-0.39.0.tar.gz (11 kB)
Collecting keras-pos-embd>=0.12.0
  Downloading keras-pos-embd-0.12.0.tar.gz (6.0 kB)
Collecting keras-multi-head>=0.28.0
  Downloading keras-multi-head-0.28.0.tar.gz (14 kB)
Collecting keras-layer-normalization>=0.15.0
```

```
  Downloading keras-layer-normalization-0.15.0.tar.gz (4.2 kB)
Collecting keras-position-wise-feed-forward>=0.7.0
  Downloading keras-position-wise-feed-forward-0.7.0.tar.gz (4.5 kB)
Collecting keras-embed-sim>=0.9.0
  Downloading keras-embed-sim-0.9.0.tar.gz (4.1 kB)
Collecting keras-self-attention>=0.50.0
  Downloading keras-self-attention-0.50.0.tar.gz (12 kB)
```

```
#Import libraries

import numpy as np
import pandas as pd
import tensorflow as tf
import seaborn as sns
import ktrain
from ktrain import text
from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import re
```

```
df= pd.read_csv('/content/drive/MyDrive/Colab Notebooks/CapstoneGL/imdbgb08162021bertsentimen
```

```
df.sample(5)
```

	Unnamed: 0	clean_reviews	sentiment
15569	15569	mutiny on the bounty is an outstanding film wi...	Partially_Positive
20849	20849	sometimes an audience can work against you for...	Partially_Positive
15571	15571	armageddon written by jonathan hensleigh and j...	Partially_Positive
25910	25910	capsule review what would you get if robert al...	Partially_Positive
6248	6248	is there a lot brewing within anywhere but her...	Partially_Positive

```
df.drop(['Unnamed: 0'], axis=1, inplace=True)
```

```
df.sample(5)
```

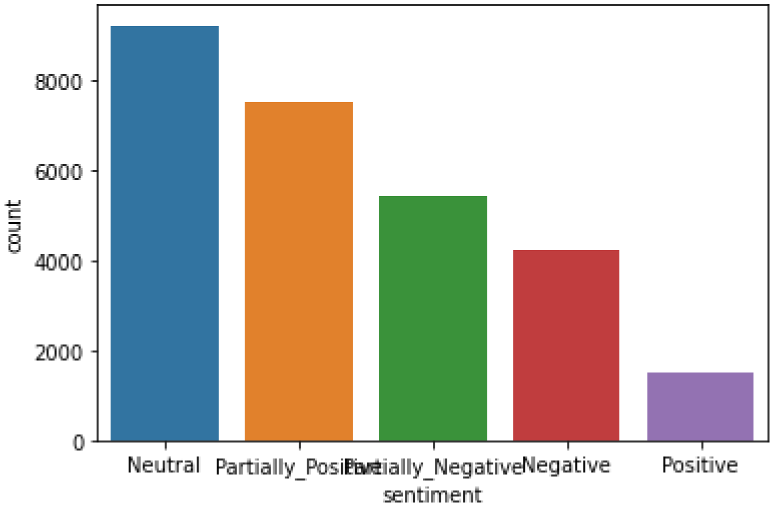
	clean_reviews	sentiment
24422	hey you want just to have some fun at the movi...	Negative
24340	the madness of king george is a movie based on...	Partially_Positive
14027	member of the internet movie critics associati...	Negative

```
s_counts = df['sentiment'].value_counts()
s_counts
```

```
Neutral               9229
Partially_Positive    7506
Partially_Negative    5423
Negative              4212
Positive              1497
Name: sentiment, dtype: int64
```

```
sns.countplot(df["sentiment"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efb58e58450>
```



```
df.isna().sum()/len(df) * 100
```

```
clean_reviews    0.0933
sentiment        0.0000
dtype: float64
```

```
df.isnull().sum()
```

```
clean_reviews    26
sentiment         0
dtype: int64
```

```
df.dropna(inplace=True)
```

```
df.isna().sum()/len(df) * 100
```

```
clean_reviews    0.0
sentiment        0.0
dtype: float64
```

```
s_counts = df['sentiment'].value_counts()
s_counts
```

```
Neutral               9229
Partially_Positive    7480
Partially_Negative    5423
Negative              4212
Positive              1497
Name: sentiment, dtype: int64
```

```
s_counts = df['sentiment'].value_counts()
s_counts
```

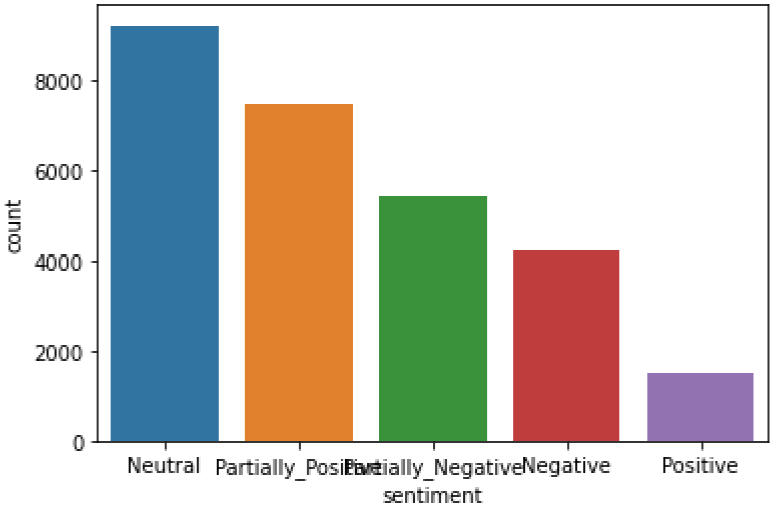
```
Neutral               9229
Partially_Positive    7480
Partially_Negative    5423
Negative              4212
Positive              1497
Name: sentiment, dtype: int64
```

```
s_counts.sum()
```

```
27841
```

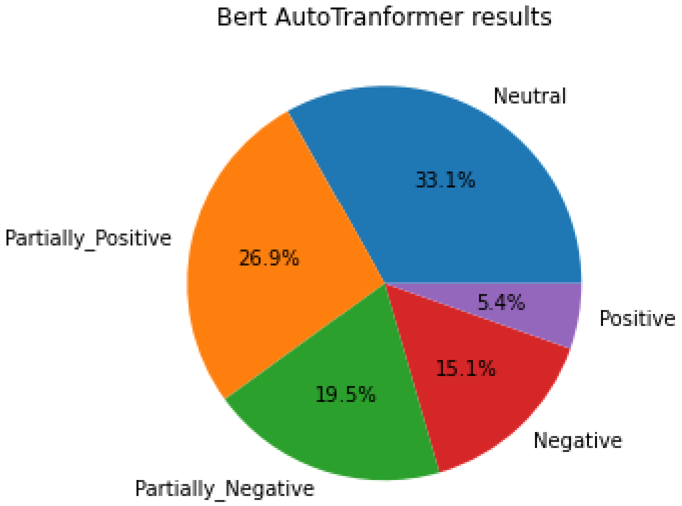
```
sns.countplot(df["sentiment"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efb58dd1150>
```



```
plt.figure(figsize=(15,7))
plt.subplot(1,3,1)
plt.title("Bert AutoTranformer results")
plt.pie(s_counts.values, labels = s_counts.index, explode = None, autopct='%1.1f%%', shadow=F
```

```
    ([<matplotlib.patches.Wedge at 0x7efb54fd5dd0>,
      <matplotlib.patches.Wedge at 0x7efb54fe3550>,
      <matplotlib.patches.Wedge at 0x7efb54fe3dd0>,
      <matplotlib.patches.Wedge at 0x7efb54feb710>,
      <matplotlib.patches.Wedge at 0x7efb54ff6250>],
     [Text(0.5555088100093617, 0.9494261224560777, 'Neutral'),
      Text(-1.074736070577655, 0.23439790655912848, 'Partially_Positive'),
      Text(-0.3559797707356584, -1.0408066116368535, 'Partially_Negative'),
      Text(0.7559518643433558, -0.7990849634399366, 'Negative'),
      Text(1.084343188254864, -0.18493201476563492, 'Positive')],
     [Text(0.3030048054596518, 0.5178687940669514, '33.1%'),
      Text(-0.586219674860539, 0.12785340357770641, '26.9%'),
      Text(-0.19417078403763186, -0.5677126972564656, '19.5%'),
      Text(0.4123373805509213, -0.43586452551269267, '15.1%'),
      Text(0.5914599208662893, -0.10087200805398268, '5.4%')])
```



```
s_counts.sum()
```

```
    27841
```

```
TRAIN_SIZE = 22000
TEST_SIZE = 5840

data_train = df[:TRAIN_SIZE]
data_test = df[TRAIN_SIZE:].reset_index(drop=True)
```

```
data_train.head()
```

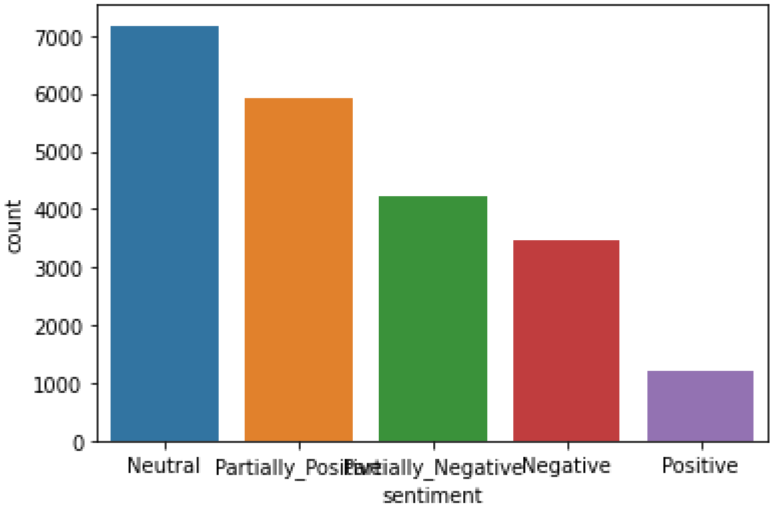
	clean_reviews	sentiment
0	capsule this very dark scifi fantasy is magnif...	Neutral

```
data_train['sentiment'].value_counts()
```

```
Neutral               7174
Partially_Positive    5903
Partially_Negative    4234
Negative              3476
Positive              1213
Name: sentiment, dtype: int64
```

```
sns.countplot(data_train["sentiment"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efb54fabdd0>
```



```
data_train.isna().sum()/len(data_train) * 100
```

```
clean_reviews    0.0
sentiment        0.0
dtype: float64
```

```
data_test.head()
```

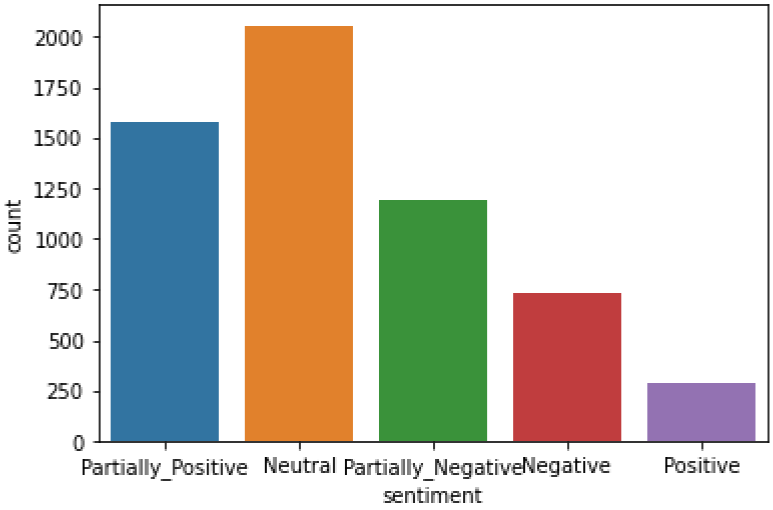
	clean_reviews	sentiment
0	jim jarmuschs stranger than paradise down by l...	Partially_Positive
1	venezuela running length mpaa classification n...	Neutral
2	united states us release date beginning limite...	Partially_Negative
3	united states us release date beginning wideru...	Partially_Negative
4	franceguinea running length mpaa classificatio...	Partially_Negative

```
data_test['sentiment'].value_counts()
```

```
Neutral               2055
Partially_Positive    1577
Partially_Negative    1189
Negative               736
Positive               284
Name: sentiment, dtype: int64
```

```
sns.countplot(data_test["sentiment"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efb54f8d090>
```



```
data_test.isna().sum()/len(data_test) * 100
```

```
clean_reviews    0.0
sentiment        0.0
dtype: float64
```

```
#dimension of the dataset

print("Size of train dataset: ",data_train.shape)
print("Size of test dataset: ",data_test.shape)
```

```
Size of train dataset:  (22000, 2)
Size of test dataset:  (5841, 2)
```

```
# maxlen means it is considering that much words and rest are getting trucated
# preprocess_mode means tokenizing, embedding and transformation of text corpus(here it is co


(X_train, y_train), (X_test, y_test), preproc = text.texts_from_df(train_df=data_train,
                                                                   text_column = 'clean_revie
                                                                   label_columns = 'sentiment
                                                                   val_df = data_test,
                                                                   maxlen = 500,
                                                                   ngram_range=2
```

```
                                                          ngram_range=2,
                                                          preprocess_mode = 'bert')
```

```
['Negative', 'Neutral', 'Partially_Negative', 'Partially_Positive', 'Positive']
   Negative  Neutral  Partially_Negative  Partially_Positive  Positive
0       0.0      1.0                 0.0                 0.0       0.0
1       0.0      1.0                 0.0                 0.0       0.0
2       0.0      0.0                 0.0                 1.0       0.0
3       0.0      0.0                 0.0                 1.0       0.0
4       0.0      0.0                 0.0                 1.0       0.0
['Negative', 'Neutral', 'Partially_Negative', 'Partially_Positive', 'Positive']
   Negative  Neutral  Partially_Negative  Partially_Positive  Positive
0       0.0      0.0                 0.0                 1.0       0.0
1       0.0      1.0                 0.0                 0.0       0.0
2       0.0      0.0                 1.0                 0.0       0.0
3       0.0      0.0                 1.0                 0.0       0.0
4       0.0      0.0                 1.0                 0.0       0.0
downloading pretrained BERT model (uncased_L-12_H-768_A-12.zip)...
[██████████████████████████████████████████████████]
extracting pretrained BERT model...
done.

cleanup downloaded zip...
done.

preprocessing train...
language: en
done.
Is Multi-Label? False
preprocessing test...
language: en
done.
```

```
len(X_train[1])
```

```
22000
```

```
X_train[0].shape
```

```
(22000, 500)
```

```
print('review: \n', X_train[0])
print('label: \n', y_train[0])
```

```
review:
 [[  101 18269  2023 ...  2011 17512   102]
 [  101  5074 22660 ... 19104  1037   102]
 [  101 12114  9552 ... 23805 23808   102]
 ...
 [  101  1996  2732 ... 17729  4945   102]
 [  101  3459  3744 ...  5000  2247   102]
 [  101  1996  2034 ...  2046  1996   102]]
label:
 [0. 1. 0. 0. 0.]
```

BERT Model Building

```
# name = "bert" means, here we are using BERT model.

model = text.text_classifier(name = 'bert',
                             train_data = (X_train, y_train),
                             preproc = preproc)
```

```
Is Multi-Label? False
maxlen is 500
done.
```

```
model.summary()
```

```
Model: "model_1"
__________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================
Input-Token (InputLayer)        [(None, 500)]        0
__________________________________________________________________________________
Input-Segment (InputLayer)      [(None, 500)]        0
__________________________________________________________________________________
Embedding-Token (TokenEmbedding [(None, 500, 768), ( 23440896    Input-Token[0][0]
__________________________________________________________________________________
Embedding-Segment (Embedding)   (None, 500, 768)     1536        Input-Segment[0][0]
__________________________________________________________________________________
Embedding-Token-Segment (Add)   (None, 500, 768)     0           Embedding-Token[0][0
                                                                 Embedding-Segment[0]
__________________________________________________________________________________
Embedding-Position (PositionEmb (None, 500, 768)     384000      Embedding-Token-Segm
__________________________________________________________________________________
Embedding-Dropout (Dropout)     (None, 500, 768)     0           Embedding-Position[0
__________________________________________________________________________________
Embedding-Norm (LayerNormalizat (None, 500, 768)     1536        Embedding-Dropout[0]
__________________________________________________________________________________
Encoder-1-MultiHeadSelfAttentio (None, 500, 768)     2362368     Embedding-Norm[0][0]
__________________________________________________________________________________
Encoder-1-MultiHeadSelfAttentio (None, 500, 768)     0           Encoder-1-MultiHeadS
__________________________________________________________________________________
Encoder-1-MultiHeadSelfAttentio (None, 500, 768)     0           Embedding-Norm[0][0]
                                                                 Encoder-1-MultiHeadS
__________________________________________________________________________________
Encoder-1-MultiHeadSelfAttentio (None, 500, 768)     1536        Encoder-1-MultiHeadS
__________________________________________________________________________________
Encoder-1-FeedForward (FeedForw (None, 500, 768)     4722432     Encoder-1-MultiHeadS
__________________________________________________________________________________
Encoder-1-FeedForward-Dropout ( (None, 500, 768)     0           Encoder-1-FeedForwar
__________________________________________________________________________________
Encoder-1-FeedForward-Add (Add) (None, 500, 768)     0           Encoder-1-MultiHeadS
                                                                 Encoder-1-FeedForwar
__________________________________________________________________________________
Encoder-1-FeedForward-Norm (Lay (None, 500, 768)     1536        Encoder-1-FeedForwar
__________________________________________________________________________________
```

```
Encoder-2-MultiHeadSelfAttentio (None, 500, 768)     2362368     Encoder-1-FeedForwar

Encoder-2-MultiHeadSelfAttentio (None, 500, 768)     0           Encoder-2-MultiHeadS

Encoder-2-MultiHeadSelfAttentio (None, 500, 768)     0           Encoder-1-FeedForwar
                                                                 Encoder-2-MultiHeadS

Encoder-2-MultiHeadSelfAttentio (None, 500, 768)     1536        Encoder-2-MultiHeadS

Encoder-2-FeedForward (FeedForw (None, 500, 768)     4722432     Encoder-2-MultiHeadS

Encoder-2-FeedForward-Dropout ( (None, 500, 768)     0           Encoder-2-FeedForwar

Encoder-2-FeedForward-Add (Add) (None, 500, 768)     0           Encoder-2-MultiHeadS
                                                                 Encoder-2-FeedForwar

Encoder-2-FeedForward-Norm (Lay (None, 500, 768)     1536        Encoder-2-FeedForwar

Encoder-3-MultiHeadSelfAttentio (None, 500, 768)     2362368     Encoder-2-FeedForwar
```

```
#here we have taken batch size as 6 as from the documentation it is recommend to use this wit

learner = ktrain.get_learner(model=model, train_data=(X_train, y_train),
                   val_data = (X_test, y_test),
                   batch_size = 6)
```

```
#Essentially fit is a very basic training loop, where as fit one cycle uses the one cycle pol

learner.fit_onecycle(lr = 2e-5, epochs = 1)
```

```
begin training using onecycle policy with max lr of 2e-05...
3667/3667 [==============================] - 4085s 1s/step - loss: 1.2004 - accuracy: 0
<tensorflow.python.keras.callbacks.History at 0x7efb4010c090>
```

```
predictor = ktrain.get_predictor(learner.model, preproc)
predictor.save("/content/drive/MyDrive/Colab Notebooks/CapstoneGL/new_model")
```

```
df.loc[4, 'sentiment']
```

```
'Partially_Positive'
```

```
#sample dataset to test on

data = ['movie was half good watchable but not great','this movie was horrible, the plot was
        'the fild is really sucked. there is not plot and acting was bad',
        'what a beautiful movie. great plot. acting was good. will see it again',]
```

```
predictor_load.predict(data)
```

```
['Neutral', 'Partially_Negative', 'Negative', 'Partially_Positive']
```

```
#return_proba = True means it will give the prediction probabilty for each class

predictor_load.predict(data, return_proba=True)
```

```
array([[3.8568873e-03, 8.1107748e-01, 1.5770593e-01, 2.6736544e-02,
        6.2313885e-04],
       [3.1286815e-01, 7.5921856e-02, 6.0201305e-01, 7.7268970e-03,
        1.4700212e-03],
       [6.7886770e-01, 1.2580841e-02, 3.0633354e-01, 1.7213557e-03,
        4.9658422e-04],
       [1.4146597e-02, 7.1383864e-02, 2.0277960e-02, 5.3569973e-01,
        3.5849184e-01]], dtype=float32)
```

```
#classes available

predictor_load.get_classes()
```

```
['Negative', 'Neutral', 'Partially_Negative', 'Partially_Positive', 'Positive']
```

SCPrediction

```
#!pip install ktrain
```

```
#Import libraries

import numpy as np
import pandas as pd
import tensorflow as tf
import seaborn as sns
import ktrain
from ktrain import text
from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import re
```

```
import os
os.chdir(r'/content/drive/MyDrive/Colab Notebooks/CapstoneGL/new_model')
```

```
os.listdir()
```

```
['tf_model.h5', 'tf_model.preproc']
```

```
for file in os.listdir():
    print(f"{file}: {round(os.path.getsize(file)/1e+6,2)} MB")
```

```
tf_model.h5: 1314.47 MB
tf_model.preproc: 1.08 MB
```

```
#loading the model

predictor_load = ktrain.load_predictor("/content/drive/MyDrive/Colab Notebooks/CapstoneGL/new
```

```
predictor_load.get_classes()
```

```
['Negative', 'Neutral', 'Partially_Negative', 'Partially_Positive', 'Positive']
```

```
#sample dataset to test on

data = ['The public went berserk for “Psycho” in 1960, but critics were not as crazy about Al
        'movie was half good watchable but not great','this movie was horrible, the plot was
        'the fild is really sucked. there is not plot and acting was bad',
        'what a beautiful movie. great plot. acting was good. will see it again',]
```

```
predictor_load.predict(data)
```

```
['Partially_Negative',
 'Neutral',
 'Partially_Negative',
 'Negative',
 'Partially_Positive']
```

```
#new_data = ["this movie is shit, feels like i have wasted my time", "best movie i have seen"
new_data = ["The public went berserk for “Psycho” in 1960, but critics were not as crazy abou
            "this movie is shit, feels like i have wasted my time",
            "best movie i have seen",
            "i will rate this movie as average",
            "you are a kind man",
            "worst kind of movie ever created in MCU",
            "I have seen this movie"
            ]
new_prediction = predictor_load.predict(new_data, return_proba=True)
```

```
predictor_load.predict(new_data)
```

```
['Partially_Negative',
 'Negative',
 'Positive',
 'Neutral',
 'Partially_Positive',
 'Negative',
 'Negative']
```

```
#return_proba = True means it will give the prediction probabilty for each class

predictor_load.predict(new_data, return_proba=True)
```

```
array([[0.02619144, 0.2952079 , 0.653383  , 0.02408519, 0.00113248],
       [0.97995764, 0.00110503, 0.01233785, 0.00195646, 0.00464307],
       [0.02984951, 0.00467052, 0.00744052, 0.04353297, 0.9145065 ],
       [0.07645532, 0.53776073, 0.28896317, 0.09120732, 0.00561343],
       [0.04019133, 0.11195118, 0.04184788, 0.5200695 , 0.2859401 ],
       [0.9804819 , 0.00133124, 0.00577038, 0.00380445, 0.00861199],
       [0.35349956, 0.0769375 , 0.08074535, 0.23584509, 0.25297242]],
      dtype=float32)
```

```
Pred = new_data[5]
new_prediction = predictor_load.predict(new_data, return_proba=True)
for i, pred in enumerate(new_prediction):
  print(np.argmax(pred))
```

```
2
0
4
1
3
0
0
```

```
#new_data = ["this movie is shit, feels like i have wasted my time", "best movie i have seen"
new_data = ["The public went berserk for “Psycho” in 1960, but critics were not as crazy abou
            "this movie is shit, feels like i have wasted my time",
            "best movie i have seen",
            "i will rate this movie as average",
            "you are a kind man",
            "worst kind of movie ever created in MCU",
            "I have seen this movie"
            ]
new_prediction = predictor_load.predict(new_data, return_proba=True)
new_prediction
```

```
array([[0.02619144, 0.2952079 , 0.653383  , 0.02408519, 0.00113248],
       [0.97995764, 0.00110503, 0.01233785, 0.00195646, 0.00464307],
       [0.02984951, 0.00467052, 0.00744052, 0.04353297, 0.9145065 ],
       [0.07645532, 0.53776073, 0.28896317, 0.09120732, 0.00561343],
       [0.04019133, 0.11195118, 0.04184788, 0.5200695 , 0.2859401 ],
       [0.9804819 , 0.00133124, 0.00577038, 0.00380445, 0.00861199],
       [0.35349956, 0.0769375 , 0.08074535, 0.23584509, 0.25297242]],
      dtype=float32)
```

```
Pred = new_data[6]
new_prediction = predictor_load.predict(new_data, return_proba=True)
for i, pred in enumerate(new_prediction):
  print(np.argmax(pred))
```

```
2
0
4
1
3
0
0
```

```
for i, pred in enumerate(new_prediction):
    if np.argmax(pred) == 4:
        print(f"{new_data[i]} => \n {pred} => Positive")
    elif np.argmax(pred) == 3:
        print(f"{new_data[i]} => \n {pred} => Partially_Positive")
    elif np.argmax(pred) == 2:
        print(f"{new_data[i]} => \n {pred} => Neutral")
    elif np.argmax(pred) == 1:
        print(f"{new_data[i]} => \n {pred} => Partially_Negative")
    else:
        print(f"{new_data[i]} => \n {pred} => Negative")
```

```
The public went berserk for “Psycho” in 1960, but critics were not as crazy about Alfre
 [0.02619144 0.2952079  0.653383   0.02408519 0.00113248] => Neutral
this movie is shit, feels like i have wasted my time =>
 [0.97995764 0.00110503 0.01233785 0.00195646 0.00464307] => Negative
best movie i have seen =>
 [0.02984951 0.00467052 0.00744052 0.04353297 0.9145065 ] => Positive
i will rate this movie as average =>
 [0.07645532 0.53776073 0.28896317 0.09120732 0.00561343] => Partially_Negative
you are a kind man =>
 [0.04019133 0.11195118 0.04184788 0.5200695  0.2859401 ] => Partially_Positive
worst kind of movie ever created in MCU =>
 [0.9804819  0.00133124 0.00577038 0.00380445 0.00861199] => Negative
I have seen this movie =>
 [0.35349956 0.0769375  0.08074535 0.23584509 0.25297242] => Negative
```

On Yelp

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch
import requests
from bs4 import BeautifulSoup
import re
```

```
r = requests.get('https://www.yelp.com/biz/social-brew-cafe-pyrmont')
soup = BeautifulSoup(r.text, 'html.parser')
```

```
regex = re.compile('.*comment.*')
results = soup.find_all('p', {'class':regex})
reviews = [result.text for result in results]
```

```
reviews
```

```
['Still one of the favorite coffee shop in Sydney. Staffs have excellent knowledge about
 'I came to Social brew cafe for brunch while exploring the city and on my way to the ac
 "Ricotta hot cakes! These were so yummy. I ate them pretty fast and didn't share with a
 'Good coffee and toasts. Straight up and down - hits the spot with nothing mind blowing
 "Ron & Jo are on the go down under and Wow! \xa0We've found our breakfast place in Sydn
 "Great coffee and vibe. That's all \xa0you need. Crab was outstanding but not good fing
 "Great coffee and vibe. That's all \xa0you need. Crab was outstanding but not good fing
 "We came for brunch twice in our week-long visit to Sydney. Everything on the menu not
 'This is my fave brunch café in and around Sydney. Just love the ambience, food and dri
 "Delicious. The waitress was hot. The burger was juicy but messy that was the only thin
 'This cafe is one of the most popular cafes where we can enjoy eating nice breakfast in
```

```
yelpdf = pd.DataFrame(np.array(reviews), columns=['review'])
```

```
yelpdf['review'].iloc[0]
```

```
'Still one of the favorite coffee shop in Sydney. Staffs have excellent knowledge about
beans , flavor, brew skills. They make own pastries which are also tasty too.'
```

```
yelpdf.head()
```

```
                                         review
0         Still one of the favorite coffee shop in Sydne...
1        I came to Social brew cafe for brunch while ex...
2      Ricotta hot cakes! These were so yummy. I ate ...
3       Good coffee and toasts. Straight up and down -...
4  Ron & Jo are on the go down under and Wow!  We...
```

```
def sentiment_score(review):
    tokens = tokenizer.encode(review, return_tensors='pt')
    result = model(tokens)
    return int(torch.argmax(result.logits))+1
```

```
sentiment_score(yelpdf['review'].iloc[1])
```

```
5
```

```
yelpdf['sentiment'] = yelpdf['review'].apply(lambda x: sentiment_score(x[:512]))
```

```
yelpdf
```

	review	sentiment
0	Still one of the favorite coffee shop in Sydne...	5
1	I came to Social brew cafe for brunch while ex...	5
2	Ricotta hot cakes! These were so yummy. I ate ...	5
3	Good coffee and toasts. Straight up and down -...	5
4	Ron & Jo are on the go down under and Wow! We...	5
5	Great coffee and vibe. That's all you need. C...	5
6	Great coffee and vibe. That's all you need. C...	4
7	We came for brunch twice in our week-long visi...	4
8	This is my fave brunch café in and around Sydn...	5
9	Delicious. The waitress was hot. The burger wa...	4
10	This cafe is one of the most popular cafes whe...	5

```
Class = { 1: 'Negative',2: 'Partially_Negative',3: 'Neutral',4: 'Partially_Positive',5: 'Posi
```

```
yelpdf.sentiment =[Class[item] for item in yelpdf.sentiment]
```

```
yelpdf
```

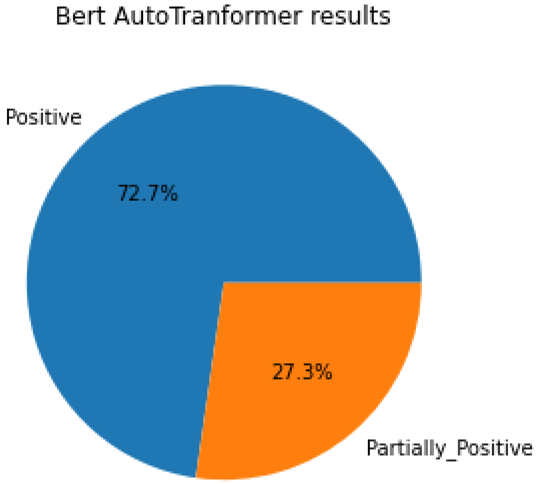
```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
     1          I came to Social brew cafe for brunch while ex...                    Positive
```

```
yelp_counts= yelpdf['sentiment'].value_counts()
```

```
plt.figure(figsize=(15,7))
plt.subplot(1,3,1)
plt.title("Bert AutoTranformer results")
plt.pie(yelp_counts.values, labels = yelp_counts.index, explode = None, autopct='%1.1f%%', sh
```

```
    ([<matplotlib.patches.Wedge at 0x7fae13522ed0>,
      <matplotlib.patches.Wedge at 0x7fae134fb350>],
     [Text(-0.7203468639465174, 0.8313244827396927, 'Positive'),
      Text(0.7203469417807291, -0.8313244152959488, 'Partially_Positive')],
     [Text(-0.3929164712435549, 0.45344971785801413, '72.7%'),
      Text(0.39291651369857944, -0.4534496810705174, '27.3%')])
```



IMDB _ What...if?

https://www.imdb.com/title/tt10168312/reviews?spoiler=hide&sort=reviewVolume&dir=desc&ratingF

```
r = requests.get('https://www.imdb.com/title/tt10168312/reviews')
soup = BeautifulSoup(r.text, 'html.parser')
regex = re.compile('.*text show-more__control.*')
results = soup.find_all('div', {'class':regex})
imdb_pipe_reviews = [result.text for result in results]
```

```
imdb_pipe_reviews
```

```
    ['The first episode was plain, with cringy catchphrases and lazy writing.',
     "WHAT IF...? Could have been an intriguing series. If the debut episode is anything to
     "Great story and great animation but wow did it ever feel rushed. I felt like it was ir
```

```
 "First episode so far was very bad. The pacing was off felt rushed. Terrible dialogue
 'The animation along with almost every line of dialogue in just the first episode has
 "Cant believe the drop in quality this had compared to marvel's other tv shows. I can
 "I had to ff numerous times during my watch to get through the first episode... But I
 "A very straight forward episode of rushing things to set those pieces right into the
 "This is meant for kids. It's pure entertainment. (Although not that interesting in my
 'I thought it will be different plot but its almost the same story but a minor change.
 'I was very excited about the What If series, on a level similar to LOKI.The first epi
 "Look at the comics. What if Spiderman had kept his 6 arms. What if hulk killed wolver
 "First episode is nothing special, was expecting more unusual What If.. but all we got
 'From the animation style and tone, it\'s quite clearly aimed for children 5-13. It\'s
 "It was horrible. The writing was all over the place. The thing that bothered me the m
 "Nobody asked for this show. Animation is sub par and writing is worse. It's the same
 "I know it's probably too early to write that after the first episode. The plot looks
 'The cartoon is like a Saturday morning cartoon people use to get up for.... Except wo
 'Is this the best they can do for a "What If...?" show? I hope this gets better but ne
 'This is for first episode only so far, I will be coming back to change this as the se
 "First episode has some cool concepts like the Hydra Stomper but has a very high chees
 'Poor writing ruined the first episode. It was awful. It had a promising first half. I
 "Don't like the animation, especially there mouths. It was overhyped for me, guess we
 'IVE ONLY WATCHED THE FIRST EPISODE: I was super excited for this but i fell asleep ha
 'Stay well clear. Just retelling the same stories with one or two small changes, no im
```

```
imdb_pipe_reviews_df = pd.DataFrame(np.array(imdb_pipe_reviews), columns=['review'])
```

```
imdb_pipe_reviews_df.head()
```

```
                                          review
0      The first episode was plain, with cringy catch...
1   WHAT IF...? Could have been an intriguing seri...
2     Great story and great animation but wow did it...
3     First episode so far was very bad. The pacing ...
4      The animation along with almost every line of ...
```

```
sentiment_score(imdb_pipe_reviews_df['review'].iloc[1])
```

```
2
```

```
imdb_pipe_reviews_df['sentiment'] = imdb_pipe_reviews_df['review'].apply(lambda x: sentiment_
```

```
imdb_pipe_reviews_df
```

	review	sentiment
0	The first episode was plain, with cringy catch...	2
1	WHAT IF...? Could have been an intriguing seri...	2
2	Great story and great animation but wow did it...	4
3	First episode so far was very bad. The pacing ...	2
4	The animation along with almost every line of ...	2
5	Cant believe the drop in quality this had comp...	2
6	I had to ff numerous times during my watch to ...	2
7	A very straight forward episode of rushing thi...	3
8	This is meant for kids. It's pure entertainmen...	3
9	I thought it will be different plot but its al...	3
10	I was very excited about the What If series, o...	2
11	Look at the comics. What if Spiderman had kept...	1
12	First episode is nothing special, was expectin...	3
13	From the animation style and tone, it's quite ...	3
14	It was horrible. The writing was all over the ...	1
15	Nobody asked for this show. Animation is sub p...	1
16	I know it's probably too early to write that a...	3
17	The cartoon is like a Saturday morning cartoon...	2
18	Is this the best they can do for a "What If......	3
19	This is for first episode only so far, I will ...	3
20	First episode has some cool concepts like the ...	3
21	Poor writing ruined the first episode. It was ...	1
22	Don't like the animation, especially there mou...	2

```
imdb_pipe_reviews_df.sentiment =[Class[item] for item in imdb_pipe_reviews_df.sentiment]
```

24	Stay well clear. Just retelling the same stori...	2

```
imdb_pipe_reviews_df
```

	review	sentiment
0	The first episode was plain, with cringy catch...	Partially_Negative
1	WHAT IF...? Could have been an intriguing seri...	Partially_Negative
2	Great story and great animation but wow did it...	Partially_Positive
3	First episode so far was very bad. The pacing ...	Partially_Negative
4	The animation along with almost every line of ...	Partially_Negative
5	Cant believe the drop in quality this had comp...	Partially_Negative
6	I had to ff numerous times during my watch to ...	Partially_Negative
7	A very straight forward episode of rushing thi...	Neutral
8	This is meant for kids. It's pure entertainmen...	Neutral
9	I thought it will be different plot but its al...	Neutral
10	I was very excited about the What If series, o...	Partially_Negative
11	Look at the comics. What if Spiderman had kept...	Negative
12	First episode is nothing special, was expectin...	Neutral
13	From the animation style and tone, it's quite ...	Neutral
14	It was horrible. The writing was all over the ...	Negative
15	Nobody asked for this show. Animation is sub p...	Negative
16	I know it's probably too early to write that a...	Neutral
17	The cartoon is like a Saturday morning cartoon...	Partially_Negative
18	Is this the best they can do for a "What If......	Neutral
19	This is for first episode only so far, I will ...	Neutral
20	First episode has some cool concepts like the ...	Neutral

```
imdb_pipe_reviews_df['review'].iloc[10]
```

```
'I was very excited about the What If series, on a level similar to LOKI.The first epis
ode has Agent Carter becoming Captain Carter, and the episode feels as if someone was d
escribing a movie to me in a rushed manner. Obviously there are references to the origi
nal MCU for YouTubers to microanalyze, but it almost comes off as a simple kids show wi
th characters fighting and blowing stuff up.The show could have had a much better stor
y, because after all, we are dealing with the multiverse here. Instead Captain Carter s
```

```
imdb_pipe_reviews_df.iloc[10]
```

```
review       I was very excited about the What If series, o...
sentiment                                   Partially_Negative
Name: 10, dtype: object
```

```
imdb_pipe_reviews_df['review'].iloc[21]
```

```
'Poor writing ruined the first episode. It was awful. It had a promising first half. It
all went downhill at the later half. The corny dialogues, uneven pacing and bad plot.At
first I was sceptical of the bad reviews. The first few minutes were not perfect but it
looked promising. Animation wa            lth            sync was a little out of place) and
```

+ Code + Text

```
imdb_pipe_reviews_df.iloc[21]
```

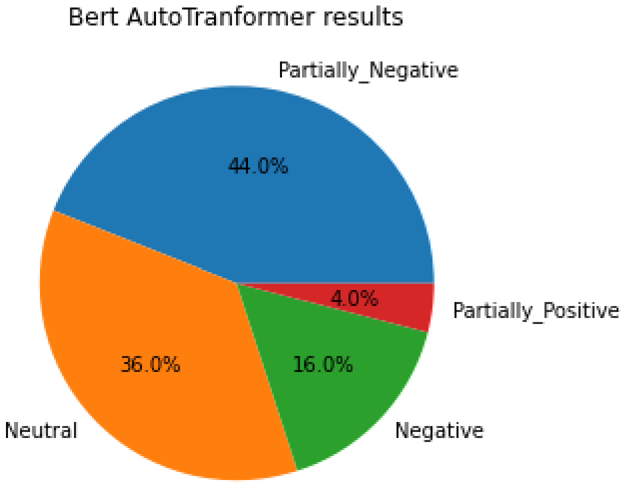
```
review       Poor writing ruined the first episode. It was ...
sentiment                                             Negative
Name: 21, dtype: object
```

```
imdb_pipe_Sentiment_Count = imdb_pipe_reviews_df['sentiment'].value_counts()
```

```
imdb_pipe_Sentiment_Count
```

```
Partially_Negative    11
Neutral                9
Negative               4
Partially_Positive     1
Name: sentiment, dtype: int64
```

```
plt.figure(figsize=(15,7))
plt.subplot(1,3,1)
plt.title("Bert AutoTranformer results")
plt.pie(imdb_pipe_Sentiment_Count.values, labels = imdb_pipe_Sentiment_Count.index, explode =
     ([<matplotlib.patches.Wedge at 0x7fae10b32610>,
       <matplotlib.patches.Wedge at 0x7fae10aa9790>,
       <matplotlib.patches.Wedge at 0x7fae10ab1050>,
       <matplotlib.patches.Wedge at 0x7fae10ab1950>],
      [Text(0.2061194541375137, 1.080515974257694, 'Partially_Negative'),
       Text(-0.8018654676031948, -0.753001840545898, 'Neutral'),
       Text(0.8018655381043113, -0.7530017654698318, 'Negative'),
       Text(1.0913261782871635, -0.13786650276675014, 'Partially_Positive')],
      [Text(0.11242879316591656, 0.5893723495951058, '44.0%'),
       Text(-0.4373811641471971, -0.4107282766613989, '36.0%'),
       Text(0.4373812026023515, -0.4107282357108173, '16.0%'),
       Text(0.595268824520271, -0.07519991060004552, '4.0%')])
```



✓ 0s completed at 6:55 AM