

A PROJECT REPORT ON

PREDICTING LIFE EXPECTANCY USING MACHINE LEARNING

DONE BY : SRICHARAN L N NIBHANUPUDI

CONTENTS

- 1.) INTRODUCTION
 - 1.1) Overview
 - 1.2) Purpose
 - 2.) LITERATURE SURVEY
 - 2.1.) Existing problem
 - 2.2) Proposed solution
 - 3.) THEORITICAL ANALYSIS
 - 3.1.) Block diagram
 - 3.2.) Hardware / Software designing
 - 4.) EXPERIMENTAL INVESTIGATIONS
 - 5.) FLOWCHART
 - 6.) RESULT
 - 7.) ADVANTAGES & DISADVANTAGES
 - 8.) APPLICATIONS
 - 9.) CONCLUSION
 - 10.) FUTURE SCOPE
 - 11.) BIBILOGRAPHY
- APPENDIX
- A. Source code

INTRODUCTION

OVERVIEW :

The project Predicting the Life Expectancy of a Country using Machine Learning. Here, We have to predict the Life Expectancy of a country. The most common measure of life expectancy is life expectancy at birth. Life expectancy is a hypothetical measure. It assumes that the age-specific death rates for the year in question will apply throughout the lifetime of individuals born in that year. In this project tries to create a new model based on the data provided is to evaluate the life expectancy in India.

PURPOSE :

The purpose of the project is to predict the life expectancy of a given country. Life expectancy is the average people live in a country. This depends on various factors such as GDP, population, diseases, expenditure etc.

With this we can say that how safe is that particular country for people and we can also conclude how developed is the country.

LITERATURE SURVEY :

Literature survey is the important in project development process. Before developing the project, it is necessary to determine time factor, economy, scope of the project etc.

EXISTING PROBLEM :

The problem in most of the countries is they do not have required resources to tackle problems which the people of that country. Few under developed countries do not have a good GDP rate, high expenditure and low income, also

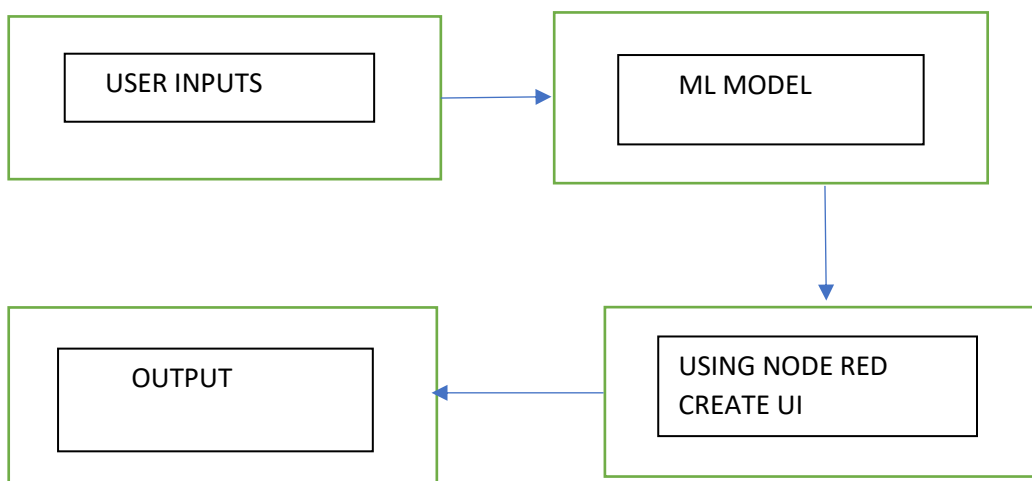
they have poor health infrastructure with which they cannot control the outbreak of a disease.

PROPOSED SOLUTION :

The proposed solution is when the countries get to know about the life expectancy of their country, they'll get to know what are the problems they are facing clearly and they can get sorted one by one.

THEORITICAL ANALYSIS :

BLOCK DIAGRAM :



HARDWARE/SOFTWARE DESIGNING:

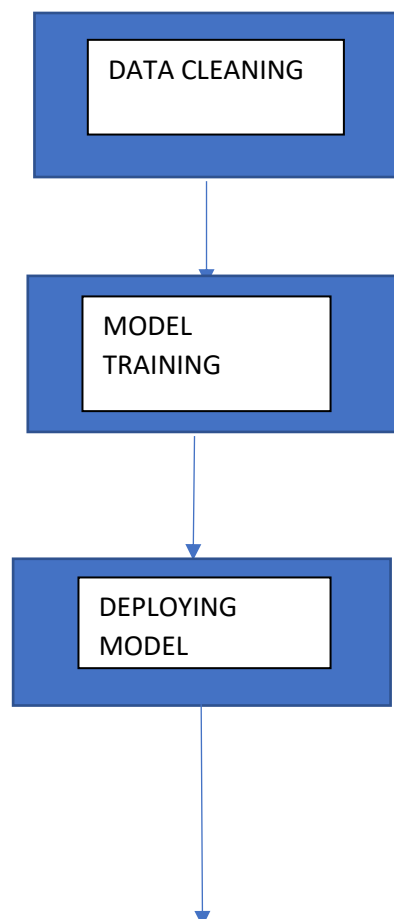
The requirements in designing this model are :

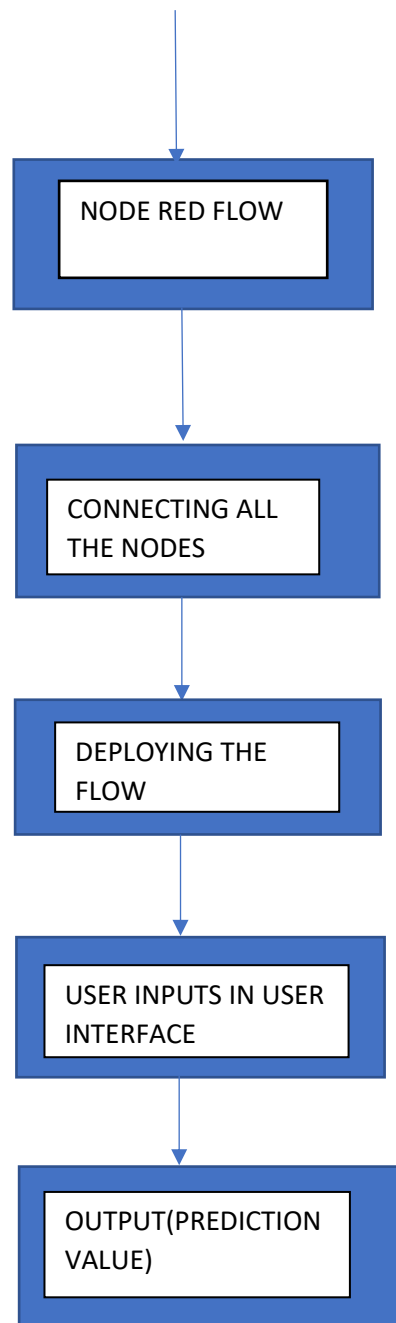
- 1.) Windows 7 or more
- 2.) IBM Watson Studio
- 3.) Python 3.0 or more
- 4.) Node Red
- 5.) i3 or higher processor for better speed.

EXPERIMENTAL INVESTIGATIONS :

The experimental investigation is a fair test done on our project to check whether our model is giving the user a fair value. In our project, the model is giving an accuracy 95.6% which is a good accuracy. All the values which are given by the user to the fields are producing accurate results.

FLOW CHART :





RESULT :

The result of the project is the user gets to know the average lifespan or life expectancy of the country after giving all the required inputs such as Year, Adult Mortality, GDP, Population, Hepatitis etc.

ADVANTAGES AND DISADVANTAGES :

Advantages of Life Expectancy :

- 1.) Can predict life expectancy at any age.
- 2.) Life expectancy is generally used to monitor health inequalities in a country.

Disadvantages of Life Expectancy :

- 1.) At smaller geographies may be influenced by nursing homes in the area.
- 2.) This can instill anxiety and fear in ones mind after knowing about their lifespan.

APPLICATIONS :

The applications of Life Expectancy are :

- 1.) Construction of Life Table , A life Table is a table which shows , for a person age, what the probability is that they die before their birthday.
- 2.) Summary measures of mortality and overall mortality rate.
- 3.) In public sectors, they are used by insurance companies and actuary departments.

CONCLUSION :

The main objective of the project is to predict the average life span of a person taking different factors such as GDP, Population, No. of people affected with disease, Resources, expenditure of the country into account and determine the life span of the people in the country. The results which we get is accurate upto 95 percent.

FUTURE SCOPE :

As future work, I am going to replace files with database. As the data which contains different factors gets updated if there is a change in the number.

And the prediction value gets automated each time if there is a change in the factors.

BIBLIOGRAPHY :

The list of all the resources used in the project work are as follows:

- 1.) <https://www.kaggle.com/kumarajarshi/life-expectancy-who>
- 2.) <https://www.geeksforgeeks.org/pandas-tutorial/>
- 3.) <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

APPENDIX :

SOURCE CODE :

```
life = pd.read_csv(body)
life.head()
life.head()
life.isnull().sum()
life['Alcohol'].fillna(life['Alcohol'].mean(), inplace = True)
life['Hepatitis B'].fillna(life['Hepatitis B'].mean(), inplace = True)
life['Polio'].fillna(life['Polio'].mean(), inplace = True)
life['Income composition of resources'].fillna(life['Income composition of resources'].mean(), inplace = True)
life['Schooling'].fillna(life['Schooling'].mean(), inplace = True)
life.dropna(inplace = True)
life.isnull().sum()

life.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 ":"Under_Five_Deaths", "Diphtheria ":"Diphtheria",
```

```

        " HIV/AIDS":"HIVAIDS", " thinness 1-19 years":"thinness_1
to19_years", " thinness 5-9 years":"thinness_5to9_years", "Income composition
of resources":"Income_Comp_Of_Resources",
        "Total expenditure":"Tot_Exp"}, inplace=True)

```

In [8]:

```

col_dict = {'Life_Expectancy':1, 'Adult_Mortality':2, 'Infant_Deaths':3, 'Alco
hol':4, 'Percentage_Exp':5, 'HepatitisB':6, 'Measles':7, 'BMI':8, 'Under_Five_De
aths':9, 'Polio':10, 'Tot_Exp':11, 'Diphtheria':12, 'HIVAIDS':13, 'thinness_1to1
9_years':14, 'thinness_5to9_years':15, 'Income_Comp_Of_Resources':16, 'Schooli
ng':17}

```

```

plt.figure(figsize=(20,30))

```

```

for variable,i in col_dict.items():
    plt.subplot(5,4,i)
    plt.boxplot(life[variable],whis=1.5)
    plt.title(variable)

```

```

plt.show()

```

```

lower_bound = 0.1

```

```

higher_bound = 0.95

```

```

result = life.quantile([lower_bound, higher_bound])

```

```

result

```

```

from sklearn.preprocessing import LabelEncoder, StandardScaler
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_err
or
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor

```

In [10]:

```

X = life.drop('Life_Expectancy',axis=1)

```

```

Y = life['Life_Expectancy']

```

```

X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size = 0.2,ran
dom_state=42)

```

In [11]:

```

l = life.columns

```

```

lb = LabelEncoder()

```

```

for i in l:

```

```

    life[i]= lb.fit_transform(life[i])

```

In [12]:

```

numeric_features = ['Year', 'BMI',
                    'Adult_Mortality', 'Infant_Deaths', 'Alcohol', 'Percentage_Exp',
                    'HepatitisB', 'Under_Five_Deaths', 'Polio',

```



```

        'Diphtheria', 'HIVAIDS', 'GDP', 'Population', 'thinness_1to19_years'
    ,
        'thinness_5to9_years', 'Income_Comp_Of_Resources', 'Schooling',
        'Measles']
categorical_features = ['Country', 'Status']

```

In [13]:

```

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder

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

```

In [14]:

```

from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler

numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median'))

])

```

In [15]:

```

from sklearn.compose import ColumnTransformer

preprocessor = ColumnTransformer(
    transformers=[
        ('cat', categorical_transformer, categorical_features),
        ('num', numeric_transformer, numeric_features)
    ]
)

```

In [16]:

```

l = life.columns
lb = LabelEncoder()
for i in l:
    life[i] = lb.fit_transform(life[i])

```

In [28]:

```

model = Pipeline([
    ('preprocessor', preprocessor),
    ('RFRegressor', RandomForestRegressor())
])

```

In [29]:

```

model.fit(X_train,Y_train)

```

Out[29]

In [30]:

```

predict= model.predict(X_test)
r2_score(predict, Y_test)

```

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

```
from watson_machine_learning_client import WatsonMachineLearningAPIClient
```

In [33]:

```
wml_credentials = {  
    "apikey": "mR6nWB1g5T64XP9sM6BwCupvertTiZa0G9u-mm12skHv",  
    "instance_id": "3387f989-9275-4849-9c95-e5d0fc87c1f0",  
    "url": "https://eu-gb.ml.cloud.ibm.com"  
}
```

In [34]:

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

In [35]:

```
model_props = {client.repository.ModelMetaNames.AUTHOR_NAME: "Sricharan",  
               client.repository.ModelMetaNames.AUTHOR_EMAIL: "srichunln@gmail.com",  
               client.repository.ModelMetaNames.NAME: "LifeExpectancy"}
```

In [36]:

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

In [37]:

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

In [40]:

```
published_model_uid  
client.deployments.list()  
client.deployments.delete('4c85d2ca-94ff-4886-aed6-1ef719ae7a74')  
deployment = client.deployments.create(published_model_uid, name="life")  
scoring_endpoint = client.deployments.get_scoring_url(deployment)
```

In [43]:

```
scoring_endpoint
```

