Name: Pavithran Gnanasekaran
Email: pgnanase@buffalo.edu
UBID: pgnanase
UB Number: 50604192

PART 1

1. We began by importing the dataset in CSV format into a Pandas DataFrame using the **pd.read_csv** function. To gain an overview of the dataset, we used .**shape** to check the number of rows and columns, followed by **.describe** to get summary statistics. Next, we identified the numerical and categorical columns using **select_dtypes**. For missing values, we filled the numerical columns with the mean value using **.fillna** and the categorical columns with the mode to ensure completeness.

To maintain uniformity, categorical columns were capitalized using str.capltilize. Outliers were handled using the Interquartile Range (IQR) method, where we calculated the 30th and 70th percentiles of the data. The IQR was then computed as Q3 − Q1, and we determined the lower and upper bounds to filter data points within this range.

We encoded the categorical columns using cat.codes, checked the correlation values using .corr(), and dropped columns with low correlation values to reduce dimensionality. Finally, we normalized the numerical columns by subtracting the minimum value and dividing by the range (max - min), standardizing the data for further analysis

2. Dataset1:Penguins.csv
a. Overview of the Dataset

Domain:

This dataset relates to Penguins studies, specifically penguin species' physical and behavioral characteristics.

Type of Data:

The dataset includes both numerical (e.g., body mass, bill length) and categorical data (e.g., species, island, gender).

Number of Samples: The dataset contains 344 samples.

Features:

species: Categorical, representing the species of the penguin (Adelie, Gentoo etc).

island: Categorical, indicating the island where the penguin was observed.

calorie requirement: Numerical, representing daily caloric intake.

average sleep duration: Numerical, indicating the average hours of sleep.

bill_length_mm: Numerical, representing the bill length in millimeters.

bill_depth_mm: Numerical, representing the bill depth in millimeters.

flipper_length_mm: Numerical, indicating the flipper length in millimeters.

body_mass_g: Numerical, representing the body mass in grams.

gender: Categorical, indicating the gender of the penguin (male or female).

year: Numerical, indicating the year of observation.

b. Key Statistics
Mean, Standard deviation, Number of missing values:

Calorie Requirement: [5270.002907, 1067.959116 , 0]

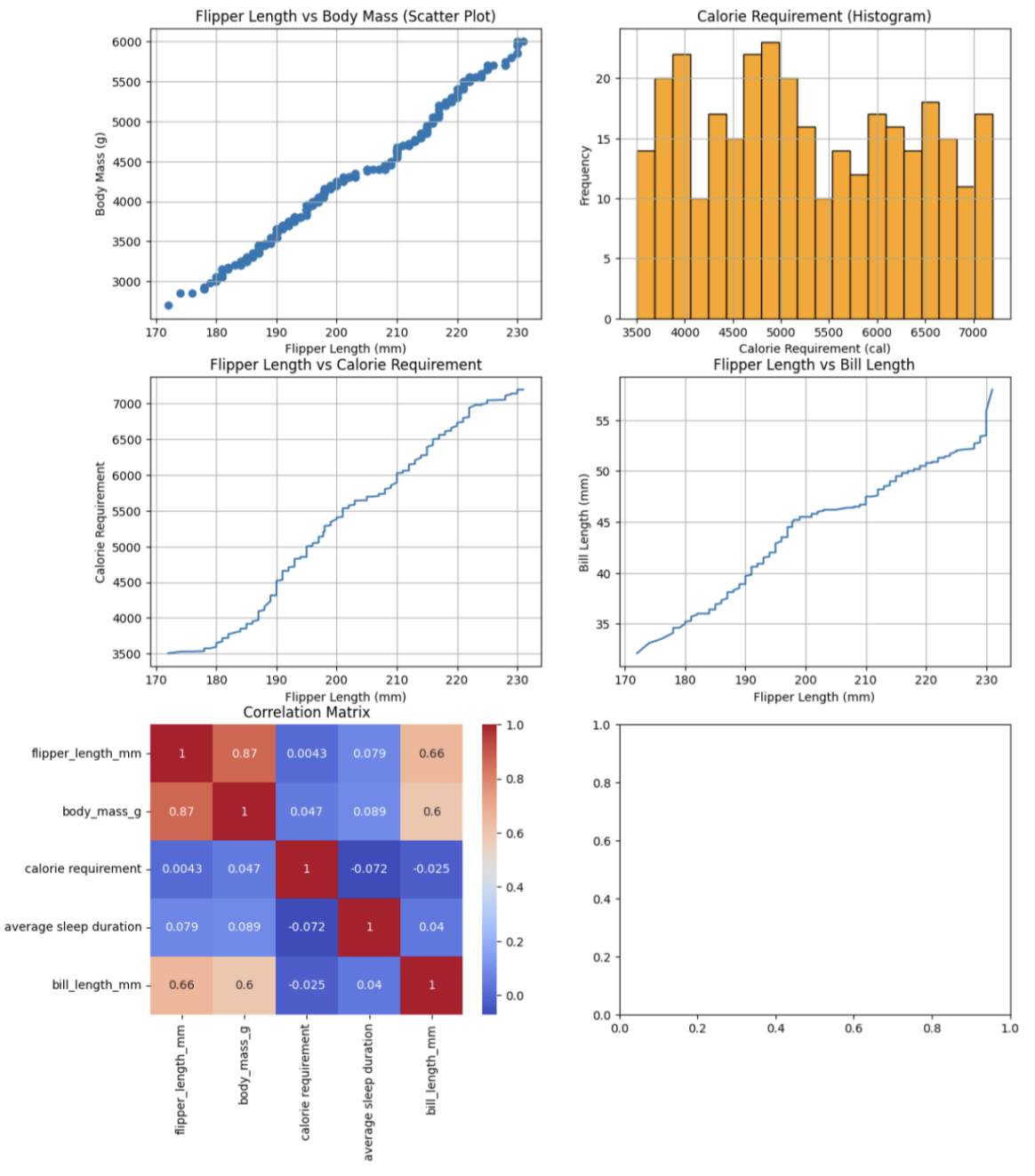
Average Sleep Duration: [10.447674 , 2.265895 , 0]

Bill Length: [45.494214 , 10.815787 , 7]

Bill Depth: [18.018318, 9.241384, 11]

Flipper Length [197.764881, 27.764491, 8]

Body Mass:[4175.463127, 858.713267, 5]



1. Flipper Length vs Body Mass (Scatter Plot):

This scatter plot shows a strong positive correlation between flipper length and body mass, suggesting that penguins with longer flippers tend to have higher body mass.

2. Calorie Requirement (Histogram):

The histogram indicates that calorie requirements vary widely across the dataset, with a somewhat uniform distribution across different ranges. There are peaks in the frequency for certain calorie ranges, suggesting that penguins may have certain typical calorie requirements.

3. Flipper Length vs Calorie Requirement (Line Plot):

The line plot demonstrates a positive correlation between flipper length and calorie requirement. Penguins with longer flippers tend to have higher calorie requirements, which might be related to their larger body size or higher energy expenditure.

4. Flipper Length vs Bill Length (Line Plot):

This line plot also shows a positive trend, indicating that penguins with longer flippers tend to have longer bills, which might reflect a proportional growth relationship between different physical characteristics.

5. Correlation Matrix:

The correlation matrix highlights the relationships between various numerical features. For instance:

Flipper Length and Body Mass have a strong positive correlation (0.87).

Calorie Requirement has a low correlation with most features, indicating that it may not be directly influenced by other physical traits.

Bill Length and Flipper Length are moderately correlated (0.66).

Dataset 2

Overview of the Dataset:

Domain:

This dataset is related to the diamond mining industry and focuses on the relationship between the number of diamonds mined, their physical characteristics, pricing, and socio-economic factors like average US salary.

Type of Data:

The dataset includes both numerical and categorical features, with a focus on the physical attributes of diamonds and their mining output.

Number of Samples: The dataset consists of 53,940 rows (observations).

Features:

Unnamed: 0: Index column

carat: The weight of diamonds in carats

average us salary: The average US salary in dollars

number of diamonds mined (millions): The number of diamonds mined (in millions)

depth: The depth percentage of diamonds

table: The width of the top of the diamond relative to its widest point

price: The price of diamonds in dollars

x, y, z: Physical dimensions of the diamond (length, width, height)

b. Key Statistics

Mean, Standard deviation, Number of missing values:

carat : [0.797823, 0.473747, 2867]

average us salary: [39521.990100, 5486.892971, 0]

number of diamonds mined (millions): [2.902669, 1.325985 , 0]

depth: [61.750175, 1.433485, 2074]

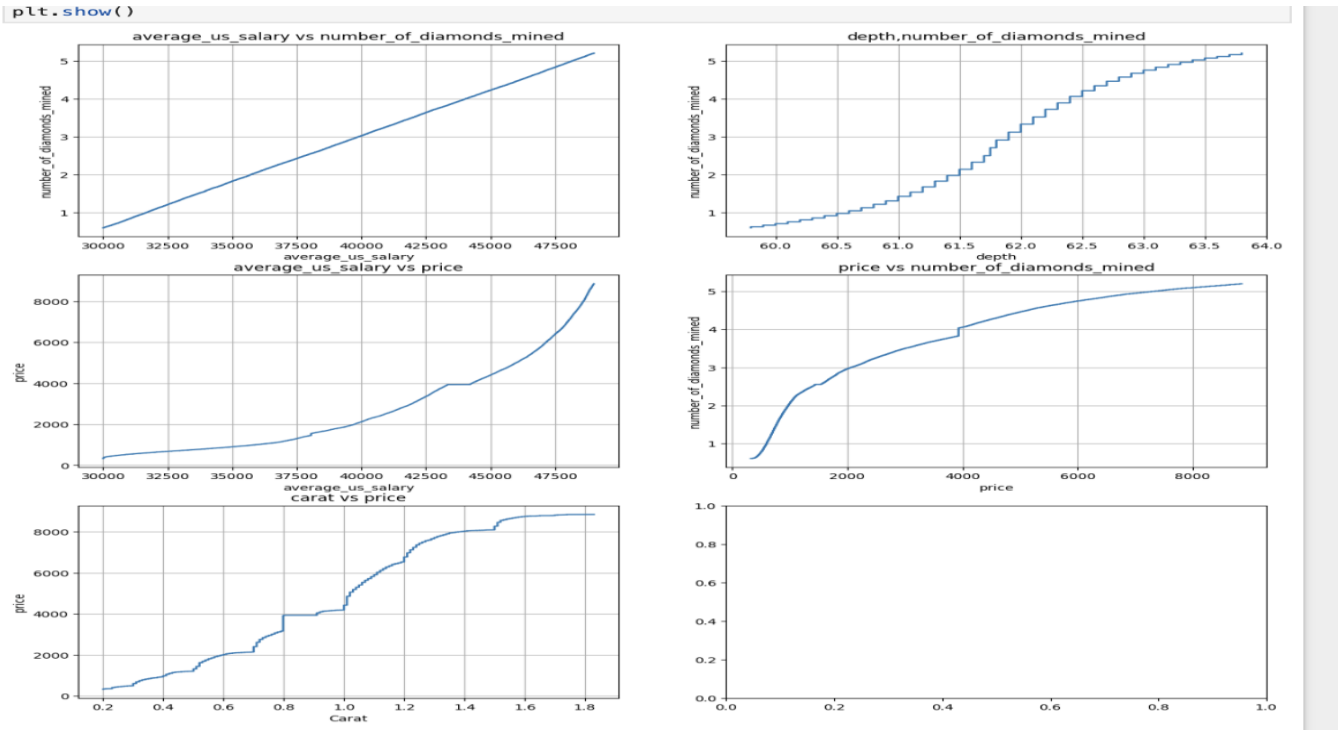
table: [197.764881, 27.764491, 8]

x: [5.731451, 1.121433, 2913]

y :[5.734517 , 1.142543 , 1732]

z : [3.538203, 0.706057 , 2408]

C)



1. Average US Salary vs. Number of Diamonds Mined:

   There appears to be a positive linear relationship between average US salary and the number of diamonds mined. As average US salary increases, the number of diamonds mined also tends to increase.

2. Depth vs. Number of Diamonds Mined:

   The relationship between depth and the number of diamonds mined seems to be somewhat complex. There might be a slight positive correlation, indicating that deeper mines might yield more diamonds.

3. Average US Salary vs. Price:

   There appears to be a positive relationship between average US salary and the price of diamonds. This might suggest that higher-income individuals are more likely to purchase more expensive diamonds.

4. Price vs. Number of Diamonds Mined:

   The relationship between price and the number of diamonds mined seems to be somewhat nonlinear. There might be a slight increase in the number of diamonds mined as prices rise.

5. Carat vs. Price:

   There's a strong positive correlation between carat weight and price. Larger diamonds generally have higher prices.

Dataset 3:

Overview of the Dataset:

Domain:

This dataset is related to the emission by country and focuses on the relationship between the total emissions, GDP, per capita, emissions due to oil, Gas, Coat, Cement, Flaring and other

Type of Data: The dataset primarily consists of numerical features, with historical data on emissions and related attributes across countries and years.

Number of Samples: The dataset contains 63,104 rows (observations).

Features:

Year: The year of the recorded data.

Temperature: The average global or country-specific temperature.

GDP Per Capita (USD): The GDP per capita in USD.

Coal, Oil, Gas, Cement, Flaring, Other: Quantities of various fossil fuels and materials contributing to emissions.

Per Capita: Emissions per capita.

Total: Total emissions for the given year and country.

Key Statistics (Mean, Standard deviation, Number of missing values):

Year: [1888.267097, 121.874172, 979]

Temperature: [49.497813, 17.292092, 0]

GDP Per Capita (USD): [39026.539015, 10975.539432, 0]

Coal: [127.387271, 398.439103, 41307]

Oil: [153.480038, 394.046238, 41330]

Gas: [125.162671, 301.757895, 41387]

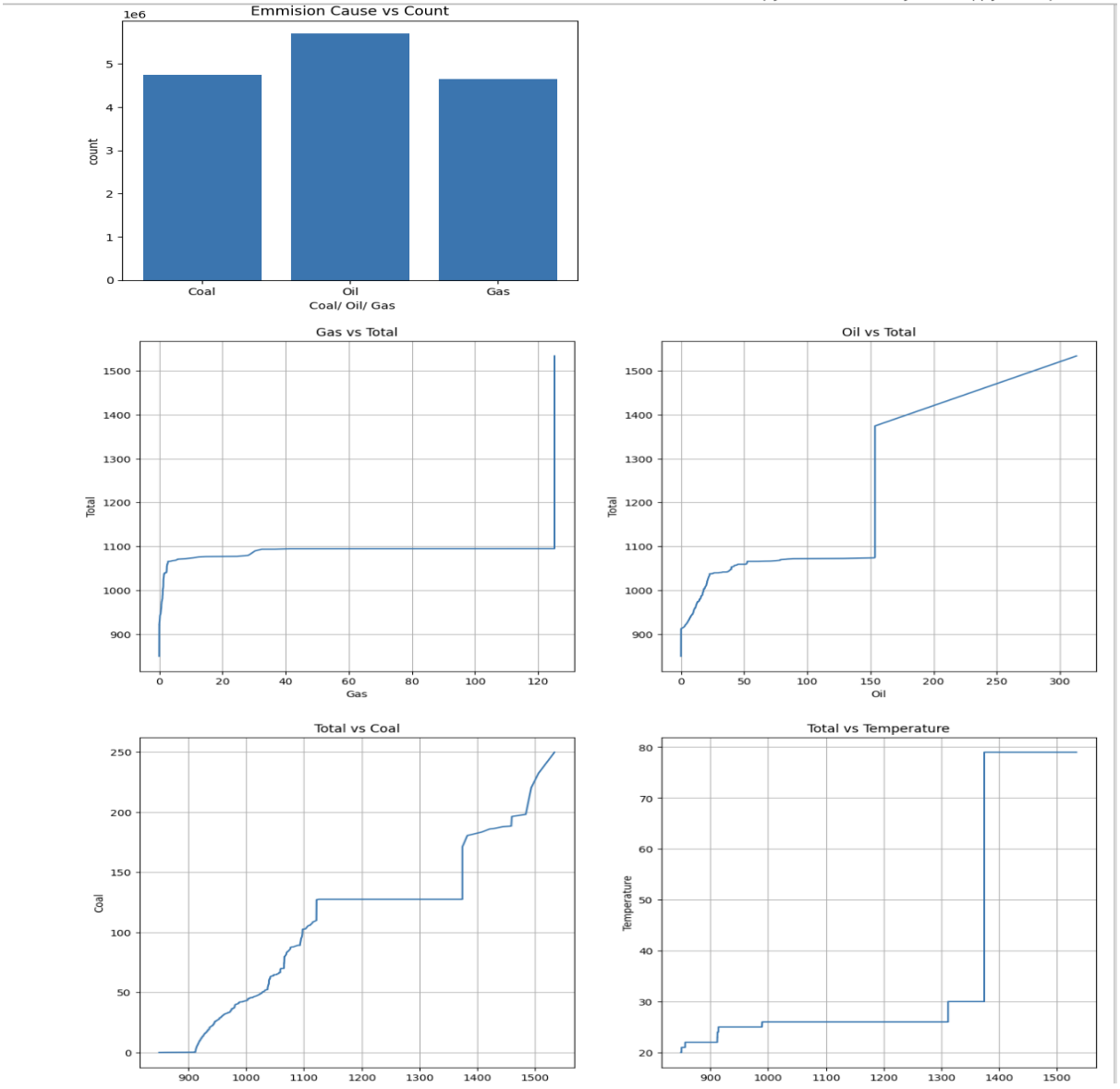
Cement: [62.599364, 201.658675, 42616]

Flaring: [56.074327, 196.327663, 41766]

Other: [849.395127, 217.631433, 60419]

Per Capita: [121.565443, 271.259969, 43712]

Total: [1374.098797, 1050.283290, 723]



1. Emission Cause vs Count: Oil is the largest contributor to emissions, followed by coal and gas. This highlights oil-related activities as a key area for emission reduction. Addressing oil use could significantly reduce overall emissions.

2. Gas vs Total: Total emissions increase as gas usage rises, though there are periods where emissions remain constant. This suggests other factors might also be influencing total emissions besides gas consumption.

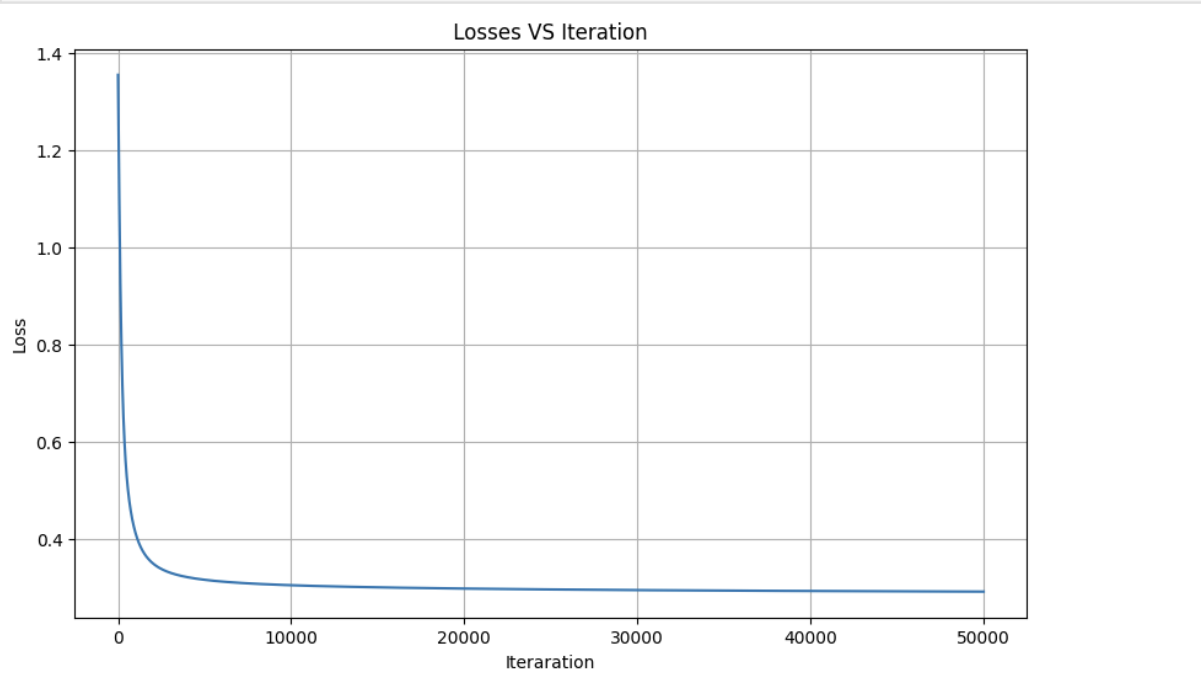
3. Oil vs Total: A strong correlation exists between oil usage and total emissions, with noticeable jumps at intervals. This implies oil has a significant impact on overall emission levels.

4. Total vs Coal: Coal usage shows a gradual but uneven impact on total emissions, with sharp increases at certain points. The volatility suggests coal may have additional influencing factors.

5. Total vs Temperature: The temperature rises as total emissions increase, though the pattern is irregular, likely due to missing data being filled with mean values. This indicates potential environmental effects from rising emissions.

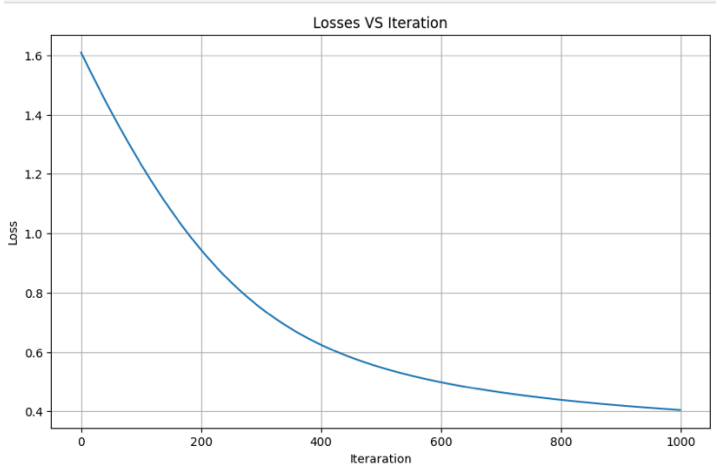
PART 2

1. Best Accuracy: 90.76923076923077 , learning rate= 0.005 , iterations= 50000
2. Loss vs Iteration Graph:



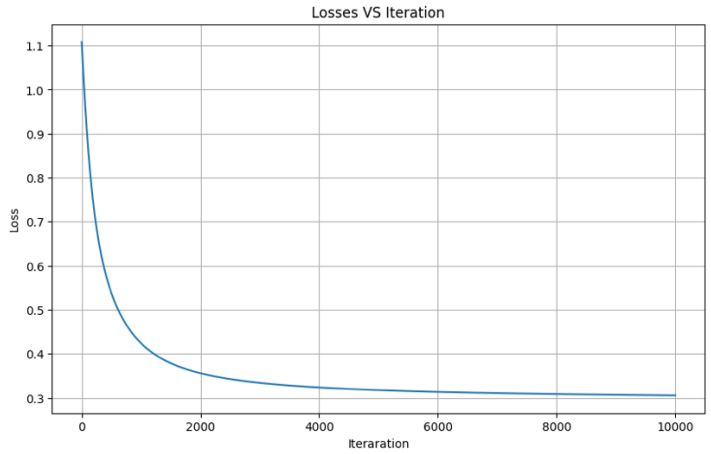
Initially, the loss is high and decreases sharply as iterations progress. After several iterations, the loss begins to saturate, or its decrease becomes negligible as the iterations continue.

3. a) learning rate= 0.005, iterations = 1000, Accuracy = 75.38461538461539



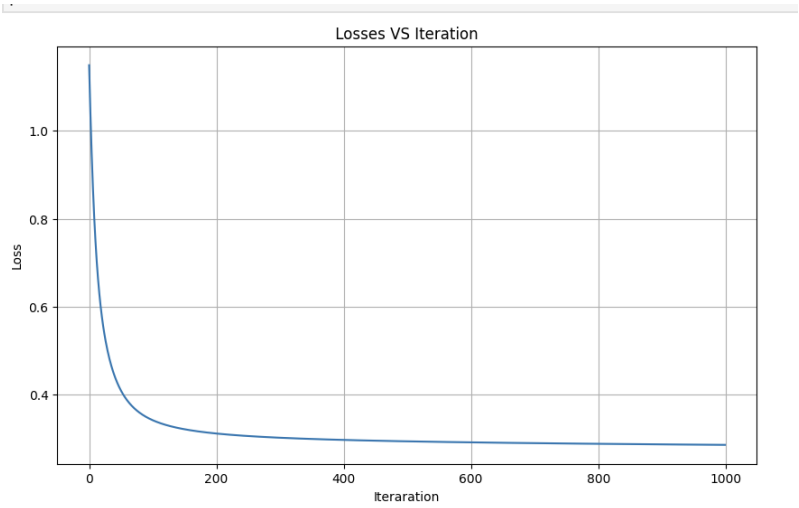
Setup2:

Learning rate= 0.005, iterations= 10000, Accuracy= 89.23076923076924



Setup3:

Learning rate= 0.1, iterations= 1000, Accuracy= 83.07692307692308



The loss decreases as the iterations increase up to a certain point, after which the decrease becomes negligible as the iterations continue.

The learning rate should be set to an ideal value for effective model training. If the learning rate is too high, it can cause the loss to oscillate between extreme values, potentially diverging to infinity or dropping to zero. On the other hand, a very low learning rate may result in slow convergence, leading to an under-trained model.

Benefits of Using Logistic Regression

1.Simplicity and Interpretability: Easy to implement and interpret with clear relationships between features and outcomes.

2. Efficiency: Computationally efficient, handling large datasets well without extensive resources.

3. Probabilistic Outputs: Provides probabilities, allowing for nuanced predictions beyond binary outcomes.

4. Robust to Noise: Performs reasonably well even with noisy or irrelevant features.

5. No Need for Feature Scaling: Less sensitive to feature scales, eliminating the need for normalization.

Drawbacks of Using Logistic Regression

1. Linearity Assumption: Assumes a linear relationship between features and log odds, limiting its effectiveness with nonlinear patterns.

2. Limited Complexity: May struggle with complex datasets compared to advanced models like decision trees or neural networks.

3. Binary Outcomes: Primarily suited for binary classification, with multi-class handling being less straightforward.

4. Sensitive to Outliers: Performance can be affected by outliers, skewing results.

5. Feature Independence: Assumes features are independent; multicollinearity can lead to unreliable coefficient estimates.

PART 3

Dataset Used: Diamond.csv

Domain:

This dataset is related to the diamond mining industry and focuses on the relationship between the number of diamonds mined, their physical characteristics, pricing, and socio-economic factors like average US salary.

Type of Data:

The dataset includes both numerical and categorical features, with a focus on the physical attributes of diamonds and their mining output.

Number of Samples: The dataset consists of 53,940 rows (observations).

Features:

Unnamed: 0: Index column

carat: The weight of diamonds in carats

average us salary: The average US salary in dollars

number of diamonds mined (millions): The number of diamonds mined (in millions)

depth: The depth percentage of diamonds

table: The width of the top of the diamond relative to its widest point

price: The price of diamonds in dollars

x, y, z: Physical dimensions of the diamond (length, width, height)

b. Key Statistics
Mean, Standard deviation, Number of missing values:

carat : [0.797823, 0.473747, 2867]

average us salary: [39521.990100, 5486.892971, 0]

number of diamonds mined (millions): [2.902669, 1.325985 , 0]

depth: [61.750175, 1.433485, 2074]

table: [197.764881, 27.764491, 8]

x: [5.731451, 1.121433, 2913]

y :[5.734517 , 1.142543 , 1732]

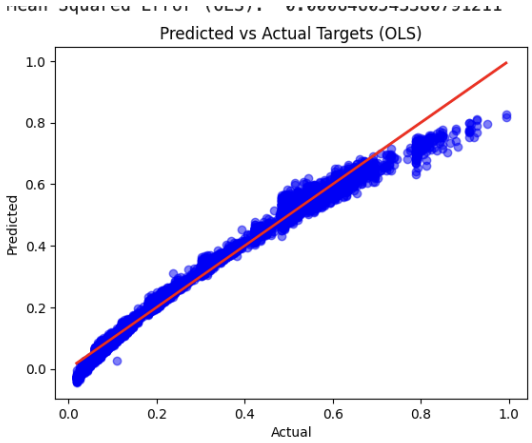
z : [3.538203, 0.706057 , 2408]

2.

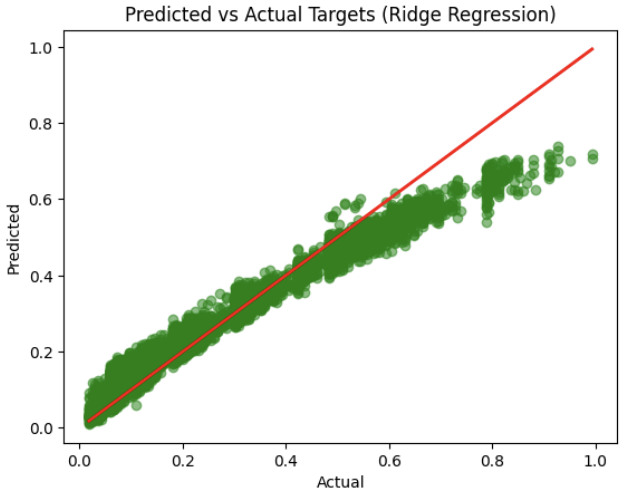
Mean Squared Error (MSE) for Linear regression: 0.0006460543380791211

Mean Squared Error (MSE) for Ridge regression: 0.0024989318740349918

Linear Regression Graph (predicted vs actual:



Ridge Regression Graph (predicted vs actual values):



Benefits and Drawbacks of Using OLS for Weight Computation

Benefits:

Closed-Form Solution: Ordinary Least Squares (OLS) provides a straightforward analytical solution for weight computation.

Efficiency: OLS is computationally efficient for small to moderate-sized datasets.

Interpretability: The model coefficients from OLS are easy to interpret.

Drawbacks:

Sensitivity to Outliers: OLS can be significantly affected by outliers, leading to skewed results and unreliable predictions.

Assumptions: It assumes linearity, homoscedasticity, and independence of errors, which may not hold true in many real-world scenarios.

Multicollinearity: High correlation among independent variables can inflate the variance of the coefficient estimates, making them unstable.

Strengths and Weaknesses of Linear Regression

Strengths:

Simplicity: Linear regression is simple to understand and implement, making it a good starting point for regression tasks.

Interpretability: The results are easy to interpret, allowing insights into the relationship between variables.

Fast Training: Linear regression trains quickly, making it suitable for large datasets.

Weaknesses:

Linear Assumption: It assumes a linear relationship between predictors and the response variable, which may not be appropriate for all datasets.

Overfitting: In complex datasets, linear regression may overfit, especially if the model includes too many features.

Limited Flexibility: It struggles to model complex relationships, requiring feature engineering or transformations for non-linear patterns.

Motivation for L2 Regularization:

L2 regularization (Ridge regression) helps mitigate overfitting by adding a penalty term to the loss function based on the size of the coefficients. This discourages overly complex models and improves generalization on unseen data.

Improvements Over Linear Regression:

Reduced Overfitting: By penalizing large coefficients, Ridge regression effectively reduces the risk of overfitting, especially in high-dimensional datasets.

Handles Multicollinearity: Ridge regression can stabilize the coefficient estimates in the presence of multicollinearity, providing more reliable predictions.

Benefits of Ridge Regression:

Better Generalization: It Tends to perform better on unseen data compared to plain linear regression.

Regularized Coefficients: The Coefficients are shrunk towards zero, leading to simpler models that are less sensitive to noise.

Limitations Compared to Linear Regression:

Interpretability: The coefficients in Ridge regression are more complex due to the regularization, making them less straightforward to explain.

Computational Complexity: The addition of the regularization term can increase computational complexity, particularly in very large datasets.

Bias Introduction: Ridge regression introduces bias in the estimates, which can be a drawback if the true model is already simple and linear.

PART 4

Dataset Used: emission_by_country.csv

Domain:

This dataset is related to the emission by country and focuses on the relationship between the total emissions, GDP, per capita, emissions due to oil, Gas, Coat, Cement, Flaring and other

Type of Data: The dataset primarily consists of numerical features, with historical data on emissions and related attributes across countries and years.

Number of Samples: The dataset contains 63,104 rows (observations).

Features:

Year: The year of the recorded data.

Temperature: The average global or country-specific temperature.

GDP Per Capita (USD): The GDP per capita in USD.

Coal, Oil, Gas, Cement, Flaring, Other: Quantities of various fossil fuels and materials contributing to emissions.

Per Capita: Emissions per capita.

Total: Total emissions for the given year and country.

Key Statistics (Mean, Standard deviation, Number of missing values):

Year: [1888.267097, 121.874172, 979]

Temperature: [49.497813, 17.292092, 0]

GDP Per Capita (USD): [39026.539015, 10975.539432, 0]

Coal: [127.387271, 398.439103, 41307]

Oil: [153.480038, 394.046238, 41330]

Gas: [125.162671, 301.757895, 41387]

Cement: [62.599364, 201.658675, 42616]

Flaring: [56.074327, 196.327663, 41766]

Other: [849.395127, 217.631433, 60419]

Per Capita: [121.565443, 271.259969, 43712]

Total: [1374.098797, 1050.283290, 723]

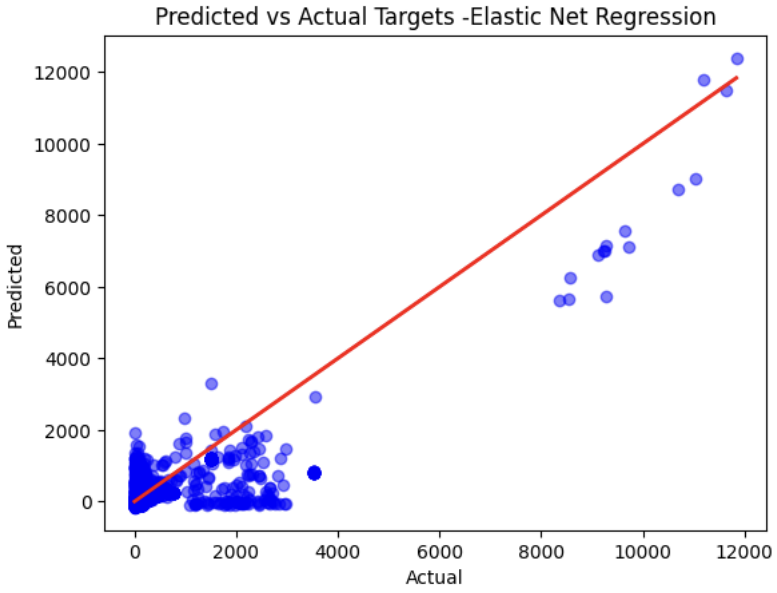
2. The loss Value for different weight initialization of the model

1. Using zero initialization , final loss: 42529.410130859986

2. Using random initialization, final loss: 42529.55442434408

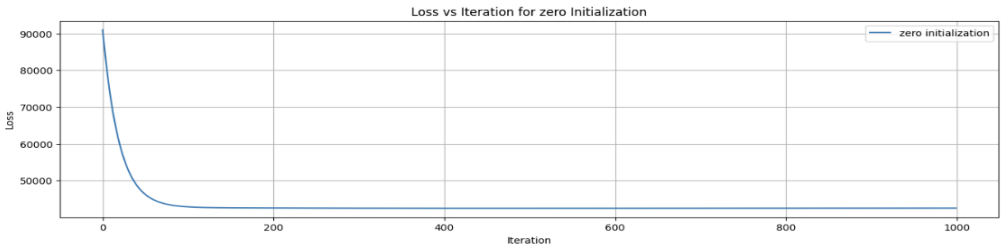
3. Using xavier initialization, final loss: 42529.25212540156

Actual vs Predicted value plot

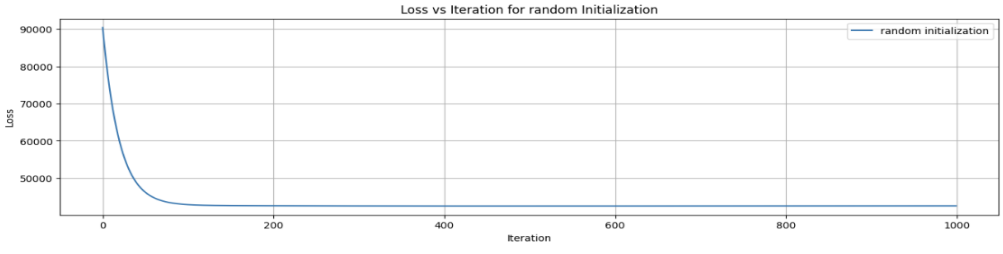


Effects of different weight initialization:

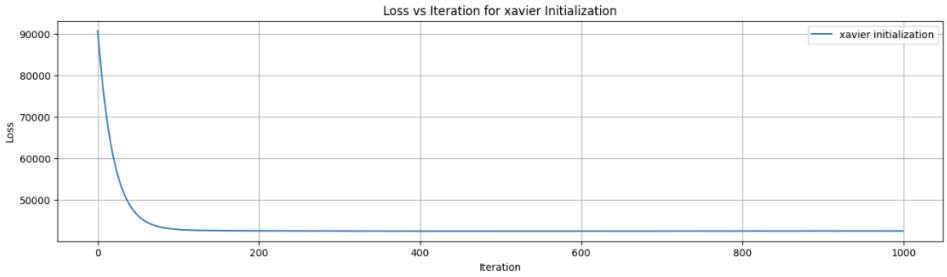
```
Using zero initialization:
Iteration 0: Loss = 91006.78951518751
Iteration 100: Loss = 42869.43221518909
Iteration 200: Loss = 42560.0021677685
Iteration 300: Loss = 42510.30169231144
Iteration 400: Loss = 42489.34912639852
Iteration 500: Loss = 42485.995203224295
Iteration 600: Loss = 42491.471458973996
Iteration 700: Loss = 42500.68978664603
Iteration 800: Loss = 42510.90195271425
Iteration 900: Loss = 42520.71301292637
Weights: [-23.10221059  27.05314174  -4.78356594 156.96401839   0.37176165
   0.95697565  71.55223616 114.98854233  32.71916525  35.00660266
  -3.99773177   6.65221068]
final loss:  42529.410130859986
```




```
Using random initialization:
Iteration 0: Loss = 90456.29947914794
Iteration 100: Loss = 42862.84627920926
Iteration 200: Loss = 42558.4071151768
Iteration 300: Loss = 42509.61404306312
Iteration 400: Loss = 42489.11353510841
Iteration 500: Loss = 42485.99572684731
Iteration 600: Loss = 42491.587613133976
Iteration 700: Loss = 42500.85290888707
Iteration 800: Loss = 42511.074359110244
Iteration 900: Loss = 42520.87550809798
Weights: [-23.09409581  27.04448321  -4.78460693 156.98890386   0.37172331
   0.95708029  71.52805989 114.98794415  32.72190549  35.00939335
  -3.99719069   6.65286187]
final loss:  42529.55442434408
```



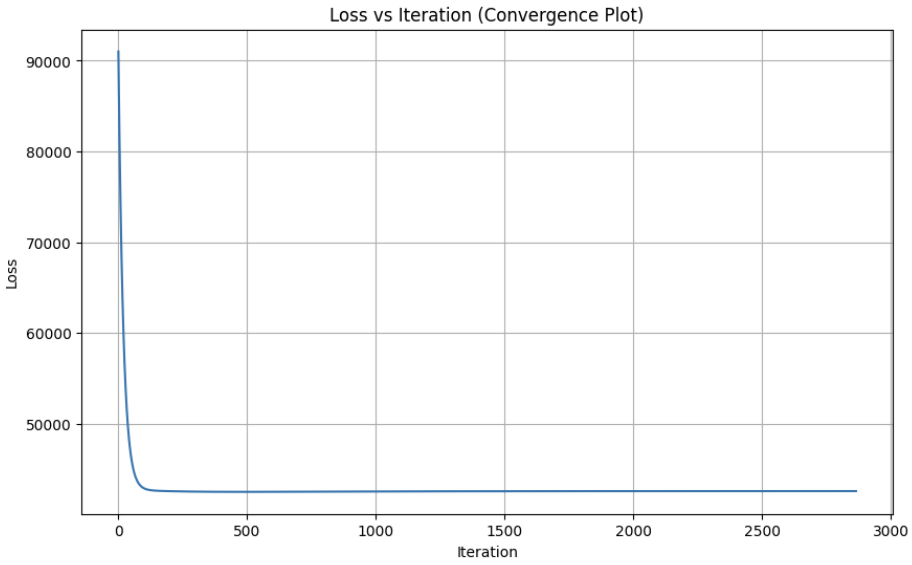
```
Using xavier initialization:
Iteration 0: Loss = 90719.41392473296
Iteration 100: Loss = 42873.686615146755
Iteration 200: Loss = 42561.933165008326
Iteration 300: Loss = 42511.10370015184
Iteration 400: Loss = 42489.63184118546
Iteration 500: Loss = 42486.00847775465
Iteration 600: Loss = 42491.352091836285
Iteration 700: Loss = 42500.515514036444
Iteration 800: Loss = 42510.71538456555
Iteration 900: Loss = 42520.5359382017
Weights: [-23.12179971  27.07343007  -4.78216709 156.9319122    0.37183021
   0.95682991  71.58313274 114.9884513   32.71561425  35.00326624
  -3.99906754   6.65154702]
final loss:  42529.25212540156
```



We can see loss is less for weights initialised by Xavier initialization compared to zero initialization and random initialization.

Early Stopping:

```
Iteration 0: Loss = 91006.78951518751
Iteration 1000: Loss = 42529.49188186058
Iteration 2000: Loss = 42564.84095382464
Stopping early at iteration 2865 due to small gradient.
Using gradient condition -0.01 < gradient < 0.01
Weights: [-24.62651487  28.48866862  -4.93076509 160.91393013   0.35906213
   0.97064654  67.47467085 114.83765666  33.16211691  35.44517667
  -4.13103828   6.78155907]
Final loss:  42568.120724224034
```



The early stopping due low gradient condition or limited iteration increase the final loss of the model thereby reducing accuracy of the model. Therefore iteration should be good enough for model to learn efficiently also gradient condition should be kept low.

3. **Comparison of Elastic Net Regularization with L1 and L2 Regularization**

Elastic Net regularization combines both L1 (Lasso) and L2 (Ridge) regularization, allowing for a more balanced approach to feature selection and coefficient shrinkage. While L1 regularization encourages sparsity by setting some coefficients to zero, making it effective for feature selection, L2 regularization shrinks coefficients uniformly, which helps in retaining all features but may not select the most relevant ones.

In applications where the number of predictors is much larger than the number of observations, or where predictors are highly correlated, Elastic Net provides advantages over L1 and L2 alone. It can select groups of correlated features together and mitigate the limitations of Lasso, which may arbitrarily select one variable from a group of correlated variables while ignoring others.

Benefits and Potential Drawbacks of Using Elastic Net Regression and Gradient Descent

Benefits:

Flexibility: Elastic Net combines the strengths of both L1 and L2 regularization, allowing for a more adaptable model that can effectively handle various types of data and relationships.

Feature Selection: The L1 component encourages sparsity, which can simplify models by excluding irrelevant features, leading to better interpretability.

Handling Multicollinearity: Elastic Net is particularly effective in situations with multicollinearity, where features are highly correlated, by selecting and retaining a group of correlated features.

Potential Drawbacks:

Hyperparameter Tuning: Elastic Net requires careful tuning of its hyperparameters (lambda1 and lambda2) to balance the contributions of L1 and L2 regularization, which can increase model complexity and computational cost.

Convergence Issues: Using gradient descent for optimization may lead to convergence issues, especially if the learning rate is not set properly. Poor convergence can result in suboptimal solutions and longer training times.

Interpretability: While the Elastic Net model can simplify feature selection, the inclusion of both L1 and L2 terms can complicate the interpretation of the final model coefficients, especially in cases with many correlated predictors.