In [1]:

```
import warnings
warnings.filterwarnings("ignore")
import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer
import re  # Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle
from tqdm import tqdm
import os
from plotly import plotly
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from collections import Counter
```

In [2]:

```
df=pd.read_csv('preprocessed_data.csv')
```

In [3]:

```
df.head(2)
```

Out[3]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_s
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL

2 rows × 26 columns

In [8]:

```
import warnings
warnings.filterwarnings("ignore")
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
```

Out[8]:

```
True
```

In [26]:

```
sid = SentimentIntensityAnalyzer()
for_sentiment = 'a person is a person no matter how small dr seuss i teach the smallest
students with the biggest enthusiasm for learning my students learn in many different w
ays using all of our senses and multiple intelligences i use a wide range of techniques
to help all my students succeed students in my class come from a variety of different b
ackgrounds which makes for wonderful sharing of experiences and cultures including nati
ve americans our school is a caring community of successful learners which can be seen
 through collaborative student project based learning in and out of the classroom kinde
rgarteners'
ss = sid.polarity_scores(for_sentiment)
for k in ss:
    print('{0}: {1}, '.format(k, ss[k]), end='')
```

```
neg: 0.02, neu: 0.744, pos: 0.236, compound: 0.9712,
```

In [27]:

```
print(ss)
```

```
{'neg': 0.02, 'neu': 0.744, 'pos': 0.236, 'compound': 0.9712}
```

In [25]:

```
print(ss[k])
```

```
0.9712
```

In [29]:

```
ss['pos']
```

Out[29]:

```
0.236
```

In [30]:

```
ss.get('compound')
```

Out[30]:

```
0.9712
```

In [31]:

```
a='very disappointed'
b=sid.polarity_scores(a)
print(b)
```

```
{'neg': 0.772, 'neu': 0.228, 'pos': 0.0, 'compound': -0.5256}
```

In [32]:

```
sent_score=[]
for sent in tqdm(df['preprocessed_essays'].values):
    ss = sid.polarity_scores(sent)
    score=ss.get('compound')
    sent_score.append(score)
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 109248/109248 [04:47
<00:00, 379.52it/s]
```

In [39]:

```
sent_score[5]
```

Out[39]:

```
0.9802
```

In [40]:

```
df['sentimental_score']=sent_score
```

In [42]:

```
df.head(3)
```

Out[42]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_s
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL
2	21895	p182444	3465aaf82da834c0582ebd0ef8040ca0	Ms.	AZ

3 rows × 27 columns

In [43]:

```
#word count in each preprocessed essays text
essay_count=[]
for text in tqdm(df['preprocessed_essays'].values):
    c=len(text.split()) #https://www.geeksforgeeks.org/python-program-to-count-words-in-a-sentence/
    essay_count.append(c)
```

```
100%|██████████████████████████████████████████████████████████████████████| 109248/109248 [00:01<00:00, 65372.26it/s]
```

In [46]:

```
essay_count[9999]
```

Out[46]:

```
232
```

In [47]:

```
df['preprocessed_essay_word_count']=essay_count
```

In [48]:

```
df.head(2)
```

Out[48]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_s
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL

2 rows × 28 columns

In [68]:

```
#word count in each preprocessed titles sentence
import re
title_count=[]
for title in tqdm(df['preprocessed_titles'].values):
    #title=re.sub('[^A-Za-z0-9]+', '', title)
    d=len(str(title).split())
    title_count.append(d)
```

```
100%|██████████████████████████████████████████████████████████████████
██████████████████████████████████████████| 109248/109248 [00:00<0
0:00, 560047.08it/s]
```

In [71]:

```
title_count[3661]
```

Out[71]:

4

In [72]:

```
df['preprocessed_title_word_count']=title_count
```

In [73]:

```
df.head(2)
```

Out[73]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_s
0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	Mrs.	IN
1	140945	p258326	897464ce9ddc600bced1151f324dd63a	Mr.	FL

2 rows × 29 columns

In [75]:

```
export_csv = df.to_csv (r'D:\PGS\Applied AI course\Assignments\Mandatory\DP_senti.csv',
index = None, header=True)
```