PROJECT REPORT

Name: Moksh Grover

Email: mokshmg@gmail.com

Title: PREDICTING LIFE EXPECTANCY OF A COUNTRY

Category: Machine Learning

Web Page Link:

https://node-red-dloyf.eu-gb.mybluemix.net/ui/#!/0?socketid=CBfqsvWgsOxyJr_MAAAy

INDEX

1. INTRODUCTION

1.1 Overview

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. It is very important to predict average life expectancy of a country to analyse further requirements to increase its rate of growth or stabilise the rate of growth in that country .So, the scope of this project is to create a model that can be used to predict the Life Expectancy of a country for the upcoming future using Multiple Regression Analysis technique.

The end product will be a webpage where you need to give all the required inputs and then submit it. Afterwards it will predict the life expectancy value based on your regression technique.

Project Requirements: Python, IBM Cloud, IBM Watson

Functional Requirements: IBM cloud

Technical Requirements: ML, WATSON Studio, Python, Node-Red

Software Requirements: Watson Studio, Node-Red

Project Deliverables: Smartinternz Internship

Project Team: Moksh Grover

1.2 Purpose

The purpose of the project is to design a model for predicting Life Expectancy rate of a country given various features 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.

2. LITERATURE SURVEY

2.1 Existing Problem

The typical regression model that can predict average life expectancy of the country based on some user inputted values such as GDP, BMI, HIV/AIDS, Year, Alcohol intake and etc.

2.2 Proposed Solution

Steps:

1. Create IBM cloud services
2. Configure Watson Studio
3. Create Node-Red Flow to connect all services together
4. Deploy and run Node-Red app

2.2.1. Create IBM cloud Services

- Watson Studio
- Machine Learning resource
- Node-Red

2.2.2. Configure Watson Studio

1. After creating all services, go to resource list and launch Watson studio then get started with Watson studio. Then create an empty project and add machine learning resource as associated services in settings. Create a token as editor type.
2. Then add dataset and empty Jupyter notebook into Assets.
3. After that go to notebook and write your code to build model and get the scoring endpoint URL.

Steps for notebook:

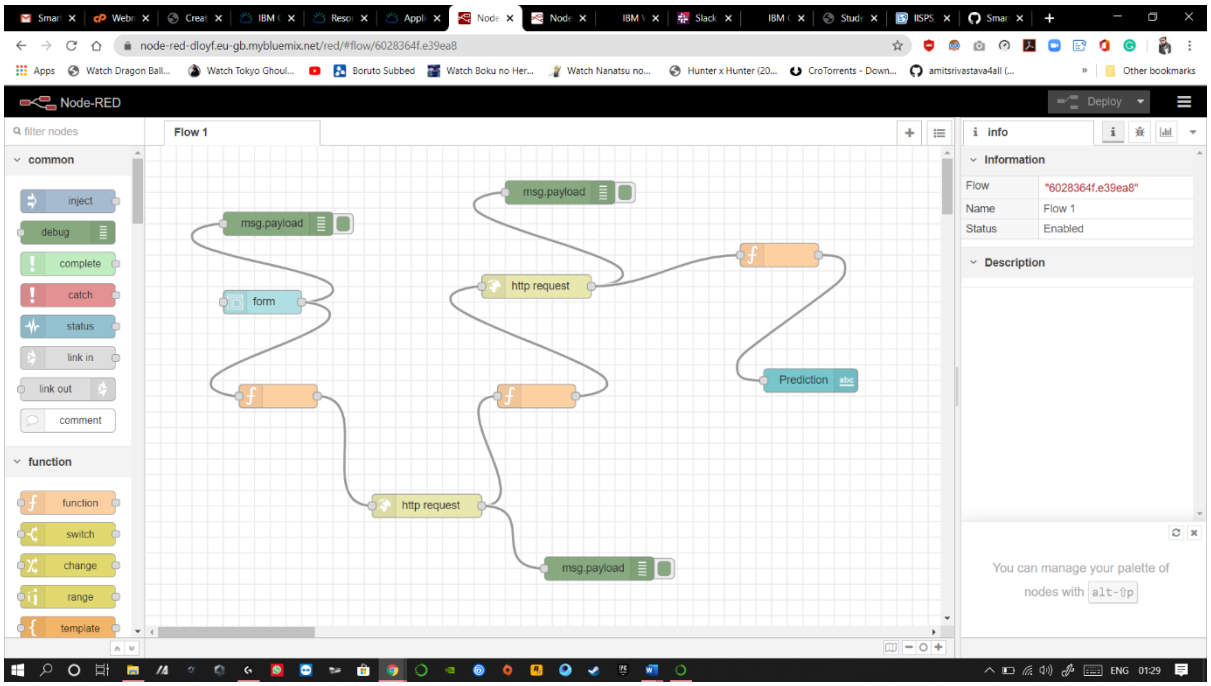
- Install Watson_machine_learning_client
- Import necessary libraries
- Import dataset
- Data Preprocessing
  - Replacing nan values if any with their mean values.
- Exploratory Data Analysis

- Plotting a heatmap to check if dimensional reduction can be performed.

- Train and Test
  The dataset was split into two parts i.e. Input and Output.
  As Life Expectancy needs to be predicted so it is to be treated as output and all other columns are treated as Input.
  Afterwards as we need regression technique to build our model so each and every column needs to be numeric. So, then we check for numeric and categoric columns and handle them using
  After that a regressor pipeline was designed using the regression technique. So,I have used MultiLinearRegressor technique of as my regression algorithm because it best fits my dataset. Then dataset was fitted and predicted. Where we obtained an accuracy of 96.73.
- Model Building and Deployment At first the machine learning service credentials was stored in a variable and passed into WatsonMachineLearningAPIClient.
  wml_credentials = {
  "apikey": "9G1kW3d3oxDMMCuPS4rTUdBe1BmPy-Ytm_4-991gwIfo",
  "iam_apikey_description": "Auto-generated for key ef07dea5-825e-4968-b73e-e1cfd2072a0e",
  "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/e6537b6021944416b6850964a7048cd1::serviceid:ServiceId-79e26329-b075-49cc-a2f0-27419a94dabd",
  "instance_id": "60506c7f-fdf6-42d7-b4b1-d0774c4e2e41",
  "url": "https://eu-gb.ml.cloud.ibm.com"
  }
- Then the model was deployed and scoring_endpoint url was generated.

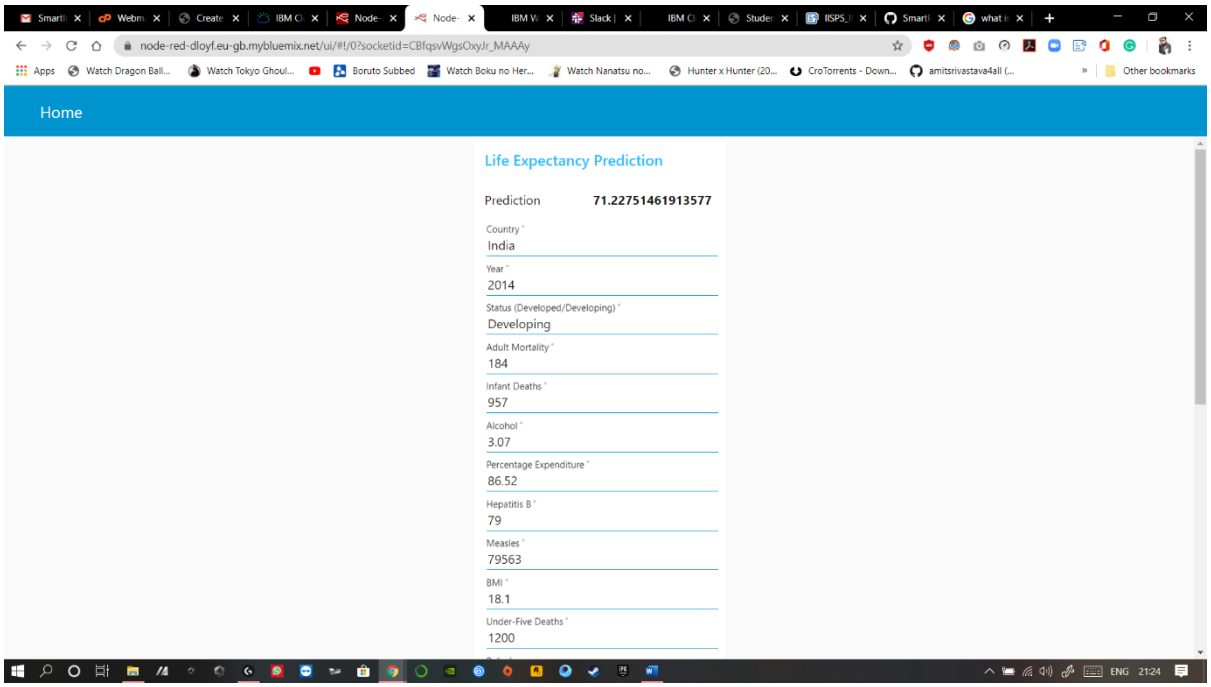
2.2.3. Create Node-Red Flow to connect all services together

- Go to Node-Red Editor from resource list.
- Install node-red Dashboard from manage pallete.
- Now create the flow with the help of following node.
  - Inject
  - Debug
  - Function
  - Ui_Form
  - Ui_Text



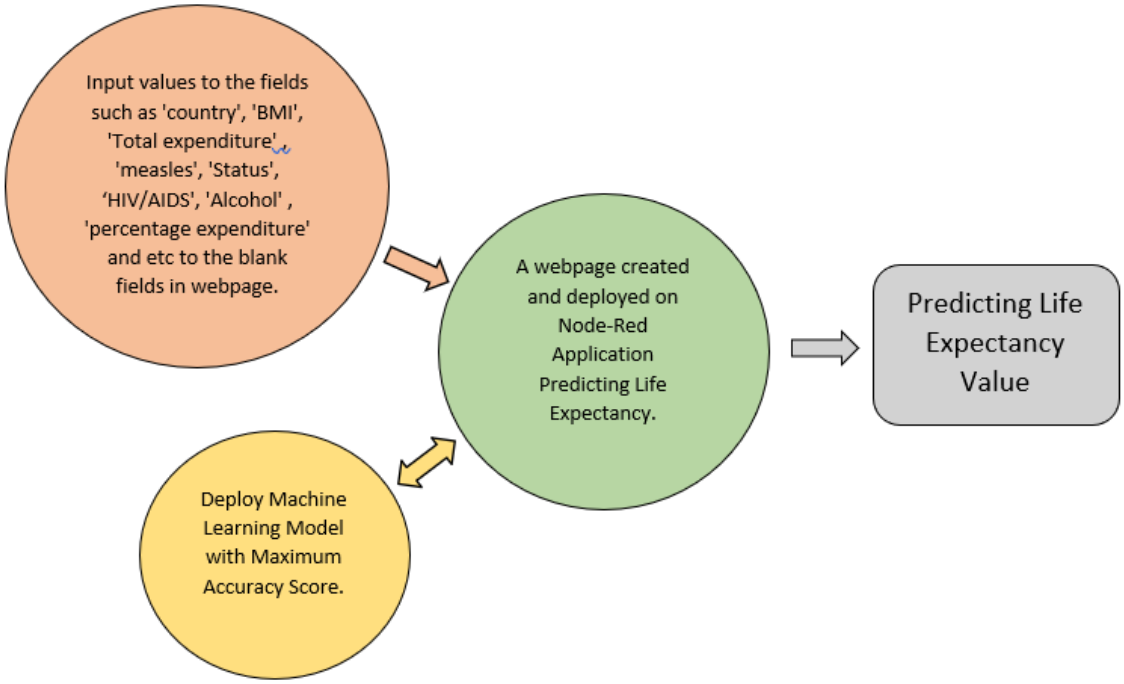
- Deploy and run Node Red app.

Deploy the Node Red flow. Then copy the link URL upto .net/ and paste at a new tab by UI at the end of the URL like this-



3. THEORETICAL ANALYSIS

3.1 Block Diagram

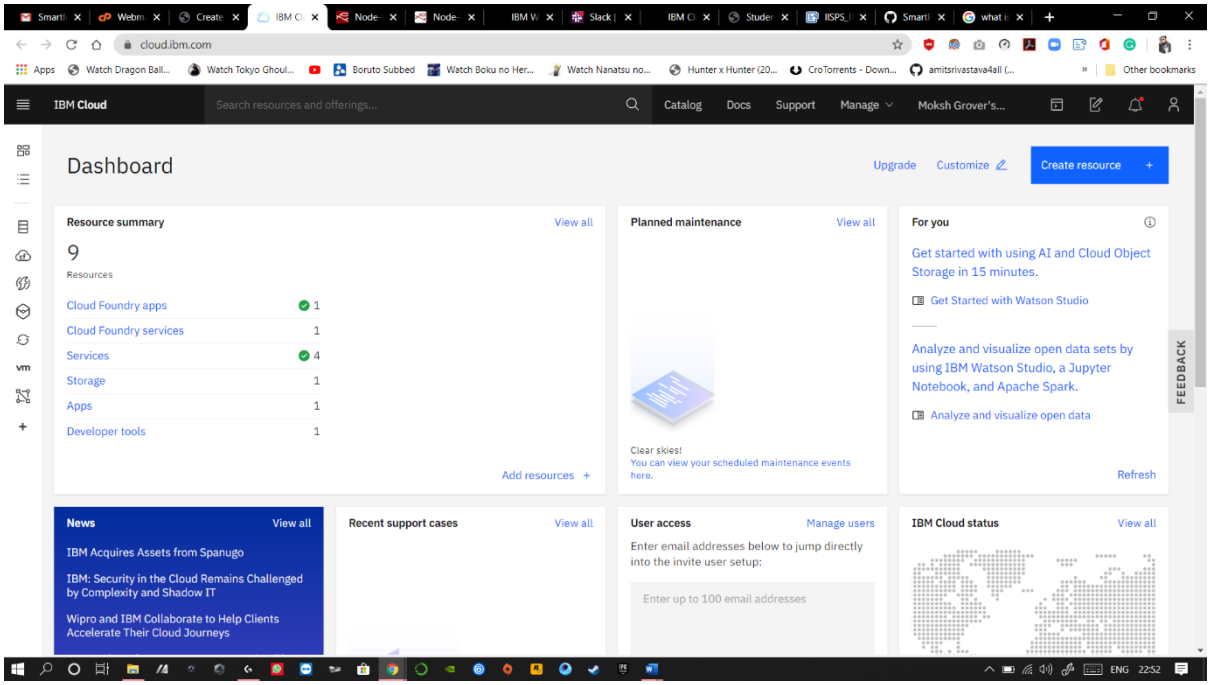


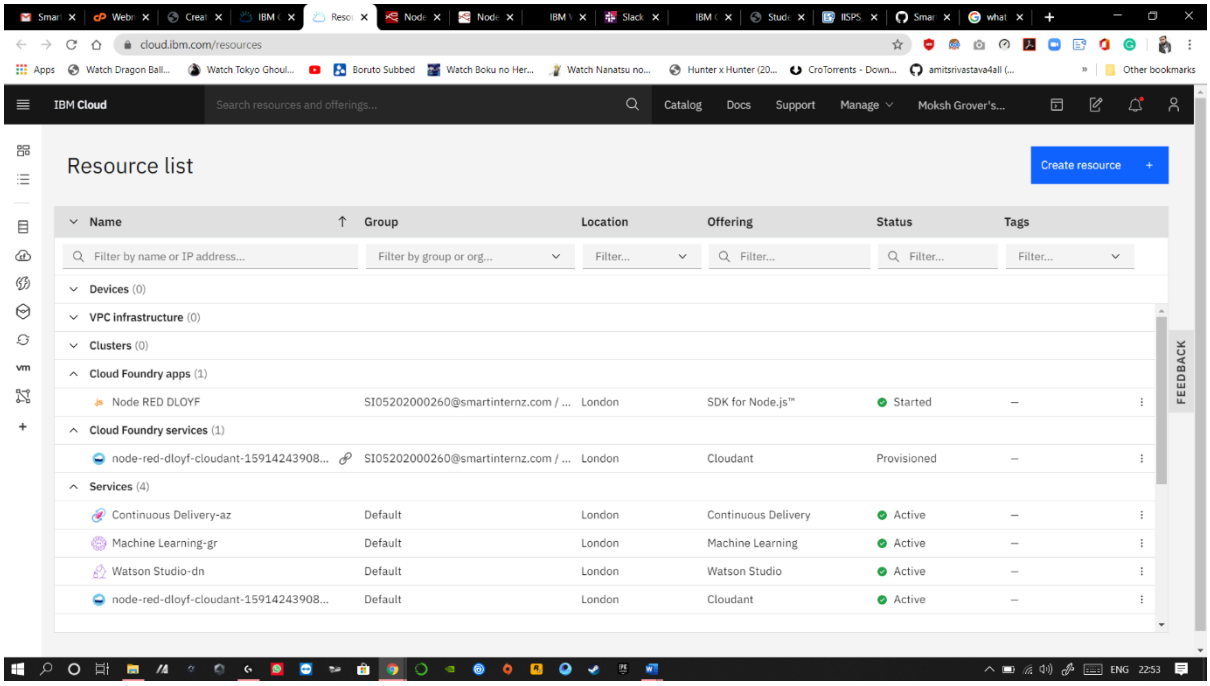
3.2 Hardware / Software Designing

- **Project Requirements:** Python, IBM Cloud, IBM Watson
- **Functional Requirements:** IBM cloud
- **Technical Requirements:** ML, WATSON Studio, Python, Node-Red
- **Software Requirements:** Watson Studio, Node-Red

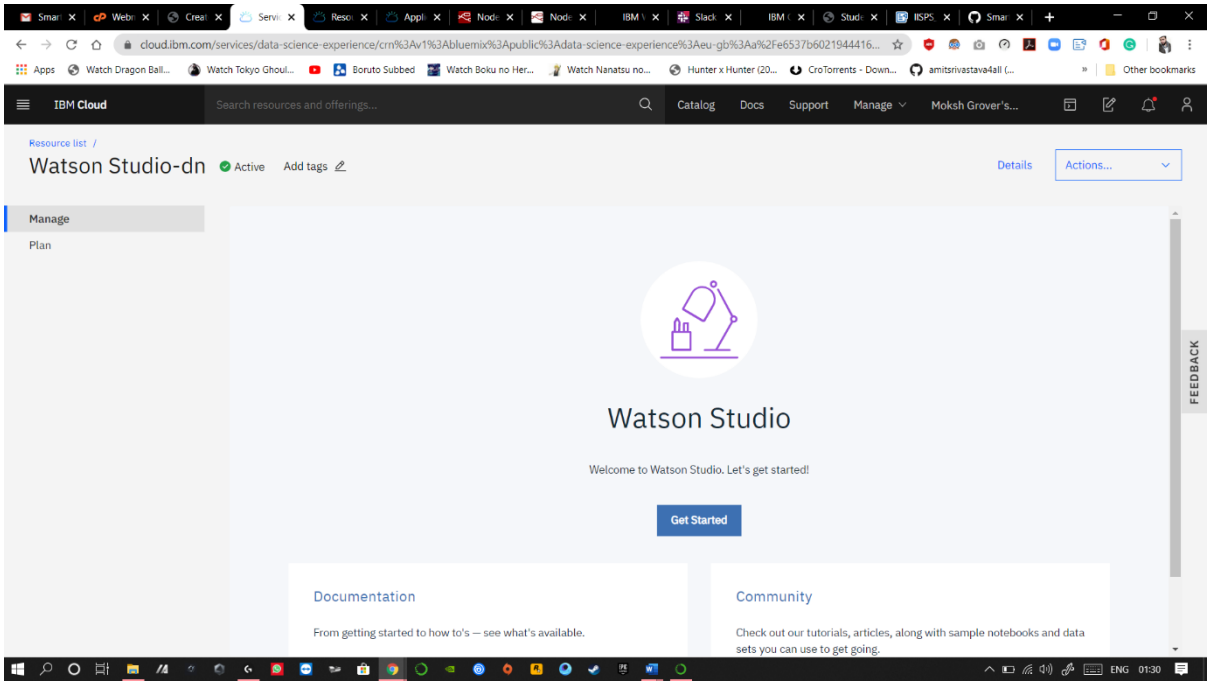
4. EXPERIMENTAL INVESTIGATION

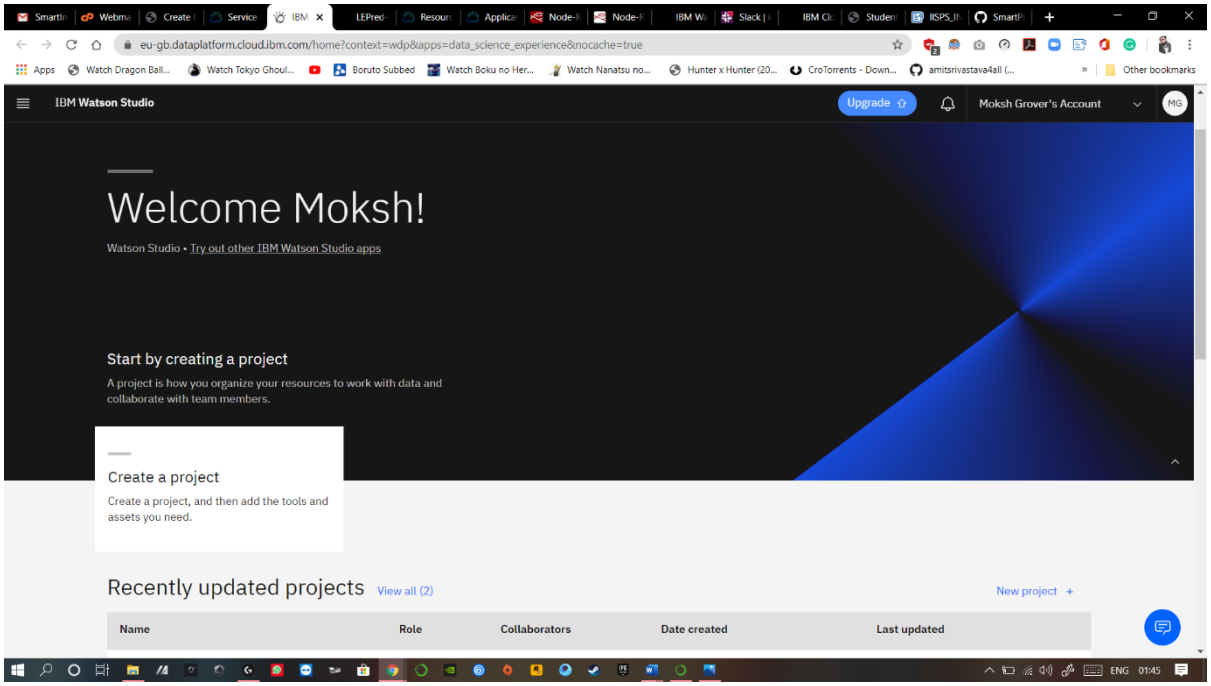
A. IBM Cloud Resource List



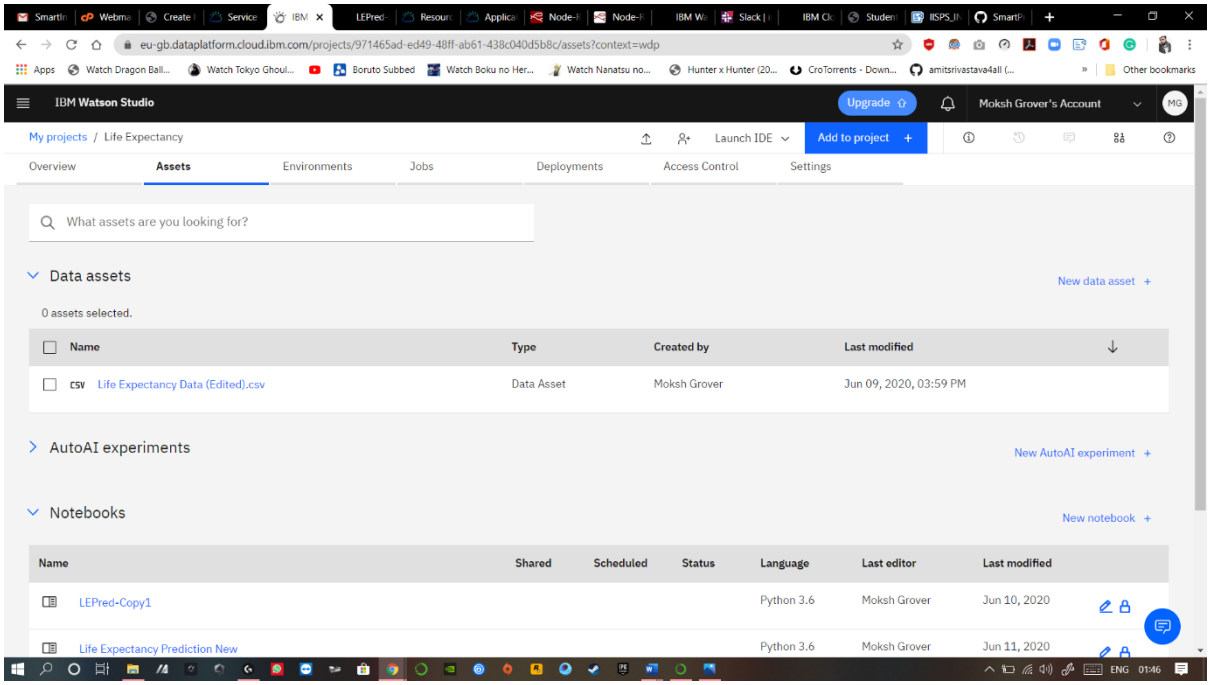


B. IBM Watson Studios

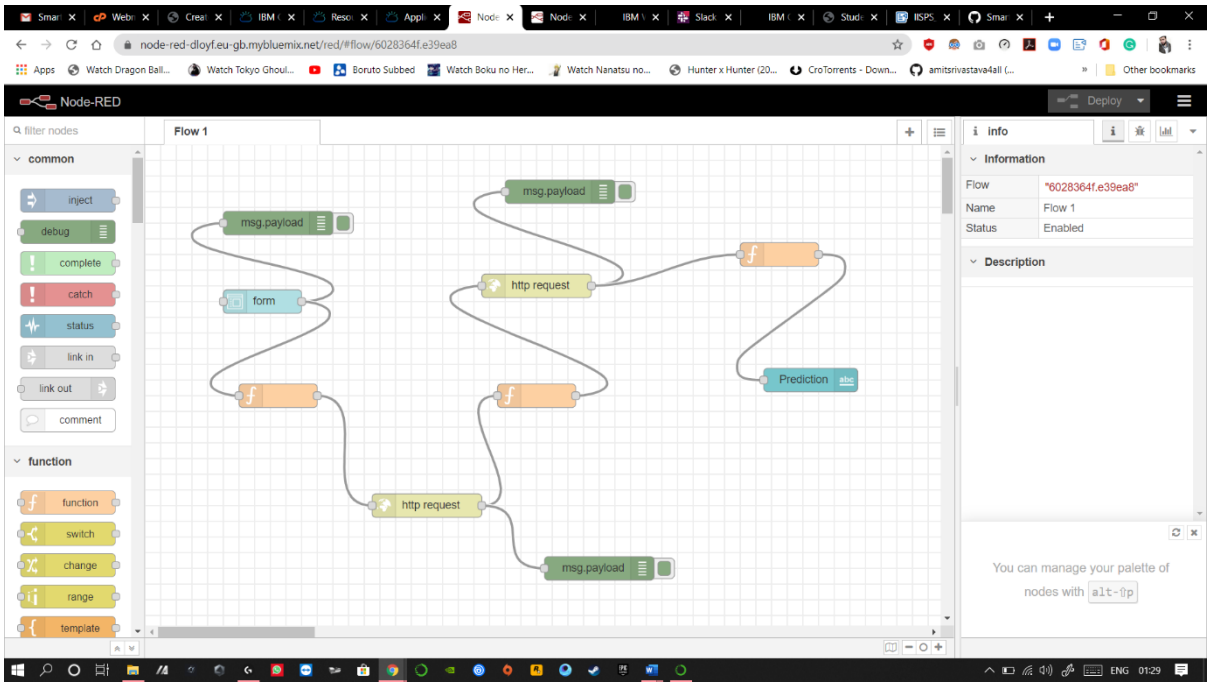




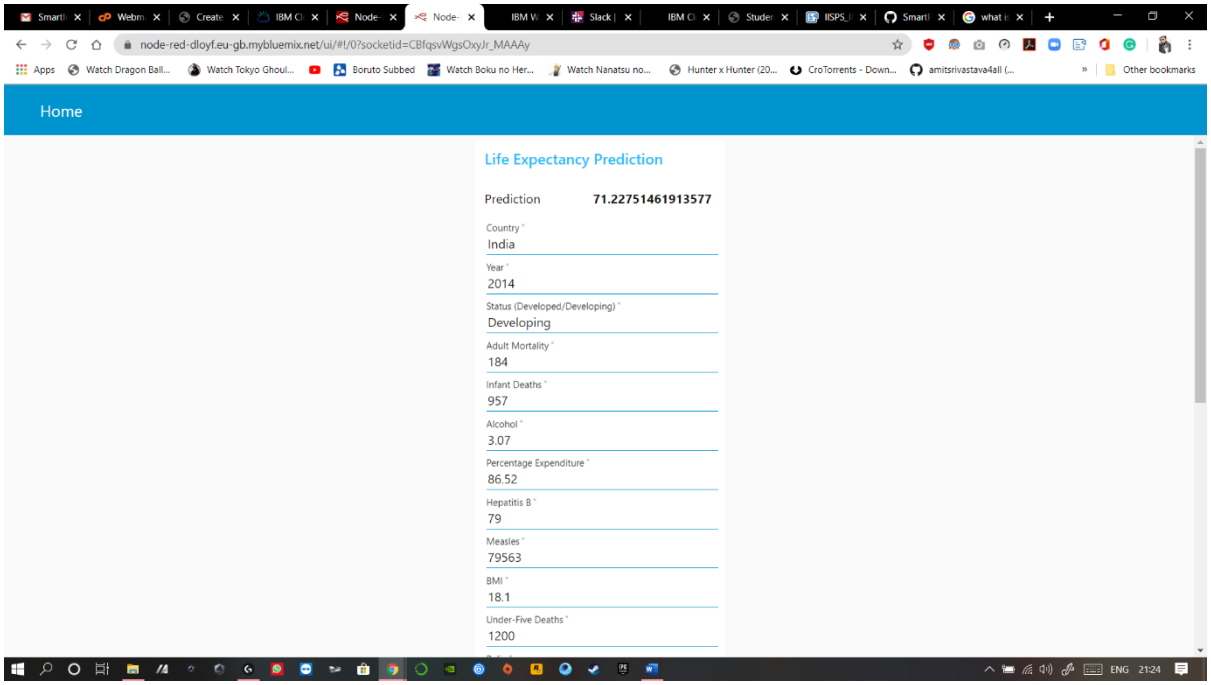
C. IMB Cloud Project Details



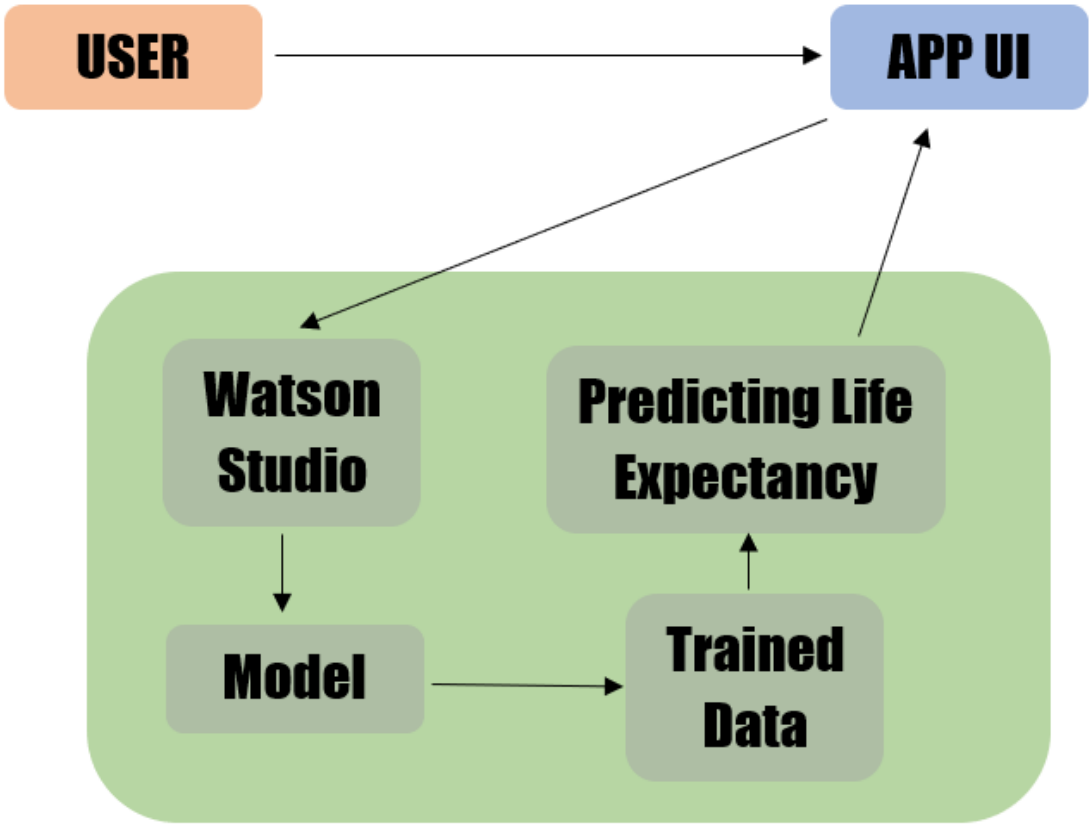
D. Node-Red Flow



E. Life Expectancy Prediction UI

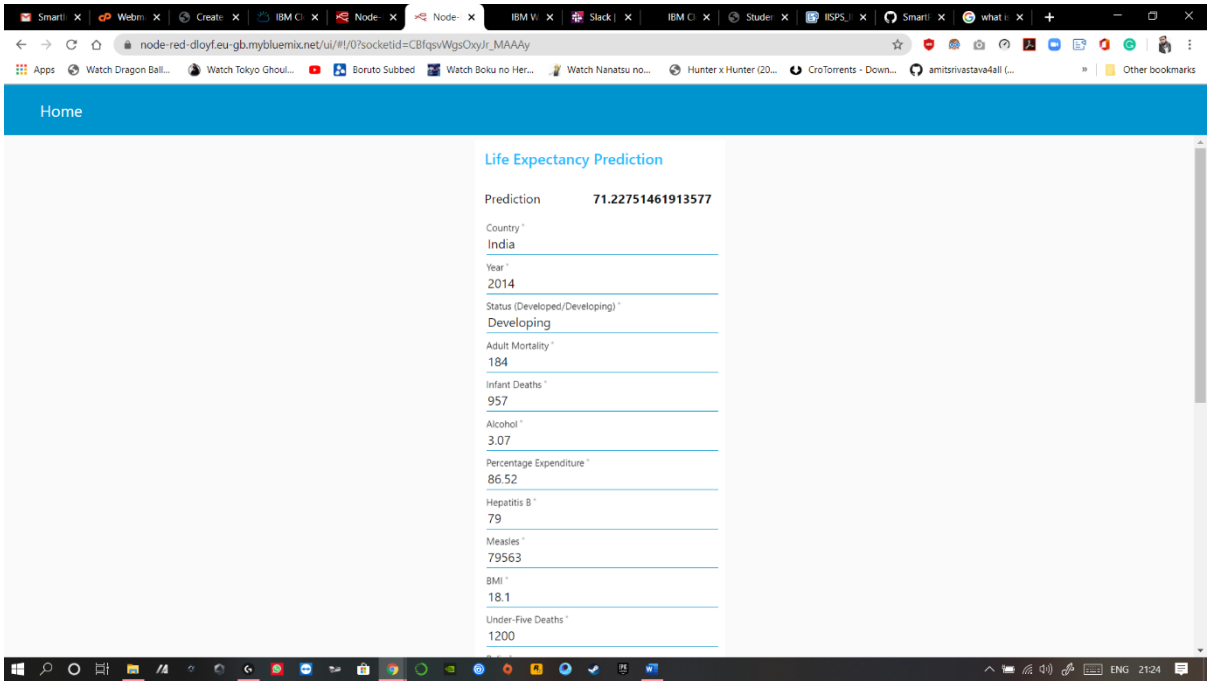


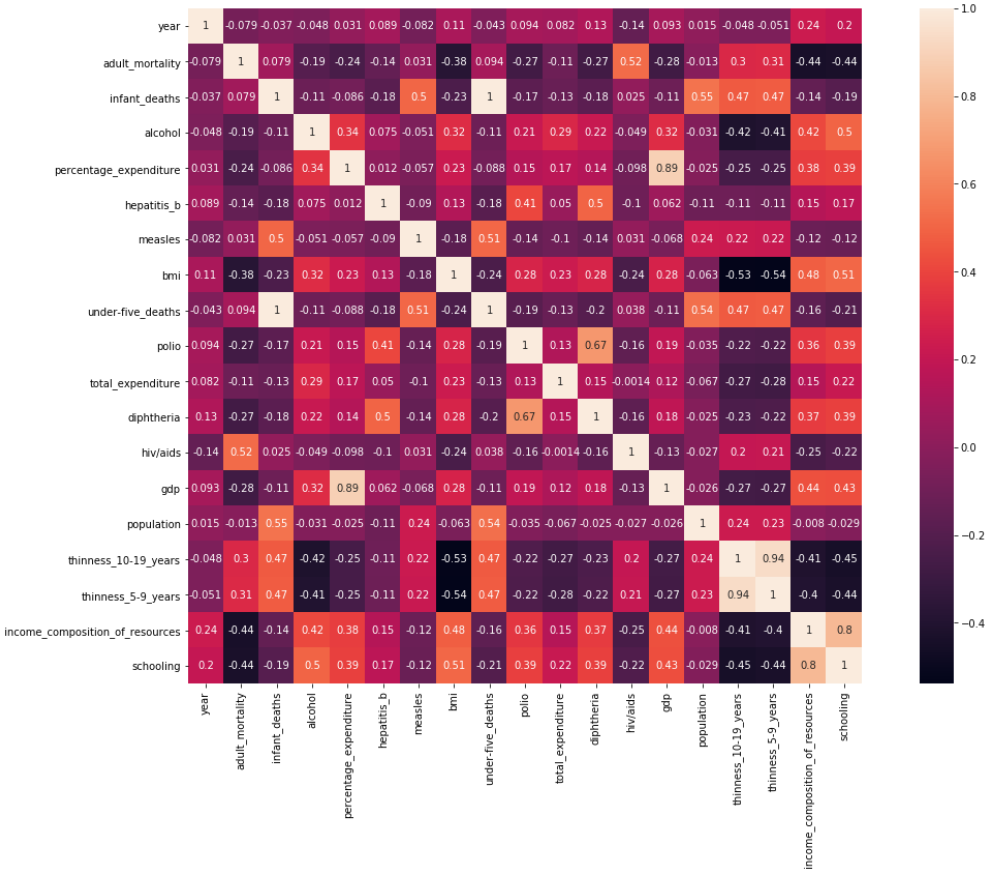
5. FLOWCHART



6. RESULT

Final result is this Life Expectancy Predictor User Interface-





7. ADVANTAGES & DISADVANTAGES

7.1 Advantages

1. Health Inequalities: Life expectancy has been used nationally to monitor health inequalities of a country.
2. Reduced Costs: This is a simple webpage and can be accessed by any citizen of a country to calculate life expectancy of their country and does not required any kind of payment neither for designing nor for using.
3. User Friendly Interface: This interface requires no background knowledge of how to use it. It's a simple interface and only ask for required values and predict the output.

7.2 Disadvantages

1. Wrong Prediction: As it depends completely on user, so if user provides some wrong values then it will predict wrong value.
2. Average Prediction: The model predicts average or approximate value with 97.07% accuracy but not accurate value.

8. APPLICATIONS

1. It can be used to monitor health inequalities of a country.
2. It is user friendly and can be used by anyone.
3. It can be used to develop statistics for country development process.
4. It can be used to analyse the factors for high life expectancy.

9. CONCLUSION

This user interface will be useful for the user to predict life expectancy value of their own country or any other country based on some required details such as GDP, BMI, Year, Alcohol Intake, Total expenditure and etc.

10. FUTURE SCOPE

Future Scope of the Model can be:

1. **Attractive UI-**

   It is a simple webpage only asking inputs and predict output. In future I have decided to make it more user friendly by providing some

useful information about the country in the webpage itself so that user does not need to do any kind of prior research for the values.

2. **Feature Reduction-**

   It requires much more data about 21 columns to be known prior for predicting life expectancy which can be again difficult for a normal user to gather such data so we can do some kind of feature reduction or replacement of some features as individuals or groups to make it more user friendly.

3. Integrating with services such as **speech recognition.**

11. BIBLOGRAPHY

- https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform
- https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
- https://nodered.org/
- https://github.com/watson-developer-cloud/node-red-labs
- https://www.youtube.com/embed/r7E1TJ1HtM0
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html
- https://www.kaggle.com/kumarajarshi/life-expectancy-who
- https://www.youtube.com/watch?v=DBRGlAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23
- https://www.youtube.com/watch?v=CUi8GezG1I&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L&index=2
- https://www.youtube.com/watch?v=Jtej3Y6uUng
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service
- https://machinelearningmastery.com/columntransformer-for-numerical-and-categorical-data/

APPENDIX

A. Source Code

- **Watson Studio Notebook**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as mp
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score
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_90e79a690de24db4858b1f884d0c43c2 = ibm_boto3.client(service_name
='s3',
    ibm_api_key_id='2qkO7L0LTKHsulkX8g0JHtzAwFtx9Ty8GSGXFWdm77lX',
    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_90e79a690de24db4858b1f884d0c43c2.get_object(Bucket='lifee
xpectancy-donotdelete-pr-zqnwsro8jcori0',Key='Life Expectancy Data (Edi
ted).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 )

dataset = pd.read_csv(body)

value_counts = dataset['Country'].value_counts()
to_remove = value_counts[value_counts == 1].index

dataset = dataset[~dataset.Country.isin(to_remove)]

df = dataset

countries = pd.unique(dataset['Country'])

country_index = {k: v for v, k in enumerate(countries)}
df['Country'] = df['Country'].replace(country_index, regex=True)

train = df[df['Year'] != 2015].values
```

```
X_train = train[:,:-1]

y_train = train[:,-1]

test = df[df['Year'] == 2015].values

X_test = test[:,:-1]

y_test = test[:,-1]

imputer = SimpleImputer(missing_values=np.nan , strategy = 'constant',
fill_value= 0)
X_train[:,3:] = imputer.fit_transform(X_train[:,3:])
X_test[:,3:] = imputer.fit_transform(X_test[:,3:])

ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [2])]
, remainder='passthrough')
X_train = np.array(ct.fit_transform(X_train))
X_test = np.array(ct.fit_transform(X_test))

regressor = LinearRegression()
regressor.fit(X_train, y_train)

y_pred = regressor.predict(X_test)

error = []
for i in range(len(y_test)):
    a=abs(y_test[i]-y_pred[i])
    error.append(a)
acc=100-(sum(error)/len(y_test))
acc

# Watson Deployement
from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_credentials = {
 "apikey": "9G1kW3d3oxDMMCuPS4rTUdBe1BmPy-Ytm_4-991gwIfo",
  "iam_apikey_description": "Auto-generated for key ef07dea5-825e-4968-b73e
-e1cfd2072a0e",
  "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/e6537b6021944
416b6850964a7048cd1::serviceid:ServiceId-79e26329-b075-49cc-a2f0-27419a94da
bd",
  "instance_id": "60506c7f-fdf6-42d7-b4b1-d0774c4e2e41",
  "url": "https://eu-gb.ml.cloud.ibm.com"
}
client = WatsonMachineLearningAPIClient(wml_credentials)
print(client.version)

client.repository.list_models()
client.deployments.list()

meta_props={client.repository.ModelMetaNames.NAME: "Life Expectancy Pre
diction Model"}
```

```
published_model = client.repository.store_model(model=regressor, meta_p
rops={client.repository.ModelMetaNames.NAME: "Life Expectancy Predictio
n Model"})

client.repository.list_models()

# get UID of our just stored model
model_uid = client.repository.get_model_uid(published_model)
print("Model id: {}".format(model_uid))

created_deployment = client.deployments.create(model_uid, name="life_ex
p_pred_model")

# new list of deployments
client.deployments.list()

# get UID of our new deployment
deployment_uid = client.deployments.get_uid(created_deployment)
print("Deployment id: {}".format(deployment_uid))
print(created_deployment)

scoring_endpoint = client.deployments.get_scoring_url(created_deploymen
t)
print(scoring_endpoint)
```

- **Node-Red Application**

[{"id":"6028364f.e39ea8","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"f2f2649a.0d0d98","type":"debug","z":"6028364f.e39ea8","name":"","active":true,"console":"false","complete":"false","x":170,"y":100,"wires":[]},{"id":"83f6ea7d.069b98","type":"ui_form","z":"6028364f.e39ea8","name":"","label":"","group":"789d2ee0.8a001","order":0,"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},{"label":"Status (Developed/Developing)","value":"c","type":"text","required":true,"rows":null},{"label":"Adult Mortality","value":"d","type":"number","required":true,"rows":null},{"label":"Infant Deaths","value":"e","type":"number","required":true,"rows":null},{"label":"Alcohol","value":"f","type":"number","required":true,"rows":null},{"label":"Percentage Expenditure","value":"g","type":"number","required":true,"rows":null},{"label":"Hepatitis B","value":"h","type":"number","required":true,"rows":null},{"label":"Measles","value":"i","type":"number","required":true,"rows":null},{"label":"BMI","value":"j","type":"number","required":true,"rows":null},{"label":"Under-Five Deaths","value":"k","type":"number","required":true,"rows":null},{"label":"Polio","value":"l","type":"number","required":true,"rows":null},{"label":"Total expenditure","value":"m","type":"number","required":true,"rows":null},{"label":"Diphtheria","value":"n","type":"number","required":true,"rows":null},{"label":"HIV/AIDS","value":"o","type":"number","required":true,"rows":null},{"label":"GDP","value":"p","type":"number","required":true,"rows":null},{"label":"Population","value":"q","type":"number","required":true,"rows":null},{"label":"Thinness 1-19 Years","value":"r","type":"number","required":true,"rows":null},{"label":"Thinness 5-9 Years","value":"s","type":"number","required":true,"rows":null},{"label":"Income Composition of Resources","value":"t","type":"number","required":true,"rows":null},{"label":"Schooling","value":"u","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":"","u":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":150,"y":200,"wires":[["f2f2649a.0d0d98","f2b0ab1.50b0258"]]},{"id":"f2b0ab1.50b0258","type":"function","z":"6028364f.e39ea8","name":"","func":"global.set(\"a\",msg.payload.a);\nglobal.

set(\"b\",msg.payload.b);\nglobal.set(\"c\",msg.payload.c);\nglobal.set(\"d\",msg.payload.d);\nglobal.set(\"e\",msg.payload.e);\nglobal.set(\"f\",msg.payload.f);\nglobal.set(\"g\",msg.payload.g);\nglobal.set(\"h\",msg.payload.h);\nglobal.set(\"i\",msg.payload.i);\nglobal.set(\"j\",msg.payload.j);\nglobal.set(\"k\",msg.payload.k);\nglobal.set(\"l\",msg.payload.l);\nglobal.set(\"m\",msg.payload.m);\nglobal.set(\"n\",msg.payload.n);\nglobal.set(\"o\",msg.payload.o);\nglobal.set(\"p\",msg.payload.p);\nglobal.set(\"q\",msg.payload.q);\nglobal.set(\"r\",msg.payload.r);\nglobal.set(\"s\",msg.payload.s);\nglobal.set(\"t\",msg.payload.t);\nglobal.set(\"u\",msg.payload.u);\n\nvar apikey = \"9G1kW3d3oxDMMCuPS4rTUdBe1BmPy-Ytm_4-991gwIfo\";\nmsg.headers = {\"content-type\":\"application/x-www-form-urlencoded\"};\nmsg.payload = {\"grant_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey};\nreturn msg;","outputs":1,"noerr":0,"x":170,"y":320,"wires":[["8e24244b.353708"]]},{"id":"8e24244b.353708","type":"http request","z":"6028364f.e39ea8","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"basic","x":360,"y":460,"wires":[["268c355c.e2222a","979ab75f.b33e68"]]},{"id":"268c355c.e2222a","type":"function","z":"6028364f.e39ea8","name":"","func":"var token = msg.payload.access_token;\nvar instance_id = \"60506c7f-fdf6-42d7-b4b1-d0774c4e2e41\";\nmsg.headers = {\"Content-Type\":'application/json',\"Authorization\":\"Bearer\"+token,\"ML-Instance-ID\":instance_id};\n\nvar a = global.get(\"a\");\nif(a == \"Afghanistan\") {a = 0;}\nelse if(a == \"Albania\") {a = 1;}\nelse if(a == \"Algeria\") {a = 2;}\nelse if(a == \"Angola\") {a = 3;}\nelse if(a == \"Antigua and Barbuda\") {a = 4;}\nelse if(a == \"Argentina\") {a = 5;}\nelse if(a == \"Armenia\") {a = 6;}\nelse if(a == \"Australia\") {a = 7;}\nelse if(a == \"Austria\") {a = 8;}\nelse if(a == \"Azerbaijan\") {a = 9;}\nelse if(a == \"Bahamas\") {a = 10;}\nelse if(a == \"Bahrain\") {a = 11;}\nelse if(a == \"Bangladesh\") {a = 12;}\nelse if(a == \"Barbados\") {a = 13;}\nelse if(a == \"Belarus\") {a = 14;}\nelse if(a == \"Belgium\") {a = 15;}\nelse if(a == \"Belize\") {a = 16;}\nelse if(a == \"Benin\") {a = 17;}\nelse if(a == \"Bhutan\") {a = 18;}\nelse if(a == \"Plurinational State of Bolivia\") {a = 19;}\nelse if(a == \"Bosnia and Herzegovina\") {a = 20;}\nelse if(a == \"Botswana\") {a = 21;}\nelse if(a == \"Brazil\") {a = 22;}\nelse if(a == \"Brunei Darussalam\") {a = 23;}\nelse if(a == \"Bulgaria\") {a = 24;}\nelse if(a == \"Burkina Faso\") {a = 25;}\nelse if(a == \"Burundi\") {a = 26;}\nelse if(a == \"Côte d'Ivoire\") {a = 27;}\nelse if(a == \"Cabo Verde\") {a = 28;}\nelse if(a == \"Cambodia\") {a = 29;}\nelse if(a == \"Cameroon\") {a = 30;}\nelse if(a == \"Canada\") {a = 31;}\nelse if(a == \"Central African Republic\") {a = 32;}\nelse if(a == \"Chad\") {a = 33;}\nelse if(a == \"Chile\") {a = 34;}\nelse if(a == \"China\") {a = 35;}\nelse if(a == \"Colombia\") {a = 36;}\nelse if(a == \"Comoros\") {a = 37;}\nelse if(a == \"Congo\") {a = 38;}\nelse if(a == \"Costa Rica\") {a = 39;}\nelse if(a == \"Croatia\") {a = 40;}\nelse if(a == \"Cuba\") {a = 41;}\nelse if(a == \"Cyprus\") {a = 42;}\nelse if(a == \"Czechia\") {a = 43;}\nelse if(a == \"Democratic People's Republic of Korea\") {a = 44;}\nelse if(a == \"Democratic Republic of the Congo\") {a = 45;}\nelse if(a == \"Denmark\") {a = 46;}\nelse if(a == \"Djibouti\") {a = 47;}\nelse if(a == \"Dominican Republic\") {a = 48;}\nelse if(a == \"Ecuador\") {a = 49;}\nelse if(a == \"Egypt\") {a = 50;}\nelse if(a == \"El Salvador\") {a = 51;}\nelse if(a == \"Equatorial Guinea\") {a = 52;}\nelse if(a == \"Eritrea\") {a = 53;}\nelse if(a == \"Estonia\") {a = 54;}\nelse if(a == \"Ethiopia\") {a = 55;}\nelse if(a == \"Fiji\") {a = 56;}\nelse if(a == \"Finland\") {a = 57;}\nelse if(a == \"France\") {a = 58;}\nelse if(a == \"Gabon\") {a = 59;}\nelse if(a == \"Gambia\") {a = 60;}\nelse if(a == \"Georgia\") {a = 61;}\nelse if(a == \"Germany\") {a = 62;}\nelse if(a == \"Ghana\") {a = 63;}\nelse if(a == \"Greece\") {a = 64;}\nelse if(a == \"Grenada\") {a = 65;}\nelse if(a == \"Guatemala\") {a = 66;}\nelse if(a == \"Guinea\") {a = 67;}\nelse if(a == \"Bissau\") {a = 68;}\nelse if(a == \"Guyana\") {a = 69;}\nelse if(a == \"Haiti\") {a = 70;}\nelse if(a == \"Honduras\") {a = 71;}\nelse if(a == \"Hungary\") {a = 72;}\nelse if(a == \"Iceland\") {a = 73;}\nelse if(a == \"India\") {a = 74;}\nelse if(a == \"Indonesia\") {a = 75;}\nelse if(a == \"Islamic Republic of Iran\") {a = 76;}\nelse if(a == \"Iraq\") {a = 77;}\nelse if(a == \"Ireland\") {a = 78;}\nelse if(a == \"Israel\") {a = 79;}\nelse if(a == \"Italy\") {a = 80;}\nelse if(a == \"Jamaica\") {a = 81;}\nelse if(a == \"Japan\") {a = 82;}\nelse if(a == \"Jordan\") {a = 83;}\nelse if(a == \"Kazakhstan\") {a = 84;}\nelse if(a == \"Kenya\") {a = 85;}\nelse if(a == \"Kiribati\") {a = 86;}\nelse if(a == \"Kuwait\") {a = 87;}\nelse if(a == \"Kyrgyzstan\") {a = 88;}\nelse if(a == \"Lao People's Democratic Republic\") {a = 89;}\nelse if(a == \"Latvia\") {a = 90;}\nelse if(a == \"Lebanon\") {a = 91;}\nelse if(a == \"Lesotho\") {a = 92;}\nelse if(a == \"Liberia\") {a = 93;}\nelse if(a == \"Libya\") {a = 94;}\nelse if(a == \"Lithuania\") {a = 95;}\nelse if(a == \"Luxembourg\") {a = 96;}\nelse if(a == \"Madagascar\") {a = 97;}\nelse if(a == \"Malawi\") {a = 98;}\nelse if(a == \"Malaysia\") {a = 99;}\nelse if(a == \"Maldives\") {a = 100;}\nelse if(a == \"Mali\") {a = 101;}\nelse if(a == \"Malta\") {a = 102;}\nelse if(a == \"Mauritania\") {a = 103;}\nelse if(a ==

\"Mauritius\") {a = 104;}\nelse if(a == \"Mexico\") {a = 105;}\nelse if(a == \"Federated States of Micronesia\") {a = 106;}\nelse if(a == \"Mongolia\") {a = 107;}\nelse if(a == \"Montenegro\") {a = 108;}\nelse if(a == \"Morocco\") {a = 109;}\nelse if(a == \"Mozambique\") {a = 110;}\nelse if(a == \"Myanmar\") {a = 111;}\nelse if(a == \"Namibia\") {a = 112;}\nelse if(a == \"Nepal\") {a = 113;}\nelse if(a == \"Netherlands\") {a = 114;}\nelse if(a == \"New Zealand\") {a = 115;}\nelse if(a == \"Nicaragua\") {a = 116;}\nelse if(a == \"Niger\") {a = 117;}\nelse if(a == \"Nigeria\") {a = 118;}\nelse if(a == \"Norway\") {a = 119;}\nelse if(a == \"Oman\") {a = 120;}\nelse if(a == \"Pakistan\") {a = 121;}\nelse if(a == \"Panama\") {a = 122;}\nelse if(a == \"Papua New Guinea\") {a = 123;}\nelse if(a == \"Paraguay\") {a = 124;}\nelse if(a == \"Peru\") {a = 125;}\nelse if(a == \"Philippines\") {a = 126;}\nelse if(a == \"Poland\") {a = 127;}\nelse if(a == \"Portugal\") {a = 128;}\nelse if(a == \"Qatar\") {a = 129;}\nelse if(a == \"Republic of Korea\") {a = 130;}\nelse if(a == \"Republic of Moldova\") {a = 131;}\nelse if(a == \"Romania\") {a = 132;}\nelse if(a == \"Russian Federation\") {a = 133;}\nelse if(a == \"Rwanda\") {a = 134;}\nelse if(a == \"Saint Lucia\") {a = 135;}\nelse if(a == \"Saint Vincent and the Grenadines\") {a = 136;}\nelse if(a == \"Samoa\") {a = 137;}\nelse if(a == \"Sao Tome and Principe\") {a = 138;}\nelse if(a == \"Saudi Arabia\") {a = 139;}\nelse if(a == \"Senegal\") {a = 140;}\nelse if(a == \"Serbia\") {a = 141;}\nelse if(a == \"Seychelles\") {a = 142;}\nelse if(a == \"Sierra Leone\") {a = 143;}\nelse if(a == \"Singapore\") {a = 144;}\nelse if(a == \"Slovakia\") {a = 145;}\nelse if(a == \"Slovenia\") {a = 146;}\nelse if(a == \"Solomon Islands\") {a = 147;}\nelse if(a == \"Somalia\") {a = 148;}\nelse if(a == \"South Africa\") {a = 149;}\nelse if(a == \"South Sudan\") {a = 150;}\nelse if(a == \"Spain\") {a = 151;}\nelse if(a == \"Sri Lanka\") {a = 152;}\nelse if(a == \"Sudan\") {a = 153;}\nelse if(a == \"Suriname\") {a = 154;}\nelse if(a == \"Swaziland\") {a = 155;}\nelse if(a == \"Sweden\") {a = 156;}\nelse if(a == \"Switzerland\") {a = 157;}\nelse if(a == \"Syrian Arab Republic\") {a = 158;}\nelse if(a == \"Tajikistan\") {a = 159;}\nelse if(a == \"Thailand\") {a = 160;}\nelse if(a == \"The former Yugoslav republic of Macedonia\") {a = 161;}\nelse if(a == \"Timor-Leste\") {a = 162;}\nelse if(a == \"Togo\") {a = 163;}\nelse if(a == \"Tonga\") {a = 164;}\nelse if(a == \"Trinidad and Tobago\") {a = 165;}\nelse if(a == \"Tunisia\") {a = 166;}\nelse if(a == \"Turkey\") {a = 167;}\nelse if(a == \"Turkmenistan\") {a = 168;}\nelse if(a == \"Uganda\") {a = 169;}\nelse if(a == \"Ukraine\") {a = 170;}\nelse if(a == \"United Arab Emirates\") {a = 171;}\nelse if(a == \"United Kingdom of Great Britain and Northern Ireland\") {a = 172;}\nelse if(a == \"United Republic of Tanzania\") {a = 173;}\nelse if(a == \"United States of America\") {a = 174;}\nelse if(a == \"Uruguay\") {a = 175;}\nelse if(a == \"Uzbekistan\") {a = 176;}\nelse if(a == \"Vanuatu\") {a = 177;}\nelse if(a == \"Bolivarian Republic of Venezuela\") {a = 178;}\nelse if(a == \"Viet Nam\") {a = 179;}\nelse if(a == \"Yemen\") {a = 180;}\nelse if(a == \"Zambia\") {a = 181;}\nelse if(a == \"Zimbabwe\") {a = 182;}\nvar b = global.get(\"b\");\nvar c = global.get(\"c\");\nif (c == 'Developed')\n {c = 0;}\nelse if (c == 'Developing')\n {c = 1;}\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\nmsg.payload =\n{\"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\"],\"values\":[[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u]]};\nreturn msg;","outputs":1,"noerr":0,"x":500,"y":320,"wires":[["83f56166.91cc"]]},{"id":"83f56166.91cc","type":"http request","z":"6028364f.e39ea8","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/60506c7f-fdf6-42d7-b4b1-d0774c4e2e41/deployments/2ae63419-4189-4531-8593-44f451b38e71/online","tls":"","persist":false,"proxy":"","authType":"","x":500,"y":180,"wires":[["9e0052bc.be545","74e478cc.44ed88"]]},{"id":"9e0052bc.be545","type":"debug","z":"6028364f.e39ea8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":530,"y":60,"wires":[]},{"id":"74e478cc.44ed88","type":"function","z":"6028364f.e39ea8","name":"","func":"msg.payload = msg.payload.values[0][0];\nreturn msg;","outputs":1,"noerr":0,"x":810,"y":140,"wires":[["d6470214.b43eb"]]},{"id":"d6470214.b43eb","type":"ui_text","z":"6028364f.e39ea8","group":"789d2ee0.8a001","order":1,"width":0,"height":0,"name":"","label":"Prediction","format":"{{msg.payload}}","layout":"row-

spread","x":850,"y":300,"wires":[]},{"id":"979ab75f.b33e68","type":"debug","z":"6028364f.e39ea8","name":""
,"active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":580,"y":540,"wires":[]},{"
id":"789d2ee0.8a001","type":"ui_group","z":"","name":"Life Expectancy
Prediction","tab":"5899c6a8.e1cee8","order":1,"disp":true,"width":"6","collapse":false},{"id":"5899c6a8.e1cee
8","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]