**LABORATORY PROGRAM – 10**

**Write a simple streaming program in Spark to receive text data streams on a particular port, perform basic text cleaning (like white space removal, stop words removal, lemmatization, etc.), and print the cleaned text on the screen.**

# Install NLTK and download required data (run once)

!pip install nltk

import nltk

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')

from pyspark.sql import SparkSession

from pyspark.sql.functions import col, lower, regexp\_replace, split, explode, udf

from pyspark.sql.types import ArrayType, StringType

from pyspark.ml.feature import StopWordsRemover

from nltk.stem import WordNetLemmatizer

# Initialize SparkSession

spark = SparkSession.builder.appName("TextProcessing").getOrCreate()

# Define your input lines

lines = [

"Hello, I hate you.",

"I hate that I love you.",

"Don't want to, but I can't put",

"nobody else above you."

]

# Create DataFrame from lines

df = spark.createDataFrame(lines, "string").toDF("value")

# Step 1: Lowercase and remove punctuation

df\_clean = df.select(regexp\_replace(lower(col("value")), "[^a-zA-Z\\s]", "").alias("cleaned"))

# Step 2: Tokenize the cleaned text

df\_tokens = df\_clean.select(split(col("cleaned"), "\\s+").alias("tokens"))

# Step 3: Remove stop words

remover = StopWordsRemover(inputCol="tokens", outputCol="filtered")

df\_filtered = remover.transform(df\_tokens)

# Step 4: Lemmatization using NLTK WordNetLemmatizer with UDF

lemmatizer = WordNetLemmatizer()

def lemmatize\_words(words):

return [lemmatizer.lemmatize(word) for word in words]

lemmatize\_udf = udf(lemmatize\_words, ArrayType(StringType()))

df\_lemmatized = df\_filtered.withColumn("lemmatized", lemmatize\_udf(col("filtered")))

# Step 5: Explode the lemmatized words and show results

df\_lemmatized.select(explode(col("lemmatized")).alias("word")).show(truncate=False)

