```
!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 1.8 MB/s eta 0:10:36tcmalloc: large al
     |█████████████████               | 1055.7 MB 1.5 MB/s eta 0:10:27tcmalloc: large a
     |█████████████████████▏          | 1336.2 MB 1.4 MB/s eta 0:07:27tcmalloc: large a
     |███████████████████████████▎    | 1691.1 MB 1.2 MB/s eta 0:04:04tcmalloc: large a
     |████████████████████████████████| 1982.2 MB 1.3 MB/s eta 0:00:01tcmalloc: large al
tcmalloc: large alloc 2477727744 bytes == 0x5580fd3ae000 @  0x7f6211870615 0x557f946240
     |████████████████████████████████| 1982.2 MB 1.1 kB/s
Collecting torchvision==0.9.1+cu111
  Downloading https://download.pytorch.org/whl/cu111/torchvision-0.9.1%2Bcu111-cp37-cp3
     |████████████████████████████████| 17.6 MB 44.0 MB/s
Collecting torchaudio===0.8.1
  Downloading torchaudio-0.8.1-cp37-cp37m-manylinux1_x86_64.whl (1.9 MB)
     |████████████████████████████████| 1.9 MB 4.9 MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from tor
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packag
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 5.1 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)
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: filelock in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (fro
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 32.0 MB/s
Collecting pyyaml>=5.1
  Downloading PyYAML-5.4.1-cp37-cp37m-manylinux1_x86_64.whl (636 kB)
     |████████████████████████████████| 636 kB 52.7 MB/s
Collecting sacremoses
  Downloading sacremoses-0.0.45-py3-none-any.whl (895 kB)
     |████████████████████████████████| 895 kB 46.1 MB/s
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (fro
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: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
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 (
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-
```

```
!pip install ktrain
```

```
Collecting ktrain
  Downloading ktrain-0.27.2.tar.gz (25.3 MB)
     |████████████████████████████████| 25.3 MB 107 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 43.6 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 33.0 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 64.9 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 42.1 MB/s
Collecting sentencepiece
  Downloading sentencepiece-0.1.96-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_
     |████████████████████████████████| 1.2 MB 47.9 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 37.1 MB/s
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-2.2.0-py3-none-any.whl (12 kB)
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: Keras>=2.2.4 in /usr/local/lib/python3.7/dist-packages
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)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-pac
```

```
import ktrain
from ktrain import text
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
0	0	final fantasy the spirits within 2001	evelyn c leeper	capsule this very dark scifi fantasy is magnif...
1	1	sexy beast 2000	mark r leeper	roger ebert asks in his review of sexy beast w...
2	2	final fantasy the spirits within 2001	robin clifford	aliens beings have taken over the earth the gr...

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

```
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...
3	jurassic park iii 2001	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
the enemy if this film had been done in liveaction the scenes more spectacular than tho
```

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, 19.9kB/s]

Downloading: 100%                                    872k/872k [00:00<00:00, 728kB/s]

Downloading: 100%                                    112/112 [00:00<00:00, 2.08kB/s]

Downloading: 100%                                    39.0/39.0 [00:00<00:00, 847B/s]

Downloading: 100%                                    669M/669M [00:22<00:00, 31.4MB/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)
```

```
================================================================================
Layer (type:depth-idx)                                        Param #
================================================================================
BertForSequenceClassification                                 --
├─BertModel: 1-1                                              --
│    └─BertEmbeddings: 2-1                                    --
│    │    └─Embedding: 3-1                                    81,315,072
│    │    └─Embedding: 3-2                                    393,216
│    │    └─Embedding: 3-3                                    1,536
│    │    └─LayerNorm: 3-4                                    1,536
│    │    └─Dropout: 3-5                                      --
│    └─BertEncoder: 2-2                                       --
│    │    └─ModuleList: 3-6                                   --
│    │    │    └─BertLayer: 4-1                               --
│    │    │    │    └─BertAttention: 5-1                      --
│    │    │    │    │    └─BertSelfAttention: 6-1             --
│    │    │    │    │    │    └─Linear: 7-1                   590,592
│    │    │    │    │    │    └─Linear: 7-2            590,592
│    │    │    │    │    │    └─Linear: 7-3            590,592
│    │    │    │    │    │    └─Dropout: 7-4           --
│    │    │    │    │    └─BertSelfOutput: 6-2         --
│    │    │    │    │    │    └─Linear: 7-5            590,592
│    │    │    │    │    │    └─LayerNorm: 7-6         1,536
│    │    │    │    │    │    └─Dropout: 7-7           --
│    │    │    │    └─BertIntermediate: 5-2            --
│    │    │    │    │    └─Linear: 6-3                 2,362,368
│    │    │    │    └─BertOutput: 5-3                  --
│    │    │    │    │    └─Linear: 6-4                 2,360,064
│    │    │    │    │    └─LayerNorm: 6-5              1,536
│    │    │    │    │    └─Dropout: 6-6                --
│    │    │    └─BertLayer: 4-2                        --
│    │    │    │    └─BertAttention: 5-4               --
│    │    │    │    │    └─BertSelfAttention: 6-7      --
│    │    │    │    │    │    └─Linear: 7-8            590,592
│    │    │    │    │    │    └─Linear: 7-9            590,592
│    │    │    │    │    │    └─Linear: 7-10           590,592
│    │    │    │    │    │    └─Dropout: 7-11          --
│    │    │    │    │    └─BertSelfOutput: 6-8         --
│    │    │    │    │    │    └─Linear: 7-12           590,592
│    │    │    │    │    │    └─LayerNorm: 7-13        1,536
│    │    │    │    │    │    └─Dropout: 7-14          --
│    │    │    │    └─BertIntermediate: 5-5            --
│    │    │    │    │    └─Linear: 6-9                 2,362,368
│    │    │    │    └─BertOutput: 5-6                  --
│    │    │    │    │    └─Linear: 6-10                2,360,064
│    │    │    │    │    └─LayerNorm: 6-11             1,536
│    │    │    │    │    └─Dropout: 6-12               --
│    │    │    └─BertLayer: 4-3                        --
│    │    │    │    └─BertAttention: 5-7               --
│    │    │    │    │    └─BertSelfAttention: 6-13     --
│    │    │    │    │    │    └─Linear: 7-15           590,592
│    │    │    │    │    │    └─Linear: 7-16           590,592
│    │    │    │    │    │    └─Linear: 7-17           590,592
│    │    │    │    │    │    └─Dropout: 7-18          --
│    │    │    │    │    └─BertSelfOutput: 6-14        --
│    │    │    │    │    │    └─Linear: 7-19           590,592
│    │    │    │    │    │    └─LayerNorm: 7-20        1,536
│    │    │    │    │    │    └─Dropout: 7-21          --
│    │    │    │    └─BertIntermediate: 5-8            --
```

```
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(
            (self): BertSelfAttention(
```

```
intertr
```

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...
3	jurassic park iii 2001	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
```

```
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)))
```

```
df.head()
```

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

```
df.to_csv(r'/content/drive/MyDrive/Colab Notebooks/CapstoneGL/imdbautomodelgb08152021type2.cs
```

```
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
```

```
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)))
```

```
df.head()
```

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

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

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

```
df.to_csv(r'/content/drive/MyDrive/Colab Notebooks/CapstoneGL/imdbgb08162021bertsentimenttype
```

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

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

```
!pip install ktrain
```

model init

```
#Import libraries

import numpy as np
import pandas as pd
import tensorflow as tf
import seaborn as sns
import matplotlib.pyplot as plt
#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.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
15712	15712	at the cineplex on sunday the sexes were segre...	Neutral
16503	16503	release date april starring scott bakula corbi...	Neutral
8942	8942	i have lived in suburbia for most of my life m...	Positive
19900	19900	pretty much everyone knows the story about how...	Partially_Positive
18422	18422	kiss or kill starts with a shocking immolation...	Neutral

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

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

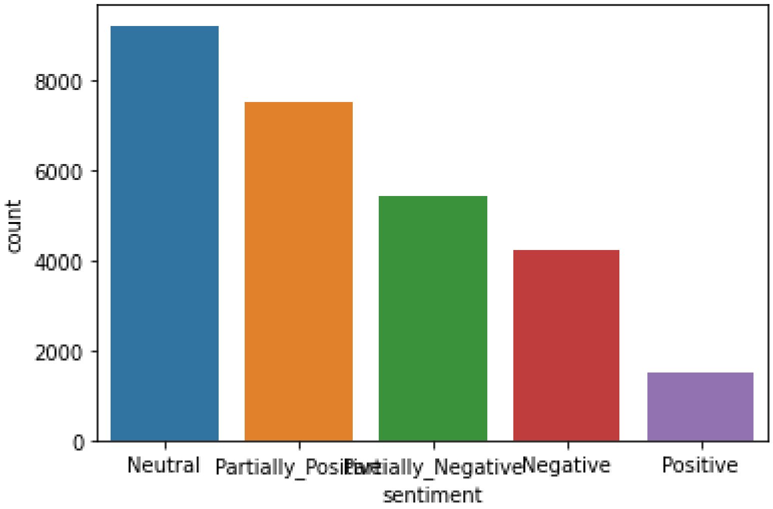
	clean_reviews	sentiment
22784	heart like a wheel isnt a great piece of b mov...	Partially_Positive
8578	after a two year absence largely spent cutting...	Partially_Positive
8587	the red violin aka le violon rougerated r runn...	Neutral
3699	beach red director cornel wilde screenwriters ...	Neutral
20487	con air directed by simon west written by scot...	Partially_Positive

```
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 0x7fd049aa2d10>
```



```
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
```

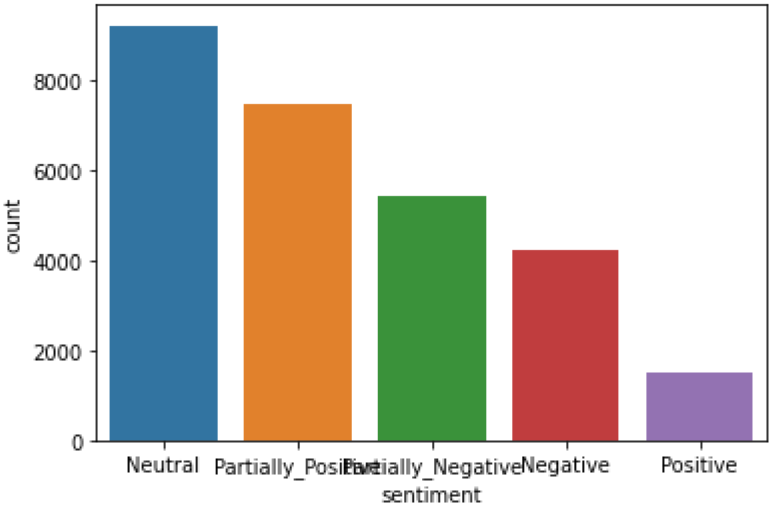
```
Partially_Positive    7480
Partially_Negative    5423
Negative              4212
Positive              1497
Name: sentiment, dtype: int64
```

```
s_counts.sum()
```

```
27841
```

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd045169b50>
```



```
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 0x7fd045049a10>,
       <matplotlib.patches.Wedge at 0x7fd0450541d0>,
       <matplotlib.patches.Wedge at 0x7fd045054a50>,
       <matplotlib.patches.Wedge at 0x7fd04505e390>,
       <matplotlib.patches.Wedge at 0x7fd04505eed0>],
      [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%'),
```

```
s_counts.sum()
```

```
27841
```

```
TRAIN_SIZE = 22000
TEST_SIZE = 5840

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

TRAIN_SIZE = 21000 TEST_SIZE = 6841

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

```
data_train.head()
```

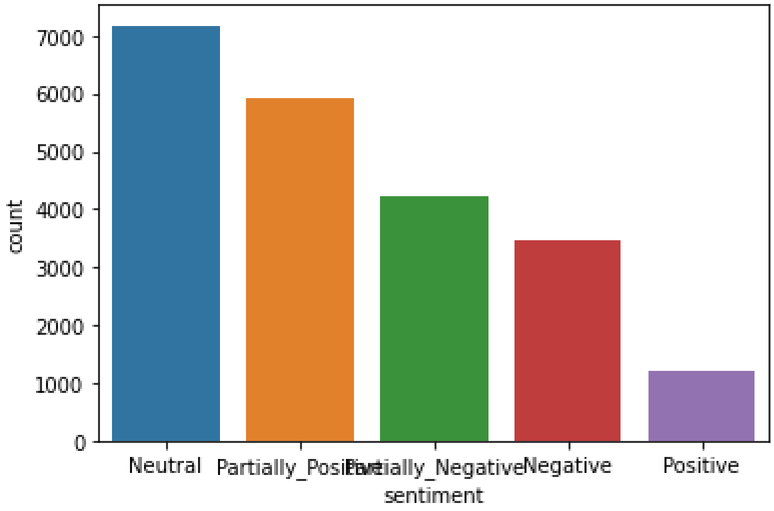
	clean_reviews	sentiment
0	capsule this very dark scifi fantasy is magnif...	Neutral
1	roger ebert asks in his review of sexy beast w...	Neutral
2	aliens beings have taken over the earth the gr...	Partially_Positive
3	susan grangers review of jurassic park iii uni...	Partially_Positive
4	susan grangers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

```
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 0x7fd045017350>
```



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

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

```
data_test.head()
```

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

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

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

```
#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,
                                                                   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...
1060/3667 [=======>......................] - ETA: 28:56:47 - loss: 1.4609 - accuracy: 0
```

```
learner.validate(class_names=preproc.get_classes())
```

```
predictor = ktrain.get_predictor(learner.model, preproc)
predictor.save("/content/drive/MyDrive/Colab Notebooks/CapstoneGL/modelv2_210822")
```

```
predictor.explain('Jesus Christ is the central figure of Christianity.')
```

```
predictor.explain('Jesus Christ the fild is really sucked. there is not plot and acting was b
```

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

```
#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)
```

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

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

```
#classes available

predictor_load.get_classes()
```

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/modelv2_210822')
```

```
os.listdir()
```

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

```
#loading the model

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

```
predictor_load.get_classes()
```

```
#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)
```

```
#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)
```

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

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

```
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))
```

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

```
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))
```

```
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")
```

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
```

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

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

```
yelpdf.head()
```

```
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])
```

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

```
yelpdf
```

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

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

```
yelpdf
```

```
reviews[0]
```

```
predictor_load.predict(reviews)
```

```
predictedresult=predictor_load.predict(reviews)
```

```
predictedresult = pd.DataFrame(predictedresult,columns=['PredictedSentiment'])
```

```
predictedresult
```

```
predictedresult.value_counts()
```

```
predictedbymodel_counts= predictedresult['PredictedSentiment'].value_counts()
```

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

```
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
```

```
plt.figure(figsize=(15,7))
plt.subplot(1,3,1)
plt.title("Results of predicted by model")
plt.pie(predictedbymodel_counts.values, labels = predictedbymodel_counts.index, explode = Non
```

IMDB _ The Suicide Squad-2021

```
r = requests.get('https://www.imdb.com/title/tt6334354/reviews')
soup = BeautifulSoup(r.text, 'html.parser')
regex = re.compile('.*text show-more__control.*')
results = soup.find_all('div', {'class':regex})
imdb_pipe_reviews2 = [result.text for result in results]
```

```
imdb_pipe_reviews2
```

```
imdb_pipe_reviews_df2 = pd.DataFrame(np.array(imdb_pipe_reviews2), columns=['review'])
```

```
predicted_TSS2021=predictor_load.predict(imdb_pipe_reviews)
```

```
predicted_TSS2021_Sentiment = pd.DataFrame(predicted_TSS2021,columns=['PredictedSentiment'])
```

```
predicted_TSS2021_Sentiment
```

```
predicted_TSS2021_Sentiment.value_counts()
```

```
Sentiment_count=predicted_TSS2021_Sentiment.value_counts()
```

```
plt.figure(figsize=(15,7))
plt.subplot(1,3,1)
plt.title("Sentiment predicted by model")
plt.pie(Sentiment_count.values, labels = Sentiment_count.index, explode = None, autopct='%1.1
```

Prediction justification

Positive

```
n =0
print(imdb_pipe_reviews2[n])
print(' \n Predicted Sentiment: ',predicted_TSS2021_Sentiment['PredictedSentiment'].iloc[n])
```

Partially_Positive

```
n = 3
print(imdb_pipe_reviews2[n])
print(' \n Predicted Sentiment: ',predicted_TSS2021_Sentiment['PredictedSentiment'].iloc[n])
```

Neutral

```
n = 2
print(imdb_pipe_reviews2[n])
print(' \n Predicted Sentiment: ',predicted_TSS2021_Sentiment['PredictedSentiment'].iloc[n])
```

Partially_Negative

```
n = 5
print(imdb_pipe_reviews2[n])
print(' \n Predicted Sentiment: ',predicted_TSS2021_Sentiment['PredictedSentiment'].iloc[n])
```

Negative

```
n = 7
print(imdb_pipe_reviews2[n])
print(' \n Predicted Sentiment: ',predicted_TSS2021_Sentiment['PredictedSentiment'].iloc[n])
```

End

● ×