PROJECT REPORT

Name: Prerak Shah

Title: **Predicting Life Expectancy of A Country**

Category: **Machine Learning**

WebPage Link:

https://node-red-mudim.eu-gb.mybluemix.net/ui/#!/0?socketid=6UM2x7BQUQVW8uYRAAAC

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. So this is a typical Regression Machine Learning project that leverages historical data to predict insights into the future.

The end product will be a webpage where you need to give all the required inputs and then submit it . Afterwards it will predict the life expectancy value based on your regression technique.

1.2. Purpose

The purpose of the project is to design a model for predicting Life Expectancy rate of a country given various features 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.

2. LITERATURE SURVEY

2.1. Existing Problem

In order to increase the life expectancy of any given country first we need to identify which factors are responsible in determining the life expectancy of a given country. By analyzing the results we can identify those features which play a crucial role, and also we can predict the life expectancy by giving certain inputs like GDP, BMI, HIV/AIDS, Year, Alcohol intake and etc.

2.2. Proposed Solution

Solution for this problem is that we can create a machine learning regression model which can predict the life expectancy of people based on different affecting factors which will help in knowing what to implement for betterment of humans. In order to do so we would take the help of IBM cloud services, Watson studio and Node – red for deploying the model.

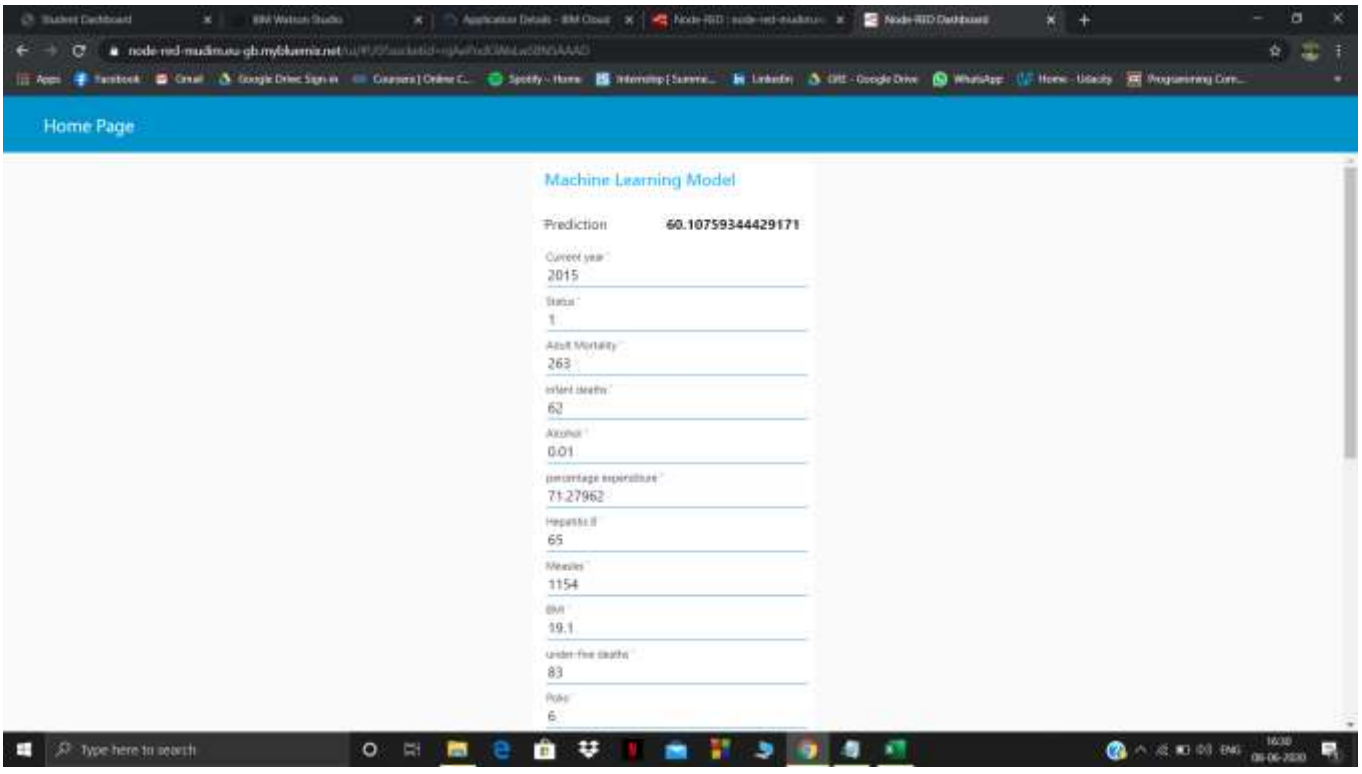
2.2.1. Create IBM cloud Services

- Watson Studio
- Machine Learning resource
- Node-Red

2.2.2. Configure Watson Studio

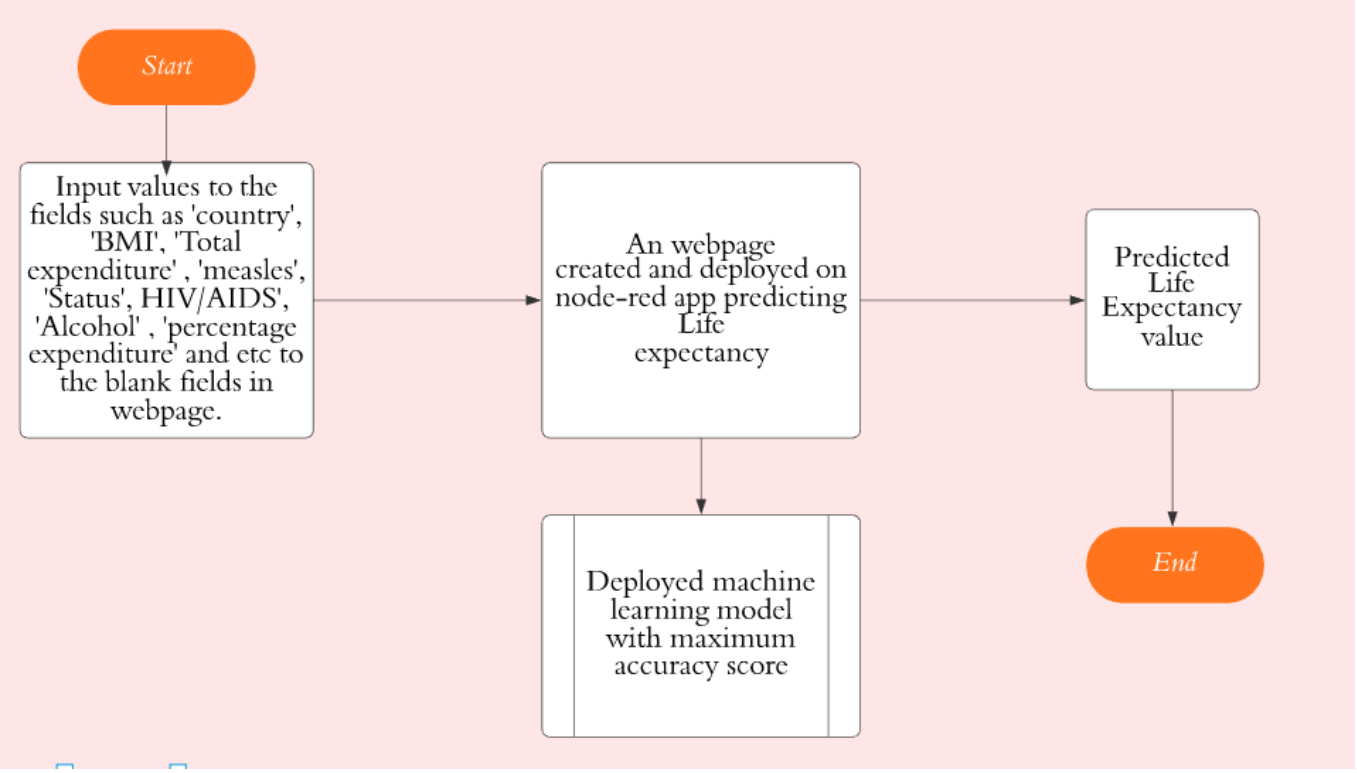
2.2.3. Create Node-Red Flow to connect all services together

Deploy and run Node Red app.
Deploy the Node Red flow. Then copy the link url upto .net/ and paste at a new tab by ui at the end of the url like this . An example screenshot is attached below:



3. THEORETICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. HARDWARE / SOFTWARE DESIGNING

- **Project Requirements**: Python, IBM Cloud, IBM Watson
- **Functional Requirements**: IBM cloud
- **Technical Requirements**: ML, WATSON Studio, Python, Node-Red
- **Software Requirements**: Watson Studio, Node-Red

4. EXPERIMENTAL INVESTIGATIONS

A) IBM Cloud Resource List

It consists of all the resources that we are currently working on. It is the one stop destination for all the things that we are working with.

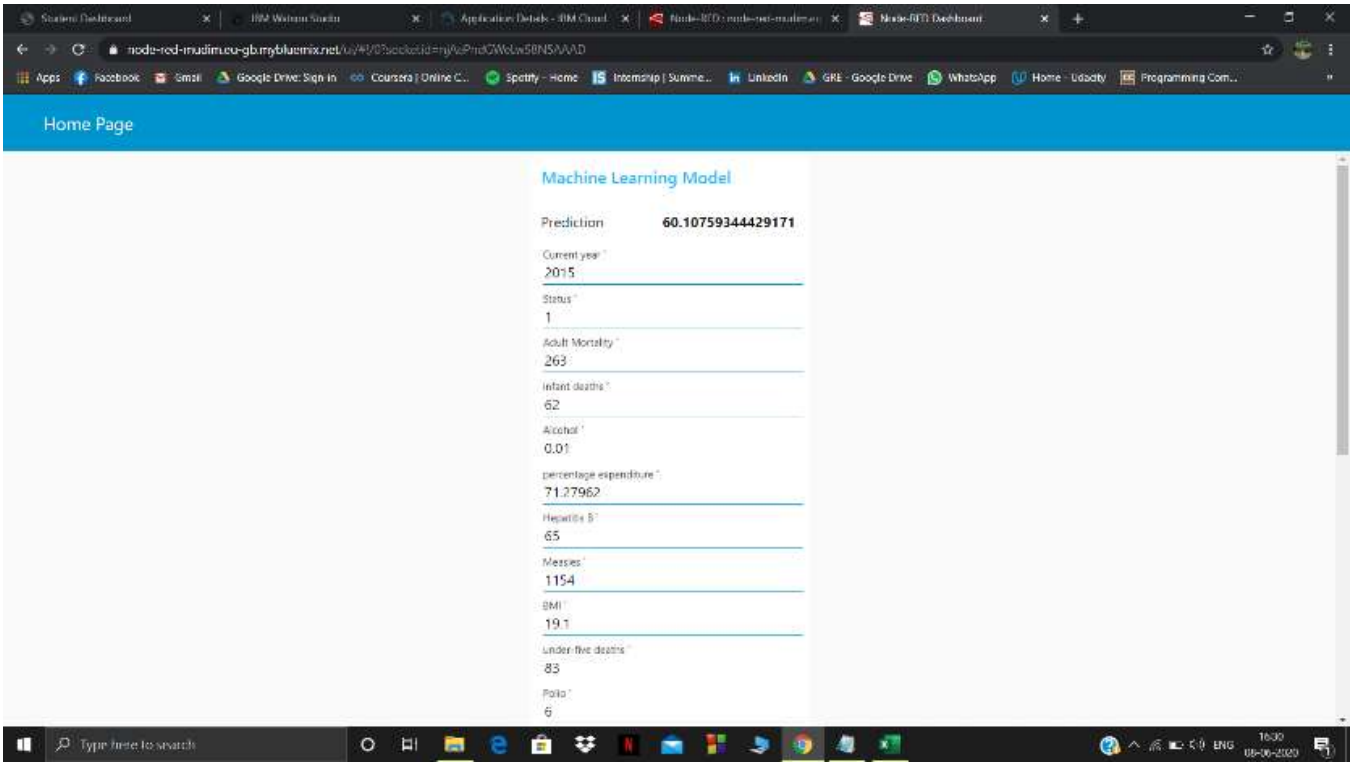
B) IBM Watson Studio

Here we work with the python notebook in order to use a particular model to predict the life expectancy.

C)Node-Red Flow

It is here that we connect all the dots. The connection between backend and front end is done here using the help of node.js.

D) Life Expectancy Prediction UI

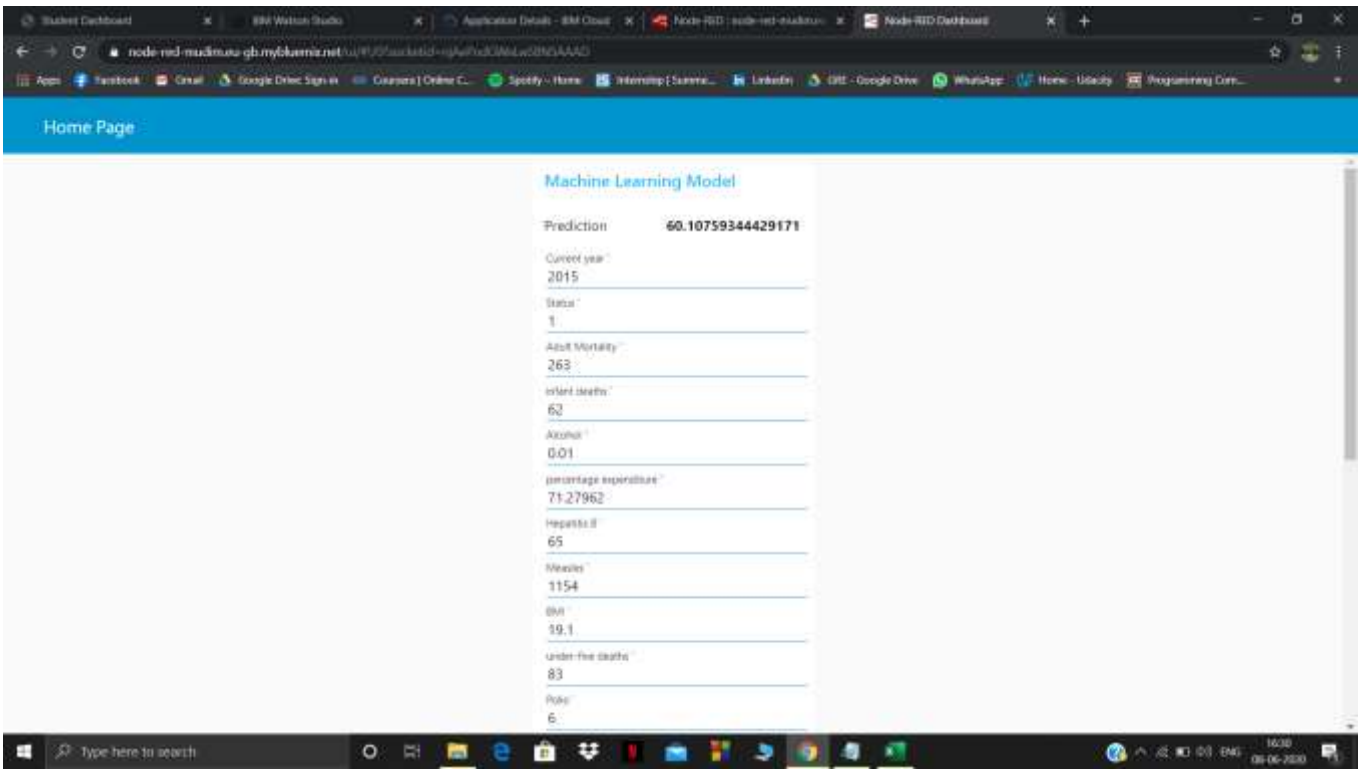


5. FLOWCHART

- The user input all the required values (in the website url) .
- The scoring_endpoint url matches with the deployed model.
- Now these values would be passed to the pre-trained model.
- Then the model would predict the life expectancy value.
- The value predicted is prompted on the website screen.

6. RESULT

This is the Life Expectancy UI.



7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

a) This is a straightforward site page and can be accessed by any resident of a nation to figure life expectancy of their nation and doesn't required any sort of installment. One can directly enter the values into the dashboard.

b) This interface requires no foundation information on the most proficient method to utilize it. It's a straightforward interface and just request required qualities and anticipate the yield.

c) Using PCA we can also determine which factors affect life expectancy most.

d) By utilizing this crucial information the governments of respective countries can make changes in the policies in order to increase their life expectancy.

DISADVANTAGES:

a) Wrong Prediction: As it depends completely on user, so if user provides some wrong values then it will predict wrong value.

b) Still, the accuracy of the model is not that great. So, sometimes the result fetched may nit be accurate.

8. APPLICATION

a) It can be used to monitor health inequalities of a country.
b) It can be used to develop statistics for country development process.
c) The government can make amendments to their existing policies in order to improve their life expectancy.
d) It can be used to analyze the factors for high life expectancy.
e) The countries with high life expectancy can be taken as a model for others, and help others in achieving such excellent results.

9. CONCLUSION

The project we created here will serve the human society for their betterment , we have made a regression model using ml which is connected with ui by node red and will interact with user and predict average life expectancy of person there.

10. FUTURE SCOPE

Future Scope of the Model can be:

a) Appealing UI

It is a simple webpage only asking inputs and predict output. In future I have decided to make it more user friendly by providing some useful information about the country in the webpage itself so that user does not need to do any kind of prior research for the values.

b) Reduce the number of features

It requires much more data about 21 columns to be known prior for predicting life expectancy which can be again difficult for a normal user to gather such data, so I have decided to do some kind of feature reduction. I can also use PCA to identify which features are more important. Later the less important features can be dropped.

11. BIBLIOGRAPHY

Sources :

- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service
- https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform
- https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
- https://nodered.org/
- https://github.com/watson-developer-cloud/node-red-labs

APPENDIX: Source Code

1. **Python Notebook**

```
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_86698b2566e248a196fbb3aaf10d5467 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='FUmtkroIw59zK2vA5ctKn39R_-vLGR_rG6EzuNRpwgCg',
```

```
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_86698b2566e248a196fbb3aaf10d5467.get_object(Bucket='predictlifeexpectancy-
donotdelete-pr-cmtqwuhzvyq7fn',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 )

df = pd.read_csv(body)
df.head()

import numpy as np

df=df.drop(['Country'],axis=1)
housing_map = {'Developing': 1, 'Developed': 0}
df['Status'] = df['Status'].map(housing_map)

df.fillna(df.mean(), inplace=True)

x=df.drop(['Life expectancy '],axis=1)
y=df['Life expectancy ']
print(x.head())
print(y.head())

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4,random_state=101)

from sklearn.linear_model import LinearRegression
model=LinearRegression()

model.fit(x_train,y_train)

y_pred=model.predict(x_test)

print(y_pred)

from sklearn.metrics import r2_score
from sklearn import metrics

score=r2_score(y_test,y_pred)
print("Score is :",score)

print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test,y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

!pip install watson-machine-learning-client

from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_creds={
  "apikey": "-IhIljfmEsdnRf1pt79aQIHBMNZci9n750dNX72CDCKQ",
  "instance_id": "b8d72273-60b0-4920-b67e-0c108a784c8a",
  "url": "https://us-south.ml.cloud.ibm.com"
}

client = WatsonMachineLearningAPIClient( wml_creds )
```

```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Prerak",
            client.repository.ModelMetaNames.AUTHOR_EMAIL: "prerak.sot17@sot.pdpu.ac.in",
            client.repository.ModelMetaNames.NAME: "Pred_Life_Expectancy"}

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

published_model_uid = client.repository.get_model_uid(model_artifact)
published_model_uid

deployment = client.deployments.create(published_model_uid, name="Pred_Life_Expectancy")

scoring_endpoint = client.deployments.get_scoring_url(deployment)
```