Statistics 101C Final Project

Predicting Obesity Status

Cameryn Harvey, Ernest Leung, Jisun Min, Keawn Tandon

Department of Statistics and Data Science, UCLA

Contents

Abstract

The overarching theme of this Kaggle project is to predict Obesity status by exploring various statistical prediction models and methods and applying them to the given datasets. This report will detail the steps our team took to predict Obesity Status. This includes the introduction, exploratory data analysis, data preprocessing, model exploration, model simplification, and conclusion in which we discuss limitations and areas of improvement of our approach.

Our final model is a Random Forest Model of 16 predictors ('CH2O,' 'FAF,' 'FCVC,' 'NCP,' 'CAEC,' 'SCC,' 'FAVC,' 'TUE,' 'SMOKE,' 'MTRANS,' 'CALC,' 'race,' 'height,' 'age,' 'gender,' and 'family_history_with_overweight'). This model achieved a 0.99456 accuracy score on Kaggle, placing our team at 11th on the leaderboard.

1. Introduction

a. Background

Obesity is a significant public health issue, driven by various factors such as demographic characteristics, lifestyle habits, and genetic predispositions. This project leverages a dataset containing 32,014 observations with 29 predictors, capturing diverse elements like physical activity frequency, daily water intake, dietary habits, and family history of being overweight. By examining these variables, we aim to better understand how different factors contribute to obesity status, laying the groundwork for creating predictive models that can aid in early identification and prevention of obesity.

b. Objective

The objective of this project is to explore and analyze the relationships between 29 predictors and obesity status through statistical and machine learning methods. Our work involves performing detailed exploratory data analysis, identifying key predictors, and evaluating multiple models to develop a robust and interpretable prediction framework. By focusing on simplicity and accuracy, this project seeks to provide meaningful insights into the primary contributors to obesity and create models that can effectively classify individuals as “Obese” or “Not Obese”.

2. Exploratory Data Analysis

a. Data Overview and Distribution

The data consists of 11 numerical variables and 19 categorical variables, with target variable ‘ObStatus’ consisting of categorical values ‘Not Obese’ and ‘Obese’. The distribution of the target variable is as follows in Figure 1.

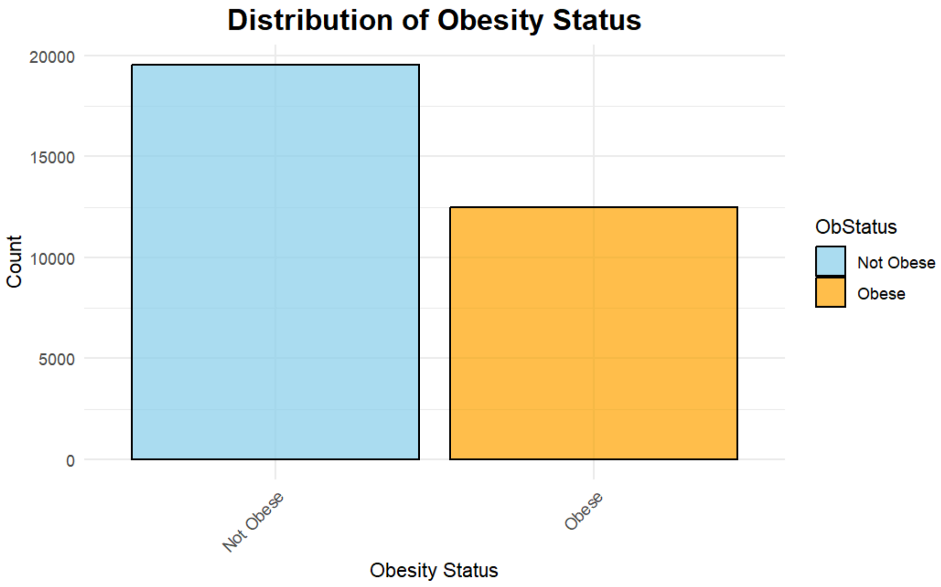

Figure 1. Distribution of Obesity Status

Below in Figures 2-5 is the distribution of four main predictors in relation to the target variable: numerical variables ‘FAF’ and ‘CH2O,’ and categorical variables ‘Gender’ and ‘family_history_with_overweight.’

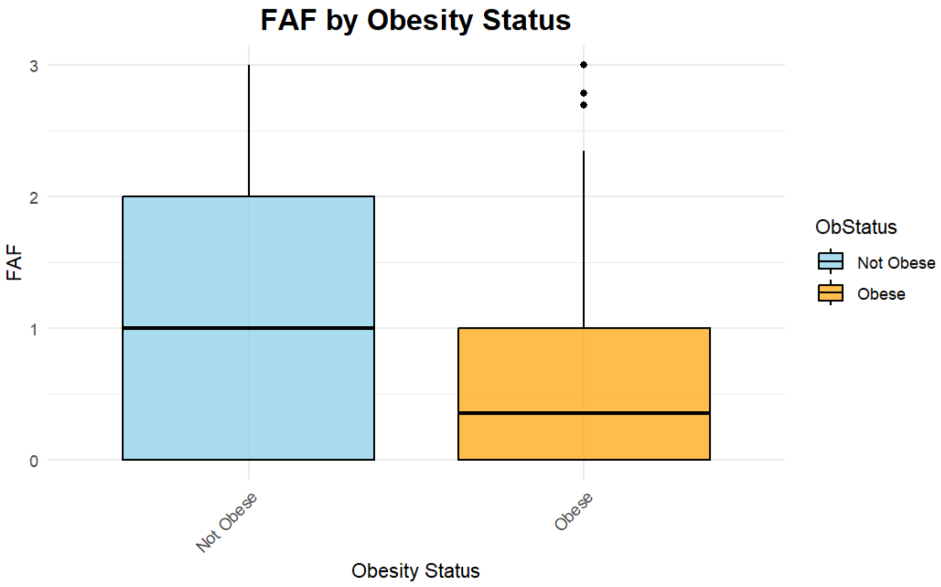

Figure 2. ‘FAF’ by Obesity Status

The ‘Obese’ group tends to have lower ‘FAF’ relative to the ‘Not Obese’ group by Figure 2, while the ‘Not Obese’ group shows a wider distribution.

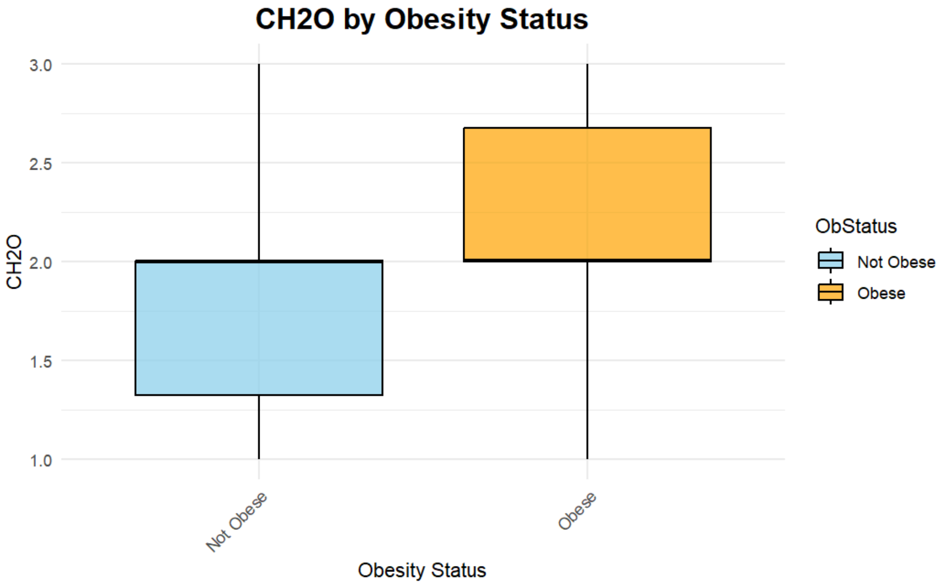

Figure 3. ‘CH2O’ by Obesity Status

Higher ‘CH2O’ is seen in the ‘Obese’ compared to the ‘Not Obese’ group as shown in Figure 3.

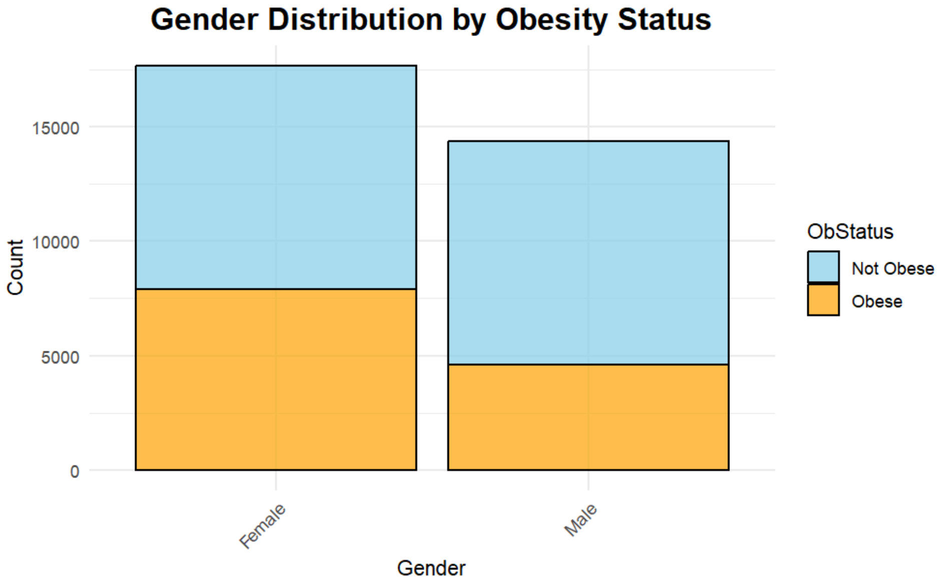

Figure 4. Gender Distribution by Obesity Status

Figure 4 showcases that females in the data have a higher 'Obese' count, while both males and females have similar 'Not Obese' observations.

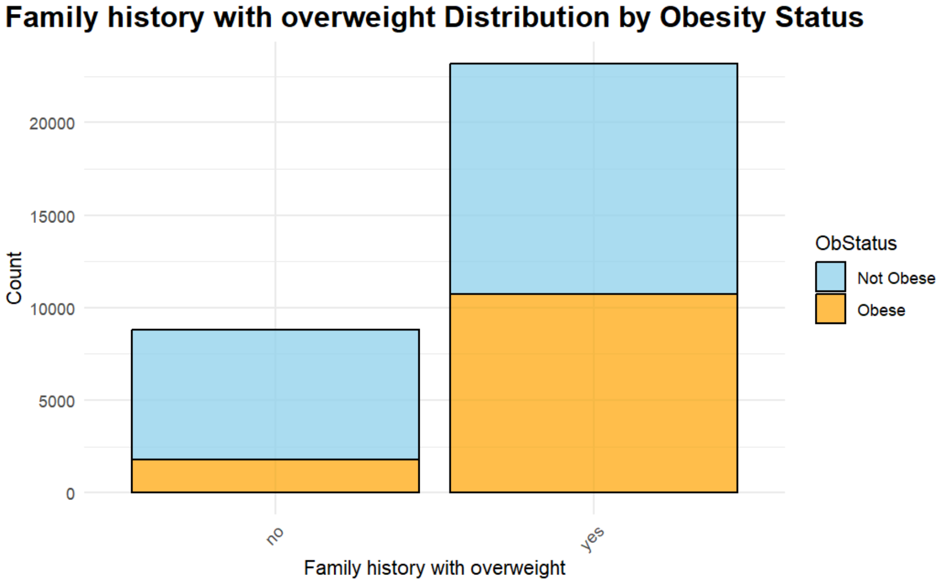

Figure 5. Family History with Overweight Distribution by Obesity Status

The data indicates that those with a family history of overweight tend to have higher counts of 'Obese' compared to the 'Not Obese' group as seen in Figure 5.

b. Correlation Analysis

As correlated variables have redundant information and can harm the model performance, we conducted correlation analysis by drawing a heatmap and testing the multicollinearity using the variance inflation factors (VIF).

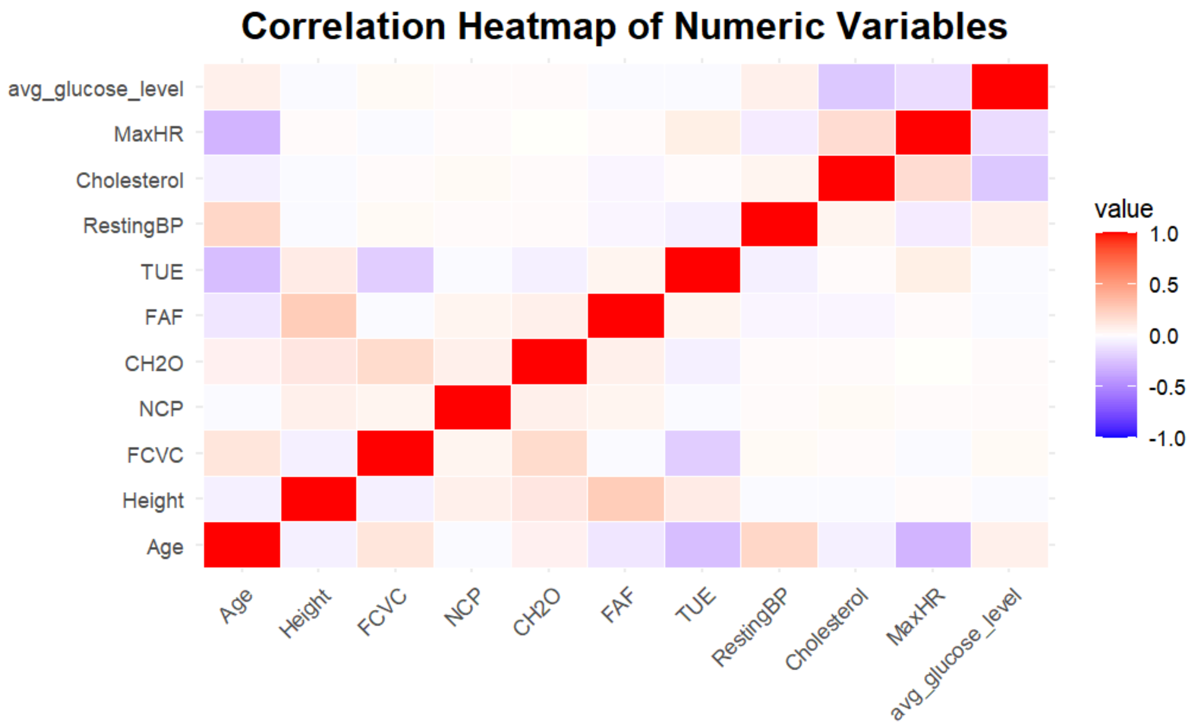

Figure 6. Correlation Heatmap of Numeric Variables

First, we drew the heatmap only using numerical variables. We could see variables with some extent of correlation in Figure 6, such as ‘MaxHR’ with ‘Age’, and ‘TUE’ with ‘Age.’ But, after calculating VIF to check the multicollinearity of the variables, it was seen that all resulting values were less than 2. As such, we did not diagnose significant multicollinearity among the predictors.

3. Methodology

We chose five models to use as candidate models based on each of their strengths and weaknesses. The five models we explored were logistic regression, linear discriminant analysis (LDA), qualitative descriptive analysis (QDA), K-nearest neighbors (KNN), and Random Forest.

They are all commonly used machine learning models for classification tasks. Logistic Regression is a simple and efficient model that works well when the relationship between the features and target variable is linear, offering easy interpretability. LDA, while also linear, excels in maximizing class separation when the data follows a Gaussian distribution and has a common covariance structure, making it ideal for certain problems with simpler class distributions. QDA builds upon LDA by allowing each class to have its own covariance matrix, which results in quadratic decision boundaries. This flexibility makes QDA more suited for complex data where class distributions vary significantly, but come with a higher risk of overfitting, especially with smaller datasets. KNN, a non-parametric model, does not assume a specific distribution for the data and can handle non-linear decision boundaries, offering great flexibility. However, it becomes computationally expensive with large datasets, particularly in prediction, as it needs to calculate distances to all training points. Random Forest, an ensemble method, uses multiple decision trees to make predictions, making it robust to noise and capable of capturing complex relationships in the data. However, its computational cost during training is higher, and it is less interpretable compared to the other models.

Each of these models has its advantages based on the nature of the data and the problem at hand. Some models, like Logistic Regression and LDA, offer efficiency and interpretability for simpler tasks, while others like QDA and Random Forest handle more complexity at the cost of higher computation. Given these differences, the choice of model will ultimately depend on the specific goal, which in our case is to select the model that achieves the highest accuracy for the classification task at hand.

Therefore, we used misclassification rates as our evaluation metric. This is calculated by dividing the number of misclassified points—the sum of false positives and false negatives—by the total number of points.

4. Model Selection

We trained the five models using all the predictors. As some of the candidate models were sensitive to the unit and the variance, we scaled all 11 numerical predictors with mean equals to 0 and variance equals to 1. We then selected the ‘k-value’ for KNN that gave the lowest training misclassification rate, which was 5. Since we accounted for all the predictors, our initial Random Forest model was a Bagging model, with *ntree* set to be 500 to stabilize the error. Additionally, we found that setting *mtry* to 7 was best as it resulted in the lowest training misclassification rate. Using 10-fold cross-validation, we obtained the following results in Table 1 and Figure 7 for the mentioned models.

Model	Training Misclassification Rate	Testing Misclassification Rate
Logistic Regression	0.745	0.254
Linear Discriminant Analysis (LDA)	0.257	0.255
Qualitative Discriminant Analysis (QDA)	0.289	0.275
K-nearest neighbors (KNN)	0.053	0.049
Random Forest (Bagging)	**0.013**	**0.018**

Table 1. Model Misclassification Rates

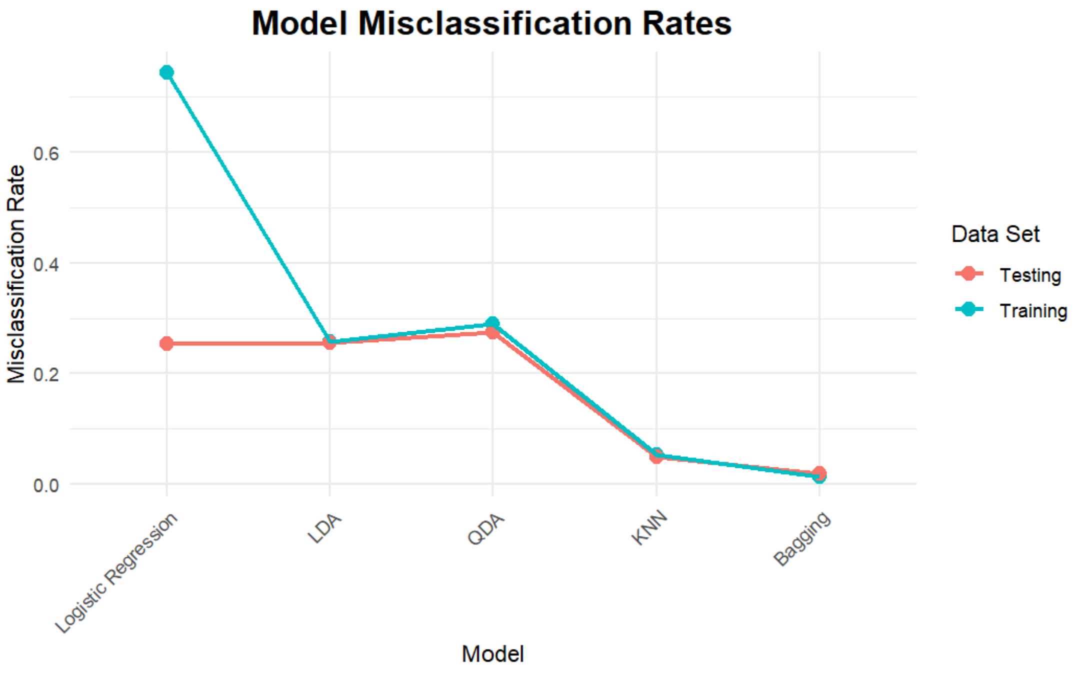

Figure 7. Model Misclassification Rates

Among the five models, Bagging had the lowest misclassification rates (0.013 for training and 0.018 for testing). Therefore, we proceeded with Random Forest to develop our final model.

5. Improving Model Performance

a. Motivation

As we observed, Bagging yielded the best results, but it is inefficient to pass all 29 predictors from the data set. Hence, the following will outline how we streamlined the complexities of the Bagging model to achieve higher accuracy by creating a simpler Random Forest model with lower *mtry* and *ntree* values.

b. Simplified Model (16 Predictors)

To eliminate variables that might be redundant or less important, we first created a Variable Importance Plot to rank all 29 predictors. From the plot in Figure 8, we were able to see three clusters of variables, each separated by a drop-off in Mean Decreasing Accuracy.

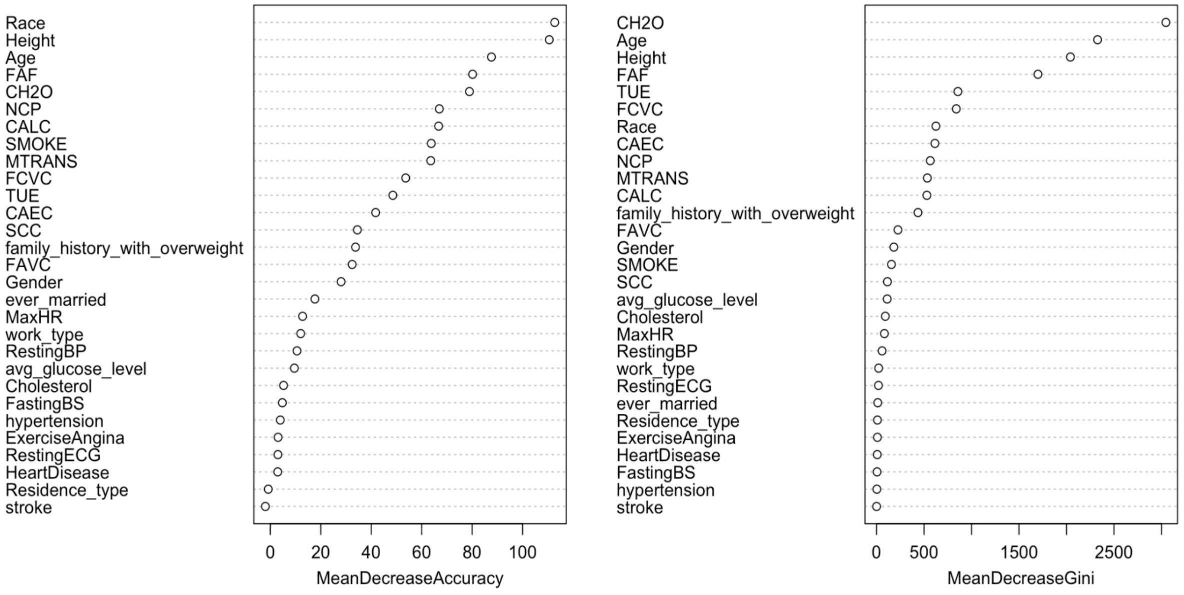

Figure 8. Variable Importance Plot (Full Model)

By taking the first 16 predictors that have at least 30 mean decrease accuracy scores, we were able to select our first subset of predictors that lower the complexities of the Bagging model and increase the model accuracy. The 16 variables selected are: ‘CH2O,’ ‘FAF,’ ‘FCVC,’ ‘NCP,’ ‘CAEC,’ ‘SCC,’ ‘FAVC,’ ‘TUE,’ ‘SMOKE,’ ‘MTRANS,’ ‘CALC,’ ‘race,’ ‘height,’ ‘age,’ ‘gender,’ and ‘family_history_with_overweight.’

After subsetting the variables, we aimed to further simplify the model by finding optimal values for the *mtry* and *ntree* hyperparameters. We trained multiple models, with *ntree* = 500 and *mtry* from 1 to 16, as there are 16 variables in the model. Upon testing these model accuracies with the training data, we found that *mtry* of 7 resulted with the highest accuracy. Following a similar process, with *mtry* = 7 and varying *ntree* values, we determined that *ntree* of 15 was most optimal. These results are visualized in Figures 9 and 10.

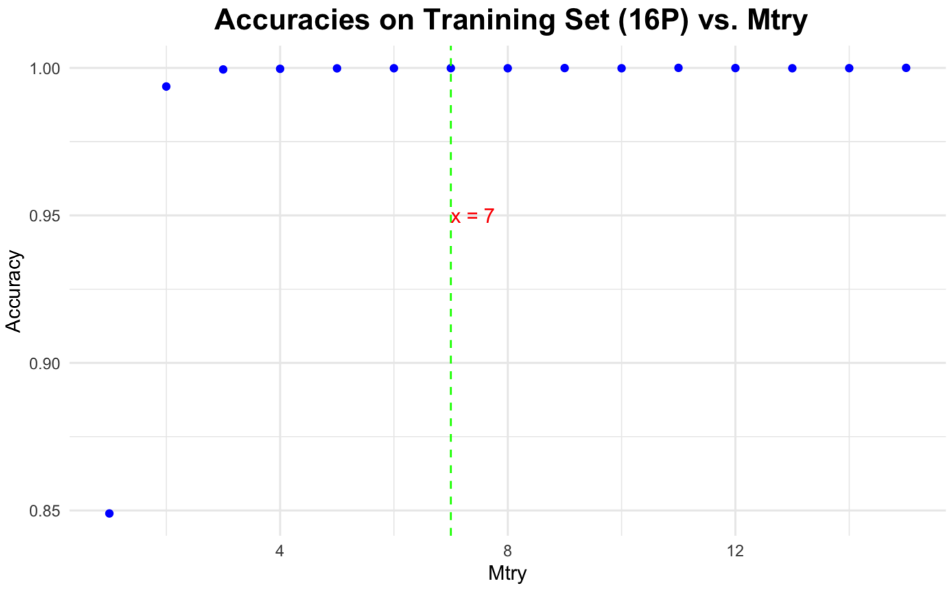

Figure 9. Accuracy of the 16P Model vs. Mtry Values

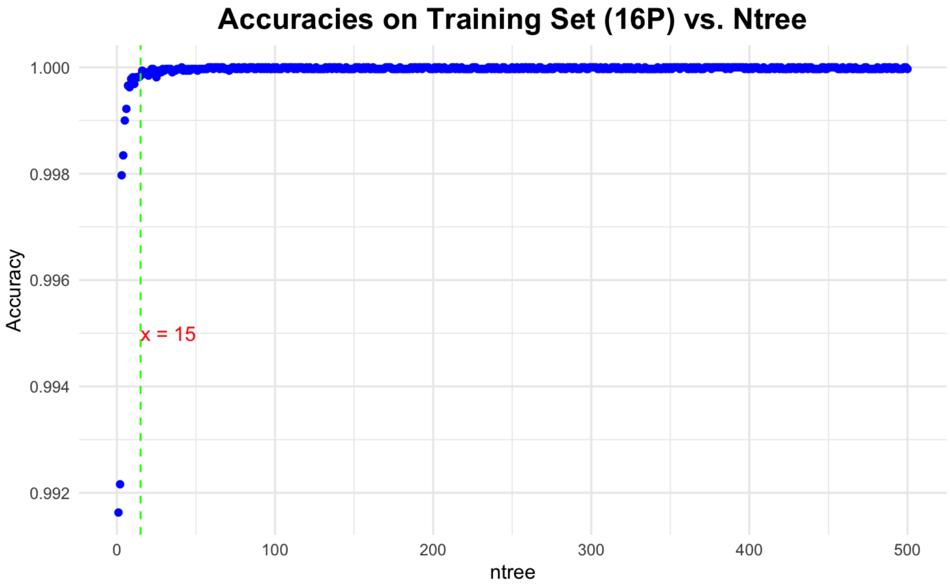

Figure 10. Accuracy of the 16P Model vs. Ntree Values

With a simpler Random Forest model consisting of 16 predictors, *mtry* set to 7, and *ntree* set to 15, we improved our accuracy on Kaggle from 0.99419 to 0.99456. Although not a sizable jump in accuracy, the improvement remained significant as it was achieved using a much simpler and more efficient model.

c. More Simplified Model (11 Predictors)

We decided to find an even simpler model with fewer predictors and gauged its performance in an attempt to improve our accuracy score. Referencing the Variable Importance Plot in Figure 11, we decided to subset only the 11 most important variables from the model of 16 predictors. The 11 variables are as follows: ‘age,’ ‘race,’ ‘height,’ ‘FAF,’ ‘CALC,’ ‘CH2O,’ ‘MTRANS,’ ‘NCP,’ ‘CAEC,’ ‘SMOKE,’ and ‘FCVC.’

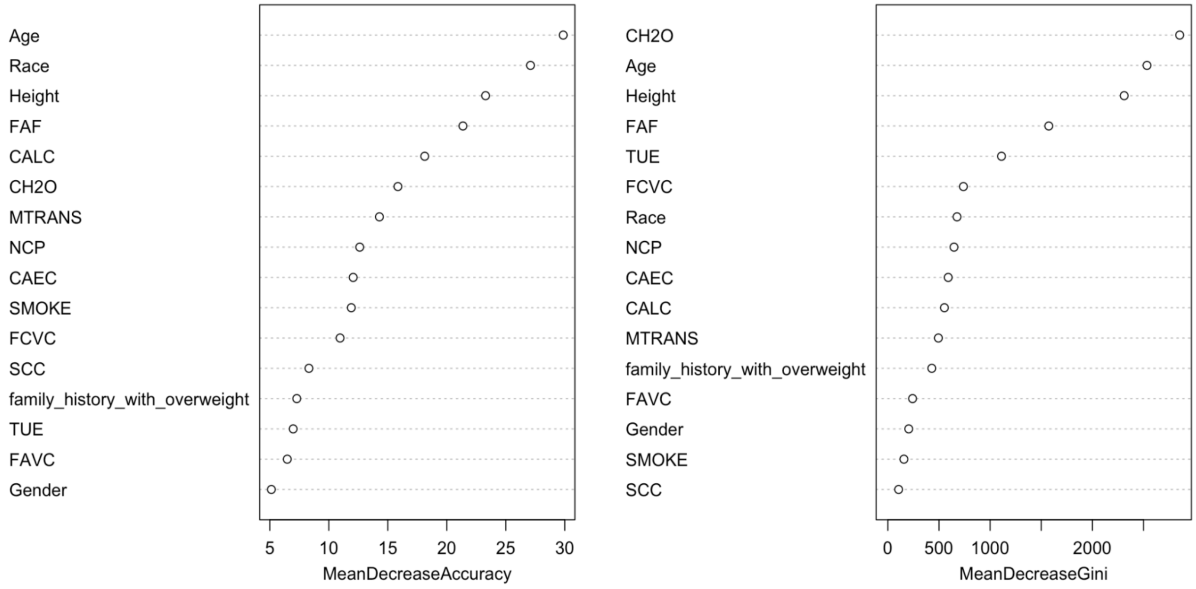

Figure 11. Variable Importance Plot of the 16P Model

Repeating the process for the 11 predictor model, we identified the ‘best’ values for *mtry* and *ntree* were again 15 and 7 respectively.

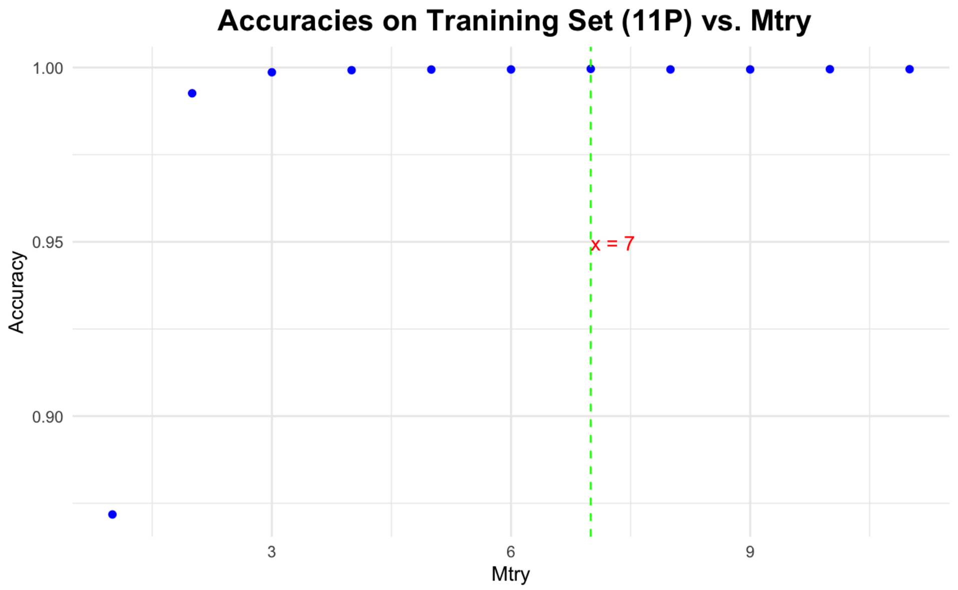

Figure 12. Accuracy of the 11P Model vs. Mtry Values

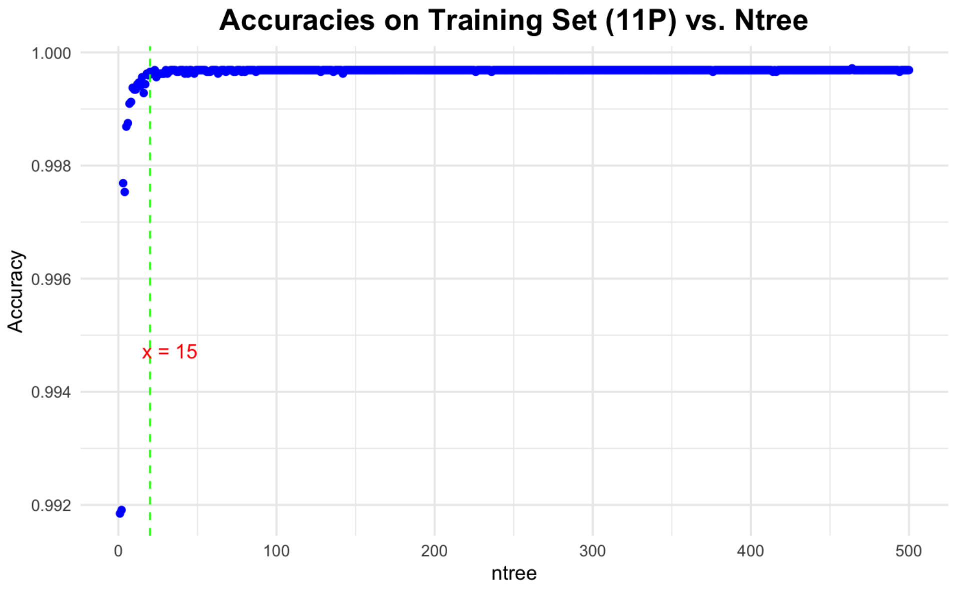

Figure 13. Accuracy of the 11P Model vs. Ntree Values

However, the 11 predictor model’s accuracy resulted in a lower value of 0.9829 on Kaggle.

d. Model of Choice

Balancing model accuracy and simplicity, we ultimately chose the Random Forest model with 16 predictors as our final model.

6. Results and Conclusion

Even though we achieved a 99.456% accuracy and ranked 11th on the Kaggle leaderboard, our model still had its limitations. Firstly, our method of data cleaning has potential downsides. Using the mean to impute numerical values can artificially reduce variance as the mean is sensitive to outliers. Similarly, the mode can significantly reduce variance and give misleading conclusions as the mode value could become heavily overrepresented giving way to increased bias. Secondly, Random Forest models are extremely complex and hence hard to interpret. The Random Forest algorithm relies on the aggregation of multiple decision trees built using random subsets of the data and features. This randomness and complexity make it difficult to interpret the specific contributions of individual predictors or understand the precise decision-making behind the model's output. To address this limitation, combining Random Forest with simpler, more interpretable models like linear regression can provide additional insights into the relationships between variables, offering a complementary perspective while maintaining strong predictive performance. Finally, our selection processes as well as our model itself rely heavily on randomness, and our results differ depending on the seed value set. This seed dependency introduces a limitation because it means that the reproducibility and consistency of our results are dictated by the chosen seed, and as such can lead to variability in model performance and outcomes if a different seed was used.

From this project, we learned how the many aspects that go into modeling are necessary and differ in importance depending on the data set. Since the dataset for this project was a mixture of categorical and numerical variables, random forest models performed well with the data. Additionally, we saw that oversimplification of a model can result in lower accuracy, but that aiming for the simplest model possible is ideal for the best results and ease of interpretability. With more time, our accuracy could have been more refined with different attempts at imputations, variable selections, and tunings of the model hyperparameters.

7. Reference

Almohalwas, Akram. *Predicting Obesity Status*. Kaggle, *n.d.*, https://www.kaggle.com/competitions/predicting-obesity-status. Accessed 30 Nov. 2024.