

**REPORT**

**PREDICTING LIFE EXPECTANCY USING  
MACHINE LEARNING**

**DEEKSHYA DASH**

## **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.

This problem statement provides a way to predict average life expectancy of people living in a country when various factors 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 in a dataset.

A Supervised Machine learning Regression algorithm with maximum accuracy which is trained and tested on the dataset works as the base model. The Project requires in depth knowledge of IBM Services. The User Interface is built on Node-Red which is an IBM Application and the backend uses Machine Learning Algorithm which is a typical Regression Model.

**Software Requirements:** IBM Cloud ,IBM Watson Studio, Node-red .

**Project delivers** a user interface which works on machine learning to predict the life expectancy by taking an input dataset consisting of all the various attributes that affect the model and observations to train it and provide accurate results.

### **1.2. PURPOSE**

The purpose is to provide a way to predict average life expectancy of people living in a country when various factors 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. EXISTING PROBLEM:**

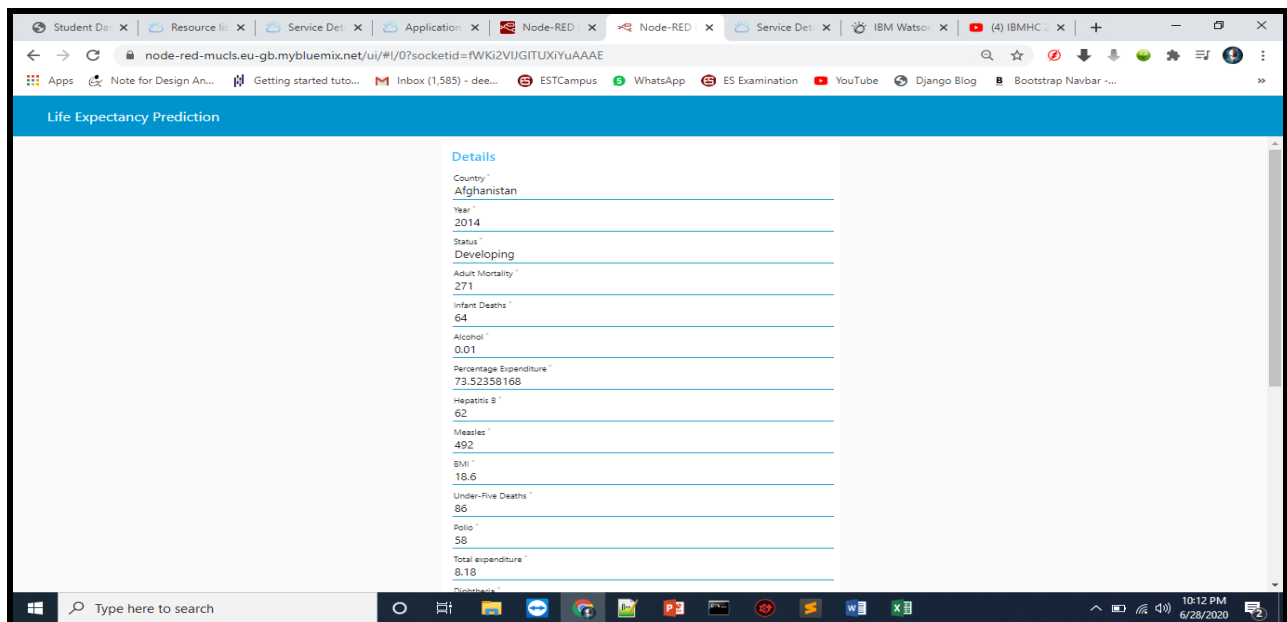
This problem statement is aimed at predicting Life Expectancy rate of a country given various features that can help determine it in the best way possible.

### **2.2. PROPOSED SOLUTION:**

The solution encourages the creation of a regression model which, by taking various features from the dataset, into consideration gives the best possible algorithm to predict the Life Expectancy accurately.

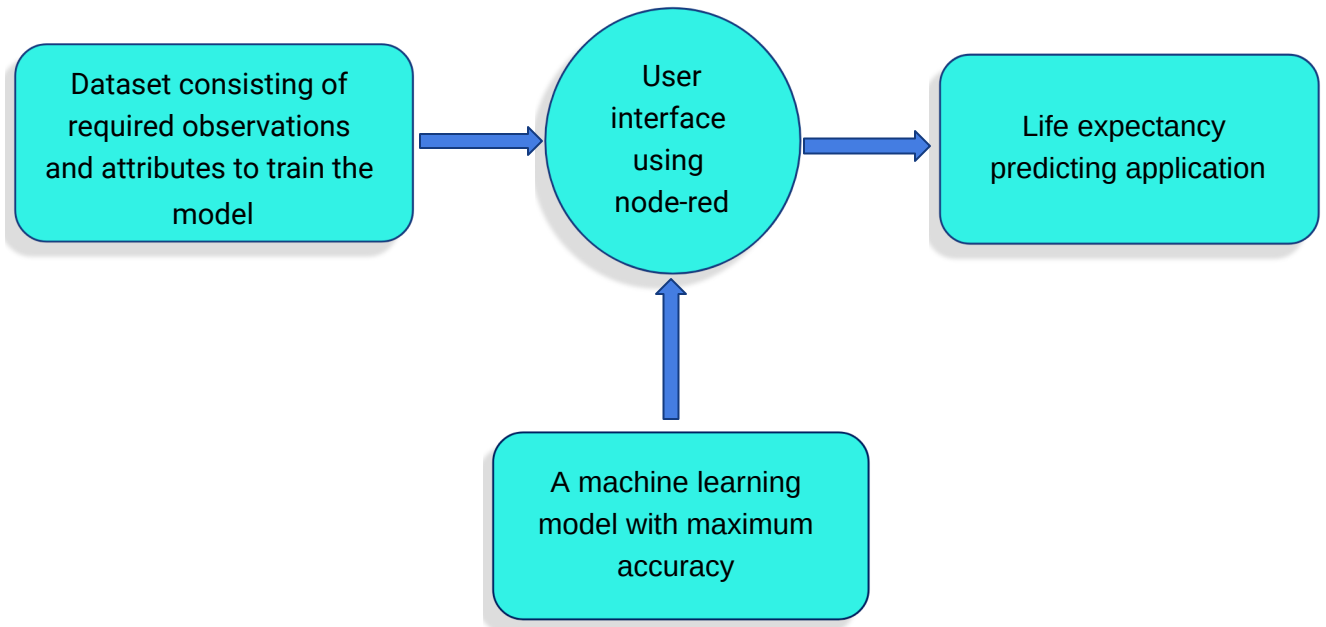
The solution is predicted by forming models using both python and Auto AI, made with the help of various Watson Services available on the IBM Cloud platform. After proper training, testing and deployment of the models, a User Interface is made on node-red which takes in the input , uses the proposed model in the backend and provides the output there itself.

## **THE UI**



### **3. THEORETICAL ANALYSIS:**

#### **3.1. BLOCK DIAGRAM:**



#### **3.2. HARDWARE/SOFTWARE DESIGNING:**

Steps required for software designing:

- Exploration of Cloud Platform
- Exploration of Watson Services
- Building a Machine Learning model on Jupyter
- Experimenting on Auto AI
- User Interface on Node-RED

## 4. EXPERIMENTAL INVESTIGATIONS:

The experiment comprised of two methods,

- with Python :
  - The Dataset was downloaded from Kaggle site.
  - A Linear Regression model helped in finding the  $R^2$  and RMSE values that determined Life Expectancy.
  - The model was trained and tested before deployment.
  - The respective values of  $R^2$  and RMSE resulted to be 0.8190807877191667 and 4.111825564792904 accurately.

### Notebook:

```
Life_Expectancy_Prediction - IBM x
eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/dadc0ffd-8be2-4676-80f5-987c6148e019?projectid=20841c9c-d8f1-46f5-bfe4-9bc0eb...
IBM Watson Studio
My projects / MyProject2 / Life_Expectancy_Prediction
File Edit View Insert Cell Kernel Help
Python 3.6
PREDICTING LIFE EXPECTANCY
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

Import Data
In [2]: import types
import pandas as pd
from botocore.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='COWRNUKF-63SL4H1EHTX911ZFWUZ_0165f52qutIjE',
    aws_secret_access_key='https://iam.cloud.ibm.com/oidc/token',
    config=Config(signature_version='s3v4'),
    endpoint_url='https://s3.eu-gb-objectstorage.service.networklayer.com')

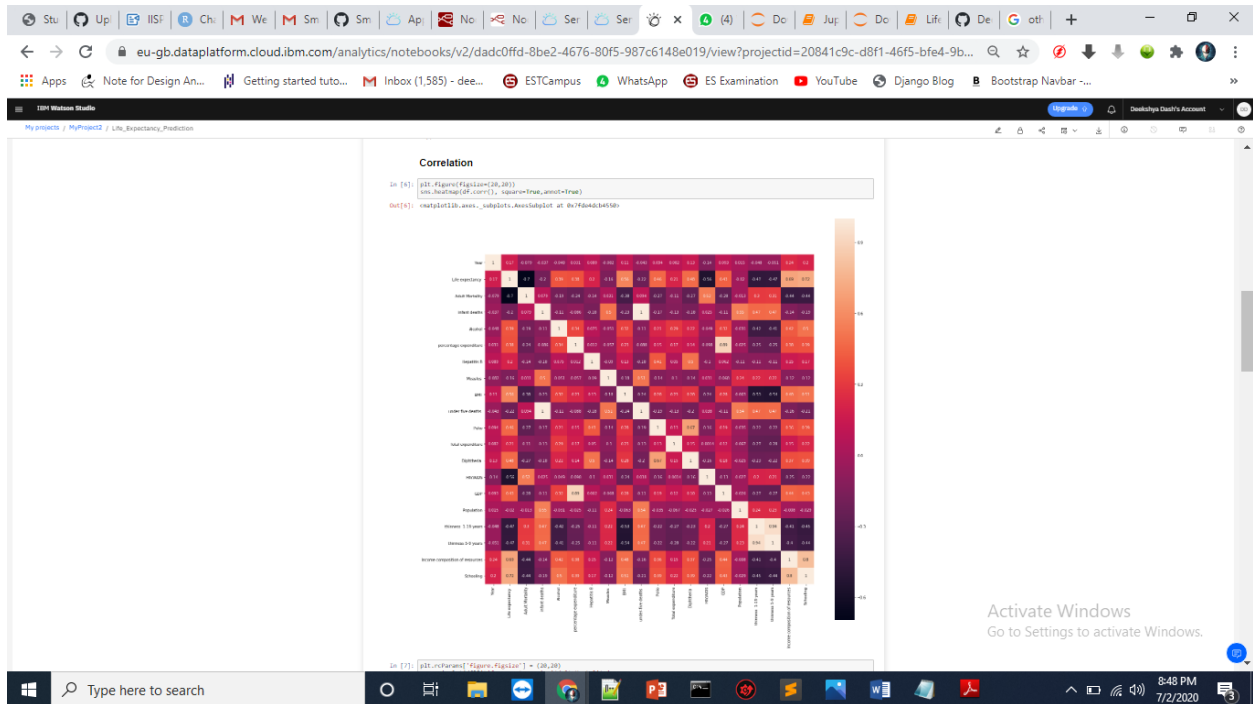
body = client.get_object(Bucket='myproject2-donotdelete-pr-svxyuentsfwop4', Key='Life Expectancy Data.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, '__iter__'): body.__iter__ = types.MethodType(__iter__, body)

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

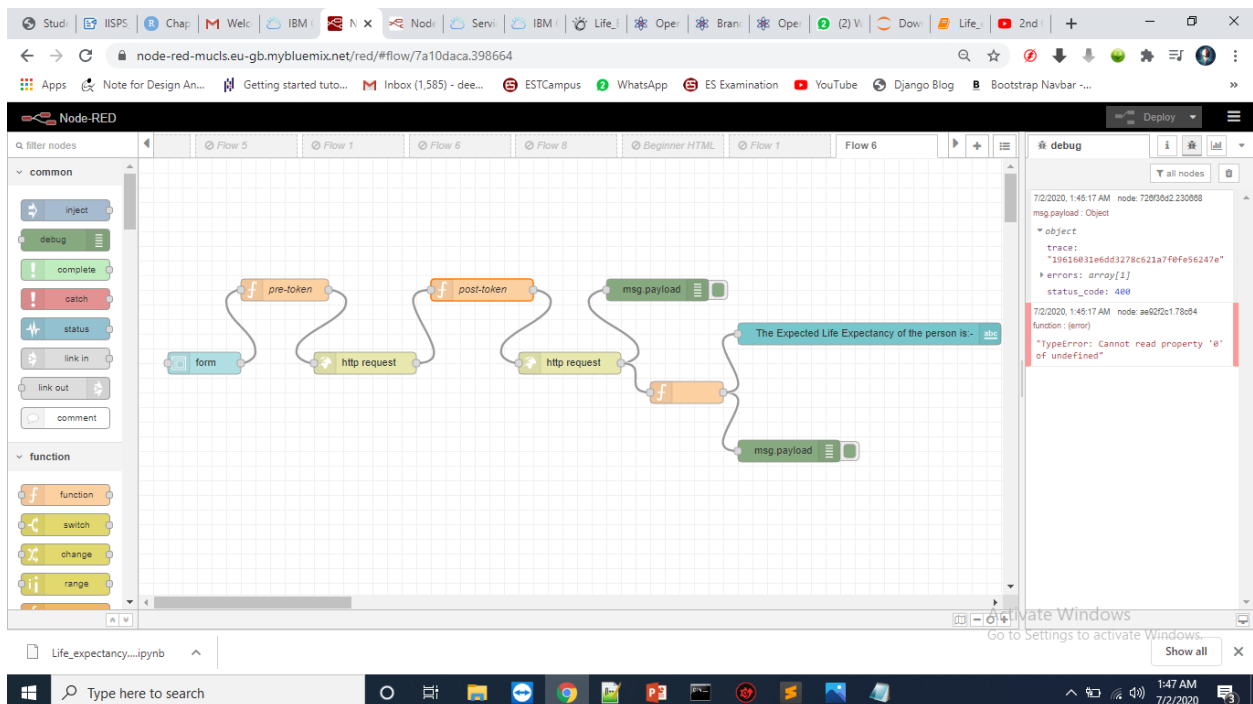
the link:

[https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/dadc0ffd-8be2-4676-80f5-987c6148e019/view?access\\_token=d74a48e8eddb164d6ec2a8a2a62477e731bcf283ef43c427f3bc2ab9a5a43d15](https://eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/dadc0ffd-8be2-4676-80f5-987c6148e019/view?access_token=d74a48e8eddb164d6ec2a8a2a62477e731bcf283ef43c427f3bc2ab9a5a43d15)

## Correlation:



## Node-RED flow:



## User-Interface:

The screenshot shows a web browser window with the URL `node-red-mucls.eu-gb.mybluemix.net/ui/#/0?socketid=7uCYDK-nJGCrhRdWAAAn`. The page title is "Life Expectancy Prediction". The form contains the following fields and values:

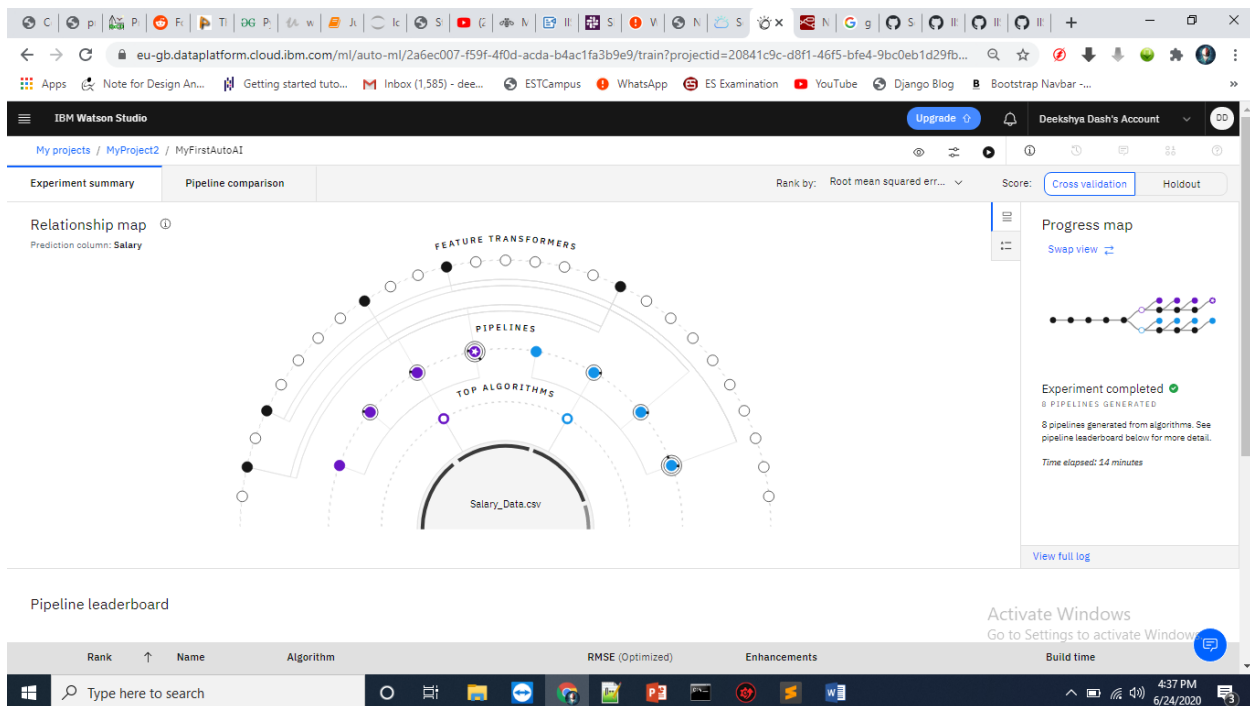
Field	Value
Polio	86
Total expenditure	58
Diphtheria	8.18
HIV/AIDS	62
GDP	0.1
Population	612.696514
thinness 1-19 years	327582
thinness 5-9 years	17.5
Income composition of resources	17.5
Schooling	0.476
	10

Below the form are two buttons: "SUBMIT" and "CANCEL". At the bottom of the form, it says: "The Expected Life Expectancy of the person is:- 62.18890196264788".

- **without Python:**

- The downloaded dataset is uploaded on the Watson Studio
- The Auto AI experiment chooses the best algorithm to predict the column that we want.
- The best model is saved after running the experiment and deployed.
- The Extra Tree Regressor best explained the model with  $R^2$  of 0.95 which means 95% of variance.

## Auto AI Experiment:



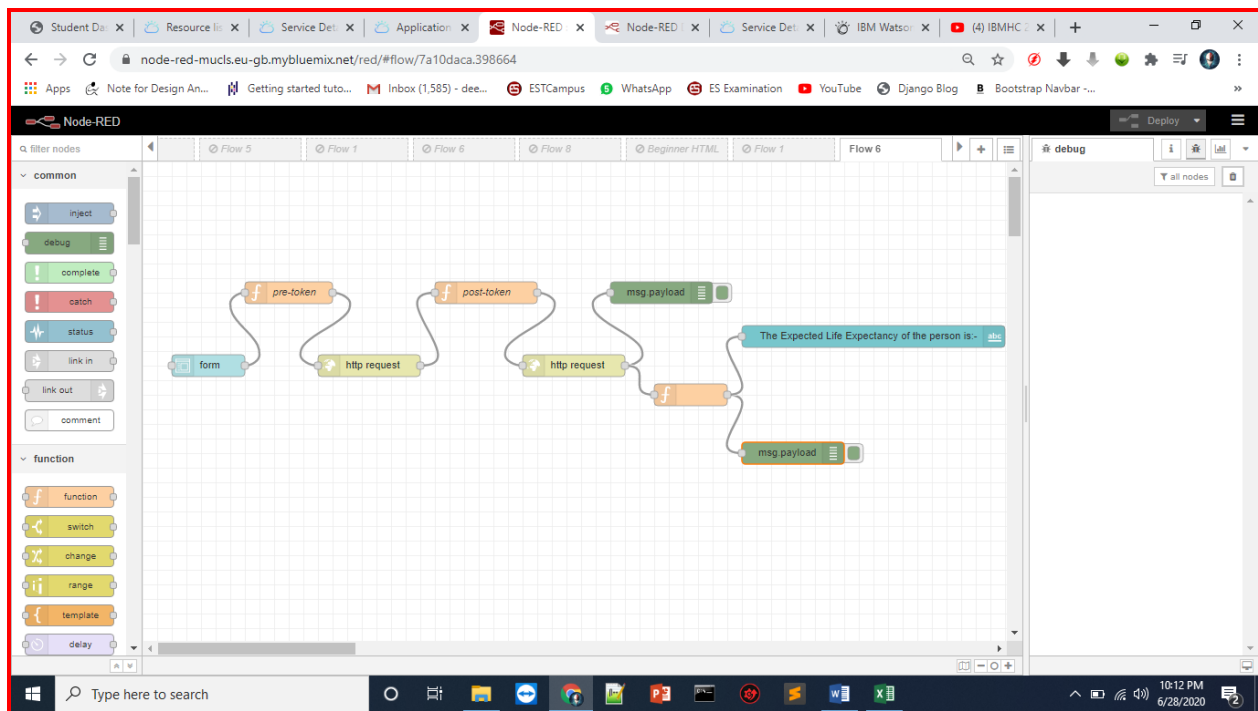
## Best chosen model:

The screenshot shows the IBM Watson Studio interface for an Auto AI experiment. The top navigation bar includes 'My projects / MyProject2 / Life Expectancy'. The main area displays a 'Pipeline leaderboard' showing a table with columns: Rank, Name, Algorithm,  $R^2$ , Enhancements, and Build time. The table lists 8 pipelines, with Pipeline 3 highlighted as the best chosen model.

Rank	Name	Algorithm	$R^2$	Enhancements	Build time
1	Pipeline 3	Extra Trees Regressor	0.956	HPO-1, FE	00:00:52
2	Pipeline 4	Extra Trees Regressor	0.956	HPO-1, FE, HPO-2	00:00:36
3	Pipeline 1	Extra Trees Regressor	0.953	None	00:00:01
4	Pipeline 2	Extra Trees Regressor	0.953	HPO-1	00:00:11
5	Pipeline 7	Decision Tree Regressor	0.918	HPO-1, FE	00:00:39
6	Pipeline 8	Decision Tree Regressor	0.918	HPO-1, FE, HPO-2	00:00:08
7	Pipeline 5	Decision Tree Regressor	0.914	None	00:00:01
8	Pipeline 6	Decision Tree Regressor	0.914	HPO-1	00:00:01



## Node-red flow to integrate Auto AI:



## User-Interface:

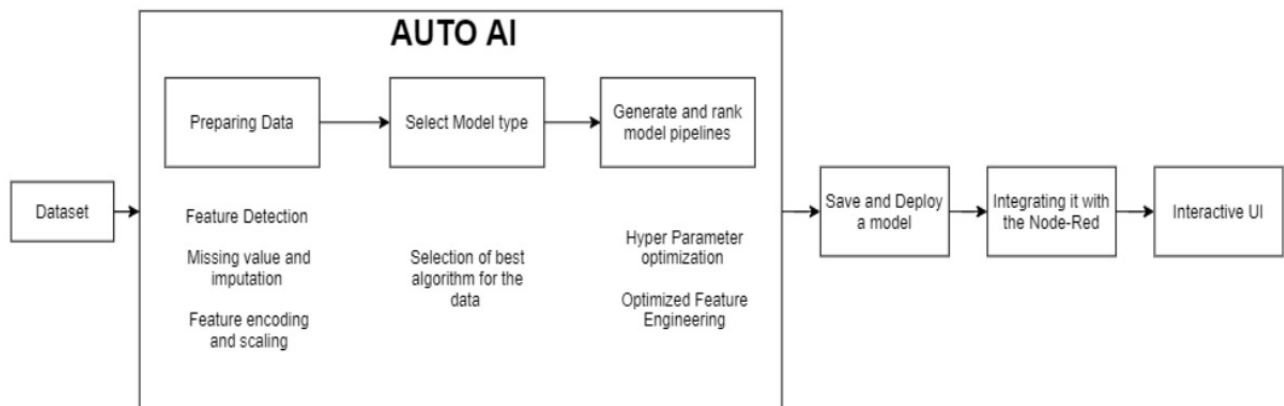
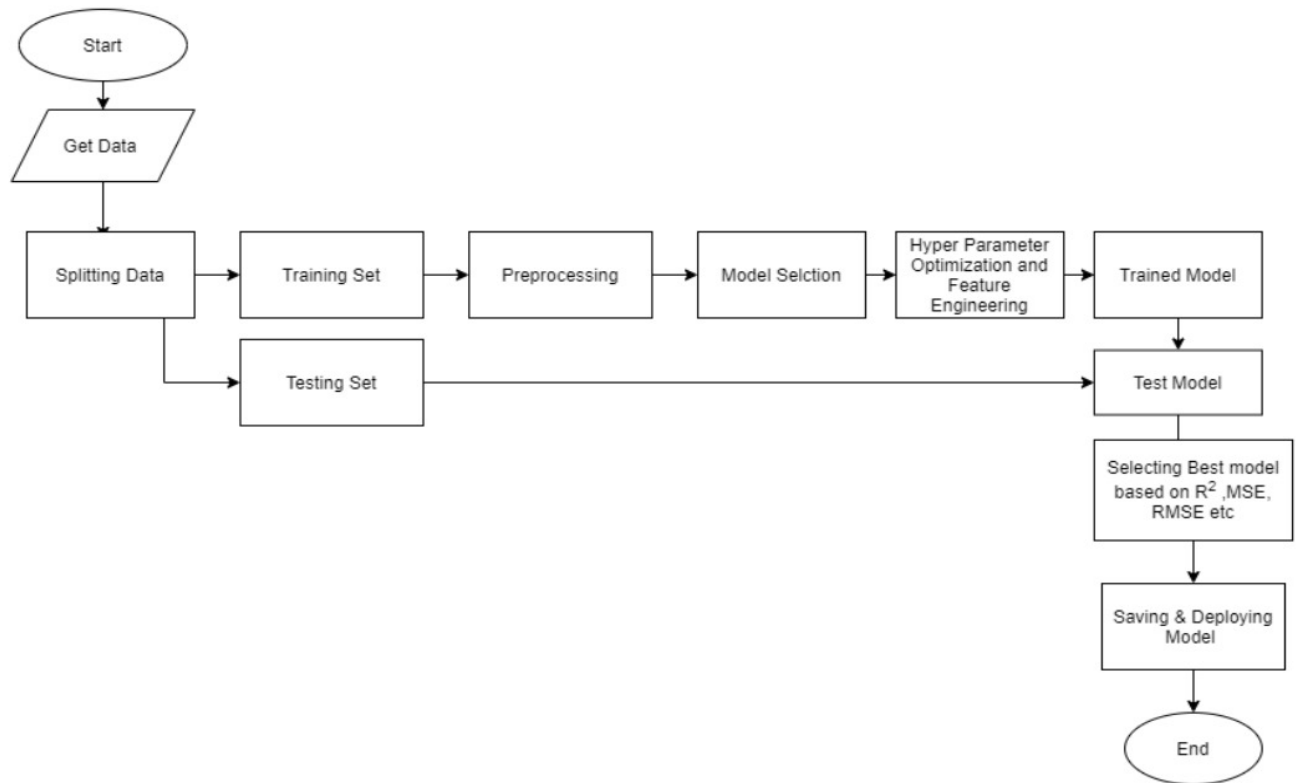
The screenshot shows a web application interface for Life Expectancy Prediction. The interface includes a table of data with columns: Country, Year, Status, Life expectancy, Adult Mortality, Infant mortality, Alcohol consumption, percentage Hepatitis, and Measles. The predicted life expectancy is shown as 59.8800144958496.

Country	Year	Status	Life expectancy	Adult Mortality	Infant mortality	Alcohol consumption	percentage Hepatitis	Measles
Afghanistan	2015	Developing	65	263	62	0.01	71.27962	65
Afghanistan	2014	Developing	59.9	271	64	0.01	73.52358	62
Afghanistan	2013	Developing	59.9	268	66	0.01	73.21924	64
Afghanistan	2012	Developing	59.5	272	69	0.01	78.18422	67
Afghanistan	2011	Developing	59.2	275	71	0.01	70.97109	68
Afghanistan	2010	Developing	58.8	279	74	0.01	79.67937	66
Afghanistan	2009	Developing	58.6	281	77	0.01	56.76222	63
Afghanistan	2008	Developing	58.1	287	80	0.03	25.87393	64
Afghanistan	2007	Developing	57.5	295	82	0.02	10.91016	63
Afghanistan	2006	Developing	57.3	295	84	0.03	17.17152	64
Afghanistan	2005	Developing	57.3	291	85	0.02	1.388648	66
Afghanistan	2004	Developing	57	293	87	0.02	15.29607	67
Afghanistan	2003	Developing	56.7	295	87	0.01	11.08905	65
Afghanistan	2002	Developing	56.2	3	88	0.01	16.88735	64
Afghanistan	2001	Developing	55.3	316	88	0.01	10.57473	63
Afghanistan	2000	Developing	54.8	321	88	0.01	10.42496	62
Albania	2015	Developing	77.8	74	0	4.6	364.9752	99
Albania	2014	Developing	77.5	8	0	4.51	428.7491	98
Albania	2013	Developing	77.2	84	0	4.76	430.877	99
Albania	2012	Developing	76.9	86	0	5.14	412.4434	99
Albania	2011	Developing	76.6	88	0	5.37	437.0621	99
Albania	2010	Developing	76.2	91	1	5.28	41.82276	99

Comparison between given value of dataset and predicted life expectancy value shown in the UI.

## 5. FLOWCHART:

### FLOWCHART OF AUTO AI



## 6. RESULT:

Data given to the User Interface gives the predicted Life Expectancy as result.

The screenshot shows a web browser window with the URL `node-red-mucls.eu-gb.mybluemix.net/ui/#/0?socketid=fWKi2VUjGITUXYuAAAE`. The page title is "Life Expectancy Prediction". On the right side, there is a "Details" section with a list of input fields and their corresponding values:

- Country: Afghanistan
- Year: 2014
- Status: Developing
- Adult Mortality: 271
- Infant Deaths: 64
- Alcohol: 0.01
- Percentage Expenditure: 73.52358168
- Hepatitis B: 62
- Measles: 492
- BMI: 18.6
- Under-Five Deaths: 86
- Polio: 58
- Total expenditure: 8.18
- Diphtheria: 62

The screenshot shows the same web browser window as above, but with the Excel data table visible on the left side. The table is titled "Life Expectancy Data - Excel (Product Activation Failed)". The table has columns for Country, Year, Status, Life expectancy, Adult Mortality, Infant Deaths, Alcohol, percentage, Hepatitis, and Measles. The data is as follows:

Country	Year	Status	Life expectancy	Adult Mortality	Infant Deaths	Alcohol	percentage	Hepatitis	Measles
Afghanistan	2015	Developing	65	263	62	0.01	71.27962	65	115
Afghanistan	2014	Developing	59.91	271	64	0.01	73.52358	62	492
Afghanistan	2013	Developing	59.9	268	66	0.01	73.21924	64	436
Afghanistan	2012	Developing	59.5	272	69	0.01	78.18422	67	278
Afghanistan	2011	Developing	59.2	275	71	0.01	7.097109	68	301
Afghanistan	2010	Developing	58.8	279	74	0.01	79.67937	66	198
Afghanistan	2009	Developing	58.6	281	77	0.01	56.76222	63	286
Afghanistan	2008	Developing	58.1	287	80	0.03	25.87393	64	159
Afghanistan	2007	Developing	57.5	295	82	0.02	10.91016	63	114
Afghanistan	2006	Developing	57.3	295	84	0.03	17.17152	64	199
Afghanistan	2005	Developing	57.3	291	85	0.02	1.388648	66	129
Afghanistan	2004	Developing	57	293	87	0.02	15.29607	67	466
Afghanistan	2003	Developing	56.7	295	87	0.01	11.08905	65	79
Afghanistan	2002	Developing	56.2	3	88	0.01	16.88735	64	248
Afghanistan	2001	Developing	55.3	316	88	0.01	10.57473	63	876
Afghanistan	2000	Developing	54.8	321	88	0.01	10.42496	62	653
Albania	2015	Developing	77.8	74	0	4.6	364.9752	99	0
Albania	2014	Developing	77.5	8	0	4.51	428.7491	98	0
Albania	2013	Developing	77.2	84	0	4.76	430.877	99	0
Albania	2012	Developing	76.9	86	0	5.14	412.4434	99	0
Albania	2011	Developing	76.6	88	0	5.37	437.0621	99	21
Albania	2010	Developing	76.2	91	1	5.28	41.82276	99	16

The right side of the screenshot shows the "Life Expectancy Prediction" form with the following values:

- Polio: 58
- Total expenditure: 8.18
- Diphtheria: 62
- HIV/AIDS: 0.1
- GDP: 612.696514
- Population: 327582
- thinness 1-19 years: 17.5
- thinness 5-9 years: 17.5
- Income composition of resources: 0.476
- Schooling: 10

At the bottom, the "Expected Life Expectancy of the person is:" is displayed as **59.88000144958496**.

## **7. ADVANTAGES AND DISADVANTAGES:**

### **ADVANTAGES:**

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

- **Wrong Prediction:** As it depends completely on user, so if user provides some wrong values then it will predict wrong value.
- **Average Prediction:** The model predicts average or approximate value with 95% accuracy but not accurate value.

## **8. APPLICATIONS:**

- a) It can be used to monitor health inequalities of a country. Used in health industries.
- 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) Advanced Features:**

The UI asks for various data which can be difficult for a normal user to gather so I have decided to do some kind of feature modification which can help them to understand each feature and provide data for the same which may appear more user friendly.

### **b) More Interactive 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://www.youtube.com/watch?v=LOCkV-mENq8&feature=youtu.be>

<https://github.com/>

<https://slack.com/intl/en-in/>

<https://cloud.ibm.com/login>

<https://www.ibm.com/cloud/get-started>

<https://github.com/watson-developer-cloud/node-red-labs>

<https://www.youtube.com/watch?v=NmdjteZQMSM>

<https://developer.ibm.com/tutorials/watson-studio-auto-ai/>

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

<https://www.youtube.com/watch?v=-CUi8GezG1I&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L&index=2>

<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=Tv\\_5DHwIYdE&list=PLjJJFiCdXMIInlWHEsgsnY3P55kGdSDh\\_&index=8&t=0s](https://www.youtube.com/watch?v=Tv_5DHwIYdE&list=PLjJJFiCdXMIInlWHEsgsnY3P55kGdSDh_&index=8&t=0s)

## **12. APPENDIX:**

Source code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df= pd.read_csv(body)
df.head()
df.isnull().sum()
```

```
#FILL NULL VALUES TO AVOID TRAIN AND TEST ERROR
df=df.fillna(df.mean())
df.isnull().sum()
```

```
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(), square=True,annot=True)
```

```
plt.rcParams['figure.figsize'] = (20,20)
sns.pairplot(df[['Life expectancy ', 'Adult Mortality',
                'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
                'Measles ']])
```

```
# deleting the non numeric values
df = df.drop(['Country','Year','Status'], axis=1)
df.head()
```

```
# labels(y) and data(X_all)
y = df['Life expectancy '].values
X_all = df.drop(['Life expectancy '], axis=1).values
```

```

# splitting the data to train and test parts
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_all, y, test_size=0.3,
random_state=42)
from sklearn.linear_model import LinearRegression
# create the model
model = LinearRegression()
# fitting the model to the train data
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
# accuracy
from sklearn.metrics import mean_squared_error
r2 = model.score(X_test, y_test)
rmse = np.sqrt(mean_squared_error(y_pred, y_test))
r2
rmse

from watson_machine_learning_client import
WatsonMachineLearningAPIClient
client = WatsonMachineLearningAPIClient(wml_credentials)
metadata = {
    client.repository.ModelMetaNames.AUTHOR_NAME : '-----',
    client.repository.ModelMetaNames.AUTHOR_EMAIL : '-----',
    client.repository.ModelMetaNames.NAME : '-----'
}
stored_data = client.repository.store_model(model, meta_props=metadata)
stored_data
guid = client.repository.get_model_uid(stored_data)
guid
deploy = client.deployments.create(guid)
scoring_endpoints = client.deployments.get_scoring_url(deploy)
scoring_endpoints

```