A Project Report on

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

Being Submitted by :-

SAJJA BHANU PRAKASH

Project Mentor :-

HEMANT KUMAR GAHLOT

Start Date of Internship : 06-06-2020

End Date of Internship : 06-07-2020

SUMMER INTERNSHIP PROGRAM
IN
MACHINE LEARNING

SMART BRIDGE SOLUTIONS PRIVATE LIMITED
JUNE-JULY (2020)

Contents: -

1.INTRODUCTION

The term ***life expectancy*** refers to the number of years a person can expect to live. By definition, life expectancy is based on an estimate of the average age that members of a particular population group will be when they die. 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: Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth and other demographic 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.

Where do we get data to predict the life expectancy?

The Global Health Observatory (GHO) data repository under the World Health Organization (WHO) keeps track of the health status as well as many other related factors for all countries the data-sets are made available to the public for the purpose of health data analysis. The data-set related to life expectancy, health factors for 193 countries have been collected from the same 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

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

1.1Overview: -

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. Understanding potential trajectories in health and drivers of health is crucial to guiding long-term investments and policy implementation. Past work on forecasting has provided an incomplete landscape of future health scenarios, highlighting a need for a more robust modelling platform from which policy options and potential health trajectories can be assessed.

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: -

Good prognostication 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 in a country. 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. However, Physicians tend to overestimate life expectancy, and miss the window of opportunity to initiate Advance Care Planning

***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.

***Growth in social activities**: -*

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.

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

2. LITERATURE SURVEY

2.1 Existing Problem: -

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: -

The project tries to create a model based on data provided by the World Health Organization (WHO) to evaluate the life expectancy for different countries in years. The data offers a timeframe from 2000 to 2015. The project relies on accuracy of data. The Global Health Observatory (GHO) data repository under World Health Organization (WHO) keeps track of the health status as well as many other related factors for all countries the data-sets are made available to public for the purpose of health data analysis.

The data-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation 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. On initial visual inspection of the data showed some missing values. As the data-sets were from WHO, we found no evident errors. Missing data was handled by using Python software. The result indicated that most of the missing data was for population, Hepatitis B and GDP. The missing data were from less known countries like Vanuatu, Tonga, Togo, Cabo

Verde etc. Finding all data for these countries was difficult and hence, it was decided that we exclude these countries from the final model data-set. The final merged file (final dataset) consists of 22 Columns and 2938 rows which meant 20 predicting variables. All predicting variables was then divided into several broad categories: Immunization related factors, Mortality factors, Economical factors and Social factors.

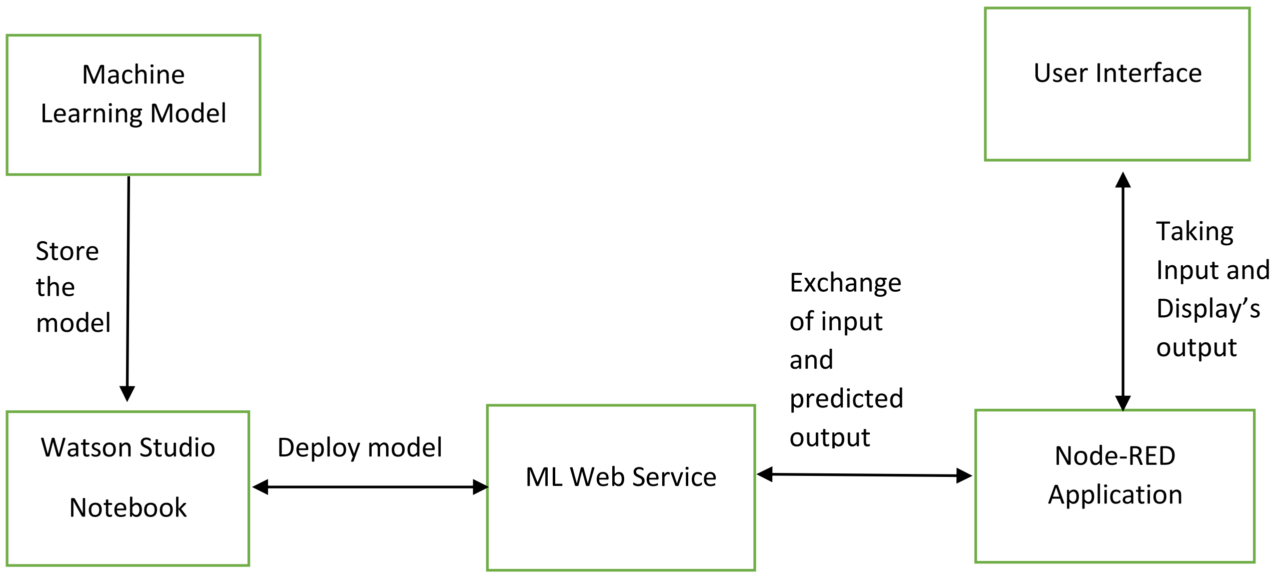
The output algorithms have been used to test if they can maintain their accuracy in predicting the life expectancy for data they haven't been trained. Two algorithms have been used:

1)*Decision Tree Regression*

2)*Random Forest Regression*

3. THEORITICAL ANALYSIS

3.1 Block Diagram: -



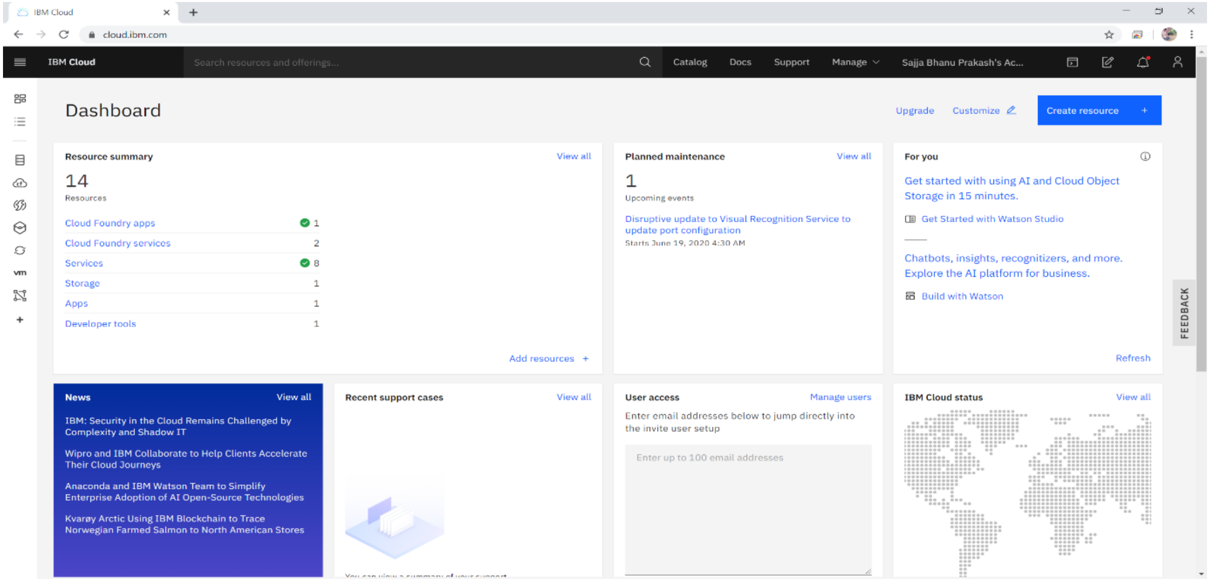
3.2 Hardware / Software designing: -

<u>Software Designing: -</u>

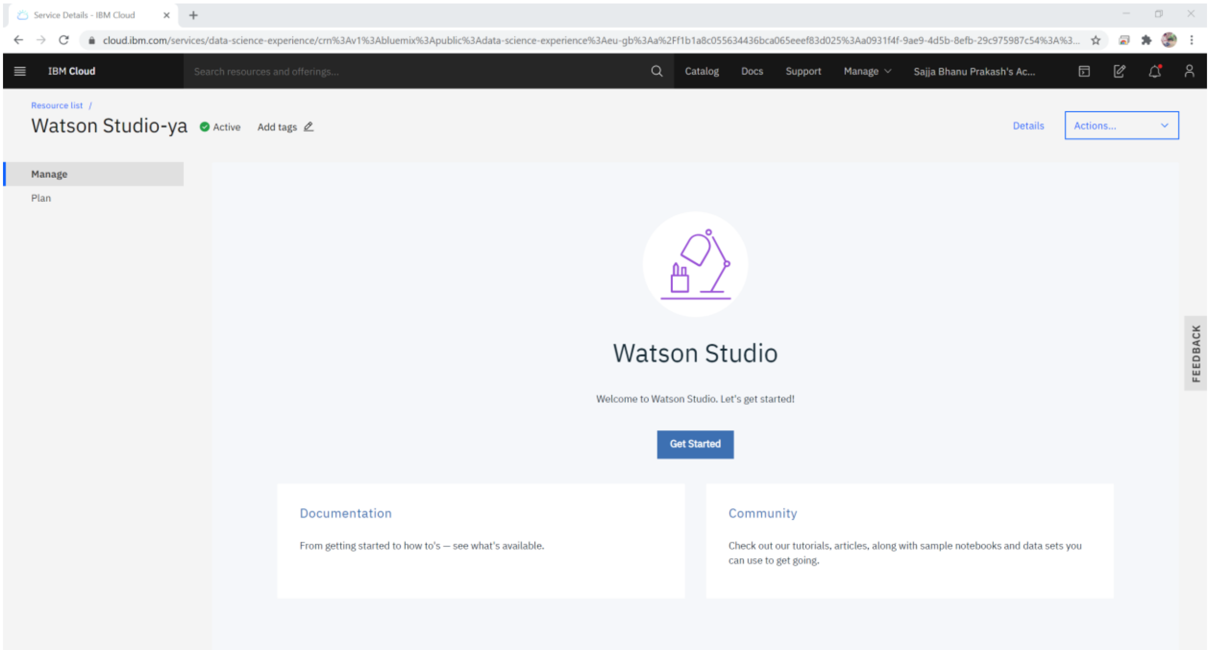
The regression model built in python is deployed on IBM cloud. The Node-RED application then sends HTTP request with all the required parameters to the trained model. The model then sends the HTTP response which is then parsed and displayed on the UI.

<u>Model Designing (Watson Studio)</u>: -

Step1: - Login to the IBM Cloud. This is how Dashboard look like.

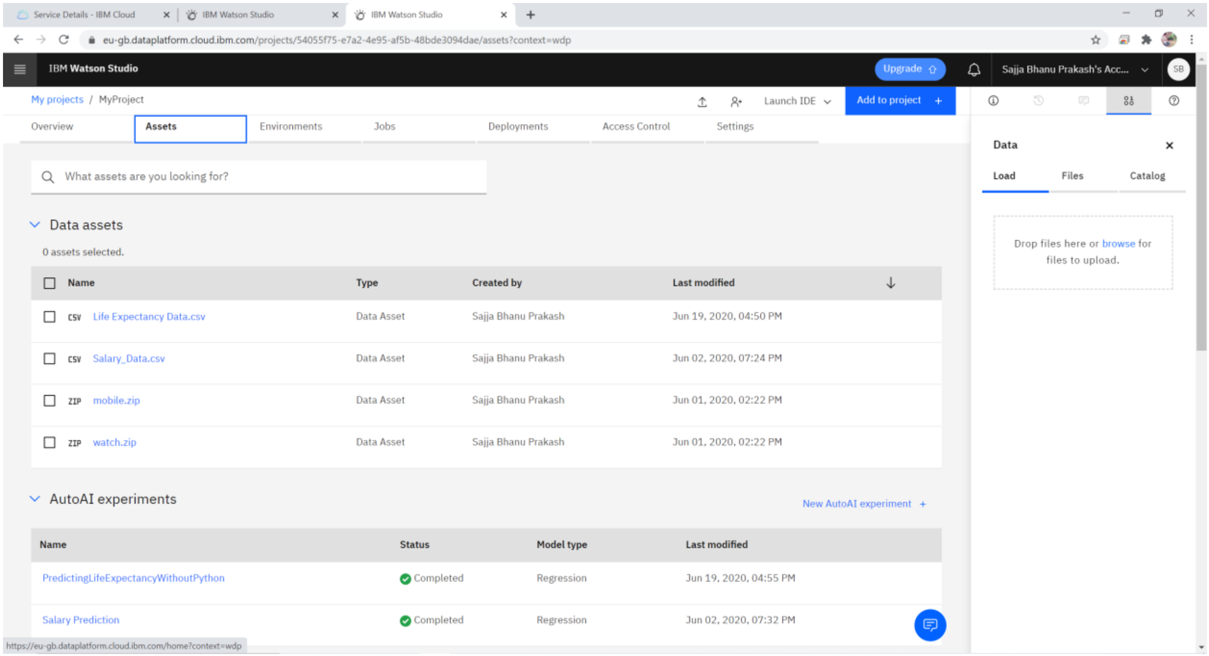


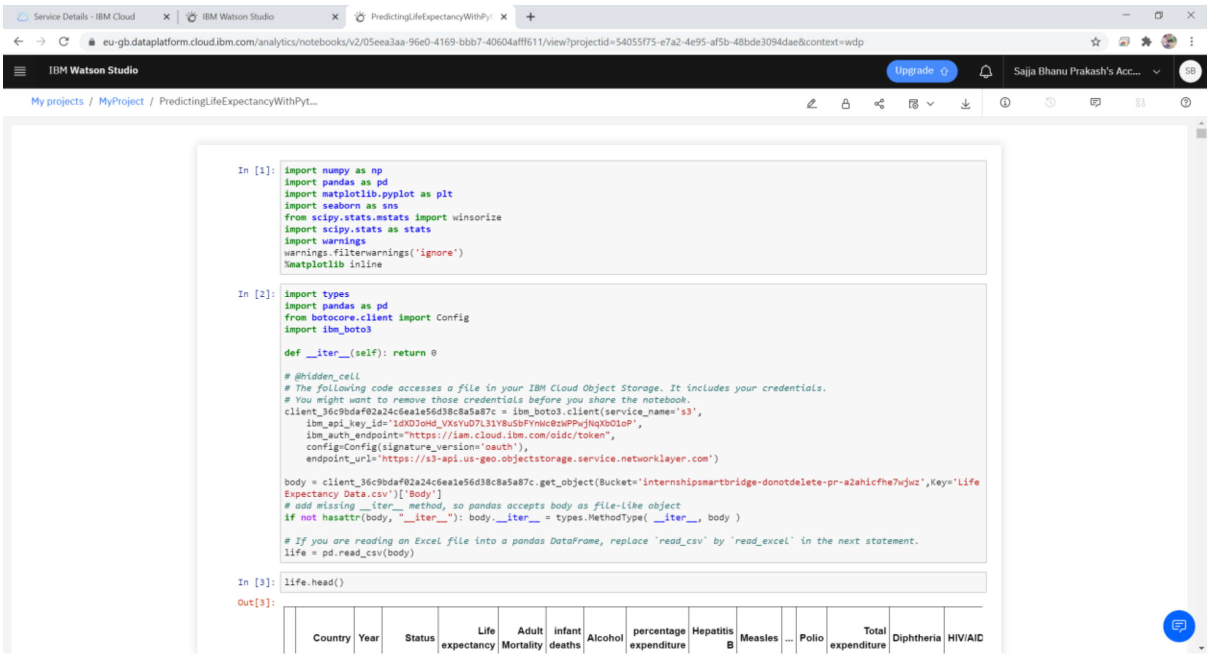
Step2: -Go to Resource List, in services we can find the Watson studio.



(Predicting Life Expectancy using Python)

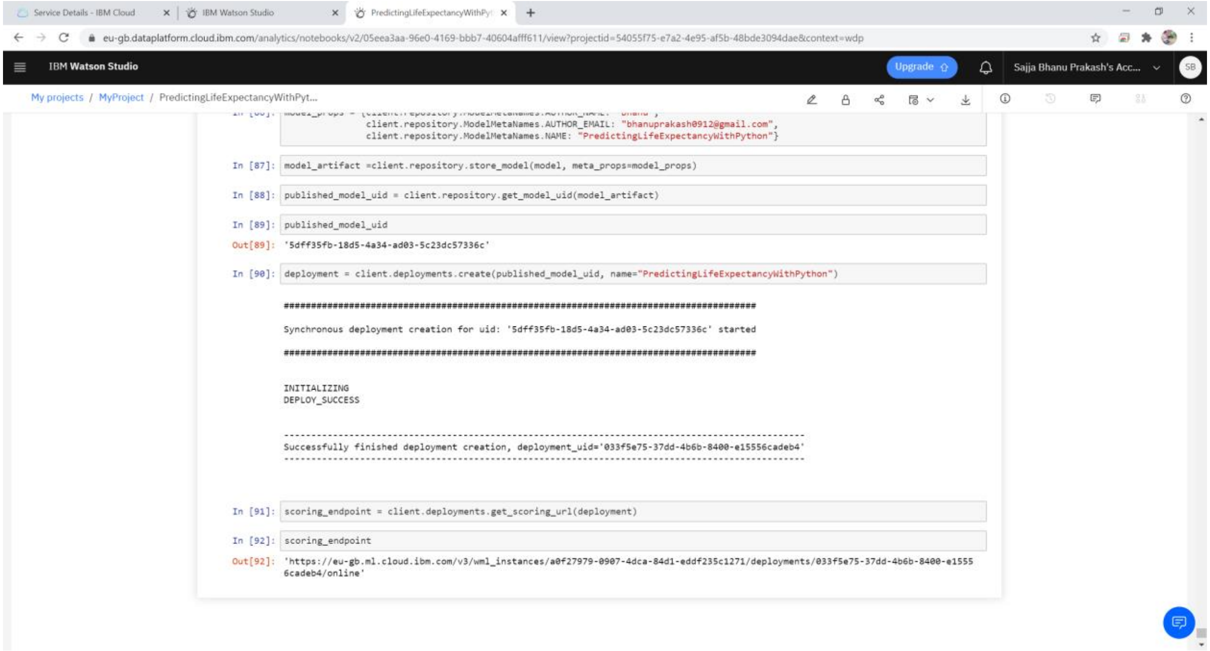
Step3: - New Project => Create an empty Project => Give project name => Click Create => Add to Project => Notebook





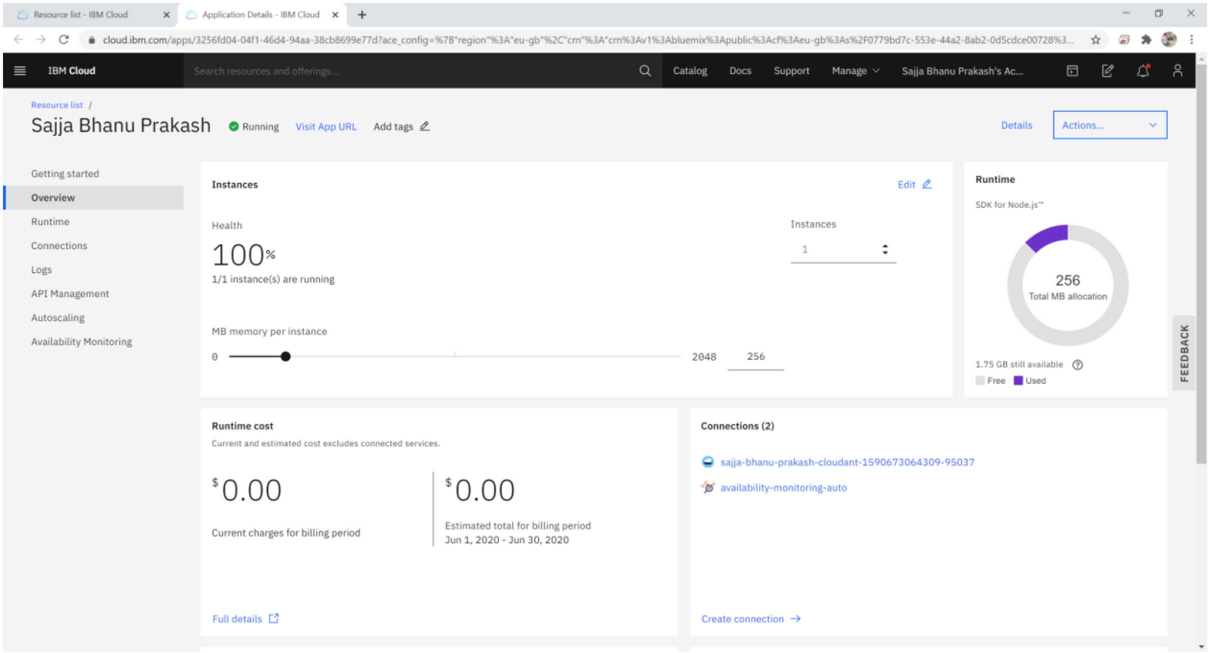
Scoring Endpoint: - For WML credentials, replace with your own credentials of the service. Services => Machine Learning Service => Service Credentials =>

Copy the credentials.



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

Resources => Cloud Foundry Apps => Node-RED => Visit App URL



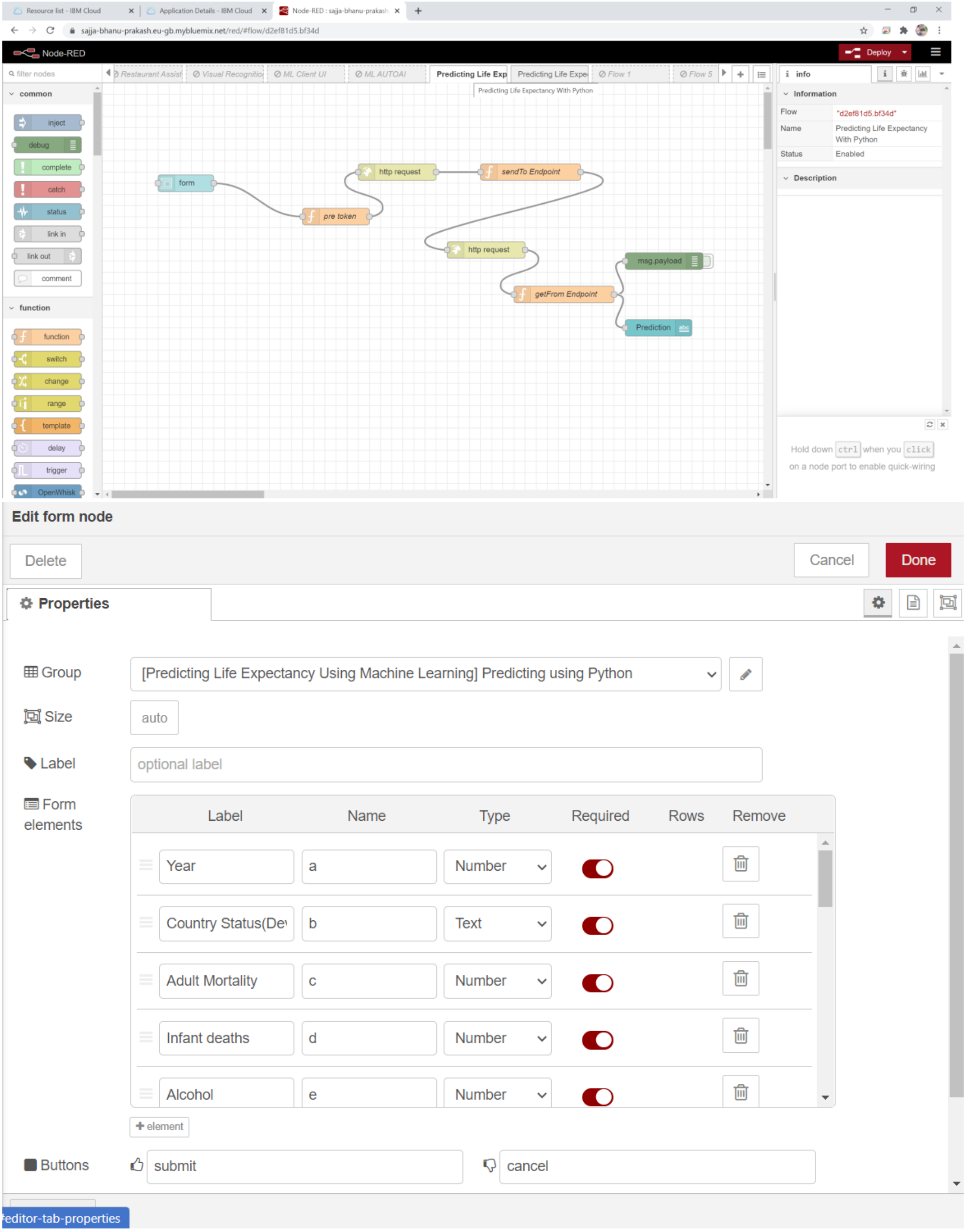
Nodes:

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.

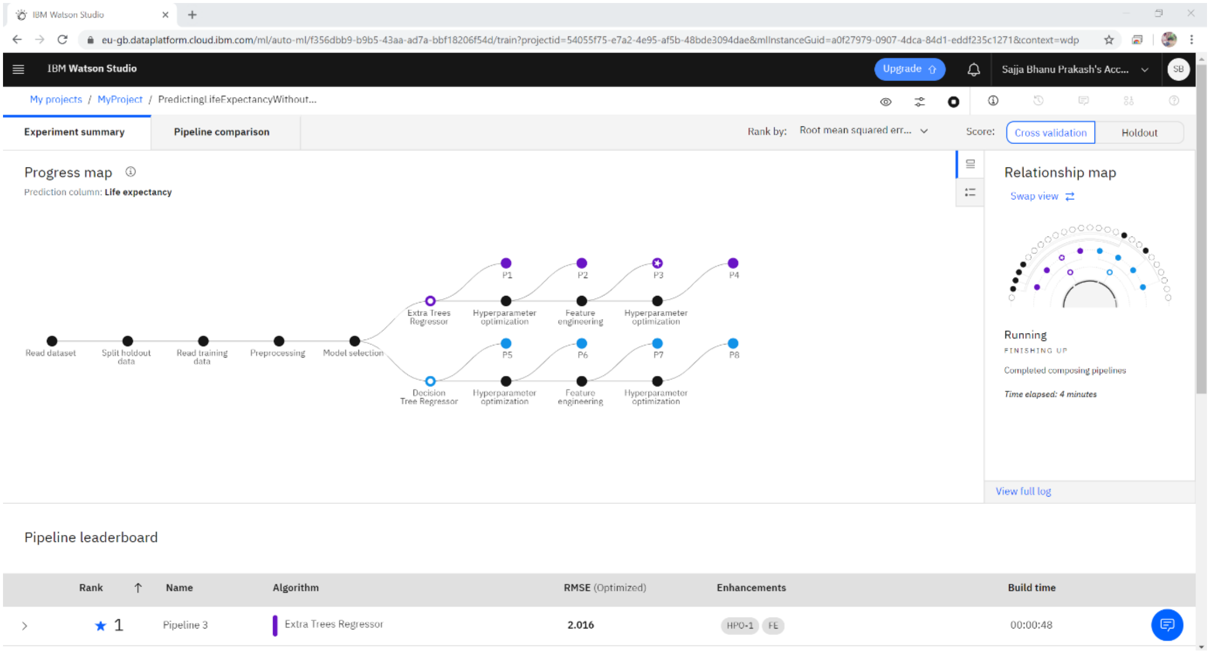


(Predicting Life Expectancy using Auto AI)

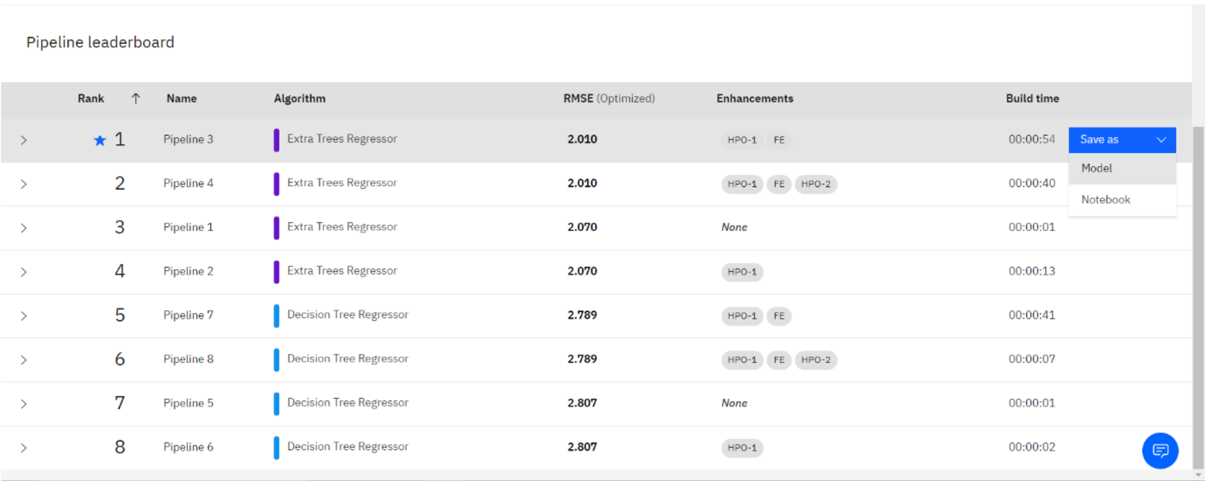
Follow the step 1 and step2 on above;

Step4: - New Project => Create an empty Project => Give project name => Click Create => Add to Project => Auto AI Experiment.

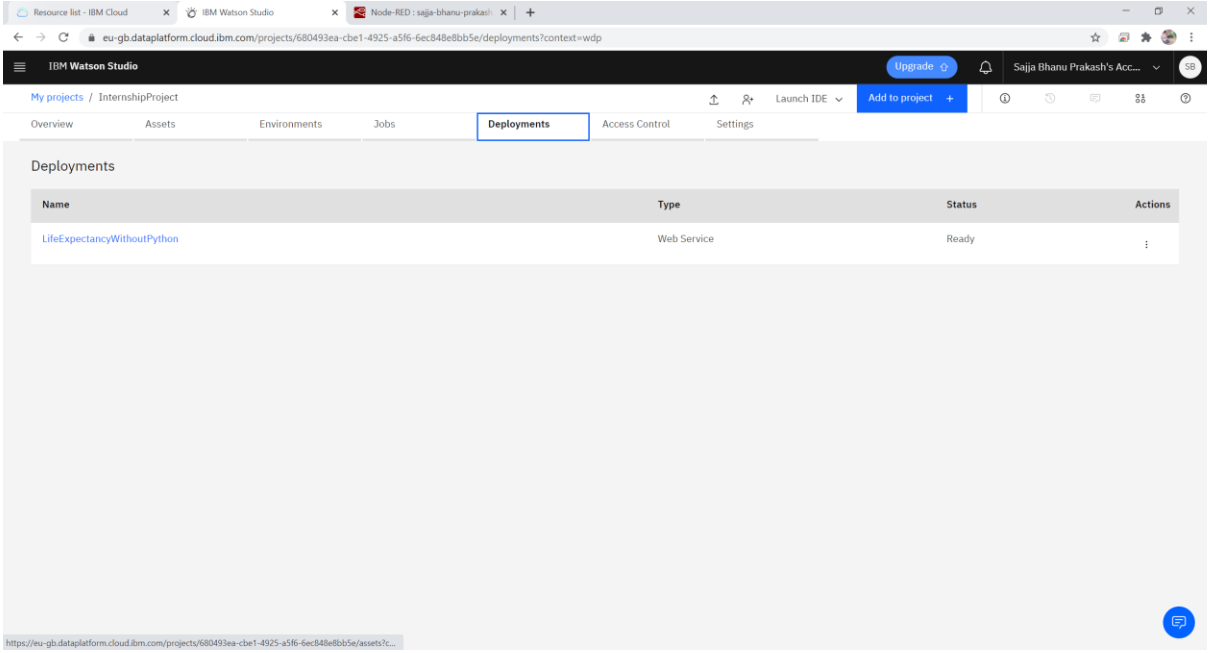
Step5: -Import the Dataset => Select Prediction Column =>Once Check Experiment Settings => Run Experiment.



Step 6: -After Running the Experiment. Select the high rated Pipeline and save as model and after model saved click on ***view project***. Go to Deployments =>Add Deployment => After DEPLOY_SUCCESS => View Deployment.

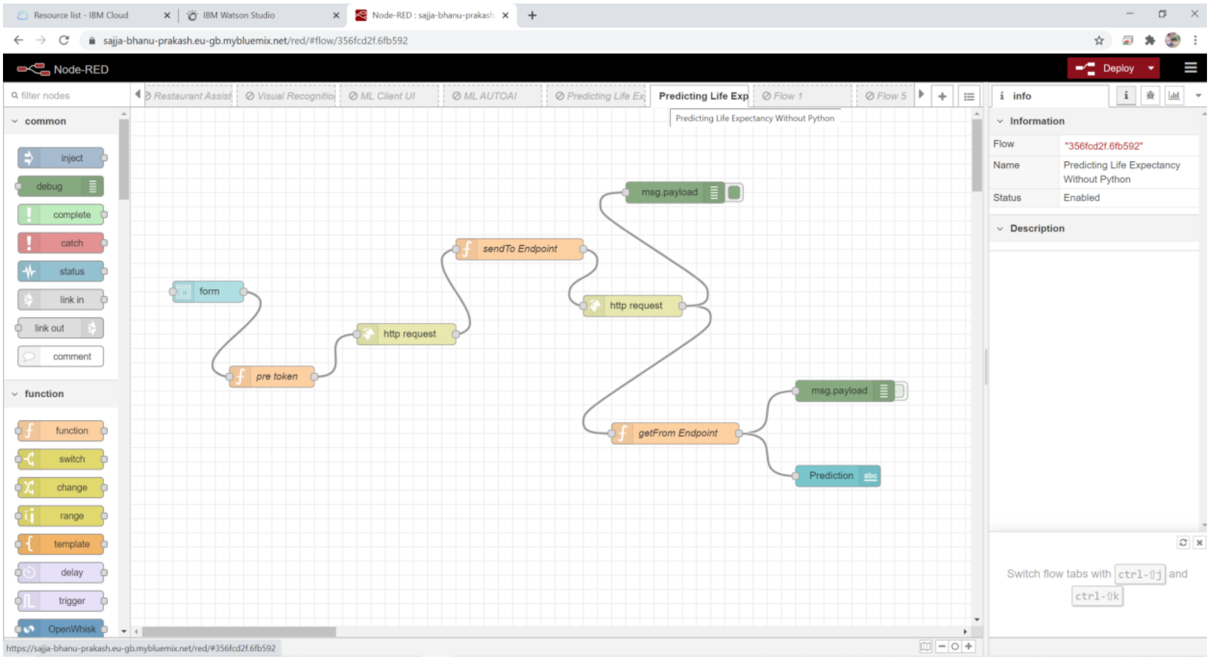


After Deployment, you can Test the Model in Test Tab.



User Interface Integration with AUTO AI(Node- Red) :

Resources => Cloud Foundry Apps => Node-RED => Visit App URL



4. EXPERIMENTAL INVESTIGATIONS

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

4.1. Factors affecting Life Expectancy: -

After Importing the Dataset in Notebook, I analysed the Dataset like Variable Descriptions. Below are the factors (given in the dataset) which affect life expectancy of a country.

1. *Adult Mortality:* Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)
2. *Infant Deaths:* Number of Infant Deaths per 1000 population
3. *Alcohol:* Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)
4. *Percentage Expenditure:* Expenditure on health as a percentage of Gross Domestic Product per capita (%)
5. *Hepatitis B:* Hepatitis B immunization coverage among 1-year-olds (%)
6. *Measles:* Measles - number of reported cases per 1000 population
7. *BMI:* Average Body Mass Index of the entire population
8. *Under-five deaths:* Number of under-five deaths per 1000 population
9. *Polio:* Polio (Pol3) immunization coverage among 1-year-olds (%)
10. *Total Expenditure:* General government expenditure on health as a percentage of total government expenditure (%)
11. *Diphtheria:* Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)
12. *HIV/AIDS:* Deaths per 1 000 live births HIV/AIDS (0-4 years)
13. *GDP:* Gross Domestic Product per capita (in USD)
14. *Population:* Population of the country

15.Thinness 5-9 years: Prevalence of thinness among children for Age 5 to 9(%)

16. Thinness 1-19 years: Prevalence of thinness among children and adolescents for Age 10 to 19 (%)

17. Income composition of resources: Human Development Index in terms of income composition of resources (index ranging from 0 to 1)

18. Schooling: Number of years of Schooling(years)

4.2 Import the Dataset to IBM Cloud: -

Importing the dataset in IBM cloud => Go to Find and Add Data => Adding Dataset as Pandas Data frame.

This is what I learnt.

In [2]:

```python
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_b9c47853ffe043638b1b7e7e6d6187bf = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='PGF_-KPG79WdHM8qAfljiqpODDzX-JZaHD0-PIDVP0eo',
    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_b9c47853ffe043638b1b7e7e6d6187bf.get_object(Bucket='myfirstproject-donotdelete-pr-ixs3sckbx4dbd9',Key='datasets_12
603_17232_Life Expectancy Data (1).csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

# If you are reading an Excel file into a pandas DataFrame, replace `read_csv` by `read_excel` in the next statement.
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/AID
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

4.3 Describing the Data: -

By Describing the data, We Know the computational factors of each Column like average mean, Standard Deviation, Count, Maximum Values, Minimum Values and So…on.

```
In [13]: df.describe().transpose()
```

Out[13]:

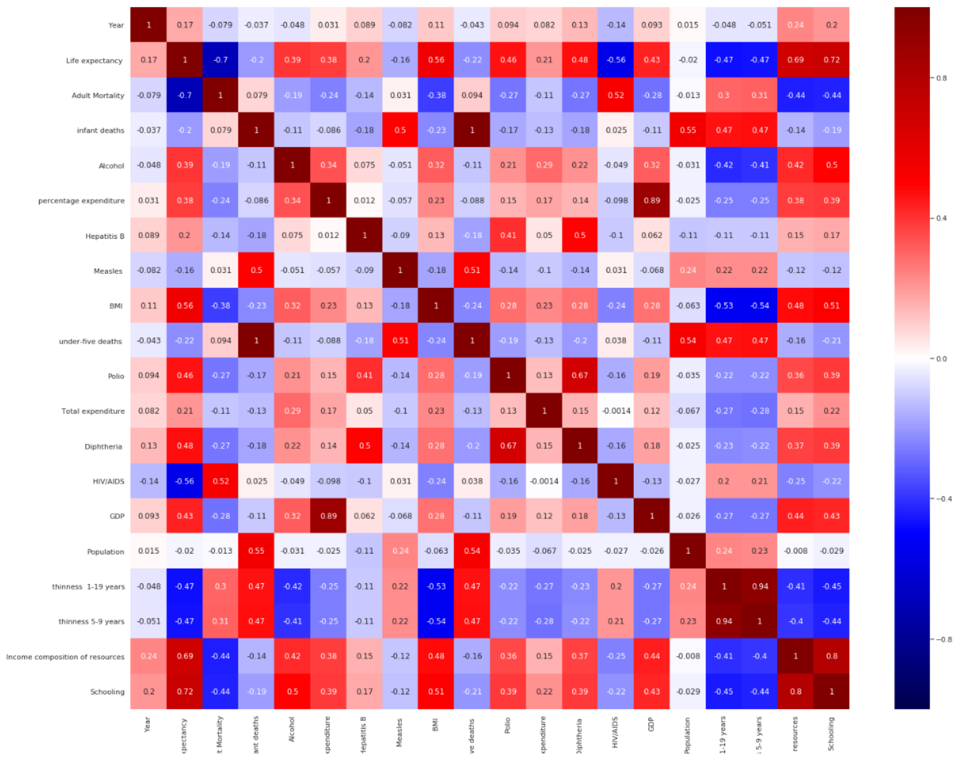
	count	mean	std	min	25%	50%	75%	max
Year	2938.0	2.007519e+03	4.613841e+00	2000.00000	2004.000000	2.008000e+03	2.012000e+03	2.015000e+03
Life expectancy	2938.0	6.922493e+01	9.507640e+00	36.30000	63.200000	7.200000e+01	7.560000e+01	8.900000e+01
Adult Mortality	2938.0	1.647964e+02	1.240803e+02	1.00000	74.000000	1.440000e+02	2.270000e+02	7.230000e+02
infant deaths	2938.0	3.030395e+01	1.179265e+02	0.00000	0.000000	3.000000e+00	2.200000e+01	1.800000e+03
Alcohol	2938.0	4.602861e+00	3.916288e+00	0.01000	1.092500	4.160000e+00	7.390000e+00	1.787000e+01
percentage expenditure	2938.0	7.382513e+02	1.987915e+03	0.00000	4.685343	6.491291e+01	4.415341e+02	1.947991e+04
Hepatitis B	2938.0	8.094046e+01	2.258685e+01	1.00000	80.940461	8.700000e+01	9.600000e+01	9.900000e+01
Measles	2938.0	2.419592e+03	1.146727e+04	0.00000	0.000000	1.700000e+01	3.602500e+02	2.121830e+05
BMI	2938.0	3.832125e+01	1.992768e+01	1.00000	19.400000	4.300000e+01	5.610000e+01	8.730000e+01
under-five deaths	2938.0	4.203574e+01	1.604455e+02	0.00000	0.000000	4.000000e+00	2.800000e+01	2.500000e+03
Polio	2938.0	8.255019e+01	2.335214e+01	3.00000	78.000000	9.300000e+01	9.700000e+01	9.900000e+01
Total expenditure	2938.0	5.938190e+00	2.400274e+00	0.37000	4.370000	5.938190e+00	7.330000e+00	1.760000e+01
Diphtheria	2938.0	8.232408e+01	2.364007e+01	2.00000	78.000000	9.300000e+01	9.700000e+01	9.900000e+01
HIV/AIDS	2938.0	1.742103e+00	5.077785e+00	0.10000	0.100000	1.000000e-01	8.000000e-01	5.060000e+01
GDP	2938.0	7.483158e+03	1.313680e+04	1.68135	580.486996	3.116562e+03	7.483158e+03	1.191727e+05
Population	2938.0	1.275338e+07	5.381546e+07	34.00000	418917.250000	3.675929e+06	1.275338e+07	1.293859e+09
thinness 1-19 years	2938.0	4.839704e+00	4.394535e+00	0.10000	1.600000	3.400000e+00	7.100000e+00	2.770000e+01
thinness 5-9 years	2938.0	4.870317e+00	4.482708e+00	0.10000	1.600000	3.400000e+00	7.200000e+00	2.860000e+01
Income composition of resources	2938.0	6.275511e-01	2.048197e-01	0.00000	0.504250	6.620000e-01	7.720000e-01	9.480000e-01
Schooling	2938.0	1.199279e+01	3.264381e+00	0.00000	10.300000	1.210000e+01	1.410000e+01	2.070000e+01

4.3 Correlation between factors and Life Expectancy:

A heatmap further showed the correlation between different columns

```
In [14]: sns.set(rc={'figure.figsize':(25,20)})
         sns.heatmap(df.corr(),
                     cmap='seismic',annot=True,vmin=-1,vmax=1)
```

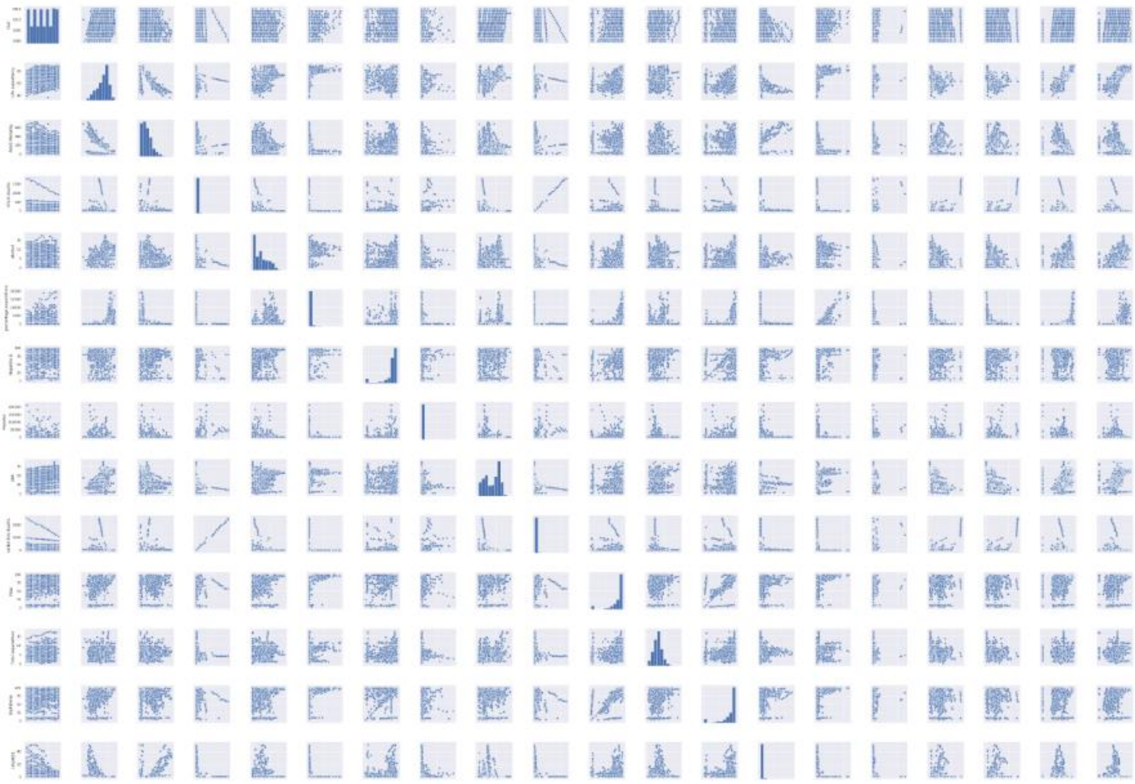
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7cb60717b8>



The legend tells that the warmer colours show higher and positive correlation, while the colder low or negative. There is a very high correlation between thinness of 5-9-year-old and that of 1-19-year-old. Also, between population and infant deaths, under 5 deaths, another is between schooling and income composition of resources. On the other hand, Life expectancy and Adult Mortality are very highly negatively correlated.

4.4 Pairplot of the Columns:-

```
In [15]: sns.pairplot(df)
Out[15]: <seaborn.axisgrid.PairGrid at 0x7f7c991294e0>
```



By seeing some of the graohs from the Pairplot.It seems to be a positive correlation between the Percentage of Healthcare Expenditure, Schooling, GDP and BMI and Life Expectancy, while there is a negative one between Adult 8 Mortality, AIDS and Life Expectancy, there does not seem to have any correlation between Alcohol, under 5 years – old deaths and Life Expectancy.

4.5 Implementing Regression Models

Two Regression Models were Applied :-

1)Decision Trees: -

A decision tree-based model builds a set of rules from the training data to be able to predict the outcome. For the sake of understanding, this algorithm is compared to trees formed through decisions. The model contains branches that represent the rules that lead to the path of the outcome, that is, the leaf. Each prediction path leads to a leaf that contains multiple values. The same principle is applied to classification-type problems as well. For regression-type problems, the final prediction is usually the average of all of the values contained in the leaf it falls under.

```
In [23]: from sklearn.tree import DecisionTreeRegressor
         DT = DecisionTreeRegressor(max_depth=15, min_samples_leaf=100)
         DT.fit(x_train,y_train)
         cv_score= np.sqrt(-cross_val_score(DT,x_train,y_train, cv=10, scoring='neg_mean_squared_error'))
         rmse = np.mean(cv_score)
         print(rmse)

         3.5750783942462334
```

After Checking using decision tree we are getting an root mean square is 3.5750783942462334. So,Let's try for the another model

2)Random Forest Trees: -

Decision trees are generally considered weak models because their performance usually is not up to the expected mark when the data set is relatively large. However, when several decision trees are combined into a single model, they provide greater accuracy. Each decision tree within this random forest is built using a subset of the training data. The number of decision trees that make this random forest is an arbitrary number that can be tuned to see the changes in accuracy. When a value to be predicted is run through this resulting model, it is the average of the values acquired from each of these individual trees.

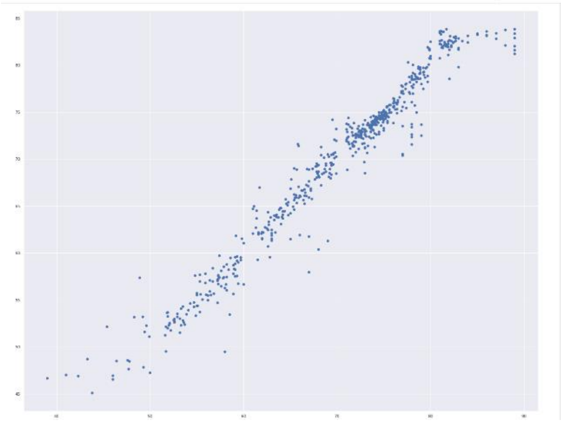
```
In [24]: from sklearn.ensemble import RandomForestRegressor
         rf = RandomForestRegressor(n_estimators = 100, random_state = 42)
         rf.fit(x_train,y_train)
         cv_score= np.sqrt(-cross_val_score(rf,x_train,y_train, cv=10, scoring='neg_mean_squared_error'))
         rmse = np.mean(cv_score)
         print(rmse)

         1.878059235090641
```

Since Random Forest is better we'll use RF

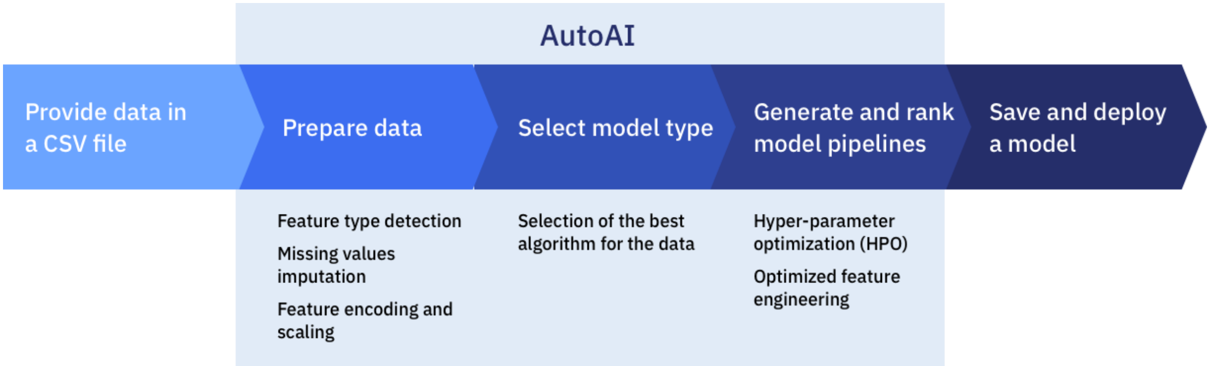
It is observable that the Mean Absolute Error in Random Forest Regression is lower than that in Decision Trees. So, for the final deployment, Random Forest Regression is used.

4.6 Prediction Plot: -



4.7 AUTO AI: -

Using Auto AI, we can build and deploy a machine learning model with sophisticated training features and no coding. The tool does most of the work for us.



Data Pre-processing: -

Most data sets contain different data formats and missing values, but standard machine learning algorithms work with numbers and no missing values. Auto AI applies various algorithms, or estimators, to analyse, clean, and prepare your raw data for machine learning. It automatically detects and categorizes features based on data type, such as categorical or numerical. Depending on the categorization, it uses hyper-parameter optimization to determine the best combination of strategies for missing value imputation, feature encoding, and feature scaling for your data.

Automated Model Selection: -

The next step is automated model selection that matches your data. Auto AI uses a novel approach that enables testing and ranking candidate algorithms against small subsets of the data, gradually increasing the size of the subset for the most promising algorithms to arrive at the best match. This approach saves time without sacrificing performance. It enables ranking a large number of candidate algorithms and selecting the best match for the data.

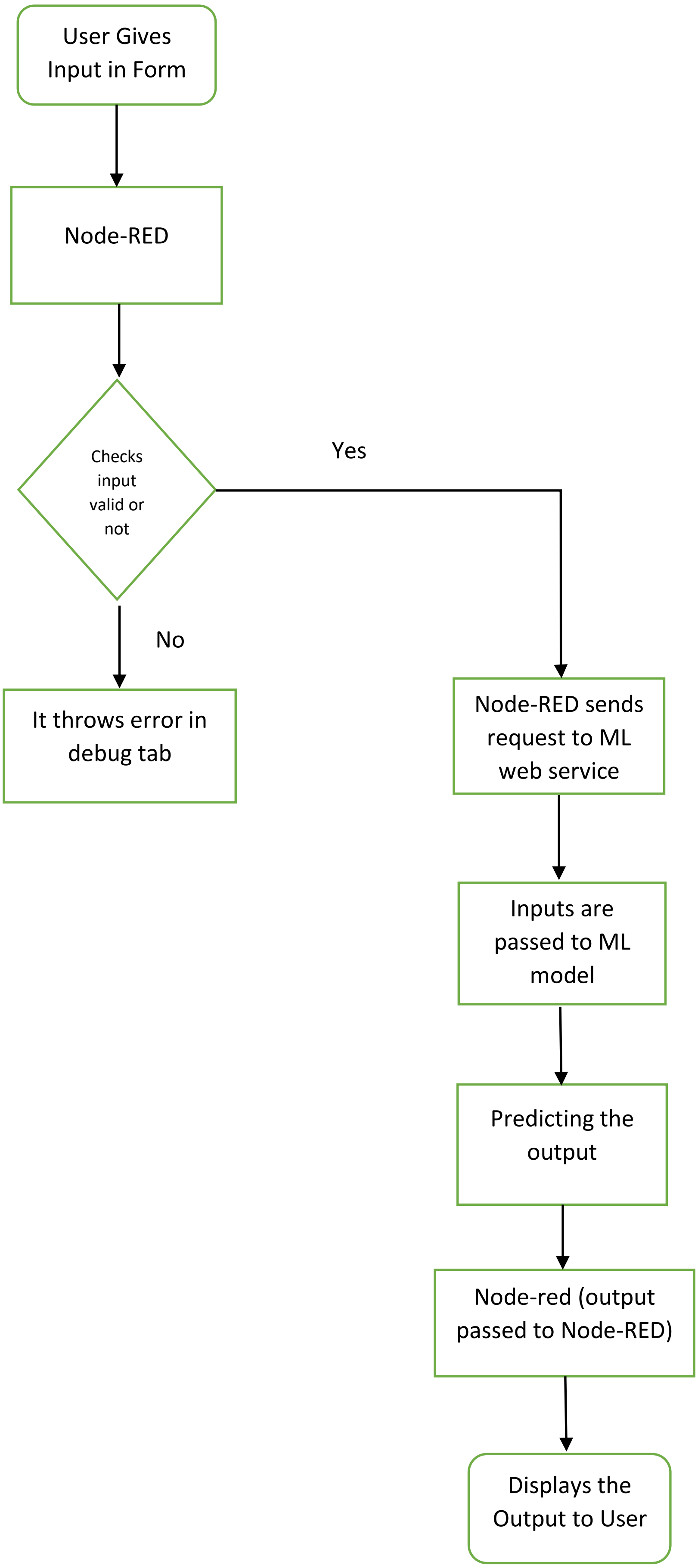
Automated feature engineering: -

Feature engineering attempts to transform the raw data into the combination of features that best represents the problem to achieve the most accurate prediction. Auto AI uses a unique approach that explores various feature construction choices in a structured, non-exhaustive manner, while progressively maximizing model accuracy using reinforcement learning. This results in an optimized sequence of transformations for the data that best match the algorithms of the model selection step.

Hyperparameter optimization: -

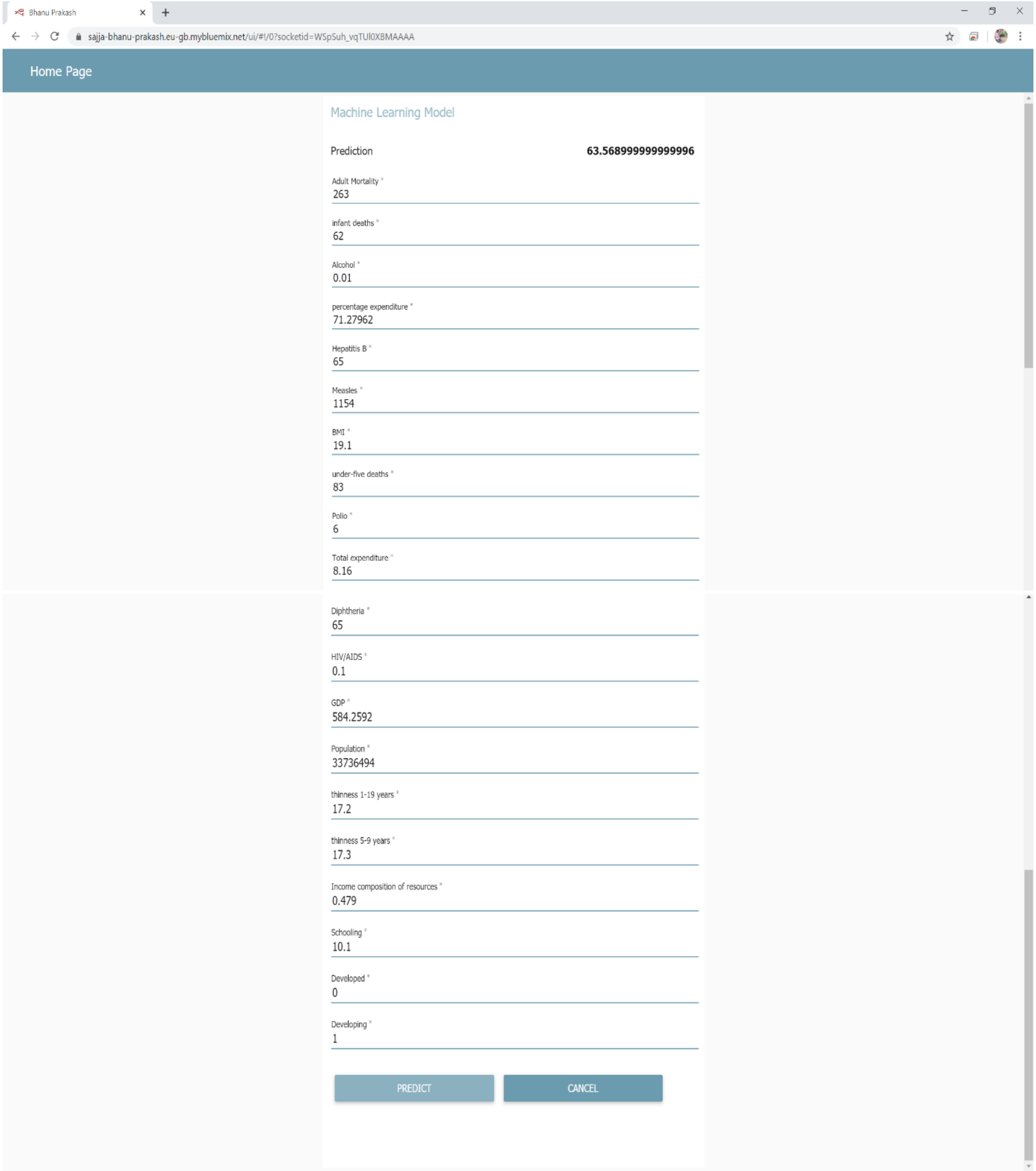
Finally, a hyper-parameter optimization step refines the best performing model pipelines. Auto AI uses a novel hyper-parameter optimization algorithm optimized for costly function evaluations such as model training and scoring that are typical in machine learning. This approach enables fast convergence to a good solution despite long evaluation times of each iteration.

5. FLOWCHART


User Gives Input in Form
Node-RED
Checks input valid or not
Yes
No
It throws error in debug tab
Node-RED sends request to ML web service
Inputs are passed to ML model
Predicting the output
Node-red (output passed to Node-RED)
Displays the Output to User

6.RESULT

6.1 Prediction Using Python: -



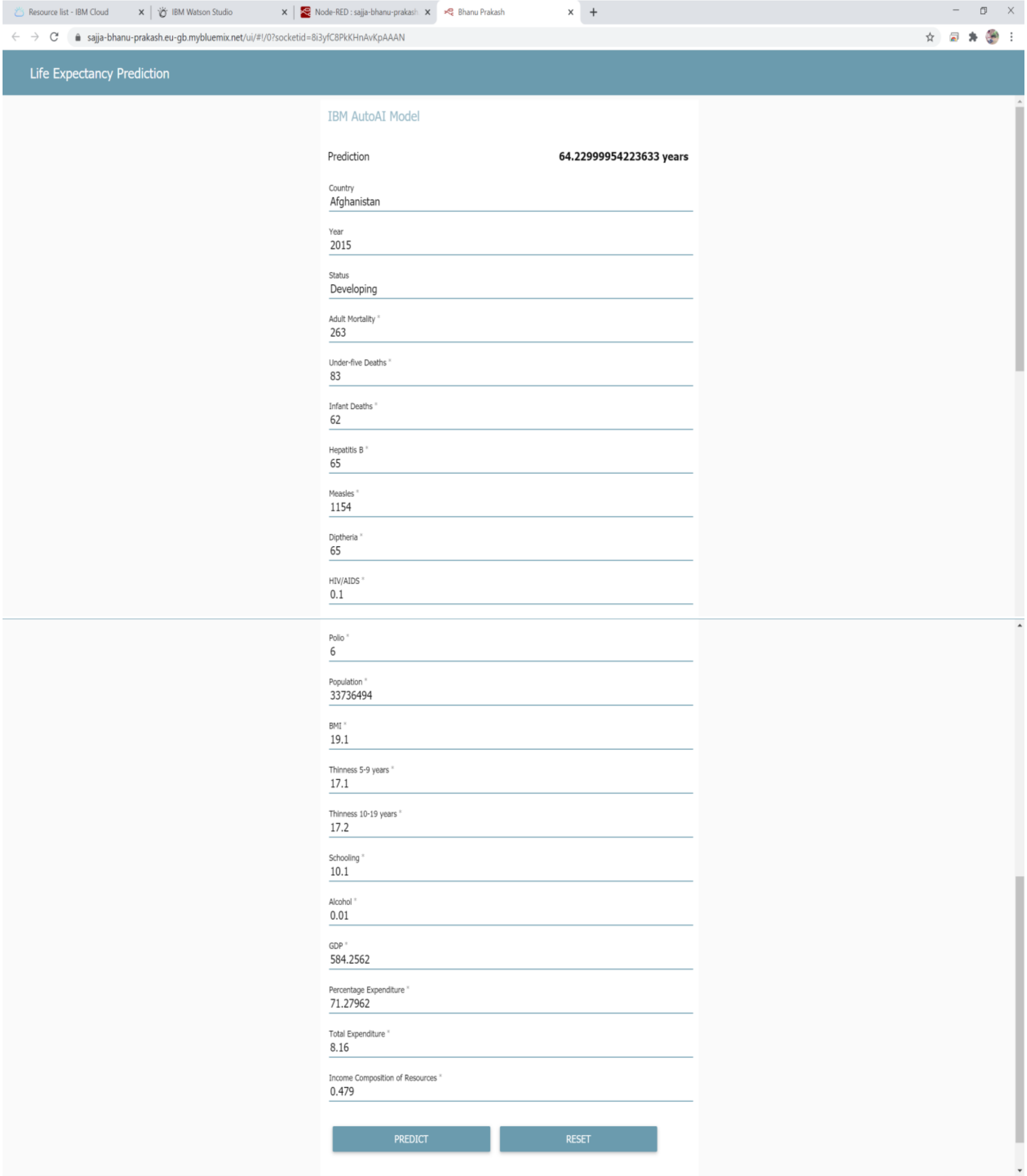
For Afghanistan: 2014

Actual Value: 63.6

Predicted value: 63.56

Error Percentage: (63.6-63.56)/63.6 = 0.06%

6.2 Predicting Using Auto AI: -



For Afghanistan: 2015

Actual Value: 64

Predicted value: 64.22

Error Percentage: (64-64.22)/64 = 0.34%(negative)

7. ADVANTAGES & DISADVANTAGES

7.1 Advantages: -

- The machine learning algorithm used in this 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.
- Random Forest is usually robust to outliers and can handle them automatically. It is comparatively less impacted by noise data.
- 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.
- We can create a user interface easily with help of Node-RED and give the input to the model and predicts the Life expectancy.
- No risk of front-end HTML and CSS programming.

7.2 Disadvantages: -

- **Complexity:** 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 ski-learn library. To do so, this algorithm requires much more computational power and resources. On the other hand, decision tree is simple and does not require so much computational resources. Random Forest require much more time to train as compared to decision trees. But, Predicts the most accuracy value.
- 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.
- The main disadvantage is that no one can predict the future. No one knows when someone will die, who will get cancer or not, who will recover and who won't. Statistics work in generalities. Humans, however, do not.
- It may happen that our model will not predict right when sudden influencing factors affect human life like Ex : COVID-19

8. APPLICATIONS

- Life expectancy predictions have the potential to be beneficial to individuals, health service providers and governments. For instance, they would make people more aware of their general health, and its improvement or deterioration over time. This may motivate them to make healthier lifestyle choices.
- They could also be used by insurance companies to provide individualised services, such as how some car insurance companies use black-box technology to reduce premiums for more cautious drivers.
- 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.
- Advance Care Planning.

9. CONCLUSION

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

User can interact with the system via a simple user interface which is in the form of a form with input spaces which the user needs to fill the inputs into the form.

10. FUTURE SCOPE

Planning Health Services: -

The government can plan health services better using the data and future predictions. Life expectancy plays a major role in development of a country, hence, using predictions and trends, the health infrastructure can be improved. A mobile application can be developed that uses personal health data (from Smart Watch and Health apps) and historical data of the country that user lives in and predict the expected life span of that user

Future Usage: -

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. we can connect the model to the database to have the record of predictions. This will help us analyse the trends in the life span.

Pharmaceutical companies can check which diseases impact more people and therefore impact life expectancy and based on this manufacture medicine.

11. BIBILOGRAPHY

1.Product and Services: -

https://www.ibm.com/watson/products-services

2.Machine Learning Reference: -

https://developer.ibm.com/technologies/machine-learning/series/learning-path-machine-learning-for-developers/

https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html

3.Auto AI: -

https://developer.ibm.com/tutorials/watson-studio-auto-ai/

https://www.youtube.com/watch?v=IDKCmC1fCiU

https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/autoai-overview.html

4.Data-set: -https://www.kaggle.com/kumarajarshi/life-expectancy-who

5.Smart Bridge(Bootcamp): - https://www.youtube.com/channel/UCvB8PgOZdb2y7lgToPE-Dfw

6.Additional Websites: -https://towardsdatascience.com/what-really-drives-higher-life-expectancy-e1c1ec22f6e1

APPENDIX

A. Source Code: -

1)Machine Learning Notebook:

Notebook.ipynb

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
```

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 noteb
ook.
client_b9c47853ffe043638b1b7e7e6d6187bf = ibm_boto3.client(service_name
='s3',
    ibm_api_key_id='PGF_-KPG79WdHM8qAfljiqpODDzX-JZaHD0-PIDVP0eo',
    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_b9c47853ffe043638b1b7e7e6d6187bf.get_object(Bucket='myfir
stproject-donotdelete-pr-ixs3sckbx4dbd9',Key='datasets_12603_17232_Life
Expectancy Data (1).csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like obje
ct
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __i
ter__, body )

# If you are reading an Excel file into a pandas DataFrame, replace `re
ad_csv` by `read_excel` in the next statement.
df = pd.read_csv(body)
df.head()
```

Out [2]:

Out[2]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AID
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.columns
```

Out [4]:

```
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortalit
y','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'
],dtype='object')
```

In [5]:

```
df.info()
df.isnull().sum()
```

Out [5]:

```
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 [6]:

```
cateogry_col=df.select_dtypes(include=['object']).columns.tolist()
integer_col=df.select_dtypes(include=['int64','float64']).columns.tolis
t()

for column in df:
    if df[column].isnull().any():
```

```
        if(column in cateogry_col):
            df[column]=df[column].fillna(df[column].mode()[0])
        else:
            df[column]=df[column].fillna(df[column].mean())
df.isnull().sum()
```

Out [6]:

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

In [7]:

```
df.head(5)
```

Out [7]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIL
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 [8]:

```
df.describe().transpose()
```

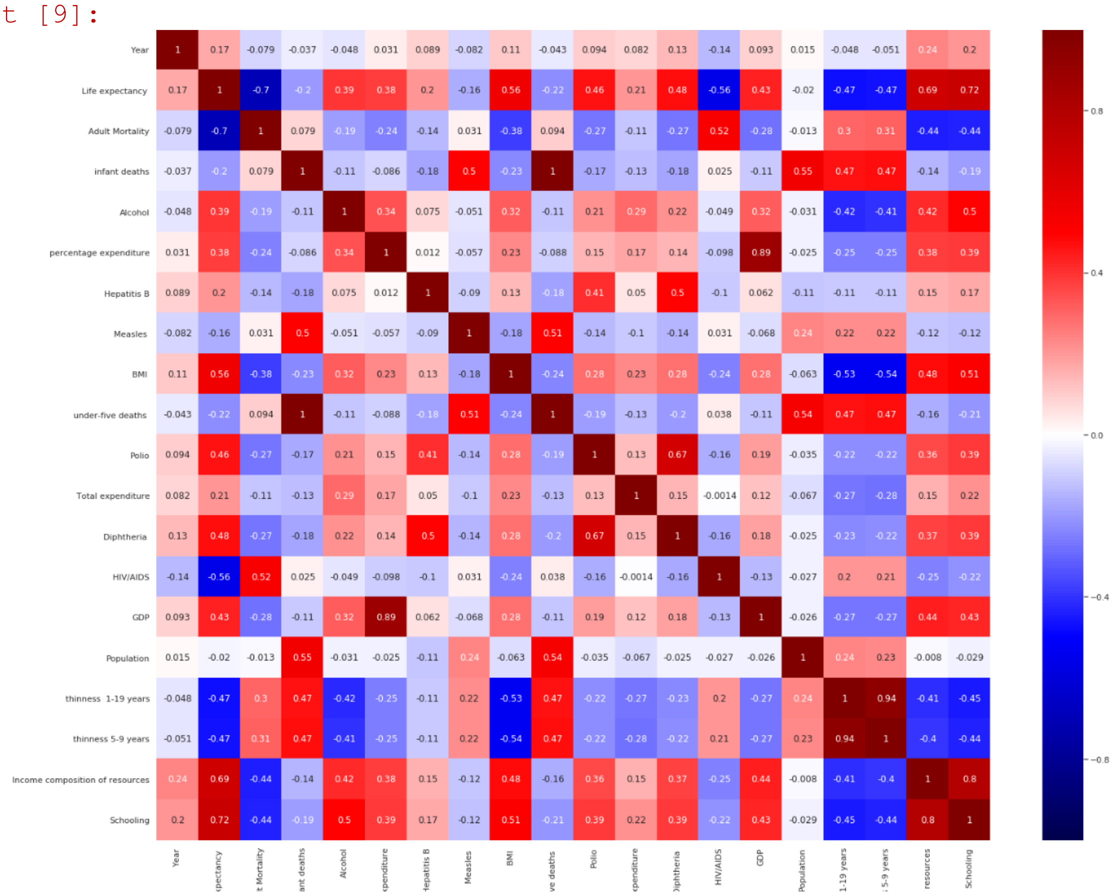
Out [8]:

	count	mean	std	min	25%	50%	75%	max
Year	2938.0	2.007519e+03	4.613841e+00	2000.00000	2004.000000	2.008000e+03	2.012000e+03	2.015000e+03
Life expectancy	2938.0	6.922493e+01	9.507640e+00	36.30000	63.200000	7.200000e+01	7.560000e+01	8.900000e+01
Adult Mortality	2938.0	1.647964e+02	1.240803e+02	1.00000	74.000000	1.440000e+02	2.270000e+02	7.230000e+02
infant deaths	2938.0	3.030395e+01	1.179265e+02	0.00000	0.000000	3.000000e+00	2.200000e+01	1.800000e+03
Alcohol	2938.0	4.602861e+00	3.916288e+00	0.01000	1.092500	4.160000e+00	7.390000e+00	1.787000e+01
percentage expenditure	2938.0	7.382513e+02	1.987915e+03	0.00000	4.685343	6.491291e+01	4.415341e+02	1.947991e+04
Hepatitis B	2938.0	8.094046e+01	2.258685e+01	1.00000	80.940461	8.700000e+01	9.600000e+01	9.900000e+01
Measles	2938.0	2.419592e+03	1.146727e+04	0.00000	0.000000	1.700000e+01	3.602500e+02	2.121830e+05
BMI	2938.0	3.832125e+01	1.992768e+01	1.00000	19.400000	4.300000e+01	5.610000e+01	8.730000e+01
under-five deaths	2938.0	4.203574e+01	1.604455e+02	0.00000	0.000000	4.000000e+00	2.800000e+01	2.500000e+03
Polio	2938.0	8.255019e+01	2.335214e+01	3.00000	78.000000	9.300000e+01	9.700000e+01	9.900000e+01
Total expenditure	2938.0	5.938190e+00	2.400274e+00	0.37000	4.370000	5.938190e+00	7.330000e+00	1.760000e+01
Diphtheria	2938.0	8.232408e+01	2.364007e+01	2.00000	78.000000	9.300000e+01	9.700000e+01	9.900000e+01
HIV/AIDS	2938.0	1.742103e+00	5.077785e+00	0.10000	0.100000	1.000000e-01	8.000000e-01	5.060000e+01
GDP	2938.0	7.483158e+03	1.313680e+04	1.68135	580.486996	3.116562e+03	7.483158e+03	1.191727e+05
Population	2938.0	1.275338e+07	5.381546e+07	34.00000	418917.250000	3.675929e+06	1.275338e+07	1.293859e+09
thinness 1-19 years	2938.0	4.839704e+00	4.394535e+00	0.10000	1.600000	3.400000e+00	7.100000e+00	2.770000e+01
thinness 5-9 years	2938.0	4.870317e+00	4.482708e+00	0.10000	1.600000	3.400000e+00	7.200000e+00	2.860000e+01
Income composition of resources	2938.0	6.275511e-01	2.048197e-01	0.00000	0.504250	6.620000e-01	7.720000e-01	9.480000e-01
Schooling	2938.0	1.199279e+01	3.264381e+00	0.00000	10.300000	1.210000e+01	1.410000e+01	2.070000e+01

In [9]:

```
sns.set(rc={'figure.figsize':(25,20)})
sns.heatmap(df.corr(),
            cmap='seismic',annot=True,vmin=-1,vmax=1)
```

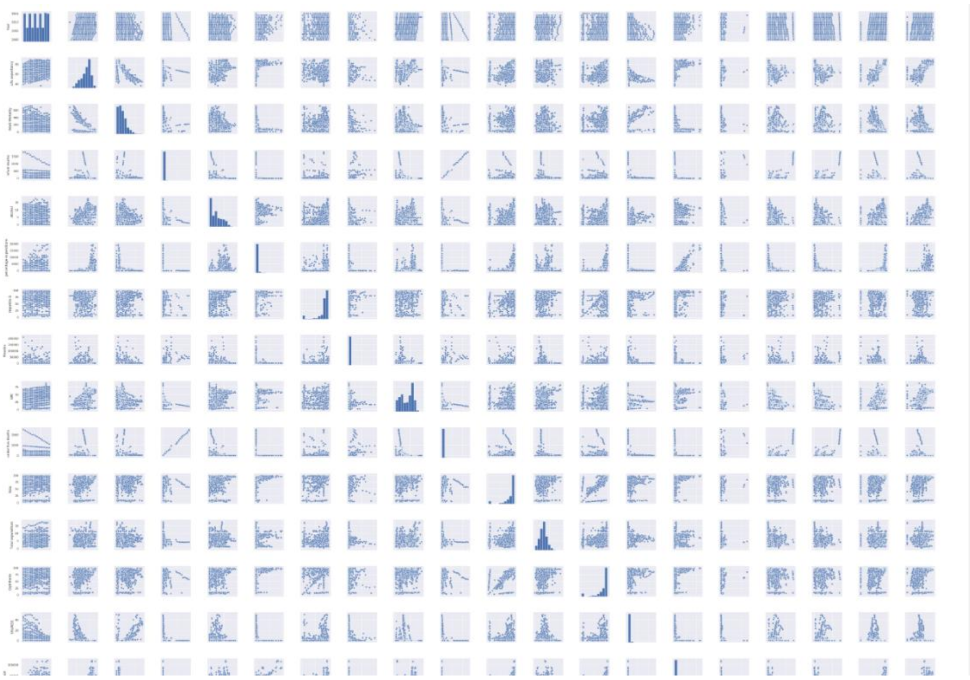
Out [9]:



In [10]:

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

Out [10]:



In [11]:
```
y = df["Life expectancy "]
x= df.drop(["Life expectancy ", "Country"], axis = 1)
x.head(5)
```
Out [11]:

	Year	Status	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDI
0	2015	Developing	263.0	62	0.01	71.279624	65.0	1154	19.1	83	6.0	8.16	65.0	0.1	584.25921
1	2014	Developing	271.0	64	0.01	73.523582	62.0	492	18.6	86	58.0	8.18	62.0	0.1	612.69651
2	2013	Developing	268.0	66	0.01	73.219243	64.0	430	18.1	89	62.0	8.13	64.0	0.1	631.74497
3	2012	Developing	272.0	69	0.01	78.184215	67.0	2787	17.6	93	67.0	8.52	67.0	0.1	669.959000
4	2011	Developing	275.0	71	0.01	7.097109	68.0	3013	17.2	97	68.0	7.87	68.0	0.1	63.537231

In [12]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
x["Status"] = labelencoder_X_1.fit_transform(x["Status"])
```
In [13]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0
.2, random_state = 0)
```
In [14]:
```
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
```
In [15]:
```
#Decsion Tree
from sklearn.tree import DecisionTreeRegressor
DT = DecisionTreeRegressor(max_depth=15, min_samples_leaf=100)
DT.fit(x_train,y_train)
cv_score= np.sqrt(-cross_val_score(DT,x_train,y_train, cv=10, scoring='
neg_mean_squared_error'))
rmse = np.mean(cv_score)
```

```
print(rmse)
3.5750783942462334
```

In [16]:

```
Random Forest
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators = 100, random_state = 42)
rf.fit(x_train,y_train)
cv_score= np.sqrt(-cross_val_score(rf,x_train,y_train, cv=10, scoring='
neg_mean_squared_error'))
rmse = np.mean(cv_score)
print(rmse)
1.878059235090641
```

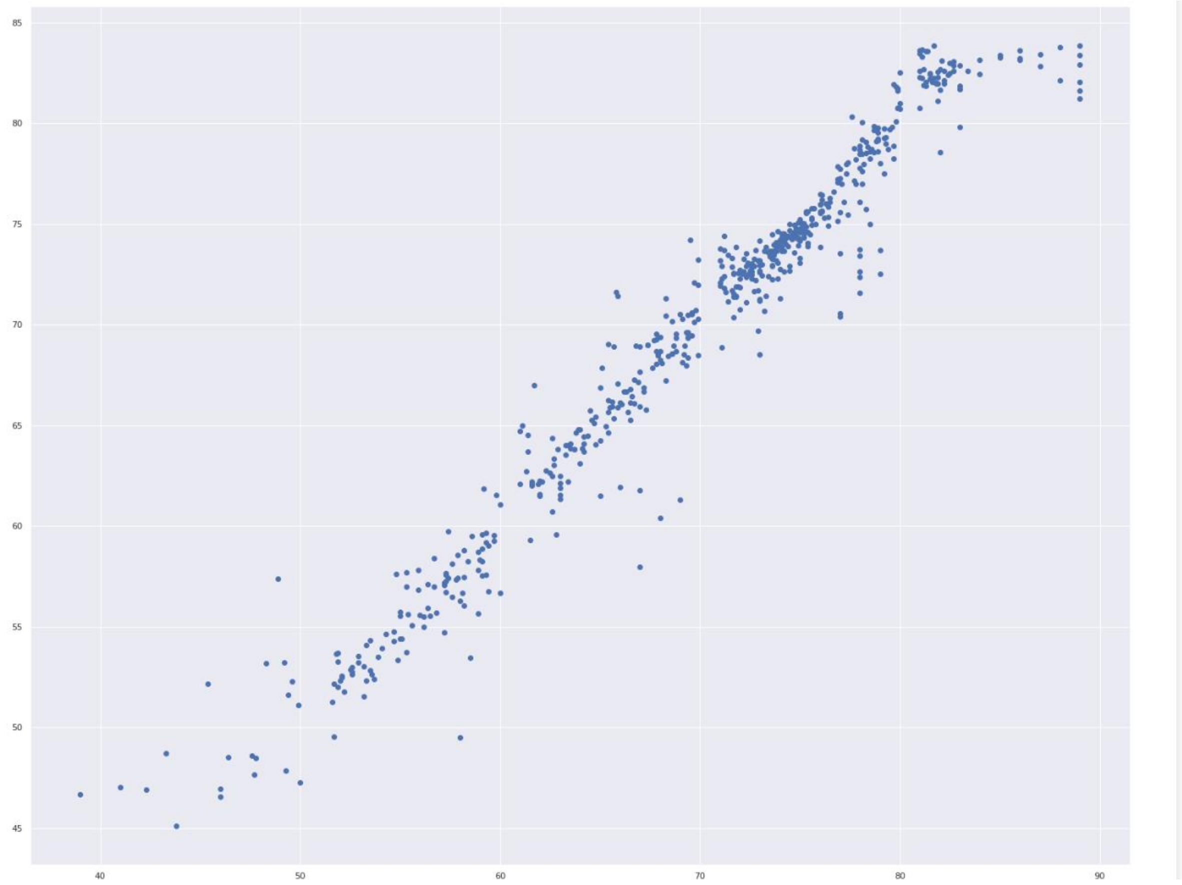
In [17]:

```
from sklearn import metrics
prediction = rf.predict(x_test)
np.sqrt(metrics.mean_squared_error(y_test, prediction))
1.92662552242872
```

In [18]:

```
plt.scatter(y_test,prediction)
```

Out [18]:



In [19]:

```
from watson_machine_learning_client import WatsonMachineLearningAPIClie
nt
```

In [20]:

```
wml_credentials = {
  "apikey": "5QWkze9fQzBn5KB6WytKrf7FqdVQIAhgUACn_znqB_JU",
```

```
  "iam_apikey_description": "Auto-generated for key b4ec8787-b995-4fcd-85e8-76d31fd2540e",
  "iam_apikey_name": "Service credentials-2",
  "iam_role_crn": "crn:v1:bluemix:public:iam:::serviceRole:Writer",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/cce355ed39684472b1bfba7441836eb5::serviceid:ServiceId-5abfa71a-b73a-4eef-afae-11be95bd0b6a",
  "instance_id": "784c1a90-ec07-4f99-9003-1a281662cd53",
  "url": "https://eu-gb.ml.cloud.ibm.com"
}
```

In [21]:

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

In [22]:

```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Bhanu",
               client.repository.ModelMetaNames.AUTHOR_EMAIL: "bhanuprakash0912@gmail.com",
               client.repository.ModelMetaNames.NAME: "PredictingLifeExpectancyWithPython"}
```

In [23]:

```
model_artifact =client.repository.store_model(model, meta_props=model_props)
```

In [24]:

```
published_model_uid = client.repository.get_model_uid(model_artifact)
published_model_uid
```

Out [24]:

```
'5dff35fb-18d5-4a34-ad03-5c23dc57336c'
```

In [25]:

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

Out [25]:

```
#######################################################################################

Synchronous deployment creation for uid: '5dff35fb-18d5-4a34-ad03-5c23dc57336c' started

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

INITIALIZING
DEPLOY_SUCCESS

------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='033f5e75-37dd-4b6b-8400-e15556cadeb4'
------------------------------------------------------------------------------------------------
```

In [26]:

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

Out [26]:

```
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/a0f27979-0907-4dca-84d1-eddf235c1271/deployments/033f5e75-37dd-4b6b-8400-e15556cadeb4/online'
```

2)Node-RED Flow .Json File: -

Jupyter.json

[{"id":"d2ef81d5.bf34d","type":"tab","label":"Predicting Life Expectancy With
Python","disabled":true,"info":""},{"id":"58fb4084.ae5c2","type":"ui_form","z":"d2ef81d5.b
f34d","name":"","label":"","group":"7fbba1c4.4643e","order":0,"width":0,"height":0,"options
":[{"label":"Adult
Mortality","value":"a","type":"number","required":true,"rows":null},{"label":"infant
deaths","value":"b","type":"number","required":true,"rows":null},{"label":"Alcohol","value":
"c","type":"number","required":true,"rows":null},{"label":"percentage
expenditure","value":"d","type":"number","required":true,"rows":null},{"label":"Hepatitis
B","value":"e","type":"number","required":true,"rows":null},{"label":"Measles
","value":"f","type":"number","required":true,"rows":null},{"label":" BMI
","value":"g","type":"number","required":true,"rows":null},{"label":"under-five deaths
","value":"h","type":"number","required":true,"rows":null},{"label":"Polio","value":"i","type
":"number","required":true,"rows":null},{"label":"Total
expenditure","value":"j","type":"number","required":true,"rows":null},{"label":"Diphtheria
","value":"k","type":"number","required":true,"rows":null},{"label":"
HIV/AIDS","value":"l","type":"number","required":true,"rows":null},{"label":"GDP","value
":"m","type":"number","required":true,"rows":null},{"label":"Population","value":"n","type":
"number","required":true,"rows":null},{"label":" thinness 1-19
years","value":"o","type":"number","required":true,"rows":null},{"label":" thinness 5-9
years","value":"p","type":"number","required":true,"rows":null},{"label":"Income
composition of
resources","value":"q","type":"number","required":true,"rows":null},{"label":"Schooling","v
alue":"r","type":"number","required":true,"rows":null},{"label":"Developed","value":"s","typ
e":"number","required":true,"rows":null},{"label":"Developing","value":"t","type":"number",
"required":true,"rows":null}],"formValue":{"a":"","b":"","c":"","d":"","e":"","f":"","g":"","h"
:"","i":"","j":"","k":"","l":"","m":"","n":"","o":"","p":"","q":"","r":"","s":"","t":""},"payload":"
","submit":"Predict","cancel":"cancel","topic":"","x":150,"y":180,"wires":[["59265cc9.e363c
4"]]},{"id":"59265cc9.e363c4","type":"function","z":"d2ef81d5.bf34d","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);\nglobal.set(\"d\",msg.payload.d);\nglobal.set(\"e\",msg.payload.e);\nglobal.s
et(\"f\",msg.payload.f);\nglobal.set(\"g\",msg.payload.g);\nglobal.set(\"h\",msg.payload.h);\n
global.set(\"i\",msg.payload.i);\nglobal.set(\"j\",msg.payload.j);\nglobal.set(\"k\",msg.payload
.k);\nglobal.set(\"l\",msg.payload.l);\nglobal.set(\"m\",msg.payload.m);\nglobal.set(\"n\",msg.
payload.n);\nglobal.set(\"o\",msg.payload.o);\nglobal.set(\"p\",msg.payload.p);\nglobal.set(\"
q\",msg.payload.q);\nglobal.set(\"r\",msg.payload.r);\nglobal.set(\"s\",msg.payload.s);\ngloba
l.set(\"t\",msg.payload.t);\n\n\n\n\n\n\n\n\n//following are required to receive a token\nvar
apikey=\"NQkBB6-
vvr8QffziNL3vqTcz8awbjSUKfbQVlSNozGax\";\nmsg.headers={\"content-
type\":\"application/x-www-form-
urlencoded\"};\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-
type:apikey\",\"apikey\":apikey};\nreturn
msg;\n","outputs":1,"noerr":0,"x":420,"y":240,"wires":[["4df18024.75b86"]]},{"id":"7ac5d4f
6.b6caac","type":"http

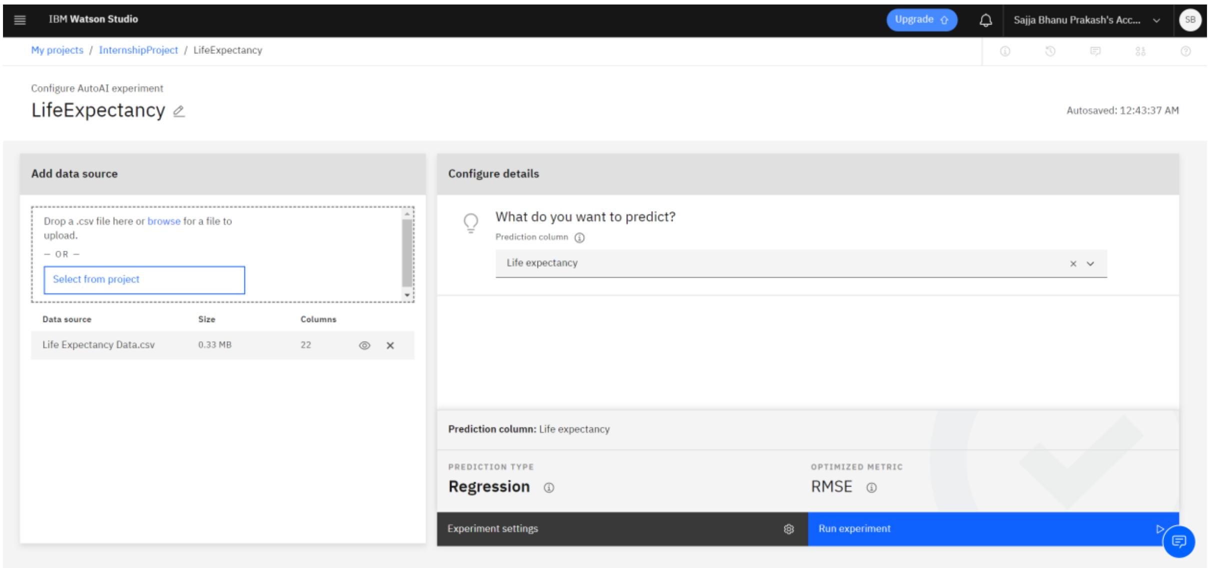
request","z":"d2ef81d5.bf34d","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://us-south.ml.cloud.ibm.com/v3/wml_instances/3a7111b6-225d-43df-bcbe-53d193704c48/deployments/76d6eda7-bcf3-4a57-81b1-092511242aa8/online","tls":"","persist":false,"proxy":"","authType":"","x":690,"y":300,"wires":[["4bdf7624.434c28"]]},{"id":"20dd1a07.5c2a86","type":"debug","z":"d2ef81d5.bf34d","name":"","active":false,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":1010,"y":320,"wires":[]},{"id":"4bdf7624.434c28","type":"function","z":"d2ef81d5.bf34d","name":"getFrom Endpoint","func":"msg.payload=msg.payload.values[0][0];\nreturn msg;","outputs":1,"noerr":0,"x":830,"y":380,"wires":[["20dd1a07.5c2a86","a05765f5.3330d8"]]},{"id":"17cf0cfa.82db83","type":"function","z":"d2ef81d5.bf34d","name":"sendTo Endpoint","func":"//get token and make headers\nvar token=msg.payload.access_token;\nvar instance_id=\"3a7111b6-225d-43df-bcbe-53d193704c48\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance_id}\n\n//get variables that are set earlier\nvar a = global.get(\"a\");\nvar b = global.get(\"b\");\nvar c = global.get(\"c\");\nvar d = global.get(\"d\");\nvar e = global.get(\"e\");\nvar f = global.get(\"f\");\nvar g = global.get(\"g\");\nvar h = global.get(\"h\");\nvar i = global.get(\"i\");\nvar j = global.get(\"j\");\nvar k = global.get(\"k\");\nvar l = global.get(\"l\");\nvar m = global.get(\"m\");\nvar n = global.get(\"n\");\nvar o = global.get(\"o\");\nvar p = global.get(\"p\");\nvar q = global.get(\"q\");\nvar r = global.get(\"r\");\nvar s = global.get(\"s\");\nvar t = global.get(\"t\");\n\n//send the user values to service endpoint\nmsg.payload = \n{\"fields\":['Adult Mortality', 'infant deaths', 'Alcohol',\n   'percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ',\n   'under-five deaths ', 'Polio', 'Total expenditure', 'Diphtheria ',\n   ' HIV/AIDS', 'GDP', 'Population', ' thinness 1-19 years',\n   ' thinness 5-9 years', 'Income composition of resources', 'Schooling',\n   'Developed', 'Developing'],\n\"values\":[[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t]]};\n\nreturn msg;\n","outputs":1,"noerr":0,"x":770,"y":160,"wires":[["7ac5d4f6.b6caac"]]},{"id":"4df18024.75b86","type":"http request","z":"d2ef81d5.bf34d","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"basic","x":530,"y":160,"wires":[["17cf0cfa.82db83"]]},{"id":"a05765f5.3330d8","type":"ui_text","z":"d2ef81d5.bf34d","group":"7fbba1c4.4643e","order":1,"width":0,"height":0,"name":"","label":"Prediction","format":"{{msg.payload}}","layout":"row-spread","x":1000,"y":440,"wires":[]},{"id":"7fbba1c4.4643e","type":"ui_group","z":"","name":"Machine Learning Model","tab":"65434790.c91f18","order":1,"disp":true,"width":"6","collapse":false},{"id":"65434790.c91f18","type":"ui_tab","z":"","name":"Home Page","icon":"dashboard","disabled":false,"hidden":false}]

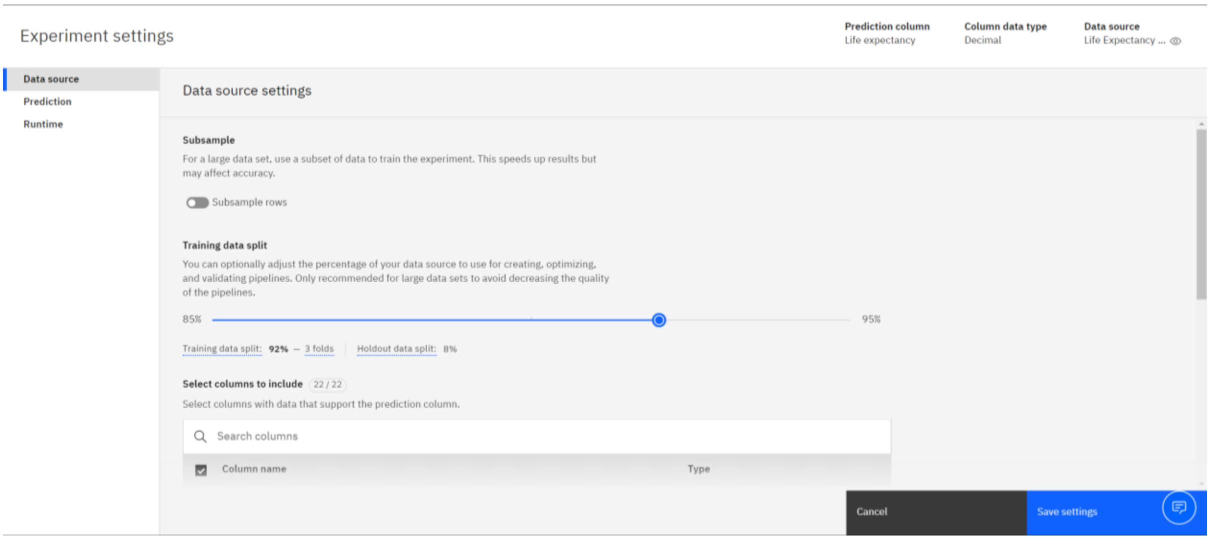
3)Node-RED URL: -

https://sajja-bhanu-prakash.eu-gb.mybluemix.net/ui/#!/0?socketid=8i3yfC8PkKHnAvKpAAAN

4)Auto-AI: -

Auto AI Configuration





5)Node-RED Flow. Json File: -

Autoai.json

[{"id":"356fcd2f.6fb592","type":"tab","label":"Predicting Life Expectancy Without Python","disabled":false,"info":""},{"id":"1c2628d3.0fad77","type":"function","z":"356fcd2f.6fb592","name":"pre token","func":"global.set('country', msg.payload.country);\nglobal.set('year', msg.payload.year);\nglobal.set('status', msg.payload.stat);\nglobal.set('am', msg.payload.am);\nglobal.set('ufd', msg.payload.ufd);\nglobal.set('idr', msg.payload.idr);\nglobal.set('hepb', msg.payload.hepb);\nglobal.set('mesls', msg.payload.mesls);\nglobal.set('polio', msg.payload.polio);\nglobal.set('dipt', msg.payload.dipt);\nglobal.set('hiv', msg.payload.hiv);\nglobal.set('popn', msg.payload.popn);\nglobal.set('bmi', msg.payload.bmi);\nglobal.set('thin5', msg.payload.thin5);\nglobal.set('thin10',

msg.payload.thin10);\nglobal.set('scl', msg.payload.scl);\nglobal.set('alc', msg.payload.alc);\nglobal.set('gdp',
msg.payload.gdp);\nglobal.set('perexp', msg.payload.perexp);\nglobal.set('totexp', msg.payload.totexp);\nglobal.set('icr',
msg.payload.icr);\n\n\n\n\n\n//following are required to receive a token\nvar
apikey=\"j8ShAS_2esqevTx7EvUgowFLHLlkvsOhpfNVId2XwDwb\";\nmsg.headers={\"content-type\":\"application/x-
www-form-urlencoded\"}\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-
type:apikey\",\"apikey\":apikey}\nreturn
msg;\n","outputs":1,"noerr":0,"x":200,"y":380,"wires":[["2095b8e5.e98168"]]},{"id":"180f4a94.e90435","type":"http
request","z":"356fcd2f.6fb592","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://eu-
gb.ml.cloud.ibm.com/v4/deployments/50b7eceb-2268-4922-8489-
f7f9a801b644/predictions","tls":"","persist":false,"proxy":"","authType":"","x":710,"y":280,"wires":[["728efeef.25065","ba
16e537.8b50b8"]]},{"id":"1dca7c34.a86e44","type":"debug","z":"356fcd2f.6fb592","name":"","active":false,"tosidebar":tru
e,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":1010,"y":400,"wires":[]},{"id":"728efeef.25
065","type":"function","z":"356fcd2f.6fb592","name":"getFrom Endpoint","func":"msg.payload =
msg.payload.predictions[0].values[0][0]+' years';\nreturn
msg;","outputs":1,"noerr":0,"x":770,"y":460,"wires":[["1dca7c34.a86e44","6f58acdf.3788a4"]]},{"id":"2095b8e5.e98168","
type":"http
request","z":"356fcd2f.6fb592","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.ibm.com/id
entity/token","tls":"","persist":false,"proxy":"","authType":"","x":390,"y":320,"wires":[["5c2e4d83.471814"]]},{"id":"6f58a
cdf.3788a4","type":"ui_text","z":"356fcd2f.6fb592","group":"e358087e.b44258","order":1,"width":0,"height":0,"name":"","
label":"Prediction","format":"{{msg.payload}}","layout":"row-
spread","x":1000,"y":520,"wires":[]},{"id":"5c2e4d83.471814","type":"function","z":"356fcd2f.6fb592","name":"sendTo
Endpoint","func":"var country=global.get('country');\nvar year=global.get('year');\nvar stat=global.get('stat');\nvar
am=global.get('am');\nvar ufd=global.get('ufd');\nvar idr=global.get('idr');\nvar hepb=global.get('hepb');\nvar
mesls=global.get('mesls');\nvar polio=global.get('polio');\nvar dipt=global.get('dipt');\nvar hiv=global.get('hiv');\nvar
popn=global.get('popn');\nvar bmi=global.get('bmi');\nvar thin5=global.get('thin5');\nvar thin10=global.get('thin10');\nvar
scl=global.get('scl');\nvar alc=global.get('alc');\nvar gdp=global.get('gdp');\nvar perexp=global.get('perexp');\nvar
totexp=global.get('totexp');\nvar icr=global.get('icr');\nvar token=msg.payload.access_token;\nvar instance_id=\"b79d237d-
5fa7-4286-b0bf-9049559d3930\"\nmsg.headers = {\n \"Content-type\":'application/json',\n \"Authorization\":
'Bearer'+token,\n \"ML-Instance-ID\": instance_id\n}\nmsg.payload = {\n \"input_data\": [{\"fields\": [\"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\"], \n
\"values\": [[country, year, stat, am, idr, alc, perexp, hepb, mesls, bmi, ufd, polio, totexp, dipt, hiv, gdp, popn, thin10, thin5,
icr, scl]]}]\n}\nreturn
msg;","outputs":1,"noerr":0,"x":550,"y":200,"wires":[["180f4a94.e90435"]]},{"id":"ba16e537.8b50b8","type":"debug","z":"
356fcd2f.6fb592","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":770,"y":12
0,"wires":[]},{"id":"7400ae76.68a6d","type":"ui_form","z":"356fcd2f.6fb592","name":"","label":"","group":"e358087e.b44
258","order":1,"width":0,"height":0,"options":[{"label":"Country","value":"country","type":"text","required":false,"rows":n
ull},{"label":"Year","value":"year","type":"number","required":false,"rows":null},{"label":"Status","value":"stat","type":"te
xt","required":false,"rows":null},{"label":"Adult
Mortality","value":"am","type":"number","required":true,"rows":null},{"label":"Under-five
Deaths","value":"ufd","type":"number","required":true,"rows":null},{"label":"Infant
Deaths","value":"idr","type":"number","required":true,"rows":null},{"label":"Hepatitis
B","value":"hepb","type":"number","required":true,"rows":null},{"label":"Measles","value":"mesls","type":"number","requi
red":true,"rows":null},{"label":"Diptheria","value":"dipt","type":"number","required":true,"rows":null},{"label":"HIV/AIDS
","value":"hiv","type":"number","required":true,"rows":null},{"label":"Polio","value":"polio","type":"number","required":tr
ue,"rows":null},{"label":"Population","value":"popn","type":"number","required":true,"rows":null},{"label":"BMI","value":

"bmi","type":"number","required":true,"rows":null},{"label":"Thinness 5-9 years","value":"thin5","type":"number","required":true,"rows":null},{"label":"Thinness 10-19 years","value":"thin10","type":"number","required":true,"rows":null},{"label":"Schooling","value":"scl","type":"number","required":true,"rows":null},{"label":"Alcohol","value":"alc","type":"number","required":true,"rows":null},{"label":"GDP","value":"gdp","type":"number","required":true,"rows":null},{"label":"Percentage Expenditure","value":"perexp","type":"number","required":true,"rows":null},{"label":"Total Expenditure","value":"totexp","type":"number","required":true,"rows":null},{"label":"Income Composition of Resources","value":"icr","type":"number","required":true,"rows":null}],"formValue":{"country":"","year":"","stat":"","am":"","ufd":"","idr":"","hepb":"","mesls":"","dipt":"","hiv":"","polio":"","popn":"","bmi":"","thin5":"","thin10":"","scl":"","alc":"","gdp":"","perexp":"","totexp":"","icr":""},"payload":"","submit":"Predict","cancel":"Reset","topic":"","x":110,"y":260,"wires":[["1c2628d3.0fad77"]]},{"id":"e358087e.b44258","type":"ui_group","z":"","name":"IBM AutoAI Model","tab":"2f9a6cd4.bf9a74","order":1,"disp":true,"width":"6","collapse":false},{"id":"2f9a6cd4.bf9a74","type":"ui_tab","z":"","name":"Life Expectancy Prediction","icon":"dashboard","disabled":false,"hidden":false}]

6) Node-RED URL: -

https://sajja-bhanu-prakash.eu-gb.mybluemix.net/ui/#!/0?socketid=8i3yfC8PkKHnAvKpAAAN

For Code References Please Visit my GitHub Repo: -

https://github.com/SmartPracticeschool/llSPS-INT-2669-Predicting-Life-Expectancy-using-Machine-Learning