Predicting Obesity based on Lifestyle Choices

Building a Random Forest Model that can Predict Probability of Becoming Obese Depending on Daily Habits and Choices

Ryan Divigalpitiya
CS 9637 Introduction to Data Science
Western University

Abstract

Background: Obesity is an ever rising concern across numerous counties and, in the US, contributes to an estimated 112,000 deaths per year, all of which were preventable had they taken the necessary steps to reduce obesity onset. These increasing rates of obesity are also reported on a global scale by the World Health Organization, leaving many health professionals to view obesity as a very serious epidemic. Being able to predict the probability of succumbing to obesity based on one's current lifestyle choices may allow individuals to take action and correct their habits in order to avoid obesity and all of its potential health ailments that may come with it. **Objective:** To develop a predictive model that can show the probability of someone falling into a class of obesity given that they maintain their current lifestyle choices and habits. Our motivation behind building this model is to hopefully provide incentives for someone using this model to correct their lifestyle choices if the model predicts that they may fall into a class of obesity given their inputed lifestyle. **Methods:** Obesity data used in training our models came from data containing estimations of obesity levels in people, ages from 14 to 61, from the countries of Mexico, Peru and Colombia, with diverse eating habits and physical condition. The data consisted of 2111 rows and 16 features. The target variable had 7 possible classes: Obesity I, Obesity II, Obesity III, Overweight I, Overweight II, Normal Weight, and Insufficient Weight. Data was split into training and testing data. Three supervised machine learning models were trained and evaluated (Logistic Regression, Random Forest, XGBoosting). Automated hyper-parameter tuning and feature-space tuning were employed to improve accuracy of our prediction models. **Results:** Selected model was a Random Forest-based Multi-Classifier that was hyper-parameter tuned and feature-tuned. Selected model achieved an accuracy of 86.55% on the test set. Averaged Precision score — averaged across the 7 classes — was 0.8; Recall score was 0.87, and F1-score was 0.86. Averaged Sensitivity and Specificity scores were 0.87 and 0.98. Averaged Balanced Accuracy was 0.92. Macro-Averaged AUROC was 0.98. **Conclusion:** We built a moderately accurate prediction model that classifies people's risk of falling into a specific class of obesity (along with a normal weight class and underweight class) given that they maintain their existing lifestyle choices. Tuning strategies provided a decent improvement in performance. We achieved decent sensitivity and high specificity. Performance of our model is believed to be held back by the relatively small size of our dataset (especially when considering we have 7 classes to learn from) and, for any future work, acquisition of a larger dataset would be utmost priority to improve our model. Some limitations of our model are also discussed.

Introduction

Obesity is defined as abnormal or excessive fat accumulation that may present health risks for the diagnosed patient [1]. The prevalence of those diagnosed with obesity in recent decades have dramatically increased. For example, in the United States (US), obesity rates in children have tripled and rates in adults have doubled [2, 3]. As a result of these dramatic increases in obesity rates, obesity contributes to an estimated 112,000 deaths per year in the US, all of them preventable had they taken the necessary steps to reduce obesity onset [2]. These rates are also reported to be similar on a global scale, not just in the US, as reported by the World Health Organization, and, thus, obesity rates are viewed by health professionals as a very serious epidemic affecting the human race [4, 5]. This report will focus on predicting the obesity level based on a person's lifestyle choices if they were to maintain those lifestyle choices; examples of lifestyle choices include choice of transportation, smoking and how much water they drank. Our motivation behind this project is as such: an obesity predictive model would be of value if a person, who is not yet classified as obese, wanted to know the probability of them falling into a class of obesity given that they maintained their current lifestyle choices (there are multiple classes of obesity which will be further explained in the Data section of this report). In other words, a person may want to know: *will my current lifestyle choices lead to obesity or not?* What would make such a model even more relevant, is its potential use in a clinical setting: a doctor or health professional could ask a series of questions during a physical exam, and then input the patient's answers to those questions into our model, outputting a probability of how likely that person will eventually fall into a class of obesity assuming that patient maintains their existing lifestyle choices that they answered. Hopefully, such insight may motivate a patient to make changes to their lifestyle *if* the model were to predict they may become obese. This means that we assume our model does not know whether they already *are* obese or not as it does not take as input their current weight and height, just simply their lifestyle choices (since a patient's current body mass index can already be determined very simply during a physical exam, and, thus whether they are already obese or not).

Methods: Data

The data we needed for our objective would have to contain answers to a series of lifestyle questions that were given in correspondence to whether that person was classified as obese or not. After searching for datasets that matched the above criteria, we found such an instance that was curated by Palechor et al., 2019, for the purpose of building machine learning models [6], [7] that was originally taken and re-balanced from a dataset created by De-La-Hoz-Correa et al., 2019 [8]. Obesity data from this group (Palechor et al) contained estimations of obesity levels in people, ages from 14 to 61, from the countries of Mexico, Peru and Colombia, with diverse eating habits and physical condition. The target variable was called “NObesity” and contained seven obesity levels that people were classified as — shown in Figure 1.

	“NObesity” Class Labels	Body Mass Index
1	*Underweight*	less than 18.5
2	*Normal Weight*	between 18.5 — 24.9
3	*Overweight I*	between 25.0 — 27.45
4	*Overweight II*	between 27.45 — 29.9
5	*Obesity I*	between 30.0 — 34.9
6	*Obesity II*	between 35.0 — 34.9
7	*Obesity II*	higher than 40

Figure 1. Target Variable is called “NObesity" and has 7 possible class labels.

where body mass index is calculated as,

$$Mass\ body\ index = \frac{Weight}{height*height}$$

We should note that the original dataset from La-Hoz-Correa et al was very unbalanced towards 'normal' body mass indexes and, thus, Palechor et al had employed synthetic data generation using the Weka tool [9] with the SMOTE filter to balance their data — SMOTE (ie. Synthetic Minority Oversampling Technique) is a widely used approach to synthesize new examples of a minority class in a data set and was described by Nitesh Chawla, et al in their 2002 paper, found in our references [10]. The composition of the resulting dataset after applying the SMOTE filter was 77% synthetic and 23% collected directly from users through a surveying web platform. Since such a large portion of the dataset is synthetically generated in order to balance the dataset, we encourage readers to check out the brief summary of how this data was generated using Weka and SMOTE, given by Palechor et al in their Data section. The results of their re-balancing efforts are shown below in Figure 2.

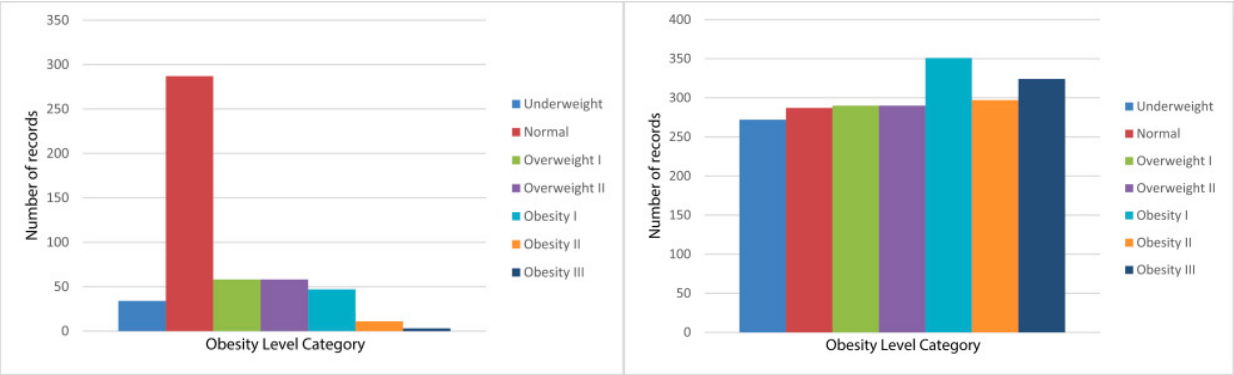

Figure 2. Class label proportions before and after balancing via applying SMOTE [7].

The resulting dataset contains 2111 records. Next, we'll go over the 16 features of the dataset, splitting them up between numerical and categorical in the below table.

		Features	*Description*	*Possible Categorical Answers*
Numerical Features	1	**Height**	*Measured in meters*	
	2	**Weight**	*Measured in Kg*	
	3	**Age**		
	4	**FCVC**	*Frequency of Consumption of Vegetables*	
	5	**NCP**	*Number of Main Meals*	
	6	**CH2O**	*Consumption of Water*	
	7	**FAF**	*Physical Activity Frequency*	
	8	**TUE**	*Time using Technological Devices*	
Categorical Features	9	**CALC**	*Consumption of Alcohol*	*"No, Sometimes, Frequently, Always"*
	10	**CAEC**	*Consumption of Food between Meals*	*"No, Sometimes, Frequently, Always"*
	11	**MTRANS**	*Transportation Used*	*"Public, Auto, Walking, Bike"*
	14	**SCC**	*Calories Consumption Monitoring*	*"Yes, No"*
	15	**FAVC**	*Frequency of Eating High Caloric Food*	*"Yes, No"*
	12	**SMOKE**		*"Yes, No"*
	13	**GENDER**		*"M, F"*
	16	**Fam History**		*"Any family member = Yes, otherwise No"*

Figure 3. 16 Numerical and Categorical Features. Height and weight were not used, explained below.

Given that there are only 2111 examples and 7 class labels, its important to note that this may impact model learning performance as there is not a lot of data per class to learn from. Below, we display the class proportions and counts of number of examples in each class in figure 4.

Class Labels	Row Counts	Proportions
Obesity I	351	16.6%
Obesity II	297	14.1%
Obesity III	324	15.3%
Overweight I	290	13.7%
Overweight II	290	13.7%
Normal Weight	287	13.6%
Underweight	272	12.9%
Mean:	**302**	**14.27%**

Figure 4. Balanced but Low Class Counts and Proportions of Data belonging to Each Class

Methods: Data Pre-Processing

All pre-processing efforts were done in Python 3.6 [11]. As mentioned in Figure 2 above, Height and Weight were dropped before building our models as training on those features makes no sense since obesity classification depends on Body Mass Index (BMI) where BMI is calculated by Weight divided by Height squared. Also, recall that we only want to train our models on features based on lifestyle-oriented choices. It is important to note that there is still the Age and Family_History_With_Overweight features after dropping Weight and Height which are not lifestyle-oriented — we decided to both include and exclude them during our model training processes in order to compare how much of an accuracy trade-off we experience when not including those two attributes. If there is a significant drop in accuracy, we reason that it does not hurt to include those features and still produce a model that achieves our original objective — we'll just have to slightly modify our original objective to allow for requiring the input of a user's age and a yes/no answer to the question of "Do you have any family members with a history of obesity?". This is quite acceptable considering that the ultimate application our model would be to apply it in a clinical setting where a health professional should not have much difficulty acquiring answers to those questions, especially if they are already pursuing the other lifestyle-oriented answers from a patient being assessed during a physical. For the rest of our pre-processing efforts, our pipeline consisted of normalizing our numerical features and encoding our categorical features. Note that there were no missing values to deal with in this dataset. No feature engineering strategies were deemed useful for this dataset either. Finally, data was split into training and test sets according to a 75:25 split — 75% of data was used for model training and 25% of data was set aside for testing. Class proportions were maintained during data splitting.

Methods: Model Types & Tuning Strategies

Model types that were trained and evaluated included Logistic Regression, Random Forest and X-Gradient Boosting. Neural networks were not considered since our dataset is quite small and neural networks require large datasets in order to learn effectively.

Logistic Regression: Logistic Regression is a type of classification model that applies a sigmoid function to a linear regression function in order to calculate the probability of whether an example belongs to a certain class or not [13]. For multi-classification, we employed the one-versus-rest algorithm provided by SciKit-Learn's python package (version 0.21.3) [12]. The classification problem can be stated as follows: Given an observation with characteristic x=($x_{1,}$….x_n), determine its class c from a predetermined set of classes {$c_{1,}$….,c_m}. The classes from c_1 to c_m are known beforehand and it can be set to our desired classes — for example, in our project, the target variable is called "NObesity" which is the set of Obesity classifications from 1 to 7 from **Figure 1**. As mentioned before, Logistic Regression is a linear regression with a transformation such that the output is always between 0 and 1 and thus can be interpreted as a probability of a person belonging to, for example, Obesity Type I, according to a person's lifestyle. Once the model has been estimated using historical data, we can use it to score or assign probabilities to new data.

Random Forest: This algorithm's strategy is to run an estimating model multiple times and then use voting across the ensemble of estimators to get a more robust result [14]; instead of sampling like bootstrapping, the random forest creates a large number of uncorrelated trees. The tree only chooses from a subset of variables for each level, stops when a minimum leaf size is reached, then repeats, changing the sample until all the trees are created, it can be used to predict the average of each tree's prediction.

X-Gradient Boosting: The theory of XGBoosting is the same as the Random Forest model type, but Random Forests require less tuning on the parameters [15]. Meanwhile, the model of XGBoosting can be smaller than the Random Forest, and the result is more robust to small sample sizes. When there is a large sample, and the models' size is not an issue, we usually use Random Forest; otherwise, using XGBoosting. In this report, it is interesting to study the performance of both models on the dataset.

Feature Tuning: When constructing our models, we wanted to determine the best combination of features that would yield the highest model accuracy. We utilized Python's itertools.combinations function within our custom-built function (called `optimalFeatureCombos()` — found in our Github link) to generate every possible combination of the 16 features. We then subtracted features, one-by-one, repeating the previous feature list generation step. We then repeated these steps again by subtracting 2 features, then 3 features, and so on, until we were left with a feature list of size 2. We then trained non-hyper-parameter tuned versions of each model type on every combination of features using our `optimalFeatureCombos()` function, which then reported the most optimal combination of features that resulted in the highest accuracy for that model type. The feature combinations that were generated and searched over, included up to 16,369 possible combinations of the original 16 feature variables. The formula we used to calculate this number is given by the n-choose-k formula:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Where, n = number of features (total is 14 since we drop Height and Weight) and k = the size of the combination, which was iterated from 16 down to 2. This formula gives the length of a list of combined items from a collection. This is different from permutations since we do not care about the order of features, we just want unique combinations of these features after reducing the feature set one-by-one.

Hyper-Parameter Tuning: For our Random Forest model and XGBoost model, we defined a grid of hyper-parameters to search over in order to find the set of hyper-parameters that yielded the highest performing models. An explanation of why we didn't do so for the Logistic Regression model is given in the Results section.

Random Forest Hyper-Parameter Space	`n_estimators`	`= [100,500,1000,10000]`
	`bootstrap`	`= [True,False]`
	`max_depth`	`= [3,5,10,20,50,75,100,None]`
	`max_features`	`= ['auto','sqrt']`
	`min_samples_leaf`	`= [1,2,4,10]`
	`min_samples_split`	`= [2,5,10]`
XGBoost Hyper-Parameter Space	`n_estimators`	`= [600,10000]`
	`colsample_bytree`	`= [0.75,0.8,0.85]`
	`max_depth`	`= [20, 50, 75, 'auto']`
	`reg_alpha`	`= [1]`
	`reg_lambda`	`= [2, 5, 10]`
	`subsample`	`= [0.55, 0.6, .65]`
	`learning_rate`	`= [0.5]`
	`gamma`	`= [.5,1,2]`
	`min_child_weight`	`= [0.01]`
	`sampling_method`	`= ['uniform']`

Figure 5. Hyper-parameter space that we searched over

Results

Baseline performance results for each of the models are shown below in Figure 6 (no tuning).

Model	Accuracy		Difference
Log Regression	**58.52%**	**49.24%**	-9%
Random Forest	**81.25%**	**75.01%**	-6%
XGBoost	**77.08%**	**71.4%**	-6%
	Training on all features	**Training on all features except Age, Family History**	

Figure 6. Baseline Performances. Note that accuracy drops significantly when we remove the Age and Family History features

Accuracy here was measured according to the proportion of correct guesses by the model. Note that Logistical Regression performed very poorly right off the bat, so we decided to drop that model before proceeding to our tuning steps. Since accuracy seemed to have dropped significantly when removing Age and Family History from the set of features, we decided to include them moving forward and slightly modify our original objective by including Age and a yes/no answer to the question of whether a person has obesity in their family's history (as discussed in the Methods: Data Pre-Processing section).

Optimal features found for each model are shown in Figure 7 below.

Model	Tuned Accuracy	Baseline Accuracy from Figure 6	
Random Forest	**82.77%**	**81.25%**	
	Features Selected:	`'Gender','Age','FCVC', 'NCP','CAEC', 'SMOKE','CH2O','SCC','FAF','TUE','CALC', 'MTRANS','family_history_with_overweight',`	*13 / 14 features*
XGBoost	**77.46%**	**77.08%**	
	Features Selected:	`'Gender','FAVC','NCP','CAEC','CH2O','SCC','FAF' ,'TUE','CALC','Age', 'family_history_with_overweight'`	*11 / 14 features*

Figure 7. Optimal features found for each model and their corresponding tuned accuracies. Note that the original set of features contained a total of 14 features.

We can see that Feature-tuning in this instance did not deliver significant performance gains. For the Random Forest model, the selected features were not reduced significantly, only removing one feature: *FAVC (Frequency of Eating High Caloric Food)*. For the XGBoost model, 3 features were removed: '*FCVC*' *(Frequency of Consumption of Vegetables),* '*MTRANS*' *(Transportation Method),* and '*SMOKE*'.

Feature importances were pulled out from each model to visualize how each feature impacted learning, shown in Figure 8.

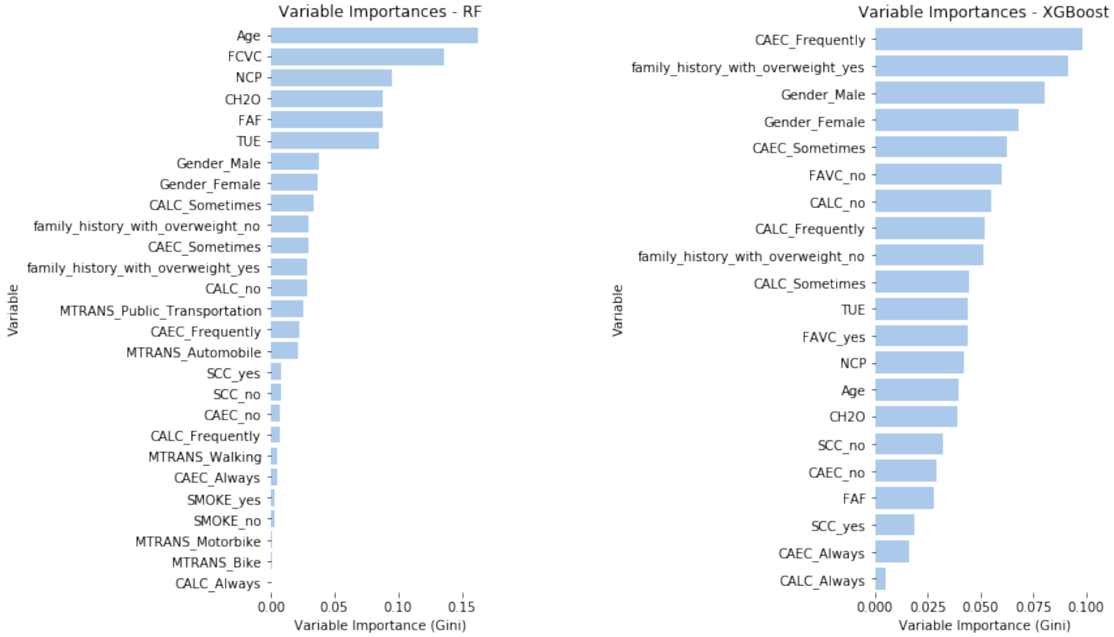

Figure 8. Feature Importance for our Random Forest model (left) and our XGBoost model (right)

Despite marginal performance gains from this step, we used the selected features found via Feature-tuning when moving to the next step: Hyper-Parameter tuning.

Random Forest Selected Hyper-Parameters	`n_estimators`	`= [10000]`
	`bootstrap`	`= [False]`
	`max_depth`	`= [50]`
	`max_features`	`= ['auto']`
	`min_samples_leaf`	`= [1]`
	`min_samples_split`	`= [2]`
XGBoost Selected Hyper-Parameters	`n_estimators`	`= [600]`
	`colsample_bytree`	`= [0.75]`
	`max_depth`	`= [20]`
	`reg_alpha`	`= [1]`
	`reg_lambda`	`= [5]`
	`subsample`	`= [0.6]`
	`learning_rate`	`= [0.5]`
	`gamma`	`= [.5]`
	`min_child_weight`	`= [0.01]`
	`sampling_method`	`= ['uniform']`

Figure 9. Selected Hyper-Parameters as a result of searching over specified hyper-parameter space

Finally, we show each model's fully-tuned performance in Figure 10, selecting the Random Forest model as our highest performing model from our tests (highlighted in blue).

		Accuracy
Feature & Hyper-Parameter-Tuned Models:	**Random Forest:**	**86.55%**
	XGBoost:	82.58%

Figure 10. Performance for our fully-tuned models. Random Forest was our selected model.

A multi-classification confusion matrix, along with precision, recall and F1-scores are shown in Figure 11 for the selected Random Forest model that scored the highest accuracy.

Confusion Matrix		**Actual**						
	Classes	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Predicted	**1**	80	0	0	6	4	2	0
	2	0	81	0	0	0	0	0
	3	2	0	70	3	2	2	0
	4	2	0	0	55	2	3	3
	5	2	0	2	3	60	5	1
	6	2	0	2	5	4	56	9
	7	0	0	0	1	0	4	55

Classes:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Average**	**Stdev**
Precision:	0.91	1.00	0.95	0.75	0.83	0.78	0.81	**0.86**	0.09
Recall:	0.87	1.00	0.89	0.85	0.82	0.72	0.92	**0.87**	0.09
F1-Score:	0.89	1.00	0.92	0.80	0.83	0.75	0.86	**0.86**	0.08
Sensitivity	0.87	1.00	0.89	0.85	0.82	0.72	0.92	**0.87**	0.09
Specificity	0.97	1.00	0.98	0.98	0.97	0.95	0.99	**0.98**	0.02
Balanced Acc	0.92	1.00	0.93	0.91	0.90	0.83	0.95	**0.92**	0.05

Figure 11. Confusion Matrix, Precision, Recall, F1-Scores, Specificity, Sensitivity, & Balanced Accuracy for all 7 classes.

Typically, ROC curves are generated for binary classification, not multi-classification, but, if we binarize the y-values, we can draw a single ROC curve (code shown in our Github link). We chose to use macro-averaging in doing so since our classes are balanced and macro-averaging gives equal weights to the classification of each label. Our Random Forest model achieved an AUROC of 0.98.

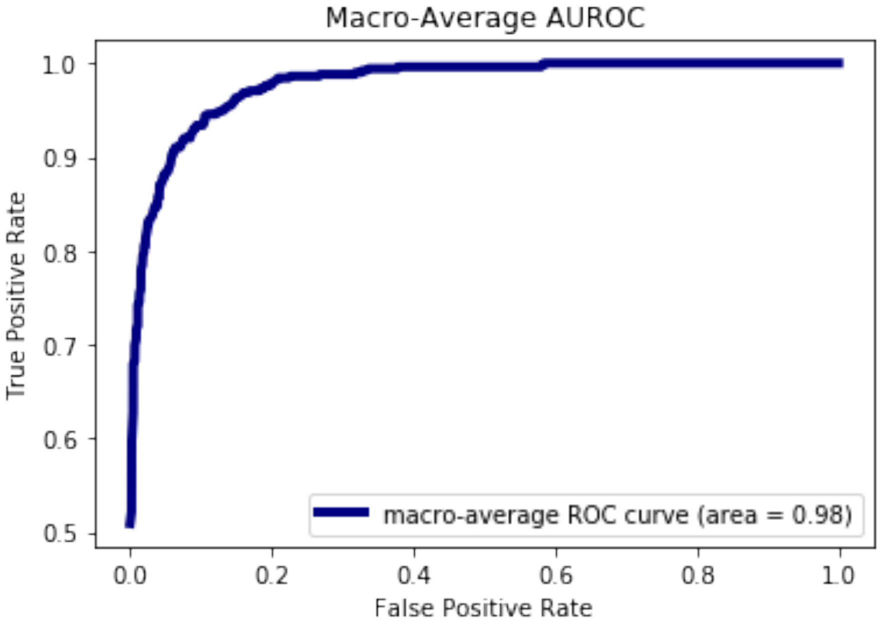

Figure 12. Macro-Average AUROC

Discussion, Limitations & Future Work

We were able to build a moderately accurate prediction model that classifies people's risk of falling into a specific class of obesity (along with a normal weight class and underweight class) if those people were to maintain their existing lifestyle choices according to the answers they give to various lifestyle choice-based questions (in addition to their age and a yes/no answer to whether anyone in their family has a history of obesity).

We believe that performance of our model is being held back by the fact that there were a small number of training examples per class (75% of ~300 = 225 training examples per class, in reference to Figure 4) for our model to learn from. Acquiring a dataset with more data would be a top priority for any future work on this project.

Overall, the model predicts moderately well and could be useful for clinical deployment, however, we must take into account several limitations. This model was trained on data from the specific regions of Mexico, Peru and Colombia, so it most likely should only be used for incoming patients from that region, especially since Palechor et al mentions that, after calculating body mass index for each individual, they were then compared to the Mexican Normativity when aggregating their dataset. We already mentioned that we would want to seek larger datasets in any potential future work — in addition to that, we would want to seek out datasets from more than one region in order to train region specific models that could then be deployed in specific clinics. It would also be interesting to see, during the data exploratory phase of this suggested future work, the specific differences in obesity rate across regions, even though the WHO states that obesity rates are increasing at dramatic rates globally (as discussed in the Introduction section).

We must also recognize that this dataset was trained mostly on synthetic data and acquiring more organic data, in addition to larger, region-specific datasets, would most likely help the quality and confidence of this model.

References

Introduction	1	No Authors Listed. *Clinical Guidelines on the Identification, Evaluation, and Treatment of Overweight and Obesity in Adults—The Evidence Report.* National Institutes of Health, 1998. https://pubmed.ncbi.nlm.nih.gov/15840860/
	2	Flegal KM, Carroll MD, Ogden CL, Curtin LR. *Prevalence and trends in obesity among US adults, 1999–2008*. JAMA, 2010. https://pubmed.ncbi.nlm.nih.gov/20071471/
	3	Ogden CL, Carroll MD, Curtin LR, Lamb MM, Flegal KM. *Prevalence of high body mass index in US children and adolescents, 2007–2008*. JAMA, 2010. https://pubmed.ncbi.nlm.nih.gov/20071470/
	4	World Health Organization, *Obesity and Overweight*, April 2020. https://www.who.int/en/news-room/fact-sheets/detail/obesity-and-overweight
	5	The Surgeon General (US). *The Surgeon General's Vision for a Healthy and Fit Nation.* Office of The Surgeon General (US), 2010. https://www.ncbi.nlm.nih.gov/books/NBK44656/#:~:text=The%20prevalence%20of%20obesity%20changed,children%20during%20the%20same%20period.
	6	Fabio Mendoza Palechor, Alexis de la Hoz Manotas. *Estimation of obesity levels based on eating habits and physical condition Data Set.* UCI Machine Learning Repository, 2019. https://archive.ics.uci.edu/ml/datasets/Estimation+ of+obesity+levels+based+on+eating+habits+and+physical+condition+
	7	Fabio Mendoza Palechor, Alexis de la Hoz Manotas. *Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico.* Data in Brief, Volume 25, 2019. https://www.sciencedirect.com/science/article/pii/S2352340919306985?via%3Dihub
Methods: Pre-Processing	8	Eduardo De-La-Hoz-Correa, Fabio E. Mendoza-Palechor, Alexis De-La-Hoz-Manotas, Roberto C. Morales-Ortega, Beatriz Adriana Sánchez Hernández. *Obesity level estimation software based on decision Trees.* J. Comput. Sci., 15, Issue 1, 2019. http://thescipub.com/abstract/10.3844/jcssp.2019.67.77
	9	Machine Learning Group at University of Waikato. *WEKA: The workbench for machine learning*. https://www.cs.waikato.ac.nz/ml/weka/
	10	N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique.* Journal Of Artificial Intelligence Research, Volume 16, 2011. https://arxiv.org/abs/1106.1813
Methods: Model Types	11	Python Core Team . *Python: A dynamic, open source programming language.* Python Software Foundation, 2020. https://www.python.org/
	12	Scikit-learn Core Team. *Scikit-learn: Machine Learning in Python.* Various Groups, 2020. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
	13	Borucka, A. (2020). Logistic regression in modeling and assessment of transport services, Open Engineering, https://doi.org/10.1515/eng-2020-0029
	14	Gerard Biau. *Analysis Of A Random Forests Model.* JMLR, 2010. https://www.jmlr.org/papers/volume13/biau12a/biau12a.pdf
	15	H. Li, Y. Cao, S. Li, J. Zhao and Y. Sun, *XGBoost Model and Its Application to Personal Credit Evaluation.* IEEE Intelligent Systems, Vol. 35, June 2020
Github Link*:		GitHub.com/link-here

*Github repository contains all code used in this project, the raw dataset, and model save file (pickle format)