```
import pandas as pd

df = pd.read_csv("play.csv")
```

```
df.head(2)
```

| | Reviews |
|---|---|
| 0 | One of the best thing about Netflix is, it con... |
| 1 | Good collection of movies across languages and... |

Next steps: View recommended plots

```
df.columns
```

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

```
print(df.head())
```

```
                                             Reviews
0  One of the best thing about Netflix is, it con...
1  Good collection of movies across languages and...
2  The new app layout is so poor, so tough to nav...
3  I am using netflix more than 4 years. Yesterda...
4  Any app have feature to replay the media from ...
```

```
print(df.tail())
```

```
                                             Reviews
95  There is 3 of us in the household and we all w...
96  I love using Netflix and use it all the time o...
97  So far good overall experience. Recent update ...
98  If I could give zero stars I would. My family ...
99  It's impossible to please everyone as you read...
```

```
from textblob import TextBlob
from collections import Counter
import re
```

```
def clean_text(text):
    if isinstance(text, str):
        # Remove newlines and extra whitespaces
        text = re.sub(r'\s+', ' ', text)
        return text
    else:
        return ''

# Function to perform sentiment analysis
def get_sentiment(text):
    analysis = TextBlob(text)
    # Return polarity as sentiment
    return analysis.sentiment.polarity
```

```
# Clean text
df['Cleaned_Review'] = df['Reviews'].apply(clean_text)

# Perform sentiment analysis
df['Sentiment'] = df['Cleaned_Review'].apply(get_sentiment)

# Identify negative reviews
negative_reviews_df = df[df['Sentiment'] < 0]

# Issues identified based on reviews (could be extended)
issues = {
    'Theatre': ['screen','movie','sound','theatre'],
    'Food Court': ['food court', 'food'],
    'Cleanliness': ['maintained', 'cleanliness','pathetic'],
}
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Provided reviews dataset
reviews = negative_reviews_df.apply(str).tolist()

# Combine all reviews into a single string
text = ' '.join(reviews)

# Generate word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
```

```
plt.show()
```


```
# Function to identify issues
def identify_issues(review, issues):
    identified_issues = []
    for issue, keywords in issues.items():
        for keyword in keywords:
            if keyword in review.lower():
                identified_issues.append(issue)
                break
    return identified_issues
```

```
# Apply issue identification
df['Identified_Issues'] = df['Cleaned_Review'].apply(lambda x: identify_issues(x, issues))

# Count frequency of each issue
issue_counter = Counter([issue for sublist in df['Identified_Issues'] for issue in sublist])

# Assign priority based on frequency
priority_list = {issue: index + 1 for index, (issue, _) in enumerate(issue_counter.most_common())}

print("Priority list of issues based on frequency:")
for issue, priority in priority_list.items():
    print(f"{issue}: Priority {priority}")
```

```
Priority list of issues based on frequency:
Theatre: Priority 1
Cleanliness: Priority 2
```

```
# Count number of negative reviews for each issue
print("\nNumber of negative reviews for each issue:")
for issue, count in issue_counter.items():
    print(f"{issue}: {count}")
```

```
Number of negative reviews for each issue:
Theatre: 41
Cleanliness: 1
```