Data Mining—Weight

By Shubh Shah

**Introduction:**

Obesity is a major epidemic facing the modern world, with many contributing factors to its cause. This includes the demographic and lifestyle. Utilizing this dataset from the UCI Machine Learning Repository, this study would focus on the variable of Weight as the central measure. Using exploratory data analysis, and the different predictive models (Linear Regression, Regression Tree, and Random Forest), we can analyze trends and explore the factors that contribute to obesity level.

A screenshot of a computer program

Description automatically generatedA green text on a white background

Description automatically generatedData Information and Structure:

This data is through the UCI Machine Learning Repository. From the dataset, we can see the 17 variables. The picture below shows their name, the class it belongs to, and the meaning behind it. For this project itself, we will be using ***Weight*** as the dependent variable itself. The next step to understand the data would be producing the summary to show the minimum and max for the continuous values (***Age***, ***Height***, and ***Weight***)

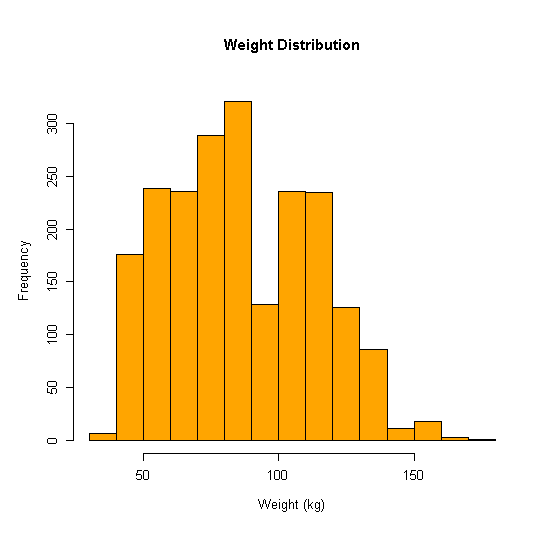
**Part I: Exploratory Data Analysis (EDA)**

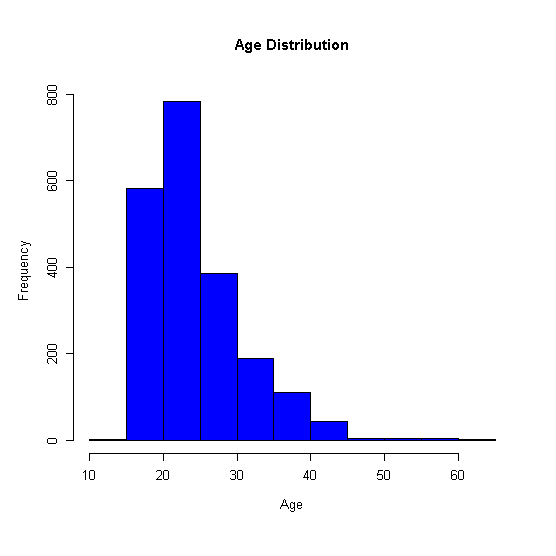
A screenshot of a computer screen

Description automatically generatedFrom our summary, we can see that our minimum age would be 14, whereas the maximum age would be 61. We can understand that the minimum weight within the whole dataset would be 39.00 kg, whereas the max was 173 kg. The average weight was 83 kg.

A screenshot of a computer

Description automatically generatedThe image on the left would represent the variables that we are working with. This shows the variable name, the classification, and the meaning behind the acronyms. For our data, we will be working with ***Weight*** (as mentioned before above).

A red graph with white text

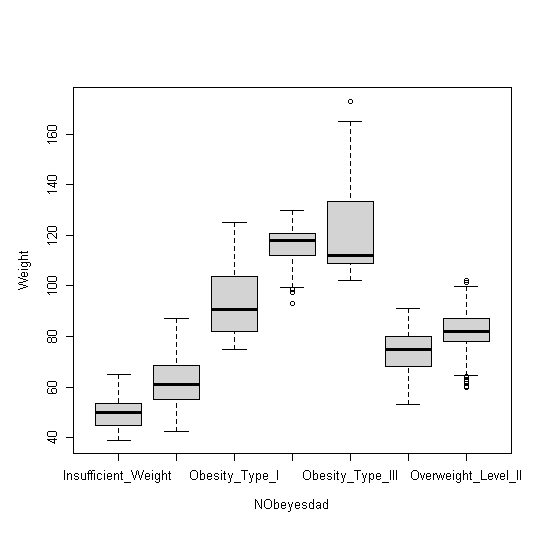
Description automatically generated.

The images above represent histograms of certain variables (***Age***, ***Height***, and ***Weight***), where it provides us with a lot of information. In ***Age***, it’s mentioned that the maximum age is 61 years old. However, our data is right skewed, which suggests that majority of the data is focused on the younger age. The middle image represents the frequency of heights collected, where it is symmetrical. We can see that most individuals have heights that vary between 1.6 meters and 1.8 meters. Finally, we have the weight distribution, where we can see that the data is symmetrical with it skewing towards the right, as most of the weights would fall between 60kg and 100kg. From these histograms, we can get a quick summary of the dataset’s demographic and characteristics. Below would be the scatter plot which would represent between height and weight.

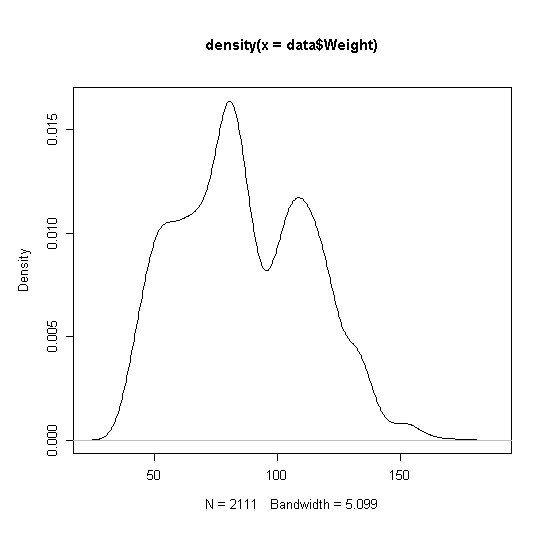
A diagram of weight

Description automatically generatedA green text on a white background

Description automatically generated

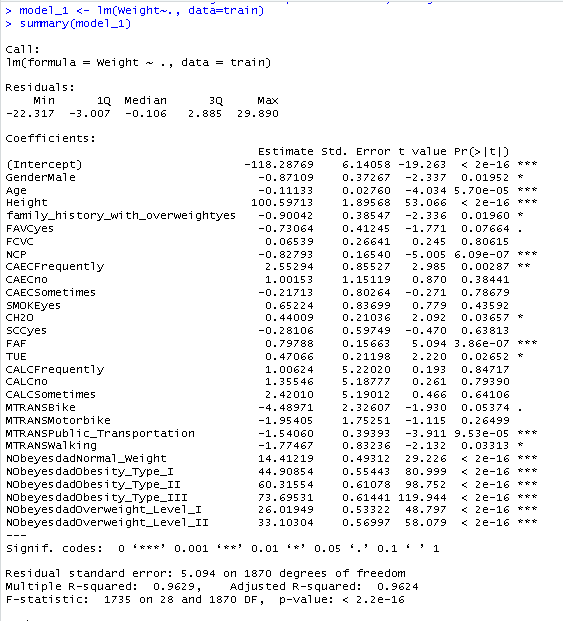
From the scatterplot, we had set the Weight as the X-Axis and Height as the Y-Axis, where we are able to see that there is a positive correlation. We are able to see multiple clusters, which could represent a common trend between the height and weight itself. We are able to see a considerable amount of variability. For example, if we had selected a height of 1.7m tall, we can see that weights range between 50kg to 120kg which would suggest that other factors, such as lifestyle and health conditions, have an influence. We can see outliers such as low height, but high weight. Using this model, we can see the general relationship between the dependent variable (weight) and one of the variables (height).

The second image on the previous page a boxplot between the variable ***NObeyesdad*** and ***Weight***. This would represent a summary of how weight varies from the different obesity levels. We can see that the increase in weight also increases the level itself. From here we can see the classifications, where Obesity Type III was the highest. We can also see the outlier. The variability within our data can tell us that weight is a strong indicator of obesity level, however other factors such as height or age could also influence it too.

The last image would be the density plot, which would show the distribution of the weight. From the image on the right, we can see two peaks within the weight distribution. The first peak occurs around 60-70 which could represent the “Normal Weight” or the lighter obesity categories. The second peak occurs around 90-100kg range, which suggests that could be associated with the heavier obesity categories such as Obesity Type I or Type II.

**Part II: Linear Regression with Stepwise**

A close-up of a word

Description automatically generatedStarting off, I had created a subset with a 90% training and 10% testing. After this, I created a model to see the fit within the data. I had used ***Weight*** as my dependent variable, where I would be using all the variables within it. After the model was made, I used *summary(model\_1)* to provide the summary of the training subset. The summary had shown us that several variables such as ***Height, Weight,*** and ***NObeyesdad*** showed a strong statistical relationship with weight. The model had achieved an R-Squared value of 0.9629, which means that 96% of the variation in weight could be explained through the model. Certain predictors such as SMOKE and FCVC are not important to the model as their probability value (p-value) is greater than 0.05 (5%), meaning that they wouldn’t contribute to weight predictions. Overall, the model fits very well, however it would be improved if non-significant predictors were removed. This will be explored further within this model.

A screenshot of a computer program

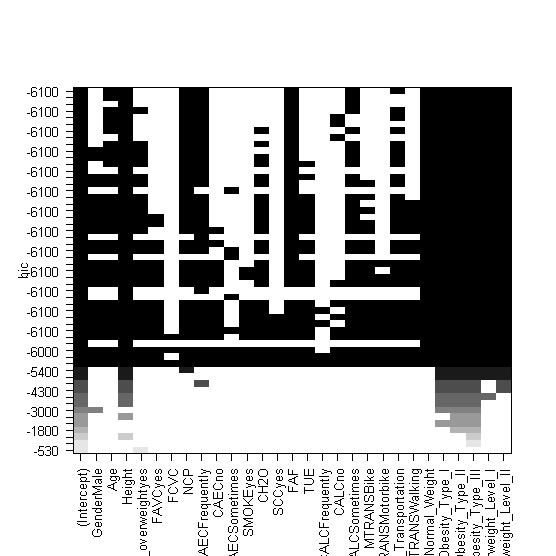
Description automatically generatedA screenshot of a computer code

Description automatically generatedEvaluating model fitness, we can see an MSE of 25.95, an R squared of 0.9629406 and an adjusted r squared of 0.9623857. To interpret this, the MSE would measure how well the model would fit the training data. It would represent the residuals (errors) of Weight by finding the difference between the actual values and the training predicted model values. We are also able to see an AIC of 11603.23 and a BIC of 11769.7. With these predictors, they concentrate on the number of parameters within the model. AIC is a bit more lenient as the BIC would penalize on how complex the model is. Lower values for these metrics would indicate a better performance. After finding the MSE, AIC, BIC, I worked on the out-of-sample prediction where I had got the test error. Since the model works on future data, I had used the predict function. My test MSE was 23.84588, which is a bit lower than the training MSE, suggesting that it generalizes well to the test data. I then calculated the Mean Absolute Error which represents the absolute difference between the predicted and actual values. On average, my models’ predictions deviate by 3.7kg from the actual values, which is good. With a low in-sample-prediction and also a low out-of-sample prediction, the model performs consistently.

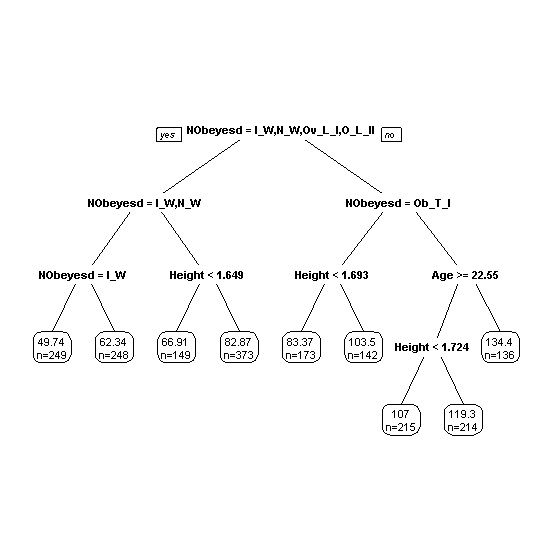
A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer program

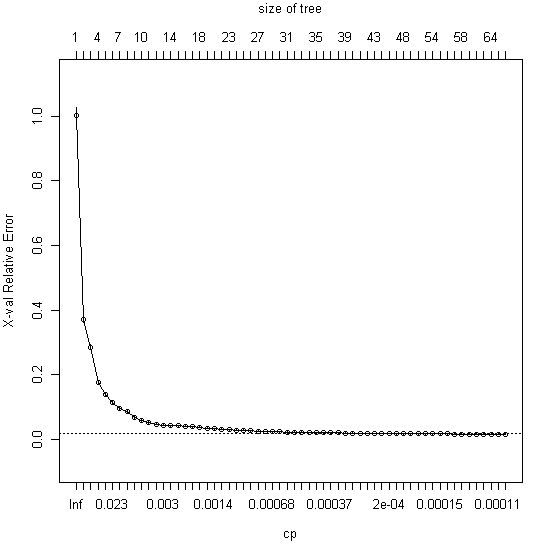
Description automatically generatedI had then focused on doing a stepwise regression to figure out which variables would be important to my model. I had first loaded the *leaps* library, in which I was able to use *regsubsets*, which would used for variable selection. I then printed a BIC visual plot. The black cells means that the variable is included in the model, whereas the white cell would exclude them. Looking at this, we can see certain categories such as Height, Age, and family\_history\_with\_overweightyes could appear frequently in other optimal models. As you move down, the number of predictors increases. I then moved on and performed a stepwise regression. There are two models shown, the first one would be the null model with no predictors, whereas the full model would have all the predictors. I had then set the scopes and set the direction for both. Forward selection has no predictors, adds them one at a time, whereas backwards eliminations start with all predictors and removes it one at a time. With both, it would combine both forward selection and backward elimination and remove each step. After performing it, we are able to get an AIC of 6206.96. Variables such as ***Height*** and ***NObeyesdad*** represent a strong predictor to weight, whereas other variables such as ***SMOKE*** and ***SCC*** do not improve the model. The stepwise model has a lower AIC, which would suggest that is a better fit than a full linear model.

**Part III: Regression Tree**

The second model was made was a regression tree. Since weight is a continuous variable, we wouldn’t be a able to use a classification tree. I had first loaded in the *rpart* library, then also loaded in *rpart.plot* to show the graph itself . From the split, we can see that the split is based on the ***NObeyesdad*** variable. The root node represents the starting points, where we moved down to the first split (node 2 and 3). Node 2 has Insufficient Weight, Normal Weight, Overweight Level 1 and Overweight Level 2. If we look at Insufficient Weight (first split towards the “yes” side) we can see that it has 249 observations with an average weight of 49.7 kg. Normal weight has around 248 observations where the average weight was around 62.3 kg. Moving on to the second split, the branch further divides the individuals to Overweight Level I and Overweight Level II based upon the categories of height. For a height less than 1.649 meters, the predicted weight was 66.91 kgs, where the number of observations was 149. For height greater than 1.649 meters, the predicted weight was 82.87 kg where the number of observations was 373. For the right-hand side of the tree, we will be observing Obesity Type I. Here we can see that people who have a weight less than 1.693 meters would have a predicted weight of 83.37 kg, where the number of observations was 173. For a height greater than 1.693 meters, the predicted weight is 103kg where the number of observations was 142. These are some of the takeaways from the graph itself, however we can see that the most significant factor would be ***Nobeyesdad*** as it separates lower weights categories from the higher weight categories. We can also see that height plays a factor in weight, especially within the higher A screenshot of a computer code

Description automatically generatedweight categories. If you look at the “Age” node, we could see that it influences the predictors for individuals in the higher weight class. It’s surprising to see younger individuals having higher weights. One takeaway I was able to see within it is that higher height would generally lead to higher predicted weights.

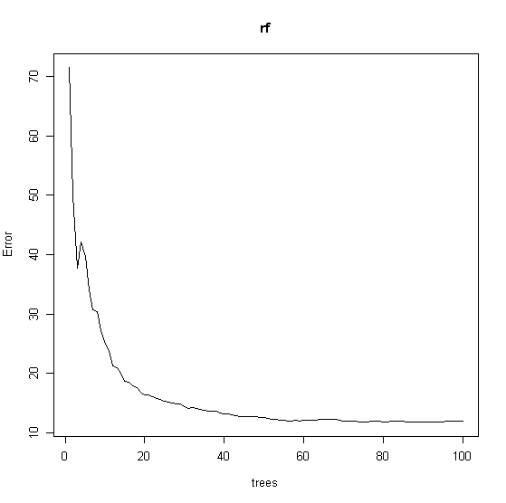
A computer code with blue text

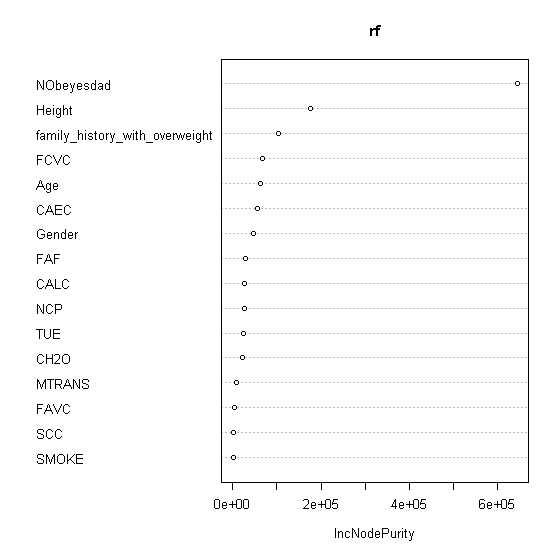
Description automatically generatedI had then moved on to calculating the in sample and out of sample predictions by using the *predict()* function. My in-sample prediction value was 46.66214 whereas my out-sample prediction was 53.09319. We can observe that the training error is lower than the testing error, which suggests that the model fits the training data better than the testing. We can see that the difference between these two values are around 7, which isn’t large enough to consider overfitting. The MSE and MPSE suggest that the regression tree is performing well for this dataset. However, if we compare the out of sample prediction error from the regression tree to the linear regression model performed earlier, we are able to conclude that the linear regression model performed better as it had a lower out of sample prediction value. This had suggested that I need to focus on pruning the tree. To prune the tree, I had first built a large tree model. I had then plotted the complexity point, where I had focused on the leftmost value that falls just a bit below the line. In my scenario, this would be 0.00033. However, to get a closer look at it, we would need to print the complexity parameter table to show how much additional complexity in the tree would decrease the error. It’s important to note that a higher complexity A screenshot of a computer

Description automatically generatedparameter would represent in a simple tree, whereas a lower complexity parameter (as seen within the image on the right) would allow for more complex tree. The root node error would represent the MSE if we had used the average of ***Weight*** as a prediction for all observations. For example, within our dataset, we had a root node error of 686.84. As more splits start to happen (nsplit), the relative error decreases as the model becomes more complex and fits better towards the training data. However, the validation error (xerror) doesn’t always decrease, which could be a sign of overfitting when it becomes to complex. An example could be row three, where we had 2 splits. After the split, the relative error had decreased to 0.2711, and the cross validation would be 0.2860 with a standard deviation of 0.0084. Likewise, if we had looked at row 18, which is 17 splits, the relative error would decrease to 0.0323, whereas the validation error would be 0.0381. This would be a great place to prune the data itself., as it has a CP of 0.0015, and its where the cross validation would stabilize. For the in-sample MSE for the large tree, I had received a value of 7.070959, and for my out of sample MSE was 20.75. Both in sample and out of sample MSE are much smaller than what was calculated earlier. Especially comparing to either the linear regression or even the baseline error for the regression tree. The in-sample value of 7.07 indicates that the tree does capture the patterns in the training dataset, whereas the out of sample value of 20.75 indicaes that the model does generalize well, but it can be argued that there is overfitting, which would require further pruning to reduce the gap.

**Part IV: Random Forest**

The last model I had made was the random forest model. I had used multiple sources for this. The first resource was **R and Data Mining: Examples and Case Studies** by Yanchang Zhao. His example was using the iris dataset, but since ***Weight*** is a continuous variable, I had to look through other resources. A screenshot of a computer code

Description automatically generatedThey will be listed below. I had first loaded in the *randomForest* library, in which I had used ***Weight*** as the dependent variable, and had it use all of the columns in the training dataset as predictors. I had set it to 100 trees and had disabled the proximity measure between the values. We can also see that the model considers 5 randomly selected predictors at each tree. The in-sample MSE for the training dataset is 12.35058, which is very good as it indicates a better model performance with the training dataset. The random forest model was able to 98.2% of variances in the training set, indicating that it is a good fit. From the random forest plot, we can observe that the error is high with fewer tree models, however that error rapidly decreases and eventually A screenshot of a computer program

Description automatically generateddoes stabilize. Essentially, adding more trees will not significantly reduce the error. The random forest would reach an optimal performance around 70-80 trees. I then printed variable importance. From the observations, we can see that ***NObeyesdad*** is the most important variable as it has the highest IncNodePurity, with ***Height***, and ***family\_history\_with\_overweight*** being the next most significant. We can also see variables such as ***SMOKE, SCC, and MTRANS*** have low importance, suggesting that they contribute insignificantly to the model. I had then plotted variable importance plot where it was hierarchical, based upon the highest value of the IncNodePurity. From the plot, we can see the ***NObeyesdad*** being the most important, followed by ***Height***, and ***family\_history\_with\_overweight*** being third most important. I had then calculated my out of sample MSE by using a similar format used within linear/logistic regression and also the regression tree. That is where I got an out of sample value of 59.18562. Now looking at this value, I am able to see or even understand that the model has some sort of overfitting. Interpreting my out of sample, I can guess that the performance of the model would drop when there are predictions on unforeseen data. The best way to lower it possibly would be to decrease the number of trees, or cross validatation to help the model generalize well on unseen data, which could lower the out-of-sample MSE.

**Part V: Summary:**

This project evalutes obesity trends and the contributing factors using three different models: linear regression, regression tree, and random forest. The dependent variable was ***Weight*** and finding the most effective model in predicting it.

|  |  |  |
| --- | --- | --- |
|  | In Sample | Out of Sample |
| Linear Regression | 25.95 | 23.85 |
| Regression Tree | 7.07 | 20.75 |
| Random Forest | 12.35 | 59.19 |

Starting off with our first model, the linear regression had an in-sample MSE of 25.95, whereas the out of sample was 23.85. The linear regression does generalize well with unforeseen data, achieving an r-squared value of 96.29. Our second model was the regression tree, where I had got an in sample of 7.07 and an out of sample of 20.75. The tree captured the training patterns for the in-sample MSE but does increase in the out-of-sample MSE, with some sort of overfitting. Pruining the tree helped it by removing complexity and improving performance. There can be more pruning to fix overfitting errors. The last model would be Random Forest, where I had got an in-sample of 12.35 and an out of sample of 59.19. This does suggest that the random forest had overfitted to the training set. However, it does capture the low in-sample error. The model performance had plateaued around the 70-80 trees. For this dataset, I had found linear regression to be the best model due to the low MSE values. I believe with further tuning on Random Forest and Regression Tree could help reduce the out-of-sample value.

**Part VI: Sources and References**

* Statology. "Random Forest in R: A Practical Guide." *Statology*, <https://www.statology.org/random-forest-in-r/>. Accessed 1 Dec. 2024.
* Liaw, Andy, and Matthew Wiener. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. CRAN, <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>. Accessed 1 Dec. 2024.
* ListenData. "Random Forest with R." *ListenData*, [https://www.listendata.com/2014/11/random-forest-with-r.html. Accessed 3 Dec. 2024](https://www.listendata.com/2014/11/random-forest-with-r.html.%20Accessed%203%20Dec.%202024).
* Tian, Shaonan. **Regression Trees**. Data Mining Course Material, n.d. PDF.
* Tian, Shaonan. **Cross Validation**. Data Mining Course Material, n.d. PDF.
* Tian, Shaonan. **Lab 2: Regression and Variable Selection**. Data Mining Course Material, n.d. PDF.
* Zhao, Yanchang. *R and Data Mining: Examples and Case Studies.* 2013. Retrieved from <http://www.RDataMining.com>. (Textbook)
* Tian, Shaonan. **Explore the Iris Dataset with R**. Data Mining Course Material, n.d. PDF. (Data Exploration)
* Estimation of Obesity Levels Based On Eating Habits and Physical Condition [Dataset]. (2019). UCI Machine Learning Repository. https://doi.org/10.24432/C5H31Z.