In [1]:
```
import pandas as pd
```

In [3]:
```
df = pd.read_csv("Downloads/emotion_speech_dataset.csv")
```

In [5]:
```
print(df.head())
```

```
        filename                   transcription emotion
0  audio_001.wav            I am so happy today!   Happy
1  audio_002.wav            Why did you do that?   Angry
2  audio_003.wav              I feel really sad.     Sad
3  audio_004.wav      This is the best day ever!   Happy
4  audio_005.wav  I can't believe this happened!   Angry
```

In [7]:
```
print(df.isnull().sum())
```

```
filename         0
transcription    0
emotion          0
dtype: int64
```

In [9]:
```
print(df["emotion"].value_counts())
```

```
emotion
Happy    4
Angry    3
Sad      3
Name: count, dtype: int64
```

In [11]:
```
import re
```

In [13]:
```
def clean_text(text):
    text = text.lower()  # Convert to lowercase
    text = re.sub(r'[^a-zA-Z\s]', '', text)  # Remove punctuation
    return text
```

In [15]:
```
df["transcription"] = df["transcription"].apply(clean_text)
```

In [17]:
```
from sklearn.preprocessing import LabelEncoder
```

In [19]:
```
encoder = LabelEncoder()
df["emotion_encoded"] = encoder.fit_transform(df["emotion"])
```

In [21]:
```
print(dict(zip(encoder.classes_, encoder.transform(encoder.classes_))))
```

```
{'Angry': 0, 'Happy': 1, 'Sad': 2}
```

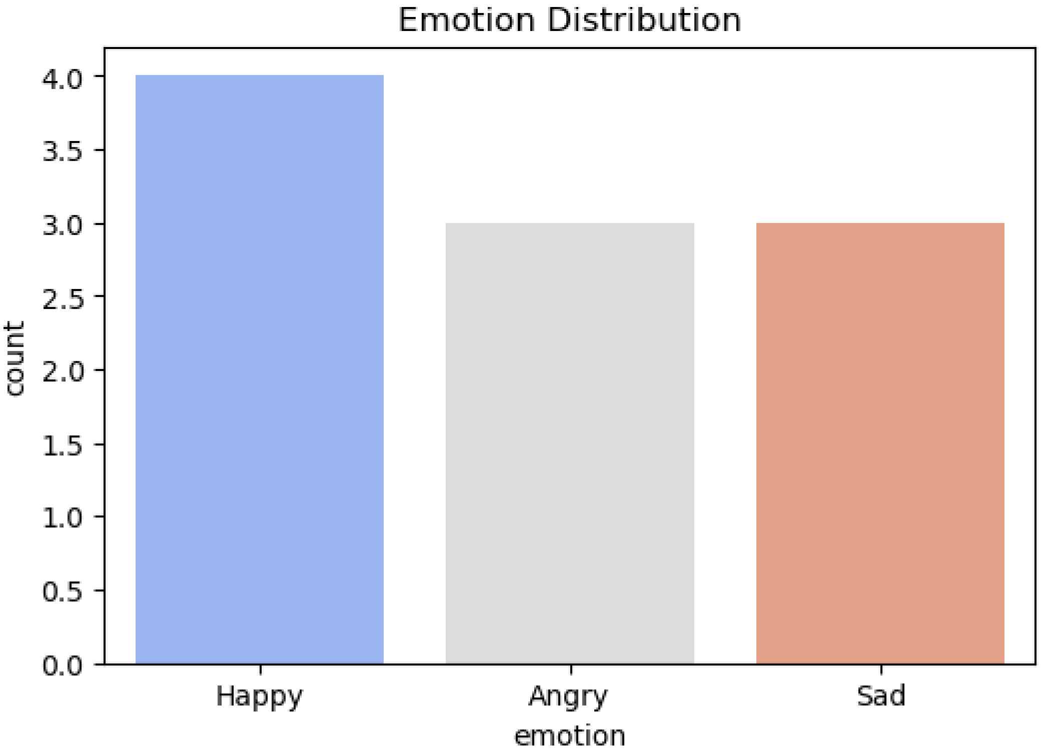
In [23]:
```
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure(figsize=(6,4))
sns.countplot(x=df["emotion"], palette="coolwarm")
plt.title("Emotion Distribution")
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11200\2093129265.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(x=df["emotion"], palette="coolwarm")
```



In [25]:
```
from collections import Counter
```

In [27]:
```
all_words = " ".join(df["transcription"]).split()
word_freq = Counter(all_words).most_common(10)
```

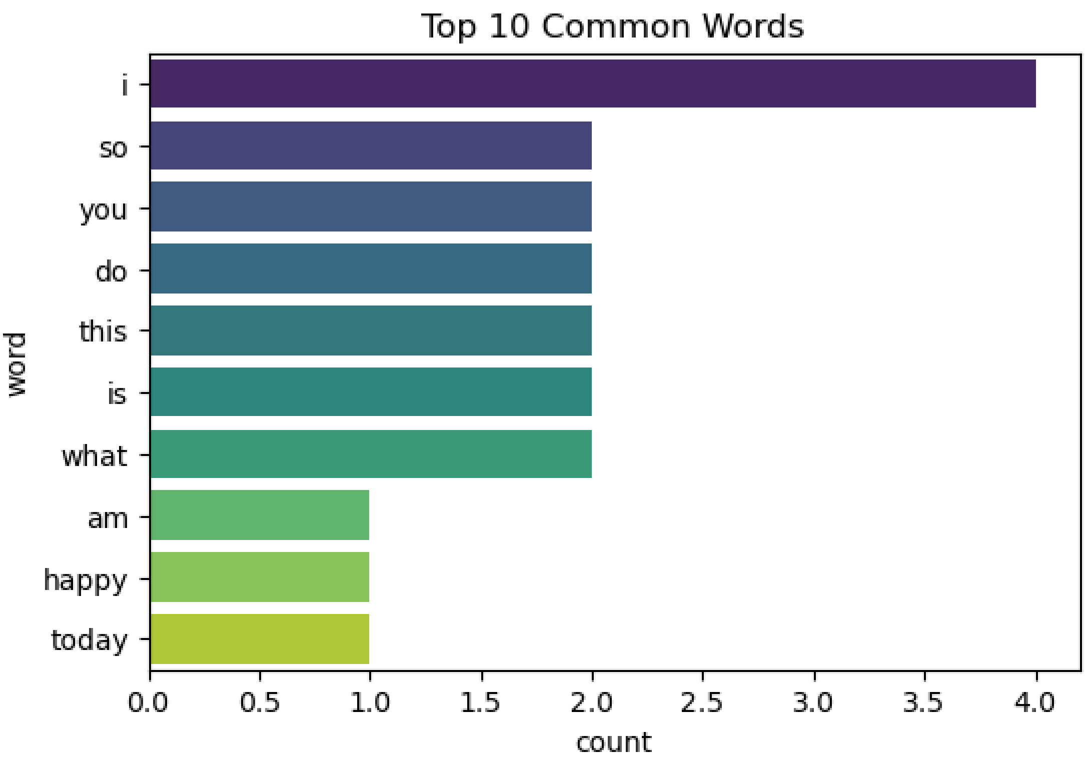
In [29]:
```
word_df = pd.DataFrame(word_freq, columns=["word", "count"])
```

In [31]:
```
plt.figure(figsize=(6,4))
sns.barplot(x="count", y="word", data=word_df, palette="viridis")
plt.title("Top 10 Common Words")
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_11200\3465060624.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x="count", y="word", data=word_df, palette="viridis")
```



In [35]:

```python
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=100)
text_features = vectorizer.fit_transform(df["transcription"]).toarray()
```

In [37]:

```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(text_features, df["emotion_e
```

In [39]:

```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [41]:

```python
model = LogisticRegression()
model.fit(X_train, y_train)
```

Out[41]:

```
▾ LogisticRegression
LogisticRegression()
```

In [43]:

```python
y_pred = model.predict(X_test)
```

In [45]:

```python
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.0
```

In [ ]: