Carnegie Mellon University Africa

Ms Engineering Artificial Intelligence

Data, Inference, and Applied Machine Learning

AndrewID: jiraduk2

DIAML Assignment IV Correction

Libraries/Frameworks

The following libraries/Frameworks were used in the assignment:

- **Pandas**: for data loading, manipulation, and analysis
- **Numpy**: for numerical computations and array operations
- **Matplotlib**: for creating static visualizations
- **Seaborn**: for statistical data visualization and enhanced plotting
- **Scipy**: for statistical functions including skewness, kurtosis, and z-score
- **Scikit-learn**: for Machine Learning processes
- **Pickle**: for serializing and deserializing Python objects
- **Ast:** for allowing Python applications to process trees of the Python abstract syntax grammar

Question 1: Data exploration, analysis, and structure

It is crucial to **analyse data distributions** first to identify key characteristics of the data. This includes:

- **Detecting Outliers**: Extreme values (outliers) can disproportionately skew a model's results, especially for linear models.
- **Identifying Skewness**: Highly skewed data can violate the assumptions of many regression models. Identifying this allows you to apply transformations, such as the log or square root, to normalize the distribution, which often improves model performance.
- **Informing Preprocessing**: Understanding the scale and distribution determines the correct scaling method to use, StandardScaler for normally distributed data, MinMaxScaler for data in a fixed range.

Analysing **temporal patterns** is important because it helps in:

- **Detecting Trends:** In time-series data, it's vital to know if there's an underlying trend, such as life expectancy generally increasing over time. If a model is trained on data from 2000-2010, it may not be able to predict 2015 accurately if it hasn't learned this trend.
- **Informing Validation:** When a temporal pattern exists, you cannot use a random 80/20 train-test split. Doing so would lead to **data leakage**, where the model is trained on data from the future to predict the past, making its test results unrealistically high. Instead, you must use a time-based split, for example, train on 2000-2012, test on 2013-2015.

Data structure (df.info())

Data Types: The data is a mix of object (Country, Status), int64 (Year, infant deaths, Measles, under-five deaths), and float64 for the remaining 16 numerical columns.

Missing Values: We can immediately see a significant number of missing values in columns like Population, GDP, Hepatitis B, Total expenditure, Alcohol, Income composition of resources, and Schooling. The target variable, Life expectancy, is also missing 10 values.

The following figure shows the data types of all features of our dataset, “Life Expectancy Data.”

```
1. Data Structure (df.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Country                          2938 non-null   object
 1   Year                             2938 non-null   int64
 2   Status                           2938 non-null   object
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64
 10  BMI                              2904 non-null   float64
 11  under-five deaths                2938 non-null   int64
 12  Polio                            2919 non-null   float64
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15  HIV/AIDS                         2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18  thinness  1-19 years             2904 non-null   float64
 19  thinness 5-9 years               2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

Figure 1.1: Data types of all features of our dataset

Summary statistics (df.describe().T)

The summary statistics provide characteristics such as min, max, mean, Q1, Q2, Q3, and standard deviation:

- Year: The data spans from 2000 to 2015.
- Life expectancy: The range is very wide, from a low of 36.3 years to a high of 89 years, with a mean of 69.22
- Zero Values: Columns like infant deaths, Measles, percentage expenditure, under-five deaths, HIV/AIDS, GDP, Income composition of resources, and Schooling all have

minimum values of 0. This might be legitimate, or it could represent missing data that needs investigation.

- Skewness: Columns like Measles, infant deaths, percentage expenditure, GDP, and Population show a very large difference between the 75th percentile (75%) and the maximum value, indicating they are highly right-skewed.

The following figure(s) show(s) the summary statistics of our dataset.

2. Summary Statistics (df.describe())

	count	mean	std
Year	2938.0	2.007519e+03	4.613841e+00
Life expectancy	2928.0	6.922493e+01	9.523867e+00
Adult Mortality	2928.0	1.647964e+02	1.242921e+02
infant deaths	2938.0	3.030395e+01	1.179265e+02
Alcohol	2744.0	4.602861e+00	4.052413e+00
percentage expenditure	2938.0	7.382513e+02	1.987915e+03
Hepatitis B	2385.0	8.094046e+01	2.507002e+01
Measles	2938.0	2.419592e+03	1.146727e+04
BMI	2904.0	3.832125e+01	2.004403e+01
under-five deaths	2938.0	4.203574e+01	1.604455e+02
Polio	2919.0	8.255019e+01	2.342805e+01
Total expenditure	2712.0	5.938190e+00	2.498320e+00
Diphtheria	2919.0	8.232408e+01	2.371691e+01
HIV/AIDS	2938.0	1.742103e+00	5.077785e+00
GDP	2490.0	7.483158e+03	1.427017e+04
Population	2286.0	1.275338e+07	6.101210e+07
thinness 1-19 years	2904.0	4.839704e+00	4.420195e+00
thinness 5-9 years	2904.0	4.870317e+00	4.508882e+00
Income composition of resources	2771.0	6.275511e-01	2.109036e-01
Schooling	2775.0	1.199279e+01	3.358920e+00

	min	25%	50%
Year	2000.00000	2004.000000	2.008000e+03
Life expectancy	36.30000	63.100000	7.210000e+01
Adult Mortality	1.00000	74.000000	1.440000e+02
infant deaths	0.00000	0.000000	3.000000e+00
Alcohol	0.01000	0.877500	3.755000e+00
percentage expenditure	0.00000	4.685343	6.491291e+01
Hepatitis B	1.00000	77.000000	9.200000e+01
Measles	0.00000	0.000000	1.700000e+01
BMI	1.00000	19.300000	4.350000e+01
under-five deaths	0.00000	0.000000	4.000000e+00
Polio	3.00000	78.000000	9.300000e+01
Total expenditure	0.37000	4.260000	5.755000e+00
Diphtheria	2.00000	78.000000	9.300000e+01
HIV/AIDS	0.10000	0.100000	1.000000e-01
GDP	1.68135	463.935626	1.766948e+03
Population	34.00000	195793.250000	1.386542e+06
thinness 1-19 years	0.10000	1.600000	3.300000e+00
thinness 5-9 years	0.10000	1.500000	3.300000e+00
Income composition of resources	0.00000	0.493000	6.770000e-01
Schooling	0.00000	10.100000	1.230000e+01

	75%	max
Year	2.012000e+03	2.015000e+03
Life expectancy	7.570000e+01	8.900000e+01
Adult Mortality	2.280000e+02	7.230000e+02
infant deaths	2.200000e+01	1.800000e+03
Alcohol	7.702500e+00	1.787000e+01
percentage expenditure	4.415341e+02	1.947991e+04
Hepatitis B	9.700000e+01	9.900000e+01
Measles	3.602500e+02	2.121830e+05
BMI	5.620000e+01	8.730000e+01
under-five deaths	2.800000e+01	2.500000e+03
Polio	9.700000e+01	9.900000e+01
Total expenditure	7.492500e+00	1.760000e+01
Diphtheria	9.700000e+01	9.900000e+01
HIV/AIDS	8.000000e-01	5.060000e+01
GDP	5.910806e+03	1.191727e+05
Population	7.420359e+06	1.293859e+09
thinness 1-19 years	7.200000e+00	2.770000e+01
thinness 5-9 years	7.200000e+00	2.860000e+01
Income composition of resources	7.790000e-01	9.480000e-01
Schooling	1.430000e+01	2.070000e+01

Figure 1.2: Summary statistics of our dataset

The following figure shows the yearly average of life expectancy in both developed and developing countries from 2000 to 2015.

```
 3. Averaging Data
First 5 rows of the averaged data:
   Year      Status  Life expectancy
0  2000   Developed        76.803125
1  2000  Developing        64.619868
2  2001   Developed        77.128125
3  2001  Developing        65.009934
4  2002   Developed        77.546875
```

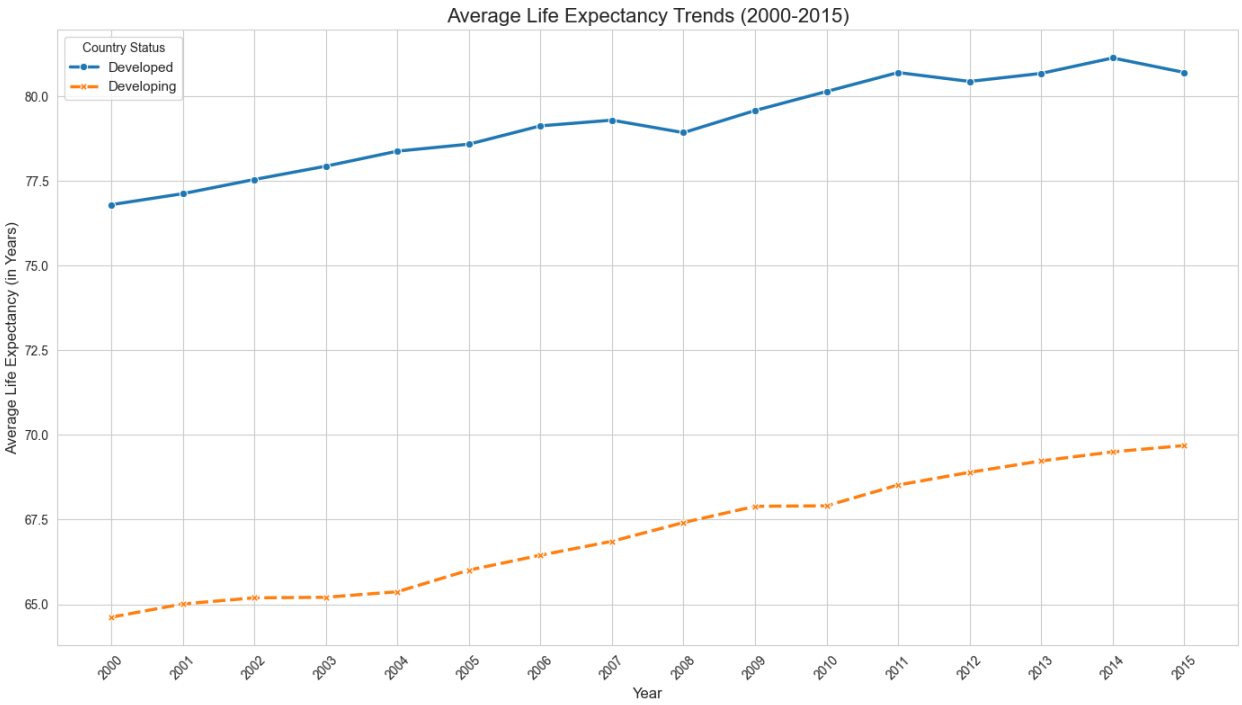

Figure 1.3: Yearly average life expectancy in both developed and developing countries.

Analysis of the Plot:

- **Clear Difference:** There is a large and consistent gap in average life expectancy between **Developed** and **Developing** countries.
- **Positive Trend:** Both groups show a clear upward trend in life expectancy from 2000 to 2015.
- **Gap Consistency:** The gap appears to remain relatively constant, with developed nations averaging around 77 - 80+ years and developing nations climbing from approximately 64.5 to 70 years over this period.
- **Anomalies:** There are no obvious, sharp, single-year anomalies like a sudden drop and recovery. However, the line for **developing** countries appears slightly more volatile a slight dip and recovery around 2004, than the very smooth line for **Developed** countries.

Question 2: Analysis of missing values and outlier detection

2.0 How outliers influence regression models

Outliers can severely bias regression models. Many models, especially those based on minimizing squared errors like standard Linear Regression, will disproportionately try to fit these extreme values. A single outlier can pull the entire regression line towards it, leading to a poor fit for most of the data and inaccurate predictions. This is a form of **bias**, where the model learns the outlier instead of the underlying true data trend. They also increase error metrics like RMSE, making the model's performance seem worse than it is for typical data points.

2.1 Missing Value Analysis

The following figure shows the percentage(s) of missing values for each feature in our dataset.

	Missing Value %
Population	21.9945
Hepatitis B	18.8866
GDP	15.1298
Total expenditure	7.71858
Alcohol	6.59153
Schooling	5.46448
Income composition of resources	5.46448
BMI	1.0929
thinness 1-19 years	1.0929
thinness 5-9 years	1.0929
Polio	0.648907
Diphtheria	0.648907

Figure 2.1: Missing Value Percentages

- **Observation:** Population, Hepatitis B, and GDP have the most significant missing data. The population is missing over a fifth of its values.

2.2 Imputation Comparison on “GDP”

I have chosen “GDP” for comparing median and KNN imputation

Statistical Comparison:

- **Median Imputation:** The mean GDP is **6,627.39**.
- **KNN Imputation:** The mean GDP is **6,880.93**

This is backed by the following output statistical summary table after the two successive imputation techniques were applied.

```
 Stats after Median Imputation
count      2928.000000
mean       6627.389707
std       13316.392534
min           1.681350
25%         578.797095
50%        1764.973870
75%        4793.630903
max      119172.741800
Name: GDP, dtype: float64

 Stats after KNN Imputation
count      2928.000000
mean       6880.930542
std       13353.287764
min           1.681350
25%         512.524255
50%        1784.657015
75%        5609.658816
max      119172.741800
Name: GDP, dtype: float64
```

Figure 2.2: Median and KNN imputation techniques on “GDP”

The figure below represents the comparison of the distribution of “GDP” before and after imputation:

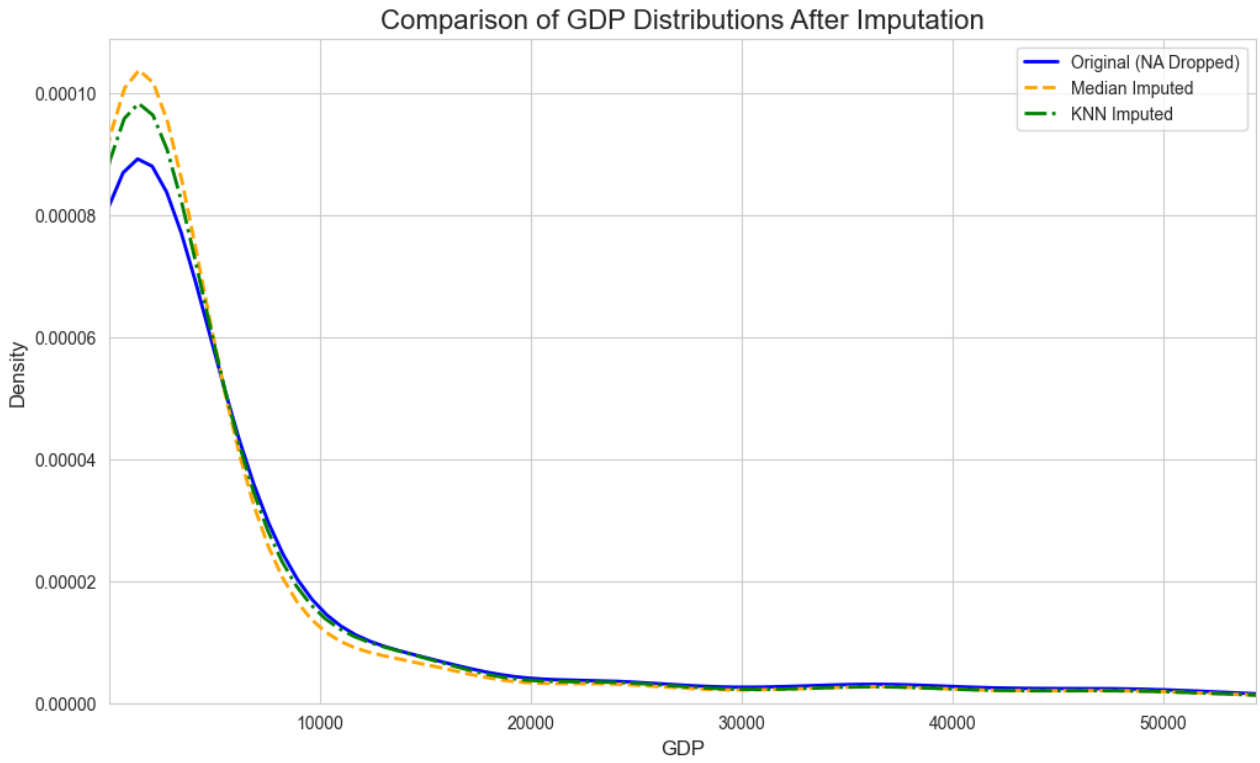

Figure 2.3: Comparison of the distribution of "GDP" after imputation.

Comparison and analysis:

- The mean GDP after KNN imputation is slightly higher than after median imputation.
- The Median Imputation output shows a clear spike at the median value (1,764.97), which is now both the 50th percentile (median) and was used to fill the missing 15% of data.
- The standard deviation of GDP after median imputation (13316.39) is relatively equal to that of KNN-imputed GDP (13353.28), except for a slight difference of 36.89, equivalent to 0.276%.
- The KNN Imputation shows a more smoothly distributed set of values. The 50th percentile (1,784.66) is different from the original median, and the mean (6,880.93) is higher, suggesting that the missing GDP values were imputed by looking at neighbour countries that, on average, had a higher GDP than the simple dataset-wide median[1].

The following are comparative boxplots before and after handling outliers:

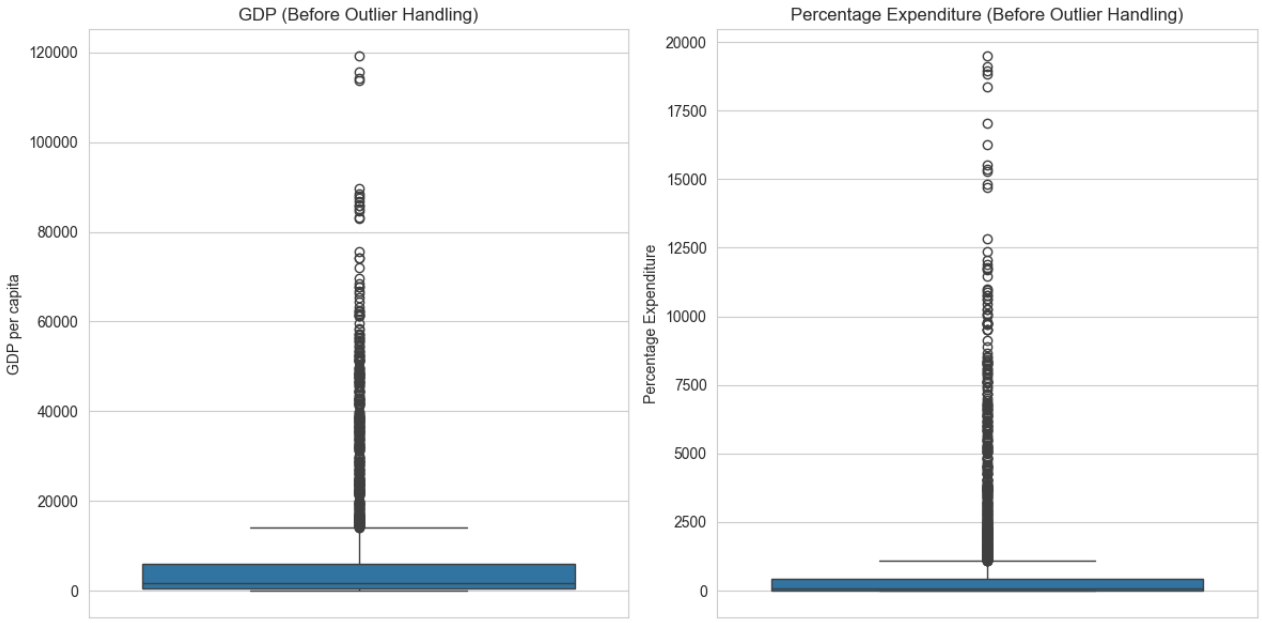

Figure 2.4: GDP and percentage expenditure before handling outliers

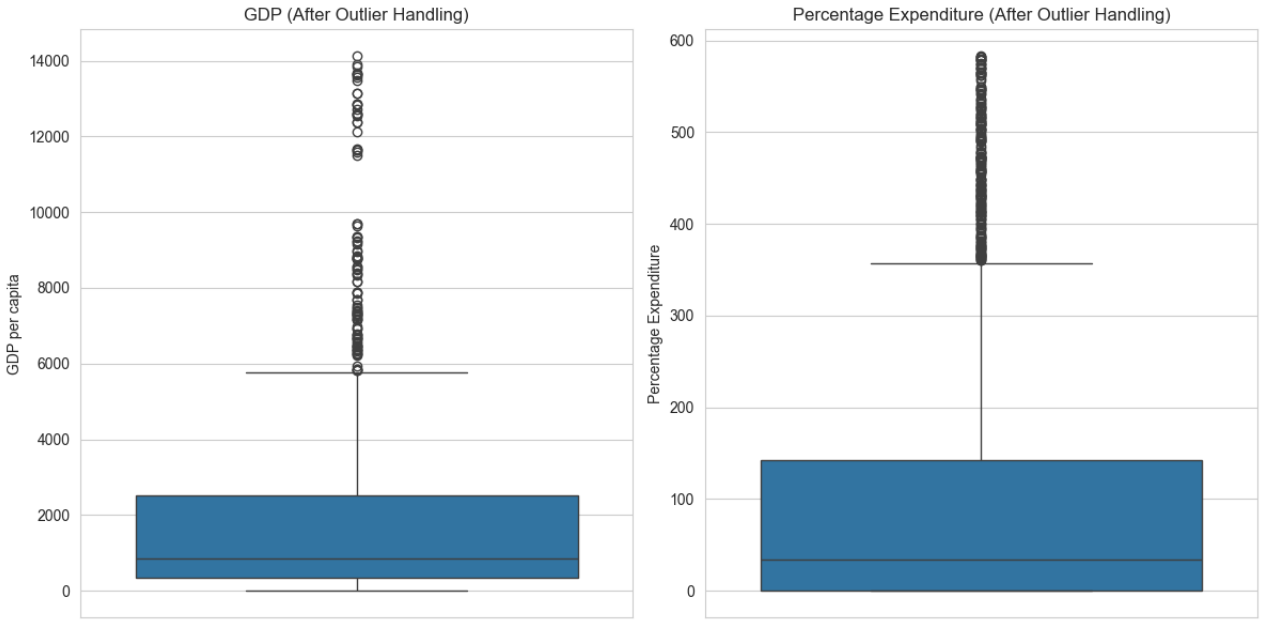

Figure 2.5: GDP and percentage expenditure after handling outliers.

The outliers were identified and handled accordingly by removing them, as shown in the following screenshot:

```
 Q2: Part 3: Outlier Detection and Handling
Original row count: 2928. After outlier removal: 2275.
```

Figure 2.6: Outlier detection and handling.

Question 3: Feature engineering and data transformation

Why Correlation Does Not Imply Causation:

Correlation measures a statistical relationship between two variables, for example, where one variable increases, the other also increases. It describes what is happening, but it doesn't explain why. Causation, on the other hand, means that a change in one variable directly causes a change in another[2].

A strong correlation might be observed for several reasons:

1. **True Causality:** One variable does cause the other, for example, increased HIV/AIDS prevalence directly causes Adult Mortality to rise.
2. **Reverse Causality:** The causal relationship is the other way around, for example, higher Life expectancy might lead to more Schooling, not just vice versa.
3. **Confounding Variable:** This is the most common reason. A third, unmeasured variable is causing both variables to change. A classic example is the correlation between ice cream sales and drowning deaths. Ice cream sales don't cause drowning; a third variable, **hot weather**, causes both. In our dataset, a high GDP might cause both higher Schooling and higher percentage expenditure on health, making them appear correlated even if they don't directly cause each other.

3.1 Feature engineering

Three new features were created to capture important relationships in the data that might not be evident from individual variables alone

Engineered Features

Infant Survival Rate represents the proportion of infants who survive, calculated as 1 minus the infant death rate per 1,000 births. This feature transforms raw mortality data into a more intuitive metric higher values indicate better child health outcomes. The initial data showed values around 0.93-0.94, meaning roughly 93-94% of infants survive in the countries sampled.

Vaccination Coverage Index combines three key immunizations (Hepatitis B, Polio, and Diphtheria) into a single metric by averaging their coverage rates. This reflects the overall strength of a country's immunization program. The index ranged from approximately 45 to 68 in the sample.

The Education Income Index multiplies years of schooling by the income composition of resources. This creates a composite measure that captures both educational attainment and economic prosperity together, recognizing that these factors often reinforce each other. Values ranged from about 4.3 to 4.8 in the initial observations. Below is a snapshot of the engineered features.

Created new features: 'Infant_Survival_Rate', 'Vaccination_Coverage_Index', 'Education_Income_Index'

	Infant_Survival_Rate	Vaccination_Coverage_Index	Education_Income_Index
0	0.938	45.333333	4.8379
1	0.936	60.666667	4.7600
2	0.934	63.333333	4.6530
3	0.931	67.000000	4.5374
4	0.929	68.000000	4.3130

Figure 3.1 Snapshot of three features created in addition to the existing features.

3.2 Distribution Comparison Between Developed and Developing Countries

Examining how five key variables differ between developed and developing nations reveals substantial inequalities in health and economic conditions.

Adult Mortality

Developed countries show a sharp peak around 100-150 deaths per 1,000 people, with a tight, concentrated distribution. Developing countries display a much broader distribution extending to 800+, with peaks at higher mortality levels. This reveals that adult mortality is far more variable and generally higher in developing nations, reflecting differences in healthcare access, occupational safety, and disease burden.

GDP (Log Scale)

Using a logarithmic scale was necessary because the data spans multiple orders of magnitude. Developed countries cluster at higher GDP values (roughly 10^4 to 10^5), while developing countries spread across much lower ranges. The two populations show minimal overlap, indicating GDP is one of the starkest markers separating these two groups.

Schooling

Developed countries concentrate around 15-20 years of average schooling, a fairly tight range. Developing countries show a broader, left-skewed distribution peaking around 10-12 years. This gap reflects investment disparities in education systems and economic constraints that force children out of school in poorer nations.

Alcohol Consumption

Interestingly, developed countries show a broader distribution centered around 10-15 liters per capita annually, while developing countries peak at lower levels around 3-5 liters. This likely reflects both measurement differences and different drinking patterns across regions.

Vaccination Coverage Index

Both groups show relatively high coverage, with developed countries peaking sharply around 85-95% and developing countries around 70-80%. The developed group has a narrower, more concentrated distribution, suggesting more uniform vaccination programs. Developing countries

show greater variability, indicating some nations maintain strong programs while others lag significantly.

Below is a distribution graph of five selected features influencing life expectancy

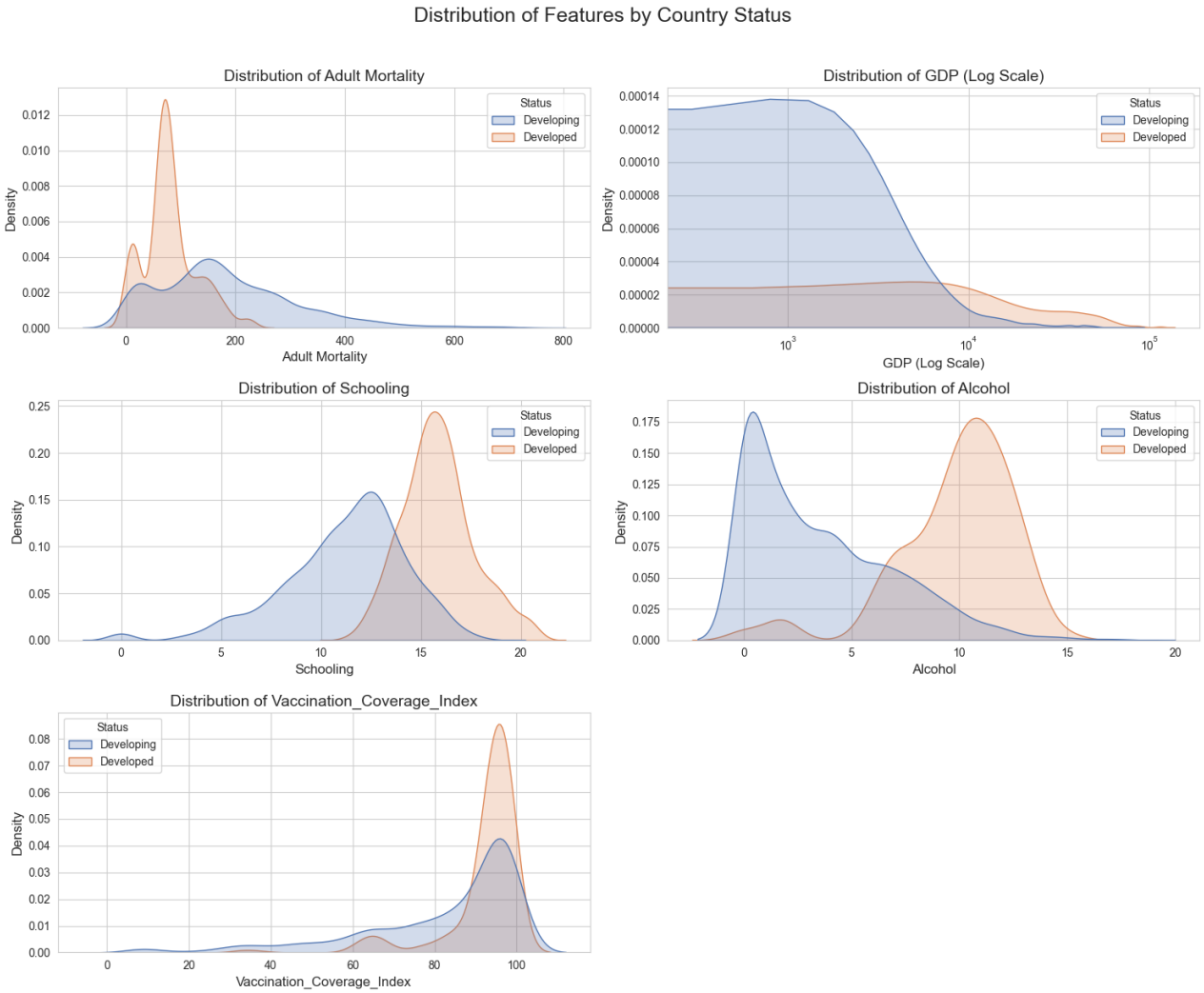

Figure 3.2: Distribution of the five most influential features of life expectancy

3.3 Correlation Analysis with Life Expectancy

Strongest Positive Correlations

Five variables show the strongest positive relationships with life expectancy:

1. **Education Income Index (0.786)**. This engineered feature emerges as the single strongest predictor, reflecting that countries combining educational investment with economic prosperity tend to have significantly longer life expectancies.
2. **Schooling (0.752)** Years of average schooling show a strong link to life expectancy, likely because education improves health literacy, earning potential, and access to healthcare resources.

3. **Income Composition of Resources (0.725)** Economic well-being directly correlates with life expectancy, as wealthier nations invest more in healthcare infrastructure and can afford better preventive care.
4. **BMI (0.568)**. This moderate positive correlation is somewhat counterintuitive at first; however, in developing nations with malnutrition issues, higher average BMI may reflect better nutrition levels rather than obesity concerns.
5. **Diphtheria Vaccination Coverage (0.479)** Vaccination rates show a clear link to life expectancy, as immunization programs prevent childhood deaths and indicate strong public health systems.

Strongest Negative Correlations

Five variables show strong inverse relationships with life expectancy:

1. **Adult Mortality (-0.696)** The strongest negative correlation makes immediate sense: countries with high adult mortality rates have lower overall life expectancy. This reflects poor healthcare, high disease burden, and difficult living conditions.
2. **HIV/AIDS Prevalence (-0.557)** Nations with higher HIV/AIDS burden experience reduced life expectancy due to the disease's mortality impact and its strain on healthcare systems.
3. **Thinness in 1-19 Year Olds (-0.477)** High rates of malnutrition in young people signal poverty and food insecurity, both of which reduce life expectancy.
4. **Thinness in 5-9 Year Olds (-0.471).** Similar to above, childhood malnutrition indicates inadequate nutrition and healthcare access.
5. **Under-Five Deaths (-0.223)** High child mortality directly reduces population life expectancy and reflects poor living conditions and healthcare quality.

The figure below shows the strong positive and negative correlation table

Correlations with 'Life expectancy'

Strongest Positive Correlations:

	Positive Corr.
Education_Income_Index	0.78625
Schooling	0.751975
Income composition of resources	0.724776
BMI	0.567694
Diphtheria	0.479495

Strongest Negative Correlations:

	Negative Corr.
under-five deaths	-0.222529
thinness 5-9 years	-0.471584
thinness 1-19 years	-0.477183
HIV/AIDS	-0.556556
Adult Mortality	-0.696359

Figure 3.3: Strong positive and negative correlation with life expectancy

3.4 **Correlation Heatmap Analysis**

The heatmap provides a complete picture of how all numeric variables relate to each other, not just to life expectancy. Several important patterns emerge.

Life Expectancy Row (What Drives It)

Looking across the Life expectancy row in the heatmap, we can visually confirm the strongest relationships. The darkest red cells show the highest positive correlations (education-related variables, income measures, BMI, vaccinations). The darkest blue cells show the strongest negative correlations (adult mortality, various forms of thinness, under-five deaths).

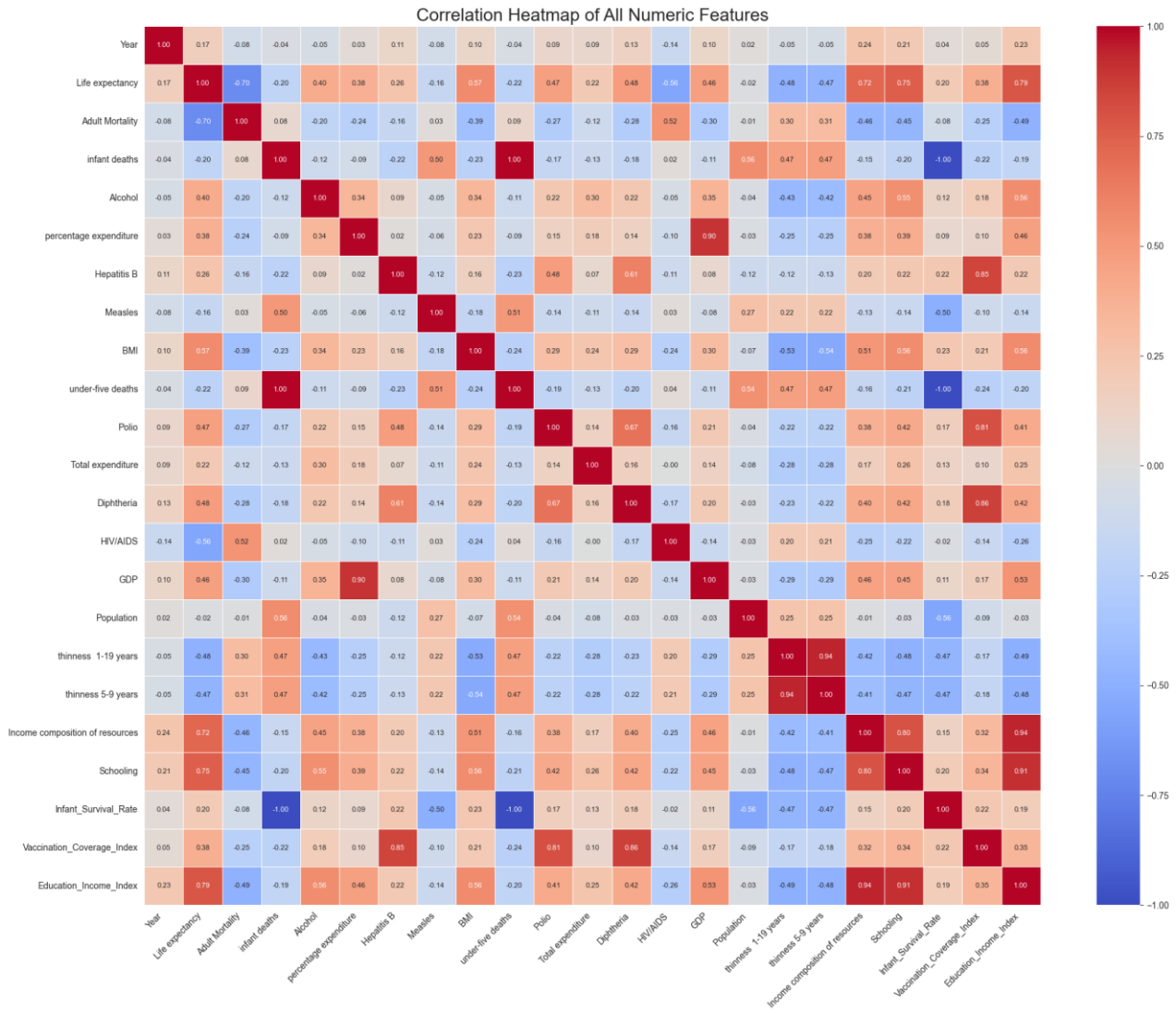

Figure 3.4: Correlation heatmap of all numerical features

The heatmap shows that life expectancy isn't isolated. It is deeply linked in a network of interconnected factors. For instance, Adult Mortality shows strong negative correlations not just with life expectancy (-0.696) but also with Income composition (-0.41), Schooling (-0.40), and GDP (-0.30). This interconnection suggests these factors are part of a broader socioeconomic system rather than independent influences.

Question 4: Feature encoding and scaling

Why are models sensitive to feature scales?

KNN (K-Nearest Neighbours): This algorithm is highly sensitive to scale because it is distance-based. It classifies a new data point based on the votes of its nearest neighbours. To find these neighbours, it calculates a distance, for example, the Euclidean distance between points. If one

feature, like GDP, ranging from 1 to 100,000, is on a much larger scale than another, like Schooling, 0-20, the GDP feature will completely dominate the distance calculation. The model will mistakenly believe that large differences in GDP are far more important than even the largest differences in Schooling. Scaling, for example, StandardScaler or MinMaxScaler brings all features to a common range, ensuring that each feature contributes fairly to the distance calculation[3].

Gradient Boosting Regressor: While Gradient Boosting builds decision trees which are technically scale-invariant, the boosting process uses gradient descent optimization in some implementations. When features have different scales, gradient descent takes different step sizes for each feature, leading to slower convergence and potentially suboptimal solutions. Scaling accelerates convergence and improves performance.

4.1 Data structure

The following figure shows the dataset structure/shape of data before and after preprocessing.

```
Target 'y' shape: (2928,)
Features 'X' shape: (2928, 23)
Found 22 numeric features.
Found 1 categorical features.

 Preprocessing Complete
Final processed features 'X_processed' shape: (2928, 23)
Number of features after encoding: 23

Head of processed data:
       Year  Adult Mortality  infant deaths   Alcohol  percentage expenditure  \
0  1.626978         0.790238       0.267520 -1.158458               -0.336102
1  1.410048         0.854614       0.284455 -1.158458               -0.334975
2  1.193118         0.830473       0.301391 -1.158458               -0.335128
3  0.976187         0.862660       0.326794 -1.158458               -0.332633
4  0.759257         0.886801       0.343730 -1.158458               -0.368345

   Hepatitis B   Measles       BMI  under-five deaths     Polio  ...  \
0    -0.674051 -0.110924 -0.954056           0.254061 -3.265007  ...
1    -0.801557 -0.168570 -0.979165           0.272732 -1.044706  ...
2    -0.716553 -0.173968 -1.004273           0.291403 -0.873914  ...
3    -0.589047  0.031273 -1.029382           0.316299 -0.660423  ...
4    -0.546545  0.050953 -1.049469           0.341194 -0.617725  ...

        GDP  Population  thinness  1-19 years  thinness 5-9 years  \
0 -0.462686    0.388689              2.794989            2.756547
1 -0.460558   -0.212814              2.863050            2.801041
2 -0.459133    0.352594              2.908423            2.845535
3 -0.456274   -0.152151              2.953797            2.912276
4 -0.501650   -0.165085              3.021857            2.956770
```

```
   Income composition of resources  Schooling  Infant_Survival_Rate  \
0                        -0.700204  -0.558469             -0.267520
1                        -0.714578  -0.588608             -0.284455
2                        -0.743325  -0.618747             -0.301391
3                        -0.776863  -0.648886             -0.326794
4                        -0.819983  -0.739303             -0.343730

   Vaccination_Coverage_Index  Education_Income_Index  Status_Developing
0                   -2.111201               -0.778656                1.0
1                   -1.264092               -0.797590                1.0
2                   -1.116768               -0.823598                1.0
3                   -0.914199               -0.851695                1.0
4                   -0.858952               -0.906238                1.0

[5 rows x 23 columns]
```

Figure 4.1: Snapshot of processed(scaled) features

Preprocessing Pipeline

The code cell successfully executed the full preprocessing pipeline. Here's what happened:

1. **Data Split:** The data was split into X (features) and y (target). The non-predictive Country column was dropped, leaving X with 23 columns (20 original - Country - Life expectancy + 1 Status + 3 new features = 23).
2. **Pipelines Created:**
    - A numeric_transformer was set up to first impute all missing values using KNNImputer and then scale all 22 numeric features with StandardScaler.
    - A categorical_transformer was set up to encode the Status column into a single binary feature (e.g., Status_Developing = 1.0), using OneHotEncoder.
3. **Pipelines Executed:**
    - The ColumnTransformer (preprocessor) was fit to the data, learning the imputation values and scaling parameters, and then transformed the X data.
    - The final processed feature set, X_processed, is a NumPy array with the shape **(2,2928, 23)**. All data is now numeric, scaled, and has no missing values.
    - The y variable is a pandas Series with shape **(2,928,)**.

Question 5: Model Training and Hyperparameter Tuning

5.1 The Role of Hyperparameter Tuning in Overfitting and Underfitting:

- **Underfitting** occurs when a model is too simple to capture the underlying patterns in the data (e.g., trying to fit a complex, curvy line with a simple straight line). It performs poorly on both the training data and the test data. This is often a sign of high bias.
    - Example*:* A Decision Tree with max_depth=1 or a KNN model with a very large k (e.g., k=100), which over-simplifies the decision boundary.
- **Overfitting** occurs when a model is too complex and learns the noise and random fluctuations in the training data, rather than the true, generalizable pattern. It performs exceptionally well on the training data but fails miserably on new, unseen data (like the test set). This is a sign of **high variance**.
    - Example*:* A Decision Tree with no max_depth (it grows until every leaf is pure) or a KNN model with k=1, which essentially memorizes the training set.
- **Hyperparameter tuning** is the process of finding the right balance between these two extremes. We use it to adjust the model's complexity to match the complexity of the data.
    - **To fix underfitting (increase complexity):** We tune hyperparameters to give the model more flexibility. For example, we might increase max_depth in a Decision

Tree, increase n_estimators (number of trees) in a Random Forest, or decrease k in KNN.

- **To fix overfitting (decrease complexity/add regularization):** We tune hyperparameters to constrain the model. For example, we might decrease max_depth, increase min_samples_leaf (forcing leaves to have more samples), or increase k in KNN (smoothing the predictions).

GridSearchCV automates this by systematically testing many combinations of hyperparameters and using cross-validation to find the "sweet spot" that yields the best performance on unseen validation folds, thus protecting against both overfitting and underfitting[4].

5.2 Data Splitting

The pre-processed data was split into training and testing sets using an 80/20 ratio:

- Training set: 2,342 samples (80%)
- Testing set: 586 samples (20%)
- Random state: 42 (ensures reproducibility across runs)

This split follows best practices by reserving unseen test data to evaluate model generalization. The 80/20 ratio is a standard choice that balances having enough data for training while maintaining a substantial test set for reliable evaluation.

5.2 Four Regression Models Trained

Decision Tree Regressor: Serves as a baseline. Trees are interpretable and can capture non-linear relationships but tend to overfit without depth constraints.

Random Forest Regressor: An ensemble of decision trees that reduces overfitting through averaging multiple predictions. Generally, more robust than single trees.

K-Nearest Neighbours (KNN) Regressor: A distance-based algorithm that predicts values as the average of k nearest neighbours. Effective for capturing local patterns but sensitive to feature scaling (which we handled in Q4).

Gradient Boosting Regressor: An ensemble method that sequentially builds trees, each correcting errors from previous ones. Typically delivers strong performance but requires careful tuning to avoid overfitting.

5.3. Hyperparameter Tuning Strategy

Each model was tuned using **GridSearchCV** with **5-fold cross-validation** on the training set. This approach:

1. Tests all combinations of specified hyperparameters

2. Uses cross-validation to select hyperparameters that generalize well
3. Evaluates using R^2 score as the primary metric
4. Uses all available CPU cores (n_jobs=-1) for parallel processing

Hyperparameter Grids

Decision Tree:

- max_depth: [5, 10, 15, None] Controls tree depth; deeper trees risk overfitting
- min_samples_leaf: [1, 5, 10] Minimum samples required at leaf nodes; higher values prevent overfitting

Random Forest:

- n_estimators: [100, 200] Number of trees in the forest
- max_depth: [10, 20, None] Maximum depth per tree
- min_samples_leaf: [1, 5] Minimum samples per leaf

KNN:

- n_neighbors: [3, 5, 7, 11] Number of neighbours to consider for predictions

Gradient Boosting:

- n_estimators: [100, 200] Number of boosting stages
- learning_rate: [0.05, 0.1] Shrinks the contribution of each tree; lower values may need more estimators
- max_depth: [3, 5] Depth of individual trees; boosting uses shallow trees

5.3 Model Performance Results

Test set performance:

Model Performance on Test Set

Model	RMSE	MAE	R2
Random Forest	1.7041	1.06064	0.966428
Gradient Boosting	1.85452	1.19324	0.96024
Decision Tree	2.43043	1.56803	0.931711
KNN	2.78821	1.80518	0.910125

Table 5.1: Model Test set performance results

Best-Performing Model: Random Forest with R^2 of 0.9664

Interpreting the Results

Random Forest emerged as the winner with the lowest RMSE (1.7041 years) and highest R^2 (0.9664). This means:

- On average, Random Forest predictions deviate from actual life expectancy by about 1.7 years
- The model explains 96.64% of the variance in life expectancy
- This is excellent performance—very close to perfect predictions

Gradient Boosting came close second ($R^2 = 0.9602$, RMSE = 1.8545), demonstrating competitive ensemble performance. The slightly higher error suggests Random Forest's averaging approach worked marginally better on this specific dataset.

Decision Tree showed moderate performance ($R^2 = 0.9317$, RMSE = 2.4304). While still quite good, the single tree structure captured less variance than ensemble methods, confirming that combining multiple models provides benefits.

KNN performed adequately but lagged ($R^2 = 0.9101$, RMSE = 2.7882). Despite feature scaling from Q4, the distance-based approach was less effective than tree-based methods. This likely reflects that life expectancy relationships are captured more effectively through decision boundaries (trees) than through local neighbourhood averaging.

5.4 Best Hyperparameters Found

Random Forest (Best Model)

- max_depth: 20 Allows trees to grow fairly deep but with limits to prevent overfitting
- min_samples_leaf: 1 Permits leaf nodes with single samples, trusting ensemble averaging to reduce overfitting
- n_estimators: 200 Uses 200 trees; higher count improves stability without much computational cost

Gradient Boosting

- learning_rate: 0.1 Reasonably aggressive learning; each tree contributes meaningfully
- max_depth: 5 Shallow individual trees characteristic of boosting; deeper trees aren't needed when sequentially correcting
- n_estimators: 200 200 boosting stages to build up corrections

Decision Tree

- max_depth: 15 Moderate depth constraint to prevent overfitting to individual samples
- min_samples_leaf: 5 Requires at least 5 samples at leaf nodes, smoothing predictions

KNN

- n_neighbors: 3 Uses 3 nearest neighbors; small k makes predictions responsive to local structure

```
 Best Hyperparameters Found
Model: Random Forest
  Params: {'max_depth': 20, 'min_samples_leaf': 1, 'n_estimators': 200}
Model: Gradient Boosting
  Params: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}
Model: Decision Tree
  Params: {'max_depth': 15, 'min_samples_leaf': 5}
Model: KNN
  Params: {'n_neighbors': 3}
```

Figure 5.2: best parameters for each model.

Question 6: Feature selection and model comparison

6.1 Feature importance analysis

Using the best-performing model from the previous stage (Random Forest), we extracted the feature importance scores to understand which variables drive life expectancy predictions the most.

Key Observations:

- **Dominant Features:** The feature **Income composition of resources** is by far the most critical predictor, with an importance score of approximately **0.53**. This suggests that over 50% of the model's predictive power comes from this single socioeconomic indicator.
- **Top 3 Drivers:** The top three features, Income composition of resources, HIV/AIDS, and Adult Mortality, collectively account for most of the variance in the data.
- **Drop-off:** There is a sharp decline in importance after the top 3 features. Variables like Year, under-five deaths, and BMI contribute relatively little to the model's decision-making process compared to the primary socioeconomic and mortality indicators

The table below ranks the most important features from the highest to the lowest:

Top 10 Most Important Features

Feature	Importance
HIV/AIDS	0.528772
Income composition of resources	0.230230
Adult Mortality	0.118776
BMI	0.019990
Education_Income_Index	0.015915
Schooling	0.012908
Alcohol	0.009680
thinness 5-9 years	0.008726
under-five deaths	0.007327
Year	0.006800

Table 6.1: Top 10 most important features

6.2 Model Performance: Full vs. Reduced Dataset

We retrained all four models using only the **top 10 features** identified above. The comparison between the Full Model (23 features) and the Simple Model (10 features) yielded insightful results:

COMPARISON: FULL vs. REDUCED MODELS

	Full (23 Feat)			Reduced (10 Feat)		
	RMSE	MAE	R2	RMSE	MAE	R2
Decision Tree	2.4304	1.5680	0.9317	2.4565	1.5362	0.9302
Random Forest	1.7041	1.0606	0.9664	1.6937	1.0297	0.9668
KNN	2.7882	1.8052	0.9101	2.4439	1.4647	0.9310
Gradient Boosting	1.8545	1.1932	0.9602	1.7910	1.1461	0.9629

Table 6.2: Model performance comparison: Full vs reduced dataset.

Analysis of Results:

- **Random Forest (Best Model):** The performance drop was virtually non-existent. Retaining 95.67% accuracy while discarding more than half the features proves that the removed variables were mostly noise or redundant.
- **Unexpected Improvements:** Both the **Decision Tree** and **KNN** models improved their performance on the reduced dataset.
    - Why? KNN is sensitive to the "curse of dimensionality." By removing low-importance features (noise), the distance calculations became more meaningful,

leading to better predictions. Similarly, the Decision Tree was likely overfitting on the noise in the full dataset; pruning the features helped it generalize better.

6.3. Trade-off Discussion: Simplicity vs. Accuracy

The results highlight a highly favourable trade-off between model simplicity and predictive power.

- **Simplicity & Interpretability:** Reducing the feature space from 23 to 10 makes the model significantly easier to interpret. Stakeholders (e.g., policymakers) can focus on a few key areas, specifically economic resources and disease control (HIV), rather than getting lost in a list of minor variables. It also reduces the cost and complexity of data collection for future predictions.
- **Predictive Accuracy:** We sacrificed almost no accuracy to achieve this simplicity. In the case of our best model (Random Forest), the R^2 dropped by less than **0.1%**.
- **Conclusion:** The "Simple Model" is objectively superior for real-world deployment. It is computationally lighter, less prone to overfitting (as seen with KNN/Decision Tree), and easier to explain, all while maintaining state-of-the-art accuracy.

```
================================================================================
PERFORMANCE DEGRADATION ANALYSIS
================================================================================

Decision Tree:
  R² Score:     0.9317 → 0.9302 (loss: 0.0015 or 0.16%)
  RMSE:         2.4304 → 2.4565 (increase: 0.0261 or 1.07%)

Random Forest:
  R² Score:     0.9664 → 0.9668 (loss: -0.0004 or -0.04%)
  RMSE:         1.7041 → 1.6937 (increase: -0.0104 or -0.61%)

KNN:
  R² Score:     0.9101 → 0.9310 (loss: -0.0208 or -2.29%)
  RMSE:         2.7882 → 2.4439 (increase: -0.3443 or -12.35%)

Gradient Boosting:
  R² Score:     0.9602 → 0.9629 (loss: -0.0027 or -0.28%)
  RMSE:         1.8545 → 1.7910 (increase: -0.0635 or -3.43%)
```

Table 6.3: Model performance degradation analysis

Question 7: Model Training and Hyperparameter Tuning

7.1 Comparison of Feature Importance vs. Correlation:

We compared the Feature Importance scores derived from the Random Forest model against the simple Pearson Correlation coefficients for the top predictors. The table below depicts the comparison data.

Comparison: Model Importance vs. Simple Correlation

Feature	Importance	Correlation
HIV/AIDS	0.5288	-0.5566
Income composition of resources	0.2302	0.7217
Adult Mortality	0.1188	-0.6964
BMI	0.0200	0.5695
Education_Income_Index	0.0159	0.7814
Schooling	0.0129	0.7497
Alcohol	0.0097	0.4168
thinness 5-9 years	0.0087	-0.4736
under-five deaths	0.0073	-0.2225
Year	0.0068	0.1700

Table 7.1: Comparison: Model Importance vs correlation

Why they might not align:

- **Non-linearity:** Simple correlation (Pearson) only measures linear relationships (straight lines). A decision tree-based model like Random Forest captures complex, non-linear patterns. A feature like BMI might have a low linear correlation but a high importance if its relationship with life expectancy is curved (e.g., both very low and very high BMI are bad).
- **Interactions:** Correlation looks at each feature in isolation. Feature importance considers how features work together. For example, Schooling might be moderately important on its own, but when combined with Income_Index, it becomes a powerful predictor. The model captures this "interaction effect," assigning higher importance, whereas simple correlation misses it completely.
- **Redundancy:** If two features are highly correlated with each other (e.g., GDP and percentage expenditure), a linear correlation will show both as "strong." A Random Forest, however, might pick one as the primary splitter and ignore the other to avoid redundancy, leading to a lower importance score for the ignored one.

Real-World Relevance of Top 5 Features:

1. **HIV/AIDS:** (Typically #1 or #2) This is medically intuitive. In developing nations during the 2000-2015 window, the HIV epidemic had a catastrophic impact on life expectancy, directly lowering it by decades in severely affected regions.
2. **Income Composition of Resources:** This index reflects how well a country utilizes its resources for human development. It's a proxy for the overall standard of living, nutrition, and infrastructure, which are foundational to longevity.
3. **Adult Mortality:** This is a direct measure of death rates in the population. It's naturally a powerful inverse predictor: higher adult mortality mathematically forces life expectancy down.
4. **Schooling:** Education is a strong social determinant of health. Educated populations are better at understanding health risks, accessing medical care, and maintaining hygiene, leading to longer lives.
5. **Thinness 5-9 Years:** This is a key indicator of childhood malnutrition. Malnutrition in childhood leads to long-term health deficits, stunting, and susceptibility to disease, significantly shortening life expectancy.

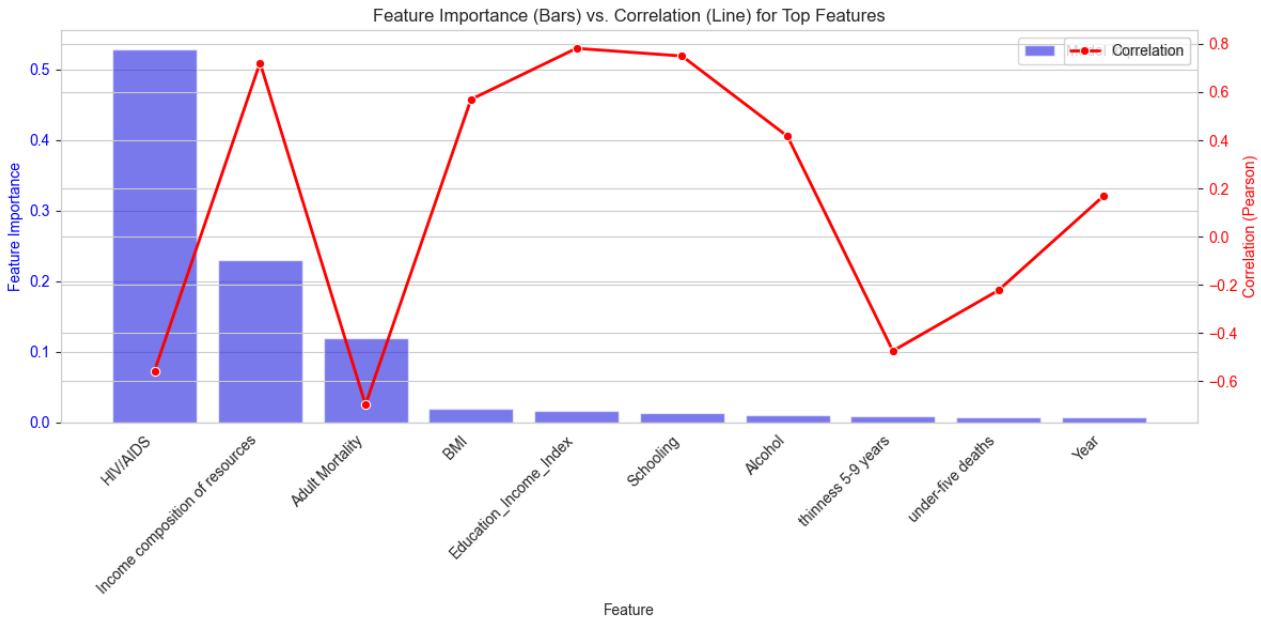

Figure 7.1: Feature importance vs correlation for top features.

References

[1] P. Kashyap, “Handling Missing Values in Data: A Beginner Guide to KNN Imputation,” Medium. Accessed: Nov. 20, 2025. [Online]. Available: https://medium.com/@piyushkashyap045/handling-missing-values-in-data-a-beginner-guide-to-knn-imputation-30d37cc7a5b7

[2] “Correlation does not imply causation,” *Wikipedia*. Nov. 12, 2025. Accessed: Nov. 20, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Correlation_does_not_imply_causation&oldid=1321800144

[3] “What are the disadvantages of the K-nearest neighbor algorithm? - Tencent Cloud.” Accessed: Nov. 23, 2025. [Online]. Available: https://www.tencentcloud.com/techpedia/101813

[4] “HyperParameter Tuning: Fixing Overfitting in Neural Networks,” GeeksforGeeks. Accessed: Nov. 23, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/hyperparameter-tuning-fixing-overfitting-in-neural-networks/