# Video

Child labor is a global human rights violation that continues to steal the futures of millions of children. As the QuantaQuartet team, through this competition, we aim to draw attention to this ongoing issue within the constantly shifting global landscape by sharing insights from our data analyses.

Unfortunately—as we also mentioned earlier in our presentation—our dataset is limited to survey results and only includes data collected after 2010. We'd like to emphasize this once again. However, despite the scarcity of available data, our analyses have revealed several striking patterns and noteworthy findings that shed light on the broader realities of child labor around the world.

FIGURE-1

```
df_country_median = (
    df.groupby(["country_code","country_name"], as_index=False)
      .agg({
          'child_employment_rate': 'median',
          'log_child_employment_rate': 'median',
          'log_gdp_per_capita_ppp': 'median',
          'agriculture_value_added_pct': 'median',
          'primary_school_enrollment': 'median',
          'life_expectancy': 'median',
          'urban_population_pct': 'median',
          'rural_electricity_access': 'median',
          'child_population_pct': 'median',
          'youth_unemployment_rate': 'median',
          'log_private_consumption_per_capita': 'median',
          'governance_index': 'median'
      })
) #✅ Purpose: To reduce random fluctuations in the time series and represent the country's typical
  #level.


#  Interaction terms

# Does governance combined with GDP per capita impact child employment?
df_country_median['governance_x_gdp'] = (df_country_median['governance_index'] *
                                         df_country_median['log_gdp_per_capita_ppp'])
# Do urbanization and agriculture jointly influence child employment?
df_country_median['urban_x_agriculture'] = (df_country_median['urban_population_pct'] *
                                            df_country_median['agriculture_value_added_pct'])

# Does education combined with governance impact child employment?
df_country_median['education_x_governance'] = (df_country_median['primary_school_enrollment'] *
                                               df_country_median['governance_index'])

base_features = [
    'log_gdp_per_capita_ppp', 'agriculture_value_added_pct', 'primary_school_enrollment',
    'life_expectancy', 'urban_population_pct', 'rural_electricity_access', 'child_population_pct',
    'youth_unemployment_rate', 'log_private_consumption_per_capita', 'governance_index',
    'governance_x_gdp', 'urban_x_agriculture', 'education_x_governance'
]
```

FIGURE-1 Explanation

We began with several socioeconomic indicators related to education, economy, and infrastructure.

Because many countries had irregular or missing yearly data, we used the median value per country to capture each nation's typical situation and reduce outlier effects.

We also introduced three interaction terms to explore how governance, education, and economic structure might interact."

FIGURE-2

```
================================================================================
                                   DATA PREPARATION
================================================================================

Sample (n): 58
Number of features (p): 13
n/p ratio: 4.46 (ideal: >10)

✅ All features standardized (mean=0, std=1)

================================================================================
                          ⚠️ MULTICOLLINEARITY ANALYSIS (VIF)
================================================================================

VIF Interpretation: <5 (good), 5-10 (moderate), >10 (bad), >100 (catastrophic)

                           Feature         VIF
                  governance_index  209.185369
                 governance_x_gdp  138.017937
           education_x_governance  118.959618
             log_gdp_per_capita_ppp   17.058354
log_private_consumption_per_capita   12.556855
          agriculture_value_added_pct    8.955503
           rural_electricity_access    7.901274
               child_population_pct    6.952630
               urban_population_pct    6.823608
                    life_expectancy    6.053609
                 urban_x_agriculture    4.591081
             youth_unemployment_rate    1.584173
            primary_school_enrollment    1.361424
```

FIGURE-2 Explanation

It is observed that the number of observations is small. Since the data exhibits a multicollinearity problem, all variables have been standardized. To address this issue, Ridge, Lasso, ElasticNet, and Polynomial Ridge methods were used instead of classical regression (OLS).

FIGURE-3

```
lasso.fit(X_scaled, y)
selected_features = X.columns[lasso.coef_ != 0].tolist()
print(f"Lasso selection ({len(selected_features)} features)::")
print(selected_features)


X_reduced = X_scaled[selected_features]
ridge_reduced = RidgeCV(alphas=alphas, cv=cv)
ridge_reduced.fit(X_reduced, y)
cv_r2_reduced = cross_val_score(ridge_reduced, X_reduced, y, cv=cv, scoring='r2').mean()
print(f"Reduced model CV R²: {cv_r2_reduced:.4f}")
```

✓ 16.6s

```
Lasso selection (4 features)::
['agriculture_value_added_pct', 'life_expectancy', 'urban_population_pct', 'rural_electricity_access']
Reduced model CV R²: 0.2886
```

FIGURE-3 Explanation

Our dataset had many correlated indicators (VIF > 100) and only around 50–70 observations, the classical OLS model was unstable.

This multicollinearity inflates variances of the coefficients, making them unreliable."

we used Lasso for variable selection.

Lasso applies an L1 penalty, shrinking some coefficients to zero — effectively removing uninformative variables.

This allowed us to identify the four most relevant indicators — agriculture, life expectancy, urban population, and rural electricity access — while keeping the model interpretable."

FIGURE-4

```python
df = pd.read_csv('analyze data.csv', index_col=0)

df_country_median = (
    df.groupby(["country_code","country_name"], as_index=False)
      .agg({
          'log_child_employment_rate': 'median',
          'agriculture_value_added_pct': 'median',
          'life_expectancy': 'median',
          'urban_population_pct': 'median',
          'rural_electricity_access': 'median',
      })
)
base_features = ['agriculture_value_added_pct',
          'life_expectancy',
          'urban_population_pct',
          'rural_electricity_access',]
target = "log_child_employment_rate"
df_clean = df_country_median.dropna(subset=[target]+ base_features)
```

FIGURE-4 Explanation

We re-filtered the data based on the indicators selected by the Lasso model.

FIGURE-5

```
==========================================================================================
                              DATA PREPARATION
==========================================================================================

Sample (n): 78
Number of features (p): 4
n/p ratio: 19.50 (ideal: >10)

✅ All features standardized (mean=0, std=1)

==========================================================================================
                         ⚠️ MULTICOLLINEARITY ANALYSIS (VIF)
==========================================================================================

VIF Interpretation: <5 (good), 5-10 (moderate), >10 (bad), >100 (catastrophic)

                    Feature      VIF
            life_expectancy 3.639950
    rural_electricity_access 2.986257
agriculture_value_added_pct 2.514534
       urban_population_pct 1.800986
```

FIGURE-5 Explanation

The number of observations increased, the n/p ratio improved, and multicollinearity was eliminated.

FIGURE-6

```
==========================================================================================
                              🤖 MODEL TRAINING
==========================================================================================

Running 5-fold CV...

✓ OLS completed
✓ Ridge completed
✓ Lasso completed
✓ ElasticNet completed
✓ Polynomial Ridge completed

==========================================================================================
                              📊 MODEL COMPARISON
==========================================================================================
           Model  Train_R²    CV_R²  CV_RMSE  Bias_Var_Gap Overfitting
           Ridge  0.393773 0.152979 0.740115      0.240794        High
      ElasticNet  0.399229 0.146451 0.741209      0.252778        High
           Lasso  0.404536 0.119018 0.747182      0.285517        High
             OLS  0.404653 0.116223 0.745828      0.288430        High
Polynomial Ridge  0.417248 0.053076 0.773474      0.364172        High

==========================================================================================
🏆 BEST MODEL: Ridge (CV R² = 0.1530)
==========================================================================================
```

FIGURE-6 Explanation

We compared five different models and found that Ridge performed the best. Since our dataset has a small number of observations and a heterogeneous distribution, the risk of overfitting is high.

FIGURE 7

```
==========================================================================================
                         🔍 DETAILED ANALYSIS: Ridge
==========================================================================================

📌 Ridge Regularization:
   • Optimal α: 40.3702
   • All 4 features used (coefficients shrunk)

📊 Performance:
   • Training R²: 0.3938
   • CV R²: 0.1530
   • RMSE: 0.6947
```

FIGURE-7 Explanation

The Ridge model retained all variables while shrinking their coefficients and demonstratedn the most balanced performance. While the training $R^2$ indicates a reasonable fit, the cross-validation results suggest limited generalization ability. This implies that despite the small and heterogeneous sample — which increases the risk of overfitting — Ridge provided the most stable solution.

FIGURE 8

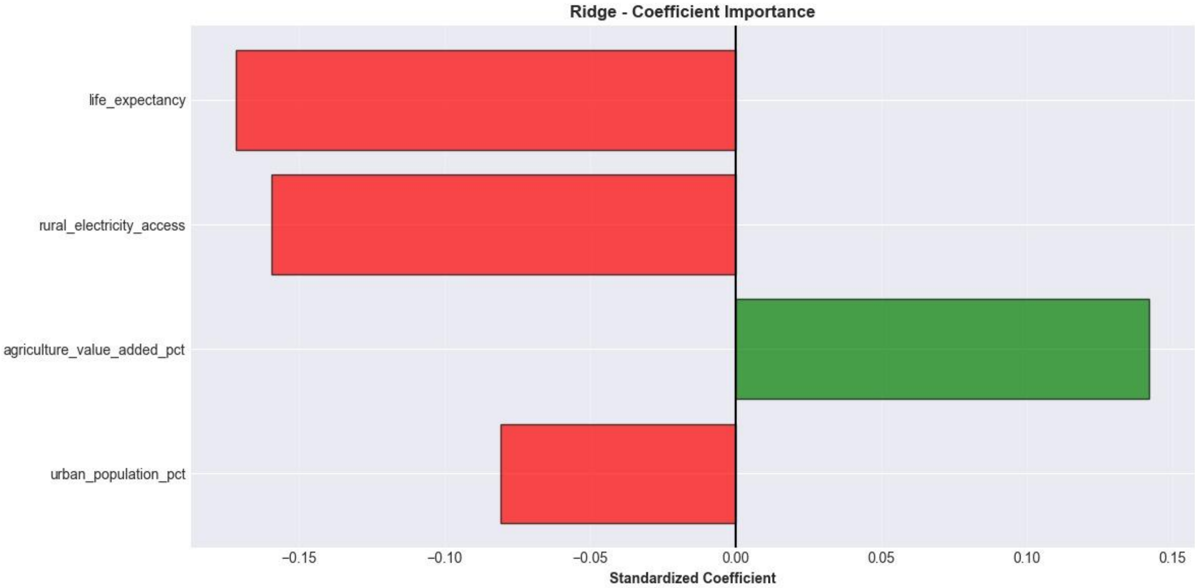



FIGURE -8 Explanation

The Ridge coefficient plot shows that **agriculture_value_added_pct** has the strongest **positive** effect on the target variable, meaning higher agricultural contribution is associated with higher child employment rates.

Conversely, **life_expectancy** and **rural_electricity_access** have the strongest **negative** coefficients, suggesting that better living conditions and infrastructure reduce child labor. **Urban_population_pct** also has a mild negative impact, indicating that more urbanized areas tend to have lower child employment levels

-SUGGESTIONS FOR SUSTAINABLE DEVOLOPMENT-

"Our regression shows that child labour is mainly driven by low health, weak infrastructure, and agricultural dependence —

and all three are rooted in governance and development inequality.

So, sustainable development must start from the ground: invest in health, power, and diversification."

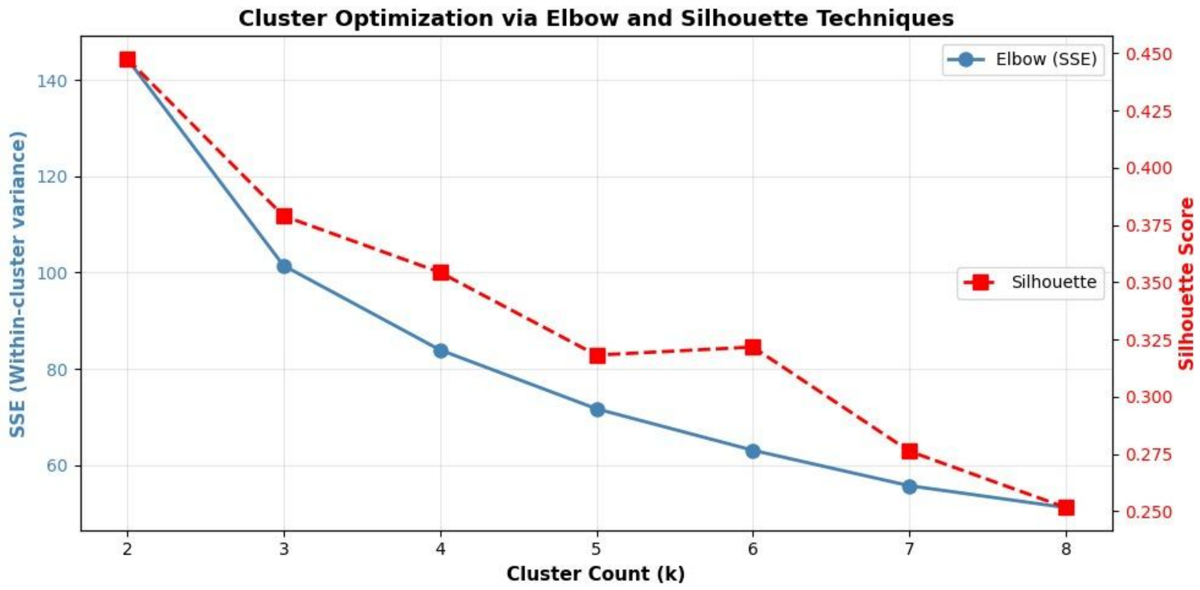
-CLUSTERING-

After identifying which factors statistically affect child employment,

we used clustering to see how countries group together based on these development indicators.

The goal was to reveal global development patterns — showing that child labour is not an isolated issue,

but part of broader structural inequalities between developing and developed regions




to decide how many clusters best represent our data,

we applied the Elbow and Silhouette methods.

Both curves show that performance stabilizes around k = 2,

meaning the countries naturally separate into two clear groups —

developed and developing — based on development indicators.

FIGURE-9

```python
# 🔄 Final K-Means model
kmeans = KMeans(n_clusters=best_k, random_state=42, n_init=10)
clusters = kmeans.fit_predict(X_scaled)

# Cluster bilgilerini dataframe'e ekle
df_clusters = X.copy()
df_clusters['cluster'] = clusters

print(f"\n✅ The K-Means model was fitted")
print(f"   • Cluster Count: {best_k}")
print(f"   • Total Country: {len(df_clusters)}")
print(f"\n📊 Cluster distribution:")
print(df_clusters['cluster'].value_counts().sort_index())
```

✓ 0.0s

```
✅ The K-Means model was fitted
   • Cluster Count: 2
   • Total Country: 78

📊 Cluster distribution:
cluster
0    35
1    43
Name: count, dtype: int64
```

We used K means model for clustering

FIGURE-10

```python
# 6️⃣ One-way ANOVA

print("\n" + "="*90)
print(" "*30 + "📊 ANOVA TEST")
print("="*90)
print("\n Feature-wise differences across clusters\n")

for col in features:
    groups = [df_clusters[df_clusters['cluster']==i][col] for i in range(best_k)]
    f_stat, p_val = f_oneway(*groups)
    significance = "✅ Statistically significant" if p_val < 0.05 else "⚠️  Statistically insignificant"
    print(f"{col:.<50} F={f_stat:>6.2f}, p={p_val:.4f} {significance}")

print("\n  p < 0.05 → Clusters show a statistically significant difference in this feature.")
print("  High F-statistic → Large variance between clusters. ")
```

✓ 0.0s

```
==========================================================================================
                              📊 ANOVA TEST
==========================================================================================

 Feature-wise differences across clusters

agriculture_value_added_pct....................... F= 86.22, p=0.0000 ✅ Statistically significant
life_expectancy................................... F=117.05, p=0.0000 ✅ Statistically significant
urban_population_pct.............................. F= 42.72, p=0.0000 ✅ Statistically significant
rural_electricity_access.......................... F=141.28, p=0.0000 ✅ Statistically significant

  p < 0.05 → Clusters show a statistically significant difference in this feature.
  High F-statistic → Large variance between clusters.
```

We used a one-way ANOVA test to check if the two clusters are statistically different.

All four indicators — agriculture, life expectancy, urbanization, and electricity access —

showed significant differences between clusters,

confirming that our clustering is statistically valid