STA363Proj3 - Formal Report

Rickey huang

4/19/2021

Abstract/ Executive Summary

This report explores the relationship among the nutrient contents in the servings of McDonalds carefully. The relationship among other variables and the variable *"Calories"* is explored using the regression tree model called Tree1. The relationship among features with the variable *"Categories"* with levels Beef & Pork, Breakfast, Chicken & Fish, and Desserts is explored using the classification tree model called Tree2. The visualizations for the both tree models are created in the following report. Also, the Bagged Regression Forest and Random Regression Forest models are created to predict the *"Calories"* in the McDonalds servings. There are two most important variables, which are *"Carbohydrates"* and *"Total.Fat"*. Their partial dependence are also explored in the report. The Bagged Regression Forest model are chosen to do the prediction, since it has a least test RMSE, which is only 31.20229.

Section 1: Data Cleaning

In this section, the correlations between data and the meanings of the data themselves are explored. Some dulicaptive variables for the analysis is deleted.

Section 1.1: Refining the Variables

By computing the correlation matrix of the numeric variable in the data, the correlation between every pair of numeric variables in this data is explored. There are some highly correlated variables. The correlation between the variables *"Calories.from.Fat"* and *"Total.Fat"* is 0.99957642, and the possible reason for this strong correlation is the fact that one gram of fat normally contained 9 calories, which is confirmed by the data. Hence, the *"Calories.from.Fat"* is removed from he data set. However, since the variable *"Calories"* is the number of calories in per serving, it should be kept in the data set because it measures the total calories in the serving but not only the calories from the fats of the serving, which means it is different from the variable *"Calories.from.Fat"*. The correlation between the variables *"Total.Fat"* and *"Total.Fat....Daily.Value."* is 0.99970351 for the reason that the *"Total.Fat....Daily.Value."* is measure by divide the *"Total.Fat"* by the total recommended of fat and then times 100. For this reason, the variable *"Total.Fat....Daily.Value."* can be deleted from the data set. For the similar reason, the daily values *"Saturated.Fat....Daily.Value."*, *"Cholesteril....Daily.Value."*, *"Sodium....Daily.Value"*, *"Carbohydrate....Daily.Value."*, and *"Dietary.Fiber....Daily.Value."* should be removed since they have correlations of 0.9992613, 0.99985282, 0.999919583, 0.99961372, and 0.98592990 with their corresponding actual values.

Finally, since the *"Item"* variable is a unique variable that represents the name of each serving at McDonalds, it could be used as an identifier for the each observation, but it is not useful for exploring the relationship with *"Calories"*, so that it would be removed from the data set in order to do the further exploration.

After cleaning and refining the data set, we get a data with 259 observations and 16 variables. Among those variables, only one of them is the categorical variable, which is *"Category"*, while the rest of variables are numeric variables that measures the size and contents in the servings from McDonalds.

Section 2: Modeling Calories

Section 2.1: Distribution of the Response Variable

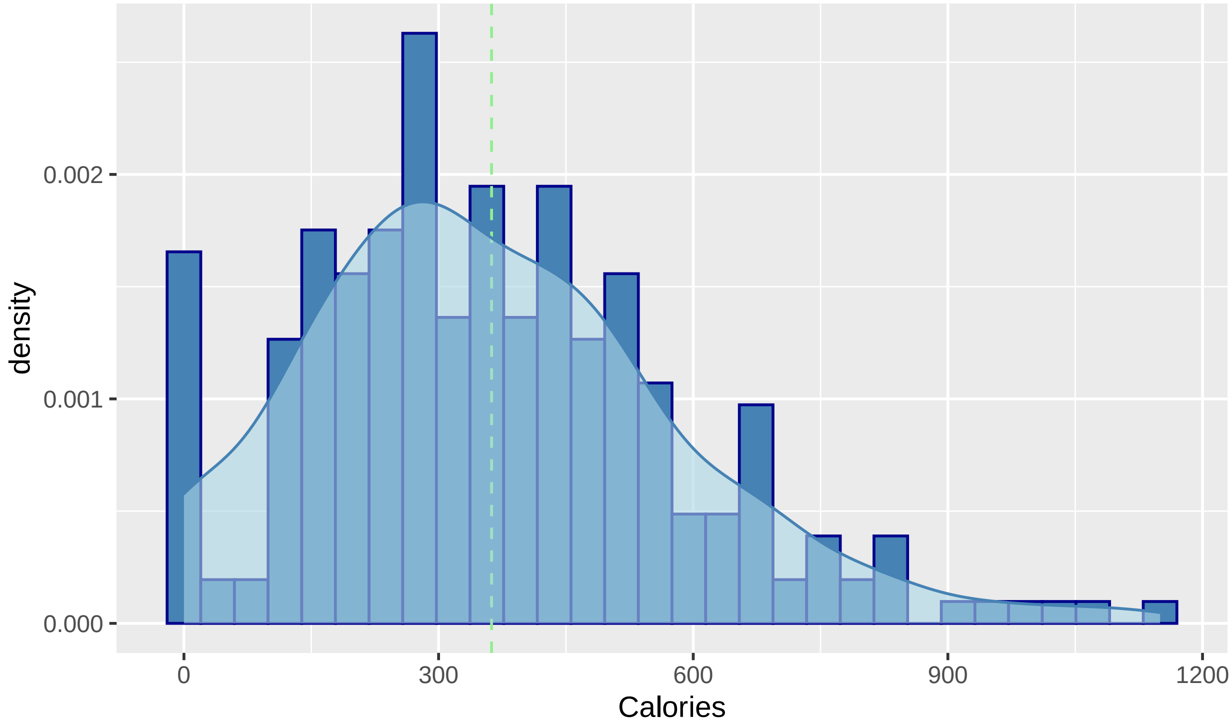

Figure 1: Histogram for Calories

In order to show the trend in the response variable *"Calories"*, a histogram for it is created as shown in the Figure 1. From the histogram, the mean 362.4324 is indicated by the dashed light blue vertical line in the visualization. By the relative relation between the mean and the median reveal from the histogram and the density curve in the histogram, the distribution of the Calories should be identified as skewed to the right.

Section 2.2: the Tree1 Model

Section 2.2.1: Reasons for Choosing the Regression Tree Model

Since the client wants to understand the relationship between different features and the amount of Calories in the food and also wants a clear explanation, a regression tree model would be a good choice for our client to use to create a great visualization to explore and show the relationship. Since our response variable *"Calories"* is a numerical variable, we could use the regression tree to do show the numerical relationship between the response variable and features.

Section 2.2.2: Training the Tree Model Using All Features (fullTree1)

First, all features are included to fit the tree model fullTree1. The fullTree1 model has 8 splits and the Root Node Error (RNE) for the fullTree1 is 48872. In order to evaluate how this tree model performs by creating each split, the cp table for the fullTree1 model is computed as shown in the Table 1. From the cp table, the percentage change from the RNE (xerror in the table) by creating each split is shown row by row. When the 8^{th} split is created, the percentage change from the RNE for the test MSE is 16.34231%, so that according to the Formula 1, the test MSE for the fullTree1 is $48872 \times 16.34231\% = 7986.814$.

$$test\ MSE = RNE \times xerror \tag{1}$$

Table 1: The cp table for the fullTree1

CP	nsplit	rel error	xerror	xstd
0.5988049	0	1.0000000	1.0039544	0.1013384
0.1312561	1	0.4011951	0.4271343	0.0514122
0.0761375	2	0.2699390	0.3635852	0.0454102
0.0444821	3	0.1938015	0.2379931	0.0256588
0.0215682	4	0.1493194	0.2196494	0.0212518
0.0192588	5	0.1277512	0.1965999	0.0206247
0.0116562	6	0.1084923	0.1800554	0.0205555
0.0111753	7	0.0968361	0.1650205	0.0197030
0.0100000	8	0.0856608	0.1634231	0.0196925

Section 2.2.3: Methodology for Growing and Pruning the Regression Tree

Each split in the fullTree1 model is created by minimizing the RSS of the model. However, the fullTree1 may overfit on some points which would results an inaccurate clarification for the relationship. In order to avoid this overfitting problem that may exist in the fullTree1 model, the cost complexity pruning technique is used to prune the fullTree model to produce a more appropriate tree model. In stead of minimizing the RSS, the cost complexity metric (C_α) is minimized. The formula for C_α is expressed in the Formula 2, where $\alpha\,|T|$ is the penalty term add to the RSS to do the pruning, α is the tuning parameter, and T is the number of leaves in the tree model.

$$C_\alpha = RSS + \alpha\,|T| \tag{2}$$

Section 2.2.4: Pruning the fullTree1 to create the Tree1 Model

The cp table is used to choose a convincing tuning parameter by looking one with a smallest test MSE. Also, for the convenience for explanation, we want a model with as least splits as possible. Using this rule, and from the Table 1, a 7-split tree model should be chosen since one more split after the 7-th split only result in a 0.002 reduction in xerror, which would not bring us much new information about the relationship with *"Calories."*. The percentage change from the RNE for the 7-th split is 0.1650205, which would produce a test MSE of 48872 × 16.50205% = 8064.882. Then the tuning parameter chosen is 0.0111753. Also, the cp plot in the Figure 2 that visualizes the relationship between the percent change from RNE and cp values also confirms this choice of α.

With $\alpha = 0.0111753$, the Tree1 model could be fitted. The visualization of the regression tree model Tree1 is shown in the Figure 3.

Section 2.2.4: Information revealed by the Tree1 Model

From the visualization in the Figure 3, each split is decided by minimizing the cost complexity metric C_α. For example, the reason that the *"Total.Fat"* is used as the first split and it splits at 14 is that comparing to using other features to make the first split or splitting at other values, the first split on *"Total.Fat"* at 14 would result an optimal cost complexity metric. Similarly, the further splits are made in this way as well.

The visualization also uses the colors to indicate the amount of calories of the servings. The higher calories the serving are, the color for that leaf would be darker. Hence, a high value of *"Total.Fat"* and *"Carbohydrates"* would result in a relatively high calorie serving, while a low value of *"Total.Fat"*, *"Carbohydrates"*, and *"Sodium"* is more likely to be related with low calorie foods. The detailed quantified splits using combinations of these four features can be explored in the visualization. For example, if a serving has *"Total.Fat"* content of 15 and *"Carbohydrates"* content of 93, the prediction for the calories in this food would be 530, which is a relatively high calories.

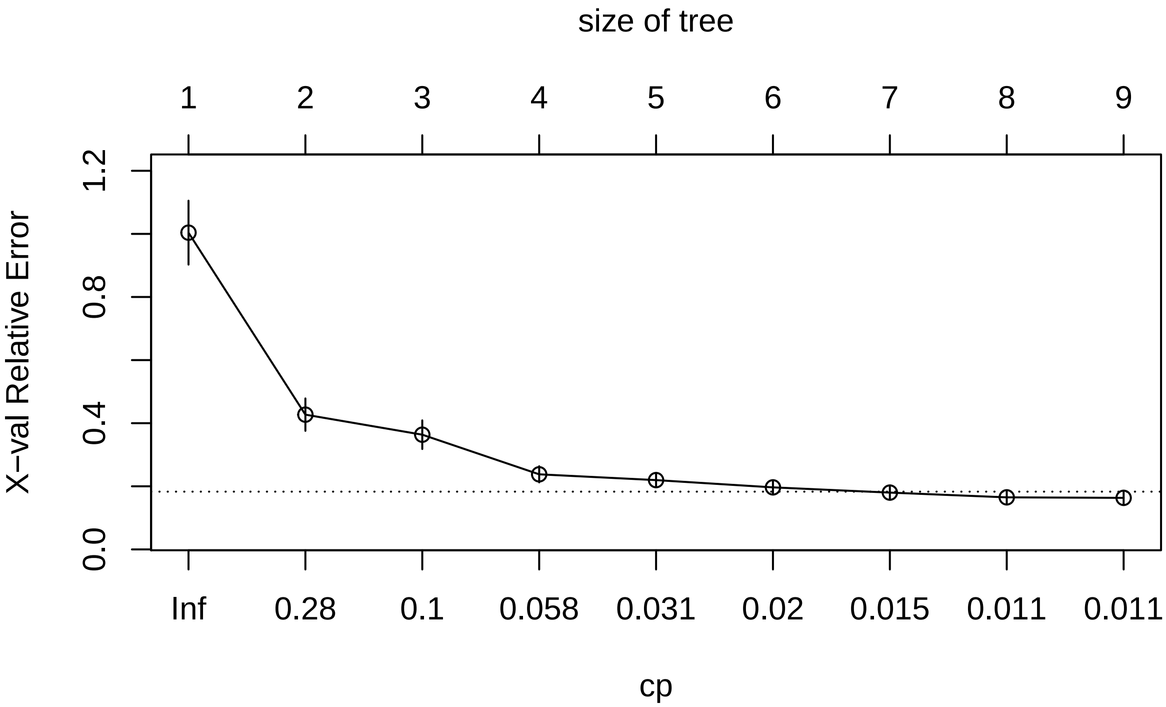

Figure 2: cp plot for fullTree1

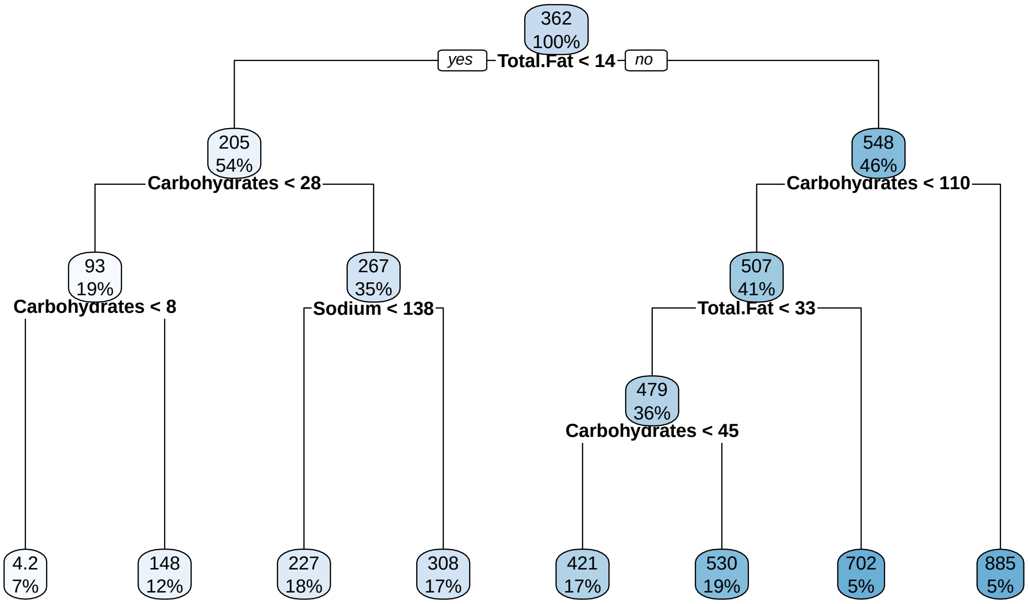

Figure 3: Visualization for Tree1

Section 2.2.5: Evaluating the Tree1 Model

In order to evaluate the accuracy of this tree model, the training and test RMSE are calculated. The training RMSE is 68.79363, while the test RMSE could be calculated by taking the square root of the test MSE that we obtained before using the Formula 1. Hence the test RMSE is $\sqrt{8064.882} = 89.80469$.

Section 3: Modeling Category

Section 3.1: Distribution of the Response Variable

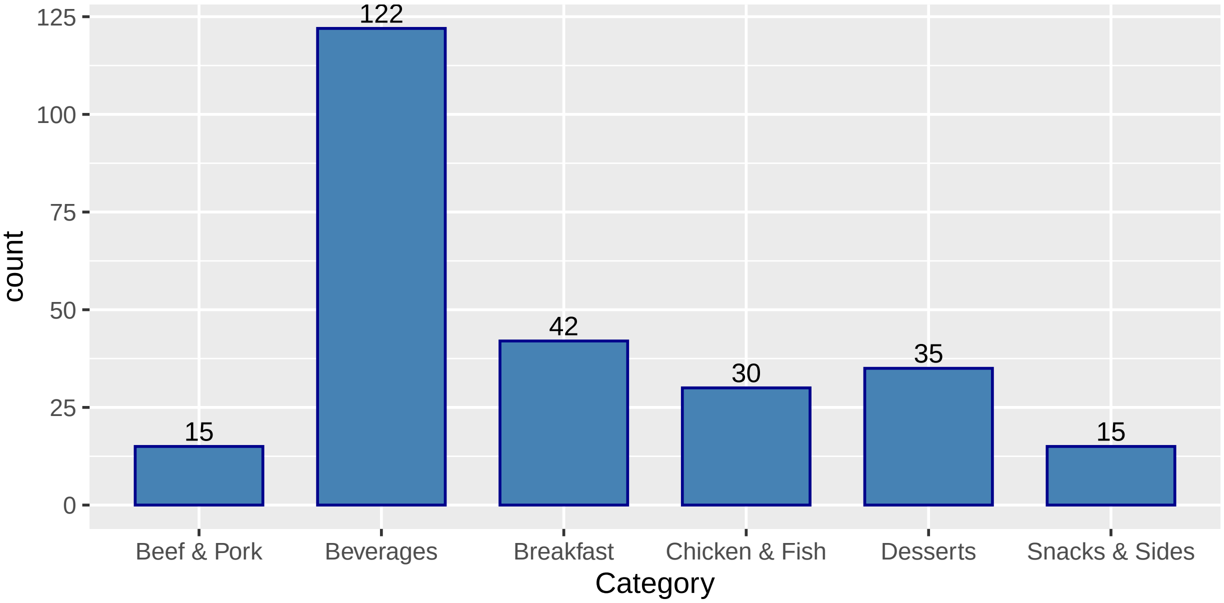

Figure 4: the Distribution of the Responce Variable Category

The distribution of the *"category"* variable is shown in the Figure 4. Since the level of Beverages has a count of 122 which is much more than other levels, so the choice I made is to only include levels of Beef & Pork, Breakfast, Chicken & Fish, and Desserts in the response variable.

Section 3.2: the Tree2 Model

Section 3.2.1: Reasons for Choosing the Classification Tree Model

As the reason for the Tree1 model, a tree model is a good fit for the client's need. However, since the response variable this time is a categorical variable, the classification tree instead of the regression tree is considered to model the relationship between the *"Category"* and other features.

Section 3.2.2: Training the Tree Model Using All Features (fullTree2)

As the tree growing process in the first tree model, all features are first tried to fit a classification tree that names fullTree2. This full model has 5 splits, and the Root Node Error for fullTree2 is 0.65574, and the RNE for the classification model represents the training Classification Error Rate (CER). In order to evaluate how the fullTree2 performs, the test CER for the model could be calculated by using the formula 3. As revealed from the cp table computed in the Table 2, the percentage change from the RNE for the 5^{th} split (xerror in the table) is 0.3875. Hence the test CER for fullTree2 is $0.65574 \times 38.75\% = 0.2540992$.

$$test\ CER = RNE \times xerror \tag{3}$$

Table 2: The cp table for the fullTree2

CP	nsplit	rel error	xerror	xstd
0.3875000	0	1.0000	1.0000	0.0655994
0.1291667	1	0.6125	0.6125	0.0676846
0.0125000	4	0.2250	0.3625	0.0587720
0.0100000	5	0.2125	0.3875	0.0601079

Section 3.2.2: Methodology for Growing and Pruning the Classification Tree

Different from the regression tree model, instead of minimizing the RSS, the Gini Index is the metric minimized in the full model here. The Gini Index is a metric that indicate how stable our classification is. To be more specific, the smaller the Gini Index is, the stable our classification is. The formula for the Gini Index is shown in the Formula 4, where $|T|$ is the number of leafs in the tree, and $G(Leaf_l)$ is the Impurity Score at a certain leaf, which can be computed using the Formula 5. In this formula, n represents the number of levels in the response variable.

$$Gini\ Index = \sum_{l=1}^{|T|} \frac{n_l}{n} G(Leaf_l) \tag{4}$$

$$G(Leaf_i) = 1 - \sum_{j=1}^{m} \hat{p}^2_{(Y=level_j, Leaf_i)} \tag{5}$$

The fullTree2 model is the result of the tree model minimizing the Gini Index using all features. For example the reason of the fullTree2's first split on the content of *"Sugar"* at 38 as shown in the Figure 5 is that when training all the data, among other choices of splitting on other features or splitting at a different value on *"Sugar"*, the split on *"Sugar"* at a value of 38 creates minimal Gini Index, which in turn means this is the most stable split among all the choices.

Similarly, due to the need of avoiding the overfitting in of the model, the pruning process needs to be done to the fullTree2 model. as the regression tree does, tunning parameter are chosen for the classification tree to add a penalty term in order to prune the tree and improve the quality of the classification.

Section 3.3.3: Pruning the fullTree2 to Grow the Tree2 Model

In order to choose an optimal tuning parameter α, Table 2 is referenced. In order to have a lower test CER, a lower percent change in the RNE is considered. Plus, from the cp plot in the Figure 6, we can see that with 4 splits, the percent change in the RNE is the smallest. Since 4 split is not a large number of split to achieve and explain. We could use the corresponding $\alpha = 0.0125$ to prune the tree.

Section 3.3.4: Information revealed by the Tree2 Model

Based on the tuning parameter we choose, the classification tree model is built and visualized in the Figure 7. From this detailed visualization, we can see the proportion of observations in each leaf at each level. For example, the prediction for servings with *"Sugar"* less than 38, *"Trans.Fat"* greater than or equals to 0.75, and *"Saturated.Fat"* greater than or equals to 11 is very stable, since all of the observations (16% of all data) in this leaf are breakfast. Similar observation can be made for servings that have *"Sugar"* greater than or equals to 38. all of observations in this leaf are all desserts.

In order to make the result easier to be explained to the client's patients, a conciser and simpler visualization without seemly confusing metrics is created in the Figure 8. The color in this figure indicates the different levels of the response variable, and the legend showing the color-level correspondence is at the upper left corner of the tree diagram. Also, the darker a certain color is, the prediction is more stable.

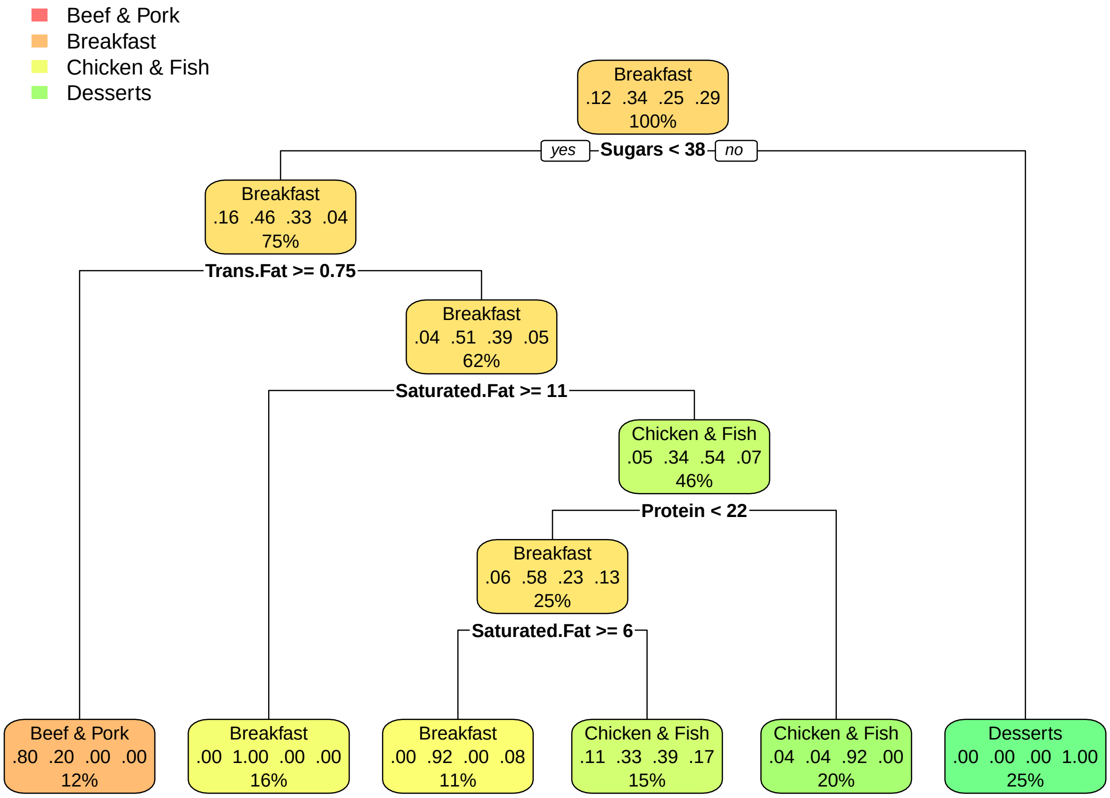

Figure 5: The Visualization for the fullTree2 Model, fig.asp = 0.6

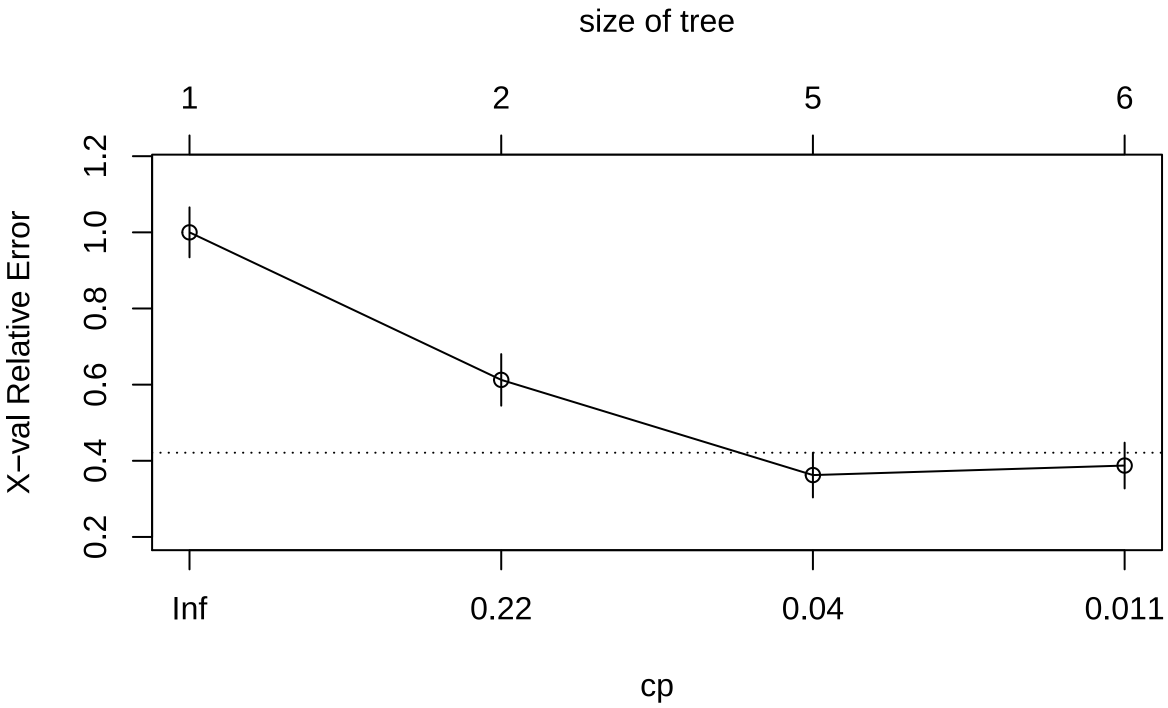

Figure 6: cp plot for fullTree2

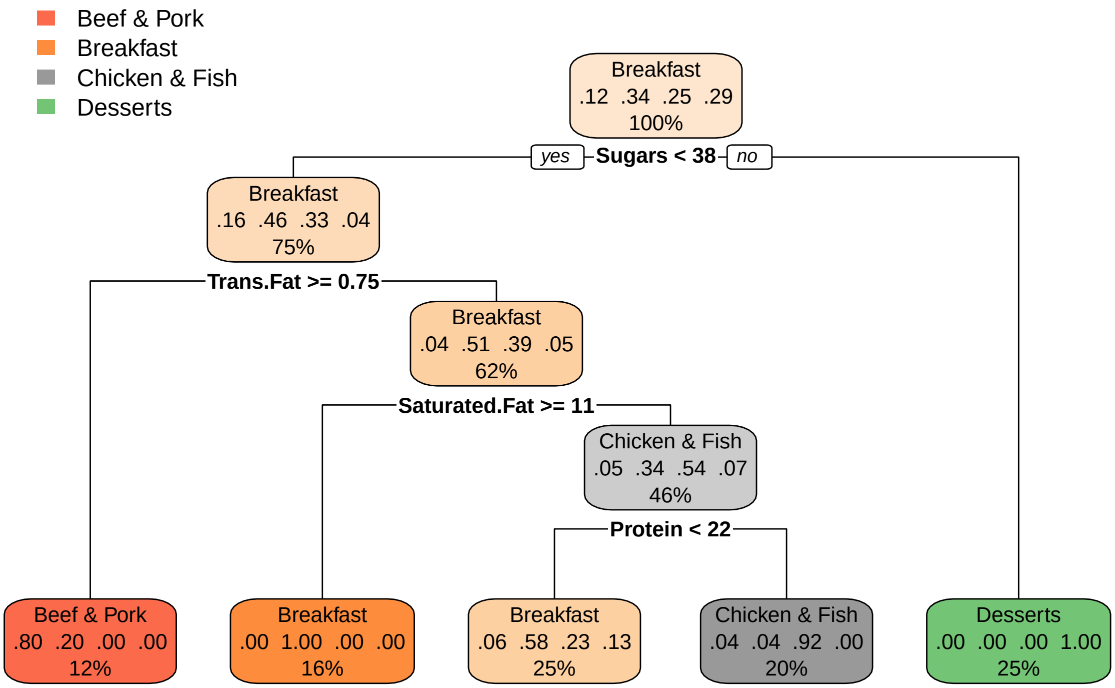

Figure 7: The Detailed Visualization for Tree2

From the splits, the Tree2 model shows that a higher value of *"Sugar"* is related to Desserts category. A higher value of *"Trans.Fat"* is related to the Beef & Pork category. A higher value of *"Sturated.Fat"*. A lower value of *"Protein"* is related to the breakfast category, and a higher value of *"Protein"* would more likely result in a serving at the category of Chicken & Fish.

More numerical combinations of these four features can be used to make the prediction in *"Category"*. For example, if a serving has *"Sugar"* of 35, *"Trans.Fat"* of 0.6, and *"Saturated.Fat"* of 15, the prediction from the Tree2 model would be the Breakfast category.

Section 3.3.5: Evaluating the Tree2 Model

In order to evaluate the accuracy of this Tree model. The test CER for the pruned Tree2 could be obtained by the Formula 3. The percent change in the RNE (xerror) for the pruned tree using 4 splits is 0.3625 from the Table 2. Hence, the test CER for Tree2 is $0.65574 \times 0.3625 = 0.2377057(23.771\%)$. Then, $122 \times 23.771\% \approx 29$ out of 122 observations are not correctly predicted.

Section 4: Predicting Calories

This project will model and predict the *"Calories"*.

Sections 4.1: Reasons for Using the Forest Technique

Since here our client want an accurate prediction on *"Calories"*, the forest models instead of the tree models are chosen. Since one tree, even with pruning, would result in overfitting problem, a forest combining several trees would increase the predictive accuracy. The Bagged Regression Forest and Random Regression Forest are used.

Section 4.2: the Bagged Forest Model (BgForest)

Section 4.2.1: Methodology of the Bagged Forest Model

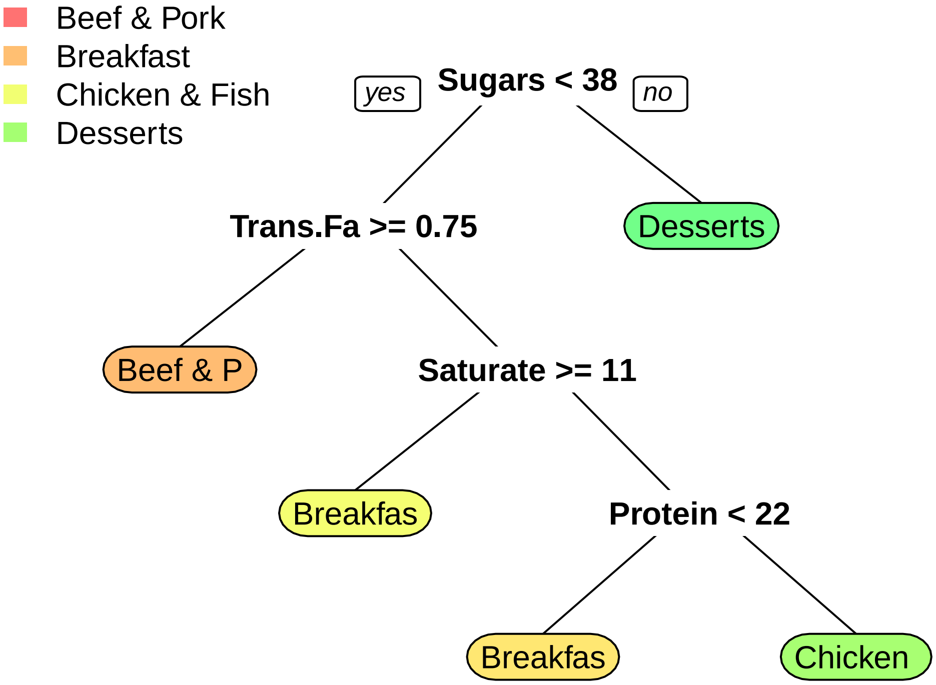

Figure 8: The Concise Visualization for Tree2

To produce a better prediction, the Bagged Forest method, create bootstrap samples, and use this samples to fit tree models, and use all these tree models to make the final prediction. The prediction is made using the Formula 6, where B is the number of bootstrap samples used in the forest.

$$\hat{Y}_B^{bagged} = \frac{1}{B}\sum_{b=1}^{B}\hat{Y}_b \tag{6}$$

Section 4.2.2: Fitting the Bagged Forest Model

1000 bootstrap samples are created to train the Bagged Forest model (BgForest). Since for each sample, some of observations are not include in that sample, so that they are used as the test data for the tree model we created for each sample. These observations are called the Out-Of-Bag (OOB) rows, and the prediction made using these rows are called the OOB prediction. Also, use the OOB predictions, the test MSE could be created, and denoted as the OOB error estimate. For the BgForest model, the test RMSE is the square root of the OOB error estimate, which is 31.20229. Comparing the test RMSE of the Tree1 model which is 89.80469, the BgForest model improves 65.26%.

Section 4.3: the Random Forest Model (RdForest)

Section 4.3.1: Methodology of the Random Forest Model

The Random Forest technique still create bootstrap samples and use these samples to create tree models, and it makes the final prediction as the Bagged Forest does. However, the Random Forest only use $\lfloor\sqrt{p}\rfloor$ of the total number of features (p) to train the tree models for each bootstrap sample, which makes the trees become more representative for a wider range of subsets of the population.

Section 4.3.2: Fitting the Random Forest Model

1000 bootstrap samples are also made to train the Random Forest model (RdForest). The test RMSE, which is the square root of the OOB error estimate, is 32.90963. Thus, the RdForest model improve from the Tree1 model by 63.35%.

Section 4.4: Evaluate the Importance of Variables

Section 4.4.1: Meaning of the Importance

The importance measures the contribution of a certain variable to the prediction. The importance is determined by finding the percent increace in the OOB error estimate after doing the permutation in a variable. First, a certain variable would be randomly permuted and then fit the forest model again. Then, the new OOB error estimate would be computed using the Formula 7.

$$Percent\ Increase = \frac{(OOB\ error\ estimate_{Permuted} - OOB\ error\ estimate_{Original})}{OOB\ error\ estimate_{Original}} \tag{7}$$

Section 4.4.2: The Importance from the BgForest Model

The importance chart is shown in the Figure 9. From the importance plot, there are two important variables, which are *"Carbohydrates"* and *"Total.Fat"* since after permuting, the percent change in the OOB estimate error increase a lot, which means their existence in the model would lower the OOB estimate error. This result confirms the result from the Tree1 model, since the first several splits are also created on these two feature as shown in the Figure 3.

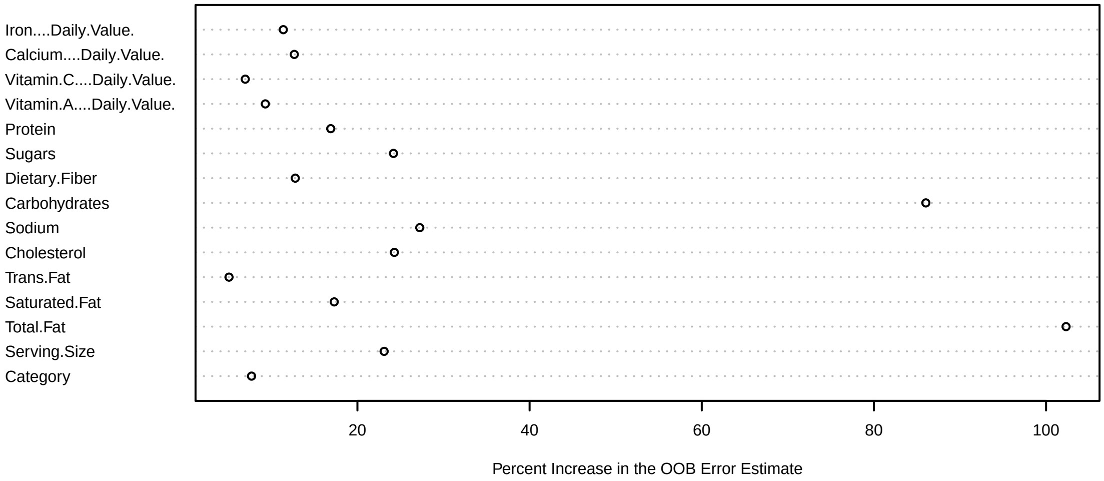

Figure 9: The Importance Plot for BgForest

Section 4.4.3: The Importance from the RdForest Model

The importance chart is shown in the Figure 10. From the importance plot, there are three important variables, which are *"Sugar"*, *"Carbohydrates"* and *"Total.Fat"*. This result also confirms the result from the Tree1 model, since the first several splits are also created on these two feature as shown in the Figure 3, while the importance of the feature *"Sugar"* is not shown in the Tree1 model.

Section 4.5: Comparing the predictive accuracy

In order to choose a final model that would be used for prediction, two forest models are compared using the test RMSE obtained by the OOB error estimates. From the results shown in the Table 3, we can see that the BgForest model is the model with the best predictive accuracy. It has a test RMSE of 31.20229, and improve from the Tree1 model by 65.26%. Hence, the BgForest model is chosen to do the prediction job asked by the client.

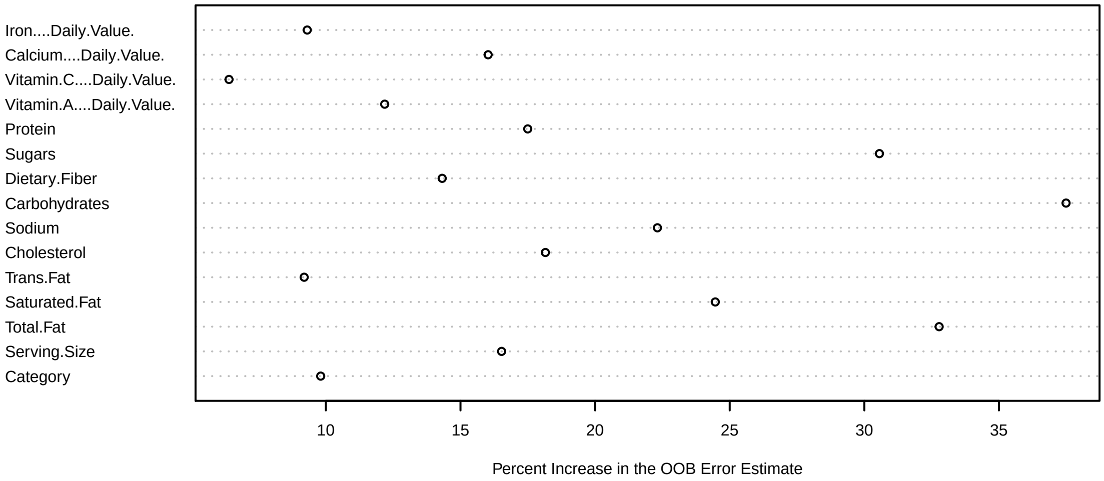

Figure 10: The Importance Plot for RdForest

Table 3: Models and their test RMSE

models	**test RMSE**	**Improvement from Tree1**
Tree1 model	89.80469	
BgForest model	31.20229	65.26%
RdForest model	32.90963	63.35%

Section 4.6: Exploring the Partial Dependence

Other than the prediction, the dependence between each feature and the response variable *"Calories"* could be visualized. The two variables chosen are the *"Carbohydrates"* and *"Total.Fat"*. The dependence between the *"Carbohydrates"* and *"Calories"* is explored in the Figure 11, and it is a almost a linear relation. The dependence between the *"Total.Fat"* and *"Calories"* is explored in the Figure 12, and it is also a almost a linear relation. The slope is steep between 10 and 20, while the increase in *"Calories"* become slower as the *"Total.Fat"* keeps increasing.

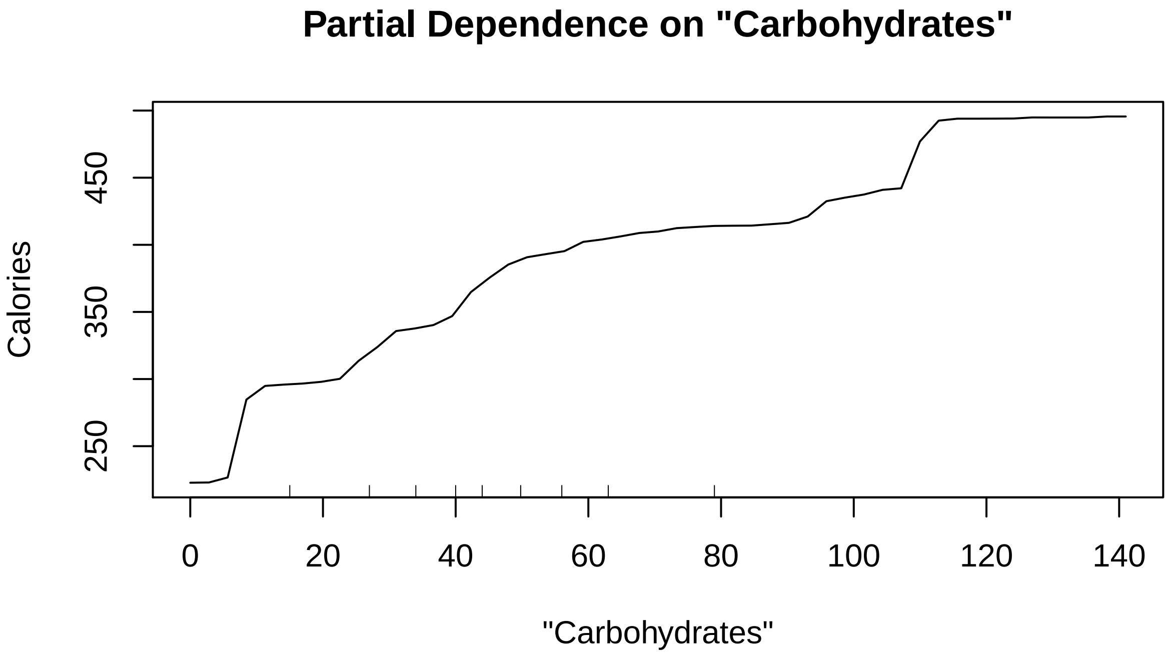

Figure 11: Partial Dependence on Carbohydrates

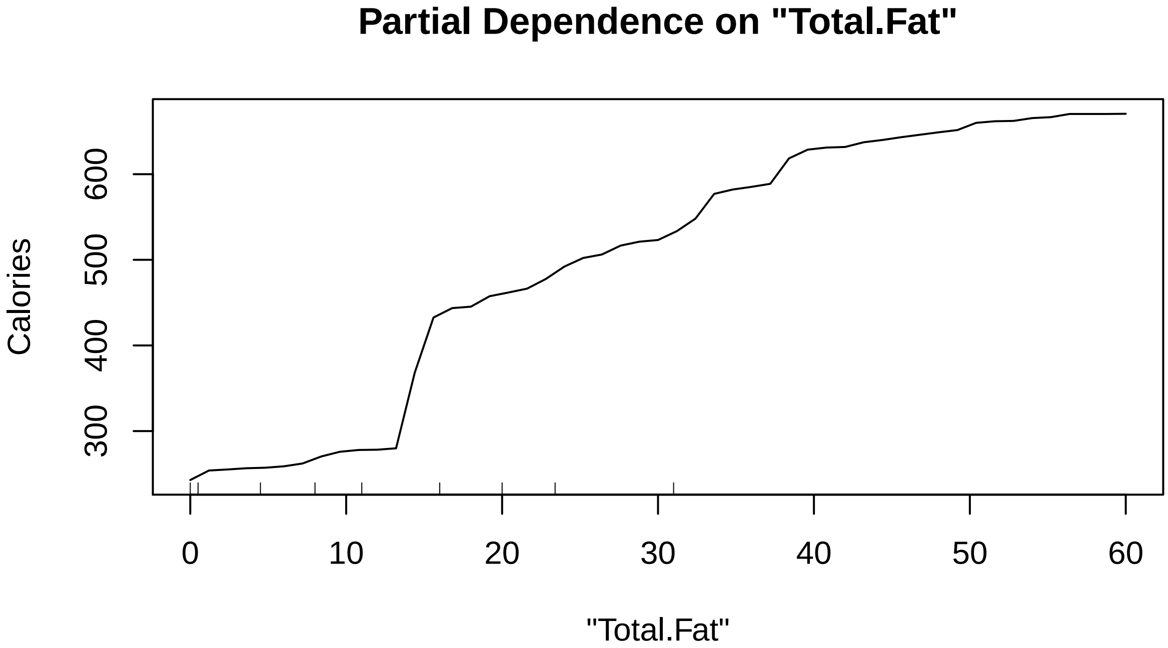

Figure 12: Partial Dependence on Total.Fat