PROJECT REPORT

PREDICTION OF LIFE EXPECTANCY USING MACHINE LEARNING

BY SMARTBRIDGE

Submitted by – Divya Arora

Email Id- divyaarora2k@gmail.com

Node Red Form link - https://node-red-rgzhv.eu-gb.mybluemix.net/ui/#!/0?socketid=MJtvUk4-jY-VWxSHAAAM

TABLE OF INDEX

1. INTRODUCTION

1.1 Overview

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. It is very important to predict average life expectancy of a country to analyze further requirements to increase its rate of growth or stabilize the rate of growth in that country.

1.2 Purpose

The purpose of this project is to build a machine learning model for the prediction of life expectancy. This Project helps to find average life expectancy of people living in different countries including various factors that affect life expectancy of a person. Predicting the life expectancy will give the country an idea of the factors which can increase the lifespan of the people living by improving the health care facilities or immunization vaccines for infants.

This project helps in learning and experiencing how to implement machine learning algorithms in real life. This project also helps in learning use of IBM tools like IBM cloud, NODE RED, WATSON STUDIO and WATSON MACHINE LEARNING etc.

2. LITERATURE SURVEY

2.1 Existing Problem

Predicting a human's life expectancy has been a long term question to humankind and there have been many attempts to make prediction accurate. While the calculation of life expectancy is a complicated process and requires many variables and circumstances to take into account, there have been several attempts to create an equation despite it being impractical to simplify these variables into one equation. Someone has tried to predict life expectancy using fuzzy set based model. There has been many works in context of prediction of Life Expectancy.

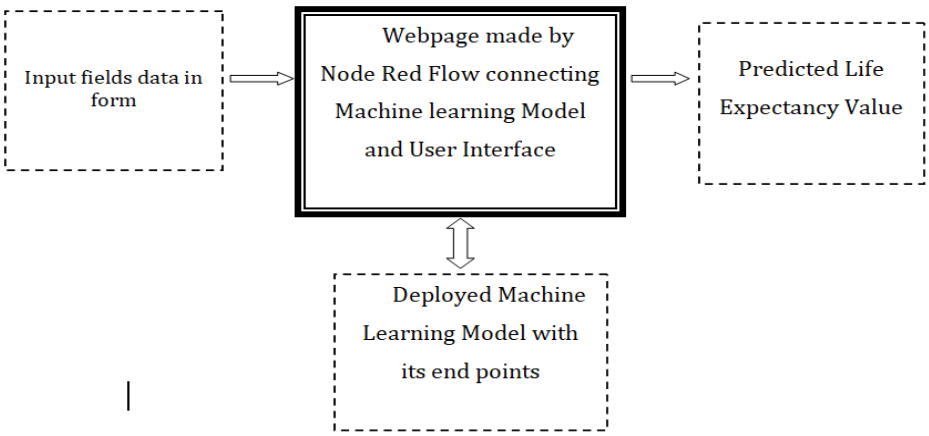
2.2 Proposed Solution

To get better insights and predict the life expectancy more accurately, dataset for this project is taken from Kaggle. It is a platform for predictive modeling and Analytics competitions. Here organization and researchers post the data. Statisticians and data scientist from all over the world compete to produce the best models. In this dataset all factors affecting life of person has been considered. This dataset has been given by WHO (World Health Organization). Many additional factors has been considered like surroundings, country, country status, economy spent on health etc. affecting life of person.

This dataset helped in making machine learning model based on all factors. Machine learning algorithms that can be used in this case are: Regression, Decision Tree, Random Forest, Clustering techniques, so that we can achieve high accuracy for our model. Due to large dataset we are using IBM cloud service for fast work, less time of response and increasing project proficiency.

3. THEORETICAL ANALYSIS

3.1 Block Diagram



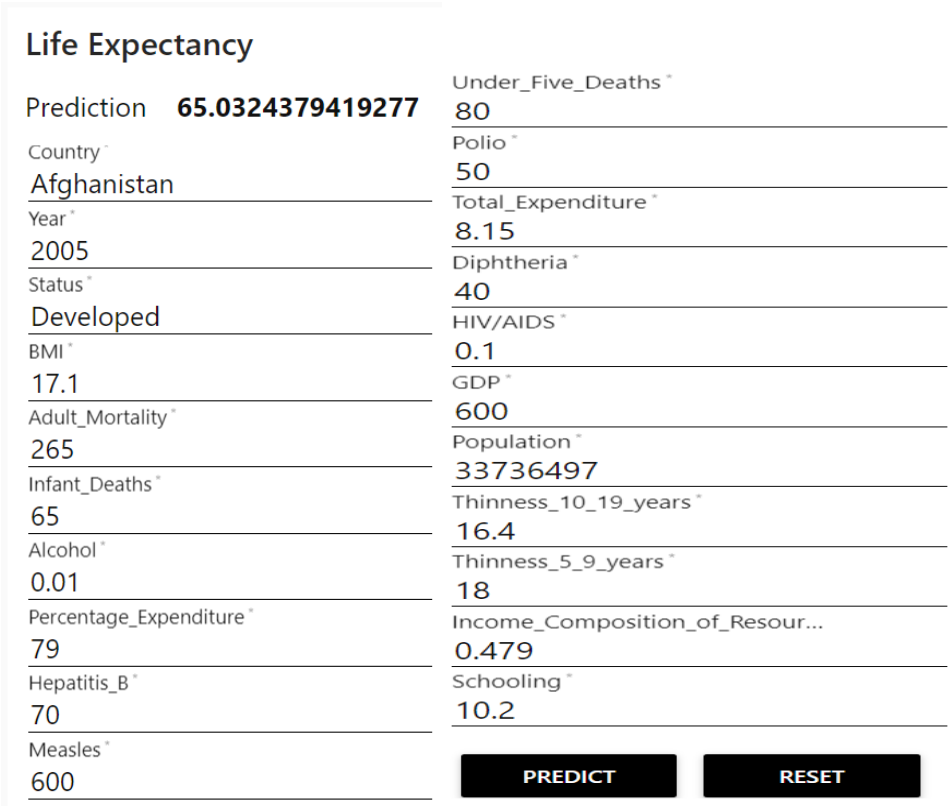
3.2 Hardware/Software Designing

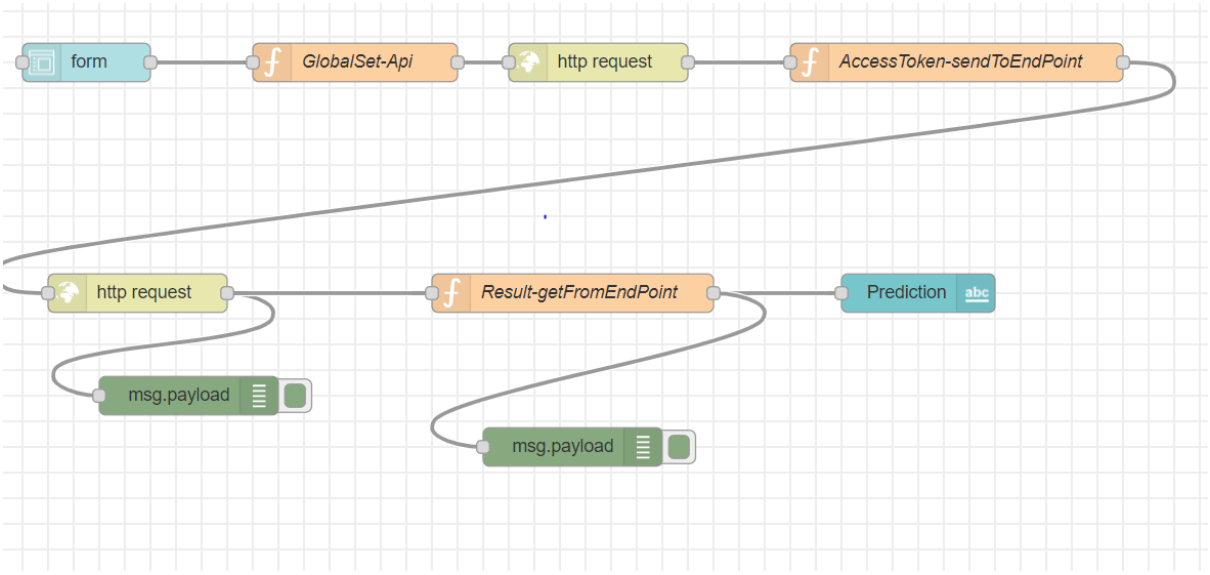
IBM WATSON SERVICE AND NODE RED APP

IBM WATSON STUDIO along with the service of IBM Machine learning for making model of life expectancy by machine learning algorithms. Firstly we added assets in Watson studio that is dataset and notebook. With code we build model for predicting life expectancy based on dataset. And then that model is deployed using Watson_machine_learning_client. Then the model was build and stored in model_artifact. Then the model was deployed and scoring_endpoint url was generated.

Then Node Red Flow is used to connect all the services Machine learning model and User interface. A node red flow is used for making user interface for people can easily access.

Demonstration of Node Red Form





NODE RED FLOW

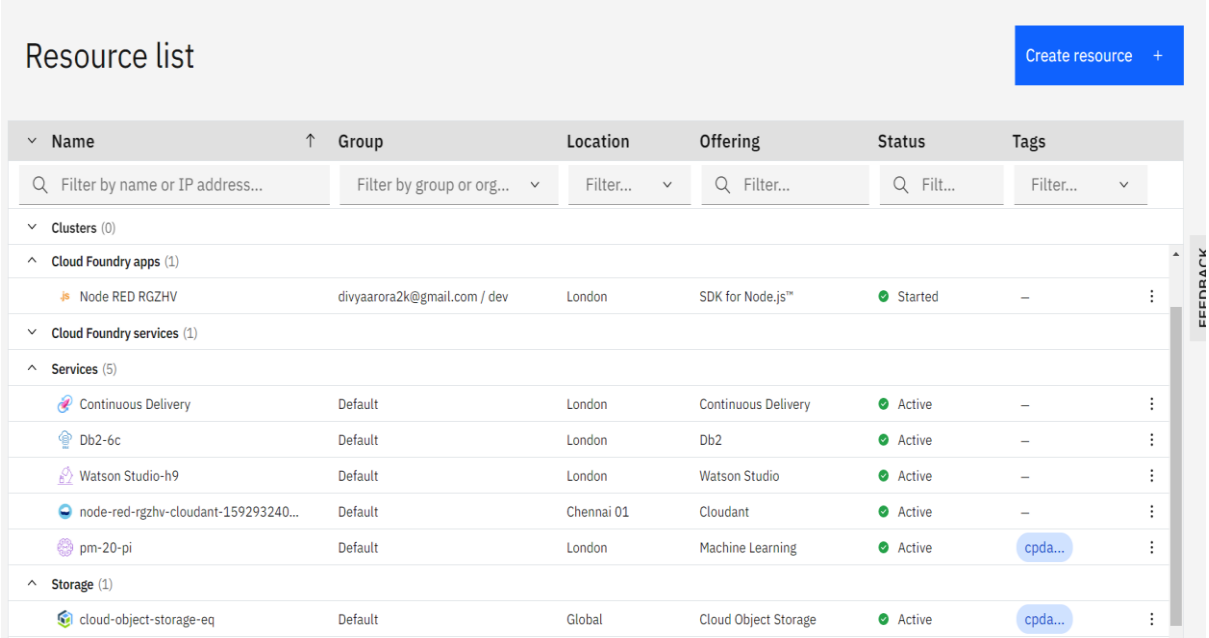
4. EXPERIMENTAL INVESTIGATIONS

- Collection of data set from Kaggle.

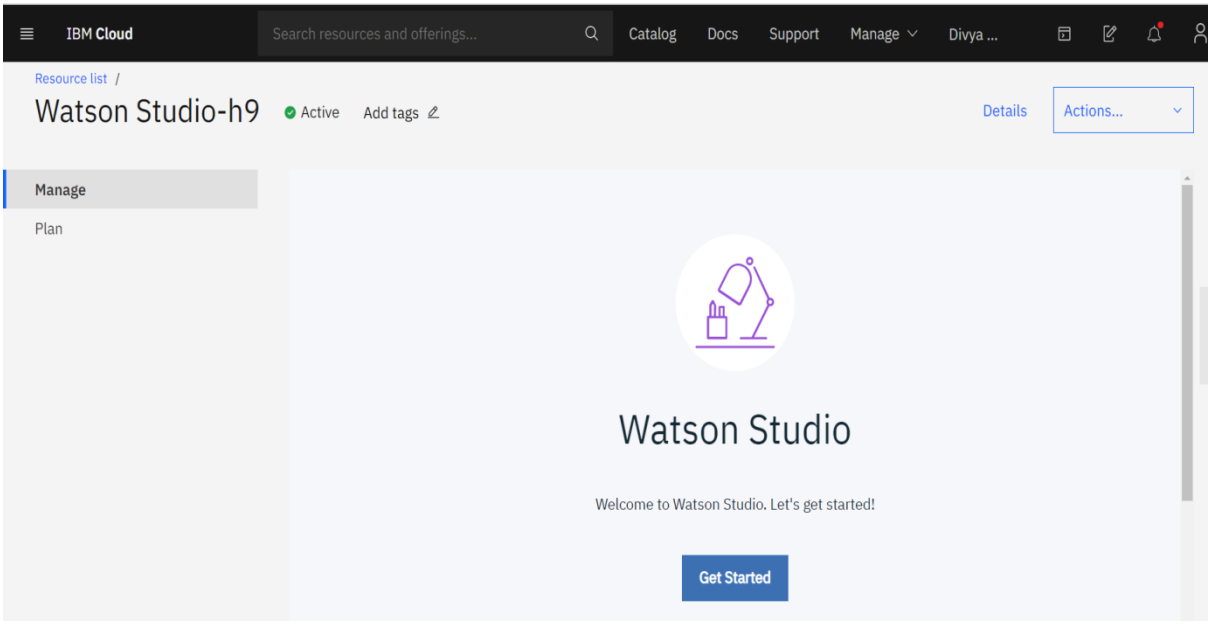
  https://www.kaggle.com/c/predicting-life-expectancy/data

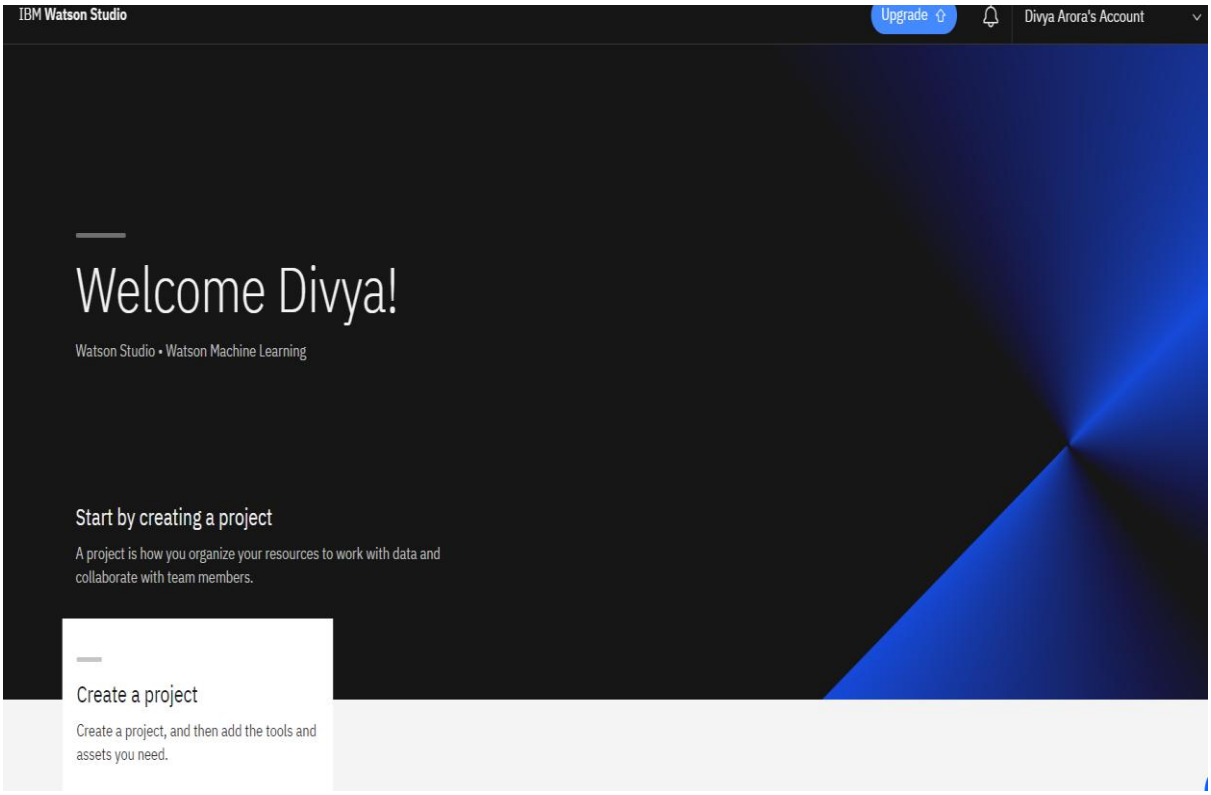
  Build a machine learning model to predict life expectancy

  - Create account on ibm cloud.
  - Create account on ibm Watson studio
  - Add new project in Watson studio.
  - Add datasets to the assets.
  - Then create a new notebook and start writing code.
  - After that create model using ibm machine learning service addition.
  - Deploy model and make end points.
  - After make user interface using node red flow.

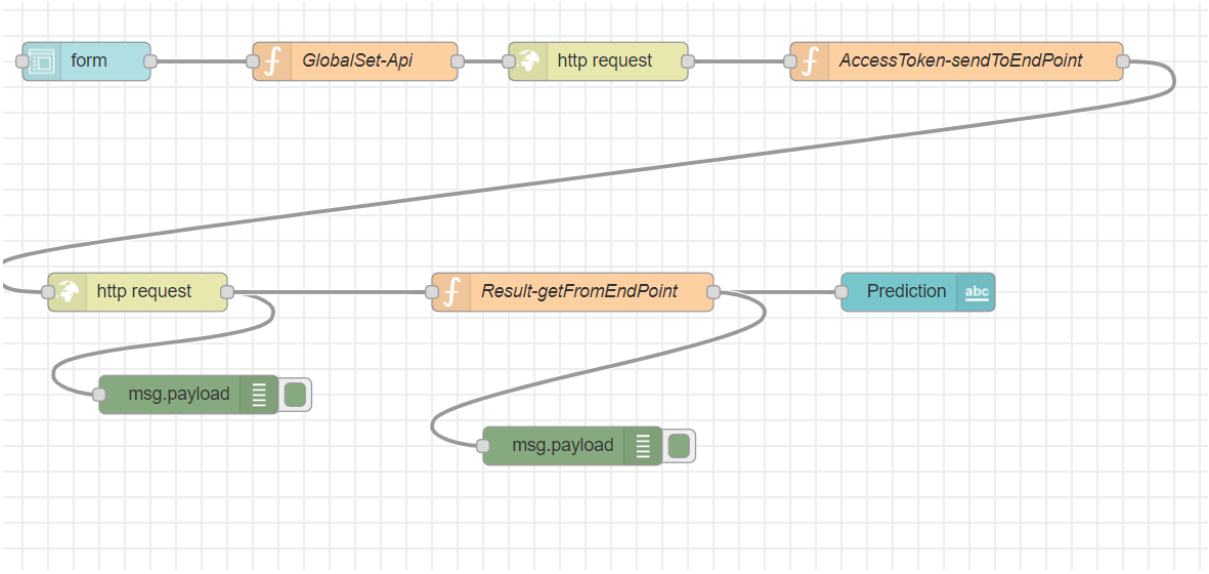


IBM Resource List



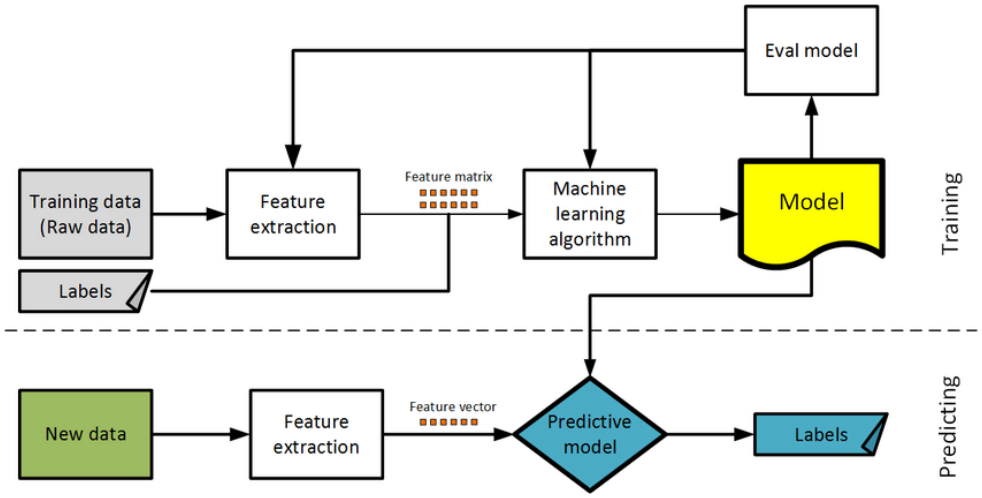


Ibm Watson Studio

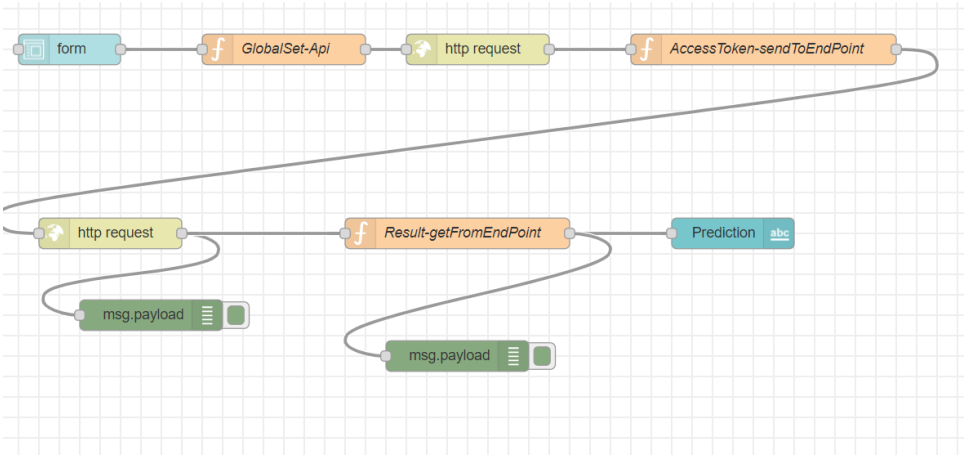


Node Red Flow

5. FLOWCHART



Machine learning Model Training and prediction Flow



Node Red Flowchart

6. RESULT

Life Expectancy UI

https://node-red-rgzhv.eu-gb.mybluemix.net/ui/#!/0?socketid=MJtvUk4-jY-VWxSHAAAM

Life Expectancy

Prediction **65.0324379419277**

Country * Afghanistan

Year * 2005

Status * Developed

BMI * 17.1

Adult_Mortality * 265

Infant_Deaths * 65

Alcohol * 0.01

Percentage_Expenditure * 79

Hepatitis_B * 70

Measles * 600

Under_Five_Deaths * 80

Polio * 50

Total_Expenditure * 8.15

Diphtheria * 40

HIV/AIDS * 0.1

GDP * 600

Population * 33736497

Thinness_10_19_years * 16.4

Thinness_5_9_years * 18

Income_Composition_of_Resour... 0.479

Schooling * 10.2

PREDICT RESET

Correlations between different columns are:

Year	0.170033
Life expectancy	1.000000
Adult Mortality	-0.696359
infant deaths	-0.196557
Alcohol	0.404877
percentage expenditure	0.381864
Hepatitis B	0.256762
Measles	-0.157586
BMI	0.567694
under-five deaths	-0.222529
Polio	0.465556
Total expenditure	0.218086
Diphtheria	0.479495
HIV/AIDS	-0.556556
GDP	0.461455
Population	-0.021538
thinness 1-19 years	-0.477183
thinness 5-9 years	-0.471584
Income composition of resources	0.724776
Schooling	0.751975

- There is negative correlation between Adult Morality and life expectancy.
- There is moderately positive correlation between alcohol and life expectancy. Hence means alcohol drinking people has less life expectancy. Because alcohol affect very much people's health.
- There is moderately affecting BMI on life expectancy. Health affects very much people's health. So BMI should be less for more life.
- Polio drops are very necessary for anybody affecting moderately life of people.
- Some diseases like diphtheria, HIV/AIDS, Hepatitis B badly affecting person's health and hence their life's too.
- There is a strong positive correlation between 'Schooling' and 'Life Expectancy'. This may be because education is more established and prevalent in wealthier countries. This means people in countries with less welfare and less economy spent on schooling have less life.

- Even more with that rich countries people are much healthier that other countries with much less resources and economy.

Similarly more result can be abstract.

7. ADVANTAGES & DISADVANTAGES

Advantages:

- The life expectancy predictor will give important insights and help people achieve good quality of life in future.
- The country can plan and improve various healthcare facilities.
- This is a simple webpage and can be accessed by any citizen of a country to calculate life expectancy of their country and does not required any kind of payment neither for designing nor for using. It is very user friendly interface.
- Advantages of using IBM Cloud: Easy to use and deploy, easy to connect with UI, takes care of large storage space.

Disadvantages:

- Requires internet connection.
- As it depends completely on user, so if user provides some wrong values then it will predict wrong value.
- The model predicts average or less accurate values.
- User input is not saved in any database.
- Input should be in range only to predict accurate values.

8. APPLICATIONS

- To analyze country's people life.
- To analyze country's growth statistics in future years.
- To check if the country need more health facilities.
- If any disease leading to less life of person
- To check how much different affecting person's health

- Can analyze how country's health can be improved by increasing facilities and banning on bad things like alcohol.

9. CONCLUSIONS

Hence, we have developed machine learning model of Prediction of Life Expectancy of a person. User has to fill the form about the details of the fields affecting life expectancy. User can interact with the system using with user interface which is in the form.

10. FUTURE SCOPE

a. More Accurate model

   Model can be more accurate by using best machine learning algorithm for best accuracy. We can increase accuracy by more data and including country column in training.

b. Feature Reduction

   Features can be reduced as this model requires 22 columns to predict life expectancy which is very difficult for normal human being to collect data. We can decrease number of features. Less correlated columns can be removed and can make it more users friendly.

c. Attractive UI

   Can be made more attractive and user friendly by providing more information to users. We can add dropdowns for country and status. We can add range of other values too and defining what word means.

11. BIBLOGRAPHY

- How to create a Kick Off Meeting Agenda
  https://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa69-9fab-40ee-8457-ea0e5df8c8de
- GitHub account creation
  https://github.com/
  https://www.youtube.com/watch?v=7YUTc4Cigc8&feature=youtu.be

- Learn how to add, edit, delete text using Writer.
  https://www.zoho.com/writer/help/working-with-text.html
- IBM Academic Initiative account
  https://my15.digitalexperience.ibm.com/b73a5759-c6a6-4033-ab6b-d9d4f9a6d65b/dxsites/151914d1-03d2-48fe-97d9-d21166848e65/
- Create Node-red application
  https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
  https://www.youtube.com/watch?v=s7wmiS2mSXY&feature=youtu.be
  https://www.w3schools.com/howto/howto_make_a_website.asp
- How to use Watson Studio
  https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html

12. APPENDIX

A. SOURCE CODE

```
#import dataset
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_355c0859d0494b609117a82dc08334a1 =
ibm_boto3.client(service_name='s3',
   ibm_api_key_id='4D2lzd3w7_7TeDF5zFogYDIBocb9CNBELsG8xDDIdo4a',
   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_355c0859d0494b609117a82dc08334a1.get_object(Bucket='smartinternzproject-donotdelete-pr-
```

```
98w9kzqqpseuat',Key='datasets_12603_17232_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 )

df_data_1 = pd.read_csv(body)
df_data_1.tail()
# Data Preprocessing
df_data_1.dtypes
df_data_1.shape
cols = df_data_1.columns
#check NA values
df_data_1.isnull().sum()


#check correlation between cols
col1 = df_data_1["Life expectancy "]
correlation = df_data_1.corrwith(col1, method = 'pearson')
print(correlation)

import numpy as np

#drop NA values rows
df_data_2 = df_data_1.dropna()

#categorical cols are label and encoded by pandas dummies
dummies = pd.get_dummies(df_data_2['Status'])
dummies.tail()
merged = pd.concat([df_data_2,dummies],axis = 'columns')
final = merged.drop(['Status'],axis = 'columns')
print(final.columns)
data = pd.DataFrame(final)

#now dataset are divided into dependent and independent cols y and x
y = data["Life expectancy "]
x_cols = [d for d in data.columns if d != 'Life expectancy 'and d !='Country']
print(x_cols)
x = data[x_cols]
print(x.shape)

#linear regression model is imported and created its object
from sklearn.linear_model import LinearRegression
linear_reg = LinearRegression()

#dataset is divided into train and test
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20,
random_state=42)
```

```
#and now linear_reg model is trained by fitting training x and y
linear_reg.fit(x_train,y_train)

#after training now predict values of x_test and check model performance
y_pred = linear_reg.predict(x_test)
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test, y_pred))
```