USING MACHINE LEARNING FOR MEDICAL HEALTH ASSESSMENT

A report on training machine learning models and evaluating final results

Abstract

This report outlines an almost complete machine learning workflow that uses a publicly available dataset. A companion Jupyter notebook is also available for anyone wishing to review the underlying Python code.

Prepared by: Babak Eslamieh (July 3, 2025)

Table of Contents

Supervised Machine Learning – Classification

This is the final assessment project for course 3 of IBM Machine Learning Specialization.

About the data

We'll be working with Obesity dataset provided by UCI Machine Learning Repository. This dataset includes data for the estimation of obesity levels in individuals from the countries of Mexico, Peru and Colombia.

Using an online survey, data has been collected regarding each participant's eating and drinking habits, as well as their overall lifestyle. Based on the answers, an obesity level was estimated for each person.

Dataset Features

- Gender : (Male/Female)
- Age : Person's age
- Height : Person's height, in meters
- Weight : Person's weight, in kilograms
- family_history_with_overweight : Has a family member suffered or suffers from overweight? (yes/no)
- FAVC : Do you eat high caloric food frequently? (yes/no)
- FCVC : Do you usually eat vegetables in your meals? (numeric)
- NCP : How many main meals do you have daily? (numeric)
- CAEC : Do you eat any food between meals? (ordinal)
- SMOKE : Do you smoke? (yes/no)
- CH2O : How much water do you drink daily? (numeric)
- SCC : Do you monitor the calories you eat daily? (yes/no)
- FAF : How often do you have physical activity? (numeric)
- TUE : How much time do you use technological devices such as cell phone, videogames, television, computer and others? (numeric)
- CALC : How often do you drink alcohol? (ordinal)
- MTRANS : Which transportation do you usually use? (nominal)
- NObeyesdad : Obesity level (target class)

Objectives

Having more than 2 class labels, this dataset requires multi-class classification to predict one of the following obesity levels :

- Insufficient Weight
- Normal Weight
- Overweight Level I
- Overweight Level II
- Obesity Type I
- Obesity Type II
- Obesity Type III

The objectives for this project can be summarized as follows :

- Using dataset features to predict obesity level for a new person. This can be helpful as some kind of preliminary medical assessment.
- Using model interpretation techniques to gain insights about most important features that may help avoiding overweight/obesity disorders.
- Building a model with the following baseline performance :
    - Accuracy score >= 97%
    - F1 score >= 90%

Exploratory Data Analysis

To get familiar with important dataset characteristics, an exhaustive data analysis was made before preparing dataset for modeling. The following sections briefly summarize the results of this analysis.

Initial data exploration

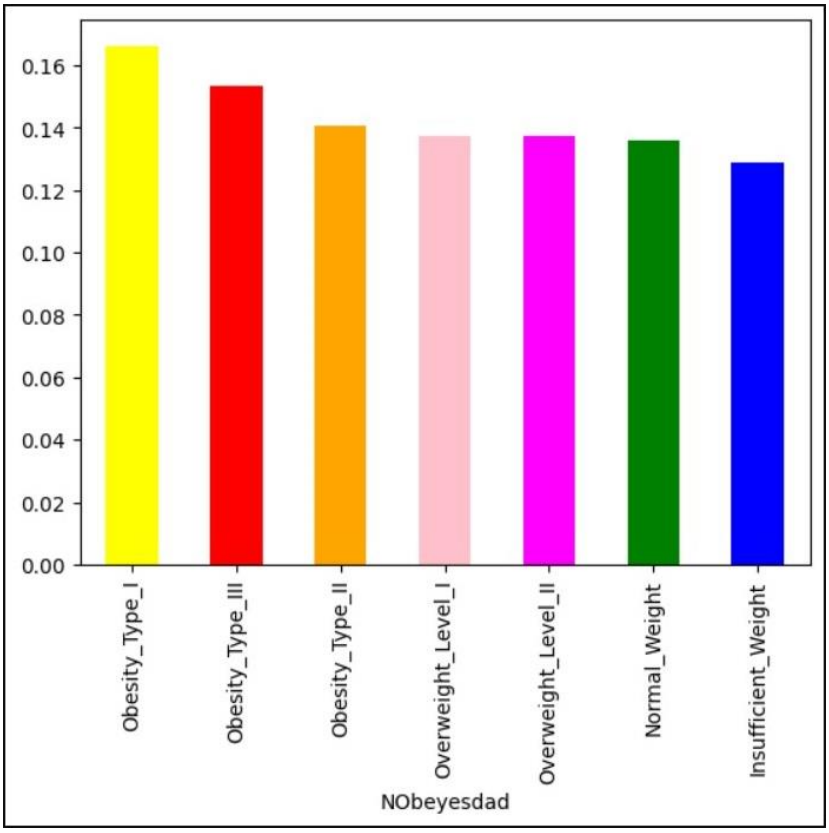
Preliminary examination of the dataset revealed the following characteristics:

- Dataset has 2111 instances, so it is rather small and using all instances in a 70/30 or 80/20 split will not slow down most learning algorithms.
- Dataset has 16 features, half numerical and half categorical.
- Dataset has all kinds of categorical features, namely: binary (yes/no), ordinal and nominal.
- Dataset has no missing values in any features, so no data imputing will be required later.
- Summary statistics of numeric features shows different value ranges, meaning we should probably scale them to improve performance and stability of linear (e.g., Logistic Regression) and distance-based (e.g., KNN and Linear SVC) models.

Class balance analysis

Description provided by the dataset donor states that this dataset includes both real and synthetic data, meaning class labels in original data has been explicitly balanced using a specific form of SMOTE resampling.

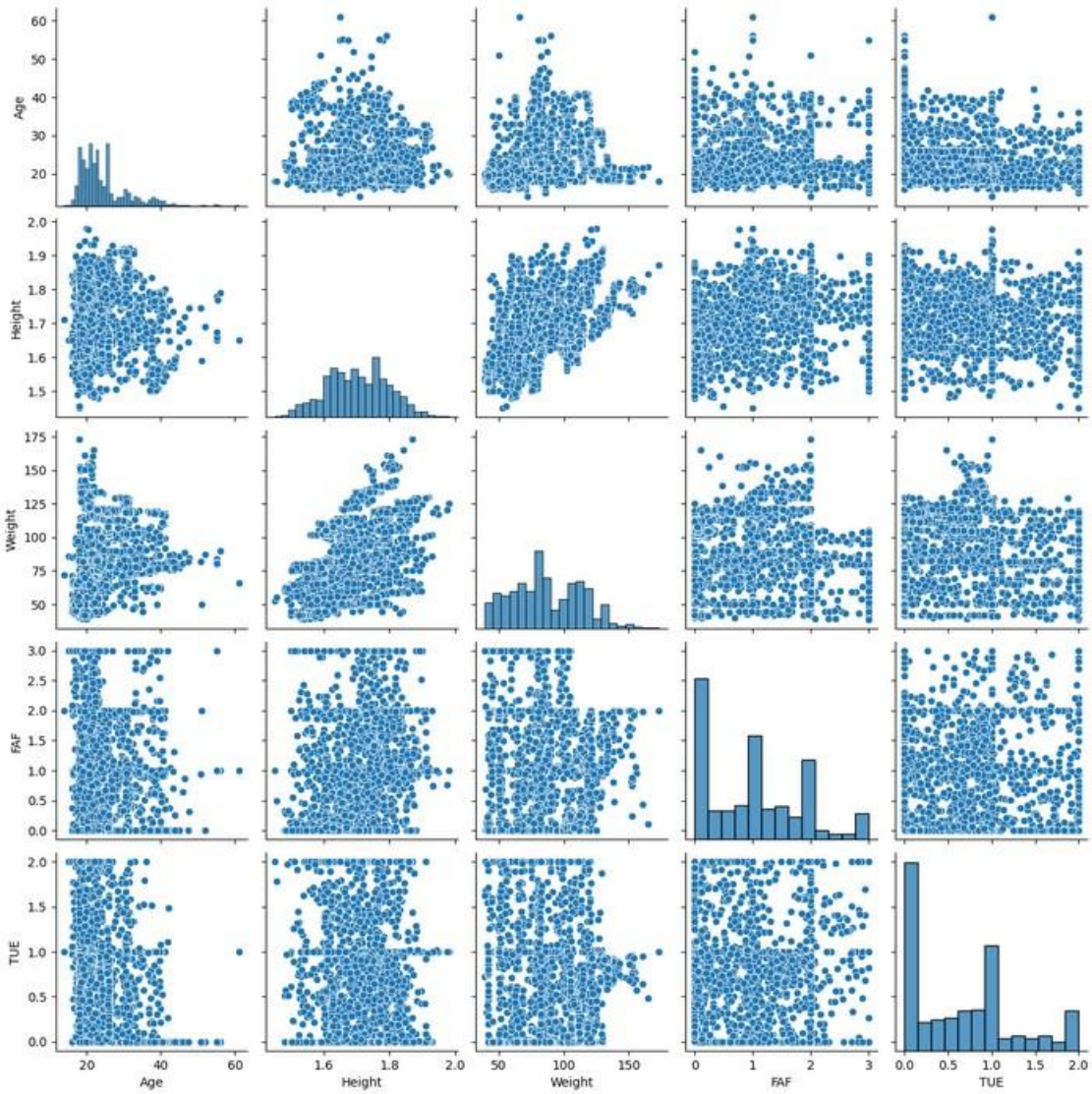
This statement was verified in code and the following image shows that class labels actually have good balance.



Correlation analysis

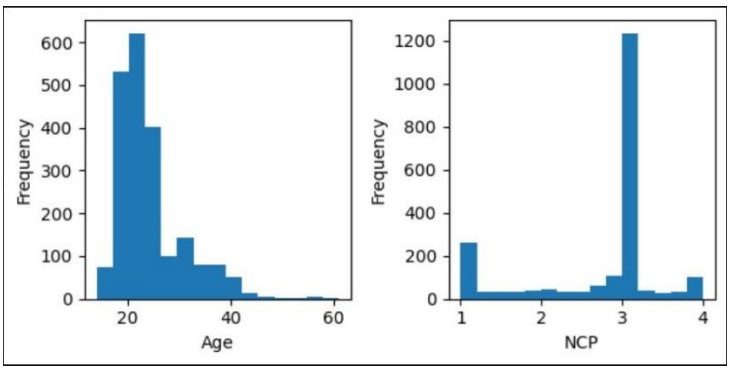
Correlation analysis using numeric features revealed that no significant correlation exists between these features. The highest positive correlation value (0.46) exists between Height and Weight, which seems almost natural except in the most extreme cases (e.g., a very tall underweight or a very short overweight person).

This can be further verified using the following (partial) pair plot.



Normality analysis

Distribution of most numeric features is fairly close to a normal distribution. Only two features (Age and NCP) have significant skewness (i.e., skew > 0.75), as visualized in the following histograms:



To improve performance of linear models like logistic regression, we may later consider transforming numeric features using a suitable transform (Log, Square Root or BoxCox).

Feature selection and engineering

Obesity dataset has different types of categorical features. The following steps summarize the feature selection/engineering approaches used for each.

Feature selection

The MTRANS (means of transportation) feature was removed, for two reasons:

- It doesn't seem to have useful information (other than Walking, the other values don't seem to contribute much to weight control)
- Values don't imply any order and must be one-hot encoded, which adds several correlated features.

Feature engineering

- As a convenience, two features were renamed ('family_history_with_overweight' renamed to 'fam_history' and 'NObeyesdad' renamed to 'Obesity')
- All binary features (Gender, FAVC, SMOKE, etc.) were encoded using LabelEncoder with values of 0 and 1.
- Ordinal features (CAEC and CALC) were encoded with a custom mapping, using the obvious ordering implied by values: no (0), Sometimes (1), Frequently (2), Always (3)
- Class labels were encoded using LabelEncoder with values from 0 to 6 (for 7 class labels)
- Whenever required by a model, all features were scaled using MinMaxScaler.

Classification models

To reach the objectives of this project, several models were trained. The whole modeling process can be summarized in the following stages:

- Train/Test split: The whole dataset was split into 70% train and 30% test sets, using stratification to maintain class balance.
- Logistic regression: A pipeline was used to scale data before fitting the model. To avoid complications with l1_ratio parameter (used by ElasticNet regularization), 3 regularization modes (None, L1 and L2) were used while tuning the model with GridSearch.
- Logistic regression: A separate model was trained with ElasticNet and was tuned using different L1/L2 ratios.
- Support Vector Machines (SVM): A pipeline was used to scale data before fitting the model. The model was tuned using different values for 'C' and 'kernel' hyperparameters.

- Decision Tree: Trained and tuned a model using 'criterion', 'max_depth' and 'min_samples_leaf' hyperparameters. To achieve better interpretability, limited 'max_depth' values to 5.
- Random Forest: Trained and tuned a model using 'n_estimators', 'criterion', 'max_depth' and 'max_features' hyperparameters. To achieve better interpretability, limited 'max_depth' values to 4.
- Gradient Boosting: Trained and tuned a model using 'n_estimators', 'criterion', 'learning_rate' and 'max_features' hyperparameters.
- Aggregated performance metrics of each (best) model for easier comparison.

Best model

After training all models and aggregating important metrics, it's time to choose the best model that will ultimately help us reach all objectives for this project. We can see the performance summary of all trained models in the following image:

model	train accuracy	test accuracy	train f1 score	test f1 score
Log. Regression	0.972241	0.973186	0.972301	0.973139
LR ElasticNet	0.884225	0.873817	0.882977	0.869432
SVM	0.965471	0.957413	0.965458	0.957026
Decision Tree	0.848341	0.807571	0.848272	0.803906
Random Forest	0.859851	0.824921	0.857012	0.819702
Gradient Boosting	0.961408	0.900631	0.961305	0.900081

We can make the following notes regarding the above performance summary:

- The "LR ElasticNet", "Decision Tree" and "Random Forest" models have inferior performance and will be discarded.
- The "SVM" model's performance is slightly lower than "Log. Regression" model, but since SVM is not sufficiently interpretable, it doesn't fit project requirements.
- As a tree-based model, "Gradient Boosting" has a high potential for interpretability, but it's overfitting the data. It probably needs more careful tuning.

Based on the above considerations, the winner is our "vanilla" **Logistic Regression** model.

Insights and key findings

Although our final model provides excellent predictive results, it may not be suitable for interpretation. Because a Logistic Regression model may lead to an interpretable model if one or more of the following conditions are met:

- We're dealing with a binary classification problem, where final model parameters (coefficients) may have a direct impact on positive or negative classes.
- Dataset does not have too many features.
- We can effectively utilize feature selection potentials in L1 regularization, without losing useful information in data.

Our dataset is not feature-intensive (it has only 16 features), but even with this many features, we cannot leverage interpretability benefits of simple models like KNN. So, we may have to reconsider tree-based models like Decision Tree and Gradient Boosting.

Next steps

After going through this data analysis and model training experiment, it's time to contemplate about what we accomplished and how to proceed.

Possible flaws

This project provided an excellent ground for experimenting with several classification models. However, one logical fault comes to mind:

Regarding the basic requirement for model interpretability, we could have skipped some hard-to-interpret models (like SVM and Random Forest) altogether. This kind of smart choice will definitely save modeler's precious time in a real project.

Action plan

- Although our "vanilla" logistic regression model performed beyond expectations, one may consider training a KNN model, after applying PCA with a few components (3 or 4). This may lead to better model interpretability.
- Although our decision tree model didn't beat baseline performance, one may consider more careful tuning of Gradient Boosting model, which may lead to a tree model that doesn't suffer from overfitting.

Acknowledgements

Thank you so much for taking the time to review and grade this project.

I'd also like to thank our brilliant instructors (especially Dr. Joseph Santarcangelo) for their excellent learning materials. It's been a real pleasure taking this course.