Data-Driven Analytics for Obesity Management and Business Strategy

Author: Zhetan Zhang, Mingqi Yang

Team: Team ZY

1. Executive Summary

1.1 Decisions to be impacted

- Does this person have obesity?
- How to prevent obesity by changing lifestyle?
- What action should companies do to help obese people?
- What are the most important features that cause obesity?

1.2 Business value

- Nutrition companies and medical companies can design products that are more acceptable and effective for obese population based on their dietary habits
- Rehabilitation centers can provide more personalized services for obesity in order to broaden their prospective customers
- Internet companies can design mobile applications that can help potentially obese people to monitor their weights.

1.3 Data assets

- Estimation of obesity levels based on eating habits and physical condition . (2019). UCI Machine Learning Repository. https://doi.org/10.24432/C5H31Z.

This data set included an abundant set of variables, we can generate a wide-ranging view about how various factors collectively contribute to obesity. For example, by analyzing the features related to individuals' eating habits and their corresponding obesity levels, we can offer guidance to nutrition companies on developing more effective slimming products that are not only appealing to obese population but also effective in reducing weight. Besides, examining physical condition features also provide insights on how to suggest infrastructure companies such as public transit services or medical rehabilitation centers on optimizing convenience or construct appropriate exercise regimens for obese populations.

2. Data Preprocessing

2.1 Data Description

The dataset we are using for this project is called "Estimation of obesity levels based on eating habits and physical condition, " sourced from the UCI Machine Learning Repository. The dataset essentially serves as classification tasks containing 16 features and 2111 samples, with the goal of predicting obesity levels in different individuals. Among 16 features, 6 of them contain nominal data (e.g. Gender, Transportation Used), 2 of them contain ordinal data (e.g.

Consumption of food between meals), 1 of them contains interval data, which is Age, and 7 of them contain ratio data. (e.g. Height, Weight)

To visualize the data, we separate it into two parts— numerical data and categorical data. For numerical data, we use bar plots to show the cumulative distribution of the data set. We have two bar plots which show the distribution of the raw data and the data after normalizing. (See figure 1 and figure 2)

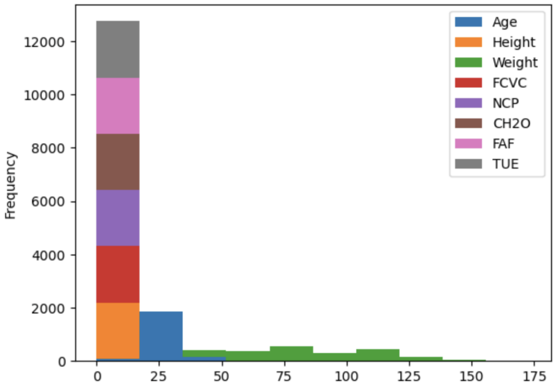


Figure 1: Raw data

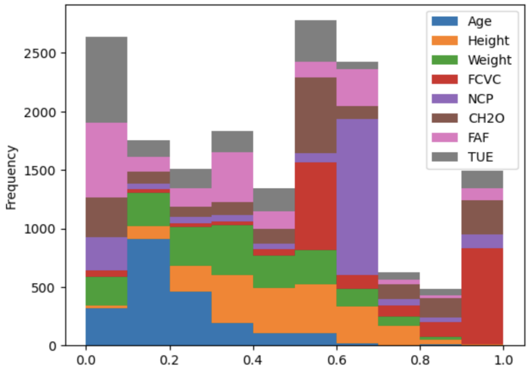


Figure 2: Normalized data

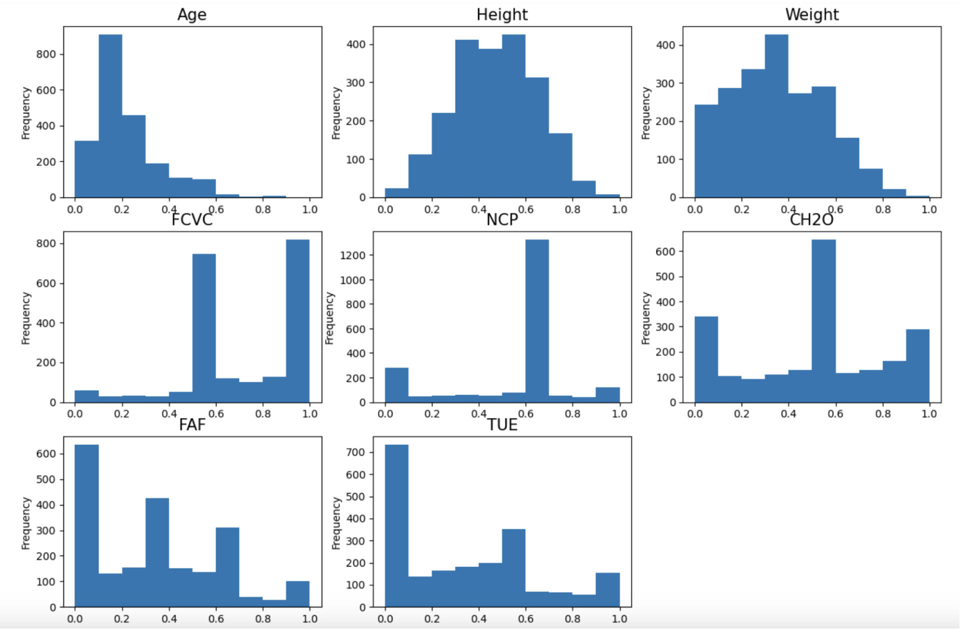


Figure 3: Data distribution

After normalizing, we can see that the data are all shrink into the range [0,1] and have better distribution than the raw data. Then, we can use correlation matrix to see if there any features have high correlation.(See figure 4)

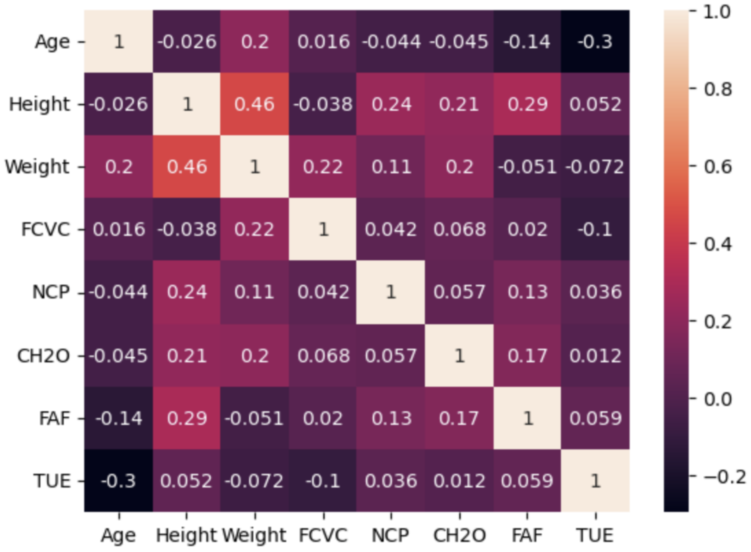


Figure 4: Correlation Heat map

From the correlation heat map, we can see there are no pairs of numeric features that have a correlation greater or equal to 0.8 which means that we can assume each numeric feature is independent of each other.

For categorical data, we used pie charts to show the distribution.(See figure 5)

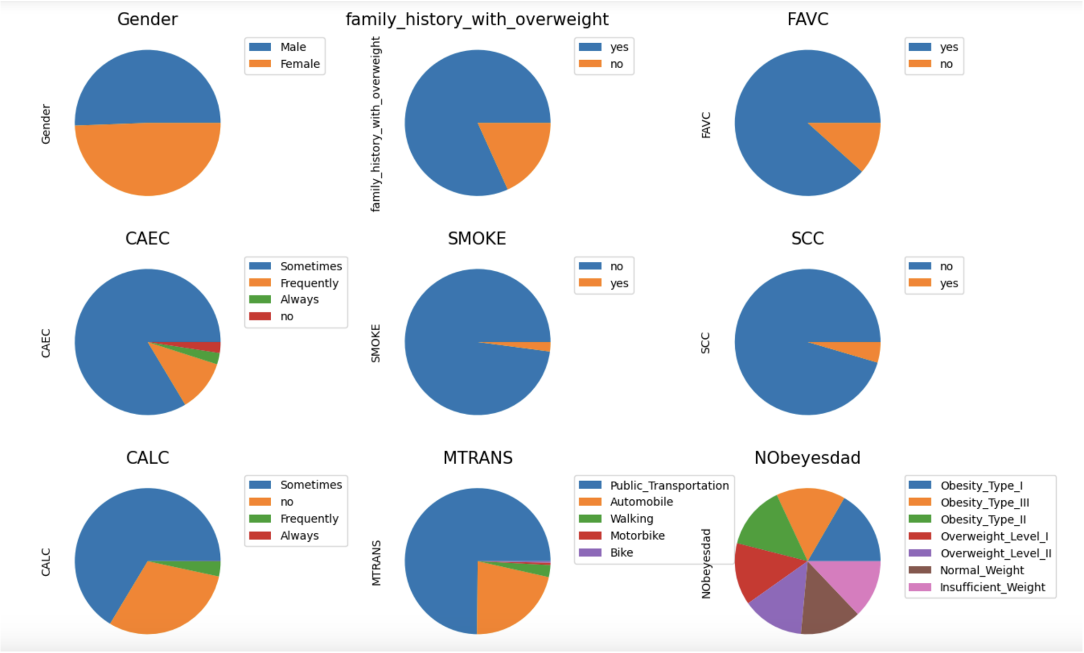


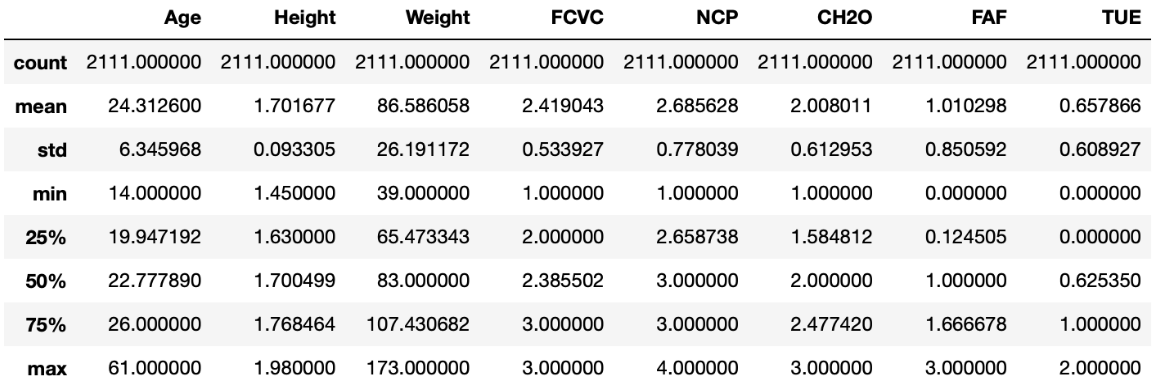
Figure 5: Categorical data Pie charts

2.2 Data Cleaning

We used one-hot-encoding to encode categorical data to transform categorical data into numeric data. For each unique category in the categorical variable, one-hot-encoding will create a new binary column. In the new binary columns, you assign a value of 1 if the original data point belongs to the category represented by that column, and 0 if it does not. These binary columns form a new feature matrix, where each row represents a data point, and each column represents a category from the original categorical variable.

2.3 Outlier Detection

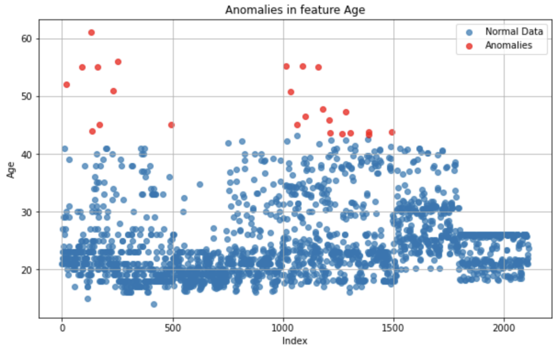
For identifying outliers in our dataset, we primarily employed different statistical methods on both numerical and categorical features. Specifically, for each numerical attribute, we calculate its corresponding standard deviation and mean value, which has shown below.



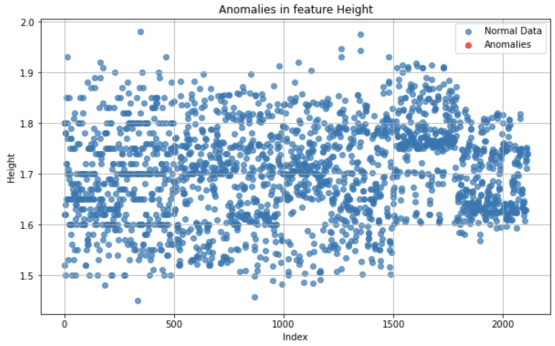
	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
count	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000	2111.000000
mean	24.312600	1.701677	86.586058	2.419043	2.685628	2.008011	1.010298	0.657866
std	6.345968	0.093305	26.191172	0.533927	0.778039	0.612953	0.850592	0.608927
min	14.000000	1.450000	39.000000	1.000000	1.000000	1.000000	0.000000	0.000000
25%	19.947192	1.630000	65.473343	2.000000	2.658738	1.584812	0.124505	0.000000
50%	22.777890	1.700499	83.000000	2.385502	3.000000	2.000000	1.000000	0.625350
75%	26.000000	1.768464	107.430682	3.000000	3.000000	2.477420	1.666678	1.000000
max	61.000000	1.980000	173.000000	3.000000	4.000000	3.000000	3.000000	2.000000

Then, we consider the feature values that are 3 standard deviations from the mean as anomalies. The graphs below shows the results of outlier detection on numeric

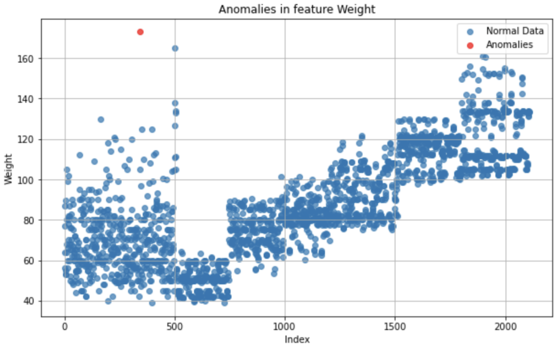
```
Feature Age:
Number of normal data points: 2087
Number of anomalies: 24
Percentage of anomalies: 1.149976042165788%
```



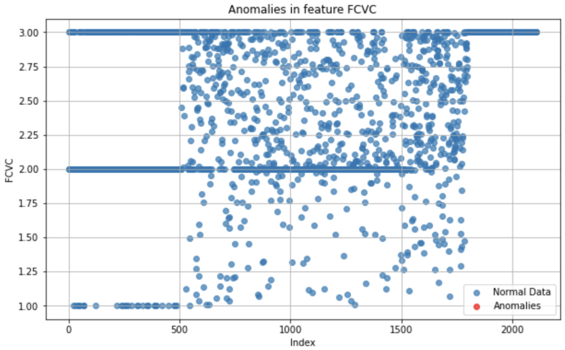
```
Feature Height:
Number of normal data points: 2111
Number of anomalies: 0
Percentage of anomalies: 0.0%
```



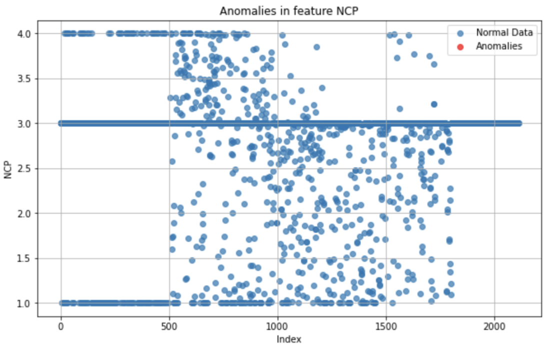
```
Feature Weight:
Number of normal data points: 2110
Number of anomalies: 1
Percentage of anomalies: 0.047393364928909956%
```



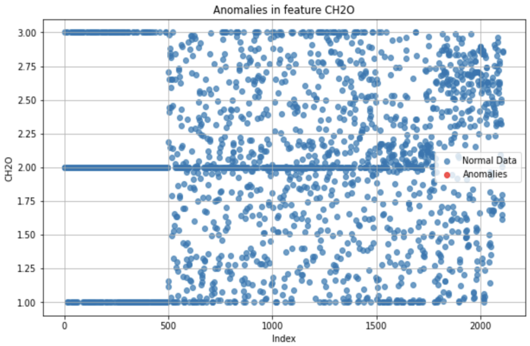
```
Feature FCVC:
Number of normal data points: 2111
Number of anomalies: 0
Percentage of anomalies: 0.0%
```



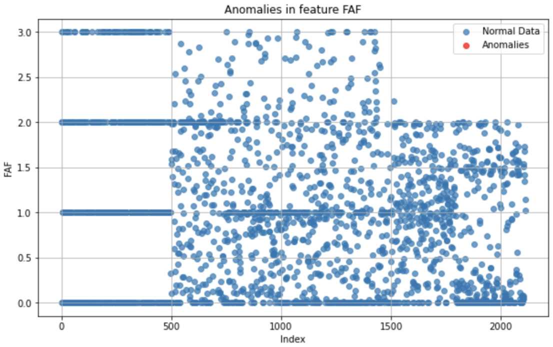
```
Feature NCP:
Number of normal data points: 2111
Number of anomalies: 0
Percentage of anomalies: 0.0%
```



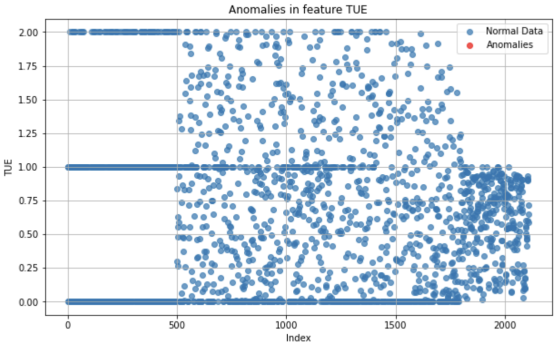
```
Feature CH2O:
Number of normal data points: 2111
Number of anomalies: 0
Percentage of anomalies: 0.0%
```



```
Feature FAF:
Number of normal data points: 2111
Number of anomalies: 0
Percentage of anomalies: 0.0%
```

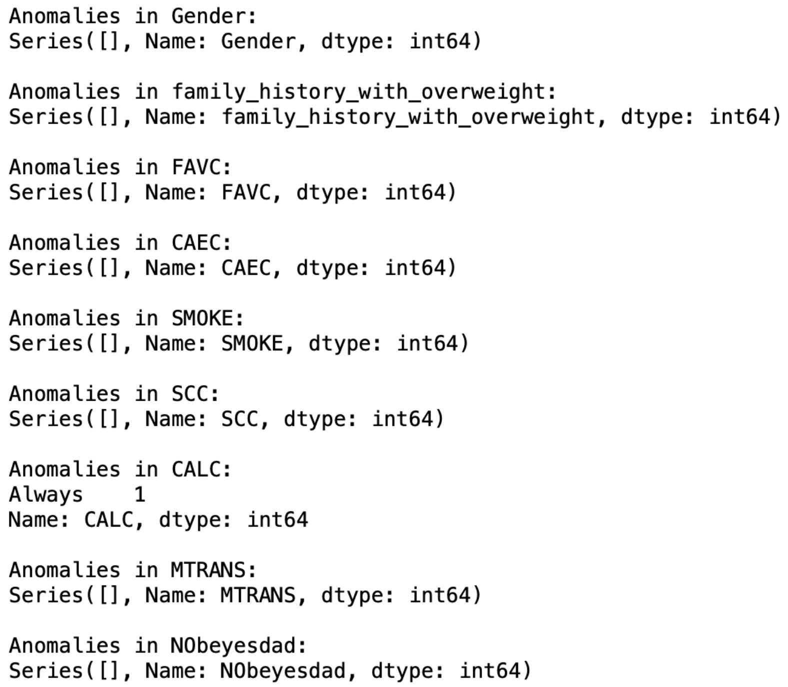


```
Feature TUE:
Number of normal data points: 2111
Number of anomalies: 0
Percentage of anomalies: 0.0%
```



Based on the graphs above, we can see that for most of the features, there is no outlier within the dataset. However, when examining the “age” feature, we observed that samples with ages above 40 are outliers; when examining the “weight” feature, we observe only one sample with weight over 160 kg is outlier. To address “age” outliers, we intend to initially apply models like logistic regression to assess their effect on the models’ accuracy. For the outlier in “weight” feature, we decide to ignore it because it is unlikely to significantly impact the general performance of the model.

For each categorical feature, we have chosen to classify feature values that appears fewer than 5 times as the outliers. Given that our dataset comprises around 2000 samples, we believe this outlier definition is justifiable. The graph below shows the results of outlier detection on categorical dataset:



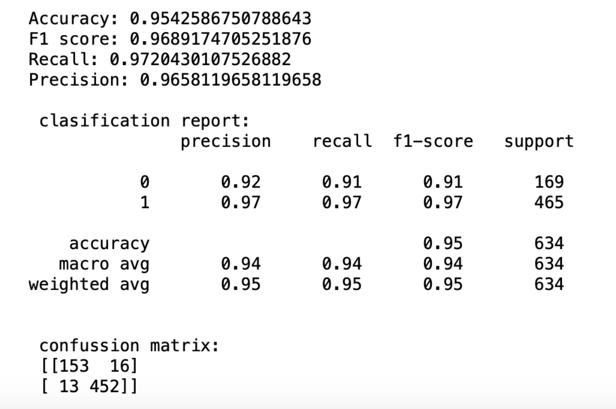
```
Anomalies in Gender:
Series([], Name: Gender, dtype: int64)

Anomalies in family_history_with_overweight:
Series([], Name: family_history_with_overweight, dtype: int64)

Anomalies in FAVC:
Series([], Name: FAVC, dtype: int64)

Anomalies in CAEC:
Series([], Name: CAEC, dtype: int64)

Anomalies in SMOKE:
Series([], Name: SMOKE, dtype: int64)

Anomalies in SCC:
Series([], Name: SCC, dtype: int64)

Anomalies in CALC:
Always    1
Name: CALC, dtype: int64

Anomalies in MTRANS:
Series([], Name: MTRANS, dtype: int64)

Anomalies in NObeyesdad:
Series([], Name: NObeyesdad, dtype: int64)
```

To be specific, we try to create panda series for the outliers within each feature. Based on the graph, you can see that for most of the features, no outlier exists. However, for the feature ‘Consumption of alcohol (CALC)’, we identified a sample with the value "always" as an anomaly. Similar as before, we decide to ignore this single outlier because it will not have a significant effect on the model’s performance.

3. Model Updates

3.1 Logistic Regression

Our first purpose is to use some features to predict whether a person has obesity or not. This problem should be a binary classification problem and we need to separate the target feature “NObeyesdad” into two parts. One part should be “obesity” including all the obesity types as 1(positive class) and “no obesity” including normal weight and insufficient weight as 0(negative class). The model we choose to apply first is logistic regression since logistic regression is a powerful model in binary classification. In addition, by using logistic regression we can see the importance of each feature. This can help us figure out which factors affect obese people the most and help them change their lifestyle. Before applying logistic regression, we split the dataset into training(70%) and testing(30%) set by using train_test_split in sklearn package.
By applying logistic regression, we have the following testing result.



```
Accuracy: 0.9542586750788643
F1 score: 0.9689174705251876
Recall: 0.9720430107526882
Precision: 0.9658119658119658

 clasification report:
               precision    recall  f1-score   support

           0       0.92      0.91      0.91       169
           1       0.97      0.97      0.97       465

    accuracy                           0.95       634
   macro avg       0.94      0.94      0.94       634
weighted avg       0.95      0.95      0.95       634


 confussion matrix:
 [[153  16]
 [ 13 452]]
```

Figure 6: Evaluation matrix

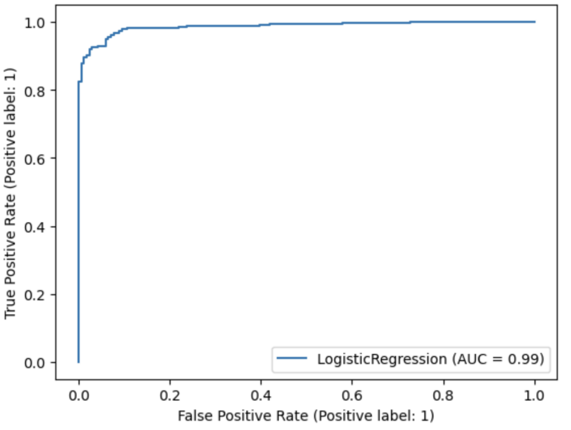
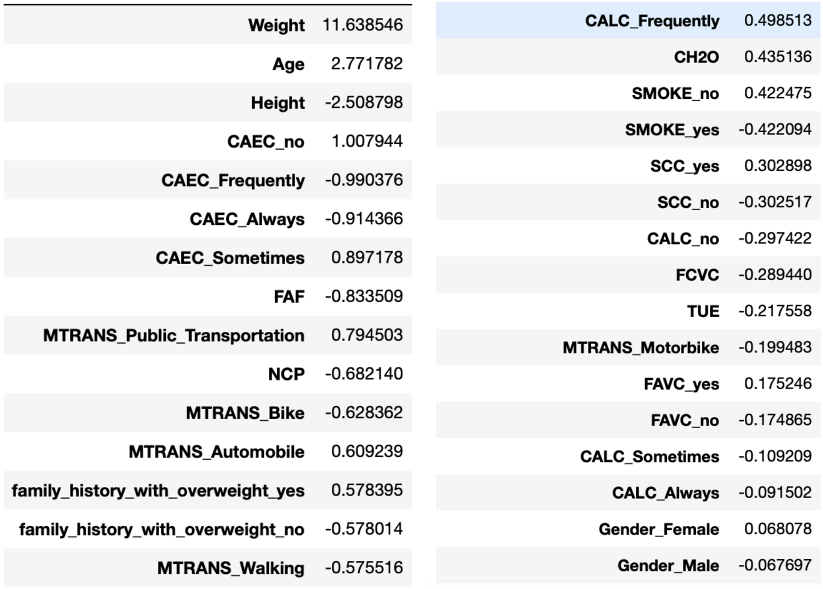


Figure 7: ROC

We can see the accuracy of this model is around 95% and the area under the curve is 0.99 which is really good. Besides this, we can see the weight of each feature. By checking the absolute value of each weight, we have the largest top 3 absolute weight features: “Weight”, “Age”, “Height” which make sense since these are the main factors to determine if a person is overweight or not. See figure 8 as the whole weight list of features.



Weight	11.638546
Age	2.771782
Height	-2.508798
CAEC_no	1.007944
CAEC_Frequently	-0.990376
CAEC_Always	-0.914366
CAEC_Sometimes	0.897178
FAF	-0.833509
MTRANS_Public_Transportation	0.794503
NCP	-0.682140
MTRANS_Bike	-0.628362
MTRANS_Automobile	0.609239
family_history_with_overweight_yes	0.578395
family_history_with_overweight_no	-0.578014
MTRANS_Walking	-0.575516

CALC_Frequently	0.498513
CH2O	0.435136
SMOKE_no	0.422475
SMOKE_yes	-0.422094
SCC_yes	0.302898
SCC_no	-0.302517
CALC_no	-0.297422
FCVC	-0.289440
TUE	-0.217558
MTRANS_Motorbike	-0.199483
FAVC_yes	0.175246
FAVC_no	-0.174865
CALC_Sometimes	-0.109209
CALC_Always	-0.091502
Gender_Female	0.068078
Gender_Male	-0.067697

Figure 8: Feature weight

We can also see that for “CAEC_no” and “CAEC_Sometimes”, they have positive weight. “CAEC_Frequently” and “CAEC_Always” have negative weights. CAEC is the Consumption of food between meals. The positive weight means this term will affect more on predicting the positive result(obesity) and negative weights will affect more on predicting the negative result(not obesity). By this model, we can conclude that more frequent consumption of food between meals will more likely lead to obesity. By using this pattern, we also can conclude the following conditions will likely lead to obesity. Low Physical activity frequency(FAF), less

Number of main meals(NCP), use public transportation or automobile, family history with overweight, frequently Consumption of alcohol(CALC), more Consumption of water daily(CH2O), do not smoke, will do Calories consumption monitoring(SCC), Frequent consumption of high caloric food(FAVC).

3.2 Machine Learning Workflow

The first thing we've done for the workflow is the data checking. Specifically, we acquired a dataset from UCI Machine Learning Repository, where the input features are elements of space **X** and the target variables are elements of space **Y**. Then, examined the dataset to see if there are any missing values and found no absent entries. If it does, we will perform a function ***f*** to handle these missing values,

$$\forall x \in X : x = \begin{cases} x, & \text{if } x \neq \text{null} \\ f(X), & \text{otherwise} \end{cases}$$

where ***x*** is the data point in **X** and ***f(X)*** is the function we apply to fulfill the missing values

Then, we perform descriptive analytics on the dataset by visualizing the distribution of each feature. In particular, we employ bar plots and histogram for numerical features and pie charts for categorical features. Besides, we also draw the correlation matrix to determine whether there are internal relationships among different variables.

After that, we applied one-hot-encoding on the categorical features to prepare them for our model analysis. Moreover, we normalized all the data values between 0 and 1, which is convenient for our potential linear model applications. Mathematically, suppose ***c*** represents each column of **X** and ***D*** represent the domain, we perform one-hot encoding like this:

$$encode(c) = D \in [0, 1]$$

Finally, we choose a logistic regression algorithm for our obesity level estimation. The reason is that our goal is to correctly classify the samples to be obese or not. Then, we randomly split the dataset into a training set and a test set, allocating 70% of the data for training and 30% of the data for testing. In MLM form, we can express our steps like this:

- The input space is $\mathbf{X} = \mathbf{R^m}$ and the output space is $\mathbf{Y} = \mathbf{R}$
- Since we do not embed prior knowledge for now, the error prior $P(\Theta) = 1$
- The corresponding learning Morphism is $z = \Theta^T x$
- The risk function is

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(p^{(i)}) + (1 - y^{(i)})\log(1 - p^{(i)})]$$

where $p = \frac{1}{1+e^{-z}}$

- Then, the optimal parameters are given by

$$\theta^* = \arg\min_\theta J(\theta)$$

and the MLM can be written as

$$ML_{LR} = \left(\mathbb{R}^m, \mathbb{R}, \theta^{\mathbf{T}}\mathbf{x}, P(\theta) = 1, -y\log\left(\frac{1}{1+e^{-\theta^T\mathbf{x}}}\right) - (1-y)\log\left(1 - \frac{1}{1+e^{-\theta^T\mathbf{x}}}\right)\right)$$

4. Source Code

https://github.com/Sonic-zzt/ESE-527-Project.git

5. Next Steps

5.1 Feature Engineering

For now, we are just processing all the features into the model and making the prediction. To further gain more insights into the obesity level, we may want to reduce the importance of some of the features (e.g. Height, Weight, and Ages) to see how the rest of the features interact with obesity levels. Additionally, we may also want to perform obesity estimation solely on eating habits or physical condition to determine which set of features contributed the most to overweight. We may perform Principal Component Analysis (PCA) on the dataset to reduce the noise within the dataset.

5.2 Modeling

Other than logistic regression model, we consider using more classification models to determine which one is the most suitable for our estimation task. More precisely, we may want to use Naive Bayes classifier, Random Forest and even Neural Networks to perform the classification task. Then, we will compare the prediction accuracy, training speed and model complexity across all of these models to determine which one is the best for us to offer specific recommendations to the companies.