Enrique Posada Lozano A01700711

Estimation of Obesity Levels and Weight

Introduction :

Over the last decade, Latin America has had a recurring problem of obesity, from which around 57% of adults are considered to be obese or overweight. Being overweight or obese is a malnutrition problem defined as an abnormal or excessive accumulation of fat up to the point of presenting a health risk. As a result of obesity, many other health risks and diseases may follow and affect a person under this category. However, this can be prevented by taking necessary precautions and making better choices regarding food choice and physical activity. Although this is a problem around many parts of the world, this report will solely focus on data pertaining to latin american countries in order to create models that can help both determine the level/type of obesity a person belongs to as well as getting an approximate calculation of the weight a person might have, given different attributes.

The importance of this project is to also project the difference in performance and accuracy a model can obtain with different approaches, both with SciKit's implementation for trees and linear regression and the ones developed from zero in order to determine what might be a better choice for a particular regression task.

Dataset :

In order to develop said models, the dataset analyzed and used to train the models comes from the UCI Machine Learning repository and includes data from individuals in Mexico, Peru and Colombia. This dataset can be used for classification, regression and clustering tasks. Data was collected through surveys and consisted in asking the following questions:

- What is your Gender? ("Male/Female")
- What is your Age?
- What is your Height? (Meters)
- What is your Weight? (Kg)
- Has a family member suffered or suffers from overweight?("Yes/No")
- Do you eat high caloric food frequently? ("Yes/No")
- Do you usually eat vegetables in your meals?
- How many main meals do you have daily?
- Do you eat any food between meals?
- Do you smoke?
- How much water do you drink daily?
- Do you monitor the calories you eat daily?
- How often do you have physical activity?
- How much time do you use technological devices such as cell phones, video games, television, computers and others?
- How often do you drink alcohol?
- How often do you drink alcohol?
- Which transportation do you usually use?

From this information, 17 attributes were obtained which consist of Gender, Age, Height, Weight, family history with overweight conditions, and smoking habits, followed by eating habit attributes like frequent consumption of high caloric food (*FAVC*), frequency of consumption of vegetables (*FCVC*), number of main meals (*NCP*), consumption of food between meals (*CAEC*), consumption of water in daily basis (*CH20*) and consumption of alcohol (*CALC*) and last but not least, attributes related to physical condition are calories consumption monitoring (*SCC*), frequency of physical activity (*FAF*), frequency of hours spent using electronic devices (*TUE*) and type of transportation used (*MTRANS*). The last attribute is the level of obesity an individual belongs to (termed as *NObeyesdad*), which has 7 levels: Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, and Obesity Type III.

Models :

With the information from the dataset, there are going to be three models being developed without a framework like SciKit, one for classification and two for regression. The model used for classification consists of a decision tree using entropy as the criterion for impurity and it looks to estimate the obesity level of an individual based on their physical condition as well as their eating habits. For regression, a decision tree regressor will be used and compared against a linear regression, both using MSE (mean square error) and RMSE. MSE is used in order to measure how close the model is to the real data while RMSE (square root of MSE) is used after training the model for model accuracy and a better and easier interpretation of the error during training. The implementation of both the decision tree regressor and decision tree classifier are made for it to be a CART tree, which stands for Classification And Regression Tree which in turn is a tree able of doing both tasks but each with a different cost function.

Finally, in order for the models to predict adequately, preprocessing the data is needed. Fortunately, it is suggested by the owners of the dataset that their data has been preprocessed for data mining, but still there is some preprocessing needed for the model to work. For classification tasks, columns that consist of strings, either having two values (“male/female” or “sometimes/frequently/never”) are One-Hot Encoded in order to separate them and differentiate them in the dataset since there is no order in those values. And since the value we want to predict is also a string but has an order existing between them such as it goes from least weight (insufficent weight) to most weight (obese), Label Encoding is used to convert the label as numeric data, assigning each class a number from 0 to 6, indicating each of the 7 levels of obesity.

For regression, normalization is applied using Min-Max scaling in order to scale continuous features such as Age, Height and Weight for faster convergence of the model.

Enrique Posada Lozano A01700711

CART Tree :

A CART tree as previously mentioned is a Classification And Regression Tree, which is a Decision Tree able to classify or do regression tasks. The point in the tree itself is to find the best possible split for a given set of features and labels, which might be determined in a classifier by the tree with the highest information gain or the minimum MSE for a regression tree (Shown with the formula below).

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta))$$

Basically, the tree traverses each feature and its value and splits them in the values less than or equal to a given value as left nodes and the rest as right nodes. Then it applies the impurity function (entropy, MSE) and multiplies it with the average of nodes it contains, and the sum of them determines the quality of a split (MSE -> minimum value, informationGain -> greatest value). In the end the model results in a “big” and “smart” tree that might be prone to overfitting, in this case it did not occur with the dataset provided.
The tree created would then be used for predictions by being given data and traversing the tree. For regression trees, RMSE and R-Squared was used to measure the performance/accuracy of the models. RMSE was used for interpreting the error (MSE) as the error there is between the expected values and the predicted data. R-Squared was also used to measure how close the data is to the fitted line as a percentage, so the closer it is to 1, the better the model is.

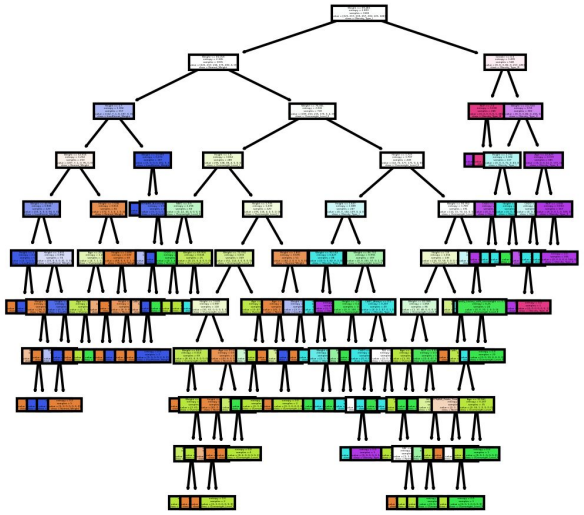
Decision Tree Classifier :

The decision tree classifier aims to predict one of the 7 classes provided (Insufficient Weight, Normal Weight, Overweight I, Overweight II, Obesity Type I, Obesity Type II, Obesity Type III) encoded with values from 0 to 6. In order to select the best split, entropy and information gain are used, by calculating the entropy of the left nodes and of the right nodes and then comparing them with the system entropy of a node.
For the impurity function, entropy was the criteria chosen for this project, adding up the values multiplied by the log2 of the value.

Shannon's entropy equation:

$$H(X) = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$

In the image below, one can observe how the tree for the model looks like, having colors representing each of the levels of obesity.

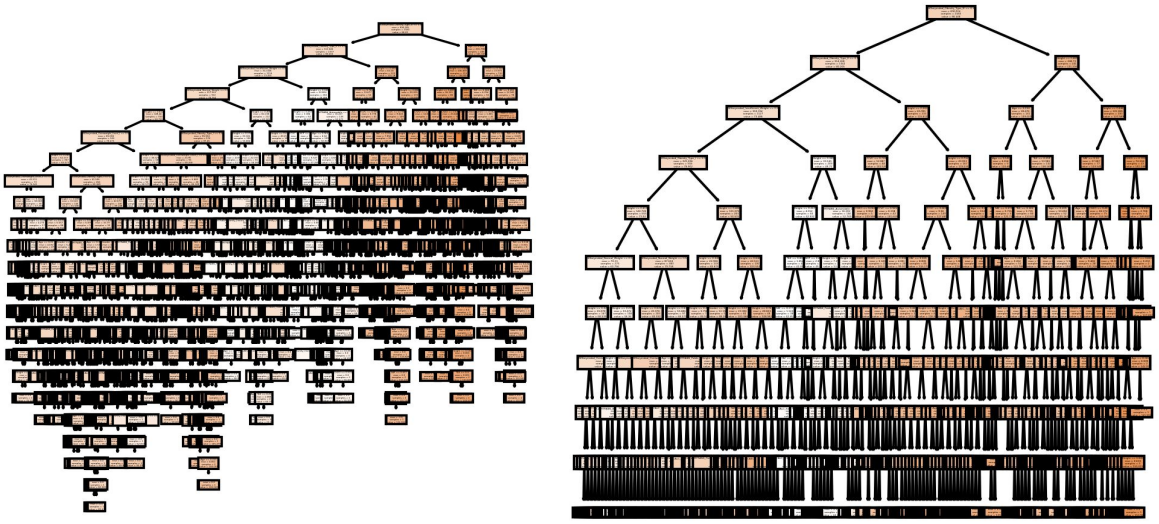


Decision Tree of the model plotted with SciKit (Link)

Decision Tree Regressor :

In contrast to a decision tree classifier, a regression tree model is much bigger (as seen in the image below), having more leaves and more branches. The regression task for this model is to predict the weight of a person given the attributes mentioned before, and the way it handles the previous label of obesity levels is now one hot encoded instead label encoded, resulting in a tree that works with 32 columns, working around 6 minutes in order to create the tree. It makes use of the Mean Squared Error (MSE) as the impurity function for each node.

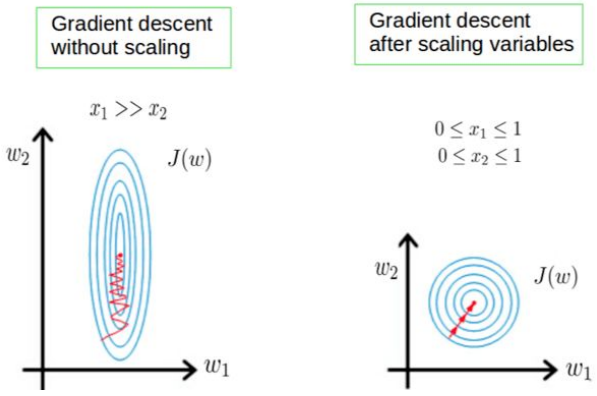
Different to the decision tree classifier, one key observation is the depth and length of the tree, which is considerably huge, thus one way to minimize this is by establishing a max depth for the tree. Testing this with Scikit's implementation of a decision tree regressor cuts the length extensively while conserving a similar RMSE, meaning this must be applied to the personal implementation so that in a way it can also speed the model. After implementing a control for max depth of a tree (10 levels be specific), the model takes much less time to run, and having half of the original tree levels, it manages to achieve a less RMSE than the original tree (from ≈ 5. down to ≈ 4.). In contrast, the model had an R-Squared value of 97.8% while SciKit had a value of 98.2%.



Decision Tree of the model plotted with SciKit (Link) | Decision Tree Regressor after having a max depth of 10

Linear Regression :

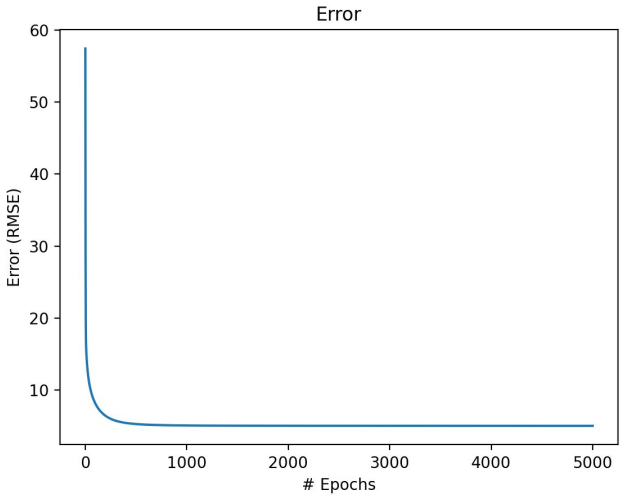
Just like the regression tree, the aim of this model is to predict the weight of a person given the same attributes as the tree in order to later compare both models. The training for the model can conclude any of three ways : either by reaching a given number of epochs (in my case, 5000 is the max number of epochs declared initially), having the same value between the previous coefficients and the new calculated coefficients (implying no change between them), or having the error reach a minimum (in this case, having a RMSE lesser than 5).



During the development process, the main problem with the model was not scaling data, causing the model to greatly diverge up to the point where the coefficients/weights were prone to becoming “infinite”. After scaling the features, the model proved a better and faster convergence.

A comparison of gradient descent with normalization

The best learning rate for the model appeared to be 0.305, since it allowed the fastest convergence of the model while also diminishing the problem diverging and having coefficients run up to “infinite”. After training the model, RMSE was used to interpret the distance (on average) from the model, and it gives an error estimated around 5, while giving an R-Squared value of 95.7% and Scikit an R-Squared value of 95.7% (meaning both models provide similar results).



Error provided by the model

Random Forest Classification :

Random Forests are another way to extract information from a set of data. The appeals of this type of model are:

- It emphasizes feature selection — weighs certain features as more important than others.
- It does not assume that the model has a linear relationship — like regression models do.
- It utilizes ensemble learning. If we were to use just 1 decision tree, we wouldn't be using ensemble learning. A random forest takes random samples, forms many decision trees, and then averages out the leaf nodes to get a clearer model.

In this analysis we will classify the data with random forest, compare the results with logistic regression, and discuss the differences. Take a look at the previous logistic regression analysis to see what we'll be comparing it to.

MIGHT BE REPLACED!!

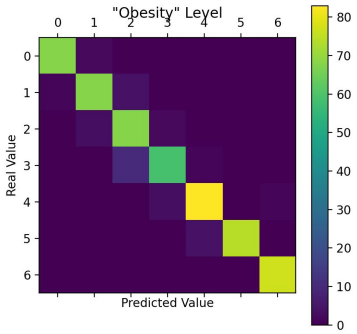
Model Validation and Results :

Overall, each model trains with 75% of the dataset and uses the remaining data (25%) to test the models and validate their accuracy.
For classification, both the decision tree implementation and Scikit's decision tree classifier achieve an accuracy over 90%, each with similar accuracy and minimal difference. Max depth was initially used to avoid the chance of overfitting by cutting one or two of the levels from the model, but it affected more the accuracy of the model, therefore there is no limit on the max depth of the model. One notable observation between these two models was performance, however an idea proposed to speed up the process consisted in rounding values in Age and Height so they would all be within the same format, eliminating values like 24.567 to just 25 (or 24), but while doing this does in fact accelerate performance, it comes

at the cost of losing accuracy in both decision trees, so data is left untouched (not rounded) to conserve a high accuracy in predictions.

```
SCIKIT Decision Tree Classifier Accuracy : 0.9507575757575758 %
Accuracy of Model with Test Data (%) :  0.9412878787878788
```

As seen on the picture above, both the Scikit classifier tree and the implemented version of the tree provide a similar accuracy greater than 90%, having a minimum difference between both models.
As one can observe on the following confusion matrix, the model does make accurate predictions, leaving a couple to be close but wrong.



Confusion Matrix for Decision Tree Classifier implementation

Moving on to the regression models, the linear regression is faster at converging and reaching a minimum, however the decision tree regressor is much slower but comes to having less error at making predictions as shown in this sample of predictions between both models. Below is an example of the expected value (real value), followed by the prediction made by the linear regression model and the prediction from the decision tree.

```
Expected -> 108.251 , Predicted LR Value -> 103.958 , Predicted Regression Tree -> 108.316
Expected -> 89.982 , Predicted LR Value -> 91.212 , Predicted Regression Tree -> 90.000
Expected -> 80.000 , Predicted LR Value -> 76.348 , Predicted Regression Tree -> 79.494
Expected -> 75.094 , Predicted LR Value -> 75.131 , Predicted Regression Tree -> 75.384
Expected -> 112.089 , Predicted LR Value -> 114.955 , Predicted Regression Tree -> 112.226
Expected -> 49.928 , Predicted LR Value -> 55.349 , Predicted Regression Tree -> 49.714
Expected -> 80.000 , Predicted LR Value -> 75.646 , Predicted Regression Tree -> 77.133
Expected -> 79.697 , Predicted LR Value -> 77.364 , Predicted Regression Tree -> 77.133
Expected -> 73.000 , Predicted LR Value -> 72.560 , Predicted Regression Tree -> 66.500
Expected -> 102.556 , Predicted LR Value -> 113.090 , Predicted Regression Tree -> 102.363
Expected -> 104.921 , Predicted LR Value -> 114.270 , Predicted Regression Tree -> 109.960
Expected -> 64.696 , Predicted LR Value -> 65.616 , Predicted Regression Tree -> 65.000
Expected -> 60.000 , Predicted LR Value -> 63.237 , Predicted Regression Tree -> 55.000
Expected -> 42.006 , Predicted LR Value -> 38.294 , Predicted Regression Tree -> 42.542
Expected -> 80.000 , Predicted LR Value -> 82.837 , Predicted Regression Tree -> 80.000
Expected -> 94.744 , Predicted LR Value -> 94.711 , Predicted Regression Tree -> 98.441
Expected -> 133.666 , Predicted LR Value -> 128.284 , Predicted Regression Tree -> 133.747
Expected -> 78.090 , Predicted LR Value -> 81.359 , Predicted Regression Tree -> 80.351
Expected -> 40.343 , Predicted LR Value -> 32.010 , Predicted Regression Tree -> 44.642
Expected -> 70.000 , Predicted LR Value -> 70.231 , Predicted Regression Tree -> 70.000
Expected -> 153.149 , Predicted LR Value -> 133.310 , Predicted Regression Tree -> 153.960
Expected -> 83.000 , Predicted LR Value -> 90.768 , Predicted Regression Tree -> 78.000
Expected -> 65.000 , Predicted LR Value -> 67.023 , Predicted Regression Tree -> 67.194
Expected -> 55.010 , Predicted LR Value -> 57.847 , Predicted Regression Tree -> 56.266
Expected -> 87.000 , Predicted LR Value -> 84.519 , Predicted Regression Tree -> 86.145
Expected -> 52.094 , Predicted LR Value -> 55.525 , Predicted Regression Tree -> 51.553
Expected -> 43.088 , Predicted LR Value -> 40.961 , Predicted Regression Tree -> 44.642
Expected -> 79.545 , Predicted LR Value -> 83.225 , Predicted Regression Tree -> 79.470
Expected -> 121.042 , Predicted LR Value -> 121.918 , Predicted Regression Tree -> 120.975
```

In average, the accuracy of the tree is lower, but in contrast to the linear model, it fits edge cases quite better as seen in the picture, thus achieving a better accuracy overall.

Also, below it can be observed through a confusion matrix that the model accurately predicts the true values and fails to do in a couple of instances.

Enrique Posada Lozano A01700711

Conclusion :

In conclusion, a decision tree such as CART is really useful when a dataset for tasks that may need to handle both regression and classification tasks, and even then it still is useful in handling these tasks independently. However, implementing these models from scratch does offer useful insight into some factors one may need to consider a model over the other (since Scikit either way has fast implementations of these models). A decision tree for classification (if not considering a random forest), is faster and easier to handle label encoding than other classifiers such as a logistic regression since it requires less calculation overall. However, in a regression tree, if a max depth is not defined then a tree appears to take much more time building just to have a little better accuracy than other types of models. Another useful insight as a conclusion is to not underestimate the importance of decimals (or having many decimals) even in features where they might not look important, because there might be the case in which analyzing those cases is what may provide far better accuracy in test and validation results.

References :

- https://towardsdatascience.com/is-random-forest-better-than-logistic-regression-a-comparison-7a0f068963e4
- https://www.sciencedirect.com/science/article/pii/S2352340919306985?via%3Dihub
- https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35
- https://scikit-learn.org/stable/modules/tree.html
- https://www.vernier.com/til/1014
- https://www.who.int/health-topics/obesity#tab=tab_1
- https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight
- https://www.thelancet.com/campaigns/kidney/updates/obesity-and-overweight-populations-in-latin-america