**Internship Project Report**

**On**

**Machine Learning**



**Topic: Predicting Life Expectancy using Machine Learning**

**Company : SMARTBRIDGE**

**Duration Of Internship : 4 Weeks**

**Submitted By:**

**Paras Thareja**

**(BVCOE College)**

**ACKNOWLEDGEMENT**

I express my deep gratitude to **Mr. Hemant , Mr Prashanth , Mr Charan and Ms. Lalitha**

for their valuable guidance and suggestions throughout my project work.

Last but not the least I would like to thank my friends, family members and all those people who helped me for the completion and deeper understanding of the concept of machine learning. Working on this project was an enlightening experience for me.

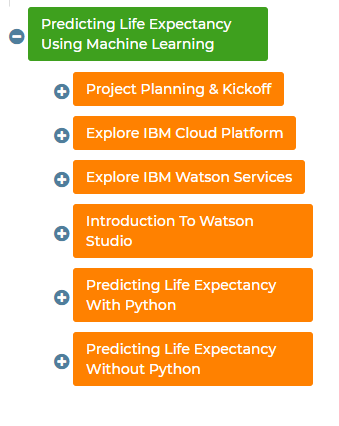
**Sign**

Paras Thareja,

BVCOE, Delhi

**PROJECT SCHEDULE:**

**This Project Schedule Was Provided to us Through the SmartInternz Platform :**



1. **Introduction:**

**Project Overview:**

Life Expectancy :

This can be simply defined as the expected lifetime period till which a person will survive.

There are various factors that affect life expectancy of a person. Most of them I have included in the project (Like : Adult Mortality, Polio level, Under Five Deaths etc.)

This factors have a linear results with the Life Expectancy that is to be predicted. Also, since the Life Expectancy that is to be predicted is a continous range of values, therefore, for this problem Regression was used.

There are multiple regressions that are used. The ones I tried on my project were:

* Multiple Linear Regression
* Lasso Regression
* Ridge Regression
* SVM

The project was deployed in the form of Web interface via Node-Red service of IBM Cloud, which is a very good option due to it's interactive surface.

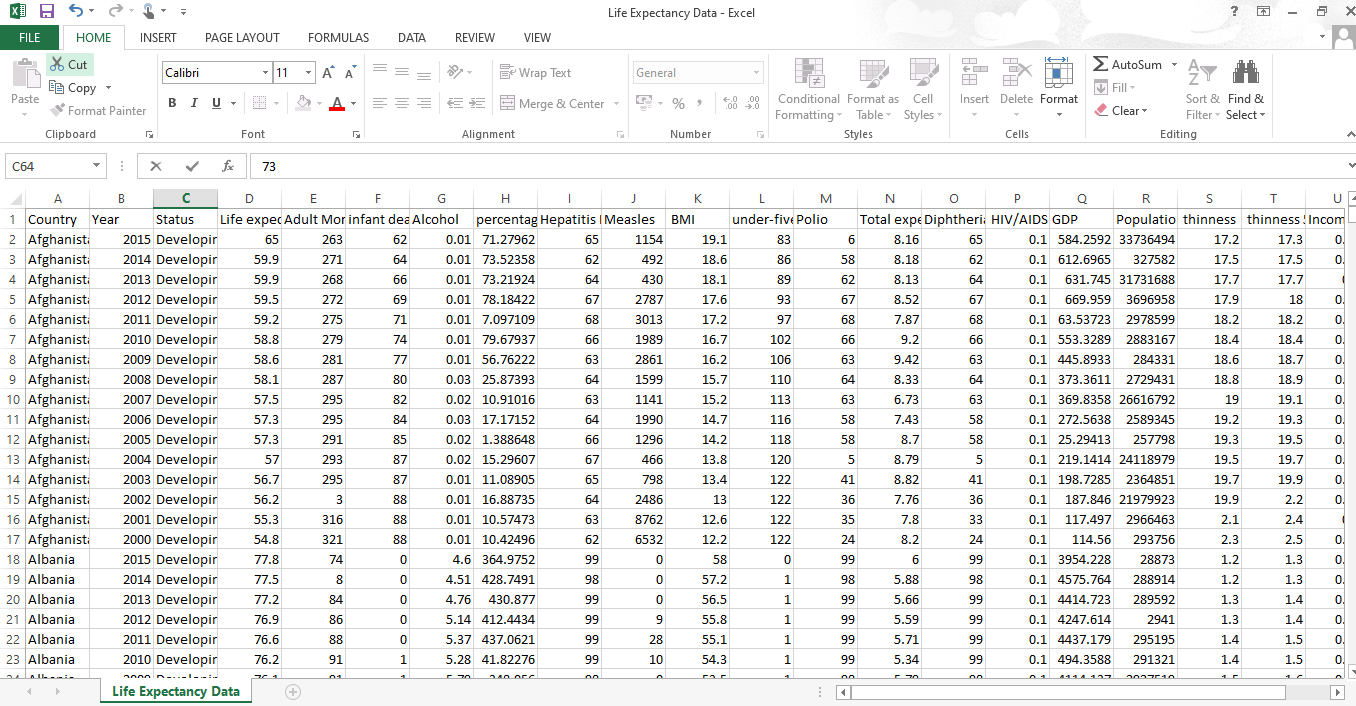
The deliverable of the project is an interactive interface instead of simple results in the form of python output or some web interface.

**About The Dataset :**

In this project our aim is to build a new model based on the data provided is to evaluate the life expectancy in India.

The data offers a timeframe from 2015 to 2022. The output algorithms have been used to test if model to be built has accuracy in predicting the life expectancy for data they haven't been trained.

First Look at The Dataset :



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 datasets are made available to public for the purpose of health data analysis. The dataset 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 dataset. On initial visual inspection of the data showed some missing values. As the datasets 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 dataset. 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.

**Problem Statement:**

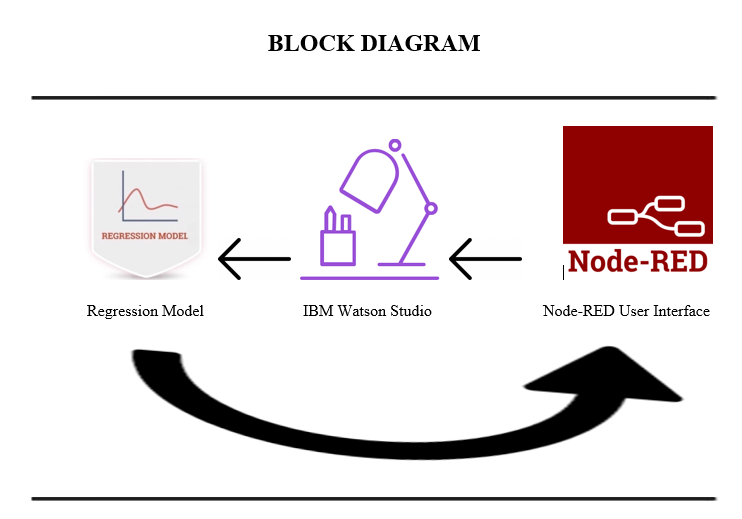
Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that affect of immunization and human development index was not taken into account in the past. Also, some of the 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.

**GOAL OF THE PROJECT**

The main goals of the specified projects are :

* How to develop a test harness to develop a robust evaluation of a model and establish a baseline of performance for this task.
* How to explore extensions to a baseline model to improve learning and model capacity.
* How to develop a finalized model, evaluate the performance of the final model, and
* use it to make predictions on data.

**BODY:**



**Python basics:**

Important Characters and Sets of Characters

> tab - \t > new line - \n

> backslash- \\ > string- " " or ' '

> comparison operators- == , < , > , <= , >= , !=

> Python type boolean- True , False.

> Logical operators- not , and , or

Six Steps to Defining a Function

1. What should your function do? Type a couple of example calls.

2. Pick a meaningful name (often a verb or verb phrase): What is a short answer to "What does your function do"?

3. Decide how many parameters the function takes and any return values

4. Describe what your function does and any parameters and return values in the docstring

5. Write the body of the function 6. Test your function. Think about edge cases.

Integers and Strings

>>> int(45)

45

>>> int('45')

45

>>> str(45)

'45'

>>> str('45')

'45'

Conditionals and Branching

if elif else

> We have a boolean logic expression for if which works when the Boolean evaluates to True

String Indexing and Slicing

(s[a:b] means index a to length (b-a) or a to b index but not including b)

s[2:3] , s[0], s[:5] , S[4:]

String is immutable (ex. s[4]='a' will not replace 'a' and index 4 of s)

String Methods

> A method is a function inside of an object.

> The general form of a method call is:

. object.method(arguments)

. dir(str)

. help(str.method)

**STEPS OF MACHINE LEARNING-**

Machine learning has several steps , any project which is started consist of the following processes :

**1. Collecting the data**

Be it the raw data from excel, access, text files, images, video etc., this step (gathering past data) forms the foundation of future learning. The better the variety, density and volume of relevant data, the better the learning prospects for the machine becomes.

**2. Preparing the data**

Bad data always lead to bad insights that leads to problems. Any analytical process thrives on the quality of the data used. One needs to spend time determining the quality of data and then taking steps for fixing issues such as missing data and treatment of outliers.

**3. Training the model**

This step involves choosing the appropriate algorithm and representation of data in the form of the model. In layman terms model representation is a process to represent our real-life problem statement into a mathematical model for the computer to understand. The cleaned data is split into three parts – Training, Validation and Test - proportionately depending on the scenario. The training part is then given to the model to learn the relationship / function

**4. Evaluating the model**

Quite often, we don’t train just one model but many. So, to compare the performance of the different models, we evaluate all these models on the validation data. As it has not been seen by any of the models, validation data helps us evaluate the real-world performance of models.

**5. Improving the Performance**

Often, the performance of the model is not satisfactory at first and hence we need to revisit earlier choices we made in deciding data representations and model parameters. We may choose to use different variables (features) or even collect some more data. We might need to change the whole architecture to get the better performance in the worst case.

**6. Report the Performance**

Once we are satisfied by the performance of the model on the validation set, we evaluate our chosen model on the test set and this provides us with a fair idea of the performance of our model on real-world data that it has not seen before.

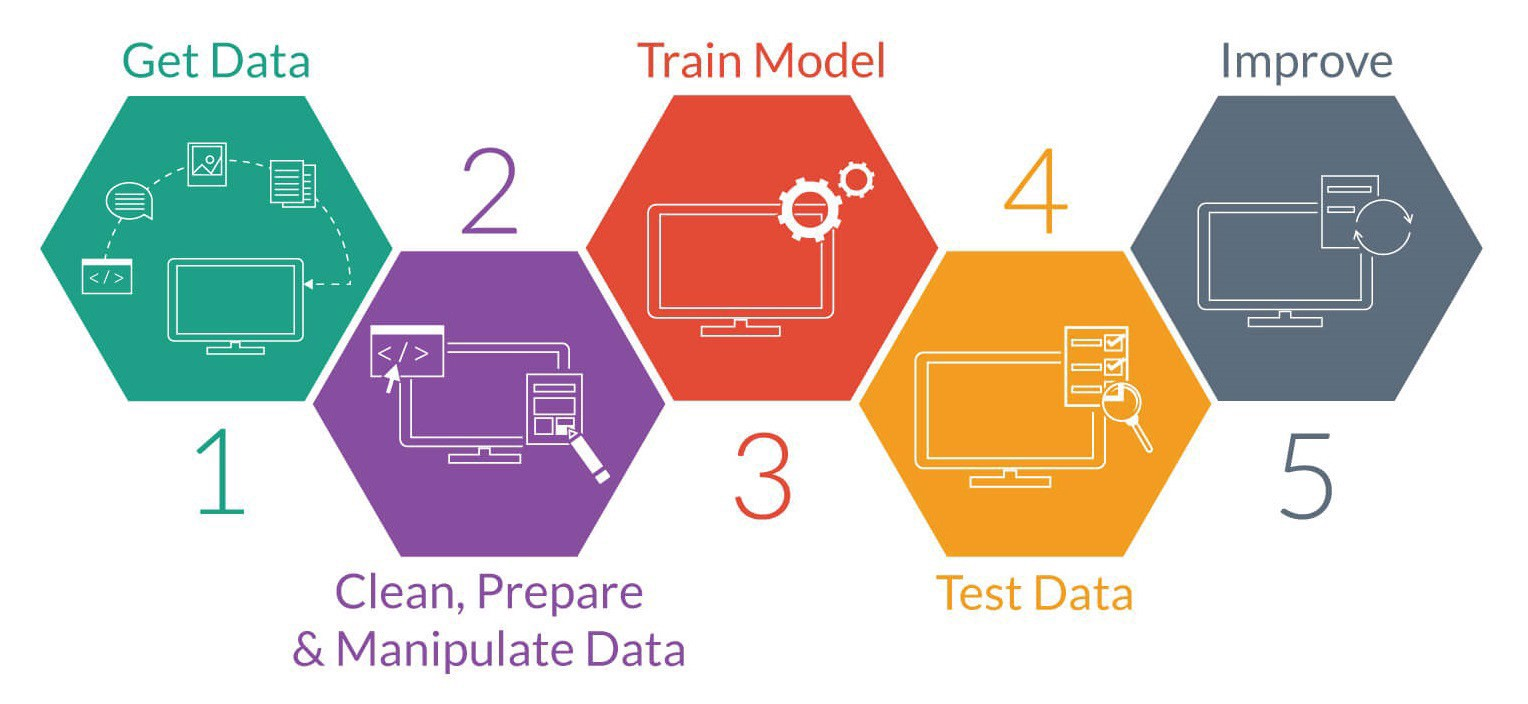


Figure 3 : Steps Of Machine Learning

**SOME PYTHON LIBRARIES FOR MACHINE LEARNING:**

Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. A more general definition given by Arthur Samuel is – “Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.” They are typically used to solve various types of life problems.

**NUMPY:**

Numpy is a general-purpose array-processing package. It provides a high-performance

multidimensional array object, and tools for working with these arrays. It is the fundamental

package for scientific computing with Python.

Array in Numpy is a table of elements , all of the same type, indexed by a

tuple of positive integers. In Numpy, number of dimensions of the array is called the rank of

the array.A tuple of integers giving the size of the array along each dimension is known as

shape of the array. An array class in Numpy is called as ndarray. Elements in Numpy arrays

are accessed by using square brackets and can be initialized by using nested Python Lists**.**

**PANDAS:**

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and a wide variety of tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

**MATPLOTLIB:**

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar charts, etc.

**SCIKIT-LEARN:**

[Scikit-learn](http://scikit-learn.org/stable/user_guide.html) is one the most popular ML libraries. It supports many supervised and unsupervised learning algorithms. Examples include linear and logistic regressions, decision trees, clustering, k-means and so on.

It builds on two basic libraries of Python, NumPy and SciPy. It adds a set of algorithms for common machine learning and data mining tasks, including clustering, regression and classification. Even tasks like transforming data, feature selection and ensemble methods can be implemented in a few lines.

**The following machine Learning Algorithms implemented during our summer training:**

1. **LINEAR REGRESSION**
2. **LASSO REGRESSION**
3. **RIDGE REGRESSION**
4. **SUPPORT VECTOR MACHINE**
5. **Creating IBM Account :**

The Project mainly used IBM Cloud services . Project was done both , with python as well as without python.

Step-1 was the creation of IBM Account.

The IBM® cloud platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud™ spins up fast and performs reliably in a tested and supported environment you can trust.

The platform is built to support your needs whether it's working only in the public cloud or taking advantage of a multicloud deployment model. With our open-source technologies, such as Kubernetes, Red Hat OpenShift, and a full range of compute options, including virtual machines, containers, bare metal, and serverless, you have as much control and flexibility as you need to support workloads in your hybrid environment. You can deploy cloud-native apps while also ensuring workload portability.

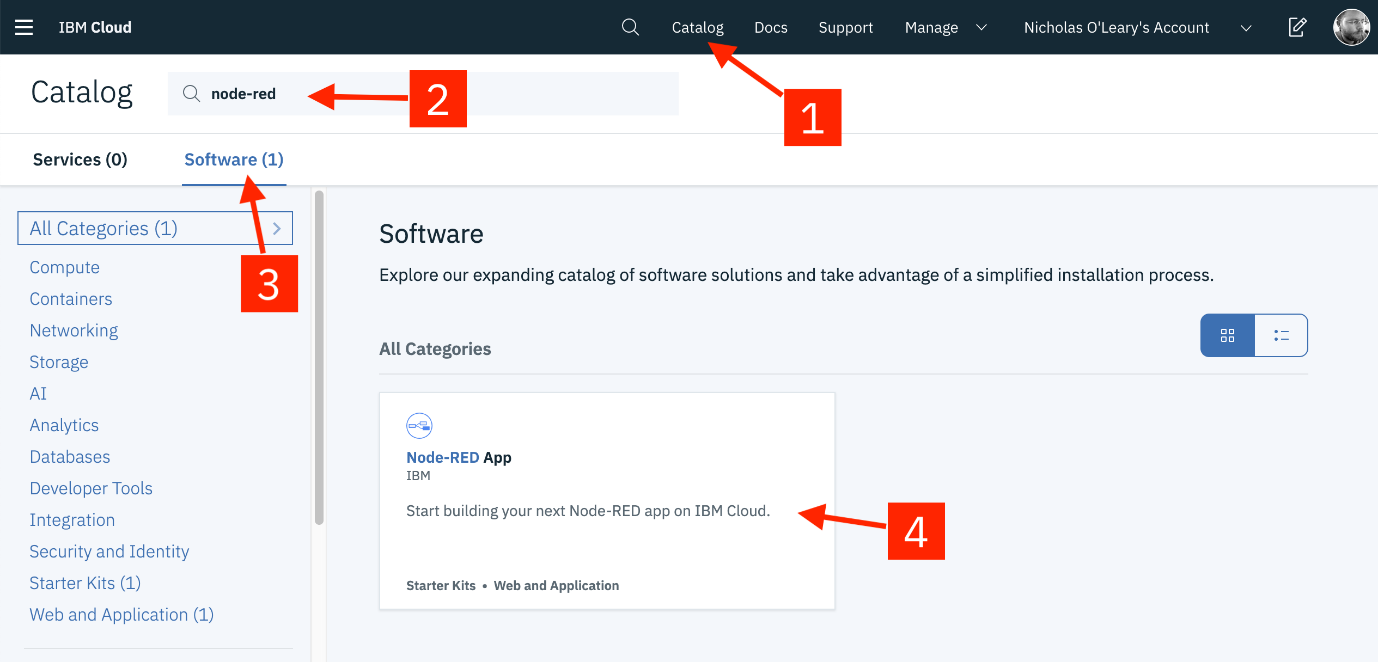
Link for creation of IBM Account is [2].

1. **Creating Node-red Starter Application:**

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

**Steps To Create Node-Red Service:**

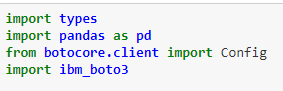


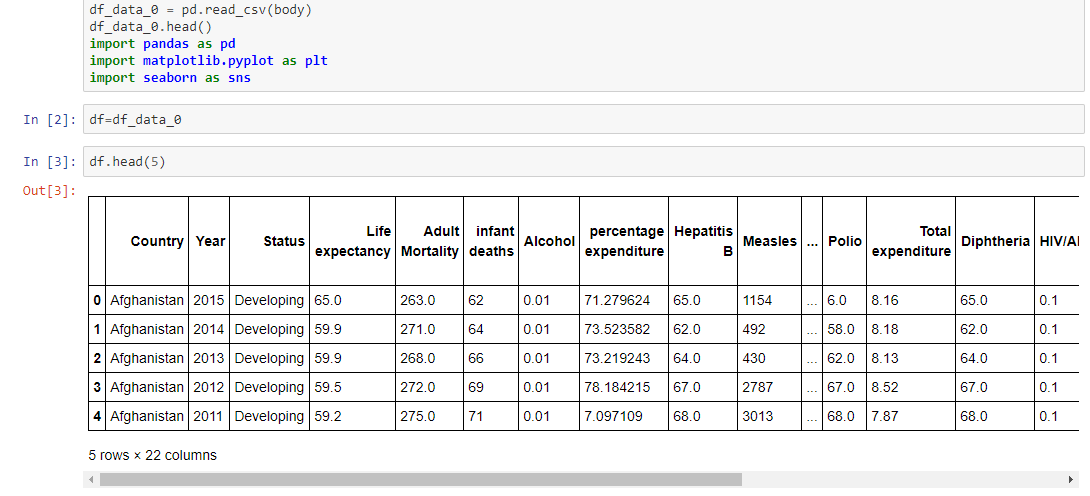
1. After logging in, from the catalog, search for Node-red.
2. Tap on Node-red and create it.
3. Click on Deploy .
4. Go to “IBM CLOUD DASHBOARD” , from “Cloud Foundry “, open Node-red.
5. Click on “Visit App URL”.
6. Click on “Go to Your NODE-RED flow editor”.

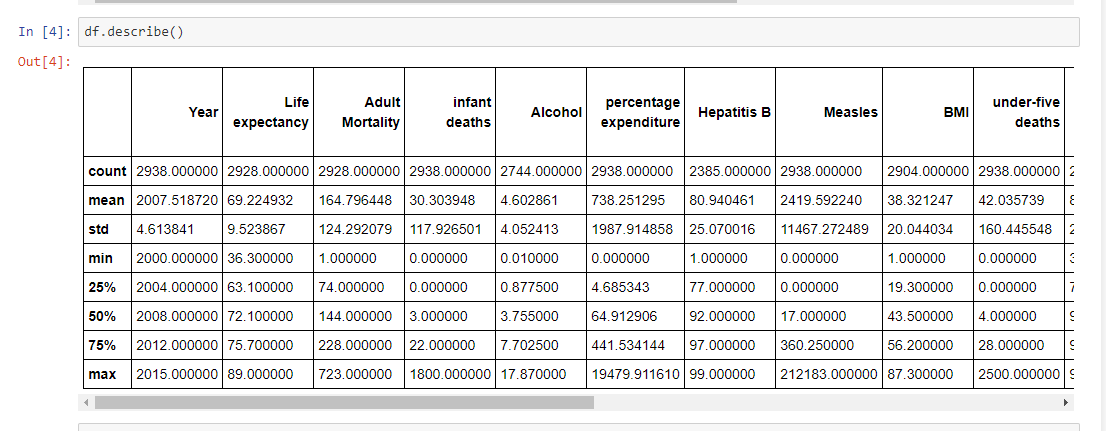
Then, I Explored Node-red services , created a web-page and about API.

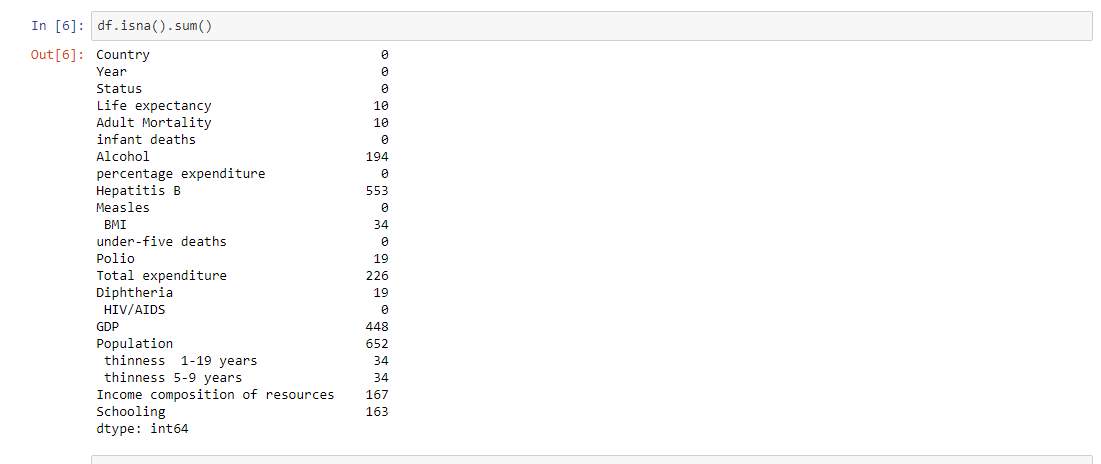
1. **Predicting Machine Learning With Python:**

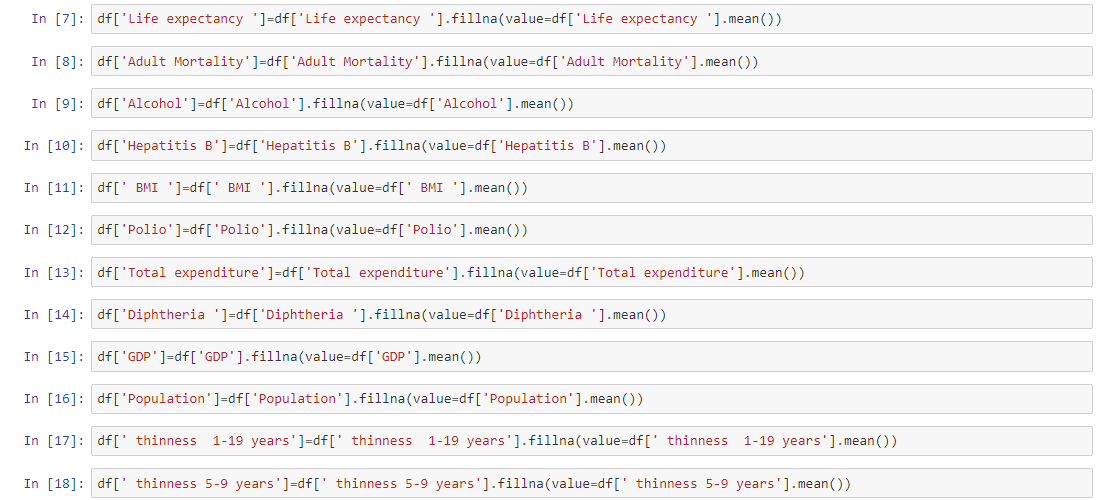
**CODE:**

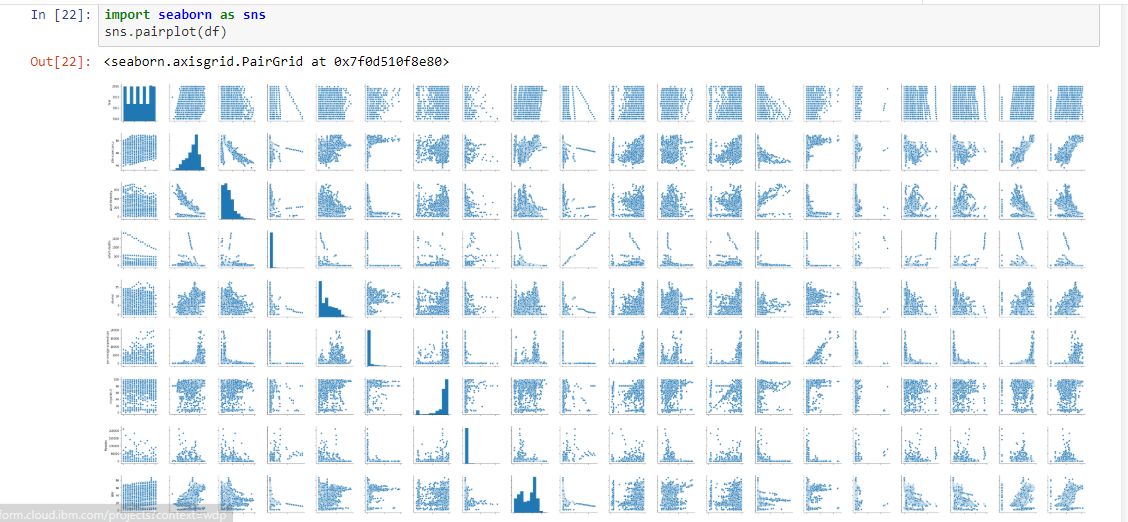


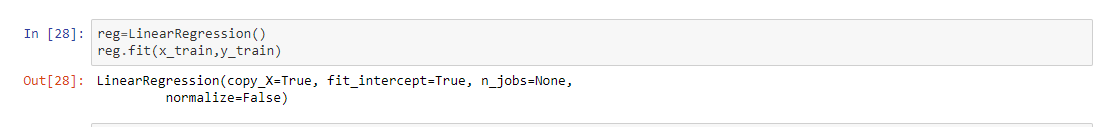
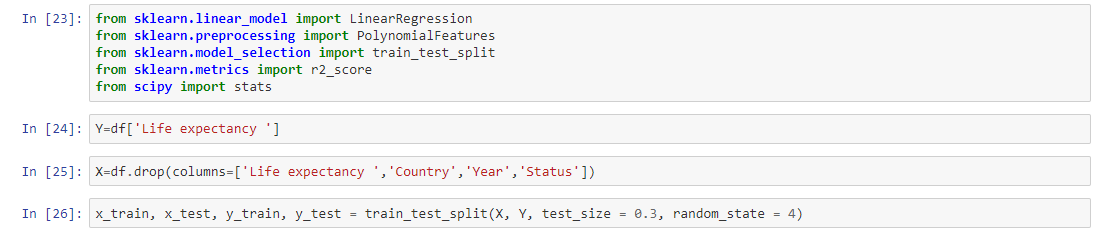


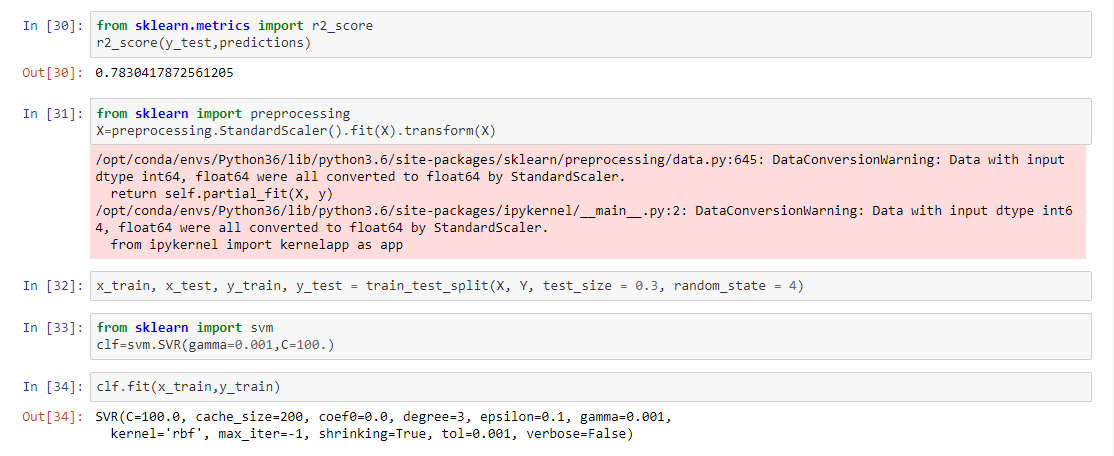












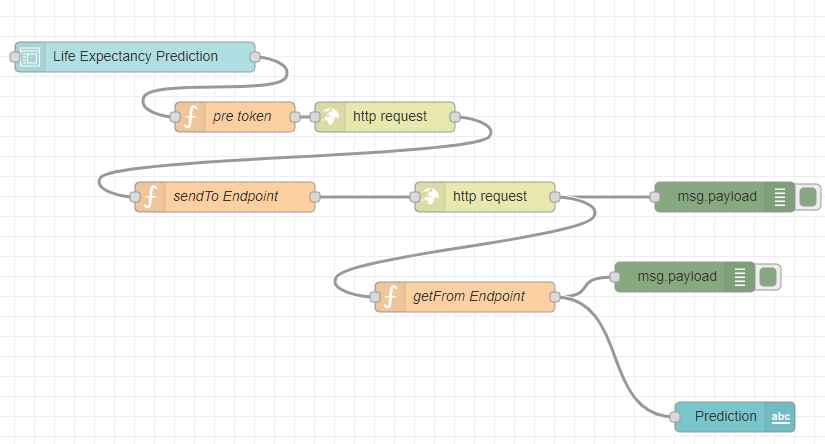




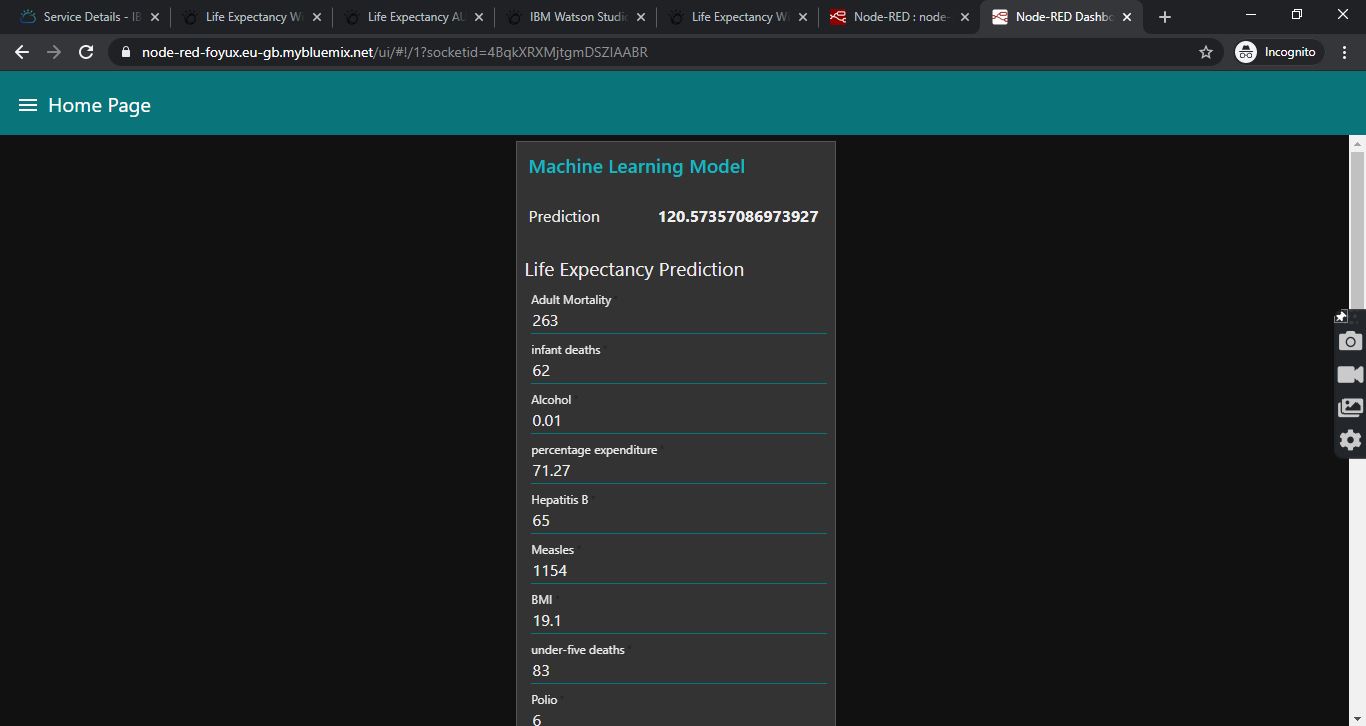
**ALGORITHMS USED:**

* **Linear Regression**
* **SVM**
* **Lasso Regression**
* **Ridge Regression**

**NODE-RED Flow:**



**Output:**

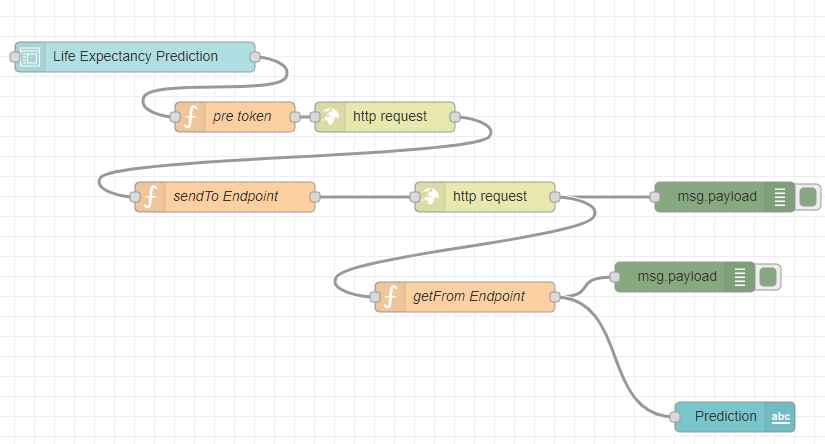
****

1. **Life Expectancy Without Python:**

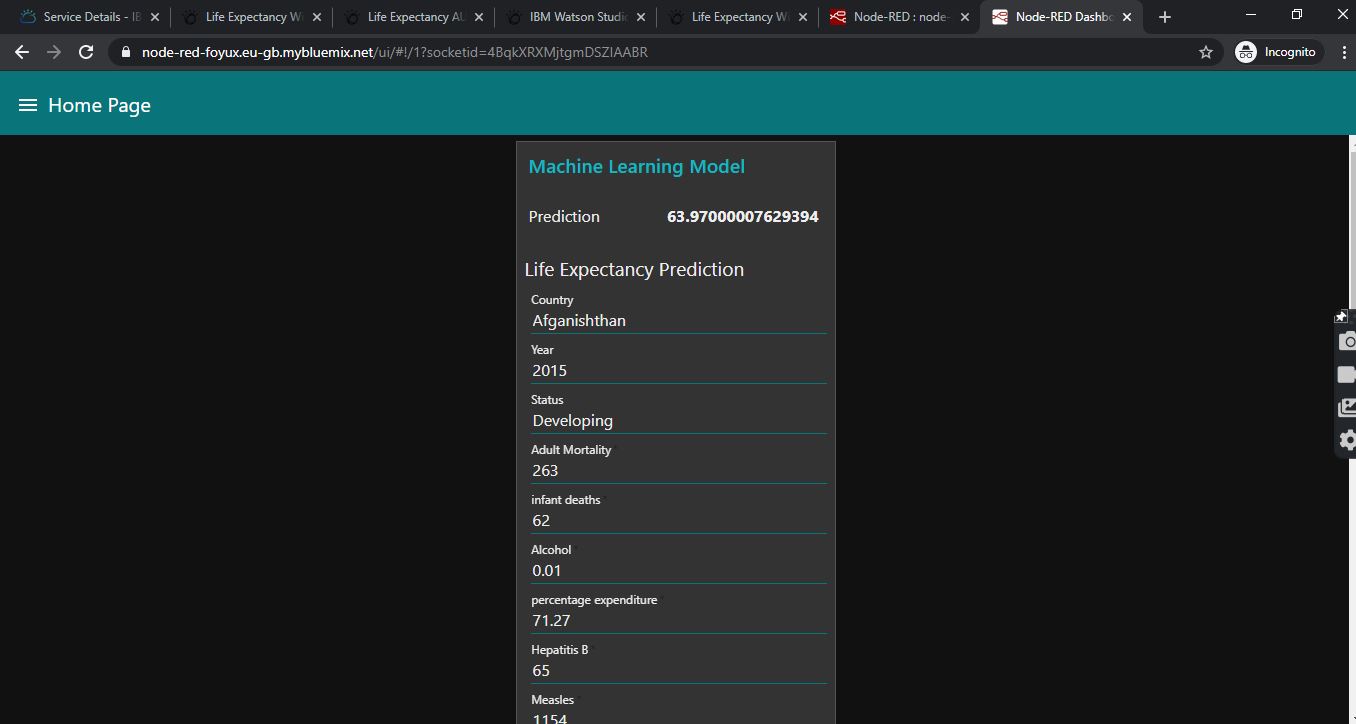
Second Part of Project to use IBM cloud’s AUTOAI service for predicting life Expectancy .

Algorithm : Extra Trees Regressor

**NODE-RED FLOW:**



**Output:**

****

**RESULTS:**

The results obtained for the AUTOAI model were :

Extra Trees Regressor- 96%

The results obtained for Python Model:

Linear Regression: 78%

SVM- 85%

Ridge- 82%

Lasso- 81%