In [2]:

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:

```python
df=pd.read_csv("C:\\Users\\asdfgh\\Desktop\\ds prodigy infotech\\twitter_data.csv")
df
```

Out[3]:

	2401	Borderlands	Positive	im getting on borderlands and i will murder you all ,
0	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
1	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
2	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
3	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
4	2401	Borderlands	Positive	im getting into borderlands and i can murder y...
...	...	...	...	...
74676	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74677	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74678	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74679	9200	Nvidia	Positive	Just realized between the windows partition of...
74680	9200	Nvidia	Positive	Just like the windows partition of my Mac is l...

74681 rows × 4 columns

In [4]:

```python
df.loc[-1] = df.columns
df.index = df.index + 1
df = df.sort_index()
df.columns = range(df.shape[1])
print(df)
```

```
          0           1         2  \
0      2401  Borderlands  Positive
1      2401  Borderlands  Positive
2      2401  Borderlands  Positive
3      2401  Borderlands  Positive
4      2401  Borderlands  Positive
...     ...          ...       ...
74677  9200       Nvidia  Positive
74678  9200       Nvidia  Positive
74679  9200       Nvidia  Positive
74680  9200       Nvidia  Positive
74681  9200       Nvidia  Positive


                                                      3
0      im getting on borderlands and i will murder yo...
1      I am coming to the borders and I will kill you...
2      im getting on borderlands and i will kill you ...
3      im coming on borderlands and i will murder you...
4      im getting on borderlands 2 and i will murder ...
...                                                  ...
74677  Just realized that the Windows partition of my...
74678  Just realized that my Mac window partition is ...
74679  Just realized the windows partition of my Mac ...
74680  Just realized between the windows partition of...
74681  Just like the windows partition of my Mac is l...

[74682 rows x 4 columns]
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 74682 entries, 0 to 74681
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       74682 non-null  object
 1   1       74682 non-null  object
 2   2       74682 non-null  object
 3   3       73996 non-null  object
dtypes: object(4)
memory usage: 2.8+ MB
```

In [6]:

```
df.describe()
```

Out[6]:

	0	1	2	3
count	74682	74682	74682	73996
unique	12448	32	4	69489
top	5203	TomClancysRainbowSix	Negative	
freq	6	2400	22542	172

In [7]:

```
#ID
df[0].unique()
```

Out[7]:

```
array(['2401', 2401, 2402, ..., 9198, 9199, 9200], dtype=object)
```

In [8]:

```
#Entities
df[1].unique()
```

Out[8]:

```
array(['Borderlands', 'CallOfDutyBlackopsColdWar', 'Amazon', 'Overwatch',
       'Xbox(Xseries)', 'NBA2K', 'Dota2', 'PlayStation5(PS5)',
       'WorldOfCraft', 'CS-GO', 'Google', 'AssassinsCreed', 'ApexLegends',
       'LeagueOfLegends', 'Fortnite', 'Microsoft', 'Hearthstone',
       'Battlefield', 'PlayerUnknownsBattlegrounds(PUBG)', 'Verizon',
       'HomeDepot', 'FIFA', 'RedDeadRedemption(RDR)', 'CallOfDuty',
       'TomClancysRainbowSix', 'Facebook', 'GrandTheftAuto(GTA)',
       'MaddenNFL', 'johnson&johnson', 'Cyberpunk2077',
       'TomClancysGhostRecon', 'Nvidia'], dtype=object)
```

In [9]:

```
#Sentiments
df[2].unique()
```

Out[9]:

```
array(['Positive', 'Neutral', 'Negative', 'Irrelevant'], dtype=object)
```

In [10]:

```
#Tweets
df[3].unique()
```

Out[10]:

```
array(['im getting on borderlands and i will murder you all ,',
       'I am coming to the borders and I will kill you all,',
       'im getting on borderlands and i will kill you all,', ...,
       'Just realized the windows partition of my Mac is now 6 years behind on Nvi
dia drivers and I have no idea how he didn’t notice',
       'Just realized between the windows partition of my Mac is like being 6 year
s behind on Nvidia drivers and cars I have no fucking idea how I ever didn ’ t not
ice',
       'Just like the windows partition of my Mac is like 6 years behind on its dr
ivers So you have no idea how I didn’t notice'],
      dtype=object)
```

In [11]:

```
col_names=["Id","Entities","Sentiments","Tweets"]
df.columns=col_names
df
```

Out[11]:

	Id	Entities	Sentiments	Tweets
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo...
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you...
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you ...
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you...
4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder ...
...	...	...	...	...
74677	9200	Nvidia	Positive	Just realized that the Windows partition of my...
74678	9200	Nvidia	Positive	Just realized that my Mac window partition is ...
74679	9200	Nvidia	Positive	Just realized the windows partition of my Mac ...
74680	9200	Nvidia	Positive	Just realized between the windows partition of...
74681	9200	Nvidia	Positive	Just like the windows partition of my Mac is l...

74682 rows × 4 columns

In [13]:

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

Out[13]:

```
Id              0
Entities        0
Sentiments      0
Tweets        686
dtype: int64
```

In [14]:

```
df.drop_duplicates(inplace=True)
df.isnull().sum()
```

Out[14]:

```
Id              0
Entities        0
Sentiments      0
Tweets        326
dtype: int64
```

In [15]:

```
df.reset_index(drop=True,inplace=True)
df.isnull().sum()
```

Out[15]:

```
Id              0
Entities        0
Sentiments      0
Tweets        326
dtype: int64
```

In [16]:

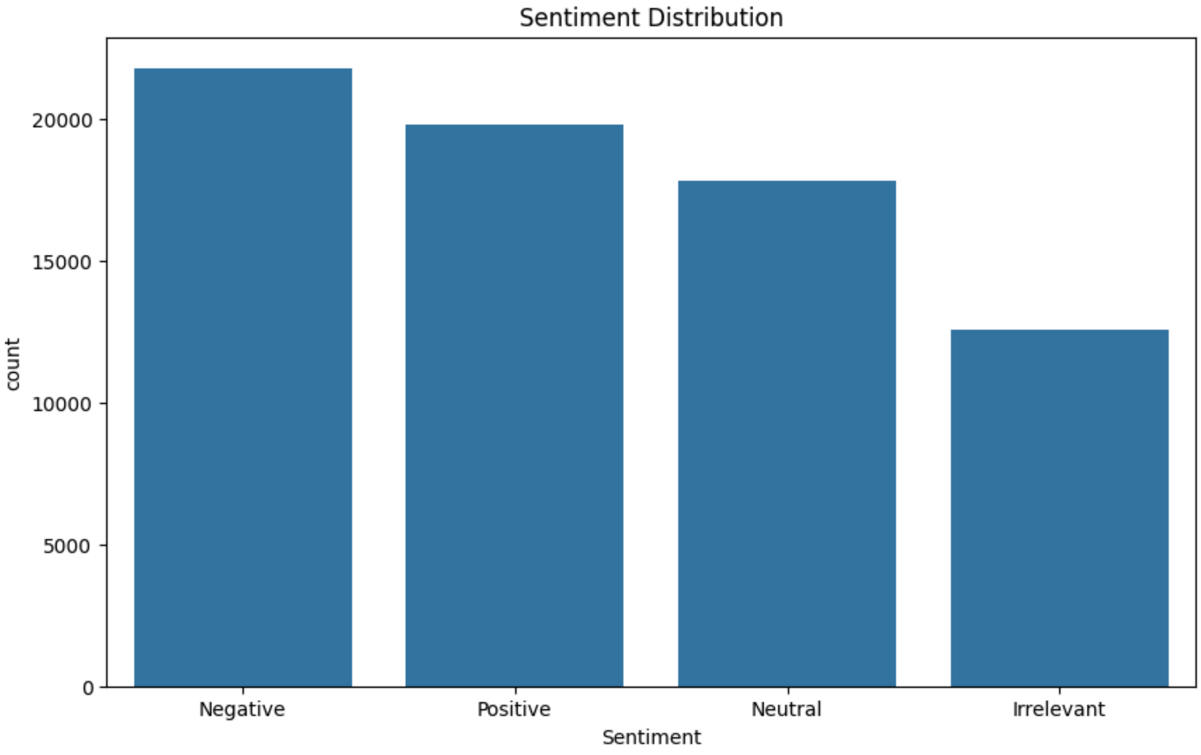
```
sentiments_counts=df["Sentiments"].value_counts()
sentiments_counts
```

Out[16]:

```
Sentiments
Negative      21787
Positive      19811
Neutral       17799
Irrelevant    12584
Name: count, dtype: int64
```

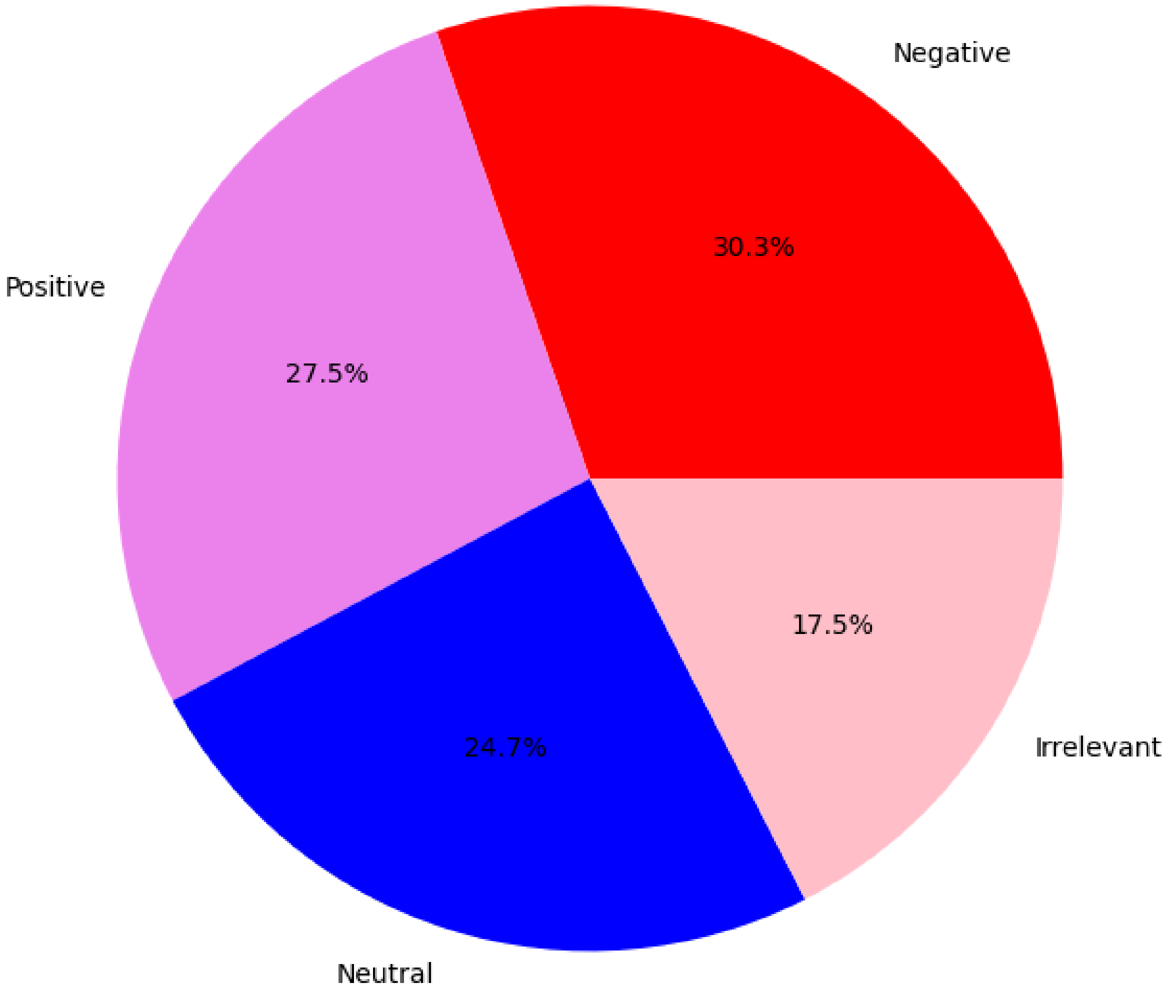
In [17]:

```
plt.figure(figsize=(10,6))
sns.countplot(data=df, x="Sentiments", order=df["Sentiments"].value_counts().index)
plt.title("Sentiment Distribution")
plt.xlabel("Sentiment")
plt.show()
```



In [18]:

```
c = ["red","violet","blue","pink"]
plt.figure(figsize=(8,8))
plt.pie(sentiments_counts, labels=sentiments_counts.index, autopct="%1.1f%%",colors
plt.show()
```



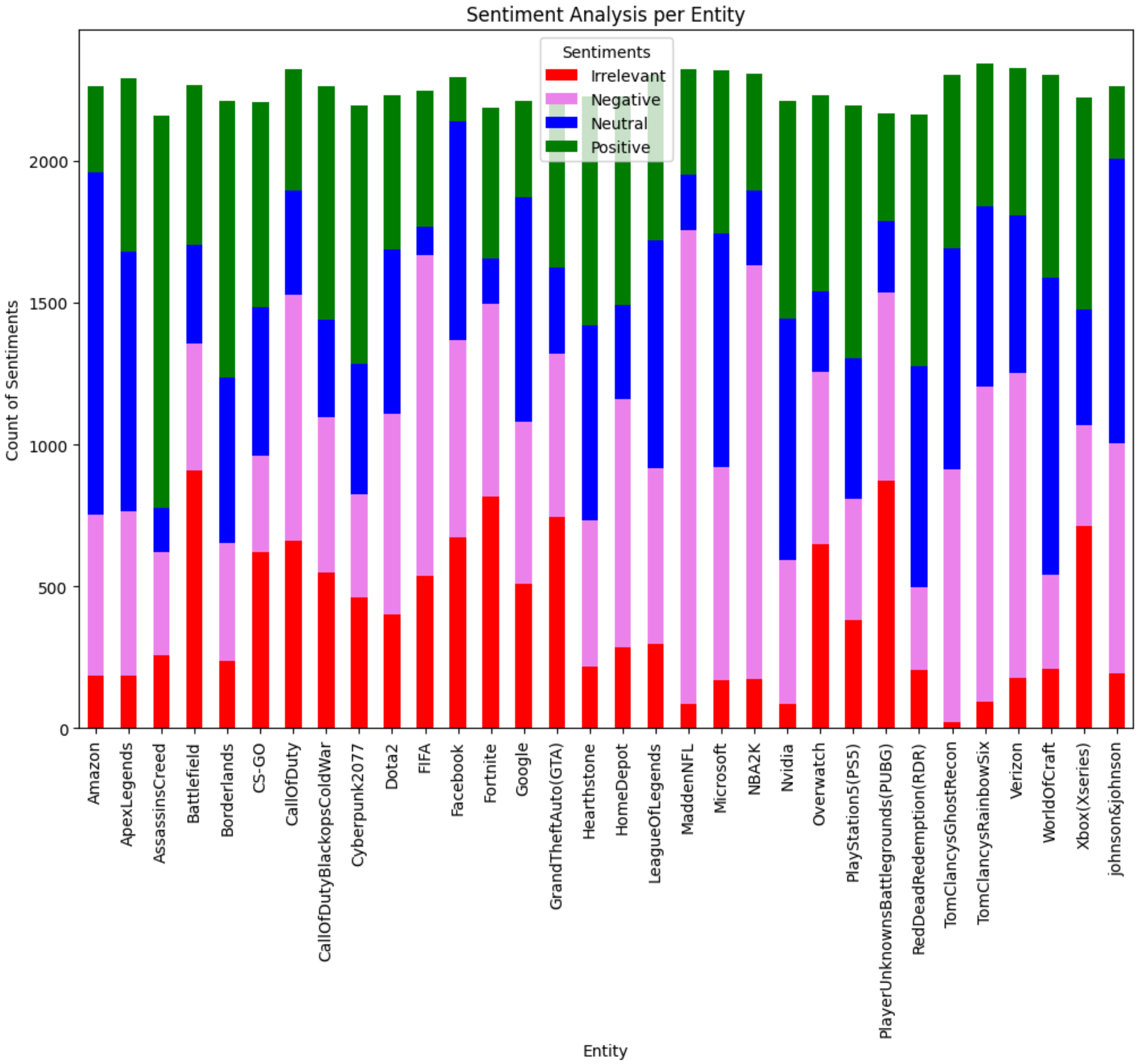
In [19]:

```
entity_sentiment_counts = df.groupby(["Entities", "Sentiments"]).size().unstack().f
print(entity_sentiment_counts)
```

```
Sentiments                         Irrelevant  Negative  Neutral  Positive
Entities
Amazon                                    187       566     1207       304
ApexLegends                               185       579      915       610
AssassinsCreed                            256       366      153      1385
Battlefield                               908       449      345       565
Borderlands                               238       415      584       974
CS-GO                                     622       337      525       723
CallOfDuty                                660       866      368       428
CallOfDutyBlackopsColdWar                 549       546      344       822
Cyberpunk2077                             462       363      458       910
Dota2                                     402       706      579       542
FIFA                                      538      1130      100       477
Facebook                                  674       692      773       154
Fortnite                                  818       676      161       532
Google                                    508       571      792       339
GrandTheftAuto(GTA)                       746       573      303       592
Hearthstone                               218       515      687       807
HomeDepot                                 285       875      331       735
LeagueOfLegends                           298       620      802       583
MaddenNFL                                  86      1671      193       373
Microsoft                                 169       752      823       575
NBA2K                                     175      1455      265       411
Nvidia                                     86       505      851       769
Overwatch                                 648       607      284       690
PlayStation5(PS5)                         382       425      495       894
PlayerUnknownsBattlegrounds(PUBG)         871       664      252       380
RedDeadRedemption(RDR)                    204       291      779       888
TomClancysGhostRecon                       23       889      781       608
TomClancysRainbowSix                       92      1113      634       505
Verizon                                   179      1074      555       520
WorldOfCraft                              210       329     1049       716
Xbox(Xseries)                             712       357      406       747
johnson&johnson                           193       810     1005       253
```

In [24]:

```python
c = ["red","violet","blue","green"]
entity_sentiment_counts.plot(kind="bar", stacked=True, figsize=(12, 8),color=c)
plt.title("Sentiment Analysis per Entity")
plt.xlabel("Entity")
plt.ylabel("Count of Sentiments")
plt.show()
```



```
In [20]: entity_sentiment_counts = df.groupby(["Entities", "Sentiments"]).size().unstack().f
         print(entity_sentiment_counts.sort_values(by="Positive"))
```

```
Sentiments                          Irrelevant  Negative  Neutral  Positive
Entities
Facebook                                   674       692      773       154
johnson&johnson                            193       810     1005       253
Amazon                                     187       566     1207       304
Google                                     508       571      792       339
MaddenNFL                                   86      1671      193       373
PlayerUnknownsBattlegrounds(PUBG)          871       664      252       380
NBA2K                                      175      1455      265       411
CallOfDuty                                 660       866      368       428
FIFA                                       538      1130      100       477
TomClancysRainbowSix                        92      1113      634       505
Verizon                                    179      1074      555       520
Fortnite                                   818       676      161       532
Dota2                                      402       706      579       542
Battlefield                                908       449      345       565
Microsoft                                  169       752      823       575
LeagueOfLegends                            298       620      802       583
GrandTheftAuto(GTA)                        746       573      303       592
TomClancysGhostRecon                        23       889      781       608
ApexLegends                                185       579      915       610
Overwatch                                  648       607      284       690
WorldOfCraft                               210       329     1049       716
CS-GO                                      622       337      525       723
HomeDepot                                  285       875      331       735
Xbox(Xseries)                              712       357      406       747
Nvidia                                      86       505      851       769
Hearthstone                                218       515      687       807
CallOfDutyBlackopsColdWar                  549       546      344       822
RedDeadRedemption(RDR)                     204       291      779       888
PlayStation5(PS5)                          382       425      495       894
Cyberpunk2077                              462       363      458       910
Borderlands                                238       415      584       974
AssassinsCreed                             256       366      153      1385
```

In [21]:

```python
import re
from textblob import TextBlob

# Function to clean tweet text
def clean_tweet(tweet):
    tweet = tweet.lower()  # Convert to lowercase
    tweet = re.sub(r'http\S+|www\S+|https\S+', '', tweet, flags=re.MULTILINE)  # Re
    tweet = re.sub(r'\@\w+|\#', '', tweet)  # Remove mentions and hashtags
    tweet = re.sub(r'[^a-zA-Z\s]', '', tweet)  # Remove special characters
    return tweet

# Example tweet
example_tweet = "I love the new features of the product! Check it out: https://exam

# Clean the tweet
cleaned_tweet = clean_tweet(example_tweet)

# Perform sentiment analysis
blob = TextBlob(cleaned_tweet)
sentiment = blob.sentiment  # Returns a tuple (polarity, subjectivity)
```

```python
print(f"Cleaned Tweet: {cleaned_tweet}")
print(f"Polarity: {sentiment.polarity}, Subjectivity: {sentiment.subjectivity}")
```

```
Cleaned Tweet: i love the new features of the product check it out  excited
Polarity: 0.3371212121212121, Subjectivity: 0.6015151515151514
```

In [22]:

```python
# Function to classify overall sentiment based on polarity
def classify_sentiment(polarity):
    if polarity > 0:
        return 'Positive'
    elif polarity < 0:
        return 'Negative'
    else:
        return 'Neutral'

# Classify the overall sentiment for the cleaned tweet
overall_sentiment = classify_sentiment(sentiment.polarity)

# Display the results
print(f"Cleaned Tweet: {cleaned_tweet}")
print(f"Polarity: {sentiment.polarity}, Subjectivity: {sentiment.subjectivity}")
print(f"Overall Sentiment: {overall_sentiment}")
```

```
Cleaned Tweet: i love the new features of the product check it out  excited
Polarity: 0.3371212121212121, Subjectivity: 0.6015151515151514
Overall Sentiment: Positive
```

In [23]:

```python
import re

def clean_tweets(text):
    # Example of removing URLs and converting to lowercase
    text = re.sub(r'http\S+', '', text)  # Remove URLs
    text = re.sub(r'[^a-zA-Z\s]', '', text)  # Remove non-alphabetic characters
    text = text.lower()  # Convert to lowercase
    return text
print(df.columns)
def clean_tweets(text):
    if pd.isnull(text):
        return ""  # or np.nan, or any default value you want
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    return text.lower()

df['Tweets'].dropna(inplace=True)
df['Cleaned_Tweets'] = df['Tweets'].apply(clean_tweets)
print(df['Cleaned_Tweets'])
```

```
Index(['Id', 'Entities', 'Sentiments', 'Tweets'], dtype='object')
0        im getting on borderlands and i will murder yo...
1        i am coming to the borders and i will kill you...
2        im getting on borderlands and i will kill you all
3        im coming on borderlands and i will murder you...
4        im getting on borderlands  and i will murder y...
                               ...                        
71976    just realized that the windows partition of my...
71977    just realized that my mac window partition is ...
71978    just realized the windows partition of my mac ...
71979    just realized between the windows partition of...
71980    just like the windows partition of my mac is l...
Name: Cleaned_Tweets, Length: 71981, dtype: object
```

In [24]:

```
from textblob import TextBlob
df['Cleaned_Tweets'] = df['Cleaned_Tweets'].fillna('')
print(df.columns)
df['Polarity'] = df['Cleaned_Tweets'].apply(lambda tweet: TextBlob(tweet).sentiment
df['Subjectivity'] = df['Cleaned_Tweets'].apply(lambda tweet: TextBlob(tweet).senti
print(df['Polarity'])
print(df['Subjectivity'])
```

```
Index(['Id', 'Entities', 'Sentiments', 'Tweets', 'Cleaned_Tweets'], dtype='object')
0        0.00
1        0.00
2        0.00
3        0.00
4        0.00
         ... 
71976   -0.40
71977   -0.40
71978   -0.40
71979   -0.05
71980   -0.40
Name: Polarity, Length: 71981, dtype: float64
0        0.00
1        0.00
2        0.00
3        0.00
4        0.00
         ... 
71976    0.70
71977    0.70
71978    0.70
71979    0.75
71980    0.70
Name: Subjectivity, Length: 71981, dtype: float64
```

In [25]:

```
# Define classify_sentiment function
def classify_sentiment(polarity):
    if polarity > 0:
        return "Positive"
    elif polarity < 0:
        return "Negative"
    else:
        return "Neutral"
```

```
# Classify Overall Sentiment
df['Overall_Sentiment'] = df['Polarity'].apply(classify_sentiment)
print(df['Overall_Sentiment'])

# Display the DataFrame
print(df[['Tweets', 'Cleaned_Tweets', 'Polarity', 'Subjectivity', 'Overall_Sentimen
```

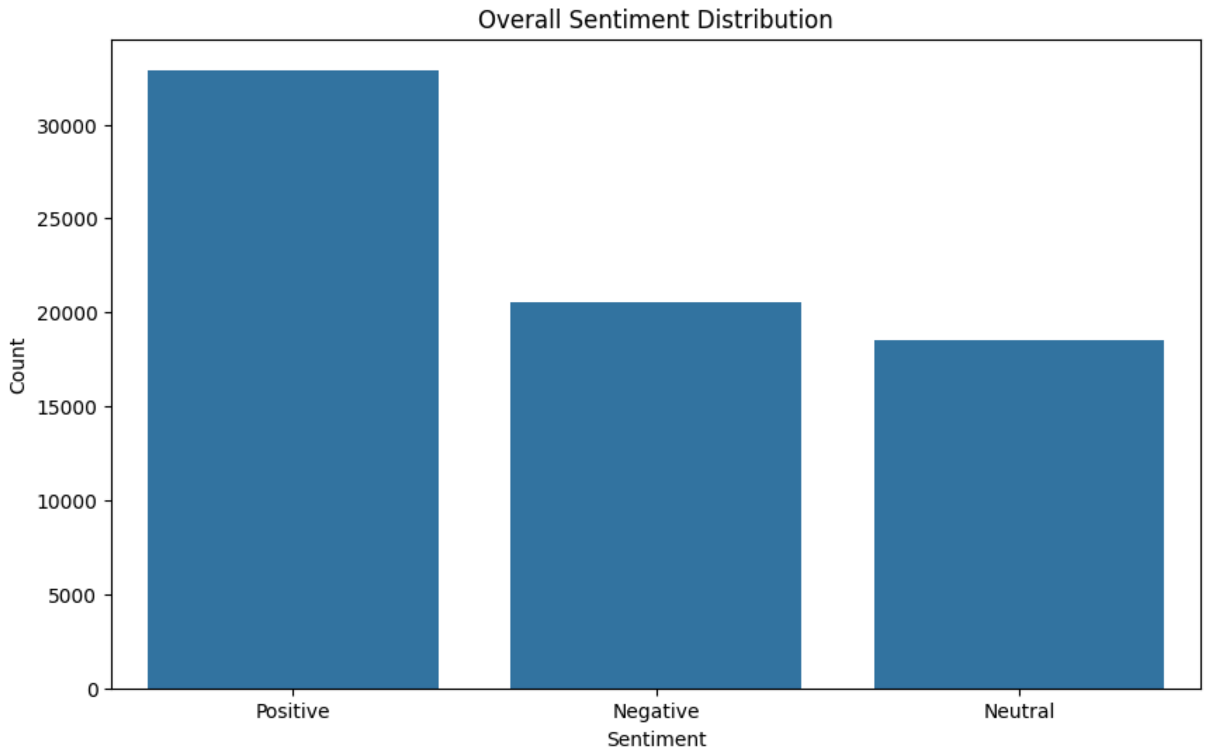
```
0         Neutral
1         Neutral
2         Neutral
3         Neutral
4         Neutral
           ...
71976    Negative
71977    Negative
71978    Negative
71979    Negative
71980    Negative
Name: Overall_Sentiment, Length: 71981, dtype: object
                                              Tweets  \
0  im getting on borderlands and i will murder yo...
1  I am coming to the borders and I will kill you...
2  im getting on borderlands and i will kill you ...
3  im coming on borderlands and i will murder you...
4  im getting on borderlands 2 and i will murder ...

                                      Cleaned_Tweets  Polarity  Subjectivity  \
0  im getting on borderlands and i will murder yo...       0.0           0.0
1  i am coming to the borders and i will kill you...       0.0           0.0
2  im getting on borderlands and i will kill you all       0.0           0.0
3  im coming on borderlands and i will murder you...       0.0           0.0
4  im getting on borderlands  and i will murder y...       0.0           0.0

  Overall_Sentiment
0           Neutral
1           Neutral
2           Neutral
3           Neutral
4           Neutral
```

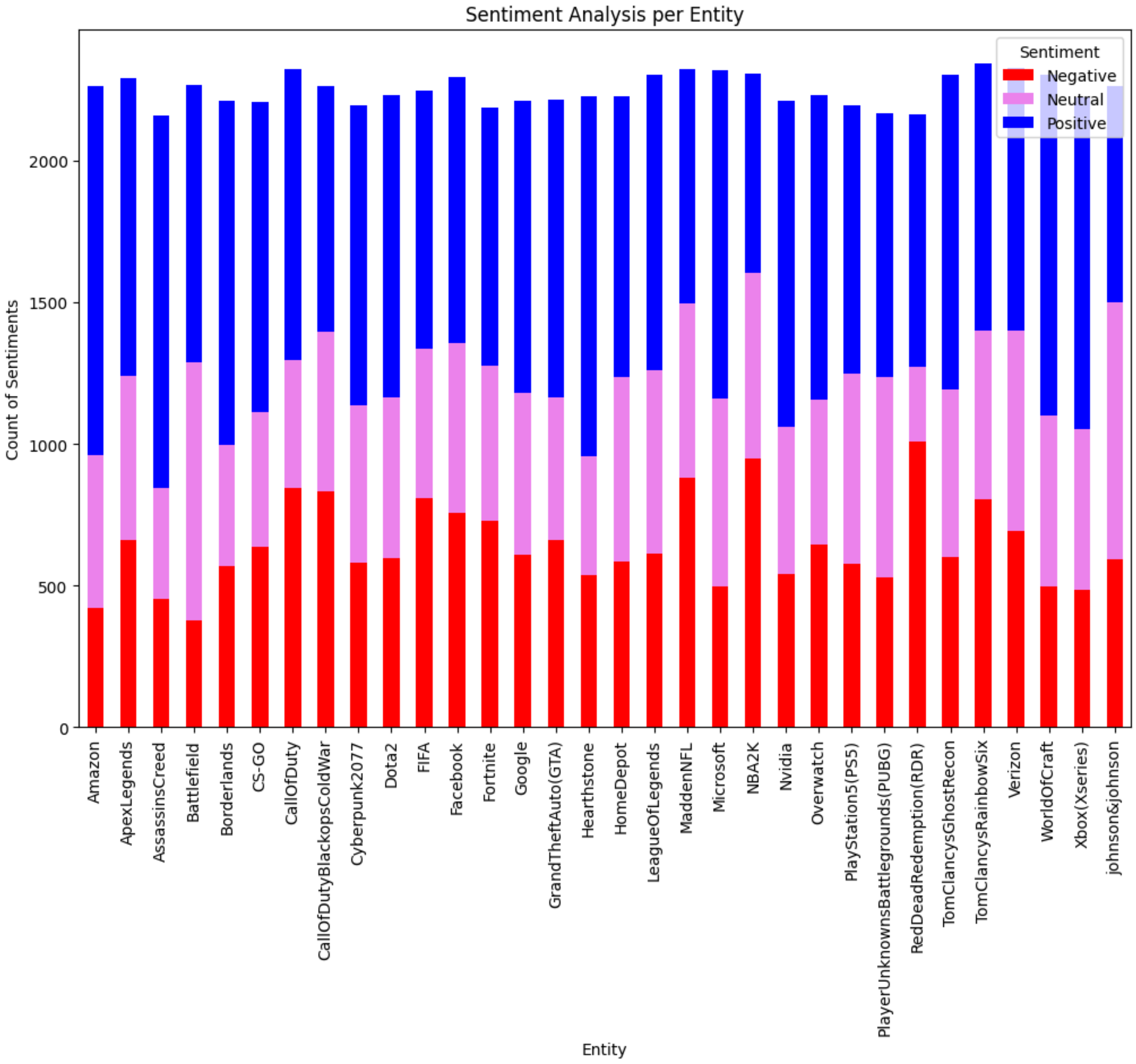
In [27]:

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Overall_Sentiment', order=df['Overall_Sentiment'].value_c
plt.title('Overall Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```



In [28]:

```
# Sentiment counts by entities
entity_sentiment_counts = df.groupby(["Entities", "Overall_Sentiment"]).size().unst
c = ["red","violet","blue"]
# Plotting
entity_sentiment_counts.plot(kind="bar", stacked=True, figsize=(12, 8),color=c)
plt.title("Sentiment Analysis per Entity")
plt.xlabel("Entity")
plt.ylabel("Count of Sentiments")
plt.legend(title='Sentiment')
plt.show()
```



In [29]:

```
summary = df[['Polarity', 'Subjectivity']].describe()
print(summary)
```

```
           Polarity  Subjectivity
count  71981.000000  71981.000000
mean       0.072764      0.439938
std        0.339364      0.312131
min       -1.000000      0.000000
25%       -0.066667      0.166667
50%        0.000000      0.475000
75%        0.255556      0.662500
max        1.000000      1.000000
```

In [30]:

```
# Set a threshold for negative sentiment ratio (e.g., 20%)
threshold = 20  # Adjust this value as needed

# Calculate negative sentiment ratio
negative_sentiment_count = df[df['Overall_Sentiment'] == 'Negative'].shape[0]
total_tweets = df.shape[0]
negative_sentiment_ratio = (negative_sentiment_count / total_tweets) * 100

print("Total tweets:", total_tweets)
```

```python
print("Negative sentiment ratio:", negative_sentiment_ratio)

# Check if negative sentiment ratio exceeds threshold
if negative_sentiment_ratio > threshold:
    print("Alert: High negative sentiment detected!")
else:
    print("Negative sentiment is within normal range.")
```

```
Total tweets: 71981
Negative sentiment ratio: 28.551978994456874
Alert: High negative sentiment detected!
```

In [31]:

```python
# Display column names to ensure 'Cleaned_Tweets' exists
print("Columns:", df.columns)

# Check a sample of 'Cleaned_Tweets' to verify contents
print(df['Cleaned_Tweets'].head())
```

```
Columns: Index(['Id', 'Entities', 'Sentiments', 'Tweets', 'Cleaned_Tweets', 'Polarit
y',
       'Subjectivity', 'Overall_Sentiment'],
      dtype='object')
0    im getting on borderlands and i will murder yo...
1    i am coming to the borders and i will kill you...
2    im getting on borderlands and i will kill you all
3    im coming on borderlands and i will murder you...
4    im getting on borderlands  and i will murder y...
Name: Cleaned_Tweets, dtype: object
```

In [33]:

```python
from gensim.models import LdaModel

try:
    lda_model = LdaModel(corpus, num_topics=5, id2word=dictionary, passes=15)
    topics = lda_model.print_topics(num_words=5)
    for topic in topics:
        print(topic)
except Exception as e:
    print("Error building LDA model:", e)
```

```
Error building LDA model: name 'corpus' is not defined
```

In [34]:

```python
import tweepy

# Replace with your own credentials
consumer_key = 'YOUR_API_KEY'
consumer_secret = 'YOUR_API_SECRET_KEY'
access_token = 'YOUR_ACCESS_TOKEN'
access_token_secret = 'YOUR_ACCESS_TOKEN_SECRET'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

# Check if you're connected
try:
    api.verify_credentials()
    print("Authentication successful")
except:
    print("Authentication failed")
```

```
Authentication failed
```

In [36]:

```
from textblob import TextBlob

# Function to perform sentiment analysis
def get_sentiment(text):
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity  # Range: -1 (negative) to 1 (positive)
    subjectivity = blob.sentiment.subjectivity  # Range: 0 (objective) to 1 (subjec

    if polarity > 0:
        return "Positive"
    elif polarity < 0:
        return "Negative"
    else:
        return "Neutral"

# Example tweet sentiment analysis
tweet = "I love this product!"
print(f"Sentiment: {get_sentiment(tweet)}")
```

```
Sentiment: Positive
```

In [37]:

```
import pandas as pd

# Sample DataFrame
df = pd.DataFrame({
    'Tweets': [
        "I love the new features of the product! Check it out: https://example.com
        "Not satisfied with the recent update :(",
        "The product is amazing! #happy",
        "I don't like the new changes... #disappointed"
    ]
})
```

In [38]:

```
print(df.columns)
```

```
Index(['Tweets'], dtype='object')
```

In [39]:

```
import re
from textblob import TextBlob

# Function to clean tweet text
def clean_tweet(tweet):
    tweet = tweet.lower()  # Convert to lowercase
    tweet = re.sub(r'http\S+|www\S+|https\S+', '', tweet)  # Remove URLs
    tweet = re.sub(r'\@\w+|\#', '', tweet)  # Remove mentions and hashtags
    tweet = re.sub(r'[^a-zA-Z\s]', '', tweet)  # Remove special characters
    return tweet

# Clean the "Tweets" column
df['Cleaned_Tweets'] = df['Tweets'].apply(clean_tweet)
```

```
# Add polarity and subjectivity columns
df['Polarity'] = df['Cleaned_Tweets'].apply(lambda tweet: TextBlob(tweet).sentiment
df['Subjectivity'] = df['Cleaned_Tweets'].apply(lambda tweet: TextBlob(tweet).senti

# Display the first few rows to confirm the new columns
print(df[['Tweets', 'Cleaned_Tweets', 'Polarity', 'Subjectivity']])
```

```
                                              Tweets  \
0  I love the new features of the product! Check ...   
1            Not satisfied with the recent update :(   
2                     The product is amazing! #happy   
3      I don't like the new changes... #disappointed   

                                      Cleaned_Tweets  Polarity  Subjectivity  
0  i love the new features of the product check i...  0.337121      0.601515  
1               not satisfied with the recent update  -0.125000      0.625000  
2                       the product is amazing happy  0.700000      0.950000  
3           i dont like the new changes disappointed -0.306818      0.602273  
```

In [40]:

```
from gensim.corpora import Dictionary
# Tokenize the tweets into words
tokenized_tweets = [tweet.split() for tweet in df['Cleaned_Tweets']]
# Create a dictionary
dictionary = Dictionary(tokenized_tweets)
# Create a corpus
corpus = [dictionary.doc2bow(tweet) for tweet in tokenized_tweets]

print(tokenized_tweets)
print(dictionary)
print(corpus)
```

```
[['i', 'love', 'the', 'new', 'features', 'of', 'the', 'product', 'check', 'it', 'ou
t', 'excited'], ['not', 'satisfied', 'with', 'the', 'recent', 'update'], ['the', 'pr
oduct', 'is', 'amazing', 'happy'], ['i', 'dont', 'like', 'the', 'new', 'changes', 'd
isappointed']]
Dictionary<23 unique tokens: ['check', 'excited', 'features', 'i', 'it']...>
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (1
0, 2)], [(10, 1), (11, 1), (12, 1), (13, 1), (14, 1), (15, 1)], [(9, 1), (10, 1), (1
6, 1), (17, 1), (18, 1)], [(3, 1), (6, 1), (10, 1), (19, 1), (20, 1), (21, 1), (22,
1)]]
```

In [41]:

```
lda_model = LdaModel(corpus, num_topics=5, id2word=dictionary, passes=15)
topics = lda_model.print_topics(num_words=5)
for topic in topics:
    print(topic)
```

```
(0, '0.043*"product" + 0.043*"the" + 0.043*"amazing" + 0.043*"happy" + 0.043*"new"')
(1, '0.043*"product" + 0.043*"the" + 0.043*"is" + 0.043*"i" + 0.043*"new"')
(2, '0.043*"product" + 0.043*"the" + 0.043*"amazing" + 0.043*"is" + 0.043*"i"')
(3, '0.133*"the" + 0.072*"product" + 0.072*"i" + 0.072*"new" + 0.072*"check"')
(4, '0.142*"the" + 0.053*"dont" + 0.053*"changes" + 0.053*"like" + 0.053*"disappoint
ed"')
```

In [42]:

```
from gensim import corpora
from gensim.models import LdaModel

# Preprocess tweets
```

```
def preprocess_tweets(tweets):
    stopwords = set(["the", "and", "in", "to", "for", "of"])  # Add more stopwords
    return [[word for word in tweet.split() if word not in stopwords] for tweet in

processed_tweets = preprocess_tweets(df['Cleaned_Tweets'])

# Create dictionary and corpus
dictionary = corpora.Dictionary(processed_tweets)
corpus = [dictionary.doc2bow(tweet) for tweet in processed_tweets]

# Train LDA model
lda_model = LdaModel(corpus, num_topics=3, id2word=dictionary, passes=15)

# Display topics
topics = lda_model.print_topics(num_words=5)
for topic in topics:
    print(topic)
```

```
(0, '0.111*"with" + 0.111*"recent" + 0.111*"update" + 0.111*"not" + 0.111*"satisfie
d"')
(1, '0.117*"product" + 0.067*"features" + 0.067*"love" + 0.067*"check" + 0.067*"exci
ted"')
(2, '0.103*"i" + 0.103*"new" + 0.102*"changes" + 0.102*"disappointed" + 0.102*"lik
e"')
```

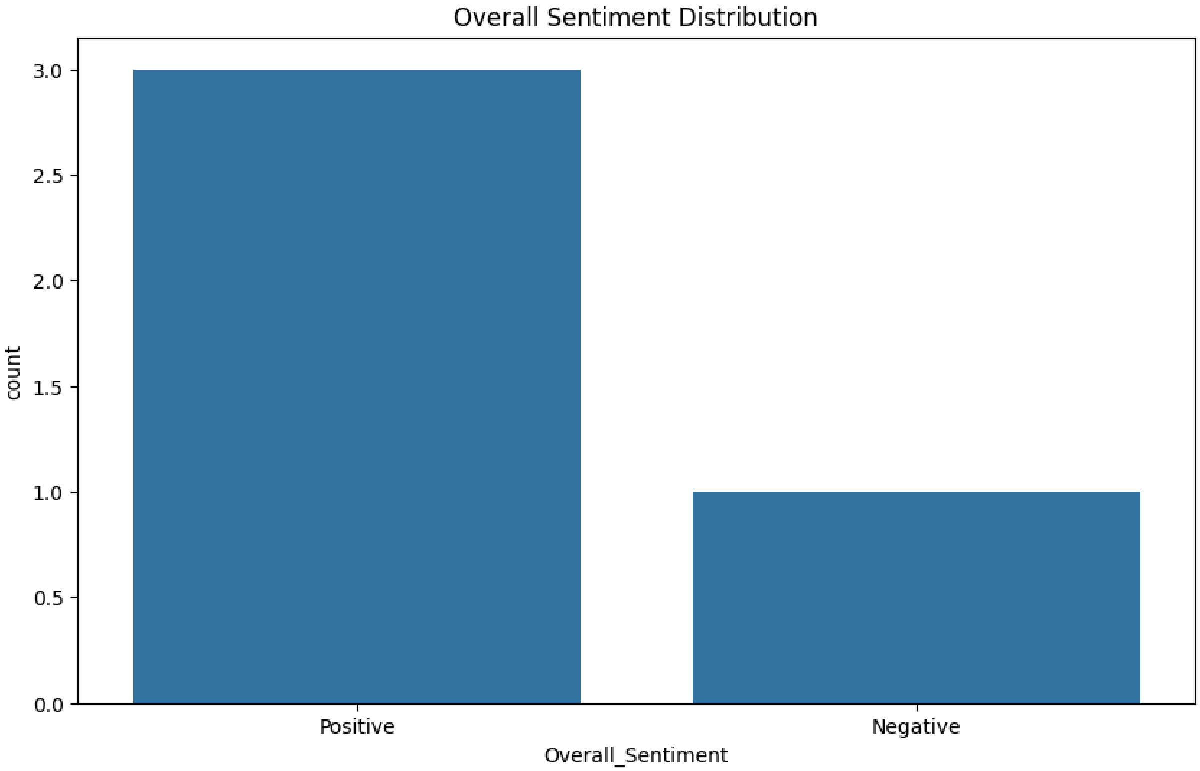
In [44]:

```
from transformers import pipeline
# Data Preprocessing Function
def clean_text(text):
    text = text.lower()
    text = re.sub(r'http\S+|www\S+', '', text)  # Remove URLs
    text = re.sub(r'@\w+|\#\w+', '', text)       # Remove mentions and hashtags
    text = re.sub(r'[^a-z\s]', '', text)         # Remove special characters
    return text

df['Cleaned_Tweets'] = df['Tweets'].apply(clean_text)

# Sentiment Analysis using TextBlob
df['Polarity'] = df['Cleaned_Tweets'].apply(lambda tweet: TextBlob(tweet).sentiment
df['Subjectivity'] = df['Cleaned_Tweets'].apply(lambda tweet: TextBlob(tweet).senti
df['Overall_Sentiment'] = df['Polarity'].apply(lambda x: 'Positive' if x > 0 else (

# Visualization
plt.figure(figsize=(10,6))
sns.countplot(data=df, x="Overall_Sentiment")
plt.title("Overall Sentiment Distribution")
plt.show()
```


Overall Sentiment Distribution
count
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Positive
Negative
Overall_Sentiment

In [ ]: