PROJECTREPORT

PREDICTING LIFE EXPECTANCY USING MACHINELEARNING

Submitted by-
AMEY SAWANT

EMAIL ID:
Ameys2907@gmail.com

GITHUB: **:**
https://github.com/SmartPracticeschool/llSPS-INT-2822-Predicting-Life-Expectancy-using-Machine-Learning

PROJECT LINK:
https://node-red-myqqu.eu-gb.mybluemix.net/ui/#!/0?socketid=ad5Yek0O22CixOV7AAAA

<u>Contents</u>

1.INTRODUCTION:

1.1 Overview

The fundamental goal of the undertaking is to foresee the future of an individual relying upon a few components dependent on an individual or the living nation. Elements like the GDP of the nation, social insurance office framework, personal satisfaction, mental and physical ailment, age, sexual orientation, training and other territorial, segment and financial variables are considered to anticipate the life expectancy of the individual utilizing machine learning calculations. Machine learning algorithms that can be used in this case are: Regression, Decision Tree, Random Forest so that we can achieve high accuracy for our model.

1.2 Purpose

Predicting the life expectancy will give the nation a thought of the components which can be improved to expand the life expectancy of the individuals living, as by improving the human services offices or vaccination antibodies for babies. By making changes in way of life, an individual can live a long, sound and great quality life. This will likewise profit the nation by expanding labor that will add to the financial development. We should exploit this new period cutting edge innovation to improve the future by foreseeing it in the present.

2. LITERATURE SURVEY:

2.1 Existing Problem

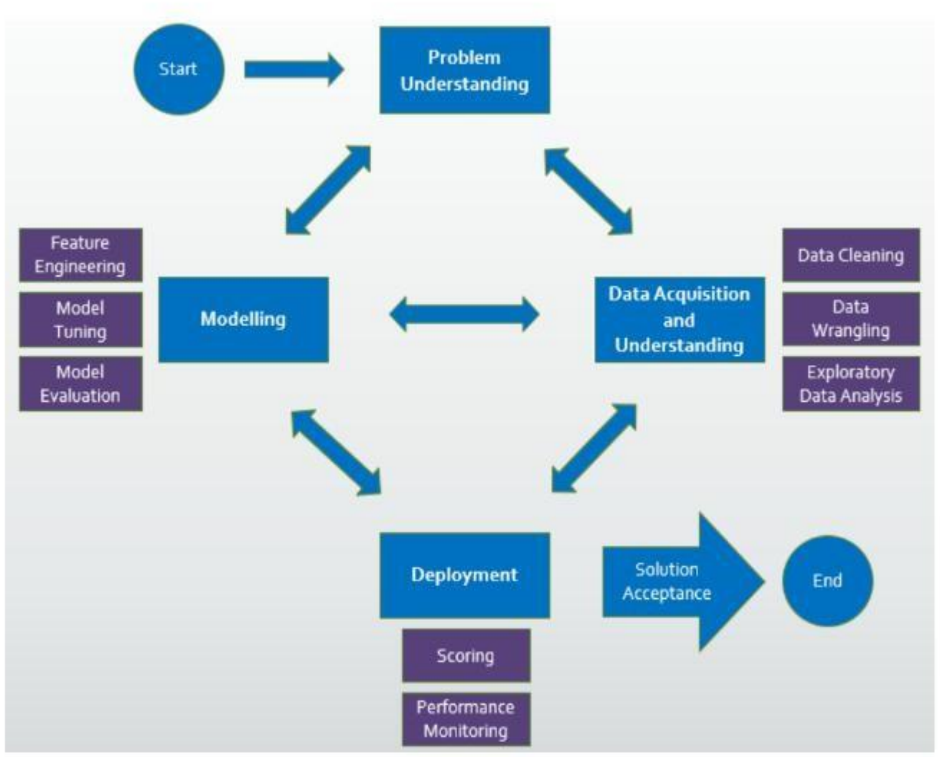
We have assessed existing works and methods utilized in the forecast of the human future, and arrived at a resolution that it is doable for people utilizing developing advancements and gadgets wearables and portable wellbeing checking gadgets. We have likewise distinguished that the elements utilized for anticipating were simply close to home causes and not identified with the encompassing, medicinal services offices, segment, social, provincial and monetary components of the nation he lives. These nation subordinate components can likewise be a significant element to anticipate the future of a person. In this way, we need more information to foresee all the more precisely.

2.2 Proposed Solution

To improve bits of knowledge and anticipate the future all the more precisely, we have to consider some extra highlights, for example, nation or encompassing ward highlights. The past elements were increasingly human based yet it is critical to know the prudent, territorial, social and segment factors like the GDP, populace, instruction, vaccinations, history of ailment, medicinal services office, reserves assigned by the administration, plans, clinical uses like on the off chance that it is exceptionally high, at that point individuals will stay away to get standard clinical tests, and some more. Because of the enormous information, we will utilize IBM Cloud to construct our model which will expand the task productivity.

3.THEORITICAL ANALYSIS:

3.1 Block Diagram



3.2 Hardware / Software designing:

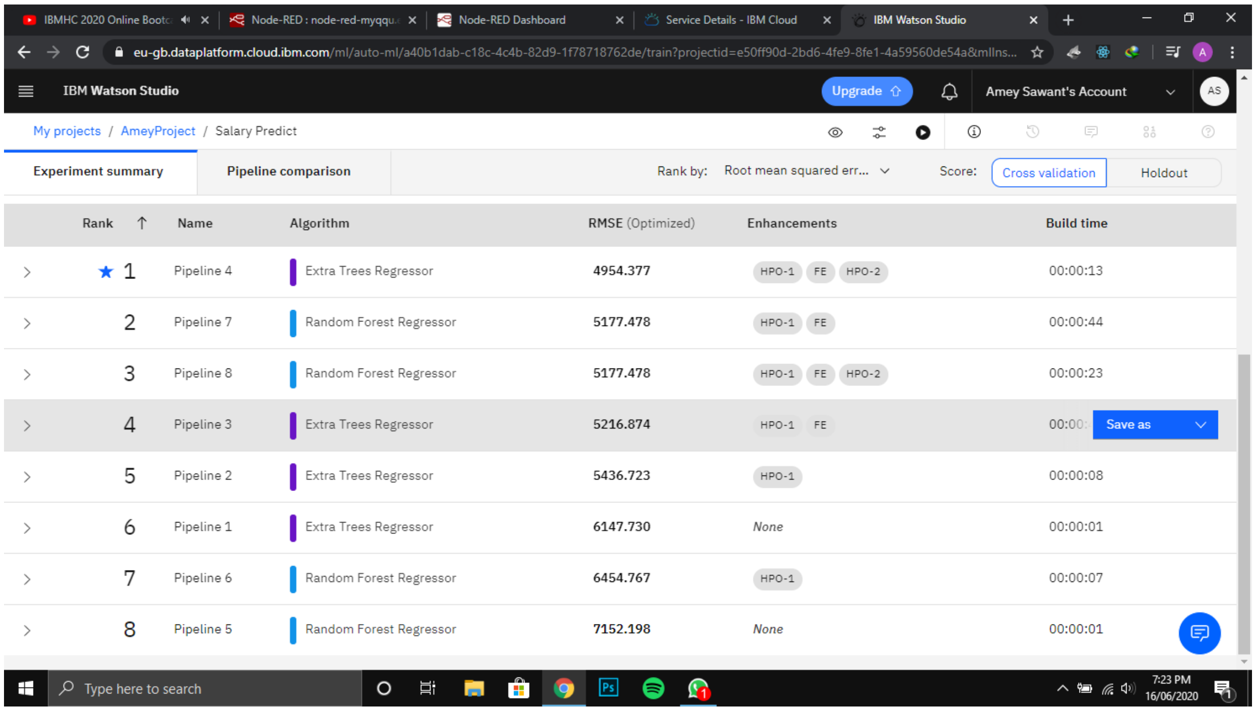
Functional Requirements:

1. The dataset should be preprocessed before modelling.
2. Feature Engineering should be performed to select co-related and integer values.
3. Efficient machine learning algorithm should be used.
4. The project should be implemented using IBM Watson Studio which should then connected to Node-Red App for UI and deployed.

Software Requirements:

Python IDE, IBM Watson Studio, IBM Machine Learning Services, IBM Cloud, Node-Red App, Excel.

4. EXPERIMENTAL INVESTIGATION:



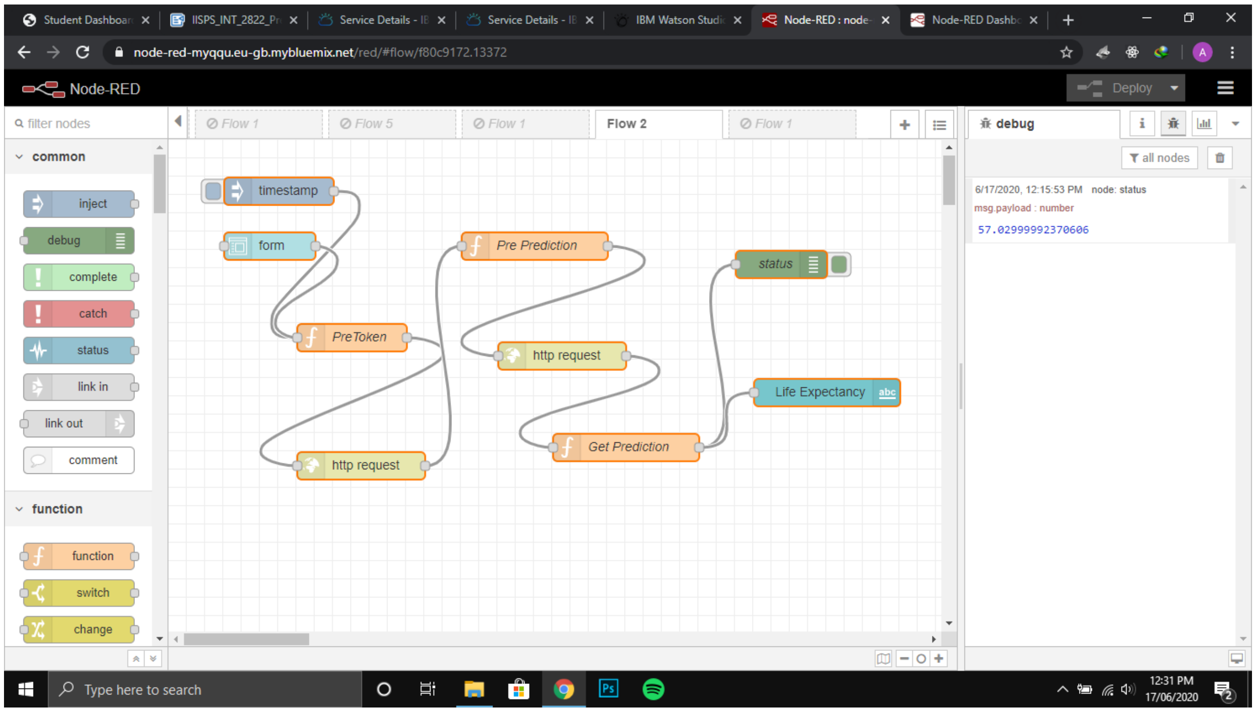
Rank by: Root mean squared err...

Score: Cross validation | Holdout

Rank	Name	Algorithm	RMSE (Optimized)	Enhancements	Build time
★ 1	Pipeline 4	Extra Trees Regressor	4954.377	HPO-1 FE HPO-2	00:00:13
2	Pipeline 7	Random Forest Regressor	5177.478	HPO-1 FE	00:00:44
3	Pipeline 8	Random Forest Regressor	5177.478	HPO-1 FE HPO-2	00:00:23
4	Pipeline 3	Extra Trees Regressor	5216.874	HPO-1 FE	00:00:
5	Pipeline 2	Extra Trees Regressor	5436.723	HPO-1	00:00:08
6	Pipeline 1	Extra Trees Regressor	6147.730	None	00:00:01
7	Pipeline 6	Random Forest Regressor	6454.767	HPO-1	00:00:07
8	Pipeline 5	Random Forest Regressor	7152.198	None	00:00:01

The model is built using linear regression that had the RMSE score of 5344.72 and then, AutoAI Extra Tree Regressor was implemented, which gave an RMSE score of 4954.377. So, this gives us a better accuracy of the model.

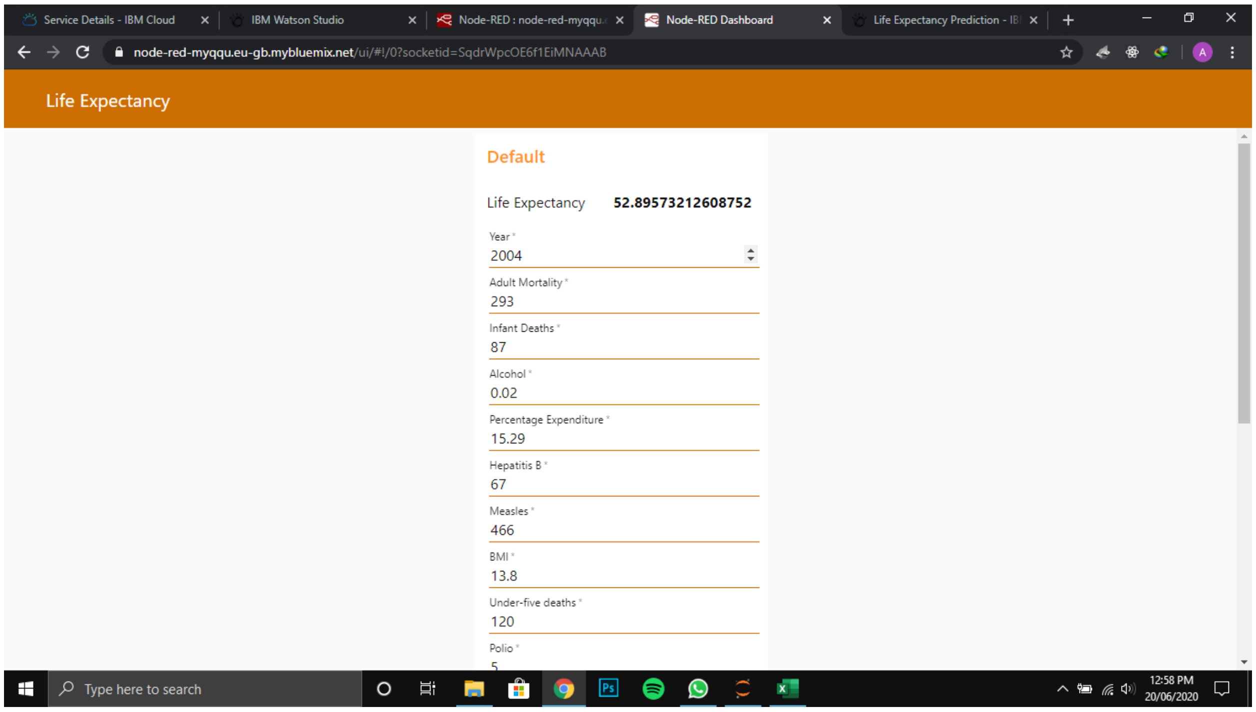
5. FLOWCHART:



```
6/17/2020, 12:15:53 PM   node: status
msg.payload : number
57.02999992370606
```

6. RESULT:

The model predicts life expectancy accurately based on user inputs on the node RED user interface. The application is cloud based and works efficiently.


Life Expectancy

Default

Life Expectancy 52.89573212608752

Year *	2004
Adult Mortality *	293
Infant Deaths *	87
Alcohol *	0.02
Percentage Expenditure *	15.29
Hepatitis B *	67
Measles *	466
BMI *	13.8
Under-five deaths *	120
Polio *	5

7. ADVANTAGES & DISADVANTAGES:

Advantages:

- The life expectancy predictor will give important insights and help people achieve good quality of life in future. The country can plan and improve various healthcare facilities.
- Benefit the country's growth.
- Advantages of using IBM Cloud: Easy to use and deploy, easy to connect with UI, takes care of large storage space.

Disadvantages:

- Requires internet connection.
- User input is not saved in any database.
- Input should be in range only to predict accurate values.

8. APPLICATIONS:

- To analyze country's growth statistics in future years.
- To help government prepare life insurance policies for people. This will benefit the people.
- To analyze all the factors and plan out measures to increase the life expectancy of the country.

9. CONCLUSION:

In this project, we have made a cloud-based ML application which will foresee the future of an individual living in a particular area. Different elements like Adult Mortality, Population, under 5 Deaths, Thinness 1-5 Years, Alcohol, HIV, Hepatitis B, GDP, Percentage Expenditure and a lot more assume a significant job in the expectation. Client can associate with the framework by means of a straightforward UI which is in the structure.

10. FUTURE SCOPE:

Prognostication of life expectancy is difficult for humans. Our research shows that machine learning techniques offer a feasible and promising approach to predicting life expectancy. The research has potential for real-life applications, such as supporting timely recognition of the right moment to start Advance Care Planning.

11. BIBLIOGRAPHY:

https://bookdown.org/caoying4work/watsonstudio-workshop/jn. html
https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

12. APPENDIX:

A. Source Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats.mstats import winsorize
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_er
ror
```

```
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_eab848c2c8384710bfd4d52a3a8c56ce = ibm_boto3.client(service_name='s
3',
    ibm_api_key_id='5QJ7L6IcHehgbMzb1UN7p0RhO6L2SgjWneecP-Zwlqho',
    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_eab848c2c8384710bfd4d52a3a8c56ce.get_object(Bucket='ameyproj
ect-donotdelete-pr-iyhtsqozclqsce',Key='Life_Expectancy.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()
df.isnull().sum()
```

Out[4]:

```
df.dropna(inplace=True)
```

```
df.rename(columns={" BMI ":"BMI","Life expectancy ":"Life_Expectancy","Adult
Mortality":"Adult_Mortality",
                    "infant deaths":"Infant_Deaths","percentage expenditure":"
Percentage_Exp","Hepatitis B":"HepatitisB",
                   "Measles ":"Measles"," BMI ":"BMI","under-five deaths ":"Un
der_Five_Deaths","Diphtheria ":"Diphtheria",
                   " HIV/AIDS":"HIV/AIDS"," thinness  1-19 years":"thinness_1t
o19_years"," thinness 5-9 years":"thinness_5to9_years","Income composition of
resources":"Income_Comp_Of_Resources",
                    "Total expenditure":"Tot_Exp"},inplace=True)
```

In [12]:

```
col_dict = {'Life_Expectancy':1 , 'Adult_Mortality':2 ,
        'Alcohol':3 , 'Percentage_Exp': 4, 'HepatitisB': 5,
       'Measles' : 6, 'BMI': 7, 'Under_Five_Deaths' : 8, 'Polio' : 9, 'Tot
_Exp' :10,
       'Diphtheria':11, 'HIV/AIDS':12,
       'thinness_1to19_years' :15, 'thinness_5to9_years' :16,
       'Income_Comp_Of_Resources' : 17, 'Schooling' :18, 'Infant_Deaths':1
9}
```

```
for variable in col_dict.keys():
    q75, q25 = np.percentile(df[variable], [75 ,25])
    iqr = q75 - q25
    min_val = q25 - (iqr*1.5)
    max_val = q75 + (iqr*1.5)
    print("Number of outliers in {} : {} ".format(variable,len((np.where((
df[variable] > max_val) | (df[variable] < min_val))[0]))))

winsorize(df["Life_Expectancy"],(0.01,0), inplace=True)
winsorize(df["Adult_Mortality"],(0,0.03), inplace=True)
winsorize(df["Infant_Deaths"],(0,0.10), inplace=True)
winsorize(df["Alcohol"],(0,0.01), inplace=True)
winsorize(df["Percentage_Exp"],(0,0.12), inplace=True)
winsorize(df["HepatitisB"],(0.11,0), inplace=True)
winsorize(df["Measles"],(0,0.19), inplace=True)
winsorize(df["Under_Five_Deaths"],(0,0.12), inplace=True)
winsorize(df["Polio"],(0.09,0), inplace=True)
winsorize(df["Tot_Exp"],(0,0.01), inplace=True)
winsorize(df["Diphtheria"],(0.10,0), inplace=True)
winsorize(df["HIV/AIDS"],(0,0.16), inplace=True)
winsorize(df["thinness_1to19_years"],(0,0.04), inplace=True)
winsorize(df["thinness_5to9_years"],(0,0.04), inplace=True)
winsorize(df["Income_Comp_Of_Resources"],(0.05,0), inplace=True)
winsorize(df["Schooling"],(0.02,0.01), inplace=True)
```

```
X = df.drop(['Life_Expectancy'], axis=1).values
Y = df.iloc[:, 1].values.reshape(-1,1)
```

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

In [20]:

```
model = LinearRegression(fit_intercept=True, normalize=True).fit(X_train, Y_t
rain)
```

In [21]:

```
predictions= model.predict(X_test)
```

In [22]:

```
mean_squared_error(predictions, Y_test)
```

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient

wml_credentials = {
  "apikey": "Qwg7NXlnJ6B2n-DEu4HQcSdiQr14THwEI0KjbJRKhJDN",
  "iam_apikey_description": "Auto-generated for key 45b803bb-9e5a-4c34-ba8
5-b8bc4b84af4d",
  "iam_apikey_name": "Service credentials-3",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/047d7e5db328
4f559d41b37e6d5ea6ae::serviceid:ServiceId-bd772675-728c-4069-a2e2-98625bc7
1c9a",
  "instance_id": "516ce7de-7708-43d2-8e8a-c088cdf276e0",
  "url": "https://eu-gb.ml.cloud.ibm.com"
}
```

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

In [27]:

```
model_props = {
    client.repository.ModelMetaNames.AUTHOR_NAME : "Amey",
    client.repository.ModelMetaNames.AUTHOR_EMAIL : 'ameys2907@gmail.com',
    client.repository.ModelMetaNames.NAME : "Life_Expectancy"
}
```

In [28]:

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

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

deploy = client.deployments.create(guid, name="Life_Exp")
scoring_url = client.deployments.get_scoring_url(deploy)
```

```
scoring_url
```

Out[36]:

```
'https://eu-gb.ml.cloud.ibm.com/v3/wml_instances/516ce7de-7708-43d2-8e8a-c088
cdf276e0/deployments/2fa3b583-22fe-4f47-bb7f-f827e6cd3afe/online'
```