Predicting Life Expectancy Using Machine Learning

BY:Anjali Praveen Mishra

INDEX

1. Introduction

1.1. 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.

1.2. Purpose

<u>Economic growth</u>

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.

<u>Population Growth</u>

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.

<u>Personal growth</u>

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.

<u>Growth in Health Sector</u>

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.

<u>Insurance Companies</u>

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

2. Literature Survey

2.1. 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.

2.2. Proposed Solution

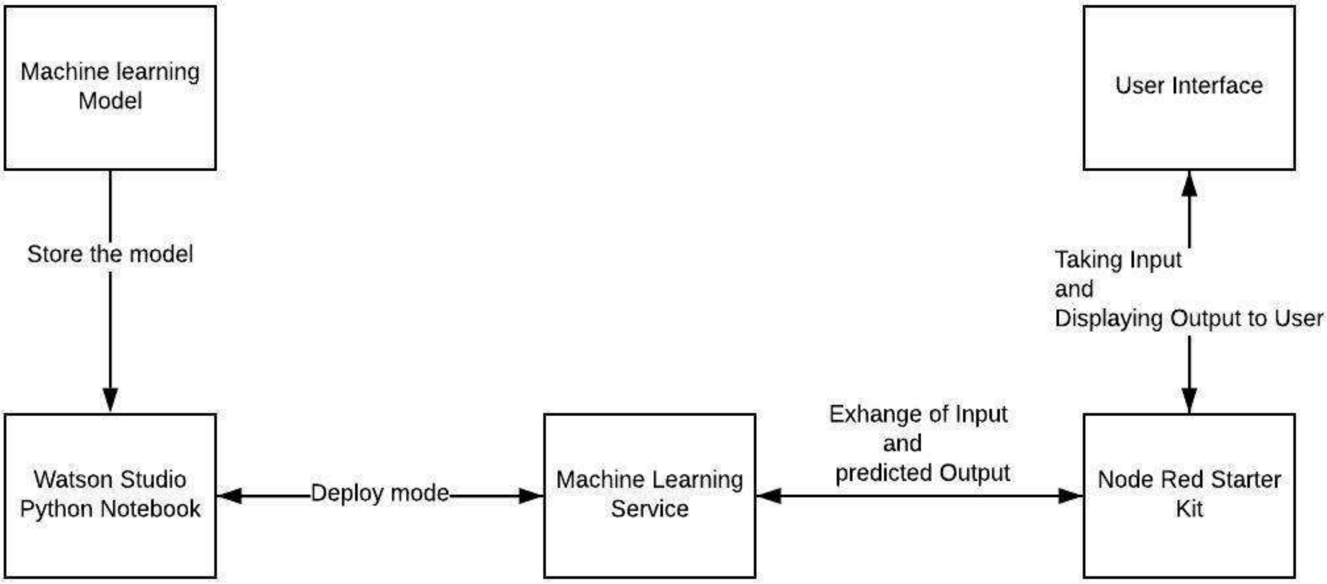
There has been an explosion of breakthroughs in the field of Machine Learning over the past few years. Machine Learning algorithms are capable of a lot and can-do wonders for the healthcare sector.

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.

3. Theoretical Analysis

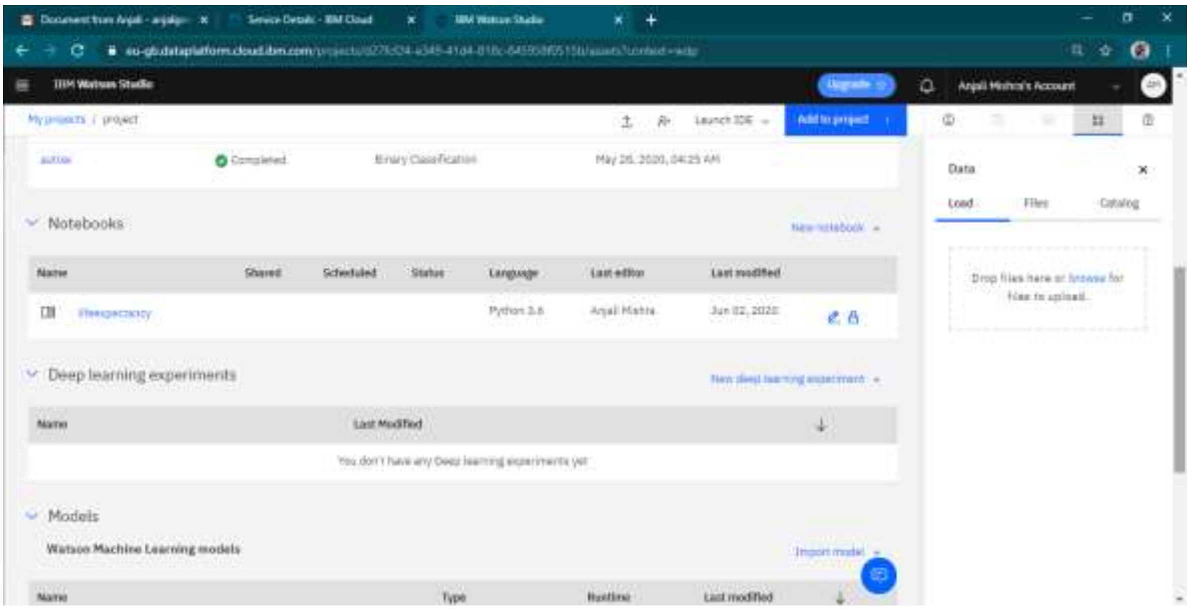
3.1. Block Diagram



3.2. Hardware/ Software Designing

Model Designing **(Watson Studio)** :

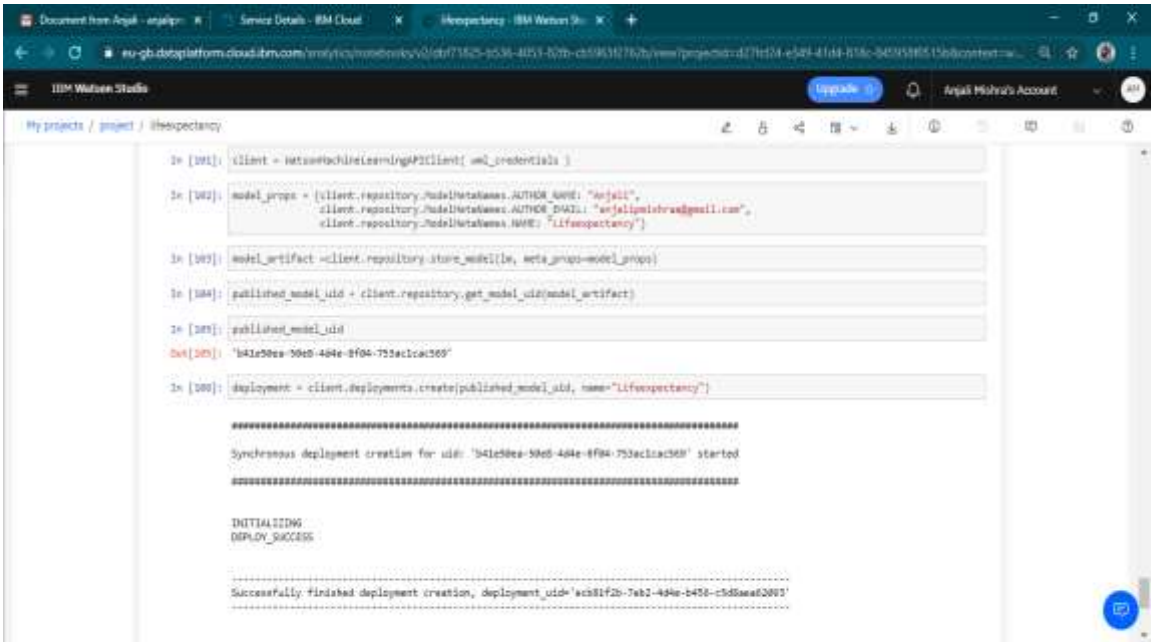
Steps: New Project => Create an empty Project => Give project name => Click Create => Add to Project => Notebook

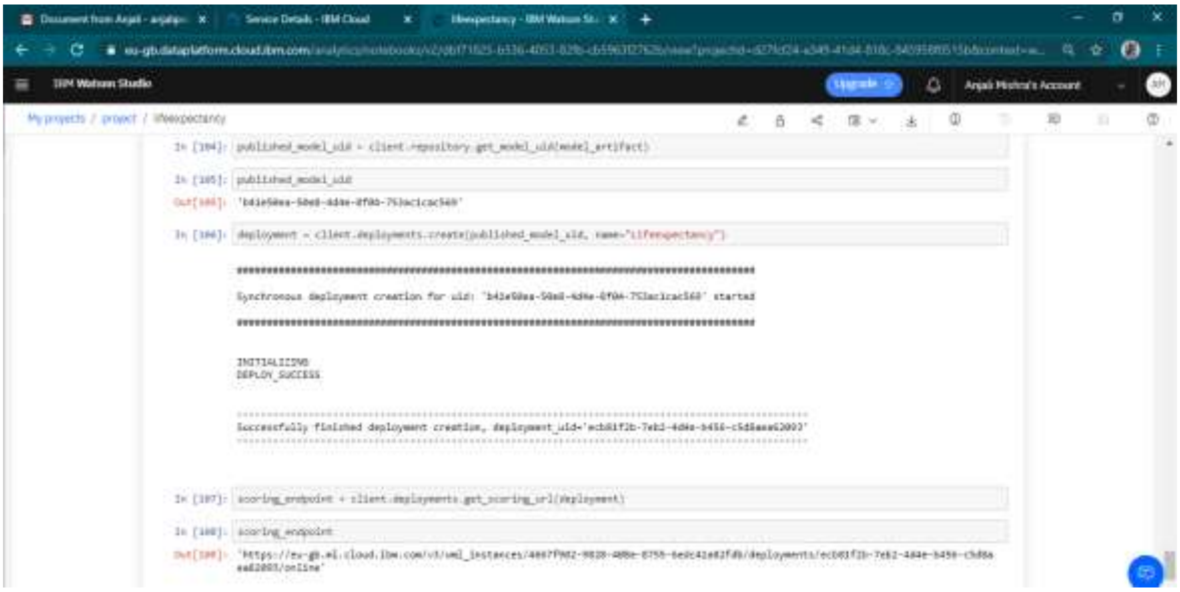


<u>Scoring Endpoint:</u>

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

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

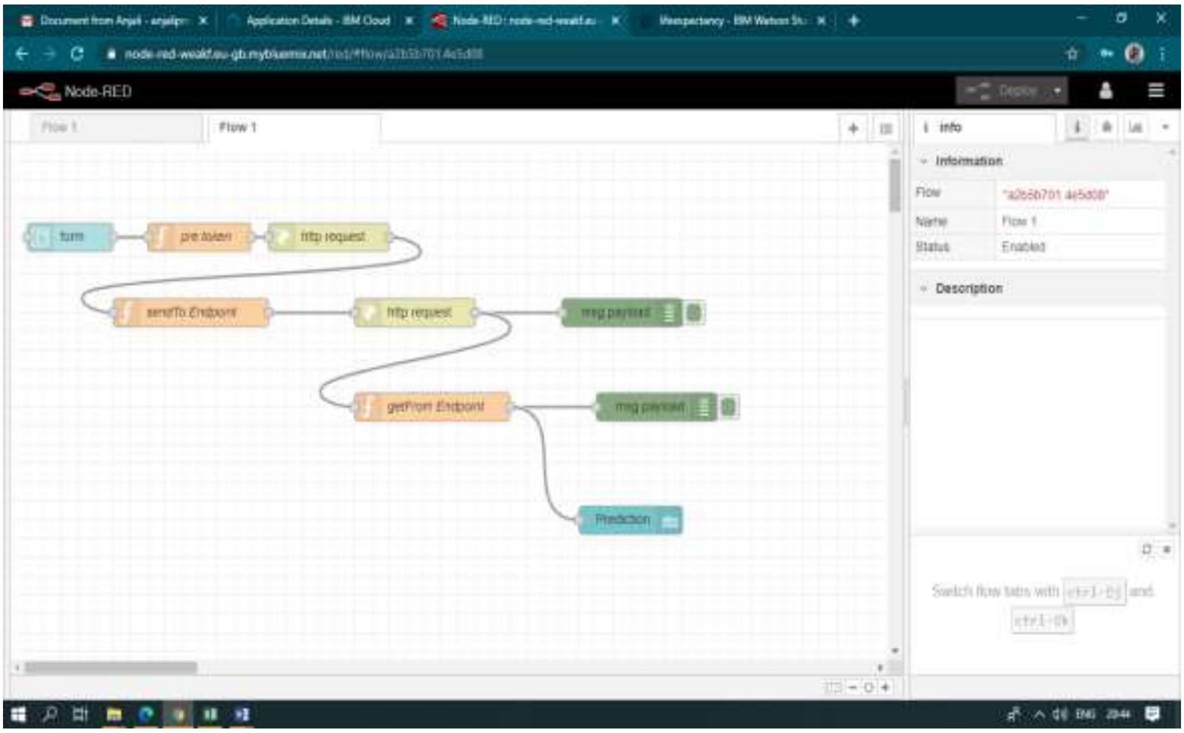


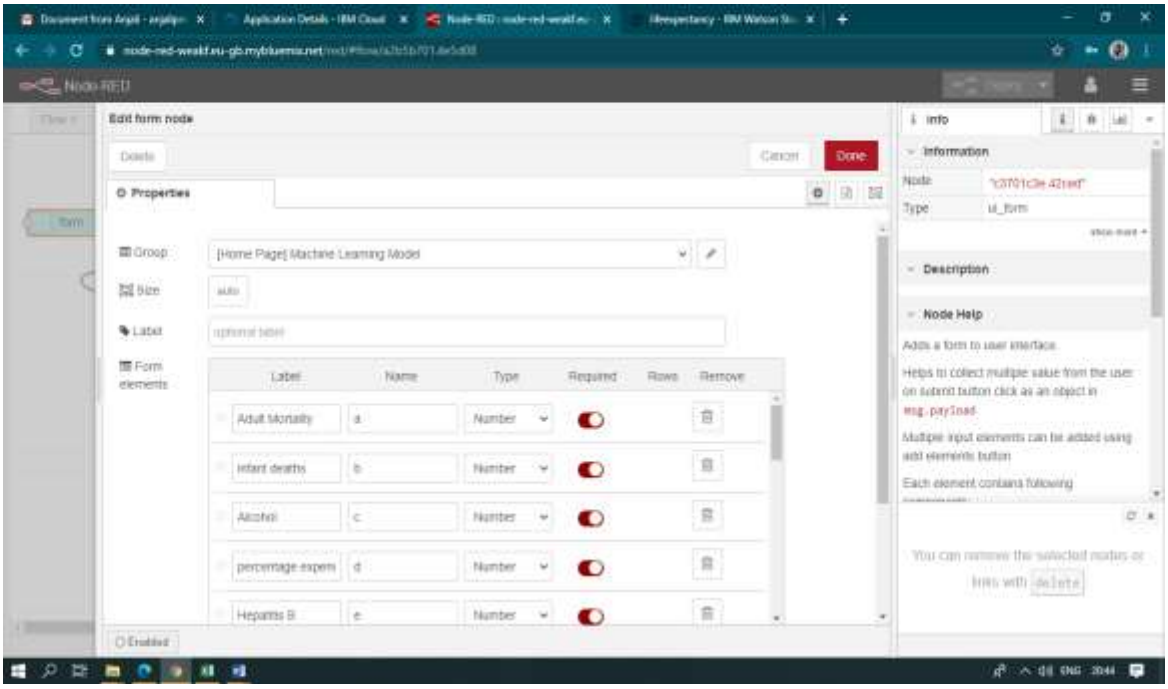


User Interface Integration with ML Model **(Node-Red)** : Nodes: 1) Form Node: Edit => Add New UI Tab

2) Function Node: To obtain access to Machine Learning Services. Requires API Key

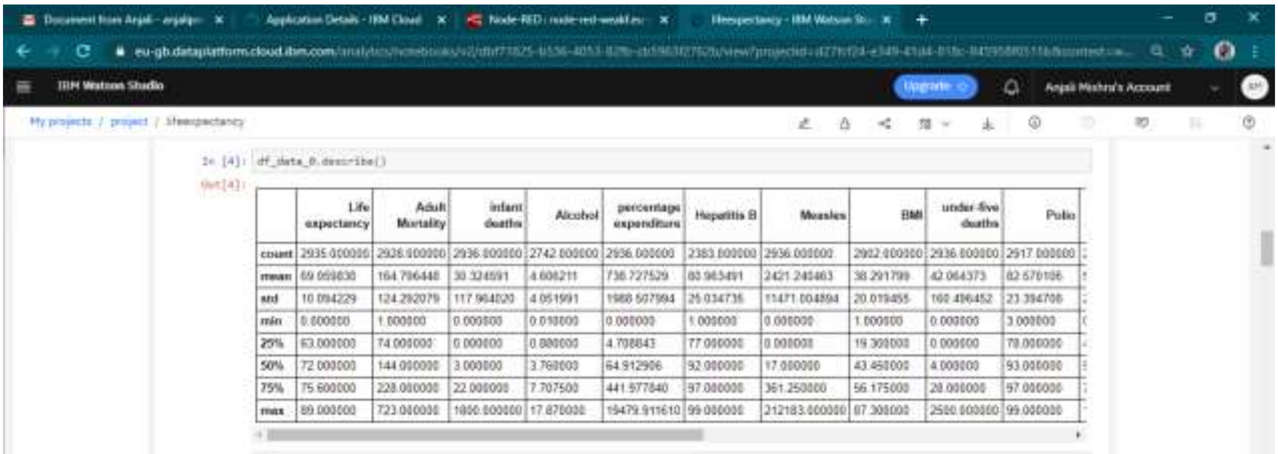
3) HTTP Request Node: POST method and returns a parsed JSON object. Gains access to Machine Learning services.


Node-RED
Flow 1
form
pre token
http request
sendTo Endpoint
http request
msg.payload
getFrom Endpoint
msg.payload
Prediction
info
Information
Flow
Name
Flow 1
Status
Enabled
Description

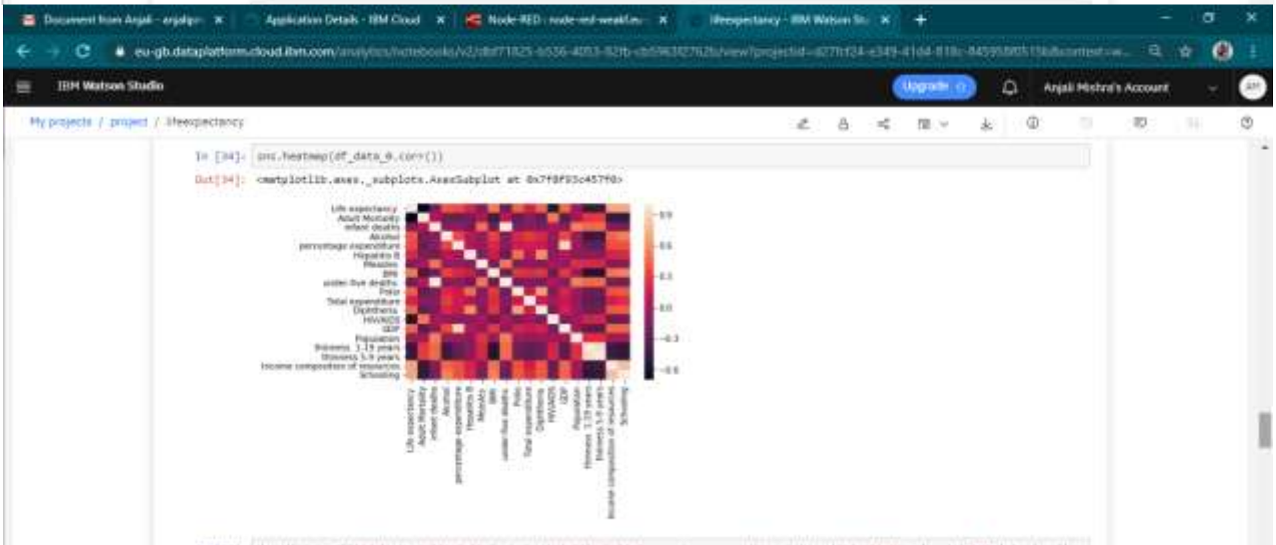

Node-RED
Edit form node
Delete
Cancel
Done
Properties
Group
[Home Page] Machine Learning Model
Size
auto
Label
optional label
Form elements
Label
Name
Type
Required
Rows
Remove
Adult Mortality
a
Number
Infant deaths
b
Number
Alcohol
c
Number
percentage expen
d
Number
Hepatitis B
e
Number
Enabled
info
Information
Node
Type
ui_form
Description
Node Help
Adds a form to user interface.

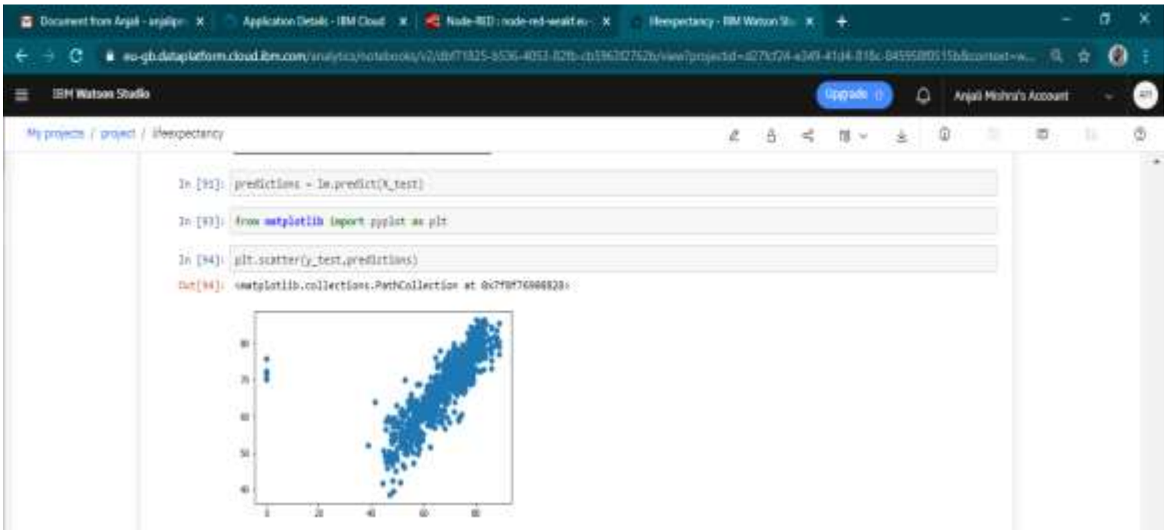
4. Experimental Investigations

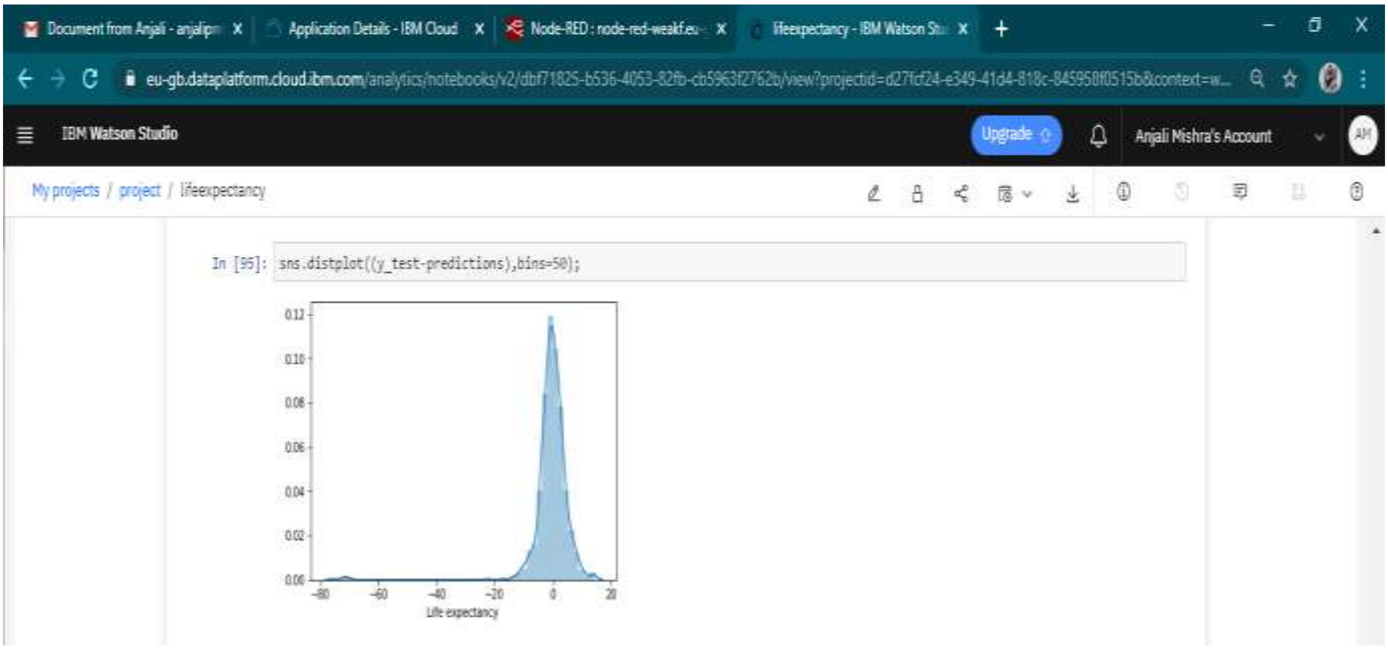
Analyzing the relations between various features can help us improve the performance of the model as well as decide which model would be more suitable.



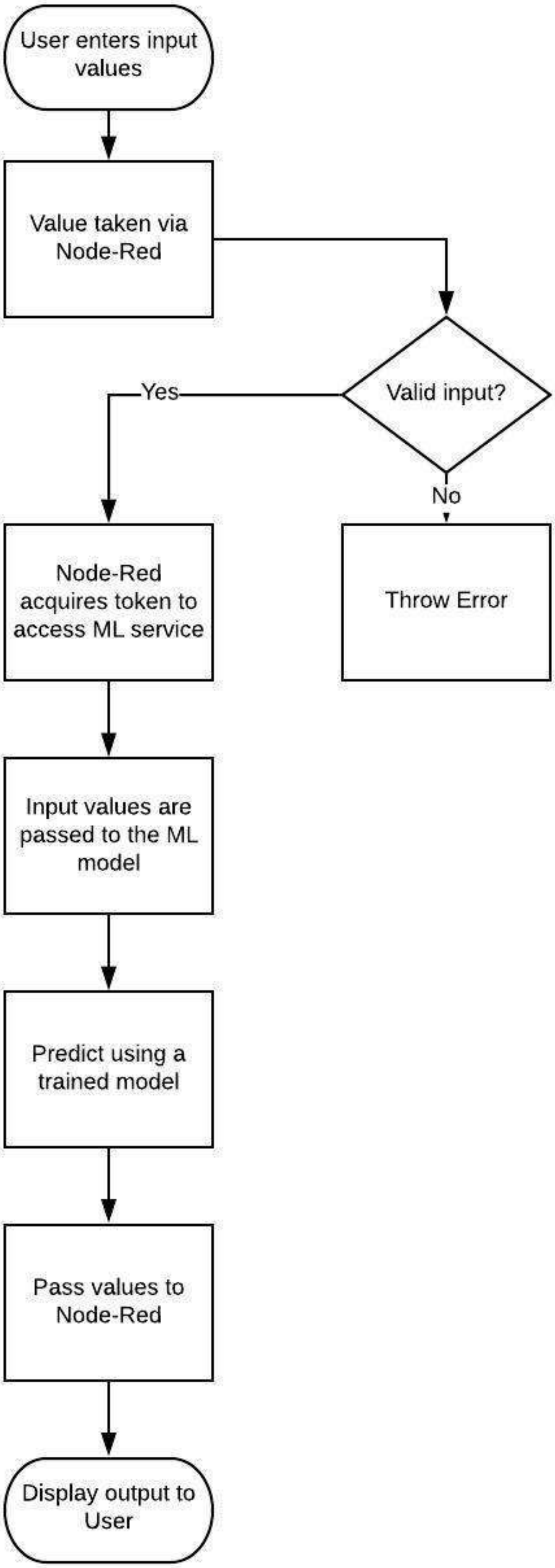
	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio
count	2935.000000	2928.000000	2938.000000	2742.000000	2938.000000	2383.000000	2938.000000	2902.000000	2938.000000	2917.000000
mean	69.059830	164.796448	30.324591	4.608211	738.727529	80.963491	2421.240463	38.291799	42.064373	82.670106
std	10.094229	124.292079	117.964020	4.051991	1988.507994	25.034735	11471.004894	20.019455	160.496452	23.394708
min	0.000000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000	3.000000
25%	63.000000	74.000000	0.000000	0.880000	4.708843	77.000000	0.000000	19.300000	0.000000	78.000000
50%	72.000000	144.000000	3.000000	3.760000	64.912906	92.000000	17.000000	43.450000	4.000000	93.000000
75%	75.600000	228.000000	22.000000	7.707500	441.977840	97.000000	361.250000	56.175000	28.000000	97.000000
max	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2500.000000	99.000000



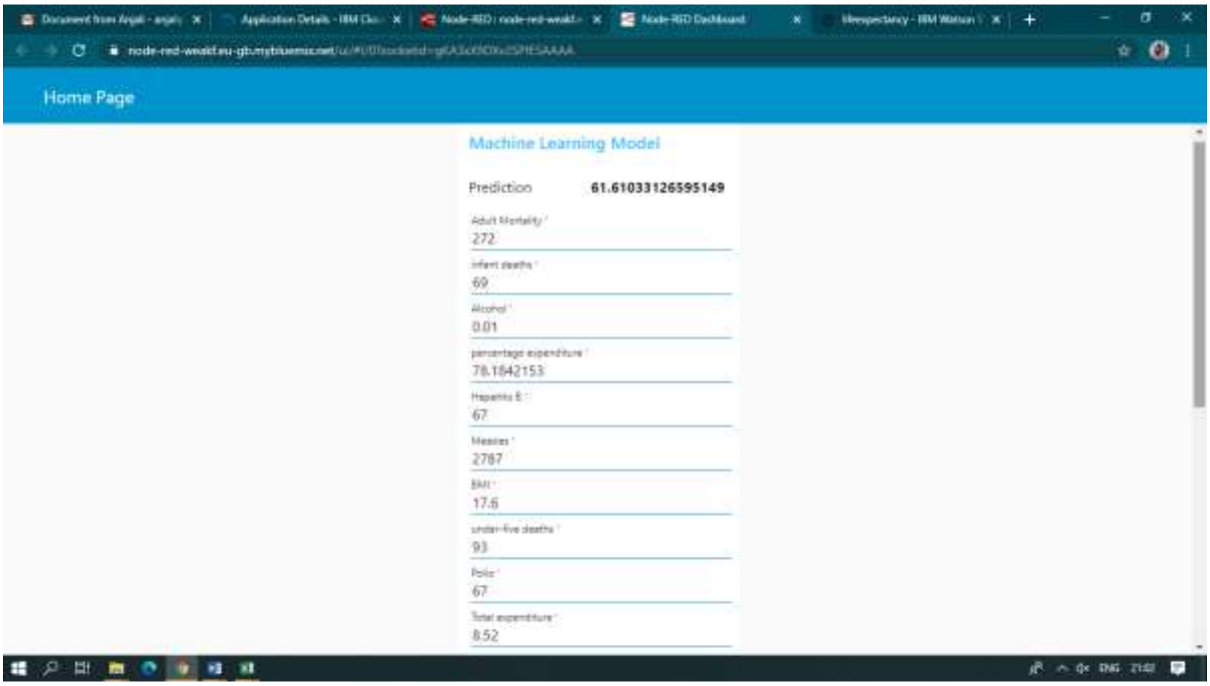


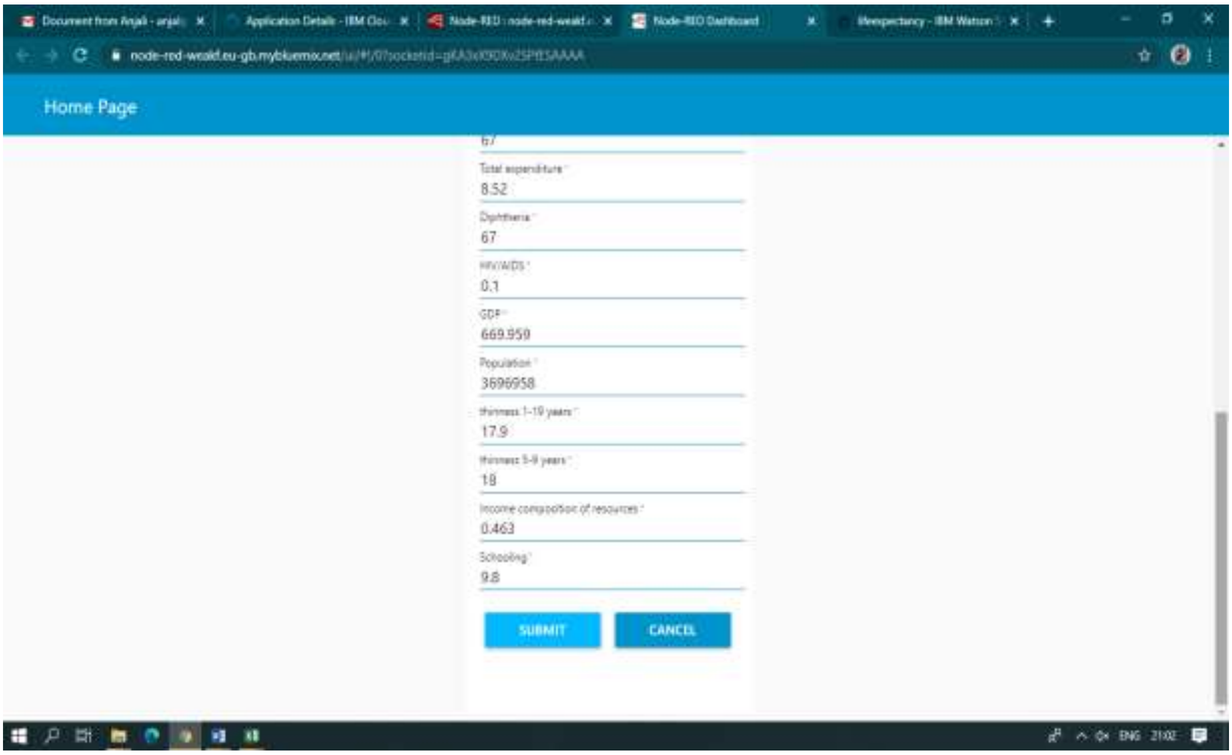

IBM Watson Studio
My projects / project / lifeexpectancy
In [95]: sns.distplot((y_test-predictions),bins=50);
Life expectancy

5. Flowchart


User enters input values
Value taken via Node-Red
Valid input?
Yes
No
Node-Red acquires token to access ML service
Throw Error
Input values are passed to the ML model
Predict using a trained model
Pass values to Node-Red
Display output to User

6. Result


Home Page
Machine Learning Model
Prediction
61.61033126595149
272
69
0.01
78.1842153
67
2787
17.6
93
67
8.52


Home Page
Total expenditure
8.52
Diphtheria
67
HIV/AIDS
0.1
GDP
669.959
Population
3696958
thinness 1-19 years
17.9
thinness 5-9 years
18
Income composition of resources
0.463
Schooling
9.8
SUBMIT
CANCEL

7. 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.

8. 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.

3) 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.

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

9. 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.

10. 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.

11. 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://www.ibm.com/watson/products-services
- https://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa69-9fab-40ee-8457-ea0e5df8c8de

12. Appendix

12.1. Source Code

Services Used:

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

Python Notebook:

```
import types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0
# 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_data_0 = pd.read_csv(body)
df_data_0.head()
df_data_0.info()
df_data_0.columns
REMOVE MISSING VALUES
import numpy as np
df_data_0['Adult Mortality'] = df_data_0['Adult
Mortality'].replace(np.nan,df_data_0['Adult Mortality'].mean())
df_data_0['Alcohol'] =
df_data_0['Alcohol'].replace(np.nan,df_data_0['Alcohol'].mean())
df_data_0['Hepatitis B'] = df_data_0['Hepatitis
B'].replace(np.nan,df_data_0['Hepatitis B'].mean())
df_data_0[' BMI '] = df_data_0[' BMI '].replace(np.nan,df_data_0[' BMI '].mean())
df_data_0['Polio'] = df_data_0['Polio'].replace(np.nan,df_data_0['Polio'].mean())
df_data_0['Total expenditure'] = df_data_0['Total
expenditure'].replace(np.nan,df_data_0['Total expenditure'].mean())
df_data_0['Diphtheria '] = df_data_0['Diphtheria
'].replace(np.nan,df_data_0['Diphtheria '].mean())
df_data_0['GDP'] = df_data_0['GDP'].replace(np.nan,df_data_0['GDP'].mean())
df_data_0['Population'] =
df_data_0['Population'].replace(np.nan,df_data_0['Population'].mean())
df_data_0[' thinness  1-19 years'] = df_data_0[' thinness  1-19
years'].replace(np.nan,df_data_0[' thinness  1-19 years'].mean())
```

```python
df_data_0[' thinness 5-9 years'] = df_data_0[' thinness 5-9
years'].replace(np.nan,df_data_0[' thinness 5-9 years'].mean())
df_data_0['Income composition of resources'] = df_data_0['Income composition of
resources'].replace(np.nan,df_data_0['Income composition of resources'].mean())
df_data_0['Schooling'] =
df_data_0['Schooling'].replace(np.nan,df_data_0['Schooling'].mean())
import seaborn as sns
sns.pairplot(df_data_0)
sns.distplot(df_data_0['Life expectancy '])
sns.heatmap(df_data_0.corr())
X = df_data_0[['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']]
y = df_data_0['Life expectancy ']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=101)
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train,y_train)
np.where(df_data_0.values >= np.finfo(np.float64).max)
df_data_0 = df_data_0.replace([np.inf, -np.inf], np.nan)
print(lm.intercept_)
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
predictions = lm.predict(X_test)
from matplotlib import pyplot as plt
plt.scatter(y_test,predictions)
sns.distplot((y_test-predictions),bins=50);
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```