Stage 1 Roadmap (5 Phases, ≈ 350 words)

Phase 1 – Scoping & Data Sourcing

- **Key task:** Finalize metrics for the three pillars and bookmark trustworthy, API-friendly sources.
- **Tools:** Google Sheet or Notion for tracker, Python `requests`.
- **Datasets:**
  • *World Bank WDI* (population, life-expectancy, literacy, GDP, debt) World Bank Data
  • *IMF World Economic Outlook* (GDP per capita PPP, inflation, unemployment) IMF

Phase 2 – Extraction & Cleaning

- **Key task:** Pull CSV/JSON via each API, harmonize ISO-3166 country codes, and handle missing values.
- **Tools:** `pandas`, `pycountry`, `numpy`, `pandas-api-types`.
- **Datasets:**
  • *Transparency International CPI* (corruption scores) Transparency International
  • *SIPRI Military Expenditure* (absolute & % GDP defence spending) SIPRI

Phase 3 – Feature Engineering & EDA

- **Key task:** Scale numeric fields, derive per-capita/percentage ratios, and plot pairplots to spot outliers.
- **Tools:** `pandas`, `scikit-learn StandardScaler`, `matplotlib`.
- **Datasets:** Use the cleaned master table from Phase 2 (no new external sources).

Phase 4 – Pillar Models

- **Key task:** Train one model per pillar (e.g., PCA → KMeans for clustering or RandomForest for scoring) to produce a single numeric "pillar score" per country.

- **Tools:** `scikit-learn` (PCA, KMeans, RandomForest), `joblib` for persistence.

- **Datasets:** The pillar-specific feature subsets built in Phase 3.

Phase 5 – Ensemble & Dashboard

- **Key task:** Concatenate the three pillar-score vectors into a **stacked feature matrix**, fit a meta-model (e.g., `LinearRegression` or `GradientBoosting`) to compute the final ranking, and expose results in an interactive Plotly Dash app.

- **Tools:** `scikit-learn` `StackingRegressor`, `plotly`, `dash`, optional `leaflet.js` for maps.

- **Datasets:** Output from Phases 3-4; GeoJSON for country boundaries (free via Natural Earth).

Terminology Fixes

- **"Meta-matrix"** → **"stacked feature matrix"** (standard term in stacked/ensemble learning).

- **"Latent features"** usually refers to hidden representations learned by deep models; here you simply want the *pillar scores* produced by earlier models. Using "model outputs" or "intermediate features" will be clearer.

Next Step: Confirm this roadmap or ask to zoom into any single phase (e.g., Python code to query the World Bank API or a template Dash layout).

Work *horizontally*, not one pillar at a time—i.e., finish Phase 1 for **all three pillars**, then Phase 2 for all three, and so on.

Why a horizontal workflow is smoother

Aspect	Horizontal (all pillars per phase)	Vertical (one pillar through all phases)
Data consistency	You harmonize ISO-3166 codes + country names once, then reuse.	You may discover mismatched codes later and need to re-clean earlier pillars.
ETL scripts	One set of generic extraction/cleaning functions reused across pillars.	Duplicate or near-duplicate scripts for each pillar.
Feature reuse	Shared features (e.g., population) are available to every pillar early on.	You might engineer a useful ratio in pillar 1 and re-implement it later.
Team coordination	Clear checkpoints—everyone finishes Phase 1 before anyone moves on.	Team members may be in different phases, causing merge conflicts or knowledge gaps.
Compute efficiency	Merged master table built once, cached, and sliced per pillar.	Each pillar builds partial tables, then must be merged later for stacking.

Practical sequence

1. **Phase 1 – Sourcing**
   Create a "data catalog" sheet listing each metric, its API/URL, refresh rate, and file path.

2. **Phase 2 – Extraction/Cleaning**
   Write parameterized functions like `fetch_world_bank(indicator_id)` and `clean_cpi(df)`.
   Output a single `master_countries.parquet` file with all raw metrics.

3. **Phase 3 – Feature Engineering/EDA**
   From the master table, slice three NumPy/Pandas views:

   ○ `demo_df` (demographics)

   ○ `econ_df` (economics)

    - `infl_df` (influence)
      Scale, impute, visualize each; but common preprocessing steps stay centralized.

4. **Phase 4 – Pillar Models**
   Train three separate models, save their `pillar_score` columns back into the master table.

5. **Phase 5 – Ensemble & Dashboard**
   Stack the three score columns into a new matrix, fit the meta-model, feed results to Plotly Dash.

Here's how the snapshot files you just pulled line up with the three pillars we sketched earlier. I'm listing the **filename → clean column name** (from the `file_map` dict) and giving a one-line "why it belongs" note for context.

Pillar	Snapshot file (**`data/raw/worldbank/…`**)	Clean column	Why it fits
Demographics	`total_population.csv`	`pop_total`	Overall people power—base for per-capita ratios.
	`population_density.csv`	`pop_density`	Crowding / land-pressure signal.
	`urbanization_rate.csv`	`urban_pct`	Urban share affects infrastructure & labor markets.
	`population_growth_pct.csv`	`pop_growth_pct`	Demographic momentum indicator.
	`literacy_rate.csv`	`literacy_pct`	Human-capital baseline for workforce quality.
	`life_expectancy.csv`	`life_expect`	General health & development proxy.
Economic	`gdp_total_usd.csv`	`gdp_usd`	Absolute economic size.

	gdp_per_capita_ppp.csv	gdp_pc_ppp	Living-standards / productivity proxy.
	real_gdp_growth_pct.csv	gdp_growth_pct	Current economic dynamism.
	inflation_cpi_pct.csv	inflation_pct	Macro-stability gauge.
	unemployment_rate.csv	unemployment_pct	Labor-market slack indicator.
Influence / Power	gini_index.csv	gini_index	Governance & social-cohesion proxy (inequality).
	military_expenditure_pct_gdp.csv	mil_exp_pct_gdp	Hard-power spend relative to economy.

Gemini Prompt:
I am a data analytics student working on a final project. I have chosen the subject of taking a snapshot of the current geopolitical status of each country around the world, and I am using machine learning methods to compile a rankings or leaderboards if you will. Taking a deep look into categories ranging from population demographics, economics, and military and governmental influence using ML models will hopefully allow me to make the most accurate depiction of worldwide prosperity per country as possible.

So far I have compiled data and created a master CSV that I have scaled. I would upload it for you to analyze but it appears the upload does not recognize the file in my local drive. Regardless, I need you help determining the next best steps. I would like for you to outline different machine learning methods such as PCA, Hierarchical/Agglomerative/BIRCH clustering, factor analysis, or others you may think more useful. I have divided the data into 3 pillars (demographics, economics, influence) and was wondering what sorts of individual analysis we could do with those before combining all data into an ensemble approach. Maybe creating some sort of a composite score at the end. All of these are just suggestions and not meant to direct you towards what you deem the best route. Please outline different routes to take, their pros and cons, and please keep in somewhat simple and beginner-friendly.