CxC 2025 SAP Challenge

By Shami-uz Zaman, Tony Ngo, Roah Cho, Angad Ahluwalia

February 24, 2025

Introduction

Introduction

Objective: Develop a Multidimensional Poverty Index (MPI) using socioeconomic and sustainability indicators.

Key Goals

1. **Build a Composite Index**
   - Aggregate poverty-related indicators into a single ranking system.
2. **Identify Key Influencers**
   - Determine which factors contribute the most to poverty.
3. **Recommend Policy Interventions**
   - Propose targeted strategies to reduce poverty and improve sustainability.

Why This Matters?

- Poverty is a **multidimensional issue** requiring **data-driven solutions**.
- Our approach combines **statistical analysis**, **machine learning**, and **clustering** to develop actionable insights.

Data Overview

Sources & Key Indicators

Dataset: Extracted from SAP's socioeconomic and sustainability indicators.

Key Indicators Used:

- **Education**
  - Literacy rate (ages 15-24), gender parity index.
- **Health**
  - Per capita health expenditure.
- **Living Standards**
  - Access to clean cooking fuels (% of population).

Why These Indicators?

- These three categories represent **core dimensions of poverty** as outlined by the United Nations and Sustainable Development Goals (SDGs).

Data Cleaning & Preparation

How We Processed the Data

Data Cleaning:

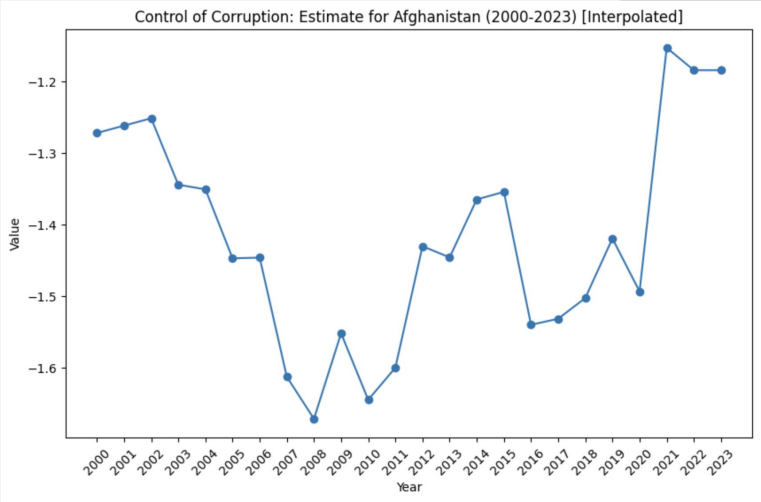
- Removed unnecessary columns (e.g. text descriptions).
- Converted year-based columns into numeric formats.
- Handled missing values using interpolation.

Feature Engineering:

- Normalized data using MinMax Scaling.
- Inverted values so that higher values indicate higher poverty levels.

Why?

- Ensures a consistent scale across indicators for meaningful analysis.



Interpolated time series of 'Control of Corruption: Estimate' for Afghanistan (2000-2023). Missing values were filled using linear interpolation to enable trend analysis.

Constructing the MPI

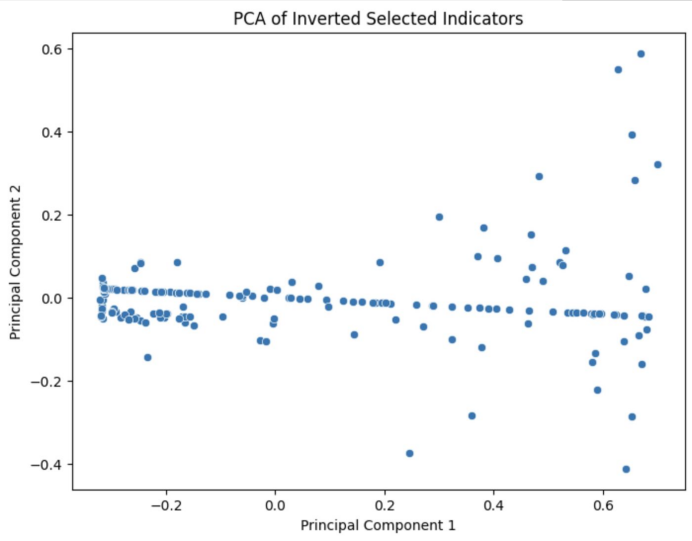
Multidimensional Poverty Index (MPI)

How We Built the Index:

1. Selected key indicators.
2. Normalized and inverted values.
3. Computed two MPI versions:
    - **Simple Average MPI** – **Equal weighting** of all indicators.
    - **PCA-Weighted MPI** – Uses Principal Component Analysis to assign **data-driven weights**.

Why PCA?

- Prevents single indicators from **dominating** the index.
- Assigns **higher weights** to the most statistically **influential** factors.



Scatter plot of the first two principal components of the inverted selected indicators. The horizontal spread indicates the variance captured by Principal Component 1, which was used to determine the weights for the Multidimensional Poverty Index (MPI).

Clustering Analysis for Targeted Interventions

Grouping Countries Based on MPI Scores

K-Means Clustering

- Used to **segment** countries into **three distinct groups** based on their **MPI** scores.

Why Clustering?

- Allows for **customized interventions** rather than one-size-fits-all solutions.

Top 10 most affected countries (highest MPI):

Indicator Name	MPI
Country Name	
Niger	0.855383
Chad	0.828730
Benin	0.788741
Guinea-Bissau	0.782751
Somalia	0.756234
Africa Western and Central	0.697301
Sierra Leone	0.682328
Congo, Dem. Rep.	0.680941
South Sudan	0.663937
Liberia	0.661270

Key Clusters

High-Poverty Cluster

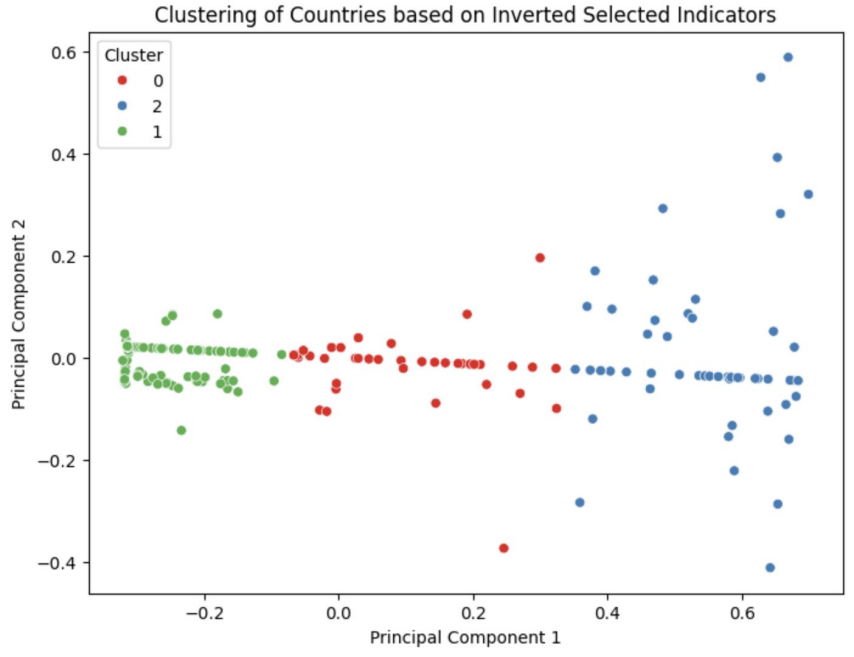
Countries with low access to clean fuels & high poverty index.

Moderate-Poverty Cluster

Countries with some access to health services but weak education systems.

Low-Poverty Cluster

Countries with strong health and education indicators but minor economic disparities.



Policy Recommendations

How Can We Reduce Multidimensional Poverty?

1. Prioritize Clean Cooking Solutions

- Expand access to clean energy sources (e.g., LPG, solar stoves).
- Provide government subsidies for clean fuel alternatives.

2. Integrate Education & Health Interventions

- Build integrated education & health programs for sustainable development.
- Increase healthcare investments in low-income regions.

3. Tailored Regional Policies

- Use clustering insights to design region-specific policies.
- Target infrastructure development in high-poverty areas.

Business & Sustainability Implications

Why This Matters for SAP's Sustainability Goals

How This Supports SAP's Sustainability Vision

- Aligns with SAP's commitment to **reducing global poverty** through **data-driven solutions**.
- Provides **actionable insights** for businesses, policymakers, and NGOs.

Future Applications

- **Real-time** monitoring dashboards for tracking MPI trends.
- Collaboration with **governments** & **organizations** for sustainable interventions.

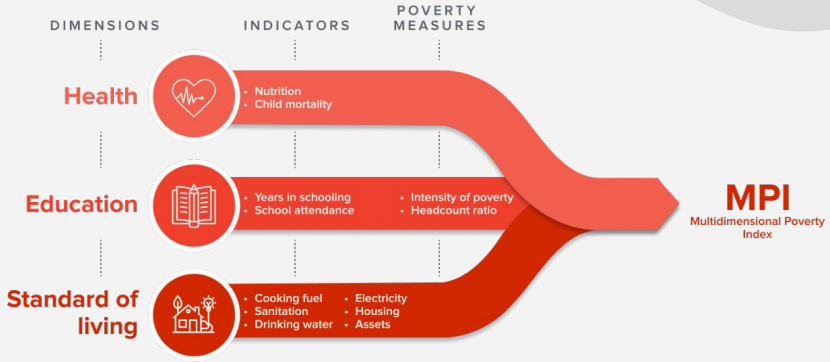
Conclusion & Future Work

Key Takeaways

- MPI provides a comprehensive **poverty ranking** based on **education**, **health**, and **living standards**.
- Access to **clean cooking fuels** is the **strongest predictor** of poverty.
- **Clustering analysis** enables **targeted interventions** for different poverty levels.

Next Steps

- Expand analysis to more **socioeconomic** indicators.
- Develop an **interactive dashboard** for policymakers.
- Integrate **real-time data** sources for monitoring **poverty trends**.



Thank you!

NaN values

- Removed columns with large amounts of missing values
- Took out short and long description column

```
# Drop columns not needed for our analysis
df = df.drop(columns=["short description", "long description"])
```

```
# Check for missing values
df.isnull().sum()
print("Missing values per column:")
print(df.isnull().sum())
```

```
Missing values per column:
Country Name             0
Country Code             0
Indicator Name           0
Topic                    0
short description    21547
long description       266
Indicator Code           0
Unit of measure          0
2000                 11765
2001                 13565
2002                 13132
2003                 13129
2004                 12912
2005                 12343
2006                 12479
2007                 12599
2008                 12500
2009                 12326
2010                 10886
2011                 11828
2012                 11686
2013                 12063
2014                 11786
2015                 11009
2016                 12103
2017                 12303
2018                 12252
2019                 11679
2020                 12707
2021                 13167
2022                 14686
2023                 21176
dtype: int64
```

Conversion

- Converted the year columns to numeric types
- Cleaned text fields
- Leads to consistency

```
# Define the list of year columns (2000 to 2023) and convert them to numeric
year_columns = [str(year) for year in range(2000, 2024)]
df[year_columns] = df[year_columns].apply(pd.to_numeric, errors='coerce')

# Remove extra whitespace from key text columns
df["Country Name"] = df["Country Name"].astype(str).str.strip()
df["Indicator Name"] = df["Indicator Name"].astype(str).str.strip()
```

```
country = "Afghanistan"
indicator = "Control of Corruption: Estimate"

# Filter data for Afghanistan and the specified indicator
afghanistan_data = df[(df["Country Name"] == country) & (df["Indicator Name"] == indicator)]
print("Filtered data for Afghanistan and indicator:")
print(afghanistan_data)

if not afghanistan_data.empty:
    # Use years 2000 to 2023
    values = afghanistan_data[year_columns].iloc[0]
    # Interpolate to fill missing values in the time series
    values_interpolated = values.interpolate(method='linear')
    plt.figure(figsize=(10, 6))
    plt.plot(year_columns, values_interpolated, marker='o')
    plt.title(f"{indicator} for {country} (2000-2023) [Interpolated]")
    plt.xlabel("Year")
    plt.ylabel("Value")
    plt.xticks(rotation=45)
    plt.show()
else:
    print(f"No data found for {country} with indicator '{indicator}'.")
```