Stats 101C Final Project

Predicting Obesity Status

Raymond Durr, Emma Vidal, Nicholas Cassol-Pawson, Albert Carreno, and Vinuk Ekanayake

Table of Contents

Abstract

We were given a dataset of various characteristics of people associated with their obesity status and were asked to predict the obesity status of observations from a second dataset of the same characteristics without their obesity status. Both of these datasets were missing about 8% of their observations, so we imputed the data using three different algorithms. We then ran several machine learning algorithms on the imputed data, finding that we could get the highest predictive accuracy using a random forest algorithm with 5 trees and subsets of the data using 5 of the top 14 predictors. We also note that a logistic regression model with the top 12 predictors has decent predictive accuracy while being highly interpretable. Finally, we found that imputation using single chained equations worked better than the alternative methods for the obesity data.

1. Introduction

Obesity is a multifaceted condition linked to a wide array of health issues, including type 2 diabetes, cardiovascular diseases, certain cancers, and reduced quality of life. Obesity is defined as an excessive accumulation of body fat that poses a risk to health[1], and is typically classified using the body mass index (BMI). A BMI of 30 or above is considered obese.

$$BMI = \frac{Weight\ (in\ kilograms)}{Height\ (in\ meters)^2}$$

The growing prevalence and risks of obesity underscore the urgency to understand the factors that contribute to its rise so that we can find ways to address the root causes of this public health challenge.

We are working with obesity data that includes 29 variables related to an individual's health, lifestyle, and family history, but does not include both height and weight so that our predictions can rely on factors that do not go directly into computing BMI. The train data also includes values of the target variable (whether an individual is obese or not). There are 32014 observations in the train data and 10672 in the test data. There are 11 numerical variables, such as age, height, and cholesterol, and 18 categorical variables, such as gender, hypertension, and

[1] Obesity. World Health Organization. https://www.who.int/health-topics/obesity#tab=tab_1

work type. Our goal is to build a classification model that accurately predicts the binary categorical variable, obesity status, in the testing data by selecting key predictors. We are also interested in building an interpretable model that can be helpful for inference tasks.

2. Data Analysis

We first performed exploratory data analysis to uncover patterns, trends, and relationships within the data and identify potential anomalies or missing values that could impact subsequent analysis or modeling.

A. Missing Values and Imputation

The table below indicates that approximately 8% of the data is missing. This is evenly distributed across all variables except for the target variable, ObStatus, in the training data, which has no missing values. Dealing with missing values is essential, as unhandled missing data can compromise model quality and lead to incorrect or misleading results.

	Train Data	**Test Data**
Missing values (count)	74272	24759
Missing values (%)	7.99%	7.99%
Observations with a missing values (%)	90.84%	90.79%

Rather than drop rows that contained missing values, which would remove more than 90% of our data, we chose to impute the missing predictor values. We conducted this imputation across the training and testing data simultaneously to ensure that the estimates of missing values were consistent across the two datasets, and then we split the imputed data sets back into the training and testing subsets. We tried three separate algorithms for imputing missing values: missForest, mice, and Amelia. The missForest method is a single imputation algorithm, which may lead to some bias in the imputed value if the missing values are not missing completely at random

(MCAR)[2]. We tried it, as we believed there to be a decent chance that the missing values were MCAR. We also ran mice and Amelia, two multiple imputation techniques, which take into account relationships between different variables in the data, which is especially useful if there are interaction effects in the non-missing values that could help model missing values[3]. All three algorithms traded computational efficiency for imputative power and could handle both categorical and continuous data.

The first technique that we tried was missForest. This was the slowest technique to run — taking about three and a half hours — but had an out-of-bag (OOB) error rate (which is an estimate of the normalized root mean square difference between the underlying true and the imputed values[4]) of 0.133, which is decently low.

The other two methods we used do not report any OOB error rates, but we can compare a metric that we created — call it the scaled imputed deviance from missForest (the SID) — defined for each variable j, where i is the index of each observation that was originally missing for the predictor j. C_j refers to the value of the SID_j for a continuous predictor and D_j for a categorical one, $\#(i)$ refers to the number of missing values for predictor j, MF_{ij} refers to the ith imputed value for predictor j in the missForest algorithm and Imp_{ij} refers to the imputed value for the other technique (either Amelia or mice):

$$C_j = \sqrt{\frac{1}{\#(i)} \sum_{k \in i} (MF_{ij} - Imp_{ij})^2} \qquad D_j = \frac{1}{\#(i)} \sum_{k \in i} (MF_{ij} \neq Imp_{ij})$$

Due to significant right skew in these SID_j values, we chose to use the median rather than the mean as a sample statistic of SID, so

$$SID = med(C_j, D_j) \, \forall j$$

With our new metric in hand, we next tried running mice. This was a more efficient technique, taking half an hour to run. Its SID was 0.21. Unfortunately, the SID provides no

[2] Hong, S., Lynn, H.S. Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction. BMC Med Res Methodol 20, 199 (2020). https://doi.org/10.1186/s12874-020-01080-1

[3] Doove, L.L., Van Buuren, S., Dusseldorp E. Recursive partitioning for missing data imputation in the presence of interaction effects. Computational Statistics & Data Analysis 72 (2014). https://doi.org/10.1016/j.csda.2013.10.025

[4] Stekhoven, D.J. Package ‘missForest’ Documentation. Comprehensive R Archive Network (2022). https://cran.r-project.org/web/packages/missForest/missForest.pdf

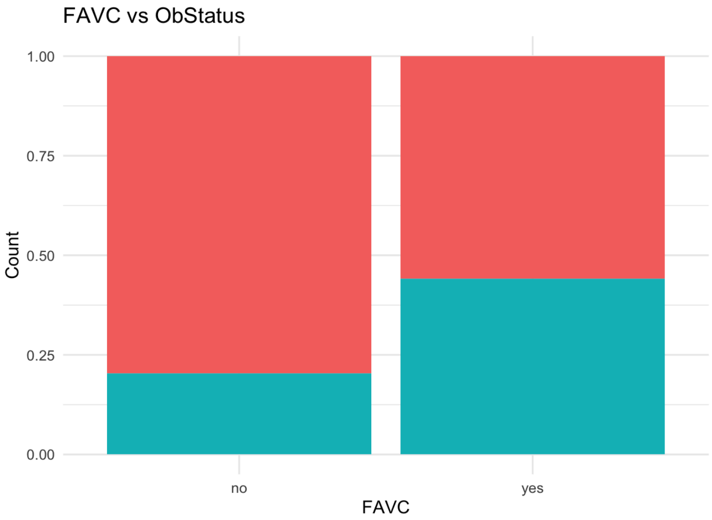
indication as to whether the OOB error rate improved or declined but can be used for now to get a sense that mice performed either somewhat better or somewhat worse than missForest — it certainly did not predict all of the values the same.

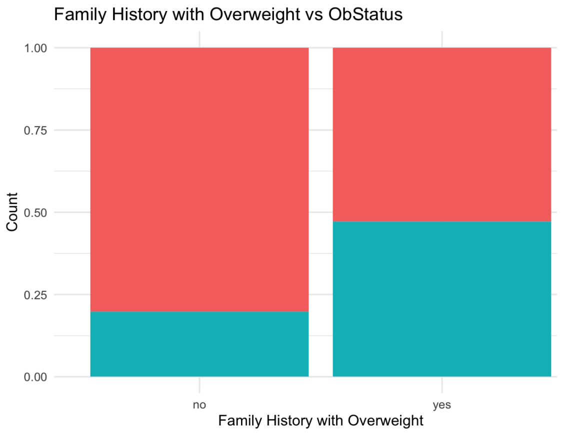
Finally, we tried Amelia. The algorithm was super quick to run, taking only about 15 seconds, and gave a SID of 0.41. This indicates a much larger deviance from missForest than mice displayed, and, combined with the speediness of the imputation, makes us doubtful that these imputed values are close to the true ones.

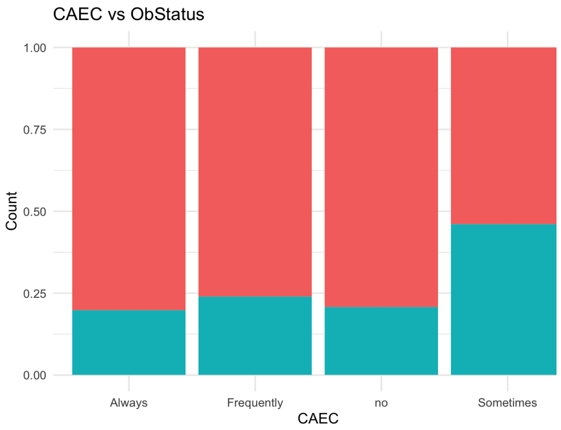
However, we had three imputed data sets and error metrics (ranging from the good — the OOB error — to the bad — the SID). Since we didn't have enough evidence to determine which dataset was the best, we chose to build all of our models with the three of them. We then would figure out the optimal imputation method for the obesity data by comparing the training and testing error metrics of our models.

B. Variable Analysis

We used subset selection on the three data sets and found the most important categorical predictors — irregular or frequent eating patterns (CAEC), a family history of overweight, and frequent consumption of high-calorie foods (FAVC) — and created stacked bar charts with obesity status subcategories.

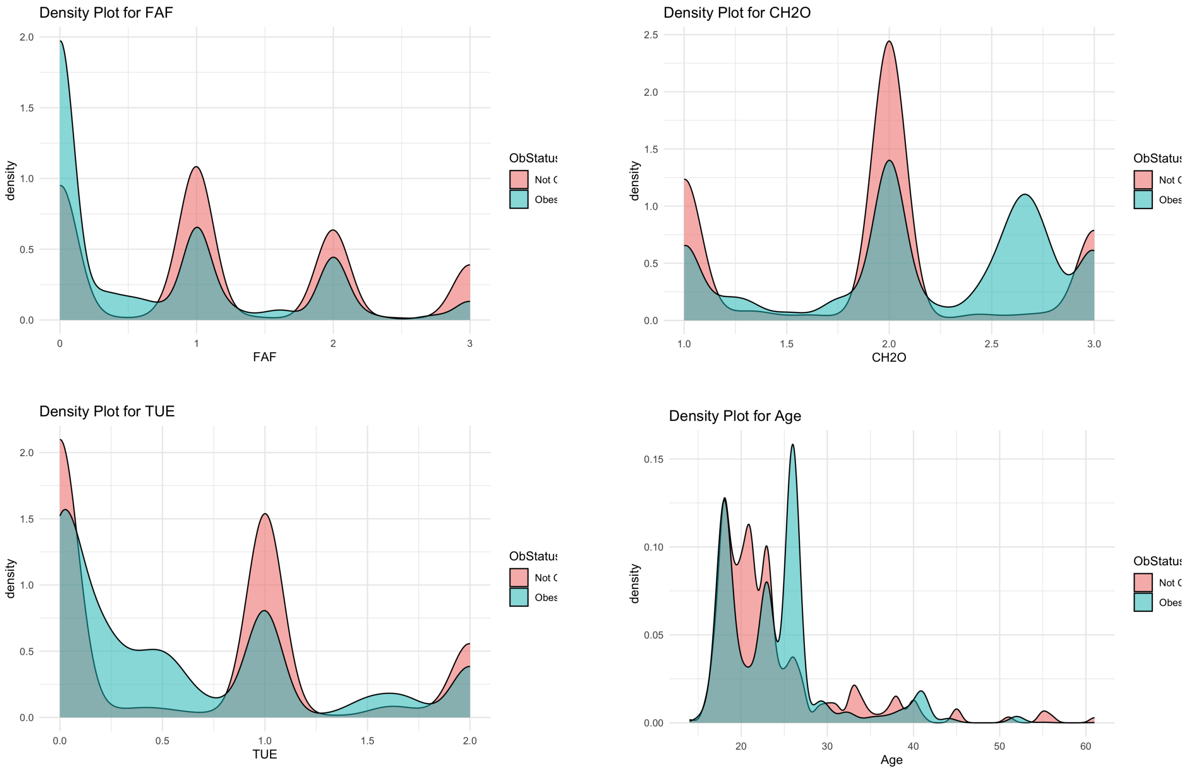






Irregular or frequent eating patterns, a family history of being overweight, and frequent consumption of high-calorie foods all are correlated with higher obesity rates. These findings highlight the importance of dietary moderation and addressing inherited or familial tendencies when managing obesity risks.

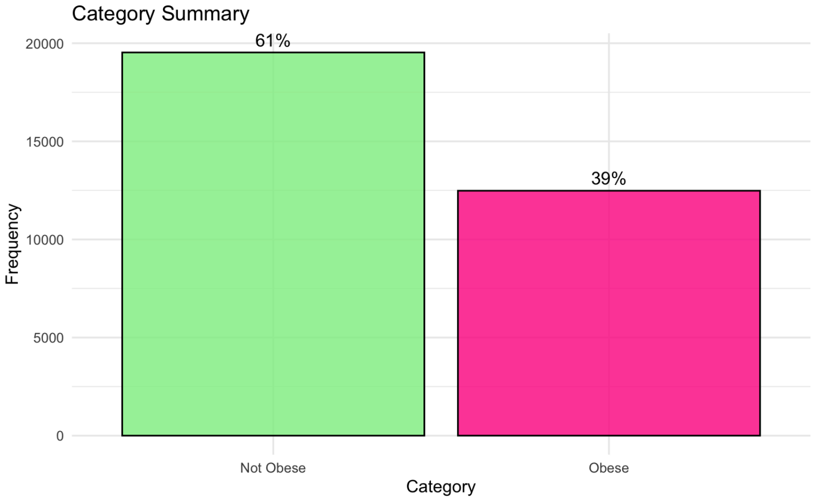
Then, we used stepwise regression to find the four most important numerical predictors: frequency of physical activity (FAF), daily water intake (CH2O), technology use time (TUE), and age. We then made density plots for them according to obesity status:



The density plots reveal significant differences between obese and non-obese individuals across these four variables. For FAF, non-obese individuals exhibit higher densities at moderate activity levels (1–2 sessions per week), while obese individuals show a higher density at 0, indicating that a lack of physical activity correlates with obesity. Similarly, CH2O shows that non-obese individuals have higher densities around 2 liters of daily water intake, while obese individuals are more concentrated at lower levels (1.5 liters), suggesting moderate hydration is associated with healthier weight. In terms of TUE, obese individuals show higher densities at very low technology use durations (0–1 hours), possibly reflecting sedentary behavior patterns. Lastly, age distribution shows obesity being more prevalent among younger individuals, with differences tapering off beyond 40 years. These trends suggest that lifestyle factors like physical activity, hydration, and sedentary behavior significantly influence obesity, with younger populations appearing more affected. Further statistical analysis is needed to quantify these relationships.

C. Maximum Allowable Error Rate

Determining the maximum allowable error rate is important to ensure that model performance is better than random guessing or naive predictors.



The bar chart above represents the frequency distribution of the two categories of the target variable in the training data. A model that naively predicts the majority class (Not Obese) will achieve a baseline accuracy of 0.61. Therefore, the maximum allowable error rate for our models is 0.39 to show that they are indeed learning meaningful patterns and not just relying on class imbalance.

3. Methods and Models

Before creating our predictive models we decided to split the training data set into a second training and testing subset, to get estimates of testing misclassification rates (MCR) without having to submit our predictions to Kaggle. We then tried a collection of models, attempting to balance both predictive power, complexity, and marginal effect inference. These models are Random Forest/Decision Trees, Logistic Regression, KNN, and QDA/LDA on the principal components of the data.

As our data set contains factor variables, one-hot-encoding was performed for KNN in order to calculate the ‘distance’ between two data points. It is noted that this is not a perfect solution

— as distances between factors are meaningless, however, this is a valid workaround to include factors into the calculation.

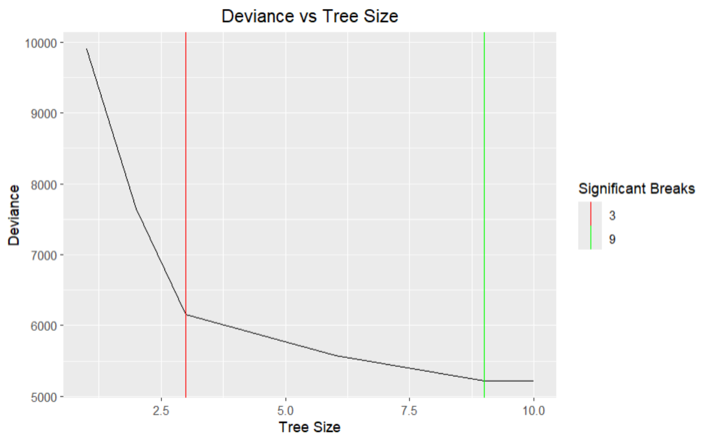
Please note that, while we did test the data using all three imputed data sets, missForest performed exceptionally better than the rest in all models. Our analysis therefore will focus on the missForest version of these data sets. We have, however, included asides to note interesting differences between the data sets.

A. Decision Trees and Random Forest

Decision Trees and Random Forests are somewhat similar, with Random Forests being able to be conceptualized as multiple Decision Trees based on a number of randomly selected columns of the original data set. Both Decision Trees and Random Forests are powerful in that they do not assume a functional form of our data distribution or the relationship between explanatory and response variables. This makes them highly flexible. We therefore decided to try out Decision Trees first, then move on to Random Forests.

a. Decision Trees

We constructed multiple different decision trees, varying their size and compared their deviances.



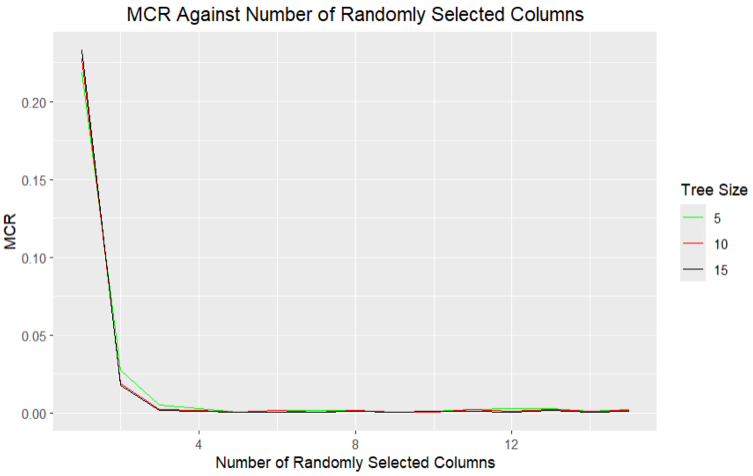
As we can see, there are two significant breaks at a tree size of 3 and 9[5]. While the tree with 3 terminal nodes saw lower marginal decreases in deviances than those using 1-2 terminal nodes, the deviance of the tree with 3 terminal nodes was significantly higher than with 9, so we continued our analysis with the 9-node decision tree. This decision tree only used 6 predictors: CH2O, age, height, CAEC, FAF, and CALC.

[5] Aside: The break at 9 is visible for all datasets except for mice. The full decision tree for the mice imputation actually only includes 8 terminal nodes.

After testing the model, we obtained an MCR for the training data set of 0.21 and for the testing set an MCR of 0.20. This was below our maximum allowable MCR, but was not great. We therefore moved on to Random Forests.

b. Random Forest

As previously stated, Random Forests can be considered a collection of decision trees created using a random subset of columns from the original data set. They therefore have two main hyperparameters: the number of trees and the number of randomly selected variables[6].



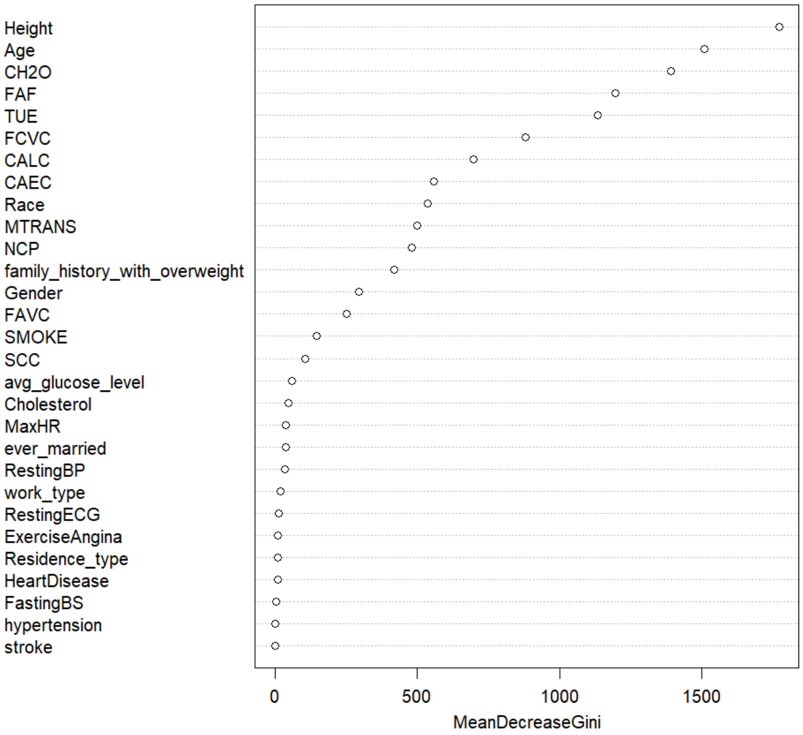
We tested the number of randomly selected columns against the MCR for forests with varying numbers of trees[7]. All of the tree sizes tended to perform fairly similarly, with forests that built trees using fewer than 5 randomly selected columns performing noticeably worse than those with at least 5 columns, regardless of the number of decision trees in the forest. In fact, for all tree sizes we saw two significant breaks: one at 3 randomly selected columns and one at 5 randomly selected columns. Since using 5 randomly selected columns allowed us to significantly reduce the number of trees created while only marginally increasing the number of columns used, we chose to build the random forest with 5 trees and 5 random columns model.

This model performs extremely well, with an MCR of 0.00078.

[6] Decision trees have a third hyperparameter: the maximum depth/number of terminal nodes within each tree. We do not, however, explore optimization along this parameter.

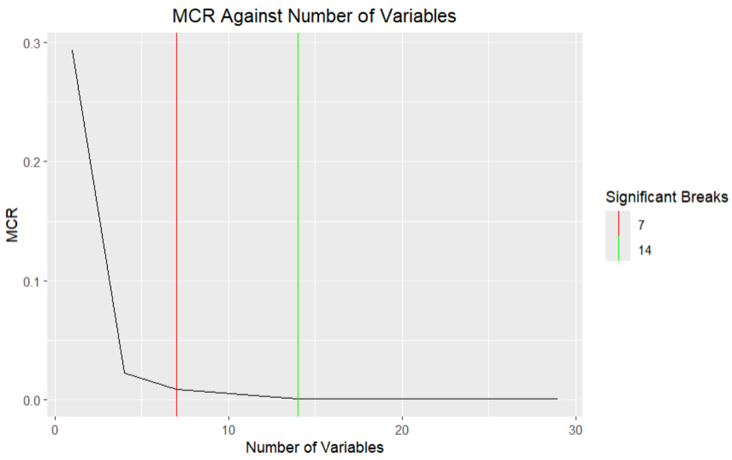
[7] We did not include all of the tested tree sizes nor all of the number of selected columns within this graph to improve legibility.

We can also identify the most important variables in this model. Here, importance is measured by the mean decrease in the Gini index for models which do not use that variable. We note that just like in the first decision tree that we built, height, age, CH2O, and FAF are among the most important predictors, further indicating that there is some meaningful relation between obesity status and these predictors. More attention could be given to people with levels of these variables that are indicative of an increased risk of obesity.



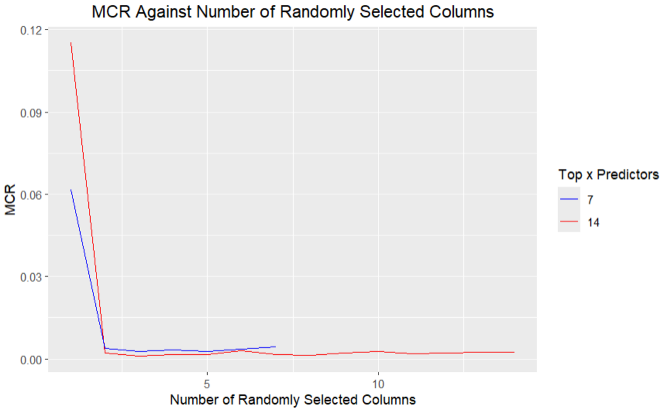
We then attempted to reduce the number of variables by selecting the 5 columns to build the model from only the top *x* important variables[8].

After plotting the MCR against the number of variables used as the selection pool, we saw two significant breaks at pools with sizes of 7 and 14 variables. After further investigation, we decided to use the model with a pool of 14 variables. While both models performed somewhat similarly, the 14-variable one had a lower MCR for the same number of randomly selected columns[9].



[8] Note that if *x* was less than 5, the model would be guaranteed to select the same column multiple times, due to the bootstrapping method used by the random forest algorithm.

[9] We decided to hold the number of trees constant at 5 trees per forest.



As we can see in the graph to the right, these models perform well even with just 2 randomly selected columns. This model has an MCR for the testing data set of 0.02811. This is about a 27.7 times increase in our MCR from using the model with all predictors. Therefore, although we were able to decrease the number of variables by a little more than half, and the number of random columns by 3, we still chose the full model with 5 trees and 5 random columns. This model performed exceptionally well and was relatively simple for a random forest.

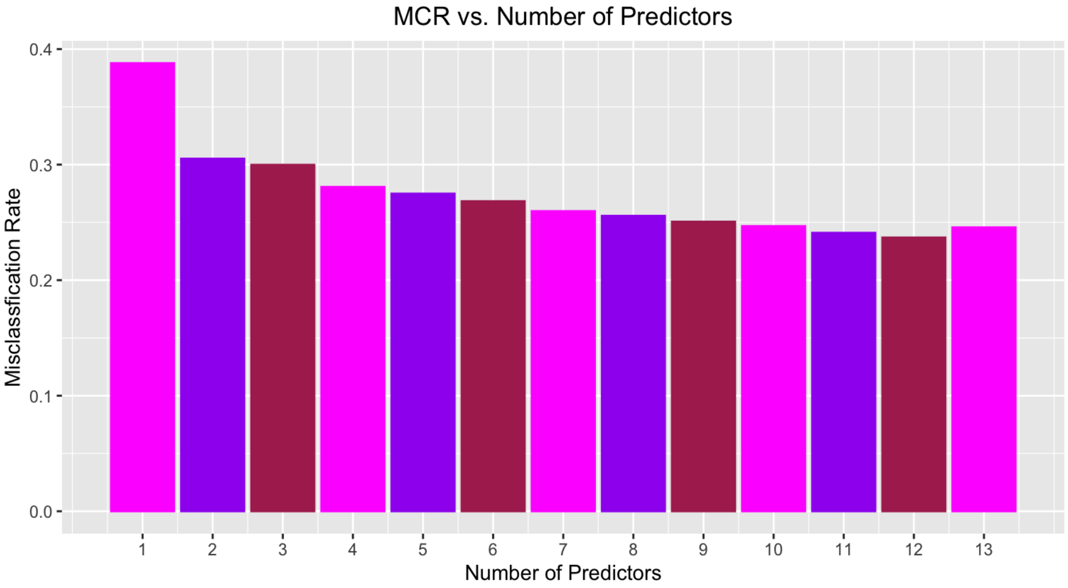
B. Logistic Regression

As part of our effort to find the best possible model for predicting obesity status, we also looked into logistic regression. This type of model is simple, highly interpretable, and computationally inexpensive. We first used the entire training dataset to fit a full model as a baseline and see how it compared to a simplified model that had properly gone through feature selection and hyperparameter tuning.

The full model had an MCR of 0.2584 but, since every predictor was used to construct this model, it was highly likely that we could achieve a lower MCR by not overfitting. To determine what predictors to include in the model, we first used backward elimination. This method takes the full model and removes a predictor from the model if removing it improves the model fit. This process is performed iteratively until removing the next "worst" predictor would not cause the model fit to decline according to some criterion. We chose to use AIC as the model fit criterion, as it punishes model complexity to a higher degree than adjusted R^2, but is not as extreme as BIC in doing so. We were able to successfully remove 9 variables with this method: "Age," "SMOKE," "TUE," "RestingBP," "MaxHR," "HeartDisease," "hypertension," "ever_married," and "Residence_type." We concluded that the inclusion of these variables caused the model to be overfit, so we took them out and moved on to the next method: the χ^2 test using the `drop1()` function in R.

By using this function and statistical test, we could see the impact of removing each of the remaining predictors on the model. If the impact on the model of removing a variable is negligible according to a χ^2 test, then we chose to remove that variable to make the model more parsimonious and interpretable. Using this method, we identified 7 more variables that were unnecessary to include in our final model because they added little to no additional information to boost our model's predictive power. These variables were "Cholesterol," "Fasting BS," "RestingECG," "ExerciseAngina," "work_type," "stroke," "avg_glucose_level." At this point, we had reduced the model to 13 predictors from the 29 we started with — a much simpler model.

However, because we are looking for the most accurate model possible, we took it one step further. We ran a 5-fold cross validation algorithm to find whether we could drop any more of the predictors from the model, looking for the subset of the 13 best predictors that would minimize the MCR. We generated a logistic regression model for each of the 5 folds of the training data using the first *x* important predictors and computed the average MCR across the 5 folds. The 1-predictor model included the most important predictor as determined by the `varImp()` function, which ranks a model's predictors by importance or their overall contribution to model accuracy, and the 13-variable model added the least important predictor to the 12 most important ones. We visualized the results with the bar chart below:



As we can see, though it is close, the 12-predictor model had the lowest cross-validated MCR for predicting obesity status using the missForest test data at 0.2369. This rate might only be a roughly 2 percentage point reduction from the full model, but any MCR reduction makes a difference, especially for large datasets. Given this fact, we decided to make the 12-predictor model our final model. After submitting our test predictions from this model on Kaggle, we found it had an MCR of 0.2389.

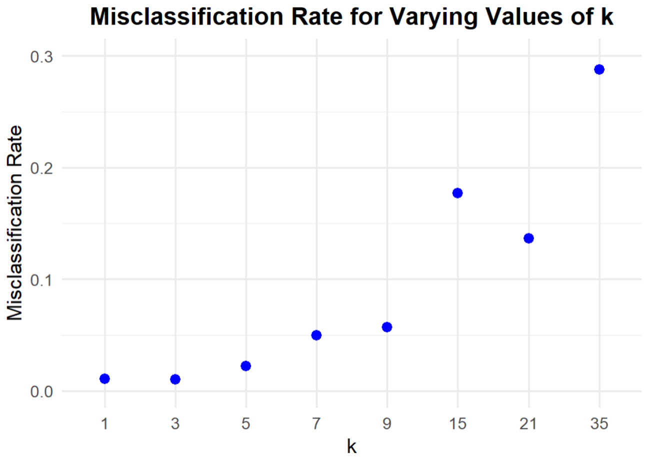
In summary, our final model does end up outperforming the full logistic regression model in terms of predictive performance showing that it was worthwhile to eliminate unnecessary variables and isolate the most important ones to create a simpler model. Our top five most important variables in predicting obesity status using logistic regression end up being "FAF, "family_history_with_overweight," "CH20," "Height," and "FAVC."

C. KNN

In the exploratory data analysis, it was clear that the relationship between the predictors and the obesity rate was highly non-linear. This notion was reinforced with logistic regression's relatively low MCR. As such, it was clear that a non-parametric machine learning technique could potentially address this specific data set.

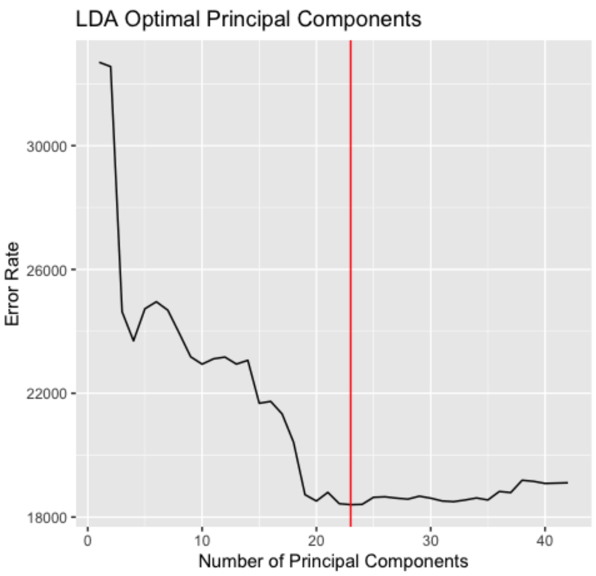
K Nearest Neighbours (KNN) is the most straightforward non-parametric model, as for any given test data point, it directly predicts the "closest" training data point to be the output of this test data. This model can be generalized by rather than selecting the "closest" 1 training data point to be the prediction of the test data, it selects the "closest" k training data points and conducts a majority vote to make its final prediction.

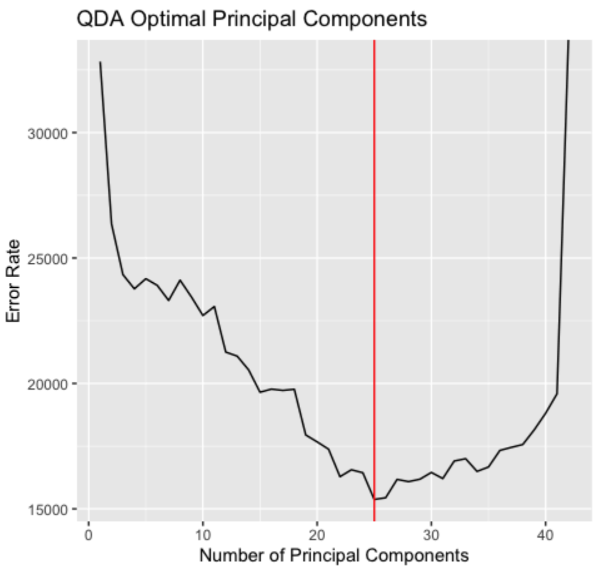
As shown in the graph to the right, the optimal value of k is 3 — with $k = 1$ following very closely. It is important to note that on average, as k increases, the test accuracy decreases. And since lower values of k indicate a more complex model (closely follows the training data patterns), this validates our hypothesis that a more complex model is required to accurately predict on this data set.



D. LDA-PC and QDA-PC

Finally, noting that the data was highly non-linear, we separated the three imputed data sets data into their principal components[10]. We then ran both LDA and QDA on these principal components (we will denote these models, respectively, as LDA-PC and QDA-PC), as we realized that the data did not display a strong linear relationship, even among the principal components, but may display in some high dimension a decision boundary that could discriminate between the obesity classes. Recognizing that using all 43 components would lead to a highly complex model that would likely overfit the training data, we chose instead to run a 10-fold cross-validation algorithm on all three imputed training data sets. We created a 90-10 split on the training data, then trained a model using the first through lth principal components, where $l \in \{1, 2, \cdots, 43\}$. We repeated this process 10 times for each dataset in both the LDA-PC and QDA-PC algorithms, choosing a value of l such that the MCR, penalized by a function of the number of predictors similar to the BIC penalty, was minimized on the test subset of the data. We then chose the optimal number of principal components based on this error metric averaged across the 10 folds. The following are plots of the error metric across the number of principal components for both LDA and QDA on the missForest-imputed dataset, where it had an optimal value of 23 for the missForest LDA-PC and 25 for the missForest QDA-PC.





[10] There were 43 principal components in total, as we had to create k-1 dummy variables for each of the categorical variables with k > 2 levels.

Error Rates for Optimal Components			
	Amelia	**MissForest**	**Mice**
LDA	19501.07	18394.45	19253.67
QDA	16873.89	15370.45	16633.59

Next, we looked at the average value of the metric across all 3 imputed data sets. We found that it was lowest on missForest in both the LDA-PC and QDA-PC models, and that the worst QDA-PC model was still better than the best LDA-PC models. So, we chose to submit as our final predictions from the PC-discriminant-analysis-model family those from QDA-PC on the missForest-imputed testing dataset, which had an MCR of 0.216.

The final accuracy that we got from this model is not horrible, it is certainly permissible under our maximum allowable error rate, but it is blown out of the water by the accuracy rates that we saw using other methods. Additionally, the model that we used was not very interpretable, as principal components already obscure the effects of individual variables, an effect that is compounded by using 25 principal components and then running QDA, a highly opaque algorithm. The performance of the QDA-PC model is clearly not good enough to justify its use for the data sets.

4. Model Comparison

We now wish to compare the performance of the four models that we built. We will compare them based on their complexity, interpretability, and predictive accuracy on the testing data set.

Comparison of Models across Metrics

	Complexity (Low is Good)	Interpretability (High is Good)	MCR (Small is Good)
Random Forest	Moderate	Moderate	0.00014
Logistic Regression	Low	High	0.24885
KNN	High	Low	0.01012
QDA-PC	Moderate	Low	0.2158

The simplest model that we built was logistic regression — the parameters that it outputs have a highly interpretable relationship with the response variable, where a one unit increase in the predictor associated with a coefficient leads to a respective percentage increase in the odds of the probability of the observation being not obese to the probability of it being obese. However, this model also had the highest predictive MCR of 0.23885.

The next simplest model was random forest, which only used 5 trees with 5 predictors each, so it had a low level of complexity. This makes the model quite interpretable as well, as we can walk each observation down the 5 trees and see how changing a specific variable alters its outcome. However, there is no simple way that we can talk about a specific variable's impact on the response in general terms, so the interpretability is not quite as good as the logistic regression. This model also had the lowest predictive MCR of 0.00014, which is extremely good.

The QDA-PC model is more complex than the random forest model, using all of the variables across 25 principal components. This is still less complex than KNN, which used all of the variables to create a highly flexible model, but more so than the 5-tree model seen in the random forest. Interpretability of the QDA-PC model is also close to non-existent, as the dimensionality reduction from the PCA selection keeps all of the variables in complex weightings and then adds the close-to-black-box-edness of QDA on top of this. Additionally, the predictive MCR on the test data of 0.2158 is only slightly better than logistic regression, at the expense of becoming much less interpretable, so across all of these metrics, logistic regression is a better model than QDA-PC.

Finally, we have KNN. This model is non-parametric, which causes it to be uninterpretable, as it is extremely difficult to determine which predictors significantly affect the outcome variable. However, the model had an MCR of 0.01012, which is very good, but is overshadowed by the much more interpretable random forest.

So we are left with 2 final candidate models — the easily interpretable yet quite inaccurate logistic regression or the highly accurate but much less interpretable random forest.

5. Conclusion

Of all of the models that we tried, the logistic regression and random forest models outperformed the others in different ways. We recommend using the logistic regression model for getting a sense of the marginal effects of different predictors and the random forest one for getting a highly accurate prediction of the data. For this project, where the goal is to accurately predict the test data set, the obvious best model is random forest. It had an MCR of close to 0, and was extremely parsimonious for a random forest model. However, it was difficult to use this model to analyze the effects of different predictors on the outcome variable. To have this more interpretable result, we would recommend using logistic regression, which gives a sense of the specific characteristics that lead to people being obese or not.

Finally a note on the imputed data sets. As we indicated throughout the project, the missForest-algorithm imputed data set led to the highest accuracy of all of the imputed datasets on all of the models. This indicates that it was probably the best imputation method for this data, even though the single imputation technique that it implements is not as good for certain datasets. This confirms that our assumption of the missing values being MCAR was well-founded and that the processing time to impute was worth it.