

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

**By
Saurav Chaudhary**

CONTENTS:

1.Introduction.....	3
1.1 Overview.....	3
1.2 Purpose.....	3
2. Literature Survey.....	4
2.1 Existing solution.....	4
2.2 Proposed solution.....	4
3. Theoretical Diagram.....	5
3.1 Block Diagram.....	5
3.2 Hardware/software Designing.....	5-8
4. Experimental Investigation.....	9
5. Flow chart.....	10
6. Result.....	11
7. Advantages and Disadvantages.....	12
7.1 Advantages.....	12
7.2 Disadvantages.....	12
8. Applications.....	12
9. Conclusion.....	12-13
10. Future scope.....	13

11. Bibliography.....	13
12. Appendix.....	13-17

1. INTRODUCTION

1.1 Overview

Life expectancy is the average number of years a person in a population could expect to live after age x. It is the life table parameter most commonly used to compare the survival experience of populations. The age most often selected to make comparisons is 0.0 (i.e., birth), although, for many substantive and policy analyses, other ages such as 65+ and 85+ are more relevant and may be used (e.g., for determining person-years of Medicare and Social Security benefit entitlement).

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

Life expectancy is perhaps the most important measure of health. Life expectancy increases due to healthcare improvements like the introduction of vaccines, the development of drugs or positive behavior changes like the reduction in smoking or drinking rates.

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.

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

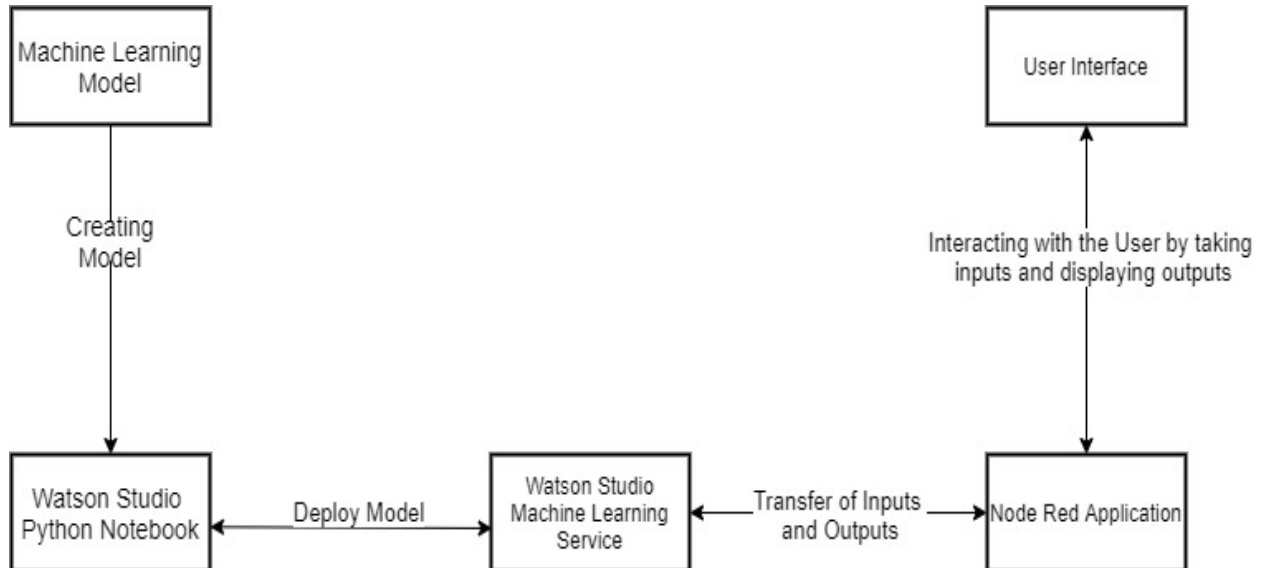
The project tries to build a model based on the given dataset. The first step was to clean the data, this included detecting and dealing with both missing values and outliers. The variables and dataset were given a general description so that a better understanding of what the variables mean could be gathered. Then both explicit and implicit missing values were detected. Implicit missing values were values that didn't make sense for a variable given the nature of the data.

Our first step in this project is DATA PRE PROCESSING, is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. It refers to the technique of preparing the raw data to make it suitable for a building and training. We will drop required columns which will not be used in Regression. Analyzing data sets to summarize their main characteristics, often with visual methods is done. We build coefficient matrix and we obtain boxplots to analyze the outliers.

We train our regression models we will need to first split up our data into an A array that contains the features to train on, and a B array with the target variable, here it is "Life Expectancy column". We split the data into a training set and a testing set. We will train our model on the training set and then use the test set to evaluate the model and the best model is chosen to evaluate the predictions.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware/Software Designing:

Project Requirements:

i) Functional Requirements:

To be able to predict the life expectancy accurately using Machine Learning models.

ii) Technical Requirements:

Any working laptop/PC with minimum 2.2Ghz processor and at least 8GB of memory with an Internet connection.

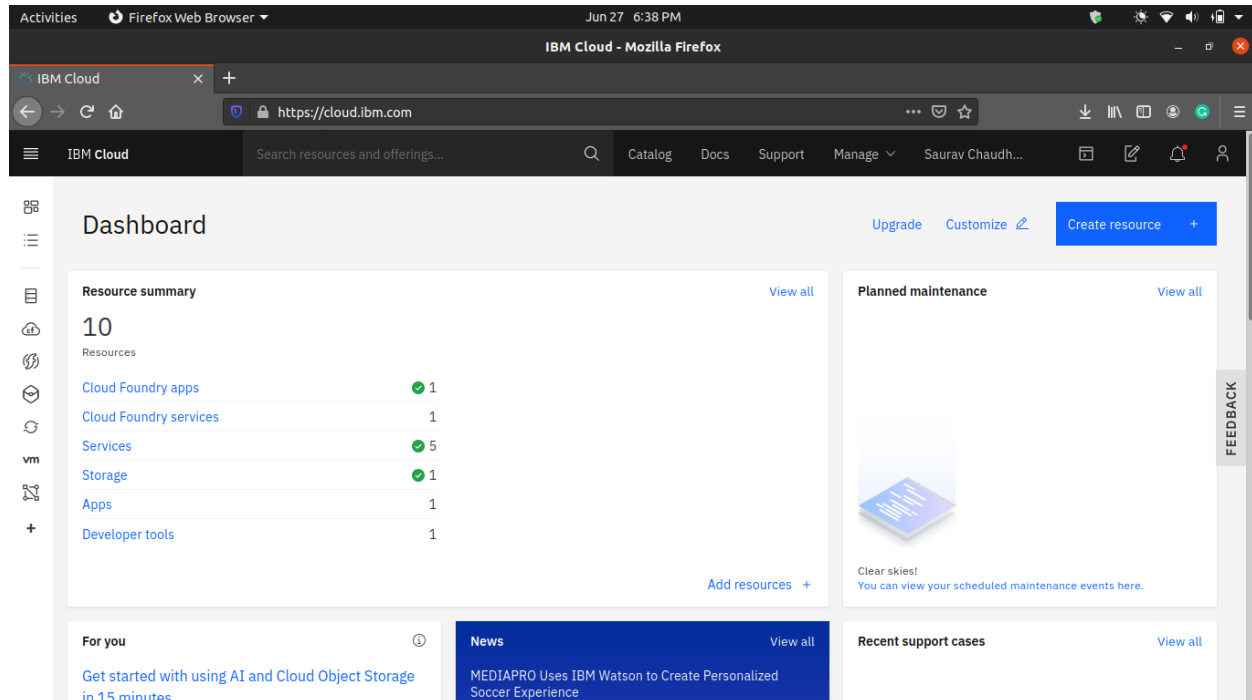
iii) Software Requirements:

- a) Python
- b) IBM Cloud

c) IBM Watson

Model Designing (Watson Studio) :

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



Activities Firefox Web Browser Jun 27 6:41 PM

IBM Watson Studio - Mozilla Firefox

IBM Watson Studio Upgrade Saurav Chaudhary's Account

Start by creating a project

A project is how you organize your resources to work with data and collaborate with team members.

Create a project

Create a project, and then add the tools and assets you need.

Search a catalog

Find the assets you need in a catalog.

Recently updated projects [View all \(2\)](#) [New project +](#)

Name	Role	Collaborators	Date created	Last updated
Life Expectancy Prediction	Admin		Jun 15, 2020	Jun 17, 2020
First ML Model	Admin		Jun 14, 2020	Jun 14, 2020

Activities Firefox Web Browser Jun 27 6:42 PM

IBM Watson Studio - Mozilla Firefox

IBM Watson Studio Upgrade Saurav Chaudhary's Account

[My projects](#) / [Life Expectancy Prediction](#) [Launch IDE](#) [Add to project +](#)

[Overview](#) [Assets](#) [Environments](#) [Jobs](#) [Deployments](#) [Access Control](#) [Settings](#)

What assets are you looking for?

▼ Data assets [New data asset +](#)

0 assets selected.

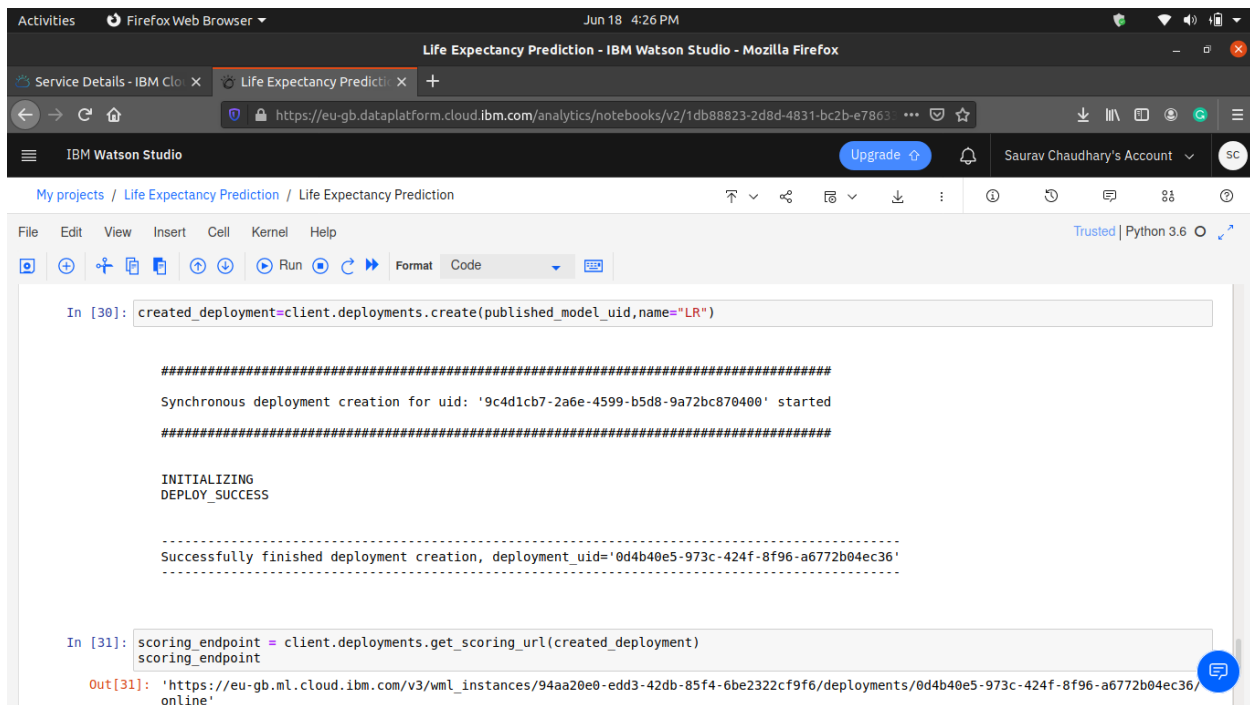
<input type="checkbox"/>	Name	Type	Created by	Last modified
<input type="checkbox"/>	CSV Life_Expectancy_Data.csv	Data Asset	Saurav Chaudhary	Jun 16, 2020, 9:44 AM

▼ AutoAI experiments [New AutoAI experiment +](#)

Name	Status	Model type	Last modified
You don't have any AutoAI experiments yet.			

This is the backend of our project. In order to connect our front end and backend we need to generate a scoring point.

Scoring point Generation:



The screenshot shows the IBM Watson Studio interface in a Firefox browser. The notebook contains the following Python code and output:

```
In [30]: created_deployment=client.deployments.create(published_model_uid,name="LR")

#####

Synchronous deployment creation for uid: '9c4d1cb7-2a6e-4599-b5d8-9a72bc870400' started
#####

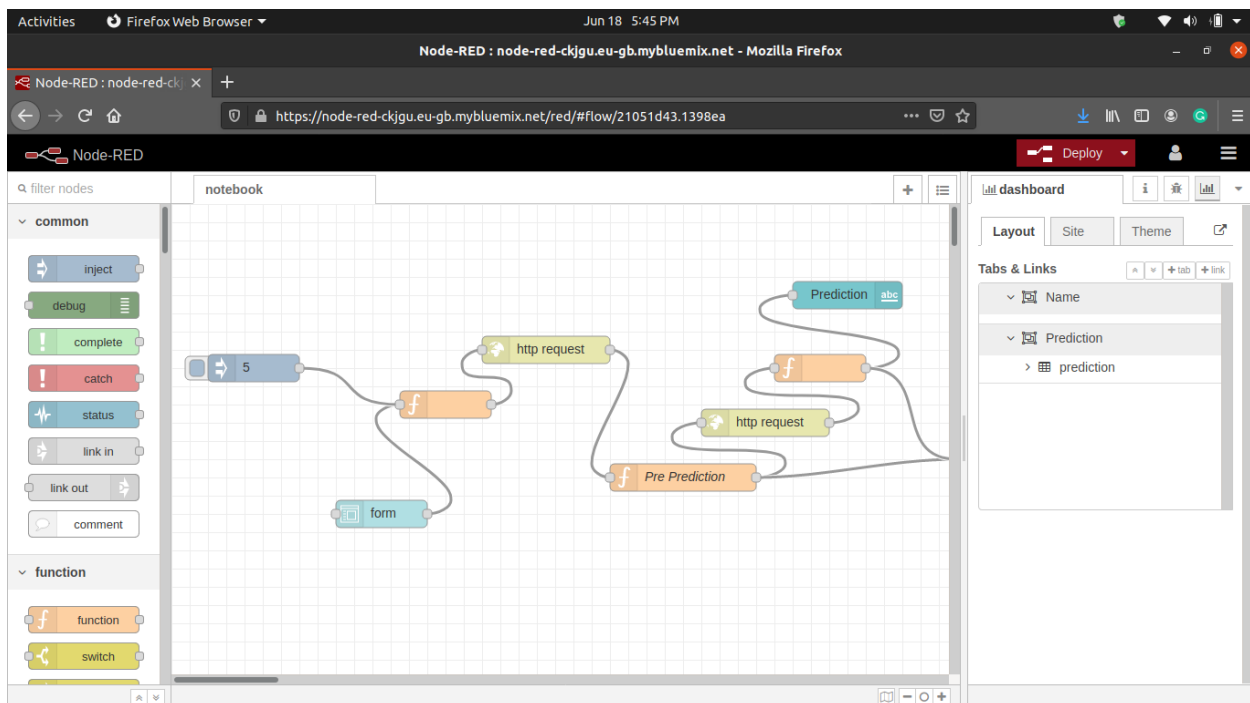
INITIALIZING
DEPLOY_SUCCESS

-----
Successfully finished deployment creation, deployment_uid='0d4b40e5-973c-424f-8f96-a6772b04ec36'
-----

In [31]: scoring_endpoint = client.deployments.get_scoring_url(created_deployment)
scoring_endpoint

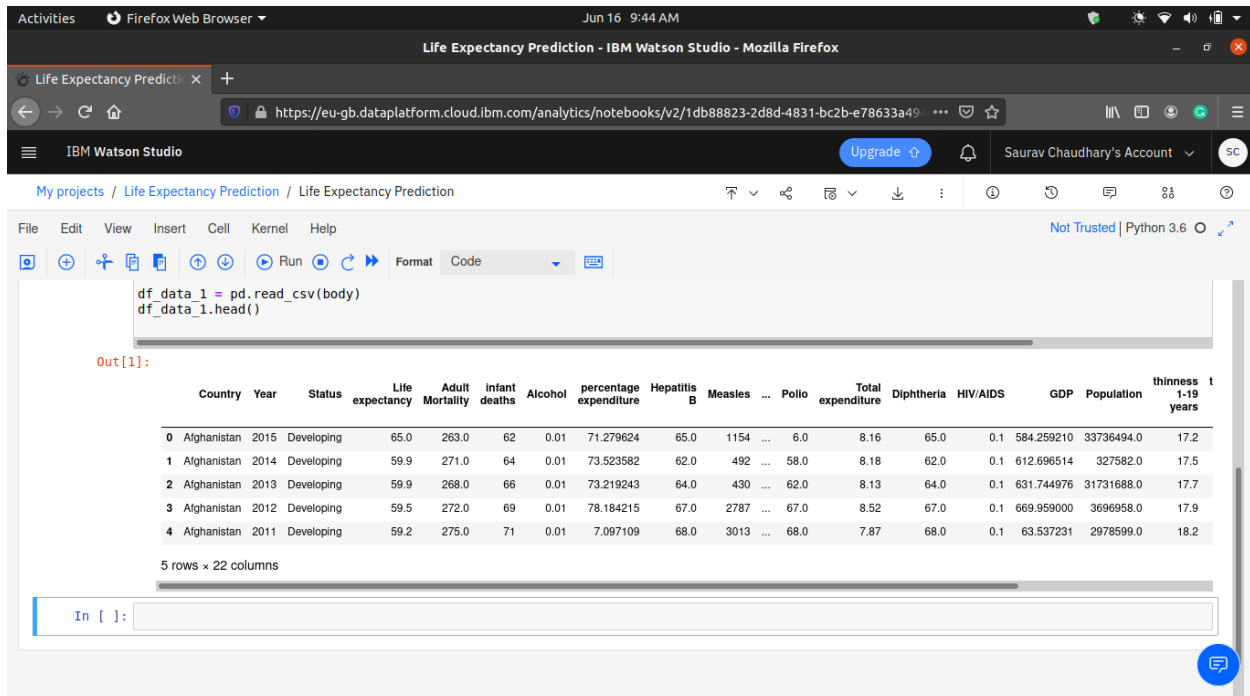
Out[31]: 'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/94aa20e0-edd3-42db-85f4-6be2322cf9f6/deployments/0d4b40e5-973c-424f-8f96-a6772b04ec36/online'
```

Nodered integration with ML model:



4. EXPERIMENTAL INVESTIGATIONS

Analyzing every feature in our dataset is very important which helps us to build a model which gives more accurate result.



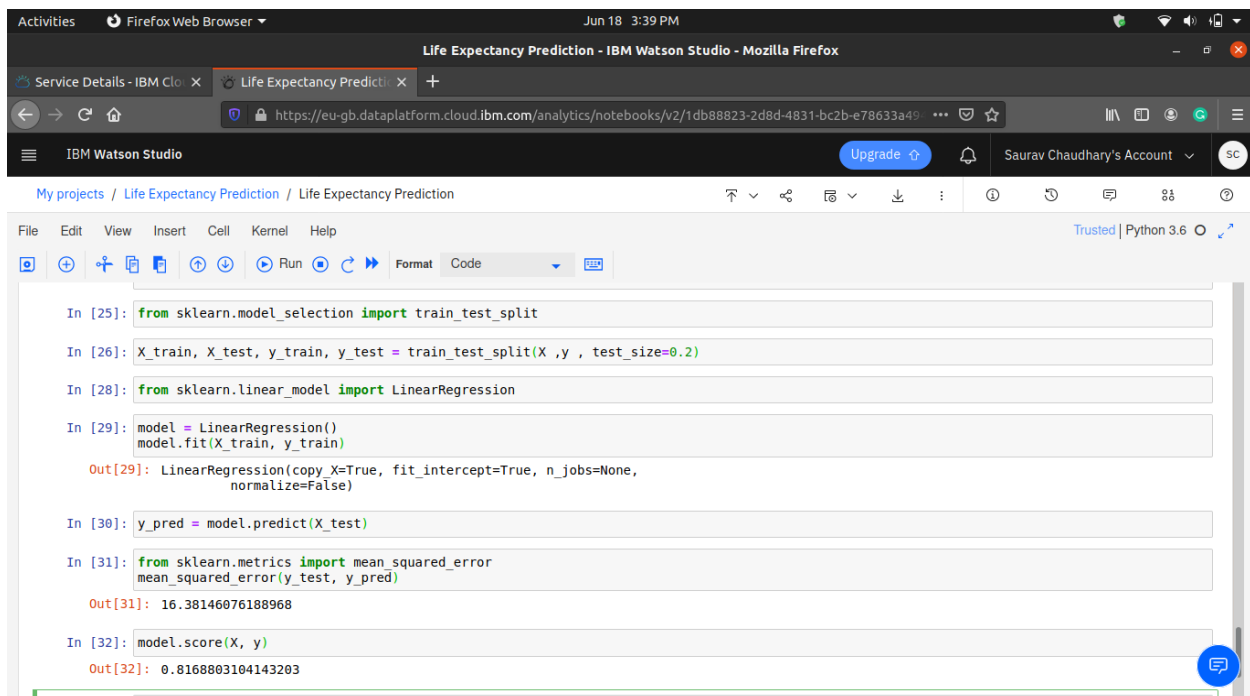
The screenshot shows the IBM Watson Studio interface with a notebook titled "Life Expectancy Prediction". The code cell contains the following Python code:

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

The output displays the first five rows of the dataset, showing 22 columns: Country, Year, Status, Life expectancy, Adult Mortality, Infant deaths, Alcohol, percentage expenditure, Hepatitis B, Measles, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP, Population, and thinness 1-19 years. The data is for Afghanistan across different years (2011-2015).

	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	
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
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
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
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
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

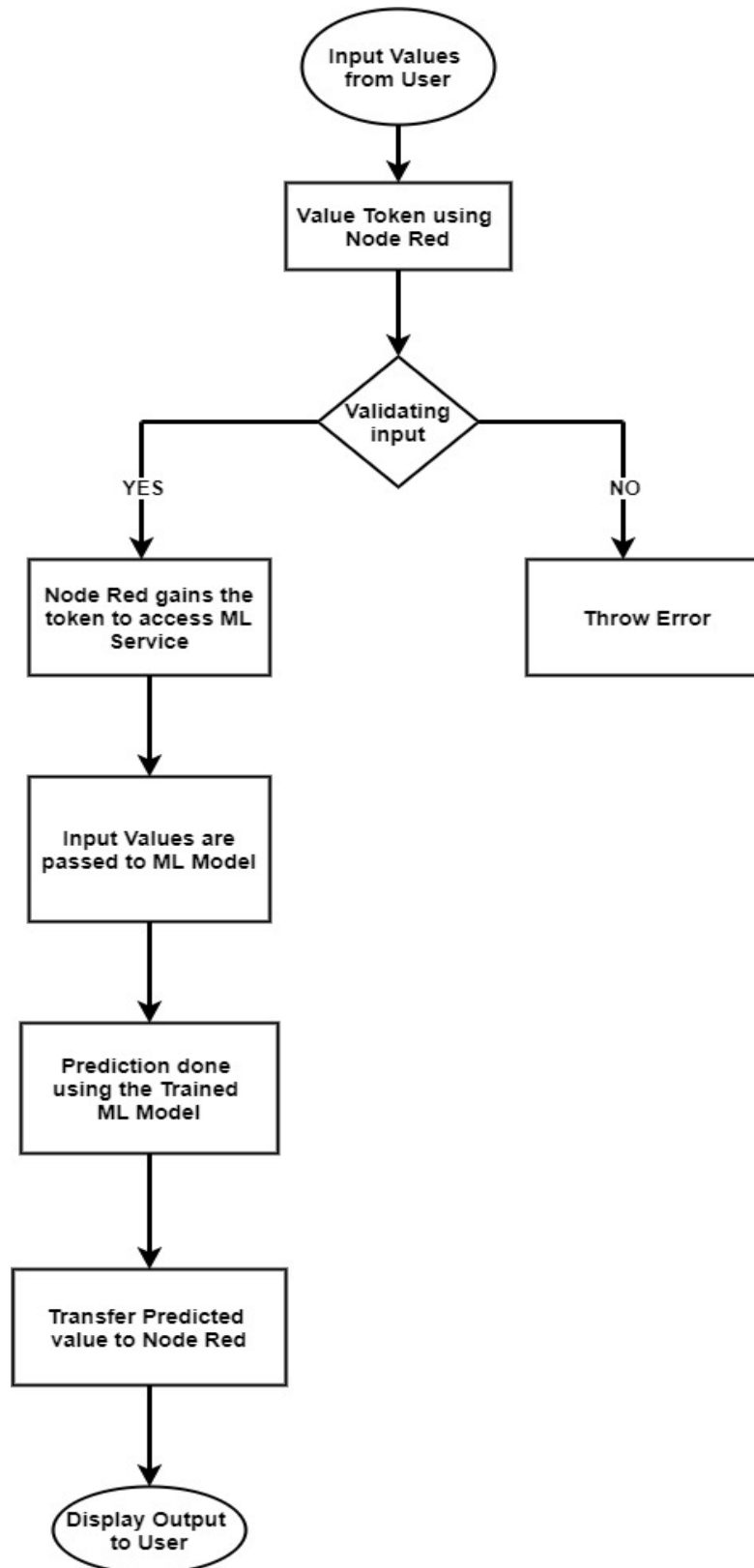
5 rows x 22 columns



The screenshot shows the same IBM Watson Studio notebook with the following code cells:

```
In [25]: from sklearn.model_selection import train_test_split
In [26]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
In [28]: from sklearn.linear_model import LinearRegression
In [29]: model = LinearRegression()
          model.fit(X_train, y_train)
Out[29]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
          normalize=False)
In [30]: y_pred = model.predict(X_test)
In [31]: from sklearn.metrics import mean_squared_error
          mean_squared_error(y_test, y_pred)
Out[31]: 16.38146076188968
In [32]: model.score(X, y)
Out[32]: 0.8168803104143203
```

5. FLOW CHART



6. RESULT

Home

Default

Prediction55.25999999999999

Adult Mortality *275.0

infant deaths *71

Alcohol *0.01

percentage expenditure *7.097109

Hepatitis B *68.0

Measles *3013

BMI68.0

under-five deaths *8

Home

Default

Prediction63.467499999999994

Year *2015

Status(0 if Deceloping 1 if Developed) *0

Adult Mortality *263

Infant Deaths *62

Alcohol *0.01

Percentage Expenditure *71.279624

Hepatitis B *65

Measles *1154

BMI *19.1

7. ADVANTAGES AND DISADVANTAGES

7.1 Advantages

Life expectancy can be estimated at any age, e.g. life expectancy at 65 years. Gives more weight to deaths at younger ages. Life expectancy has been used nationally to monitor health inequalities.

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.

7.2 Disadvantages

This model is developed using Machine Learning in which human involvement is very less and might cause some errors and if any error occurs it takes a lot of time for the developer to identify the root cause.

8. APPLICATIONS

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

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.

9. CONCLUSION

Prognostication of life expectancy is difficult for humans. Our research shows that machine learning and natural language processing techniques offer a feasible and promising approach to predicting life expectancy. The research has potential for real-life applications, such as supporting timely recognition of the right moment to start

Advance Care Planning. 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.

10. FUTURE SCOPE

One can plan to explore methods for gaining more insight in the nature of the patterns that are detected by neural networks, as well as making the determinants of a certain prediction transparent. 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#deploy-model-as-web-service>
- <https://www.ibm.com/watson/products-services>

12. APPENDIX

- **Watson Assistant:**
Watson Assistant is a conversation AI platform that helps you provide customers fast, straightforward and accurate answers to their questions, across any application, device or channel.
- **Watson Studio:**
Analysts prepare data and build models at scale across any cloud. Build models using images with IBM Watson Visual Recognition and texts with IBM Watson Natural Language Classifier. Deploy and run models through one-click integration with IBM Watson Machine Learning.

- **IBM Cloud Function:**

IBM Cloud provides a full-stack, public cloud platform with a variety of offerings in the catalog, including compute, storage, and networking options, end-to-end developer solutions for app development, testing and deployment, security management services, traditional and open-source databases

- **Node-Red:**

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.

CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import r2_score, mean_absolute_error,
mean_squared_error
from scipy import stats
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')

import types
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

client_07c87c809f76416e8866730cc6bead46 =
ibm_boto3.client(service_name='s3',
                 ibm_api_key_id='q2_LTVZubdJPuAANb4y29nkTr0w7NkcDU1xi5dZN7Ihr',
                 ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
                 config=Config(signature_version='oauth'),

endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

body =
client_07c87c809f76416e8866730cc6bead46.get_object(Bucket='lifeexpectancyp
rediction-donotdelete-pr-du84wfqfkonzkp', 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()
```

```
df.shape
```

```
df.info()
```

```
df.describe()
```

```
df.columns
```

```
df.isnull().sum()
```

```
df1 = df.iloc[:, :].values
```

```
from sklearn.impute import SimpleImputer
```

```
imputer = SimpleImputer(missing_values = np.nan, strategy = 'mean')
imputer = imputer.fit(df1[:, 3:])
df1[:, 3:] = imputer.transform(df1[:, 3:])
```

```
df1 = pd.DataFrame(df1)
df1.columns = df.columns
```

```
df1.isnull().sum()
```

```
X = df1.drop('Life expectancy ', axis = 1)
x_cols = X.columns
X = X.values
y = df1['Life expectancy ']
```

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
X[:, 0] = encoder.fit_transform(X[:, 0])
X[:, 2] = encoder.fit_transform(X[:, 2])
X = pd.DataFrame(X)
X.columns = x_cols
y = y.astype('int')
```

```
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)
print('Training Features Shape:', X_train.shape)
print('Training Labels Shape:', y_train.shape)
print('Testing Features Shape:', X_test.shape)
print('Testing Labels Shape:', y_test.shape)
```

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```

model1.fit(X_train, y_train)

pred1=model1.predict(X_test)

plt.scatter(y_test, pred1)

sns.distplot((y_test-pred1), bins=50)

from sklearn.ensemble import RandomForestRegressor
model2 = RandomForestRegressor(n_estimators = 30, random_state = 40)
model2.fit(X_train, y_train)

pred2 = model2.predict(X_test)

plt.scatter(y_test, pred2)

sns.distplot((y_test-pred2), bins=50)

print("R2 score for Linear Regression Model: ", end='')
print(r2_score(lmpredictions, y_test))
print("R2 score for RandomForest Regression Model: ", end='')
print(r2_score(rfpredictions, y_test))

print('MAE:', mean_absolute_error(y_test, pred2))
print('MSE:', mean_squared_error(y_test, pred2))
print('RMSE:', np.sqrt(mean_squared_error(y_test, pred2)))

from watson_machine_learning_client import WatsonMachineLearningAPIClient
wml_credentials={
    "apikey": "s8y8iujKIU4Yp0Gpa0GCCJeW2VYi65viMG2qV2UQhaq9",
    "iam_apikey_description": "Auto-generated for key
bfed9624-7960-433e-8246-84065fe705bd",
    "iam_apikey_name": "wdp-writer",
    "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
    "iam_serviceid_crn":
"crn:v1:bluemix:public:iam-identity::a/f69c66dba660431eac4a575ea9480d28::s
erviceid:ServiceId-c63f0953-93c3-461e-ae6b-cd2595e83b54",
    "instance_id": "94aa20e0-edd3-42db-85f4-6be2322cf9f6",
    "url": "https://eu-qb.ml.cloud.ibm.com"
}

client=WatsonMachineLearningAPIClient(wml_credentials)

metadata={
    client.repository.ModelMetaNames.DESRIPTION:'life expectancy data',
    client.repository.ModelMetaNames.AUTHOR_NAME:'Saurav',
    client.repository.ModelMetaNames.NAME:"linear",
    client.repository.ModelMetaNames.FRAMEWORK_NAME:"scikit-learn",
    client.repository.ModelMetaNames.FRAMEWORK_VERSION:"0.22"
}

```



```
}  
  
model_details=client.repository.store_model(model2,meta_props=metadata)  
  
published_model_uid=client.repository.get_model_uid(model_details)  
  
published_model_uid  
  
created_deployment=client.deployments.create(published_model_uid,name="LR"  
)  
  
scoring_endpoint = client.deployments.get_scoring_url(created_deployment)  
scoring_endpoint  
  
client.deployments.list()
```