PROJECT REPORT

NAME : Ashutosh Pandab

EMAIL : ashutoshpandab25@gmail.com

TITLE : Predicting Life Expectancy of a Country

CATEGORY : Machine Learning

Webpage Link :

https://node-red-sayot.mybluemix.net/ui/#!/0?socketid=7oELOXScuvdTQB2gAAAW

1. INTRODUCTION

1.1. Overview

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. 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, this is a typical Regression Machine Learning project that leverages historical data to predict insights into the future.

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: Life Expectancy Prediction Webpage

Project Team: Ashutosh Pandab

Project Duration: 23.5 Days

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. Proposed 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:

a) Create IBM cloud services

b) Configure Watson Studio

c) Create Node-Red Flow to connect all services together

d) 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

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.

Then add dataset and empty jupyter notebook into Assets. 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 Pre-processing**
    - Removing unusual species in column names using rename function.

o Replacing nan values if any with their mean values.

▪ **Exploratory Data Analysis**

o Plotting a heatmap to check if dimensional reduction can be performed

o Plotting a pairplot for analysing pairwise relatonship among features.

▪ **Train and Test**

o The dataset was splitted 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

o 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

o Then we standardize the numeric and categoric columns using pipelining.

o At first independent pipelines for both the parts were designed then they were joined using columntransform

o After that a regressor pipeline was designed using the regression technique.

o So I have used ExtraTreesRegressor technique of sklearn.essemble as my regression algorithm because it best fits my dataset.

o Then train and test split was peformed and 80% of dataset were trained data and 20% were test data. o Then dataset was fitted and predicted.

o Then error and accuracy was estimated and the mean squared error is 2.98 whereas the R2_score or accuracy is 96.80%.

▪ **Model Building and Deployment**

o At first the machine learning service credentials was stored in a variable and passed into WatsonMachineLearningAPIClient.

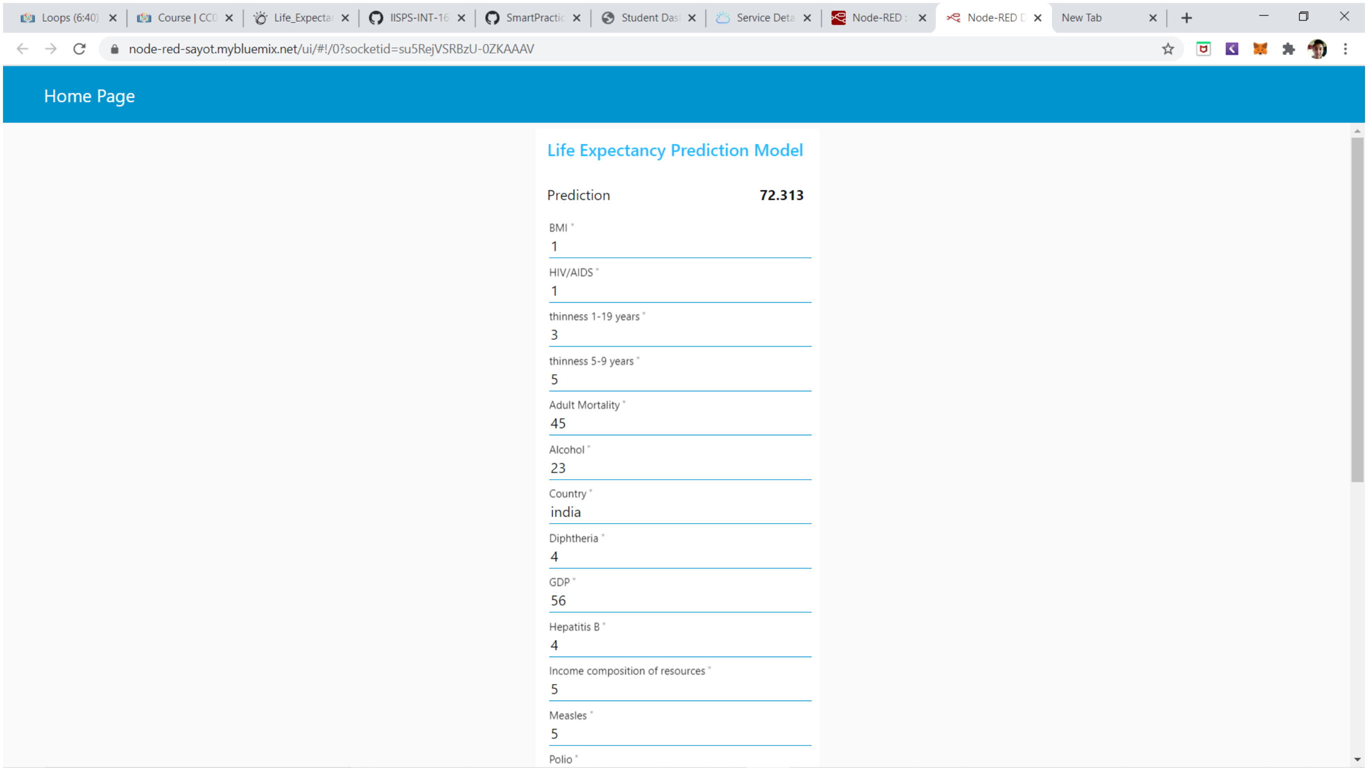
o Then the model was build and stored in model_artifact.

o Then the model was deployed and scoring_endpoint url was generated.

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

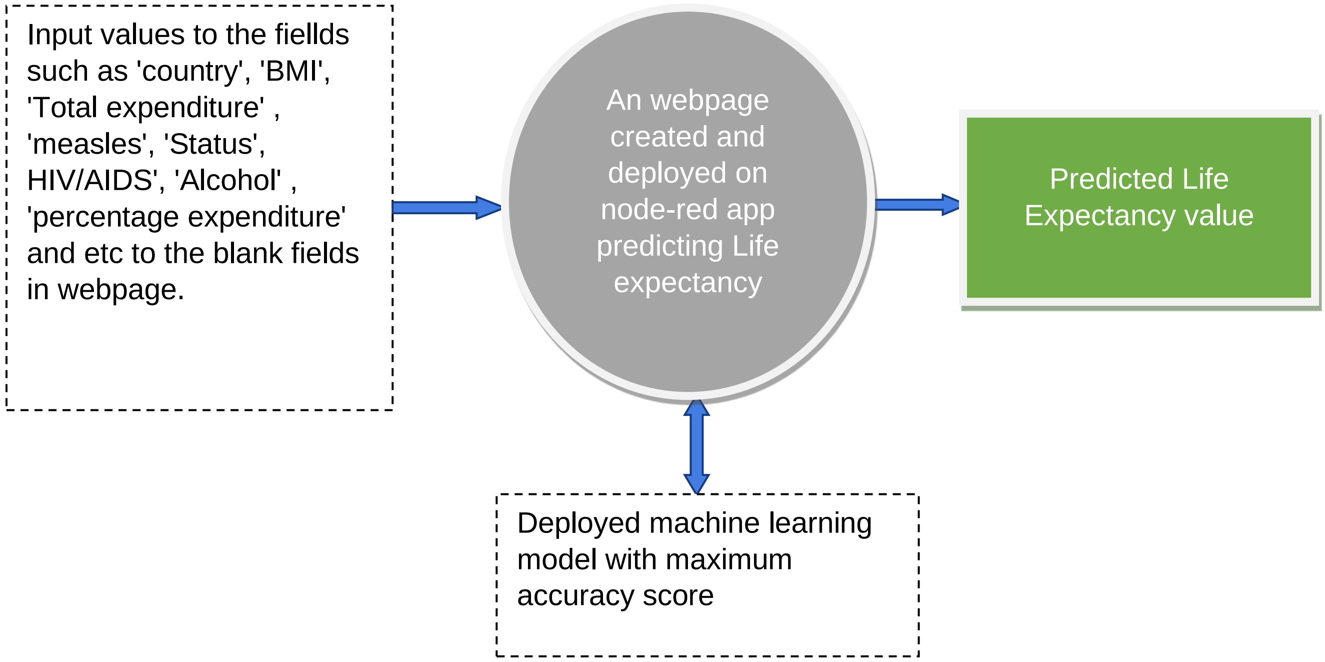
- Go to Node-Red Editor from resource list.
- Install node-red Dashboard from manage pallete.
- Now create the flow with the help of following node.
  - o Inject o Debug
  - o Function
  - o Ui_Form
  - o Ui_Text
- Deploy and run Node Red app.

  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.



3. THEORETICAL ANALYSIS

3.1. BLOCK DIAGRAM

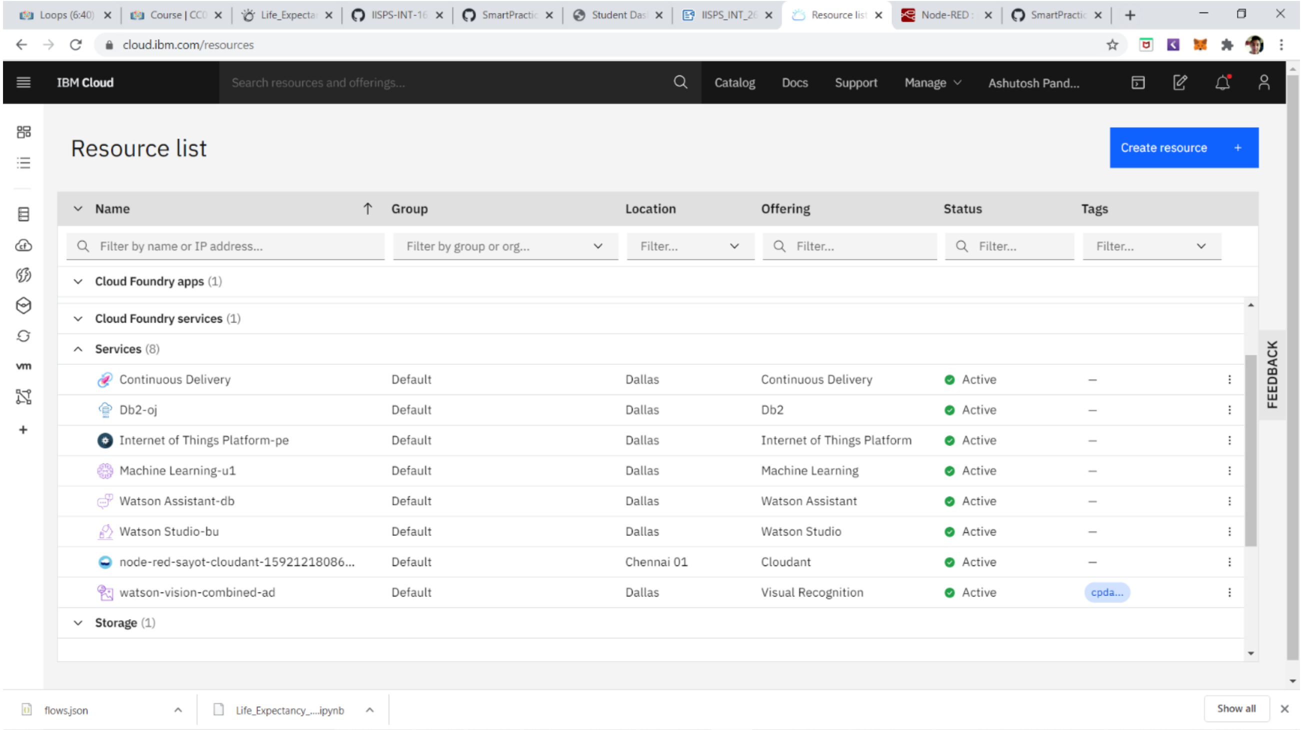


3.2. HARDWARE / SOFTWARE DESIGNING

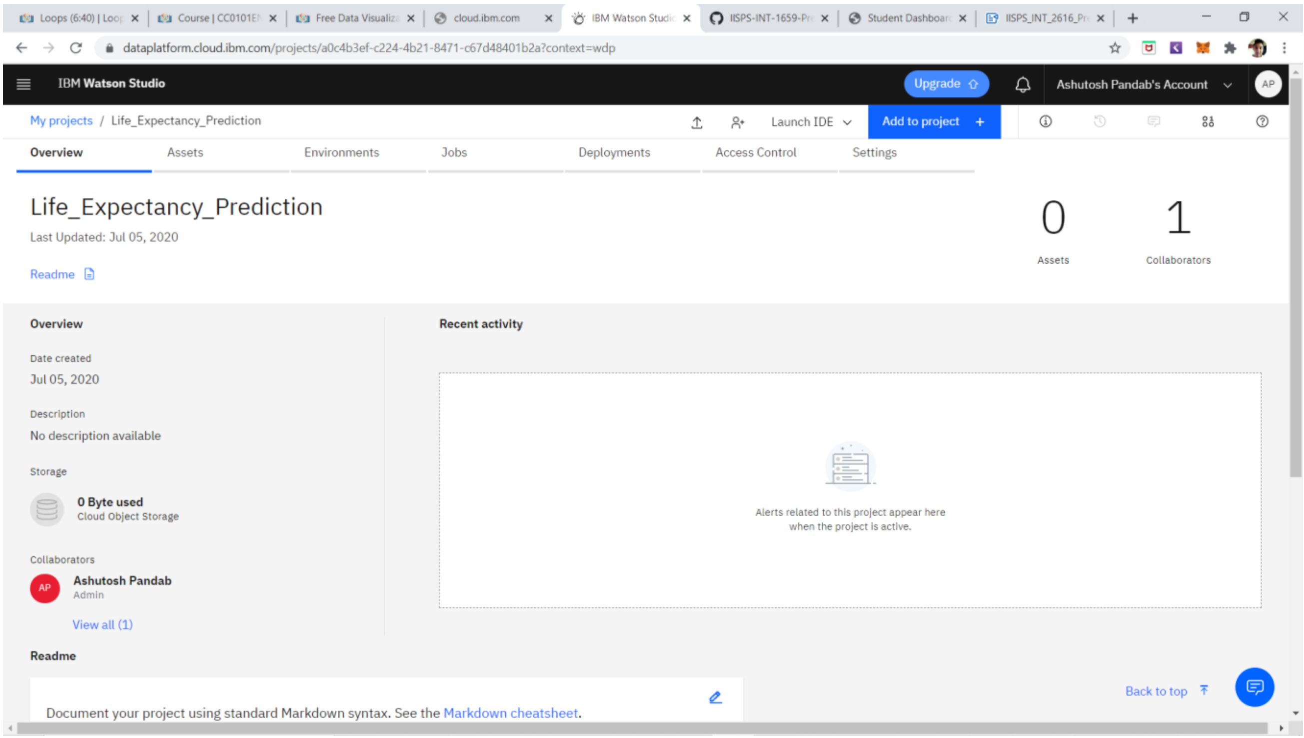
- **Project Requirements:** Python, IBM Cloud, IBM Watson
- **Functional Requirements:** IBM cloud
- **Technical Requirements:** ML, WATSON Studio, Python, Node-Red
- **Software Requirements:** Watson Studio, Node-Red

4. EXPERIMENTAL INVESTIGATIONS

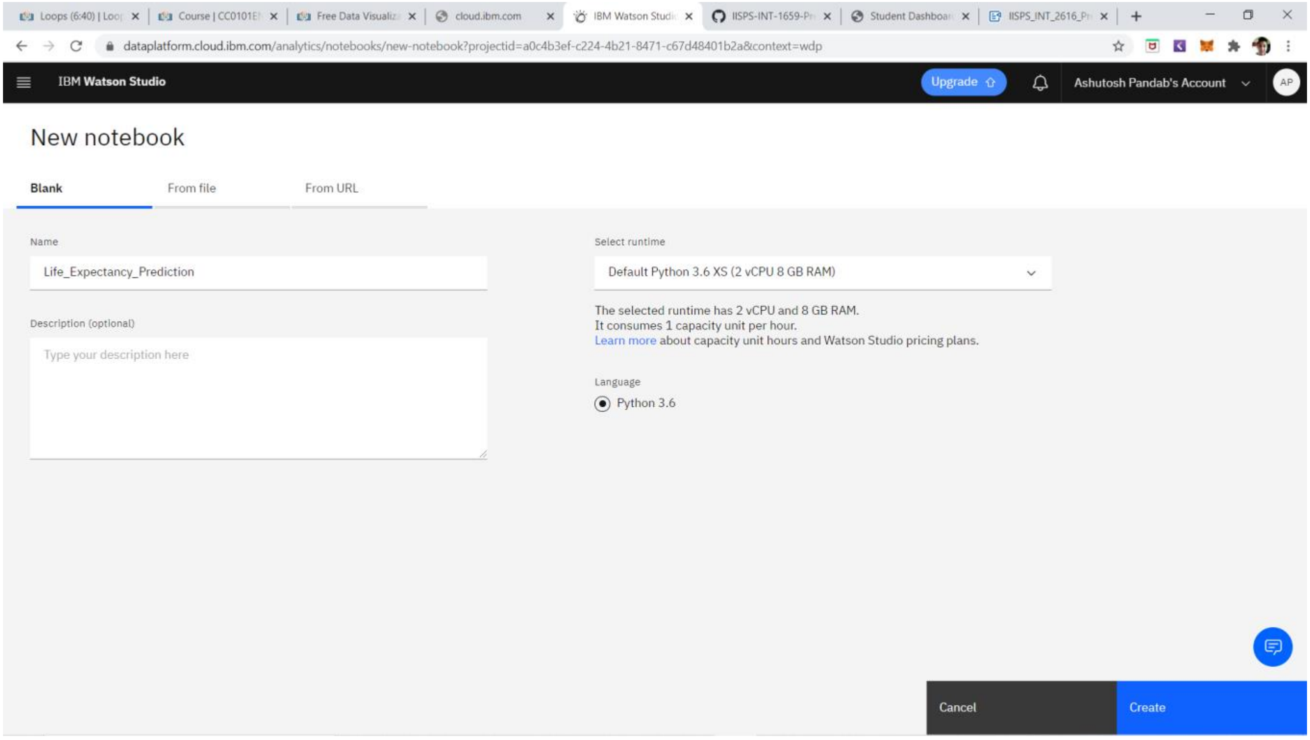
A) IBM Cloud Resource List



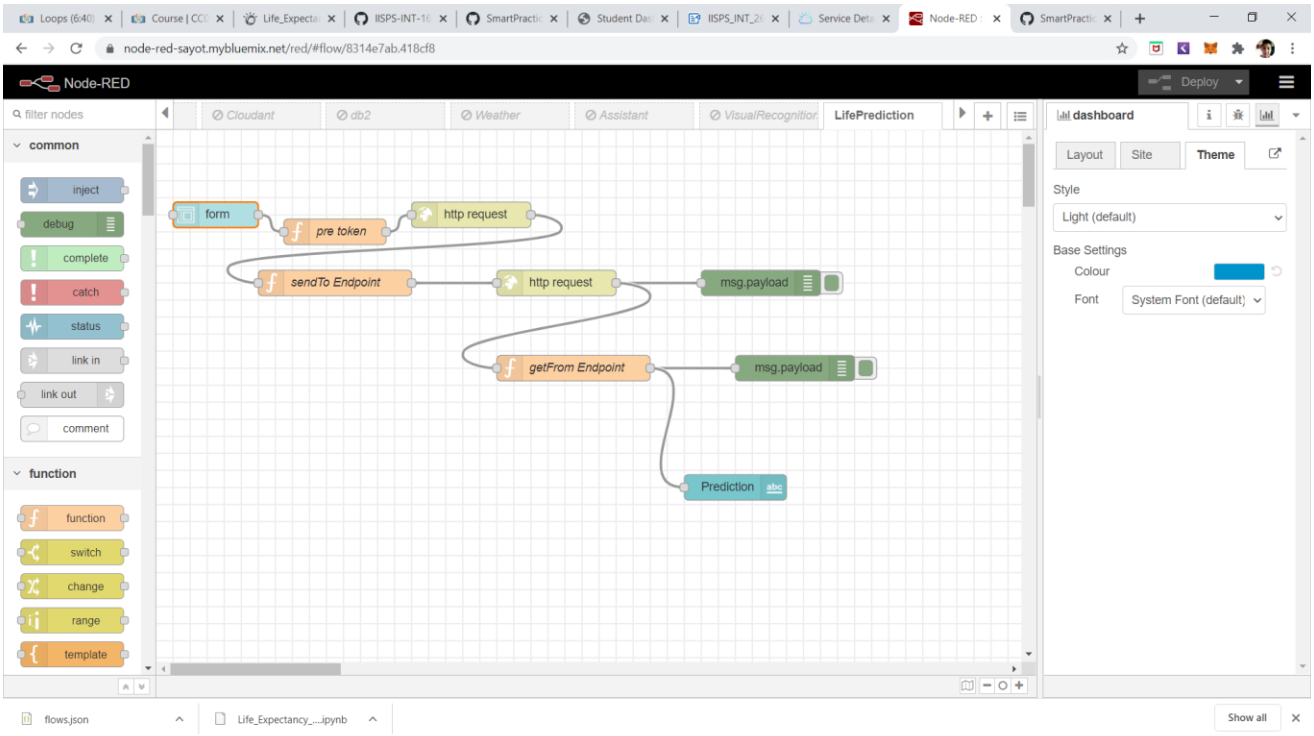
B) IBM Watson Studio



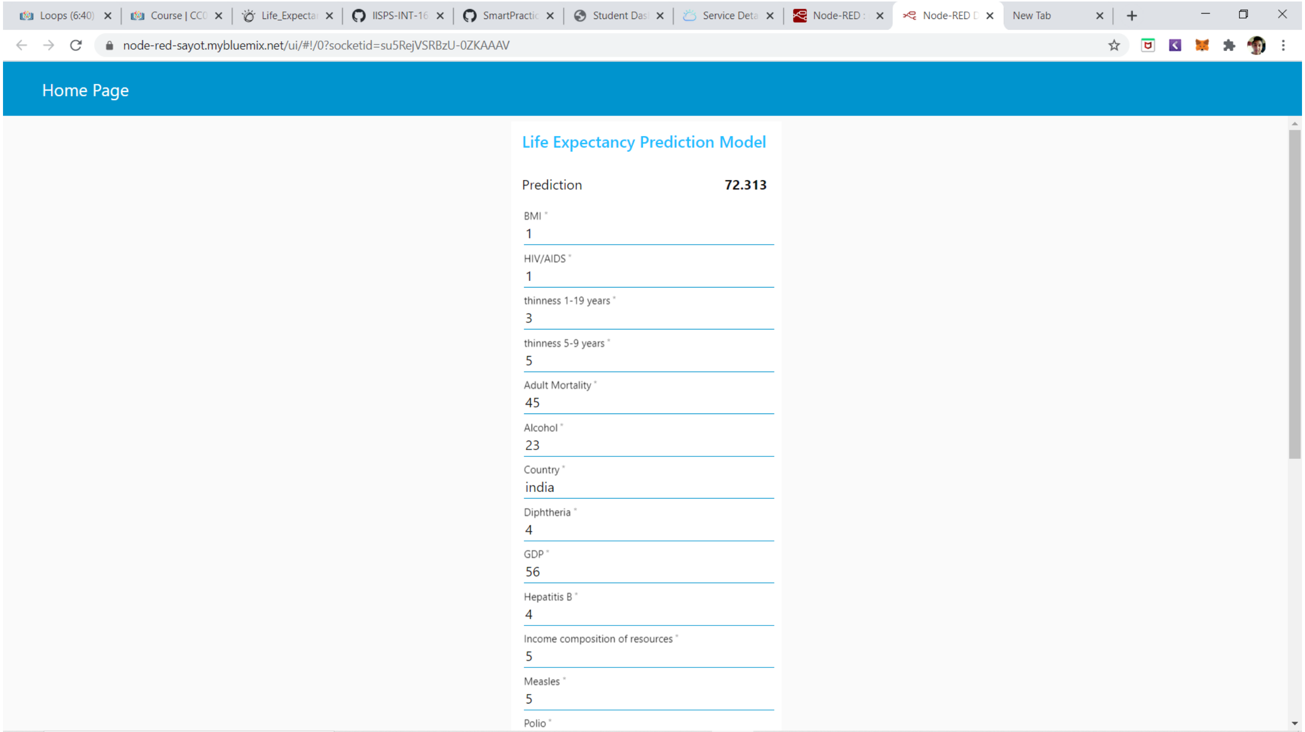
C) IBM Cloud Project Details



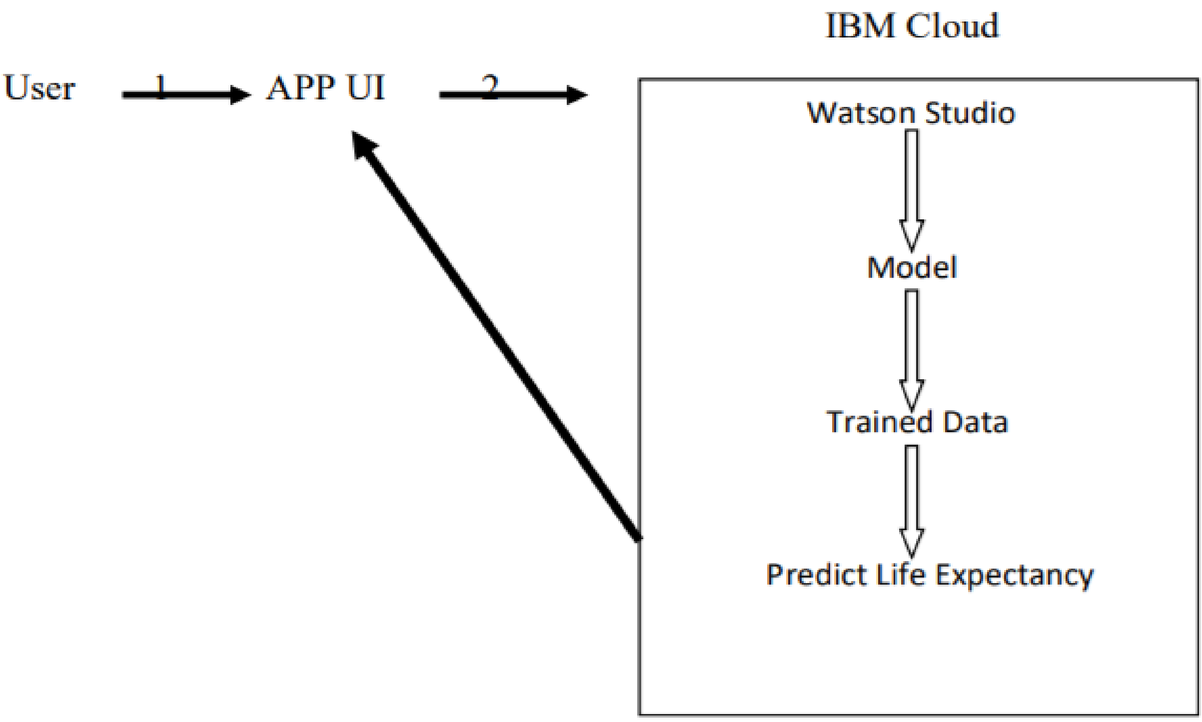
D)Node-Red Flow



E) Life Expectancy Prediction UI



5.FLOWCHART



a) The user input all the required values in the app

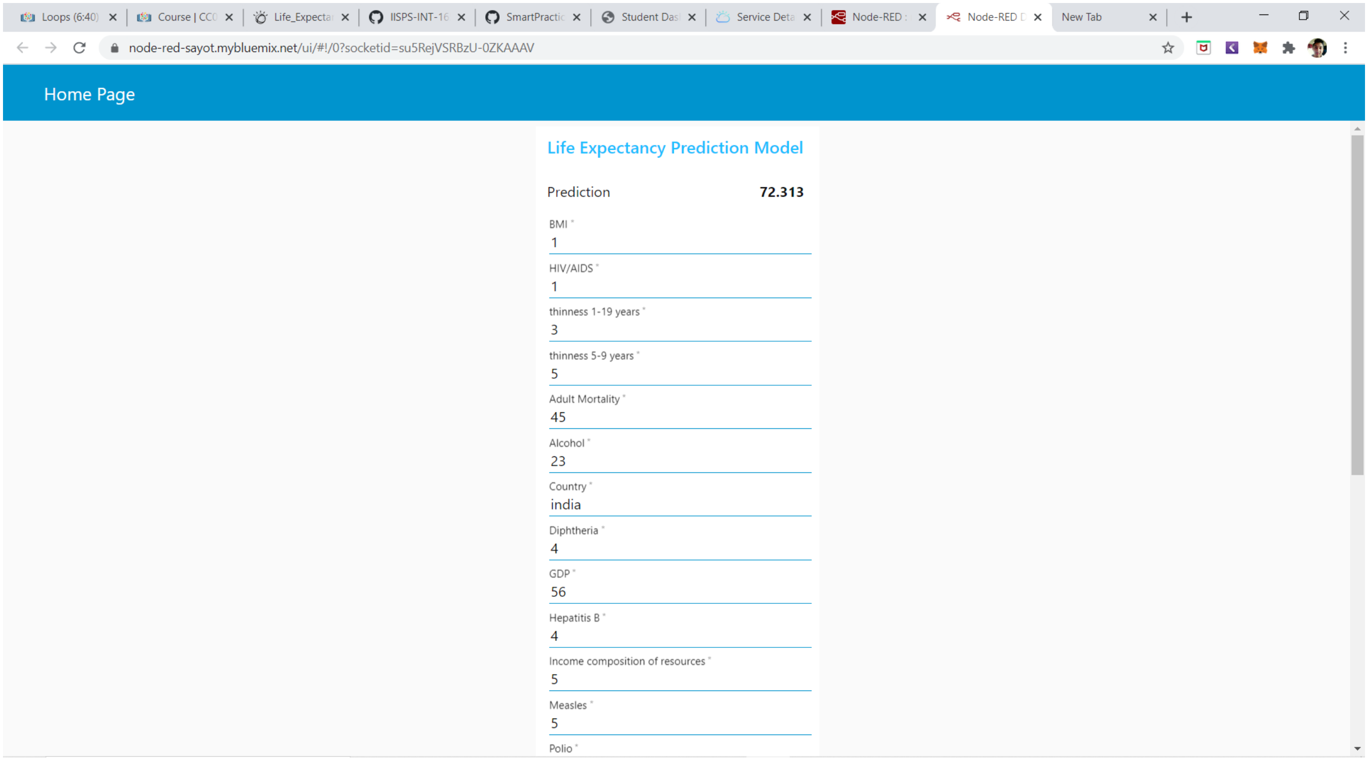
b) The data then entered into watson and the scoring_endpoint url matches with the deployed model.

c) Then it enters into trained data and predict the life expectancy value

d) The value predicted is prompted in the app screen.

6. RESULT

This is the Life Expectancy UI.



7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

a) Health Inequalities: Life expectancy has been used nationally to monitor health inequalities of a country.

b) 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 doesnot required any kind of payment neither for designing nor for using.

c) 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.

DISADVANTAGES:

a) Wrong Prediction: As it depends completely on user, so if user provides some wrong values then it will predict wrong value.

b) Average Prediction: The model predicts average or approximate value with 97.07% accuracy but not accurate value.

8. APPLICATION

a) It can be used to monitor health inequalities of a country.

b) It can be used to develop statistics for country development process.

c) It can be used to analyse the factors for high life expectancy.

d) It is user friendly and can be used by anyone.

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:

a) 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 I have decided to do some kind of feature reduction or replacement of some features as individuals or groups to make it more user friendly.

b) 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.

c) Integrating with services such as speech recognition

11. BIBLIOGRAPHY

- 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=Jtej3Y6uUng
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-webservice
- https://machinelearningmastery.com/columntransformer-for-numerical-and-categorical-data/

APPENDIX: Source Code

1.NOTEBOOK:

```
!pip install watson_machine_learning_client

#import basic libraries for preprocessing and EDA

import pandas as pd

import numpy as np

import os

import matplotlib.pyplot as plt

import seaborn as sns

pd.options.display.float_format='{:.5f}'.format

import warnings

import math

#import libraries for pipelining

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import OneHotEncoder

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler
```

```
from sklearn.compose import ColumnTransformer
#import libraries for train and test
from sklearn.model_selection import train_test_split
#import ExtraTreesRegressor for model fit and prediction
from sklearn.ensemble import ExtraTreesRegressor
#import libraries for accuracy and error calculation
from sklearn.metrics import mean_squared_error, r2_score
#import libraries for model building and deployment
from watson_machine_learning_client import WatsonMachineLearningAPIClient

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_6335f21ad6804054a184318f65209b87 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='QpZmvn67LG8KdUnzreSwKmWr6xAEgHGHvkrAk5lY33Sh',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_6335f21ad6804054a184318f65209b87.get_object(Bucket='lifeexpectancyprediction-donotdelete-pr-ssoj3fivcdgyt3',Key='Life Expectancy Data.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )
```

```
df = pd.read_csv(body)

df.head()

df.columns

df=df.rename(columns={'Life expectancy ':'Life expectancy','Measles ':'Measles',' BMI ':'BMI','Diphtheria ':'Diphtheria',' HIV/AIDS':'HIV/AIDS',' thinness  1-19 years':'thinness  1-19 years',' thinness 5-9 years':'thinness 5-9 years'})

df.isnull().sum()

#FILLING NULL VALUES TO AVOID TRAIN AND TEST ERROR

df=df.fillna(df.mean())

df.isnull().sum()

#PLOTTING A HEATMAP

df_kor=df.corr()

plt.figure(figsize=(10,10))

sns.heatmap(df_kor,vmin=-1,vmax=1,annot=True,linewidth=0.1)

#PLOTTING A PAIRPLOT

sns.pairplot(df)

#SPLITTING THE DATASET

Y=df['Life expectancy']

X=df[df.columns.difference(['Life expectancy'])]

#SEE NUMERICAL COLUMNS

df.select_dtypes(include=['int64', 'float64']).columns

#SEE CATEGORICAL COLUMNS

df.select_dtypes(include=['object', 'bool']).columns

#IDENTIFY THE CATEGORICAL VALUES FOR COLUMNTRANSFORM

categorical_features = ['Country', 'Status']

categorical_feature_mask = X.dtypes==object

categorical_features = X.columns[categorical_feature_mask].tolist()

#DEFINE CATEGORICAL PIPELINE

categorical_transformer = Pipeline(steps=[
```

```
  ('onehot', OneHotEncoder(handle_unknown='ignore')),
])
#IDENTIFY THE NUMERIC VALUES FOR COLUMNTRANSFORM
numeric_features = ['Year','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']
numeric_feature_mask = X.dtypes!=object
numeric_features = X.columns[numeric_feature_mask].tolist()
#DEFINE NUMERIC PIPELINE
numeric_transformer = Pipeline(steps=[
  ('imputer', SimpleImputer(strategy='median')),
  ('scaler', StandardScaler()),
])
#PIPELINNING USING COLUMNTRANSFORM
preprocessor = ColumnTransformer(
  transformers=[
    ('num', numeric_transformer, numeric_features),
    ('cat', categorical_transformer, categorical_features)
  ]
)
#DEFINE A REGRESSOR MODEL USING PIPELINE FUNCTION
ExtraTreeRegressor = Pipeline([
   ('preprocessor', preprocessor),
   ('ExtraTreeRegressor', ExtraTreesRegressor(n_estimators=100, random_state=0))
])
#TRAIN-TEST SPLIT
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)
#FIT THE TRAINING MODEL
```

```
reg = ExtraTreeRegressor.fit(X_train, Y_train)

#PREDICT THE TEST DATA VALUE

test_pred=reg.predict(X_test)

print(test_pred)

#ESTIMATING ERROR

print('Mean squared error: ',mean_squared_error(Y_test, test_pred))

print('R2 score: ',r2_score(Y_test, test_pred)*100)

wml_credentials = {

  "apikey": "x9X4scGRQoIGqUYZ6cD989CRIbUilrWWwt-6FZ4vGsmm",

  "iam_apikey_description": "Auto-generated for key d07a3e05-d212-440a-af58-530655e43537",

  "iam_apikey_name": "Service credentials-1",

  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",

  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-
identity::a/450bb85f820c42d4acfd9248d7357b13::serviceid:ServiceId-eb85bf8d-8ddb-4b5b-b630-
2af41fe0366d",

  "instance_id": "732600fd-4b72-439e-9a73-6cbadf5988cb",

  "url": "https://us-south.ml.cloud.ibm.com"

}

client = WatsonMachineLearningAPIClient(wml_credentials)

print(client.service_instance.get_url())

metadata = {

    client.repository.ModelMetaNames.AUTHOR_NAME: "Ashutosh",

    client.repository.ModelMetaNames.AUTHOR_EMAIL: "ashutoshpandab25@gmail.com",

    client.repository.ModelMetaNames.NAME: "LifeExpectancyPrediction"

}

#STORING THE MACHINE LEARNING MODEL

stored_data = client.repository.store_model(ExtraTreeRegressor, meta_props = metadata)

#GET MODEL UID

guid = client.repository.get_model_uid(stored_data)

#DEPLOYING THE MODEL
```

deploy = client.deployments.create(guid, name="LifeExpectancyPrediction")

#GET SCORING END-POINT URL

scoring_endpoint = client.deployments.get_scoring_url(deploy)

print(scoring_endpoint)

2. NODE-RED FLOW:

[{"id":"8314e7ab.418cf8","type":"tab","label":"LifePrediction","disabled":false,"info":""},{"id":"6c0799d5.793e58","type":"ui_form","z":"8314e7ab.418cf8","name":"","label":"","group":"ba8a9017.3ba3","order":0,"width":0,"height":0,"options":[{"label":"BMI\t","value":"a","type":"number","required":true,"rows":null},{"label":"HIV/AIDS","value":"b","type":"number","required":true,"rows":null},{"label":"thinness 1-19 years","value":"c","type":"number","required":true,"rows":null},{"label":"thinness 5-9 years","value":"d","type":"number","required":true,"rows":null},{"label":"Adult Mortality","value":"e","type":"number","required":true,"rows":null},{"label":"Alcohol","value":"f","type":"number","required":true,"rows":null},{"label":"Country","value":"g","type":"text","required":true,"rows":null},{"label":"Diphtheria ","value":"h","type":"number","required":true,"rows":null},{"label":"GDP","value":"i","type":"number","required":true,"rows":null},{"label":"Hepatitis B","value":"j","type":"number","required":true,"rows":null},{"label":"Income composition of resources","value":"k","type":"number","required":true,"rows":null},{"label":"Measles ","value":"l","type":"number","required":true,"rows":null},{"label":"Polio","value":"m","type":"number","required":true,"rows":null},{"label":"Population","value":"n","type":"number","required":true,"rows":null},{"label":"Schooling","value":"o","type":"number","required":true,"rows":null},{"label":"Status","value":"p","type":"text","required":true,"rows":null},{"label":"Total expenditure","value":"q","type":"number","required":true,"rows":null},{"label":"Year","value":"r","type":"number","required":true,"rows":null},{"label":"infant deaths","value":"s","type":"number","required":true,"rows":null},{"label":"percentage expenditure","value":"t","type":"number","required":true,"rows":null},{"label":"under-five deaths ","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":70,"y":100,"wires":[["b816fb83.351118"]]},{"id":"e556e56b.7822b8","type":"http request","z":"8314e7ab.418cf8","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://us-south.ml.cloud.ibm.com/v3/wml_instances/732600fd-4b72-439e-9a73-6cbadf5988cb/deployments/38c95db6-3ac9-4a8e-8989-b15120e083ca/online","tls":"","persist":false,"proxy":"","authType":"","x":470,"y":180,"wires":[["6dd7f63b.4218f8","61fc472c.4b5678"]]},{"id":"117c7131.4915bf","type":"debug","z":"8314e7ab.418cf8","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":750,"y":280,"wires":[]},{"id":"61fc472c.4b5678","type":"function","z":"8314e7ab.418cf8","name":"getFrom Endpoint","func":"msg.payload=msg.payload.values[0][0];\n//msg.payload=msg.payload.predictions[0].values[0][0];\nreturn msg;","outputs":1,"noerr":0,"x":490,"y":280,"wires":[["117c7131.4915bf","9c09338a.2bc7c"]]},{"id":"6d

d7f63b.4218f8","type":"debug","z":"8314e7ab.418cf8","name":"","active":true,"tosidebar":true,"consol
e":false,"tostatus":false,"complete":"payload","targetType":"msg","x":710,"y":180,"wires":[]},{"id":"999
c5f2b.d634","type":"function","z":"8314e7ab.418cf8","name":"sendTo Endpoint","func":"//get token
and make headers\nvar token=msg.payload.access_token;\nvar instance_id=\"732600fd-4b72-439e-
9a73-6cbadf5988cb\"\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\");\nvar u = global.get(\"u\");\n\n//send the user values to service
endpoint\nmsg.payload = \n{\"fields\":[\"BMI\", \"HIV/AIDS\", \"thinness 1-19 years\", \"thinness 5-9
years\",\n   \"Adult Mortality\", \"Alcohol\", \"Country\", \"Diphtheria \", \"GDP\",\n   \"Hepatitis
B\", \"Income composition of resources\", \"Measles \", \"Polio\",\n   \"Population\", \"Schooling\",
\"Status\", \"Total expenditure\", \"Year\",\n   \"infant deaths\", \"percentage expenditure\",
\"under-five deaths \"],\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":[["e556e56b.7822b8"]]},{"id":"e0fc0c29.8ec26","
type":"http
request","z":"8314e7ab.418cf8","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://i
am.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"basic","x":370,"y":100,
"wires":[["999c5f2b.d634"]]},{"id":"b816fb83.351118","type":"function","z":"8314e7ab.418cf8","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.payl
oad.c);\nglobal.set(\"d\",msg.payload.d);\nglobal.set(\"e\",msg.payload.e);\nglobal.set(\"f\",msg.paylo
ad.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);\n
global.set(\"s\",msg.payload.s);\nglobal.set(\"t\",msg.payload.t);\nglobal.set(\"u\",msg.payload.u);\n\n
//following are required to receive a token\nvar apikey=\"x9X4scGRQoIGqUYZ6cD989CRIbUilrWWwt-
6FZ4vGsmm\";\nmsg.headers={\"content-type\":\"application/x-www-form-
urlencoded\"};\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-
type:apikey\",\"apikey\":apikey};\n\nreturn
msg;","outputs":1,"noerr":0,"x":210,"y":120,"wires":[["e0fc0c29.8ec26"]]},{"id":"9c09338a.2bc7c","type
":"ui_text","z":"8314e7ab.418cf8","group":"ba8a9017.3ba3","order":2,"width":0,"height":0,"name":"","
label":"Prediction","format":"{{msg.payload}}","layout":"row-
spread","x":680,"y":420,"wires":[]},{"id":"ba8a9017.3ba3","type":"ui_group","z":"","name":"Life
Expectancy Prediction
Model","tab":"3cffe80b.8606b8","order":1,"disp":true,"width":"6","collapse":false},{"id":"3cffe80b.860
6b8","type":"ui_tab","z":"","name":"Home Page","icon":"dashboard","disabled":false,"hidden":false}]