PROJECT REPORT

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

A

Project

Report

On

Predicting Life Expectancy using Machine Learning

Internship

under:

TheSMARTBRIDGE

NAME: ASHUTOSH SHARMA

EMAIL: ashutosh284200@gmail.com

PROJECT ID: SPS_PRO_215

INTERNSHIP TITLE: Predicting Life Expectancy using Machine Learning - SB39716

Category: Machine Learning

TABLE OF CONTENT

CHAPTER 1 INTRODUCTION

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 research tests the potential of using machine learning techniques for predicting life expectancy medical records.

1.1 Overview

This project "Predicting Life Expectancy using Machine Learning" is a web application that predict the expected average life span of human based on diverse datasets, in a demographic region. The life of a human depends on various factors such as Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical illnesses, Education, Year of their birth and other demographic factors. The project aims to predict an average life expectancy based on these and several other factors. This project is built using IBM services (Watson studio, Node Red, Watson machine learning, Python Flask).

This project finds the expected solution using various machine learning algorithms such as:

a) Linear Regression
b) Decision Tree
c) Random Forest

And many more, the aim of the project is to find the relationship of the various factors with the lifespan of an individual using the ML Algorithms mentioned above. 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. The dataset used for the prediction contains data from year 2000 to 2015. It contains more than 2500 entries and around 22 columns with various features such as Population, Alcohol Consumption, Infant Mortality Rate etc., which aids the prediction of the model.

1.2 Purpose

If life expectancy is longer in a certain country, it speaks about the conditions of the place. It tells information on the health factors as well as the quality of life. If the conditions in a country and in its economy are good, obviously the life expectancy would be more and greater number of people would like to live in the same country. Life expectancy is the most important factor for decision making. By predicting life expectancy and having good prognostication can help in making valuable decision like the course of treatment and helps to anticipate the procurement of health care services and facilities. Accurate prognosis of life expectancy is essential for general practitioners (GPs) to decide when to introduce the topic of ACP (Advance Care Planning) to the patient, and it is a key determinant in end-of-life decisions. Increasing the accuracy of prognoses has the potential to benefit patients in various ways by enabling more consistent ACP, earlier and better anticipation on palliative needs, and preventing excessive treatment.

2.1 EXISTING PROBLEM

As we already know few works have been done to provide an individually customized life expectancy prediction. We already have reviewed existing works and techniques in the prediction of human Life Expectancy and reached a conclusion that it is feasible to predict Life Expectancy for individuals using evolving technologies and devices such as big data, AI, machine learning techniques, and PHDs, wearable devices and mobile health monitoring devices. We also know that the collection of data for research/making a model will be a huge challenge due to the privacy and government policy considerations, which will require collaboration of various bodies in the health industry. Despite these challenges, accurate prognosis is notoriously difficult, a possibility of a PLE prediction by proposing an approach of data collection and application by smart phone, with which users can enter their information to access the cloud server to obtain their own PLE, was shown.

Although 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.

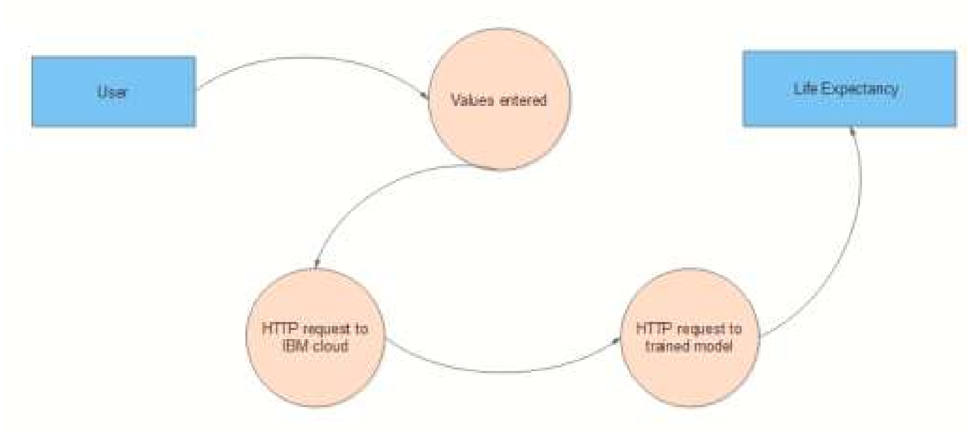
2.2 PROPOSED SOLUTION

Although 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. So, 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. The data will contain important factors which are crucial for our model like Hepatitis B, Polio and Diphtheria. In a study will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. 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 model of" Predicting Life Expectancy using Machine Learning" uses IBM Cloud services, which helps to avoid any storage issues. The UI Presented to the users is a website URL and hence they need not download any application to predict the results, which saves the storage space as that is the need of the hour.

CHAPTER 3 THEORITICAL ANALYSIS

3.1 BLOCK/FLOW DIAGRAM



The above diagram shows the flow of the model, the user will give input to the model through "Form" element in Node-Red which is use to create UI for user so that they can easily interact with model. After input receive an HTTP request is made to the IBM cloud that further makes an HTTP request to the deployed model using model's instance id. After verification of id, the model sends an HTTP response which is finally parsed by the Node-Red application and the result is displayed on the user screen.

3.2 HARDWARE/SOFTWARE DESIGNING

The steps follow for hardware/software designing are as follows:

1. Create an IBM Cloud account
2. Create necessary IBM Cloud services
3. Create Watson studio project
4. Configure Watson Studio
5. Create IBM Machine Learning instance
6. Import data for training as well as testing for model from Kaggle
7. Create machine learning model (either use Jupiter notebook or AutoAI)
8. Deploy the machine learning model
9. Create flow and configure node
10. Integrate node red with machine learning model
11. Deploy and run Node Red app.

NOTE: You can also make an UI using Python Flask and deploy it on IBM Cloud.

CHAPTER 4 EXPERIMENTAL INVESTIGATIONS

This project is fundamentally designed to predicting the life expectancy of a human in any country. The primary requirement of the project is the suitable dataset which will aid the prediction. Thus, the data set has been taken from the WHO, who has provided the data itself, publicly. The machine learning model is trained on the basis of the data provided, such that it can predict the average lifespan of an individual in the coming years in any demographic location on Earth.

There are 21 factors which are taken into account for predicting the life expectancy of a country are as follows:

1. **Country**
2. **Status:** Developed or Developing status of the country.
3. **Year**
4. **Adult mortality:** Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population).
5. **Infant deaths:** Number of Infant Deaths per 1000 population.
6. **Alcohol:** Alcohol, recorded per capita (15+) consumption.
7. **Percentage Expenditure**: Expenditure on health as a percentage of Gross Domestic Product per capita (%).
8. **Hepatitis B**: Immunization coverage among 1-year-olds (%).
9. **Measles:** Number of reported cases per 1000 population.
10. **BMI:** Average Body Mass Index of entire population.
11. **Under-five deaths:** Number of under-five deaths per 1000 population.
12. **Polio:** Immunization coverage among 1-year-olds (%).
13. **Total expenditure:** General government expenditure on health as a percentage of total government expenditure (%).
14. **Diphtheria:** Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year olds (%).
15. **HIV/AIDS:** Deaths per 1 000 live births HIV/AIDS (0-4 years).
16. **GDP:** Gross Domestic Product per capita (in USD).
17. **Population:** Population of the country.
18. **Thinness 10-19 years:** Prevalence of thinness among children and adolescents for Age 10 to 19(%).
19. **Thinness 5-9 years:** Prevalence of thinness among children for Age 5 to 9(%).
20. **Income composition of resources:** Human Development Index in terms of income composition of resources (index ranging from 0 to 1).
21. **Schooling:** Number of years of schooling

Finding the most suitable algorithm: Random forest gives highest accuracy

Random Forest Regression

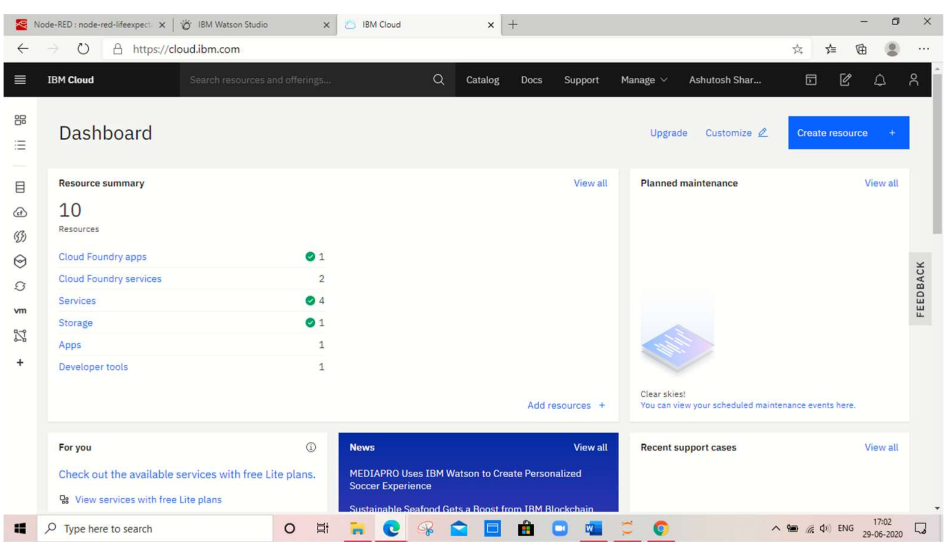
```
In [28]: rfr = RandomForestRegressor(n_estimators=1000,random_state=0)

In [29]: rfr_model = rfr.fit(xtrain,ytrain)

In [30]: random_forest_score = cross_val_score(rfr_model,xtrain,ytrain, cv = 5)

In [31]: rfr_pred = rfr.predict(xtest)

In [32]: print("mean cross validation score: %.2f" % np.mean(random_forest_score))
         print("score without cv: %.2f" % rfr_model.score(xtrain, ytrain))
         print("R^2 score on the test data %.2f" %r2_score(ytest, rfr_pred))

         mean cross validation score: 0.96
         score without cv: 0.99
         R^2 score on the test data 0.96
```

Steps

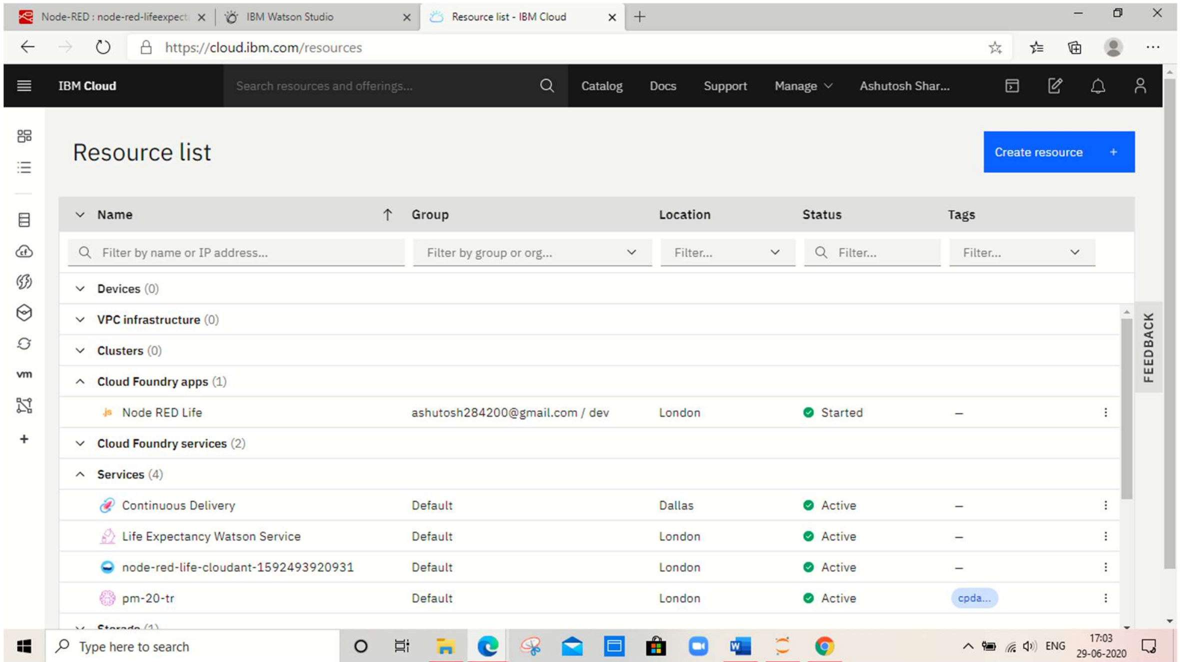
1. Create IBM Cloud services
   - Watson Studio
   - Watson Machine Learning
   - Node Red
2. Create Watson Studio service instance.
   - Select Catalog found at the top right of the page.
   - Click on Watson from the menu on the left, which you can find under Platform services.
   - Select Watson Studio.
   - Enter the Service name or keep the default value and make sure to select the US South as the region/location and your desired organization, and space.
   - Select Lite for the Plan, which you can find under Pricing Plans and is already selected. Please note you are only allowed one instance of a Lite plan per service.
   - Click on Create.
   - You will be taken to the main page of the service. Click on Get Started.
   - Create a New Project
3. Add WML service
   - Click on the Settings in the project view, locate Associated services => Add Service => Watson.
   - You should also create an Access Token in the project setting. Click on New token, give it a name, then click Create.
   - Create Notebook
   - Click Add to project => Notebook
   - And create your Model here.
   - Deploy Model as Web Service
4. Build Node-RED Flow to Integrate ML Services

SCREENSHOTS

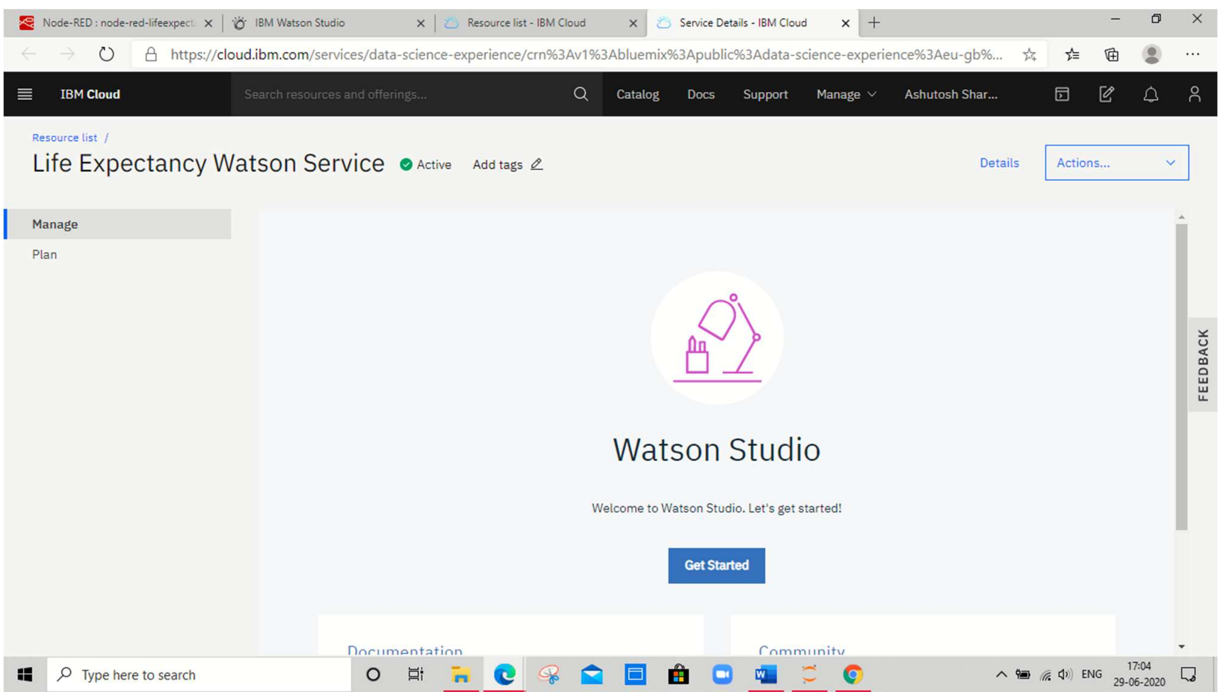
IBM CLOUD DASHBOARD:



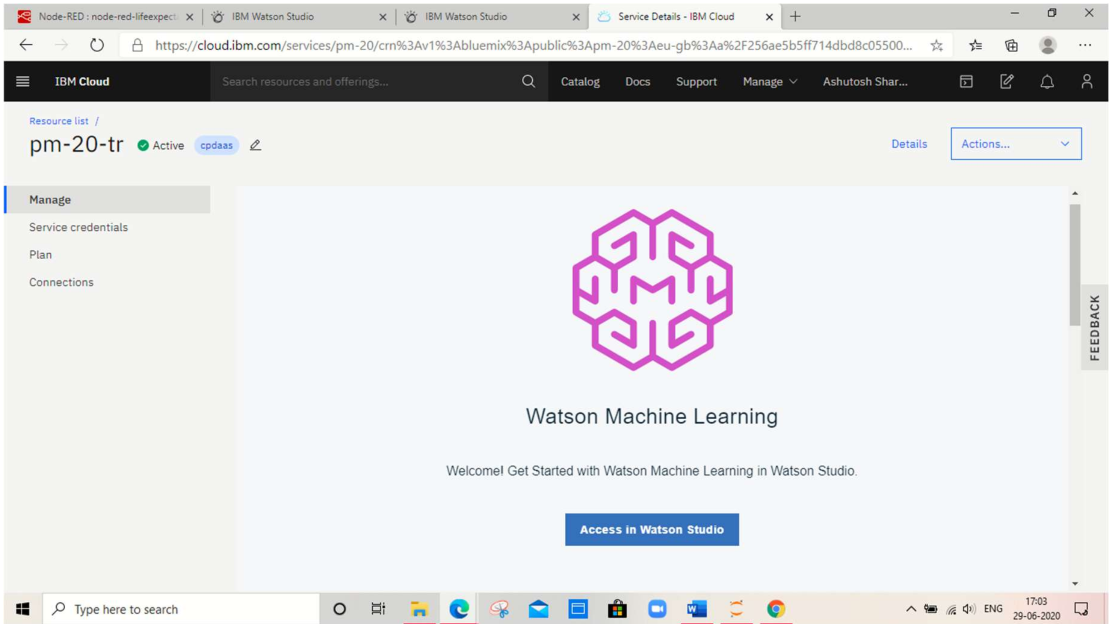
RESOURCE LIST:



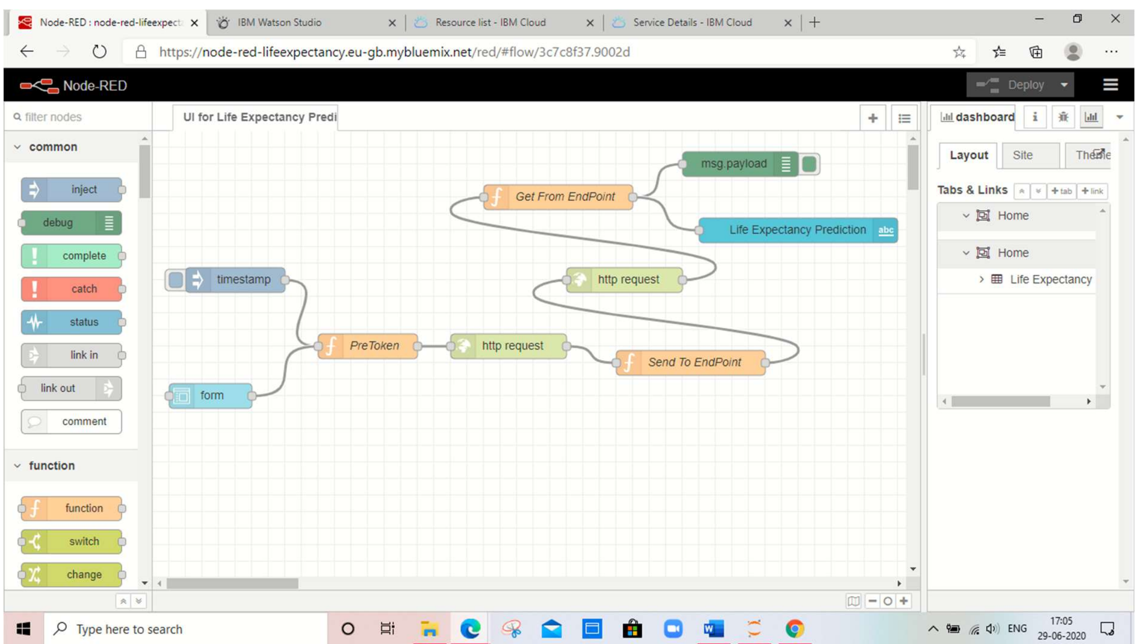
WATSON STUDIO:



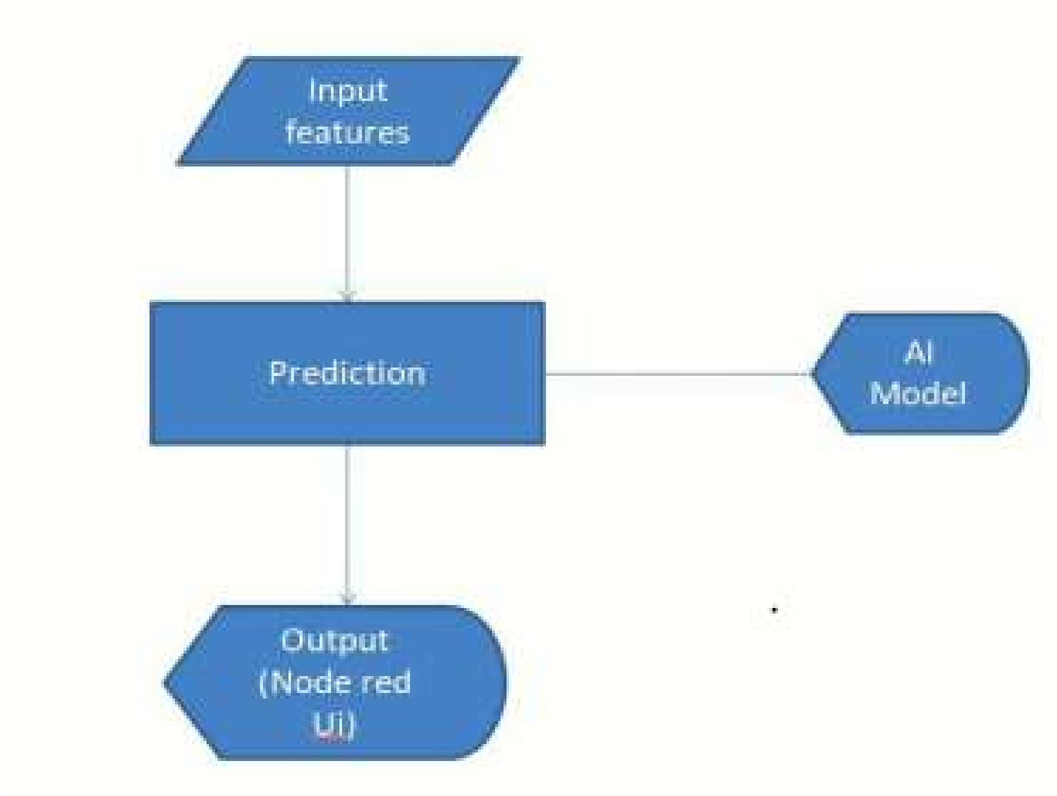
WATSON MACHINE LEARNING SERVICE:



NODE RED FLOW:

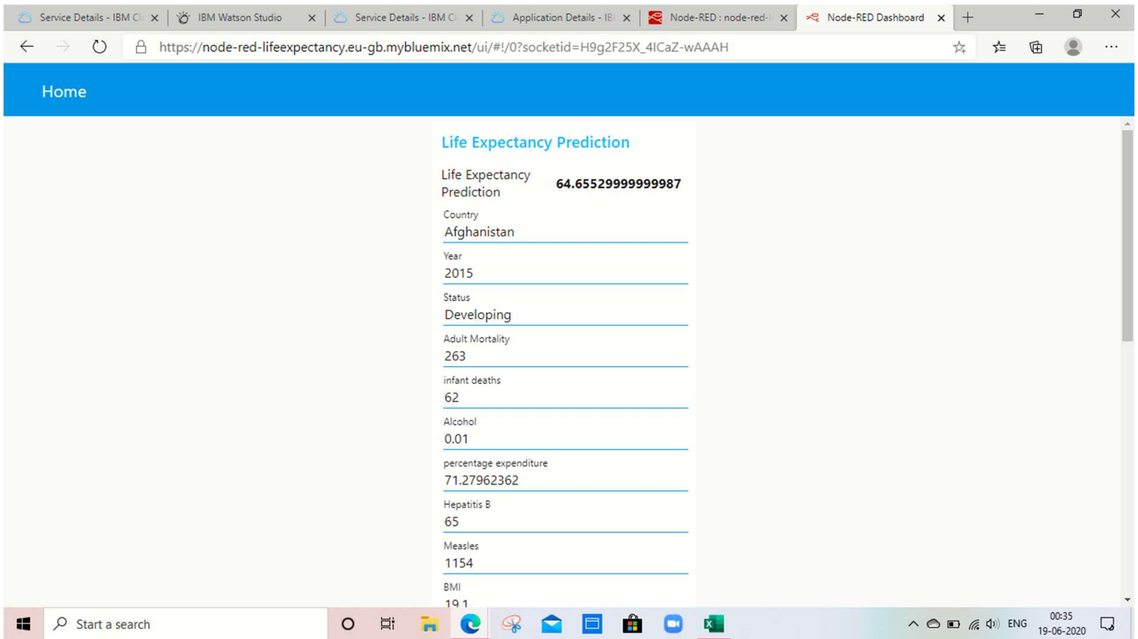


A flowchart is a diagram which depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. A flow chart helps improve understanding of what exactly is being implemented and how it takes different routes for different inputs and targets.



CHAPTER 6 RESULT

The model appears to the user in the form of an interface as shown below. The user has to fill the inputs and click on “Predict” button at the end of the form. On clicking the “Predict” button, the user will be displayed the predicted life expectancy, based on the inputs provided, at the top of the page as shown below. Once all the data is input by a user, the data is analysed by the Machine Learning model prepared using the service end point that which is given as a node in the Node-RED Flow. Data is run through the ML model and finally the predicted Life Expectancy is shown to the user, as shown below.



CHAPTER 7 ADVANTAGES & 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:**
   - Only available in English language (Limits areas of use)
   - Maintenance
   - Provides paid services
   - Doesn’t process structured data directly
   - Increasing rate of data, with limited resources
2. Requires high speed internet connection.

CHAPTER 8 APPLICATIONS

The application of predicting life expectancy are as following:

- This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.
- It will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy and can be used in various organization to improve the quality of service.
- The project can be used as a basis to develop personalized health applications.
- The governments can plan and develop their health infrastructures by keeping the most correlated factors in mind.
- The project can help governments to keep track of their country's health status so they can plan for the future accordingly.
- It can be used by researchers to make meaningful researches out of it and thus, bring about something that will help increase the expectancy consider the impact of a specific factor on the average lifespan of people in a specific country.
- Insurance companies consider age, lifestyle choices, family medical history, and several other factors when determining premium rates for individual life insurance policies. The principle of life expectancy suggests that you should purchase a life insurance policy for yourself and your spouse sooner rather than later.

CHAPTER 9 CONCLUSION

Thus, the developed model which is created by using IBM cloud services like IBM Watson studio, Watson machine learning and Node-Red services 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, Alcohol consumption, HIV, Hepatitis B, GDP, Percentage Expenditure and many more.

User can interact with the system via a simple user interface which is created by Node-red and the UI is simple just like a Google form which required an input to give you an output.

CHAPTER 10 FUTURE SCOPE

- Look at class within a particular country and see if these same factors are same in determining life expectancy for an individual.
- Use the Twitter API to incorporate NLP analysis for a country to see how it relates to Life Expectancy.
- 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.
- Mental Health versus Life Expectancy
- As more data comes, that can be fed to the model for more accurate predictions.
- Currently, the project is just a web application. It can be developed to support other platforms like Android, IOS and Windows Mobile.
- Other regression models can also be used for prediction and later the best among them should be chosen.

CHAPTER 11 BIBILOGRAPHY

- A Systematic Literature Review of Studies Analysing Inequalities in Health Expectancy among the Older Population (Benedetta Pongiglione, Bianca L. De Stavola, George B. Ploubidis)
- https://cloud.ibm.com/
- IBM Developer, "IBM Watson Studio: Create a project", 2019. [Online]. Available: https://www.youtube.com/watch?v=CUi8GezG1I&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L&index=2
- IBM Developer, "IBM Watson Studio: Jupyter notebook basics", 2019 [Online].Available: https://www.youtube.com/watch?v=Jtej3Y6uUng
- IBM Cloud setup [Online]. Available: https://www.ibm.com/cloud/get-started.
- IBM Developer, "Node-RED starter tutorial" [Online]. Available: https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/.
- "Node-RED labs on the use of the Watson Developer Cloud services – Watson developer-cloud/node-red-labs." [Online]. Available: https://github.com/watsondeveloper-cloud/node-red-labs
- Infuse AI into your applications with Watson AI to make more accurate predictions". [Online]. Available: https://www.ibm.com/watson/products-services.
- IBM Watson, "Intro to IBM Watson", 2018 [Online]. Available: https://www.youtube.com/watch?v=W3iPbFTAAds&feature=youtu.be
- "Get an understanding of the principles of machine learning." [Online]. Available: https://developer.ibm.com/technologies/machine-learning/series/learning-pathmachine-learning-for-developers/.
- IBM Developer, "IBM Watson Machine Learning: Get Started in IBM Cloud", 2020[Online]. Available: https://www.youtube.com/watch?v=NmdjtezQMSM.
- Watson Studio Workshop, "Chapter 4 Build and Deploy models in Jupyter Notebooks" [Online]. Available: https://bookdown.org/caoying4work/watsonstudioworkshop/jn.html
- Kumar Rajarshi, "Life Expectancy (WHO) Statistical Analysis on factors influencing Life Expectancy", 2018. [Online]. Available: https://www.kaggle.com/kumarajarshi/lifeexpectancy-who
- IBM Developer, "IBM Watson: Sign up for Watson Studio and Watson Knowledge Catalog", 2019. [Online]. Available: https://www.youtube.com/watch?v=DBRGlAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L

APPENDIX

A. SOURCE CODE

1. Data Set

First 20 rows

1	Year	Status	Adult Mortality	infant deaths	Alcohol	percentage e	Hepatitis B	Measles	BMI	under-fiv	Polio	Total exp	Diphther	HIV/AIDS	GDP	Populatio	thinness	thinness	Income c	Schooling	Life expectancy
2	2015	Developing	263	62	0.01	71.27962362	65	1154	19.1	83	6	8.16	65	0.1	584.259	3.4E+07	17.2	17.3	0.479	10.1	65
3	2014	Developing	271	64	0.01	73.52358168	62	492	18.6	86	58	8.18	62	0.1	612.697	327582	17.5	17.5	0.476	10	59.9
4	2013	Developing	268	66	0.01	73.21924272	64	430	18.1	89	62	8.13	64	0.1	631.745	3.2E+07	17.7	17.7	0.47	9.9	59.9
5	2012	Developing	272	69	0.01	78.1842153	67	2787	17.6	93	67	8.52	67	0.1	669.959	3696958	17.9	18	0.463	9.8	59.5
6	2011	Developing	275	71	0.01	7.097108703	68	3013	17.2	97	68	7.87	68	0.1	63.5372	2978599	18.2	18.2	0.454	9.5	59.2
7	2010	Developing	279	74	0.01	79.67936736	66	1989	16.7	102	66	9.2	66	0.1	553.329	2883167	18.4	18.4	0.448	9.2	58.8
8	2009	Developing	281	77	0.01	56.76221682	63	2861	16.2	106	63	9.42	63	0.1	445.893	284331	18.6	18.7	0.434	8.9	58.6
9	2008	Developing	287	80	0.03	25.87392536	64	1599	15.7	110	64	8.33	64	0.1	373.361	2729431	18.8	18.9	0.433	8.7	58.1
10	2007	Developing	295	82	0.02	10.91015598	63	1141	15.2	113	63	6.73	63	0.1	369.836	2.7E+07	19	19.1	0.415	8.4	57.5
11	2006	Developing	295	84	0.03	17.17151751	64	1990	14.7	116	58	7.43	58	0.1	272.564	2589345	19.2	19.3	0.405	8.1	57.3
12	2005	Developing	291	85	0.02	1.388647732	66	1296	14.2	118	58	8.7	58	0.1	25.2941	257798	19.3	19.5	0.396	7.9	57.3
13	2004	Developing	293	87	0.02	15.29606643	67	466	13.8	120	5	8.79	5	0.1	219.141	2.4E+07	19.5	19.7	0.381	6.8	57
14	2003	Developing	295	87	0.01	11.08905273	65	798	13.4	122	41	8.82	41	0.1	198.729	2364851	19.7	19.9	0.373	6.5	56.7
15	2002	Developing	3	88	0.01	16.88735091	64	2486	13	122	36	7.76	36	0.1	187.846	2.2E+07	19.9	2.2	0.341	6.2	56.2
16	2001	Developing	316	88	0.01	10.5747282	63	8762	12.6	122	35	7.8	33	0.1	117.497	2966463	2.1	2.4	0.34	5.9	55.3
17	2000	Developing	321	88	0.01	10.42496	62	6532	12.2	122	24	8.2	24	0.1	114.56	293756	2.3	2.5	0.338	5.5	54.8
18	2015	Developing	74	0	4.6	364.9752287	99	0	58	0	99	6	99	0.1	3954.23	28873	1.2	1.3	0.762	14.2	77.8
19	2014	Developing	8	0	4.51	428.7490668	98	0	57.2	1	98	5.88	98	0.1	4575.76	288914	1.2	1.3	0.761	14.2	77.5
20	2013	Developing	84	0	4.76	430.8769785	99	0	56.5	1	99	5.66	99	0.1	4414.72	289592	1.3	1.4	0.759	14.2	77.2

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

2. Watson Studio

- **Life Expectancy Code: -**

Loading packages¶

In [1]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
from sklearn.preprocessing import LabelEncoder
```

Importing data

In [2]:

```
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_b0ddd268e7ab4b6f853921db0bb6d9c8 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='sdPkjBgs0xTat05fo5BEdMQPp_eWT9_RgP_bEwyji33K',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
```

```
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

body = client_b0ddd268e7ab4b6f853921db0bb6d9c8.get_object(Bucket='predictinglifeexpectancy-donotdelete-pr-c2fohwcni1ybca',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.head()
```

Out[2]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AI
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1

5 rows × 22 columns

In [3]:

```
df.shape
```

Out[3]:

(2938, 22)

In [4]:

```
df.describe()
```

Out[4]:

	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 [5]:

```
df.describe(include='all')
```

Out[5]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	
count	2938	2938.000000	2938	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000	2938.000000	...	29
unique	193	NaN	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
top	Central African Republic	NaN	Developing	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
freq	16	NaN	2426	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
mean	NaN	2007.518720	NaN	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461	2419.592240	...	82
std	NaN	4.613841	NaN	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016	11467.272489	...	23
min	NaN	2000.000000	NaN	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	...	3.(
25%	NaN	2004.000000	NaN	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000	0.000000	...	78
50%	NaN	2008.000000	NaN	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000	17.000000	...	93
75%	NaN	2012.000000	NaN	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000	360.250000	...	97
max	NaN	2015.000000	NaN	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	...	99

11 rows × 22 columns

In [6]:

```
pd.isnull(df).sum()
```

Out[6]:

```
Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
```

In [7]:

```
temp=pd.DataFrame(index=df.columns)
temp['data_types']=df.dtypes
temp['null_count']=df.isnull().sum()
temp['unique_count']=df.nunique()
```

In [8]:

```
temp
```

Out[8]:

	data_types	null_count	unique_count
Country	object	0	193
Year	int64	0	16
Status	object	0	2
Life expectancy	float64	10	362
Adult Mortality	float64	10	425
infant deaths	int64	0	209
Alcohol	float64	194	1076
percentage expenditure	float64	0	2328
Hepatitis B	float64	553	87
Measles	int64	0	958
BMI	float64	34	608
under-five deaths	int64	0	252
Polio	float64	19	73
Total expenditure	float64	226	818
Diphtheria	float64	19	81
HIV/AIDS	float64	0	200
GDP	float64	448	2490
Population	float64	652	2278
thinness 1-19 years	float64	34	200
thinness 5-9 years	float64	34	207
Income composition of resources	float64	167	625
Schooling	float64	163	173

In [9]:

```
# Developing:1
# Developed:0
le = LabelEncoder()
df['Status'] = le.fit_transform(df['Status'])
```

EDA

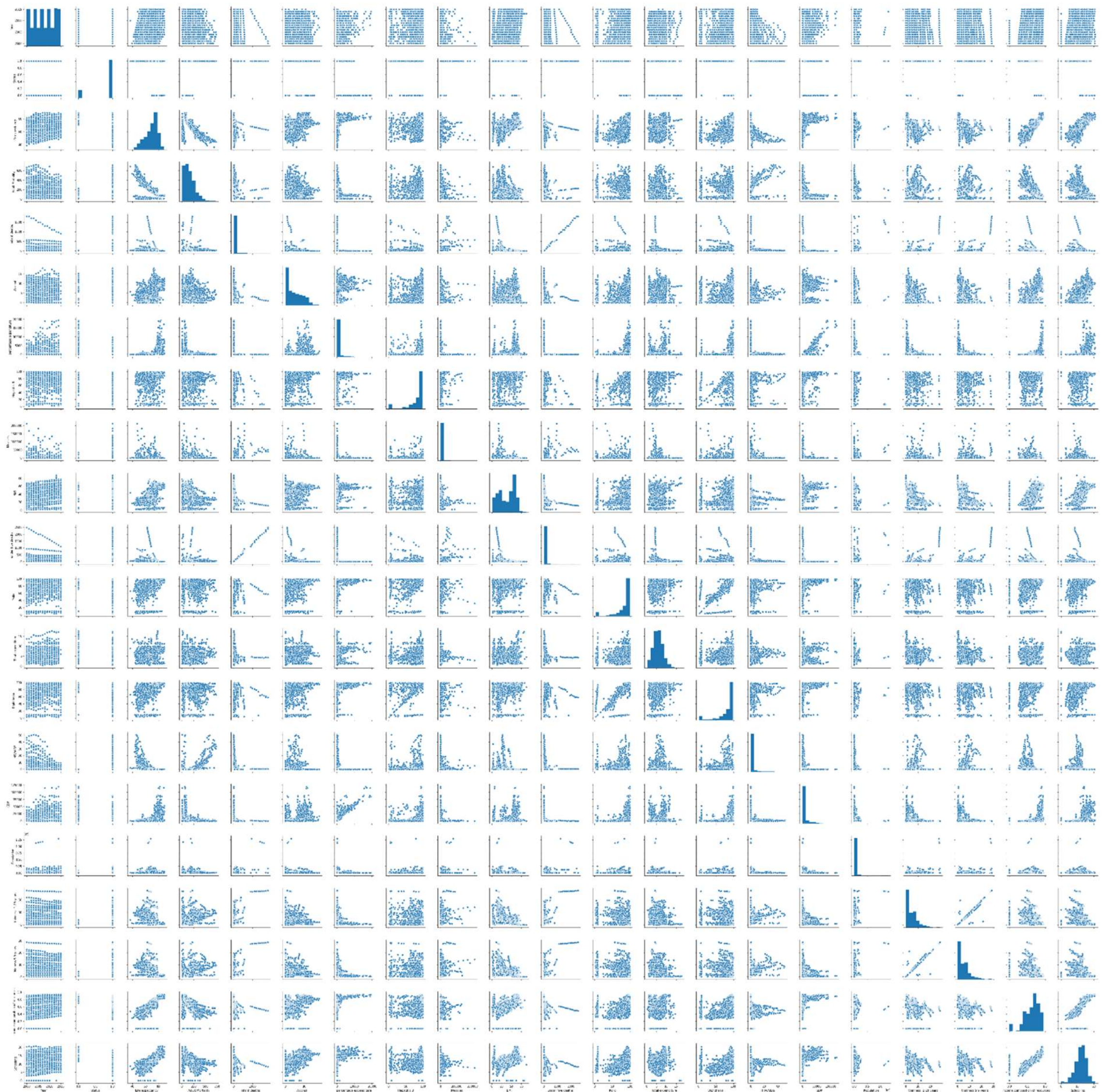
In [10]:

```
sns.pairplot(df)
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/numpy/lib/histograms.py:754: RuntimeWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
/opt/conda/envs/Python36/lib/python3.6/site-packages/numpy/lib/histograms.py:755: RuntimeWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

Out[10]:

```
<seaborn.axisgrid.PairGrid at 0x7f7ec00ca2b0>
```

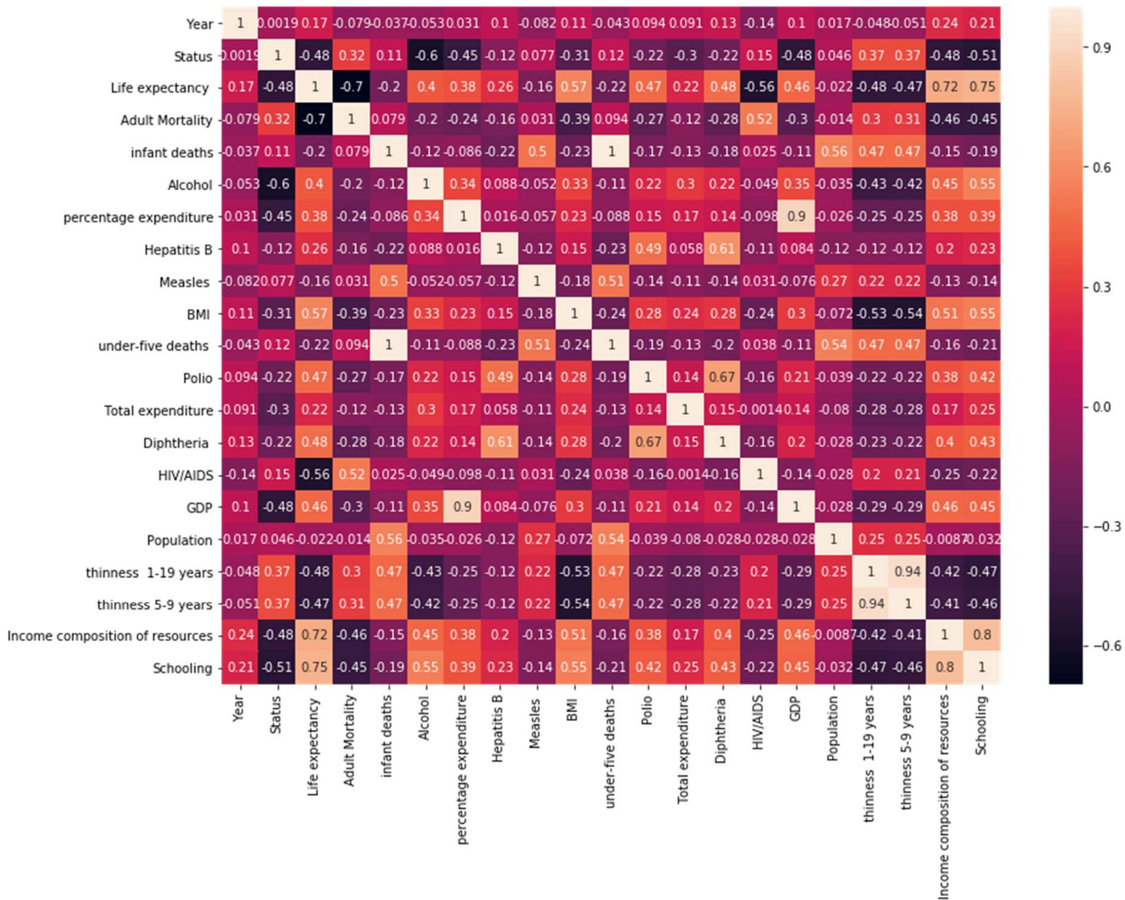


In [11]:

```
plt.figure(figsize = (14, 10))
sns.heatmap(df.corr(),annot=True)
```

Out[11]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f7eb2359400>



Preprocessing the data

In [12]:

```
country_list = df.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']
```

In [13]:

```
for country in country_list:
    df.loc[df['Country'] == country,fill_list] = df.loc[df['Country'] == country,fill_list].interpolate()

df=df.dropna()
```

In [14]:

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

Out[14]:

	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	
0	2015	1	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	...	6.0	8.16	65.0	0.1	584.25
1	2014	1	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	...	58.0	8.18	62.0	0.1	612.69
2	2013	1	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	...	62.0	8.13	64.0	0.1	631.74
3	2012	1	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	...	67.0	8.52	67.0	0.1	669.95
4	2011	1	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	...	68.0	7.87	68.0	0.1	63.537

5 rows × 21 columns

In [15]:

```
# Divide the dataset into dependent and independent variables
# x:features and y:labels
x = df.drop(['Life expectancy '],axis = 1)
y = df['Life expectancy ']
```

In [16]:

```
x.shape,y.shape
```

Out[16]:

((1987, 20), (1987,))

In [17]:

```
# Splitting the data into Train and Validation set
xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.2, random_state=0)
```

In [18]:

```
xtrain.shape,ytrain.shape
```

Out[18]:

((1589, 20), (1589,))

In [19]:

```
xtest.shape,ytest.shape
```

Out[19]:

((398, 20), (398,))

Linear Regression

In [20]:

```
lr = LinearRegression()
lr.fit(xtrain,ytrain)
```

Out[20]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [21]:

```
print('R_square score on the training: %.2f' % lr.score(xtrain,ytrain))
```

R_square score on the training: 0.84

In [22]:

```
lr_pred = lr.predict(xtest)
```

In [23]:

```
print("Mean squared error: %.2f" % mean_squared_error(ytest,lr_pred))
print("Mean absolute error: %.2f" % mean_absolute_error(ytest, lr_pred))
print('R_square score: %.2f' % r2_score(ytest, lr_pred))
```

```
Mean squared error: 16.04
Mean absolute error: 3.04
R_square score: 0.82
```

Decision Tree Regression

In [24]:

```
dtr = DecisionTreeRegressor()
dtr_model = dtr.fit(xtrain,ytrain)
```

In [25]:

```
decision_tree_score = cross_val_score(dtr_model,xtrain,ytrain,cv=5)
```

In [26]:

```
dtr_pred = dtr.predict(xtest)
```

In [27]:

```
print("mean cross validation score: %.2f"  % np.mean(decision_tree_score))
print("score without cv: %.2f" % dtr_model.score(xtrain,ytrain))
print("R^2 score on the test data %.2f"% r2_score(ytest,dtr_pred))
```

```
mean cross validation score: 0.91
score without cv: 1.00
R^2 score on the test data 0.91
```

Random Forest Regression

In [28]:

```
rfr = RandomForestRegressor(n_estimators=1000,random_state=0)
```

In [29]:

```
rfr_model = rfr.fit(xtrain,ytrain)
```

In [30]:

```
random_forest_score = cross_val_score(rfr_model,xtrain,ytrain, cv = 5)
```

In [31]:

```
rfr_pred = rfr.predict(xtest)
```

In [32]:

```
print("mean cross validation score: %.2f" % np.mean(random_forest_score))
print("score without cv: %.2f" % rfr_model.score(xtrain, ytrain))
print("R^2 score on the test data %.2f" %r2_score(ytest, rfr_pred))
```

```
mean cross validation score: 0.96
score without cv: 0.99
R^2 score on the test data 0.96
```

In [33]:

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

In [34]:

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

In [35]:

```
wml_credentials = {
  "apikey": "************************",
  "iam_apikey_description": "**************************************",
  "iam_apikey_name": "Service credentials-1",
  "iam_role_crn": "*****************************",
  "iam_serviceid_crn": "**********************************************",
  "instance_id": "****************************************",
  "url": "****************************"
}
```

In [36]:

```
client = WatsonMachineLearningAPIClient(wml_credentials)
```

In [37]:

```
metadata = {
   client.repository.ModelMetaNames.AUTHOR_NAME : "Ashutosh Sharma",
   client.repository.ModelMetaNames.AUTHOR_EMAIL : "ashutosh284200@gmail.com",
   client.repository.ModelMetaNames.NAME : "LifeExpectancyPrediction"
}
```

In [38]:

```
stored_data = client.repository.store_model(rfr,meta_props=metadata)
```

In [39]:

```
model_uid = client.repository.get_model_uid(stored_data)
```

In [40]:

```
# Model deployment
deploy = client.deployments.create(model_uid)
```

In [41]:

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

In [42]:

```
scoring_endpoint
```

- **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]]

3. **Node Red**

- **Flows.json**

[{"id":"3c7c8f37.9002d","type":"tab","label":"UI for Life Expectancy Prediction","disabled":false,"info":""},{"id":"c33ae67b.11da38","type":"function","z":"3c7c8f37.9002d","name":"PreToken","func":"// global.set(\"c_name\",msg.payload.c_name)\nglobal.set(\"ye\",msg.payload.ye)\nglobal.set(\"status\",msg.payload.stat)\nglobal.set(\"adult_mort\",msg.payload.adult_mort)\nglobal.set(\"in_death\",msg.payload.in_death)\nglobal.set(\"alcohol\",msg.payload.alcohol)\nglobal.set(\"per_expen\",msg.payload.per_expen)\nglobal.set(\"hepa_b\",msg.payload.hepa_b)\nglobal.set(\"measles\",msg.payload.measles)\nglobal.set(\"bmi\",msg.payload.bmi)\nglobal.set(\"un_five_death\",msg.payload.un_five_death)\nglobal.set(\"polio\",msg.payload.polio)\nglobal.set(\"total_expen\",msg.payload.total_expen)\nglobal.set(\"diphth\",msg.payload.diphth)\nglobal.set(\"hiv\",msg.payload.hiv)\nglobal.set(\"gdp\",msg.payload.gdp)\nglobal.set(\"population\",msg.payload.population)\nglobal.set(\"thin_19\",msg.payload.thin_19)\nglobal.set(\"thin_9\",msg.payload.thin_9)\nglobal.set(\"income_comp\",msg.payload.income_comp)\nglobal.set(\"schooling\",msg.payload.schooling)\n\nvar apikey=\"Q5OTnXXpHC_NAFPq0PQFGrg1AV4VeS7hUHr80lsPgTo1\";\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":260,"y":260,"wires":[["9d3b48c5.5d80e8"]]},{"id":"9d3b48c5.5d80e8","type":"http request","z":"3c7c8f37.9002d","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"","x":430,"y":260,"wires":[["9111ba94.358e98"]]},{"id":"6dcc1a68.2d6bd4","type":"inject","z":"3c7c8f37.9002d","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":100,"y":180,"wires":[["c33ae67b.11da38"]]},{"id":"b53e9c8b.de2df","type":"debug","z":"3c7c8f37.9002d","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":710,"y":40,"wires":[]},{"id":"9111ba94.358e98","type":"function","z":"3c7c8f37.9002d","name":"Send To EndPoint","func":"var token=msg.payload.access_token\nvar instance_id=\"28607ee8-f59c-42a8-87e7-5941b3198461\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance_id}\n\n// var c_name = global.get('c_name')\nvar ye = global.get('ye')\nvar status = global.get('status')\nvar adult_mort = global.get('adult_mort')\nvar in_death = global.get('in_death')\nvar alcohol = global.get('alcohol')\nvar per_expen = global.get('per_expen')\nvar hepa_b = global.get('hepa_b')\nvar measles = global.get('measles')\nvar bmi = global.get('bmi')\nvar un_five_death = global.get('un_five_death')\nvar polio = global.get('polio')\nvar total_expen = global.get('total_expen')\nvar diphth = global.get('diphth')\nvar hiv = global.get('hiv')\nvar gdp = global.get('gdp')\nvar population = global.get('population')\nvar thin_19 = global.get('thin_19')\nvar thin_9 = global.get('thin_9')\nvar income_comp = global.get('income_comp')\nvar schooling = global.get('schooling')\n\nmsg.payload={\"fields\": \n['Year', 'Status', 'Adult Mortality', 'infant deaths', \n'Alcohol', 'percentage expenditure', 'Hepatitis B', 'Measles ', \n' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure',\n'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population', ' thinness 1-19 years',\n' thinness 5-9 years', 'Income composition of resources', 'Schooling'],\n\"values\":\n[[ye, status, adult_mort, in_death, alcohol, \nper_expen,hepa_b, measles, bmi, un_five_death,polio,

\ntotal_expen,diphth,hiv,gdp,population,thin_19,
\nthin_9,income_comp,schooling]]\n}\nreturn
msg;","outputs":1,"noerr":0,"x":650,"y":280,"wires":[["d8502ff9.25e77"]]},{"id":"d8502ff9.25e77","type":"http
request","z":"3c7c8f37.9002d","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/28607ee8-f59c-42a8-87e7-5941b3198461/deployments/d4c61f56-0f6e-4f02-b21c-53f34fb3e563/online","tls":"","persist":false,"proxy":"","authType":"","x":570,"y":180,"wires":[["7706e593.4602cc"]]},{"id":"7706e593.4602cc","type":"function","z":"3c7c8f37.9002d","name":"Get From
EndPoint","func":"msg.payload=msg.payload.values[0][0];\nreturn
msg;\n","outputs":1,"noerr":0,"x":490,"y":80,"wires":[["b53e9c8b.de2df","b4a0fecc.5c50c"]]},{"id":"af8c088.67ea2f8","type":"ui_form","z":"3c7c8f37.9002d","name":"","label":"","group":"2ab45756.fcf838","order":0,"width":0,"height":0,"options":[{"label":"Country","value":"name","type":"text","required":false,"rows":null},{"label":"Year","value":"ye","type":"number","required":false,"rows":null},{"label":"Status","value":"stat","type":"text","required":false,"rows":null},{"label":"Adult
Mortality","value":"adult_mort","type":"number","required":false,"rows":null},{"label":"infant
deaths","value":"in_death","type":"number","required":false,"rows":null},{"label":"Alcohol","value":"alcohol","type":"number","required":false,"rows":null},{"label":"percentage
expenditure","value":"per_expen","type":"number","required":false,"rows":null},{"label":"Hepatitis
B","value":"hepa_b","type":"number","required":false,"rows":null},{"label":"Measles ","value":"measles","type":"number","required":false,"rows":null},{"label":" BMI ","value":"bmi","type":"number","required":false,"rows":null},{"label":"under-five deaths
","value":"un_five_death","type":"number","required":false,"rows":null},{"label":"Polio","value":"polio","type":"number","required":false,"rows":null},{"label":"Total
expenditure","value":"total_expen","type":"number","required":false,"rows":null},{"label":"Diphtheria
","value":"diphth","type":"number","required":false,"rows":null},{"label":"
HIV/AIDS","value":"hiv","type":"number","required":false,"rows":null},{"label":"GDP","value":"gdp","type":"number","required":false,"rows":null},{"label":"Population","value":"population","type":"number","required":false,"rows":null},{"label":"
thinness 1-19
years","value":"thin_19","type":"number","required":false,"rows":null},{"label":"
thinness 5-9
years","value":"thin_9","type":"number","required":false,"rows":null},{"label":"Income composition of
resources","value":"income_comp","type":"number","required":false,"rows":null},{"label":"Schooling","value":"schooling","type":"number","required":false,"rows":null}],"formValue":{"name":"","ye":"","stat":"","adult_mort":"","in_death":"","alcohol":"","per_expen":"","hepa_b":"","measles":"","bmi":"","un_five_death":"","polio":"","total_expen":"","diphth":"","hiv":"","gdp":"","population":"","thin_19":"","thin_9":"","income_comp":"","schooling":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":70,"y":320,"wires":[["c33ae67b.11da38"]]},{"id":"b4a0fecc.5c50c","type":"ui_text","z":"3c7c8f37.9002d","group":"2ab45756.fcf838","order":1,"width":0,"height":0,"name":"","label":"Life Expectancy
Prediction","format":"{{msg.payload}}","layout":"row-

spread","x":780,"y":120,"wires":[]},{"id":"2ab45756.fcf838","type":"ui_group","z":"","name":"Life Expectancy Prediction","tab":"c777de86.27fd3","order":1,"disp":true,"width":"6","collapse":false},{"id":"c777de86.27fd3","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]

Node-Red App Link:
https://noderedlifeexpectancy.eugb.mybluemix.net/ui/#!/0?socketid=1M7RNpEfdRZx8JywAAAB

GitHub Link: https://github.com/SmartPracticeschool/llSPS-INT-2835-Predicting-Life-Expectancy-using-Machine-Learning