PROJECT REPORT

Predicting Life Expectancy using Machine learning

By

Dhanush H Amin

NMAM institute of Technology

Email: dhanushamin10@yahoo.com

Under the guidance of

The SmartBridge

Table of contents

Pg.No

1.Introduction

1.1 Overview

We're in an unprecedented era where humans are living longer and longer. It's no secret, though, that life expectancy varies widely across the globe. Life expectancy is one of the most important factors in end-of-life decision making. 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 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. This project tests the potential of using machine learning techniques for predicting life expectancy from previous medical records.

1.2. Purpose

The purpose of this project is that the people from various places can easily predict their life expectancy by providing the inputs asked by the model. This software can be used by all people in the world because the training part of this model contains inputs and predictions of number of countries.

- Inspection of underlying causes of Life Expectancy in various regions.
- To concentrate on factors leading to decline in life expectancy and thereby efficiently come up with a solution to negate the cause.
- Predicting life expectancy would play a vital role in judging the growth and development of the economy.
- Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.
- Helps the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years.

2.Literature Survey

1.1 Existing Problem

In our regular prediction system, there are many problems exist, such as whole concept of life expectancy depends on the interpretation given to “full health”. The factors used to predict the life expectancy of people are based on some associated specific features of particular fields like

- Morbidity and mortality (smoking, alcohol consumption, overweight and obesity, and physical activity)
- Health related disease
- Occupational or social class, area level deprivation, geographical area of residence (urban and rural), housing tenure
- Race-based inequalities.

There have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that effect of immunization and human development index was not taken into account in the past. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries.

The World Health Organization (WHO) began producing annual life tables for all Member States in 1999. These life tables are a basic input to all WHO estimates of global, regional and country-level patterns and trends in all-cause and cause-specific mortality. After the publication of life tables for years to 2009 in the 2011 edition of World Health Statistics, WHO has shifted to a two year cycle for the updating of life tables for all Member States. Even still the model is not really updated in every fields. WHO applies standard methods to the analysis of Member State data to ensure comparability of estimates across all countries. This will inevitably result in differences for some Member States with official estimates for quantities such as life expectancy, where a variety of different projection methods and other methods are used.

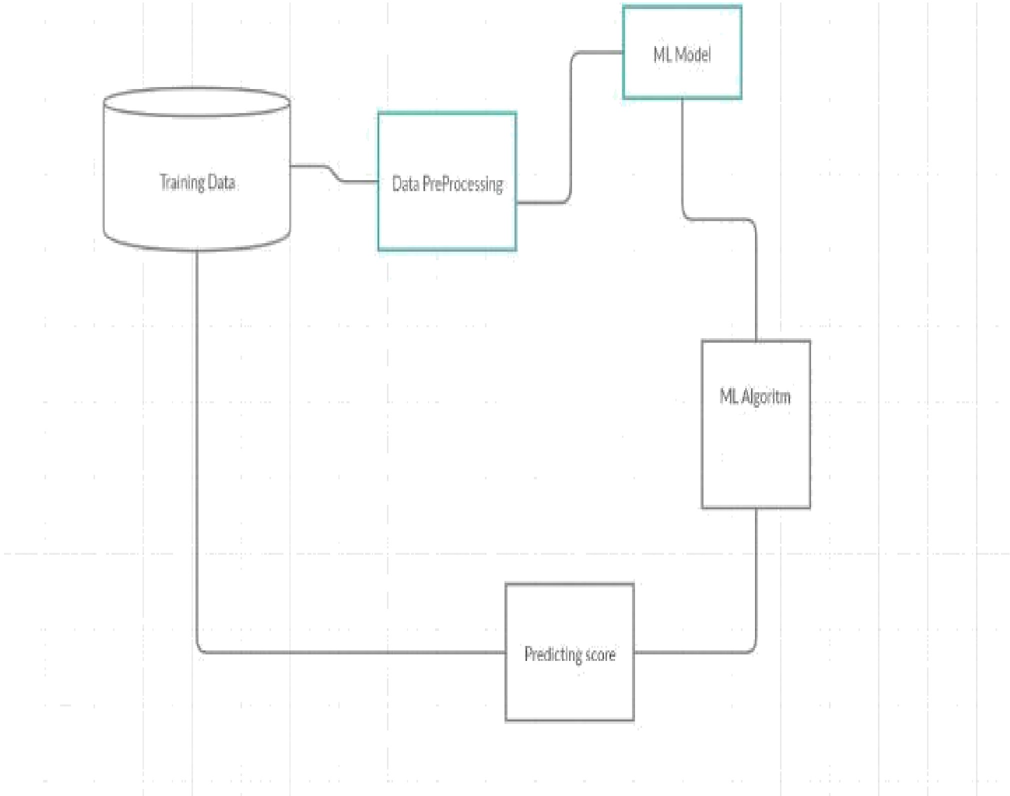
1.2 Proposed Solution

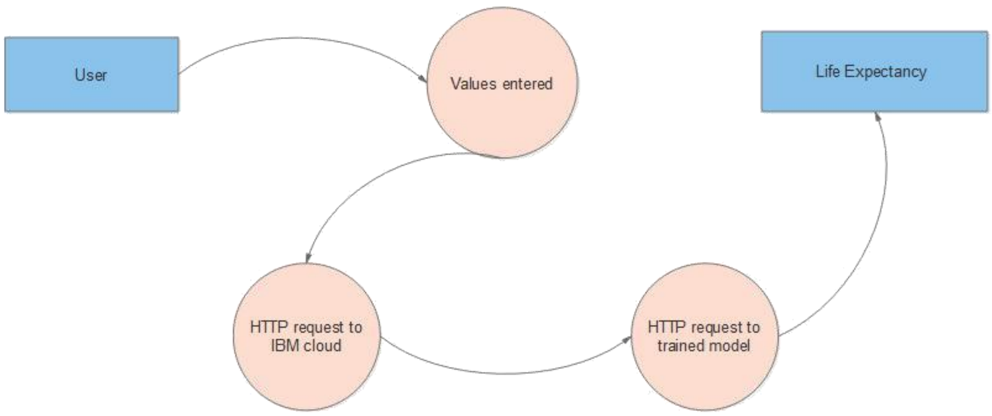
In this project I have proposed a solution to predict the life expectancy using machine learning. Machine Learning is the process of training the computer to think and decide solutions like human. The reason why I have chosen this architecture was only with the help of Machine Learning, deep understanding of the data and an ability to create a model can be done. Design a Regression model to predict life expectancy ratio of a given country based on some features provided such as year, GDP (gross domestic product), education, alcohol intake of people in the country, expenditure on healthcare system and some specific disease related deaths that happened in the country.

For the above proposed solution we have a dataset which consist of various factors. In this system we have taken all the correlated features into consideration. So the output variable would be average life expectancy which would depend on variety of factors of different fields. The project uses immunization factors, mortality factors, economic factors to predict the life expectancy of a country for a given year using a machine learning model.

3. Theoretical Analysis

3.1 Block Diagram





3.2 Hardware and Software designing

1. Create required IBM Cloud services
2. Build a Watson studio project
3. Configure Watson Studio
4. Create IBM Machine Learning instance
5. Create machine learning model in Watson studio
6. Deploy the machine learning model and get the endpoint
7. Create Node red Flow
8. Integrate node red with machine learning model
9. Deploy and run the application.

4. Experimental Investigations

Dataset

The dataset used is a life expectancy dataset released by the World Health Organization. The data set has the following features.

The data is saved as a csv file as LifeExpectancy.csv and it is read and stored in the life data variable. The Year column is dropped as it will not be used in the analysis. The first 5 rows are shown below. The data contains 21 columns and 2938 rows with the header row. The table contains data about

- Countries
- Status
- Life Expectancy
- Adult Mortality
- 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

Pre-processing and cleaning the datasets

- Before the data can be imported using the machine learning libraries and can be trained, the data needs to be cleaned and pre-processed.

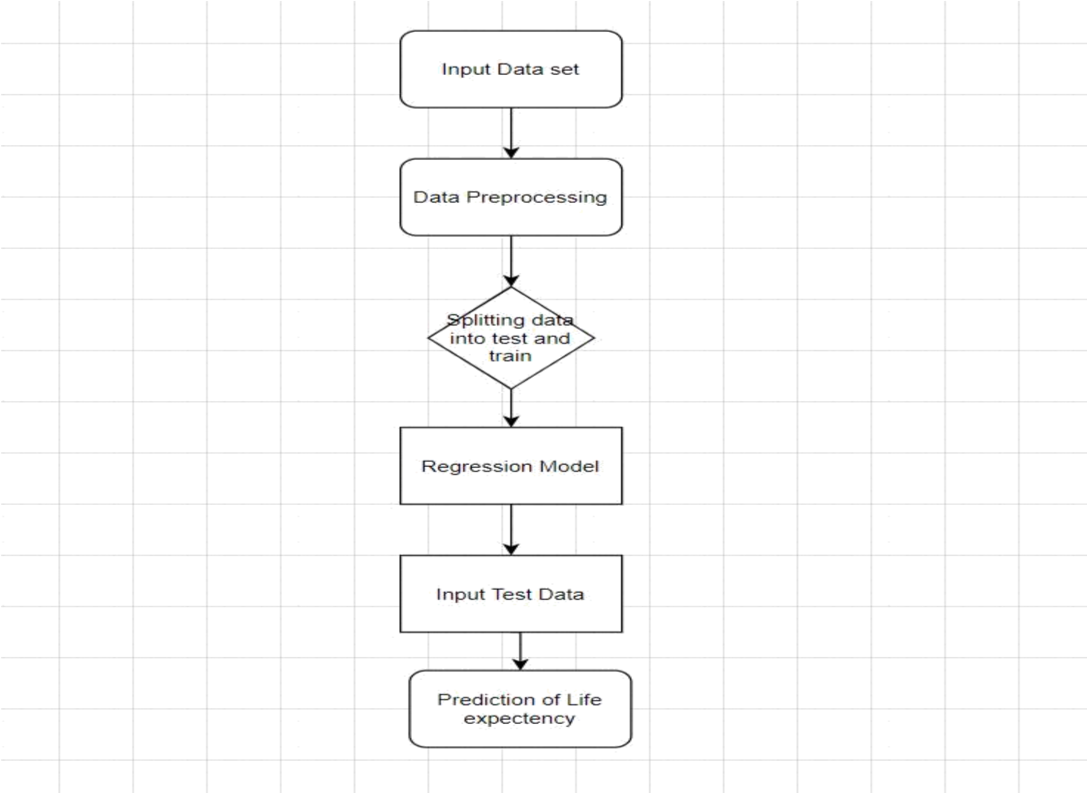
- All the null values in the data set need to set equal to the mean value.

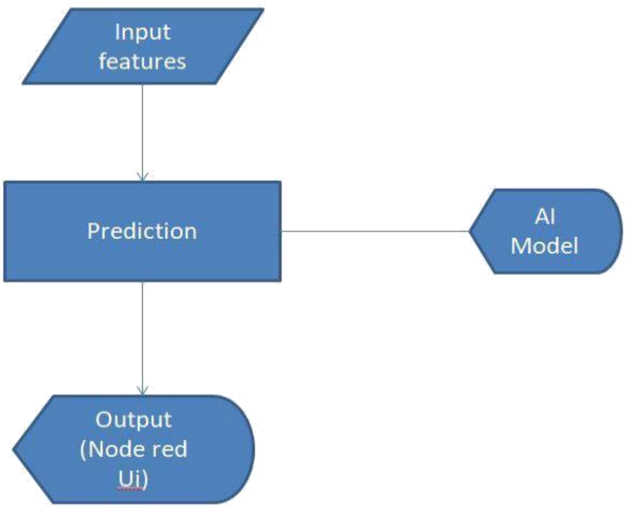
- In the cleaning process, I have set the null values as 0 for the ease of calculation and maintaining the accuracy of the model.

Finding the most suitable algorithm

- After experimenting with many Regression models, I found that Random forest regression gives the highest accuracy and lowest mean squared error value

5. Flowchart


Input Data set
Data Preprocessing
Splitting data into test and train
Regression Model
Input Test Data
Prediction of Life expectency

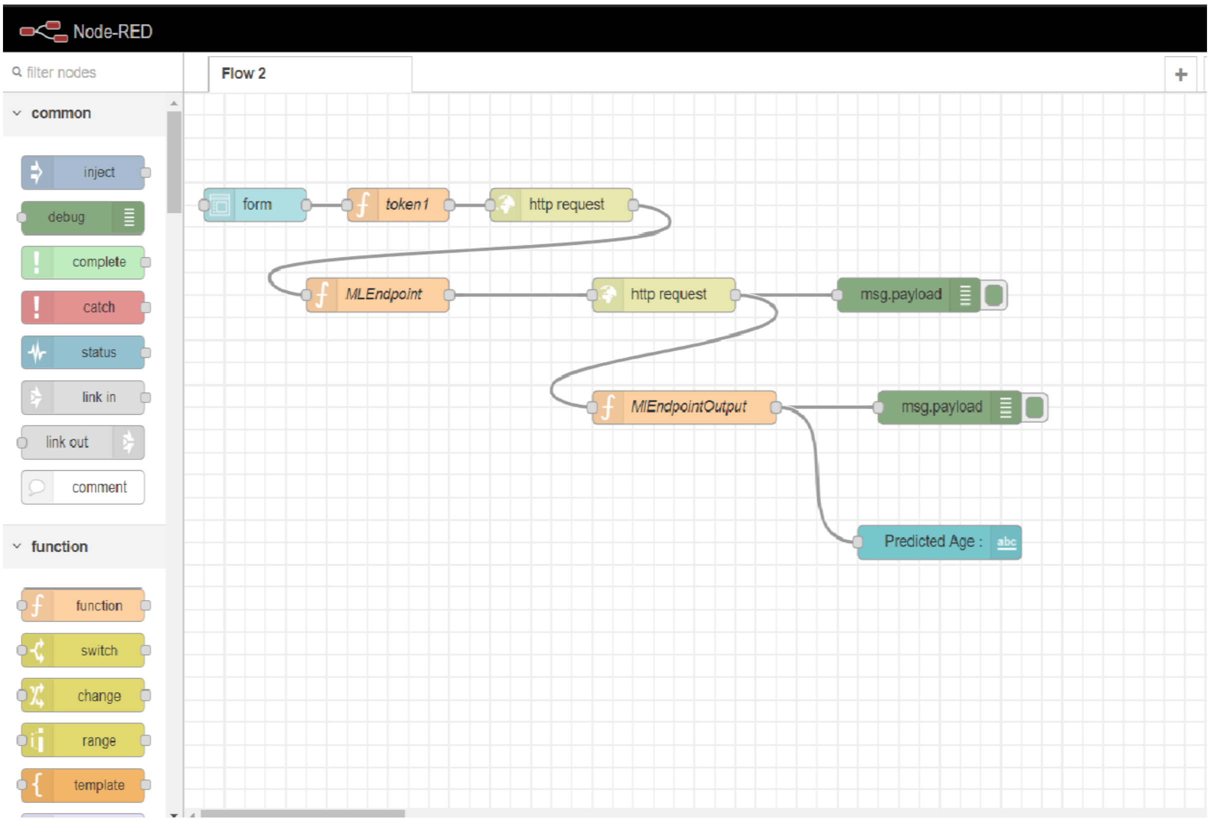

Input features
Prediction
AI Model
Output (Node red Ui)

Node red

To develop a User-Interface for our model we use node red service of IBM cloud.
Components of the flow are:

- Form: The form contains all the elements of the UI. All the labels are associated with a variable.
- Http requests: To setup the flow, we need two http requests. The first http request requires a token to connect to the machine learning service of the Watson studio
- The second http request helps us in integrating the model using the endpoint URL.
- Once the flow has been setup, we deploy the model.

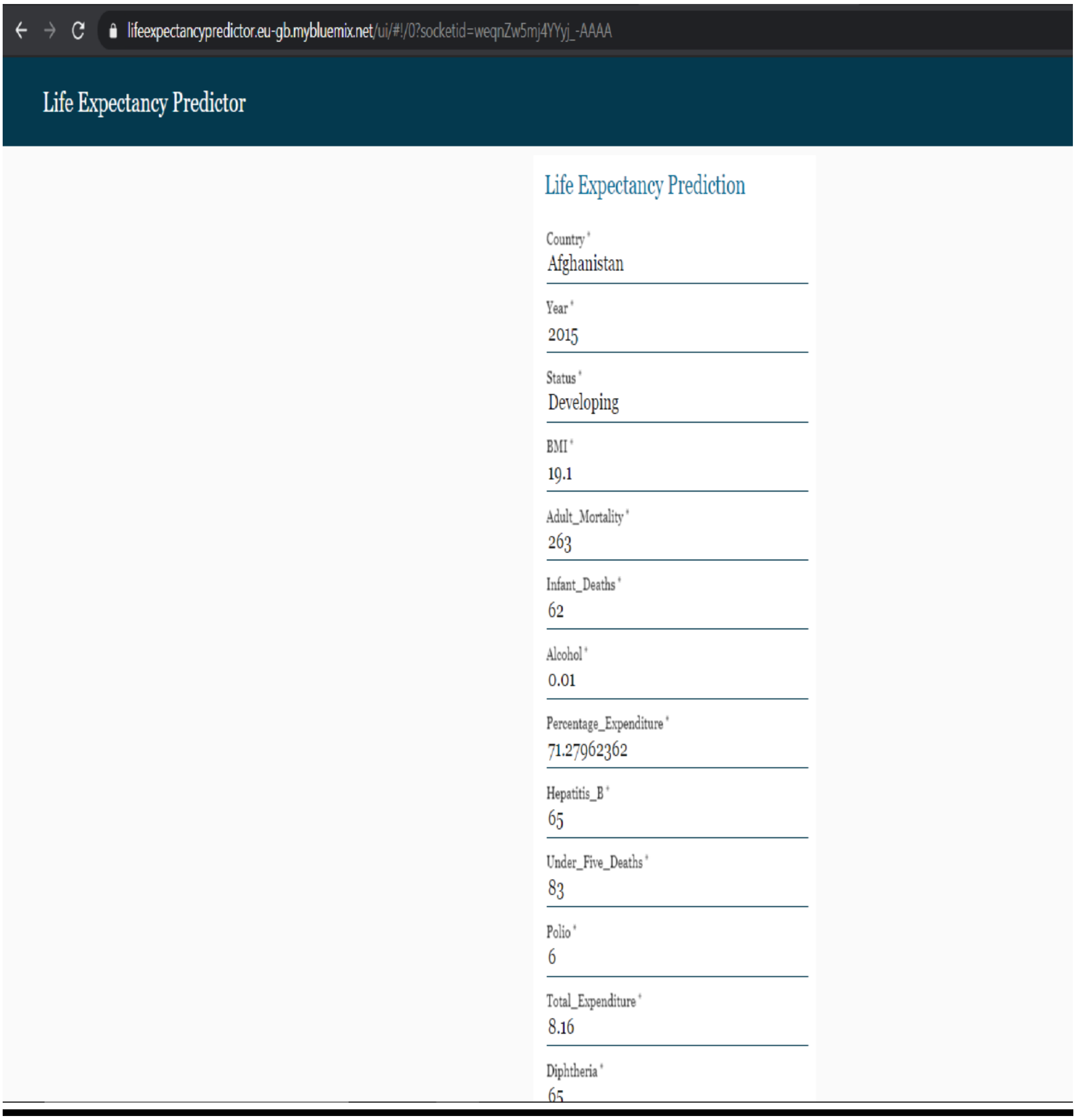
Node Red Flow

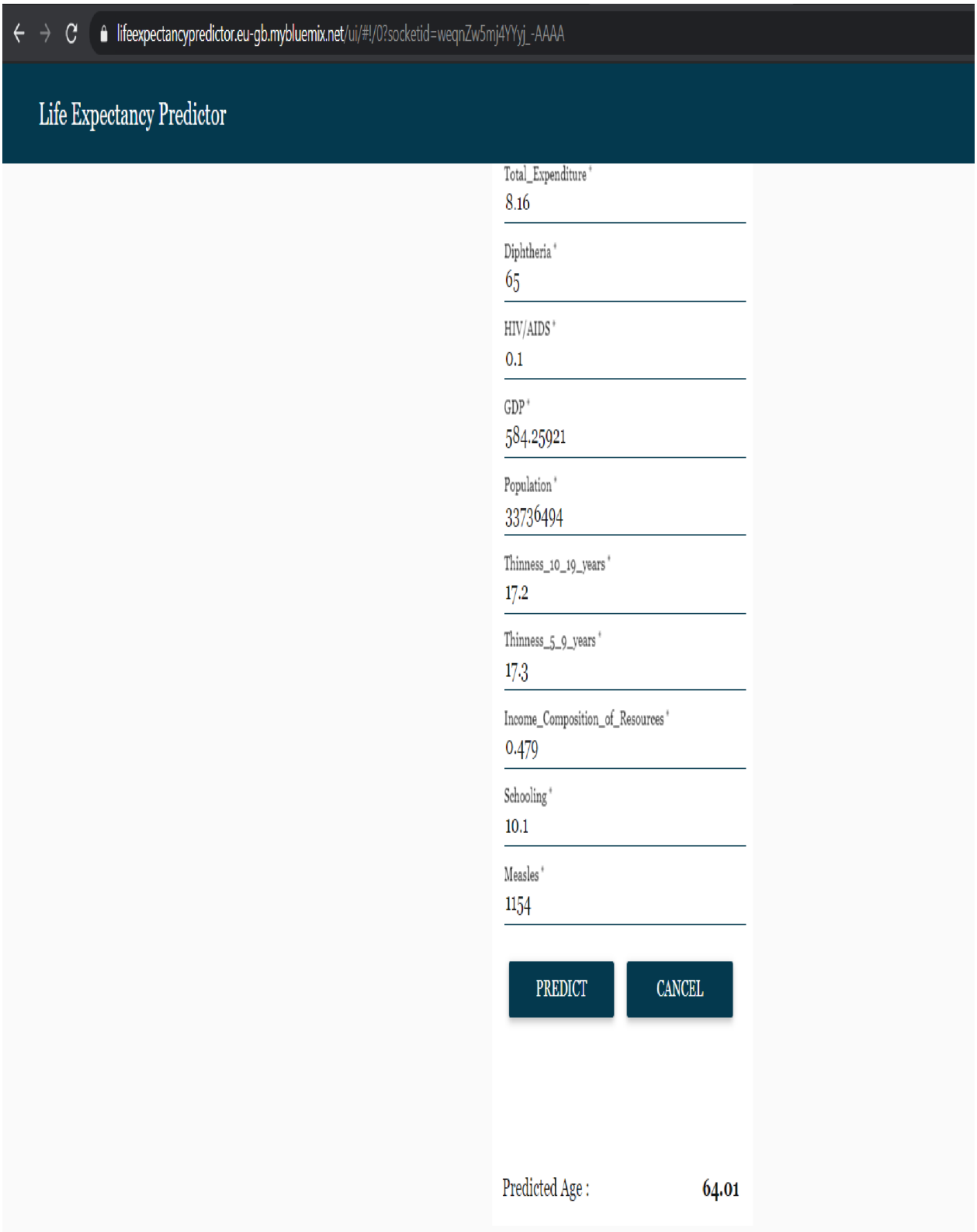


6. Result

Web based UI was developed by integrating all the services using NODE-RED.

URL: https://lifeexpectancypredictor.eu-gb.mybluemix.net/ui/




lifeexpectancypredictor.eu-gb.mybluemix.net/ui/#!/0?socketid=weqnZw5mj4YYyj_-AAAA
Life Expectancy Predictor
Total_Expenditure *
8.16
Diphtheria *
65
HIV/AIDS *
0.1
GDP *
584.25921
Population *
33736494
Thinness_10_19_years *
17.2
Thinness_5_9_years *
17.3
Income_Composition_of_Resources *
0.479
Schooling *
10.1
Measles *
1154
PREDICT
CANCEL
Predicted Age :
64.01

7. Advantages & 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.
- Since the observations this 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.
- The Machine learning algorithm used is Random forest regression which is based on ensemble learning technique. It creates as many trees on subset of data and combines the output of all the trees. In this way it reduces overfitting and improves accuracy
- Random Forest algorithm is very stable. Even if a new data point is introduced in the dataset, the overall algorithm is not affected much since the new data may impact one tree, but it is very hard for it to impact all the trees.
- 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

- 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.
- As the model is deployed in cloud it requires active internet connection to use the application
- The UI needs to make a POST request to the machine learning service and also another POST request to the deployed model hence the application can be little slow.

8. Application

- The project can be used as a basis to develop personalized health applications.
- Depending on the factors used to calculate life expectancy of an individual and the outcome, health care will be able to fund and provide better services to those with greater need
- Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.
- It could help the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years.

9. Conclusion

The end product is a web app created and deployed on node-red app of IBM cloud. The backend of webpage is a linear regression model created and deployed on Watson Studio using machine learning service.
The potential use of project is not limited to health care in practice, but could also be useful in other clinical applications such as clinical trials. The project makes a good use of machine learning in predicting life expectancy of a country that can help respective government in making policies that will serve for the benefit of the nation and entire humankind.

10.Future Scope

- The scalability and flexibility of the application can also be improved with advancement in technology and availability of new and improved resources. Also, with the growth in Artificial Neural networks and Deep learning, one can integrate that with our existing application. With the help of Convolutional Neural networks and Computer vision, we can also try to take into account the physical health and appearance of a person.
- Make the Application support other platforms like iOS, android etc
- Increase the dataset size with continuing UN and Global Data to incorporate new added features like population, GDP, environmental, and etc in order to test and clarify country groupings.

11 Bibliography

- Node-RED Starter Application https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-applicat ion/
- Watson Studio Cloud :https://bookdown.org/caoying4work /watsonstudio-workshop/jn.html
- Dataset Reference https://www.kaggle.com/kumaraja rshi/life-expectancy-who
- IBM Cloud Services : https://www.youtube.com/watch?v=DBRGlAHdj48&l ist=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L
- Import the Dataset into Jupyter Notebook : https://www.youtube.com/watch?v=Jtej3Y6uUng

12 Appendix

LIFE EXPECTANCY PREDICTION

Source Code

Importing Dataset to IBM watson Studio

```
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_e08f5880c4f7463fba124cab9de66159 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='',
    ibm_auth_endpoint="",
    config=Config(signature_version='oauth'),
    endpoint_url='')

body = client_e08f5880c4f7463fba124cab9de66159.get_object(Bucket='lifeexpectancyrandomforest-donotdelet
e-pr-zc9us8qxthugdk',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 state
ment.
data = pd.read_csv(body)
data.head()
```

#DATA PREPROCESSING

```
data.info()
data.isnull().sum()
country_list = data.Country.unique()
len(country_list)
country_list = data.Country.unique()
fill_list = ['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']
```

```
for country in country_list:
    data.loc[data['Country'] == country,fill_list] = data.loc[data['Country'] == country,fill_list].interpolate()
data.dropna(inplace=True)
data.isna().sum()
```

RENAMING COLUMNS

```
data.rename(columns={" BMI ":"BMI",'Life expectancy ':'Life expectancy',
                "under-five deaths ":"under-five deaths","Measles ":"Measles","Diphtheria ":"Diphtheria",
                ' HIV/AIDS':"HIV/AIDS",
                " thinness  1-19 years":"thinness 10-19 years"," thinness 5-9 years":"thinness 5-9 years"},inplace=Tr
ue)
col_dict = {'Life expectancy':1 , 'Adult Mortality':2 ,
        'Alcohol':3 , 'percentage expenditure': 4, 'Hepatitis B': 5,
       'Measles' : 6, 'BMI': 7, 'under-five deaths' : 8, 'Polio' : 9, 'Total expenditure' :10,
       'Diphtheria':11, 'HIV/AIDS':12, 'GDP':13, 'Population' :14,
       'thinness 10-19 years' :15, 'thinness 5-9 years' :16,
       'Income composition of resources' : 17, 'Schooling' :18, 'infant deaths':19}
from scipy.stats.mstats import winsorize
New_Life_Expectancy = winsorize(data['Life expectancy'],(0.01,0))
New_Adult_Mortality = winsorize(data['Adult Mortality'],(0,0.03))
New_Infant_Deaths = winsorize(data['infant deaths'],(0,0.10))
New_Alcohol = winsorize(data['Alcohol'],(0,0.01))
New_Percentage_Exp = winsorize(data['percentage expenditure'],(0,0.12))
New_HepatitisB = winsorize(data['Hepatitis B'],(0.11,0))
New_Measles = winsorize(data['Measles'],(0,0.19))
New_Under_Five_Deaths = winsorize(data['under-five deaths'],(0,0.12))
New_Polio = winsorize(data['Polio'],(0.09,0))
New_Tot_Exp = winsorize(data['Total expenditure'],(0,0.01))
New_Diphtheria = winsorize(data['Diphtheria'],(0.10,0))
New_HIV = winsorize(data['HIV/AIDS'],(0,0.16))
New_GDP = winsorize(data['GDP'],(0,0.13))
New_Population = winsorize(data['Population'],(0,0.14))
New_thinness_10_19_years = winsorize(data['thinness 10-19 years'],(0,0.04))
New_thinness_5_9_years = winsorize(data['thinness 5-9 years'],(0,0.04))
New_Schooling = winsorize(data['Schooling'],(0.02,0.01))
New_list = [New_Life_Expectancy,New_Adult_Mortality,New_Alcohol,New_Measles,New_Infant_Deaths,
           New_Percentage_Exp,New_HepatitisB,New_Under_Five_Deaths,New_Polio,New_Tot_Exp,New_Diph
theria,
           New_HIV,winsorized_GDP,New_Population,New_thinness_10_19_years,New_thinness_5_9_years,
          New_Income_Comp_Of_Resources,New_Schooling]
data['New_Life_Expectancy'] = winsorized_Life_Expectancy
data['New_Adult_Mortality'] = winsorized_Adult_Mortality
data['New_Infant_Deaths'] = winsorized_Infant_Deaths
data['New_Alcohol'] = winsorized_Alcohol
data['New_Percentage_Exp'] = winsorized_Percentage_Exp
data['New_HepatitisB'] = winsorized_HepatitisB
data['New_Under_Five_Deaths'] = winsorized_Under_Five_Deaths
```

```
data['New_Polio'] = winsorized_Polio
data['New_Tot_Exp'] = winsorized_Tot_Exp
data['New_Diphtheria'] = winsorized_Diphtheria
data['New_HIV'] = winsorized_HIV
data['New_GDP'] = winsorized_GDP
data['New_Population'] = winsorized_Population
data['New_thinness_10_19_years'] = winsorized_thinness_10_19_years
data['New_thinness_5_9_years'] = winsorized_thinness_5_9_years
data['New_Income_Comp_Of_Resources'] = winsorized_Income_Comp_Of_Resources
data['New_Schooling'] = winsorized_Schooling
data['New_Measles'] = winsorized_Measles
new_data=pd.DataFrame(data=data,columns=['Country', 'Year', 'Status',
        'BMI', 'New_Adult_Mortality',
       'New_Infant_Deaths', 'New_Alcohol',
       'New_Percentage_Exp', 'New_HepatitisB',
       'New_Under_Five_Deaths', 'New_Polio',
       'New_Tot_Exp', 'New_Diphtheria', 'New_HIV',
       'New_GDP', 'New_Population',
       'New_thinness_10_19_years', 'New_thinness_5_9_years',
       'New_Income_Comp_Of_Resources', 'New_Schooling',
       'New_Measles',
       'New_Life_Expectancy'])
new_data.rename(columns={
            'New_Adult_Mortality':'Adult_Mortality',
       'New_Infant_Deaths' :'Infant_Deaths',
       'New_Alcohol':'Alcohol',
       'New_Percentage_Exp':'Percentage_Expenditure',
       'New_HepatitisB':'Hepatitis_B',
       'New_Under_Five_Deaths':'Under_Five_Deaths',
       'New_Polio':'Polio',
       'New_Tot_Exp':'Total_Expenditure',
       'New_Diphtheria':'Diphtheria',
       'New_HIV':'HIV/AIDS',
       'New_GDP':'GDP',
       'New_Population':'Population',
       'New_thinness_10_19_years':'Thinness_10_19_years',
       'New_thinness_5_9_years':'Thinness_5_9_years',
       'New_Income_Comp_Of_Resources':'Income_Composition_of_Resources',
       'New_Schooling':'Schooling',
       'New_Measles':'Measles',
       'New_Life_Expectancy':'Life_Expectancy' } ,inplace=True)
X = new_data.drop('Life_Expectancy', axis=1)
Y = pd.DataFrame(data=new_data,columns=['Life_Expectancy'])
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer, make_column_transformer
from sklearn.pipeline import make_pipeline
```

```
from sklearn.impute import SimpleImputer
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from collections import OrderedDict
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score,mean_squared_error
#Training and testing Data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
numeric_features = ['Year', 'BMI',
       'Adult_Mortality', 'Infant_Deaths', 'Alcohol', 'Percentage_Expenditure',
       'Hepatitis_B', 'Under_Five_Deaths', 'Polio', 'Total_Expenditure',
       'Diphtheria', 'HIV/AIDS', 'GDP', 'Population', 'Thinness_10_19_years',
       'Thinness_5_9_years', 'Income_Composition_of_Resources', 'Schooling',
       'Measles']
categorical_features = ['Country', 'Status']
#Encoding Categorical Data
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder

categorical_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(handle_unknown='ignore')),
])
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler

numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median'))

])
from sklearn.compose import ColumnTransformer

preprocessor = ColumnTransformer(
    transformers=[
        ('cat', categorical_transformer, categorical_features),
        ('num', numeric_transformer, numeric_features)
    ]
)
RFmodel = Pipeline([
     ('preprocessor', preprocessor),
     ('RFmodel', RandomForestRegressor())
])
RFmodel.fit(X_train,Y_train)
predict= RFmodel.predict(X_test)
r2_score(predict, Y_test)
!pip install watson-machine-learning-client
```

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient
wml_credentials={
 "apikey":################,
  "iam_apikey_description": #######################,
  "iam_apikey_name":  #######################,
  "iam_role_crn":  #######################,
  "iam_serviceid_crn":  #######################,
  "instance_id":  #######################,
  "url": #######################
}
client = WatsonMachineLearningAPIClient( wml_credentials )
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Dhanush Amin",
             client.repository.ModelMetaNames.AUTHOR_EMAIL: "dhanushamin10@gmail.com",
             client.repository.ModelMetaNames.NAME: "LifeExpectancy"}
model_artifact =client.repository.store_model(RFmodel, meta_props=model_props)
published_model_uid = client.repository.get_model_uid(model_artifact)
deployment = client.deployments.create(published_model_uid, name="life_expectancy")
scoring_endpoint = client.deployments.get_scoring_url(deployment)
```

```
Scoring ENDPOINT
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/8aa1bd59-fd42-450b-921
5-86d97ac93f45/deployments/e097cf54-e144-48e0-b6bc-50320bb7fa52/online'
```

JSON data for testing after deployment:

{"fields":["Country", "Year", "Status",

"BMI", "Adult_Mortality", "Infant_Deaths", "Alcohol", "Percentage_Expenditure", "Hepatitis_B", "Under_Five_Deaths", "Polio", "Total_Expenditure", "Diphtheria", "HIV/AIDS", "GDP","Population", "Thinness_10_19_years", "Thinness_5_9_years", "Income_Composition_of_Resources", "Schooling", "Measles"], "values":[["Zimbabwe",2000, "Developing", 25.5,491.0,24,1.68,0.0,79.0,39,78.0,

7.10,78.0,3.2,547.358879,12222251.0, 11.0,11.2,0.434,9.8,1154]]

DATASET

https://www.kaggle.com/kumarajarshi/life-expectancy-who?rvi=1

NODE RED FLOW JSON EXPORT

Flows.json


```
[
   {
      "id": "ff6e6506.0d3608",
      "type": "tab",
      "label": "Flow 2",
      "disabled": false,
      "info": ""
   },
   {
      "id": "6d0bc02b.ef338",
      "type": "ui_form",
      "z": "ff6e6506.0d3608",
      "name": "",
      "label": "",
      "group": "d26bfaff.247118",
      "order": 1,
      "width": 0,
      "height": 0,
      "options": [
         {
            "label": "Country",
            "value": "a",
            "type": "text",
            "required": true,
            "rows": null
         },
         {
```

```
        "label": "Year",
        "value": "b",
        "type": "number",
        "required": true,
        "rows": null
    },
    {
        "label": "Status",
        "value": "c",
        "type": "text",
        "required": true,
        "rows": null
    },
    {
        "label": "BMI",
        "value": "d",
        "type": "number",
        "required": true,
        "rows": null
    },
    {
        "label": "Adult_Mortality",
        "value": "e",
        "type": "number",
        "required": true,
        "rows": null
    },
    {
        "label": "Infant_Deaths",
        "value": "f",
        "type": "number",
        "required": true,
```

```
    "rows": null
},
{
    "label": "Alcohol",
    "value": "g",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Percentage_Expenditure",
    "value": "h",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Hepatitis_B",
    "value": "i",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Under_Five_Deaths",
    "value": "j",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Polio",
```

"value": "k",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Total_Expenditure",
"value": "l",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Diphtheria",
"value": "m",
"type": "number",
"required": true,
"rows": null
},
{
"label": "HIV/AIDS",
"value": "n",
"type": "number",
"required": true,
"rows": null
},
{
"label": "GDP",
"value": "o",
"type": "number",
"required": true,
"rows": null

```
},
{
    "label": "Population",
    "value": "p",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Thinness_10_19_years",
    "value": "q",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Thinness_5_9_years",
    "value": "r",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Income_Composition_of_Resources",
    "value": "s",
    "type": "number",
    "required": true,
    "rows": null
},
{
    "label": "Schooling",
    "value": "t",
```

"type": "number",
"required": true,
"rows": null
},
{
"label": "Measles",
"value": "u",
"type": "number",
"required": true,
"rows": null
}
],
"formValue": {
"a": "",
"b": "",
"c": "",
"d": "",
"e": "",
"f": "",
"g": "",
"h": "",
"i": "",
"j": "",
"k": "",
"l": "",
"m": "",
"n": "",
"o": "",
"p": "",
"q": "",
"r": "",
"s": "",

"t": "",

"u": ""

},

"payload": "",

"submit": "Predict",

"cancel": "cancel",

"topic": "",

"x": 70,

"y": 100,

"wires": [

[

"f0e37a6e.5c9de8"

]

]

},

{

"id": "f0e37a6e.5c9de8",

"type": "function",

"z": "ff6e6506.0d3608",

"name": "token1",

"func": "//make user given values as global variables\nglobal.set(\"a\",msg.payload.a);\nglobal.set(\"b\",msg.payload.b);\nglobal.set(\"c\",msg.payload.c);\nglobal.set(\"d\",msg.payload.d);\nglobal.set(\"e\",msg.payload.e);\nglobal.set(\"f\",msg.payload.f);\nglobal.set(\"g\",msg.payload.g);\nglobal.set(\"h\",msg.payload.h);\nglobal.set(\"i\",msg.payload.i);\nglobal.set(\"j\",msg.payload.j);\nglobal.set(\"k\",msg.payload.k);\nglobal.set(\"l\",msg.payload.l);\nglobal.set(\"m\",msg.payload.m);\nglobal.set(\"n\",msg.payload.n);\nglobal.set(\"o\",msg.payload.o);\nglobal.set(\"p\",msg.payload.p);\nglobal.set(\"q\",msg.payload.q);\nglobal.set(\"r\",msg.payload.r);\nglobal.set(\"s\",msg.payload.s);\nglobal.set(\"t\",msg.payload.t);\nglobal.set(\"u\",msg.payload.u);\n\n//following are required to receive a token\nvar apikey=\"grAzNrbgQ-0EGhONUeI6vYVgVeaIGJ1nVq7SHKLpWD-2\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"};\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey};\nreturn msg;\n",

"outputs": 1,

"noerr": 0,

"x": 210,

"y": 100,
"wires": [
[
"5b6da7d7.d34d78"
]
]
},
{
"id": "5f7d2785.cc28e8",
"type": "http request",
"z": "ff6e6506.0d3608",
"name": "",
"method": "POST",
"ret": "obj",
"paytoqs": false,
"url": "https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/8aa1bd59-fd42-450b-9215-86d97ac93f45/deployments/e097cf54-e144-48e0-b6bc-50320bb7fa52/online",
"tls": "",
"persist": false,
"proxy": "",
"authType": "basic",
"x": 470,
"y": 180,
"wires": [
[
"c7c3e6ec.f360c8",
"d5b29ddd.f07a5"
]
]
},
{
"id": "642cb198.014a9",
"type": "debug",

```
    "z": "ff6e6506.0d3608",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "false",
    "x": 750,
    "y": 280,
    "wires": []
},
{
    "id": "d5b29ddd.f07a5",
    "type": "function",
    "z": "ff6e6506.0d3608",
    "name": "MlEndpointOutput",
    "func": "msg.payload=msg.payload.values[0][0].toFixed(2);\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 490,
    "y": 280,
    "wires": [
        [
            "642cb198.014a9",
            "53c562ff.cf6a4c"
        ]
    ]
},
{
    "id": "c7c3e6ec.f360c8",
    "type": "debug",
    "z": "ff6e6506.0d3608",
```

"name": "",
"active": true,
"tosidebar": true,
"console": false,
"tostatus": false,
"complete": "payload",
"targetType": "msg",
"x": 710,
"y": 180,
"wires": []
},
{
"id": "773944cb.19349c",
"type": "function",
"z": "ff6e6506.0d3608",
"name": "MLEndpoint",
"func": "//get token and make headers\nvar token=msg.payload.access_token;\nvar instance_id=\"8aa1bd59-fd42-450b-9215-86d97ac93f45\";\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance_id}\n\n//get variables that are set earlier\nvar a = global.get(\"a\");\nvar b = global.get(\"b\");\nvar c = global.get(\"c\");\nvar d = global.get(\"d\");\nvar e = global.get(\"e\");\nvar f = global.get(\"f\");\nvar g = global.get(\"g\");\nvar h = global.get(\"h\");\nvar i = global.get(\"i\");\nvar j = global.get(\"j\");\nvar k = global.get(\"k\");\nvar l = global.get(\"l\");\nvar m = global.get(\"m\");\nvar n = global.get(\"n\");\nvar o = global.get(\"o\");\nvar p = global.get(\"p\");\nvar q = global.get(\"q\");\nvar r = global.get(\"r\");\nvar s = global.get(\"s\");\nvar t = global.get(\"t\");\nvar u = global.get(\"u\");\n\n//send the user values to service endpoint\nmsg.payload = \n{\"fields\":[\"Country\", \"Year\", \"Status\", \n\"BMI\", \"Adult_Mortality\", \"Infant_Deaths\", \"Alcohol\", \"Percentage_Expenditure\", \"Hepatitis_B\", \"Under_Five_Deaths\", \"Polio\", \"Total_Expenditure\", \"Diphtheria\", \"HIV/AIDS\", \"GDP\",\"Population\", \"Thinness_10_19_years\", \"Thinness_5_9_years\",\n \"Income_Composition_of_Resources\", \"Schooling\", \"Measles\"],\n\"values\":[[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u]]};\n\nreturn msg;\n",
"outputs": 1,
"noerr": 0,
"x": 190,
"y": 180,

```
        "wires": [
            [
                "5f7d2785.cc28e8"
            ]
        ]
    },
    {
        "id": "5b6da7d7.d34d78",
        "type": "http request",
        "z": "ff6e6506.0d3608",
        "name": "",
        "method": "POST",
        "ret": "obj",
        "paytoqs": false,
        "url": "https://iam.cloud.ibm.com/identity/token",
        "tls": "",
        "persist": false,
        "proxy": "",
        "authType": "basic",
        "x": 370,
        "y": 100,
        "wires": [
            [
                "773944cb.19349c"
            ]
        ]
    },
    {
        "id": "53c562ff.cf6a4c",
        "type": "ui_text",
        "z": "ff6e6506.0d3608",
        "group": "d26bfaff.247118",
```

```
        "order": 2,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "Predicted Age :",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 740,
        "y": 400,
        "wires": []
    },
    {
        "id": "d26bfaff.247118",
        "type": "ui_group",
        "z": "",
        "name": "Life Expectancy Prediction",
        "tab": "aace190f.ec5268",
        "order": 1,
        "disp": true,
        "width": "6",
        "collapse": false
    },
    {
        "id": "aace190f.ec5268",
        "type": "ui_tab",
        "z": "",
        "name": "Life Expectancy Predictor",
        "icon": "dashboard",
        "disabled": false,
        "hidden": false
    }
]
```