Import Libraries

```
!pip install pandas numpy nltk scikit-learn matplotlib seaborn
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from panda
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.5.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from mat
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplot
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from ma
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from ma
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matp
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from mat
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-date
```

Load and Explore Dataset

```
from IPython.display import display
from ipywidgets import FileUpload
import pandas as pd
import io

# Create upload widget
upload = FileUpload(accept='.csv', multiple=False)
display(upload)

# After file is uploaded, run this to read it
def load_uploaded_file(upload_widget):
    if upload_widget.value:
        name = list(upload_widget.value.keys())[0]
        content = upload_widget.value[name]['content']
        df = pd.read_csv(io.BytesIO(content))
        print("File loaded successfully!")
        return df
    else:
        print("Please upload a file first.")

# Run this after uploading
# df = load_uploaded_file(upload)
```

Upload (1)

```
df = load_uploaded_file(upload)
```

```
File loaded successfully!
```

Text Cleaning & Preprocessing

```
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r"http\S+|www\S+|@\w+|#", "", text)  # Remove URLs and mentions
    text = re.sub(r"[^a-zA-Z\s]", "", text)           # Remove non-letter characters
    text = " ".join([word for word in text.split() if word not in stop_words])
    return text

# Apply cleaning function
df['clean_text'] = df['text'].apply(clean_text)

# Preview cleaned data
print(df[['text', 'clean_text']].head())
```

```
                                                text  \
0  I love this product! It works great and I'm ve...
1  This is the worst experience I've ever had. Co...
2   Not bad, but could be better. It's okay overall.
3  Absolutely fantastic! Highly recommend it to e...
4              Terrible service. Will not buy again.

                                          clean_text
0              love product works great im satisfied
1  worst experience ive ever completely disappointed
2                      bad could better okay overall
3     absolutely fantastic highly recommend everyone
4                                terrible service buy
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Train/Test Split

```python
from sklearn.model_selection import train_test_split

X = df['clean_text']
y = df['sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Text Vectorization (TF-IDF)

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=5000)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

Model Training (Logistic Regression)

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(X_train_vec, y_train)
```

```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

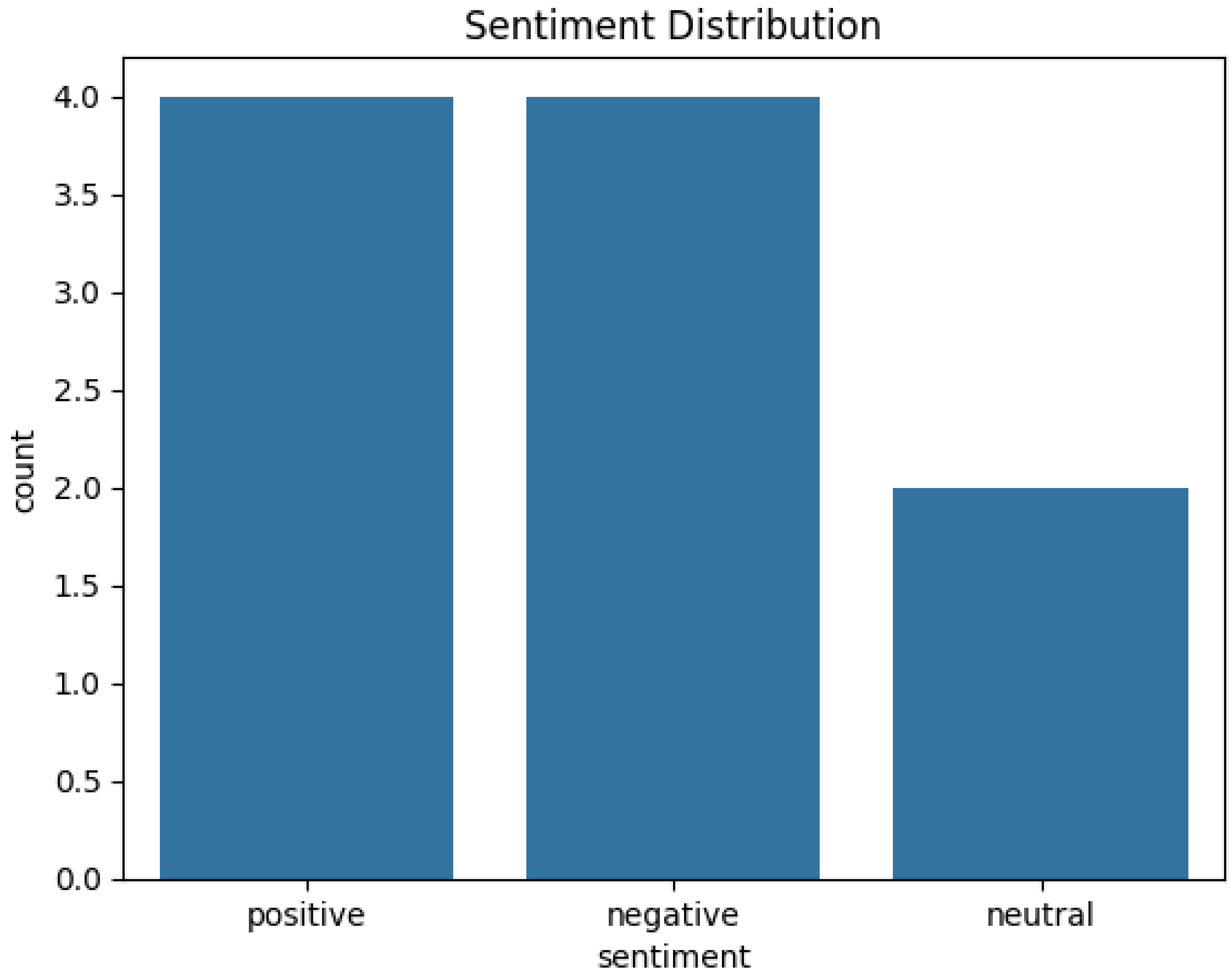
Model Evaluation

```
from sklearn.metrics import classification_report, accuracy_score

y_pred = model.predict(X_test_vec)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 1.0

Classification Report:
               precision    recall  f1-score   support

    negative       1.00      1.00      1.00         1
    positive       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2
```

Visualization (Optional but Useful)

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x='sentiment', data=df)
plt.title("Sentiment Distribution")
plt.show()
```



Double-click (or enter) to edit