Home | Untitled6 | Untitled4

localhost:8888/notebooks/Untitled6.ipynb

jupyter Untitled6 Last Checkpoint: 22 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

Open in... Python 3 (ipykernel)

[4]:

```
# Import necessary libraries
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
from textblob import TextBlob

# Download required nltk packages (only needed for first time)
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')

# Example text
text = "The product is not good"

# Step 1: Tokenization (splitting the text into words)
tokens = word_tokenize(text)
print("Tokens:", tokens)

# Step 2: Remove stopwords (filter out common, non-essential words)
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]
```

Home | Untitled6 | Untitled4

localhost:8888/notebooks/Untitled6.ipynb

jupyter Untitled6 Last Checkpoint: 22 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

Open in... Python 3 (ipykernel)

```
# Step 2: Remove stopwords (filter out common, non-essential words)
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words]
print("Filtered Tokens:", filtered_tokens)

# Step 3: Stemming (reduce words to their root form)
ps = PorterStemmer()
stemmed_tokens = [ps.stem(word) for word in filtered_tokens]
print("Stemmed Tokens:", stemmed_tokens)

# Step 4: Lemmatization (reduce words to their base form considering the context)
lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [lemmatizer.lemmatize(word) for word in filtered_tokens]
print("Lemmatized Tokens:", lemmatized_tokens)

# Step 5: Part-of-Speech (POS) Tagging (assign grammatical categories to each word)
pos_tags = nltk.pos_tag(filtered_tokens)
print("POS Tags:", pos_tags)

# Step 6: Sentiment Analysis using TextBlob and classify as positive, negative, or neutral
blob = TextBlob(text)
polarity = blob.sentiment.polarity
```

CAD/INR -0.21%

ENG IN

10:00 PM 9/12/2024

Home | Untitled6 | Untitled4

localhost:8888/notebooks/Untitled6.ipynb

jupyter Untitled6 Last Checkpoint: 22 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

Open in... Python 3 (ipykernel)

```
print("Lemmatized Tokens:", lemmatized_tokens)

# Step 5: Part-of-Speech (POS) Tagging (assign grammatical categories to each word)
pos_tags = nltk.pos_tag(filtered_tokens)
print("POS Tags:", pos_tags)

# Step 6: Sentiment Analysis using TextBlob and classify as positive, negative, or neutral
blob = TextBlob(text)
polarity = blob.sentiment.polarity

# Determine if the sentiment is positive, negative, or neutral
if polarity > 0:
    sentiment_label = "Positive"
elif polarity < 0:
    sentiment_label = "Negative"
else:
    sentiment_label = "Neutral"

print(f"Sentiment: {sentiment_label}")
```

```
Tokens: ['The', 'product', 'is', 'not', 'good']
Filtered Tokens: ['product', 'good']
Stemmed Tokens: ['product', 'good']
Lemmatized Tokens: ['product', 'good']
POS Tags: [('product', 'NN'), ('good', 'JJ')]
```

CAD/INR -0.21%

ENG IN 10:00 PM 9/12/2024