SMARTINTERNZ INTERNSHIP PROJECT REPORT ON PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

BY RAMIJ AMED SARDAR

1. Introduction:

Since ancient times, there are a lot of change in the behaviours and cultures of people in different places. According to their way of living, the health care and life expectancy of people varies among each other.

1.1. Overview

Life expectancy is a statistical measure of the average time a human being is expected to live. 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. 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 in a dataset. In order to predict life expectancy rate of a given country, we will be using Machine Learning algorithms to draw inferences from the given dataset and give an output. For better usability by the customer, we are also going to be creating a UI for the user to interact with using Node-Red.

1.2. Purpose

The purpose of this project is that the people from various places can easily predict their life expectancy by providing the inputs asked by the model. This software can be used by all people in the world because the training part of this model contains inputs and predictions of more number of countries. Economic growth:

Predicting life expectancy would play a vital role in judging the growth and development of the economy. Across countries, high life expectancy is associated with high income per capita. Increase in life expectancy also leads to an increase in the “manpower” of a country. The knowledge
asset of a country increases with the number of individuals in a country. Population Growth:
Helps the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years. Personal growth:
This project would also help an individual assess his/her lifestyle choices and alter them accordingly to lead a longer and healthier life. It would make them more aware of their general health and its improvement or deterioration over time.
Growth in Health Sector:
Based on the factors used to calculate life expectancy of an individual and the outcome, health care will be able to fund and provide better services to those with greater need. Insurance Companies:
Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

2. Literature Survey:

There are so many organizations that are making research in the prediction of life expectancy. Many research papers dealing with the creation of this model under many algorithms such as Machine Learning, Deep learning and programming languages such as Python and Java script.

2.1. Existing Problem

The World Health Organization (WHO) began producing annual life tables for all Member States in 1999. These life tables are a basic input to all WHO estimates of global, regional and country-level patterns and trends in all-cause and cause-specific mortality. After the publication of life
tables for years to 2009 in the 2011 edition of World Health Statistics, WHO has shifted to a two year cycle for the updating of life tables for all Member States.

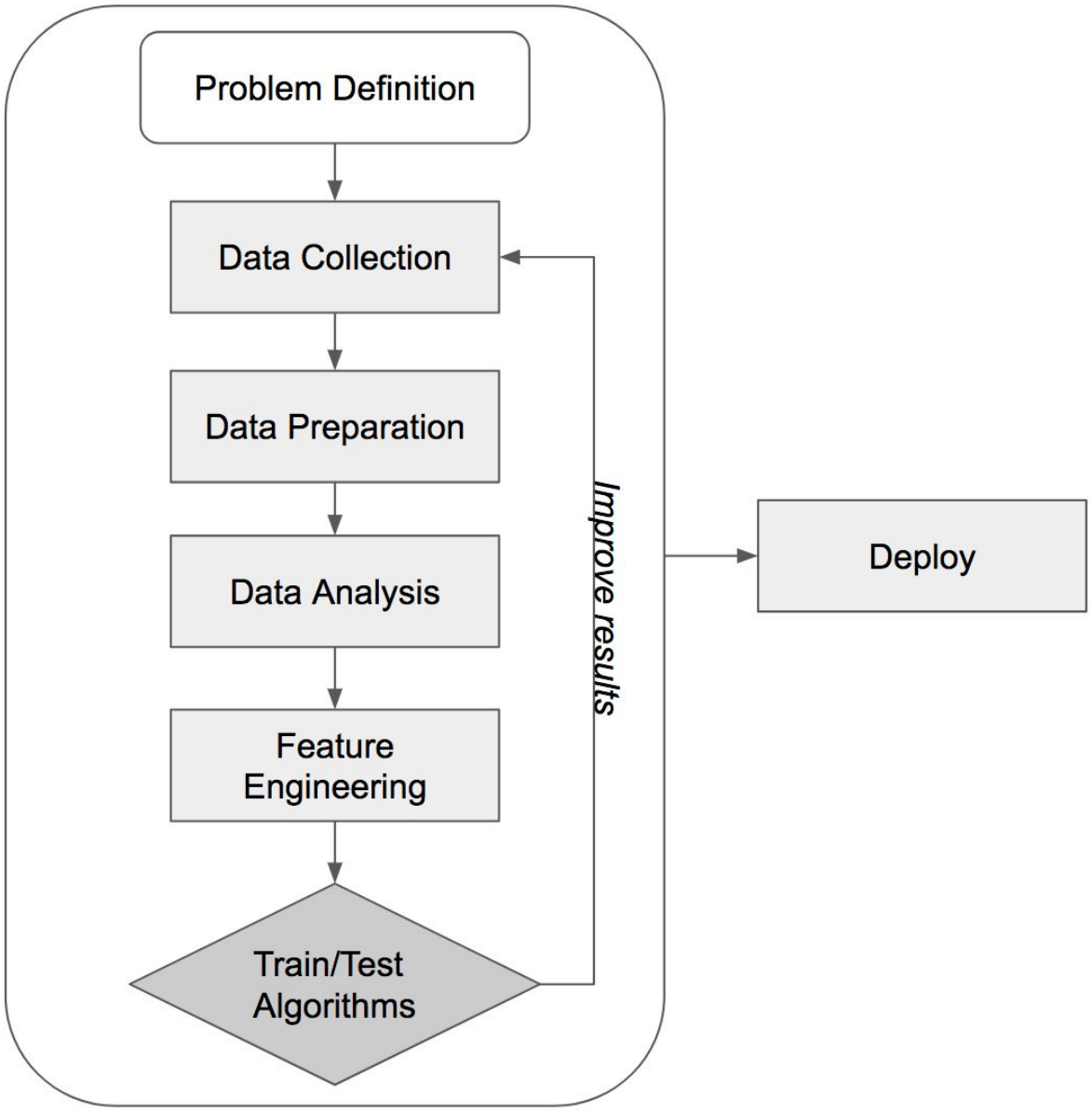
Even still the model is not really updated in every fields. WHO applies standard methods to the analysis of Member State data to ensure comparability of estimates across countries. This will inevitably result in differences for some Member States with official estimates for quantities such as life expectancy, where a variety of different projection methods and other methods are used.

2.2. Proposed Solution

So many people were expecting to use a model of life expectancy prediction. In order to that, many institutions and companies are leading their team to build that model. In my project, I have proposed a solution to predict the life expectancy using machine learning. Machine Learning is the process of training the computer to think and decide solutions like human. The reason why I have chosen this architecture was only with the help of Machine Learning, deep understanding of the data and an ability to create a model can be done. Design a Regression model to predict life expectancy ratio of a given country based on some features provided such as year, GDP (gross domestic product), education, alcohol intake of people in the country, expenditure on healthcare system and some specific disease related deaths that happened in the country.

3. Theoretical Analysis:

3.1. Block Diagram



3.2. Hardware / Software Designing

1. PROJECT PLANNING AND KICKOFF:

a. Understanding the project description and analyze the data and attributes in the given dataset.

b. Creating Github account
c. Installing Slack and create account with the mail id
d. Learning to use Zoho writer.

2. EXPLORE IBM CLOUD PLATFORM:

a. Creating IBM cloud account with the mail id
b. Creating IBM academic initiative account with the mail id
c. Create a Node-Red starter application.

3. EXPLORE IBM WATSON SERVICES:

a. Exploring IBM Watson use cases.
b. Learning about IBM Watson Machine Learning.

4. INTRODUCTION TO WATSON STUDIO:

a. Learning to build own Machine Learning model using
IBM Watson.
b. Automate the Machine Learning Model

5. PREDICTING LIFE EXPECTANCY WITH PYTHON:

a. Collecting Data set from www.kaggle.com
b. Creating IBM Watson services
c. Create a jupyter notebook and import data from Object
storage.

6. PREDICTING LIFE EXPECTANCY WITHOUT PYTHON:

a. Created Node-Red model and integrated with Machine
Learning model.

4. Experimental Investigation

Life Expectancy Dataset:

The dataset used is a life expectancy dataset released by the World Health Organization.

The data set has the following features:

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. The first 5 rows are shown below. The data contains 21 columns and 2938 rows with the header row. The table contains data about:

- Countries
- Status
- Life Expectancy
- Adult Mortality
- 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

5.Flowchart

To integrate the ML model with the UI, we would be using the Node Red functionality provided by the IBM Watson Studio.

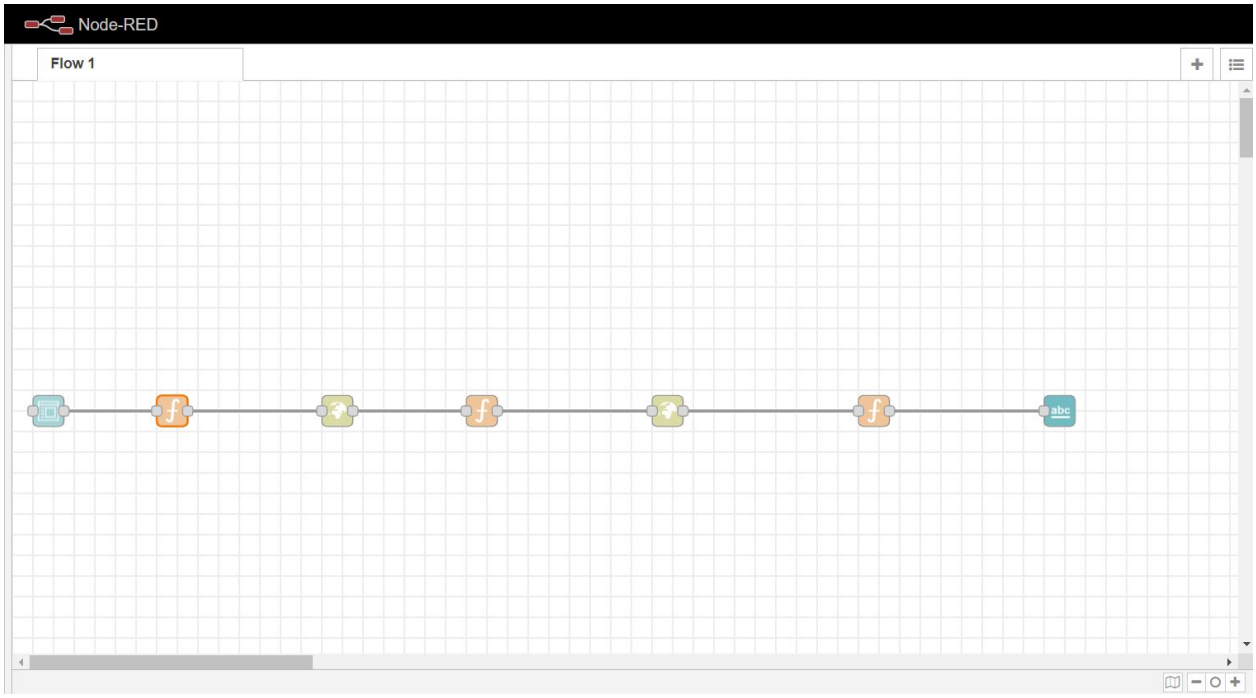
To design the UI, we need to import the flow of the UI.

Once, we have setup the flow, we need to integrate the ML model with it. To integrate the ML Model with it we need to access the endpoint URL of our ML Model.

Components of the flow are:

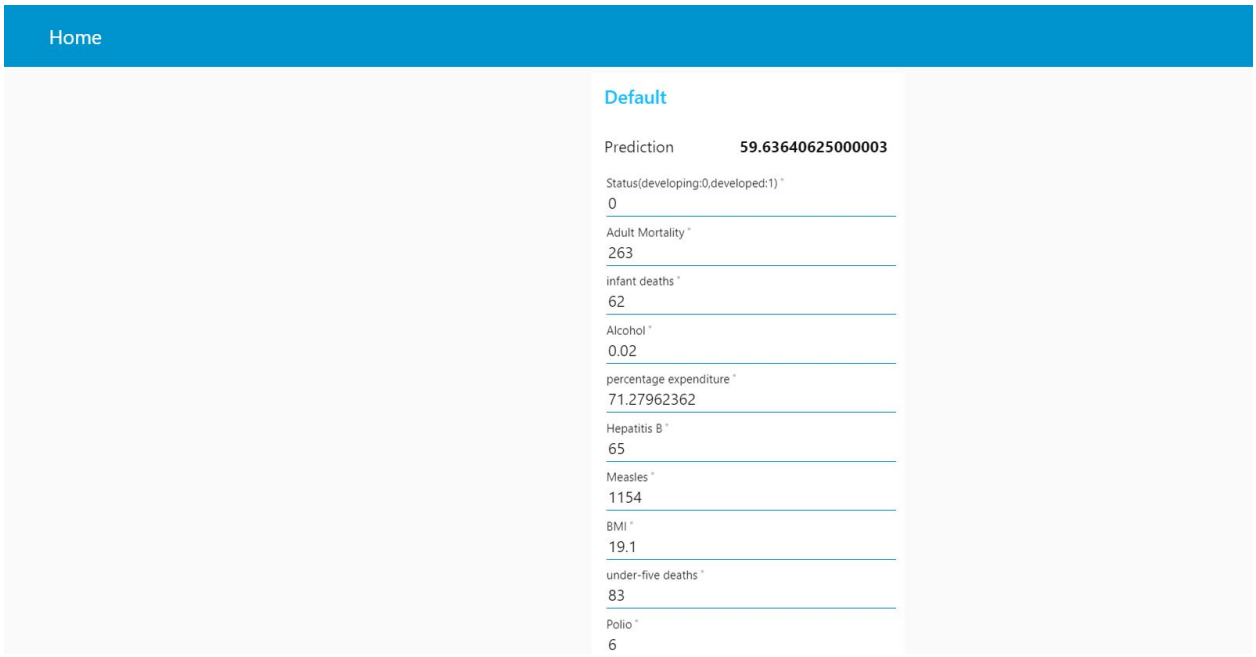
Form: The form contains all the elements of the UI. All the labels are associated with a variable.

Http requests: To setup the flow, we need two http requests.
The first http request requires a token to connect to the
machine learning service of the Watson studio.
The second http request helps us in integrating the model using
the endpoint URL.
Once the flow has been setup, we deploy the model.



6. Result

Web based UI was developed by integrating all the services using NODE-RED.
URL for UI
Dashboard:https://node-red-cejwj.eu-gb.mybluemix.net/ui/#!/0?socketid=k9kdsxMVtQsAblr3AAAX



7. ADVANTAGE & DISADVANTAGES

7.1. Advantages:

1. Advantages of using IBM Watson:

• Processes unstructured data

• Fills human limitations

• Acts as a decision support system, doesn’t replace humans

• Improves performance + abilities by giving best available data

• Improve and transform customer service

• Handle enormous quantities of data

• Sustainable Competitive Advantage 2. Easy for user to interact with the model via the UI. 3. User-friendly. 4. Easy to build and deploy. 5. Doesn’t require much storage space.

7.2. Disadvantages:

1. Disadvantages of using IBM Watson:

• Seen as disruptive technology

• Maintenance

• Doesn't process structured data directly

• Increasing rate of data, with limited resources 2. Not connected to database, hence no record of input. 3. Requires internet connection.

8. Applications

• Personalized Life Expectancy: Individuals can predict their own life expectancy by inputting values in the corresponding fields. This could help make people more aware of their general health, and its improvement or deterioration over time. This may motivate them to make healthier lifestyle choices.

• Government: It could help the government bodies take appropriate measures to control the population growth and also direct the utilization of the increase in human resources and skillset acquired by people over many years. Across countries, high life expectancy is
associated with high income per capita. Increase in life expectancy also leads to an increase in the "manpower" of a country. The knowledge asset of a country increases with the number of individuals in a country.

• Health Sector: Based on the factors used to calculate life expectancy of an individual and the outcome, health care will be able to fund and provide better services to those with greater need.

• Insurance Companies: Insurance sector will be able to provide individualized services to people based on the life expectancy outcomes and factors.

9. Conclusion

Thus, we have developed a model that will predict the life expectancy of a specific demographic region 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, and 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.

10. Future Scope

For future use, one can integrate the life expectancy prediction with providing suggestions and medications to the individual using the application. This will help predict as well as increase the individual's life expectancy. The scalability and

flexibility of the application can also be improved with advancement in technology and availability of new and improved resources. Also, with the growth in Artificial Neural networks and Deep learning, one can integrate that with our existing application. With the help of Convolutional Neural networks and Computer vision, we can also try to take into account the physical health and appearance of a person. Mental health can also be taken into account while predicting life expectancy with the help of sentiment analysis systems as well.

11. Bibliography

1. **Node-RED Starter Application :** https://developer.ibm.com/tutorials/how-to-create-a- node- red-starter-applicat ion/
2. **Watson Studio Cloud :** https://bookdown.org/caoying4work /watsonstudio-workshop/jn.html
3. **Dataset Reference :** https://www.kaggle.com/kumaraja rshi/life-expectancy-who
4. **IBM Cloud Services :** https://www.youtube.com/watch? v=DBRGlAHdj48&l ist=PLzpeuWUENMK2PYtasCaKK4bZjaYzhW23L 5. **Import the Dataset into Jupyter Notebook :** https:// www.youtube.com/watch?v=Jtej3Y6uUng

Appendix A.

Source Code

Python 3.6

Importing all Required libraries

In [1]:

```
from sklearn import metrics
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import cross_val_score
from scipy import stats
import seaborn as sns
import types
import pandas as pd
```

```
from botocore.client import Config
import ibm_boto3
```

Importing Dataset

In [2]:

```
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_00f5c14df33a4a429f58790dc2580d85 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='CQJmbL9nmJxFwbbVRl7gyVOlSVLotDqmaVojLRHc2Puy',
    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_00f5c14df33a4a429f58790dc2580d85.get_object(Bucket='myproject-donotdel
ete-pr-jqjywilxpofymz',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 )

d = pd.read_csv(body)
```

Reading and Analysing the Data

In [3]:

```
d.head()
```

Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 1-19 years	thinness 5-9 years	Income composition of resources	Schooling
Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.259210	33736494.0	17.2	17.3	0.479	10.1
Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.696514	327582.0	17.5	17.5	0.476	10.0
Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.744976	31731688.0	17.7	17.7	0.470	9.9
Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.959000	3696958.0	17.9	18.0	0.463	9.8
Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.537231	2978599.0	18.2	18.2	0.454	9.5

ws × 22 columns

In [4]:

```
d.columns
```

Out[4]:

```
Index([' Country', 'Year', 'Status', 'Life expectancy ', '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'],
      dtype='object')
```

Dropping and Renaming of columns

In [5]:

```
df=d.drop('Year',axis=1)
df.replace(to_replace=['Developing', 'Developed'],
           value= [0, 1],
           inplace=True)
```

Out[5]:

Grouping the data by Country

In [6]:

```
df=df.groupby(' Country').mean()
df.isnull().sum()
```

Out[6]:

Status	0
Life expectancy	10
Adult Mortality	10
infant deaths	0
Alcohol	2
percentage expenditure	0
Hepatitis B	9
Measles	0
BMI	4
under-five deaths	0
Polio	0
Total expenditure	2
Diphtheria	0
HIV/AIDS	0
GDP	30
Population	48
thinness 1-19 years	4
thinness 5-9 years	4
Income composition of resources	17
Schooling	13

dtype: int64

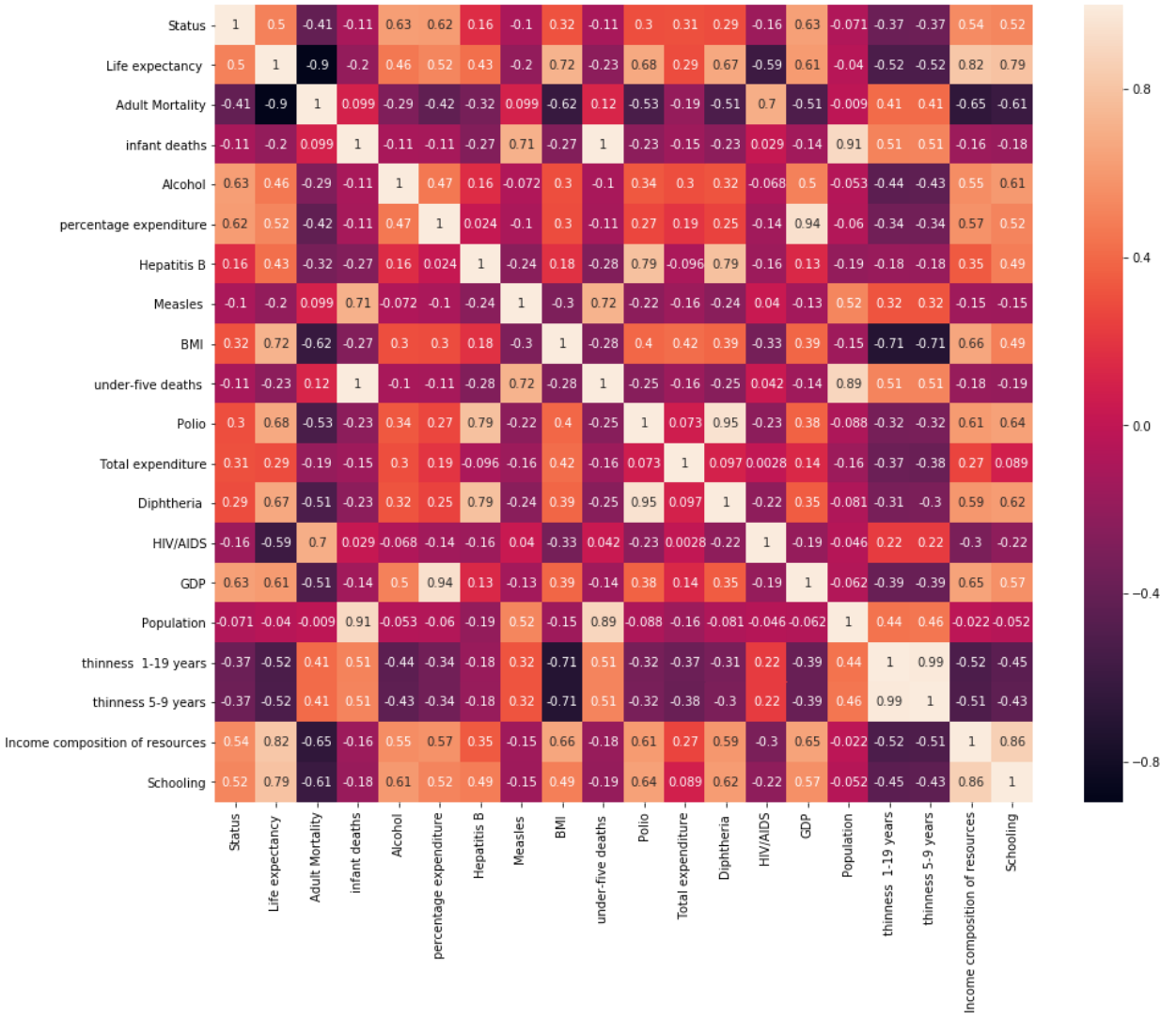
Visualizing the data

In [7]:

```
plt.figure(figsize = (15, 12))
sns.heatmap(df.corr(), annot = True)
```

Out[7]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f04740e5358>



Data preprocessing

In [8]:

```
#filling the empty place with average value
```

In [9]:

```
df.fillna(value = df.mean(), inplace = True)
X=df.drop(['Life expectancy '],axis=1)
Y=df['Life expectancy ']
```

Feature Scaling

In [10]:

```
min_max_scaler = MinMaxScaler()
```

```
X = min_max_scaler.fit_transform(X)
```

Splitting data set into train and test dataset

In [11]:

```
x_train,x_test,y_train,y_test=train_test_split(
       X,Y, train_size = 0.7, test_size = 0.3)
```

Import random forest regressor model and train the model

In [12]:

```
model=RandomForestRegressor(n_estimators=40,random_state=50)
model.fit(x_train,y_train)
```

Out[12]:

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
       max_features='auto', max_leaf_nodes=None,
       min_impurity_decrease=0.0, min_impurity_split=None,
       min_samples_leaf=1, min_samples_split=2,
       min_weight_fraction_leaf=0.0, n_estimators=40, n_jobs=None,
       oob_score=False, random_state=50, verbose=0, warm_start=False)
```

Accuracy

In [13]:

```
model.score(x_test,y_test)*100
```

Out[13]:

92.37113561596323

In [14]:

```
random_forest_model_predict = model.predict(x_test)
```

Errors

In [15]:

```
print("Mean Square 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))
print('Root Mean Square
Error',np.sqrt(metrics.mean_squared_error(y_test,random_forest_model_predict)
))
```

Mean Square Error: 8.30
Mean Absolute Error: 2.09
Root Mean Square Error 2.880266201318131

In [16]:

```
df.columns
```

Out[16]:

```
Index(['Status', 'Life expectancy ', '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'],
      dtype='object')
```

In [17]:

```
y=pd.DataFrame({'Status':[0],'Adult Mortality':[263], 'infant deaths':[62],
'Alcohol':[0.01],
      'percentage expenditure':[71.27962362], 'Hepatitis B':[65], 'Measles
':[1154], ' BMI ':[19.1],
      'under-five deaths ':[83], 'Polio':[6], 'Total expenditure':[8.16],
'Diphtheria ':[65],
      'HIV/AIDS':[0.1], 'GDP':[584.25921], 'Population':[33736494], '
thinness  1-19 years':[17.2],
      ' thinness 5-9 years':[17.3], 'Income composition of
resources':[0.479], 'Schooling':[10.1],})
prediction=model.predict(y)
print(prediction)
```

[59.63640625]

In [18]:

```
df=pd.DataFrame({'Actual':y_test, 'Predicted':random_forest_model_predict})
print(df.head(25))
```

Country	Actual	Predicted
Samoa	73.618750	71.966717
Lithuania	72.806250	75.238750
Uganda	55.706250	55.029844
Congo	59.043750	55.775000
Iceland	82.443750	80.855000
Cyprus	79.675000	78.035156
United Kingdom of Great Britain and Northern Ir...	80.793750	76.857969
Botswana	56.050000	59.549844
Mauritania	62.800000	61.678278
Tuvalu	69.224932	67.423098
Angola	49.018750	55.742969
Ecuador	74.725000	73.239375
Slovakia	74.750000	73.642500
Democratic Republic of the Congo	55.687500	56.336406
Benin	57.568750	57.923281
Uruguay	76.075000	73.860155
Burundi	55.537500	58.810000
Saint Lucia	73.600000	72.671094
Togo	56.662500	56.412500
Guinea	56.012500	56.667969
Dominican Republic	72.343750	70.471250
Libya	72.487500	71.854219
Greece	81.218750	79.583438
Germany	81.175000	81.507656
India	65.418750	68.767969

Creation Of Endpoints

In [19]:

```
!pip install watson-machine-learning-client
```

Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python36/lib/python3.6/site-packages (1.0.376)
Requirement already satisfied: tabulate in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (0.8.2)
Requirement already satisfied: pandas in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (0.24.1)
Requirement already satisfied: certifi in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (2020.4.5.1)
Requirement already satisfied: requests in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (2.21.0)
Requirement already satisfied: tqdm in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (4.31.1)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (2.4.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-client) (1.24.1)
Requirement already satisfied: lomond in /opt/conda/envs/Python36/lib/python3.6/site-packages

(from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: python-dateutil>=2.5.0 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
pandas->watson-machine-learning-client) (2.7.5)
Requirement already satisfied: pytz>=2011k in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
pandas->watson-machine-learning-client) (2018.9)
Requirement already satisfied: numpy>=1.12.0 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
pandas->watson-machine-learning-client) (1.15.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
requests->watson-machine-learning-client) (2.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
requests->watson-machine-learning-client) (3.0.4)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.*,>=2.0.0 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
ibm-cos-sdk->watson-machine-learning-client) (2.4.3)
Requirement already satisfied: ibm-cos-sdk-core==2.*,>=2.0.0 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
ibm-cos-sdk->watson-machine-learning-client) (2.4.3)
Requirement already satisfied: six>=1.10.0 in /opt/conda/envs/Python36/lib/python3.6/site-packages
(from lomond->watson-machine-learning-client) (1.12.0)
Requirement already satisfied: docutils>=0.10 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
ibm-cos-sdk-core==2.*,>=2.0.0->ibm-cos-sdk->watson-machine-learning-client) (0.14)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in
/opt/conda/envs/Python36/lib/python3.6/site-packages (from
ibm-cos-sdk-core==2.*,>=2.0.0->ibm-cos-sdk->watson-machine-learning-client) (0.9.3)

In [20]:

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

2020-06-14 18:36:38,466 - watson_machine_learning_client.metanames - WARNING -
'AUTHOR_EMAIL' meta prop is deprecated. It will be ignored.

In [21]:

```
wml_credentials={
 "apikey": "rxDQ00TQGVgqslBaz1sBG0-0Bo2Hulh4C3YSCb79KEk9",
 "instance_id": "4cc19d10-b8b5-4381-b60e-702137e18431",
 "url": "https://eu-gb.ml.cloud.ibm.com"
}
```

In [22]:

```
client=WatsonMachineLearningAPIClient(wml_credentials)
```

In [23]:

```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Ramij",
               client.repository.ModelMetaNames.AUTHOR_EMAIL:
"ramijnalpur@gmail.com",
               client.repository.ModelMetaNames.NAME: "lifeExpectancy"}
```

In [24]:

```
model_artifact =client.repository.store_model(model, meta_props=model_props)
```

In [35]:

```
model_artifact
```

Out[35]:

{'metadata': {'guid': '6b7ba03c-3b96-416b-aee0-b1785b49b0df',
  'url':
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/4cc19d10-b8b5-4381-b60e-702137e18431/published_models/6b7ba03c-3b96-416b-aee0-b1785b49b0df',
  'created_at': '2020-06-14T18:36:38.666Z',
  'modified_at': '2020-06-14T18:36:38.717Z'},
 'entity': {'runtime_environment': 'python-3.6',
  'learning_configuration_url':
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/4cc19d10-b8b5-4381-b60e-702137e18431/published_models/6b7ba03c-3b96-416b-aee0-b1785b49b0df/learning_configuration',
  'author': {'name': 'Ramij'},
  'name': 'lifeExpectancy',
  'learning_iterations_url':
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/4cc19d10-b8b5-4381-b60e-702137e18431/published_models/6b7ba03c-3b96-416b-aee0-b1785b49b0df/learning_iterations',
  'feedback_url':
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/4cc19d10-b8b5-4381-b60e-702137e18431/published_models/6b7ba03c-3b96-416b-aee0-b1785b49b0df/feedback',
  'latest_version': {'url':
'https://eu-gb.ml.cloud.ibm.com/v3/ml_assets/models/6b7ba03c-3b96-416b-aee0-b1785b49b0df/versions/cd786cca-7da1-49fb-ba3b-5cc37b7eb56e',
   'guid': 'cd786cca-7da1-49fb-ba3b-5cc37b7eb56e',
   'created_at': '2020-06-14T18:36:38.717Z'},
  'model_type': 'scikit-learn-0.20',
  'deployments': {'count': 0,
   'url':
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/4cc19d10-b8b5-4381-b60e-702137e18431/published_models/6b7ba03c-3b96-416b-aee0-b1785b49b0df/deployments'},
  'evaluation_metrics_url':
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/4cc19d10-b8b5-4381-b60e-702137e18431/published_models/6b7ba03c-3b96-416b-aee0-b1785b49b0df/evaluation_metrics'}}

In [36]:

```
guid = client.repository.get_model_uid(model_artifact)
```

In [37]:

```
guid
```

Out[37]:

'6b7ba03c-3b96-416b-aee0-b1785b49b0df'

In [38]:

```
deployment = client.deployments.create(guid, name="lifeExpectancy")
```

```
########################################################################################
#######
```

Synchronous deployment creation for uid: '6b7ba03c-3b96-416b-aee0-b1785b49b0df' started

```
########################################################################################
#######
```

INITIALIZING
DEPLOY_SUCCESS

Successfully finished deployment creation,
deployment_uid='e4ce9e47-ba0b-48ff-b95a-f71c7a81b436'

In [39]:

```
client.deployments.list()
```

```
------------------------------------ ------------- ------ ------------- ---------------------- ----------------
------------
GUID                                 NAME          TYPE   STATE         CREATED
FRAMEWORK        ARTIFACT TYPE
e4ce9e47-ba0b-48ff-b95a-f71c7a81b436  lifeExpectancy  online  DEPLOY_SUCCESS
2020-06-14T18:38:33.572Z  scikit-learn-0.20  model
------------------------------------ ------------- ------ ------------- ---------------------- ----------------
------------
```

In [41]:

```
#client.deployments.delete('5f972903-dc04-4180-95bf-68796188fa40')
```

In [42]:

```
scoring_endpoint = client.deployments.get_scoring_url(deployment)
```

In [43]:

```
scoring_endpoint
```

Out[43]:

'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/4cc19d10-b8b5-4381-b60e-702137e18431/deployments/e4ce9e47-ba0b-48ff-b95a-f71c7a81b436/online'

In [ ]:

```
```