A Project Report

On

Predicting Life Expectancy using Machine Learning

Internship

Under:

TheSMARTBRIDGE

NAME : Saurabh Kumar

EMAIL : sauryaprasad0697@gmail.com

PROJECT ID : SPS_PRO_215

INTERNSHIP TITLE : Predicting Life Expectancy using Machine Learning - SB12352
Category: Machine Learning

Table Of Content

DESCRIPTION	**PAGE NO**
1. INTRODUCTION	
1.1. Overview	2
1.2. Purpose	3
2. LITERATURE SURVEY	
2.1 Existing Problem	4
2.2 Proposed solution	4
3. THEORITICAL ANALYSIS	
2.3 Block Diagram	5
2.4 Hardware/Software design	5
4. EXPERIMENTAL INVESTIGATION	6
SCREENSHOTS	8
5. FLOWCHART	11
6. RESULT	12
7. ADVANTAGES & DISADVANTAGES	13
8. APPLICATION	15
9. CONCLUSION	15
10. FUTURE SCOPE	15
8. BIBILIOGRAPHY	16
APPENDIX	
A. SOURCE CODE	17

1. INTRODUCTION

1.1. Overview

This project “Predicting Life Expectancy using Machine Learning” is an web application that predict the expected average life span of people of a given country based on various features. This project is built using IBM services(Watson studio, Node Red, Watson machine learning).

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.

Life expectancy is a statistical measure of the average time a human being is expected to live, Life expectancy depends on various factors: Country, Status, infant deaths, GDP, Population, BMI, other factors. 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.

- Project Requirements: IBM Cloud, IBM Watson, Node- RED
- Functional Requirements: IBM cloud
- Technical Requirements: WATSON Machine Learning
- Software Requirements: Python, Watson Studio, Node-Red
- Project Deliverables: Smartinternz Internship
- Project Team: Saurabh Kumar
- Project Duration:28 days

1.2. Purpose:

Life expectancy is the most important factor for 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.

Scope of Work:

- This project is an end-to end project which will formulate a model while considering data from a period of 2000 to 2015 for all the countries.
- This project will predict the average time a human being is expected to live based on some factors .
- A country can predict the expected life of their citizens.According to that , the   country can take necessary preventive measures to improve the healthcare system.
- This will serve as an example for countries to assess to improve life expectancy for their citizens.
- This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

2. LITERATURE SURVEY

2.1 Existing problem:

In our regular prediction system, there are many problems exist, such as :

○ whole concept of life expectancy depends on the interpretation given to “full health”.

- ○ Or the factors used to predict the life expectancy of people are based on some associated specific features of particular fields like :
- ○ morbidity and mortality (smoking, alcohol consumption, overweight and obesity, and physical activity)
- ○ Health related disease
- ○ occupational or social class, area level deprivation, geographical area of residence
(urban and rural), housing
tenure ○ Race-based
inequalities.

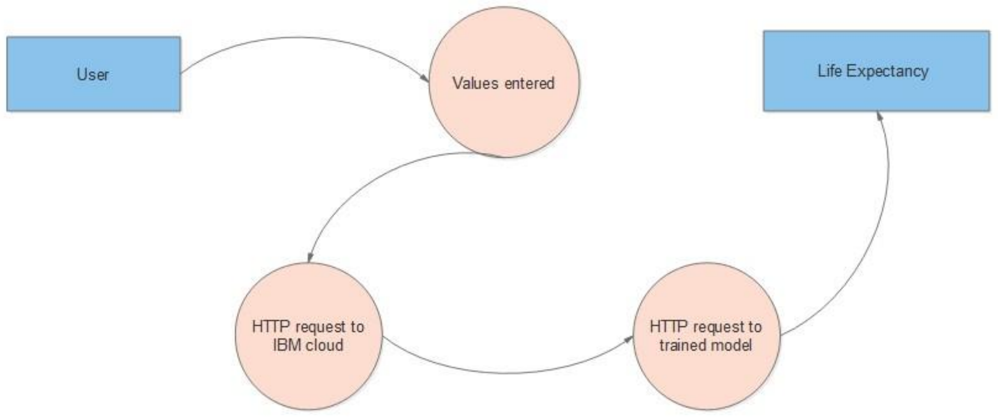
Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that effect of immunization and human development index was not taken into account in the past. Also, some of the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

2.2 Proposed solution:

- ✓ For the above problem to get solved we have a dataset consist of various factors .In this system we have taken all the correlated features into consideration. So the target output variable i.e expected life span of the people depends upon variety of factors and not factors of particular fields.
- ✓ Important immunization like Hepatitis B, Polio and Diphtheria are also considered.
- ✓ The data-set related to life expectancy, health factors for 193 countries has been collected from WHO data repository website and its corresponding economic data was collected from the United Nations website. Among all categories of health-related factors only those critical factors were chosen which are more representative. It has been observed that in the past 15 years, there has been a huge development in health sector resulting in improvement of human mortality rates especially in the developing nations in comparison to the past 30 years. Therefore, in this project we have considered data from year 2000-2015 for 193 countries for further analysis. The individual data files have been merged together into a single data-set.
- ✓ The project uses immunization factors, mortality factors, economic factors, social factors and other health related factors to predict life expectancy of a country for a given year using a machine learning model.
- ✓ Since the observations in 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. This will help in suggesting a country, which area should be given importance in order to efficiently improve the life expectancy of its population.

3.THEORITICAL ANALYSIS

3.1 Block/Flow Diagram:



3.2 Hardware / Software designing:

1. Create necessary IBM Cloud services
2. Create Watson studio project
3. Configure Watson Studio
4. Create IBM Machine Learning instance
5. Create machine learning model in Jupyter notebook
6. Deploy the machine learning model
7. Create flow and configure node
8. Integrate node red with machine learning model
9. Deploy and run Node Red app.

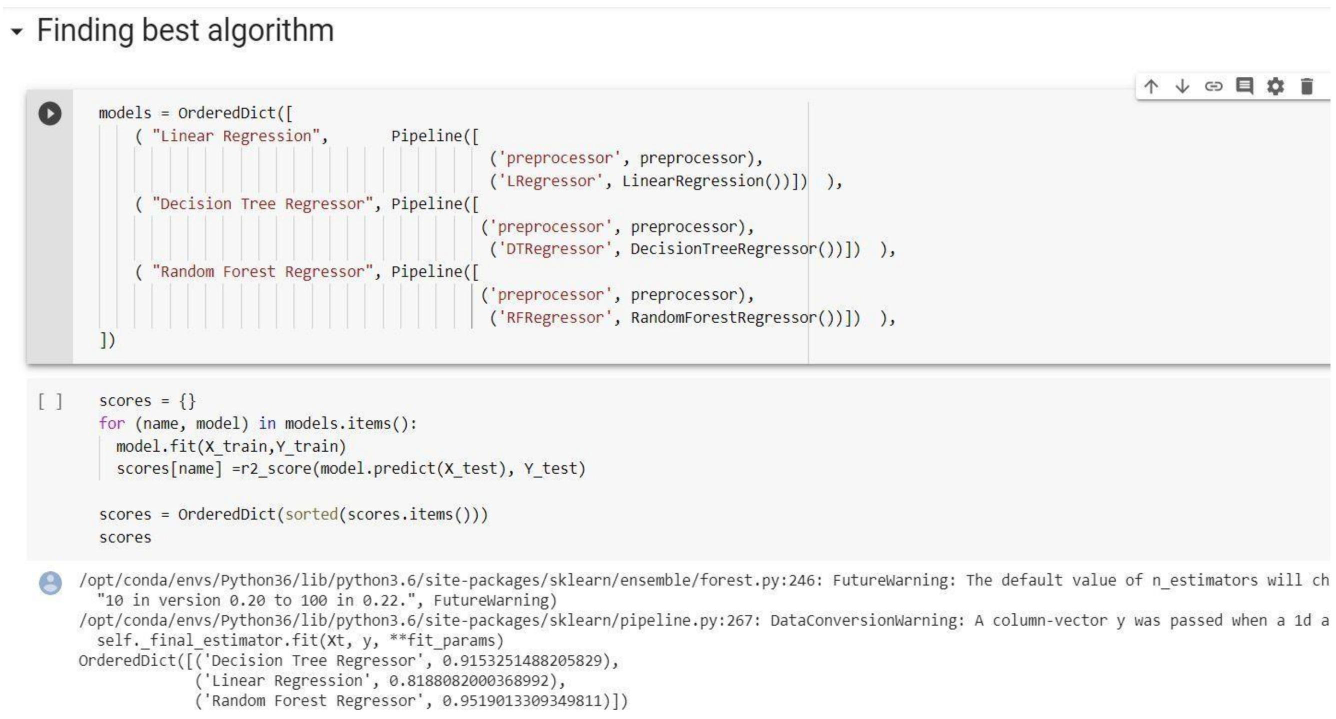
Input is taken from the user using a “Form” element in Node-Red. Then, an HTTP request is made to the IBM cloud that further makes an HTTP request to the deployed model using model’s instance id. After verification of id, the model sends an HTTP response which is finally parsed by the Node-Red application and the result is displayed on the user screen.

4. EXPERIMENTAL INVESTIGATIONS

Following factors are taken into account for predicting the life expectancy of a country.

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: Hepatitis B =immunization coverage among 1-year-olds (%).
9. Measles: 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: Polio (Pol3) 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-yearolds (%).
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



Finding best algorithm

```
models = OrderedDict([
    ( "Linear Regression",       Pipeline([
                                             ('preprocessor', preprocessor),
                                             ('LRegressor', LinearRegression())])  ),
    ( "Decision Tree Regressor", Pipeline([
                                            ('preprocessor', preprocessor),
                                             ('DTRegressor', DecisionTreeRegressor())])  ),
    ( "Random Forest Regressor", Pipeline([
                                            ('preprocessor', preprocessor),
                                             ('RFRegressor', RandomForestRegressor())])  ),
])
```

```
scores = {}
for (name, model) in models.items():
  model.fit(X_train,Y_train)
  scores[name] =r2_score(model.predict(X_test), Y_test)

scores = OrderedDict(sorted(scores.items()))
scores
```

```
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/ensemble/forest.py:246: FutureWarning: The default value of n_estimators will ch
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/pipeline.py:267: DataConversionWarning: A column-vector y was passed when a 1d a
  self._final_estimator.fit(Xt, y, **fit_params)
OrderedDict([('Decision Tree Regressor', 0.9153251488205829),
             ('Linear Regression', 0.8188082000368992),
             ('Random Forest Regressor', 0.9519013309349811)])
```

Steps

Create IBM Cloud services

- Watson Studio
- Watson Machine Learning
- Node Red

❖ 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

❖ Add WML service ○ Click on the **Settings** in the project view, locate **Associated services** => **Add Service** => **Watson**.

○ You should also create a **Access Token** in the project setting. Click on **New token**, give it a name, then click **Create**.

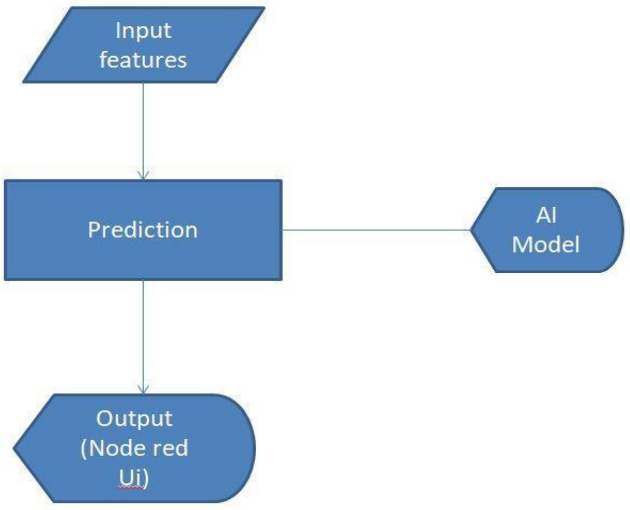
- Create Notebook
  Click **Add to project** => **Notebook**
  And create your Model here.

- Deploy Model as Web Service

- Build Node-RED Flow To Integrate ML Services

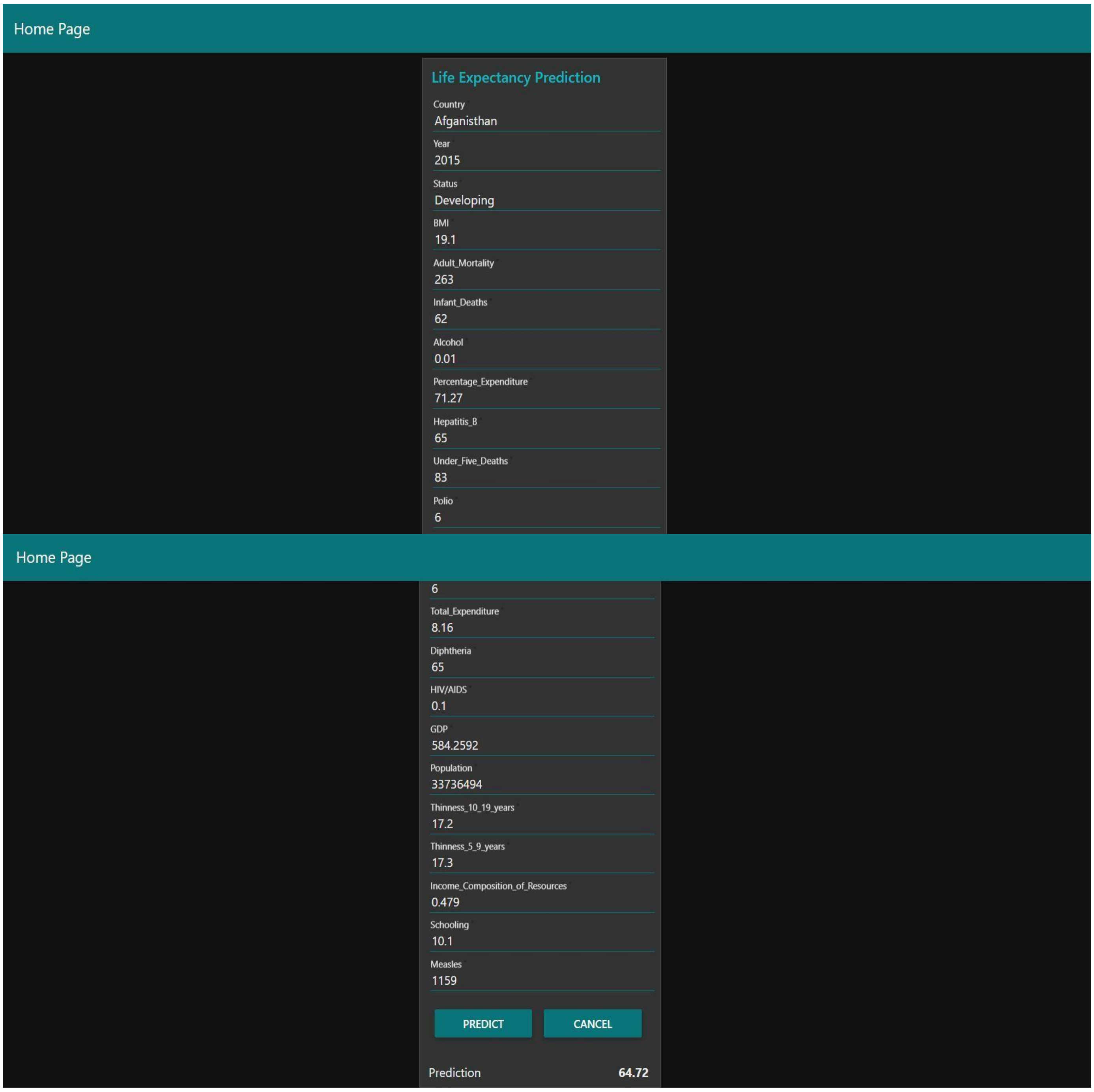
WATSON STUDIO :

5.FLOWCHART



6.RESULTS

Finally our Node-RED dash board integrates all the components and displayed in the Dashboard UI by typing URL-https://node-red-aqwyb.eu-gb.mybluemix.net/ui/#!/0?socketid=MjrMCRV9PVZTKowoAAAD in browser.


Home Page

Life Expectancy Prediction

Country
Afganisthan

Year
2015

Status
Developing

BMI
19.1

Adult_Mortality
263

Infant_Deaths
62

Alcohol
0.01

Percentage_Expenditure
71.27

Hepatitis_B
65

Under_Five_Deaths
83

Polio
6

Home Page

6

Total_Expenditure
8.16

Diphtheria
65

HIV/AIDS
0.1

GDP
584.2592

Population
33736494

Thinness_10_19_years
17.2

Thinness_5_9_years
17.3

Income_Composition_of_Resources
0.479

Schooling
10.1

Measles
1159

Prediction 64.72

7.ADVANTAGES & DISADVANTAGES

Advantages :

1) 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.

2) The data-sets are made available to public for the purpose of health data analysis.
3) Can be used in any organization to analyze the data.
4) The observations in the dataset used are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country, which area should be given importance in order to efficiently improve the life expectancy of its population.
5) Some of the past research was done considering multiple linear regression based on data set of one year for all the countries. But the dataset used for training the model contained data of past 15 years to give a fairly better prediction.
6) The application is easy and simple to use.
7) The machine learning algorithm used in the project is Random Forest regression which is based on the bagging algorithm and uses Ensemble Learning technique. It creates as many trees on the subset of the data and combines the output of all the trees. In this way it reduces over fitting problem in decision trees and also reduces the variance and therefore improves the accuracy.
8) Random Forest algorithm is very stable. Even if a new data point is introduced in the dataset, the overall algorithm is not affected much since the new data may impact one tree, but it is very hard for it to impact all the trees.
9) Random Forest is comparatively less impacted by noise.

Disadvantages :

1) Can be only used by the people having the knowledge of data analysis.
2) As the model is deployed on cloud, so one requires good internet connection to use the application.
3) The model used is Random Forest regression and Random Forest creates a lot of trees (unlike only one tree in case of decision tree) and combines their outputs. By default, it creates 100 trees in Python sklearn library. To do so, this algorithm requires much more computational power and resources.
4) Random Forest require much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decision on the majority of votes.
5) The Node-Red application needs to make HTTP request to IBM cloud and then another HTTP request to the model before providing the prediction. That makes the application a bit slow.

8.APPLICATIONS

- 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.

9.CONCLUSION

By doing the above procedure and all we successfully created Life expectancy prediction system using IBM Watson studio, Watson machine learning and Node-RED service. The potential use of project is not limited to health care in practice, but could also be useful in other clinical applications such as clinical trials. The project makes a good use of machine learning in predicting life expectancy of a country that can help respective government in making policies that will serve for the benefit of the nation and entire humankind.

10.FUTURE SCOPE

- 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

11.BIBILOGRAPHY

A Systematic Literature Review of Studies Analyzing 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 - watsondeveloper-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/life-expectancy-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 .

- 16

APPENDIX

A. Source code

1. Watson Studio

➢ Life_expectancy_prediction.ipynb :

```
# -*- coding: utf-8 -*-
"""Life_expectancy_prediction.ipynb

Automatically generated by Colaboratory.

Original file is located at
https://colab.research.google.com/drive/1z7pthWiDWwcUn19OqYUbPY43A2e8nz1v

# Analysing the dataset
```

****Importing required libraries****

```
"""

import pandas as pd import numpy as np import
matplotlib.pyplot as plt import seaborn as sns from
sklearn.preprocessing import OneHotEncoder from
sklearn.model_selection import train_test_split from
sklearn.neural_network import MLPClassifier from
sklearn.neighbors import KNeighborsClassifier from
sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.compose import ColumnTransformer, make_column_transformer
from sklearn.pipeline import make_pipeline from
sklearn.impute import SimpleImputer
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF from
sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB from sklearn.preprocessing
import LabelEncoder from sklearn.metrics import accuracy_score from
collections import OrderedDict from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import ShuffleSplit from
sklearn.model_selection import cross_val_score from
sklearn.linear_model import LinearRegression from
sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score,mean_squared_error
```

"""****Reading the dataset****"""

```
data=pd.read_csv('/content/drive/My Drive/Smartbrige/Life Expectancy Data.csv')

"""in ibm watson studio"""

import types import
pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials. #
You might want to remove those credentials before you share the notebook.
***************************
*
*
# If you are reading an Excel file into a pandas DataFrame, replace `read_csv` by `read_excel` in the
next statement.
data= pd.read_csv(body)

"""Country- Country
Year- Year
Status- Developed or Developing status
Life Expectancy- Age(years)
Adult Mortality- Adult Mortality Rates of both sexes(probability of dying between 15&60 years per
1000 population)
Infant Deaths- Number of Infant Deaths per 1000 population
Alcohol- Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)
Percent Expenditure- Expenditure on health as a percentage of Gross Domestic Product per capita(%)
Hep B- Hepatitis B (HepB) immunization coverage among 1-year-olds(%)
Measles- number of reported measles cases per 1000 population
BMI- Average Body Mass Index of entire population
U-5 Deaths- Number of under-five deaths per 1000 population
Polio- Polio(Pol3) immunization coverage among 1-year-olds(%)
Total Expenditure- General government expenditure on health as a percentage of total government
expenditure(%)
Diphtheria- Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-
yearolds(%)
HIV/AIDS- Deaths per 1000 live births HIV/AIDS(0-4 years)
GDP- Gross Domestic Product per capita(in USD)
Population- Population
Thinness 10-19- Prevalence of thinness among children and adolescents for Age 10 to 19(%)
Thinness 5-9- Prevalence of thinness among children for Age 5 to 9(%)
Income Composition- Human Development Index in terms of income composition of resources(0-1)
Schooling- Number of years of Schooling
"""

data.head()

data.shape #(2938, 22)
data.describe()

data.info()

data.isnull().sum()

"""# Handling Missing value"""

country_list = data.Country.unique() len(country_list)

country_list = data.Country.unique()
fill_list = ['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure',
'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
' thinness  1-19 years', ' thinness 5-9 years',
'Income composition of resources', 'Schooling']

"""**Filling missing value according to country column using interpolate()**"""

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

data.shape  #(1987, 22) size reduced

data.isna().sum()

"""# Corelation matrix"""

corrMatrix = data.corr()
corrMatrix.style.background_gradient(cmap='plasma', low=.5, high=0).highlight_null('red')

"""Thinness 1-19 years must be thinness 10-19 years

**Renaming the columns as it contains trailing spaces**
"""

data.rename(columns={" BMI ":"BMI",'Life expectancy ':'Life expectancy',
"under-five deaths ":"under-five deaths","Measles ":"Measles","Diphtheria ":"Diphtheria",
' HIV/AIDS':"HIV/AIDS",
" thinness  1-19 years":"thinness 10-19 years"," thinness 5-9 years":"thinness 5-9
years"},inplace=True)

"""# Removing outliers

Taking numeric features , (country,year, status columns are excluded) """
```

```
col_dict = {'Life expectancy':1 , 'Adult Mortality':2 , 'Alcohol':3
, 'percentage expenditure': 4, 'Hepatitis B': 5,
'Measles' : 6, 'BMI': 7, 'under-five deaths' : 8, 'Polio' : 9, 'Total expenditure' :10, 'Diphtheria':11,
'HIV/AIDS':12, 'GDP':13, 'Population' :14,
'thinness 10-19 years' :15, 'thinness 5-9 years' :16,
'Income composition of resources' : 17, 'Schooling' :18, 'infant deaths':19}

"""**Showing outliers using box plot**"""

import matplotlib.pyplot as plt
plt.figure(figsize=(20,30))

for variable,i in col_dict.items(): plt.subplot(5,4,i)
plt.boxplot(data[variable],whis=1.5)
plt.title(variable)

plt.show()

"""**BMI has no outliers**"""

import numpy as np

for variable in col_dict.keys(): q75, q25 = np.percentile(data[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((data[variable] > max_val) | (data[variable] < min_val))[0])),
len((np.where((data[variable] > max_val) | (data[variable] < min_val))[0]))*100/1987))

"""**18 columns having outliers**

as BMI has no outliers
"""

from scipy.stats.mstats import winsorize winsorized_Life_Expectancy =
winsorize(data['Life expectancy'],(0.01,0)) winsorized_Adult_Mortality =
winsorize(data['Adult Mortality'],(0,0.03)) winsorized_Infant_Deaths =
winsorize(data['infant deaths'],(0,0.10)) winsorized_Alcohol =
winsorize(data['Alcohol'],(0,0.01))
winsorized_Percentage_Exp = winsorize(data['percentage expenditure'],(0,0.12))
winsorized_HepatitisB = winsorize(data['Hepatitis B'],(0.11,0)) winsorized_Measles
= winsorize(data['Measles'],(0,0.19))
winsorized_Under_Five_Deaths = winsorize(data['under-five deaths'],(0,0.12))
winsorized_Polio = winsorize(data['Polio'],(0.09,0)) winsorized_Tot_Exp =
winsorize(data['Total expenditure'],(0,0.01)) winsorized_Diphtheria =
winsorize(data['Diphtheria'],(0.10,0)) winsorized_HIV =
winsorize(data['HIV/AIDS'],(0,0.16)) winsorized_GDP =
winsorize(data['GDP'],(0,0.13)) winsorized_Population =
winsorize(data['Population'],(0,0.14)) winsorized_thinness_10_19_years =
winsorize(data['thinness 10-19 years'],(0,0.04)) winsorized_thinness_5_9_years =
winsorize(data['thinness 5-9 years'],(0,0.04))
winsorized_Income_Comp_Of_Resources = winsorize(data['Income composition of
resources'],(0.05,0))
```

```
winsorized_Schooling = winsorize(data['Schooling'],(0.02,0.01))

winsorized_list =
[winsorized_Life_Expectancy,winsorized_Adult_Mortality,winsorized_Alcohol,winsorized_Measles,
winsorized_Infant_Deaths,
winsorized_Percentage_Exp,winsorized_HepatitisB,winsorized_Under_Five_Deaths,winsorized_Poli
o,winsorized_Tot_Exp,winsorized_Diphtheria,
winsorized_HIV,winsorized_GDP,winsorized_Population,winsorized_thinness_10_19_years,winsoriz
ed_thinness_5_9_years,
winsorized_Income_Comp_Of_Resources,winsorized_Schooling]

for variable in winsorized_list:
q75, q25 = np.percentile(variable, [75 ,25]) iqr
= q75 - q25

min_val = q25 - (iqr*1.5)
max_val = q75 + (iqr*1.5)

print("Number of outliers after winsorization in  : {} ".format(len(np.where((variable > max_val) |
(variable < min_val))[0])))

"""**Adding 18 new columns having no outliers to the dataframe**"""

data['winsorized_Life_Expectancy'] = winsorized_Life_Expectancy
data['winsorized_Adult_Mortality'] = winsorized_Adult_Mortality
data['winsorized_Infant_Deaths'] = winsorized_Infant_Deaths data['winsorized_Alcohol']
= winsorized_Alcohol
data['winsorized_Percentage_Exp'] = winsorized_Percentage_Exp data['winsorized_HepatitisB']
= winsorized_HepatitisB
data['winsorized_Under_Five_Deaths'] = winsorized_Under_Five_Deaths
data['winsorized_Polio'] = winsorized_Polio data['winsorized_Tot_Exp'] =
winsorized_Tot_Exp data['winsorized_Diphtheria'] = winsorized_Diphtheria
data['winsorized_HIV'] = winsorized_HIV data['winsorized_GDP'] =
winsorized_GDP data['winsorized_Population'] = winsorized_Population
data['winsorized_thinness_10_19_years'] = winsorized_thinness_10_19_years
data['winsorized_thinness_5_9_years'] = winsorized_thinness_5_9_years
data['winsorized_Income_Comp_Of_Resources'] = winsorized_Income_Comp_Of_Resources
data['winsorized_Schooling'] = winsorized_Schooling data['winsorized_Measles'] =
winsorized_Measles

data.shape #More 18 columns are added
```

"""# **EDA**"""

```
data.columns

sns.distplot(data['Life expectancy '],kde=True)

disease_cols=data[['Life expectancy ','Alcohol','Hepatitis B','Measles ',' BMI ','Polio','Diphtheria ','
HIV/AIDS','Adult Mortality',
'infant deaths','under-five deaths ',' thinness  1-19 years',' thinness 5-9 years','Schooling',
'percentage expenditure','Total expenditure','GDP','Population','Income composition of resources']]

disease_cols.corr()

sns.pairplot(disease_cols,diag_kind='kde')

"""**Hence all the features are significant to predict the target variable**"""

col = ['Life expectancy','winsorized_Life_Expectancy','Adult
Mortality','winsorized_Adult_Mortality','infant deaths',
'winsorized_Infant_Deaths','Alcohol','winsorized_Alcohol','percentage
expenditure','winsorized_Percentage_Exp','Hepatitis B',
'winsorized_HepatitisB','under-five
deaths','winsorized_Under_Five_Deaths','Polio','winsorized_Polio','Total expenditure',
'winsorized_Tot_Exp','Diphtheria','winsorized_Diphtheria','HIV/AIDS','winsorized_HIV','GDP','wins
orized_GDP',
'Population','winsorized_Population','thinness 10-19
years','winsorized_thinness_10_19_years','thinness 5-9 years',
'winsorized_thinness_5_9_years','Income composition of
resources','winsorized_Income_Comp_Of_Resources',
'Schooling','winsorized_Schooling','Measles','winsorized_Measles','GDP','winsorized_GDP']

plt.figure(figsize=(15,75))

for i in range(len(col)):
plt.subplot(19,2,i+1) plt.hist(data[col[i]])
plt.title(col[i]) plt.show()

data.describe(include= 'O') #include specifies the list of datatype to be incluyded .here is Object

plt.figure(figsize=(6,6))
plt.bar(data.groupby('Status')['Status'].count().index,data.groupby('Status')['winsorized_Life_Expectan
cy'].mean()) plt.ylabel("Avg Life_Expectancy") plt.title("Life_Expectancy w.r.t Status") plt.show()

country_data =
data.groupby('Country')['winsorized_Life_Expectancy'].mean().sort_values(ascending=True)
country_data.plot(kind='bar' ,figsize=(50,15),fontsize=30,color='g')
plt.title("Life_Expectancy          w.r.t          Country",fontsize=30)
plt.xlabel("Country",fontsize=30)                         plt.ylabel("Avg
Life_Expectancy")
plt.show()

plt.figure(figsize=(7,5))
plt.bar(data.groupby('Year')['Year'].count().index,data.groupby('Year')['winsorized_Life_Expectancy']
.mean())
plt.xlabel("Year",fontsize=12)
plt.ylabel("Avg Life_Expectancy",fontsize=12) plt.show()

cor_matrix=data.corr()
print(cor_matrix['winsorized_Life_Expectancy'].sort_values(ascending=False))
```

```
import seaborn as sns
from pandas.plotting import scatter_matrix attributes=
['winsorized_Life_Expectancy','winsorized_Income_Comp_Of_Resources','winsorized_Schooling'
,'winsorized_Diphtheria','winsorized_Polio','winsorized_Adult_Mortality','winsorized_Alcohol','winso
rized_Measles','winsorized_Infant_Deaths',
'winsorized_Percentage_Exp','winsorized_HepatitisB','winsorized_Under_Five_Deaths','winsorized_T
ot_Exp',
'winsorized_HIV','winsorized_GDP','winsorized_Population','winsorized_thinness_10_19_years','win
sorized_thinness_5_9_years'] cormat=data[attributes].corr() plt.figure(figsize=(15,15))
sns.heatmap(cormat, square=True, annot=True, linewidths=.5) plt.show()

round(data[['Status','winsorized_Life_Expectancy']].groupby(['Status']).mean(),2)

"""Since 'status' is a categorical feature, we have to find the correlation with Life expectancy"""

import scipy.stats as stats
stats.ttest_ind(data.loc[data['Status']=='Developed','winsorized_Life_Expectancy'],data.loc[data['Statu
s']=='Developing','winsorized_Life_Expectancy'])

data.columns

"""**Now our data has no null values and no outliers**
```

# Creating a new dataframe with refined data

```
"""

new_data=pd.DataFrame(data=data,columns=['Country', 'Year', 'Status',
'BMI', 'winsorized_Adult_Mortality',
'winsorized_Infant_Deaths', 'winsorized_Alcohol',
'winsorized_Percentage_Exp', 'winsorized_HepatitisB',
'winsorized_Under_Five_Deaths', 'winsorized_Polio',
'winsorized_Tot_Exp', 'winsorized_Diphtheria', 'winsorized_HIV',
'winsorized_GDP', 'winsorized_Population',
'winsorized_thinness_10_19_years', 'winsorized_thinness_5_9_years',
'winsorized_Income_Comp_Of_Resources', 'winsorized_Schooling',
'winsorized_Measles',
'winsorized_Life_Expectancy'])

new_data.shape
new_data.rename(columns={
'winsorized_Adult_Mortality':'Adult_Mortality',
'winsorized_Infant_Deaths' :'Infant_Deaths',
'winsorized_Alcohol':'Alcohol',
'winsorized_Percentage_Exp':'Percentage_Expenditure',
'winsorized_HepatitisB':'Hepatitis_B',
'winsorized_Under_Five_Deaths':'Under_Five_Deaths',
'winsorized_Polio':'Polio',
'winsorized_Tot_Exp':'Total_Expenditure',
'winsorized_Diphtheria':'Diphtheria',
'winsorized_HIV':'HIV/AIDS',
'winsorized_GDP':'GDP',
'winsorized_Population':'Population',
'winsorized_thinness_10_19_years':'Thinness_10_19_years',
'winsorized_thinness_5_9_years':'Thinness_5_9_years',
'winsorized_Income_Comp_Of_Resources':'Income_Composition_of_Resources',
'winsorized_Schooling':'Schooling',
'winsorized_Measles':'Measles',
'winsorized_Life_Expectancy':'Life_Expectancy' } ,inplace=True)

new_data.head()

new_data.columns

"""**Separating the input features and label**"""

X = new_data.drop('Life_Expectancy', axis=1)
Y = pd.DataFrame(data=new_data,columns=['Life_Expectancy'])
```

"""****Spilitting the data into train set and test set****"""

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)

"""# Creating a pipeline"""

numeric_features = ['Year', 'BMI',
'Adult_Mortality', 'Infant_Deaths', 'Alcohol', 'Percentage_Expenditure',
'Hepatitis_B', 'Under_Five_Deaths', 'Polio', 'Total_Expenditure',
'Diphtheria', 'HIV/AIDS', 'GDP', 'Population', 'Thinness_10_19_years',
'Thinness_5_9_years', 'Income_Composition_of_Resources', 'Schooling',
'Measles']
categorical_features = ['Country', 'Status']

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder

categorical_transformer = Pipeline(steps=[
('onehot', OneHotEncoder(handle_unknown='ignore')),
])

from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler

numeric_transformer = Pipeline(steps=[ ('imputer',
SimpleImputer(strategy='median'))
 ])
from sklearn.compose import ColumnTransformer

preprocessor = ColumnTransformer(
transformers=[
('cat', categorical_transformer, categorical_features),
('num', numeric_transformer, numeric_features)
]
```

"""# **Finding best algorithm**"""

```
models = OrderedDict([
( "Linear Regression",      Pipeline([
('preprocessor', preprocessor),
('LRegressor', LinearRegression())])  ),
( "Decision Tree Regressor", Pipeline([
('preprocessor', preprocessor),
('DTRegressor', DecisionTreeRegressor())])  ),
( "Random Forest Regressor", Pipeline([
('preprocessor', preprocessor),
('RFRegressor', RandomForestRegressor())])  ),

])

scores = {} for (name, model) in
models.items():
model.fit(X_train,Y_train)
scores[name] =r2_score(model.predict(X_test), Y_test)

scores = OrderedDict(sorted(scores.items())) scores
```

"""**Hence Random forest regression is the most suitable algorithm for this dataset**

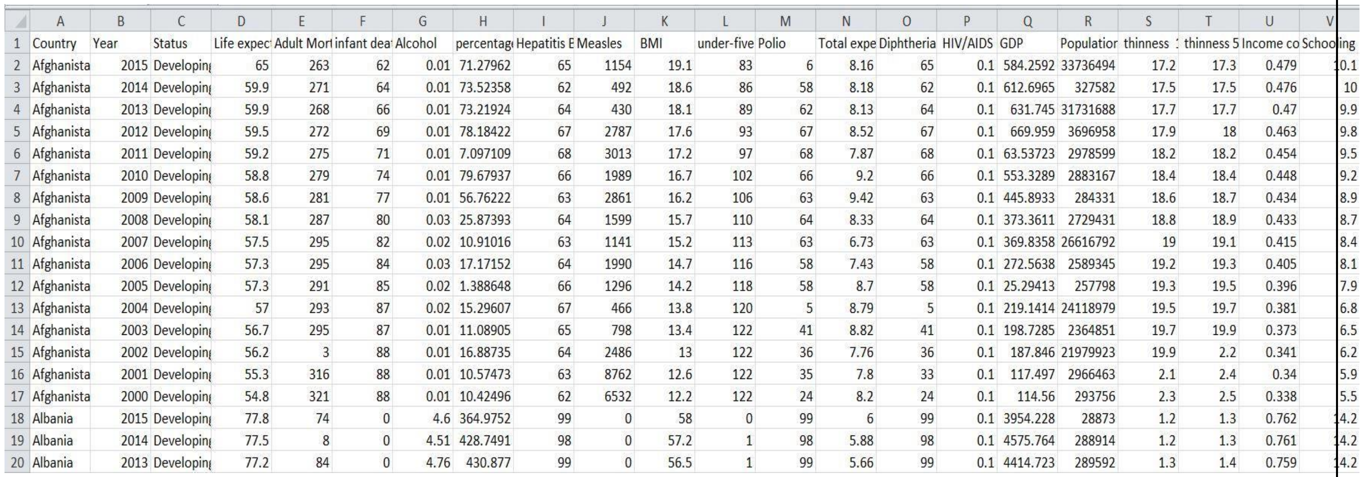
# **Random forest regression**

```
RFRegressor = Pipeline([
('preprocessor', preprocessor),
('RFRegressor', RandomForestRegressor())
])

RFRegressor.fit(X_train,Y_train)

predict= RFRegressor.predict(X_test)

r2_score(predict, Y_test)

"""# Deploying model"""

!pip install watson-machine-learning-client

from watson_machine_learning_client import WatsonMachineLearningAPIClient

# @hidden_cell

wml_credentials={
```

```
"apikey": ***********************,
"iam_apikey_description": *********************************,
"iam_apikey_name": "wdp-writer",
"iam_role_crn":*********************************,
"iam_serviceid_crn":******************************,
"instance_id": ****************************,
"url": *************
}

client = WatsonMachineLearningAPIClient( wml_credentials )

model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "saurabh",
client.repository.ModelMetaNames.AUTHOR_EMAIL: "sauryaprasad0697@gmail.com",
client.repository.ModelMetaNames.NAME: "Life_expectancy"}
model_artifact =client.repository.store_model(RFRegressor, 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="life_expectancy")
scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint
```

➢ Dataset

First 20 rows



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Country	Year	Status	Life expec	Adult Mort	infant dea	Alcohol	percentag	Hepatitis E	Measles	BMI	under-five	Polio	Total expe	Diphtheria	HIV/AIDS	GDP	Populatior	thinness :	thinness 5	Income co	Schooling
2	Afghanista	2015	Developin	65	263	62	0.01	71.27962	65	1154	19.1	83	6	8.16	65	0.1	584.2592	33736494	17.2	17.3	0.479	10.1
3	Afghanista	2014	Developin	59.9	271	64	0.01	73.52358	62	492	18.6	86	58	8.18	62	0.1	612.6965	327582	17.5	17.5	0.476	10
4	Afghanista	2013	Developin	59.9	268	66	0.01	73.21924	64	430	18.1	89	62	8.13	64	0.1	631.745	31731688	17.7	17.7	0.47	9.9
5	Afghanista	2012	Developin	59.5	272	69	0.01	78.18422	67	2787	17.6	93	67	8.52	67	0.1	669.959	3696958	17.9	18	0.463	9.8
6	Afghanista	2011	Developin	59.2	275	71	0.01	7.097109	68	3013	17.2	97	68	7.87	68	0.1	63.53723	2978599	18.2	18.2	0.454	9.5
7	Afghanista	2010	Developin	58.8	279	74	0.01	79.67937	66	1989	16.7	102	66	9.2	66	0.1	553.3289	2883167	18.4	18.4	0.448	9.2
8	Afghanista	2009	Developin	58.6	281	77	0.01	56.76222	63	2861	16.2	106	63	9.42	63	0.1	445.8933	284331	18.6	18.7	0.434	8.9
9	Afghanista	2008	Developin	58.1	287	80	0.03	25.87393	64	1599	15.7	110	64	8.33	64	0.1	373.3611	2729431	18.8	18.9	0.433	8.7
10	Afghanista	2007	Developin	57.5	295	82	0.02	10.91016	63	1141	15.2	113	63	6.73	63	0.1	369.8358	26616792	19	19.1	0.415	8.4
11	Afghanista	2006	Developin	57.3	295	84	0.03	17.17152	64	1990	14.7	116	58	7.43	58	0.1	272.5638	2589345	19.2	19.3	0.405	8.1
12	Afghanista	2005	Developin	57.3	291	85	0.02	1.388648	66	1296	14.2	118	58	8.7	58	0.1	25.29413	257798	19.3	19.5	0.396	7.9
13	Afghanista	2004	Developin	57	293	87	0.02	15.29607	67	466	13.8	120	5	8.79	5	0.1	219.1414	24118979	19.5	19.7	0.381	6.8
14	Afghanista	2003	Developin	56.7	295	87	0.01	11.08905	65	798	13.4	122	41	8.82	41	0.1	198.7285	2364851	19.7	19.9	0.373	6.5
15	Afghanista	2002	Developin	56.2	3	88	0.01	16.88735	64	2486	13	122	36	7.76	36	0.1	187.846	21979923	19.9	2.2	0.341	6.2
16	Afghanista	2001	Developin	55.3	316	88	0.01	10.57473	63	8762	12.6	122	35	7.8	33	0.1	117.497	2966463	2.1	2.4	0.34	5.9
17	Afghanista	2000	Developin	54.8	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
18	Albania	2015	Developin	77.8	74	0	4.6	364.9752	99	0	58	0	99	6	99	0.1	3954.228	28873	1.2	1.3	0.762	14.2
19	Albania	2014	Developin	77.5	8	0	4.51	428.7491	98	0	57.2	1	98	5.88	98	0.1	4575.764	288914	1.2	1.3	0.761	14.2
20	Albania	2013	Developin	77.2	84	0	4.76	430.877	99	0	56.5	1	99	5.66	99	0.1	4414.723	289592	1.3	1.4	0.759	14.2

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

➢ 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]]
```

2. Node Red

➢ Flows.json

[{"id":"f14d51f6.782d9","type":"tab","label":"Flow
2","disabled":false,"info":""},{"id":"a1df2c02.e6e5d","type":"ui_form","z":"f14d51f6.782d9","name":"","label"
:"","group":"62ccd85d.e1ac08","order":1,"width":0,"height":0,"options":[{"label":"Country","value":"a","type":
"text","required":true,"rows":null},{"label":"Year","value":"b","type":"number","required":true,"rows":null},{"l
abel":"Status","value":"c","type":"text","required":true,"rows":null},{"label":"BMI","value":"d","type":"number
","required":true,"rows":null},{"label":"Adult_Mortality","value":"e","type":"number","required":true,"rows":n
ull},{"label":"Infant_Deaths","value":"f","type":"number","required":true,"rows":null},{"label":"Alcohol","valu
e":"g","type":"number","required":true,"rows":null},{"label":"Percentage_Expenditure","value":"h","type":"nu
mber","required":true,"rows":null},{"label":"Hepatitis_B","value":"i","type":"number","required":true,"rows":n
ull},{"label":"Under_Five_Deaths","value":"j","type":"number","required":true,"rows":null},{"label":"Polio","v
alue":"k","type":"number","required":true,"rows":null},{"label":"Total_Expenditure","value":"l","type":"numbe
r","required":true,"rows":null},{"label":"Diphtheria","value":"m","type":"number","required":true,"rows":null},
{"label":"HIV/AIDS","value":"n","type":"number","required":true,"rows":null},{"label":"GDP","value":"o","ty
pe":"number","required":true,"rows":null},{"label":"Population","value":"p","type":"number","required":true,"r
ows":null},{"label":"Thinness_10_19_years","value":"q","type":"number","required":true,"rows":null},{"label"
:"Thinness_5_9_years","value":"r","type":"number","required":true,"rows":null},{"label":"Income_Compositio
n_of_Resources","value":"s","type":"number","required":true,"rows":null},{"label":"Schooling","value":"t","ty
pe":"number","required":true,"rows":null},{"label":"Measles","value":"u","type":"number","required":true,"row
s":null}],"formValue":{"a":"","b":"","c":"","d":"","e":"","f":"","g":"","h":"","i":"","j":"","k":"","l":"","m":"","n":
"","o":"","p":"","q":"","r":"","s":"","t":"","u":""},"payload":"","submit":"Predict","cancel":"cancel","topic":"","x
":70,"y":100,"wires":[["8a2b89d3.bc79a8"]]},{"id":"8a2b89d3.bc79a8","type":"function","z":"f14d51f6.782d9"
,"name":"pre token","func":"//make user given values as global
variables\nglobal.set(\"a\",msg.payload.a);\nglobal.set(\"b\",msg.payload.b);\nglobal.set(\"c\",msg.payload.c);\n
global.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.payl
oad.j);\nglobal.set(\"k\",msg.payload.k);\nglobal.set(\"l\",msg.payload.l);\nglobal.set(\"m\",msg.payload.m);\ngl
obal.set(\"n\",msg.payload.n);\nglobal.set(\"o\",msg.payload.o);\nglobal.set(\"p\",msg.payload.p);\nglobal.set(\"
q\",msg.payload.q);\nglobal.set(\"r\",msg.payload.r);\nglobal.set(\"s\",msg.payload.s);\nglobal.set(\"t\",msg.payl
oad.t);\nglobal.set(\"u\",msg.payload.u);\n\n//following are required to receive a token\nvar
apikey=\"7gdP_fk8LIlGY_3BdmylH8Mm6XcRsdDi_97-
_v3jxx7F\";\nmsg.headers={\"contenttype\":\"application/x-www-form-
urlencoded\"};\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:granttype:apikey\",\"apikey\":apikey};\nr
eturn
msg;\n","outputs":1,"noerr":0,"x":220,"y":100,"wires":[["dc289f99.2cedd"]]},{"id":"9e4b736b.e13b5","type":"h
ttp
request","z":"f14d51f6.782d9","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://ussouth.ml
.cloud.ibm.com/v3/wml_instances/7673ab83-acfd-42c3-98bc-642a09b4fa11/deployments/d3b97720effa-4bd2-
8193-
8810c0e35aba/online","tls":"","persist":false,"proxy":"","authType":"basic","x":470,"y":180,"wires":[["186f0c5
e.cd1e24","47dbb552.65cf1c"]]},{"id":"714664ca.dbe25c","type":"debug","z":"f14d51f6.782d9","name":"","act
ive":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":750,"y":280,"wires":[]},{"id":"
47dbb552.65cf1c","type":"function","z":"f14d51f6.782d9","name":"getFrom
Endpoint","func":"msg.payload=msg.payload.values[0][0];\nreturn
msg;","outputs":1,"noerr":0,"x":490,"y":280,"wires":[["714664ca.dbe25c","9443a681.bc4438"]]},{"id":"186f0c
5e.cd1e24","type":"debug","z":"f14d51f6.782d9","name":"","active":true,"tosidebar":true,"console":false,"tostat
us":false,"complete":"payload","targetType":"msg","x":710,"y":180,"wires":[]},{"id":"8c310af.e649df8","type"
:"function","z":"f14d51f6.782d9","name":"sendTo Endpoint","func":"//get token and make headers\nvar
token=msg.payload.access_token;\nvar instance_id=\"7673ab83-acfd-42c3-98bc-
642a09b4fa11\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer
\"+token,\"MLInstance-ID\":instance_id}\n\n//get variables that are set earlier\nvar a = global.get(\"a\");\nvar b
= global.get(\"b\");\nvar c = global.get(\"c\");\nvar d = global.get(\"d\");\nvar e = global.get(\"e\");\nvar f =

global.get(\"f\");\nvar g = global.get(\"g\");\nvar h = global.get(\"h\");\nvar i = global.get(\"i\");\nvar j =
global.get(\"j\");\nvar k = global.get(\"k\");\nvar l = global.get(\"l\");\nvar m = global.get(\"m\");\nvar n =
global.get(\"n\");\nvar o = global.get(\"o\");\nvar p = global.get(\"p\");\nvar q = global.get(\"q\");\nvar r =
global.get(\"r\");\nvar s = global.get(\"s\");\nvar t = global.get(\"t\");\nvar u = global.get(\"u\");\n\n//send the
user values to service endpoint\nmsg.payload = \n{\"fields\":[\"Country\", \"Year\", \"Status\", \n\"BMI\",
\"Adult_Mortality\", \"Infant_Deaths\", \"Alcohol\", \"Percentage_Expenditure\", \"Hepatitis_B\",
\"Under_Five_Deaths\", \"Polio\", \"Total_Expenditure\", \"Diphtheria\", \"HIV/AIDS\",
\"GDP\",\"Population\", \"Thinness_10_19_years\", \"Thinness_5_9_years\",\n
\"Income_Composition_of_Resources\", \"Schooling\",
\"Measles\"],\n\"values\":[[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u]]};\n\nreturn
msg;\n","outputs":1,"noerr":0,"x":210,"y":180,"wires":[["9e4b736b.e13b5"]]},{"id":"dc289f99.2cedd","type":"h
ttp
request","z":"f14d51f6.782d9","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.
ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"basic","x":370,"y":100,"wires":[["8c310
af.e649df8"]]},{"id":"9443a681.bc4438","type":"ui_text","z":"f14d51f6.782d9","group":"62ccd85d.e1ac08","or
der":2,"width":0,"height":0,"name":"","label":"Prediction","format":"{{msg.payload}}","layout":"rowspread","
x":720,"y":400,"wires":[]},{"id":"62ccd85d.e1ac08","type":"ui_group","z":"","name":"Life Expectancy
Prediction","tab":"c257c6a3.e9b5d8","order":1,"disp":true,"width":"6","collapse":false},{"id":"c257c6a3.e9b5d
8","type":"ui_tab","z":"","name":"Home Page","icon":"dashboard","disabled":false,"hidden":false}

THANK YOU