TEXT PREPROCESSING

Dataset:

This is an E-commerce Flipkart Dataset with exactly 20,000 samples. It has 15 columns with a lot of information. You can use is to predict which category it might fall under, considering "Description" for a product, analyse it.

Dataset Link:

https://www.kaggle.com/datasets/atharvjairath/flipkart-ecommerce-dataset/data

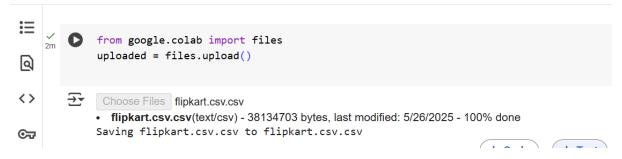
Text Preprocessing Algorithm

Input: Raw product description text

Output: Cleaned and processed textual data

- 1. Transform all text to lowercase.
- 2. Eliminate all punctuation marks from the text.
- 3. Remove any numeric characters or digits present.
- 4. Split the text into individual word tokens.
- 5. Filter out common stop words from the token list.
- 6. Perform stemming to reduce words to their base/root form.
- 7. Apply lemmatization to convert words to their valid dictionary base forms.
- 8. Finally, merge the processed tokens back into a single coherent string.

Uploading CSV File:



Load the Dataset

```
import pandas as pd
:= V
             df = pd.read_csv("flipkart.csv.csv")
             df = df[['description']].dropna()
Q
             df.head()
        →
<>
                                                    description
                    Key Features of Alisha Solid Women's Cycling S...
೦ಫ
              1 FabHomeDecor Fabric Double Sofa Bed (Finish Co...
Key Features of AW Bellies Sandals Wedges Heel...
                    Key Features of Alisha Solid Women's Cycling S...
                      Specifications of Sicons All Purpose Arnica Do...
```

Install and Download NLTK Resources:

```
\equiv
           import nltk
             nltk.download('punkt')
Q
             nltk.download('stopwords')
             nltk.download('wordnet')
<>
             nltk.download('punkt_tab')

→ [nltk_data] Downloading package punkt to /root/nltk_data...
⊙
             [nltk_data]
                         Unzipping tokenizers/punkt.zip.
             [nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
             [nltk\_data] \ \ Downloading \ package \ wordnet \ to \ /root/nltk\_data...
             [nltk_data] Downloading package punkt_tab to /root/nltk_data...
            [nltk_data] Unzipping tokenizers/punkt_tab.zip.
             True
```

Define the Preprocessing Function:

```
import string
            import re
            from nltk.tokenize import word_tokenize
Q
            from nltk.corpus import stopwords
            from nltk.stem import PorterStemmer, WordNetLemmatizer
<>
            # Initialize tools
            stop_words = set(stopwords.words('english'))
⊙
            stemmer = PorterStemmer()
            lemmatizer = WordNetLemmatizer()
def preprocess(text):
                # Step 1: Lowercase
                text = text.lower()
                # Step 2: Remove punctuation
                text = text.translate(str.maketrans('', '', string.punctuation))
                # Step 3: Remove numbers
                text = re.sub(r'\d+', '', text)
```

```
# Step 4 & 5: Tokenize and remove stopwords
tokens = word_tokenize(text)
tokens = [word for word in tokens if word not in stop_words]

# Step 6: Stemming
tokens = [stemmer.stem(word) for word in tokens]

# Step 7: Lemmatization
tokens = [lemmatizer.lemmatize(word) for word in tokens]

return ' '.join(tokens)
```

Apply Preprocessing:

