Project Report

on

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

By : SOHAM R. SAHARE

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

Project Summary :

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 measure of the average time a human being is expected to live, Life expectancy depends on various factors: Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth and other demographic factors. This problem statement provides a way to predict average life expectancy of people living in a country when various factors such as year, GDP, education, alcohol intake of people in the country, expenditure on healthcare system and some specific disease related deaths that happened in the country are given.

The output algorithms have been used to test if they can maintain their accuracy in predicting the life expectancy for data they have not been trained. Some of the algorithms that can be possibly used are:

- Linear Regression
- Random Forest Regression
- Decision Trees

And by Hyperparameter Tuning increasing the accuracy of predictive model.

Project Requirement :

This project fundamentally aims in predicting the life expectancy. The primary requirement of the project is the suitable dataset which will aid the prediction. The dataset will provide various details like kind of diseases leading to the death. By using supervised machine learning techniques, we can extract a model that will be able to predict the life expectancy of future years.

Functional Requirement :

Create a data model present on the database. The data set are made available to the public to the purpose of health data analysis. It is related to the different countries depending on the different countries while finding the data set in different countries might be difficult and hence some countries are excluded from the final data set.

Technical Requirement :

The merged data set by using the databases in the .csv formats from Kaggle. Datasets need to be integrated into the Python IDE.

Software Requirements :

- Python IDE
- IBM Cloud
- IBM Watson
- IBM Node-Red

Project Deliverables :

The project is scheduled for 29 days from 15th May 2020 to 14th June 2020.

Project Team :

The project is done individually by Soham R. Sahare.

Phrases of the development

1) Project Planning & Kickoff :

   - Writing Project Scope, Schedule, Team & Deliverables for the project.
   - Setup the Development Environment – Creating GitHub account, Slack account, working with zoho writer and Understanding GitHub.

2) Explore IBM Cloud Platform :

   - Creating IBM account
   - Creating a Node-Red Starter Application

3) Explore IBM Watson Services :

   - Exploring IBM Watson usecases.
   - Exploring IBM Watson for Machine Learning.

4) Introduction To Watson Studio :

   - Building a Machine Learning model in IBM Watson Studio.

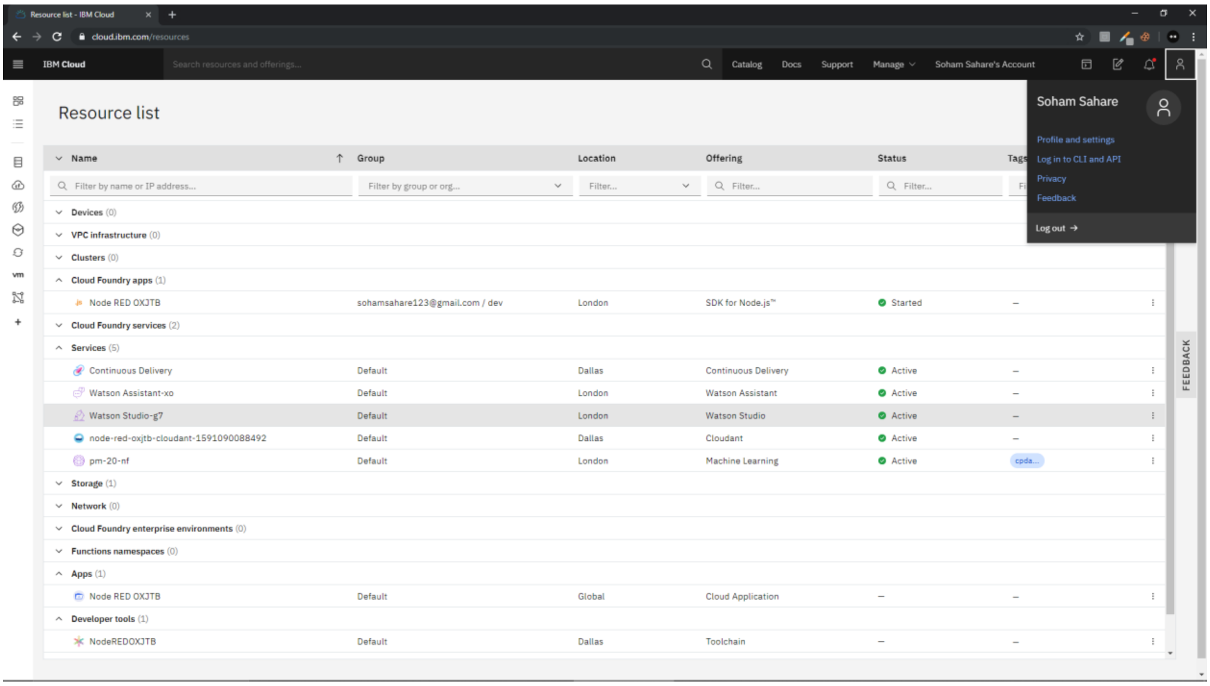
- Automating Machine Learning Model.

5) Predicting Life Expectancy with Python :

- Collecting Dataset from Kaggle.
- Creating Necessary IBM Cloud Services.
- Watson Studio Project
- Creating Machine Learning Service
- Creating a Jupyter Notebook in IBM Watson Studio and Import the dataset.
- Building a Machine Learning Model And Create Endpoints For Node-RED Integration.
- Build Node-RED Flow To Integrate ML Services

Some Screenshots while Project Develpoment :

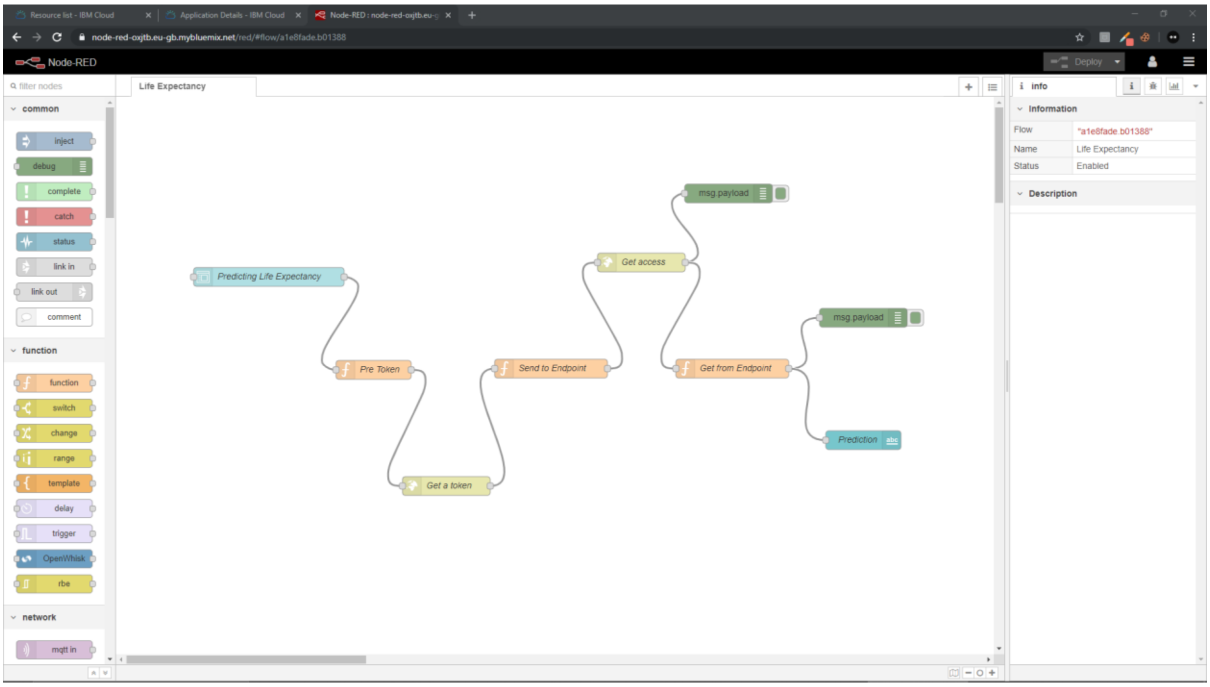
1) Resource list of all the apps and services developed in IBM Cloud.



Resource list

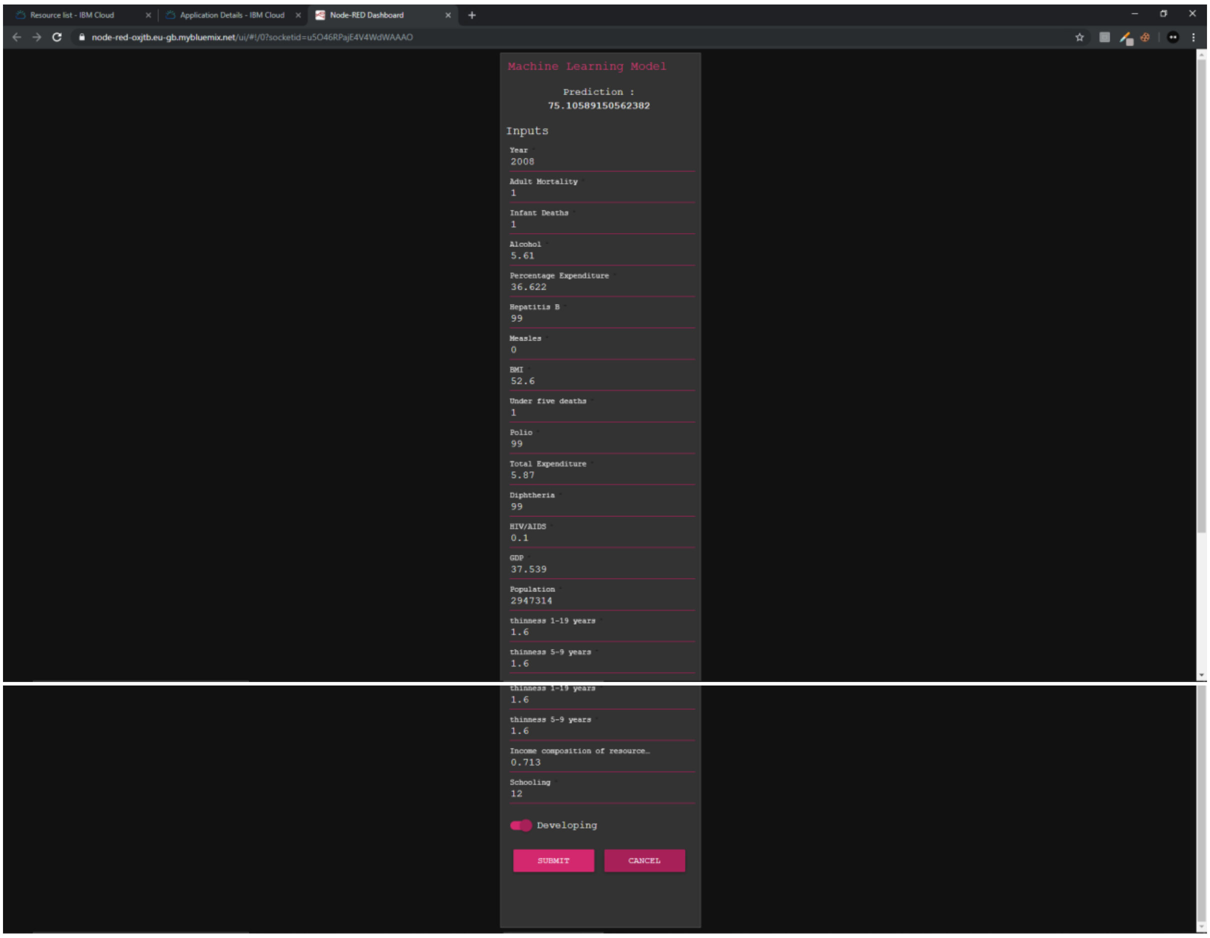
Name	Group	Location	Offering	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Fi
Devices (0)					
VPC infrastructure (0)					
Clusters (0)					
Cloud Foundry apps (1)					
Node RED OXJTB	sohamsahare123@gmail.com / dev	London	SDK for Node.js™	Started	–
Cloud Foundry services (2)					
Services (5)					
Continuous Delivery	Default	Dallas	Continuous Delivery	Active	–
Watson Assistant-xo	Default	London	Watson Assistant	Active	–
Watson Studio-g7	Default	London	Watson Studio	Active	–
node-red-oxjtb-cloudant-1591090088492	Default	Dallas	Cloudant	Active	–
pm-20-nf	Default	London	Machine Learning	Active	cpda...
Storage (1)					
Network (0)					
Cloud Foundry enterprise environments (0)					
Functions namespaces (0)					
Apps (1)					
Node RED OXJTB	Default	Global	Cloud Application	–	–
Developer tools (1)					
NodeREDOXJTB	Default	Dallas	Toolchain	–	–

2) Node-Red Flow Editor.



Flow	"a1e8fade.b01388"
Name	Life Expectancy
Status	Enabled

3) Node-Red Deployed.


Resource list - IBM Cloud
Application Details - IBM Cloud
Node-RED Dashboard
node-red-oxjtb.eu-gb.mybluemix.net/ui/#!/0?socketid=u5O46RPaJE4V4WdWAAAO

Machine Learning Model

Prediction :
75.10589150562382

Inputs

Year
2008

Adult Mortality
1

Infant Deaths
1

Alcohol
5.61

Percentage Expenditure
36.622

Hepatitis B
99

Measles
0

BMI
52.6

Under five deaths
1

Polio
99

Total Expenditure
5.87

Diphtheria
99

HIV/AIDS
0.1

GDP
37.539

Population
2947314

thinness 1-19 years
1.6

thinness 5-9 years
1.6

thinness 1-19 years
1.6

thinness 5-9 years
1.6

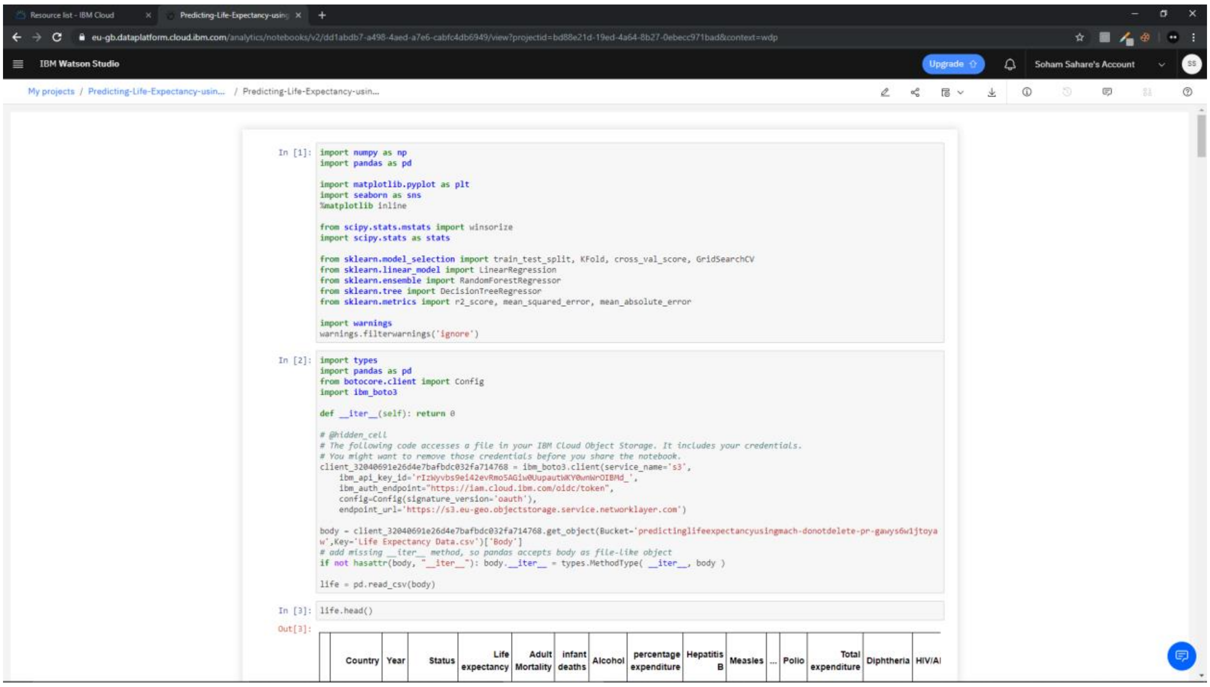
Income composition of resource…
0.713

Schooling
12

● Developing

SUBMIT CANCEL

4) Jupyter Notebook in IBM Watson Studio for Development of Machine Learning Model.


Resource list - IBM Cloud
Predicting-Life-Expectancy-using
eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/dd1abdb7-a498-4aed-a7e6-cabfc4db6949/view?projectid=bd88e21d-19ed-4a64-8b27-0ebecc971bad&context=wdp
IBM Watson Studio
Upgrade
Soham Sahare's Account
My projects / Predicting-Life-Expectancy-usin… / Predicting-Life-Expectancy-usin…

In [1]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from scipy.stats.mstats import winsorize
import scipy.stats as stats

from sklearn.model_selection import train_test_split, KFold, cross_val_score, GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

import warnings
warnings.filterwarnings('ignore')
```

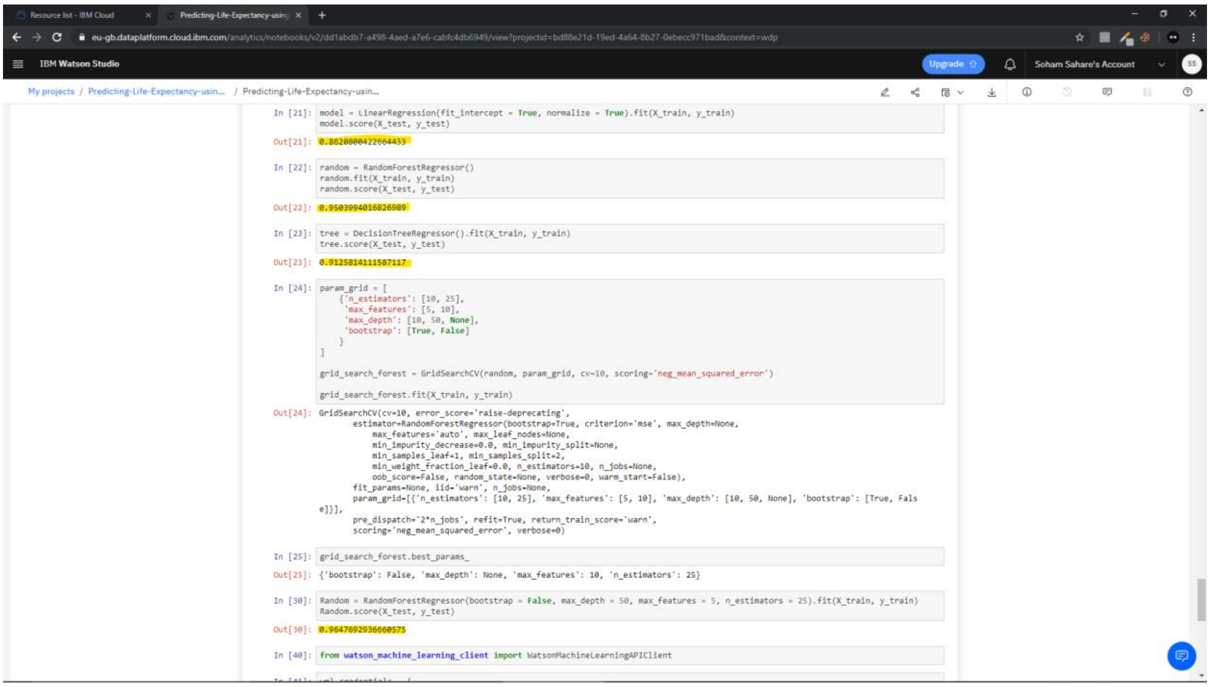
In [2]:
```
import types
import pandas as pd
from botocore.client import Config
import ibm_boto3

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_32040691e26d4e7bafbdc032fa714768 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='rIzWyvbs9ei42evRmo5AGiw0UupautWKY0wnWrOIBMd_',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

body = client_32040691e26d4e7bafbdc032fa714768.get_object(Bucket='predictinglifeexpectancyusingmach-donotdelete-pr-gawys6w1jtoya
w',Key='Life Expectancy Data.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

life = pd.read_csv(body)
```

In [3]:
```
life.head()
```

Out[3]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AI

5) Accuracy of different Machine Learning Models.


Resource list - IBM Cloud
Predicting-Life-Expectancy-using
eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/dd1abdb7-a498-4aed-a7e6-cabfc4db6949/view?projectid=bd88e21d-19ed-4a64-8b27-0ebecc971bad&context=wdp
IBM Watson Studio
Upgrade
Soham Sahare's Account
SS
My projects / Predicting-Life-Expectancy-usin... / Predicting-Life-Expectancy-usin...

```
In [21]: model = LinearRegression(fit_intercept = True, normalize = True).fit(X_train, y_train)
         model.score(X_test, y_test)

Out[21]: 0.8620800422664433

In [22]: random = RandomForestRegressor()
         random.fit(X_train, y_train)
         random.score(X_test, y_test)

Out[22]: 0.9503994016826989

In [23]: tree = DecisionTreeRegressor().fit(X_train, y_train)
         tree.score(X_test, y_test)

Out[23]: 0.9125814111587117

In [24]: param_grid = [
             {'n_estimators': [10, 25],
              'max_features': [5, 10],
              'max_depth': [10, 50, None],
              'bootstrap': [True, False]
             }
         ]

         grid_search_forest = GridSearchCV(random, param_grid, cv=10, scoring='neg_mean_squared_error')

         grid_search_forest.fit(X_train, y_train)

Out[24]: GridSearchCV(cv=10, error_score='raise-deprecating',
                estimator=RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
                    max_features='auto', max_leaf_nodes=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
                    oob_score=False, random_state=None, verbose=0, warm_start=False),
                fit_params=None, iid='warn', n_jobs=None,
                param_grid=[{'n_estimators': [10, 25], 'max_features': [5, 10], 'max_depth': [10, 50, None], 'bootstrap': [True, Fals
         e]}],
                pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
                scoring='neg_mean_squared_error', verbose=0)

In [25]: grid_search_forest.best_params_

Out[25]: {'bootstrap': False, 'max_depth': None, 'max_features': 10, 'n_estimators': 25}

In [30]: Random = RandomForestRegressor(bootstrap = False, max_depth = 50, max_features = 5, n_estimators = 25).fit(X_train, y_train)
         Random.score(X_test, y_test)

Out[30]: 0.9647692936660575

In [40]: from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

Conclusion :

From this project we can predict Life Expectancy with 96.47 % accuracy. Using Hyper parameterized Random Forest Regressor.

Important Links :

Node-Red URL : https://node-red-oxjtb.eu-gb.mybluemix.net/ui/#!/0?socketid=YdWb57ES7FMjDH76AAAH

Node-Red Editor URL : https://node-red-oxjtb.eu-gb.mybluemix.net/red/#flow/a1e8fade.b01388

Website video : https://drive.google.com/file/d/1zj4NXGgM2DkH_eSawpeNB6SQiyef26p-/view?usp=sharing

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