

***BY:Yash Chaturvedi***

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Title** | **Page No.** |
| 1 | Introduction | 3 |
| 2 | Literature Survey | 4 |
| 3 | Theoretical Analysis | 5-7 |
| 4 | Experimental Investigations | 8 |
| 5 | Flowchart | 9 |
| 6 | Result | 10-12 |
| 7 | Advantages and Disadvantages | 13 |
| 8 | Applications | 14 |
| 9 | Conclusion | 14 |
| 10 | Future Scope | 15 |
| 11 | Bibliography | 15 |
| 12 | Source Code | 16 |

# Introduction

## Overview

Life expectancy refers to the number of years a person is expected to live. 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.

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.

## Purpose

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.

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.

Personal growth

This project would also help an individual assess his/her lifestyle choices and alter them accordingly to lead a longer and healthier life. It would make them more aware of their general health and its improvement or deterioration over time.

Growth in Health Sector

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.

Insurance Companies

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

# Literature Survey

## Existing Solution

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.

## Proposed Solution

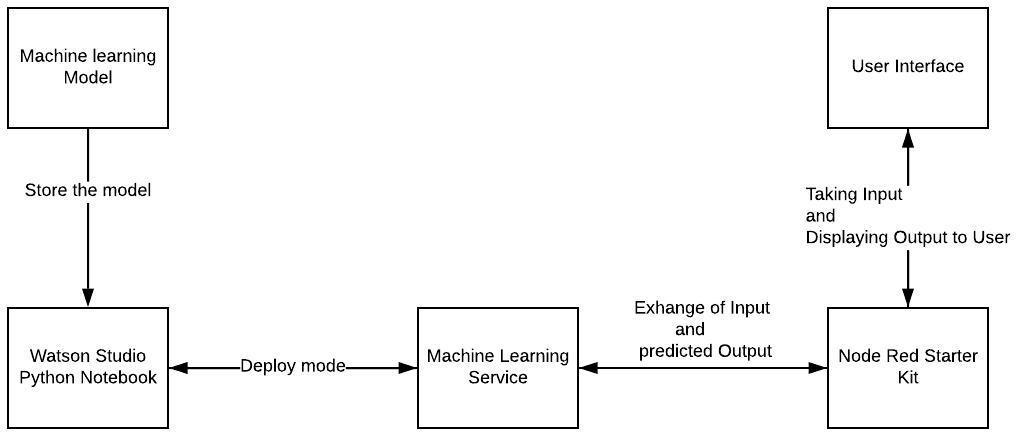
There has been an explosion of breakthroughs in the field of Machine Learning over the past few years. Machine Learning algorithms are capable of a lot and can-do wonders for the healthcare sector.

The proposed solution involves the use of Machine Learning algorithms specifically Regression models such as Linear Regression, Ridge regression, etc. Life expectancy is highly correlated over time among countries and between males and females. These associations can be used to improve forecasts. Here we propose a method for forecasting life expectancy of an individual from a country taking into certain factors such as Adult Mortality rate, Infant deaths, Alcohol, Hepatitis B, Measles, BMI, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP of a country, Population, Income composition of resources, Schooling and status of the country in terms of Developing or Developed.

This machine learning model will be made accessible to the users by integrating it with Node-Red to create an interactive and user-friendly User Interface.

# Theoretical Analysis

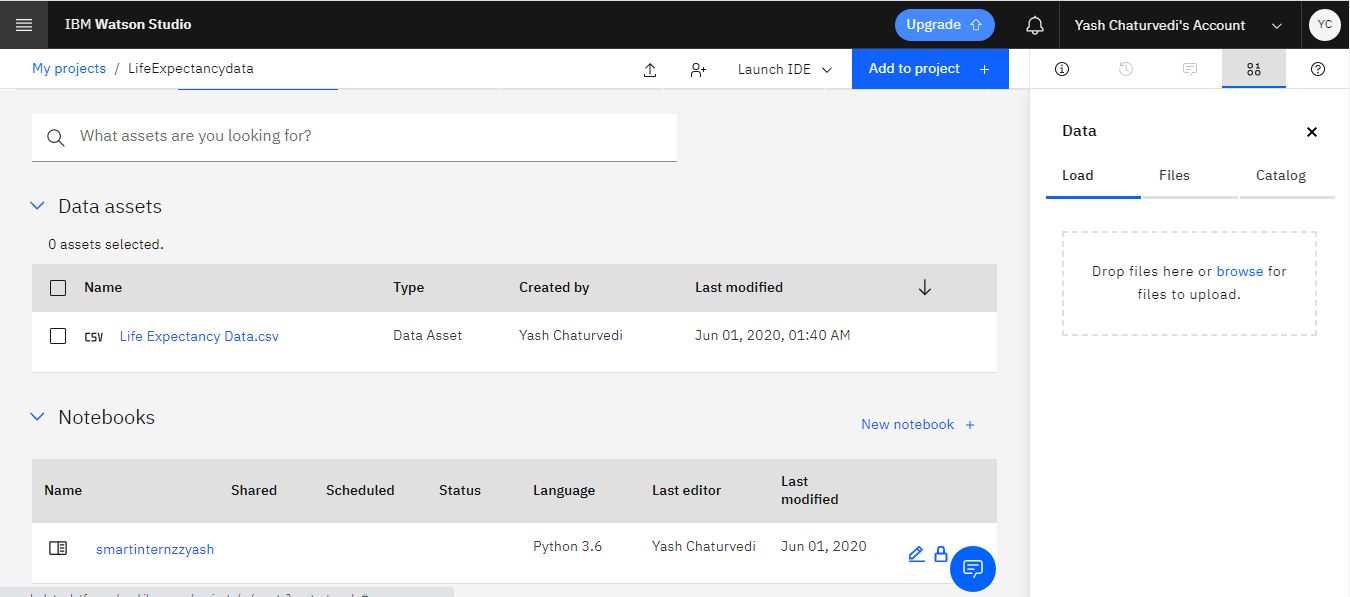
## Block Diagram

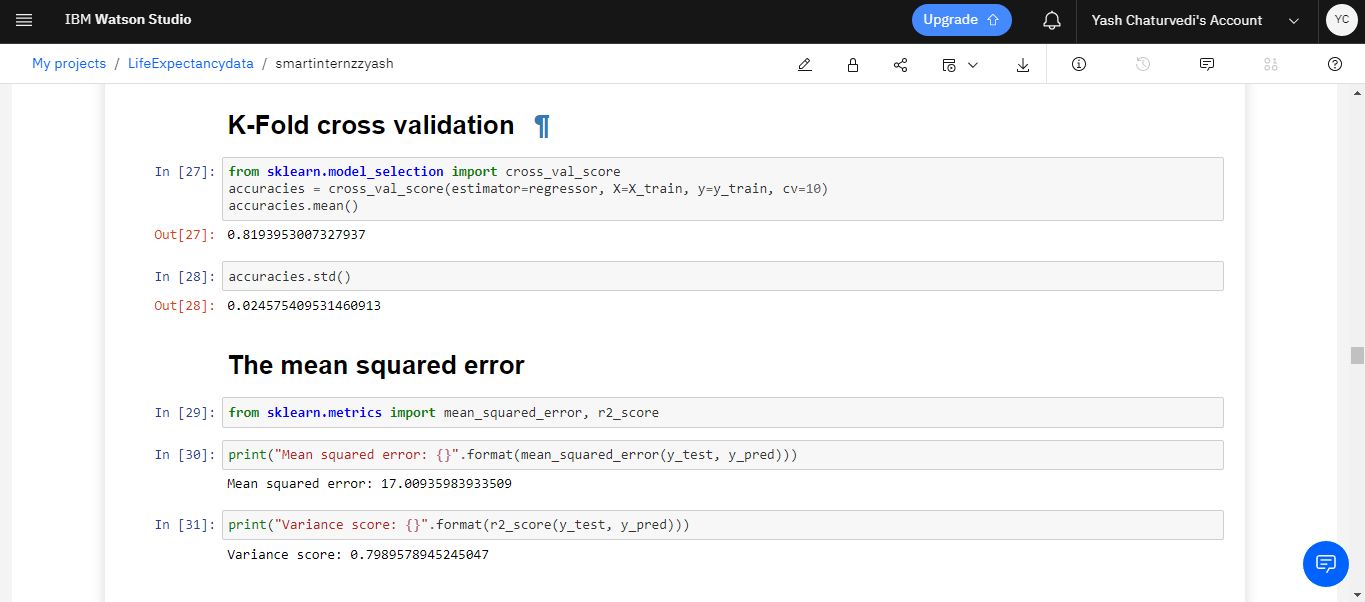


* 1. **Hardware/ Software Designing**

Model Designing **(Watson Studio)** :

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

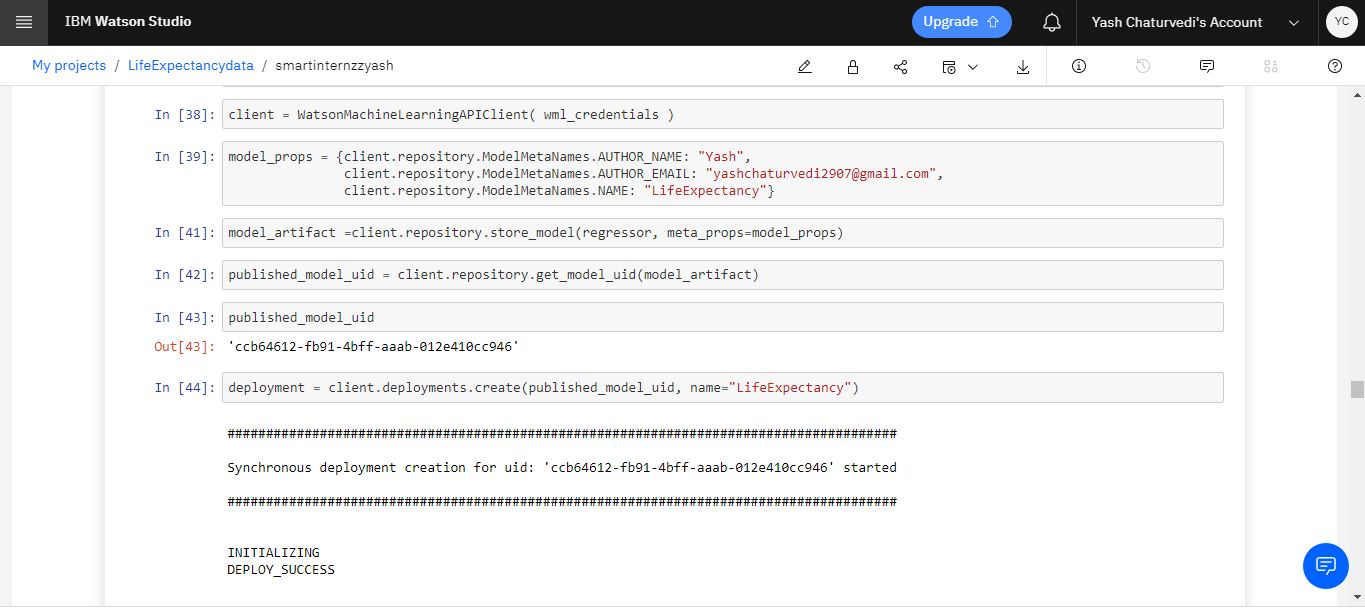


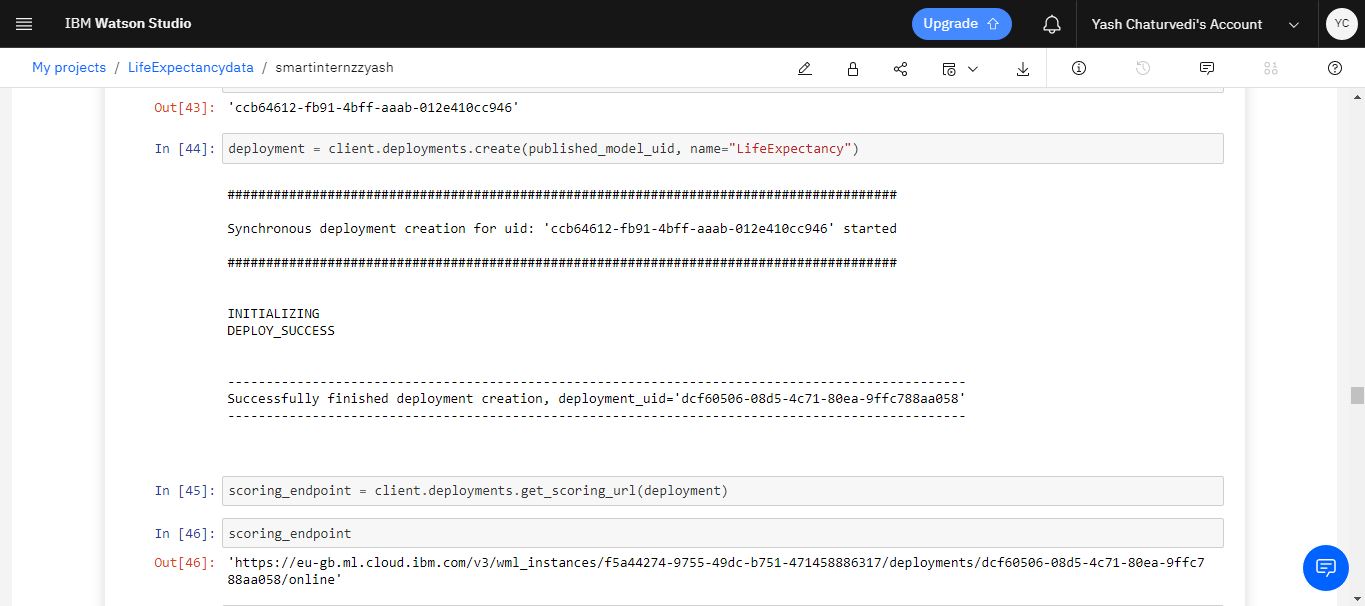


Scoring Endpoint:

For wml credentials, replace with your own credentials of the service.

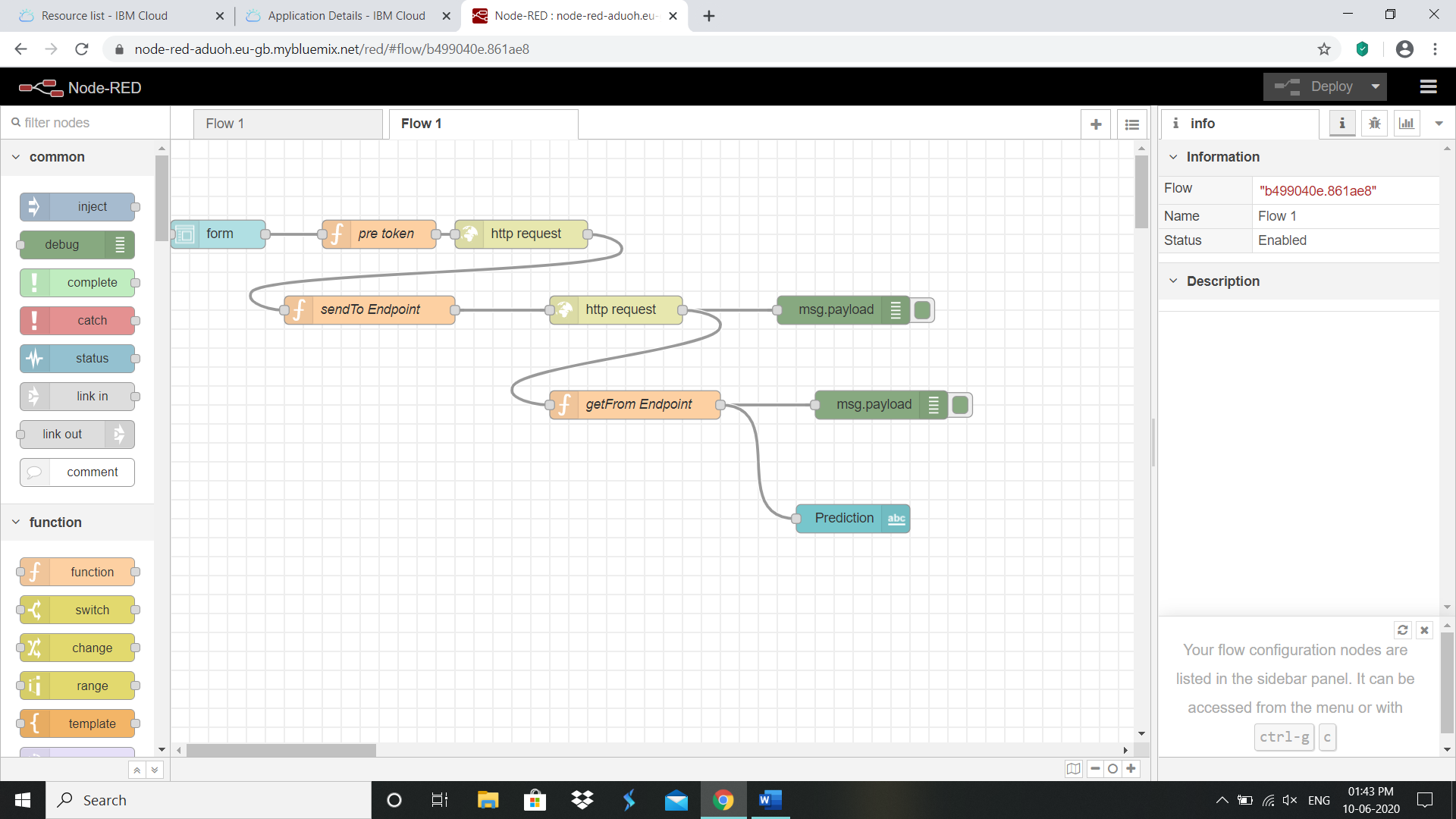
Services => Machine Learning Service => Service Credentials => Copy the credentials

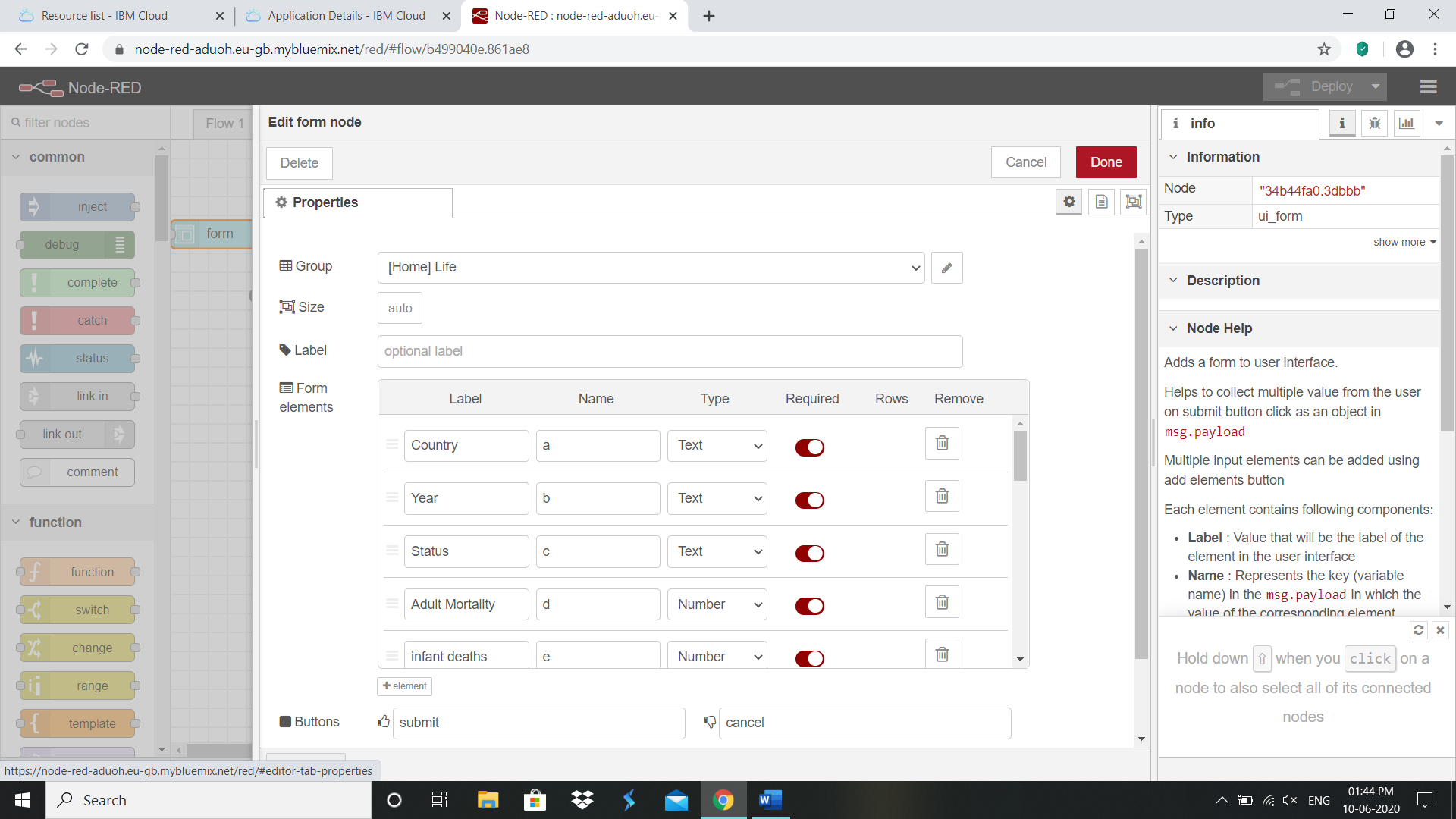




User Interface Integration with ML Model **(Node- Red)** : Nodes: 1) Form Node: Edit => Add New UI Tab

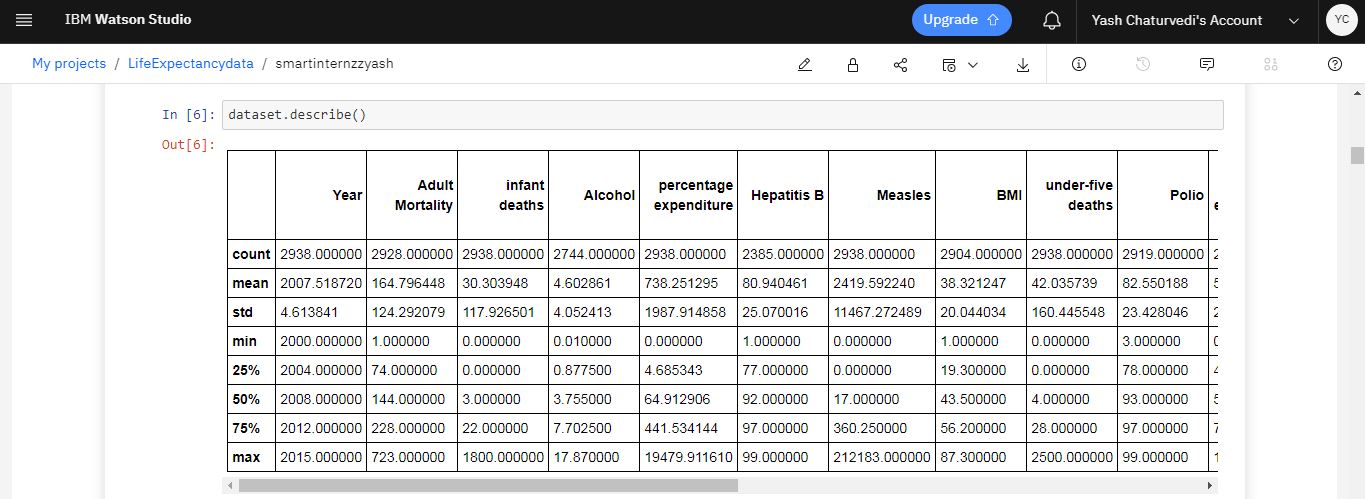
1. Function Node: To obtain access to Machine Learning Services. Requires API Key
2. HTTP Request Node: POST method and returns a parsed JSON object. Gains access to Machine Learning services.

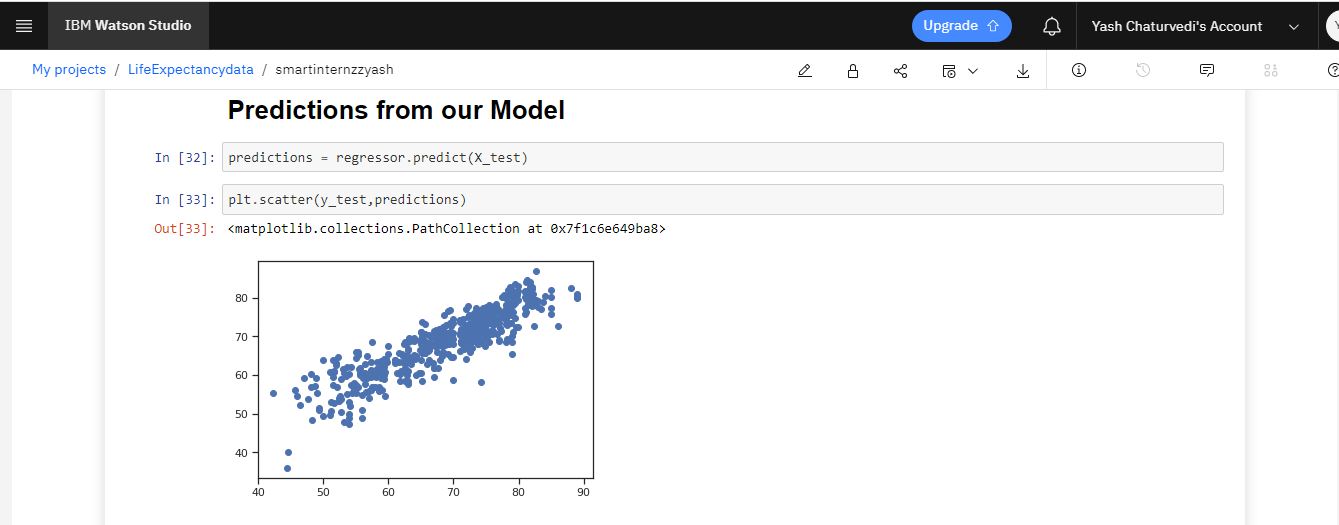


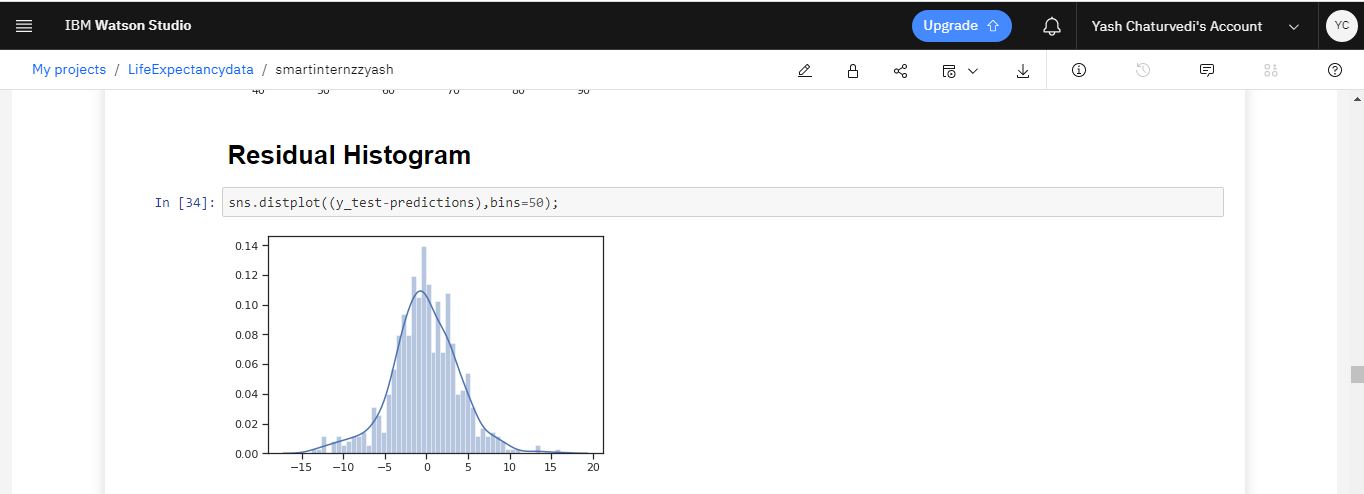


# Experimental Investigations

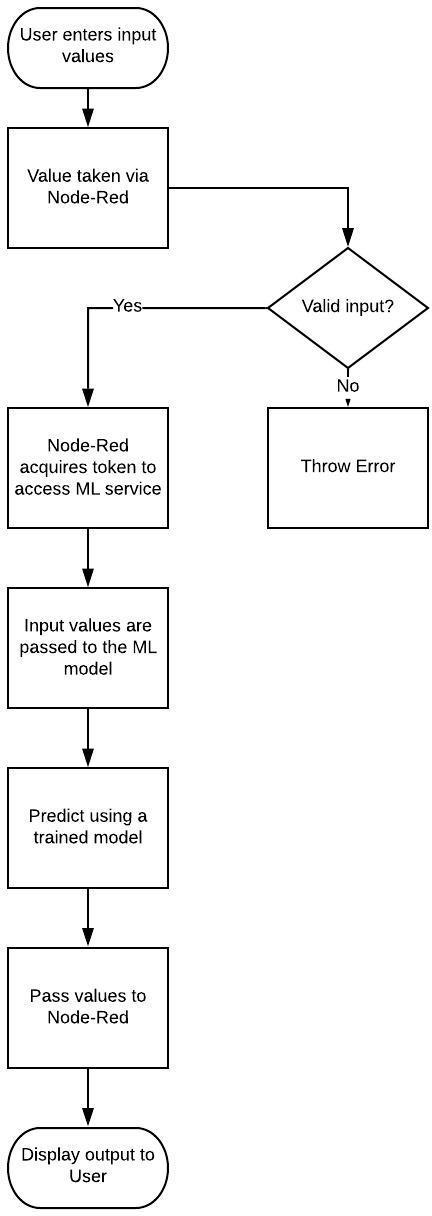
Analyzing the relations between various features can help us improve the performance of the model as well as decide which model would be more suitable.



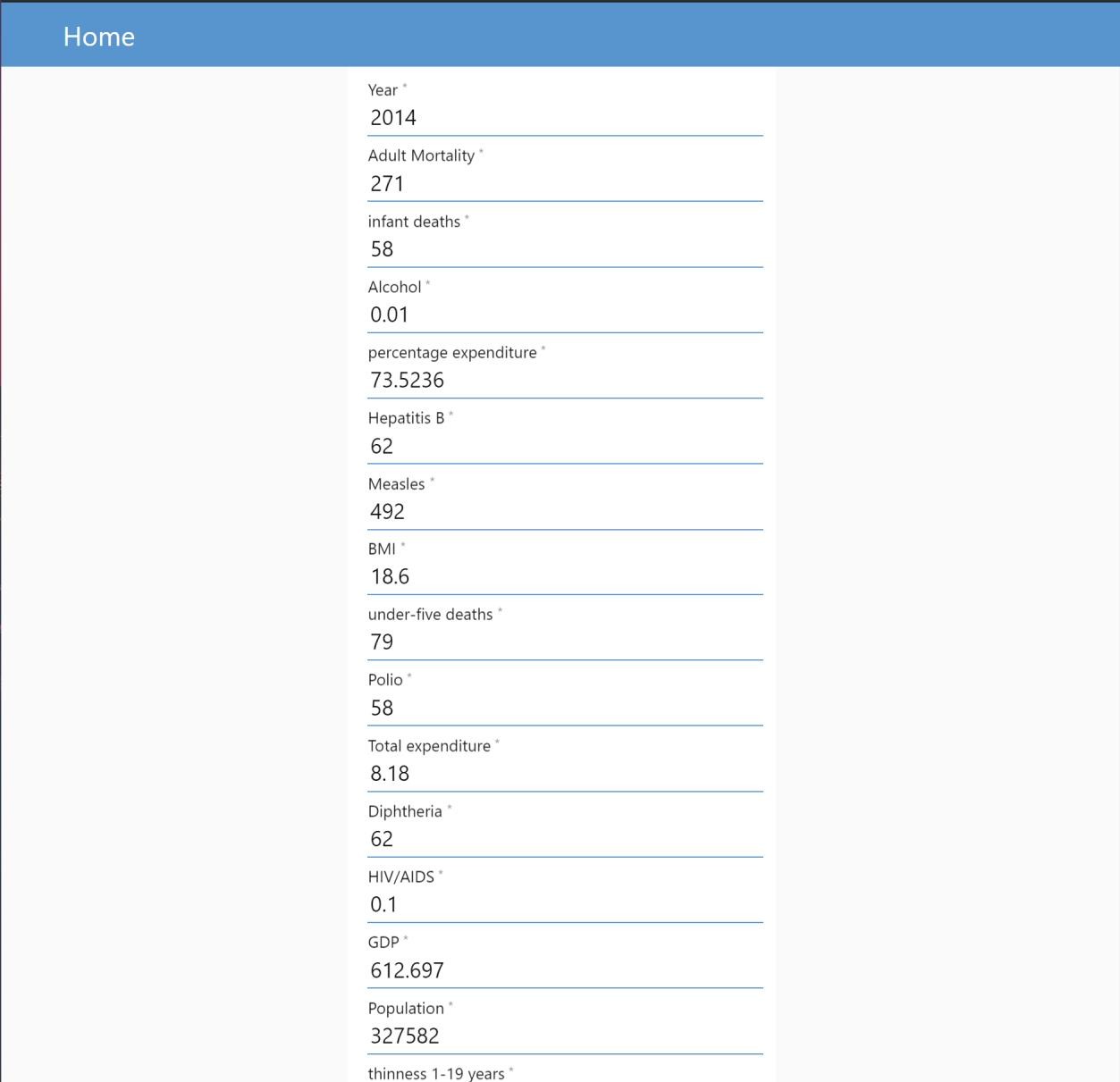


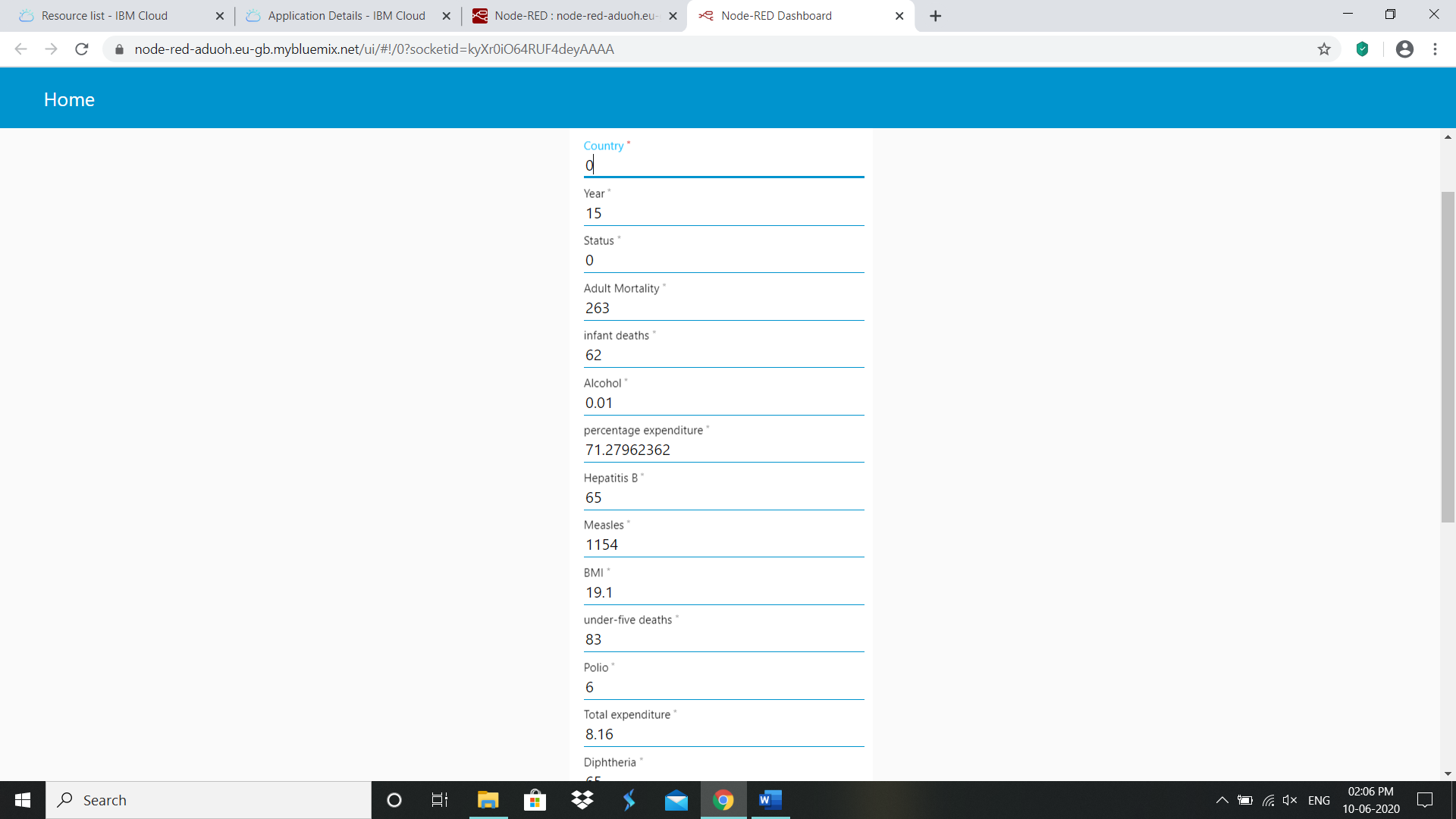


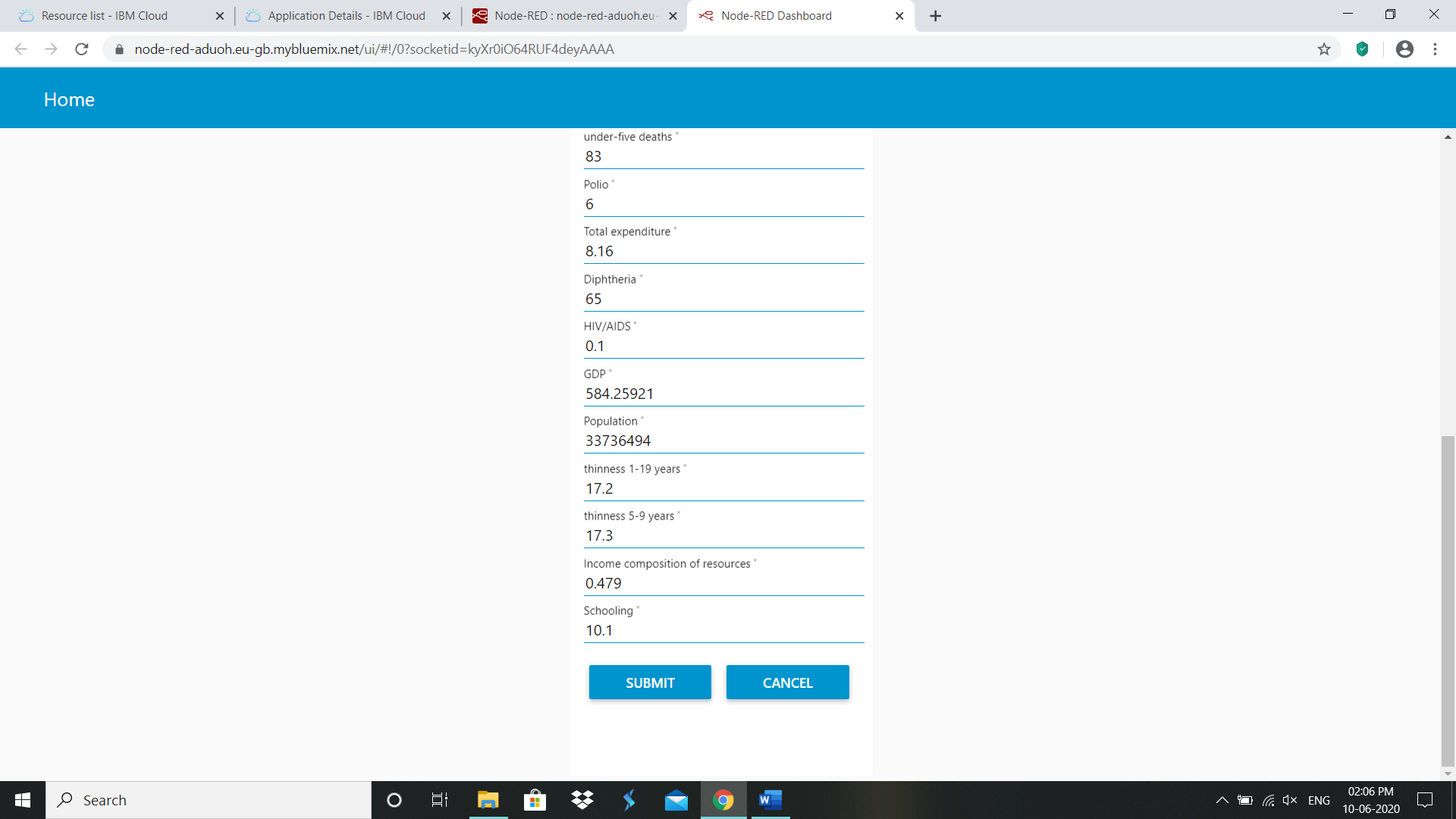
# Flowchart



1. **Result**







1. **Advantages and Disadvantages**

Advantages:

One of the biggest advantages of embedding machine learning algorithms is their ability to improve over time. Machine learning technology typically improves efficiency and accuracy thanks to the ever-increasing amounts of data that are processed.

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.

Using Node-Red also simplifies the effort put into a creating the front-end. The programmer doesn’t need extensive knowledge on HTML and JavaScript. It also makes the integration between Machine learning model and the UI much easier.

Disadvantages:

Using machine learning interface comes with its own problems. Since the whole point of it is minimize human involvement, it also makes error detection and fixing much more problematic. It takes a lot of time to identify the root cause for the problem.

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.

At the same time, Node-Red does not give many features to customize our UI.

# Applications

1. Personalized Life Expectancy: Individuals can predict their own life expectancy by inputting values in the corresponding fields. This could help make people more aware of their general health, and its improvement or deterioration over time. This may motivate them to make healthier lifestyle choices.
2. Government: 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.

1. Health Sector: 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.
2. Insurance Companies: Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

# Conclusion

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

# Future Scope

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.

Also, with the growth in Artificial Neural networks and Deep learning, one can integrate that with our existing application. With the help of Convolutional Neural networks and Computer vision, we can also try to take into account the physical health and appearance of a person.

Mental health can also be taken into account while predicting life expectancy with the help of sentiment analysis systems as well.

# Bibliography

* + <https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>
  + <https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html>
  + [https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as- web-service](https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service)
  + <https://www.ibm.com/watson/products-services>
  + [https://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa 69-9fab-40ee-8457-ea0e5df8c8de](https://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa69-9fab-40ee-8457-ea0e5df8c8de)

# Appendix

## Source Code

**Services Used:**

* + - **Watson Assistant**
    - **Watson Studio**
    - **IBM Cloud Function**
    - **Node-Red**

**Python Notebook:**

Multiple Linear Regression

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns; sns.set(style="ticks", color\_codes=True)

import scipy.stats as stats

import statsmodels.api as sm

# If you are reading an Excel file into a pandas DataFrame, replace `read\_csv` by read\_excel` in the next statement.

dataset= pd.read\_csv(body)

dataset.head()

# Data Preprocessing

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 21].values

dataset.head()

dataset.info()

dataset.describe()

dataset.columns

sns.heatmap(dataset.isnull(),yticklabels=False,cbar=False,cmap='viridis')

dataset.corr()

# Taking care of missing data

from sklearn.impute import SimpleImputer #sklearn is library

imputer = SimpleImputer(missing\_values=np.nan, strategy='mean') #nan is not a number

imputer = imputer.fit(dataset.iloc[:, 3:]) # fit calculates mean

dataset.iloc[:, 3:] = imputer.transform(dataset.iloc[:, 3:]) #transform means to apply

# Encoding categorical data i.e. converting categorical to numerical

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder\_X = LabelEncoder()

dataset.iloc[:, :1] = labelencoder\_X.fit\_transform(dataset.iloc[:, :1])

# Encoding categorical data i.e. converting categorical to numerical

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder\_X = LabelEncoder()

dataset.iloc[:, 1] = labelencoder\_X.fit\_transform(dataset.iloc[:, 1])

# Encoding categorical data ie converting categorical to numerical

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder\_X = LabelEncoder()

dataset.iloc[:, 2] = labelencoder\_X.fit\_transform(dataset.iloc[:, 2])

dataset.iloc[:, :1]

dataset.iloc[:, 1]

dataset.iloc[:, 2]

dataset

X= dataset.iloc[:, :-1].values

y = dataset.iloc[:, 21].values

X

Y

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)

# Fitting Multiple Linear Regression to the Training set

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, y\_train)

# Predicting the Test set results

y\_pred = regressor.predict(X\_test)

y\_pred

y\_test

# K-Fold cross validation

from sklearn.model\_selection import cross\_val\_score

accuracies = cross\_val\_score(estimator=regressor, X=X\_train, y=y\_train, cv=10)

accuracies.mean()

accuracies.std()

# The mean squared error

# from sklearn.metrics import mean\_squared\_error, r2\_score

print("Mean squared error: {}".format(mean\_squared\_error(y\_test, y\_pred)))

print("Variance score: {}".format(r2\_score(y\_test, y\_pred)))

# Predictions from our Model

predictions = regressor.predict(X\_test)

plt.scatter(y\_test,predictions)

# Residual Histogram

sns.distplot((y\_test-predictions),bins=50);

!pip install watson-machine-learning-client

from watson\_machine\_learning\_client import WatsonMachineLearningAPIClient

client = WatsonMachineLearningAPIClient( wml\_credentials )

model\_artifact =client.repository.store\_model(regressor, meta\_props=model\_props)

published\_model\_uid = client.repository.get\_model\_uid(model\_artifact)

published\_model\_uid

deployment = client.deployments.create(published\_model\_uid, name="LifeExpectancy")

scoring\_endpoint = client.deployments.get\_scoring\_url(deployment)

scoring\_endpoint

## Node Red Flow:

