Iteration 4 - BDAS

Steven Schmidt (ssch162)
INFOSYS722

University of Auckland
Auckland, NZ

Table of Contents

1. Business Understanding:

1.1. Business Situation.

The United Nations (UN) was established in 1945 and is made up of 192 member countries whose purpose is defined by a guiding charter. **(Nations, UN Charter, 1945)**

Driven by its charter, the **UN** has an active program titled "**Sustainable Development Goals**" which they feel will address various global issues facing mankind today. The program has 17 goals aimed at addressing global challenges including poverty, inequality, climate change, environmental degradation, peace, and justice. **(Nations, Home, 2024)**

A full list of goals can be viewed at https://www.un.org/sustainabledevelopment/sustainable-development-goals/

One of the **UN Sustainable Development Goals** is the "**Goal 2**, **Zero Hunger"** program. It aims to solve world hunger by **2030** by addressing areas that affect hunger such as poverty, inequality, climate change, conflict, and building resources to grow food within the affected countries.

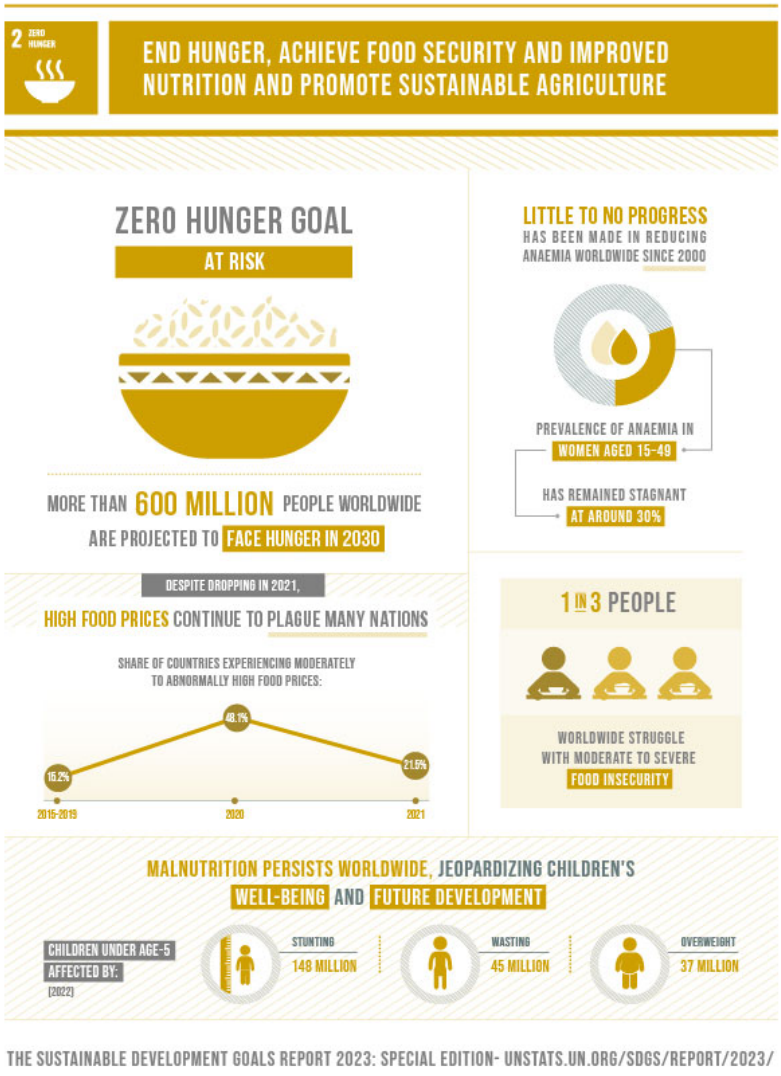

Figure 1

In 2022, 821 million people (9.2%) faced severe starvation around the world.

In 2017, 2 billion people faced unreliable food supply and nutrition issues regularly. This grew to 2.4 billion in 2022. **(Nations, Goal 2: Zero Hunger, n.d.)** Figure 1 graphically details this. **(Nations, Goals, 2024)**

The UN has declared that they plan to solve this problem and achieve this goal by 2030.

The primary question I have is "**Is the date 2030 achievable for Goal 2?**".

This paper attempts to answer this question using Data Mining techniques to analyze information the UN uses to track progress.

1.2. The Problem.

For this project, the first problem we will face is identifying a suitable and reliable data source that can be analyzed to answer the question raised in 1.1.

The program has been running since 2000, over 24 years. As such, there will already be various means of tracking progress in place. Reporting is something that the UN pays a lot of attention to. Finding the right information will be the main issue.

Then identifying and building a model that will allow me to create future projections will be the next challenge. If the data is in a suitable format, it will make identifying and building easier.

1.3. Data Mining Objective.

This paper asks one primary question and, depending on the answer, two supplementary questions.

1. **Is the stated date, 2030, achievable for Goal 2 based on historical data?**

   If **2030** is **not** achievable –

2. **Why isn't 2030 achievable?**
3. **What is the earliest date the goal could be completed?**

Questions 2 & 3 will only need to be answered if **2030** is not achievable.

I feel all the questions can be answered as the UN is already tracking the progress of the goal so historical data will be available in some format. The challenge will be identifying the right data to analyze. Using prediction modeling and historical data, we can with some certainty, predict whether the UN will reach its goal by 2030.

The main objectives will be:

- Identify a **reliable source** of data.
- Look for **patterns** during the **mining** process that will allow us to gain **insight** into the progress of **Zero Hunger**.
- Using **Regression**, identify if **2030** is an achievable date based on the sourced data.
- If 2030 is not achievable,
    - Report on why 2030 was not achievable.
    - Using the same data, **identify** the new completion date.

Ultimately, the **best** outcome is that the **projections** indicate the **UN** is still **on course** to meet the target date of **2030** or within the decade following.

Failing this, **the next best outcome** would be to identify a **new target date** and with constant **reviewing** and continuous **improvement**, identify a **mechanism(s)** to **pair back** that **date** over time.

1.4. Assessment.

1.4.1. Resource Inventory.

- Ubuntu or equivalent Linux Server
  - Python3 installed.
  - Pip installed. (Including required modules listed in section 7.2)
  - Jupyter installed.
- Historical information on hunger in a suitable data format
- Computing resources to develop the Python scripts and run them to get results.

1.4.2. Requirements, Assumptions, and Constraints.

- **Requirement**: Computer with enough processing resources
- **Requirement**: A clean data source
- **Constraint**: My knowledge of data analysis and interpretation.
  - This is my first real delve into data analysis.
  - Some terms are completely foreign.
  - Some mathematical terms in the modeling are foreign.
  - Data interpretation knowledge is limited.
  - The learning curve is steep.

1.4.3. Risk Analysis.

Risk	Mitigation
historical data sets not found	Identify suitable sources and create the required data sets. If required, prepare a suitable data source based on other associated reliable data sources.
The data set is not complete	If the dataset has most of the required information but is not complete. Extrapolate and pad out the information but ensure it falls between the valid information markers.
Failure of computing hardware during development	Create a model on a virtual computer that is replicated at other locations
Unable to interpret information	Access external references and information sites that have the models I am using to gain more insight into the algorithms and outcomes.

Table 1

1.4.4. Cost/Benefit Analysis.

Cost	Benefit
Estimated $267bn annually	Increased productivity, improved health outcomes, reduced poverty and enhanced overall well-being for individuals and communities within the benefiting countries
Supporting counties contributing to funding and developing hunger elimination programs	Future generations of developing countries produce citizens that can contribute on a local and world basis.

Table 2

In March 2023, Greyhound International stated:

“Ending world hunger would require significant financial resources. According to estimates from the United Nations Food and Agriculture Organization (FAO), an annual investment of around $267 billion is needed to achieve Zero Hunger by 2030”. (Bezawit Beyene Chichaibelu, 2024)

A cost-benefit analysis of hunger is a very hard thing to perform from a humanitarian point of view. Although careful financial management should be taken, hunger should be eradicated no matter what the cost is. No one should go hungry, and mankind’s existence depends on it.

1.5. Plan.

If we can find a source that has historical measurement data on hunger levels per country known as the **Global Hunger Index** value, (**GHI**) we can attempt to identify if the UN target of **2030** is achievable by calculating the **Mean** value of the Global Hunger Index value on a time basis and chart this value using a predictive model. The model can provide the data to plot out a timeline for both current and predicted trend lines to identify when Zero Hunger will possibly be achieved.

1.5.1. Process.

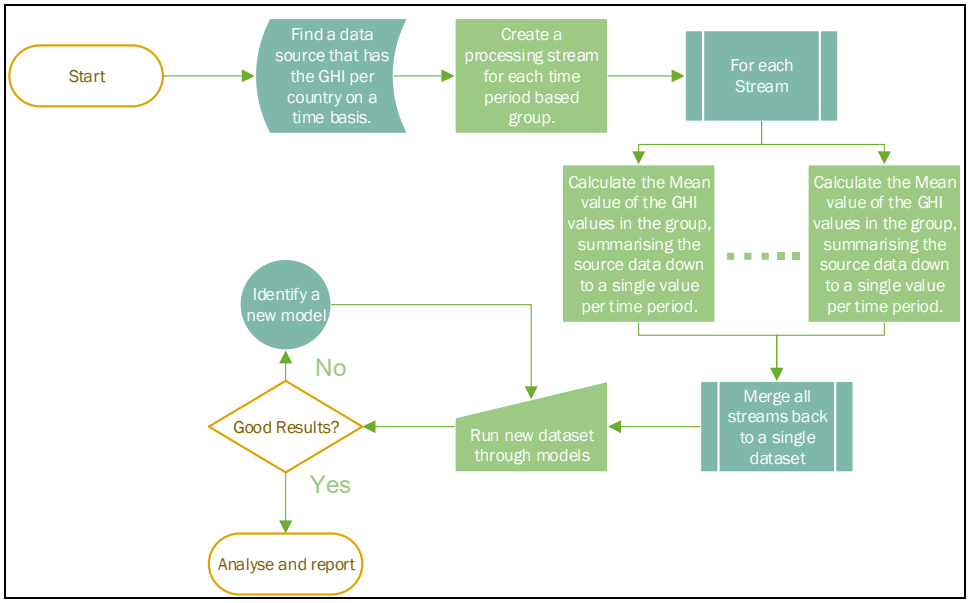

Figure 2

1.5.2. Task List and Gantt chart.

Phase ID	Task Description
1	Identify suitable Data source
	Internet search
	Review identified sources
	Decide on the source and retrieve
2	Data Understanding
	Load select data
	Review data
	Identify target data
	Transform data
	Produce a new data set
3	Model Build
4	Interpret results & Findings

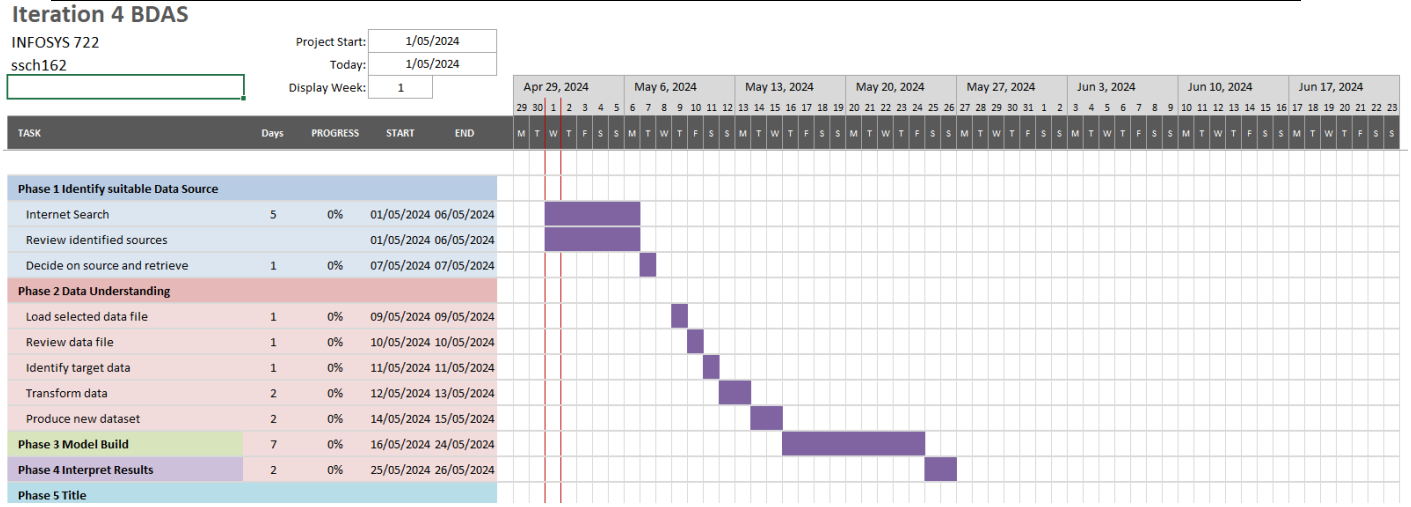

Figure 3

2. Data understanding

2.1 Initial data collection.

A data source has been identified. The website Kaggle has 4 downloadable CSV files that contain historical information about child hunger and a Global Hunger Index rating known as GHI. **(Chauhan, 2022)**

The historical information has been collected since the year 2000 and is a rating for countries that have hunger problems, and that the UN is trying to help. The rating provides a method for the UN to monitor if their programs are making a difference. The information is gathered from programs that have been implemented for child hunger. We will utilize the **GHI** value as a basis for analyzing and projecting in this paper.

Original data files used in this project are available for download here:

https://www.kaggle.com/datasets/whenamancodes/the-global-hunger-index

2.2 Describe the data

Global Hunger Index (GHI) is a calculation based on a series of other studies into various health conditions related to child hunger around the world. (WHO, 2024)

Ultimately 4 areas of child health are looked at in more detail:

- **Undernourishment**: the proportion of undernourished people as a percentage of the population (reflecting the share of the population with insufficient caloric intake);
- **Child wasting**: the proportion of children under the age of five who suffer from wasting (low weight for their height, reflecting acute undernutrition);
- **Child stunting**: the proportion of children under the age of five who suffer from stunting (low height for their age, reflecting chronic undernutrition); and
- **Child mortality**: the mortality rate of children under the age of five (partially reflecting the fatal synergy of inadequate nutrition and unhealthy environments).

From this, a rating for each is derived. Each derived **rating** is added together to produce the **GHI** for a country. The **GHI** is a single value that, at best, helps to monitor world hunger. (Index, 2024)

I have identified a suitable dataset which contains 471 observations with the following attributes:

- Country (Entity)
- Country Code (Code)
- Date (Year)
- GHI (Global Hunger Index)
- 411773-annotations

```
[3]: # Print the schema of the DataFrame. Get a view of the datasource structure
     data.printSchema()

root
 |-- Entity: string (nullable = true)
 |-- Code: string (nullable = true)
 |-- Year: integer (nullable = true)
 |-- Global Hunger Index (2021): double (nullable = true)
 |-- 411773-annotations: string (nullable = true)
```

Figure 4

2.3 Explore the data

The **Global Hunger Index** is an attribute that is of interest to us. This calculation is mentioned in section 2.2 above and rates each country that appears in the original data.

Each **Observation** is based on the **Country, Year,** and an associated **GHI** value.

i.e. Afghanistan has 4 observations, one for each year information was collected.

```
[4]: # Dump the first 10 instances of the datasource
     data.head(10)
```

```
[4]: [Row(Entity='Afghanistan', Code='AFG', Year=2000, Global Hunger Index (2021)=50.9, 411773-annotations=None),
      Row(Entity='Afghanistan', Code='AFG', Year=2006, Global Hunger Index (2021)=42.7, 411773-annotations=None),
      Row(Entity='Afghanistan', Code='AFG', Year=2012, Global Hunger Index (2021)=34.3, 411773-annotations=None),
      Row(Entity='Afghanistan', Code='AFG', Year=2021, Global Hunger Index (2021)=28.3, 411773-annotations=None),
      Row(Entity='Albania', Code='ALB', Year=2000, Global Hunger Index (2021)=20.7, 411773-annotations=None),
      Row(Entity='Albania', Code='ALB', Year=2006, Global Hunger Index (2021)=15.9, 411773-annotations=None),
      Row(Entity='Albania', Code='ALB', Year=2012, Global Hunger Index (2021)=8.8, 411773-annotations=None),
      Row(Entity='Albania', Code='ALB', Year=2021, Global Hunger Index (2021)=6.2, 411773-annotations=None),
      Row(Entity='Algeria', Code='DZA', Year=2000, Global Hunger Index (2021)=14.5, 411773-annotations=None),
      Row(Entity='Algeria', Code='DZA', Year=2006, Global Hunger Index (2021)=11.7, 411773-annotations=None)]
```

Figure 5

The downloaded data does not have an even historical flow, but we can use it as a starting point as it has a start date and an end date with entries in between.

i.e. There is a gap of **6 years** between the first **3 entries 2000, 2006 & 2012**, and a **9-year gap** between the last 2, **2012 & 2021**.

```
[30]: RowCounts = data.groupBy("Year").count()
      ChkRowCounts = RowCounts.orderBy("Year")
      ChkRowCounts.show()
```

```
+----+-----+
|Year|count|
+----+-----+
|2000|  112|
|2006|  115|
|2012|  116|
|2021|  128|
+----+-----+
```

There is a reason that was identified, while processing and exploring the data, for the uneven timeline. It helps support the conclusion, which we will discuss later in the paper.

The initial identified data has no target data and as such, is classified as "**Unlabeled**". During **Data Transformation**, we will build a target variable by summarizing observations grouped by **Year** and call this the **Mean GHI**. The **Mean GHI** value will be a **Mean** calculation of all the **GHI** values for each year. From these calculations, we will have a **Year** variable **(X)** and a **Mean** value **(Y)** for each year.

In the previous iteration, due to the way SPSS handled dates, I added a day & month (01-01) to the source attribute **Year** to fulfill the requirements of the date routines within SPSS. This was not required with Python, as the data functions handle this without a problem. This is the way.

2.4 Verify the Data.

The calculation of the GHI for each country looks at 3 main components from subprograms within Zero Hunger.

1. Undernourishment
2. Child Stunting
3. Child Wasting

The source data for these 3 components that were used to calculate the GHI is also available at Kaggle (Chauhan, 2022) and available for download from the same page listed in section 2.1. Author Aman Chauhan details how the supporting data sets were used in the calculations.

At this stage, this data is superfluous to needs but will be kept for possible validation purposes if required later. Once we have transformed the data, we will have the information ready to perform predictive modeling against.

We can validate the **Mean GHI** by using the detail generated by the "**.describe()**" function.

Taking the **Mean GHI** values and calculating a **Mean** value from these should be close to the **Mean** value we see in the "**.describe()**" function call.

```
[16]: # Get a breakdown on info contained in dataframe
      data.toPandas().describe()
```

[16]:

	Year	Global Hunger Index (2021)
count	471.000000	471.000000
mean	2010.127389	19.824628
std	7.852951	13.449570
min	2000.000000	2.500000
25%	2006.000000	8.800000
50%	2012.000000	17.000000
75%	2021.000000	28.550000
max	2021.000000	65.100000

Figure 6

This will give us an indication if the calculations are close.

```
[30]: RowCounts = data.groupBy("Year").count()
      ChkRowCounts = RowCounts.orderBy("Year")
      ChkRowCounts.show()
```

```
+----+-----+
|Year|count|
+----+-----+
|2000|  112|
|2006|  115|
|2012|  116|
|2021|  128|
+----+-----+
```

Also, the instance counts are not uniform over the years. This is due to new countries being added to the program after it had started.

3. Data Preparation

3.1 Selecting the Data.

After I spent some time reviewing it in its raw format, I identified the following attributes that can be ignored and have been removed from the input stream.

Fields

- **Code** is not required as it is a shorthand representation of the Country.
- **411773-annotations** are not required as this is effectively a “notes” field.

04-DT

Data Transform - Process data source to produce Mean values for each year.

```
[6]: #dRemove unused columns. Not nessary but keeps data clean.
     data = data.drop('Code')
     data = data.drop('411773-annotations')
     data.head(4)
```

```
[6]: [Row(Entity='Afghanistan', Year=2000, Global Hunger Index (2021)=50.9),
      Row(Entity='Afghanistan', Year=2006, Global Hunger Index (2021)=42.7),
      Row(Entity='Afghanistan', Year=2012, Global Hunger Index (2021)=34.3),
      Row(Entity='Afghanistan', Year=2021, Global Hunger Index (2021)=28.3)]
```

Figure 7

This removed any “NaN” in the raw data as they were only contained in these variables.

3.2 Clean the Data

The Attribute **Year** and **Global Hunger Index** are the main required attributes for my prediction models.

The attribute **Entity** may be useful later if we look to explore the information deeper, perhaps looking at and predicting individual countries but for now, this is outside the scope of this work.

3.3 Construct the Data.

Now that we have the raw data, we can start to calculate the mean GHI value for each year.

With SPSS a complete logic flow was created to generate the Mean values for each year.

With Jupyter and Python, the ability to “stack” function calls mean we can complete the calculation within one statement. Also, renaming the attribute to something meaningful at the same time.

```
[7]: # Calculate the Mean GHI for each year.
df_raw = data.groupBy("Year").mean("Global Hunger Index (2021)")
df_raw = df_raw.withColumnRenamed("avg(Global Hunger Index (2021))", "MeanGHIraw")
df_raw.show()
```

```
+----+-----------------+
|Year|       MeanGHIraw|
+----+-----------------+
|2006|21.04347826086957|
|2012|17.50862068965517|
|2000|24.43928571428571|
|2021|        16.790625|
+----+-----------------+
```

Figure 8

The above statement isolates the variable “**Year**” which is the timescale and the variable “**Global Hunger Index (2021)**” which is the variable of the mean calculation that will take place.

- This is fed into “**.groupby(“Year”)”** function which sorts the information into year order (Feature)
- This is then fed into the mean function which calculates the **Mean GHI** value for each year. (Target)
- The output from this calculation can be seen in **Figure 8** above.
- We then set up the data so that it can be utilized in the same fashion as a relational database.

```
[8]: # Clean up MeanGHI and prep data for SQL
df_raw.summary().show()
df_raw.createOrReplaceTempView("df_Raw_Table")
df = spark.sql("select Year, round(MeanGHIraw, 4) as MeanGHI from df_Raw_Table")
df
df.createOrReplaceTempView("df_Table")
```

```
+-------+-----------------+------------------+
|summary|             Year|        MeanGHIraw|
+-------+-----------------+------------------+
|  count|                4|                 4|
|   mean|          2009.75| 19.94550241620261|
| stddev|8.958236433584458|3.5256811798596144|
|    min|             2000|         16.790625|
|    25%|             2000|         16.790625|
|    50%|             2006| 17.50862068965517|
|    75%|             2012| 21.04347826086957|
|    max|             2021| 24.43928571428571|
+-------+-----------------+------------------+
```

Figure 9

3.4 Integrate various Data Sources.

Within the stacked function, the ".**groupby"** function allows us to easily isolate observations for each **Year** giving a data stream for each date in the file regardless of **Country**. Four summary dates were identified, 2000, 2006, 2012 & 2021.

3.1 Format the Data.

I prepped the data for SQL access. This provides a mechanism for easily querying and reformatting of the data.

The "**round**" function in the SQL statement rounds the **Mean** to 4 decimal places producing a uniform calculation for each year.

At the end of the processing, we will have **4 Mean GHI** (target) summarized values, one for each **Year** (feature) which can then be fed into the **Data Transformation** process.

[30]:
```
# Clean up MeanGHI and prep data for SQL
df_raw.summary().show()
df_raw.createOrReplaceTempView("df_Raw_Table")
df = spark.sql("select Year, round(MeanGHIraw, 4) as MeanGHI from df_Raw_Table")
df
df.createOrReplaceTempView("df_Table")
```

```
+-------+-----------------+------------------+
|summary|             Year|        MeanGHIraw|
+-------+-----------------+------------------+
|  count|                4|                 4|
|   mean|          2009.75| 19.94550241620261|
| stddev|8.958236433584458|3.5256811798596144|
|    min|             2000|         16.790625|
|    25%|             2000|         16.790625|
|    50%|             2006| 17.50862068965517|
|    75%|             2012| 21.04347826086957|
|    max|             2021| 24.43928571428571|
+-------+-----------------+------------------+
```

4. Data Transformation(s)

4.1 Reduce the Data.

At this point, we already have the required information reduced and ready for prediction modeling.

4.2 Project the Data.

```
[34]: ## Data Graphs - Actual MeanGHI
      ## Plot Initial MeanGHI data
      df1=spark.sql("Select * from df_Table order by Year")
      df1.show()
      pdf1=df1.toPandas()
      pdf1.plot(kind='line', x='Year',y='MeanGHI', linestyle="solid",
                marker="o", color="blue", title="Actual MeanGHI", xlabel='Year', ylabel='MeanGHI')
      actual = pdf1
```

```
+----+-------+
|Year|MeanGHI|
+----+-------+
|2000|24.4393|
|2006|21.0435|
|2012|17.5086|
|2021|16.7906|
+----+-------+
```

Figure 10

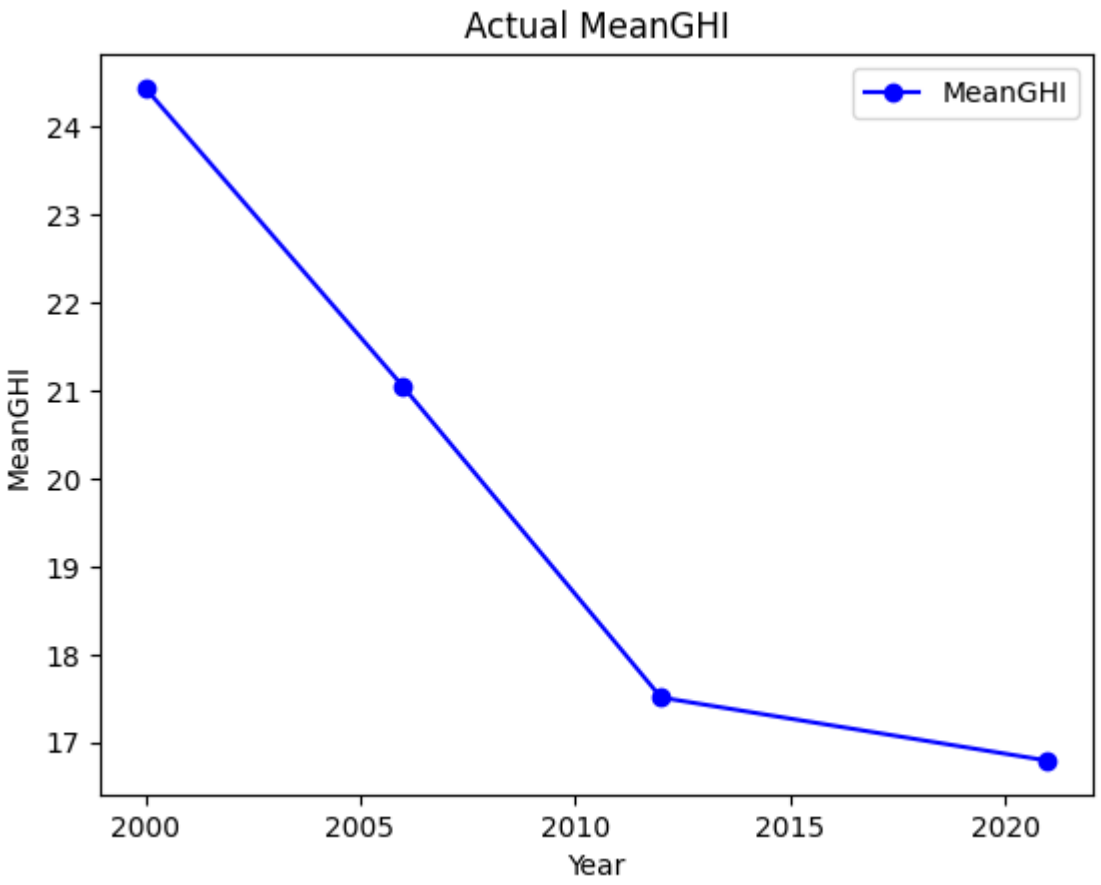

Figure 11

5. Data-Mining Method(s) employed.

5.1 Data Mining Objectives v Methods.

The objective as listed in Section 1.3 is to be able to predict whether the UN will meet its targeted goal of 2030 for Zero Hunger.

After examining the data deeper, it was apparent that there was no direct target that could be used within the supplied data, meaning the selected data can be classed as "**Unsupervised**".

I found observations that contained **Year-based** GHI readings for each **Country,** but I needed an overall total (target) for each year. Within the **Data Transformation**, I was able to create a **summarized** set of observations using the **Clustering** technique with **Year** as the grouping, calculating a **Mean GHI** for each selected **Year**. From this, the Year becomes the Feature and **Mean GHI** becomes the target attribute.

Once the data had been transformed and summarized with a target, it could be considered "**Supervised**" and as such, **Supervised** tasks, primarily **Linea Regression and Time Series Forecasting**, can then be considered.

Mean GHI was chosen as the **target** for this report. Being able to chart the Mean GHI for the entire program gives an indicator of how successful the program is overall. Some **countries** will achieve success **individually** and this is to be **applauded**. Some countries were included as the program progressed so have incomplete data. Looking at them **individually** can cause **abnormalities** with the data but using the **Mean GHI** means the overall program is **analyzed**, even with **changing conditions** allowing for a better validation of the entire program.

5.2 Appropriate Methods based on 5.1.

Linea Regression

LineaRegression is a class in the sklearn library that provides API functions that allow for predicting and extending a Linea set of numeric values.

i.e. We have a series of values associated with the Global Hunger Index rating (GHI) that have been created regularly. Based on the pattern in the values, we can build a model of how the values will look in **N steps** time.

Time-Based Series

TBS expands on **Regression** with the numeric values structured in a time-based sequence and when specifying **N steps**, it will represent a time interval.

i.e. We take the time each reading is taken and use this as an index when calculating Regression allowing us to get a clearer view over time which we can then apply when making forward predictions.

These two methods will allow us to make projections on the data.

6. Data-mining algorithm(s) employed.

6.1 Exploratory Analysis.

The results of the model run returned Figure 9 and gave a view of Planned v actual results.

Using **LinearRegression** and **time plot graph** and feeding it the **Date** & **Mean** attributes, provided the first indication that the UN is not going to meet the 2030 goal.

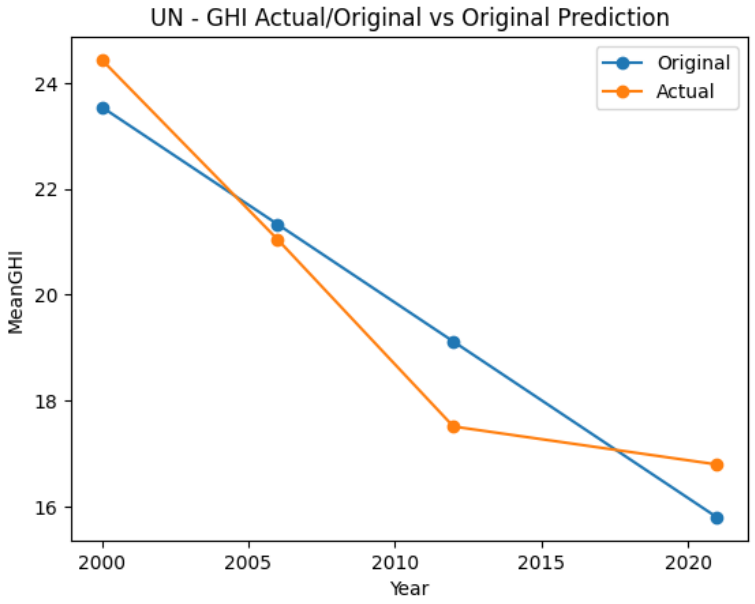

Figure 12

A section that stands out, as highlighted in Figure 10, shows in the period between 2012 & 2021 something occurred that affected the timeline directly, changing the trajectory of the trend line producing an upswing. This is reflected in both trend lines. More on this in the Interpretation Section.

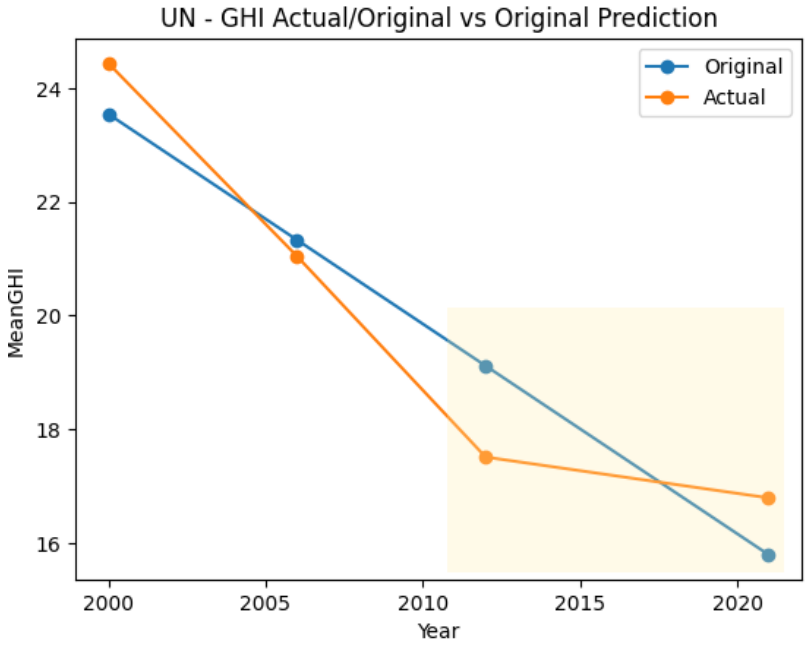

Figure 13

6.2 Select Data Mining Algorithms.

The algorithm I chose to use is **LinearRegression** located in the **sklearn** API library.

Scikit-learn (scikit, 2024) defines the algorithm as

"**LinearRegression"** fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation."

Simply put, the algorithm uses the assumption that the relationship between the independent variables (consider timescale) and the dependent variable (our target) is linear and can make informed calculations based on this assumption.

6.3 Build/Select Model.

```
[10]: ## Data Graphs - 1. UN - Original GHI prediction Values only
      ## Plot Initial MeanGHI data minus the last reading after COVID
      df1b=spark.sql("Select * from df_Table order by Year limit 3")
      #df1b.show()
      pdf1=df1b.toPandas()

      X = np.array(df1b.select("Year").collect())
      Y = np.array(df1b.select("MeanGHI").collect())

      reg1 = LinearRegression().fit(X, Y)
      print('-------------------------------------------------')
      print('Model Run - 1. Original GHI prediction - Outputs')
      print(f'R2 score: {reg1.score(X, Y)}')
      print(f'Coefficients: {reg1.coef_}')
      print(f'Intercept: {reg1.intercept_}')
      print('-------------------------------------------------')

      # Kept adding a year on and rerunning until the value reached/past Zero.
      predict_years = [ [2000], [2006], [2012], [2021] ]
      future = reg1.predict(np.array(predict_years))
```

Figure 14

I have chosen to Predictive & Regression modeling and in particular **LinearRegression**

LinearRegression is the primary method used to perform predictions.

Figure 14 details a piece of code that builds a class that takes 2 attributes, a feature and a target, listed below to implement the **Linear Regression** method returning requested future predictions.

.fit – Builds the **LinearRegression** class and loads the linear model with the attributes. Training data can be loaded via this API call as well.

.predict – The attributes passed are future dates that the prediction is to take place on. The model can take the information loaded in **.fit** and run the prediction(s) based on the time scale parameters passed in the API call returning the corresponding future values of the target. (In this case **Mean GHI**).

With regression, the output is a continuous flow based on specified time scales. This gives the ability to predict the values moving forward.

Different sets of data are built based on the **Mean GHI** calculation. These datasets are used with the same base template for the API call to produce different calculations, information, and graphs.

7. Data-Mining

7.1 Test Designs.

Sectional Testing was performed as the model was being developed.

No separate test data or partitions were created or used.

As the final dataset was very small, I could use it directly as the test dataset. This does not allow for the comparison of results from testing and training runs.

7.2 Conduct Data Mining.

To run the model the following steps must be followed.

Note: The following modules may need to be installed if you don't already have them.

Enter each line at a command prompt to initiate pip and install the desired module.

```
pip install findspark
pip install pyspark
pip install numpy
pip install matplotlib
pip install pandas
pip install wrapt
pip install pyarrow
pip install scikit-learn
```

1. Login using the username **ubuntu**.
2. Start **Jupyter**
    a. Enter "**jupyter notebook"** at the prompt and press enter. The following will be displayed.

```
ubuntu@ip-172-31-30-236:~/INFOSYS722-I4-BDAS-ssch162$ jupyter notebook
[I 05:35:31.213 NotebookApp] Serving notebooks from local directory: /home/ubuntu/INFOSYS722-I4-BDAS-ssch162
[I 05:35:31.213 NotebookApp] Jupyter Notebook 6.4.11 is running at:
[I 05:35:31.213 NotebookApp] https://ip-172-31-30-236:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
[I 05:35:31.213 NotebookApp]  or https://127.0.0.1:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
[I 05:35:31.213 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 05:35:31.218 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/ubuntu/.local/share/jupyter/runtime/nbserver-2472-open.html
    Or copy and paste one of these URLs:
        https://ip-172-31-30-236:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
     or https://127.0.0.1:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
```

3. In the bottom left-hand corner of the screen, something resembling the graphic will be displayed. Take note of the public IP address (**PublicIP**)

4. Highlight the line indicated below and paste it into a browser URL.

```
ubuntu@ip-172-31-30-236:~/INFOSYS722-I4-BDAS-ssch162$ jupyter notebook
[I 05:35:31.213 NotebookApp] Serving notebooks from local directory: /home/ubuntu/INFOSYS722-I4-BDAS-ssch162
[I 05:35:31.213 NotebookApp] Jupyter Notebook 6.4.11 is running at:
[I 05:35:31.213 NotebookApp] https://ip-172-31-30-236:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
[I 05:35:31.213 NotebookApp]  or https://127.0.0.1:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
[I 05:35:31.213 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 05:35:31.218 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/ubuntu/.local/share/jupyter/runtime/nbserver-2472-open.html
    Or copy and paste one of these URLs:
        https://ip-172-31-30-236:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
     or https://127.0.0.1:8888/?token=6915f5a26b2ac76c97d3566555a6fff0787eb21dea17cb5d
```

5. Swap the publicIP noted in step3 for the IP address in the highlighted string and enter

Figure 15

6. This will open jupyter up in the browser.
    a. Navigate to the subdirectory "INFOSYS722-I4-BDAS-ssch162" if not already there
7. Double-click on the filename "**Iteration 4 BDAS.ipynb**" (See Figure 14)

Figure 16

8. Click the menu item “Kernel” followed by “Restart Kernel and Clear Outputs of all Cells”

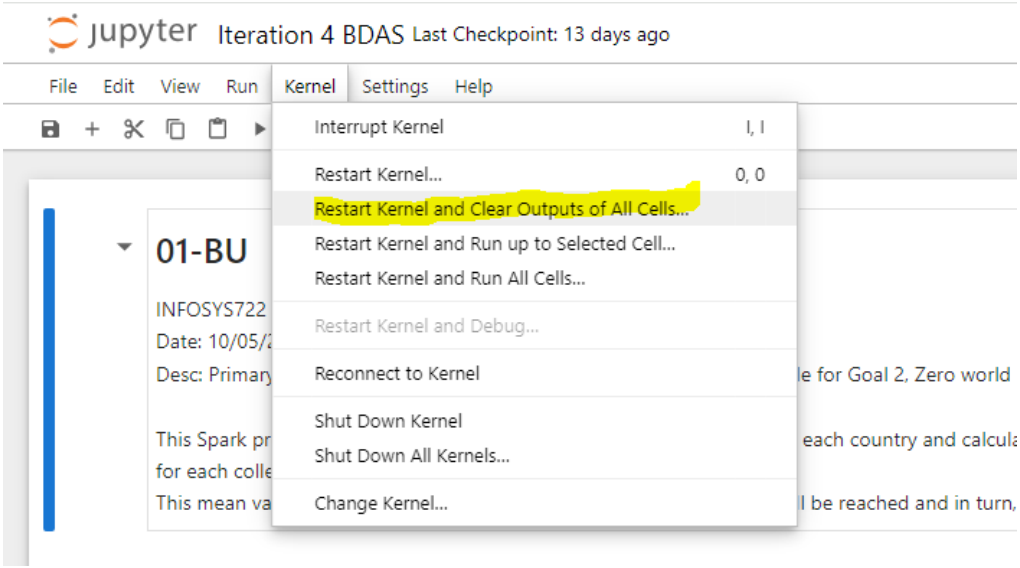

Figure 17

9. Click the menu item “Cell” followed by “Run All”

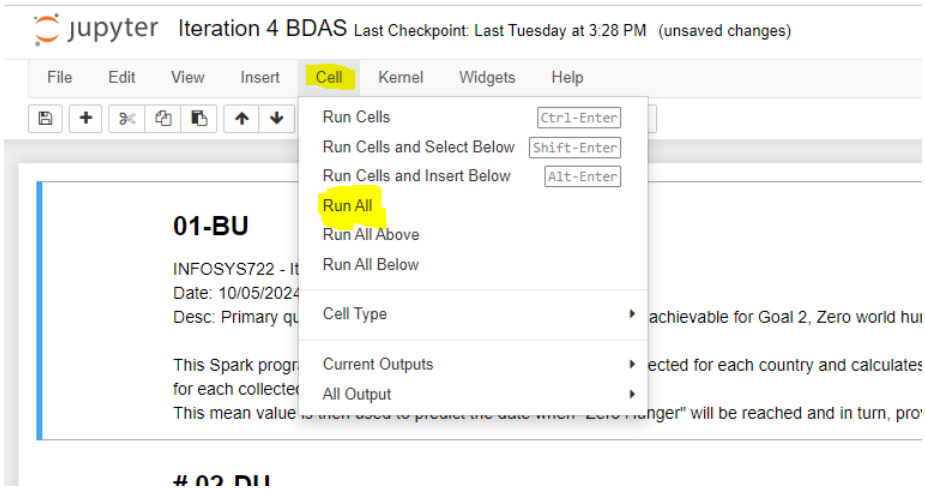

Figure 18

10. This will run the program and produce all the required graphs. Figure 19 should show on completion.

```
------------------------------------------------
Model Run - 6. Original/Actual/NewPredict - Outputs
R2 score: 0.9998657480831022
Coefficients: [[-0.57755833]]
Intercept: [1179.57915]
------------------------------------------------
Run Completed.
```

Figure 19

7.3 Model Output.

```
------------------------------------------------
Model Run - 1. Original GHI prediction - Outputs
R2 score: 0.9998657480831022
Coefficients: [[-0.57755833]]
Intercept: [1179.57915]
------------------------------------------------
```

Figure 20

Figure 20 shows the output from **Model Run 1**. There is a similar output for each model run. The **coefficients** are the most **useful** piece of information in these **outputs**. It indicates the **change value calculated** via algorithms of either gain or loss per year that is **applied** to the **target** to make **predictions**. This can be used to validate the run of the figures presented in the predictions by comparing the input data.

i.e.

First Instance GHI	**= 24.4393**
6 Years x Coef (-0.5776)	**= -3.4656**
Next calculated GHI	**= 20.9737 (24.4393 – 3.4656)**
	========
Second Instance GHI	**= 21.04**
Difference	**= 0.0663** (Acceptable)

We can ignore the **intercept** value returned here as it does not provide much useful information in this context. The **intercept** is based on **Continuous Time** and uses calculations based on **Year 0 being -9985** and counts from there. It does not have any direct relationship to the current time in the current context. Other methods can be utilized to change this but are not required.

The **R2 score** is an overall model rating and indicates the level of effectiveness the model has achieved. (Ranges from 0 >>>> 1. 1 being the model predicts the target perfectly)

With an **R2** of **.99**, the calculated **coef** of **-0.5776** in the model can be considered **accurate**.

Figure 21 below shows all the model outputs for the run.

```
------------------------------------------------
Model Run - 1. Original GHI prediction - Outputs
R2 score: 0.9998657480831022
Coefficients: [[-0.57755833]]
Intercept: [1179.57915]
------------------------------------------------
------------------------------------------------
Model Run - 2. GHI Original Tracked & Variation Point - Outputs
R2 score: 0.9998657480831022
Coefficients: [[-0.57755833]]
Intercept: [1179.57915]
------------------------------------------------
------------------------------------------------
Model Run - 3. GHI Actual/Original vs Original Prediction - Outputs
R2 score: 0.8559879854020668
Coefficients: [[-0.3692947]]
Intercept: [762.13553146]
------------------------------------------------
------------------------------------------------
Model Run - 4. Actual v New Prediction - Outputs
R2 score: 0.8559879854020668
Coefficients: [[-0.3692947]]
Intercept: [762.13553146]
------------------------------------------------
------------------------------------------------
Model Run - 5. Current Future prediction - Outputs
R2 score: 0.8559879854020668
Coefficients: [[-0.3692947]]
Intercept: [762.13553146]
------------------------------------------------
------------------------------------------------
Model Run - 6. Original/Actual/NewPredict - Outputs
R2 score: 0.9998657480831022
Coefficients: [[-0.57755833]]
Intercept: [1179.57915]
------------------------------------------------
```

Figure 21

8. Interpretation

Even though the reduced dataset is small, some trends and identifying data points were evident that supported a determination of the primary objective:

“Will the United Nations reach the goal of Zero Hunger by the year 2030”.

8.1 Discuss the mined patterns.

A pattern that showed in the mined data was the variation of the Actual compared to the Predicted. (See Figures 14-16 in 8.2 Visualizations)

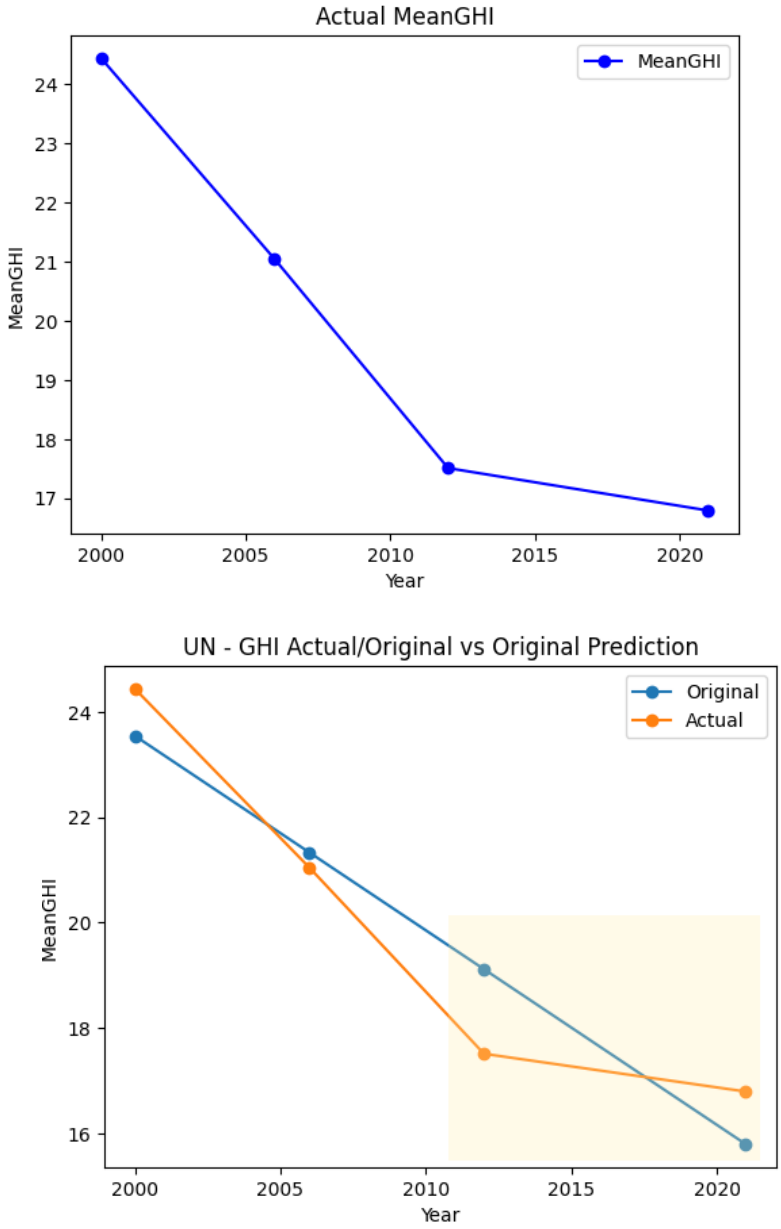

Figure 22

Graph 1 in Figure 22 shows the progress as of 2022. In it, it shows an upward swing that had taken place between 2012 and 2021 with the trajectory lifting from the downward trend that had been taking place. This indicated some sort of change, or events had occurred. Graph 2 in Figure 22 also reinforces the change showing what the projected trajectory was supposed to be compared to the actual.

Another pattern that stood out was the change in time gaps between readings.

This aligned with the amount of change in the **Mean GHI** for 2012-2021.

Year	Mean GHI	Change
2000	24.4393	
2006	21.0435	**-3.3958**
2012	17.5086	**-3.5349**
2021	16.7906	**-0.718**

- 2000 to 2006 = 6 years
- 2006 to 2012 = 6 years
- 2012 to 2021 = 9 years

This also indicates that some sort of event(s) had impacted both the timeline as well the progress of the project.

8.2 Visualizations.

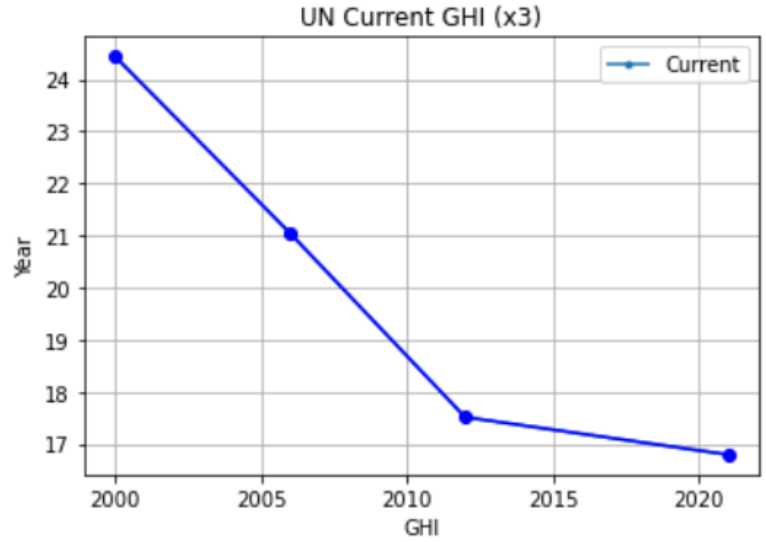

Figure 23

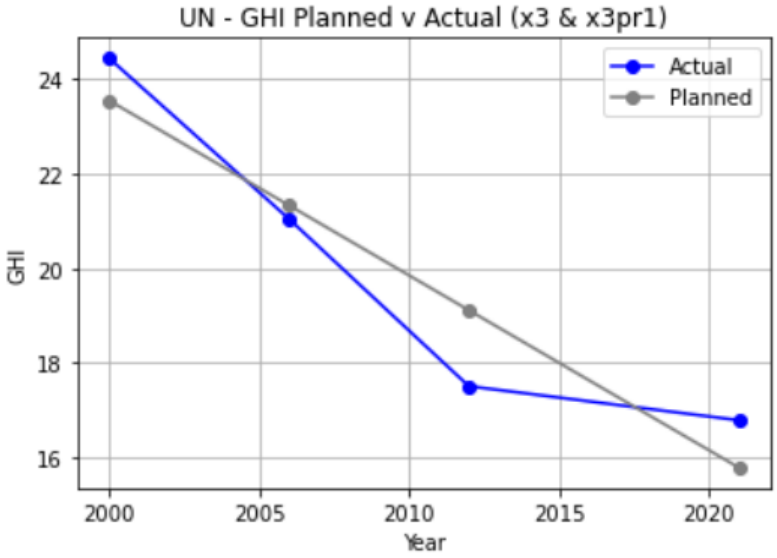

Figure 24

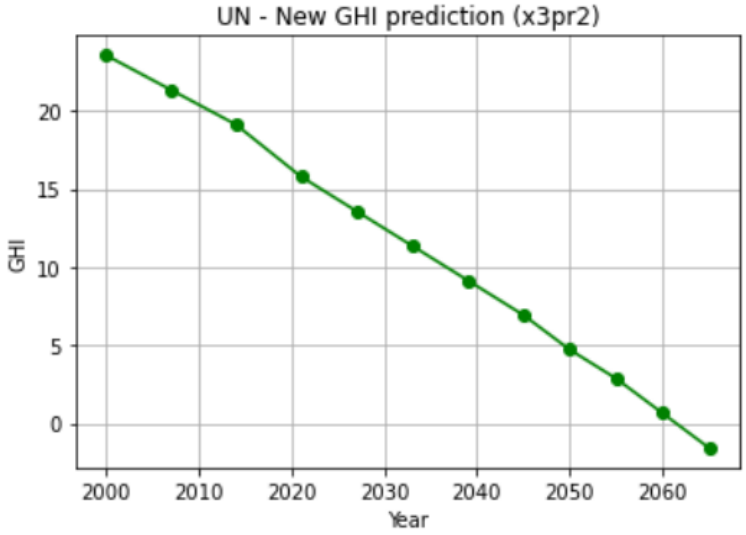

Figure 25

8.3 Interpret.

An early observation was that the target of 2030 was not achievable.

	Actual	Future
2000	24.4393	None
2006	21.0435	None
2012	17.5086	None
2021	16.7906	None
2027	13.575166	13.575166
2033	None	11.359398
2039	None	9.14363
2043	None	7.666451
2049	None	5.450683
2055	None	3.234915
2061	None	1.019146

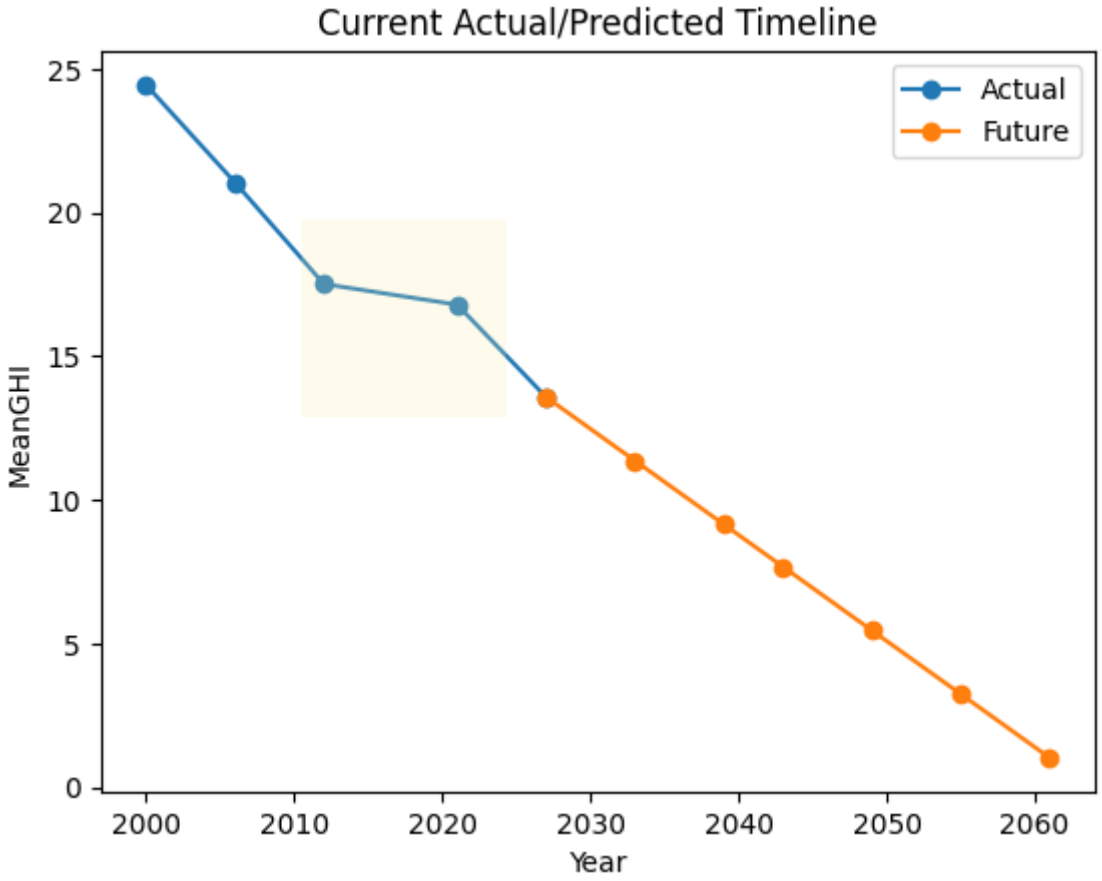

Figure 26

The time plot indicated that steady progress had been made up until the period between 2012 and 2021. Not only was there a sudden upswing in the trend line from the downward trend, but a larger time gap existed between instances. Before 2012, instances were evenly spaced at 6 years apart, the one between 2012 & 2012 was 9 years apart, 3 years longer. Because of these, new predictions indicate that the new target would likely be in the decade following 2060 if current progress was maintained. Figure 27 below illustrates this.

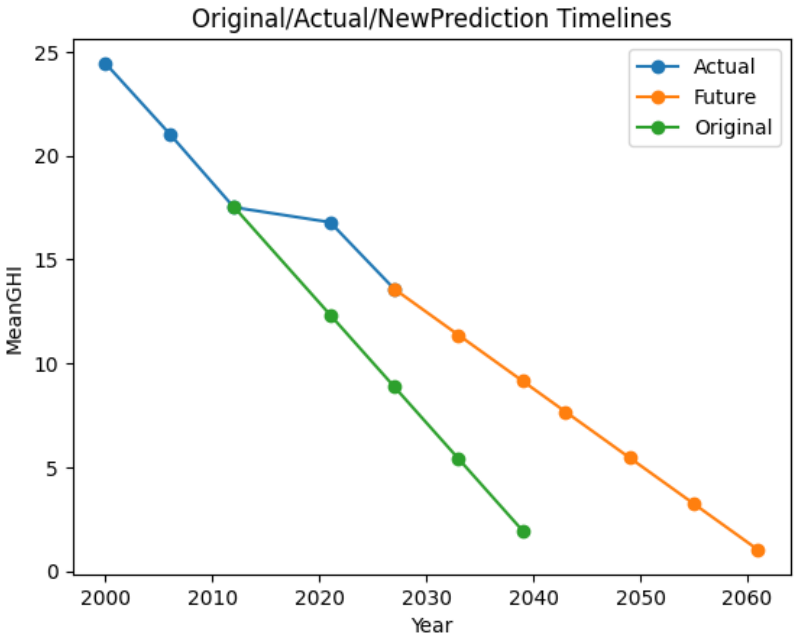

Figure 27

8.4 Assess & Evaluate Results

The same regression model was run twice on two separate sets of data to answer the overall question **"Will the UN complete the project by 2030"**

1. The first set of data had the last entry for 2021 removed to predict what would have happened had the events between 2012 & 2021 **NOT** taken place.

```
plt_1st
0    24.4393
1    21.0435
2    17.5086
3     0.0000
```

Figure 28

2. The 2nd set of data used the full set to do the current projection as of 2022.

```
+----+-------+
|Year|MeanGHI|
+----+-------+
|2000|24.4393|
|2006|21.0435|
|2012|17.5086|
|2021|16.7906|
+----+-------+
```

Figure 29

The actual plot in Figure 27 shows a definite upswing in the 2021 reading from the overall downward trend in the previous readings. This indicated that something had affected progress.

During the period 2012 – 2021, **COVID-19** and a **recession** hit most countries to varying degrees. NZ alone experienced 2 major bouts of **COVID** between 2016 & 2020.

A linear progression is effectively a trend projection based on the previous values. A second set of data would answer the objective's question. Figure 27 also illustrates this.

To answer the question "Will the UN achieve the date of 2030 for the Zero Hunger Program?"

"The date of 2030 is not achievable at the current progress rate."

If these event(s) mentioned had not taken place or were not as severe then as indicated in Figure 30 below, the date of 2030, or the decade following would have been achievable.

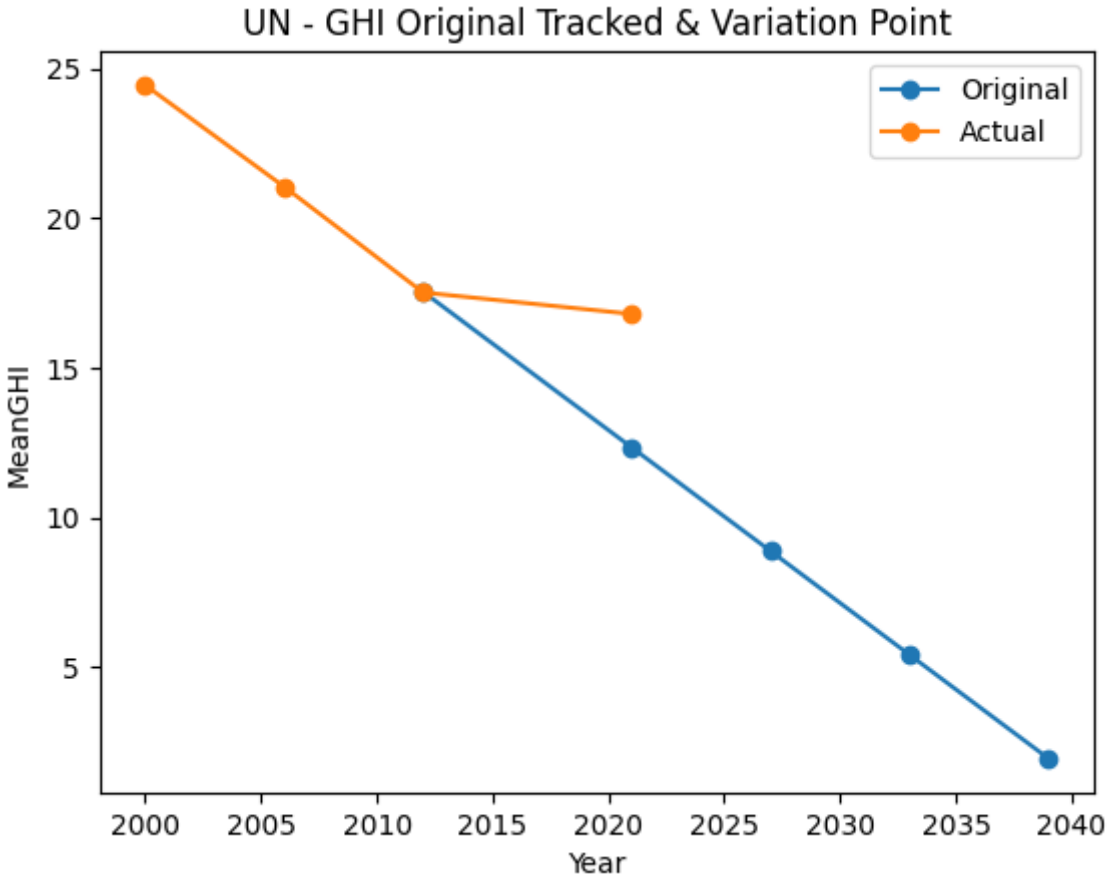

Figure 30

Figure 30 above is a projection based on the original Mean calculation data with the last period removed. Interestingly the original projection data before the upswing indicates Zero Hunger could have been achieved within the decade following 2030 but not exactly in 2030.

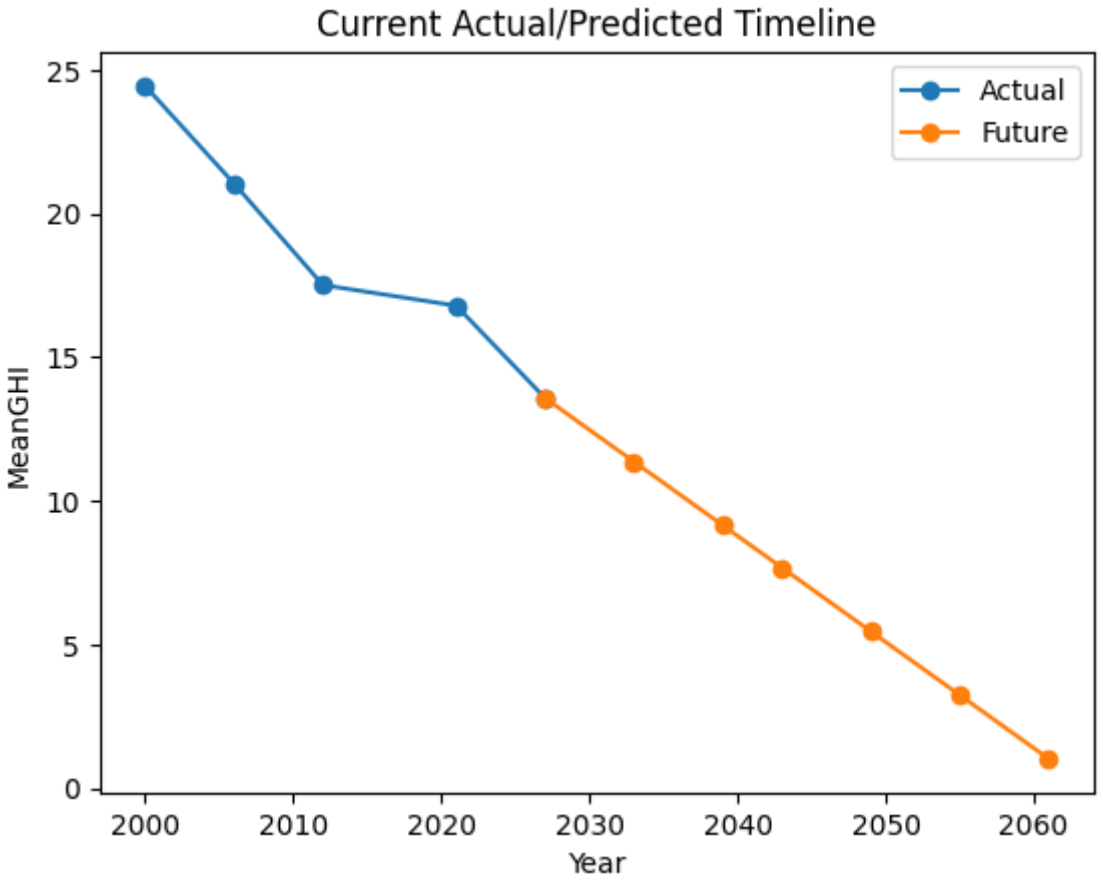

Figure 31

Figure 24 above is based on the current Mean model calculation data, and it indicates that the goal at its current rate, would not reach Zero Hunger until the decade following 2060. The **COVID** upswing directly influenced the completion date of the project.

8.5 Iterate

This document is the 3rd iteration based on the second iteration for assignment 2. Using a different software tool, the same processes have been followed.

It took me a few tries to get the right data streams and extracts working. I had to review and retry this a couple of times, to find the right API calls that would be what I required.

After finding the API's that would provide the outputs I required, and getting them to work, I decided to validate the output by fully projecting the linear trendline to their completion.

The **LineaRegression** library is a very powerful library with many features. With more time and more iterations, more features could be used to give an even deeper insight.

Some predictions did not balance out correctly with the R2 and took multiple attempts to finally get the R2 in a good position.

When trying to produce the data to get the trend line correct for the graph in Figure 32, I kept getting straight lines like in Figure 33. It took multiple attempts and rechecking the R2 before realizing my data types were inconsistent. After a few more attempts my R2 reached .88 and although it was not the same as the lofty .99 in other regressions, I was happy with this. In various attempts, I had to drop into debug mode and watch the values of some variables change to get an idea as to what was going on.

```
------------------------------------------------
Model Run - 4. Actual v New Prediction - Outputs
R2 score: 0.8804406036328536
Coefficients: [[-0.3692947]]
Intercept: [762.13553146]
------------------------------------------------
```

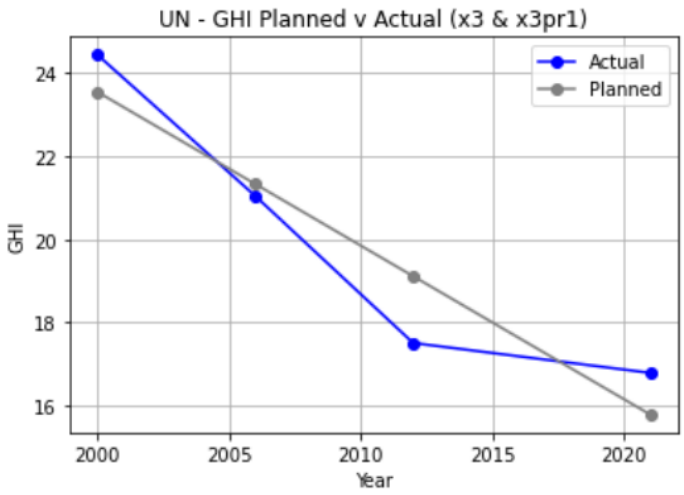

Figure 32

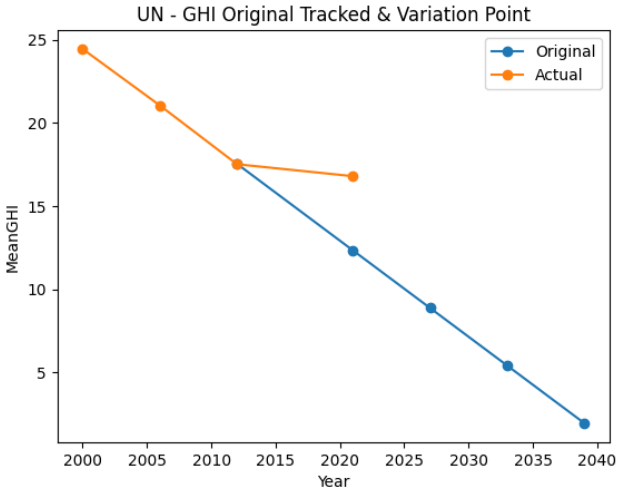

Figure 33

9. Action

9.1 How would you apply and deploy the implementation?

The Implementation is a validation model and only needs to be run when new data has been updated.

Whenever the model is run the main output graph (Figure 23) as well as the Excel spreadsheet graphs can be added to a website as a new page and the brief surrounding these added provides a historical analysis with each page representing a point in time.

9.2 How would you monitor the implementation?

Watching for updates from the UN regarding GHI ratings would be the main task. On update, the new ratings can be added, and the model is run again. The resulting outputs can be added as per section 9.1.

9.3 How would you maintain the implementation?

Updates only happen when new readings are available. At this point, any maintenance required can be assessed. Any future changes in the results may require changes to the overall system model to accommodate.

9.4 How would you enhance the implementation?

10. References

Bezawit Beyene Chichaibelu, M. B. (2024, March 01). Food Policy. *The global cost of reaching a world without hunger: Investment costs and policy action opportunities*, p. 1. Retrieved from https://www.sciencedirect.com/science/article/pii/S0306919221001299#t0005

Chauhan, A. (2022, 03 01). *Kaggle*. Retrieved from The Global Hunger Index: https://www.kaggle.com/datasets/whenamancodes/the-global-hunger-index

Index, G. H. (2024, March 20). *Methodology*. Retrieved from Global Hunger Index: https://www.globalhungerindex.org/methodology.html

Nations, U. (1945, Oct 19). *UN Charter*. Retrieved from United Nations: https://www.un.org/en/about-us/un-charter

Nations, U. (2024, March 01). *Goals*. Retrieved from UN Sustainable Development: https://sdgs.un.org/goals/goal2

Nations, U. (2024, 03 20). *Home*. Retrieved from United Nations Sustainable Development Goals: https://www.un.org/sustainabledevelopment/sustainable-development-goals/

Nations, U. (n.d.). *Goal 2: Zero Hunger*. Retrieved from United Nations: https://www.un.org/sustainabledevelopment/hunger/

scikit. (2024). *sklearn.linear_model.LinearRegression*. Retrieved from scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

WHO. (2024, March 20). *Global Hunger Index (GHI)*. Retrieved from World Health Organisation: https://www.who.int/data/nutrition/nlis/info/global-hunger-index-(ghi)#:~:text=The%20global%20hunger%20index%20captures,extent%2C%20attributable%20to%20undernutrition).

11. Disclaimer

"I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright. I also acknowledge that I have appropriate permission to use the data that I have utilized in this project. (For example, if the data belongs to an organization and the data has not been published in the public domain, then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data."