Name: Nguyen Quy Toan

26/04/2025

Project Spring 2024

Student-mat_.csv dataset

- Data description: how many samples? How many features? What type of features?
    - There are 395 samples in this dataset
    - There are 31 features (Example: School, Sex, Age, …etc.)
    - The features include both categorical (object) and numerical (int64, float64) data types, where categorical features represent qualitative data (school, sex, address, …etc.), and numerical features represent quantitative data (age, Medu, failures,...etc.).

```
Number of samples: 395

Number of features: 31

Feature types:
school        object
sex           object
```

- Data preprocessing: are there any null values or outliers? How did you deal with them? How did you handle scaling?
    - Yes, there are many null and outlier. Some columns (like age, studytime, failures, absences) have crazy extreme values (ex: age goes up to 90, absences up to 900, studytime 999!).
    - Solution:

      + Null: using "df = df.dropna()" After dropping, all null values were removed.

```
# Drop rows with NaN
df = df.dropna()
```

      + Outliers: I use IQR or Z-score, and after comparing the two, I will choose the Z-score method because it retains 374 rows out of 380, compared to only 60 rows out of 380 with the IQR method (which can go up to a maximum of 94, regardless of how many ranges I add to it). You can also look at the Histograms.

    - For handle scaling I use this code: Using StandardScaler(), which standardizes the features by removing the mean and scaling to unit variance. This ensures all features have the same scale, helping models like linear regression perform better. The scaling is applied to both the training and test data for consistency.

```
# 🎯 Feature Scaling
X = df_no_outliers_z.drop('G3', axis=1)    # Features
y = df_no_outliers_z['G3']                 # Target

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

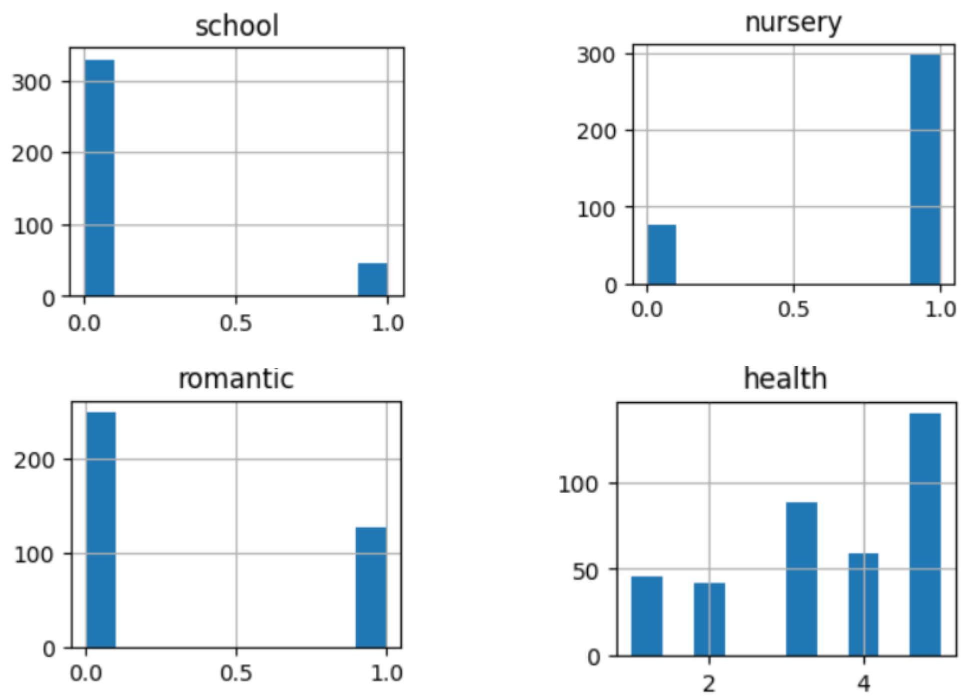
- Exploratory data analysis: visualize the data with graphs and describe your findings. Did you find any patterns?
    - Using the Histograms of Features:

      + I found that most of the students in the dataset come from Gabriel Pereira (GP) compared to Mousinho da Silveira (MS). (0 mean GP |1 mean MS)

      + I also found that most of the student here attended nursery school. (0 mean No|1 mean Yes)

      + ~33% of student in a romantic relationship (Very interesting)

      + >20% had a very bad/bad current health status (Very interesting because they attended nursery school) (1-Very bad, 2-Bad. 3-Normal)



- Using Correlation Map:

  Positive:

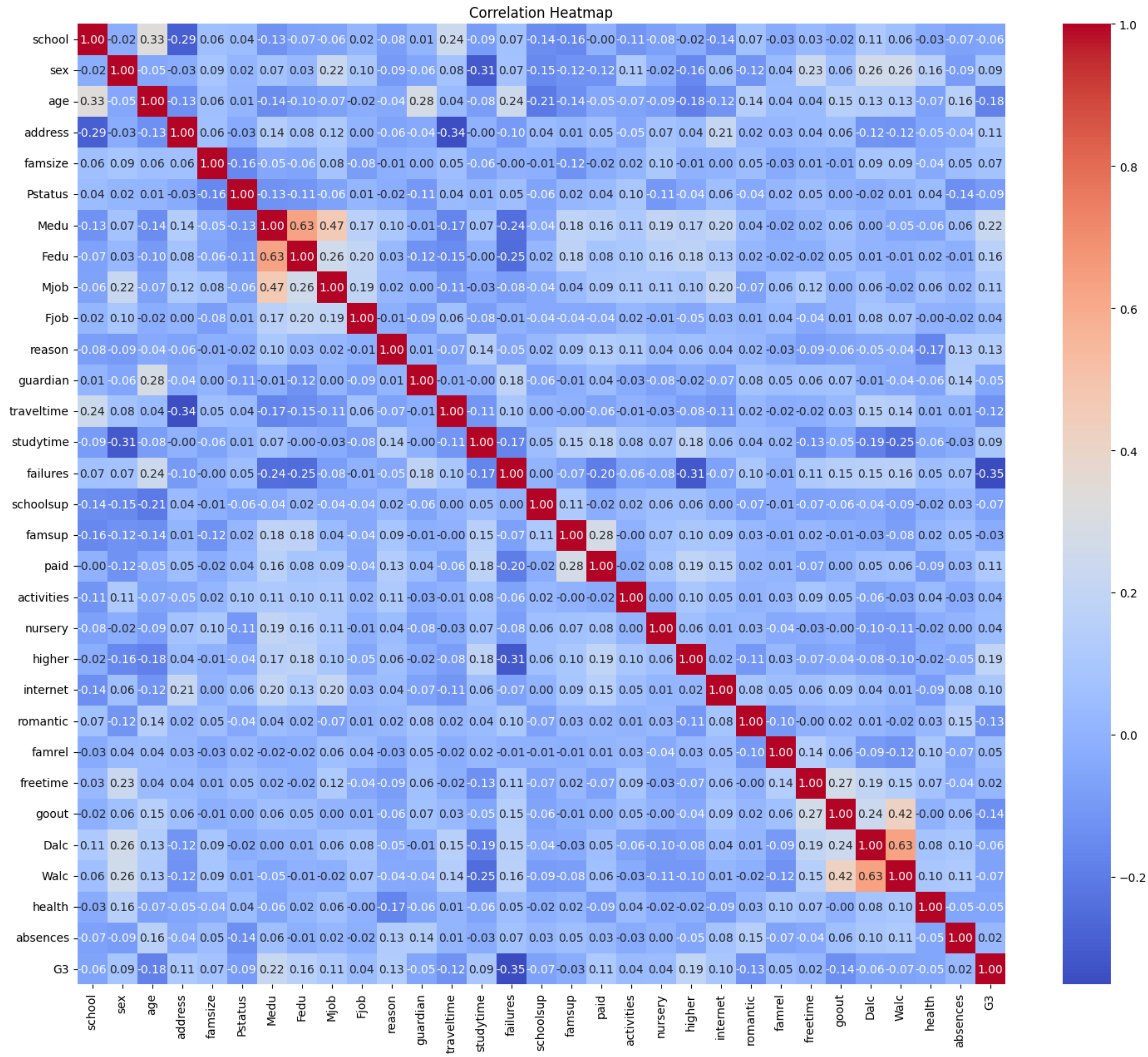
  + ‘Medu’ and ‘Fedu’ (mother's and father's education) are strongly correlated. (0.63) ➔ parent education

  _ ‘Medu’ and ‘Mjob’ ➔ mother education is relative with mother job

  Negative:

  + ‘Failures’ and ‘G3’ have a negative correlation of about -0.35 ➔ More past failures → lower final grades.

  + ‘address’ and ‘traveltime’ have a negative correlation of about -0.34



- Model development: state the hyperparameters selected for the models and how/why you selected those hyperparameters

- Linear Regression:
  + Hyperparameters: No specific hyperparameters selected (uses defaults).
  + Reasoning: Linear Regression does not require tuning and fits the data with default settings.
- Ridge Regression:
  + Hyperparameters: alpha=1.0 (regularization strength).
  + Reasoning: The default alpha value is chosen to balance regularization and model fitting, preventing overfitting without affecting performance too much.
- Decision Tree:
  + Hyperparameters: max_depth=5 (limits tree depth). random_state=42 (ensures reproducibility).
  + Reasoning: A depth of 5 prevents overfitting by limiting the tree's complexity, and setting the random state ensures consistent results.

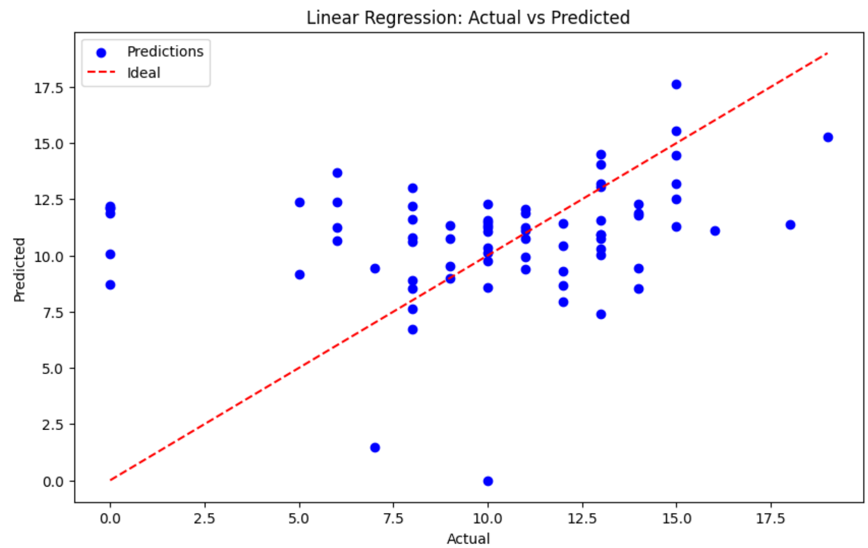
- Performance evaluation: state the model results - accuracy, loss, precision, recall, f1-score, confusion matrix, etc.

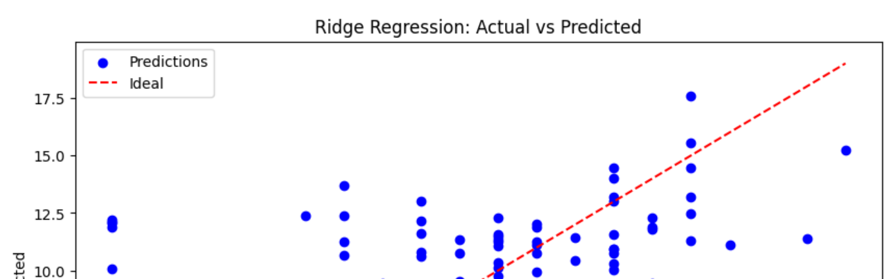
Linear Regression:

RMSE: 4.48

MAE: 3.20

R^2 Score: -0.15





Ridge Regression

RMSE: 4.47

MAE: 3.20

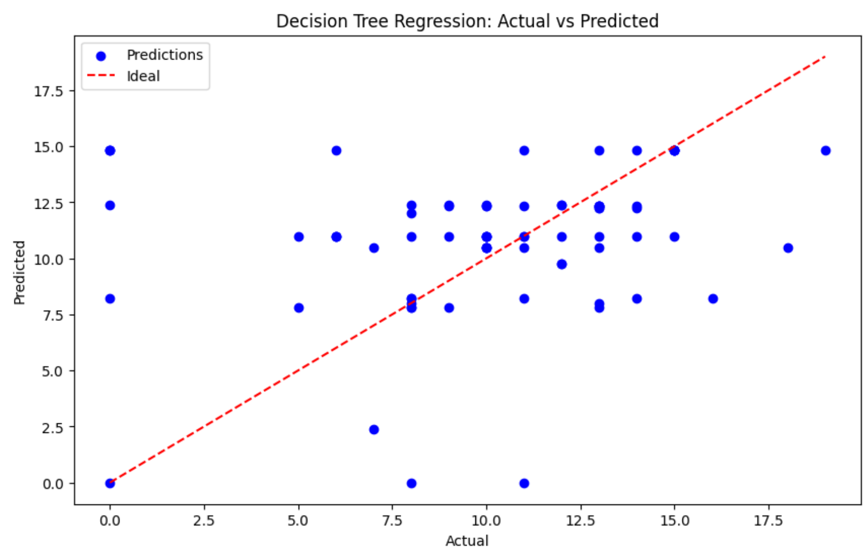
R^2 Score: -0.15

Decision Tree Regression

RMSE: 4.75

MAE: 3.10

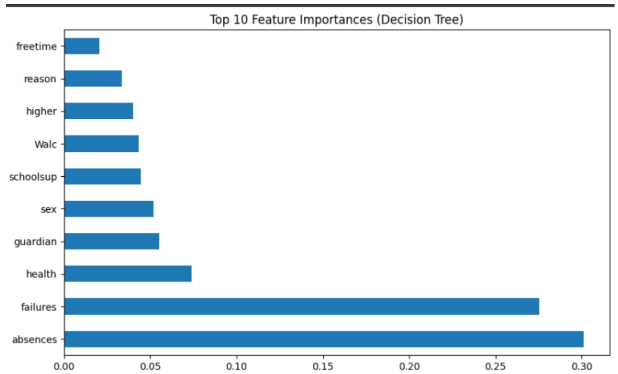
R^2 Score: -0.29



- Interpretation: state why a model is performing better/worse. What are the most significant features? Why is the model classifying or clustering to a specific cluster? How do the model results relate to the dataset and the problem?
  - The Linear Regression and Ridge Regression models perform similarly, with Ridge slightly reducing overfitting but not improving R^2 much. The Decision Tree model performs worse, maybe because due to overfitting on the training data despite setting a max_depth = 5 (we can change it to find the better setting)

  - The most significant features, based on the Decision Tree's feature importance, are factors like study time, failures, and absences, which directly impact students' final grades (G3).

- The models struggle because the dataset has many small, correlated factors influencing grades, making it harder to predict with high accuracy. Also, noise and hidden factors (like personal motivation) that aren't captured in the data may affect the model's ability to predict G3 precisely.



obesity_prediction.csv

```
Number of samples: 2112
Number of features: 16

Data Types:
 Gender          object
Age             float64
Height          float64
```

- Data description: how many samples? How many features? What type of features?
    - There are 2112 samples in this dataset
    - There are 16 features (Example: Age, Weight, Height …etc.)
    - Both numerical (e.g., Age, Height, Weight, etc.) and categorical (e.g., Gender, family_history_with_overweight, Obesity, etc.) features are present.
    - Numerical features are continuous values.
    - Categorical features are strings or categories that were later label encoded.

- Data preprocessing: are there any null values or outliers? How did you deal with them? How did you handle scaling?
    - Yes, there are some null and outlier. Some columns (like Weight, FCVC, CH2O) have crazy extreme values (ex: Weight goes up to 9999, FCVC up to 5000, CH2O -60.)

```
Original Data Shape: (2112, 17)

Null values in the dataset:
Gender            0
Age               0
Height            0
Weight            0
family_history    1
FAVC              1
FCVC              0
NCP               0
CAEC              1
SMOKE             1
CH2O              1
SCC               0
FAF               0
TUE               0
CALC              0
MTRANS            1
Obesity           0
dtype: int64

Shape after dropping nulls: (2106, 17)
```

```
Statistical Summary:
               Age       Height       Weight          FCVC          NCP  \
count  2112.000000  2112.000000  2112.000000   2112.000000  2112.000000
mean     24.311255     1.701674    91.290404     26.092140     2.685610
std       6.344766     0.093283   217.273701   1087.932843     0.777855
min      14.000000     1.450000    39.000000      1.000000     1.000000
25%      19.947666     1.630000    65.815202      2.000000     2.658599
50%      22.774751     1.700357    83.000000      2.386464     3.000000
75%      26.000000     1.768450   107.501904      3.000000     3.000000
max      61.000000     1.980000  9999.000000  50000.000000     4.000000

              CH2O          FAF          TUE
count  2111.000000  2112.000000  2112.000000
mean      1.978641     1.011063     0.658198
std       1.482269     0.851119     0.608974
min     -60.000000     0.000000     0.000000
25%       1.575789     0.125965     0.000000
50%       2.000000     1.000000     0.625360
75%       2.477420     1.667464     1.000000
max       3.000000     3.000000     2.000000
```

- Solution:

  + Null: using “df = df.dropna()” After dropping, all null values were removed.

  ```
  # Drop rows with NaN
  df = df.dropna()
  ```

  + Outliers: I use Z-score, and after comparing the two, I will choose the Z-score method because it retains 2079 rows out of 2106, You can also look at the Histograms or Statistical Summary to find the outliers.

- For handle scaling I use this code:

  I used StandardScaler to standardize the feature values (mean = 0, std = 1) before training models.

  ```
  # Handle categorical variables
  label_encoder = LabelEncoder()
  for col in df_obesity.columns:
      if df_obesity[col].dtype == 'object':
          df_obesity[col] = label_encoder.fit_transform(df_obesity[col])

  # Separate features and target
  X_obesity = df_obesity.drop('Obesity', axis=1)  # Features
  y_obesity = df_obesity['Obesity']  # Target

  # Feature Scaling
  scaler_obesity = StandardScaler()
  X_scaled_obesity = scaler_obesity.fit_transform(X_obesity)
  ```

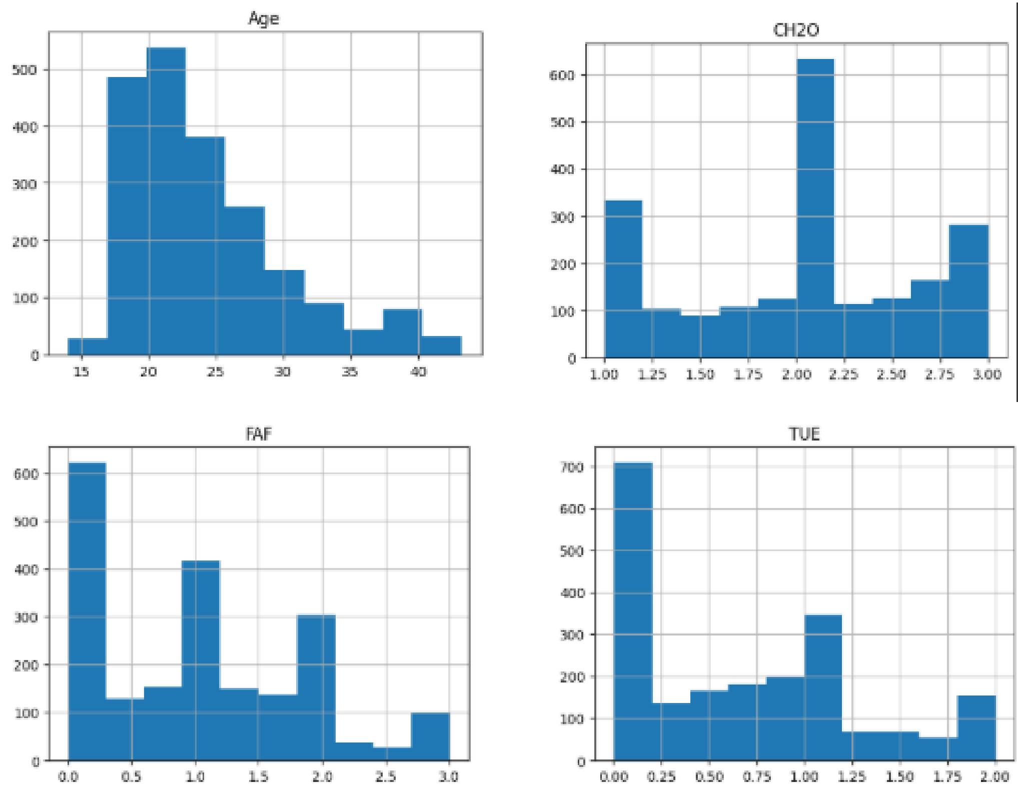
- Exploratory data analysis: visualize the data with graphs and describe your findings. Did you find any patterns?
  - Exploratory data analysis: visualize the data with graphs and describe your findings. Did you find any patterns?
    - Using the Histograms of Features:

      + I found that most of the people researching are at the age of 17-27 (It is a very young age when the body is in its healthiest state.)

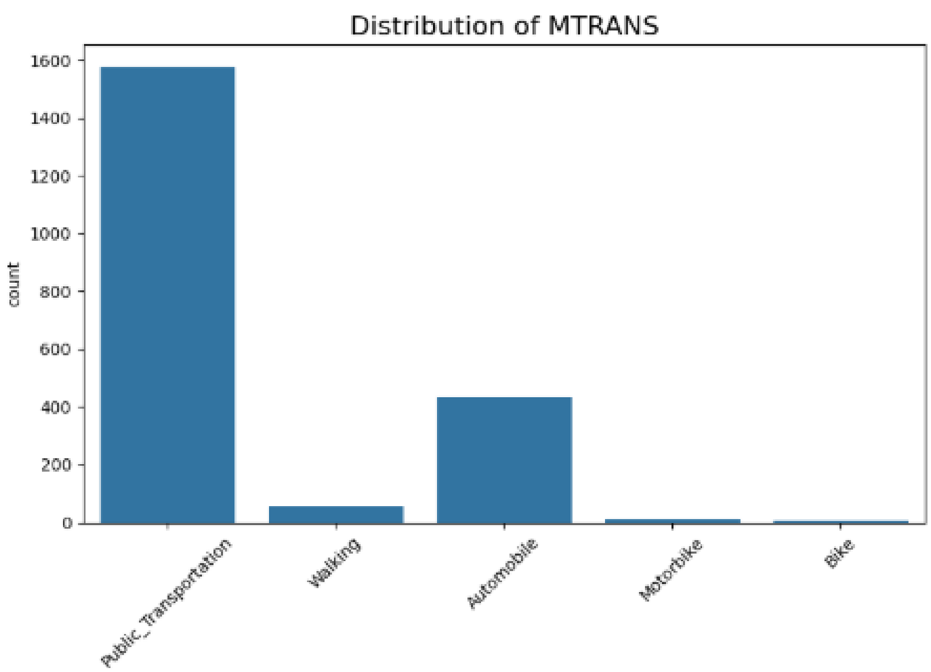
      + I also found that people drinking ok amount of water, 2.0-2.24 liters of water, consider According to WHO Men should aim for about 2.9 liters and 2.2 for, but there still a lot of people drinking 1-1.25 liters/day, which is need to be reconsidered. (Water help you fulfill your stomach so you not eat too much food and it also helps fat breakdown)

      + Look at the FAF (How often do you have physical activity) and Tue (How much time do you use technological devices) I am surprise that people don’t or rarely do activity and they not using any technological devices (cell phone, games…) this is quite opposite.

      + People mostly use public transport (This is very rare)


Age
CH2O
FAF
TUE
0 100 200 300 400 500
15 20 25 30 35 40
0 100 200 300 400 500 600
1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00
0 100 200 300 400 500 600
0.0 0.5 1.0 1.5 2.0 2.5 3.0
0 100 200 300 400 500 600 700
0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00

CALC


Distribution of MTRANS
count
0 200 400 600 800 1000 1200 1400 1600
Public_Transportation
Walking
Automobile
Motorbike
Bike

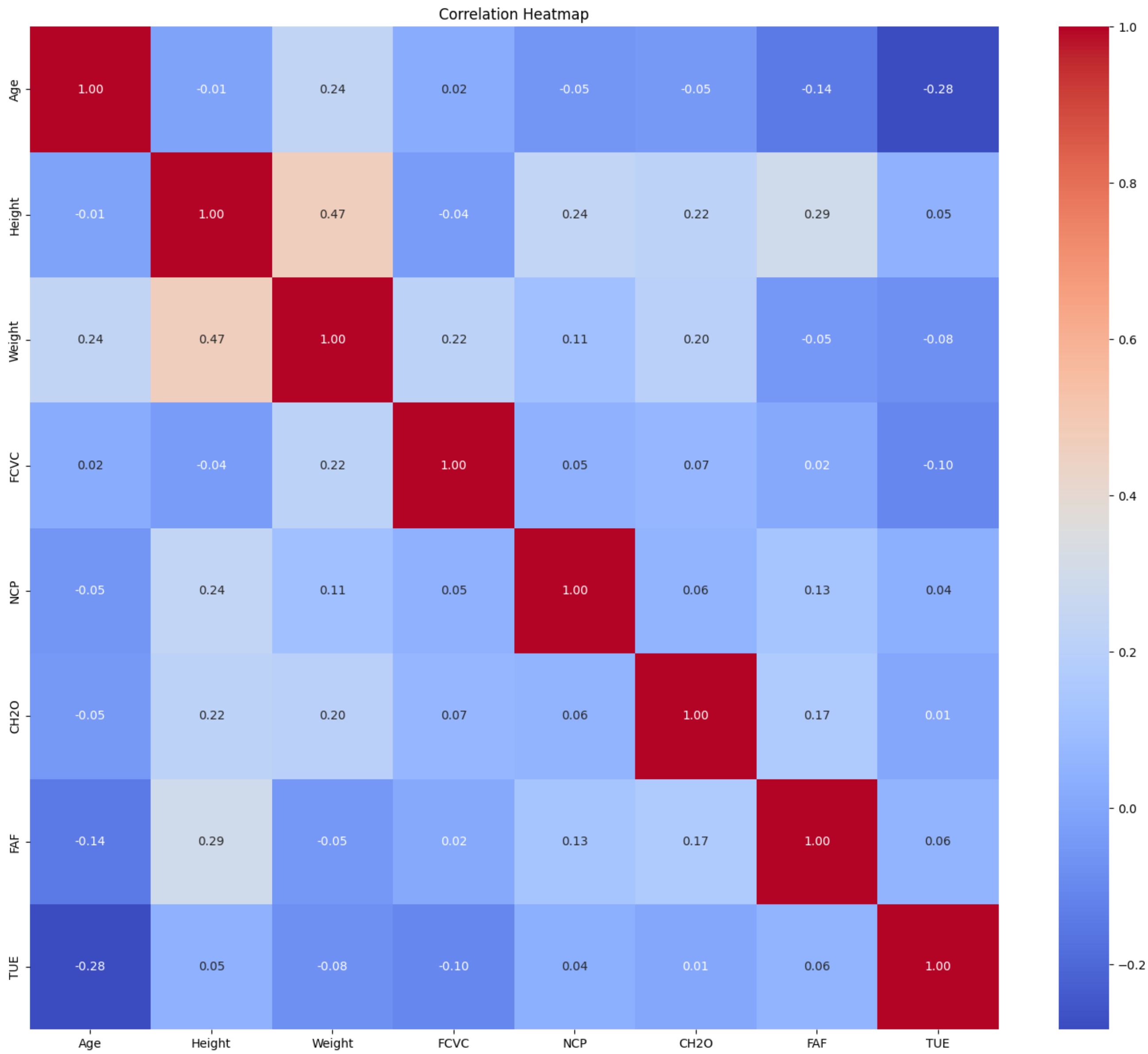
- Using Correlation Map:

  Positive:

  + ‘Weight’ and ‘Height’ has a strong relation which is 0.47 (This is made sense because you need more energy and also bones weight is adding up too)

  Negative:

  + ‘Age’ and ‘Tue’ have negative correlation -0.28 (It mean age is not affecting How much time do they use technological) also which FAF -0.14 (How often do they have physical activity)



- Model development: state the hyperparameters selected for the models and how/why you selected those hyperparameters
  - Logistic Regression:

    + Hyperparameters: max_iter=1000, random_state=42

    + max_iter=1000 is selected to ensure convergence (some logistic regression models may not converge if the iteration limit is too low).

    + random_state=42 for reproducibility.

```
# Logistic Regression Model
logreg_obesity = LogisticRegression(max_iter=1000, random_state=42)
logreg_obesity.fit(X_train_obesity, y_train_obesity)
y_pred_logreg = logreg_obesity.predict(X_test_obesity)
```

- K-Nearest Neighbors (KNN)
  - + Hyperparameters: n_neighbors=5
  - + k=5 is a common default choice, balancing bias and variance. It can be change later for optimization.

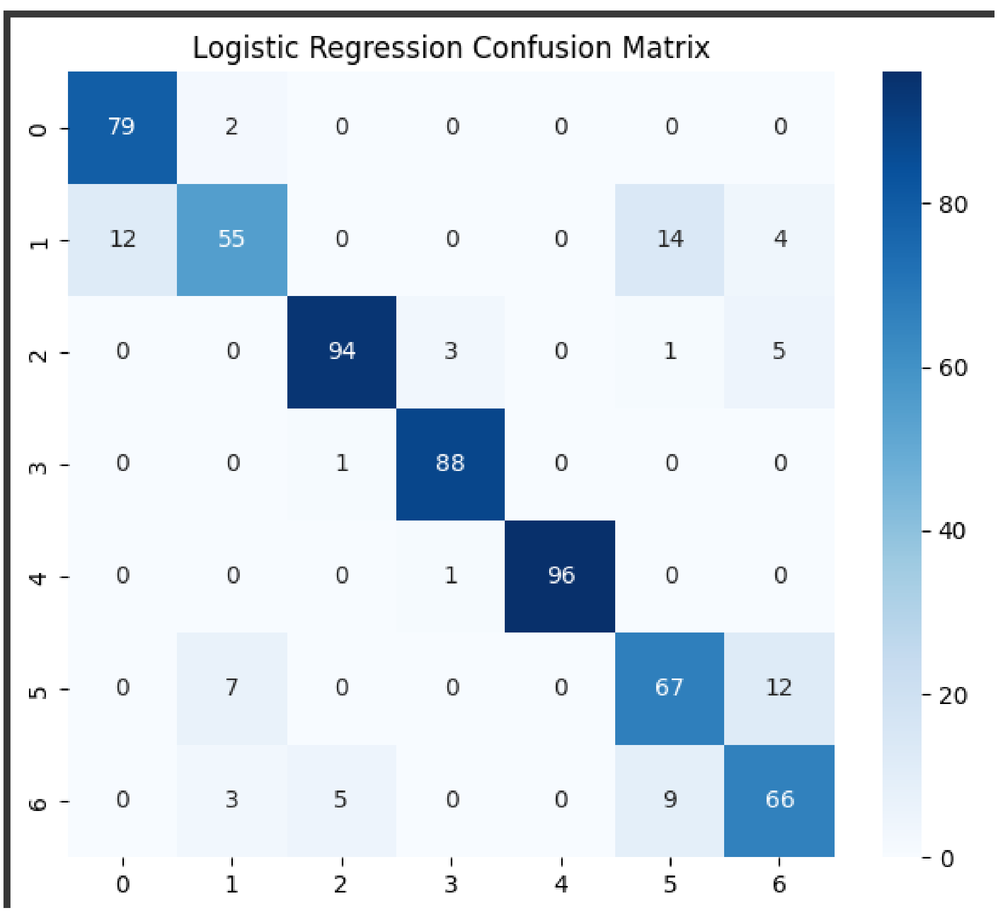
```python
# K-Nearest Neighbors Model
knn_obesity = KNeighborsClassifier(n_neighbors=5)  # k=5
knn_obesity.fit(X_train_obesity, y_train_obesity)
y_pred_knn = knn_obesity.predict(X_test_obesity)
```

- Performance evaluation: state the model results - accuracy, loss, precision, recall, f1-score, confusion matrix, etc.
  - Logistic Regression:
    - + Accuracy: 0.873
    - + Precision: 0.87
    - + Recall: 0.87
    - + F1-Score: 0.87

```
--- Logistic Regression Results ---
Accuracy: 0.8733974358974359
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.98      0.92        81
           1       0.82      0.65      0.72        85
           2       0.94      0.91      0.93       103
           3       0.96      0.99      0.97        89
           4       1.00      0.99      0.99        97
           5       0.74      0.78      0.76        86
           6       0.76      0.80      0.78        83

    accuracy                           0.87       624
   macro avg       0.87      0.87      0.87       624
weighted avg       0.87      0.87      0.87       624
```

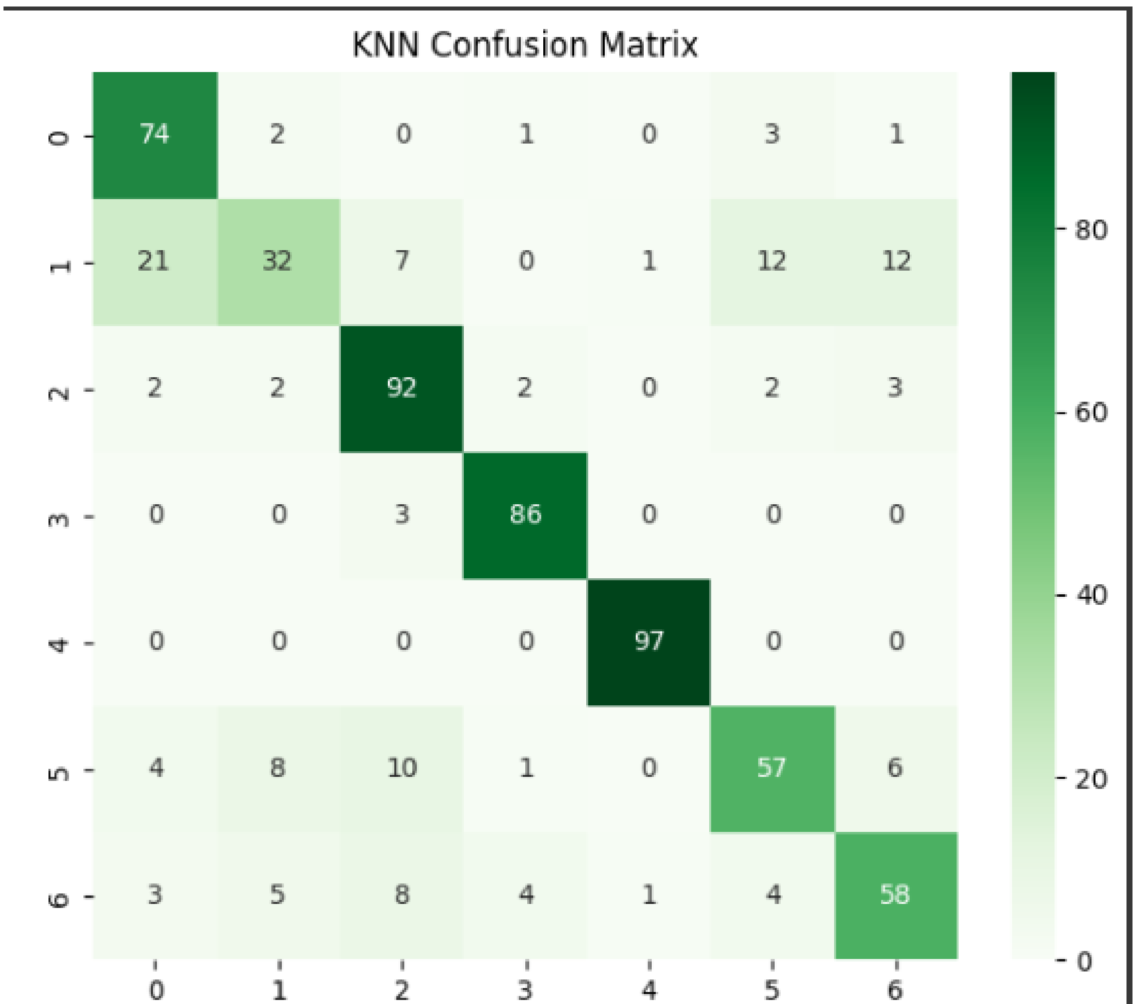


- KNN:
    - + Accuracy: 0.7948
    - + Precision: 0.78-0.79
    - + Recall: 0.79
    - + F1-Score: 0.78

```
--- KNN Results ---
Accuracy: 0.7948717948717948
Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.91      0.80        81
           1       0.65      0.38      0.48        85
           2       0.77      0.89      0.83       103
           3       0.91      0.97      0.94        89
           4       0.98      1.00      0.99        97
           5       0.73      0.66      0.70        86
           6       0.72      0.70      0.71        83

    accuracy                           0.79       624
   macro avg       0.78      0.79      0.78       624
weighted avg       0.79      0.79      0.78       624
```



- Interpretation: state why a model is performing better/worse. What are the most significant features? Why is the model classifying or clustering to a specific cluster? How do the model results relate to the dataset and the problem?
    - Logistic Regression performed better with an accuracy of 87.34%, compared to KNN's 79.49%. Maybe because After scaling the dataset, features became more linearly separable, which Logistic Regression can exploit effectively.
    - Most Significant Features: Weight, Height, and Age were the most important features based on Logistic Regression coefficients. Weight had by far the largest impact.

Feature Importance (Logistic Regression Coefficients):

	Coefficient
Weight	11.586561
Height	2.902029
Age	0.943931
family_history	0.567839
CAEC	0.415472
FAVC	0.256607
FAF	0.171727
Gender	0.150314
MTRANS	0.146824
SMOKE	0.138984
TUE	0.113140
CH2O	0.086430
CALC	0.074425
NCP	0.046570
FCVC	0.026217
SCC	0.011407

    - Relation to Dataset and Problem:
      Since obesity is closely tied to factors like Weight and Age, models that capture these strong patterns (like Logistic Regression) performed better in predicting obesity levels accurately.

Project Summary:

Student-mat_.csv Dataset:

- **Data Description:**
    - 395 samples, 31 features.
    - Features: Categorical (e.g., school, sex) and Numerical (e.g., age, studytime).
- **Data Preprocessing:**
    - Null values were dropped using `df.dropna()`.
    - Outliers were handled using Z-score method, which kept more data.
    - StandardScaler was used for scaling.
- **Exploratory Data Analysis:**
    - Key patterns found:
        - Most students from Gabriel Pereira (GP).
        - ~33% of students in romantic relationships.
        - ~20% had bad health status.
- **Model Development:**
    - **Linear Regression:** RMSE: 4.48, R^2: -0.15.
    - **Ridge Regression:** RMSE: 4.47, R^2: -0.15.
    - **Decision Tree Regression:** RMSE: 4.75, R^2: -0.29.
- **Interpretation:**
    - Linear and Ridge models performed similarly, while Decision Tree overfitted.
    - Significant features: study time, failures, absences.

Obesity_prediction.csv Dataset:

- **Data Description:**
  - 2112 samples, 16 features.
  - Features: Numerical (e.g., weight, height) and Categorical (e.g., gender, family history).
- **Data Preprocessing:**
  - Null values dropped with `df.dropna()`.
  - Z-score used to handle outliers.
  - StandardScaler applied for scaling.
- **Exploratory Data Analysis:**
  - Key patterns found:
    - Most people are aged 17-27.
    - A lot drink less water than recommended.
    - People mostly use public transport.
- **Model Development:**
  - **Logistic Regression:** Accuracy: 87.34%, F1-Score: 0.87.
  - **KNN:** Accuracy: 79.49%, F1-Score: 0.78.
- **Interpretation:**
  - Logistic Regression outperformed KNN, likely due to its better handling of linearly separable data.
  - Significant features: weight, height, age.

Conclusions:

- Both projects highlight the importance of feature preprocessing (handling nulls, scaling, and outliers).
- Logistic Regression performed best in both datasets due to its ability to handle important feature correlations.

- References

Kumbhar, R. (2020). *Obesity prediction* [Dataset]. Kaggle. https://www.kaggle.com/datasets/ruchikakumbhar/obesity-prediction

Cortez, P., & Silva, A. (2008). *Student performance* [Dataset]. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/320/student+performance

World Health Organization. (2020). *Obesity: Preventing and managing the global epidemic* (Report No. 9789240015241). World Health Organization. https://iris.who.int/bitstream/handle/10665/338044/9789240015241-eng.pdf

Data School. (2018, June 13). *Outlier detection and removal: z score, standard deviation | Feature engineering tutorial python # 3* [Video]. YouTube. https://www.youtube.com/watch?v=KFuEAGR3HS4&t=734s

OpenAI. (2025). *ChatGPT conversation* [Personal communication]. https://chatgpt.com/c/680da5ca-adc0-8004-8b14-4a75398cd9f5