```
import pandas as pd
from sklearn.model_selection import train_test_split
import joblib

# Load the dataset
df =
pd.read_csv("/Users/marwahfaraj/Desktop/ms_degree_application_and_doc/final_projects/
C_new_topic_classification/data/bbc_news_text_complexity_summarization.csv")

# Split into train, validation, and test sets
train_val_df, test_df = train_test_split(df, test_size=0.2, random_state=42, stratify
train_df, val_df = train_test_split(train_val_df, test_size=0.25, random_state=42, st

# Save the test set separately (untouched)
test_df.to_csv("test_set.csv", index=False)
print("Test set saved successfully.")


import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA
from sklearn.svm import SVC
from nltk.corpus import stopwords

# Preprocessing function
def clean_text_column(df, column_name):
    stop_words = set(stopwords.words('english'))
    cleaned_texts = []

    for text in df[column_name]:
        if not isinstance(text, str):
            text = ""  # Handle non-string or NaN values
        text = text.replace('\n', '')  # Remove newlines
        words = re.findall(r'\b\w+\b', text.lower())  # Tokenize and lowercase
        filtered_words = [word for word in words if word not in stop_words]  # Remove
        cleaned_texts.append(' '.join(filtered_words))  # Join back into a string

    return cleaned_texts

# Clean train and validation data
train_df['cleaned_text'] = clean_text_column(train_df, 'text')
val_df['cleaned_text'] = clean_text_column(val_df, 'text')
```