**Project Report**

***on***

**Predicting Life Expectancy**

**using**

**Machine Learning**

***by***

**ALLURI SURAJ REDDY**

**1.Introduction..........................................................**

1.1 Overview................................................................................

1.2 Purpose..................................................................................

**2. Literature Survey .............................................**

2.1 Existing Problem.....................................................................

2.2 Proposed Solution..................................................................

**3. Theoretical Analysis............................................**

3.1 Block Diagram.........................................................................

3.2 Hardware / Software Designing**..............................................**

**4. Experimental Investigations..............................**

**5. Flowchart ..........................................................**

**6. Result ...................................................................**

**7. Advantages & Disadvantages............................**

**8. Applications........................................................**

**9. Conclusion............................................................**

**10. Future Scope.....................................................**

**11. Bibliography ....................................................**

**Appendix ..............................................................**

1. Source Code...........................................................................

**1. Introduction**

Since ancient times, there are a lot of change in the behavior and cultures of people in different places. According to their way of living, the health care and life expectancy of people varies among each other. These differences are may be based on various factors such as Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth and other demographic factors.

**1.1. Overview**

Life expectancy is a statistical measure of the average time a human being is expected to live. A typical Regression Machine Learning project leverages historical data to predict insights into the future. This problem statement is aimed at predicting Life Expectancy rate of a country given various features.

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 health care system and some specific disease related deaths that happened in the country are given in a data set.

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**

The purpose of this project is that the people from various places can easily predict their life expectancy by providing the inputs asked by the model.

This software can be used by all people in the world because the training part of this model contains inputs and predictions of more number of countries.

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 capital. 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.

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 skill set 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**

There are so many organizations that are making research in the prediction of life expectancy. Many research papers dealing with the creation of this model under many algorithms such as Machine Learning, Deep learning and programming languages such as Python and Java script.

**2.1. Existing Problem**

The World Health Organization (WHO) began producing annual life tables for all Member States in 1999. These life tables are a basic input to all WHO estimates of global, regional and country-level patterns and trends in allcause and cause-specific mortality. After the publication of life tables for years to 2009 in the 2011 edition of World Health Statistics, WHO has shifted to a two year cycle for the updating of life tables for all Member States. Even still the model is not really updated in every fields. WHO applies standard methods to the analysis of Member State data to ensure comparability of estimates across countries. This will inevitably result in differences for some Member States with official estimates for quantities such as life expectancy, where a variety of different projection methods and other methods are used.

**2.2. Proposed Solution**

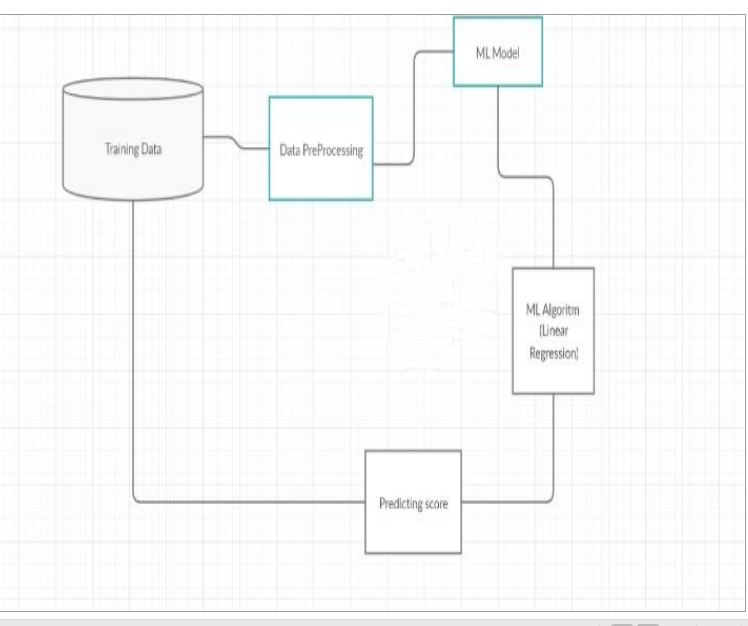
So many people were expecting to use a model of life expectancy prediction. In order to that, many institutions and companies are leading their team to build that model. In my project, I have proposed a solution to predict the life expectancy using machine learning. Machine Learning is the process of

training the computer to think and decide solutions like human. The reason why I have chosen this architecture was only with the help of Machine Learning, deep understanding of the data and an ability to create a model can be done. Design a Regression model to predict life expectancy ratio of a given

country based on some features provided such as year, GDP (gross domestic product), education, alcohol intake of people in the country, expenditure on health care system and some specific disease related deaths that happened in the country.

**3. Theoretical Analysis**

**3.1. Block Diagram**

****

**3.2. Hardware / Software Designing**

1. PROJECT PLANNING AND KICKOFF:

a. Understanding the project description and analyze the data and attributes in the given data set.

b. Creating Git hub account

c. Installing Slack and create account with the mail id

d. Learning to use Zoho writer.

2. EXPLORE IBM CLOUD PLATFORM:

a. Creating IBM cloud account with the mail id

b. Creating IBM academic initiative account with the mail id

c. Create a Node-Red starter application.

3. EXPLORE IBM WATSON SERVICES:

a. Exploring IBM Watson use cases.

b. Learning about IBM Watson Machine Learning.

4. INTRODUCTION TO WATSON STUDIO:

a. Learning to build own Machine Learning model using IBM Watson.

b. Automate the Machine Learning Model

5. PREDICTING LIFE EXPECTANCY WITH PYTHON:

a. Collecting Data set from www.kaggle.com

b. Creating IBM Watson services

c. Create a jupyter notebook and import data from Object storage.

6. PREDICTING LIFE EXPECTANCY WITHOUT PYTHON:

a. Created Node-Red model and integrated with Machine Learning model.

**4. Experimental Investigation**

Life Expectancy Data set:

The data set used is a life expectancy data set released by the World Health Organization.

The data set has the following features:

The data is saved as a csv file as LifeExpectancy.csv and it is read and stored in the life data variable. The Year column is dropped as it will not be used in the analysis. The first 5 rows are shown below. The data contains 21 columns and 2938 rows with the header row. The table contains data about:

• Countries

• Status

• Life Expectancy

• Adult Mortality

• 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

Prepossessing and cleaning the data sets:

• Before the data can be imported using the machine learning

libraries and can be trained, the data needs to be cleaned and

Pre processed.

• All the null values in the data set need to be either set to 0,

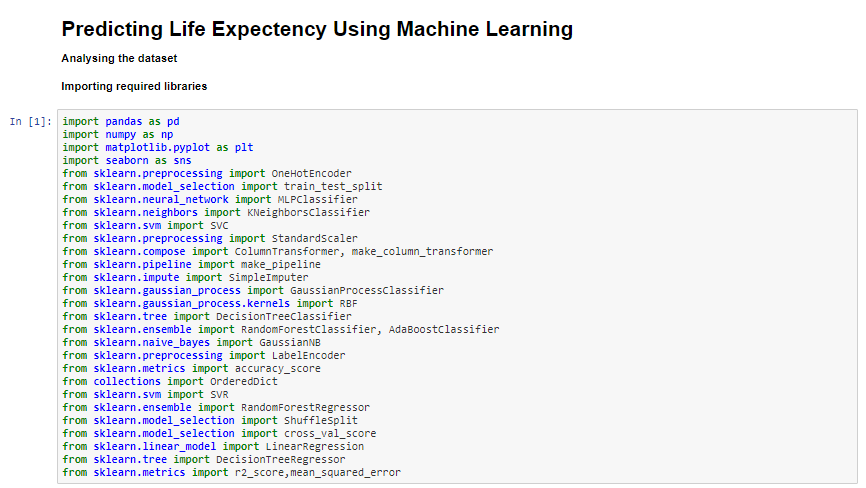
deleted or set equal to the mean value.

• In the cleaning process, I have set the null values as 0 for the

ease of calculation and maintaining the accuracy of the

model.

Importing the required libraries:



Loading the packages:

The following packages have been imported NumPy, Pandas, Matplotlib, SciPy, and Seaborn. Sklearn is the most widely used package for the machine learning process. The following subpackages have been used:

1. train\_test\_split

2. linear\_model

3. model selection

4. metrics

5. tree

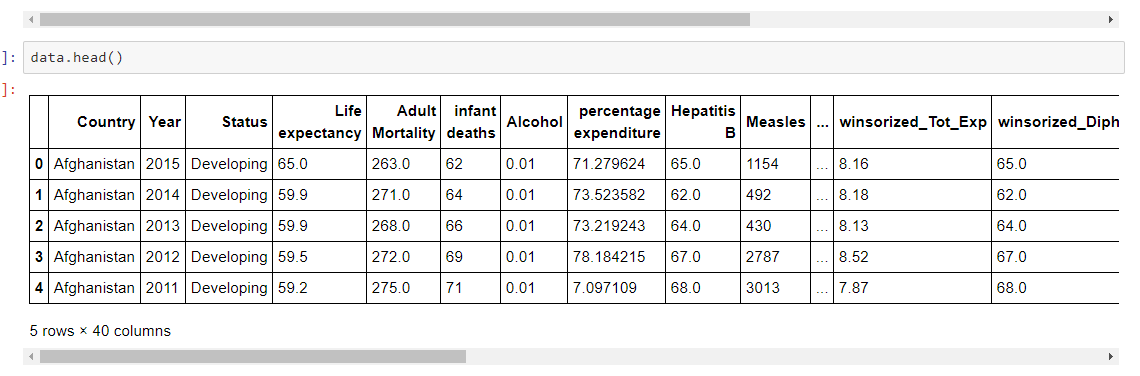
6. ensemble

7. pre-processing

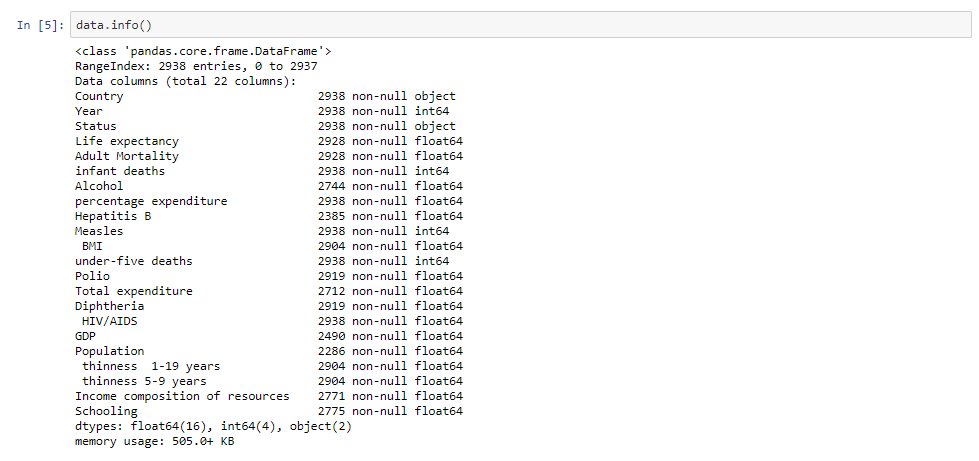
Importing the data set:

The required data set in the csv file is imported as the panda data frame.





Displaying the structure of the dataset:



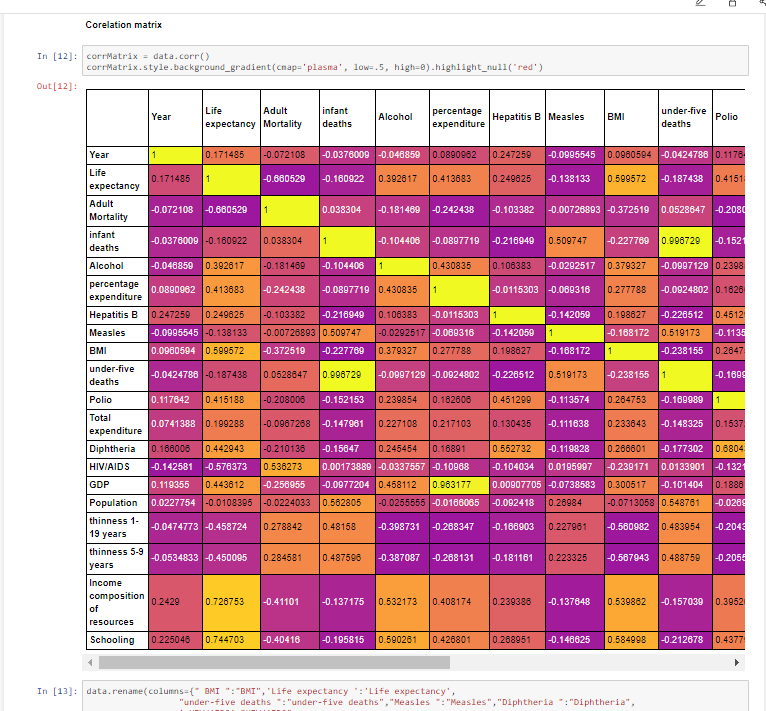
EDA:



Now we will plot the correlation matrix visualizing it with a heat map. The legend tells that the warmer colors show higher and positive correlation, while the colder low or negative.

There is a very high correlation between thinness of 5-9 year-old and that of 1- 19 year-old. Also between population and infant deaths, under 5 deaths, another is between schooling and income composition of resources. On the other hand, Life expectancy and Adult Mortality are very highly negatively correlated.

Corelation martix:-





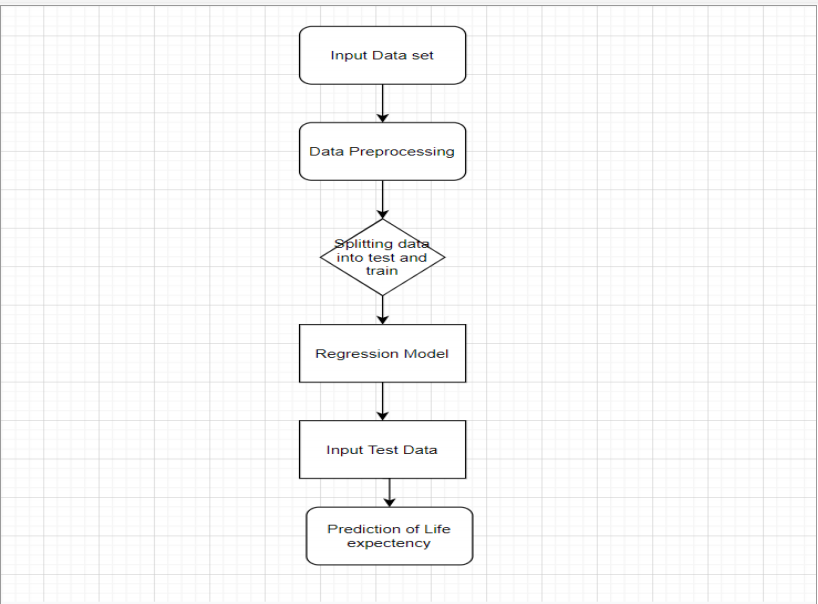
Linear regression with polynomial functions:

Three models have been created. The Algorithms have been used to test if they can provide good prediction with fewer errors while predicting the life expectancy for new data. The Model Algorithms used are:

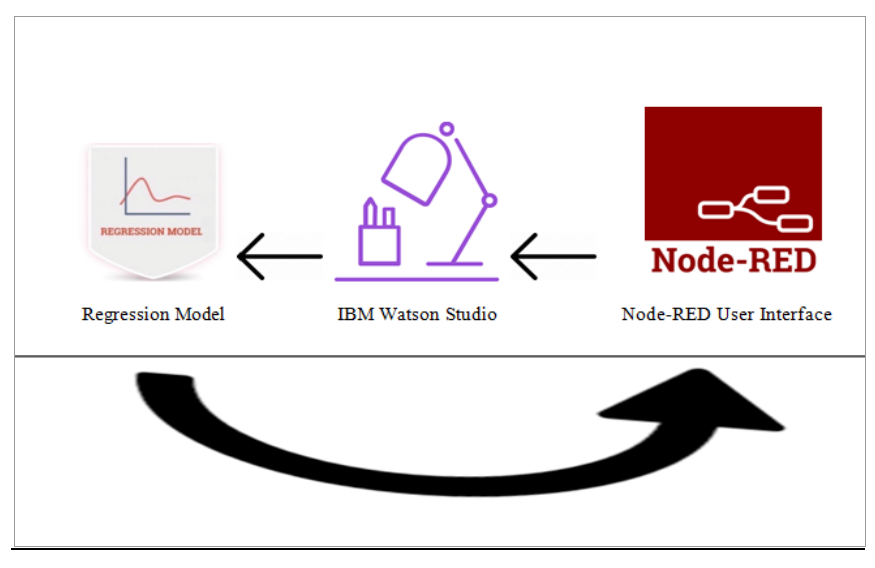
• Decision Tree Regression

• Linear Regression

• Random Forest Regression

1. **Flowchart **

**UI USING THE NODE RED**

****

To integrate the ML model with the UI, we would be using

the Node Red functionality provided by the IBM Watson

Studio.

To design the UI, we need to import the flow of the UI.

Once, we have setup the flow, we need to integrate the ML model with it. To integrate the ML Model with it we need to access the endpoint URL of our ML Model.

Components of the flow are:

Form: The form contains all the elements of the UI. All

the labels are associated with a variable.

Http requests: To setup the flow, we need two http requests.

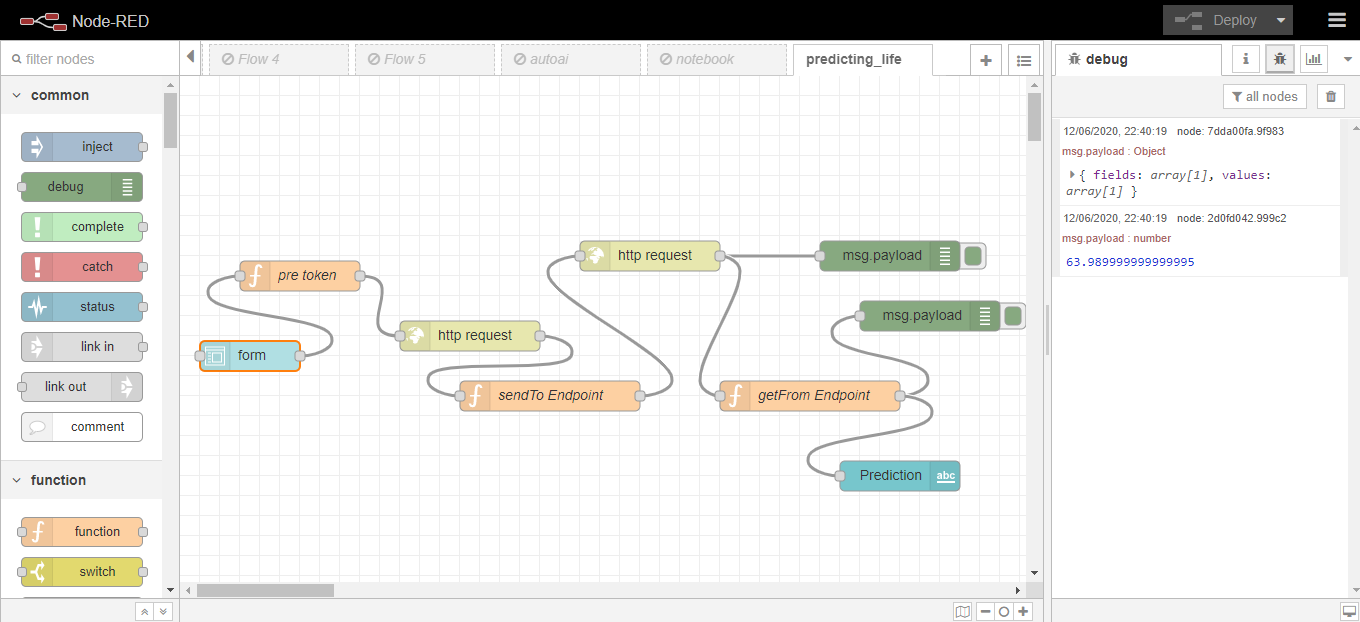
The first http request requires a token to connect to the

machine learning service of the Watson studio.

The second http request helps us in integrating the model using

the endpoint URL.

Once the flow has been setup, we deploy the model.

****

**6. Result**

Web based UI was developed by integrating all the services using NODERED.

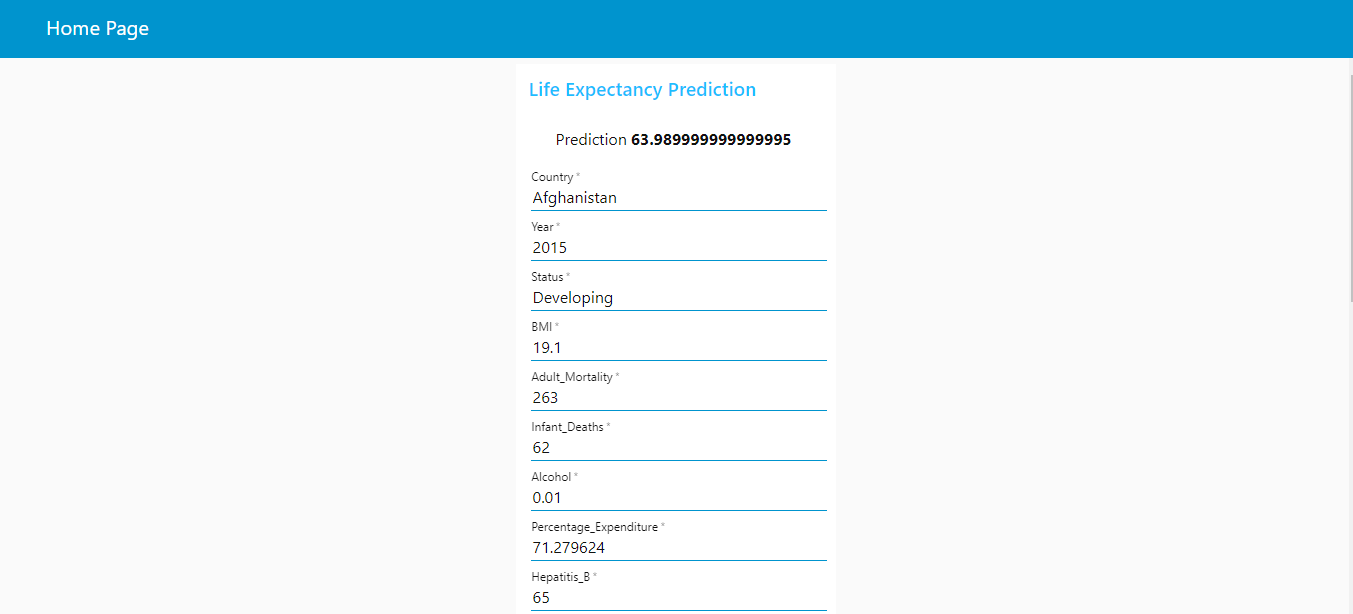
URL for UI Dashboard:<https://node-red-socib.eu-gb.mybluemix.net/ui/>

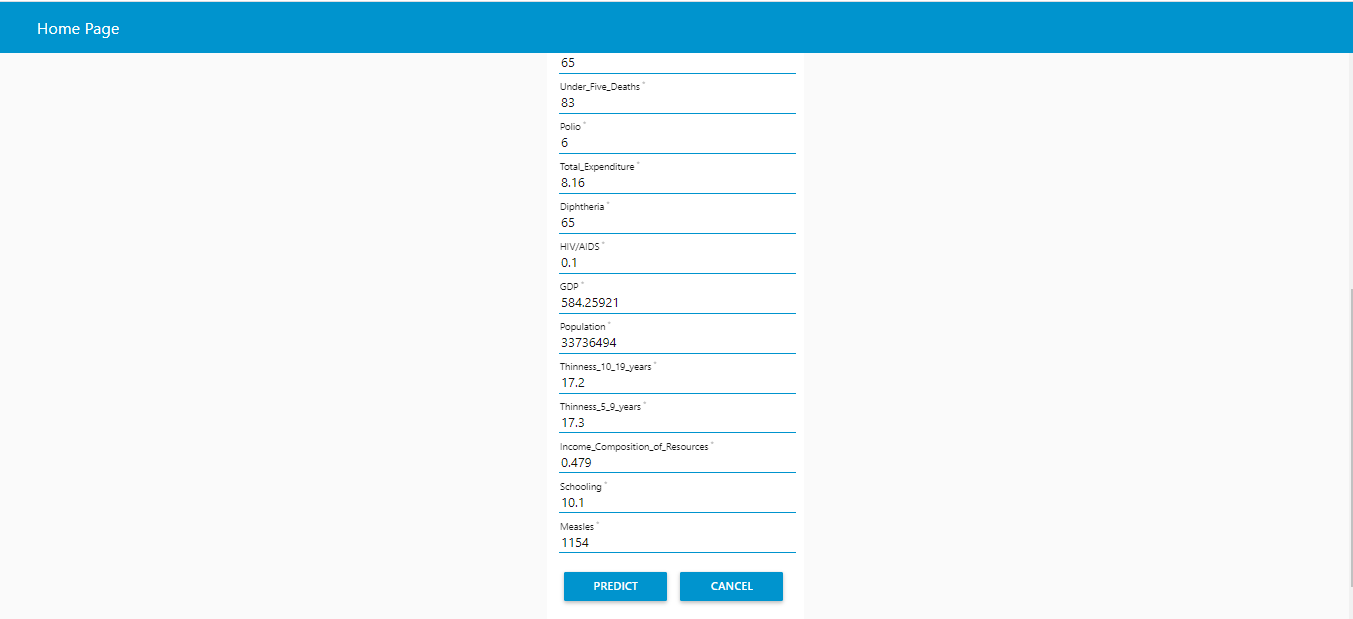
URL for Notebook:

https://github.com/SmartPracticeschool/llSPS-INT-2051-Predicting-Life-Expectancy-using-Machine-Learning.git/Predicting\_Life\_Expectency.ipynb

While giving the inputs for the country Afghanistan in the year 2015, the

life expectancy value **63.989999999999995**has been predicted.





Accuracy of my machine learning code is :-0.9563978774982961 =    95.6%

As per the data set the prediction is 65, where as my code could give **63.989999999999995**means about 95% prediction.

**7. Advantages & 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 frontend. 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**

• 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.

• 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.

• 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**

• The end product is a webpage created and deployed on node-red app of IBM cloud. The backend of webpage is a linear regression model created and deployed on Watson Studio using machine learning service.

• This model can be used to predict the life expectancy of people in different places.

• This model contains various factors such as Country, Year, Status, Life Expectancy, 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.

• With the help of all these input values, the model will predict the life expectancy of such people.

• The accuracy level of prediction in my model is more than 95%.

• From the help of this model, the life expectancies of more than 190 countries can be detected.

**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**

1. Node-RED Starter Application :

https://developer.ibm.com/tutorials/how-to-create-a-node

red-starter-applicat ion/

2. Watson Studio Cloud :

https://bookdown.org/caoying4work

/watsonstudio-workshop/jn.html

3. Dataset Reference:

https://www.kaggle.com/kumaraja

rshi/life-expectancy-who

4. IBM Cloud Services :

https://www.youtube.com/watch?v=DBRGlAHdj48&l

ist=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L

5. Import the Dataset into Jupyter Notebook :

https://www.youtube.com/watch?v=Jtej3Y6uUng

**Appendix**

**A. Source Code**

**Node-RED Flow code**

[{"id":"cd6986ec.6ccc08","type":"tab","label":"predicting\_life","disabled":false,"info":""},{"id":"4d5e4443.d2ba2c","type":"ui\_form","z":"cd6986ec.6ccc08","name":"","label":"","group":"625772b2.1d1aec","order":2,"width":0,"height":0,"options":[{"label":"Country","value":"a","type":"text","required":true,"rows":null},{"label":"Year","value":"b","type":"number","required":true,"rows":null},{"label":"Status","value":"c","type":"text","required":true,"rows":null},{"label":"BMI","value":"d","type":"number","required":true,"rows":null},{"label":"Adult\_Mortality","value":"e","type":"number","required":true,"rows":null},{"label":"Infant\_Deaths","value":"f","type":"number","required":true,"rows":null},{"label":"Alcohol","value":"g","type":"number","required":true,"rows":null},{"label":"Percentage\_Expenditure","value":"h","type":"number","required":true,"rows":null},{"label":"Hepatitis\_B","value":"i","type":"number","required":true,"rows":null},{"label":"Under\_Five\_Deaths","value":"j","type":"number","required":true,"rows":null},{"label":"Polio","value":"k","type":"number","required":true,"rows":null},{"label":"Total\_Expenditure","value":"l","type":"number","required":true,"rows":null},{"label":"Diphtheria","value":"m","type":"number","required":true,"rows":null},{"label":"HIV/AIDS","value":"n","type":"number","required":true,"rows":null},{"label":"GDP","value":"o","type":"number","required":true,"rows":null},{"label":"Population","value":"p","type":"number","required":true,"rows":null},{"label":"Thinness\_10\_19\_years","value":"q","type":"number","required":true,"rows":null},{"label":"Thinness\_5\_9\_years","value":"r","type":"number","required":true,"rows":null},{"label":"Income\_Composition\_of\_Resources","value":"s","type":"number","required":true,"rows":null},{"label":"Schooling","value":"t","type":"number","required":true,"rows":null},{"label":"Measles","value":"u","type":"number","required":true,"rows":null}],"formValue":{"a":"","b":"","c":"","d":"","e":"","f":"","g":"","h":"","i":"","j":"","k":"","l":"","m":"","n":"","o":"","p":"","q":"","r":"","s":"","t":"","u":""},"payload":"","submit":"Predict","cancel":"cancel","topic":"","x":70,"y":280,"wires":[["ed7da5f1.639598"]]},{"id":"ed7da5f1.639598","type":"function","z":"cd6986ec.6ccc08","name":"pre token","func":"//make user given values as global variables\nglobal.set(\"a\",msg.payload.a);\nglobal.set(\"b\",msg.payload.b);\nglobal.set(\"c\",msg.payload.c);\nglobal.set(\"d\",msg.payload.d);\nglobal.set(\"e\",msg.payload.e);\nglobal.set(\"f\",msg.payload.f);\nglobal.set(\"g\",msg.payload.g);\nglobal.set(\"h\",msg.payload.h);\nglobal.set(\"i\",msg.payload.i);\nglobal.set(\"j\",msg.payload.j);\nglobal.set(\"k\",msg.payload.k);\nglobal.set(\"l\",msg.payload.l);\nglobal.set(\"m\",msg.payload.m);\nglobal.set(\"n\",msg.payload.n);\nglobal.set(\"o\",msg.payload.o);\nglobal.set(\"p\",msg.payload.p);\nglobal.set(\"q\",msg.payload.q);\nglobal.set(\"r\",msg.payload.r);\nglobal.set(\"s\",msg.payload.s);\nglobal.set(\"t\",msg.payload.t);\nglobal.set(\"u\",msg.payload.u);\n\n//following are required to receive a token\nvar apikey=\"xhl4i0H7TEwuw1DXZxE0QDoFqRMMFDbZKLQbYv\_ODpLf\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"};\nmsg.payload={\"grant\_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey};\nreturn msg;\n","outputs":1,"noerr":0,"x":120,"y":200,"wires":[["bccee596.985d68"]]},{"id":"e0912681.849d08","type":"http request","z":"cd6986ec.6ccc08","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://us-south.ml.cloud.ibm.com/v3/wml\_instances/d93ee0ab-4ed6-4bf4-9d92-38db57e60242/deployments/a11b5676-0b4c-4678-9e27-ab7fa0d112b2/online","tls":"","persist":false,"proxy":"","authType":"basic","x":470,"y":180,"wires":[["7dda00fa.9f983","2ddb69c1.5e1816"]]},{"id":"2d0fd042.999c2","type":"debug","z":"cd6986ec.6ccc08","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":750,"y":240,"wires":[]},{"id":"2ddb69c1.5e1816","type":"function","z":"cd6986ec.6ccc08","name":"getFrom Endpoint","func":"msg.payload=msg.payload.values[0][0];\nreturn msg;","outputs":1,"noerr":0,"x":630,"y":320,"wires":[["2d0fd042.999c2","a58ca827.532ff8"]]},{"id":"7dda00fa.9f983","type":"debug","z":"cd6986ec.6ccc08","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"payload","targetType":"msg","x":710,"y":180,"wires":[]},{"id":"a3e48b5.73d4d78","type":"function","z":"cd6986ec.6ccc08","name":"sendTo Endpoint","func":"//get token and make headers\nvar token=msg.payload.access\_token;\nvar instance\_id=\"3f962ec7-535b-4fc4-9592-d889f145ba8f\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance\_id}\n\n//get variables that are set earlier\nvar a = global.get(\"a\");\nvar b = global.get(\"b\");\nvar c = global.get(\"c\");\nvar d = global.get(\"d\");\nvar e = global.get(\"e\");\nvar f = global.get(\"f\");\nvar g = global.get(\"g\");\nvar h = global.get(\"h\");\nvar i = global.get(\"i\");\nvar j = global.get(\"j\");\nvar k = global.get(\"k\");\nvar l = global.get(\"l\");\nvar m = global.get(\"m\");\nvar n = global.get(\"n\");\nvar o = global.get(\"o\");\nvar p = global.get(\"p\");\nvar q = global.get(\"q\");\nvar r = global.get(\"r\");\nvar s = global.get(\"s\");\nvar t = global.get(\"t\");\nvar u = global.get(\"u\");\n\n//send the user values to service endpoint\nmsg.payload = \n{\"fields\":[\"Country\", \"Year\", \"Status\", \n\"BMI\", \"Adult\_Mortality\", \"Infant\_Deaths\", \"Alcohol\", \"Percentage\_Expenditure\", \"Hepatitis\_B\", \"Under\_Five\_Deaths\", \"Polio\", \"Total\_Expenditure\", \"Diphtheria\", \"HIV/AIDS\", \"GDP\",\"Population\", \"Thinness\_10\_19\_years\", \"Thinness\_5\_9\_years\",\n \"Income\_Composition\_of\_Resources\", \"Schooling\", \"Measles\"],\n\"values\":[[a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u]]};\n\nreturn msg;\n","outputs":1,"noerr":0,"x":370,"y":320,"wires":[["e0912681.849d08"]]},{"id":"bccee596.985d68","type":"http request","z":"cd6986ec.6ccc08","name":"","method":"POST","ret":"obj","paytoqs":false,"url":"https://iam.cloud.ibm.com/identity/token","tls":"","persist":false,"proxy":"","authType":"basic","x":290,"y":260,"wires":[["a3e48b5.73d4d78"]]},{"id":"a58ca827.532ff8","type":"ui\_text","z":"cd6986ec.6ccc08","group":"625772b2.1d1aec","order":1,"width":0,"height":0,"name":"","label":"Prediction","format":"{{msg.payload}}","layout":"row-center","x":720,"y":400,"wires":[]},{"id":"625772b2.1d1aec","type":"ui\_group","z":"","name":"Life Expectancy Prediction","tab":"588b62e9.d8a85c","order":1,"disp":true,"width":"6","collapse":false},{"id":"588b62e9.d8a85c","type":"ui\_tab","z":"","name":"Home Page","icon":"dashboard","disabled":false,"hidden":false}]

**NOTEBOOK**

# **Predicting Life Expectency Using Machine Learning**

****Analysing the dataset****

****Importing required libraries****

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**from** sklearn.preprocessing **import** OneHotEncoder

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.neural\_network **import** MLPClassifier

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn.svm **import** SVC

**from** sklearn.preprocessing **import** StandardScaler

**from** sklearn.compose **import** ColumnTransformer, make\_column\_transformer

**from** sklearn.pipeline **import** make\_pipeline

**from** sklearn.impute **import** SimpleImputer

**from** sklearn.gaussian\_process **import** GaussianProcessClassifier

**from** sklearn.gaussian\_process.kernels **import** RBF

**from** sklearn.tree **import** DecisionTreeClassifier

**from** sklearn.ensemble **import** RandomForestClassifier, AdaBoostClassifier

**from** sklearn.naive\_bayes **import** GaussianNB

**from** sklearn.preprocessing **import** LabelEncoder

**from** sklearn.metrics **import** accuracy\_score

**from** collections **import** OrderedDict

**from** sklearn.svm **import** SVR

**from** sklearn.ensemble **import** RandomForestRegressor

**from** sklearn.model\_selection **import** ShuffleSplit

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

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.tree **import** DecisionTreeRegressor

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

**Dataset:-**

**import** types

**import** pandas **as** pd

**from** botocore.client **import** Config

**import** ibm\_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\_ae8f8423fd0745edae54655bc34938b9 **=** ibm\_boto3.client(service\_name**=**'s3',

ibm\_api\_key\_id**=**'vZMOoHD60Qn0Npd4F8wAxH6LtS2VMNlDntP-G9bigoqu',

ibm\_auth\_endpoint**=**"https://iam.cloud.ibm.com/oidc/token",

config**=**Config(signature\_version**=**'oauth'),

endpoint\_url**=**'https://s3-api.us-geo.objectstorage.service.networklayer.com')

​

body **=** client\_ae8f8423fd0745edae54655bc34938b9.get\_object(Bucket**=**'internship-donotdelete-pr-miejikeevvgb8v',Key**=**'datasets\_12603\_17232\_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 )

​

data **=** pd.read\_csv(body)

data.head()

​

data.head()

data.describe()

data.info()

data.isnull().sum()

country\_list **=** data.Country.unique()

len(country\_list)

****Handling Missing value:-****

country\_list **=** data.Country.unique()

fill\_list **=** ['Country', 'Year', 'Status', 'Life expectancy ', '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']

Filling missing value according to country column using interpolate()

**for** country **in** country\_list:

data.loc[data['Country'] **==** country,fill\_list] **=** data.loc[data['Country'] **==** country,fill\_list].interpolate()

data.dropna(inplace**=True**)

data.shape *#(1987, 22) size reduced*

data.isna().sum()

****Corelation matrix:-****

corrMatrix **=** data.corr()

corrMatrix.style.background\_gradient(cmap**=**'plasma', low**=**.5, high**=**0).highlight\_null('red')

data.rename(columns**=**{" BMI ":"BMI",'Life expectancy ':'Life expectancy',

"under-five deaths ":"under-five deaths","Measles ":"Measles","Diphtheria ":"Diphtheria",

' HIV/AIDS':"HIV/AIDS",

" thinness 1-19 years":"thinness 10-19 years"," thinness 5-9 years":"thinness 5-9 years"},inplace**=True**)

****Removing outliers:-****

col\_dict **=** {'Life expectancy':1 , 'Adult Mortality':2 ,

'Alcohol':3 , 'percentage expenditure': 4, 'Hepatitis B': 5,

'Measles' : 6, 'BMI': 7, 'under-five deaths' : 8, 'Polio' : 9, 'Total expenditure' :10,

'Diphtheria':11, 'HIV/AIDS':12, 'GDP':13, 'Population' :14,

'thinness 10-19 years' :15, 'thinness 5-9 years' :16,

'Income composition of resources' : 17, 'Schooling' :18, 'infant deaths':19}

​

**import** matplotlib.pyplot **as** plt

plt.figure(figsize**=**(20,30))

​

**for** variable,i **in** col\_dict.items():

plt.subplot(5,4,i)

plt.boxplot(data[variable],whis**=**1.5)

plt.title(variable)

​

plt.show()

​

**import** numpy **as** np

​

**for** variable **in** col\_dict.keys():

q75, q25 **=** np.percentile(data[variable], [75 ,25])

iqr **=** q75 **-** q25

min\_val **=** q25 **-** (iqr**\***1.5)

max\_val **=** q75 **+** (iqr**\***1.5)

print("Number of outliers and percentage of it in {} : {} and {}".format(variable,

len((np.where((data[variable] **>** max\_val) **|** (data[variable] **<** min\_val))[0])),

len((np.where((data[variable] **>** max\_val) **|** (data[variable] **<** min\_val))[0]))**\***100**/**1987))

**from** scipy.stats.mstats **import** winsorize

winsorized\_Life\_Expectancy **=** winsorize(data['Life expectancy'],(0.01,0))

winsorized\_Adult\_Mortality **=** winsorize(data['Adult Mortality'],(0,0.03))

winsorized\_Infant\_Deaths **=** winsorize(data['infant deaths'],(0,0.10))

winsorized\_Alcohol **=** winsorize(data['Alcohol'],(0,0.01))

winsorized\_Percentage\_Exp **=** winsorize(data['percentage expenditure'],(0,0.12))

winsorized\_HepatitisB **=** winsorize(data['Hepatitis B'],(0.11,0))

winsorized\_Measles **=** winsorize(data['Measles'],(0,0.19))

winsorized\_Under\_Five\_Deaths **=** winsorize(data['under-five deaths'],(0,0.12))

winsorized\_Polio **=** winsorize(data['Polio'],(0.09,0))

winsorized\_Tot\_Exp **=** winsorize(data['Total expenditure'],(0,0.01))

winsorized\_Diphtheria **=** winsorize(data['Diphtheria'],(0.10,0))

winsorized\_HIV **=** winsorize(data['HIV/AIDS'],(0,0.16))

winsorized\_GDP **=** winsorize(data['GDP'],(0,0.13))

winsorized\_Population **=** winsorize(data['Population'],(0,0.14))

winsorized\_thinness\_10\_19\_years **=** winsorize(data['thinness 10-19 years'],(0,0.04))

winsorized\_thinness\_5\_9\_years **=** winsorize(data['thinness 5-9 years'],(0,0.04))

winsorized\_Income\_Comp\_Of\_Resources **=** winsorize(data['Income composition of resources'],(0.05,0))

winsorized\_Schooling **=** winsorize(data['Schooling'],(0.02,0.01))

winsorized\_list **=** [winsorized\_Life\_Expectancy,winsorized\_Adult\_Mortality,winsorized\_Alcohol,winsorized\_Measles,winsorized\_Infant\_Deaths,

winsorized\_Percentage\_Exp,winsorized\_HepatitisB,winsorized\_Under\_Five\_Deaths,winsorized\_Polio,winsorized\_Tot\_Exp,winsorized\_Diphtheria,

winsorized\_HIV,winsorized\_GDP,winsorized\_Population,winsorized\_thinness\_10\_19\_years,winsorized\_thinness\_5\_9\_years,

winsorized\_Income\_Comp\_Of\_Resources,winsorized\_Schooling]

​

**for** variable **in** winsorized\_list:

q75, q25 **=** np.percentile(variable, [75 ,25])

iqr **=** q75 **-** q25

​

min\_val **=** q25 **-** (iqr**\***1.5)

max\_val **=** q75 **+** (iqr**\***1.5)

print("Number of outliers after winsorization in : {} ".format(len(np.where((variable **>** max\_val) **|** (variable **<** min\_val))[0])))

​

data['winsorized\_Life\_Expectancy'] **=** winsorized\_Life\_Expectancy

data['winsorized\_Adult\_Mortality'] **=** winsorized\_Adult\_Mortality

data['winsorized\_Infant\_Deaths'] **=** winsorized\_Infant\_Deaths

data['winsorized\_Alcohol'] **=** winsorized\_Alcohol

data['winsorized\_Percentage\_Exp'] **=** winsorized\_Percentage\_Exp

data['winsorized\_HepatitisB'] **=** winsorized\_HepatitisB

data['winsorized\_Under\_Five\_Deaths'] **=** winsorized\_Under\_Five\_Deaths

data['winsorized\_Polio'] **=** winsorized\_Polio

data['winsorized\_Tot\_Exp'] **=** winsorized\_Tot\_Exp

data['winsorized\_Diphtheria'] **=** winsorized\_Diphtheria

data['winsorized\_HIV'] **=** winsorized\_HIV

data['winsorized\_GDP'] **=** winsorized\_GDP

data['winsorized\_Population'] **=** winsorized\_Population

data['winsorized\_thinness\_10\_19\_years'] **=** winsorized\_thinness\_10\_19\_years

data['winsorized\_thinness\_5\_9\_years'] **=** winsorized\_thinness\_5\_9\_years

data['winsorized\_Income\_Comp\_Of\_Resources'] **=** winsorized\_Income\_Comp\_Of\_Resources

data['winsorized\_Schooling'] **=** winsorized\_Schooling

data['winsorized\_Measles'] **=** winsorized\_Measles

data.shape *#More 18 columns are added*

****EDA:-****

​

data.columns

sns.distplot(data['Life expectancy'],kde**=True**)

disease\_cols**=**data[['Life expectancy','Alcohol','Hepatitis B','Measles','BMI','Polio','Diphtheria','HIV/AIDS','Adult Mortality',

'infant deaths','under-five deaths','thinness 5-9 years','Schooling',

'percentage expenditure','Total expenditure','GDP','Population','Income composition of resources']]

disease\_cols.corr()

sns.pairplot(disease\_cols,diag\_kind**=**'kde')

col **=** ['Life expectancy','winsorized\_Life\_Expectancy','Adult Mortality','winsorized\_Adult\_Mortality','infant deaths',

'winsorized\_Infant\_Deaths','Alcohol','winsorized\_Alcohol','percentage expenditure','winsorized\_Percentage\_Exp','Hepatitis B',

'winsorized\_HepatitisB','under-five deaths','winsorized\_Under\_Five\_Deaths','Polio','winsorized\_Polio','Total expenditure',

'winsorized\_Tot\_Exp','Diphtheria','winsorized\_Diphtheria','HIV/AIDS','winsorized\_HIV','GDP','winsorized\_GDP',

'Population','winsorized\_Population','thinness 10-19 years','winsorized\_thinness\_10\_19\_years','thinness 5-9 years',

'winsorized\_thinness\_5\_9\_years','Income composition of resources','winsorized\_Income\_Comp\_Of\_Resources',

'Schooling','winsorized\_Schooling','Measles','winsorized\_Measles','GDP','winsorized\_GDP']

​

plt.figure(figsize**=**(15,75))

​

**for** i **in** range(len(col)):

plt.subplot(19,2,i**+**1)

plt.hist(data[col[i]])

plt.title(col[i])

​

plt.show()

data.describe(include**=** 'O') *#include specifies the list of datatype to be incluyded .here is Object*

plt.figure(figsize**=**(6,6))

plt.bar(data.groupby('Status')['Status'].count().index,data.groupby('Status')['winsorized\_Life\_Expectancy'].mean())

plt.ylabel("Avg Life\_Expectancy")

plt.title("Life\_Expectancy w.r.t Status")

plt.show()

​

country\_data **=** data.groupby('Country')['winsorized\_Life\_Expectancy'].mean().sort\_values(ascending**=True**)

country\_data.plot(kind**=**'bar' ,figsize**=**(50,15),fontsize**=**30,color**=**'g')

plt.title("Life\_Expectancy w.r.t Country",fontsize**=**30)

plt.xlabel("Country",fontsize**=**30)

plt.ylabel("Avg Life\_Expectancy")

plt.show()

plt.figure(figsize**=**(7,5))

plt.bar(data.groupby('Year')['Year'].count().index,data.groupby('Year')['winsorized\_Life\_Expectancy'].mean())

plt.xlabel("Year",fontsize**=**12)

plt.ylabel("Avg Life\_Expectancy",fontsize**=**12)

plt.show()

cor\_matrix**=**data.corr()

print(cor\_matrix['winsorized\_Life\_Expectancy'].sort\_values(ascending**=False**))

**import** seaborn **as** sns

**from** pandas.plotting **import** scatter\_matrix

attributes**=** ['winsorized\_Life\_Expectancy','winsorized\_Income\_Comp\_Of\_Resources','winsorized\_Schooling'

,'winsorized\_Diphtheria','winsorized\_Polio','winsorized\_Adult\_Mortality','winsorized\_Alcohol','winsorized\_Measles','winsorized\_Infant\_Deaths',

'winsorized\_Percentage\_Exp','winsorized\_HepatitisB','winsorized\_Under\_Five\_Deaths','winsorized\_Tot\_Exp',

'winsorized\_HIV','winsorized\_GDP','winsorized\_Population','winsorized\_thinness\_10\_19\_years','winsorized\_thinness\_5\_9\_years']

cormat**=**data[attributes].corr()

plt.figure(figsize**=**(15,15))

sns.heatmap(cormat, square**=True**, annot**=True**, linewidths**=**.5)

plt.show()

subplot

​

plt.figure(figsize**=**(18,40))

​

plt.subplot(6,3,1)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Adult\_Mortality"])

plt.title("LifeExpectancy vs AdultMortality")

​

plt.subplot(6,3,2)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Infant\_Deaths"])

plt.title("LifeExpectancy vs Infant\_Deaths")

​

plt.subplot(6,3,3)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Alcohol"])

plt.title("LifeExpectancy vs Alcohol")

​

plt.subplot(6,3,4)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Percentage\_Exp"])

plt.title("LifeExpectancy vs Percentage\_Exp")

​

plt.subplot(6,3,5)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_HepatitisB"])

plt.title("LifeExpectancy vs HepatitisB")

​

plt.subplot(6,3,6)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Under\_Five\_Deaths"])

plt.title("LifeExpectancy vs Under\_Five\_Deaths")

​

plt.subplot(6,3,7)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Polio"])

plt.title("LifeExpectancy vs Polio")

​

plt.subplot(6,3,8)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Tot\_Exp"])

plt.title("LifeExpectancy vs Tot\_Exp")

​

​

plt.subplot(6,3,9)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Diphtheria"])

plt.title("LifeExpectancy vs Diphtheria")

​

plt.subplot(6,3,10)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_HIV"])

plt.title("LifeExpectancy vs HIV")

​

plt.subplot(6,3,11)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_GDP"])

plt.title("LifeExpectancy vs GDP")

​

plt.subplot(6,3,12)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Population"])

plt.title("LifeExpectancy vs Population")

​

*#plt.subplot(6,3,13)*

*#plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_thinness\_1to19\_years"])*

*#plt.title("LifeExpectancy vs thinness\_1to19\_years")*

​

*#plt.subplot(6,3,14)*

*#plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_thinness\_5to9\_years"])*

*#plt.title("LifeExpectancy vs thinness\_5to9\_years")*

​

plt.subplot(6,3,13)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Income\_Comp\_Of\_Resources"])

plt.title("LifeExpectancy vs Income\_Comp\_Of\_Resources")

​

plt.subplot(6,3,14)

plt.scatter(data["winsorized\_Life\_Expectancy"], data["winsorized\_Schooling"])

plt.title("LifeExpectancy vs Schooling")

​

​

plt.show()

round(data[['Status','winsorized\_Life\_Expectancy']].groupby(['Status']).mean(),2)

**import** scipy.stats **as** stats

stats.ttest\_ind(data.loc[data['Status']**==**'Developed','winsorized\_Life\_Expectancy'],data.loc[data['Status']**==**'Developing','winsorized\_Life\_Expectancy'])

​

data.columns

****Creating a new dataframe with refined data:-****

new\_data**=**pd.DataFrame(data**=**data,columns**=**['Country', 'Year', 'Status',

'BMI', 'winsorized\_Adult\_Mortality',

'winsorized\_Infant\_Deaths', 'winsorized\_Alcohol',

'winsorized\_Percentage\_Exp', 'winsorized\_HepatitisB',

'winsorized\_Under\_Five\_Deaths', 'winsorized\_Polio',

'winsorized\_Tot\_Exp', 'winsorized\_Diphtheria', 'winsorized\_HIV',

'winsorized\_GDP', 'winsorized\_Population',

'winsorized\_thinness\_10\_19\_years', 'winsorized\_thinness\_5\_9\_years',

'winsorized\_Income\_Comp\_Of\_Resources', 'winsorized\_Schooling',

'winsorized\_Measles',

'winsorized\_Life\_Expectancy'])

new\_data.shape

new\_data.head()

​

new\_data.rename(columns**=**{

'winsorized\_Adult\_Mortality':'Adult\_Mortality',

'winsorized\_Infant\_Deaths' :'Infant\_Deaths',

'winsorized\_Alcohol':'Alcohol',

'winsorized\_Percentage\_Exp':'Percentage\_Expenditure',

'winsorized\_HepatitisB':'Hepatitis\_B',

'winsorized\_Under\_Five\_Deaths':'Under\_Five\_Deaths',

'winsorized\_Polio':'Polio',

'winsorized\_Tot\_Exp':'Total\_Expenditure',

'winsorized\_Diphtheria':'Diphtheria',

'winsorized\_HIV':'HIV/AIDS',

'winsorized\_GDP':'GDP',

'winsorized\_Population':'Population',

'winsorized\_thinness\_10\_19\_years':'Thinness\_10\_19\_years',

'winsorized\_thinness\_5\_9\_years':'Thinness\_5\_9\_years',

'winsorized\_Income\_Comp\_Of\_Resources':'Income\_Composition\_of\_Resources',

'winsorized\_Schooling':'Schooling',

'winsorized\_Measles':'Measles',

'winsorized\_Life\_Expectancy':'Life\_Expectancy' } ,inplace**=True**)

new\_data.shape

new\_data.head()

​

new\_data.rename(columns**=**{

'winsorized\_Adult\_Mortality':'Adult\_Mortality',

'winsorized\_Infant\_Deaths' :'Infant\_Deaths',

'winsorized\_Alcohol':'Alcohol',

'winsorized\_Percentage\_Exp':'Percentage\_Expenditure',

'winsorized\_HepatitisB':'Hepatitis\_B',

'winsorized\_Under\_Five\_Deaths':'Under\_Five\_Deaths',

'winsorized\_Polio':'Polio',

'winsorized\_Tot\_Exp':'Total\_Expenditure',

'winsorized\_Diphtheria':'Diphtheria',

'winsorized\_HIV':'HIV/AIDS',

'winsorized\_GDP':'GDP',

'winsorized\_Population':'Population',

'winsorized\_thinness\_10\_19\_years':'Thinness\_10\_19\_years',

'winsorized\_thinness\_5\_9\_years':'Thinness\_5\_9\_years',

'winsorized\_Income\_Comp\_Of\_Resources':'Income\_Composition\_of\_Resources',

'winsorized\_Schooling':'Schooling',

'winsorized\_Measles':'Measles',

'winsorized\_Life\_Expectancy':'Life\_Expectancy' } ,inplace**=True**)

new\_data.head()

new\_data.columns

X **=** new\_data.drop('Life\_Expectancy', axis**=**1)

Y **=** pd.DataFrame(data**=**new\_data,columns**=**['Life\_Expectancy'])

​

Y.head()

​

X\_train, X\_test, Y\_train, Y\_test **=** train\_test\_split(X, Y, test\_size **=** 0.2, random\_state **=** 42)

****Creating a pipeline:-****

numeric\_features **=** ['Year', 'BMI',

'Adult\_Mortality', 'Infant\_Deaths', 'Alcohol', 'Percentage\_Expenditure',

'Hepatitis\_B', 'Under\_Five\_Deaths', 'Polio', 'Total\_Expenditure',

'Diphtheria', 'HIV/AIDS', 'GDP', 'Population', 'Thinness\_10\_19\_years',

'Thinness\_5\_9\_years', 'Income\_Composition\_of\_Resources', 'Schooling',

'Measles']

categorical\_features **=** ['Country', 'Status']

​

**from** sklearn.pipeline **import** Pipeline

**from** sklearn.preprocessing **import** OneHotEncoder

​

categorical\_transformer **=** Pipeline(steps**=**[

('onehot', OneHotEncoder(handle\_unknown**=**'ignore')),

])

**from** sklearn.impute **import** SimpleImputer

**from** sklearn.preprocessing **import** StandardScaler

​

numeric\_transformer **=** Pipeline(steps**=**[

('imputer', SimpleImputer(strategy**=**'median'))

])

**from** sklearn.compose **import** ColumnTransformer

​

preprocessor **=** ColumnTransformer(

transformers**=**[

('cat', categorical\_transformer, categorical\_features),

('num', numeric\_transformer, numeric\_features)

]

)

****Finding best algorithm:-****

models **=** OrderedDict([

( "Linear Regression", Pipeline([

('preprocessor', preprocessor),

('LRegressor', LinearRegression())]) ),

( "Decision Tree Regressor", Pipeline([

('preprocessor', preprocessor),

('DTRegressor', DecisionTreeRegressor())]) ),

( "Random Forest Regressor", Pipeline([

('preprocessor', preprocessor),

('RFRegressor', RandomForestRegressor())]) ),

])

scores **=** {}

**for** (name, model) **in** models.items():

model.fit(X\_train,Y\_train)

scores[name] **=**r2\_score(model.predict(X\_test), Y\_test)

scores **=** OrderedDict(sorted(scores.items()))

Scores

****Random forest regression:-****

​

RFRegressor **=** Pipeline([

('preprocessor', preprocessor),

('RFRegressor', RandomForestRegressor())

])

RFRegressor.fit(X\_train,Y\_train)

predict**=** RFRegressor.predict(X\_test)

​

r2\_score(predict, Y\_test)

****Deploying model:-****

*#!pip install watson-machine-learning-client*

**from** watson\_machine\_learning\_client **import** WatsonMachineLearningAPIClient

wml\_credentials**=**{

"apikey": "xhl4i0H7TEwuw1DXZxE0QDoFqRMMFDbZKLQbYv\_ODpLf",

"iam\_apikey\_description": "Auto-generated for key 1a9575d8-a1a0-478a-b987-43a1b4c01df7",

"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/09b34b516e7b43cc832c72171edf6c4d::serviceid:ServiceId-e65c4862-400d-4ca8-b5e8-55ecc07df011",

"instance\_id": "d93ee0ab-4ed6-4bf4-9d92-38db57e60242",

"url": "https://us-south.ml.cloud.ibm.com"

}

​

client **=** WatsonMachineLearningAPIClient(wml\_credentials)

metadata**=**{

client.repository.ModelMetaNames.DESCRIPTION:'Predicting life expectency',

client.repository.ModelMetaNames.AUTHOR\_NAME:'Suraj Reddy Alluri',

client.repository.ModelMetaNames.NAME:"Life\_expectancy",

client.repository.ModelMetaNames.FRAMEWORK\_NAME:"scikit-learn",

client.repository.ModelMetaNames.FRAMEWORK\_VERSION:"0.22"

}

model\_details**=**client.repository.store\_model(RFRegressor, 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**=**"life\_expectancy")

​

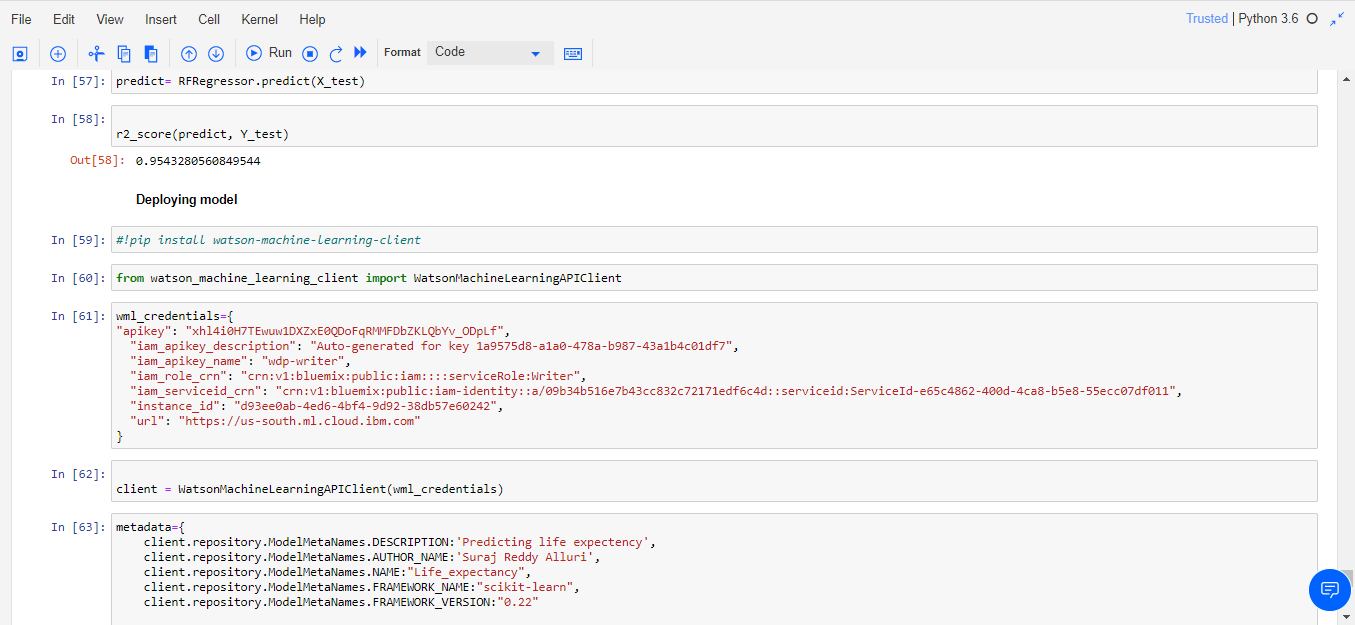
scoring\_endpoint **=** client.deployments.get\_scoring\_url(created\_deployment)

scoring\_endpoint

client.deployments.list()

**ACCURACY:-**

0.9543280560849544



- REPORT SUBMITTED

BY

ALLURI SURAJ REDDY