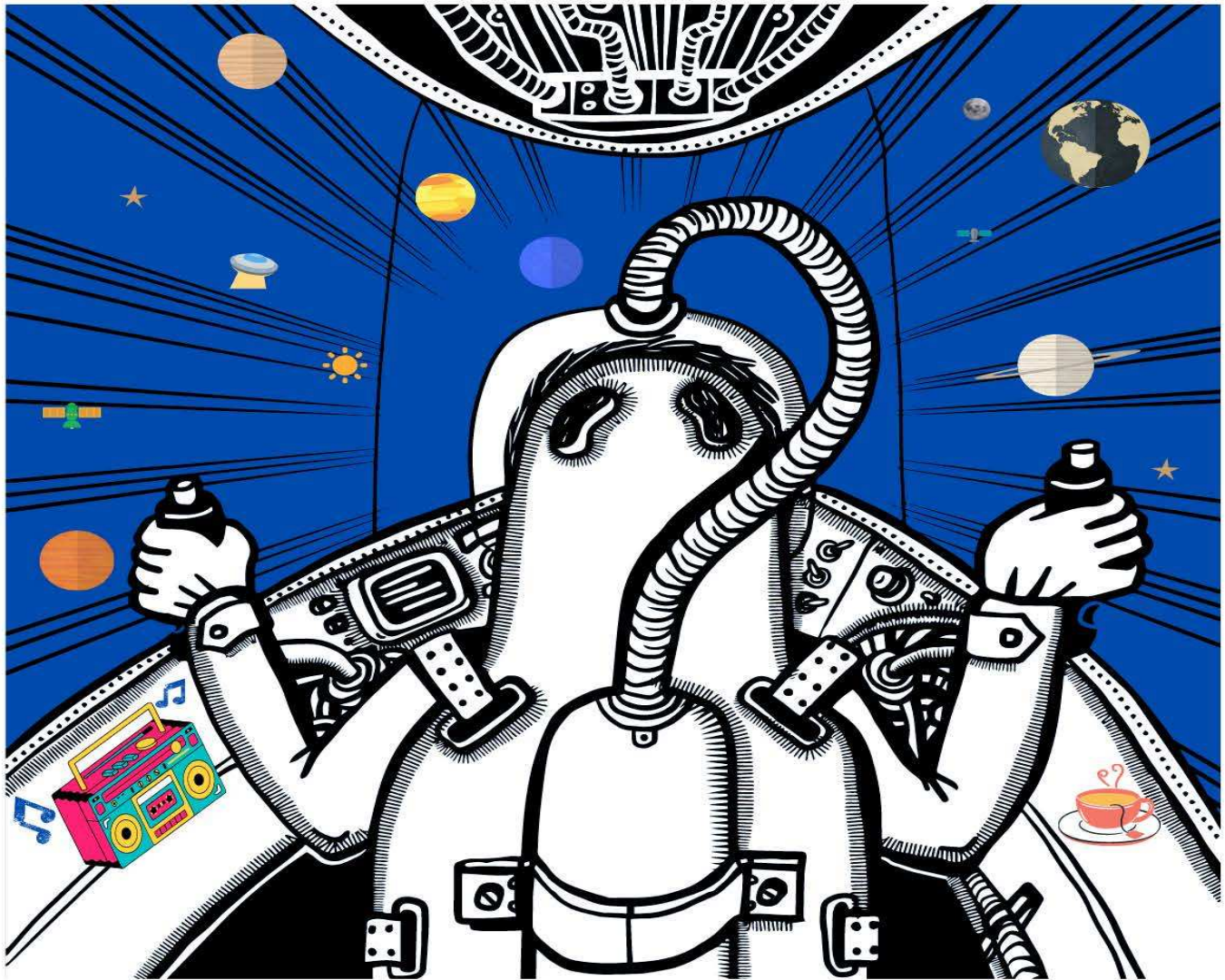


MACHINE LEARNING TUTORIAL-PROJECT 1
WORLD HAPPINESS REPORT ANALYSIS
WITH PYTHON



JOURNEY TO MACHINE LEARNING

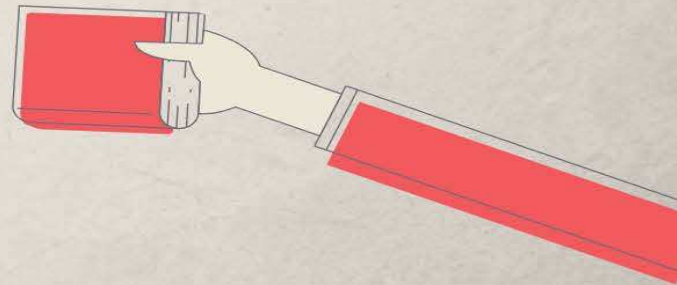


BERKANT ASLAN

PREFACE



I am sharing this work that I have prepared to enable Machine Learning to be learned with Python language and project. For your questions and suggestions, you can send an e-mail to my e-mail address berkantaslan@hotmail.com.tr. There may be some updates on this book over time and I will share updates as they become available. You can follow me to be informed.



**TO MY LOVE, MY FATHER,
MOTHER AND MY SISTER...**

I hope this book will be a compass on Machine Learning for you...





WHO IS THIS BERKANT ASLAN?


He is doing his master's degree in Artificial Intelligence in the field of Mechatronics Engineering at Selcuk University; with interests in Artificial Intelligence, Data Science, Image Processing, Big Data, Business Intelligence and Cyber Security; who loves people very much and is devoted to humanity; Turkey lover; respectful and dear; Isparta's rose :) and he is very happy with his lover :)



 berkantaslan@hotmail.com.tr

 [/berkantaslan](https://www.linkedin.com/company/berkantaslan)

 [@berkantaslan](https://twitter.com/berkantaslan)

 [/berkantaslan](https://github.com/berkantaslan)


 [/berkantaslan](https://kik.me/berkantaslan)

TABLE OF CONTENTS

PREFACE	
TABLE of CONTENTS	1
1. INTRODUCTION	2
2. ABOUT THE DATA	3
3. PROJECT DEFINITION	3
4. METHODOLOGY	4
4.1. CRISP-DM Method	4
4.2. Understanding Data	5
4.3.Data Preprocessing	5
4.3.1. Libraries and Data Importing	5
4.3.2. Data Cleaning	6
4.3.3. Suitable Data Editing to Use.....	6
4.3.4. Data Adding to Data Frame	6
4.4.Exploratory Analysis	7
4.5.Deciding Multiple Linear Regression Method	7
4.6.Data Splitting as Test and Train Cluster	7
4.7.Feature Scaling	7
4.8.Be Careful to Dummy Variable Trap	8
4.9.Fine Tuning	8
4.9.1. Look at the P-Values	8
4.9.2. Comparison of Method's Success with R^2 and Adjusted R^2	8
5. DATA ANALYSIS	10
6. RESULTS	35
7. REFERENCES	35

1. INTRODUCTION

Everybody desires to be happy in life and interestingly the requirements to be happy vary from person to person. We all invest different meanings in the concept of happiness. People give happiness different values at different part of our lives. However, there are some vital factors that are commonly regarded as the main ingredients to be happy in life. Physical and psychological sturdiness is very important to be happy and people can truly realize that when they get sick. Economic freedom - ability to fulfil the needs in life, is the another most important factor to be happy, in my opinion. It is always said that being happy with an empty stomach is not possible. Individual birth – right freedom is also considered a great influence to be happy in life. Someone without birth - right freedom cannot feel the happiness.

Main analysis will consist correlations (regression; output=happiness, variables=economy, health, freedom, etc.), creating a Machine-Learning algorithm (clustering countries according to their happiness rank and score). For example, a country where the happiness score is more than 7.00 is a developed country, a country where the happiness score is between 5.00 and 7.00 is a developing country, and a country where the happiness score is less than 5.00 is undeveloped country.

This analysis will evaluate a connection between people's happiness and their countries. In my opinion, the location where we live is a huge factor of our happiness proportion. The World Happiness Report, produced by the United Nations, ranks about 155 countries by how happy their citizens see themselves to be. It's based on factors including economic wealth, life expectancy, social support, freedom to make life choices and levels of government corruption. The happiness scores and rankings use data from the Gallup World Poll. The scores are based on answers to the main life evaluation question asked in the poll.

Louise Millard do data mining and analysis of Global Happiness with Machine Learning Methods in 2011. This report including 123 countries with happiness values. This report is approached economics, health, climate data. [1]

Natasha Jaques et al do analysis of Predicting Students' Happiness from Physiology, Phone, Mobility, and Behavioral Data with Machine Learning Methods in This report includes physiological data with electrodermal activity (a measure of physiological stress), and 3-axis

accelerometer (a measure of steps and physical activity); survey data with questions related to academic activity, sleep, drug and alcohol use, and exercise; Phone data with phone call, SMS, and usage patterns; location data with coordinates logged throughout the day. In this paper they analysis a machine learning algorithm to distinguish between happy and unhappy college students, assess which measures provide the most information about happiness, and evaluate the relationship between different components of wellbeing including happiness, health, energy, alertness and stress. [2]

2. ABOUT THE DATA

The first used data that is the World Happiness Report is a landmark survey of the state of global happiness. The report that ranks about 155 countries according to happiness levels 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 etc. describe how measurements of well-being can be used effectively to assess the progress of nations.

The Second used data that is the Global Terrorism Database (GTD) is an open-source database including information on terrorist attacks around the world from 1970 through 2017. The GTD includes systematic data on domestic as well as international terrorist incidents that have occurred during this time period and now includes more than 180,000 attacks.

3. PROJECT DEFINITION

The study examines effects of Happiness Score. People's happiness level can be affected by some situations. Machine Learning with CRISP-DM method measures at least to the most effects. Values say the most important things for happiness. The aim of the project provides tricks to governments, organizations and civil society. Multiple Linear Regression model is selected to set up.

On the other hand, these datasets and all outputs will be analyzed again with global terrorist attacks. My plan is creating a new independent variable that does not exist on "World Happiness Reports". The main goal of this purpose is to see whether the geography is a trigger of terrorist attacks and these attacks has a role on citizens' happiness or not.

This project main aim is approaching sociology and politics with machine learning and statistics. It will be very interesting because this is a good example for understanding the usefulness of data science on our lives. No matter data contains numerical or verbal values, we can make analysis and inferences to help building better world.

4. METHODOLOGY

We use Python programming language and Multiple Linear Regression Machine Learning Method for data analysis. We take suitable data from The Global Terrorism Report as country, weapon attack etc. And these columns are added to our World Happiness Report. This data frame splits as test (1/3) and train (2/3) randomly. And multiple linear regression method works on this data frame. And we find which feature affects more than others to happiness score. If any problem for method, we will change to other methods and compare their output values.

4.1. CRISP-DM Method

CRISP-DM (Cross Industry Standard Process for Data Mining) method that a model to define standard processes for data mining are used for data mining projects to become more effective, faster, safer, less costly. This method begins with understanding problem and what we need to do and continues with data preparation, modelling and evaluation for Machine Learning.

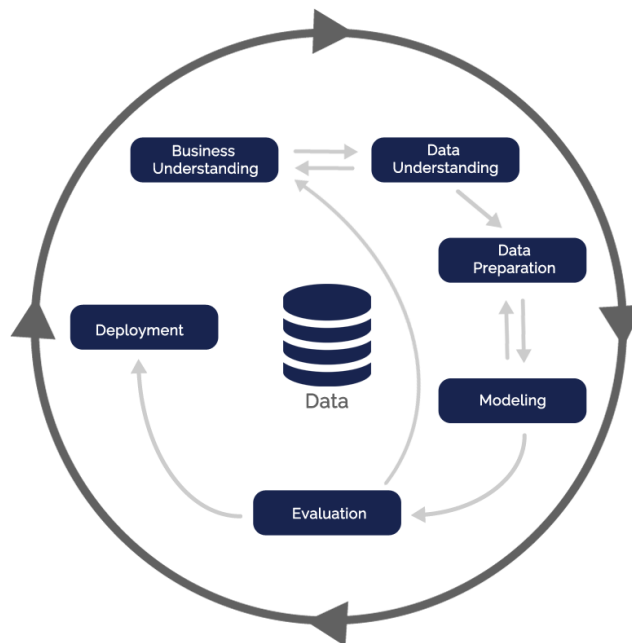


Figure-1: Structure of CRISP-DM Method, Source: ...

4.2. Understanding Data

World Happiness Report includes countries with happiness scores, economy situation (GDP per Capita), health situation (Life Expectancy), freedom situation, trusting situation (Government Corruption), generosity situation. We understand to find relation happiness scores with these values.

Global Terrorism Database report includes years with countries of terrorist events, attack types, weapon types etc. We understand to take and add new features to our data frame. Years and Countries are enough to take.

4.3. Data Preprocessing

Data preprocessing part is important for processing data. This part includes libraries and data importing, data cleaning, suitable data editing to use, data adding to data frame to pass to Machine Learning part. We need to preprocesses for Machine Learning part.

4.3.1. Libraries and Data Importing

We decide to which libraries are used according to processes. In Python, Pandas library is used for data and data frame processes, NumPy library is used for calculates, Matplotlib and Seaborn library is used for plotting, and Sci-Kit Learn library is used for Machine Learning process using “import” command.

Using Data can be csv (comma separated values), excel, html etc. and is taken “read_csv/excel(“file_name”)” command. Data file must be working directory. If it is not in working directory, you can write index of file for file_name. And this defines as any variable.

```
import numpy as np
import pandas as pd

data1 = pd.read_csv('../input/world-happiness-report-with-terrorism/WorldHappinessReportwithTerrorism-2015.csv')
data2 = pd.read_csv('../input/world-happiness-report-with-terrorism/WorldHappinessReportwithTerrorism-2016.csv')
data3 = pd.read_csv('../input/world-happiness-report-with-terrorism/WorldHappinessReportwithTerrorism-2017.csv')
data4 = pd.read_csv('../input/world-happiness/2018.csv')
data5 = pd.read_csv('../input/world-happiness/2019.csv')
data6 = pd.read_csv('../input/world-happiness-report/2020.csv')
data7 = pd.read_csv('../input/global-terrorism-report-for-world-happiness-report/GlobalTerrorismReport-2015.csv')
data8 = pd.read_csv('../input/global-terrorism-report-for-world-happiness-report/GlobalTerrorismReport-2016.csv')
data9 = pd.read_csv('../input/global-terrorism-report-for-world-happiness-report/GlobalTerrorismReport-2017.csv')
```


4.3.2. Data Cleaning

Some Machine Learning algorithms cannot run data with missing values. It must be fixed for processes. Missing values can be seen NaN (not a number), “?”, blank etc. You can impute as for numerical missing values can be fixed statistical methods (mean etc.), putting specific number or deleting rows. For our study, it is not used because data is without missing value and zeros are not affect our value.

We need to diagnose data before exploring, because column name inconsistency like upper-lower case letter or space between words or different languages. Upper-lower case letter or space problem is fixed. We have to change column names, if our data's columns names have upper-lower letter or space. You can change columns names with coding on IDE or typing on Excel.

Terrorism report has a lot of unneeded knowledges for our report. So, we edit this report according to our report as deleting unneeded columns. And we keep only terrorist attacks in 2015, 2016 and 2017.

4.3.3. Suitable Data Editing to Use

Data can be categoric (nominal or ordinal) or numerical (ration or interval). If we have only numerical data, we don't have to edit for suitable data. If we have not, we have to edit data. For example, in our report we have 2 categoric data. First is index number, second is countries. If these things affect happiness score, we have to edit as we change these data to categoric as number. But index number and countries don't affect happiness score. So, we ignore these for Machine Learning process.

We want to see effect of terrorism to world happiness report. So, we have to take suitable data from Terrorism Report. For suitable data, we take only countries with how many terrorist attacks are in same year. We ignore other features and countries that does not exist in World Happiness Report. We need only affectable for happiness report. We take terrorism event counts from edited Global Terrorism Report with only years and countries separately as years.

4.3.4. Data Adding to Data Frame

Now we have 2 data frames: World Happiness data frame and Terrorism data frame with only countries and counts. We have to add terrorism feature to World Happiness Data Frame. We add

these values at same year and same countries. We add terrorism event counts to our World Happiness Report.

4.4. Exploratory Analysis

We want to understand and think about our data. So firstly, we take summary and info from data, like statistical knowledges, how many rows and columns data frames are and etc. And exploratory analysis is need for diagnosing data.

4.5. Deciding Multiple Linear Regression Method

Multiple Linear Regression is to use output is affected by multiple independent values in problem. Happiness scores are affected by multiple independent values. So, I began to try this method for this problem.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

Figure-2: Formula of Multiple Linear Regression Method

In the formula, β_0 is constant number, $\beta_{1,2,3,4,\dots}$ is multiplier of variables, and ϵ is error rate. From this formula, we understand to building straight for problem. And this means that ϵ (error) can be and is need to add calculation. Multiple Linear Regression can be more than 3 dimensions in space.

4.6. Data Splitting as Test and Train Cluster

Data splitting as test and train cluster is done for measuring success Machine Learning Algorithm, aim is developing evaluation. Generally, for test 1/3 and for train 2/3 is used randomly, known as percentage split. Machine learning algorithm run on train data and predict on test data, prediction and test data's happiness score are compared. And success is seen, and we can decide good or not Machine Learning Method. According to success we can change Machine Learning Method is used.

4.7. Feature Scaling

Feature Scaling process is important for Machine Learning. Different columns have different data movements and statistical features (mean, min/max values, standard deviation etc.). So, effects of columns are different and this is problem. We have to fix this problem using feature

scaling. Feature Scaling is done with two classic methods: Standardization and Normalization. We use normalization method. Numerical data are scaled between 0 and 1.

$$Z = \frac{x - \mu}{\sigma} \qquad Z = \frac{x - \min(x)}{[\max(x) - \min(x)]}$$

Figure-3: Formulas of Standardization and Normalization

4.8. Be Careful to Dummy Variable Trap

Dummy variable is situation that data frame has variables have same values. Some Machine Learning Algorithm can be affected from dummy variable. If any situation in data frame, we have to cancel one. For example, man and woman values can be encoded 0 and 1 to transform string value to numeral value. And if we use these two encoded columns, we could have problem. Output affects two times from situation that can be man or woman. Only one column is enough and second one is canceled. For our data frame, it is not needed.

4.9. Fine Tuning

We look at the success of the data and the Machine Learning method chosen. If we have any problems, we will change the method. We look at P-Values for data success and R^2 and Adjusted R^2 for method success.

4.9.1. Look at the P-Values

A p-value indicates the significance of a result, representing the probability the result would occur by chance. A low value means the result is unlikely to occur randomly and hence is statistically significant. Threshold values of 5% and 1% are commonly used, below which a result can be stated as significant. Statistical tests can be relative to test parameters such as the size of the dataset, and a p-value provides a comparable value that takes into account such aspects of the data. [1]

4.9.2. Comparison of Method's Success with R^2 and Adjusted R^2

R^2 is also known as the coefficient of determination, and is an extension of the R value. It represents the proportion of variation in the label that can be accounted for by the regression model. However, R is relative to the number of variables used in the model and therefore is not comparable

when this differs. Adjusted R^2 takes into consideration the number of variables used in the regression. [1]

The closer the R^2 and Adjusted R^2 values are to 1, the better the case for the chosen method. The Multiple Linear Regression Method is good for our analysis. We analyze the effects of happiness scores. All values in the dataset are important because all values affect happiness scores. Therefore, we do not cancel or ignore any features.

5. DATA ANALYSIS

World Happiness Rank 2015

RangeIndex: 158 entries, 0 to 157	
Data columns (total 12 columns):	
country	158 non-null object
region	158 non-null object
happinessrank	158 non-null int64
happinessscore	158 non-null float64
standarderror	158 non-null float64
economysituation	158 non-null float64
family	158 non-null float64
healthlifeexpectancy	158 non-null float64
freedom	158 non-null float64
governmentcorruption	158 non-null float64
generosity	158 non-null float64
dystopiasresidual	158 non-null float64
terrorismevent	158 non-null int64

Table-1: World Happiness Rank 2015 Data Summary

We understand from Table-1 that our 2015 data includes 158 countries, 12 features with non-null values.

We use the Tableau program with Business Intelligence tools to see the names of countries as word bags according to their happiness scores.



Figure-5: Countries by Happiness Scores in 2015 World Happiness Rankings

```
print(data1.columns)
print(data1.info())
print(data1.describe())

df1 = data1["country"]
dff1 = df1.value_counts()
```

We find that our 2015 data includes 158 countries, 12 features with non-blank values, and all other statistical information from the code output.

```
x1 = data1.iloc[:,5:].values
y1 = data1.iloc[:,3:4].values
```

They are assigned to the variable x and y as the output of the code, x as the input columns affecting the happiness score, and y as the resultant happiness score column.

```
from sklearn.model_selection import train_test_split
x_train1, x_test1, y_train1, y_test1 = train_test_split(x1, y1, test_size=0.33, random_state=0)
```

Test and training sets are created as code output.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train1 = sc.fit_transform(x_train1)
X_test1 = sc.fit_transform(x_test1)
Y_train1 = sc.fit_transform(y_train1)
Y_test1 = sc.fit_transform(y_test1)
```

Normalization is done as a code output.

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train1, y_train1)
print("b0: ", lr.intercept_)
print("other b: ", lr.coef_)
```

A machine learning model is created as a code output and weight factor values are calculated according to the model.

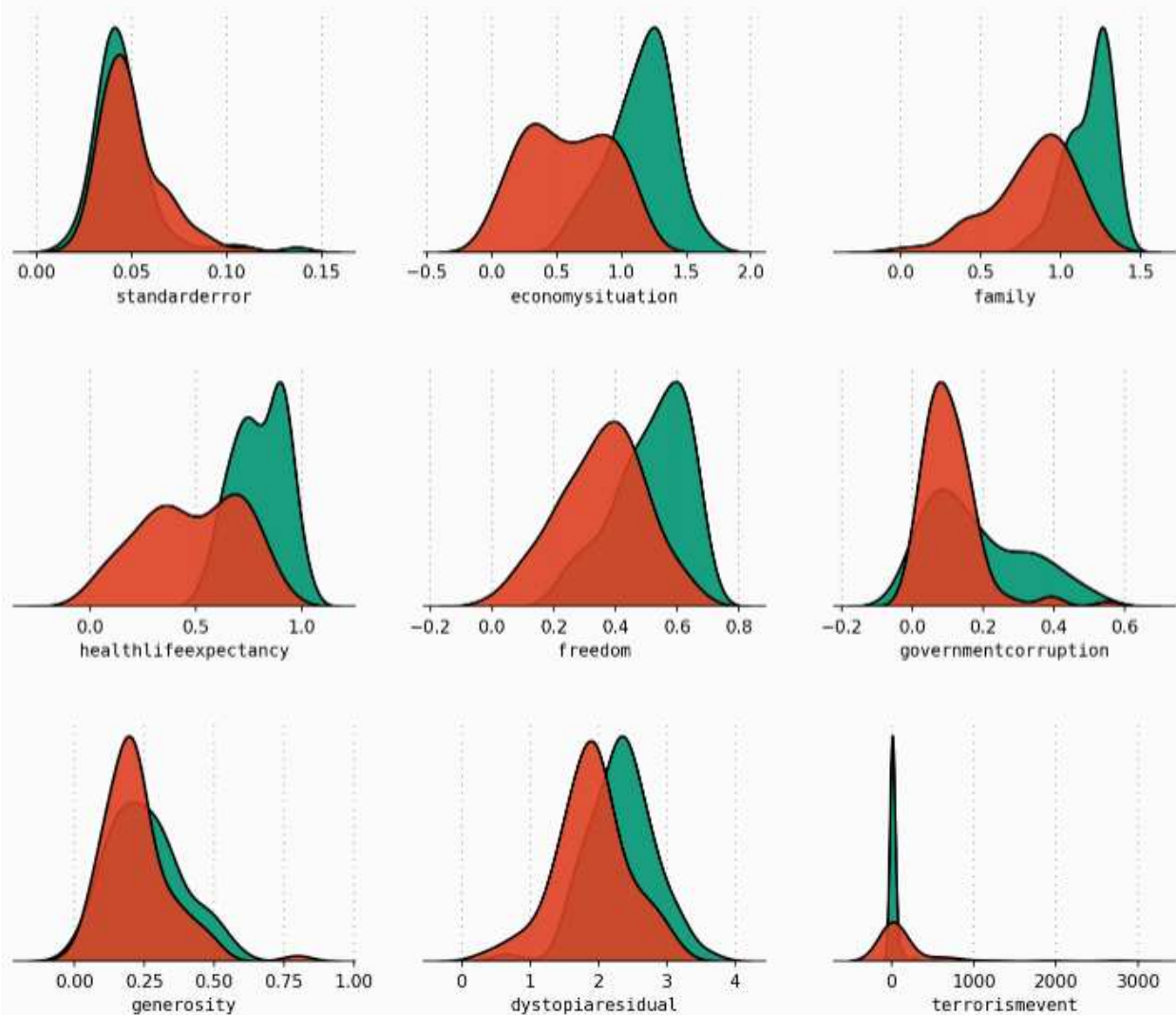
```
y_pred1 = lr.predict(x_test1)
prediction1 = lr.predict(np.array([[1.16492,0.87717,0.64718,0.23889,0.12348,0.04707,2.29074,542]]))
print("Prediction is ", prediction1)
```

```
y_pred1 = lr.predict(x_test1)
prediction1 = lr.predict(np.array([[1.198274,1.337753,0.637606,0.300741,0.099672,0.046693,1.879278,181]]))
print("Prediction is ", prediction1)
```

As code outputs, we see from the report that the real value for Turkey in the 2015 report will be 5,332 happiness points, our model estimates it to be 5.38965305 from the 2016 report and 5.50012402 from the 2017 report. We estimate that the reason for the 2017 value and others' differences is that the characteristics and the impact of terrorism have changed.

Differences Between Happy & Unhappy Countries

There are large differences, with GDP & Social Support being clear perhaps more interesting though, unhappy countries appear to be more generous.



```

low_c = '#dd4124'
high_c = '#009473'
background_color = '#fbfbfb'
fig = plt.figure(figsize=(12, 10), dpi=150, facecolor=background_color)
gs = fig.add_gridspec(3, 3)
gs.update(wspace=0.2, hspace=0.5)

newdata1 = data1.iloc[:,4:]
categorical = [var for var in newdata1.columns if newdata1[var].dtype=='O']
continuous = [var for var in newdata1.columns if newdata1[var].dtype!='O']

happiness_mean = data1['happinessscore'].mean()

data1['lower_happy'] = data1['happinessscore'].apply(lambda x: 0 if x < happiness_mean else 1)

plot = 0
for row in range(0, 3):
    for col in range(0, 3):
        locals()["ax"+str(plot)] = fig.add_subplot(gs[row, col])
        locals()["ax"+str(plot)].set_facecolor(background_color)
        locals()["ax"+str(plot)].tick_params(axis='y', left=False)
        locals()["ax"+str(plot)].get_yaxis().set_visible(False)
        locals()["ax"+str(plot)].set_axisbelow(True)
        for s in ["top", "right", "left"]:
            locals()["ax"+str(plot)].spines[s].set_visible(False)
        plot += 1

plot = 0

Yes = data1[data1['lower_happy'] == 1]
No = data1[data1['lower_happy'] == 0]

for variable in continuous:
    sns.kdeplot(Yes[variable],ax=locals()["ax"+str(plot)], color=high_c,ec='black', shade=True, linewidth=1.5, alpha=0.9, zorder=3, legend=False)
    sns.kdeplot(No[variable],ax=locals()["ax"+str(plot)], color=low_c, shade=True, ec='black',linewidth=1.5, alpha=0.9, zorder=3, legend=False)
    locals()["ax"+str(plot)].grid(which='major', axis='x', zorder=0, color='gray', linestyle=':', dashes=(1,5))
    locals()["ax"+str(plot)].set_xlabel(variable, fontfamily='monospace')
    plot += 1

Xstart, Xend = ax0.get_xlim()
Ystart, Yend = ax0.get_ylim()

ax0.text(Xstart, Yend+(Yend*0.5), 'Differences Between Happy & Unhappy Countries', fontsize=15, fontweight='bold', fontfamily='sansserif',color='#323232')
ax0.text(Xstart, Yend+(Yend*0.25), 'There are large differences, with GDP & Social Support being clear perhaps more interesting though,unhappy\ncountries appear to be more generous.', fontsize=10, fontweight='light', fontfamily='monospace',color='gray')

plt.show()

import statsmodels.regression.linear_model as sm
X1 = np.append(arr = np.ones((158,1)).astype(int), values=x1, axis=1)
r_ols1 = sm.OLS(endog = y1, exog = X1)
r1 = r_ols1.fit()
print(r1.summary())

```

R squared:	1.000
Adj. R squared:	1.000

Table-2: The Success of Machine Learning Method in World Happiness Ranking 2015 Data

We can see from Table-2 that our model is very successful. Therefore, we are not changing our selected Machine Learning method. R-squared and Adj. R-squared values (1.000 and 1.000) show the Success of Machine Learning Method in World Happiness Ranking 2015 Data. We understand that our model is very successful if these values are 1 or close to 1. Therefore, we are not changing our selected Machine Learning method.

Variables	P-Value
constant	0.586
economysituation	0.000
family	0.000
healthlifeexpectancy	0.000
freedom	0.000
governmentcorruption	0.000
generosity	0.000
dystopiaresidual	0.000
terrorismevent	0.846

Table-3: The Success of Variables on World Happiness Ranking 2015 Data

$P > |t|$ value indicates the Success of the Variables on the World Happiness Ranking 2015 Data. We can see from Table-3 that the p values of the fixed value we added as 1 and the added terrorist event values are higher than the 5% or 1% threshold values. This means that the values of terrorism events are not appropriate and affectable for our model.

World Happiness Rank 2016

RangeIndex: 157 entries, 0 to 156	
Data columns (total 13 columns):	
country	157 non-null object
region	157 non-null object
happinessrank	157 non-null int64
happinessscore	157 non-null float64
lowerconfidenceinterval	157 non-null float64
upperconfidenceinterval	157 non-null float64
economysituation	157 non-null float64
family	157 non-null float64
healthlifeexpectancy	157 non-null float64
freedom	157 non-null float64
governmentcorruption	157 non-null float64
generosity	157 non-null float64
dystopiasresidual	157 non-null float64
terrorismevent	157 non-null int64

Table-4: World Happiness Rank 2016 Veri Özeti

From Table-4, we understand that our 2016 data are 157 countries, 13 features with non-blank values.

We use the Tableau program with Business Intelligence tools to see the names of countries as word bags according to their happiness scores.



Figure-6: Countries by Happiness Scores in the 2016 World Happiness Rankings

```
print(data2.columns)
print(data2.info())
print(data2.describe())
df2 = data2["country"]
dff2 = df2.value_counts()
```

We find that our 2016 data includes 157 countries, 13 features with non-blank values, and all other statistical information from the code output.

```
x2 = data2.iloc[:,6:].values
y2 = data2.iloc[:,3:4].values
```

They are assigned to the variable x and y as the output of the code, x as the input columns affecting the happiness score, and y as the resultant happiness score column.

```
from sklearn.model_selection import train_test_split
x_train2, x_test2, y_train2, y_test2 = train_test_split(x2, y2, test_size=0.33, random_state=0)
```

Test and training sets are created as code output.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train2 = sc.fit_transform(x_train2)
X_test2 = sc.fit_transform(x_test2)
Y_train2 = sc.fit_transform(y_train2)
Y_test2 = sc.fit_transform(y_test2)
```

Normalization is done as a code output.

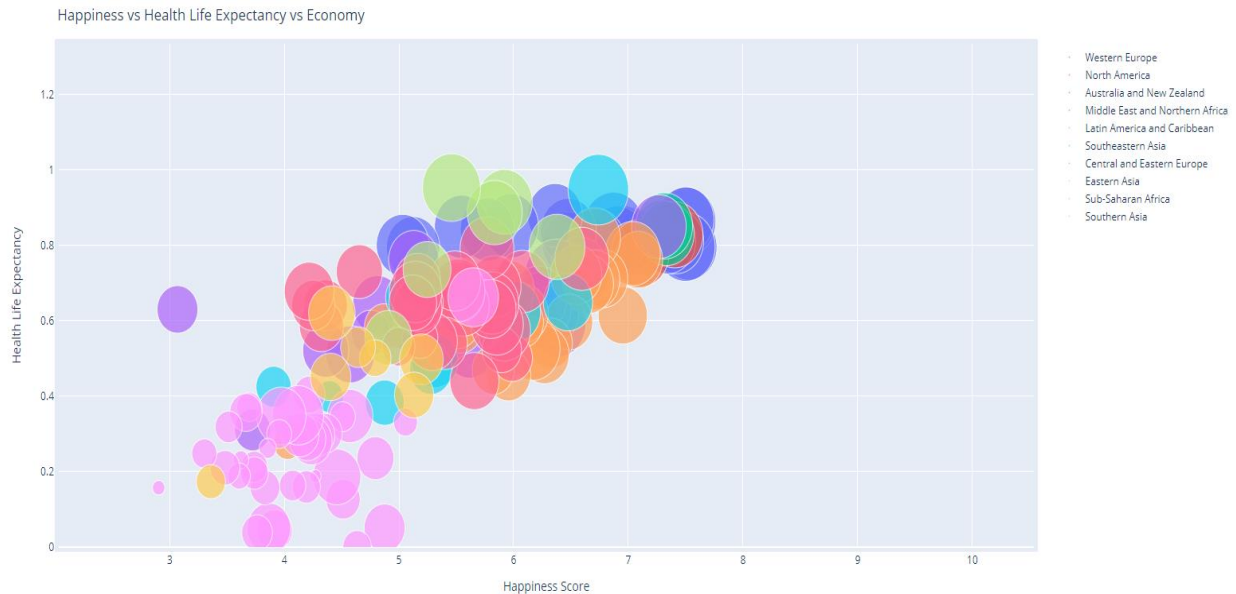
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train2, y_train2)
print("b0: ", lr.intercept_)
print("other b: ", lr.coef_)
```

A machine learning model is created as a code output and weight factor values are calculated according to the model.

```
y_pred2 = lr.predict(x_test2)
prediction2 = lr.predict(np.array([[1.06098,0.94632,0.73172,0.22815,0.15746,0.12253,2.08528,422]]))
print("Prediction is ", prediction2)
```

```
y_pred2 = lr.predict(x_test2)
prediction2 = lr.predict(np.array([[1.198274,1.337753,0.637606,0.300741,0.099672,0.046693,1.879278,181]]))
print("Prediction is ", prediction2)
```

We see from the code output that the real value for Turkey in the 2016 report will be 5,389 happiness points, our model estimates it as 5.33218602 from the 2015 report and 5.49986729 from the 2017 report. We estimate that the reason for the 2017 value and others' differences is that the characteristics and the impact of terrorism.



```
figure = bubbleplot(dataset = data2, x_column = 'happinessscore', y_column = 'healthlifeexpectancy',
                    bubble_column = 'country', size_column = 'economicsituation', color_column = 'region',
                    x_title = "Happiness Score", y_title = "Health Life Expectancy", title = 'Happiness vs Health Life Exp
ectancy vs Economy',
                    x_logscale = False, scale_bubble = 1, height = 650)
po.ipplot(figure)
```

```
import statsmodels.regression.linear_model as sm
X2 = np.append(arr = np.ones((157,1)).astype(int), values=x2, axis=1)
r_ols2 = sm.OLS(endog = y2, exog = X2)
r2 = r_ols2.fit()
print(r2.summary())
```

R squared:	1.000
Adj. R squared:	1.000

Table-5: The Success of Machine Learning Method in World Happiness Ranking 2016 Data

R-squared and Adj. R-squared values (1.000 and 1.000) show the Success of Machine Learning Method in World Happiness Ranking 2016 Data. We understand that our model is very successful if these values are 1 or close to 1. Therefore, we are not changing our selected Machine Learning method.

Variables	P-Value
constant	0.281
economysituation	0.000
family	0.000
healthlifeexpectancy	0.000
freedom	0.000
governmentcorruption	0.000
generosity	0.000
dystopiarresidual	0.000
terrorismevent	0.619

Table-6: The Success of Variables on World Happiness Ranking 2016 Data

$P > |t|$ value indicates the Success of the Variables on the World Happiness Ranking 2016 Data. We understand from Table-6 that the p values of our constant and added terror event values, which we added as 1, are more than the threshold values of 5% or 1%. This means that the values of terrorism events are not appropriate and affectable for our model. However, we see that these p-values decrease, which means that their effect on our model increases.

World Happiness Rank 2017

RangeIndex: 155 entries, 0 to 154	
Data columns (total 12 columns):	
country	155 non-null object
happinessrank	155 non-null int64
happinessscore	155 non-null float64
whiskerhigh	155 non-null float64
whiskerlow	155 non-null float64
economysituation	155 non-null float64
family	155 non-null float64
healthlifeexpectancy	155 non-null float64
freedom	155 non-null float64
generosity	155 non-null float64
governmentcorruption.	155 non-null float64
dystopiaresidual	155 non-null float64
terrorismevent	155 non-null int64

Table-7: World Happiness Rank 2017Veri Özeti

From Table-7 we understand that our 2017 data is 155 countries, 12 features with non-blank values.

We use the Tableau program, which includes Business Intelligence tools, to see the names of countries according to their happiness scores as a word bag with the shape of the countries.

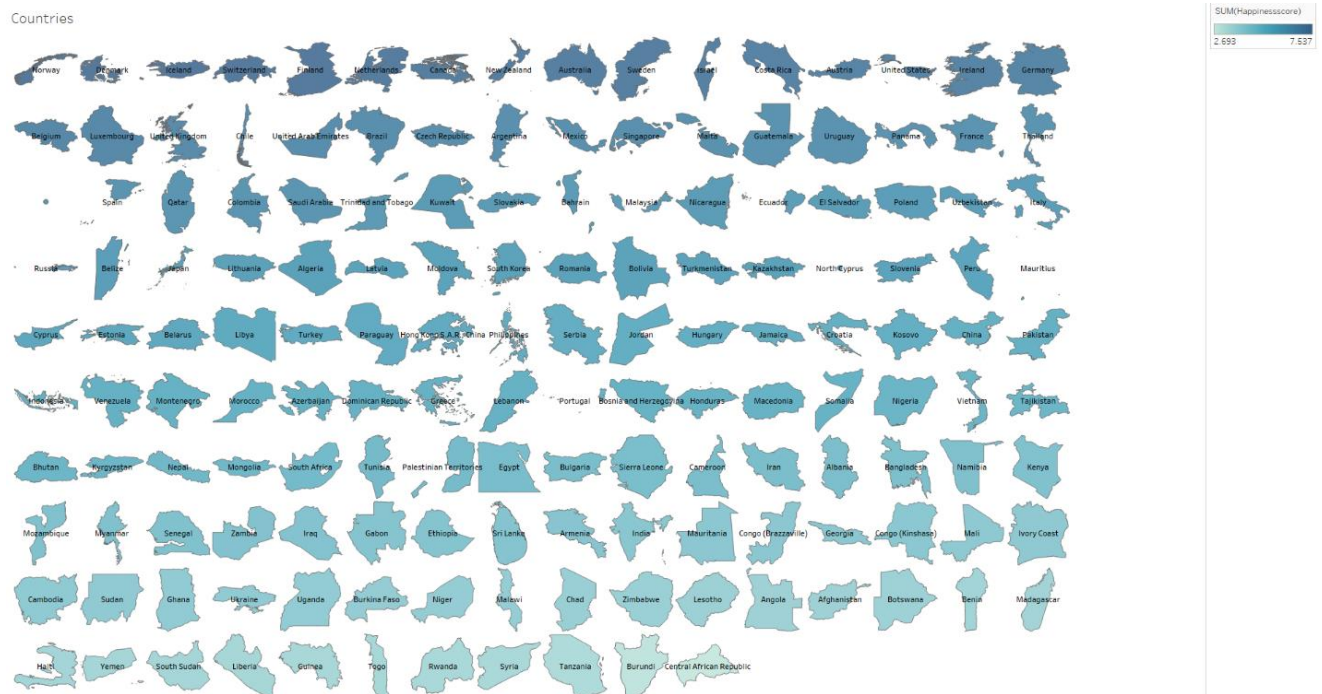


Figure-7: Countries by Happiness Scores in 2017 World Happiness Rankings

```
print(data3.columns)
print(data3.info())
print(data3.describe())
df3 = data3["country"]
dff3 = df3.value_counts()
```

We find that our 2017 data includes 155 countries, 12 features with non-blank values, and all other statistical information from the code output.

```
x3 = data3.iloc[:,5:].values
y3 = data3.iloc[:,2:3].values
```

They are assigned to the variable x and y as the output of the code, x as the input columns affecting the happiness score, and y as the resultant happiness score column.

```
from sklearn.model_selection import train_test_split
x_train3, x_test3, y_train3, y_test3 = train_test_split(x3, y3, test_size=0.33, random_state=0)
```

Test and training sets are created as code output.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train3 = sc.fit_transform(x_train3)
X_test3 = sc.fit_transform(x_test3)
Y_train3 = sc.fit_transform(y_train3)
Y_test3 = sc.fit_transform(y_test3)
```

Normalization is done as a code output.

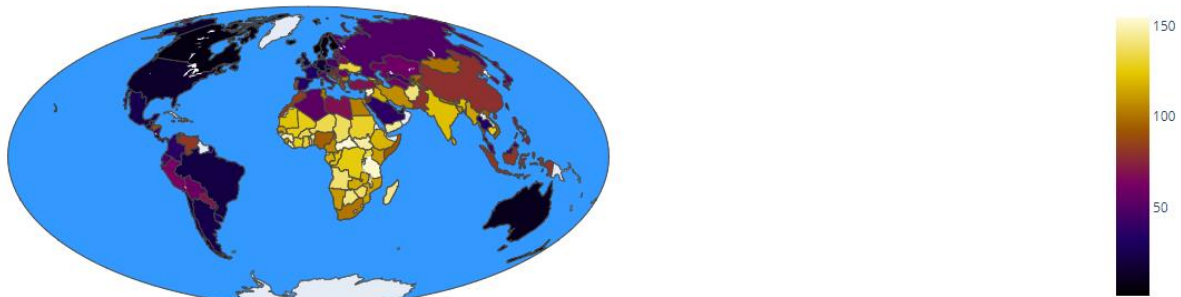
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train3, y_train3)
print("b0: ", lr.intercept_)
print("other b: ", lr.coef_)
```

A machine learning model is created as a code output and weight factor values are calculated according to the model.

```
y_pred3 = lr.predict(x_test3)
prediction3 = lr.predict(np.array([[1.06098,0.94632,0.73172,0.22815,0.15746,0.12253,2.08528,422]]))
print("Prediction is ", prediction3)
```

```
y_pred3 = lr.predict(x_test3)
prediction3 = lr.predict(np.array([[1.16492,0.87717,0.64718,0.23889,0.12348,0.04707,2.29074,542]]))
print("Prediction is ", prediction3)
```

As a code output, we see from the report that the real value for Turkey in the 2017 report will be 5.5 happiness points, our model estimates it to be 5.33249758 from the 2015 report and 5.38949664 from the 2016 report. If you want to use this model, you can use the 2015 and 2016 reports because of its equivalence.



```

trace1 = [go.Choropleth(
    colorscale = 'Electric',
    locationmode = 'country names',
    locations = data3['country'],
    text = data3['country'],
    z = data3['happinessrank'],
)]

layout = dict(title = 'Happiness Rank',
    geo = dict(
        showframe = True,
        showocean = True,
        showlakes = True,
        showcoastlines = True,
        projection = dict(
            type = 'hammer'
        )
    ))

projections = [ "equirectangular", "mercator", "orthographic", "natural earth", "kavrayskiy7",
    "miller", "robinson", "eckert4", "azimuthal equal area", "azimuthal equidistant",
    "conic equal area", "conic conformal", "conic equidistant", "gnomonic", "stereographic",
    "mollweide", "hammer", "transverse mercator", "albers usa", "winkel tripel" ]

buttons = [dict(args = ['geo.projection.type', y],
    label = y, method = 'relayout') for y in projections]

annot = list([ dict( x=0.1, y=0.8, text='Projection', yanchor='bottom',
    xref='paper', xanchor='right', showarrow=False )])

# Update Layout Object
layout[ 'updatemenus' ] = list([ dict( x=0.1, y=0.8, buttons=buttons, yanchor='top' )])
layout[ 'annotations' ] = annot

fig = go.Figure(data = trace1, layout = layout)
po.iplot(fig)

```



```
import statsmodels.regression.linear_model as sm
X3 = np.append(arr = np.ones((155,1)).astype(int), values=x3, axis=1)
r_ols3 = sm.OLS(endog = y3, exog = X3)
r3 = r_ols3.fit()
print(r3.summary())
```

R squared:	1.000
Adj. R squared:	1.000

Table-8: The Success of Machine Learning Method in World Happiness Ranking 2017 Data

We can see from Table-8 that our model is very successful. R-squared and Adj. R-squared values (1.000 and 1.000) show the Success of Machine Learning Method in World Happiness Ranking 2017 Data. We understand that our model is very successful if these values are 1 or close to 1. Therefore, we are not changing our selected Machine Learning method.

Variable	P-Value
constant	0.259
economysituation	0.000
family	0.000
healthlifeexpectancy	0.000
freedom	0.000
governmentcorruption	0.000
generosity	0.000
dystopiaresidual	0.000
terrorismevent	0.939

Table-9: The Success of Variables on World Happiness Ranking 2017 Data

$P > |t|$ value indicates the Success of the Variables on the World Happiness Ranking 2017 Data. We can see from Table-9 that the p values of the fixed value we added as 1 and the added terrorist event values are higher than the 5% or 1% threshold values. This means that the values of terrorism events are not appropriate and affectable for our model.

For the 2017 report only, we use Tableau with Business Intelligence tools to see graphs of countries' trait values for happiness scores.

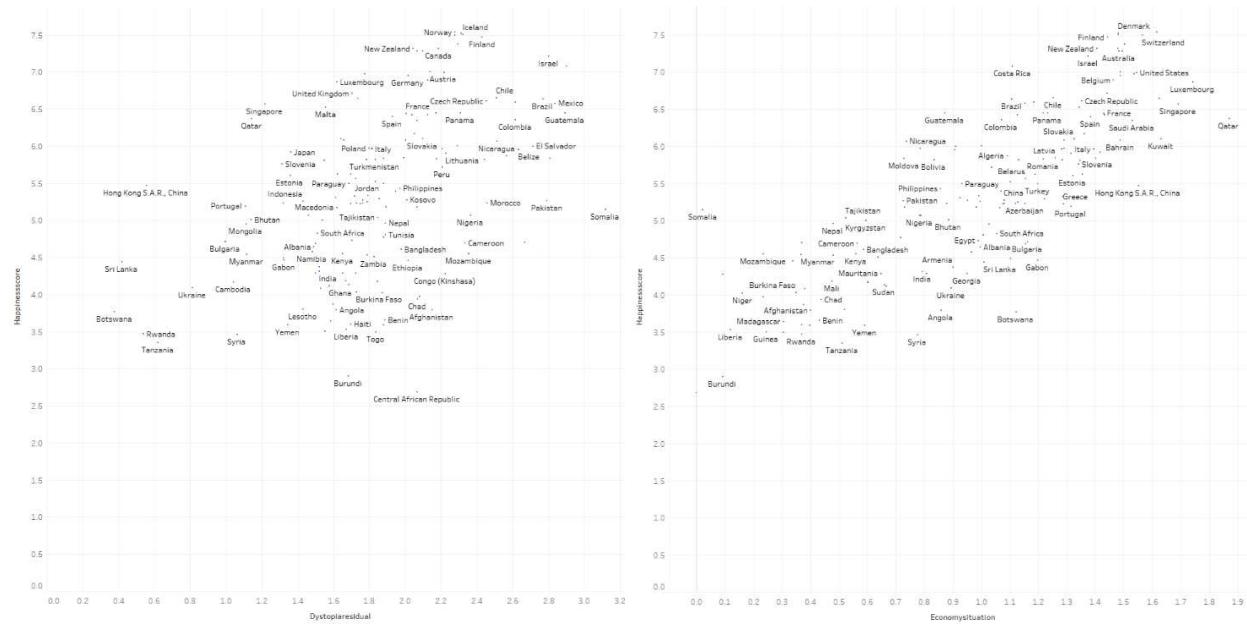


Figure-8: Dystopia Residual and Economy to Happiness Score Chart

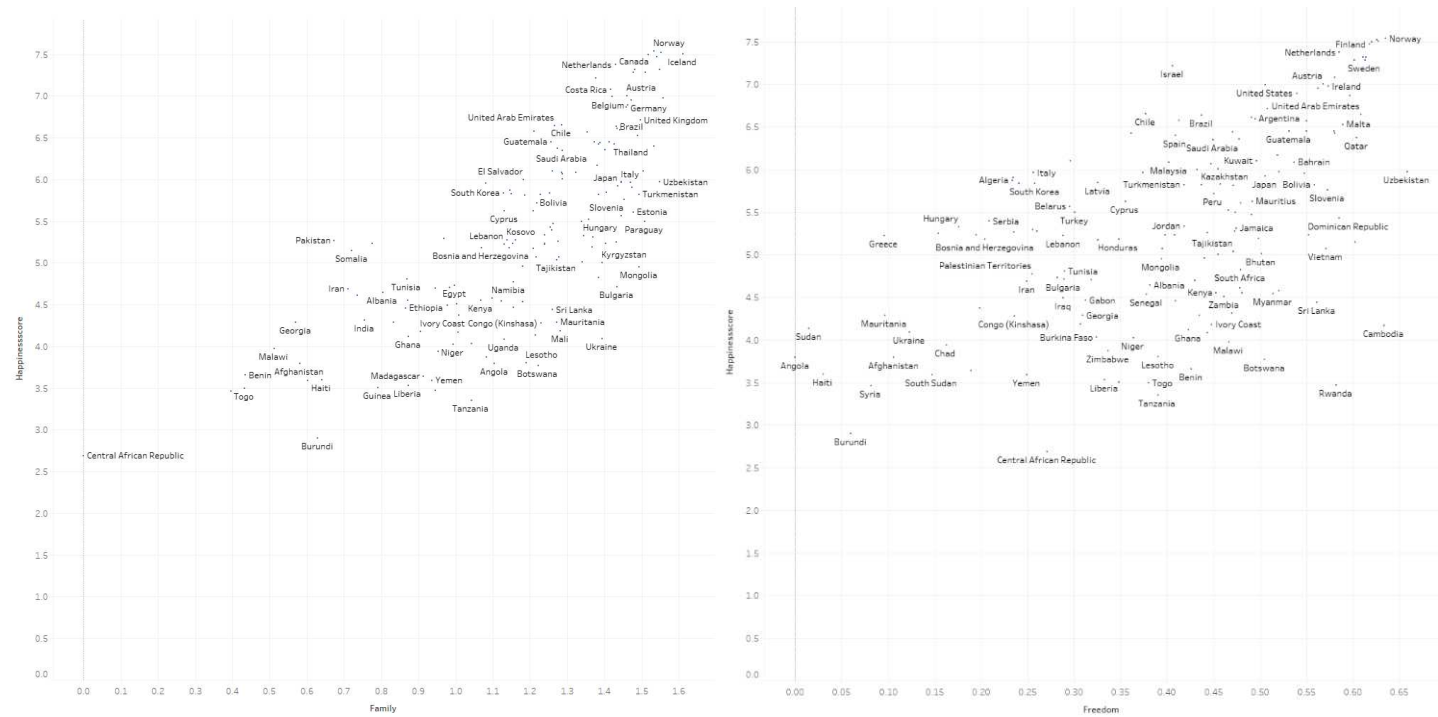


Figure-9: Family and Freedom to Happiness Score Chart

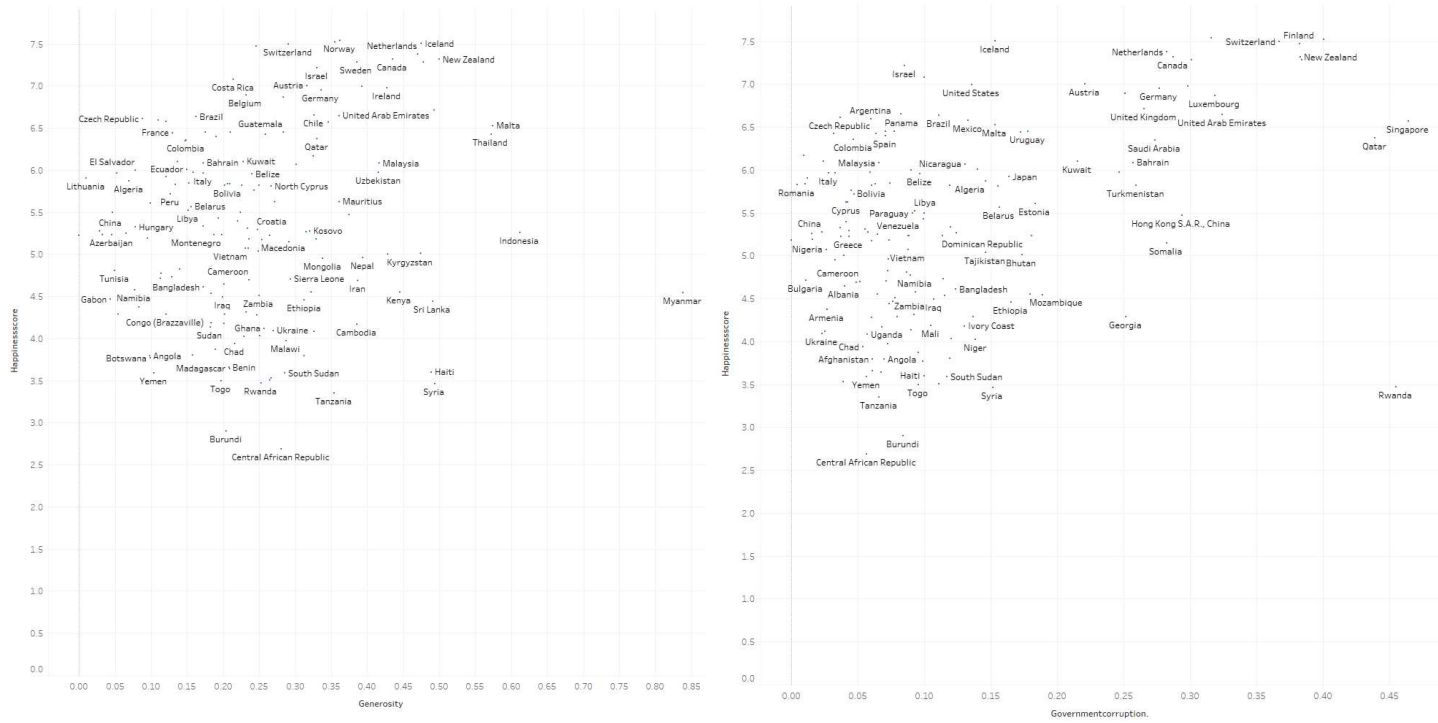


Figure-10: Generosity and Government Corruption to Happiness Score Chart

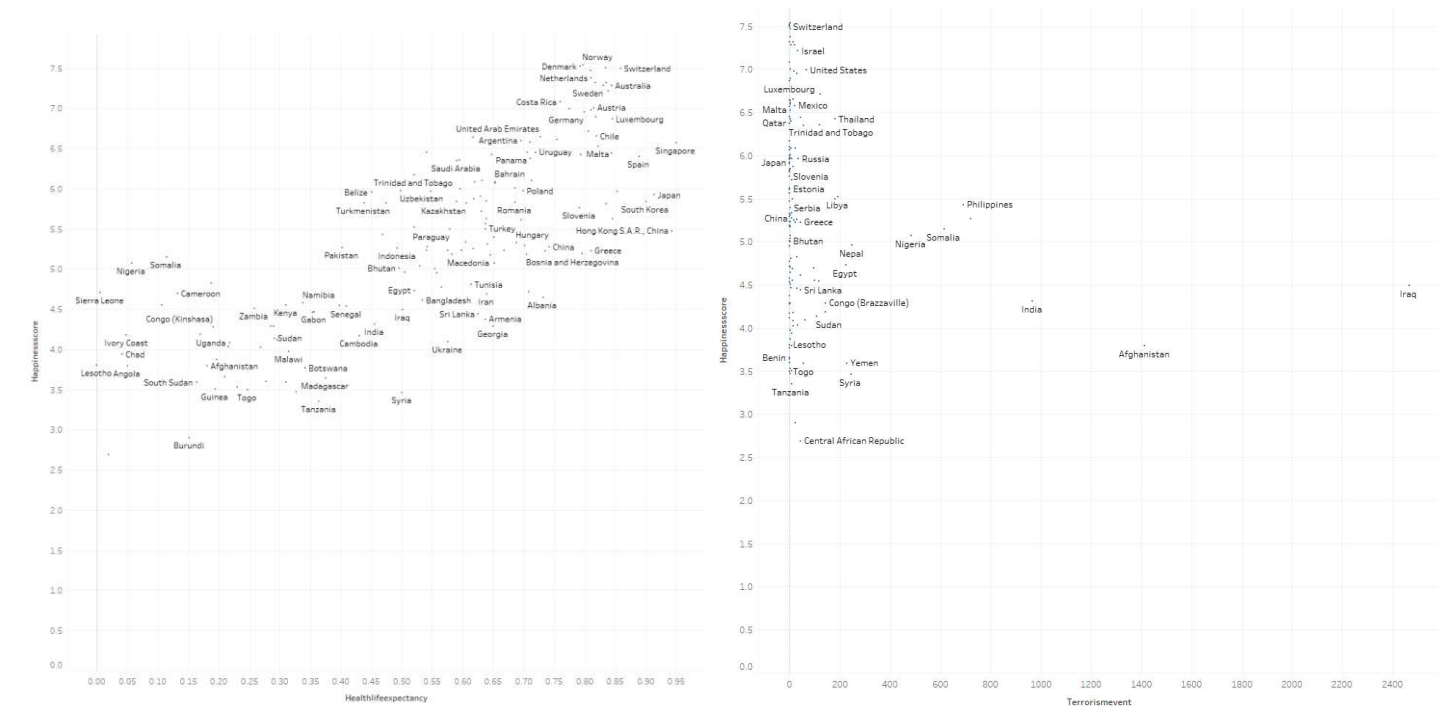


Figure-11: Health Life Expectancy and Terrorism Events to Happiness Score Chart

World Happiness Rank 2018

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value= 0)
impdata = data4.iloc[:,2:9].values
imputer = imputer.fit(impdata[:,2:9])
impdata[:,2:9] = imputer.transform(impdata[:,2:9])
data4.iloc[:,2:9] = impdata[:,:]
```

As a code output, the missing data is filled as 0.

Range Index: 156 entries, 0 to 155	
Data columns (total 9 columns)	
Country or region	156 non-null object
Score	156 non-null int64
GDP per capita	156 non-null float64
Social support	156 non-null float64
Healty life expectancy	156 non-null float64
Freedom to make life choices	156 non-null float64
Generosity	156 non-null float64
Perceptions of corruption	156 non-null float64

Table-10: World Happiness Rank 2018 Data Summary

```
print(data4.columns)
print(data4.info())
print(data4.describe())
df4 = data4["Country or region"]
dff4 = data4.value_counts()
```

We find that our 2018 data includes 156 countries, 7 features with non-blank values, and all other statistical information from the code output.

```
x4 = data4.iloc[:,3:].values
y4 = data4.iloc[:,2:3].values
```

They are assigned to the variable x and y as the output of the code, x as the input columns affecting the happiness score, and y as the resultant happiness score column.

```
from sklearn.model_selection import train_test_split
x_train4, x_test4, y_train4, y_test4 = train_test_split(x4, y4, test_size=0.33, random_state=0)
```

Test and training sets are created as code output.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train4 = sc.fit_transform(x_train4)
X_test4 = sc.fit_transform(x_test4)
Y_train4 = sc.fit_transform(y_train4)
Y_test4 = sc.fit_transform(y_test4)
```

Normalization is done as a code output.

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train4, y_train4)
print("b0: ", lr.intercept_)
print("other b: ", lr.coef_)
```

A machine learning model is created as a code output and weight factor values are calculated according to the model.

```
import statsmodels.regression.linear_model as sm
X4 = np.append(arr = np.ones((156,1)).astype(int), values=x4, axis=1)
r_ols4 = sm.OLS(endog = y4, exog = X4)
r4 = r_ols4.fit()
print(r4.summary())
```

R squared:	0.789
Adj. R squared:	0.781

Table-11: The Success of Machine Learning Method in World Happiness Ranking 2018 Data

Variables	P-Value
constant	0.000
GDP per capita	0.000
Social support	0.000
Healty life expectancy	0.015
Freedom to make life choices	0.000
Generosity	0.222
Perceptions of corruption	0.200

Table-12: The Success of Variables on World Happiness Ranking 2018 Data

World Happiness Rank 2019

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value= 0)
impdata = data5.iloc[:,3:9].values
imputer = imputer.fit(impdata[:,3:9])
impdata[:,3:9] = imputer.transform(impdata[:,3:9])
data5.iloc[:,3:9] = impdata[:,:]
```

As a code output, the missing data is filled as 0.

Range Index: 153 entries, 0 to 152		
Data columns (total 20 columns)		
Country name	153 non-null	object
Regional indicator	153 non-null	object
Ladder score	153 non-null	float64
Standard error of ladder score	153 non-null	float64
upperwhisker	153 non-null	float64
lowerwhisker	153 non-null	float64
Logged GDP per capita	153 non-null	float64
Social support	153 non-null	float64
Healthy life expectancy	153 non-null	float64
Freedom to make life choices	153 non-null	float64
Generosity	153 non-null	float64
Perceptions of corruption	153 non-null	float64
Ladder score in Dystopia	153 non-null	float64
Explained by: Logged GDP per capita	153 non-null	float64
Explained by: Social support	153 non-null	float64
Explained by: Healthy life expectancy	153 non-null	float64
Explained by: Freedom to make life choices	153 non-null	float64
Explained by: Generosity	153 non-null	float64
Explained by: Perceptions of corruption	153 non-null	float64
Dystopia + residual	153 non-null	float64

Table-13: World Happiness Rank 2019 Data Summary

```
print(data5.columns)
print(data5.info())
print(data5.describe())

df5 = data5["Country or region"]
dff5 = data5.value_counts()
```

We find that our 2019 data includes 156 countries, 7 features with non-blank values, and all other statistical information from the code output.

```
x5 = data5.iloc[:,3:].values
y5 = data5.iloc[:,2:3].values
```

They are assigned to the variable x and y as the output of the code, x as the input columns affecting the happiness score, and y as the resultant happiness score column.

```
from sklearn.model_selection import train_test_split
x_train5, x_test5, y_train5, y_test5 = train_test_split(x5, y5, test_size=0.33, random_state=0)
```

Test and training sets are created as code output.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train5 = sc.fit_transform(x_train5)
X_test5 = sc.fit_transform(x_test5)
Y_train5 = sc.fit_transform(y_train5)
Y_test5 = sc.fit_transform(y_test5)
```

Normalization is done as a code output.

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train5, y_train5)
print("b0: ", lr.intercept_)
print("other b: ", lr.coef_)
```

A machine learning model is created as a code output and weight factor values are calculated according to the model.

```
import statsmodels.regression.linear_model as sm
X5 = np.append(arr = np.ones((156,1)).astype(int), values=x5, axis=1)
r_ols5 = sm.OLS(endog = y5, exog = X5)
r5 = r_ols5.fit()
print(r5.summary())
```

R squared:	1.000
Adj. R squared:	1.000

Table-14: The Success of Machine Learning Method in World Happiness Ranking 2019 Data

Variables	P-Value
constant	0.194
Logged GDP per capita	0.081
Social support	0.524
Healty life expectancy	0.714
Freedom to make life choices	0.207
Generosity	0.181
Perceptions of corruption	0.172
Ladder score in Dystopia	0.194
Explained by: Logged GDP per capita	0.576
Explained by: Social support	0.000
Explained by: Healthy life expectancy	0.410
Explained by: Freedom to make life choices	0.605
Explained by: Generosity	0.996
Explained by: Perceptions of corruption	0.959
Dystopia + residual	0.000

Table-15: The Success of Variables on World Happiness Ranking 2019 Data

World Happiness Rank 2020

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value= 0)
impdata = data6.iloc[:,4:20].values
imputer = imputer.fit(impdata[:,4:20])
impdata[:,4:20] = imputer.transform(impdata[:,4:20])
data6.iloc[:,4:20] = impdata[:,:]
```

As a code output, the missing data is filled as 0.

Range Index: 156 entries, 0 to 155	
Data columns (total 9 columns):	
Country or region	156 non-null object
Score	156 non-null int64
GDP per capita	156 non-null float64
Social support	156 non-null float64
Healty life expectancy	156 non-null float64
Freedom to make life choices	156 non-null float64
Generosity	156 non-null float64
Perceptions of corruption	156 non-null float64

Table-16: World Happiness Rank 2020 Data Summary

```
print(data6.columns)
print(data6.info())
print(data6.describe())
df6 = data6["Country name"]
dff6 = data6.value_counts()
```

We find that our 2020 data includes 153 countries, 16 features with non-blank values, and all other statistical information from the code output.

```
x6 = data6.iloc[:,4:].values
y6 = data6.iloc[:,2:3].values
```

They are assigned to the variable x and y as the output of the code, x as the input columns affecting the happiness score, and y as the resultant happiness score column.

```
from sklearn.model_selection import train_test_split
x_train6, x_test6, y_train6, y_test6 = train_test_split(x6, y6, test_size=0.33, random_state=0)
```

Test and training sets are created as code output.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train6 = sc.fit_transform(x_train6)
X_test6 = sc.fit_transform(x_test6)
Y_train6 = sc.fit_transform(y_train6)
Y_test6 = sc.fit_transform(y_test6)
```

Normalization is done as a code output.

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train6, y_train6)
print("b0: ", lr.intercept_)
print("other b: ", lr.coef_)
```

A machine learning model is created as a code output and weight factor values are calculated according to the model.

```
import statsmodels.regression.linear_model as sm
X6 = np.append(arr = np.ones((153,1)).astype(int), values=x6, axis=1)
r_ols6 = sm.OLS(endog = y6, exog = X6)
r6 = r_ols6.fit()
print(r6.summary())
```

R squared:	0.779
Adj. R squared:	0.770

Table-17: The Success of Machine Learning Method in World Happiness Ranking 2020 Data

Variables	P-Value
constant	0.000
GDP per capita	0.001
Social support	0.000
Healty life expectancy	0.002
Freedom to make life choices	0.000
Generosity	0.327
Perceptions of corruption	0.075

Table-18: The Success of Variables on World Happiness Ranking 2020 Data

Terrorism Report 2015

RangeIndex: 14963 entries, 0 to 14962	
Data columns (total 2 columns):	
year	14963 non-null int64
country	14963 non-null object

Table-19: Summary of Terrorism Report 2015 Data

```
print(data7.columns)
print(data7.info())
print(data7.describe())
df7 = data7["country"]
dff7 = df7.value_counts()
print(dff7)
```

From Table-19 and output of code, we understand that our 2015 Terrorism Report data is the year and countries with 14963 non-zero values.

Country	Terrorism Counts
Iraq	2750
Afghanistan	1928
Pakistan	1243
India	884
Philippines	721

Table-20: Examples of Terrorism Numbers Related to Terrorism Report 2015 Data

For example, countries are ranked according to the number of terrorisms. All values have been added as a new feature to our 2015 report model.

Terrorism Report 2016

RangeIndex: 13587 entries, 0 to 13586	
Data columns (total 2 columns):	
year	13587 non-null int64
country	13587 non-null object

Table-21: Summary of Terrorism Report 2016 Data

```
print(data8.columns)
print(data8.info())
print(data8.describe())
df8 = data8["country"]
dff8 = df8.value_counts()
print(dff8)
```

From Table-21 and output of code, we understand that our 2016 Terrorism Report data is the year and countries with 13587 non-zero values.

Country	Terrorism Counts
Iraq	3360
Afghanistan	1617
India	1025
Pakistan	864
Philippines	632
Somalia	602
Turkey	542

Table-22: Examples of Terrorism Numbers Related to Terrorism Report 2016 Data

For example, countries are ranked according to their terror count. All values have been added as a new feature to our 2016 report model.

Terrorism Report 2017

RangeIndex: 10900 entries, 0 to 10899	
Data columns (total 2 columns):	
year	10900 non-null int64
country	10900 non-null object

Table-23: Summary of Terrorism Report 2017 Data

From Table-23, we understand that our Terrorism Report 2017 data are countries with year and 10900 non-zero values.

```
print(data9.columns)
print(data9.info())
print(data9.describe())
df9 = data9["country"]
dff9 = df9.value_counts()
print(dff9)
```

Country	Terrorism Counts
Iraq	2466
Afghanistan	1414
India	966
Pakistan	719
Philippines	692
Somalia	614

Table-24: Examples of Terrorism Numbers Related to Terrorism Report 2017 Data

For example, countries are ranked according to the number of terrorists. All values have been added as a new feature to our 2017 report model.

6. RESULTS

In the 2015, 2016 and 2017 World Happiness Reports, we used the Global Terrorism Report to track which factor affects the World Happiness Report and how much, and the impact of terrorist incidents on this ranking. At the end of the analysis, we estimate the happiness score with logical random values using Machine Learning and found that they really work. We found the Multilinear Regression Machine Learning Method to be a good option for our model. We found that the values of terrorist incidents were not affected by happiness scores, but the effect increased in 2016. Since there is no Terrorism data for 2018, 2019 and 2020 data, only their own data were analyzed.

7. REFERENCES

- [1] Louise Millard (2011) Data Mining and Analysis of Global Happiness: A Machine Learning Approach
- [2] Natasha Jaques, Sara Taylor, Asaph Azaria, Asma Ghandeharioun, Akane Sano, Rosalind Picard (...) Predicting Students' Happiness from Physiology, Phone, Mobility, and Behavioral Data