Remote Summer Internship Program 2020

Machine Learning, Career Basic Program

Smartinternz, SmartBridge



Predicting Life Expectancy using

Machine Learning

Internship Report

By

Aditya Reddy Lellapati



From:

SmartBridge Educational Services Pvt Ltd. Plot No 132, Above DCB bank, 2nd floor, Bapuji Nagar, Habsiguda,

Nacharam Main Road, Hyderabad – 500076

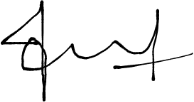
Date: 17/05/2020.

Dear **Lellapati Aditya Reddy**

**SmartBridge Educational Services Pvt Ltd**, is pleased to offer a training cum internship opportunity. During this period you would be associated with our mentors and The Smart Practice School Platform.

For further details you can contact us on +91 8499004200.

Thanks and Regards,



Ch. Jaya Prakash

Program Manager – SIP2020, Date: 17/05/2020.

1 **INTRODUCTION**

1.1 Overview

1.2 Purpose

**2 LITERATURE SURVEY**

2.1 Existing problem

2.2 Proposed solution

**3 THEORITICAL ANALYSIS**

3.1 Block diagram

3.2 Hardware / Software designing

**4 EXPERIMENTAL INVESTIGATIONS**

**5 FLOWCHART**

**6 RESULT**

**7 ADVANTAGES & DISADVANTAGES**

**8 CONCLUSION**

**9 BIBILOGRAPHY**

**APPENDIX**

A. Source code

**INTRODUCTION**

**1.1 Overview**

**Program Information**

Summer Internship Program by Smartbridge in collaboration with IBM is an annual initiative taken up by them to teach and prepare students across the globe for industry experience. They believe that experiential learning and development in a professional like environment can only bridge the gap between students and industries opening ways for both of them to achieve better results. This initiative enables students to better their resume to embark upon a successful industrial journey. They provide various roles for internship according to the possible aptitude of the students such as Artificial Intelligence, Machine Learning, Internet of Things etc.

Remote Summer Internship Program 2020 is a 4-Weeks Industry Driven, Project based, Mentor supported Program. Selected interns will be joined to a project team, where he will learn the technology and complete the assigned tasks. SmartInternz platform provides the company like collaborative environment to build professional skills along with technical skills.

A project is assigned to students individually with access to the platform wherein the students code and deploy their model just like in industries. This not only introduces them to environments like IBM cloud but also helps them in understanding the industrial environment better. Classes are organised according to the technologies to be taught and doubt sessions are taken up by the mentors to help students complete their projects too.

**1.2 Purpose**

**Project title :** Predicting Life Expectancy using Machine Learning

**Category :** Machine Learning

**Skills required :** Python, IBM Cloud, IBM Watson

**Problem Description**

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.

Life Expectancy is a statistical measue of the average time a human being is expected to live, Life expectancy depends on various factors : Regional variations, Economic Circumstacnes, Sex Differences, Mental Illnesses ,Physical Illnesses , Education, Year of the ir birth ans 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.

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 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.

**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.

**LITERATURE SURVEY**

**2.1 Existing Problem**

Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that affect of immunization and human development index was not taken into account in the past. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries. Hence, this gives motivation to resolve both the factors stated previously by formulating a regression model based on mixed effects model and multiple linear regression while considering data from a period of 2000 to 2015 for all the countries. Important immunization like Hepatitis B, Polio and Diphtheria will also be considered. In a nutshell, this study will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. Since the observations this dataset are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

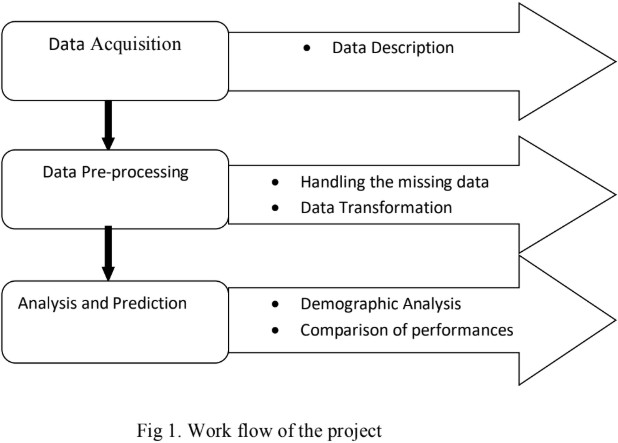
The project relies on accuracy of data. The Global Health Observatory (GHO) data repository under World Health Organization (WHO) keeps track of the health status as well as many other related factors for all countries The data-sets are made available to public for the purpose of health data analysis. The data-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation website. Among all categories of health-related factors only those critical factors were chosen which are more representative. It has been observed that in the past 15 years , there has been a huge development in health sector resulting in improvement of human mortality rates especially in the developing nations in comparison to the past 30 years. Therefore, in this project we have considered data from year 2000-2015 for 193 countries for further analysis. The individual data files have been merged together into a single data-set. On initial visual inspection of the data showed some missing values. As the data-sets were from WHO, we found no evident errors. Missing data was handled in R software by using Missmap command. The result indicated that most of the missing data was for population, Hepatitis B and GDP. The missing data were from less known countries like Vanuatu, Tonga, Togo, Cabo Verde etc. Finding all data for these countries was difficult and hence, it was decided that we exclude these countries from the final model data-set. The final merged file(final dataset) consists of 22 Columns and 2938 rows which meant 20 predicting variables. All predicting variables was then divided into several broad categories:​Immunization related factors, Mortality factors, Economical factors and Social factors.

**2.2 Proposed Solution**

The project tries to create a model based on data provided by the World Health Organization (WHO) to evaluate the life expectancy for different countries in years. The data offers a timeframe from 2000 to 2015. The data originates from here: [https://www.kaggle.com/kumarajarshi/life-expectancy-who/data](https://www.kaggle.com/kumarajarshi/life-expectancy-who/data%20). The output algorithms have been used to test if they can maintain their accuracy in predicting the life expectancy for data they haven’t been trained.

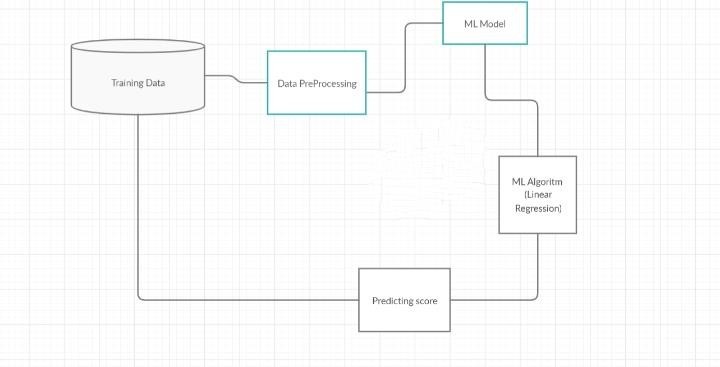
So many people were expecting to use a model of life expectancy prediction. In order to that, many institutions and companies are leading their team to build that model in this regard. In my project, I have proposed a solution to predict the life expectancy using machine learning. Machine Learning is 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 healthcare system and some specific disease related deaths that happened in the country.

**Methodology used**

Work flow for this project can be divided into three sub-tasks. These include acquiring the data and understanding various features of the data, preprocessing the data set to align it with our requirements and remove any inconsistency, and finally analyzing the data using regression based prediction algorithm with the key performance index being accuracy of prediction.

Work Flow of the project

**THEORITICAL ANALYSIS**

**3.1 Block diagram**

**3.2 Hardware / Software designing**

Project Planning and Kickoff

* Understanding the project description and analyze the data and attributes in the given dataset.
* Creating GitHub account
* Installing Slack and create account with the mail id
* Learning to use Zoho writer.

Explore IBM Cloud Platform

* Creating IBM cloud account with the mail id
* Creating IBM academic initiative account with the mail id
* a Node-Red starter application.

Explore IBM Watson services

* Exploring IBM Watson use cases.
* Learning about IBM Watson Machine Learning.

Introduction to Watson Studio

* Learning to build own Machine Learning model using IBM Watson.
* Automate the Machine Learning Model

Predicting Life Expectancy with Python

* Collecting Data set from [www.kaggle.com](http://www.kaggle.com/)
* Creating IBM Watson services
* Create a jupyter notebook and import data from Object storage.

Predicting Life Expectancy without Python

* Import data set and create AutoAI experiment
* Created Node-Red model and integrated with AutoAI

**EXPERIMENTAL INVESTIGATIONS**

# Machine Learning Model

**Algorithm Formulated to solve the given Problem Statement**

Algorithm steps:

Step 1 : Import the Data set

Step 2 : Read and Understand the data

Step3 : Explore the Data set

Step4 : Decide the amount of data for training data and testing data

Step5 : Give 70 percent data for training and remaining data for testing.

Step6 : Assign train data set to the models

Step7 : Choose the algorithm and create the model

Step8 : Make predictions for test data set.

Step9 : Calculate accuracy for the algorithm

Step 10 : Apply the model for further predictions.

**Setting Up the Environment**

An IBM cloud account was set up to access various services to create and deploy the model. The following services have been used in the project:

1. Watson Studio - This is where the notebook has been created in a project to write the regression code along with the data set.

1. Node Red - Node Red is the front end application that uses interconnecting nodes to interact with machine learning services of the cloud and the model to show predictions when inputs are given

## Data Acquisition

### The data set on Statistical Analysis on factors influencing Life Expectancy is provided in the smart internz platform . The below link helps in downloading the data set required to complete this project

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

## Model Requirements

**Python**

Python is a multi-paradigm, general purpose, high level programming language, which focuses on code read- ability. It has a large library, which provides tools for many tasks and has a wide support base. This project uses python 3.5.

**Python Libraries**

**Pandas :** Pandas is used for data manipulation and analysis through operations and data structures on numerical tables and time series.

**Numpy :** It adds support as well as contains high-level mathematical functions to operate on large multidimensional arrays and matrices.

**Matplotlib :** It is a plotting library that that enables 2d diagramming and begetting of bar charts, histograms and so forth.

**Sci-kit learn :** It is a free software machine learning library that features various regression, clustering and classification algorithms. It works in conjunction with numPy and python scientific library sciPy.

**Data Preprocessing**

Data preprocessing is an essential step in order to increase the accuracy of machine learning models. It involves handling inaccurate and missing data, noisy data in the form of outliers, and inconsistent data in the form of duplication and others.

**Data Cleaning**

Data was often not consistent; missing values or values out of range was common. The methods used for cleaning is to replace the missing or noisy values by forward filling them using mean of the feature.

**Analysis and Prediction**

Linear Regression and Random Forest algorithm has been applied to the data set to train the model and increase the accuracy for prediction of the the life expectancy of any given country.

**Data Transformation**

**Scaling :** Scaling is required to standardize the independent feature in the dataset to a fixed range. Primarily, two types of feature scaling methods:

1. Min-max scaling (Normalization) (value - min)/(max - min) Sklearn provides a class called MinMaxScaler for this
2. Standardization (value - mean)/std Sklearn provides a class called StandardScaler for this

**Evaluation Metrics**

Evaluation metrics calculated are:

1. Mean Cross validation score
2. Score without cv
3. R2*score*

**Linear Regression**

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

**Random Forest Regression**

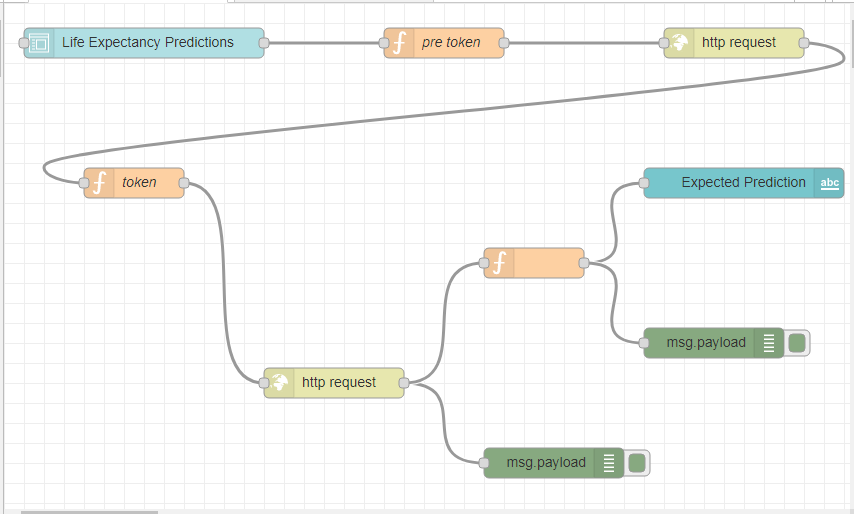
Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

It technically is an ensemble method (based on the divide-and-conquer approach) of decision trees gener- ated on a randomly split dataset. This collection of decision tree classifiers is also known as the forest. The individual decision trees are generated using an attribute selection indicator such as information gain, gain ra- tio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and the most popular class is chosen as the final result. In the case of regression, the average of all the tree outputs is considered as the final result. It is simpler and more powerful compared to the other non-linear classification algorithms.

**Node Red Flow**

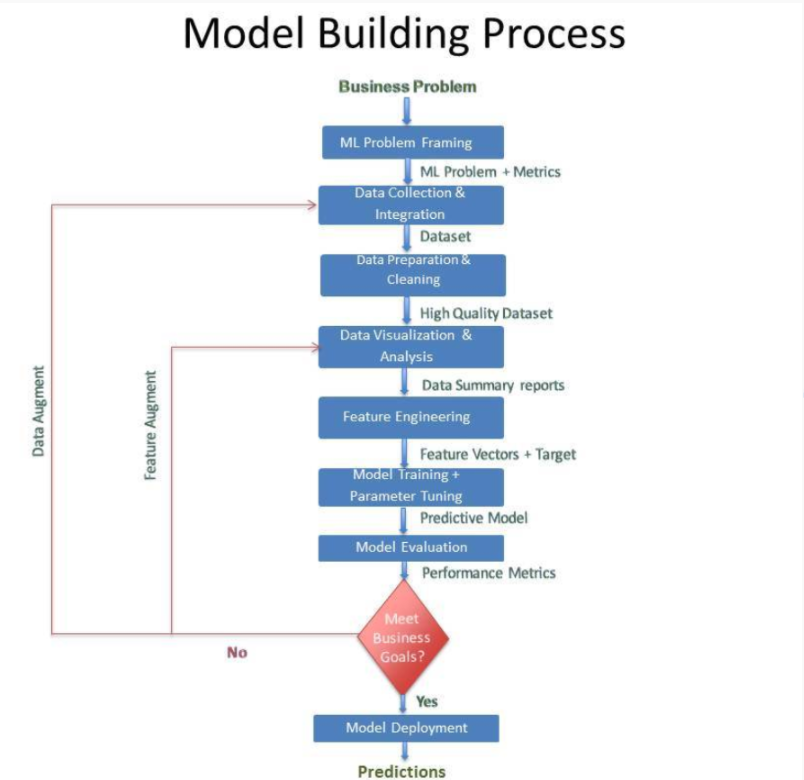
A Node RED starter application was created to implement the front end of the project. In the starter appli- cation, nodes are dragged and dropped to create a flow to integrate the application with the machine learning model.

The following form appears after deployment of the app wherein the user can input values and life expectancy prediction is displayed acccording to the inputs.



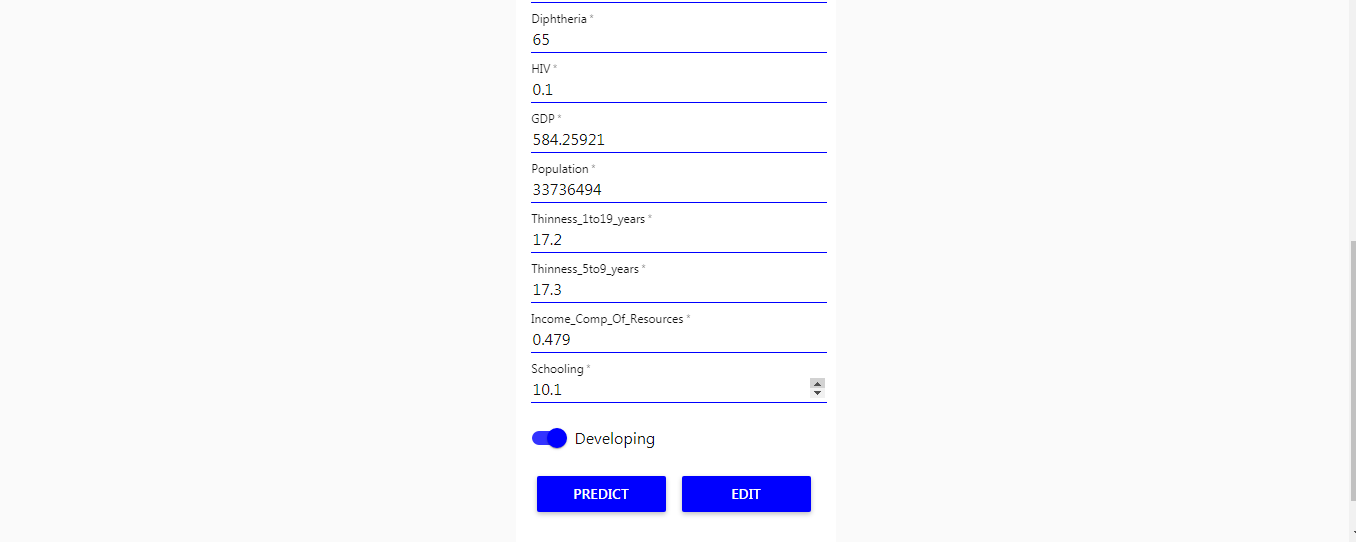
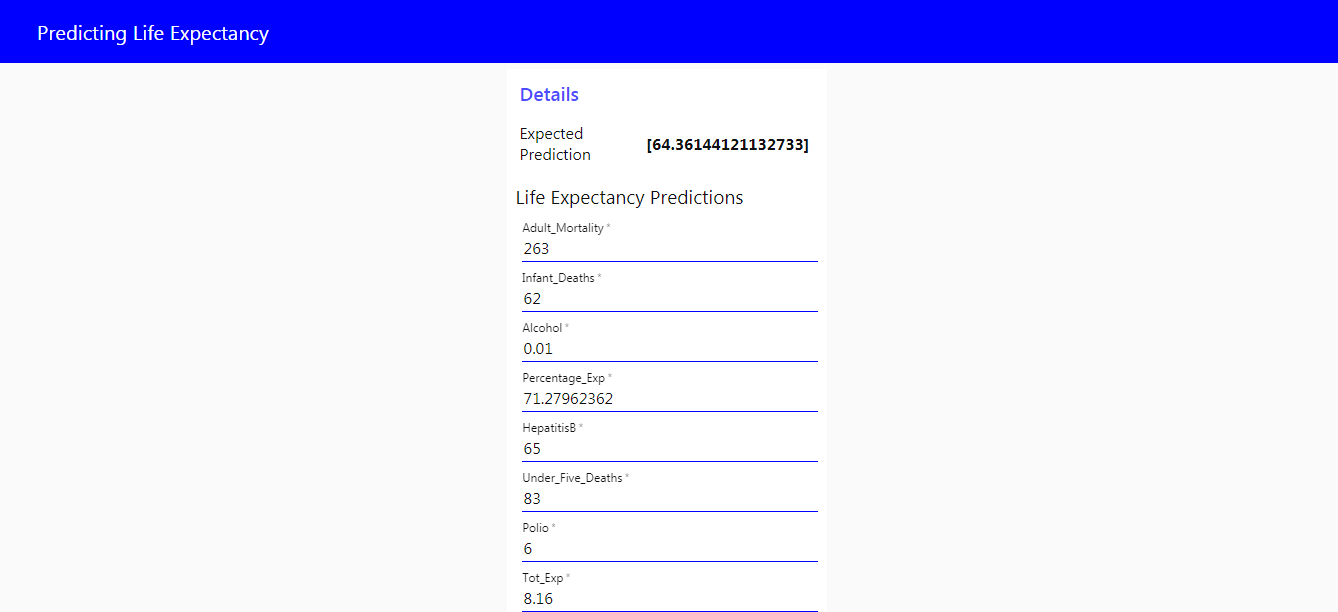
NodeRed flow of the project

**FLOWCHART**

****

**RESULT**

**The final UI of the project**



**ADVANTAGES & DISADVANTAGES**

## Every coin has two faces, each face has its own property and features. It’s time to uncover the faces of ML. A very powerful tool that holds the potential to revolutionize the way things work.

### Advantages

1.Advantages of using IBM Watson:

* Processes unstructured data
* Fills human limitations
* Acts as a decision support system, doesn’t replace humans
* Improves performance + abilities by giving best available data
* Improve and transform customer service
* Handle enormous quantities of data
* Sustainable Competitive Advantage

2. Easy for user to interact with the model via the UI.

3. User-friendly

4. Easy to build and deploy.

5. Doesn’t require much storage space.

### Disadvantages

1.Disadvantages of using IBM Watson

* Seen as disruptive technology
* Maintenance
* Doesn't process structured data directly
* Increasing rate of data, with limited

2. Not connected to database, hence no record of input.

3. Requires internet connect

**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**

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.

**CONCLUSION**

I have developed a model that will predict the life expectancy of a specific demographic region based on the inputs provided. Various factors have a significant impact on the life span such as Adult Mortality, Population, Under 5 Deaths, Thinness 1-5 Years, and Alcohol, HIV, Hepatitis B, GDP, Percentage Expenditure and many more. User can interact with the system via a simple user interface which is in the form of a form with input spaces which the user needs to fill the inputs into.

This was my first internship experience, through this I was able to enhance my Machine Learning skills. Though I am a beginner in programming , with the help of the platform provided , It was an enriching experience as I got to work in a professional like environment. The mentors were very helpful with the webinars they conducted on how to proceed with the project. Our doubts were solved on the slack channel regularly.

I particularly found the IBM cloud experience new and useful in improving my industrial skills. The challenging task in this internship is in creating the Node Red flow as a beginner I have faced difficulties but with the help of the mentors through slack channel platform I succeeded in my project , I found it to be valuable and productive in developing my front end integration skills.

This internship has given me a transparent idea on how to proceed with improving my machine learning skills and I am grateful to Smart Bridge for letting me be a part of the Remote summer internship.

**Finally I can proudly say that my lockdown period was very productive.**

**BIBILOGRAPHY**

**Node-RED Starter Application**

<https://developer.ibm.com/tutorials/how-to-create-a->[node-red-starter-application/](https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/)

**Watson studio cloud**

[https://bookdown.org/caoying4work](https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html)

**Dataset Reference**

[[https://www.kaggle.com/kumaraja rshi/life-expectancy-who](https://www.kaggle.com/kumarajarshi/life-expectancy-who)](https://www.kaggle.com/kumarajarshi/life-expectancy-who)

**IBM cloud services**

<https://www.youtube.com/watch?v=DBRGlAHdj48&amp;list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L>

**Import the data set into the jupyter notebook**

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

**APPENDIX**

Source code : Machine Learning Model

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

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

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

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

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

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

**import** **warnings**

warnings.filterwarnings('ignore')

%matplotlib inline

**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\_923e8e384f264883a1ce9afc6be73e3b = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='kVvlaW0F-JK4Kw8MjI6KZi0DyeqxCcK2BKRPyuqIZ1Z2',

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\_923e8e384f264883a1ce9afc6be73e3b.get\_object(Bucket='predictinglifeexpectancyusingmach-donotdelete-pr-l1e3nrmukdnsd9',Key='Life\_Expectacy\_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 )

*# If you are reading an Excel file into a pandas DataFrame, replace `read\_csv` by `read\_excel` in the next statement.*

life = pd.read\_csv(body)

life.head()

life.size

life.shape

life.columns

life.describe()

life.info()

life.isnull().sum()

country\_list = life.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']

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

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

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

life.isna().sum()

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 1-19 years' :15, ' thinness 5-9 years' :16,

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

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

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

plt.subplot(5,4,i)

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

plt.title(variable)

plt.show()

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

q75, q25 = np.percentile(life[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((life[variable] > max\_val) | (life[variable] < min\_val))[0])),

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

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Life\_Expectancy = life['Life expectancy ']

plt.boxplot(original\_Life\_Expectancy)

plt.title("original\_Life\_Expectancy")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Life\_Expectancy)

plt.title("winsorized\_Life\_Expectancy")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Adult\_Mortality = life['Adult Mortality']

plt.boxplot(original\_Adult\_Mortality)

plt.title("original\_Adult\_Mortality")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Adult\_Mortality)

plt.title("winsorized\_Adult\_Mortality")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Infant\_Deaths = life['infant deaths']

plt.boxplot(original\_Infant\_Deaths)

plt.title("original\_Infant\_Deaths")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Infant\_Deaths)

plt.title("winsorized\_Infant\_Deaths")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Alcohol = life['Alcohol']

plt.boxplot(original\_Alcohol)

plt.title("original\_Alcohol")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Alcohol)

plt.title("winsorized\_Alcohol")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Percentage\_Exp = life['percentage expenditure']

plt.boxplot(original\_Percentage\_Exp)

plt.title("original\_Percentage\_Exp")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Percentage\_Exp)

plt.title("winsorized\_Percentage\_Exp")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_HepatitisB = life['Hepatitis B']

plt.boxplot(original\_HepatitisB)

plt.title("original\_HepatitisB")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_HepatitisB)

plt.title("winsorized\_HepatitisB")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Measles = life['Measles ']

plt.boxplot(original\_Measles)

plt.title("original\_Measles")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Measles)

plt.title("winsorized\_Measles")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Under\_Five\_Deaths = life['under-five deaths ']

plt.boxplot(original\_Under\_Five\_Deaths)

plt.title("original\_Under\_Five\_Deaths")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Under\_Five\_Deaths)

plt.title("winsorized\_Under\_Five\_Deaths")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Polio = life['Polio']

plt.boxplot(original\_Polio)

plt.title("original\_Polio")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Polio)

plt.title("winsorized\_Polio")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Tot\_Exp = life['Total expenditure']

plt.boxplot(original\_Tot\_Exp)

plt.title("original\_Tot\_Exp")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Tot\_Exp)

plt.title("winsorized\_Tot\_Exp")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Diphtheria = life['Diphtheria ']

plt.boxplot(original\_Diphtheria)

plt.title("original\_Diphtheria")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Diphtheria)

plt.title("winsorized\_Diphtheria")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_HIV = life[' HIV/AIDS']

plt.boxplot(original\_HIV)

plt.title("original\_HIV")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_HIV)

plt.title("winsorized\_HIV")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_GDP = life['GDP']

plt.boxplot(original\_GDP)

plt.title("original\_GDP")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_GDP)

plt.title("winsorized\_GDP")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Population = life['Population']

plt.boxplot(original\_Population)

plt.title("original\_Population")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Population)

plt.title("winsorized\_Population")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_thinness\_1to19\_years = life[' thinness 1-19 years']

plt.boxplot(original\_thinness\_1to19\_years)

plt.title("original\_thinness\_1to19\_years")

plt.subplot(1,2,2)

winsorized\_thinness\_1to19\_years = winsorize(life[' thinness 1-19 years'],(0,0.04))

plt.boxplot(winsorized\_thinness\_1to19\_years)

plt.title("winsorized\_thinness\_1to19\_years")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_thinness\_5to9\_years = life[' thinness 5-9 years']

plt.boxplot(original\_thinness\_5to9\_years)

plt.title("original\_thinness\_5to9\_years")

plt.subplot(1,2,2)

winsorized\_thinness\_5to9\_years = winsorize(life[' thinness 5-9 years'],(0,0.04))

plt.boxplot(winsorized\_thinness\_5to9\_years)

plt.title("winsorized\_thinness\_5to9\_years")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Income\_Comp\_Of\_Resources = life['Income composition of resources']

plt.boxplot(original\_Income\_Comp\_Of\_Resources)

plt.title("original\_Income\_Comp\_Of\_Resources")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Income\_Comp\_Of\_Resources)

plt.title("winsorized\_Income\_Comp\_Of\_Resources")

plt.show()

plt.figure(figsize=(18,6))

plt.subplot(1,2,1)

original\_Schooling = life['Schooling']

plt.boxplot(original\_Schooling)

plt.title("original\_Schooling")

plt.subplot(1,2,2)

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

plt.boxplot(winsorized\_Schooling)

plt.title("winsorized\_Schooling")

plt.show()

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\_1to19\_years,winsorized\_thinness\_5to9\_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])))

life['winsorized\_Life\_Expectancy'] = winsorized\_Life\_Expectancy

life['winsorized\_Adult\_Mortality'] = winsorized\_Adult\_Mortality

life['winsorized\_Infant\_Deaths'] = winsorized\_Infant\_Deaths

life['winsorized\_Alcohol'] = winsorized\_Alcohol

life['winsorized\_Percentage\_Exp'] = winsorized\_Percentage\_Exp

life['winsorized\_HepatitisB'] = winsorized\_HepatitisB

life['winsorized\_Under\_Five\_Deaths'] = winsorized\_Under\_Five\_Deaths

life['winsorized\_Polio'] = winsorized\_Polio

life['winsorized\_Tot\_Exp'] = winsorized\_Tot\_Exp

life['winsorized\_Diphtheria'] = winsorized\_Diphtheria

life['winsorized\_HIV'] = winsorized\_HIV

life['winsorized\_GDP'] = winsorized\_GDP

life['winsorized\_Population'] = winsorized\_Population

life['winsorized\_thinness\_1to19\_years'] = winsorized\_thinness\_1to19\_years

life['winsorized\_thinness\_5to9\_years'] = winsorized\_thinness\_5to9\_years

life['winsorized\_Income\_Comp\_Of\_Resources'] = winsorized\_Income\_Comp\_Of\_Resources

life['winsorized\_Schooling'] = winsorized\_Schooling

life['winsorized\_Measles'] = winsorized\_Measles

life.head()

life.size

life.shape

life.describe()

all\_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 1-19 years','winsorized\_thinness\_1to19\_years',' thinness 5-9 years',

'winsorized\_thinness\_5to9\_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(all\_col)):

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

plt.hist(life[all\_col[i]])

plt.title(all\_col[i])

plt.show()

life.describe(include= 'O')

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

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

plt.xlabel("Status",fontsize=12)

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

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

plt.show()

le\_country = life.groupby('Country')['winsorized\_Life\_Expectancy'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Life\_Expectancy w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Life\_Expectancy",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_GDP'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Average GDP w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg GDP",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Adult\_Mortality'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Adult\_Mortality w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Adult Mortality",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Alcohol'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Alcohol w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Alcohol Comsumption",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Diphtheria'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Diphtheria w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Diphtheria",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_HepatitisB'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("HepatitisB w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg HepatitisB",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_HIV'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("HIV w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg HIV cases",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Income\_Comp\_Of\_Resources'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Income Composition of Resources w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg income composition of resourses",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Infant\_Deaths'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Infant Deaths w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Infant Deaths",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Measles'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Measles w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Measles cases",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Percentage\_Exp'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Percentage Expenditure w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg percentage expenditure",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Polio'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Polio w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Polio Cases",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Population'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Population w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Population",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Schooling'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Schooling w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Schooling",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_thinness\_1to19\_years'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title(" Thinness 1to19 years w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Thinness 1 to 19 Years",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_thinness\_5to9\_years'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title("Thinness 5 to 9 years w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg thinness 5 to 9 years ",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Tot\_Exp'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title(" Total Expenditure w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg Total Expenditure",fontsize=35)

plt.show()

le\_country = life.groupby('Country')['winsorized\_Under\_Five\_Deaths'].mean().sort\_values(ascending=**True**)

le\_country.plot(kind='bar', figsize=(50,15), fontsize=25)

plt.title(" Under five Deaths w.r.t Country",fontsize=40)

plt.xlabel("Country",fontsize=35)

plt.ylabel("Avg under 5 deaths",fontsize=35)

plt.show()

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

plt.bar(life.groupby('Year')['Year'].count().index,life.groupby('Year')['winsorized\_Life\_Expectancy'].mean(),color='green',alpha=0.65)

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

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

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

plt.show()

corr= life.corr()

sns.heatmap(corr)

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

plt.subplot(6,3,1)

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

plt.title("LifeExpectancy vs AdultMortality")

plt.subplot(6,3,2)

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

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

plt.subplot(6,3,3)

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

plt.title("LifeExpectancy vs Alcohol")

plt.subplot(6,3,4)

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

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

plt.subplot(6,3,5)

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

plt.title("LifeExpectancy vs HepatitisB")

plt.subplot(6,3,6)

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

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

plt.subplot(6,3,7)

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

plt.title("LifeExpectancy vs Polio")

plt.subplot(6,3,8)

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

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

plt.subplot(6,3,9)

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

plt.title("LifeExpectancy vs Diphtheria")

plt.subplot(6,3,10)

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

plt.title("LifeExpectancy vs HIV")

plt.subplot(6,3,11)

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

plt.title("LifeExpectancy vs GDP")

plt.subplot(6,3,12)

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

plt.title("LifeExpectancy vs Population")

plt.subplot(6,3,13)

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

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

plt.subplot(6,3,14)

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

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

plt.subplot(6,3,15)

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

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

plt.subplot(6,3,16)

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

plt.title("LifeExpectancy vs Schooling")

plt.show()

round(life[['Status','Life expectancy ']].groupby(['Status']).mean(),2)

stats.ttest\_ind(life.loc[life['Status']=='Developed','Life expectancy '],life.loc[life['Status']=='Developing','Life expectancy '])

feature\_df = life[['Status','winsorized\_Life\_Expectancy','winsorized\_Income\_Comp\_Of\_Resources','winsorized\_HIV','winsorized\_Adult\_Mortality']]

feature\_df = pd.concat([feature\_df,pd.get\_dummies(feature\_df['Status'],drop\_first=**True**)],axis=1)

final = feature\_df.drop('Status',axis=1)

final.sample(10)

life = pd.concat([life,pd.get\_dummies(life['Status'],drop\_first=**True**)],axis=1)

life = life.drop('Status',axis=1)

life.sample(10)

life.columns

X= life[['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\_1to19\_years', 'winsorized\_thinness\_5to9\_years',

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

'winsorized\_Measles', 'Developing']]

Y = life['winsorized\_Life\_Expectancy']

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

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state= 42)

print(X\_train.shape)

print(Y\_train.shape)

print(X\_test.shape)

print(Y\_test.shape)

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

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

model = LinearRegression(fit\_intercept=**True**, normalize=**True**).fit(X\_train, Y\_train)

predictions= model.predict(X\_test)

len(predictions)

predictions.shape

r2\_score(predictions, Y\_test)

mean\_squared\_error(predictions, Y\_test)

mean\_absolute\_error(predictions, Y\_test)

X = life[['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\_1to19\_years', 'winsorized\_thinness\_5to9\_years',

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

'Developing']]

Y = life['winsorized\_Life\_Expectancy']

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state= 42)

model = LinearRegression(fit\_intercept=**True**, normalize=**True**).fit(X\_train, Y\_train)

predictions= model.predict(X\_test)

len(predictions)

r2\_score(predictions, Y\_test)

mean\_squared\_error(predictions, Y\_test)

mean\_absolute\_error(predictions, Y\_test)

!pip install watson-machine-learning-client

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

client = WatsonMachineLearningAPIClient( wml\_credentials )

model\_props = {client.repository.ModelMetaNames.AUTHOR\_NAME: "Aditya Reddy Lellapati",

client.repository.ModelMetaNames.AUTHOR\_EMAIL: "19311a04c8@sreenidhi.edu.in",

client.repository.ModelMetaNames.NAME: "Life\_Expectancy"}

model\_artifact =client.repository.store\_model(model, meta\_props=model\_props)

published\_model\_uid = client.repository.get\_model\_uid(model\_artifact)

published\_model\_uid

deployment = client.deployments.create(published\_model\_uid, name="Life\_Expectancy")

scoring\_endpoint = client.deployments.get\_scoring\_url(deployment)

scoring\_endpoint