Project Report

on

Predicting Life Expectancy using

Machine Learning

by

Md Mobbasher Ansari

(mobby0022@gmail.com)

1. Introduction

Since ancient times, there are a lot of change in the behaviours and cultures of people in different places. According to their way of living, the health care and life expectancy of people varies among each other.

1.1. Overview

Life expectancy is a statistical measure of the average time a human being is expected to live. A typical Regression Machine Learning project leverages historical data to predict insights into the future. This problem statement is aimed at predicting Life Expectancy rate of a country given various features. This problem statement 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 in a dataset.

In order to predict life expectancy rate of a given country, we will be using Machine Learning algorithms to draw inferences from the given dataset and give an output. For better usability by the customer, we are also going to be creating a UI for the user to interact with using Node-Red.

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 more number of countries.

Economic growth:

Predicting life expectancy would play a vital role in judging the growth and development of the economy.

Across countries, high life expectancy is associated with high income per capita. Increase in life expectancy also leads to an increase in the “manpower” of a country. The knowledge asset of a country increases with the number of individuals in a country.

Population Growth:

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.

Personal growth:

This project would also help an individual assess his/her lifestyle choices and alter them accordingly to lead a longer and healthier life. It would make them more aware of their general health and its improvement or deterioration over time.

Growth in Health Sector:

Based 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 Companies:

Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

2. Literature Survey

There are so many organizations that are making research in the prediction of life expectancy. Many research papers dealing with the creation of this model under many algorithms such as Machine Learning, Deep learning and programming languages such as Python and Java script.

2.1. Existing Problem

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

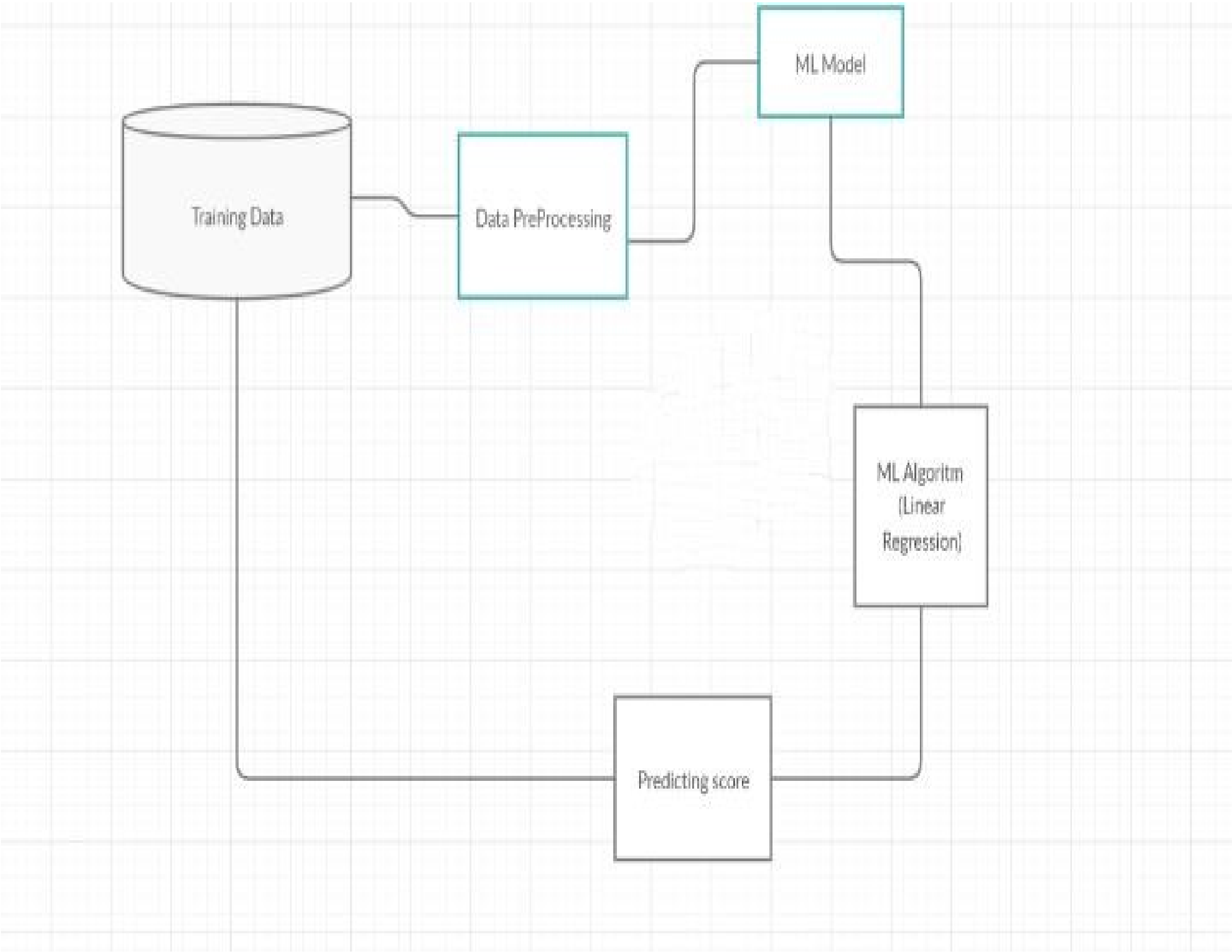
a two year cycle for the updating of life tables for all Member States. Even still the model is not really updated in every fields. WHO applies standard methods to the analysis of Member State data to ensure comparability of estimates across countries. This will inevitably result in differences for some Member States with official estimates for quantities such as life expectancy, where a variety of different projection methods and other methods are used.

2.2. Proposed Solution

So many people were expecting to use a model of life expectancy prediction. In order to that, many institutions and companies are leading their team to build that model. In my 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.

3. Theoretical Analysis

3.1. Block Diagram



3.2. Hardware / Software Designing

1. PROJECT PLANNING AND KICKOFF:

a. Understanding the project description and analyze the data and attributes in the given dataset.
b. Creating Github account
c. Installing Slack and create account with the mail id
d. Learning to use Zoho writer.

2. EXPLORE IBM CLOUD PLATFORM:

a. Creating IBM cloud account with the mail id
b. Creating IBM academic initiative account with the mail id
c. Create a Node-Red starter application.

3. EXPLORE IBM WATSON SERVICES:

a. Exploring IBM Watson use cases.
b. Learning about IBM Watson Machine Learning.

4. INTRODUCTION TO WATSON STUDIO:

a. Learning to build own Machine Learning model using IBM Watson.
b. Automate the Machine Learning Model

5. PREDICTING LIFE EXPECTANCY WITH PYTHON:

a. Collecting Data set from www.kaggle.com
b. Creating IBM Watson services
c. Create a jupyter notebook and import data from Object storage.

6. PREDICTING LIFE EXPECTANCY WITHOUT PYTHON:

a. Created Node-Red model and integrated with Machine Learning model.

4. Experimental Investigation

Life Expectancy 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

Importing libraries

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df.head()
```

Out[2]:

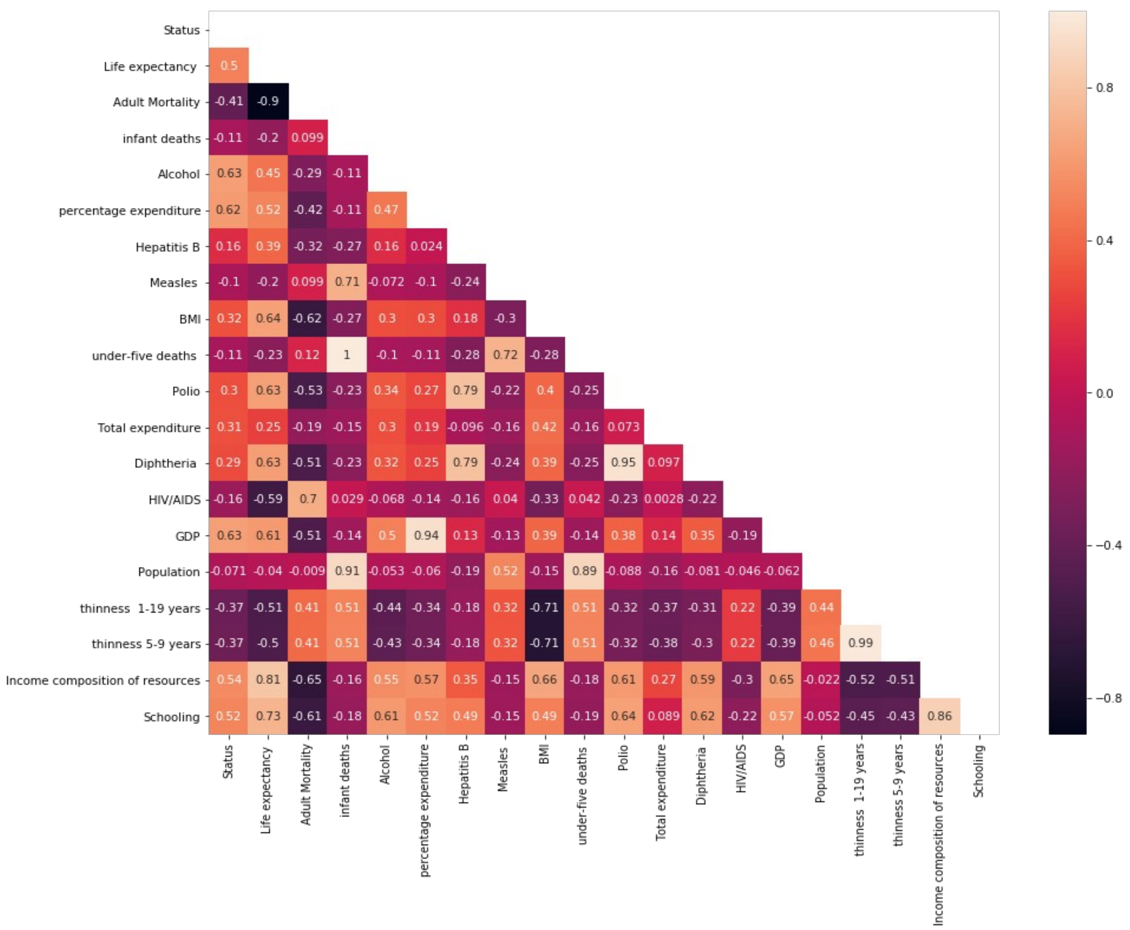
	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 1-19 years	thinr y
0	Afghanistan	2015	0	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.259210	33736494.0	17.2	
1	Afghanistan	2014	0	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.696514	327582.0	17.5	
2	Afghanistan	2013	0	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.744976	31731688.0	17.7	
3	Afghanistan	2012	0	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.959000	3696958.0	17.9	
4	Afghanistan	2011	0	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.537231	2978599.0	18.2	

5 rows × 22 columns

Visualizing Data

In [20]:
```
plt.figure(figsize = (15, 12))
matrix = np.triu(LifeData.corr())
sns.heatmap(LifeData.corr(), annot=True, mask=matrix)
```

Out[20]: `<matplotlib.axes._subplots.AxesSubplot at 0x7f5981fc9b38>`



Spliting into testing and training data

In [10]:
```
from sklearn.model_selection import train_test_split
LifeFeatures_train, LifeFeatures_test, LifeLabels_train, LifeLabels_test = train_test_split(
        LifeFeatures, LifeLabels, train_size = 0.7, test_size = 0.3)
```

Model Training

In [11]:
```
from sklearn.ensemble import RandomForestRegressor
p=RandomForestRegressor(n_estimators=40,random_state=50)

p.fit(LifeFeatures_train,LifeLabels_train)
```

Out[11]:
```
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)
```

Errors

In [12]:
```
from sklearn import metrics

predictions = p.predict(LifeFeatures_test)
print('MAE',metrics.mean_absolute_error(LifeLabels_test,predictions))
print('MSE',metrics.mean_squared_error(LifeLabels_test,predictions))
print('RMSE',np.sqrt(metrics.mean_squared_error(LifeLabels_test,predictions)))
```

```
MAE 1.8649918886612031
MSE 5.474815527594329
RMSE 2.3398323716869824
```

Accuracy

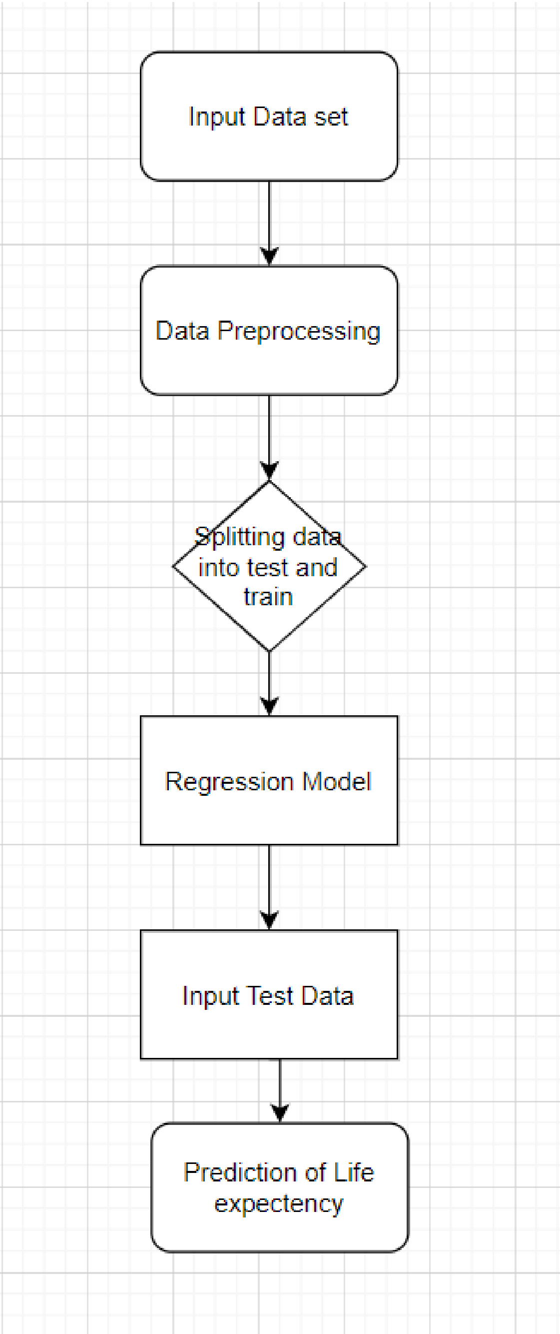
In [13]:
```
metrics.explained_variance_score(LifeLabels_test,predictions)*100
```

Out[13]: 91.4422203779417

-

Country	Actual	Predicted
Barbados	74.356250	76.607031
Trinidad and Tobago	71.068750	71.719214
Mauritania	62.800000	63.294059
Cook Islands	69.224932	69.651511
Israel	81.300000	81.422969
Cambodia	64.343750	61.343750
Azerbaijan	70.731250	72.424688
Jordan	72.987500	74.607031
Monaco	69.224932	69.358082
El Salvador	71.743750	72.842969
South Africa	57.500000	53.728750
New Zealand	81.337500	80.923281
Lesotho	48.781250	53.580156
Malta	80.362500	78.524062
Costa Rica	78.593750	76.595156
Netherlands	81.131250	81.833594
Nepal	66.481250	66.519687
Panama	76.487500	73.653281
Tuvalu	69.224932	68.115587
Venezuela (Bolivarian Republic of)	73.387500	72.503280
Cuba	77.975000	74.954844
Guatemala	71.731250	71.889217
Montenegro	74.500000	72.799684

5.Flowchart



To integrate the ML model with the UI, we would be using the Node Red functionality provided by the IBM Watson Studio.

To design the UI, we need to import the flow of the UI.

Once, we have setup the flow, we need to integrate the ML model with it. To integrate the ML Model with it we need to access the endpoint

URL of our ML Model.

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.

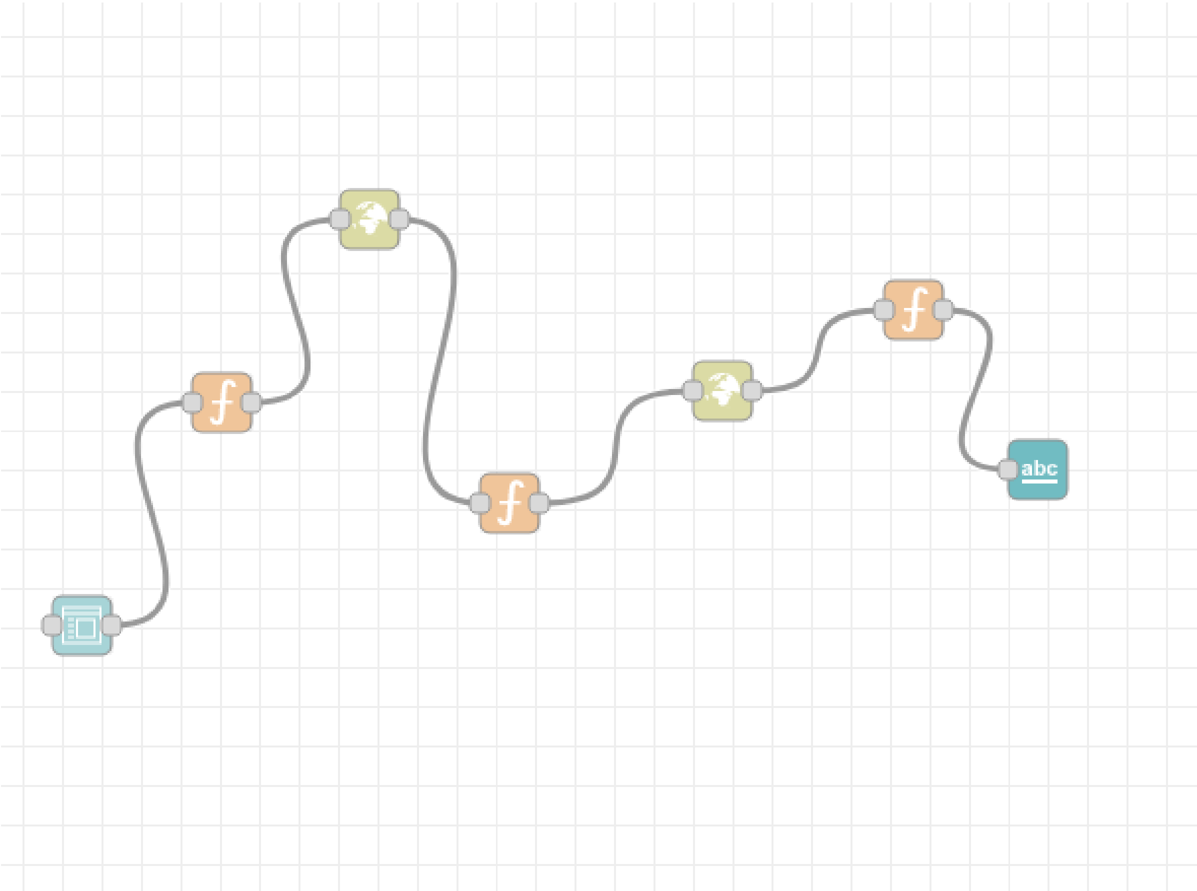
The first http request requires a token to connect to the

machine learning service of the Watson studio.

The second http request helps us in integrating the model using

the endpoint URL.

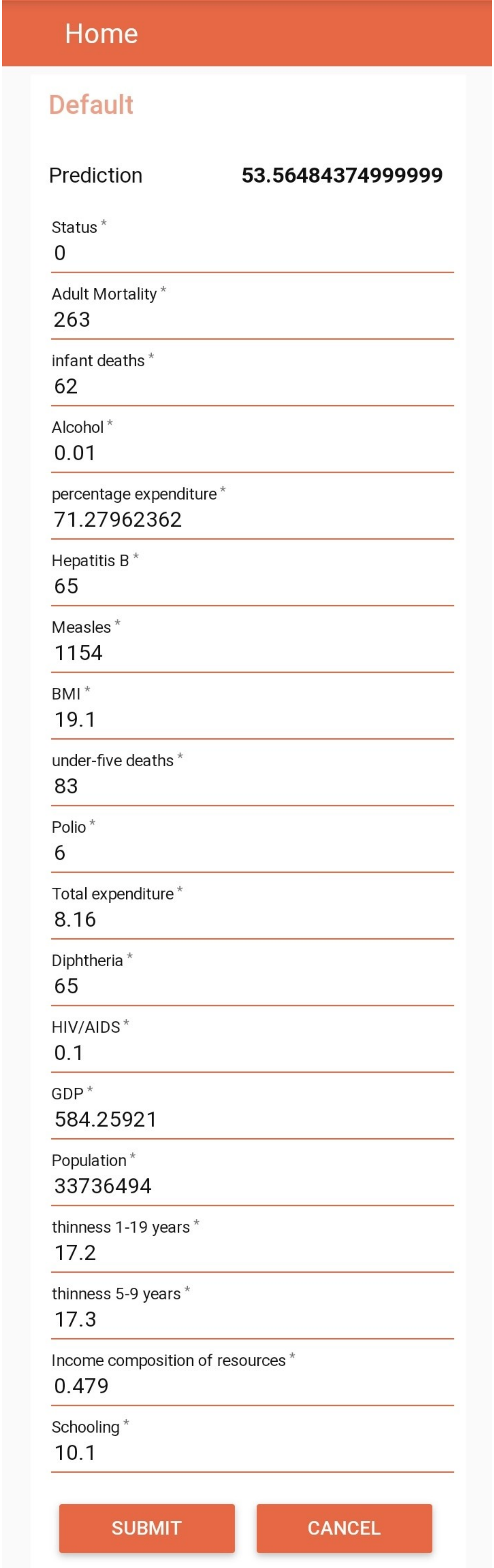
Once the flow has been setup, we deploy the model.



6. Result

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

URL for UI Dashboard: https://node-red-mobbasher.mybluemix.net/ui/#!/0?socketid=YMU4HwIWBe28sHvyAAAC


Home
Default
Prediction
53.56484374999999
Status *
0
Adult Mortality *
263
infant deaths *
62
Alcohol *
0.01
percentage expenditure *
71.27962362
Hepatitis B *
65
Measles *
1154
BMI *
19.1
under-five deaths *
83
Polio *
6
Total expenditure *
8.16
Diphtheria *
65
HIV/AIDS *
0.1
GDP *
584.25921
Population *
33736494
thinness 1-19 years *
17.2
thinness 5-9 years *
17.3
Income composition of resources *
0.479
Schooling *
10.1
SUBMIT
CANCEL

7. ADVANTAGE & DISADVANTAGES

7.1. Advantages:

1. Advantages of using IBM Watson:
   - Processes unstructured data
   - Fills human limitations
   - Acts as a decision support system, doesn’t replace humans
   - Improves performance + abilities by giving best available data
   - Improve and transform customer service
   - Handle enormous quantities of data
   - Sustainable Competitive Advantage
2. Easy for user to interact with the model via the UI.
3. User-friendly.
4. Easy to build and deploy.
5. Doesn’t require much storage space.

7.2. Disadvantages:

1. Disadvantages of using IBM Watson:
   - Seen as disruptive technology
   - Maintenance
   - Doesn't process structured data directly
   - Increasing rate of data, with limited resources
2. Not connected to database, hence no record of input.
3. Requires internet connection.

8. Applications

• Personalized Life Expectancy: Individuals can predict their own life expectancy by inputting values in the corresponding fields. This could help make people more aware of their general health, and its improvement or deterioration over time. This may motivate them to make healthier lifestyle choices.

• Government: 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. Across countries, high life expectancy is associated with high income per capita. Increase in life expectancy also leads to an increase in the “manpower” of a country. The knowledge asset of a country increases with the number of individuals in a country.

• Health Sector: Based 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 Companies: Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

9. Conclusion

Thus, we have developed a model that will predict the life expectancy of a specific demographic region based on the inputs provided. Various factors have a significant impact on the life span such as Adult Mortality, Population, Under 5 Deaths, Thinness 1-5 Years, and Alcohol, HIV, Hepatitis B, GDP, Percentage Expenditure and many more. User can interact with the system via a simple user interface which is in the form of a form with input spaces which the user needs to fill the inputs into.

10. Future Scope

For future use, one can integrate the life expectancy prediction with providing suggestions and medications to the individual using the application. This will help predict as well as increase the individual’s life expectancy. 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. Mental health can also be taken into account while predicting life expectancy with the help of sentiment analysis systems as well.

11. Bibliography

1. **Node-RED Starter Application :** https://developer.ibm.com/tutorials/how-to-create-a- node-red-starter-applicat ion/

2. **Watson Studio Cloud :** https://bookdown.org/caoying4work /watsonstudio-workshop/jn.html

3. **Dataset Reference :** https://www.kaggle.com/kumarajarshi/life-expectancy-who

4. **IBM Cloud Services :** https://www.youtube.com/watch?v=DBRGlAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L

5. **Import the Dataset into Jupyter Notebook :** https://www.youtube.com/watch?v=Jtej3Y6uUng

Appendix
A. Source Code

Importing libraries

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

Importing 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_b3482f8df8b645bd81d527c106c7c007 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='IFajjGfHefn_pW7C0hn1VbtMMLqou9HcSn_DVgL9RO_F',
    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_b3482f8df8b645bd81d527c106c7c007.get_object(Bucket='predictinglifeexpec
tancyusingmach-donotdelete-pr-y2lowwhgqvyo32',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 )

df = pd.read_csv(body)
```

```
df.replace(to_replace=['Developing', 'Developed'],
           value= [0, 1],
           inplace=True)
df.head()

LifeData=df.drop('Year',axis=1)
```

Analysing Data

```
(LifeData.describe())
LifeData.columns


# ##renaming the before spaced name column
LifeData.rename(columns={' thinness  1-19 years':'thinness  1-19 years',
                     ' thinness 5-9 years':'thinness 5-9 years',
                     ' HIV/AIDS':'HIV/AIDS',
                      ' BMI ':'BMI'},inplace=True)

LifeData.info()
```

Visualizing Data

```
plt.figure(figsize = (15, 12))
matrix = np.triu(LifeData.corr())
sns.heatmap(LifeData.corr(), annot=True, mask=matrix)
LifeData.head()
```

Changing categorical Column

```
LifeData = LifeData.groupby('Country').mean()
LifeData.head(20)
```

Making X & Y

```
LifeLabels = LifeData['Life expectancy ']
LifeFeatures = LifeData.drop('Life expectancy ', axis = 1)
```

FILL MISSING VALUES

```
print(LifeFeatures.isnull().sum())
print(LifeLabels.isnull().sum())
LifeFeatures.fillna(value = LifeFeatures.mean(), inplace = True)
LifeLabels.fillna(value = LifeLabels.mean(), inplace = True)
print('-------------------------------------------------------')
print(LifeFeatures.isnull().sum())
print(LifeLabels.isnull().sum())
```

Feature Scaling

```
from sklearn.preprocessing import MinMaxScaler
min_max_scaler = MinMaxScaler()
LifeFeatures = min_max_scaler.fit_transform(LifeFeatures)
```

```
LifeFeatures
```

Spliting into testing and training data

```
from sklearn.model_selection import train_test_split
LifeFeatures_train, LifeFeatures_test, LifeLabels_train, LifeLabels_test =
train_test_split(
        LifeFeatures, LifeLabels, train_size = 0.7, test_size = 0.3)
```

Model Training

```
from sklearn.ensemble import RandomForestRegressor
p=RandomForestRegressor(n_estimators=40,random_state=50)

p.fit(LifeFeatures_train,LifeLabels_train)
```

Errors

```
from sklearn import metrics

predictions = p.predict(LifeFeatures_test)
print('MAE',metrics.mean_absolute_error(LifeLabels_test,predictions))
print('MSE',metrics.mean_squared_error(LifeLabels_test,predictions))
print('RMSE',np.sqrt(metrics.mean_squared_error(LifeLabels_test,predictions)))
```

Accuracy

```
metrics.explained_variance_score(LifeLabels_test,predictions)*100

df=pd.DataFrame({'Actual':LifeLabels_test, 'Predicted':predictions})
print(df.head(25))
```

Testing with custom inputs

```
x=pd.DataFrame({'Status':[0],'Adult Mortality':[263], 'infant deaths':[62],
'Alcohol':[0.01],
       'percentage expenditure':[71.27962362], 'Hepatitis B':[65], 'Measles ':
[1154], 'BMI ':[19.1],
       'under-five deaths ':[83], 'Polio':[6], 'Total expenditure':[8.16],
'Diphtheria ':[65],
       'HIV/AIDS':[0.1], 'GDP':[584.25921], 'Population':[33736494], 'thinness
1-19 years':[17.2],
       'thinness 5-9 years':[17.3], 'Income composition of resources':[0.479],
'Schooling':[10.1],})
```

```
prediction=p.predict(x)
print(prediction)
```

Creation of end-point

```
!pip install watson-machine-learning-client

from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_credentials={
  "apikey": "",
  "instance_id": "",
  "url": ""
}

client = WatsonMachineLearningAPIClient( wml_credentials )

model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "",
               client.repository.ModelMetaNames.AUTHOR_EMAIL: "@gmail.com",
               client.repository.ModelMetaNames.NAME: "lifeExpectancy"}

model_artifact =client.repository.store_model(p, meta_props=model_props)

published_model_uid = client.repository.get_model_uid(model_artifact)

published_model_uid

deployment = client.deployments.create(published_model_uid,
name="lifeExpectancy")

scoring_endpoint = client.deployments.get_scoring_url(deployment)

scoring_endpoint
```

https://us-south.ml.cloud.ibm.com/v3/wml_instances/466a9b47-1657-4009-a824-fcce122b8f28/deployments/bf6fa95f-0d3b-41ee-9c72-0c16c18650de/online

Node-RED Flow code


```
[{"id":"fc3e1573.1da558","type":"tab","label":"Flow
1","disabled":false,"info":""},
{"id":"24889cde.3077c4","type":"ui_form","z":"fc3e1573.1da558","name":"","labe
l":"","group":"53329c69.6f02d4","order":0,"width":0,"height":0,"options":
[{"label":"Status","value":"a","type":"number","required":true,"rows":null},
{"label":"Adult
Mortality","value":"b","type":"number","required":true,"rows":null},
{"label":"infant
deaths","value":"c","type":"number","required":true,"rows":null},
{"label":"Alcohol","value":"d","type":"number","required":true,"rows":null},
{"label":"percentage
expenditure","value":"e","type":"number","required":true,"rows":null},
{"label":"Hepatitis
```

```
B","value":"f","type":"number","required":true,"rows":null},
{"label":"Measles","value":"g","type":"number","required":true,"rows":null},
{"label":"BMI","value":"h","type":"number","required":true,"rows":null},
{"label":"under-five
deaths","value":"i","type":"number","required":true,"rows":null},
{"label":"Polio","value":"j","type":"number","required":true,"rows":null},
{"label":"Total
expenditure","value":"k","type":"number","required":true,"rows":null},
{"label":"Diphtheria","value":"l","type":"number","required":true,"rows":null}
,{"label":"HIV/AIDS","value":"m","type":"number","required":true,"rows":null},
{"label":"GDP","value":"n","type":"number","required":true,"rows":null},
{"label":"Population","value":"o","type":"number","required":true,"rows":null}
,{"label":"thinness  1-19
years","value":"p","type":"number","required":true,"rows":null},
{"label":"thinness 5-9
years","value":"q","type":"number","required":true,"rows":null},
{"label":"Income composition of
resources","value":"r","type":"number","required":true,"rows":null},
{"label":"Schooling","value":"s","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":""},"payload":"","submit"
:"submit","cancel":"cancel","topic":"","x":90,"y":339,"wires":
[["fe01ebdc.e1654"]],"l":false},
{"id":"fe01ebdc.e1654","type":"function","z":"fc3e1573.1da558","name":"pre-
token","func":"global.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);\n\nvar
apikey=\"NG82plxgI8G5obwSF-t7X4Npo2ReoOoha2iS2aS-pDNX\";\
nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"};\
nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-
type:apikey\",\"apikey\":apikey};\nreturn
msg;","outputs":1,"noerr":0,"x":161,"y":226,"wires":
[["48dda523.af7a44"]],"l":false},{"id":"48dda523.af7a44","type":"http
request","z":"fc3e1573.1da558","name":"","method":"POST","ret":"obj","paytoqs"
:false,"url":"https://iam.cloud.ibm.com/identity/
token","tls":"","persist":false,"proxy":"","authType":"basic","x":236,"y":133,
"wires":[["5c88f121.bfca78"]],"l":false},
{"id":"5c88f121.bfca78","type":"function","z":"fc3e1573.1da558","name":"","fun
c":"var token=msg.payload.access_token;\nvar instance_id=\"466a9b47-1657-4009-
a824-fcce122b8f28\"\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\");\n\n\n//send the user values to service endpoint\
nmsg.payload = \n{\"fields\":[\"Status\", \"Adult Mortality\", \"infant
deaths\", \"Alcohol\", \"percentage expenditure\", \"Hepatitis
B\", \"Measles\", \"BMI\", \"under-five deaths\", \"Polio\", \"Total
expenditure\", \"Diphtheria\",
\"HIV/AIDS\", \"GDP\", \"Population\", \"thinness 10-19 years\", \"thinness
5-9 years\", \"Income composition of resources\", \"Schooling\"],\
n   \"values\":[[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s]]};\n   \
nreturn msg;","outputs":1,"noerr":0,"x":307,"y":277,"wires":
[["ca0e7470.5ba0c"]],"l":false},
{"id":"3cfb24a1.b04fac","type":"function","z":"fc3e1573.1da558","name":"getFro
m
Endpoint","func":"msg.payload=msg.payload.values[0][0];\n\n//msg.payload=msg.p
ayload.predictions[0].values[0][0];\nreturn
msg;","outputs":1,"noerr":0,"x":512,"y":179,"wires":
[["ad19a4d8.bed06"]],"l":false},
{"id":"ad19a4d8.bed06","type":"ui_text","z":"fc3e1573.1da558","group":"53329c6
9.6f02d4","order":1,"width":0,"height":0,"name":"","label":"Prediction","forma
t":"{{msg.payload}}","layout":"row-spread","x":575,"y":260,"wires":
[],"l":false},{"id":"ca0e7470.5ba0c","type":"http
request","z":"fc3e1573.1da558","name":"","method":"POST","ret":"obj","paytoqs"
:false,"url":"https://us-south.ml.cloud.ibm.com/v3/wml_instances/466a9b47-
1657-4009-a824-fcce122b8f28/deployments/bf6fa95f-0d3b-41ee-9c72-0c16c18650de/
online","tls":"","persist":false,"proxy":"","authType":"","x":415,"y":220,"wir
es":[["3cfb24a1.b04fac"]],"l":false},
{"id":"53329c69.6f02d4","type":"ui_group","z":"","name":"Default","tab":"1980f
79f.97dd6","order":1,"disp":true,"width":"6","collapse":false},
{"id":"1980f79f.97dd6","type":"ui_tab","z":"","name":"Home","icon":"dashboard"
,"disabled":false,"hidden":false}]