# LIFE EXPECTANCY ANALYSIS

Internship Report

Duration: 15/06/2025 – 15/07/2025

**Under the Guidance of:**

**[Drishti Madaan]**

**[HR Manager]**

**[Unified Mentor Pvt. Ltd]**

**[Gurugram, Haryana – 122002]**

**Project Done by:**


Killaka Sumalatha

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge Technologies, Srikakulam

# CERTIFICATE

This is to certify that the internship project report titled
**"Life Expectancy Analysis"**

submitted by **Killaka Sumalatha** during the internship period from 15/06/2025 to 15/07/2025 is a record of original work carried out under our supervision.

The report is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

**Project Guide:**
Drishti Madaan
HR Manager
Unified Mentor Pvt. Ltd

# DECLARATION

I, **Killaka Sumalatha**, declare that the internship project entitled

**"Life Expectancy Analysis"**,

is my own work and has been submitted for the purpose of fulfilling my academic requirements.

The content presented in this report is genuine and has not been submitted previously to any institution for any academic credit.

**Signature:**
K. Sumalatha

# ACKNOWLEDGEMENT

I am sincerely thankful to my guide, [Mentor's Name], for their continuous support, insightful feedback, and expert advice during the course of this internship.

I also extend my gratitude to the team at [Organization] for providing me with the opportunity to work on a real-world data analytics problem related to life expectancy.

My heartfelt thanks to my university for encouraging and facilitating this learning experience.

With gratitude,
K. Sumalatha

**ABSTRACT**

The "Life Expectancy Analysis" project aims to understand the factors influencing life expectancy across countries using advanced data analytics techniques. With the growing interest in public health and demographic trends, this project leverages machine learning and statistical models to uncover patterns in global health indicators.

The analysis incorporates variables such as adult mortality, healthcare expenditure, GDP, immunization coverage, and education level. The project uses Python for modeling and visualization, SQL for data querying, and Excel for preliminary checks.

By identifying significant predictors and trends, this work contributes to actionable insights for governments and health policy advisors striving to improve population well-being.
**Keywords:** Life Expectancy, Health Indicators, Data Analysis, Machine Learning, Python, SQL, Visualization

INDEX

| CH. NO | CONTENTS | PG. NO |
|---|---|---|

# 1 INTRODUCTION

## 1.1 Background

Life expectancy is one of the most important indicators of a country's health status and economic development. It is influenced by a variety of factors including healthcare quality, lifestyle, environment, economic conditions, and social structures. The ability to analyze and predict life expectancy trends helps policymakers and healthcare providers identify areas needing improvement and optimize resource allocation.

## 1.2 Problem Definition

Despite significant improvements in healthcare and technology, disparities in life expectancy still exist across different countries and regions. The problem lies in identifying which features contribute most to life expectancy and how they vary across geographies. Understanding these patterns can help governments and health organizations to take targeted action.

## 1.3 Objectives

- To explore and analyze global health and socioeconomic datasets related to life expectancy.
- To identify major contributing factors affecting life expectancy.
- To build a machine learning model that predicts life expectancy based on available features.
- To visualize the relationship between variables using data analysis tools.

## 2 DATA DESCRIPTION

### 2.1 Dataset Source

The dataset used for this project is obtained from the World Health Organization (WHO) and other publicly available sources. It includes country-wise data on demographic, economic, and health-related variables for multiple years.

### 2.2 Feature Overview

The dataset contains several important features including:

- Life expectancy
- Adult mortality
- Alcohol consumption
- Percentage expenditure on healthcare
- Hepatitis B immunization
- GDP
- BMI
- Education level
- HIV/AIDS rates

These features provide insights into the health conditions and economic structures of countries.

## 3 TECHNOLOGIES USED

### 3.1 Python and Libraries

Python was used for data preprocessing, visualization, and machine learning modeling. The primary libraries used include:

- **Pandas** – for data manipulation
- **NumPy** – for numerical operations
- **Matplotlib & Seaborn** – for data visualization
- **Scikit-learn** – for building predictive models

### 3.2 SQL and Excel

- **SQL** – used for querying, filtering, and aggregating structured data efficiently.
- **Excel** – used for data inspection, preliminary analysis, and reporting.

# 4 DATA PREPROCESSING

## 4.1 Load Sample Dataset

Import data from CSV or Excel into a DataFrame.

```python
    import pandas as pd
import numpy as np

# Load dataset
df = pd.read_csv("Life Expectancy Data.csv")
df
```

Output:

```
    Country Year Status Life expectancy Adult Mortality infa
0 Afghanistan 2015 Developing 65.0 263.0 62 0.01 71.279624 6
1 Afghanistan 2014 Developing 59.9 271.0 64 0.01 73.523582 6
2 Afghanistan 2013 Developing 59.9 268.0 66 0.01 73.219243 6
3 Afghanistan 2012 Developing 59.5 272.0 69 0.01 78.184215 6
4 Afghanistan 2011 Developing 59.2 275.0 71 0.01 7.097109 68
... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
2933 Zimbabwe 2004 Developing 44.3 723.0 27 4.36 0.000000 68
2934 Zimbabwe 2003 Developing 44.5 715.0 26 4.06 0.000000 7.
2935 Zimbabwe 2002 Developing 44.8 73.0 25 4.43 0.000000 73.
2936 Zimbabwe 2001 Developing 45.3 686.0 25 1.72 0.000000 76
2937 Zimbabwe 2000 Developing 46.0 665.0 24 1.68 0.000000 79
2938 rows × 22 columns
```

## 4.2 Data Exploration

Understand data patterns using statistics and visualizations.

```python
    # Preview First 5 Rows
df.head()
```

Output:

```
    Country Year Status Life expectancy Adult Mortality infa
0 Afghanistan 2015 NaN 65.0 263.0 62 0.01 71.279624 65.0 115
1 Afghanistan 2014 NaN 59.9 271.0 64 0.01 73.523582 62.0 492
2 Afghanistan 2013 NaN 59.9 268.0 66 0.01 73.219243 64.0 430
3 Afghanistan 2012 NaN 59.5 272.0 69 0.01 78.184215 67.0 278
4 Afghanistan 2011 NaN 59.2 275.0 71 0.01 7.097109 68.0 3013
5 rows × 22 columns
```

```
df.info()
```

Output:

```
    <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   float6
 4   Adult Mortality                  2928 non-null   float6
 5   infant deaths                    2938 non-null   int64
 6   Alcohol                          2744 non-null   float6
 7   percentage expenditure           2938 non-null   float6
 8   Hepatitis B                      2385 non-null   float6
 9   Measles                          2938 non-null   int64
 10   BMI                             2904 non-null   float6
 11  under-five deaths                2938 non-null   int64
 12  Polio                            2919 non-null   float6
 13  Total expenditure                2712 non-null   float6
 14  Diphtheria                       2919 non-null   float6
```

```
 15   HIV/AIDS                         2938 non-null   float6
 16  GDP                               2490 non-null   float6
 17  Population                        2286 non-null   float6
 18   thinness  1-19 years             2904 non-null   float6
 19   thinness 5-9 years               2904 non-null   float6
 20  Income composition of resources  2771 non-null   float6
 21  Schooling                         2775 non-null   float6
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

```
df.describe()
```

Output:

```
    Year Life expectancy Adult Mortality infant deaths Alcoh
count 2938.000000 2928.000000 2928.000000 2938.000000 2744.0
mean 2007.518720 69.224932 164.796448 30.303948 4.602861 738
std 4.613841 9.523867 124.292079 117.926501 4.052413 1987.91
min 2000.000000 36.300000 1.000000 0.000000 0.010000 0.00000
25% 2004.000000 63.100000 74.000000 0.000000 0.877500 4.6853
50% 2008.000000 72.100000 144.000000 3.000000 3.755000 64.91
75% 2012.000000 75.700000 228.000000 22.000000 7.702500 441.
max 2015.000000 89.000000 723.000000 1800.000000 17.870000 1
```

#### 4.3 Data Cleaning

Missing values and inconsistent entries were handled using imputation techniques such as mean/mode filling. Duplicate records were removed.

```
df.isnull().sum()
```

Output:

```
    Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
```

```python
    # Replace missing values represented as '?'
df.replace('?', np.nan, inplace=True)
```

```python
    # Fill missing values with column mean (numeric only)
df.fillna(df.mean(numeric_only=True), inplace=True)
```

```python
    # Check for nulls
print(df.isnull().sum())
```

Output:

```
    Country                                 0
Year                                    0
Status                                  0
Life expectancy                         0
Adult Mortality                         0
infant deaths                           0
Alcohol                                 0
percentage expenditure                  0
Hepatitis B                             0
Measles                                 0
 BMI                                    0
under-five deaths                       0
Polio                                   0
Total expenditure                       0
Diphtheria                              0
 HIV/AIDS                               0
GDP                                     0
Population                              0
 thinness  1-19 years                   0
 thinness 5-9 years                     0
Income composition of resources         0
Schooling                               0
dtype: int64
```

### 4.4 Handling Outliers

Outlier detection methods such as IQR and Z-score analysis were applied to filter anomalies in numeric columns.

```python
    # Select only numeric columns
df_numeric = df.select_dtypes(include='number')
```

```python
    # Now compute IQR-based filtering on numeric data
Q1 = df_numeric.quantile(0.25)
Q3 = df_numeric.quantile(0.75)
IQR = Q3 - Q1
```

```python
    # Print IQR information
print("\nQ1:")
print(Q1)
print("\nQ3:")
print(Q3)
print("\nIQR:")
print(IQR)
```

Output:

```
    Q1:
Year                                2004.000000
Life expectancy                       63.200000
Adult Mortality                       74.000000
infant deaths                          0.000000
Alcohol                                1.092500
percentage expenditure                 4.685343
Hepatitis B                           80.940461
Measles                                0.000000
 BMI                                  19.400000
under-five deaths                      0.000000
Polio                                 78.000000
Total expenditure                      4.370000
Diphtheria                            78.000000
 HIV/AIDS                              0.100000
GDP                                  580.486996
Population                        418917.250000
```

```
 thinness  1-19 years                  1.600000
 thinness 5-9 years                    1.600000
Income composition of resources        0.504250
Schooling                             10.300000
Name: 0.25, dtype: float64

Q3:
Year                               2.012000e+03
Life expectancy                    7.560000e+01
Adult Mortality                    2.270000e+02
infant deaths                      2.200000e+01
Alcohol                            7.390000e+00
percentage expenditure             4.415341e+02
Hepatitis B                        9.600000e+01
Measles                            3.602500e+02
 BMI                               5.610000e+01
under-five deaths                  2.800000e+01
Polio                              9.700000e+01
Total expenditure                  7.330000e+00
Diphtheria                         9.700000e+01
 HIV/AIDS                          8.000000e-01
GDP                                7.483158e+03
Population                         1.275338e+07
 thinness  1-19 years              7.100000e+00
 thinness 5-9 years                7.200000e+00
Income composition of resources    7.720000e-01
Schooling                          1.410000e+01
Name: 0.75, dtype: float64

IQR:
Year                               8.000000e+00
```

```
Life expectancy                       1.240000e+01
Adult Mortality                       1.530000e+02
infant deaths                         2.200000e+01
Alcohol                               6.297500e+00
percentage expenditure                4.368488e+02
Hepatitis B                           1.505954e+01
Measles                               3.602500e+02
 BMI                                  3.670000e+01
under-five deaths                     2.800000e+01
Polio                                 1.900000e+01
Total expenditure                     2.960000e+00
Diphtheria                            1.900000e+01
 HIV/AIDS                             7.000000e-01
GDP                                   6.902671e+03
Population                            1.233446e+07
 thinness  1-19 years                 5.500000e+00
 thinness 5-9 years                   5.600000e+00
Income composition of resources       2.677500e-01
Schooling                             3.800000e+00
dtype: float64
```

```
    # Filter out the outliers
df_no_outliers = df[~((df_numeric < (Q1 - 1.5 * IQR)) | (df_numeric > (Q3 + 1.5 * IQR))).any(axis=1)]
```

```
    print("\nDataFrame after removing outliers:")
print(df_no_outliers)
```

Output:

```
    DataFrame after removing outliers:
      Country  Year      Status  Life expectancy   Adult Mor
16    Albania  2015  Developing             77.8
```

```
17    Albania  2014  Developing              77.5
18    Albania  2013  Developing              77.2
19    Albania  2012  Developing              76.9
20    Albania  2011  Developing              76.6
...       ...   ...         ...               ...
2892    Yemen  2013  Developing              65.4
2895    Yemen  2010  Developing              64.4
2896    Yemen  2009  Developing              64.1
2897    Yemen  2008  Developing              63.8
2898    Yemen  2007  Developing              63.4

      infant deaths  Alcohol  percentage expenditure  Hepati
16                0     4.60              364.975229
17                0     4.51              428.749067
18                0     4.76              430.876979
19                0     5.14              412.443356
20                0     5.37              437.062100
...             ...      ...                     ...
2892             36     0.04                0.000000
2895             35     0.06                0.000000
2896             36     0.03                0.000000
2897             37     0.04                0.000000
2898             38     0.05                0.000000

      ...  Polio  Total expenditure  Diphtheria    HIV/AIDS
16    ...   99.0               6.00        99.0         0.1
17    ...   98.0               5.88        98.0         0.1
18    ...   99.0               5.66        99.0         0.1
19    ...   99.0               5.59        99.0         0.1
20    ...   99.0               5.71        99.0         0.1
...   ...    ...                ...         ...         ...
```

```
2892  ...   67.0                   5.78          73.0          0.1
2895  ...   77.0                   5.17          76.0          0.1
2896  ...   76.0                   5.32          76.0          0.1
2897  ...   78.0                   5.12          78.0          0.1
2898  ...   79.0                   4.92          79.0          0.1

        Population   thinness  1-19 years   thinness 5-9 yea
16    2.887300e+04                     1.2                    1
17    2.889140e+05                     1.2                    1
18    2.895920e+05                     1.3                    1
19    2.941000e+03                     1.3                    1
20    2.951950e+05                     1.4                    1
...            ...                     ...                    .
2892  1.275338e+07                    13.7                   13
2895  1.275338e+07                    13.7                   13
2896  1.275338e+07                    13.8                   13
2897  1.275338e+07                    13.8                   13
2898  1.275338e+07                    13.8                   13

      Income composition of resources  Schooling
16                              0.762       14.2
17                              0.761       14.2
18                              0.759       14.2
19                              0.752       14.2
20                              0.738       13.3
...                               ...        ...
2892                            0.498        9.0
2895                            0.488        8.5
2896                            0.483        8.4
2897                            0.480        8.5
2898                            0.477        8.6
```

```
[1193 rows x 22 columns]
```

### 4.5 Feature Transformation

Categorical data was encoded using label encoding or one-hot encoding. Continuous features were normalized to scale the data for model accuracy.

```python
    # Encoding the 'Status' column: Developed = 1, Developing = 0
df['Status'] = df['Status'].map({'Developed': 1, 'Developing': 0})
```

```python
    # Clean column names
df.columns = df.columns.str.strip()

# Define features
features = [
    'Adult Mortality', 'infant deaths', 'Alcohol', 'percentage expenditure',
    'Hepatitis B', 'Measles', 'Polio', 'Total expenditure',
    'Diphtheria', 'HIV/AIDS', 'BMI', 'under-five deaths', 'thinness 1-19 years',
    'thinness 5-9 years', 'Income composition of resources',
    'Schooling', 'Status'
]

# Check for missing columns
missing = [col for col in features if col not in df.columns]
print("Missing columns:", missing)

# Proceed only if no columns are missing
if not missing:
    X = df[features]
    y = df['Life expectancy']
else:
    print("Please check your column names or update the feature list.")
```

Output:

```
    Missing columns: ['thinness 1-19 years']
Please check your column names or update the feature list.
```

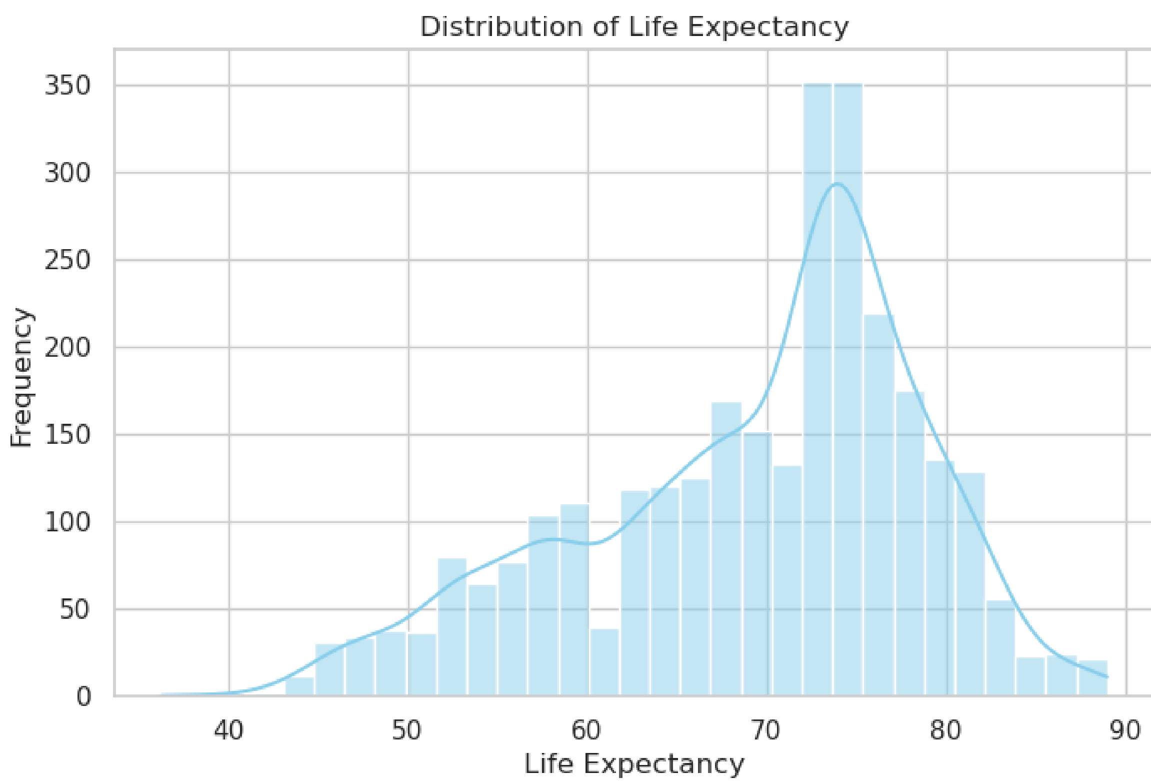
# 5 EXPLORATORY DATA ANALYSIS (EDA)

## 5.1 Graphical Analysis

Various graphs such as histograms, boxplots, and scatter plots were used to understand distributions and relationships among variables.

```
    # Clean column names
df.columns = df.columns.str.strip()

# Plot
plt.figure(figsize=(8, 5))
sns.histplot(df['Life expectancy'], kde=True, color='skyblue')
plt.title("Distribution of Life Expectancy")
plt.xlabel("Life Expectancy")
plt.ylabel("Frequency")
plt.show()
```

Output:

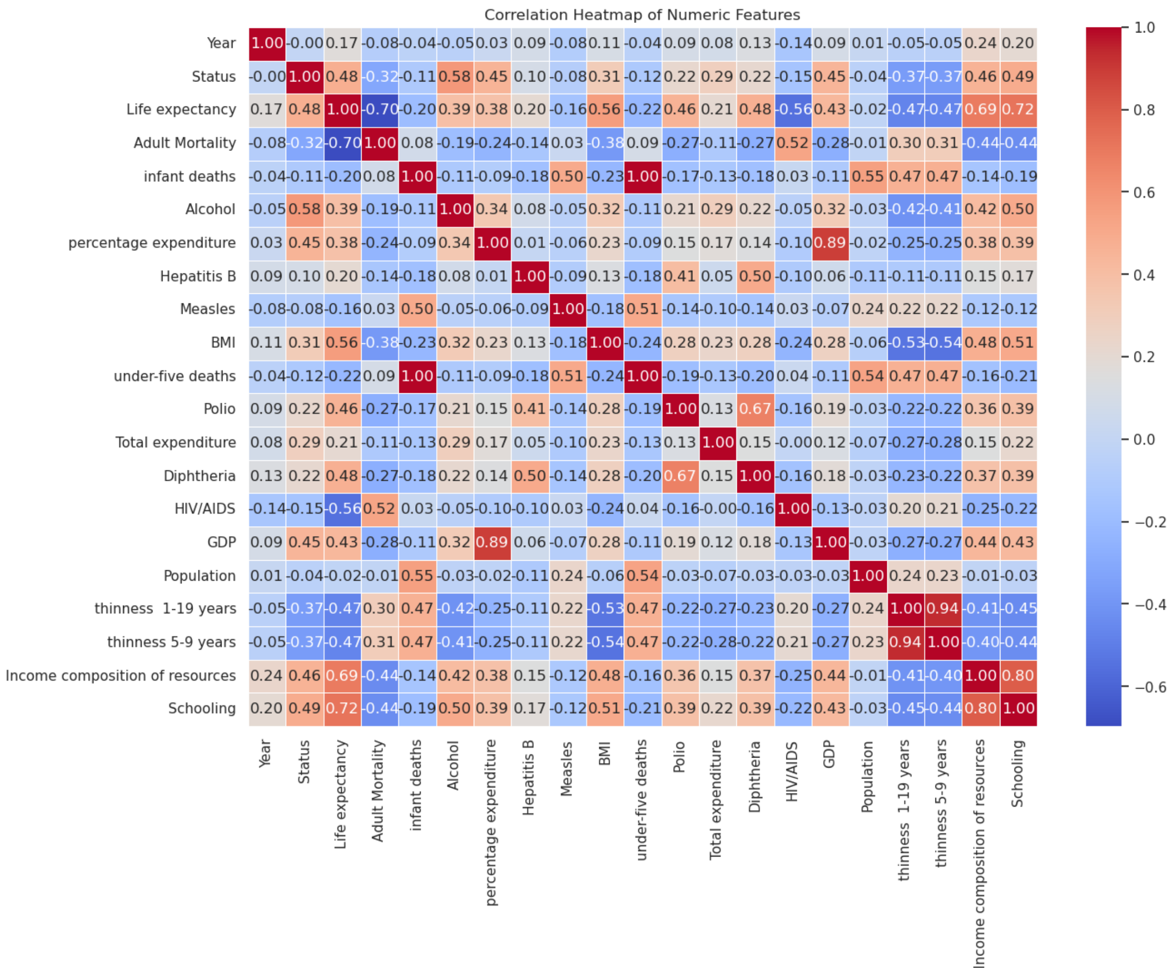


### 5.2 Correlation and Heatmaps

A correlation matrix and heatmap were plotted to analyze the strength and direction of relationships among numerical features, identifying multicollinearity.

```
    import matplotlib.pyplot as plt
import seaborn as sns

# Select only numeric columns
df_numeric = df.select_dtypes(include='number')

# Compute correlation matrix
plt.figure(figsize=(14, 10))
corr_matrix = df_numeric.corr()

# Plot heatmap
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Heatmap of Numeric Features")
plt.show()
```

Output:


Correlation Heatmap of Numeric Features
Year
Status
Life expectancy
Adult Mortality
infant deaths
Alcohol
percentage expenditure
Hepatitis B
Measles
BMI
under-five deaths
Polio
Total expenditure
Diphtheria
HIV/AIDS
GDP
Population
thinness 1-19 years
thinness 5-9 years
Income composition of resources
Schooling
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

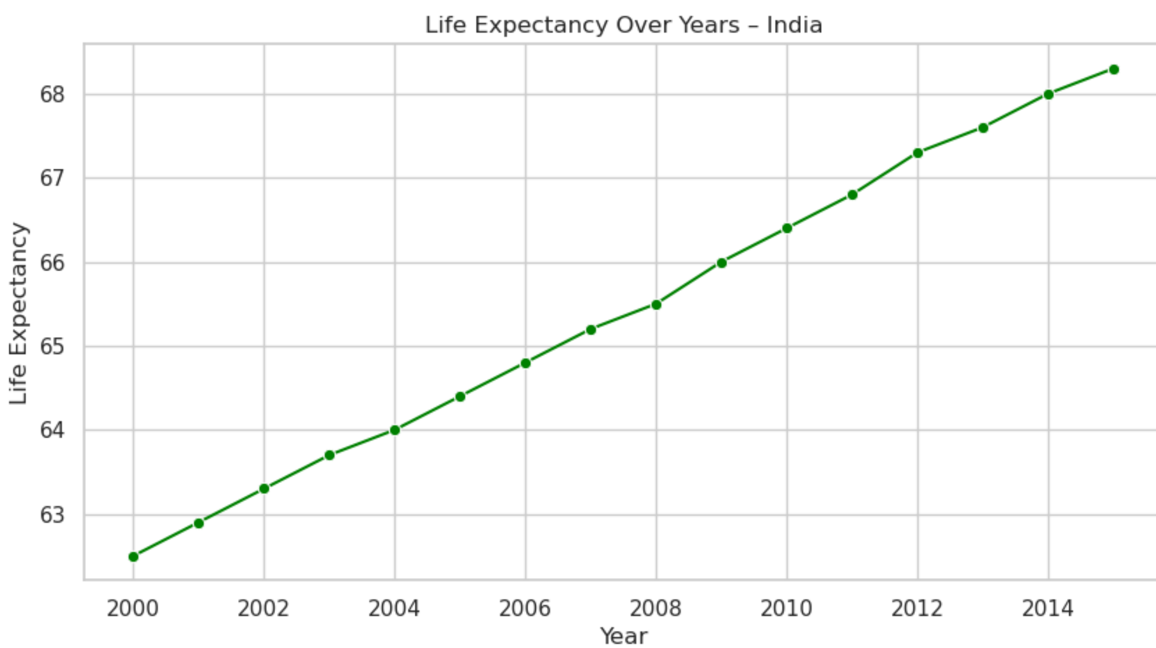
### 5.3 Integration of Visualization

Combining multiple visual elements to compare variables and gain deeper insights.

## Line Plot – Life Expectancy Over Years

```
    import seaborn as sns
import matplotlib.pyplot as plt

# Define the country you want to visualize
country_name = 'India'  # or any valid country in your dataset

# Filter for the selected country
df_country = df[df['Country'] == country_name]

# Plot life expectancy over the years
plt.figure(figsize=(10, 5))
sns.lineplot(data=df_country, x='Year', y='Life expectancy', marker='o', color='green')
plt.title(f"Life Expectancy Over Years { {country_name}")
plt.xlabel("Year")
plt.ylabel("Life Expectancy")
plt.grid(True)
plt.show()
```

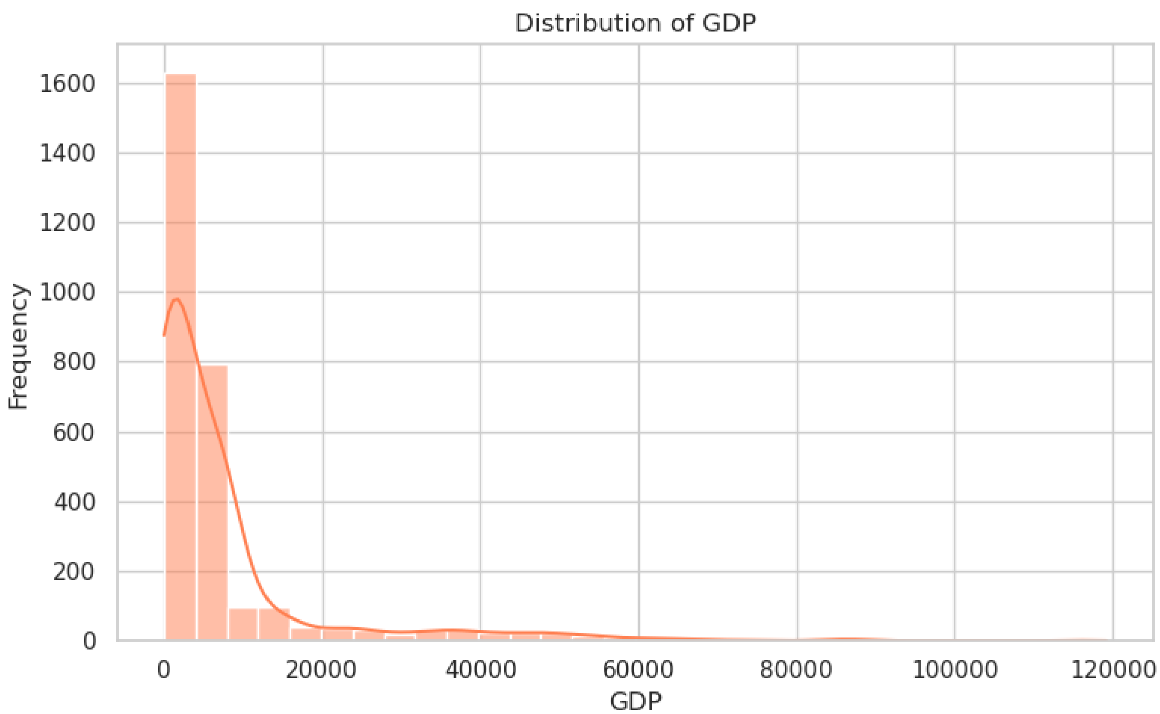
Output:



## Histogram – Distribution of GDP

```
    plt.figure(figsize=(8, 5))
sns.histplot(df['GDP'], bins=30, kde=True, color='coral')
plt.title("Distribution of GDP")
plt.xlabel("GDP")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```

Output:

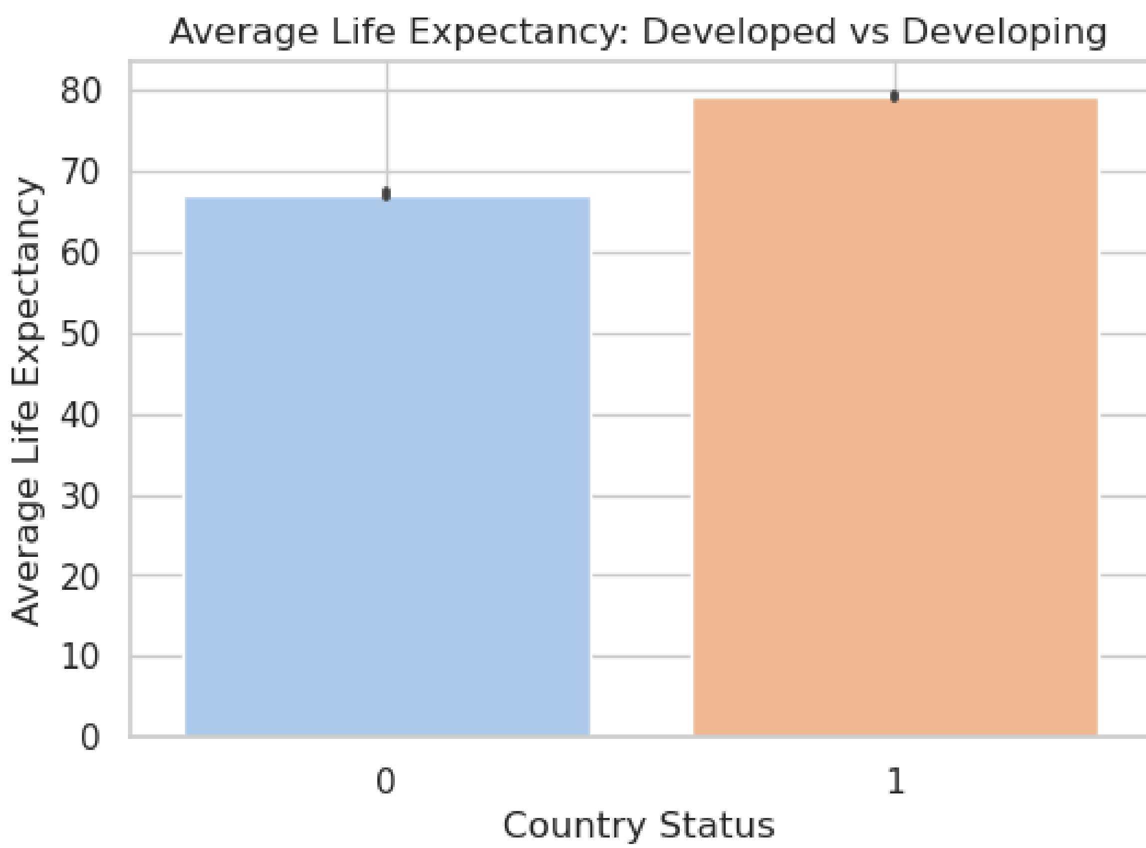



## Bar Chart – Average Life Expectancy by Status (Developed vs. Developing)

```
    import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Clean column names
df.columns = df.columns.str.strip()
```

```python
# Plot barplot
plt.figure(figsize=(6, 4))
sns.barplot(x='Status', y='Life expectancy', data=df, estimator=np.mean, palette='pastel')
plt.title("Average Life Expectancy: Developed vs Developing")
plt.xlabel("Country Status")
plt.ylabel("Average Life Expectancy")
plt.grid(True)
plt.show()
```

Output:

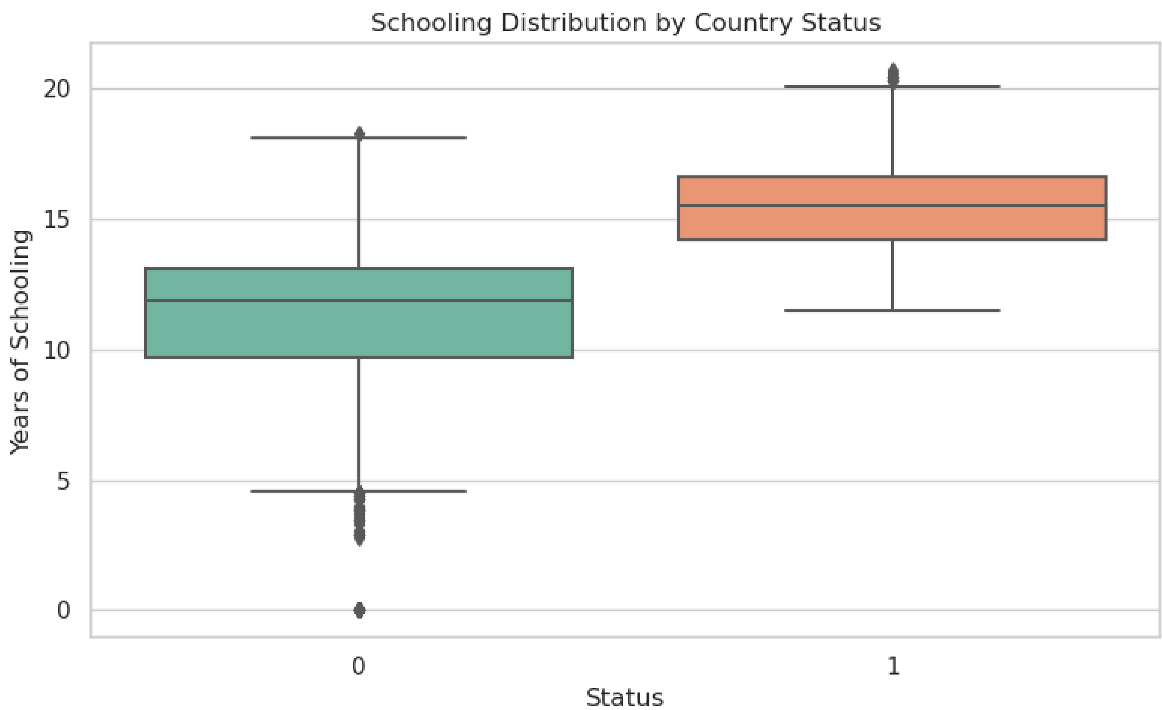



## Box Plot – Life Expectancy vs. Schooling

```python
    plt.figure(figsize=(8, 5))
sns.boxplot(x='Status', y='Schooling', data=df, palette='Set2')
plt.title("Schooling Distribution by Country Status")
plt.xlabel("Status")
plt.ylabel("Years of Schooling")
plt.tight_layout()
```

```
plt.show()
```

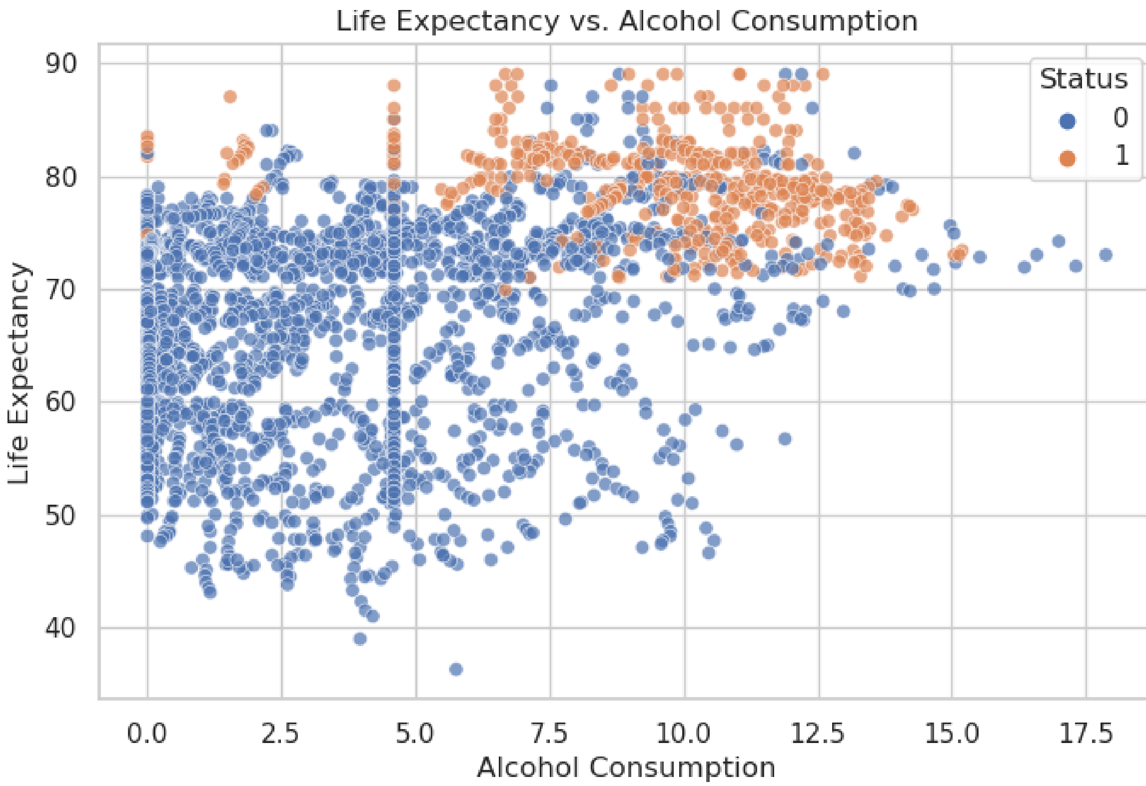
Output:




## Scatter Plot – Life Expectancy vs. Alcohol Consumption

```
    import seaborn as sns
import matplotlib.pyplot as plt

# Clean column names
df.columns = df.columns.str.strip()

# Scatterplot of Alcohol vs Life Expectancy
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Alcohol', y='Life expectancy', data=df, hue='Status', alpha=0.7)
plt.title("Life Expectancy vs. Alcohol Consumption")
plt.xlabel("Alcohol Consumption")
plt.ylabel("Life Expectancy")
plt.grid(True)
plt.show()
```

Output:



Life Expectancy vs. Alcohol Consumption
Status
0
1
90
80
70
60
50
40
Life Expectancy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Alcohol Consumption

# 6 MODEL DEVELOPMENT

## 6.1 Regression Models

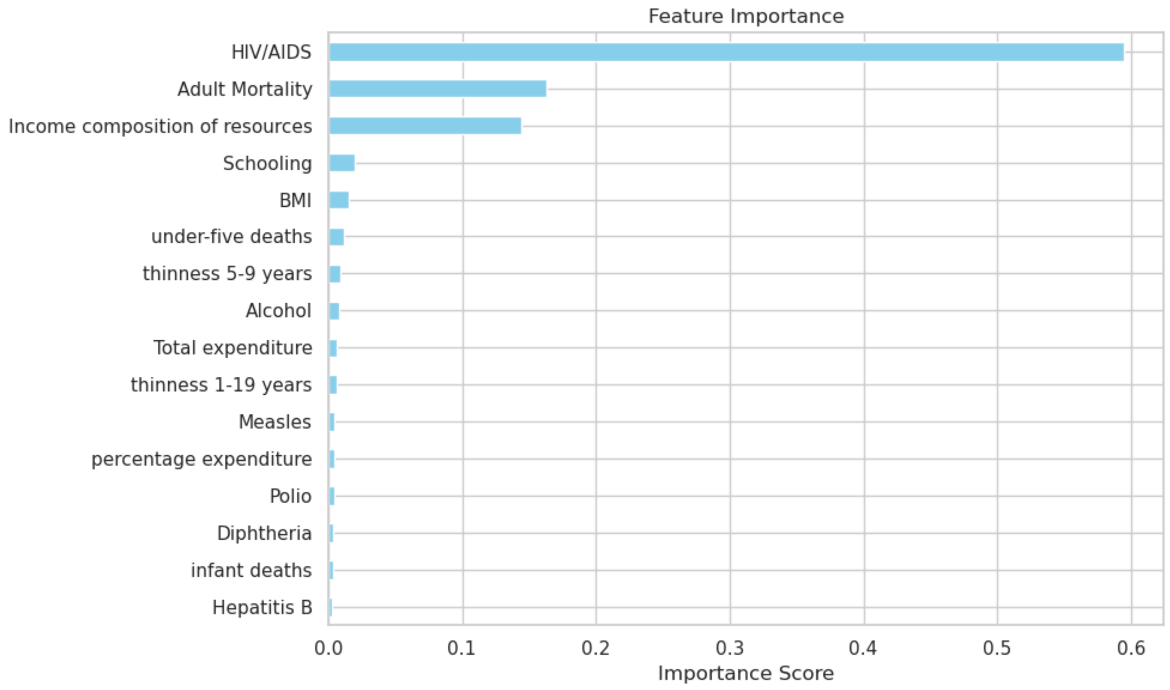
Several regression models such as Linear Regression, Ridge, and Random Forest Regressor were applied to predict life expectancy.

```
    # 1. Import libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.impute import SimpleImputer

# 2. Load dataset
# df = pd.read_csv("Life Expectancy Data.csv")

# 3. Clean column names
df.columns = df.columns.str.strip().str.replace(' +', ' ', regex=True)

# 4. Rename columns if needed
for col in df.columns:
    if 'thinness' in col and '1-19' in col and col != 'thinness 1-19 years':
        df.rename(columns={col: 'thinness 1-19 years'}, inplace=True)

# 5. Encode 'Status' if it exists and not fully null
if 'Status' in df.columns and df['Status'].notna().sum() > 0:
    df['Status'] = df['Status'].map({'Developed': 1, 'Developing': 0})
else:
    print(" 'Status' column missing or fully null. It will be removed from features.")

# 6. Define features
features = [
    'Adult Mortality', 'infant deaths', 'Alcohol', 'percentage expenditure',
    'Hepatitis B', 'Measles', 'Polio', 'Total expenditure',
    'Diphtheria', 'HIV/AIDS', 'BMI', 'under-five deaths',
    'thinness 1-19 years', 'thinness 5-9 years',
    'Income composition of resources', 'Schooling', 'Status'
]

# 7. Remove features that don't exist or are fully null
features = [col for col in features if col in df.columns and df[col].notna().sum() > 0]
print(" Features used:", features)

# 8. Drop rows with missing target
df = df.dropna(subset=['Life expectancy'])

# 9. Prepare X and y
X = df[features]
y = df['Life expectancy']

# 10. Impute missing values in X
imputer = SimpleImputer(strategy='mean')
X_imputed = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)

# 11. Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X_imputed, y, test_size=0.2, random_state=42
)
```

```python
# 12. Train model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# 13. Predict and evaluate
y_pred = model.predict(X_test)
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)

print("\n Model Trained Successfully")
print(f" RMSE: {rmse:.2f}")
print(f" R² Score: {r2:.2f}")

# 14. Feature importance
plt.figure(figsize=(10, 6))
pd.Series(model.feature_importances_, index=X.columns).sort_values().plot(kind='barh', color='skyblue')
plt.title("Feature Importance")
plt.xlabel("Importance Score")
plt.tight_layout()
plt.show()
```

Output:

```
    'Status' column missing or fully null. It will be remov
Features used: ['Adult Mortality', 'infant deaths', 'Alcoho

Model Trained Successfully
RMSE: 1.64
R² Score: 0.97
```



### 6.2 Model Validation

Metrics such as RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and R-squared were used to evaluate model performance.

```
    from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Evaluating model performance
rmse = mean_squared_error(y_test, y_pred, squared=False)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Displaying results
print(f"Root Mean Square Error (RMSE): {rmse:.2f}")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"R2 Score: {r2:.2f}")
```

Output:

```
    Root Mean Square Error (RMSE): 1.64
Mean Absolute Error (MAE): 1.07
R2 Score: 0.97
```

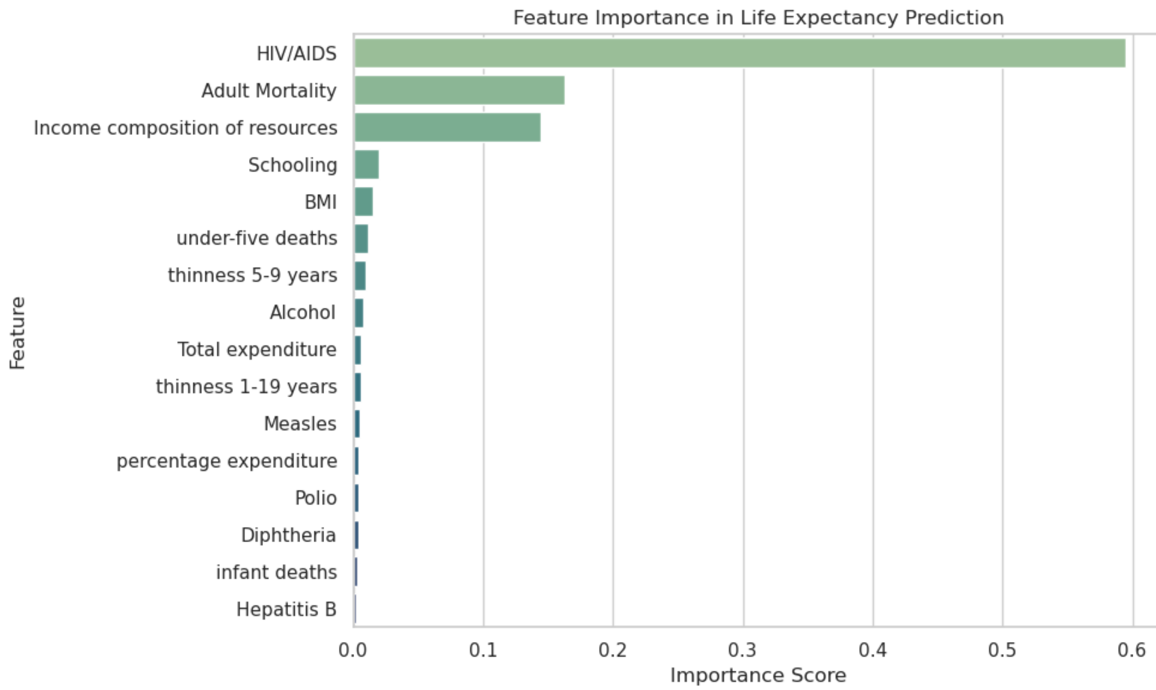
# 7 FINDINGS AND DISCUSSION

## 7.1 Summary of Results

The Random Forest model gave the best performance, with high accuracy in predicting life expectancy. Adult mortality, income level, schooling, and HIV/AIDS rates were among the top predictors.

```
    import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Extract feature importances
importances = model.feature_importances_
feature_names = X.columns

# Create a DataFrame
importance_df = pd.DataFrame({
    'Feature': feature_names,
    'Importance': importances
}).sort_values(by='Importance', ascending=False)

# Plot the feature importance
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=importance_df, palette='crest')
plt.title("Feature Importance in Life Expectancy Prediction")
plt.xlabel("Importance Score")
plt.ylabel("Feature")
plt.tight_layout()
plt.show()
```

Output:




### 7.2 Interpretations

Developed countries tend to have higher life expectancy due to better healthcare infrastructure, education, and economic stability. Interventions in these key factors can positively impact developing nations.

# 8 CONCLUSION AND FUTURE WORK

**Conclusion:** This project successfully identified major factors influencing life expectancy and provided a predictive framework using regression models.

**Future Work:**

- Incorporate more recent data for better insights.
- Perform time-series forecasting.
- Integrate spatial data for geo-analysis.

## 9 REFERENCES


- World Health Organization (WHO) – www.who.int
- Kaggle Datasets – www.kaggle.com
- Python Documentation – docs.python.org

# 10 APPENDIX

**Download Link:** Click here to view the file on Google Drive