LIFE EXPECTANCY PREDICTION OF HUMAN

Data Science With Python Lab Project Report

Bachelor
in
Computer Science

BY
LIKHITH AND SHAHUL
S200624
S200778

Rajiv Gandhi University Of Knowledge And Technologies
S.M. Puram , Srikakulam-532410
Andhra Pradesh,India

ABSTRACT

Our data science project "the life expentancy prediction " focuses on analyzing a dataset for predicting the age factor of a human. By considering various statistical and machine learning techniques, we aim to discover the factors of human survival. We look into lots of data like adult mortality,alcohol consumption, BMI, healthcare, habits and many more.And even where people live to understand what affects how long they live. By understanding this,governments and organizations prepare for the changing needs of aging people and ensure sustainable social security systems, doctors treat people better and it allow people to know about how their behaviors and environments impact their longevity,this can inspire people to acquire healthier habits and improve overall quality of life.. This can all lead to people living longer and happier lives!

Contents

1 Introduction

1.1 Introduction To Our Project

In our data science project, we're looking into the concept of lifespan of human life. By analyzing the dataset on life expectancy, we aim to discover the factors that leads to longer and healthier lives. we hope to gain a deeper understanding of what influences how long we live and how we can improve overall well-being.our project has the potential to create awareness for healthcare professionals like doctors, policymakers, and individuals alike, ultimately contributing to better health outcomes and quality of life for everyone.

1.2 Applications

The project on human life expectancy has several real-world applications across various fields:

- **Healthcare Planning:**Governments and healthcare organizations can use insights from the project to organise healthcare initiatives, and plan for the healthcare needs of aging populations. This includes retirement plans, and healthcare service distribution.

- **Disease Prevention and Management:** Understanding the factors influencing life expectancy can aid in the prevention and management of diseases. By identifying high-risk populations and targeting interventions accordingly, healthcare professionals can implement preventive measures, screenings, and treatments to reduce the burden of diseases associated with premature mortality.

- **Policy Development:** Policymakers can utilize findings from the project to develop evidence-based policies aimed at improving public health outcomes and addressing health disparities. This includes policies related to healthcare access, education, social support systems, urban planning, and environmental regulations.

1.3 Motivation Towards The Project

1. Public Health Awareness: Knowing what affects how long people live helps us create strategies to keep everyone healthy and living longer. Finding out the main things that make people live longer can help us fix unfair health differences and encourage healthier habits

2. Policy Implications: Awareness gained from the project can inform policymakers about the effectiveness of existing healthcare policies and guide the development of new strategies to enhance healthcare access, improve healthcare quality, and address social determinants of health.

3. Medical Advances: Discovering correlations between life expectancy and various health indicators can drive medical research and innovation. By identifying risk factors for premature mortality, researchers can develop new treatments, preventive measures, and screening protocols to combat diseases and prolong life.

1.4 Problem Statement

Our project aims to develop machine learning model which can predict the life expectancy of human.The dataset for this project is taken from KAGGLE website.Our project will look into the factors affecting human life such as adult mortality,alcohol consumption, BMI, healthcare, habits and many more.By analzing these factors machine learning model should predict life expectancy.

2 Code

2.1 Pandas

- Pandas is a popular open source library in python
- Pandas allows us to analyze big data and make conclusions based
- It is uses for data manipulation and data analysis.
- Pandas can clean messy data sets and make them readable and relavent
- It has functions for data analyzing,cleaning,exploring,maniplating. on stastical theories.

Importing Essential Libraries

```
import pandas as pd
import numpy as np
import warnings
warnings.simplefilter('ignore')
```

Importing csv file to notebook using pandas

```
df=pd.read_csv("LifeExpectancyData.csv")
df
```

Output :

In [99]: df

Out[99]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.25
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.69
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.74
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.95
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.53
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	...	67.0	7.13	65.0	33.6	454.36
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	...	7.0	6.52	68.0	36.7	453.35
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	...	73.0	6.53	71.0	39.8	57.34
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	...	76.0	6.16	75.0	42.1	548.58
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	...	78.0	7.10	78.0	43.5	547.35

2938 rows × 22 columns

The fig 2.1 describes the imported dataset values in the form of DataFrame

Data Cleaning

```
df.isna().sum()
df.interpolate(inplace=True)
```

Head and Tail

```
df.head()
```

Output :

In [100]: df.head()

Out[100]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GD
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.25921
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.69651
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.74497
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.9590
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.53723

5 rows × 22 columns

The fig 2.2 describes head() function in Pandas which displays the top 5 rows of a DataFrame

```
df.head(7)
```

Output :

```
In [72]: df.head(7)
Out[72]:
```

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GD
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.25921
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.69651
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.74497
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.95900
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.53723
5	Afghanistan	2010	Developing	58.8	279.0	74	0.01	79.679367	66.0	1989	...	66.0	9.20	66.0	0.1	553.32894
6	Afghanistan	2009	Developing	58.6	281.0	77	0.01	56.762217	63.0	2861	...	63.0	9.42	63.0	0.1	445.89329

7 rows × 22 columns

The fig 2.3 describes head(7) function in Pandas which displays the top 7 rows of a DataFrame

```
df.tail(7)
```

Output :

```
Out[101]:
```

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	(
2931	Zimbabwe	2006	Developing	45.4	7.0	28	4.57	34.262169	68.0	212	...	71.0	5.12	7.0	26.8	414.796
2932	Zimbabwe	2005	Developing	44.6	717.0	28	4.14	8.717409	65.0	420	...	69.0	6.44	68.0	30.3	444.765
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	...	67.0	7.13	65.0	33.6	454.366
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	...	7.0	6.52	68.0	36.7	453.351
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	...	73.0	6.53	71.0	39.8	57.348
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	...	76.0	6.16	75.0	42.1	548.587
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	...	78.0	7.10	78.0	43.5	547.358

7 rows × 22 columns

The fig 2.4 describes tail() function in Pandas which dis- plays the last 5 rows of a DataFrame

shape

df.shape

Output :

```
Out[75]: (2938, 22)
```

The fig 2.6 describes shape attribute in Pandas which dis- plays the number of elements in each dimension.

info

df.info()

Output :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Country                          2938 non-null   object
 1   Year                             2938 non-null   int64
 2   Status                           2938 non-null   object
 3   Life expectancy                  2938 non-null   float64
 4   Adult Mortality                  2938 non-null   float64
 5   infant deaths                    2938 non-null   int64
 6   Alcohol                          2938 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2938 non-null   float64
 9   Measles                          2938 non-null   int64
 10   BMI                             2938 non-null   float64
 11  under-five deaths                2938 non-null   int64
 12  Polio                            2938 non-null   float64
 13  Total expenditure                2938 non-null   float64
 14  Diphtheria                       2938 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2938 non-null   float64
 17  Population                       2938 non-null   float64
 18   thinness  1-19 years            2938 non-null   float64
 19   thinness 5-9 years              2938 non-null   float64
 20  Income composition of resources  2938 non-null   float64
 21  Schooling                        2938 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

The fig 2.7 describes info() function in Pandas which displays the information about the DataFrame. The information contains columns,column labels,column datatypes,range index,number of cells in each column.

describe

```
df.describe()
```

Output :

Out[79]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	exp
count	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.00000	2938.000000	2938.000000	293
mean	2007.518720	69.214874	164.744554	30.303948	4.550179	738.251295	78.528421	2419.592240	38.29869	42.035739	82.474643	5
std	4.613841	9.510819	124.226596	117.926501	3.994827	1987.914858	25.183772	11467.272489	19.99140	160.445548	23.420689	2
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.00000	0.000000	3.000000	0
25%	2004.000000	63.100000	74.000000	0.000000	0.902500	4.685343	71.000000	0.000000	19.32500	0.000000	78.000000	4
50%	2008.000000	72.000000	144.000000	3.000000	3.702500	64.912906	89.000000	17.000000	43.25000	4.000000	93.000000	5
75%	2012.000000	75.600000	228.000000	22.000000	7.550000	441.534144	96.000000	360.250000	56.10000	28.000000	97.000000	7
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.30000	2500.000000	99.000000	17

The fig 2.8 describes describe() function in Pandas which is a convenient way to get a quick overview of our Data.It pro- vides the count,mean,standard deviation,minimum,maximum etc...

columns

```
df.columns
```

Output :

```
Out[81]: Index(['Country', 'Year', 'Status', 'Life expectancy ', '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'],
      dtype='object')
```

The fig 2.9 describes columns attribute in Pandas which display the column labels of a DataFrame.

Retriving the values based on the condition

```
df[df["Life expectancy "]>70]
```

Output :

Out[83]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	C
16	Albania	2015	Developing	77.8	74.0	0	4.60	364.975229	99.00	0	...	99.0	6.00	99.0	0.1	3954.227
17	Albania	2014	Developing	77.5	8.0	0	4.51	428.749067	98.00	0	...	98.0	5.88	98.0	0.1	4575.763
18	Albania	2013	Developing	77.2	84.0	0	4.76	430.876979	99.00	0	...	99.0	5.66	99.0	0.1	4414.723
19	Albania	2012	Developing	76.9	86.0	0	5.14	412.443356	99.00	9	...	99.0	5.59	99.0	0.1	4247.614
20	Albania	2011	Developing	76.6	88.0	0	5.37	437.062100	99.00	28	...	99.0	5.71	99.0	0.1	4437.178
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2885	Viet Nam	2004	Developing	74.2	136.0	29	2.86	0.000000	94.00	217	...	96.0	5.90	96.0	0.2	1380.729
2886	Viet Nam	2003	Developing	74.0	137.0	30	2.19	0.000000	78.00	2297	...	96.0	4.84	99.0	0.2	1377.546
2887	Viet Nam	2002	Developing	73.8	137.0	30	2.03	0.000000	75.75	6755	...	92.0	4.70	75.0	0.2	1374.363
2888	Viet Nam	2001	Developing	73.6	138.0	32	1.84	0.000000	73.50	12058	...	96.0	5.17	96.0	0.1	1371.180
2889	Viet Nam	2000	Developing	73.4	139.0	33	1.60	0.000000	71.25	16512	...	96.0	4.89	96.0	0.1	1367.998

1622 rows × 22 columns

The fig 2.10 describes retriving the data of a delivery person whose age is 32.

Mode

```
df["Life expectancy "].mode()
```

Output :

```
Out[85]: 0    73.0
         Name: Life expectancy , dtype: float64
```

The fig 2.11 describes the mode value of a particular column

value counts

```
df["Country"].value_counts()
```

Output :

```
Out[88]: Country
         Afghanistan            16
         Peru                   16
         Nicaragua              16
         Niger                  16
         Nigeria                16
                                ..
         Niue                    1
         San Marino              1
         Nauru                   1
         Saint Kitts and Nevis   1
         Dominica                1
         Name: count, Length: 193, dtype: int64
```

The fig 2.12 describes the count of unique values of a particular column

groupby

```
df1=df.groupby(["Country"])
df1.first()
```

Output :

Out[93]:

Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	
Afghanistan	2015	Developing	65.0	263.0	62	0.010	71.279624	65.0	1154	19.1	...	6.0	8.160	65.0	0.1	58
Albania	2015	Developing	77.8	74.0	0	4.600	364.975229	99.0	0	58.0	...	99.0	6.000	99.0	0.1	395
Algeria	2015	Developing	75.6	19.0	21	1.835	0.000000	95.0	63	59.5	...	95.0	6.735	95.0	0.1	413
Angola	2015	Developing	52.4	335.0	66	4.290	0.000000	64.0	118	23.3	...	7.0	3.400	64.0	1.9	369
Antigua and Barbuda	2015	Developing	76.4	13.0	0	5.205	0.000000	99.0	0	47.7	...	86.0	4.165	99.0	0.2	1356
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
Venezuela (Bolivarian Republic of)	2015	Developing	74.1	157.0	9	3.840	0.000000	87.0	0	62.1	...	87.0	4.270	87.0	0.1	146
Viet Nam	2015	Developing	76.0	127.0	28	6.050	0.000000	97.0	256	17.5	...	97.0	6.305	97.0	0.1	141
Yemen	2015	Developing	65.7	224.0	37	0.805	0.000000	69.0	468	41.3	...	63.0	5.265	69.0	0.1	136
Zambia	2015	Developing	61.8	33.0	27	0.040	0.000000	9.0	9	23.4	...	9.0	4.565	9.0	4.1	131
Zimbabwe	2015	Developing	67.0	336.0	22	4.560	0.000000	87.0	0	31.8	...	88.0	6.800	87.0	6.2	11

193 rows × 21 columns

The fig 2.13 describes the groupby function in Pandas is used for grouping the data according to the categories and applying function to the categories.

Transform

```
df.groupby('Life expectancy ')[' BMI '].transform(lambda
    x:x-x.mean())
```

Output :

```
In [98]: df.groupby('Life expectancy ')[' BMI '].transform(lambda x:x-x.mean())

Out[98]: 0      -19.107692
         1        0.971429
         2        0.471429
         3       -1.420000
         4      -10.785714
                   ...
         2933     3.700000
         2934    -0.350000
         2935    -0.300000
         2936     9.100000
         2937     2.140000
         Name:  BMI , Length: 2938, dtype: float64
```

The fig 2.14 describes the transform method which allows us to execute for each value of the DataFrame

3 Approach To Your Project

3.1 Explain About Your Project

This project is focused on predicting life expectancy using various statistical and machine learning methods. The goal is to analyze a range of factors that influence life expectancy, such as adult mortality rates, lifestyle choices, economic factors, and healthcare availability. By accurately predicting life expectancy, this project aims to provide valuable insights for public health policy and individual healthcare planning.

3.2 Data Set

The Dataset for this Life expectancy Prediction project is taken from the Kaggle website. The dataset contains various features such as 'Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality','infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B', etc.

Country: The name of the country

Year: The calendar year the data corresponds to.

Status: Developed or Developing

Life expectancy: The average number of years a person

Adult Mortality: The probability of dying between 15 and 60 years.

Infant deaths: The number of infant deaths (under one year) per 1000.

Alcohol: Recorded per capita (15+) consumption (in liters of pure alcohol).

Percentage expenditure: Expenditure on health as a percentage of Gross Domestic Product per capita.

3.3 Prediction technique

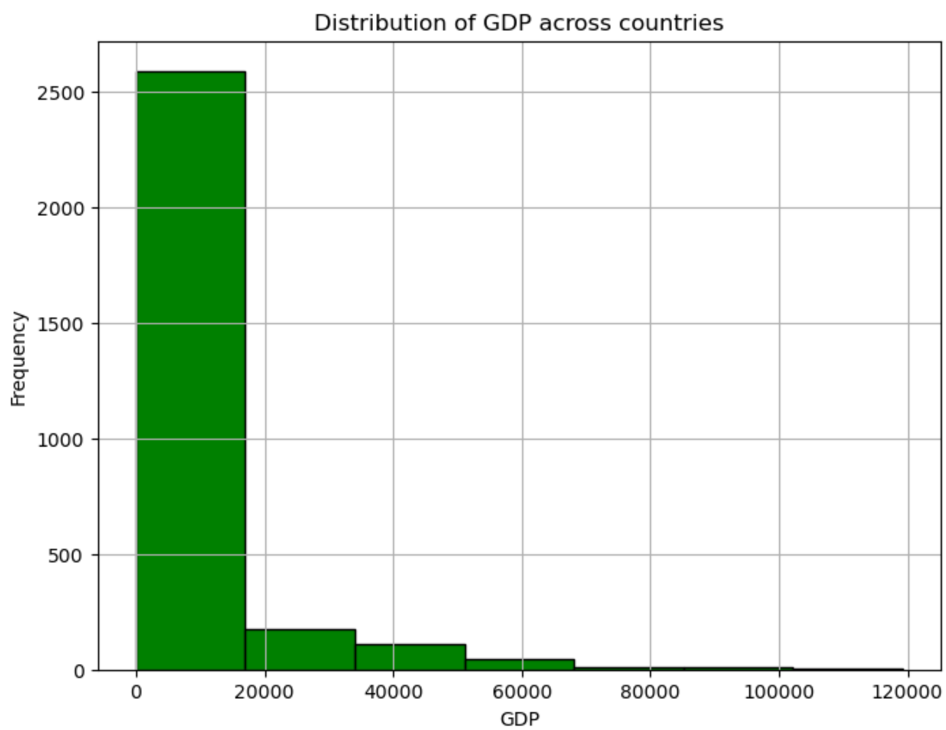
Our prediction techinques are Linear Regression ,DecisionTree and RandomForestRegressor.Linear Regression is a model that shows the relation between dependent and independent variables.we select a suitable regression model for our prediction that is RandomForest Regressor.we use Random forest as it predicts output with high accuracy, even for the large dataset it runs efficiently.

3.4 Graphs

```python
import matplotlib.pyplot as plt
import seaborn as sns
```

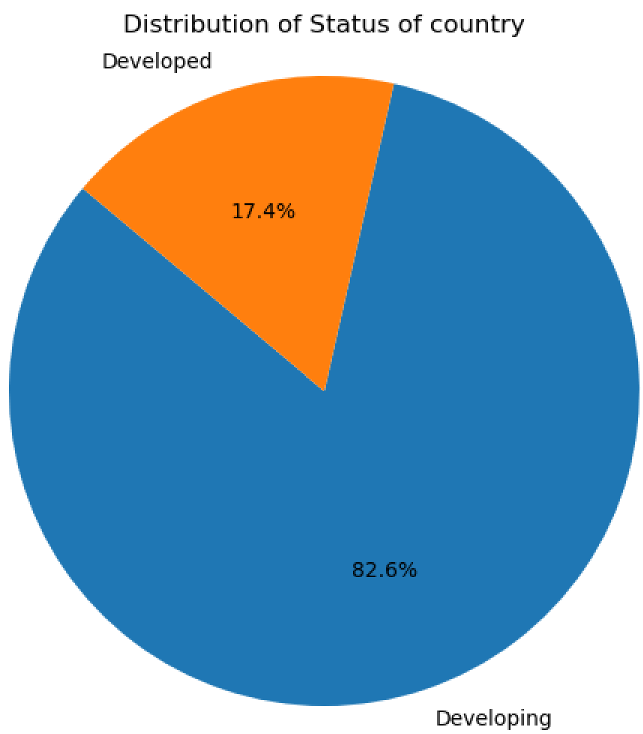
Histogram plot for GDP across countries

```python
# Plotting a basic histogram
plt.figure(figsize=(8, 6))
plt.hist(df['GDP'], bins=7, color='green', edgecolor='black')
plt.title('Distribution of GDP across countries')
plt.xlabel('GDP')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



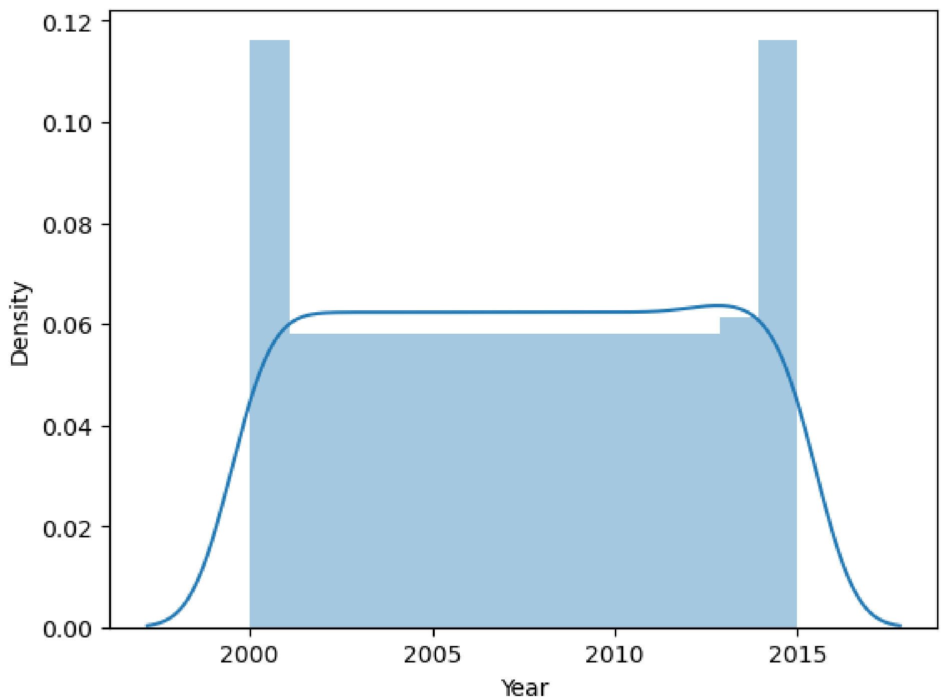
Pie chart for Country Status

```
Country_Status = df['Status'].value_counts()

# Plotting a basic pie chart
plt.figure(figsize=(8, 6))
plt.pie(Country_Status, labels=Country_Status.index,
autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Status of country')
plt.axis('equal')
plt.show()
```



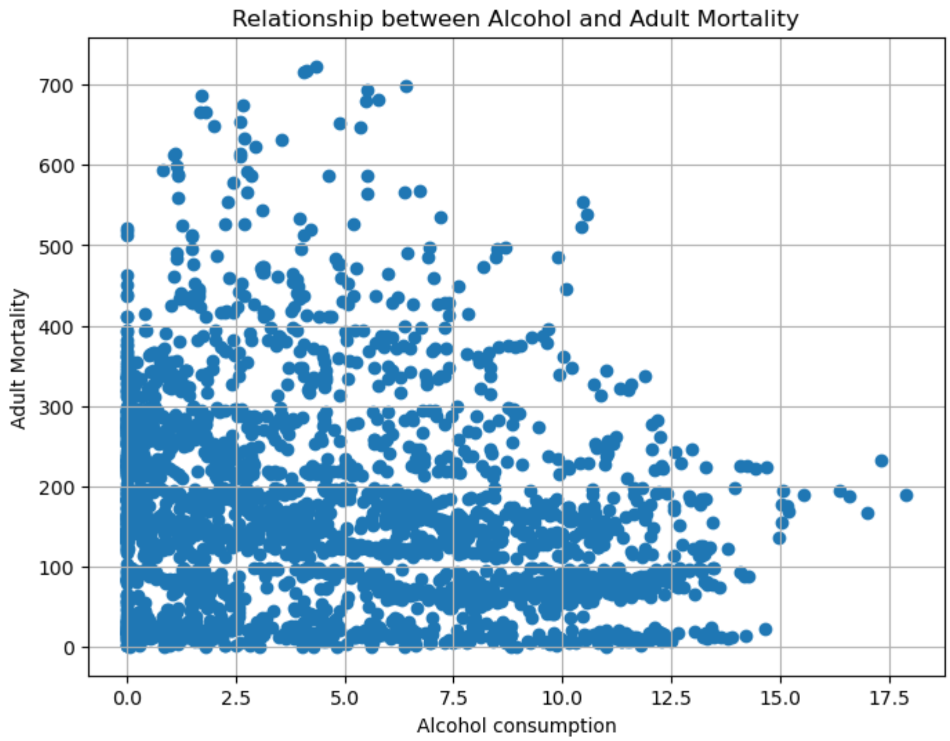
Distplot For Year

```
sns.distplot(df["Year"])
plt.show()
```



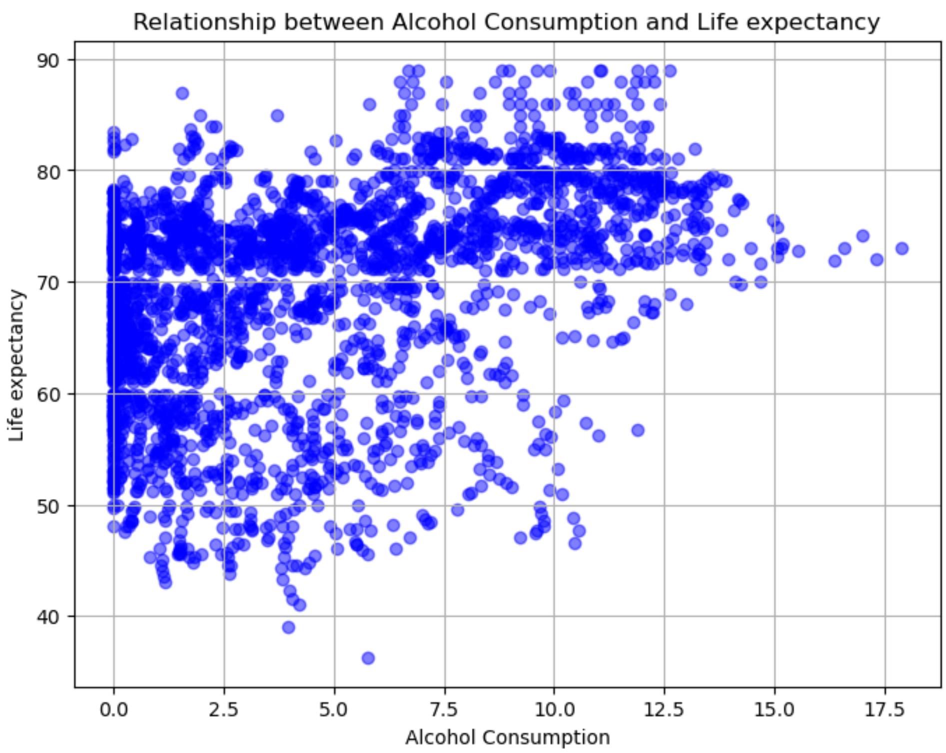
Scatter Plot Between Adult Mortality and alcohol consumption

```
plt.figure(figsize=(8, 6))
plt.scatter(df['Alcohol'], df['Adult Mortality'])
plt.title('Relationship between Alcohol and Adult Mortality')
plt.xlabel('Alcohol consumption')
plt.ylabel('Adult Mortality')
plt.grid(True)
plt.show()
```



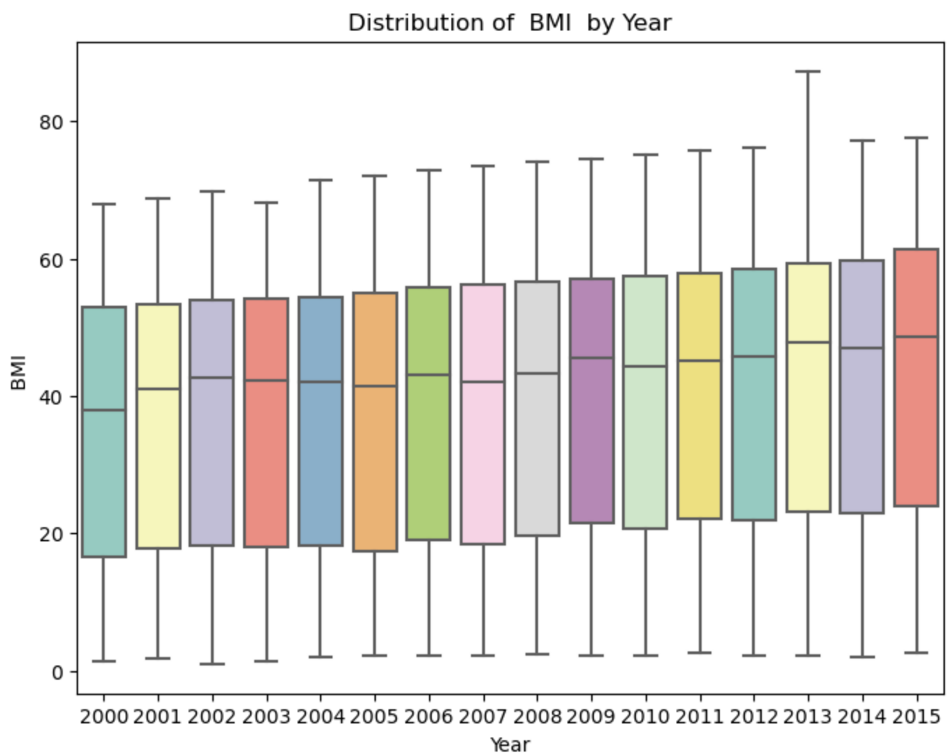
Scatter Plot Between alcohol consumption and life expectancy

```
import matplotlib.pyplot as plt
# Plotting a basic scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(df['Alcohol'], df['Life expectancy '],
    color='blue', alpha=0.5)
plt.title('Relationship between Alcohol Consumption and
    Life expectancy')
plt.xlabel('Alcohol Consumption')
plt.ylabel('Life expectancy')
plt.grid(True)
plt.show()
```



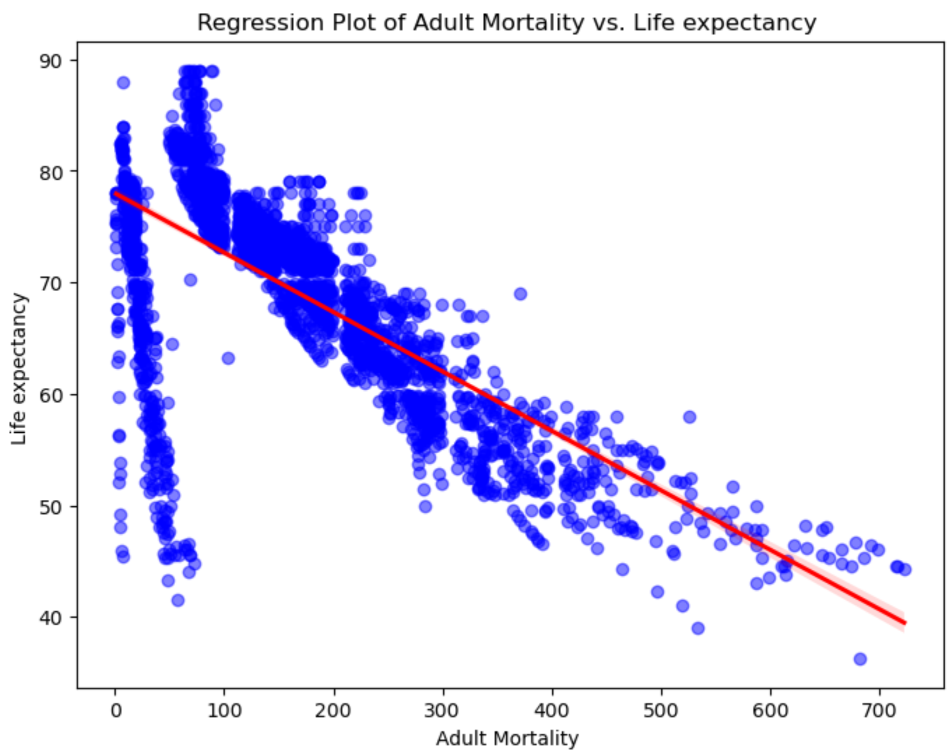
Box Plot for BMI and Year

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))

sns.boxplot(x='Year', y=' BMI ', data=df, palette='Set3')

plt.title('Distribution of  BMI  by Year')
plt.xlabel('Year')
plt.ylabel(' BMI  ')
plt.show()
```



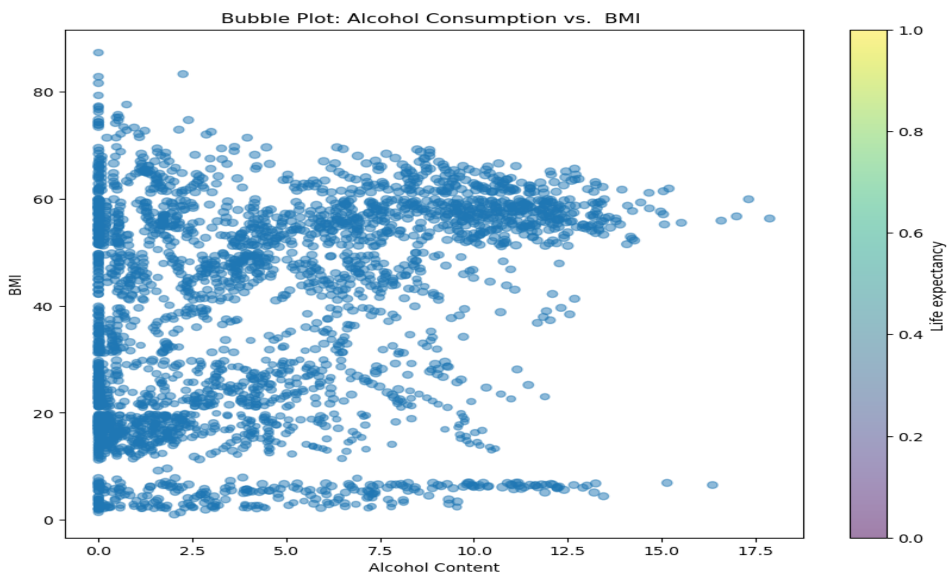
Regression Plot for Adult mortality and life expectancy

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
sns.regplot(x='Adult Mortality', y='Life expectancy ',
    data=df, scatter_kws={'alpha':0.5, 'color':'blue'},
    line_kws={'color':'red'})
plt.title('Reg Plot of Adult Mortality vs Life expectancy')
plt.xlabel('Adult Mortality ')
plt.ylabel('Life expectancy ')
plt.savefig('linear_graph.png', dpi=2000)
plt.show()
```



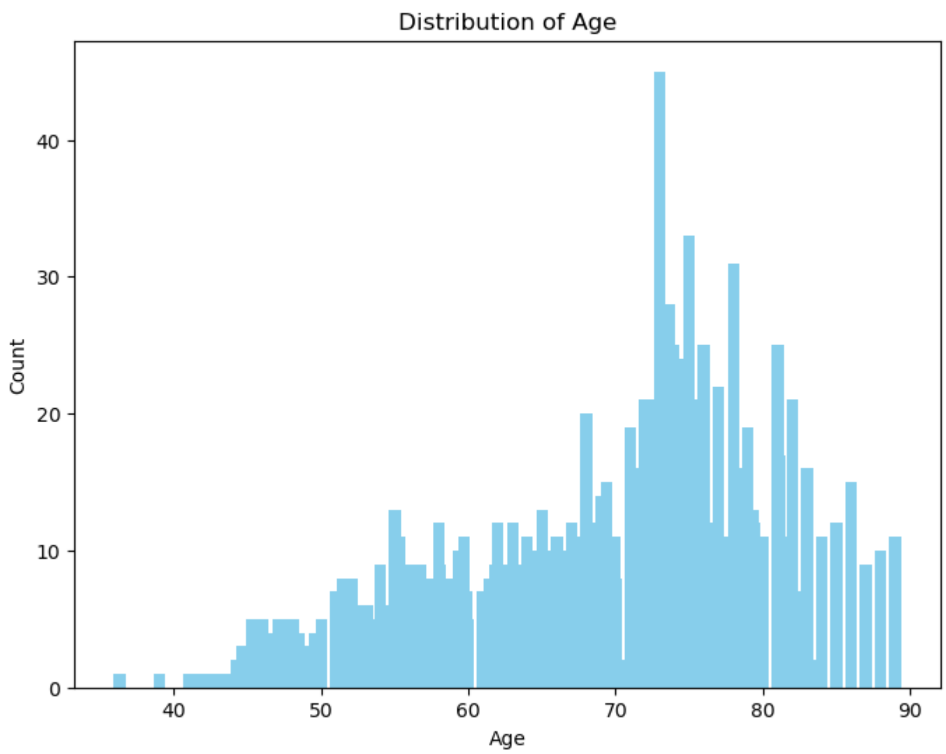
Bubble Plot for alcohol consumption,BMI and life expectancy

```
import matplotlib.pyplot as plt
x = df['Alcohol']  # x-axis: Alcohol content
y = df[' BMI ']  # y-axis:  BMI
z = df['Life expectancy ']  # bubble size: Life expectancy

plt.figure(figsize=(10, 8))
bubble = plt.scatter(x, y, s=z*0.5, alpha=0.5)

plt.title('Bubble Plot: Alcohol Consumption vs.  BMI ')
plt.xlabel('Alcohol Content')
plt.ylabel(' BMI ')

plt.colorbar(bubble, label='Life expectancy ')

plt.show()
```



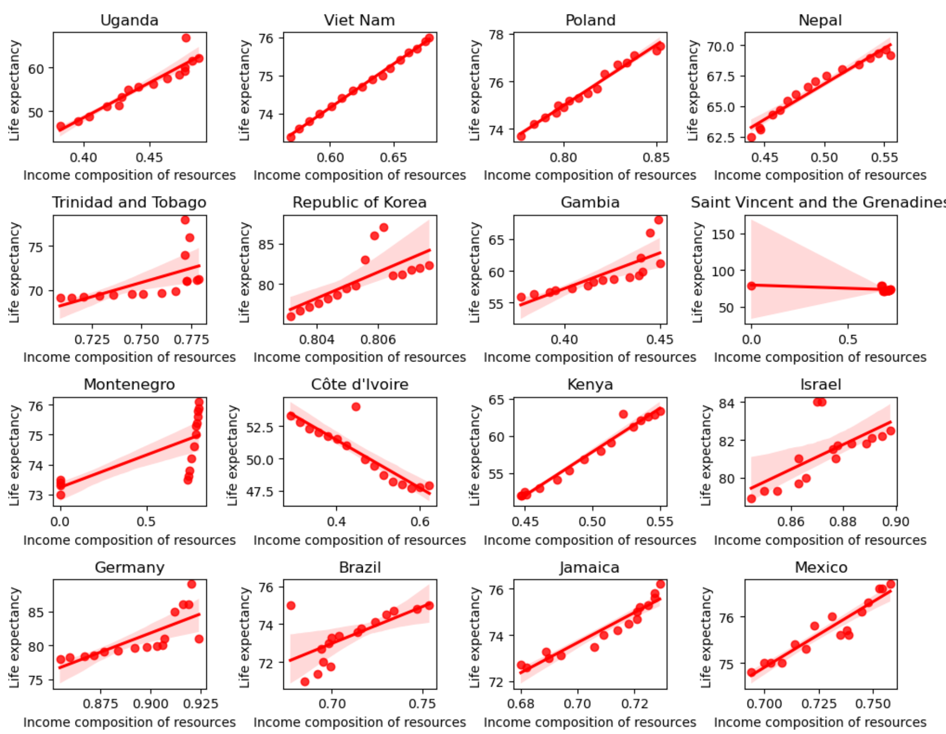
Bar Graph for Age Distribution

```
import matplotlib.pyplot as plt
Age_counts = df['Life expectancy '].value_counts()

plt.figure(figsize=(8, 6))
plt.bar(Age_counts.index, Age_counts.values, color='skyblue')

plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Count')

plt.show()
```



Regression Plots

```
fig, axs = plt.subplots(4, 4, figsize=(10, 8))
for Country, ax in zip(set(df["Country"]), axs.flat):
    Countries =df[df["Country"] == Country]
    sns.regplot(x =Countries['Income composition of resources'],
    y =Countries["Life expectancy "],
    color ='red',ax=ax).set_title(Country)
plt.tight_layout()
plt.show()
```



4 Machine Learning Models

Loading Data

```
import pandas as pd
df=pd.read_csv("LifeExpectancyData.csv")
df
```

Output :

In [99]: df

Out[99]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	0.1	584.25
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	0.1	612.69
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	0.1	631.74
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	0.1	669.95
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	0.1	63.53
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	...	67.0	7.13	65.0	33.6	454.36
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	...	7.0	6.52	68.0	36.7	453.35
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	...	73.0	6.53	71.0	39.8	57.34
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	...	76.0	6.16	75.0	42.1	548.58
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	...	78.0	7.10	78.0	43.5	547.35

2938 rows × 22 columns

Importing Essential Libraries

```
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.metrics import mean_absolute_error,
    mean_squared_error, r2_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
```

Data Separation as x and y:

```
x=df.drop(["Life expectancy",
"Country","Status","Year"],axis=1)
y=df["Life expectancy "]
```

Data Splitting:

```
x_train,x_test,y_train,y_test=
train_test_split(x,y,test_size=0.2,random_state=70)
```

Test and Train data:

```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

Output :
(2350, 18)
(2350,)
(588, 18)
(588,)

Model Building

4.1 Linear Regression

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. By fitting a linear equation to observed data, it aims to predict the value of the dependent variable based on the values of the independent variables. Simple linear regression involves a single independent variable, while multiple linear regression deals with multiple independent variables. Linear regression finds applications in predictive analysis, forecasting, and identifying the strength and nature of relationships between variables.

Training the model

```
reg=linear_model.LinearRegression()
reg.fit(x_train,y_train)
```

Output :

```
In [77]: reg.fit(x_train,y_train)
Out[77]: ▾ LinearRegression
         LinearRegression()
```

Making Prediction

```
y_pred=reg.predict(x_test)
y_pred
```

Output :

```
In [78]: pred=reg.predict(x_test)
         pred

Out[78]: array([72.6649736 , 76.06575987, 57.82057197, 74.0995542 , 75.99819589,
                75.80507699, 57.15881986, 71.69394636, 68.14785422, 46.6310533 ,
                80.53243638, 70.60201175, 59.94838741, 70.51472431, 60.52597098,
                70.79286426, 79.2978651 , 84.47888559, 65.32641135, 76.31366853,
```

Evaluating Model Performance:

```
from sklearn.metrics import r2_score,
mean_absolute_error, mean_squared_error,r2_score
r2 =r2_score(y_test,y_pred)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R-squared:",r2)
```

Output :

```
Mean Absolute Error: 3.0314541994814834
Mean Squared Error: 17.510151234400585
Root Mean Squared Error: 4.184513261348395
R-squared: 0.8012416266635316
```

4.2 Decision tree regression

Decision tree regression is a supervised machine learning technique used for predicting continuous values. It constructs a tree-like model by recursively splitting the dataset based on feature values. At each node, the algorithm selects the best feature to split the data, aiming to minimize the error (e.g., mean squared error). Pruning is applied to prevent overfitting. The final model predicts output values based on input features. In Python, Scikit-learn provides the DecisionTreeRegressor module for implementing decision tree regression.

Training the model

```
model = DecisionTreeRegressor(random_state=42)
model.fit(x_train, y_train)
```

Output :

```
In [96]: model = DecisionTreeRegressor(random_state=42)
         model.fit(x_train, y_train)

Out[96]: ▾ DecisionTreeRegressor
         DecisionTreeRegressor(random_state=42)
```

Making Prediction

```
y_pred = model.predict(x_test)
y_pred
```

Output :

```
In [97]: y_pred = model.predict(x_test)
         y_pred

Out[97]: array([78. , 74. , 56.2, 72.5, 78.7, 75.8, 58.3, 67.2, 66.6, 56.5, 81.5,
                59.3, 61.4, 75.9, 59.6, 64.8, 79.6, 88. , 63.1, 75.1, 75.4, 54. ,
                52.9, 62.1, 65.5, 59.3, 57.4, 81.7, 79.4, 53.1, 52.5, 69.8, 87. ,
                73.3, 79.7, 73.3, 72.4, 71.8, 79.3, 72.9, 69.6, 74.5, 71.5, 61.4,
```

Evaluating Model Performance:

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error: {mae}")
print(f"Mean Squared Error: {mse}")
```

```python
print("Root Mean Squared Error:", rmse)
print("R-squared:",r2)
```

Output :

```
Mean Absolute Error: 1.5609693877551019
Mean Squared Error: 7.896075680272109
Root Mean Squared Error: 2.8099956726429505
R-squared: 0.9103713533399269
```

4.3 Random forest Regression

Random forest regression** is a supervised machine learning algorithm that constructs an ensemble of decision trees during training. Each tree is built using a random subset of the dataset and features. By aggregating predictions from individual trees, random forest reduces over fitting and provides reliable forecasts for continuous target variables. In Python, RandomForestRegressor module is commonly used for implementing random forest regression.

Training the model

```
model = RandomForestRegressor(random_state=42)
model.fit(x_train, y_train)
```

Output :

```
In [81]: model = RandomForestRegressor(random_state=42)
         model.fit(x_train, y_train)

Out[81]: ▾ RandomForestRegressor
         RandomForestRegressor(random_state=42)
```

Making Prediction

```
y_pred = model.predict(x_test)
y_pred
```

Output :

```
In [100]: y_pred = model.predict(x_test)
          y_pred

Out[100]: array([73.237 , 73.632 , 55.695 , 73.157 , 77.482 , 75.896 , 56.618 ,
                 67.296 , 72.529 , 55.019 , 83.648 , 59.579 , 61.621 , 74.696 ,
                 59.444 , 65.053 , 80.061 , 83.527 , 64.06  , 74.941 , 75.454 ,
```

Evaluating Model Performance:

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)
print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R-squared:",r2)
```

Output :

```
Mean Absolute Error: 1.208349489795919
Mean Squared Error: 3.851453173894555
Root Mean Squared Error: 1.9625119550959569
R-squared: 0.9562820127834798
```

5 Conclusion to our project

In this project, I explored a dataset related to life expectancy.Then started by exploring the dataset, visualizing key features like age, income, and health indicators, and gaining insights. This step is crucial for understanding the data distribution and identifying patterns.The graphs we created helped visualize relationships between variables, such as age, income, and health indicators.We performed feature engineering to enhance model performance. This could involve creating new features, handling missing values.

After performing feature engineering, I built three models: linear regression, random forest, and decision tree. These models aimed to predict life expectancy based on input features.

To evaluate their performance, I used metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Moving forward, I recommend promoting healthier lifestyles, improving healthcare access, and addressing socioeconomic disparities to enhance life expectancy.