

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

PROJECT LINK:

<https://node-red-jrhkp.eu-gb.mybluemix.net/ui/#!/0?socketid=spWAnKBt0nU2QGJFAAAA>

Created by

KANAUJ GUHARROY

INTRODUCTION

A prediction of Life Expectancy helps to analyze the average life span and thus constitute in making life decisions for the generations to come easier.

Life expectancy is one of the most important factors in end-of-life decision making. Good prognostication for example helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more. By using supervised machine learning techniques, one can extract a model that will be able to predict the life expectancy of future years. One method of approach is to use LSTM models to achieve this task.

The project tries to create a model based on data provided by the World Health Organization (WHO) to evaluate the life expectancy for different countries in years. The data offers a timeframe from 2000 to 2015. The output algorithms have been used to test if they can maintain their accuracy in predicting the life expectancy for data they haven't been trained. Following algorithms can be used:

- ✚ Linear Regression
- ✚ Ridge Regression
- ✚ Lasso Regression
- ✚ Elastic Net Regression
- ✚ Linear Regression with Polynomic features
- ✚ Decision Tree Regression
- ✚ Random Forest Regression

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.

Overview

Predicting Life Expectancy using Machine Learning aims to predict the lifespan of a human, based on a very diverse dataset, in a demographic region. The life of a human depends on various factors such as Regional variations, Economic Circumstances, Sex Differences, Mental Illnesses, Physical Illnesses, Education, Year of their birth and other demographic factors. The project aims to predict an average life expectancy based on these and several other factors. This project finds the expected solution using various machine learning algorithms such as:

- ✚ SVM
- ✚ Linear Regression
- ✚ Logistic Regression
- ✚ Polynomic Regression
- ✚ Clustering

The aim of the project is to find the relationship of the various factors with the lifespan of an individual using the ML Algorithms mentioned above.

Purpose

If life expectancy is longer in a certain country, it shows a lot about the conditions of that place. It speaks lengths about the health care facilities as well as the quality of life in general. If the conditions in a country and its economy is good, the life expectancy becomes higher. It must be a healthy life too. A lot of people spend their life in miserable conditions. With the present advancements and developing technologies in various fields, it is more possible than ever that a long and healthy life is lived by most people.

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. 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.

LITERATURE REVIEW

Existing Problem

Very few works have been done to provide an exclusive **Life Expectancy Prediction**. We have reviewed existing works and techniques in the prediction of human Life Expectancy, and reached a conclusion that it is feasible to predict a PLE for individuals using evolving technologies and devices such as big data, AI, machine learning techniques, and PHDs, wearables and mobile health monitoring devices. We also identified that the collection of data will be a huge challenge due to the privacy and government policy considerations, which will require collaboration of various bodies in the health industry. The interworking of a heterogeneous health network is also a challenge for data collection. Despite these challenges, a possibility of a PLE prediction by proposing an approach of data collection and application by smartphone, with which users can enter their information to access the cloud server to obtain their own PLE, was shown. To verify the accuracy of PLE 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. As artificial intelligence technology is evolving and being applied rapidly, feasibility may be increasing to collect health data from the public as well as existing health agencies such as centralized health servers.

Proposed Solution

It was found that the effect 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 focuses 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.

The model of **Predicting Life Expectancy using Machine Learning** uses IBM Cloud services, which helps to avoid any storage issues. The UI Presented to the users is a website and hence they need not download any application to predict the results, which saves the storage space as that is the need of the hour.

THEORETICAL ANALYSIS

Algorithms

Machine Learning:

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to learn automatically and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that one provides. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

The machine learning algorithm is classified into Supervised, unsupervised and reinforced learning. For this project a type of supervised model is used.

Supervised Model

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training data set, the learning algorithm produces an inferred function to make predictions about the output values. The system can provide targets for any new input after sufficient training. Supervised learning problems can be further grouped into Regression and Classification problems

Classification

A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes.

Regression

A regression problem is when the output variable is a real or continuous value. Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points. In this project I am using Elastic Net Regression as it produces the least error rate among other algorithms available.

Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models. The technique combines both the lasso and ridge regression methods by learning from their shortcomings to improve on the regularization of statistical models.

The elastic net method improves on lasso's limitations, i.e., where lasso takes a few samples for high dimensional data, the elastic net procedure provides the inclusion of "n" number of variables until saturation. In a case where the variables are correlated groups, lasso tends to choose one variable from such groups and ignore the rest entirely.

To eliminate the limitations found in lasso, the elastic net includes a quadratic expression in the penalty, which, when used in isolation, becomes ridge regression. The quadratic expression in the penalty elevates the loss function toward being convex. The elastic net draws on the best of both worlds – i.e., lasso and ridge regression.

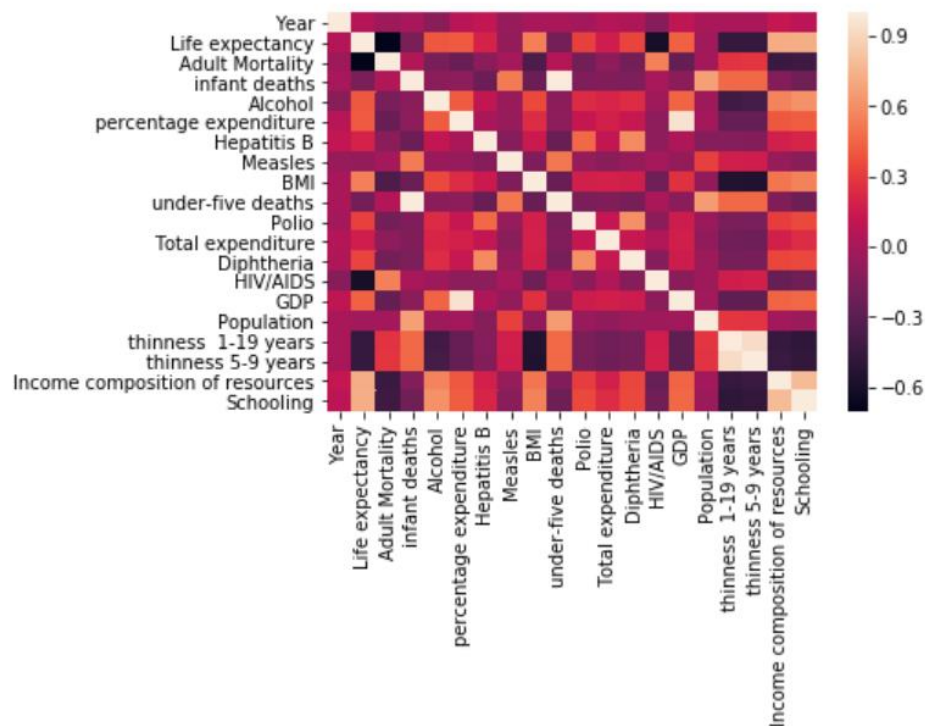
Project Model

The model was created using Watson studio and Jupyter notebook.

The data set contains 22 features.

Since regression can only use number values the data set is grouped using countries, to avoid conflict.

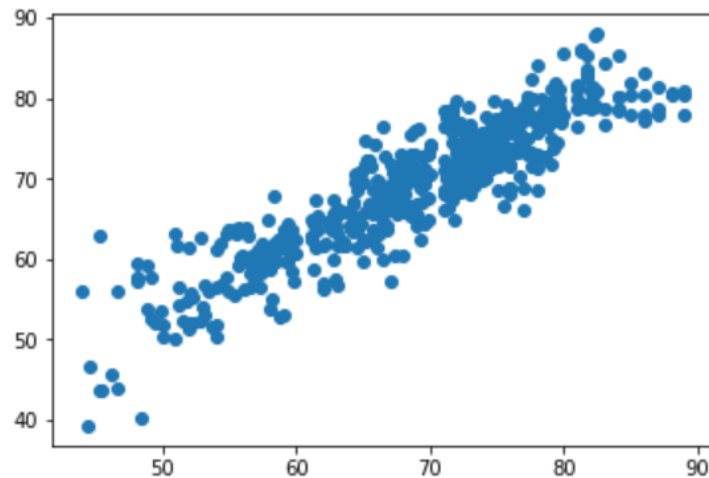
The null values were filled with their mean and the data was then split into training and testing data. The following heat map was generated from the data



After training the data using Elastic net regression the prediction models were obtained

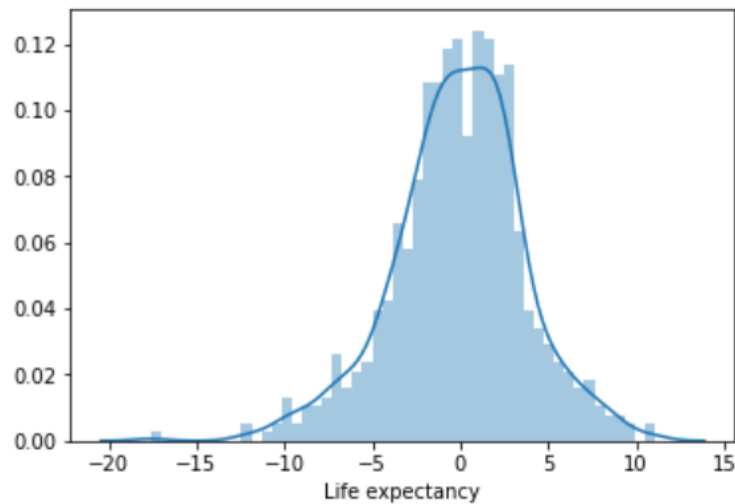
```
In [23]: plt.scatter(y_test,predictions)
```

```
Out[23]: <matplotlib.collections.PathCollection at 0x7f21ff97f9e8>
```



Residual Histogram

```
In [24]: sns.distplot((y_test-predictions),bins=50);
```



PROJECT REQUIREMENTS








This project fundamentally aims in predicting the life expectancy. The primary requirement of the project is the suitable dataset which will aid the prediction. The machine learning model is trained on the basis of the data provided, such that it could predict the average lifespan of an individual in the coming years.

The project will help in determining the Life Expectancy on the basis of the dataset from





<https://www.kaggle.com/kumarajarshi/life-expectancy-who>

The dataset will provide various information like kind of diseases leading to the deaths, etc. Thus, with the help of these information, the Life Expectancy shall be predicted.





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

-  train_test_split
-  linear_model
-  model_selection
-  metrics
-  tree
-  ensemble
-  preprocessing







Functional Requirements

-  The dataset should be preprocessed before applying prediction.
-  The data model must be created on the basis of preprocessed data.
-  The data model must then be converted into a module for further use, after the data is updated.
-  The data should be implemented using IBM Watson which should then be connected to Node Red for the User Interface.

Technical Requirements

-  The dataset must be in csv format.
-  Machine Learning Algorithms must be applied with the help of Python.
-  IBM cloud account.
-  IBM Watson and Node-Red flow.

Software Requirements

-  Python IDE
-  Excel
-  IBM Cloud
-  IBM Watson
-  IBM Machine Learning
-  IBM Node-Red Service

EXPERIMENTAL INVESTIGATION

1. Collecting the Dataset:

The most important thing for any project is collecting the data as per requirement of the model. Thus, firstly one collect the data from the given source. For the project the dataset was "Life Expectancy". The dataset was provided by the WHO in order for the analysis purpose. One have used this dataset for the prediction purpose.

2. Setting up IBM Cloud Services:

For using the various Cloud services for the project development. One must first create an IBM Cloud account. Once the account is created, one can access various services used for ML projects.

3. Creating a Watson Project:

Once the services required for the project is enabled, one can go with for the creation of the project. Watson Studio allows us to create various project using different tools like Jupyter notebook, Auto AI, R Studio etc.

Configure the Watson studio:

Once one is done with the creation of the Watson project, one can configure the various services associated with it. Also, one can look for the various tools associated with it.

4. Creating Machine Learning Services:

As one is creating the Machine Learning Model for the prediction of the Life Expectancy one must create the Machine learning services in IBM cloud which will help in building up the model.

a) Create Jupyter Notebook and Import Dataset:

Firstly in the project one need to add the Jupyter Notebook(It is the platform for developing the model and actual implementation). Once the Jupyter notebook is created one must import the data. The data set is Inserted to code in pandas data frame.

b) Choosing the appropriate Model for Prediction:

One can use any model for the prediction person and with the help of it one can train and test the dataset. For the project I have been choosing the Random Forest Regression Model for the development purpose.

The screenshot displays the IBM Watson Studio web interface. At the top, the header includes the IBM Watson Studio logo, an 'Upgrade' button, and the user's account name 'KANAUJ GUHARROY's Acco...'. Below the header, the breadcrumb navigation shows 'My projects / Predicting Life Expectancy Using ... / Predicting Life Expectancy'. The main workspace contains three sections of code:

- X and y arrays**: A code cell with In [14] showing the loading of data from a file named 'led' into two arrays, X and y. X contains various socio-economic and health indicators, and y contains 'Life expectancy'.
- Train Test Split**: A text block explaining the next step: 'Now let's split the data into a training set and a testing set. We will train our model on the training set and then use the test set to evaluate the model.' This is followed by two code cells: In [15] imports 'train_test_split' from 'sklearn.model_selection', and In [16] uses it to split the data into X_train, X_test, y_train, and y_test with a test_size of 0.4 and random_state of 101.
- Creating and Training the Model**: Three code cells follow: In [17] imports 'LinearRegression' from 'sklearn.linear_model'; In [18] creates a 'LinearRegression()' object named 'lm'; and In [19] calls 'lm.fit(X_train, y_train)'. The output of the last cell is shown as 'Out[19]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)'.

The bottom of the screenshot shows a Windows taskbar with the search bar and several application icons, including Edge, File Explorer, and Chrome. The system clock indicates 7:39 PM on 6/5/2020.

c) Deployment of Model:

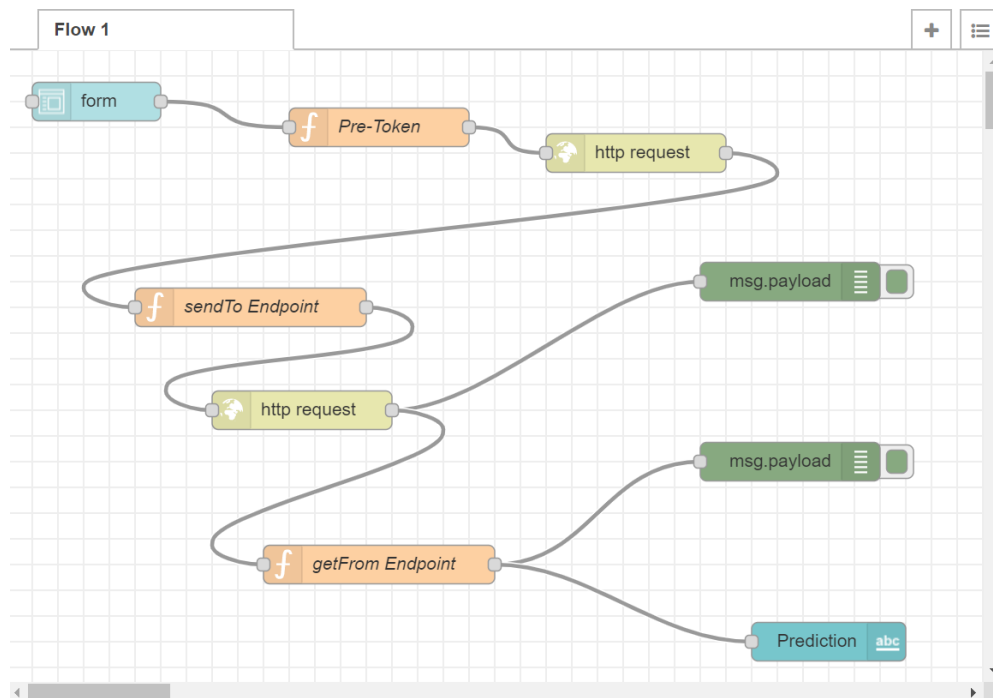
Once we're done with building the model, one must deploy the model. The deployed model will be stored in IBM Cloud Storage.

5. Create a Node-red Flow:

Once the model is deployed, one can create the node red flow to create an API for the model. The API will act as the front end to the model. From here one will get the Life Expectancy prediction data.

FLOWCHART

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.



Node-red flow

RESULT

The model appears to the user in the form of an interface as shown in the Figure 2. The user has to fill in the inputs and click on **SUBMIT** button at the end of the form. On clicking the **SUBMIT** button, the user will be displayed the predicted life expectancy, based on the inputs provided, at the top of the page as shown below.

Home Page

Machine Learning Model

Prediction **63.343463166584485**

Year *
2010

Adult Mortality *
261

Infant Deaths *
63

Alcohol *
0.01

Percentage Expenditure *
71.65675

Hepatitis B *
65

Measles *
1154

BMI *
19.1

Under-Five Deaths *
81

Polio *
9

Total Expenditure *
8.345

Diphtheria *
67

HIV/AIDS *
0.1

GDP *
584.22783

Population *
13435576

Thinness 1-19 years *
17.2

Thinness 5-9 years *
17.3

Income Composition of Resources *
0.479

Schooling *
10.1

SUBMIT

CANCEL

Result

ADVANTAGES AND DISADVANTAGES:

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.

Disadvantages:

1. Disadvantages of using IBM Watson:
 - ✚ Only in English (Limits areas of use)
 - ✚ 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.

APPLICATIONS

When will I die?

This question has endured across cultures and civilizations. It has given rise to a plethora of religions and spiritual paths over thousands of years, and more recently, some highly amusing apps. This system will be used for people wondering with such questions.

Life expectancy is the primary factor in determining an individual's risk factor and the likelihood they will to make a claim. Insurance companies consider age, lifestyle choices, family medical history, and several other factors when determining premium rates for individual life insurance policies. The principle of life expectancy suggests that you should purchase a life insurance policy for yourself and your spouse sooner rather than later. Not only will you save money through lower premium costs, but you will also have longer for your policy to accumulate value and become a potentially significant financial resource as you age.

It can be used by researchers to make meaningful researches out of it and thus, bring about something that will help increase the expectancy consider the impact of a specific factor on the average lifespan of people in a specific country. It can be used to monitor health inequalities of a country. The government can make amendments to their existing policies in order to improve their life expectancy. It can be used to analyze the factors for high life expectancy.

CONCLUSION

From the project, I conclude that the Prediction of Life Expectancy from the given dataset by working on Watson studio as the Back end and Node Red Flow as the Front end is very much possible as shown.

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, 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.

FUTURE SCOPE

In the future, we can connect the model to the database to have the record of predictions. This will help one analyze the trends in the life span. A model with country wise bifurcation can be made, which will help in the segregation of the data demographically.

APPENDIX

Source Code

```
#!/usr/bin/env python
# coding: utf-8

# ____
#
#

# In[1]:

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

```
get_ipython().run_line_magic('matplotlib', 'inline')

# ### Check out the Data

# In[2]:

import types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

led = pd.read_csv(body)
led.head()

# In[3]:

led.head()

# In[4]:

led.info()

# In[5]:

led.describe()

# In[6]:

led.columns = list(map(str.strip, led.columns.tolist()))

# In[7]:

led.isnull().sum()
```

```
# In[8]:

led.dropna(axis=0, inplace=True)

# In[9]:

led.shape

# In[10]:

led.columns

# # EDA
#
# Let's create some simple plots to check out the data!

# In[11]:

sns.pairplot(led)

# In[12]:

sns.distplot(led['Life expectancy'])

# In[13]:

sns.heatmap(led.corr())

# ## Training a Linear Regression Model
#
# Let's now begin to train out regression model! We will need to first split up our data into an X array that contains the features to train on, and a y array with the target variable, in this case the Price column. We will toss out the Address
```



```

s column because it only has text info that the linear regression model can't use
.
#
# ### X and y arrays

# In[14]:

X = led[['Year', 'Adult Mortality', 'infant deaths', 'Alcohol', 'percentage expenditu
re', 'Hepatitis B', 'Measles', 'BMI', 'under-
five deaths', 'Polio', 'Total expenditure', 'Diphtheria', 'HIV/AIDS', 'GDP', 'Populati
on', 'thinness 1-19 years', 'thinness 5-
9 years', 'Income composition of resources', 'Schooling']]
y = led['Life expectancy']

# ## Train Test Split
#
# Now let's split the data into a training set and a testing set. We will train o
ut model on the training set and then use the test set to evaluate the model.

# In[15]:

from sklearn.model_selection import train_test_split

# In[16]:

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_s
tate=101)

# ## Creating and Training the Model

# In[17]:

from sklearn.linear_model import LinearRegression

# In[18]:

lm = LinearRegression()

```

```
# In[19]:

lm.fit(X_train,y_train)

# ## Model Evaluation
#
# Let's evaluate the model by checking out it's coefficients and how we can interpret them.

# In[20]:

# print the intercept
print(lm.intercept_)

# In[21]:

coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df

# ## Predictions from our Model
#
# Let's grab predictions off our test set and see how well it did!

# In[22]:

predictions = lm.predict(X_test)

# In[23]:

plt.scatter(y_test,predictions)

# **Residual Histogram**

# In[24]:
```

```

sns.distplot((y_test-predictions),bins=50);

# ## Regression Evaluation Metrics
#
#
# Here are three common evaluation metrics for regression problems:
#
# Mean Absolute Error (MAE) is the mean of the absolute value of the errors:
#
# 
$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

#
# Mean Squared Error (MSE) is the mean of the squared errors:
#
# 
$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

#
# Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:
#
# 
$$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

#
# Comparing these metrics:
#
# - MAE is the easiest to understand, because it's the average error.
# - MSE is more popular than MAE, because MSE "punishes" larger errors, which tends to be useful in the real world.
# - RMSE is even more popular than MSE, because RMSE is interpretable in the "y" units.
#
# All of these are loss functions, because we want to minimize them.

# In[25]:

from sklearn import metrics

# In[26]:

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))

#

```

```
# In[27]:
```

```
get_ipython().system('pip install watson-machine-learning-client')
```

```
# In[28]:
```

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

```
# In[29]:
```

```
wml_credentials={  
    "apikey": ,  
    "instance_id": ,  
    "password": ,  
    "url": ,  
    "username":  
}
```

```
# In[ ]:
```

```
# In[30]:
```

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

```
# In[31]:
```

```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "KANAUJ GUHARROY",  
               client.repository.ModelMetaNames.AUTHOR_EMAIL: "gkanauj@gmail.com"  
               ,  
               client.repository.ModelMetaNames.NAME: "Predicting Life Expectancy  
Using Machine Learning"}
```

```
# In[32]:
```

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

```
# In[33]:
```

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

```
# In[34]:
```

```
published_model_uid
```

```
# In[35]:
```

```
deployment = client.deployments.create(published_model_uid, name="Predicting Life  
Expectancy Using Machine Learning")
```

```
# In[36]:
```

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

```
# In[37]:
```

```
scoring_endpoint
```

```
# In[ ]:
```