7641 Project Final Report - Group 49

Rahul Iyer, Pranav Viswanadha, Deepak Mattipalli, Arya Bhargava, Vineeth Charugundla

Introduction/Background

We seek to help people improve their lifestyle by predicting whether or not they are at risk for obesity.

The dataset we will be using was created based on a survey of people between the ages of 14 and 61 across Mexico, Peru, and Colombia. The dataset has 17 attributes and 2111 rows. 23% of the data was taken from the survey, and the remaining 77% of the data was generated using the SMOTE technique [1]. The dataset groups people into 7 weight categories based on their BMI. The dataset also provides physical characteristics, eating habits, smoking and drinking habits, and lifestyle habits of the person.

Dataset

The study conducted with this dataset attempted to predict obesity level using decision trees, naive bayes classifier, and logistic regression. Decision trees had the best performance at a 97.4% accuracy which makes us curious to evaluate that method as well [2]. A similar study found that CatBoost was the most effective model at predicting obesity [3].

Problem Definition

We are trying to discover the underlying factors behind obesity. Our motivation for exploring this subject is that obesity is a pandemic across the world. It is crucial for people to understand the impact facets of their lives can have on their weight. The prevention and treatment of obesity has benefits that extend beyond weight loss to include improved metabolic health, reduced fat, and enhanced overall quality of life.

Methods, Results, and Discussion

Data Preprocessing

For the data preprocessing, we created a BMI variable and added it to our dataset by taking the weight and dividing it by the square of the height for each row. During preprocessing, we also mapped each value for each category to a number for easier model readability. Here are the category mappings for our dataset:

Gender: Female is coded as 0, and Male as 1
Family History with Overweight: If there is no family history of overweight, it is coded as 0; if there is a family history, it is coded as 1.
Frequent Consumption of High-Calorie Foods (FAVC): If the answer is "no," it is coded as 0, while "yes" is coded as 1.
Eating Habits in Regards to Control (CAEC): "Always" is coded as 0, "Frequently" as 1, "Sometimes" as 2, and "no" as 3.
Smoking Status (SMOKE): "No" is coded as 0, and "yes" as 1.
Consumption of Alcohol (SCC): "No" is coded as 0, and "yes" as 1.
Consumption of High-Calorie Foods Outside of Meals (CALC): "Always" is coded as 0, "Frequently" as 1, "Sometimes" as 2, and "no" as 3.
Mode of Transportation (MTRANS): "Automobile" is coded as 0, "Bike" as 1, "Motorbike" as 2, "Public Transportation" as 3, and "Walking" as 4.
Obesity Level (NObeyesdad): "Insufficient Weight" is coded as 0, "Normal Weight" as 1, "Obesity Type I" as 2, "Obesity Type II" as 3, "Obesity Type III" as 4, "Overweight Level I" as 5, and "Overweight Level II" as 6.

Scikit-learn was utilized within the model to create the training testing split of 70% and 30% respectively. We also used the Pipeline feature to conduct the scaling of the numerical variables as another preprocessing method. Therefore the final pipeline that we used for each model consisted of the training and testing split, the scaling of numerical data, and the fitting of each model.

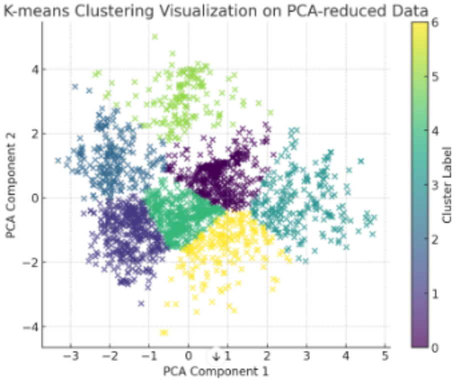
We also used Principal Component Analysis (PCA) as a means of data preprocessing to simplify our dataset and make clustering more effective. PCA helped us by reducing the feature space to a smaller set of key components that captured the most important patterns and differences, allowing easier visualization as well as the removal of the irrelevant details and

Unsupervised Learning

K-Means

In our analysis, we applied K-means clustering to the data, reducing its dimensionality using Principal Component Analysis (PCA) to facilitate visualization and enhance interpretability. The goal of PCA is to reduce the number of dimensions in the data while retaining as much variance as possible. By projecting the data onto two primary components, we can observe cluster patterns in a simplified, two-dimensional space.

The PCA components represent new features that are linear combinations of the original features, capturing the directions of maximum variance. In the K-means clustering visualization, shown below, each data point is plotted based on its values for these two principal components, labeled as PCA Component 1 and PCA Component 2. Different colors represent clusters identified by K-means, where the algorithm groups points based on their proximity within this PCA-reduced space.



To assess the quality of the clustering, we calculated the Silhouette Score, which is a metric that indicates how well-separated the clusters are. The Silhouette Score ranges from -1 to 1:

1: Indicates perfect clustering, where each point is far from other clusters.

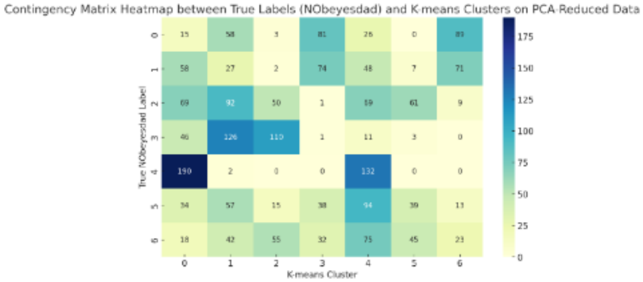
0: Points are close to the boundary between clusters.

-1: Poor clustering, with points likely assigned to the wrong clusters.

The Silhouette Score for our clustering on this dataset is approximately **0.384**, indicating moderate clustering quality with some overlap between clusters. This score suggests that while the clusters are somewhat distinct, there may still be some ambiguity at the cluster boundaries, potentially due to overlapping features or insufficient separation in the data's structure. This could come from the possible overlap between obesity categories.

To further investigate the quality of the model and if the clusters correlated to the obesity categories, we compared the K-means cluster assignments with the true labels of the data. The contingency matrix below highlights the degree of alignment between the clusters and the actual categories. Each cell in the heatmap represents the count of data points belonging to a true label and being assigned to a specific K-means cluster. High counts along the diagonal indicate a better match between clusters and true labels.

From the contingency matrix, we can observe that certain clusters align well with the true labels, while others show more dispersion, indicating that K-means clustering partially captures the natural grouping within the data but may not perfectly match the true categories.



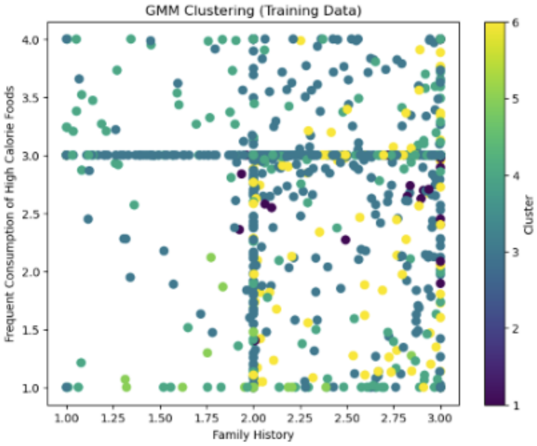
As you can see in the matrix, the clustering doesn’t seem to have done a good job. It didn’t really cluster things based on different obesity categories. We can tell this because no obesity category has a distinct k-means cluster that a majority of its data points belong to. We believe this has to do with the fact that PCA probably reduced a lot of useful information from the data.

Overall, K-means clustering on PCA-reduced data provided a useful, though imperfect, grouping of the data. Given the moderate Silhouette Score and the partial alignment in the contingency matrix, there is definite room for improvement in the model which we can explore going forward. In its current state, we wouldn’t recommend using K-Means clustering to identify potential obesity risk.

GMM

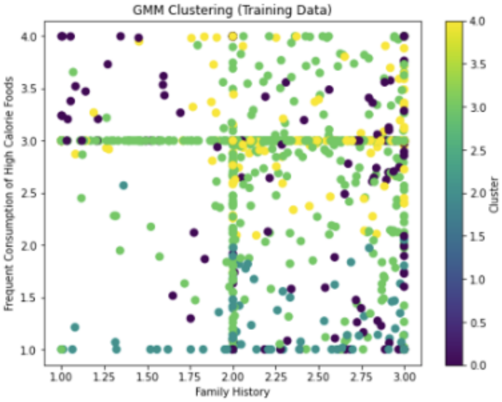
An unsupervised learning method we chose to evaluate our dataset was GMM or Gaussian Mixture Model. We chose this method as it would enable us to obtain different cluster probabilities for each data point without providing any predefined labels. To make sure GMM fairly represented our dataset, we removed the features weight and BMI. This could heavily skew the accuracy of this method, so we conducted the GMM in regards to the rest of the features.

First, we tried a model with 7 components because we had 7 types of obesity categories. We wanted to see if we could get a direct matching from component to category. Using GridSearch, we determined that the best covariance type to use was “full” out of [“full”, “tied”, “diag”, “spherical”]. Upon fitting this model, we got a silhouette score of 0.091. This score is extremely low and indicates that our model is not a good fit with 7 components.

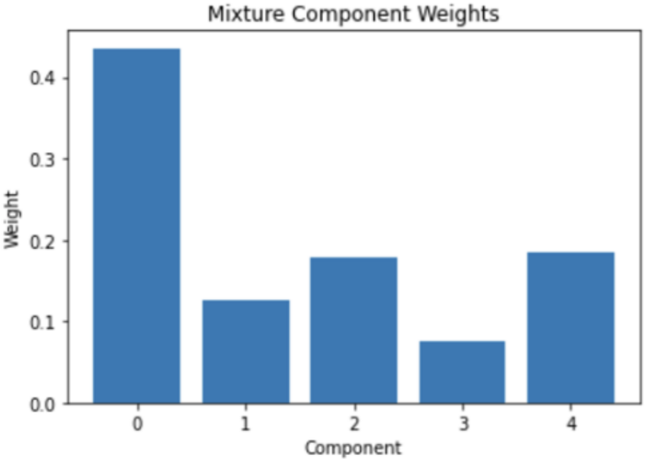


As you can see in the figure above, we created a scatter plot with two variables - family history and frequent consumption of high calorie foods in order to see if there was any separation in the clusters provided. There is no separation amongst the clusters at all, with all the points following no patterns on how they are being mapped. Once we determined that using 7 components wasn’t going to work, we tried changing how many components to use in the model through a grid search.

The parameters that were determined were the GMM covariance type: [“full”, “tied”, “diag”, “spherical”] and the number of components: [4, 5, 6, 7, 8]. The most optimal covariance type was “tied” and the most optimal number of components was 5. We found this pretty interesting as the covariance type changed and the optimal number of components was actually less than the number of categories in the response variable The silhouette score for this was 0.480, which is still relatively weak, but a large improvement on the silhouette score obtained from our initial try.



In the figure above, we created the same scatter plot as before but with our new model. We are able to clearly see that our model has not performed very well. First, we notice that there is heavy overlap between different clusters and that cluster data points are not very differentiable. From this, we are able to determine that these features are not sufficient enough to properly cluster our data. It is clear that this model did not perform well in clustering our data. However, there seems to be more improvement from the initial GMM we created. The top right seems to be mostly cluster 3, the top left seems to be mostly cluster 0, and the bottom left seems to be mostly cluster 2.



As you can see above, we created a bar chart to visualize the weight differences for each of the mixture components. However, the weights are heavily skewed suggesting data isn't balanced between clusters, and that cluster 0 dominates the dataset as it has the largest weight by far. This clearly shows that GMM isn't effectively clustering our data, as the majority of the components have relatively less influence on data points and one component has the majority of the weight. Our data was pretty evenly split up, which indicates that these probabilities are most likely not true. Ultimately, GMM isn't a good model to use to predict obesity classification, especially considering that under its optimal state, we can't match categories to components one-to-one.

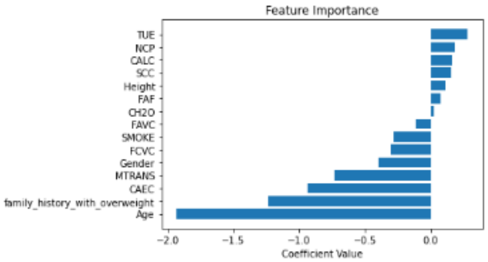
Supervised Learning

Logistic Regression

One supervised learning method we tested is Logistic Regression. Logistic Regression is used to predict the probabilities that a data point belongs to a specific obesity category. The datapoint is then assigned to the category for which it has the highest probability. We wanted to use this because it was simple to implement and easy to interpret the results of. Additionally, we believe taking a probabilistic approach is more cognizant of the fact that the data points are people who all have different obesity risks although they may share some characteristics.

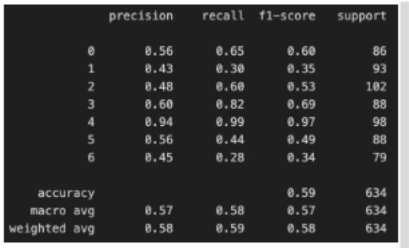
Prior to fitting the model, we removed the columns weight and BMI because they heavily skewed the accuracy of the model. As a part of the logistic regression model, we performed hyperparameter tuning using GridSearch in order to ensure that we used the most optimal model based on log loss. Log loss measures how well a model's predicted probability distribution aligns with the true labels, penalizing large differences between predicted probabilities and actual outcomes. The parameters we search acrossed were: C = [0.01, 0.1, 1, 10, 100], Penalty = [l1, l2], Solver = [liblinear, lbfgs, newton-cg], and Tolerance = [1e-4, 1e-3, 1e-2, 1e-1]. By searching this entire grid, we searched across 400 models (total number is 600, but some solvers are not compatible with some penalties). After conducting this search, the best model we found was a logistic regression model with C = 1, Penalty = L2, Solver = lbfgs, Tolerance = 0.01 with a negative log loss of -1.147.

We were curious about the feature weight in this regression model, and we received the following results:



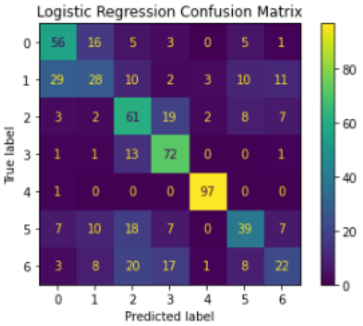
It seems that age, family history with being overweight, whether they eat food between their meals (CAEC), and their method of transportation (MTRANS) are the most weighted features (have the highest coefficient weights). We are curious why the transportation method and the age are listed as features with a high weight; however it does make sense the family history with being overweight and whether they eat food between meals have high weights.

We then wanted to understand the accuracy of the model. This model provided us the following results:



```
              precision    recall  f1-score   support

           0       0.56      0.65      0.60        86
           1       0.43      0.30      0.35        93
           2       0.48      0.60      0.53       102
           3       0.60      0.82      0.69        88
           4       0.94      0.99      0.97        98
           5       0.56      0.44      0.49        88
           6       0.45      0.28      0.34        79

    accuracy                           0.59       634
   macro avg       0.57      0.58      0.57       634
weighted avg       0.58      0.59      0.58       634
```

As you can see, we have an accuracy of 59% with the ability to predict obesity class 4 very well and others not so well. We also saw the following confusion matrix:



True label \ Predicted label	0	1	2	3	4	5	6
0	56	16	5	3	0	5	1
1	29	28	10	2	3	10	11
2	3	2	61	19	2	8	7
3	1	1	13	72	0	0	1
4	1	0	0	0	97	0	0
5	7	10	18	7	0	39	7
6	3	8	20	17	1	8	22

There seemed to be a lot of confusion between class 1 being guessed as both class 0 and class 1. This makes sense however, as these categories are closely related with 0 being underweight and 1 being normal weight. There was also a lot of confusion for class 6 being guessed as class 2, 3, and 6. This also makes sense as 2, 3, 6 are close together with 6 being overweight level II, 2 being obesity level I, and 3 being obesity level II. Overall, this model wasn't as accurate as the other supervised learning technique we did. We would not recommend using this approach as it mismatched a lot of the results. We hypothesize that this model might perform better if we reduce the number of classes to underweight, normal, overweight, and obesity rather than split them up into levels and will consider this in our next steps.

KNN

Another supervised learning method we chose to evaluate our dataset was KNN or K-nearest neighbors. We chose this method as it would allow us to easily classify our data points based on similarity to other neighbor points. To make sure KNN properly represented our dataset, we removed the features weight and BMI as they would heavily skew the accuracy of our method. We used hyperparameter tuning to figure out which model would provide the highest accuracy. The parameters we tuned were the number of neighbors: [3, 5, 7, 9, 11] and the weights: [distance, uniform].

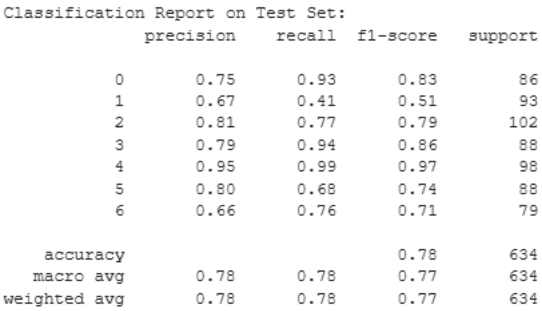
Through a grid search, we were able to find that these were the best parameters to use, and that is shown below. The most optimal number of neighbors is 3 and the most optimal weight metric is distance. The cross-validation score (in this case accuracy) for the optimal model was 0.794.

Best parameters found: {'knn__n_neighbors': 3, 'knn__weights': 'distance'}

Best cross-validation score: 0.7935003385240352

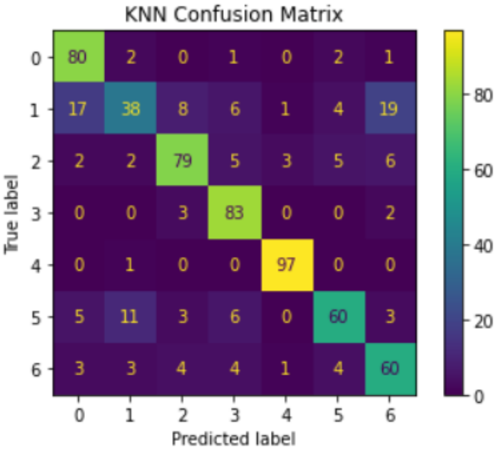
Best estimator: Pipeline(steps=[('scaler', StandardScaler()), ('knn', KNeighborsClassifier(n_neighbors=3, weights='distance'))])

This was the classification report we were able to obtain after running KNN on our dataset. From this, we are able to see that the accuracy was 0.78. Our model was able to perform relatively well because we had a sufficiently large and representative dataset. In addition to this, our classes were well balanced.



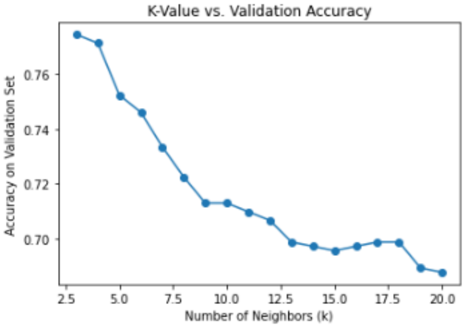
```
Classification Report on Test Set:
              precision    recall  f1-score   support

           0       0.75      0.93      0.83        86
           1       0.67      0.41      0.51        93
           2       0.81      0.77      0.79       102
           3       0.79      0.94      0.86        88
           4       0.95      0.99      0.97        98
           5       0.80      0.68      0.74        88
           6       0.66      0.76      0.71        79

    accuracy                           0.78       634
   macro avg       0.78      0.78      0.77       634
weighted avg       0.78      0.78      0.77       634
```

Another visualization we decided to display our results was the KNN confusion matrix. We decided to use this, as it compared how predictions actually compared to the true values of each class. In our matrix, you can see that predicting class 1 did not perform too well; this is because class 0 is insufficient_weight and class 1 is normal_weight. So, this might cause some inaccuracy as it could be hard for the model to correctly distinguish the two, and there were 17 misclassified. Then for class 5 and class 6, they are overweight level_1 and overweight level_2, which also had a relatively lower amount of correct predictions. This is because overweight level_1 and normal_weight might be slightly tough to distinguish as we see some incorrect predictions for that class. Furthermore, the overweight level_1 and overweight level_2 might also be tough to distinguish from each other.



True label \ Predicted label	0	1	2	3	4	5	6
0	80	2	0	1	0	2	1
1	17	38	8	6	1	4	19
2	2	2	79	5	3	5	6
3	0	0	3	83	0	0	2
4	0	1	0	0	97	0	0
5	5	11	3	6	0	60	3
6	3	3	4	4	1	4	60

The final visualization we used for KNN was the K-Value vs Validation Accuracy. We chose this visualization as it would clearly display the accuracy of our results regarding the number of neighbors we specified. As you can see, here the optimal number of neighbors is 3, with 4 being very close. However, after this there is a visible dropoff in the accuracy. Though, this is a little over a percent drop, this is very important in creating the most accurate model we can have.

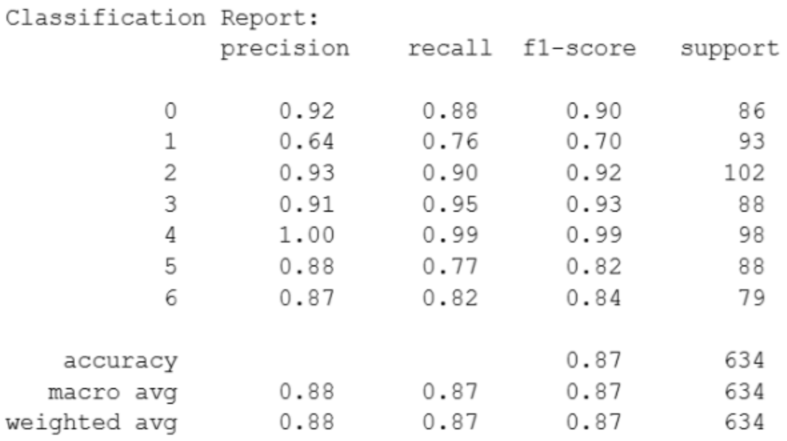


Random Forest

Another supervised learning model we chose to evaluate our dataset was Random Forest. This method was selected for its ability to be able to handle high-dimensional data and to avoid overfitting through the use of multiple decision trees. To make sure Random Forest properly represented our dataset, we removed the features weight and BMI as they would heavily skew the accuracy of our method. To optimize our Random Forest model, we performed hyperparameter tuning using grid search and the following parameters were explored: n_estimators: [50, 100, 150], max_depth: [None, 10, 20, 30], min_samples_split: [2, 5, 10], min_samples_leaf: [1, 2, 4].

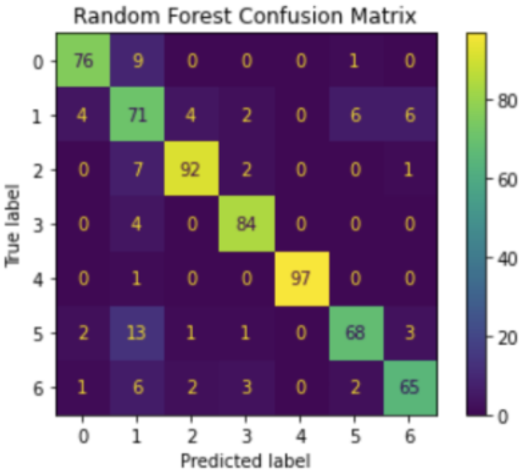
Through a grid search, we were able to find that the best parameters to use are max_depth = 20, min_samples_leaf = 1, min_samples_split = 2, and n_estimators = 150. The cross-validation score (in this case accuracy) for the optimal model was 0.879.

This was the classification report we were able to obtain after running Random Forest on our dataset. Our model was able to perform extremely well because we had a sufficiently large and representative dataset.

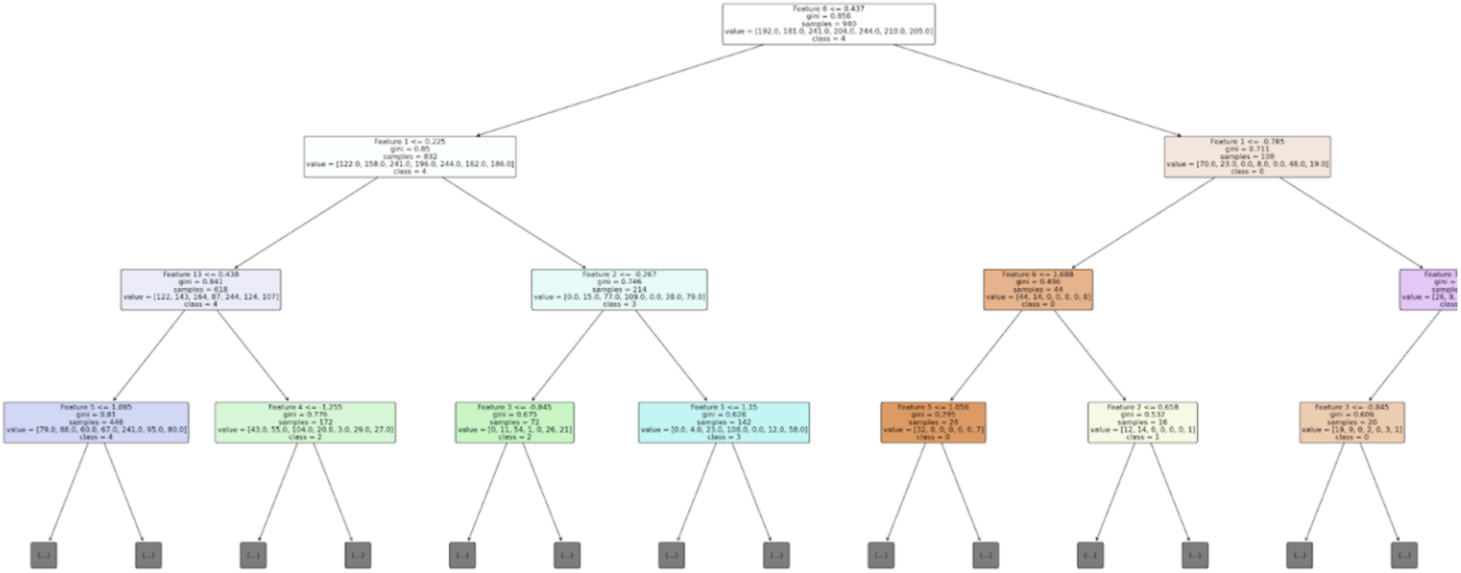


```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.88      0.90        86
           1       0.64      0.76      0.70        93
           2       0.93      0.90      0.92       102
           3       0.91      0.95      0.93        88
           4       1.00      0.99      0.99        98
           5       0.88      0.77      0.82        88
           6       0.87      0.82      0.84        79

    accuracy                           0.87       634
   macro avg       0.88      0.87      0.87       634
weighted avg       0.88      0.87      0.87       634
```

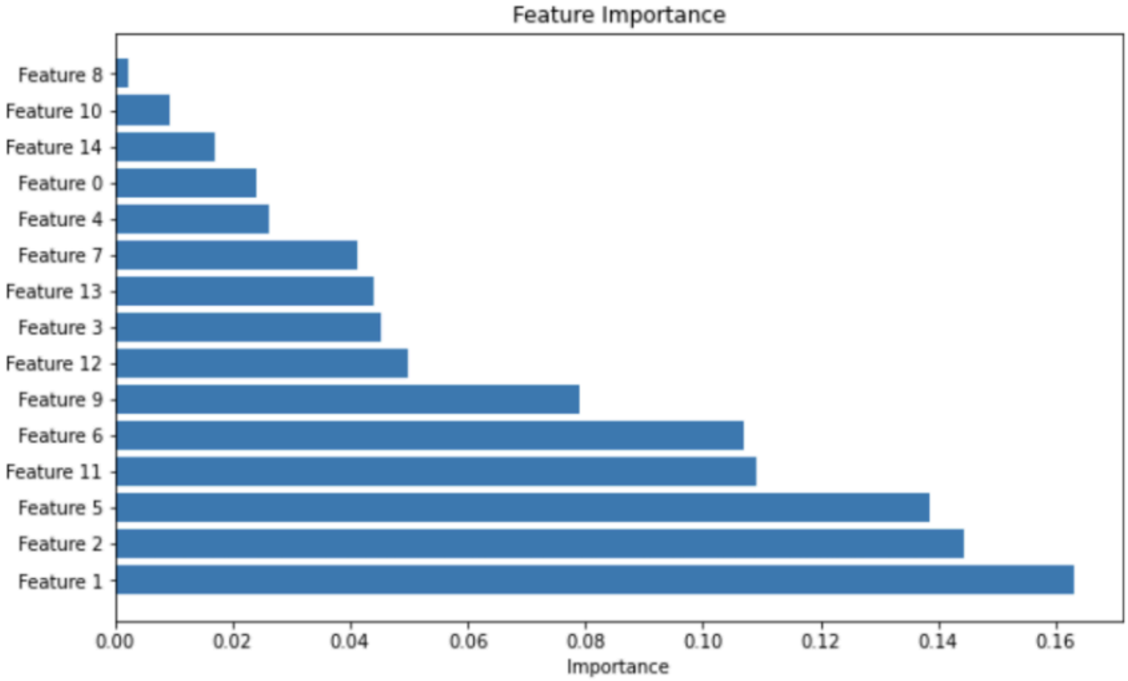
Every class was able to be classified really well except for class 1. We created a confusion matrix below to explore why this was the case. The diagonal entries represent correct predictions, and these values are generally high, indicating strong performance overall. Class 4 (Obesity Level 3) has near-perfect predictions, with 97 instances correctly classified and no significant misclassifications. Similarly, classes 0 (Insufficient Weight), 2 (Obesity Level 1), and 3 (Obesity Level 2) exhibit strong accuracy, with minimal confusion. However, class 1 (Normal Weight) shows notable misclassification, with a pretty even spread of what these points are being misclassified as. These misclassifications make sense, however as the data points are being classified as the neighboring weight classes. Similarly, class 5 (Overweight Level 1) was being predicted as Normal weight which makes sense as it is the neighboring classification.



We depict a few levels of a single decision tree below. It starts with Feature 6, splitting the dataset into two groups with high impurity (Gini = 0.856). Further splits based on features like Feature 1 and Feature 6 refine the groups, reducing impurity and making the class predictions more accurate. As the tree grows the Gini index reduces which is a good sign, showing that our tree will become more pure as it grows.

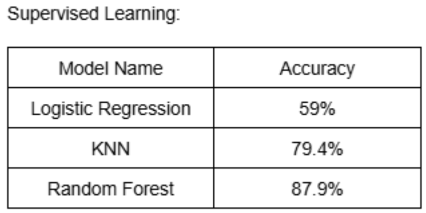


Our final visualization for Random Forest is a feature importance bar graph seen below. Feature 1 is the most important, as it contributes significantly to the model's decision-making, followed by Feature 2, Feature 5, and Feature 11, which also play big roles in distinguishing between classes. Moderately important features, such as Feature 6, Feature 9, and Feature 12, provide support in classification but are less pivotal in decision making. In contrast, features like Feature 8, Feature 10, and Feature 14 have a minimized impact on the model, fostering us to conclude that they contribute little to improving accuracy and could be the subjects of feature reduction. This indicates to us that the more important features most likely are significant factors to improve when reducing the chances of obesity.



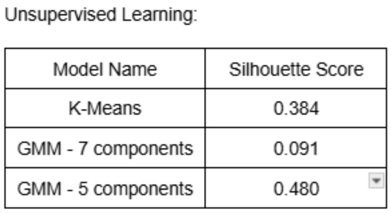
Model Comparison & Next Steps

It seems as though unsupervised learning is not the most appropriate way to go about predicting obesity. Both the unsupervised learning models we tried performed poorly and didn't allow us to clearly classify data points into obesity categories. On the other hand, supervised learning performed pretty well achieving accuracies of ~60% or greater. We feel confident that if provided new data points we would be able to classify the obesity category well. Out of the supervised learning techniques, Random Forest performed the strongest with an accuracy of 87.9%. This is outstanding performance in a model and indicates that studies should lean towards using random forest classification or some sort of decision tree classification. You can see the tables below comparing our models and their scoring metrics.



Supervised Learning:

Model Name	Accuracy
Logistic Regression	59%
KNN	79.4%
Random Forest	87.9%



Unsupervised Learning:

Model Name	Silhouette Score
K-Means	0.384
GMM - 7 components	0.091
GMM - 5 components	0.480

We think Random Forest excelled in capturing the complex relationships that exist between the different features of the data. It also does well at handling high-dimensional data. Most of the other models don't perform well with high-dimensional data, so Random Forest really stood out. However, a significant limitation of this model is its complexity/lack of interpretability. If using this model in the real world, we would have to explain it to medical professionals and patients who might not have the knowledge to understand how it works and therefore not necessarily believe the results. This could hamper its effectiveness as a tool to help people live better lifestyles.

KNN excels in its simplistic nature and the fact that it was easy to implement. However, this simplicity is also a reason it didn't work as strongly as RF. It is hard to capture the relationships in high-dimensional data for KNN, so that is probably why it didn't perform as well for our data. Also, it doesn't really capture non-linear data well which is what our data is. We hypothesize if we were to reduce our feature space to two dimensions using PCA that it might perform even stronger. Similarly, Logistic Regression works well because of its interpretability and its fast nature. However, it makes some key assumptions about the data that need to be true in order for it to perform well. In our case, these assumptions are violated as our data is not linear and very complex. This is probably why LogReg didn't perform that well.

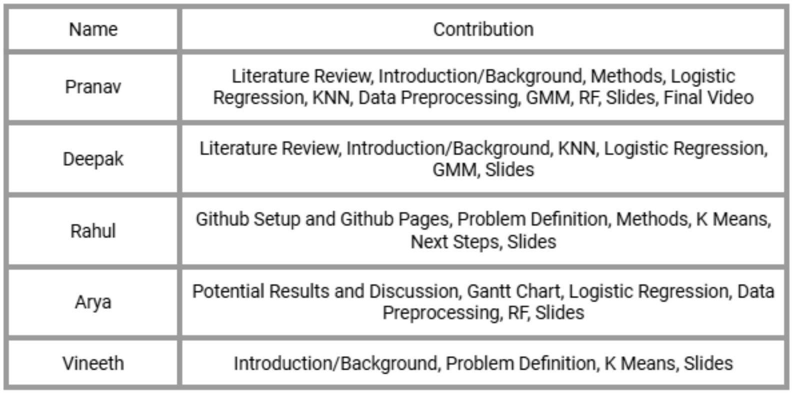
For unsupervised learning, GMM did better than K-Means because of its ability to model overlapping clusters really well because it uses covariance matrices. However, its main pitfall - similar to LogReg - is that it has key assumptions that may be violated with our data. It assumes the underlying data comes from a Gaussian distribution, which is probably not true for some of our variables. Models like KNN and RF handle this limitation which is why they performed better than GMM did. Also, similar to RF, it is highly complex and lacks interpretability which is a tradeoff for its ability to model overlapping clusters. Ultimately, these limitations outweigh the positives - especially since a GMM with 7 components performed exceptionally poor amongst our approaches.

Lastly, K-Means excels in its simplistic nature and easy interpretability. However a key limitation - which was seen in our data - is that it really only performs well on well separated data; because of this, it assumes there are spherical clusters. We elected to perform PCA on our data to help it be easier for K-Means to perform well, but it still didn't perform as well as we hoped. GMM can handle both of these issues pretty well which is why the GMM model with 5 components ultimately outperformed it. K-Means did outperform GMM in terms of its separability of clusters. In the figure, you can see the clusters are very distinct and there are patterns for why each datapoint was assigned a given cluster.

In terms of next steps, we would want to continue to explore different models and see how we could continue to refine our own models or even combine them. We are very curious how a neural network would perform if used for this classification task. We also would like to see how we can use our model to help other people and research real-world applications for being able to predict obesity risk based on these lifestyle characteristics. Ultimately, our goal is to help people stay fit and make the right lifestyle choices, so we would want to research applying data science towards this goal.

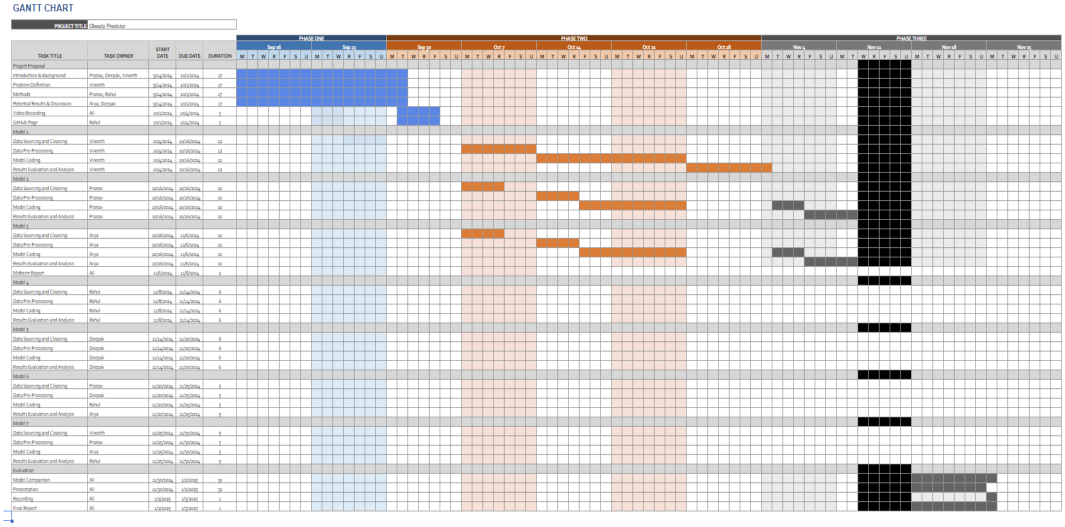
Contribution Table

Final Contribution Table



Name	Contribution
Pranav	Literature Review, Introduction/Background, Methods, Logistic Regression, KNN, Data Preprocessing, GMM, RF, Slides, Final Video
Deepak	Literature Review, Introduction/Background, KNN, Logistic Regression, GMM, Slides
Rahul	Github Setup and Github Pages, Problem Definition, Methods, K Means, Next Steps, Slides
Arya	Potential Results and Discussion, Gantt Chart, Logistic Regression, Data Preprocessing, RF, Slides
Vineeth	Introduction/Background, Problem Definition, K Means, Slides

Gantt Chart



GANTT CHART

PROJECT TITLE Obesity Predictor

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
Project Proposal				
Introduction & Background	Pranav, Deepak, Vineeth	9/14/2024	10/1/2024	17
Problem Definition	Vineeth	9/14/2024	10/1/2024	17
Methods	Pranav, Rahul	9/14/2024	10/1/2024	17
Potential Results & Discussion	Arya, Deepak	9/14/2024	10/1/2024	17
Video Recording	All	10/1/2024	10/4/2024	3
GitHub Page	Rahul	10/1/2024	10/4/2024	3
Model 1				
Data Sourcing and Cleaning	Vineeth	10/4/2024	10/16/2024	12
Data Pre-Processing	Vineeth	10/4/2024	10/16/2024	12
Model Coding	Vineeth	10/4/2024	10/16/2024	12
Results Evaluation and Analysis	Vineeth	10/4/2024	10/16/2024	12
Model 2				
Data Sourcing and Cleaning	Pranav	10/16/2024	10/26/2024	10
Data Pre-Processing	Pranav	10/16/2024	10/26/2024	10
Model Coding	Pranav	10/16/2024	10/26/2024	10
Results Evaluation and Analysis	Pranav	10/16/2024	10/26/2024	10
Model 3				
Data Sourcing and Cleaning	Arya	10/26/2024	11/6/2024	10
Data Pre-Processing	Arya	10/26/2024	11/6/2024	10
Model Coding	Arya	10/26/2024	11/6/2024	10
Results Evaluation and Analysis	Arya	10/26/2024	11/6/2024	10
Midterm Report	All	11/6/2024	11/8/2024	2
Model 4				
Data Sourcing and Cleaning	Rahul	11/8/2024	11/14/2024	6
Data Pre-Processing	Rahul	11/8/2024	11/14/2024	6
Model Coding	Rahul	11/8/2024	11/14/2024	6
Results Evaluation and Analysis	Rahul	11/8/2024	11/14/2024	6
Model 5				
Data Sourcing and Cleaning	Deepak	11/14/2024	11/20/2024	6
Data Pre-Processing	Deepak	11/14/2024	11/20/2024	6
Model Coding	Deepak	11/14/2024	11/20/2024	6
Results Evaluation and Analysis	Deepak	11/14/2024	11/20/2024	6
Model 6				
Data Sourcing and Cleaning	Pranav	11/20/2024	11/25/2024	5
Data Pre-Processing	Deepak	11/20/2024	11/25/2024	5
Model Coding	Rahul	11/20/2024	11/25/2024	5
Results Evaluation and Analysis	Arya	11/20/2024	11/25/2024	5
Model 7				
Data Sourcing and Cleaning	Vineeth	11/25/2024	11/30/2024	5
Data Pre-Processing	Pranav	11/25/2024	11/30/2024	5
Model Coding	Arya	11/25/2024	11/30/2024	5
Results Evaluation and Analysis	Rahul	11/25/2024	11/30/2024	5
Evaluation				
Model Comparison	All	11/30/2024	1/2/2025	32
Presentation	All	11/30/2024	1/2/2025	32
Recording	All	1/2/2025	1/3/2025	1
Final Report	All	1/2/2025	1/3/2025	1

References

[1] F. M. Palechor and A. de la H. Manotas, "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico," Data in Brief, vol. 25, 2019, Art. no. 104344.

[2] E. De-La-Hoz-Correa, F. E. Mendoza-Palechor, A. De-La-Hoz-Manotas, R. C. Morales-Ortega, and B. A. Sánchez Hernández, "Obesity Level Estimation Software based on Decision Trees," Journal of Computer Science, vol. 15, no. 1, pp. 67-77, 2019.

[3] W. Lin, S. Shi, H. Huang, J. Wen, and G. Chen, "Predicting risk of obesity in overweight adults using interpretable machine learning algorithms," Frontiers in Endocrinology, vol. 14, Art. no. 1292167, 2023.

[4] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.