# Code Documentation

## Purpose

The purpose of this challenge develop a machine learning model to perform linear regression on a dataset containing information on deforestation, and use the trained model to make predictions on areas / counties in Kenya that are most vulnerable to deforestation. The predicted values are then saved to a CSV file and the accuracy of the predictions is calculated and printed. The hotspots predictions will help efficiently allocate resources to the most vulnerable areas to produce the most optimum results.

## Data Acquisition

For this project, I acquired the data on Kenya's counties from a variety of sources, including Kenya's meteorological department's website and other local websites. The meteorological department's website provided data on climate and weather patterns in the different counties, while other local websites provided information on population, area, and other demographic data. To ensure the accuracy and completeness of the data, I cross-checked information from multiple sources and verified it against other credible sources. The data was then cleaned and preprocessed to remove any inconsistencies or missing values before being used in the analysis. By utilizing multiple sources and ensuring data quality, the resulting analysis was able to provide a comprehensive picture of Kenya's counties and their characteristics.

## Data Cleaning

The data we collected had to be cleaned in order to be worked upon by my machine learning algorithm. An example of how we did this was by removing comas and percentage signs from the numerical values collected from the different websites. An example of the code used includes:

# Remove commas from population column and convert to float

df['Population Census 2009'] = df['Population Census 2009'].str.replace(',', '').astype(float)

df['Area (km2)'] = df['Area (km2)'].str.replace(',', '').astype(float)

df['Forest Cover Percentage'] = df['Forest Cover Percentage'].str.replace('%', '').astype(float)

df['Tree cover loss (2001-20) %'] = df['Tree cover loss (2001-20) %'].str.replace('%', '').astype(float)

The ode above removes all the comas from numerical values in the data and stores these numerical values as of type float.

## Libraries Used

* pandas: for loading and manipulating the dataset
* sklearn: for performing linear regression and splitting the dataset into training and testing sets

## Steps

1. Load the dataset from the CSV file 'DeforestationData.csv'
2. Remove non-numeric columns from the dataset, including the index and county name.
3. Remove commas from population and area columns, and convert them to float data type.
4. Remove the percentage sign from 'Forest Cover Percentage' and 'Tree cover loss (2001-20) %' columns, and convert them to float data type.
5. Separate the input features (X) and target variable (y).
6. Split the data into training and testing sets using a 80:20 ratio.
7. Train a linear regression model on the training data.
8. Use the trained model to make predictions on the test data.
9. Append the predicted values to the original dataframe.
10. Save the predicted values to a CSV file named 'PredictedDeforestationData.csv'.
11. Calculate the accuracy of the predictions and print it to the console.

## Inputs

The input to the code is the 'DeforestationData.csv' file, which contains information on deforestation, including population, area, forest cover percentage, and tree cover loss.

## Outputs

* The code produces the following outputs:
* A CSV file named 'PredictedDeforestationData.csv', which contains the predicted values for the 'Tree cover loss (2001-20) %' column.
* The accuracy of the predictions, printed to the console.

## Model Evaluation

Linear regression was the ideal model for analyzing the data collected above for several reasons. Firstly, the objective was to model the relationship between the target variable (tree cover loss) and a set of predictor variables (population, area, and forest cover percentage) that are known to have a linear relationship with the target variable. Therefore, linear regression, which assumes a linear relationship between the target and predictor variables, was the most appropriate method for this analysis.

Secondly, linear regression is a simple and transparent model that is easy to interpret and communicate to non-technical audiences. This makes it ideal for situations where the goal is to provide actionable insights to decision-makers.

Thirdly, linear regression is computationally efficient, making it suitable for large datasets with many variables. In contrast, more complex models such as neural networks or decision trees can be computationally intensive and may require large amounts of memory and processing power.

Finally, linear regression has a well-established statistical theory that allows for hypothesis testing and confidence interval estimation, which enables the analyst to make statistical inferences about the population based on the sample data.

In conclusion, linear regression was the most appropriate model for the data collected above, given its simplicity, interpretability, efficiency, and statistical properties. While other models such as decision trees, random forests, or neural networks may provide higher accuracy, they are more complex, computationally intensive, and less transparent, which may make them less suitable for certain applications.

## Feature Engineering

Feature engineering is the process of selecting and transforming raw data into features that are more informative and useful for machine learning models. It involves analyzing the data, identifying important features, and creating new features that capture important relationships between the input and target variables.

In the context of weather and deforestation data in Kenyan counties, feature engineering might involve identifying key weather variables such as temperature, precipitation, wind speed, and humidity, and creating new features that capture how these variables impact deforestation. For example, one could calculate the average temperature and precipitation for each county and create a new feature that captures the interaction between these variables and deforestation rates.

Here are some steps you could take to implement feature engineering for weather and deforestation data in Kenyan counties:

* Load the weather and deforestation data into Python using a library like pandas.
* Clean and preprocess the data, removing any missing or invalid values.
* Explore the data to identify important features and relationships between variables.
* Create new features that capture important relationships between the weather variables and deforestation rates. For example, you could create a new feature that captures the interaction between temperature and precipitation, or one that captures the seasonality of deforestation rates.
* Use the transformed data to train a machine learning model to predict deforestation rates in different counties.
* Evaluate the performance of the model using appropriate metrics such as mean squared error or R-squared.

Overall, the goal of feature engineering is to create a set of features that capture the most important relationships between the input and target variables, and that enable the machine learning model to make accurate predictions. By carefully selecting and transforming the input features, you can improve the performance of your machine learning model and make more accurate predictions about deforestation rates in Kenyan counties.

## Model Deployment

In the case of the data on Kenyan counties and weather patterns, the linear regression model can be deployed to provide insights and predictions that can inform policy decisions and guide resource allocation. For example, the model can be used to estimate the impact of population growth or deforestation on tree cover loss in different counties, or to identify the counties that are most vulnerable to climate change. The results of the analysis can be visualized using maps, charts, or other graphical representations, which can help decision-makers to understand the patterns and trends in the data and make informed decisions. The model can also be updated periodically with new data to ensure that the predictions remain accurate and relevant over time.

## Challenges

Data collection was really a tiresome process due to the fact that data wasn’t readily available from a single websites and we had to collect it from different sites distributed over the internet. At the same time, we had to develop our app prototype as soon as possible but In the end, we joined efforts and managed to submit the work at the expected time and at the same time, learn from each other.