SC1015 Project: WHO Life Expectancy

SC11 - Group 1
Jeremy Ong
Qian Cheng
Rhys Wong

TABLE OF CONTENTS

01. Practical Motivations

02. Exploratory analysis and data preparation

03. Models and Machine Learning

04. Conclusion

01 .Practical Motivations

-Life expectancy: *is a statistical measure the average period that a person may expect to live*

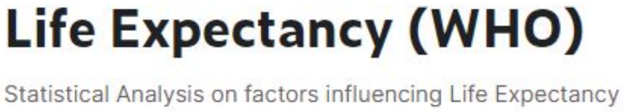
-Key metric for assessing population health.

Goal: Seek to improve the overall population health of a country

Dataset used:

kaggle

World Health Organization



Life Expectancy (WHO)

Statistical Analysis on factors influencing Life Expectancy

Our Problem Statement

Out of all of the predictors, how do they affect life expectancy?

Find out the more significant factors in affecting life expectancy.

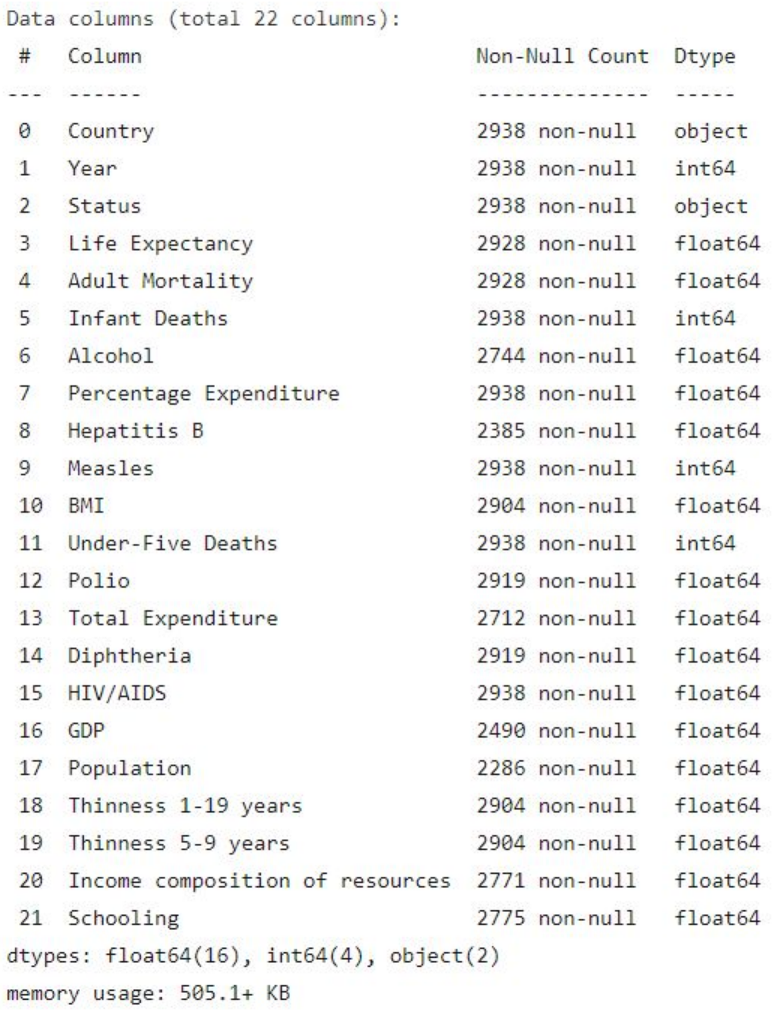
01. Practical Motivations ✓

02. Exploratory analysis and data preparation

03. Models and Machine Learning

04. Conclusion

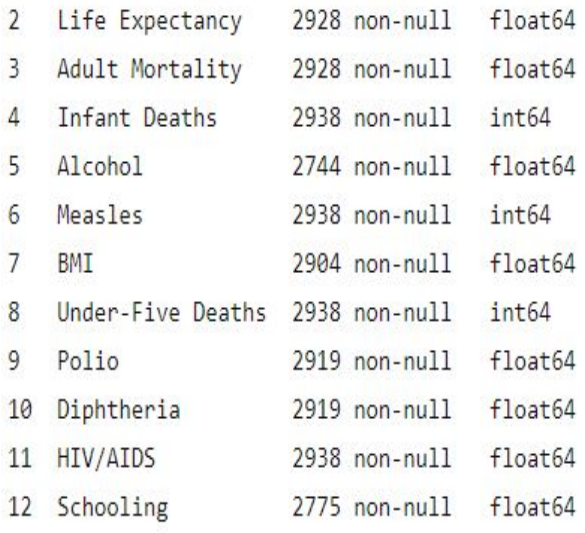
02 Exploratory Analysis and Data Preparation



```
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                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   Infant Deaths                    2938 non-null   int64
 6   Alcohol                          2744 non-null   float64
 7   Percentage Expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64
 10  BMI                              2904 non-null   float64
 11  Under-Five Deaths                2938 non-null   int64
 12  Polio                            2919 non-null   float64
 13  Total Expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15  HIV/AIDS                         2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18  Thinness 1-19 years              2904 non-null   float64
 19  Thinness 5-9 years               2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

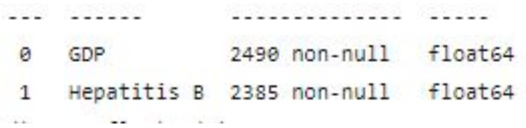
Data exploration using basic statistical exploration

To use



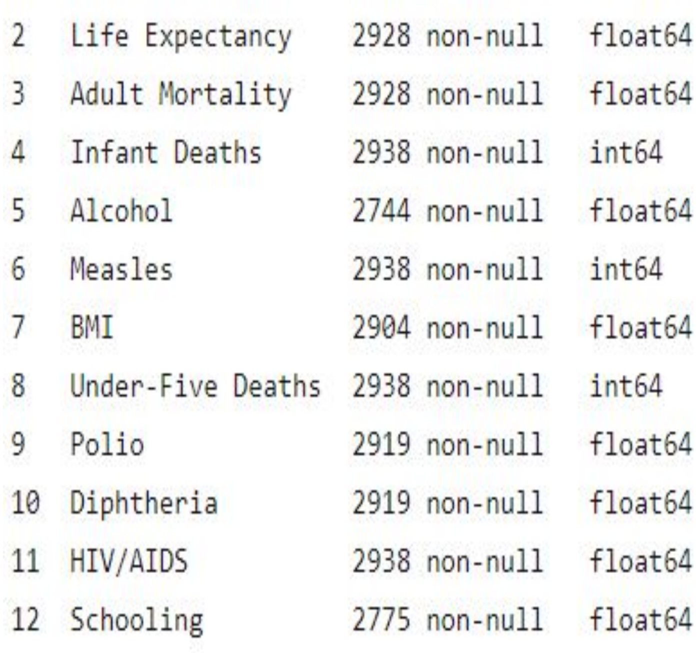
```
2   Life Expectancy   2928 non-null   float64
3   Adult Mortality   2928 non-null   float64
4   Infant Deaths     2938 non-null   int64
5   Alcohol           2744 non-null   float64
6   Measles           2938 non-null   int64
7   BMI               2904 non-null   float64
8   Under-Five Deaths 2938 non-null   int64
9   Polio             2919 non-null   float64
10  Diphtheria        2919 non-null   float64
11  HIV/AIDS          2938 non-null   float64
12  Schooling         2775 non-null   float64
```

To remove



```
---  ------       --------------  -----
 0   GDP          2490 non-null   float64
 1   Hepatitis B  2385 non-null   float64
```

Final Predictors used for future analysis

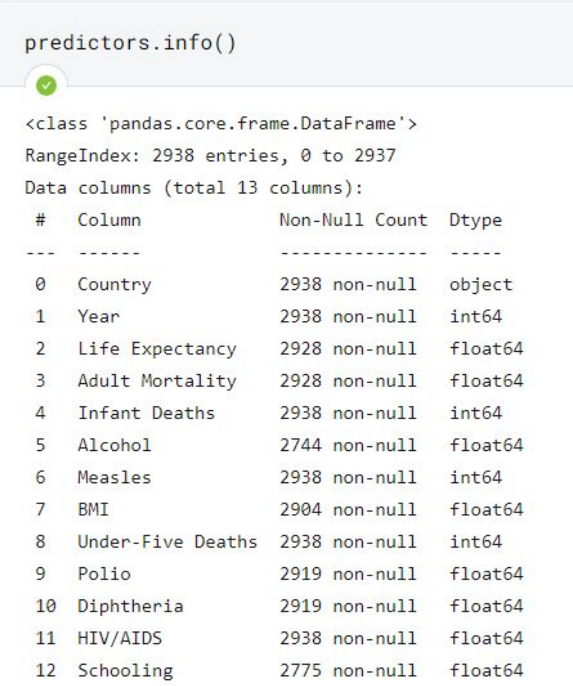


```
2   Life Expectancy    2928 non-null   float64
3   Adult Mortality    2928 non-null   float64
4   Infant Deaths      2938 non-null   int64
5   Alcohol            2744 non-null   float64
6   Measles            2938 non-null   int64
7   BMI                2904 non-null   float64
8   Under-Five Deaths  2938 non-null   int64
9   Polio              2919 non-null   float64
10  Diphtheria         2919 non-null   float64
11  HIV/AIDS           2938 non-null   float64
12  Schooling          2775 non-null   float64
```

Predictors

+ Adult mortality: (No of deaths per 1000)
+ Infant death: (No of deaths per 1000)
+ Alcohol : (total per capita (15+ years) consumption (in litres of pure alcohol))
+ Measles: (Total No of Cases)
+ BMI :(Average of entire Pop)
+ Under-five deaths (Number per 1000)
+ Polio (Immunization % of 1-year-olds)
+Diphtheria (Immunization % of 1-year-olds)
+ HIV / AIDS (Deaths per 1000)
+ Schooling (Average number of years studied)

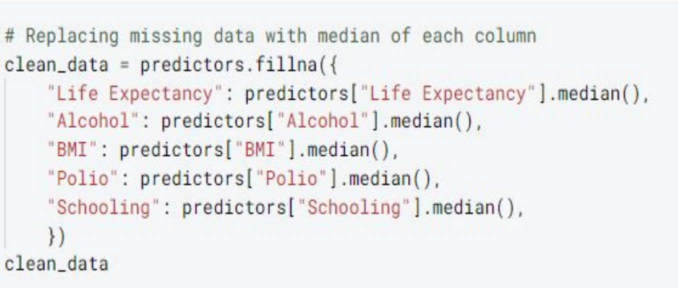
Missing Data



```
predictors.info()
```

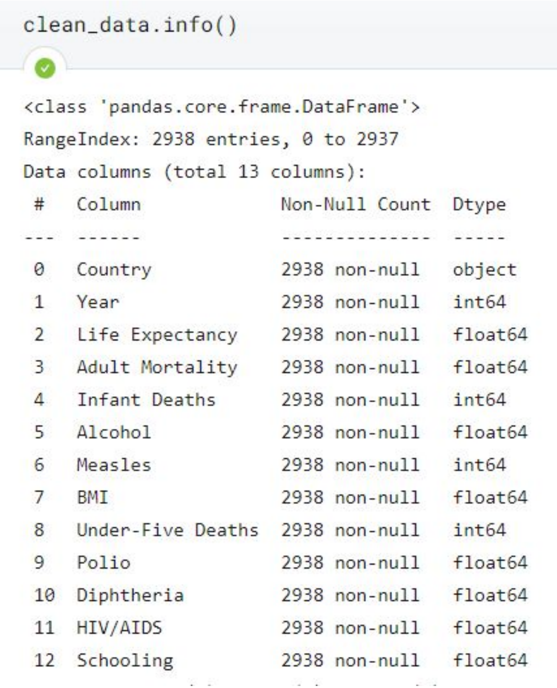
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Country            2938 non-null   object 
 1   Year               2938 non-null   int64  
 2   Life Expectancy    2928 non-null   float64
 3   Adult Mortality    2928 non-null   float64
 4   Infant Deaths      2938 non-null   int64  
 5   Alcohol            2744 non-null   float64
 6   Measles            2938 non-null   int64  
 7   BMI                2904 non-null   float64
 8   Under-Five Deaths  2938 non-null   int64  
 9   Polio              2919 non-null   float64
 10  Diphtheria         2919 non-null   float64
 11  HIV/AIDS           2938 non-null   float64
 12  Schooling          2775 non-null   float64
```

Before Filling



```
# Replacing missing data with median of each column
clean_data = predictors.fillna({
    "Life Expectancy": predictors["Life Expectancy"].median(),
    "Alcohol": predictors["Alcohol"].median(),
    "BMI": predictors["BMI"].median(),
    "Polio": predictors["Polio"].median(),
    "Schooling": predictors["Schooling"].median(),
    })
clean_data
```

Filling in

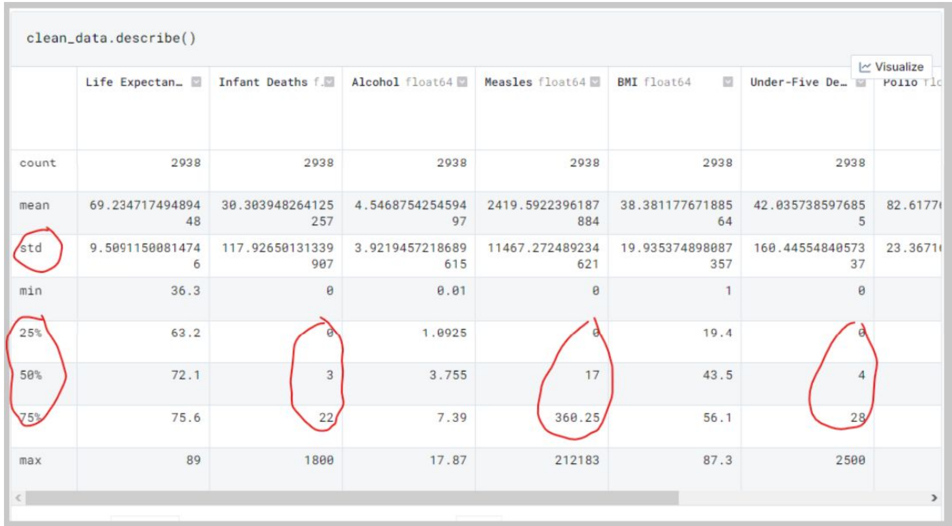


```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Country            2938 non-null   object 
 1   Year               2938 non-null   int64  
 2   Life Expectancy    2938 non-null   float64
 3   Adult Mortality    2938 non-null   float64
 4   Infant Deaths      2938 non-null   int64  
 5   Alcohol            2938 non-null   float64
 6   Measles            2938 non-null   int64  
 7   BMI                2938 non-null   float64
 8   Under-Five Deaths  2938 non-null   int64  
 9   Polio              2938 non-null   float64
 10  Diphtheria         2938 non-null   float64
 11  HIV/AIDS           2938 non-null   float64
 12  Schooling          2938 non-null   float64
```

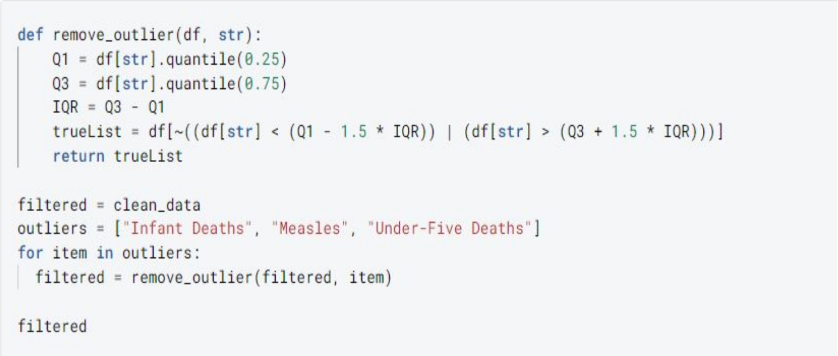
After Filling

Dealing with Outliers



```
clean_data.describe()
```

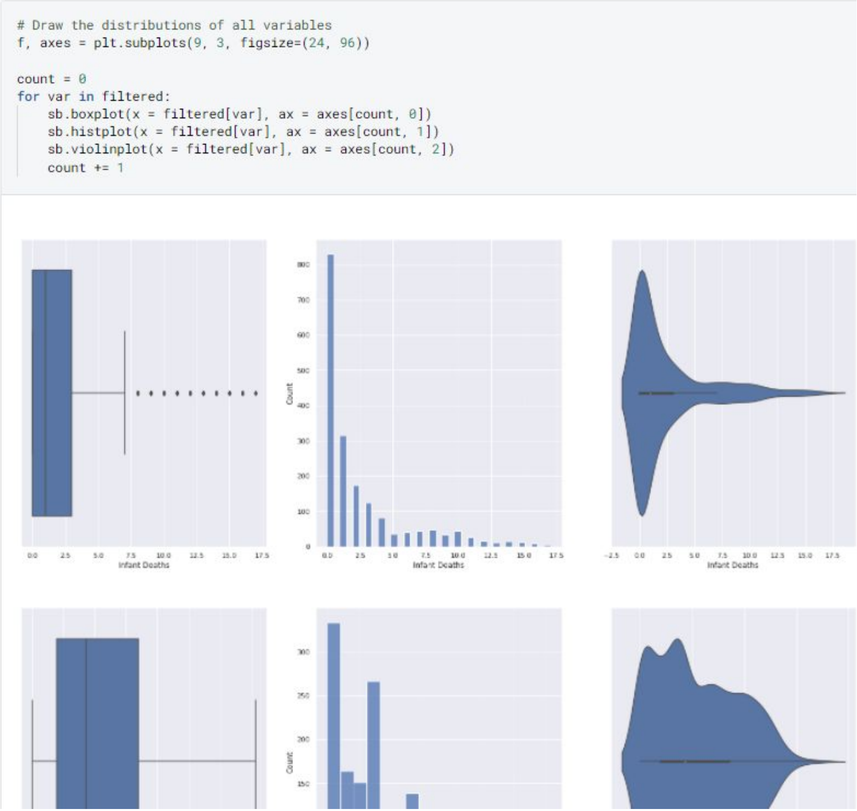
	Life Expectan…	Infant Deaths f…	Alcohol float64	Measles float64	BMI float64	Under-Five De…	Polio fl
count	2938	2938	2938	2938	2938	2938	
mean	69.23471749489448	30.303948264125257	4.546875425459497	2419.5922396187884	38.38117767188564	42.0357385976855	82.6177
std	9.50911500814746	117.92650131339907	3.9219457218689615	11467.272489234621	19.935374898087357	160.4455484057337	23.3671
min	36.3	0	0.01	0	1	0	
25%	63.2	0	1.0925	0	19.4	0	
50%	72.1	3	3.755	17	43.5	4	
75%	75.6	22	7.39	360.25	56.1	28	
max	89	1800	17.87	212183	87.3	2500	



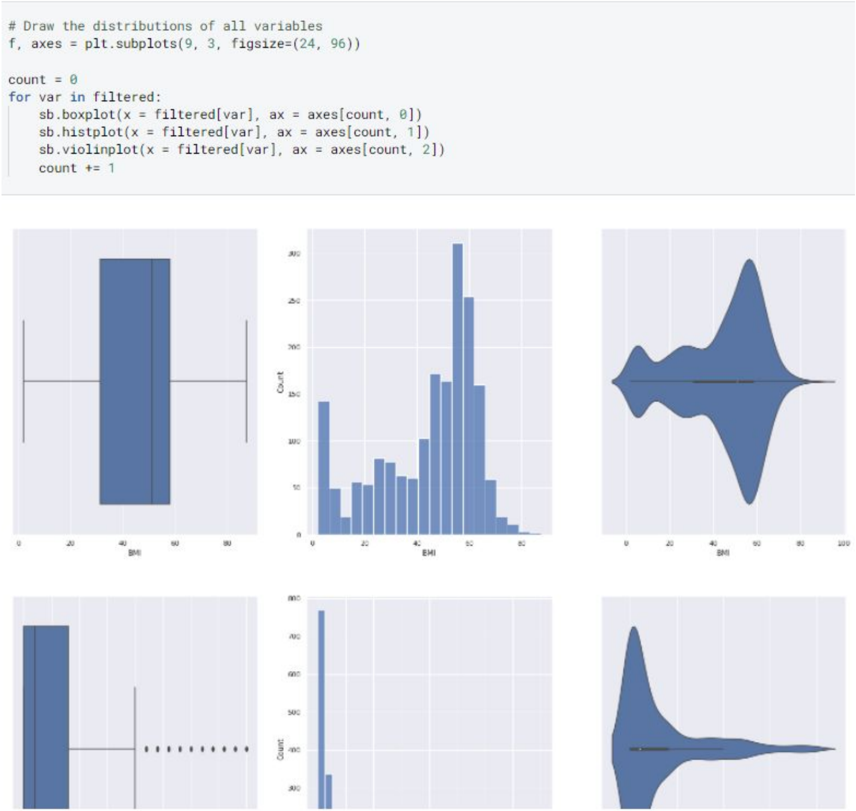
```
def remove_outlier(df, str):
    Q1 = df[str].quantile(0.25)
    Q3 = df[str].quantile(0.75)
    IQR = Q3 - Q1
    trueList = df[~((df[str] < (Q1 - 1.5 * IQR)) | (df[str] > (Q3 + 1.5 * IQR)))]
    return trueList

filtered = clean_data
outliers = ["Infant Deaths", "Measles", "Under-Five Deaths"]
for item in outliers:
  filtered = remove_outlier(filtered, item)

filtered
```

Remove: 1.5 IQR below 1st quartile &
1.5 IQR above 3rd quartile

Data Visualization

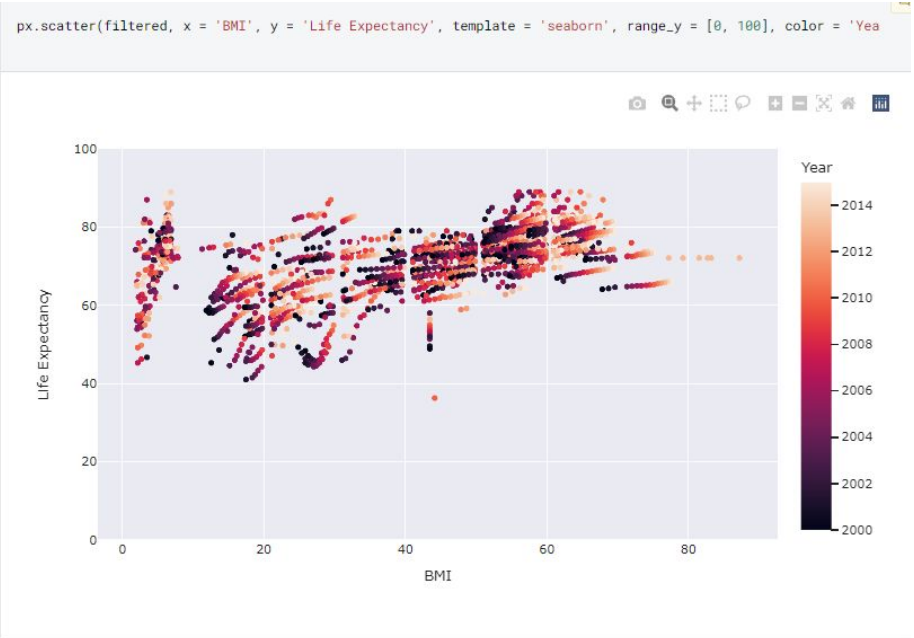


```
# Draw the distributions of all variables
f, axes = plt.subplots(9, 3, figsize=(24, 96))

count = 0
for var in filtered:
    sb.boxplot(x = filtered[var], ax = axes[count, 0])
    sb.histplot(x = filtered[var], ax = axes[count, 1])
    sb.violinplot(x = filtered[var], ax = axes[count, 2])
    count += 1
```



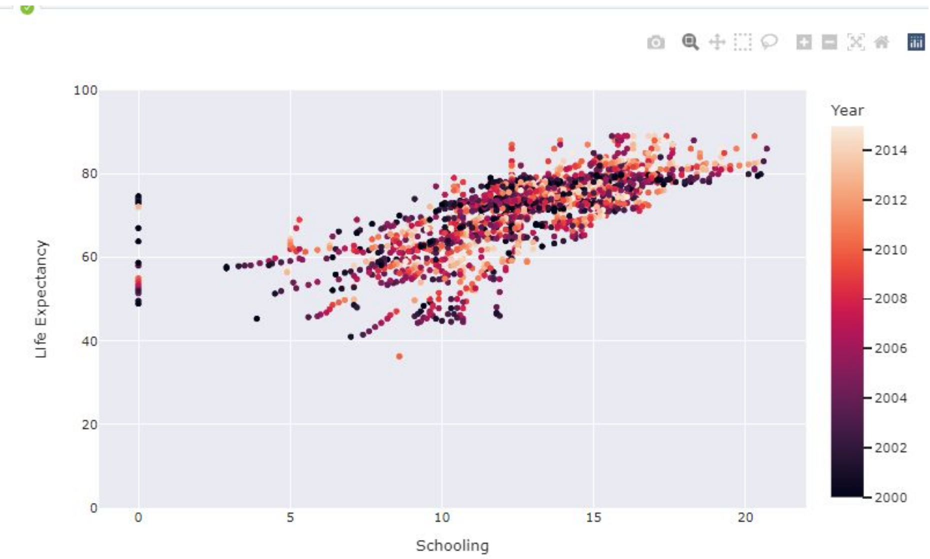
```
# Draw the distributions of all variables
f, axes = plt.subplots(9, 3, figsize=(24, 96))

count = 0
for var in filtered:
    sb.boxplot(x = filtered[var], ax = axes[count, 0])
    sb.histplot(x = filtered[var], ax = axes[count, 1])
    sb.violinplot(x = filtered[var], ax = axes[count, 2])
    count += 1
```

Data Visualization (Plotly)



```
px.scatter(filtered, x = 'BMI', y = 'Life Expectancy', template = 'seaborn', range_y = [0, 100], color = 'Yea
```

Life Expectancy and BMI



Life Expectancy and Alcohol

01. Practical Motivations

✓

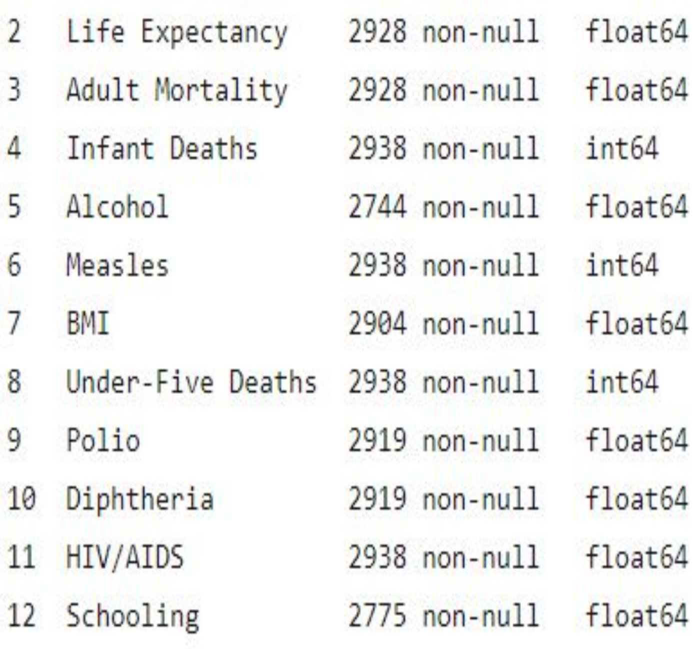
02. Exploratory analysis and data preparation

✓

03. Models and Machine Learning

04. Conclusion

03 Models and machine learning



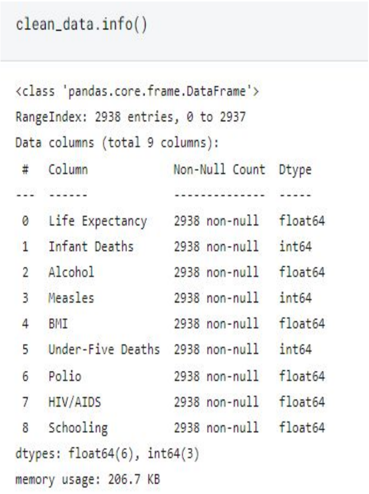
```
2   Life Expectancy    2928 non-null   float64
3   Adult Mortality    2928 non-null   float64
4   Infant Deaths      2938 non-null   int64
5   Alcohol            2744 non-null   float64
6   Measles            2938 non-null   int64
7   BMI                2904 non-null   float64
8   Under-Five Deaths  2938 non-null   int64
9   Polio              2919 non-null   float64
10  Diphtheria         2919 non-null   float64
11  HIV/AIDS           2938 non-null   float64
12  Schooling          2775 non-null   float64
```

Predictors

+ Adult mortality: (No of deaths per 1000)
+ Infant death: (No of deaths per 1000)
+ Alcohol : (total per capita (15+ years) consumption (in litres of pure alcohol))
+ Measles: (Total No of Cases)
+ BMI :(Average of entire Pop)
+ Under-five deaths (Number per 1000)
+ Polio (Immunization % of 1-year-olds)
+Diphtheria (Immunization % of 1-year-olds)
+ HIV / AIDS (Deaths per 1000)
+ Schooling (Average number of years studied)

Multivariate linear regression

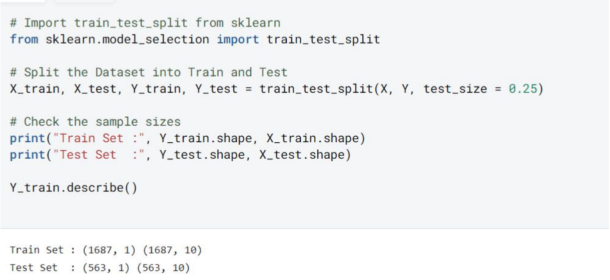
03. Models and Machine Learning



```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Life Expectancy   2938 non-null   float64
 1   Infant Deaths     2938 non-null   int64  
 2   Alcohol           2938 non-null   float64
 3   Measles           2938 non-null   int64  
 4   BMI               2938 non-null   float64
 5   Under-Five Deaths 2938 non-null   int64  
 6   Polio             2938 non-null   float64
 7   HIV/AIDS          2938 non-null   float64
 8   Schooling         2938 non-null   float64
dtypes: float64(6), int64(3)
memory usage: 206.7 KB
```

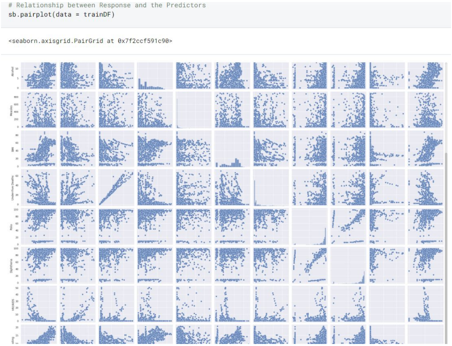
Predictors & Life expectancy



```
# Import train_test_split from sklearn
from sklearn.model_selection import train_test_split

# Split the Dataset into Train and Test
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25)

# Check the sample sizes
print("Train Set :", Y_train.shape, X_train.shape)
print("Test Set  :", Y_test.shape, X_test.shape)

Y_train.describe()
```

```
Train Set : (1687, 1) (1687, 10)
Test Set  : (563, 1) (563, 10)
```

75:25 split



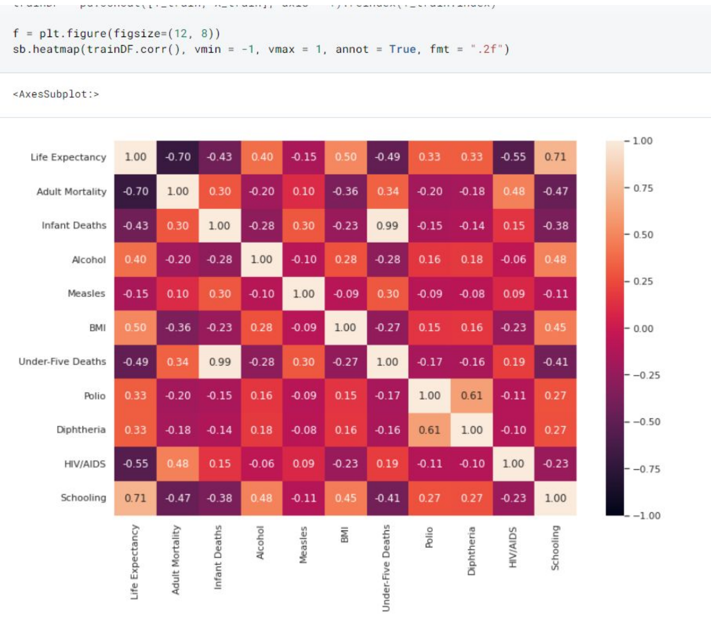
```
# Relationship between Response and the Predictors
sb.pairplot(data = trainDF)
```

```
<seaborn.axisgrid.PairGrid at 0x7f2ccf591c90>
```

Train Model

Multivariate linear regression(cont.)

03. Models and Machine Learning

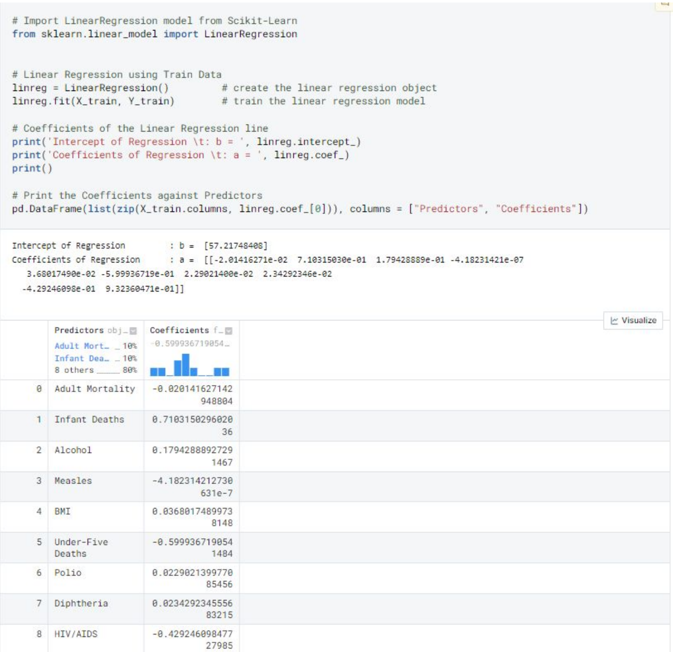


```
f = plt.figure(figsize=(12, 8))
sb.heatmap(trainDF.corr(), vmin = -1, vmax = 1, annot = True, fmt = ".2f")
```

```
<AxesSubplot:>
```

	Life Expectancy	Adult Mortality	Infant Deaths	Alcohol	Measles	BMI	Under-Five Deaths	Polio	Diphtheria	HIV/AIDS	Schooling
Life Expectancy	1.00	-0.70	-0.43	0.40	-0.15	0.50	-0.49	0.33	0.33	-0.55	0.71
Adult Mortality	-0.70	1.00	0.30	-0.20	0.10	-0.36	0.34	-0.20	-0.18	0.48	-0.47
Infant Deaths	-0.43	0.30	1.00	-0.28	0.30	-0.23	0.99	-0.15	-0.14	0.15	-0.38
Alcohol	0.40	-0.20	-0.28	1.00	-0.10	0.28	-0.28	0.16	0.18	-0.06	0.48
Measles	-0.15	0.10	0.30	-0.10	1.00	-0.09	0.30	-0.09	-0.08	0.09	-0.11
BMI	0.50	-0.36	-0.23	0.28	-0.09	1.00	-0.27	0.15	0.16	-0.23	0.45
Under-Five Deaths	-0.49	0.34	0.99	-0.28	0.30	-0.27	1.00	-0.17	-0.16	0.19	-0.41
Polio	0.33	-0.20	-0.15	0.16	-0.09	0.15	-0.17	1.00	0.61	-0.11	0.27
Diphtheria	0.33	-0.18	-0.14	0.18	-0.08	0.16	-0.16	0.61	1.00	-0.10	0.27
HIV/AIDS	-0.55	0.48	0.15	-0.06	0.09	-0.23	0.19	-0.11	-0.10	1.00	-0.23
Schooling	0.71	-0.47	-0.38	0.48	-0.11	0.45	-0.41	0.27	0.27	-0.23	1.00

correlation matrix



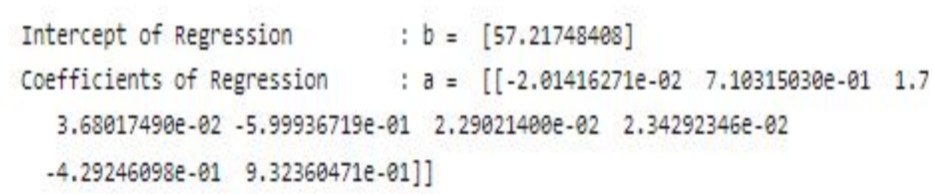
```
# Import LinearRegression model from Scikit-Learn
from sklearn.linear_model import LinearRegression


# Linear Regression using Train Data
linreg = LinearRegression()         # create the linear regression object
linreg.fit(X_train, Y_train)        # train the linear regression model

# Coefficients of the Linear Regression line
print('Intercept of Regression \t: b = ', linreg.intercept_)
print('Coefficients of Regression \t: a = ', linreg.coef_)
print()

# Print the Coefficients against Predictors
pd.DataFrame(list(zip(X_train.columns, linreg.coef_[0])), columns = ["Predictors", "Coefficients"])
```

```
Intercept of Regression         : b =  [57.21748408]
Coefficients of Regression      : a =  [[-2.01416271e-02  7.10315030e-01  1.79428889e-01 -4.18231421e-07
   3.68017490e-02 -5.99936719e-01  2.29021400e-02  2.34292346e-02
  -4.29246098e-01  9.32360471e-01]]
```

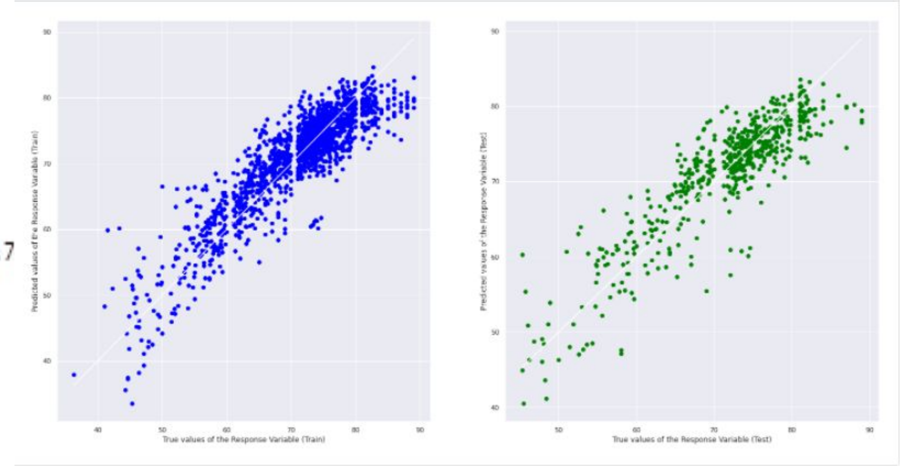
	Predictors obj…	Coefficients f…
0	Adult Mortality	-0.020141627142948804
1	Infant Deaths	0.71031502960236
2	Alcohol	0.17942888927291467
3	Measles	-4.182314212730631e-7
4	BMI	0.03680174899738148
5	Under-Five Deaths	-0.5999367190541484
6	Polio	0.022902139977085456
7	Diphtheria	0.02342923455683215
8	HIV/AIDS	-0.42924609847727985

The linear regression model

Multivariate linear regression(cont.)

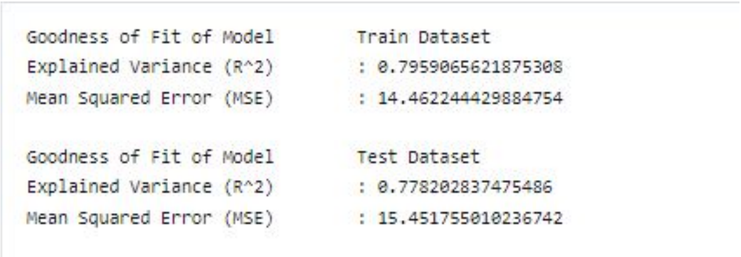


```
Intercept of Regression          : b =  [57.21748408]
Coefficients of Regression       : a =  [[-2.01416271e-02  7.10315030e-01  1.7
   3.68017490e-02 -5.99936719e-01  2.29021400e-02  2.34292346e-02
  -4.29246098e-01  9.32360471e-01]]
```



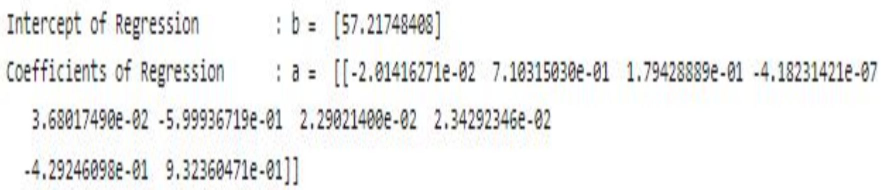
Life expectancy as a linear combination of the predictors.

Goodness of fit



```
Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)        : 0.795906562187530B
Mean Squared Error (MSE)        : 14.462244429884754

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)        : 0.778202837475486
Mean Squared Error (MSE)        : 15.451755010236742
```

Problem with multivariate regression



```
Intercept of Regression          : b =  [57.21748408]
Coefficients of Regression       : a =  [[-2.01416271e-02  7.10315030e-01  1.79428889e-01 -4.18231421e-07
   3.68017490e-02 -5.99936719e-01  2.29021400e-02  2.34292346e-02
  -4.29246098e-01  9.32360471e-01]]
```

$$Y = 57.217 + (-2.01e\text{^}{-2})X_1 + (7.10e\text{^}{-1})X_2 + (1.79e\text{^}{1})X_3 + (-4.18e\text{^}{-7})X_4 + (3.60e\text{^}{-2})X_5 + \ldots. + (9.32e\text{^}{-1})X_{10}$$

For X_n is the predictor variable. And Y life expectancy

Problem: Too many variables!

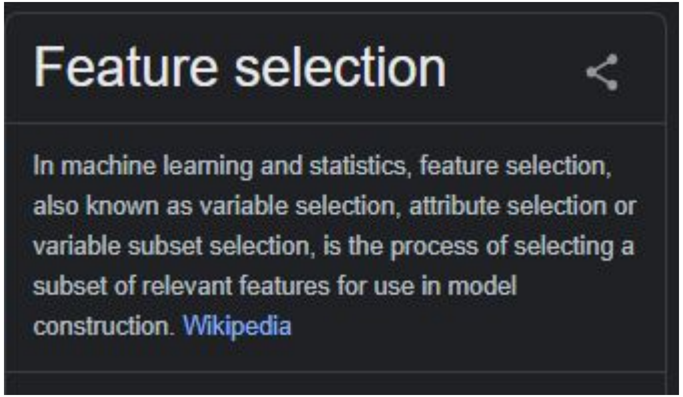
which one are the more important ones?

Solution: Feature Selection

$$Y = 57.217 + (-2.01e\text{^}-2)X_1 + (7.10e\text{^}-1)X_2 + (1.79e\text{^}1)X_3 + (-4.18e\text{^}-7)X_4 + (3.60e\text{^}-2)X_5 + \ldots + (9.32e\text{^}-1)X_{10}$$

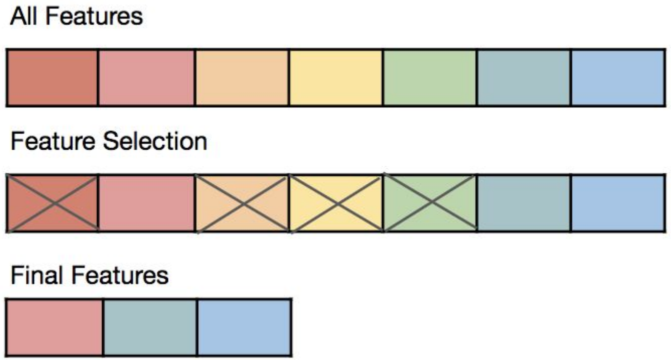
For X_n is the predictor variable. And Y life expectancy

Goal: Cut down to only the 3 most important variables



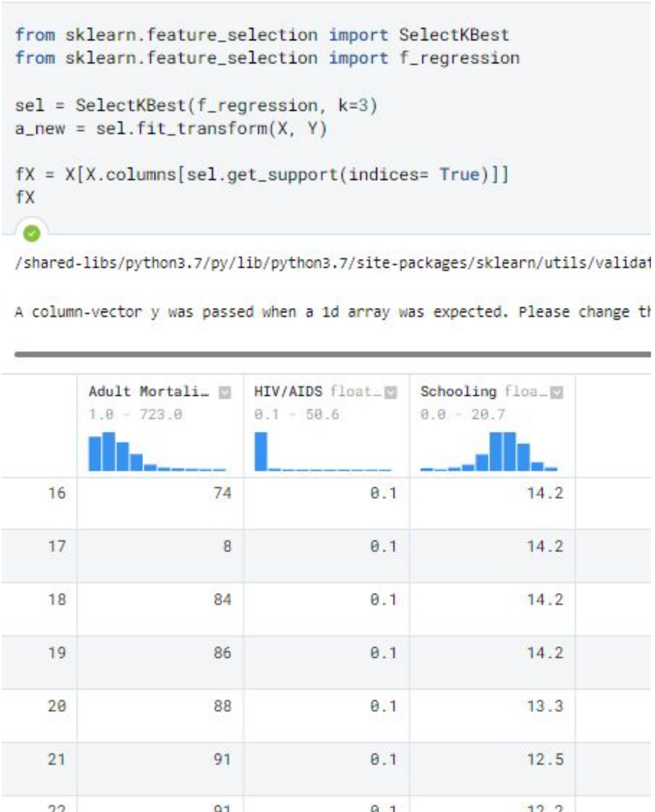
Feature selection

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in model construction. Wikipedia



Feature Selection

According to feature selection: The top 3 most important predictor variables are “Adult mortality”, “Schooling” and “HIV/AIDS”

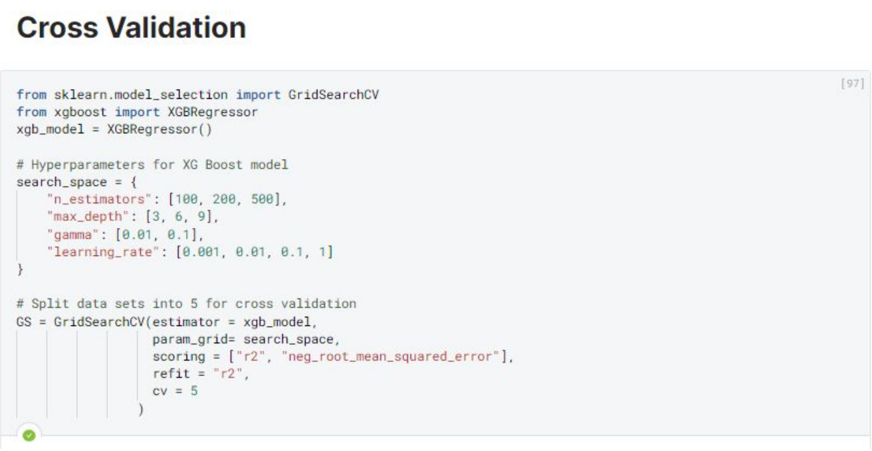


```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression

sel = SelectKBest(f_regression, k=3)
a_new = sel.fit_transform(X, Y)

fX = X[X.columns[sel.get_support(indices= True)]]
fX
```

```
/shared-libs/python3.7/py/lib/python3.7/site-packages/sklearn/utils/validat

A column-vector y was passed when a 1d array was expected. Please change th
```

	Adult Mortali… 1.0 - 723.0	HIV/AIDS float… 0.1 - 50.6	Schooling floa… 0.0 - 20.7
16	74	0.1	14.2
17	8	0.1	14.2
18	84	0.1	14.2
19	86	0.1	14.2
20	88	0.1	13.3
21	91	0.1	12.5
22	91	0.1	12.2

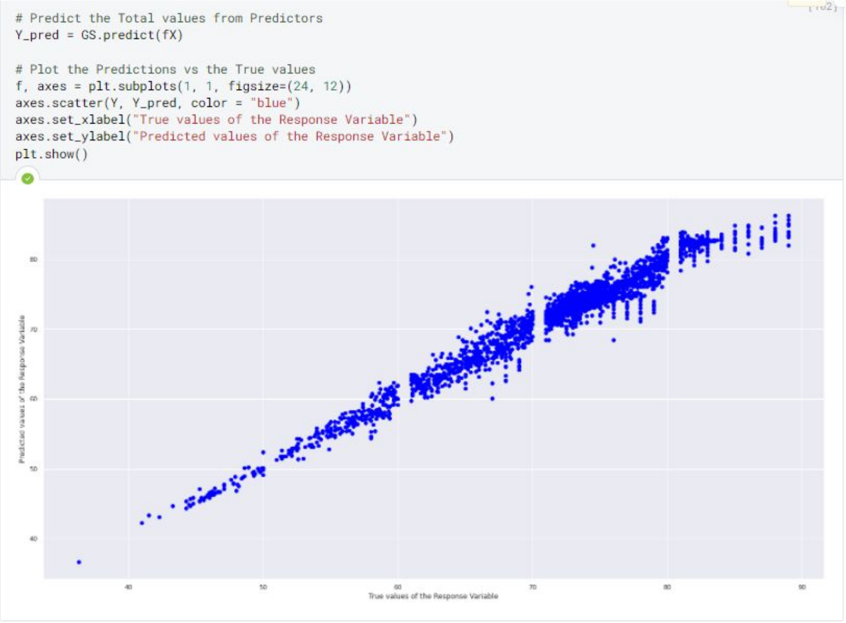
Comparing it against another model



Cross Validation

[97]

```
from sklearn.model_selection import GridSearchCV
from xgboost import XGBRegressor
xgb_model = XGBRegressor()

# Hyperparameters for XG Boost model
search_space = {
    "n_estimators": [100, 200, 500],
    "max_depth": [3, 6, 9],
    "gamma": [0.01, 0.1],
    "learning_rate": [0.001, 0.01, 0.1, 1]
}

# Split data sets into 5 for cross validation
GS = GridSearchCV(estimator = xgb_model,
                  param_grid= search_space,
                  scoring = ["r2", "neg_root_mean_squared_error"],
                  refit = "r2",
                  cv = 5
                  )
```



```
# Predict the Total values from Predictors
Y_pred = GS.predict(fX)

# Plot the Predictions vs the True values
f, axes = plt.subplots(1, 1, figsize=(24, 12))
axes.scatter(Y, Y_pred, color = "blue")
axes.set_xlabel("True values of the Response Variable")
axes.set_ylabel("Predicted values of the Response Variable")
plt.show()
```

Predicted values of the Response Variable
80
70
60
50
40
40
50
60
70
80
90
True values of the Response Variable

Cross-validation:
Partitioning the data set into 5 portions

Using XGB:
Explained Variance : 0.890

01. Practical Motivations ✓

02. Exploratory analysis and data preparation ✓

03. Models and Machine Learning ✓

04. Conclusion

04

Conclusion and insights

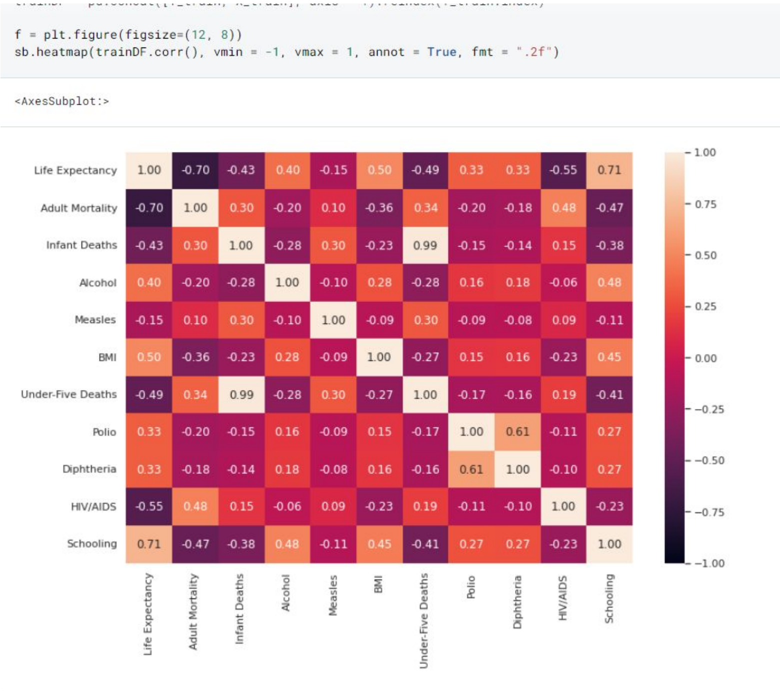
Our Problem Statement

Out of all of the predictors, how do they affect life expectancy?
Find out the more significant factors in affecting life expectancy.

What we have done

1. Coefficient Matrix
2. (Model 1)Multivariate Linear Regression Model with Feature Selection
3. (Model 2) XGBoost with Cross Validation

Out of all of the predictors, how do they affect life expectancy?



```
trainDF = pd.concat([y_train, X_train], axis = 1).reindex(y_train.index)

f = plt.figure(figsize=(12, 8))
sb.heatmap(trainDF.corr(), vmin = -1, vmax = 1, annot = True, fmt = ".2f")
```

```
<AxesSubplot:>
```

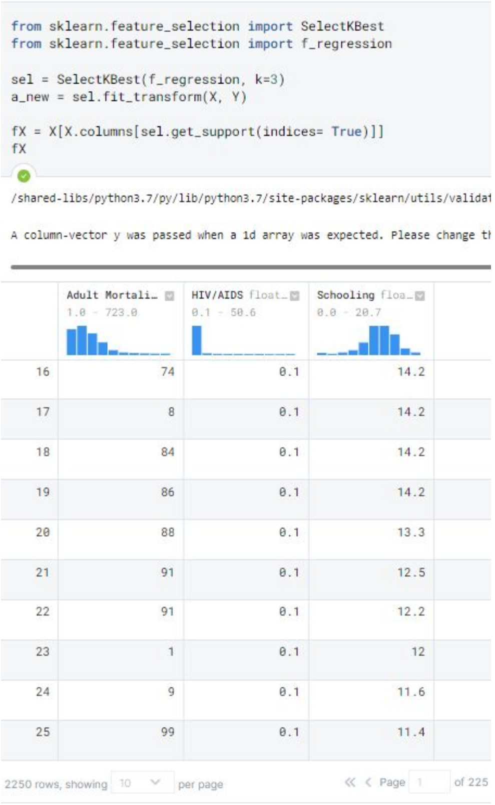
	Life Expectancy	Adult Mortality	Infant Deaths	Alcohol	Measles	BMI	Under-Five Deaths	Polio	Diphtheria	HIV/AIDS	Schooling
Life Expectancy	1.00	-0.70	-0.43	0.40	-0.15	0.50	-0.49	0.33	0.33	-0.55	0.71
Adult Mortality	-0.70	1.00	0.30	-0.20	0.10	-0.36	0.34	-0.20	-0.18	0.48	-0.47
Infant Deaths	-0.43	0.30	1.00	-0.28	0.30	-0.23	0.99	-0.15	-0.14	0.15	-0.38
Alcohol	0.40	-0.20	-0.28	1.00	-0.10	0.28	-0.28	0.16	0.18	-0.06	0.48
Measles	-0.15	0.10	0.30	-0.10	1.00	-0.09	0.30	-0.09	-0.08	0.09	-0.11
BMI	0.50	-0.36	-0.23	0.28	-0.09	1.00	-0.27	0.15	0.16	-0.23	0.45
Under-Five Deaths	-0.49	0.34	0.99	-0.28	0.30	-0.27	1.00	-0.17	-0.16	0.19	-0.41
Polio	0.33	-0.20	-0.15	0.16	-0.09	0.15	-0.17	1.00	0.61	-0.11	0.27
Diphtheria	0.33	-0.18	-0.14	0.18	-0.08	0.16	-0.16	0.61	1.00	-0.10	0.27
HIV/AIDS	-0.55	0.48	0.15	-0.06	0.09	-0.23	0.19	-0.11	-0.10	1.00	-0.23
Schooling	0.71	-0.47	-0.38	0.48	-0.11	0.45	-0.41	0.27	0.27	-0.23	1.00

correlation matrix

Positive correlation	Negative Correlation
-**Alcohol** (Total consumption per capita), -**Polio** (Immunisation % of 1 year old) -**BMI** -**Diphtheria**(Immunisation % of 1 year olds) -**HIV/AIDS**(Immunisation % of 1 year olds) -**Schooling**(Average number of years studied)	-**Adult mortality**: (No of deaths per 1000) -**Infant death**: (No of deaths per 1000) -**Measles**: (Total No of Cases) -**Under-five-deaths** (number per 1000)

Which are the more significant factors in affecting life expectancy?

According to feature selection: The top 3 most important predictor variables are “Adult mortality”, “Schooling” and “HIV/AIDS”



```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression

sel = SelectKBest(f_regression, k=3)
a_new = sel.fit_transform(X, Y)

fX = X[X.columns[sel.get_support(indices= True)]]
fX
```

```
/shared-libs/python3.7/py/lib/python3.7/site-packages/sklearn/utils/validat

A column-vector y was passed when a 1d array was expected. Please change th
```

	Adult Mortali… 1.0 - 723.0	HIV/AIDS float… 0.1 - 50.6	Schooling floa… 0.0 - 20.7
16	74	0.1	14.2
17	8	0.1	14.2
18	84	0.1	14.2
19	86	0.1	14.2
20	88	0.1	13.3
21	91	0.1	12.5
22	91	0.1	12.2
23	1	0.1	12
24	9	0.1	11.6
25	99	0.1	11.4

2250 rows, showing 10 per page

Page 1 of 225

Final Insights and recommendations

-Schooling(education), seems to have the greatest impact.

-To improve life expectancy : Focus on predictors that have a greater impact – namely schooling, HIV/aids immunization and preventing adult mortality

Credits

Projection done by: Jeremy Ong, Qian Cheng, Rhys Wong . Class of 2022, SC1015

Github Repo: https://github.com/iiJoe/WHOLifeExpectancy

Data set taken from :
https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik and illustrations