Internship Report

Predicting Life Expectancy using Machine Learning

14th May 2020 to 11st June 2020

Organization:
Smartbridge Trading Solutions Pvt Ltd
A 12/13 , B & B Genesis Building,
Noida, Uttar Pradesh 201301

Report Submitted By:
Akshat Chauhan
Enroll. No. 41510402717
CSE 3rd Year
Amity School of Engineering and Technology, New Delhi

Table of Contents

List of Figures

1. Introduction

1.1. Overview

The intern program is intended to create a Life Expectancy prediction model with a User Interface. A Regression Machine Learning model is created that leverages historical data to predict Life Expectancy of a country given various features. The development is done on IBM cloud using various services i.e. IBM Watson Studio, Machine Learning Service, Node-RED and Cloudant.

1.2. Purpose

Life expectancy is one of the most important factors in end-of-life decision making. Good prognostication helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more broadly: facilitates Advance Care Planning in a country. Advance Care Planning improves the quality of the final phase of life by stimulating doctors to explore the preferences for end-of-life care with their patients, and people close to the patients. Physicians, however, tend to overestimate life expectancy, and miss the window of opportunity to initiate Advance Care Planning.

2. 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 project will focus on immunization factors, mortality

factors, economic factors, social factors and other health related factors as well. Since the observations the 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.

2.2. Proposed Solution

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. The machine learning model built using historical data 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.

3. Theoretical Analysis

3.1. Block Diagram

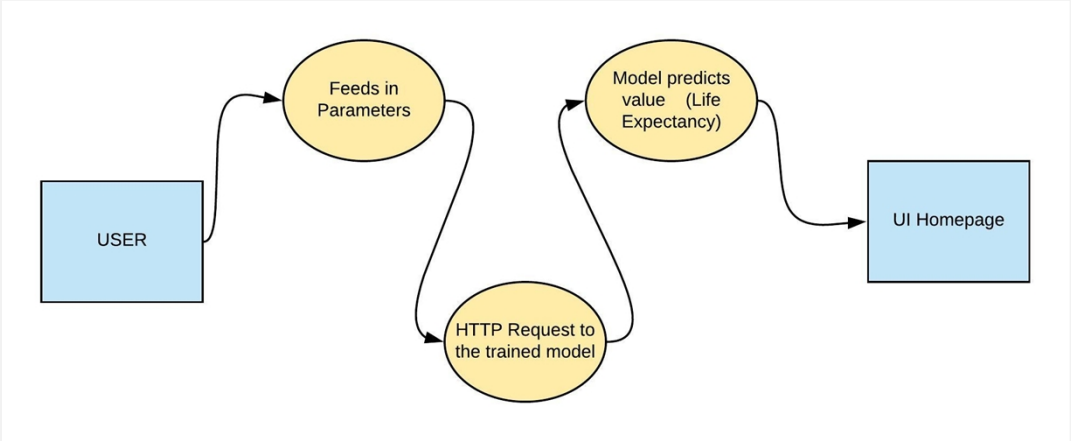

Fig. 1 Block Diagram of the system

3.2. Software Design

The regression model built in python is deployed on IBM cloud. The Node-RED application then sends HTTP request with all the required parameters to the trained model. The model then sends the HTTP response which is then parsed and displayed on the UI.

4. Experimental Investigations

4.1. Factors affecting Life Expectancy

Below are the factors (given in the dataset) which affect life expectancy of a country.

1. **Adult Mortality:** Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)
2. **Infant Deaths:** Number of Infant Deaths per 1000 population
3. **Alcohol:** Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)
4. **Percentage Expenditure**: Expenditure on health as a percentage of Gross Domestic Product per capita(%)
5. **Hepatitis B:** Hepatitis B immunization coverage among 1-year-olds (%)
6. **Measles:** Measles - number of reported cases per 1000 population
7. **BMI:** Average Body Mass Index of the entire population
8. **Under-five deaths:** Number of under-five deaths per 1000 population
9. **Polio:** Polio (Pol3) immunization coverage among 1-year-olds (%)
10. **Total Expenditure:** General government expenditure on health as a percentage of total government expenditure (%)
11. **Diphtheria:** Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)
12. **HIV/AIDS:** Deaths per 1 000 live births HIV/AIDS (0-4 years)
13. **GDP:** Gross Domestic Product per capita (in USD)
14. **Population:** Population of the country
15. **Thinness 5-9 years:** Prevalence of thinness among children for Age 5 to 9(%)
16. **Thinness 1-19 years:** Prevalence of thinness among children and adolescents for Age 10 to 19 (%)
17. **Income composition of resources:** Human Development Index in terms of income composition of resources (index ranging from 0 to 1)
18. **Schooling:** Number of years of Schooling(years)

4.2. Correlation between factors and Life Expectancy

Below is the correlation heat map of the dataset.

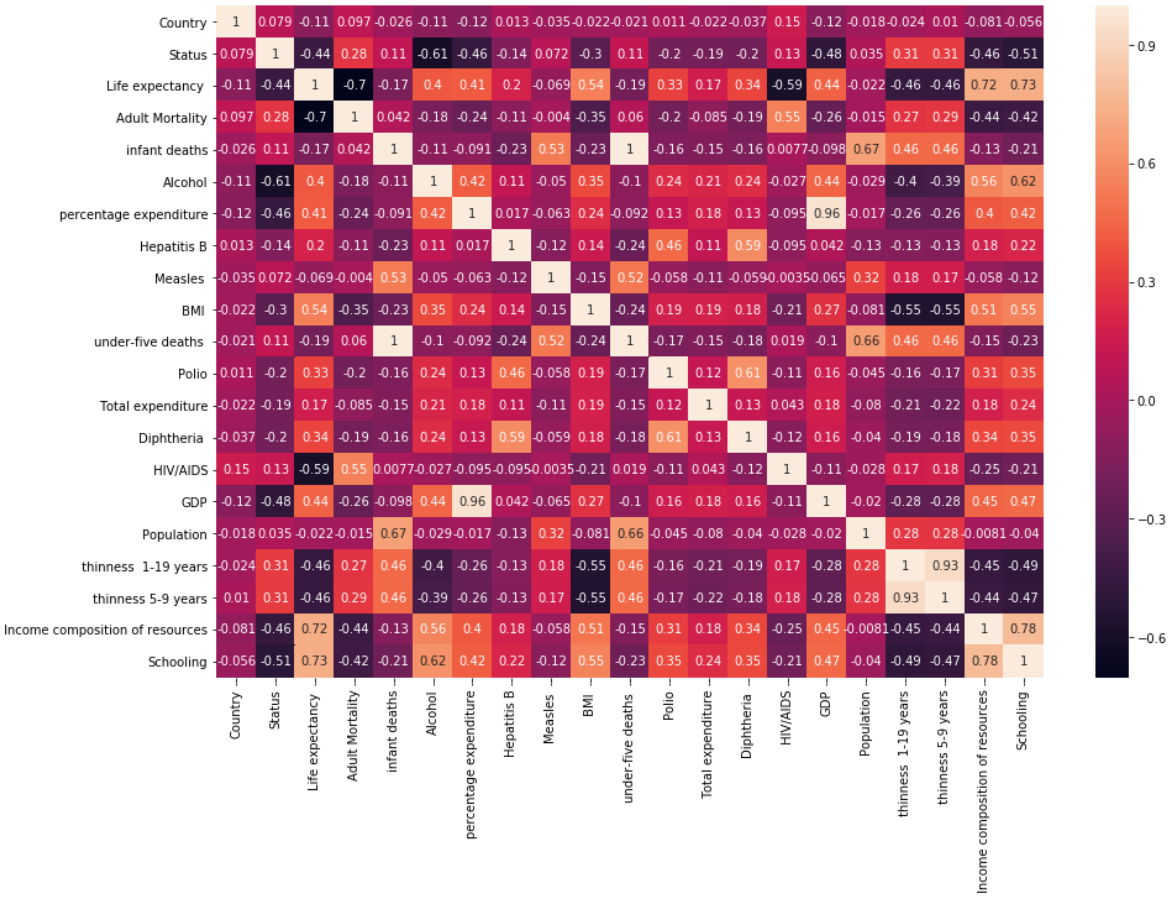

Fig.2 Correlation Heat Map

It is observable that Schooling, Income composition of resources and Adult Mortality are highly correlated to Average Life Expectancy.

4.3. Implementing Regression Models

Two regression models are applied on the dataset.

4.3.1. Multiple Linear Regression

Linear regression is a **linear model**, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric. Different techniques can be used to prepare or train the linear regression equation from data. The one used in this project

is called Mean Absolute Error, which is, measure of errors between paired observations expressing the same phenomenon. R squared value is the proportion of the variance in the dependent variable that is predictable from the independent variable.

The Mean Absolute Error and R-squared value during testing of this model are given below.

```
In [20]: y_pred1 = regr1.predict(X_test)
         print('Simple Linear Regression : ')
         print('Mean Absolute Error : ',mae(y_test, y_pred1))
         print('R_square score: %.4f' % r2_score(y_test, y_pred1))

         Simple Linear Regression :
         Mean Absolute Error :  2.7269763634286406
         R_square score: 0.8545
```

4.3.2. Random Forest Regression

A Random Forest is an **ensemble technique** capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

```
In [24]: y_pred2 = regr2.predict(X_test)

         print('Random Forest Regression : ')
         print('Mean Absolute Error : ',mae(y_test, y_pred2))
         print('R_square score : %4f' % r2_score(y_test, y_pred2))

         Random Forest Regression :
         Mean Absolute Error :  1.1507102272727396
         R_square score : 0.959955
```

It is observable that the Mean Absolute Error in Random Forest Regression is lower than that in Multiple Linear Regression. So, for the final deployment, Random Forest Regression is used.

5. Flowchart

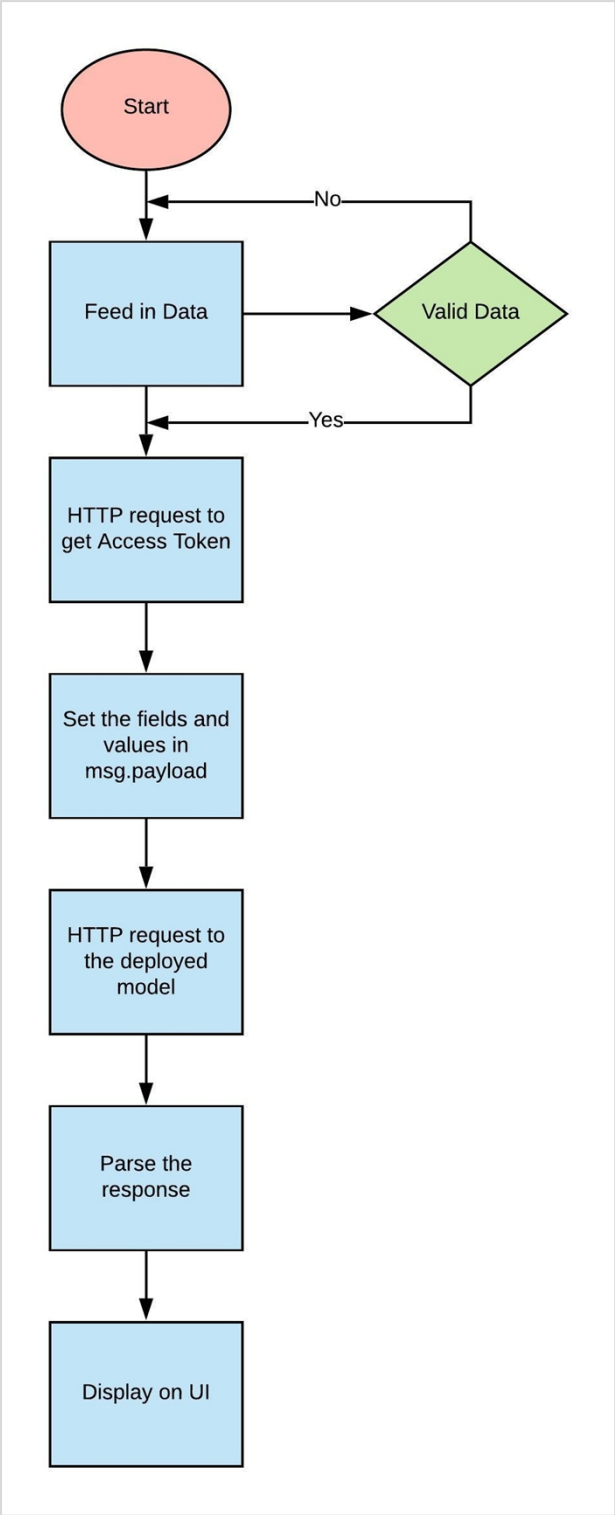

Fig.3 System Flowchart

6. Result

6.1. Node-RED Flow

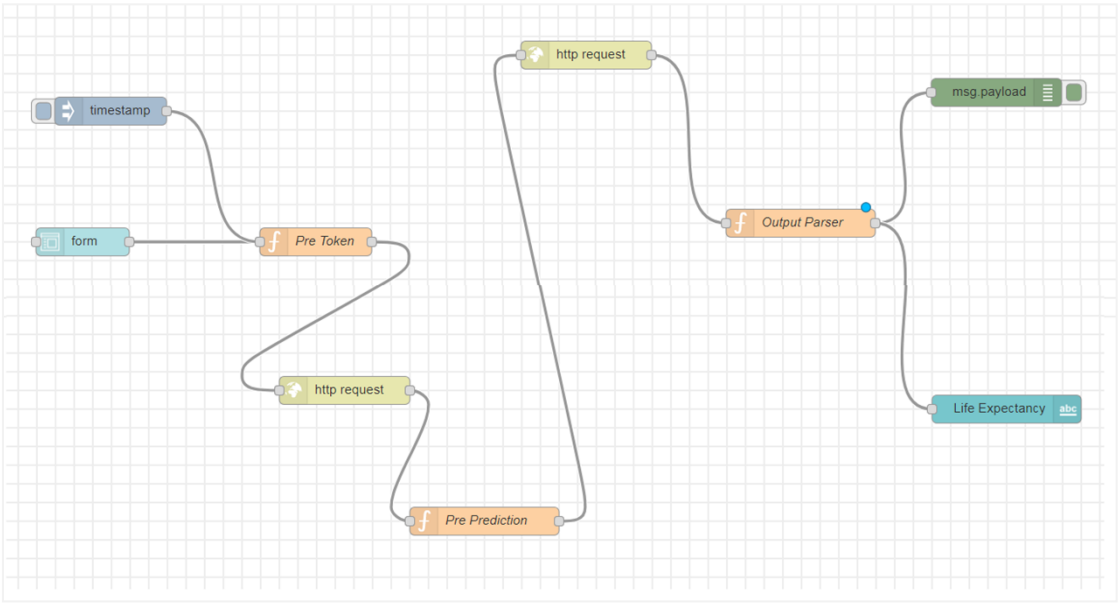

Fig. 4 Node-RED Flow

6.2. User Interface

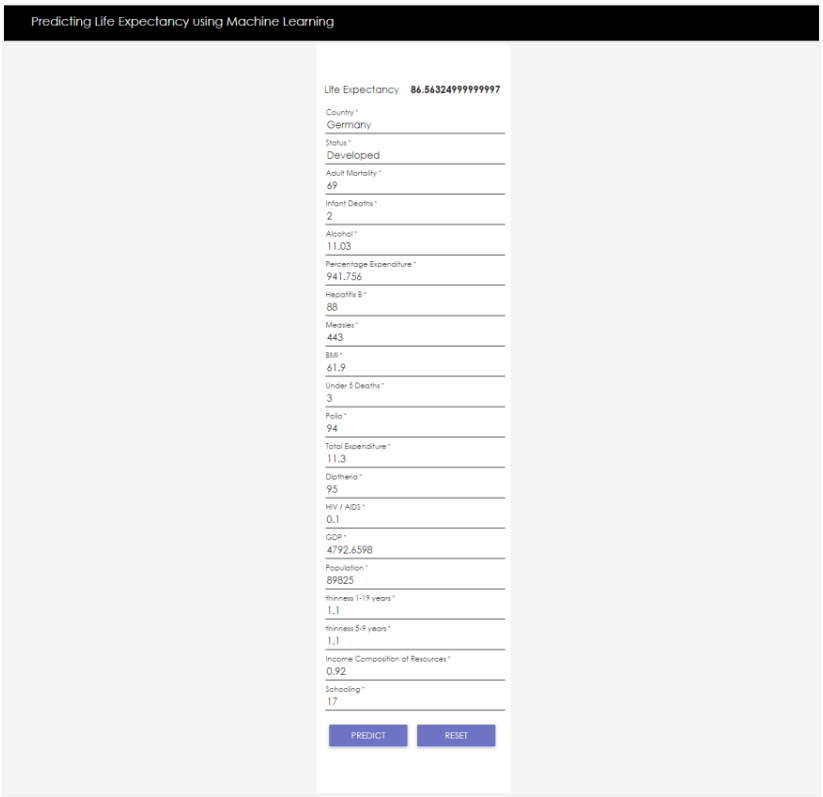

Fig. 5 User Interface

7. Advantages and Disadvantages

7.1. Advantages

1. The machine learning algorithm used in this project is Random Forest regression, which is based on the **bagging** algorithm and uses **Ensemble Learning** technique. It creates as many trees on the subset of the data and combines the output of all the trees. In this way it reduces over fitting problem in decision trees and also reduces the variance and therefore improves the accuracy.
2. Non linear parameters don't affect the performance of a Random Forest unlike curve based algorithms. So, if there is high non-linearity between the independent variables, Random Forest may outperform as compared to other curve based algorithms.
3. Random Forest is usually robust to outliers and can handle them automatically.
4. Random Forest is comparatively **less impacted by noise**.

7.2. Disadvantages

1. **Complexity:** Random Forest creates a lot of trees (unlike only one tree in case of decision tree) and combines their outputs. By default, it creates 100 trees in Python sklearn library. To do so, this algorithm requires much more computational power and resources. On the other hand decision tree is simple and does not require so much computational resources.
2. **Longer Training Period:** Random Forest require much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decision on the majority of votes.

8. Applications

Life expectancy is one of the most important factors in end-of-life decision making. Good prognostication helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more broadly: facilitates Advance Care Planning in a country. Advance Care Planning improves the quality of the final phase of life by stimulating doctors to explore the preferences for end-of-life care with their patients, and people close to the patients. Physicians, however, tend to overestimate life expectancy, and miss the window of opportunity to initiate Advance Care Planning.

9. Conclusion

This project gives a great insight into the factors affecting the life expectancy of a country. And by using the same, we can take measures to improve life expectancy by modifying and bringing new laws and rules in the system. Healthcare planning can be done in the critical areas of a country. Incentives can be provided to people to bring changes to their lifestyle so that they can live longer, which will eventually be helpful in development of the nation.

10. Future Scope

10.1. Planning Health Services

The government can plan health services better using the data and future predictions. Life expectancy plays a major role in development of a country, hence, using predictions and trends, the health infrastructure can be improved.

10.2. Personalized Health Apps

A mobile application can be developed that uses personal health data (from Smart Watch and Health apps) and historical data of the country that user lives in and predict the expected life span of that user.

11. Bibliography

1. Statistical Analysis on factors influencing Life Expectancy. Source: https://www.kaggle.com/kumarajarshi/life-expectancy-who/metadata
2. Deploying an AutoAI model – IBM. Source: https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/autoai-deploy-model.html
3. Using the machine learning model – IBM. Source: https://cloud.ibm.com/docs/watson-knowledge-studio?topic=watson-knowledge-studio-publish-ml
4. Welcome to the documentation of Watson Studio. Source: https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/welcome-main.html?context=analytics
5. Infuse AI into your applications with Watson AI to make more accurate predictions. Source: https://www.ibm.com/watson/products-services
6. Get an understanding of Machine Learning. Source: https://developer.ibm.com/technologies/machine-learning/series/learning-path-machine-learning-for-developers/

Appendix

A. Source Code:

Importing all the required libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import seaborn as sns
from sklearn.metrics import mean_absolute_error as mae
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
```

Data Preprocessing

```
# Replacing the nan values with mean values
#dataset = dataset.fillna(value = dataset.mean())
dataset = dataset.dropna(how = 'any')
# Dropping the year column
dataset = dataset.drop('Year', axis = 1)
# Encoding the Status column
le = LabelEncoder()
le.fit(dataset['Status'])
dataset['Status'] = le.transform(dataset['Status'])
dataset.head()
# Encoding the Country column
le2 = LabelEncoder()
le2.fit(dataset['Country'])
dataset['Country'] = le2.transform(dataset['Country'])
```

Heat map of correlation between features and Life Expectancy

```
# Plotting a heat map representing correlation of each column with Li
fe Expectancy
# Setting the figure size
plt.figure(figsize = (15,10))
sns.heatmap(dataset.corr(), annot = True)
```

Dependent and Independent variables

```
# Creating Independent and Dependent variables
y = dataset['Life expectancy '].values

X = dataset.drop(columns = ['Life expectancy ']).values
```

Splitting the dataset into training and testing data

```
# Splitting the data into training and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 0)
```

Linear Regression

```
# Applying simple Linear Regression
regr1 = LinearRegression()
regr1.fit(X_train, y_train)
regr1.coef_
y_pred1 = regr1.predict(X_test)
print('Simple Linear Regression : ')
print('Mean Absolute Error : ',mae(y_test, y_pred1))
print('R_square score: %.4f' % r2_score(y_test, y_pred1))
```

Random Forest Regression

```
# Implementing Random Forest Regression
regr2 = RandomForestRegressor(n_estimators = 800, random_state = 0)
regr2.fit(X_train, y_train)
X_test[0]
y_pred2 = regr2.predict(X_test)

print('Random Forest Regression : ')
print('Mean Absolute Error : ',mae(y_test, y_pred2))
print('R_square score : %4f' % r2_score(y_test, y_pred2))
regr2.predict(X_test)
```