INTRODUCTION

Overview : The project relies on accuracy of data. The Global Health Observatory (GHO) data repository under World Health Organization (WHO) keeps track of the health status as well as many other related factors for all countries The data-sets are made available to public for the purpose of health data analysis. The data-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation website. Among all categories of health-related factors only those critical factors were chosen which are more representative. It has been observed that in the past 15 years , there has been a huge development in health sector resulting in improvement of human mortality rates especially in the developing nations in comparison to the past 30 years. Therefore, in this project we have considered data from year 2000-2015 for 193 countries for further analysis. In this project we will develop an Machine Learning algorithm to predict Life Expectancy. We will try to develop an accurate and more precised algorithm . We will use regression techniques such as Extra Tree Regressor and Linear Regressor to predict Life Expectancy.

Purpose : Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that affect of immunization and human development index was not taken into account in the past. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries. Hence, this gives motivation to resolve both the factors stated previously by formulating a regression model based on mixed effects model and multiple linear regression while considering data from a period of 2000 to 2015 for all the countries. Important immunization like Hepatitis B, Polio and Diphtheria will also be considered. In a nutshell, this study will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. Since the observations this dataset are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

Literature Survey

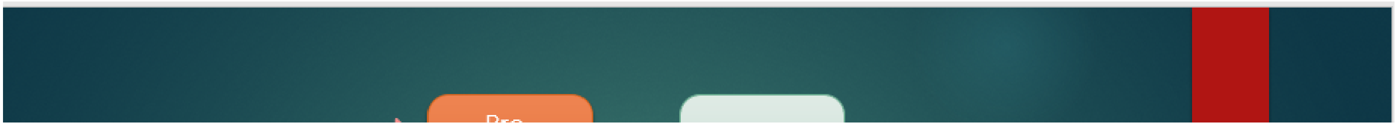
Existing Problem : To Predict Life Expectancy of a person using the WHO Dataset. We need to develop a Machine Learning Model to give more accurate results. We need to develop a web service also where we could get the results.

1. Does various predicting factors which has been chosen initially really affect the Life expectancy? What are the predicting variables actually affecting the life expectancy?
2. Should a country having a lower life expectancy value(<65) increase its healthcare expenditure in order to improve its average lifespan?
3. How does Infant and Adult mortality rates affect life expectancy?
4. Does Life Expectancy has positive or negative correlation with eating habits, lifestyle, exercise, smoking, drinking alcohol etc.
5. What is the impact of schooling on the lifespan of humans?
6. Does Life Expectancy have positive or negative relationship with drinking alcohol?
7. Do densely populated countries tend to have lower life expectancy?
8. What is the impact of Immunization coverage on life Expectancy?

Proposed Solution : A Machine Learning Model developed using Extra Tree Regressor Technique to predict Life Expectancy. Node-Red is used to create a Web - Service. We are using correlation technique to predict the impacts of different factors like Schooling, Immunization coverage, Infant Deaths, GDP, Population, diseases like Polio, HIV/AIDS etc, Habits such as Exercise, Alcohol consumption etc. effect of expenditure on Life Expectancy .We will use heatmap and horizontal bar plot to understand and correlate the effect of different factors.

THEORITICAL ANALYSIS

BLOCK DIAGRAM :



Software Design :

- Dataset Collection
- ML Notebook Creation
- Machine Learning Model Development
- Machine Learning Model Deployment
- Node-Red Flow Development
- Project Completion and Testing

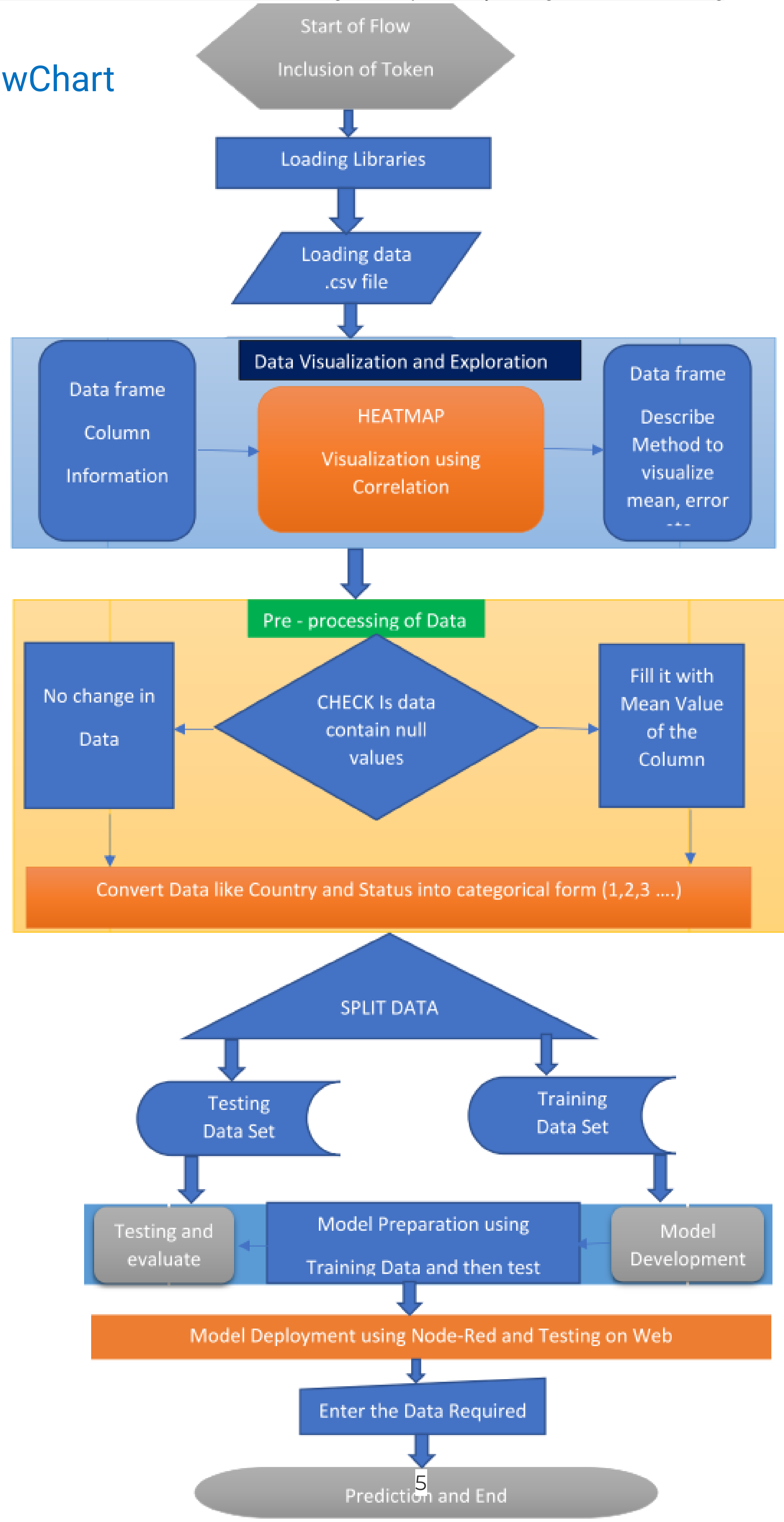
EXPERIMENTAL INVESTIGATIONS

The Correlation between Life Expectancy and these factors shows that :

Population	**0.021538**
Measles	**0.157586**
Year	**0.170033**
infant deaths	**0.196557**
Total expenditure	**0.218086**
under-five deaths	**0.222529**
Hepatitis B	**0.256762**
percentage expenditure	**0.381864**
Alcohol	**0.404877**
GDP	**0.461455**
Polio	**0.465556**
thinness 5-9 years	**0.471584**
thinness 1-19 years	**0.477183**
Diphtheria	**0.479495**
HIV/AIDS	**0.556556**
BMI	**0.567694**
Adult Mortality	**0.696359**
Income composition of resources	**0.724776**
Schooling	**0.751975**

It shows that Schooling affects the life expectancy most, and furthur income composition of resources and adult mortality also contribute majorly to the life expectancy, hence we should focus on these factor and improve it to increase life expectancy.

FlowChart


Start of Flow
Inclusion of Token
Loading Libraries
Loading data
.csv file
Data Visualization and Exploration
Data frame
Column
Information
HEATMAP
Visualization using
Correlation
Data frame
Describe
Method to
visualize
mean, error
Pre - processing of Data
No change in
Data
CHECK Is data
contain null
values
Fill it with
Mean Value
of the
Column
Convert Data like Country and Status into categorical form (1,2,3 ....)
SPLIT DATA
Testing
Data Set
Training
Data Set
Testing and
evaluate
Model Preparation using
Training Data and then test
Model
Development
Model Deployment using Node-Red and Testing on Web
Enter the Data Required
Prediction and End

RESULT

Input in JSON:

{"fields":["Country","Year","Status","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"],"values":[[1, 2015, 2, 263.0, 62, 0.01, 71.27962362, 65.0, 1154, 19.1, 83, 6.0, 8.16, 65.0, 0.1, 584.2592099999999, 33736494.0, 17.2, 17.3, 0.479, 10.1]]}

Output in JSON:

{
 " fields " : [
  " prediction "
  ],
" values " : [
 [
64.39600000000002
]
]
}

Metrics Score :

MAE: 95.53635041448281
MSE: 2.7931830132770177
RMSE: 1.6712818473486206
R2 Score: 96.77244159342855

Predicting Life Expectancy

ADVANTAGES

- Helps the government and other organizations to make policies
- Helps in predicting other key features related to a country like happiness index and other indexes.
- Helps in demontrating a model for every nation to use this information and increase their citizens life expectancy.

DISADVANTAGES

- The model may not accurate enough to predict one countries life expectancy it my fail for specific countries.
- Model may get widely affected due to disasters and may get fail.
- Life Expectancy is very versatile, and it may be affected for an individual easily and hence it will not be a good measure for individual life expectancy

APPLICATIONS

1. Prediction of Life Expectancy help in making more health awareness among the people.

2. It helps the health service providers and government to make policy which is more suitable as per the situation.

3. Awaring people to how much invest to mantain their fitness and health .

4. Awaring about the risk like diseases like Hepatitis B, Measles etc.

5. Helping the government to know what others factor effect the life expectancy.

6. Helps the organisations to develop countries features like happiness index, human development index etc.

FUTURE SCOPE

- It will help us to increase average life expectancy of a country by modifying some factors.

- It will tell us in which sector we should invest more to increase life expectancy .

- Increased life expectancy also add major improvements in country like economy.

- Life expectancy will be a major issue in future and a good factor to show which countries people are more happy.

- It will also decide which country is more favourable and good to live.

BIBILOGRAPY

References:

1. IBM Cloud Platform
2. SmartInternz Platform
3. Slack and Github Platform
4. Coursera Machine Learning
5. Youtube
6. towardsdatascience.com
7. kaggle

Software :

1. Node-Red (https://nodered.org/)
2. Jupyter Notebook
3. IBM Watson and Machine Larning Services

APPENDIX

Source Code

```
# @hidden_cell
# The project token is an authorization token that is used to access project resources like data sources, connections, and used by platform APIs.
from project_lib import Project
project = Project(project_id='93798796-b8aa-441b-bdd7-4068bf30ee63', project_access_token='p-2b49d4d9fb2792e4bebe71785d22609b14103ee9')
pc = project.project_context
```

Regression Model To Predict Life Expectancy

Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that affect of immunization and human development index was not taken into account in the past. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries. Hence, this gives motivation to resolve both the factors stated previously by formulating a regression model based on mixed effects model and multiple linear regression while considering data from a period of 2000 to 2015 for all the countries. Important immunization like Hepatitis B, Polio and Diphtheria will also be considered. In a nutshell, this study will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. Since the observations this dataset are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

Loading Libraries

In [2]:

```
import numpy as np
from sklearn.metrics import mean_squared_error, r2_score
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import ExtraTreesRegressor
from sklearn import metrics
%matplotlib inline
```

Exploring and Analysing Data

In [3]:

```
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_ad8f9cbb843d44eaa4524195e1c21af0 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='KSfszhHlPjdWf4xAhvFtuOQlxvsVZufSniLcw3uHsgtz',
    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_ad8f9cbb843d44eaa4524195e1c21af0.get_object(Bucket='mllifeexpectancy-donotdelete-pr-e86ccge4xyqv6m',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 )

df = pd.read_csv(body)
df.head()
```

Out[3]:

	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
0	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
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
2	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
3	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
4	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

5 rows × 22 columns

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
Country                    2938 non-null object
Year                       2938 non-null int64
Status                     2938 non-null object
Life expectancy            2928 non-null float64
Adult Mortality            2928 non-null float64
infant deaths              2938 non-null int64
Alcohol                    2744 non-null float64
percentage expenditure     2938 non-null float64
Hepatitis B                2385 non-null float64
Measles                    2938 non-null int64
 BMI                       2904 non-null float64
under-five deaths          2938 non-null int64
Polio                      2919 non-null float64
Total expenditure          2712 non-null float64
Diphtheria                 2919 non-null float64
 HIV/AIDS                  2938 non-null float64
GDP                        2490 non-null float64
Population                 2286 non-null float64
 thinness  1-19 years      2904 non-null float64
 thinness 5-9 years        2904 non-null float64
Income composition of resources    2771 non-null float64
Schooling                  2775 non-null float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.0+ KB
```

In [5]:

```
df.describe()
```

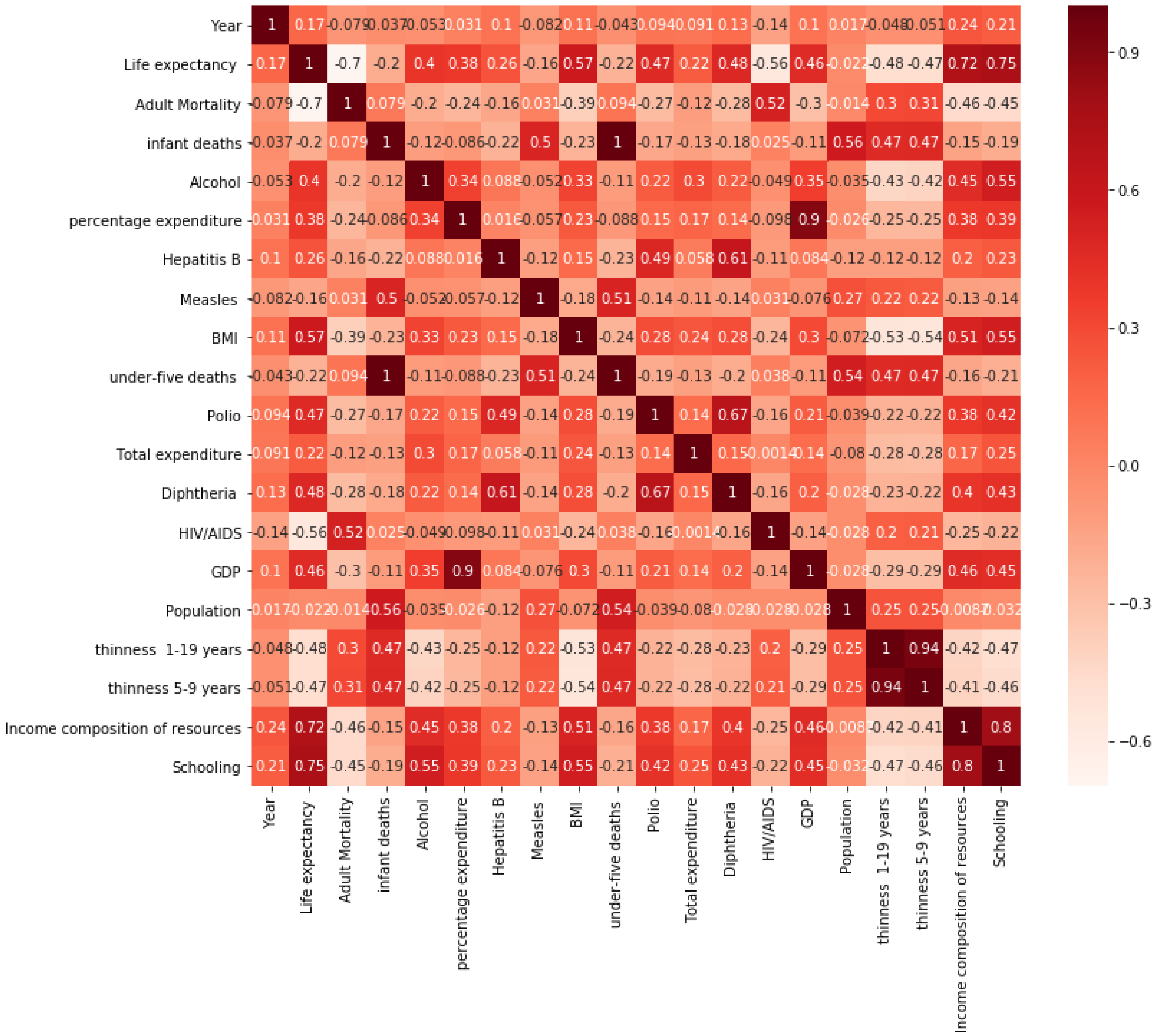
Out[5]:

	Year	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
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000	2938.000000	2904.000000	2938.000000	2919.000000	2712.000000	2919.000000	2938.000000	2490.000000	2.286000e+03	2904.000000	2904.000000	2771.000000	2775.000000
	200	69.	164	30.	4.6	738	80.	241	38.	42.	82.	5.9	82.	1.7	748	1.2	4.8	4.8	0.6	11.

mean	7.518720	224932	.796448	303948	02861	.251295	940461	9.592240	321247	035739	550188	3819	324084	42103	3.158469	75338e+07	39704	70317	27551	992793
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016	11467.272489	20.044034	160.445548	23.428046	2.49832	23.716912	5.077785	14270.169342	6.101210e+07	4.420195	4.508882	0.210904	3.358920
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000	3.000000	0.37000	2.000000	0.100000	1.681350	3.400000e+01	0.100000	0.100000	0.000000	0.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000	0.000000	19.300000	0.000000	78.000000	4.26000	78.000000	0.100000	463.935626	1.957932e+05	1.600000	1.500000	0.493000	10.100000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000	17.000000	43.500000	4.000000	93.000000	5.75500	93.000000	0.100000	1766.947595	1.386542e+06	3.300000	3.300000	0.677000	12.300000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000	360.250000	56.200000	28.000000	97.000000	7.49250	97.000000	0.800000	5910.806335	7.420359e+06	7.200000	7.200000	0.779000	14.300000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2500.000000	99.000000	17.600000	99.000000	50.600000	119172.741800	1.293859e+09	27.700000	28.600000	0.948000	20.700000

In [6]:

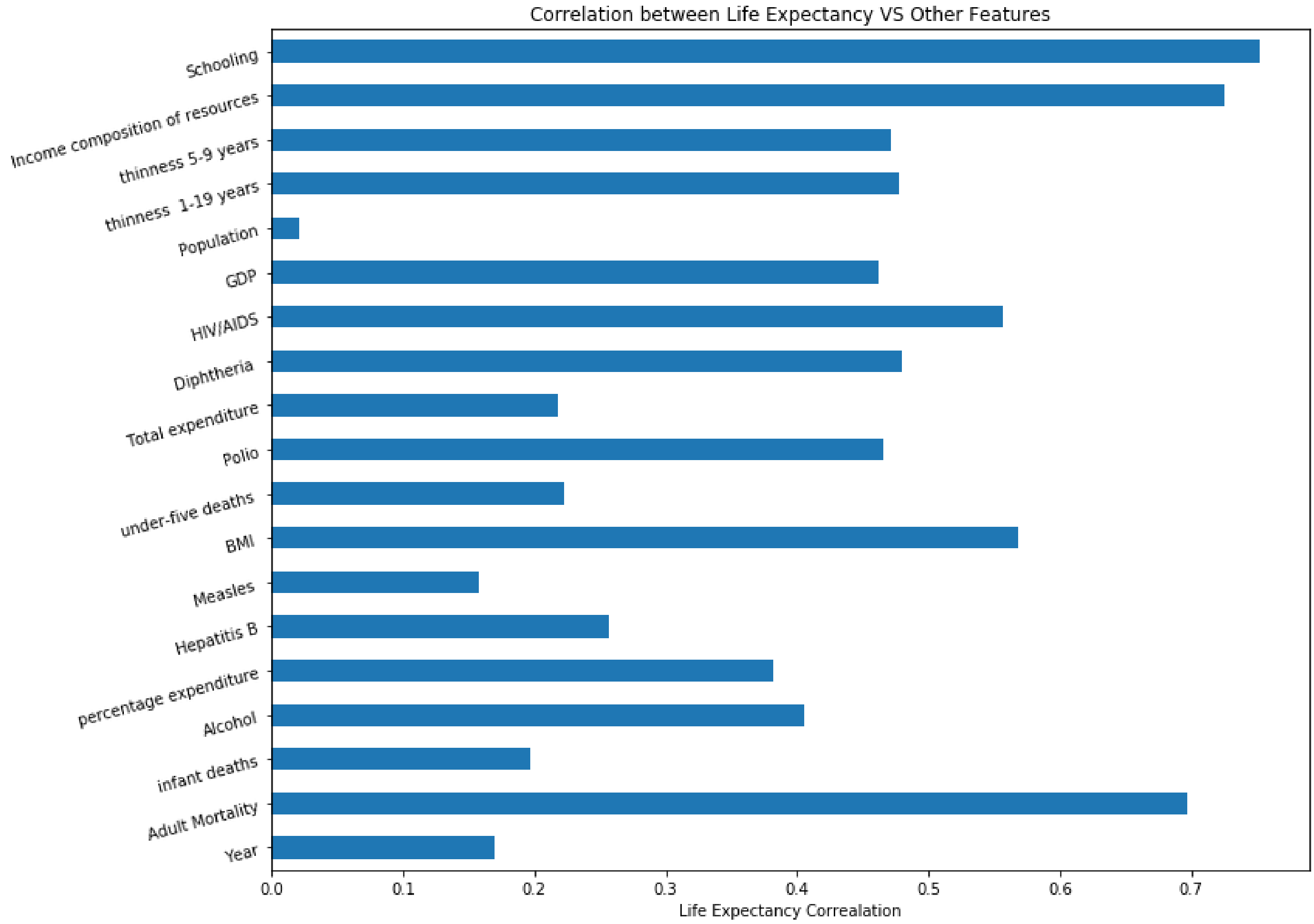
```
plt.figure(figsize=(12,10))
cor = df.corr()
sns.heatmap(cor, annot=True, cmap=plt.cm.Reds)
plt.show()
```



Preprocessing Data

In [7]:

```
#Correlation with output variable
plt.figure(figsize=(12,10))
cor_target = abs(cor["Life expectancy "])
#Selecting highly correlated features
relevant_features = cor_target[cor_target < 1]
relevant_features.plot.barh(rot=15, title="Correlation between Life Expectancy VS Other Features");
plt.xlabel("Life Expectancy Correalation")
plt.show(block=True);
```



In [8]:

```
df.columns
```

Out[8]:

```
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')
```

In [9]:

```
df['Country'] = df['Country'].replace(['Afghanistan' , 'Albania' , 'Algeria' , 'Angola' , 'Antigua and Barbuda' , 'Argentina' ,
'Armenia' , 'Australia' , 'Austria' , 'Azerbaijan' , 'Bahamas' , 'Bahrain' , 'Bangladesh' , 'Barbados' , 'Belarus' , 'Belgium' ,
'Belize' , 'Benin' , 'Bhutan' , 'Bolivia (Plurinational State of)' , 'Bosnia and Herzegovina' , 'Botswana' , 'Brazil' , 'Brunei
Darussalam' , 'Bulgaria' , 'Burkina Faso' , 'Burundi' , "Côte d'Ivoire" , 'Cabo Verde' , 'Cambodia' , 'Cameroon' ,
'Canada' , 'Central African Republic' , 'Chad' , 'Chile' , 'China' , 'Colombia' , 'Comoros' , 'Congo' , 'Costa Rica' ,
'Croatia' , 'Cuba' , 'Cyprus' , 'Czechia' , "Democratic People's Republic of Korea" , 'Democratic Republic of the
Congo' , 'Denmark' , 'Djibouti' , 'Dominican Republic' , 'Ecuador' , 'Egypt' , 'El Salvador' , 'Equatorial Guinea' , 'Eritrea'
, 'Estonia' , 'Ethiopia' , 'Fiji' , 'Finland' , 'France' , 'Gabon' , 'Gambia' , 'Georgia' , 'Germany' , 'Ghana' , 'Greece' ,
'Grenada' , 'Guatemala' , 'Guinea' , 'Guinea-Bissau' , 'Guyana' , 'Haiti' , 'Honduras' , 'Hungary' , 'Iceland' , 'India' ,
'Indonesia' , 'Iran (Islamic Republic of)' , 'Iraq' , 'Ireland' , 'Israel' , 'Italy' , 'Jamaica' , 'Japan' , 'Jordan' , 'Kazakhstan' ,
'Kenya' , 'Kiribati' , 'Kuwait' , 'Kyrgyzstan' , "Lao People's Democratic Republic" , 'Latvia' , 'Lebanon' , 'Lesotho' ,
```

```
'Liberia' , 'Libya' , 'Lithuania' , 'Luxembourg' , 'Madagascar' , 'Malawi' , 'Malaysia' , 'Maldives' , 'Mali' , 'Malta' ,
'Mauritania' , 'Mauritius' , 'Mexico' , 'Micronesia (Federated States of)' , 'Mongolia' , 'Montenegro' , 'Morocco' ,
'Mozambique' , 'Myanmar' , 'Namibia' , 'Nepal' , 'Netherlands' , 'New Zealand' , 'Nicaragua' , 'Niger' , 'Nigeria' ,
'Norway' , 'Oman' , 'Pakistan' , 'Panama' , 'Papua New Guinea' , 'Paraguay' , 'Peru' , 'Philippines' , 'Poland' , 'Portugal'
, 'Qatar' , 'Republic of Korea' , 'Republic of Moldova' , 'Romania' , 'Russian Federation' , 'Rwanda' , 'Saint Lucia' ,
'Saint Vincent and the Grenadines' , 'Samoa' , 'Sao Tome and Principe' , 'Saudi Arabia' , 'Senegal' , 'Serbia' ,
'Seychelles' , 'Sierra Leone' , 'Singapore' , 'Slovakia' , 'Slovenia' , 'Solomon Islands' , 'Somalia' , 'South Africa' , 'South
Sudan' , 'Spain' , 'Sri Lanka' , 'Sudan' , 'Suriname' , 'Swaziland' , 'Sweden' , 'Switzerland' , 'Syrian Arab Republic' ,
'Tajikistan' , 'Thailand' , 'The former Yugoslav republic of Macedonia' , 'Timor-Leste' , 'Togo' , 'Tonga' , 'Trinidad and
Tobago' , 'Tunisia' , 'Turkey' , 'Turkmenistan' , 'Uganda' , 'Ukraine' , 'United Arab Emirates' , 'United Kingdom of
Great Britain and Northern Ireland' , 'United Republic of Tanzania' , 'United States of America' , 'Uruguay' ,
'Uzbekistan' , 'Vanuatu' , 'Venezuela (Bolivarian Republic of)' , 'Viet Nam' , 'Yemen' , 'Zambia' , 'Zimbabwe' , 'Cook
Islands' , 'Dominica' , 'Marshall Islands' , 'Monaco' , 'Nauru' , 'Niue' , 'Palau' , 'Saint Kitts and Nevis' , 'San Marino' ,
'Tuvalu'], [1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17 ,18 ,19 ,20 ,21 ,22 ,23 ,24 ,25 ,26 ,27 ,28 ,29 ,30 ,31 ,32
,33 ,34 ,35 ,36 ,37 ,38 ,39 ,40 ,41 ,42 ,43 ,44 ,45 ,46 ,47 ,48 ,49 ,50 ,51 ,52 ,53 ,54 ,55 ,56 ,57 ,58 ,59 ,60 ,61 ,62 ,63
,64 ,65 ,66 ,67 ,68 ,69 ,70 ,71 ,72 ,73 ,74 ,75 ,76 ,77 ,78 ,79 ,80 ,81 ,82 ,83 ,84 ,85 ,86 ,87 ,88 ,89 ,90 ,91 ,92 ,93 ,94
,95 ,96 ,97 ,98 ,99 ,100 ,101 ,102 ,103 ,104 ,105 ,106 ,107 ,108 ,109 ,110 ,111 ,112 ,113 ,114 ,115 ,116 ,117 ,118
,119 ,120 ,121 ,122 ,123 ,124 ,125 ,126 ,127 ,128 ,129 ,130 ,131 ,132 ,133 ,134 ,135 ,136 ,137 ,138 ,139 ,140 ,141
,142 ,143 ,144 ,145 ,146 ,147 ,148 ,149 ,150 ,151 ,152 ,153 ,154 ,155 ,156 ,157 ,158 ,159 ,160 ,161 ,162 ,163 ,164
,165 ,166 ,167 ,168 ,169 ,170 ,171 ,172 ,173 ,174 ,175 ,176 ,177 ,178 ,179 ,180 ,181 ,182 ,183 ,184 ,185 ,186 ,187
,188 ,189 ,190 ,191 ,192 ,193])
```

In [10]:

```
df['Status'] = df['Status'].replace(['Developing', 'Developed'],[1, 2])
```

In [11]:

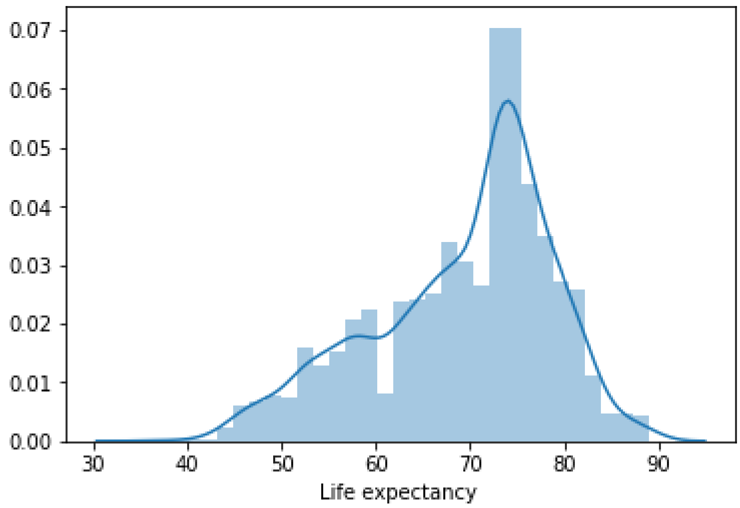
```
df = df.fillna(df.mean())
```

In [12]:

```
sns.distplot(df['Life expectancy '])
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2c96b00dd8>
```



Splitting the data into training and testing Sets

In [13]:

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(df, test_size=0.3, random_state=111)
```

In [14]:

```
train_x = train.loc[:, train.columns != "Life expectancy "]
test_x = test.loc[:, test.columns != "Life expectancy "]
train_y = train["Life expectancy "]
test_y = test["Life expectancy "]
```

Fitting the data to Extra Tree regression model

In [15]:

```
model = ExtraTreesRegressor(n_estimators = 50)
model.fit(train_x, train_y)
```

Out[15]:

```
ExtraTreesRegressor(bootstrap=False, 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=50, n_jobs=None,
          oob_score=False, random_state=None, verbose=0, warm_start=False)
```

Predicting the Test Set Target Variable

In [16]:

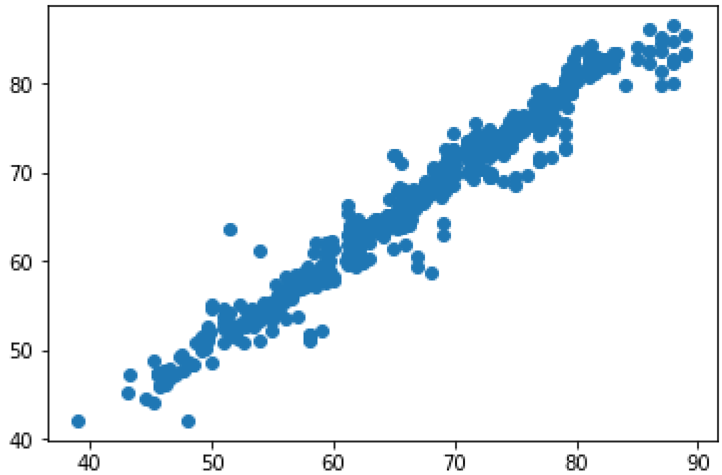
```
test_pred = model.predict(test_x)
```

In [17]:

```
plt.scatter(test_y, test_pred)
```

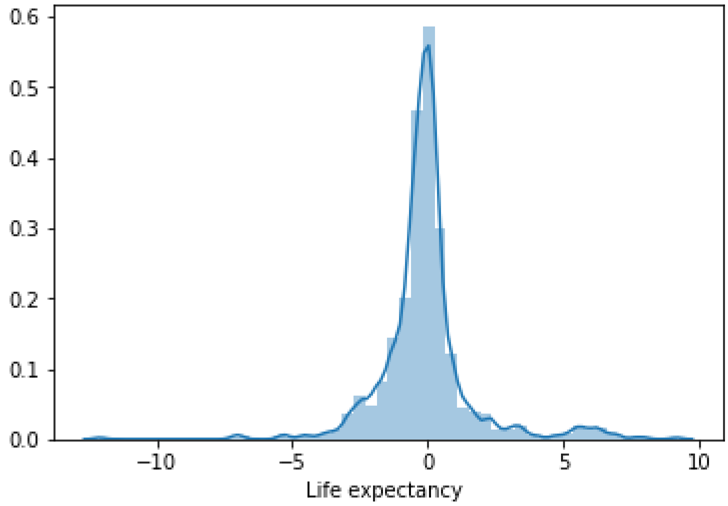
Out[17]:

```
<matplotlib.collections.PathCollection at 0x7f2c96af3f28>
```



In [18]:

```
sns.distplot((test_y-test_pred),bins=50);
```



Analysing the Results

In [19]:

```
print('MAE: ', metrics.mean_absolute_error(test_y, test_pred))
print('MSE: ', metrics.mean_squared_error(test_y, test_pred))
print('RMSE: ', np.sqrt(metrics.mean_squared_error(test_y, test_pred)))
print('R2 Score: ',r2_score(test_y, test_pred)*100)
```

```
MAE:  1.0298079888603913
MSE:  3.061130363078195
RMSE:  1.7496086314025188
R2 Score:  96.59734744887703
```

In [20]:

```
!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: tqdm in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-Client) (4.31.1)
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: requests in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-Client) (2.21.0)
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: 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: certifi in /opt/conda/envs/Python36/lib/python3.6/site-packages (from watson-machine-learning-Client) (2020.4.5.1)
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: 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: 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: 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-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: 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: 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: 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)
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)

In [21]:

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

2020-06-04 18:05:25,743 - watson_machine_learning_client.metanames - WARNING - 'AUTHOR_EMAIL' meta prop is deprecated. It will be ignored.

In [22]:

```
wml_credentials = {
  "apikey": "IuO-M0sxOMI3cUzxsuZ0frC6-5p13t-Gvye1UWTk3ZTF",
  "iam_apikey_description": "Auto-generated for key 711e6ff7-d99f-47de-a8ce-aafe1fbf8eec",
  "iam_apikey_name": "Service credentials-1",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
  "iam_serviceid_crn":
"crn:v1:bluemix:public:iam-identity::a/f20187c41bc747bca9ea0a12092289e5::serviceid:ServiceId-2916bfab-475f-424d-a999-e0e864061652",
  "instance_id": "e33d8820-4ba4-45f7-9ce8-a7ce67837b49",
  "url": "https://us-south.ml.cloud.ibm.com"
}
```

In [23]:

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

In [24]:

```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Tej Pratap",
           client.repository.ModelMetaNames.AUTHOR_EMAIL: "tejpartap957@gmail.com",
           client.repository.ModelMetaNames.NAME: "Life Expectancy using Extra Tree Regressor"}
```

In [25]:

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

In [26]:

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

In [27]:

```
published_model_uid
```

Out[27]:

'9ea92f16-a214-4771-863e-06c4ef2a51a3'

In [28]:

```
deployment = client.deployments.create(published_model_uid, name="Life Expectancy Prediction using ExtraTreeRegressor")
```

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

Synchronous deployment creation for uid: '9ea92f16-a214-4771-863e-06c4ef2a51a3' started

#######################################################################################


INITIALIZING
DEPLOY_SUCCESS


------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='7e97c3d4-2aa6-48a2-89e0-1401624e391b'
------------------------------------------------------------------------------------------------
```

In [29]:

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

In [30]:

```
scoring_endpoint
```

Out[30]:

'https://us-south.ml.cloud.ibm.com/v3/wml_instances/e33d8820-4ba4-45f7-9ce8-a7ce67837b49/deployments/7e97c3d4-2aa6-48a2-89e0-1401624e391b/online'

In [ ]: