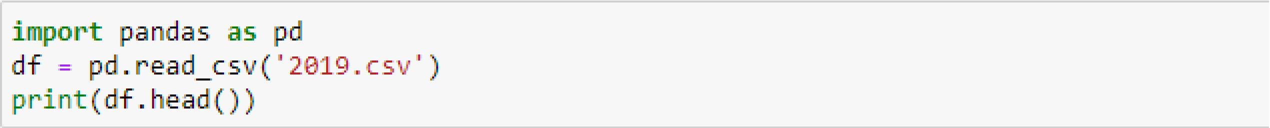
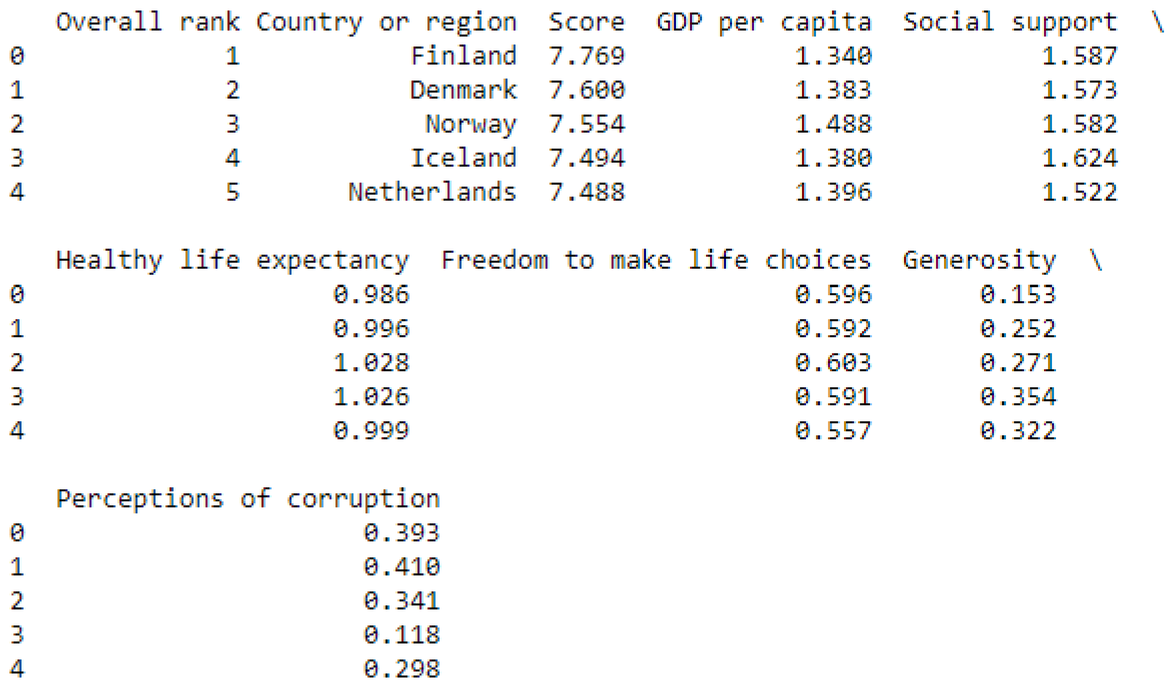
Analyze World Happiness 2019

1. What parameter needs to be a world happines in 2019?

I use world happines report dataset from Kaggle, that can access through this link (https://www.kaggle.com/unsdsn/world-happiness?select=2019.csv). I insert this codes below to see the features and the target.



```python
import pandas as pd
df = pd.read_csv('2019.csv')
print(df.head())
```



```
   Overall rank Country or region  Score  GDP per capita  Social support  \
0             1           Finland  7.769           1.340           1.587   
1             2           Denmark  7.600           1.383           1.573   
2             3            Norway  7.554           1.488           1.582   
3             4           Iceland  7.494           1.380           1.624   
4             5       Netherlands  7.488           1.396           1.522   

   Healthy life expectancy  Freedom to make life choices  Generosity  \
0                    0.986                         0.596       0.153   
1                    0.996                         0.592       0.252   
2                    1.028                         0.603       0.271   
3                    1.026                         0.591       0.354   
4                    0.999                         0.557       0.322   

   Perceptions of corruption  
0                      0.393  
1                      0.410  
2                      0.341  
3                      0.118  
4                      0.298  
```

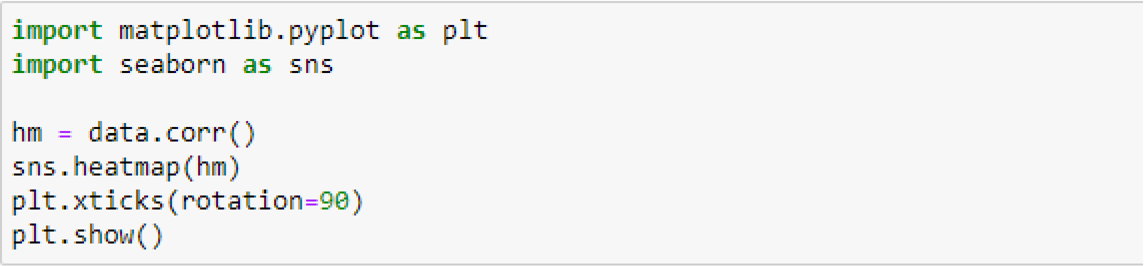
The result show that the dataset has 9 columns (Overall rank, country or region, score, GDP per capita, social suppport, healthy life expectancy, freedom to make life choices, generosity, and perception of corruption).

At this analyze, i want to know what are the main parameter for scoring the rank. To answer this question, i would like to answer it at next step.

2. What does score means related to Overall rank?

We know that the score are the final result of the features. The features at this dataset would be GDP per capita, social support, health life expectancy, freedom to make life choices, generosity, and perception of corruption.

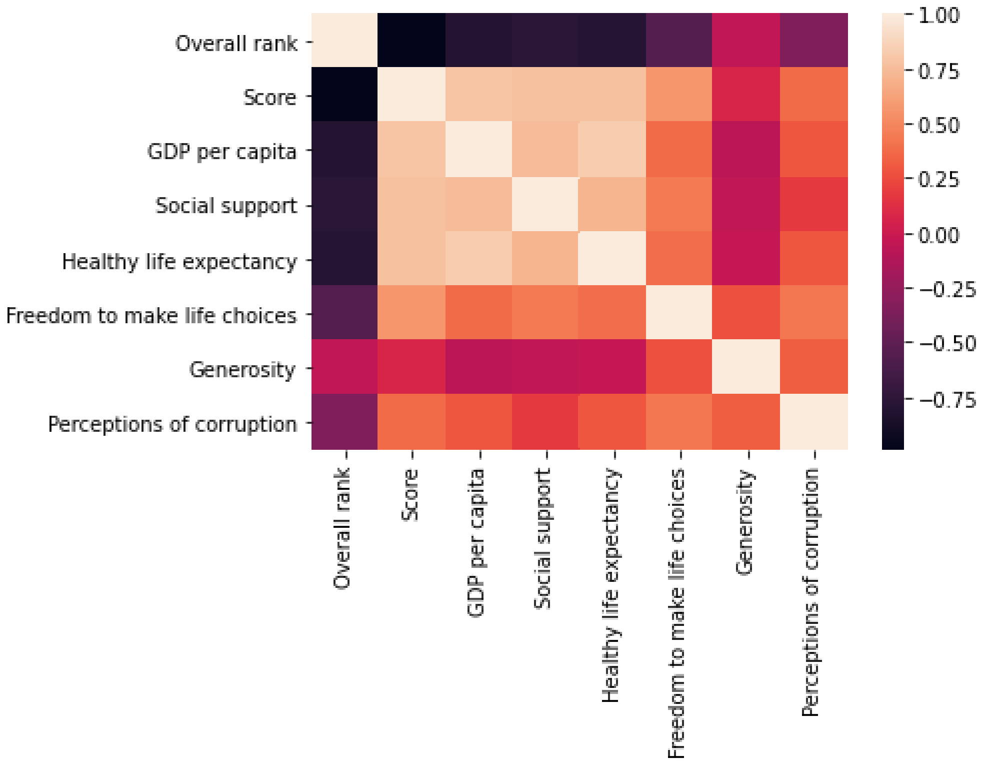
But what are the most affect of the features related to score? I use correlation between them to know the close value of this scoring. Here the code shown below



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

hm = data.corr()
sns.heatmap(hm)
plt.xticks(rotation=90)
plt.show()
```

The result of these codes above can see at this graph heatmap.



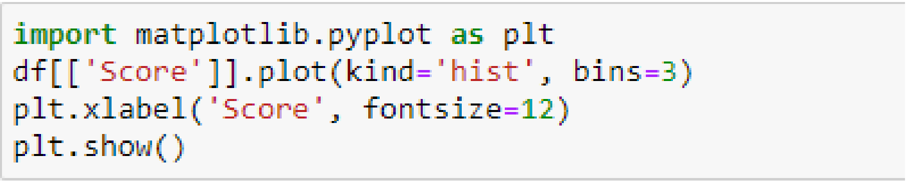
The score at these heatmap mostly are related to GDP per capita, social support, freedom to make life choice. And fewer are related to generosity and perception of corruption.

According to this insight that the country could say the country happines if the GDP per capita, social support, and healthy life are the high value. But we have to remember strightly this dataset are quantity and quality obejctive component.

As an objective analyze, i have to decide the quality of the country by classifying that the country is good, intermediate or poor?. To classify this three classes, i decide to see the score. The highest score it seen, the more top world happines it be. This objective analyze will shown at next step.

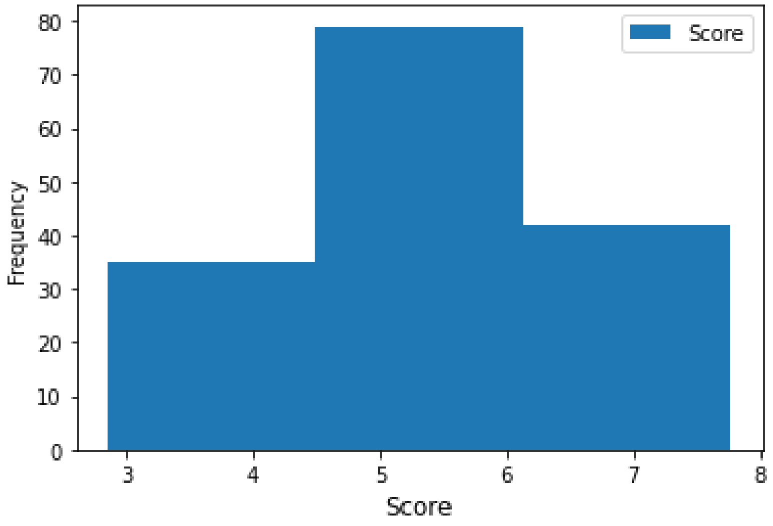
3. What the best method to predict the country? It is good country, poor country, or intermediate?

By knowing first the score that is related to overal rank. I classify this score into three classes. To classify this score, i use histogram to plot the data distribution with value bins is three. The code will shown below:



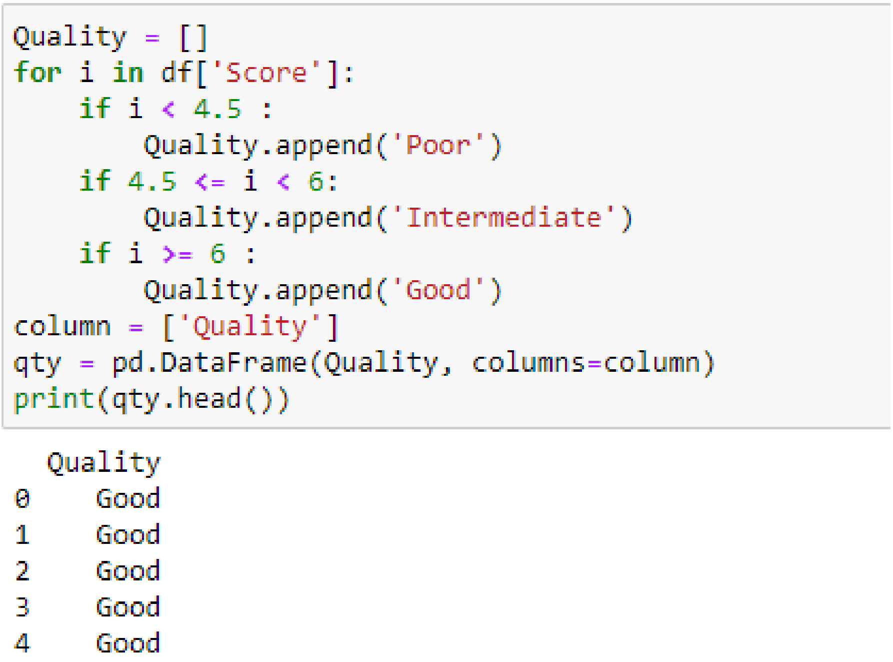
```
import matplotlib.pyplot as plt
df[['Score']].plot(kind='hist', bins=3)
plt.xlabel('Score', fontsize=12)
plt.show()
```

The result of this histogram could be see below:



This histogram tell that the score less than around 4.5 could classify as poor country, score between 4.5 and 6.0 could classify as intermediate country, and score more than 6.0 could classify as good country.

According this new data. I would to make new target as quality component using conditional statement.

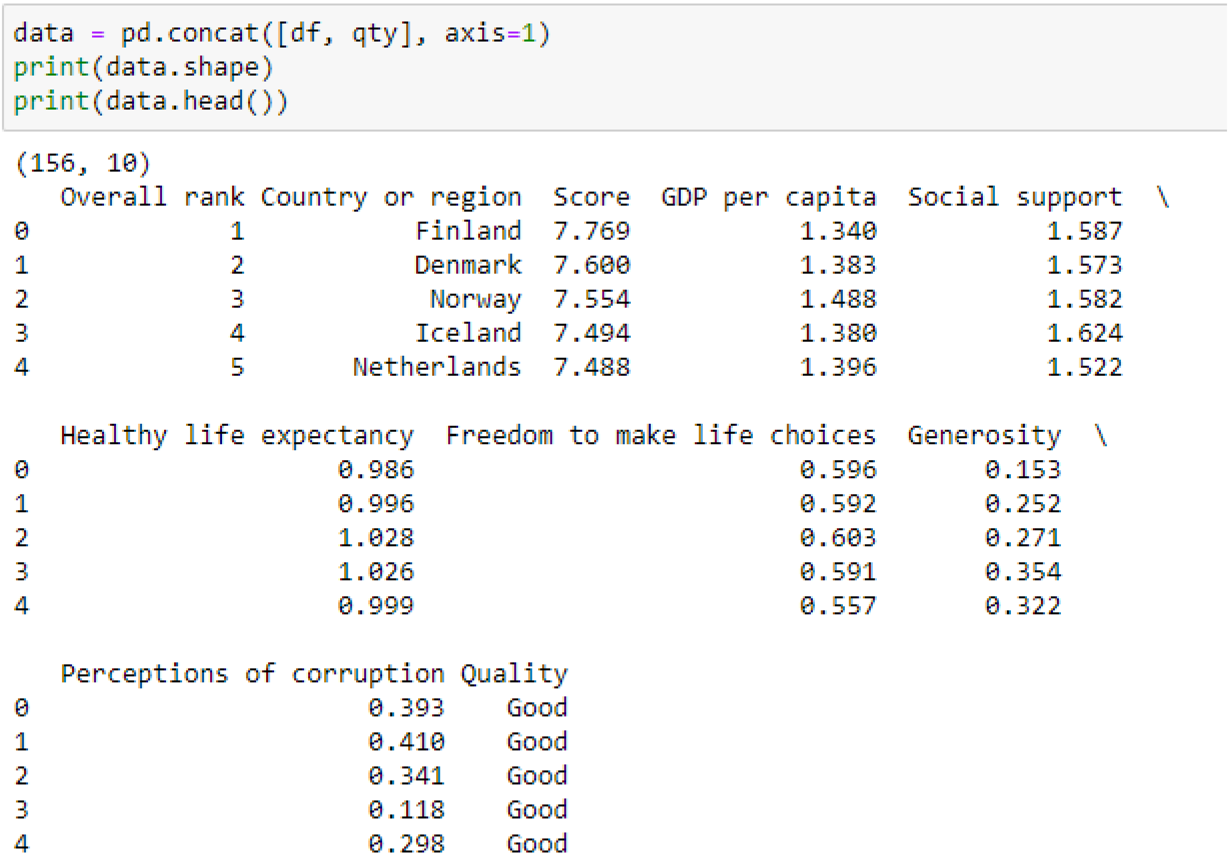


```
Quality = []
for i in df['Score']:
    if i < 4.5 :
        Quality.append('Poor')
    if 4.5 <= i < 6:
        Quality.append('Intermediate')
    if i >= 6 :
        Quality.append('Good')
column = ['Quality']
qty = pd.DataFrame(Quality, columns=column)
print(qty.head())
```

```
  Quality
0    Good
1    Good
2    Good
3    Good
4    Good
```

After i have classified three classes. I make it to new dataframe as target ‘Quality’ name for analysis purpose in machine learning. The code show that dataframe have built success and show the five rows head are good country.

To merge this two dataset between world happines and quality of country. The function of concat could be used at this analyze shown below with the result.



```
data = pd.concat([df, qty], axis=1)
print(data.shape)
print(data.head())
```

```
(156, 10)
   Overall rank Country or region  Score  GDP per capita  Social support  \
0             1           Finland  7.769           1.340           1.587   
1             2           Denmark  7.600           1.383           1.573   
2             3            Norway  7.554           1.488           1.582   
3             4           Iceland  7.494           1.380           1.624   
4             5       Netherlands  7.488           1.396           1.522   

   Healthy life expectancy  Freedom to make life choices  Generosity  \
0                    0.986                         0.596       0.153   
1                    0.996                         0.592       0.252   
2                    1.028                         0.603       0.271   
3                    1.026                         0.591       0.354   
4                    0.999                         0.557       0.322   

   Perceptions of corruption Quality  
0                      0.393    Good  
1                      0.410    Good  
2                      0.341    Good  
3                      0.118    Good  
4                      0.298    Good  
```

We can see that the column added to the dataset. At the next step i will analyze using machine learning to classify the dataset and comparing to mind classification.

4. What the best method of machine learning use for classify?

Machine learning is one solution usually used in data science. At this case, the dataset of world happines could be a supervised learning – classification, and regression.

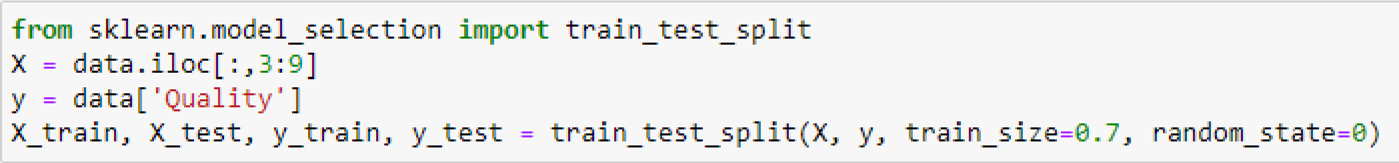
At supervised learning – classification, we can classify this country and predict it to good, intermediate, or poor. Meanwhile supervised learning – regression, we can predict and estimate score by new data features.

There are four mehtods i use at this supervised learning.

1. Logistic regression: it could tell does the country is good or intermediate or poor?
2. Decision Tree: it could tell where the country path to be by tree graph
3. Random forest : as same as decision tree, but it more complicated
4. Gradient boosting: it could tell the country by value and gradient statistically.

Note : i had classified the country before by using histogram, the result might not be high precicy and high accuracy.

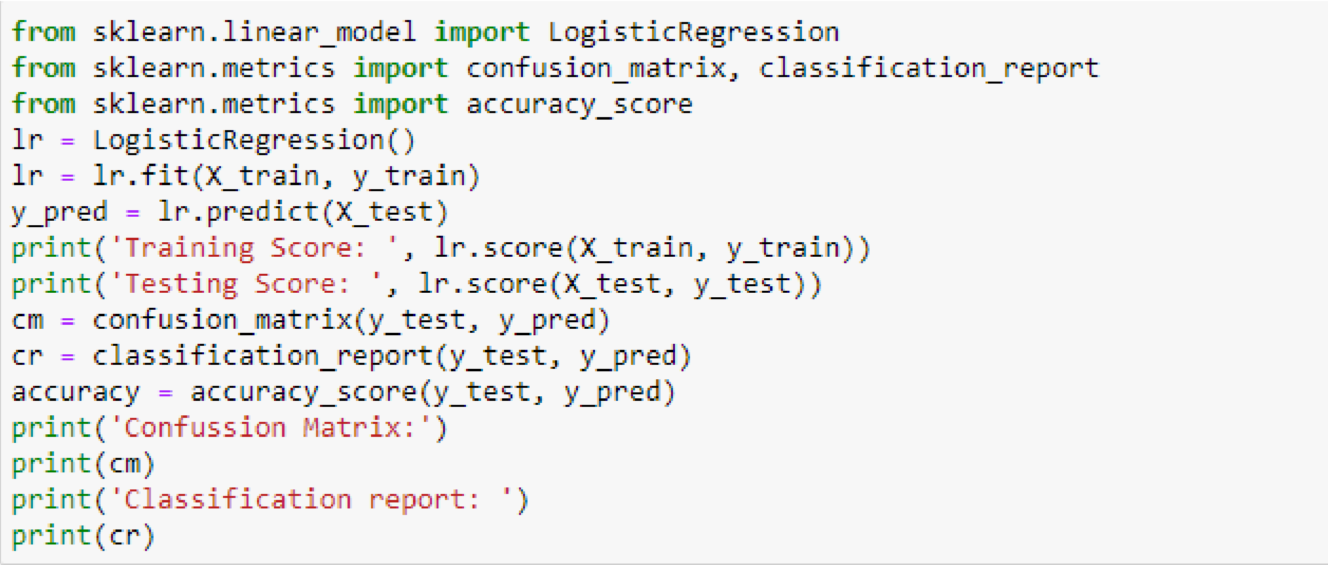
Before fitting to machine learning, we have to split the data into training and testing. At this case i use 70% size of training data and 30% size of testing data, because this dataset is no more than 200 rows. At this process we can use train test split module from skit-learn packages.



```
from sklearn.model_selection import train_test_split
X = data.iloc[:,3:9]
y = data['Quality']
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=0)
```

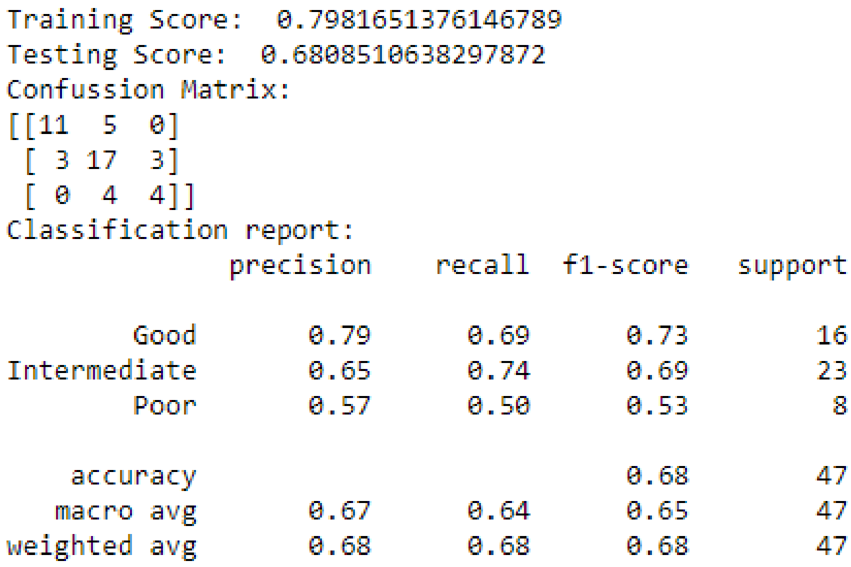
The code for training and testing data has successed.

After this step, train and test. I have to fit machine learing into this new data. I have tried four methods above and the best method for this machine learning is logistic regression. Because this logistic are simple categorization by good or intermediate or poor . The code for this machine learning – supervised shown below;



```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
lr = LogisticRegression()
lr = lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
print('Training Score: ', lr.score(X_train, y_train))
print('Testing Score: ', lr.score(X_test, y_test))
cm = confusion_matrix(y_test, y_pred)
cr = classification_report(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
print('Confussion Matrix:')
print(cm)
print('Classification report: ')
print(cr)
```

The result of this code are shown below;



```
Training Score:  0.7981651376146789
Testing Score:  0.6808510638297872
Confussion Matrix:
[[11  5  0]
 [ 3 17  3]
 [ 0  4  4]]
Classification report:
              precision    recall  f1-score   support

        Good       0.79      0.69      0.73        16
Intermediate       0.65      0.74      0.69        23
        Poor       0.57      0.50      0.53         8

    accuracy                           0.68        47
   macro avg       0.67      0.64      0.65        47
weighted avg       0.68      0.68      0.68        47
```

The result of this method closer to dataset than othes methods. Training score for this method is 79% and Testing score is 68%. It could be prevent of overfitting. The confussion matrix can be seen that accuray of 47 rows testing data, there are 15 mistake classification. The accuracy of this method is 68%. This value is bad, but important!!!. This classify is obejctive component that might be wrong because the quality of the country is quality component.

I would to share the accuray of others methods use in this dataset to compare.

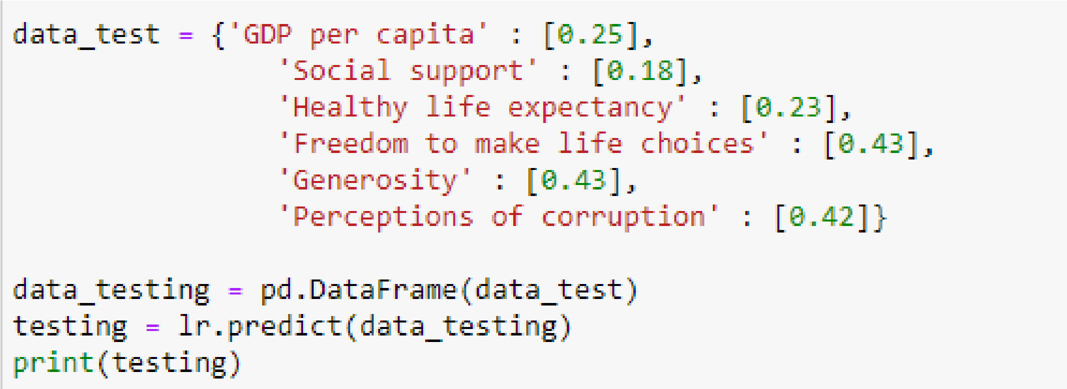
1. Logistic regression:
   Training score: 79%
   Test score: 68%
2. Decision tree:
   Training score : 100%
   Test score : 63%
3. Random forest :
   Training score: 100%
   Test score : 65%
4. Gradient Boosting:
   Training score : 100%
   Test score : 65%

At this value above, decision tree, random forest, and gradient boosting has 100% score at training score. This score indicate the model is overfitting. So the best model could be logistic regresion with accuracy is 68%.

The score could be bad accuracy because there are some factors include, the data, value, and objective component at this dataset.

5. If we have a new data unknown what is the country, can we classify it? And what classification could be ?

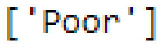
Absolutelly yes, we can predict the new data and we can classify. This is example of new data at unknown country.



```
data_test = {'GDP per capita' : [0.25],
                'Social support' : [0.18],
                'Healthy life expectancy' : [0.23],
                'Freedom to make life choices' : [0.43],
                'Generosity' : [0.43],
                'Perceptions of corruption' : [0.42]}

data_testing = pd.DataFrame(data_test)
testing = lr.predict(data_testing)
print(testing)
```

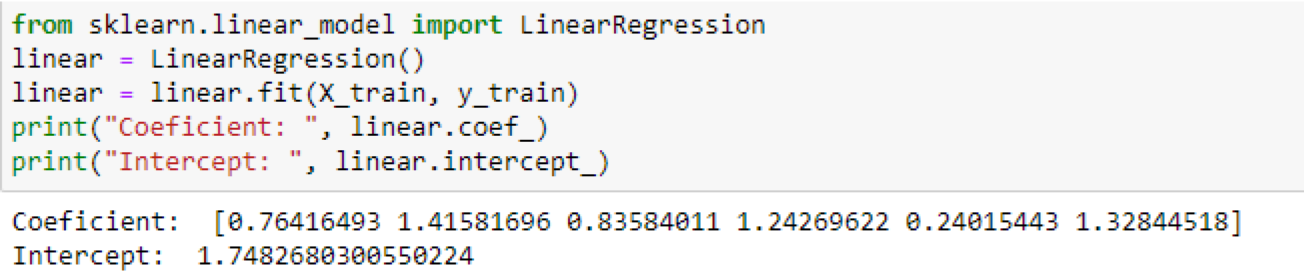
The result of this data of unknown country could classify as :



```
['Poor']
```

6. Can we estimate the score of new this data of unknown country?

Yes, before we estimate the score. We have to make regression model first. At this dataset, we have some features and the target is score. So, the regression could be multiple linear regression. Here the code shown for multiple linear regression model at supervised learning – regression.



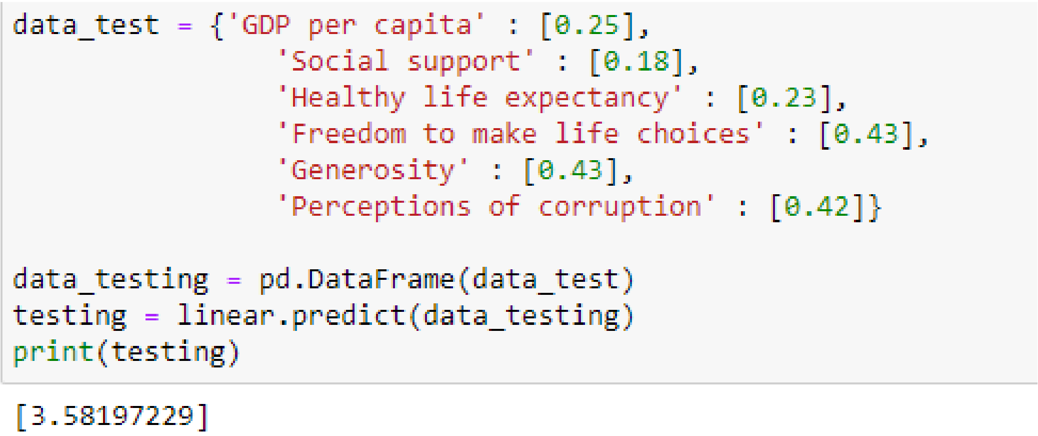
```
from sklearn.linear_model import LinearRegression
linear = LinearRegression()
linear = linear.fit(X_train, y_train)
print("Coeficient: ", linear.coef_)
print("Intercept: ", linear.intercept_)
```

```
Coeficient:  [0.76416493 1.41581696 0.83584011 1.24269622 0.24015443 1.32844518]
Intercept:  1.7482680300550224
```

According the resut, shown that the coeficient of each independent features like GDP per capita (*x1*), social support (*x2*), healthy life expectancy (*x3*), freedom to make life choices (*x4*), generosity (*x5*) and perception of corruption (*x6*) . Thus, this is the formula for this model.

$$Y = 0.764x1 + 1.416x2 + 0.836x3 + 1.242x4 + 0.240x5 + 1.328x6 + 1.748$$

After we have this formula, we can predict new data by filling the independent featurs. Here the example data to estimate the value of score.

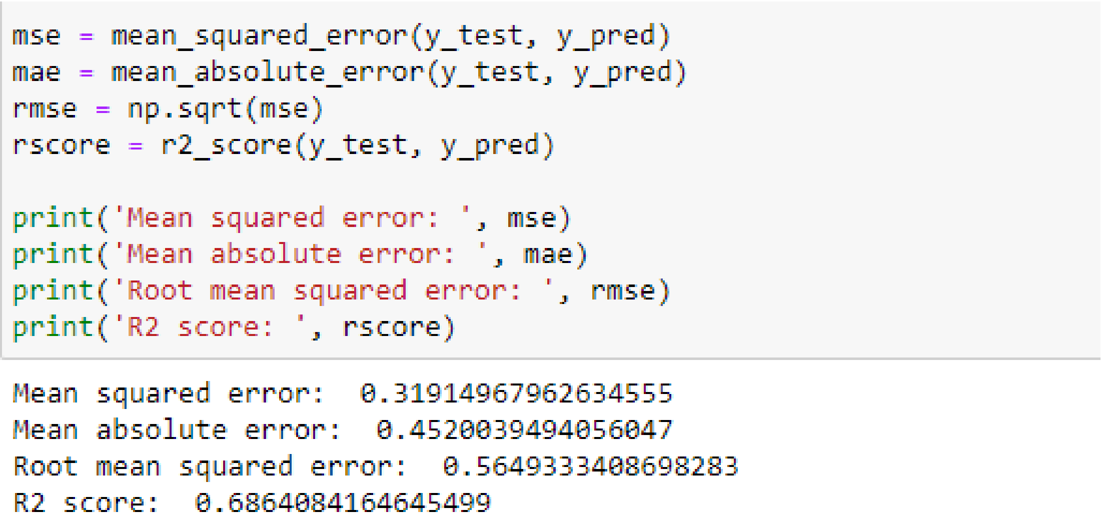


```
data_test = {'GDP per capita' : [0.25],
                'Social support' : [0.18],
                'Healthy life expectancy' : [0.23],
                'Freedom to make life choices' : [0.43],
                'Generosity' : [0.43],
                'Perceptions of corruption' : [0.42]}

data_testing = pd.DataFrame(data_test)
testing = linear.predict(data_testing)
print(testing)
```

```
[3.58197229]
```

The result shown that the value of score is 3.58. The value is in poor classification as we had predict it by logistic regression. The error can predict by skit-learn metric that the code shown below;



```
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mse)
rscore = r2_score(y_test, y_pred)

print('Mean squared error: ', mse)
print('Mean absolute error: ', mae)
print('Root mean squared error: ', rmse)
print('R2 score: ', rscore)
```

```
Mean squared error:  0.31914967962634555
Mean absolute error:  0.4520039494056047
Root mean squared error:  0.5649333408698283
R2 score:  0.6864084164645499
```

The accuracy at this linear regression method as same as the logistic regression is around in 68%.

The conclusion at new data is has score 3.58 with poor classification.