CCT College Dublin

Assessment Cover Page

Module Title:	**Strategic Thinking CA2**
Assessment Title:	Global plastic usage: Future Impact Awareness
Lecturer Name:	**James Garza**
Student Full Name:	**Hodan Mohamed Abdi**
Student Number:	Sba23416
Assessment Due Date:	**17th December 2023**
Date of Submission:	**17th December 2023**

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Global plastic usage: Future Impact Awareness

Table of content:

Introduction:

The main purpose of this report is to examine the usage of plastic and the global impact that it has to bring awareness to the readers.

Plastic pollution has emerged as one of the most pressing global environment challenges of our time. The report will delve into the complexities of the situation and its far-reaching significance, aiming to shed light on the myriad issues that underlie this crisis.

By examining the contributing factors of the increasing plastic usage, the economics of plastic production this report seeks to highlight a future forecast about the situation.

Objectives:

This assignment aims to explore the following objectives:

1- Global usage of plastic: Investigate and examine the current usage of plastic worldwide. Highlight main factors of consumption based on previous and actual data with mass production trends.

2- Impact of plastic utilisation: Forecast will indicate: What are the consequences? How will pollution and the ecosystem be?

3- Waste management: Exhibit the recycling data in order to respond to possible future risks.

By addressing these objectives, the goal is to raise awareness to have hypothetical sustainable solutions.

Problem Definition:

This report will identify the key issues of global plastic usage. The global problem of plastic usage presents a multifaceted crisis that demands immediate attention. With each day passing, the excessive reliance on plastic deepend, leading to dire consequences for the environment and future generations.

Awareness is crucial for this report will delve into the gravity of the situation, emphasising the extreme need for immediate action. By addressing this crisis as early as possible this is essential to safe-guarding and preserving the planet.

Although, By addressing it holistically, this include:

1. Reducing single-use plastics
2. Recycling resources
3. Fostering global cooperation

Challenge of the project are the following:

- Appropriate data: finding dataset that have enough rows to support the objective of the analysis
- Avoid Bias in the analysis: staying neutral and focusing on the facts.
- Finding sustainable solutions for the problem supported with reliable dataset: the lack of insufficient data has led to continuous research. The project aims to raise awareness which gives the reader an opportunity to research sustainable solutions.

The context of the problem and the important of this to be addressed are:

> *"Plastic pollution is a planetary threat, affecting nearly every marine and freshwater ecosystem globally. In response, multilevel mitigation strategies are being adopted but with a lack of quantitative assessment of how such strategies reduce plastic emissions The global threat from plastic pollution" (Borrelle et al. 2020, p.1).*

Scope:

Over the two semester the scope of the project is to analyse the following topics and try to answer the following questions:

- Usage of plastic worldwide: which factors are impacting, mass production, forecast for future production of the plastic if nothing will change.

- Pollution and Ecosystem: How much the pollution increased and the ecosystem degraded? What are the causes?

- Analyse the recycling waste: what do we recycle? What is the capacity of the recycling facilities? Based on the analysis will we be able to plan and respond to the demand?

The project aims to bring at the end of the two semester attention on the topic to be more conscious about the long term effect on this concept.

Inclusions of the project:

- Definition of the problem and objective
- Analysis of the dataset of worldwide plastic usage
- Forecast on plastic usage
- Analysis of global pollution dataset
- Forecast about pollution
- Analysis of general waste
- Forecast about capacity of recycling facilities
- Conclusion of the analysis to bring awareness on the topic

Exclusions:

- Bias
- Avoid using personal data
- Not providing sustainable solution

Role and responsibilities:

Role	**Who:**
Project Manager	Cristina/Hodan
Researcher	Cristina/Hodan
Data collection and cleaning	Cristina/Hodan
Data Analysis	Cristina/Hodan
Data visualisation	Cristina/Hodan
Writer	Cristina/Hodan

Table 1: Role and Responsibilities.

Boundaries:

- Dataset limitation
- To avoid the limitation of the geographic area the project aims to analyse the situation on a global scale as to why the project title evolved during the time
- Research of solution to the argument
- Time frame of the project: Two semesters

In-depth analysis:

	First semester	Second semester
Focus	Foundational research: extensive literature review and data set collection	Analysis of garthing, the synthesis of findings and the formulation of rememendations
Aims	Prepare the material to move to the second semester phase.	Allow for a deeper exploration of global policies, case studies as well best practices and will also provide ample time for additional review if required which will ensure that the report quality.

Table 2: Analysis of tasks

Planning:

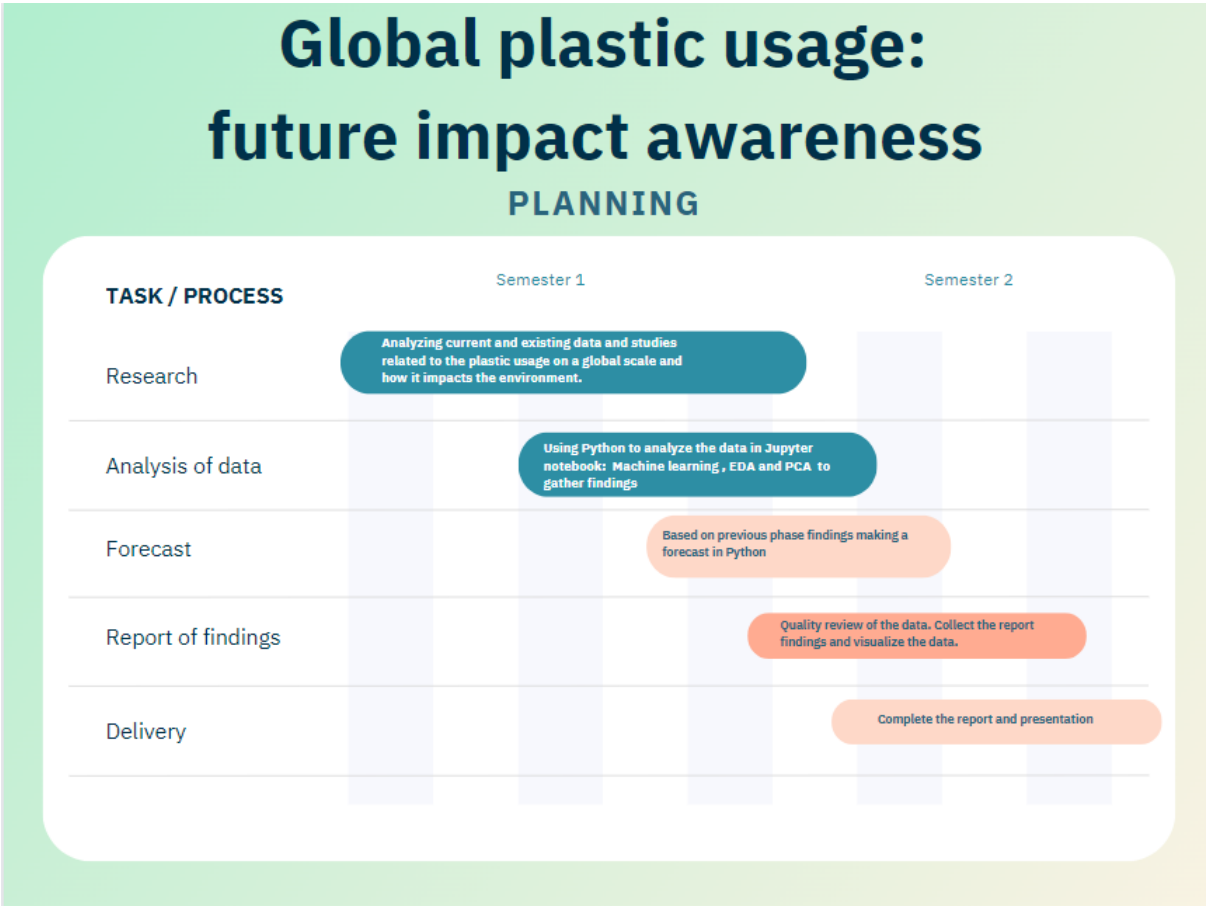


Table 3: Project planning

Project accomplishment:

The main recommendations and strategies for raising awareness, driving change as well as promoting recycling best practices. This approach will ensure a well rounded, evidence-based exploration of the global plastic usage issue and future awareness.

Final consideration on the project scope:

By the end of semester two we aim to deliver a comprehensive academic report that includes the following:

1. Extensive research: a well-researched report featuring a thorough literature review, primary and secondary research, data analysis and an in depth exploration of global plastic usage. This will include:
    - Daily plastic consumption
    - Global pollution
    - Recycling waste facilities
    - Forecast based on the data
2. Data-driven recommendations: This will include evidence based recommendations for raising awareness to mitigate the future impact of plastic usage

Potential data for the project:

Data Source	Data amount	Permission
www.kaggle.com. (n.d.). Global Plastic Pollution. [online] Available at: https://www.kaggle.com/datasets/sohamgade/plastic-datasets	Full	open resources allowed by their terms and conditions
Datopian (n.d.). Daily_csv plastic monkey 78. [online] DataHub. Available at: https://www.datahub.io/gitchenze/daily_csv-plastic-monkey-78 [Accessed 15 Oct. 2023]	Full	open resources allowed by their terms and conditions
Our World in Data. (n.d.). Extrapolated change in plastic fate. [online] Available at: https://ourworldindata.org/grapher/plastic-fate-to-2050.	Full	open resources allowed by their terms and conditions

Table 4: Data sources

The data that has been found as potential for the project are from open resources so they are allowed to be used by their terms and conditions.

Ethical considerations:

While the report global plastic usage: future impact awareness does not directly involve sensitive data, user privacy or potential societal impact, however we believe ethical considerations remain essential to the report.

The report will prioritise transparency and accuracy in the use of data, adhering to proper dataset usage.

All the sources referenced will be appropriately cited under the guidelines of Harvard Reference to acknowledge the contribution of others to avoid plagiarism.

Additionally the report will emphasise the importance of responsible data handling and the ethical use of information for academic and educational purposes.

Practical Coding Artefact

The purpose of this practical artefact is to explore the dataset "Per-Capita-Plastic-Waste-vs-GDP-Per-Capita_df applying data analysis techniques.

Data Preparation/Evaluation Methods and EDA:

Data prep, evaluation, and EDA are crucial for dataset quality. Initial inspection shows 7 variables, requiring further exploration for handling

```
In [3]:  1 GB_df.head()
```

Out[3]:

	Entity	Code	Year	Per capita plastic waste (kg/person/day)	GDP per capita, PPP (constant 2011 international $)	Total population (Gapminder, HYDE & UN)	Continent
0	Abkhazia	OWID_ABK	2015	NaN	NaN	NaN	Asia
1	Afghanistan	AFG	2002	NaN	1063.635574	22601000.0	NaN
2	Afghanistan	AFG	2003	NaN	1099.194507	23681000.0	NaN
3	Afghanistan	AFG	2004	NaN	1062.249360	24727000.0	NaN
4	Afghanistan	AFG	2005	NaN	1136.123214	25654000.0	NaN

Size of the dataset : In this section i have identified the size of the dataset using the follow code below: Dataset size: (48168, 7)

Image: Data Head()

GB_df.shape	
48168	7

Table : Data shape

Pandas classifies 7 columns: 1 integer, others objects. Variables should be floats or integers; investigation needed. File size is 78+ MB.

Column	Non-Null Count	Dtype
Entity	48168 non-null	object
Code	46154	non-null object
Year	48168	non-null int64
Per capita plastic waste (kg/person/day)	186 non-null	float64
GDP per capita, PPP (constant 2011 international $)	6407	non-null float64
Total population (Gapminder, HYDE & UN)	46883	non-null float64
Continent	285	non-null object
dtypes: object(3)	float64(3)	int64(1)

Table: Variables & Types

Table summarises non-null counts and data types for columns, encompassing entities, years, plastic waste, GDP, population, and continents.

	Count	Mean	STD	Min	25%	50%	75%	Max
Year	48168.0	1.903147e+03	3.157168e+02	-10000.000000	1859.000000	1.920000e+03	1.975000e+03	2.019000e+03
Per capita plastic waste (kg/person/day)	186.0	1.798118e-01	1.230064e-01	0.01000	0.103000	1.440000e-01	2.520000e-01	6.860000e-01
GDP per capita, PPP (constant 2011 international $)	6407.0	1.492610e+04	1.773975e+04	247.43654	3021.071807	8.447264e+03	1.960754e+04	1.353188e+05
Total population (Gapminder, HYDE & UN)	46883.0	2.982790e+07	2.530860e+08	905.00000	201733.500000	1.542937e+06	5.886795e+06	7.713468e+09

Table: Statics

These descriptive statistics reveal how the different variables are spread and clustered in the dataset.

Observing numerous "NaN" values, we check the target variable's unique values for potential "na" occurrences.

Identifying Unique	
GB_df["Entity"].unique()	GB_df["Year"].unique()
GB_df["Continent"].unique()	GB_df["Per capita plastic waste (kg/person/day)"].unique()
GB_df["Total population (Gapminder, HYDE & UN)"].unique()	

Table: Identifying Unique Values

GB_df.isnull().sum().sum()	140925

Table: Identifying Null Values

Standardisation

Consistent representation of missing values, like converting "NA," "N/A," "NaN," to np.nan, simplifies data processing..

Handling Various Missing values
missing_value_formats = ["n.a.","NA","na","n/a","n\a","?","--"] GB_df = pd.read_csv("per-capita-plastic-waste-vs-gdp-per-capita.csv", na_values = missing_value_formats)

Table: Handling Missing Values

The code creates missing_value_formats to handle diverse missing values, promoting uniformity, ensuring consistency, and enhancing data quality during CSV file reading into a Pandas DataFrame.

Continuous identifying NaN values within the database: "Array"

GB_df["Continent"].unique()	
Asia	nan
Europe	Africa
Oceania	Antarctica
North America	South
America'	

Table: GB_df[Continent] with NaN Values

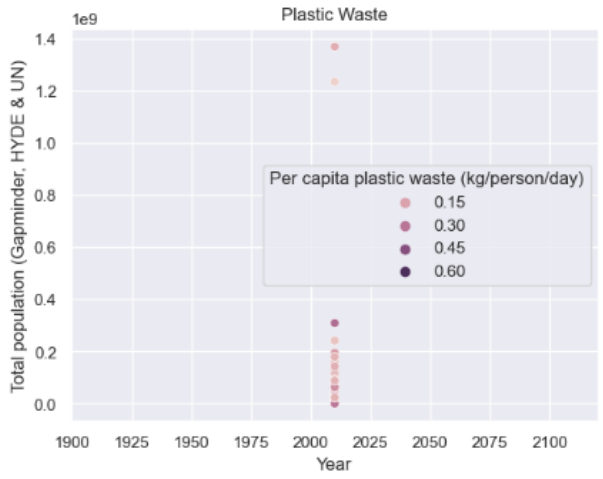
Transforming Missing Values

The code evaluates missing values and targets specific columns in the GB_df dataset, creating a new DataFrame (f5_l5_columns) for focused analysis. Descriptive statistics offer insights into the selected columns, aiding data exploration. Strategies for handling

missing values and assessing the "Continent" column's significance are recommended for informed data exploration and modelling.

Scatterplot Chart - Visualisation

Scatterplots visually summarise data distribution, revealing outliers. Positive values cluster right in the first two variables but disperse in the last two. Outliers and spread influence missing value imputation precision.



Graph:Scatterplot: Plastic Waste / Year / Total Population

14,925 missing values (41.80% of total) have implications for data analysis and modelling.

Exploring The Missing Values		
Number of Missing Values:		140925
Percentage of Missing Values		41.80%

Table: Exploring The missing Values

The dataset exhibits a substantial 41.80% missing values, impacting data completeness, quality, and introducing potential bias. Feature exclusion or specialised imputation methods may be employed. Imputation strategies, like mean or median, depend on data nature and goals. Addressing missing values is vital for reliable analyses and models, emphasising careful preprocessing for meaningful insights and robust modelling.

Visualising of Missing Values

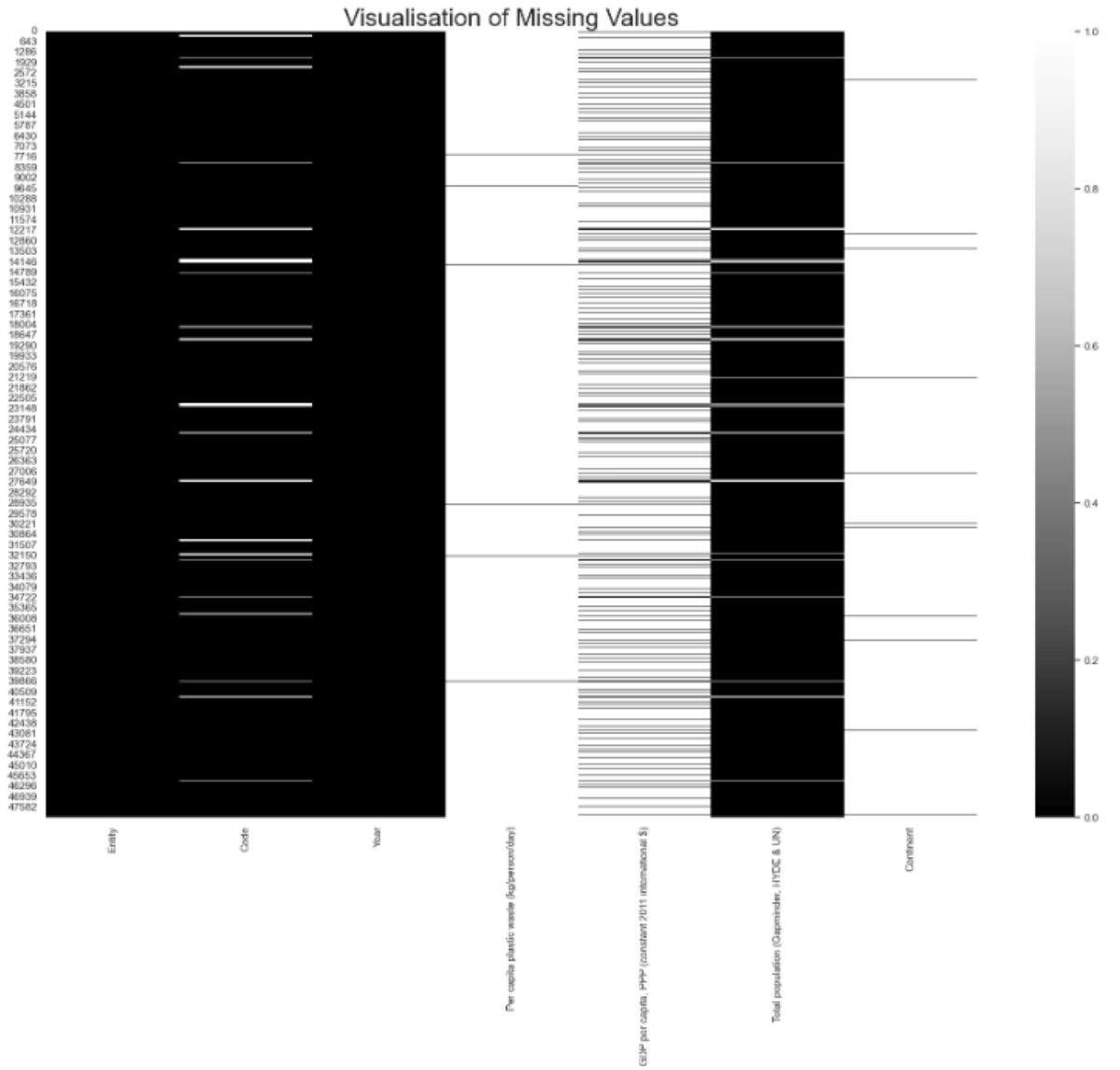


Image: Missing Values

The DataFrame indicates 6 instances of zero values, constituting about 0.00% of the total, suggesting a very low frequency.

Data Exploration:

Examining zero values frequency and location aids EDA, understanding variable features, and guiding preprocessing for sensible data interpretation and analysis.

Identifying Duplicate Rows
duplicate_rows = GB_df[GB_df.duplicated()] duplicate_rows

Table: Identifying Duplicate Rows

No rows are duplicated in GB_df, indicated by an empty variable for duplicate rows. This suggests a well-organised, unique dataset, reducing bias and errors in analyses and modelling, ensuring trustworthiness and effectiveness.

NaN Count		
Number of rows with more than 90% NaN values:		0
List of rows with more than 90% NaN values:		[]

Table: NaN counts

The dataset's quality, with no rows having over 90% NaN values, enhances analysis, trustworthiness, and model development.

Identifying 90% NaN

Empty DataFrameColumns:	
Entity	**Code**
Per capita plastic waste (kg/person/day)	**Year**
GDP per capita, PPP (constant 2011 international $)	**Total population (Gapminder, HYDE & UN)**
Continent	

Table: Identifying 90% NaN

Graph Chart Missing Values

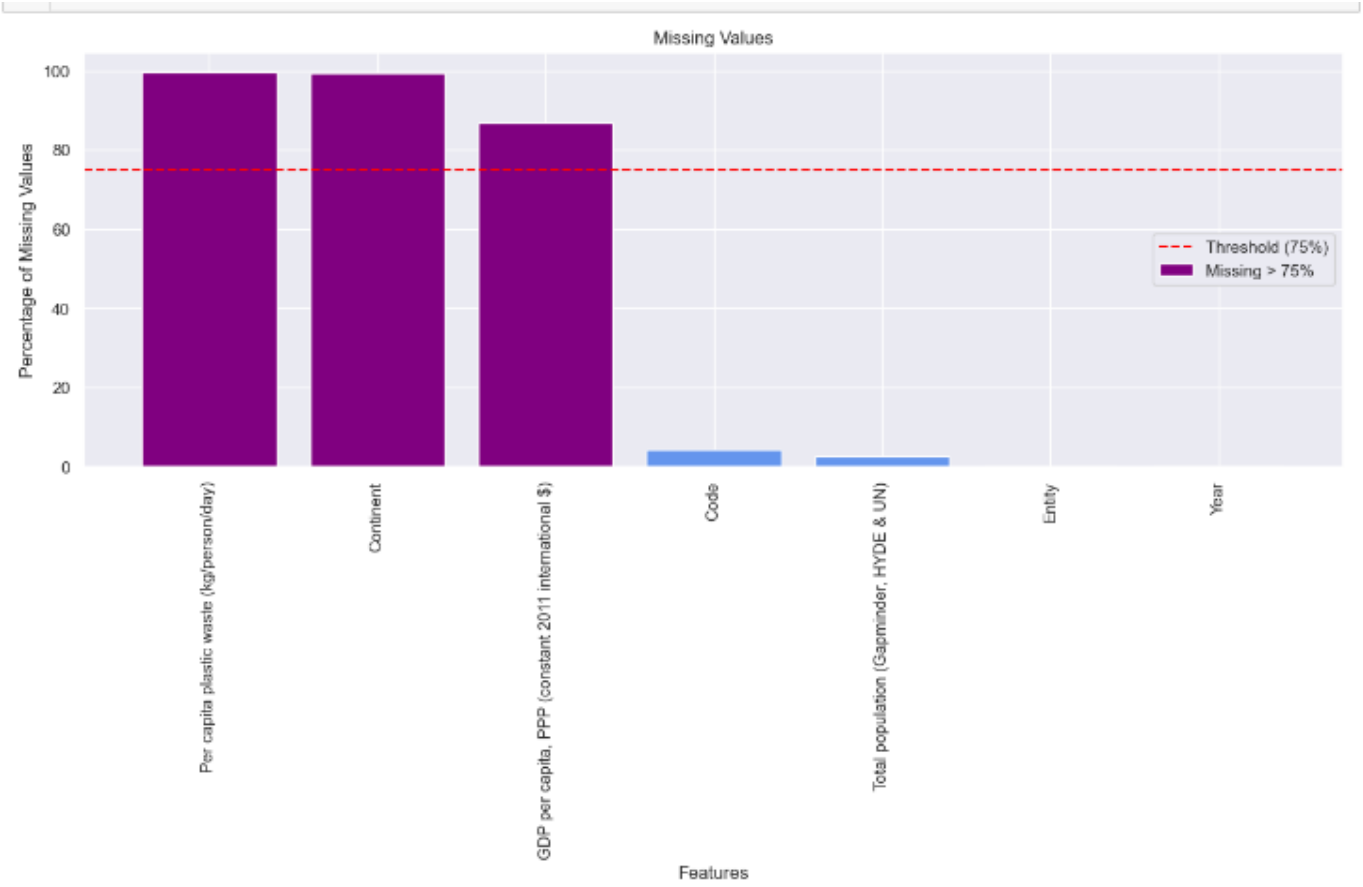


Image: Graph Chart: Missing Values

Exploring Variables with the most Missing Data

Variables with The Most Missing Data		
Per capita plastic waste (kg/person/day)	GDP per capita, PPP (constant 2011 international $)	Continent

Table: Variables with The Most missing Data

Removing Variables 75% Missing Values

GB_df = GB_df.drop(columns = var_75_plus_miss)	
Before	**After**
(48168, 7)	**(48168, 4)**

Table: Removing Variables >75& Missing Values

Visualisation: Missing Values After Dropping >75%

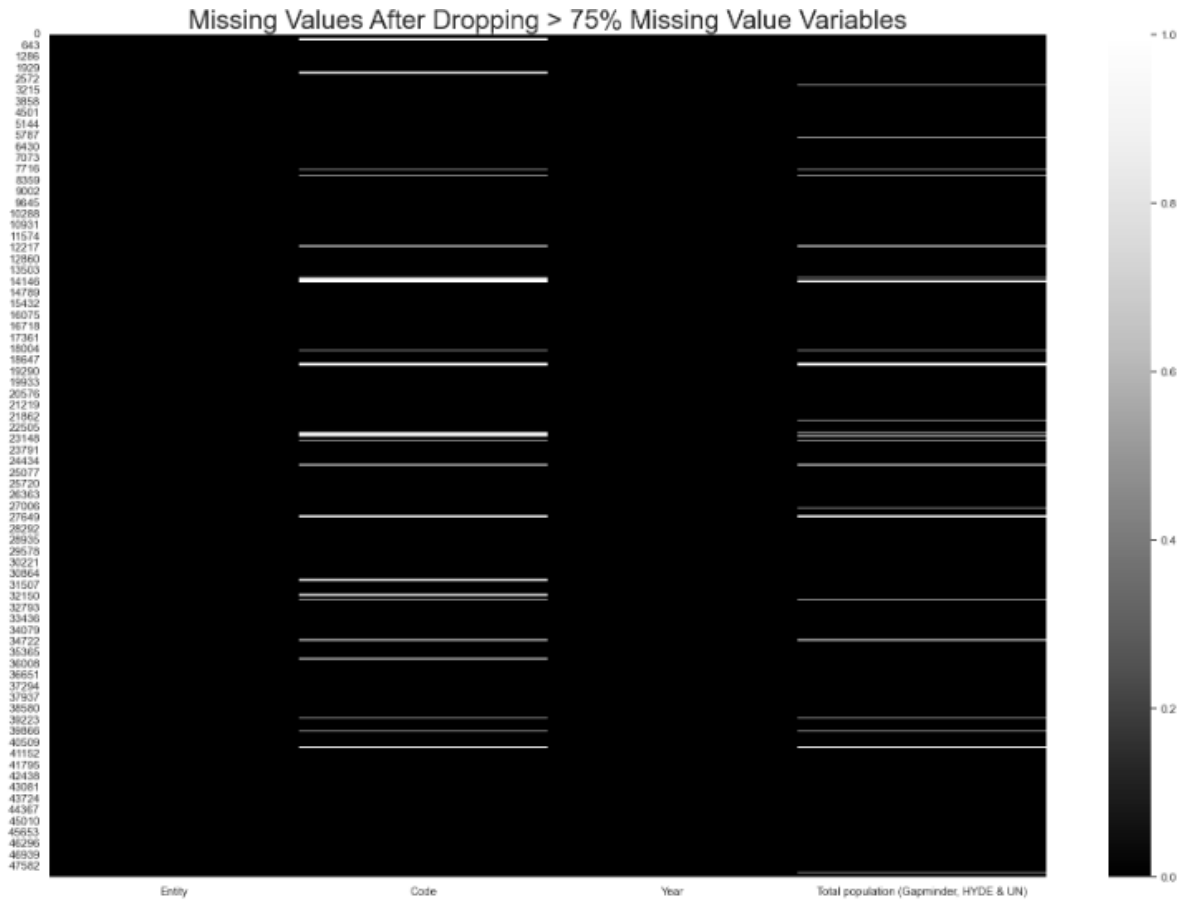


Image: Missing Values after Dropping >75%

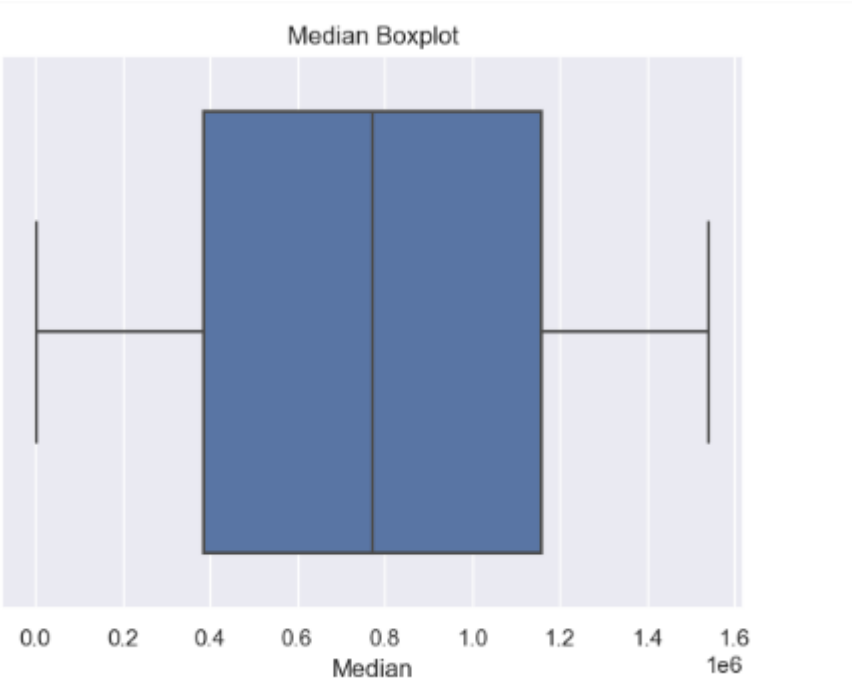
Exploring Median, Min & Max Values

	Variable	Median	Min	Max
0	Year	1920.0	-10000	2019
1	Total Population (Gapminder, HYDE & UN)	1542937.0	905.0	7723467904.0

Table: Exploring, Median, Min & Max

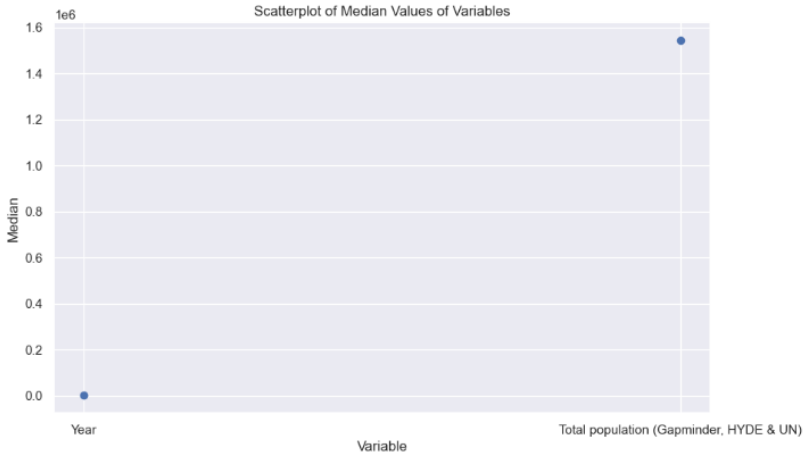
Table displays stats for two variables: Year (Median: 1920.0, Min: -10000, Max: 2019) and Total Population (Median: 1542937.0, Min: 905.0, Max: 7713467904.0).

Visualisation: Boxplot



Graph:Boxplot: Median

Visualisation: Scatterplot: Median Values of Variables



Graph:Scatterplot: Median Values of Variables

Percent of Missing Values Dropped	
Percent of Missing Values Dropped:	97.65903849565372
Number of Missing Values Currently:	3299
Percentage of Missing Values: 1.71%	1.71%

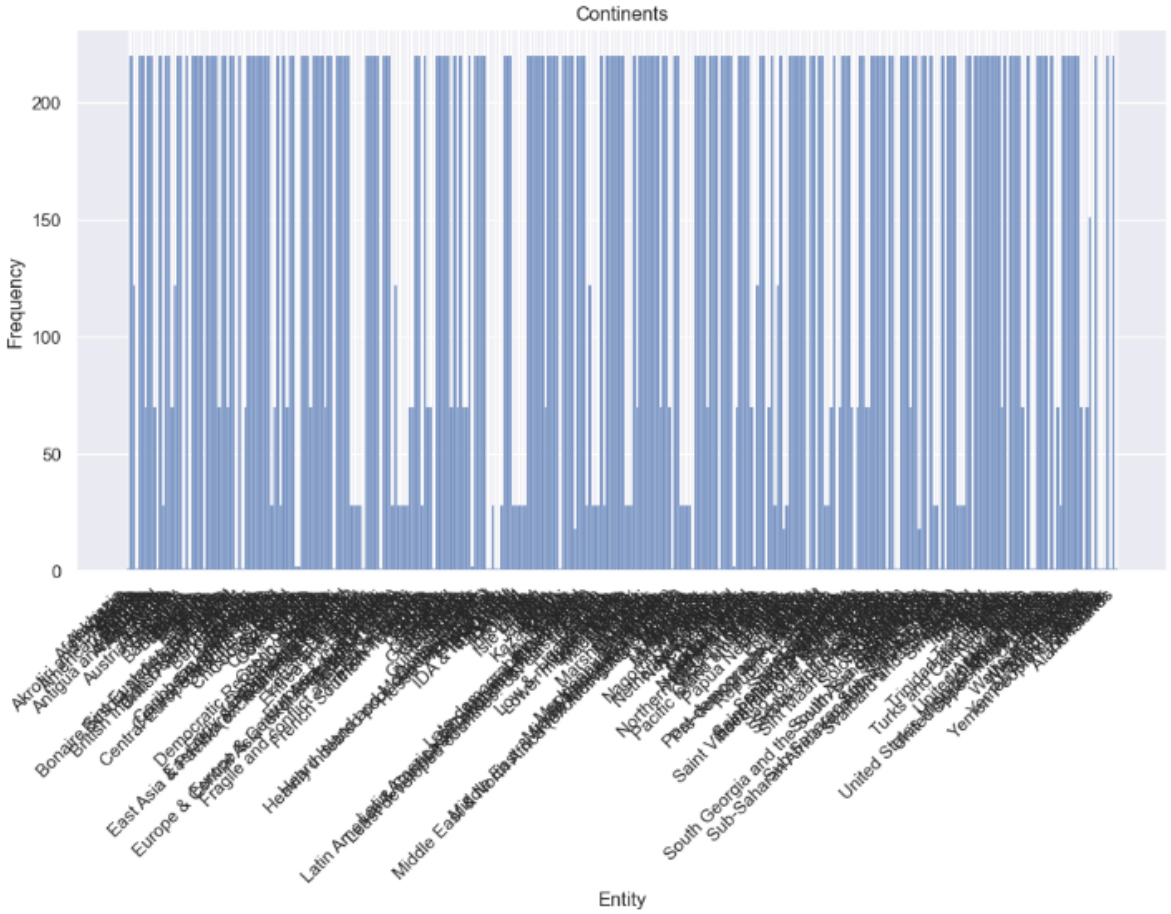
Table: Percent of missing Values Dropped

Summary reveals dataset gaps: 97.66% deleted, 3299 gaps remain (1.71% of total dataset).

Exploring Target Variables

Value counts analysis pre-scaling and imputation shows the unbalanced target variable in "Entity," "Year," and "Total Population." Note for future ML impact on accuracy and precision.

Histogram visualises data distribution in the "Entity" column, offering quick insight into entity or continent frequency.



Scaling Dataset:

Scaling Data	
X = GB_df.drop("Entity", axis = 1)	y = GB_df["Entity"]
Variable dataset to be scaled:	(48168, 3)
Target Variable - Entity:	(48168,)
Target Variable - Year:	(48168,)

Table: Scaling Dataset

In machine learning, separating input features (X) and target variable (y) is crucial for model development.

Encoding Data

Ecoding Data:

```
X_encoded = pd.get_dummies(X)
```

```
X_numeric = X.select_dtypes(include=['number'])
```

```
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, RobustScaler, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.feature_extraction import DictVectorizer
import pandas as pd

# X is the feature matrix

# Identify numeric and non-numeric columns
numeric_cols = X.select_dtypes(include=['number']).columns
non_numeric_cols = X.select_dtypes(exclude=['number']).columns

#Transformers for numeric and non-numeric data
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
```

```
print("Transformed Data:")
print(X_robust[:100, :])  # Print the first 5 rows
```

```
Transformed Data:
[[ 0.96424619  0.          0.         ...  0.          0.
   0.        ]
 [ 0.85217722 -1.09874187  0.         ...  0.          0.
   0.        ]
 [ 0.86079791 -0.93454398  0.         ...  0.          0.
   0.        ]
 ...
 [-0.19954692 -3.86293808  0.         ...  0.          0.
   0.        ]
 [-0.19092623 -3.85871393  0.         ...  0.          0.
   0.        ]
 [-0.18230554 -3.85437014  0.         ...  0.          0.
   0.        ]]
```

Image: Encoding & Transformed Data

Results:

Transformed data is displayed as a matrix, post scaling and preprocessing. Features centred around zero, and some columns have constant zero values, suggesting successful preprocessing for machine learning.

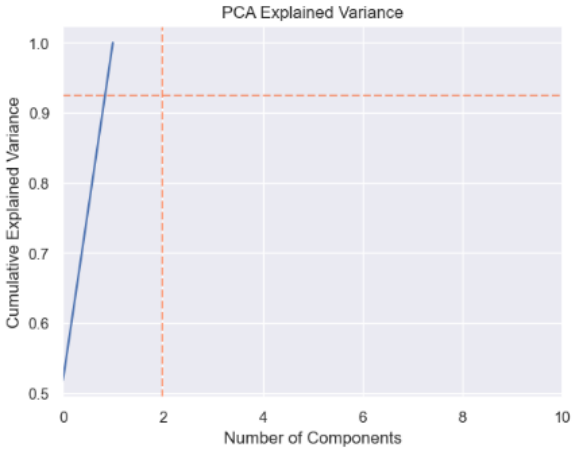
KNN Impute Missing Values

The code preprocesses GB_df by identifying numeric and categorical columns. It uses KNNImputer for numeric and common value fill for categorical, enhancing data completeness for machine learning analysis.

PCA

The code conducts PCA on GB_df's numeric columns, selecting and applying PCA while retaining 92% variance. It plots the cumulative explained variance ratio to guide dimension reduction and information retention.

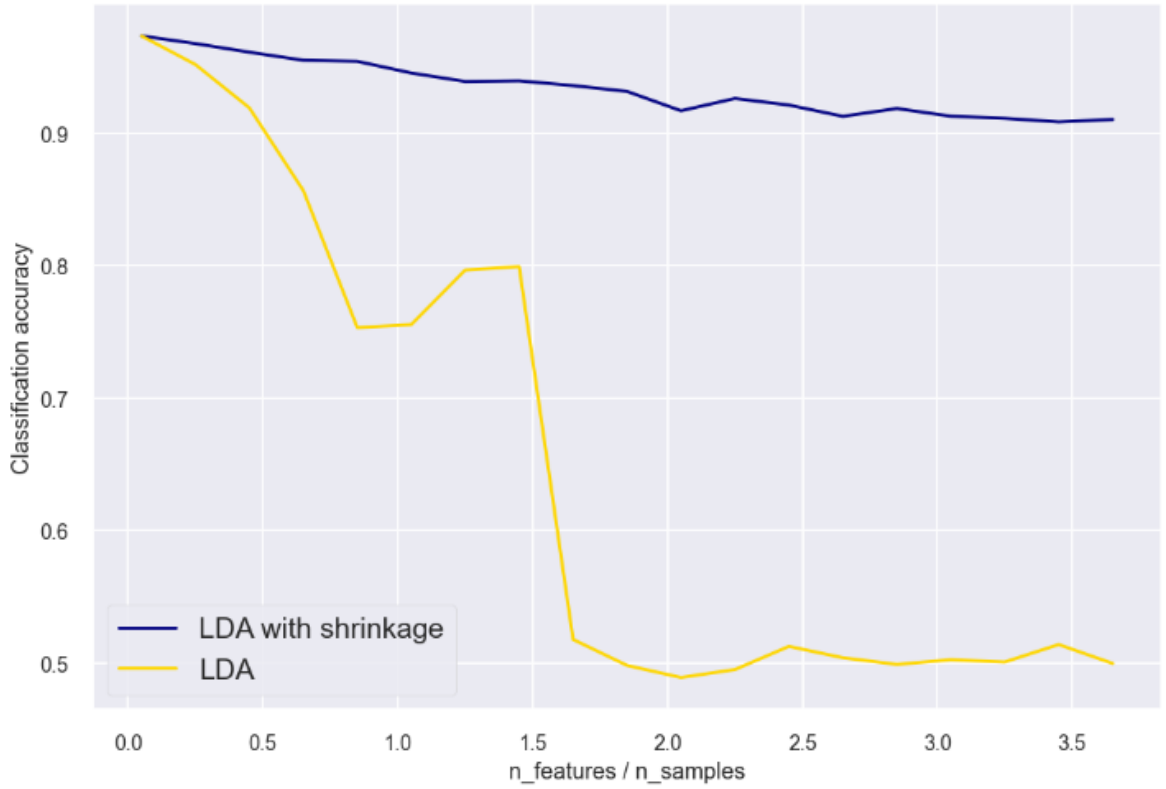
GB_df's standardised numeric columns undergo PCA with StandardScaler, visualising cumulative explained variance, aiding dimensionality reduction, and understanding



Graph: PCA

PC1 explains 51.76%, representing the primary data variation direction; PC2, orthogonal to PC1, explains 48.24%

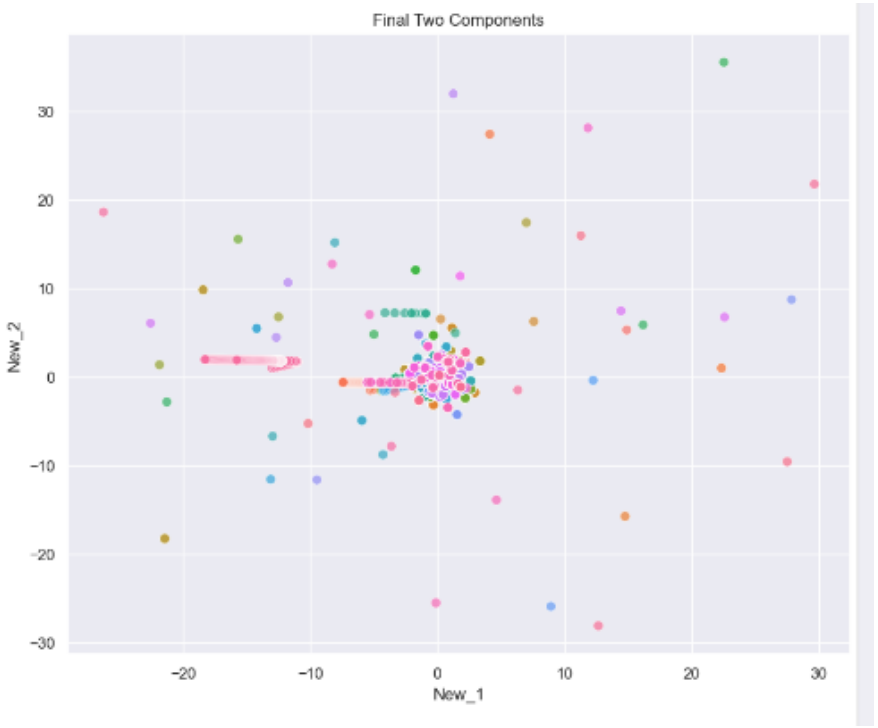
LDA



Graph: LDA

New Dataset

The dataset has only 1 independent feature and 1 dependent feature and 48,168 observations & 2 Features after the PCA. The data is completed by adding back the target variable, "Total population (Gapminder, HYDE & UN)" & "Year".



Scatter Chart: New 1 & New 2 Final Components

Machine learning Models:

Spearman's correlation coefficient for selected columns Year			
Year	1.000000	Total population (Gapminder, HYDE & UN)	0.264378

Table: Spearman's Correlation Coefficient for Year

Spearman's correlation coefficient for selected columns Total population (Gapminder, HYDE & UN)	
Year	0.264378
Total population (Gapminder, HYDE & UN)	1.000000

Table: Spearman's Correlation Coefficient for (Gapminder, HYDE & UN)

Spearman Correlation Heatmap

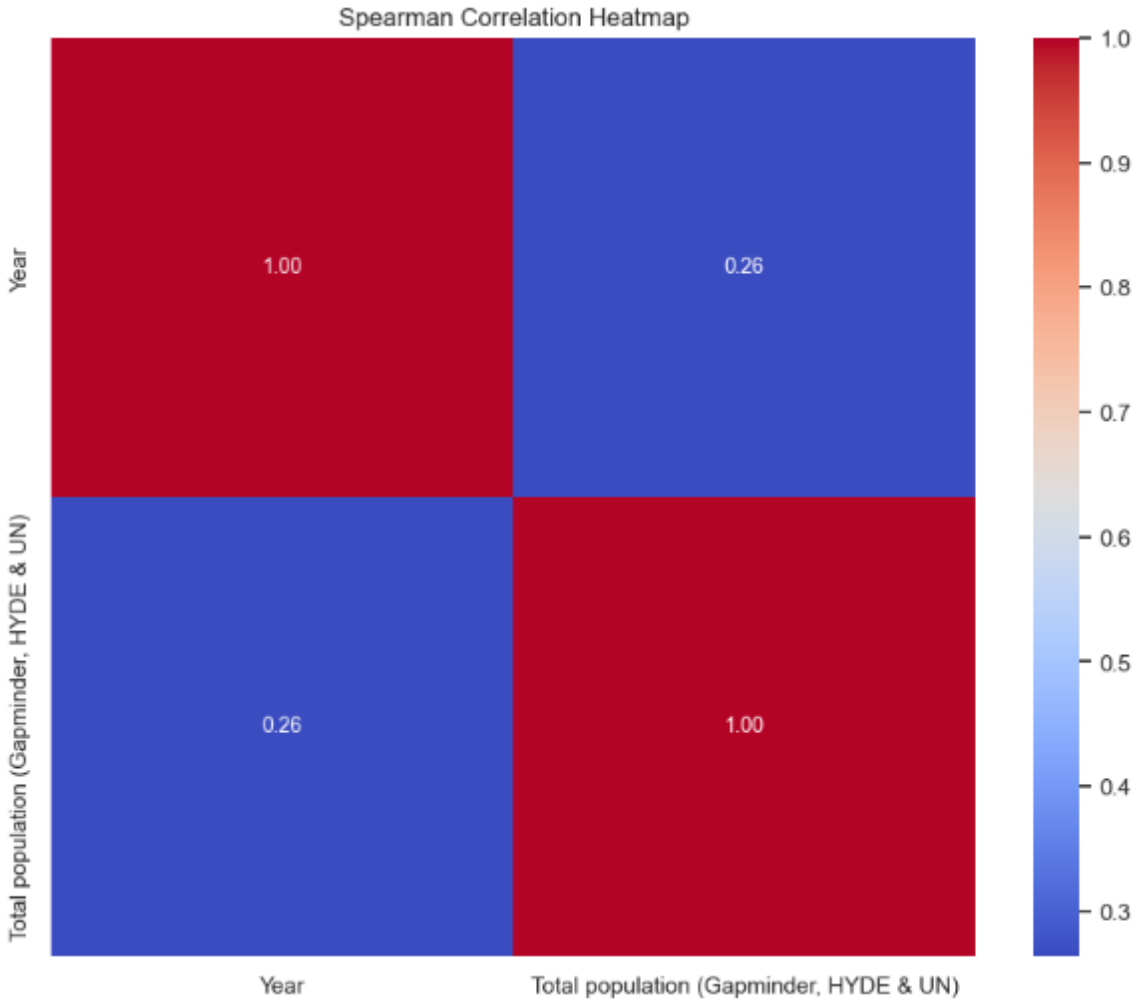


Table: Spearman's Correlation HeatMap

Decision Tree:

Like SVMs, Decision Trees are versatile Machine Learning algorithms that can perform both classification and regression tasks, and even multioutput tasks. They are very powerful algorithms, capable of fitting complex datasets.

GéronA. (2017). Hands-on Machine Learning with Scikit-Learn and TensorFlow :

Decision Tree Classifier Results:

Training Model: Decision Tree Classifier	
Accuracy:	0.032358965841776006
Accuracy:	0.03

Table: Module Evaluation

Model accuracy at 3.24% suggests poor performance. Improve with feature engineering, hyperparameter tuning, and alternative algorithms. Refine iteratively.

20%	
Accuracy 20%:	0.010587502594976126
Accuracy 20%:	0.01
30%	
Accuracy 30%:	0.011141097501902982
Accuracy 30%:	0.01

Table: Module Evaluation

Decision Tree Regressor:

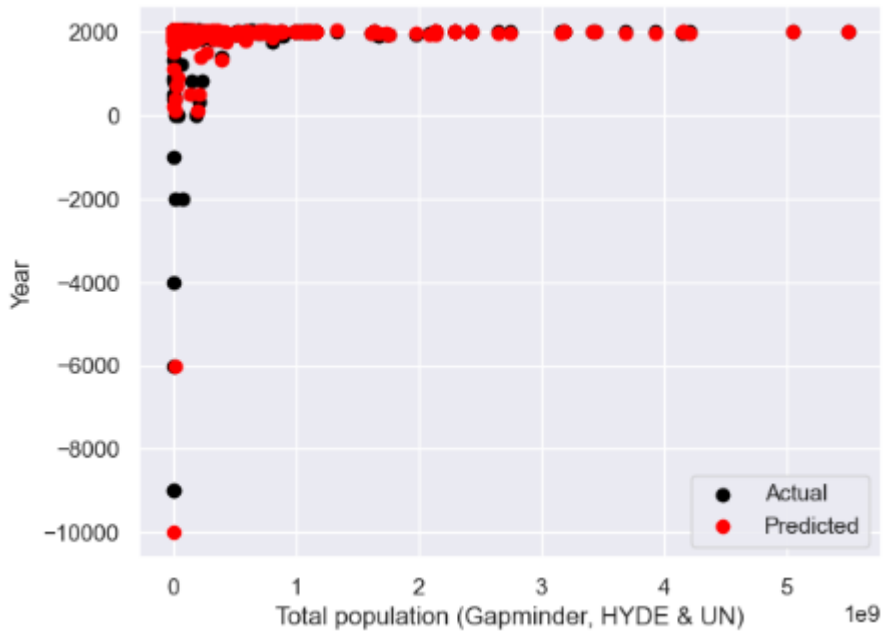
```
DecisionTreeRegressor
DecisionTreeRegressor(random_state=42)
```

Image:Decision Tree Regressor

DecisionTreeRegressor, a regression model using a decision tree algorithm, is created with a random state (42). Train, predict, evaluate, and refine.

Y_predication to evaluate the Model: y_pre=regressor.predict(X_test)

<u>Visualisation: Scatter plot Chart: Year & Total Population</u>



Graph:Scatter plot Chart: Year & Total Population

LinerRegression Model:

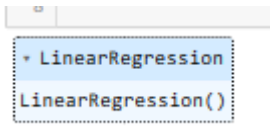


image:Linear Regression Model

Metrics:

with the Mean Square Error: This indicates that Baseline model required

Predicting the mean of the target variable	
Baseline Mean Squared Error:	84297.27022406812

Table: Predicting the mean of the Target Variable

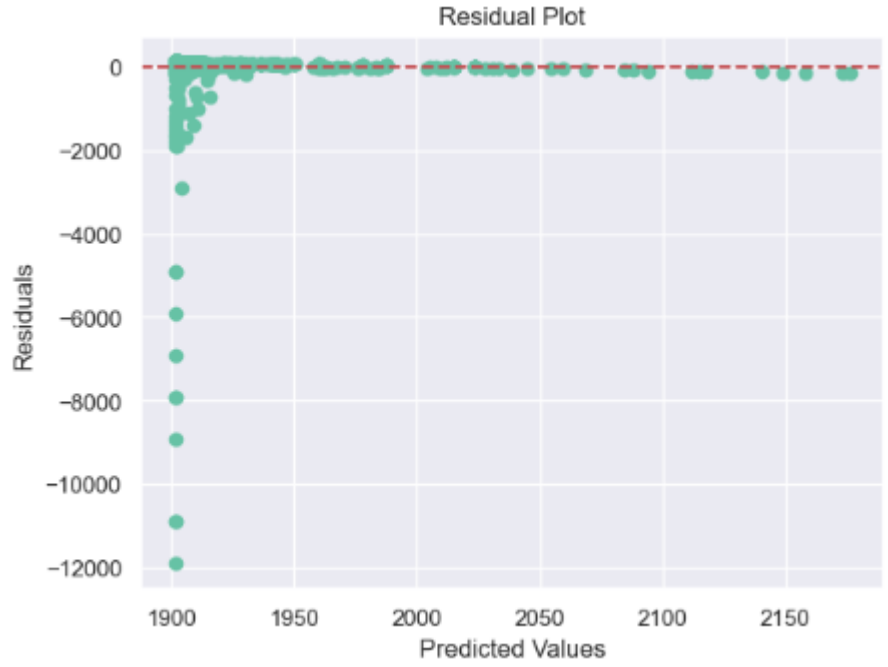
Model Evaluation	
Model Mean Squared Error:	84181.24754863395
Improvement over Baseline:	116.0226754341711

Table: Module Evaluation

Residual Analysis:

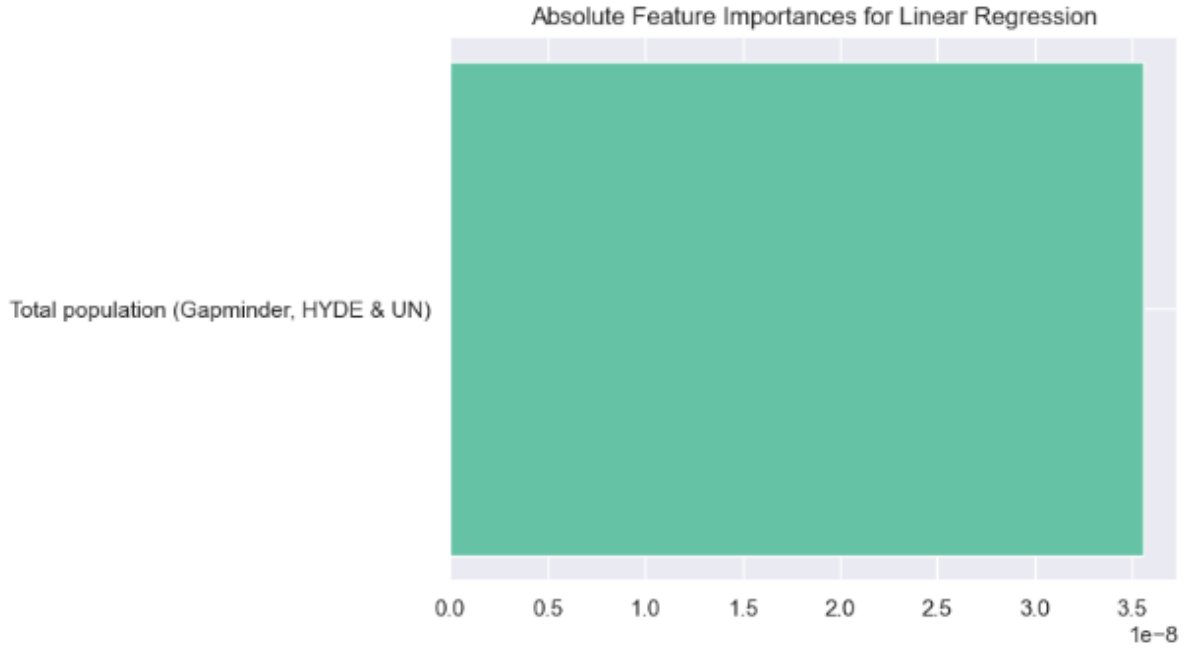
After training, analysing residuals (predicted vs. actual values gaps) is crucial to identify trends or areas requiring model improvement..

Residual Scatter plot Chart

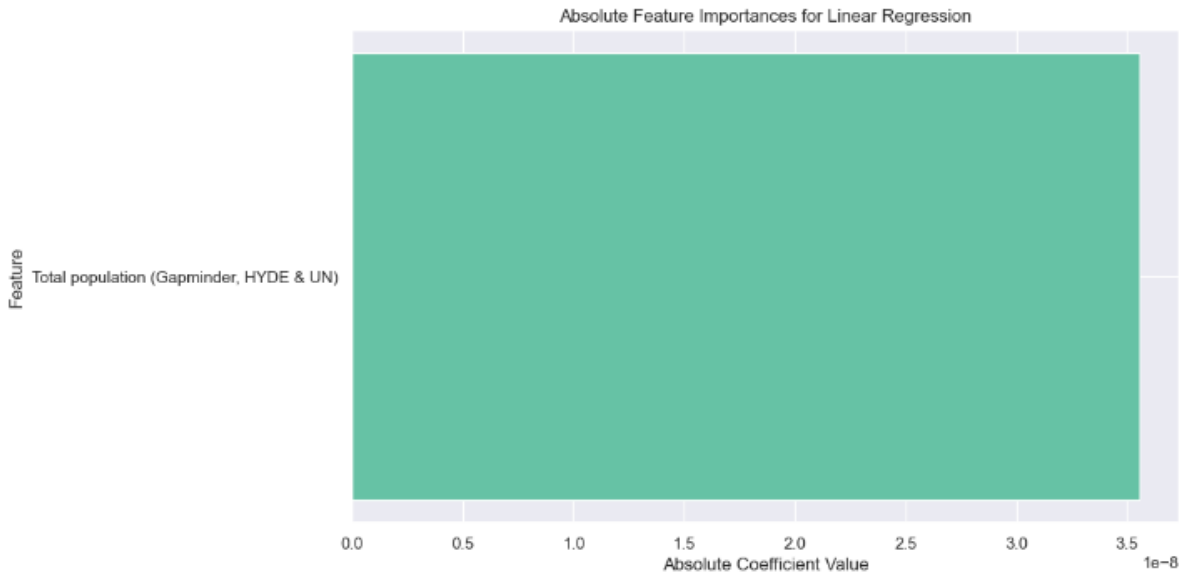


Graph:Residual Scatterplot chart

Feature importance is crucial: examine coefficients in linear regression, trees in models.



Graph:Absolute Feature Importance for Linear Regression



Graph:Absolute Feature Importance for Coefficient Value

Outliers Results

Outliers	
Empty DataFrame Columns	Total population (Gapminder, HYDE & UN)

Table: Outliers

No outliers found; an empty "Outliers" DataFrame indicates no significant deviation of data points from the general distribution, ensuring minimal impact on analyses or machine learning models.

Feature Selection:

By selecting the most relevant features, this feature selection process improves model performance and reduces overfitting. It is especially helpful when working with datasets that have many features.

Feature Selection	
Selected features	**Total population (Gapminder, HYDE & UN)**

Table: Feature Selection

Relevant features enhance model performance, prevent overfitting, and boost interpretability. Fewer features simplify model interpretation and reduce data dimensionality.

Feature Engineering:

Utilises scikit-learn to implement polynomial regression with Linear Regression model on polynomial features.

Continue with Splitting, training & evaluating the model:

5 Actuals & Predicated			
Actual:	**1996.0,**	**Predicted:**	**2031.3180512305207**
Actual:	**1891.0,**	**Predicted:**	**1901.062206135452**
Actual:	**1907.0,**	**Predicted:**	**1901.0109857873558**
Actual:	**1974.0,**	**Predicted:**	**1900.9086987372104**
Actual:	**1918.0,**	**Predicted:**	**1901.0630587430526**

Table: 5 Actuals & Predicated

Scatter Chart: Actuals Vs Predicted Values

Summary:

- The code shows the usual steps for a regression task.
- Dividing the dataset into train and test sets.
- Making a Linear Regression model and fitting it with the train data.
- Predicting on the test set.

Feature Scaling

Feature Scaling:

from sklearn.preprocessing import StandardScaler
 	
# Standardize features 	
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
 	
# Continue with splitting, training, and evaluating the model 	

Table: Feature Scaling Code

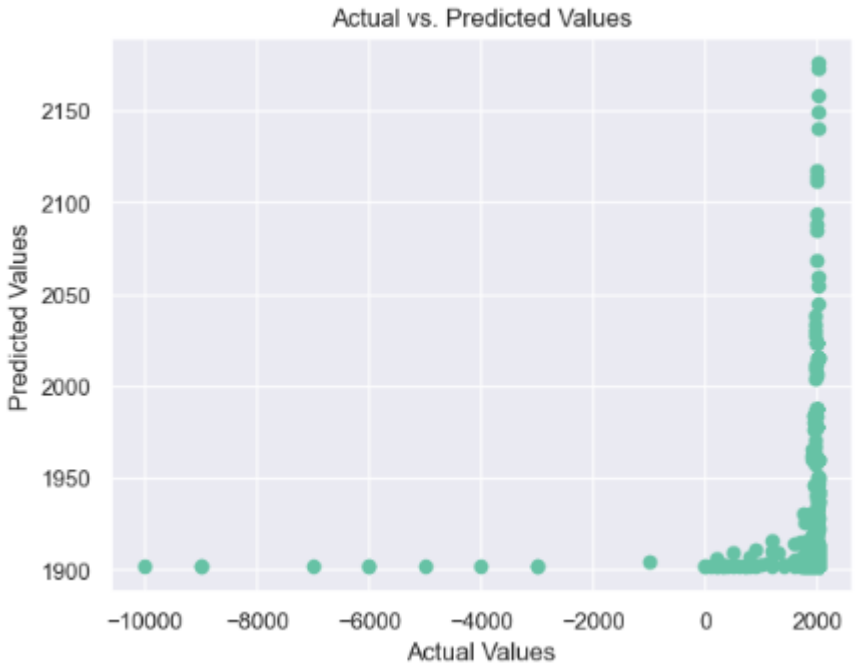
Continue with splitting, training, and evaluating the model
from sklearn.model_selection import train_test_split from sklearn.linear_model import LinearRegression X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42) model = LinearRegression() model.fit(X_train, y_train) y_pred = model.predict(X_test)

Table: Continuing with splitting, training & evaluating

The code uses scikit-learn's StandardScaler to standardise dataset X for models sensitive to feature scale.

for i in range(10): print(f"Actual: {y_test.iloc[i]}, Predicted: {y_pred[i]}")			
Actual:	**1996.0**	**Predicted:**	**2023.253041659319**
Actual:	**1891.0,**	**Predicted:**	**1901.7577200378253**
Actual:	**1907.0**	**Predicted:**	**1901.7329977252107**
Actual:	**1974.0**	**Predicted:**	**1901.6836417777106**
Actual:	**1918.0,**	**Predicted:**	**1901.7581316034684**
Actual:	**1882.0**	**Predicted:**	**1901.6517722676947**
Actual:	**1873.0,**	**Predicted:**	**1901.6339149411199**
Actual:	**1981.0**	**Predicted:**	**1901.6377384303896**

Table: Predication



Scatter Chart: Actuals Vs Predicted Values

Handling Categorical Variables

Hyperparameter Tuning:

Tune model performance with "Best Parameters: {'max_depth': 10}" for improved decision tree classifier results. Adjust max_depth carefully.

Best Parameters: {'max_depth': 10}

Table: Hyperparameter Tuning: Best Parameters

Cross Validation:

5-fold cross-validation	
Cross-Validation Mean MSE:	**99601.62856375493**
Cross-Validation Variance MSE:	**33291.53823576776**

Tabel: Cross Validation-5 folds

Results

Cross-Validation Mean MSE: ~99,601.63, gauging average squared difference. Variance MSE: ~33,291.54, indicating stable model performance across folds.

Advance Models:

Gradient Boosting Regressor	
Gradient Boosting Mean Squared Error:	**86352.0652390284**

Tabel: Gradient Boosting Regressor

Summary:
Gradient Boosting model yields a mean squared error (MSE) of ~89,843.02, signifying precise predictions on the dataset.

Model Tuning: Gradient Boosting Regressor

```
GradientBoostingRegressor
GradientBoostingRegressor(learning_rate=0.01, n_estimators=50)
```

Image:Gradient Boosting Regressor

GradientBoostingRegressor has a learning rate of 0.01 and 50 base learners, balancing accuracy, and computational efficiency.

Feature Importance:

Gradient Boosting Chart: Feature Importance

Cross Validation:

Results	
Cross-Validation Mean MSE:	**99515.02184506392**
Cross-Validation Variance MSE:	**33218.92847187535**

Table: Cross validation Result

Precision value of 0.0123 indicates low accuracy in correctly identifying positive instances. Higher precision values reflect improved accuracy.

Precision and Recall:

The precision value of 0.0123 suggests that the model's positive predictions are mainly false positives, indicating low accuracy.

Recall:
The recall of 0.0123 suggests the model detects only a small fraction of actual positives, with many false negatives.

F1 Score:
The F1 score of 0.0123 indicates a model imbalance, struggling with precision and recall in binary classification.

In practice, the choice between precision, recall, and F1 score depends on the specific requirements and constraints of the problem. It's important to consider the trade-offs between false positives and false negatives based on the application's goals. Adjusting the model or decision threshold may be necessary to achieve a more desirable balance between precision and recall, consequently improving the F1 score.

Conclusion:

The purpose of the dataset is to predict Per-Capita-Plastic-Waste-vs-GDP-Per-Capita to bring awareness. Targeted Variables have been tested.

By carefully preparing and exploring the data, it is possible to build strong models and generate useful insights. The results of various machine learning models, such as Decision Tree, Linear Regression, and Gradient Boosting Regressor, were evaluated, and their performance was analysed.

Recommendation & Contraints:

Based on the results, it is recommended to continue refining the models through feature engineering, hyperparameter tuning, and the use of alternative algorithms. Careful handling of missing data, feature selection, and feature scaling can also improve model performance. It is also important to carefully evaluate the precision, recall, and F1 score of the models to ensure their accuracy and effectiveness.

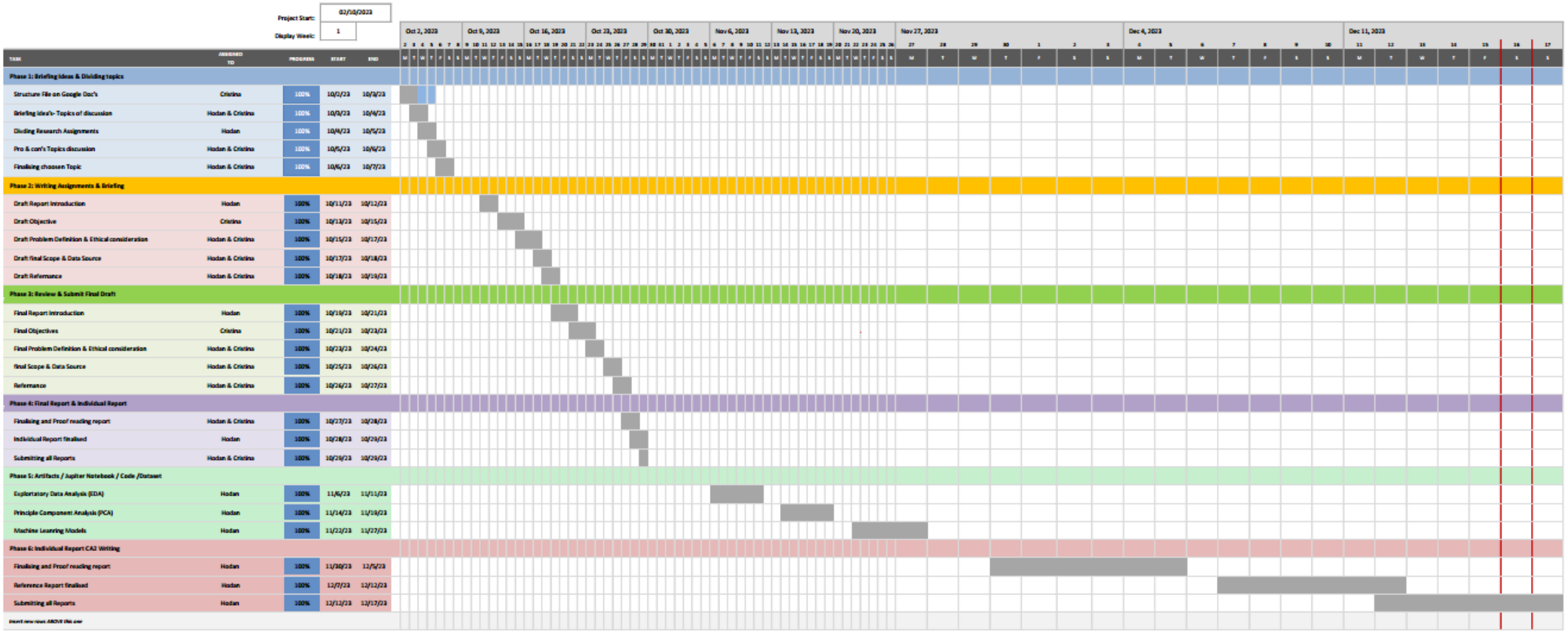
Future enhancements may involve broadening the scope of analysis to encompass multiple datasets, incorporating additional variables for a more comprehensive assessment. This expanded approach aims to provide readers and potential stakeholders, including recycling industries and sustainable advocates, with a holistic overview.

By considering diverse datasets and incorporating additional variables, the analysis strives to offer a deeper understanding of the subject matter, fostering more informed decision-making. Such improvements not only enhance the analytical depth but also cater to the interests of stakeholders invested in sustainable practices.

This multifaceted analysis is designed to empower readers and stakeholders to proactively address potential future repercussions, contributing to a more resilient and environmentally conscious approach in various sectors. As we evolve and refine our analytical methodologies, the goal is to create a robust framework that aligns with the principles and responsible resource management.

Gantt Chart: Hodan Mohamed Abdi CA2: Strategic Thinking

Strategic Thinking CA2



GitHub Link:

- https://github.com/sba23416/SBA23416_StrategicThinkingCA2.git

References:

1. www.kaggle.com. (n.d.). Global Plastic Pollution. [online] Available at: https://www.kaggle.com/datasets/sohamgade/plastic-datasets.

2. Datopian (n.d.). Daily_csv plastic monkey 78. [online] DataHub. Available at: https://www.datahub.io/gitchenze/daily_csv-plastic-monkey-78 [Accessed 15 Oct. 2023].

3. Our World in Data. (n.d.). Extrapolated change in plastic fate. [online] Available at: https://ourworldindata.org/grapher/plastic-fate-to-2050.

4. Oa, A. (2019). Public and Environmental Health Effects of Plastic Wastes Disposal: A Review. clinmedjournals.org, [online] 5(1). doi:https://doi.org/10.23937/2572-4061.1510021.

5. Borrelle, Stephanie B., et al. "Predicted Growth in Plastic Waste Exceeds Efforts to Mitigate Plastic Pollution." Science, vol. 369, no. 6510, 18 Sept. 2020, pp. 1515–1518, https://doi.org/10.1126/science.aba3656

6. Table 1: Role and Responsibilities

7. Table 2: Analysis of tasks

8. Table 3: Project planning

9. Table 4: Data sources

1. Aurelien Geron (2019). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Beijing: O'reilly.
2. GéronA. (2017). *Hands-on Machine Learning with Scikit-Learn and TensorFlow : concepts, tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA: O'Reilly Media.
3. Cross Validated. (n.d.). *Finding the Fitted and Predicted Values for a Statistical Model*. [online] Available at: https://stats.stackexchange.com/questions/34076/finding-the-fitted-and-predicted-values-for-a-statistical-model [Accessed 16 Dec. 2023].
4. datagy. (2022). *How to Calculate Mean Squared Error in Python • Datagy*. [online] Available at: https://datagy.io/mean-squared-error-python/.

5. matplotlib.org. (2023). *The Histogram (hist) Function with Multiple Data Sets — Matplotlib 3.7.1 Documentation*. [online] Available at: https://matplotlib.org/stable/gallery/statistics/histogram_multihist.html.
6. McCullum, N. (2023). *How to Concatenate DataFrames in Pandas*. [online] www.nickmccullum.com. Available at: https://www.nickmccullum.com/advanced-python/how-to-concatenate-pandas-dataframes/ [Accessed 16 Dec. 2023].
7. pandas.pydata.org. (n.d.). *pandas.DataFrame.select_dtypes — Pandas 1.4.2 Documentation*. [online] Available at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.select_dtypes.html.
8. pandas.pydata.org. (2023). *pandas.DataFrame.size — Pandas 2.1.4 Documentation*. [online] Available at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.size.html [Accessed 16 Dec. 2023].
9. pandas.pydata.org. (n.d.). *pandas.get_dummies — Pandas 1.2.0 Documentation*. [online] Available at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html.
10. scikit-learn (2019). *sklearn.model_selection.GridSearchCV — scikit-learn 0.22 Documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.
11. scikit-learn. (n.d.). *Classifier Comparison*. [online] Available at: https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html.
12. Scikit-learn.org. (2009). *3.2.4.3.6. sklearn.ensemble.GradientBoostingRegressor — scikit-learn 0.21.2 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html.
13. Scikit-learn.org. (2019). *sklearn.model_selection.cross_val_score — scikit-learn 0.22 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html.
14. scikit-learn.org. (n.d.). *sklearn.feature_selection.SelectKBest — scikit-learn 0.23.0 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html.

15. scikit-learn.org. (n.d.). *sklearn.impute.KNNImputer — scikit-learn 0.23.1 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html.
16. scikit-learn.org. (n.d.). *sklearn.impute.SimpleImputer — scikit-learn 0.24.1 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html.
17. scikit-learn.org. (n.d.). *sklearn.metrics.mean_squared_error — scikit-learn 0.24.2 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html.
18. scikit-learn.org. (n.d.). *sklearn.model_selection.cross_val_predict — scikit-learn 0.23.2 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_predict.html.
19. scikit-learn.org. (n.d.). *sklearn.preprocessing.PolynomialFeatures — scikit-learn 0.23.2 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html.
20. scikit-learn.org. (n.d.). *sklearn.tree.DecisionTreeRegressor — scikit-learn 0.23.2 Documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html.
21. seaborn.pydata.org. (n.d.). *An Introduction to Seaborn — Seaborn 0.12.0 Documentation*. [online] Available at: https://seaborn.pydata.org/tutorial/introduction.
22. Stack Overflow. (2016a). *Dealing with Missing Data in Pandas read_csv*. [online] Available at: https://stackoverflow.com/questions/39812493/dealing-with-missing-data-in-pandas-read-csv [Accessed 16 Dec. 2023].
23. Stack Overflow. (2016b). *How to Run Python Code with '%matplotlib inline'?* [online] Available at: https://stackoverflow.com/questions/55687832/how-to-run-python-code-with-matplotlib-inline.
24. Stack Overflow. (2021a). *How to Plot a Seaborn Boxplot for Each Month and Year*. [online] Available at: https://stackoverflow.com/questions/68475034/how-to-plot-a-seaborn-boxplot-for-each-month-and-year [Accessed 16 Dec. 2023].

25. Stack Overflow. (n.d.). *How to Print a Groupby Object*. [online] Available at: https://stackoverflow.com/questions/22691010/how-to-print-a-groupby-object [Accessed 16 Dec. 2023].
26. Stack Overflow. (n.d.). *How to Use Pandas Correctly to Print First Five Rows*. [online] Available at: https://stackoverflow.com/questions/63516367/how-to-use-pandas-correctly-to-print-first-five-rows [Accessed 16 Dec. 2023].
27. Stack Overflow. (n.d.). *Pandas pd.options.display.max_rows Not Working as Expected*. [online] Available at: https://stackoverflow.com/questions/57860775/pandas-pd-options-display-max-rows-not-working-as-expected [Accessed 16 Dec. 2023].
28. Stack Overflow. (2021b). *Python - Filter Pandas DataFrame by Substring Criteria*. [online] Available at: https://stackoverflow.com/questions/11350770/filter-pandas-dataframe-by-substring-criteria.
29. Stack Overflow. (n.d.). *Python - Find out the Percentage of Missing Values in Each Column in the Given Dataset*. [online] Available at: https://stackoverflow.com/questions/51070985/find-out-the-percentage-of-missing-values-in-each-column-in-the-given-dataset.
30. Stack Overflow. (2014). *Python - How Do I Count the NaN Values in a Column in Pandas DataFrame?* [online] Available at: https://stackoverflow.com/questions/26266362/how-do-i-count-the-nan-values-in-a-column-in-pandas-dataframe.
31. Stack Overflow. (2018). *Python - Find out the Percentage of Missing Values in Each Column in the Given Dataset*. [online] Available at: https://stackoverflow.com/questions/51070985/find-out-the-percentage-of-missing-values-in-each-column-in-the-given-dataset.
32. Stack Overflow. (n.d.). *Python - How Do I Find Numeric Columns in Pandas?* [online] Available at: https://stackoverflow.com/questions/25039626/how-do-i-find-numeric-columns-in-pandas.
33. Stack Overflow. (n.d.). *Python - How to Change the Figure Size of a Seaborn Axes or Figure Level Plot*. [online] Available at: https://stackoverflow.com/questions/31594549/how-to-change-the-figure-size-of-a-seaborn-axes-or-figure-level-plot.
34. Stack Overflow. (n.d.). *Python - Linear Regression - Get Feature Importance Using MinMaxScaler() - Extremely Large Coefficients*. [online] Available at:

https://stackoverflow.com/questions/65439843/linear-regression-get-feature-importance-using-minmaxscaler-extremely-larg.

35. Stack Overflow. (2019). *Python 3.x - How How iloc[:,1:] Works ? Can Any One Explain [:,1:] Params*. [online] Available at: https://stackoverflow.com/questions/56311638/how-how-iloc-1-works-can-any-one-explain-1-params.
36. Stack Overflow. (n.d.). *Python Pandas: Get Index of Rows Where Column Matches Certain Value*. [online] Available at: https://stackoverflow.com/questions/21800169/python-pandas-get-index-of-rows-where-column-matches-certain-value.
37. Stack Overflow. (n.d.). *Python- How to Import DecisionTree Classifier from sklearn.tree*. [online] Available at: https://stackoverflow.com/questions/71090431/python-how-to-import-decisiontree-classifier-from-sklearn-tree [Accessed 16 Dec. 2023].
38. Stack Overflow. (n.d.). *Residual Plot for Residual Vs Predicted Value in Python*. [online] Available at: https://stackoverflow.com/questions/62681388/residual-plot-for-residual-vs-predicted-value-in-python [Accessed 16 Dec. 2023].