Project Report

Predicting Life Expectancy

Submitted by :

Name – Abhinav Jain

Email – abhinavj65@gmail.com

Designation – Machine Learning Intern

Project Title - Predicting Life Expectancy

Webpage link –

https://node-red-gxlqz.eu-gb.mybluemix.net/ui/#!/5?socketid=8s3L3BFDdeUwP69FAAAj

Index

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.

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 given country associated with various criterias.

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.

1.2. Purpose

The purpose of the project is to make the user aware about their life expectancy according to the conditions and environment they are living into. It is expected that people will get motivated and adopt better and healthy environment and habits by understanding the factors affecting their life.

2. Project Requirements

The project requirements have been categorised as follows into three parts :-

➢ Functional Requirements :

- Dataset (containing the history of life expectancy in different countries influenced by various factors)]
- Tools for predicting life expectancy

➢ Technical Requirements:

One must be aware with the following skills:

- Python
- Machine Learning
- IBM Cloud
- IBM Watson

➢ Software Requirements:

- IDLE Python
- Jupyter Notebook
- IBM Cloud Services
- IBM Watson Tools

3. Project Deliverables

- Documents - Project Scope and Project Report
- Project – Life expectancy prediction model
- Node-red file
- Video Demonstration of project via youtube

4. LITERATURE SURVEY

4.1. 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.

4.2. 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

4.2.1. Create IBM cloud Services

- Watson Studio
- Machine Leaarning resource
- Node-Red

4.2.2. 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:

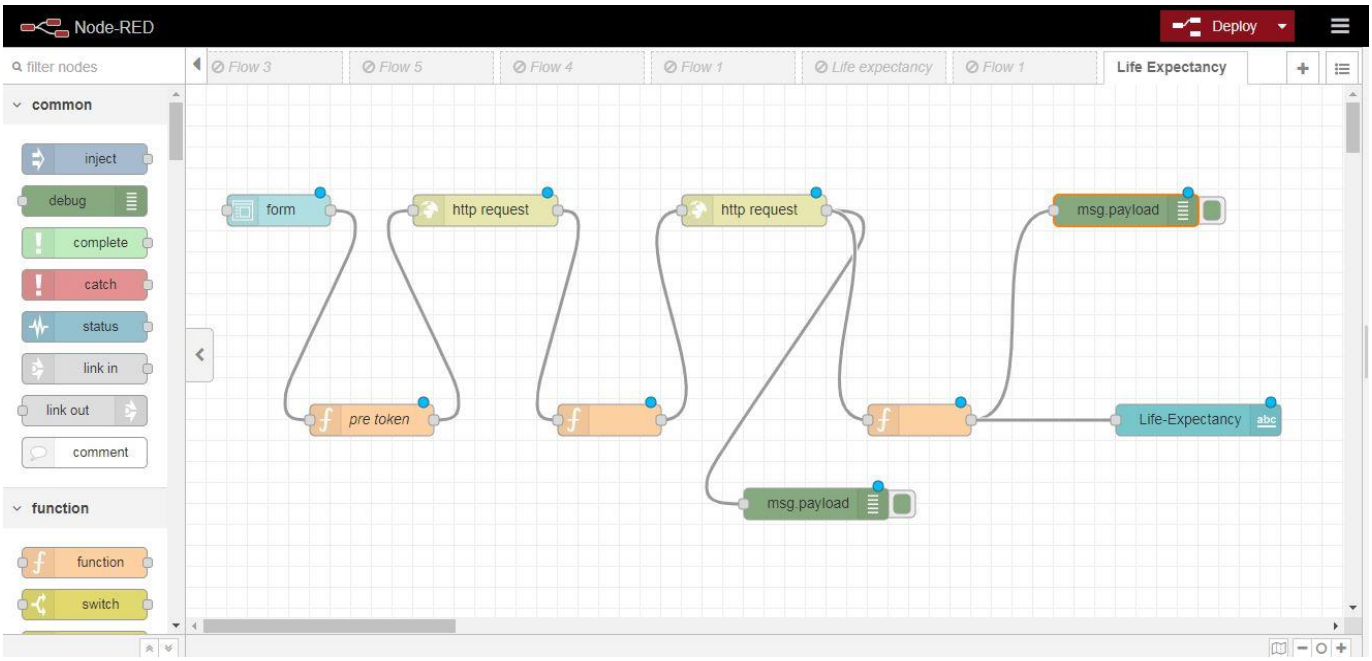
- Install Watson_machine_learning_client
- Import necessary libraries
- Import dataset
- Data Preprocessing
  - Removing unusal species in column names using rename function.
  - Replacing nan values if any with their mean values.
- Exploratory Data Analysis
  - Plotting a heatmap to check if dimensional reduction can be performed.
  - Plotting a pairplot for analysing pairwise relatonship among features.
- Train and Test
  - 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.
  - 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.
  - Then train and test split was peformed and 80% of dataset were trained data and 20% were test data.
  - Then dataset was fitted and predicted.
- Model Building and Deployment
  - At first the machine learning service credentials was stored in a variable and passed into WatsonMachineLearningAPIClient.
  - Then the model was build and stored in model_artifact.
  - Then the model was deployed and scoring_endpoint url was generated.

4.2.3. Create Node-Red Flow

To connect all services together

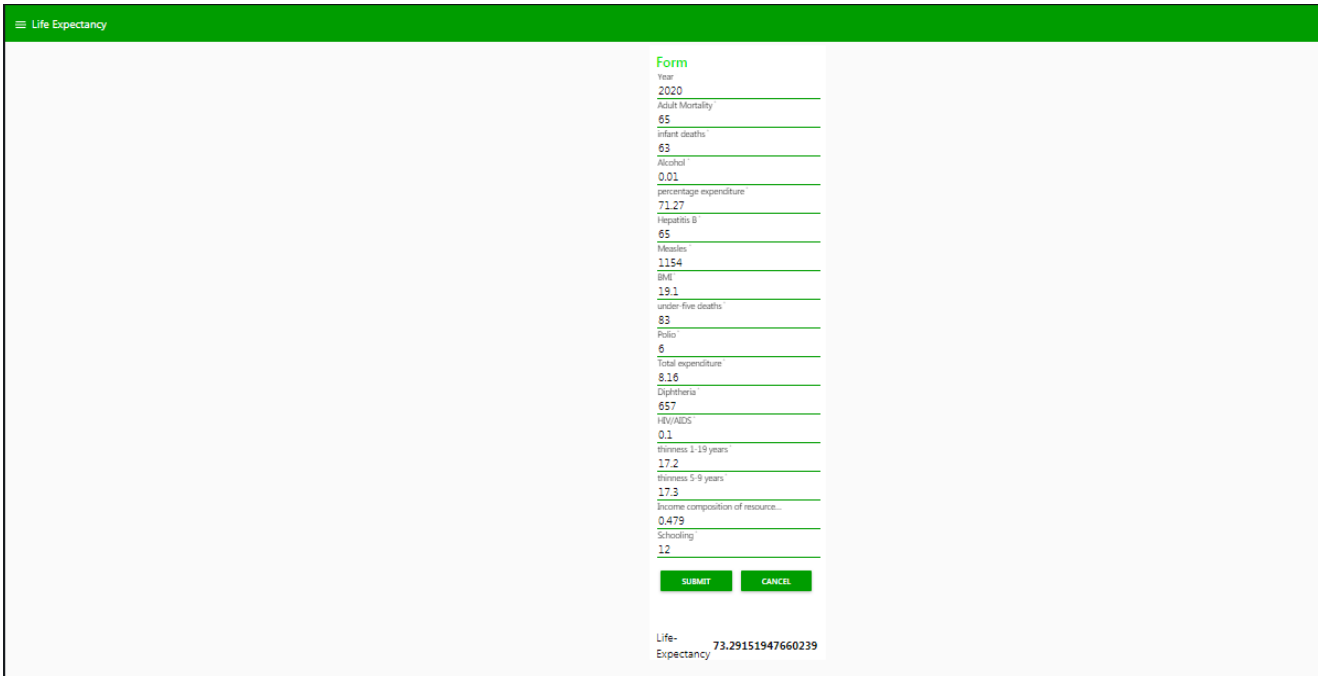
- Go to Node-Red Editor from resource list.
- Install node-red Dashboard from manage pallete.
- Now create the flow with the help of following node.

1. Inject
2. Debug
3. Function
4. Ui_Form
5. Ui_Text
6. Http request



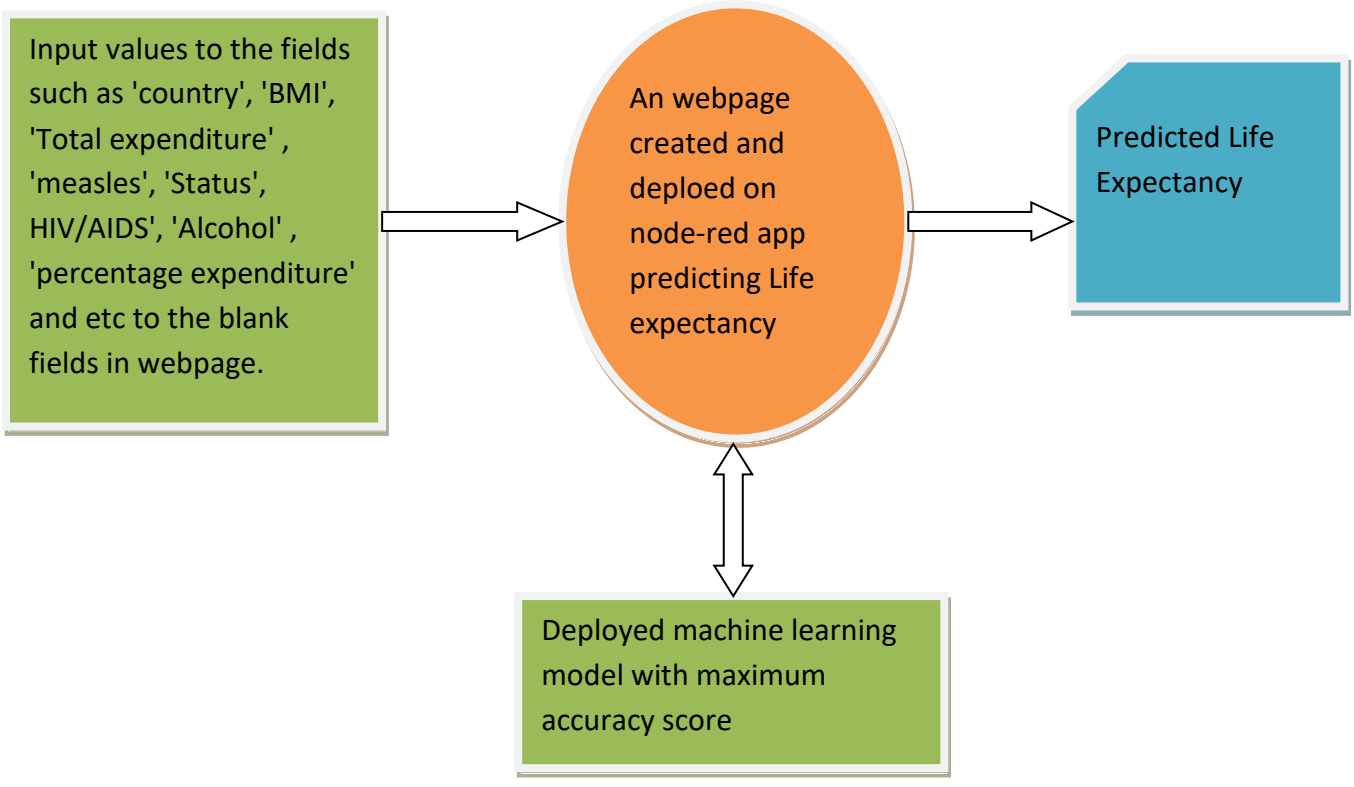
▪ Deploy and run Node Red app

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 .



5. THEORETICAL ANALYSIS

5.1. BLOCK DIAGRAM

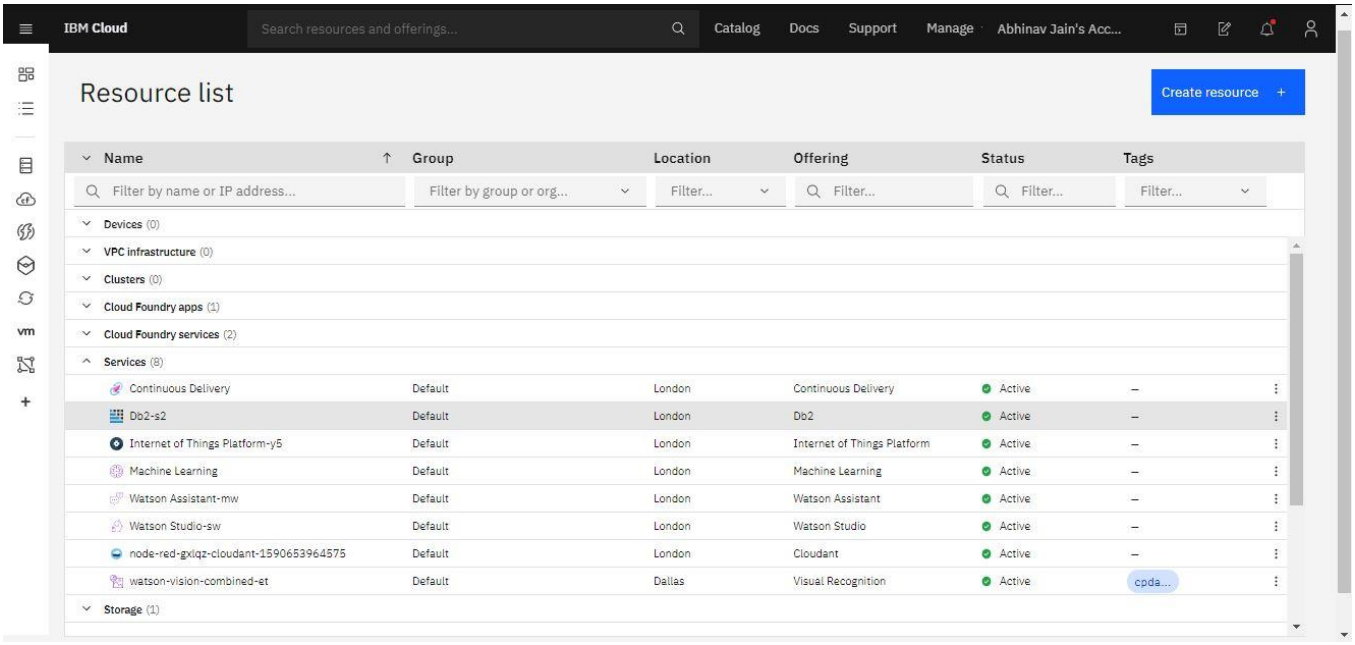


5.2. HARDWARE / SOFTWARE DESIGNING

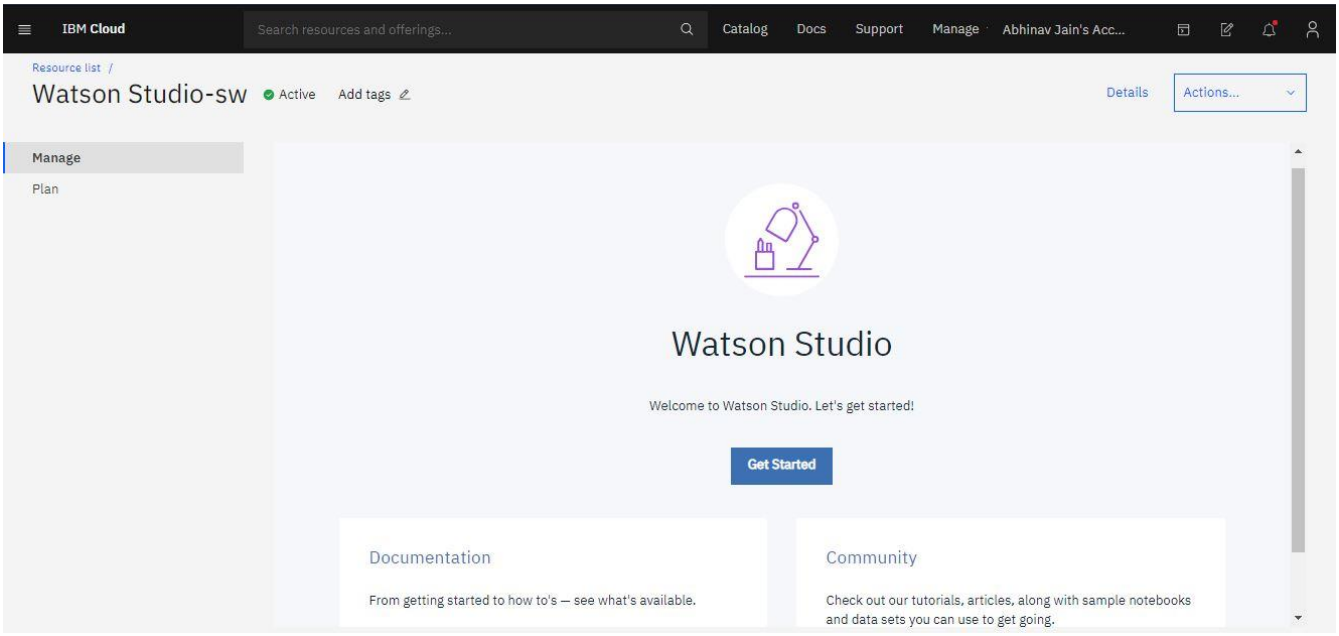
- Project Requirements: Python, IBM Cloud, IBM Watson
- Functional Requirements: IBM cloud
- Technical Requirements: ML, WATSON Studio, Python, Node-Red
- Software Requirements: Watson Studio, Node-Red

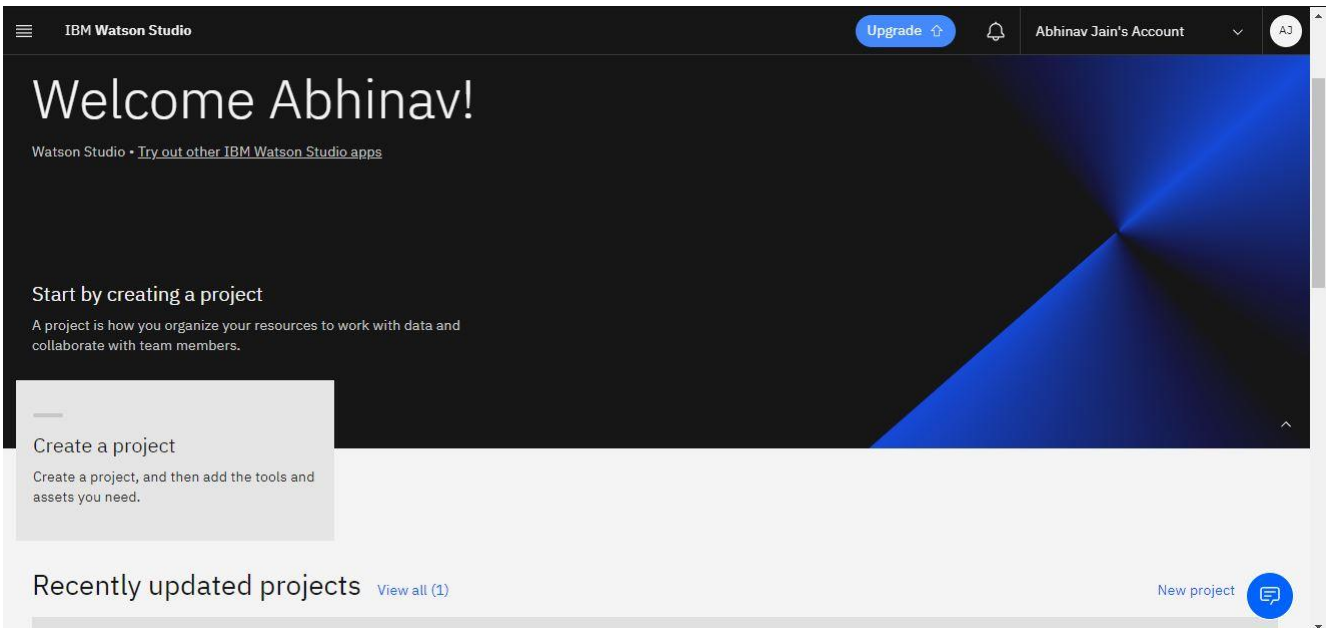
6. EXPERIMENTAL INVESTIGATIONS

A) IBM Cloud Resource List

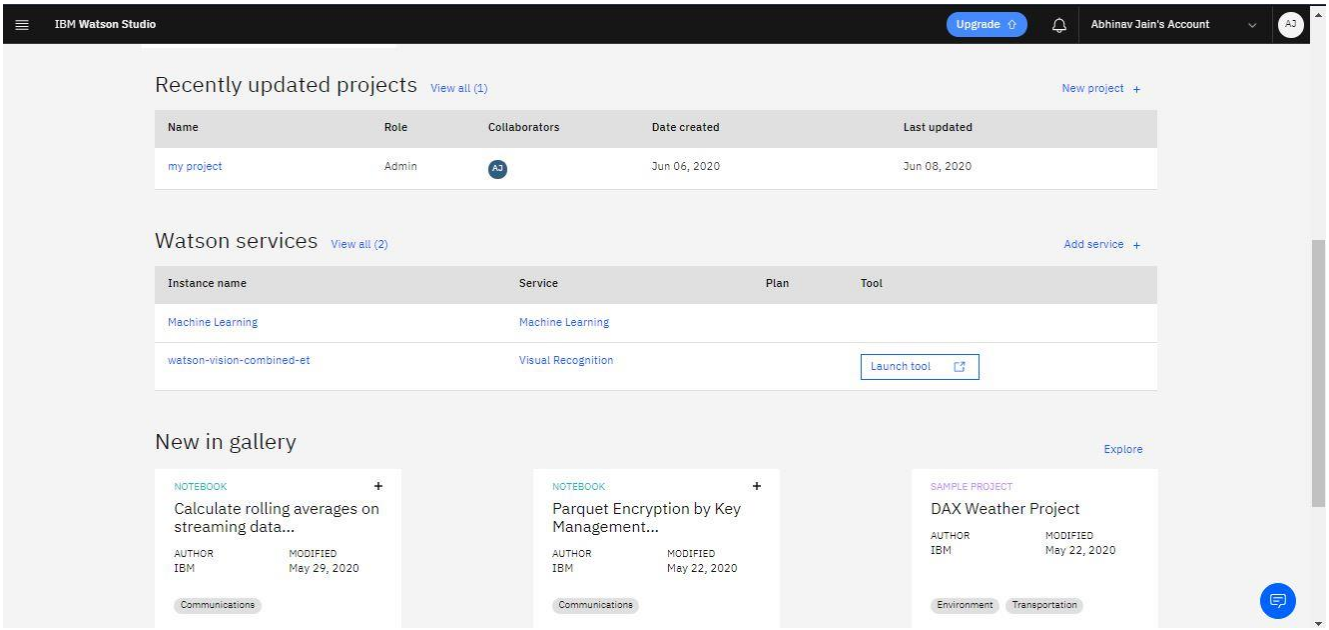


B) IBM Watson Studio

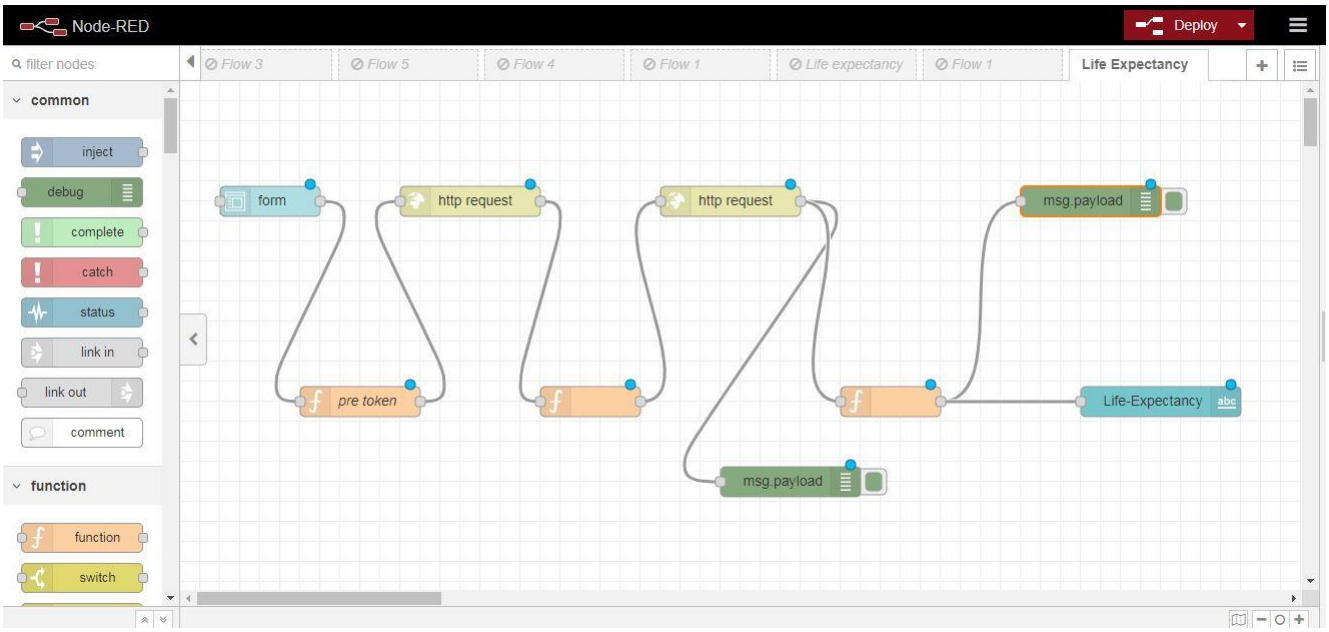




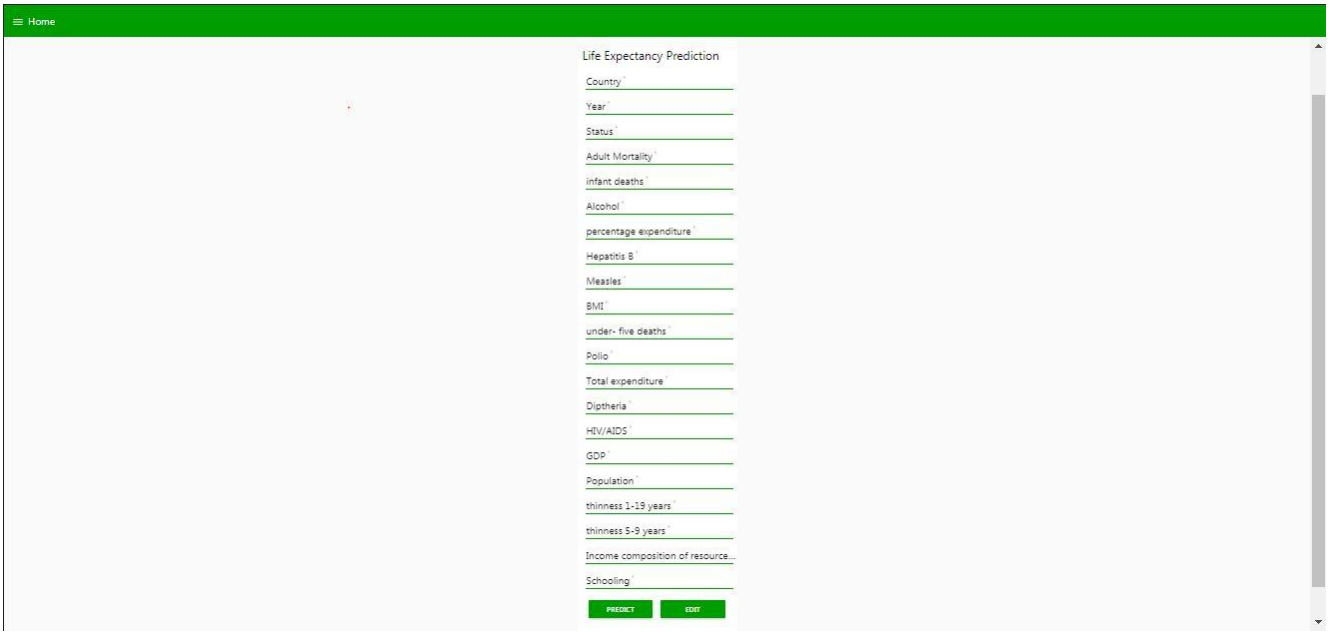
C) IBM Cloud Project Details



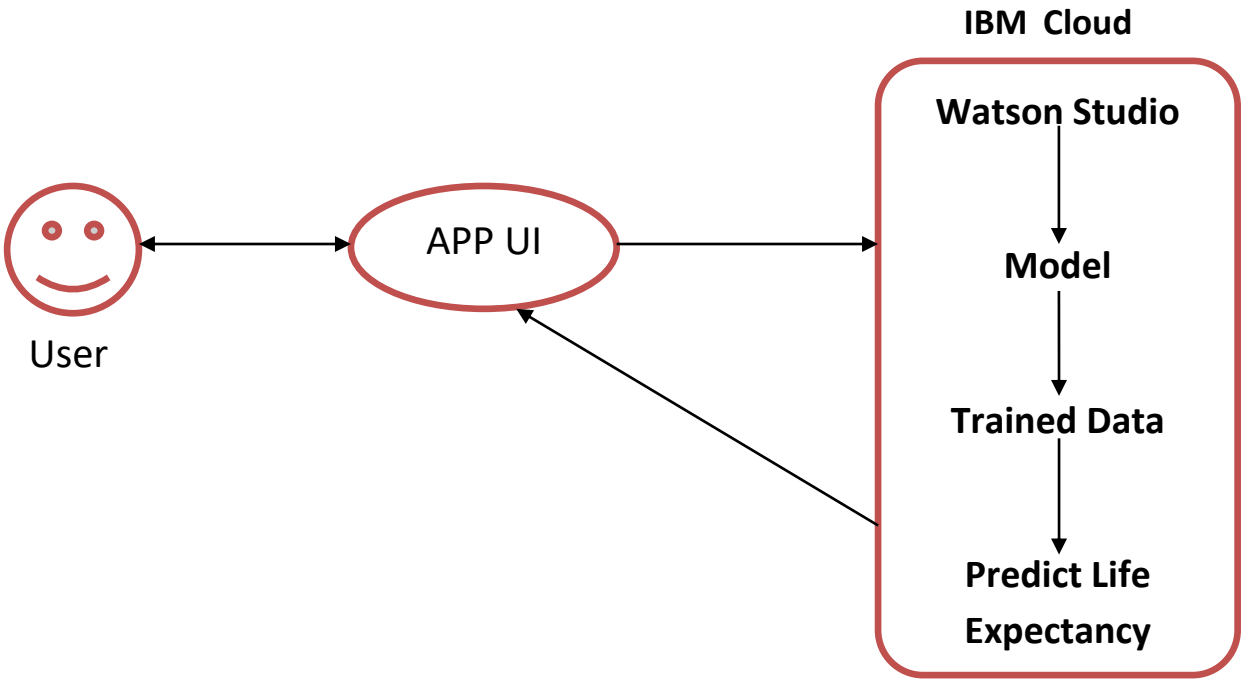
D) Node-Red Flow



E) Life Expectancy Prediction UI



7.FLOWCHART



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

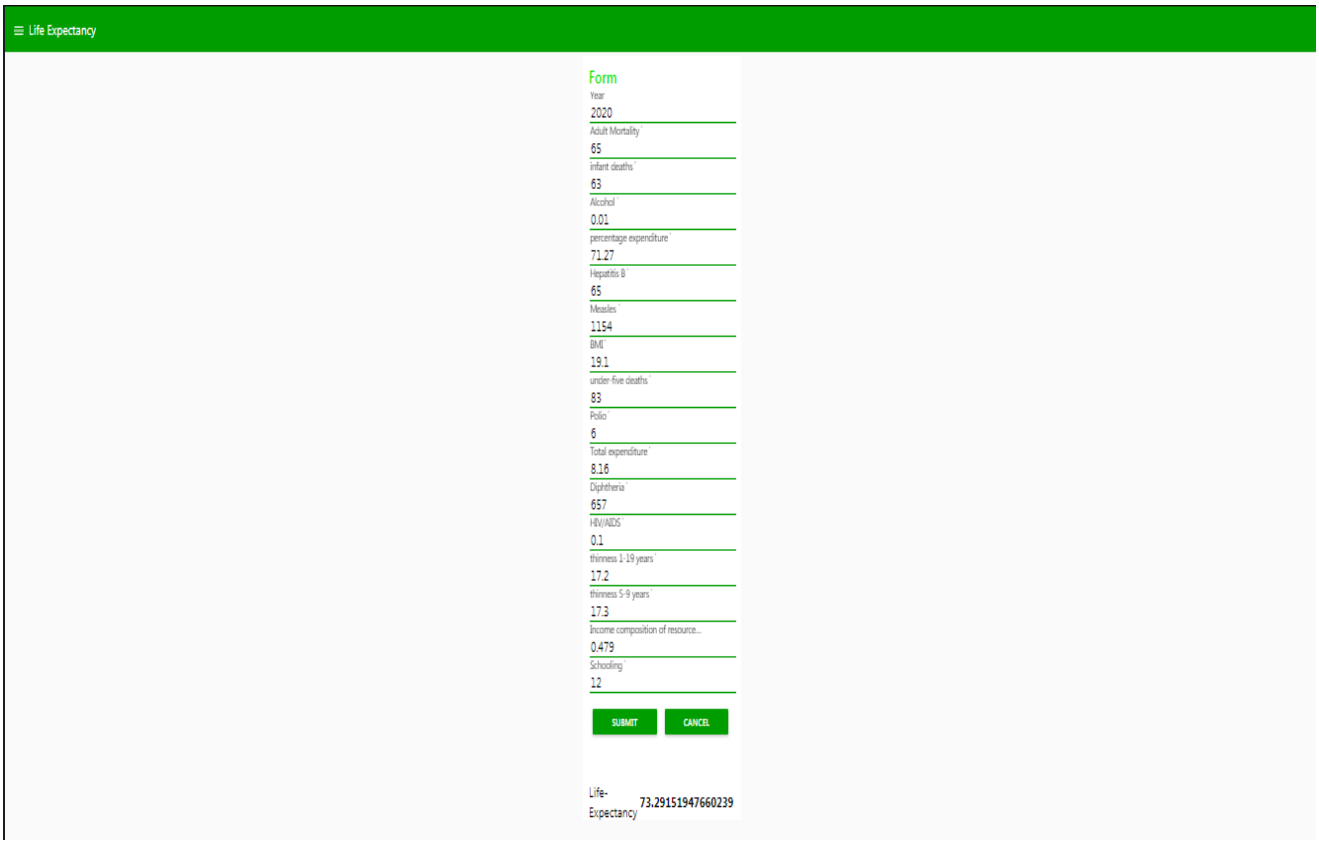
b) The data then entered into watson and the scoring_endpoint url matches with the deployed model.

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

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

8. RESULT

This is the Life Expectancy UI



9. 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.

10. 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.

11. 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.

12. 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 datas 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.

13. BIBLIOGRAPHY

• https://www.kaggle.com/kumarajarshi/life-expectancy-who

• https://www.youtube.com/watch?v=DBRGlAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L

• https://www.youtube.com/watch?v=Jtej3Y6uUng

- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service
- https://bookdown.org/caoying4work/watsonstudio-workshop/auto.html#add-asset-as-auto-ai
- https://www.youtube.com/watch?v=LOCkV-mENq8&feature=youtu.be
- https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

14. APPENDIX

#import libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.model_selection import LinearRegression
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

#Model Building and Deployment

wml_credentials=

{

"apikey": "**********",

"instance_id": "**********",

"url": "**********"

}

client= WatsonMachineLearningAPIClient(wml_credentials)

```
model_props= {
       client.repository.ModelMetaNames.AUTHOR_NAME : "Abhinav Jain",
       client.repository.ModelMetaNames.AUTHOR_EMAIL:"abhinavj65@gmail.com
",     client.repository.ModelMetaNames.NAME : "Life-Expectancy-
Prediction"
    }
```

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

```
deploy = client.deployments.create(guid,name="Life-Expectancy-Prediction")
```

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