Stopword Removal from Text using NLTK how would

**ChatGPT said:**

Here's a complete and clean version of your project titled:

**📝 Stopword Removal from Text using NLTK**

This project demonstrates how to use Python's **NLTK** library along with **lambda functions** to remove stopwords from text. Stopwords are common words like *"is"*, *"the"*, *"in"*, etc., which carry less meaningful information and are often removed in Natural Language Processing (NLP) tasks.

**✅ Step 1: Install & Import Libraries**

python

CopyEdit

# Install NLTK if not already installed

# pip install nltk

import pandas as pd

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

# Download required NLTK resources

nltk.download('punkt')

nltk.download('stopwords')

**✅ Step 2: Create Sample Text Data**

python

CopyEdit

# Sample dataset

df = pd.DataFrame({

"raw\_text": [

"This is an example sentence.",

"We are learning lambda functions in Python!",

"Stopwords should be removed from this text."

]

})

**✅ Step 3: Define Stopwords and Cleaning Function**

python

CopyEdit

# Get English stopwords

stop\_words = set(stopwords.words('english'))

# Define function to remove stopwords from a sentence

def remove\_stopwords(text, stop\_words):

words = word\_tokenize(text) # Tokenize sentence

return ' '.join([word for word in words if word.lower() not in stop\_words])

**✅ Step 4: Apply Stopword Removal with Lambda Function**

python

CopyEdit

# Use lambda function to clean each row

if "raw\_text" in df.columns:

df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

**✅ Step 5: Output Result**

python

CopyEdit

print(df)

**Output:**

| **raw\_text** | **clean\_text** |
| --- | --- |
| This is an example sentence. | example sentence . |
| We are learning lambda functions in Python! | learning lambda functions Python ! |
| Stopwords should be removed from this text. | Stopwords removed text . |

**🧠 What You Should Explain to Your Professor**

I used a lambda function to apply a stopword removal function row-wise on a column of text in a pandas DataFrame. This is a common preprocessing step in NLP. The lambda function lambda x: remove\_stopwords(x, stop\_words) is used to pass each sentence (x) to the function that removes common English stopwords using the NLTK library.

Would you like me to export this into a **PDF**, **.ipynb notebook**, or something else for submission?

**You said:**

i did not get step 4

**ChatGPT said:**

No problem, Rubica! Let me explain **Step 4: Apply Stopword Removal with Lambda Function** in **very simple terms**, step by step:

**✅ What is Step 4 Doing?**

You have a **DataFrame** (df) with a column named "raw\_text" that contains sentences.

You want to **clean each sentence** by removing stopwords like "is", "the", "this", etc.

To do that, you use this line:

python

CopyEdit

df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

**🔍 Let's Break It Down:**

**1. df["raw\_text"]**

👉 This accesses the column that has your original sentences.

Example:

python

CopyEdit

df["raw\_text"]

# Output:

# 0 "This is an example sentence."

# 1 "We are learning lambda functions in Python!"

# 2 "Stopwords should be removed from this text."

**2. .apply(...)**

👉 This is a **pandas method** that lets you apply a function to **each value in a column**.

**3. lambda x: remove\_stopwords(x, stop\_words)**

👉 This is a **lambda function** (anonymous function).

It means:

* For every x in raw\_text, do this:
* Call remove\_stopwords(x, stop\_words) and return the result.

**✅ Think of it like:**

python

CopyEdit

for each sentence in df["raw\_text"]:

cleaned = remove\_stopwords(sentence, stop\_words)

save cleaned in df["clean\_text"]

**✨ Full Meaning of Step 4:**

We are **creating a new column** "clean\_text" by:

* Taking every sentence in "raw\_text"
* Passing it into the remove\_stopwords() function
* Using lambda to do this quickly and clearly
* Saving the result back into the new column

**🧠 Visual Example**

| **raw\_text** | **clean\_text** |
| --- | --- |
| This is an example sentence. | example sentence . |
| We are learning lambda functions. | learning lambda functions . |

Let me know if you'd like this written in Urdu or a flow diagram for better understanding!

**You said:**

did not get quiz me

**ChatGPT said:**

Perfect! Let’s turn this into a short quiz to help you **understand Step 4 (lambda with apply)** in a fun way. Try to answer these—I'll explain after each one.

**❓ Q1. What does .apply() do in pandas?**

A) It adds a new column  
B) It loops over each row or column and applies a function  
C) It removes rows  
D) It creates a plot

**❓ Q2. In this code, what is x?**

python

CopyEdit

df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

A) The column name  
B) A random number  
C) Each sentence (each row in the column)  
D) A function

**❓ Q3. What does this line of code do?**

python

CopyEdit

df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

A) It copies the text from raw\_text to clean\_text.  
B) It replaces stopwords in the raw\_text column with emojis.  
C) It removes stopwords from each sentence in raw\_text and stores the result in clean\_text.  
D) It deletes the column raw\_text.

**❓ Q4. Why do we use lambda here?**

A) To avoid writing a full def function  
B) To plot data  
C) To import data from Excel  
D) To sort the DataFrame

**❓ Q5. What is the role of remove\_stopwords(x, stop\_words)?**

A) It adds stopwords  
B) It prints the column  
C) It removes stopwords from x using the stop\_words list  
D) It changes the column name

Would you like to answer these first? Then I’ll give feedback 💡  
Or should I give you the correct answers directly?

**You said:**

b.c,c,a,c

**ChatGPT said:**

Great job giving it a try, Rubica! Let’s go over your answers and see how you did:

**❓ Q1. What does .apply() do in pandas?**

**Your answer:** B ✅  
**Correct!**  
✔️ .apply() is used to apply a function to each row or column in a DataFrame. You got it right!

**❓ Q2. In this code, what is x?**

**Your answer:** C ✅  
**Correct!**  
✔️ x represents **each sentence** (each value in the raw\_text column). Well done!

**❓ Q3. What does this line do?**

python

CopyEdit

df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

**Your answer:** C ✅  
**Correct!**  
✔️ This line removes stopwords from every sentence and saves the result in a new column called "clean\_text".

**❓ Q4. Why do we use lambda here?**

**Your answer:** A ✅  
**Correct!**  
✔️ lambda is a quick way to define a small function without using def. Perfect answer!

**❓ Q5. What is the role of remove\_stopwords(x, stop\_words)?**

**Your answer:** C ✅  
**Correct!**  
✔️ This function takes x (a sentence) and removes all stopwords from it using the NLTK stopword list.

**🟢 Score: 5/5 – Excellent! 🟢**

You clearly understand how **lambda functions with apply()** work for text processing! 🎉

Do you want a certificate-style note or PDF to show your professor that you completed this concept?

**You said:**

# === Clean article text if available === if "raw\_text" in df.columns: df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

**ChatGPT said:**

Yes, Rubica! That line of code is **100% correct** and professional. Here's a breakdown again with a polished explanation you can show your professor:

**✅ Explanation of Your Code**

python

CopyEdit

# === Clean article text if available ===

if "raw\_text" in df.columns:

df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

**🔍 What This Code Does:**

1. **Checks** if your DataFrame df has a column called "raw\_text".
2. **If it exists**, it:
   * Goes through **each row** (sentence) in the "raw\_text" column.
   * Applies a **lambda function** that:
     + Calls your remove\_stopwords() function.
     + Passes the sentence x and a set of stopwords (stop\_words) to that function.
   * Saves the cleaned text (with stopwords removed) in a **new column** called "clean\_text".

**🧠 What You Can Tell Your Professor:**

I used a lambda function inside the apply() method to clean a column of text data. The lambda applies a custom remove\_stopwords() function to each row in the "raw\_text" column. This is a typical use of lambda in data preprocessing workflows, especially in Natural Language Processing (NLP). It helped me understand how anonymous functions can be passed to pandas operations for clean, efficient code.

Would you like me to turn this explanation into a short paragraph or image-based proof (e.g. a screenshot style with annotations)?

**You said:**

break sdown the code

**ChatGPT said:**

Of course, Rubica! Let's break down this line of code **step by step**, in the simplest way possible:

**🧾 Code:**

python

CopyEdit

if "raw\_text" in df.columns:

df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

**🪄 Step-by-step Breakdown:**

**🧩 Part 1: if "raw\_text" in df.columns:**

* ✅ This checks **if the column** "raw\_text" exists in your DataFrame.
* 🔍 Why? To avoid an error if the column is missing.
* 📦 df.columns is a list of all column names.
* 👀 It's like saying:  
  *“If there's a column called raw\_text, then do the next step.”*

**🧩 Part 2: df["raw\_text"]**

* 📜 This accesses the "raw\_text" column of your DataFrame.
* 🧠 Each value here is a full sentence or paragraph (a string).
* Example:

python

CopyEdit

df["raw\_text"] = [

"This is a sample sentence.",

"We are learning Python.",

"Stopwords should be removed."

]

**🧩 Part 3: .apply(...)**

* 🧰 .apply() is used to apply a function to **each row** (or each cell) in a column.
* It helps you clean, modify, or analyze each value **without writing a loop**.
* Think of it like:  
  *"For every sentence in this column, do something to it."*

**🧩 Part 4: lambda x:**

* 🐑 lambda means: “I’m creating a mini function right here, without giving it a name.”
* x is a **placeholder** for each individual sentence in "raw\_text".

**🧩 Part 5: remove\_stopwords(x, stop\_words)**

* 🎯 This calls your **custom function** named remove\_stopwords.
* It passes:
  + x → the sentence from the column (e.g. "This is a test.")
  + stop\_words → a list or set of words to remove (like "is", "a", "the").
* The function returns a **cleaned sentence** with stopwords removed.

**🔁 All Together Now**

This line:

python

CopyEdit

df["clean\_text"] = df["raw\_text"].apply(lambda x: remove\_stopwords(x, stop\_words))

Can be imagined like this loop:

python

CopyEdit

new\_column = []

for sentence in df["raw\_text"]:

cleaned = remove\_stopwords(sentence, stop\_words)

new\_column.append(cleaned)

df["clean\_text"] = new\_column

**✅ Final Output**

You get a **new column** called "clean\_text" where all the stopwords have been removed!

Would you like a visual (diagram-style) breakdown or a short notebook version with output?

**You said:**

quize me

**ChatGPT said:**

Great, Rubica! Let's quiz you on the breakdown of the line of code we discussed. Answer each question and I’ll give you feedback after.

**❓ Q1. What does if "raw\_text" in df.columns: check for?**

A) If the DataFrame has any data  
B) If "raw\_text" is the name of a row  
C) If "raw\_text" is a column in the DataFrame  
D) If "raw\_text" is a function

**❓ Q2. What does .apply() do in this context?**

A) Deletes a column  
B) Applies a function to every element in the column  
C) Adds stopwords to each sentence  
D) Converts the column into a list

**❓ Q3. In lambda x: remove\_stopwords(x, stop\_words), what is x?**

A) A stopword  
B) A column name  
C) A single sentence from the raw\_text column  
D) A DataFrame

**❓ Q4. What is the purpose of remove\_stopwords(x, stop\_words)?**

A) To count how many words are in x  
B) To clean the DataFrame  
C) To remove stopwords from the sentence x  
D) To sort the stopwords

**❓ Q5. What happens after the .apply() runs?**

A) The original column is deleted  
B) A new column clean\_text is created with the processed sentences  
C) The column names change  
D) Nothing happens

Answer when you're ready! 😊

**You said:**

Q1: C Q2: B Q3: C Q4: C Q5: B

**ChatGPT said:**

Awesome, Rubica! You nailed it — all your answers are correct! 🎉

**Your answers:**

* Q1: C ✅
* Q2: B ✅
* Q3: C ✅
* Q4: C ✅
* Q5: B ✅

You clearly understand how this code works. If you want, I can help you create a short summary or proof to show your professor based on this quiz and explanation. Just let me know!