In [ ]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
```

In [ ]:
```
tf.test.gpu_device_name()
```

Out[ ]:
```
'/device:GPU:0'
```

Grader Function 1

In [ ]:
```
def grader_tf_version():
    assert((tf.__version__)>'2')
    return True
grader_tf_version()
```

Out[ ]:
```
True
```

Pre Processing

In [ ]:
```
!unzip '/content/drive/MyDrive/NLP/Reviews.csv-20211125T113054Z-001.zip'
```

```
Archive:  /content/drive/MyDrive/NLP/Reviews.csv-20211125T113054Z-001.zip
  inflating: Reviews.csv
```

In [ ]:
```
reviews = pd.read_csv('/content/Reviews.csv')
reviews.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
 #   Column                  Non-Null Count   Dtype 
---  ------                  --------------   ----- 
 0   Id                      568454 non-null  int64 
 1   ProductId               568454 non-null  object
 2   UserId                  568454 non-null  object
 3   ProfileName             568438 non-null  object
 4   HelpfulnessNumerator    568454 non-null  int64 
 5   HelpfulnessDenominator  568454 non-null  int64 
 6   Score                   568454 non-null  int64 
 7   Time                    568454 non-null  int64 
 8   Summary                 568427 non-null  object
 9   Text                    568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

In [ ]:
```
# Extracting only Text and Score Columns and Dropping the NAN values
reviews = reviews[['Text','Score']]
reviews.dropna(inplace=True)
```

In [ ]:
```
#if score> 3, set score = 1
#if score<=2, set score = 0
#if score == 3, remove the rows.
reviews.loc[reviews['Score']<=2,'Score'] = 0
reviews.loc[reviews['Score']>3,'Score']  = 1
reviews.drop(reviews[reviews['Score']==3].index,inplace=True)
```

Grader Function 2

In [ ]:
```
def grader_reviews():
    temp_shape = (reviews.shape == (525814, 2)) and (reviews.Score.value
_counts()[1]==443777)
    assert(temp_shape == True)
    return True
grader_reviews()
```

Out[ ]:
```
True
```

In [ ]:
```
def get_wordlen(x):
    return len(x.split())
reviews['len'] = reviews.Text.apply(get_wordlen)
reviews = reviews[reviews.len<50]
reviews = reviews.sample(n=100000, random_state=30)
```

In [ ]:
```
#remove HTML from the Text column and save in the Text column only
import re as re
def remove_tags(string):
    result = re.sub('<.*?>','',string)
    return result
reviews['Text']=reviews['Text'].apply(lambda cw : remove_tags(cw))
```

In [ ]:
```
reviews.head(5)
```

Out[ ]:

	Text	Score	len
64117	The tea was of great quality and it tasted lik...	1	30
418112	My cat loves this. The pellets are nice and s...	1	31
357829	Great product. Does not completely get rid of ...	1	41
175872	This gum is my favorite! I would advise every...	1	27
178716	I also found out about this product because of...	1	22

In [ ]:
```
reviews.to_csv('/content/drive/MyDrive/NLP/preprocessed.csv', index=Fals
e)
```