PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

Submitted by:

Ojaswita Negi

ojaswitanegi@gmail.com

Application ID **SPS_APL_20200003286**

CONTENTS

Introduction

Overview

Life expectancy is a statistical measure of the average (see below) time an organism is expected to live, based on the year of its birth, its current age, and other demographic factors including gender. In this project, a regression machine learning model has been used for the prediction of life expectancy of an individual based on the given dataset compiled and provided by WHO. The User interface is based on the Node-Red application service provided by IBM cloud services and the back end has been developed in Watson Studio.

Problem Statement- Predicting Life Expectancy using Machine Learning.

Purpose

The purpose of this document is to give a detailed description of the project Predicting Life Expectancy using Machine Learning.

Literature Survey

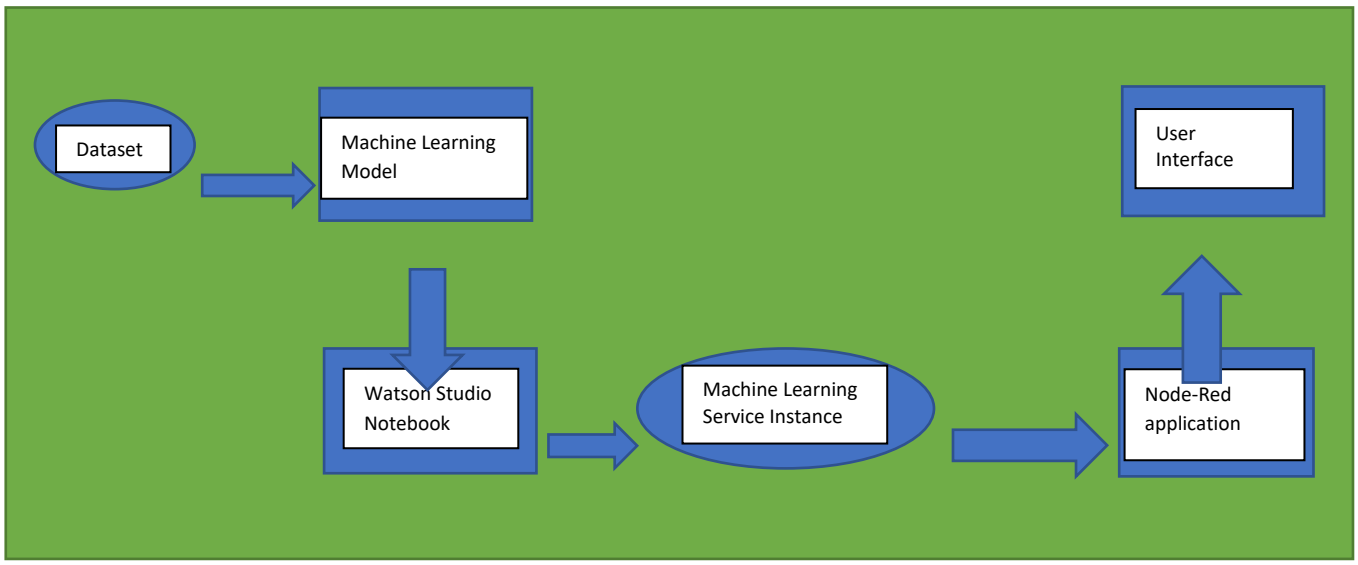
Existing Solution

Due to the rampant technological advancements and better healthcare facilities, the life expectancy of an individual has increased considerably. Understanding the life expectancy analysis as well as its repercussions on other aspects such as the population of a country are important measures to comprehend the various resource allocation on an administrative level for a better life. Many health monitors have been put to use to monitor the well-being of an individual that have a very narrow scope.

Proposed Solution

This project takes into consideration the dataset provided by WHO that related the life expectancy of an individual based on various factors such as adult mortality, infant death, 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 as well as schooling. Based on these features, our machine learning model will predict a value of life expectancy. There were a lot of inconsistencies within the dataset. The mossing values and NaN values have been replaced with the mean of the features. This processed data has been divided into training and test sets in the ratio 7:3.

Block Diagram



Project Requirements

Functional Requirements

The user shall be able to provide the values of the following features of the machine learning model in the Node-Red application.

1. Adult Mortality
2. Infant Death,
3. Alcohol.
4. Percentage expenditure

5. Hepatitis B

6. Measles

7. BMI

8. Under-five deaths

9. Polio

10. Total Expenditure

11. Diphtheria

12. HIV/AIDS

13. GDP

14. Population

15. thinness 1-19 years

16. thinness 5-9 years

17. Income composition of resources

18. Schooling

Following the insertion of the above data in the node-red application, a predicted value shall be displayed on the node-red application immediately after submission of these values.

Technical Requirements

- Software Requirements

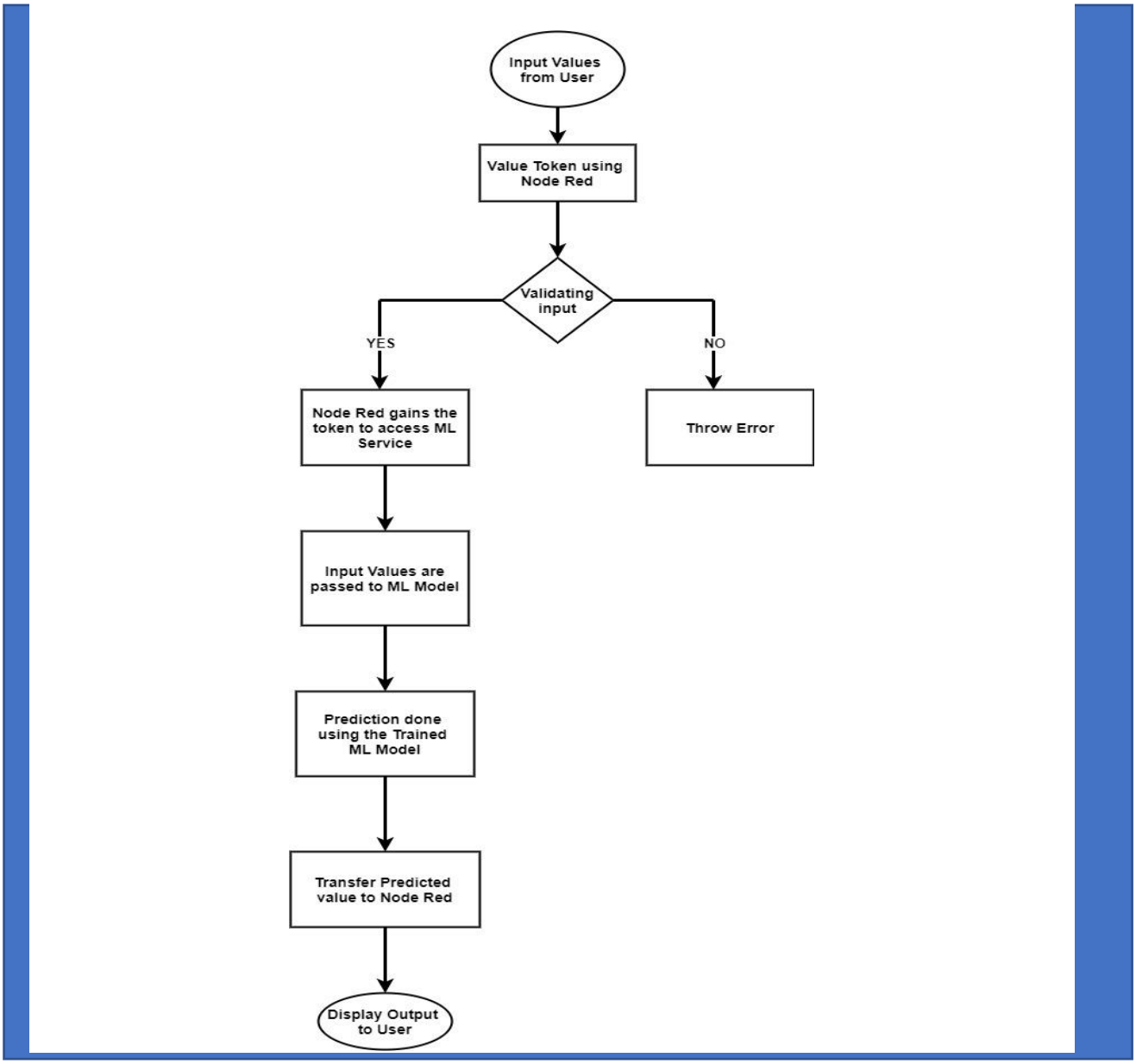
Th development software s required are the Node-Red application development service on the IBM cloud Service, the instance of Machine Learning that will be used to generate access for the machine learning model built, the Watson studio

application to create a blank notebook and build the machine learning model in it.

- Hardware Requirements

Any working laptop/PC with minimum 2.2Ghz processor and at least 8GB of memory with an Internet connection.

Flow Chart



Designing the Model

: Open Watson studio

=> New Project

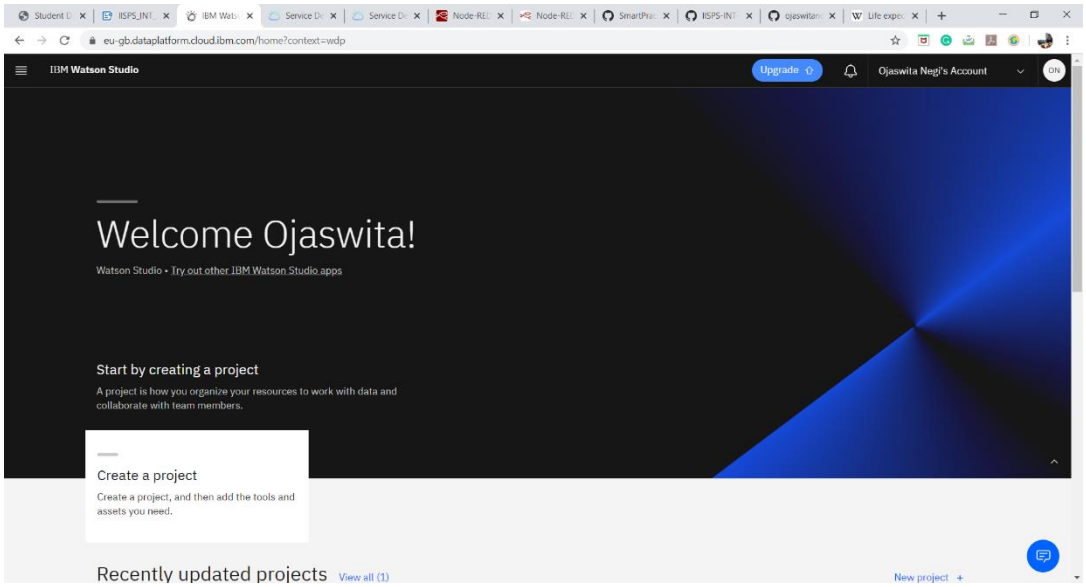
=> Create an empty Project

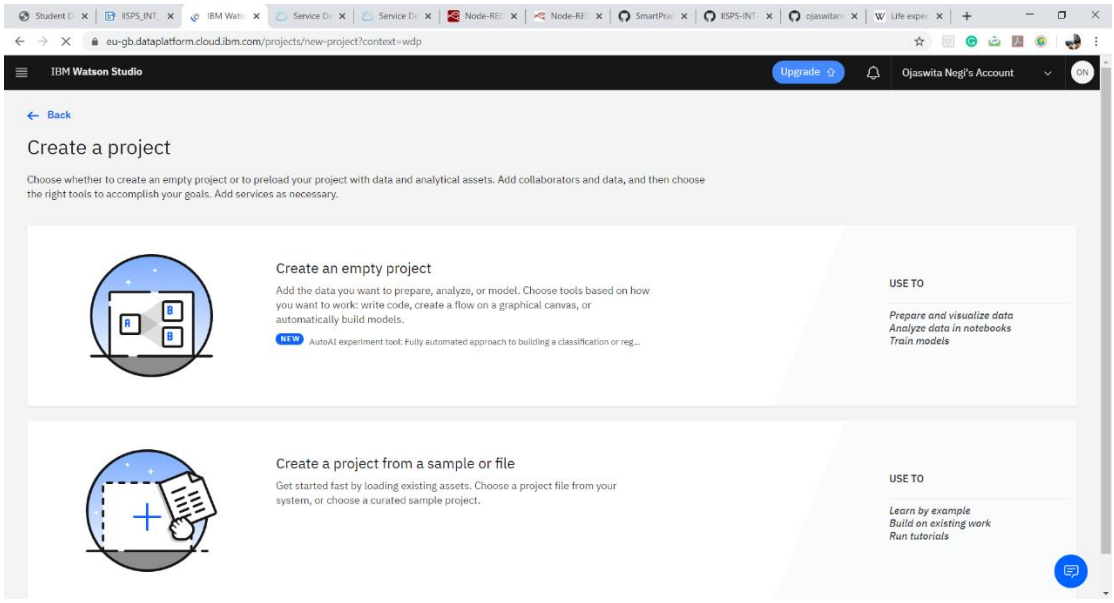
=> Give project name

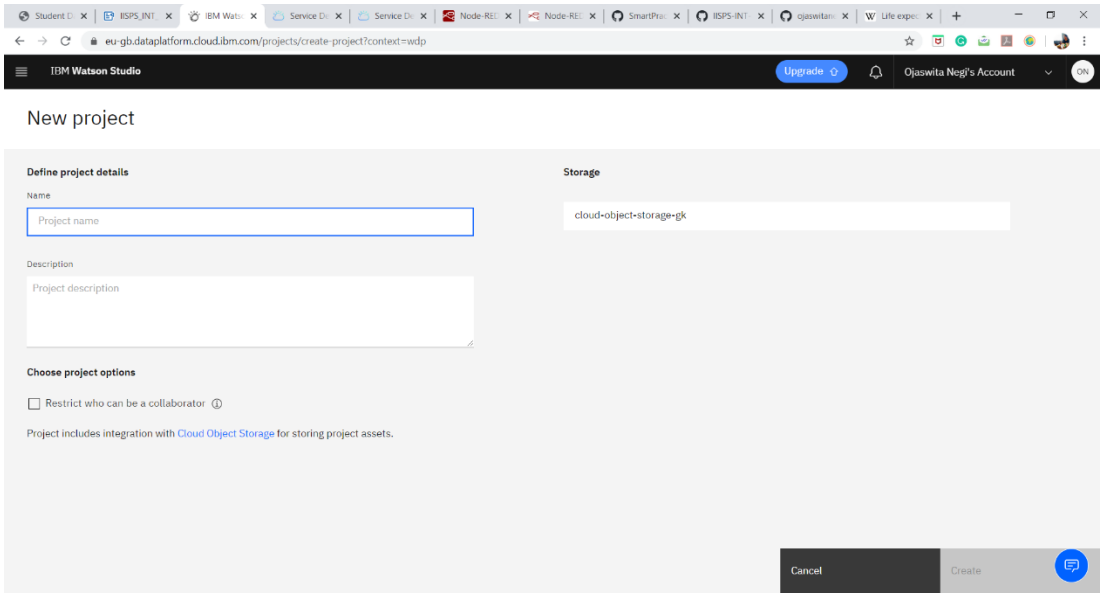
=>Click Create

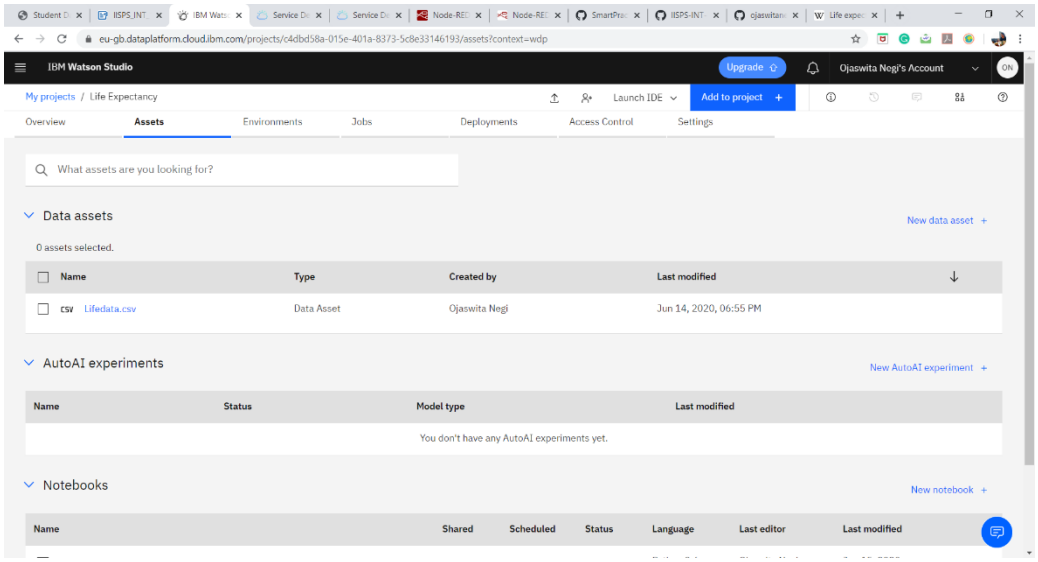
=>Add to Project

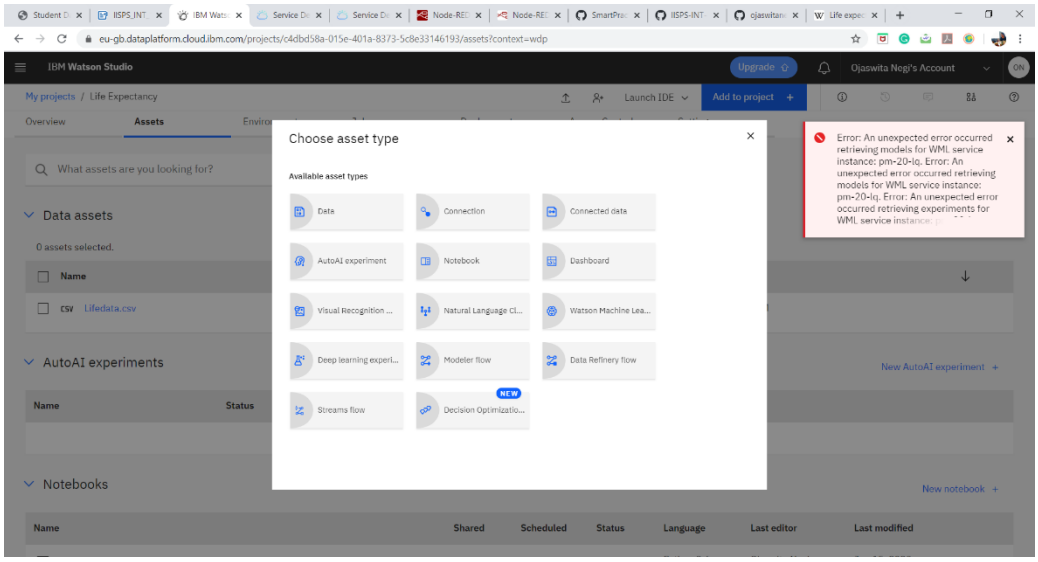
=> Notebook



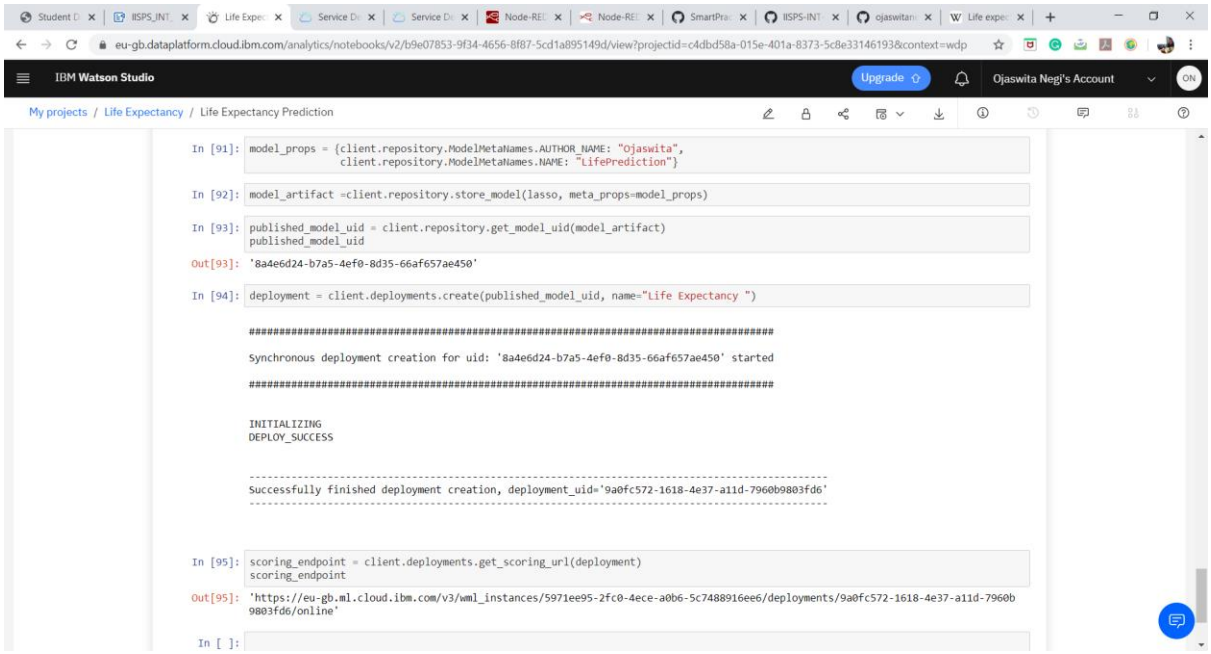

IBM Watson Studio
Upgrade
Ojaswita Negi's Account
Back
Create a project
Choose whether to create an empty project or to preload your project with data and analytical assets. Add collaborators and data, and then choose the right tools to accomplish your goals. Add services as necessary.
Create an empty project
Add the data you want to prepare, analyze, or model. Choose tools based on how you want to work: write code, create a flow on a graphical canvas, or automatically build models.
NEW
AutoAI experiment tool: Fully automated approach to building a classification or reg...
USE TO
Prepare and visualize data
Analyze data in notebooks
Train models
Create a project from a sample or file
Get started fast by loading existing assets. Choose a project file from your system, or choose a curated sample project.
USE TO
Learn by example
Build on existing work
Run tutorials


IBM Watson Studio
Upgrade
Ojaswita Negi's Account
New project
Define project details
Name
Project name
Description
Project description
Choose project options
Restrict who can be a collaborator
Project includes integration with Cloud Object Storage for storing project assets.
Storage
cloud-object-storage-gk
Cancel
Create

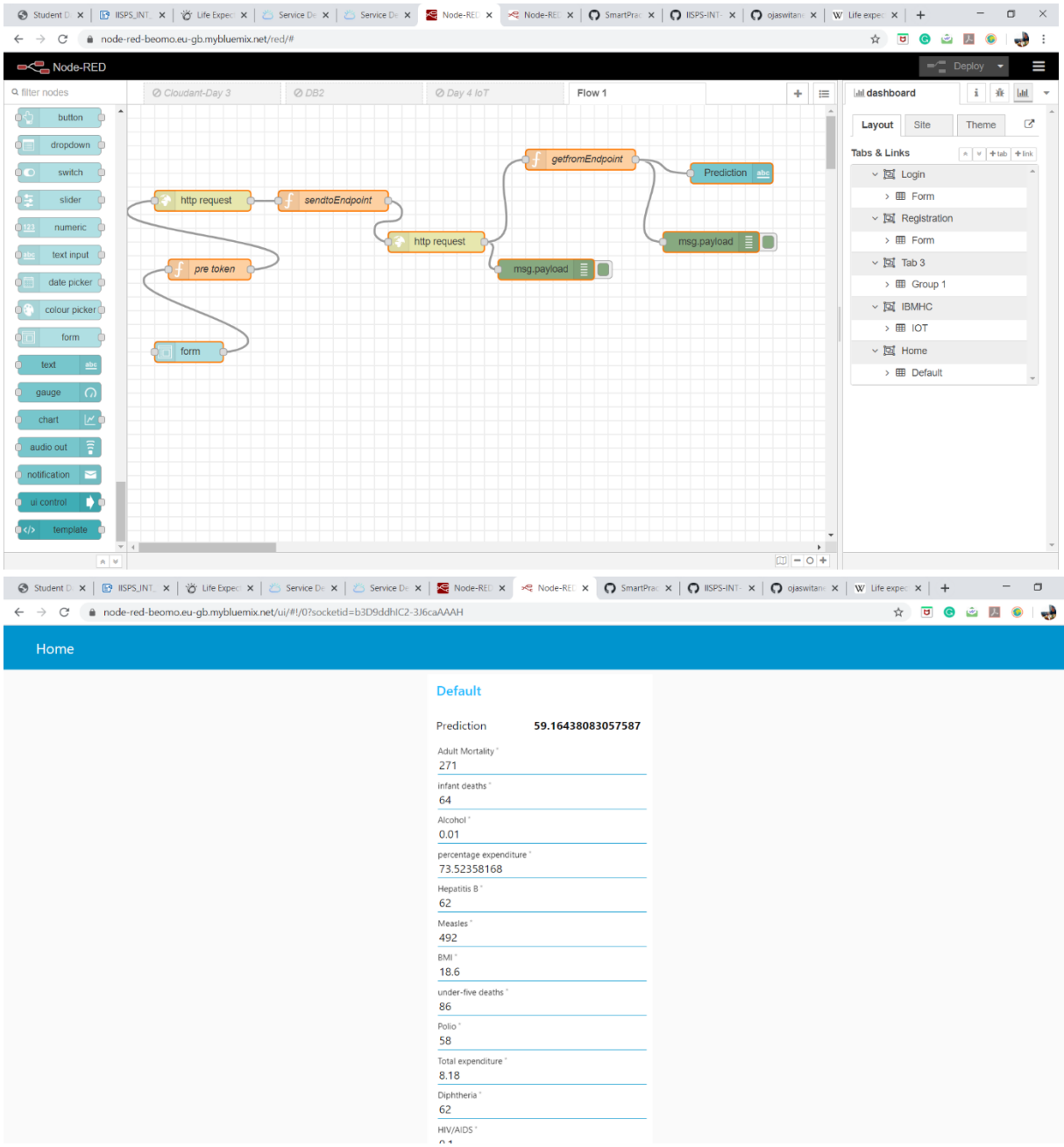

IBM Watson Studio
Upgrade
Ojaswita Negi's Account
My projects / Life Expectancy
Launch IDE
Add to project
Overview
Assets
Environments
Jobs
Deployments
Access Control
Settings
What assets are you looking for?
Data assets
New data asset
0 assets selected.
Name
Type
Created by
Last modified
csv Lifedata.csv
Data Asset
Ojaswita Negi
Jun 14, 2020, 06:55 PM
AutoAI experiments
New AutoAI experiment
Name
Status
Model type
Last modified
You don't have any AutoAI experiments yet.
Notebooks
New notebook
Name
Shared
Scheduled
Status
Language
Last editor
Last modified



Generate Scoring point



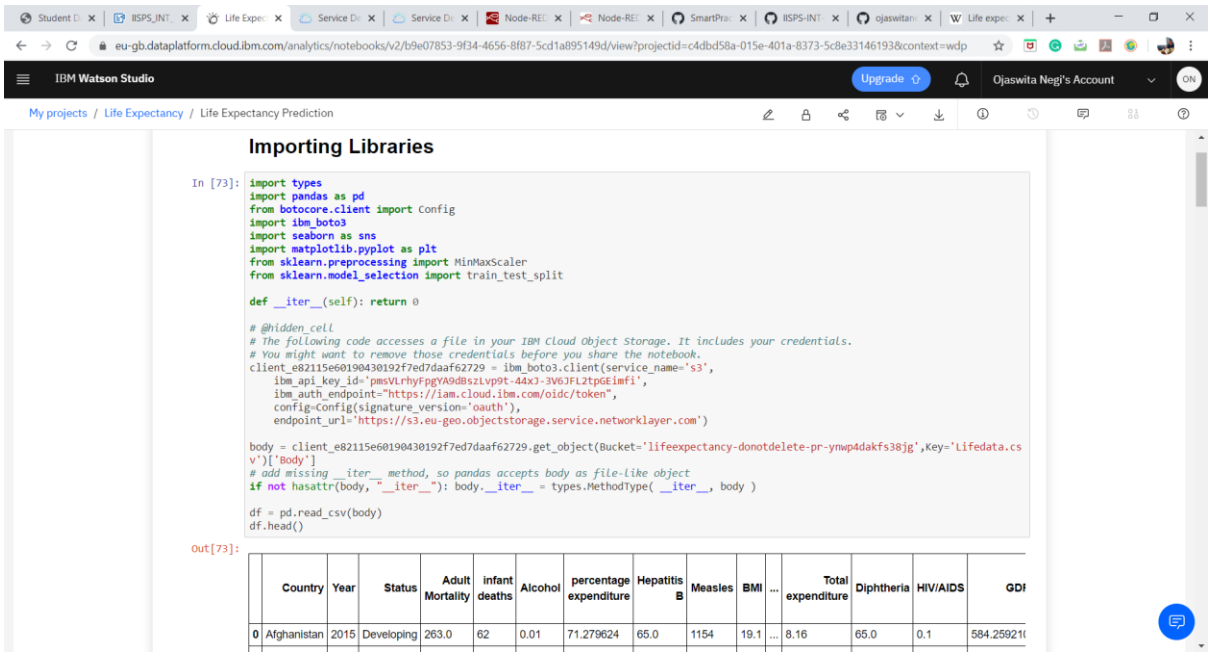
Node-Red Flow



Resources

- https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as- web-service
- https://www.ibm.com/watson/products-services

Code

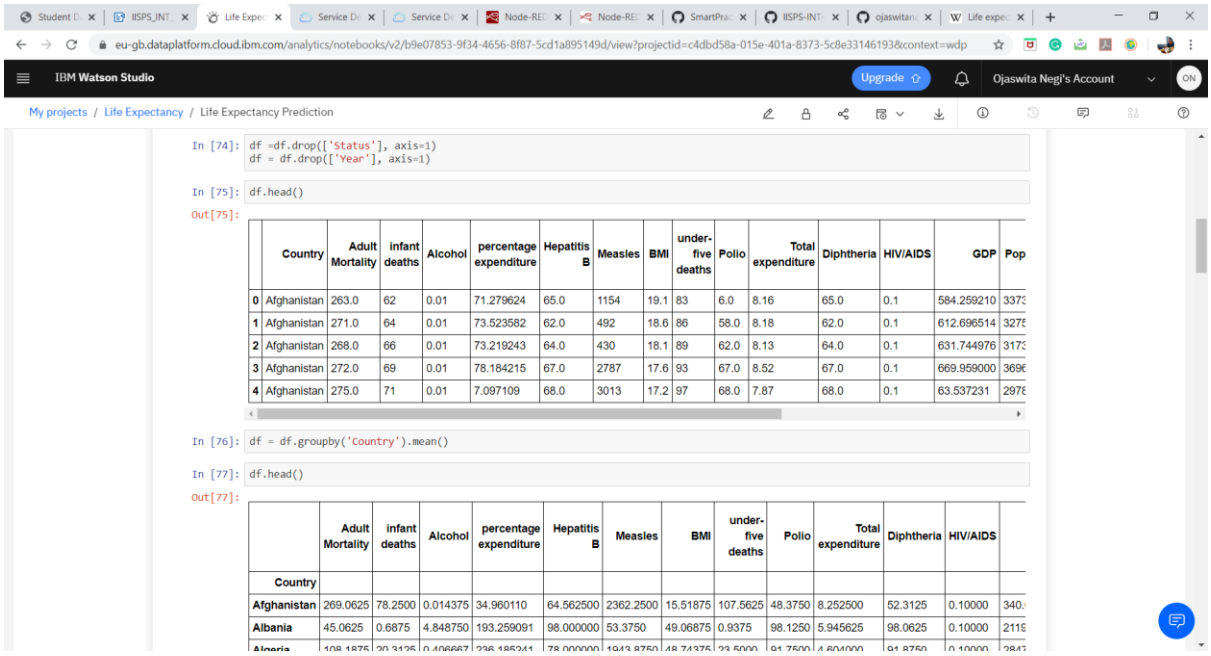


Importing Libraries

In [73]:
```
import types
import pandas as pd
from botocore.client import Config
import ibm_boto3
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

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_e82115e60190430192f7ed7daaf62729 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='pmsVLrhyFpgYA9dBszLvp9t-44xJ-3V6JFL2tpGEimfi',
    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_e82115e60190430192f7ed7daaf62729.get_object(Bucket='lifeexpectancy-donotdelete-pr-ynwp4dakfs38jg',Key='Lifedata.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()
```

Out[73]:

	Country	Year	Status	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	...	Total expenditure	Diphtheria	HIV/AIDS	GDP
0	Afghanistan	2015	Developing	263.0	62	0.01	71.279624	65.0	1154	19.1	...	8.16	65.0	0.1	584.25921(



In [74]:
```
df =df.drop(['Status'], axis=1)
df = df.drop(['Year'], axis=1)
```

In [75]:
```
df.head()
```

Out[75]:

	Country	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Pop
0	Afghanistan	263.0	62	0.01	71.279624	65.0	1154	19.1	83	6.0	8.16	65.0	0.1	584.259210	3373
1	Afghanistan	271.0	64	0.01	73.523582	62.0	492	18.6	86	58.0	8.18	62.0	0.1	612.696514	3275
2	Afghanistan	268.0	66	0.01	73.219243	64.0	430	18.1	89	62.0	8.13	64.0	0.1	631.744976	3173
3	Afghanistan	272.0	69	0.01	78.184215	67.0	2787	17.6	93	67.0	8.52	67.0	0.1	669.959000	3696
4	Afghanistan	275.0	71	0.01	7.097109	68.0	3013	17.2	97	68.0	7.87	68.0	0.1	63.537231	2978

In [76]:
```
df = df.groupby('Country').mean()
```

In [77]:
```
df.head()
```

Out[77]:

	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	
Country													
Afghanistan	269.0625	78.2500	0.014375	34.960110	64.562500	2362.2500	15.51875	107.5625	48.3750	8.252500	52.3125	0.10000	340.
Albania	45.0625	0.6875	4.848750	193.259091	98.000000	53.3750	49.06875	0.9375	98.1250	5.945625	98.0625	0.10000	2119
Algeria	108.1875	20.3125	0.406667	236.185241	78.000000	1943.8750	48.74375	23.5000	91.7500	4.604000	91.8750	0.10000	2847

Data Visualization

In [78]:
```
plt.figure(figsize = (15, 15))
sns.heatmap(df.corr(), annot = True)
```

Out[78]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4194383e48>

eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/b9e07853-9f34-4656-8f87-5cd1a895149d/view?projectid=c4dbd58a-015e-401a-8373-5c8e33146193&context=wdp

IBM Watson Studio | Upgrade | Ojaswita Negi's Account

My projects / Life Expectancy / Life Expectancy Prediction

In [79]:
```
Y = df['Life expectancy ']
X = df.drop('Life expectancy ', axis = 1)
```

In [80]:
```
X.isnull().sum()
```

Out[80]:
```
Adult Mortality                  10
infant deaths                     0
Alcohol                           2
percentage expenditure            0
Hepatitis B                       9
Measles                           0
 BMI                              4
under-five deaths                 0
Polio                             0
Total expenditure                 2
Diphtheria                        0
 HIV/AIDS                         0
GDP                              30
Population                       48
 thinness  1-19 years             4
 thinness 5-9 years               4
Income composition of resources   17
Schooling                         13
dtype: int64
```

Replacing NaN Values

In [81]:
```
X.fillna(value = X.mean(), inplace = True)
Y.fillna(value = Y.mean(), inplace = True)
```

Rescaling

In [82]:
```
#scaler = MinMaxScaler()
#X = scaler.fit_transform(X)
```

Splitting Data

In [83]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,train_size = 0.7, test_size = 0.3)
```

Building the Model

In [84]:
```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import Lasso

#fitting the model
lasso=Lasso(alpha=0,max_iter=10)
lasso.fit(X_train,Y_train)
lasso.predict(X_test)
lasso.score(X_train,Y_train)
```

Out[84]:
```
0.9234931584917856
```

In [85]:
```
lasso.score(X_test,Y_test)
```

Out[85]:
```
0.9059674200168311
```

In [96]:
```
lasso.predict([[ 271.0, 64, 0.01, 73.52358168,
       62.0, 492, 18.6, 86, 58.0, 8.18, 62.0, 0.1, 612.696514, 327582.0,
       17.5, 17.5, 0.4760000000000003, 10.0]])
```

Out[96]:
```
array([59.16438083])
```

Accuracy of Model

In [86]:
```
rsq
print(prediction)
```

```
[69.41537253 64.02650173 69.81204545 58.23550322 81.14262556 72.30126279
 64.1048612  70.29405071 52.32361168 64.59888789 67.36307947 70.76176646
```

eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/b9e07853-9f34-4656-8f87-5cd1a895149d/view?projectid=c4dbd58a-015e-401a-8373-5c8e33146193&context=wdp

IBM Watson Studio | Upgrade | Ojaswita Negi's Account

My projects / Life Expectancy / Life Expectancy Prediction

In [88]:
```
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

In [89]:
```
wml_credentials={
    "apikey": "h6DbGIe1nB3EEDwggCWgYu1dHoiv5Z6aufmg_Pnh9em5",
    "instance_id": "5971ee95-2fc0-4ece-a0b6-5c7488916ee6",
    "url": "https://eu-gb.ml.cloud.ibm.com"
}
```

In [90]:
```
client = WatsonMachineLearningAPIClient( wml_credentials )
```

In [91]:
```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Ojaswita",
               client.repository.ModelMetaNames.NAME: "LifePrediction"}
```

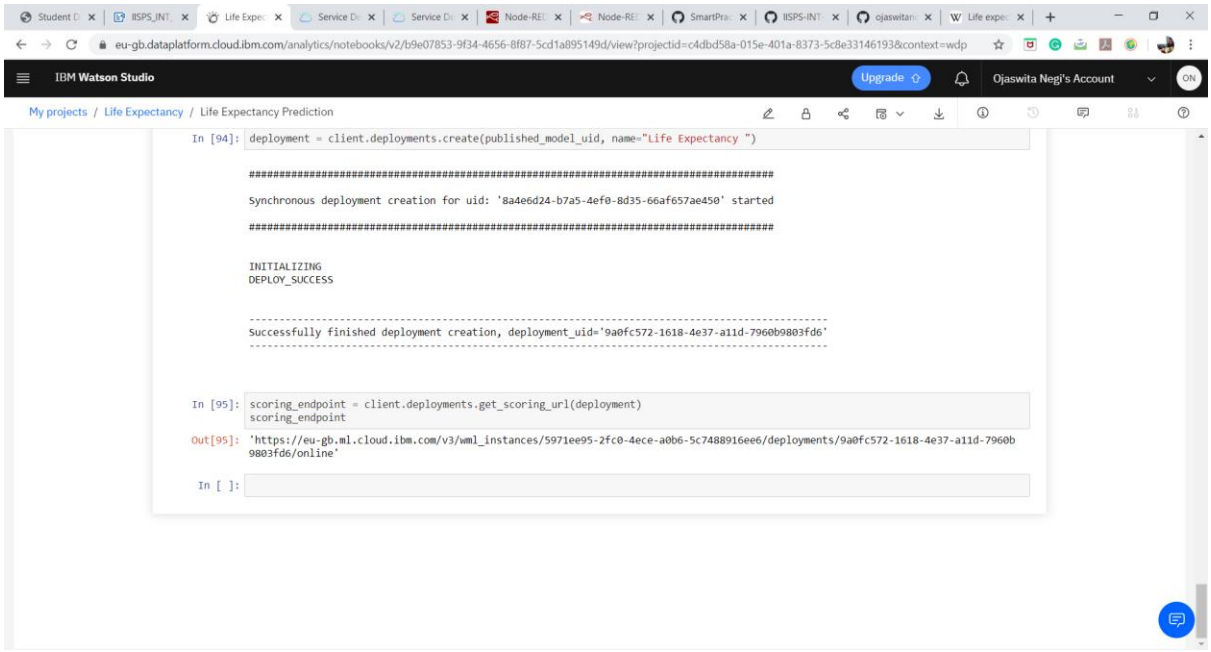
In [92]:
```
model_artifact =client.repository.store_model(lasso, meta_props=model_props)
```

In [93]:
```
published_model_uid = client.repository.get_model_uid(model_artifact)
published_model_uid
```

Out[93]:
```
'8a4e6d24-b7a5-4ef0-8d35-66af657ae450'
```

In [94]:
```
deployment = client.deployments.create(published_model_uid, name="Life Expectancy ")
```

```
#######################################################################################

Synchronous deployment creation for uid: '8a4e6d24-b7a5-4ef0-8d35-66af657ae450' started

#######################################################################################


INITIALIZING
DEPLOY_SUCCESS


------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='9a0fc572-1618-4e37-a11d-7960b9803fd6'
```


IBM Watson Studio
Upgrade
Ojaswita Negi's Account
My projects / Life Expectancy / Life Expectancy Prediction
In [94]:
deployment = client.deployments.create(published_model_uid, name="Life Expectancy ")
Synchronous deployment creation for uid: '8a4e6d24-b7a5-4ef0-8d35-66af657ae450' started
INITIALIZING
DEPLOY_SUCCESS
Successfully finished deployment creation, deployment_uid='9a0fc572-1618-4e37-a11d-7960b9803fd6'
In [95]:
scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint
Out[95]:
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/5971ee95-2fc0-4ece-a0b6-5c7488916ee6/deployments/9a0fc572-1618-4e37-a11d-7960b9803fd6/online'
In [ ]: