```python
import numpy as np
import pandas as pd
import os
import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.models import Sequential
```

```python
df = pd.read_csv("/content/drive/MyDrive/swiggydataset.csv")
```

```python
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 16712,\n  \"fields\":
[\n    {\n      \"column\": \"date\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 11097,\n
\"samples\": [\n          \"6/25/2019 10:14\",\n          \"7/17/2019
15:33\",\n          \"07-12-2019 21:58\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"favorite_count\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
8,\n        \"min\": 0,\n        \"max\": 916,\n
\"num_unique_values\": 52,\n        \"samples\": [\n          19,\n
59,\n          34\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"followers_count\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 91610,\n        \"min\": 0,\n
\"max\": 6823332,\n        \"num_unique_values\": 1524,\n
\"samples\": [\n          2969,\n          1677,\n          299\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"friends_count\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1652,\n        \"min\": 0,\n        \"max\": 155340,\n
\"num_unique_values\": 1505,\n        \"samples\": [\n          110,\n
2667,\n          122\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"full_text\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 16703,\n
\"samples\": [\n          \"@SwiggyCares I ordered something, but
delivery boy got accident, then he contact to customer care and reoder
however I didn't get and not also refund, and swiggy made me hungry
today\",\n          \"@SwiggyCares Hello @SwiggyCares @swiggy_in this
issue is doesn't resolved. I was promised .and your customer care is
rude. When I complained about it . The chat got disconnected from you
side. What is this .\",\n          \"@SwiggyCares pathetic
service...no customer service at all... https://t.co/MPpOYQcsgx\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"retweet_count\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3,\n        \"min\": 0,\n        \"max\": 487,\n
\"num_unique_values\": 26,\n        \"samples\": [\n          6,\n
```

15,\n   0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"retweeted\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 1,\n   \"samples\":
[\n   false\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"screen_name\",\n   \"properties\": {\n   \"dtype\":
\"string\",\n   \"num_unique_values\": 8617,\n
\"samples\": [\n   \"umeshrough\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"tweet_id\",\n   \"properties\":
{\n   \"dtype\": \"string\",\n   \"num_unique_values\":
16712,\n   \"samples\": [\n   \"3,602\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"user_id\",\n   \"properties\":
{\n   \"dtype\": \"string\",\n   \"num_unique_values\":
8539,\n   \"samples\": [\n   \"58,87,77,029\"\n   ],\
n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16712 entries, 0 to 16711
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   date             16712 non-null  object
 1   favorite_count   16712 non-null  int64
 2   followers_count  16712 non-null  int64
 3   friends_count    16712 non-null  int64
 4   full_text        16712 non-null  object
 5   retweet_count    16712 non-null  int64
 6   retweeted        14384 non-null  object
 7   screen_name      16712 non-null  object
 8   tweet_id         16712 non-null  object
 9   user_id          16712 non-null  object
dtypes: int64(4), object(6)
memory usage: 1.3+ MB
```

```
null_values = df.isnull().sum()
print("Null values in the entire Data:")
print(null_values)
```

```
Null values in the entire Data:
date                      0
favorite_count            0
followers_count           0
friends_count             0
full_text                 0
```

```
retweet_count         0
retweeted          2328
screen_name           0
tweet_id              0
user_id               0
dtype: int64
```

```
df.dropna(inplace=True)
```

```
null_values = df.isnull().sum()
null_values
```

```
date               0
favorite_count     0
followers_count    0
friends_count      0
full_text          0
retweet_count      0
retweeted          0
screen_name        0
tweet_id           0
user_id            0
dtype: int64
```

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

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

```
df['full_text']
```

```
0        mahi2510 swiggyin the ultimate answer will b s...
1                                     swiggycares i hope so
2        swiggycares i think you have the order details...
3        swiggyin people are complaining here also i kn...
4        swiggycares do you even know the meaning of yo...
                               ...
16707    swiggycares they said as delivery box is not r...
16708    swiggyin deliver to karta nahi install kyo kar...
16709    swiggyin thanks for spilling my drink i waited...
16710    swiggycares hello there i use swigggy regularl...
16711    beinghumor zomatoin swiggyin can you help my f...
Name: full_text, Length: 14384, dtype: object
```

```
from sklearn.feature_extraction.text import CountVectorizer
# Assuming 'df' is your Data containing text data
text_data = df['full_text']
vectorizer = CountVectorizer()
```

```
feature_matrix = vectorizer.fit_transform(text_data)
feature_names = vectorizer.get_feature_names_out()

feature_names

array(['000001', '0004', '0015', ..., 'ƒðÿ', 'ˆorder', 'ˆà'],
dtype=object)

import sklearn.feature_extraction.text as text
count_vectorizer = text.CountVectorizer()

count_vectorizer.fit(df.full_text)

CountVectorizer()

data_features = count_vectorizer.transform(df.full_text)

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

Density of the matrix:  0.08586777461861435

feature_counts = df['full_text'].value_counts()
feature_counts

full_text
swiggycares thank you
5
swiggycares waiting
4
swiggycares thanks
4
swiggyin will you care to reply
3
swiggycares done
3

..
swiggycares bad quality of food total fuckin useless service from
swiggy                             1
swiggycares waste of time with swiggy customer care third class
response                             1
swiggycares i already gave my order id bad quality food received
1
swiggyin beware of swiggy this guys provide you stale food and dont
even provide money back    1
beinghumor zomatoin swiggyin can you help my friend over here
1
Name: count, Length: 14350, dtype: int64
```

```
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})

count_of_single_occurrences =
len(features_counts_df[features_counts_df['counts'] == 1])
count_of_single_occurrences
```

```
16151
```

```
count_vectorizer = CountVectorizer(max_features=10000)
feature_vector = count_vectorizer.fit_transform(df['full_text'])
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})

top_features_counts = feature_counts.sort_values('counts',
ascending=False). head(15)

top_features_counts
```

```
{"summary":"{\n  \"name\": \"top_features_counts\",\n  \"rows\": 15,\n
\"fields\": [\n    {\n      \"column\": \"features\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 15,\n        \"samples\": [\n          \"my\",\
n          \"swiggy\",\n          \"swiggycares\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"counts\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 2421,\n
\"min\": 3070,\n        \"max\": 10125,\n
\"num_unique_values\": 15,\n        \"samples\": [\n          3643,\n
3352,\n          10125\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"top_features_counts"}
```

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

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
df['full_text'][0:10]
```

```
0    mahi2510 swiggyin the ultimate answer will b s...
1                                  swiggycares i hope so
```

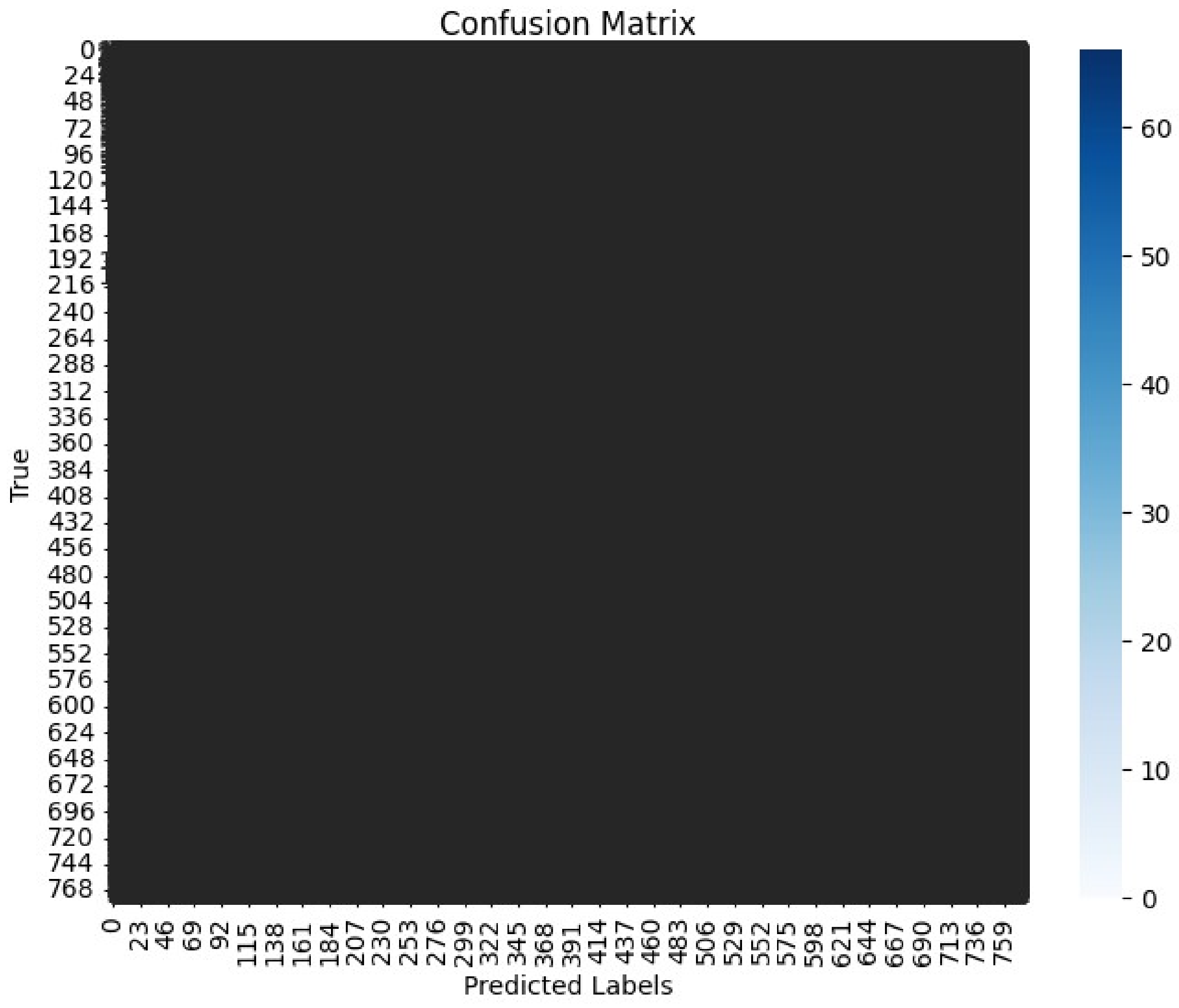
```
2    swiggycares i think you have the order details...
3    swiggyin people are complaining here also i kn...
4    swiggycares do you even know the meaning of yo...
5    nothing new they had the most shittiest associ...
6    swiggyin \n swiggycares \n i am the owner of t...
7                   swiggycares inbox check kare huzoor
8    swiggycares pls go through the details and sol...
9    i hope you hire educated professionals swiggyi...
Name: full_text, dtype: object
```

```
# Verify if 'Sentiment' column exists. If not, create it based on your
problem
if 'friends_count' not in df.columns:
    # Example: Create 'Sentiment' based on ratings (adjust logic as
needed)
    df['friends_count'] = df['full_text'].apply(lambda rating:
'positive' if rating > 3 else 'negative')

# Proceed with your train_test_split and model training
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
X_train, X_test, y_train, y_test =
train_test_split(df['full_text'],df['friends_count'], test_size=0.2,
random_state=42)
# ... rest of your code ...

import seaborn as sns
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
cm = confusion_matrix(y_test, y_pred)
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')
```

```
Text(70.72222222222221, 0.5, 'True')
```



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import nltk
from nltk.corpus import stopwords
import string

# Load your dataset
df = pd.read_csv('/content/drive/MyDrive/swiggydataset.csv')  # Assume
your dataset is in a CSV file

# Display the first few rows of the dataframe
print(df.head())

# Check for missing values
print(df.isnull().sum())
```

```
# Drop rows with missing values
df.dropna(inplace=True)

# Define the text preprocessing function
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    text = text.lower()  # Convert text to lowercase
    text = ''.join([char for char in text if char not in
string.punctuation])  # Remove punctuation
    text = ' '.join([word for word in text.split() if word not in
stop_words])  # Remove stopwords
    return text

# Apply text preprocessing
df['friends_count'] = df['full_text'].apply(preprocess_text)

# Initialize the TF-IDF Vectorizer
vectorizer = TfidfVectorizer(max_features=5000)

# Fit and transform the processed text data
X = vectorizer.fit_transform(df['full_text']).toarray()

# Define the target variable
y = df['friends_count']  # Assume the target column is named
'sentiment'

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Initialize the RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the classifier
classifier.fit(X_train, y_train)

# Make predictions on the test data
y_pred = classifier.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

# Print classification report
print(classification_report(y_test, y_pred))
```

```
              date  favorite_count  followers_count  friends_count  \
0  7/18/2019 22:47               0               82            219
```

```
1  7/18/2019 22:43              0              102              129
2  7/18/2019 22:37              0              102              129
3  7/18/2019 22:35              0               13               16
4  7/18/2019 22:25              0              102              129

                                           full_text  retweet_count  
retweeted  \
0  @Mahi_2510 @swiggy_in The ultimate answer will...              0  
False
1                             @SwiggyCares I hope so.              0  
False
2  @SwiggyCares I think you have the order detail...              0  
False
3  @swiggy_in People are complaining here also, i...              0  
False
4  @SwiggyCares Do you even know the meaning of y...              0  
False

     screen_name tweet_id                     user_id
0     syamantak1        1                 6,19,59,419
1  Bharatbbhushn        2                14,32,84,383
2  Bharatbbhushn        3                14,32,84,383
3   taifkhalid01        4  8,30,34,00,00,00,00,00,000
4  Bharatbbhushn        5                14,32,84,383
date                 0
favorite_count       0
followers_count      0
friends_count        0
full_text            0
retweet_count        0
retweeted         2328
screen_name          0
tweet_id             0
user_id              0
dtype: int64

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

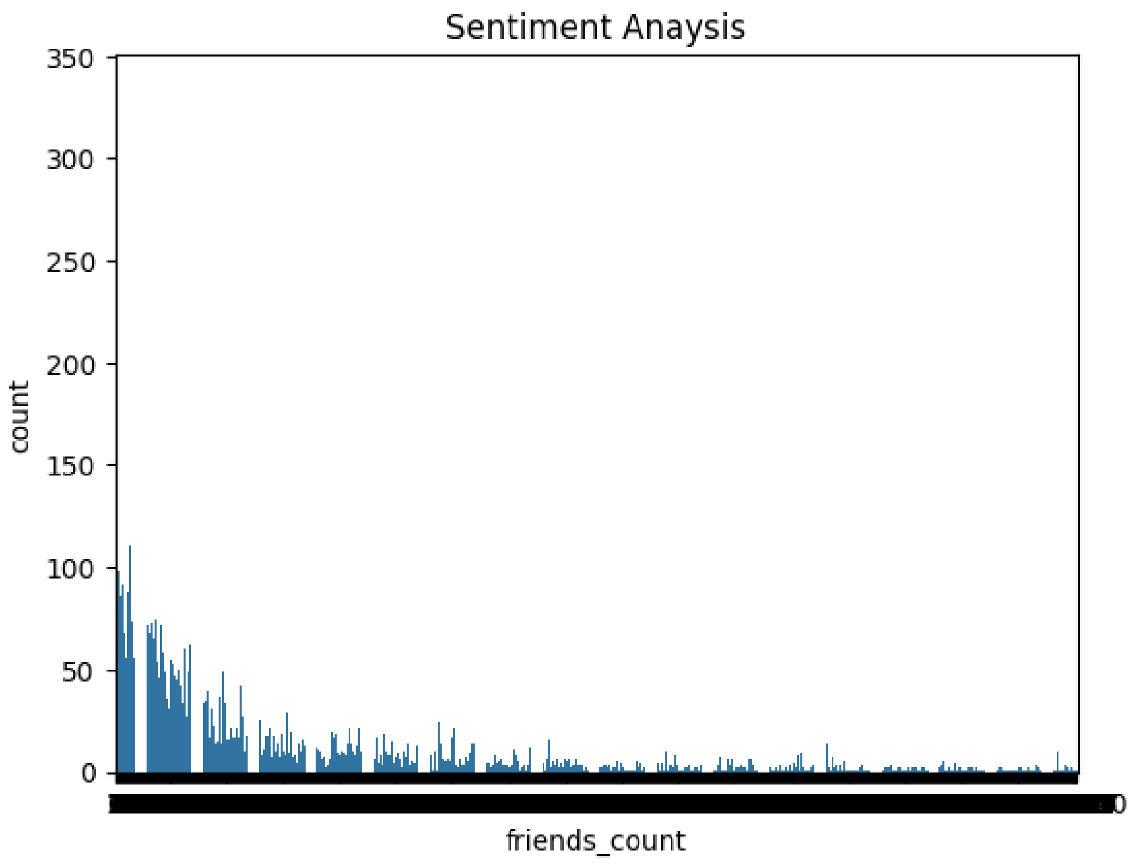
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset (make sure the path is correct)
df = pd.read_csv('/content/drive/MyDrive/swiggydataset.csv')

sns.histplot(df['followers_count'])
plt.title('followers_count')
plt.show()
```

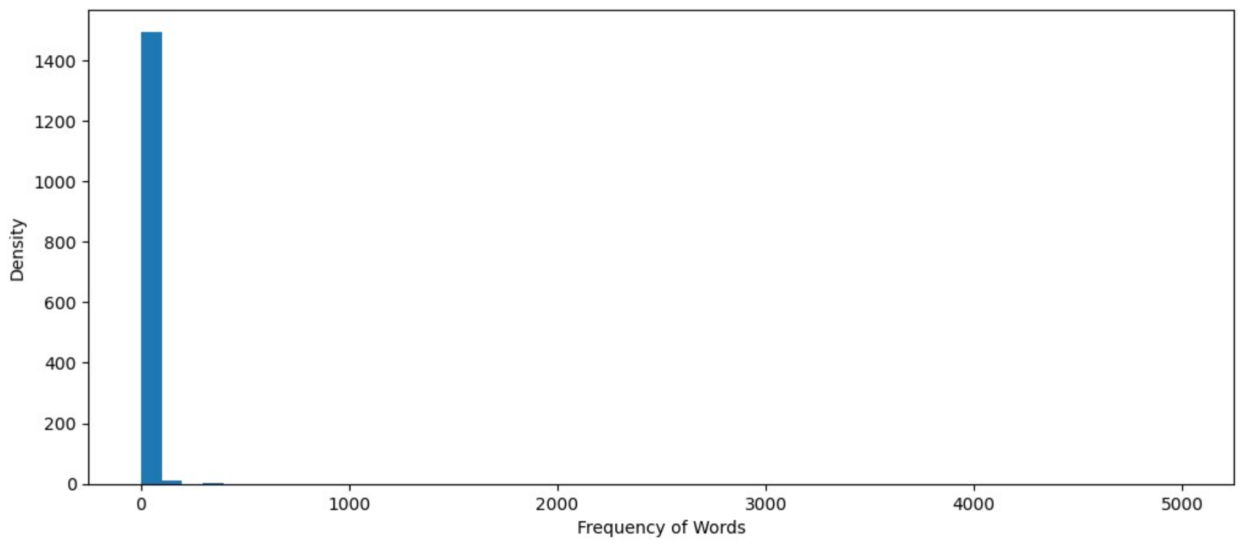
Product Price
Count
5000
4000
3000
2000
1000
0
0
1
2
3
4
5
6
7
1e6
followers_count

```
sns.countplot(data=df, x='friends_count')
plt.title('Sentiment Anaysis')
plt.show()
```



```
import matplotlib.pyplot as plt
# Assuming 'df' is your DataFrame and it has a column named
'friends_count'
features_counts_df = df['friends_count'].value_counts().reset_index()
features_counts_df.columns = ['word', 'counts']

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()
```


Density
Frequency of Words
0
200
400
600
800
1000
1200
1400
0
1000
2000
3000
4000
5000