sentimentanalysis-1

January 20, 2024

[90]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

[91]:
```
df = pd.read_csv('threads_reviews.csv')
```

[92]:
```
df.head()
```

[92]:
```
        source                                 review_description  rating  \
0  Google Play  Meh. Not the greatest experience on a Chromebo…       2
1  Google Play  Pretty good for a first launch!! Its easy to u…       3
2  Google Play  For a brand new app, it's very well optimized…       3
3  Google Play  Great app with a lot of potential! However, th…       3
4  Google Play  The app is good, but it needs a lot of functio…       3

           review_date
0  2023-07-08 14:18:24
1  2023-07-19 20:52:48
2  2023-07-06 23:03:11
3  2023-07-10 00:53:25
4  2023-07-06 16:57:43
```

[93]:
```
df.shape
```

[93]:
```
(32910, 4)
```

[94]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32910 entries, 0 to 32909
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   source              32910 non-null  object
 1   review_description  32910 non-null  object
 2   rating              32910 non-null  int64
 3   review_date         32910 non-null  object
dtypes: int64(1), object(3)
memory usage: 1.0+ MB
```

[95]:

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

[95]:

```
source                0
review_description    0
rating                0
review_date           0
dtype: int64
```

[96]:

```
df.duplicated().sum()
```

[96]:

```
1
```

[97]:

```
df.drop_duplicates(keep = "first",inplace = True)
df.duplicated().sum()
```

[97]:

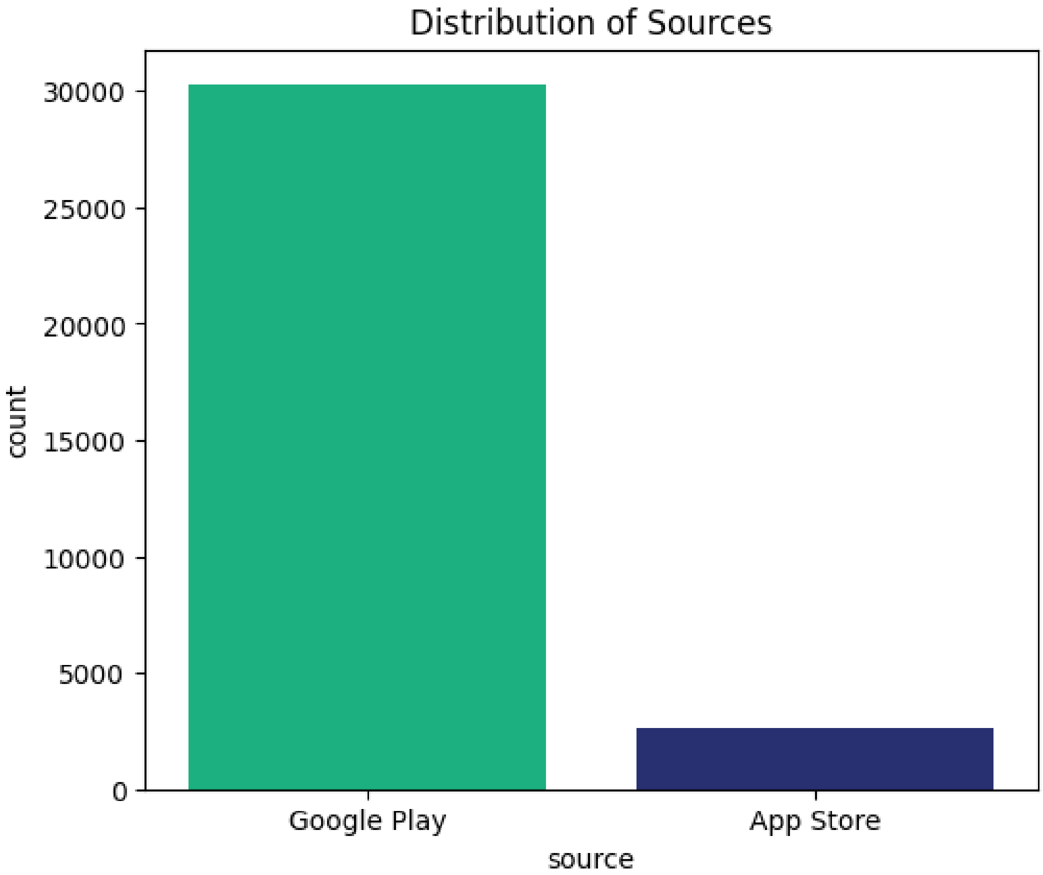
```
0
```

0.1 Data Visualization

0.1.1 Distribution of Ratings

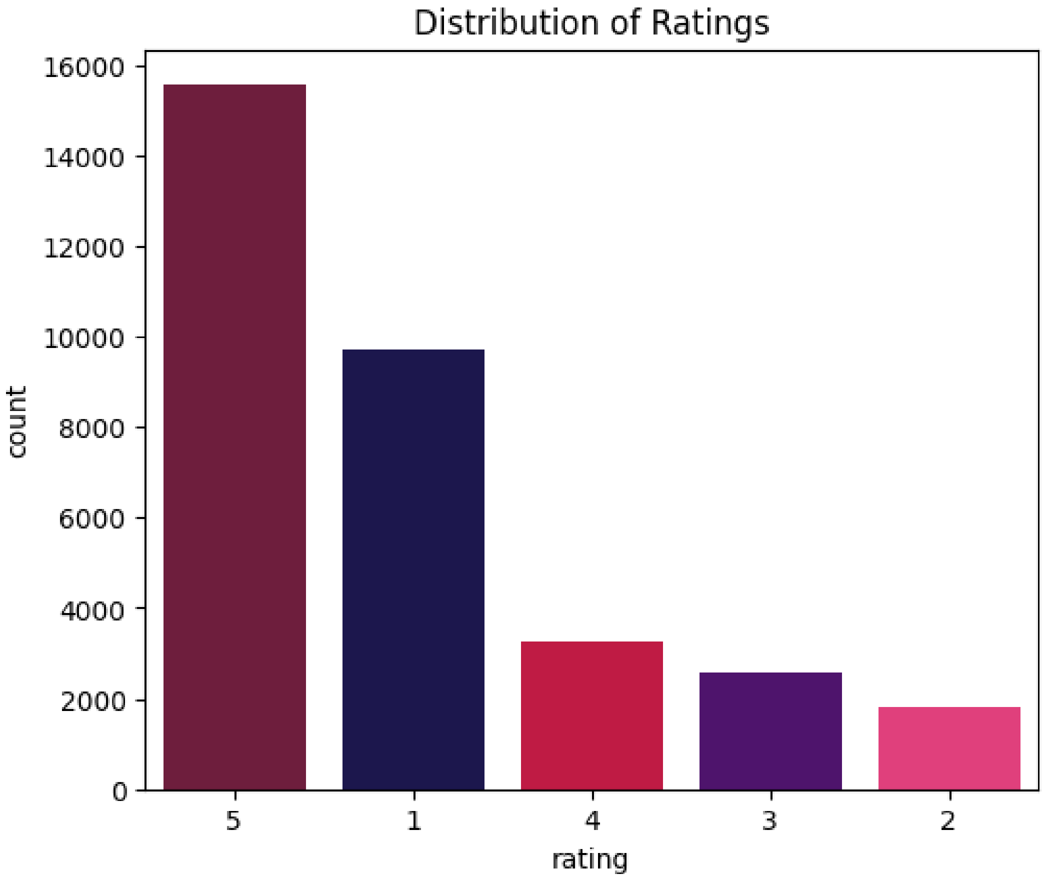
[98]:

```
plt.figure(figsize=(6, 5))
sns.countplot(x='source', data=df, palette = ["#03C988","#1D267D"],␣
  ↪order=df['source'].value_counts().index)
plt.title('Distribution of Sources')
plt.show()
```



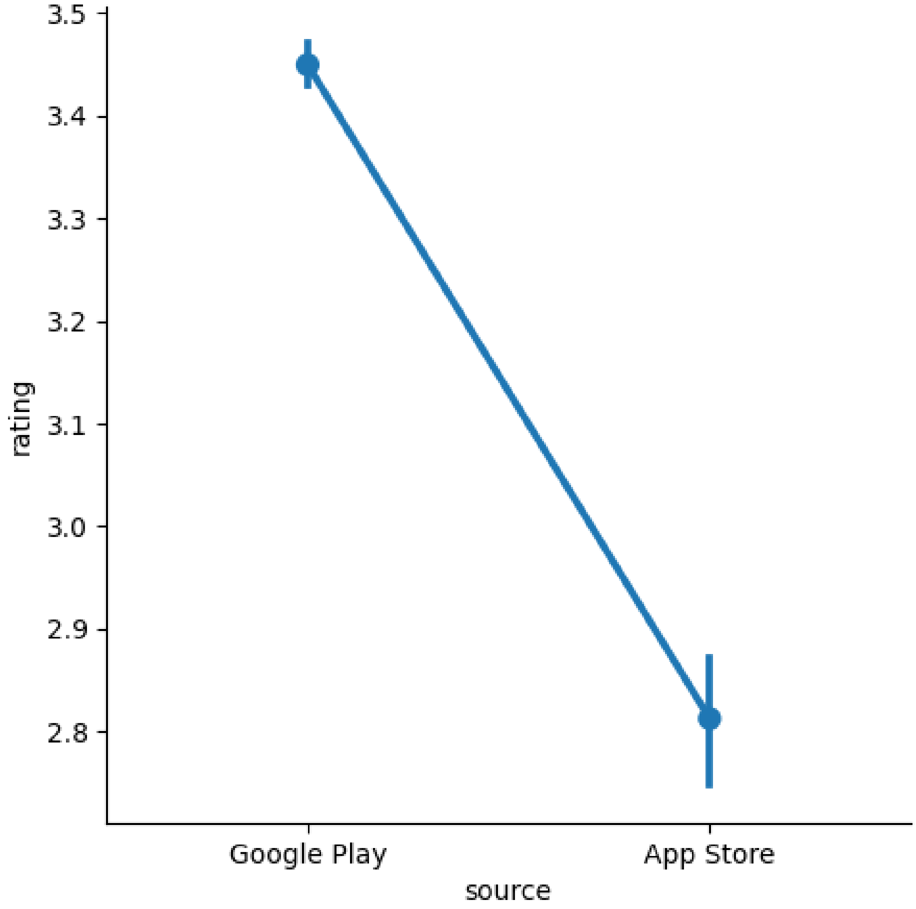
[99]:

```
plt.figure(figsize=(6, 5))
sns.countplot(x='rating', data=df, palette=["#7B113A", "#150E56", "#DA0037",␣
  ↪"#52057B", "#FB2576"], order=df['rating'].value_counts().index)
plt.title('Distribution of Ratings')
plt.show()
```



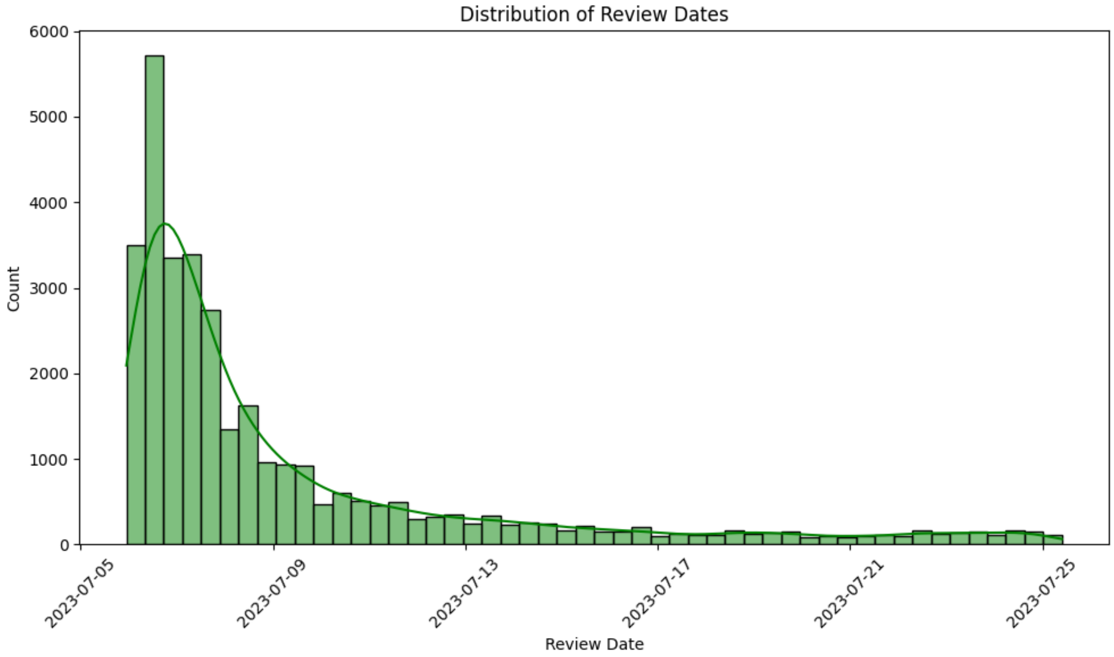
```
[100]: sns.catplot(data = df, x = "source",y = "rating",kind = "point")
```

```
[100]: <seaborn.axisgrid.FacetGrid at 0x7c6c3e81b5b0>
```



[101]:

```
df['review_date'] = pd.to_datetime(df['review_date'])
plt.figure(figsize=(10,6))
sns.histplot(data=df, x='review_date',color='green', bins=50, kde=True)
plt.title('Distribution of Review Dates')
plt.xlabel('Review Date')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

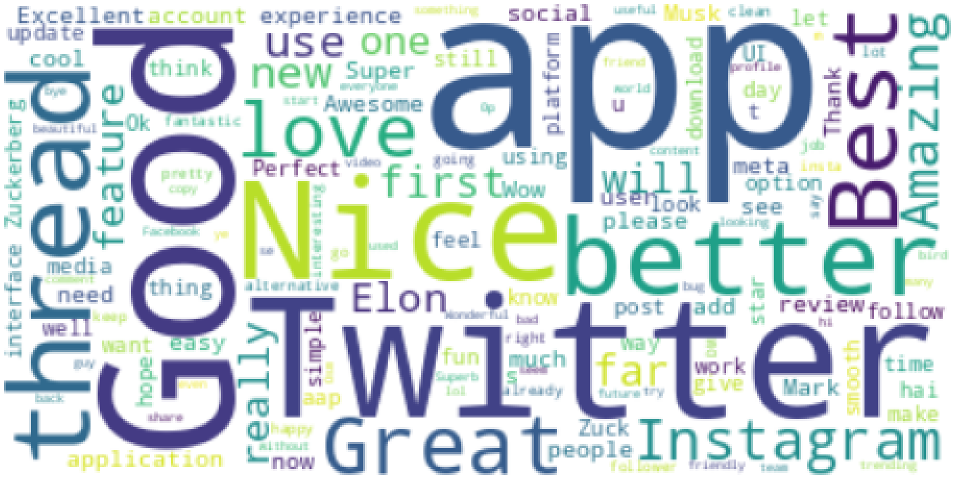


0.1.2 Word Cloud of Reviews

[102]:

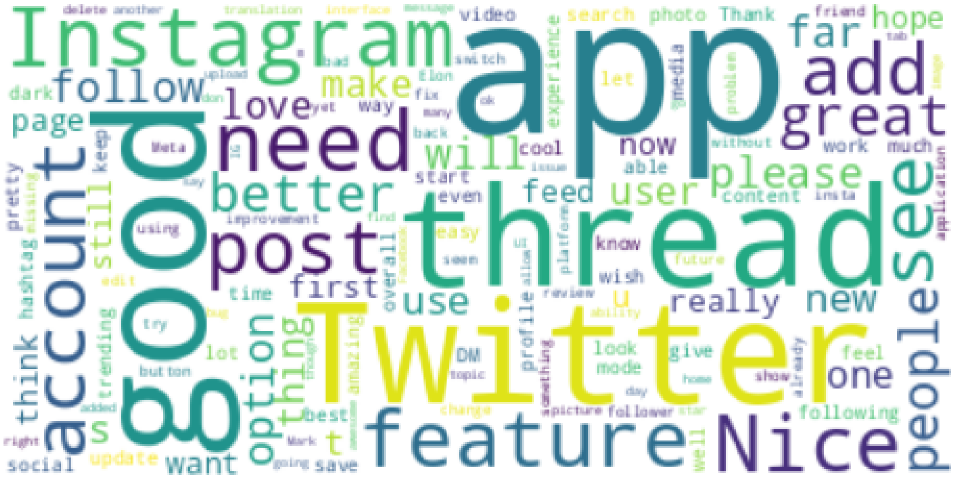
```
from wordcloud import WordCloud

def create_cloud(data):
    text = df[df['rating']==data]
    text_cloud = ' '.join(txt for txt in text['review_description'])
    word_cloud = WordCloud(collocations=False,background_color='white').
  ↪generate(text_cloud)
    plt.imshow(word_cloud,interpolation='bilinear')
    plt.title(f'Rating {data}')
    plt.axis('off')

create_cloud(5)
```

Rating 5



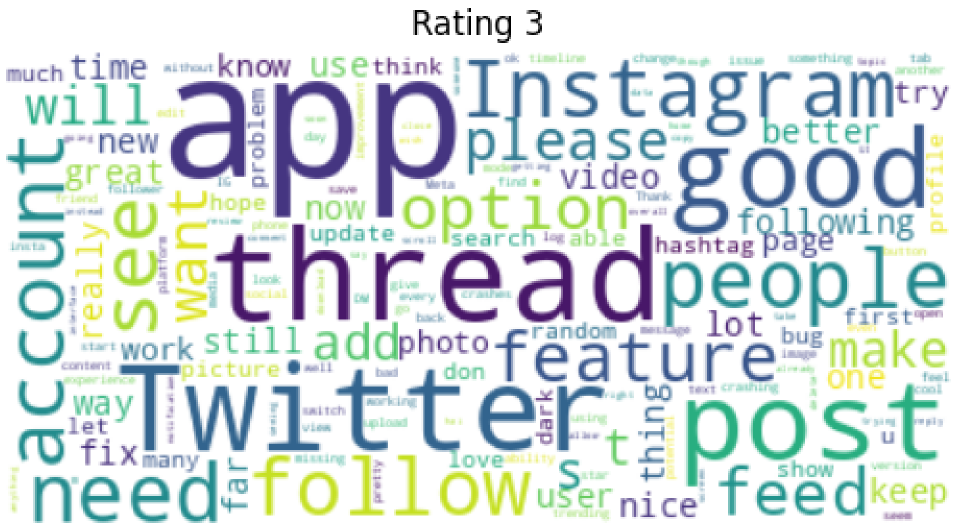
[103]:
```
from wordcloud import WordCloud

def create_cloud(data):
    text = df[df['rating']==data]
    text_cloud = ' '.join(txt for txt in text['review_description'])
    word_cloud = WordCloud(collocations=False,background_color='white').
  ↪generate(text_cloud)
    plt.imshow(word_cloud,interpolation='bilinear')
    plt.title(f'Rating {data}')
    plt.axis('off')

create_cloud(4)
```

Rating 4



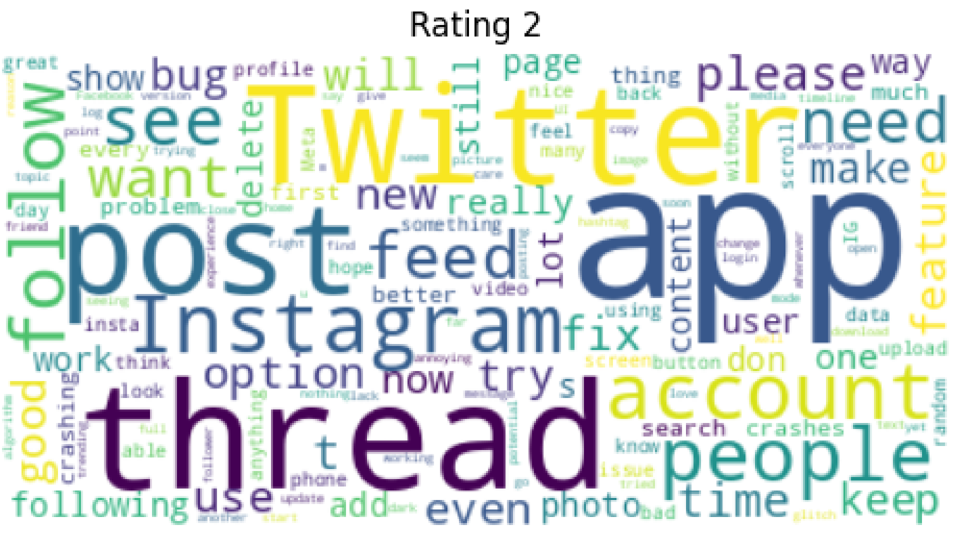
[104]:

```
from wordcloud import WordCloud

def create_cloud(data):
    text = df[df['rating']==data]
    text_cloud = ' '.join(txt for txt in text['review_description'])
    word_cloud = WordCloud(collocations=False,background_color='white').
  ↪generate(text_cloud)
    plt.imshow(word_cloud,interpolation='bilinear')
    plt.title(f'Rating {data}')
    plt.axis('off')

create_cloud(3)
```



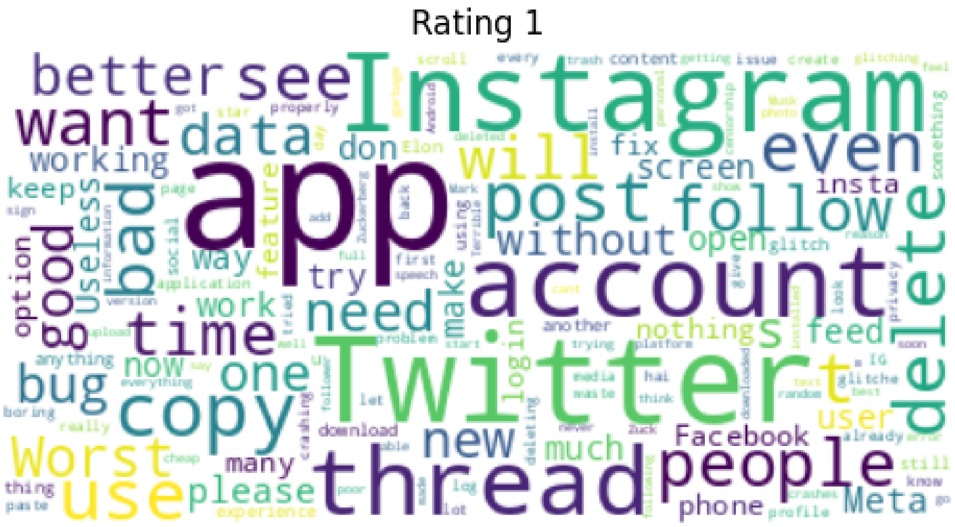
[105]:

```
from wordcloud import WordCloud

def create_cloud(data):
    text = df[df['rating']==data]
    text_cloud = ' '.join(txt for txt in text['review_description'])
    word_cloud = WordCloud(collocations=False,background_color='white').
  ↪generate(text_cloud)
    plt.imshow(word_cloud,interpolation='bilinear')
    plt.title(f'Rating {data}')
    plt.axis('off')

create_cloud(2)
```



Rating 2

```
[106]: from wordcloud import WordCloud

       def create_cloud(data):
           text = df[df['rating']==data]
           text_cloud = ' '.join(txt for txt in text['review_description'])
           word_cloud = WordCloud(collocations=False,background_color='white').
        ↪generate(text_cloud)
           plt.imshow(word_cloud,interpolation='bilinear')
           plt.title(f'Rating {data}')
           plt.axis('off')

       create_cloud(1)
```



Rating 1

0.1.3 Cleaning Text

```
[107]: !pip install emoji
```

```
Requirement already satisfied: emoji in /usr/local/lib/python3.10/dist-packages
(2.8.0)
```

```
[108]: import string
       import emoji
       import re
       def cleaning_text(text):
           text = text.lower()
           text = re.sub(r'\d+','',text)
           text = emoji.demojize(text)
           text = ''.join([i for i in text if i not in string.punctuation])
           return text
       df['new_review'] = df['review_description'].apply(cleaning_text)
```

```
[109]: df.head()
```

```
[109]:
        source                                  review_description  rating  \
0  Google Play  Meh. Not the greatest experience on a Chromebo…        2
1  Google Play  Pretty good for a first launch!! Its easy to u…        3
2  Google Play  For a brand new app, it's very well optimized…        3
3  Google Play  Great app with a lot of potential! However, th…        3
4  Google Play  The app is good, but it needs a lot of functio…        3

          review_date                                          new_review
0 2023-07-08 14:18:24  meh not the greatest experience on a chromeboo…
1 2023-07-19 20:52:48  pretty good for a first launch its easy to use…
2 2023-07-06 23:03:11  for a brand new app its very well optimized ho…
3 2023-07-10 00:53:25  great app with a lot of potential however ther…
4 2023-07-06 16:57:43  the app is good but it needs a lot of function…
```

0.1.4 Stemming for finding word root

```
[110]: !pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages
(3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages
(from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages
(from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in
```

```
/usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from nltk) (4.66.1)
```

[111]:

```python
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer

ps = PorterStemmer()

words = word_tokenize(df['new_review'].iloc[5])
for w in words:
    print(w, ':', ps.stem(w))
```

```
currently : current
its : it
very : veri
challenging : challeng
to : to
use : use
its : it
in : in
dark : dark
mode : mode
and : and
i : i
want : want
to : to
change : chang
it : it
to : to
brighten : brighten
it : it
up : up
but : but
apparently : appar
i : i
can : can
only : onli
change : chang
it : it
through : through
instagram : instagram
why : whi
dark : dark
mode : mode
needs : need
```

```
to : to
be : be
improved : improv
its : it
tough : tough
on : on
the : the
eyes : eye
and : and
the : the
ui : ui
is : is
visually : visual
unbearable : unbear
twitter : twitter
nearly : nearli
perfected : perfect
the : the
ui : ui
i : i
think : think
its : it
a : a
decent : decent
start : start
but : but
i : i
cant : cant
use : use
it : it
for : for
more : more
than : than
a : a
couple : coupl
of : of
seconds : second
even : even
if : if
i : i
wanted : want
to : to
```

```
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[112]:

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[112]:

```
True
```

[113]:

```python
stemmed = []
for i in range(len(df)):
    stem_list = []
    words = word_tokenize(df['new_review'].iloc[i])
    for w in words:
        stem_list.append(ps.stem(w))
    stemmed.append(stem_list)

df['Stemmed'] = stemmed
df['Stemmed'] = df['Stemmed'].apply(' '.join)
```

[114]:

```python
from textblob import TextBlob
def sentiment_analysis(ds):
    sentiment = TextBlob(ds["Stemmed"]).sentiment
    return pd.Series([sentiment.subjectivity, sentiment.polarity])
```

[115]:

```python
df[["subjectivity", "polarity"]] = df.apply(sentiment_analysis, axis=1)
df.head()
```

[115]:

```
        source                               review_description  rating  \
0  Google Play  Meh. Not the greatest experience on a Chromebo…       2
1  Google Play  Pretty good for a first launch!! Its easy to u…       3
2  Google Play  For a brand new app, it's very well optimized…       3
3  Google Play  Great app with a lot of potential! However, th…       3
4  Google Play  The app is good, but it needs a lot of functio…       3

          review_date                                      new_review  \
0 2023-07-08 14:18:24  meh not the greatest experience on a chromeboo…
1 2023-07-19 20:52:48  pretty good for a first launch its easy to use…
2 2023-07-06 23:03:11  for a brand new app its very well optimized ho…
3 2023-07-10 00:53:25  great app with a lot of potential however ther…
4 2023-07-06 16:57:43  the app is good but it needs a lot of function…

                                             Stemmed  subjectivity  polarity
0  meh not the greatest experi on a chromebook se…      0.591667  0.145000
1  pretti good for a first launch it easi to use …      0.447619  0.330357
2  for a brand new app it veri well optim howev i…      0.574527  0.082670
3  great app with a lot of potenti howev there is…      0.237500  0.200000
4  the app is good but it need a lot of function …      0.544167  0.250833
```

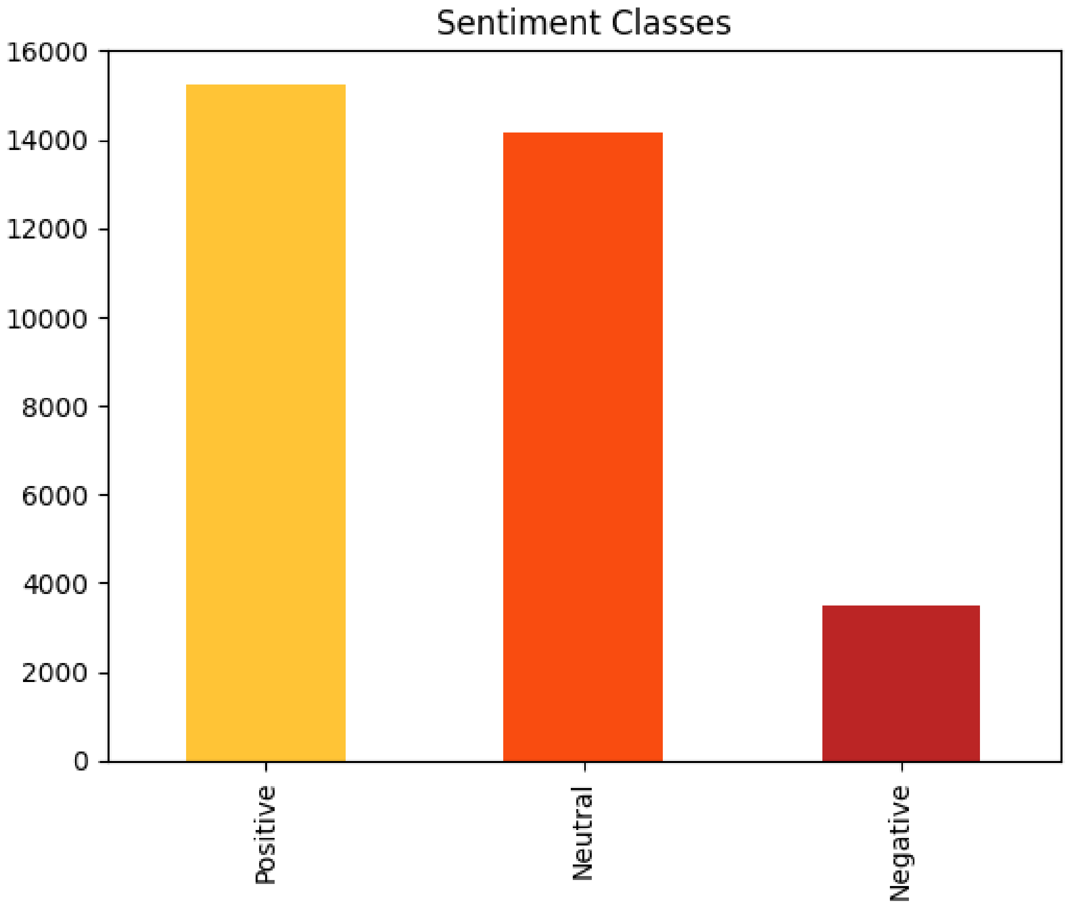
0.1.5 Sentiment Analysis

```
[116]: def analysis(score):
           if score < 0:
               return "Negative"
           elif score == 0:
               return "Neutral"
           else:
               return "Positive"
```

```
[117]: df["sentiment_analysis"] = df["polarity"].apply(analysis)
       df.head()
```

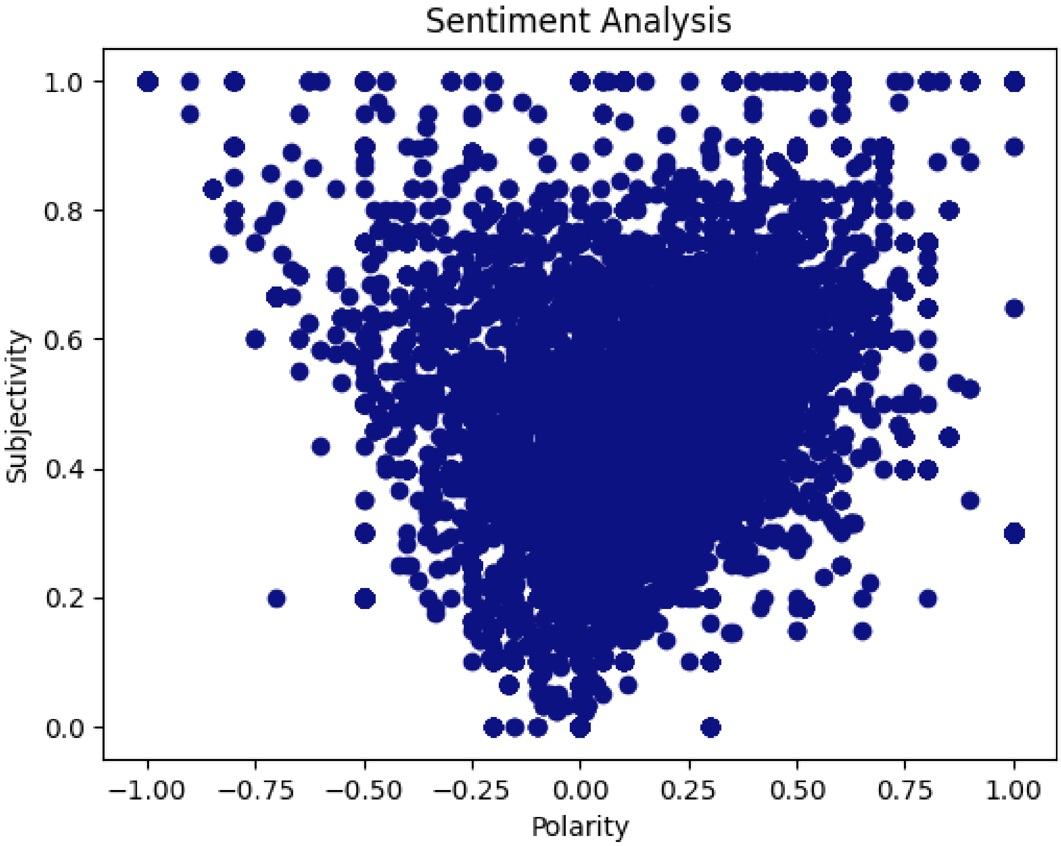
```
[117]:         source                                 review_description  rating  \
       0  Google Play  Meh. Not the greatest experience on a Chromebo…       2
       1  Google Play  Pretty good for a first launch!! Its easy to u…       3
       2  Google Play  For a brand new app, it's very well optimized…       3
       3  Google Play  Great app with a lot of potential! However, th…       3
       4  Google Play  The app is good, but it needs a lot of functio…       3

                   review_date                                          new_review  \
       0 2023-07-08 14:18:24  meh not the greatest experience on a chromeboo…
       1 2023-07-19 20:52:48  pretty good for a first launch its easy to use…
       2 2023-07-06 23:03:11  for a brand new app its very well optimized ho…
       3 2023-07-10 00:53:25  great app with a lot of potential however ther…
       4 2023-07-06 16:57:43  the app is good but it needs a lot of function…

                                                    Stemmed  subjectivity  polarity  \
       0  meh not the greatest experi on a chromebook se…      0.591667  0.145000
       1  pretti good for a first launch it easi to use …      0.447619  0.330357
       2  for a brand new app it veri well optim howev i…      0.574527  0.082670
       3  great app with a lot of potenti howev there is…      0.237500  0.200000
       4  the app is good but it need a lot of function …      0.544167  0.250833

         sentiment_analysis
       0           Positive
       1           Positive
       2           Positive
       3           Positive
       4           Positive
```

```
[118]: df.sentiment_analysis.value_counts().plot(kind='bar',␣
         ↪color=["#FFC436","#F94C10","#BB2525"])
       plt.title('Sentiment Classes')
       plt.show()
```



[119]:
```
plt.scatter(df.polarity, df.subjectivity, color='#0D1282')
plt.title('Sentiment Analysis')
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
```

[119]:
```
Text(0, 0.5, 'Subjectivity')
```


Sentiment Analysis
Subjectivity
Polarity
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00