



Summer Internship Project Report
Predicting Life Expectancy using Machine Learning
21/05/2020 - 19/06/2020

Submitted by:

Anuj Kumar Pradhan

College of Engineering and Technology, Bhubaneswar

Email : anujpradhan208@gmail.com

Index

1. INTRODUCTION

1.1 Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 Proposed solution

3. THEORITICAL ANALYSIS

3.1 Block diagram

3.2 Hardware / Software designing

4. EXPERIMENTAL INVESTIGATIONS

5. FLOWCHART

6. RESULT

7. ADVANTAGES AND DISADVANTAGES

8. APPLICATIONS

9. CONCLUSION

10.FUTURE SCOPE

11. BIBILOGRAPHY

APPENDIX

A. Source code

1. INTRODUCTION

1.1 OVERVIEW

Life expectancy is a focal point that acts as a passive root as well as an active root in expansion of industries all over the world. Health projection and possible future framework can deliver as crucial aids in far-reaching strategy and expenditure in health specifically in respect of creating variant possibility, their probable effects, and the respective definiteness related with each choice. It can assist the corporations to hire employees as well as governments to watch over the features and lead the way with enhancement to widen the citizens lifetimes of their countries. To have a complete idea on actual orientations in health and operators of health is vital for managing deep-rooted expenditure and policy execution. At the current kind of situation there is a demand for an individual who can convey the prediction regarding how additional years a person can reside and their topographical life span. As the technology is on the up and up than ever and computers are grasping skill finer and mastering the case summaries, its power can be utilized to await Life span of an individual constructed on a few inputs of the area/country they reside in such as: either if they reside in a advanced country or in a evolving one, how many infant dies there and many more, what's the Adult Morality of that country, what amount of alcohol they take.

Project Requirements:

Functional Requirements:

Predicting the life expectancy rate of a country.

Technical Requirements:

Python, Machine Learning IBM Cloud, IBM Watson.

Hardware Requirements:

Processor: i3 7th gen or higher

Speed: 2GHz or more

HARD DISK: 10 GB or more

RAM: 4 GB or more

Software Requirements:

Watson Studio, Node-Red

➤ PURPOSE

Building a machine learning model for the prediction of life expectancy.

The main purpose of the project is to design a model for predicting Life Expectancy rate of a country given various features such as year, Adult Mortality, education, 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

Rising incomes have increased life expectancy across the globe, new data released by the World Health Organisation (WHO) show, even as it warns that poor countries continue to perform badly and are still lagging below the global average in the measurement of wealth and longevity. “Despite the largest gains in both indicators being due primarily to the progress made in reducing child mortality and fighting infectious diseases, low-income and lower-middle-income countries continue to suffer from the poorest overall health outcomes,” WHO says in the report that tracked data between 2000 and 2016. Kenya is ranked as a lower middle-income country, meaning that it is among nations that need to invest more to improve incomes and life expectancy. According to WHO, life expectancy at birth for Kenyan men stands at 64.4 years, compared to 68.9 for women. The average for both sexes is 68.7 years. This is about six years above the average for poor countries, which stood at 62.7 years. However, it is about 12 years lower than life expectancy at birth in high-income economies where expectancy stands at 80.8 years at birth. “Low income countries have seen the biggest recent gains in life expectancy: On average in those countries, it rose by 21 percent between 2000 and 2016 (or 11 years), compared with 8 percent (five years) globally and 4 percent (three years) in high-income countries, says the survey released last week. The report also notes that whereas there was a general increase in the quality of health services globally, poor countries recorded the biggest gains in this respect. And whereas longer life expectancy and improved health services are indications of improved quality for life for individuals, it also has policy and fiscal implications for governments and pension funds. When citizens have access to better health care, they live longer, meaning that retired civil servants will continue to exert pressure on tax-funded pension funds while private sectors workers will draw from their retirement funds for longer or suffer from old age poverty in instances where their pension payment is limited. Older people are also likely to spend more, including their own disposable income,

on drugs and hospital visits. “Out-of-pocket health spending can also push people into poverty,” the report warns. “Most of the people pushed into extreme poverty (surviving on less than \$2 per person per day) by out of-pocket payments live in lower-middle-income countries and South-East Asia”. Sadly, the number of people who have become poorer as a result of out-of-pocket spending on medicine and treatment is on a sharp increase globally. However, it is not all doom and gloom for poor countries as the report indicates that fewer children are dying between birth and the age of five while the deaths of mothers during childbirth is generally on the decline. However, there is still a disproportionate number of infants who die from respiratory and diarrheal diseases, both of which are either treatable or preventable. “In high-income countries, 80 percent of new borns are expected to live beyond the age of 70,” says the report. In adulthood, however, the challenges increase since healthcare systems globally have not done enough to reduce deaths arising from lifestyle and non-communicable diseases like hypertension and cancer. This is despite progress made in combating communicable diseases. By contrast, heart disease, lung cancer and suicides are the three top causes of premature death in rich countries.

2.2 Proposed solution

Steps:

- a) Create IBM cloud services
- b) Configure Watson Studio
- c) Create Machine Learning Notebook
- d) Save and Deploy Model in Notebook
- e) Create Node-Red Flow to connect all services together
- f) Deploy and run Node-Red app

2.2.1. Create IBM cloud Services

- Watson Studio
- Machine Learning model instance
- Node-Red

2.2.2. Configure Watson Studio

After creation of all the services, go to the resource list and launch Watson studio and get started with Watson studio. Then we have to create an empty project and add machine learning resource as associated services in settings. Create a token as editor type. We have to create empty Jupyter notebook into Assets and add dataset.

Later then go to notebook and write your code to build model and get the scoring endpoint URL.

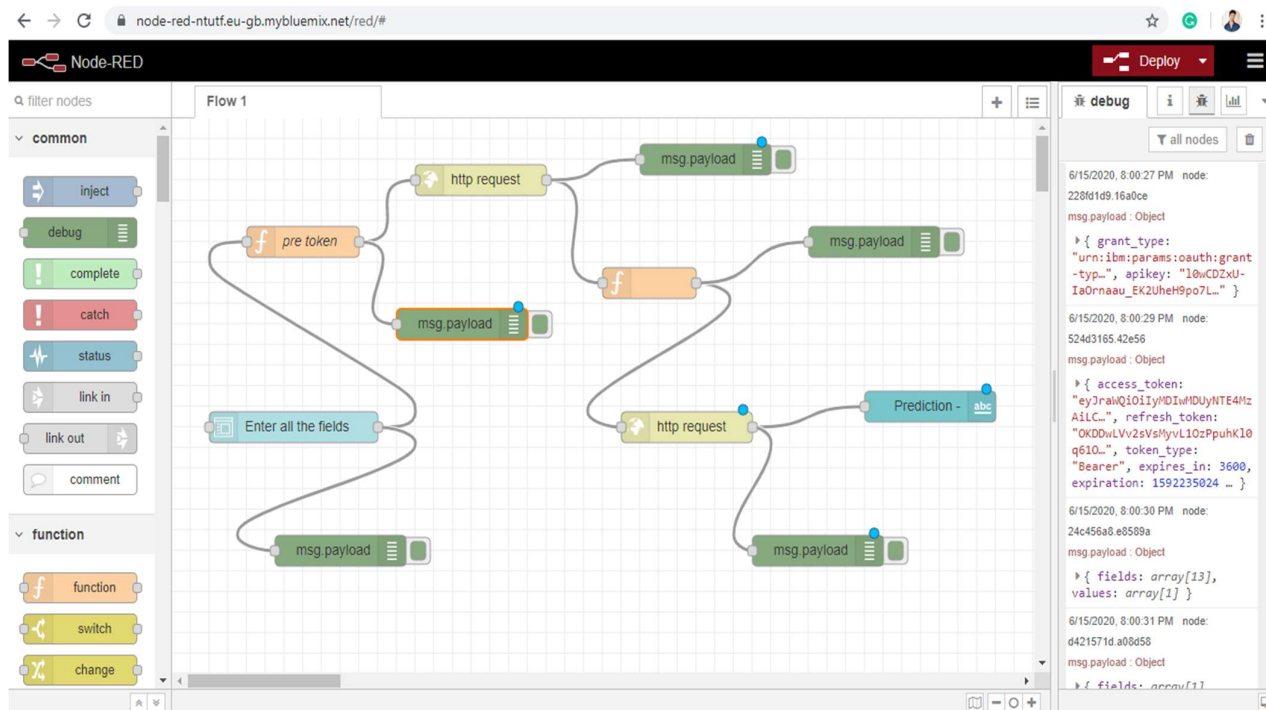
Steps for notebook:

- Install `Watson_machine_learning_client`
- Import necessary libraries
- Import Dataset
- Descriptive Analysis of Data
 - Removing unusual species in column names using rename function.
 - Replacing nan values if any with their mean values.
 - Plotting a heatmap to check if dimensional reduction can be performed.
- Calculating P-value to know the impact of features on the target value and remove the features with higher p-value.
- Train and Test
 - The dataset was split into two parts i.e. Input and Output. As Life Expectancy needs to be predicted so it is to be treated as output and all other columns are treated as input variables.
 - Afterwards as we need regression technique to build our model so each and every column needs to be numeric. So, then we check for numeric and categoric columns.
 - Then we use LabelEncoder to convert categoric variables to numeric.
 - Or we can standardize the numeric and categoric columns using pipelining.
 - At first independent pipelines for both the parts were designed then they were joined using column transform.
 - After that a regressor pipeline was designed using the regression technique.
 - We apply different regression techniques including MultiLinear Regression, Decision Tree Regression, Random Forest Regression on our dataset.
 - So I have used Random Forest Regressor technique of `sklearn.ensemble` as my regression algorithm as it gives best accuracy.
 - Then train and test split were performed and 80% of dataset were trained data and 20% were test data.

- Then dataset was fitted and predicted.
- Then error and accuracy were estimated and the mean squared error is 3.4 whereas the R2_score or accuracy is 96.07%.
- Model Building and Deployment
 - At first the machine learning service credentials was stored in a variable and passed into WatsonMachineLearningAPIClient.
 - Then the model was built and stored in model_artifact.
 - Then the model was deployed and scoring_endpoint URL was generated.

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

- Go to Node-Red Editor from resource list.
- Install node-red Dashboard from manage palette.
- Now create the flow with the help of following node.
 - Inject
 - Debug
 - Function
 - Ui_Form
 - Ui_Text



- Deploy and run Node Red app.

Deploy the Node Red flow. Then go to the dashboard and click on the UI url.

LIFE EXPECTANCY PREDICTION

Enter all the fields

year *

2015

status *

0

adult_mortality *

121

hepatitis_b *

94

measles *

17

bmi *

59

polio *

95

total_expenditure *

6.5

diphtheria *

94

hiv/aids *

0.1

thinness_10-19_years *

1

income_composition_of_resources *

0.8

schooling *

12

SUBMIT

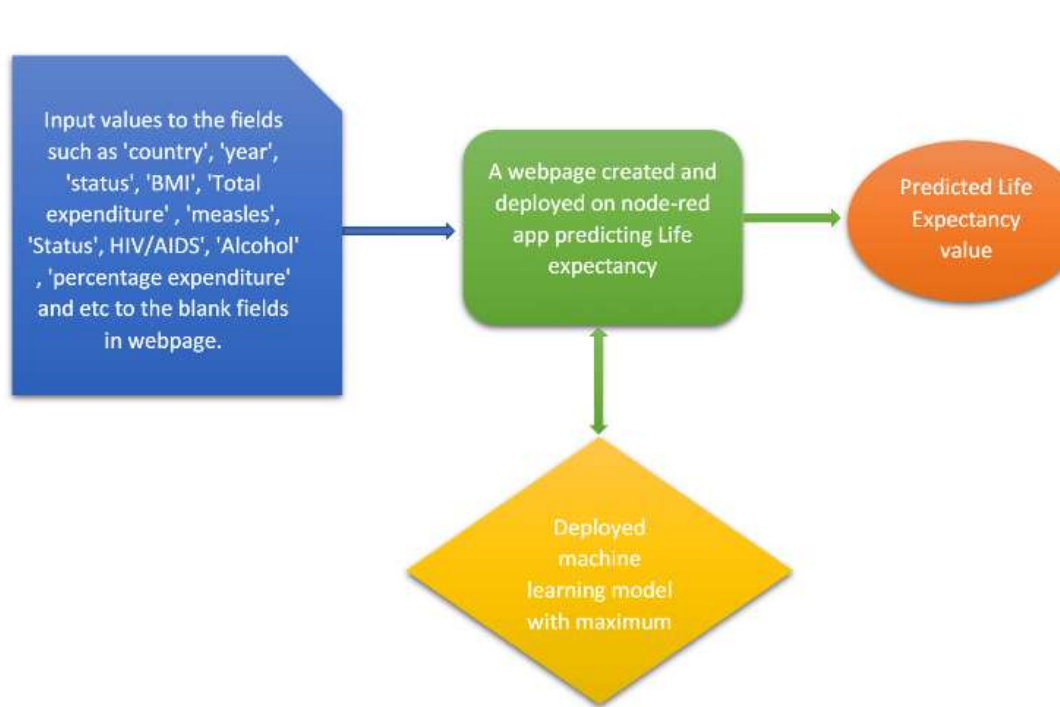
CANCEL

Prediction -

[76.218]

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:

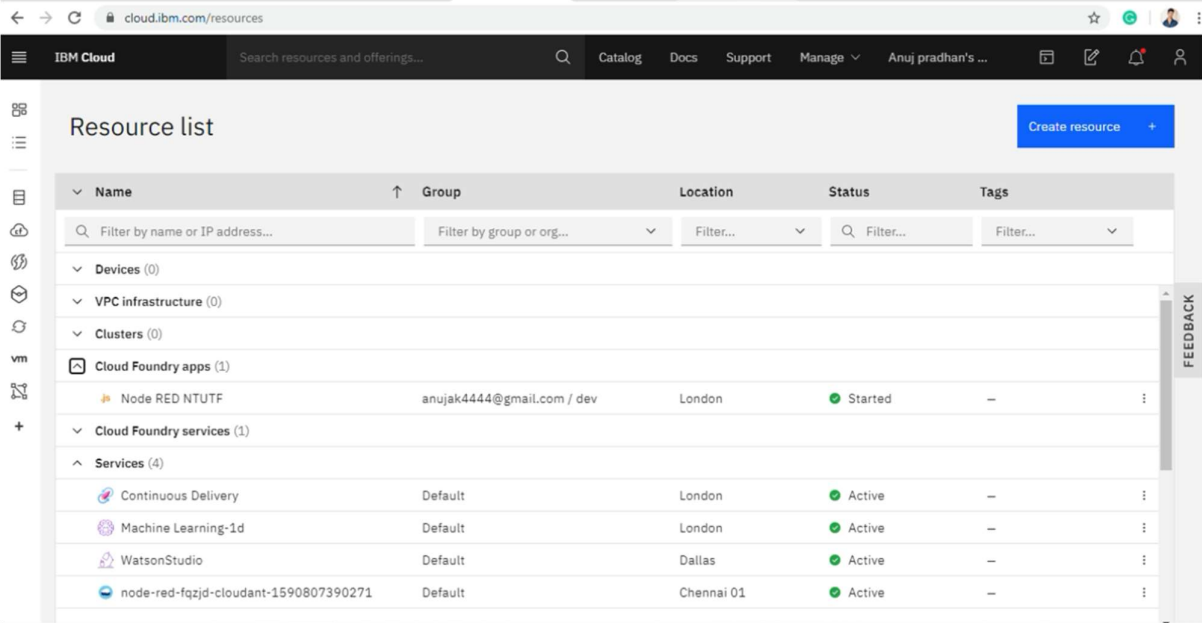
- ML
- WATSON Studio
- Python, Node-Red

Software Requirements:

- Watson Studio
- Node-Red

4. EXPERIMENTAL INVESTIGATIONS

A) IBM Cloud Resource List



Resource list

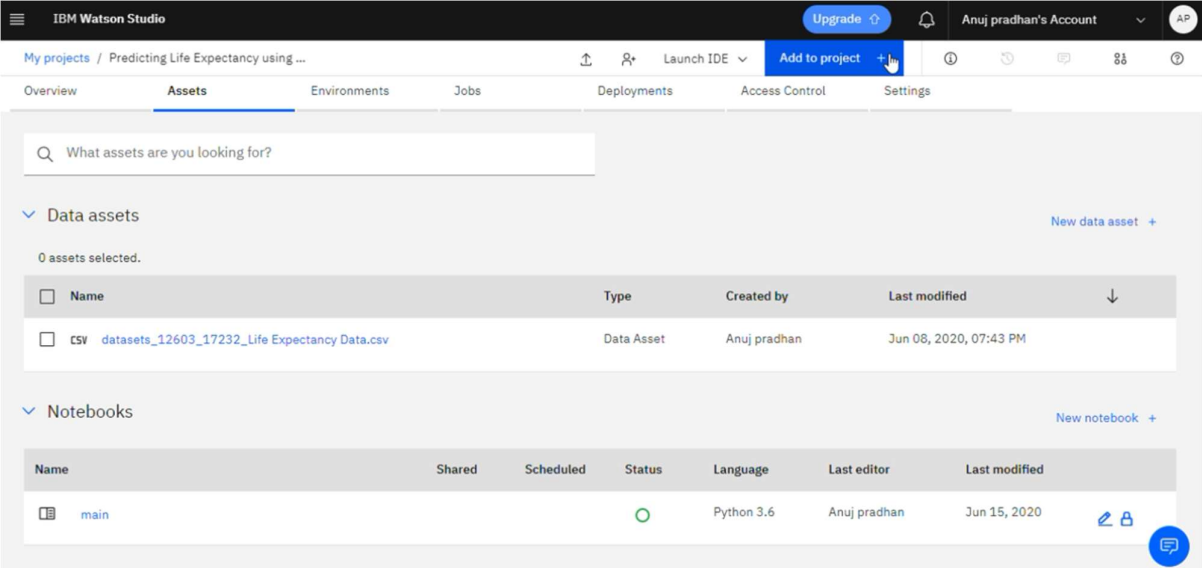
Create resource +

Name	Group	Location	Status	Tags
Filter by name or IP address... Filter by group or org... Filter... Filter... Filter...				
Devices (0)				
VPC infrastructure (0)				
Clusters (0)				
Cloud Foundry apps (1)				
Node RED NTUTF	anujak4444@gmail.com / dev	London	Started	—
Cloud Foundry services (1)				
Services (4)				
Continuous Delivery	Default	London	Active	—
Machine Learning-1d	Default	London	Active	—
WatsonStudio	Default	Dallas	Active	—
node-red-fqzjd-cloudant-1590807390271	Default	Chennai 01	Active	—

FEEDBACK

B) IBM Watson Studio

C) IBM Cloud Project Details



IBM Watson Studio

Upgrade + Anuj pradhan's Account

My projects / Predicting Life Expectancy using ... Launch IDE + Add to project +

Overview Assets Environments Jobs Deployments Access Control Settings

What assets are you looking for?

Data assets

New data asset +

0 assets selected.

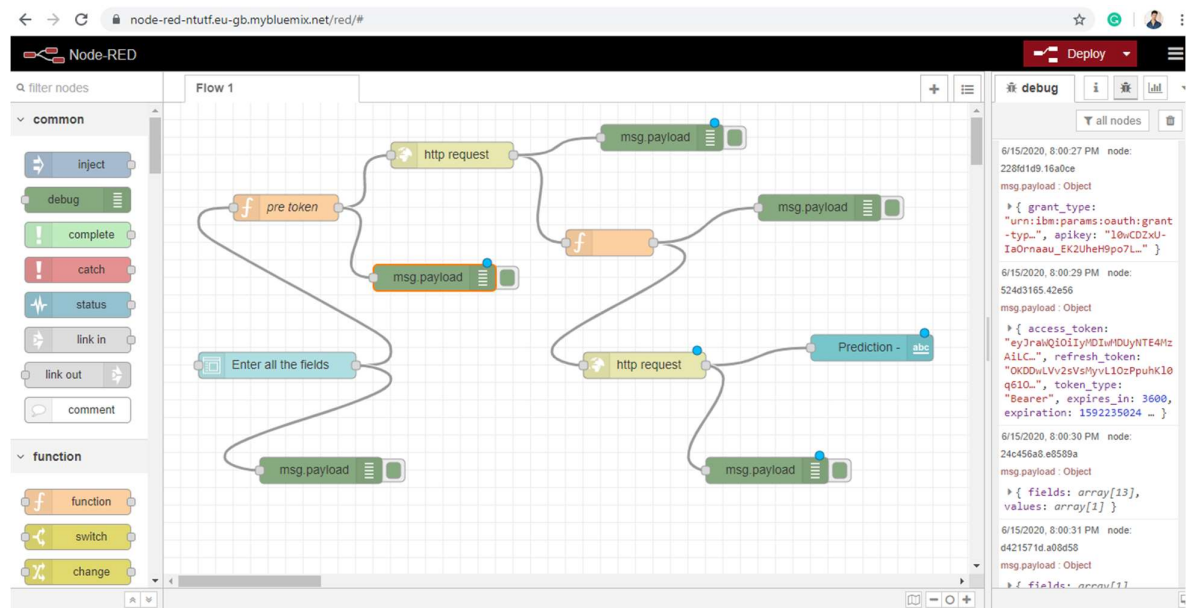
Name	Type	Created by	Last modified
CSV datasets_12603_17232_Life Expectancy Data.csv	Data Asset	Anuj pradhan	Jun 08, 2020, 07:43 PM

Notebooks

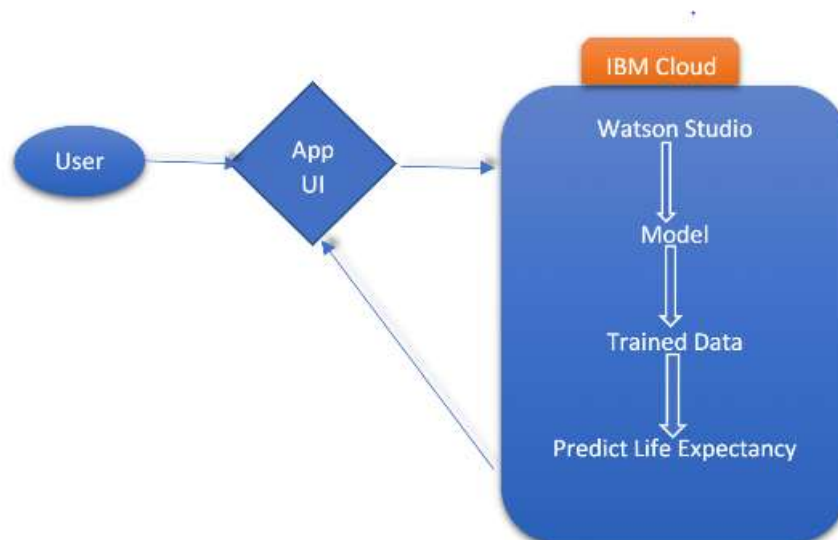
New notebook +

Name	Shared	Scheduled	Status	Language	Last editor	Last modified
main				Python 3.6	Anuj pradhan	Jun 15, 2020

D) Node-Red Flow



5. FLOWCHART



6. RESULT

LIFE EXPECTANCY PREDICTION

Enter all the fields

year *

2015

status *

0

adult_mortality *

121

hepatitis_b *

94

measles *

17

bmi *

59

polio *

95

total_expenditure *

6.5

diphtheria *

94

hiv/aids *

0.1

thinness_10-19_years *

1

income_composition_of_resources *

0.8

schooling *

12

SUBMIT

CANCEL

Prediction -

[76.218]

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Random forests is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It can handle thousands of input variables without variable deletion.

DISADVANTAGES

- Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.
- For data including categorical variables with different number of levels, random forests are biased in favour of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

8.APPLICATIONS

- Insurance companies can use this for prediction.
- Government can utilize this for the steps towards human welfare.
- It can be used to analyse the factors for high life expectancy.
- Common people can use this to be aware about their health.

9.CONCLUSION

The interface developed is user friendly and can be used by anyone and will be useful to predict the life span of an individual or predicting life expectancy rate of a country given on some features values such as year, GDP, education, alcohol intake of people in the country, expenditure on healthcare system.

10. FUTURE SCOPE

This application has very bright future and large number of applicable cases, it must need to be upgraded and taken into account in school as well as college syllabus for the welfare and knowledge purpose. This provides insights in various factors and their levels required to keep the life expectancy rate as high as expected. It can be used to suggest good health practices and life style to the users based on their daily activities and provide suggestions for exercises for improving their health. Pharmaceutical companies can check which diseases impact more people and therefore impact life expectancy and based on this

manufacture medicine. We can also say it is a time machine which predict the life of someone who haven't born yet on the factors of his/her country's Adult Mortality, Population, Under 5 Deaths, Thinness 1-5 Years, Alcohol, HIV, Hepatitis B, GDP, Percentage Expenditure, and others.

11.BIBILOGRAPHY

<https://www.youtube.com/watch?v=LOCkV-mENq8&feature=youtu.be>

<https://www.youtube.com/watch?v=DBRGLAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L>

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

<https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service>

<https://nodered.org/>

<https://bookdown.org/caoying4work/watsonstudio-workshop/auto.html#add-asset-as-auto-ai>

12.APPENDIX

Source code –

1. main.ipynb Notebook

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
from sklearn.ensemble import RandomForestRegressor

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 i
ncludes your credentials.
# You might want to remove those credentials before you share the notebook.
client_7d161f2cdceb45adb285134567419295 = ibm_boto3.client(service_name='s3
',
    ibm_api_key_id='1AXZsItUms5S6Vio0luy9qn61n4Hiiu7a-kfRH0BzSOv',
    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_7d161f2cdceb45adb285134567419295.get_object(Bucket='predictin
glifeexpectancyusingmach-donotdelete-pr-083yhjtifbr2k',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.head()
df = df_data_1
df.head()
```

```
df.rename(columns = lambda x: x.strip().replace(' ', '_').lower(),
          inplace=True)
df.rename(columns = {'thinness__1-19_years':'thinness_10-19_years'},
          inplace=True)

plt.figure(figsize=(25,10))
df.groupby('country')['life_expectancy'].mean().head(30).plot.bar()
```

```
df.isnull().sum()
```

```
#list of all columns containing NA values
na_cols = ['life_expectancy', 'adult_mortality', 'alcohol', 'hepatitis_b',
           'bmi', 'polio', 'total_expenditure', 'diphtheria', 'gdp', 'population',
           'thinness_10-19_years', 'thinness_5-9_years',
           'income_composition_of_resources', 'schooling']
```

```
#filling NA cells with overall mean
for col in na_cols:
    df[col].fillna(df[col].mean(), inplace=True)
```

In [8]:

```
train_data=df.drop(['life_expectancy'],axis=1)
train_labels = df['life_expectancy']

corr = train_data.corr()
plt.figure(figsize=(20,12))
print('**correlation table**')
sns.heatmap(corr,square=True,cmap="BuPu",annot=True)
```

```
col_corr = []
for i in range(len(corr.columns)):
    for j in range(i):
        if (corr.iloc[i, j] >= 0.9) and (corr.columns[j] not in col_corr):
            colname = corr.columns[i] # getting the name of column
            if colname not in col_corr:
                col_corr.append(colname)
            if colname in train_data.columns:
                del train_data[colname]
```

```
#'under-five_deaths', 'thinness_5-9_years' columns are removed
```

In [10]:

```
X = train_data.iloc[:,3:].values
y = train_labels.values
```



```
import statsmodels.api as sm

mod = sm.OLS(y,X)
fii = mod.fit()
p_values = fii.summary2().tables[1]['P>|t|']

print(p_values)
```

```
columns = train_data.columns
reduce = []
for i in range(len(p_values)):
    if(p_values[i]>0.05):
        print(columns[i+3],p_values[i])
        reduce.append(columns[i+3])

#drop columns with higher p values
train_data=train_data.drop(reduce,axis=1)

columns = train_data.columns
print(columns)
```

```
train_data = train_data.drop(['country'],axis=1)
```

```
X = train_data.iloc[:,:].values
y = train_labels.values
```

```
from sklearn.preprocessing import LabelEncoder
# labelencoder_X = LabelEncoder()
# X[:,0] = labelencoder_X.fit_transform(X[:,0])
labelencoder_X_2 = LabelEncoder()
X[:,1] = labelencoder_X_2.fit_transform(X[:,1])
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
```

In [13]:

```
reger = RandomForestRegressor(n_estimators= 50, random_state = 0)
reger.fit(X_train,y_train)
```

```
y_pred = reger.predict(X_test)

#mse
mse=mean_squared_error(y_test, y_pred)
#r2
```

```
r2=r2_score(y_test,y_pred)
#mae
mae=mean_absolute_error(y_test,y_pred)
#rmse
rmse=np.sqrt(mae)
{'mse':[mse], 'r2':[r2], 'mae':[mae], 'rmse':[rmse]}
```

```
#accuracy
error = []
for i in range(len(y_test)):
    a=abs(y_test[i]-y_pred[i])
    error.append(a)
acc=100-(sum(error)/len(y_test))
acc
```

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_credentials={
    "apikey": "l0wCDZxU-IaOrnaau_EK2UheH9po7LFNHLTHkRkRc7Qd",
    "instance_id": "d0f2c189-ad40-47ad-a119-ea6d88a5b15c",
    "url": "https://eu-gb.ml.cloud.ibm.com"
}

client = WatsonMachineLearningAPIClient( wml_credentials )

model_props = {client.repository.ModelMetaNames.AUTHOR_NAME:
    "ANUJ KUMAR PRADHAN",
                client.repository.ModelMetaNames.AUTHOR_EMAIL:
    "anujpradhan208@gmail.com",
                client.repository.ModelMetaNames.NAME:
    "LIFE EXPECTANCY USING MACHINE LEARNING"}

model_artifact =client.repository.store_model(reger, 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="lifeexpectancy")

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