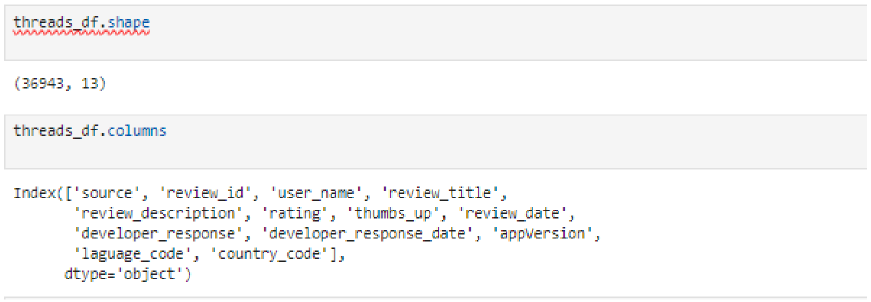
EXPLORATORY DATA ANALYSIS ON THREADS APP REVIEWS DATASET

Introduction: The purpose of this report is to present the findings of an exploratory data analysis (EDA) conducted on the Threads App Reviews dataset obtained from Kaggle. I tried to analyze the reviews of the Threads app, a social media app, to understand the users' feedback and how to improve user retention.
The Threads app is a camera-first messaging app that lets users share photos, videos, messages and stories with their close friends on Instagram. It was officially launched on July 5, 2023 and competes with other social media apps like Snapchat, WhatsApp and Messenger.The Threads App is a popular mobile application used for communication and collaboration among users. The dataset contains information about user reviews of the Threads App, including ratings, review text and other relevant details.

The main research questions are:

- What is the average rating of the product or service across all sources and reviews?
- How does the total rating vary by source? Is there a significant difference between the sources in terms of rating distribution?
- What are the most frequently mentioned words or phrases in the reviews?
- How does the rating change over time? Is there a trend or seasonality in the rating patterns?

Dataset Overview: The dataset used for this analysis is the Threads app reviews dataset from kaggle, which contains 36,943 reviews scrapped from Google Play store and Apple Store and collected between July 5 and August 5, 2023. The dataset has 13 columns.



```
threads_df.shape
```

```
(36943, 13)
```

```
threads_df.columns
```

```
Index(['source', 'review_id', 'user_name', 'review_title',
       'review_description', 'rating', 'thumbs_up', 'review_date',
       'developer_response', 'developer_response_date', 'appVersion',
       'laguage_code', 'country_code'],
      dtype='object')
```

Methodology: The methods and techniques used for this analysis include exploratory data analysis, content analysis and text mining. The tools used for this analysis include Python, matplotlib, pandas and NLTK. The analysis consists of the following steps:

1. **Data Pre-processing:** The raw data was prepared and transformed into a suitable format for this analysis. The steps involved include;

- **Data cleaning**: The dataset was examined for missing values, duplicates and inconsistencies. Data cleaning procedures were applied to ensure the integrity and quality of the dataset.

```
threads_df.isnull().sum()
```

```
source                      0
review_id                   0
user_name                   0
review_title            34943
review_description          0
rating                      0
thumbs_up                   0
review_date                 0
developer_response      36943
developer_response_date 36943
appVersion              12088
laguage_code                0
country_code                0
dtype: int64
```

Since some rows have so many null values and they do not affect our analysis, we will drop them.

```
#drop irrelevant columns
threads_df = threads_df.drop(['review_title', 'developer_response_date', 'developer_response', 'appVersion'], axis=1)
threads_df.head()
```

There were no duplicate values in the dataset. I also pre processed the data for our wordcloud analysis to show the most frequent/important words users used in their reviews.

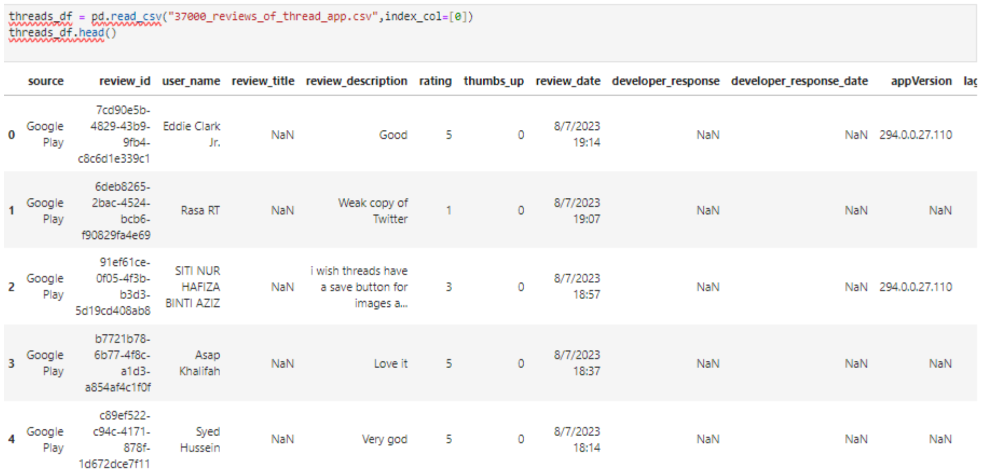
```
## pre process the text
def clean_the_code(text):
    text = text.lower()
    text = re.sub('[^a-z0-9]', ' ', text)  # Remove non-alphanumeric characters
    text = word_tokenize(text)

    stop_words = set(stopwords.words('english'))
    text = [word for word in text if word not in stop_words]  # Remove stopwords
    stemmer = SnowballStemmer("english")
    text = [stemmer.stem(word) for word in text]  # Stemming
    text = [word for word in text if len(word) > 1]  # Remove single characters
    return ' '.join(text)

# Assuming threads_df['review_description'] contains text data
threads_df['transformed_text'] = threads_df['review_description'].apply(clean_the_code)
threads_df.head()
```

- **Exploratory Data Analysis (EDA)**: Various statistical and graphical techniques were used to explore the dataset. This included summary statistics, distribution of ratings and word frequency analysis of review text. The steps incliude:

- **Import Libraries**:

```
# Import pandas.
import pandas as pd
import numpy as np
import re
import nltk
nltk.download('punkt')

# Import nltk stopwords.
nltk.download('stopwords')
from nltk.corpus import stopwords
from collections import Counter
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer

# Import wordcloud and matplotlib.
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import seaborn as sns
```

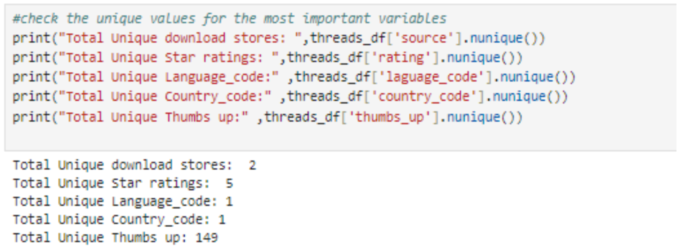
- **Load the dataset:**



```
threads_df = pd.read_csv("37000_reviews_of_thread_app.csv",index_col=[0])
threads_df.head()
```

	source	review_id	user_name	review_title	review_description	rating	thumbs_up	review_date	developer_response	developer_response_date	appVersion	lag
0	Google Play	7cd90e5b-4829-43b9-9fb4-c8c6d1e339c1	Eddie Clark Jr.	NaN	Good	5	0	8/7/2023 19:14	NaN	NaN	294.0.0.27.110	
1	Google Play	6deb8265-2bac-4524-bcb6-f90829fa4e69	Rasa RT	NaN	Weak copy of Twitter	1	0	8/7/2023 19:07	NaN	NaN	NaN	
2	Google Play	91ef61ce-0f05-4f3b-b3d3-5d19cd408ab8	SITI NUR HAFIZA BINTI AZIZ	NaN	i wish threads have a save button for images a...	3	0	8/7/2023 18:57	NaN	NaN	294.0.0.27.110	
3	Google Play	b7721b78-6b77-4f8c-a1d3-a854af4c1f0f	Asap Khalifah	NaN	Love it	5	0	8/7/2023 18:37	NaN	NaN	NaN	
4	Google Play	c89ef522-c94c-4171-878f-1d672dce7f11	Syed Hussein	NaN	Very god	5	0	8/7/2023 18:14	NaN	NaN	NaN	

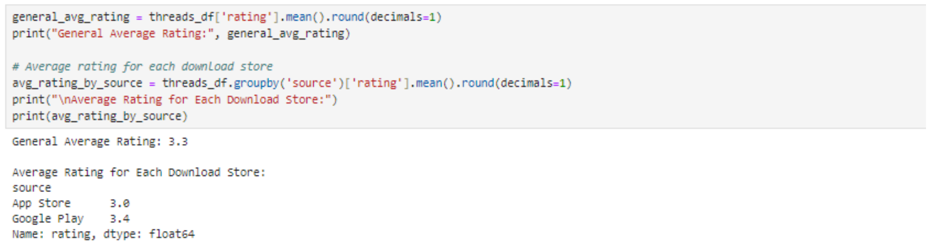
- **Check unique values:** I checked the unique values for the very important columns we will be using for this analysis. The language code and country code columns both have only a single value so both of the columns were dropped.



```
#check the unique values for the most important variables
print("Total Unique download stores: ",threads_df['source'].nunique())
print("Total Unique Star ratings: ",threads_df['rating'].nunique())
print("Total Unique Language_code:" ,threads_df['laguage_code'].nunique())
print("Total Unique Country_code:" ,threads_df['country_code'].nunique())
print("Total Unique Thumbs up:" ,threads_df['thumbs_up'].nunique())
```

```
Total Unique download stores:  2
Total Unique Star ratings:  5
Total Unique Language_code: 1
Total Unique Country_code: 1
Total Unique Thumbs up: 149
```

- **Check the average rating of the app:** I checked the general average ratings and also checked the average ratng of individual download source.

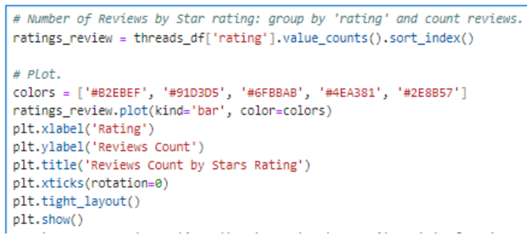


```
general_avg_rating = threads_df['rating'].mean().round(decimals=1)
print("General Average Rating:", general_avg_rating)

# Average rating for each download store
avg_rating_by_source = threads_df.groupby('source')['rating'].mean().round(decimals=1)
print("\nAverage Rating for Each Download Store:")
print(avg_rating_by_source)
```

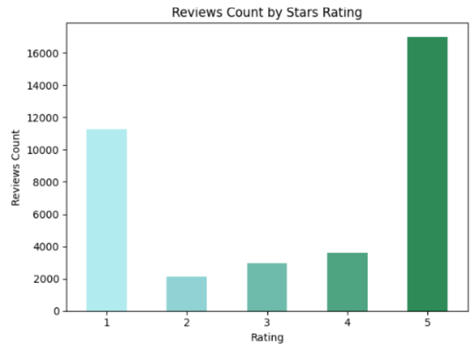
```
General Average Rating: 3.3

Average Rating for Each Download Store:
source
App Store      3.0
Google Play    3.4
Name: rating, dtype: float64
```

From the above, we can see that the average rating of the app is 3.3 out of 5, with Google Play having a higher average rating (3.4) than App Store (3.0).

- **Number of reviews by rating:** For each star rating category, we checked the total review.

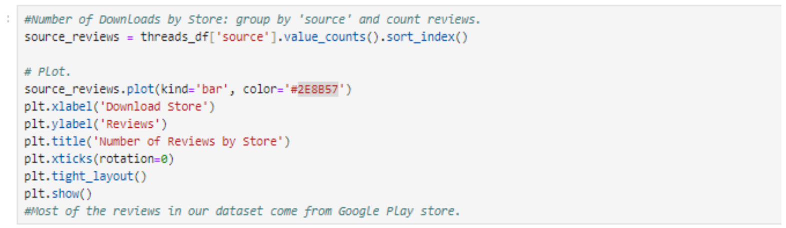


```
# Number of Reviews by Star rating: group by 'rating' and count reviews.
ratings_review = threads_df['rating'].value_counts().sort_index()

# Plot.
colors = ['#B2EBEF', '#91D3D5', '#6FBBAB', '#4EA381', '#2E8B57']
ratings_review.plot(kind='bar', color=colors)
plt.xlabel('Rating')
plt.ylabel('Reviews Count')
plt.title('Reviews Count by Stars Rating')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

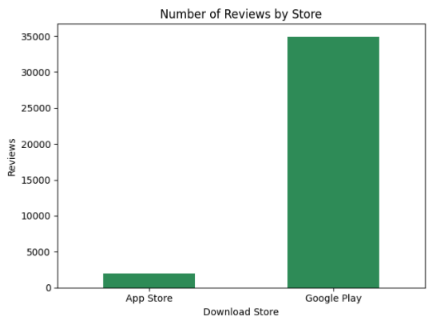


From the chart above, we can see that the reviews are almost polarized. 1 and 5 star ratings each have review counts above 10,000 while the 2, 3 and 4 star ratings are all below 4,000.

- **Check total number of reviews by download Store:**

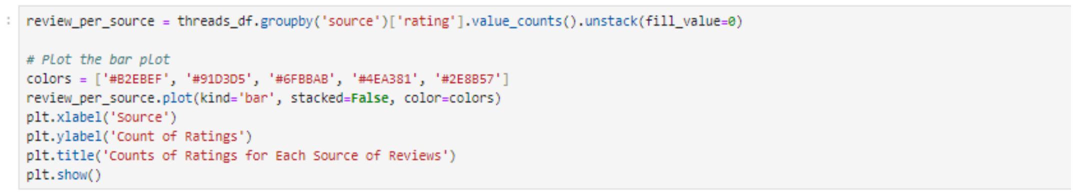


```
#Number of Downloads by Store: group by 'source' and count reviews.
source_reviews = threads_df['source'].value_counts().sort_index()

# Plot.
source_reviews.plot(kind='bar', color='#2E8B57')
plt.xlabel('Download Store')
plt.ylabel('Reviews')
plt.title('Number of Reviews by Store')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
#Most of the reviews in our dataset come from Google Play store.
```

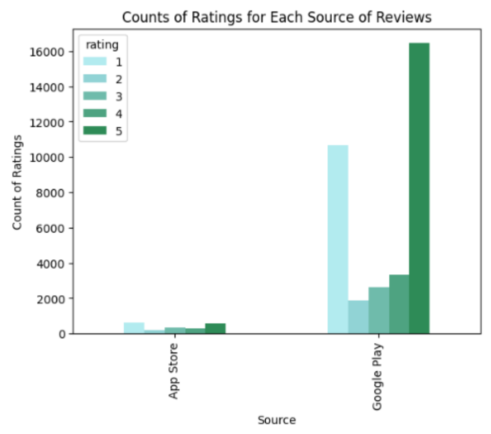


Majority of the reviews came from Google playstore.

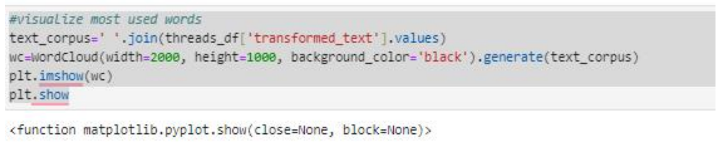
- **Check the Counts of Ratings for Each Source of Reviews:** I checked how many reviews have been given for each star rating category and for each source of reviews (such as Google Play or App Store). This can help you understand the distribution and frequency of ratings and reviews across different sources and categories.



```
review_per_source = threads_df.groupby('source')['rating'].value_counts().unstack(fill_value=0)

# Plot the bar plot
colors = ['#B2EBEF', '#91D3D5', '#6FBBAB', '#4EA381', '#2E8B57']
review_per_source.plot(kind='bar', stacked=False, color=colors)
plt.xlabel('Source')
plt.ylabel('Count of Ratings')
plt.title('Counts of Ratings for Each Source of Reviews')
plt.show()
```

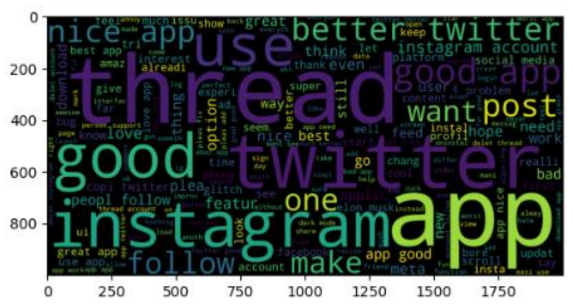


- **Create a wordcloud to visualize most used words in the reviews column:**



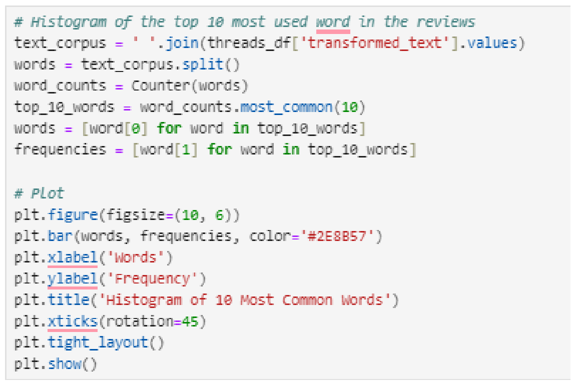
```
#visualize most used words
text_corpus=' '.join(threads_df['transformed_text'].values)
wc=WordCloud(width=2000, height=1000, background_color='black').generate(text_corpus)
plt.imshow(wc)
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

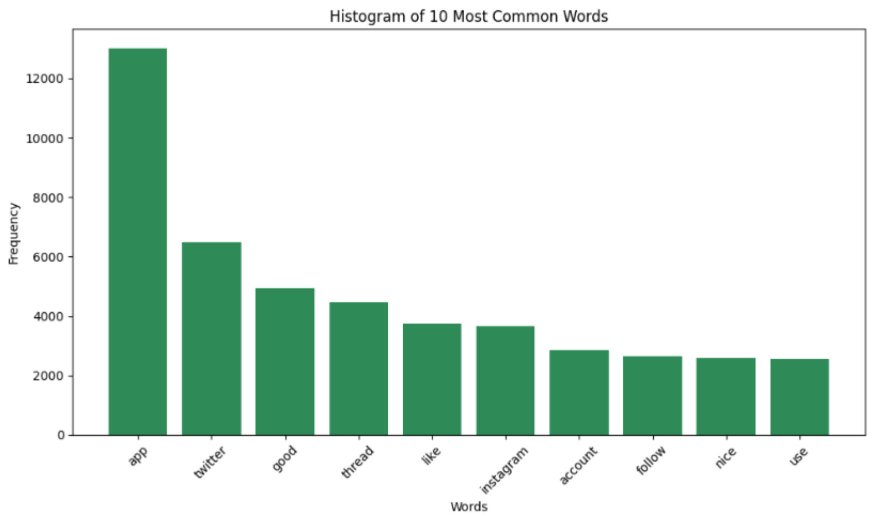


The wordcloud above indicates that there may be positive sentiment associated with the app due to the multiple mention of “Good”, “better” and “nice”. The presence of platform names like "twitter" and "Instagram" suggests that the Threads app may integrate or interact with these social media platforms. Users might be comparing cross-platform functionality or experiences.

- **Historgram of the top 10 common words:**

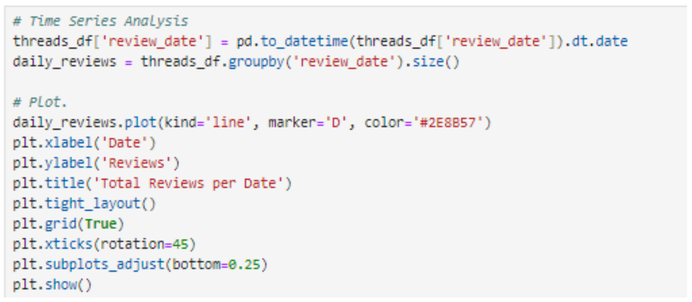


```
# Histogram of the top 10 most used word in the reviews
text_corpus = ' '.join(threads_df['transformed_text'].values)
words = text_corpus.split()
word_counts = Counter(words)
top_10_words = word_counts.most_common(10)
words = [word[0] for word in top_10_words]
frequencies = [word[1] for word in top_10_words]

# Plot
plt.figure(figsize=(10, 6))
plt.bar(words, frequencies, color='#2E8B57')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.title('Histogram of 10 Most Common Words')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

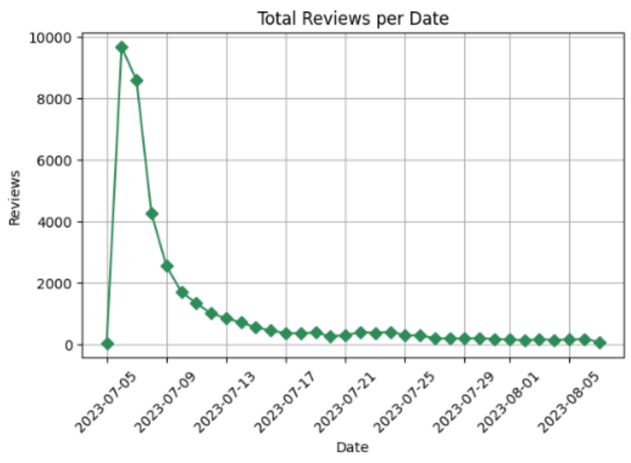


The frequent appearance of the word "app" suggests that users are referring to the Threads app itself. This could indicate discussions about its features, updates, functionalities, or troubleshooting. The presence of "twitter" in the top 10 indicates that users might be talking about features related to the integration of Threads with Twitter.

- **Time series analysis**: I did a time series analysis to visualize patterns, trends and relationship



```
# Time Series Analysis
threads_df['review_date'] = pd.to_datetime(threads_df['review_date']).dt.date
daily_reviews = threads_df.groupby('review_date').size()

# Plot.
daily_reviews.plot(kind='line', marker='D', color='#2E8B57')
plt.xlabel('Date')
plt.ylabel('Reviews')
plt.title('Total Reviews per Date')
plt.tight_layout()
plt.grid(True)
plt.xticks(rotation=45)
plt.subplots_adjust(bottom=0.25)
plt.show()
```



The high number of reviews on the first day suggests that there was significant anticipation and excitement surrounding the release of the Threads app. Users might have been eager to try out the app and share their initial impressions.

Observations:

1. The average rating is slightly above average (3.3), but there is a difference between the ratings from different sources. Google Play users tend to rate the product or service higher than Apple Store users, which is primarilly due to the number of reviews we had for the 2 sources.

2. We can see that the reviews are polarized, the users either love or hate the product or service and there are few users who have a moderate or neutral opinion. This could be that the app has some features or aspects that are very appealing to some users but very disappointing or frustrating to others.

3. The most frequent words in the reviews are "app", "twitter", "good", "threads" and "like".

4. The spike in reviews on the first day of release indicates strong initial engagement and interest in the app. This suggests that the app was able to capture users' attention during the launch phase.
5. The decline in reviews over time suggests a decrease in user engagement and interest in the app after the initial launch period.

Recommendations:

1. Strengthen integration with popular social media platforms like Twitter and Instagram to enhance user engagement and connectivity.

2. Conduct user surveys or feedback sessions to understand the reasons behind user disengagement and identify areas for improvement. Use this feedback to make targeted changes to the app that addresses user needs and preferences.

3. Conduct regular updates and bug fixes to address technical issues and improve app performance.

Conclusion: The EDA of the Threads App Reviews dataset has provided valuable insights into common issues and areas for improvement. Overall, the Threads App enjoys a positive reputation among its users, but there are opportunities for further enhancement in terms of technical performance, feature development and customer support. The findings of this analysis can inform strategic decision-making and product development efforts aimed at enhancing user satisfaction and retention.