

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

By: Neha Ashok Jha

INDEX

Sr, No	Title	Pg.No
1	Introduction	3
2	Literature Survey	5
3	Theoretical Analysis	6
4	Experimental Investigations	9
5	Flowchart	11
6	Result	12
7	Advantages and Disadvantages	13
8	Applications	14
9	Conclusion	15
10	Future Scope	16
11	Bibliography	17
12	Source Code	18

1. Introduction

1.1.Overview

A Typical Regression **Machine Learning** project leverages historical data to predict insights into the future. This problem statement is aimed at predicting at Life Expentancy rate of a country given various features.

Life Expentancy is a statistical measure of the average time an organism is expected to live. **Life expectancy** refers to the number of years a person is expected to live based on the statistical average. **Life expectancy** varies by geographical area and by era. Life expentancy depends upon various factors: Regional Variations, Economic Circumstances, Sex Differences, Mental illnesses, Physical Illnesses, Education, Year of their birth and other demographic factors. This problem statement provides a way to predict average life expentancy of people living in a country when various factors such as year, GDP, Education, alcohol intake of people, expenditure on healthcare system and some specific disease related deaths that happened in the country are given.

The scope of this project is to predict the guesstimate given by my current knowledge and the limited amount of time I have spent researching and thinking about this question, is that there is a 15% chance that life expectancy will decline in the future. 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

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.

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

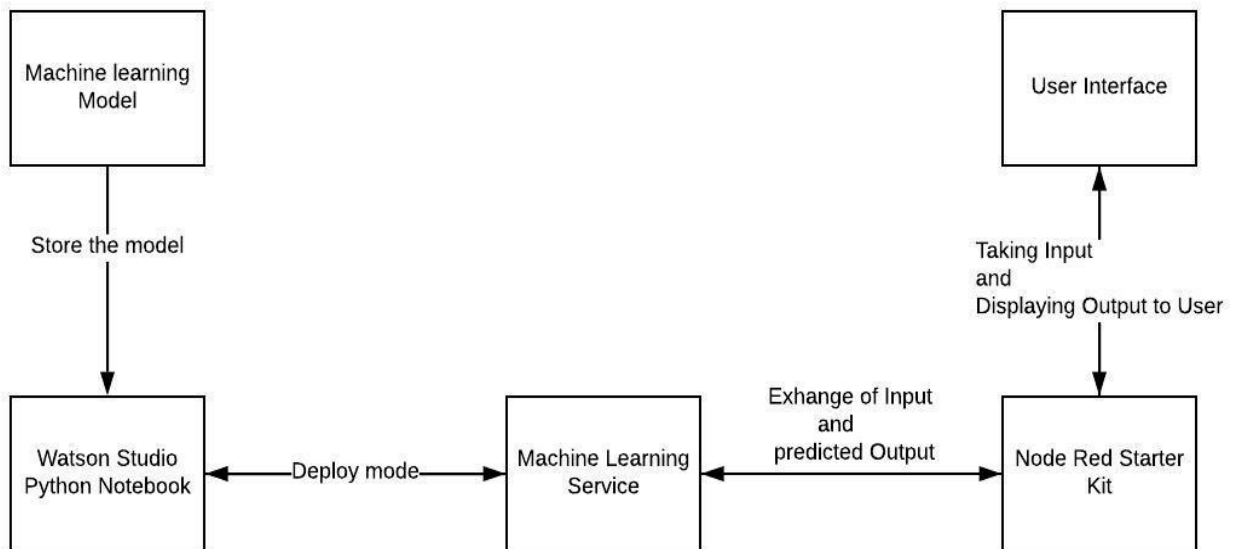
There has been an explosion of breakthroughs in the field of Machine Learning over the past few years. Machine Learning algorithms are capable of a lot and can-do wonders for the healthcare sector.

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

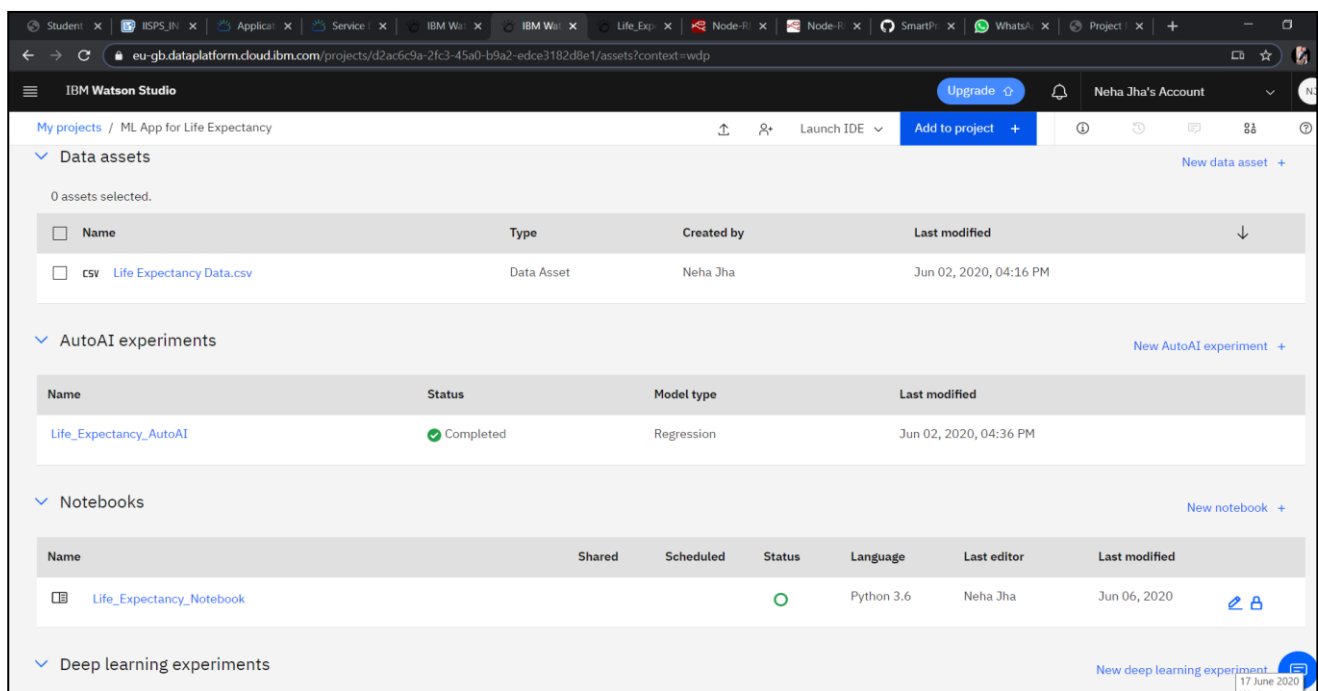
3.1. Block Diagram



3.2. Hardware/ Software Designing

Model Designing (Watson Studio) :

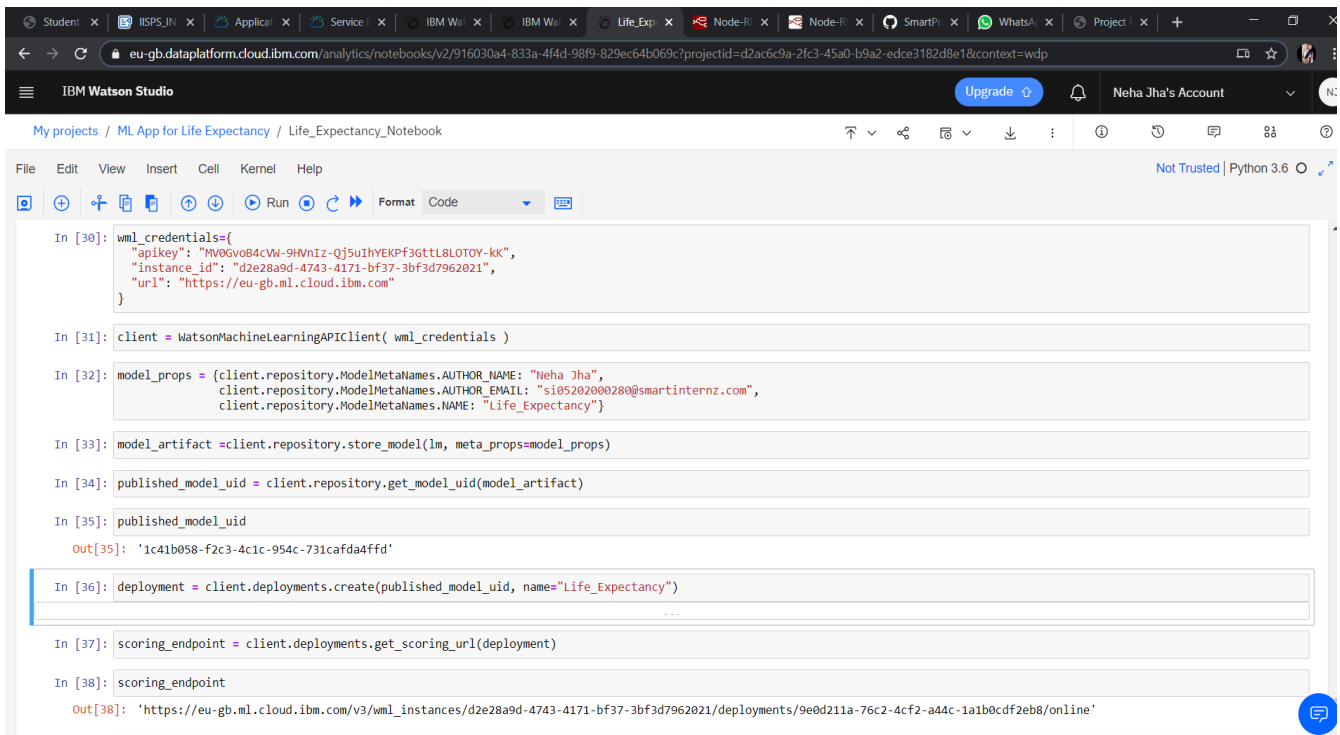
Steps: New Project => Create an empty Project => Give project name => Click Create => Add to Project => Notebook



Scoring Endpoint:

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

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



```
In [30]: wml_credentials={
        "apikey": "MV0GvoB4cVW-9HvNiz-Qj5uIhYEKPF3GttL8LOTOY-kk",
        "instance_id": "d2e28a9d-4743-4171-bf37-3bf3d7962021",
        "url": "https://eu-gb.ml.cloud.ibm.com"
    }

In [31]: client = WatsonMachineLearningAPIClient( wml_credentials )

In [32]: model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Neha Jha",
                        client.repository.ModelMetaNames.AUTHOR_EMAIL: "si05202000280@smartinternz.com",
                        client.repository.ModelMetaNames.NAME: "Life_Expectancy"}

In [33]: model_artifact = client.repository.store_model(lm, meta_props=model_props)

In [34]: published_model_uid = client.repository.get_model_uid(model_artifact)

In [35]: published_model_uid
Out[35]: '1c41b058-f2c3-4c1c-954c-731cafd44ffd'

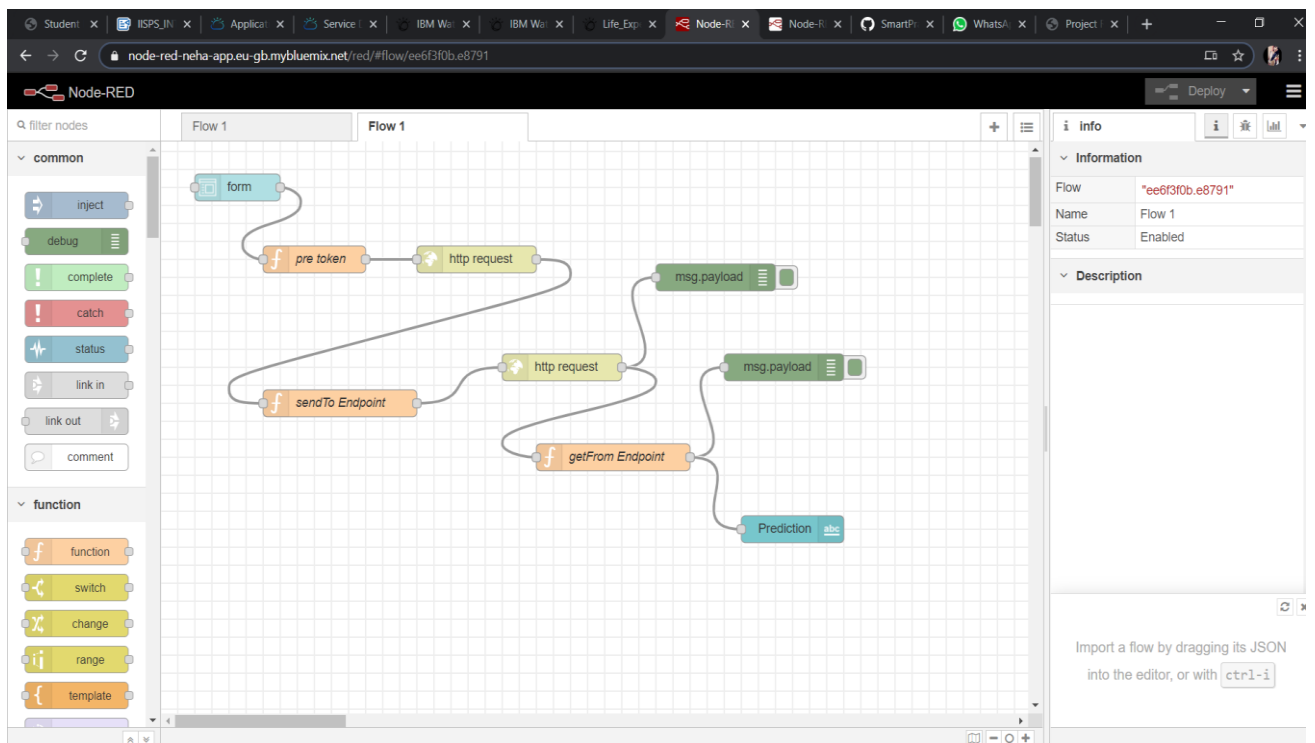
In [36]: deployment = client.deployments.create(published_model_uid, name="Life_Expectancy")
        ...

In [37]: scoring_endpoint = client.deployments.get_scoring_url(deployment)

In [38]: scoring_endpoint
Out[38]: 'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/d2e28a9d-4743-4171-bf37-3bf3d7962021/deployments/9e0d211a-76c2-4cf2-a44c-1a1b0cdf2eb8/online'
```

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 interface showing the 'Edit form node' dialog. The dialog has tabs for 'Properties' and 'Form elements'. The 'Properties' tab is active, showing fields for Group, Size, Label, and Form elements. The 'Form elements' table lists five elements: Adult Mortality, Infant Deaths, Alcohol, Percentage Expen, and Hepatitis B. Each element has a Name, Type (Number), Required status, and a Remove button. The 'Buttons' section shows 'submit' and 'cancel' buttons. The 'Info' sidebar on the right shows the node name '395a6fc7.91552' and type 'ui_form'.

Label	Name	Type	Required	Rows	Remove
Adult Mortality	d	Number	<input checked="" type="checkbox"/>		
Infant Deaths	e	Number	<input checked="" type="checkbox"/>		
Alcohol	f	Number	<input checked="" type="checkbox"/>		
Percentage Expen	g	Number	<input checked="" type="checkbox"/>		
Hepatitis B	h	Number	<input checked="" type="checkbox"/>		

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.

IBM Watson Studio interface showing a Jupyter Notebook with Python code and a data preview.

```
import pandas as pd
from boto3.client import Config
import boto3

def __iter__(self): return 0

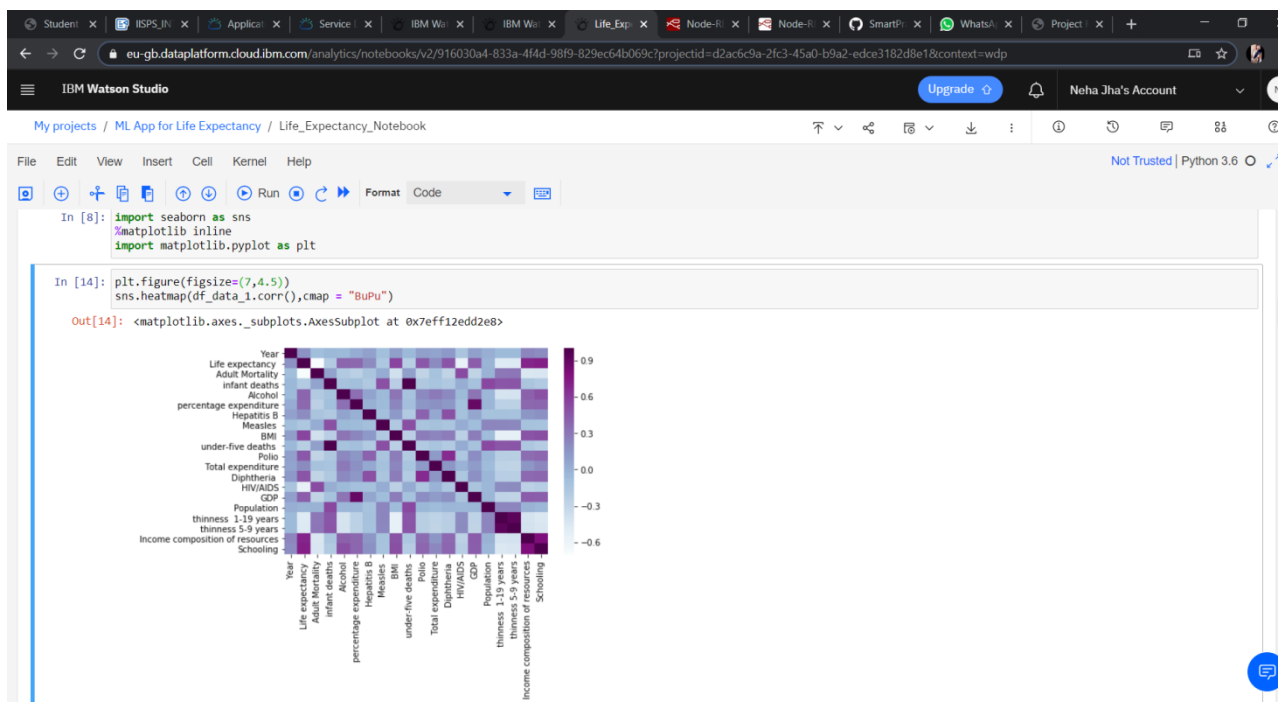
# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client = boto3.client(service_name='s3',
                      aws_access_key_id='o62683_CGBUJ44u0P85uIyt9dX1f02h0726fku-hwz2',
                      aws_secret_access_key='...',
                      endpoint_url='https://iam.cloud.ibm.com/oidc/token',
                      config=Config(signature_version='oauth'),
                      endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

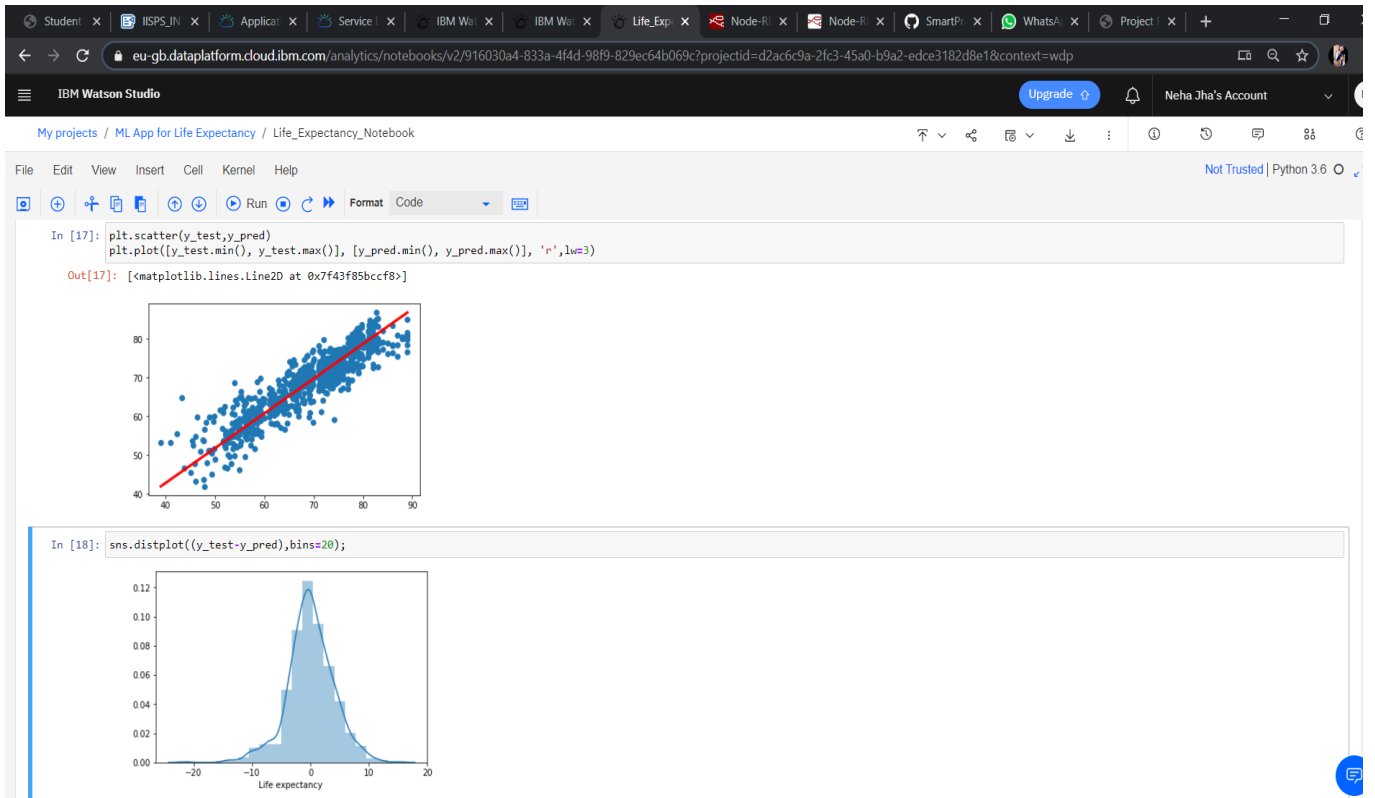
body = client.get_object(Bucket='mlappforlifeexpectancy-donotdelete-pr-jh2jedizy1a41l', Key='Life Expectancy Data.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(__iter__, body)

df_data_1 = pd.read_csv(body)
df_data_1.head()
```

Out[1]:

	Country	Year	Status	Life expectancy	Adult Mortality	Infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 1-19 years	thinness 5-9 years	Income composition of resources	Scho
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.259210	33736494.0	17.2	17.3	0.479	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.696514	327582.0	17.5	17.5	0.476	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.744976	31731688.0	17.7	17.7	0.470	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.959000	3696958.0	17.9	18.0	0.463	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.537231	2978599.0	18.2	18.2	0.454	





IBM Watson Studio

My projects / ML App for Life Expectancy / Life_Expectancy_Notebook

File Edit View Insert Cell Kernel Help

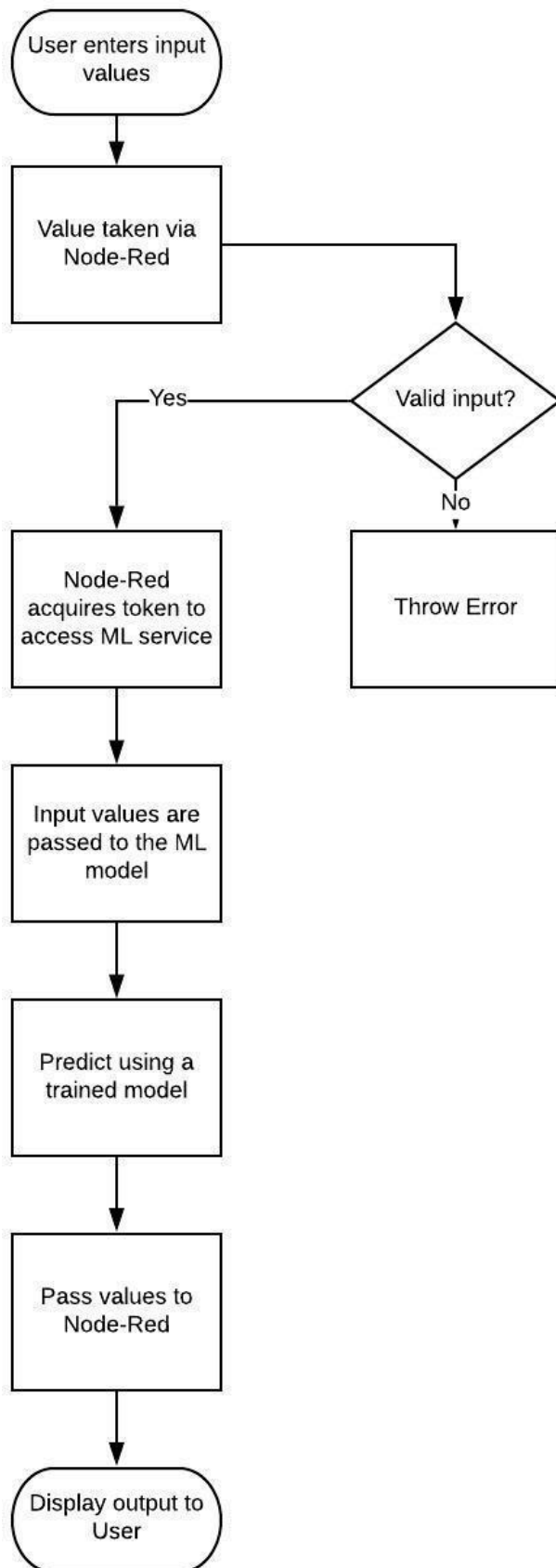
Not Trusted | Python 3.6

```
In [19]: df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df1 = df.head(25)
df1
```

Out[19]:

	Actual	Predicted
867	77.3	77.178044
1780	66.6	66.041224
621	52.6	53.235924
2715	61.5	66.964187
2717	60.0	59.700718
2855	69.3	60.978915
1410	71.1	71.707676
933	81.5	79.999151
2572	74.3	70.961675
1376	51.9	46.943382
1554	65.5	68.222745
503	87.0	81.026917
672	76.9	72.686196
22	76.1	74.150540
298	65.0	60.448431
1588	74.6	71.958912
2718	59.3	60.411205
338	64.2	66.820289
1311	73.0	70.533410
1540	81.4	81.727123
1615	71.8	69.126079
1724	65.9	74.630422
30	73.6	73.688917
2518	79.9	80.147671
2541	71.7	67.231873

5. Flowchart



6. Result

Student x IISPS_JN x Applicat x Service x IBM Wai x IBM Wai x Life_Exp x Node-R x Node-R x SmartPr x WhatsApp x Project x

node-red-neha-app.eu-gb.mybluemix.net/ui/#/0?socketId=GV1jgCZQhkuH5AAgAAMc

Welcome

Machine Learning Model ▲

Prediction

60.55696618179673

Adult Mortality *

263

Infant Deaths *

62

Alcohol *

0.01

Percentage Expenditure *

71.27962

Hepatitis B *

65

Measles *

1154

BMI *

19.1

under-five deaths *

83

Polio *

6

Total Expenditure *

8.16

Diphtheria *

65

HIV/AIDS *

0.1

GDP *

584.2592

Population *

33736494

Thinness 1-19 years *

17.2

Thinness 5-9 years *

17.3

Income composition of resources *

0.479

Schooling *

10.1

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 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()

HANDLE MISSING VALUES

df_data_1.isnull().sum()
df_data_1.fillna(df_data_1.mean(), inplace=True)
df_data_1.isnull().sum()

import seaborn as sns

%matplotlib inline

import matplotlib.pyplot as plt
plt.figure(figsize=(7,4.5))

sns.heatmap(df_data_1.corr(),cmap = "BuPu")

from pandas.plotting import scatter_matrix
axes=scatter_matrix(df_data_1, alpha=0.2, figsize=(40,40),diagonal='kde')
```

```
sns.distplot(df_data_1['Life expectancy '])
```

```
df_data_1.shape
```

```
df_data_1.info()
```

```
X=df_data_1.iloc[:,4:22]
```

```
y=df_data_1.iloc[:,3]
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)
```

```
from sklearn.linear_model import LinearRegression
```

```
lm = LinearRegression()
```

```
lm.fit(X_train,y_train)
```

```
print(lm.intercept_)
```

```
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
```

```
coeff_df
```

```
y_pred = lm.predict(X_test)
```

```
plt.scatter(y_test,y_pred)
```

```
plt.plot([y_test.min(), y_test.max()], [y_pred.min(), y_pred.max()], 'r',lw=3)
```

```
sns.distplot((y_test-y_pred),bins=20);
```

```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
```

```
df1 = df.head(25)
```

```
df1
```

```
df1.plot(kind='bar',figsize=(16,10))
```

```
plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')
```

```
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')
```

```
plt.show()
```

```
import statsmodels.api as sm
```

```
X = sm.add_constant(X)
```

```
est = sm.OLS(y, X).fit()
```

```
est.summary()
```

```
from sklearn import metrics
```

```
import numpy as np
```

```
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
```

```
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
```

```
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```