# Report

## Part 1:

This report aims to develop multiple linear regression and prediction models to forecast the life expectancy of each country. If you've ever wondered about the factors contributing to a longer and healthier life, this report is for you. We seek to uncover which societal features correlate with the highest life expectancy among citizens. For instance, does living in countries with higher GDP per capita, where individuals are generally more satisfied with their lives, translate to a longer life expectancy?

Utilizing Machine Learning (ML) techniques, we aim to pinpoint the target attribute of life expectancy based on various relevant features of countries. With access to data from 2015, we can construct models capable of predicting life expectancy. Such accurate prediction models can be invaluable not only for countries but also for individuals. By understanding factors like happiness and wealth, individuals may potentially forecast their own life expectancy.

```
Coefficients:
[ 1.56249089 13.58610771]
Intercept:
51.98207541232717
```

**Prediction = 1.562 * Happiness Score + 13.6 * GDP per Capita + 52.0**

The disparity in weights between GDP per Capita and the Happiness Score suggests that GDP per Capita has a more significant influence on predicting Life Expectancy compared to the Happiness Score. Specifically, an increase in GDP per Capita by one unit is associated with a greater change in Life Expectancy than a corresponding increase in the Happiness Score by one unit.

To illustrate, if we hold the GDP per Capita constant and increase the Happiness Score by one unit, the Life Expectancy rises from 83.29 to 84.86. Conversely, if we maintain the Happiness Score and increase GDP per Capita by one unit, the Life Expectancy surges from 83.29 to 96.88. This demonstrates the considerable impact of GDP per Capita on predicting Life Expectancy, overshadowing the influence of the Happiness Score.

```
----- Sample case -----
Happiness Score: 7
GDP per Capita: 1.5
Predicted length of Life Expectancy at Birth: [83.29867318]
----- Sample case -----
Happiness Score: 7
GDP per Capita: 2.5
Predicted length of Life Expectancy at Birth: [96.88478089]
----- Sample case -----
Happiness Score: 8
GDP per Capita: 1.5
Predicted length of Life Expectancy at Birth: [84.86116406]
```

## Part 2:

## Tools and Packages used for coding:

```
import pandas as pd
from math import sqrt
from sklearn import linear_model
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
```

## Code for First Predictive model:

### First, reading the dataset and creating pandas data frame:

```
df = pd.read_csv('final_data1.csv', thousands=',')
```

### Converting String to Float:

```
df['GNI per Capita'] = pd.to_numeric(df['GNI per Capita'])
```

### Splitting features and target variables:

```
x = df.values[:, 1:3]
y = df.values[:, 0]
```

### Setting Training and Testing sets with test size of 30% and train size of 70%

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

### Fitting Linear Regression and Building model using training sets:

```
regr = linear_model.LinearRegression().fit(X_train, y_train)
```

### Predicting the test set results:

```
y_pred = regr.predict(X_test)
```

### Comparing results:

```
df_pred = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(df_pred)
```

### Prediction model Evaluation:

```
mse = metrics.mean_squared_error(y_test, y_pred)
print('Root mean squared error (RMSE):', sqrt(mse))
print('R-squared score:', metrics.r2_score(y_test, y_pred))
```

## Code for Second Predictive Model:

### Splitting features and target variables:

### Choosing different sets of features:

```
x = df.values[:, 3:]
y = df.values[:, 0]
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

### Feature scaling: (all codes from this point is same as previous model)

**Note: I used feature scaling because there was huge difference between scaling of GNI per capita and Freedom.**

```
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

```
regr = linear_model.LinearRegression().fit(X_train, y_train)
print('Coefficients:')
print(regr.coef_)
print('Intercept:')
print(regr.intercept_)
y_pred = regr.predict(X_test)
df_pred = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(df_pred)
mse = metrics.mean_squared_error(y_test, y_pred)
print('Root mean squared error (RMSE):', sqrt(mse))
print('R-squared score:', metrics.r2_score(y_test, y_pred))
```

For both predictive models, Linear Regression was chosen due to the dataset consisting solely of quantitative attributes that are related to each other. Before splitting the features and target variables, I assessed the linearity assumption between each of the four possible features and the target variable. (Refer to Table 1 for details.)

Table 1 indicates that the Happiness Score and GDP per Capita meet the linearity assumption well. Conversely, GNI per Capita and Freedom do not adhere to this assumption. Additionally, examining the residual plots (see Figures 5, 6, and 8), we observe a random dispersion across the horizontal line, indicating compliance with linearity assumptions. However, Figure 7 does not exhibit this random dispersion, suggesting a deviation from the linearity assumption.

Based on these assessments, I determined the features for Prediction Models 1 and 2.

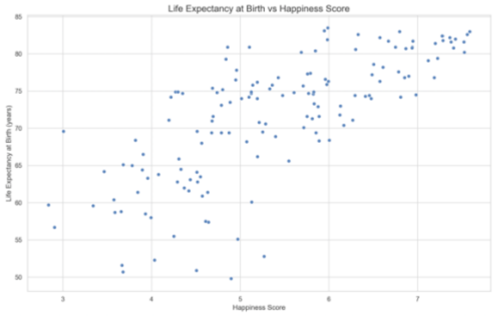
<table>
<tr><td colspan="2">Checking Linearity</td></tr>
<tr><td colspan="2">Scatter plot</td></tr>
<tr><td>

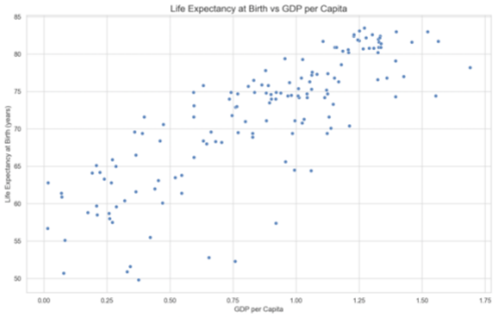
*Figure 1 Scatterplot*</td><td>

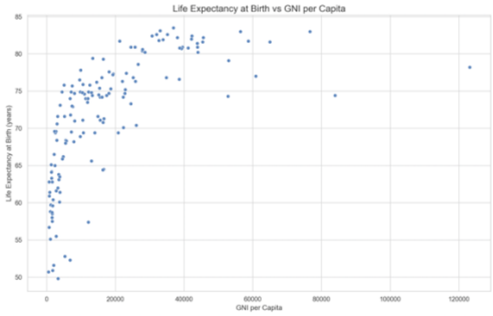
*Figure 2 Scatterplot*</td></tr>
<tr><td>

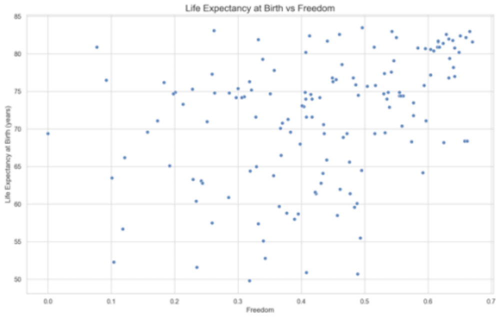
*Figure 3 Scatterplot*</td><td>

*Figure 4 Scatterplot*</td></tr>
</table>

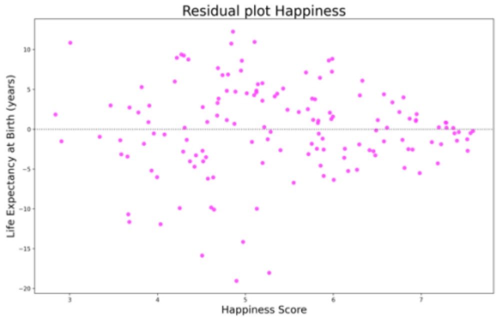
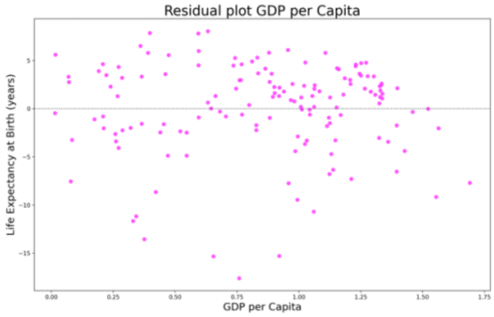
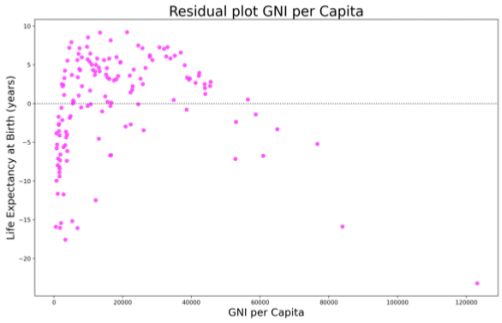
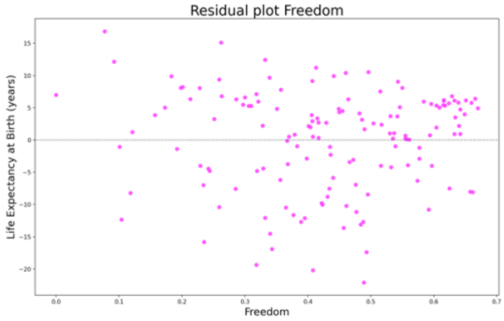
| Residual plot | |
|---|---|
| <br> <br>*Figure 5 Residual plot* | <br> <br>*Figure 6 Residual* |
| <br> <br>*Figure 7 Residual plot* | <br> <br>*Figure 8 Residual plot* |

*Table 1. Linearity Assumption*

After assigning the feature variables and target variable, I divided the dataset into training and testing sets. For the first model, 30% of the data were reserved for testing, while 70% were allocated for the training set. I experimented with different test sizes and evaluated the performance using metrics such as R-squared (R2) and Root Mean Squared Error (RMSE). It was observed that the best performance was achieved when the test size was set to 30%.

However, for the second model, a test size of 40% yielded better results. Therefore, I adjusted the test size to 40% and allocated 60% of the data for the training set in the second model. This optimization was made based on the performance metrics to ensure the models' accuracy and generalization capability.

Evaluation:

Evaluating the predictive model is an essential step in the process. In this report, I specifically utilized Root Mean Square Error (RMSE) and R-squared scores for evaluation. RMSE measures the average magnitude of the error obtained by the model, providing insight into the accuracy of the predictions. On the other hand, R-squared represents the squared correlation between observed and predicted values, indicating how well the model fits the data.

Lower RMSE values and higher R-squared scores indicate better predictive performance. These metrics are widely used due to their simplicity and ease of interpretation. However, it's important to note that R-squared alone may not detect overfitting of the model, which is a limitation.

For the first predictive model:

- Root mean squared error (RMSE): 4.51
- R-squared score: 0.74

For the second predictive model:

- Root mean squared error (RMSE): 6.95
- R-squared score: 0.37

These results provide insights into the accuracy and performance of each predictive model, aiding in the assessment of their effectiveness in predicting life expectancy based on the chosen features.

Conclusion and Decision:

The RMSE for the first model was 4.51 years, which, considering the human age range of 0-122 years, is relatively small. Additionally, the R-squared value of 0.74 indicates a high level of accuracy for this model. While these results suggest that the first predictive model was not the perfect fit, it still demonstrated accuracy in predicting life expectancy.

In contrast, the second model exhibited an RMSE of 6.95 years and an R-squared value of 0.37. Although the RMSE was only slightly higher than that of the first model, the significantly lower R-squared value indicates that the second model was not as accurate in predicting life expectancy.

Furthermore, analysis of Table 1 revealed that features such as GNI per Capita and Freedom in the second predictive model were not linearly related to the target variable, Life Expectancy. For instance, Figure 3 exhibited a shape more akin to a polynomial, while Figure 4 showed no discernible association. On the other hand, Figures 1 and 2 displayed clear positive linearity.

Considering these factors, the first predictive model, which predicts each country's life expectancy at birth based on its Happiness Score and GDP per Capita, is deemed superior to the second model.

## Reference list:

Moreno, A. (2019). Simple and multiple linear regression with Python. Retrieved 20 November 2020, from https://towardsdatascience.com/simple-and-multiple-linear-regression-with-python-c9ab422ec29c

Albert Einstein College of Medicine. (2016, October 5). Maximum human lifespan has already been reached. ScienceDaily. Retrieved November 20, 2020 from www.sciencedaily.com/releases/2016/10/161005132823.htm