***PROJECT REPORT:***

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING-SB44106

SUMMER INTERNSHIP

AT

SMARTBRIDGE

BY: **SHUBHANKAR JAIN**

EMAIL: **shubrock755@gmail.com**

|  |  |  |
| --- | --- | --- |
| Project ID | : | SPS\_PRO\_215 |

Contents

1.

INTRODUCTION

1.1 Overview

1.2 Purpose

2.

LITERATURE SURVEY

2.1 Existing problem

2.2 Proposed solution

3.

THEORITICAL ANALYSIS

3.1 Block diagram

3.2 Hardware / Software designing

4.

EXPERIMENTAL INVESTIGATIONS

5.

FLOWCHART

6.

RESULT

7.

ADVANTAGES & DISADVANTAGES

8.

APPLICATIONS

9.

CONCLUSION

10.

FUTURE SCOPE

11.

BIBILOGRAPHY

APPENDIX

A. Source code

**1.Intoduction**

**1.1 Overview**

Life expectancy is a statistical measure of the average time a human being is expected to live, Life expectancy depends on various factors: Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth and other demographic factors. This project provides a way to predict average life expectancy of people living in a country when various factors such as year, GDP, education, alcohol intake of people in the country, expenditure on healthcare system and some specific disease related deaths that happened in the country are given. The user shall be able to interact with the program using the UI built with Node-RED.

**1.2 Purpose**

Prognostication of life expectancy is difficult for humans. Our research shows that machine learning and natural language processing techniques offer a feasible and promising approach to predicting life expectancy. The research has potential for real-life applications, such as supporting timely recognition of the right moment to start Advance Care Planning.

The purpose of this model is to estimate the life expectancy of the world population so that the government has a benchmark in determining policies to further improve the health of the people in their respective countries. The estimation stated in this model will use the Machine Learning method like Multiple Linear Regression. The data used in this model is the number of world population. Data sources come from the Global Health Observatory (GHO) under World Health Organization (WHO). The results of this study are expected to be a reference for the governments of each country to pay more attention to the level of health and welfare of its population so that the life expectancy of the population will be higher.

**2.Literature Survey**

**2.1 Existing problem**

Health forecasts and alternative future scenarios can serve as vital inputs into long-term planning and investments in health, particularly in terms of framing different choices, their potential effects, and the relative certainty associated with each option. Past work to generate health-focused forecasts includes that from the UN Population Division, and the Austrian Wittgenstein Center, which produces life expectancy forecasts with different scenarios to the end of the 21st century. Longer-range forecasts have also been developed to assess the potential effects of climate change on mortality. Furthermore, various national agencies produce country-level mortality forecasts, and forecasts for individual causes of death have been produced periodically as well. Comprehensive forecasts of cause-specific and all-cause mortality were developed as part of the Global Burden of Disease Study 1990 (GBD 1990); those methods were then applied for the period from 2002–30. The primary purpose of these past modelling efforts was to generate reference or baseline forecasts of what was likely to occur on the basis of past trends; however, few—if any—offered insights into a range of future scenarios while accounting for independent drivers of potential health changes.

**2.2 Proposed Solution**

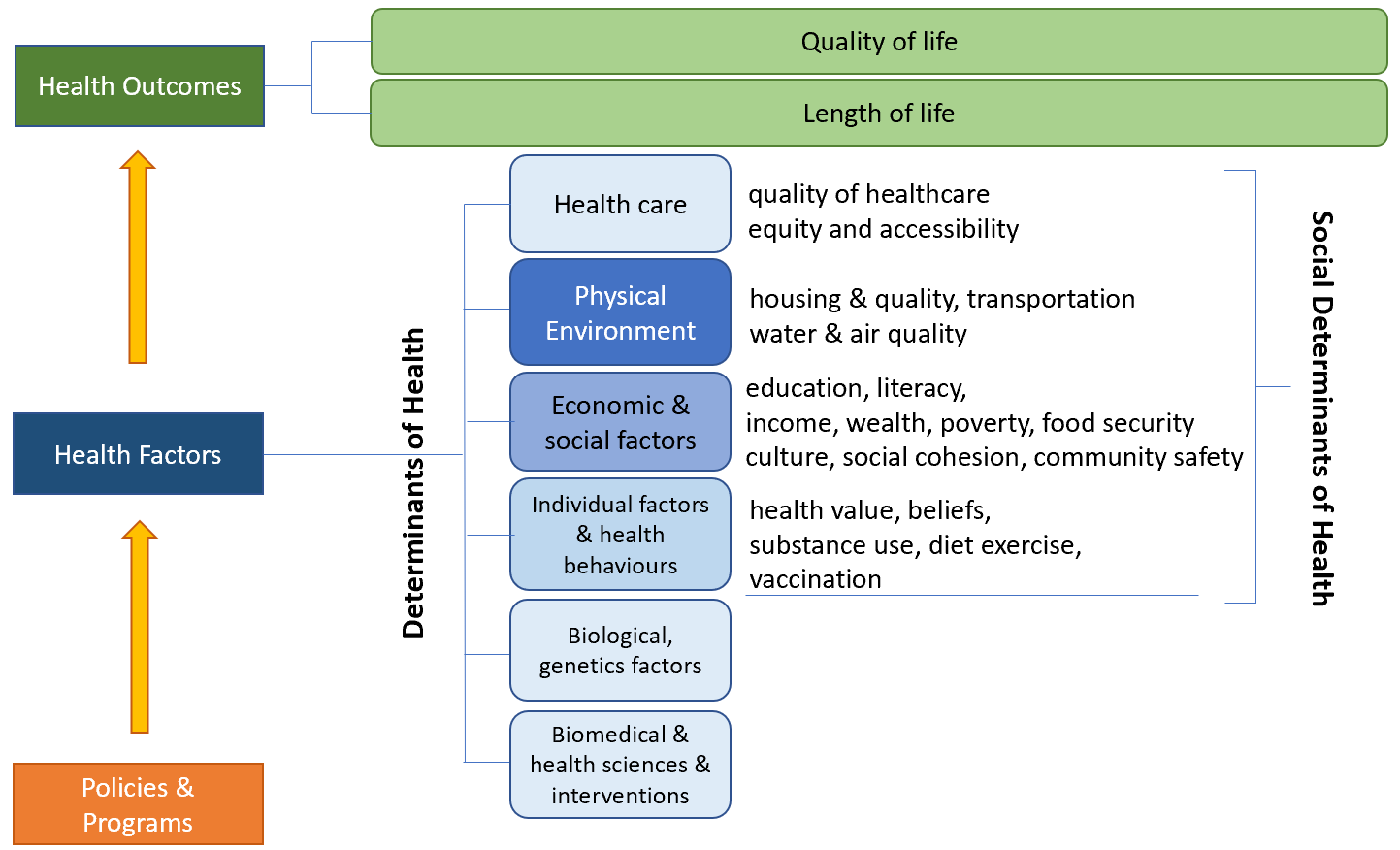
There has been an explosion of breakthrough in the field of machine learning over the past few years. Machine Learning algorithms are capable of a lot and can do wonders for the healthcare sector.

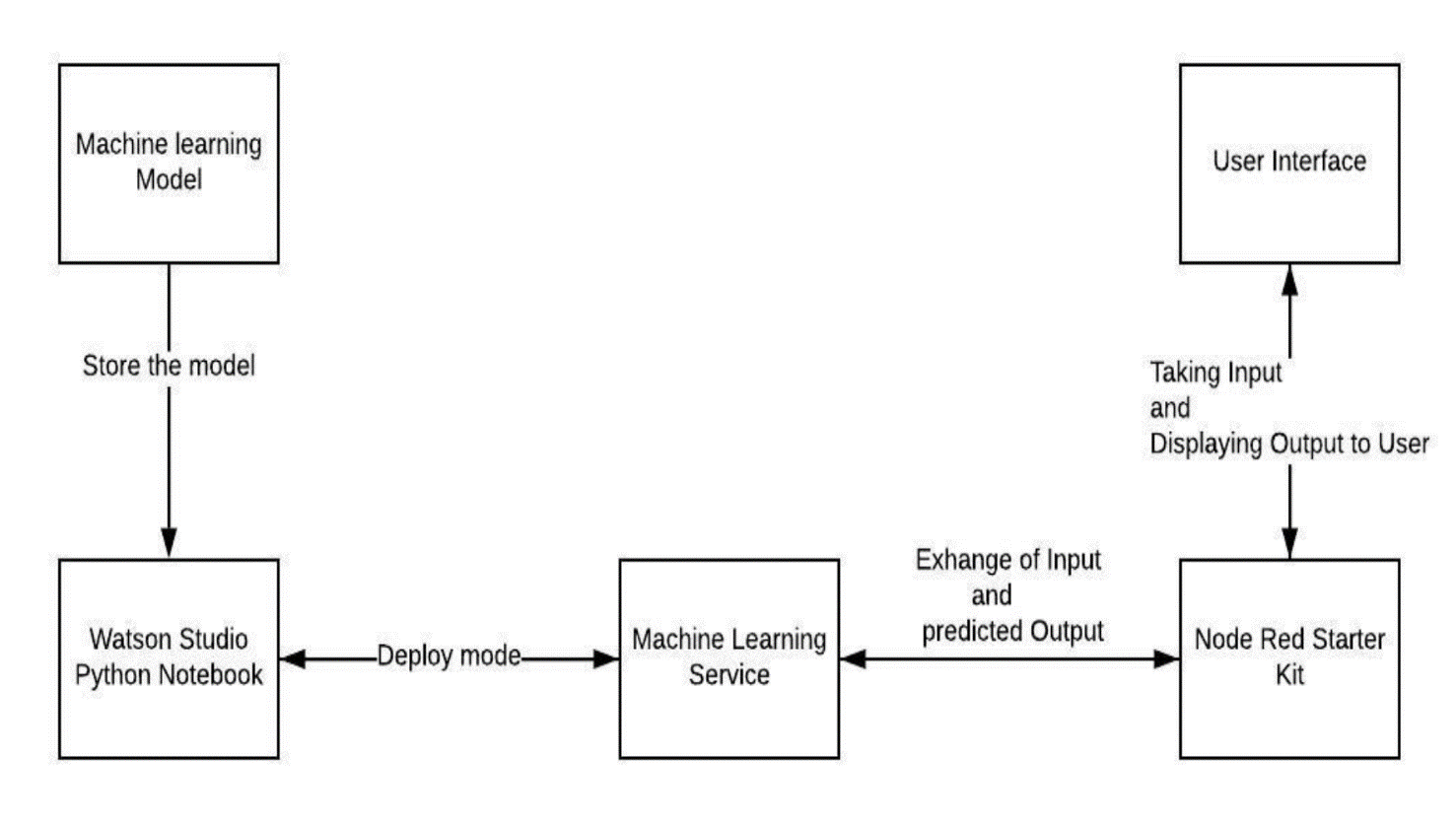
The proposed solution involves the use of Machine learning algorithms such as linear regression and random forest etc. Life expectancy is highly correlated over time among countries and between males and females. these associations can be used to improve forecasts. Here we propose a method for forecasting life expectancy of an individual from a country into certain factors such as adult mortality rate ,infant deaths ,alcohol ,hepatitis B ,measles ,BMI ,polio ,total expenditure ,diphtheria ,HIV/aids ,GDP of a country ,Population ,Income ,Schooling and status of the country in terms of Developing or Developed.

This machine learning model will be made accessible to the users by integrating it with Node-Red to create an user friendly User interface.

**3.Theoritical Analysis**

**3.1 Block Diagram**





**3.2 Hardware/Software designing**

* IBM CLOUD
* IBM WATSON STUDIO
* IMB CLOUD SERVICE- NODE RED APP
* ASSETS
* PROGRAMMING ENVIRONMENT
* JUPYTER NOTEBOOK
* DEFAULT PYTHON

**4.Experimental Investigations**

After importing the data in the Jupyter notebook, I analyzed the dataset.

Variable Descriptions

Format: variable (type) - description

• country (Nominal) - the country in which the indicators are from (i.e. United States of America or Congo)

• year (Ordinal) - the calendar year the indicators are from (ranging from 2000 to 2015)

• status (Nominal) - whether a country is considered to be 'Developing' or 'Developed' by WHO standards

• life expectancy (Ratio) - the life expectancy of people in years for a particular country and year

• adult mortality (Ratio) - the adult mortality rate per 1000 population (i.e. number of people dying between 15 and 60 years per 1000 population); if the rate is 263 then that means 263 people will die out of 1000 between the ages of 15 and 60; another way to think of this is that the chance an individual will die between 15 and 60 is 26.3%

• infant deaths (Ratio) - number of infant deaths per 1000 population; similar to above, but for infants

• alcohol (Ratio) - a country's alcohol consumption rate measured as litres of pure alcohol consumption per capita

• percentage expenditure (Ratio) - expenditure on health as a percentage of Gross Domestic Product (GDP)

• hepatitis-b (Ratio) - number of 1 year olds with Hepatitis B immunization over all 1 year olds in population

• measles (Ratio) - number of reported Measles cases per 1000 population

• BMI (Interval/Ordinal) - average Body Mass Index (BMI) of a country's total population

• under-five deaths (Ratio) - number of people under the age of five deaths per 1000 population

• polio (Ratio) - number of 1 year olds with Polio immunization over the number of all 1 year olds in population

• total expenditure (Ratio) - government expenditure on health as a percentage of total government expenditure

• diphtheria (Ratio) - Diphtheria tetanus toxoid and pertussis (DTP3) immunization rate of 1 year olds

• HIV/aids (Ratio) - deaths per 1000 live births caused by HIV/AIDS for people under 5; number of people under 5 who die due to HIV/AIDS per 1000 births

• GDP (Ratio) - Gross Domestic Product per capita

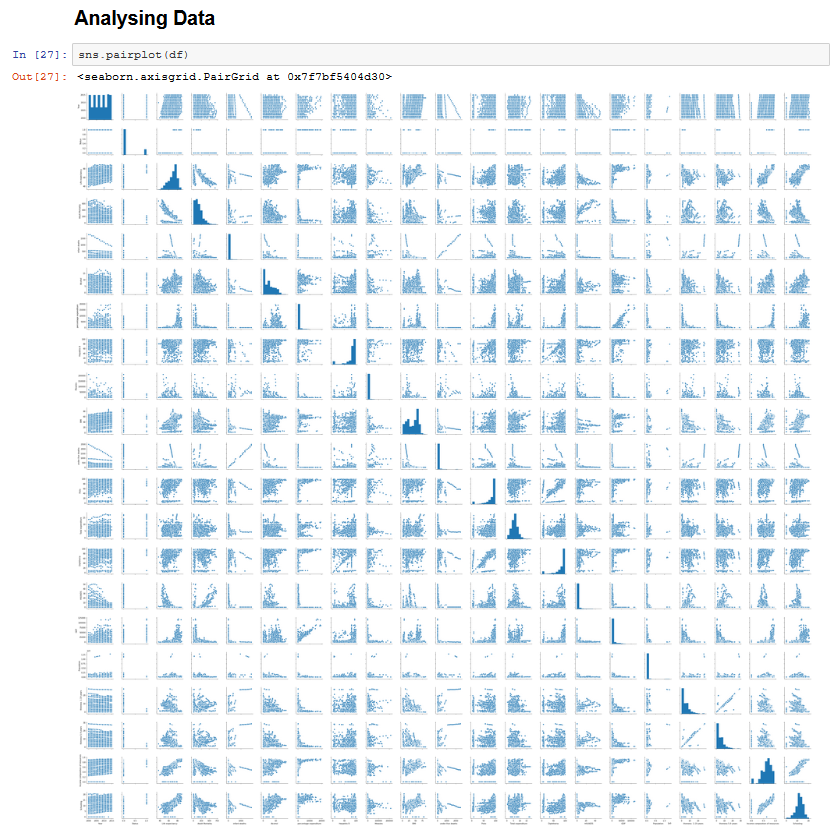
• population (Ratio) - population of a country

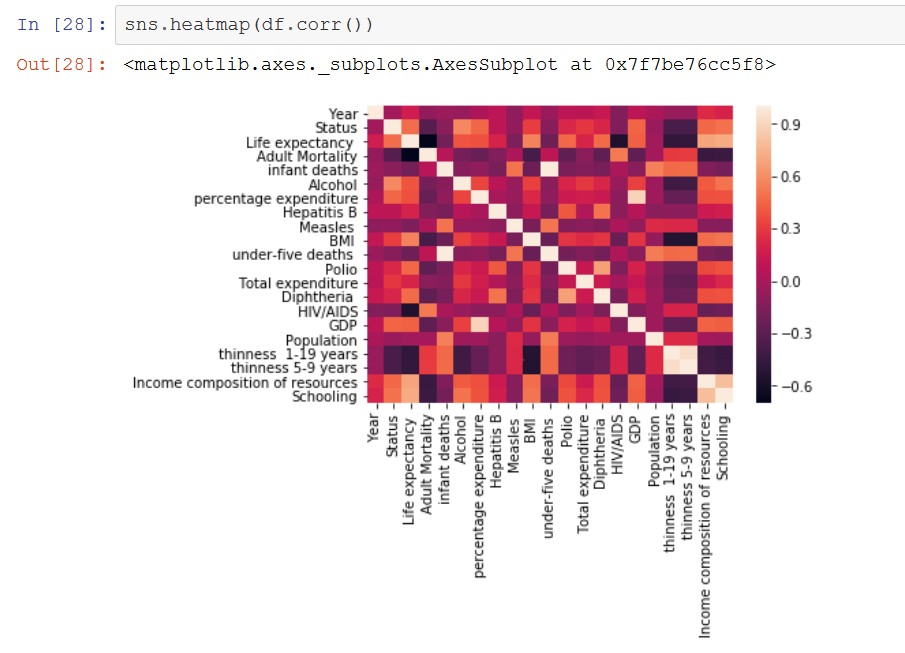
• thinness\_1-19\_years (Ratio) - rate of thinness among people aged *10-19* (Note: variable should be renamed to *thinness\_10-19\_years* to more accurately represent the variable)

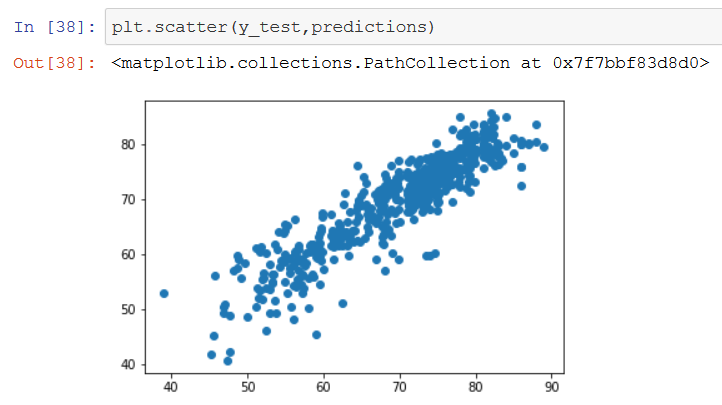
• thinness\_5-9\_years (Ratio) - rate of thinness among people aged 5-9

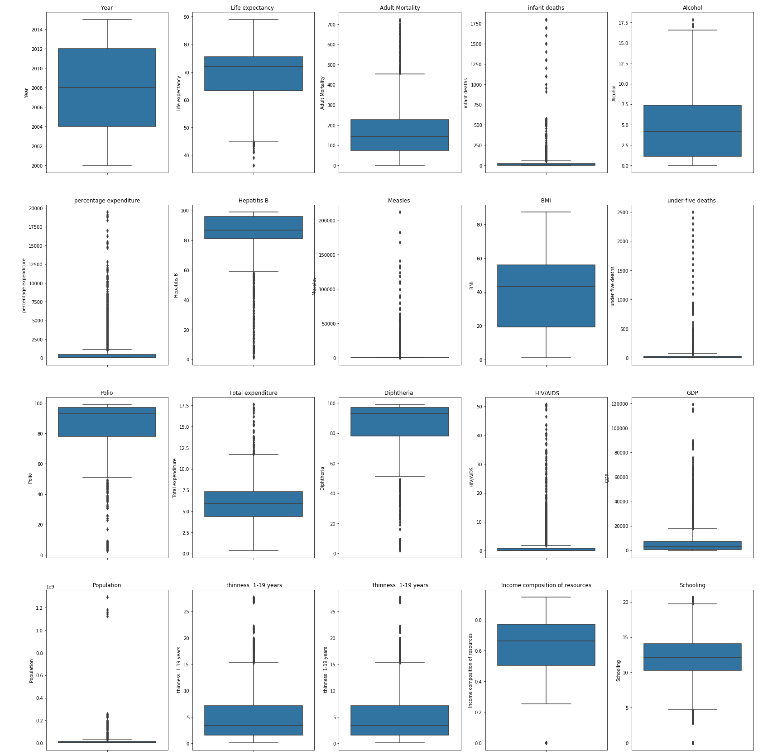
• income composition of resources (Ratio) - Human Development Index in terms of income composition of resources (index ranging from 0 to 1)

• schooling (Ratio) - average number of years of schooling of a population

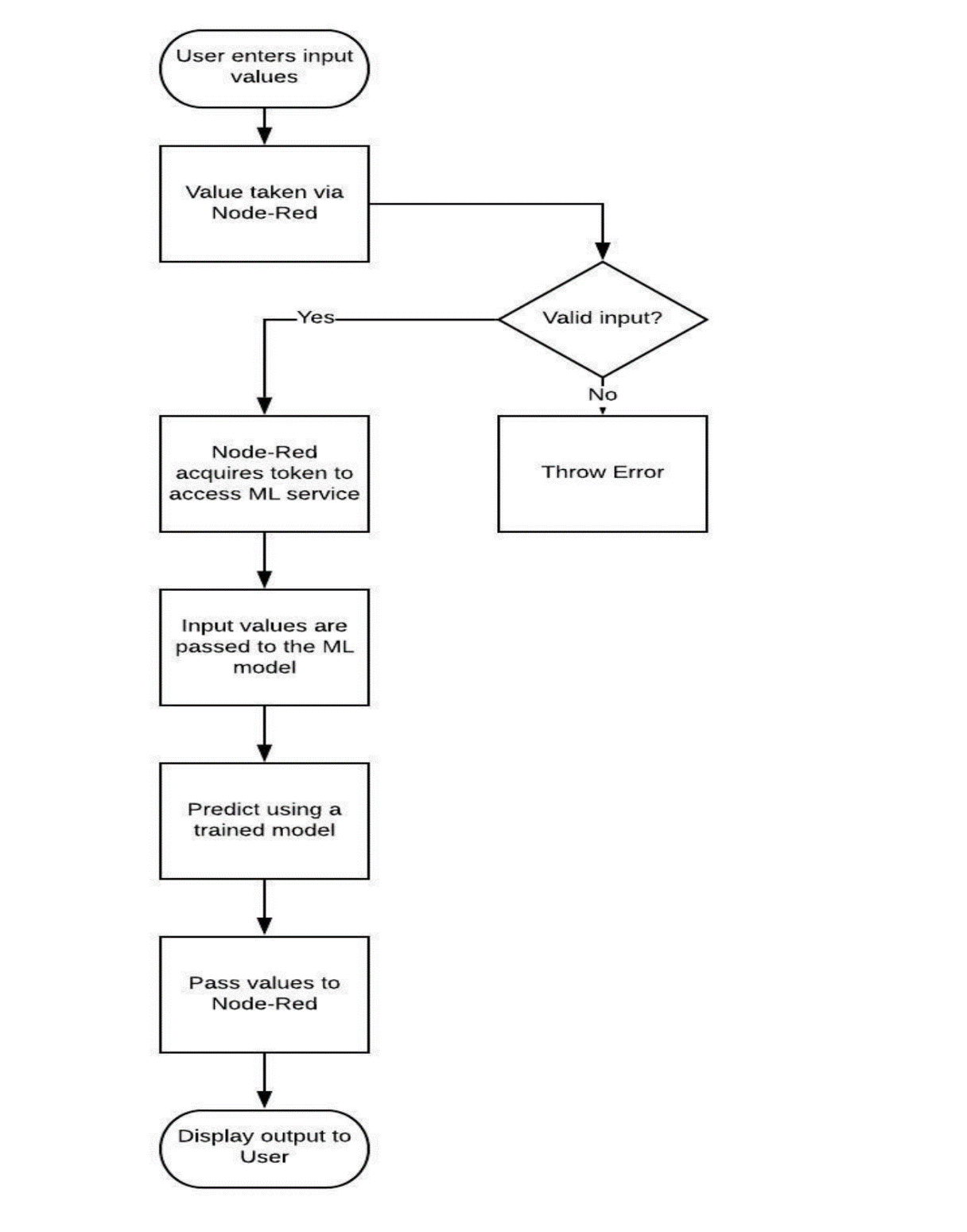








**5.Flowchart**



****

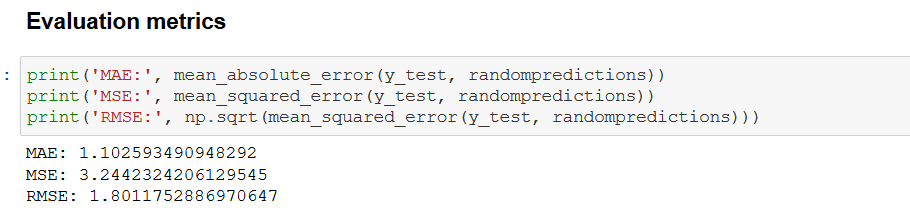
**6.Result**

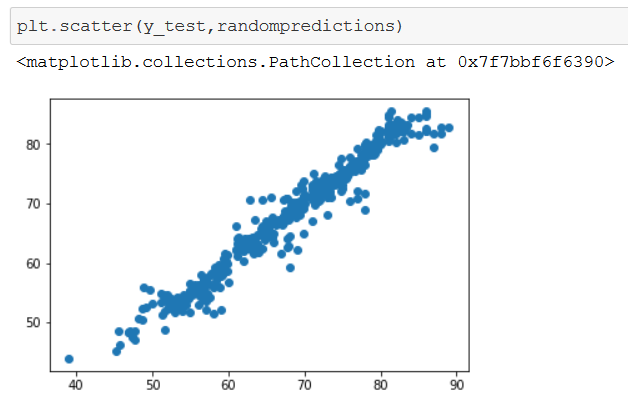
The evaluation metrics that are used are Mean Absolute Error, Mean Squared Error, Root Mean Squared Error and Accuracy.

**Mean Absolute Error (MSE)** of a model refers to the mean of the absolute values of each prediction error on all instances of the test data-set. It’s the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

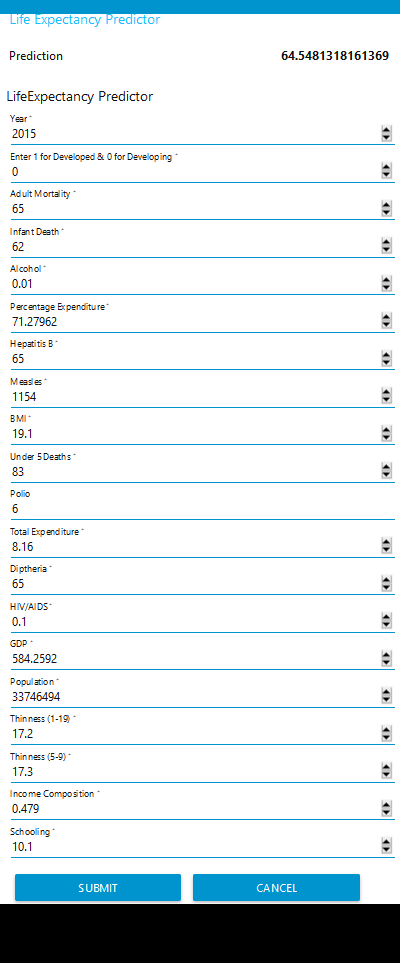
**Mean Squared Error (MSE)**: MSE is the average of the squared error that is used as the loss function for least squares regression: It is the sum, over all the data points, of the square of the difference between the predicted and actual target variables, divided by the number of data points. MSE is calculated by taking the average of the square of the difference between the original and predicted values of the data.

**Root mean squared error (RMSE)**: RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It’s the square root of the average of squared differences between prediction and actual observation.





Here is my UI:



**7.Advantages and Disadvantages**

**Advantages**:

One of the biggest advantages of embedding machine learning algorithms is their ability to improve over time. Machine learning technology typically improves efficiency and accuracy thanks to the ever-increasing amounts of data that are processed.

The application learns the patterns and trends hidden within the data without human intervention which makes predicting much simpler and easier. The more data is fed to the algorithm, the higher the accuracy of the algorithm is. It is also the key component in technologies for automation.

Using Node-Red also simplifies the effort put into a creating the front-end. The programmer doesn’t need extensive knowledge on HTML and JavaScript. It also makes the integration between Machine learning model and the UI much easier.

**Disadvantages**:

Using machine learning interface comes with its own problems. Since the whole point of it is minimize human involvement, it also makes error detection and fixing much more problematic. It takes a lot of time to identify the root cause for the problem.

Machine learning can also be very time-consuming. When the size of the data fed to the machine learning is very large, the computational cost and the time taken to train the model on the data increases drastically. This can increase the cost of resources required to implement the application on a large scale.

At the same time, Node-Red does not give many features to customize our UI.

**8.Applications**

Life expectancy predictions have the potential to be beneficial to individuals, health service providers and governments.

For instance, they would make people more aware of their general health, and its improvement or deterioration over time. This may motivate them to make healthier lifestyle choices.

They could also be used by insurance companies to provide individualised services, such as how some car insurance companies use black-box technology to reduce premiums for more cautious drivers.

Governments may be able to use predictions to more efficiently allocate limited resources, such as social welfare assistance and health care funding, to individuals and areas of greater need.

**9.Conclusion**

Predicting lifespan of human beings can greatly alter our lives. Human behaviour and activities are so unpredictable, it may almost be impossible to correctly predict lifespan. However, with the help of Machine learning algorithms such as Regression models, we can get close to predicting a roundabout value.

This breakthrough can widely impact health sectors and economic sectors by improving the resources, funds and services provided to the common people. It can also increase the ease of access to the individuals.

With the help of Machine Learning algorithms, one can ease the process of automating the application and predicting the expectancy with an admirable accuracy. It also reduces the effort and time put into deploying the application and making it more accessible to the users.

**10.Future Scope**

Government policies affecting water quality and health care have many benefits, such as morbidity and workdays lost, benefits not limited to the life expectancy benefits focussed on here. However, the life expectancy benefits of affecting those variables should be added to other benefits of education, improved access to clean water and drugs, and AIDS prevention. Doing so offers the potential to result in better allocation of costly scarce resources in countries at various stages of development.

Future research could build a forecasting model incorporating mortality trends along with the impacts of socioeconomic variables investigated here, as well as others that are likely to become available as data measurement and accessibility improves over time.

**11.Bibliography**

• <https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html>

• <https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service>

• <https://www.ibm.com/watson/products-services>

• https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application

**12.Appendix**

**A. Source Code**

# PREDICTING LIFE EXPECTANCY USING PYTHON

# Import necessary Libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

# Import Dataset

import types

import pandas as pd

from botocore.client import Config

import ibm\_boto3

def \_\_iter\_\_(self): return 0

# @hidden\_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.

# You might want to remove those credentials before you share the notebook.

client\_5dfc2efa260b44c4837666de787f874d = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='BSOK-H8chL0SvklCjlUSk94LjObvQ2ZGX-fLlRl\_35N2',

ibm\_auth\_endpoint="https://iam.cloud.ibm.com/oidc/token",

config=Config(signature\_version='oauth'),

endpoint\_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client\_5dfc2efa260b44c4837666de787f874d.get\_object(Bucket='lifeexpectancyprediction-donotdelete-pr-12qc7sofgvnhyl',Key='Life Expectancy Data.csv')['Body']

# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object

if not hasattr(body, "\_\_iter\_\_"): body.\_\_iter\_\_ = types.MethodType( \_\_iter\_\_, body )

# If you are reading an Excel file into a pandas DataFrame, replace `read\_csv` by `read\_excel` in the next statement.

df = pd.read\_csv(body)

df.head()

5 rows × 22 columns

# Reading data

df.shape

(2938, 22)

df.columns

Index(['Country', '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'],

dtype='object')

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2938 entries, 0 to 2937

Data columns (total 22 columns):

Country 2938 non-null object

Year 2938 non-null int64

Status 2938 non-null object

Life expectancy 2928 non-null float64

Adult Mortality 2928 non-null float64

infant deaths 2938 non-null int64

Alcohol 2744 non-null float64

percentage expenditure 2938 non-null float64

Hepatitis B 2385 non-null float64

Measles 2938 non-null int64

BMI 2904 non-null float64

under-five deaths 2938 non-null int64

Polio 2919 non-null float64

Total expenditure 2712 non-null float64

Diphtheria 2919 non-null float64

HIV/AIDS 2938 non-null float64

GDP 2490 non-null float64

Population 2286 non-null float64

thinness 1-19 years 2904 non-null float64

thinness 5-9 years 2904 non-null float64

Income composition of resources 2771 non-null float64

Schooling 2775 non-null float64

dtypes: float64(16), int64(4), object(2)

memory usage: 505.0+ KB

df.describe()

# Preprocessing Data

df = df.drop(['Country'], axis=1)

df.columns

Index(['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'],

dtype='object')

# Manipulating data

df.isnull().sum(axis=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

#removing null values

df=df.fillna(df.mean())

df.isnull().sum()

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

df.replace(to\_replace=['Developing', 'Developed'],

value= [0, 1],

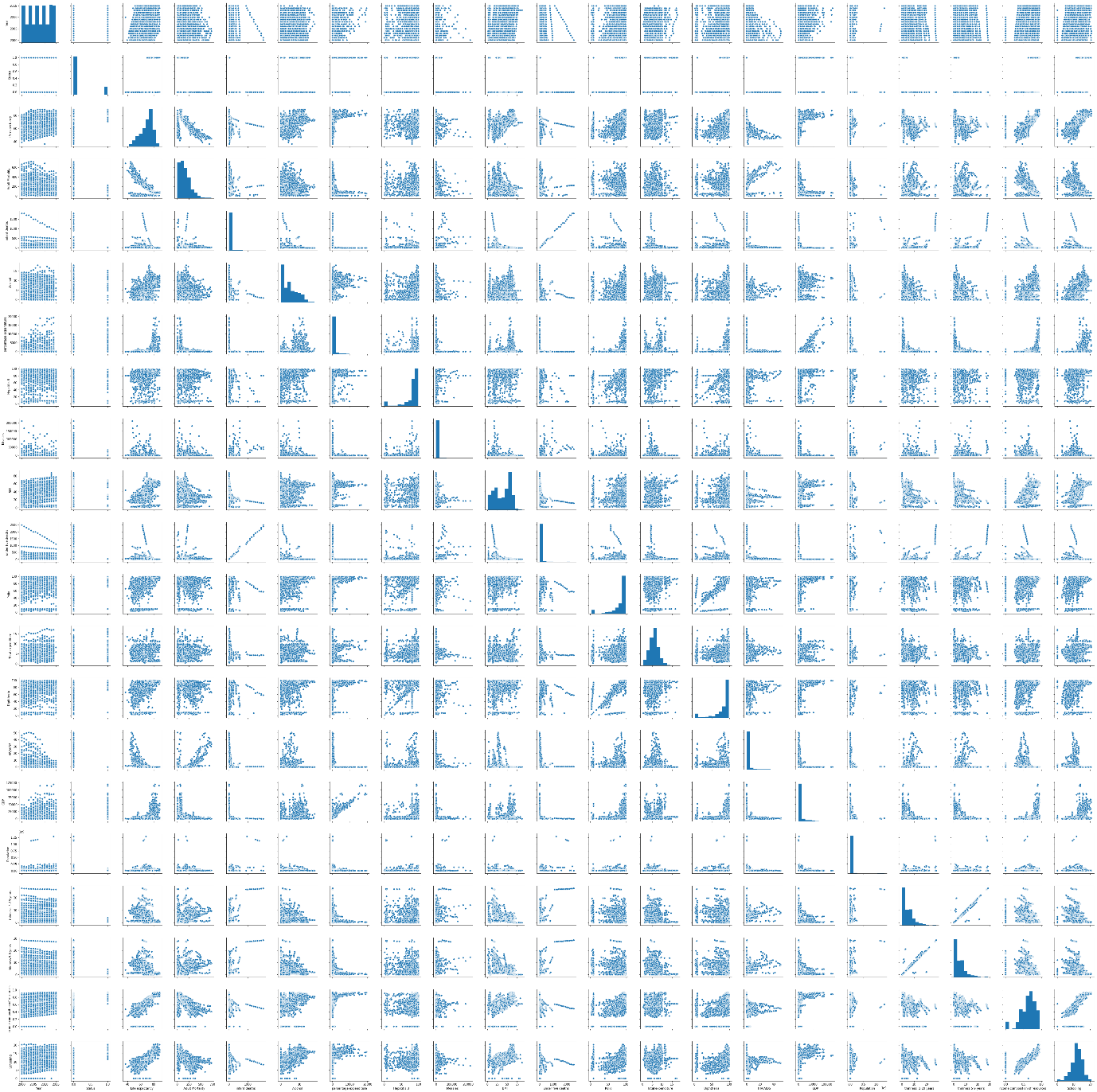
inplace=True)

df.head()

Analysing Data

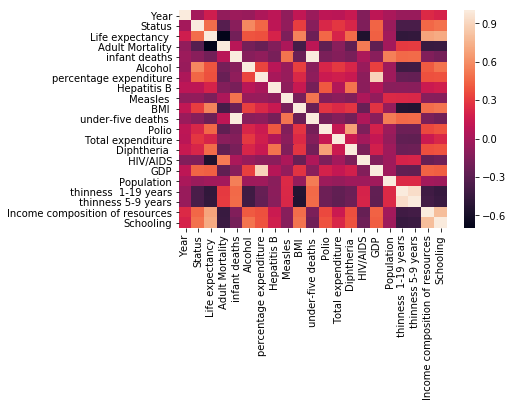
sns.pairplot(df)

<seaborn.axisgrid.PairGrid at 0x7f7bf5404d30>



sns.heatmap(df.corr())

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f7be76cc5f8>



#box plots

columns = {1:'Year', 2: 'Life expectancy ', 3: 'Adult Mortality', 4: 'infant deaths',

5: 'Alcohol' , 6: 'percentage expenditure', 7: 'Hepatitis B',

8: 'Measles ', 9: ' BMI ', 10: 'under-five deaths ', 11: 'Polio', 12: 'Total expenditure',

13: 'Diphtheria ', 14: ' HIV/AIDS', 15: 'GDP', 16: 'Population',

17: ' thinness 1-19 years', 18: ' thinness 1-19 years',

19: 'Income composition of resources', 20: 'Schooling'}

plt.figure(figsize=(28, 30))

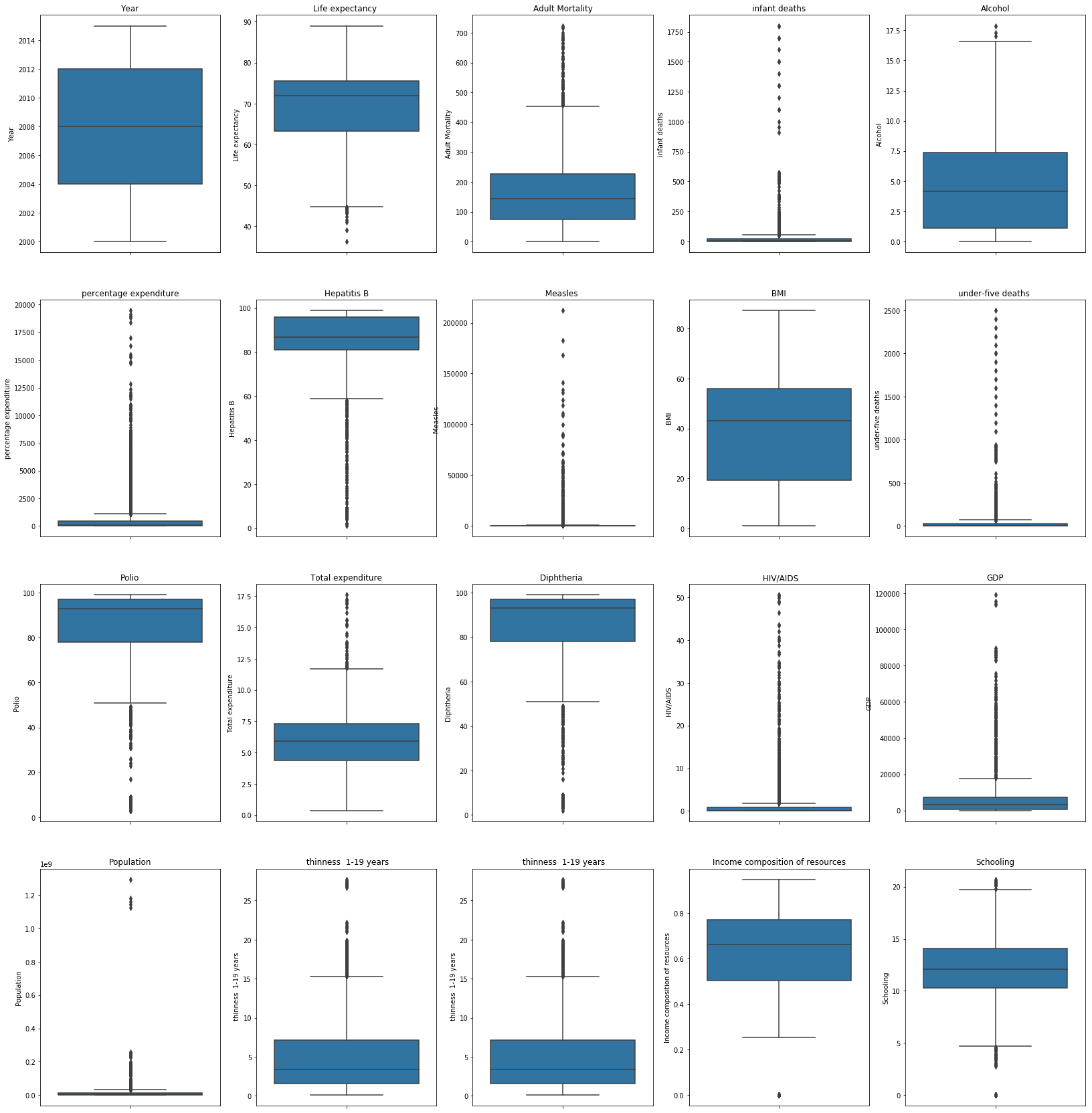
for i, column in columns.items():

plt.subplot(4,5,i)

sns.boxplot(df[column], orient='v')

plt.title(column)

plt.show()



y=df["Life expectancy "]

x=df.drop(["Life expectancy "],axis=1)

x.head()

# Training Model

#splitting the data set inton training and testing set

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=101)

print("Training features shape:",x\_train.shape)

print("Training label shape:",y\_train.shape)

print("Testing features shape:",x\_test.shape)

print("Testing label shape:",y\_test.shape)

Training features shape: (2350, 20)

Training label shape: (2350,)

Testing features shape: (588, 20)

Testing label shape: (588,)

# Imposing Machine Learing models

## 1.Linear Regression model

from sklearn.linear\_model import LinearRegression

model=LinearRegression()

model.fit(x\_train,y\_train)

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None,

normalize=False)

print(model.intercept\_)

68.1227338510878

coeff\_df=pd.DataFrame(model.coef\_,x.columns,columns=['Coefficient'])

coeff\_df

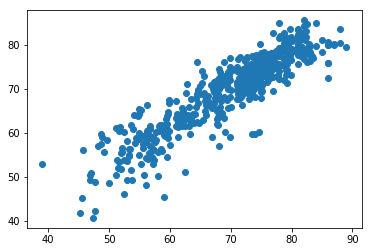
|  | **Coefficient** |
| --- | --- |
| **Year** | -6.563948e-03 |
| **Status** | 1.255024e+00 |
| **Adult Mortality** | -1.937106e-02 |
| **infant deaths** | 1.080129e-01 |
| **Alcohol** | 6.672955e-02 |
| **percentage expenditure** | 4.911270e-05 |
| **Hepatitis B** | -1.814479e-02 |
| **Measles** | -2.169758e-05 |
| **BMI** | 4.580954e-02 |
| **under-five deaths** | -8.097175e-02 |
| **Polio** | 3.096221e-02 |
| **Total expenditure** | 6.698179e-02 |
| **Diphtheria** | 3.883636e-02 |
| **HIV/AIDS** | -4.656855e-01 |
| **GDP** | 4.403791e-05 |
| **Population** | 1.482320e-10 |
| **thinness 1-19 years** | -1.115789e-01 |
| **thinness 5-9 years** | 3.065350e-02 |
| **Income composition of resources** | 5.950169e+00 |
| **Schooling** | 6.492965e-01 |

### Predictions from model

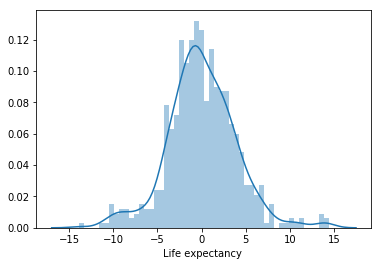
predictions=model.predict(x\_test)

plt.scatter(y\_test,predictions)

<matplotlib.collections.PathCollection at 0x7f7bbf83d8d0>



sns.distplot((y\_test-predictions),bins=50);



### Regression Evaluation Metrics

from sklearn import metrics

print("MAE:",metrics.mean\_absolute\_error(y\_test,predictions))

print("MSE:",metrics.mean\_squared\_error(y\_test,predictions))

print("RSME:",np.sqrt(metrics.mean\_squared\_error(y\_test,predictions)))

MAE: 2.9323808691589193

MSE: 15.26405732399841

RSME: 3.906924279276271

# 2.Random Forest Regression

from sklearn.ensemble import RandomForestRegressor

random=RandomForestRegressor(n\_estimators=40,random\_state=50)

random.fit(x\_train,y\_train)

RandomForestRegressor(bootstrap=True, criterion='mse', max\_depth=None,

max\_features='auto', max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=40, n\_jobs=None,

oob\_score=False, random\_state=50, verbose=0, warm\_start=False)

randompredictions=random.predict(x\_test)

plt.scatter(y\_test,randompredictions)

<matplotlib.collections.PathCollection at 0x7f7bbf6f6390>

sns.distplot((y\_test-randompredictions),bins=50)

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f7bbf71a4a8>

from sklearn.metrics import r2\_score, mean\_absolute\_error, mean\_squared\_error

print("R² score for Linear Regression Model: ", end='')

print(r2\_score(predictions, y\_test))

print("R² score for RandomForest Regression Model: ", end='')

print(r2\_score(randompredictions, y\_test))

R² score for Linear Regression Model: 0.7846189748453709

R² score for RandomForest Regression Model: 0.9613932696938012

As R² score of random forest model is higher than linear regression model ,so we will use random forest as our model

### Evaluation metrics

print('MAE:', mean\_absolute\_error(y\_test, randompredictions))

print('MSE:', mean\_squared\_error(y\_test, randompredictions))

print('RMSE:', np.sqrt(mean\_squared\_error(y\_test, randompredictions)))

MAE: 1.102593490948292

MSE: 3.2442324206129545

RMSE: 1.8011752886970647

# Model Deployment

#######################################################################################

Synchronous deployment creation for uid: 'ce081e78-7032-43c7-9cc9-ab92457798d6' started

############################################################