A
Project Report
on

Predicting Life Expectancy Using Machine Learning

21/05/2020-18/06/2020

Internship under:
The Smart Bridge

Name : VIDYASREE VANKAM
Email : r151701@rguktrkv.ac.in

Project Id : SPS_PRO_215
Internship Title: - Predicting Life Expectancy using Machine Learning SB39157

Category : Machine Learning

Table of Contents

1. Introduction

1.1 Overview:

In this project a web application is designed that integrates node red app with ibm cloud services that predicts the life expectancy of a person based on the features.

A typical Regression **Machine Learning** project leverages historical data to predict insights into the future. This problem statement is aimed at predicting **Life Expectancy rate of a country** given various features.

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.

The dataset is extracted from Kaggle ‘https://www.kaggle.com/kumarajarshi/life-expectancy-who’

PROJECT REQUIREMENTS:

Technical Requirements: IBM Cloud, IBM Watson, Node- RED

Hardware Requirements: Processor: i5 ; Speed:2.6 GHZ or more; Hard disk space: 30 GB or more ; Ram: minimum 4 GB or more ;

Software Requirements: Operating system: Linux(Ubuntu 18.04) or Windows 10 ; Browser: Google Chrome, Mozilla Firefox ; Terminal or Command Prompt; Python 3.7.4, Jupyter notebook anaconda navigator, numpy,scipy,matplotlib installed. Or Ibm Watson Machine Learning installed.

1.2 Purpose

Predicting life expectancy using Machine Learning based on the features given as input like BMI, Adult mortality etc.,

Life expectancy is a statistical measure of the average time an organism is expected to live, based on the year of its birth, its current age, and other demographic factors including gender.

Life expectancy is affected by many **factors** such as: socioeconomic status, including employment, income, education and economic wellbeing; the quality of the health system and the ability of people to access it; health behaviours such as tobacco and excessive alcohol consumption, poor nutrition and lack of exercise;

2. Literature Survey

2.1 Existing Problem

Many courses of disease that lead to death can be influenced by personal lifestyles. ‘Unhealthy’ behaviours impede successful, active aging. Based on a number of survey waves, informational gaps can be closed with regard to further increasing life expectancy and the growing percentage of older people.

Over the course of recent decades, chronic diseases, cardiovascular diseases and malignant neoplasms have gained increased significance as causes of death. As recent analyses show, many of these ailments are influenced by personal behaviours, living arrangements and environmental conditions and therefore are also frequently ‘avoidable’. Unhealthy behaviours also impede successful, active aging. To examine the present life situation and a change in living circumstances with their impacts on earlier, present and anticipated health.

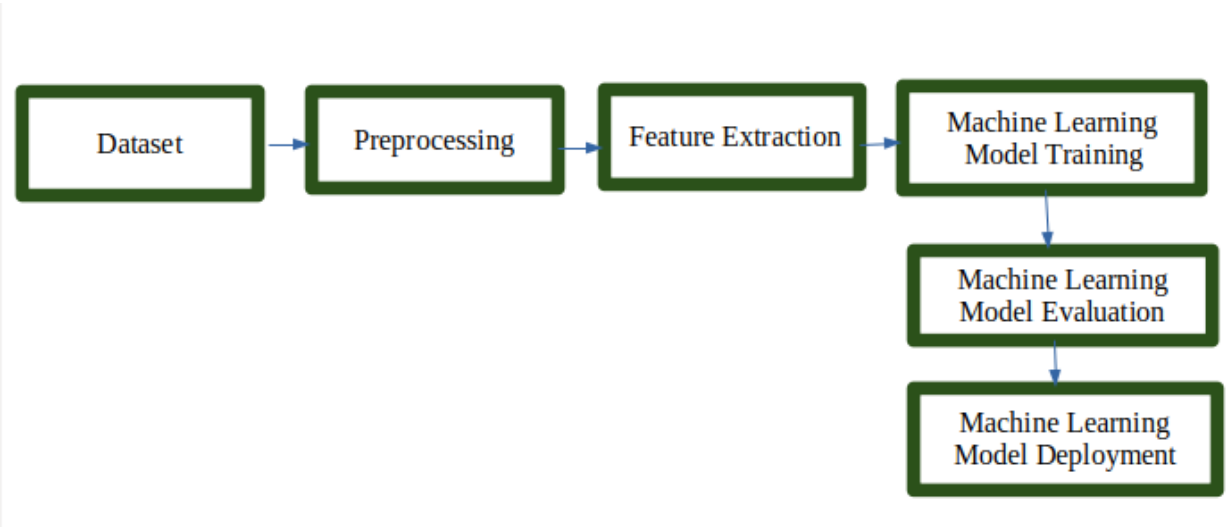
2.2 Proposed Solution

Context: 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. Hence, this gives motivation to resolve both the factors stated previously by formulating a regression model based on mixed effects model and multiple linear regression 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 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.

We use different features like BMI, Adult Mortality, Polio, Hepatitis, Income Expenditure, Infant_deaths etc., to find the prediction of Life Expectancy of a person.

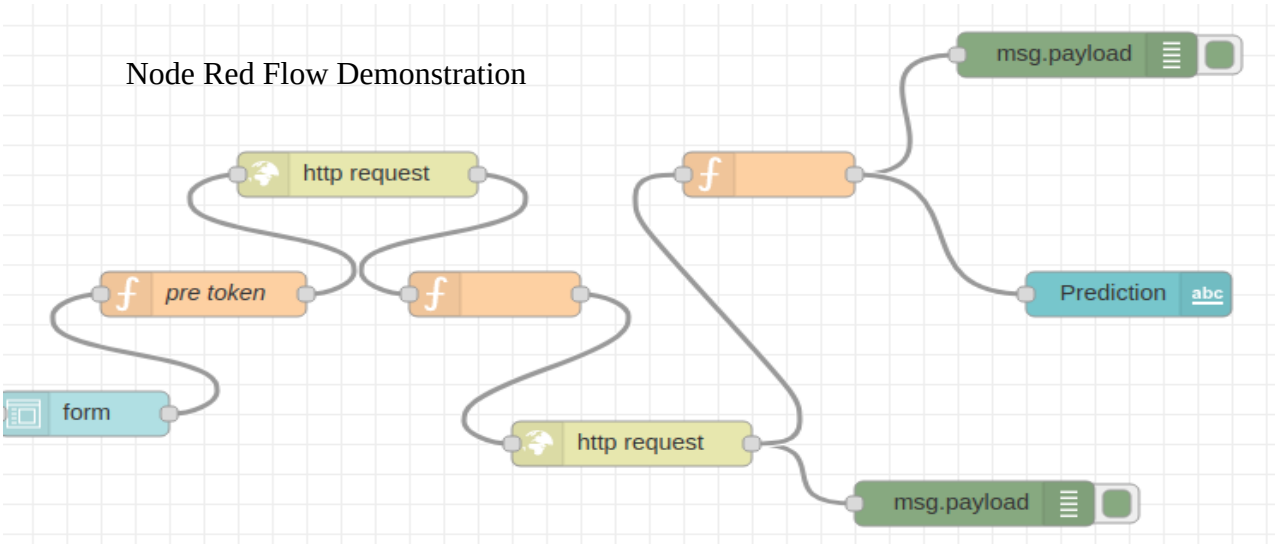
3.THEORITICAL ANALYSIS

3.1 Block Diagram

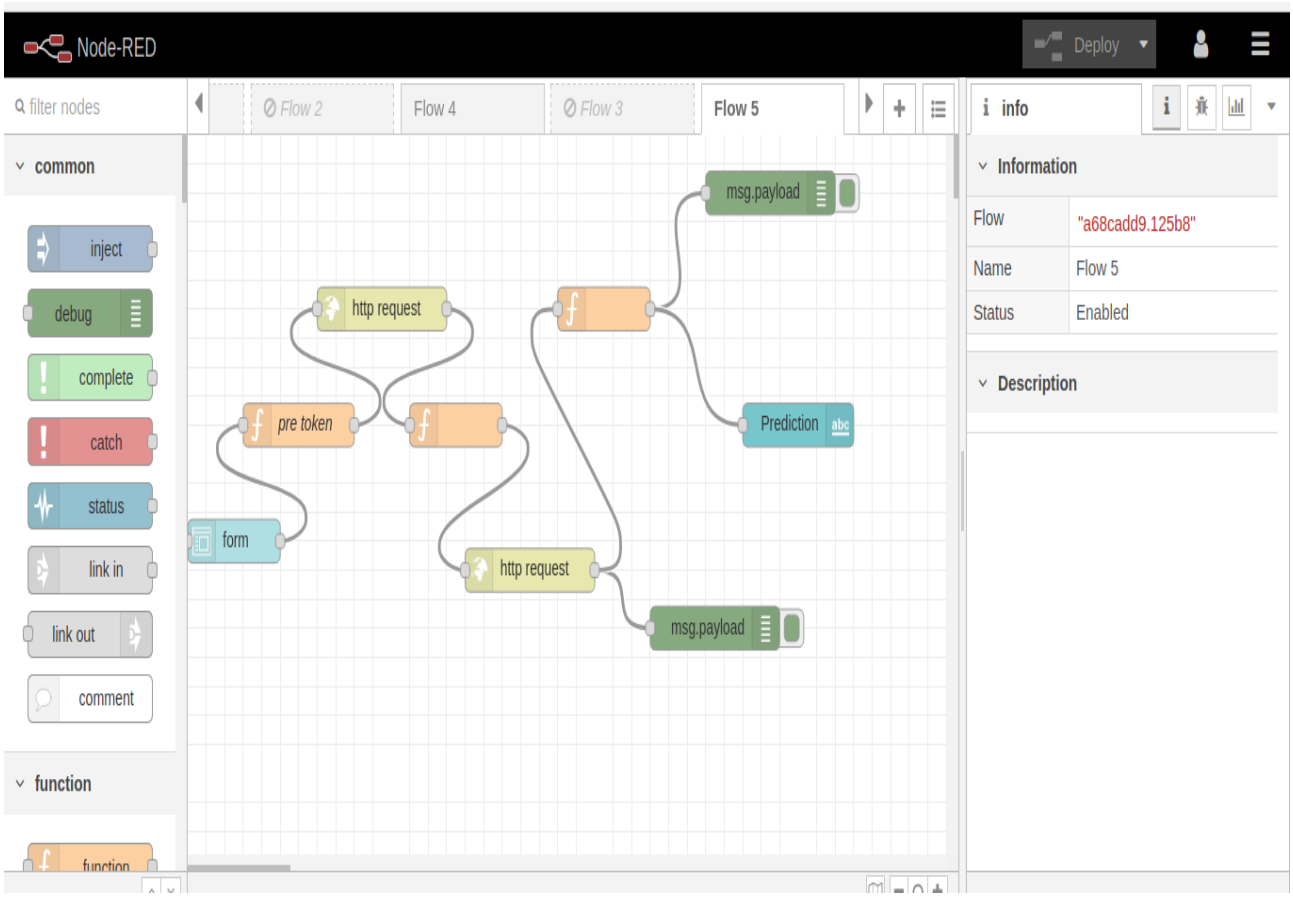


3.2 Hardware or Software Designing

- Create Necessary IBM Cloud Services
- Creating a Watson Studio Project
- Configuring Waston Studio
- Creating Machine Learning Service
- Creating / Importing Jupyter Notebook in IBM Watson and Importing Data
- Building a Machine Learning Model and Creating End Points for Node Red Integration
- Building Node Red Flow to Integrate ML Services



Node Red Flow Editor created to predict life expectancy



4. Experimental Investigations

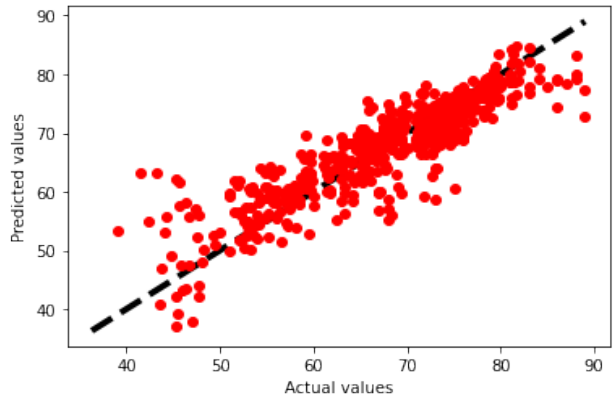
We consider the following factors to predict the life expectancy.

Following factors are taken into account for predicting the life expectancy of a country.

1. Country
2. Status: Developed or Developing status of the country.
3. Year
4. Adult mortality: Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population).
5. Infant deaths: Number of Infant Deaths per 1000 population.
6. Alcohol: Alcohol, recorded per capita (15+) consumption.
7. Percentage Expenditure: Expenditure on health as a percentage of Gross Domestic Product per capita(%).
8. Hepatitis B: Hepatitis B =immunization coverage among 1-year-olds (%).
9. Measles: Measles - number of reported cases per 1000 population.
10. BMI: Average Body Mass Index of entire population.
11. Under-five deaths: Number of under-five deaths per 1000 population.
12. Polio: Polio (Pol3) immunization coverage among 1-year-olds (%).
13. Total expenditure: General government expenditure on health as a percentage of total government expenditure (%).
14. Diphtheria: Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year olds (%).
15. HIV/AIDS: Deaths per 1000 live births HIV/AIDS (0-4 years).
16. GDP: Gross Domestic Product per capita (in USD).
17. Population: Population of the country.
18. Thinness 10-19 years: Prevalence of thinness among children and adolescents for Age 10 to 19(%).
19. Thinness 5-9 years: Prevalence of thinness among children for Age 5 to 9(%).
20. Income composition of resources: Human Development Index in terms of income composition of resources (index ranging from 0 to 1).
21. Schooling: Number of years of schooling.

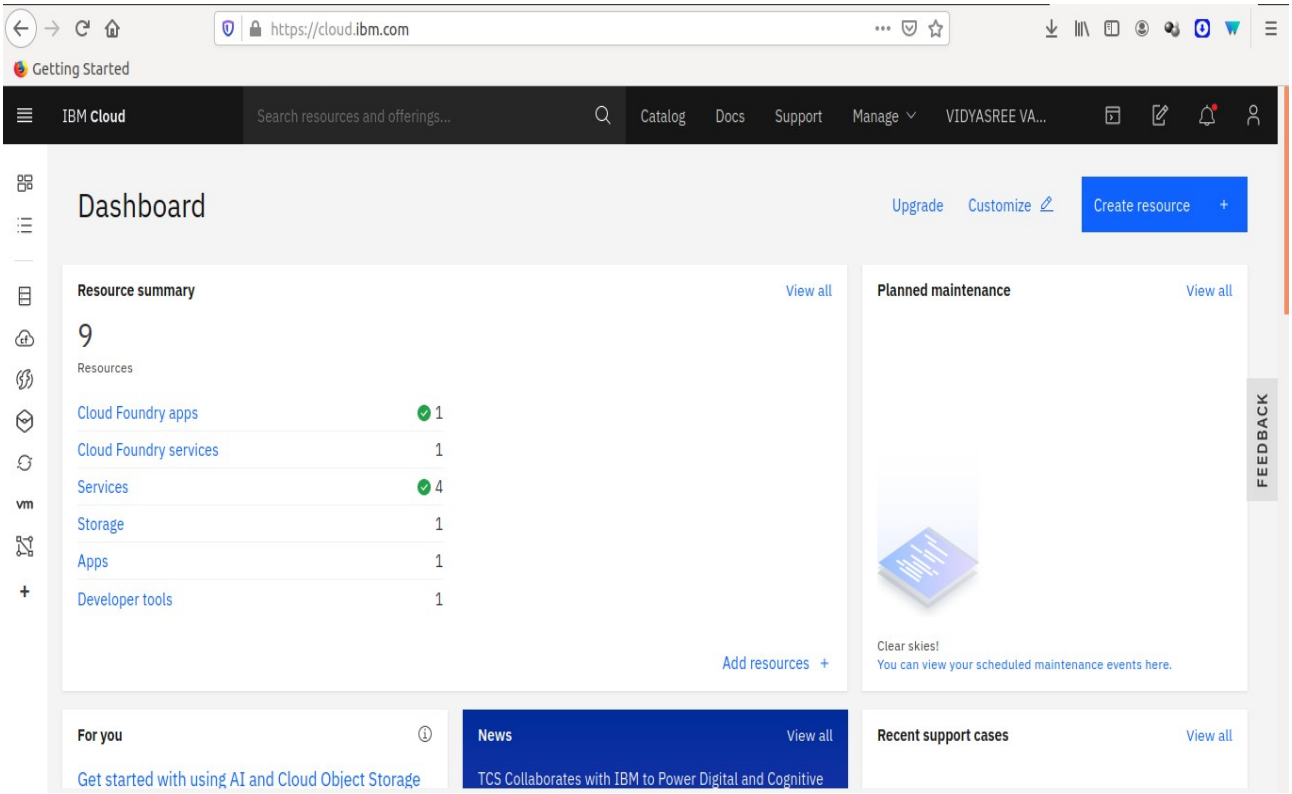
Linear Regression is used as the best suitable algorithm for the prediction.

```
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
model=linear_model.LinearRegression()
model.fit(X_train,y_train)
from sklearn import metrics
predictions = model.predict(X_test)
```

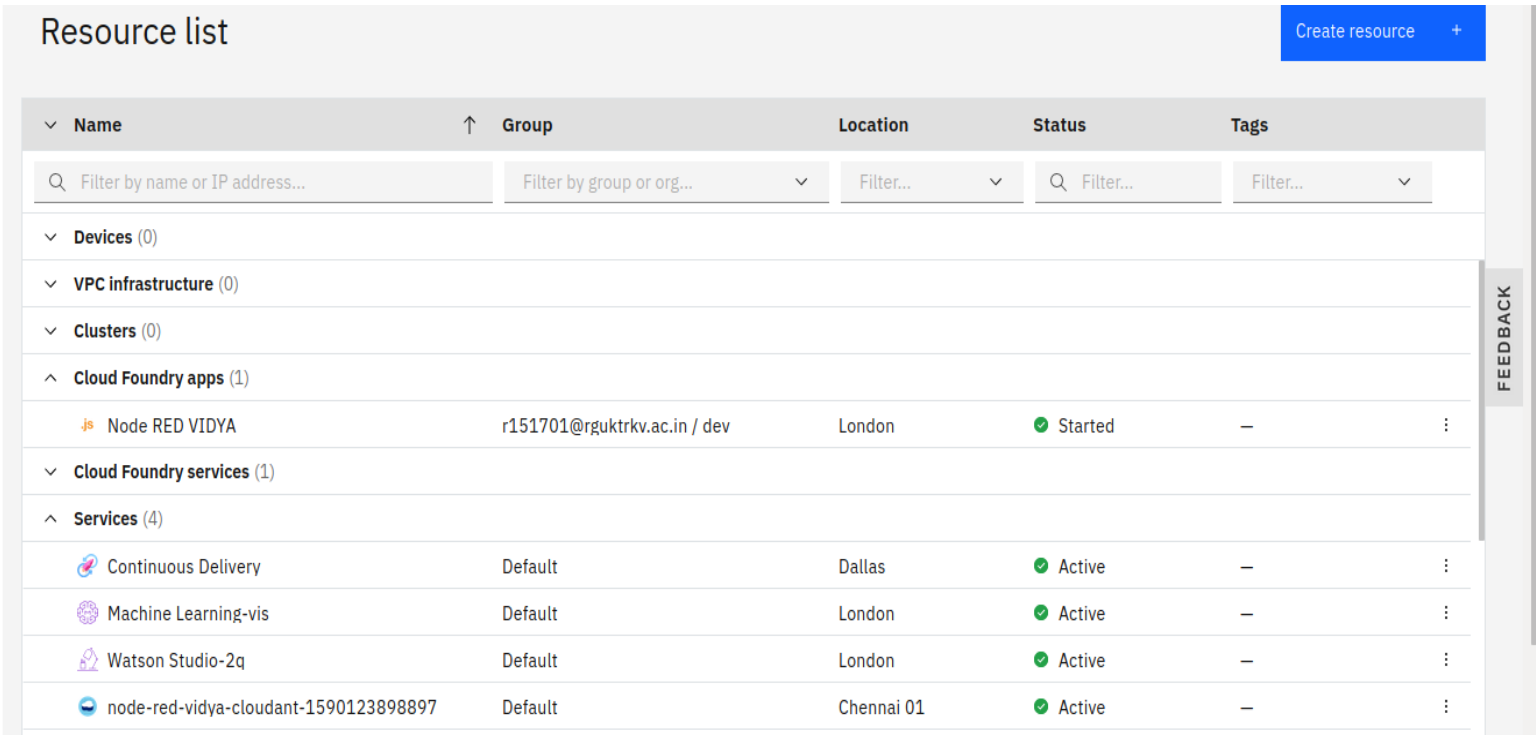


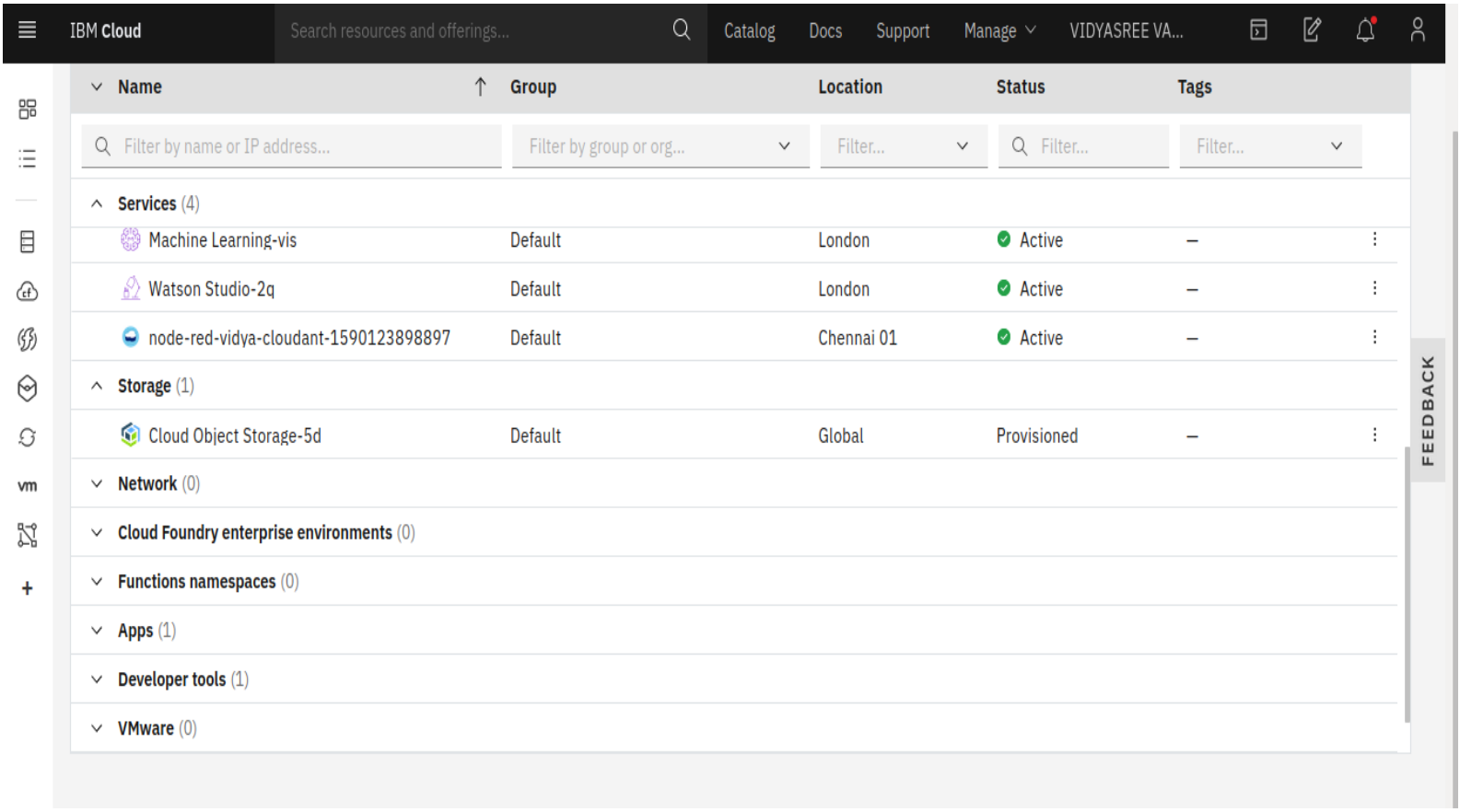
SCREENSHOTS

IBM CLOUD DASHBOARD

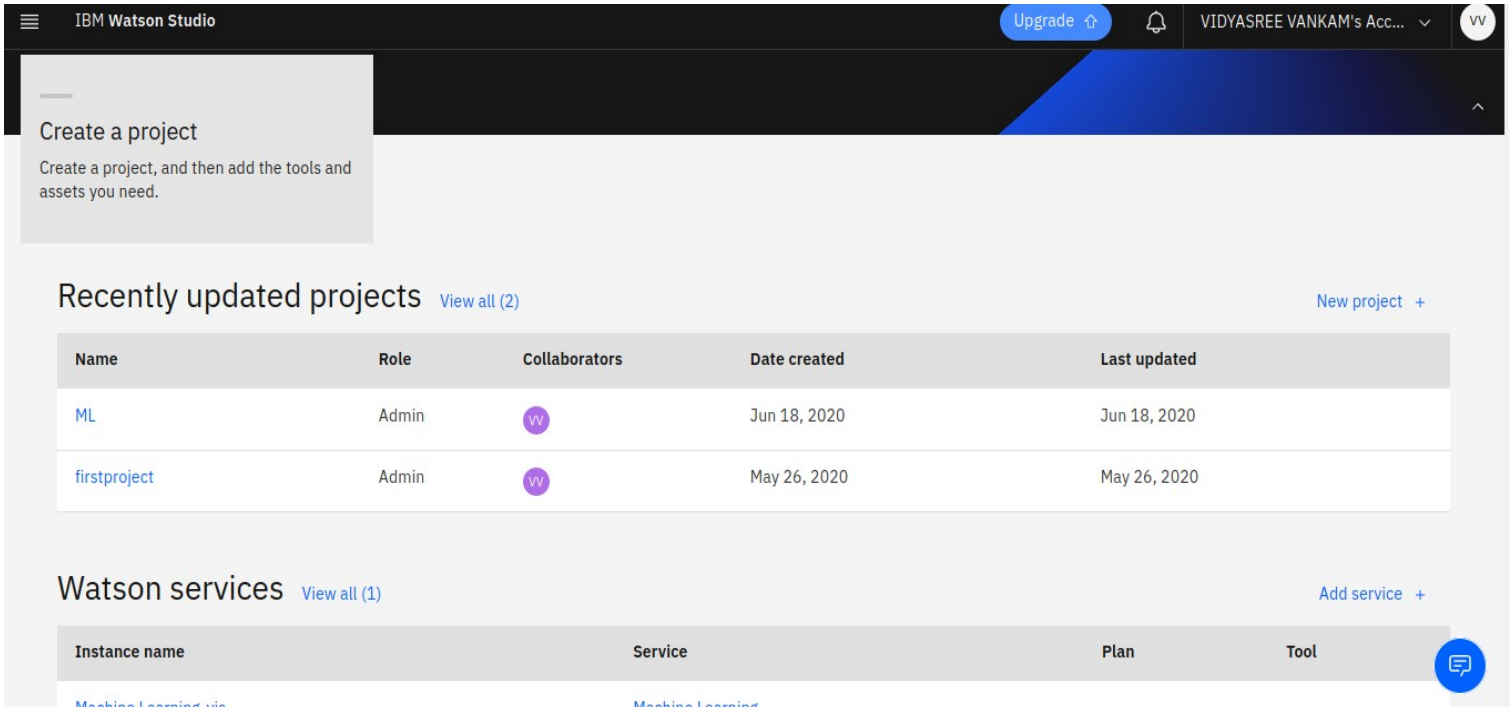


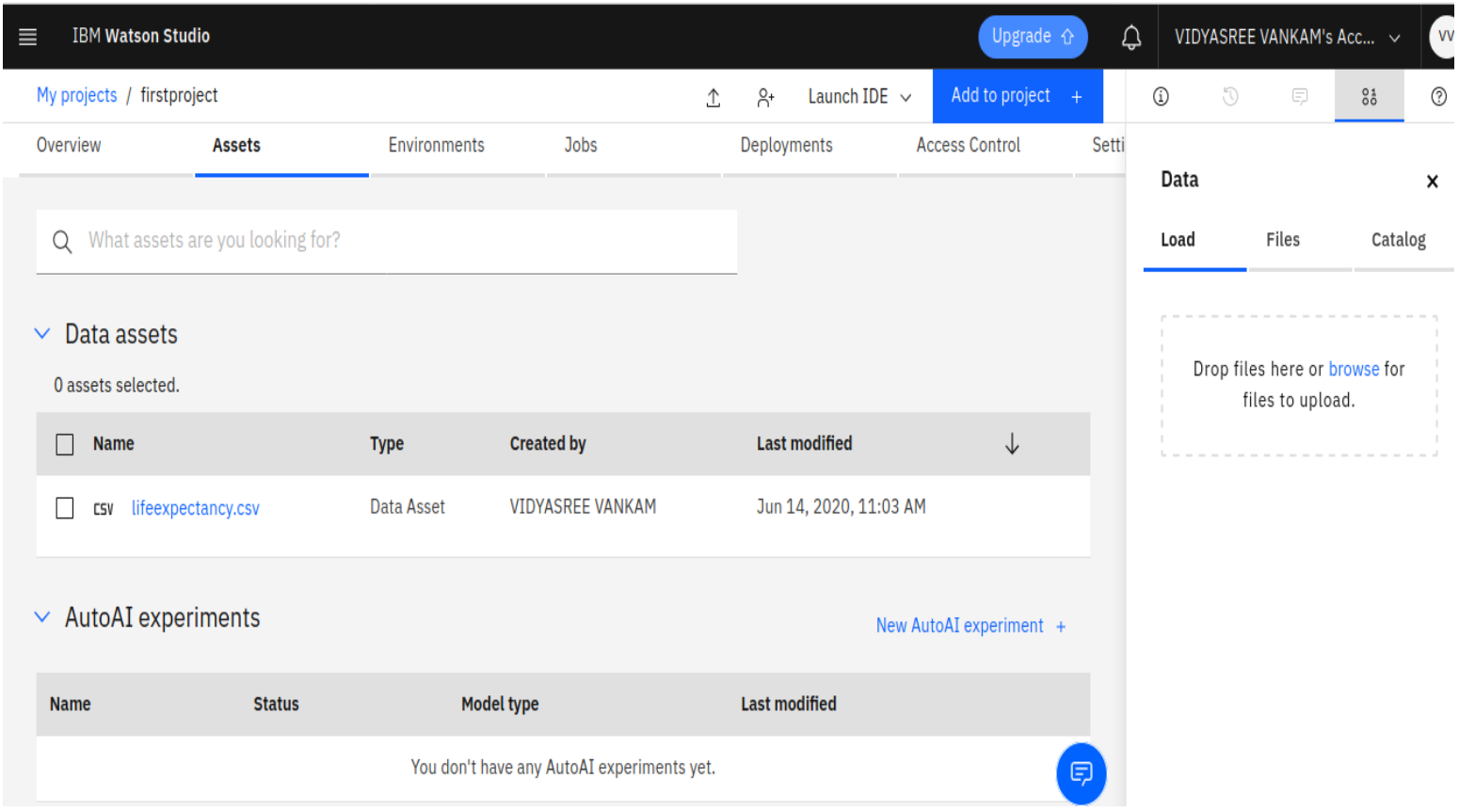
IBM RESOURCE LIST



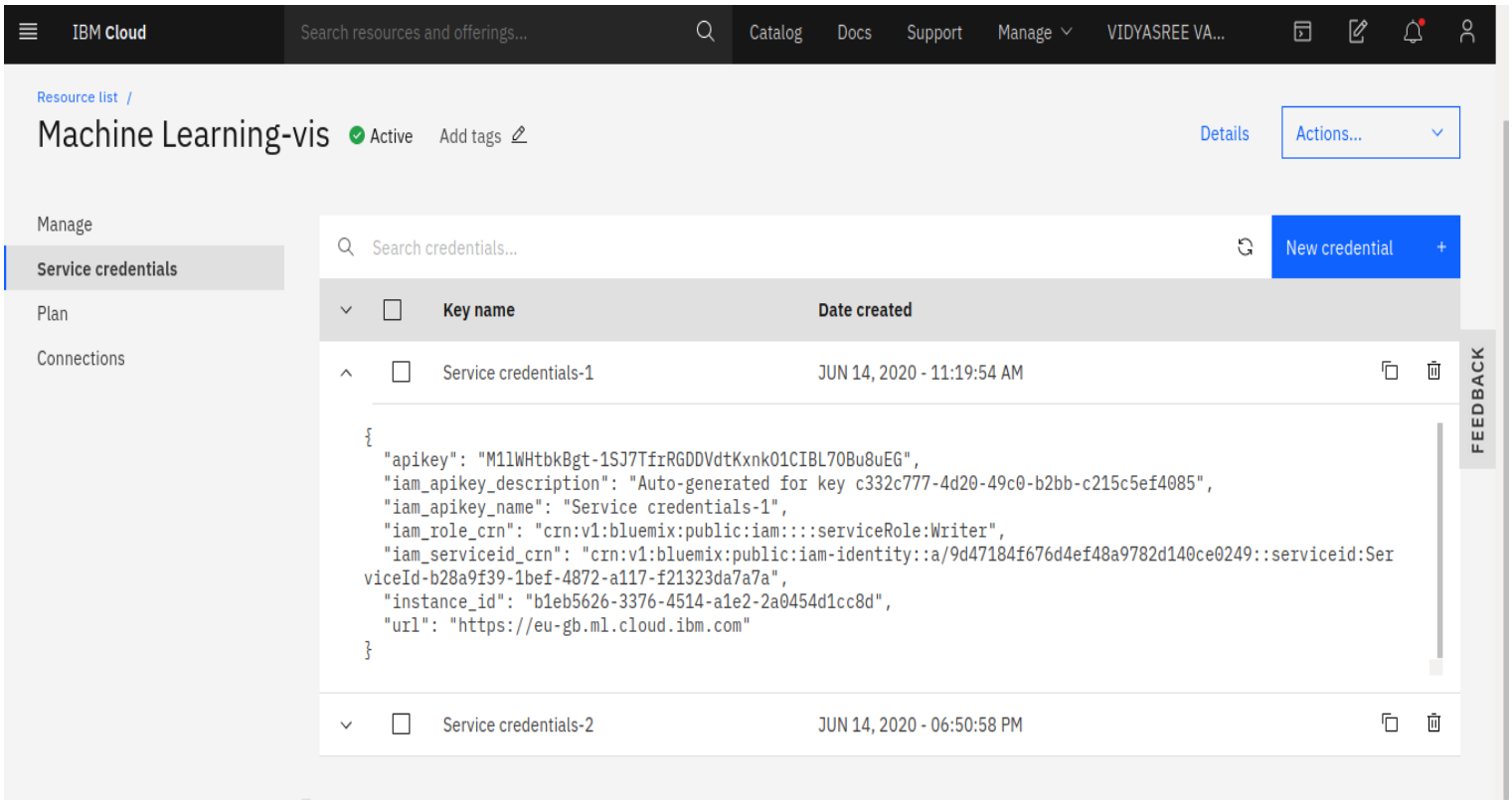


WATSON STUDIO

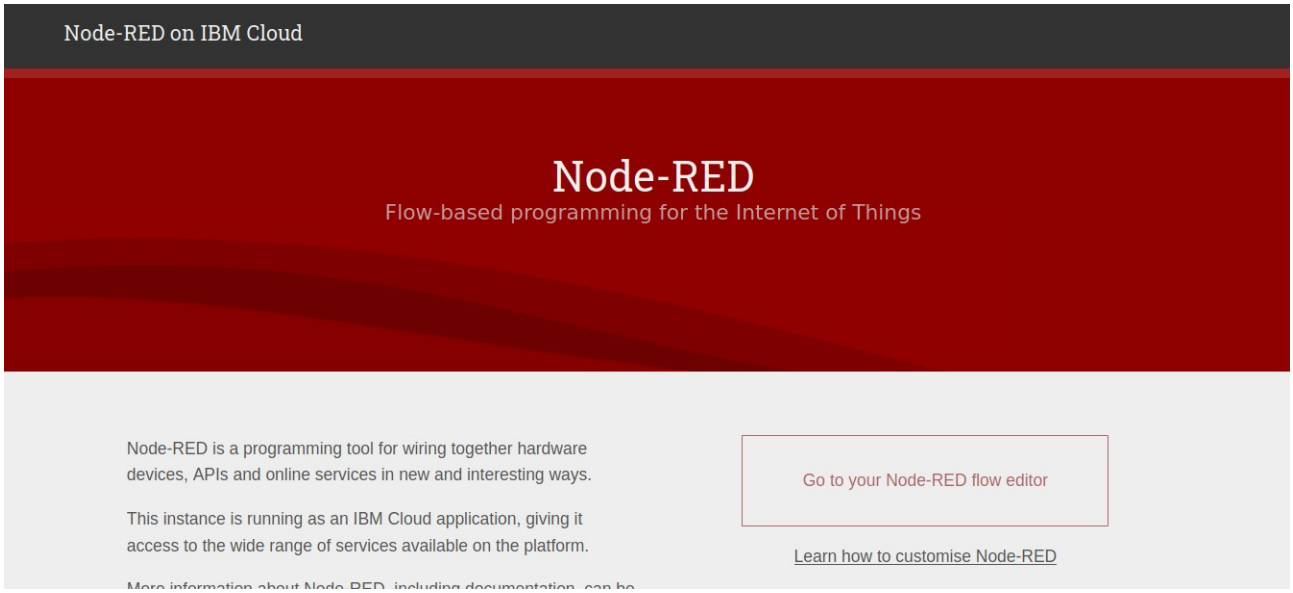




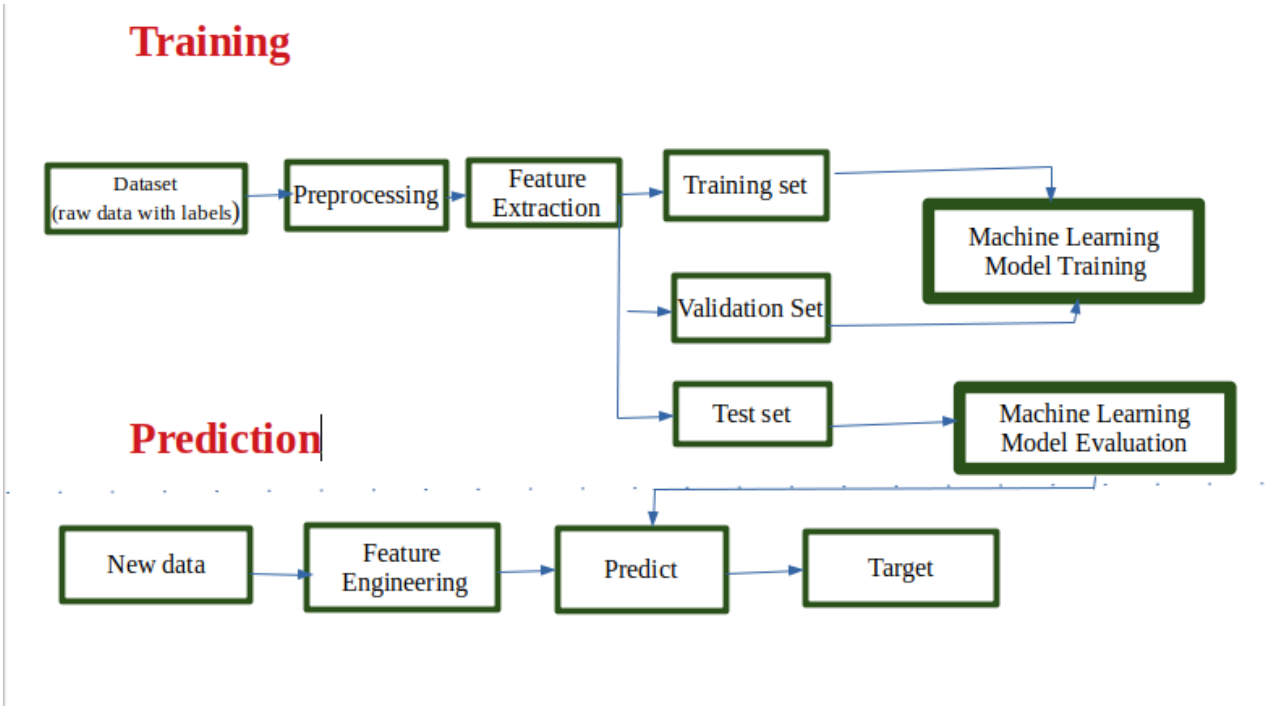
WATSON MACHINE LEARNING SERVICE



NODE RED



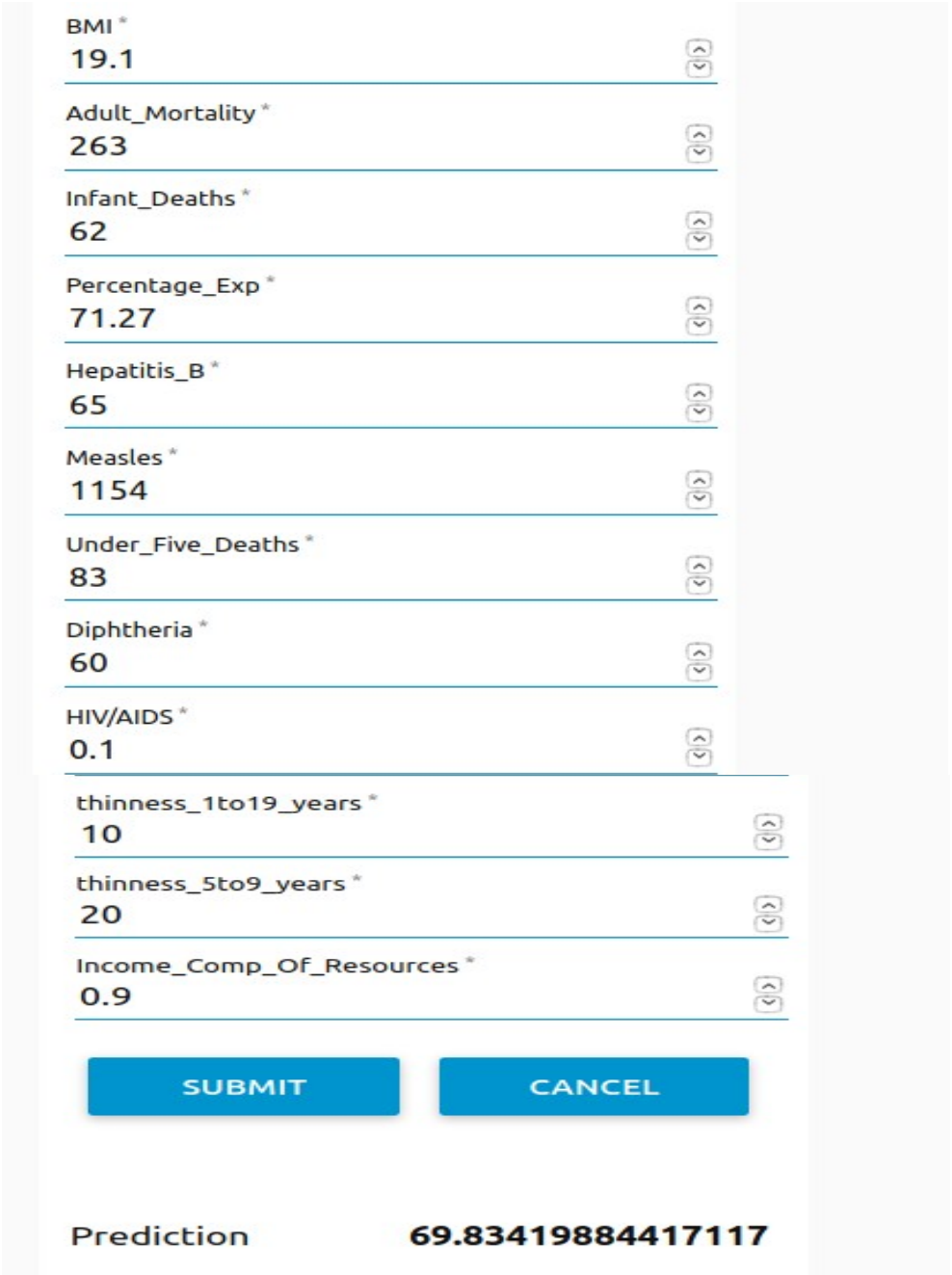
5. Flow Chart



6.Result

Node Red App integrates all the necessary components by using url:

https://node-red-vidya.eu-gb.mybluemix.net/ui/


BMI *
19.1
Adult_Mortality *
263
Infant_Deaths *
62
Percentage_Exp *
71.27
Hepatitis_B *
65
Measles *
1154
Under_Five_Deaths *
83
Diphtheria *
60
HIV/AIDS *
0.1
thinness_1to19_years *
10
thinness_5to9_years *
20
Income_Comp_Of_Resources *
0.9
SUBMIT
CANCEL
Prediction
69.83419884417117

7. Advantages and Disadvantages

Advantages:

Since the data is available to all in the website it is easy to predict the life expectancy country wise with simple Machine Learning algorithms.

We can also find out which factors are effecting the life of a person and can take certain measures.

With the help of Node red app integration we can easily predict the life expectancy just by entering values.

Disadvantages:

People with good understanding of data can handle this.

We need to deploy the model always in cloud which requires good internet connection

8. Applications

Life Expectancy of a person can be easily predicted.

Very helpful for the government to inspect various factors that effect the person's life.

One's country's health status can be tracked.

9. Future Scope

We can integrate NLP and speech to text to give voice input i.e., features as input by voice command and predict the output and then convert that text to speech. We can give input as speech by voice commands and output also we can hear the audio.

10.Conclusions

Life expectancy prediction can be done easily by integrating node red app with necessary things like IBM CLOUD Watson Studio, IBM Machine Learning. We can easily predict one's health with this app by entering the correct inputs i.e., features.

11. Bibilography

Ibm Academic Initiative Home

https://my15.digitalexperience.ibm.com/b73a5759-c6a6-4033-ab6b-d9d4f9a6d65b/dxsites/151914d1-03d2-48fe-97d9-d21166848e65/

IBM cloud login

https://cloud.ibm.com/login

Start building on the IBM Cloud. Build, deploy and scale apps for AI, IoT, data and mobile.

https://www.ibm.com/cloud/get-started

how to create a Node-RED starter application in the IBM Cloud, including a Cloudant database to store the application flow configuration.

https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

https://nodered.org/

https://github.com/watson-developer-cloud/node-red-labs

API:

https://www.youtube.com/watch?v=s7wmiS2mSXY&feature=youtu.be

Infuse AI into your applications with Watson AI to make more accurate predictions

https://www.ibm.com/watson/products-services

https://www.youtube.com/watch?v=W3iPbFTAAds&feature=youtu.be

Get an understanding of the principles of machine learning. Learn the different phases and tasks and get details on data transformation, model training, evaluation, and deployment.

https://developer.ibm.com/technologies/machine-learning/series/learning-path-machine-learning-for-developers/

https://www.youtube.com/watch?v=W3iPbFTAAds&feature=youtu.be

https://www.youtube.com/watch?v=NmdjtezQMSM

This is an introductory workshop for Watson Studio Cloud.

https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html

Auto AI – Evaluating model performance

https://developer.ibm.com/tutorials/watson-studio-auto-ai/

https://www.youtube.com/watch?v=IDKCmC1fCiU

Statistical Analysis on factors influencing Life Expectancy

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

IBM service

https://www.youtube.com/watch?v=DBRGlAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L

how to create both an empty project and a project based on a sample in IBM Watson Studio.https://www.youtube.com/watch?v=-CUi8GezG1I&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L&index=2

End point creation reference

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

Node red app

https://node-red-vidya.eu-gb.mybluemix.net/red/#flow/a68cadd9.125b8

Appendix

Source Code:

```
#!/usr/bin/env python
# coding: utf-8

# Import libraries
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt



# In[3]:
le = pd.read_csv('/home/apiiit-rkv/Desktop/kaggle/SMART INTERNZ/lifeexpectancy.csv',
delimiter=',')
le.dataframeName = 'led.csv'
# In[4]:
le.head(5)
# In[5]:
le.rename(columns={" BMI ":"BMI","Life expectancy ":"Life_Expectancy","Adult
Mortality":"Adult_Mortality",
                  "infant deaths":"Infant_Deaths","percentage expenditure":"Percentage_Exp","Hepatitis
B":"HepatitisB",
                 "Measles ":"Measles"," BMI ":"BMI","under-five deaths
":"Under_Five_Deaths","Diphtheria ":"Diphtheria",
                 " HIV/AIDS":"HIV/AIDS"," thinness  1-19 years":"thinness_1to19_years"," thinness 5-
9 years":"thinness_5to9_years","Income composition of
resources":"Income_Comp_Of_Resources",
                  "Total expenditure":"Tot_Exp"},inplace=True)
```

```
# In[6]:
le.describe()
# In[7]:
le.info()
# In[8]:
le.isnull().mean()*100
# In[9]:
country_list = le.Country.unique()
# In[10]:
le.isnull().mean()
# In[11]:
le.fillna(le.mean())
# In[12]:
le.isnull().sum()
# In[13]:
le.fillna(le.mean())
# In[14]:
country_list = le.Country.unique()
country_list
# In[15]:
country_list = le.Country.unique()
fill_list =
['Life_Expectancy','Adult_Mortality','Alcohol','HepatitisB','BMI','Polio','Tot_Exp','Diphtheria','GDP
','Population','thinness_1to19_years','thinness_5to9_years','Income_Comp_Of_Resources','Schoolin
g']
for country in country_list:
    le.loc[le['Country'] == country,fill_list] = le.loc[le['Country'] == country,fill_list].interpolate()
    # Drop remaining null values after interpolation.
le.dropna(inplace=True)
# In[16]:
```

```
col_dict =
{'Life_Expectancy':1,'Adult_Mortality':2,'Infant_Deaths':3,'Alcohol':4,'Percentage_Exp':5,'Hepatitis
B':6,'Measles':7,'BMI':8,'Under_Five_Deaths':9,'Polio':10,'Tot_Exp':11,'Diphtheria':12,'HIV/
AIDS':13,'GDP':14,'Population':15,'thinness_1to19_years':16,'thinness_5to9_years':17,'Income_Co
mp_Of_Resources':18,'Schooling':19}
# Detect outliers in each variable using box plots.
plt.figure(figsize=(20,30))
for variable,i in col_dict.items():
                plt.subplot(5,4,i)
                plt.boxplot(le[variable],whis=1.5)
                plt.title(variable)

plt.show()
# In[17]:
le.isnull().sum()
# In[18]:
X=le[['BMI','Adult_Mortality','Infant_Deaths','Percentage_Exp','HepatitisB','Measles','BMI','Under
_Five_Deaths','Diphtheria','HIV/
AIDS','thinness_1to19_years','thinness_5to9_years','Income_Comp_Of_Resources']]
#X=le[['BMI','Adult_Mortality']]
y=le['Life_Expectancy']
# In[19]:
from sklearn.model_selection import train_test_split
# In[20]:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
# In[21]:
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
# In[22]:
from sklearn.linear_model import LinearRegression
```

```
# In[23]:
from sklearn import linear_model
# In[24]:
model=linear_model.LinearRegression()
# In[25]:
model.fit(X_train,y_train)
# In[26]:
from sklearn import metrics
predictions = model.predict(X_test)
predictions
# In[27]:
X_test
# In[28]:
print('MAE',metrics.mean_absolute_error(y_test,predictions))
# In[29]:
print('MSE',metrics.mean_squared_error(y_test,predictions))
print('RMSE',np.sqrt(metrics.mean_squared_error(y_test,predictions)))
metrics.explained_variance_score(y_test,predictions)
# In[30]:
from sklearn.metrics import r2_score
# In[31]:
r2_score(y_test,predictions)
# In[39]:
import matplotlib.pyplot as plt
# In[38]:
fig,ax=plt.subplots()
ax.plot([y.min(),y.max()],[y.min(),y.max()],'k--',lw=4)
plt.plot(y_test, predictions, 'ro')
plt.xlabel('Actual values')
plt.ylabel('Predicted values')
plt.show()
# In[ ]:
```

```
get_ipython().system('pip install watson-machine-learning-client')
# In[ ]:
from watson_machine_learning_client import WatsonMachineLearningAPIClient
# In[ ]:
wml_credentials={
  "apikey": "WGdvuJ8NXNTXuLbiwUuNhjg0HxnRfXrKU6u-I5xJWJ77",
  "instance_id": "08b6a241-b518-4c54-9dff-fe8d45e8b956",
  "url": "https://us-south.ml.cloud.ibm.com"
}
# In[ ]:
client = WatsonMachineLearningAPIClient( wml_credentials )
# In[ ]:
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Vidyasree",
               client.repository.ModelMetaNames.AUTHOR_EMAIL: "r151701@rguktrkv.ac.in",
               client.repository.ModelMetaNames.NAME: "Life_Expectancy"}
# In[ ]:
model_artifact =client.repository.store_model(model, meta_props=model_props)
# In[ ]:
published_model_uid = client.repository.get_model_uid(model_artifact)
published_model_uid
# In[ ]:
deployment = client.deployments.create(published_model_uid, name="Life_Expectancy")
# In[ ]:
scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint
```

GITHUB LINK:

https://github.com/SmartPracticeschool/llSPS-INT-2121-Predicting-Life-Expectancy-using-Machine-Learning

PROJECT DEMONSTRATION LINK:

https://youtu.be/lfmrCxR6tA4

NODE RED APP LINK:

https://node-red-vidya.eu-gb.mybluemix.net/red/#flow/d15b7a21.d9c07

THANK YOU