REPORT

Predicting Life Expectancy using Machine Learning

Abraham Tony

Email id: abraham.tony@ieee.org
Github: https://github.com/abru1999
Project Link:
https://node-red-xzywz.eu-gb.mybluemix.net/ui/#!/0?socketid=sle9RIUeAYvAKxQ6AAA
B

Contents

1.INTRODUCTION

1.1. Overview

Life expectancy is one of the most important factors in end-of-life decision making.
The main objective of the project is to predict the life expectancy of a person depending on several factors based on an individual or the residing country. Factors like the GDP of the country, health care facility system, quality of life, mental and physical illness, age, gender, education and other regional, demographic and economic factors are considered to predict the lifespan of the person using machine learning algorithms.

1.2. Purpose

The purpose is to predict Life Expectancy by looking at the positive and negatively correlated factors to improve the Life Quality. By making changes in lifestyle, a person can live a long, healthy and good quality life. This will also benefit the country by increasing manpower that will contribute to the economical growth. We should take full advantage of this new era advanced technology to improve the future by predicting it in the present.

2. LITERATURE SURVEY

2.1. Existing Problem

As we all know, Life expectancy is one of the most important factors in end-of-life decision making. So, using the certain factors like Schooling, GDP, Adult Mortality Rate, Child Date, etc. life expectancy is predicted. All the factors are negatively or positively correlated.

When you are deciding when to start receiving retirement benefits, one important factor to take into consideration is how long you might live.
These country dependent factors can also be an important feature to predict the life expectancy of an individual. So we need more data to predict more accurately.

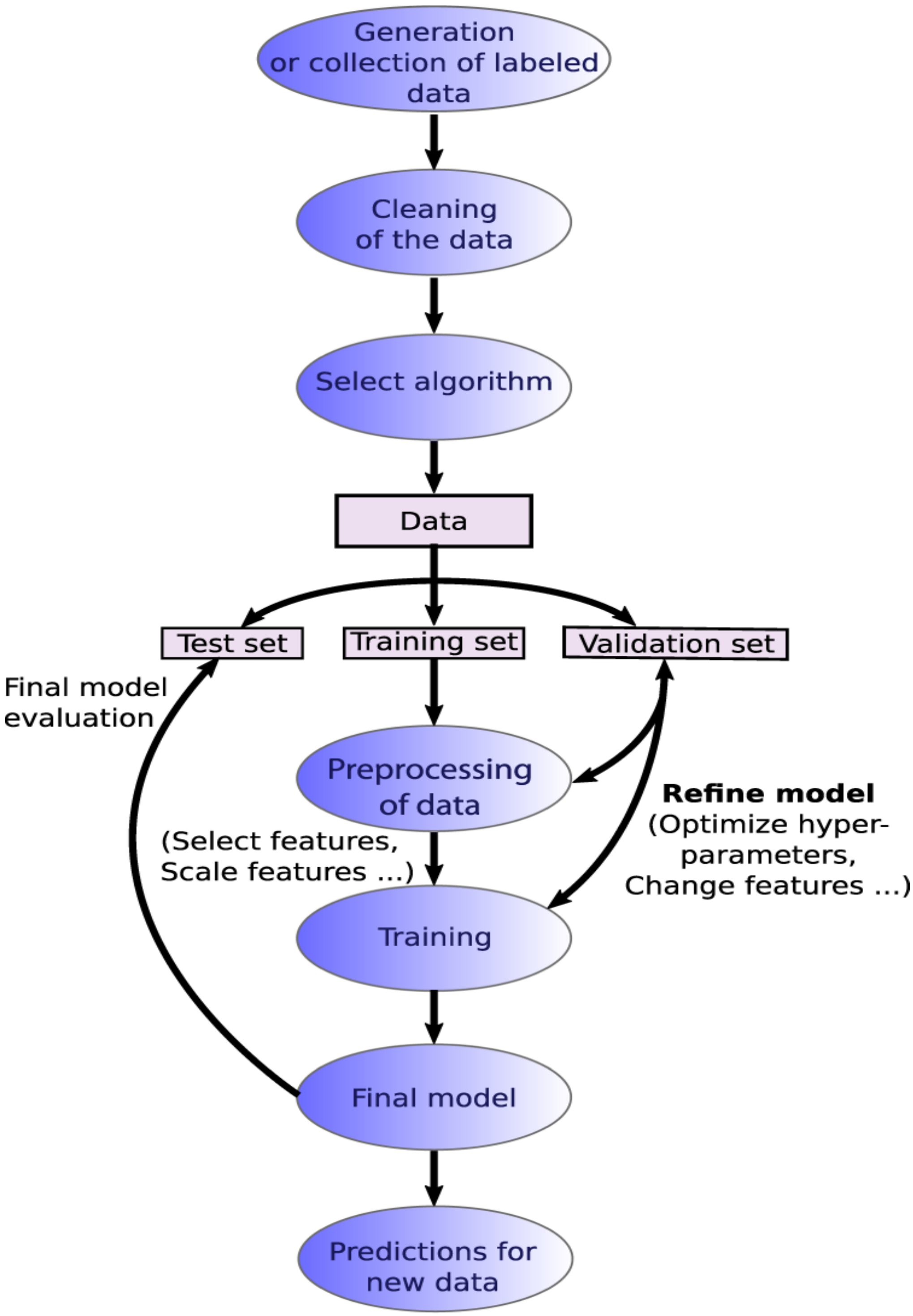
2.2 Proposed Solution

Using this model, life expectancy of a person can be predicted by taking some input features from the user.
Life Expectancy depends on the following features-

- Country
- Status
- Life Expectancy
- Adult Mortality
- 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

3. Theoretical Analysis

3.1 Block Diagram



3.2 Software Designing

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

4.EXPERIMENTAL INVESTIGATIONS

Data was collected from “https://www.kaggle.com/kumarajarshi/life-expectancy-who/data” and then pre-processed so that it is understood by the Machine Learning Algorithms Properly.

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

Out[4]:

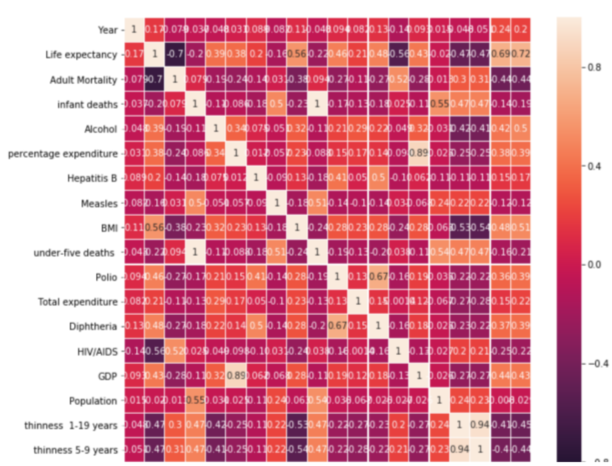
	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Pop
0	Afghanistan	2015	Developing	65.00000	263.00000	62	0.01000	71.27962	65.00000	1154	...	6.00000	8.16000	65.00000	0.10000	584.25921	3373649
1	Afghanistan	2014	Developing	59.90000	271.00000	64	0.01000	73.52358	62.00000	492	...	58.00000	8.18000	62.00000	0.10000	612.69651	32758
2	Afghanistan	2013	Developing	59.90000	268.00000	66	0.01000	73.21924	64.00000	430	...	62.00000	8.13000	64.00000	0.10000	631.74498	3173168
3	Afghanistan	2012	Developing	59.50000	272.00000	69	0.01000	78.18422	67.00000	2787	...	67.00000	8.52000	67.00000	0.10000	669.95900	369695
4	Afghanistan	2011	Developing	59.20000	275.00000	71	0.01000	7.09711	68.00000	3013	...	68.00000	7.87000	68.00000	0.10000	63.53723	297859

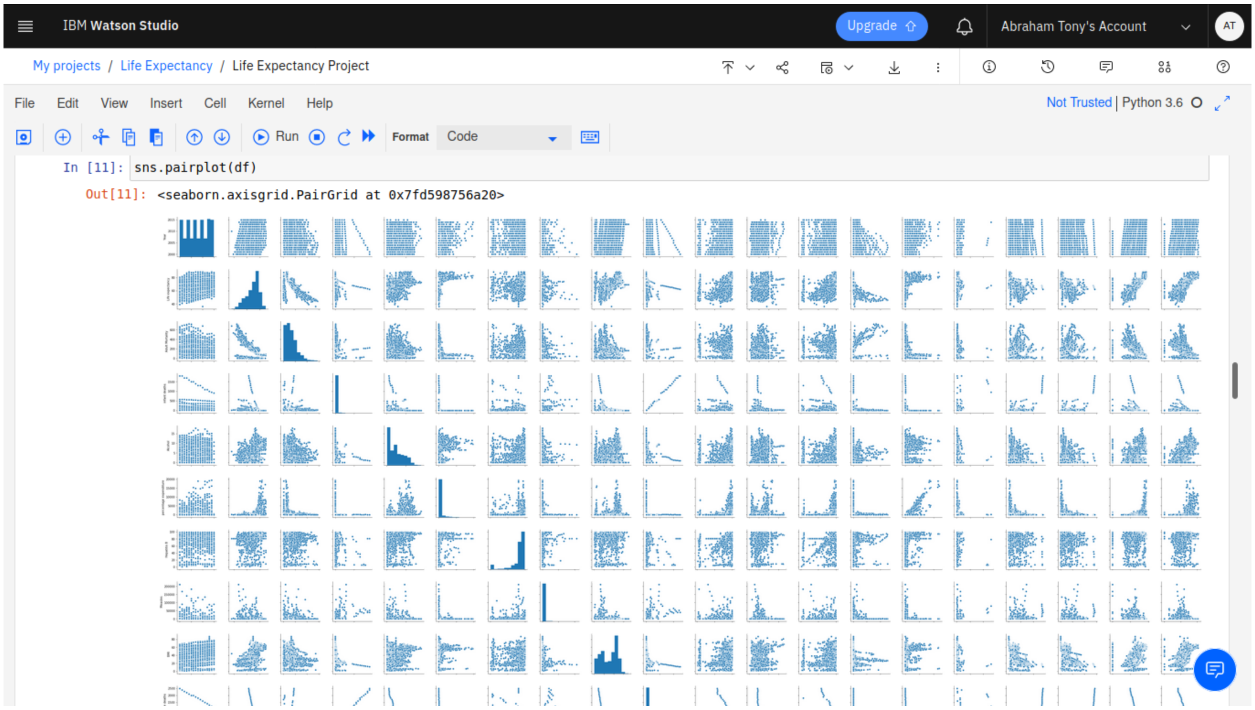
5 rows × 22 columns

Data Visualisation

```
In [10]: df_kor=df.corr()
         plt.figure(figsize=(10,10))
         sns.heatmap(df_kor,vmin=-1,vmax=1,annot=True,linewidth=0.1)
```

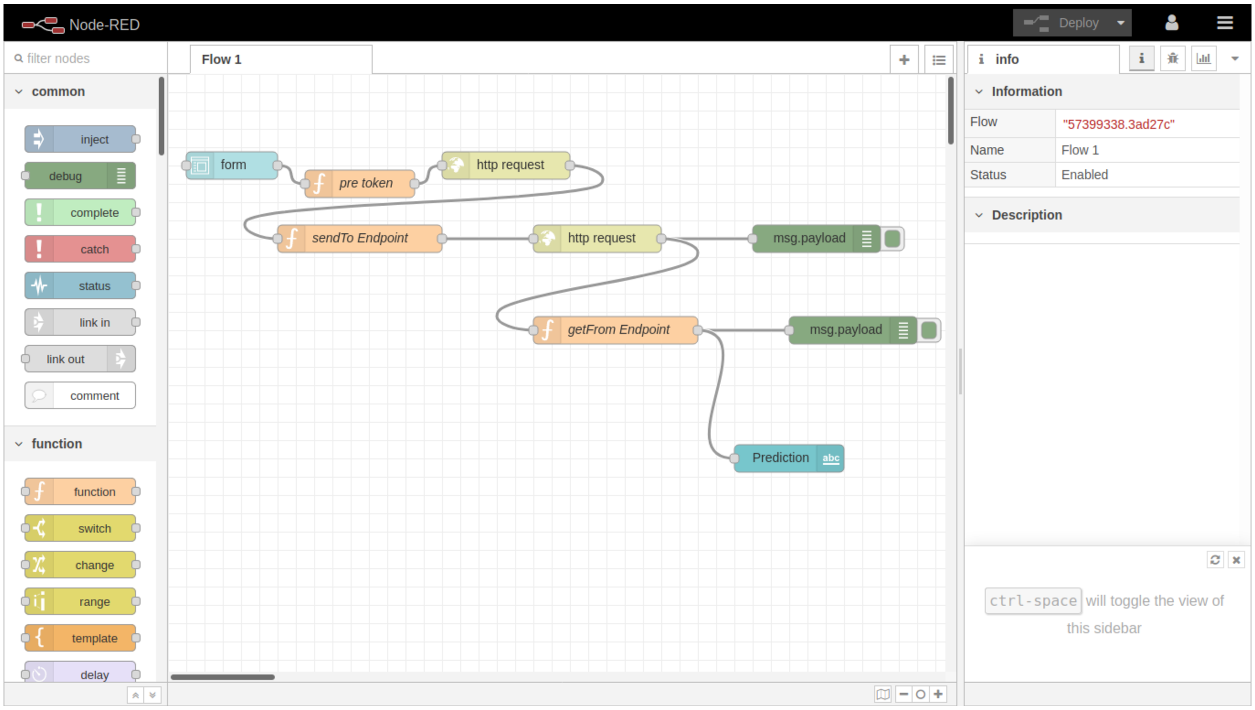
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd59ac1ad68>





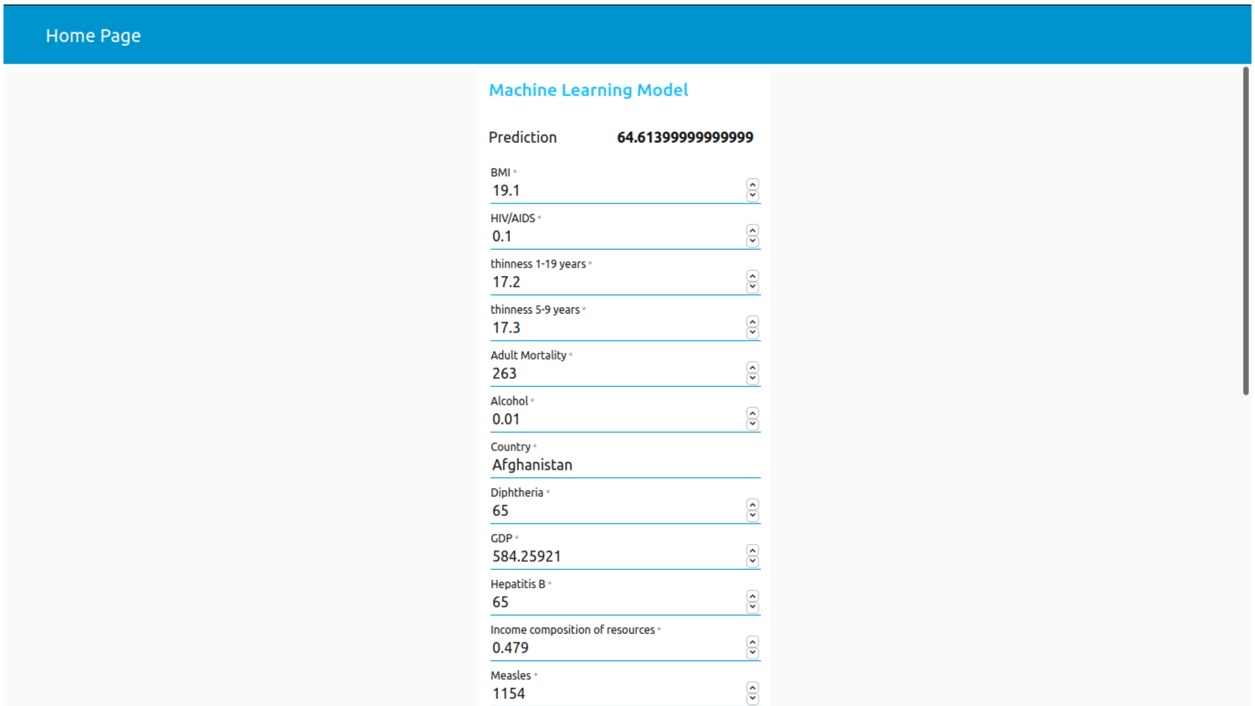
Then, different Regression Algorithms were applied and then accuracy is checked for each, so as to find the best fitted algorithm. Fine Tuning was done, in order to find the best parameters so that we get the best possible accuracy.

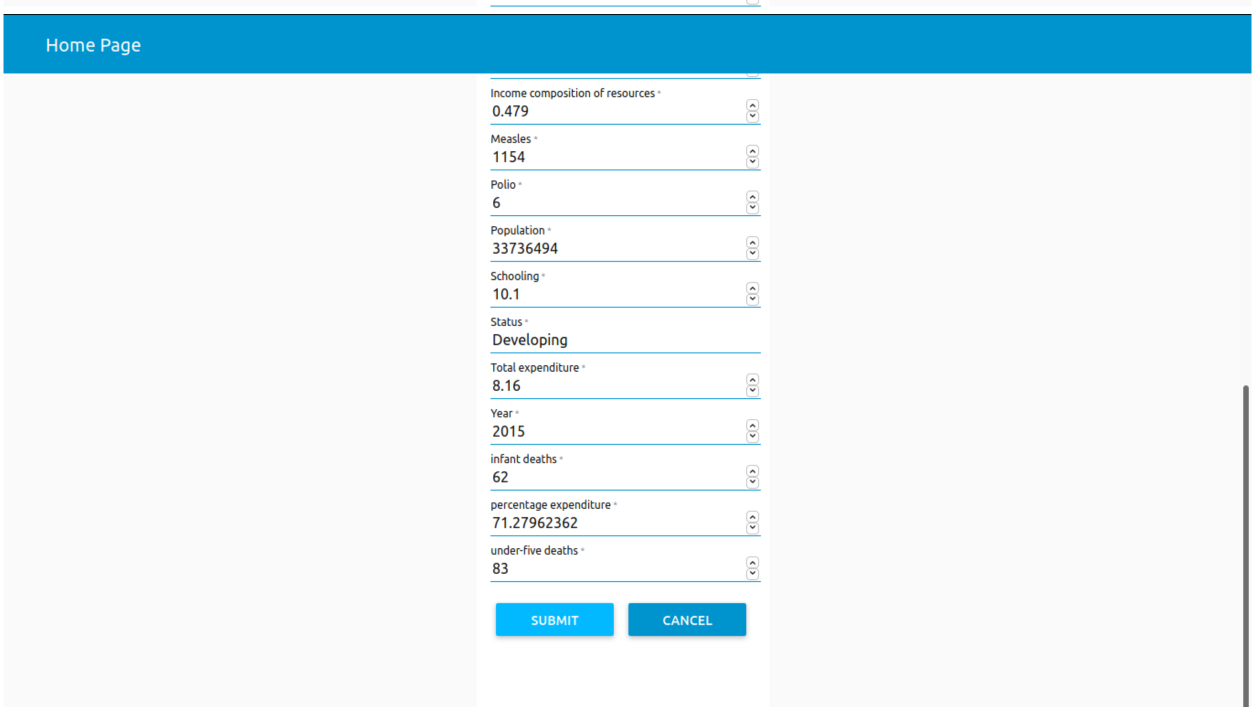
5. FLOWCHART



Node Red Flow

6. RESULTS


Home Page
Machine Learning Model
Prediction 64.61399999999999
BMI
19.1
HIV/AIDS
0.1
thinness 1-19 years
17.2
thinness 5-9 years
17.3
Adult Mortality
263
Alcohol
0.01
Country
Afghanistan
Diphtheria
65
GDP
584.25921
Hepatitis B
65
Income composition of resources
0.479
Measles
1154


Home Page
Income composition of resources
0.479
Measles
1154
Polio
6
Population
33736494
Schooling
10.1
Status
Developing
Total expenditure
8.16
Year
2015
infant deaths
62
percentage expenditure
71.27962362
under-five deaths
83
SUBMIT
CANCEL

7. ADVANTAGES & DISADVANTAGES

Advantages:

- Life Expectancy can be predicted depending on certain parameters with great accuracy.
- Benefit the country's growth.

Disadvantages:

- Though, the accuracy of the model is very high. Still there is some chance that the does not give the exact Life Expectancy.
- 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

Thus, we have developed a model that will predict the life expectancy of a person living in a specific region. Various factors like Adult Mortality, Population, Under 5 Deaths, Thinness 1-5 Years, Alcohol, HIV, Hepatitis B, GDP,Percentage Expenditure and many more play an important role in the prediction.

10. FUTURE SCOPE

Look at class within a particular country and see if these same factors are same in determining life expectancy for an individual. The accuracy of the model can be increased. This can be done by training more data. Also, the website can be added with many more features to improve the user experience.

11. BIBILOGRAPHY

- https://www.kaggle.com/kumarajarshi/life-expectancy-who/data
- https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

APPENDIX

A. SOURCE CODE

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
pd.options.display.float_format='{:.5f}'.format
import warnings
import math
#import libraries for pipelining
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.compose import ColumnTransformer
#import libraries for train and test
from sklearn.model_selection import train_test_split
#import ExtraTreesRegressor for model fit and prediction
from sklearn.ensemble import ExtraTreesRegressor
#import libraries for accuracy and error calculation
from sklearn.metrics import mean_squared_error, r2_score
#import libraries for model building and deployment
from watson_machine_learning_client import WatsonMachineLearningAPIClient

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

def __iter__(self): return 0


df = pd.read_csv(body)
df.head()
df.columns
df=df.rename(columns={'Life  expectancy  ':'Life  expectancy','Measles  ':'Measles',
```

```
BMI ':'BMI','Diphtheria ':'Diphtheria',' HIV/AIDS':'HIV/AIDS',' thinness 1-19 years':'thinness  1-19 years',' thinness 5-9 years':'thinness 5-9 years'})
df.isnull().sum()
df=df.fillna(df.mean())
df.isnull().sum()
df_kor=df.corr()
plt.figure(figsize=(10,10))
sns.heatmap(df_kor,vmin=-1,vmax=1,annot=True,linewidth=0.1)
sns.pairplot(df)

Y=df['Life expectancy']
X=df[df.columns.difference(['Life expectancy'])]
df.select_dtypes(include=['int64', 'float64']).columns
df.select_dtypes(include=['object', 'bool']).columns
categorical_features = ['Country', 'Status']
categorical_feature_mask = X.dtypes==object
categorical_features = X.columns[categorical_feature_mask].tolist()
#DEFINE CATEGORICAL PIPELINE
categorical_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(handle_unknown='ignore')),
])
numeric_features = ['Year','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']
numeric_feature_mask = X.dtypes!=object
numeric_features = X.columns[numeric_feature_mask].tolist()
#DEFINE NUMERIC PIPELINE
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler()),
])
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ]
)
```

```
ExtraTreeRegressor = Pipeline([
    ('preprocessor', preprocessor),
                        ('ExtraTreeRegressor',      ExtraTreesRegressor(n_estimators=100,
random_state=0))
])
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)

reg = ExtraTreeRegressor.fit(X_train, Y_train)
test_pred=reg.predict(X_test)
print(test_pred)
print('Mean squared error: ',mean_squared_error(Y_test, test_pred))
print('R2 score: ',r2_score(Y_test, test_pred)*100)

wml_credentials={
 "apikey": "qFow6rcn7lJa3-qBCJ-fSynVVqKAh_sx7yLLFo8wYQuV",
 "instance_id": "cb28290c-4ed7-4288-a580-a3291bffd339",
  "url": "https://eu-gb.ml.cloud.ibm.com"
}
client = WatsonMachineLearningAPIClient(wml_credentials)
print(client.service_instance.get_url())
model_props  =  {client.repository.ModelMetaNames.AUTHOR_NAME:  "Abraham
Tony",
                                          client.repository.ModelMetaNames.AUTHOR_EMAIL:
"abraham.tony@ieee.org",
            client.repository.ModelMetaNames.NAME: "LifeExpectancy"}
#STORE THE MACHINE LEARNING MODEL
model_artifact=client.repository.store_model(ExtraTreeRegressor,
meta_props=model_props)

#GET MODEL UID
model_uid = client.repository.get_model_uid(model_artifact)
#DEPLOY THE MODEL
create_deployment                  =                 client.deployments.create(model_uid,
name="LifeExpectancyPrediction")

scoring_endpoint = client.deployments.get_scoring_url(create_deployment)
print(scoring_endpoint)
```