This an NLP assessment on Text Classification for a low resource language *(your understanding of the language in the dataset is not required hence anyone should be able to complete this assessment regardless of the languages your speak)* . You can only use **one** of the algorithms listed below.

```
              1. kNN (with a k value of 3)
              2. Naive Bayes
              3. Decision Trees
              4. Random Forrest
```

You'll be using a dataset from Hugging Face hence you'll need to install the datasets library. The dataset used is from a **Low-Resource Language** thus remember to take this into account.

- The code to load the data has already been provided thus you can just run the cell. Some random sampling has been done in advance to reduce plagiarism cases. (A copy of this code is shared below for reference)

```
import pandas as pd
import numpy as np
from datasets import load_dataset
import warnings
warnings.filterwarnings('ignore')
dataset = load_dataset("swahili_news")
train_texts = dataset["train"]["text"]
train_labels = dataset["train"]["label"]
test_texts = dataset["test"]["text"]
test_labels = dataset["test"]["label"]
df = pd.DataFrame()
df["text"]  = train_texts + test_texts
df["label"]  = train_labels + test_labels
df["text"].iloc[np.random.choice(df.index,  100, replace=False)]  = np.nan
df = df.sample(np.random.randint(24000,  24500))
df = df.reset_index(drop=True)

print("Your final dataset has have {} rows".format(df.shape[0]))
print("You have the following value counts for the label column:")
print(df["label"].value_counts())
df.head()
```

- You have liberty to tweak any other paramaeters for whatever algorithm your choose to achieve a better pefromance.

- You are limited to use either **Bag of Words** OR **Term Frequency - Inverse Document Frequency** for feature extraction.

- You can also use Scikit-learn's (Sklearn) for other tasks such as feature extraction.

- For evaluation, you can also use Scikit-learn's to generate a confusion matrix for your model. However, you are required to **develop 4 custom python functions** to generate the metrics listed below based on the results obtained from the confusion matrix.

- **DO NOT USE** *'from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score, classification_report'* **or similar varreints**
  - Accuracy (For your chossen algorithm)
  - Precision *(Least Peforming Class, as per your confussion matrix)*
  - Recall *(Least Peforming Class, as per your confussion matrix)*
  - F1 -Score *(Least Peforming Class, as per your confussion matrix)*

- You are free to access any shared files in class to help you complete this assessment. However, any slight sign of plagiarism will lead to an automatic fail (Zero Score) for this assessment.

- You are required to submit your work on or before **11:59 am Today (November 10, 2022)**

- Use the provided link to upload your final Jupyter / Google Colab Notebook file. After uploading, fill in answers to the questions listed on the same Google Form.
  - Google Forms Link: https://forms.gle/vtuVRNQDGajoxwPV8

You need to sign in to your Strathmore Gmail account to access the above link. If the above link fails, reach out to be given an alternative link.

All links will be inactive at exactly 11:59:59 am today November 10, 2022.
In case of anything, reach out directly via an email to: eolang@strathmore.edu . I've sent a copy of this file via email thus you can also respond/reply to that email for queries etc.