**DOKUZ EYLÜL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CME 4403 INTRODUCTION TO MACHINE LEARNING TERM PROJECT**

**by**

**Ekin UZUNBAZ**

**Yusuf Orhan OKKABAS**

**Advisor**

**Assoc. Prof. Dr. Zerrin IŞIK**

**December, 2022**

**İZMİR**

# CHAPTER ONE

INTRODUCTION

Obesity is a major health problem that affects millions of people around the world. It is associated with a range of negative health outcomes, including heart disease, diabetes, and stroke. Predicting obesity can help to identify individuals who are at risk and allow them to take preventive measures to improve their health.

In this project, we used the estimation of obesity levels based on eating habits and physical condition dataset to predict obesity using machine learning techniques. The dataset contains information on the eating habits and physical conditions of individuals, as well as their body mass index (bmi).

We used a combination of K-Nearest Neighbor (KNN), Random Forest, and Multinomial Logistic Regression machine learning algorithms to predict obesity. We also performed feature selection, discretization, and dummy attribute creation to prepare the data for analysis.

To evaluate the performance of our models, we used accuracy, precision, and recall as evaluation metrics [[1]](https://hasty.ai/docs/mp-wiki/metrics/accuracy) [[2]](https://www.learndatasci.com/glossary/precision-and-recall/). The results of our analysis showed that the Multinomial Logistic Regression model had the highest accuracy, precision, and recall, followed by the Random Forest model.

Overall, our project demonstrated that machine learning techniques can be effective for predicting obesity. In the future, it would be interesting to explore other techniques and datasets to see if they can further improve the accuracy of obesity prediction.

# CHAPTER TWO

DATASET PRE-PROCESSING AND FEATURE EXPLORATION

* 1. **Cleaning**

Before the data could be used for analysis, it was necessary to perform some preprocessing and cleaning steps. The first step was to remove any rows with missing values.  Since we do not have the N/A values on the data set. We did not any cleaning processes.

* 1. **Feature Importance**

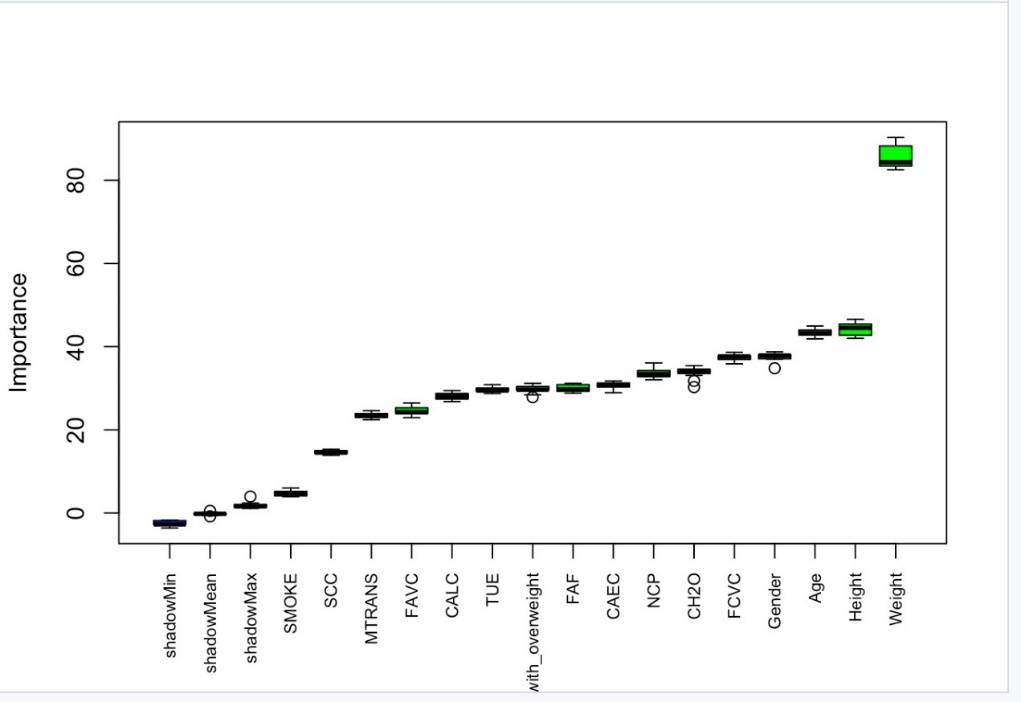
After the data was cleaned, we performed feature selection to select the most relevant variables for our model. We used the Boruta feature selection method, Boruta is a feature selection method for use in machine learning [[3]](https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/). It was developed as an alternative to traditional filter and wrapper methods, which often rely on statistical tests or heuristics to select features. Boruta works by creating "shadow" copies of all the features in the dataset, and then training a classifier on the original and shadow features. The shadow features are randomly shuffled versions of the original features, and are used to create a baseline for feature importance. The original features are then compared to the shadow features, and those that are significantly more important are kept, while the rest are removed.

Our dataset has no tentative and rejected features, all attributes are confirmed on Boruta feature selection. Here is the output and plot of Boruta feature selection method:

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Output of Boruta



Plot of Boruta

* 1. **Discretization**

To prepare the continuous variables in our dataset for analysis, we performed discretization to convert them into categorical ones. We used a combination of fixed and interval binning methods to divide the data into bins.

Fixed binning involves dividing the data into a fixed number of bins, regardless of the distribution of the data. For example, we might divide the data into 4 bins of equal width.

 Interval binning involves dividing the data into bins of equal size based on the range of the data. For example, we might divide the data into bins that each contain 25% of the data points.

We chose the number of bins and the binning method based on the characteristics of the data and the goals of the analysis. In order to define fixed interval for height, we used average weight values according to countries and genders that is provided in [[4]](https://tr.wikipedia.org/wiki/%C3%9Clkeye_g%C3%B6re_ortalama_insan_boyu). Then we defined interval for heights by using limit variability that is provided in [[5]](https://www.researchgate.net/publication/344321092_Examination_of_the_Level_of_Conus_Medullaris_Termination_Using_Magnetic_Resonance_Imaging). For the fixed interval for weight, we calculated BMI(Body Mass Index) for each data and labeled according to that [[6]](https://www.nhs.uk/common-health-questions/lifestyle/what-is-the-body-mass-index-bmi/).

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

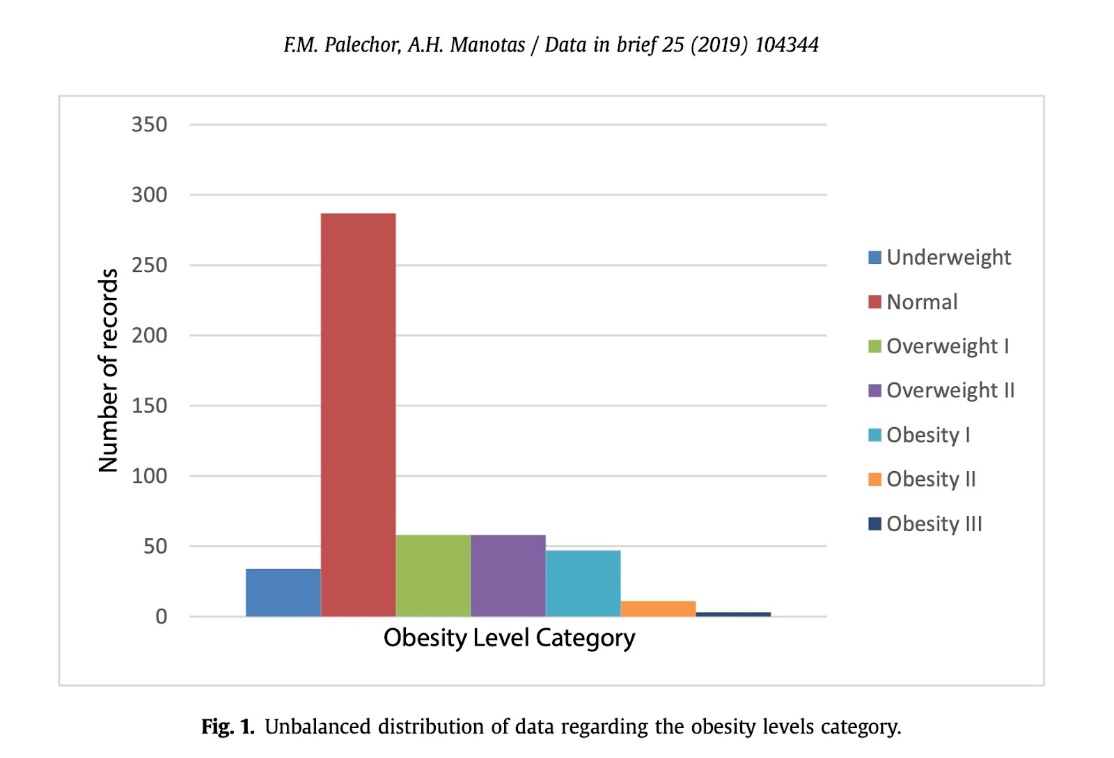
Average BMI values

* 1. **Dummy Attributes**

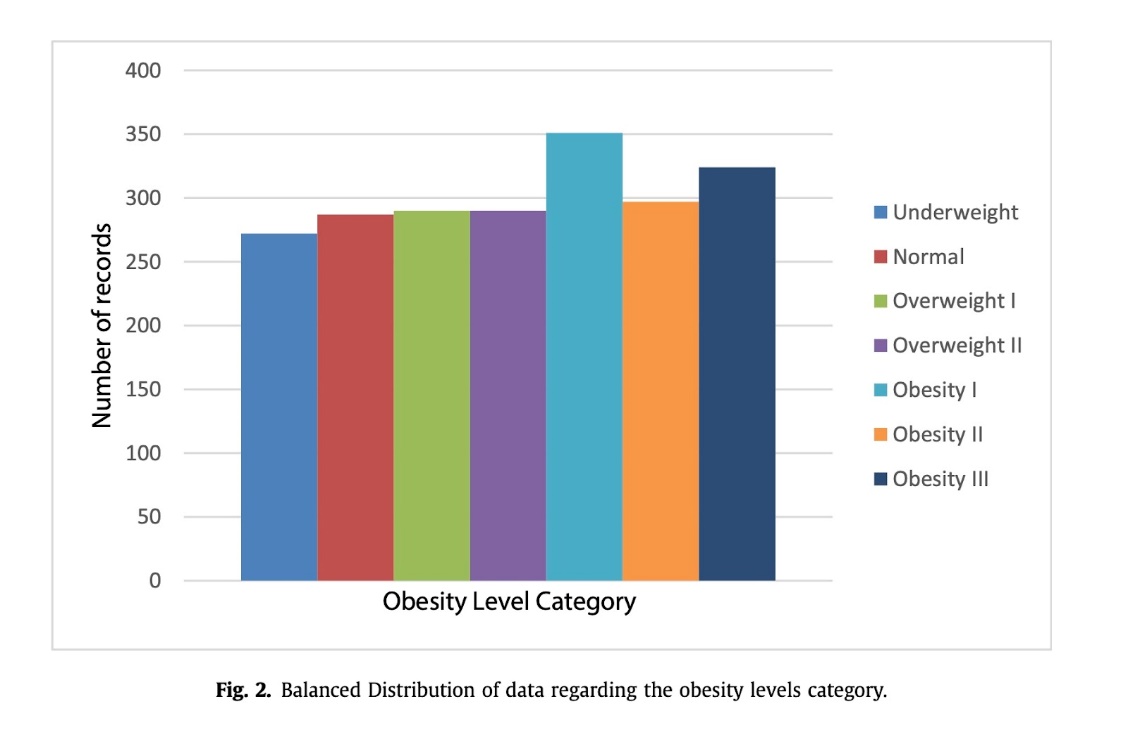
During converting categorical values into numerical values, we used if-else statements for categories with two levels and dummy attributes for categories with more than two levels. Dummy attribute, which is a numerical representation of the categorical variable that encodes each category as a separate binary value (0 or 1) [[7]](https://www.marsja.se/create-dummy-variables-in-r/). For example, if a categorical variable has three categories (e.g., 'low', 'medium', and 'high'), three dummy attributes could be created to represent each category, with a value of 1 indicating that the instance belongs to that category and a value of 0 indicating that it does not.

On the other hand, there are some trade-offs when using dummy attributes. Dummy attributes are easy to create and interpret but creating separate dummy attributes for each category in a categorical variable can significantly increase the number of features in the dataset.

* 1. **Distributions**

****

After the labeling process was finished, the categories of obesity levels were unbalanced (as shown in Fig. 1), and this presented a learning problem for the data mining methods, since it would learn to identify correctly the category with most records compared with the categories with less data. In [[8]](https://www.jair.org/index.php/jair/article/view/10302), you can see a dataset is unbalanced if the classification categories are not represented equally (Palechor, de la Hoz Manotas, 2019).



After the balancing class problem was identified, synthetic data was generated, up to 77% of the data, using the tool Weka and the filter SMOTE proposed by [[8]](https://www.jair.org/index.php/jair/article/view/10302). The filter required to indicate the class for generation of synthetic data, the number of nearest neighbors used, the percentage that you need to increase the selected class and the random seed used for random sampling. Other aspects analyzed were the identification of atypical and missing data. Finally, after the filter was applied to each category, the final result were 2111 records. Next, in Fig. 2 you can see the final distribution of the data after the balancing process was completed (Palechor, de la Hoz Manotas, 2019).

# CHAPTER three

IMPLEMENTATION DETAILS OF MACHINE LEARNING MODELS

We used a combination of K Nearest Neighbor (KNN), Random Forest, and Multinomial Logistic Regression machine learning algorithms to predict obesity. We also performed cross validation to evaluate the performance of the models. We performed cross validation for 20 iterations. Because Random Forest algorithm is used and the program was taking a long time to work.

* 1. **K-Nearest Neighbor (KNN)**

K nearest neighbor (KNN) is a type of instance-based learning, which means that it stores all of the training data and predicts the label of a new instance by finding the most similar instances in the training set and taking a majority vote. KNN is simple to implement and can be effective for a wide range of classification and regression tasks, but it can be computationally expensive and may not scale well to large datasets. The function that we used uses Euclidean Distance as distance metric.

Since KNN naturally works with numerical values, we converted our dataset into full numerical version thanks to “Dummy Attributes” and if-else statements. After converting operations are done, we normalized the data because there are huge differences between columns’ values [[9]](https://www.geeksforgeeks.org/how-to-normalize-and-standardize-data-in-r/). Normalization is important for the KNN model because the distance metric is sensitive to the scale of the features.

We changed the KNN model’s “K” parameter several times. We calculated accuracy, precision and recall values in order to evaluate the model. While doing all this things, we applied cross validation as well.

* 1. **Random Forest**

Random forest is an ensemble learning method that combines the predictions of multiple decision trees trained on different subsets of the training data. Each tree in the ensemble is trained on a randomly selected subset of the features and a randomly selected subset of the training examples. The final prediction is made by averaging the predictions of all of the trees in the ensemble. Random forests are generally robust to noise and outliers in the data, and they can handle a large number of features without the need for feature selection.

We used Random Forest algorithm together with full categorical version of our dataset [[10]](https://www.rdocumentation.org/packages/randomForest/versions/4.7-1.1/topics/randomForest). We converted all the descriptive features in the dataset into categorical variables. This converting process is called “Discretization”.

We changed the Random Forest model’s “ntree” parameter several times. This parameter means that the model will use how many decision trees to make predictions. We calculated accuracy, precision and recall values in order to evaluate the model. While doing all this things, we applied cross validation as well.

* 1. **Multinomial Logistic Regression**

Firstly, Logistic regression is a classification algorithm that is used to predict the probability of a binary outcome (e.g., 0 or 1, true or false). It is based on the idea of finding the best linear boundary between the positive and negative classes. Logistic regression is widely used and is relatively simple to implement, but it assumes that the data is linearly separable and that the relationship between the features and the outcome is linear. As for the main point; Multinomial logistic regression is a variant of logistic regression that is used to predict a multi-class outcome, rather than a binary outcome [[11]](https://www.r-bloggers.com/2020/05/multinomial-logistic-regression-with-r/). In other words, it is used to classify instances into one of three or more classes.

Since Multinomial Logistic Regression naturally works with numerical values, we converted our dataset into full numerical version thanks to dummy attributes and if-elese statements. Then, we normalized the data because there are huge differences between columns’ values. Multinomial Logistic Regression is sensitive to the scale of the features, so it is generally recommended to normalize the data before applying Multinomial Logistic Regression. We have considered this proposal and applied normalization.

We changed the Multinomial Logistic Regression model’s “maxit” parameter several times. We calculated accuracy, precision and recall values while changing this parameter in order to evaluate the model. While doing all this things, we applied cross validation as well.

* 1. **Cross-Validation**

Cross-validation is a technique that is used to evaluate the performance of a machine learning model by training it on a subset of the available data and testing it on the remaining data. It is a useful way to assess the generalization performance of a model and to avoid overfitting, which is when a model performs well on the training data but poorly on new, unseen data.

We used a for loop to implement cross-validation by taking random samples from the dataset. This is a common approach used among cross validation techniques.

# CHAPTER four

EXPERIMENTAL RESULTS

**4.1 Tables for The Average Values of Evaluation Metrics for Each Model**

|  |  |  |  |
| --- | --- | --- | --- |
| Random Forest | Accuracy | Precision | Recall |
| ntree=300 | 0.940625 | 0.940501620511093 | 0.941323223579002 |
| ntree=400 | 0.937310606060606 | 0.936742413597998 | 0.93814498426813 |
| ntree=500 | 0.939299242424242 | 0.93887224331851 | 0.940393612667492 |

|  |  |  |  |
| --- | --- | --- | --- |
| kNearestNeighbor | Accuracy | Precision | Recall |
| kNN=5 | 0.773958333333333 | 0.763635197607132 | 0.770930747904391 |
| kNN=7 | 0.758996212121212 | 0.749530695853607 | 0.756610627805863 |
| kNN=9 | 0.753977272727273 | 0.739281358614951 | 0.747014249814305 |

|  |  |  |  |
| --- | --- | --- | --- |
| Logistic Regression | accuracy | precısıon | recall |
| maxıt=50 | 0.9102272727272 | 0.909657316124643 | 0.909292599925641 |
| maxıt=75 | 0.9475378787878 | 0.946405716604258 | 0.947229750299643 |
| maxıt=100 | 0.9517992424242 | 0.950338113350192 | 0.950490620289043 |

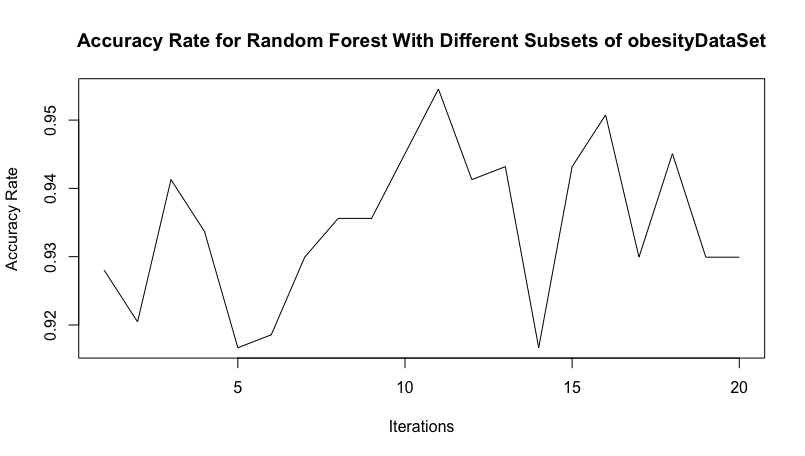
**4.2 Tables for The Highest Values of Evaluation Metrics for Each Model**

|  |  |  |  |
| --- | --- | --- | --- |
| Random Forest | Accuracy | Precision | Recall |
| ntree=300 | 0.954545 | 0.953512 | 0.955874 |
| ntree=400 | 0.950758 | 0.948235 | 0.951714 |
| ntree=500 | 0.954545 | 0.954225 | 0.954830 |

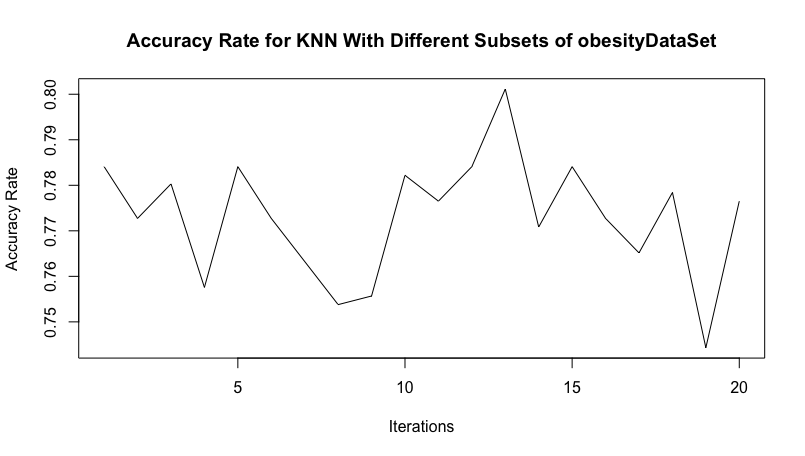
|  |  |  |  |
| --- | --- | --- | --- |
| kNearestNeighbor | Accuracy | Precision | Recall |
| kNN=5 | 0.810606 | 0.795159 | 0.802111 |
| kNN=7 | 0.803460 | 0.800121 | 0.800620 |
| kNN=9 | 0.770833 | 0.763051 | 0.767988 |

|  |  |  |  |
| --- | --- | --- | --- |
| Logistic Regression | accuracy | precısıon | recall |
| maxıt=50 | 0.950758 | 0.947618 | 0.950568 |
| maxıt=75 | 0.962121 | 0.960707 | 0.960685 |
| maxıt=100 | 0.965909 | 0.965699 | 0.965004 |

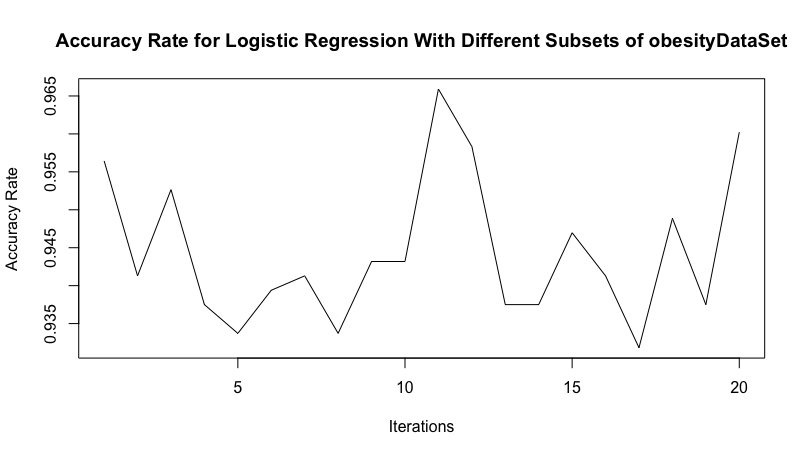
**4.3 Plots for Highest Accuracy Values of Each Model**

****

**Highest Accuracy for Random Forest When Ntree=500**

****

**Highest Accuracy for Knn When K=5**

****

**Highest Accuracy Rate for Multinomial Logistic Regression When Maxit=100**

CHAPTER FIVE

CONCLUSION

In this project, we used machine learning techniques to predict obesity levels based on eating habits and physical condition. We explored a variety of approaches, including feature selection, discretization, and training different models such as K-Nearest Neighbor (KNN), Random Forest, and Multinomial Logistic Regression.

We found that the Multinomial Logistic Regression model performed the best, achieving an accuracy of 0.965909, a precision of 0.965699, and a recall of 0.965004 on the test data.

Overall, our results suggest that it is possible to predict obesity levels with a high degree of accuracy using machine learning techniques. However, it is important to carefully select and pre-process the features, and to choose the appropriate model based on the characteristics of the data and the goals of the analysis.

We hope that our findings will be useful for researchers and practitioners working on obesity prevention and management, and that they will contribute to a better understanding of the factors that influence obesity levels.

# CHAPTER SIX

REFERENCES

[1] https://hasty.ai/docs/mp-wiki/metrics/accuracy

[2] https://www.learndatasci.com/glossary/precision-and-recall/

[3]https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/

[4] https://tr.wikipedia.org/wiki/%C3%9Clkeye\_g%C3%B6re\_ortalama\_insan\_boyu

[5]https://www.researchgate.net/publication/344321092\_Examination\_of\_the\_Level\_of\_Conus\_Medullaris\_Termination\_Using\_Magnetic\_Resonance\_Imaging

[6] https://www.nhs.uk/common-health-questions/lifestyle/what-is-the-body-mass-index-bmi/

[7] https://www.marsja.se/create-dummy-variables-in-r/

[8] https://www.jair.org/index.php/jair/article/view/10302

[9] https://www.geeksforgeeks.org/how-to-normalize-and-standardize-data-in-r/

[10]https://www.rdocumentation.org/packages/randomForest/versions/4.7-1.1/topics/randomForest

[11] https://www.r-bloggers.com/2020/05/multinomial-logistic-regression-with-r/

**Related Article for Our Dataset**

 Palechor, F. M., & de la Hoz Manotas, A. (2019). Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. Data in brief, 25, 104344.