SMARTBRIDGE
Let's Bridge the Gap

Project Report

Predicting Life Expectancy Using Machine Learning

Table of Contents

I. Introduction

A. 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. This problem statement provides a way to predict average life expectancy of people living in a country when various factors 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.

B. Purpose

The purpose of this project is to create a model and a corresponding website to take several attributes as input and output the life expectancy predicted by the model. The model is created based on data provided by the World Health Organization (WHO) to evaluate the life expectancy for different countries in years. The data offers a timeframe from 2000 to 2015.

II. Literature Survey

A. Existing Problem

We're in an unprecedented era where humans are living longer and longer. It's no secret though, that life expectancy varies widely across the globe. Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that affect of immunization and human development index was not taken into account in the past. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries.

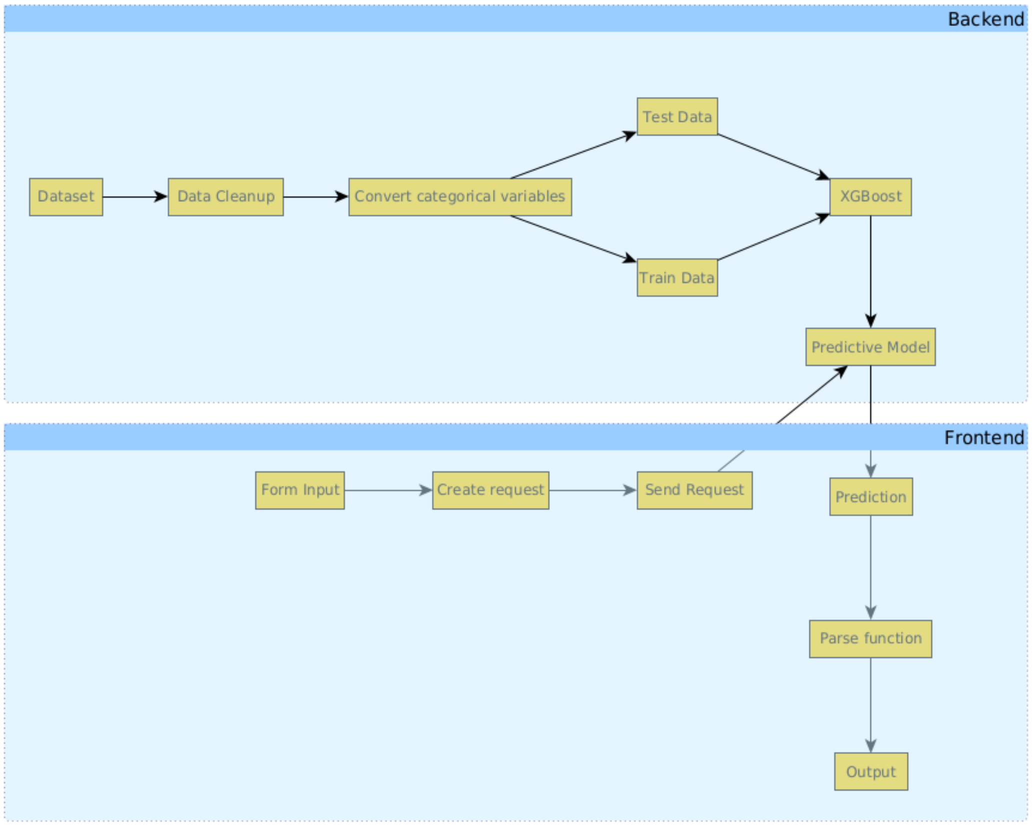
B. Proposed Solution

The proposed solution is to formulate a regression model using while considering data from a period of 2000 to 2015 for all the countries. Important immunization like Hepatitis B, Polio and Diphtheria will also be considered. In a nutshell, this study will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. Since the observations in this dataset are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

There are two possible approaches to this problem. One is to train a model using Python and IBM Watson's Machine Learning service. The second is to use IBM's AutoAI service to train the model. We will use Python to train an Xgboost model.

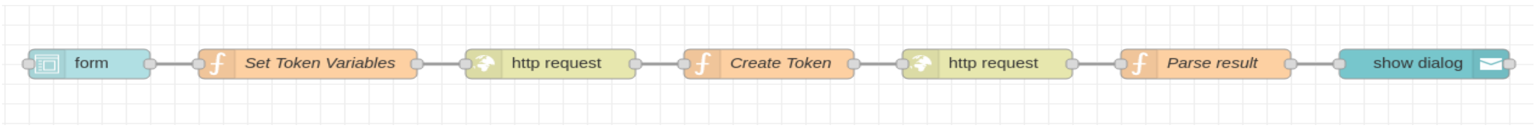
III. Theoretical Analysis

A. Block Diagram

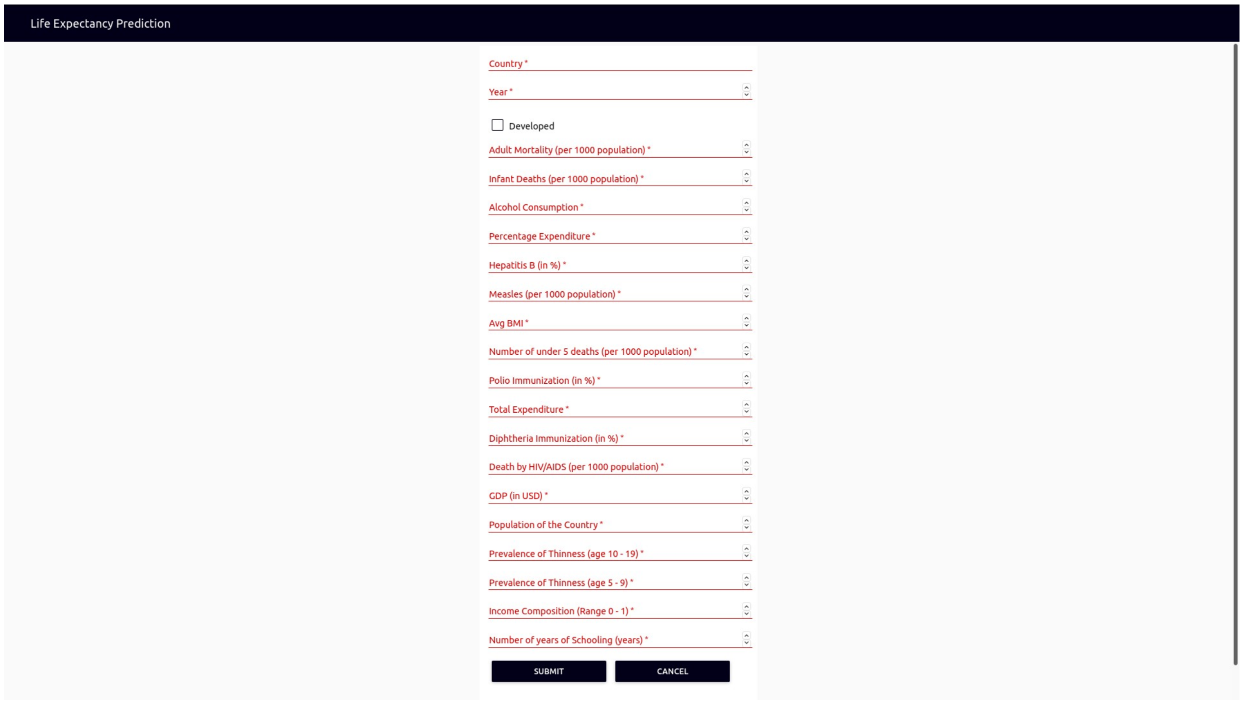


B. Hardware/Software Designing

The project was developed and run on IBM Cloud. The machine learning model was trained using IBM Watson Studio. The user interface was built using a Node-RED app.



Node Red Flow



Life Expectancy Prediction

Country *

Year *

☐ Developed

Adult Mortality (per 1000 population) *

Infant Deaths (per 1000 population) *

Alcohol Consumption *

Percentage Expenditure *

Hepatitis B (in %) *

Measles (per 1000 population) *

Avg BMI *

Number of under 5 deaths (per 1000 population) *

Polio Immunization (in %) *

Total Expenditure *

Diphtheria Immunization (in %) *

Death by HIV/AIDS (per 1000 population) *

GDP (in USD) *

Population of the Country *

Prevalence of Thinness (age 10 - 19) *

Prevalence of Thinness (age 5 - 9) *

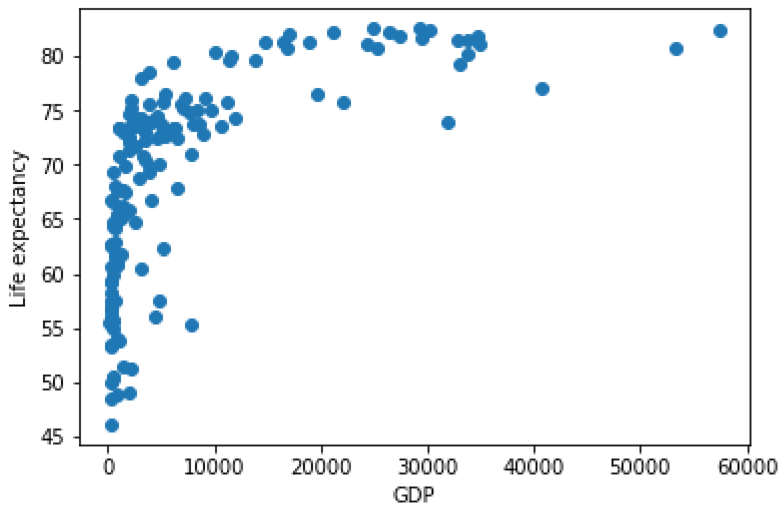
Income Composition (Range 0 - 1) *

Number of years of Schooling (years) *

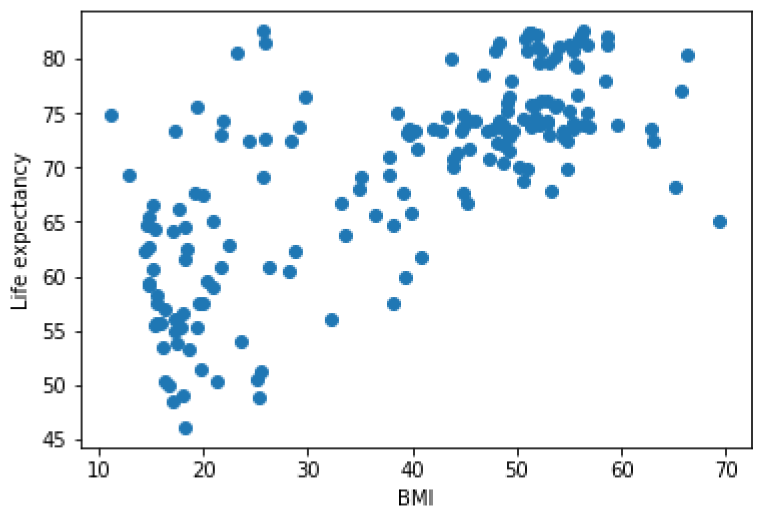
SUBMIT CANCEL

User Interface

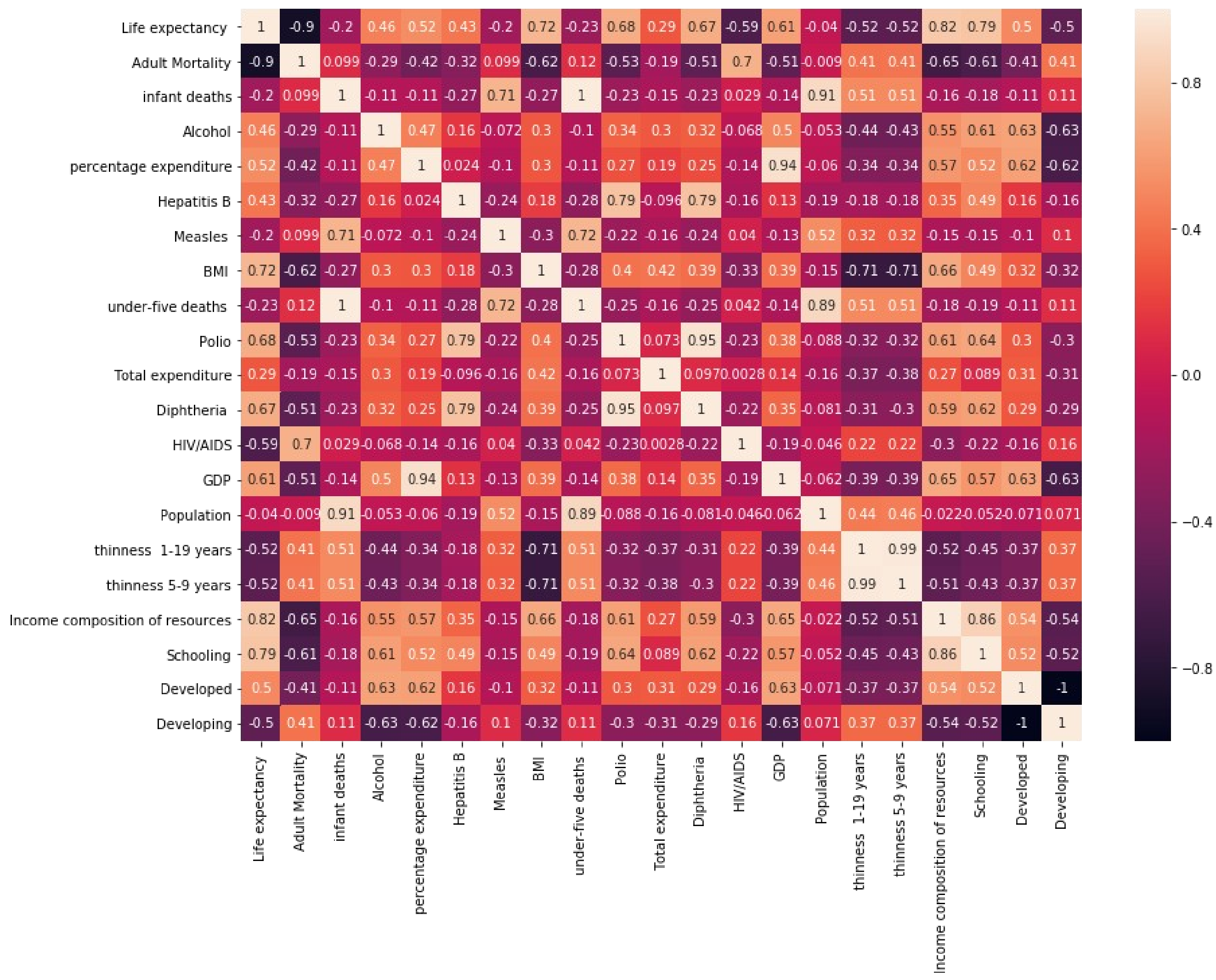
IV. Experimental Investigations



Life Expectancy vs GDP

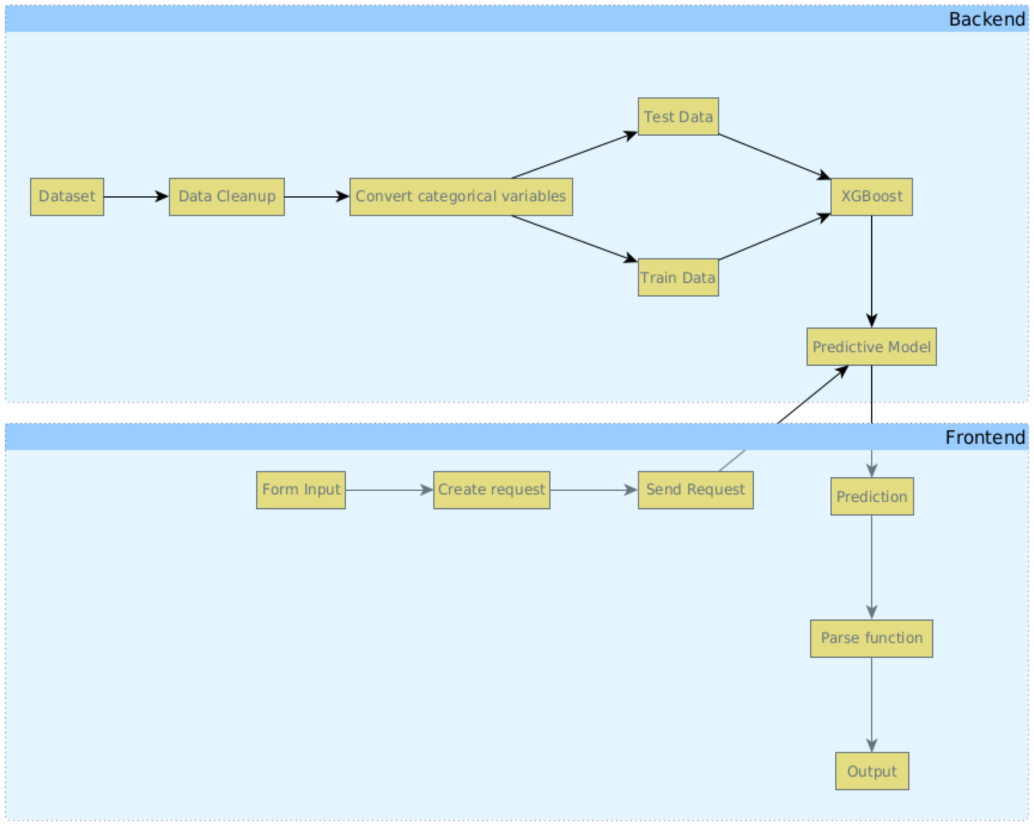


Life Expectancy vs BMI

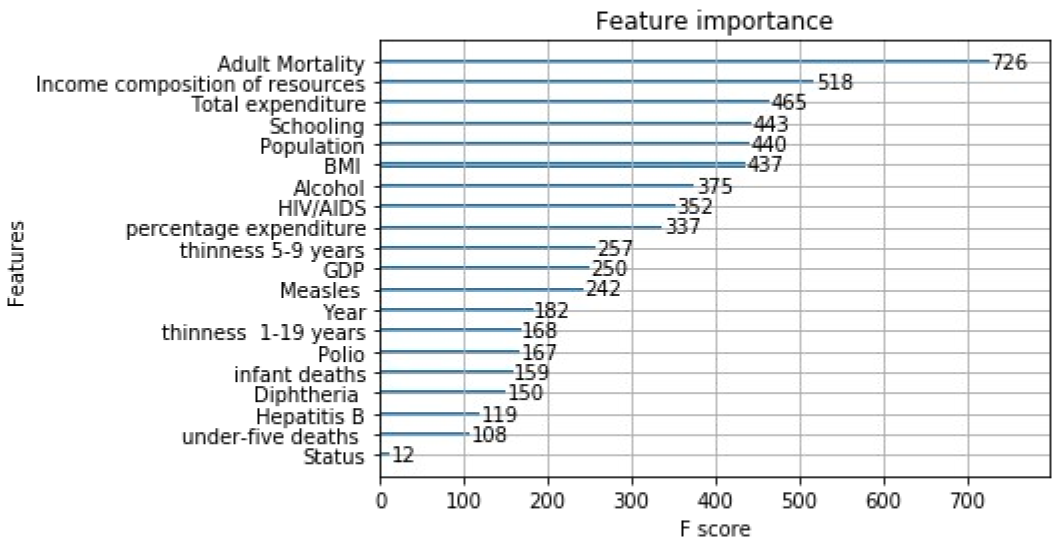


Feature Importance Graph

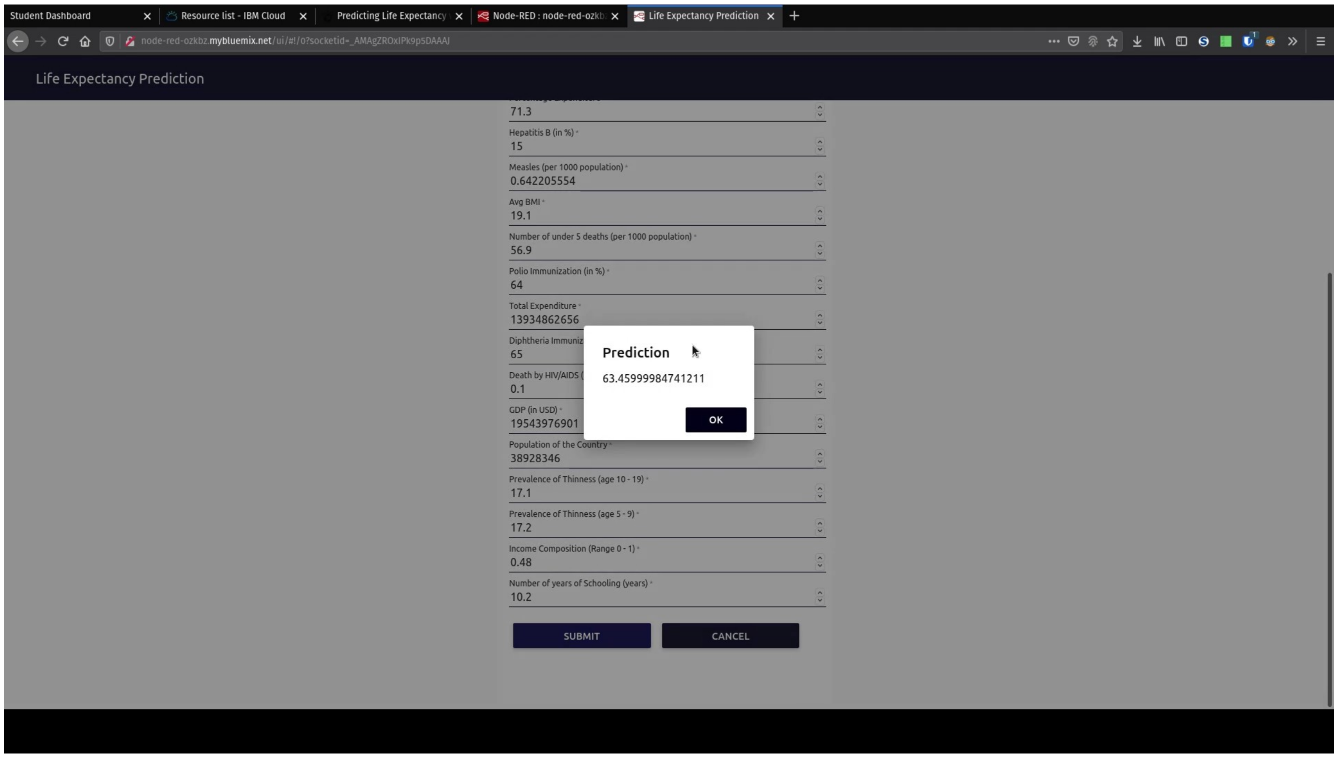
V. Flowchart


Backend
Dataset
Data Cleanup
Convert categorical variables
Test Data
Train Data
XGBoost
Predictive Model
Frontend
Form Input
Create request
Send Request
Prediction
Parse function
Output

VI. Result



Feature Importance Graph


Student Dashboard
Resource list - IBM Cloud
Predicting Life Expectancy
Node-RED : node-red-ozkb
Life Expectancy Prediction
node-red-ozkbz.mybluemix.net/ui/#!/0?socketid=_AMAgZROxIPk9p5DAAAJ
SUBMIT
CANCEL
OK

Life Expectancy Prediction

71.3

Hepatitis B (in %) *
15

Measles (per 1000 population) *
0.642205554

Avg BMI *
19.1

Number of under 5 deaths (per 1000 population) *
56.9

Polio Immunization (in %) *
64

Total Expenditure *
13934862656

Diphtheria Immuniz
65

Death by HIV/AIDS (
0.1

GDP (in USD) *
19543976901

Population of the Country *
38928346

Prevalence of Thinness (age 10 - 19) *
17.1

Prevalence of Thinness (age 5 - 9) *
17.2

Income Composition (Range 0 - 1) *
0.48

Number of years of Schooling (years) *
10.2

Prediction

63.45999984741211

Sample Prediction

The XGBoost model had an RMSE score of 2.15626

VII. Advantages & Disadvantages

Advantages

- XGBoost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting
- XGBoost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model.
- XGBoost has an in-built capability to handle missing values. When XGBoost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.

Disadvantages

- Black box nature. If you need effect sizes, XGBoost won't give them to you.

VIII. Applications

Predicting average life expectancy can help countries look at the various factors that contribute to the life expectancy of their population. They can then use that data to focus their budget into useful programs that target the factors that affect life expectancy the most.

IX. Conclusion

There is no single predictor of life expectancy. There are things that contribute to life expectancy that is not in this dataset. But, based on the data used in this project, multiple factors greatly affects the life expectancy. The factors that had the most impact on life expectancy are number of deaths from government expenditure in healthcare, number of years in school, and the body mass index of the population.

Using these factors, one can predict the life expectancy of a population using health, social, and economic variables. It would be a stretch to use this information to predict the life expectancy of an individual because there are many more life variables involved than the variables presented in this project. This project and data analysis is most useful at predicting life expectancy at the population level.

X. Future Scope

The dataset used only contains data until 2015. More recent data could be used to update the dataset and get more relevant findings. The dataset also contains very specific factors that may not be the only contributing factors to life expectancy. Unfortunately it is extremely difficult to get more data points from a significant enough number of countries around the world.

XI. Bibliography

IBM Cloud

Dashboard

https://cloud.ibm.com/

IBM Watson Studio

https://dataplatform.cloud.ibm.com/

Node RED

node-red-ozkbz.mybluemix.net/

Kaggle

https://www.kaggle.com/

Github

https://github.com/

XII. Appendix

A. Source Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sb
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import make_scorer
import xgboost as xgb
import json
import os

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_ = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='',
    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_57e243b527ff4d188ab18a3720ab635f.get_object(Bucket='predictlifeexpecta
ncywithpython-donotdelete-pr-m0wpobl1fe1rec',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_data_1 = pd.read_csv(body)
df_data_1.head()
df = df_data_1

df.replace(['Developing', 'Developed'], [0, 1], inplace=True)

lf = df['Life expectancy ']

df.drop(labels=["Life expectancy ", "Country"], axis = 1, inplace=True)
df.insert(0, "Life Expectancy", lf)
df.dropna(inplace=True)
df.head()

X = list(df)

del(X[0])
for i in X:
    print(i)

X_train, X_test, Y_train, Y_test = train_test_split(df[X], df['Life
Expectancy'], test_size=0.1, random_state=10)

X_train.head()

def plot_importance(model):

    k = list(zip(X, model.feature_importances_))
    k.sort(key=lambda tup: tup[1])

    labels, vals = zip(*k)

    plt.barh(np.arange(len(X)), vals, align='center')
    plt.yticks(np.arange(len(X)), labels)



xgboost_tree = xgb.XGBRegressor(

    n_estimators = 2000,
    max_depth = 2,
    tree_method = 'exact',
    reg_alpha = 0.05,
    silent = 0,
    random_state = 1023
)

xgboost_tree.fit(X_train[X], Y_train,
                 eval_set = [(X_train[X], Y_train), (X_test[X], Y_test)],
                 early_stopping_rounds = 300
                )

xgb.plot_importance(xgboost_tree)

# create client to access our WML service

from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_credentials = {
  "apikey": "3UsOqjaTqWKw05X8ZYYGPZYSo9qh3jHlmLEXmo_Zw8Gc",
  "iam_apikey_description": "Auto-generated for key 111e1079-444f-4800-9233-
644efb40ee5a",
  "iam_apikey_name": "wdp-writer",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
  "iam_serviceid_crn":
"crn:v1:bluemix:public:iam-identity::a/d92ab8e8868a4fe1b67343b310afa26f::serv
iceid:ServiceId-8691377a-6320-4351-8c82-9a22564f1af8",
  "instance_id": "12048521-9148-44fa-9f65-4fecbb255880",
  "url": "https://us-south.ml.cloud.ibm.com"
}

client = WatsonMachineLearningAPIClient(wml_credentials)
print(client.version)

client.repository.list_models()

client.deployments.list()

meta_props={client.repository.ModelMetaNames.NAME: "XGBoost model to predict
life expectancy"}

published_model = client.repository.store_model(model=xgboost_tree,
meta_props={client.repository.ModelMetaNames.NAME: "XGBoost model to predict
life expectancy"})

client.repository.list_models()


# get UID of our just stored model
model_uid = client.repository.get_model_uid(published_model)
print("Model id: {}".format(model_uid))

# create deployment

created_deployment = client.deployments.create(model_uid,
name="life_expectancy_model_xgb")

# new list of deployments
client.deployments.list()

# get UID of our new deployment
```

```
deployment_uid = client.deployments.get_uid(created_deployment)
print("Deployment id: {}".format(deployment_uid))
print(created_deployment)

scoring_endpoint = client.deployments.get_scoring_url(created_deployment)

print(scoring_endpoint)



scoring_payload = {

    'fields': ['Year', 'Status', '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'],
    'values': [[2007, 0, 26, 2, 3.79, 126.698109, 98, 12, 43.9, 2,99, 5.6,
95, 0.1, 1634.81431, 259167, 2.2, 2.4, 0.661, 13]]
}

predictions = client.deployments.score(scoring_endpoint, scoring_payload)

print('prediction',json.dumps(predictions, indent=2))
```