Name :Chanda dunani

Date : 1 /03/2022

Subject Name : : **Introduction to Data Science**

Github Link : https://github.com/chandadunani/ids7

# Data Cleaning

This was the very first step to be performed before using it. It is important as our model is based on it and we must identify any missing , irrelevant or null values.

Following steps were followed to clean the data :

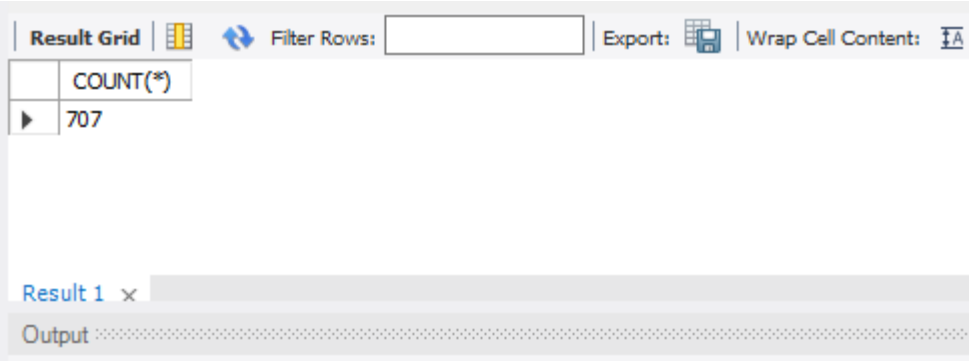
1. Data was loaded into MYSQL Workbench.
2. Rows with missing values were identified and deleted.
3. Rows with value corresponding to zero for **Life Expectancy**, **Adult Mortality**, **Percentage Expenditure**, **BMI**, **Total Expenditure**, **GDP,** and **Population** were removed using the following Query.

```
DELETE FROM life_expectancy WHERE
    `Year`= 0 OR
    Life_Expectancy = 0 OR
    Adult_Mortality = 0 OR
    Total_Expenditure = 0 OR
    GDP = 0 OR
    Population = 0;
```

# Importing the Data

Importing the data from SQL was done and was loaded into a data frame to perform further operations.

Data can be loaded form the local directory also.


Result Grid
Filter Rows:
Export:
Wrap Cell Content:
COUNT(*)
707
Result 1
Output

```sql
#List of countries with the highest and lowest average mortality rates.
SELECT * FROM(
    SELECT country,dataset.Year,dataset.Life_Expectancy,'Min' AS VAL  FROM dataset JOIN
    (SELECT year,min(Life_Expectancy) AS Life_Expectancy  FROM dataset GROUP BY year) AS tbl
    ON dataset.year =tbl.year AND tbl.Life_Expectancy=dataset.Life_Expectancy
UNION ALL
    SELECT country,dataset.Year,dataset.Life_Expectancy,'Max' AS VAL FROM dataset JOIN
    (SELECT year,Max(Life_Expectancy) AS Life_Expectancy  FROM dataset GROUP BY YEAR) AS tbl
    ON dataset.year =tbl.year and tbl.Life_Expectancy=dataset.Life_Expectancy) AS tblall ORDER BY Year;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| country | Year | Life_Expectancy | VAL |
|---|---|---|---|
| Haiti | 2010 | 36.3 | Min |
| Netherlands | 2010 | 88 | Max |
| Sierra Leone | 2011 | 48.9 | Min |
| Austria | 2011 | 88 | Max |
| Luxembourg | 2011 | 88 | Max |
| Sierra Leone | 2012 | 49.7 | Min |
| Austria | 2012 | 88 | Max |
| Central African Republic | 2013 | 49.9 | Min |
| Belgium | 2013 | 87 | Max |
| Finland | 2013 | 87 | Max |
| Sierra Leone | 2014 | 48.1 | Min |
| Belgium | 2014 | 89 | Max |

Result 2

```sql
#List of countries with the highest and lowest average GDP (years 2010-2015)
SELECT * FROM(
    SELECT country,dataset.Year,dataset.GDP,'Min' AS VAL  FROM dataset JOIN
    (SELECT year,min(GDP) AS GDP  FROM dataset GROUP BY year) AS tbl
    ON dataset.year =tbl.year AND tbl.GDP=dataset.GDP
UNION ALL
    SELECT country,dataset.Year,dataset.GDP,'Max' AS VAL FROM dataset JOIN
    (SELECT year,Max(GDP) AS GDP  FROM dataset GROUP BY YEAR) AS tbl
    ON dataset.year =tbl.year and tbl.GDP=dataset.GDP) AS tblall ORDER BY Year;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| country | Year | GDP | VAL |
|---|---|---|---|
| Mauritius | 2010 | 8.376432 | Min |
| Norway | 2010 | 87646.75346 | Max |
| Senegal | 2011 | 18.25321 | Min |
| Luxembourg | 2011 | 115761.577 | Max |
| Guinea | 2012 | 52.3485646 | Min |
| Switzerland | 2012 | 83164.38795 | Max |
| Tajikistan | 2013 | 14.214412 | Min |
| Luxembourg | 2013 | 113751.85 | Max |
| Romania | 2014 | 12.27733 | Min |
| Luxembourg | 2014 | 119172.7418 | Max |
| Afghanistan | 2015 | 584.25921 | Min |
| Albania | 2015 | 3954.22783 | Max |

Result 4

```sql
#Which countries have the highest and lowest average Alcohol consumption (years 2010-2015)
SELECT * FROM(
    SELECT country,dataset.Year,dataset.Alcohol,'Min' AS VAL  FROM dataset JOIN
    (SELECT year,min(Alcohol) AS Alcohol  FROM dataset GROUP BY year) AS tbl
    ON dataset.year =tbl.year AND tbl.Alcohol=dataset.Alcohol
UNION ALL
    SELECT country,dataset.Year,dataset.Alcohol,'Max' AS VAL FROM dataset JOIN
    (SELECT year,Max(Alcohol) AS Alcohol  FROM dataset GROUP BY YEAR) AS tbl
    ON dataset.year =tbl.year and tbl.Alcohol=dataset.Alcohol) AS tblall ORDER BY Year;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| country | Year | Alcohol | VAL |
|---|---|---|---|
| Afghanistan | 2010 | 0.01 | Min |
| Bangladesh | 2010 | 0.01 | Min |
| Mauritania | 2010 | 0.01 | Min |
| Estonia | 2010 | 14.97 | Max |
| Afghanistan | 2011 | 0.01 | Min |
| Bangladesh | 2011 | 0.01 | Min |
| Estonia | 2011 | 0.01 | Min |
| Fiji | 2011 | 0.01 | Min |
| Mauritania | 2011 | 0.01 | Min |
| Mongolia | 2011 | 0.01 | Min |
| Belarus | 2011 | 17.31 | Max |
| South Sudan | 2012 | 0 | Min |
| Belarus | 2012 | 16.35 | Max |
| South Sudan | 2013 | 0 | Min |

Result 6

```sql
#List of countries with the highest and lowest average Schooling  (years 2010-2015)
SELECT * FROM(
    SELECT country,dataset.Year,dataset.Schooling,'Min' AS VAL  FROM dataset JOIN
    (SELECT year,min(Schooling) AS Schooling  FROM dataset GROUP BY year) AS tbl
    ON dataset.year =tbl.year AND tbl.Schooling=dataset.Schooling
UNION ALL
    SELECT country,dataset.Year,dataset.Schooling,'Max' AS VAL FROM dataset JOIN
    (SELECT year,Max(Schooling) AS Schooling  FROM dataset GROUP BY YEAR) AS tbl
    ON dataset.year =tbl.year and tbl.Schooling=dataset.Schooling) AS tblall ORDER BY Year;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| country | Year | Schooling | VAL |
|---|---|---|---|
| Niger | 2010 | 4.5 | Min |
| Australia | 2010 | 19.5 | Max |
| Niger | 2011 | 4.8 | Min |
| Australia | 2011 | 19.8 | Max |
| South Sudan | 2012 | 4.9 | Min |
| Australia | 2012 | 20.1 | Max |
| South Sudan | 2013 | 4.9 | Min |
| Australia | 2013 | 20.3 | Max |
| South Sudan | 2014 | 4.9 | Min |
| Australia | 2014 | 20.4 | Max |
| Afghanistan | 2015 | 10.1 | Min |
| Albania | 2015 | 14.2 | Max |

Result 5

```
63    #Do densely populated countries tend to have lower life expectancy?
64 •  SELECT Country, Population , Life_Expectancy FROM dataset GROUP BY Country ORDER BY Population DESC;
65    #not much relation is seen b/w the two variables.
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| Country | Population | Life_Expectancy |
|---|---|---|
| Indonesia | 242524123 | 68.1 |
| Brazil | 196796269 | 73.8 |
| Nigeria | 158578261 | 52 |
| Russian Fe... | 142849449 | 68.4 |
| Mexico | 117318941 | 75.6 |
| Philippines | 93726624 | 67.9 |
| Turkey | 72326914 | 74.2 |
| Italy | 59277417 | 81.8 |
| Spain | 46576897 | 81.9 |
| Argentina | 41223889 | 75.5 |
| Algeria | 36117637 | 74.7 |
| Sudan | 34385963 | 62.5 |
| Uganda | 33915133 | 58.4 |
| Peru | 29373646 | 73.7 |

dataset 7 ×

In [ ]:
```
pip install mysql-connector-python
```

In [53]:
```
import pandas as pd import
mysql.connector as sql import
seaborn as sns import numpy
as np
import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression from
sklearn.model_selection import train_test_split from
sklearn.metrics import r2_score,mean_squared_error
```

In [7]:
```
db_connection = sql.connect(host='127.0.0.0', database='db', user='root', password='passwo
db_cursor = db_connection.cursor()
db_cursor.execute('SELECT * FROM dataset')

table_rows = db_cursor.fetchall() file
= pd.DataFrame(table_rows) '''
```

## Reading the file

```
data = file.copy()
data.info() data.head()
```

# Copying the file to prevent accidental changes.

64]:

```
<class 'pandas.core.frame.DataFrame'> RangeIndex:
707 entries, 0 to 706
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
0   Country                 707 non-null    object
1   Year                    707 non-null    int64
2   Life_Expectancy         707 non-null    float64
3   Adult_Mortality         707 non-null    int64
4   Alcohol                 707 non-null    float64
5   Percentage_Expenditure  707 non-null    float64
6   BMI                     707 non-null    float64
7   Total_Expenditure       707 non-null    float64
8   GDP                     707 non-null    float64
9   Population              707 non-null    int64    10  Schooling               707 non-null
    float64 dtypes: float64(7), int64(3), object(1) memory usage: 60.9+ KB
```

ut[64]:

| | Country | Year | Life_Expectancy | Adult_Mortality | Alcohol | Percentage_Expenditure | BMI | Total_Expenditure | |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Afghanistan2010 | | 58.8 | 279 | 0.01 | 79.679367 | 16.7 | 9.20 | 553 |
| **1** | Afghanistan2011 | | 59.2 | 275 | 0.01 | 7.097109 | 17.2 | 7.87 | 63 |
| **2** | Afghanistan2012 | | 59.5 | 272 | 0.01 | 78.184215 | 17.6 | 8.52 | 669 |
| **3** | Afghanistan2013 | | 59.9 | 268 | 0.01 | 73.219243 | 18.1 | 8.13 | 631 |
| **4** | Afghanistan2014 | | 59.9 | 271 | 0.01 | 73.523582 | 18.6 | 8.18 | 612 |

# Plotting the Corelation Matrix to get better insights.

# Based on our observation on the Corelation Matrix obtained we will choose various variables for our model.

```
data.corr()
#Plotting the Corelation Matrix to get better insights.
#Based on our observation on the Corelation Matrix obtained we
```

In [16]:

Out[16]:

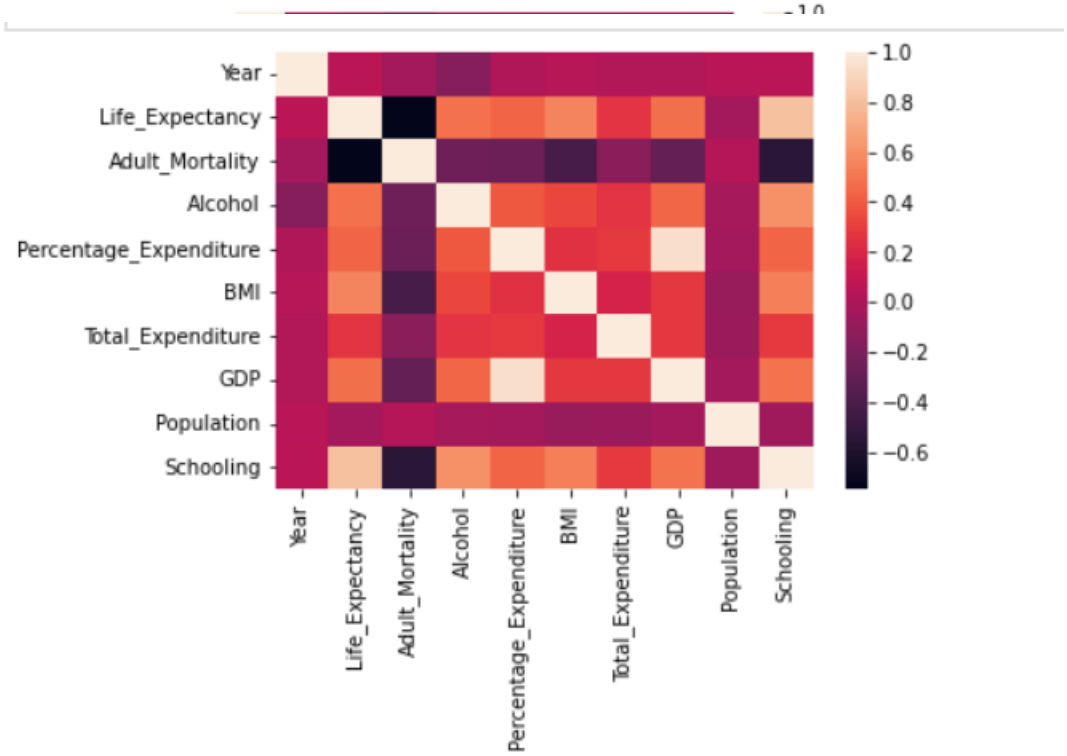
| | Year | Life_Expectancy | Adult_Mortality | AlcoholPercentage_Expenditure | BMI | To |
|---|---|---|---|---|---|---|
| **Year** | 1.000000 | 0.055936 | -0.035259 | -0.160970 | 0.013894 | 0.036295 |
| **Life_Expectancy** | 0.055936 | 1.000000 | -0.751148 | 0.478888 | 0.427136 | 0.548947 |
| **Adult_Mortality** | -0.035259 | -0.751148 | 1.000000 | -0.253955 | -0.270039 | -0.416356 |
| **Alcohol** | -0.160970 | 0.478888 | -0.253955 | 1.000000 | 0.387217 | 0.324022 |
| **Percentage_Expenditure** | 0.013894 | 0.427136 | -0.270039 | 0.387217 | 1.000000 | 0.242853 |
| **BMI** | 0.036295 | 0.548947 | -0.416356 | 0.324022 | 0.242853 | 1.000000 |
| **Total_Expenditure** | 0.018778 | 0.257310 | -0.148852 | 0.257711 | 0.277196 | 0.177937 |
| **GDP** | 0.020748 | 0.471575 | -0.298142 | 0.436485 | 0.940297 | 0.273065 |
| **Population** | 0.048307 | -0.034404 | 0.024392 | -0.032376 | -0.033992 | -0.083094 |
| **Schooling** | 0.055423 | 0.801730 | -0.558152 | 0.599283 | 0.425707 | 0.534159 |

In [19]:

```
#Plotting the heatmap to have a visual representatin of our Coorelation Matrix
corr = data.corr() sns.heatmap(corr,
        xticklabels=corr.columns,
yticklabels=corr.columns)
```

Out[19]: <AxesSubplot:>

```
#Created a Linear Model from sklearn libary

lin_reg_model = LinearRegression()
y = data['Life_Expectancy'].values.reshape(-1,1)
```




# HEATMAP

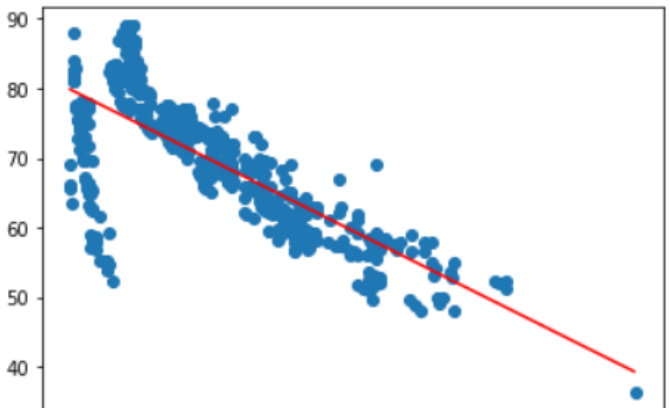
In [78]:

```
#X = Adult Mortality
x = data.Adult_Mortality.values.reshape(-1,1) lin_reg_model.fit(x,y)

#Predicted Line info
x_array = np.arange(min(data.Adult_Mortality),max(data.Adult_Mortality)).reshape(-1,1)
plt.scatter(x,y)
y_head = lin_reg_model.predict(x_array)
plt.plot(x_array,y_head,color="red") plt.show()

#Printing the various metrics
print("Mean Squared Error: ", mean_squared_error(x_array,y_head))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(x_array, y_head)))
print("R2 Score " ,r2_score(y, lin_reg_model.predict(x)))
print("Model Equation : y =",lin_reg_model.coef_[0][0],"x +",*lin_reg_model.intercept_)
print("Where Slope =",lin_reg_model.coef_[0][0], "\nIntercept =",*lin_reg_model.intercept_
```

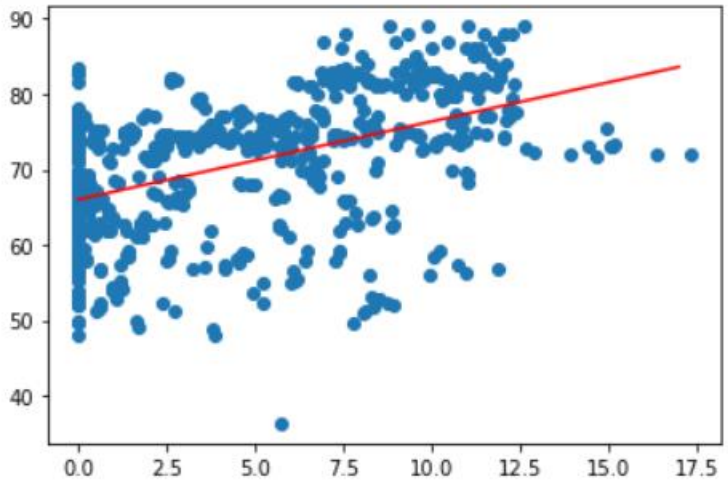
```
Mean Squared Error:  122764.170538361
Root Mean Squared Error:  350.3771832445158
R2 Score  0.5642234434438707
Model Equation : y = -0.059759966142484564 x + 79.97218981181695
Where Slope = -0.059759966142484564
Intercept = 79.97218981181695
```

In [79]:

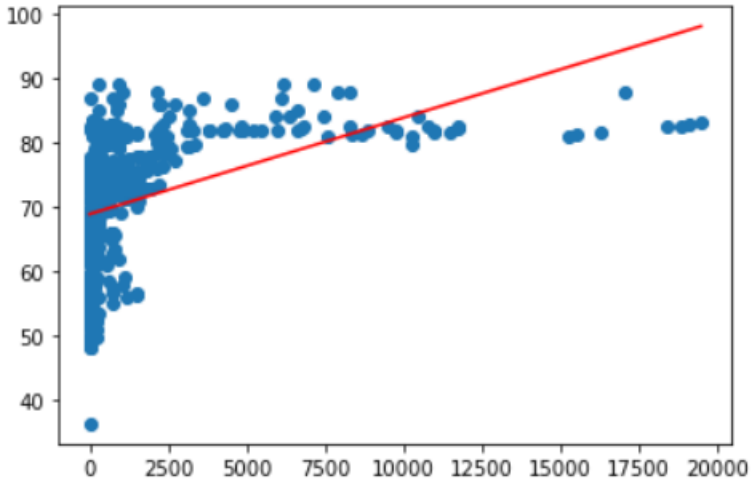
```
#X = Alcohol
x = data.Alcohol.values.reshape(-1,1) lin_reg_model.fit(x,y)

#Predicted Line info
x_array = np.arange(min(data.Alcohol),max(data.Alcohol)).reshape(-1,1)    plt.scatter(x,y)
y_head = lin_reg_model.predict(x_array)
plt.plot(x_array,y_head,color="red") plt.show()


#Printing the various metrics
print("Mean Squared Error: ", mean_squared_error(x_array,y_head))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(x_array, y_head)))
print("R2 Score " ,r2_score(y, lin_reg_model.predict(x)))
print("Model Equation : y =",lin_reg_model.coef_[0][0],"x +",*lin_reg_model.intercept_)
print("Where Slope =",lin_reg_model.coef_[0][0], "\nIntercept =",*lin_reg_model.intercept_
```



```
#X = Percentage_Expenditure
x = data.Percentage_Expenditure.values.reshape(-1,1) lin_reg_model.fit(x,y)

#Predicted Line info
x_array = np.arange(min(data.Percentage_Expenditure),max(data.Percentage_Expenditure)).res
plt.scatter(x,y)
y_head = lin_reg_model.predict(x_array)
plt.plot(x_array,y_head,color="red") plt.show()


#Printing the various metrics
print("Mean Squared Error: ", mean_squared_error(x_array,y_head))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(x_array, y_head)))
print("R2 Score " ,r2_score(y, lin_reg_model.predict(x)))
print("Model Equation : y =",lin_reg_model.coef_[0][0],"x +",*lin_reg_model.intercept_)
print("Where Slope =",lin_reg_model.coef_[0][0], "\nIntercept =",*lin_reg_model.intercept_
```




```
Mean Squared Error:  4397.267282761441
Root Mean Squared Error:  66.31189397658191
R2 Score  0.22933384569354576
Model Equation : y = 1.035864310383005 x + 66.00678628713952
Where Slope = 1.035864310383005
Intercept = 66.00678628713952
```

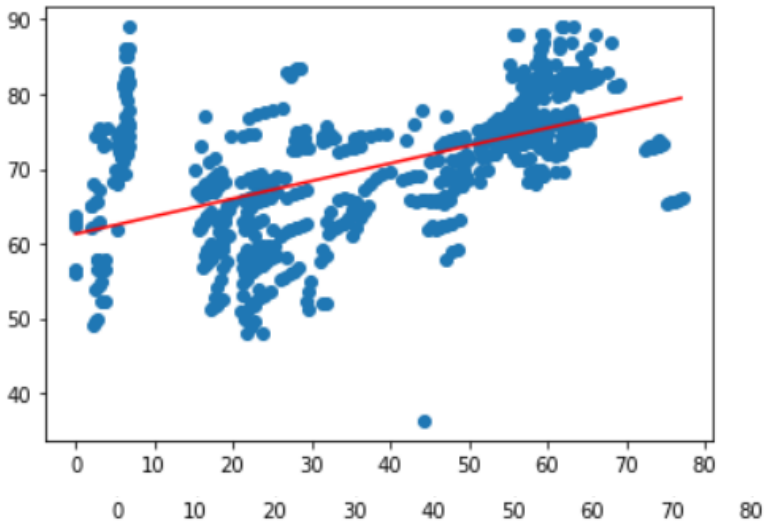


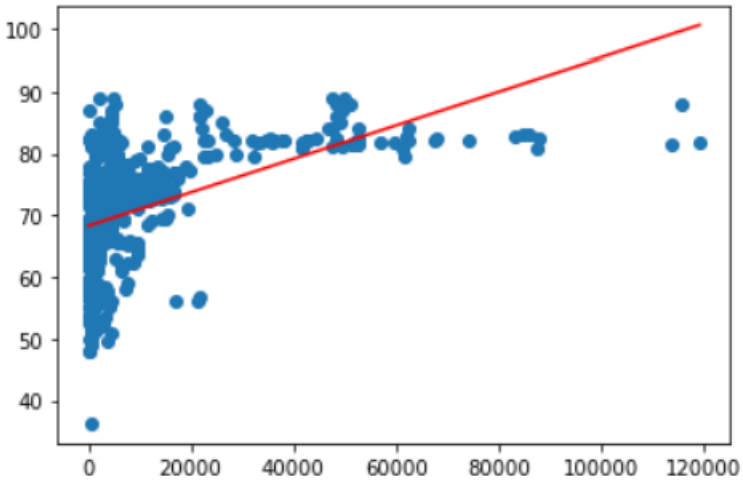

```
Mean Squared Error:  124767590.63668308
Root Mean Squared Error:  11169.941389133744
R2 Score  0.18244520340149972
Model Equation : y = 0.0015007079679564452 x + 68.91083837385924
Where Slope = 0.0015007079679564452
Intercept = 68.91083837385924
```

```
#X = BMI
x = data.BMI.values.reshape(-1,1) lin_reg_model.fit(x,y)

#Predicted Line info
x_array = np.arange(min(data.BMI ),max(data.BMI )).reshape(-1,1)
plt.scatter(x,y)
y_head = lin_reg_model.predict(x_array)
plt.plot(x_array,y_head,color="red") plt.show()


#Printing the various metrics
print("Mean Squared Error: ", mean_squared_error(x_array,y_head))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(x_array, y_head)))
print("R2 Score " ,r2_score(y, lin_reg_model.predict(x)))
print("Model Equation : y =",lin_reg_model.coef_[0][0],"x +",*lin_reg_model.intercept_)
print("Where Slope =",lin_reg_model.coef_[0][0], "\nIntercept =",*lin_reg_model.intercept_
```

```
Mean Squared Error:  1311.7473230203739
Root Mean Squared Error:  36.21805244654072
R2 Score  0.301343087099337
Model Equation : y = 0.23624568858648645 x + 61.28011728908278
Where Slope = 0.23624568858648645  Intercept
= 61.28011728908278
```

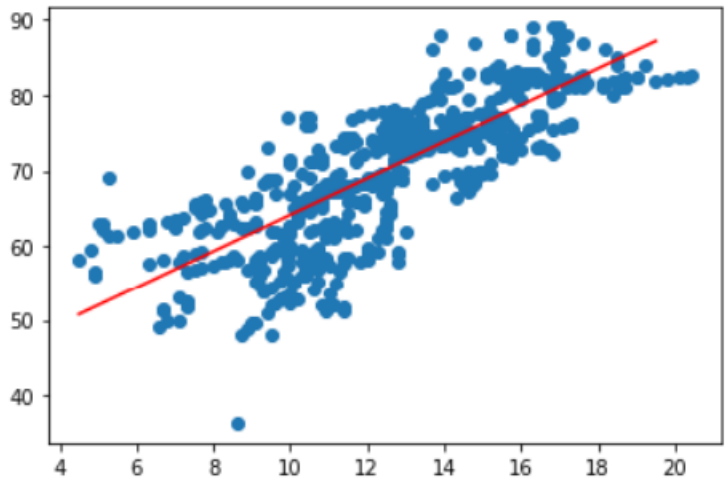
```
#X = GDP
x = data.GDP.values.reshape(-1,1) lin_reg_model.fit(x,y)

#Predicted Line info
x_array = np.arange(min(data.GDP  ),max(data.GDP  )).reshape(-1,1)
plt.scatter(x,y)
y_head = lin_reg_model.predict(x_array)
plt.plot(x_array,y_head,color="red") plt.show()


#Printing the various metrics
print("Mean Squared Error: ", mean_squared_error(x_array,y_head))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(x_array, y_head)))
print("R2 Score " ,r2_score(y, lin_reg_model.predict(x)))
print("Model Equation : y =",lin_reg_model.coef_[0][0],"x +",*lin_reg_model.intercept_)
print("Where Slope =",lin_reg_model.coef_[0][0], "\nIntercept =",*lin_reg_model.intercept_
```




```
Mean Squared Error:  4723670841.625562
Root Mean Squared Error:  68728.96653977537
R2 Score  0.222382925948318
Model Equation : y = 0.00027124291132882343 x + 68.29218749386874
Where Slope = 0.00027124291132882343
Intercept = 68.29218749386874
```

```
#X = Schooling
x = data.Schooling.values.reshape(-1,1)
lin_reg_model.fit(x,y)

#Predicted Line info
x_array = np.arange(min(data.Schooling),max(data.Schooling)).reshape(-1,1)
plt.scatter(x,y)
y_head = lin_reg_model.predict(x_array)
plt.plot(x_array,y_head,color="red") plt.show()


#Printing the various metrics
print("Mean Squared Error: ", mean_squared_error(x_array,y_head))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(x_array, y_head)))
print("R2 Score " ,r2_score(y, lin_reg_model.predict(x)))
print("Model Equation : y =",lin_reg_model.coef_[0][0],"x +",*lin_reg_model.intercept_)
print("Where Slope =",lin_reg_model.coef_[0][0], "\nIntercept =",*lin_reg_model.intercept_
```




```
Mean Squared Error:  3290.175489513466
Root Mean Squared Error:  57.36005133813485
R2 Score  0.6427713989793805
Model Equation : y = 2.428981465040257 x + 39.832773872890975
Where Slope = 2.428981465040257  Intercept
= 39.832773872890975
```

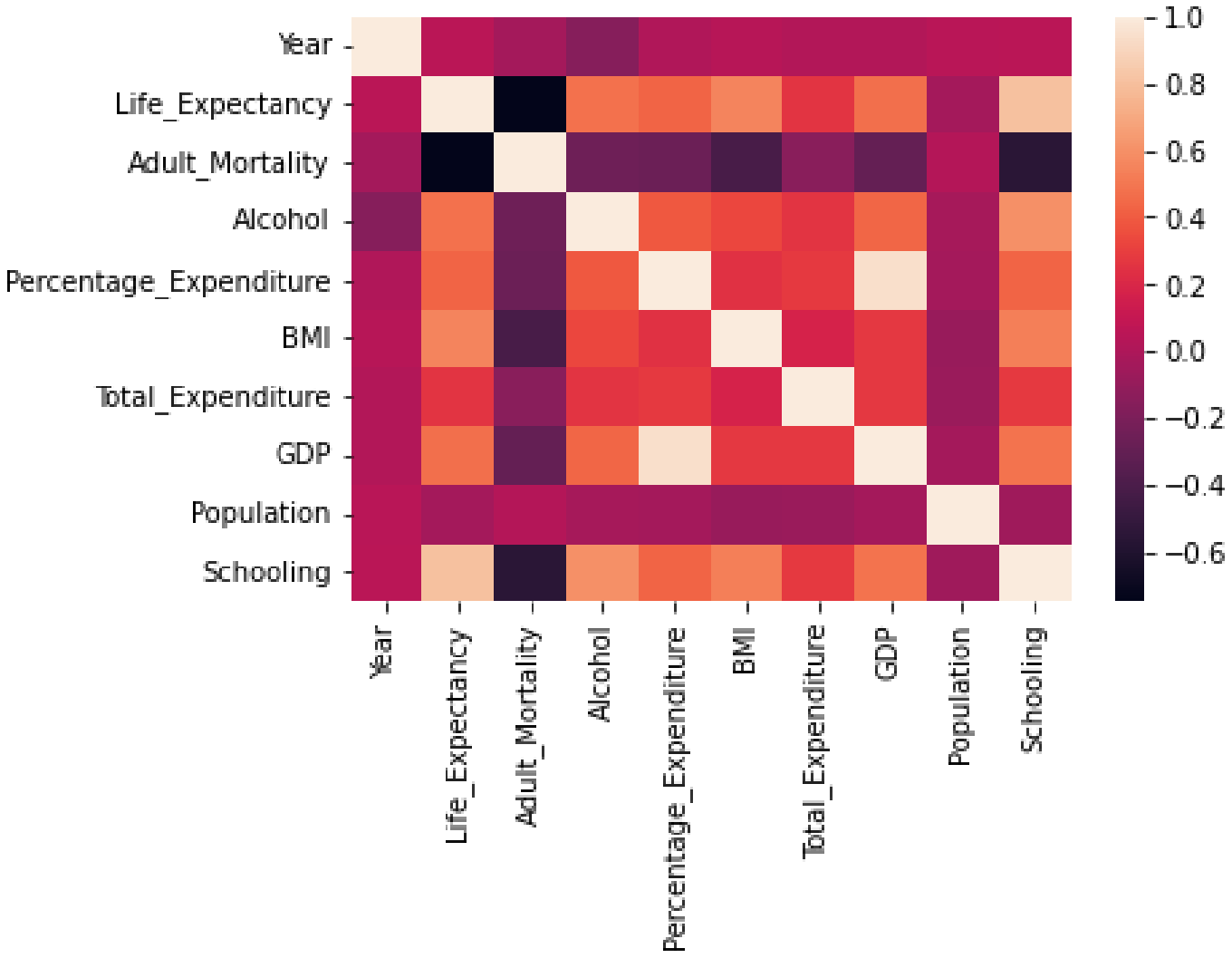
# Plotting the Correlation Matrix

Correlation matrix is a table which depicts the correlation coefficients b/w different variables.

This is helpful to give better insights of the data and to help which factors affect our desired variable the most. It is an important step influencing our variable selection for our model.

| | Year | Life_Expectancy | Adult_Mortality | Alcohol | Percentage_Expenditure | BMI | Total_Expenditure | GDP | Population | Schooling |
|---|---|---|---|---|---|---|---|---|---|---|
| **Year** | 1.000000 | 0.055936 | -0.035259 | -0.160970 | 0.013894 | 0.036295 | 0.018778 | 0.020748 | 0.048307 | 0.055423 |
| **Life_Expectancy** | 0.055936 | 1.000000 | -0.751148 | 0.478888 | 0.427136 | 0.548947 | 0.257310 | 0.471575 | -0.034404 | 0.801730 |
| **Adult_Mortality** | -0.035259 | -0.751148 | 1.000000 | -0.253955 | -0.270039 | -0.416356 | -0.148852 | -0.298142 | 0.024392 | -0.558152 |
| **Alcohol** | -0.160970 | 0.478888 | -0.253955 | 1.000000 | 0.387217 | 0.324022 | 0.257711 | 0.436485 | -0.032376 | 0.599283 |
| **Percentage_Expenditure** | 0.013894 | 0.427136 | -0.270039 | 0.387217 | 1.000000 | 0.242853 | 0.277196 | 0.940297 | -0.033992 | 0.425707 |
| **BMI** | 0.036295 | 0.548947 | -0.416356 | 0.324022 | 0.242853 | 1.000000 | 0.177937 | 0.273065 | -0.083094 | 0.534159 |
| **Total_Expenditure** | 0.018778 | 0.257310 | -0.148852 | 0.257711 | 0.277196 | 0.177937 | 1.000000 | 0.272106 | -0.077060 | 0.280332 |
| **GDP** | 0.020748 | 0.471575 | -0.298142 | 0.436485 | 0.940297 | 0.273065 | 0.272106 | 1.000000 | -0.036691 | 0.484713 |
| **Population** | 0.048307 | -0.034404 | 0.024392 | -0.032376 | -0.033992 | -0.083094 | -0.077060 | -0.036691 | 1.000000 | -0.057181 |
| **Schooling** | 0.055423 | 0.801730 | -0.558152 | 0.599283 | 0.425707 | 0.534159 | 0.280332 | 0.484713 | -0.057181 | 1.000000 |

# Heat Map




# Observations

- Does various predicting factors which has been chosen initially really affect the Life expectancy? What are the predicting variables actually affecting the life expectancy?

| | |
|---|---|
| **Schooling** | 0.801730 |
| **Adult Mortality** | -0.751148 |
| **BMI** | 0.548947 |
| Alcohol | 0.478888 |
| Percentage Expenditure | 0.427136 |
| Total Expenditure | 0.257310 |
| GDP | 0.471575 |

- Should a country having a lower life expectancy value(<65) increase its healthcare expenditure in order to improve its average lifespan?

- How does Adult mortality rates affect life expectancy?
  **It inversely affects the life expectancy.**
- Does Life Expectancy have positive or negative correlation with eating habits, social factors, drinking alcohol, etc.?
  **Positive Corelation can be seen with Alcohol and BMI ( Eating Habits )**

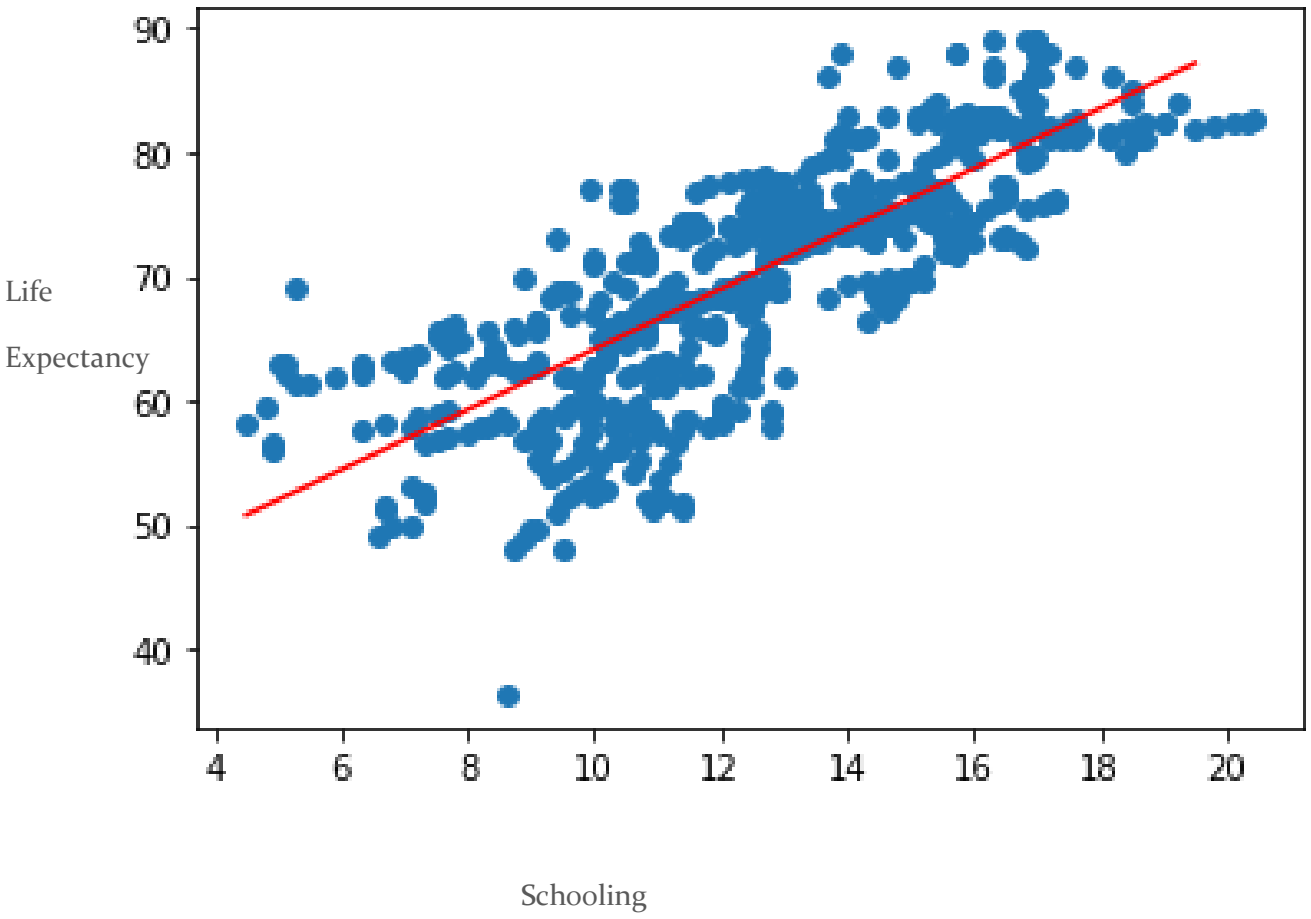
- What is the impact of schooling on the lifespan of humans?
  **It has strong corelation with the the lifespan of human , higher the Schooling more is the lifespan of humans.**

- Does Life Expectancy have positive or negative relationship with drinking alcohol?
  **It has significant positive relationship with Life Expectancy.**

- Do densely populated countries tend to have lower life expectancy?
  **No it is unlikely as per the corelation coefficient.**

# Model

As per our finding our model works best for Schooling as our independent variable.




The following are the metrics :

```
Mean Squared Error:  3290.175489513466
Root Mean Squared Error:  57.36005133813485
R2 Score  0.6427713989793805
Model Equation : y = 2.428981465040257 x + 39.832773872890975
Where Slope = 2.428981465040257
Intercept = 39.832773872890975
```

# Conclusion

Based on our data analysis we can conclude that Schooling is the most important factor which is quite evident if we see, as schooling provides people with better opportunity and promises better life.

```
Mean Squared Error:  3290.175489513466
Root Mean Squared Error:  57.36005133813485
R2 Score  0.6427713989793805
Model Equation : y = 2.428981465040257 x + 39.832773872890975
Where Slope = 2.428981465040257
Intercept = 39.832773872890975
```