

# *A COMPLETE ANALYSIS OF WORLD HAPPINESS DATASET*

BY-

ABHINAV KUMAR

## *Brief Overview of World Happiness Dataset with Linear Regression*

**What is Happiness index??** And how it is calculated.....

CONTEXT:-

The World Happiness Report is a landmark survey of the state of global happiness, it is the annual publication of the United Nation's sustainable development network solution. The first report was published in 2012, the second in 2013, the third in 2015, and the fourth in the 2016. The World Happiness 2017, which ranks 158 countries by their happiness levels, was released at the United Nations at an event celebrating International Day of Happiness on March 20th. The report continues to gain global recognition as governments, organizations and civil society increasingly use happiness indicators to inform their policy-making decisions. Leading experts across fields – economics, psychology, survey analysis, national statistics, health, public policy and more – describe how measurements of well-being can be used effectively to assess the progress of nations. The reports review the state of happiness in the world today and show how the new science of happiness explains personal and national variations in happiness.

What I believe, the purpose of life is to be happy, what all are the factors that let the people of the country to be happy?? According to our Dataset Switzerland is the happiest country in the world, then the question arises in mind that why Switzerland is the happiest country in the world....and the answer is that On a societal level, **Switzerland** success can be attributed to its rigid social safety network, culture of trust, high-quality education, and a strong commitment to gender equality. On a personal level, many People of the country cite their connection to nature as an important source of happiness.

The rankings of national happiness are based on a Cantril ladder survey. Nationally representative samples of respondents are asked to think of a ladder, the

best possible life for them being a 10, and the worst possible experience is a 0. They are then asked to rate their own current lives on 0 to 10 scale. The report correlates the results with various life factors. In the reports, experts in economics, psychology, survey analysis, and national statistics describe how well-being measurements can be used effectively to assess nations' progress and other topics.

### FEATURES ANALYZED:-

The following features describe the extent to which these factors contribute in evaluating the happiness in each country:-

- **HAPPINESS RANK-** It is the order in which country is arranged according to the rank given by Cantril ladder survey.
- **HAPPINESS SCORE-** A metric measured by asking the sampled people the question: "How would you rate your happiness on an average out of 10.
- **ECONOMY-** It is the GDP per capita is a measure of a country's economic output that accounts for its number of people.
- **HEALTH-** Healthy Life Expectancy is the average number of years that a newborn can expect to live in "full health" — in other words, not hampered by disabling illnesses or injuries.
- **FREEDOM-** It is the Freedom of choice describes an individual's opportunity and autonomy to perform an action selected from at least two available options, unconstrained by external parties.
- **GOVERNMENT CORRUPTION-** It is defined as the corruption in their government within the country. The Corruption Perceptions Index (CPI) is an index published annually by Transparency International since 1995, which ranks countries "by their perceived levels of public sector corruption, as determined by expert assessments and opinion surveys."
- **GENEROSITY-** It is defined as the residual of regressing the national average of responses to the question, "Have you donated money to a charity in past months?" on GDP capita.

### **WHAT IS DYSTOPIA?**

Dystopia is an imaginary country that has the world's least-happy people. The purpose in establishing Dystopia is to have a benchmark against which all countries can be favorably compared (no country performs more poorly than Dystopia) in terms of each of the six key variables, thus allowing each sub-bar to be of positive width. The lowest scores observed for the six key variables,

therefore, characterize Dystopia. Since life would be very unpleasant in a country with the world's lowest incomes, lowest life expectancy, lowest generosity, most corruption, least freedom and least social support, it is referred to as “Dystopia,” in contrast to Utopia.

So the above are some of the features that are analyzed in our dataset. Going further will be predicting the target by building a strong machine learning based model to calculate the happiness score which we will be treating as our target variable, from the other features which are the independent variables.

Here we will be using a Linear Regression algorithm, since the target variable is in continuous form. So we will start with importing all the useful libraries like numerical python and pandas to read the csv file, then a Linear Regression model from model selection library and other useful metrics to calculate the error and r2 score which we will discuss further. A snip of the dataset is shown below.

```
7]: import warnings
    warnings.filterwarnings('ignore')
    import pandas as pd
    import numpy as np

3]: from sklearn.linear_model import LinearRegression
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

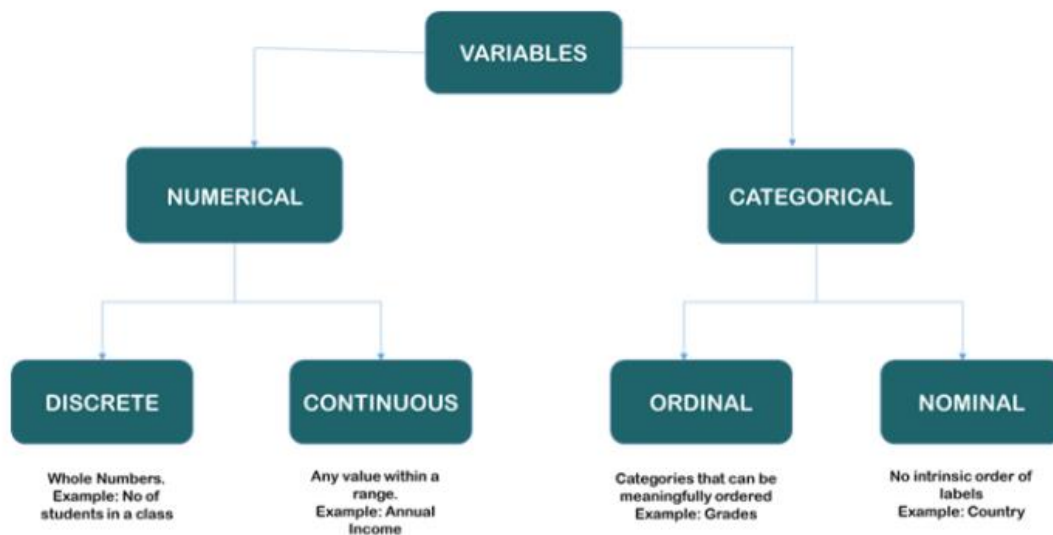
3]: #importing the dataset

3]: df1=pd.read_csv('happiness.csv')

1]: df1.head()
```

|   | Country     | Region         | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family  | Health (Life Expectancy) | Freedom | Trust (Government Corruption) | Generosity | Dystopia Residual |
|---|-------------|----------------|----------------|-----------------|----------------|--------------------------|---------|--------------------------|---------|-------------------------------|------------|-------------------|
| 0 | Switzerland | Western Europe | 1              | 7.587           | 0.03411        | 1.39651                  | 1.34951 | 0.94143                  | 0.66557 | 0.41978                       | 0.29678    | 2.51738           |
| 1 | Iceland     | Western Europe | 2              | 7.561           | 0.04884        | 1.30232                  | 1.40223 | 0.94784                  | 0.62877 | 0.14145                       | 0.43630    | 2.70201           |
| 2 | Denmark     | Western Europe | 3              | 7.527           | 0.03328        | 1.32548                  | 1.36058 | 0.87464                  | 0.64938 | 0.48357                       | 0.34139    | 2.49204           |

As we have read the dataset, using pandas we will further perform the **EXPLORATORY DATA ANALYSIS** which will help us in understanding the dataset. The very first step in exploratory data analysis is to identify the type of variables in the dataset. Variables are of two types — Numerical and Categorical. They can be further classified as follows:



Classification of Variables

Now we will check the shape of the data set which gives us the number of rows and columns using `df.shape` and Data types of the corresponding column using `df.info()` as given below.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Country                              158 non-null    object
1   Region                               158 non-null    object
2   Happiness Rank                       158 non-null    int64
3   Happiness Score                      158 non-null    float64
4   Standard Error                      158 non-null    float64
5   Economy (GDP per Capita)            158 non-null    float64
6   Family                               158 non-null    float64
7   Health (Life Expectancy)            158 non-null    float64
8   Freedom                             158 non-null    float64
9   Trust (Government Corruption)        158 non-null    float64
10  Generosity                          158 non-null    float64
11  Dystopia Residual                    158 non-null    float64
dtypes: float64(9), int64(1), object(2)
memory usage: 14.9+ KB
```

So what we infer from the above is,

- The dataset does not contain any null values.
- Different column and its corresponding data types.
- Memory consumed by the above.

We can perform some more data analysis like checking the unique values in a particular column or checking the value counts, etc.

Next step is to convert the object data type to int or float data type which can be done with the help of label encoder or one hot encoder, as we can see from the above that 'Country' and 'Region' column is having object data type so converting them to int dtype can be useful for us while doing data visualization which we will take in the later part.

```
: from sklearn.preprocessing import LabelEncoder
: le=LabelEncoder()
: df['Country']=le.fit_transform(df['Country'])
: df["Region"]=le.fit_transform(df['Region'])
: df.head()
```

|   | Country | Region | Happiness Rank | Happiness Score | Standard Error |
|---|---------|--------|----------------|-----------------|----------------|
| 0 | 135     | 9      | 1              | 7.587           | -3.37816       |
| 1 | 58      | 9      | 2              | 7.561           | -3.01920       |
| 2 | 37      | 9      | 3              | 7.527           | -3.40279       |
| 3 | 105     | 9      | 4              | 7.522           | -3.24933       |
| 4 | 24      | 5      | 5              | 7.427           | -3.33737       |

df.describe() will show us descriptive statistics like mean, standard deviation and min/max values as shown below.

```
df.describe()
```

|       | Happiness Rank | Happiness Score | Standard Error | Economy (GDP per Capita) | Family     | Health (Life Expectancy) | Freedom    | Trust (Gove Corruption) |
|-------|----------------|-----------------|----------------|--------------------------|------------|--------------------------|------------|-------------------------|
| count | 158.000000     | 158.000000      | 158.000000     | 158.000000               | 158.000000 | 158.000000               | 158.000000 | 158.000000              |
| mean  | 79.493671      | 5.375734        | 0.047885       | 0.846137                 | 0.991046   | 0.630259                 | 0.428615   | 0.143422                |
| std   | 45.754363      | 1.145010        | 0.017146       | 0.403121                 | 0.272369   | 0.247078                 | 0.150693   | 0.120034                |
| min   | 1.000000       | 2.839000        | 0.018480       | 0.000000                 | 0.000000   | 0.000000                 | 0.000000   | 0.000000                |
| 25%   | 40.250000      | 4.526000        | 0.037268       | 0.545808                 | 0.856823   | 0.439185                 | 0.328330   | 0.061675                |
| 50%   | 79.500000      | 5.232500        | 0.043940       | 0.910245                 | 1.029510   | 0.696705                 | 0.435515   | 0.107220                |
| 75%   | 118.750000     | 6.243750        | 0.052300       | 1.158448                 | 1.214405   | 0.811013                 | 0.549092   | 0.180255                |
| max   | 158.000000     | 7.587000        | 0.136930       | 1.690420                 | 1.402230   | 1.025250                 | 0.669730   | 0.551910                |

## EDA CONCLUDING REMARKS:-

So what we can conclude from the above is that some of the columns are less prone to outliers as we can see from the above that none of the max value in the column is far away from the third quartile value i.e. 75% of the values which we will further analyze when we plot the boxplot and the second thing, that we can conclude from the above dataset is that columns are not skewed, and the skewness is measured from mean and median, mean and the 50 percent value is almost same for most of the columns which will be better analyzed once we plot the histograms for the columns.

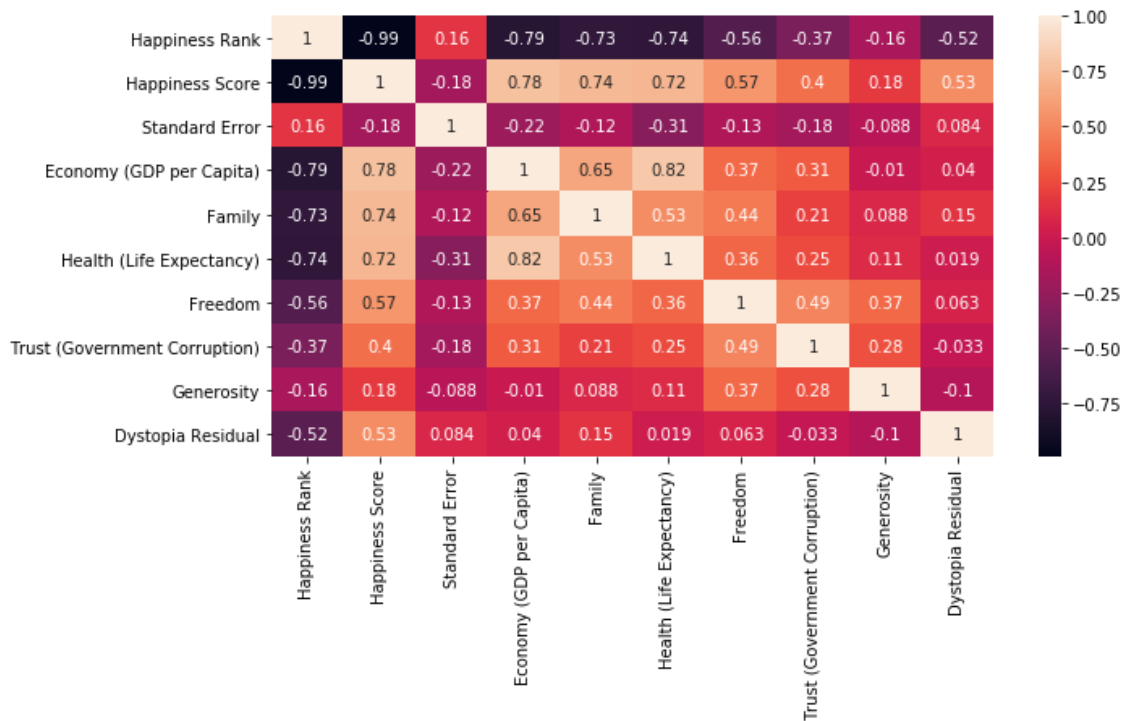
## DATA VISUALIZATION:-

Now we will be proceeding towards the data visualization part.

Below is the heatmap of the dataset which shows the correlation between different columns, as we can see that the darker part is showing the negatively correlated values and the lighter part in the dataset shows that columns are positively correlated amongst each other.

```
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True)
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x80f8b2dc48>



We can check the skewness in the columns using `df.skew()` function and further we will remove the skewness using `boxcox` from scientific python library or using a `log` function as shown below:-

```

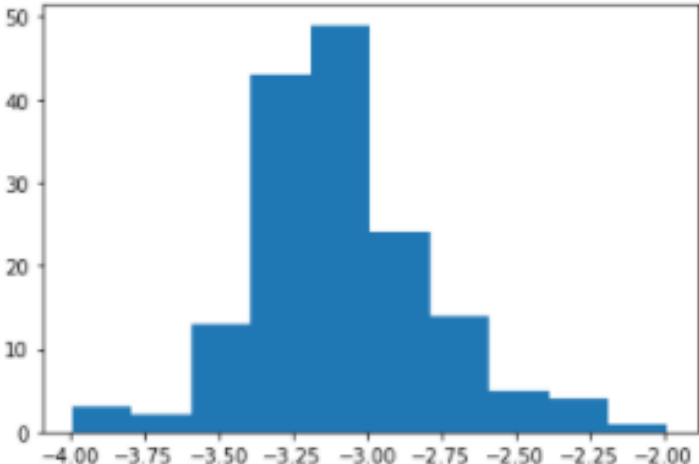
: df.skew()
: Happiness Rank          0.000418
  Happiness Score         0.097769
  Standard Error          1.983439
  Economy (GDP per Capita) -0.317575
  Family                  -1.006893
  Health (Life Expectancy) -0.705328
  Freedom                 -0.413462
  Trust (Government Corruption) 1.385463
  Generosity              1.001961
  Dystopia Residual        -0.238911
  dtype: float64

: #removing skewness of a particular column
  from scipy.stats import boxcox

: df["Standard Error"]=boxcox(df['Standard Error'],0)

: plt.hist(df["Standard Error"])
: (array([ 3.,  2., 13., 43., 49., 24., 14.,  5.,  4.,  1.]),
  array([-3.99106621, -3.79078813, -3.59051006, -3.39023198, -3.1899539 ,
        -2.98967582, -2.78939774, -2.58911967, -2.38884159, -2.18856351,
        -1.98828543]),
  <a list of 10 Patch objects>)

```



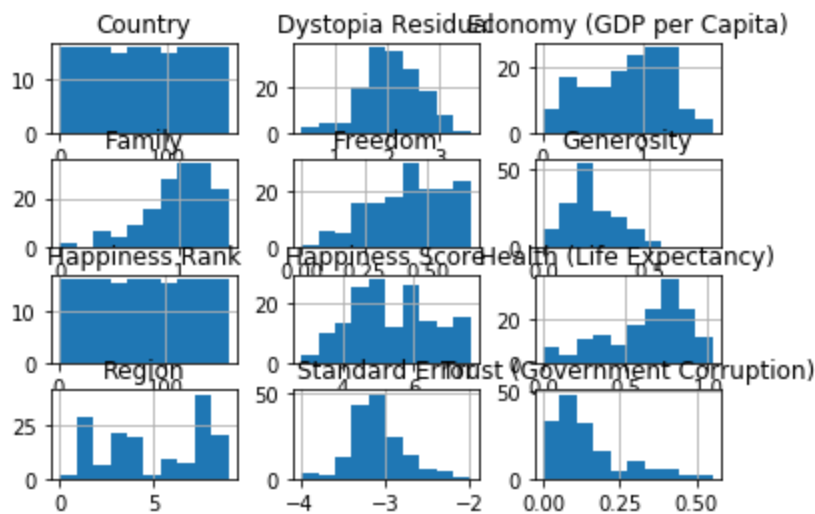
The histogram shows the frequency of 'Standard Error' values. The x-axis represents the 'Standard Error' values, ranging from -4.00 to -2.00. The y-axis represents the frequency, ranging from 0 to 50. The distribution is roughly bell-shaped, centered around -3.00, indicating that the data follows a Gaussian curve after removing skewness.

We can see from the above that the column follows a Gaussian curve after removing the skewness.

Let us have a look at the histogram plotted for different columns. Data is finely spread and also the skewness is low.



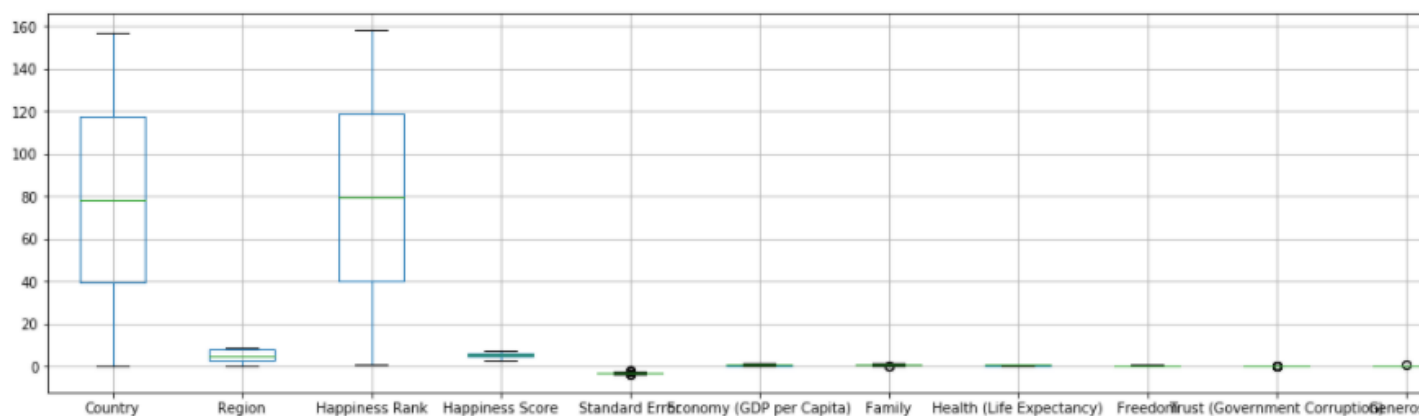
```
df.hist()
plt.show()
```



Below is the boxplot plotted for each column, from that we can see that a very few outliers is present in the dataset which we will remove using scientific python library. For removing the outliers we will keep threshold as 3 and remove all the values which has z score greater than 3 as shown below.

```
df.boxplot(figsize=(20,5))
plt.figure()
```

<Figure size 432x288 with 0 Axes>



```
df_new=df[(z<3).all(axis=1)]
```

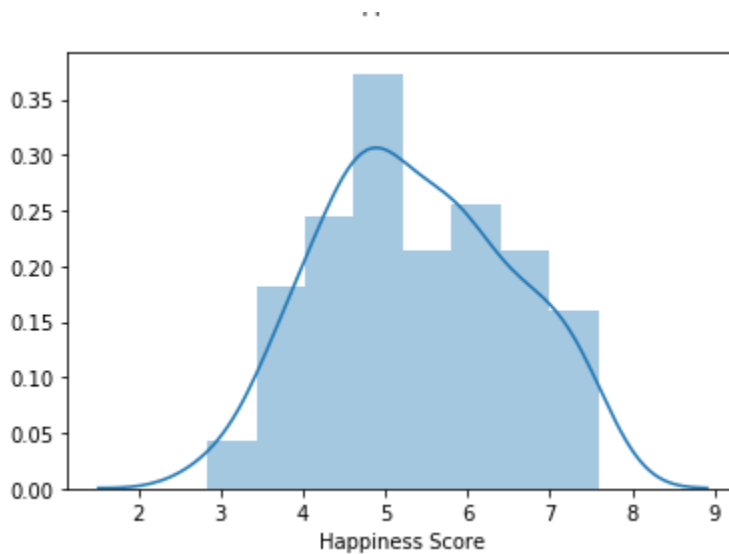
```
df.shape      #with outliers
```

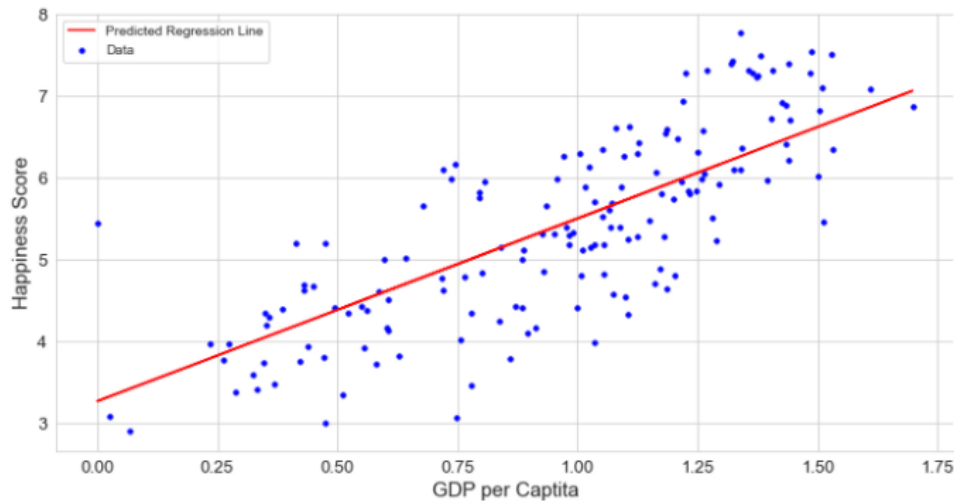
```
(158, 12)
```

```
df_new.shape   #without outliers
```

```
(151, 12)
```

Below is the distribution plot for the Happiness score. As we can see below, the Happiness Score has values above 2.85 and below 7.76. So there is no single country which has a Happiness Score above 8.





The above shows the relationship between the GDP per capita and Happiness Score, from the above we can infer that both are having a high positive correlation i.e. for a country with high GDP the happiness score will also be good.

## BUILDING A LINEAR REGRESSOR MODEL:-

Now we will be building a Linear regression model by importing Linear Regression from linear model library of scikitlearn, but before that we will be splitting the dataset in 'x' and 'y' and as we know we have happiness score as our target variable i.e. 'y' so we will separate it from the dataset then we will be Splitting the dataset in x\_train, x\_test, y\_train, y\_test using train test split method from model selection library of scikitlearn as shown below.

```
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=42)
lin=LinearRegression()
```

Here we are defining test size equal to 33 percent of the dataset, in layman terms we can say that 33 percent of the data will undergo testing phase which is the standard value and rest 67 percent of the data will go under training as this is the supervised learning so our model will first learn with some amount of data. Here we will be checking for the random state in the range of 42 to 100. We will be predicting the result with our x\_test data and then the result is compared with the original y\_test data to check the accuracy of the model and this accuracy is calculated with the help of the metric called as R2 score which carries two parameter i.e. predicted value and original value as said earlier. The same has been illustrated below:-

```
lr=LinearRegression()
lr.fit(x_train,y_train)
pred=lr.predict(x_test)
r_score=r2_score(pred,y_test)
```

And finally we are getting a maximum r2 score of 0.999 i.e. 99 percent at a random state of 54 with test size of 33 percent.

```
the max r2 score for the final random state: 54 is: 0.9999999556367143
```

```
: lr.coef_
: array([[ -1.14942768e-06,  -1.17836299e-05,  -7.50214576e-06,
          -3.85935431e-05,   9.99826058e-01,   9.99781981e-01,
           9.99383777e-01,   9.99233344e-01,   9.99735440e-01,
           9.99932291e-01,   9.99700391e-01]])

: lr.intercept_
: array([0.00234773])
```

The above are the coefficient and intercept of our Linear Regressor model i.e. we know that  $y = a_1x_1 + a_2x_2 + C$ , so here  $a_1, a_2$  are the slopes/coefficients and 'C' is the intercept. Now we will perform regularization of the model in order to avoid the underfitting and overfitting of the model. So basically Lasso and Ridge is used to perform regularization which is available in scikitlearn package. It will bring the coefficients to near 0 as shown below.

```
from sklearn.linear_model import Lasso,Ridge
```

```
ls=Lasso(alpha=0.01)
```

```
ls.fit(x_train,y_train)
```

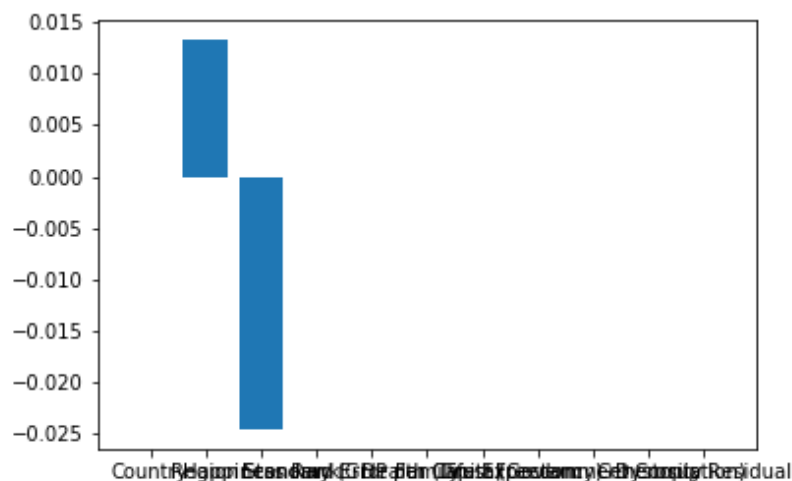
```
Lasso(alpha=0.01, copy_X=True, fit_intercept=True, max_iter=1000,  
      normalize=False, positive=False, precompute=False, random_state=None,  
      selection='cyclic', tol=0.0001, warm_start=False)
```

```
ls.coef_
```

```
array([-0.00016069,  0.01326329, -0.02454029, -0.         ,  0.         ,  
        0.         ,  0.         ,  0.         ,  0.         ,  0.         ,  
        0.         ])
```

```
plt.bar(x.columns,ls.coef_)
```

```
<BarContainer object of 11 artists>
```



One more technique that we can apply to save our model from under and overfitting is using cross validation score, in which the dataset is tested in different phase based on the given **CV** value. It will also help us to improve our score.

Now we will calculate the mean absolute and mean squared error of the data, which is also considered to be one of the important metrics to evaluate. **Mean Absolute Error** of your model refers to the **mean** of the **absolute** values of each prediction **error** on all instances of the test data-set and **MSE** is the average of the **squared error** that is used as the loss function for least squares regression.

```
print(mean_absolute_error(predlr,y_test))
```

```
0.00020786037093478284
```

```
print(mean_squared_error(predlr,y_test))
```

```
6.468263738646996e-08
```

---

So finally we will finalize our model and dump this model using serialization, various packages are available for dumping the model such as pickle, joblib. So here we will use joblib to dump our model so that it can be used again for the prediction.

### **FINAL CONCLUSION:-**

Linear regressor model proves to be the best model for our happiness score prediction with the accuracy of almost all 100 percent so we can say that it is predicting all the values correctly. So, finally we can say that to remain happy factors like GDP, family, corruption plays more important role, the country with these factors high seems to be happier than other countries, other factors like these might serve as an improvement.