PROJECT REPORT

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

Submitted by-

SHREYA SAXENA

EMAIL ID :
saxena.shreyaa@gmail.com

GITHUB :
https://github.com/SmartPracticeschool/llSPS-INT-1707-Predicting-Life-Expectancy-using-Machine-Learning

PROJECT LINK :
https://node-red-mjktp.eu-gb.mybluemix.net/ui/#!/0?socketid=zEAUMl51kYBVPLXTAAAV

VIDEO DEMONSTRATION :
https://drive.google.com/file/d/1JQ1t6jI4GNEooHPlZtm6CTAQKRiwrAJQ/view?usp=drivesdk

Contents

1. INTRODUCTION

1.1 Overview

The main objective of the project is to predict the life expectancy of a person depending on several factors based on an individual or the residing country. Factors like the GDP of the country, health care facility system, quality of life, mental and physical illness, age, gender, education and other regional, demographic and economic factors are considered to predict the lifespan of the person using machine learning algorithms.

Machine learning algorithms that can be used in this case are: Regression, Decision Tree, Random Forest, Clustering techniques, so that we can achieve high accuracy for our model.

1.2 Purpose

Predicting the life expectancy will give the country an idea of the factors which can be improved to increase the lifespan of the people living, like by improving the health care facilities or immunization vaccines for infants. By making changes in lifestyle, a person can live a long, healthy and good quality life. This will also benefit the country by increasing manpower that will contribute to the economical growth. We should take full advantage of this new era advanced technology to improve the future by predicting it in the present.

2. LITERATURE SURVEY

2.1 Existing Solution

We have reviewed existing works and techniques used in the prediction of the human life expectancy, and reached a conclusion that it is feasible for individuals using evolving technologies and devices wearables and mobile health monitoring devices. We have also identified that the factors used for predicting were just personal causes and not related to the surrounding, healthcare facilities, demographic, social, regional and economical factors of the country he resides. These country dependent factors can also be an important feature to predict the life expectancy of an individual. So we need more data to predict more accurately.

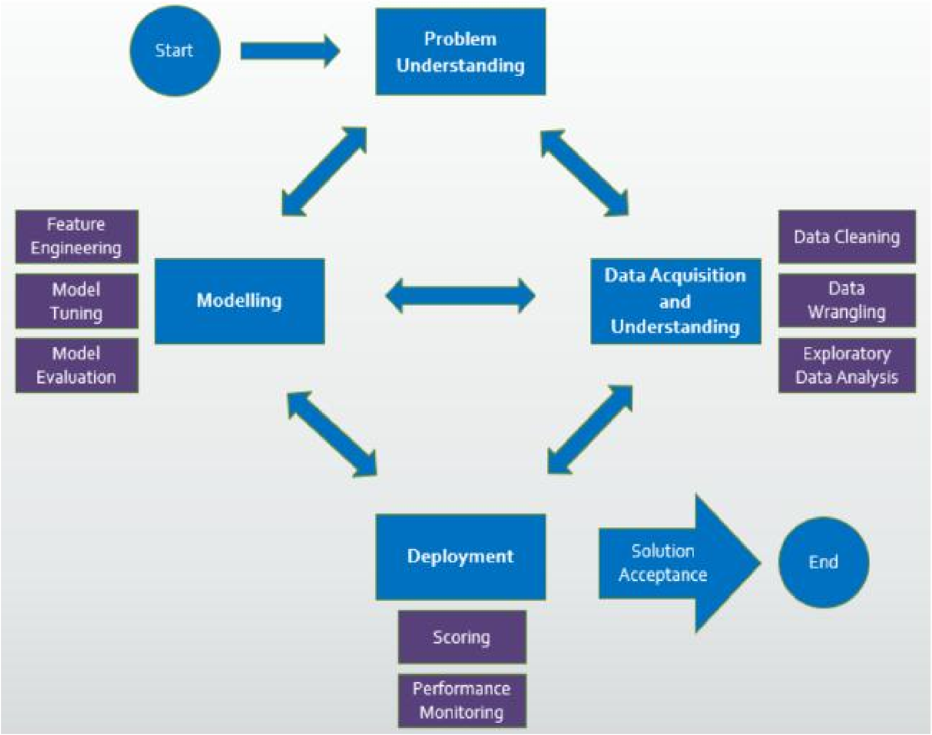
2.2 Proposed Solution

To get better insights and predict the life expectancy more accurately, we need to consider some additional features such as country or surrounding dependent features. The previous factors were more human based but it is important to know the economical, regional, social and demographic factors like the GDP, population, education, immunizations, history of illness, health care facility, funds allocated by the government, schemes, medical expenditures like if it is very high then people will shy away to get regular medical checkups, and many more.
Due to the large data, we will use IBM Cloud to build our model which will increase the project efficiency.

We will be using the dataset : Life Expectancy by WHO from Kaggle.

3. THEORITICAL ANALYSIS

3.1 Block diagram:



3.2 Hardware / Software designing:

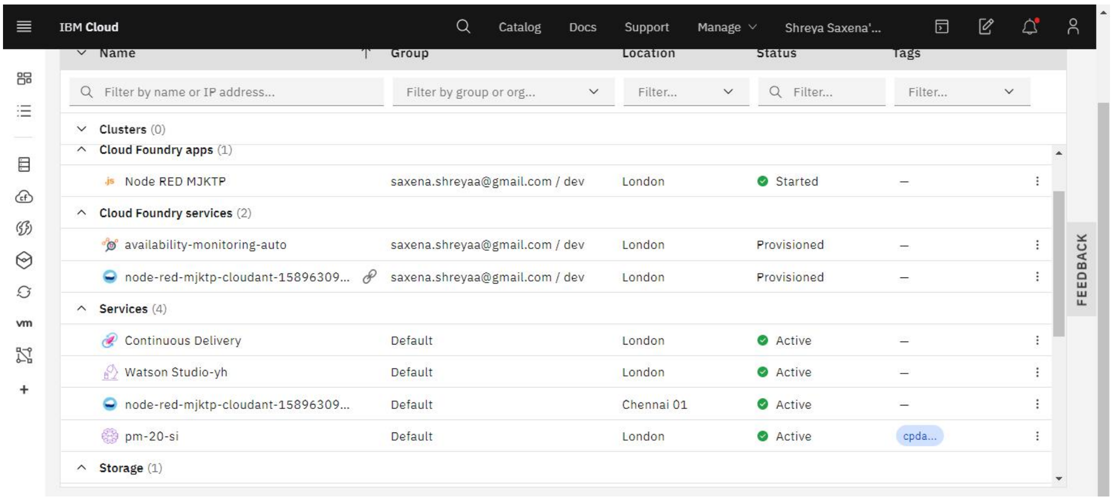
Functional Requirements:

1. The dataset should be preprocessed before modelling.
2. Feature Engineering should be performed to select co-related and integer values.
3. Appropriate machine learning algorithm should be used.
4. The project should be implemented using IBM Watson Studio which should then connected to Node-Red App for UI and deployed.

Software Requirements:

Python IDE, IBM Watson Studio, IBM Machine Learning Services, IBM Cloud, Node-Red App, Excel.

4.EXPERIMENTAL INVESTIGATIONS

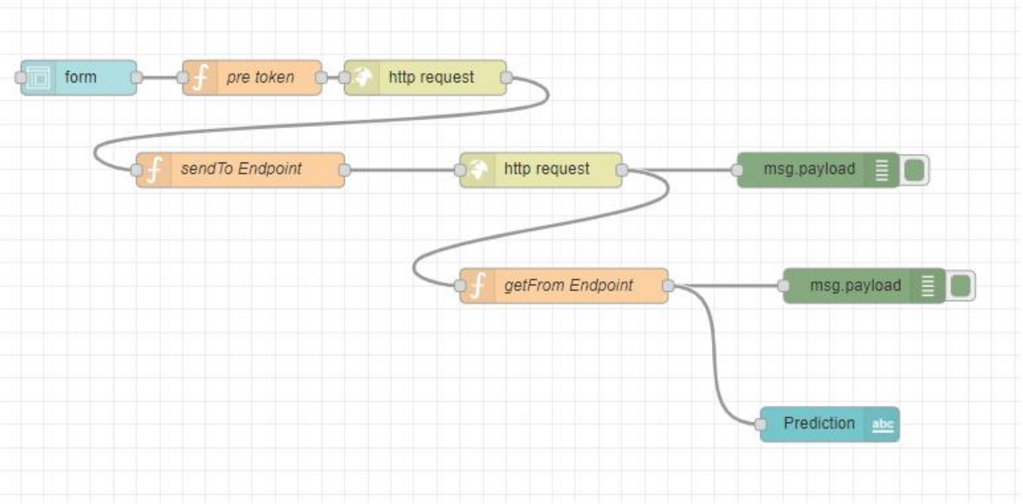


Name	Group	Location	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...
Clusters (0)				
Cloud Foundry apps (1)				
Node RED MJKTP	saxena.shreyaa@gmail.com / dev	London	Started	–
Cloud Foundry services (2)				
availability-monitoring-auto	saxena.shreyaa@gmail.com / dev	London	Provisioned	–
node-red-mjktp-cloudant-15896309...	saxena.shreyaa@gmail.com / dev	London	Provisioned	–
Services (4)				
Continuous Delivery	Default	London	Active	–
Watson Studio-yh	Default	London	Active	–
node-red-mjktp-cloudant-15896309...	Default	Chennai 01	Active	–
pm-20-si	Default	London	Active	cpda...
Storage (1)				

Resourse list

The model is first built using linear regression that had the R2 score of 0.82 and then random forest regressor was implemented, which gave an R2 score of 0.96. So, this gives us an better accuracy of the model.

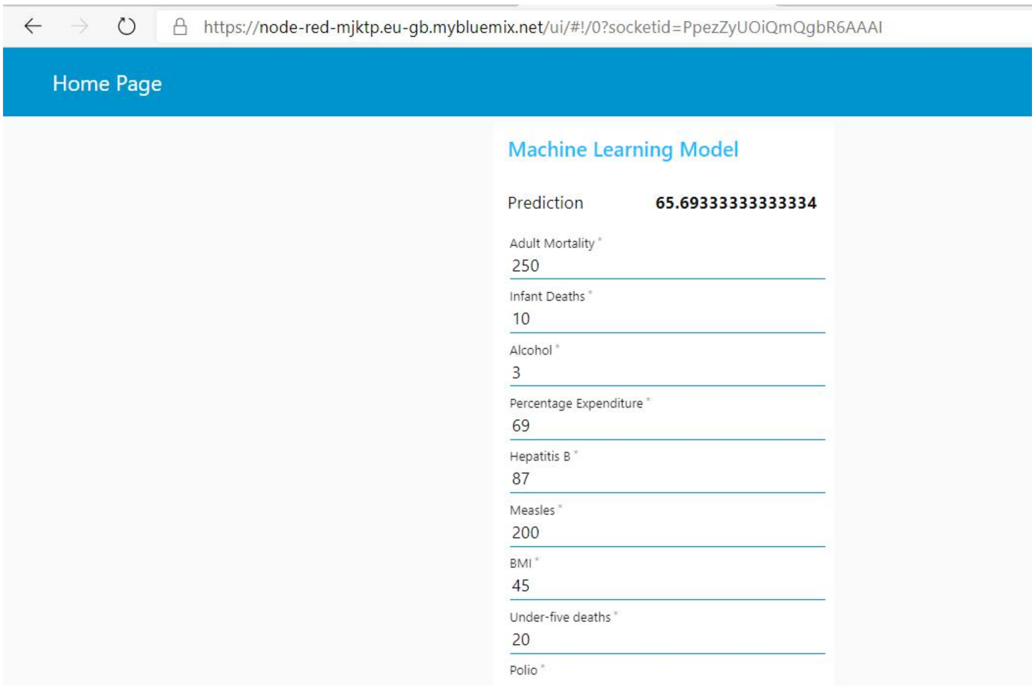
5. FLOWCHART



Node-Red Flow

6. RESULTS

User has to fill the form with various fields and the prediction will come after clicking on the submit button.



https://node-red-mjktp.eu-gb.mybluemix.net/ui/#!/0?socketid=PpezZyUOiQmQgbR6AAAI

Home Page

Machine Learning Model

Prediction 65.693333333333334

Adult Mortality *
250

Infant Deaths *
10

Alcohol *
3

Percentage Expenditure *
69

Hepatitis B *
87

Measles *
200

BMI *
45

Under-five deaths *
20

Polio *

7. ADVANTAGES & DISADVANTAGES

Advantages:

- The life expectancy predictor will give important insights and help people achieve good quality of life in future. The country can plan and improve various healthcare facilities.
- Benefit the country's growth.
- Advantages of using IBM Cloud: Easy to use and deploy, easy to connect with UI, takes care of large storage space.

Disadvantages:

- Requires internet connection.
- User input is not saved in any database.
- Input should be in range only to predict accurate values.

8. APPLICATIONS:

- To analyze country’s growth statistics in future years.
- To help government prepare life insurance policies for people. This will benefit the people.
- To analyze all the factors and plan out measures to increase the life expectancy of the country.

9. CONCLUSION

Thus, we have developed a model that will predict the life expectancy of a person living in a specific region. Various factors like Adult Mortality, Population, Under 5 Deaths, Thinness 1-5 Years, Alcohol, HIV, Hepatitis B, GDP,Percentage Expenditure and many more play an important role in the prediction. User can interact with the system via a simple user interface which is in the form.

10. FUTURE SCOPE

The accuracy of the model can be increased. This can be done by training more data. Also, the website can be added with many more features to improve the user experience. The user input can be connected to the database for future purposes.

11. BIBILOGRAPHY

https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html
https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
https://github.com/SmartPracticeschool/llSPS-INT-1707-Predicting-Life-Expectancy-using-Machine-Learning

12. APPENDIX

A. Source code

```
import types
import numpy as np # for numeric calculation
import pandas as pd # for data analysis and manupulation
import matplotlib.pyplot as plt # for data visualization
import seaborn as sns # for data visualization
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
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_82e38e728a2b4498b0106e3a414a28a9 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='0yEJ4xOS-jcMr50Vx1CLVy4cPDOsvlpYXMlI4gXTAjFf',
    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_82e38e728a2b4498b0106e3a414a28a9.get_object(Bucket='lifeexpectancy-don
otdelete-pr-xhe5gejavgecm9',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 )

# If you are reading an Excel file into a pandas DataFrame, replace `read_csv` by
`read_excel` in the next statement.
df= pd.read_csv(body)
df.head()
df.columns
df.info()
df.isna().sum()
df=df.fillna(df.mean())
df.isna().sum()
df.describe()
plt.figure(figsize=(12,9))

sns.heatmap(df.corr(), annot = True)
plt.title("Heatmap using correlation matrix", fontsize = 25)

d = {'Life expectancy ':1 , 'Adult Mortality':2 ,
          'Alcohol':3 , 'percentage expenditure': 4, 'Hepatitis B': 5,
         'Measles ' : 6, ' BMI ': 7, 'under-five deaths ' : 8, 'Polio' : 9, 'Total
expenditure' :10,
         'Diphtheria ':11, ' HIV/AIDS':12, 'GDP':13, 'Population' :14,
         ' thinness  1-19 years' :15, ' thinness 5-9 years' :16,
         'Income composition of resources' : 17, 'Schooling' :18, 'infant deaths':19}

plt.figure(figsize=(20,30))
```

```
for variable,i in d.items():
                                  plt.subplot(4,5,i)
                                  sns.boxplot(df[variable],orient='v')
                                  plt.title(variable)

plt.show()

plt.figure(figsize=(170,70))
sns.pairplot(df)

df.groupby('Country')['Life expectancy '].mean().sort_values(ascending=False)

country = df.groupby('Country')['Life expectancy
'].mean().sort_values(ascending=False)
country.plot(kind='bar', figsize=(60,20), fontsize=25)
plt.title("Life Expectancy",fontsize=40)
plt.xlabel("Country",fontsize=35)
plt.ylabel("Avg Life_Expectancy",fontsize=35)
plt.show()

sns.distplot(df['Life expectancy '])
df['Status'].unique()
df.groupby('Status')['Country'].count()
df.groupby('Status')['Life expectancy '].mean()
status_dummy=pd.get_dummies(df['Status'])
df.drop(['Status'],inplace=True,axis=1)
df=pd.concat([df,status_dummy],axis=1)
df.drop(['Country'],inplace=True,axis=1)
y= df['Life expectancy ']
df= df.drop('Life expectancy ',axis=1)
df.drop('Year',inplace=True,axis=1)
df
X_train, X_test, y_train, y_test = train_test_split(df, y, test_size=0.2,
random_state=20)

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)

rfmodel = RandomForestRegressor(n_estimators = 30, random_state = 70)
rfmodel.fit(X_train, y_train)

pred= rfmodel.predict(X_test)

print("Training score:",rfmodel.score(X_train, y_train))
print("Testing Score:",rfmodel.score(X_test, y_test))

print("RMSE:",mean_squared_error(y_test,pred)**(0.5))
print("R2 score:",r2_score(y_test,pred))
```

R2 score with linear regression - 0.82

R2 score with random forest regressor - 0.96

```
sns.distplot(pred)
plt.scatter(pred,y_test)

!pip install watson-machine-learning-client
from watson_machine_learning_client import WatsonMachineLearningAPIClient
wml_credentials={
   "apikey": "HOuF5adpBS3yWfhklWBdGt8hE_PuSpTntkszEMhlkNCR",
   "instance_id": "880979e9-4c4e-4cb7-b9ce-952885ac2ddb",
   "url": "https://eu-gb.ml.cloud.ibm.com"
}

client = WatsonMachineLearningAPIClient( wml_credentials )

model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Shreya",
client.repository.ModelMetaNames.AUTHOR_EMAIL:
"saxena.shreyaa@gmail.com",
client.repository.ModelMetaNames.NAME: "LifeExp"}
model_artifact =client.repository.store_model(rfmodel, 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="LifeExp")
@hidden_cell
scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint
```