Summer Internship Report

Name: Himanshu Nemade

Email: himanshu.nemade16@vit.edu

Title: Predicting Life Expectancy of a Country.

(21/5/2020 – 18/6/2020)

Domain: Machine Learning

1. INTRODUCTION

1.1. 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.

This project is to build a model while considering historical data from a period of 2000 to 2015 for all the countries. The model trained in this project will be able to predict the average lifetime of a human being given some input factors .With the help of this project any country is able to predict the expected lifetime of their countrymen and then accordingly take preventive measures to improve on their healthcare measures. This will also help countries in improving a particular field such as GDP ,alcohol intake,etc which have a high impact on a country's life expectancy.

Good prognostication helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more broadly: facilitates Advance Care Planning. So this problem statement is aimed at predicting Life Expectancy rate of a country given various features. It predicts the average lifetime of a human being and predicts on the basis of various factors like Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth and other demographic factors. So the end product will predict the future life expectancy of the person with the help of prior given appropriate matrix of features by the user like current 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.

Project Requirements: Python, IBM Cloud, IBM Watson

Functional Requirements: IBM cloud

Technical Requirements: ML, WATSON Studio, Python, Node-Red

Software Requirements: Watson Studio, Node-Red

Project Deliverables: Smartinternz Internship

Project Team: Himanshu Nemade

Project Duration: 23.5 Days

1.2. Purpose

The purpose of the project is to design a model for predicting Life Expectancy rate of a country given various features 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.

2. LITERATURE SURVEY

2.1. Proposed Problem

The typical regression model that can predict average life expectancy of the country based on some user inputted values such as GDP, BMI, HIV/AIDS, Year, Alcohol intake and etc.

2.2. Proposed Solution

Steps:

a) Create IBM cloud services
b) Configure Watson Studio
c) Create Node-Red Flow to connect all services together
d) Deploy and run Node-Red app

2.2.1. Create IBM cloud Services

- Watson Studio
- Machine Leaarning resource
- Node-Red

2.2.2. Configure Watson Studio

After creating all services, Go to resource list and launch watson studio then get started with watson studio. Then create an empty project and add machine learning resource as associated services in settings. Create a token as editor type.

Then add dataset and empty jupyter notebook into Assets.

After that go to notebook and write your code to build model and get the scoring endpoint url.

Steps for notebook:

- ✦ Install Watson_machine_learning_client
- ✦ Import necessary libraries
- ✦ Import dataset
- ✦ Data Preprocessing
  - Removing unusal species in column names using rename function.
  - Replacing nan values if any with their mean values.
- ✦ Exploratory Data Analysis
  - Plotting a heatmap to check if dimensional reduction can be performed
  - Plotting a pairplot for analysing pairwise relatonship among features.
- ✦ Train and Test :- The dataset was splitted into two parts i.e Input and Output. As Life Expectancy needs to be predicted so it is to be treated as output and all other columns are treated as Input.
  - Afterwards as we need regression technique to build our model so each and every column needs to be numeric . So then we check for numeric and categoric columns
  - Then we standardize the numeric and categric columns using pipelining.
  - At first independent pipelines for both the parts were designed then they were joined using columntransform
  - After that a regressor pipeline was designed using the regression technique.
  - So I have used ExtraTreesRegressor technique of sklearn.essemble as my regression algorithm because it best fits my dataset.
  - Then train and test split was peformed and 80% of dataset were trained data and 20% were test data.
  - Then dataset was fitted and predicted.
  - Then error and accuracy was estimated and the mean squared error is 2.4 whereas the R2_score or accuracy is 97.07%.

- ✦ Model Building and Deployment

  - At first the machine learning service credentials was stored in a variable and passed into WatsonMachineLearningAPIClient.

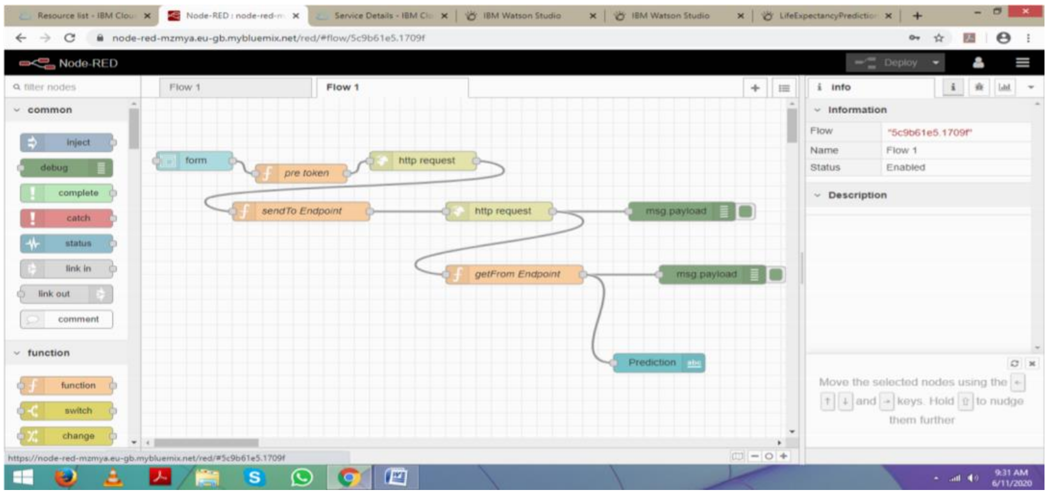
```
wml_credentials={
"apikey": "bolp9z-fIcxOwWd2OVgW_ME3LRehhTcjoN_4KWMGYIx3",
 "instance_id": "6ee18429-4d62-454e-b51e-fc9d4b9cec1e",
 "url": "https://eu-gb.ml.cloud.ibm.com"
}       client =
WatsonMachineLearningAPIClient(wml_credentials)
```

  - Then the model was build and stored in model_artifact.

  - Then the model was deployed and scoring_endpoint url was generated.

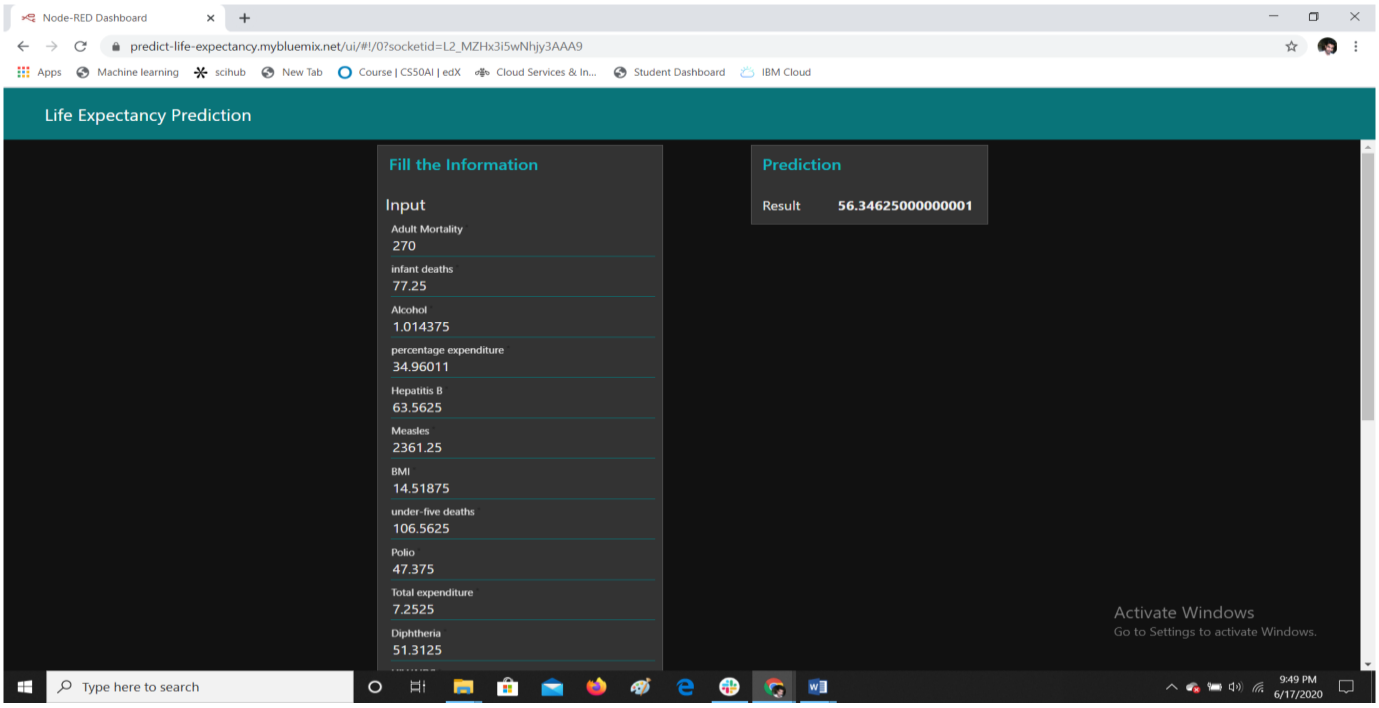
2.2.3. Create Node-Red Flow to connect all services together

- Go to Node-Red Editor from resource list.
- Install node-red Dashboard from manage pallete.
- Now create the flow with the help of following node.
  - Inject
  - Debug
  - Function
  - Ui_Form
  - Ui_Text



- Deploy and run Node Red app.

Deploy the Node Red flow. Then copy the link url upto .net/ and paste at a new tab by ui at the end of the url like this



predict-life-expectancy.mybluemix.net/ui/#!/0?socketid=L2_MZHx3i5wNhjy3AAA9

Life Expectancy Prediction

Fill the Information

Input

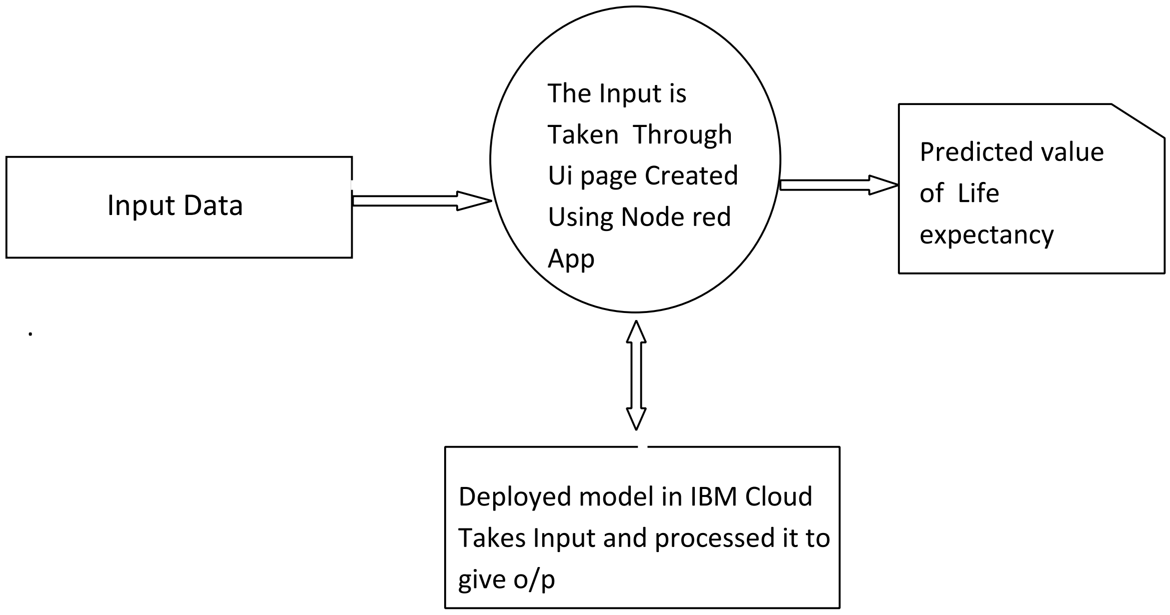
Adult Mortality	270
infant deaths	77.25
Alcohol	1.014375
percentage expenditure	34.96011
Hepatitis B	63.5625
Measles	2361.25
BMI	14.51875
under-five deaths	106.5625
Polio	47.375
Total expenditure	7.2525
Diphtheria	51.3125

Prediction

Result 56.34625000000001

3. THEORETICAL ANALYSIS

3.1. BLOCK DIAGRAM


Input Data
The Input is Taken Through Ui page Created Using Node red App
Predicted value of Life expectancy
Deployed model in IBM Cloud Takes Input and processed it to give o/p

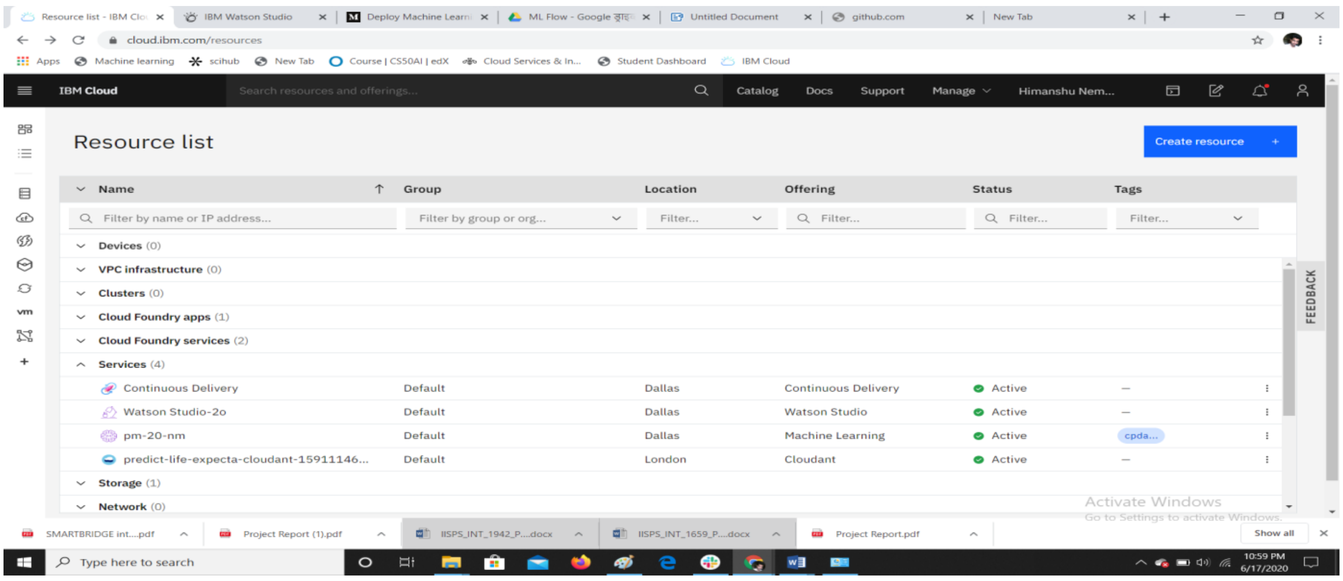
3.2. HARDWARE / SOFTWARE DESIGNING

○ **Project Requirements**: Python, IBM Cloud, IBM Watson ○ **Functional Requirements**: IBM cloud

○ **Technical Requirements**: ML, WATSON Studio, Python, Node-Red ○ **Software Requirements**: Watson Studio, Node-Red

4. EXPERIMENTAL INVESTIGATIONS

A) IBM Cloud Resource List



cloud.ibm.com/resources

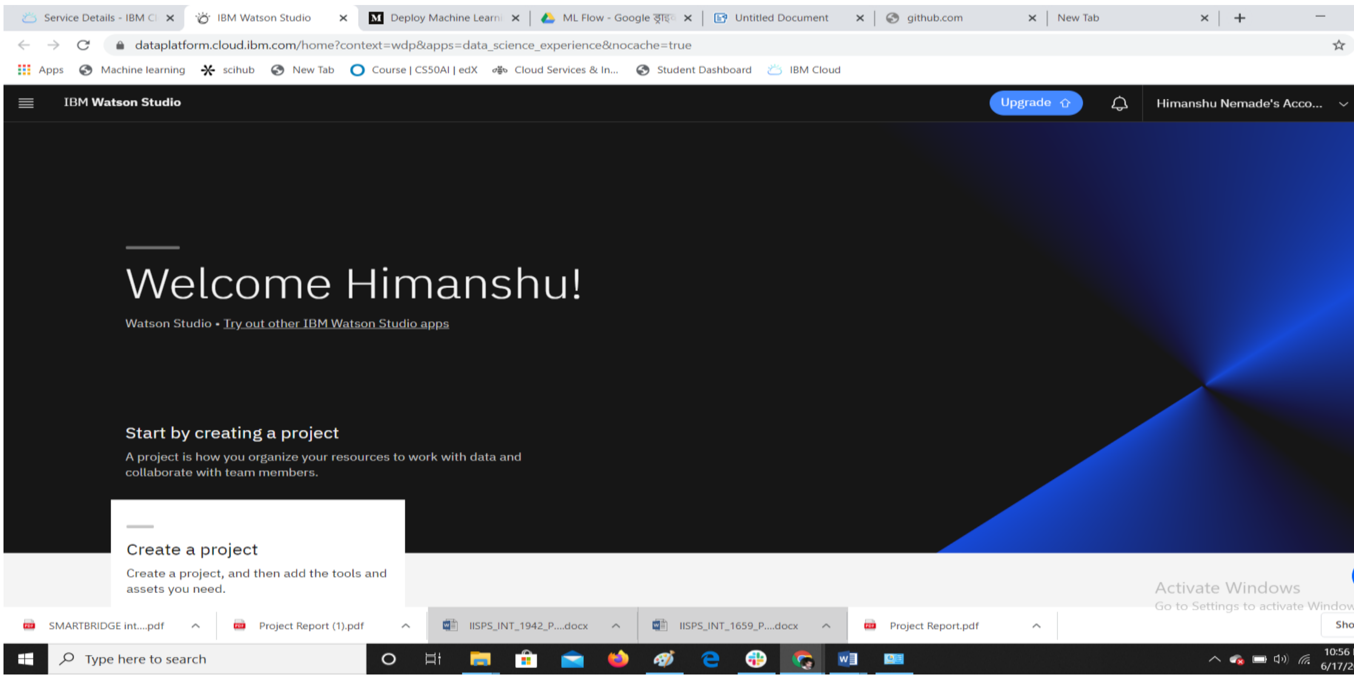
IBM Cloud

Resource list

Create resource +

Name	Group	Location	Offering	Status	Tags
Devices (0)					
VPC infrastructure (0)					
Clusters (0)					
Cloud Foundry apps (1)					
Cloud Foundry services (2)					
Services (4)					
Continuous Delivery	Default	Dallas	Continuous Delivery	Active	–
Watson Studio-2o	Default	Dallas	Watson Studio	Active	–
pm-20-nm	Default	Dallas	Machine Learning	Active	cpda...
predict-life-expecta-cloudant-15911146...	Default	London	Cloudant	Active	–
Storage (1)					
Network (0)					

B) IBM Watson Studio



dataplatform.cloud.ibm.com/home?context=wdp&apps=data_science_experience&nocache=true

IBM Watson Studio

Upgrade

Himanshu Nemade's Acco...

Welcome Himanshu!

Watson Studio • Try out other IBM Watson Studio apps

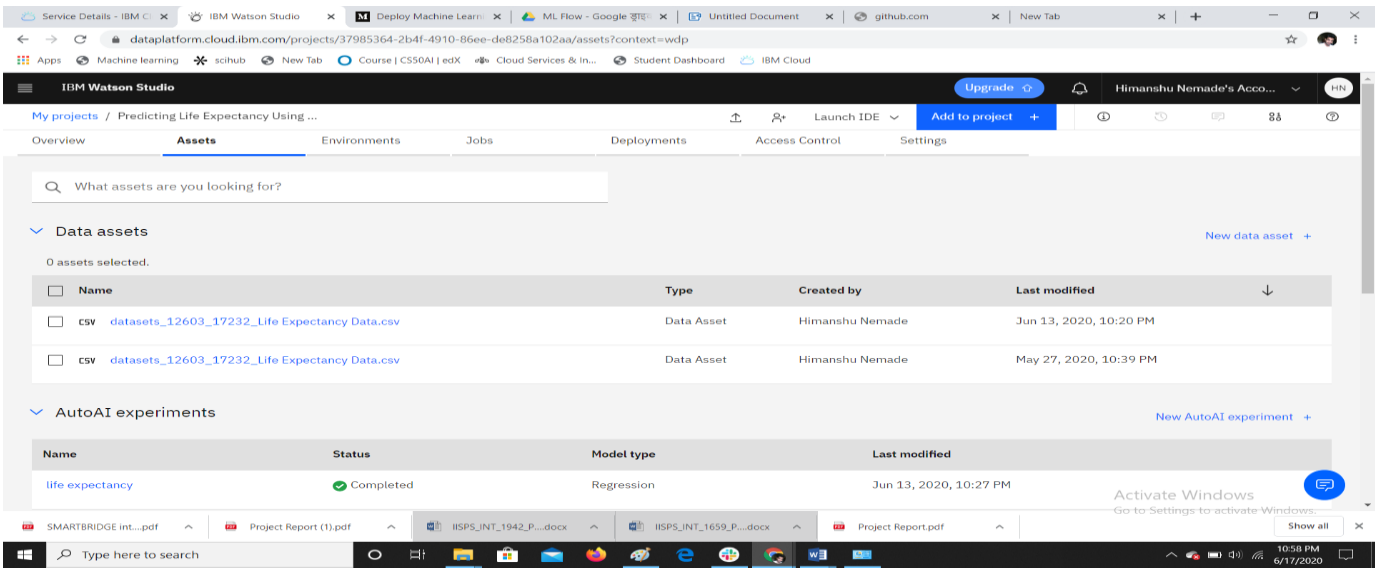
Start by creating a project

A project is how you organize your resources to work with data and collaborate with team members.

Create a project

Create a project, and then add the tools and assets you need.

C) IBM Cloud Project Details



IBM Watson Studio

My projects / Predicting Life Expectancy Using ...

Overview | Assets | Environments | Jobs | Deployments | Access Control | Settings

What assets are you looking for?

Data assets

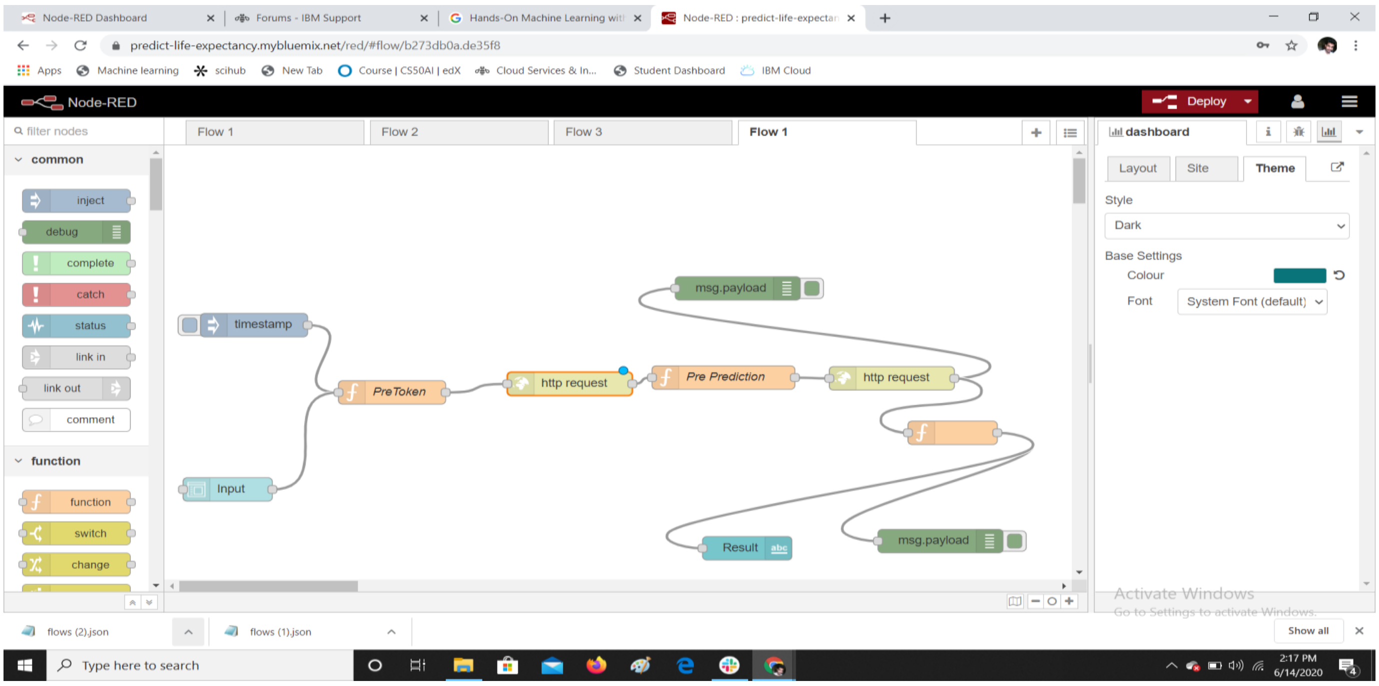
0 assets selected.

Name	Type	Created by	Last modified
CSV datasets_12603_17232_Life Expectancy Data.csv	Data Asset	Himanshu Nemade	Jun 13, 2020, 10:20 PM
CSV datasets_12603_17232_Life Expectancy Data.csv	Data Asset	Himanshu Nemade	May 27, 2020, 10:39 PM

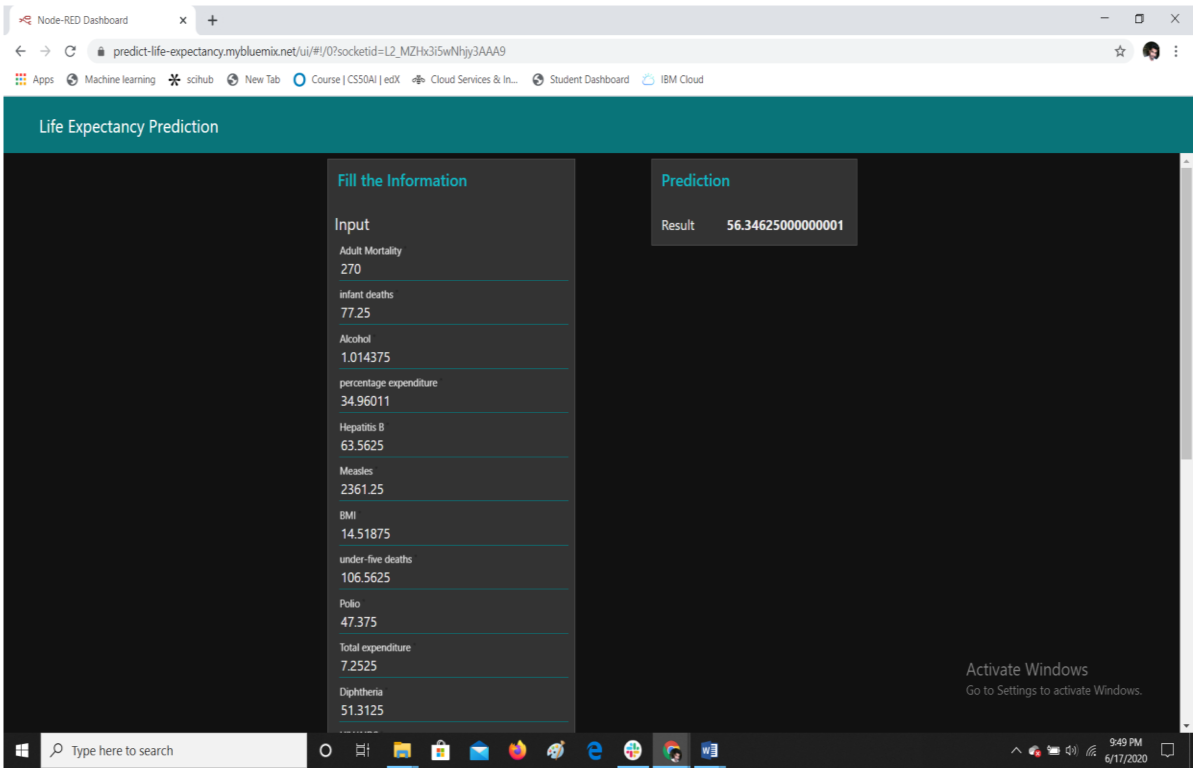
AutoAI experiments

Name	Status	Model type	Last modified
life expectancy	Completed	Regression	Jun 13, 2020, 10:27 PM

D)Node-Red Flow



E) Life Expectancy Prediction UI



Life Expectancy Prediction

Fill the Information

Input

Adult Mortality
270

infant deaths
77.25

Alcohol
1.014375

percentage expenditure
34.96011

Hepatitis B
63.5625

Measles
2361.25

BMI
14.51875

under-five deaths
106.5625

Polio
47.375

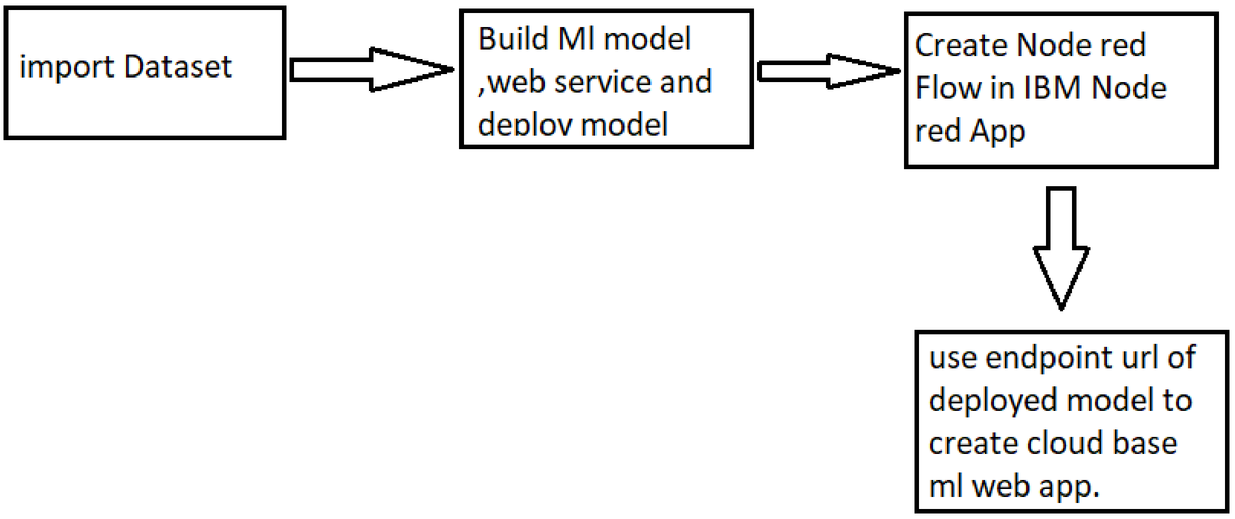
Total expenditure
7.2525

Diphtheria
51.3125

Prediction

Result 56.34625000000001

5. FLOWCHART



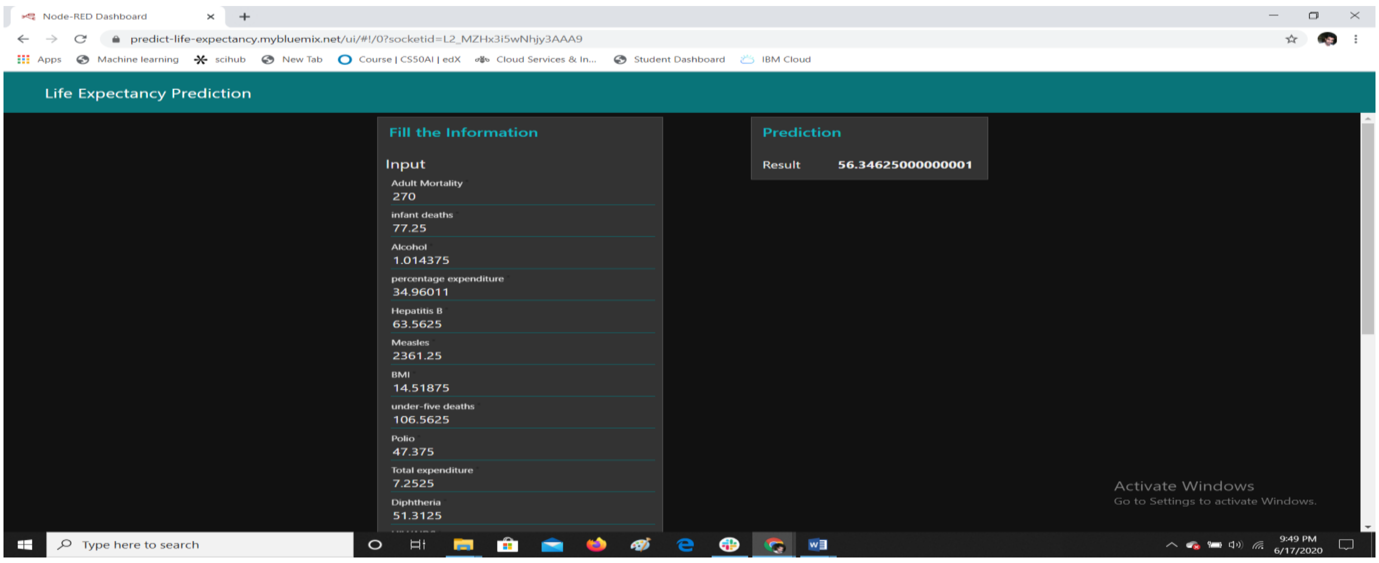
import Dataset
Build Ml model ,web service and deploy model
Create Node red Flow in IBM Node red App
use endpoint url of deployed model to create cloud base ml web app.

a) The user input all the required values in the app
b) The data then entered into watson and the scoring_endpoint url matches with the deployed model.

c) Then it enters into trained data and predict the life expetancy value
d) The value predicted is prompted in the app screen.

6. RESULT

This is the Life Expectancy UI.



Node-RED Dashboard

predict-life-expectancy.mybluemix.net/ui/#!/0?socketid=L2_MZHx3i5wNhjy3AAA9

Apps Machine learning scihub New Tab Course | CS50AI | edX Cloud Services & In... Student Dashboard IBM Cloud

Life Expectancy Prediction

Fill the Information

Input

Adult Mortality
270
infant deaths
77.25
Alcohol
1.014375
percentage expenditure
34.96011
Hepatitis B
63.5625
Measles
2361.25
BMI
14.51875
under-five deaths
106.5625
Polio
47.375
Total expenditure
7.2525
Diphtheria
51.3125

Prediction

Result 56.34625000000001

Activate Windows
Go to Settings to activate Windows.

Type here to search

9:49 PM
6/17/2020

7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

a) Health Inequalities: Life expectancy has been used nationally to monitor health inequalities of a country.
b) Reduced Costs: This is a simple webpage and can be accessed by any citizen of a country to calculate life expectancy of their country and doesnot required any kind of payment neither for designing nor for using.
c) User Friendly Interface: This interface requires no background knowledge of how to use it. It's a simple interface and only ask for required values and predict the output.

DISADVANTAGES:

a) Wrong Prediction: As it depends completely on user, so if user provides some wrong values then it will predict wrong value.
b) Average Prediction: The model predicts average or approximate value with 97.07% accuracy but not accurate value.

8. APPLICATION

a) It can be used to monitor health inequalities of a country.
b) It can be used to develop statistics for country development process.
c) It can be used to analyse the factors for high life expectancy.
d) It is user friendly and can be used by anyone.

9. CONCLUSION

This user interface will be useful for the user to predict life expectancy value of their own country or any other country based on some required details such as GDP, BMI, Year, Alcohol Intake, Total expenditure and etc.

10. FUTURE SCOPE

Future Scope of the Model can be:

a) Feature Reduction

It requires much more data about 21 columns to be known prior for predicting life expectancy which can be again difficult for a normal user to gather such datas so I have decided to do some kind of feature reduction or replacement of some features as individuals or groups to make it more user friendly.

b) Attractive UI

It is a simple webpage only asking inputs and predict output. In future I have decided to make it more user friendly by providing some useful information about the country in the webpage itself so that user does not need to do any kind of prior research for the values.

c) Integrating with services such as speech recognition

11. BIBLIOGRAPHY

- https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform
- 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
- https://www.youtube.com/embed/r7E1TJ1HtM0
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html
- https://www.kaggle.com/kumarajarshi/life-expectancy-who
- https://www.youtube.com/watch?v=DBRGlAHdj48&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L
- https://www.youtube.com/watch?v=-CUi8GezG1I&list=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L&index=2
- https://www.youtube.com/watch?v=Jtej3Y6uUng
- https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-webservice
- https://machinelearningmastery.com/columntransformer-for-numerical-and-categorical-data/

APPENDIX: Source Code

1)Notebook

```
#import librarires import
pandas as pd import
numpy as np import os
import matplotlib.pyplot as plt
import seaborn as sns import
warnings
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 from
sklearn.model_selection import train_test_split from
sklearn.ensemble import ExtraTreesRegressor from
sklearn.metrics import mean_squared_error, r2_score
from watson_machine_learning_client import WatsonMachineLearningAPIClient

#import dataset #Data
Preprocessing
df=df.rename(columns={‘old name:new name’})
df=df.fillna(df.mean()) #Exploratory
data Analysis df_kor=df.corr()
plt.figure(figsize=(10,10))
#heatmap
sns.heatmap(df_kor,vmin=-1,vmax=1,annot=True,linewidth=0.1)
#pairplot sns.pairplot(df)
#Train&Test
Y=df['Life expectancy']
X=df[df.columns.difference(['Life expectancy'])] categorical_features
= ['Country', 'Status'] categorical_feature_mask = X.dtypes==object
categorical_features = X.columns[categorical_feature_mask].tolist() 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() numeric_transformer
= Pipeline(steps=[
   ('imputer', SimpleImputer(strategy='median')),
   ('scaler', StandardScaler()),
])
random_forest_model = RandomForestRegressor()
random_forest_fit =
random_forest_model.fit(X_train, y_train)
random_forest_score =
cross_val_score(random_forest_fit, X_train, y_train,
cv = 5)
print("mean cross validation score: %.2f"%
np.mean(random_forest_score))
print("score without cv: %.2f"%
random_forest_fit.score(X_train, y_train))
print("R^2 score on the test data
%.2f"%r2_score(y_test,
random_forest_fit.predict(X_test)))
```

```
random_forest_model_predict =
random_forest_model.predict(X_test)

scoring = make_scorer(r2_score)
grid_cv = GridSearchCV(RandomForestRegressor(),
param_grid={'min_samples_split': range(2, 10)},
scoring=scoring, cv=5, refit=True)
grid_cv.fit(X_train, y_train)
grid_cv.best_params_

result = grid_cv.cv_results_
print("Best Parameters: " +
str(grid_cv.best_params_))
result = grid_cv.cv_results_
print("R^2 score on training data: %.2f"  %
grid_cv.best_estimator_.score(X_train, y_train))
print("R^2 score: %.2f"% r2_score(y_test,
grid_cv.best_estimator_.predict(X_test)))
print("Mean squared error: %.2f"
    % mean_squared_error(y_test,
random_forest_model_predict))
print("Mean absolute error: %.2f"
    % mean_absolute_error(y_test,
random_forest_model_predict))

#Model Building and Deployment  wml_credentials={
 "apikey": "********************************",
 "instance_id": "********************************",
  "url": "************************************"
}
client = WatsonMachineLearningAPIClient(wml_credentials)

model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "****",
client.repository.ModelMetaNames.AUTHOR_EMAIL: "**********",
client.repository.ModelMetaNames.NAME: "LifeExpectancy"}
model_artifact=client.repository.store_model(ExtraTreeRegressor, meta_props=model_props) model_uid
= client.repository.get_model_uid(model_artifact)
create_deployment = client.deployments.create(model_uid, name="LifeExpectancyPrediction")
scoring_endpoint = client.deployments.get_scoring_url(create_deployment) print(scoring_endpoint)
```

2) Flow.json

```
[{"id":" }]
```