Predicting Life Expectancy Using Machine Learning

Project Report

By - R. Harini

INDEX

1. Introduction

1.1. Overview

Life expectancy refers to the number of years a person is expected to live. In mathematical terms, life expectancy refers to the expected number of years remaining for an individual at any given age.

The life expectancy for a particular person or population group depends on several variables such as their lifestyle, access to healthcare, diet, economic status and the relevant mortality and morbidity data. However, as life expectancy is calculated based on averages, a person may live for many years more or less than expected.

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

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

2.1. Existing Solution

As a result of the evolution of biotechnologies and related technologies such as the development of sophisticated medical equipment, humans are able to enjoy longer life expectancies than previously before. Predicting a human's life expectancy has been a long-term question to humankind. Many calculations and research have been done to create an equation despite it being impractical to simplify these variables into one equation.

Currently there are various smart devices and applications such as smartphone apps and wearable devices that provide wellness and fitness tracking. Some apps provide health related data such as sleep monitoring, heart rate measuring, and calorie expenditure collected and processed by the devices and servers in the cloud. However no existing works provide the Personalized Life expectancy.

2.2. Proposed Solution

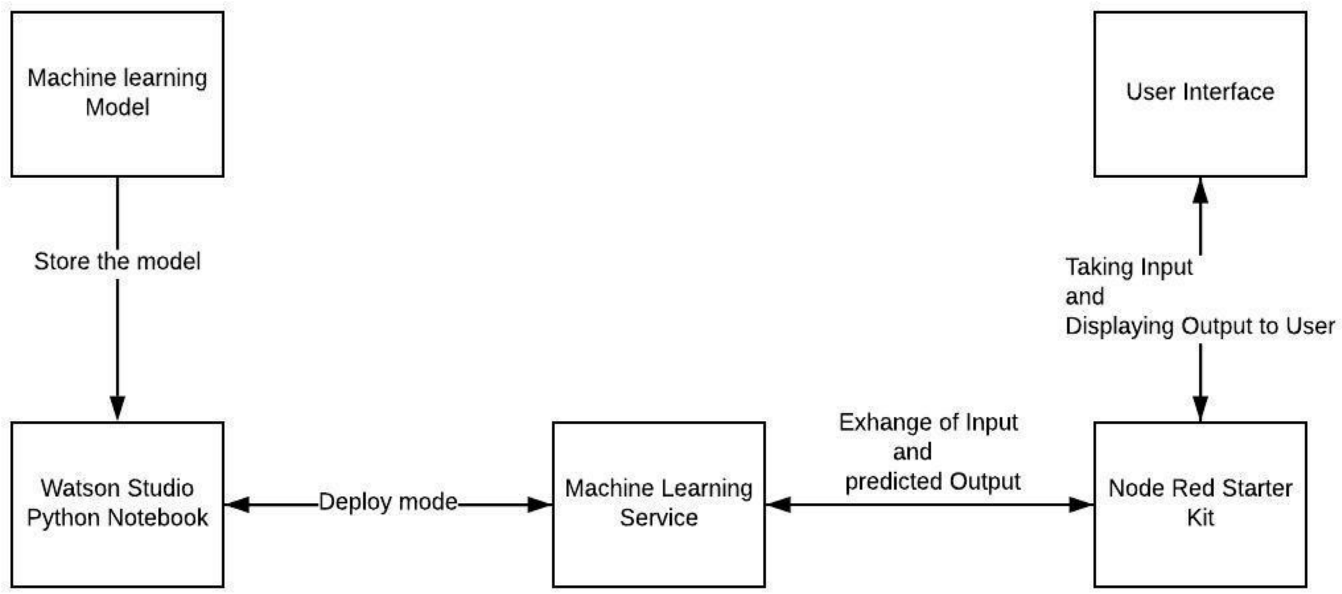
There has been an explosion of breakthroughs in the field of Machine Learning over the past few years. Machine Learning algorithms are capable of a lot and can-do wonders for the healthcare sector.

The proposed solution involves the use of Machine Learning algorithms specifically Regression models such as Linear Regression, Ridge regression, 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 taking 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 composition of resources, 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 interactive and user-friendly User Interface.

3. Theoretical Analysis

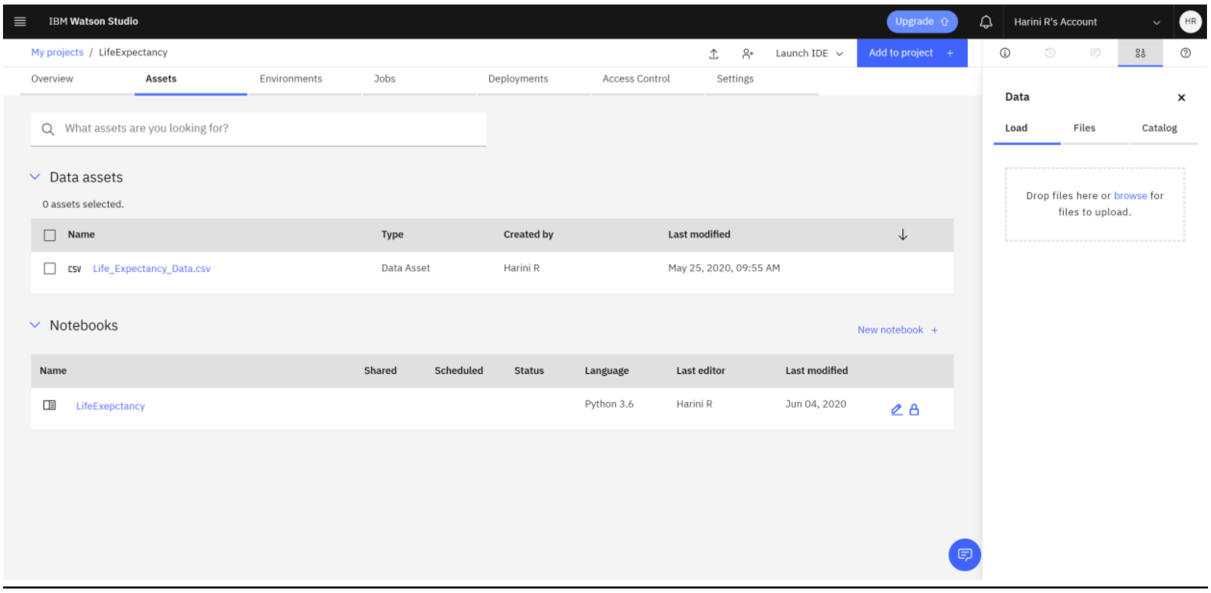
3.1. Block Diagram


Machine learning Model
Store the model
Watson Studio Python Notebook
Deploy mode
Machine Learning Service
Exhange of Input and predicted Output
Node Red Starter Kit
Taking Input and Displaying Output to User
User Interface

3.2. Hardware/ Software Designing

Model Designing **(Watson Studio) :**

Steps: New Project => Create an empty Project => Give project name => Click Create => Add to Project => Notebook



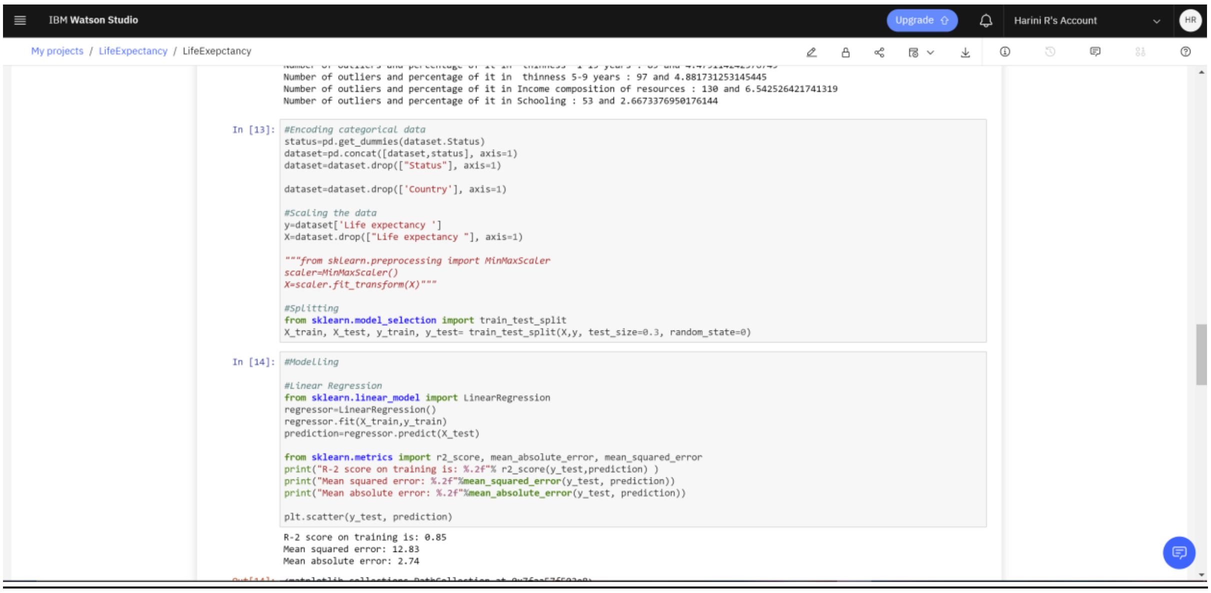
Data assets

0 assets selected.

Name	Type	Created by	Last modified
☐ CSV Life_Expectancy_Data.csv	Data Asset	Harini R	May 25, 2020, 09:55 AM

Notebooks

Name	Shared	Scheduled	Status	Language	Last editor	Last modified
LifeExepctancy				Python 3.6	Harini R	Jun 04, 2020

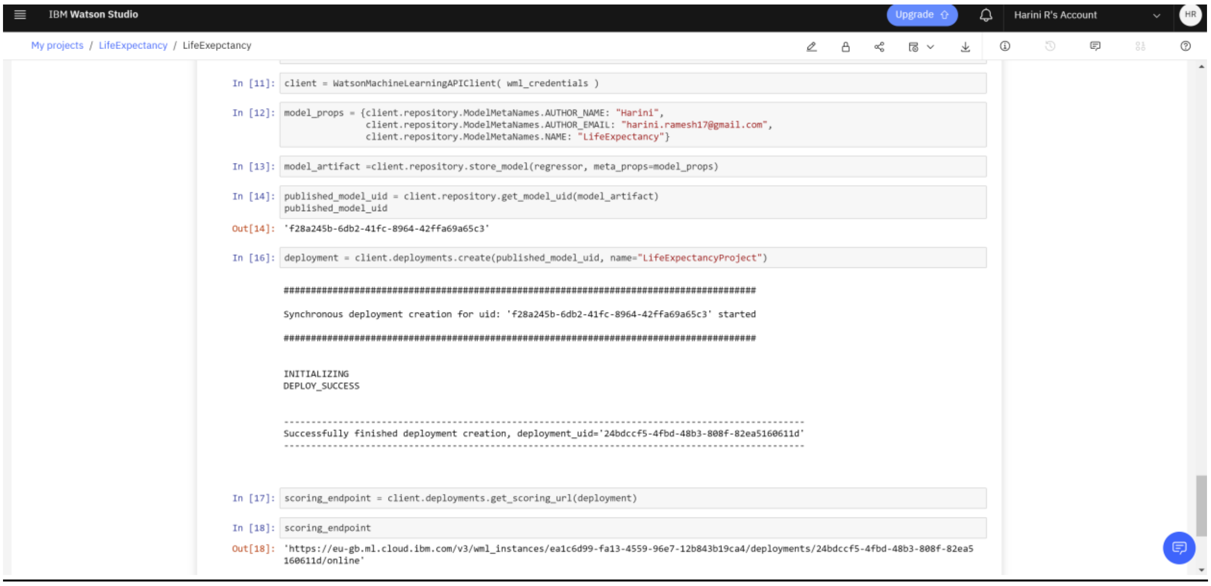


```
Number of outliers and percentage of it in  thinness 5-9 years : 97 and 4.881731253145445
Number of outliers and percentage of it in Income composition of resources : 130 and 6.542526421741319
Number of outliers and percentage of it in Schooling : 53 and 2.6673376950176144

In [13]: #Encoding categorical data
         status=pd.get_dummies(dataset.Status)
         dataset=pd.concat([dataset,status], axis=1)
         dataset=dataset.drop(["Status"], axis=1)

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

         #Scaling the data
         y=dataset['Life expectancy ']
         X=dataset.drop(["Life expectancy "], axis=1)

         """from sklearn.preprocessing import MinMaxScaler
         scaler=MinMaxScaler()
         X=scaler.fit_transform(X)"""

         #Splitting
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test= train_test_split(X,y, test_size=0.3, random_state=0)

In [14]: #Modelling

         #Linear Regression
         from sklearn.linear_model import LinearRegression
         regressor=LinearRegression()
         regressor.fit(X_train,y_train)
         prediction=regressor.predict(X_test)

         from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
         print("R-2 score on training is: %.2f"% r2_score(y_test,prediction) )
         print("Mean squared error: %.2f"%mean_squared_error(y_test, prediction))
         print("Mean absolute error: %.2f"%mean_absolute_error(y_test, prediction))

         plt.scatter(y_test, prediction)

         R-2 score on training is: 0.85
         Mean squared error: 12.83
         Mean absolute error: 2.74
```

Scoring Endpoint:

For wml credentials, replace with your own credentials of the service.

Services => Machine Learning Service => Service Credentials => Copy the credentials



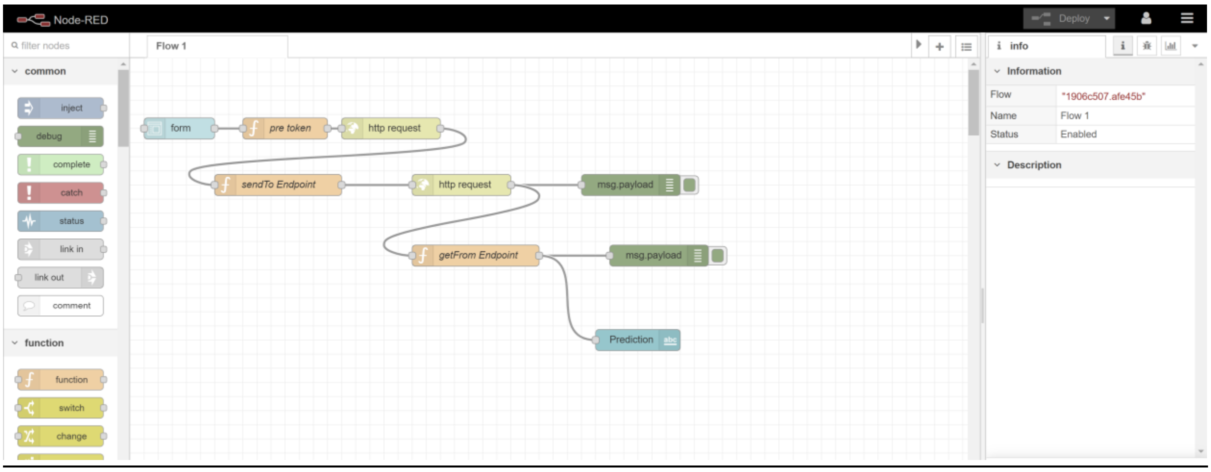
```
In [11]: client = WatsonMachineLearningAPIClient( wml_credentials )

In [12]: model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Harini",
                        client.repository.ModelMetaNames.AUTHOR_EMAIL: "harini.ramesh17@gmail.com",
                        client.repository.ModelMetaNames.NAME: "LifeExpectancy"}

In [13]: model_artifact =client.repository.store_model(regressor, meta_props=model_props)

In [14]: published_model_uid = client.repository.get_model_uid(model_artifact)
         published_model_uid

Out[14]: 'f28a245b-6db2-41fc-8964-42ffa69a65c3'

In [16]: deployment = client.deployments.create(published_model_uid, name="LifeExpectancyProject")

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

         Synchronous deployment creation for uid: 'f28a245b-6db2-41fc-8964-42ffa69a65c3' started

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


         INITIALIZING
         DEPLOY_SUCCESS


         ------------------------------------------------------------------------------------------------
         Successfully finished deployment creation, deployment_uid='24bdccf5-4fbd-48b3-808f-82ea5160611d'
         ------------------------------------------------------------------------------------------------


In [17]: scoring_endpoint = client.deployments.get_scoring_url(deployment)

In [18]: scoring_endpoint

Out[18]: 'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/ea1c6d99-fa13-4559-96e7-12b843b19ca4/deployments/24bdccf5-4fbd-48b3-808f-82ea5
         160611d/online'
```

User Interface Integration with ML Model **(Node- Red)** :

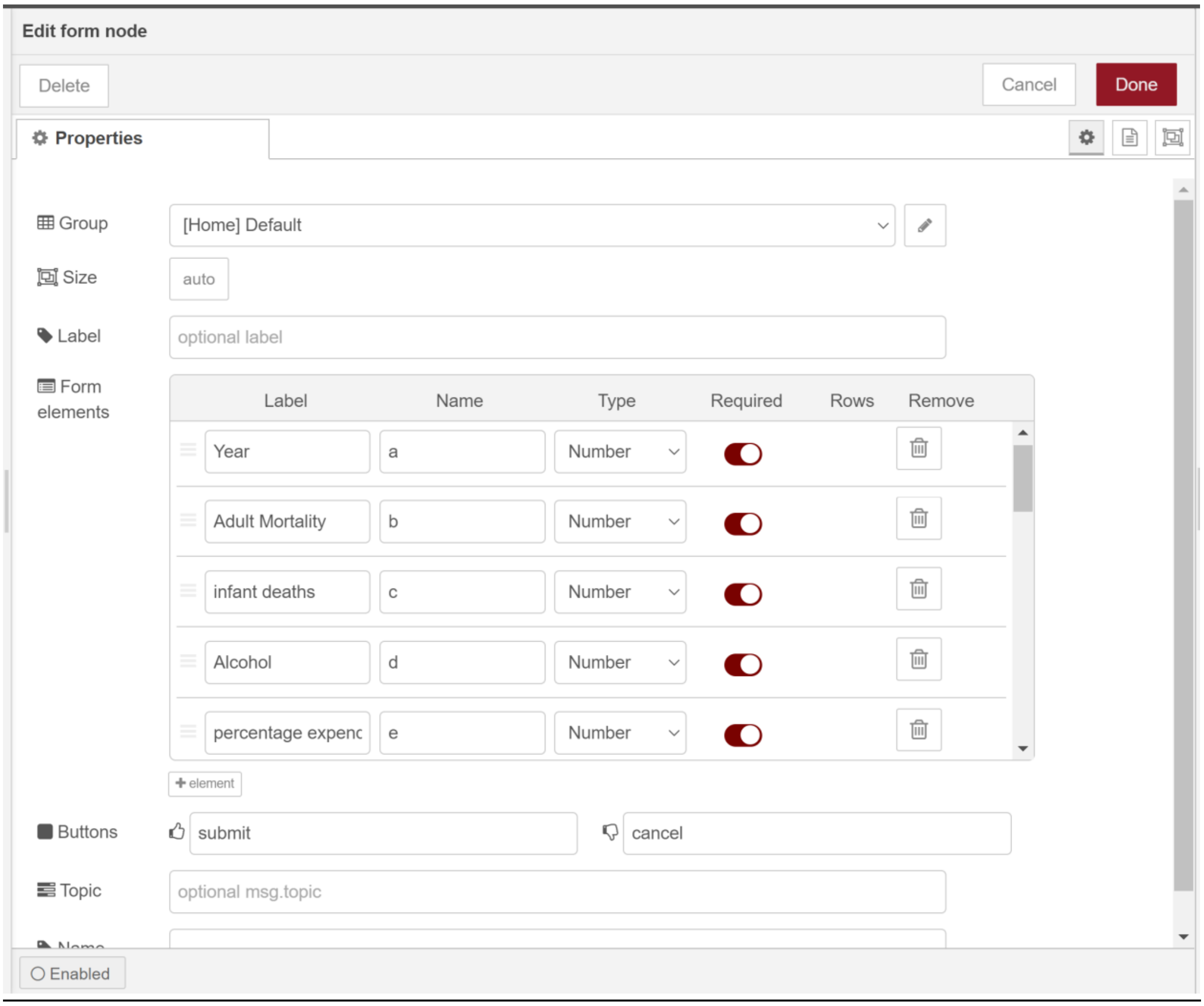
Nodes: 1) Form Node: Edit => Add New UI Tab

2) Function Node: To obtain access to Machine Learning Services. Requires API Key

3) HTTP Request Node: POST method and returns a parsed JSON object. Gains access to Machine Learning services.



Flow	"1906c507.afe45b"
Name	Flow 1
Status	Enabled



Group: [Home] Default

Size: auto

Label: optional label

Form elements

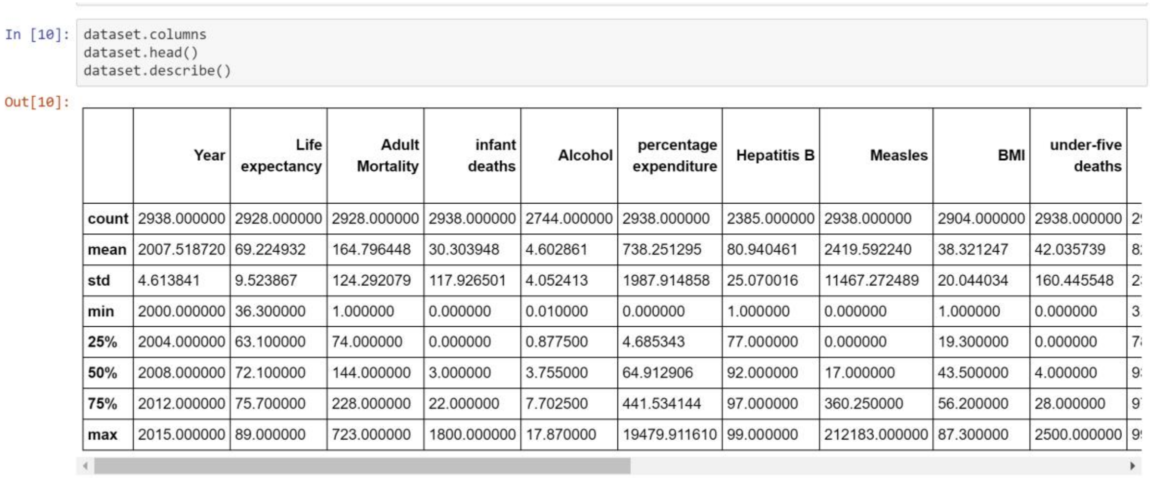
Label	Name	Type	Required	Rows	Remove
Year	a	Number	●		
Adult Mortality	b	Number	●		
infant deaths	c	Number	●		
Alcohol	d	Number	●		
percentage expenc	e	Number	●		

Buttons: submit | cancel

Topic: optional msg.topic

4. Experimental Investigations

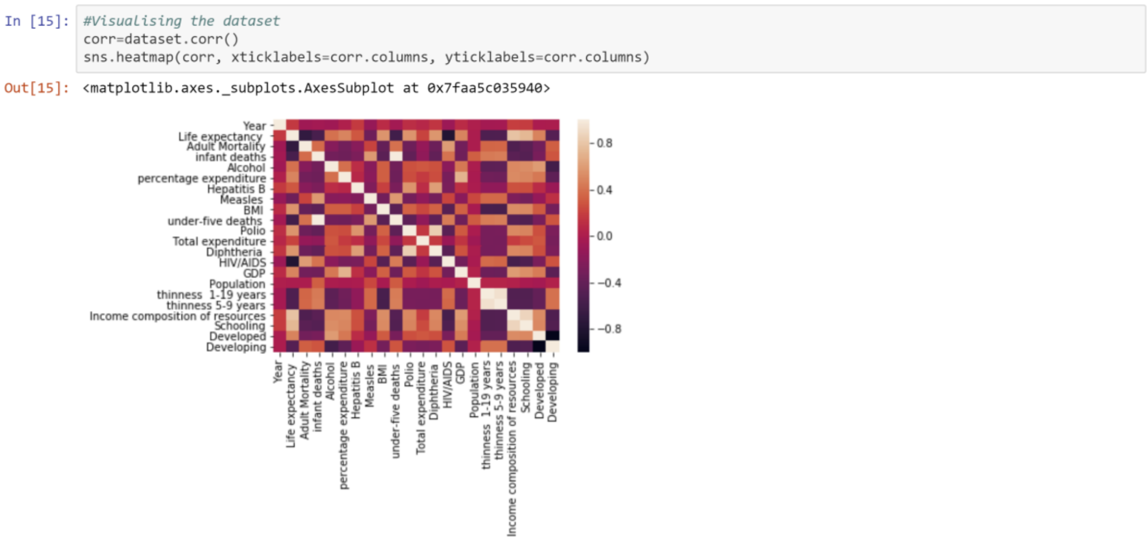
Analyzing the relations between various features can help us improve the performance of the model as well as decide which model would be more suitable.



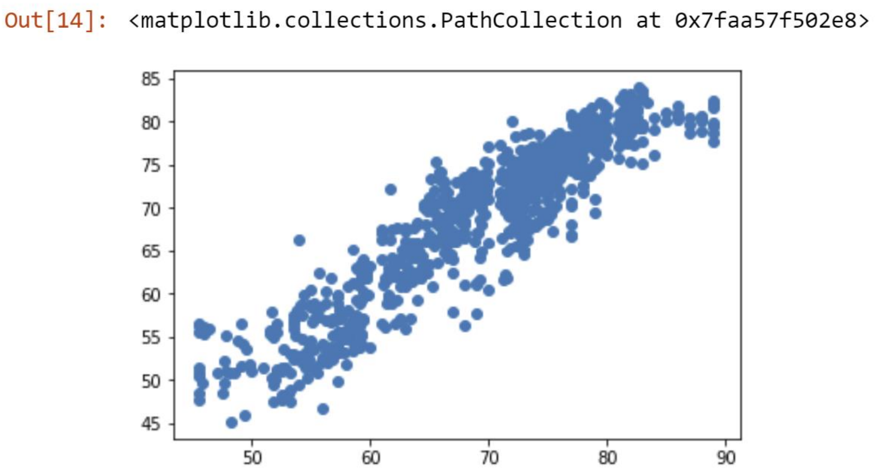
```
In [10]: dataset.columns
         dataset.head()
         dataset.describe()
```

Out[10]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000	2938.000000	2904.000000	2938.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461	2419.592240	38.321247	42.035739
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016	11467.272489	20.044034	160.445548
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000	0.000000	19.300000	0.000000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000	17.000000	43.500000	4.000000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000	360.250000	56.200000	28.000000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2500.000000

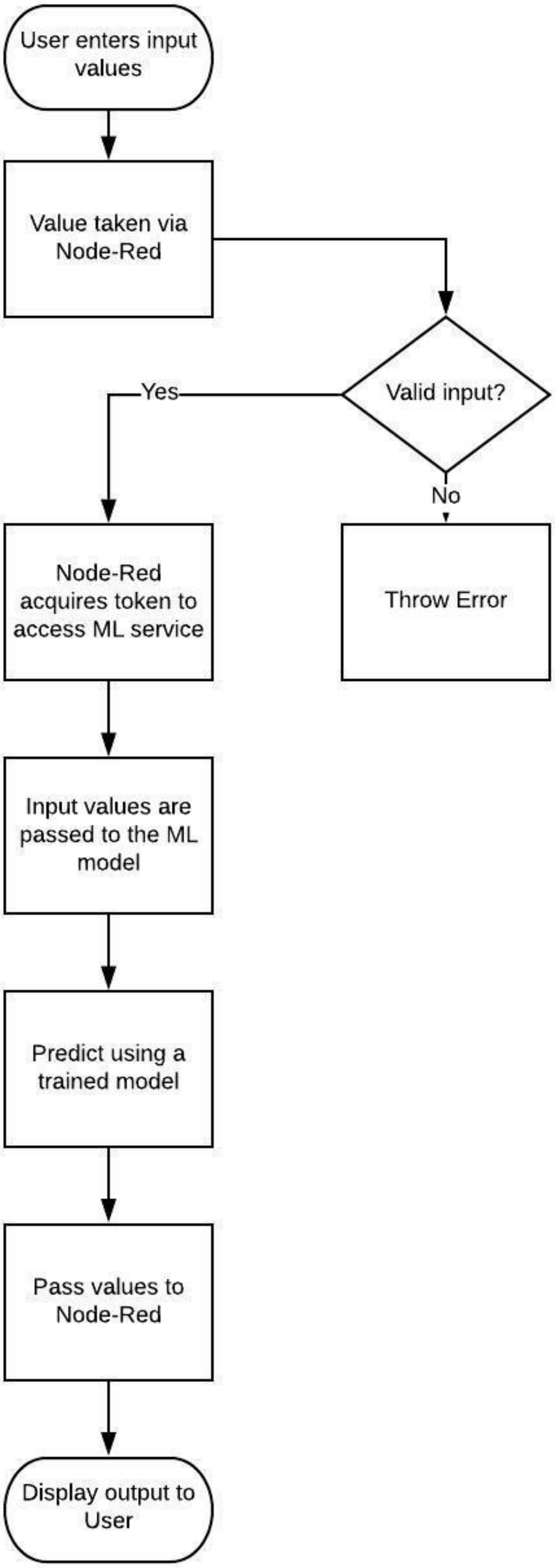


```
In [15]: #Visualising the dataset
         corr=dataset.corr()
         sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns)

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7faa5c035940>
```

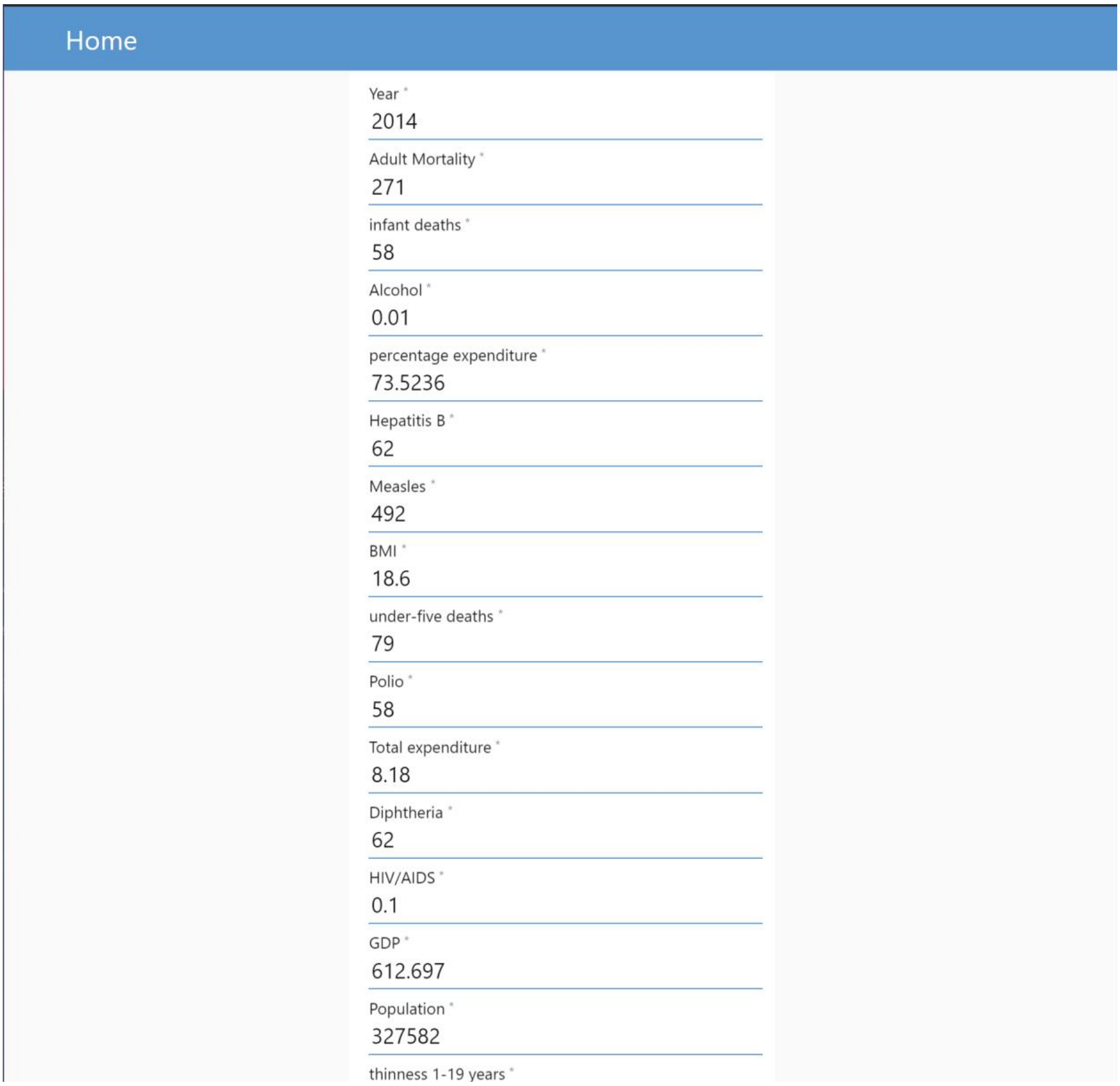


```
Out[14]: <matplotlib.collections.PathCollection at 0x7faa57f502e8>
```

5. Flowchart


User enters input values
Value taken via Node-Red
Valid input?
Yes
No
Node-Red acquires token to access ML service
Throw Error
Input values are passed to the ML model
Predict using a trained model
Pass values to Node-Red
Display output to User

6. Result



Year *
2014

Adult Mortality *
271

infant deaths *
58

Alcohol *
0.01

percentage expenditure *
73.5236

Hepatitis B *
62

Measles *
492

BMI *
18.6

under-five deaths *
79

Polio *
58

Total expenditure *
8.18

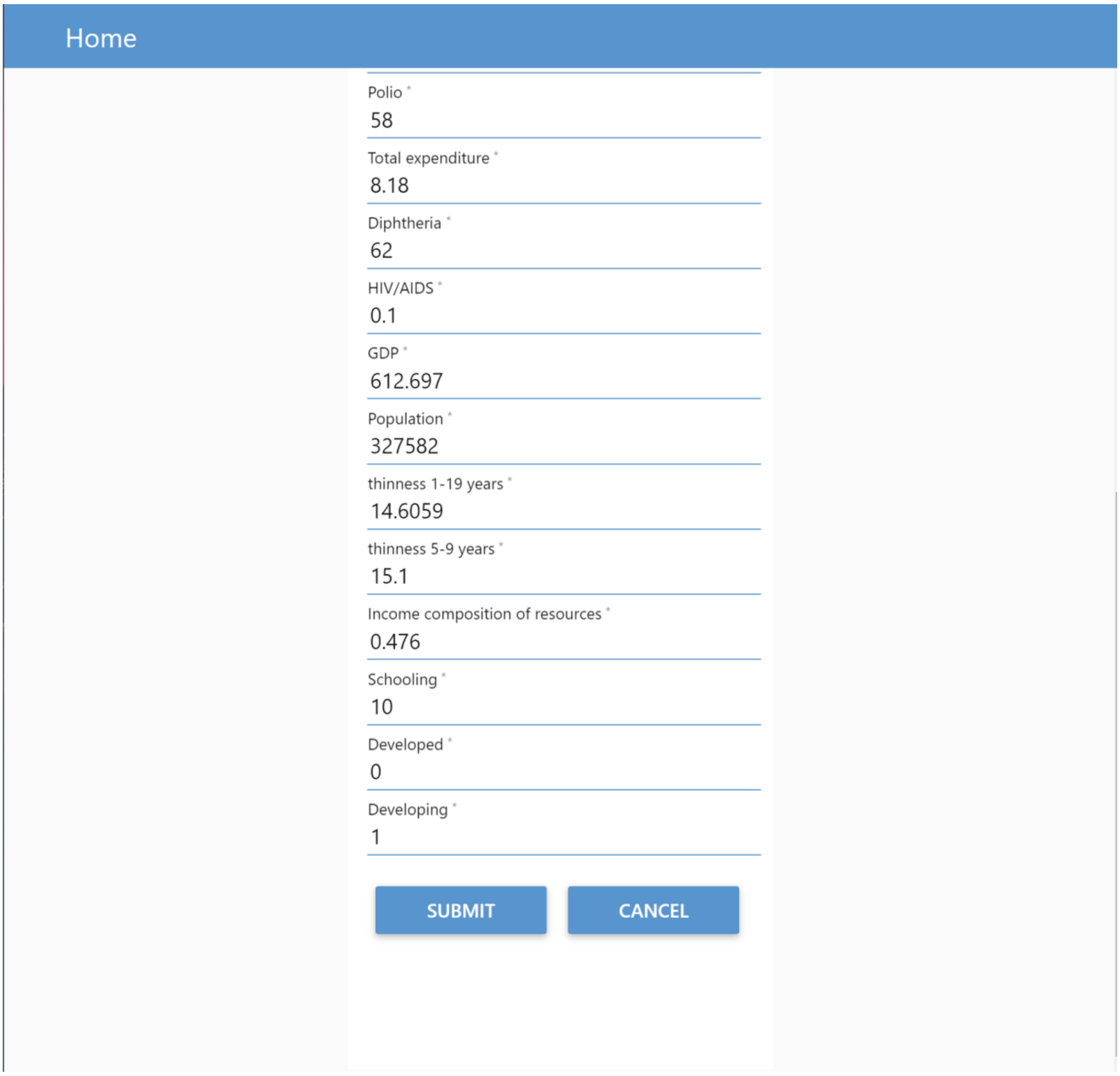
Diphtheria *
62

HIV/AIDS *
0.1

GDP *
612.697

Population *
327582

thinness 1-19 years *



Polio *
58

Total expenditure *
8.18

Diphtheria *
62

HIV/AIDS *
0.1

GDP *
612.697

Population *
327582

thinness 1-19 years *
14.6059

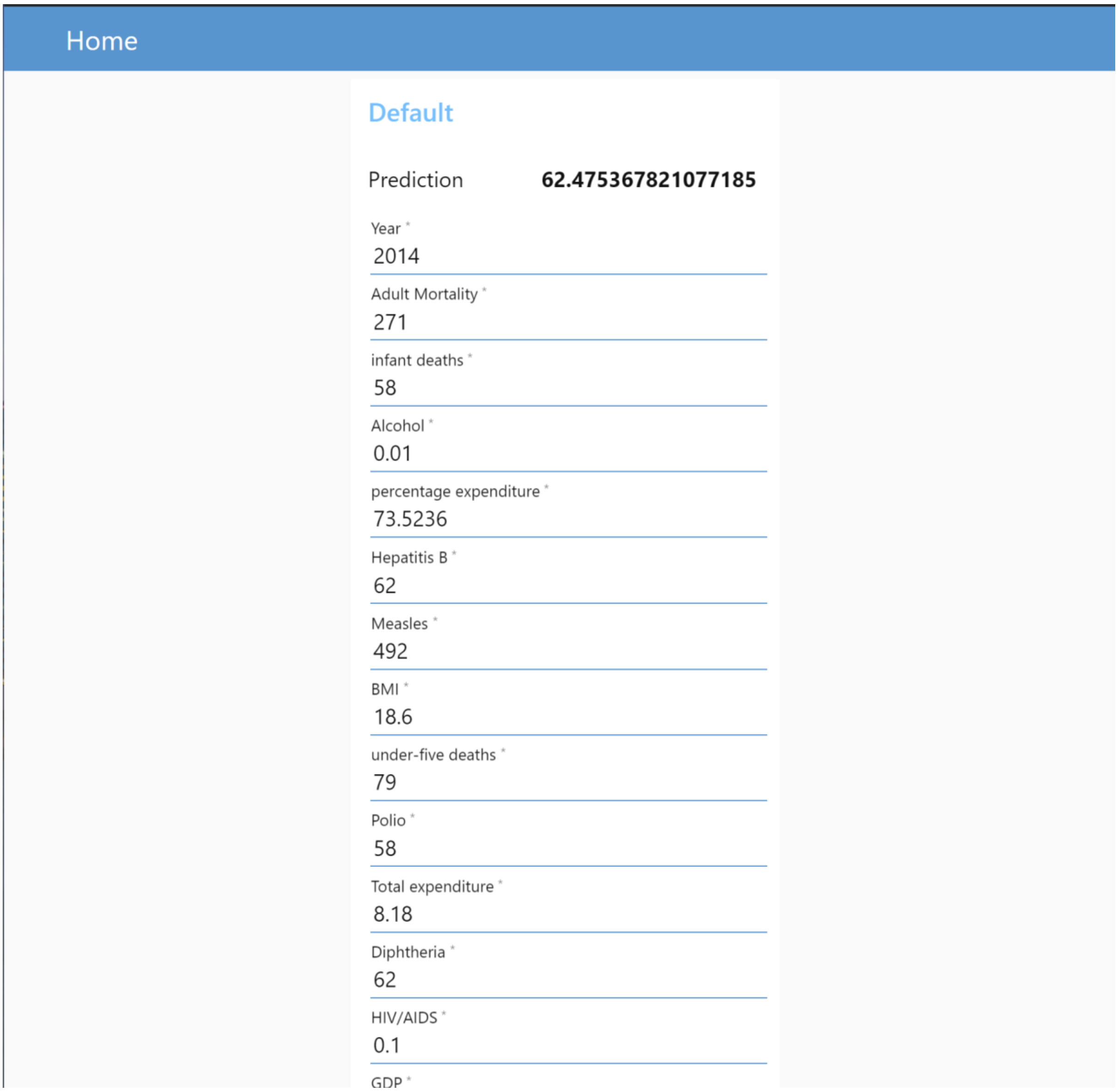
thinness 5-9 years *
15.1

Income composition of resources *
0.476

Schooling *
10

Developed *
0

Developing *
1



Default

Prediction 62.475367821077185

Year *
2014

Adult Mortality *
271

infant deaths *
58

Alcohol *
0.01

percentage expenditure *
73.5236

Hepatitis B *
62

Measles *
492

BMI *
18.6

under-five deaths *
79

Polio *
58

Total expenditure *
8.18

Diphtheria *
62

HIV/AIDS *
0.1

GDP *

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

1) 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.

2) 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.

3) 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.

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

9. Conclusion

Predicting lifespan of human beings can greatly alter our lives. Human behavior 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

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

- https://theconversation.com/dont-die-wondering-apps-may-soon-be-able-to-predict-your-life-expectancy-but-do-you-want-to-know-129068
- https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
- 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://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa69-9fab-40ee-8457-ea0e5df8c8de

12.Appendix

12.1. Source Code

Services Used:

- **Watson Assistant**
- **Watson Studio**
- **IBM Cloud Function**
- **Node-Red**

Python Notebook:

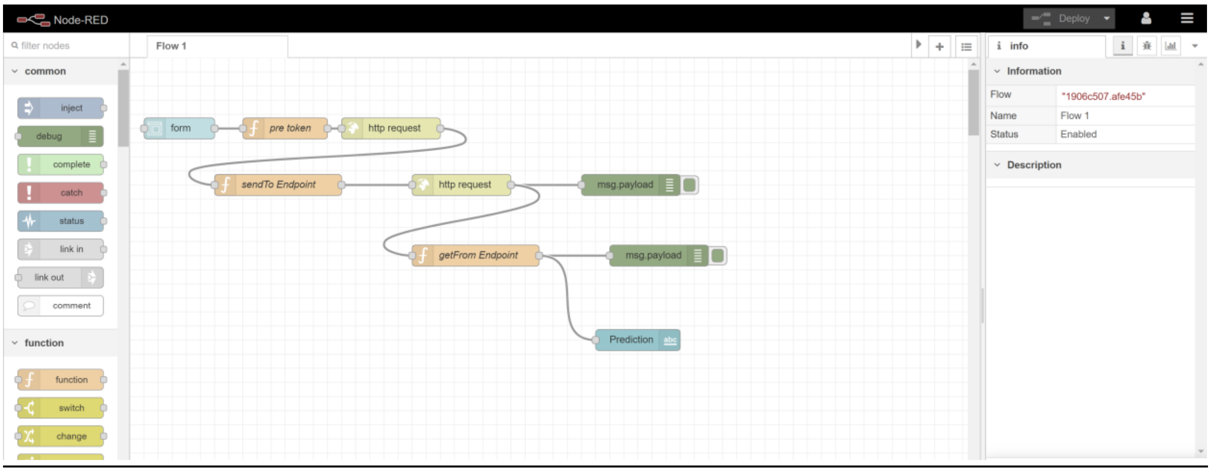
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def __iter__(self): return 0
body =
client_febaef7d68e841d388b92aaf0802061e.get_object(Bucket='lifeexpectancy-
donotdelete-pr-bjyv4bhazqwvva',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
)

dataset=pd.read_csv("Life_Expectancy_Data.csv")
new=dataset
dataset.columns
dataset.head()
dataset.describe()

features=['Country', 'Year', 'Status', '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']


#Visualising the dataset
corr=dataset.corr()
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns)
```

```
#Data Cleaning

dataset.isna().sum()

countries=dataset.Country.unique()

dataset=dataset.interpolate(method="linear",limit_direction="forward")
new=dataset

#dataset.fillna(dataset.mean(axis=0), inplace=True)

#Finding outliers
newfeatures=['Adult Mortality','Life expectancy ',
       '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 variable in newfeatures:
    q75, q25 = np.percentile(dataset[variable], [75 ,25])
    iqr = q75 - q25
    min_val = q25 - (iqr*1.5)
    max_val = q75 + (iqr*1.5)
    print("Number of outliers and percentage of it in {} : {} and {}".format(variable,
                                                                    len((np.where((dataset[variable] >
max_val) | (dataset[variable] < min_val))[0])),
                                                                    len((np.where((dataset[variable] >
max_val) | (dataset[variable] < min_val))[0]))*100/1987))

#Removing outliers
from scipy.stats.mstats import winsorize

dataset['Life expectancy '] = winsorize(dataset['Life expectancy '],(0.01,0))
dataset['Adult Mortality']= winsorize(dataset['Adult Mortality'],(0,0.03))
dataset['infant deaths'] = winsorize(dataset['infant deaths'],(0,0.10))
dataset['Alcohol'] = winsorize(dataset['Alcohol'],(0,0.01))
dataset['percentage expenditure']= winsorize(dataset['percentage
expenditure'],(0,0.12))
dataset['Hepatitis B']= winsorize(dataset['Hepatitis B'],(0.11,0))
dataset['Measles '] = winsorize(dataset['Measles '],(0,0.19))
dataset['under-five deaths ']= winsorize(dataset['under-five deaths '],(0,0.12))
dataset['Polio'] = winsorize(dataset['Polio'],(0.09,0))
dataset['Total expenditure'] = winsorize(dataset['Total expenditure'],(0,0.01))
```

```
dataset['Diphtheria ']= winsorize(dataset['Diphtheria '],(0.10,0))
dataset[' HIV/AIDS'] = winsorize(dataset[' HIV/AIDS'],(0,0.16))
dataset['GDP'] = winsorize(dataset['GDP'],(0,0.13))
dataset['Population'] = winsorize(dataset['Population'],(0,0.14))
dataset[' thinness  1-19 years']= winsorize(dataset[' thinness  1-19 years'],(0,0.04))
dataset[' thinness 5-9 years'] = winsorize(dataset[' thinness 5-9 years'],(0,0.04))
dataset['Income composition of resources'] = winsorize(dataset['Income composition of resources'],(0.05,0))
dataset['Schooling'] = winsorize(dataset['Schooling'],(0.02,0.01))


#Encoding categorical data
status=pd.get_dummies(dataset.Status)
dataset=pd.concat([dataset,status], axis=1)
dataset=dataset.drop(["Status"], axis=1)

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

#Scaling the data
y=dataset['Life expectancy ']
X=dataset.drop(["Life expectancy "], axis=1)

"""from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
X=scaler.fit_transform(X)"""

#Splitting
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test= train_test_split(X,y, test_size=0.3, random_state=0)


#Modelling

#Linear Regression
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(X_train,y_train)
prediction=regressor.predict(X_test)

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
print("R-2 score on training is: %.2f"% r2_score(y_test,prediction) )
print("Mean squared error: %.2f"%mean_squared_error(y_test, prediction))
print("Mean absolute error: %.2f"%mean_absolute_error(y_test, prediction))
```

```
plt.scatter(y_test, prediction)

from watson_machine_learning_client import WatsonMachineLearningAPIClient

client = WatsonMachineLearningAPIClient( wml_credentials )

model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Harini",
            client.repository.ModelMetaNames.AUTHOR_EMAIL:
"harini.ramesh17@gmail.com",
            client.repository.ModelMetaNames.NAME: "LifeExpectancy"}

model_artifact =client.repository.store_model(regressor,
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="LifeExpectancyProject")

scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint
```

Node Red Flow:



Flow	"1906c507.afe45b"
Name	Flow 1
Status	Enabled