Project on

# PREDICTION OF LIFE EXPECTANCY USING MACHINE LEARNING


Submitted by-

**Miss. Prayukta Awati**

Email- awatiprayukta@gmail.com


GitHub- https://github.com/PrayuktaAwati

UI link- https://node-red-asorj.eu-gb.mybluemix.net/ui/#!/0?socketid=O9b8BVRkeqJg1qSiAAAA

Demo Video link- https://drive.google.com/file/d/1YO9-xk-sJz4QFFi6sme3UidYtT9Ek7xa/view?usp=sharing

# Introduction-

## Overview-

The given problem statement is aimed at predicting Life Expectancy rate of a country, we have provided with various features like Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth, GDP, Education, Alcohol intake etc. From this given feature and problem statement we come to the conclusion that this is Regression Machine Learning project problem statement in which based on historical data we have to predict life expectancy of individual.

There are different requirements while implementation: we have following algorithm that can be used for regression- Linear Regression, Random Forest Algorithm Linear Regression with Polynomic features, Decision Tree Regression, etc. For model creation we have to follow following steps: data visualization, data preprocessing, data splitting, model building, check for accuracy and predicating new values.

## Purpose

Life expectancy will give brief information about different factors like quality of life, circumstances of country, economic condition of country, overall life style of people in country, growth rate, primary services, etc. and all this factors are directly effects the life expectancy of particular in that country. Higher the good atmosphere higher the life expectancy rate. So by applying all the development condition which can afford by country we can try to increase the life expectancy rate of country. Also to provide good health facilities to old age people, to ensure good quality life, to expect long and healthy life of individual we encourage to use new technology and this project will totally helpful in that aspect.

# Literature survey-

## Existing problem-

For now scope is to verify the accuracy of life expectancy prediction and validation of data quality, big data techniques and analysis algorithms need to be developed and tested in a real-life situation with several sample groups. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population. The dataset used for the prediction contains data from year 2000 to 2015. It contains more than 2500 entries and around 22 columns with various features like Population, Status, Alcohol, Infant Deaths etc., which aids the prediction of the model. This also includes to give answers to the questions like, What are the factors affecting life expectancy of a country? Do people in rich nations live longer?

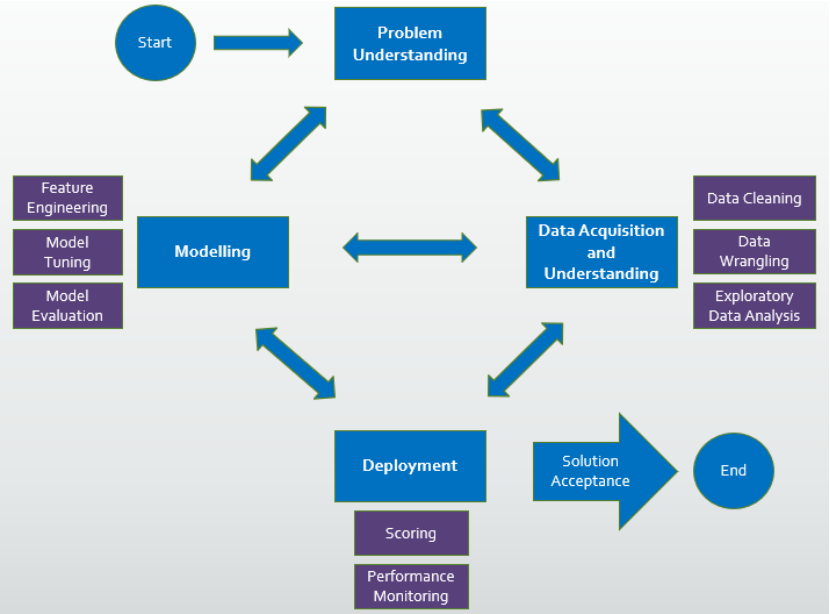
## Proposed solution-

The best way is always divide problem in small tasks, hence after applying this rule over here we conclude first task is to read dataset and preprocessing of data. After this data visualization is an important aspect to choose our algorithm and to create model. While considering data from a period of 2000 to 2015 for about 193 the countries with more than 2500 rows of data and 22 different features there seem to be a positive correlation between The Percentage of Healthcare Expenditure, Schooling, GDP and BMI and Life Expectancy, while there is a negative one between Adult Mortality, AIDS and Life Expectancy, there does not seem to have any correlation between Alcohol, under 5 years – old deaths and Life Expectancy. Considering all this aspect we will choose more accuracy dataset and will implement the model.

To build the model of Predicting Life Expectancy using Machine Learning uses IBM Cloud services. As we use IBM cloud services we can easily build and deploy our model and also can easily integrate it with UI too.

# Theoretical Analysis-

## Block diagram-




## Hardware/software designing-

Software Requirements-

IBM cloud: Watson assistant, Machine learning service, Node-Red Service; Excel.

Functional Requirement-

There are different requirements while implementation: we have following algorithm that can be used for regression- Linear Regression, Random Forest Algorithm Linear Regression with Polynomic features, Decision Tree Regression, etc. For this different steps are data visualization, data preprocessing, data splitting, model building, check for accuracy and predicating new values.

# Experimental investigation-

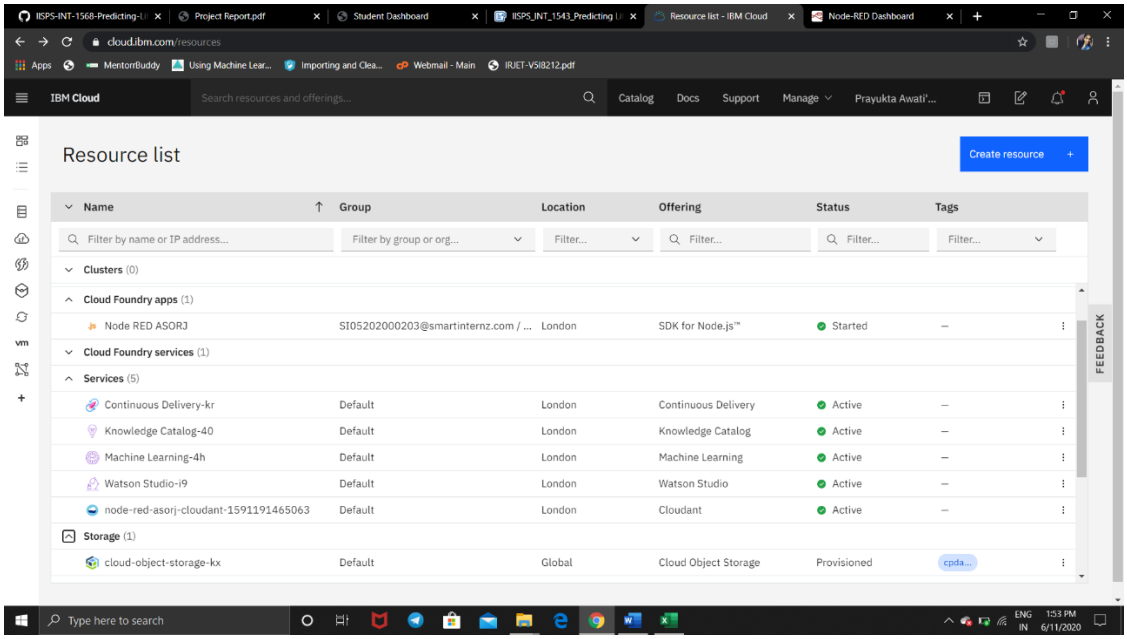



Fig.: Resource list

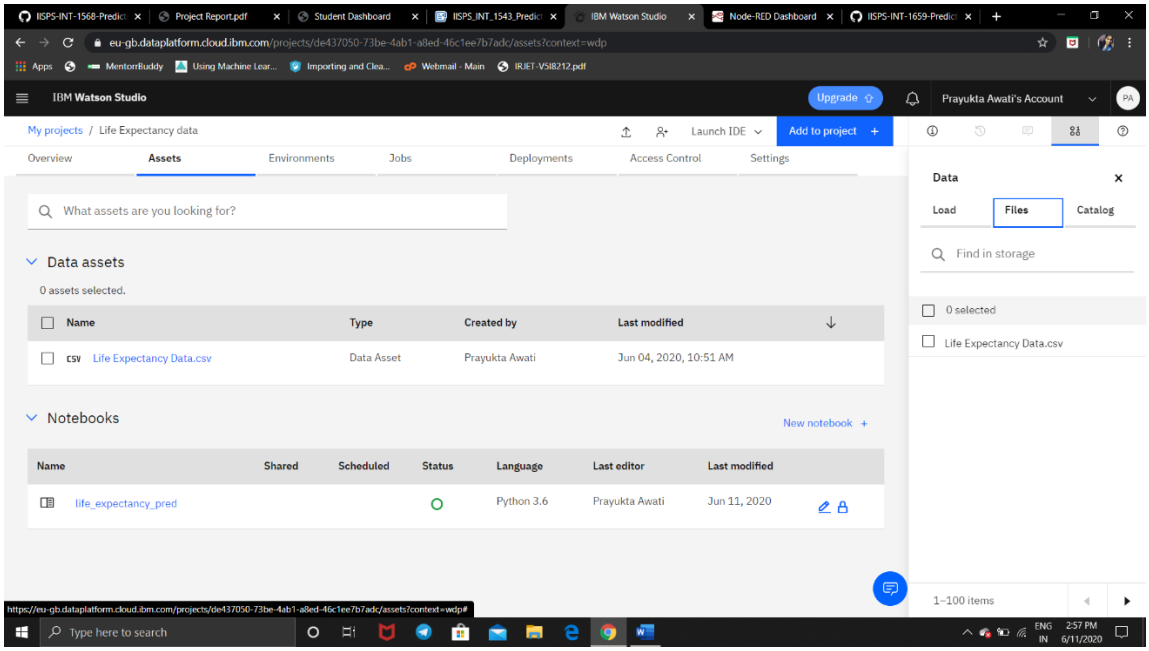



Fig.: Watson studio

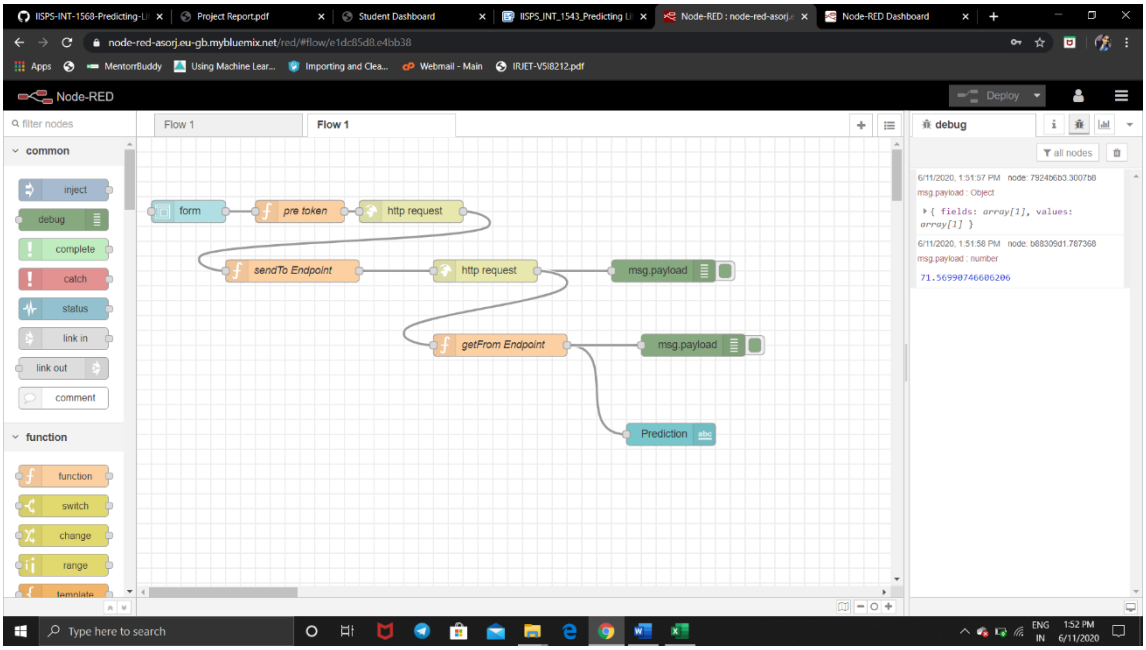



Fig.: Node-Red flow

## Result-

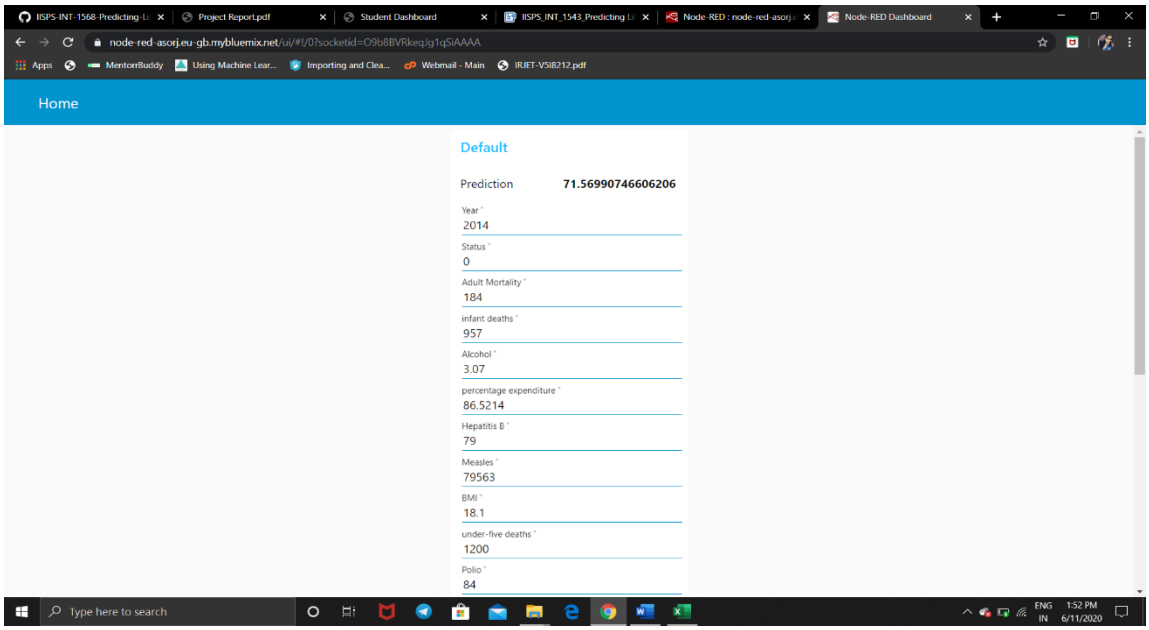



Fig.: Prediction (ui)

# Advantages & disadvantages-

## Advantages-

- Life expectancy plays an important role when decisions about the final phase of life need to be made.
- By increasing life expectancy of particular we can contribute to healthy environment.
- It directly contribute toward the growth rate of country.
- Advantages of using IBM Watson: Easy for user to interact with the model via the UI. User-friendly. Easy to build and deploy. Doesn't require much storage space.

## Disadvantages-

- As it depends completely on user, so if user provides some wrong values then it will predict wrong value.
- The model predicts average or approximate value but not accurate value.

# Applications-

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

## Conclusion-

Thus, we have developed a model that will predict the life expectancy of based on the inputs provided. Various factors have a significant impact on the life span such as Adult Mortality, Population, Under 5 Deaths, Thinness 1-5 Years, Alcohol, HIV, Hepatitis B, GDP, Percentage Expenditure and many more. User can interact with the system via a simple user interface which is in the form of a form with input spaces which the user needs to fill the inputs into. And we will get our predicted life expectancy.

## Future scope-

- We can reduce number of feature depends on their contribution at giving final value.
- We can connect input submission and prediction in storage(database) so that after having large data accuracy will increase accordingly.
- We can make interesting UI by some modifications.

## 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-labshttps://www.youtube.com/embed/r7E1TJ1HtM0
https://bookdown.org/caoying4work/watsonstudioworkshop/jn.html
https://www.kaggle.com/kumarajarshi/life-expectancy-who
https://www.youtube.com/watch?v=Jtej3Y6uUng

https://www.datacamp.com/users/sign_in?redirect=https://learn.datacamp.com/projects/758

## Appendix

### A source code-

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import types

import pandas as pd

from botocore.client import Config

import ibm_boto3


def __iter__(self): return 0


# @hidden_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.

# You might want to remove those credentials before you share the notebook.

client_3f9688f9615b4905ac51c6a765daa457 = ibm_boto3.client(service_name='s3',

    ibm_api_key_id=-----,

    ibm_auth_endpoint="-----",

    config=Config(signature_version=--),
```

```
    endpoint_url='---')


body = client_3f9688f9615b4905ac51c6a765daa457.get_object(Bucket='lifeexpectancydata-donotdelete-pr-2qg4cpdsrzfzus',Key='Life Expectancy Data.csv')['Body']

# add missing __iter__ method, so pandas accepts body as file-like object

if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )


data = pd.read_csv(body)

data.head()

plt.figure(figsize = (14, 10))

sns.heatmap(data.corr(), annot = True)

print('\n Data shape: ',data.shape)

print('\n Columns in data set: ',data.columns)

print('\n',data.info())

print('\n',data.corr())

print('\n',data.isnull().sum(axis=0))

print('\n',data.isnull().sum(axis=0))

from sklearn.preprocessing import LabelEncoder

lb = LabelEncoder()

data['Status'] = lb.fit_transform(data['Status'])

data = data.drop('Country', axis=1)

data.fillna(value=data.mean(),inplace=True)

data_shape = data.shape

data_shape

X = data.drop(['Life expectancy '], axis=1).values

y = data['Life expectancy '].values

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.3,random_state=75)

from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
lr.fit(X_train,y_train)
y_prediction = lr.predict(X_test)
y_prediction
from sklearn.metrics import mean_squared_error, accuracy_score, mean_absolute_error
print('\naccuracy(R^2): {}%'.format(lr.score(X_test, y_test)*100))
print('\nmae: {}'.format(mean_absolute_error(y_test, y_prediction)))
print('\nrmse: {}'.format(np.sqrt(mean_squared_error(y_test, y_prediction))))
!pip install watson-machine-learning-client
from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_credentials={
  "apikey": "----" ,
   "instance_id": "----",
  "url": "---"
   }
client=WatsonMachineLearningAPIClient(wml_credentials)
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "--",
client.repository.ModelMetaNames.AUTHOR_EMAIL: "---@gmail.com",
client.repository.ModelMetaNames.NAME: "----"}
model_artifact =client.repository.store_model(lr, 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="-----")
scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint
```