**PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING**

**A PROJECT REPORT**

Submitted as a course project for

**MACHINE LEARNING INTERNSHIP**

**Under SMARTBRIDGE**

Submitted by

ADARSH PAL (adarshpal999@gmail.com)

**Under the guidance of**

Mr. Prashanth V.

Mr. M. Charan Reddy

**1. INTRODUCTION**

* 1. **) Overview**
* A typical Regression Machine Learning project leverages historical data to predict insights into the future. This project 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.

**1.2) Purpose**

* This project 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.

**2. LITERATURE SURVEY**

**2.1) Existing Problem**

* Life expectancy of any given country is a very important feature and should be taken well care of. Life expectancy of any country describes how well the health of the people is and how good the health infrastructure of that country is.
* For better services for the residents of the country, the country needs to know the average life expectancy and how the value of the life expectancy varies according to the various factors such as illness from many different types of diseases and alcohol intake, education, expenditure on healthcare system and many more.
* This would help the country to better their healthcare system and ultimately would help them to save lives of their citizens.

**2.2) Proposed Solution**

* A simple solution to this problem is to create a machine learning regression model using the dataset that is available for various countries and this dataset is available freely at kaggle.
* The dataset used is record for 193 countries and is made during a span of 15 years (2000-2015).
* This data is then used to build a machine learning model with high accuracy of prediction and then that value can be used for further analysis.

**3. THEORITICAL ANALYSIS**

**3.1) Block Diagram**

Input values to the fields such as 'country', 'BMI', 'Total expenditure' , 'measles', 'Status', HIV/AIDS', 'Alcohol' , 'percentage expenditure' and etc to the blank fields in webpage

Predicted Life Expectancy value

Deployed machine learning model with maximum score of accuracy

**3.2) Software Designing**

* The following packages have been imported NymPy, Pandas, Matplotlib, Scipy, Seaborn. Sklearn is the most widely used package for the machine learning process. The following sub packages have been used:

1. train\_test\_split
2. linear\_model
3. model\_selection
4. metrics
5. tree
6. ensemble
7. preprocessing

* The data is saved as a csv file as LifeExpectancy.csv and it is read and stored in the life\_data variable. The Year column is dropped as it will not be used in the analysis. Below the first 5 rows are shown. The data contains 21 columns and 2938 rows with the header row.
* Using the scatter plot we plot the Life Expectancy against some other variables to see if there is any correlation between them.  
  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, less than 5 years – old deaths and Life Expectancy.
* Now we will plot the correlation matrix visualizing it with a heat map. The legend tells that the warmer colors show higher and positive correlation, while the colder low or negative.  
  There is a very high correlation between thinness of 5-9 year-old and that of 1-19 year-old. Also between population and infant deaths, under 5 deaths, another is between schooling and income composition of resources. On the other hand Life expectancy and Adult Mortality are very highly negatively correlated.

**3.3) PROJECT RESOURCES**

* **IBM Cloud** : For data storage
* **IBM Watson Studio** : Creation and Deployment of ML Model
* **NODE RED** : For Web UI and Interfacing with Model
* **Slack** : Communication with teammates and mentors

 3.**4) PROJECT DELIVERABLES**

* A machine learning model using regression algorithms is made and deployed with an accuracy of 90%+ that uses data provided by WHO available at kaggle to train and test the model.
* After deployment of model, a web UI is created using Node-Red available on IBM Cloud and is used to integrate the machine learning service with the web UI.
* After the completion of the project all the necessary files and source codes for model and UI is uploaded on github for further improvisation or distribution.

**3.5) PROJECT TIMELINE**

* **Week 1 :**Exploratory Data Analysis of the Dataset provided
* **Week 2 :**Build Model, Training model and Testing for accuracy
* **Week 3 :**Creation of Web UI using Node-red in IBM Cloud
* **Week 4 :**Integrating Model with UI and test its working

**4. EXPERIMENTAL INVESTIGATIONS**

**4.1) Exploratory Data Analysis**

* The raw data is not suitable for us to start building a model so some preprocessing will be done.
* First the Status of the country is turned into numerical with the get\_dummies function, so we get 2 new columns.
* The original column is being dropped. Second the data is being grouped by the country and we find the mean values during the 2000 – 2015 year period.
* Then the Life expectancy column is removed to form the life\_labels variable or the output, and the rest is stored as the life features variable.
* Now we consider that we have some null values in the table, the isnull function has been used to find the with the Boolean True.
* Below that the number of null values are displayed in each separate column.
* It is mostly situated in the Population and GDP columns.
* Now the missing values are filled with the mean of its respective column.
* This will create some distortions, but the other option in removing parts of the table will shrink the data so it will be avoided here because the number of rows is not that high.
* The final shape for the life\_features is 193 rows to 20 columns.
* Finally considering the large differences in the values of the columns, there will be some scaling with the MinMaxScaler function.
* Now we will split the data into a training part of 80% and a testing of 20%.
* Cross validation will be initialized with the creation of 5 fold split.

**5. FLOWCHART**

Input values to the fields such as 'country', 'BMI', 'Total expenditure' , 'measles', 'Status', HIV/AIDS', 'Alcohol' , 'percentage expenditure' and etc to the blank fields in webpage

Predicted Life Expectancy value

Deployed machine learning model with maximum score of accuracy

**6. RESULT**

* After comparing all the algorithms we can conclude the Lasso and the Ridge Regression offer which are the same:
* Best Parameters: {‘alpha’: 0, ‘max\_iter’: 10}
* R square on the test data of 93%
* MAE of 1.83
* MSE of 6.05

**7. ADVANTAGES AND DISADVANTAGES**

**7.1) Advantages**

* + The model that is made using regression analysis fits the dataset provided with high accuracy.
  + It can be used to predict life expectancy based on various characteristics provided.
  + The predicted value can be used to further analyze the state of healthcare system in any country.

**7.2) Disadvantages**

* + As this model is only based on data which was present in the past, there is no guarantee that it will predict accurate values for the future as well.
  + Some widespread disease causing large number deaths may impact the accuracy of the model, Ex: Corona virus
  + Any disease or factor that may affect the lifespan on any individual if not included in the data may result in reduced accuracy and precision of the model.

**8. APPLICATIONS**

* Life expectancy of any given country is a very important feature and should be taken well care of. Life expectancy of any country describes how well the health of the people is and how good the health infrastructure of that country is.
* For better services for the residents of the country, the country needs to know the average life expectancy and how the value of the life expectancy varies according to the various factors such as illness from many different types of diseases and alcohol intake, education, expenditure on healthcare system and many more.
* This would help the country to better their healthcare system and ultimately would help them to save lives of their citizens.

**9. CONCLUSION**

* + After comparing all the algorithms we can conclude the Lasso and the Ridge Regression offer which are the same:
  1. Best Parameters: {‘alpha’: 0, ‘max\_iter’: 10}
  2. R square on the test data of 93%
  3. MAE of 1.83
  4. MSE of 6.05
  5. A simple solution to this problem is to create a machine learning regression model using the dataset that is available for various countries and this dataset is available freely at kaggle.
  6. The dataset used is record for 193 countries and is made during a span of 15 years (2000-2015).
  7. This data is then used to build a machine learning model with high accuracy of prediction and then that value can be used for further analysis.

**10. FUTURE SCOPE**

* + Life expectancy of any given country is a very important feature and should be taken well care of. Life expectancy of any country describes how well the health of the people is and how good the health infrastructure of that country is.
  + For better services for the residents of the country, the country needs to know the average life expectancy and how the value of the life expectancy varies according to the various factors such as illness from many different types of diseases and alcohol intake, education, expenditure on healthcare system and many more.
  + This would help the country to better their healthcare system and ultimately would help them to save lives of their citizens.
  + As this model is only based on data which was present in the past, there is no guarantee that it will predict accurate values for the future as well.
  + Some widespread disease causing large number deaths may impact the accuracy of the model, Ex: Corona virus

**11. BIBLIOGRAPHY**

The following sources have been used:

* <https://www.kaggle.com/kumarajarshi/life-expectancy-who/data>
* <https://datasciencesociety.com>
* <https://cloud.ibm.com>
* <https://stackoverflow.com>
* Introduction to Machine Learning with Python by Andreas C. Müller & Sarah Guido

**12. APPENDIX**

* **Source Code:**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

from sklearn.linear\_model import Ridge

from sklearn.linear\_model import Lasso

from sklearn.linear\_model import ElasticNet

from sklearn.metrics import make\_scorer

import seaborn as sns

import types

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\_9b57de40ef994bb1be3c8a2dfe52311a = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*',

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\_9b57de40ef994bb1be3c8a2dfe52311a.get\_object(Bucket='lifeexpectancy-donotdelete-pr-hfccnyjh4kp9b4',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.

life\_data = pd.read\_csv(body)

life\_data.head()

life\_data = pd.read\_csv(r"C:\Users\asus\Downloads\Life Expectancy Data.csv")

life\_data.head()

life\_data = life\_data.drop('Year',axis = 1)

status = pd.get\_dummies(life\_data.Status)

life\_data = pd.concat([life\_data, status], axis = 1)

life\_data = life\_data.drop(['Status'], axis=1)

life\_data.rename(columns = {'Developing' : 0, 'Developed' : 1})

life\_data = life\_data.groupby('Country').mean()

life\_data.head()

life\_labels = life\_data['Life expectancy ']

life\_features = life\_data.drop('Life expectancy ', axis = 1)

life\_features.fillna(value = life\_features.mean(), inplace = True)

life\_labels.fillna(value = life\_labels.mean(), inplace = True)

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

sns.heatmap(life\_data.corr(), annot = True)

life\_features\_train, life\_features\_test, life\_labels\_train, life\_labels\_test = train\_test\_split(life\_features, life\_labels, train\_size = 0.8, test\_size = 0.2)

linear\_model = LinearRegression()

linear\_model.fit(life\_features\_train, life\_labels\_train)

print('R\_square score on the training data : %.2f' % linear\_model.score(life\_features\_train, life\_labels\_train))

linear\_model\_predict = linear\_model.predict(life\_features\_test)

print('Coefficients: \n', linear\_model.coef\_)

print("Mean squared error(MSE) : %.2f"

% mean\_squared\_error(life\_labels\_test, linear\_model\_predict))

print("Mean absolute error(MAE) : %.2f"

% mean\_absolute\_error(life\_labels\_test, linear\_model\_predict))

print('R\_square score on test data: %.2f' % r2\_score(life\_labels\_test, linear\_model\_predict))

scoring = make\_scorer(r2\_score)

grid\_cv = GridSearchCV(Ridge(), param\_grid={'alpha': range(0, 10), 'max\_iter' : [10, 100, 1000]}, scoring=scoring, cv=5, refit=True)

grid\_cv.fit(life\_features\_train, life\_labels\_train)

print("Best Parameters: " + str(grid\_cv.best\_params\_))

print("R\_square score on training data : %.2f" %grid\_cv.score(life\_features\_train, life\_labels\_train))

print("R\_square score on test data : %.2f" % r2\_score(life\_labels\_test, grid\_cv.best\_estimator\_.predict(life\_features\_test)))

print("Mean squared error(MSE) : %.2f" % mean\_squared\_error(life\_labels\_test, linear\_model\_predict))

print("Mean absolute error(MAE) : %.2f" % mean\_absolute\_error(life\_labels\_test, linear\_model\_predict))

scoring = make\_scorer(r2\_score)

grid\_cv = GridSearchCV(Lasso(),param\_grid={'alpha': range(0, 10), 'max\_iter' : [10, 100, 1000]},scoring=scoring, cv=5, refit=True)

grid\_cv.fit(life\_features\_train, life\_labels\_train)

print("Best Parameters: " + str(grid\_cv.best\_params\_))

print("R\_square score on training data: %.2f" % grid\_cv.score(life\_features\_train, life\_labels\_train))

print("R\_square score on test data: %.2f" % r2\_score(life\_labels\_test, grid\_cv.best\_estimator\_.predict(life\_features\_test)))

print("Mean squared error(MSE): %.2f" % mean\_squared\_error(life\_labels\_test, linear\_model\_predict))

print("Mean absolute error(MAE): %.2f" % mean\_absolute\_error(life\_labels\_test, linear\_model\_predict))

scoring = make\_scorer(r2\_score)

grid\_cv = GridSearchCV(ElasticNet(),param\_grid={'alpha': range(0, 10), 'max\_iter' : [10, 100, 1000], 'l1\_ratio' : [0.1, 0.4, 0.8]},scoring=scoring, cv=5, refit=True)

grid\_cv.fit(life\_features\_train, life\_labels\_train)

print("Best Parameters: " + str(grid\_cv.best\_params\_))

print("R\_square score on training data: %.2f" % grid\_cv.score(life\_features\_train, life\_labels\_train))

print("R\_square score on test data: %.2f" % r2\_score(life\_labels\_test, grid\_cv.best\_estimator\_.predict(life\_features\_test)))

print("Mean squared error(MSE) : %.2f" % mean\_squared\_error(life\_labels\_test, linear\_model\_predict))

print("Mean absolute error(MAE) : %.2f" % mean\_absolute\_error(life\_labels\_test, linear\_model\_predict))

from watson\_machine\_learning\_client import WatsonMachineLearningAPIClient

wml\_credentials={

"apikey": "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",

"instance\_id": "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",

"password": "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",

"url": "https://eu-gb.ml.cloud.ibm.com",

"username": "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

}

client = WatsonMachineLearningAPIClient( wml\_credentials )

model\_props = {client.repository.ModelMetaNames.AUTHOR\_NAME: "Adarsh Pal",

client.repository.ModelMetaNames.AUTHOR\_EMAIL: "adarshpal999@gmail.com",

client.repository.ModelMetaNames.NAME: "LifeExpectancy"}

model\_artifact =client.repository.store\_model(grid\_cv, 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