BINF 5527 MACHINE LEARNING IN BIOINFORMATICS FINAL: OBESITY LEVEL ESTIMATION SOFTWARE BASED ON DECISION TREES

Mehmet Cihan Sakman
sakmancihan@gmail.com

Friday 10th September, 2021

Abstract

In this project *Obesity Level Estimation Software based on Decision Trees* article published by Eduardo De-La-Hoz-Correa, Fabio E. Mendoza-Palechor, Alexis De-La-Hoz-Manotas, Roberto C. Morales-Ortega and Sánchez Hernández Beatriz Adriana has been tried to replicated. Obesity has become a global epidemic that has doubled since 1980, authors handled that problem authors applied the SEMMA data mining methodology, to select, explore and model the data set and then three methods were selected: Decision trees, Bayesian networks (Naïve Bayes) and Logistic Regression, obtaining the best results with Decision trees based on the metrics: Precision, recall, TP Rate and FP Rate.

1 Introduction

The World Health Organization (WHO) (OMS, 2016), describes obesity and overweight as excessive fat accumulation in certain body areas that can be harmful for health, the number of people that suffers from obesity has doubled since 1980 and also in 2014 more than 1900 million adults, 18 years old or older, are suffering from alteration of their weight. Obesity is a public health problem worldwide and it can emerge in adults, teens and children.

Several authors have studies to analyze the disease and generate web tools to calculate the obesity level of a person, nevertheless such tools are limited to the calculation of the body mass index, omitting relevant factors such as family background and time dedicated to. Based on this, the authors considered an intelligent tool was needed to be able to detect obesity levels on people more efficiently. In this study, volunteers were surveyed with a series of questions to identify their obesity level, considering several factors such as age, weight, sex, physical activity frequency, fast food intake and others, that could help to describe the behavior of obese people.

The methods and techniques used in the experimentation process of this study, refer to Decision Trees, Naïve Bayes and Logistic Regression.

Necessary information and the data itself can be achieved from: `https://archive.ics.uci.edu/ml/datasets/Estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition+`

2 Description of Methods

In this study, the stages based on the SEMMA methodology performed. First, the dataset creation proceeded, from the information collected by the survey, as described in *Figure 1*. After the

dataset creation, the data have been validated, looking for missing values, atypical data, and the correlation level between variables, which is lower than 0.5, so can be sure that the stored data and the basis for the software implementation and the data mining methods are correct. Once the dataset was validated and prepared, the data mining techniques and methods were applied.

Attributes	Values
Sex	H: Male
	M: Female
Age	Integer Numeric Values
Height	Integer Numeric Values (Mt)
Weight	Integer Numeric Values (Kg)
Family with overweight / Obesity	Yes
	No
Fast Food Intake	Yes
	No
Vegetables Consumption Frequency	S: Always
	A: Sometimes
	CN: Rarely
Number of main meals daily	1 to 2: UD
	3: TR
	More than 3: MT
Food intake between meals	S: Always
	CS: Usually
	A: Sometimes
	CN: Rarely
Smoking	Yes
	No
Liquid intake daily	MU: Less than one liter
	UAD: Between 1 and 2 liters
	MD: More than 2 liters
Calories Consumption Calculation	Yes
	No
Physical Activity	UOD: 1 to 2 days
	TAC: 3 to 4 days
	COS: 5 to 6 days
	NO: No physical activity
Schedule dedicated to technology	CAD: 0 to 2 hours
	TAC: 3 to 5 hours
	MC: More than 5 hours
Alcohol consumption	NO: No consumo de alcohol
	CF: Rarely
	S: Weekly
	D: Daily
Type of Transportation used	TP: Public transportation
	MTA: Motorbike
	BTA: Bike
	CA: Walking
	AU: Automobile
IMC	WHO Classification
Vulnerable	Based on the WHO Classification

Figure 1: Dataset Description

In this study, the methods used were **Decision Trees, Bayesian Networks (Naïve Bayes)**, and **Logistic Regression**. To validate the model and selecting the best technique, the Precision metrics Recall, TP Rate, and FP Rate were used. For the training process, cross-validation was used, part of the data for training and other part for testing to guarantee optimal results and avoiding over-training issues. The proposed model considers classes or categories, the values of *underweight*, *normal*, *overweight*, *obesity level I*, *obesity level II* and *obesity level III*.

3 Description of Data

The dataset were updated after the published of this version. Before the updated version there were 712 records. After the update %67 of the data generated synthetically by the SMOTE filter, and %33 of the data was collected directly from the survey. The dataset contains 17 attributes and 2111 records.

List of important features for Obesity levels based on eating habits and physical condition data set:

- **Gender** : Gender information
- **Age** : Age
- **Height** : Integer Numeric Values (Mt)
- **Weight** : Integer Numeric Values (kg)
- **family_history_with_overweight** : Has a family member suffered or suffers from overweight.
- **FAVC** : Do you eat high caloric food frequently?
- **FCVC** : Do you usually eat vegetables in yor meals?
- **NCP** : How many main meals do you have?
- **CAEC** : Do you eat any food between meals?
- **SMOKE**: Do you smoke?
- **CH2O** : How much water do you drink daily?
- **SCC** : Do you monitor the calories you eat daily?
- **FAF** : How often do you have physical activity?
- **TUE** : How much time do you use technological devices?
- **CALC**: How often do you drink alcohol?
- **MTRANS**: Which transportation do you usually use?
- **NObeyesdad** : The target value of obesity level.

Further information could be found in the *Table 1*.

4 Expected Results in Paper

Based on the data the technique with the best results was Decision Trees. The technique also obtained better results than the values from techniques such as Bayesian Networks and Logistic Regression.

The results for **Decision Trees**: *Precision:* 97.4%, *Recall:* 97.8%, *TP Rate:* 97.8%, *FP Rate:* 0.2%.
The results for **Naive Baes**: *Precision:* 90.1%, *Recall:* 91.1%, *TP Rate:* 91.1%, *FP Rate:* 6.0%.
The resulst for **Logistic Regression**: *Precision:* 90.4%, *Recall:* 91.6%, *TP Rate:* 91.6%, *FP Rate:* 4.1%.

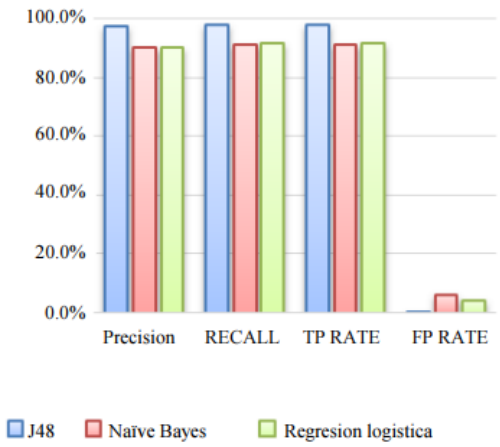

Figure 2: Algorithms Results

5 Preprocessing Steps

Data set were already preprocessed by the authors as mentioned in *Description of Methods* part. But there were still some categorical features and these features were in the character format. After that to improve the article work some feature selection methods have been tried to handled and finally parameter tuning with Grid Search applied.

5.1 Encoding

5.1.1 Ordinal Features

In dataset, there are three ordinal columns as follow: CAEC, CALC and NObeyesdad. These columns mapped as below.

```
# We'll apply mapping to the ordinal columns manually.
#CAEC and CALC have the same orders
CAEC_CALC_ord_map = {'no': 0, 'Sometimes': 1, 'Frequently': 2, 'Always': 3}
data['CAEC'] = data['CAEC'].map(CAEC_CALC_ord_map)
data['CALC'] = data['CALC'].map(CAEC_CALC_ord_map)


obesity_level_map = {'Insufficient_Weight': 0, 'Normal_Weight': 1, 'Overweight_Level_I': 2, 'Overweight_Level_II': 2,
                'Obesity_Type_I':3, 'Obesity_Type_II':4, 'Obesity_Type_III':5}

data['NObeyesdad'] = data['NObeyesdad'].map(obesity_level_map)
```

Figure 3: Mapping

5.1.2 One Hot Encoding

In dataset, there is only one column which has nominal categorical values called *MTRANS* with following categories 'Public Transportation', 'Walking', 'Automobile', 'Motorbike' and 'Bike'. One Hot Encoding have been applied to that feature by using pandas' *get_dummies* feature.

5.1.3 Binary Columns

Label Encoding has been applied to the binary columns with character type values.

5.1.4 Rounding Problem

In the updated version of the article, SMOTE has been applied to data to produce new variables avoid to the unbalanced class problem. That approach produces new variables such as float variables for age or some other categorical columns which we don't want to have that type variables. Therefore, the rounding approach applied to these features to transform the values into integer types.

5.2 Results and Comparison

Due to the dataset used in that article has been changed by applying SMOTE our results may differ from the original article. Also, some criteria haven't mentioned in the article by authors and we tried to find the best parameter settings and also try to split training and testing data by ourselves.

Our results obtained by using Decision Trees, Logistic Regression and Naive Bayes can be find below figures.

	class	Precision	Recall	TP	FP
0	Logistic Regression	0.720702	0.731861	0.731861	0.268139
1	DecisionTree	0.929981	0.930599	0.930599	0.069401
2	Naive Bayes	0.683997	0.580442	0.580442	0.419558

Figure 4: Results Table

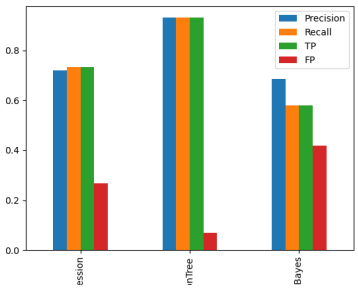

Figure 5: Results Figure

When the resulst compare with the original article there is a huge difference between the Naive Baes scores and Logistic Regression scores. The general scores for the Decision Tree algorithm almost same and very close. But these results obtained by default parameter settings. We'll try to develop these scores in the following sections.

5.3 Feature Selection

5.3.1 Feature Selection in Numeric Columns

Dataset has only three numeric columns but the feature selection algorithm still applied and tries to find if there is an unnecessary column or not. In this section ANOVA based on selection

applied and numeric columns eliminated transform into one by one. But the feature selection algorithm for numeric columns didn't improve the model.

5.3.2 Feature Selection in Categorical Columns

There are two popular feature selection techniques that can be used for categorical input data and a categorical (class) target variable. They are: *Chi-Squared Statistic* and *Mutual Information Statistic*. Pearson's chi-squared statistical hypothesis test is an example of a test for independence between categorical variables. The results of this test can be used for feature selection, where those features that are independent of the target variable can be removed from the dataset. On the other hand, Mutual information from the field of information theory is the application of information gain (typically used in the construction of decision trees) to feature selection. Mutual information is calculated between two variables and measures the reduction in uncertainty for one variable given a known value of the other variable.

These two popular feature selection techniques have been applied and conclude that the Mutual Information Statistic was more beneficial than the Chi-squared statistic. The categorical columns have been eliminated into three columns and obtained the best results as follows.

	class	Precision	Recall	TP	FP
0	Logistic Regression	0.720702	0.731861	0.731861	0.268139
1	DecisionTree	0.929981	0.930599	0.930599	0.069401
2	Naive Bayes	0.683997	0.580442	0.580442	0.419558

Figure 6: Before Feature Selection

	class	Precision	Recall	TP	FP
0	Logistic Regression	0.767671	0.769716	0.769716	0.230284
1	DecisionTree	0.975075	0.974763	0.974763	0.025237
2	Naive Bayes	0.643853	0.659306	0.659306	0.340694

Figure 7: After Feature Selection

As we can see above, there is a significant improvement for the Logistic Regression and Decision Tree but on the other hand that feature selection technique negatively affected the Naive Bayes algorithm

5.4 Model Hyperparameter Optimization

Machine learning models have hyperparameters. Hyperparameters are points of choice or configuration that allow a machine learning model to be customized for a specific task or dataset. Machine learning models also have parameters, which are the internal coefficients set by training or optimizing the model on a training dataset. Parameters are different from hyperparameters. Parameters are learned automatically; hyperparameters are set manually to help guide the learning process.

Help of the **model_selection** library of scikit-learn we can use the GridSearchCV to find best hyperparameters. GridSearchCV, evaluate models for a given hyperparameter vector using cross-validation, hence the "CV" suffix of each class name. GridSearchCV requires two arguments. The first is the model that you are optimizing. This is an instance of the model with values of hyperparameters set that you want to optimize. The second is the search space. This is defined as a dictionary where the names are the hyperparameter arguments to the model and the values are discrete values or a distribution of values to sample in the case of a random search. Here is an example from our code, we're trying to find best hyperparameters for the Logistic Regression.

After applying GridSearchCV for all learning algorithms we try to come up with best hyperparameters and plot the results again.

```
parameters = [{'solver': ['newton-cg'],'C': [1,10], 'multi_class': ['auto', 'ovr','multinomial']},
              {'solver': ['lbfgs'], 'C': [1,10], 'multi_class': ['auto', 'ovr','multinomial']},
              {'solver': ['liblinear'], 'multi_class': ['auto', 'ovr'],'penalty': ['l1', 'l2', 'elasticnet'],'C': [1,10]},
              {'solver': ['sag'], 'multi_class': ['auto', 'ovr','multinomial'], 'C': [1,10]},
              {'solver': ['saga'], 'multi_class': ['auto', 'ovr','multinomial'], 'C': [1,10]}]

classification_method = LogisticRegression()
classification_method= GridSearchCV(classification_method , parameters, cv=10)
classification_method.fit(X_train, y_train)
```

Figure 8: GridSearchCV for Logistic Regression

	class	Precision	Recall	TP	FP
0	Logistic Regression	0.767671	0.769716	0.769716	0.230284
1	DecisionTree	0.975875	0.974763	0.974763	0.025237
2	Naive Bayes	0.643853	0.659306	0.659306	0.340694

Figure 9: Before GridSearch

	class	Precision	Recall	TP	FP
0	Logistic Regression	0.911791	0.911672	0.911672	0.088328
1	DecisionTree	0.969087	0.968454	0.968454	0.031546
2	Naive Bayes	0.675486	0.687697	0.687697	0.312303

Figure 10: After GridSearch

6 Conclusion

In conclusion, in this article authors try to predict the obesity level of a person. The dataset was almost fully preprocessed and ready for applying to necessary algorithms. After just applying the OneHotEncoding, LabelEncoding, and mapping methods for the ordinal columns data was fully preprocessed. Due to our dataset was not that large the Decision Tree proved its effect once more. In the beginning, my scores are really different from the scores in the article. One reason for that was the update on the dataset. New variables were produced by using SMOTE filter. The other reason was the feature selection and hyperparameter tuning. After dropping the relevant features we obtained better results for Logistic Regression and Decision Tree model. When we applied the GridSearchCV to find the best hyperparameters we obtained much better results than the beginning. The logistic Regression model has been improved by almost %15. If there would be a clearer explanation of the article I believe that we could achieve much higher accuracy scores and improve the current algorithm better.