CS 210 TERM PROJECT

Yunus Emre Çay – 31260

INTRODUCTION

This Project aims to investigate whether a constant and regular diet can help us lose weight. The motivation for this exploration stems from the widespread interest in identifying effective strategies for weight management, particularly through dietary modifications. As obesity and related health problems continue to pose significant challenges worldwide, the need to distinguish the subtle dynamics between calorie consumption and weight fluctuations is becoming increasingly imperative.

This report presents the outcome of the investigation of the hypothesis which states that a reduced calorie diet can lead to measurable weight loss.

PROBLEM

Is there a correlation between calorie intake and weight changes?

METHODOLOGY

EDA, visualization, and several machine learning techniques were used during this project.

DATA

The data includes my calorie intake, protein intake, water intake and corresponding weights for several months.

1. EXPLORING THE DATA, VISUALIZATION

Here is a brief introduction to my data:

```
[50]  1 import pandas as pd
      2
      3 excel_file_path = 'diet.xlsx'
      4
      5 df = pd.read_excel(excel_file_path)
      6
      7 print(df)
```

```
          DATE  CALORIE (KCAL)  PROTEIN (GR)  WATER (ML)  WEIGHT (KG)
0   2023-10-12            1592         130.0        2000           97
1   2023-10-13            1298          98.0        3500           97
2   2023-10-14            1526         112.0        3000           97
3   2023-10-15            1441         107.0        2750           97
4   2023-10-16            1069          74.0        3500           97
..         ...             ...           ...         ...          ...
86  2024-01-06            1335         117.0        1750           92
87  2024-01-07            1102          96.0        2250           92
88  2024-01-08            1485         124.0        2000           92
89  2024-01-09            1320          99.0        3000           92
90  2024-01-10            1034          95.0        2500           92

[91 rows x 5 columns]
```

```
[51]  1 dataset_shape = df.shape
      2 dataset_shape
```

```
(91, 5)
```

```
[52]  1 dataset_summary = df.info()
      2 dataset_summary
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91 entries, 0 to 90
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   DATE            91 non-null     datetime64[ns]
 1   CALORIE (KCAL)  91 non-null     int64
 2   PROTEIN (GR)    91 non-null     float64
 3   WATER (ML)      91 non-null     int64
 4   WEIGHT (KG)     91 non-null     int64
dtypes: datetime64[ns](1), float64(1), int64(3)
memory usage: 3.7 KB
```

```python
numeric_summary = df.describe()
print("Summary Statistics of Numeric Columns:")
print(numeric_summary)

```

```
Summary Statistics of Numeric Columns:
       CALORIE (KCAL)  PROTEIN (GR)   WATER (ML)  WEIGHT (KG)
count       91.000000     91.000000    91.000000    91.000000
mean      1366.186813    117.963516  2795.604396    94.967033
std        213.230752     25.125659   641.424125     1.642833
min        917.000000     74.000000  1500.000000    92.000000
25%       1278.000000    101.500000  2250.000000    93.500000
50%       1358.000000    114.000000  2750.000000    95.000000
75%       1487.000000    131.200000  3225.000000    96.000000
max       1854.000000    202.000000  4500.000000    97.000000
```

I created a copy of the dataset because I wanted to overlook the outliers to use machine learning models better:

```python
import pandas as pd

# Copy the DataFrame
df_copy = df.copy()


```

```python
# Remove outliers based on specified ranges

# Calorie range: 1200-1700
calorie_mask = (df_copy['CALORIE (KCAL)'] >= 1200) & (df_copy['CALORIE (KCAL)'] <= 1700)

# Water range: 2500-4000
water_mask = (df_copy['WATER (ML)'] >= 2500) & (df_copy['WATER (ML)'] <= 4000)

# Protein range: 100-170
protein_mask = (df_copy['PROTEIN (GR)'] >= 100) & (df_copy['PROTEIN (GR)' ] <= 170)

# Apply the masks to filter the DataFrame
df_copy = df_copy[calorie_mask & water_mask & protein_mask]

```

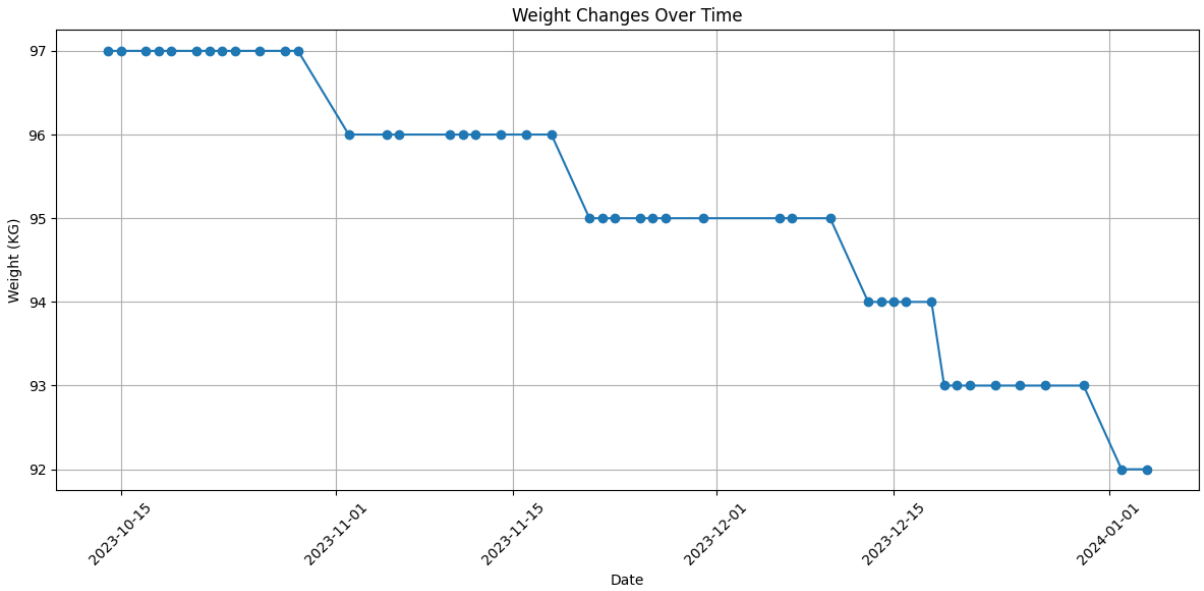
[63]
```
numeric_summary = df_copy.describe()
print("Summary Statistics of Numeric Columns:")
print(numeric_summary)
```

```
Summary Statistics of Numeric Columns:
       CALORIE (KCAL)  PROTEIN (GR)   WATER (ML)  WEIGHT (KG)  DATE_NUMERIC
count       45.000000     45.000000    45.000000    45.000000  4.500000e+01
mean      1407.200000    120.255556  3015.555556    95.177778  1.700619e+18
std        100.749148     13.724170   423.838282     1.556349  2.175405e+15
min       1220.000000    100.000000  2500.000000    92.000000  1.697242e+18
25%       1321.000000    107.000000  2750.000000    94.000000  1.698538e+18
50%       1381.000000    120.000000  3000.000000    95.000000  1.700611e+18
75%       1474.000000    132.000000  3250.000000    97.000000  1.702598e+18
max       1634.000000    149.210000  4000.000000    97.000000  1.704326e+18
```

```
calorie_mean = df_copy['CALORIE (KCAL)'].mean()

print(f"The mean of the 'CALORIE' column is: {calorie_mean}")

water_mean = df_copy['WATER (ML)'].mean()

print(f"The mean of the 'WATER' column is: {water_mean}")

protein_mean = df_copy['PROTEIN (GR)'].mean()

print(f"The mean of the 'PROTEIN' column is: {protein_mean}")


```

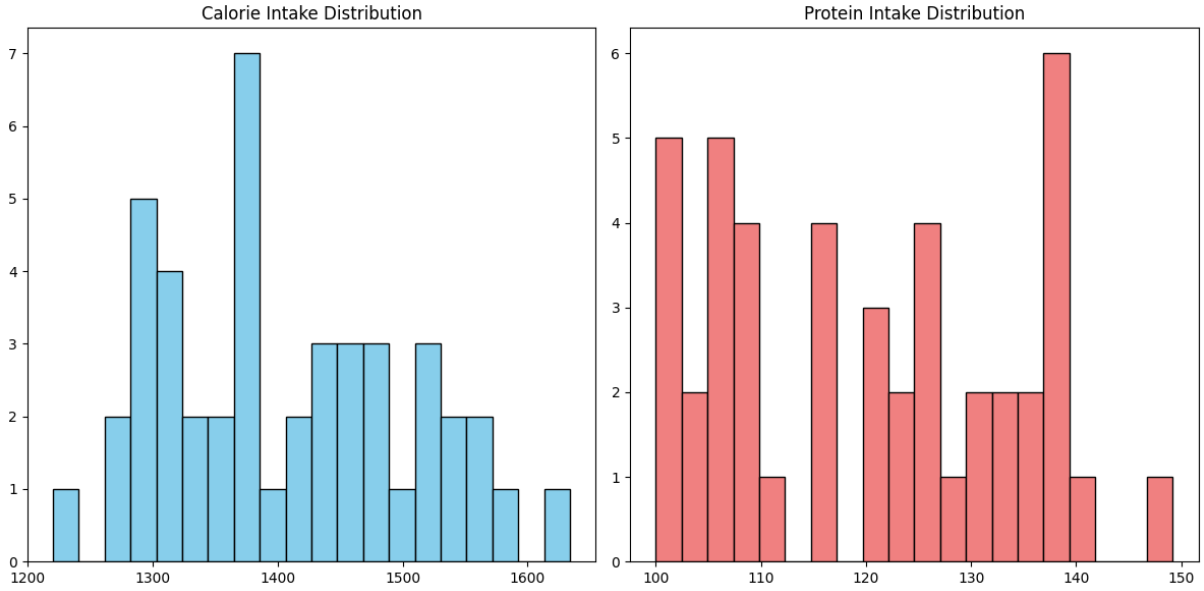
```
The mean of the 'CALORIE' column is: 1407.2
The mean of the 'WATER' column is: 3015.5555555555557
The mean of the 'PROTEIN' column is: 120.25555555555556
```

Visualization of the weight loss I achieved:



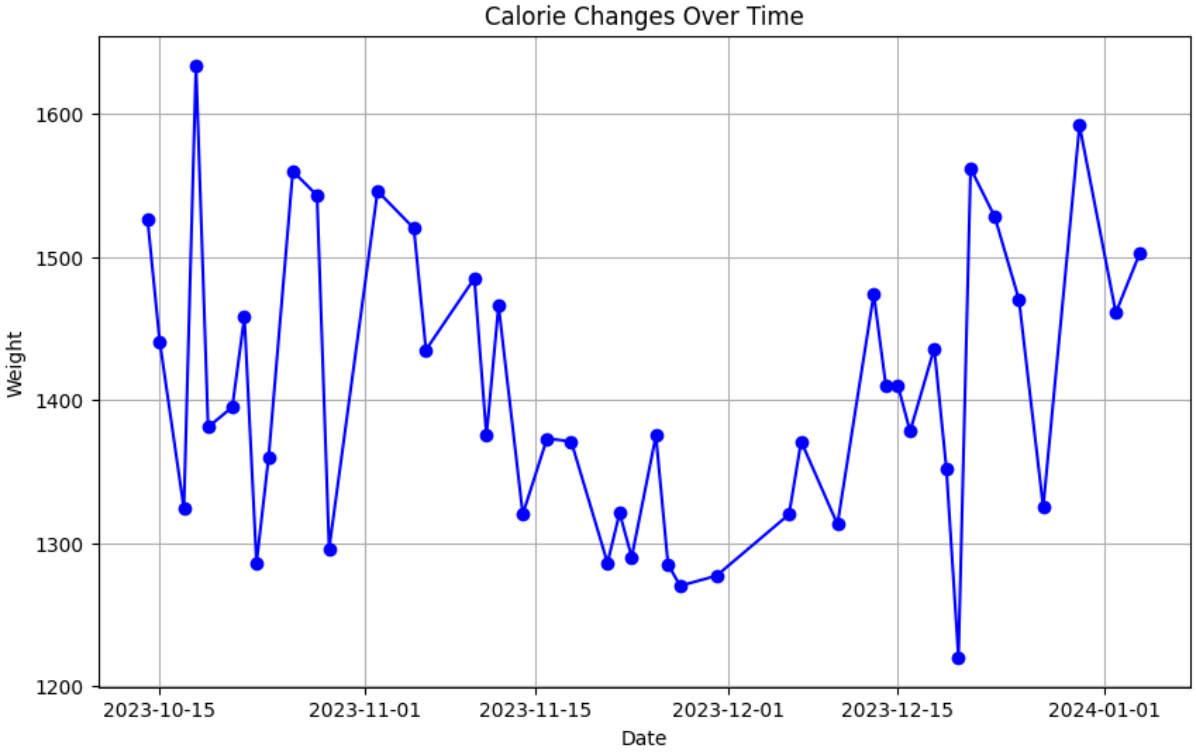
As can be seen, there is a clear weight loss during these months.

My calorie and protein intake as histograms:

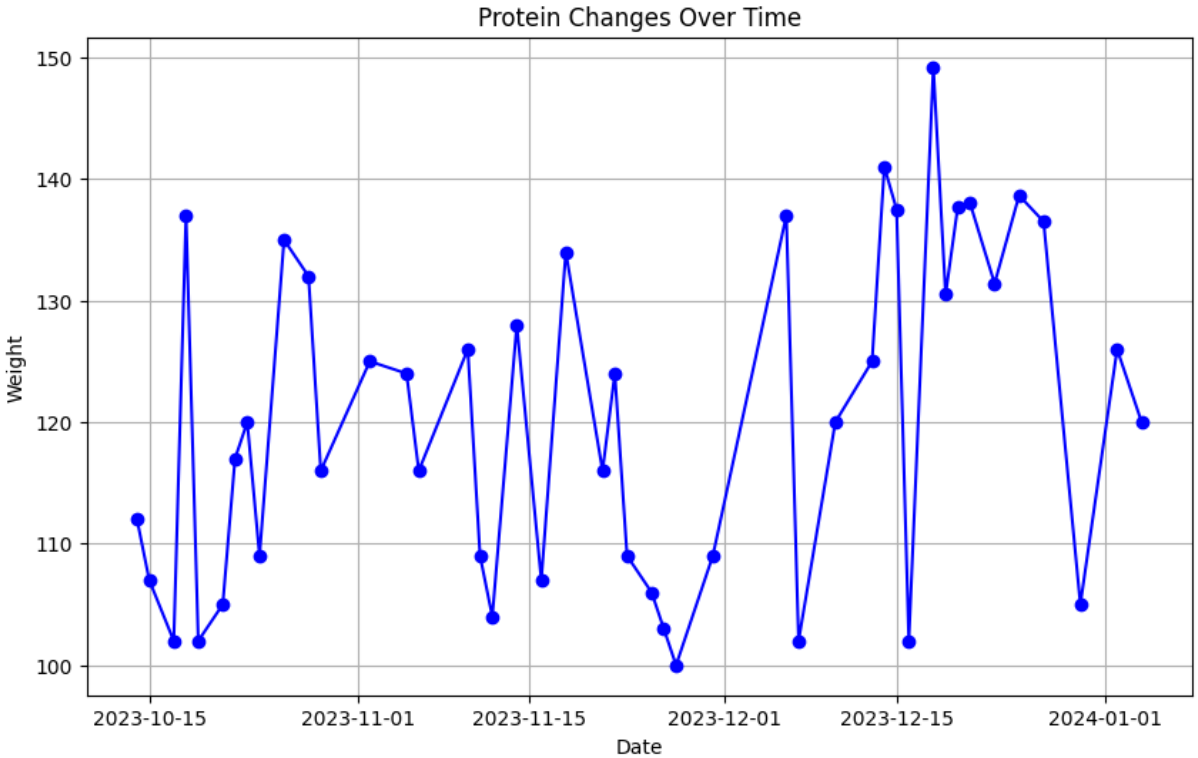


I tried to reduce my calorie intake as much as possible and tried to replace useless calories with proteins.

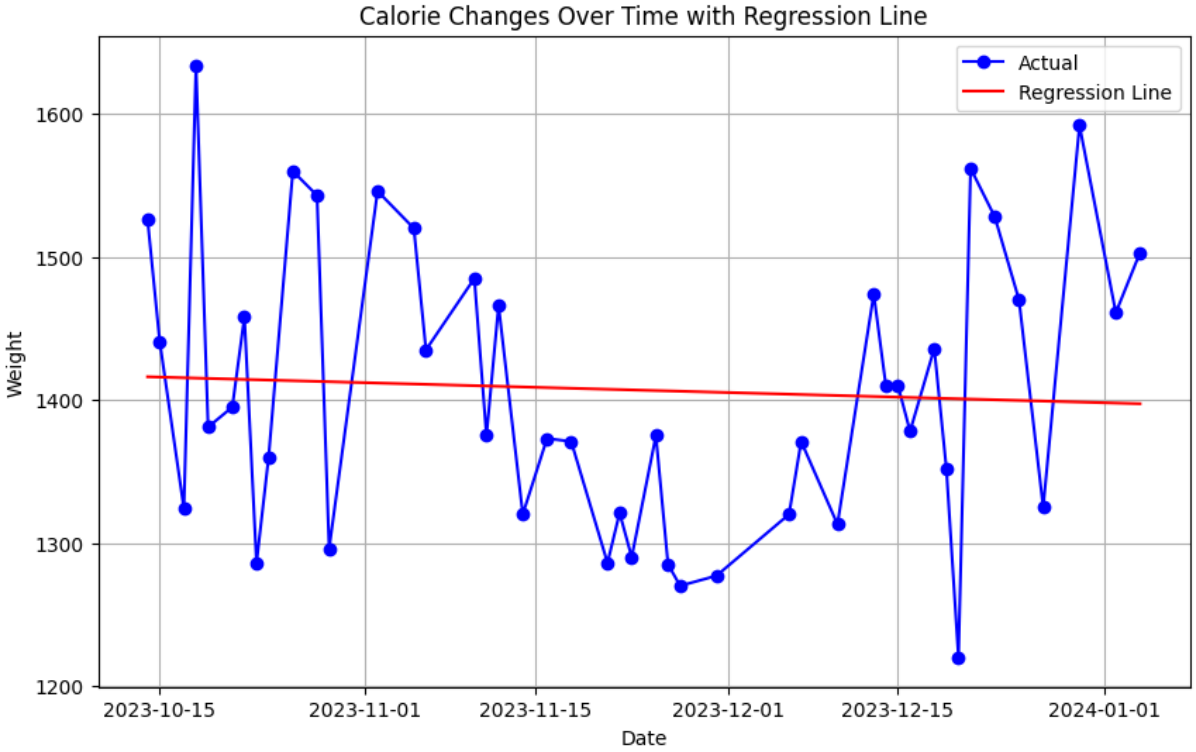
To better see the calorie intake:



To better see the protein intake:

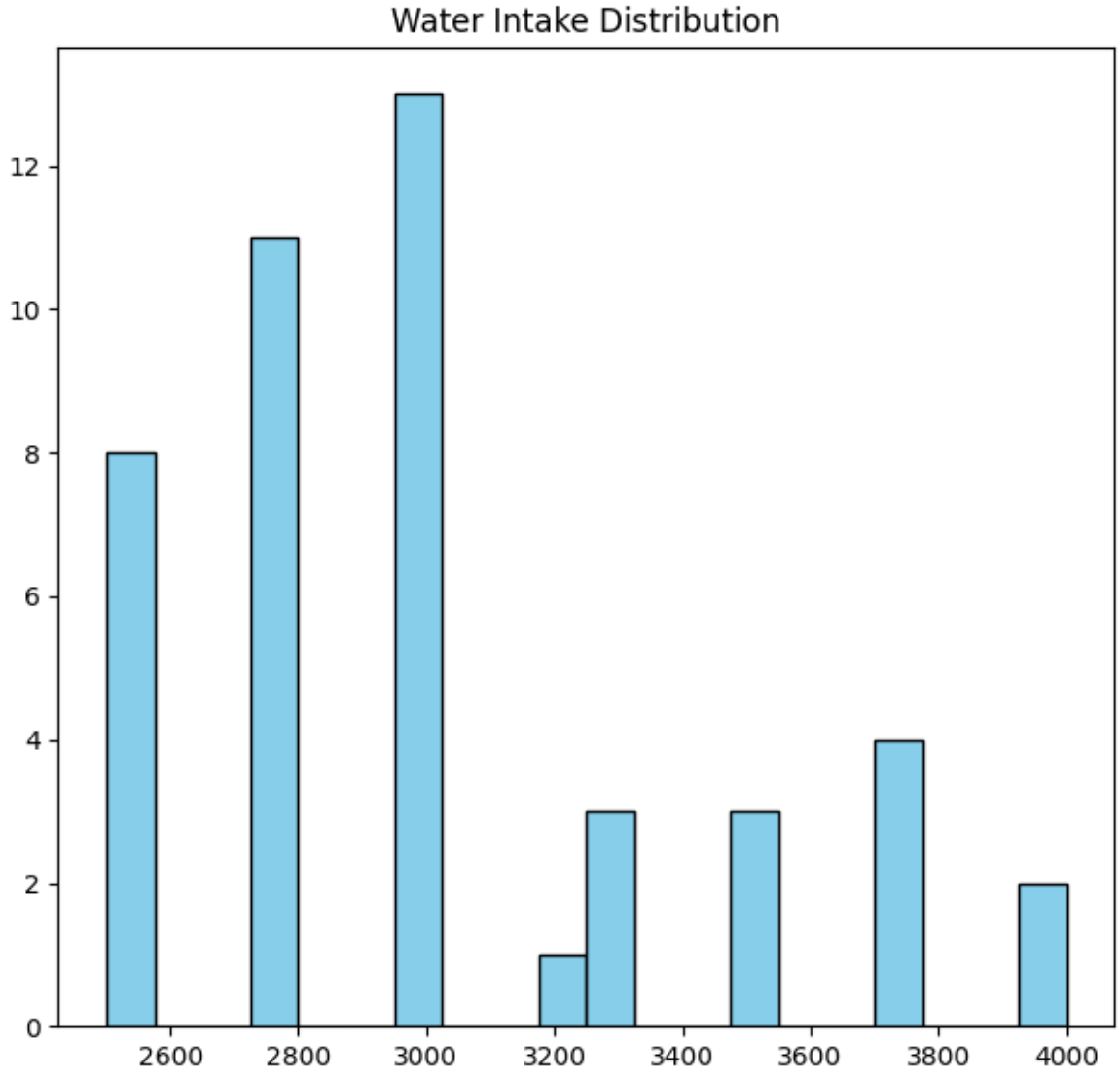


My calorie intake with a regression line:



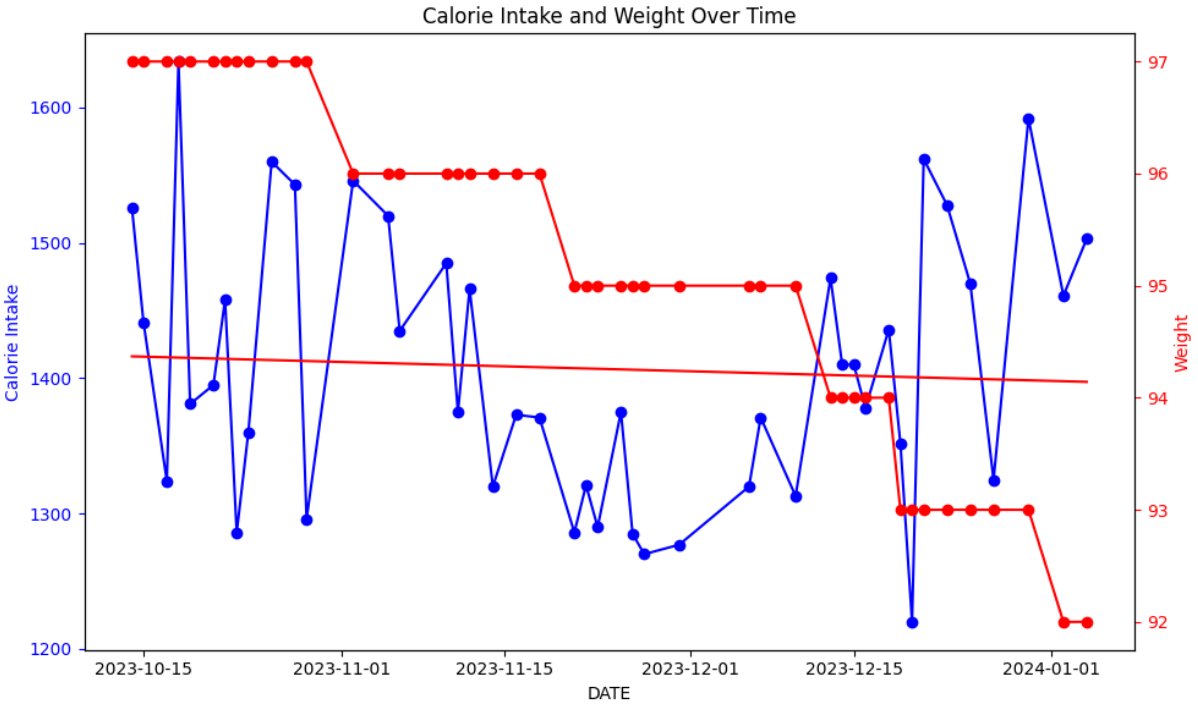
Even though I was trying to do a steady diet, there is still a reduction in calorie intake. Although it can still be ignored considering how little the slope is.

My water intake as histogram:

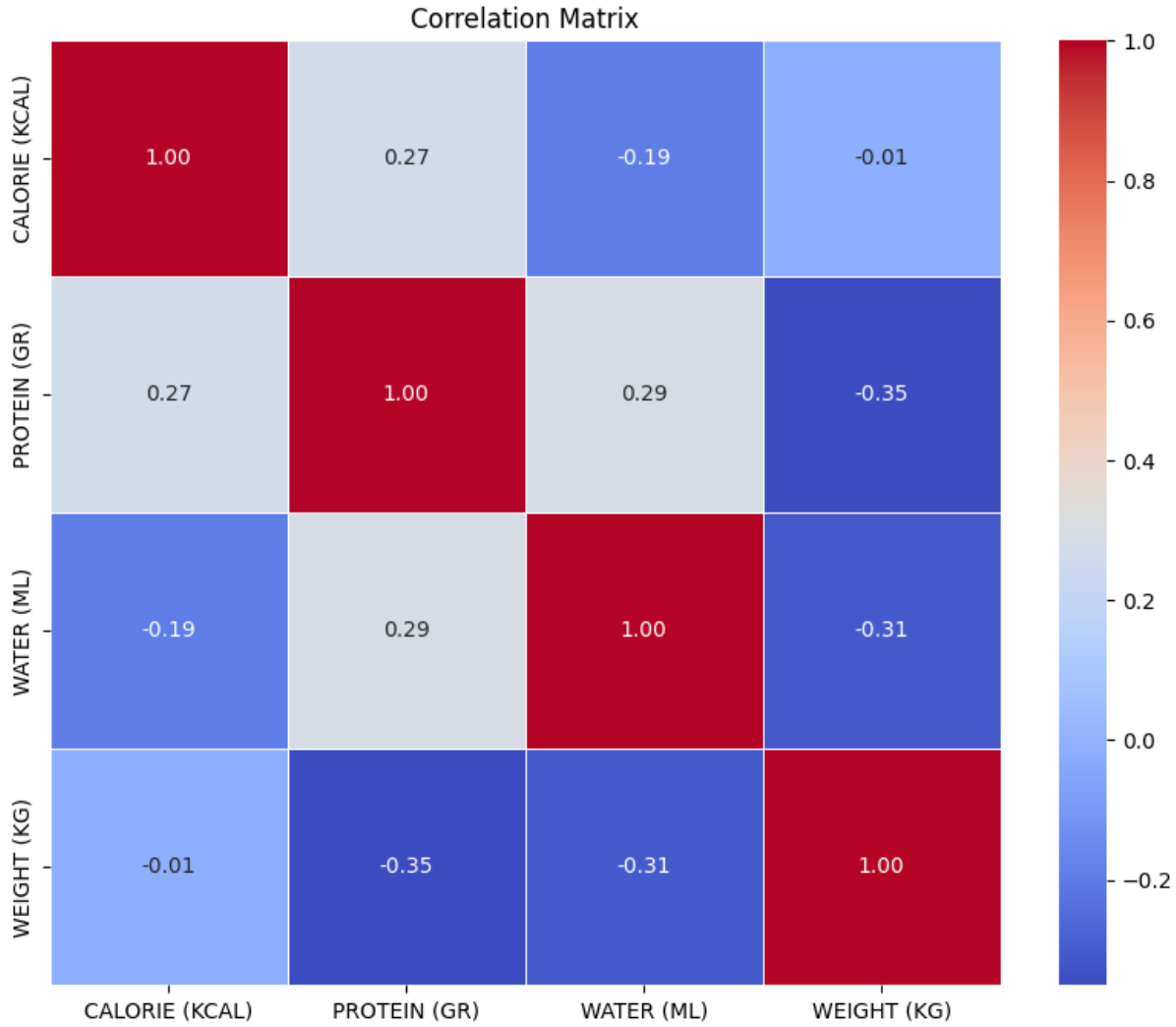


I tried to drink as much water as I can. I couldn’t measure it perfectly since I used my own bottle throughout the diet. So, it looks like I always drank like multiples of 5.

My calorie intake and weight loss combined:



Heat Map



I used heatmap to see the correlations between the variables. As can be seen there is a balance between calories, water, and protein intake.

2. MACHINE LEARNING

Correlation between calorie and weight:

```python
# Extract the correlation between Calorie and Weight
calorie_weight_corr = df_copy['CALORIE (KCAL)'].corr(df_copy['WEIGHT (KG)'])
print(f'Correlation between Calorie and Weight: {calorie_weight_corr}')

# Extract the correlation between Calorie and Weight
calorie_weight_corr = df['CALORIE (KCAL)'].corr(df['WEIGHT (KG)'])
print(f'Correlation between Calorie and Weight: {calorie_weight_corr}')


```

```
Correlation between Calorie and Weight: -0.008203805939660136
Correlation between Calorie and Weight: 0.09346080570475393
```

As can be seen there is little to no correlation between calorie and weight data. What causes is that the calorie data is stable (remember the graph with the regression line) unlike the weight data which always decreases.

Hypothetical features:

```python
# Hypothetical features
df_copy['NutrientDensityScore'] = df_copy['PROTEIN (GR)'] / df_copy['CALORIE (KCAL)']  # Nutrient
df_copy['HydrationIndex'] = df_copy['WATER (ML)'] / df_copy['CALORIE (KCAL)']  # Hydration Index

# Correlation with the target variable
correlation_nutrient_density = df_copy['NutrientDensityScore'].corr(df_copy['HydrationIndex'])

print(f"Correlation between Nutrient Density Score and WEIGHT: {correlation_nutrient_density}")


# bunların ne anlama geldiğini açıkla
```

```
Correlation between Nutrient Density Score and WEIGHT: 0.49201417858026253
```

I chose these two hypothetical features because I thought they would be suitable for my dataset.

Nutrient density score corresponds to the proportion of protein in the overall diet.

Hydration index corresponds to the proportion of water in the overall diet.

TRAIN/TEST SPLIT:

TRAIN / TEST SPLIT

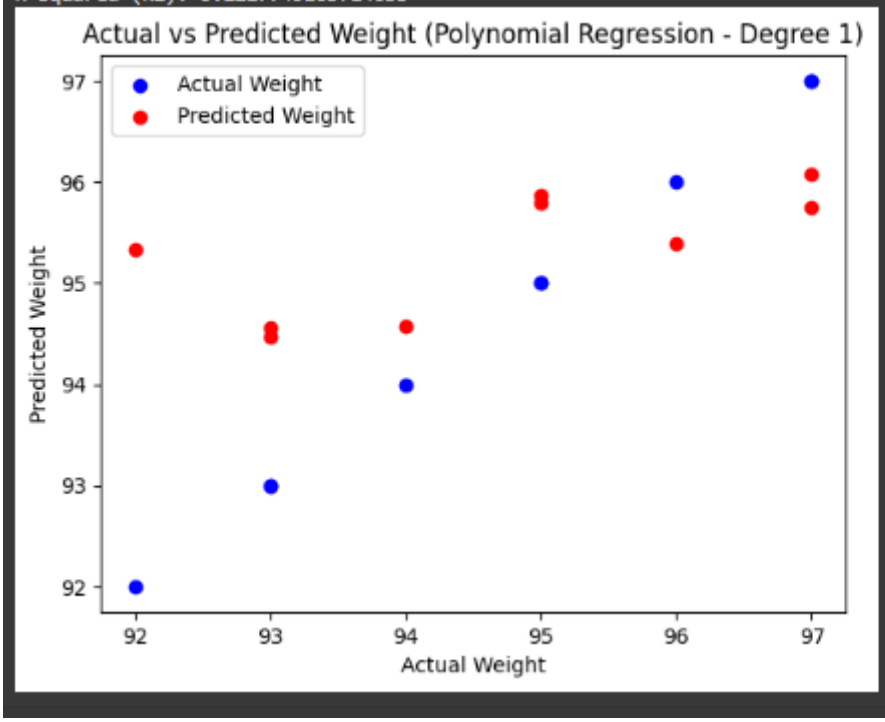
```python
from sklearn.model_selection import train_test_split

# Assuming df_copy is your DataFrame
# Assuming 'WEIGHT' is the target variable, and 'CALORIE', 'PROTEIN', 'WATER' are features

# Define features (X) and target variable (y)
features = df_copy[['CALORIE (KCAL)', 'PROTEIN (GR)', 'WATER (ML)']]
target = df_copy['WEIGHT (KG)']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Display the shapes of the resulting sets
print("Training set shape - Features:", X_train.shape, "Target:", y_train.shape)
print("Testing set shape - Features:", X_test.shape, "Target:", y_test.shape)
```

```
Training set shape - Features: (36, 3) Target: (36,)
Testing set shape - Features: (9, 3) Target: (9,)
```

I split the data with the ratio of 0.2 and 0.8 to use in machine learning.

WEIGHT PREDICTION:

```
Mean Squared Error (MSE): 2.2482058144826125
R-squared (R2): 0.2217749103714033
```



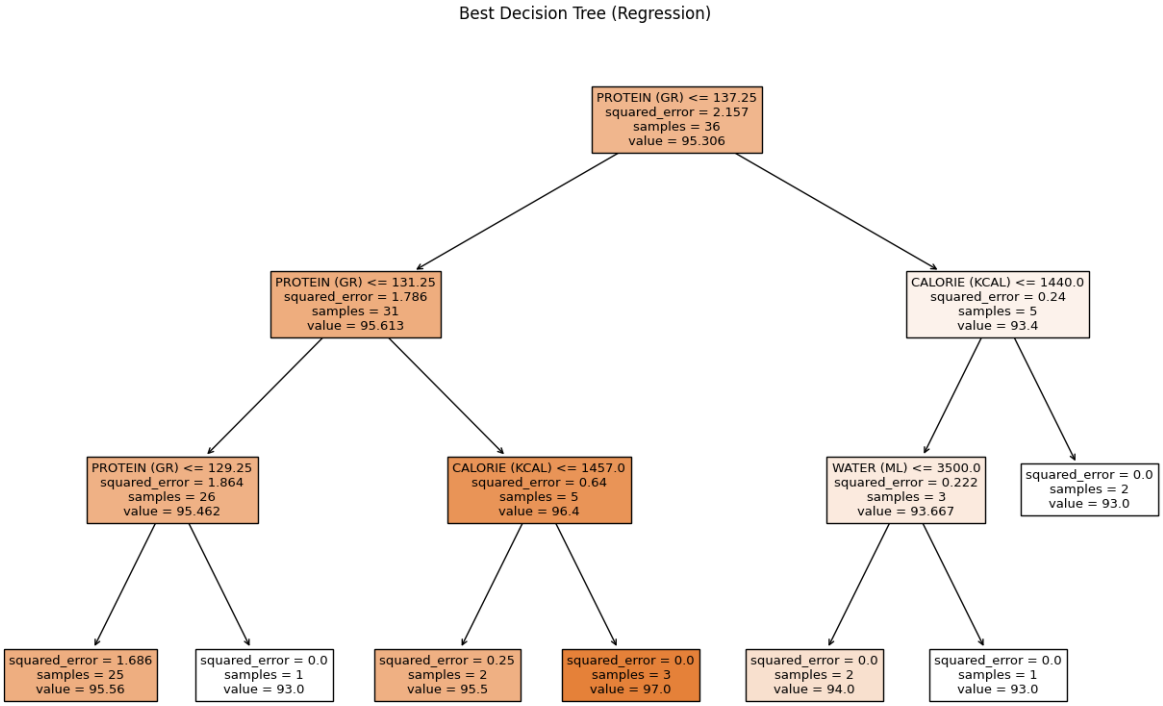
MSE interpretation:

Lower MSE indicates that the machine learning model is making predictions that are, on average, close to the actual weights in the dataset. This indicates a reasonable level of accuracy in weight predictions.

R-SQUARED interpretation:

The R-squared value of 0.22 indicates that approximately 22% of the variability in weight changes can be attributed to the features considered in the model. It also implies that there are likely other influential factors contributing to weight changes that are not accounted for in our current feature set.

DECISION TREE:



With depth 3 and split 2. With accuracy 0.31.

MACHINE LEARNING METHODS:

```
Random Forest Accuracy: -0.2369346153846159
SVM Accuracy: -0.01986974439053646
K-Neighbors Accuracy: -0.16615384615384765
Linear Regression Accuracy: 0.2217749103714024
Neural Network Accuracy: -22.041823533579866
Naive Bayes Accuracy: 0.15384615384615385
Gradient Boosting Accuracy: -0.7630111188600504
```

As can be seen, linear regression and naive bayes worked a bit. However, the other models didn't fit enough to the data. As discussed before, the value of

linear regression 0.22 points out that only 22% of the data can correlate with the model that trained.

P VALUE:

```python
import pandas as pd
from scipy.stats import pearsonr

# Extract relevant columns
calorie_intake = df_copy['CALORIE (KCAL)']
weight_changes = df_copy['WEIGHT (KG)']

# Calculate Pearson correlation coefficient and p-value
correlation_coefficient, p_value = pearsonr(calorie_intake, weight_changes)

print(f"Pearson Correlation Coefficient: {correlation_coefficient}")
print(f"P-value: {p_value}")

```

```
Pearson Correlation Coefficient: -0.008203805939660233
P-value: 0.9573454634008214
```

The null hypothesis assumes there is no correlation between calorie intake and weight changes. The p value of 0.95 indicates that we don't have enough evidence to reject the null hypothesis. A high p value indicates the results are more likely random.

Conclusion:

As can be seen, we failed to reject the null hypothesis. The data does not provide strong support for the existence of a relationship between calories and weight. One needs to remember that the findings don't conclusively demonstrate no correlation; instead, they underscore the limitations of the dataset in establishing a clear relationship between these variables.

Recommendations:

The data could be more organized so it would be easier for us to reject the null hypothesis.