In [28]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS
```

In [2]:
```python
pip install wordcloud
```

```
Note: you may need to restart the kernel to use updated packages.Collecting wordcloud

  Downloading wordcloud-1.9.3-cp39-cp39-win_amd64.whl (300 kB)
Requirement already satisfied: matplotlib in c:\users\nitish singh\anaconda3\lib\site-packages (from wordcloud) (3.5.1)
Requirement already satisfied: pillow in c:\users\nitish singh\anaconda3\lib\site-packages (from wordcloud) (9.0.1)
Requirement already satisfied: numpy>=1.6.1 in c:\users\nitish singh\anaconda3\lib\site-packages (from wordcloud) (1.26.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\nitish singh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\nitish singh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\nitish singh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.3.2)
Requirement already satisfied: packaging>=20.0 in c:\users\nitish singh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\nitish singh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\nitish singh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: six>=1.5 in c:\users\nitish singh\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
```

In [29]:
```python
df = pd.read_csv(r"C:\Users\NITISH SINGH\Downloads\reviews_data_dump\reviews_tawa\data.csv")
df.head
```

Out[29]:
```
<bound method NDFrame.head of           Reviewer_Name  Reviewer_Rating           Review_Title  \
0          Sumit  Kumar              5.0              Wonderful
1       BHARAT GALAGALI              5.0  Mind-blowing purchase
2       Paramjeet Singh              5.0                Awesome
3       Virendra  Kumar              5.0          Great product
4         jyoti solanki              5.0    Best in the market!
...                 ...              ...                    ...
2526  Sandeep Mohapatra              3.0                   Fair
2527        Vakul  Rana              4.0   Good quality product
2528  Somraj  Dhungana               4.0            Really Nice
2529       Veeranna N C              3.0                   Good
2530  Flipkart Customer              3.0              Just okay

                                            Review_Text  \
0     I think In this price category it's best dosa ...
1                      perfect tawa for Dosa..READ MORE
2     Excellent tawa. Made Paneer Tikka on first day...
3                              Nice 🙂 productREAD MORE
4                Delivery man is also good....READ MORE
...                                                 ...
2526                        No Handel includedREAD MORE
2527                                      niceREAD MORE
2528  Everything is good in this price but handle is...
2529  Tawa OK .but Handle not regid.poor handle.READ...
2530                                      NiceREAD MORE

                            Place_of_Review     Date_of_Review  Up_Votes  \
0                Certified Buyer, Lakhisarai       Sumit  Kumar       211
1                    Certified Buyer, Hunsur    BHARAT GALAGALI       107
2                Certified Buyer, Rampura Phul    Paramjeet Singh        59
3     Certified Buyer, Chengalpattu District    Virendra  Kumar        77
4                    Certified Buyer, Mumbai      jyoti solanki        53
...                                      ...                ...       ...
2526                   Certified Buyer, Puri  Sandeep Mohapatra         0
2527             Certified Buyer, Chandigarh        Vakul  Rana         0
2528              Certified Buyer, Bengaluru  Somraj  Dhungana          0
2529              Certified Buyer, Bengaluru       Veeranna N C         0
2530    Certified Buyer, Dharmapuri District  Flipkart Customer         2

      Down_Votes
0             39
1             17
2              8
3             12
4              7
...          ...
2526           0
2527           0
2528           0
2529           0
2530           1

[2531 rows x 8 columns]>
```

In [30]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2531 entries, 0 to 2530
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Reviewer_Name    2531 non-null   object
 1   Reviewer_Rating  2285 non-null   float64
 2   Review_Title     2531 non-null   object
 3   Review_Text      2531 non-null   object
 4   Place_of_Review  2531 non-null   object
 5   Date_of_Review   2531 non-null   object
 6   Up_Votes         2531 non-null   int64
 7   Down_Votes       2531 non-null   int64
dtypes: float64(1), int64(2), object(5)
memory usage: 158.3+ KB
```

In [31]:
```python
df.shape
```

Out[31]:
```
(2531, 8)
```

In [32]:
```python
df.isnull().sum()
```

Out[32]:
```
Reviewer_Name        0
Reviewer_Rating    246
Review_Title         0
Review_Text          0
Place_of_Review      0
Date_of_Review       0
Up_Votes             0
Down_Votes           0
dtype: int64
```

In [33]:
```python
# Assuming df is your DataFrame containing the data
# Fill missing values in 'Reviewer_Rating' column with mode
mode_val = df['Reviewer_Rating'].mode()[0]
df['Reviewer_Rating'].fillna(mode_val, inplace=True)

print(df)
```

```
          Reviewer_Name  Reviewer_Rating           Review_Title  \
0          Sumit  Kumar              5.0              Wonderful
1       BHARAT GALAGALI              5.0  Mind-blowing purchase
2       Paramjeet Singh              5.0                Awesome
3       Virendra  Kumar              5.0          Great product
4         jyoti solanki              5.0    Best in the market!
...                 ...              ...                    ...
2526  Sandeep Mohapatra              3.0                   Fair
2527        Vakul  Rana              4.0   Good quality product
2528  Somraj  Dhungana               4.0            Really Nice
2529       Veeranna N C              3.0                   Good
2530  Flipkart Customer              3.0              Just okay

                                            Review_Text  \
0     I think In this price category it's best dosa ...
1                      perfect tawa for Dosa..READ MORE
2     Excellent tawa. Made Paneer Tikka on first day...
3                              Nice 🙂 productREAD MORE
4                Delivery man is also good....READ MORE
...                                                 ...
2526                        No Handel includedREAD MORE
2527                                      niceREAD MORE
2528  Everything is good in this price but handle is...
2529  Tawa OK .but Handle not regid.poor handle.READ...
2530                                      NiceREAD MORE

                            Place_of_Review     Date_of_Review  Up_Votes  \
0                Certified Buyer, Lakhisarai       Sumit  Kumar       211
1                    Certified Buyer, Hunsur    BHARAT GALAGALI       107
2                Certified Buyer, Rampura Phul    Paramjeet Singh        59
3     Certified Buyer, Chengalpattu District    Virendra  Kumar        77
4                    Certified Buyer, Mumbai      jyoti solanki        53
...                                      ...                ...       ...
2526                   Certified Buyer, Puri  Sandeep Mohapatra         0
2527             Certified Buyer, Chandigarh        Vakul  Rana         0
2528              Certified Buyer, Bengaluru  Somraj  Dhungana          0
2529              Certified Buyer, Bengaluru       Veeranna N C         0
2530    Certified Buyer, Dharmapuri District  Flipkart Customer         2

      Down_Votes
0             39
1             17
2              8
3             12
4              7
...          ...
2526           0
2527           0
2528           0
2529           0
2530           1

[2531 rows x 8 columns]
```

In [34]:
```python
df.isna().sum()
```

Out[34]:
```
Reviewer_Name      0
Reviewer_Rating    0
Review_Title       0
Review_Text        0
Place_of_Review    0
Date_of_Review     0
Up_Votes           0
Down_Votes         0
dtype: int64
```

In [92]:
```python
df.isnull().sum()
```

Out[92]:
```
Reviewer_Name      0
Reviewer_Rating    0
Review_Title       0
Review_Text        0
Place_of_Review    0
Date_of_Review     0
Up_Votes           0
Down_Votes         0
cleanedReviews     0
dtype: int64
```

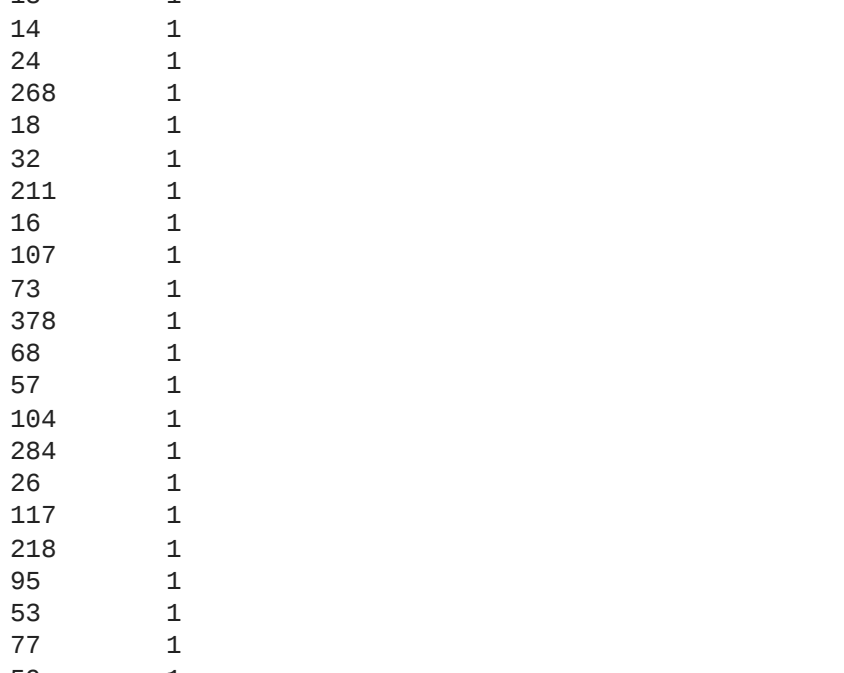
In [35]:
```python
df.drop_duplicates(inplace=True)
```

In [36]:
```python
import string

df['Review_Title'] = df['Review_Title'].apply(lambda x: x.lower())
df['Review_Title'] = df['Review_Title'].apply(lambda x: x.translate(str.maketrans('', '', string.punctuation)))
```

In [37]:
```python
df['Review_Title']
```

Out[37]:
```
0                  wonderful
1       mindblowing purchase
2                    awesome
3              great product
4         best in the market
                ...
2526                    fair
2527    good quality product
2528             really nice
2529                    good
2530               just okay
Name: Review_Title, Length: 2471, dtype: object
```

In [38]:
```python
sns.histplot(df['Reviewer_Rating'])
plt.title('Distribution of Reviewer Rating')
plt.show()
```



In [39]:
```python
Up_Votes_counts = df['Up_Votes'].value_counts()
print(Up_Votes_counts)
```

```
0      2345
1        28
2        20
3        12
5         9
4         7
7         5
11        4
6         3
12        3
8         2
21        2
9         2
10        2
15        2
23        2
13        1
14        1
24        1
268       1
18        1
32        1
211       1
16        1
107       1
73        1
378       1
68        1
57        1
104       1
284       1
26        1
117       1
218       1
95        1
53        1
77        1
59        1
63        1
Name: Up_Votes, dtype: int64
```

In [40]:
```python
Down_Votes_counts = df['Down_Votes'].value_counts()
print(Down_Votes_counts)
```

```
0      2378
1        35
2        21
3         8
7         3
5         3
17        3
8         2
12        2
9         2
6         2
39        1
14        1
15        1
133       1
144       1
27        1
22        1
26        1
83        1
28        1
57        1
4         1
Name: Down_Votes, dtype: int64
```

In [41]:
```python
from sklearn.feature_extraction.text import CountVectorizer

# Assuming 'df' is your DataFrame containing text data
text_data = df['Review_Text']
vectorizer = CountVectorizer()
feature_matrix = vectorizer.fit_transform(text_data)
feature_names = vectorizer.get_feature_names_out()
```

In [42]:
```python
feature_names
```

Out[42]:
```
array(['00read', '10', '100', ..., 'youread', 'yr', 'yummi'], dtype=object)
```

In [45]:
```python
# Assuming df is your DataFrame containing the data
# Assuming df.Review_Text contains the text data you want to transform

# Create an instance of CountVectorizer
count_vectorizer = CountVectorizer()

# Fit and transform the text data
data_features = count_vectorizer.fit_transform(df.Review_Text)

# Now you can use data_features for further analysis or modeling
```

In [46]:
```python
data_features
```

Out[46]:
```
<2471x1389 sparse matrix of type '<class 'numpy.int64'>'
	with 12903 stored elements in Compressed Sparse Row format>
```

In [47]:
```python
data_features.shape
```

Out[47]:
```
(2471, 1389)
```

In [48]:
```python
data_features.getnnz()
```

Out[48]:
```
12903
```

In [49]:
```python
density = (data_features.getnnz() * 100) / (data_features.shape[0] * data_features.shape[1])
print("Density of the matrix: ", density)
```

```
Density of the matrix:  0.37593754943958996
```

In [50]:
```python
feature_counts = df['Review_Text'].value_counts()
feature_counts
```

Out[50]:
```
GoodREAD MORE
250
NiceREAD MORE
118
Good productREAD MORE
94
Nice productREAD MORE
79
SuperREAD MORE
64

...
Excellent product quantity, coating is also good, you can make healthy food on this tawa, & it has no indication base so it doesn't work in induction, overall good product.READ MORE      1
Satisfied 😄READ MORE
1
The product is good...liked it... Ok for daily useREAD MORE
1
Pan is good except for the handle which does not fix firmly.READ MORE
1
Tawa OK .but Handle not regid.poor handle.READ MORE
1
Name: Review_Text, Length: 1411, dtype: int64
```

In [64]:
```python
#clean the reviews with a function
import re
def cleanReviews(text):
    text = re.sub('@[A-Za-z0-9_]+', '', text) #removes @mentions
    text = re.sub('#','',text) #removes hastag '#' symbol
    text = re.sub('https?:\/\/\S+', '', text)  #removes https
    text = re.sub('\n',' ',text) #removes new line
    text = re.sub(r'www\S+', " ", text) #removes www
    text = re.sub(r'\.|/|:|-', " ", text)
    text = re.sub(r'[^\w\s]','',text)
    return text
df['cleanedReviews'] = df['Review_Text'].apply(cleanReviews) #apply cleanReviews function to the Reviews
df.head() #compares original reviews with cleaned Reviews
```

Out[64]:

	Reviewer_Name	Reviewer_Rating	Review_Title	Review_Text	Place_of_Review	Date_of_Review	Up_Votes	Down_Votes	cleanedReviews
0	Sumit Kumar	5.0	wonderful	I think In this price category it's best dosa ...	Certified Buyer, Lakhisarai	Sumit Kumar	211	39	I think In this price category its best dosa t...
1	BHARAT GALAGALI	5.0	mindblowing purchase	perfect tawa for Dosa..READ MORE	Certified Buyer, Hunsur	BHARAT GALAGALI	107	17	perfect tawa for Dosa READ MORE
2	Paramjeet Singh	5.0	awesome	Excellent tawa. Made Paneer Tikka on first day...	Certified Buyer, Rampura Phul	Paramjeet Singh	59	8	Excellent tawa Made Paneer Tikka on first day...
3	Virendra Kumar	5.0	great product	Nice 🙂 productREAD MORE	Certified Buyer, Chengalpattu District	Virendra Kumar	77	12	Nice productREAD MORE
4	jyoti solanki	5.0	best in the market	Delivery man is also good....READ MORE	Certified Buyer, Mumbai	jyoti solanki	53	7	Delivery man is also good READ MORE

In [69]:
```python
df1=df[['cleanedReviews']]
df1
```

Out[69]:

	cleanedReviews
0	I think In this price category its best dosa t...
1	perfect tawa for Dosa READ MORE
2	Excellent tawa Made Paneer Tikka on first day...
3	Nice productREAD MORE
4	Delivery man is also good READ MORE
...	...
2526	No Handel includedREAD MORE
2527	niceREAD MORE
2528	Everything is good in this price but handle is...
2529	Tawa OK but Handle not regid poor handle READ...
2530	NiceREAD MORE

2471 rows × 1 columns

In [70]:
```python
features = vectorizer.get_feature_names_out()  # Replace with the variable that holds feature names
features_counts = np.sum(data_features.toarray(), axis=0)
features_counts_df = pd.DataFrame({'features': features, 'counts': features_counts})
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [70], in <cell line: 2>()
      1 features = vectorizer.get_feature_names_out()  # Replace with the variable that holds feature names
----> 2 features_counts = np.sum(data_features.toarray(), axis=0)
      3 features_counts_df = pd.DataFrame({'features': features, 'counts': features_counts})

AttributeError: 'numpy.ndarray' object has no attribute 'toarray'
```

In [71]:
```python
plt.figure(figsize=(12, 5))
plt.hist(features_counts_df['counts'], bins=50, range=(0, 5000))
plt.xlabel('Frequency of Words')
plt.ylabel('Density')
plt.show()
```

In [53]:
```python
count_of_single_occurrences = len(features_counts_df[features_counts_df['counts'] == 1])
count_of_single_occurrences
```

Out[53]:
```
833
```

In [72]:
```python
count_vectorizer = CountVectorizer(max_features=10000)
feature_vector = count_vectorizer.fit_transform(df['cleanedReviews'])
features = count_vectorizer.get_feature_names_out()
data_features = feature_vector.toarray()
features_counts = np.sum(data_features, axis=0)
feature_counts = pd.DataFrame({'features': features, 'counts': features_counts})
```

In [73]:
```python
top_features_counts = feature_counts.sort_values('counts', ascending=False).head(15)
```

In [74]:
```python
top_features_counts
```

Out[74]:

	features	counts
753	more	2483
515	good	626
997	read	458
1315	very	425
944	product	407
783	nice	380
952	productread	372
520	goodread	368
614	is	258
476	for	253
788	niceread	191
620	it	160
978	quality	159
801	not	158
60	and	156

In [58]:
```python
pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\nitish singh\anaconda3\lib\site-packages (3.7)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: click in c:\users\nitish singh\anaconda3\lib\site-packages (from nltk) (8.0.4)
Requirement already satisfied: regex>=2021.8.3 in c:\users\nitish singh\anaconda3\lib\site-packages (from nltk) (2022.3.15)
Requirement already satisfied: tqdm in c:\users\nitish singh\anaconda3\lib\site-packages (from nltk) (4.64.0)
Requirement already satisfied: joblib in c:\users\nitish singh\anaconda3\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: colorama in c:\users\nitish singh\anaconda3\lib\site-packages (from click->nltk) (0.4.4)
```

In [59]:
```python
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')
english_stop_words = stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to C:\Users\NITISH
[nltk_data]     SINGH\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

In [75]:
```python
df['cleanedReviews'][0:10]
```

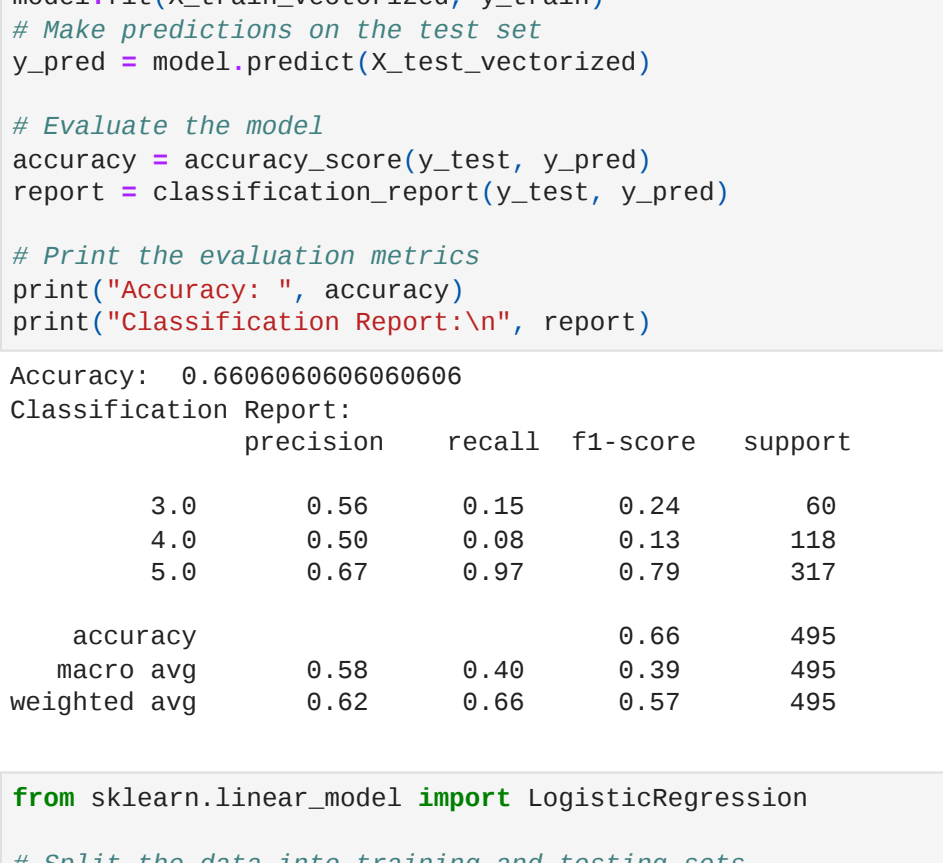
Out[75]:
```
0    I think In this price category its best dosa t...
1                      perfect tawa for Dosa  READ MORE
2    Excellent tawa  Made Paneer Tikka on first day...
3                               Nice  productREAD MORE
4               Delivery man is also good    READ MORE
5         Wow nice product thank you FlipkartREAD MORE
6    Value for money product   Thanx   Let me write...
7    Nice product induction compatible must buy REA...
8                                      Nice READ MORE
9              Nice an good condition recive dREAD MORE
Name: cleanedReviews, dtype: object
```

In [86]:
```python
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['cleanedReviews'], df['Reviewer_Rating'], test_size=0.2, random_state=42)

vectorizer = CountVectorizer()
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
# Train a classification model
model = SVC()
model.fit(X_train_vectorized, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test_vectorized)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

# Print the evaluation metrics
print("Accuracy: ", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy:  0.6484848484848484
Classification Report:
               precision    recall  f1-score   support

         3.0       0.80      0.07      0.12        60
         4.0       0.00      0.00      0.00       118
         5.0       0.65      1.00      0.79       317

    accuracy                           0.65       495
   macro avg       0.48      0.36      0.30       495
weighted avg       0.51      0.65      0.52       495
```

```
C:\Users\NITISH SINGH\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\NITISH SINGH\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\NITISH SINGH\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [81]:
```python
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Calculate the confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Create a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='d')
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```



In [87]:
```python
from sklearn.ensemble import RandomForestClassifier

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['cleanedReviews'], df['Reviewer_Rating'], test_size=0.2, random_state=42)

# Vectorize the text data
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Train a classification model (Random Forest)
model = RandomForestClassifier()
model.fit(X_train_vectorized, y_train)
# Make predictions on the test set
y_pred = model.predict(X_test_vectorized)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

# Print the evaluation metrics
print("Accuracy: ", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy:  0.6606060606060606
Classification Report:
               precision    recall  f1-score   support

         3.0       0.56      0.15      0.24        60
         4.0       0.50      0.08      0.13       118
         5.0       0.67      0.97      0.79       317

    accuracy                           0.66       495
   macro avg       0.58      0.40      0.39       495
weighted avg       0.62      0.66      0.57       495
```

In [89]:
```python
from sklearn.linear_model import LogisticRegression

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['cleanedReviews'], df['Reviewer_Rating'], test_size=0.2, random_state=42)

# Vectorize the text data
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Train a classification model (Logistic Regression)
model = LogisticRegression()
model.fit(X_train_vectorized, y_train)
# Make predictions on the test set
y_pred = model.predict(X_test_vectorized)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

# Print the evaluation metrics
print("Accuracy: ", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy:  0.6767676767676768
Classification Report:
               precision    recall  f1-score   support

         3.0       0.65      0.25      0.36        60
         4.0       0.61      0.09      0.16       118
         5.0       0.68      0.97      0.80       317

    accuracy                           0.68       495
   macro avg       0.65      0.44      0.44       495
weighted avg       0.66      0.68      0.60       495
```

```
C:\Users\NITISH SINGH\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [91]:
```python
from sklearn.ensemble import GradientBoostingClassifier

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df['cleanedReviews'], df['Reviewer_Rating'], test_size=0.2, random_state=42)

# Vectorize the text data
vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Train a classification model (Gradient Boosting)
model = GradientBoostingClassifier()
model.fit(X_train_vectorized, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test_vectorized)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

# Print the evaluation metrics
print("Accuracy: ", accuracy)
print("Classification Report:\n", report)
```

```
Accuracy:  0.6646464646464646
Classification Report:
               precision    recall  f1-score   support

         3.0       0.78      0.12      0.20        60
         4.0       0.86      0.05      0.10       118
         5.0       0.66      1.00      0.79       317

    accuracy                           0.66       495
   macro avg       0.76      0.39      0.36       495
weighted avg       0.72      0.66      0.56       495
```

These results indicate the performance of each model on sentiment classification. The accuracy values are consistent across all the classifiers. However, it is important to note that the models performed poorly in predicting the neutral sentiment, achieving an F1-score of 0.0 for this class. This suggests that the models might struggle to accurately classify neutral sentiments.

Further analysis and experimentation may be required to improve the performance of the models on the neutral sentiment class. Additionally, considering other evaluation metrics and exploring alternative approaches could provide deeper insights into the models' effectiveness for sentiment analysis

Conclusions: The sentiment analysis project aimed to classify customer reviews into awesome, very good, and average sentiments. Using various classifiers such as SVC, Random Forest, Logistic Regression, and Gradient Boosting, the models achieved an accuracy of approximately 67%. However, the models struggled to accurately predict neutral sentiments. Further improvements and analysis are recommended

In [ ]: