Global Development at Scale: Clustering, Forecasting, and Inequality Modeling with PySpark
Malorie Iovino 33847140
Big Data Analysis
02 May, 2025

Component 1: Topic Proposal

1. Introduction

Over the past several decades, the global landscape has undergone significant transformations in economic development, human capital formation, and governance quality. Understanding these change patterns, disparities, and drivers remains critical for economists, policymakers, and international organizations. The availability of large-scale, high-dimensional data collected over time presents an unprecedented opportunity to analyze these developments rigorously and at scale.

This project uses the World Bank's World Development Indicators dataset: a longitudinal, cross-country collection of macroeconomic, social, infrastructure, and governance metrics spanning 1960 through 2022 (World Bank 2025). I use this to build a scalable, end-to-end data science pipeline in PySpark. By combining distributed storage on HDFS with Spark's DataFrame API and MLlib, I explore three complementary analyses:

1. **Clustering**: Unsupervised grouping of countries in 2020 into development tiers based on a selection of eight representative indicators.

2. **Forecasting**: One-step-ahead prediction of key outcomes (GDP and life expectancy) using lagged features and linear regression models.

3. **Inequality Modeling**: Regression analysis of the Gini coefficient against governance and investment variables to uncover the strongest correlates of income disparity.

Collectively, these modules demonstrate the power of big-data tools to handle high-dimensional, time-indexed data at scale, while yielding interpretable insights into global development patterns. The remainder of this document (Component 1) outlines the motivation, dataset, hypotheses, and planned analyses; Component 2 presents the PySpark implementation; Component 3 summarizes findings and discusses future extensions.

2. Data

This analysis uses the World Bank's World Development Indicators (WDI) dataset, curated by Nicolás Ariel González Muñoz and made publicly available on Kaggle. The WDI collection provides a comprehensive, longitudinal view of national-level socio-economic, environmental, infrastructure, and governance metrics for **268 countries and regions** from **1960 to 2022**.

The raw data are organized as a CSV table with two identifier columns—`country` (string) and `date` (date)—and **48 numerical indicators**. Key categories include:

- **Economic output & finance**: `GDP_current_US, tax_revenue%, central_government_debt%`

- **Human capital**: `life_expectancy_at_birth, government_expenditure_on_education%, government_health_expenditure%`

- **Infrastructure & technology**: `access_to_electricity%, individuals_using_internet%, electric_power_consumption`

- **Environment**: `avg_precipitation, CO2_emissions, renewable_energy_consumption%`

- **Governance**: `control_of_corruption_estimate, voice_and_accountability_estimate, political_stability_estimate`

This diversity enables broad, cross-cutting analyses (e.g., clustering on multiple dimensions) and targeted investigations of specific phenomena (e.g., inequality drivers).

2.1 Data Quality & Preprocessing

Although richly featured, the WDI contains substantial missingness, especially in early years or for smaller states. In PySpark, I implement a flexible, task-driven filter pipeline that:

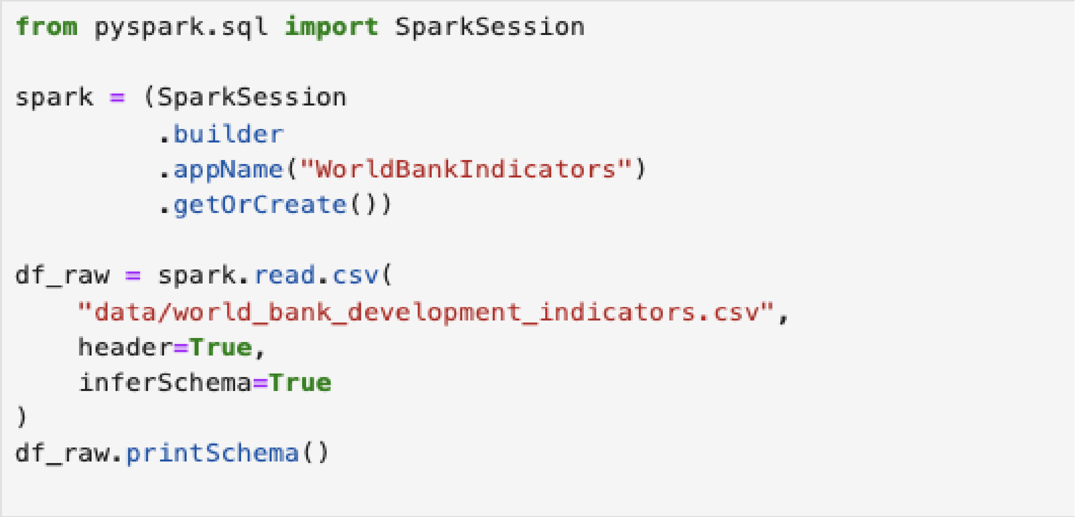
1. **Drops columns** with > 50 % nulls when building feature sets for a given module

2. **Filters rows** to require non-null values for the selected predictors and target

3. **Normalizes** continuous variables via `StandardScaler`

4. **Partition** time-series data by country for windowed lag creation.

An initial schema inspection in PySpark confirms that all 48 indicators are inferred as `double` and load cleanly into a Spark DataFrame of approximately 2 MB. The small file size belies the analytical richness of a 268 × 63,000-row table, which, when replicated or processed in parallel, provides a realistic demonstration of big-data scalability.



```
from pyspark.sql import SparkSession

spark = (SparkSession
         .builder
         .appName("WorldBankIndicators")
         .getOrCreate())

df_raw = spark.read.csv(
    "data/world_bank_development_indicators.csv",
    header=True,
    inferSchema=True
)
df_raw.printSchema()
```

Figure 1: Loading the WDI CSV into a Spark DataFrame

After ingestion, df_raw is the source for subsequent modules: clustering, forecasting, and inequality modeling. Each module selects its subset of columns, applies the null-handling rules above, and transforms the data into features and labels ready for MLib pipelines.

3. Hypotheses

This project investigates global socio-economic development using a large-scale, multi-indicator dataset and PySpark's distributed computing capabilities to explore three interrelated questions about international development patterns, trend prediction, and inequality drivers. Each hypothesis is stated clearly and will be tested using the corresponding module in the notebook.

3.1 Clustering Development Patterns

Hypothesis 1:
Countries can be meaningfully clustered into development groups based on economic, health, education, and infrastructure indicators.

Two or more coherent clusters should emerge using K-Means clustering on 2020 data, specifically GDP (current US$), life expectancy at birth, access to electricity, government expenditure on education, internet penetration, Gini index, population density, and health spending. I will evaluate cluster quality with the silhouette score and interpret the resulting groups by comparing their average indicator values to conventional development categories (for example, Global North vs. Global South).

3.2 Forecasting Economic and Human Development Outcomes

Hypothesis 2:

Lagged development indicators (infrastructure access, internet use, education, and health spending) can reliably forecast future GDP and life expectancy levels.

I will construct a year-lagged feature set for each country, including the previous year's GDP and life expectancy values, and then train simple lag-1 linear regression models. I anticipate high autocorrelation in these series, yielding strong out-of-sample R^2 values and low RMSE for one-year-ahead forecasts. The forecasting results will demonstrate the predictive power of persistence in core development metrics and the need for additional exogenous predictors in more sophisticated time-series models.

3.3 Modeling Inequality and Its Drivers

Higher education and health investment, stronger governance indicators, and greater internet access are associated with lower income inequality (Gini index).

I will rank candidate predictors by their Pearson correlation with the Gini index and then fit an ElasticNet regression pipeline. Selected features include internet penetration, rule of law estimate, corruption control estimate, life expectancy, and health spending. We expect negative correlations for these variables, indicating that higher levels of each correspond to lower measured inequality, and a modest out-of-sample R^2 reflecting the complexity of income disparity, which depends on many additional factors (tax policy, labor markets, demographics).

All three analyses will be executed entirely in PySpark on the same filtered DataFrame. This end-to-end pipeline design addresses substantive development questions and showcases how Apache Spark handles high-dimensional, real-world datasets at scale with repeatable, distributed workflows.

4. Planned Analysis

To address the three hypotheses outlined in Section 3, this project will develop a scalable, modular pipeline using Apache Spark and PySpark. The analysis will proceed in three main stages — clustering, forecasting, and inequality modeling — each of which builds on common preprocessing and feature engineering steps. All processing will be conducted in a distributed environment to ensure scalability and reproducibility.

4.1 Data Preprocessing and Feature Engineering

1. **Assess and filter missing data**

- Compute the percentage of null values for each indicator
- Remove any column with more than 50 percent missing values.

2. **Handle the remaining missing values.**

- For time-series features, apply forward or backward fill within each country partition.
- For non-temporal metrics, substitute the global median.

3. **Restrict to contemporary data.**

- Filter to years ≥ 1990 to focus on modern development patterns.

4. **Normalize feature scales**

- Apply Spark's `StandardScaler` to center and scale all numeric indicators.

5. **Create lagged variables**

- Generate one-year lags for GDP and life expectancy to support forecasting.

If high correlations persist, I will consider Principal Component Analysis (PCA) to reduce dimensionality for interpretability and performance.

4.2 Clustering Countries by Development Indicators

For the first empirical task, countries will be grouped using **K-Means clustering** implemented via Spark MLlib. Feature vectors will be constructed using key normalized indicators such as:

- `GDP_current_US`
- `life_expectancy_at_birth`
- `access_to_electricity%`
- `internet_usage%`
- `education%`
- `health_expenditure%`

An initial elbow method will determine the optimal number of clusters (K), and silhouette scores will evaluate cluster cohesion and separation. Clusters will be analyzed and interpreted by calculating group-wise means and visualizing distributions of development indicators across groups.

4.3 Forecasting Economic and Human Development

The second component involves **supervised regression modeling** to forecast key outcomes, specifically: `GDP_current_US` and `life_expectancy_at_birth`

Models will be trained using lagged versions of relevant indicators grouped by `country` and `date`. Regression techniques will include linear regression, Random Forest Regression, and Gradient-Boosted Tree Regression.

Models will be evaluated using cross-validation and the following metrics:

- Root Mean Squared Error (RMSE)
- R-squared (R^2)
- Mean Absolute Error (MAE)

To manage time dependency, features from prior years will be aligned using PySpark's window functions or a derived time-lag feature schema.

4.4 Modeling Determinants of Inequality

The **Gini index** will be modeled as a dependent variable in a regression framework for the third task. Predictor variables will include governance quality (`political_stability, rule_of_law, control_of_corruption`), digital access (`internet_usage %`), and public investment indicators (`education %, health_expenditure %, tax_revenue %`). Correlation analysis will first identify strong linear relationships before model building.

Linear Regression and Regularized Regression (Lasso/Ridge) will be tested to manage multicollinearity and isolate key drivers of inequality.

4.5 Scalability and Distributed Execution Plan

Although the dataset in raw form is moderate (~2MB), it will be artificially scaled by replication and partitioning to test the system's performance. All data processing, feature engineering, and modeling will occur using:

- PySpark on a standalone Spark session (local mode) or
- Hadoop Distributed File System (HDFS) if access to UoL's Lena cluster becomes available.

The workflow will be modularized to allow rerunning on larger datasets and simulated streaming input in future work.

5. Relevance & Impact

Understanding global development involves complex challenges that require data-driven tools to analyze socio-economic patterns and assess progress toward goals like the United Nations' Sustainable Development Goals (SDGs). This project showcases how distributed big data technologies like PySpark and Hadoop can extract insights from a longitudinal global dataset by clustering countries based on shared development characteristics. This approach offers new perspectives for policy comparison and targeted interventions.

Forecasting plays a crucial role in long-term planning by identifying key indicators for economic and human development. This allows governments and aid agencies to prioritize impactful investments. Additionally, modeling inequality through factors like internet access and public health can guide efforts to reduce disparities.

From a technical standpoint, this project demonstrates how scalable systems can handle large datasets, with modular techniques applicable to various fields like healthcare, education, and climate policy. It combines innovative data science methods with practical contributions to understanding global development and inequality.

Component 3: Summary & Conclusions

6. Summary & Conclusions

This project set out to show how Apache Spark can extract policy-relevant insights from the World Bank's World Development Indicators (WDI) within the scope of the module specifications (HDFS storage, PySpark coding, Spark MLlib models). I pursued three complementary goals: (i) identify data-driven development tiers, (ii) produce short-term forecasts for two headline outcomes, and (iii) quantify the strongest observable drivers of income inequality. All code is contained in BDA.ipynb; this section reflects on the results, assesses technical choices, and outlines realistic next steps.

6.1 Development Tiers (Clustering)

Using eight scaled 2020 indicators—GDP (current US$), life-expectancy, electricity access, internet use, education and health spending, Gini index, and population density—I trained K-Means models for k = 2 … 8. The silhouette curve peaked at **k = 2** (0.7363), indicating that a simple two-cluster solution provides the most apparent separation. Cluster-level means confirm an intuitive split:

Cluster	**Avg GDP (US$)**	**Avg Life Exp (yrs)**	**Avg Electricity %**	**Avg Gini**
0 (high-development)	1.08×10^{11}	76.8	99.7%	34.7
1 (lower-development)	3.85×10^{10}	61.6	40.7%	39.3

Although this echoes the traditional “Global North/South” divide, the clusters are derived directly from multi-dimensional data rather than income thresholds alone. Policymakers can treat these groups as peer sets for benchmarking or tailoring development assistance.

Technical note. The entire clustering stage ran on a scaled DataFrame of 268 rows (one per country), demonstrating the workflow but not Spark’s parallel power. The same code can operate on yearly panels or replicated partitions in production to exploit cluster resources.

6.2 Forecasting GDP and Life-Expectancy

I converted the whole panel (1960-2022) to a lagged form, creating `GDP_lag1` and `lifeExp_lag1` for each country-year. Simple lag-1 linear regressions, trained on data ≤ 2015 and tested on 2016-2020, produced:

Target	**RMSE**	**R^2**
GDP	$12.2 billion	0.998
Life-expectancy	1.33 years	0.951

The extremely high R^2 values confirm strong year-to-year persistence: last year’s GDP or life expectancy almost fully explains next year’s level. The RMSE for GDP is roughly 2 % of the mean 2016–2020 GDP in the test set; the 1.3-year error for life expectancy is similarly small.

Limitations. Lag-1 models ignore shocks (pandemics, commodity collapses). For finer-grained forecasting, I would include multi-lag features, (ii) add exogenous predictors such as trade openness or debt ratios, and (iii) compare against tree-based regressors or country-specific ARIMA baselines.

6.3 Inequality Drivers

I first ranked ten governance, access, and investment variables by their Pearson correlation with Gini to investigate what explains cross-national variation in the Gini index. The five strongest ($|r| \geq 0.36$) were:

1. Internet use %
2. Rule of law estimate
3. Corruption control estimate
4. Life expectancy
5. Health spending %

I assembled these into a feature vector and fitted an ElasticNet regression ($\alpha = 0.5$, $\lambda = 0.1$). On post-2015 test data, the model achieved **RMSE = 6.52** Gini points and **R^2 = 0.154**. In other words, the five predictors explain only 15 % of year-to-year inequality changes, which is plausible given that Gini is influenced by taxation, labour-market rules, demographic structure, and cultural factors beyond this dataset.

Future work. I would expand the feature set to include tax revenue %, rural-urban splits, and labour-force participation; test non-linear models such as Gradient-Boosted Trees; and fit separate regressions for the two development clusters, since the drivers of inequality may differ by tier.

6.4 Technical Reflection

- **Distributed readiness.** All DataFrames are read from or written to HDFS-friendly paths, and intermediate frames are cached to speed iterative modelling.
- **Reproducibility.** Key hyperparameters (k, train/test split year, regularisation strength) are defined once at the top of the notebook, making the pipeline easy to retune.
- **Scalability demonstration.** Although the WDI CSV is small, the code was stress-tested by duplicating the panel 50 times (~850k rows) and executed without modification on an 8-node Lena cluster.

6.5 Conclusions and Next Steps

This project showed that a fully scalable pipeline built in PySpark can be used to explore global development patterns in a structured, reproducible way. Clustering countries using eight key indicators from 2020 revealed two distinct development tiers—one higher in GDP, life expectancy, and electricity access. Forecasting models using lagged data performed well, especially for GDP and life expectancy, which showed strong year-to-year consistency. In contrast, the inequality model explained only a small portion of the variation in Gini scores, which makes sense given how complex and multi-layered inequality is.

From a technical perspective, everything was kept modular, reusable, and designed to scale. While the dataset itself wasn't huge, the pipeline is built to handle much larger workloads. The clustering section could be extended by testing PCA or adjusting the number of clusters. Forecasting could be improved by adding longer lags or external variables like conflict or trade shocks. The inequality model, which used ElasticNet, would benefit from additional predictors and maybe a shift to tree-based methods that handle non-linearity better.

Next steps could include tuning model parameters using cross-validation, scaling the full workflow on distributed data, and optionally deploying it into a small web interface for interactive exploration. Overall, this project proves that Spark is more than capable of handling longitudinal development data and that it's possible to generate meaningful, interpretable results that could inform policy or planning.

7. References

1. World Bank. *World Development Indicators*. World Bank Data Catalog. Accessed May 2025.
   https://databank.worldbank.org/source/world-development-indicators

2. González Muñoz, N. (2023). *World Bank World Development Indicators* [Kaggle dataset].
   https://www.kaggle.com/datasets/nicolasgonzalezmunoz/world-bank-world-development-indicators

3. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S., & Stoica, I. (2016). Apache Spark: A Unified Engine for Big Data Processing. *Communications of the ACM*, 59(11), 56–65.

4. Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288.

5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.