Predicting Life Expectancy using Machine Learning

Machine Learning Engineer Internship

Final Report

Submitted to

The Smartbridge

Submitted By

Chandresh Maniya

INDEX

CHAPTER 1

INTRODUCTION

1.1 Overview

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 leave. It depends on different factors like Mental and Physical Illness, GDP, Education, Alcohol, Income Source, Adult Mortality, Body Mass Index, HIV/AIDS etc.

1.2 Purpose

Purpose of the project is to identify the factors which causes to the death and to finding the chance of increasing the rate of living.

CHAPTER 2
LITTERATURE SURVEY

2.1 Existing Problem

In 2018, James Jin Kang and Sasan Adibi had done Systematic Predictive Analysis of Personalized Life Expectancy using Smart Device. They predicted life using data coming from smart wireless devices such as a smart watches.

2.2 Proposed Solution

For prediting life expectancy, I am using Linear Regression Machine Learning Model. Regression basically finds the relationship between defferent parameters and based on that it genrates quation. And by using this equation, we can finds dependent values. So, using this we can predict life expectancy of a person or country by giving values of differnet parameters.

CHAPTER 3
THEORITICAL ANALYSIS

3.1 Block Diagram

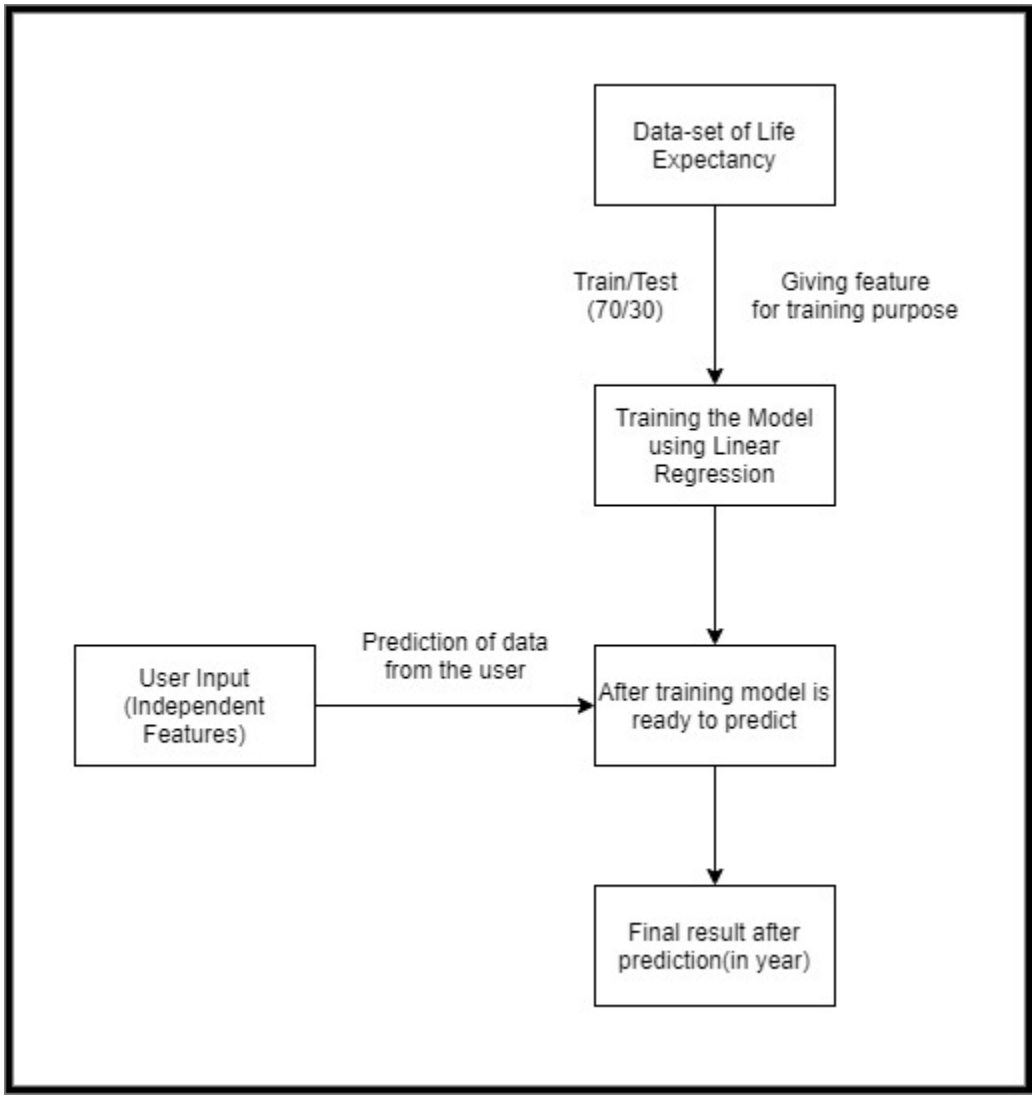

Figure 3.1 : Block Diagram

3.2 Hardware/Software Designing(Required)

Software Required: Jupyter Notebook, IBM Cloud, Python Language
Data Required: Structured Data(CSV). **[Life Expectancy WHO | Kaggle]**

CHAPTER 4

EXPERIMENTAL INVESTIGATION

For predicting life expectancy I go through across different Machine Learning models and Exploratory Data Analysis and Data Preprocessing techinques. Then I chose Linear Regression Model for solving this problem. After that I did analysis between differnet features and searched that which features are dependednt in life prediction. I achieved this task by doing some analysis techniques like ploting the graphs and visualised it. I drop the data which is not necessary for predicting life expectancy.

After completing all the necessary task I created Linear Regression Model. I took 70% data for training purpose and 30% for testing purpose. For deployment, Node-Red App is created using IBM Cloud for the UI purpose in which user can input the data and can get the result from it.

CHAPTER 5

FLOW CHART

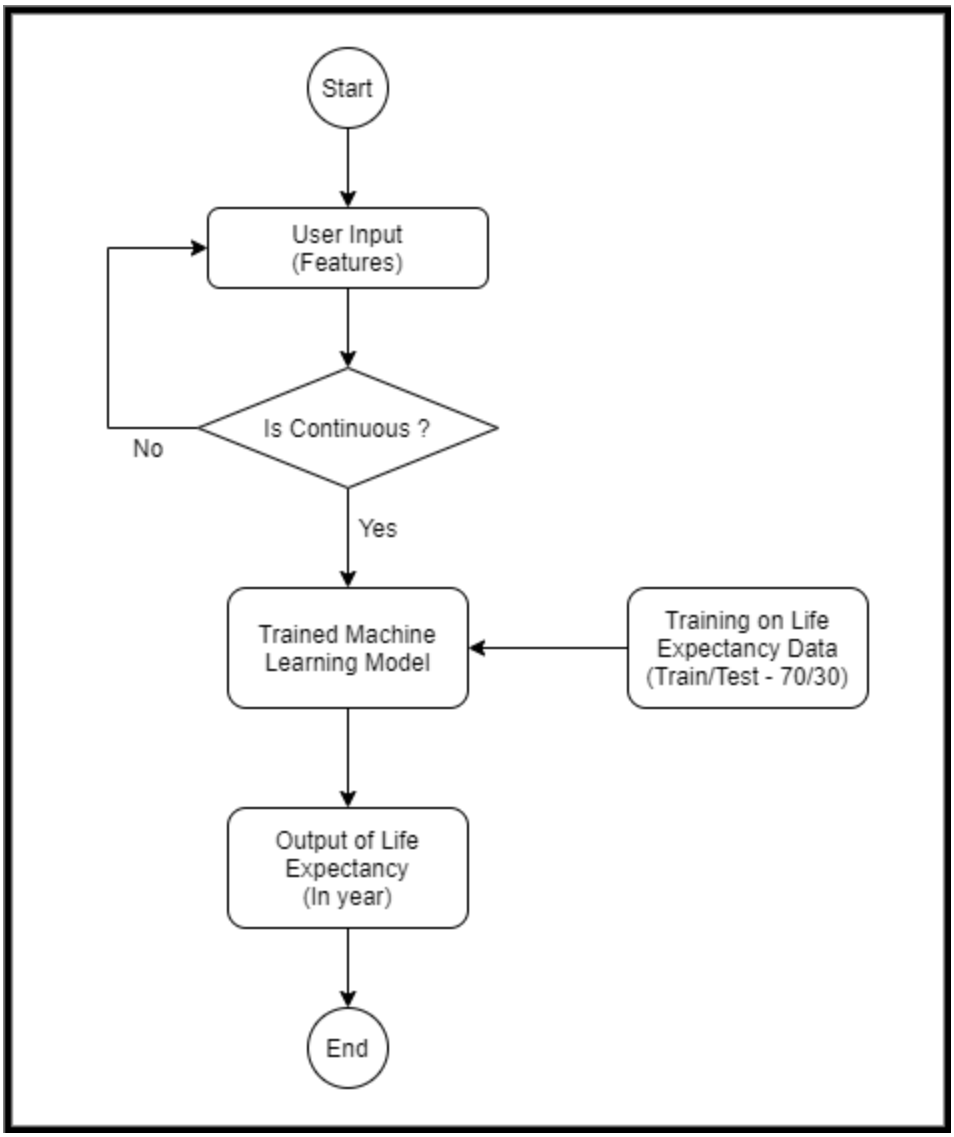

Figure 5.1 : Flow Diagram

CHAPTER 6

RESULT

Here I am atteching screenshots of the system.

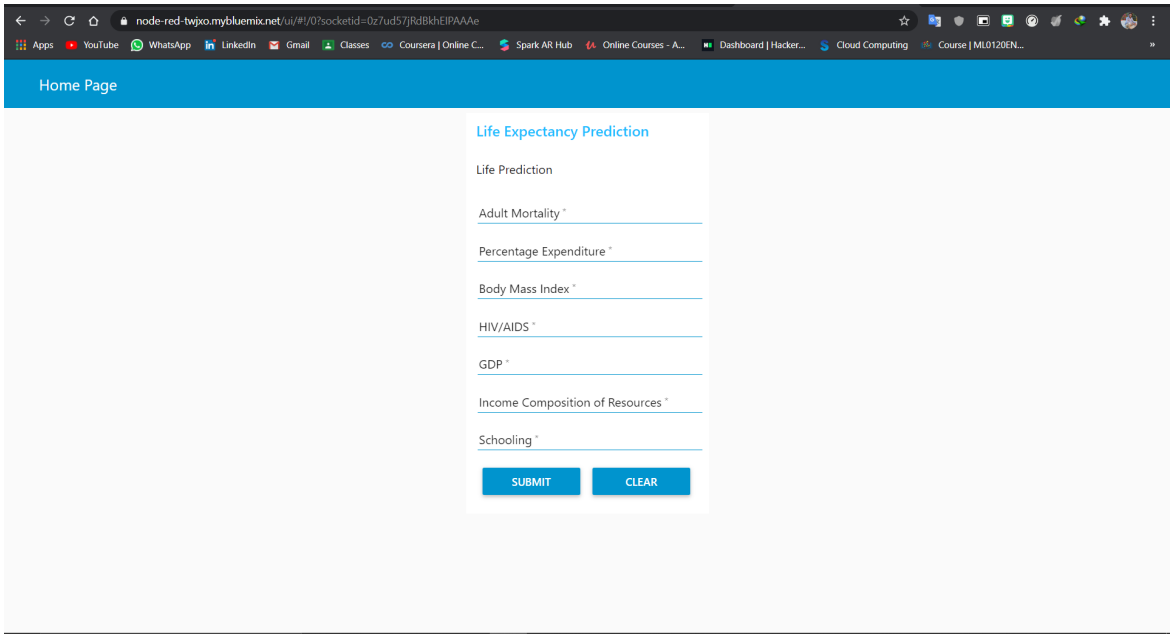

Figure 6.1 : Taking the Input from the User

As you can see in figure 6.1, user have to give different continuous values(features). After that, whenver user will hit the submit button, prediction will be done.

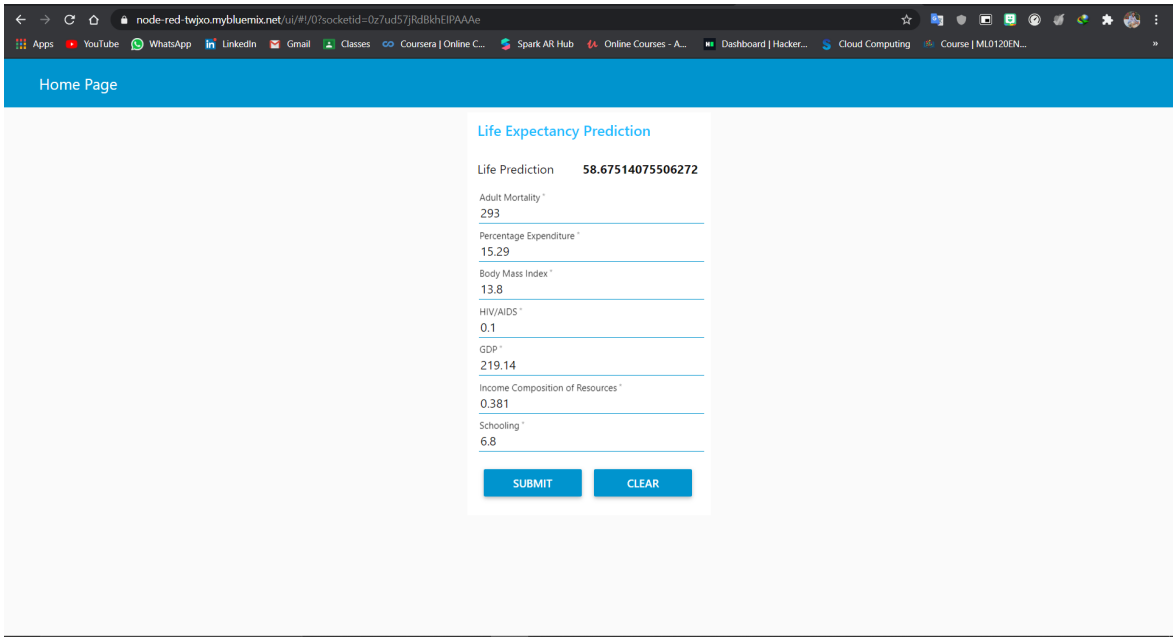

Figure 6.2 : Prediction from User Input

As you can see in figure 6.2, after the inputing all the required features, on the click of submit button, life expectancy will be predicted and will be shown on the page.

CHAPTER 7

ADVANTAGES & DISADVANTAGES

Advantages

Advantage of predicting Life Expectancy is we can identify that what are the things which is not beneficial for living healthy life. We can control all the things which is not good for health and healthy life as well. By predicting average health of country we can also make some decisions and prohibition on the things which are not good for life and we can improve the areas which impacting in a living healthy life.

Disadvantages

We can never predict our future whether it is good or bad. For example, if a person is very healthy today then suddenly he is geting sick except he is taking care of his health, vice-varsa very sick people can also get better and can work efficiently after deafiting their disease.

CHAPTER 8

APPLICATIONS

After analysing the life expectancy of the person or country, we can make apllication like health monitor, and we can develop system like in that all the required features and it's level can shown which is affect to normal life and can give the solutions of that. For example, a person is consuming alcohol then based on life expectancy data we can notify the people that please don't cross this limit of consuming the alcohot, it can cause a bad disease.

Based on the data we can analyse the cause of the death of people and we can try to fix it and we can increase the average year of living.

CHAPTER 9

CONCLUSION

No one can predict the future, but after analysing the terms which are affected to living the life then atleast we can improve our lives and we can manage the things based on prediction.

CHAPTER 10

FUTURE SCOPE

- Predicting the cause of the death and takinf decisions based on that.
- Development of app which tracks all the activity of the person and based on data which affects the life of the person, and can suggest things which are necessary for it.

CHAPTER 11

BIBLIOGRAPHY

1. https://www.mdpi.com/2227-7080/6/3/74/htm
2. https://www.who.int/healthinfo/statistics/LT_method_1990_2012.pdf
3. https://www.kaggle.com/kumarajarshi/life-expectancy-who
4. https://www.edrawmax.com/block-diagram/
5. https://www.futuretimeline.net/blog/2018/10/19.htm
6. https://www.britannica.com/science/life-expectancy
7. https://cloud.ibm.com/

SOURCE CODE

```
#importing required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

LD = pd.read_csv('Life Expectancy Data.csv')        #opening Dataset
LD = LD.drop('Year', axis = 1)                      #dropping column of 'Year'
LD.head()
status = pd.get_dummies(LD.Status)      #changing status to continuous values
LD = pd.concat([LD, status], axis = 1)
LD = LD.drop(['Status'], axis=1)
LD.head()

LD.rename(columns = {'Developing' : '0', 'Developed' : 1})
LD = LD.drop(['Country'], axis=1)
LD.head()
LD.columns
LD.info()

#Checking Relationship between HIV/AIDS and Life Expectancy
plt.scatter(LD[' HIV/AIDS'], LD['Life expectancy'])
plt.xlabel('HIV/AIDS')
plt.ylabel('Life expectancy')

#Checking Relationship between GDP and Life Expectancy
plt.scatter(LD['GDP'], LD['Life expectancy'])
plt.xlabel('GDP')
plt.ylabel('Life expectancy')

#Checking Relationship between BMI and Life Expectancy
plt.scatter(LD[' BMI '], LD['Life expectancy'])
plt.xlabel('BMI')
plt.ylabel('Life expectancy')

#Checking Relationship between UnderFiveDeaths and Life Expectancy
plt.scatter(LD['under-five deaths '], LD['Life expectancy'])
```

```
plt.xlabel('under-five deaths')
plt.ylabel('Life expectancy')

#Checking Relationship between Alcohol and Life Expectancy
plt.scatter(LD['Alcohol'], LD['Life expectancy'])
plt.xlabel('Alcohol')
plt.ylabel('Life expectancy')

#Checking Relationship between AdultMortality and Life Expectancy
plt.scatter(LD['Adult Mortality'], LD['Life expectancy'])
plt.xlabel('Adult Mortality')
plt.ylabel('Life expectancy')

#Checking Relationship between Schooling and Life Expectancy
plt.scatter(LD['Schooling'], LD['Life expectancy'])
plt.xlabel('Schooling')
plt.ylabel('Life expectancy')

#Checking Relationship between PercentageExpenditure and Life Expectancy
plt.scatter(LD['percentage expenditure'], LD['Life expectancy'])
plt.xlabel('percentage expenditure')
plt.ylabel('Life expectancy')

#Checking Relationship between Income composition of resources and Life Expectancy
plt.scatter(LD['Income composition of resources'], LD['Life expectancy'])
plt.xlabel('Income composition of resources')
plt.ylabel('Life expectancy')

#Checking Relationship between Schooling and Life Expectancy
plt.scatter(LD['Schooling'], LD['Life expectancy'])
plt.xlabel('Schooling')
plt.ylabel('Life expectancy')

plt.figure(figsize = (14, 10))
sns.heatmap(LD.corr(), annot = True)

LD.columns

#creating trainable data for prediction
X = LD[['Adult Mortality', 'percentage expenditure', ' BMI ', ' HIV/AIDS', 'GDP', 'Income
composition of resources', 'Schooling']]
```

```
Y = LD['Life expectancy']

X.isnull().sum()    #checking total null values in the dataset
Y.isnull().sum()
X.fillna(value = X.mean(), inplace = True)    #replacing null value with means
Y.fillna(value = Y.mean(), inplace = True)
Y.dropna(inplace=True)

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=101)

#importing linear regression model
from sklearn.linear_model import LinearRegression

LinearModel = LinearRegression()
LinearModel.fit(X_train, Y_train)
print(LinearModel.intercept_)
coeff_df = pd.DataFrame(LinearModel.coef_,X.columns,columns=['Coefficient'])
print(coeff_df)
prediction = LinearModel.predict(X_test)

#checking similarity between predicted value and actual value
plt.scatter(Y_test,prediction)

sns.distplot((Y_test-prediction),bins=60);     #What is bins ??

#Evaluation of the Model using Different Metrics
from sklearn import metrics

print('Mean Absolute Error: ', metrics.mean_absolute_error(Y_test, prediction))
print('Mean Squared Error: ', metrics.mean_squared_error(Y_test, prediction))
print('Root Mean Squared Error: ', np.sqrt(metrics.mean_squared_error(Y_test,
prediction)))

from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_credentials={
  "apikey": "dD5A7DumNSeO_UOjvMs45iIOQCrC47NBDFroK25SuSjc",
  "instance_id": "b2f4bd86-c4d1-4756-92de-b9f48c5d475c",
  "url": "https://eu-gb.ml.cloud.ibm.com"
}
```

```
client = WatsonMachineLearningAPIClient( wml_credentials )

model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Chandresh",
            client.repository.ModelMetaNames.AUTHOR_EMAIL:
"maniyachandresh92@gmail.com",
            client.repository.ModelMetaNames.NAME: "LifeExpectancy"}

model_artifact =client.repository.store_model(LinearModel, meta_props=model_props)

published_model_uid = client.repository.get_model_uid(model_artifact)
deployment = client.deployments.create(published_model_uid, name="LifeExpectancy")
scoring_endpoint = client.deployments.get_scoring_url(deployment)
```