Life Expectancy Prediction

Using Multiple Linear Regression

Yasha Jain

199302075

The problem

Context

Everything has an expiration date, humans are no exception either. Life expectancy measure is one of the most common ways to find out how long a person if going to live. This usually helps people in making long term decision about this life.

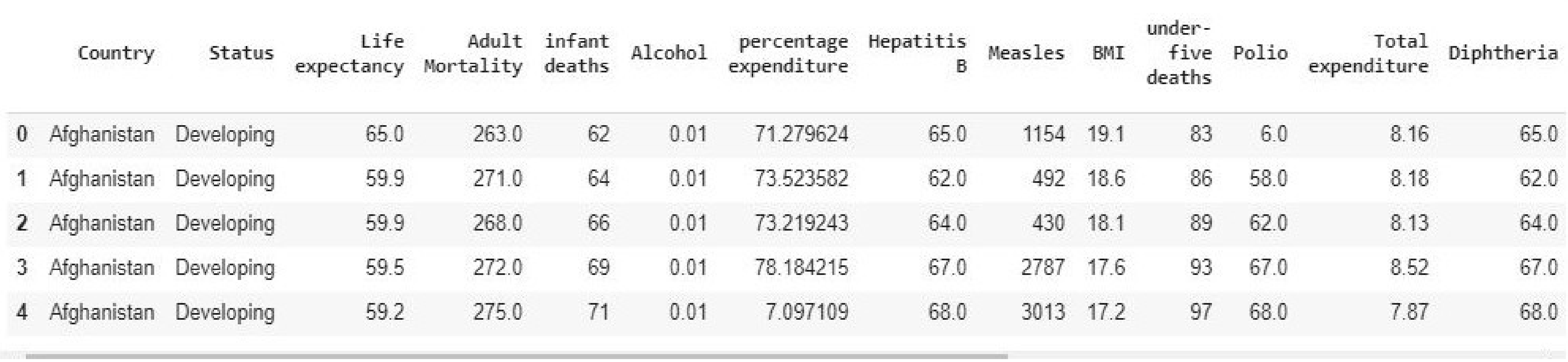
Problem statement

To **predict life expectancy using Multiple Linear Regression** which uses features like but not limited to Alcohol Consumption, various medical issues like Polio, Measles, HIV and also some factors of the country of a subject and its economic status.

Data utilized

Source: https://www.kaggle.com/kumarajarshi/life-expectancy-who/data
This data was provided by WHO.

[21 columns and 2937 rows]



	Country	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria
0	Afghanistan	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	83	6.0	8.16	65.0
1	Afghanistan	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	86	58.0	8.18	62.0
2	Afghanistan	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	89	62.0	8.13	64.0
3	Afghanistan	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	93	67.0	8.52	67.0
4	Afghanistan	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	97	68.0	7.87	68.0

Country

Status

Life Expectancy

Polio

Adult Mortality

Alcohol

percentage expenditure

Total expenditure

Hepatitis B

Measles

BMI

under-five deaths

Diphtheria

HIV/AIDS

GDP

Population

thinness 1-19 years

thinness 5-9 years

Income composition of resources

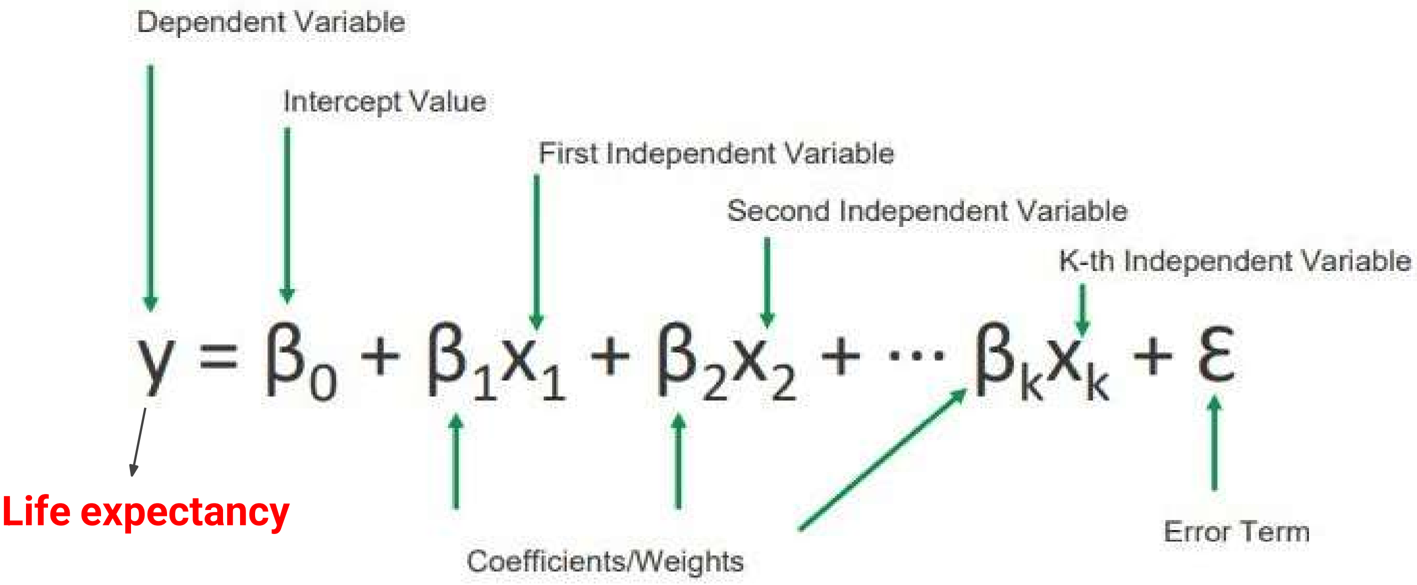
Schooling

Approach

Linear Regression

Feature Selection

Multiple Linear Regression



$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_k x_k + \varepsilon$$

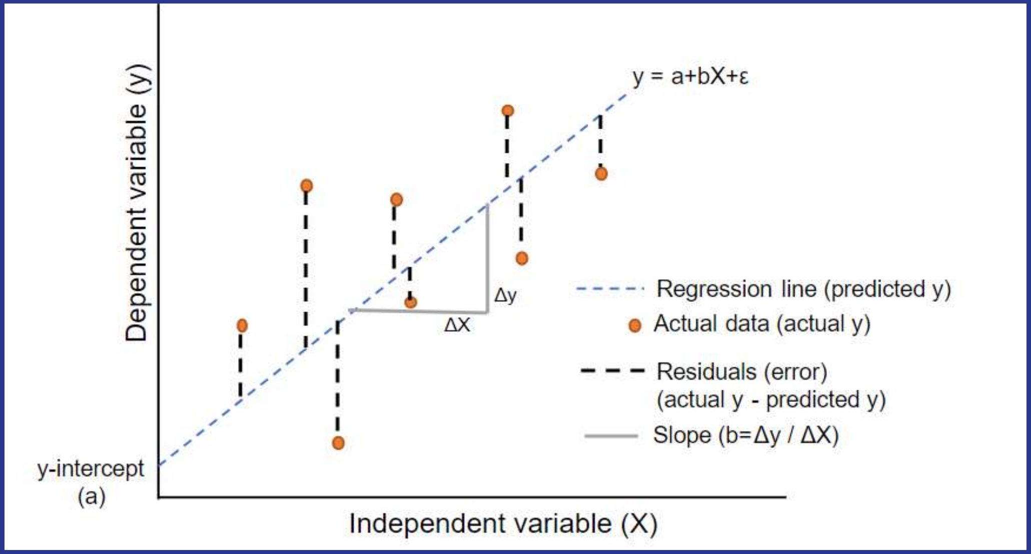
Dependent Variable
Intercept Value
First Independent Variable
Second Independent Variable
K-th Independent Variable
Coefficients/Weights
Error Term

Life expectancy

Linear Regression Model

Linear Regression

Linear Regression is a regression algorithm with a linear approach. It’s a supervised regression algorithm where we try to predict a continuous value of a given data point by generalizing the data we have in hand. The linear part indicates the linear approach for the generalization of data.



The idea is to predict the dependent variable (Y) using a given independent variable (X). This can be accomplished by fitting a best fit line in the data. A line providing the least sum of residual error is the best fit line or regression line.

Libraries used

1) **NumPy** :-NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is particularly useful for linear algebra, Fourier transform, and random number capabilities.

2) **Scikit-learn** :- Scikit-learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy.

3) **Pandas** :- Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

4) **Matplotlib** :- Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc.

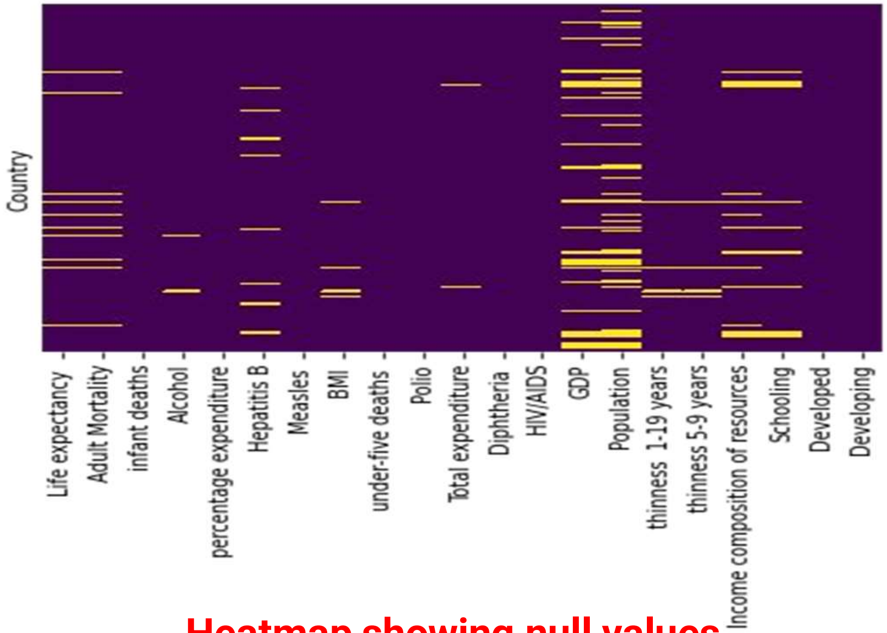
5) **Seaborn** :-- Seaborn is a Python data visualization library based on the Matplotlib library. It provides a high-level interface for drawing attractive and informative statistical graphs.

6) **SciPy** :- SciPy is a free and open-source Python library used for scientific computing and technical computing. It is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

Steps

1. Data Cleaning/Preprocessing

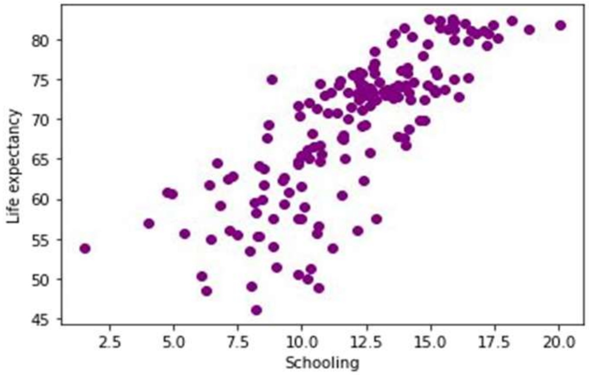
1. Year column is dropped as it has no significance in the prediction process
2. Non-numeric columns:
   “Status” -One hot encoded
   “Country” -Data grouped by country
3. All NaN cells/values are replaced by their respective column mean values

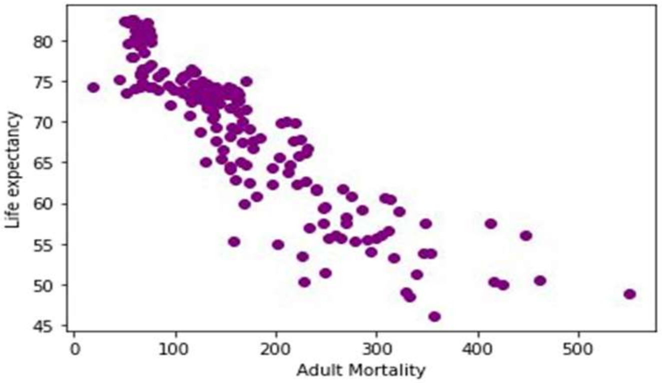


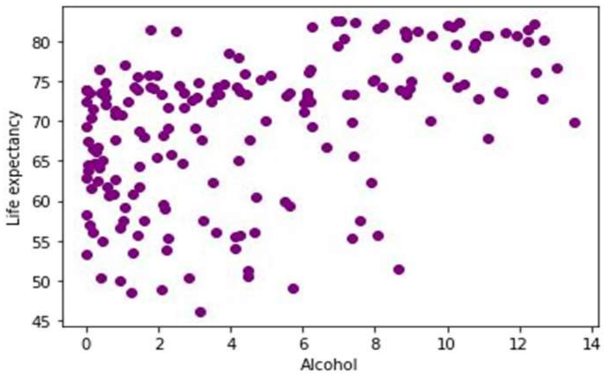
Heatmap showing null values

2. Exploratory Data Analysis

1. Positive correlation between The Percentage of Healthcare Expenditure, Schooling, GDP and BMI and Life Expectancy
2. Negative correlation between Adult Mortality, AIDS and Life Expectancy
3. No correlation between Alcohol, under 5 years – old deaths and Life Expectancy.







3. Linear Regression Assumptions

1. Data must be quantitative
2. No multicollinearity between the features. (Checked with a correlation matrix)

Results of Initial Linear Regression

After processing and cleaning the data, performing exploratory data analysis and taking assumptions we implement Linear Regression on the data set to find out the results. We divide our data in 70:30 ratio to train the model and then test the model.

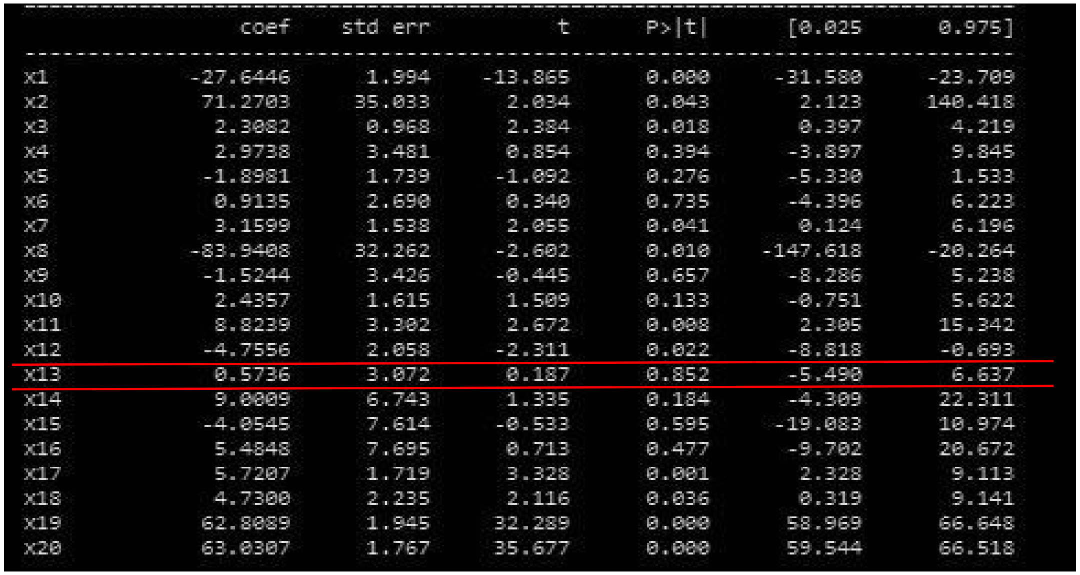
Following is the result of our first run of Linear regression:

Mean Squared Error	7.17
Mean Absolute Error	2.05
Root Mean Squared Error	2.68
R_Square score	0.91

As you can see, we have very high value of errors along with our R_Sqaure value as 0.91. This means if we use this model to predict the life expectancy only 91% of the variations, it is good but we can increase it by dropping certain features, which have lesser impact on Life Expectancy as compared to other features.

FEATURE SELECTION WITH OLS MODEL

1. We find the p value using the OLS regression model to find the significant parameters.
2. We drop the feature with the greatest p value on each iterative fit.
3. P>|t| is the p-value for this hypothesis test. A low p-value means, that you can reject the null-hypothesis and accept the alternative hypothesis (coef!=0). A p-value of less than 0.05 is considered to be statistically significant. (95% CI)



```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1           -27.6446      1.994    -13.865      0.000     -31.580     -23.709
x2            71.2703     35.033      2.034      0.043       2.123     140.418
x3             2.3082      0.968      2.384      0.018       0.397       4.219
x4             2.9738      3.481      0.854      0.394      -3.897       9.845
x5            -1.8981      1.739     -1.092      0.276      -5.330       1.533
x6             0.9135      2.690      0.340      0.735      -4.396       6.223
x7             3.1599      1.538      2.055      0.041       0.124       6.196
x8           -83.9408     32.262     -2.602      0.010    -147.618     -20.264
x9            -1.5244      3.426     -0.445      0.657      -8.286       5.238
x10            2.4357      1.615      1.509      0.133      -0.751       5.622
x11            8.8239      3.302      2.672      0.008       2.305      15.342
x12           -4.7556      2.058     -2.311      0.022      -8.818      -0.693
x13            0.5736      3.072      0.187      0.852      -5.490       6.637
x14            9.0009      6.743      1.335      0.184      -4.309      22.311
x15           -4.0545      7.614     -0.533      0.595     -19.083      10.974
x16            5.4848      7.695      0.713      0.477      -9.702      20.672
x17            5.7207      1.719      3.328      0.001       2.328       9.113
x18            4.7300      2.235      2.116      0.036       0.319       9.141
x19           62.8089      1.945     32.289      0.000      58.969      66.648
x20           63.0307      1.767     35.677      0.000      59.544      66.518
```

- Here we observe feature X13 has the maximum p>|t| value, which indicates this feature can be discarded.
- We discard X13 and reiterate OLS regression to find the next insignificant feature, and we repeat this process 4-5 time.

Results of Multiple Linear Regression

After dropping insignificant features from consideration we re run the Linear Regression on the data set to find out the results. Performing the linear regression again after making improvements, is called Multiple Linear Regression.
This time again we divide our data in 70:30 ratio to train the model and then test the model.

Following is the result of our Multiple Linear regression:

Mean Squared Error	5.59
Mean Absolute Error	1.92
Root Mean Squared Error	2.37
R_Square score	0.93

We observe a significant drop in the error values and improvement in R_Sqaure value which is now 0.93. This means if we use this model to predict the life expectancy then 93% of the variations can be explained, which is a great result.

Conclusion

Final accuracy -93%

Final r^2 -0.93 Final

RMSE -2.37

After taking 15 features and performing Multiple Linear Regression on the relevant data, our model can now predict the life expectancy of an individual with 93% explanation of variations about the mean.

THANK YOU