Icon

Description automatically generated

[**MACHINE LEARNING AND PATTERN RECOGNITION**](https://elearning.dbs.ie/course/view.php?id=15834#section-0)

**(B9DA109)**

**CA1**

**SUPERVISED MACHINE LEARNING – REGRESSION**

**Submitted To:**

**Prof. Satya Prakash**

1. **Saima Salim (10603707)**
2. **Anjali Dhondiba Patil (10601725)**
3. **Adithya Ajitha (10605164)**

Table of Contents

[The Dataset 2](#_Toc99318965)

[Dataset Description 2](#_Toc99318966)

[Variable Types 2](#_Toc99318967)

[Data Preparation 3](#_Toc99318968)

[Mode 4](#_Toc99318969)

[Median 4](#_Toc99318970)

[Mean 5](#_Toc99318971)

[Outliers Detection and Capping 6](#_Toc99318972)

[Outliers Assignments into Dataset 9](#_Toc99318973)

[Statistical representation of Data 9](#_Toc99318974)

[Linear Regression 10](#_Toc99318975)

[Heatmap 13](#_Toc99318976)

[Regularization 13](#_Toc99318977)

[Stochastic gradient 13](#_Toc99318978)

[Conclusion 16](#_Toc99318979)

[References 17](#_Toc99318980)

# The Dataset

The dataset is taken from the website [www.kaggle.com](http://www.kaggle.com). Following is the link to the dataset:

<https://www.kaggle.com/kumarajarshi/life-expectancy-who>

The concerned dataset is about the analysis of the factors that affect the life expectancy in the developing and developed countries. The data ranges from 2000-2015. The analysis of the dataset involves the search to find the variables that have a major impact on the life expectancy factor of the humans. For example, analysing and predicting the variables with the aim to study the relationship and to determine the extent to which the variables like adult mortality, smoking, different disease variables have an impact on the life expectancy. This also involves predicting and determine whether economic factors have also an impact on the life expectancy. This whole task is done my means of finding the correlation among the variables (Kaggle)

## Dataset Description

There are 22 columns and 2938 rows.

Following are the columns:

Year, Status, country, life expectancy, AdultMortality, infantdeaths, Alcohol, percentageExpenditure, HepatitisB, Measles, BMI, under-five deaths, Polio, Totalexpenditure, Diphtheria, HIV/AIDS, GDP, Population, thinness (1-19 years), thinness (5-9 years), IncomecCmposition-of- resources, Schooling.

The dataset is taken into consideration just as it was when downloaded, except that the spaces between the variable names are removed to make it easy to process the code. For example, the original column is named as Hepatitis B and for the analysis, the name is changed to HepatitisB.

The predicting variables that are taken into consideration are as given below:

Life expectancy, AdultMortality, infantdeaths, Alcohol, percentageExpenditure, HepatitisB, Measles, BMI, under-five deaths, Polio, Totalexpenditure, Diphtheria, HIV/AIDS, GDP, Population, IncomecCmposition-of- resources, Schooling.

## Variable Types

Quantitative variables are numeric and measurable e.g., age or height of a person. Quantitative variables are further classified into two categories i.e., Discrete and Continuous.

Discrete variables are those whose value is obtained by counting and in a certain amount of time e.g. number of passengers in a train. In this dataset, Following are the discrete variables:

AdultMortality, infantdeaths, HepatitisB, Polio, under-five deaths, Diphtheria, Population

Continuous Variable are those that take infinite amount of time to count. Infact, in between two points, there are infinite number of values e.g. weight. In this dataset, following are the continuous variables:

Lifeexpect, Alcohol, percentageExpenditure, BMI, Totalexpenditure, GDP, HIV/AIDs, thinness(1-19 years),thinness(5-9 years), IncomecCmposition-of- resources,Schooling

Qualitative variables cannot be measured and are categorical e.g. gender (male or female).

These are classified as Nominal and Ordinal variables.

Nominal variables: These variables cannot be ordered. For example, names of countries or persons. In this dataset, Country, Year are the nominal variables.

Ordinal variables: These variables can be sorted or ordered in either way. For example, level of humidity that is high medium or low. In this dataset, Status is the ordinal variable.

Choice of dependent and independent variables

As we are considering the effect of various variables on the life expectancy, therefore, life expectancy is taken as dependent variable and the rest are taken as independent variables.

Further, we have excluded the variables like infantdeaths, under-five deaths, thinness(1-19 years),thinness(5-9 years), population because these have not got any apparent effect on the life expectancy variable.

Selection of Algorithm

As life expectancy is a continuous variable, we select linear regression as the algorithm for our purpose.

# Data Preparation

**Statistics of the dataset**

df.describe(include =['int64','float64'] )

is used to know the mean, standard deviation, minimum, maximum, count values of the column variables.

**Checking for Duplicate Values**

The dataset is checked for the presence of the duplicate values as given below:

df.duplicated()

It is found that there are no duplicate values in the dataset.

**Binarizing**

The Status variable is categorized as developed and developing. These entities are binarized as:

* Developed =1
* Developing= 0

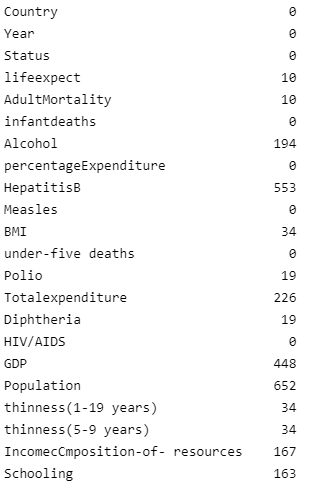
df['Status'] = df['Status'].map({'Developed':1,'Developing':0})

**Null Values and their Treatment**

The following code is used to find the null values:

print(df.isnull().sum())

The outcome of the code is:

****

As we can see, population, GDP and HepatitisB have higher number of null values. For filling in the null values we first check the skewness of the data columns.

There are different ways to deal with the null values. These can either be omitted (removed) or can be filled with mean, median or mode. Appropriate method is to fill the null values by mean, median or mode.

Mean, median mode are the measures of central tendency. A measure of central tendency is a ‘point estimate’ that describes the entire data or the distribution using a single value (Azizi S. 2022).

# Mode

Mode is the frequency of the occurrence of a data point in a data or distribution. For example, in the list given below, mode is 3 because it has more times than 1 and 2 (Hayes, 2021):

1,1,2,2,3,3,3

In a list, there can be more than one mode. For example:

1,1,1,2,2,3,3,3

The mode in the above example is: 1 and 3.

**Advantages**

* This is not affected by extreme values, outliers, skewed data.
* It is used mostly for qualitative data and categorical variables.

**Disadvantages**

* Mode does not consider on all values in a dataset.
* In a small dataset, it is not reliable.

# Median

Median is the middle value of the distribution when the data is sorted. For example, consider the following list:

S= {-1,5,3,0,2,10,11}

After sorting:

S= { -1,0,2,3,5,10,11}

In this case there are seven numbers and middle value is 3. So median is 3.

**Advantages**

* Median is not affected by the outliers or skewed data.
* If the data is skewed, then median is an appropriate measure as because mean is affected by the extreme values.

**Disadvantages**

* Median does not account for all the values in data. It accounts for only the middle values.

# Mean

Mean is the average of the values of the distribution.

For example:

D=2,5,7,4,6,10

Mean=2+5+7+4+6+10/6= 5.67

**Advantages**

* Unlike median and mode, it accounts for all the values in a data.

**Disadvantages**

* Mean is sensitive to the outliers.

As mean is sensitive to the extreme value and outliers, mode is used for categorical variables, the best way to deal with the null values is to use median.

Therefore, for the dataset under consideration, median is used to fill in the null values.

df['AdultMortality'] = df['AdultMortality'].fillna(df['AdultMortality'].median())

df['Alcohol'] = df['Alcohol'].fillna(df['Alcohol'].median())

df['HepatitisB'] = df['HepatitisB'].fillna(df['HepatitisB'].median())

df['BMI'] = df['BMI'].fillna(df['BMI'].median())

df['Polio'] = df['Polio'].fillna(df['Polio'].median())

df['Totalexpenditure'] = df['Totalexpenditure'].fillna(df['Totalexpenditure'].median())

df['Diphtheria '] = df['Diphtheria '].fillna(df['Diphtheria '].median())

df['GDP'] = df['GDP'].fillna(df['GDP'].median())

df['Population'] = df['Population'].fillna(df['Population'].median())

df['Schooling'] = df['Schooling'].fillna(df['Schooling'].median())

df['IncomecCmposition-of- resources'] = df['IncomecCmposition-of- resources'].fillna(df['IncomecCmposition-of- resources'].median())

df['lifeexpect'] = df['lifeexpect'].fillna(df['lifeexpect'].median())

df[' thinness(1-19 years)'] = df[' thinness(1-19 years)'].fillna(df[' thinness(1-19 years)'].median())

df[' thinness(5-9 years)'] = df[' thinness(5-9 years)'].fillna(df[' thinness(5-9 years)'].median())

df[' HIV/AIDS'] = df[' HIV/AIDS'].fillna(df[' HIV/AIDS'].median())

**(geeksforgeeks, 2020; DataAnalytics, 2022)**

After filling in the null values, we check again whether all the null values have been filled with the median. The screenshot shows that now there are no null values in the dataset.

df.isnull.sum()

Graphical user interface, text, application

Description automatically generated

# Outliers Detection and Capping

Outliers are the data points that are different or outside the range of the given data. Outliers if present and left untreated affect the accuracy of the result. Therefore, detection of the outliers and their capping is an essential part of data pre-processing in machine learning algorithms.

There are different ways to determine the presence of outliers as given below:

If the data is normally distributed, z-score method is used.

However, if the data is skewed, boxplot method is used.

The best way to determine the presence of outliers is by using boxplot diagram.

To check whether the data is skewed or normally distributed, we have used boxplot and distribution plot. We have done the analysis for all the data columns in the dataset.

plt.figure(figsize=(16,8))

plt.subplot(8,4,1)

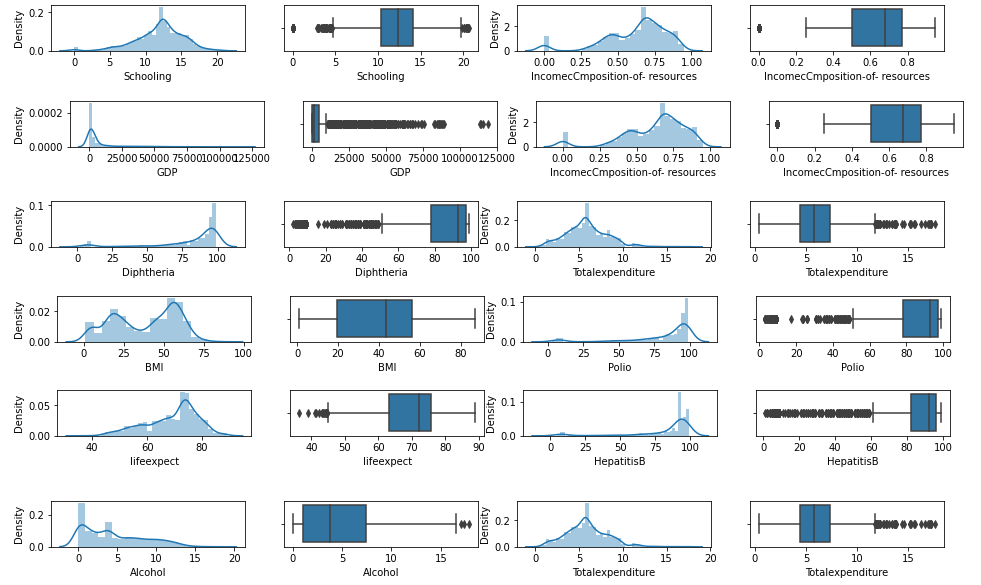
sns.distplot(df['ColumnName'])

plt.subplot(8,4,2)

sns.boxplot(df['ColumnName'])

plt.show()

**(Kaggle, 2021)**



From the diagrams, we can see that except Schooling variable, there is skewness in all the dataset variables. Schooling variable is normally distributed.

The plotted boxplots are showing the presence of outliers. For the treatment of the outliers, we have used capping method by IQR (Interquartile range) method, based on percentile methods.

percentile25\_ ColumnName = df['ColumnName'].quantile(0.25)

percentile75\_ ColumnName = df[' ColumnName '].quantile(0.75)

iqr\_ ColumnName =percentile75\_ ColumnName -percentile25\_ ColumnName

upper\_limit\_ ColumnName = percentile75\_Schooling + 1.5 \* iqr\_ ColumnName

lower\_limit\_ ColumnName = percentile25\_Schooling - 1.5 \* iqr\_ ColumnName

df[df[' ColumnName '] > upper\_limit\_ ColumnName]

df[df[' ColumnName '] < lower\_limit\_ ColumnName]

#Trimming

new\_df\_ ColumnName = df[df[' ColumnName '] < upper\_limit\_ ColumnName]

new\_df\_ ColumnName.shape

#Capping

new\_df\_cap\_ ColumnName = df.copy()

new\_df\_cap\_ ColumnName [' ColumnName '] = np.where(

new\_df\_cap\_ ColumnName [' ColumnName’] > upper\_limit\_ ColumnName,

upper\_limit\_ ColumnName,

np.where(

new\_df\_cap\_ ColumnName [' ColumnName '] < lower\_limit\_ ColumnName,

lower\_limit\_ ColumnName,

new\_df\_cap\_ ColumnName [' ColumnName’]

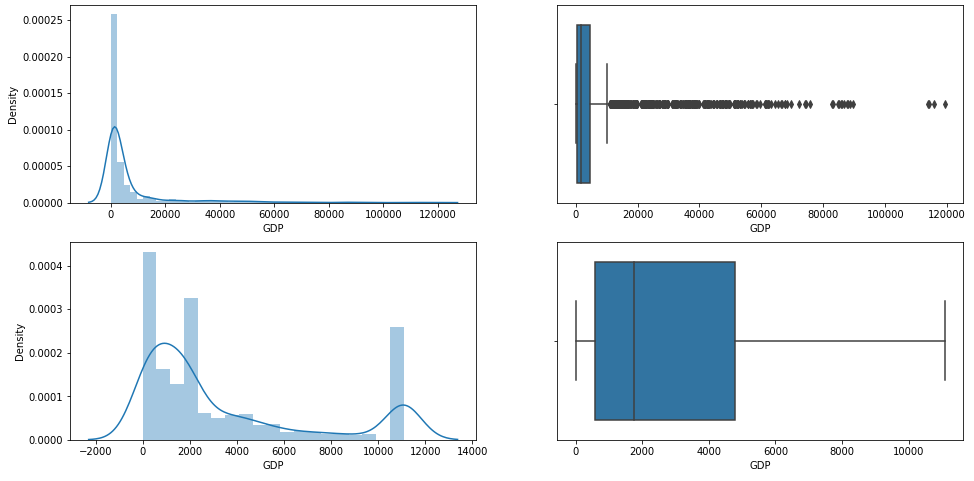
)

)

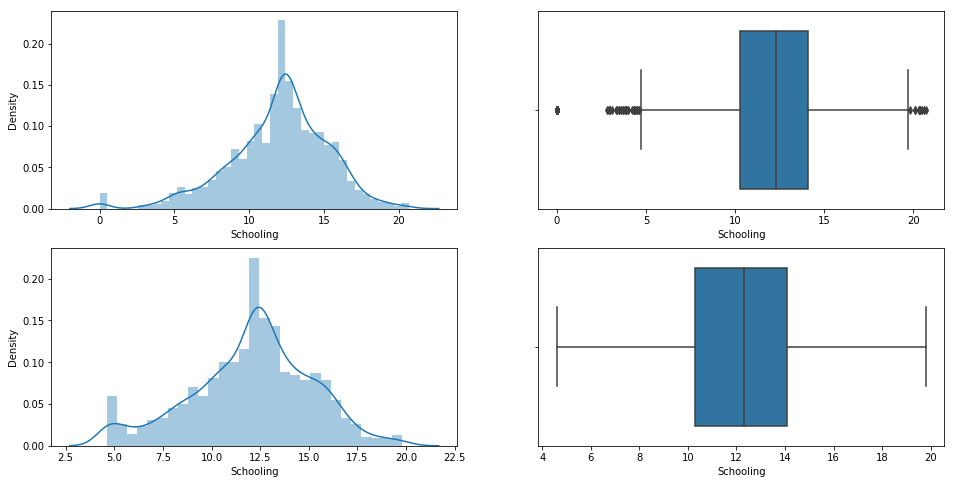
After capping the outliers, we checked again the dataset for the presence of the outliers.

**(Analytics Vidhya, 2021)**

**Output:**



**Output:**



# Outliers Assignments into Dataset

After detecting outliers from dataset, we have assigned it to columns in order to perform regression method.

Assigned trimmed and capped values to new column as below

Table

Description automatically generated

# Statistical representation of Data

Statistical representation of Data after pre-processing of data. Normal Distribution and probability plot can be seen in below figure.

Chart, line chart, histogram

Description automatically generated

Feature Scaling

It is used to accommodate drastically fluctuating magnitudes, values, or units during data pre-processing. The fluctuating variance of these separate features is reduced to a fixed range via feature scaling.

Table

Description automatically generated

Coefficient

By indicating the direction of association with the coefficient, you can say that when the predictor variable(y) increases, the response variable(X) increases as well.

Table

Description automatically generated

# Linear Regression

Linear regression is a supervised machine learning method where it finds a best fit linear line between the dependent and independent variable. It finds the linear relationship between those two variables It works based on two principles.

1. Ordinary least square
2. Mean square error.

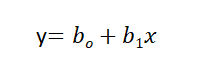
Linear regression of two types:

1. Simple linear regression
2. Multiple simple linear regression

In both the above methods, only one independent variable is available and the model has to find the linear relationship with the dependent variable.

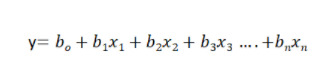
In Multiple linear regression, there are more than one independent variable is present in the model to find the linear relationship

Equation for simple linear relationship



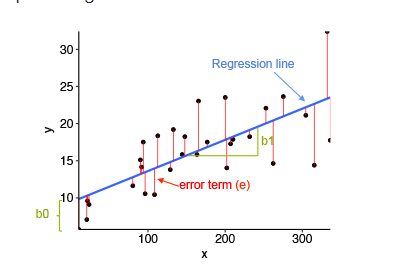
bo is the intercept, b1 is coefficient or slope, x is the independent variable and y is the dependent variable.

Equation of Multiple Linear Regression



where bo is the intercept, b1, b2, b3, b4…, bn are coefficients or slopes of the independent variables x1, x2, x3, x4…, xn and y is the dependent variable.

The main aim of linear regression is to find the best fit linear line and optimal values of intercept and other coefficient with the error minimised.



(Image Source: Statistical tools for high-throughput data analysis)

The key features of linear regression are

1. Assumptions
2. The type of output variable
3. Link function

**Assumptions** under linear regression include

1. Conditional independence

If Ei-prediction error of the sample I, Ej -prediction of the sample j, then Ei is assumed as independent of Ej

1. Gaussianity

Assumed to be distributed normally, prediction error shows that the output variable should be normally distributed.

1. Linearity

It shows the linear relation between input and output variables.

**The type of output variable**

It is mainly used when the variables are continuous real numbers.

**Link function**

The link function is a mathematical function that connects output variable (s) to the set of input variables in a SML system.

For linear regression, the following steps should be applied

1. Preparing the data
2. Training the model
3. Predicting and accuracy check

We have already cleaned the data and selected lifeexpect as dependent variable and independent variables are AdultMortality, Alcohol, percentageExpenditure, HepatitisB, BMI, Polio, Totalexpenditure, Diphtheria, HIV/AIDS, GDP, IncomecCmposition-of- resources, Schooling.

We have also done training the model and predict the test set result.

Now moving on to the linear regression.Tuning the SGDRegressor parameters 'eta0' (learning rate) and 'max\_iter' using Grid Search

from sklearn.model\_selection import GridSearchCV

sgdr = SGDRegressor(random\_state = 1, penalty = None)

grid\_param = {'eta0': [.0001, .001, .01, .1, 1], 'max\_iter':[10000, 20000, 30000, 40000]}

gd\_sr = GridSearchCV(estimator=sgdr, param\_grid=grid\_param, scoring='r2', cv=5)

gd\_sr.fit(x\_after\_Standardisation, y)

results = pd.DataFrame.from\_dict(gd\_sr.cv\_results\_)

print("Cross-validation results:\n", results)

best\_parameters = gd\_sr.best\_params\_

print("Best parameters: ", best\_parameters)

best\_result = gd\_sr.best\_score\_ # Mean cross-validated score of the best\_estimator

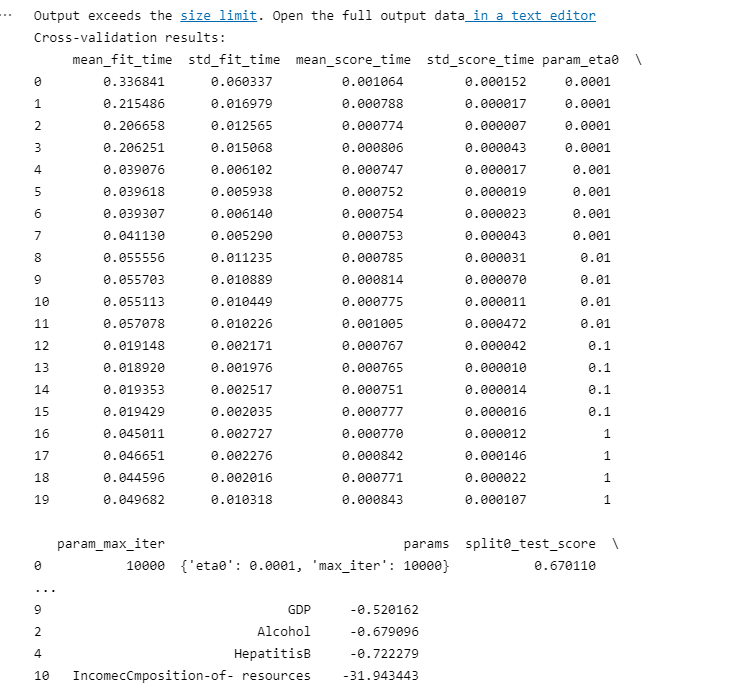
print("Best result: ", best\_result)

best\_model = gd\_sr.best\_estimator\_

print("Intercept: ", best\_model.intercept\_)

print(pd.DataFrame(zip(X.columns, best\_model.coef\_), columns=['Features','Coefficients']).sort\_values(by=['Coefficients'],ascending=False))

**The output is:**



# Heatmap

We will build a heatmap with Python by using the Seaborn library, which is a graphical representation of a matrix of values.

Chart

Description automatically generated

In heatmap values for features with respect to target variable y which is life expectancy, if have value nearer to 1 are selected for model training and evaluation and which are nearer to -1 can be dropped out as it has lowest value.

For example Infantdeaths, Measles, under-five deaths, thinness(1-19 years), thinness(5-9 years),HIV/AIDS, population with respect Life expectancy has lowest values in heatmap and have been dropped in feature selection.

However, life expect is correlated to Schooling, Income Composition-of-resources,GDP,Diphteria,Totalexpenditure, Polio, BMI, HepatitisB,percentageexpenditure, Alcohol and are selected during model development.

# Regularization

Regularization entails calibrating machine learning models in order to reduce the adjusted loss function and reduce overfitting and underfitting. By doing so, we are able to fit our machine learning model appropriately on our test data and thus maximize the performance.

## Stochastic gradient

Stochastic gradient descent is commonly utilized in machine learning applications as an inexact but powerful way of determining the parameters of a model that correspond to the best fit between expected and actual outputs.

In Grid Search, the hyperparameters and their values are all combined and the performance is calculated for each combination, and the best hyperparameter value is selected.

Table

Description automatically generated

Table

Description automatically generated

Comparing Predicted and actual variable using regressor object from liner regression model.

Table

Description automatically generated

In order to calculate Mean Absolute error, Mean Squared error and Root mean squared error we have imported metrics from sklearn library.

**R Squared:** According to R-Squared, the variance explained by independent variables in a regression model is equal to the variance of the dependent variable.

**Mean Absolute error:** Essentially, absolute error is the difference between the prediction and the true value of an observation. MAE evaluates the magnitude of errors by taking the average of absolute errors for a group of predictions and observations.

**MSE:** The MSE (Mean Square Error) is obtained by squaring the error value and later taking the mean of them. The difference between the observed and predicted values of that dataset, this error value is computed.

**RMSE:** Root Mean Square Error RMSE i.e., Root Mean Square Error RMSE is a standard method for calculating a model's error when predicting quantitative data. It is calculated in the same units as that of the dependent variable and is also the square root of mean square error (MSE).

Text

Description automatically generated

# Conclusion

As per the calculations done above, we got sgdr value as 0.684796 and linear regressor as 0. 7377737.Sgdr is method used to increase the efficiency of a regression model. But in our dataset (Life expectancy) score for linear regression is greater than sgdr. Sdgr is low because our dataset has only 2939 rows which is comparatively low.

Model’s error which is RMSE error for the dataset is 4.798

# References

DBS. (n.d.). *Linear Regression*. Retrieved from study.dbs.ie: https://study.dbs.ie/2122/msc-data/B9DA109/u1/index.html#/lessons/xh7roomMu5ucuNUP7IoVUAazXPidSmwN

GOYAL, C. (2021, May 19). *Feature Engineering – How to Detect and Remove Outliers (with Python Code)*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/

Hayes, A. (2021, October 21). *Mode*. Retrieved from investopedia: https://www.investopedia.com/terms/m/mode.asp#:~:text=In%20statistics%2C%20the%20mode%20is,average%20value%20in%20the%20data.

ISLAM, A. (2021, March). *Tutorial: Python Subplots*. Retrieved from Kaggle: https://www.kaggle.com/code/asimislam/tutorial-python-subplots/notebook

Kumar, A. (2021, October 3). *Python – Replace Missing Values with Mean, Median & Mode*. Retrieved from vitalflux: https://vitalflux.com/pandas-impute-missing-values-mean-median-mode/

Kumar, S. (2020, November 16). *Replacing missing values using Pandas in Python*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/replacing-missing-values-using-pandas-in-python/

RAJARSHI, K. (2018, March). *Life Expectancy (WHO)*. Retrieved from Kaggle: https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who

Robinson, S. (2021, June 7). *Linear Regression in Python with Scikit-Learn*. Retrieved from stackabuse: https://stackabuse.com/linear-regression-in-python-with-scikit-learn/