Data Premier League

Documentation

1. Tool & Framework Specification

Python Dependencies

All dependencies are listed in requirements.txt. Main packages include:

- **Core Data Handling:**
    - pandas – data loading, cleaning, merging
    - numpy – numerical computation
- **Visualization & Analysis:**
    - matplotlib / seaborn – plotting heatmaps, shock maps
    - networkx – trade network graphs
- **Machine Learning:**
    - scikit-learn – RandomForestRegressor, preprocessing, pipelines, evaluation metrics
- **Other Utilities:**
    - pycountry – ISO3 code mapping
    - warnings – suppressing warnings

2. Model Development & Training

2.1 Data Preparation & Splitting

- Input files are merged to build final_cleaned_dataset.csv.
- New engineered indicators are added to give final_with_indexes.csv.

- For forecasting,
    - Training set: all years before the last 2 available years.
    - Validation set: last 2 years of data (hold-out).
    - Split ensures temporal consistency and avoids leakage.

2.2 Preprocessing

- Cleaning: Missing numeric values → median imputation; categorical → mode imputation.
- Normalization:
    - GDP, trade, imports/exports scaled (millions USD).
    - Percent variables clipped between [0, 100].
- Feature Engineering
    - TDI – Trade Dependency Index.
    - ResilienceScore – composite of GDP per capita, TDI, poverty.
    - SpendingEfficiency – GDP growth relative to trade openness.
    - ShockImpact – disaster + unemployment + trade sensitivity.

2.3 Model Architecture

- RandomForestRegressor (sklearn) for forecasting socio-economic indicators to 2030.
- Targets:
    - GDP growth (annual %)
    - Poverty headcount ratio ($3/day PPP)
    - Resilience Score
- Features include TDI, ResilienceScore, ShockImpact, trade indicators, inflation, unemployment, population growth, etc.

2.4 Hyperparameter Tuning

- Selected empirically (fixed for reproducibility):
    - n_estimators=600
    - max_depth=None (grow fully, controlled by leaf size)
    - min_samples_leaf=2
    - random_state=42
- Scaling: StandardScaler inside a sklearn Pipeline.
- Validation:
    - R^2 and MAE reported for last-2-year holdout.