Summer Internship Report

Smart Bridge Educational Services Pvt. Ltd

in collaboration with **IBM**

Project Title:

Predicting Life Expectancy using

Machine Learning

Project report submitted by:

Tushar Jindal

Roll number: 18csu218

NCU
THE NORTHCAP UNIVERSITY
NAAC ACCREDITED
(Formerly ITM University, Gurugram)

Internship Offer Letter

SMARTBRIDGE
Let's Bridge the Gap

From:

SmartBridge Educational Services Pvt Ltd.
Plot No 132, Above DCB bank, 2nd floor,
Bapuji Nagar, Habsiguda,
Nacharam Main Road, Hyderabad – 500076

Date: 28/05/2020.

Dear **Tushar Jindal**

SmartBridge Educational Services Pvt Ltd, is pleased to offer a training cum internship opportunity. During this period you would be associated with our mentors and The Smart Practice School Platform.

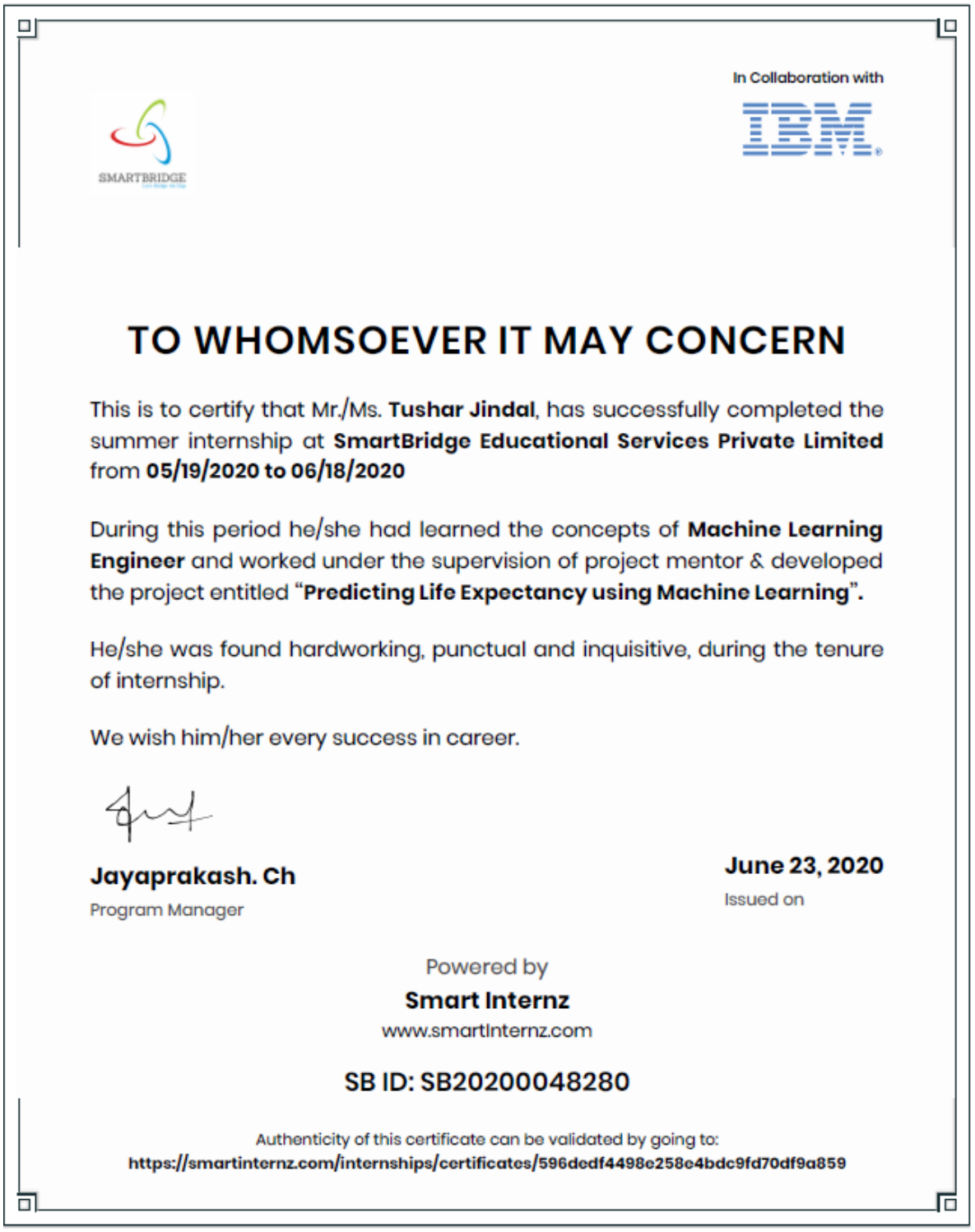
For further details you can contact us on +91 8499004200.

Thanks and Regards,

Ch. Jaya Prakash
Program Manager – SIP2020,
Date: 16/05/2020.

NCU
THE NORTHCAP UNIVERSITY
NAAC ACCREDITED
(Formerly ITM University, Gurugram)

Certificate



In Collaboration with

IBM

SMARTBRIDGE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./Ms. **Tushar Jindal**, has successfully completed the summer internship at **SmartBridge Educational Services Private Limited** from **05/19/2020 to 06/18/2020**

During this period he/she had learned the concepts of **Machine Learning Engineer** and worked under the supervision of project mentor & developed the project entitled "**Predicting Life Expectancy using Machine Learning**".

He/she was found hardworking, punctual and inquisitive, during the tenure of internship.

We wish him/her every success in career.

Jayaprakash. Ch
Program Manager

June 23, 2020
Issued on

Powered by
Smart Internz
www.smartinternz.com

SB ID: SB20200048280

Authenticity of this certificate can be validated by going to:
https://smartinternz.com/internships/certificates/596dedf4498e258e4bdc9fd70df9a859

NCU
THE NORTHCAP UNIVERSITY
NAAC ACCREDITED
(Formerly ITM University, Gurugram)

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **SmartBridge Educational Services Private Limited** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards mentors of **SmartBridge Educational Services Private Limited** for their kind co-operation and encouragement which helped me in completion of this project.

I would like to express my special gratitude and thanks to **IBM** for their collaboration with **SmartBridge Educational Services Private Limited** on this project. IBM allowed me to use their IBM cloud services such as Watson studio and node red for this project.

NCU
THE NORTHCAP UNIVERSITY
NAAC ACCREDITED
(Formerly ITM University, Gurugram)

Abstract

A project to predict the life expectancy using certain factors such as Adult Mortality rate, Infant deaths, Alcohol, Hepatitis B, Measles, BMI, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP of a country, Population, Income composition of resources, Schooling and status of the country in terms of Developing or Developed.

In order to predict life expectancy rate of a given country, we will be using Machine Learning algorithms to draw inferences from the given dataset and give an output.

For better usability by the customer, we are also going to be creating a UI for the user to interact with using Node-Red and integrating it with my machine learning model.

The project has been done in collaboration with IBM therefore we get to use some of their machine learning tools such as Watson studio which helps data scientists and analysts prepare data and build models at scale across any cloud. With its open, flexible multi-cloud architecture and we used Node-RED which is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things.

NCU
THE NORTHCAP UNIVERSITY
NAAC ACCREDITED
(Formerly ITM University, Gurugram)

INDEX

1. Introduction

1.1. Overview

Life expectancy refers to the number of years a person is expected to live. In mathematical terms, life expectancy refers to the expected number of years remaining for an individual at any given age.

The life expectancy for a person or population group depends on several variables such as their lifestyle, access to healthcare, diet, economic status and the relevant mortality and morbidity data. However, as life expectancy is calculated based on averages, a person may live for many years more or less than expected.

In order to predict life expectancy rate of a given country, we will be using Machine Learning algorithms to draw inferences from the given dataset and give an output. For better usability by the customer, we are also going to be creating a UI for the user to interact with using Node-Red.

1.2. Purpose

Economic growth

Predicting life expectancy would play a vital role in judging the growth and development of the economy.

Across countries, high life expectancy is associated with high income per capita. Increase in life expectancy also leads to an increase in the “manpower” of a country.

Population Growth

Helps the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years.

Personal growth

This project would also help an individual assess his/her lifestyle choices and alter them accordingly to lead a longer and healthier life. It would make them more aware of their general health and its improvement or deterioration over time.

Growth in Health Sector

Based on the factors used to calculate life expectancy of an individual and the outcome, health care will be able to fund and provide better services to those with greater need.

Insurance Companies

Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

2. Literature Survey

2.1. Existing Solution

As a result of the evolution of biotechnologies and related technologies such as the development of sophisticated medical equipment, humans are able to enjoy longer life expectancies than previously before. Predicting a human's life expectancy has been a long-term question to humankind. Many calculations and research have been done to create an equation despite it being impractical to simplify these variables into one equation.

Currently there are various smart devices and applications such as smartphone apps and wearable devices that provide wellness and fitness tracking. Some apps provide health related data such as sleep monitoring, heart rate measuring, and calorie expenditure collected and processed by the devices and servers in the cloud. However no existing works provide the Personalized Life expectancy.

2.2. Proposed Solution

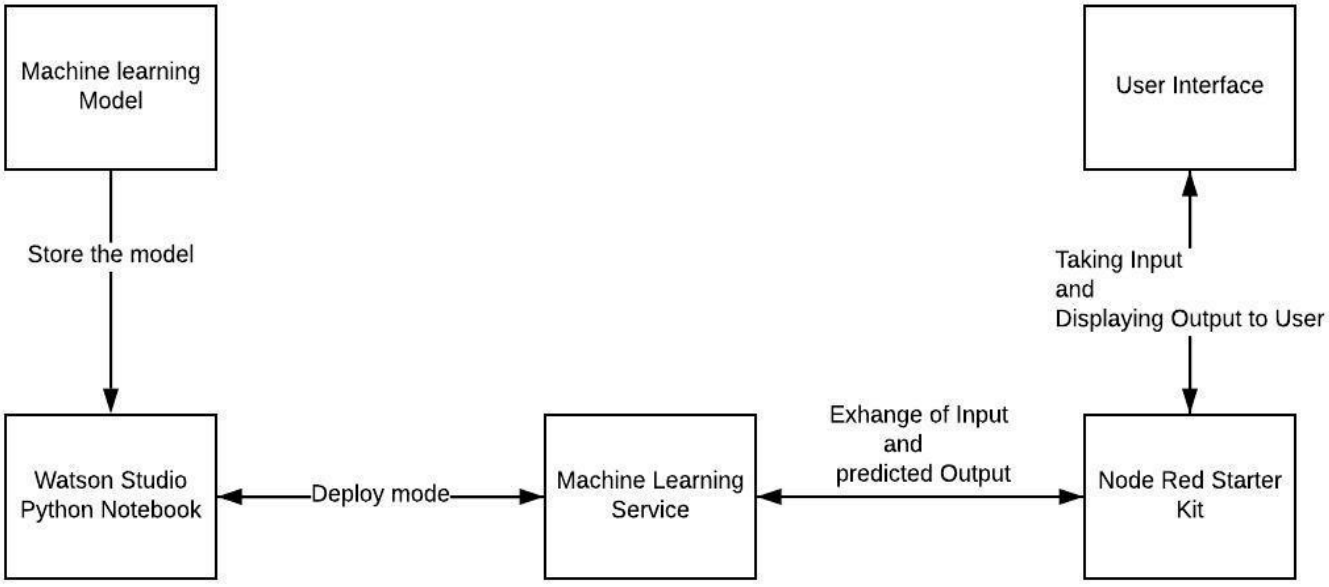
There has been an explosion of breakthroughs in the field of Machine Learning over the past few years. Machine Learning algorithms are capable of a lot and can-do wonders for the healthcare sector.

The proposed solution involves the use of Machine Learning algorithms specifically Regression models such as Linear Regression etc. Life expectancy is highly correlated over time among countries and between males and females. These associations can be used to improve forecasts. Here we propose a method for forecasting life expectancy of an individual from a country taking into certain factors such as Adult Mortality rate, Infant deaths, Alcohol, Hepatitis B, Measles, BMI, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP of a country, Population, Income composition of resources, Schooling and status of the country in terms of Developing or Developed.

This machine learning model will be made accessible to the users by integrating it with Node-Red to create an interactive and user-friendly User Interface.

3. Theoretical Analysis

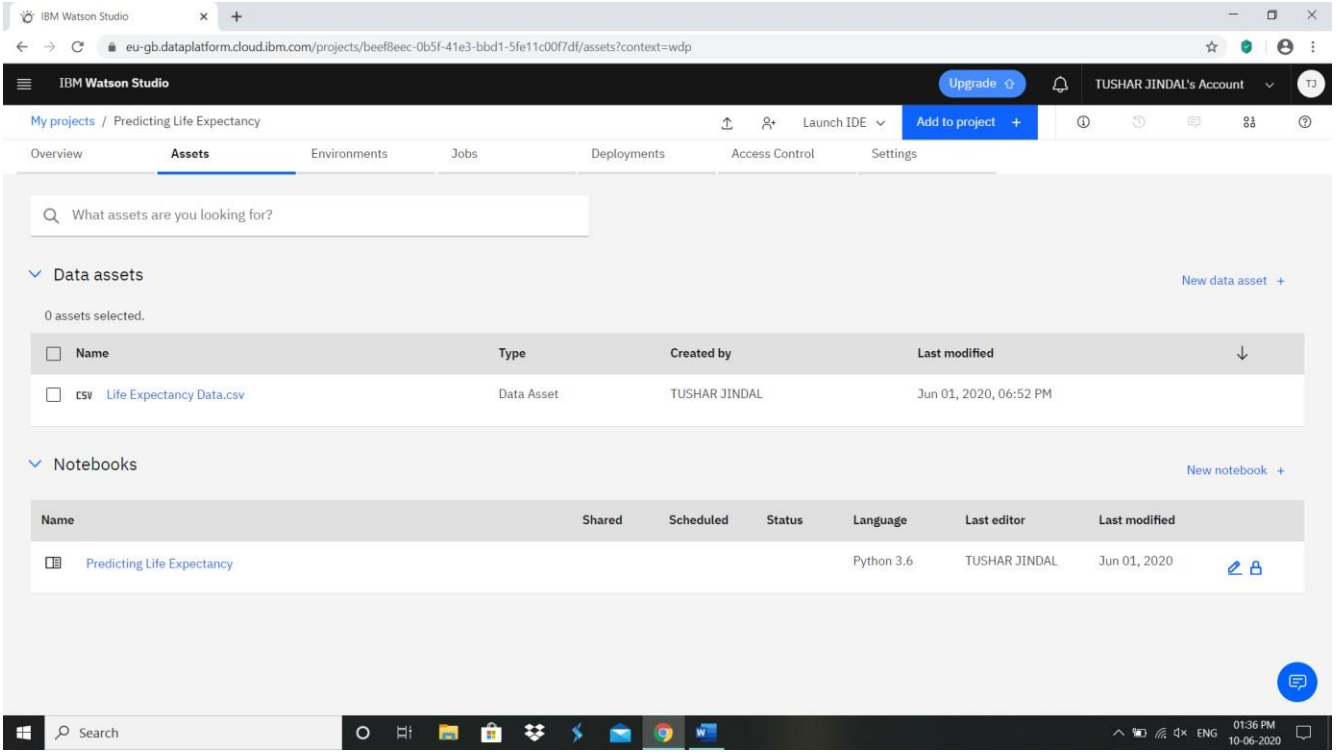
3.1. Block Diagram

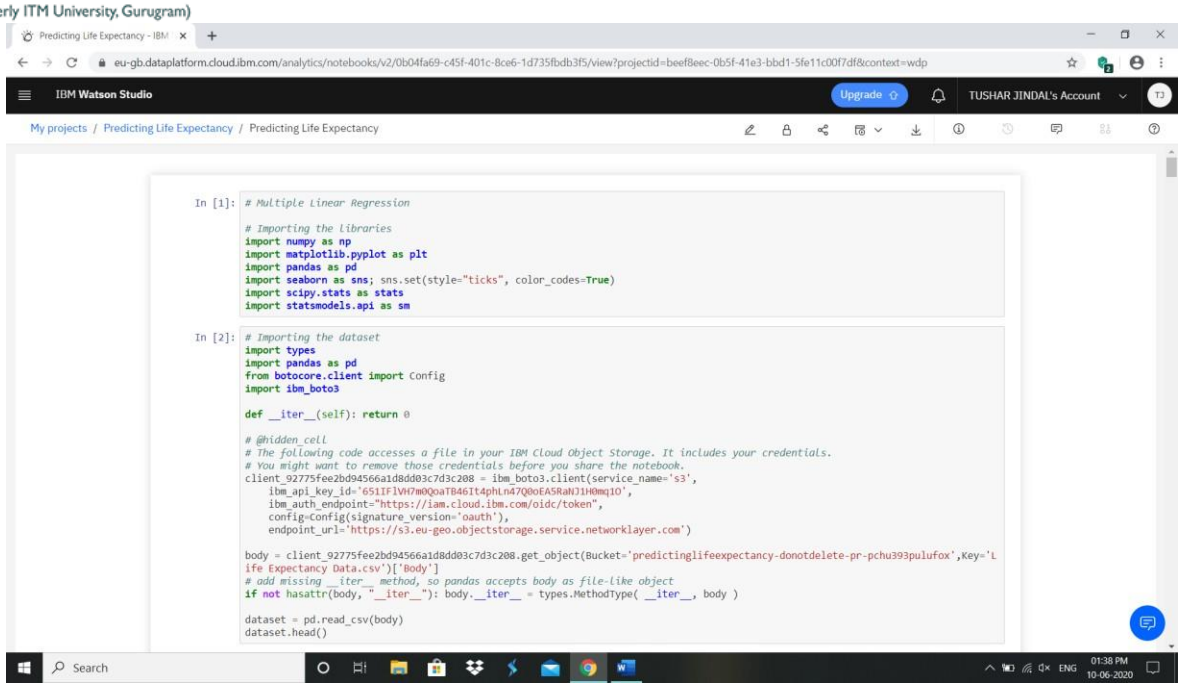


3.2. Hardware/ Software Designing

Model Designing **(Watson Studio)** :

Steps: New Project => Create an empty Project => Give project name => Click Create => Add to Project => Notebook

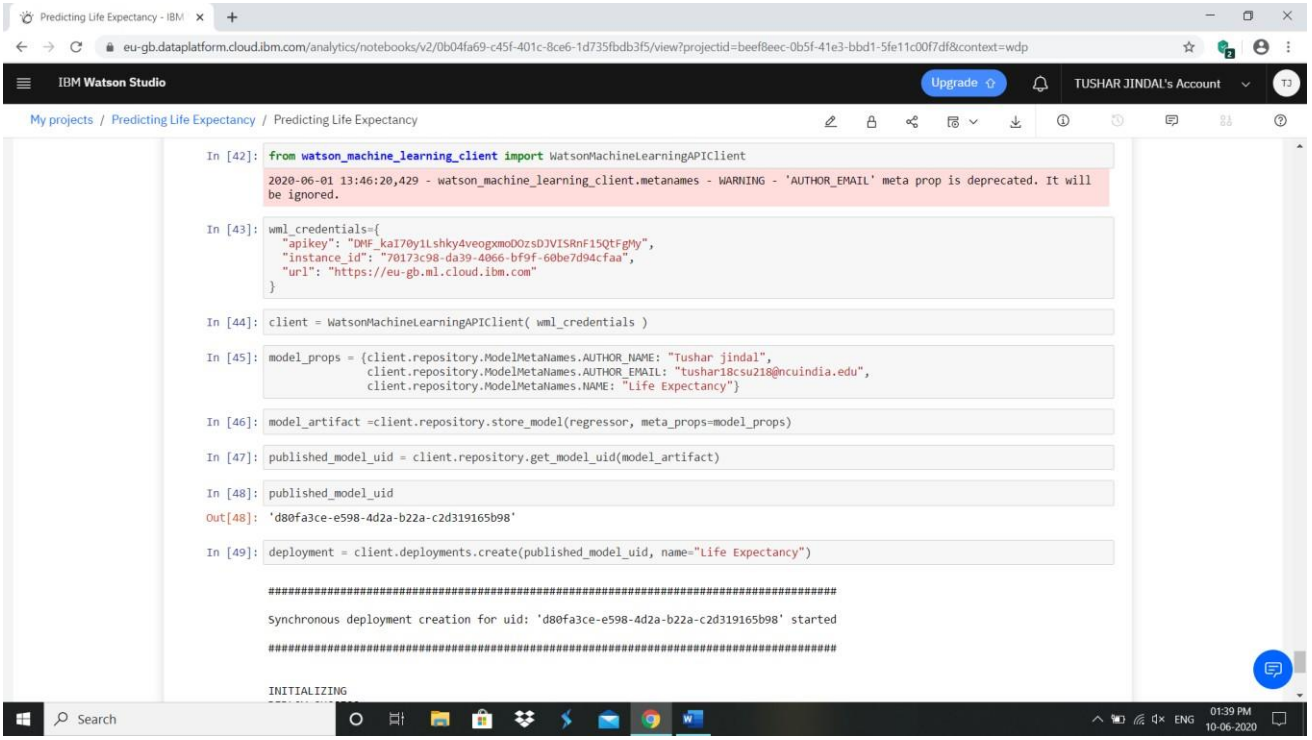


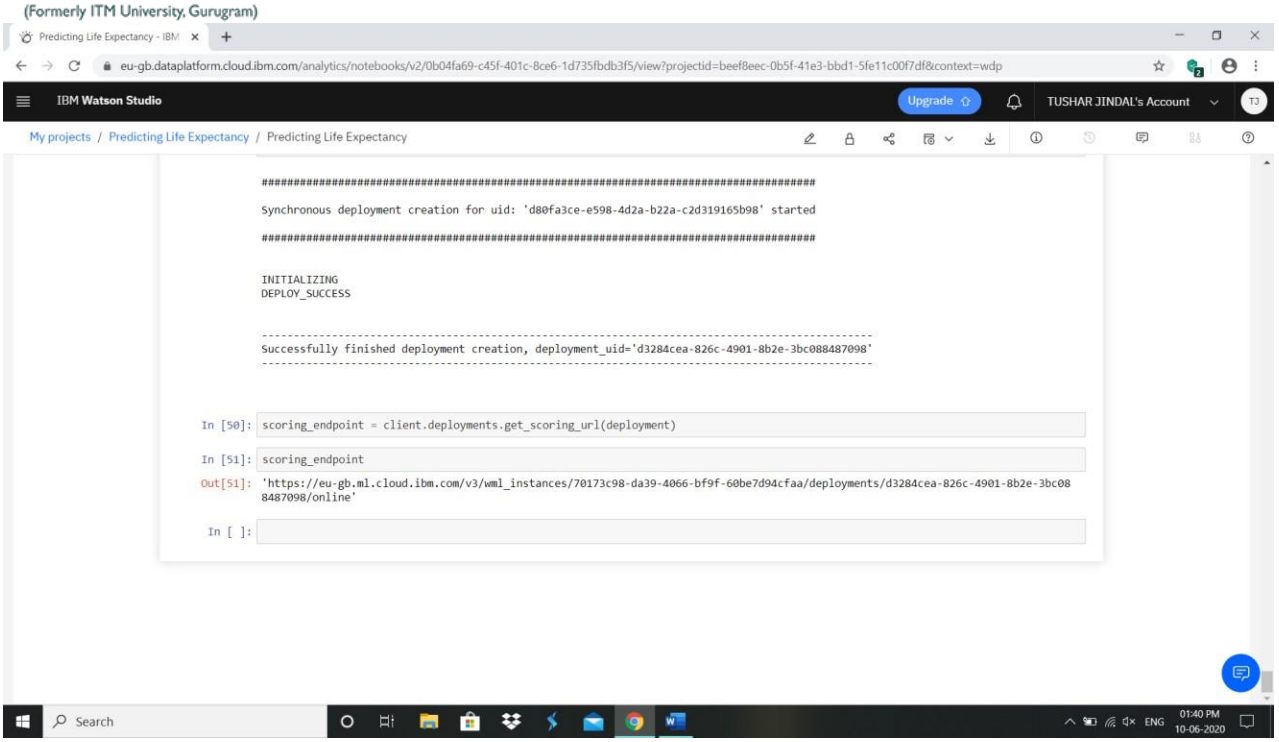


Scoring Endpoint:

For wml credentials, replace with your own credentials of the service.

Services => Machine Learning Service => Service Credentials => Copy the credentials



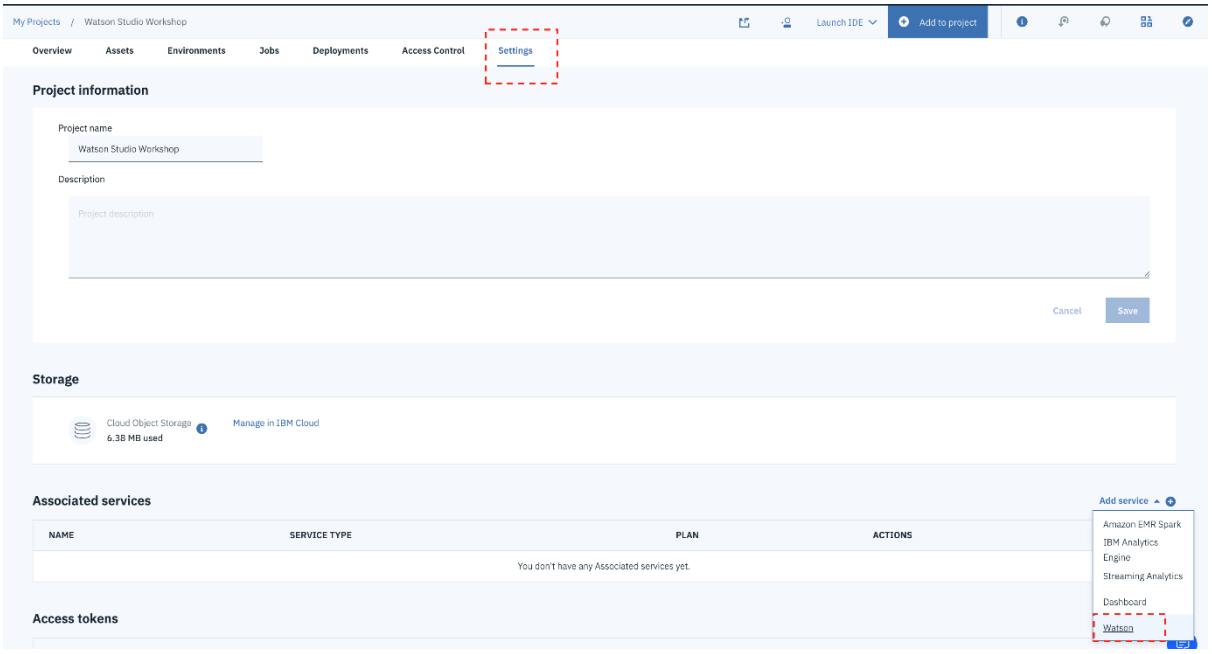


Build and Deploy models in Watson Studio

1. WML Credentials

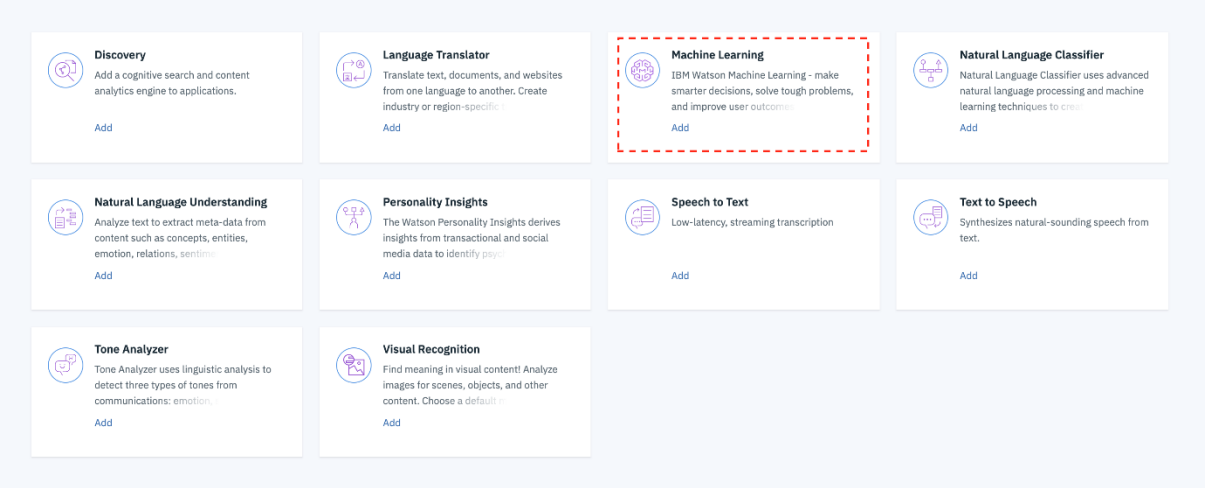
You should have a Watson Machine Learning service provisioned on IBM Cloud and associated with the current project:

Click on the **Settings** in the project view, locate **Associated services** => **Add Service** => **Watson**.

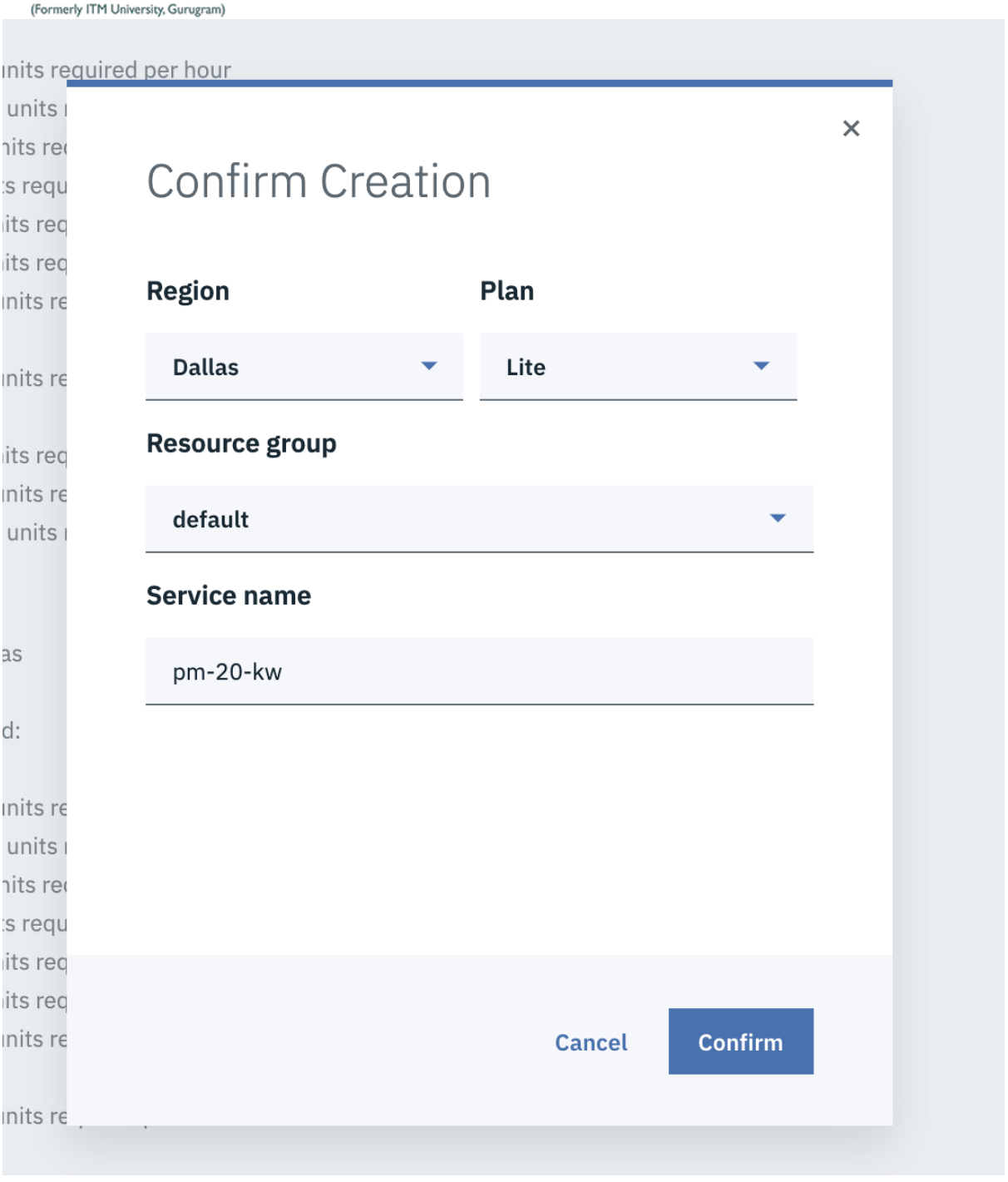


Add WML service

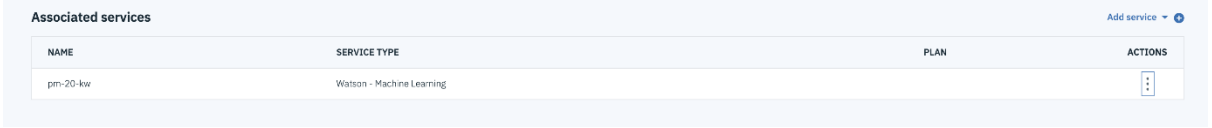
Choose **Machine Learning**, click on **Add**



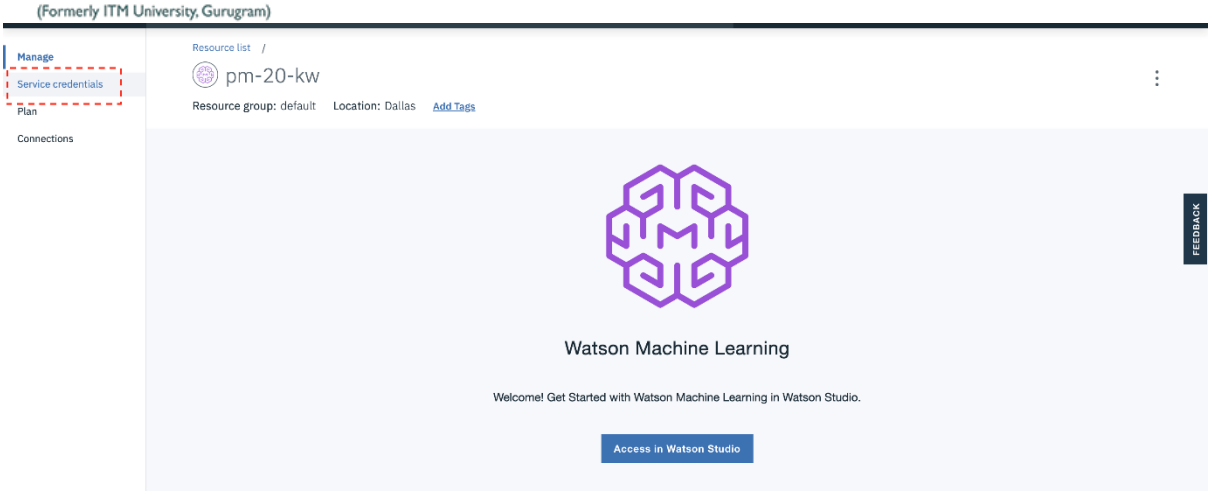
If you don’t have an existing WML service provisioned, click **New**, click **Create**. Confirm creation.



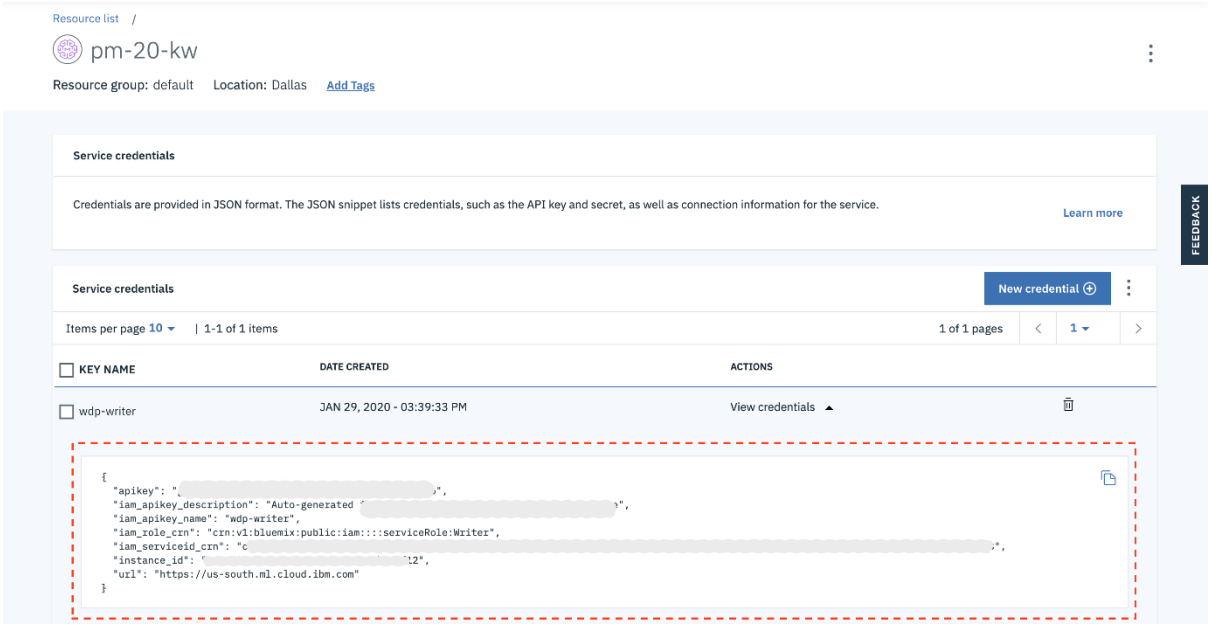
Now you should have a WML service associated with this project.



Login into IBM Cloud dashboard, it should be provisioned under the **Service** tab. Click on it to access the home page of the WML service. Click on **Service credentials"**.



Keep a copy of the service credentials, you will need it later in the notebook when deploying models.



Create a Node-Red starter application

Step 1. Find the Node-RED Starter in the IBM Cloud catalog

1. Log in to IBM cloud.
2. Open the catalog and search for **node-red**.
3. Click on the **Node-RED App** title.

   This will show you an overview of the Starter Kit and what it provides.

Step 2. Create your application

Now you need to create the Node-RED Starter application.

1. On the *Create* tab, a randomly generated **App name** will be suggested. Either accept that default name or provide a unique name for your application. This will become part of the application URL.

   Note: If the name is not unique, you will see an error message and you must enter a different name before you can continue.

2. The Node-RED Starter application requires an instance of the **Cloudant database service** to store your application flow configuration. Select the region the service should be created in.
3. Click the **Create** button to continue. This will create your application, but it is not yet deployed to IBM Cloud.

Step 3. Enable the Continuous Delivery feature

At this point, you have created the application and the resources it requires, but you have not deployed it anywhere to run. This step shows how to setup the Continuous Delivery feature that will deploy your application into the **Cloud Foundry** space of IBM Cloud.

1. On the next screen, click the **Deploy your app** button to enable the *Continuous Delivery* feature for your application.
2. You will need to create an **IBM Cloud API** key to allow the deployment process to access your resources. Click the **New** button to create the key. A message dialog will appear. You can accept the default values and confirm to close the dialog.
3. Increase the **Memory allocation per instance** slider to at least 128MB. If you do not increase the memory allocation, your Node-RED application might not have sufficient memory to run successfully.
4. The Node-RED Starter kit only supports deployment to the **Cloud Foundry** space of IBM Cloud. Select the **region** to deploy your application to. This should match the region you created your Cloudant instance in.

   Click **Next** to continue.

5. After a few moments, the **Deployment Automation** section will refresh with the details of your newly created Delivery Pipeline. The Status field of the pipeline will eventually show **In progress**. That means your application is being built and deployed.

   Click on the **Status** field to see the full status of the Delivery Pipeline.

6. The Deploy stage will take a few minutes to complete. You can click on the **View logs and history** link to check its progress. Eventually the Deploy stage will go green to show it has passed. This means your Node-RED Starter application is now running.

Step 4. Open the Node-RED application

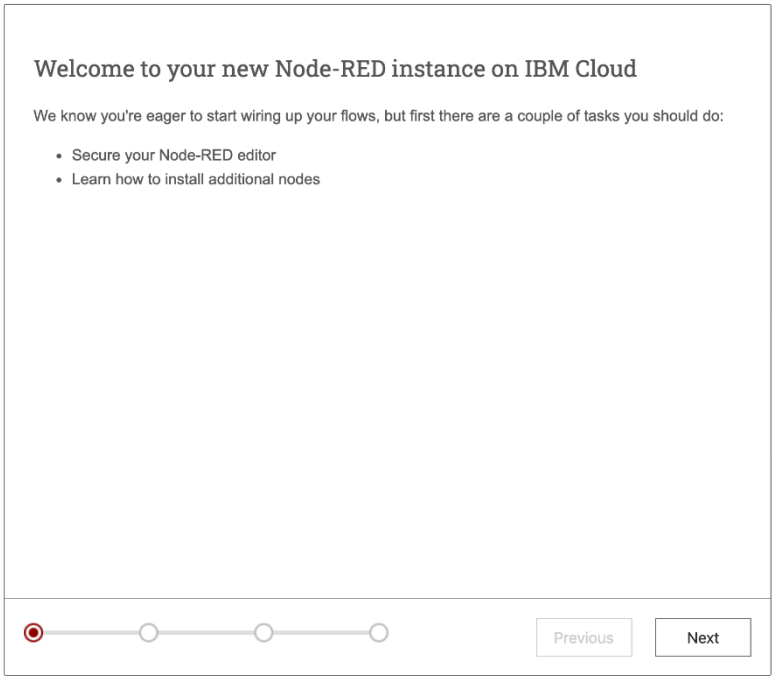
Now that you have deployed your Node-RED application, let's open it up!

1. Back on the application details page, you should now see the **App URL**, **Source** and **Deployment target** fields filled in.
2. Click on the **App URL** to open up your Node-RED application in a new browser tab.

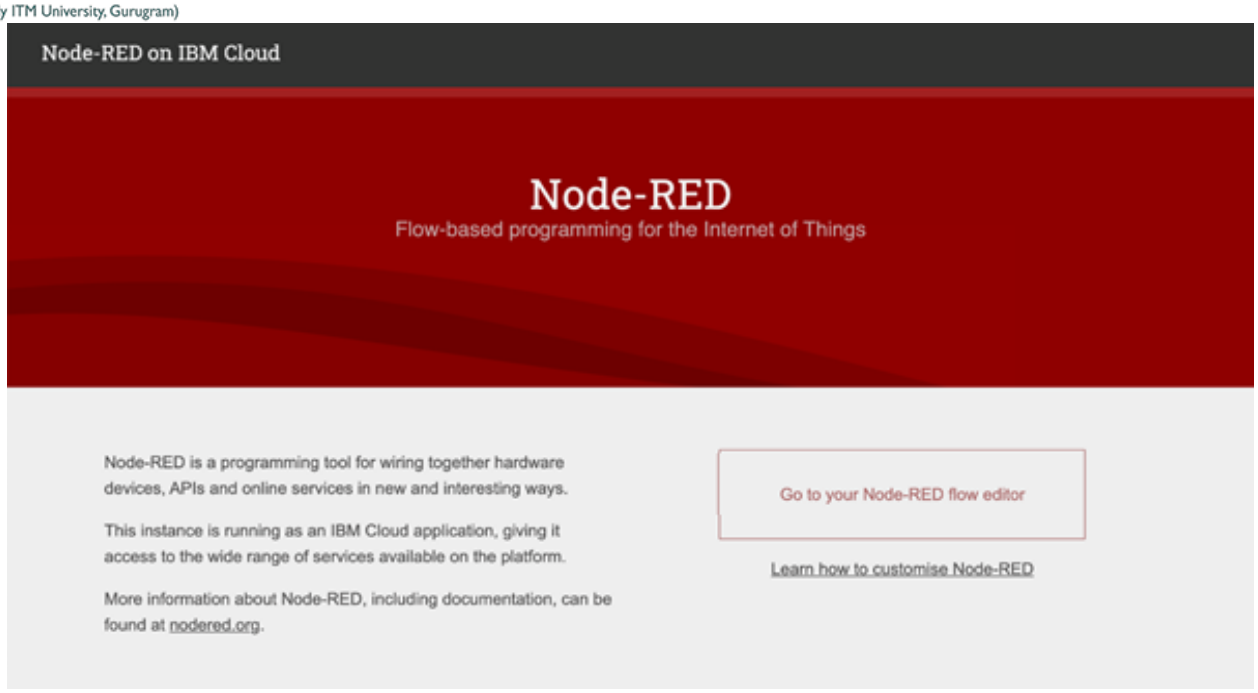
Step 5. Configure your Node-RED application

The first time you open your Node-RED app, you will need to configure it and set up security.

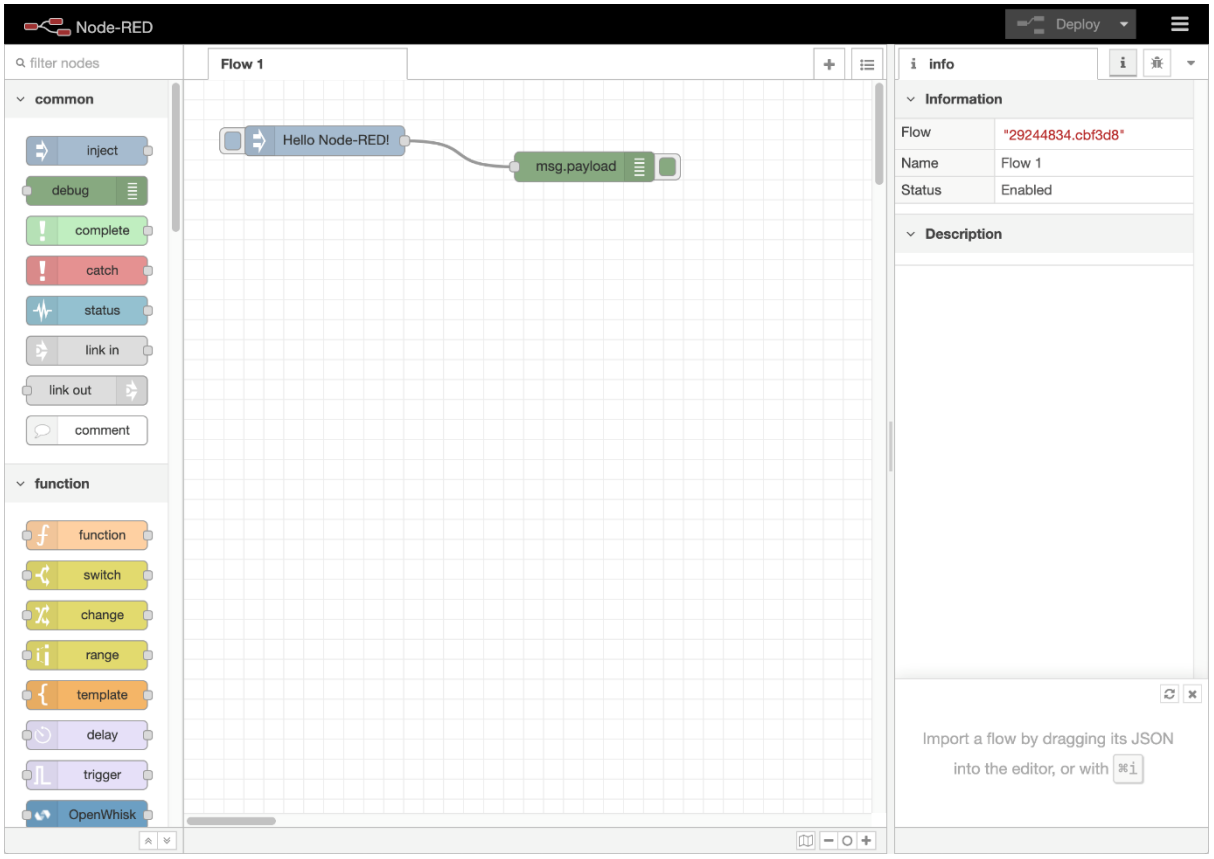
1. A new browser tab will open with the Node-RED start page.



2. On the initial screen, click **Next** to continue.
3. Secure your Node-RED editor by providing a **username** and **password**. If you need to change these at any point
4. The final screen summarizes the options you've made and highlights the environment variables you can use to change the options in the future. Click **Finish** to proceed.
5. Node-RED will save your changes and then load the main application. From here you can click the **Go to your Node-RED flow editor** button to open the editor.



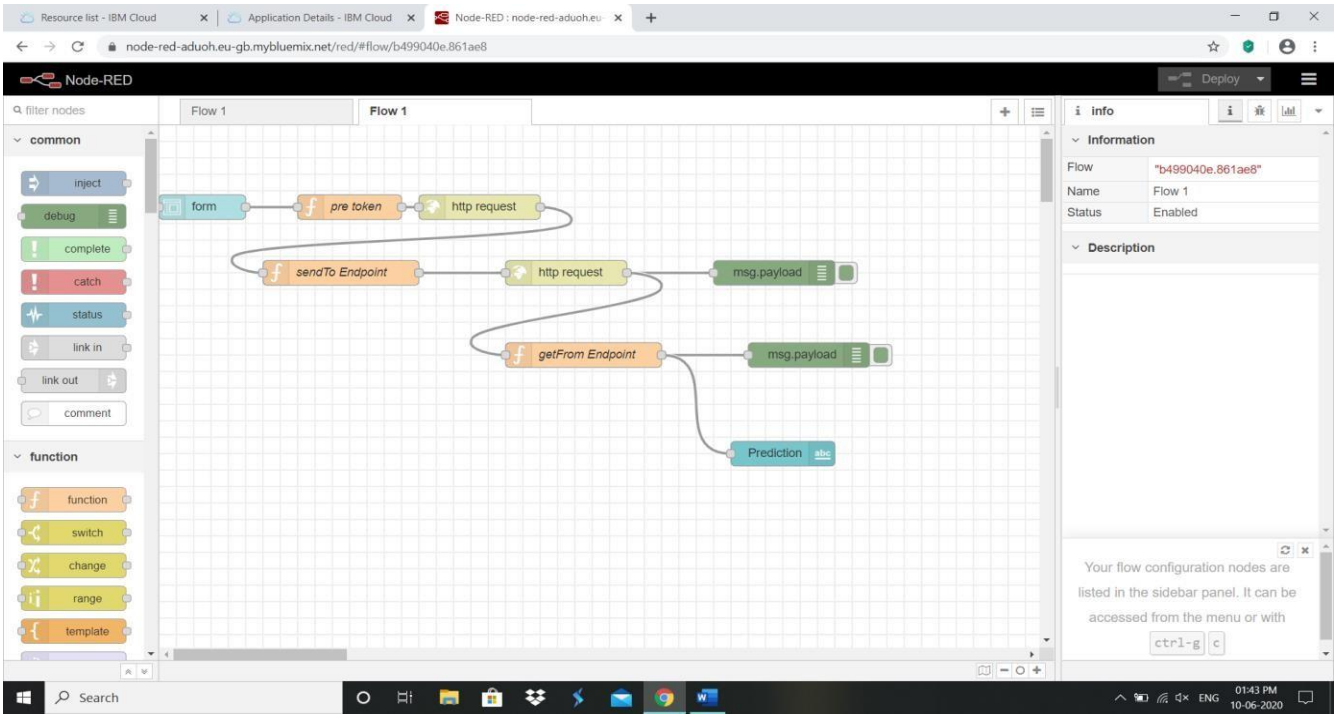
The Node-RED editor opens showing the default flow.

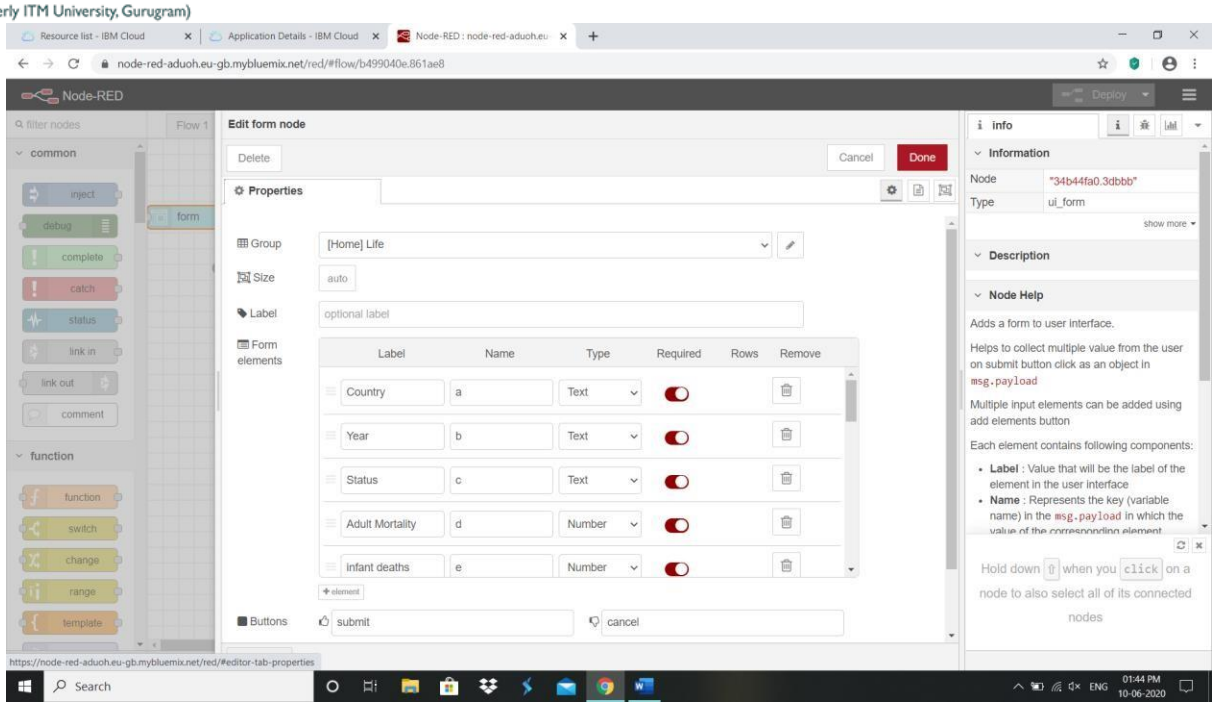


User Interface Integration with ML Model **(Node-Red)** : Nodes: 1) Form Node: Edit => Add New UI Tab

2) Function Node: To obtain access to Machine Learning Services. Requires API Key

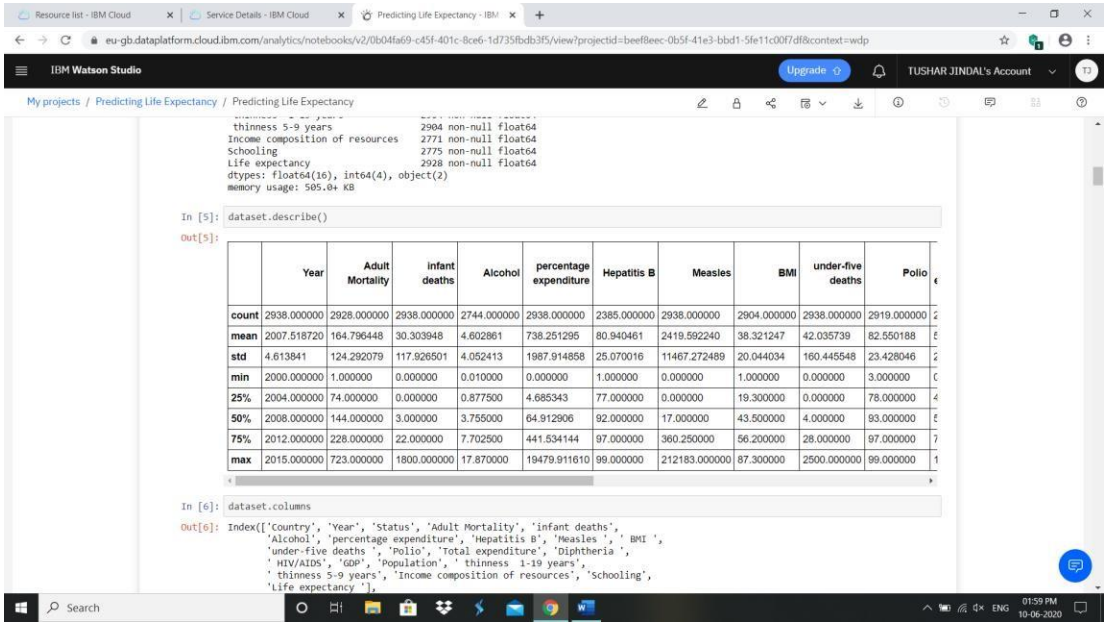
3) HTTP Request Node: POST method and returns a parsed JSON object. Gains access to Machine Learning services.



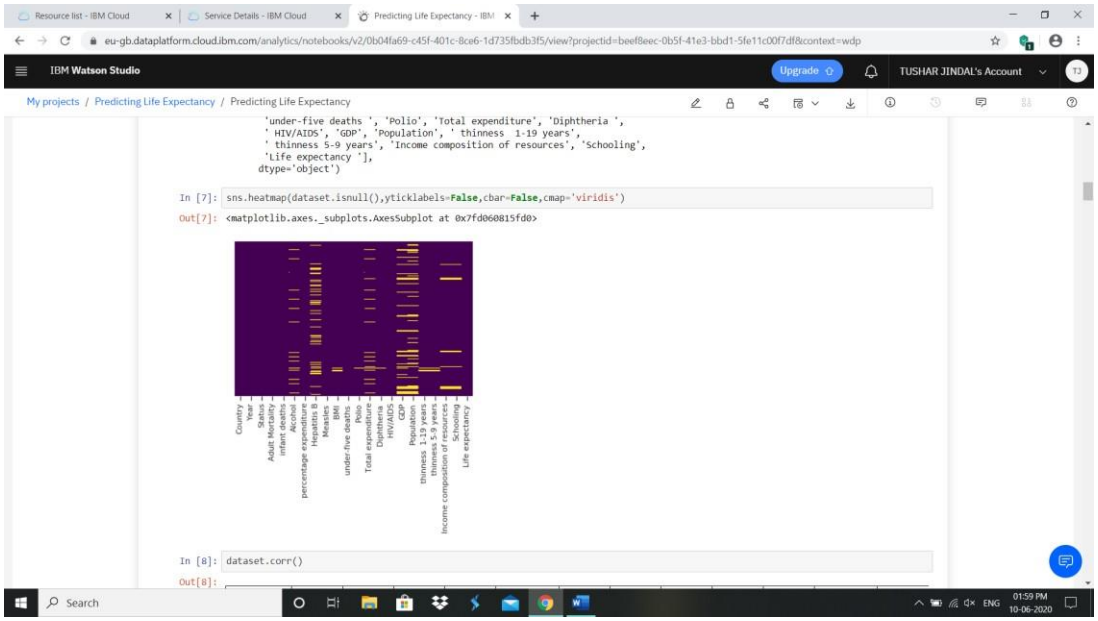

Resource list - IBM Cloud
Application Details - IBM Cloud
Node-RED : node-red-aduoh.eu
node-red-aduoh.eu-gb.mybluemix.net/red/#flow/b499040e.861ae8
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
form
Edit form node
Delete
Cancel
Done
Properties
Group
[Home] Life
Size
auto
Label
optional label
Form elements
Label
Name
Type
Required
Rows
Remove
Country
a
Text
Year
b
Text
Status
c
Text
Adult Mortality
d
Number
infant deaths
e
Number
element
Buttons
submit
cancel
info
Information
Node
"34b44fa0.3dbbb"
Type
ui_form
show more
Description
Node Help
Adds a form to user interface.
Helps to collect multiple value from the user on submit button click as an object in msg.payload
Multiple input elements can be added using add elements button
Each element contains following components:
Label : Value that will be the label of the element in the user interface
Name : Represents the key (variable name) in the msg.payload in which the
Hold down ⇧ when you click on a node to also select all of its connected nodes
https://node-red-aduoh.eu-gb.mybluemix.net/red/#editor-tab-properties
Search
ENG
01:44 PM
10-06-2020

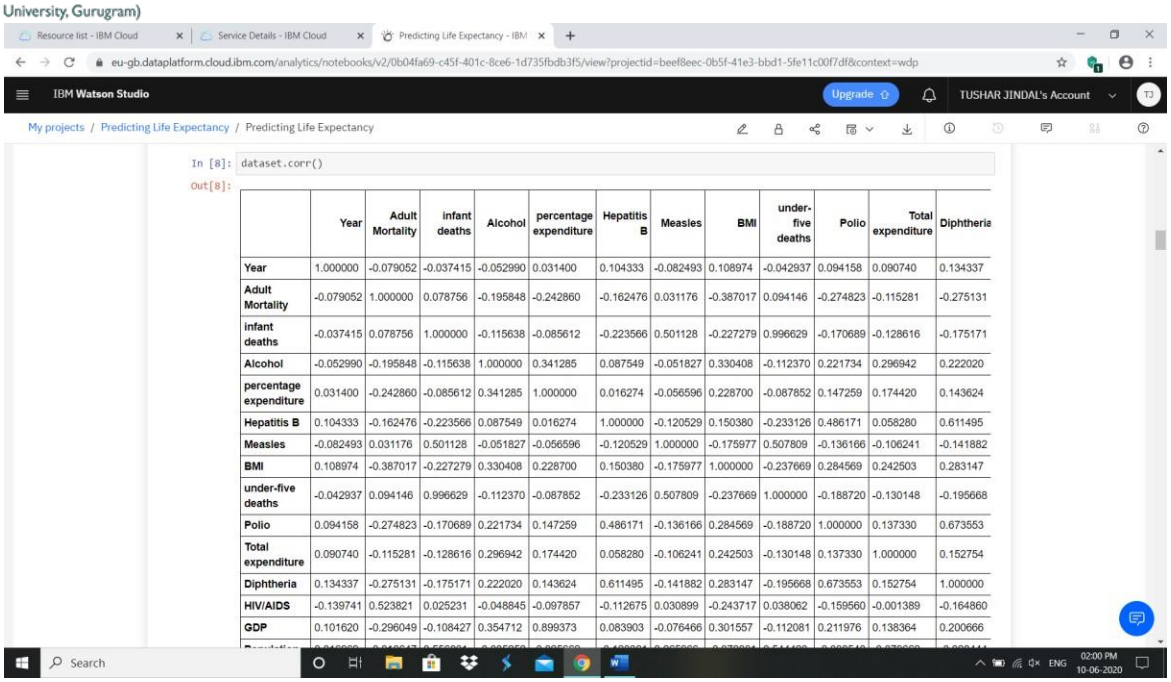
4. Experimental Investigations

Analyzing the relations between various features can help us improve the performance of the model as well as decide which model would be more suitable.

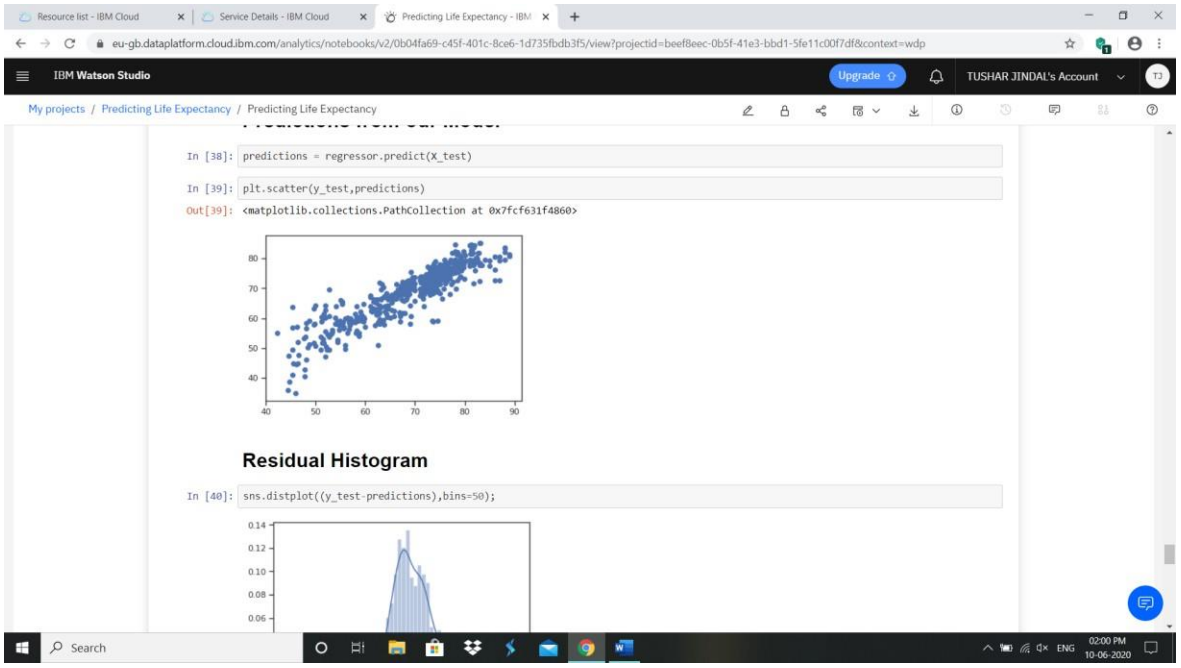


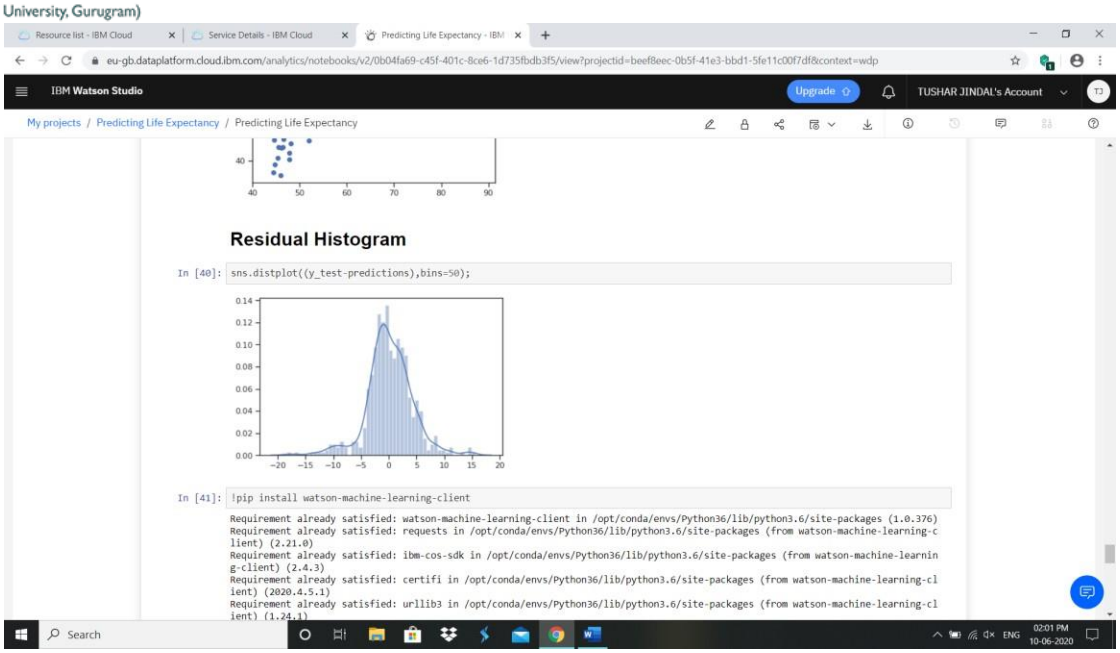
	Year	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio
count	2938.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000	2938.000000	2904.000000	2938.000000	2919.000000
mean	2007.518720	164.796448	30.303948	4.602861	738.251295	80.940461	2419.592240	38.321247	42.035739	82.550188
std	4.613841	124.292079	117.926501	4.052413	1987.914858	25.070016	11467.272489	20.044034	160.445548	23.428046
min	2000.000000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000	3.000000
25%	2004.000000	74.000000	0.000000	0.877500	4.685343	77.000000	0.000000	19.300000	0.000000	78.000000
50%	2008.000000	144.000000	3.000000	3.755000	64.912906	92.000000	17.000000	43.500000	4.000000	93.000000
75%	2012.000000	228.000000	22.000000	7.702500	441.534144	97.000000	360.250000	56.200000	28.000000	97.000000
max	2015.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2500.000000	99.000000



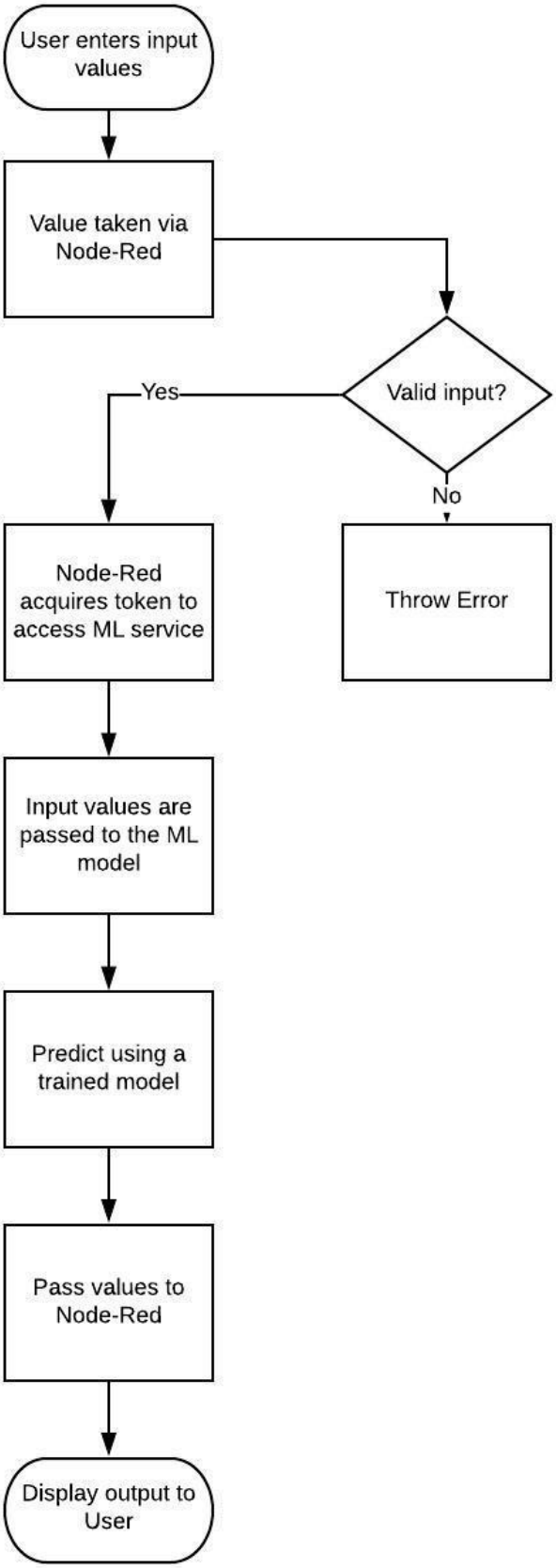


	Year	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria
Year	1.000000	-0.079052	-0.037415	-0.052990	0.031400	0.104333	-0.082493	0.108974	-0.042937	0.094158	0.090740	0.134337
Adult Mortality	-0.079052	1.000000	0.078756	-0.195848	-0.242860	-0.162476	0.031176	-0.387017	0.094146	-0.274823	-0.115281	-0.275131
infant deaths	-0.037415	0.078756	1.000000	-0.115638	-0.085612	-0.223566	0.501128	-0.227279	0.996629	-0.170689	-0.128616	-0.175171
Alcohol	-0.052990	-0.195848	-0.115638	1.000000	0.341285	0.087549	-0.051827	0.330408	-0.112370	0.221734	0.296942	0.222020
percentage expenditure	0.031400	-0.242860	-0.085612	0.341285	1.000000	0.016274	-0.056596	0.228700	-0.087852	0.147259	0.174420	0.143624
Hepatitis B	0.104333	-0.162476	-0.223566	0.087549	0.016274	1.000000	-0.120529	0.150380	-0.233126	0.486171	0.058280	0.611495
Measles	-0.082493	0.031176	0.501128	-0.051827	-0.056596	-0.120529	1.000000	-0.175977	0.507809	-0.136166	-0.106241	-0.141882
BMI	0.108974	-0.387017	-0.227279	0.330408	0.228700	0.150380	-0.175977	1.000000	-0.237669	0.284569	0.242503	0.283147
under-five deaths	-0.042937	0.094146	0.996629	-0.112370	-0.087852	-0.233126	0.507809	-0.237669	1.000000	-0.188720	-0.130148	-0.195668
Polio	0.094158	-0.274823	-0.170689	0.221734	0.147259	0.486171	-0.136166	0.284569	-0.188720	1.000000	0.137330	0.673553
Total expenditure	0.090740	-0.115281	-0.128616	0.296942	0.174420	0.058280	-0.106241	0.242503	-0.130148	0.137330	1.000000	0.152754
Diphtheria	0.134337	-0.275131	-0.175171	0.222020	0.143624	0.611495	-0.141882	0.283147	-0.195668	0.673553	0.152754	1.000000
HIV/AIDS	-0.139741	0.523821	0.025231	-0.048845	-0.097857	-0.112675	0.030899	-0.243717	0.038062	-0.159560	-0.001389	-0.164860
GDP	0.101620	-0.296049	-0.108427	0.354712	0.899373	0.083903	-0.076466	0.301557	-0.112081	0.211976	0.138364	0.200666

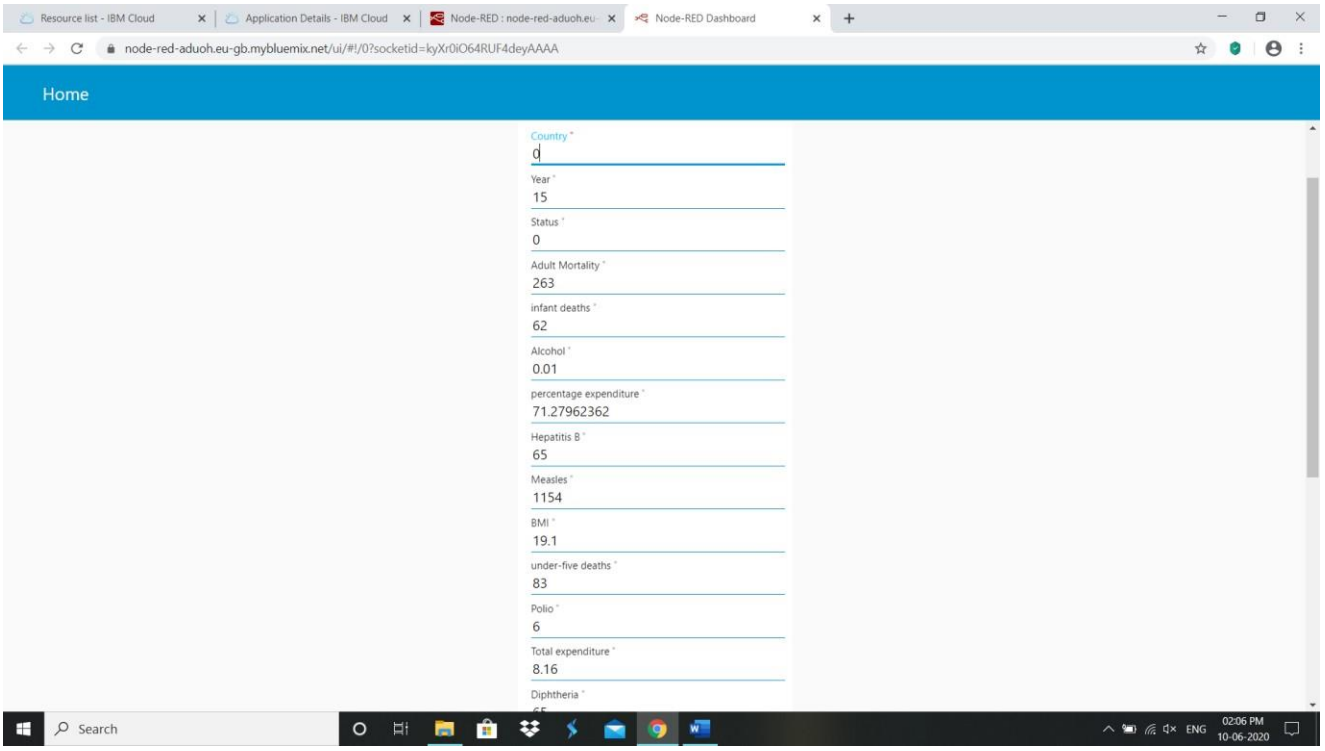


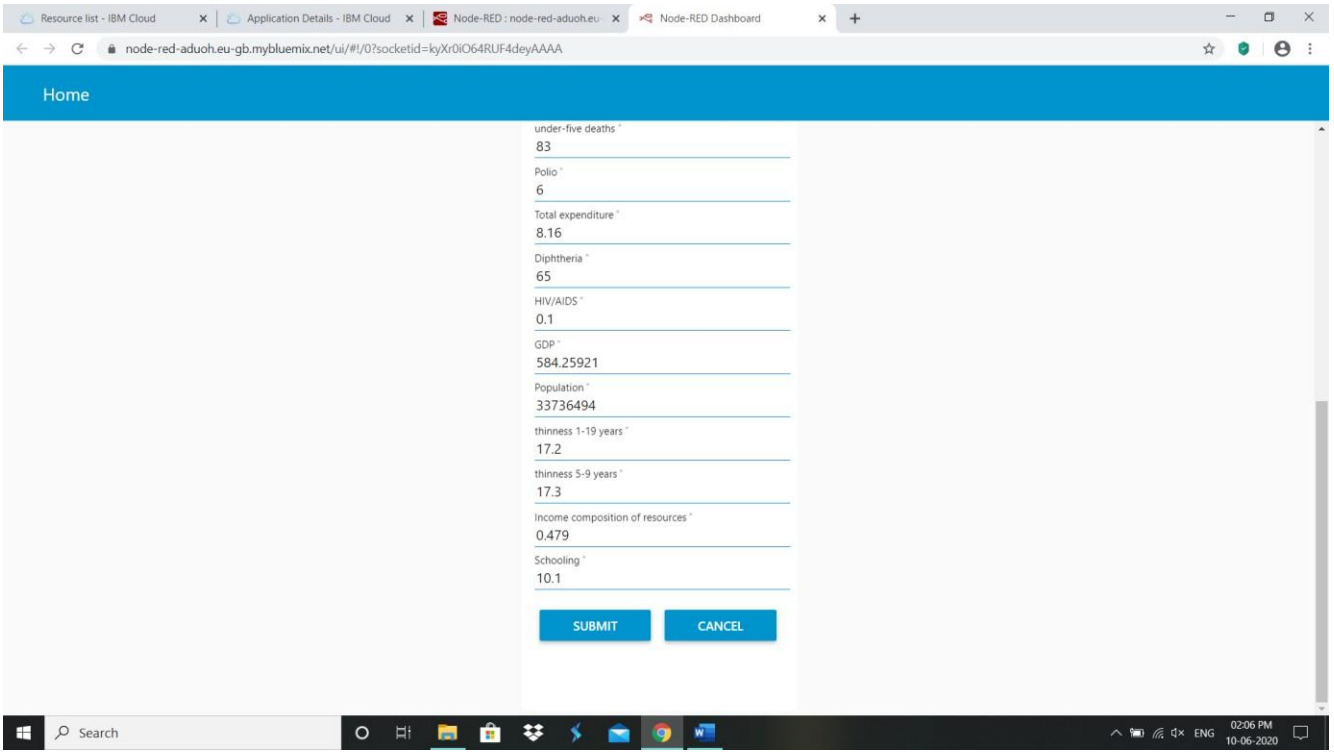

Resource list - IBM Cloud
Service Details - IBM Cloud
Predicting Life Expectancy - IBM
eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/0b04fa69-c45f-401c-8ce6-1d735fbdb3f5/view?projectid=beef8eec-0b5f-41e3-bbd1-5fe11c00f7df&context=wdp
IBM Watson Studio
Upgrade
TUSHAR JINDAL's Account
My projects / Predicting Life Expectancy / Predicting Life Expectancy
Residual Histogram
In [40]: sns.distplot((y_test-predictions),bins=50);
In [41]: !pip install watson-machine-learning-client
Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python36/lib/python3.6/site-packages (1.0.376)
Requirement already satisfied: requests in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (2.21.0)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (2.4.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (2020.4.5.1)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (1.24.1)
Search
ENG
02:01 PM
10-06-2020

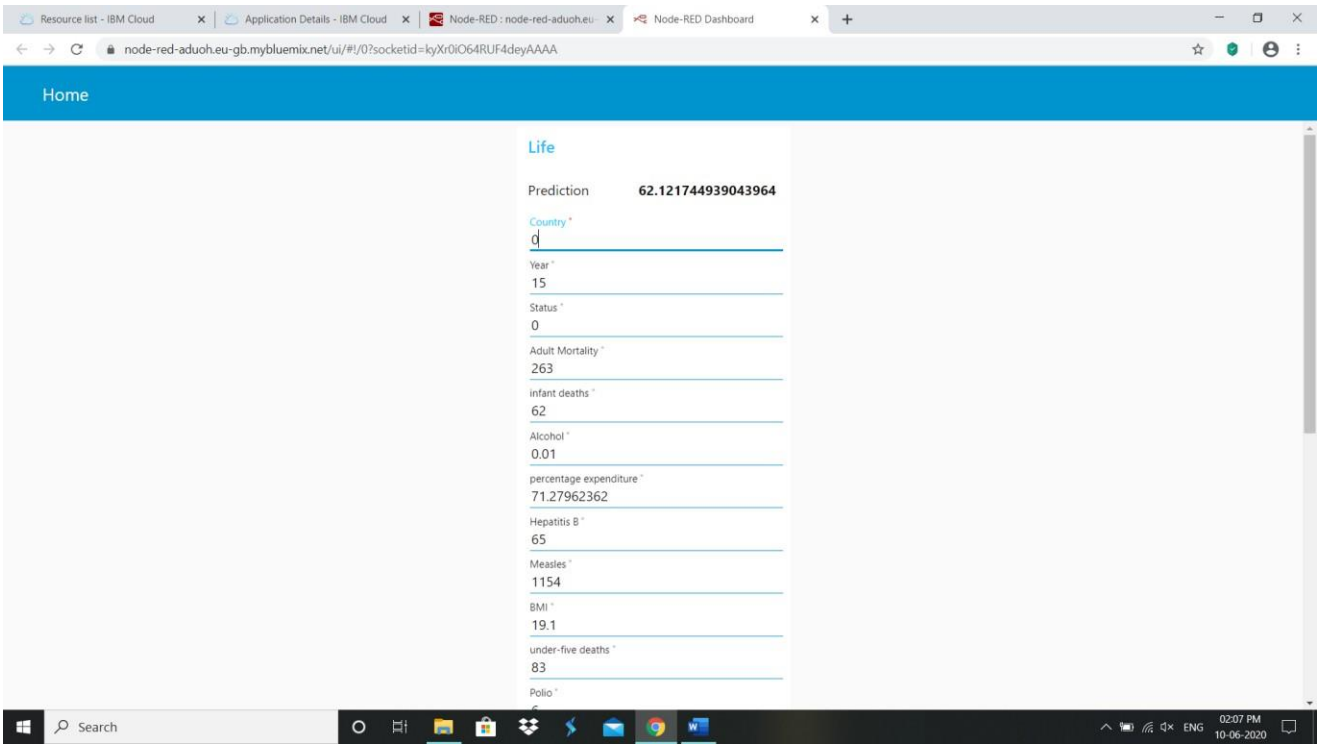
5. Flowchart


User enters input values
Value taken via Node-Red
Valid input?
Yes
No
Node-Red acquires token to access ML service
Throw Error
Input values are passed to the ML model
Predict using a trained model
Pass values to Node-Red
Display output to User

6. Result


Resource list - IBM Cloud
Application Details - IBM Cloud
Node-RED : node-red-aduoh.eu
Node-RED Dashboard
node-red-aduoh.eu-gb.mybluemix.net/ui/#!/0?socketid=kyXr0iO64RUF4deyAAAA
Home
Country *
0
Year *
15
Status *
0
Adult Mortality *
263
infant deaths *
62
Alcohol *
0.01
percentage expenditure *
71.27962362
Hepatitis B *
65
Measles *
1154
BMI *
19.1
under-five deaths *
83
Polio *
6
Total expenditure *
8.16
Diphtheria *
Search
ENG
02:06 PM
10-06-2020


Resource list - IBM Cloud
Application Details - IBM Cloud
Node-RED : node-red-aduoh.eu
Node-RED Dashboard
node-red-aduoh.eu-gb.mybluemix.net/ui/#!/0?socketid=kyXr0iO64RUF4deyAAAA
Home
under-five deaths *
83
Polio *
6
Total expenditure *
8.16
Diphtheria *
65
HIV/AIDS *
0.1
GDP *
584.25921
Population *
33736494
thinness 1-19 years *
17.2
thinness 5-9 years *
17.3
Income composition of resources *
0.479
Schooling *
10.1
SUBMIT
CANCEL
Search
ENG
02:06 PM
10-06-2020


Resource list - IBM Cloud
Application Details - IBM Cloud
Node-RED : node-red-aduoh.eu
Node-RED Dashboard
node-red-aduoh.eu-gb.mybluemix.net/ui/#!/0?socketid=kyXr0iO64RUF4deyAAAA
Home
Life
Prediction
62.121744939043964
Country *
0
Year *
15
Status *
0
Adult Mortality *
263
infant deaths *
62
Alcohol *
0.01
percentage expenditure *
71.27962362
Hepatitis B *
65
Measles *
1154
BMI *
19.1
under-five deaths *
83
Polio *
Search
ENG
02:07 PM
10-06-2020

7. Advantages and Disadvantages

Advantages:

One of the biggest advantages of embedding machine learning algorithms is their ability to improve over time. Machine learning technology typically improves efficiency and accuracy thanks to the ever-increasing amounts of data that are processed.

The application learns the patterns and trends hidden within the data without human intervention which makes predicting much simpler and easier. The more data is fed to the algorithm, the higher the accuracy of the algorithm is. It is also the key component in technologies for automation.

Using Node-Red also simplifies the effort put into a creating the front-end. The programmer doesn't need extensive knowledge on HTML and JavaScript. It also makes the integration between Machine learning model and the UI much easier.

Disadvantages:

Using machine learning interface comes with its own problems. Since the whole point of it is minimize human involvement, it also makes error detection and fixing much more problematic. It takes a lot of time to identify the root cause for the problem.

Machine learning can also be very time-consuming. When the size of the data fed to the machine learning is very large, the computational cost and the time taken to train the model on the data increases drastically. This can increase the cost of resources required to implement the application on a large scale.

At the same time, Node-Red does not give many features to customize our UI.

8. Applications

1) Personalized Life Expectancy: Individuals can predict their own life expectancy by inputting values in the corresponding fields. This could help make people more aware of their general health, and its improvement or deterioration over time. This may motivate them to make healthier lifestyle choices.

2) Government: It could help the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years. Across countries, high life expectancy is associated with high income per capita. Increase in life
expectancy also leads to an increase in the “manpower” of a country. The knowledge asset of a country increases with the number of individuals in a country.

3) Health Sector: Based on the factors used to calculate life expectancy of an individual and the outcome, health care will be able to fund and provide better services to those with greater need.

4) Insurance Companies: Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

9. Conclusion

Predicting lifespan of human beings can greatly alter our lives. Human behavior and activities are so unpredictable, it may almost be impossible to correctly predict lifespan. However, with the help of Machine learning algorithms such as Regression models, we can get close to predicting a roundabout value.

This breakthrough can widely impact health sectors and economic sectors by improving the resources, funds and services provided to the common people. It can also increase the ease of access to the individuals.

With the help of Machine Learning algorithms, one can ease the process of automating the application and predicting the expectancy with an admirable accuracy. It also reduces the effort and time put into deploying the application and making it more accessible to the users.

Future Scope

For future use, one can integrate the life expectancy prediction with providing suggestions and medications to the individual using the application. This will help predict as well as increase the individual's life expectancy.

The scalability and flexibility of the application can also be improved with advancement in technology and availability of new and improved resources. Also, with the growth in Artificial Neural networks and Deep learning, one can integrate that with our existing application. With the help of Convolutional Neural networks and Computer vision, we can also try to take into account the physical health and appearance of a person.

Mental health can also be taken into account while predicting life expectancy with the help of sentiment analysis systems as well.

Bibliography

- https://theconversation.com/dont-die-wondering-apps-may-soon-be-able-to-predict- your-life-expectancy-but-do-you-want-to-know-129068
- https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service
- https://www.ibm.com/watson/products-services
- https://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa69-9fab-40ee-8457-ea0e5df8c8de

Appendix

12.1. Source Code

Services Used:

- **Watson Assistant**
- **Watson Studio**
- **IBM Cloud Function**
- **Node-Red**

Python Notebook:

Multiple Linear Regression

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns; sns.set(style="ticks", color_codes=True)
import scipy.stats as stats
import statsmodels.api as sm

# If you are reading an Excel file into a pandas DataFrame, replace `read_csv` by
read_excel` in the next statement.
dataset= pd.read_csv(body)
dataset.head()
# Data Preprocessing
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 21].values
dataset.head()
dataset.info()
dataset.describe()
dataset.columns
sns.heatmap(dataset.isnull(),yticklabels=False,cbar=False,cmap='viridis')
dataset.corr()
```

Taking care of missing data

```
from sklearn.impute import SimpleImputer #sklearn is library
imputer = SimpleImputer(missing_values=np.nan, strategy='mean') #nan is not a number
imputer = imputer.fit(dataset.iloc[:, 3:]) # fit calculates mean
dataset.iloc[:, 3:] = imputer.transform(dataset.iloc[:, 3:]) #transform means to apply
# Encoding categorical data i.e. converting categorical to numerical
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X = LabelEncoder()
dataset.iloc[:, :1] = labelencoder_X.fit_transform(dataset.iloc[:, :1])
# Encoding categorical data i.e. converting categorical to numerical
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X = LabelEncoder()
dataset.iloc[:, 1] = labelencoder_X.fit_transform(dataset.iloc[:, 1])
# Encoding categorical data ie converting categorical to numerical
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

```
dataset.iloc[:, :1]
dataset.iloc[:, 1]
dataset.iloc[:, 2]
dataset
X= dataset.iloc[:, :-1].values
y = dataset.iloc[:, 21].values
X
Y
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)
# Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
# Predicting the Test set results
y_pred = regressor.predict(X_test)
y_pred
y_test
```

K-Fold cross validation

```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=regressor, X=X_train, y=y_train, cv=10)
accuracies.mean()
accuracies.std()
```

The mean squared error

```
from sklearn.metrics import mean_squared_error, r2_score
print("Mean squared error: {}".format(mean_squared_error(y_test, y_pred)))
print("Variance score: {}".format(r2_score(y_test, y_pred)))
```

Predictions from our Model

```
predictions = regressor.predict(X_test)
plt.scatter(y_test,predictions)
```

Residual Histogram

```
sns.distplot((y_test-predictions),bins=50);
!pip install watson-machine-learning-client
from watson_machine_learning_client import WatsonMachineLearningAPIClient
client = WatsonMachineLearningAPIClient( wml_credentials )
model_artifact =client.repository.store_model(regressor, 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
```

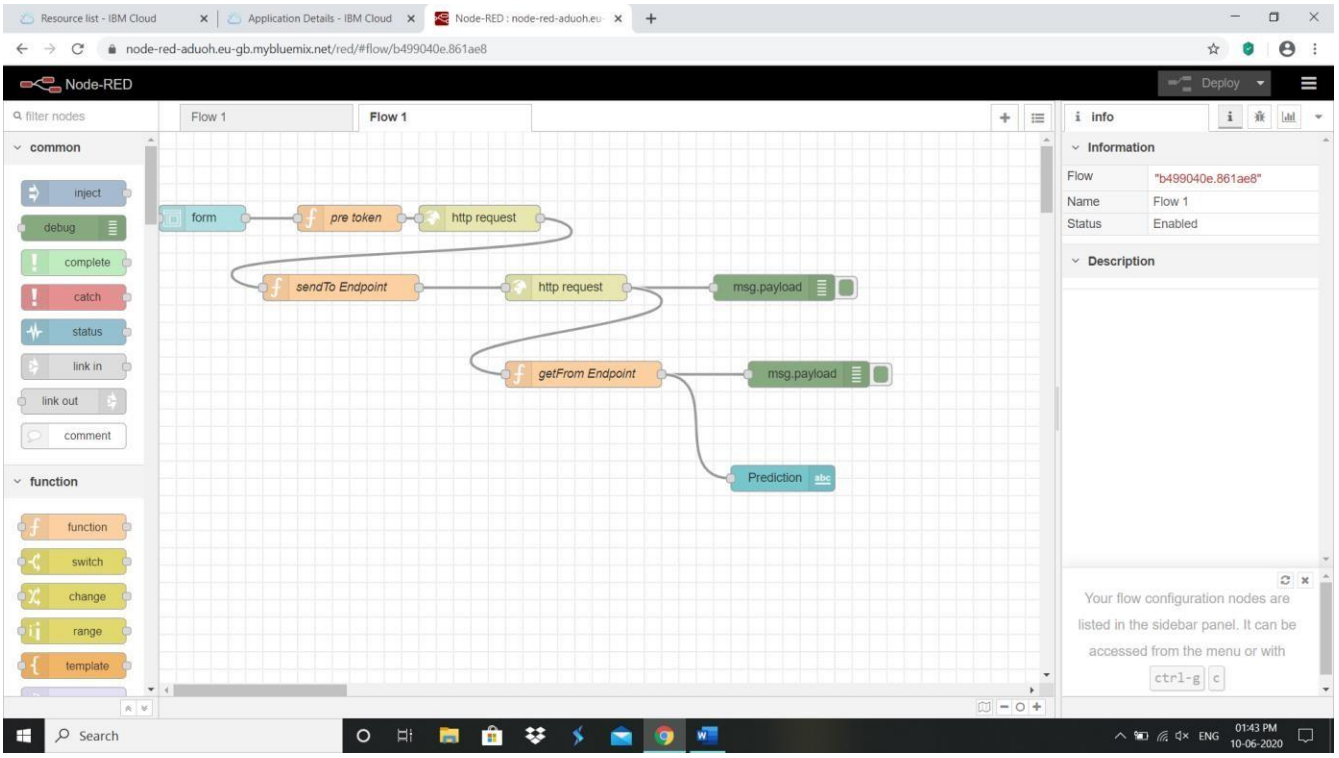
Model Building and Deployment

```
  wml_credentials={                                    #see below instruction for credentials "apikey":

"*****************",
  "instance_id": "*****************",
    "url": "*****************"
}
client = WatsonMachineLearningAPIClient(wml_credentials)
print(client.service_instance.get_url())
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME:"Tushar Jindal",

                        client.repository.ModelMetaNames.AUTHOR_EMAIL:
"tushar18csu218@ncuindia.edu",
                        client.repository.ModelMetaNames.NAME:
"LifeExpectancyPrediction"}
#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="Predicting
Life Expectancy using Machine Learning")
#GET SCORING END-POINT URL
scoring_endpoint = client.deployments.get_scoring_url(create_deployment) print(scoring_endpoint)
```

Node Red Flow:



Instruction:

For service credentials, open ibm cloud account and get in credentials tabs of machine learning instance.
Path to get that:- Go To dashboard->Resource list->cloud Foundry services->machine learning ** ->service credentials->copy and paste
wml_credentials={
  "apikey": "**********",
  "instance_id": "**********",
  "url": "**********"
}

Node-Red code:

```
[
    {
        "id": "a733e1fa.bd6678",
        "type": "tab",
```

"label": "Flow
1", "disabled":
false, "info": ""
},
{
"id": "f945e4a0.31da68",
"type": "ui_form",
"z": "a733e1fa.bd6678",
"name": "",
"label": "",
"group":
"f60ef5b9.5856a",
"order": 0,
"width": 0,
"height": 0,
"options": [

{
"label": "Country",
"value": "a",
"type":
"number",
"required":
true, "rows":
null
}
{

"label": "BMI\t",
"value": "b",
"type":
"number",
"required":
true, "rows":

null
},
{
"label": "HIV/AIDS",
"value": "c",
"type":
"number",
"required":
true, "rows":
null
},
{
"label": "thinness 1-19
years", "value": "d",
"type":
"number",
"required":
true, "rows":

null

},
{
"label": "thinness 5-9
years", "value": "e",
"type": "number",
"required":
true, "rows":
null
},
{
"label": "Adult
Mortality", "value": "f",
"type":
"number",

"required":
true, "rows":
null
},
{
"label": "Alcohol",
"value": "g",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Diphtheria

", "value": "h",
"type":
"number",
"required":
true, "rows":
null
},
{
"label": "GDP",
"value": "i",
"type": "number", "required": true, "rows": null
},
{
"label": "Hepatitis
B", "value": "j",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Income composition of resources",

"value": "k",
"type":
"number",
"required":
true, "rows":
null
},
{
"label": "Measles ",
"value": "l",
"type": "number",
"required": true,
"rows": null
},

{
"label": "Polio",
"value": "m",
"type":
"number",
"required":
true, "rows":

null
},
{
"label":
"Population",
"value": "n",
"type":
"number",
"required":
true, "rows":
null
},

{
"label":
"Schooling",
"value": "o",
"type":
"number",
"required":
true, "rows":
null
},
{
"label": "Status",
"value": "p",
"type": "text",
"required":
true, "rows":

null
},
{
"label": "Total
expenditure", "value":
"q",

"type":
"number",
"required":
true, "rows":
null
},
{
"label": "Year",
"value": "r",
"type": "number",
"required": true,

"rows": null
},
{
"label": "infant
deaths", "value": "s",
"type": "number",
"required": true,
"rows": null
},
{
"label": "percentage
expenditure", "value": "t",
"type":
"number",
"required":
true, "rows":
null
},

{
"label": "under-five
deaths ", "value": "u",
"type":
"number",

"required":
true, "rows":
null
}
],

END