SC1015 PROJECT

Predicting Life Expectancy

By: Rohit, Husen & Kok wai

LIFE EXPECTANCY

PLANNING FOR THE FUTURE

QUALITY OF LIFE

HEALTHCARE DECISIONS

SOCIETAL PLANNING

OUR MOTIVATIONS

- Global life expectancy at birth has increased significantly from an average of 31 to 72 (1900 -> 2016)
- Life expectancy is influenced by a range of factors
- But out of these factors, which is the most important?

OUR PROJECT

What is the most critical factor in determining one's life expectancy?

DATASETS USED

01 World Happiness Report

02 Population Data

CLEANING THE DATA

sorted the values alphabetically

added 2 datasets together

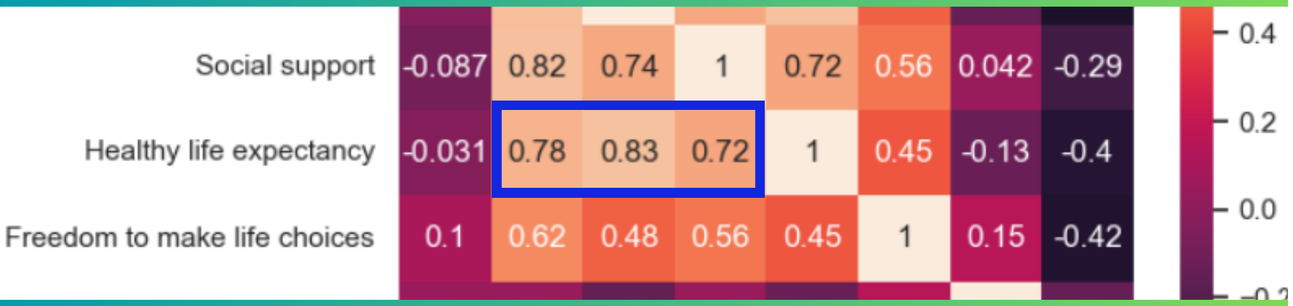
removed the NAN values

DATA EXPLORATION

Correlation Matrix to find the relationship of variables compared to each other

DATA EXPLORATION

Best 3 Correlation Variables



LOGGED GDP PER CAPITA

Happiness Score

Social Support

MACHINE LEARNING

Linear Regression

OR

Classification Tree

09

LINEAR REGRESSION

THE TRAINING DATASET FOR LINEAR REGRESSION WITH R^2 VALUES

GDP and HLE Data

OLS Regression Results

Dep. Variable:	Healthy life expectancy	**R-squared:**	0.696
Model:	OLS	**Adj. R-squared:**	0.692
Method:	Least Squares	**F-statistic:**	198.9
Date:	Fri, 21 Apr 2023	**Prob (F-statistic):**	3.39e-24
Time:	11:42:03	**Log-Likelihood:**	-225.38
No. Observations:	89	**AIC:**	454.8
Df Residuals:	87	**BIC:**	459.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	28.7253	2.610	11.008	0.000	23.539	33.912
Logged GDP per capita	3.8457	0.273	14.104	0.000	3.304	4.388

Omnibus:	20.771	**Durbin-Watson:**	1.730
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	58.773
Skew:	0.693	**Prob(JB):**	1.73e-13
Kurtosis:	6.732	**Cond. No.**	77.3

Happiness and HLE Data

OLS Regression Results

Dep. Variable:	Healthy life expectancy	**R-squared:**	0.580
Model:	OLS	**Adj. R-squared:**	0.575
Method:	Least Squares	**F-statistic:**	120.1
Date:	Fri, 21 Apr 2023	**Prob (F-statistic):**	4.59e-18
Time:	11:42:15	**Log-Likelihood:**	-239.73
No. Observations:	89	**AIC:**	483.5
Df Residuals:	87	**BIC:**	488.4
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	39.6421	2.367	16.747	0.000	34.937	44.347
Happiness score	4.4691	0.408	10.958	0.000	3.659	5.280

Omnibus:	0.211	**Durbin-Watson:**	2.100
Prob(Omnibus):	0.900	**Jarque-Bera (JB):**	0.069
Skew:	0.068	**Prob(JB):**	0.966
Kurtosis:	3.013	**Cond. No.**	36.9

Social Support and HLE Data

OLS Regression Results

Dep. Variable:	Healthy life expectancy	**R-squared:**	0.550
Model:	OLS	**Adj. R-squared:**	0.545
Method:	Least Squares	**F-statistic:**	106.5
Date:	Fri, 21 Apr 2023	**Prob (F-statistic):**	8.96e-17
Time:	11:42:24	**Log-Likelihood:**	-242.75
No. Observations:	89	**AIC:**	489.5
Df Residuals:	87	**BIC:**	494.5
Df Model:	1		
Covariance Type:	nonrobust		

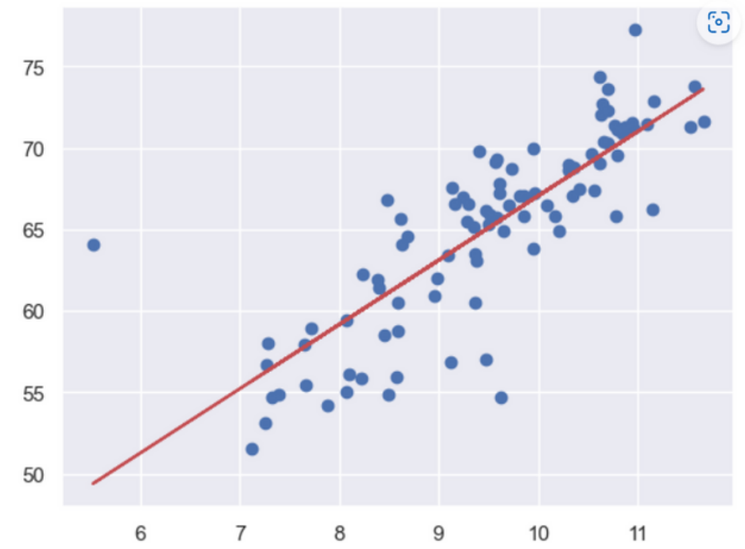
	coef	std err	t	P>|t|	[0.025	0.975]
const	39.5607	2.519	15.702	0.000	34.553	44.568
Social support	31.9568	3.096	10.321	0.000	25.803	38.111

Omnibus:	1.831	**Durbin-Watson:**	1.859
Prob(Omnibus):	0.400	**Jarque-Bera (JB):**	1.289
Skew:	-0.270	**Prob(JB):**	0.525
Kurtosis:	3.237	**Cond. No.**	12.9

LINEAR REGRESSION LINE

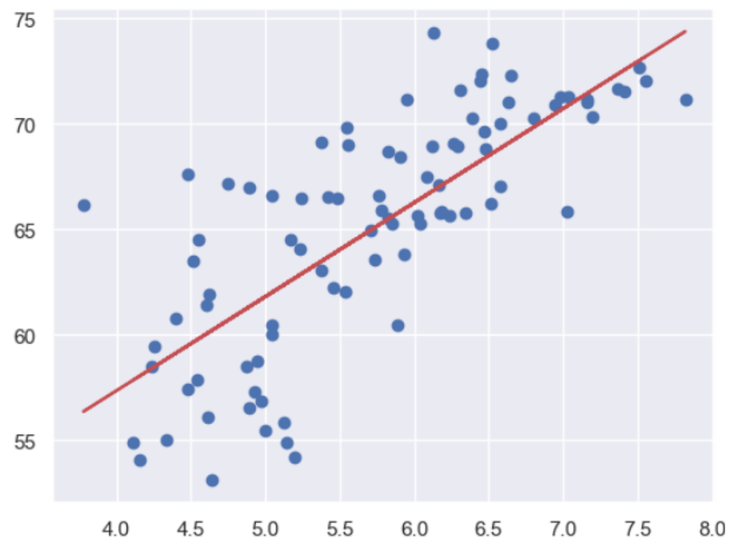
GDP and HLE Data

In [39]:
```
# Visualizing the regression line
plt.scatter(X_train, y_train)
plt.plot(X_train, 27.5494 + 3.9509*X_train, 'r')
plt.show()
```



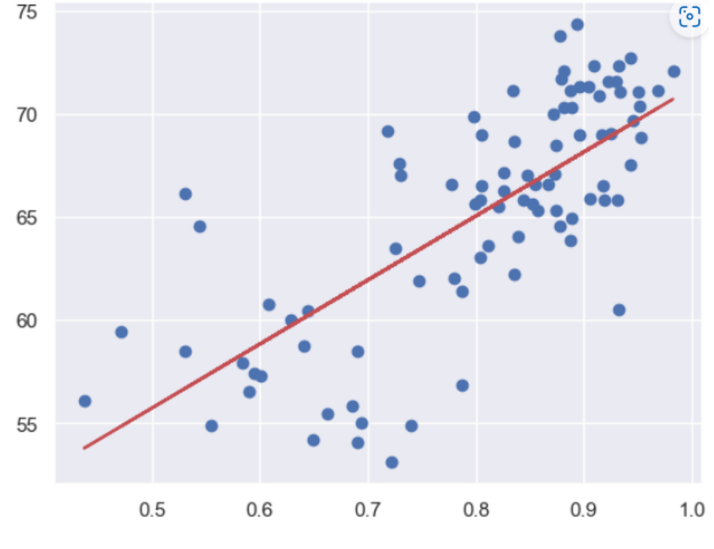
Happiness and HLE Data

```
# Visualizing the regression line
plt.scatter(X_train, y_train)
plt.plot(X_train, 39.5179 + 4.4594*X_train, 'r')
plt.show()
```



Social Support and HLE Data

In [47]:
```
# Visualizing the regression line
plt.scatter(X_train, y_train)
plt.plot(X_train, 40.2326 + 31.0050*X_train, 'r')
plt.show()
```

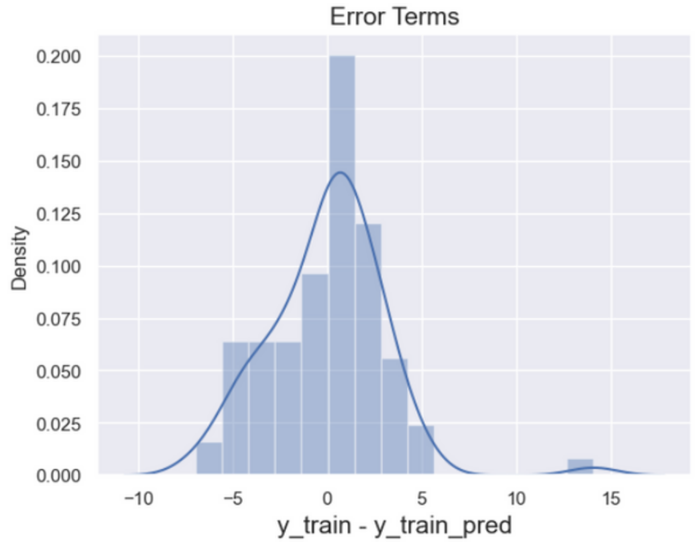


HISTOGRAM OF RESIDUAL ERRORS

GDP and HLE Data

```
# Plotting the histogram using the residual values
fig = plt.figure()
sb.distplot(res, bins = 15)
plt.title('Error Terms', fontsize = 15)
plt.xlabel('y_train - y_train_pred', fontsize = 15)
plt.show()
```

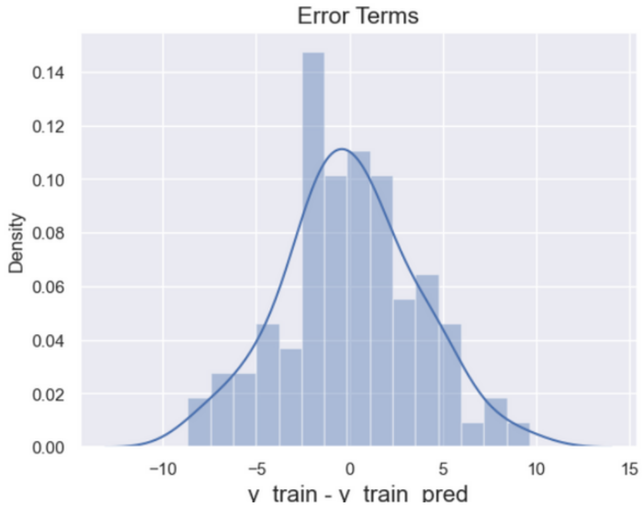
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fut
n and will be removed in a future version. Please adapt your code to use eith
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



Happiness and HLE Data

```
# Plotting the histogram using the residual values
fig = plt.figure()
sb.distplot(res, bins = 15)
plt.title('Error Terms', fontsize = 15)
plt.xlabel('y_train - y_train_pred', fontsize = 15)
plt.show()
```

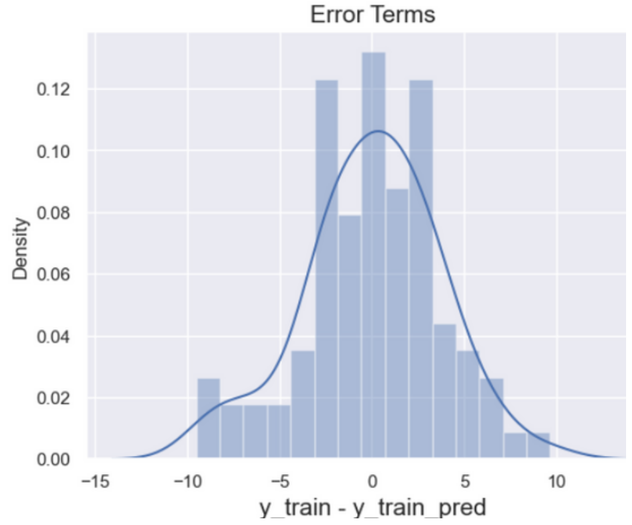
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619:
tureWarning: `distplot` is a deprecated function and will be removed in a
ture version. Please adapt your code to use either `displot` (a figure-le
function with similar flexibility) or `histplot` (an axes-level function
histograms).
  warnings.warn(msg, FutureWarning)
```



Social Support and HLE Data

```
# Plotting the histogram using the residual values
fig = plt.figure()
sb.distplot(res, bins = 15)
plt.title('Error Terms', fontsize = 15)
plt.xlabel('y_train - y_train_pred', fontsize = 15)
plt.show()
```

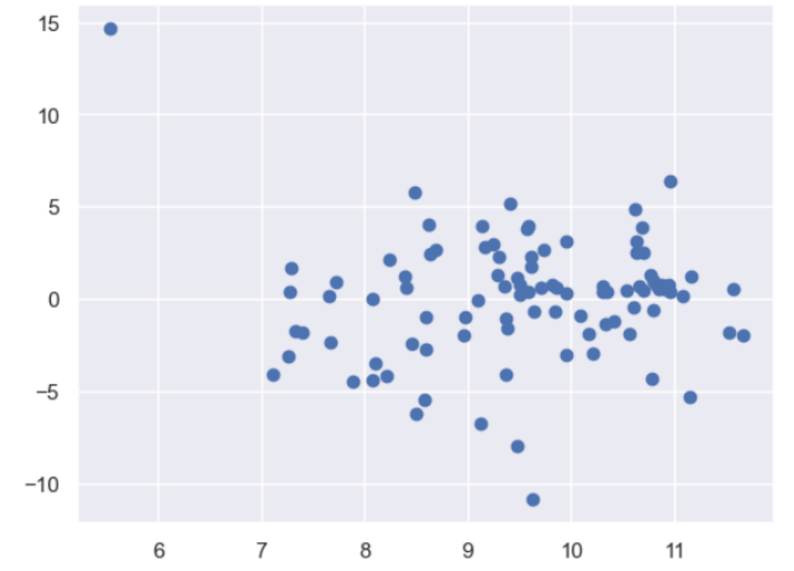
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:
n and will be removed in a future version. Please adapt your code to
ilar flexibility) or `histplot` (an axes-level function for histogra
  warnings.warn(msg, FutureWarning)
```



FINDING PATTERNS IN RESIDUAL

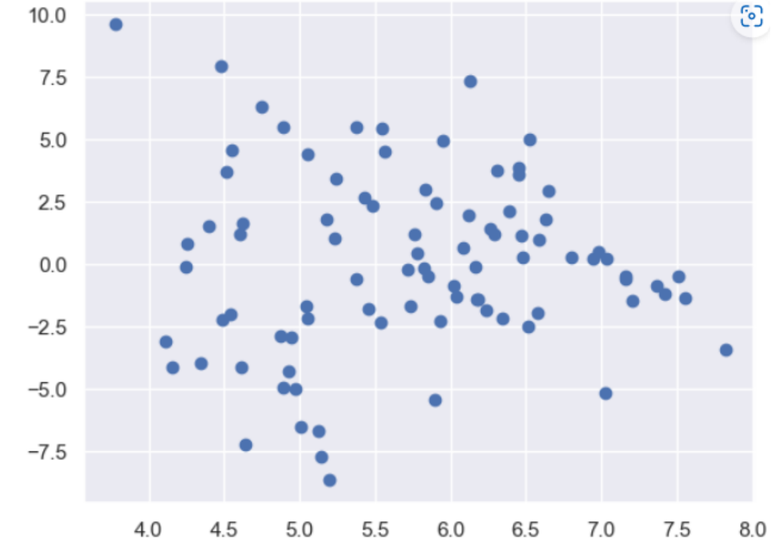
GDP and HLE Data

```
# Looking for any patterns in the residuals
plt.scatter(X_train,res)
plt.show()
```



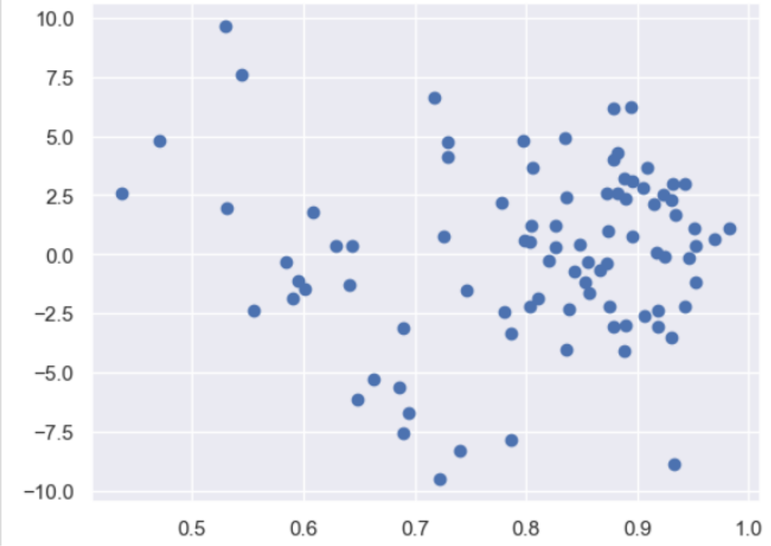
Happiness and HLE Data

```
# Looking for any patterns in the residuals
plt.scatter(X_train,res)
plt.show()
```



Social Support and HLE Data

```
# Looking for any patterns in the residuals
plt.scatter(X_train,res)
plt.show()
```



FITTING THE LINE ON THE TEST DATA

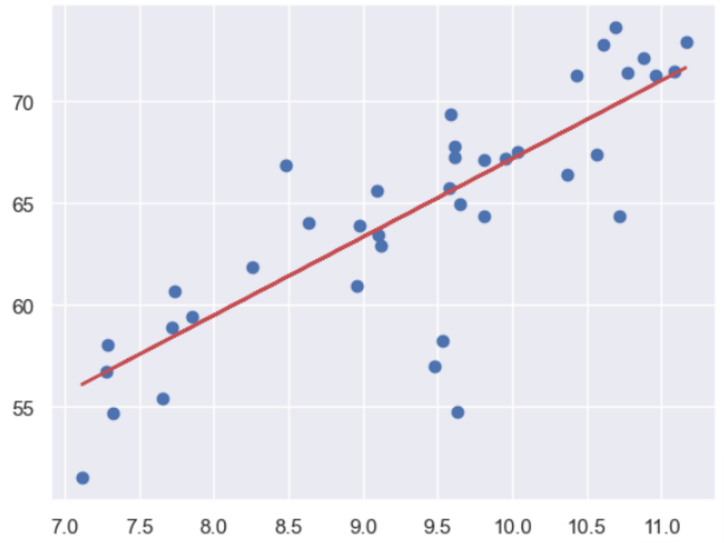
GDP and HLE Data

```
# Importing r2_square
from sklearn.metrics import r2_score

# Checking the R-squared value
r_squared = r2_score(y_test, y_test_pred)
r_squared
```

```
0.6674145255500075
```

```
# Visualize the line on the test set
plt.scatter(X_test, y_test)
plt.plot(X_test, y_test_pred, 'r')
plt.show()
```



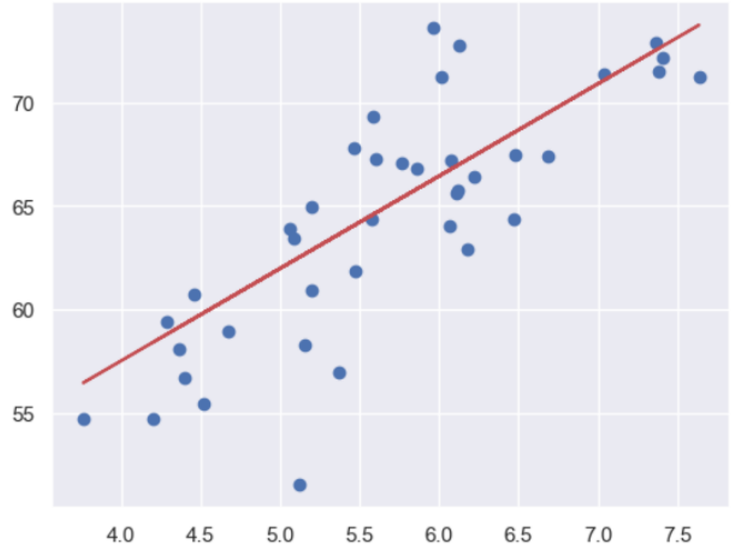
Happiness and HLE Data

```
# Importing r2_square
from sklearn.metrics import r2_score

# Checking the R-squared value
r_squared = r2_score(y_test, y_test_pred)
r_squared
```

```
0.636833239321823
```

```
# Visualize the line on the test set
plt.scatter(X_test, y_test)
plt.plot(X_test, y_test_pred, 'r')
plt.show()
```



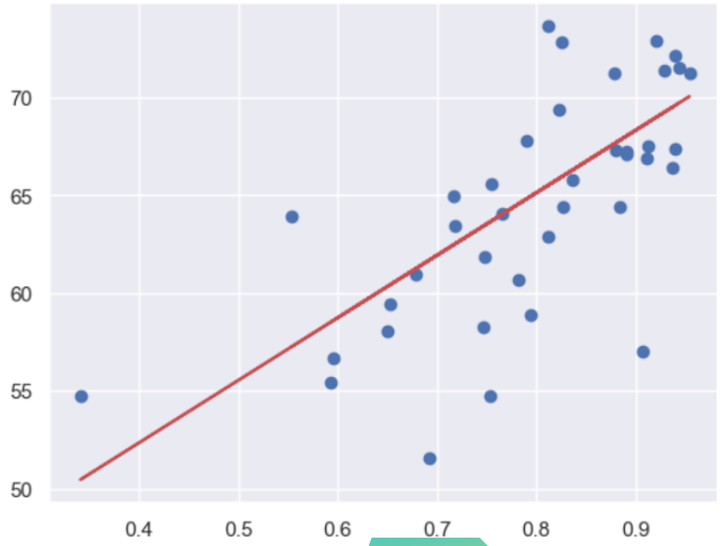
Social Support and HLE Data

```
# Importing r2_square
from sklearn.metrics import r2_score

# Checking the R-squared value
r_squared = r2_score(y_test, y_test_pred)
r_squared
```

```
0.44972725362574306
```

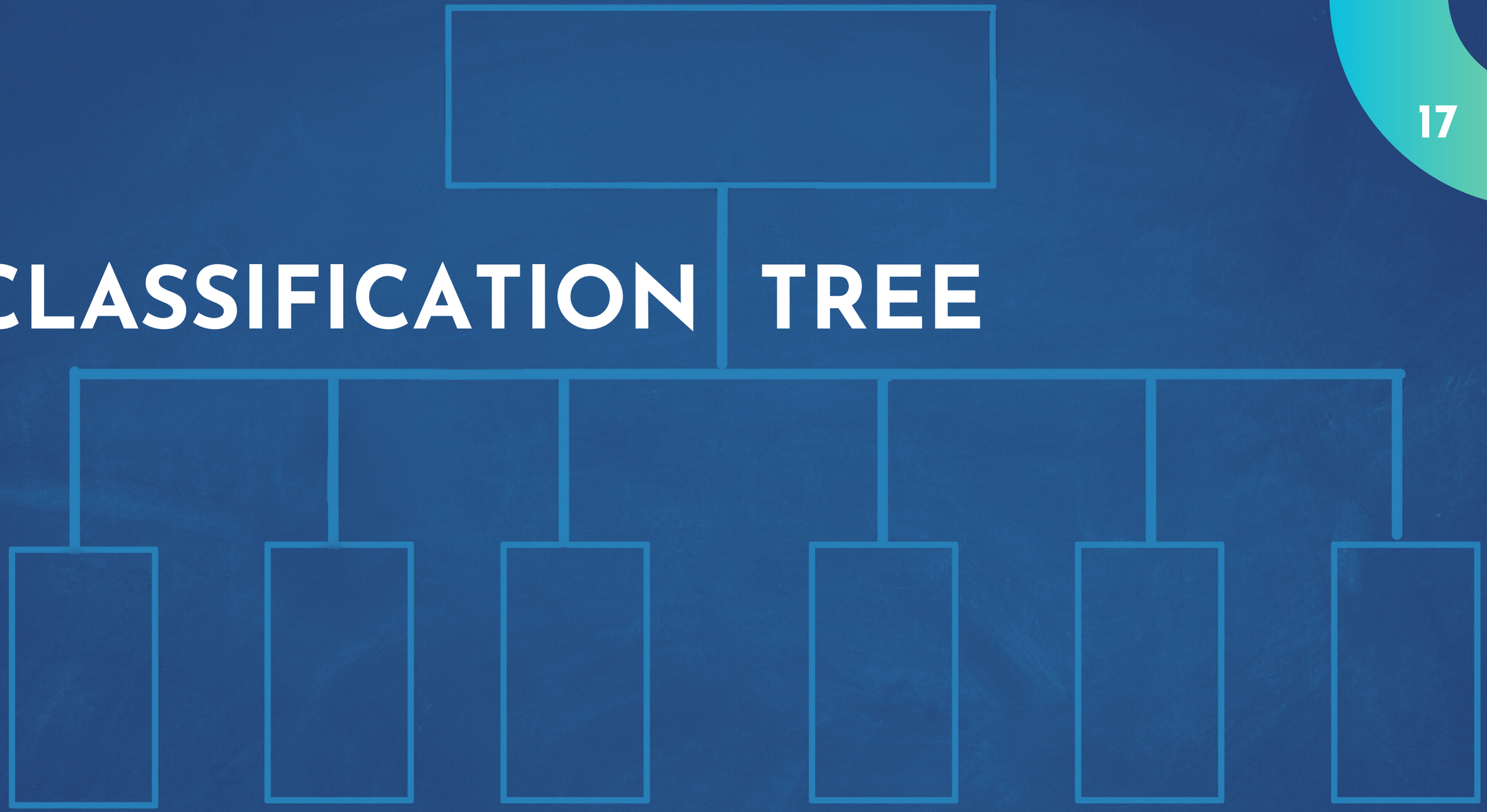
```
# Visualize the line on the test set
plt.scatter(X_test, y_test)
plt.plot(X_test, y_test_pred, 'r')
plt.show()
```



LINEAR REGRESSION

GDP has the highest correlation to Healthy Life Expectancy

CLASSIFICATION TREE

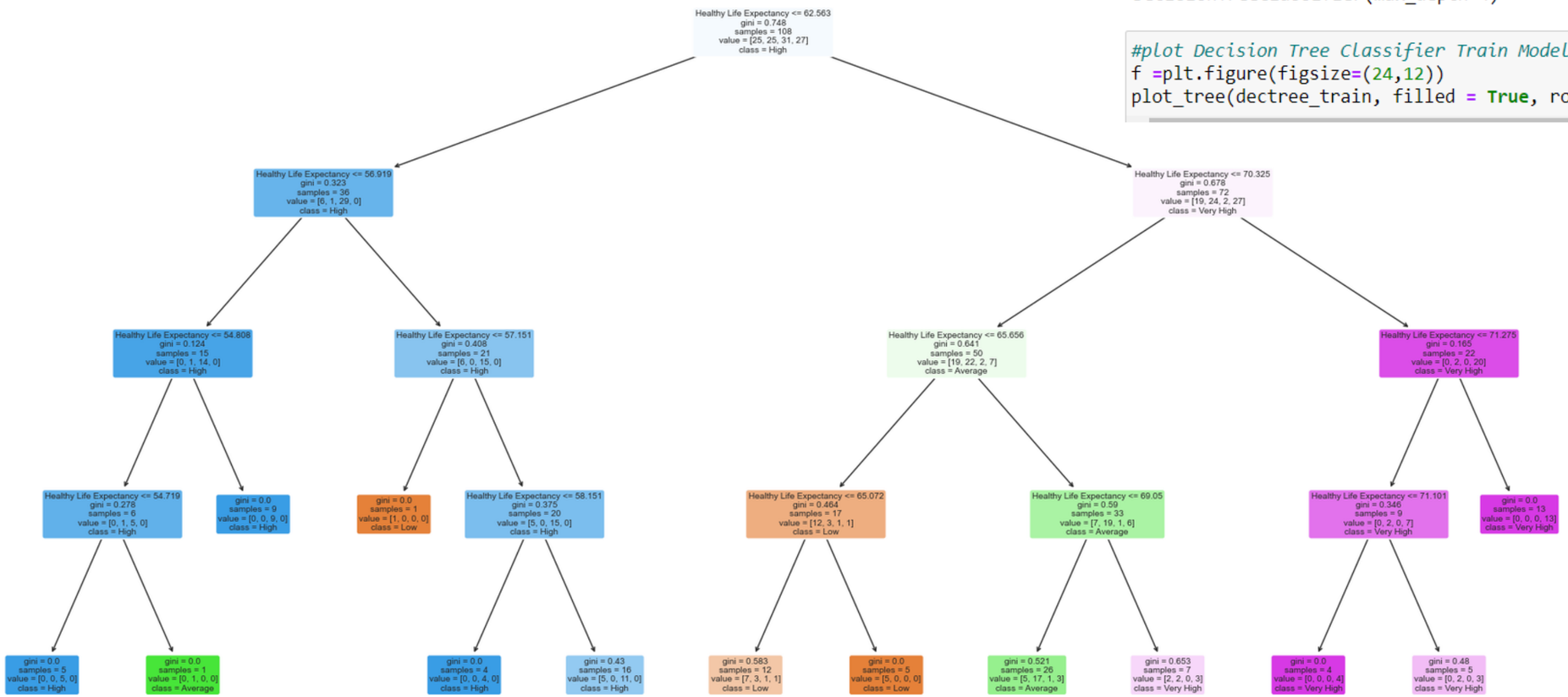


CLASSIFICATION TREE


```
# Train the Decision Tree Classifier train model
dectree_train.fit (Healthylife_train,Happy_train)
```

```
DecisionTreeClassifier(max_depth=4)
```

```
#plot Decision Tree Classifier Train Model
f =plt.figure(figsize=(24,12))
plot_tree(dectree_train, filled = True, rounded = True, feature_names = ["Healthy Life Expectancy
```



In [91]:
```
# Split the data into training and test sets
Healthylife_train,Healthylife_test,Happy_train,Happy_test = train_test_split(targets,Happy_categorical,test_size = 0.2)
dectree_train = DecisionTreeClassifier(max_depth = 4)
dectree_test = DecisionTreeClassifier(max_depth = 4)
```

ISSUES FACED

A

The data is not in categorical format

B

Predictors variables are in 1D array

WHAT DID WE DO?

A. Discretization

B. Reshape Array

C. Changed Continuous Variables to Categorical Variables

D. Changing the predictors variable to 2D

DISCRETIZATION

Creating Interval through Median and Standard Deviation

In [88]:
```
#to find a suitable range to be put for the labels to change the d
lower= np.percentile(predictors,25)
middle = np.median(predictors)
upper = np.percentile(predictors,75)
maximum = predictors.max()
print(lower)
print(middle)
print(upper)
print(maximum)
```

Proceed to change the data to categorical with the range made

39]:
```
#changing the datas to categorical
bins = [0,lower,middle,upper,maximum+1]
labels = ['Low','Average','High','Very High']
```

90]:
```
Happy_categorical = pd.cut(predictors, bins=bins, labels=labels)
Happy_categorical
```

RESHAPE ARRAY

Changing the Predictor Data to a 2D array

```
#reshape the data from a 1D array to a 2D array
Healthylife_train = np.array(Healthylife_train)
Healthylife_train = Healthylife_train.reshape(-1, 1)
Healthylife_train
```

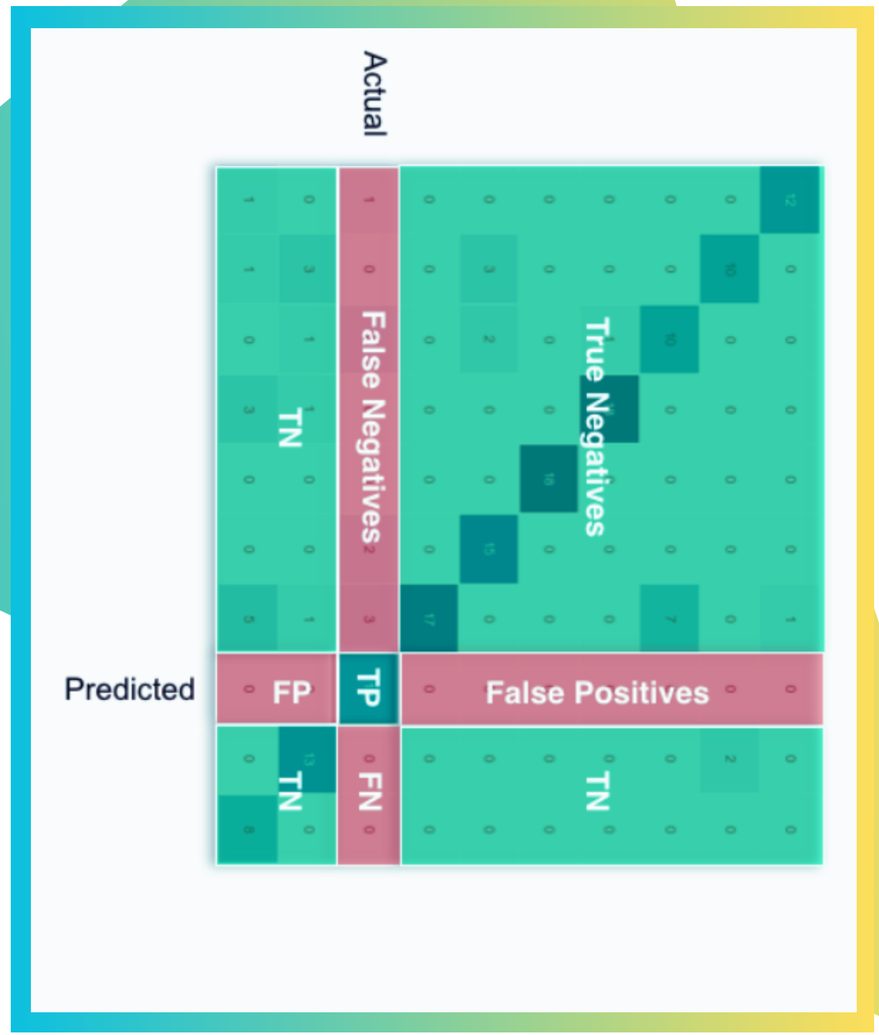
CONFUSION MATRIX

To show the Accuracy of the Classification Model

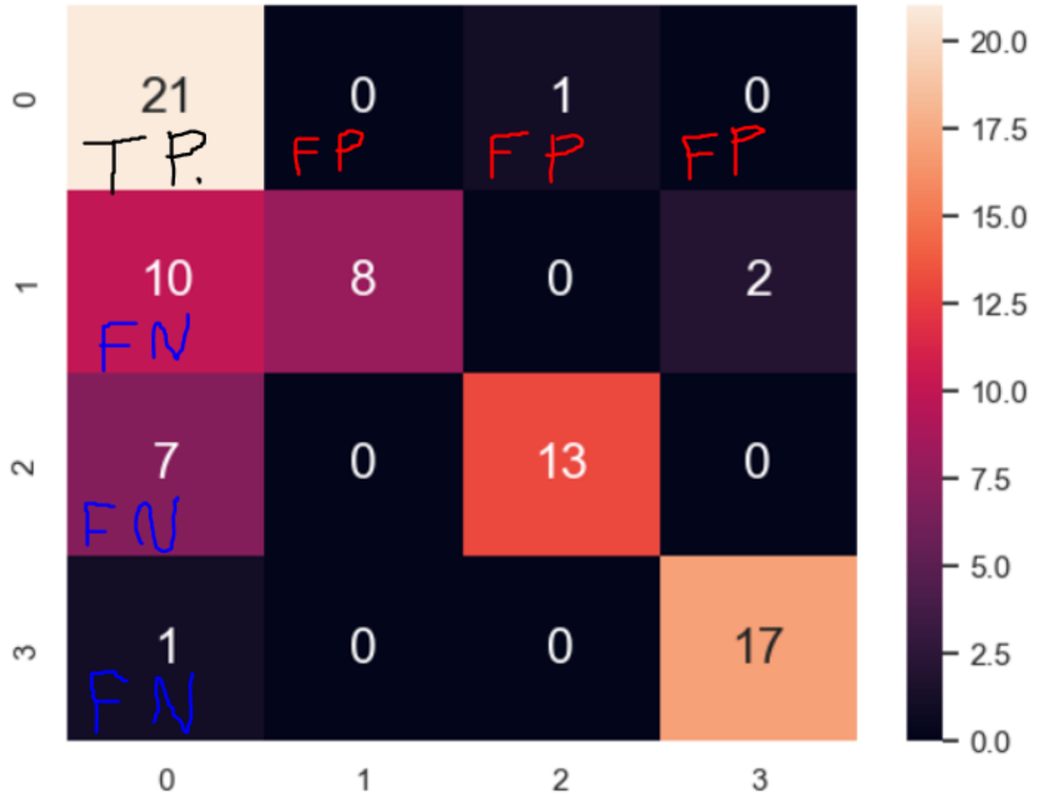
```
# Evaluate the model
#To check for accuracies of model
print("Classification Accuracy for Train \t:", dectree_train.score(Healthylife_train, Happy_train))
```

4 by 4 Confusion Matrics Calculation

```
Classify_train = confusion_matrix(Happy_train,HL_train_pred)
#True Positive Rate
print("Train - True Positive Rate for low:",Classify_train[0][0]/(Classify_train[0][0]+Classify_train[0][1]+Classify_train[0][2]+Classify_train[0][3]))
print("Train - True Positive Rate for Average:",Classify_train[1][1]/(Classify_train[0][1]+Classify_train[1][1]+Classify_train[2][1]+Classify_train[3][1]))
print("Train - True Positive Rate for High:",Classify_train[2][2]/(Classify_train[0][2]+Classify_train[2][1]+Classify_train[2][2]+Classify_train[2][3]))
print("Train - True Positive Rate for Very High:",Classify_train[3][3]/(Classify_train[0][3]+Classify_train[1][3]+Classify_train[2][3]+Classify_train[3][3]))
```



CALCULATION OF 4 X 4 CONFUSION MATRICES



ACCURACY

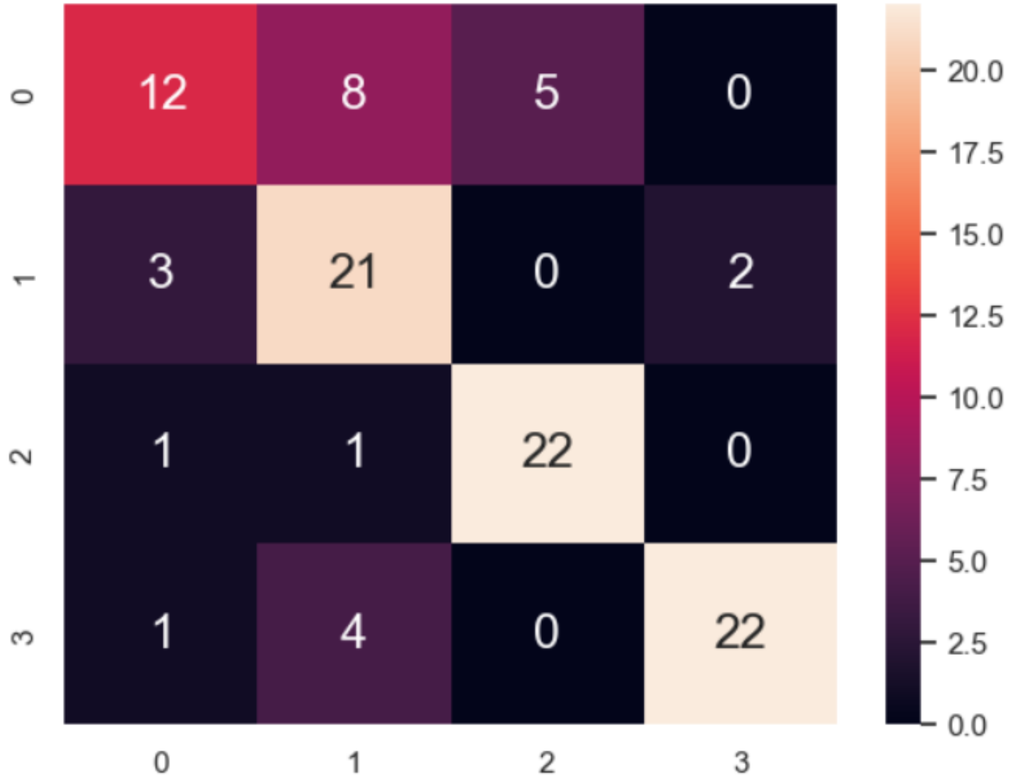
Logged GDP Per Capita:

Classification Accuracy(Train): 0.755

Classification Accuracy(Test): 0.731

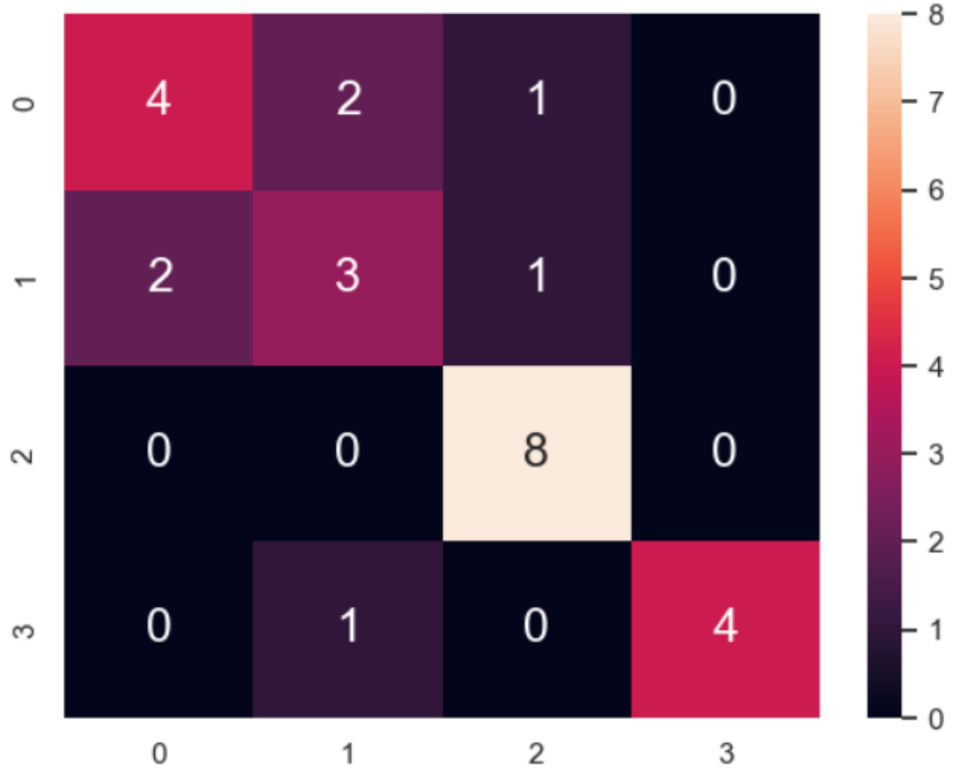
TRAIN DATA

CLASS	TPR (3dp)	FPR (3dp)	TNR (3dp)	FNR (3dp)
0 (low)	0.480	0.152	0.847	0.520
1 (Average)	0.618	0.073	0.926	0.382
2 (High)	0.786	0.026	0.973	0.214
3 (Very High)	0.917	0.064	0.935	0.083



TEST DATA

CLASS	TPR (3dp)	FPR (3dp)	TNR (3dp)	FNR (3dp)
0 (low)	0.571	0.150	0.847	0.425
1 (Average)	0.500	0.150	0.926	0.500
2 (High)	0.888	0.000	1.000	0.111
3 (Very High)	1.000	0.045	0.954	0.000



ACCURACY

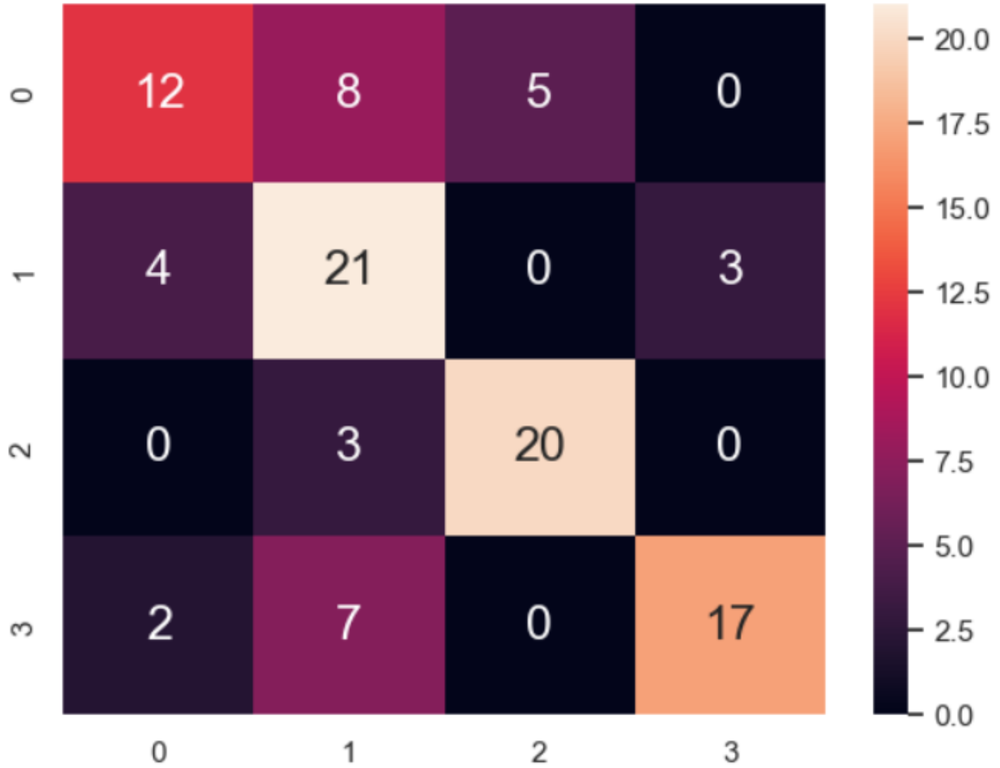
Happiness Score:

Classification Accuracy (Train) : 0.686

Classification Accuracy(Test): 0.615

TRAIN DATA

CLASS	TPR (3dp)	FPR (3dp)	TNR (3dp)	FNR (3dp)
0 (low)	0.480	0.154	0.845	0.520
1 (Average)	0.538	0.111	0.888	0.461
2 (High)	0.714	0.038	0.961	0.285
3 (Very High)	0.850	0.109	0.890	0.150



TEST DATA

CLASS	TPR (3dp)	FPR (3dp)	TNR (3dp)	FNR (3dp)
0 (low)	0.285	0.238	0.761	0.714
1 (Average)	0.500	0.000	1.000	0.500
2 (High)	0.600	0.176	0.823	0.400
3 (Very High)	1.000	0.090	0.909	0.000

ACCURACY

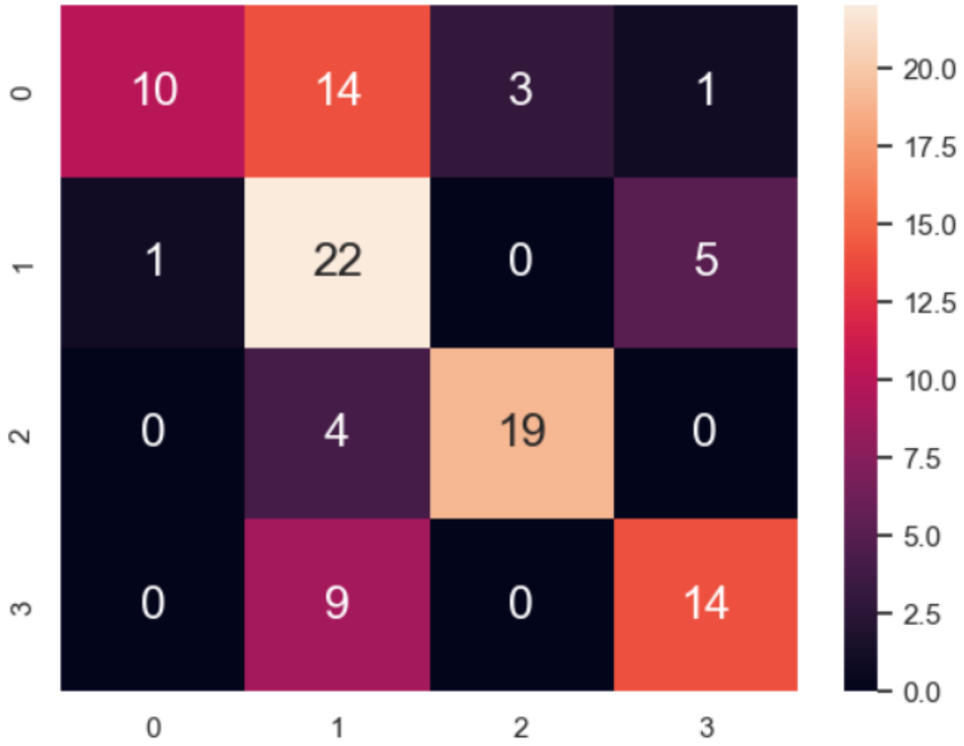
Social Support:

Classification Accuracy (Train) : 0.637

Classification Accuracy(Test): 0.423

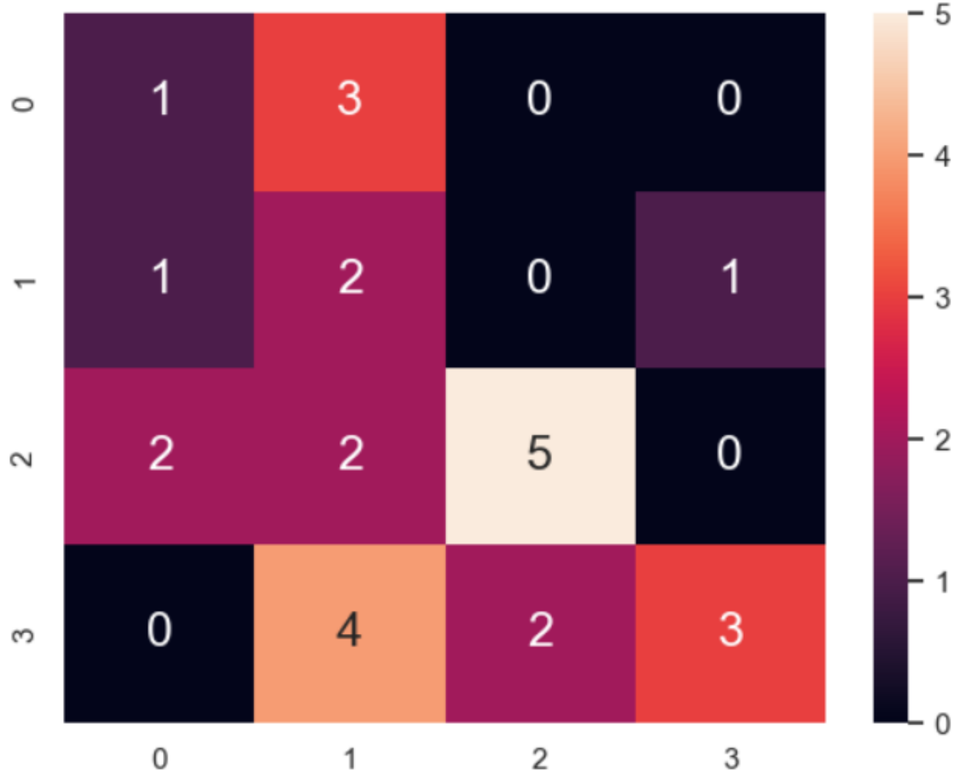
TRAIN DATA

CLASS	TPR (3dp)	FPR (3dp)	TNR (3dp)	FNR (3dp)
0 (low)	0.357	0.197	0.802	0.642
1 (Average)	0.448	0.113	0.886	0.551
2 (High)	0.730	0.050	0.950	0.269
3 (Very High)	0.700	0.109	0.890	0.300



TEST DATA

CLASS	TPR (3dp)	FPR (3dp)	TNR (3dp)	FNR (3dp)
0 (low)	0.250	0.136	0.863	0.750
1 (Average)	0.181	0.133	0.866	0.818
2 (High)	0.714	0.210	0.789	0.285
3 (Very High)	0.750	0.272	0.727	0.250



CLASSIFICATION TREE

GDP has the highest correlation to Healthy Life Expectancy

CONCLUSION

What did we learn from this project

1. Usage of Discretization to convert continuous variable to categorical varible
2. How to Reshape Arrays
3. Calculating of a 4 x 4 confusion matrics

CONCLUSION

Linear Regression Vs Classification Tree

- Both method shows similar results that GDP is the best in predicting life expectancy
- Discretization does not ensure accuracy as its affected by the intervals
- Having too much data near the median value affects Discretization

CONCLUSION

Outcome of Project

- Linear Regression is the best ML method to help us predict what affects life expectancy
- The result from Linear Regression shows GDP is the more important factor in life expectancy
- Prioritizing GDP is very important to live a long life

REFERENCES

- Master of science in data science (MSDS). Corporate NTU. (n.d.). Retrieved April 21, 2023, from https://www.ntu.edu.sg/education/graduate-programme/master-of-science-in-data-science-%28msds%29

- Taulli, T. (2019, August 12). Hiring for the AI (artificial intelligence) revolution - part I. Forbes. Retrieved April 21, 2023, from https://www.forbes.com/sites/tomtaulli/2019/01/26/hiring-for-the-ai-artificial-intelligence-revolution-part-i/?sh=a2c86f751b3e

- lucidv01dlucidv01d 2. (1962, April 1). Scikit-learn: How to obtain true positive, true negative, false positive and false negative. Stack Overflow. Retrieved April 21, 2023, from https://stackoverflow.com/questions/31324218/scikit-learn-how-to-obtain-true-positive-true-negative-false-positive-and-fal