In [1]:
```
import re
import numpy as np
import pandas as pd
import seaborn as sns
from wordcloud import WordCloud
from textblob import TextBlob
import matplotlib.pyplot as plt
from nltk.stem import WordNetLemmatizer
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import BernoulliNB
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix, classification_report
```

In [2]:
```
df = pd.read_csv("C:/Users/jsaty/OneDrive/Desktop/NLP Lab/adipurush_tweets.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	Date Created	Number of Likes	Tweets
0	2023-06-30 09:21:00+00:00	0	Movie is so good
1	2023-06-30 09:20:57+00:00	0	Now Playing!! Book Your Ticket Now!! 🎥 🍿 🎟 \n@go...
2	2023-06-30 09:20:22+00:00	0	@ponilemova #Adipurush
3	2023-06-30 09:20:00+00:00	3	Adipurush VS 72 Hoorain VS The Kerala Story Co...
4	2023-06-30 09:15:22+00:00	3	ST: #Adipurush https://t.co/lsGKcgQuKL

In [4]:
```
df.columns
```

Out[4]:
```
Index(['Date Created', 'Number of Likes', 'Tweets'], dtype='object')
```

In [5]:
```
print('length of data is', len(df))
```

```
length of data is 10001
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10001 entries, 0 to 10000
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Date Created     10001 non-null  object
 1   Number of Likes  10001 non-null  int64 
 2   Tweets           10001 non-null  object
dtypes: int64(1), object(2)
memory usage: 234.5+ KB
```

In [7]:

```
np.sum(df.isnull().any(axis=1))
```

Out[7]:

```
0
```

In [8]:

```
df['Tweets'] = df['Tweets'].str.lower()
```

In [9]:

```
df['Tweets'].head()
```

Out[9]:

```
0                                     movie is so good
1    now playing!! book your ticket now!! 🎥 🍿 🎟\n@go...
2                               @ponilemova #adipurush
3    adipurush vs 72 hoorain vs the kerala story co...
4               st: #adipurush https://t.co/lsgkcgqukl
Name: Tweets, dtype: object
```

In [10]:

```
stopwordlist = ['a', 'about', 'above', 'after', 'again', 'ain', 'all', 'am', 'an',
             'and','any','are', 'as', 'at', 'be', 'because', 'been', 'before',
             'being', 'below', 'between','both', 'by', 'can', 'd', 'did', 'do',
             'does', 'doing', 'down', 'during', 'each','few', 'for', 'from',
             'further', 'had', 'has', 'have', 'having', 'he', 'her', 'here',
             'hers', 'herself', 'him', 'himself', 'his', 'how', 'i', 'if', 'in',
             'into','is', 'it', 'its', 'itself', 'just', 'll', 'm', 'ma',
             'me', 'more', 'most','my', 'myself', 'now', 'o', 'of', 'on', 'once',
             'only', 'or', 'other', 'our', 'ours','ourselves', 'out', 'own', 're','
             't', 'than', 'that', "thatll", 'the', 'their', 'theirs', 'them',
             'themselves', 'then', 'there', 'these', 'they', 'this', 'those',
             'through', 'to', 'too','under', 'until', 'up', 've', 'very', 'was',
             'we', 'were', 'what', 'when', 'where','which','while', 'who', 'whom',
             'why', 'will', 'with', 'won', 'y', 'you', "youd","youll", "youre",
             "youve", 'your', 'yours', 'yourself', 'yourselves']
```

In [11]:

```
STOPWORDS = set(stopwordlist)
def cleaning_stopwords(text):
    return " ".join([word for word in str(text).split() if word not in STOPWORDS])
df['Tweets'] = df['Tweets'].apply(lambda Tweets: cleaning_stopwords(Tweets))
df['Tweets'].head()
```

Out[11]:

```
0                                           movie good
1    playing!! book ticket now!! 🎥 🍿 🎟 @gopalancinem...
2                               @ponilemova #adipurush
3    adipurush vs 72 hoorain vs kerala story contro...
4               st: #adipurush https://t.co/lsgkcgqukl
Name: Tweets, dtype: object
```

In [12]:

```
import string
english_punctuations = string.punctuation
punctuations_list = english_punctuations
def cleaning_punctuations(Tweets):
    translator = str.maketrans('', '', punctuations_list)
    return Tweets.translate(translator)
df['Tweets']= df['Tweets'].apply(lambda x: cleaning_punctuations(x))
df['Tweets'].tail()
```

Out[12]:

```
9996     adipurush 1st week ww box office collections ఆ...
9997     godmorningfriday वास्तव में adipurush यानि सबस...
9998     let empowering lyrics shivoham elevate spirit ...
9999     comes choosing service product its always bene...
10000    film ramayana greatest epic could not even ear...
Name: Tweets, dtype: object
```

In [13]:

```
def cleaning_repeating_char(Tweets):
    return re.sub(r'(.)1+', r'1',Tweets)
df['Tweets'] = df['Tweets'].apply(lambda x: cleaning_repeating_char(x))
df['Tweets'].tail()
```

Out[13]:

```
9996     adipurush1st week ww box office collections ఆద...
9997     godmorningfriday वास्तव में adipurush यानि सबस...
9998     let empowering lyrics shivoham elevate spirit ...
9999     comes choosing service product its always bene...
10000    film ramayana greatest epic could not even ear...
Name: Tweets, dtype: object
```

In [14]:

```
def cleaning_URLs(data):
    return re.sub('((www.[^s]+)|(https?:// [^s]+))',' ',data)
df['Tweets'] = df['Tweets'].apply(lambda x: cleaning_URLs(x))
df['Tweets'].tail()
```

Out[14]:

```
9996     adipurush1st week ww box office collections ఆద...
9997     godmorningfriday वास्तव में adipurush यानि सबस...
9998     let empowering lyrics shivoham elevate spirit ...
9999     comes choosing service product its always bene...
10000    film ramayana greatest epic could not even ear...
Name: Tweets, dtype: object
```

In [15]:

```
def cleaning_numbers(data):
    return re.sub('[0-9]+', '', data)
df['Tweets'] = df['Tweets'].apply(lambda x: cleaning_numbers(x))
df['Tweets'].tail()
```

Out[15]:

```
9996     adipurushst week ww box office collections ఆది...
9997     godmorningfriday वास्तव में adipurush यानि सबस...
9998     let empowering lyrics shivoham elevate spirit ...
9999     comes choosing service product its always bene...
10000    film ramayana greatest epic could not even ear...
Name: Tweets, dtype: object
```

In [16]:

```
tweets = df['Tweets']
```

In [22]:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import nltk
nltk.download('vader_lexicon')

def classify_sentiment(tweet):
    sid = SentimentIntensityAnalyzer()
    sentiment_scores = sid.polarity_scores(tweet)


    if sentiment_scores['compound'] >= 0.01:
        return 'Positive'
    elif sentiment_scores['compound'] <= -0.01:
        return 'Negative'
    else:
        return 'Neutral'
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\jsaty\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

In [23]:

```
tweets = tweets.astype(str)
```

In [24]:

```
df['sentiment'] = tweets.apply(lambda x: classify_sentiment(x))
```

In [20]:

```
tweets = tweets.fillna('')
```

In [21]:

```
df.to_csv('classified_tweets.csv', index=True)
print(df.head())
```

```
                Date Created  Number of Likes  \
0  2023-06-30 09:21:00+00:00                0
1  2023-06-30 09:20:57+00:00                0
2  2023-06-30 09:20:22+00:00                0
3  2023-06-30 09:20:00+00:00                3
4  2023-06-30 09:15:22+00:00                3

                                              Tweets sentiment
0                                         movie good  Positive
1  playing book ticket now 🎥🍿🎟 gopalancinemas   ...  Positive
2                               ponilemova adipurush   Neutral
3  adipurush vs  hoorain vs kerala story controve...   Neutral
4                    st adipurush httpstcolsgkcgqukl   Neutral
```

In [ ]: