**SMARTINTERNZ INTERSHIP PROJECT REPORT**

**Internship Title :** Predicting Life Expectancy using Machine Learning-SB6071

|  |  |  |
| --- | --- | --- |
| **Project Id :** SPS\_PRO\_215 |  |  |
| **Project Title :** Predicting Life Expectancy using Machine Learning  **Name** : Videsh Loya |  |  |

# Contents

1. INTRODUCTION
   1. Overview
   2. Purpose
2. LITERATURE SURVEY
   1. Existing problem
   2. Proposed solution
3. THEORITICAL ANALYSIS
   1. Block diagram
   2. Software Requirements
4. EXPERIMENTAL INVESTIGATIONS
5. FLOWCHART
6. RESULT
7. ADVANTAGES & DISADVANTAGES
8. APPLICATIONS
9. CONCLUSION
10. FUTURE SCOPE
11. BIBILOGRAPHY

APPENDIX

A. Source code

# INTRODUCTION

* 1. **Overview**

Life expectancy is one of the most important factors in end-of-life decision making.

The main objective of the project is to predict the life expectancy of a person depending on several factors based on an individual or the residing country. Factors like the GDP of the country, health care facility system, quality of life, mental and physical illness, age, gender, education and other regional, demographic and economic factors are considered to predict the lifespan of the person using machine learning algorithms.

# Purpose

The purpose is to predict Life Expectancy by looking at the positive and negatively correlated factors to improve the Life Quality. By making changes in lifestyle, a person can live a long, healthy and good quality life. This will also beneﬁt the country by increasing manpower that will contribute to the economical growth. We should take full advantage of this new era advanced technology to improve the future by predicting it in the present.

# LITERATURE SURVEY

* 1. **Existing Problem**

As we all know, Life expectancy is one of the most important factors in end-of-life decision making. So, using the certain factors like Schooling, GDP, Adult Mortality Rate, Child Date, etc. life expectancy is predicted. All the factors are negatively or positively correlated.

When you are deciding when to start receiving retirement beneﬁts, one important factor to take into consideration is how long you might live. These country dependent factors can also be an important feature to predict the life expectancy of an individual. So we need more data to predict more accurately.

# 2.2 Proposed Solution

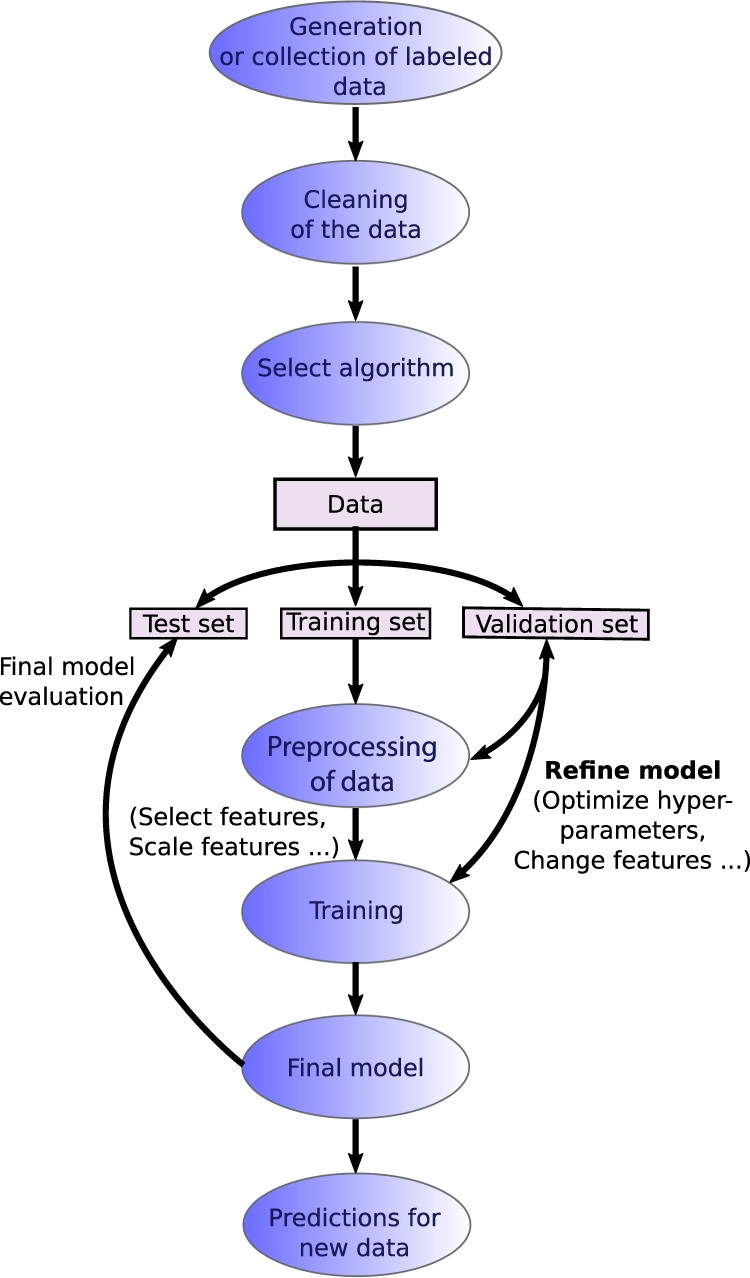
Using this model, life expectancy of a person can be predicted by taking some input features from the user.

Life Expectancy depends on the following features-

* Country
* Status
* Life Expectancy
* Adult Mortality
* Alcohol
* percentage expenditure
* Hepatitis B
* Measles
* BMI
* under-ﬁve deaths
* Polio
* Total expenditure
* Diphtheria
* HIV/AIDS
* GDP
* Population
* thinness 1-19 years
* thinness 5-9 years
* Income composition of resources
* Schooling

# Theoretical Analysis

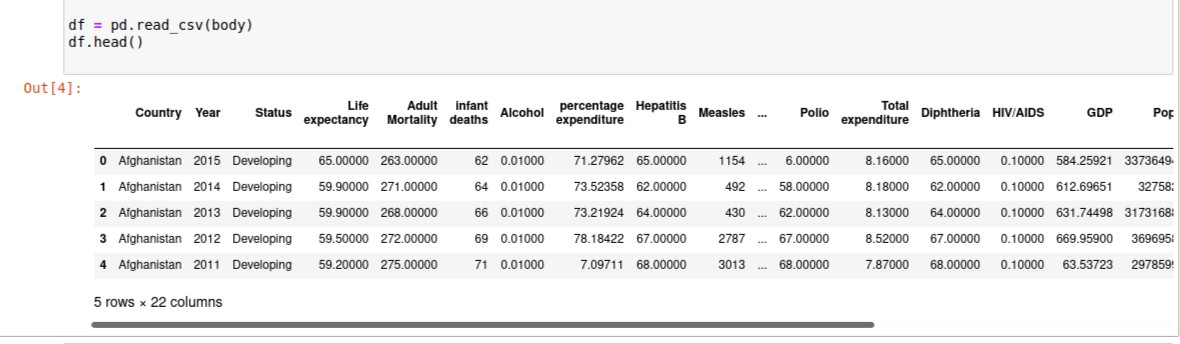
* 1. **Block Diagram**

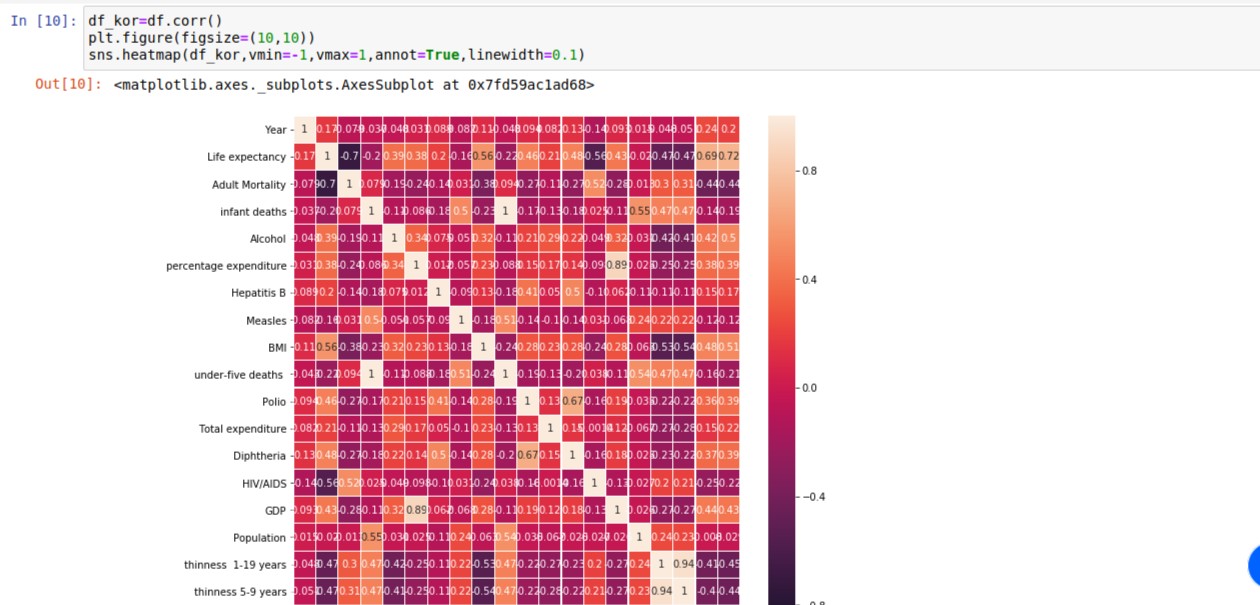


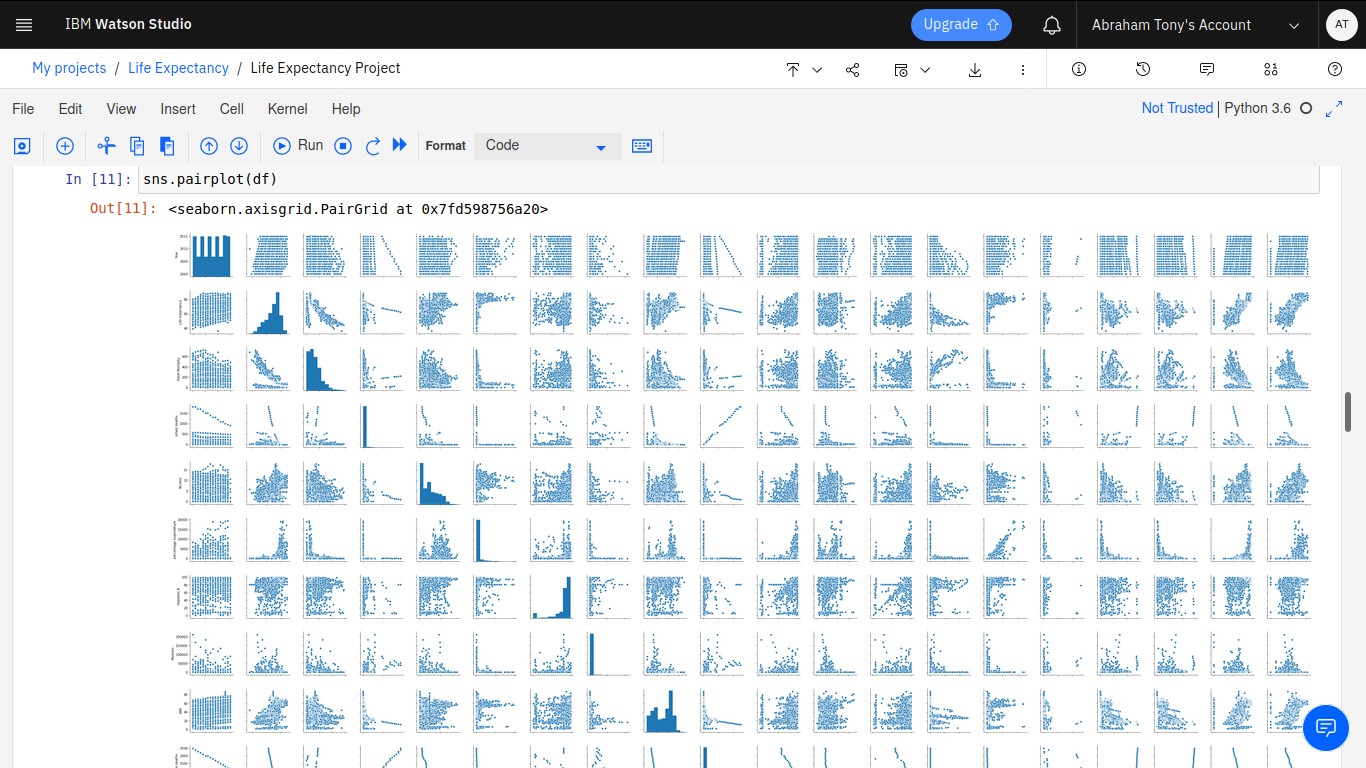
* 1. **Software Designing**

Python IDE, IBM Watson Studio, IBM Machine Learning Services, IBM Cloud, Node-Red App, Excel.

# EXPERIMENTAL INVESTIGATIONS

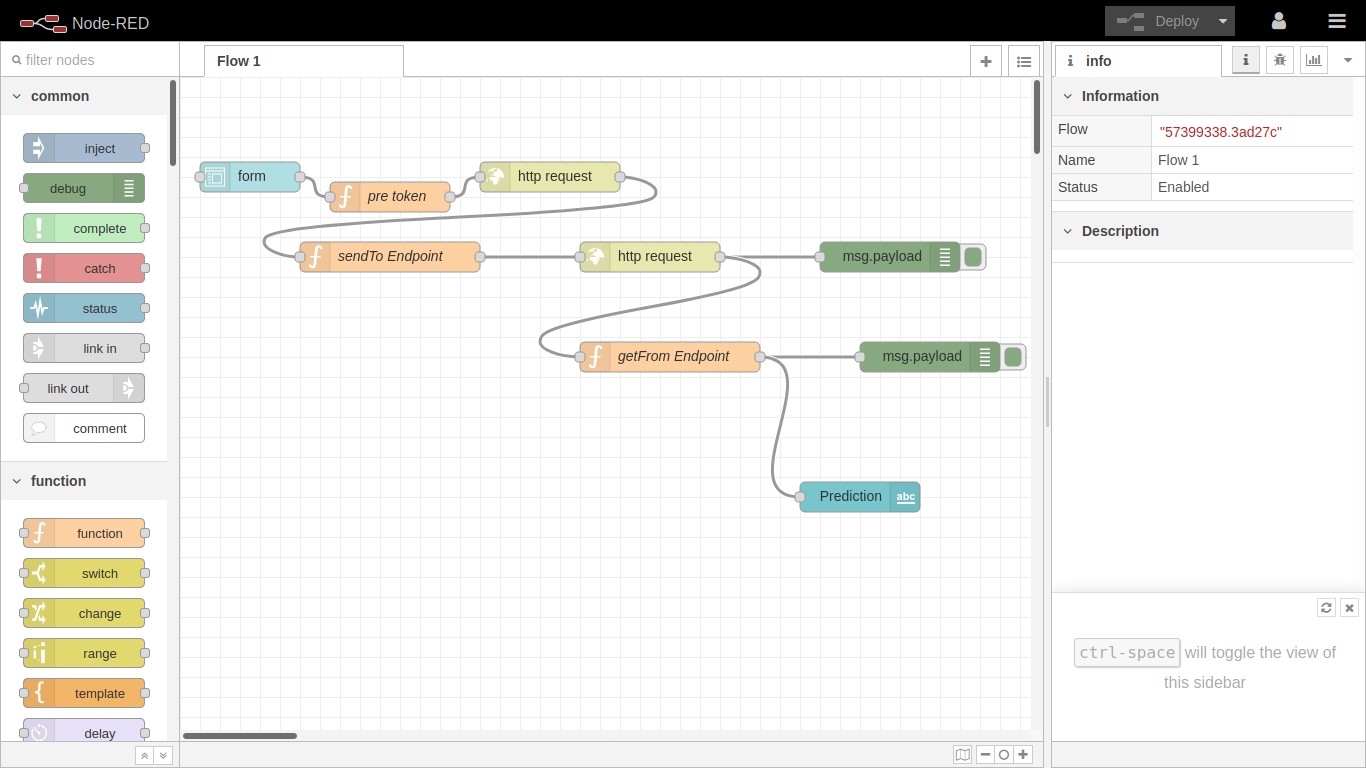
Data was collected from “https://[www.kaggle.com/kumarajarshi/life-](http://www.kaggle.com/kumarajarshi/life-) expectancy-who/data” and then pre-processed so that it is understood by the Machine Learning Algorithms Properly.

**Data Visualisation**



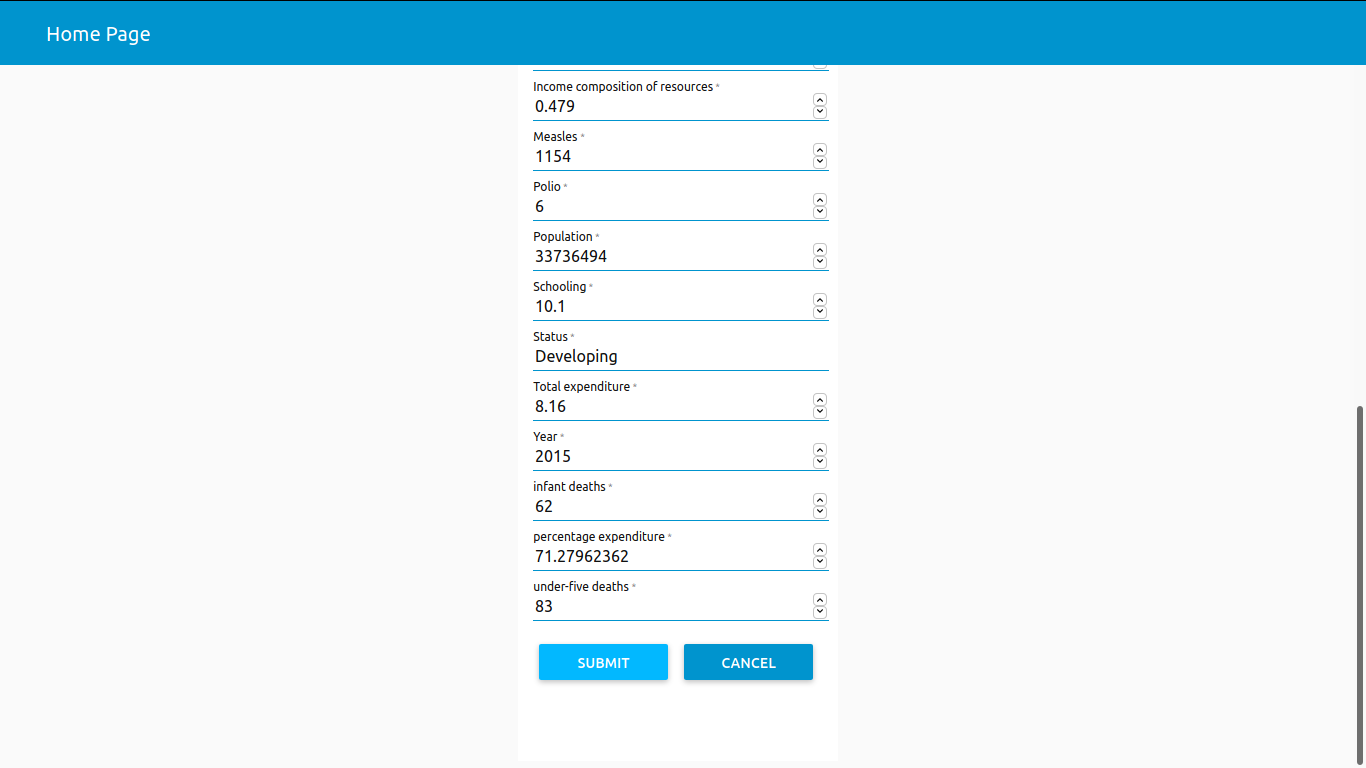
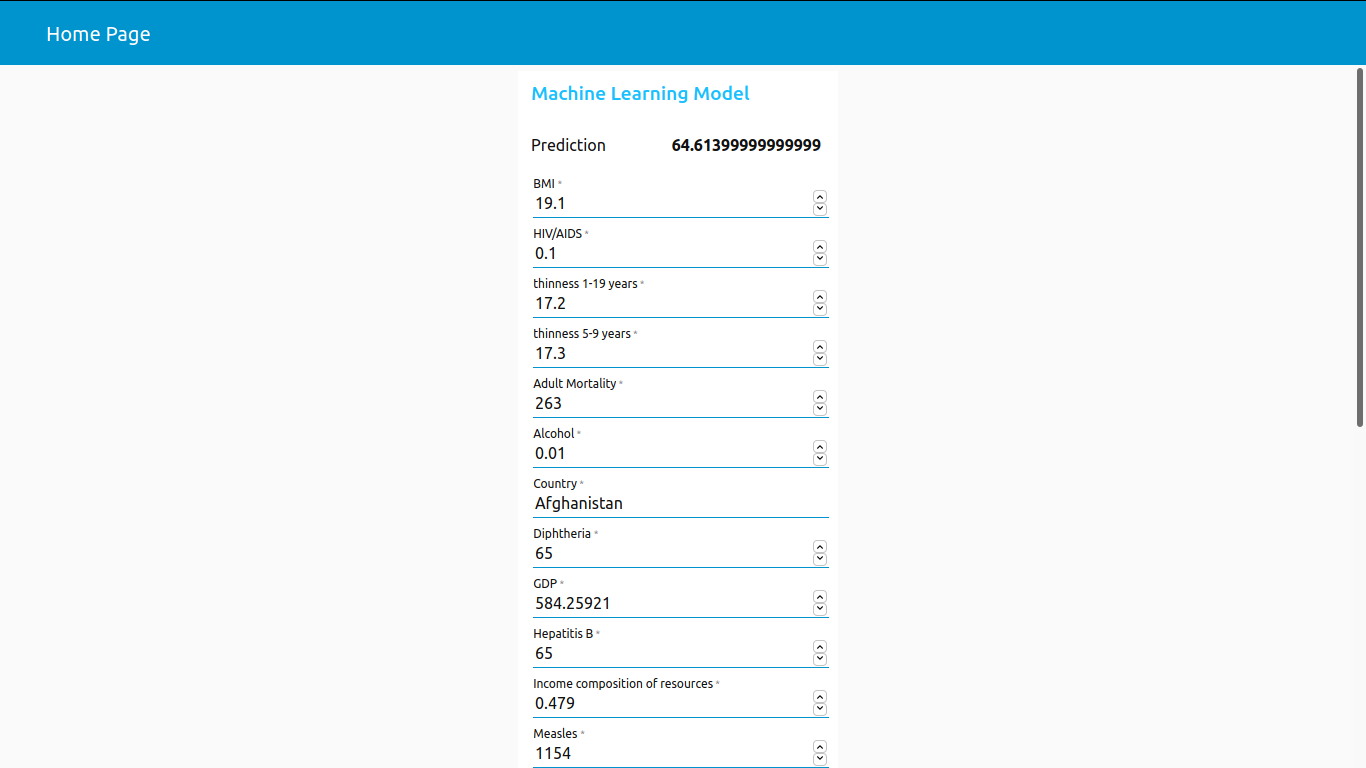
Then, different Regression Algorithms were applied and then accuracy is checked for each, so as to ﬁnd the best ﬁtted algorithm. Fine Tuning was done, in order to ﬁnd the best parameters so that we get the best possible accuracy.

# FLOWCHART



*Node Red Flow*

# RESULTS



1. **ADVANTAGES & DISADVANTAGES**

**Advantages:**

* Life Expectancy can be predicted depending on certain parameters with great accuracy.
* Beneﬁt the country’s growth.

# Disadvantages:

* Though, the accuracy of the model is very high. Still there is some chance that the does not give the exact Life Expectancy.
* Input should be in range only to predict accurate values.

# APPLICATIONS:

* To analyze country’s growth statistics in future years.
* To help government prepare life insurance policies for people. This will beneﬁt the people.
* To analyze all the factors and plan out measures to increase the life expectancy of the country.

# CONCLUSION

Thus, we have developed a model that will predict the life expectancy of a person living in a speciﬁc region. Various factors like Adult Mortality, Population, Under 5 Deaths, Thinness 1-5 Years, Alcohol, HIV, Hepatitis B, GDP,Percentage Expenditure and many more play an important role in the prediction.

# FUTURE SCOPE

Look at class within a particular country and see if these same factors are same in determining life expectancy for an individual. The accuracy of the model can be increased. This can be done by training more data. Also, the website can be added with many more features to improve the user experience.

# BIBILOGRAPHY

* https://[www.kaggle.com/kumarajarshi/life-expectancy-who/data](http://www.kaggle.com/kumarajarshi/life-expectancy-who/data)
* https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter- application/

# APPENDIX

**A. SOURCE CODE**

import pandas as pd import numpy as np import os

import matplotlib.pyplot as plt import seaborn as sns

pd.options.display.ﬂoat\_format='{:.5f}'.format import warnings

import math

#import libraries for pipelining

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import OneHotEncoder from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler from sklearn.compose import ColumnTransformer #import libraries for train and test

from sklearn.model\_selection import train\_test\_split #import ExtraTreesRegressor for model ﬁt and prediction from sklearn.ensemble import ExtraTreesRegressor #import libraries for accuracy and error calculation

from sklearn.metrics import mean\_squared\_error, r2\_score #import libraries for model building and deployment

from watson\_machine\_learning\_client import WatsonMachineLearningAPIClient

import types

import pandas as pd

from botocore.client import Conﬁg import ibm\_boto3

def iter (self): return 0

df = pd.read\_csv(body) df.head()

df.columns

df=df.rename(columns={'Life expectancy ':'Life expectancy','Measles ':'Measles','

BMI ':'BMI','Diphtheria ':'Diphtheria',' HIV/AIDS':'HIV/AIDS',' thinness 1-19 years':'thinness 1-19 years',' thinness 5-9 years':'thinness 5-9 years'}) df.isnull().sum()

df=df.ﬁllna(df.mean()) df.isnull().sum() df\_kor=df.corr() plt.ﬁgure(ﬁgsize=(10,10))

sns.heatmap(df\_kor,vmin=-1,vmax=1,annot=True,linewidth=0.1) sns.pairplot(df)

Y=df['Life expectancy'] X=df[df.columns.difference(['Life expectancy'])] df.select\_dtypes(include=['int64', 'ﬂoat64']).columns df.select\_dtypes(include=['object', 'bool']).columns categorical\_features = ['Country', 'Status'] categorical\_feature\_mask = X.dtypes==object

categorical\_features = X.columns[categorical\_feature\_mask].tolist() #DEFINE CATEGORICAL PIPELINE

categorical\_transformer = Pipeline(steps=[

('onehot', OneHotEncoder(handle\_unknown='ignore')),

])

numeric\_features = ['Year','Adult Mortality','infant deaths','Alcohol','percentage expenditure', 'Hepatitis B',

'Measles', 'BMI', 'under-ﬁve deaths ', 'Polio', 'Total expenditure','Diphtheria', 'HIV/AIDS', 'GDP', 'Population',

'thinness 1-19 years', 'thinness 5-9 years','Income composition of resources', 'Schooling']

numeric\_feature\_mask = X.dtypes!=object

numeric\_features = X.columns[numeric\_feature\_mask].tolist() #DEFINE NUMERIC PIPELINE

numeric\_transformer = Pipeline(steps=[ ('imputer', SimpleImputer(strategy='median')), ('scaler', StandardScaler()),

])

preprocessor = ColumnTransformer( transformers=[

('num', numeric\_transformer, numeric\_features), ('cat', categorical\_transformer, categorical\_features)

]

)

ExtraTreeRegressor = Pipeline([ ('preprocessor', preprocessor),

('ExtraTreeRegressor', ExtraTreesRegressor(n\_estimators=100, random\_state=0))

])

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(X,Y,test\_size=0.2)

reg = ExtraTreeRegressor.ﬁt(X\_train, Y\_train) test\_pred=reg.predict(X\_test) print(test\_pred)

print('Mean squared error: ',mean\_squared\_error(Y\_test, test\_pred)) print('R2 score: ',r2\_score(Y\_test, test\_pred)\*100)

wml\_credentials={

"apikey": " 7SIBVIcUkX6y\_-N-pOG4tuMjetVTAcHOXWFACKI0BN07", "instance\_id": " 1ad444cd-df01-4589-a1f3-84edf35cf647",

"url": "https://eu-gb.ml.cloud.ibm.com"

}

client = WatsonMachineLearningAPIClient(wml\_credentials) print(client.service\_instance.get\_url())

model\_props = {client.repository.ModelMetaNames.AUTHOR\_NAME:

"Videsh Loya",

client.repository.ModelMetaNames.AUTHOR\_EMAIL:

"vloya\_cse170557@mgit.ac.in",

client.repository.ModelMetaNames.NAME: "Life Expectancy"} #STORE THE MACHINE LEARNING MODEL

model\_artifact=client.repository.store\_model(ExtraTreeRegressor, meta\_props=model\_props)

#GET MODEL UID

model\_uid = client.repository.get\_model\_uid(model\_artifact) #DEPLOY THE MODEL

create\_deployment = client.deployments.create(model\_uid, name="LifeExpectancyPrediction")

scoring\_endpoint = client.deployments.get\_scoring\_url(create\_deployment) print(scoring\_endpoint)