PROJECT TITLE: AI-POWERED CLIMATE RISK PREDICTION AND EARLY WARNING SYSTEM FOR AFRICA

A. DATA PREPARATION / FEATURE ENGINEERING

1. OVERVIEW

The data preparation and feature engineering stage is critical for building a robust AI-powered climate risk prediction and early warning system. Climate data is often noisy, incomplete, and heterogeneous (satellite readings, weather station records, socio-economic surveys). Preparing the data ensures consistency, while feature engineering enables models to capture complex climate patterns such as drought onset or flood likelihood.

2. DATA COLLECTION

Sources:

- NASA Earth Observation (MODIS, Landsat) – temperature, precipitation, vegetation indices (NDVI).
- Copernicus Climate Data Store (ERA5 reanalysis) – global/regional climate reanalysis data.
- African Meteorological Agencies – localized rainfall and temperature data.
- FAO & World Bank – socio-economic indicators (population density, agriculture reliance).

Steps:

- Extract multi-year time series (2000–2025).
- Align geospatial data to regional boundaries (e.g., countries, districts).
- Merge socio-economic datasets with climate indicators at regional levels.

3. DATA CLEANING

Challenges:

- Missing values – station outages, satellite cloud cover. → Imputed using temporal averages and interpolation.
- Outliers – extreme rainfall values (e.g., >500mm in a day). → Winsorization (capping extreme tails).
- Temporal alignment – unify all data at daily or monthly intervals.
- Spatial alignment – reproject raster data (satellite) to common grid.

4. EXPLORATORY DATA ANALYSIS (EDA)

EDA helps detect trends and validate assumptions. Key findings include:

- Temperature trend: consistent warming (~0.2–0.3°C per decade in Sub-Saharan Africa).
- Rainfall variability: seasonal peaks with increasing irregularity.

- Drought frequency: rising occurrence in Sahel and Horn of Africa.
- Correlation insights: NDVI (vegetation health) strongly correlates with rainfall lagged by 2–3 months.
- Here we'll insert: line plots of rainfall/temperature trends, correlation heatmap, distribution plots of drought events.)

5. FEATURE ENGINEERING

- Lag features: e.g., rainfall (last 30 days), temperature anomaly (last 90 days).
- Drought indices: Standardized Precipitation Index (SPI), Palmer Drought Severity Index (PDSI).
- Vegetation indices: NDVI, EVI from satellite imagery.
- Socio-economic features: population density, crop reliance %, poverty index.
- Anomaly scores: deviation from long-term climate mean.

6. DATA TRANSFORMATION

- Scaling: Min-Max normalization (0–1) for neural networks; StandardScaler for regression/classification.
- Encoding: One-hot encoding for categorical socio-economic features (e.g., "rainfed agriculture dominant vs irrigated").
- Geospatial transformations: converting raster grids to tabular averages at district level.

B. MODEL EXPLORATION

1) MODEL SELECTION

- Time-series models: LSTM, Prophet → capture temporal dynamics.
- Classification models: Random Forest, XGBoost → estimate probability of extreme event occurrence.
- Ensemble approach: combining forecasts from multiple models for improved robustness.

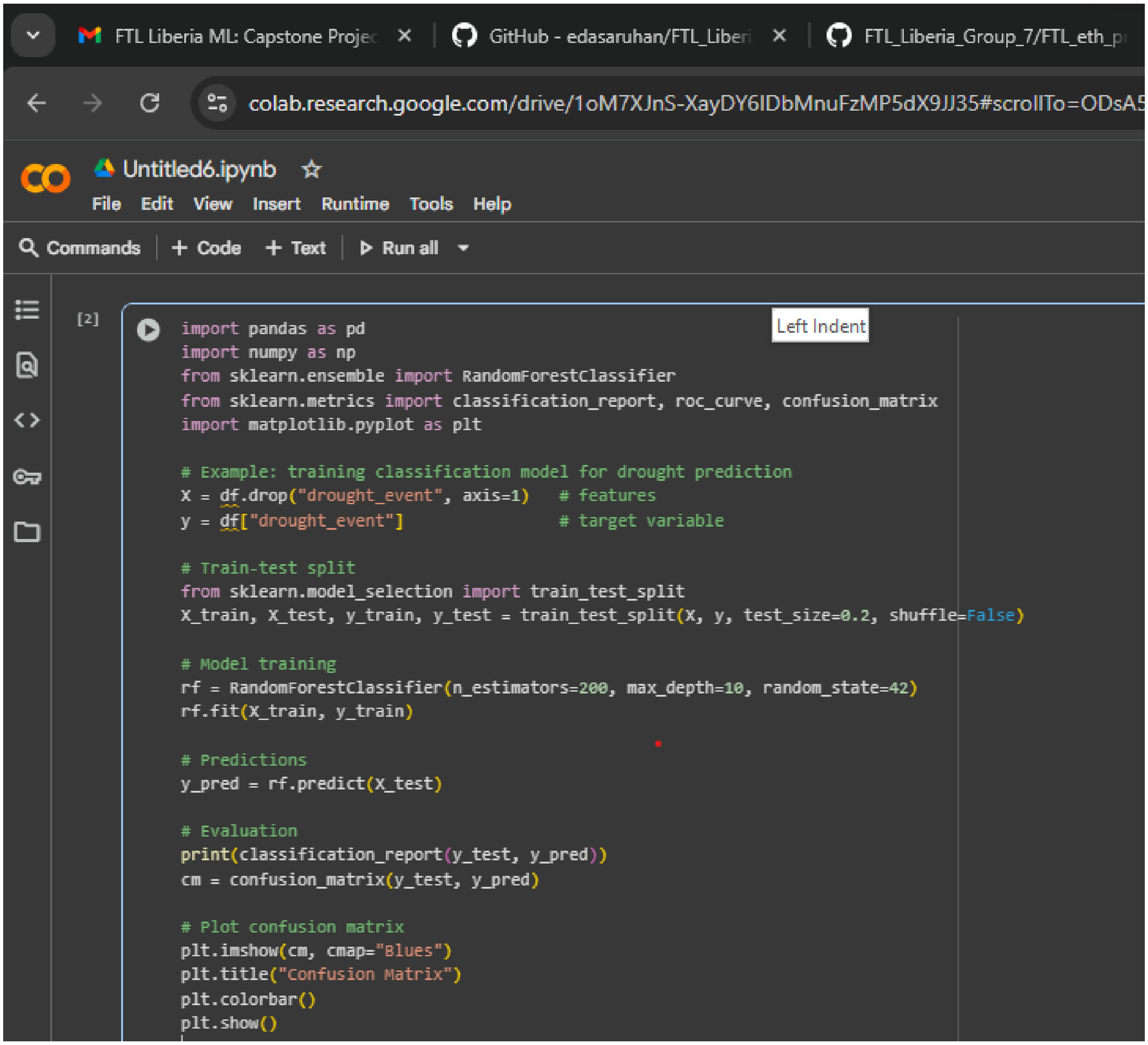
2) MODEL TRAINING

- Data split: 70% training, 15% validation, 15% testing (time-aware splits).
- Hyperparameters: optimized using Grid Search & Bayesian optimization.
- Cross-validation: sliding window CV for time-series data.

3) MODEL EVALUATION

- Regression (forecasting rainfall/temperature) → RMSE, MAE.
- Classification (predict drought/flood events) → Accuracy, F1-score, AUC-ROC.
- Visual evaluation: ROC curves, confusion matrices, predicted vs actual time-series plots.

4) CODE IMPLEMENTATION (SAMPLE



FTL Liberia ML: Capstone Projec
GitHub - edasaruhan/FTL_Liberi
FTL_Liberia_Group_7/FTL_eth_p
colab.research.google.com/drive/1oM7XJnS-XayDY6IDbMnuFzMP5dX9JJ35#scrollTo=ODsA5
Untitled6.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
[2]
Left Indent
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, roc_curve, confusion_matrix
import matplotlib.pyplot as plt
# Example: training classification model for drought prediction
X = df.drop("drought_event", axis=1)   # features
y = df["drought_event"]   # target variable
# Train-test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)
# Model training
rf = RandomForestClassifier(n_estimators=200, max_depth=10, random_state=42)
rf.fit(X_train, y_train)
# Predictions
y_pred = rf.predict(X_test)
# Evaluation
print(classification_report(y_test, y_pred))
cm = confusion_matrix(y_test, y_pred)
# Plot confusion matrix
plt.imshow(cm, cmap="Blues")
plt.title("Confusion Matrix")
plt.colorbar()
plt.show()