A PROJECT REPORT

ON

PREDICTING LIFE EXPECTANCY

USING MACHINE LEARNING

Submitted in partial fulfillment for the award of

Internship

In

Machine Learning

Submitted By

KISHORE RAJU (SASTRA Deemed To Be University)

SBID :SB20200052888

PROJECT ID : SPS_PRO_215

Under the guidance of

Mentors

CONTENTS

1. INTRODUCTION:

1.1 OVERVIEW

Life expectancy plays an important role when decisions about the final phase of life need to be made. Good prognostication for example helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more broadly: facilitates Advance Care Planning. Advance Care Planning is the process during which patients make decisions about the health care they wish to receive in the future, in case the patient loses the capacity of making decisions or communicating about them .The intern program is intended to create a Life Expectancy prediction model with a User Interface. A Regression Machine Learning model is created that leverages historical data to predict Life Expectancy of a country given various features. The development is done on IBM cloud using various services i.e. IBM Watson Studio, Machine Learning Service, Node-RED and Cloudant. 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. It is very important to predict average life expectancy of a country to analyse further requirements to increase its rate of growth or stabilise the rate of growth in that country. So this is a typical Regression Machine Learning project that leverages historical data to predict insights into the future. The end product will be a webpage where you need to give all the required inputs and then submit it . Afterwards it will predict the life expectancy value based on your regression technique.

1.2 PURPOSE

The purpose of the project is to design a model for predicting Life Expectancy rate of a country given various features such as year, GDP, education, alcohol intake of people in the country, expenditure on health care system and some specific disease related deaths that happened in the country are given. Life expectancy is one of the most important factors in end-of-life decision making. Good prognostication helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more broadly: facilitates Advance Care Planning in a country. Advance Care Planning improves the quality of the final phase of life by stimulating doctors to explore the preferences for end-of-life care with their patients, and people close to the patients. Physicians, however, tend to overestimate life expectancy, and miss the window of opportunity to initiate Advance Care Planning.

2. LITERATURE SURVEY:

2.1.EXISTING PROBLEM

Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries. Hence, this gives motivation to resolve both the factors stated previously by formulating a regression model based on mixed effects model and multiple linear regression while considering data from a period of 2000 to 2015 for all the countries.

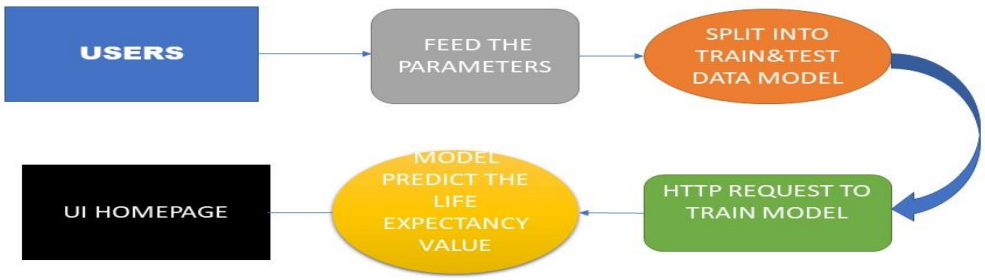
Important immunization like Hepatitis B, Polio and Diphtheria will also be considered. In a nutshell, this project will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. Since the observations the dataset are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

2.2. PROPOSED PROBLEM

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. The machine learning model built using historical data 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.

3. THEORETICAL ANALYSIS:

3.1 BLOCK DIAGRAM



The user will feeds the dataset as csv files and then split the dataset into train and test data then the trained model in the form as HTTP request. The trained model will predict the Life expectancy based on different parameters, then the predicted output will shown in Homepage in UI.

Life expectancy plays an important role when decisions about the final phase of life need to be made. Good prognostication for example helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more broadly, facilitates Advance Care Planning. Advance Care Planning is the process during which patients make decisions about the health care they wish to receive in the future, in case the patient loses the capacity of making decisions or communicating about them

3.2. SOFTWARE DESIGN

The IBM cloud offers the more services to predict the model, here we can use Auto AI or Watson Studio. Auto AI will automatically predict the accuracy by using dataset without any code .The regression model built in python is deployed on IBM cloud. The Node-RED application then sends HTTP request with all the required parameters to the trained model. The model then sends the HTTP response which is then parsed and displayed on the UI.

4 . EXPERIMENTAL INVESTIGATIONS :

4.1 FACTORS AFFECTING LIFE EXPECTANCY

Below are the factors (given in the dataset) which affect life expectancy of a country.

1.Adult Mortality: Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population).

2.Infant Deaths: Number of Infant Deaths per 1000 population

3.Alcohol: Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol).

4.Percentage Expenditure: Expenditure on health as a percentage of Gross Domestic Product per capita(%).

5.Hepatitis B: Hepatitis B immunization coverage among 1-year-olds (%).

6.Measles: Measles - number of reported cases per 1000 population.

7.BMI: Average Body Mass Index of the entire population.

8.Under-five deaths: Number of under-five deaths per 1000 population.

9.Polio: Polio (Pol3) immunization coverage among 1-year-olds (%).

10.Total Expenditure: General government expenditure on health as a percentage of total government expenditure (%).

11.Diphtheria: Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%).

12.HIV/AIDS: Deaths per 1 000 live births HIV/AIDS (0-4 years).

13.GDP: Gross Domestic Product per capita (in USD).

14.Population: Population of the country.

15.Thinness 5-9 years: Prevalence of thinness among children for Age 5 to 9(%).

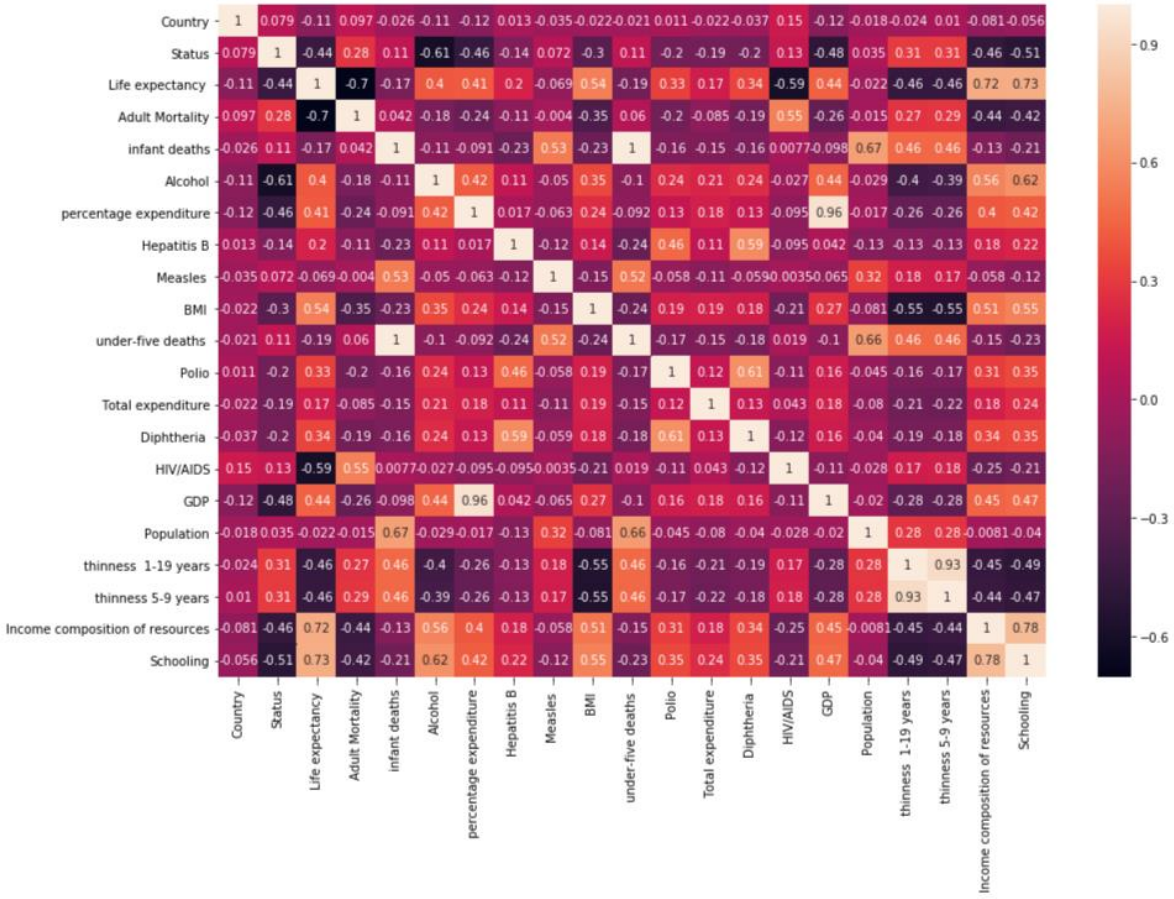
16.Thinness 1-19 years: Prevalence of thinness among children and adolescents for Age 10 to19(%).

17.Income composition of resources: Human Development Index in terms of income composition of resources (index ranging from 0 to 1).

18.Schooling: Number of years of Schooling(years).

4.2 CORRELATION BETWEEN FACTORS &LIFE EXPECTANCY:

Below is the correlation heat map of the dataset:



It is observable that Schooling, Income composition of resources and Adult Mortality are highly correlated to Average Life Expectancy.

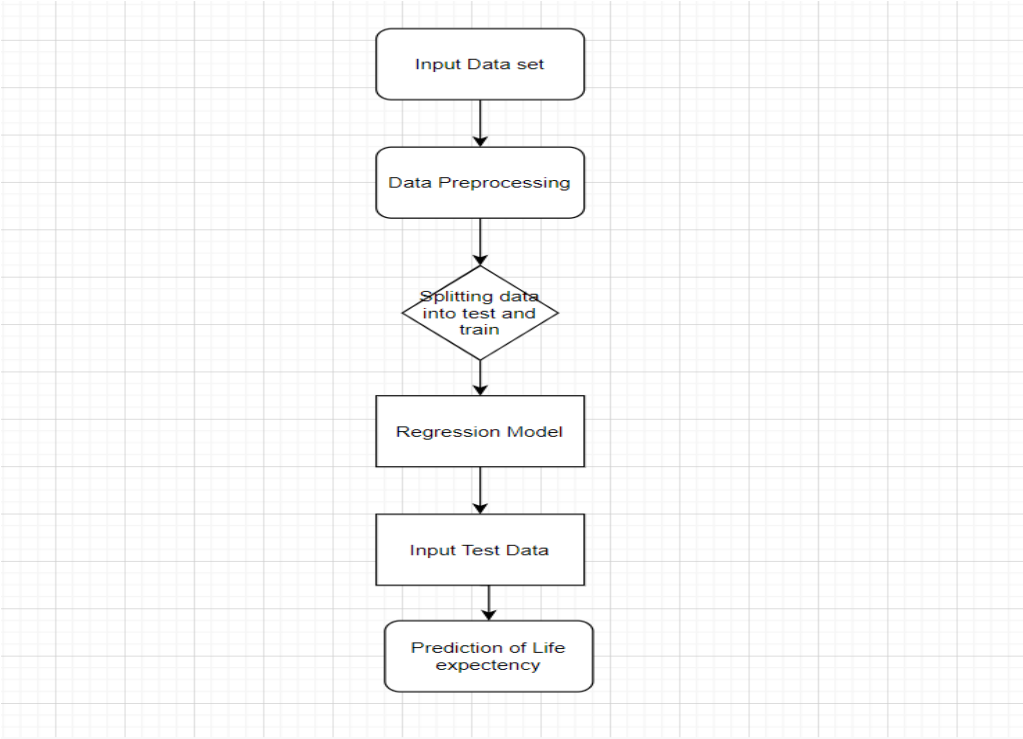
4.2 IMPLEMENTING REGRESSOR MODEL

***** An **extra-trees regressor**. This class implements a meta estimator that fits a number of randomized decision **trees** (a.k.a. **extra-trees**) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. ... The maximum depth of the **tree regressor.**

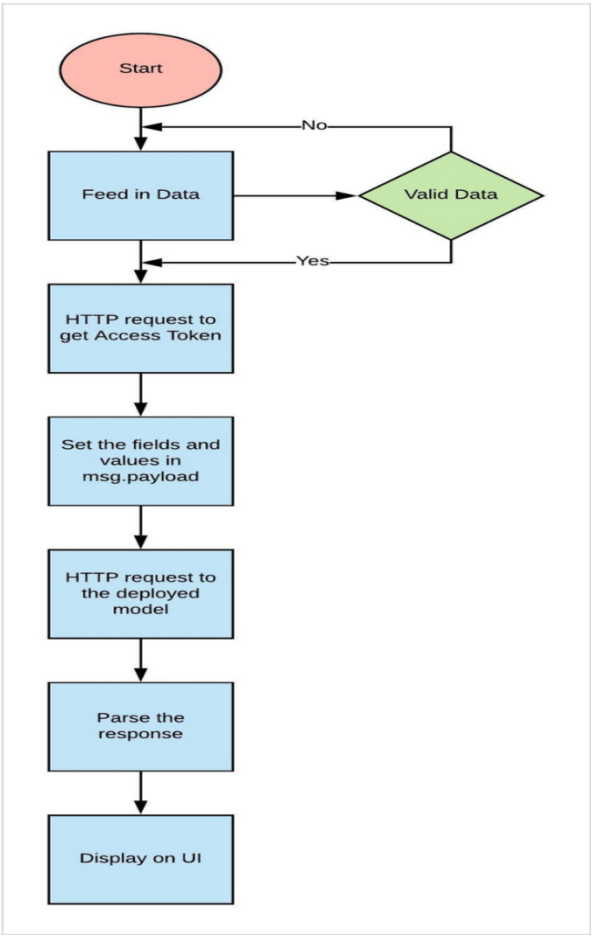
*** Extremely Randomized Trees Classifier(Extra Trees Classifier)** is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a "forest" to output it's classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

5.FLOW CHART:

5.1 FOR IBM WATSON STUDIO:

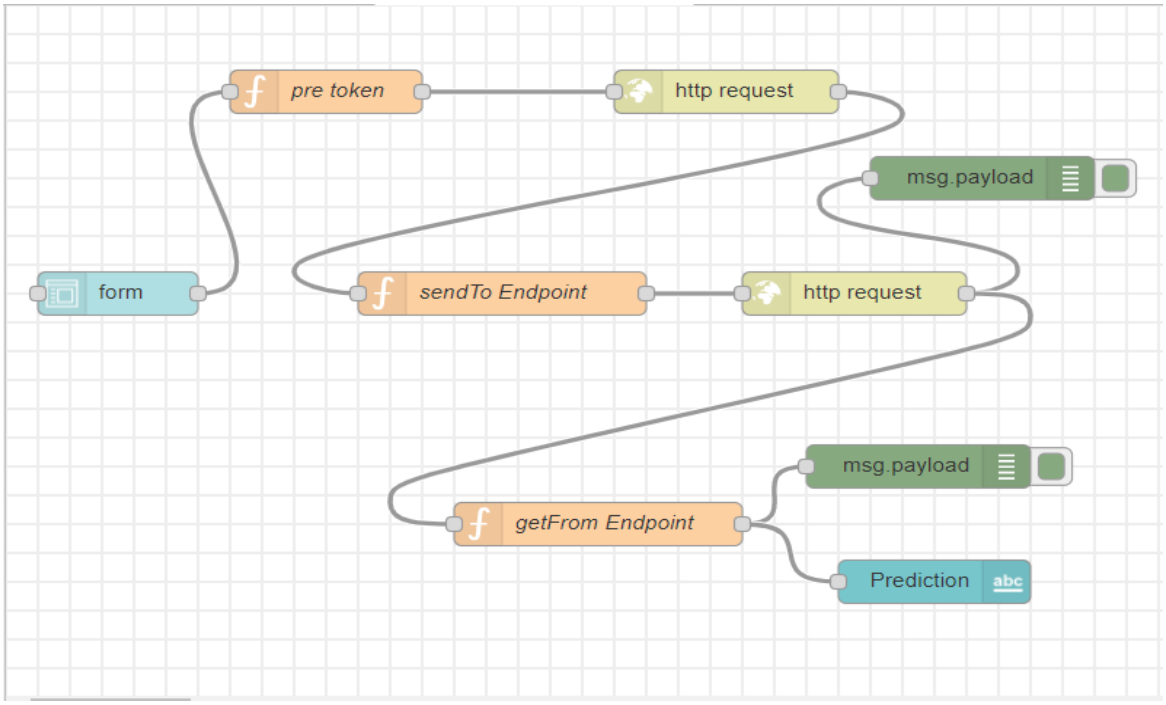


5.2 FOR UI MODEL:

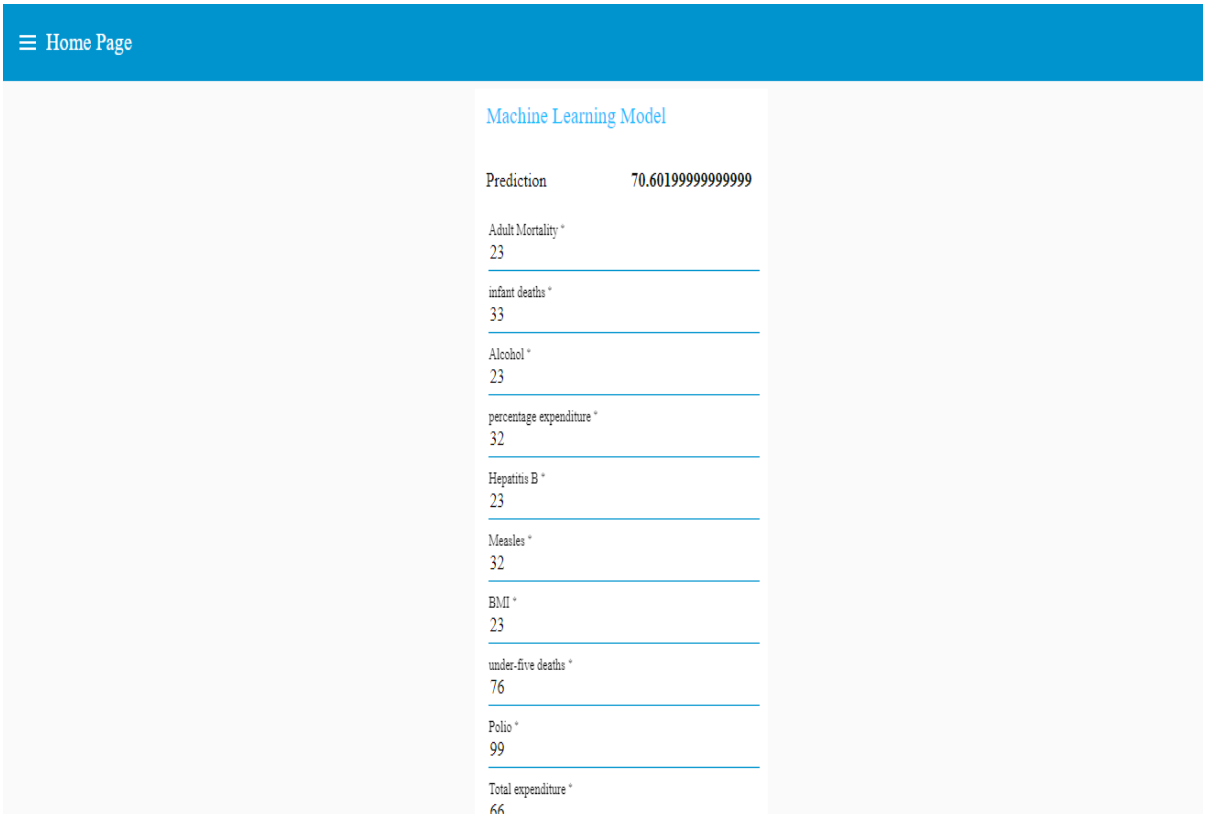


6. RESULT:

6.1 NODE-RED FLOW:



6.2 USER INTERFACE (UI):



7.ADVANTAGES AND DISADVANTAGES:

7.1 ADVANTAGE:

- Health Inequalities: Life expectancy has been used nationally to monitor health inequalities of a country.
- Reduced Costs: This is a simple webpage and can be accessed by any citizen of a country to calculate life expectancy of their country and does not required any kind of payment neither for designing nor for using.
- User Friendly Interface: This interface requires no background knowledge of how to use it. It's a simple interface and only ask for required values and predict the output.

7.2 DISADVANTAGE:

- Extra Tree Regressor has low variant when compare with Random Tree Regressor.
- Wrong Prediction: As it depends completely on user, so if user provides some wrong values then it will predict wrong value.
- Average Prediction: The model predicts average or approximate value with 97.07% accuracy but not accurate value.

8. APPLICATION :

- Life expectancy is the statistical age that a person is expected to live until, based on actuarial data.
- Based on actuarial science, life expectancy takes into account several individual-level as well as population-level factors to arrive at a figure.
- Life expectancy is used in pricing and underwriting life insurance and insurance products like annuities, as well as in retirement and pension planning.

9. CONCLUSION:

- The product is a webpage created and deployed on node-red app of IBM cloud. The backend of webpage is an Extra Tree Regressor Model with 97.07% R2 score created and deployed on Watson studio using machine learning service.
- The web-page has input fields similar to dataset columns such as Country, BMI, percentage expenditure, Alcohol etc and an output field named as prediction i.e. similar to dataset column Life expectancy which gives the life expectancy prediction based on the inputted values.

10. FUTURE SCOPE:

- The government can plan health services better using the data and future predictions. Life expectancy plays a major role in development of a country, hence, using predictions and trends, the health infrastructure can be improved.
- A mobile application can be developed that uses personal health data (from Smart Watch and Health apps) and historical data of the country that user lives in and predict the expected life span of that user.

11. BIBLOGRAPHY :

1. Statistical Analysis on factors influencing Life Expectancy Dataset:

https://www.kaggle.com/kumarajarshi/life-expectancy-who/metadata

2. Deploying an Auto AI model in IBM cloud:

https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/autoai-depoy-model.html

3.Using the machine learning model in IBM Watson studio:

https://cloud.ibm.com/docs/watsonknowledge-studio?topic=watson-knowledge-studio-publish-ml

4.Infuse AI into your applications with Watson AI to make more accurate predictions:

https://www.ibm.com/watson/products-services

5. Get an understanding of Machine Learning:

https://developer.ibm.com/technologies/machine-learning/series/learning-path-machinelearning-for-developers

6.create a Node-RED starter application in the IBM Cloud, including a Cloudant database to store the application flow configuration:

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

7.Endpoint reference for node-red integration:

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

12.APPENDIX:

A. SOURCE CODE:

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING :

IMPORTED THE REQUIRED LIBRARIES :

In [27]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
pd.options.display.float_format='{:.5f}'.format
import warnings
import math
#import libraries for pipelining
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.compose import ColumnTransformer
#import libraries for train and test
from sklearn.model_selection import train_test_split
#importExtraTreeRegressor for model fit and prediction
from sklearn.ensemble import ExtraTreesRegressor
#import libraries for accuracy and error caluclation
from sklearn.metrics import mean_squared_error,r2_score
#import libraries for model building and deployement
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

IMPORTED THE DATASET (CSV FILE) :

In [28]:
```
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_4d05ddc15f7c4cc18fbde4329e8184ea = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='Eb35cFNlTFT1A_pNT_eQh6EIlU8V-KtmoWqmtFwDHVea',
    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_4d05ddc15f7c4cc18fbde4329e8184ea.get_object(Bucket='predectinglifeexpectancy-donotdelete-pr-j9xr1doddjl66v',Key='Life Expectancy Dataset.csv.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 )

df_data_1 = pd.read_csv(body)
df_data_1.head()
```

Out[28]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	T e
0	Afghanistan	2015	Developing	65.00000	263.00000	62	0.01000	71.27962	65.00000	1154	...	6.00000	8.
1	Afghanistan	2014	Developing	59.90000	271.00000	64	0.01000	73.52358	62.00000	492	...	58.00000	8.
2	Afghanistan	2013	Developing	59.90000	268.00000	66	0.01000	73.21924	64.00000	430	...	62.00000	8.
3	Afghanistan	2012	Developing	59.50000	272.00000	69	0.01000	78.18422	67.00000	2787	...	67.00000	8.
4	Afghanistan	2011	Developing	59.20000	275.00000	71	0.01000	7.09711	68.00000	3013	...	68.00000	7.

5 rows × 22 columns

In [29]:
```
df.columns
```

Out[29]:
```
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'Health expenditure percentage',
       'Hepatitis B', 'Measles ', 'BMI', 'under-five deaths ', 'Polio',
       'Government expenditure', 'Diphtheria ', ' HIV/AIDS', 'GDP',
       'Population', 'Thinness 10-19 years', 'Thinness 5-9 years', 'Income',
       'Schooling'],
      dtype='object')
```

In [30]:
```
df.rename(columns={'Income composition of resources':'Income',
                   ' thinness  1-19 years':'Thinness 10-19 years',
                  ' thinness 5-9 years':'Thinness 5-9 years',
                   'percentage expenditure':'Health expenditure percentage',
                   'Total expenditure':'Government expenditure',
                   ' BMI ':'BMI'},inplace=True)
```

In [31]:
```
df.isnull().sum()
```

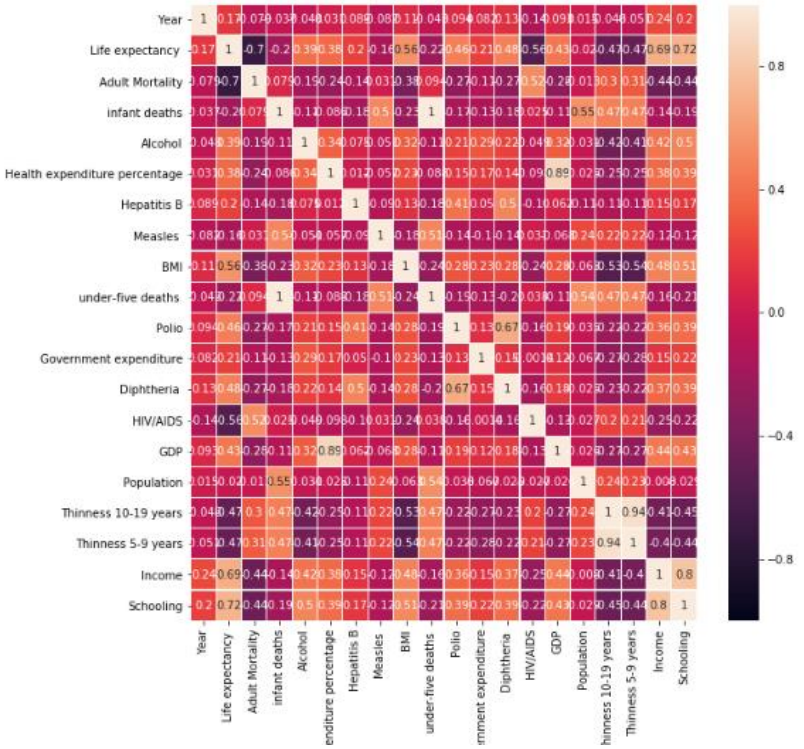
Out[31]:
```
Country                          0
Year                             0
Status                           0
Life expectancy                  0
Adult Mortality                  0
infant deaths                    0
Alcohol                          0
Health expenditure percentage    0
Hepatitis B                      0
Measles                          0
BMI                              0
under-five deaths                0
Polio                            0
Government expenditure           0
Diphtheria                       0
 HIV/AIDS                        0
GDP                              0
Population                       0
Thinness 10-19 years             0
Thinness 5-9 years               0
Income                           0
Schooling                        0
dtype: int64
```

In [32]:
```
df = df.fillna(df.mean())
```

PLOTTING THE HEAPMAP :

In [34]:
```
#PLOTTING A HEATMAP
df_kor = df.corr()
plt.figure(figsize=(10,10))
sns.heatmap(df_kor,vmin=-1,vmax=1,annot=True,linewidth=0.1)
```

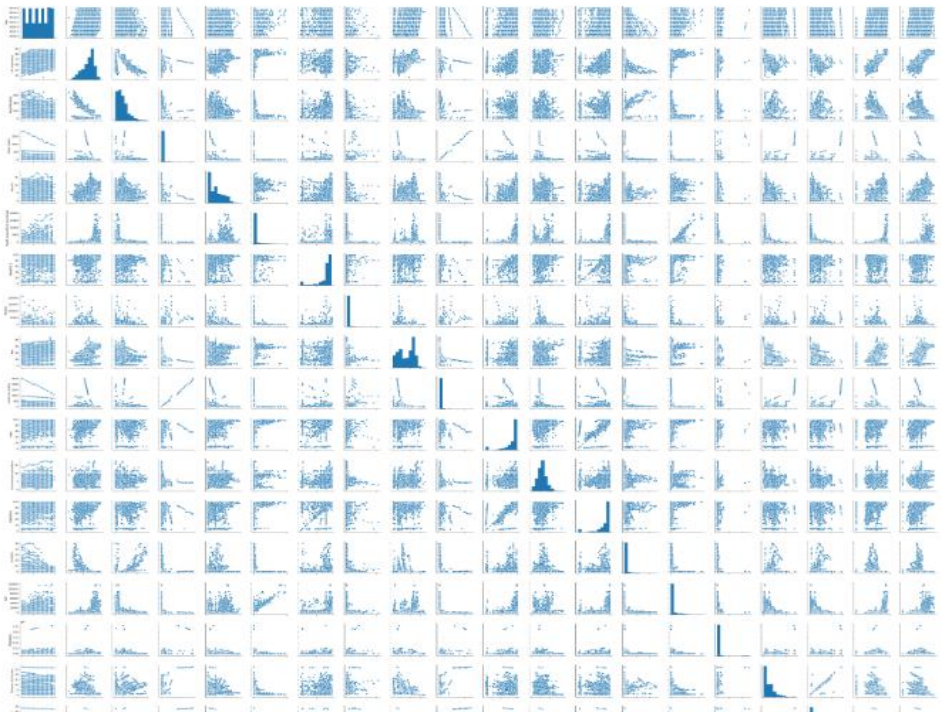
Out[34]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1d786bc198>
```



PLOTTING THE PAIRPLOT:

In [35]:
```
#PLOTTING A PAIRPLOT
sns.pairplot(df)
```

Out[35]:
```
<seaborn.axisgrid.PairGrid at 0x7f1d797dee48>
```



SPLITTING THE DATASET :

In [36]:
```
#SPLITTING THE DATASET
Y = df['Life expectancy ']
X = df[df.columns.difference(['Life expectancy '])]
```

In [37]:
```
#SEE NUMERICAL COLUMNS
df.select_dtypes(include=['int64','float64']).columns
```

Out[37]:
```
Index(['Year', 'Life expectancy ', 'Adult Mortality', 'infant deaths',
       'Alcohol', 'Health expenditure percentage', 'Hepatitis B', 'Measles ',
       'BMI', 'under-five deaths ', 'Polio', 'Government expenditure',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population', 'Thinness 10-19 years',
       'Thinness 5-9 years', 'Income', 'Schooling'],
      dtype='object')
```

In [38]:
```
#SEE CATEGORICAL COLUMNS
df.select_dtypes(include=['object','bool']).columns
```

Out[38]:
```
Index(['Country', 'Status'], dtype='object')
```

In [39]:
```
#IDENTIFY THE CATEGORICAL VALUES FOR COLUMN TRANSFORM
categorical_features = ['Country','Status']
categorical_feature_mask = X.dtypes == object
catrgorical_features = X.columns[categorical_feature_mask].tolist()
#DEFINE CATEGORICAL PIPELINE
categorical_transformer = Pipeline(steps = [('onehot',OneHotEncoder(handle_unknown='ignore')),
                                            ])
```

In [40]:
```
#IDENTIFY THE NUMERICAL VALUES FOR COLUMN TRANSFORM
numeric_features = ['Year','Adult Mortality','infant deaths','Alcohol','percentage expenditure','Hepatitis
B','Measles','BMI','under-five deaths','Polio',
                    'Total expenditure','Diptheria','HIV/AIDS','GDP','Population','thinness 1-19 years','t
hinness 5-9 years','Income composition of resources','Schooling']
numeric_feature_mask= X.dtypes!=object
numeric_features = X.columns[numeric_feature_mask].tolist()
#DEFINE NUMERIC PIPELINE
numeric_transformer = Pipeline(steps=[
    ('imputer',SimpleImputer(strategy='median')),
    ('scaler',StandardScaler())
])
```

PIPELINING USING COLUMN TRANSFORM

In [41]:
```
#PIPELINING USING COLUMN TRANSFORM
preprocessor = ColumnTransformer(
                transformers = [
                    ('num',numeric_transformer,numeric_features),
                    ('cat',categorical_transformer,categorical_features)
                ]

)
```

DEFINE A EXTRA TREE REGRESSOR USING PIPELINE:

In [42]:
```
#DEFINE A REGRESSOR MODEL USING PIPELINING FUNCTION
ExtraTreeRegressor = Pipeline([
    ('preprocessor',preprocessor),
    ('ExtraTreeRegressor',ExtraTreesRegressor(n_estimators=100,random_state=0))
])
```

SPLIT INTO TRAIN & TEST:

In [43]:
```
#TRAIN-TEST SPLIT
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2)
```

In [44]:
```
#FIT THE TRAINING MODEL
reg = ExtraTreeRegressor.fit(X_train,Y_train)
```

In [45]:
```
#PREDICT THE TEST DATA VALUES
test_pred= reg.predict(X_test)
print(test_pred)
```

```
[62.438       61.332       49.623       65.287       72.574       74.546
 82.214       48.628       68.156       65.307       68.969       57.288
 54.507       45.735       82.326       76.199       55.023       73.384
 79.45        79.267       73.477       76.801       63.123       81.574
 69.46049863 74.005       69.92449863 73.349       71.364       52.83
 79.934       47.841       49.239       74.001       82.279       58.396
 52.477       52.121       69.516       69.677       75.973       73.134
 83.74        70.09724932 49.244       72.622       79.167       77.046
 52.41        81.641       60.268       71.846       75.879       70.65749863
 61.846       75.409       54.756       72.584       77.746       82.253
 48.708       65.438       73.969       68.136       57.945       71.248
 67.128       51.714       63.215       68.351       69.845       66.00049863
 74.618       74.373       64.71        81.6         76.14        74.882
```

In [46]:
```
#ESTIMATING ERROR
print('Mean squared Error:',mean_squared_error(Y_test,test_pred))
print('R2 score:',r2_score(Y_test,test_pred)*100)
```

```
Mean squared Error: 3.265002712933994
R2 score: 96.65602722327536
```

In [47]:
```
!pip install watson-machine-learning-client
```

CREDENTIALS:

In [23]:
```
wml_credentials = {
  "apikey": "*****************",
  "instance_id": "******************",
  "url": "https://us-south.ml.cloud.ibm.com"
}
client = WatsonMachineLearningAPIClient(wml_credentials)
print(client.service_instance.get_url())
```

```
https://us-south.ml.cloud.ibm.com
```

In [24]:
```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME:"*******",
               client.repository.ModelMetaNames.AUTHOR_EMAIL:"*****",
               client.repository.ModelMetaNames.NAME:"predicting life expectancy"
              }
#STORE THE MACHINE LEARNING MODEL
model_artifact = client.repository.store_model(ExtraTreeRegressor,meta_props=model_props)
```

DEPLOY THE MODEL:

In [25]:
```
#GET MODEL UID
model_uid = client.repository.get_model_uid(model_artifact)
#DEPLOY THE MODEL
create_deployement = client.deployments.create(model_uid,name="LifeExpectancyPrediction")
```

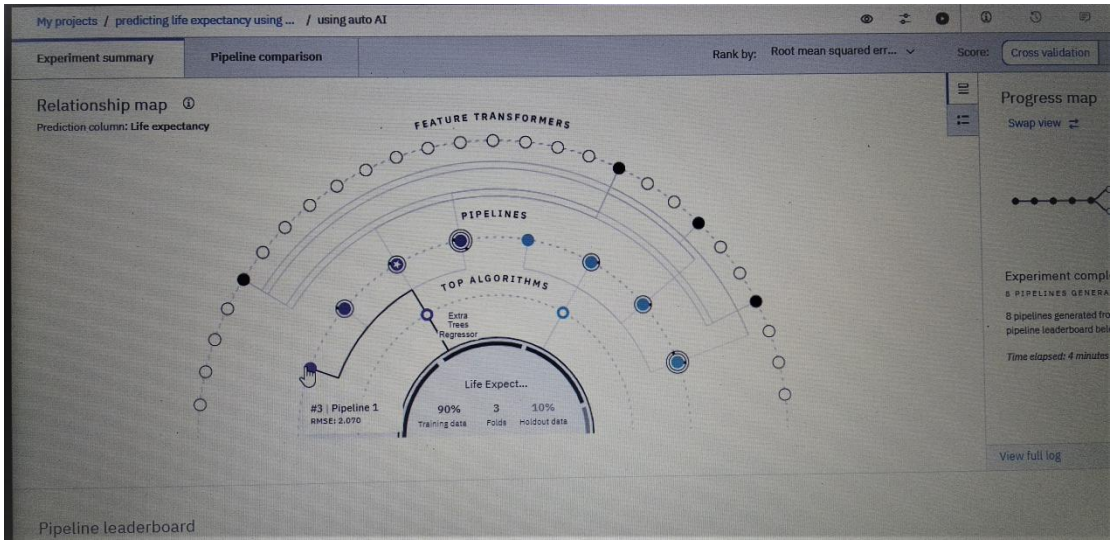
```
#######################################################################################

Synchronous deployment creation for uid: '123022d3-c92f-4ffb-831d-03b90c9f3f6b' started

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


INITIALIZING
DEPLOY_SUCCESS


------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='c8136156-2fdf-4e7c-916c-0cc25c2ffc86'
------------------------------------------------------------------------------------------------
```

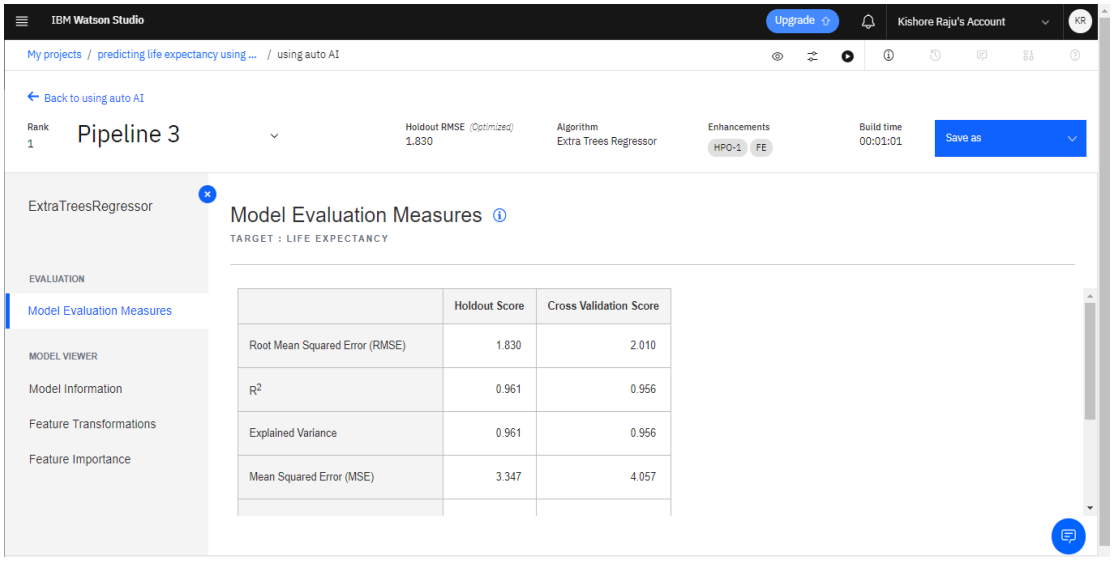
ENDPOINT URL:

In [51]:
```
#GET SCORING END POINT URL
scoring_endpoint = client.deployments.get_scoring_url(create_deployement)
print(scoring_endpoint)
```

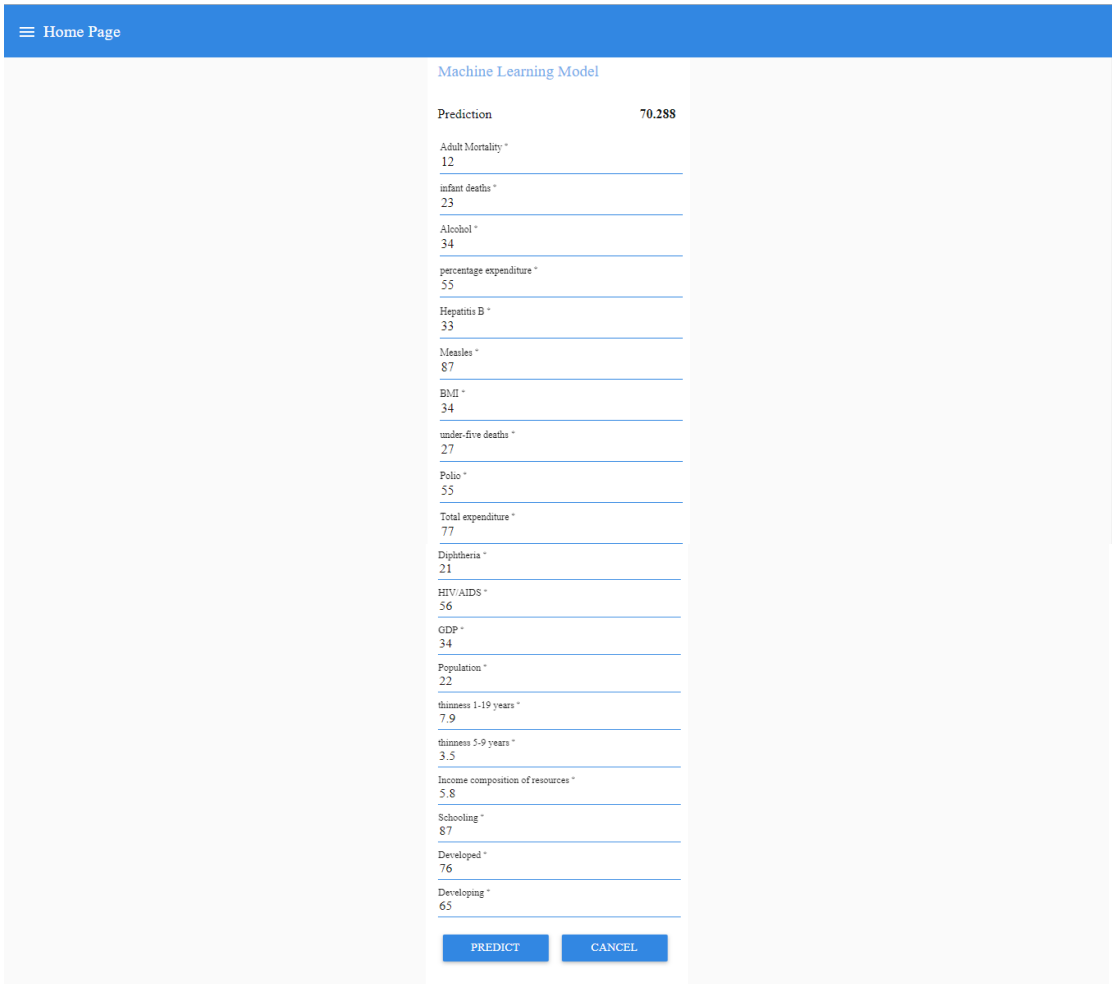
```
https://us-south.ml.cloud.ibm.com/v3/wml_instances/42c7c1be-8e3e-4a87-9cad-79d7998a65fb/deployments/23906e
5d-eeda-4234-a686-1db068f94667/online
```

AUTO AI MODEL:





UI OUTPUT :


Home Page
Machine Learning Model
Prediction
70.288
Adult Mortality *
12
infant deaths *
23
Alcohol *
34
percentage expenditure *
55
Hepatitis B *
33
Measles *
87
BMI *
34
under-five deaths *
27
Polio *
55
Total expenditure *
77
Diphtheria *
21
HIV/AIDS *
56
GDP *
34
Population *
22
thinness 1-19 years *
7.9
thinness 5-9 years *
3.5
Income composition of resources *
5.8
Schooling *
87
Developed *
76
Developing *
65
PREDICT
CANCEL