LDS Methodology

1. **Collect Language Data**:

The data is collect from different GitHub repositories. The main repository used is that belonging to **Maskhane** ([*https://github.com/masakhane-io/masakhane-ner*](https://github.com/masakhane-io/masakhane-ner)).

The MasakhaNER 1.0: NER dataset is a collection of 10 African languages (Amharic, Hausa, Igbo, Kinyarwanda, Luganda, Luo, Nigerian-Pidgin, Swahili, Wolof and Yorùbá). This annotation for the dataset was performed by volunteers from the Masakhane community, leveraging the participatory research design that has been shown to be successful for building machine translation models. From the dataset, 3 languages have been selected: Hausa, Igbo, and Yorùbá. The dataset formats are txt.

1. **Preprocess the Data**:

To begin the prepossessing, the data was converted from separate txt files to a single CSV file made of two columns: Language and Text. This is the code followed by a step-by-step explanation:

### Code:

## 1. Imports

import pandas as pd  
import os  
import csv

## 2. Read Folders

# Input:  
input\_folder = '/Users/izzymohamed/Documents/WORK/AFRICA AGILE/Hackathon/Language Detection System/Data/Original/masakhane/'  
  
# Output:  
output\_folder = '/Users/izzymohamed/Documents/WORK/AFRICA AGILE/Hackathon/Language Detection System/Data/Final/masakhane/'  
output\_csv = 'all\_masakhane.csv'

## 3. Convert Txt to CSV

def txt\_to\_csv(input\_folder, output\_folder, output\_file):  
 """  
 Convert text files to a CSV file with two columns: language and text.  
  
 Parameters:  
 input\_folder (str): Path to the folder containing the input text files.  
 output\_folder (str): Path to the folder where the output CSV file will be saved.  
 output\_file (str): Name of the output CSV file.  
 """  
 output\_path = os.path.join(output\_folder, output\_file)  
  
 # Check if the output CSV file already exists  
 if os.path.isfile(output\_path):  
 print("Output CSV file already exists. Deleting and creating a new one.")  
 os.remove(output\_path)  
  
 with open(output\_path, 'w', newline='', encoding='utf-8') as csvfile:  
 writer = csv.writer(csvfile)  
 writer.writerow(['language', 'text'])  
  
 for filename in os.listdir(input\_folder):  
 if filename.endswith('.txt'):  
 language = filename.split('.')[0].lower() # Extract language from filename  
 filepath = os.path.join(input\_folder, filename)  
 with open(filepath, 'r', encoding='utf-8') as txtfile:  
 lines = txtfile.readlines()  
 for line in lines:  
 writer.writerow([language, line.strip()])

## 4. Save Result

txt\_to\_csv(input\_folder, output\_folder, output\_csv)

Output CSV file already exists. Deleting and creating a new one.

# Explanation:

### Let's break down the code step by step:

1. The function txt\_to\_csv takes three input parameters:
   1. input\_folder: A string representing the path to the folder containing the input text files.
   2. output\_folder: A string representing the path to the folder where the output CSV file will be saved.
   3. output\_file: A string representing the name of the output CSV file.
2. output\_path = os.path.join(output\_folder, output\_file): This line creates the full path for the output CSV file by combining the output\_folder and output\_file using the os.path.join function.
3. if os.path.isfile(output\_path):: This line checks if the output CSV file already exists by using the os.path.isfile function. If the file exists, the code proceeds to delete it to avoid overwriting.
4. with open(output\_path, 'w', newline='', encoding='utf-8') as csvfile:: This line opens the output CSV file in write mode ('w') and creates a file object named csvfile. The newline='' parameter ensures that new lines are not automatically added when writing to the file, and encoding='utf-8' specifies the character encoding for the file.
5. writer = csv.writer(csvfile): This line creates a CSV writer object using the csv.writer function. The writer object will be used to write rows to the CSV file.
6. writer.writerow(['language', 'text']): This line writes the header row to the CSV file with column names 'language' and 'text'.
7. The function then proceeds to loop through the files in the input\_folder using os.listdir(input\_folder).
8. if filename.endswith('.txt'):: This line checks if the current file in the loop ends with the extension '.txt', ensuring that only text files are processed.
9. language = filename.split('.')[0].lower(): This line extracts the language from the filename. It splits the filename using the period ('.') as a separator and takes the first part, which represents the language. The .lower() method is used to convert the language to lowercase for consistency.
10. filepath = os.path.join(input\_folder, filename): This line creates the full path to the current input text file.
11. with open(filepath, 'r', encoding='utf-8') as txtfile:: This line opens the input text file in read mode ('r') and creates a file object named txtfile.
12. lines = txtfile.readlines(): This line reads all lines from the input text file and stores them as a list of strings in the lines variable.
13. The code then enters a nested loop to iterate over each line in lines.
14. writer.writerow([language, line.strip()]): This line writes a new row to the CSV file for each line in the input text file. The row consists of the 'language' and the text content of the line after removing leading and trailing whitespaces using line.strip().

Clean and preprocess the text data. This may include removing punctuation, special characters, and converting all text to lowercase.

1. **Tokenization**:

Split the text into individual words or tokens. This step is essential for any NLP task.

1. **Feature Extraction**:

Convert the tokenized text into numerical features that machine learning models can process. Common techniques include TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings like Word2Vec, FastText, etc.

1. **Label Encoding**:

Assign a unique numerical label to each language in the dataset. For instance, if you have three languages (Yoruba, Hausa, and Igbo), you can encode them as 0, 1, and 2, respectively.