

Predicting Life Expectancy using Machine Learning

under

Remote Summer Internship Program 2020 by
SmartBridge

11 June 2020 - 9 July 2020

Priya Pitre

B.Tech 2nd Year,

College of Engineering, Pune

Email: priya.pitre@gmail.com

Contact: 7038910181

Index:

1. [Introduction](#)

1.1 Overview

1.2 Purpose

2. [Literature Survey](#)

2.1 Existing Problem

2.2. Proposed Solution

3. [Theoretical Analysis](#)

3.1 Block Diagram

3.2 Hardware Software Requirements

4. [Experimental Investigations](#)

5. [Flowchart](#)

6. [Results](#)

7. [Advantages/ Disadvantages](#)

8. [Applications](#)

9. [Conclusion](#)

10. [Future Scope](#)

11. [Bibliography](#)

Introduction

Problem Statement: Predicting Life Expectancy Using Machine Learning

Overview:

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. Life expectancy varies by geographical area and by era. Since 1900 the global average life expectancy has more than doubled and is now above 70 years. The inequality of life expectancy is still very large across and within countries. In 2019 the country with the lowest life expectancy is the Central African Republic with 53 years, in Japan life expectancy is 30 years longer. Since, life expectancy differs country-wise by such a huge margin, it's necessary to prepare some prediction model which will help the countries with low life expectancy to improve the factors which affect their life expectancy rate.

Purpose:

This project gives countries a way to know which factors have the most effect on life expectancy and in what way. This project provides a crucial step towards changing life for people everywhere. For instance, the Central African Republic has the lowest life expectancy- 53 years, and also a low schooling, GDP, etc. They can now go to this model and tweak certain values (increase schooling by 4, and GDP a little more, for example) to see how life expectancy changes. Now they can set these values as their goal, and work towards it.

In conclusion,

This model helps a particular country know its average life expectancy, and which factors negatively affect life expectancy by a huge margin. Countries can then set goals accordingly and take appropriate steps against those factors to improve life expectancy.

Literature Survey:

The dataset for this project is taken from Kaggle.

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

Existing Problem:

Countries want to increase their life expectancy, but do not have exact data as to what factors affect it and to what degree. We can theoretically say that one factor affects life expectancy more than another, but we do not know by how much we need to change the factor, or by how much life expectancy then increases.

Proposed Solution:

This project takes the following factors as input: 1. Country 2. Year 3. Status 4. Schooling 5. Adult Mortality 6. Alcohol 7. Percentage Expenditure 8. Hepatitis B 9. Measles 10.BMI 11.Under-five deaths 12.Polio 13.Total Expenditure 14.Diphtheria 15.HIV/AIDS 16.GDP 17.Population 18.Thinness 1-19 years 19.Thinness 5-9 years 20.Income composition of resources

And gives life expectancy in years as output. If the life expectancy of a person can be predicted for the coming years then we can be aware of the factors that may affect the life expectancy either in a positive way or in a negative way and accordingly take necessary actions or precautions.

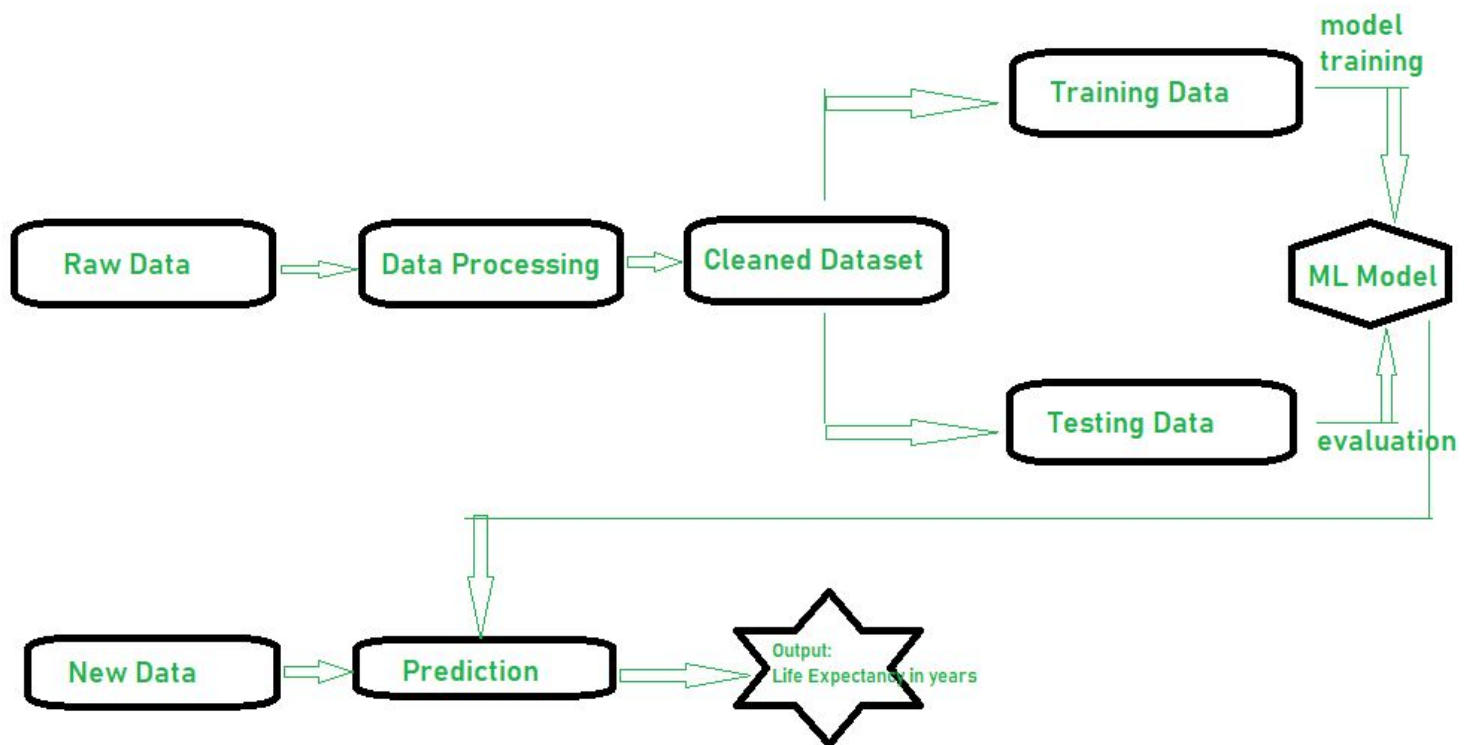
We make certain assumptions in the project, such as:

- The averages apply to the entire country, that is, there is no difference in city/village life expectancies, and even if there is, it's negligible.
- There is no/ negligible difference between male and female life expectancies.

There were a lot of null values in the data, which have been removed using the interpolate function in python. There were also outliers in the data, which have been winsorized. This is to ensure that the dataset we use will model life expectancies as accurately as possible, and there will be no discrepancies from errors.

Theoretical Analysis

Block Diagram:



Hardware and Software Requirements:

Hardware:

- Processor : i3 7th gen or more
- Speed : 2 GHz or more
- Hard Disk Space : 10 GB or more
- Ram Memory : 4 GB or more

Software:

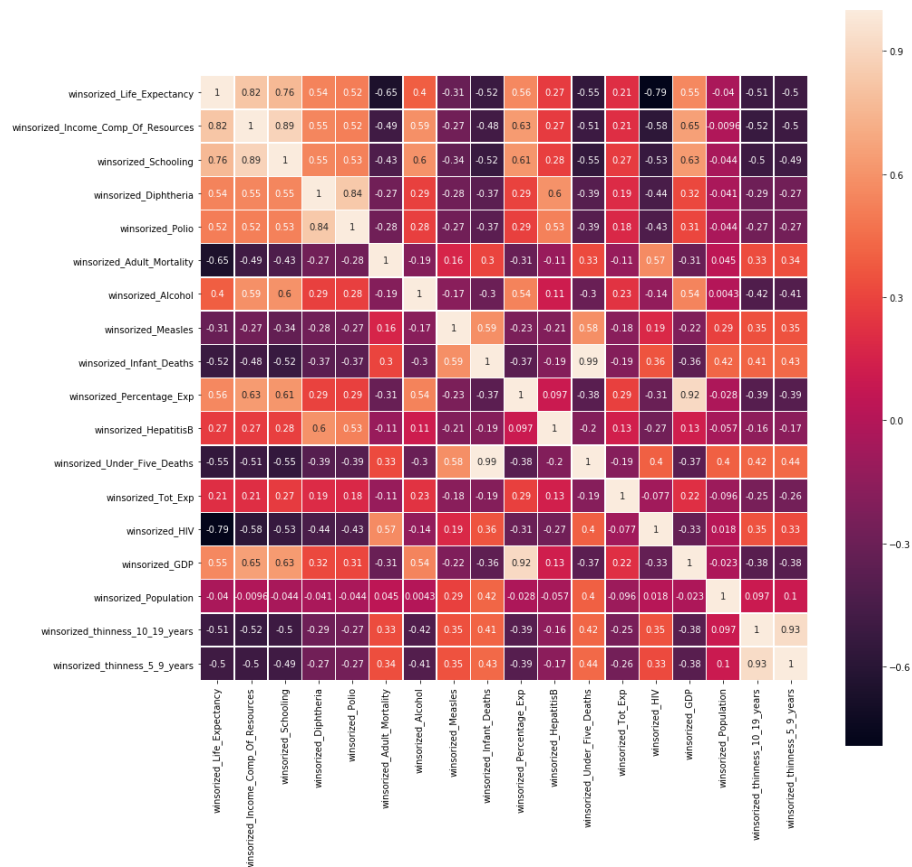
- IBM Cloud Account
 - IBM Watson Studios
 - IBM Machine Learning Service
- Browsers: Chrome, Firefox, etc
- Node Red Application

Experimental Investigations:

Steps to be taken to complete the project:

1. Download the dataset from Kaggle.

<https://www.kaggle.com/kumarajarshi/life-expectancy-who>
2. Predicting Life Expectancy using Python
 - a. Get the IBM Watson Studios service in IBM Cloud.
 - b. Create a new project called life expectancy- in which, get an empty notebook.
 - c. Import the dataset above.
 - d. Clean up the null values in the dataset using the interpolate function.
 - e. Take care of outliers using winsorize function.
 - f. Heat map of the data showing the impact of one feature on another:



- g. Split the data into test set and train set, to reduce overfitting and underfitting.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
```

- h. Importing pipeline and OneHotEncoder:

```
In [48]: from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder

categorical_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(handle_unknown='ignore')),
])
```

- i. Check which Regression achieves the best results.

```
In [51]: #this is done to find which model gives the best results.
models = OrderedDict([
    ("Linear Regression", Pipeline([
        ('preprocessor', preprocessor),
        ('LRegressor', LinearRegression())])),
    ("Decision Tree Regressor", Pipeline([
        ('preprocessor', preprocessor),
        ('DTRegressor', DecisionTreeRegressor())])),
    ("Random Forest Regressor", Pipeline([
        ('preprocessor', preprocessor),
        ('RFRegressor', RandomForestRegressor())])),
])
```

- j. Choose Random Forest Regressor.

```
In [52]: # we see that random forest regressor works the best, as our intuition suggests as well. Random Forest Regressor works
scores = {}
for (name, model) in models.items():
    model.fit(X_train, Y_train)
    scores[name] = r2_score(model.predict(X_test), Y_test)

scores = OrderedDict(sorted(scores.items()))
scores

/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/ensemble/forest.py:246: FutureWarning: The default value
of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
"10 in version 0.20 to 100 in 0.22.", FutureWarning)
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/pipeline.py:267: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
self._final_estimator.fit(Xt, y, **fit_params)

Out[52]: OrderedDict([('Decision Tree Regressor', 0.9249049807289315),
    ('Linear Regression', 0.8188082000368992),
    ('Random Forest Regressor', 0.952241495542773)])
```

- k. Train the model, get the R2 Score, and deploy to get the scoring endpoint to be used later.

```
In [56]: r2_score(predict, Y_test)
```

```
Out[56]: 0.9590849397441741
```

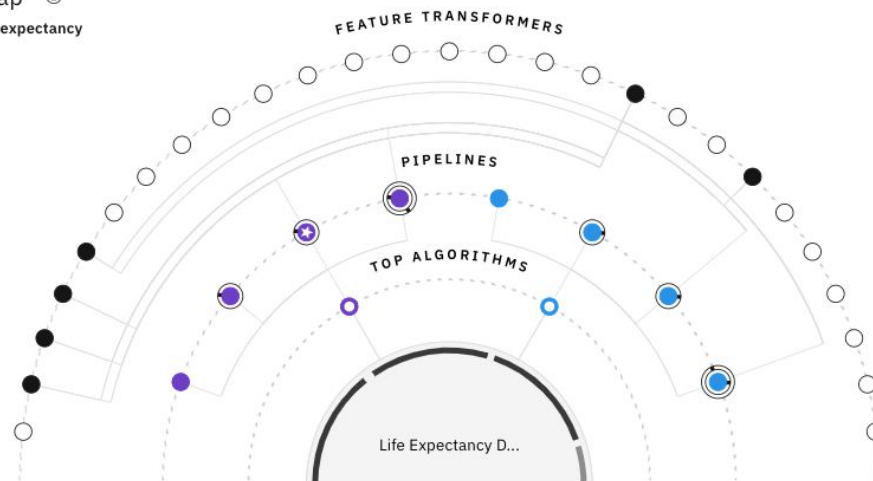
```
In [64]: deployment = client.deployments.create(published_model_uid, name="life_expectancy")
scoring_endpoint = client.deployments.get_scoring_url(deployment)
scoring_endpoint
```

3. Predicting Life Expectancy using AutoAI

- Use Add to Project feature to add an autoAI experiment to the project.
- Upload your dataset.
- Select the parameter to predicted- life expectancy.
- Choose the best pipeline:

Relationship map

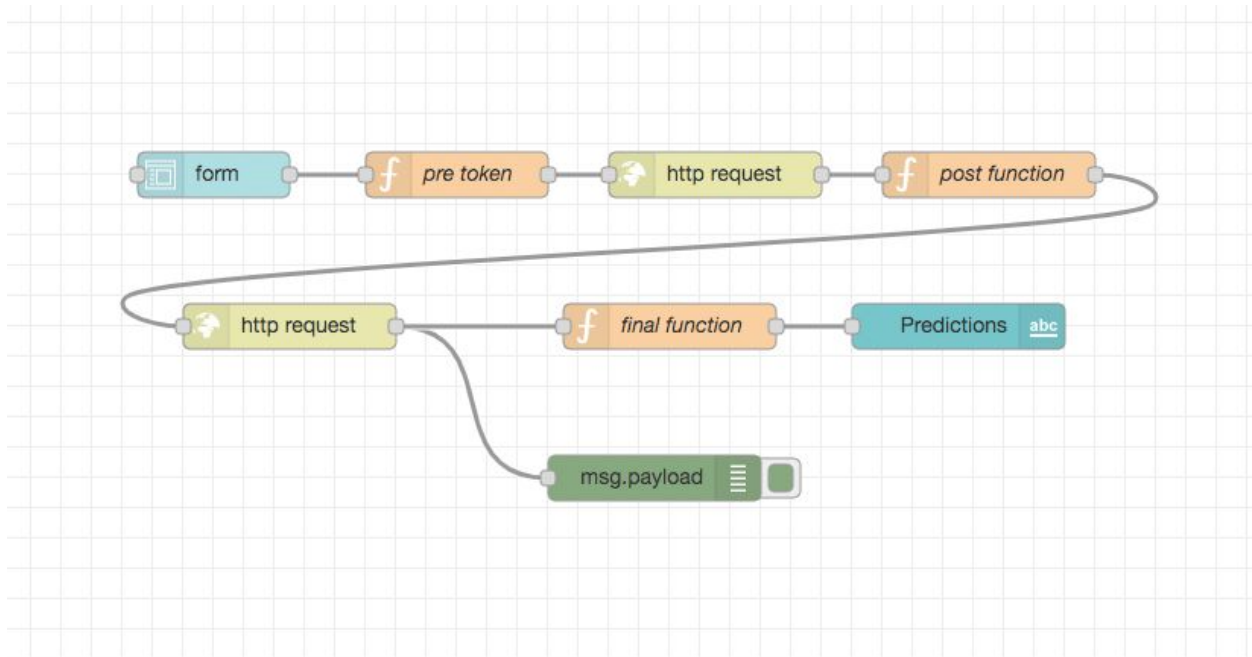
Prediction column: **Life expectancy**



Pipeline leaderboard

	Rank	↑	Name	Algorithm	↑↓	RMSE (Optimized)	Enhancements	Build time
>	★ 1		Pipeline 3	Extra Trees Regressor		2.016	HPO-1 FE	00:00:46
>	2		Pipeline 4	Extra Trees Regressor		2.016	HPO-1 FE HPO-2	00:00:32
>	3		Pipeline 1	Extra Trees Regressor		2.070	None	00:00:01
>	4		Pipeline 2	Extra Trees Regressor		2.070	HPO-1	00:00:10
>	5		Pipeline 7	Decision Tree Regressor		2.732	HPO-1 FE	00:00:37

- e. Download it as a notebook and model.
 - f. Go to the model and deploy the project.
 - g. Find the scoring endpoint.
4. Node Red UI for both:
- a. Flow for both projects:



- b. Form Node: enter labels from dataset and their type.

Group
Size
Label
Form elements

[Life Expectancy With Python] Default

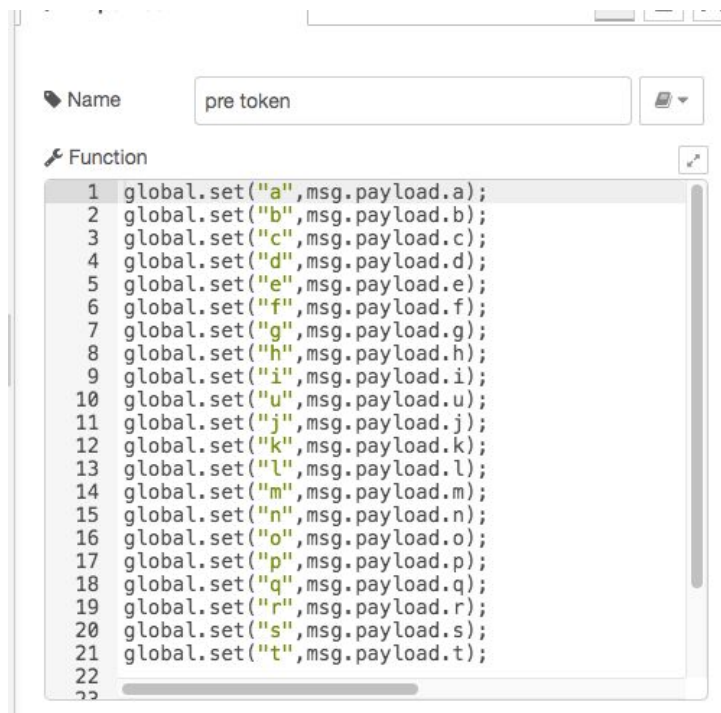
auto

optional label

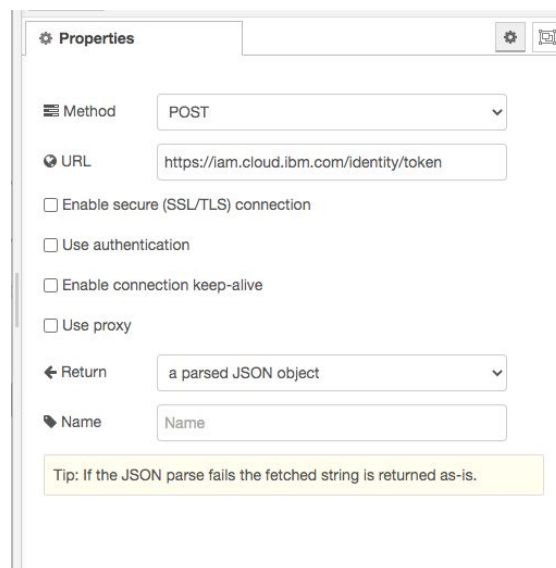
Label	Name	Type	Required	Rows	Remove
Country	a	Text	<input checked="" type="checkbox"/>		
Year	b	Number	<input checked="" type="checkbox"/>		
Status	c	Text	<input checked="" type="checkbox"/>		
BMI	d	Number	<input checked="" type="checkbox"/>		
Adult Mortality	e	Number	<input checked="" type="checkbox"/>		

+ element

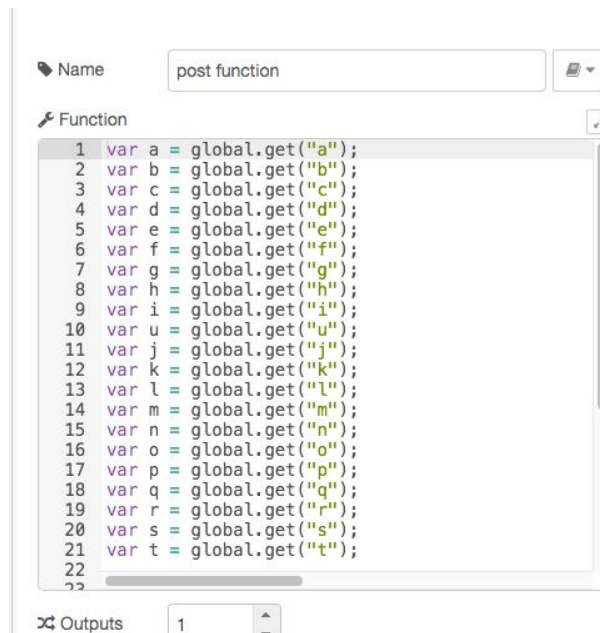
c. Pre token: set everything to global.



d. Http request: URL = where the nodered will be visible.



- e. Post function: gets all the variables.



```

1 var a = global.get("a");
2 var b = global.get("b");
3 var c = global.get("c");
4 var d = global.get("d");
5 var e = global.get("e");
6 var f = global.get("f");
7 var g = global.get("g");
8 var h = global.get("h");
9 var i = global.get("i");
10 var u = global.get("u");
11 var j = global.get("j");
12 var k = global.get("k");
13 var l = global.get("l");
14 var m = global.get("m");
15 var n = global.get("n");
16 var o = global.get("o");
17 var p = global.get("p");
18 var q = global.get("q");
19 var r = global.get("r");
20 var s = global.get("s");
21 var t = global.get("t");
22
23
  
```

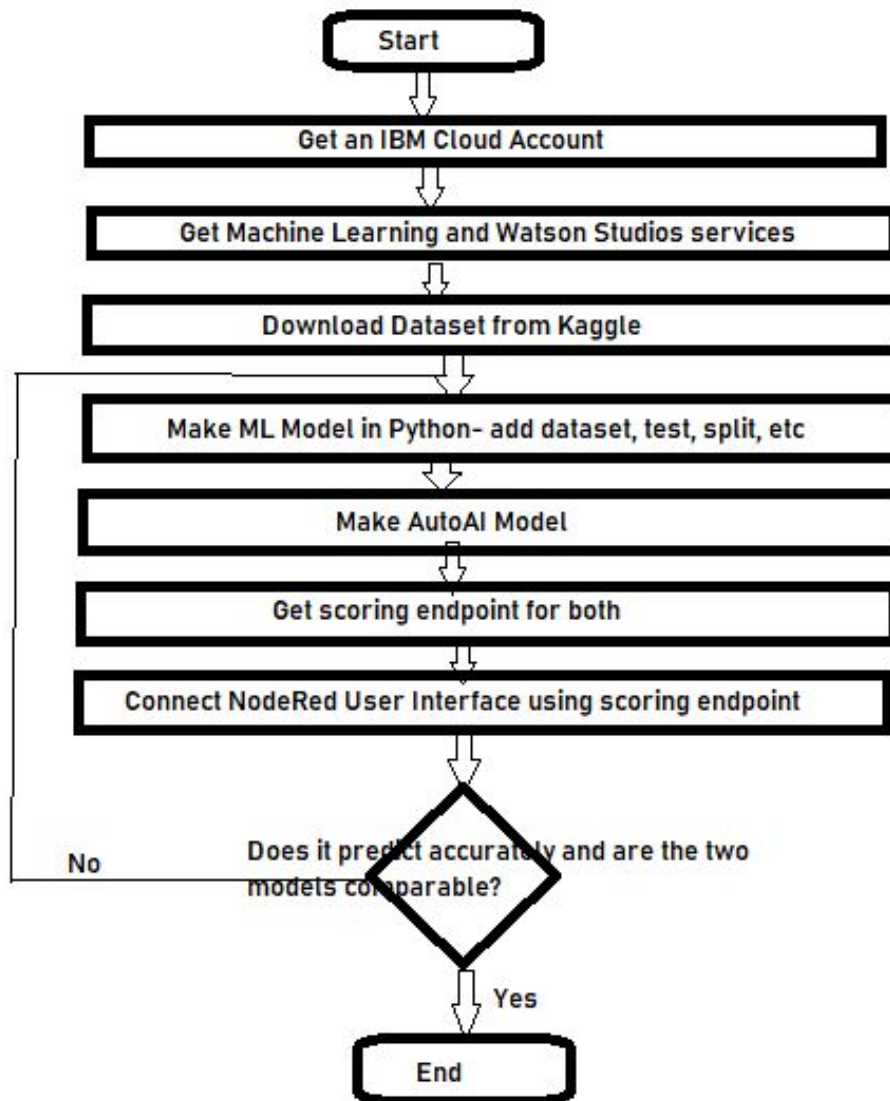
- f. Https: URL = scoring endpoint from both the models.
- g. Some other nodes that help with the prediction, make sure to set payload.values to [0][0] as that's where the output is visible. This is how to set up the experiments and UI.

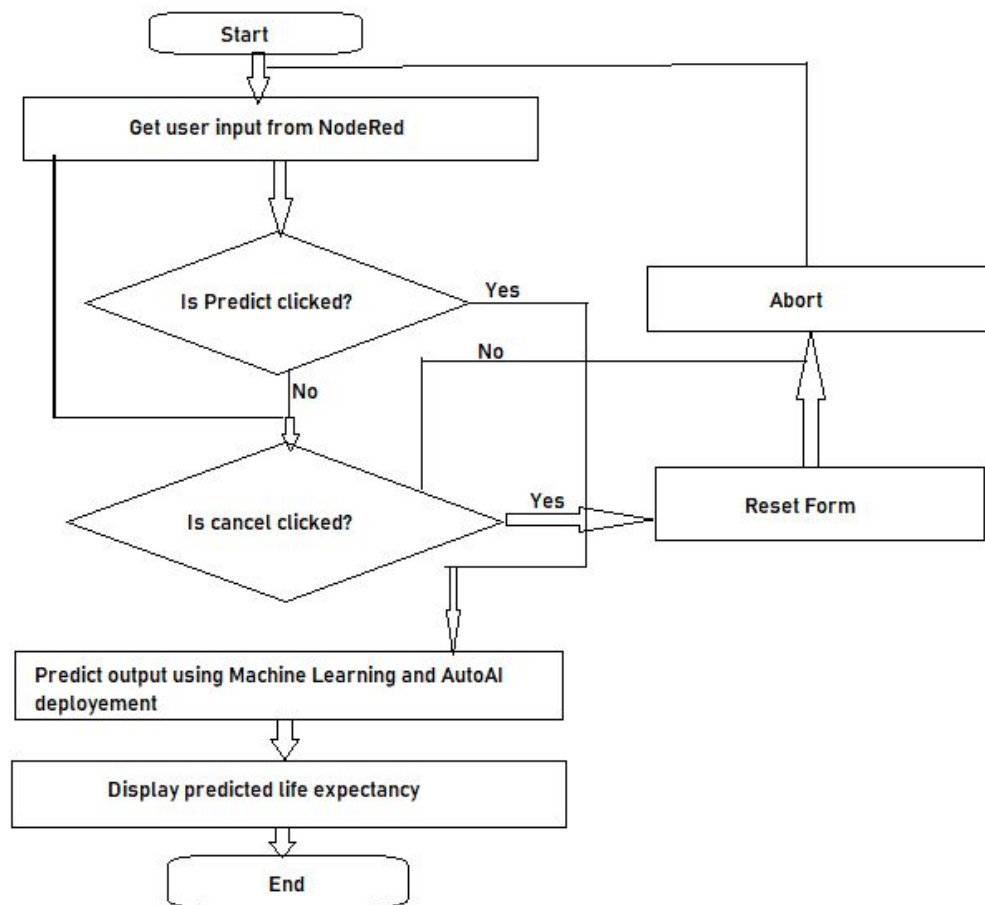
Flowcharts:

Flowchart 1: the flowchart of the whole process, described above.

Flowchart 2: a flowchart of how Node Red functions.

(Please Keep Scrolling for the flowcharts)





Results:

Python:

R2 Score for Random Forest Regressor: 95.90%

Random Forest Train Score: 96.2%

Random Forest Test Score: 92%

AutoAI: Pipeline 3 measured the best, which was an Extra Trees Regressor. This is the data for it:

Model evaluation measures

	Cross validation score	Holdout score
Explained variance	0.956	0.961
MAE	1.282	1.182
MSE	4.057	3.347
MSLE	0.001	0.001
MedAE	0.747	0.740
RMSE	2.010	1.830
RMSLE	0.031	0.028
R ²	0.956	0.961

NodeRed: Final Product with Results:

Python:

Life Expectancy With Python

Default

Predictions76.51

Country *
Albania

Year *
2011

Status *
Developing

BMI *
55.1

Adult Mortality *
88

Infant Deaths *
0

Alcohol *
5.37

Percentage Expenditure *
437

HepatitisB *
89

Under five deaths *
1

Polio *
99

Total Expenditure *
5.71

Diphtheria *
99

HIV *
0.1

GDP *
4437

Population *
295195

Thinness 10 to 19 years *
1.4

Thinness 5 to 9 years *
1.5

Income composition of resources *
0.738

Schooling *
13.3

Measles *
28

PREDICTCANCEL

AutoAI:

Life Expectancy Prediction without Python using AutoAI

Default

Country *

Albania

Year *

2011

Status *

Developing

Adult Mortality *

88

Infant Deaths *

0

Alcohol *

5.37

Percentage Expenditure *

437

Hepatitis B *

89

Measles *

28

BMI *

55.1

Under five deaths *

1

Polio *

99

Total Expenditure *

5.71

Diphtheria *

99

HIV/AIDS *

0.1

GDP *

4437

Population *

295195

Thinness 1-19 years *

1.4

Thinness 5-9 years *

1.5

Income composition of resources *

0.738

Schooling *

13.3

PREDICT

CANCEL

Life Expectancy
Prediction: **76.56999893188477**

As we can see after entering the data, both AutoAI and Python give similar values for life expectancy predictions. Hence, our models are reliable in predicting average life expectancy.

Advantages and Disadvantages:

Adv:

- The model gives a fairly accurate representation of the factors that affect the life expectancy in a specific country.
- These factors can now be modified in real life- using accurate and reliable data from our model to save lives and increase life expectancy.
- Random Forest Regressor and our AutoAI model both predict a similar value- and the value is comparable to the actual value in the dataset.

Disadv:

- Lots of assumptions: The dataset is only based on country- it doesn't take anything into account on an individual level- such as genetics, predisposition to diseases, etc. So this information isn't really useful to a lot of people on a personal level. It also doesn't take into account gender and race differences, which we know exist in real life.
- Factors hard to change in real life: Even though our model can tell us exactly what factors should be changed, it is very difficult to change these factors in real life. Poverty, corruption make it really hard to change these factors.

Applications:

- As stated before, countries can reliably look at what factors affect life expectancy and how. Here are some factors that have a strong correlation with life expectancy:
 - Schooling and life expectancy is strongly correlated. This is because good schooling is mostly only available in wealthier nations, who already have a better life expectancy as a result of better healthcare, etc. So as we can see, it is not a causation but a correlation.
 - GDP is also strongly correlated, but also not a causation, and likely the same reason.
 - Alcohol has a causation with life expectancy- so countries can start an anti alcohol, anti drugs campaign to reduce drugs and alcohol in their countries- which can help with life expectancy.

- When editing the nodes, we can see that changing the adult mortality has a huge effect on life expectancy. When increased, it significantly reduces life expectancy. Countries need to focus on building a good healthcare system and foster a healthy lifestyle in people.

Conclusion:

Life expectancy predictions can save countries a lot of resources on data analysis- and tell them exactly where they should allocate their resources to foster a better population. Creating models like this can save lives.

Future Scope:

- Better dataset, with more factors considered
- Something similar done on an individual level- where we consider genetic predisposition, etc, and consider country only part of a person's identity- not everything. (We need a reliable dataset for this first)
- Using Machine Learning in the medical field to improve data reading and predicting capabilities and saving lives.

Bibliography:

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

IBM Cloud Account and Info:

<https://content-eu-7.content-cms.com/b73a5759-c6a6-4033-ab6b-d9d4f9a6d65b/dxsites/151914d1-%2003d2-48fe-97d9-d21166848e65/academic/home>

Node Red Info:

<https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>

Links for this specific project:

My Github Source Code:

<https://github.com/SmartPracticeschool/IISPS-INT-2927-Predicting-Life-Expectancy-using-Machine-Learning>

My Youtube Video demonstrating the project and explaining every process in detail:

https://www.youtube.com/watch?v=4tiK_0z7EDQ&t=580s



Final Node Red Flows:

https://node-red-jufjq.eu-gb.mybluemix.net/ui/#!/1?socketid=Mvmhv0d8CgmR_dOIAAAJ